M8203,LUNT

M8203 STATIC DIAG. #2

CZDMSA0

AH-E235A-MC

FICHE 1 OF 1

SEP 1979

digital

MADE IN USA

```
3298
3299                    .REM @
3300
3301                                    IDENTIFICATION
3302                                    --------------
3303
3304                    PRODUCT CODE:   AC-E234A-MC
3305
3306                    PRODUCT NAME:   CZDMSA0 M8203 STATIC DIAG #2
3307
3308                    PRODUCT DATE:   JUNE 1979
3309
3310                    MAINTAINER:     DIAGNOSTIC ENGINEERING
3311
3312                    AUTHOR:         DAVID HOFFMAN
3313
3314
3315
3316
3317
3318                        THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
3319                        NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
3320                        EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO
3321                        RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.
3322
3323                        NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF
3324                        SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS
3325                        AFFILIATED COMPANIES.
3326
3327                        COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION
3328
3329                    THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:
3330
3331                        DIGITAL         PDP             UNIBUS          MASSBUS
3332                        DEC             DECUS           DECTAPE
```

CONTENTS

1.0 INTRODUCTION

THE M8203 IS A SINGLE-LINE SYNCHRONOUS LINE UNIT MODULE WHICH SUPPORTS BOTH CHARACTER ?-ORIENTED (DDCMP, BSC, ETC.) AND BIT-ORIENTED (SDLC, HDLC, ETC.) PROTOCOLS, AND WHICH IS CURRENTLY EMPLOYED IN THE DMP-11 DDCMP MULTIDROP PROJECT. 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

```
3467
3468                    DMC-11 OR KMC-11 MICROPROCESSOR
3469                    M8203 LINE UNIT AND BC08S-1 CABLE AND BERG CONNECTORS
3470                    H3254 AND H3255 TEST CONNECTORS (IF NOT PRESENT, SOME  TESTS
3471                    WILL BE SKIPPED)
3472
3473
3474                    3.0 PRELIMINARY PROGRAM REQUIREMENTS
3475
3476                    THIS  PROGRAM  OPERATES  THE  MICROPROCESSOR  EXTENSIVELY IN
3477                    ORDER  TO  TEST  THE  LINE  INIT.   FOR  THIS  REASON,  THE
3478                    MICROPROCESSOR  DIAGNOSTIC  AND  SUBSYSTEM  FUNCTIONAL TESTS
3479                    SHOULD  BE  RUN  FIRST,  AND  ANY  FAULTS  FOUND  IN  THE
3480                    MICROPROCESSOR  MODULE  SHOULD BE REPAIRED, PRIOR TO RUNNING
3481                    THE M8203 STATIC LOGIC TESTS.
3482
3483                    4.0 GENERAL PROGRAM CONSIDERATIONS
3484
3485
3486                    4.1 DIAGNOSTIC SUPERVISOR
3487
3488                    THIS PROGRAM IS COMPATIBLE WITH  THE  STANDALONE  DIAGNOSTIC
3489                    SUPERVISOR,  AND  MUST  BE LOADED TO BE CO-RESIDENT WITH THE
3490                    SUPERVISOR, OR BE PREVIOUSLY COMBINED  WITH  THE  SUPERVISOR
3491                    AND  LOADED  AS A SINGLE FILE.  IN EITHER CASE, THE COMBINED
3492                    PROGRAM WILL NOT EXCEED 16K OF MEMORY.
3493
3494
3495                    4.2 EXECUTION TIME
3496
3497                    THE MAXIMUM TIME REQUIRED TO RUN THE M8203 STATIC LOGIC  TESTS
3498                    IS ABOUT 45 SECONDS PER PASS FOR EACH UNIT.
3499
3500
3501                    4.3 XXDP+
3502
3503                    THIS  PROGRAM  MAY  BE  LOADED UNDER XXDP+, AND MAY BE RUN IN
3504                    DUMP MODE OR CHAIN MODE.
3505
3506
3507                    4.4 ACT/SLIDE
3508
3509                    THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN
3510                    IN DUMP MODE OR CHAIN MODE.
3511
3512
3513                    4.5 APT
3514
3515                    THIS  PROGRAM  MAY  BE  LOADED  BY THE APT SYSTEM (INCLUDING
3516                    APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.
3517
3518                    4.6 MEMORY MANAGEMENT
3519
3520                    MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM.  IF IT IS
3521                    INSTALLED, IT IS DISABLED BY THE PROGRAM.
3522
```

4.7 MEMORY PARITY OPTION

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

4.8 ERROR LOGGING

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

5.0 PROGRAM LOAD MEDIA

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

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

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

6.1.2 STARTING PROCEDURES

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

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

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

6.2 INITIAL DIALOGUE

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

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

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

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

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

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

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

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

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

SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

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

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

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

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

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

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

6.3.1.5 EFFECT OF START COMMAND

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

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

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

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

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

EXAMPLE:

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

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

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

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

IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

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

6.3.3 CONTINUE COMMAND

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

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

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

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

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

6.3.3.3 EFFECT OF CONTINUE COMMAND

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

6.3.4 PROCEED COMMAND

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

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

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

6.3.4.2 EFFECT OF PROCEED COMMAND

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

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

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

6.3.5 ADD COMMAND

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

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

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

6.3.5.2 EFFECT OF ADD COMMAND

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

6.3.6 DROP COMMAND

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

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

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

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.

```
3915                    6.3.10 ZFLAGS COMMAND
3916
3917                    ************************************************************
3918                    ZFL(AGS)
3919                    ************************************************************
3920
3921                    6.3.10.1 EFFECT OF ZFLAGS COMMAND
3922
3923                    ALL FLAGS ARE CLEARED.
3924
3925
3926                    6.3.11  CONTROL CHARACTERS
3927
3928                    A CONTROL C  (C)  ENTERED  DURING THE EXECUTION OF A
3929                    DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.
3930
3931                    A CONTROL Z (Z) ENTERED DURING ONE  OF  THE  THREE  OPERATOR
3932                    DIALOGUES-  HARD  CORE QUESTIONS (SEE 6.2),HARDWARE DIALOGUE
3933                    (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE
3934                    DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.
3935
3936                    A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC
3937                    CAUSES  ALL  TELETYPE  OUTPUT  TO  BE  SURPRESSED  FOR  THE
3938                    REMAINDER  OF  THE  DIAGNOSTIC  OR UNTIL ANOTHER O IS TYPED,
3939                    WHICH RESTORES NORMAL TELETYPE OUTPUT.
3940
3941
3942                    6.3.12 HARDWARE PARAMETERS
3943
3944                    THE FOLLOWING 8 QUESTIONS WILL BE ASKED ON A START  COMMAND.
3945                    THE  VALUE  LOCATED  TO THE LEFT OF THE QUESTION MARK IS THE
3946                    DEFAULT VALUE THAT  WILL  BE  TAKEN  ON  A  CARRIAGE  RETURN
3947                    RESPONSE.
3948
3949                    1.   DEVICE CSR ADDRESS : (O) 160170?
3950
3951                    THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SEL0) RESIDE
3952                    ON  THE  UNIBUS.   THE  ALLOWABLE  RANGE  IS  160000-177776
3953                    (OCTAL), AND THE DEFAULT VALUE IS 160170.
3954
3955                    2.   DEVICE VECTOR ADDRESS : (O) 300 ?
3956
3957                    THIS  IS  THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS
3958                    DEVICE.  THE ALLOWABLE RANGE IS  000-674  (OCTAL),  AND  THE
3959                    DEFAULT VALUE IS 300.
3960
3961                    3.   DEVICE PRIORITY LEVEL : (O) 5 ?
3962
3963                    THIS  IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF
3964                    THIS DEVICE WILL BE EXECUTED.  THE ALLOWABLE RANGE  IS  0-7,
3965                    AND THE DEFAULT VALUE IS 5.
3966
3967                    4.   M8203 SWITCH PACK #1 (REG 11) : (O) 0 ?
3968
3969                    THIS  IS  THE  EXPECTED  CONTENT  (OCTAL) OF SWITCH PACK #1,
3970                    WHICH RESIDES IN  INBUS  REG  11.  THE  ALLOWABLE  RANGE  IS
```

000-053, AND THE DEFAULT VALUE IS 000.

5. M8203 SWITCH PACK #2 (REG 15) : (O) 0 ?

THIS IS THE EXPECTED CONTENT (OCTAL) OF SWITCH PACK #2, WHICH RESIDES IN INBUS REG 15. THE ALLOWABLE RANGE IS 000-377, AND THE DEFAULT VALUE IS 000.

6. M8203 SWITCH PACK #3 (REG 16) : (O) 0 ?

THIS IS THE EXPECTED CONTENT (OCTAL) OF SWITCH PACK #3, WHICH RESIDES IN INBUS REG 16. THE ALLOWABLE RANGE IS 000-377, AND THE DEFAULT VALUE IS 000.

7. TURNAROUND TYPE -
(0=H3254&H3255, 1=CABLE, 2=MOD LOC, 3=MOD REM, 4=NONE)
: (O) 0 ?

THIS INDICATOR TELLS THE PROGRAM WHETHER TEST CONNECTOR(S) WILL BE MOUNTED SO THAT CERTAIN TESTS CAN BE RUN IN EXTERNAL LOOPBACK MODE. THE ALLOWABLE ANSWERS ARE 0-4, AND THE DEFAULT VALUE IS 0. 0 MEANS THAT THE H3254 AND H3255 TEST CONNECTORS WILL BE USED. 1 MEANS THAT EXTERNAL TURNAROUND WILL BE PROVIDED AT THE FAR END OF A CABLE ATTACHED TO THE J1 OR J2 CONNECTOR. 2 MEANS THAT MODEM LOCAL LOOPBACK WILL BE USED, AND 3 MEANS THAT MODEM REMOTE LOOPBACK WILL BE USED. 4 MEANS THAT NO EXTERNAL TURNAROUND WILL BE PROVIDED. WHEN 0 IS SELECTED, ALL TESTS WILL BE RUN, AND IF 1-4 IS SELECTED, CERTAIN TESTS CANNOT BE RUN, AND THE PROGRAM WILL TYPE THE NUMBER(S) OF TEST(S) TO BE SKIPPED.

8. PLEASE SELECT BAUD RATE; TYPE '0' FOR 2.4K; '1' FOR 4.8K;
'2' FOR 9.6K; '3' FOR 19.2K; '4' FOR 56K; '5' FOR 250K;
'6' FOR 500K; OR '7' FOR 1 MEG BAUD : (O) 4?

THIS IS THE BAUD RATE WHICH IS SELECTED IN THE SWITCH PACK ON THE M8203. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS 4 (FOR 56K).

6.3.13 SOFTWARE PARAMETERS

FOUR SOFTWARE PARAMETER QUESTIONS ARE ASKED BY THE M8203 STATIC LOGIC TESTS PROGRAM, PART 2. THESE QUESTIONS ARE THE FOLLOWING:

1. IS MAN. INTERVEN. DESIRED TO MOUNT TEST CONNECTOR(S) (L) N?

IF THE OPERATOR ANSWERS THE QUESTION WITH Y (YES), THE PROGRAM WILL LATER PAUSE BEFORE TESTING EACH LOGICAL UNIT AND INFORM THE OPERATOR TO INSTALL THE APPROPRIATE TEST CONNECTOR(S) ON THAT UNIT, AND THEN PROCEED TO TEST THAT UNIT. IF THE OPERATOR ANSWERS N (NO) TO THE ABOVE QUESTION, THE PROGRAM WILL PERFORM TESTING ON ALL UNITS WITHOUT ALLOWING MANUAL INTERVENTION BETWEEN UNITS. IN THIS CASE,

ALL TEST CONNECTOR(S) ON ALL UNITS SHOULD BE INSTALLED PRIOR TO RUNNING THE PROGRAM.

2. SHOULD SWITCH PACK AND AX3-15 PRINTOUT BE ALLOWED (L) N ?

IF THE OPERATOR ANSWERS YES, THE PROGRAM WILL ALLOW THE PRINTOUT OF SWITCH PACKS 1-3 AND MODEM INTERFACE REG AX3-15 ON ANY PASS IN WHICH THE CORRESPONDING TESTS ARE RUN. THE DEFAULT IS NO, WHICH ONLY ALLOWS THE PRINTOUT ON THE FIRST PASS AFTER LOADING, IF THE TESTS ARE RUN.

3. SHOULD SWITCH PACK TESTS BE ALLOWED (L) N ?

IF THE OPERATOR ANSWERS YES, THE PROGRAM WILL ALLOW THE READING AND COMPARISON OF SWITCH PACKS 1-3 TO VALUES ENTERED INTO THE HARDWARE P-TABLE FOR THIS UNIT, IF THE CORRESPONDING TEST IS RUN. IF ALLOWED, SWITCH PACK ERRORS WILL BE REPORTED. THE DEFAULT IS NO, AND THE TESTS ARE NOT RUN.

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

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

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

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

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

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

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

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

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

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D) ? 16

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

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

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

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

```
4139
4140
4141                    THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT
4142                    16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION
4143                    (NAMELY QUESTION 2).
4144
4145
4146
```

```
4148
4149
4150
4151                    7.0 DEVICE INFORMATION TABLES
4152
4153
4154
4155                    ;*****************************************************************************
4156                    * MAINTENANCE REGISTER - BSEL1
4157                    ;*****************************************************************************
4158                    RUN     = BIT7
4159                    MCLR    = BIT6
4160                    STEPLU  = BIT4
4161                    LULOOP  = BIT3
4162                    ROMO    = BIT2
4163                    ROMI    = BIT1
4164                    STEPMP  = BIT0
4165
4166                    ;*****************************************************************************
4167                    * OBUS REG 10 - TRANSMITTER BUFFER
4168                    ;*****************************************************************************
4169                    TX7     = BIT7
4170                    TX6     = BIT6
4171                    TX5     = BIT5
4172                    TX4     = BIT4
4173                    TX3     = BIT3
4174                    TX2     = BIT2
4175                    TX1     = BIT1
4176                    TX0     = BIT0
4177
4178                    ;*****************************************************************************
4179                    * OBUS REG 11
4180                    ;*****************************************************************************
4181                    OC      = BIT7
4182                    GOAH    = BIT3
4183                    ABORT   = BIT2
4184                    EOM     = BIT1
4185                    SOM     = BIT0
4186
4187                    ;*****************************************************************************
4188                    * OBUS REG 12
4189                    ;*****************************************************************************
4190                    IC      = BIT7
4191                    BPOLL   = BIT6
4192                    LULP    = BIT5
4193
4194                    ;*****************************************************************************
4195                    * OBUS REG 13
4196                    ;*****************************************************************************
4197                    POLL    = BIT7
4198                    DTR     = BIT6
4199                    SELFR   = BIT5
4200                    HDX     = BIT4
4201                    MAINT1  = BIT3
4202                    MAINT2  = BIT2
4203                    SELSBY  = BIT1
```

```
4204
4205                    ;*****************************************************************************
4206                    * OBUS REG 14
4207                    ;*****************************************************************************
4208                    TXEN    = BIT6
4209                    DISSI   = BIT5
4210                    RDAX    = BIT4
4211                    WAX     = BIT3
4212                    ENAX    = BIT2
4213                    AX2     = BIT1
4214                    AX1     = BIT0
4215
4216                    ;*****************************************************************************
4217                    * OBUS REG 17
4218                    ;*****************************************************************************
4219                    CRC2    = BIT7
4220                    CRC1    = BIT6
4221                    IDLE    = BIT5
4222                    SECA    = BIT4
4223                    STRIP   = BIT3
4224                    RDALL   = BIT2
4225                    IERR    = BIT1
4226                    DDCMP   = BIT0
4227
4228                    ;*****************************************************************************
4229                    * IBUS REG 10 - RECEIVER BUFFER
4230                    ;*****************************************************************************
4231                    RX7     = BIT7
4232                    RX6     = BIT6
4233                    RX5     = BIT5
4234                    RX4     = BIT4
4235                    RX3     = BIT3
4236                    RX2     = BIT2
4237                    RX1     = BIT1
4238                    RX0     = BIT0
4239
4240                    ;*****************************************************************************
4241                    * IBUS REG 11
4242                    ;*****************************************************************************
4243                    OC      = BIT7
4244                    OACT    = BIT6
4245                    SW3     = BIT5
4246                    ORDY    = BIT4
4247                    SW2     = BIT3
4248                    SW1     = BIT2
4249                    SW0     = BIT1
4250                    UNRR    = BIT0
4251
4252                    ;*****************************************************************************
4253                    * IBUS REG 12
4254                    ;*****************************************************************************
4255                    IC      = BIT7
4256                    IACT    = BIT6
4257                    LULP    = BIT5
4258                    IRDY    = BIT4
4259                    OVRR    = BIT3
```

```
4260                            RAB     = BIT2
4261                            EBLK    = BIT1
4262                            BCC     = BIT0
4263
4264                            ;*****************************************************************************
4265                            * IBUS REG 13
4266                            ;*****************************************************************************
4267                            RING    = BIT7
4268                            DTR     = BIT6
4269                            RTS     = BIT5
4270                            HDX     = BIT4
4271                            MODR    = BIT3
4272                            CS      = BIT2
4273                            STBY    = BIT1
4274                            CARR    = BIT0
4275
4276                            ;*****************************************************************************
4277                            * IBUS REG 14
4278                            ;*****************************************************************************
4279                            READY   = BIT7
4280                            TXEN    = BIT6
4281                            DISSI   = BIT5
4282                            RDAX    = BIT4
4283                            WAX     = BIT3
4284                            ENAX    = BIT2
4285                            AX2     = BIT1
4286                            AX1     = BIT0
4287
4288                            ;*****************************************************************************
4289                            * IBUS REG 17
4290                            ;*****************************************************************************
4291                            SIGR    = BIT7
4292                            SIGQ    = BIT6
4293                            TXDATA  = BIT5
4294                            OCOR    = BIT4
4295                            ICIR    = BIT3
4296                            TESTMD  = BIT2
4297                            MCLK  = BIT1
4298                            DDCMP   = BIT0
4299
4300                            ;*****************************************************************************
4301                            * AX0-15 - USYRT REG 0 (READ ONLY)
4302                            ;*****************************************************************************
4303                            RX7     = BIT7
4304                            RX6     = BIT6
4305                            RX5     = BIT5
4306                            RX4     = BIT4
4307                            RX3     = BIT3
4308                            RX2     = BIT2
4309                            RX1     = BIT1
4310                            RX0     = BIT0
4311
4312                            ;*****************************************************************************
4313                            * AX0-16 - USYRT REG 1 (READ ONLY)
4314                            ;*****************************************************************************
4315                            RERR    = BIT7
```

```
4316                            ASBC2   = BIT6
4317                            ASBC1   = BIT5
4318                            ASBC0   = BIT4
4319                            ROR     = BIT3
4320                            RABT    = BIT2
4321                            REOM    = BIT1
4322                            RSOM    = BIT0
4323
4324                            ;*****************************************************************************
4325                            * AX1-15 - USYRT REG 2
4326                            ;*****************************************************************************
4327                            TX7     = BIT7
4328                            TX6     = BIT6
4329                            TX5     = BIT5
4330                            TX4     = BIT4
4331                            TX3     = BIT3
4332                            TX2     = BIT2
4333                            TX1     = BIT1
4334                            TX0     = BIT0
4335
4336                            ;*****************************************************************************
4337                            * AX1-16 - USYRT REG 3
4338                            ;*****************************************************************************
4339                            TERR    = BIT7
4340                            TXGA    = BIT3
4341                            TXAB    = BIT2
4342                            TEOM    = BIT1
4343                            TSOM    = BIT0
4344
4345                            ;*****************************************************************************
4346                            * AX2-15 - USYRT REG 4
4347                            ;*****************************************************************************
4348                            SYN7    = BIT7
4349                            SYN6    = BIT6
4350                            SYN5    = BIT5
4351                            SYN4    = BIT4
4352                            SYN3    = BIT3
4353                            SYN2    = BIT2
4354                            SYN1    = BIT1
4355                            SYN0    = BIT0
4356                            SYNCH   = 226
4357
4358                            ;*****************************************************************************
4359                            * AX2-16 - USYRT REG 5
4360                            ;*****************************************************************************
4361                            APA     = BIT7
4362                            DDC     = BIT6
4363                            STR     = BIT5
4364                            SEC     = BIT4
4365                            IDL     = BIT3
4366                            CRCTY2  = BIT2
4367                            CRCTY1  = BIT1
4368                            CRCTY0  = BIT0
4369
4370                            ;*****************************************************************************
4371                            * AX3-15 - USYRT REG 6
```

```
4372                    ;*****************************************************************************
4373                    I422    = BIT7
4374                    XYZ     = BIT6
4375                    V35     = BIT4
4376                    INTGRL  = BIT3
4377                    OP      = BIT1
4378                    TEST    = BIT0
4379                    AX315U  = I422!XYZ!V35!INTGRL!OP
4380
4381                    ;*****************************************************************************
4382                    * AX3-16 - USYRT REG 7
4383                    ;*****************************************************************************
4384                    TXLEN2  = BIT7
4385                    TXLEN1  = BIT6
4386                    TXLEN0  = BIT5
4387                    RXLEN2  = BIT2
4388                    RXLEN1  = BIT1
4389                    RXLEN0  = BIT0
4390
4391
4392
4393
4394
4395
```

```
4397
4398
4399                    8.0 TEST DESCRIPTIONS
4400
4401
4402
4403
4404
4405                    ;*****************************************************************************
4406                        TEST 1 - BIT STUFFING TEST
4407                    *
4408                    * THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
4409                    * INITIATED IN BIT MODE . TWO LEADING FLAGS ARE SENT,
4410                    * FOLLOWED BY ALL SIXTEEN CHARS IN DATA PATTERN S. THIS PATTERN
4411                    * CONSISTS OF CHARACTERS WHICH REQUIRE NO BIT STUFFING AND CHARACTERS
4412                    * WHICH REQUIRE BIT STUFFING INDIVIDUALLY AND IN COMBINATION WITH
4413                    * ADJACENT CHARACTERS. ALL 16 CHARACTERS ARE READ AND COMPARED
4414                    * BY THE RECEIVER.
4415                    *   PATTERN S = 000,017,036,074,170,360,037,076,174,370,077,176,374,
4416                    *               177,376,377
4417                    ;*****************************************************************************
4418
4419
4420
4421
4422
4423                    ;*****************************************************************************
4424                        TEST 2 - RCV OVERRUN ERROR SET AND CLEAR TEST
4425                    *
4426                    * IN THIS TEST, A RCV OVERRUN ERROR IS FORCED IN EACH OF 2 SUBTESTS.
4427                    * IN THE FIRST, A MESSAGE IS INITIATED, 64 001 CHARS ARE SENT, AND THE
4428                    * RECEIVER IS NOT SERVICED IN RESPONSE TO THE USYRT RCV FLAG, WHICH CAUSES RCV
4429                    * OVERRUN TO SET. THEN, A CHECK IS MADE TO INSURE THAT OVRR IS NOT
4430                    * CLEARED BY THE LINE UNIT READING THE USYRT STATUS.
4431                    * THEN, IC IS SET TO CLEAR THE ERROR, AND THIS IS VERIFIED.
4432                    *
4433                    * IN THE SECOND SUBTEST, RCV OVRUN IS FORCED AGAIN, AND A MASTER CLEAR
4434                    * IS ISSUED TO CLEAR THE ERROR, AND THIS IS VERIFIED.
4435                    ;*****************************************************************************
4436
4437
4438
4439
4440
4441                    ;*****************************************************************************
4442                        TEST 3 - ABORT SEQUENCE TEST
4443                    *
4444                    * SET BIT MODE, CRC, AND ENABLE THE DEVICE FOR
4445                    * TRANSMIT AND RECEIVE. SEND 2 FLAGS AND 4 DATA CHARS (001).
4446                    * AS THE FIRST DATA CHAR IS BEING TRANSMITTED,
4447                    * SET THE ABORT BIT (REG 11).
4448                    * ON THE RECEIVER SIDE, CHECK FOR RECEPTION OF THE FIRST DATA CHAR
4449                    * AND THEN THE SETTING OF RAB AND REOM A CHAR TIME LATER.
4450                    * ALSO, CHECK FOR IACT = 0. THEN, CHECK THAT RAB
4451                    * IS CLEARED BY READING THE USYRT STATUS, TRANSMITTING A NEW MSG,
4452                    * RECEIVING THE FIRST CHAR (003) AND CHECKING FOR RAB CLEARED.
```

```
4453                            *
4454                            * REPEAT THE ABOVE SEQUENCE, SET IC, AND CHECK THAT
4455                            * THIS CLEARS RAB.
4456                            *
4457                            * REPEAT THE ABOVE SEQUENCE, ISSUE MASTER CLEAR, CHECK THAT THIS
4458                            * CLEARS RAB.
4459                            *
4460                            ;*************************************************************************
4461
4462
4463
4464
4465
4466                            ;*************************************************************************
4467                                TEST 4 - ABORT AND IDLE FLAGS TEST
4468                            *
4469                            * TRANSMIT THE SAME ABORT SEQUENCE AS IN THE PREVIOUS TEST, BUT
4470                            * WITH THE IDLE BIT SET. CHECK THAT FLAGS ARE SENT AND RECEIVED
4471                            * (NOT ABORT CHARACTERS) BY VERIFYING THAT RAB DOES
4472                            * NOT SET, AND THAT THE MESSAGE TERMINATES WITH EBLK = 1.
4473                            ;*************************************************************************
4474
4475
4476
4477
4478
4479                            ;*************************************************************************
4480                                TEST 5 - TRANSMITTER UNDERRUN ERROR, IDLE ABORT CHARS, BIT MODE
4481                            *
4482                            * A MESSAGE IS INITIATED IN BIT MODE, 4 001 CHARS ARE SENT, AND THE TRANSMITTER
4483                            * IS NOT SERVICED IN RESPONSE TO THE LAST TX FLAG, WHICH CAUSES TX
4484                            * UNDERRUN ERROR TO SET. ON THE RECEIVER SIDE, CHECK THAT THE DATA
4485                            * CHAR IS RECEIVED, AND THAT 8 CYCLES LATER THE RAB BIT SETS, AND
4486                            * THE DEVICE IDLES ABORT CHARACTERS.
4487                            ;*************************************************************************
4488
4489
4490
4491
4492
4493                            ;*************************************************************************
4494                                TEST 6 - RECEIVER DISABLE TEST
4495                            *
4496                            * TRANSMIT AND RECEIVE ARE ENABLED IN BIT MODE, AND 2 FLAGS
4497                            * ARE SENT, FOLLOWED BY 5 252 DATA CHARS. AFTER THE SECOND DATA CHAR HAS BEGUN
4498                            * TO BE RECEIVED, IC IS SET.
4499                            * THEN, THE PROGRAM CHECKS THAT A USYRT RCV FLAG IS NOT GENERATED, AND
4500                            * THE RECEIVER DATA PATH STOPS OPERATING IN THE MIDDLE OF THE CHAR.
4501                            ;*************************************************************************
4502
4503
4504
4505
4506
4507                            ;*************************************************************************
4508                                TEST 7 - ASSEMBLED BIT COUNT TEST
```

```
4509                            *
4510                            * THE FOLLOWING SEQUENCE IS PERFORMED 8 TIMES, EACH TIME USING A
4511                            * DIFFERENT TX CHAR LENGTH (FROM 2 TO 8 BITS) AND A RCV CHAR LENGTH = 8
4512                            * BITS :
4513                            * A MESSAGE IS INITIATED IN BIT MODE, NO CRC.
4514                            * 2 FLAGS ARE SENT, FOLLOWED BY 3 000 DATA CHARACTERS AND A
4515                            * TERMINATING FLAG. AFTER THE RECEIVER HAS RECEIVED THE MESSAGE, AX0-16
4516                            * IS READ TO RETRIEVE THE ASSEMBLED BIT COUNT. THIS COUNT IS CHECKED TO INSURE
4517                            * THAT IT IS CORRECT FOR THE TX CHAR LENGTH USED IN THAT TRANSMISSION.
4518                            ;*********************************************************************
4519
4520
4521
4522
4523
4524                            ;*********************************************************************
4525                                TEST 8 - SECONDARY STATION ADDRESS BIT TEST
4526                            *
4527                            * FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
4528                            * BIT MODE, AND THE SECA BIT (REG 17) IS SET.
4529                            * 2 FLAGS ARE SENT, FOLLOWED BY 252, 000, AND A TERMINATING FLAG.
4530                            * THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
4531                            * RECEIVED.
4532                            *
4533                            * NEXT, THE SECONDARY STATION ADDRESS BITS IN AX2-15 ARE LOADED
4534                            * WITH THE FIRST WORD OF DATA PATTERN T . 2 FLAGS ARE SENT,
4535                            * FOLLOWED BY THE FIRST WORD OF DATA PATTERN T, A 000 CHAR,
4536                            * AND A TERMINATING FLAG.
4537                            * THEN, THE RCV'D DATA IS CHECKED TO MAKE SURE THAT THE SEC STATION
4538                            * ADDRESS IS RCV'D AS THE FIRST DATA CHAR, FOLLOWED BY 000.
4539                            *
4540                            * THEN, THE SUBTEST IS REPEATED FOR EACH OF THE REMAINING WORDS OF
4541                            * DATA PATTERN T.
4542                            *   PATTERN T = 000,125,252,176,177
4543                            ;*********************************************************************
4544
4545
4546
4547
4548
4549                            ;*********************************************************************
4550                                TEST 9 - RDALL (ALL PARTIES ADDRESS) BIT TEST
4551                            *
4552                            * FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
4553                            * BIT MODE, AND THE SECA BIT IS SET.
4554                            * 2 FLAGS ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
4555                            * THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
4556                            * RECEIVED.
4557                            * NEXT, THE RDALL BIT IN REG 17 IS SET TO 1. 2 FLAGS
4558                            * ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
4559                            * THEN, THE REC'D DATA IS CHECKED TO MAKE SURE THAT 377
4560                            * IS REC'D AS THE FIRST DATA CHAR, FOLLOWED BY 125.
4561                            ;*********************************************************************
4562
4563
4564
```

```
4565
4566
4567                    ;*****************************************************************************
4568                        TEST 10 - INSERT ERROR (IERR) BIT TEST - CHAR MODE, NO CRC
4569                    *
4570                    * THE LINE UNIT IS PLACED IN DDCMP MODE WITH NO ERROR DETECTION, AND 2
4571                    * SYNCHS, A 000 CHAR, A 377 CHAR, AND 2 SYNCHS ARE LOADED INTO THE
4572                    * TRANSMITTER SILO. THEN, THE LU IS CLOCKED UNTIL THE 2ND BIT OF THE 000
4573                    * CHAR IS ABOUT TO BE SENT AND THE IERR BIT IS SET FOR A CLOCK TIME AND
4574                    * THEN CLEARED. IN THE SAME WAY, IERR IS SET PRIOR TO THE SENDING OF THE 4TH
4575                    * AND 5TH BITS OF THE 000 CHAR. IT IS ALSO SET FOR THE SENDING OF THE FIRST
4576                    * 4 BITS OF THE 377 CHAR. THE PROGRAM READS THE FIRST RCV'D CHAR FROM AX0
4577                    * AND CHECKS IT TO BE 032, AND READS THE 2ND CHAR AND CHECKS IT TO BE 377.
4578                    * THEN, A MASTER CLEAR IS DONE TO IDLE THE DEVICE.
4579                    ;*****************************************************************************
4580
4581
4582
4583
4584
4585                    ;*****************************************************************************
4586                        TEST 11 - SWITCH PACK PRINTOUT AND TEST
4587                    *
4588                    * - READ AND PRINT SWITCH PACK  1 :
4589                    * THE PROGRAM READS REG 11 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
4590                    * (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
4591                    * THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
4592                    * SWITCHES ARE IN BITS 1,2,3,5.
4593                    *
4594                    * - READ AND PRINT SWITCH PACK  2 :
4595                    * THE PROGRAM READS REG 15 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
4596                    * (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
4597                    * THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
4598                    * SWITCHES ARE IN BITS 0-7.
4599                    *
4600                    * - READ AND PRINT SWITCH PACK  3 :
4601                    * THE PROGRAM READS REG 16 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
4602                    * (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
4603                    * THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
4604                    * SWITCHES ARE IN BITS 0-7.
4605                    ;*****************************************************************************
4606
4607
4608
4609
4610
4611                    ;*****************************************************************************
4612                        TEST 12 - REG AX3-15 PRINTOUT
4613                    *
4614                    * IN THIS TEST, REG AX3-15 IS READ AND THE CONTENTS PRINTED OUT IF DESIRED BY
4615                    * THE OPERATOR, AS INDICATED IN THE SOFTWARE P-TABLE. THE DEFAULT IS TO NOT
4616                    * PRINT THE REG.
4617                    ;*****************************************************************************
4618
4619
4620
```

```
4621
4622
4623                    ;*****************************************************************************
4624                        TEST 13 - CRC GENERATION TEST
4625                    *
4626                    * - CRC-16, CHAR MODE:
4627                    * THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE WITH CRC-16 SELECTED -
4628                    * 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
4629                    * TO CLOCK THE DATA.    AT THE END OF THE MESSAGE THE
4630                    * PROGRAM CHECKS FOR BCC = 1 (IN REG 12) INDICATING NO ERROR.
4631                    *
4632                    * - CRC-CCITT - 1'S PRESET:
4633                    * THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT
4634                    * THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
4635                    *
4636                    * - CRC-CCITT - 0'S PRESET:
4637                    * THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT
4638                    * THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
4639                    ;*****************************************************************************
4640
4641
4642
4643
4644
4645                    ;*****************************************************************************
4646                        TEST 14 - CRC ERROR DETECTION TEST
4647                    *
4648                    * - CRC-16, CHAR MODE :
4649                    * THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE, WITH CRC-16 SELECTED -
4650                    * 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
4651                    * TO CLOCK THE DATA. JUST BEFORE THE FIRST BIT OF THE LAST 000 CHAR IS SENT,
4652                    * THE IERR BIT IS SET IN REG 17 TO CAUSE A 1 TO BE SENT, INTRODUCING A DATA
4653                    * ERROR. AT THE END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 0, INDICATING
4654                    * AN ERROR.
4655                    *
4656                    * - CRC-CCITT - 1'S PRESET :
4657                    * THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT THE
4658                    * END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
4659                    *
4660                    * - CRC-CCITT - 0'S PRESET :
4661                    * THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT THE
4662                    * END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
4663                    ;*****************************************************************************
4664
4665
4666
4667
4668
4669                    ;*****************************************************************************
4670                        TEST 15 - VRC PARITY GENERATION TEST
4671                    *
4672                    * SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
4673                    * THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC AND 7-BIT CHARS SELECTED.
4674                    * THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
4675                    * OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
4676                    * FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1 AND FOR THE
```

```
4677                    * LAST 4 CHARS IT SHOULD = 0.
4678                    *
4679                    * SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
4680                    * THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
4681                    * THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
4682                    * OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
4683                    * FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0 AND FOR THE
4684                    * LAST 4 CHARS IT SHOULD = 1.
4685                    *
4686                    *    DATA PATTERN Q = 000,120,125,137,040,052,057,177
4687                    ;*****************************************************************************
4688
4689
4690
4691
4692
4693                    ;*****************************************************************************
4694                        TEST 16 - VRC ERROR DETECTION TEST
4695                    *
4696                    * SUBTEST 1 - FORCING OF BCC USING ODD VRC
4697                    * THE LINE UNIT IS PLACED IN CHAR MODE WITH ODD VRC AND 7-BIT CHARS SELECTED.
4698                    * THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
4699                    * 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
4700                    * CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
4701                    * NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
4702                    * RECEIVED (INDICATING AN ERROR).
4703                    *
4704                    * SUBTEST 2 - FORCING OF BCC USING EVEN VRC
4705                    * THE LINE UNIT IS PLACED IN CHAR MODE WITH EVEN VRC AND 7-BIT CHARS SELECTED.
4706                    * THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
4707                    * 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
4708                    * CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
4709                    * NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
4710                    * RECEIVED (INDICATING AN ERROR).
4711                    *
4712                    *   DATA PATTERN R = 000,100,120,124,164,172,176,177,000,100,120,124,164,
4713                    *                    172,176.
4714                    ;*****************************************************************************
4715
4716
4717
4718
4719
4720                    ;*****************************************************************************
4721                        TEST 17 - INTEGRAL MODEM INTERFACE TEST - CHAR MODE, CRC
4722                    *
4723                    * THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM IN AX3-15, AND A
4724                    * MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
4725                    * ON THE LINE UNIT OR AT THE CABLE. THE MESSAGE CONSISTS OF
4726                    * 2 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
4727                    * UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
4728                    * SKIPPED FOR THAT UNIT.
4729                    ;*****************************************************************************
4730
4731
4732
```

```
4733
4734
4735                    ;*****************************************************************************
4736                        TEST 18 - V.35 MODEM INTERFACE TEST - CHAR MODE, CRC
4737                    *
4738                    * THE V.35 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
4739                    * MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
4740                    * ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
4741                    * OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
4742                    * 2 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
4743                    * UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
4744                    * SKIPPED FOR THAT UNIT.
4745                    ;*****************************************************************************
4746
4747
4748
4749
4750
4751                    ;*****************************************************************************
4752                        TEST 19 - RS 232C AND RS 423 MODEM INTERFACE TEST - CHAR MODE, CRC
4753                    *
4754                    * THE RS232C  RS423 (XYZ) MODEM INTERFACE IS SELECTED BY THE PROGRAM IN
4755                    * AX3-15, AND A MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURN-
4756                    * AROUND CONNECTOR ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
4757                    * OR A MODEM TEST MODE. THE MESSAGE CONSISTS
4758                    * OF 2 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE
4759                    * P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS
4760                    * PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
4761                    ;*****************************************************************************
4762
4763
4764
4765
4766
4767                    ;*****************************************************************************
4768                        TEST 20 - RS 422 MODEM INTERFACE TEST - CHAR MODE, CRC
4769                    *
4770                    * THE RS 422 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
4771                    * MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
4772                    * ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
4773                    * OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
4774                    * 2 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
4775                    * UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
4776                    * SKIPPED FOR THAT UNIT.
4777                    ;*****************************************************************************
4778
4779
4780
4781
4782
4783                    ;*****************************************************************************
4784                        TEST 21 - HALF-DUPLEX BIT (HALF DUPX) TEST
4785                    *
4786                    * THIS TEST VERIFIES THAT SETTING HALF-DUPLEX BIT IN REG 13 INHIBITS
4787                    * LOADING OF THE USYRT TRANSMITTER FROM THE TRANSMITTER SILO.
4788                    * A MASTER CLEAR IS ISSUED, DDCMP MODE IS ENTERED, AND THE HALF DUPX
```

```
4789                    * BIT IN REG 13 IS SET. A MESSAGE IS LOADED INTO THE TX SILO
4790                    * CONSISTING OF 2 SYNCHS, 000,125,252,377,000, AND 2 MORE SYNCHS.
4791                    * THE LINE UNIT IS THEN CLOCKED EXTENSIVELY, AND THE TX SILO IS CHECKED TO
4792                    * BE STILL LOADED (NO CHARS SHOULD HAVE BEEN REMOVED) AND THE RECEIVER
4793                    * IS MONITORED TO INSURE THAT NO RCV FLAGS ARE GENERATED.
4794                    ;*************************************************************************
4795
4796
4797
4798
4799
4800                    ;*************************************************************************
4801                        TEST 22 - HALF-DUPLEX RCV DISABLED TEST WITH SILOS DISABLED
4802                    *
4803                    * THIS TEST SENDS A MESSAGE IN HDX, CHAR MODE, WITH NO ERROR DETECTION, AND
4804                    * THE SILOS DISABLED. THE MSG CONSISTS OF 2 SYNCHS AND 2 000 CHARS.
4805                    * THE DATA IS SENT WITH LULOOP SET FOR TTL DATA LOOPBACK. THE PROGRAM CHECKS
4806                    * THAT THE RECEIVER NEVER BECOMES ACTIVE, BECAUSE THE RCV CLOCK IS INHIBITED
4807                    * WHEN THE HDX BIT IS SET.
4808                    ;*************************************************************************
4809
4810
4811
4812
4813
4814                    ;*************************************************************************
4815                        TEST 23 - INTERACTION OF MODEM CONTROL BITS
4816                    *
4817                    * THIS TEST WILL BE RUN ONLY IF THE P-TABLE FOR THIS UNIT INDICATES THAT
4818                    * THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED. OTHERWISE, THE TEST WILL
4819                    * BE SKIPPED FOR THE UNIT.
4820                    * THE FOLLOWING SUBTESTS ARE PERFORMED:
4821                    * - A MASTER CLEAR IS DONE AND REG 13 IS READ AND CHECKED FOR INITIALIZED
4822                    *   STATE, WITH LULOOP SET TO 1. THEN, LULOOP IS CLEARED AND REG 13 IS READ
4823                    *   AND CHECKED FOR THE PROPER STATE, WITH LULOOP CLEARED.
4824                    *   REG 13 IS THEN LOADED WITH 0'S, AND READ AND CHECKED FOR THE
4825                    *   INITIALIZED STATE.
4826                    *   REG 17 IS THEN READ AND CHECKED FOR INITIALIZED STATE.
4827                    * - RUN IS SET IN BSEL1, AND REG 13 IS READ AND CHECKED FOR RING SET.
4828                    * - POLL IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGQ SET.
4829                    * - BPOLL IS SET IN REG 12, ONLY TO LIGHT THE LED FOR THIS SIGNAL.
4830                    * - DTR IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR DTR AND MODR SET.
4831                    * - SELFR IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGR SET.
4832                    * - HDX IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR HDX SET.
4833                    * - MAINT1 IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR TEST MODE SET.
4834                    * - SELSBY IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR STBY SET.
4835                    * - A MASTER CLEAR IS DONE, 2 TSOM'S ARE LOADED ITNO THE TX SILO, THE LINE
4836                    *   UNIT IS CLOCKED UNTIL THE TRANSMITTER IS ACTIVE, AND REG 13 IS READ AND
4837                    *   CHECKED FOR RTS, CS, CARR SET.
4838                    ;*************************************************************************
4839
4840
4841
4842
4843
4844                    ;*************************************************************************
```

```
4845                                    TEST 24 - DATA TEST - BIT MODE, NO ERR DET
4846                                *
4847                                * A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH ERROR DETECTION
4848                                * INHIBITED. THE MESSAGE CONSISTS OF 2 FLAGS, PAT A REPEATED 3 TIMES,
4849                                * AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
4850                                * THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
4851                                * SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
4852                                * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
4853                                * TEST WILL NOT BE RUN.
4854                                *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
4855                                *               375,373,367,357,337,277,177
4856                                * 8-BIT CHARACTERS ARE USED.
4857                                ;*******************************************************************************
4858
4859
4860
4861
4862
4863                                ;*******************************************************************************
4864                                    TEST 25 - DATA TEST - CHAR MODE, NO ERR DET
4865                                *
4866                                * A MESSAGE IS INITIATED IN CHAR MODE, WITH ERROR DETECTION
4867                                * INHIBITED. THE MESSAGE CONSISTS OF 2 SYNCHS, PAT A REPEATED 3 TIMES,
4868                                * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
4869                                * THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
4870                                * SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
4871                                * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
4872                                * TEST WILL NOT BE RUN.
4873                                *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
4874                                *               375,373,367,357,337,277,177
4875                                * 8-BIT CHARACTERS ARE USED.
4876                                ;*******************************************************************************
4877
4878
4879
4880
4881
4882                                ;*******************************************************************************
4883                                    TEST 26 - DATA TEST - BIT MODE, CRC-CCITT-1
4884                                *
4885                                * A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-1 ERROR
4886                                * DETECTION. THE MESSAGE CONSISTS OF 2 FLAGS, PAT A REPEATED 3 TIMES,
4887                                * AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
4888                                * THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
4889                                * SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
4890                                * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
4891                                * TEST WILL NOT BE RUN.
4892                                *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
4893                                *               375,373,367,357,337,277,177
4894                                * 8-BIT CHARACTERS ARE USED.
4895                                ;*******************************************************************************
4896
4897
4898
4899
4900
```

```
4901                    ;*****************************************************************************
4902                        TEST 27 - DATA TEST - BIT MODE, CRC-CCITT-0
4903                    *
4904                    * A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-0 ERROR
4905                    * DETECTION. THE MESSAGE CONSISTS OF 2 FLAGS, PAT A REPEATED 3 TIMES,
4906                    * AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
4907                    * THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
4908                    * SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
4909                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
4910                    * TEST WILL NOT BE RUN.
4911                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
4912                    *               375,373,367,357,337,277,177
4913                    * 8-BIT CHARACTERS ARE USED.
4914                    ;*****************************************************************************
4915
4916
4917
4918
4919
4920                    ;*****************************************************************************
4921                        TEST 28 - DATA TEST - CHAR MODE, CRC-16
4922                    *
4923                    * A MESSAGE IS INITIATED IN CHAR MODE, WITH CRC-16 ERROR
4924                    * DETECTION. THE MESSAGE CONSISTS OF 2 SYNCHS, PAT A REPEATED 3 TIMES,
4925                    * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
4926                    * THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
4927                    * SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
4928                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
4929                    * TEST WILL NOT BE RUN.
4930                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
4931                    *               375,373,367,357,337,277,177
4932                    * 8-BIT CHARACTERS ARE USED.
4933                    ;*****************************************************************************
4934
4935
4936
4937
4938
4939                    ;*****************************************************************************
4940                        TEST 29 - DATA TEST - CHAR MODE, ODD VRC
4941                    *
4942                    * A MESSAGE IS INITIATED IN CHAR MODE, WITH ODD VRC ERROR DETECTION
4943                    * SELECTED. THE MESSAGE CONSISTS OF 2 SYNCHS, PAT A REPEATED 3 TIMES,
4944                    * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
4945                    * THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
4946                    * SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
4947                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
4948                    * TEST WILL NOT BE RUN.
4949                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
4950                    *               375,373,367,357,337,277,177
4951                    * 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
4952                    ;*****************************************************************************
4953
4954
4955
4956
```

```
4957
4958                    ;*************************************************************************
4959                        TEST 30 - DATA TEST - CHAR MODE, EVEN VRC
4960                    *
4961                    * A MESSAGE IS INITIATED IN CHAR MODE, WITH EVEN VRC ERROR DETECTION
4962                    * SELECTED. THE MESSAGE CONSISTS OF 2 SYNCHS, PAT A REPEATED 3 TIMES,
4963                    * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
4964                    * THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
4965                    * SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
4966                    * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
4967                    * TEST WILL NOT BE RUN.
4968                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
4969                    *               375,373,367,357,337,277,177
4970                    * 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
4971                    ;*************************************************************************
4972
4973
4974
4975
4976
4977                    ;*************************************************************************
4978                        TEST 31 - CONTIGUOUS ONES IN SEC. STA. ADRS. MODE, BIT MODE
4979                    *
4980                    * IN THIS TEST, A MESSAGE CONSISTING OF 5 ONES CHARS (377 OCT)
4981                    * IS SENT IN SECONDARY STATION ADDRESS MODE, WITH THE STATION ADRS
4982                    * FOR THIS LINE = 377. THE PROGRAM CHECKS FOR CORRECT RECEPTION OF
4983                    * THE FIRST CHARACTER (STATION ADDRESS) AND THE REMAINING 4
4984                    * ONES CHARACTERS (DATA). THIS TEST EXERCISES THE SECONDARY STATION
4985                    * ADDRESS LOGIC, AND CHECKS THAT THE SEC. STA. ADRS. CAN BE BIT-STUFFED
4986                    * AND TRANSMITTED AND RECEIVED CORRECTLY.
4987                    ;*************************************************************************
4988
4989
4990
4991
4992
4993                    ;*************************************************************************
4994                        TEST 32 - DDCMP MESSAGE TEST - CHAR MODE
4995                    *
4996                    * IN THIS TEST, THREE USYRT MESSAGES ARE SENT TO SIMULATE A DDCMP HEADER,
4997                    * DDCMP DATA MESSAGE, AND THE START OF A NEW DDCMP HEADER.
4998                    * FIRST, THE DATA IN PATTERN A IS TRANSMITTED AND RECEIVED
4999                    * AND THEN CRC (CRC-16) IS SENT, FOLLOWED BY THE DATA IN PATTERN A
5000                    * AGAIN AND THE CRC ON THAT DATA, AND FINALLY THE DATA IN 'MSG1'' IS
5001                    * SENT WITH ITS CORRESPONDING CRC.
5002                    *   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
5003                    *               375,373,367,357,337,277,177
5004                    *   MSG1 = SYNCH,SYNCH,SYNCH,SYNCH,000,125,252,377,SYNCH,SYNCH
5005                    ;*************************************************************************
5006
5007
5008
5009
5010
5011                    8.1 DATA PATTERNS USED
5012
```

```
                        ***** DATA PATTERN A *****
                        PATA:
                                .BYTE   125
                                .BYTE   252
                                .BYTE   000
                                .BYTE   377
                                .BYTE   001
                                .BYTE   002
                                .BYTE   004
                                .BYTE   010
                                .BYTE   020
                                .BYTE   040
                                .BYTE   100
                                .BYTE   200
                                .BYTE   376
                                .BYTE   375
                                .BYTE   373
                                .BYTE   367
                                .BYTE   357
                                .BYTE   337
                                .BYTE   277
                                .BYTE   177

                        ***** DATA PATTERN Q *****
                        PATQ:   .BYTE   000
                                .BYTE   120
                                .BYTE   125
                                .BYTE   137
                                .BYTE   040
                                .BYTE   052
                                .BYTE   057
                                .BYTE   177

                        ***** DATA PATTERN R *****
                        PATR:   .BYTE   000
                                .BYTE   100
                                .BYTE   120
                                .BYTE   124
                                .BYTE   164
                                .BYTE   172
                                .BYTE   176
                                .BYTE   177
                                .BYTE   000
                                .BYTE   100
                                .BYTE   120
                                .BYTE   124
                                .BYTE   164
                                .BYTE   172
                                .BYTE   176

                        ***** DATA PATTERN S *****
                        PATS:   .BYTE   000
                                .BYTE   017
                                .BYTE   036
```

```
5069                                    .BYTE   074
5070                                    .BYTE   170
5071                                    .BYTE   360
5072                                    .BYTE   037
5073                                    .BYTE   076
5074                                    .BYTE   174
5075                                    .BYTE   370
5076                                    .BYTE   077
5077                                    .BYTE   176
5078                                    .BYTE   374
5079                                    .BYTE   177
5080                                    .BYTE   376
5081                                    .BYTE   377
5082
5083                                ***** DATA PATTERN T *****
5084                                PATT:   .BYTE   000
5085                                        .BYTE   125
5086                                        .BYTE   252
5087                                        .BYTE   176
5088                                        .BYTE   177
5089
5090
5091
```

5093

9.0 ERROR INFORMATION

9.1 ERROR REPORTING

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

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

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

FAILING REG:  INBUS/OUTBUS REG 12

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

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

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

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

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

FAILING REG:  INBUS/OUTBUS REG 12
```

```
5148
5149
5150
5151
5152                                    @
5153
5154
5155
5156
5157
```

```
                        .TITLE CZDMSA M8203 STATIC TESTS #2
002000                  .=2000



                        .MCALL  SVC
002000                  SVC                             ; INITIALIZE SUPERVISOR MACROS


002000                  BGNMOD  LU2MOD

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

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

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

       002000                         POINTER BGNSW,BGNSFT,BGNAU,BGNDU

                              ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                              ;       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  CZDMS,A,0,45.,0
  (4)  002000                                                             L$NAME::
  (4)  002000        103                                                            .ASCII /C/
  (4)  002001        132                                                            .ASCII /Z/
  (4)  002002        104                                                            .ASCII /D/
  (4)  002003        115                                                            .ASCII /M/
  (4)  002004        123                                                            .ASCII /S/
  (6)  002005        000                                                            .BYTE   0
  (6)  002006        000                                                            .BYTE   0
  (5)  002007        000                                                            .BYTE   0
  (5)  002010                                                             L$REV::
  (4)  002010        101                                                            .ASCII  /A/
  (5)  002011                                                             L$DEPO::
  (4)  002011        060                                                            .ASCII  /0/
  (5)  002012                                                             L$UNIT::
  (4)  002012     000000                                                            .WORD   0
  (5)  002014                                                             L$TIML::
  (4)  002014     000055                                                            .WORD   45.
  (5)  002016                                                             L$HPCP::
  (4)  002016     036336                                                            .WORD   L$HARD
  (5)  002020                                                             L$SPCP::
  (4)  002020     037300                                                            .WORD   L$SOFT
  (5)  002022                                                             L$HPTP::
  (4)  002022     002226                                                            .WORD   L$HW
  (5)  002024                                                             L$SPTP::
  (4)  002024     002256                                                            .WORD   L$SW
  (5)  002026                                                             L$LADP::
  (4)  002026     037750                                                            .WORD   L$LAST
  (5)  002030                                                             L$STA::
  (4)  002030     000000                                                            .WORD   0
  (5)  002032                                                             L$CO::
  (4)  002032     000000                                                            .WORD   0
  (5)  002034                                                             L$DTYP::
  (4)  002034     000000                                                            .WORD   0
  (5)  002036                                                             L$APT::
  (4)  002036     000000                                                            .WORD   0
  (5)  002040                                                             L$DTP::
  (4)  002040     002124                                                            .WORD   L$DISPATCH
  (5)  002042                                                             L$EXP1::
  (4)  002042     000000                                                            .WORD   0
  (5)  002044                                                             L$EXP2::
  (4)  002044     000000                                                            .WORD   0
```

```
  (5)  002046                                    L$EXP3::
  (4)  002046  000000                                    .WORD   0
  (5)  002050                                    L$MREV::
  (4)  002050     003                                    .BYTE   C$REVISION
  (3)  002051     000                                    .BYTE   C$EDIT
  (5)  002052                                    L$EF::
  (4)  002052  000000                                    .WORD   0
  (5)  002054  000000                                    .WORD   0
  (5)  002056                                    L$SPC::
  (4)  002056  000000                                    .WORD   0
  (5)  002060                                    L$DEVP::
  (4)  002060  003162                                    .WORD   L$DVTYP
  (5)  002062                                    L$REPP::
  (4)  002062  000000                                    .WORD   0
  (5)  002064                                    L$EXP4::
  (4)  002064  000000                                    .WORD   0
  (5)  002066                                    L$EXP5::
  (4)  002066  000000                                    .WORD   0
  (5)  002070                                    L$AUT::
  (4)  002070  023144                                    .WORD   L$AU
  (5)  002072                                    L$DUT::
  (4)  002072  023062                                    .WORD   L$DU
  (5)  002074                                    L$LUN::
  (4)  002074  000000                                    .WORD   0
  (5)  002076                                    L$DESP::
  (4)  002076  003170                                    .WORD   L$DESC
  (5)  002100                                    L$LOAD::
  (4)  002100  104035                                    EMT     E$LOAD
  (5)  002102                                    L$ETP::
  (4)  002102  000000                                    .WORD   0
  (5)  002104                                    L$ICP::
  (4)  002104  022026                                    .WORD   L$INIT
  (5)  002106                                    L$CCP::
  (4)  002106  023060                                    .WORD   L$CLEAN
  (5)  002110                                    L$ACP::
  (4)  002110  023000                                    .WORD   L$AUTO
  (5)  002112                                    L$PRT::
  (4)  002112  022020                                    .WORD   L$PROT
  (5)  002114                                    L$TEST::
  (4)  002114  000000                                    .WORD   0
  (5)  002116                                    L$DLY::
  (4)  002116  000000                                    .WORD   0
  (5)  002120                                    L$HIME::
  (4)  002120  000000                                    .WORD   0
5217
5219                 ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5220                 ;      CHANGE THE ''HEADER'' TO CONTAIN THE PROPER ARGUMENTS.
5221                 ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5223
5224                 .EVEN
5225
5226
5227
5228
5229
5230
```

```
                        .SBTTL  DISPATCH TABLE

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

                              DISPATCH 32
 (4)  002122  000040                                                    .WORD   32
 (3)  002124                                                          L$DISPATCH::
 (6)  002124  023146                                                    .WORD   T1
 (6)  002126  023276                                                    .WORD   T2
 (6)  002130  023652                                                    .WORD   T3
 (6)  002132  024210                                                    .WORD   T4
 (6)  002134  024272                                                    .WORD   T5
 (6)  002136  024362                                                    .WORD   T6
 (6)  002140  024556                                                    .WORD   T7
 (6)  002142  025326                                                    .WORD   T8
 (6)  002144  025566                                                    .WORD   T9
 (6)  002146  025772                                                    .WORD   T10
 (6)  002150  026126                                                    .WORD   T11
 (6)  002152  026600                                                    .WORD   T12
 (6)  002154  026714                                                    .WORD   T13
 (6)  002156  027252                                                    .WORD   T14
 (6)  002160  027662                                                    .WORD   T15
 (6)  002162  030334                                                    .WORD   T16
 (6)  002164  031036                                                    .WORD   T17
 (6)  002166  031320                                                    .WORD   T18
 (6)  002170  031572                                                    .WORD   T19
 (6)  002172  031754                                                    .WORD   T20
 (6)  002174  032136                                                    .WORD   T21
 (6)  002176  032234                                                    .WORD   T22
 (6)  002200  032444                                                    .WORD   T23
 (6)  002202  033736                                                    .WORD   T24
 (6)  002204  034126                                                    .WORD   T25
 (6)  002206  034316                                                    .WORD   T26
 (6)  002210  034506                                                    .WORD   T27
 (6)  002212  034676                                                    .WORD   T28
 (6)  002214  035066                                                    .WORD   T29
 (6)  002216  035252                                                    .WORD   T30
 (6)  002220  035436                                                    .WORD   T31
 (6)  002222  035550                                                    .WORD   T32

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

```
                        .SBTTL  DEFAULT HARDWARE P-TABLE

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

002224                          BGNHW   DFPTBL
(3)  002224  000013                                                     .WORD   L10000-L$HW/2
(3)  002226                                                             L$HW::
(3)  002226                                                             DFPTBL::
002226  000007                  .WORD       7       ;MICROPROCESSOR TYPE = M8207
002230  160170                  .WORD  160170       ;DMC11 OR KMC11 CSR UNIBUS ADDRESS
002232  000300                  .WORD     300       ;DMC11 OR KMC11 INTERRUPT VECTOR
002234  005000                  .WORD    5000       ;DMC11 OR KMC11 INTERRUPT PRIORITY LEVEL = 5
002236  000003                  .WORD       3       ;LINE UNIT = M8203
002240  000056                  .WORD     056       ;SWITCH PACK #1 (REG 11)
002242  000000                  .WORD     000       ;SWITCH PACK #2 (REG 15)
002244  000000                  .WORD     000       ;SWITCH PACK #3 (REG 16)
002246  000000                  .WORD       0       ;H3254&H3255 USED
002250  000004                  .WORD       4       ;BAUD RATE = 56 K
002252  000001                  .WORD       1       ;RUN SWITCH ON MICROPROCESSOR IS ON

002254                          ENDHW
(3)  002254                                                             L10000:
```

```
                        .SBTTL  SOFTWARE P-TABLE

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

                                BGNSW   SFPTBL
(3)  002254  000004                                                     .WORD   L10001-L$SW/2
(3)  002256                                                         L$SW::
(3)  002256                                                         SFPTBL::

002256  000000          MIFLAG: .WORD   0       ; =1 IF MAN. INTERVENTION DESIRED, =0 IF NOT
002260  000000          PRNFLG: .WORD   0       ; =1 IF SW PACK AND AX3-15 PRINTOUT ALLOWED ALWAYS
002262  000000          SWIFLG: .WORD   0       ; =1 IF SWITCH PACK VERIFICATION TEST SHOULD BE RUN
002264  000000          TCOUNT: .WORD   0       ;INITIAL MSG TIME-OUT VALUE (0=LONGEST TIME-OUT)

                                ENDSW
(3)  002266                                                         L10001:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

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

002266                                  EQUALS
(1)                             ;
(1)                             ; BIT DIFINITIONS
(1)                             ;
(1)             100000          BIT15== 100000
(1)             040000          BIT14== 40000
(1)             020000          BIT13== 20000
(1)             010000          BIT12== 10000
(1)             004000          BIT11== 4000
(1)             002000          BIT10== 2000
(1)             001000          BIT09== 1000
(1)             000400          BIT08== 400
(1)             000200          BIT07== 200
(1)             000100          BIT06== 100
(1)             000040          BIT05== 40
(1)             000020          BIT04== 20
(1)             000010          BIT03== 10
(1)             000004          BIT02== 4
(1)             000002          BIT01== 2
(1)             000001          BIT00== 1
(1)                             ;
(1)             001000          BIT9==  BIT09
(1)             000400          BIT8==  BIT08
(1)             000200          BIT7==  BIT07
(1)             000100          BIT6==  BIT06
(1)             000040          BIT5==  BIT05
(1)             000020          BIT4==  BIT04
(1)             000010          BIT3==  BIT03
(1)             000004          BIT2==  BIT02
(1)             000002          BIT1==  BIT01
(1)             000001          BIT0==  BIT00
(1)                             ;
(1)                             ; EVENT FLAG DEFINITIONS
(1)                             ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
(1)                             ;
(1)             000040          EF.START==      32.                             ; START COMMAND WAS ISSUED
(1)             000037          EF.RESTART==    31.                             ; RESTART COMMAND WAS ISSUED
```

```
(1)   000036                EF.CONTINUE==   30.                ; CONTINUE COMMAND WAS ISSUED
(1)   000035                EF.NEW==        29.                ; A NEW PASS HAS BEEN STARTED
(1)   000034                EF.PWR==        28.                ; A POWER-FAIL/POWER-UP OCCURRED
(1)                         ;
(1)                         ;
(1)                         ; PRIORITY LEVEL DEFINITIONS
(1)                         ;
(1)   000340                PRI07== 340
(1)   000300                PRI06== 300
(1)   000240                PRI05== 240
(1)   000200                PRI04== 200
(1)   000140                PRI03== 140
(1)   000100                PRI02== 100
(1)   000040                PRI01== 40
(1)   000000                PRI00== 0
(1)                         ;
(1)                         ;OPERATOR FLAG BITS
(1)                         ;
(1)   000004                EVL==       4
(1)   000010                LOT==      10
(1)   000020                ADR==      20
(1)   000040                IDU==      40
(1)   000100                ISR==     100
(1)   000200                UAM==     200
(1)   000400                BOE==     400
(1)   001000                PNT==    1000
(1)   002000                PRI==    2000
(1)   004000                IXE==    4000
(1)   010000                IBE==   10000
(1)   020000                IER==   20000
(1)   040000                LOE==   40000
(1)   100000                HOE==  100000
5324
5325
5326
5327
5328                        ;;*************************************************************************
5329                        ;* PROGRAM EVENT FLAG DEFINITIONS
5330                        ;;*************************************************************************
5331
5332
5333
5334
5335
5336                        ;;*************************************************************************
5337                        ;* MAINTENANCE REGISTER - BSEL1
5338                        ;;*************************************************************************
5339   000200               RUN     = BIT7
5340   000100               MCLR    = BIT6
5341   000020               STEPLU  = BIT4
5342   000010               LULOOP  = BIT3
5343   000004               ROMO    = BIT2
5344   000002               ROMI    = BIT1
5345   000001               STEPMP  = BIT0
5346
5347                        ;;*************************************************************************
```

```
                        ;* OBUS REG 10 - TRANSMITTER BUFFER
                        ;;*****************************************************************************
000200                  TX7     = BIT7
000100                  TX6     = BIT6
000040                  TX5     = BIT5
000020                  TX4     = BIT4
000010                  TX3     = BIT3
000004                  TX2     = BIT2
000002                  TX1     = BIT1
000001                  TX0     = BIT0

                        ;;*****************************************************************************
                        ;* OBUS REG 11
                        ;;*****************************************************************************
000200                  OC      = BIT7
000010                  GOAH    = BIT3
000004                  ABORT   = BIT2
000002                  EOM     = BIT1
000001                  SOM     = BIT0

                        ;;*****************************************************************************
                        ;* OBUS REG 12
                        ;;*****************************************************************************
000200                  IC      = BIT7
000100                  BPOLL   = BIT6
000040                  LULP    = BIT5

                        ;;*****************************************************************************
                        ;* OBUS REG 13
                        ;;*****************************************************************************
000200                  POLL    = BIT7
000100                  DTR     = BIT6
000040                  SELFR   = BIT5
000020                  HDX     = BIT4
000010                  MAINT1  = BIT3
000004                  MAINT2  = BIT2
000002                  SELSBY  = BIT1

                        ;;*****************************************************************************
                        ;* OBUS REG 14
                        ;;*****************************************************************************
000100                  TXEN    = BIT6
000040                  DISSI   = BIT5
000020                  RDAX    = BIT4
000010                  WAX     = BIT3
000004                  ENAX    = BIT2
000002                  AX2     = BIT1
000001                  AX1     = BIT0

                        ;;*****************************************************************************
                        ;* OBUS REG 17
                        ;;*****************************************************************************
000200                  CRC2    = BIT7
000100                  CRC1    = BIT6
000040                  IDLE    = BIT5
000020                  SECA    = BIT4
```

```
                        STRIP   = BIT3
                        RDAIL   = BIT2
                        IERR    = BIT1
                        DDCMP   = BIT0

                        ;;*****************************************************************************************
                        ;* IBUS REG 10 - RECEIVER BUFFER
                        ;;*****************************************************************************************
                        RX7     = BIT7
                        RX6     = BIT6
                        RX5     = BIT5
                        RX4     = BIT4
                        RX3     = BIT3
                        RX2     = BIT2
                        RX1     = BIT1
                        RX0     = BIT0

                        ;;*****************************************************************************************
                        ;* IBUS REG 11
                        ;;*****************************************************************************************
                        OC      = BIT7
                        OACT    = BIT6
                        SW3     = BIT5
                        ORDY    = BIT4
                        SW2     = BIT3
                        SW1     = BIT2
                        SW0     = BIT1
                        UNRR    = BIT0

                        ;;*****************************************************************************************
                        ;* IBUS REG 12
                        ;;*****************************************************************************************
                        IC      = BIT7
                        IACT    = BIT6
                        LULP    = BIT5
                        IRDY    = BIT4
                        OVRR    = BIT3
                        RAB     = BIT2
                        EBLK    = BIT1
                        BCC     = BIT0

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

                        ;;*****************************************************************************************
                        ;* IBUS REG 14
                        ;;*****************************************************************************************
```

```
                        READY   = BIT7
                        TXEN    = BIT6
                        DISSI   = BIT5
                        RDAX    = BIT4
                        WAX     = BIT3
                        ENAX    = BIT2
                        AX2     = BIT1
                        AX1     = BIT0

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

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

                        ;;*************************************************************************************
                        ;* AX0-16 - USYRT REG 1 (READ ONLY)
                        ;;*************************************************************************************
                        RERR    = BIT7
                        ASBC2   = BIT6
                        ASBC1   = BIT5
                        ASBC0   = BIT4
                        ROR     = BIT3
                        RABT    = BIT2
                        REOM    = BIT1
                        RSOM    = BIT0

                        ;;*************************************************************************************
                        ;* AX1-15 - USYRT REG 2
                        ;;*************************************************************************************
                        TX7     = BIT7
                        TX6     = BIT6
                        TX5     = BIT5
                        TX4     = BIT4
                        TX3     = BIT3
                        TX2     = BIT2
                        TX1     = BIT1
                        TX0     = BIT0
```

```
                                ;;*****************************************************************************
                                ;* AX1-16 - USYRT REG 3
                                ;;*****************************************************************************
000200                          TERR    = BIT7
000010                          TXGA    = BIT3
000004                          TXAB    = BIT2
000002                          TEOM    = BIT1
000001                          TSOM    = BIT0

                                ;;*****************************************************************************
                                ;* AX2-15 - USYRT REG 4
                                ;;*****************************************************************************
000200                          SYN7    = BIT7
000100                          SYN6    = BIT6
000040                          SYN5    = BIT5
000020                          SYN4    = BIT4
000010                          SYN3    = BIT3
000004                          SYN2    = BIT2
000002                          SYN1    = BIT1
000001                          SYN0    = BIT0
000226                          SYNCH   = 226

                                ;;*****************************************************************************
                                ;* AX2-16 - USYRT REG 5
                                ;;*****************************************************************************
000200                          APA     = BIT7
000100                          DDC     = BIT6
000040                          STR     = BIT5
000020                          SEC     = BIT4
000010                          IDL     = BIT3
000004                          CRCTY2  = BIT2
000002                          CRCTY1  = BIT1
000001                          CRCTY0  = BIT0

                                ;;*****************************************************************************
                                ;* AX3-15 - USYRT REG 6
                                ;;*****************************************************************************
000200                          I422    = BIT7
000100                          XYZ     = BIT6
000020                          V35     = BIT4
000010                          INTGRL  = BIT3
000002                          OP      = BIT1
000001                          TEST    = BIT0
000332                          AX315U  = I422!XYZ!V35!INTGRL!OP

                                ;;*****************************************************************************
                                ;* AX3-16 - USYRT REG 7
                                ;;*****************************************************************************
000200                          TXLEN2  = BIT7
000100                          TXLEN1  = BIT6
000040                          TXLEN0  = BIT5
000004                          RXLEN2  = BIT2
000002                          RXLEN1  = BIT1
000001                          RXLEN0  = BIT0
```

```
5572
5573
5574
5575
5576                    ;*****************************************************************************
5577                    ;* TX CONTROL BITS DEFINED ON WORD BASIS
5578                    ;*****************************************************************************
5579    004000          TXGOA   = BIT11
5580    002000          TXABT   = BIT10
5581    001000          TXEOM   = BIT9
5582    000400          TXSOM   = BIT8
5583
5584
5585
5586
5587
5588                    ;*****************************************************************************
5589                    ;* RCV CONTROL BITS DEFINED ON WORD BASIS
5590                    ;*****************************************************************************
5591    004000          RXOVR   = BIT11
5592    002000          RXABT   = BIT10
5593    001000          RXEBL   = BIT9
5594    000400          RXBCC   = BIT8
5595
5596
5597
5598
5599                    ;*****************************************************************************
5600                    ;* ADDRESS EQUATES FOR REGISTER STORAGE TABLE (LUREG:)
5601                    ;*****************************************************************************
5602    002266          LUR10   = LUREG+0       ;LINE UNIT IBUS REG 10
5603    002270          LUR11   = LUREG+2       ;LINE UNIT IBUS REG 11
5604    002272          LUR12   = LUREG+4       ;LINE UNIT IBUS REG 12
5605    002274          LUR13   = LUREG+6       ;LINE UNIT IBUS REG 13
5606    002276          LUR14   = LUREG+10      ;LINE UNIT IBUS REG 14
5607    002300          LUR15   = LUREG+12      ;LINE UNIT IBUS REG 15
5608    002302          LUR16   = LUREG+14      ;LINE UNIT IBUS REG 16
5609    002304          LUR17   = LUREG+16      ;LINE UNIT IBUS REG 17
5610    002306          AX0.15  = LUREG+20      ;USYRT REG 0
5611    002310          AX0.16  = LUREG+22      ;USYRT REG 1
5612    002312          AX1.15  = LUREG+24      ;USYRT REG 2
5613    002314          AX1.16  = LUREG+26      ;USYRT REG 3
5614    002316          AX2.15  = LUREG+30      ;USYRT REG 4
5615    002320          AX2.16  = LUREG+32      ;USYRT REG 5
5616    002322          AX3.15  = LUREG+34      ;USYRT REG 6
5617    002324          AX3.16  = LUREG+36      ;USYRT REG 7
5618
5619
5620
5621
5622
5623    100000          CHPCHK  = BIT15
5624
5625    100000          BCCCHK  = BIT15
5626    100000          CRCCHK  = BIT15
5627
```

```
5628            100000                  TCCHEK  = BIT15
5629
5630
5631
5632
5633
5634                                    ;*****************************************************************************
5635                                    ;* MICROINSTRUCTION DEFINITIONS
5636                                    ;*****************************************************************************
5637            021000                  MVIOX   = 021000                ;MOVE IBUS TO OBUS*
5638            122000                  MVIXO   = 122000                ;MOVE IBUS* TO OBUS
5639
5640
5641
5642                                    ;***** ERROR1 BIT FLAG DEFINITIONS *****
5643            000001                  RRDYTO  = BIT0
5644            000002                  WRDYTO  = BIT1
5645
5646
5647
5648
5649
```

```
                        .SBTTL  GLOBAL DATA SECTION

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

                        ;;**************************************************************************
                        ;* STORAGE FOR DEVICE REGISTERS
                        ;;**************************************************************************
002266  000020          LUREG:  .BLKW   16.

                        ;**************************************************************************
                        ;* MISCELLANEOUS STORAGE
                        ;**************************************************************************
002326  000000          SCRACH: .WORD   0               ;GEN'L PURPOSE SCRATCH WORD
002330  000000          LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
002332  000000          PSTACK: .WORD   0               ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
002334  000000          PRIOR:  .WORD   0               ;CPU PRIORITY FOR PRINTOUT
002336  000000          SUBRPC: .WORD   0               ;PC OF SUBR CALL FOR ERROR REPORTS
002340  000000          INTFLG: .WORD   0               ;INTERRUPT RECEIVED FLAGS
                                                        ;  BIT 0 FOR TX, BIT 1 FOR RCV
002342  000000          ERRFLG: .WORD   0               ;SUBROUTINE ERROR FLAG
002344  000000          TIMFLG: .WORD   0               ;EVENT TIME-OUT FLAG
002346  000000          RETADR: .WORD   0               ;SUBR ERROR RETURN ADDRESS
002350  000000          REDBYT: .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM LU REG
002352  000000          WRIBYT: .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
002354  000000          RAX15:  .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM REG 15
002356  000000          RAX16:  .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM REG 16
002360  000000          WAX15:  .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 15
002362  000000          WAX16:  .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 16
002364  000000          REGNUM: .WORD   0               ;NUMBER (10-17) OF LINE UNIT REG BEING TESTED
002366  000000          AXNUM:  .WORD   0               ;NUMBER (0-7) OF EXTENDED REG BYTE BEING TESTED
002370  000000          GOODAT: .WORD   0               ;STORAGE FOR EXPECTED DATA
002372  000000          BADDAT: .WORD   0               ;STORAGE FOR ACTUAL DATA
002374  000000          LOADAT: .WORD   0               ;CONTAINS TEST DATA LOADED INTO REG
002376  000000          FRSTIM: .WORD   0               ;FLAG=0 IF PROGRAM JUST LOADED
002400  000000          FRSPAS: .WORD   0               ;FLAG=0 IF FIRST PASS AFTER LOAD
002402  000000          STARES: .WORD   0               ;FLAG=0 IF FIRST TIME THRU AFTER STA OR RES
002404  000000          SAVE4:  .WORD   0               ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
002406  000000          SAVE6:  .WORD   0               ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
002410  000000          ERROR1: .WORD   0               ;SUBR ERROR BIT FLAGS (DEF'D IN GLOBAL EQUATES)
002412  000000          TXWORD: .WORD   0               ;BITS 0-11 CONTAIN DATA TO LOAD INTO TX SILO
002414  000000          RXWORD: .WORD   0               ;BITS 0-11 CONTAIN DATA READ FROM RCV SILO
002416  000000          DISILO: .WORD   0               ;CONTAINS CURRENT STATE OF DISSI IN BIT 5
002420  000000          CHPTYP: .WORD   0               ;USYRT CHIP TYPE, =0 FOR SIG, ELSE =1
002422  000000          MODINT: .WORD   0               ;MODEM INTERFACE SELECTION
002424  000000          SAVLEN: .WORD   0               ;SAVED TX AND RCV CHAR LENGTHS
002426  000000          DEVMAP: .WORD   0               ;BIT MAP OF ACTIVE DEVICES
002430  000000          DEVPTR: .WORD   0               ;DEVICE MAP BIT POINTER
002432  000000          UNIT:   .WORD   0               ;CONTAINS UNIT NUMBER (1 TO N)
002434  000000          TSTNUM: .WORD   0               ;CONTAINS TEST NUMBER FOR SOME TESTS

                        ;;*********  CURRENT DEVICE PARAMETERS  *************
002436  160170          MPCSR:  .WORD   160170          ;POINTER TO MICROPROCESSOR CSR'S
002440  160171          BSEL1:  .WORD   160171          ;POINTER TO BSEL1
```

```
 5707  002442
 5708  002442  160174          BSEL4:
 5709  002442  160174          SEL4:   .WORD   160174          ;POINTER TO SEL4
 5710  002444  160176          SEL6:   .WORD   160176          ;POINTER TO SEL6
 5711  002446  000300          MPIVEC: .WORD      300          ;MICROPROCESSOR INPUT INTERRUPT VECTOR
 5712  002450  000304          MPOVEC: .WORD      304          ;MICROPROCESSOR OUTPUT INTERRUPT VECTOR
 5713  002452  000240          MPRIOR: .WORD      240          ;MICROPROCESSOR DEVICE PRIORITY
 5714  002454  000000          LUSWI1: .WORD        0          ;LINE UNIT SWITCH PACK #1
 5715  002456  000000          LUSWI2: .WORD        0          ;LINE UNIT SWITCH PACK #2
 5716  002460  000000          LUSWI3: .WORD        0          ;LINE UNIT SWITCH PACK #3
 5717  002462  000000          TSTCON: .WORD        0          ;TEST CONNECTOR INDICATOR
 5718  002464  000004          BDRATE: .WORD        4          ;BAUD RATE
 5719
 5720  002466       000        ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
 5721  002466       000        REDDAT: .BYTE   0
 5722  002467       000                .BYTE   0
 5723  002470       000                .BYTE   0
 5724  002471       000                .BYTE   0
 5725  002472       000                .BYTE   0
 5726  002473       000                .BYTE   0
 5727  002474       000                .BYTE   0
 5728  002475       000                .BYTE   0
 5729
 5730  002476  000000          ;;****  GEN'L PURPOSE SCRATCH STORAGE  *****
 5731  002476  000000          REG0:   .WORD   0
 5732  002500  000000          REG1:   .WORD   0
 5733  002502  000000          REG2:   .WORD   0
 5734  002504  000000          REG3:   .WORD   0
 5735  002506  000000          REG4:   .WORD   0
 5736  002510  000000          REG5:   .WORD   0
 5737  002512  000000          REG6:   .WORD   0
 5738  002514  000000          REG7:   .WORD   0
 5739
 5740  002516  000000          ;;****  SCRATCH STORAGE FOR MESSAGE REPORTING   *****
 5741  002516  000000          TMP0:   .WORD   0
 5742  002520  000000          TMP1:   .WORD   0
 5743  002522  000000          TMP2:   .WORD   0
 5744  002524  000000          TMP3:   .WORD   0
 5745  002526  000000          TMP4:   .WORD   0
 5746  002530  000000          TMP5:   .WORD   0
 5747  002532  000000          TMP6:   .WORD   0
 5748  002534  000000          TMP7:   .WORD   0
 5749
 5750  002536                  ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
 5751  002536                  UPBITS:
 5752  002536       000                .BYTE   000             ;MASK FOR REG 10
 5753  002537       056                .BYTE   056             ;MASK FOR REG 11
 5754  002540       000                .BYTE   000             ;MASK FOR REG 12
 5755  002541       257                .BYTE   257             ;MASK FOR REG 13
 5756  002542       100                .BYTE   100             ;MASK FOR REG 14
 5757  002543       377                .BYTE   377             ;MASK FOR REG 15
 5758  002544       377                .BYTE   377             ;MASK FOR REG 16
 5759  002545       306                .BYTE   306             ;MASK FOR REG 17
 5760
 5761  002546       200        R14NRW: .BYTE   200             ;REG 14 NON-R/W BITS
 5762
                                 ;***** MASKS FOR EXTENDED REGISTER NON-READ/WRITE BITS *****
```

```
5763  002547
5764  002547  377          ANBITS:
5765  002550  377                  .BYTE   377             ;MASK FOR AX0-15
5766  002551  000                  .BYTE   377             ;MASK FOR AX0-16
5767  002552  360                  .BYTE   000             ;MASK FOR AX1-15
5768  002553  000                  .BYTE   360             ;MASK FOR AX1-16
5769  002554  000                  .BYTE   000             ;MASK FOR AX2-15
5770  002555  004                  .BYTE   000             ;MASK FOR AX2-16
5771  002556  030                  .BYTE   004             ;MASK FOR AX3-15
5772                               .BYTE   030             ;MASK FOR AX3-16
5773
5774  002557                     ;***** DATA PATTERN A *****
5775  002557  125          PATA:
5776  002560  252                  .BYTE   125
5777  002561  000                  .BYTE   252
5778  002562  377                  .BYTE   000
5779  002563  001                  .BYTE   377
5780  002564  002                  .BYTE   001
5781  002565  004                  .BYTE   002
5782  002566  010                  .BYTE   004
5783  002567  020                  .BYTE   010
5784  002570  040                  .BYTE   020
5785  002571  100                  .BYTE   040
5786  002572  200                  .BYTE   100
5787  002573  376                  .BYTE   200
5788  002574  375                  .BYTE   376
5789  002575  373                  .BYTE   375
5790  002576  367                  .BYTE   373
5791  002577  357                  .BYTE   367
5792  002600  337                  .BYTE   357
5793  002601  277                  .BYTE   337
5794  002602  177                  .BYTE   277
5795                               .BYTE   177
5796
5797  002603                     ;***** DATA PATTERN B *****
5798  002603  000          PATB:
5799  002604  000                  .BYTE   000
5800  002605  040                  .BYTE   000
5801  002606  100                  .BYTE   040
5802  002607  220                  .BYTE   100
5803  002610  000                  .BYTE   220
5804  002611  000                  .BYTE   000
5805  002612  051                  .BYTE   000
5806                               .BYTE   051
5807
5808
5809  002613  000                ;***** DATA PATTERN Q *****
5810  002614  120          PATQ:   .BYTE   000
5811  002615  125                  .BYTE   120
5812  002616  137                  .BYTE   125
5813  002617  040                  .BYTE   137
5814  002620  052                  .BYTE   040
5815  002621  057                  .BYTE   052
5816  002622  177                  .BYTE   057
5817                               .BYTE   177
5818
                             ;***** DATA PATTERN R *****
```

```
 5819  002623  000                PATR:   .BYTE   000
 5820  002624  100                        .BYTE   100
 5821  002625  120                        .BYTE   120
 5822  002626  124                        .BYTE   124
 5823  002627  164                        .BYTE   164
 5824  002630  172                        .BYTE   172
 5825  002631  176                        .BYTE   176
 5826  002632  177                        .BYTE   177
 5827  002633  000                        .BYTE   000
 5828  002634  100                        .BYTE   100
 5829  002635  120                        .BYTE   120
 5830  002636  124                        .BYTE   124
 5831  002637  164                        .BYTE   164
 5832  002640  172                        .BYTE   172
 5833  002641  176                        .BYTE   176
 5834
 5835                             ;***** DATA PATTERN S *****
 5836  002642  000                PATS:   .BYTE   000
 5837  002643  017                        .BYTE   017
 5838  002644  036                        .BYTE   036
 5839  002645  074                        .BYTE   074
 5840  002646  170                        .BYTE   170
 5841  002647  360                        .BYTE   360
 5842  002650  037                        .BYTE   037
 5843  002651  076                        .BYTE   076
 5844  002652  174                        .BYTE   174
 5845  002653  370                        .BYTE   370
 5846  002654  077                        .BYTE   077
 5847  002655  176                        .BYTE   176
 5848  002656  374                        .BYTE   374
 5849  002657  177                        .BYTE   177
 5850  002660  376                        .BYTE   376
 5851  002661  377                        .BYTE   377
 5852
 5853                             ;***** DATA PATTERN T *****
 5854  002662  000                PATT:   .BYTE   000
 5855  002663  125                        .BYTE   125
 5856  002664  252                        .BYTE   252
 5857  002665  176                        .BYTE   176
 5858  002666  177                        .BYTE   177
 5859
 5860  002667                     ENDPAT:
 5861          002670             .EVEN
 5862
 5863
 5864
 5865
 5866
 5867                             ;***  TEST MESSAGES TO BE TRANSMITTED  ***
 5868
 5869  002670  000400             MSG1:   TXSOM
 5870  002672  000400                     TXSOM
 5871  002674  000000                     000
 5872  002676  000125                     125
 5873  002700  000252                     252
 5874  002702  000377                     377
```

```
 5875  002704  000000                        000
 5876  002706  001000                        TXEOM
 5877  002710  001000                        TXEOM
 5878  002712  001000                        TXEOM
 5879  002714  001000                        TXEOM
 5880
 5881  002716  000400                MSG2:   TXSOM
 5882  002720  000400                        TXSOM
 5883  002722  000000                        000
 5884  002724  000377                        377
 5885  002726  001000                        TXEOM
 5886  002730  001000                        TXEOM
 5887
 5888  002732  000001                MSG3:   001
 5889  002734  000001                        001
 5890  002736  000001                        001
 5891  002740  000001                        001
 5892  002742  002000                        TXABT
 5893  002744  000400                        TXSOM
 5894  002746  000400                        TXSOM
 5895  002750  000003                        003
 5896  002752  000003                        003
 5897  002754  000003                        003
 5898  002756  000003                        003
 5899  002760  000003                        003
 5900
 5901
 5902
 5903
 5904
 5905                          ;*** RECEIVED DATA BUFFER (64. WORDS) ***
 5906  002762  000100          RCVBUF: .BLKW   64.
 5907
 5908
 5909
 5910
 5911
 5912
 5913
 5914
 5915
 5916
```

```
                                        .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
                                        ;;*********************************************************************
    003162                                      DEVTYP  <M8203>
(4) 003162                                                                      L$DVTYP::
(3) 003162  034115  030062  000063                                                      .ASCIZ  /M8203/
(2)                                                                                     .EVEN

                                        ;;*********************************************************************
                                        ;;* TITLE OF PROGRAM
                                        ;;*********************************************************************
    003170                                      DESCRIPT        <M8203 STATIC LOGIC TESTS - PART 2 OF 2>
(4) 003170                                                                      L$DESC::
(3) 003170  034115  030062  020063                                                      .ASCIZ  /M8203 STATIC LO
(3) 003176  052123  052101  041511
(3) 003204  046040  043517  041511
(3) 003212  052040  051505  051524
(3) 003220  026440  050040  051101
(3) 003226  020124  020062  043117
(3) 003234  031040     000
(2)         003240                                                                      .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.
                                ;*******************************************************************************
003240                          STPCLK:
003240  152777  000006  177172          BISB    #ROMO!ROMI,@BSEL1       ;SET ROMO, ROMI BITS IN BSEL1
003246  017677  000000  177170          MOV     @(SP),@SEL6     ;PUT INSTRUCTION INTO SEL6
003254  152777  000007  177156          BISB    #ROMO!ROMI!STEPMP,@BSEL1 ;SET ROMO, ROMI, STEPMP IN BSEL1
003262  142777  000007  177150          BICB    #ROMO!ROMI!STEPMP,@BSEL1 ;CLEAR ROMO, ROMI, STEPMP IN BSEL1
003270  062716  000002                  ADD     #2,(SP)         ;FIX UP RETURN PC
003274  000207                          RTS     PC              ;RETURN



                                ;*******************************************************************************
                                ;* MSTCLR - THIS SUBROUTINE ISSUES A MASTER CLEAR AND SETS LULOOP
                                ;*******************************************************************************
003276                          MSTCLR:
003276  010146                          MOV     R1,-(SP)        ;SAVE R1
003300  112777  000100  177132          MOVB    #MCLR,@BSEL1    ;SET MASTER CLEAR BIT
003306  142777  000300  177124          BICB    #RUN!MCLR,@BSEL1 ;CLEAR RUN AND MCLR BITS
003314  012701  000024                  MOV     #20.,R1         ;INITIALIZE STALL COUNTER
003320  000240                  2$:     NOP                     ;STALL IN LOOP FOR SEVERAL MICRO-SEC
003322  005301                          DEC     R1
003324  001375                          BNE     2$
003326  152777  000010  177104          BISB    #LULOOP,@BSEL1  ;SET LU LOOP
003334  012601                          MOV     (SP)+,R1        ;RESTORE R1
003336  005037  002424                  CLR     SAVLEN          ;CLEAR SAVED CHAR LENGTH FROM SETUP
003342  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.
                                ;*******************************************************************************
003344                          READLU:
003344  010146                          MOV     R1,-(SP)        ;SAVE R1
003346  013701  002364                  MOV     REGNUM,R1       ;GET LINE UNIT REG NUMBER
003352  006301                          ASL     R1              ;SHIFT INTO SOURCE BITS 4-7
003354  006301                          ASL     R1
```

```
6009  003356  006301                          ASL     R1
6010  003360  006301                          ASL     R1
6011  003362  052701  000004                  BIS     #4,R1           ;SET DESTINATION = BSEL4
6012  003366  052701  021000                  BIS     #MVIOX,R1       ;SET REST OF MOVE INSTRUCTION
6013  003372  010137  003402                  MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
6014  003376  004737  003240                  JSR     PC,STPCLK       ;EXECUTE MOVE INSTRUCTION
6015  003402  000000              2$:         .WORD   0               ;INSTRUCTION GOES HERE
6016  003404  117737  177032  002350          MOVB    @BSEL4,REDBYT   ;GET LU REG CONTENTS INTO REDBYT
6017  003412  105037  002351                  CLRB    REDBYT+1        ;CLR HI BYTE OF STORAGE
6018  003416  012601                          MOV     (SP)+,R1        ;RESTORE R1
6019  003420  000207                          RTS     PC              ;RETURN
6020
6021
6022
6023
6024
6025                                  ;*************************************************************************************
6026                                  ;* WRITLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
6027                                  ;*       EXECUTE AN INSTRUCTION WHICH LOADS THE BYTE CONTAINED IN WRIBYT
6028                                  ;*       INTO THE LU REG WHOSE NUMBER IS PASSED IN REGNUM.
6029                                  ;*************************************************************************************
6030  003422                          WRITLU:
6031  003422  010146                          MOV     R1,-(SP)        ;SAVE R1
6032  003424  013701  002364                  MOV     REGNUM,R1       ;GET LINE UNIT REG NUMBER
6033  003430  052701  000100                  BIS     #100,R1         ;SET SOURCE = BSEL4
6034  003434  052701  122000                  BIS     #MVIXO,R1       ;SET REST OF MOVE INSTRUCTION
6035  003440  010137  003462                  MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
6036  003444  105037  002353                  CLRB    WRIBYT+1        ;CLR HI BYTE OF STORAGE
6037  003450  113777  002352  176764          MOVB    WRIBYT,@BSEL4   ;LOAD BYTE INTO BSEL4
6038  003456  004737  003240                  JSR     PC,STPCLK       ;EXECUTE MOVE INSTRUCTION
6039  003462  000000              2$:         .WORD   0
6040  003464  012601                          MOV     (SP)+,R1        ;RESTORE R1
6041  003466  000207                          RTS     PC              ;RETURN
6042
6043
6044
6045
6046
6047                                  ;*************************************************************************************
6048                                  ;* GETREG - THIS SUBROUTINE READS THE LINE UNIT REGISTERS 10-17 INTO THE
6049                                  ;*       REGISTER STORAGE TABLE (LUREG:).
6050                                  ;*************************************************************************************
6051  003470  010146                  GETREG: MOV     R1,-(SP)        ;SAVE R1
6052  003472  013746  002364                  MOV     REGNUM,-(SP)    ;SAVE CURRENT REG NO.
6053  003476  012701  002266                  MOV     #LUR10,R1       ;INIT POINTER TO REG STORAGE TABLE
6054  003502  012737  000010  002364          MOV     #10,REGNUM      ;INIT LU REG NO. TO 10
6055  003510  004737  003344          3$:     JSR     PC,READLU       ;READ A LINE UNIT REG
6056  003514  113721  002350                  MOVB    REDBYT,(R1)+    ;PUT BYTE READ INTO TABLE
6057  003520  105021                          CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
6058  003522  005237  002364                  INC     REGNUM          ;INCREMENT REG NO.
6059  003526  023727  002364  000020          CMP     REGNUM,#20      ;SEE IF ALL REGS READ YET
6060  003534  002765                          BLT     3$              ;BR IF NOT
6061  003536  012637  002364                  MOV     (SP)+,REGNUM    ;RESTORE CURRENT REG NO.
6062  003542  012601                          MOV     (SP)+,R1        ;RESTORE R1
6063  003544  000207                          RTS     PC              ;RETURN
6064
```

```
                        ;*****************************************************************************
                        ;* LOOPIN - THIS SUBROUTINE PLACES THE MICROPROCESSOR IN A LOOP ON AN
                        ;*      INSTRUCTION, BY MOVING THE INSTRUCTION FROM THE WORD FOLLOWING THE CALL
                        ;*      INTO SEL6, AND SETTING RUN AND ROMI IN BSEL1. THE SUBROUTINE RETURNS
                        ;*      WITH THE MICROPROCESSOR STUCK IN THE LOOP, AND IF IT IS DESIRED TO
                        ;*      TERMINATE THE LOOP, THE PDP-11 PROGRAM MUST CLEAR THE RUN BIT IN
                        ;*      BSEL1, OR CALL SUBROUTINE MSTCLR TO DO THIS.
                        ;*****************************************************************************
003546                  LOOPIN:
003546  152777  000006  176664          BISB    #ROMO!ROMI,@BSEL1       ;SET ROMO, ROMI BITS IN BSEL1
003554  017677  000000  176662          MOV     @(SP),@SEL6     ;PUT MICROINSTRUCTION INTO SEL6
003562  152777  000206  176650          BISB    #RUN!ROMO!ROMI,@BSEL1 ;SET RUN, ROMO, ROMI IN BSEL1
003570  062716  000002                  ADD     #2,(SP)         ;FIX UP RETURN PC
003574  000207                          RTS     PC              ;RETURN WITH MICROPROCESSOR STUCK IN SINGLE
                                                                ;    INSTRUCTION LOOP


                        ;*****************************************************************************
                        ;* READAX - THIS SUBROUTINE READS THE USYRT REG PAIR WHOSE NUMBER (0-3)
                        ;*      IS PASSED IN BITS 1,2 OF AXNUM ON ENTRY, AND RETURNS THE BYTES READ IN
                        ;*      RAX15 AND RAX16. IF THE LINE UNIT DOES NOT RESPOND WITH READY IN REG 14,
                        ;*      RRDYTO BIT IS SET IN ERROR1 ON RETURN.
                        ;*****************************************************************************
003576  010146         READAX: MOV     R1,-(SP)        ;SAVE R1
003600  013746  002364          MOV     REGNUM,-(SP)    ;STORE CURRENT REG NO.
003604  042737  000001  002410  BIC     #RRDYTO,ERROR1  ;CLEAR ERROR BIT
003612  012737  000014  002364  MOV     #14,REGNUM      ;SET LU REG NO. = 14
003620  113737  002366  002352  MOVB    AXNUM,WRIBYT    ;SET UP AX REG NO. BITS
003626  006237  002352          ASR     WRIBYT
003632  152737  000024  002352  BISB    #RDAX!ENAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
003640  053737  002416  002352  BIS     DISILO,WRIBYT   ;SET PROPER STATE OF DISSI BIT
003646  004737  003422          JSR     PC,WRITLU       ;SET RDAX AND ENAX IN REG 14
003652  005001                  CLR     R1              ;INIT TIMER
003654  004737  003344          6$:     JSR     PC,READLU       ;READ REG 14
003660  132737  000200  002350  BITB    #READY,REDBYT   ;SEE IF READY BIT SET IN REG 14 YET
003666  001006                  BNE     9$              ;BR IF READY SET
003670  005201                  INC     R1              ;INCR TIMER
003672  001370                  BNE     6$              ;BR IF TIMER DIDN'T TIME OUT YET
003674  052737  000001  002410  BIS     #RRDYTO,ERROR1  ;SET ERROR FLAG FOR TIME OUT ON READ RDY
003702  000424                  BR      12$             ;BR TO RETURN
003704  012737  000015  002364  9$:     MOV     #15,REGNUM      ;SET REG NO. = 15
003712  004737  003344          JSR     PC,READLU       ;READ REG 15
003716  113737  002350  002354  MOVB    REDBYT,RAX15    ;STORE REG AX-15
003724  105037  002355          CLRB    RAX15+1         ;CLR HI BYTE OF STORAGE
003730  012737  000016  002364  MOV     #16,REGNUM      ;SET REG NO. = 16
003736  004737  003344          JSR     PC,READLU       ;READ REG 16
003742  113737  002350  002356  MOVB    REDBYT,RAX16    ;STORE REG AX-16
003750  105037  002357          CLRB    RAX16+1         ;CLR HI BYTE OF STORAGE
003754  012637  002364          12$:    MOV     (SP)+,REGNUM    ;RESTORE CURRENT REG NO.
```

```
6121  003760  012601                        MOV     (SP)+,R1        ;RESTORE R1
6122  003762  000207                        RTS     PC              ;RETURN
6123
6124
6125
6126
6127                                  ;*************************************************************************************
6128                                  ;* WRITAX - THIS SUBROUTINE WRITES THE USYRT REG PAIR WHOSE NUMBER (0-3) IS
6129                                  ;*      PASSED IN BITS 1,2 OF AXNUM ON ENTRY, WITH THE DATA FROM WAX15 AND
6130                                  ;*      WAX16. IF LINE UNIT DOES NOT RESPOND WITH READY IN REG 14, WRDYTO BIT
6131                                  ;*      IS SET IN ERROR1 ON RETURN.
6132                                  ;*************************************************************************************
6133  003764  010146                  WRITAX: MOV     R1,-(SP)        ;SAVE R1
6134  003766  013746  002364                  MOV     REGNUM,-(SP)    ;SAVE CURRENT REG NO.
6135  003772  042737  000002  002410          BIC     #WRDYTO,ERROR1  ;CLEAR ERROR BIT
6136  004000  012737  000014  002364          MOV     #14,REGNUM      ;SET LU REG NO. = 14
6137  004006  113737  002366  002352          MOVB    AXNUM,WRIBYT    ;SET AX REG NO. BITS
6138  004014  006237  002352                  ASR     WRIBYT
6139  004020  053737  002416  002352          BIS     DISILO,WRIBYT   ;SET PROPER STATE OF DISSI BIT
6140  004026  004737  003422                  JSR     PC,WRITLU       ;SET AX NO. BITS IN REG 14
6141  004032  012737  000015  002364          MOV     #15,REGNUM      ;SET REG NO. = 15
6142  004040  105037  002361                  CLRB    WAX15+1         ;CLR HI BYTE OF STORAGE
6143  004044  113737  002360  002352          MOVB    WAX15,WRIBYT    ;SET UP BYTE TO WRITE INTO REG 15
6144  004052  004737  003422                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 15
6145  004056  005237  002364                  INC     REGNUM          ;SET REG NO. = 16
6146  004062  105037  002363                  CLRB    WAX16+1         ;CLR HI BYTE OF STORAGE
6147  004066  113737  002362  002352          MOVB    WAX16,WRIBYT    ;SET UP BYTE TO WRITE INTO REG 16
6148  004074  004737  003422                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 16
6149  004100  012737  000014  002364          MOV     #14,REGNUM      ;SET REG NO. = 14
6150  004106  113737  002366  002352          MOVB    AXNUM,WRIBYT    ;SET AX REG NO. BITS
6151  004114  006237  002352                  ASR     WRIBYT
6152  004120  152737  000014  002352          BISB    #ENAX!WAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
6153  004126  053737  002416  002352          BIS     DISILO,WRIBYT   ;SET PROPER STATE OF DISSI BIT
6154  004134  004737  003422                  JSR     PC,WRITLU       ;SET ENAX AND WAX IN REG 14
6155  004140  005001                          CLR     R1              ;INIT PROGRAM TIMER
6156  004142  004737  003344          6$:     JSR     PC,READLU       ;READ REG 14
6157  004146  132737  000200  002350          BITB    #READY,REDBYT   ;SEE IF READY BIT SET IN REG 14 YET
6158  004154  001005                          BNE     9$              ;BR IF READY SET
6159  004156  005201                          INC     R1              ;INCR TIMER
6160  004160  001370                          BNE     6$              ;BR IF TIMER DIDN'T TIME OUT YET
6161  004162  052737  000002  002410          BIS     #WRDYTO,ERROR1  ;SET ERROR FLAG BIT FOR TIME OUT ON WRITE RDY
6162  004170  012637  002364          9$:     MOV     (SP)+,REGNUM    ;RESTORE CURRENT REG NO.
6163  004174  012601                          MOV     (SP)+,R1        ;RESTORE R1
6164  004176  000207                          RTS     PC              ;RETURN
6165
6166
6167
6168
6169
6170                                  ;*************************************************************************************
6171                                  ;* GETALL - THIS SUBROUTINE READS THE LINE UNIT REGS 10-17 AND THE EXTENDED
6172                                  ;*      REGISTERS AX0-AX3 INTO REGISTER STORAGE TABLE (LUREG:).
6173                                  ;*************************************************************************************
6174  004200  010146                  GETALL: MOV     R1,-(SP)        ;SAVE R1
6175  004202  013746  002366                  MOV     AXNUM,-(SP)     ;SAVE CURRENT AX REG BYTE NO.
6176  004206  012737  015172  002516          MOV     #DH5,TMP0       ;SET AX LO BYTE NO.
```

```
6177  004214  032737  000001  002366          BIT     #BIT0,AXNUM     ;SEE IF LO OR HI BYTE
6178  004222  001403                          BEQ     1$              ;BR IF LO BYTE
6179  004224  012737  015175  002516          MOV     #DH6,TMP0       ;SET AX HI BYTE NO.
6180  004232  004737  003470          1$:     JSR     PC,GETREG       ;READ AND STORE REGS 10-17
6181  004236  142777  000010  176174          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
6182  004244  012701  002306                  MOV     #AX0.15,R1      ;INIT POINTER TO REG STORAGE TABLE
6183  004250  005037  002366                  CLR     AXNUM           ;INIT AX REG BYTE NO. TO 0
6184  004254  004737  003576          3$:     JSR     PC,READAX       ;READ 2 AX REG BYTES
6185  004260  113721  002354                  MOVB    RAX15,(R1)+     ;PUT LO BYTE READ INTO TABLE
6186  004264  105021                          CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
6187  004266  113721  002356                  MOVB    RAX16,(R1)+     ;PUT HI BYTE READ INTO TABLE
6188  004272  105021                          CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
6189  004274  062737  000002  002366          ADD     #2,AXNUM        ;INCR AX REG BYTE NO.
6190  004302  023727  002366  000010          CMP     AXNUM,#10       ;SEE IF ALL REGS READ YET
6191  004310  002761                          BLT     3$              ;BR IF NOT
6192  004312  012637  002366                  MOV     (SP)+,AXNUM     ;RESTORE CURRENT AX REG BYTE NO.
6193  004316  012601                          MOV     (SP)+,R1        ;RESTORE R1
6194  004320  013737  002366  002520          MOV     AXNUM,TMP1
6195  004326  006237  002520                  ASR     TMP1            ;GET EXTENDED REG NO. FOR PRINTOUT
6196  004332  000207                          RTS     PC              ;RETURN
6197
6198
6199
6200
6201
6202                          ;*****************************************************************************
6203                          ;* OSIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ORDY (REG 11)
6204                          ;*       AND OCOR (REG 17) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
6205                          ;*       AS PASSED IN BIT 0 (ORDY) AND BIT 1 (OCOR) OF THE WORD FOLLOWING THE
6206                          ;*       CALL.
6207                          ;*       IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS IN
6208                          ;*       RETADR.
6209                          ;*****************************************************************************
6210  004334  013746  002364  OSIRDY: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
6211  004340  013746  002336          MOV     SUBRPC,-(SP)
6212  004344  005737  002336          TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
6213  004350  001006                  BNE     1$              ;BR IF YES
6214  004352  016637  000004  002336  MOV     4(SP),SUBRPC
6215  004360  162737  000004  002336  SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
6216  004366  012737  000011  002364  1$:     MOV     #11,REGNUM      ;SET REG NO. TO 11
6217  004374  004737  003344          JSR     PC,READLU       ;READ REG 11
6218  004400  032776  000001  000004  BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF ORDY
6219  004406  001413                  BEQ     3$              ;BR IF EXPECTED ORDY = 0
6220  004410  132737  000020  002350  BITB    #ORDY,REDBYT    ;SEE IF ORDY = 1
6221  004416  001022                  BNE     9$              ;BR IF ORDY = 1
6222  004420  004737  004200          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6223                          ;REPORT ORDY NOT SET
6224  004424                  ERRDF   7,EM7,ERR4
  (4) 004424  104455                                                      TRAP    C$ERDF
  (5) 004426  000007                                                      .WORD   7
  (5) 004430  013304                                                      .WORD   EM7
  (5) 004432  016356                                                      .WORD   ERR4
6225  004434  000451                  BR      16$             ;TAKE ERROR RETURN
6226  004436  132737  000020  002350  3$:     BITB    #ORDY,REDBYT    ;SEE IF ORDY = 0
6227  004444  001407                  BEQ     9$              ;BR IF ORDY = 0
6228  004446  004737  004200          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
```

```
                                        ;REPORT ORDY NOT CLEARED
                                                ERRDF   8,EM8,ERR4
(4)  004452  104455                                                                     TRAP    C$ERDF
(5)  004454  000010                                                                     .WORD   8
(5)  004456  013321                                                                     .WORD   EM8
(5)  004460  016356                                                                     .WORD   ERR4
6231 004462  000436                            BR      16$             ;TAKE ERROR RETURN
6232 004464  012737  000017  002364  9$:      MOV     #17,REGNUM      ;SET REG NO. = 17
6233 004472  004737  003344                    JSR     PC,READLU       ;READ LU REG 17
6234 004476  132776  000002  000004          BITB    #BIT1,@4(SP)    ;GET EXPECTED STATE OF OCOR
6235 004504  001413                            BEQ     12$             ;BR IF EXPECTED OCOR = 0
6236 004506  132737  000020  002350          BITB    #OCOR,REDBYT    ;SEE IF OCOR = 1
6237 004514  001031                            BNE     20$             ;BR IF OCOR = 1
6238 004516  004737  004200                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6239                                    ;REPORT OCOR NOT SET
                                                ERRDF   9,EM9,ERR4
(4)  004522  104455                                                                     TRAP    C$ERDF
(5)  004524  000011                                                                     .WORD   9
(5)  004526  013342                                                                     .WORD   EM9
(5)  004530  016356                                                                     .WORD   ERR4
6241 004532  000412                            BR      16$             ;TAKE ERROR RETURN
6242 004534  132737  000020  002350  12$:     BITB    #OCOR,REDBYT    ;SEE IF OCOR = 0
6243 004542  001416                            BEQ     20$             ;BR IF OCOR = 0
6244 004544  004737  004200                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6245                                    ;REPORT OCOR NOT CLEARED
                                                ERRDF   10,EM10,ERR4
(4)  004550  104455                                                                     TRAP    C$ERDF
(5)  004552  000012                                                                     .WORD   10
(5)  004554  013357                                                                     .WORD   EM10
(5)  004556  016356                                                                     .WORD   ERR4
6247 004560  016637  000002  002364  16$:      MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
6248 004566  013706  002332                    MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
6249 004572  013746  002346                    MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
6250 004576  000407                            BR      23$
6251 004600  062766  000002  000004  20$:      ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
6252 004606  012637  002336                    MOV     (SP)+,SUBRPC
6253 004612  012637  002364                    MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
6254 004616  000207                  23$:      RTS     PC              ;RETURN
6255
6256
6257
6258
6259
6260                                    ;*******************************************************************************
6261                                    ;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
6262                                    ;*******************************************************************************
6263 004620  010146                  WAIT50: MOV     R1,-(SP)        ;SAVE R1
6264 004622  012701  000310                    MOV     #200.,R1        ;INIT COUNTER
6265 004626  005301                  3$:       DEC     R1              ;DECREMENT COUNTER
6266 004630  001376                            BNE     3$              ;BR IF NOT DONE YET
6267 004632  012601                            MOV     (SP)+,R1        ;RESTORE R1
6268 004634  000207                            RTS     PC              ;RETURN
6269
6270
6271
6272
```

```
                                ;*****************************************************************************
                                ;* STALL - THIS SUBROUTINE STALLS FOR ABOUT A MICRO-SEC.
                                ;*****************************************************************************
004636  000240                  STALL:  NOP
004640  000240                          NOP
004642  000240                          NOP
004644  000207                          RTS     PC


                                ;*****************************************************************************
                                ;* LDTXSI - THIS SUBROUTINE LOADS THE TX SILO (REGS 10,11) WITH THE DATA PASSED
                                ;*       IN BITS 0-11 OF TXWORD.
                                ;*****************************************************************************
004646  013746  002364          LDTXSI: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
004652  042737  170000  002412          BIC     #170000,TXWORD  ;CLEAR UNUSED BITS
004660  012737  000011  002364          MOV     #11,REGNUM      ;SET REG NO. = 11
004666  113737  002413  002352          MOVB    TXWORD+1,WRIBYT ;SET DATA TO BE WRITTEN INTO REG 11
004674  004737  003422                  JSR     PC,WRITLU       ;LOAD DATA INTO REG 11
004700  012737  000010  002364          MOV     #10,REGNUM      ;SET REG NO. = 10
004706  113737  002412  002352          MOVB    TXWORD,WRIBYT   ;SET DATA TO BE WRITTEN INTO REG 10
004714  004737  003422                  JSR     PC,WRITLU       ;LOAD DATA INTO REG 10
004720  012637  002364                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
004724  000207                          RTS     PC              ;RETURN


                                ;*****************************************************************************
                                ;* STPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NO. OF CYCLES PASSED
                                ;*       IN BITS 0-14 OF THE WORD FOLLOWING THE CALL.
                                ;*       IF BIT 15 = 1, A CHECK IS MADE TO DETERMINE IF THE USYRT CHIP TYPE
                                ;*       REQUIRES DECREMENTING THE NO. OF CYCLES BY 1.
                                ;*****************************************************************************
004726  010146                  STPLU:  MOV     R1,-(SP)        ;SAVE R1
004730  017601  000002                  MOV     @2(SP),R1       ;GET DESIRED NO. OF CYCLES
004734  001426                          BEQ     6$              ;IF DESIRED CYCLES = 0, RETURN
004736  100006                          BPL     2$              ;BR IF CHIP TYPE CHECK NOT NECESSARY
004740  042701  100000                  BIC     #BIT15,R1       ;CLEAR FLAG BIT
004744  005737  002420                  TST     CHPTYP          ;SEE IF SIG USYRT
004750  001401                          BEQ     2$              ;BR IF YES
004752  005301                          DEC     R1              ;DECREMENT CYCLE COUNT
004754  152777  000010  175456  2$:     BISB    #LULOOP,@BSEL1  ;SET LU LOOP BIT
004762  152777  000020  175450  3$:     BISB    #STEPLU,@BSEL1  ;SET THE STEPLU BIT (CLOCK THE TRANSMITTER)
004770  004737  004636                  JSR     PC,STALL        ;STALL
004774  142777  000020  175436          BICB    #STEPLU,@BSEL1  ;CLEAR THE STEPLU BIT (CLOCK THE RECEIVER)
005002  004737  004636                  JSR     PC,STALL        ;STALL
005006  005301                          DEC     R1              ;DECREMENT CYCLE COUNTER
005010  001364                          BNE     3$              ;BR IF NOT DONE YET
005012  062766  000002  000002  6$:     ADD     #2,2(SP)        ;FIX UP RETURN PC
005020  012601                          MOV     (SP)+,R1        ;RESTORE R1
005022  000207                          RTS     PC              ;RETURN
```

```
6329
6330
6331
6332
6333
6334                                    ;*****************************************************************************
6335                                    ;* OACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF OACT (REG 11) AND
6336                                    ;*      REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
6337                                    ;*      WORD FOLLOWING THE CALL.
6338                                    ;*****************************************************************************
6339  005024  013746  002364           OACTIV: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
6340  005030  013746  002336                   MOV     SUBRPC,-(SP)
6341  005034  005737  002336                   TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
6342  005040  001006                           BNE     1$              ;BR IF YES
6343  005042  016637  000004  002336           MOV     4(SP),SUBRPC
6344  005050  162737  000004  002336           SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
6345  005056  012737  000011  002364   1$:     MOV     #11,REGNUM      ;SET REG NO. = 11
6346  005064  004737  003344                   JSR     PC,READLU       ;READ REG 11
6347  005070  032776  000001  000004           BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF OACT
6348  005076  001413                           BEQ     3$              ;BR IF EXPECTED OACT = 0
6349  005100  132737  000100  002350           BITB    #OACT,REDBYT    ;SEE IF OACT = 1
6350  005106  001031                           BNE     9$              ;BR IF OACT = 1
6351  005110  004737  004200                   JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6352                                    ;REPORT OACT NOT SET
6353  005114                                    ERRDF   11,EM11,ERR4
  (4) 005114  104455                                                                TRAP    C$ERDF
  (5) 005116  000013                                                                .WORD   11
  (5) 005120  013400                                                                .WORD   EM11
  (5) 005122  016356                                                                .WORD   ERR4
6354  005124  000412                           BR      6$              ;TAKE ERROR RETURN
6355  005126  132737  000100  C02350   3$:     BITB    #OACT,REDBYT    ;SEE IF OACT = 0
6356  005134  001416                           BEQ     9$              ;BR IF OACT = 0
6357  005136  004737  004200                   JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6358                                    ;REPORT OACT NOT CLEARED
6359  005142                                    ERRDF   12,EM12,ERR4
  (4) 005142  104455                                                                TRAP    C$ERDF
  (5) 005144  000014                                                                .WORD   12
  (5) 005146  013415                                                                .WORD   EM12
  (5) 005150  016356                                                                .WORD   ERR4
6360  005152  016637  000002  002364   6$:     MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
6361  005160  013706  002332                   MOV     PSTACK,SP       ;RESTORE PROGRAM STACK TO BASE LEVEL
6362  005164  013746  002346                   MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
6363  005170  000407                           BR      12$
6364  005172  062766  000002  000004   9$:     ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
6365  005200  012637  002336                   MOV     (SP)+,SUBRPC
6366  005204  012637  002364                   MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
6367  005210  000207                   12$:    RTS     PC              ;RETURN
6368
6369
6370
6371
6372
6373                                    ;*****************************************************************************
6374                                    ;* INITRN - THIS SUBROUTINE INITIATES TRANSMISSION OF A MESSAGE, BY DOING A
6375                                    ;*      MASTER CLEAR, LOADING AX2-15 AND REG 17 WITH THE DATA PASSED IN THE 2
6376                                    ;*      WORDS FOLLOWING THE CALL, LOADING 2 SOM CHARS INTO THE TX SILO, AND
```

```
                                        ;*      CLOCKING THE LINE UNIT UNTIL THE FIRST SYNCH OR FLAG HAS BEEN SERIALIZED
                                        ;*      IN THE USYRT. THE PROGRAM MONITORS ORDY,OCOR, AND OACT FOR VALID STATES,
                                        ;*      THROUGHOUT THE PROCESS.
                                        ;*      IF THE SUBROUTINE DETECTS AN ERROR, A RETURN IS MADE TO THE TEST, AT THE
                                        ;*      ADDRESS CONTAINED IN RETADR.
                                        ;*****************************************************************************
005212  010146                  INITRN: MOV     R1,-(SP)        ;SAVE R1
005214  013746  002364                  MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
005220  013746  002366                  MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
005224  016637  000006  002336          MOV     6(SP),SUBRPC
005232  162737  000004  002336          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
005240  004737  003276                  JSR     PC,MSTCLR       ;ISSUE A MASTER CLEAR
005244  004737  004334                  JSR     PC,OSIRDY       ;CHECK ORDY=1, OCOR=0
005250  000001                          1
005252  004737  005024                  JSR     PC,OACTIV       ;CHK OACT=0
005256  000000                          0
005260  012737  000004  002366          MOV     #4,AXNUM        ;SET AX BYTE NO. = 4 FOR AX2
005266  117637  000006  002360          MOVB    @6(SP),WAX15    ;SET DATA BYTE TO LOAD INTO AX2-15
005274  012737  000400  002412          MOV     #TXSOM,TXWORD   ;SET TSOM BIT
005302  113737  002360  002412          MOVB    WAX15,TXWORD    ;SET SYNCH CHAR
005310  005037  002362                  CLR     WAX16
005314  004737  003764                  JSR     PC,WRITAX       ;LOAD AX2
005320  012737  000017  002364          MOV     #17,REGNUM      ;SET REG NO. = 17
005326  062766  000002  000006          ADD     #2,6(SP)        ;INCR POINTER TO NEXT DATA BYTE
005334  117637  000006  002352          MOVB    @6(SP),WRIBYT   ;SET DATA BYTE TO LOAD INTO REG 17
005342  004737  003422                  JSR     PC,WRITLU       ;LOAD REG 17
005346  004737  004646                  JSR     PC,LDTXSI       ;LOAD THE SILO WITH SOM CHAR
005352  004737  004646                  JSR     PC,LDTXSI       ;LOAD ANOTHER SOM INTO SILO
005356  004737  004620                  JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE
005362  004737  004334                  JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
005366  000003                          3
005370  004737  005024                  JSR     PC,OACTIV       ;CHK FOR OACT = 0
005374  000000                          0
005376  005001                          CLR     R1              ;INIT CYCLE COUNTER
005400  012737  000011  002364          MOV     #11,REGNUM      ;SET LU REG NO. = 11
005406  152777  000010  175024  6$:     BISB    #LULOOP,@BSEL1  ;SET LINE UNIT LOOP BIT
005414  152777  000020  175016          BISB    #STEPLU,@BSEL1  ;SET CLOCK BIT
005422  004737  004636                  JSR     PC,STALL        ;STALL FOR MICRO-SEC
005426  004737  003344                  JSR     PC,READLU       ;READ REG 11
005432  132737  000100  002350          BITB    #OACT,REDBYT    ;SEE IF OACT = 1 YET
005440  001014                          BNE     9$              ;BR IF OACT = 1
005442  142777  000020  174770          BICB    #STEPLU,@BSEL1  ;CLEAR CLOCK BIT
005450  004737  004636                  JSR     PC,STALL        ;STALL FOR A MICRO-SEC
005454  005201                          INC     R1              ;INCR CYCLE COUNT
005456  020127  000003                  CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
005462  002751                          BLT     6$              ;BR IF NOT
005464  004737  005024                  JSR     PC,OACTIV       ;CHK FOR OACT = 1
005470  000001                          1
005472  012737  000017  002364  9$:     MOV     #17,REGNUM      ;SET REG NO. = 17
005500  005037  002420                  CLR     CHPTYP          ;CLEAR USYRT CHIP INDICATOR
005504  004737  003344                  JSR     PC,READLU       ;READ REG 17
005510  132737  000020  002350          BITB    #OCOR,REDBYT    ;CHK FOR OCOR CLEARED YET
005516  001403                          BEQ     12$             ;BR IF YES - IT IS SIG CHIP
005520  012737  000001  002420          MOV     #1,CHPTYP       ;SET INDICATOR FOR OTHER CHIP TYPE
005526  142777  000020  174704  12$:    BICB    #STEPLU,@BSEL1  ;CLEAR CLOCK BIT
005534  004737  004636                  JSR     PC,STALL        ;STALL FOR MICRO-SEC
```

```
                        JSR     PC,OSIRDY       ;CHK FOR ORDY = 1, OCOR = 0
                        1
                        ADD     #2,6(SP)        ;FIX UP RETURN PC
                        MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
                        MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
                        MOV     (SP)+,R1        ;RESTORE R1
                        CLR     SUBRPC          ;CLEAR SUBR CALL PC
                        RTS     PC              ;RETURN


                ;*******************************************************************************
                ;* TXCHAR - THIS SUBROUTINE INITIATES TRANSMISSION OF A CHARACTER, BY LOADING
                ;*      THE TX SILO WITH DATA PASSED IN BITS 0-11 OF THE WORD FOLLOWING THE CALL
                ;*      AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES PASSED IN BITS 0-14
                ;*      OF THE SECOND WORD FOLLOWING THE CALL. IF BIT 15 = 1, A CHK IS MADE TO
                ;*      DETERMINE IF THE USYRT CHIP TYPE REQUIRES DECREMENTING THE NO. OF CYCLES
                ;*      BY 1. THE PROGRAM CHECKS FOR VALID STATES OF ORDY,
                ;*      OCOR, AND OACT THROUGHOUT THE PROCESS.
                ;*      IF AN ERROR IS DETECTED, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
                ;*      CONTAINED IN RETADR.
                ;*******************************************************************************
TXCHAR: MOV     R1,-(SP)        ;SAVE R1
        MOV     R2,-(SP)        ;SAVE R2
        MOV     4(SP),SUBRPC
        SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
        MOV     @4(SP),TXWORD   ;GET DATA TO BE TRANSMITTED
        JSR     PC,LDTXSI       ;LOAD THE TX SILO WITH THE DATA
        JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE DOWN SILO
        ADD     #2,4(SP)        ;INCR POINTER
        CLR     R1              ;INIT CYCLE COUNT
        MOV     @4(SP),R2       ;GET DESIRED NO. OF CYCLES
        TST     R2              ;SEE IF CHIP TYPE CHK SHOULD BE MADE
        BPL     9$              ;BR IF NOT
        BIC     #BIT15,R2       ;CLEAR FLAG BIT
        TST     CHPTYP          ;SEE IF SIG USYRT
        BEQ     9$              ;BR IF YES
        DEC     R2              ;DECREMENT NO. OF CYCLES
9$:     JSR     PC,OACTIV       ;CHK OACT = 1
        1
        CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
        BEQ     12$             ;BR IF YES
        JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
        3
        JSR     PC,STPLU        ;STEP LU ONE CYCLE
        1
        INC     R1              ;INCR CYCLE COUNT
        BR      9$
12$:    JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
        1
        ADD     #2,4(SP)        ;FIX UP RETURN PC
        CLR     SUBRPC          ;CLEAR SUBR CALL PC
        MOV     (SP)+,R2        ;RESTORE R2
        MOV     (SP)+,R1        ;RESTORE R1
```

```
 6489  005744  000207                          RTS     PC              ;RETURN
 6490
 6491
 6492
 6493
 6494
 6495                                  ;*************************************************************************************
 6496                                  ;* ISIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ICIR (REG 17)
 6497                                  ;*       AND IRDY (REG 12) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
 6498                                  ;*       AS PASSED IN BIT 0 (ICIR) AND BIT 1 (IRDY) OF THE WORD FOLLOWING THE
 6499                                  ;*       CALL.
 6500                                  ;*       IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS
 6501                                  ;*       IN RETADR.
 6502                                  ;*************************************************************************************
 6503  005746  013746  002364         ISIRDY: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
 6504  005752  013746  002336                 MOV     SUBRPC,-(SP)
 6505  005756  005737  002336                 TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
 6506  005762  001006                         BNE     1$              ;BR IF YES
 6507  005764  016637  000004  002336         MOV     4(SP),SUBRPC
 6508  005772  162737  000004  002336         SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
 6509  006000  012737  000012  002364  1$:    MOV     #12,REGNUM      ;SET REG NO. TO 12
 6510  006006  004737  003344                 JSR     PC,READLU       ;READ REG 12
 6511  006012  032776  000002  000004         BIT     #BIT1,@4(SP)    ;GET EXPECTED STATE OF IRDY
 6512  006020  001413                         BEQ     3$              ;BR IF EXPECTED IRDY = 0
 6513  006022  132737  000020  002350         BITB    #IRDY,REDBYT    ;SEE IF IRDY = 1
 6514  006030  001022                         BNE     9$              ;BR IF IRDY = 1
 6515  006032  004737  004200                 JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6516                                 ;REPORT IRDY NOT SET
 6517  006036                                 ERRDF   17,EM17,ERR4
  (4)  006036  104455                                                            TRAP    C$ERDF
  (5)  006040  000021                                                            .WORD   17
  (5)  006042  013436                                                            .WORD   EM17
  (5)  006044  016356                                                            .WORD   ERR4
 6518  006046  000451                         BR      16$             ;TAKE ERROR EXIT
 6519  006050  132737  000020  002350  3$:    BITB    #IRDY,REDBYT    ;SEE IF IRDY = 0
 6520  006056  001407                         BEQ     9$              ;BR IF IRDY = 0
 6521  006060  004737  004200                 JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6522                                 ;REPORT IRDY NOT CLEARED
 6523  006064                                 ERRDF   18,EM18,ERR4
  (4)  006064  104455                                                            TRAP    C$ERDF
  (5)  006066  000022                                                            .WORD   18
  (5)  006070  013453                                                            .WORD   EM18
  (5)  006072  016356                                                            .WORD   ERR4
 6524  006074  000436                         BR      16$             ;TAKE ERROR RETURN
 6525  006076  012737  000017  002364  9$:    MOV     #17,REGNUM      ;SET REG NO. = 17
 6526  006104  004737  003344                 JSR     PC,READLU       ;READ REG 17
 6527  006110  132776  000001  000004         BITB    #BIT0,@4(SP)    ;GET EXPECTED STATE OF ICIR
 6528  006116  001413                         BEQ     12$             ;BR IF EXPECTED ICIR = 0
 6529  006120  132737  000010  002350         BITB    #ICIR,REDBYT    ;SEE IF ICIR = 1
 6530  006126  001031                         BNE     20$             ;BR IF ICIR = 1
 6531  006130  004737  004200                 JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6532                                 ;REPORT ICIR NOT SET
 6533  006134                                 ERRDF   19,EM19,ERR4
  (4)  006134  104455                                                            TRAP    C$ERDF
  (5)  006136  000023                                                            .WORD   19
  (5)  006140  013474                                                            .WORD   EM19
```

```
  (5)  006142  016356                                                                    .WORD   ERR4
6534   006144  000412                    BR      16$             ;TAKE ERROR RETURN
6535   006146  132737  000010  002350  12$:  BITB    #ICIR,REDBYT    ;SEE IF ICIR = 0
6536   006154  001416                    BEQ     20$             ;BR IF ICIR = 0
6537   006156  004737  004200            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6538                                  ;REPORT ICIR NOT CLEARED
6539   006162                         ERRDF   20,EM20,ERR4
  (4)  006162  104455                                                                    TRAP    C$ERDF
  (5)  006164  000024                                                                    .WORD   20
  (5)  006166  013511                                                                    .WORD   EM20
  (5)  006170  016356                                                                    .WORD   ERR4
6540   006172  016637  000002  002364  16$:  MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
6541   006200  013706  002332            MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
6542   006204  013746  002346            MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
6543   006210  000407                    BR      23$
6544   006212  062766  000002  000004  20$:  ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
6545   006220  012637  002336            MOV     (SP)+,SUBRPC
6546   006224  012637  002364            MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
6547   006230  000207                  23$:  RTS     PC              ;RETURN
6548
6549
6550
6551
6552
6553                                  ;*****************************************************************************
6554                                  ;* IACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF IACT (REG 12) AND
6555                                  ;*      REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
6556                                  ;*      WORD FOLLOWING THE CALL.
6557                                  ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN
6558                                  ;*      RETADR.
6559                                  ;*****************************************************************************
6560   006232  013746  002364        IACTIV: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
6561   006236  013746  002336            MOV     SUBRPC,-(SP)
6562   006242  005737  002336            TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
6563   006246  001006                    BNE     1$              ;BR IF YES
6564   006250  016637  000004  002336        MOV     4(SP),SUBRPC
6565   006256  162737  000004  002336        SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
6566   006264  012737  000012  002364  1$:   MOV     #12,REGNUM      ;SET REG NO. = 12
6567   006272  004737  003344            JSR     PC,READLU       ;READ REG 12
6568   006276  032776  000001  000004        BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF IACT
6569   006304  001413                    BEQ     3$              ;BR IF EXPECTED IACT = 0
6570   006306  132737  000100  002350        BITB    #IACT,REDBYT    ;SEE IF IACT = 1
6571   006314  001031                    BNE     9$              ;BR IF IACT = 1
6572   006316  004737  004200            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6573                                  ;REPORT IACT NOT SET
6574   006322                         ERRDF   21,EM21,ERR4
  (4)  006322  104455                                                                    TRAP    C$ERDF
  (5)  006324  000025                                                                    .WORD   21
  (5)  006326  013532                                                                    .WORD   EM21
  (5)  006330  016356                                                                    .WORD   ERR4
6575   006332  000412                    BR      6$              ;TAKE ERROR EXIT
6576   006334  132737  000100  002350  3$:   BITB    #IACT,REDBYT    ;SEE IF IACT = 0
6577   006342  001416                    BEQ     9$              ;BR IF IACT = 0
6578   006344  004737  004200            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
6579                                  ;REPORT IACT NOT CLEARED
6580   006350                         ERRDF   22,EM22,ERR4
```

```
   (4)  006350  104455                                                                    TRAP    C$ERDF
   (5)  006352  000026                                                                    .WORD   22
   (5)  006354  013547                                                                    .WORD   EM22
   (5)  006356  016356                                                                    .WORD   ERR4
 6581   006360  016637  000002  002364  6$:     MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
 6582   006366  013706  002332                  MOV     PSTACK,SP       ;RESTORE PROGRAM STACK TO BASE LEVEL
 6583   006372  013746  002346                  MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
 6584   006376  000407                          BR      12$
 6585   006400  062766  000002  000004  9$:     ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
 6586   006406  012637  002336                  MOV     (SP)+,SUBRPC
 6587   006412  012637  002364                  MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
 6588   006416  000207                  12$:    RTS     PC              ;RETURN
 6589
 6590
 6591
 6592
 6593
 6594                                   ;*****************************************************************************
 6595                                   ;* RSEOM - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF RSOM AND REOM IN
 6596                                   ;* AX0-16, AND REPORTS AN ERROR IF EITHER IS NOT SET TO THE STATE PASSED IN BITS
 6597                                   ;* 0,1, RESPECTIVELY, OF THE WORD FOLLOWING THE CALL.
 6598                                   ;* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN RETADR.
 6599                                   ;*****************************************************************************
 6600   006420  013746  002366          RSEOM:  MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
 6601   006424  013746  002336                  MOV     SUBRPC,-(SP)
 6602   006430  005737  002336                  TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
 6603   006434  001006                          BNE     1$              ;BR IF YES
 6604   006436  016637  000004  002336          MOV     4(SP),SUBRPC
 6605   006444  162737  000004  002336          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
 6606   006452  012737  000001  002366  1$:     MOV     #1,AXNUM        ;SET AX BYTE NO. FOR AX0-16
 6607   006460  004737  003576                  JSR     PC,READAX       ;READ AX0
 6608   006464  032776  000001  000004          BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF RSOM
 6609   006472  001413                          BEQ     3$              ;BR IF EXPECTED RSOM = 0
 6610   006474  132737  000001  002356          BITB    #RSOM,RAX16     ;SEE IF RSOM = 1
 6611   006502  001022                          BNE     9$              ;BR IF RSOM = 1
 6612   006504  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6613                                  ;REPORT RSOM NOT SET
 6614   006510                                 ERRDF   29,EM29,ERR6
   (4)  006510  104455                                                                    TRAP    C$ERDF
   (5)  006512  000035                                                                    .WORD   29
   (5)  006514  013611                                                                    .WORD   EM29
   (5)  006516  017546                                                                    .WORD   ERR6
 6615   006520  000444                          BR      16$             ;TAKE ERROR EXIT
 6616   006522  132737  000001  002356  3$:     BITB    #RSOM,RAX16     ;SEE IF RSOM = 0
 6617   006530  001407                          BEQ     9$              ;BR IF RSOM = 0
 6618   006532  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6619                                  ;REPORT RSOM NOT CLEARED
 6620   006536                                 ERRDF   28,EM28,ERR6
   (4)  006536  104455                                                                    TRAP    C$ERDF
   (5)  006540  000034                                                                    .WORD   28
   (5)  006542  013570                                                                    .WORD   EM28
   (5)  006544  017546                                                                    .WORD   ERR6
 6621   006546  000431                          BR      16$             ;TAKE ERROR RETURN
 6622   006550  132776  000002  000004  9$:     BITB    #BIT1,@4(SP)    ;GET EXPECTED STATE OF REOM
 6623   006556  001413                          BEQ     12$             ;BR IF EXPECTED REOM = 0
 6624   006560  132737  000002  002356          BITB    #REOM,RAX16     ;SEE IF REOM = 1
```

```
                                        BNE     20$             ;BR IF REOM = 1
                                        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT REOM NOT SET
                                        ERRDF   31,EM31,ERR6
                                                                                TRAP    C$ERDF
                                                                                .WORD   31
                                                                                .WORD   EM31
                                                                                .WORD   ERR6
                                        BR      16$             ;TAKE ERROR RETURN
                                12$:    BITB    #REOM,RAX16     ;SEE IF REOM = 0
                                        BEQ     20$             ;BR IF REOM = 0
                                        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT REOM NOT CLEARED
                                        ERRDF   30,EM30,ERR6
                                                                                TRAP    C$ERDF
                                                                                .WORD   30
                                                                                .WORD   EM30
                                                                                .WORD   ERR6
                                16$:    MOV     2(SP),AXNUM     ;RESTORE AX BYTE NO.
                                        MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
                                        MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
                                        BR      23$
                                20$:    ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
                                        MOV     (SP)+,SUBRPC
                                        MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
                                23$:    RTS     PC              ;RETURN


;*****************************************************************************
;* RDRXSI - THIS SUBROUTINE READS THE RCV SILO (REGS 10,12) AND RETURNS THE
;*      SILO ENTRY IN BITS 0-11 OF RXWORD.
;*****************************************************************************
RDRXSI: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
        MOV     #12,REGNUM      ;SET REG NO. = 12
        JSR     PC,READLU       ;READ LU REG 12
        MOVB    REDBYT,RXWORD+1 ;GET HI BITS OF SILO ENTRY
        BIC     #170000,RXWORD  ;CLEAR UNUSED BITS
        MOV     #10,REGNUM      ;SET REG NO. = 10
        JSR     PC,READLU       ;READ REG 10
        MOVB    REDBYT,RXWORD   ;GET LOW BITS OF SILO ENTRY
        MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
        RTS     PC              ;RETURN


;*****************************************************************************
;* RCV1ST - THIS SUBROUTINE RECEIVES THE FIRST CHAR OF A MESSAGE, AND MONITORS
;*      STATUS OF THE RECEIVER. FIRST, A CHECK IS MADE FOR IACT = 0, IRDY = 0,
;*      ICIR = 1, AND RSOM = 0. THEN, THE LINE UNIT IS CLOCKED USING
;*      STEPLU UNTIL IRDY = 1. THE PROGRAM CHECKS FOR THIS TO OCCUR WITHIN 3
;*      CYCLES AFTER THE NO. OF CYCLES PASSED IN THE WORD FOLLOWING THE CALL.
```

```
6673                                    ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
6674                                    ;*      CONTAINED IN RETADR.
6675                                    ;*****************************************************************************
6676  006752  010146                    RCV1ST: MOV     R1,-(SP)        ;SAVE R1
6677  006754  010346                            MOV     R3,-(SP)        ;SAVE R3
6678  006756  013746  002364                    MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
6679  006762  016637  000006  002336            MOV     6(SP),SUBRPC
6680  006770  162737  000004  002336            SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
6681  006776  012737  000012  002364            MOV     #12,REGNUM      ;SET LU REG NO. = 12
6682  007004  005001                            CLR     R1              ;INIT CYCLE COUNT TO 0
6683  007006  017603  000006                    MOV     @6(SP),R3       ;GET CYCLE COUNT LIMIT
6684  007012  062703  000003                    ADD     #3,R3
6685  007016  005776  000006                    TST     @6(SP)          ;SEE IF DESIRED CYCLES = 0
6686  007022  001414                            BEQ     8$              ;BR IF YES
6687  007024  004737  006232                    JSR     PC,IACTIV       ;CHK FOR IACT = 0
6688  007030  000000                            0
6689  007032  004737  005746                    JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 0
6690  007036  000001                            1
6691  007040  004737  006420                    JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0 IN AX0-16
6692  007044  000000                            0
6693  007046  004737  004726            6$:     JSR     PC,STPLU        ;CLOCK LU FOR 1 CYCLE
6694  007052  000001                            1
6695  007054  004737  004620            8$:     JSR     PC,WAIT50       ;ALLOW SILO DATA TO RIPPLE
6696  007060  005201                            INC     R1              ;INCREMENT CYCLE COUNT
6697  007062  004737  003344                    JSR     PC,READLU       ;READ REG 12
6698  007066  132737  000020  002350            BITB    #IRDY,REDBYT    ;SEE IF IRDY = 1 YET
6699  007074  001005                            BNE     9$              ;BR IF IRDY = 1
6700  007076  020103                            CMP     R1,R3           ;SEE IF LIMIT EXCEEDED
6701  007100  002762                            BLT     6$              ;BR IF NOT YET
6702  007102  004737  005746                    JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 1
6703  007106  000003                            3
6704  007110  020176  000006            9$:     CMP     R1,@6(SP)       ;SEE IF LESS THAN REQUIRED CYCLES
6705  007114  002003                            BGE     12$             ;BR IF NOT
6706  007116  004737  005746                    JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 0
6707  007122  000001                            1
6708  007124  004737  006232            12$:    JSR     PC,IACTIV       ;CHK FOR IACT = 1
6709  007130  000001                            1
6710  007132  004737  005746                    JSR     PC,ISIRDY       ;CHK FOR ICIR = 1, IRDY = 1
6711  007136  000003                            3
6712  007140  062766  000002  000006            ADD     #2,6(SP)        ;FIX UP RETURN PC
6713  007146  012637  002364                    MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
6714  007152  012603                            MOV     (SP)+,R3        ;RESTORE R3
6715  007154  012601                            MOV     (SP)+,R1        ;RESTORE R1
6716  007156  005037  002336                    CLR     SUBRPC          ;CLEAR SUBR CALL PC
6717  007162  000207                            RTS     PC              ;RETURN
6718
6719
6720
6721
6722
6723                                    ;*****************************************************************************
6724                                    ;* STPERR - THIS SUBROUTINE LOADS THE CONTENTS OF THE FIRST WORD FOLLOWING THE
6725                                    ;*      CALL INTO REG 17, AND SETS THE IERR BIT, AND CLOCKS THE LINE UNIT
6726                                    ;*      FOR THE NO. OF CYCLES PASSED IN THE 2ND WORD FOLLOWING THE CALL. THEN,
6727                                    ;*      IT RESTORES REG 17 TO ITS ORIGINAL CONTENTS, CLEARING THE IERR BIT.
6728                                    ;*****************************************************************************
```

```
6729  007164  013746  002364          STPERR: MOV     REGNUM,-(SP)      ;SAVE LU REG NO.
6730  007170  012737  000017  002364          MOV     #17,REGNUM        ;SET LU REG NO. = 17
6731  007176  017637  000002  002352          MOV     @2(SP),WRIBYT
6732  007204  152737  000002  002352          BISB    #IERR,WRIBYT
6733  007212  004737  003422                  JSR     PC,WRITLU         ;SET IERR BIT IN REG 17
6734  007216  062766  000002  000002          ADD     #2,2(SP)          ;INCREMENT SUBR ARGUMENT POINTER
6735  007224  017637  000002  007236          MOV     @2(SP),3$         ;GET DESIRED NO. OF CYCLES
6736  007232  004737  004726                  JSR     PC,STPLU          ;CLOCK LU FOR DESIRED NO. OF CYCLES
6737  007236  000000                  3$:     .WORD   0                 ;NO. OF CYCLES GOES HERE
6738  007240  142737  000002  002352          BICB    #IERR,WRIBYT
6739  007246  004737  003422                  JSR     PC,WRITLU         ;CLEAR IERR BIT IN REG 17
6740  007252  062766  000002  000002          ADD     #2,2(SP)          ;FIX UP RETURN PC
6741  007260  012637  002364                  MOV     (SP)+,REGNUM      ;RESTORE LU REG NO.
6742  007264  000207                          RTS     PC                ;RETURN
6743
6744
6745
6746
6747
6748                                  ;*************************************************************************
6749                                  ;* CKDATA - THIS SUBROUTINE READS THE RCV SILO AND COMPARES THE SILO ENTRY
6750                                  ;*      TO BITS 0-11 OF THE FIRST WORD FOLLOWING THE CALL. IF THERE IS A
6751                                  ;*      MISMATCH, THE ERROR IS REPORTED AND A RETURN IS MADE TO THE TEST AT THE
6752                                  ;*      ADDRESS CONTAINED IN RETADR. IF BIT 15 = 0 IN THE FIRST WORD
6753                                  ;*      FOLLOWING THE CALL, THE SUBROUTINE WILL NOT CHECK THE BCC BIT (SILO
6754                                  ;*      BIT 8). IF THERE ARE NO ERRORS, THE LINE UNIT IS CLOCKED FOR THE
6755                                  ;*      NUMBER OF CYCLES PASSED IN THE SECOND WORD FOLLOWING THE CALL.
6756                                  ;*************************************************************************
6757  007266  010146                  CKDATA: MOV     R1,-(SP)          ;SAVE R1
6758  007270  013746  002364                  MOV     REGNUM,-(SP)      ;SAVE LU REG NO.
6759  007274  016637  000004  002336          MOV     4(SP),SUBRPC
6760  007302  162737  000004  002336          SUB     #4,SUBRPC         ;GET PC OF SUBR CALL
6761  007310  017601  000004                  MOV     @4(SP),R1         ;GET EXPECTED SILO ENTRY
6762  007314  042701  170000                  BIC     #170000,R1        ;CLEAR UNUSED BITS FOR COMPARE
6763  007320  004737  006672                  JSR     PC,RDRXSI         ;READ RCV SILO
6764  007324  023727  002424  000347          CMP     SAVLEN,#TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
6765  007332  001005                          BNE     4$                ;BR IF CHAR LENGTH NOT = 7
6766  007334  042701  000200                  BIC     #BIT7,R1          ;MASK OFF 8TH BIT
6767  007340  042737  000200  002414          BIC     #BIT7,RXWORD
6768  007346  120137  002414          4$:     CMPB    R1,RXWORD         ;COMPARE EXPECTED BITS 0-7 TO ACTUAL
6769  007352  001445                          BEQ     6$                ;BR IF MATCH
6770  007354  005037  002370                  CLR     GOODAT
6771  007360  110137  002370                  MOVB    R1,GOODAT         ;GET EXPECTED DATA
6772  007364  005037  002372                  CLR     BADDAT
6773  007370  113737  002414  002372          MOVB    RXWORD,BADDAT     ;GET ACTUAL DATA
6774  007376  012737  000011  002364          MOV     #11,REGNUM        ;SET REG NO. = 11
6775  007404  004737  003344                  JSR     PC,READLU         ;READ REG 11
6776  007410  132737  000001  002350          BITB    #UNRR,REDBYT      ;SEE IF TX UNDERRUN ERROR
6777  007416  001410                          BEQ     5$                ;BR IF NOT,TO REPORT DATA ERROR
6778  007420  004737  004200                  JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
6779                                  ;REPORT TX UNDERRUN ERROR
6780  007424                                  ERRDF   54,EM54,ERR4
  (4) 007424  104455                                                    TRAP    C$ERDF
  (5) 007426  000066                                                    .WORD   54
  (5) 007430  014541                                                    .WORD   EM54
  (5) 007432  016356                                                    .WORD   ERR4
```

```
 6781  007434  000137  010116                    JMP     36$                ;TAKE ERROR EXIT
 6782  007440  012737  000010  002364  5$:       MOV     #10,REGNUM         ;SET REG NO. = 10
 6783  007446  004737  004200                    JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
 6784                                  ;REPORT RCV'D DATA MISCOMPARE
 6785  007452                                    ERRDF   34,EM34,ERR8
   (4) 007452  104455                                                       TRAP    C$ERDF
   (5) 007454  000042                                                       .WORD   34
   (5) 007456  013664                                                       .WORD   EM34
   (5) 007460  020656                                                       .WORD   ERR8
 6786  007462  000137  010116                    JMP     36$                ;TAKE ERROR EXIT
 6787  007466  000301                6$:       SWAB    R1
 6788  007470  012737  000012  002364          MOV     #12,REGNUM         ;SET LU REG NO. FOR ERROR REPORTS
 6789  007476  120137  002415                    CMPB    R1,RXWORD+1        ;COMPARE EXPECTED SILO BITS 8-11 TO ACTUAL
 6790  007502  001002                          BNE     7$                 ;BR IF MISMATCH
 6791  007504  000137  010072                    JMP     22$                ;CONTINUE
 6792  007510  005037  002370          7$:       CLR     GOODAT
 6793  007514  110137  002370                    MOVB    R1,GOODAT          ;SET EXPECTED DATA
 6794  007520  005037  002372                    CLR     BADDAT
 6795  007524  113737  002415  002372          MOVB    RXWORD+1,BADDAT    ;SET ACTUAL DATA
 6796  007532  032776  100000  000004          BIT     #BCCCHK,@4(SP)     ;SEE IF BCC SHOULD BE IGNORED
 6797  007540  001433                          BEQ     10$                ;BR IF YES
 6798  007542  132701  000001                  BITB    #BCC,R1            ;SEE IF EXPECTED BIT = 1
 6799  007546  001014                          BNE     8$                 ;BR IF YES
 6800  007550  132737  000001  002415          BITB    #BCC,RXWORD+1      ;SEE IF ACTUAL BIT = 0
 6801  007556  001424                          BEQ     10$                ;BR IF YES
 6802  007560  004737  004200                  JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
 6803                                  ;REPORT BCC NOT CLEARED
 6804  007564                                    ERRDF   35,EM35,ERR8
   (4) 007564  104455                                                       TRAP    C$ERDF
   (5) 007566  000043                                                       .WORD   35
   (5) 007570  013712                                                       .WORD   EM35
   (5) 007572  020656                                                       .WORD   ERR8
 6805  007574  000137  010116                    JMP     36$                ;TAKE ERROR EXIT
 6806  007600  132737  000001  002415  8$:       BITB    #BCC,RXWORD+1      ;SEE IF ACTUAL BIT = 1
 6807  007606  001010                          BNE     10$                ;BR IF YES
 6808  007610  004737  004200                  JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
 6809                                  ;REPORT BCC NOT SET
 6810  007614                                    ERRDF   36,EM36,ERR8
   (4) 007614  104455                                                       TRAP    C$ERDF
   (5) 007616  000044                                                       .WORD   36
   (5) 007620  013732                                                       .WORD   EM36
   (5) 007622  020656                                                       .WORD   ERR8
 6811  007624  000137  010116                    JMP     36$                ;TAKE ERROR EXIT
 6812  007630                          10$:
 6813  007630  132701  000002                  BITB    #EBLK,R1           ;SEE IF EXPECTED BIT = 1
 6814  007634  001014                          BNE     12$                ;BR IF YES
 6815  007636  132737  000002  002415          BITB    #EBLK,RXWORD+1     ;SEE IF ACTUAL BIT = 0
 6816  007644  001424                          BEQ     14$                ;BR IF YES
 6817  007646  004737  004200                  JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
 6818                                  ;REPORT EBLK NOT CLEARED
 6819  007652                                    ERRDF   37,EM37,ERR8
   (4) 007652  104455                                                       TRAP    C$ERDF
   (5) 007654  000045                                                       .WORD   37
   (5) 007656  013746                                                       .WORD   EM37
   (5) 007660  020656                                                       .WORD   ERR8
 6820  007662  000137  010116                    JMP     36$                ;TAKE ERROR EXIT
```

```
 6821  007666  132737  000002  002415  12$:    BITB    #EBLK,RXWORD+1  ;SEE IF ACTUAL BIT = 1
 6822  007674  001010                          BNE     14$             ;BR IF YES
 6823  007676  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6824                                  ;REPORT EBLK NOT SET
 6825  007702                                  ERRDF   38,EM38,ERR8
   (4) 007702  104455                                                  TRAP    C$ERDF
   (5) 007704  000046                                                  .WORD   38
   (5) 007706  013767                                                  .WORD   EM38
   (5) 007710  020656                                                  .WORD   ERR8
 6826  007712  000137  010116                  JMP     36$             ;TAKE ERROR EXIT
 6827  007716                          14$:
 6828  007716  132701  000004                  BITB    #RAB,R1         ;SEE IF EXPECTED BIT = 1
 6829  007722  001014                          BNE     16$             ;BR IF YES
 6830  007724  132737  000004  002415          BITB    #RAB,RXWORD+1   ;SEE IF ACTUAL BIT = 0
 6831  007732  001424                          BEQ     18$             ;BR IF YES
 6832  007734  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6833                                  ;REPORT RAB NOT CLEARED
 6834  007740                                  ERRDF   39,EM39,ERR8
   (4) 007740  104455                                                  TRAP    C$ERDF
   (5) 007742  000047                                                  .WORD   39
   (5) 007744  014004                                                  .WORD   EM39
   (5) 007746  020656                                                  .WORD   ERR8
 6835  007750  000137  010116                  JMP     36$             ;TAKE ERROR EXIT
 6836  007754  132737  000004  002415  16$:    BITB    #RAB,RXWORD+1   ;SEE IF ACTUAL BIT = 1
 6837  007762  001010                          BNE     18$             ;BR IF YES
 6838  007764  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6839                                  ;REPORT RAB NOT SET
 6840  007770                                  ERRDF   40,EM40,ERR8
   (4) 007770  104455                                                  TRAP    C$ERDF
   (5) 007772  000050                                                  .WORD   40
   (5) 007774  014024                                                  .WORD   EM40
   (5) 007776  020656                                                  .WORD   ERR8
 6841  010000  000137  010116                  JMP     36$             ;TAKE ERROR EXIT
 6842  010004                          18$:
 6843  010004  132701  000010                  BITB    #OVRR,R1        ;SEE IF EXPECTED BIT = 1
 6844  010010  001014                          BNE     20$             ;BR IF YES
 6845  010012  132737  000010  002415          BITB    #OVRR,RXWORD+1  ;SEE IF ACTUAL BIT = 0
 6846  010020  001424                          BEQ     22$             ;BR IF YES
 6847  010022  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6848                                  ;REPORT OVRR NOT CLEARED
 6849  010026                                  ERRDF   41,EM41,ERR8
   (4) 010026  104455                                                  TRAP    C$ERDF
   (5) 010030  000051                                                  .WORD   41
   (5) 010032  014040                                                  .WORD   EM41
   (5) 010034  020656                                                  .WORD   ERR8
 6850  010036  000137  010116                  JMP     36$             ;TAKE ERROR EXIT
 6851  010042  132737  000010  002415  20$:    BITB    #OVRR,RXWORD+1  ;SEE IF ACTUAL BIT = 1
 6852  010050  001010                          BNE     22$             ;BR IF YES
 6853  010052  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 6854                                  ;REPORT OVRR NOT SET
 6855  010056                                  ERRDF   42,EM42,ERR8
   (4) 010056  104455                                                  TRAP    C$ERDF
   (5) 010060  000052                                                  .WORD   42
   (5) 010062  014061                                                  .WORD   EM42
   (5) 010064  020656                                                  .WORD   ERR8
 6856  010066  000137  010116                  JMP     36$             ;TAKE ERROR EXIT
```

```
6857  010072                                  22$:
6858  010072  062766  000002  000004          ADD     #2,4(SP)          ;INCR SUBROUTINE ARGUMENT POINTER
6859  010100  017637  000004  010112          MOV     @4(SP),24$        ;GET DESIRED CYCLE COUNT
6860  010106  004737  004726                  JSR     PC,STPLU          ;CLOCK LU FOR DESIRED CYCLES
6861  010112  000000                  24$:    .WORD   0
6862  010114  000407                          BR      38$               ;TAKE ERROR-FREE EXIT
6863  010116  011637  002364          36$:    MOV     (SP),REGNUM       ;RESTORE LU REG NO.
6864  010122  013706  002332                  MOV     PSTACK,SP         ;RESTORE PROGRAM STACK TO BASE LEVEL
6865  010126  013746  002346                  MOV     RETADR,-(SP)      ;FIX UP ERROR RETURN PC
6866  010132  000406                          BR      40$
6867  010134  062766  000002  000004  38$:    ADD     #2,4(SP)          ;FIX UP ERROR-FREE RETURN PC
6868  010142  012637  002364                  MOV     (SP)+,REGNUM      ;RESTORE LU REG NO.
6869  010146  012601                          MOV     (SP)+,R1          ;RESTORE R1
6870  010150  005037  002336          40$:    CLR     SUBRPC            ;CLEAR SUBROUTINE PC
6871  010154  000207                          RTS     PC                ;RETURN
6872
6873
6874
6875
6876
6877                                  ;*****************************************************************************
6878                                  ;* LODATA - THIS SUBROUTINE LOADS THE TRANSMITTER SILO WITH 2 SOM'S, THE DATA
6879                                  ;*      IN PATTERN A REPEATED 3 TIMES (60 CHARS), AND 2 EOM'S. IN ADDITION, THE
6880                                  ;*      DATA CHARS ARE ALSO LOADED INTO THE RECEIVED MSG BUFFER (RCVBUF:), AS
6881                                  ;*      EXPECTED DATA FOR LATER COMPARISON.
6882                                  ;*****************************************************************************
6883  010156  010146          LODATA: MOV     R1,-(SP)          ;SAVE R1
6884  010160  010346                  MOV     R3,-(SP)          ;SAVE R3
6885  010162  010446                  MOV     R4,-(SP)          ;SAVE R4
6886  010164  012737  000400  002412  MOV     #TXSOM,TXWORD
6887  010172  004737  004646          JSR     PC,LDTXSI         ;LOAD AN SOM INTO TX SILO
6888  010176  004737  004646          JSR     PC,LDTXSI         ;LOAD ANOTHER SOM
6889  010202  012701  000003          MOV     #3,R1             ;INIT COUNTER
6890  010206  012704  002762          MOV     #RCVBUF,R4        ;GET POINTER TO RCV BUF
6891  010212  012703  002557  3$:     MOV     #PATA,R3          ;GET POINTER TO DATA PATTERN
6892  010216  005037  002412  6$:     CLR     TXWORD
6893  010222  112337  002412          MOVB    (R3)+,TXWORD      ;GET A DATA CHAR
6894  010226  013724  002412          MOV     TXWORD,(R4)+      ;LOAD A DATA CHAR INTO RCV BUF
6895  010232  004737  004646          JSR     PC,LDTXSI         ;LOAD DATA CHAR INTO TX SILO
6896  010236  020327  002603          CMP     R3,#PATB          ;SEE IF AT END OF PATTERN A YET
6897  010242  103765                  BLO     6$                ;BR IF NOT YET
6898  010244  005301                  DEC     R1                ;DECREMENT COUNTER
6899  010246  001361                  BNE     3$                ;BR IF NOT DONE YET
6900  010250  052764  100400  177776  BIS     #CRCCHK!RXBCC,-2(R4) ;SET UP TO CHK BCC = 1 ON LAST DATA CHAR
6901  010256  012737  001000  002412  MOV     #TXEOM,TXWORD
6902  010264  004737  004646          JSR     PC,LDTXSI         ;LOAD AN EOM INTO TX SILO
6903  010270  004737  004646          JSR     PC,LDTXSI         ;LOAD ANOTHER EOM
6904  010274  012604                  MOV     (SP)+,R4          ;RESTORE R4
6905  010276  012603                  MOV     (SP)+,R3          ;RESTORE R3
6906  010300  012601                  MOV     (SP)+,R1          ;RESTORE R1
6907  010302  000207                  RTS     PC                ;RETURN
6908
6909
6910
6911
6912
```

```
                        ;*************************************************************************
                        ;* 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 WORD INTO AX3-16.
                        ;*************************************************************************
010304 013746 002366    SETUP:  MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
010310 013746 002364            MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
010314 012737 000004 002366     MOV     #4,AXNUM        ;SET AX BYTE NO. FOR AX2
010322 017637 000004 002360     MOV     @4(SP),WAX15
010330 005037 002362            CLR     WAX16
010334 004737 003764            JSR     PC,WRITAX       ;SET SYNCH CHAR IN AX2-15, CLEAR AX2-16
010340 012737 000017 002364     MOV     #17,REGNUM      ;SET LU REG NO. = 17
010346 062766 000002 000004     ADD     #2,4(SP)        ;INCREMENT ARGUMENT POINTER
010354 017637 000004 002352     MOV     @4(SP),WRIBYT
010362 004737 003422            JSR     PC,WRITLU       ;LOAD REG 17
010366 012737 000006 002366     MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
010374 062766 000002 000004     ADD     #2,4(SP)        ;INCREMENT ARGUMENT POINTER
010402 017637 000004 002360     MOV     @4(SP),WAX15
010410 062766 000002 000004     ADD     #2,4(SP)        ;INCREMENT ARGUMENT POINTER
010416 017637 000004 002362     MOV     @4(SP),WAX16
010424 013737 002362 002424     MOV     WAX16,SAVLEN    ;STORE TX AND RCV CHAR LENGTHS
010432 142777 000010 172000     BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
010440 004737 003764            JSR     PC,WRITAX       ;LOAD AX3-15, AX3-16
010444 152777 000010 171766     BISB    #LULOOP,@BSEL1  ;SET LULOOP
010452 062766 000002 000004     ADD     #2,4(SP)        ;FIX RETURN PC
010460 012637 002364            MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
010464 012637 002366            MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
010470 005037 002336            CLR     SUBRPC          ;CLEAR SUBROUTINE PC STORAGE
010474 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.
                        ;*************************************************************************
010476 010146           LODMSG: MOV     R1,-(SP)        ;SAVE R1
010500 010246                   MOV     R2,-(SP)        ;SAVE R2
010502 017601 000004            MOV     @4(SP),R1       ;GET MSG POINTER INTO R1
010506 062766 000002 000004     ADD     #2,4(SP)        ;INCR ARG POINTER
010514 017602 000004            MOV     @4(SP),R2       ;GET WORD COUNT INTO R2
010520 062766 000002 000004     ADD     #2,4(SP)        ;FIX UP RETURN PC
010526 012137 002412    6$:     MOV     (R1)+,TXWORD    ;GET NEXT MSG WORD
010532 004737 004646            JSR     PC,LDTXSI       ;LOAD A WORD INTO TX SILO
010536 005302                   DEC     R2              ;DECR COUNT
010540 001372                   BNE     6$              ;BR IF NOT DONE YET
010542 004737 004620            JSR     PC,WAIT50       ;WAIT FOR SILO TO RIPPLE
010546 012602                   MOV     (SP)+,R2        ;RESTORE R2
010550 012601                   MOV     (SP)+,R1        ;RESTORE R1
010552 000207                   RTS     PC              ;RETURN
```

```
                                        ;*****************************************************************************
                                        ;* LDBYTS - THIS SUBROUTINE LOADS THE NO. OF BYTES PASSED IN THE SECOND WORD
                                        ;*      FOLLOWING THE CALL FROM THE MSG BUFFER WHOSE ADDRESS IS IN THE FIRST
                                        ;*      WORD FOLLOWING THE CALL, INTO THE LOW BYTE OF THE TX SILO. FOR EACH
                                        ;*      BYTE LOADED, A 0 IS LOADED INTO THE HI 4 BITS OF THE TX SILO.
                                        ;*****************************************************************************
010554  010146                  LDBYTS: MOV     R1,-(SP)        ;SAVE R1
010556  010246                          MOV     R2,-(SP)        ;SAVE R2
010560  017601  000004                  MOV     @4(SP),R1       ;GET DATA POINTER INTO R1
010564  062766  000002  000004          ADD     #2,4(SP)        ;INCR ARGUMENT POINTER
010572  017602  000004                  MOV     @4(SP),R2       ;GET BYTE COUNT INTO R2
010576  062766  000002  000004          ADD     #2,4(SP)        ;FIX UP RETURN PC
010604  112137  002412          6$:     MOVB    (R1)+,TXWORD    ;GET NEXT DATA BYTE
010610  105037  002413                  CLRB    TXWORD+1        ;CLEAR HI BYTE
010614  004737  004646                  JSR     PC,LDTXSI       ;LOAD A SILO ENTRY
010620  005302                          DEC     R2              ;DECR BYTE COUNT
010622  001370                          BNE     6$              ;BR IF NOT DONE YET
010624  004737  004620                  JSR     PC,WAIT50       ;WAIT FOR SILO TO RIPPLE
010630  012602                          MOV     (SP)+,R2        ;RESTORE R2
010632  012601                          MOV     (SP)+,R1        ;RESTORE R1
010634  000207                          RTS     PC              ;RETURN


                                        ;*****************************************************************************
                                        ;* LDMSG1 - THIS SUBROUTINE LOADS THE TRANSMITTER SILO WITH MSG1, AND LOADS
                                        ;*      THE DATA CHARS INTO THE RCV MSG BUFFER (RCVBUF:), AS EXPECTED DATA
                                        ;*      FOR LATER COMPARISON.
                                        ;*****************************************************************************
010636  010146                  LDMSG1: MOV     R1,-(SP)        ;SAVE R1
010640  010246                          MOV     R2,-(SP)        ;SAVE R2
010642  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG1 INTO TX SILO (WITH ONLY 1 EOM)
010646  002670                          MSG1
010650  000010                          8.
010652  012701  002674                  MOV     #MSG1+4,R1      ;GET POINTER TO MSG1
010656  012702  002762                  MOV     #RCVBUF,R2      ;GET POINTER TO MSG BUF
010662  012122                  3$:     MOV     (R1)+,(R2)+     ;LOAD A CHAR INTO MSG BUF
010664  020127  002706                  CMP     R1,#MSG1+14.    ;SEE IF DID LAST DATA CHAR YET
010670  103774                          BLO     3$              ;BR IF NOT
010672  052762  100400  177776          BIS     #CRCCHK!RXBCC,-2(R2) ;SET EXPECTED BCC
010700  012722  000160                  MOV     #160,(R2)+      ;LOAD HI CRC BYTE
010704  012722  000034                  MOV     #034,(R2)+      ;LOAD LO CRC BYTE
010710  012602                          MOV     (SP)+,R2        ;RESTORE R2
010712  012601                          MOV     (SP)+,R1        ;RESTORE R1
010714  000207                          RTS     PC              ;RETURN


                                        ;*****************************************************************************
                                        ;* LODSIL - THIS SUBROUTINE REPEATEDLY LOADS THE DATA PASSED IN THE FIRST WORD
```

```
                        ;*      FOLLOWING THE CALL INTO THE TX SILO, FOR THE NO. OF TIMES PASSED IN THE
                        ;*      SECOND WORD FOLLOWING THE CALL.
                        ;*****************************************************************************
010716  010146          LODSIL: MOV     R1,-(SP)        ;SAVE R1
010720  017637  000002  002412          MOV     @2(SP),TXWORD   ;GET DATA
010726  062766  000002  000002          ADD     #2,2(SP)        ;INCR ARGUMENT POINTER
010734  017601  000002                  MOV     @2(SP),R1       ;GET COUNT
010740  004737  004646          3$:     JSR     PC,LDTXSI       ;LOAD WORD INTO TX SILO
010744  005301                  DEC     R1              ;DECR COUNT
010746  001374                  BNE     3$              ;BR IF NOT ALL LOADED YET
010750  004737  004620                  JSR     PC,WAIT50       ;ALLOW SILO DATA TO RIPPLE
010754  062766  000002  000002          ADD     #2,2(SP)        ;FIX UP RETURN PC
010762  012601                  MOV     (SP)+,R1        ;RESTORE R1
010764  000207                  RTS     PC              ;RETURN


                        ;*****************************************************************************
                        ;* CKLPBK - THIS SUBROUTINE DETERMINES IF THE TEST CALLING IT CAN BE RUN. THE
                        ;*      TEST PASSES THE DESIRED MODEM INTERFACE TYPE IN THE WORD FOLLOWING THE
                        ;*      CALL, AND IF THE PROPER EXTERNAL LOOPBACK HAS BEEN PROVIDED BY THE
                        ;*      OPERATOR FOR THAT INTERFACE, AND IF THE BAUD RATE IS CORRECT, A NORMAL
                        ;*      RETURN IS MADE TO RUN THE TEST. IF NOT, A RETURN IS MADE TO THE TEST,
                        ;*      AT THE ADDRESS IN RETADR (RETADR CONTAINS THE TEST EXIT ADDRESS, SO
                        ;*      THE TEST GETS SKIPPED).
                        ;*      IF BIT 15 IS SET IN THE WORD FOLLOWING THE CALL, THE TEST WILL NOT
                        ;*      BE RUN UNLESS THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED. IF THE
                        ;*      TEST IS TO BE RUN, THE SUBROUTINE RETURNS THE MODEM SELECT BITS FOR
                        ;*      AX3-15 IN MODINT. IF NECESSARY, THE SUBROUTINE WILL SET MAINT1, MAINT2,
                        ;*      OR DTR.
                        ;*      IF THE PROGRAM PASSES '0' IN THE WORD FOLLOWING THE CALL, THE SUBROUTINE
                        ;*      WILL ATTEMPT TO RUN WHICHEVER MODEM INTERFACE IS SELECTED BY CABLE,
                        ;*      SWITCH, OR TEST CONNECTOR. IF SUCCESSFUL, THE SELECTED INTERFACE WILL
                        ;*      BE PASSED BACK TO THE TEST IN MODINT.
                        ;*****************************************************************************
010766  013746  002364  CKLPBK: MOV     REGNUM,-(SP)    ;SAVE REG NO.
010772  013746  002366          MOV     AXNUM,-(SP)     ;SAVE AX BYTE NO.
010776  010246                  MOV     R2,-(SP)        ;SAVE R2
011000  016637  000006  002336          MOV     6(SP),SUBRPC
011006  162737  000004  002336          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
011014  012737  000333  002422          MOV     #I422!XYZ!V35!INTGRL!OP!TEST,MODINT ;INIT MODEM SELECT BITS
011022  032776  100000  000006          BIT     #TCCHEK,@6(SP)  ;SEE IF H3254,5 CHECK IS DESIRED
011030  001405                  BEQ     1$              ;BR IF NOT
011032  005737  002462                  TST     TSTCON          ;SEE IF H3254,5 INSTALLED
011036  001074                  BNE     7$              ;BR IF NOT, TO SKIP TEST
011040  000137  012102                  JMP     32$             ;BR TO RUN TEST
                        ;IF NO EXTERNAL LPBK, SKIP TEST
011044  023727  002462  000004  1$:     CMP     TSTCON,#4       ;SEE IF NO LPBK
011052  001466                  BEQ     7$              ;BR IF NO LPBK, TO SKIP TEST
011054  142777  000010  171356          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
011062  012737  000006  002366          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3-15
011070  004737  003576                  JSR     PC,READAX       ;READ AX3-15

                        ;*** SEE IF AN INTERFACE IS REQUESTED ***
```

```
7081
7082  011074  027627  000006  000010          CMP     @6(SP),#INTGRL  ;SEE IF INTEGRAL MODEM REQUESTED
7083  011102  001422                          BEQ     4$              ;BR IF INTGRL MODEM REQUESTED
7084  011104  027627  000006  000020          CMP     @6(SP),#V35     ;SEE IF V.35 REQUESTED
7085  011112  001512                          BEQ     10$             ;BR IF V.35 REQUESTED
7086  011114  027627  000006  000200          CMP     @6(SP),#I422    ;SEE IF 422 REQUESTED
7087  011122  001002                          BNE     2$              ;BR IF 422 NOT REQUESTED
7088  011124  000137  011414                  JMP     14$             ;422 REQUESTED
7089  011130  027627  000006  000100  2$:     CMP     @6(SP),#XYZ     ;SEE IF XYZ REQUESTED
7090  011136  001002                          BNE     3$              ;BR IF XYZ NOT REQUESTED
7091  011140  000137  011470                  JMP     17$             ;XYZ REQUESTED
7092  011144  000137  011546          3$:     JMP     21$             ;NONE REQUESTED, FIND AN INTERFACE TO TEST
7093
7094                                  ;SEE IF INTEGRAL MODEM CAN BE RUN
7095  011150  005737  002462          4$:     TST     TSTCON          ;SEE IF H3254 AND H3255 USED
7096  011154  001050                          BNE     8$              ;BR IF NOT
7097  011156  023727  002464  000004  5$:     CMP     BDRATE,#4       ;SEE IF BAUD RATE > OR = 56K
7098  011164  002405                          BLT     6$              ;BR IF BAUD RATE TOO SLOW FOR INTGRL MODM
7099  011166  042737  000010  002422          BIC     #INTGRL,MODINT  ;ASSERT INTEGRAL MODEM
7100  011174  000137  012102                  JMP     32$             ;GO TO RUN TEST
7101                                  ;PRINT 'BAUD RATE INCORRECT'
7102  011200                          6$:
7103  011200  023727  002402  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
7104  011206  001026                          BNE     40$             ;BR IF NOT, TO SKIP PRINTING
7105  011210                                  PRINTF  #FMT25
  (7) 011210  012746  013135                                                  MOV     #FMT25,-(SP)
  (6) 011214  012746  000001                                                  MOV     #1,-(SP)
  (3) 011220  010600                                                          MOV     SP,R0
  (4) 011222  104417                                                          TRAP    C$PNTF
  (4) 011224  062706  000004                                                  ADD     #4,SP
7106                                  ;PRINT 'TEST XX NOT RUN'
7107  011230                          7$:
7108  011230  023727  002402  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
7109  011236  001012                          BNE     40$             ;BR IF NOT, TO SKIP PRINTING
7110  011240                                  PRINTF  #FMT19,TSTNUM
  (8) 011240  013746  002434                                                  MOV     TSTNUM,-(SP)
  (7) 011244  012746  013104                                                  MOV     #FMT19,-(SP)
  (6) 011250  012746  000002                                                  MOV     #2,-(SP)
  (3) 011254  010600                                                          MOV     SP,R0
  (4) 011256  104417                                                          TRAP    C$PNTF
  (4) 011260  062706  000006                                                  ADD     #6,SP
7111  011264  013766  002346  000006  40$:     MOV     RETADR,6(SP)    ;SET TEST EXIT ADDRESS FOR ERRORS
7112  011272  000137  012110                  JMP     33$             ;GO TO SKIP TEST
7113  011276  032737  000010  002354  8$:     BIT     #INTGRL,RAX15   ;SEE IF INTEGRAL MODEM IS SELECTED
7114  011304  001324                          BNE     5$              ;BR IF YES, TO CHECK BAUD RATE
7115                                  ;PRINT 'MODEM INTERFACE NOT SELECTED'
7116  011306                          9$:
7117  011306  023727  002402  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
7118  011314  001363                          BNE     40$             ;BR IF NOT, TO SKIP PRINTING
7119  011316                                  PRINTF  #FMT26
  (7) 011316  012746  013165                                                  MOV     #FMT26,-(SP)
  (6) 011322  012746  000001                                                  MOV     #1,-(SP)
  (3) 011326  010600                                                          MOV     SP,R0
  (4) 011330  104417                                                          TRAP    C$PNTF
  (4) 011332  062706  000004                                                  ADD     #4,SP
7120  011336  000734                          BR      7$              ;GO TO PRINT 'TEST NOT RUN'
```

```
7121
7122                                        ;SEE IF V.35 CAN BE RUN
7123  011340  032737  000200  002354  10$:   BIT     #I422,RAX15     ;SEE IF 422 IS SELECTED
7124  011346  001357                        BNE     9$              ;BR IF YES, TO SKIP TEST
7125  011350  005737  002462                TST     TSTCON          ;SEE IF H3254 AND H3255 USED
7126  011354  001402                        BEQ     11$             ;BR IF YES
7127  011356  000137  011374                JMP     12$             ;H3254 AND H3255 NOT USED
7128  011362  042737  000020  002422  11$:   BIC     #V35,MODINT     ;ASSERT V.35
7129  011370  000137  011644                JMP     27$             ;GO SET DTR AND RUN THE TEST
7130  011374  032737  000020  002354  12$:   BIT     #V35,RAX15      ;SEE IF V.35 IS SELECTED
7131  011402  001402                        BEQ     13$             ;BR IF NOT
7132  011404  000137  011362                JMP     11$             ;GO ASSERT V.35 AND RUN TEST
7133  011410  000137  011306          13$:   JMP     9$              ;WRONG INTRFCE, GO SKIP TEST
7134
7135                                        ;SEE IF 422 CAN BE RUN
7136  011414  005737  002462          14$:   TST     TSTCON          ;SEE IF H3254 AND H3255 USED
7137  011420  001015                        BNE     16$             ;BR IF NOT
7138  011422  012737  000013  002364        MOV     #13,REGNUM      ;SET REG. NO. = 13
7139  011430  112737  000004  002352        MOVB    #MAINT2,WRIBYT
7140  011436  004737  003422                JSR     PC,WRITLU       ;SET MAINT2 FOR MANUFACTURING TEST OF RS422
7141  011442  042737  000200  002422  15$:   BIC     #I422,MODINT    ;ASSERT 422
7142  011450  000137  011644                JMP     27$             ;GO TO RUN TEST
7143  011454  032737  000200  002354  16$:   BIT     #I422,RAX15     ;SEE IF 422 IS SELECTED
7144  011462  001367                        BNE     15$             ;IF YES, GO ASSERT 422 AND RUN TEST
7145  011464  000137  011306                JMP     9$              ;WRONG INTRFCE, GO SKIP TEST
7146
7147                                        ;SEE IF XYZ CAN BE RUN
7148  011470  032737  000200  002354  17$:   BIT     #I422,RAX15     ;SEE IF 422 IS SELECTED
7149  011476  001402                        BEQ     18$             ;BR IF NOT
7150  011500  000137  011306                JMP     9$              ;WRONG INTRFCE, GO SKIP TEST
7151  011504  032737  000100  002354  18$:   BIT     #XYZ,RAX15      ;SEE IF XYZ IS SELECTED
7152  011512  001002                        BNE     19$             ;BR IF YES
7153  011514  000137  011306                JMP     9$              ;WRONG INTRFCE, GO SKIP TEST
7154  011520  023727  002464  000004  19$:   CMP     BDRATE,#4       ;SEE IF BAUD RATE < OR = 56K
7155  011526  003402                        BLE     20$             ;BR IF YES
7156  011530  000137  011200                JMP     6$              ;BAUD RATE TOO FAST FOR XYZ
7157  011534  042737  000100  002422  20$:   BIC     #XYZ,MODINT     ;ASSERT XYZ
7158  011542  000137  011644                JMP     27$             ;GO TO RUN TEST
7159
7160                                        ;*** NO INTERFACE REQUESTED - FIND ONE TO TEST ***
7161
7162  011546  032737  000010  002354  21$:   BIT     #INTGRL,RAX15   ;SEE IF INTEGRAL MODEM SELECTED
7163  011554  001402                        BEQ     22$             ;BR IF NOT
7164  011556  000137  011156                JMP     5$              ;SEE IF INTEGRAL MODEM CAN BE RUN
7165  011562  032737  000020  002354  22$:   BIT     #V35,RAX15      ;SEE IF V.35 SELECTED
7166  011570  001402                        BEQ     23$             ;BR IF NOT
7167  011572  000137  011340                JMP     10$             ;GO SEE IF V.35 CAN BE RUN
7168  011576  032737  000200  002354  23$:   BIT     #I422,RAX15     ;SEE IF 422 SELECTED
7169  011604  001402                        BEQ     24$             ;BR IF NOT
7170  011606  000137  011442                JMP     15$             ;GO ASSERT AND RUN 422
7171  011612  005737  002462          24$:   TST     TSTCON          ;SEE IF H3254 AND H3255 USED
7172  011616  001002                        BNE     25$             ;BR IF NOT
7173  011620  000137  011362                JMP     11$             ;GO ASSERT AND RUN V.35 BY DEFAULT
7174  011624  032737  000100  002354  25$:   BIT     #XYZ,RAX15      ;SEE IF XYZ SELECTED
7175  011632  001402                        BEQ     26$             ;BR IF NOT
7176  011634  000137  011520                JMP     19$             ;GO SEE IF XYZ CAN BE RUN
```

```
 7177  011640  000137  011306          26$:    JMP     9$              ;GO SKIP TEST
 7178
 7179                                  ;*** SET MAINT1 OR MAINT2 IF NEEDED, AND SET DTR ***
 7180
 7181                                  ;SET MAINT1 IF LOCAL LOOPBACK DESIRED
 7182  011644  012737  000013  002364  27$:    MOV     #13,REGNUM      ;SET REG NO. = 13
 7183  011652  023727  002462  000002          CMP     TSTCON,#2       ;SEE IF MODEM LOCAL LPBK DESIRED
 7184  011660  001032                        BNE     29$             ;BR IF NOT, TO CHK REMOTE LPBK
 7185  011662  112737  000010  002352        MOVB    #MAINT1,WRIBYT
 7186  011670  004737  003422                JSR     PC,WRITLU       ;SET MAINT1 IN REG 13
 7187  011674  012737  000017  002364        MOV     #17,REGNUM      ;SET REG NO. = 17
 7188  011702  012702  000000                MOV     #0,R2           ;INIT TIMER
 7189  011706  004737  003344          28$:  JSR     PC,READLU       ;READ REG 17
 7190  011712  132737  000004  002350        BITB    #TESTMD,REDBYT  ;SEE IF TEST MODE BIT SET IN REG 17 YET
 7191  011720  001050                        BNE     31$             ;BR IF YES, TO SET DTR
 7192  011722  005202                        INC     R2              ;INCREMENT TIMER
 7193  011724  001370                        BNE     28$             ;BR IF NO TIME-OUT YET
 7194  011726  004737  004200                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 7195                                  ;REPORT TEST MODE NOT SET BY MAINT1
 7196  011732                                ERRDF   52,EM52,ERR4
   (4) 011732  104455                                                      TRAP    C$ERDF
   (5) 011734  000064                                                      .WORD   52
   (5) 011736  014451                                                      .WORD   EM52
   (5) 011740  016356                                                      .WORD   ERR4
 7197  011742  000137  011230                JMP     7$              ;BR TO SKIP TEST
 7198                                  ;SET MAINT2 IF REMOTE LOOPBACK DESIRED
 7199  011746  023727  002462  000003  29$:    CMP     TSTCON,#3       ;SEE IF MODEM REMOTE LPBK DESIRED
 7200  011754  001032                        BNE     31$             ;BR IF NOT, TO SET DTR
 7201  011756  112737  000004  002352        MOVB    #MAINT2,WRIBYT
 7202  011764  004737  003422                JSR     PC,WRITLU       ;SET MAINT2 IN REG 13
 7203  011770  012737  000017  002364        MOV     #17,REGNUM      ;SET REG NO. = 17
 7204  011776  012702  000000                MOV     #0,R2           ;INIT TIMER
 7205  012002  004737  003344          30$:  JSR     PC,READLU       ;READ REG 17
 7206  012006  132737  000004  002350        BITB    #TESTMD,REDBYT  ;SEE IF TEST MODE BIT SET IN REG 17 YET
 7207  012014  001012                        BNE     31$             ;BR IF YES, TO SET DTR
 7208  012016  005202                        INC     R2              ;INCREMENT TIMER
 7209  012020  001370                        BNE     30$             ;BR IF NO TIME-OUT YET
 7210  012022  004737  004200                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 7211                                  ;REPORT TEST MODE NOT SET BY MAINT2
 7212  012026                                ERRDF   53,EM53,ERR4
   (4) 012026  104455                                                      TRAP    C$ERDF
   (5) 012030  000065                                                      .WORD   53
   (5) 012032  014505                                                      .WORD   EM53
   (5) 012034  016356                                                      .WORD   ERR4
 7213  012036  000137  011230                JMP     7$              ;BR TO SKIP TEST
 7214                                  ;SET DTR
 7215  012042  012737  000013  002364  31$:    MOV     #13,REGNUM      ;SET REG NO. = 13
 7216  012050  004737  003344                JSR     PC,READLU       ;READ REG 13
 7217  012054  013737  002350  002352        MOV     REDBYT,WRIBYT
 7218  012062  042737  000243  002352        BIC     #243,WRIBYT     ;CLEAR UNNEEDED BITS
 7219  012070  152737  000100  002352        BISB    #DTR,WRIBYT     ;SET DTR BIT TO BE WRITTEN
 7220  012076  004737  003422                JSR     PC,WRITLU       ;LOAD REG 13
 7221
 7222                                  ;*** BRANCH HERE TO RUN TEST ***
 7223  012102  062766  000002  000006  32$:    ADD     #2,6(SP)        ;INCREMENT RETURN ADRS
 7224
```

```
                                        ;*** BRANCH HERE TO SKIP TEST ***
012110  012602                  33$:   MOV     (SP)+,R2          ;RESTORE R2
012112  012637  002366                 MOV     (SP)+,AXNUM       ;RESTORE AX BYTE NO.
012116  012637  002364                 MOV     (SP)+,REGNUM      ;RESTORE LU REG NO.
012122  152777  000010  170310         BISB    #LULOOP,@BSEL1    ;SET LULOOP
012130  005037  002336                 CLR     SUBRPC            ;CLEAR SUBROUTINE CALL PC
012134  000207                        RTS     PC                ;RETURN
                                        ;*****************************************************************************
                                        ;* CHKABT - THIS SUBROUTINE READS AX0-16 AND CHECKS FOR RAB, REOM SET. IF
                                        ;*      EITHER IS NOT SET, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
                                        ;*      CONTAINED IN RETADR.
                                        ;*****************************************************************************
012136  013746  002366          CHKABT: MOV     AXNUM,-(SP)       ;SAVE AX BYTE NO.
012142  016637  000002  002336         MOV     2(SP),SUBRPC
012150  162737  000004  002336         SUB     #4,SUBRPC         ;GET PC OF SUBROUTINE CALL
012156  005037  002366                 CLR     AXNUM             ;SET AX0 ADDRESS
012162  004737  003576                 JSR     PC,READAX         ;READ REG AX0
012166  032737  000004  002356         BIT     #RABT,RAX16       ;CHK FOR RAB SET
012174  001007                        BNE     6$                ;BR IF RAB SET
012176  004737  004200                 JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                    ;REPORT RAB NOT SET
012202                              ERRDF   40,EM40,ERR6
(4)  012202  104455                                                       TRAP    C$ERDF
(5)  012204  000050                                                       .WORD   40
(5)  012206  014024                                                       .WORD   EM40
(5)  012210  017546                                                       .WORD   ERR6
012212  000412                        BR      8$
012214  032737  000002  002356  6$:   BIT     #REOM,RAX16       ;CHK FOR REOM SET
012222  001015                        BNE     9$                ;BR IF REOM SET
012224  004737  004200                 JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                    ;REPORT REOM NOT SET
012230                              ERRDF   31,EM31,ERR6
(4)  012230  104455                                                       TRAP    C$ERDF
(5)  012232  000037                                                       .WORD   31
(5)  012234  013647                                                       .WORD   EM31
(5)  012236  017546                                                       .WORD   ERR6
012240  011637  002366          8$:    MOV     (SP),AXNUM        ;RESTORE AX BYTE NO.
012244  013706  002332                 MOV     PSTACK,SP         ;RESTORE STACK POINTER TO BASE LEVEL
012250  013746  002346                 MOV     RETADR,-(SP)      ;FIX ERROR RETURN PC
012254  000402                        BR      16$
012256  012637  002366          9$:    MOV     (SP)+,AXNUM       ;RESTORE AX BYTE NO.
012262  000207                  16$:   RTS     PC                ;RETURN
```

```
 7270                                   .SBTTL  GLOBAL ERROR REPORT SECTION
 7271
 7272                                   ;//////////////////////////////////////////////////////////////////////
 7273                                   ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
 7274                                   ;/      THAT ARE USED IN MORE THAN ONE TEST.
 7275                                   ;//////////////////////////////////////////////////////////////////////
 7276
 7277
 7278  012264  052045  047445  022466  FMT1:   .ASCIZ  /%T%06%N/
       012272  000116
 7279  012274  047045  040445  040506  FMT2:   .ASCIZ  /%N%AFAILING REG: /
       012302  046111  047111  020107
       012310  042522  035107  000040
 7280  012316  040445  054105  042520  FMT3:   .ASCIZ  /%AEXPECTED: %03%S5%AACTUAL: %03%N/
       012324  052103  042105  020072
       012332  047445  022463  032523
       012340  040445  041501  052524
       012346  046101  020072  047445
       012354  022463  000116
 7281  012360  047045  052045  047045  FMT4:   .ASCIZ  /%N%T%N%T%N/
       012366  052045  047045     000
 7282  012373     045  031517  051445  FMT5:   .ASCIZ  /%03%S5%03%S5%03%S5%03%N/
       012400  022465  031517  051445
       012406  022465  031517  051445
       012414  022465  031517  047045
       012422     000
 7283  012423     045  032123  047445  FMT6:   .ASCIZ  /%S4%03%S5%03%S5%03%S5%03%N/
       012430  022463  032523  047445
       012436  022463  032523  047445
       012444  022463  032523  047445
       012452  022463  000116
 7284  012456  052045  047445  022462  FMT7:   .ASCIZ  /%T%02%N/
       012464  000116
 7285  012466  040445  054105  042524  FMT8:   .ASCIZ  /%AEXTENDED REG AX%01%A-%T%N/
       012474  042116  042105  051040
       012502  043505  040440  022530
       012510  030517  040445  022455
       012516  022524  000116
 7286  012522  052045  047045     000  FMT9:   .ASCIZ  /%T%N/
 7287  012527     045  050101  020103  FMT10:  .ASCIZ  /%APC OF SUBR CALL: %06%N/
       012534  043117  051440  041125
       012542  020122  040503  046114
       012550  020072  047445  022466
       012556  000116
 7288  012560  040445  042522  020107  FMT11:  .ASCIZ  /%AREG %02%A LOADED WITH: %03%N/
       012566  047445  022462  020101
       012574  047514  042101  042105
       012602  053440  052111  035110
       012610  022440  031517  047045
       012616     000
 7289  012617     045  046501  031070  FMT12:  .ASCIZ  /%AM8203 SW PACK #1 (REG 11) = %03%N/
       012624  031460  051440  020127
       012632  040520  045503  021440
       012640  020061  051050  043505
       012646  030440  024461  036440
       012654  022440  031517  047045
```

```
                012662    000
7290            012663    045   046501  031070  FMT13:  .ASCIZ  /%AM8203 SW PACK #2 (REG 15) = %03%N/
                012670  031460  051440  020127
                012676  040520  045503  021440
                012704  020062  051050  043505
                012712  030440  024465  036440
                012720  022440  031517  047045
                012726    000
7291            012727    045   046501  031070  FMT14:  .ASCIZ  /%AM8203 SW PACK #3 (REG 16) = %03%N/
                012734  031460  051440  020127
                012742  040520  045503  021440
                012750  020063  051050  043505
                012756  030440  024466  036440
                012764  022440  031517  047045
                012772    000
7292            012773    045   046501  042117  FMT15:  .ASCIZ  /%AMODEM INTERFACE REG (AX3-15) = %03%N/
                013000  046505  044440  052116
                013006  051105  040506  042503
                013014  051040  043505  024040
                013022  054101  026463  032461
                013030  020051  020075  047445
                013036  022463  000116
7293            013042  047045  040445  047506  FMT18:  .ASCIZ  /%N%AFOR DEVICE AT ADRS  %06%A ,%N/
                013050  020122  042504  044526
                013056  042503  040440  020124
                013064  042101  051522  020040
                013072  047445  022466  020101
                013100  022454  000116
7294            013104  047045  040445  042524  FMT19:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
                013112  052123  022440  031104
                013120  040445  047040  052117
                013126  051040  047125  047045
                013134    000
7295            013135    045   022516  041101  FMT25:  .ASCIZ  /%N%ABAUD RATE INCORRECT/
                013142  052501  020104  040522
                013150  042524  044440  041516
                013156  051117  042522  052103
                013164    000
7296            013165    045   022516  046501  FMT26:  .ASCIZ  /%N%AMODEM INTERFACE NOT SELECTED/
                013172  042117  046505  044440
                013200  052116  051105  040506
                013206  042503  047040  052117
                013214  051440  046105  041505
                013222  042524  000104
7297
7298
7299
7300            013226  042522  020107  047516  EM2:    .ASCIZ  /REG NOT INITIALIZED BY MST CLR/
                013234  020124  047111  052111
                013242  040511  044514  042532
                013250  020104  054502  046440
                013256  052123  041440  051114
                013264    000
7301            013265    122   043505  046440  EM3:    .ASCIZ  /REG MISCOMPARE/
                013272  051511  047503  050115
                013300  051101  000105
```

```
 7302  013304  051117  054504  047040  EM7:    .ASCIZ  /ORDY NOT SET/
       013312  052117  051440  052105
       013320     000
 7303  013321     117  042122  020131  EM8:    .ASCIZ  /ORDY NOT CLEARED/
       013326  047516  020124  046103
       013334  040505  042522  000104
 7304  013342  041517  051117  047040  EM9:    .ASCIZ  /OCOR NOT SET/
       013350  052117  051440  052105
       013356     000
 7305  013357     117  047503  020122  EM10:   .ASCIZ  /OCOR NOT CLEARED/
       013364  047516  020124  046103
       013372  040505  042522  000104
 7306  013400  040517  052103  047040  EM11:   .ASCIZ  /OACT NOT SET/
       013406  052117  051440  052105
       013414     000
 7307  013415     117  041501  020124  EM12:   .ASCIZ  /OACT NOT CLEARED/
       013422  047516  020124  046103
       013430  040505  042522  000104
 7308  013436  051111  054504  047040  EM17:   .ASCIZ  /IRDY NOT SET/
       013444  052117  051440  052105
       013452     000
 7309  013453     111  042122  020131  EM18:   .ASCIZ  /IRDY NOT CLEARED/
       013460  047516  020124  046103
       013466  040505  042522  000104
 7310  013474  041511  051111  047040  EM19:   .ASCIZ  /ICIR NOT SET/
       013502  052117  051440  052105
       013510     000
 7311  013511     111  044503  020122  EM20:   .ASCIZ  /ICIR NOT CLEARED/
       013516  047516  020124  046103
       013524  040505  042522  000104
 7312  013532  040511  052103  047040  EM21:   .ASCIZ  /IACT NOT SET/
       013540  052117  051440  052105
       013546     000
 7313  013547     111  041501  020124  EM22:   .ASCIZ  /IACT NOT CLEARED/
       013554  047516  020124  046103
       013562  040505  042522  000104
 7314  013570  051522  046517  047040  EM28:   .ASCIZ  /RSOM NOT CLEARED/
       013576  052117  041440  042514
       013604  051101  042105     000
 7315  013611     122  047523  020115  EM29:   .ASCIZ  /RSOM NOT SET/
       013616  047516  020124  042523
       013624  000124
 7316  013626  042522  046517  047040  EM30:   .ASCIZ  /REOM NOT CLEARED/
       013634  052117  041440  042514
       013642  051101  042105     000
 7317  013647     122  047505  020115  EM31:   .ASCIZ  /REOM NOT SET/
       013654  047516  020124  042523
       013662  000124
 7318  013664  041522  023526  020104  EM34:   .ASCIZ  /RCV'D DATA MISCOMPARE/
       013672  040504  040524  046440
       013700  051511  047503  050115
       013706  051101  000105
 7319  013712  041502  020103  047516  EM35:   .ASCIZ  /BCC NOT CLEARED/
       013720  020124  046103  040505
       013726  042522  000104
 7320  013732  041502  020103  047516  EM36:   .ASCIZ  /BCC NOT SET/
```

```
                013740  020124  042523  000124
 7321           013746  041105  045514  047040  EM37:   .ASCIZ  /EBLK NOT CLEARED/
                013754  052117  041440  042514
                013762  051101  042105     000
 7322           013767     105  046102  020113  EM38:   .ASCIZ  /EBLK NOT SET/
                013774  047516  020124  042523
                014002  000124
 7323           014004  040522  020102  047516  EM39:   .ASCIZ  /RAB NOT CLEARED/
                014012  020124  046103  040505
                014020  042522  000104
 7324           014024  040522  020102  047516  EM40:   .ASCIZ  /RAB NOT SET/
                014032  020124  042523  000124
 7325           014040  053117  051122  047040  EM41:   .ASCIZ  /OVRR NOT CLEARED/
                014046  052117  041440  042514
                014054  051101  042105     000
 7326           014061     117  051126  020122  EM42:   .ASCIZ  /OVRR NOT SET/
                014066  047516  020124  042523
                014074  000124
 7327           014076  053523  050040  041501  EM43:   .ASCIZ  /SW PACK #1 INCORRECT/
                014104  020113  030443  044440
                014112  041516  051117  042522
                014120  052103     000
 7328           014123     123  020127  040520  EM44:   .ASCIZ  /SW PACK #2 INCORRECT/
                014130  045503  021440  020062
                014136  047111  047503  051122
                014144  041505  000124
 7329           014150  053523  050040  041501  EM45:   .ASCIZ  /SW PACK #3 INCORRECT/
                014156  020113  031443  044440
                014164  041516  051117  042522
                014172  052103     000
 7330           014175     122  053103  051440  EM46:   .ASCIZ  /RCV SILO NOT CLEARED BY IC/
                014202  046111  020117  047516
                014210  020124  046103  040505
                014216  042522  020104  054502
                014224  044440  000103
 7331           014230  051501  042523  041115  EM47:   .ASCIZ  /ASSEMB BIT COUNT INCORRECT/
                014236  041040  052111  041440
                014244  052517  052116  044440
                014252  041516  051117  042522
                014260  052103     000
 7332           014263     117  042104  053040  EM48:   .ASCIZ  /ODD VRC PARITY BIT NOT SET/
                014270  041522  050040  051101
                014276  052111  020131  044502
                014304  020124  047516  020124
                014312  042523  000124
 7333           014316  042117  020104  051126  EM49:   .ASCIZ  /ODD VRC PARITY BIT NOT CLEARED/
                014324  020103  040520  044522
                014332  054524  041040  052111
                014340  047040  052117  041440
                014346  042514  051101  042105
                014354     000
 7334           014355     105  042526  020116  EM50:   .ASCIZ  /EVEN VRC PARITY BIT NOT SET/
                014362  051126  020103  040520
                014370  044522  054524  041040
                014376  052111  047040  052117
                014404  051440  052105     000
```

```
 7335  014411     105  042526  020116  EM51:   .ASCIZ  /EVEN VRC PARITY BIT NOT CLEARED/
       014416  051126  020103  040520
       014424  044522  054524  041040
       014432  052111  047040  052117
       014440  041440  042514  051101
       014446  042105     000
 7336  014451     124  051505  020124  EM52:   .ASCIZ  /TEST MODE NOT SET BY MAINT1/
       014456  047515  042504  047040
       014464  052117  051440  052105
       014472  041040  020131  040515
       014500  047111  030524     000
 7337  014505     124  051505  020124  EM53:   .ASCIZ  /TEST MODE NOT SET BY MAINT2/
       014512  047515  042504  047040
       014520  052117  051440  052105
       014526  041040  020131  040515
       014534  047111  031124     000
 7338  014541     124  020130  047125  EM54:   .ASCIZ  /TX UNDERRUN ERROR/
       014546  042504  051122  047125
       014554  042440  051122  051117
       014562     000
 7339  014563     104  051124  047040  EM55:   .ASCIZ  /DTR NOT SET/
       014570  052117  051440  052105
       014576     000
 7340  014577     122  047111  020107  EM56:   .ASCIZ  /RING NOT SET/
       014604  047516  020124  042523
       014612  000124
 7341  014614  047515  051104  047040  EM57:   .ASCIZ  /MODR NOT SET/
       014622  052117  051440  052105
       014630     000
 7342  014631     110  054104  047040  EM58:   .ASCIZ  /HDX NOT SET/
       014636  052117  051440  052105
       014644     000
 7343  014645     123  041124  020131  EM59:   .ASCIZ  /STBY NOT SET/
       014652  047516  020124  042523
       014660  000124
 7344  014662  052122  020123  047516  EM60:   .ASCIZ  /RTS NOT SET/
       014670  020124  042523  000124
 7345  014676  051503  047040  052117  EM61:   .ASCIZ  /CS NOT SET/
       014704  051440  052105     000
 7346  014711     103  051101  020122  EM62:   .ASCIZ  /CARR NOT SET/
       014716  047516  020124  042523
       014724  000124
 7347  014726  044523  050507  047040  EM63:   .ASCIZ  /SIGQ NOT SET/
       014734  052117  051440  052105
       014742     000
 7348  014743     123  043511  020122  EM64:   .ASCIZ  /SIGR NOT SET/
       014750  047516  020124  042523
       014756  000124
 7349  014760  052122  020123  047516  EM65:   .ASCIZ  /RTS NOT CLEARED/
       014766  020124  046103  040505
       014774  042522  000104
 7350  015000  040503  051122  047040  EM66:   .ASCIZ  /CARR NOT CLEARED/
       015006  052117  041440  042514
       015014  051101  042105     000
 7351
 7352
```

```
 7353
 7354  015021     111  041116  051525  DH1:    .ASCIZ  &INBUS/OUTBUS REG &
       015026  047457  052125  052502
       015034  020123  042522  020107
       015042     000
 7355  015043     114  047111  020105  DH2:    .ASCIZ  /LINE UNIT INBUS REGS :/
       015050  047125  052111  044440
       015056  041116  051525  051040
       015064  043505  020123  000072
 7356  015072  042522  030507  020060  DH3:    .ASCIZ  /REG10   REG11   REG12   REG13/
       015100  020040  042522  030507
       015106  020061  020040  042522
       015114  030507  020062  020040
       015122  042522  030507  000063
 7357  015130  020040  020040  042522  DH4:    .ASCIZ  /    REG14   REG15   REG16   REG17/
       015136  030507  020064  020040
       015144  042522  030507  020065
       015152  020040  042522  030507
       015160  020066  020040  042522
       015166  030507  000067
 7358  015172  032461     000          DH5:    .ASCIZ  /15/
 7359  015175     061  000066          DH6:    .ASCIZ  /16/
 7360  015200  044514  042516  052440  DH7:    .ASCIZ  /LINE UNIT EXTENDED REGS :/
       015206  044516  020124  054105
       015214  042524  042116  042105
       015222  051040  043505  020123
       015230  000072
 7361  015232  054101  026460  032461  DH8:    .ASCIZ  /AX0-15  AX0-16  AX1-15  AX1-16/
       015240  020040  054101  026460
       015246  033061  020040  054101
       015254  026461  032461  020040
       015262  054101  026461  033061
       015270     000
 7362  015271     040  020040  040440  DH9:    .ASCIZ  /    AX2-15  AX2-16  AX3-15  AX3-16/
       015276  031130  030455  020065
       015304  040440  031130  030455
       015312  020066  040440  031530
       015320  030455  020065  040440
       015326  031530  030455  000066
 7363
 7364                                   .EVEN
 7365
 7366
 7367
 7368
 7369
 7370  015334                         BGNMSG  ERR1
  (3)  015334
 7371  015334                         PRINTB  #FMT1,#ADDRES,MPCSR
  (9)  015334  013746  002436
  (8)  015340  012746  036452
  (7)  015344  012746  012264
  (6)  015350  012746  000003
  (3)  015354  010600
  (4)  015356  104414
  (4)  015360  062706  000010

ERR1::
        MOV     MPCSR,-(SP)
        MOV     #ADDRES,-(SP)
        MOV     #FMT1,-(SP)
        MOV     #3,-(SP)
        MOV     SP,R0
        TRAP    C$PNTB
        ADD     #10,SP
```

```
 7372  015364                          ENDMSG
  (3)  015364
  (3)  015364  104423                                              L10002:
                                                                           TRAP    C$MSG
 7373
 7374
 7375
 7376  015366                          BGNMSG  ERR2
  (3)  015366                                                      ERR2::
 7377  015366                          PRINTB  #FMT1,#ADDRES,MPCSR
  (9)  015366  013746  002436                                              MOV     MPCSR,-(SP)
  (8)  015372  012746  036452                                              MOV     #ADDRES,-(SP)
  (7)  015376  012746  012264                                              MOV     #FMT1,-(SP)
  (6)  015402  012746  000003                                              MOV     #3,-(SP)
  (3)  015406  010600                                                      MOV     SP,R0
  (4)  015410  104414                                                      TRAP    C$PNTB
  (4)  015412  062706  000010                                              ADD     #10,SP
 7378  015416                          PRINTB  #FMT2
  (7)  015416  012746  012274                                              MOV     #FMT2,-(SP)
  (6)  015422  012746  000001                                              MOV     #1,-(SP)
  (3)  015426  010600                                                      MOV     SP,R0
  (4)  015430  104414                                                      TRAP    C$PNTB
  (4)  015432  062706  000004                                              ADD     #4,SP
 7379  015436                          PRINTB  #FMT7,#DH1,REGNUM
  (9)  015436  013746  002364                                              MOV     REGNUM,-(SP)
  (8)  015442  012746  015021                                              MOV     #DH1,-(SP)
  (7)  015446  012746  012456                                              MOV     #FMT7,-(SP)
  (6)  015452  012746  000003                                              MOV     #3,-(SP)
  (3)  015456  010600                                                      MOV     SP,R0
  (4)  015460  104414                                                      TRAP    C$PNTB
  (4)  015462  062706  000010                                              ADD     #10,SP
 7380  015466                          PRINTB  #FMT3,GOODAT,BADDAT
  (9)  015466  013746  002372                                              MOV     BADDAT,-(SP)
  (8)  015472  013746  002370                                              MOV     GOODAT,-(SP)
  (7)  015476  012746  012316                                              MOV     #FMT3,-(SP)
  (6)  015502  012746  000003                                              MOV     #3,-(SP)
  (3)  015506  010600                                                      MOV     SP,R0
  (4)  015510  104414                                                      TRAP    C$PNTB
  (4)  015512  062706  000010                                              ADD     #10,SP
 7381  015516                          PRINTX  #FMT4,#DH2,#DH3
  (9)  015516  012746  015072                                              MOV     #DH3,-(SP)
  (8)  015522  012746  015043                                              MOV     #DH2,-(SP)
  (7)  015526  012746  012360                                              MOV     #FMT4,-(SP)
  (6)  015532  012746  000003                                              MOV     #3,-(SP)
  (3)  015536  010600                                                      MOV     SP,R0
  (4)  015540  104415                                                      TRAP    C$PNTX
  (4)  015542  062706  000010                                              ADD     #10,SP
 7382  015546                          PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 (11)  015546  013746  002274                                              MOV     LUR13,-(SP)
 (10)  015552  013746  002272                                              MOV     LUR12,-(SP)
  (9)  015556  013746  002270                                              MOV     LUR11,-(SP)
  (8)  015562  013746  002266                                              MOV     LUR10,-(SP)
  (7)  015566  012746  012373                                              MOV     #FMT5,-(SP)
  (6)  015572  012746  000005                                              MOV     #5,-(SP)
  (3)  015576  010600                                                      MOV     SP,R0
  (4)  015600  104415                                                      TRAP    C$PNTX
  (4)  015602  062706  000014                                              ADD     #14,SP
```

```
 7383  015606                          PRINTX  #FMT9,#DH4
  (8)  015606  012746  015130                                  MOV     #DH4,-(SP)
  (7)  015612  012746  012522                                  MOV     #FMT9,-(SP)
  (6)  015616  012746  000002                                  MOV     #2,-(SP)
  (3)  015622  010600                                          MOV     SP,R0
  (4)  015624  104415                                          TRAP    C$PNTX
  (4)  015626  062706  000006                                  ADD     #6,SP
 7384  015632                          PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
  (11) 015632  013746  002304                                  MOV     LUR17,-(SP)
  (10) 015636  013746  002302                                  MOV     LUR16,-(SP)
  (9)  015642  013746  002300                                  MOV     LUR15,-(SP)
  (8)  015646  013746  002276                                  MOV     LUR14,-(SP)
  (7)  015652  012746  012423                                  MOV     #FMT6,-(SP)
  (6)  015656  012746  000005                                  MOV     #5,-(SP)
  (3)  015662  010600                                          MOV     SP,R0
  (4)  015664  104415                                          TRAP    C$PNTX
  (4)  015666  062706  000014                                  ADD     #14,SP
 7385  015672                          ENDMSG
  (3)  015672                                          L10003:
  (3)  015672  104423                                          TRAP    C$MSG
 7386
 7387
 7388
 7389
 7390
 7391  015674                          BGNMSG  ERR3
  (3)  015674                                          ERR3::
 7392  015674                          PRINTB  #FMT1,#ADDRES,MPCSR
  (9)  015674  013746  002436                                  MOV     MPCSR,-(SP)
  (8)  015700  012746  036452                                  MOV     #ADDRES,-(SP)
  (7)  015704  012746  012264                                  MOV     #FMT1,-(SP)
  (6)  015710  012746  000003                                  MOV     #3,-(SP)
  (3)  015714  010600                                          MOV     SP,R0
  (4)  015716  104414                                          TRAP    C$PNTB
  (4)  015720  062706  000010                                  ADD     #10,SP
 7393  015724                          PRINTB  #FMT2
  (7)  015724  012746  012274                                  MOV     #FMT2,-(SP)
  (6)  015730  012746  000001                                  MOV     #1,-(SP)
  (3)  015734  010600                                          MOV     SP,R0
  (4)  015736  104414                                          TRAP    C$PNTB
  (4)  015740  062706  000004                                  ADD     #4,SP
 7394  015744                          PRINTB  #FMT8,TMP1,TMP0
  (9)  015744  013746  002516                                  MOV     TMP0,-(SP)
  (8)  015750  013746  002520                                  MOV     TMP1,-(SP)
  (7)  015754  012746  012466                                  MOV     #FMT8,-(SP)
  (6)  015760  012746  000003                                  MOV     #3,-(SP)
  (3)  015764  010600                                          MOV     SP,R0
  (4)  015766  104414                                          TRAP    C$PNTB
  (4)  015770  062706  000010                                  ADD     #10,SP
 7395  015774                          PRINTB  #FMT3,GOODAT,BADDAT
  (9)  015774  013746  002372                                  MOV     BADDAT,-(SP)
  (8)  016000  013746  002370                                  MOV     GOODAT,-(SP)
  (7)  016004  012746  012316                                  MOV     #FMT3,-(SP)
  (6)  016010  012746  000003                                  MOV     #3,-(SP)
  (3)  016014  010600                                          MOV     SP,R0
  (4)  016016  104414                                          TRAP    C$PNTB
```

```
  (4)  016020  062706  000010                                    ADD     #10,SP
7396   016024                    PRINTX  #FMT4,#DH2,#DH3
  (9)  016024  012746  015072                                    MOV     #DH3,-(SP)
  (8)  016030  012746  015043                                    MOV     #DH2,-(SP)
  (7)  016034  012746  012360                                    MOV     #FMT4,-(SP)
  (6)  016040  012746  000003                                    MOV     #3,-(SP)
  (3)  016044  010600                                            MOV     SP,R0
  (4)  016046  104415                                            TRAP    C$PNTX
  (4)  016050  062706  000010                                    ADD     #10,SP
7397   016054                    PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
  (11) 016054  013746  002274                                    MOV     LUR13,-(SP)
  (10) 016060  013746  002272                                    MOV     LUR12,-(SP)
  (9)  016064  013746  002270                                    MOV     LUR11,-(SP)
  (8)  016070  013746  002266                                    MOV     LUR10,-(SP)
  (7)  016074  012746  012373                                    MOV     #FMT5,-(SP)
  (6)  016100  012746  000005                                    MOV     #5,-(SP)
  (3)  016104  010600                                            MOV     SP,R0
  (4)  016106  104415                                            TRAP    C$PNTX
  (4)  016110  062706  000014                                    ADD     #14,SP
7398   016114                    PRINTX  #FMT9,#DH4
  (8)  016114  012746  015130                                    MOV     #DH4,-(SP)
  (7)  016120  012746  012522                                    MOV     #FMT9,-(SP)
  (6)  016124  012746  000002                                    MOV     #2,-(SP)
  (3)  016130  010600                                            MOV     SP,R0
  (4)  016132  104415                                            TRAP    C$PNTX
  (4)  016134  062706  000006                                    ADD     #6,SP
7399   016140                    PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
  (11) 016140  013746  002304                                    MOV     LUR17,-(SP)
  (10) 016144  013746  002302                                    MOV     LUR16,-(SP)
  (9)  016150  013746  002300                                    MOV     LUR15,-(SP)
  (8)  016154  013746  002276                                    MOV     LUR14,-(SP)
  (7)  016160  012746  012423                                    MOV     #FMT6,-(SP)
  (6)  016164  012746  000005                                    MOV     #5,-(SP)
  (3)  016170  010600                                            MOV     SP,R0
  (4)  016172  104415                                            TRAP    C$PNTX
  (4)  016174  062706  000014                                    ADD     #14,SP
7400   016200                    PRINTX  #FMT4,#DH7,#DH8
  (9)  016200  012746  015232                                    MOV     #DH8,-(SP)
  (8)  016204  012746  015200                                    MOV     #DH7,-(SP)
  (7)  016210  012746  012360                                    MOV     #FMT4,-(SP)
  (6)  016214  012746  000003                                    MOV     #3,-(SP)
  (3)  016220  010600                                            MOV     SP,R0
  (4)  016222  104415                                            TRAP    C$PNTX
  (4)  016224  062706  000010                                    ADD     #10,SP
7401   016230                    PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
  (11) 016230  013746  002314                                    MOV     AX1.16,-(SP)
  (10) 016234  013746  002312                                    MOV     AX1.15,-(SP)
  (9)  016240  013746  002310                                    MOV     AX0.16,-(SP)
  (8)  016244  013746  002306                                    MOV     AX0.15,-(SP)
  (7)  016250  012746  012373                                    MOV     #FMT5,-(SP)
  (6)  016254  012746  000005                                    MOV     #5,-(SP)
  (3)  016260  010600                                            MOV     SP,R0
  (4)  016262  104415                                            TRAP    C$PNTX
  (4)  016264  062706  000014                                    ADD     #14,SP
7402   016270                    PRINTX  #FMT9,#DH9
  (8)  016270  012746  015271                                    MOV     #DH9,-(SP)
```

```
   (7)  016274  012746  012522                                                  MOV     #FMT9,-(SP)
   (6)  016300  012746  000002                                                  MOV     #2,-(SP)
   (3)  016304  010600                                                          MOV     SP,R0
   (4)  016306  104415                                                          TRAP    C$PNTX
   (4)  016310  062706  000006                                                  ADD     #6,SP
 7403   016314                       PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
  (11)  016314  013746  002324                                                  MOV     AX3.16,-(SP)
  (10)  016320  013746  002322                                                  MOV     AX3.15,-(SP)
   (9)  016324  013746  002320                                                  MOV     AX2.16,-(SP)
   (8)  016330  013746  002316                                                  MOV     AX2.15,-(SP)
   (7)  016334  012746  012423                                                  MOV     #FMT6,-(SP)
   (6)  016340  012746  000005                                                  MOV     #5,-(SP)
   (3)  016344  010600                                                          MOV     SP,R0
   (4)  016346  104415                                                          TRAP    C$PNTX
   (4)  016350  062706  000014                                                  ADD     #14,SP
 7404   016354                       ENDMSG
   (3)  016354                                                          L10004:
   (3)  016354  104423                                                          TRAP    C$MSG
 7405
 7406
 7407
 7408
 7409
 7410   016356                       BGNMSG  ERR4
   (3)  016356                                                          ERR4::
 7411   016356                       PRINTB  #FMT10,SUBRPC
   (8)  016356  013746  002336                                                  MOV     SUBRPC,-(SP)
   (7)  016362  012746  012527                                                  MOV     #FMT10,-(SP)
   (6)  016366  012746  000002                                                  MOV     #2,-(SP)
   (3)  016372  010600                                                          MOV     SP,R0
   (4)  016374  104414                                                          TRAP    C$PNTB
   (4)  016376  062706  000006                                                  ADD     #6,SP
 7412   016402                       PRINTB  #FMT1,#ADDRES,MPCSR
   (9)  016402  013746  002436                                                  MOV     MPCSR,-(SP)
   (8)  016406  012746  036452                                                  MOV     #ADDRES,-(SP)
   (7)  016412  012746  012264                                                  MOV     #FMT1,-(SP)
   (6)  016416  012746  000003                                                  MOV     #3,-(SP)
   (3)  016422  010600                                                          MOV     SP,R0
   (4)  016424  104414                                                          TRAP    C$PNTB
   (4)  016426  062706  000010                                                  ADD     #10,SP
 7413   016432                       PRINTB  #FMT2
   (7)  016432  012746  012274                                                  MOV     #FMT2,-(SP)
   (6)  016436  012746  000001                                                  MOV     #1,-(SP)
   (3)  016442  010600                                                          MOV     SP,R0
   (4)  016444  104414                                                          TRAP    C$PNTB
   (4)  016446  062706  000004                                                  ADD     #4,SP
 7414   016452                       PRINTB  #FMT7,#DH1,REGNUM
   (9)  016452  013746  002364                                                  MOV     REGNUM,-(SP)
   (8)  016456  012746  015021                                                  MOV     #DH1,-(SP)
   (7)  016462  012746  012456                                                  MOV     #FMT7,-(SP)
   (6)  016466  012746  000003                                                  MOV     #3,-(SP)
   (3)  016472  010600                                                          MOV     SP,R0
   (4)  016474  104414                                                          TRAP    C$PNTB
   (4)  016476  062706  000010                                                  ADD     #10,SP
 7415   016502                       PRINTX  #FMT4,#DH2,#DH3
   (9)  016502  012746  015072                                                  MOV     #DH3,-(SP)
```

```
  (8)  016506  012746  015043                                          MOV     #DH2,-(SP)
  (7)  016512  012746  012360                                          MOV     #FMT4,-(SP)
  (6)  016516  012746  000003                                          MOV     #3,-(SP)
  (3)  016522  010600                                                  MOV     SP,R0
  (4)  016524  104415                                                  TRAP    C$PNTX
  (4)  016526  062706  000010                                          ADD     #10,SP
7416   016532                    PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
  (11) 016532  013746  002274                                          MOV     LUR13,-(SP)
  (10) 016536  013746  002272                                          MOV     LUR12,-(SP)
  (9)  016542  013746  002270                                          MOV     LUR11,-(SP)
  (8)  016546  013746  002266                                          MOV     LUR10,-(SP)
  (7)  016552  012746  012373                                          MOV     #FMT5,-(SP)
  (6)  016556  012746  000005                                          MOV     #5,-(SP)
  (3)  016562  010600                                                  MOV     SP,R0
  (4)  016564  104415                                                  TRAP    C$PNTX
  (4)  016566  062706  000014                                          ADD     #14,SP
7417   016572                    PRINTX  #FMT9,#DH4
  (8)  016572  012746  015130                                          MOV     #DH4,-(SP)
  (7)  016576  012746  012522                                          MOV     #FMT9,-(SP)
  (6)  016602  012746  000002                                          MOV     #2,-(SP)
  (3)  016606  010600                                                  MOV     SP,R0
  (4)  016610  104415                                                  TRAP    C$PNTX
  (4)  016612  062706  000006                                          ADD     #6,SP
7418   016616                    PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
  (11) 016616  013746  002304                                          MOV     LUR17,-(SP)
  (10) 016622  013746  002302                                          MOV     LUR16,-(SP)
  (9)  016626  013746  002300                                          MOV     LUR15,-(SP)
  (8)  016632  013746  002276                                          MOV     LUR14,-(SP)
  (7)  016636  012746  012423                                          MOV     #FMT6,-(SP)
  (6)  016642  012746  000005                                          MOV     #5,-(SP)
  (3)  016646  010600                                                  MOV     SP,R0
  (4)  016650  104415                                                  TRAP    C$PNTX
  (4)  016652  062706  000014                                          ADD     #14,SP
7419   016656                    PRINTX  #FMT4,#DH7,#DH8
  (9)  016656  012746  015232                                          MOV     #DH8,-(SP)
  (8)  016662  012746  015200                                          MOV     #DH7,-(SP)
  (7)  016666  012746  012360                                          MOV     #FMT4,-(SP)
  (6)  016672  012746  000003                                          MOV     #3,-(SP)
  (3)  016676  010600                                                  MOV     SP,R0
  (4)  016700  104415                                                  TRAP    C$PNTX
  (4)  016702  062706  000010                                          ADD     #10,SP
7420   016706                    PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
  (11) 016706  013746  002314                                          MOV     AX1.16,-(SP)
  (10) 016712  013746  002312                                          MOV     AX1.15,-(SP)
  (9)  016716  013746  002310                                          MOV     AX0.16,-(SP)
  (8)  016722  013746  002306                                          MOV     AX0.15,-(SP)
  (7)  016726  012746  012373                                          MOV     #FMT5,-(SP)
  (6)  016732  012746  000005                                          MOV     #5,-(SP)
  (3)  016736  010600                                                  MOV     SP,R0
  (4)  016740  104415                                                  TRAP    C$PNTX
  (4)  016742  062706  000014                                          ADD     #14,SP
7421   016746                    PRINTX  #FMT9,#DH9
  (8)  016746  012746  015271                                          MOV     #DH9,-(SP)
  (7)  016752  012746  012522                                          MOV     #FMT9,-(SP)
  (6)  016756  012746  000002                                          MOV     #2,-(SP)
  (3)  016762  010600                                                  MOV     SP,R0
```

```
   (4)  016764  104415
   (4)  016766  062706  000006
 7422   016772                    PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
  (11)  016772  013746  002324
  (10)  016776  013746  002322
   (9)  017002  013746  002320
   (8)  017006  013746  002316
   (7)  017012  012746  012423
   (6)  017016  012746  000005
   (3)  017022  010600
   (4)  017024  104415
   (4)  017026  062706  000014
 7423   017032                    ENDMSG
   (3)  017032
   (3)  017032  104423
 7424
 7425
 7426
 7427
 7428
 7429   017034                    BGNMSG  ERR5
   (3)  017034
 7430   017034                    PRINTB  #FMT1,#ADDRES,MPCSR
   (9)  017034  013746  002436
   (8)  017040  012746  036452
   (7)  017044  012746  012264
   (6)  017050  012746  000003
   (3)  017054  010600
   (4)  017056  104414
   (4)  017060  062706  000010
 7431   017064                    PRINTB  #FMT11,REGNUM,LOADAT
   (9)  017064  013746  002374
   (8)  017070  013746  002364
   (7)  017074  012746  012560
   (6)  017100  012746  000003
   (3)  017104  010600
   (4)  017106  104414
   (4)  017110  062706  000010
 7432   017114                    PRINTB  #FMT2
   (7)  017114  012746  012274
   (6)  017120  012746  000001
   (3)  017124  010600
   (4)  017126  104414
   (4)  017130  062706  000004
 7433   017134                    PRINTB  #FMT8,TMP1,TMP0
   (9)  017134  013746  002516
   (8)  017140  013746  002520
   (7)  017144  012746  012466
   (6)  017150  012746  000003
   (3)  017154  010600
   (4)  017156  104414
   (4)  017160  062706  000010
 7434   017164                    PRINTB  #FMT3,GOODAT,BADDAT
   (9)  017164  013746  002372
   (8)  017170  013746  002370
   (7)  017174  012746  012316

                                        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$PNTX
                                        ADD     #14,SP
                                L10005:
                                        TRAP    C$MSG

                                ERR5::

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

                                        MOV     LOADAT,-(SP)
                                        MOV     REGNUM,-(SP)
                                        MOV     #FMT11,-(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)
                                        MOV     SP,R0
                                        TRAP    C$PNTB
                                        ADD     #10,SP

                                        MOV     BADDAT,-(SP)
                                        MOV     GOODAT,-(SP)
                                        MOV     #FMT3,-(SP)
```

```
   (6)  017200  012746  000003                                    MOV     #3,-(SP)
   (3)  017204  010600                                            MOV     SP,R0
   (4)  017206  104414                                            TRAP    C$PNTB
   (4)  017210  062706  000010                                    ADD     #10,SP
 7435   017214                    PRINTX  #FMT4,#DH2,#DH3
   (9)  017214  012746  015072                                    MOV     #DH3,-(SP)
   (8)  017220  012746  015043                                    MOV     #DH2,-(SP)
   (7)  017224  012746  012360                                    MOV     #FMT4,-(SP)
   (6)  017230  012746  000003                                    MOV     #3,-(SP)
   (3)  017234  010600                                            MOV     SP,R0
   (4)  017236  104415                                            TRAP    C$PNTX
   (4)  017240  062706  000010                                    ADD     #10,SP
 7436   017244                    PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
  (11)  017244  013746  002274                                    MOV     LUR13,-(SP)
  (10)  017250  013746  002272                                    MOV     LUR12,-(SP)
   (9)  017254  013746  002270                                    MOV     LUR11,-(SP)
   (8)  017260  013746  002266                                    MOV     LUR10,-(SP)
   (7)  017264  012746  012373                                    MOV     #FMT5,-(SP)
   (6)  017270  012746  000005                                    MOV     #5,-(SP)
   (3)  017274  010600                                            MOV     SP,R0
   (4)  017276  104415                                            TRAP    C$PNTX
   (4)  017300  062706  000014                                    ADD     #14,SP
 7437   017304                    PRINTX  #FMT9,#DH4
   (8)  017304  012746  015130                                    MOV     #DH4,-(SP)
   (7)  017310  012746  012522                                    MOV     #FMT9,-(SP)
   (6)  017314  012746  000002                                    MOV     #2,-(SP)
   (3)  017320  010600                                            MOV     SP,R0
   (4)  017322  104415                                            TRAP    C$PNTX
   (4)  017324  062706  000006                                    ADD     #6,SP
 7438   017330                    PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
  (11)  017330  013746  002304                                    MOV     LUR17,-(SP)
  (10)  017334  013746  002302                                    MOV     LUR16,-(SP)
   (9)  017340  013746  002300                                    MOV     LUR15,-(SP)
   (8)  017344  013746  002276                                    MOV     LUR14,-(SP)
   (7)  017350  012746  012423                                    MOV     #FMT6,-(SP)
   (6)  017354  012746  000005                                    MOV     #5,-(SP)
   (3)  017360  010600                                            MOV     SP,R0
   (4)  017362  104415                                            TRAP    C$PNTX
   (4)  017364  062706  000014                                    ADD     #14,SP
 7439   017370                    PRINTX  #FMT4,#DH7,#DH8
   (9)  017370  012746  015232                                    MOV     #DH8,-(SP)
   (8)  017374  012746  015200                                    MOV     #DH7,-(SP)
   (7)  017400  012746  012360                                    MOV     #FMT4,-(SP)
   (6)  017404  012746  000003                                    MOV     #3,-(SP)
   (3)  017410  010600                                            MOV     SP,R0
   (4)  017412  104415                                            TRAP    C$PNTX
   (4)  017414  062706  000010                                    ADD     #10,SP
 7440   017420                    PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
  (11)  017420  013746  002314                                    MOV     AX1.16,-(SP)
  (10)  017424  013746  002312                                    MOV     AX1.15,-(SP)
   (9)  017430  013746  002310                                    MOV     AX0.16,-(SP)
   (8)  017434  013746  002306                                    MOV     AX0.15,-(SP)
   (7)  017440  012746  012373                                    MOV     #FMT5,-(SP)
   (6)  017444  012746  000005                                    MOV     #5,-(SP)
   (3)  017450  010600                                            MOV     SP,R0
   (4)  017452  104415                                            TRAP    C$PNTX
```

```
  (4)  017454  062706  000014                                                    ADD     #14,SP
7441   017460                    PRINTX  #FMT9,#DH9
  (8)  017460  012746  015271                                                    MOV     #DH9,-(SP)
  (7)  017464  012746  012522                                                    MOV     #FMT9,-(SP)
  (6)  017470  012746  000002                                                    MOV     #2,-(SP)
  (3)  017474  010600                                                            MOV     SP,R0
  (4)  017476  104415                                                            TRAP    C$PNTX
  (4)  017500  062706  000006                                                    ADD     #6,SP
7442   017504                    PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
 (11)  017504  013746  002324                                                    MOV     AX3.16,-(SP)
 (10)  017510  013746  002322                                                    MOV     AX3.15,-(SP)
  (9)  017514  013746  002320                                                    MOV     AX2.16,-(SP)
  (8)  017520  013746  002316                                                    MOV     AX2.15,-(SP)
  (7)  017524  012746  012423                                                    MOV     #FMT6,-(SP)
  (6)  017530  012746  000005                                                    MOV     #5,-(SP)
  (3)  017534  010600                                                            MOV     SP,R0
  (4)  017536  104415                                                            TRAP    C$PNTX
  (4)  017540  062706  000014                                                    ADD     #14,SP
7443   017544                    ENDMSG
  (3)  017544                                                           L10006:
  (3)  017544  104423                                                            TRAP    C$MSG
7444
7445
7446
7447
7448
7449   017546                    BGNMSG  ERR6
  (3)  017546                                                           ERR6::
7450   017546                    PRINTB  #FMT10,SUBRPC
  (8)  017546  013746  002336                                                    MOV     SUBRPC,-(SP)
  (7)  017552  012746  012527                                                    MOV     #FMT10,-(SP)
  (6)  017556  012746  000002                                                    MOV     #2,-(SP)
  (3)  017562  010600                                                            MOV     SP,R0
  (4)  017564  104414                                                            TRAP    C$PNTB
  (4)  017566  062706  000006                                                    ADD     #6,SP
7451   017572                    PRINTB  #FMT1,#ADDRES,MPCSR
  (9)  017572  013746  002436                                                    MOV     MPCSR,-(SP)
  (8)  017576  012746  036452                                                    MOV     #ADDRES,-(SP)
  (7)  017602  012746  012264                                                    MOV     #FMT1,-(SP)
  (6)  017606  012746  000003                                                    MOV     #3,-(SP)
  (3)  017612  010600                                                            MOV     SP,R0
  (4)  017614  104414                                                            TRAP    C$PNTB
  (4)  017616  062706  000010                                                    ADD     #10,SP
7452   017622                    PRINTB  #FMT2
  (7)  017622  012746  012274                                                    MOV     #FMT2,-(SP)
  (6)  017626  012746  000001                                                    MOV     #1,-(SP)
  (3)  017632  010600                                                            MOV     SP,R0
  (4)  017634  104414                                                            TRAP    C$PNTB
  (4)  017636  062706  000004                                                    ADD     #4,SP
7453   017642                    PRINTB  #FMT8,TMP1,TMP0
  (9)  017642  013746  002516                                                    MOV     TMP0,-(SP)
  (8)  017646  013746  002520                                                    MOV     TMP1,-(SP)
  (7)  017652  012746  012466                                                    MOV     #FMT8,-(SP)
  (6)  017656  012746  000003                                                    MOV     #3,-(SP)
  (3)  017662  010600                                                            MOV     SP,R0
  (4)  017664  104414                                                            TRAP    C$PNTB
```

```
  (4)  017666  062706  000010                                    ADD     #10,SP
7454   017672
                        PRINTX  #FMT4,#DH2,#DH3
  (9)  017672  012746  015072                                    MOV     #DH3,-(SP)
  (8)  017676  012746  015043                                    MOV     #DH2,-(SP)
  (7)  017702  012746  012360                                    MOV     #FMT4,-(SP)
  (6)  017706  012746  000003                                    MOV     #3,-(SP)
  (3)  017712  010600                                            MOV     SP,R0
  (4)  017714  104415                                            TRAP    C$PNTX
  (4)  017716  062706  000010                                    ADD     #10,SP
7455   017722
                        PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 (11)  017722  013746  002274                                    MOV     LUR13,-(SP)
 (10)  017726  013746  002272                                    MOV     LUR12,-(SP)
  (9)  017732  013746  002270                                    MOV     LUR11,-(SP)
  (8)  017736  013746  002266                                    MOV     LUR10,-(SP)
  (7)  017742  012746  012373                                    MOV     #FMT5,-(SP)
  (6)  017746  012746  000005                                    MOV     #5,-(SP)
  (3)  017752  010600                                            MOV     SP,R0
  (4)  017754  104415                                            TRAP    C$PNTX
  (4)  017756  062706  000014                                    ADD     #14,SP
7456   017762
                        PRINTX  #FMT9,#DH4
  (8)  017762  012746  015130                                    MOV     #DH4,-(SP)
  (7)  017766  012746  012522                                    MOV     #FMT9,-(SP)
  (6)  017772  012746  000002                                    MOV     #2,-(SP)
  (3)  017776  010600                                            MOV     SP,R0
  (4)  020000  104415                                            TRAP    C$PNTX
  (4)  020002  062706  000006                                    ADD     #6,SP
7457   020006
                        PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
 (11)  020006  013746  002304                                    MOV     LUR17,-(SP)
 (10)  020012  013746  002302                                    MOV     LUR16,-(SP)
  (9)  020016  013746  002300                                    MOV     LUR15,-(SP)
  (8)  020022  013746  002276                                    MOV     LUR14,-(SP)
  (7)  020026  012746  012423                                    MOV     #FMT6,-(SP)
  (6)  020032  012746  000005                                    MOV     #5,-(SP)
  (3)  020036  010600                                            MOV     SP,R0
  (4)  020040  104415                                            TRAP    C$PNTX
  (4)  020042  062706  000014                                    ADD     #14,SP
7458   020046
                        PRINTX  #FMT4,#DH7,#DH8
  (9)  020046  012746  015232                                    MOV     #DH8,-(SP)
  (8)  020052  012746  015200                                    MOV     #DH7,-(SP)
  (7)  020056  012746  012360                                    MOV     #FMT4,-(SP)
  (6)  020062  012746  000003                                    MOV     #3,-(SP)
  (3)  020066  010600                                            MOV     SP,R0
  (4)  020070  104415                                            TRAP    C$PNTX
  (4)  020072  062706  000010                                    ADD     #10,SP
7459   020076
                        PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
 (11)  020076  013746  002314                                    MOV     AX1.16,-(SP)
 (10)  020102  013746  002312                                    MOV     AX1.15,-(SP)
  (9)  020106  013746  002310                                    MOV     AX0.16,-(SP)
  (8)  020112  013746  002306                                    MOV     AX0.15,-(SP)
  (7)  020116  012746  012373                                    MOV     #FMT5,-(SP)
  (6)  020122  012746  000005                                    MOV     #5,-(SP)
  (3)  020126  010600                                            MOV     SP,R0
  (4)  020130  104415                                            TRAP    C$PNTX
  (4)  020132  062706  000014                                    ADD     #14,SP
7460   020136
                        PRINTX  #FMT9,#DH9
  (8)  020136  012746  015271                                    MOV     #DH9,-(SP)
```

```
   (7)  020142  012746  012522                                                    MOV     #FMT9,-(SP)
   (6)  020146  012746  000002                                                    MOV     #2,-(SP)
   (3)  020152  010600                                                            MOV     SP,R0
   (4)  020154  104415                                                            TRAP    C$PNTX
   (4)  020156  062706  000006                                                    ADD     #6,SP
 7461   020162                        PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
  (11)  020162  013746  002324                                                    MOV     AX3.16,-(SP)
  (10)  020166  013746  002322                                                    MOV     AX3.15,-(SP)
   (9)  020172  013746  002320                                                    MOV     AX2.16,-(SP)
   (8)  020176  013746  002316                                                    MOV     AX2.15,-(SP)
   (7)  020202  012746  012423                                                    MOV     #FMT6,-(SP)
   (6)  020206  012746  000005                                                    MOV     #5,-(SP)
   (3)  020212  010600                                                            MOV     SP,R0
   (4)  020214  104415                                                            TRAP    C$PNTX
   (4)  020216  062706  000014                                                    ADD     #14,SP
 7462   020222                        ENDMSG
   (3)  020222                                                          L10007:
   (3)  020222  104423                                                            TRAP    C$MSG
 7463
 7464
 7465
 7466
 7467
 7468   020224                        BGNMSG  ERR7
   (3)  020224                                                          ERR7::
 7469   020224                        PRINTB  #FMT1,#ADDRES,MPCSR
   (9)  020224  013746  002436                                                    MOV     MPCSR,-(SP)
   (8)  020230  012746  036452                                                    MOV     #ADDRES,-(SP)
   (7)  020234  012746  012264                                                    MOV     #FMT1,-(SP)
   (6)  020240  012746  000003                                                    MOV     #3,-(SP)
   (3)  020244  010600                                                            MOV     SP,R0
   (4)  020246  104414                                                            TRAP    C$PNTB
   (4)  020250  062706  000010                                                    ADD     #10,SP
 7470   020254                        PRINTB  #FMT2
   (7)  020254  012746  012274                                                    MOV     #FMT2,-(SP)
   (6)  020260  012746  000001                                                    MOV     #1,-(SP)
   (3)  020264  010600                                                            MOV     SP,R0
   (4)  020266  104414                                                            TRAP    C$PNTB
   (4)  020270  062706  000004                                                    ADD     #4,SP
 7471   020274                        PRINTB  #FMT7,#DH1,REGNUM
   (9)  020274  013746  002364                                                    MOV     REGNUM,-(SP)
   (8)  020300  012746  015021                                                    MOV     #DH1,-(SP)
   (7)  020304  012746  012456                                                    MOV     #FMT7,-(SP)
   (6)  020310  012746  000003                                                    MOV     #3,-(SP)
   (3)  020314  010600                                                            MOV     SP,R0
   (4)  020316  104414                                                            TRAP    C$PNTB
   (4)  020320  062706  000010                                                    ADD     #10,SP
 7472   020324                        PRINTX  #FMT4,#DH2,#DH3
   (9)  020324  012746  015072                                                    MOV     #DH3,-(SP)
   (8)  020330  012746  015043                                                    MOV     #DH2,-(SP)
   (7)  020334  012746  012360                                                    MOV     #FMT4,-(SP)
   (6)  020340  012746  000003                                                    MOV     #3,-(SP)
   (3)  020344  010600                                                            MOV     SP,R0
   (4)  020346  104415                                                            TRAP    C$PNTX
   (4)  020350  062706  000010                                                    ADD     #10,SP
 7473   020354                        PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
```

```
  (11) 020354  013746  002274                                    MOV     LUR13,-(SP)
  (10) 020360  013746  002272                                    MOV     LUR12,-(SP)
   (9) 020364  013746  002270                                    MOV     LUR11,-(SP)
   (8) 020370  013746  002266                                    MOV     LUR10,-(SP)
   (7) 020374  012746  012373                                    MOV     #FMT5,-(SP)
   (6) 020400  012746  000005                                    MOV     #5,-(SP)
   (3) 020404  010600                                            MOV     SP,R0
   (4) 020406  104415                                            TRAP    C$PNTX
   (4) 020410  062706  000014                                    ADD     #14,SP
7474   020414                PRINTX  #FMT9,#DH4
   (8) 020414  012746  015130                                    MOV     #DH4,-(SP)
   (7) 020420  012746  012522                                    MOV     #FMT9,-(SP)
   (6) 020424  012746  000002                                    MOV     #2,-(SP)
   (3) 020430  010600                                            MOV     SP,R0
   (4) 020432  104415                                            TRAP    C$PNTX
   (4) 020434  062706  000006                                    ADD     #6,SP
7475   020440                PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
  (11) 020440  013746  002304                                    MOV     LUR17,-(SP)
  (10) 020444  013746  002302                                    MOV     LUR16,-(SP)
   (9) 020450  013746  002300                                    MOV     LUR15,-(SP)
   (8) 020454  013746  002276                                    MOV     LUR14,-(SP)
   (7) 020460  012746  012423                                    MOV     #FMT6,-(SP)
   (6) 020464  012746  000005                                    MOV     #5,-(SP)
   (3) 020470  010600                                            MOV     SP,R0
   (4) 020472  104415                                            TRAP    C$PNTX
   (4) 020474  062706  000014                                    ADD     #14,SP
7476   020500                PRINTX  #FMT4,#DH7,#DH8
   (9) 020500  012746  015232                                    MOV     #DH8,-(SP)
   (8) 020504  012746  015200                                    MOV     #DH7,-(SP)
   (7) 020510  012746  012360                                    MOV     #FMT4,-(SP)
   (6) 020514  012746  000003                                    MOV     #3,-(SP)
   (3) 020520  010600                                            MOV     SP,R0
   (4) 020522  104415                                            TRAP    C$PNTX
   (4) 020524  062706  000010                                    ADD     #10,SP
7477   020530                PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
  (11) 020530  013746  002314                                    MOV     AX1.16,-(SP)
  (10) 020534  013746  002312                                    MOV     AX1.15,-(SP)
   (9) 020540  013746  002310                                    MOV     AX0.16,-(SP)
   (8) 020544  013746  002306                                    MOV     AX0.15,-(SP)
   (7) 020550  012746  012373                                    MOV     #FMT5,-(SP)
   (6) 020554  012746  000005                                    MOV     #5,-(SP)
   (3) 020560  010600                                            MOV     SP,R0
   (4) 020562  104415                                            TRAP    C$PNTX
   (4) 020564  062706  000014                                    ADD     #14,SP
7478   020570                PRINTX  #FMT9,#DH9
   (8) 020570  012746  015271                                    MOV     #DH9,-(SP)
   (7) 020574  012746  012522                                    MOV     #FMT9,-(SP)
   (6) 020600  012746  000002                                    MOV     #2,-(SP)
   (3) 020604  010600                                            MOV     SP,R0
   (4) 020606  104415                                            TRAP    C$PNTX
   (4) 020610  062706  000006                                    ADD     #6,SP
7479   020614                PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
  (11) 020614  013746  002324                                    MOV     AX3.16,-(SP)
  (10) 020620  013746  002322                                    MOV     AX3.15,-(SP)
   (9) 020624  013746  002320                                    MOV     AX2.16,-(SP)
   (8) 020630  013746  002316                                    MOV     AX2.15,-(SP)
```

```
  (7)  020634  012746  012423                                                  MOV     #FMT6,-(SP)
  (6)  020640  012746  000005                                                  MOV     #5,-(SP)
  (3)  020644  010600                                                          MOV     SP,R0
  (4)  020646  104415                                                          TRAP    C$PNTX
  (4)  020650  062706  000014                                                  ADD     #14,SP
7480   020654                        ENDMSG
  (3)  020654                                                          L10010:
  (3)  020654  104423                                                          TRAP    C$MSG
7481
7482
7483
7484
7485
7486   020656                        BGNMSG  ERR8
  (3)  020656                                                          ERR8::
7487   020656                        PRINTB  #FMT10,SUBRPC
  (8)  020656  013746  002336                                                  MOV     SUBRPC,-(SP)
  (7)  020662  012746  012527                                                  MOV     #FMT10,-(SP)
  (6)  020666  012746  000002                                                  MOV     #2,-(SP)
  (3)  020672  010600                                                          MOV     SP,R0
  (4)  020674  104414                                                          TRAP    C$PNTB
  (4)  020676  062706  000006                                                  ADD     #6,SP
7488   020702                        PRINTB  #FMT1,#ADDRES,MPCSR
  (9)  020702  013746  002436                                                  MOV     MPCSR,-(SP)
  (8)  020706  012746  036452                                                  MOV     #ADDRES,-(SP)
  (7)  020712  012746  012264                                                  MOV     #FMT1,-(SP)
  (6)  020716  012746  000003                                                  MOV     #3,-(SP)
  (3)  020722  010600                                                          MOV     SP,R0
  (4)  020724  104414                                                          TRAP    C$PNTB
  (4)  020726  062706  000010                                                  ADD     #10,SP
7489   020732                        PRINTB  #FMT2
  (7)  020732  012746  012274                                                  MOV     #FMT2,-(SP)
  (6)  020736  012746  000001                                                  MOV     #1,-(SP)
  (3)  020742  010600                                                          MOV     SP,R0
  (4)  020744  104414                                                          TRAP    C$PNTB
  (4)  020746  062706  000004                                                  ADD     #4,SP
7490   020752                        PRINTB  #FMT7,#DH1,REGNUM
  (9)  020752  013746  002364                                                  MOV     REGNUM,-(SP)
  (8)  020756  012746  015021                                                  MOV     #DH1,-(SP)
  (7)  020762  012746  012456                                                  MOV     #FMT7,-(SP)
  (6)  020766  012746  000003                                                  MOV     #3,-(SP)
  (3)  020772  010600                                                          MOV     SP,R0
  (4)  020774  104414                                                          TRAP    C$PNTB
  (4)  020776  062706  000010                                                  ADD     #10,SP
7491   021002                        PRINTB  #FMT3,GOODAT,BADDAT
  (9)  021002  013746  002372                                                  MOV     BADDAT,-(SP)
  (8)  021006  013746  002370                                                  MOV     GOODAT,-(SP)
  (7)  021012  012746  012316                                                  MOV     #FMT3,-(SP)
  (6)  021016  012746  000003                                                  MOV     #3,-(SP)
  (3)  021022  010600                                                          MOV     SP,R0
  (4)  021024  104414                                                          TRAP    C$PNTB
  (4)  021026  062706  000010                                                  ADD     #10,SP
7492   021032                        PRINTX  #FMT4,#DH2,#DH3
  (9)  021032  012746  015072                                                  MOV     #DH3,-(SP)
  (8)  021036  012746  015043                                                  MOV     #DH2,-(SP)
  (7)  021042  012746  012360                                                  MOV     #FMT4,-(SP)
```

```
  (6)  021046  012746  000003                                    MOV     #3,-(SP)
  (3)  021052  010600                                            MOV     SP,R0
  (4)  021054  104415                                            TRAP    C$PNTX
  (4)  021056  062706  000010                                    ADD     #10,SP
7493   021062                  PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 (11)  021062  013746  002274                                    MOV     LUR13,-(SP)
 (10)  021066  013746  002272                                    MOV     LUR12,-(SP)
  (9)  021072  013746  002270                                    MOV     LUR11,-(SP)
  (8)  021076  013746  002266                                    MOV     LUR10,-(SP)
  (7)  021102  012746  012373                                    MOV     #FMT5,-(SP)
  (6)  021106  012746  000005                                    MOV     #5,-(SP)
  (3)  021112  010600                                            MOV     SP,R0
  (4)  021114  104415                                            TRAP    C$PNTX
  (4)  021116  062706  000014                                    ADD     #14,SP
7494   021122                  PRINTX  #FMT9,#DH4
  (8)  021122  012746  015130                                    MOV     #DH4,-(SP)
  (7)  021126  012746  012522                                    MOV     #FMT9,-(SP)
  (6)  021132  012746  000002                                    MOV     #2,-(SP)
  (3)  021136  010600                                            MOV     SP,R0
  (4)  021140  104415                                            TRAP    C$PNTX
  (4)  021142  062706  000006                                    ADD     #6,SP
7495   021146                  PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
 (11)  021146  013746  002304                                    MOV     LUR17,-(SP)
 (10)  021152  013746  002302                                    MOV     LUR16,-(SP)
  (9)  021156  013746  002300                                    MOV     LUR15,-(SP)
  (8)  021162  013746  002276                                    MOV     LUR14,-(SP)
  (7)  021166  012746  012423                                    MOV     #FMT6,-(SP)
  (6)  021172  012746  000005                                    MOV     #5,-(SP)
  (3)  021176  010600                                            MOV     SP,R0
  (4)  021200  104415                                            TRAP    C$PNTX
  (4)  021202  062706  000014                                    ADD     #14,SP
7496   021206                  PRINTX  #FMT4,#DH7,#DH8
  (9)  021206  012746  015232                                    MOV     #DH8,-(SP)
  (8)  021212  012746  015200                                    MOV     #DH7,-(SP)
  (7)  021216  012746  012360                                    MOV     #FMT4,-(SP)
  (6)  021222  012746  000003                                    MOV     #3,-(SP)
  (3)  021226  010600                                            MOV     SP,R0
  (4)  021230  104415                                            TRAP    C$PNTX
  (4)  021232  062706  000010                                    ADD     #10,SP
7497   021236                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
 (11)  021236  013746  002314                                    MOV     AX1.16,-(SP)
 (10)  021242  013746  002312                                    MOV     AX1.15,-(SP)
  (9)  021246  013746  002310                                    MOV     AX0.16,-(SP)
  (8)  021252  013746  002306                                    MOV     AX0.15,-(SP)
  (7)  021256  012746  012373                                    MOV     #FMT5,-(SP)
  (6)  021262  012746  000005                                    MOV     #5,-(SP)
  (3)  021266  010600                                            MOV     SP,R0
  (4)  021270  104415                                            TRAP    C$PNTX
  (4)  021272  062706  000014                                    ADD     #14,SP
7498   021276                  PRINTX  #FMT9,#DH9
  (8)  021276  012746  015271                                    MOV     #DH9,-(SP)
  (7)  021302  012746  012522                                    MOV     #FMT9,-(SP)
  (6)  021306  012746  000002                                    MOV     #2,-(SP)
  (3)  021312  010600                                            MOV     SP,R0
  (4)  021314  104415                                            TRAP    C$PNTX
  (4)  021316  062706  000006                                    ADD     #6,SP
```

```
7499  021322
 (11) 021322  013746  002324          PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
 (10) 021326  013746  002322                                                  MOV     AX3.16,-(SP)
 (9)  021332  013746  002320                                                  MOV     AX3.15,-(SP)
 (8)  021336  013746  002316                                                  MOV     AX2.16,-(SP)
 (7)  021342  012746  012423                                                  MOV     AX2.15,-(SP)
 (6)  021346  012746  000005                                                  MOV     #FMT6,-(SP)
 (3)  021352  010600                                                          MOV     #5,-(SP)
 (4)  021354  104415                                                          MOV     SP,R0
 (4)  021356  062706  000014                                                  TRAP    C$PNTX
7500  021362                                                                  ADD     #14,SP
                                      ENDMSG
 (3)  021362                                                          L10011:
 (3)  021362  104423                                                          TRAP    C$MSG
7501
7502
7503
7504
7505
7506  021364                          BGNMSG  ERR10
 (3)  021364                                                          ERR10::
7507  021364                          PRINTB  #FMT1,#ADDRES,MPCSR
 (9)  021364  013746  002436                                                  MOV     MPCSR,-(SP)
 (8)  021370  012746  036452                                                  MOV     #ADDRES,-(SP)
 (7)  021374  012746  012264                                                  MOV     #FMT1,-(SP)
 (6)  021400  012746  000003                                                  MOV     #3,-(SP)
 (3)  021404  010600                                                          MOV     SP,R0
 (4)  021406  104414                                                          TRAP    C$PNTB
 (4)  021410  062706  000010                                                  ADD     #10,SP
7508  021414                          PRINTB  #FMT2
 (7)  021414  012746  012274                                                  MOV     #FMT2,-(SP)
 (6)  021420  012746  000001                                                  MOV     #1,-(SP)
 (3)  021424  010600                                                          MOV     SP,R0
 (4)  021426  104414                                                          TRAP    C$PNTB
 (4)  021430  062706  000004                                                  ADD     #4,SP
7509  021434                          PRINTB  #FMT8,TMP1,TMP0
 (9)  021434  013746  002516                                                  MOV     TMP0,-(SP)
 (8)  021440  013746  002520                                                  MOV     TMP1,-(SP)
 (7)  021444  012746  012466                                                  MOV     #FMT8,-(SP)
 (6)  021450  012746  000003                                                  MOV     #3,-(SP)
 (3)  021454  010600                                                          MOV     SP,R0
 (4)  021456  104414                                                          TRAP    C$PNTB
 (4)  021460  062706  000010                                                  ADD     #10,SP
7510  021464                          PRINTX  #FMT4,#DH2,#DH3
 (9)  021464  012746  015072                                                  MOV     #DH3,-(SP)
 (8)  021470  012746  015043                                                  MOV     #DH2,-(SP)
 (7)  021474  012746  012360                                                  MOV     #FMT4,-(SP)
 (6)  021500  012746  000003                                                  MOV     #3,-(SP)
 (3)  021504  010600                                                          MOV     SP,R0
 (4)  021506  104415                                                          TRAP    C$PNTX
 (4)  021510  062706  000010                                                  ADD     #10,SP
7511  021514                          PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 (11) 021514  013746  002274                                                  MOV     LUR13,-(SP)
 (10) 021520  013746  002272                                                  MOV     LUR12,-(SP)
 (9)  021524  013746  002270                                                  MOV     LUR11,-(SP)
 (8)  021530  013746  002266                                                  MOV     LUR10,-(SP)
 (7)  021534  012746  012373                                                  MOV     #FMT5,-(SP)
```

```
  (6)  021540  012746  000005                                    MOV     #5,-(SP)
  (3)  021544  010600                                            MOV     SP,R0
  (4)  021546  104415                                            TRAP    C$PNTX
  (4)  021550  062706  000014                                    ADD     #14,SP
7512   021554                  PRINTX  #FMT9,#DH4
  (8)  021554  012746  015130                                    MOV     #DH4,-(SP)
  (7)  021560  012746  012522                                    MOV     #FMT9,-(SP)
  (6)  021564  012746  000002                                    MOV     #2,-(SP)
  (3)  021570  010600                                            MOV     SP,R0
  (4)  021572  104415                                            TRAP    C$PNTX
  (4)  021574  062706  000006                                    ADD     #6,SP
7513   021600                  PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
 (11)  021600  013746  002304                                    MOV     LUR17,-(SP)
 (10)  021604  013746  002302                                    MOV     LUR16,-(SP)
  (9)  021610  013746  002300                                    MOV     LUR15,-(SP)
  (8)  021614  013746  002276                                    MOV     LUR14,-(SP)
  (7)  021620  012746  012423                                    MOV     #FMT6,-(SP)
  (6)  021624  012746  000005                                    MOV     #5,-(SP)
  (3)  021630  010600                                            MOV     SP,R0
  (4)  021632  104415                                            TRAP    C$PNTX
  (4)  021634  062706  000014                                    ADD     #14,SP
7514   021640                  PRINTX  #FMT4,#DH7,#DH8
  (9)  021640  012746  015232                                    MOV     #DH8,-(SP)
  (8)  021644  012746  015200                                    MOV     #DH7,-(SP)
  (7)  021650  012746  012360                                    MOV     #FMT4,-(SP)
  (6)  021654  012746  000003                                    MOV     #3,-(SP)
  (3)  021660  010600                                            MOV     SP,R0
  (4)  021662  104415                                            TRAP    C$PNTX
  (4)  021664  062706  000010                                    ADD     #10,SP
7515   021670                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
 (11)  021670  013746  002314                                    MOV     AX1.16,-(SP)
 (10)  021674  013746  002312                                    MOV     AX1.15,-(SP)
  (9)  021700  013746  002310                                    MOV     AX0.16,-(SP)
  (8)  021704  013746  002306                                    MOV     AX0.15,-(SP)
  (7)  021710  012746  012373                                    MOV     #FMT5,-(SP)
  (6)  021714  012746  000005                                    MOV     #5,-(SP)
  (3)  021720  010600                                            MOV     SP,R0
  (4)  021722  104415                                            TRAP    C$PNTX
  (4)  021724  062706  000014                                    ADD     #14,SP
7516   021730                  PRINTX  #FMT9,#DH9
  (8)  021730  012746  015271                                    MOV     #DH9,-(SP)
  (7)  021734  012746  012522                                    MOV     #FMT9,-(SP)
  (6)  021740  012746  000002                                    MOV     #2,-(SP)
  (3)  021744  010600                                            MOV     SP,R0
  (4)  021746  104415                                            TRAP    C$PNTX
  (4)  021750  062706  000006                                    ADD     #6,SP
7517   021754                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
 (11)  021754  013746  002324                                    MOV     AX3.16,-(SP)
 (10)  021760  013746  002322                                    MOV     AX3.15,-(SP)
  (9)  021764  013746  002320                                    MOV     AX2.16,-(SP)
  (8)  021770  013746  002316                                    MOV     AX2.15,-(SP)
  (7)  021774  012746  012423                                    MOV     #FMT6,-(SP)
  (6)  022000  012746  000005                                    MOV     #5,-(SP)
  (3)  022004  010600                                            MOV     SP,R0
  (4)  022006  104415                                            TRAP    C$PNTX
  (4)  022010  062706  000014                                    ADD     #14,SP
```

```
 7518  022014                          ENDMSG
  (3)  022014
  (3)  022014  104423                                    L10012:
                                                                 TRAP    C$MSG
 7519
 7520
 7521
 7522
 7523
```

```
                                .SBTTL  REPORT CODING SECTION


                                ;////////////////////////////////////////////////////////////
                                ;/ THE REPORT CODING SECTION CONTAINS THE
                                ;/ 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;////////////////////////////////////////////////////////////

022016                                  BGNRPT
(3)  022016                                                             L$RPT::

022016                                  ENDRPT
(3)  022016                                                             L10013:
(3)  022016  104425                                                             TRAP    C$RPT

                                .EVEN
```

```
                                .SBTTL  LOAD DEVICE PROTECTION TABLE

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

022020                                  BGNPROT
(3)  022020                                                                     L$PROT::
022020  177777                         .WORD   -1      ;DON'T CHK CSR ADRS
022022  177777                         .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
022024  177777                         .WORD   -1      ;DON'T CHK DRIVE NO.
022026                                  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$INIT::
                                MOV     SP,PSTACK       ;SAVE BASE-LEVEL STACK POINTER
                                CLR     SUBRPC          ;CLEAR SUBR CALL PC
                                CLR     DISILO          ;CLEAR CURRENT STATE OF DISSI
                                CLR     CHPTYP          ;CLEAR USYRT CHIP TYPE INDICATOR
                                CLR     ERROR1          ;CLEAR ERROR FLAGS
                                CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
                                TST     FRSTIM          ;SEE IF FIRST TIME THROUGH AFTER LOAD
                                BNE     6$              ;BR IF NOT
                                MOV     @#4,SAVE4       ;SAVE ERROR TRAP VECTOR
                                MOV     @#6,SAVE6
                                BR      9$
                        6$:     MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
                                MOV     SAVE6,@#6
                        9$:     MOV     #1,FRSTIM       ;MARK FLAG FOR NEXT TIME THROUGH
                        ;SEE IF PROGRAM JUST STARTED, BR IF YES
                                READEF  #EF.START
                                                                        MOV     #EF.START,R0
                                                                        TRAP    C$REFG
                                BCOMPLETE       STARST
                                                                        BCS     STARST
                        ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
                                READEF  #EF.RESTART
                                                                        MOV     #EF.RESTART,R0
                                                                        TRAP    C$REFG
                                BCOMPLETE       STARST
                                                                        BCS     STARST
                        ;SEE IF THIS IS A NEW PASS, BR IF YES
                                READEF  #EF.NEW
                                                                        MOV     #EF.NEW,R0
                                                                        TRAP    C$REFG
                                BCOMPLETE       NEWST
                                                                        BCS     NEWST
                        ;SEE IF PROGRAM WAS JUST CONTINUED
                                READEF  #EF.CONTINUE
                                                                        MOV     #EF.CONTINUE,R0
                                                                        TRAP    C$REFG
                                BCOMPLETE       ENDIT
                                                                        BCS     ENDIT
                                BR      GETPRM
                        STARST:
                                CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
                        ;CLEAR DEVICE MAP
                                CLR     DEVMAP
                        NEWST:
                                MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
                                INC     FRSPAS          ;INCREMENT NO. OF PASSES AFTER LOAD
```

```
                        INC     STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
                        MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
                ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
                ;   CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
                GETPRM:
                        INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
                        CMP     LOGDEV,L$UNIT   ;SEE IF MAXIMUM UNIT NO. EXCEEDED
                        BGE     NEWST           ;BR IF YES
                        GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
(3)  022236  013700  002330                                             MOV     LOGDEV,R0
(3)  022242  104442                                                     TRAP    C$GPHRD
(3)  022244  010001                                                     MOV     R0,R1
                        BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
(2)  022246  103403                                                     BCS     10$
                        ASL     DEVPTR          ;SHIFT DEVICE MAP BIT POINTER
                        BR      GETPRM          ;SKIP THIS DEVICE
                10$:    BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE IN DEVICE MAP
                        ASL     DEVPTR          ;SHIFT DEVICE MAP BIT POINTER
                        ADD     #2,R1           ;INCREMENT R1 PAST MICROPROCESSOR TYPE
                        MOV     (R1),MPCSR      ;STORE POINTER TO MICROPROCESSOR CSR'S
                        MOV     (R1),BSEL1
                        INC     BSEL1           ;GET POINTER TO BSEL1 (MAINTENANCE REGISTER)
                        MOV     (R1),SEL4
                        ADD     #4,SEL4         ;GET POINTER TO SEL4
                        MOV     (R1)+,SEL6
                        ADD     #6,SEL6         ;STORE POINTER TO SEL6
                        MOV     (R1),MPIVEC     ;GET MICROPROCESSOR INPUT INTRPT VECTOR
                        MOV     (R1)+,MPOVEC
                        ADD     #4,MPOVEC       ;GET MICROPROCESSOR OUTPUT INTRPT VECTOR
                        MOV     (R1)+,MPRIOR    ;GET MICROPROCESSOR DEVICE PRIORITY
                        ADD     #2,R1           ;INCREMENT R1 PAST LU TYPE
                        MOV     (R1)+,LUSWI1    ;GET LU SWITCH PACK #1
                        MOV     (R1)+,LUSWI2    ;GET LU SWITCH PACK #2
                        MOV     (R1)+,LUSWI3    ;GET LU SWITCH PACK #3
                        MOV     (R1)+,TSTCON    ;GET TEST CONNECTOR INDICATOR
                        MOV     (R1),BDRATE     ;GET BAUD RATE
                ;SEE IF MANUAL INTERVENTION DESIRED BETWEEN UNITS FOR INSTALLATION OR REMOVAL
                ;   OF TEST CONNECTORS, BR IF NOT
                        TST     MIFLAG
                        BEQ     22$
                ;SEE IF MANUAL INTERVENTION ALLOWED BY SUPERVISOR
                        MANUAL
(3)  022414  104450                                                     TRAP    C$MANI
                ;BR IF ALLOWED
                        BCOMPLETE       18$
(2)  022416  103412                                                     BCS     18$
                ;PRINT MSG THAT OPERATOR INTERVENTION IS NOT ALLOWED
                        PRINTF  #FMT16
(7)  022420  012746  022522                                             MOV     #FMT16,-(SP)
(6)  022424  012746  000001                                             MOV     #1,-(SP)
(3)  022430  010600                                                     MOV     SP,R0
(4)  022432  104417                                                     TRAP    C$PNTF
(4)  022434  062706  000004                                             ADD     #4,SP
                16$:    BREAK           ;HANG UNTIL ^C TYPED
(3)  022440  104422                                                     TRAP    C$BRK
                        BR      16$
```

```
 7649  022444                                  18$:
 7650                                          ;TYPE ''INSTALL TEST CONNECTOR(S) ON UNIT AT ADRS XXXXXX''
 7651  022444                                          PRINTF  #FMT17,MPCSR
  (8)  022444  013746  002436                                                    MOV     MPCSR,-(SP)
  (7)  022450  012746  022644                                                    MOV     #FMT17,-(SP)
  (6)  022454  012746  000002                                                    MOV     #2,-(SP)
  (3)  022460  010600                                                            MOV     SP,R0
  (4)  022462  104417                                                            TRAP    C$PNTF
  (4)  022464  062706  000006                                                    ADD     #6,SP
 7652  022470  005037  002502                  CLR     REG2
 7653  022474                                  20$:
 7654                                          ;ASK OPERATOR TO ''TYPE <Y> <CR> WHEN READY TO PROCEED''
 7655  022474                                          GMANIL  TYPEY,REG2,1,NO
  (3)  022474  104443                                                            TRAP    C$GMAN
  (3)  022476  000404                                                            BR      10000$
  (4)  022500  002502                                                            .WORD   REG2
  (5)  022502  000120                                                            .WORD   T$CODE
  (5)  022504  022733                                                            .WORD   TYPEY
  (5)  022506  000001                                                            .WORD   1
  (3)  022510                                                           10000$:
 7656  022510  023727  002502  000001          CMP     REG2,#1
 7657  022516  001366                          BNE     20$
 7658  022520                                  22$:
 7659  022520                                  ENDIT:
 7660  022520                                          ENDINIT
  (3)  022520                                                           L10015:
  (3)  022520  104411                                                            TRAP    C$INIT
 7661
 7662  022522  047045  040445  040515  FMT16:  .ASCII  /%N%AMANUAL INTERVENTION IS NOT ALLOWED!%N/
       022530  052516  046101  044440
       022536  052116  051105  042526
       022544  052116  047511  020116
       022552  051511  047040  052117
       022560  040440  046114  053517
       022566  042105  022441     116
 7663  022573     045  052101  050131          .ASCIZ  /%ATYPE CONTROL-C (^C) <CR> TO PROCEED:%N/
       022600  020105  047503  052116
       022606  047522  026514  020103
       022614  057050  024503  036040
       022622  051103  020076  047524
       022630  050040  047522  042503
       022636  042105  022472  000116
 7664  022644  047045  040445  047111  FMT17:  .ASCIZ  /%N%AINSTALL TEST CONNECTOR(S) ON UNIT AT ADRS :  %06%N/
       022652  052123  046101  020114
       022660  042524  052123  041440
       022666  047117  042516  052103
       022674  051117  051450  020051
       022702  047117  052440  044516
       022710  020124  052101  040440
       022716  051104  020123  020072
       022724  022440  033117  047045
       022732     000
 7665  022733     124  050131  020105  TYPEY:  .ASCIZ  /TYPE <Y><CR> WHEN READY TO PROCEED /
       022740  054474  036076  051103
       022746  020076  044127  047105
       022754  051040  040505  054504
```

```
        022762  052040  020117  051120
        022770  041517  042505  020104
        022776     000
  7666          023000                  .EVEN
  7667
  7668
  7669
  7670
  7671
```

```
                        .SBTTL  AUTO DROP UNIT SECTION

                        ;////////////////////////////////////////////////////////////////////////
                        ;/ THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
                        ;/ WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
                        ;////////////////////////////////////////////////////////////////////////

023000                          BGNAUTO
(3) 023000                                                              L$AUTO::
                        ;ESTABLISH PRIORITY = 7
023000                          SETPRI  #PRI07
(3) 023000  012700  000340                                                      MOV     #PRI07,R0
(3) 023004  104441                                                              TRAP    C$SPRI
023006  012737  023030  000004          MOV     #6$,@#4         ;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR
023014  012737  000340  000006          MOV     #PRI07,@#6
023022  005777  157410                  TST     @MPCSR          ;ADDRESS SEL0
023026  000405                          BR      9$              ;TAKE THIS BRANCH IF DEVICE RESPONDS
                        ;COME HERE IF DEVICE CSR IS NON-EXISTENT
023030  062706  000004          6$:     ADD     #4,SP           ;CLEAN UP THE STACK POINTER
023034                          DODU    LOGDEV          ;DROP THIS UNIT FROM TESTING
(3) 023034  013700  002330                                                      MOV     LOGDEV,R0
(3) 023040  104451                                                              TRAP    C$DODU
023042  013737  002404  000004  9$:     MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
023050  013737  002406  000006          MOV     SAVE6,@#6
023056                          ENDAUTO
(3) 023056                                                              L10016:
(3) 023056  104461                                                              TRAP    C$AUTO
```

```
                                .SBTTL  CLEANUP CODING SECTION

                                ;//////////////////////////////////////////////////////////////////////////
                                ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;/ AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
                                ;//////////////////////////////////////////////////////////////////////////

023060                                  BGNCLN
(3) 023060                                                              L$CLEAN::

                                        ENDCLN
(3) 023060                                                              L10017:
(3) 023060  104412                                                              TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

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

023062                                  BGNDU
023062                                                                  L$DU::
                                ;ISSUE UNIBUS RESET TO CLEAN UP
023062                                  BRESET
023062  104433                                                                  TRAP    C$RESET
                                ;PRINT 'UNIT XX DROPPED'
023064                                  PRINTF  #FMT27,LOGDEV
023064  013746  002330                                                          MOV     LOGDEV,-(SP)
023070  012746  023112                                                          MOV     #FMT27,-(SP)
023074  012746  000002                                                          MOV     #2,-(SP)
023100  010600                                                                  MOV     SP,R0
023102  104417                                                                  TRAP    C$PNTF
023104  062706  000006                                                          ADD     #6,SP
023110                                  ENDDU
023110                                                                  L10020:
023110  104453                                                                  TRAP    C$DU

023112  047045  040445  047125  FMT27:  .ASCIZ  /%N%AUNIT %D2%A DROPPED%N/
023120  052111  022440  031104
023126  040445  042040  047522
023134  050120  042105  047045
023142     000
023144                          .EVEN
```

```
                                .SBTTL  ADD UNIT SECTION

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

023144                                  BGNAU
(3)  023144
023144                                  ENDAU                           L$AU::
(3)  023144
(3)  023144  104452                                                     L10021:  TRAP    C$AU
```

```
                                .SBTTL  HARDWARE TESTS
                                ;;*****************************************************************************
                                .SBTTL      TEST 1 - BIT STUFFING TEST
                                ;*
                                ;* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
                                ;* INITIATED IN BIT MODE . TWO LEADING FLAGS ARE SENT,
                                ;* FOLLOWED BY ALL SIXTEEN CHARS IN DATA PATTERN S. THIS PATTERN
                                ;* CONSISTS OF CHARACTERS WHICH REQUIRE NO BIT STUFFING AND CHARACTERS
                                ;* WHICH REQUIRE BIT STUFFING INDIVIDUALLY AND IN COMBINATION WITH
                                ;* ADJACENT CHARACTERS. ALL 16 CHARACTERS ARE READ AND COMPARED
                                ;* BY THE RECEIVER.
                                ;*   PATTERN S = 000,017,036,074,170,360,037,076,174,370,077,176,374,
                                ;*               177,376,377
                                ;;*****************************************************************************
023146                          BGNTST
(3) 023146                                                                      T1::
023146  012737  023270  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
023154  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
023160  000000                          000
023162  000300                          CRC2!CRC1               ;BIT MODE, NO ERR DETECTION
023164  004737  010554                  JSR     PC,LDBYTS       ;LOAD PAT S INTO TX SILO
023170  002642                          PATS
023172  000020                          16.
023174  012737  001000  002412          MOV     #TXEOM,TXWORD
023202  004737  004646                  JSR     PC,LDTXSI       ;LOAD 2 EOM'S INTO TX SILO
023206  004737  004646                  JSR     PC,LDTXSI
023212  004737  004726                  JSR     PC,STPLU        ;CLK MORE THAN ENTIRE MSG
023216  000300                          192.
023220  012701  002642                  MOV     #PATS,R1        ;INIT PAT S POINTER
023224  112137  023234          6$:     MOVB    (R1)+,8$
023230  004737  007266                  JSR     PC,CKDATA       ;CHK A RCV'D CHAR
023234  000000          8$:             .WORD   0
023236  000000                          0
023240  020127  002661                  CMP     R1,#PATS+15.    ;SEE IF 15 CHARS CHECKED YET
023244  103767                          BLO     6$              ;BR IF NOT YET
023246  111137  023264                  MOVB    (R1),12$
023252  052737  001000  023264          BIS     #RXEBL,12$      ;GET SET TO CHK EBLK = 1
023260  004737  007266                  JSR     PC,CKDATA       ;CHK LAST CHAR AND EBLK = 1
023264  000000          12$:            .WORD   0
023266  000000                          0
023270  004737  003276          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
023274                          ENDTST
(3) 023274                                                                              L10022:
(3) 023274  104401                                                                              TRAP    C$ETST
                                ;;*****************************************************************************
                                .SBTTL      TEST 2 - RCV OVERRUN ERROR SET AND CLEAR TEST
```

```
                                        ;*
                                        ;* IN THIS TEST, A RCV OVERRUN ERROR IS FORCED IN EACH OF 2 SUBTESTS.
                                        ;* IN THE FIRST, A MESSAGE IS INITIATED, 64 001 CHARS ARE SENT, AND THE
                                        ;* RECEIVER IS NOT SERVICED IN RESPONSE TO THE USYRT RCV FLAG, WHICH CAUSES RCV
                                        ;* OVERRUN TO SET. THEN, A CHECK IS MADE TO INSURE THAT OVRR IS NOT
                                        ;* CLEARED BY THE LINE UNIT READING THE USYRT STATUS.
                                        ;* THEN, IC IS SET TO CLEAR THE ERROR, AND THIS IS VERIFIED.
                                        ;*
                                        ;* IN THE SECOND SUBTEST, RCV OVRUN IS FORCED AGAIN, AND A MASTER CLEAR
                                        ;* IS ISSUED TO CLEAR THE ERROR, AND THIS IS VERIFIED.
                                        ;;*****************************************************************************
023276                                  BGNTST
023276                                                                          T2::
023276  012737  023644  002346                MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
                                        ;-------------------------------------------------------------------------
                                        ; CAUSE OVRR, SET IC TO CLEAR IT
                                        ;-------------------------------------------------------------------------
023304  004737  005212                        JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
023310  000226                                SYNCH
023312  000311                                CRC2!CRC1!STRIP!DDCMP   ;DDCMP, NO ERR DET
023314  004737  010716                        JSR     PC,LODSIL       ;LOAD 64 001 CHARS INTO TX SILO
023320  000001                                001
023322  000100                                64.
023324  004737  006752                        JSR     PC,RCV1ST       ;CLOCK UNTIL FIRST DATA CHAR RCV'D
023330  000030                                24.
023332  004737  004726                        JSR     PC,STPLU        ;CLOCK UNTIL 59 MORE RCV'D
023336  000730                                472.
023340  004737  010716                        JSR     PC,LODSIL       ;LOAD 60 MORE INTO TX SILO
023344  000001                                001
023346  000074                                60.
023350  004737  004726                        JSR     PC,STPLU        ;CLK 8 MORE TIMES TO FORCE UNDERRUN
023354  000100                                64.
023356  012701  000100                        MOV     #64.,R1         ;READ AND CHK 64 CHARS FROM RCV SILO
023362  004737  007266                6$:     JSR     PC,CKDATA
023366  000001                                001
023370  000000                                0
023372  005301                                DEC     R1
023374  001372                                BNE     6$
023376  004737  007266                        JSR     PC,CKDATA       ;READ CHAR, CHK OVRR = 1
023402  004001                                4001
023404  000010                                8.
023406  004737  007266                        JSR     PC,CKDATA       ;READ CHAR, CHK OVRR STILL = 1
023412  004001                                4001
023414  000010                                8.
023416  012737  000012  002364                MOV     #12,REGNUM      ;SET REG NO. = 12
023424  012737  000200  002352                MOV     #IC,WRIBYT
023432  004737  003422                        JSR     PC,WRITLU       ;SET IC TO CLEAR RCVR
023436  004737  006672                        JSR     PC,RDRXSI       ;READ RCV SILO
023442  132737  000010  002415                BITB    #OVRR,RXWORD+1  ;CHK FOR OVRR CLEARED
023450  001407                                BEQ     8$              ;BR IF OVRR CLEARED
023452  004737  004200                        JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                        ;REPORT OVRR NOT CLEARED
023456                                          ERRDF   41,EM41,ERR7
104455                                                                          TRAP    C$ERDF
000051                                                                          .WORD   41
014040                                                                          .WORD   EM41
```

```
 (5)  023464  020224                                                                    .WORD   ERR7
7860  023466  000466                    BR      24$
7861  023470                      8$:
7862                              ;-------------------------------------------------------------------
7863                              ; CAUSE OVRR, SET MST CLR TO CLEAR IT
7864                              ;-------------------------------------------------------------------
7865  023470  004737  005212            JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
7866  023474  000226                    SYNCH
7867  023476  000311                    CRC2!CRC1!STRIP!DDCMP   ;DDCMP, NO ERR DET
7868  023500  004737  010716            JSR     PC,LODSIL       ;LOAD 64 001 CHARS INTO TX SILO
7869  023504  000001                    001
7870  023506  000100                    64.
7871  023510  004737  006752            JSR     PC,RCV1ST       ;CLOCK UNTIL FIRST DATA CHAR RCV'D
7872  023514  000030                    24.
7873  023516  004737  004726            JSR     PC,STPLU        ;CLOCK UNTIL 59 MORE RCV'D
7874  023522  000730                    472.
7875  023524  004737  010716            JSR     PC,LODSIL       ;LOAD 60 MORE INTO TX SILO
7876  023530  000001                    001
7877  023532  000074                    60.
7878  023534  004737  004726            JSR     PC,STPLU        ;CLK 8 MORE TIMES TO FORCE UNDERRUN
7879  023540  000100                    64.
7880  023542  012701  000100            MOV     #64.,R1         ;READ AND CHK 64 CHARS FROM RCV SILO
7881  023546  004737  007266      9$:   JSR     PC,CKDATA
7882  023552  000001                    001
7883  023554  000000                    0
7884  023556  005301                    DEC     R1
7885  023560  001372                    BNE     9$
7886  023562  004737  007266            JSR     PC,CKDATA       ;READ CHAR, CHK OVRR = 1
7887  023566  004001                    4001
7888  023570  000010                    8.
7889  023572  004737  007266            JSR     PC,CKDATA       ;READ CHAR, CHK OVRR STILL = 1
7890  023576  004001                    4001
7891  023600  000010                    8.
7892  023602  012737  000012  002364    MOV     #12,REGNUM      ;SET REG NO. = 12
7893  023610  004737  003276            JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
7894  023614  004737  006672            JSR     PC,RDRXSI       ;READ RCV SILO
7895  023620  132737  000010  002415    BITB    #OVRR,RXWORD+1  ;CHK FOR OVRR CLEARED
7896  023626  001406                    BEQ     24$             ;BR IF OVRR CLEARED
7897  023630  004737  004200            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
7898                            ;REPORT OVRR NOT CLEARED
7899  023634                            ERRDF   41,EM41,ERR7
 (4)  023634  104455                                                            TRAP    C$ERDF
 (5)  023636  000051                                                            .WORD   41
 (5)  023640  014040                                                            .WORD   EM41
 (5)  023642  020224                                                            .WORD   ERR7
7900  023644  004737  003276      24$:  JSR     PC,MSTCLR       ;ISSUE CLEAN UP MST CLR
7901  023650                      ENDTST
 (3)  023650                                                                L10023:
 (3)  023650  104401                                                            TRAP    C$ETST
7902
7903
7904
7905
7906
7907                              ;;*******************************************************************************
7908                              .SBTTL      TEST 3 - ABORT SEQUENCE TEST
```

```
                                ;*
                                ;* SET BIT MODE, CRC, AND ENABLE THE DEVICE FOR
                                ;* TRANSMIT AND RECEIVE. SEND 2 FLAGS AND 4 DATA CHARS (001).
                                ;* AS THE FIRST DATA CHAR IS BEING TRANSMITTED,
                                ;* SET THE ABORT BIT (REG 11).
                                ;* ON THE RECEIVER SIDE, CHECK FOR RECEPTION OF THE FIRST DATA CHAR
                                ;* AND THEN THE SETTING OF RAB AND REOM A CHAR TIME LATER.
                                ;* ALSO, CHECK FOR IACT = 0. THEN, CHECK THAT RAB
                                ;* IS CLEARED BY READING THE USYRT STATUS, TRANSMITTING A NEW MSG,
                                ;* RECEIVING THE FIRST CHAR (003) AND CHECKING FOR RAB CLEARED.
                                ;*
                                ;* REPEAT THE ABOVE SEQUENCE, SET IC, AND CHECK THAT
                                ;* THIS CLEARS RAB.
                                ;*
                                ;* REPEAT THE ABOVE SEQUENCE, ISSUE MASTER CLEAR, CHECK THAT THIS
                                ;* CLEARS RAB.
                                ;*
                                ;;*********************************************************************
023652                          BGNTST
023652                                                                  T3::
023652  012737  024202  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
                                ;-------------------------------------------------------------------
                                ; CAUSE ABORT, START NEW MSG TO CLEAR IT
                                ;-------------------------------------------------------------------
023660  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
023664  000000                          000
023666  000000                          000                     ;BIT MODE, CRC
023670  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
023674  002732                          MSG3
023676  000014                          12.
023700  004737  006752                  JSR     PC,RCV1ST       ;CLK AND RCV FIRST DATA CHAR
023704  000060                          48.
023706  004737  007266                  JSR     PC,CKDATA       ;CHK CHR = 001, CLK ABORT CHAR
023712  000001                          001
023714  000010                          8.
023716  004737  007266                  JSR     PC,CKDATA       ;CHK FOR RAB, EBLK, AND 001 CHAR
023722  003001                          RXABT!RXEBL!001
023724  000000                          0
023726  004737  006232                  JSR     PC,IACTIV       ;CHK FOR IACT = 0
023732  000000                          0
023734  004737  006752                  JSR     PC,RCV1ST       ;CLK AND RCV NEW MSG
023740  000060                          48.
023742  004737  007266                  JSR     PC,CKDATA       ;CHK CHAR = 003
023746  000003                          003
023750  000000                          0
                                ;-------------------------------------------------------------------
                                ; CAUSE ABORT, SET IC TO CLEAR IT
                                ;-------------------------------------------------------------------
023752  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
023756  000000                          000
023760  000000                          000                     ;BIT MODE, CRC
023762  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
023766  002732                          MSG3
023770  000014                          12.
023772  004737  006752                  JSR     PC,RCV1ST       ;CLK AND RCV FIRST DATA CHAR
023776  000060                          48.
```

```
 7964  024000  004737  007266          JSR     PC,CKDATA       ;CHK CHR = 001, CLK ABORT CHAR
 7965  024004  000001                  001
 7966  024006  000010                  8.
 7967  024010  004737  007266          JSR     PC,CKDATA       ;CHK FOR RAB, EBLK, AND 001 CHAR
 7968  024014  003001                  RXABT!RXEBL!001
 7969  024016  000000                  0
 7970  024020  012737  000012  002364  MOV     #12,REGNUM      ;SET REG NO. = 12
 7971  024026  012737  000200  002352  MOV     #IC,WRIBYT
 7972  024034  004737  003422          JSR     PC,WRITLU       ;SET IC TO CLEAR RCVR
 7973  024040  004737  006672          JSR     PC,RDRXSI       ;READ RCV SILO
 7974  024044  132737  000004  002415  BITB    #RAB,RXWORD+1   ;CHK FOR RAB CLEARED
 7975  024052  001407                  BEQ     8$              ;BR IF RAB CLEARED
 7976  024054  004737  004200          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 7977                          ;REPORT RAB NOT CLEARED
 7978  024060                          ERRDF   39,EM39,ERR7
   (4) 024060  104455                                                      TRAP    C$ERDF
   (5) 024062  000047                                                      .WORD   39
   (5) 024064  014004                                                      .WORD   EM39
   (5) 024066  020224                                                      .WORD   ERR7
 7979  024070  000444          BR      24$
 7980  024072                  8$:
 7981                          ;-------------------------------------------------------------------
 7982                          ;CAUSE ABORT, ISSUE MASTER CLEAR TO CLEAR IT
 7983                          ;-------------------------------------------------------------------
 7984  024072  004737  005212          JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 7985  024076  000000                  000
 7986  024100  000000                  000                     ;BIT MODE, CRC
 7987  024102  004737  010476          JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
 7988  024106  002732                  MSG3
 7989  024110  000014                  12.
 7990  024112  004737  006752          JSR     PC,RCV1ST       ;CLK AND RCV FIRST DATA CHAR
 7991  024116  000060                  48.
 7992  024120  004737  007266          JSR     PC,CKDATA       ;CHK CHR = 001, CLK ABORT CHAR
 7993  024124  000001                  001
 7994  024126  000010                  8.
 7995  024130  004737  007266          JSR     PC,CKDATA       ;CHK FOR RAB, EBLK, AND 001 CHAR
 7996  024134  003001                  RXABT!RXEBL!001
 7997  024136  000000                  0
 7998  024140  012737  000012  002364  MOV     #12,REGNUM      ;SET REG NO. = 12
 7999  024146  004737  003276          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8000  024152  004737  006672          JSR     PC,RDRXSI       ;READ RCV SILO
 8001  024156  132737  000004  002415  BITB    #RAB,RXWORD+1   ;CLK FOR RAB CLEARED
 8002  024164  001406                  BEQ     24$             ;BR IF RAB CLEARED
 8003  024166  004737  004200          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8004                          ;REPORT RAB NOT CLEARED
 8005  024172                          ERRDF   39,EM39,ERR7
   (4) 024172  104455                                                      TRAP    C$ERDF
   (5) 024174  000047                                                      .WORD   39
   (5) 024176  014004                                                      .WORD   EM39
   (5) 024200  020224                                                      .WORD   ERR7
 8006  024202  004737  003276  24$:    JSR     PC,MSTCLR       ;ISSUE MST CLR TO CLEAN UP
 8007  024206                  ENDTST
   (3) 024206                                                                      L10024:
   (3) 024206  104401                                                      TRAP    C$ETST
 8008
 8009
```

```
                                ;;*******************************************************************************
                                .SBTTL      TEST 4 - ABORT AND IDLE FLAGS TEST
                                ;*
                                ;* TRANSMIT THE SAME ABORT SEQUENCE AS IN THE PREVIOUS TEST, BUT
                                ;* WITH THE IDLE BIT SET. CHECK THAT FLAGS ARE SENT AND RECEIVED
                                ;* (NOT ABORT CHARACTERS) BY VERIFYING THAT RAB DOES
                                ;* NOT SET, AND THAT THE MESSAGE TERMINATES WITH EBLK = 1.
                                ;;*******************************************************************************
 8021  024210                   BGNTST
  (3)  024210                                                                          T4::
 8022  024210  012737  024264  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 8023  024216  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 8024  024222  000000                          000
 8025  024224  000040                          IDLE                    ;BIT MODE, NO ERROR DET
 8026  024226  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
 8027  024232  002732                          MSG3
 8028  024234  000005                          5
 8029  024236  004737  006752                  JSR     PC,RCV1ST       ;CLK AND RCV FIRST DATA CHAR
 8030  024242  000060                          48.
 8031  024244  004737  007266                  JSR     PC,CKDATA       ;CHK CHR = 001, CLK FLAG CHAR
 8032  024250  000001                          001
 8033  024252  000010                          8.
 8034  024254  004737  007266                  JSR     PC,CKDATA       ;CHK RAB = 0, EBLK = 1
 8035  024260  001001                          RXEBL!001
 8036  024262  000000                          0
 8037  024264  004737  003276          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8038  024270                          ENDTST
  (3)  024270                                                                                L10025:
  (3)  024270  104401                                                                              TRAP    C$ETST
 8039
 8040
 8041
 8042
 8043
                                ;;*******************************************************************************
                                .SBTTL      TEST 5 - TRANSMITTER UNDERRUN ERROR, IDLE ABORT CHARS, BIT MODE
                                ;*
                                ;* A MESSAGE IS INITIATED IN BIT MODE, 4 001 CHARS ARE SENT, AND THE TRANSMITTER
                                ;* IS NOT SERVICED IN RESPONSE TO THE LAST TX FLAG, WHICH CAUSES TX
                                ;* UNDERRUN ERROR TO SET. ON THE RECEIVER SIDE, CHECK THAT THE DATA
                                ;* CHAR IS RECEIVED, AND THAT 8 CYCLES LATER THE RAB BIT SETS, AND
                                ;* THE DEVICE IDLES ABORT CHARACTERS.
                                ;;*******************************************************************************
 8053  024272                   BGNTST
  (3)  024272                                                                          T5::
 8054  024272  012737  024354  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 8055  024300  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 8056  024304  000000                          000
 8057  024306  000000                          000
 8058  024310  012737  000100  002416          MOV     #TXEN,DISILO    ;SET TX ENB TO KEEP RTS HIGH
 8059  024316  004737  010716                  JSR     PC,LODSIL       ;LOAD 4 001 CHARS INTO TX SILO
 8060  024322  000001                          001
 8061  024324  000004                          4
```

```
 8062  024326  004737  006752                  JSR     PC,RCV1ST       ;CLK AND RCV FIRST CHAR
 8063  024332  000060                          48.
 8064  024334  004737  007266                  JSR     PC,CKDATA       ;CHK DATA = 001, CLOCK ABORT CHAR
 8065  024340  000001                          001
 8066  024342  000011                          9.
 8067  024344  004737  007266                  JSR     PC,CKDATA       ;CHK FOR RAB, EBLK, AND 001 CHAR
 8068  024350  003001                          RXABT!RXEBL!001
 8069  024352  000000                          0
 8070  024354  004737  003276          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8071  024360                          ENDTST
   (3) 024360                                                                    L10026:
   (3) 024360  104401                                                                TRAP    C$ETST
 8072
 8073
 8074
 8075
 8076
 8077                                  ;;*************************************************************************
 8078                                  .SBTTL      TEST 6 - RECEIVER DISABLE TEST
 8079                                  ;*
 8080                                  ;* TRANSMIT AND RECEIVE ARE ENABLED IN BIT MODE, AND 2 FLAGS
 8081                                  ;* ARE SENT, FOLLOWED BY 5 252 DATA CHARS. AFTER THE SECOND DATA CHAR HAS BEGUN
 8082                                  ;* TO BE RECEIVED, IC IS SET.
 8083                                  ;* THEN, THE PROGRAM CHECKS THAT A USYRT RCV FLAG IS NOT GENERATED, AND
 8084                                  ;* THE RECEIVER DATA PATH STOPS OPERATING IN THE MIDDLE OF THE CHAR.
 8085                                  ;;*************************************************************************
 8086  024362                          BGNTST
   (3) 024362                                                                    T6::
 8087  024362  012737  024550  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 8088  024370  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 8089  024374  000000                          000
 8090  024376  000000                          000                     ;BIT MODE, CRC
 8091  024400  004737  010716                  JSR     PC,LODSIL       ;LOAD 5 252 CHARS
 8092  024404  000252                          252
 8093  024406  000005                          5
 8094  024410  004737  006752                  JSR     PC,RCV1ST       ;CLK AND RCV FIRST DATA CHAR
 8095  024414  000060                          48.
 8096  024416  004737  004726                  JSR     PC,STPLU        ;CLK TO MIDDLE OF 2ND CHAR
 8097  024422  000004                          4
 8098  024424  012737  000012  002364          MOV     #12,REGNUM      ;SET REG NO. = 12
 8099  024432  012737  000200  002352          MOV     #IC,WRIBYT
 8100  024440  004737  003422                  JSR     PC,WRITLU       ;SET IC IN REG 12
 8101  024444  004737  006232                  JSR     PC,IACTIV       ;CHK IACT = 0
 8102  024450  000000                          0
 8103  024452  004737  005746                  JSR     PC,ISIRDY       ;CHK ICIR = 1, IRDY = 0
 8104  024456  000001                          1
 8105  024460  005037  002370                  CLR     GOODAT          ;SET EXPECTED DATA = 0
 8106  024464  005037  002372                  CLR     BADDAT
 8107  024470  004737  006672                  JSR     PC,RDRXSI       ;READ RCV SILO
 8108  024474  105737  002414                  TSTB    RXWORD          ;SEE IF SILO BITS 0-7 = 000
 8109  024500  001404                          BEQ     9$              ;BR IF YES
 8110  024502  012737  000010  002364          MOV     #10,REGNUM      ;SET REG NO. = 10
 8111  024510  000406                          BR      12$
 8112  024512  105737  002415          9$:     TSTB    RXWORD+1        ;SEE IF SILO BITS 8-11 = 000
 8113  024516  001414                          BEQ     24$             ;BR IF YES
 8114  024520  012737  000012  002364          MOV     #12,REGNUM      ;SET REG NO. = 12
```

```
 8115  024526  113737  002414  002372  12$:    MOVB    RXWORD,BADDAT   ;GET ACTUAL DATA
 8116  024534  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8117                                          ;REPORT RCV SILO NOT CLEARED BY IC
 8118  024540                                  ERRDF   46,EM46,ERR2
   (4) 024540  104455                                                          TRAP    C$ERDF
   (5) 024542  000056                                                          .WORD   46
   (5) 024544  014175                                                          .WORD   EM46
   (5) 024546  015366                                                          .WORD   ERR2
 8119  024550  004737  003276          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 8120  024554                          ENDTST
   (3) 024554                                                          L10027:
   (3) 024554  104401                                                          TRAP    C$ETST
 8121
 8122
 8123
 8124
 8125
 8126                                  ;;*************************************************************************
 8127                                  .SBTTL      TEST 7 - ASSEMBLED BIT COUNT TEST
 8128                                  ;*
 8129                                  ;* THE FOLLOWING SEQUENCE IS PERFORMED 8 TIMES, EACH TIME USING A
 8130                                  ;* DIFFERENT TX CHAR LENGTH (FROM 2 TO 8 BITS) AND A RCV CHAR LENGTH = 8
 8131                                  ;* BITS :
 8132                                  ;* A MESSAGE IS INITIATED IN BIT MODE, NO CRC.
 8133                                  ;* 2 FLAGS ARE SENT, FOLLOWED BY 3 000 DATA CHARACTERS AND A
 8134                                  ;* TERMINATING FLAG. AFTER THE RECEIVER HAS RECEIVED THE MESSAGE, AX0-16
 8135                                  ;* IS READ TO RETRIEVE THE ASSEMBLED BIT COUNT. THIS COUNT IS CHECKED TO INSURE
 8136                                  ;* THAT IT IS CORRECT FOR THE TX CHAR LENGTH USED IN THAT TRANSMISSION.
 8137                                  ;;*************************************************************************
 8138  024556                          BGNTST
   (3) 024556                                                          T7::
 8139  024556  012737  025314  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 8140  024564  004737  005212                  JSR     PC,INITRN       ;FIND OUT WHICH USYRT CHIP
 8141  024570  000000                          0
 8142  024572  000000                          0
 8143  024574  012701  000100                  MOV     #TXLEN1,R1      ;SET INITIAL TX LENGTH TO 2 BITS
 8144  024600  004737  003276          6$:     JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8145  024604  004737  010304                  JSR     PC,SETUP        ;PROGRAM THE USYRT
 8146  024610  000000                          000
 8147  024612  000300                          CRC2!CRC1
 8148  024614  000000                          000
 8149  024616  000000                          000
 8150  024620  012737  000014  002364          MOV     #14,REGNUM      ;SET REG NO. = 14
 8151  024626  012737  000140  002352          MOV     #TXEN!DISSI,WRIBYT
 8152  024634  004737  003422                  JSR     PC,WRITLU       ;SET TXEN AND DISSI IN REG 14
 8153  024640  012737  000140  002416          MOV     #TXEN!DISSI,DISILO      ;SET DISABLE SILO FLAG
 8154  024646  012737  000012  002364          MOV     #12,REGNUM      ;SET LU REG NO. = 12
 8155  024654  112737  000040  002352          MOVB    #LULP,WRIBYT
 8156  024662  004737  003422                  JSR     PC,WRITLU       ;SET LULP IN REG 12
 8157  024666  012737  000002  002366          MOV     #2,AXNUM        ;SET AX BYTE NO. = 2
 8158  024674  105037  002360                  CLRB    WAX15
 8159  024700  112737  000001  002362          MOVB    #TSOM,WAX16
 8160  024706  004737  003764                  JSR     PC,WRITAX       ;LOAD SOM CHAR
 8161  024712  005004                          CLR     R4              ;INIT COUNTER
 8162  024714  012737  000011  002364          MOV     #11,REGNUM      ;SET REG NO. = 11
 8163  024722  004737  004726          7$:     JSR     PC,STPLU        ;CLOCK LU FOR A CYCLE
```

```
 8164  024726  000001                          1
 8165  024730  004737  003344                  JSR     PC,READLU       ;READ REG 11
 8166  024734  132737  000100  002350          BITB    #OACT,REDBYT    ;SEE IF OACT SET YET
 8167  024742  001014                          BNE     10$             ;BR IF OACT SET
 8168  024744  005204                          INC     R4              ;INCR COUNTER
 8169  024746  020427  000004                  CMP     R4,#4           ;SEE IF COUNT TOO BIG
 8170  024752  002763                          BLT     7$              ;BR IF NOT
 8171  024754  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8172                                  ;REPORT OACT NOT SET
 8173  024760                                  ERRDF   11,EM11,ERR7
   (4) 024760  104455                                                                  TRAP    C$ERDF
   (5) 024762  000013                                                                  .WORD   11
   (5) 024764  013400                                                                  .WORD   EM11
   (5) 024766  020224                                                                  .WORD   ERR7
 8174  024770  000137  025314                  JMP     24$
 8175  024774  004737  003764          10$:    JSR     PC,WRITAX       ;LOAD ANOTHER SOM CHAR
 8176  025000  004737  004726                  JSR     PC,STPLU        ;CLK FIRST FLAG
 8177  025004  000010                          8.
 8178  025006  105037  002362                  CLRB    WAX16
 8179  025012  004737  003764                  JSR     PC,WRITAX       ;LOAD FIRST 000 CHAR
 8180  025016  004737  004726                  JSR     PC,STPLU        ;CLK SECOND FLAG
 8181  025022  000010                          8.
 8182  025024  004737  003764                  JSR     PC,WRITAX       ;LOAD SECOND 000 CHAR
 8183  025030  004737  004726                  JSR     PC,STPLU        ;CLK FIRST 000 CHAR
 8184  025034  000010                          8.
 8185  025036  012737  000006  002366          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX 3
 8186  025044  010137  002362                  MOV     R1,WAX16        ;GET TX CHAR LENGTH
 8187  025050  004737  003764                  JSR     PC,WRITAX       ;SET TX CHAR LENGTH
 8188  025054  012737  000002  002366          MOV     #2,AXNUM        ;SET AX BYTE NO. = 2
 8189  025062  105037  002362                  CLRB    WAX16
 8190  025066  005737  002420                  TST     CHPTYP          ;SEE IF SIG USYRT
 8191  025072  001403                          BEQ     5$              ;BR IF YES
 8192  025074  112737  000002  002362          MOVB    #TEOM,WAX16     ;SET TEOM WITH LAST DATA CHAR
 8193  025102  004737  003764          5$:     JSR     PC,WRITAX       ;LOAD 3RD 000 CHAR
 8194  025106  004737  004726                  JSR     PC,STPLU        ;CLK 2ND 000 CHAR
 8195  025112  000010                          8.
 8196  025114  005737  002420                  TST     CHPTYP          ;SEE IF SIG USYRT
 8197  025120  001005                          BNE     16$             ;BR IF NOT
 8198  025122  112737  000002  002362          MOVB    #TEOM,WAX16
 8199  025130  004737  003764                  JSR     PC,WRITAX       ;LOAD AN EOM CHAR
 8200  025134  012737  000001  002366  16$:    MOV     #1,AXNUM        ;SET AX BYTE NO. = 1
 8201  025142  005003                          CLR     R3
 8202  025144  004737  003576          12$:    JSR     PC,READAX       ;READ AX0
 8203  025150  132737  000002  002356          BITB    #REOM,RAX16     ;CHK FOR REOM = 1
 8204  025156  001016                          BNE     14$             ;BR IF YES
 8205  025160  004737  004726                  JSR     PC,STPLU        ;CLOCK LU FOR A CYCLE
 8206  025164  000001                          1
 8207  025166  005203                          INC     R3              ;INCR COUNT
 8208  025170  020327  000023                  CMP     R3,#19.         ;SEE IF COUNT TOO BIG
 8209  025174  002763                          BLT     12$             ;BR IF NOT
 8210  025176  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8211                                  ;REPORT REOM NOT SET
 8212  025202                                  ERRDF   31,EM31,ERR10
   (4) 025202  104455                                                                  TRAP    C$ERDF
   (5) 025204  000037                                                                  .WORD   31
   (5) 025206  013647                                                                  .WORD   EM31
```

```
  (5)  025210  021364                                                                    .WORD   ERR10
8213   025212  000440                    BR      24$
8214   025214  013702  002356    14$:    MOV     RAX16,R2        ;GET AX0-16 CONTENTS
8215   025220  042702  000217            BIC     #217,R2         ;MASK OFF ALL BUT ASSEMB BIT COUNT
8216   025224  006102                    ROL     R2
8217   025226  120201                    CMPB    R2,R1           ;CHK FOR CORRECT ASSEMB BIT COUNT
8218   025230  001421                    BEQ     9$              ;BR IF MATCH
8219   025232  010137  002370            MOV     R1,GOODAT       ;SET EXPECTED DATA
8220   025236  006237  002370            ASR     GOODAT
8221   025242  152737  000002  002370    BISB    #REQM,GOODAT
8222   025250  013737  002356  002372    MOV     RAX16,BADDAT    ;SET ACTUAL DATA
8223   025256  004737  004200            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8224                                ;REPORT ASSEMB BIT COUNT INCORRECT
8225   025262                            ERRDF   47,EM47,ERR3
  (4)  025262  104455                                                                    TRAP    C$ERDF
  (5)  025264  000057                                                                    .WORD   47
  (5)  025266  014230                                                                    .WORD   EM47
  (5)  025270  015674                                                                    .WORD   ERR3
8226   025272  000410                    BR      24$
8227   025274  005701            9$:     TST     R1              ;SEE IF ALL DONE YET
8228   025276  001406                    BEQ     24$             ;BR IF YES
8229   025300  062701  000040            ADD     #TXLEN0,R1      ;INCR TX LENGTH
8230   025304  042701  000400            BIC     #400,R1         ;MASK OFF OVERFLOW IF 8 BITS
8231   025310  000137  024600            JMP     6$              ;PROCEED
8232   025314  005037  002416    24$:    CLR     DISILO
8233   025320  004737  003276            JSR     PC,MSTCLR       ;ISSUE MASTER CLR TO CLEAN UP
8234   025324                    ENDTST
  (3)  025324                                                                    L10030:
  (3)  025324  104401                                                                    TRAP    C$ETST
8235
8236
8237
8238
8239
8240                              ;;*************************************************************************
8241                              .SBTTL      TEST 8 - SECONDARY STATION ADDRESS BIT TEST
8242                              ;*
8243                              ;* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
8244                              ;* BIT MODE, AND THE SECA BIT (REG 17) IS SET.
8245                              ;* 2 FLAGS ARE SENT, FOLLOWED BY 252, 000, AND A TERMINATING FLAG.
8246                              ;* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
8247                              ;* RECEIVED.
8248                              ;*
8249                              ;* NEXT, THE SECONDARY STATION ADDRESS BITS IN AX2-15 ARE LOADED
8250                              ;* WITH THE FIRST WORD OF DATA PATTERN T . 2 FLAGS ARE SENT,
8251                              ;* FOLLOWED BY THE FIRST WORD OF DATA PATTERN T, A 000 CHAR,
8252                              ;* AND A TERMINATING FLAG.
8253                              ;* THEN, THE RCV'D DATA IS CHECKED TO MAKE SURE THAT THE SEC STATION
8254                              ;* ADDRESS IS RCV'D AS THE FIRST DATA CHAR, FOLLOWED BY 000.
8255                              ;*
8256                              ;* THEN, THE SUBTEST IS REPEATED FOR EACH OF THE REMAINING WORDS OF
8257                              ;* DATA PATTERN T.
8258                              ;*   PATTERN T = 000,125,252,176,177
8259                              ;;*************************************************************************
8260   025326                     BGNTST
  (3)  025326                                                                    T8::
```

```
                        ;-------------------------------------------------------------------
                        ; SEND MSG WITH INVALID SEC STA ADRS
                        ;-------------------------------------------------------------------
                                BGNSUB
                                                                                T8.1:
                                                                                        TRAP    C$BSUB
025326                          MOV     #3$,RETADR      ;SET SUBTEST EXIT ADRS FOR ERRORS
025326                          JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
025326                          000                     ;SEC ADRS = 000
025330                          SECA                    ;BIT MODE, CRC, SEC ADRS MODE
025332                          JSR     PC,LODSIL       ;LOAD 252 INTO TX SILO
025336                          252
025340                          1
025342                          JSR     PC,LODSIL       ;LOAD 000 DATA INTO TX SILO
025346                          000
025350                          1
025352                          JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
025354                          TXEOM
025356                          2
025360                          JSR     PC,STPLU        ;TRANSMIT THE MSG
025362                          48.
025364                          JSR     PC,IACTIV       ;CHK IACT = 0
025366                          0
025370                          JSR     PC,STPLU        ;CLOCK 8 MORE CYCLES
025372                          8.
025374                          JSR     PC,IACTIV       ;CHK IACT = 0
025376                          0
                        3$:
025426                          ENDSUB
                                                                                L10032:
                                                                                        TRAP    C$ESUB
                        ;-------------------------------------------------------------------
                        ; SEND MSG'S WITH VALID SEC ADRS'S FROM PAT T
                        ;-------------------------------------------------------------------
025430                  MOV     #PATT,R1        ;INIT DATA PATTERN POINTER
                        A11:
025434                          BGNSUB
                                                                                T8.2:
                                                                                        TRAP    C$BSUB
025436                          MOV     #24$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
025444                          MOVB    (R1),5$         ;SET SEC ADRS
025450                          MOVB    (R1),6$         ;SET FIRST DATA CHAR
025454                          MOVB    (R1),9$         ;SET EXPECTED DATA CHAR
025460                          JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
025464                  5$:     .WORD   0
025466                          SECA                    ;BIT MODE, CRC, SEC ADRS MODE
025470                          JSR     PC,LODSIL       ;LOAD 1ST DATA CHAR INTO TX SILO
025474                  6$:     .WORD   0
025476                          1
025500                          JSR     PC,LODSIL       ;LOAD A 000 CHAR INTO TX SILO
025504                          000
025506                          1
025510                          JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
025514                          TXEOM
025516                          2
025520                          JSR     PC,RCV1ST       ;CLOCK AND RCV FIRST DATA CHAR
```

```
 8311  025524  000060                          48.
 8312  025526  004737  007266                  JSR     PC,CKDATA       ;CHK FOR CORRECT RCV'D SEC STA ADRS
 8313  025532  000000              9$:         .WORD   0
 8314  025534  000011                          9.
 8315  025536  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK 000 CHAR, EBLK=1, BCC=0
 8316  025542  101000                          CRCCHK!RXEBL!000
 8317  025544  000000                          0
 8318  025546                          24$:
 8319  025546                                  ENDSUB
  (3)  025546                                                                     L10033:
  (3)  025546  104403                                                                     TRAP    C$ESUB
 8320  025550  005201                          INC     R1              ;INCR PATTERN POINTER
 8321  025552  020127  002667                  CMP     R1,#ENDPAT      ;SEE IF ALL DONE YET
 8322  025556  103726                          BLO     A11             ;BR IF NO
 8323  025560  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8324  025564                          ENDTST
  (3)  025564                                                                     L10031:
  (3)  025564  104401                                                                     TRAP    C$ETST
 8325
 8326
 8327
 8328
 8329
 8330                                  ;;*************************************************************************
 8331                                  .SBTTL      TEST 9 - RDALL (ALL PARTIES ADDRESS) BIT TEST
 8332                                  ;*
 8333                                  ;* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
 8334                                  ;* BIT MODE, AND THE SECA BIT IS SET.
 8335                                  ;* 2 FLAGS ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
 8336                                  ;* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
 8337                                  ;* RECEIVED.
 8338                                  ;* NEXT, THE RDALL BIT IN REG 17 IS SET TO 1. 2 FLAGS
 8339                                  ;* ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
 8340                                  ;* THEN, THE REC'D DATA IS CHECKED TO MAKE SURE THAT 377
 8341                                  ;* IS REC'D AS THE FIRST DATA CHAR, FOLLOWED BY 125.
 8342                                  ;;*************************************************************************
 8343  025566                          BGNTST
  (3)  025566                                                                     T9::
 8344                                  ;-------------------------------------------------------------
 8345                                  ; SET SEC ADR = 000, SEND ADR = 377, WITH RDALL = 0
 8346                                  ;-------------------------------------------------------------
 8347  025566                                  BGNSUB
  (3)  025566                                                                     T9.1:
  (3)  025566  104402                                                                     TRAP    C$BSUB
 8348  025570  012737  025666  002346          MOV     #3$,RETADR      ;SET SUBTEST EXIT ADRS FOR ERRORS
 8349  025576  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 8350  025602  000000                          000                     ;SEC ADRS = 000
 8351  025604  000020                          SECA                    ;BIT MODE, CRC, SEC ADRS MODE
 8352  025606  004737  010716                  JSR     PC,LODSIL       ;LOAD 377 INTO TX SILO
 8353  025612  000377                          377
 8354  025614  000001                          1
 8355  025616  004737  010716                  JSR     PC,LODSIL       ;LOAD 125 DATA INTO TX SILO
 8356  025622  000125                          125
 8357  025624  000001                          1
 8358  025626  004737  010716                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
 8359  025632  001000                          TXEOM
```

```
 8360  025634  000002                        2
 8361  025636  004737  004726                JSR     PC,STPLU        ;TRANSMIT THE MSG
 8362  025642  000060                        48.
 8363  025644  004737  006232                JSR     PC,IACTIV       ;CHK IACT = 0
 8364  025650  000000                        0
 8365  025652  004737  004726                JSR     PC,STPLU        ;CLOCK 8 MORE CYCLES
 8366  025656  000010                        8.
 8367  025660  004737  006232                JSR     PC,IACTIV       ;CHK IACT = 0
 8368  025664  000000                        0
 8369  025666                          3$:
 8370  025666                                ENDSUB
   (3) 025666                                                                L10035:
   (3) 025666  104403                                                              TRAP    C$ESUB
 8371                                  ;-----------------------------------------------------------------
 8372                                  ; SET SEC ADR = 000, SEND ADR = 377, WITH RDALL = 1
 8373                                  ;-----------------------------------------------------------------
 8374  025670                                BGNSUB
   (3) 025670                                                                T9.2:
   (3) 025670  104402                                                              TRAP    C$BSUB
 8375  025672  012737  025766  002346        MOV     #24$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
 8376  025700  004737  005212                JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 8377  025704  000000                        000                     ;SEC ADRS = 000
 8378  025706  000024                        SECA!RDALL              ;BIT MODE, CRC, SEC ADRS MODE, RDALL
 8379  025710  004737  010716                JSR     PC,LODSIL       ;LOAD 1ST DATA CHAR INTO TX SILO
 8380  025714  000377                        377
 8381  025716  000001                        1
 8382  025720  004737  010716                JSR     PC,LODSIL       ;LOAD A 125 CHAR INTO TX SILO
 8383  025724  000125                        125
 8384  025726  000001                        1
 8385  025730  004737  010716                JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
 8386  025734  001000                        TXEOM
 8387  025736  000002                        2
 8388  025740  004737  006752                JSR     PC,RCV1ST       ;CLOCK AND RCV FIRST DATA CHAR
 8389  025744  000060                        48.
 8390  025746  004737  007266                JSR     PC,CKDATA       ;CHK FOR 377 CHAR RCV'D
 8391  025752  000377                        377
 8392  025754  000010                        8.
 8393  025756  004737  007266                JSR     PC,CKDATA       ;READ AND CHK 125 CHAR, EBLK=1, BCC=0
 8394  025762  101125                        CRCCHK!RXEBL!125
 8395  025764  000000                        0
 8396  025766                          24$:
 8397  025766                                ENDSUB
   (3) 025766                                                                L10036:
   (3) 025766  104403                                                              TRAP    C$ESUB
 8398  025770                          ENDTST
   (3) 025770                                                                L10034:
   (3) 025770  104401                                                              TRAP    C$ETST
 8399
 8400
 8401
 8402
 8403
 8404                                  ;;*****************************************************************************
 8405                                  .SBTTL      TEST 10 - INSERT ERROR (IERR) BIT TEST - CHAR MODE, NO CRC
 8406                                  ;*
 8407                                  ;* THE LINE UNIT IS PLACED IN DDCMP MODE WITH NO ERROR DETECTION, AND 2
```

```
                        ;* SYNCHS, A 000 CHAR, A 377 CHAR, AND 2 SYNCHS ARE LOADED INTO THE
                        ;* TRANSMITTER SILO. THEN, THE LU IS CLOCKED UNTIL THE 2ND BIT OF THE 000
                        ;* CHAR IS ABOUT TO BE SENT AND THE IERR BIT IS SET FOR A CLOCK TIME AND
                        ;* THEN CLEARED. IN THE SAME WAY, IERR IS SET PRIOR TO THE SENDING OF THE 4TH
                        ;* AND 5TH BITS OF THE 000 CHAR. IT IS ALSO SET FOR THE SENDING OF THE FIRST
                        ;* 4 BITS OF THE 377 CHAR. THE PROGRAM READS THE FIRST RCV'D CHAR FROM AX0
                        ;* AND CHECKS IT TO BE 032, AND READS THE 2ND CHAR AND CHECKS IT TO BE 377.
                        ;* THEN, A MASTER CLEAR IS DONE TO IDLE THE DEVICE.
                        ;;*************************************************************************************
025772                  BGNTST
025772                                                                          T10::
026120  012737  026120  002346          MOV     #15$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
026000  004737  005212                  JSR     PC,INITRN       ;LOAD 2 SOM'S, CLOCK THEM INTO USYRT
026004  000226                          SYNCH
026006  000011                          STRIP!DDCMP
026010  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
026014  002722                          MSG2+4
026016  000004                          4
026020  004737  004726                  JSR     PC,STPLU        ;CLOCK LU UNTIL 2ND BIT OF 000 CHAR
026024  100021                          CHPCHK!17.
026026  004737  007164                  JSR     PC,STPERR       ;SET IERR 1 CYCLE
026032  000011                          STRIP!DDCMP
026034  000001                          1
026036  004737  004726                  JSR     PC,STPLU        ;CLOCK LU UNTIL 4TH BIT OF 000 CHAR
026042  000001                          1
026044  004737  007164                  JSR     PC,STPERR       ;SET IERR FOR 2 CYCLES
026050  000011                          STRIP!DDCMP
026052  000002                          2
026054  004737  004726                  JSR     PC,STPLU        ;CLOCK LU UNTIL 1ST BIT OF 377 CHAR
026060  000003                          3
026062  004737  007164                  JSR     PC,STPERR       ;SET IERR FOR 4 CYCLES
026066  000011                          STRIP!DDCMP
026070  000004                          4
026072  004737  006752                  JSR     PC,RCV1ST       ;CLOCK AND RCV 1ST CHAR
026076  000014                          12.
026100  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE 1ST CHAR TO 032
026104  000032                          032
026106  000010                          8.
026110  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE 2ND CHAR TO 377
026114  000377                          377                     ;377 CHAR
026116  000000                          0
026120  004737  003276          15$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
026124                          ENDTST
026124                                                                  L10037:
026124  104401                                                                  TRAP    C$ETST


                        ;;*************************************************************************************
                        .SBTTL      TEST 11 - SWITCH PACK PRINTOUT AND TEST
                        ;*
                        ;* - READ AND PRINT SWITCH PACK #1 :
                        ;* THE PROGRAM READS REG 11 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
                        ;* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
```

```
                        ;* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
                        ;* SWITCHES ARE IN BITS 1,2,3,5.
                        ;*
                        ;* - READ AND PRINT SWITCH PACK #2 :
                        ;* THE PROGRAM READS REG 15 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
                        ;* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
                        ;* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
                        ;* SWITCHES ARE IN BITS 0-7.
                        ;*
                        ;* - READ AND PRINT SWITCH PACK #3 :
                        ;* THE PROGRAM READS REG 16 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
                        ;* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
                        ;* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). THE
                        ;* SWITCHES ARE IN BITS 0-7.
                        ;;*************************************************************************************
026126                  BGNTST
(3)  026126                                                                     T11::
                        ;-------------------------------------------------------------------------------
                        ; READ AND PRINT SWITCH PACK #1. IF DESIRED
                        ;-------------------------------------------------------------------------------
                                BGNSUB
(3)  026126                                                                     T11.1:
(3)  026126  104402                                                                     TRAP    C$BSUB
026130  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
026134  012737  000011  002364          MOV     #11,REGNUM      ;SET LU REG NO. = 11
026142  004737  003344                  JSR     PC,READLU       ;READ LU REG 11
026146  142737  000321  002350          BICB    #321,REDBYT     ;MASK OFF NON-SWITCH BITS
026154  023727  002400  000001          CMP     FRSPAS,#1       ;SEE IF IN FIRST PASS AFTER LOAD
026162  001403                          BEQ     3$              ;BR IF YES
026164  005737  002260                  TST     PRNFLG          ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
026170  001424                          BEQ     4$              ;BR IF NOT
026172                  3$:
                        ;PRINT DEVICE ADDRESS
026172                          PRINTF  #FMT18,MPCSR
(8)  026172  013746  002436                                                             MOV     MPCSR,-(SP)
(7)  026176  012746  013042                                                             MOV     #FMT18,-(SP)
(6)  026202  012746  000002                                                             MOV     #2,-(SP)
(3)  026206  010600                                                                     MOV     SP,R0
(4)  026210  104417                                                                     TRAP    C$PNTF
(4)  026212  062706  000006                                                             ADD     #6,SP
                        ;PRINT SWITCH PACK #1
026216                          PRINTF  #FMT12,REDBYT
(8)  026216  013746  002350                                                             MOV     REDBYT,-(SP)
(7)  026222  012746  012617                                                             MOV     #FMT12,-(SP)
(6)  026226  012746  000002                                                             MOV     #2,-(SP)
(3)  026232  010600                                                                     MOV     SP,R0
(4)  026234  104417                                                                     TRAP    C$PNTF
(4)  026236  062706  000006                                                             ADD     #6,SP
026242  005737  002262          4$:     TST     SWIFLG          ;SEE IF TEST IS ALLOWED
026246  001420                          BEQ     6$              ;BR IF NOT
026250  123737  002350  002454          CMPB    REDBYT,LUSWI1   ;COMPARE SWITCHES TO EXPECTED
026256  001414                          BEQ     6$              ;BR IF MATCH
026260  013737  002454  002370          MOV     LUSWI1,GOODAT   ;SET EXPECTED DATA
026266  013737  002350  002372          MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
026274  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT SWITCH PACK #1 INCORRECT
```

```
                        ERRDF   43,EM43,ERR2
                                                                TRAP    C$ERDF
                                                                .WORD   43
                                                                .WORD   EM43
                                                                .WORD   ERR2
                    6$:
                        ENDSUB
                                                              L10041:
                                                                TRAP    C$ESUB
                    ;-------------------------------------------------------------------------------
                    ; READ AND PRINT SWITCH PACK #2, IF DESIRED
                    ;-------------------------------------------------------------------------------
                        BGNSUB
                                                              T11.2:
                                                                TRAP    C$BSUB
                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                        MOV     #15,REGNUM      ;SET LU REG NO. = 15
                        JSR     PC,READLU       ;READ LU REG 15
                        CMP     FRSPAS,#1       ;SEE IF IN FIRST PASS AFTER LOAD
                        BEQ     3$              ;BR IF YES
                        TST     PRNFLG          ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
                        BEQ     4$              ;BR IF NOT
                    3$:
                        PRINTF  #FMT13,REDBYT
                                                                MOV     REDBYT,-(SP)
                                                                MOV     #FMT13,-(SP)
                                                                MOV     #2,-(SP)
                                                                MOV     SP,R0
                                                                TRAP    C$PNTF
                                                                ADD     #6,SP
                    4$:     TST     SWIFLG          ;SEE IF TEST IS ALLOWED
                            BEQ     6$              ;BR IF NOT
                            CMPB    REDBYT,LUSWI2   ;COMPARE SWITCHES TO EXPECTED
                            BEQ     6$              ;BR IF MATCH
                            MOV     LUSWI2,GOODAT   ;SET EXPECTED DATA
                            MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
                            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                    ;REPORT SWITCH PACK #2 INCORRECT
                            ERRDF   44,EM44,ERR2
                                                                TRAP    C$ERDF
                                                                .WORD   44
                                                                .WORD   EM44
                                                                .WORD   ERR2
                    6$:
                        ENDSUB
                                                              L10042:
                                                                TRAP    C$ESUB
                    ;-------------------------------------------------------------------------------
                    ; READ AND PRINT SWITCH PACK #3, IF DESIRED
                    ;-------------------------------------------------------------------------------
                        BGNSUB
                                                              T11.3:
                                                                TRAP    C$BSUB
                        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
                        MOV     #16,REGNUM      ;SET LU REG NO. = 16
                        JSR     PC,READLU       ;READ LU REG 16
```

```
 8536  026464  023727  002400  000001          CMP     FRSPAS,#1       ;SEE IF IN FIRST PASS AFTER LOAD
 8537  026472  001403                          BEQ     3$              ;BR IF YES
 8538  026474  005737  002260                  TST     PRNFLG          ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
 8539  026500  001412                          BEQ     4$              ;BR IF NOT
 8540  026502                          3$:
 8541  026502                          PRINTF  #FMT14,REDBYT
   (8) 026502  013746  002350                                                  MOV     REDBYT,-(SP)
   (7) 026506  012746  012727                                                  MOV     #FMT14,-(SP)
   (6) 026512  012746  000002                                                  MOV     #2,-(SP)
   (3) 026516  010600                                                          MOV     SP,R0
   (4) 026520  104417                                                          TRAP    C$PNTF
   (4) 026522  062706  000006                                                  ADD     #6,SP
 8542  026526  005737  002262          4$:     TST     SWIFLG          ;SEE IF TEST IS ALLOWED
 8543  026532  001420                          BEQ     6$              ;BR IF NOT
 8544  026534  123737  002350  002460          CMPB    REDBYT,LUSWI3   ;COMPARE SWITCHES TO EXPECTED
 8545  026542  001414                          BEQ     6$              ;BR IF MATCH
 8546  026544  013737  002460  002370          MOV     LUSWI3,GOODAT   ;SET EXPECTED DATA
 8547  026552  013737  002350  002372          MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
 8548  026560  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8549                          ;REPORT SWITCH PACK #3 INCORRECT
 8550  026564                          ERRDF   45,EM45,ERR2
   (4) 026564  104455                                                          TRAP    C$ERDF
   (5) 026566  000055                                                          .WORD   45
   (5) 026570  014150                                                          .WORD   EM45
   (5) 026572  015366                                                          .WORD   ERR2
 8551  026574                          6$:
 8552  026574                          ENDSUB
   (3) 026574                                                          L10043:
   (3) 026574  104403                                                          TRAP    C$ESUB
 8553  026576                          ENDTST
   (3) 026576                                                          L10040:
   (3) 026576  104401                                                          TRAP    C$ETST
 8554
 8555
 8556
 8557
 8558
 8559                          ;;*****************************************************************************
 8560                          .SBTTL      TEST 12 - REG AX3-15 PRINTOUT
 8561                          ;*
 8562                          ;* IN THIS TEST, REG AX3-15 IS READ AND THE CONTENTS PRINTED OUT IF DESIRED BY
 8563                          ;* THE OPERATOR, AS INDICATED IN THE SOFTWARE P-TABLE. THE DEFAULT IS TO NOT
 8564                          ;* PRINT THE REG.
 8565                          ;;*****************************************************************************
 8566  026600                  BGNTST
   (3) 026600                                                          T12::
 8567  026600  004737  003276          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8568  026604  142777  000010  153626  BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
 8569  026612  012737  000006  002366  MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3-15
 8570  026620  004737  003576          JSR     PC,READAX       ;READ AX3-15,AX3-16
 8571  026624  023727  002400  000001  CMP     FRSPAS,#1       ;SEE IF FIRST PASS AFTER LOAD
 8572  026632  001403                  BEQ     3$              ;BR IF NOT
 8573  026634  005737  002260          TST     PRNFLG          ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
 8574  026640  001424                  BEQ     4$              ;BR IF NOT
 8575  026642                  3$:
 8576                          ;PRINT DEVICE ADDRESS
```

```
8577  026642                          PRINTF  #FMT18,MPCSR
 (8)  026642  013746  002436                                        MOV     MPCSR,-(SP)
 (7)  026646  012746  013042                                        MOV     #FMT18,-(SP)
 (6)  026652  012746  000002                                        MOV     #2,-(SP)
 (3)  026656  010600                                                MOV     SP,R0
 (4)  026660  104417                                                TRAP    C$PNTF
 (4)  026662  062706  000006                                        ADD     #6,SP
8578                            ;PRINT AX3-15
8579  026666                          PRINTF  #FMT15,RAX15
 (8)  026666  013746  002354                                        MOV     RAX15,-(SP)
 (7)  026672  012746  012773                                        MOV     #FMT15,-(SP)
 (6)  026676  012746  000002                                        MOV     #2,-(SP)
 (3)  026702  010600                                                MOV     SP,R0
 (4)  026704  104417                                                TRAP    C$PNTF
 (4)  026706  062706  000006                                        ADD     #6,SP
8580  026712                    4$:
8581  026712                    ENDTST
 (3)  026712                                                L10044:
 (3)  026712  104401                                                TRAP    C$ETST
8582
8583
8584
8585
8586
8587                            ;;**************************************************************************************
8588                            .SBTTL      TEST 13 - CRC GENERATION TEST
8589                            ;*
8590                            ;* - CRC-16, CHAR MODE:
8591                            ;* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE WITH CRC-16 SELECTED -
8592                            ;* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
8593                            ;* TO CLOCK THE DATA.    AT THE END OF THE MESSAGE THE
8594                            ;* PROGRAM CHECKS FOR BCC = 1 (IN REG 12) INDICATING NO ERROR.
8595                            ;*
8596                            ;* - CRC-CCITT - 1'S PRESET:
8597                            ;* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT
8598                            ;* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
8599                            ;*
8600                            ;* - CRC-CCITT - 0'S PRESET:
8601                            ;* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT
8602                            ;* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
8603                            ;;**************************************************************************************
8604  026714                    BGNTST
 (3)  026714                                                        T13::
8605                            ;-------------------------------------------------------------------------------
8606                            ; CRC 16, CHAR MODE
8607                            ;-------------------------------------------------------------------------------
8608  026714  012737  027244  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
8609  026722  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
8610  026726  004737  010304                  JSR     PC,SETUP        ;PROGRAM THE USYRT
8611  026732  000226                          SYNCH
8612  026734  000011                          STRIP!DDCMP
8613  026736  000000                          000
8614  026740  000000                          000
8615  026742  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
8616  026746  002670                          MSG1
8617  026750  000011                          9.
```

```
 8618  026752  004737  004726          JSR     PC,STPLU        ;CLOCK THE MSG
 8619  026756  000136                  94.
 8620  026760  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000
 8621  026764  000000                  000
 8622  026766  000000                  0
 8623  026770  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
 8624  026774  000125                  125
 8625  026776  000000                  0
 8626  027000  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
 8627  027004  000252                  252
 8628  027006  000000                  0
 8629  027010  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
 8630  027014  000377                  377
 8631  027016  000000                  0
 8632  027020  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 1
 8633  027024  100400                  CRCCHK!400
 8634  027026  000000                  0
 8635
 8636                          ;-----------------------------------------------------------
 8637                          ; CRC-CCITT-1'S PRESET, BIT MODE
 8638                          ;-----------------------------------------------------------
 8639  027030  004737  003276          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8640  027034  004737  010304          JSR     PC,SETUP        ;PROGRAM THE USYRT
 8641  027040  000000                  000
 8642  027042  000000                  000
 8643  027044  000000                  000
 8644  027046  000000                  000
 8645  027050  004737  010476          JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
 8646  027054  002670                  MSG1
 8647  027056  000011                  9.
 8648  027060  004737  004726          JSR     PC,STPLU        ;CLOCK THE MSG
 8649  027064  000146                  102.
 8650  027066  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000
 8651  027072  000000                  000
 8652  027074  000000                  0
 8653  027076  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
 8654  027102  000125                  125
 8655  027104  000000                  0
 8656  027106  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
 8657  027112  000252                  252
 8658  027114  000000                  0
 8659  027116  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
 8660  027122  000377                  377
 8661  027124  000000                  0
 8662  027126  004737  007266          JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
 8663  027132  101000                  CRCCHK!1000
 8664  027134  000000                  0
 8665
 8666
 8667                          ;-----------------------------------------------------------
 8668                          ; CRC-CCITT-0'S PRESET, BIT MODE
 8669                          ;-----------------------------------------------------------
 8670  027136  004737  003276          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 8671  027142  004737  010304          JSR     PC,SETUP        ;PROGRAM THE USYRT
 8672  027146  000000                  000
 8673  027150  000100                  CRC1
```

```
 8674  027152  000000                          000
 8675  027154  000000                          000
 8676  027156  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
 8677  027162  002670                          MSG1
 8678  027164  000011                          9.
 8679  027166  004737  004726                  JSR     PC,STPLU        ;CLOCK THE MSG
 8680  027172  000146                          102.
 8681  027174  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000
 8682  027200  000000                          000
 8683  027202  000000                          0
 8684  027204  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
 8685  027210  000125                          125
 8686  027212  000000                          0
 8687  027214  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
 8688  027220  000252                          252
 8689  027222  000000                          0
 8690  027224  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
 8691  027230  000377                          377
 8692  027232  000000                          0
 8693  027234  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
 8694  027240  101000                          CRCCHK!1000
 8695  027242  000000                          0
 8696
 8697  027244  004737  003276          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 8698  027250                          ENDTST
  (3)  027250                                                                  L10045:
  (3)  027250  104401                                                                 TRAP    C$ETST
 8699
 8700
 8701
 8702
 8703
 8704                                  ;;*****************************************************************************
 8705                                  .SBTTL      TEST 14 - CRC ERROR DETECTION TEST
 8706                                  ;*
 8707                                  ;* - CRC-16, CHAR MODE :
 8708                                  ;* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE, WITH CRC-16 SELECTED -
 8709                                  ;* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
 8710                                  ;* TO CLOCK THE DATA. JUST BEFORE THE FIRST BIT OF THE LAST 000 CHAR IS SENT,
 8711                                  ;* THE IERR BIT IS SET IN REG 17 TO CAUSE A 1 TO BE SENT, INTRODUCING A DATA
 8712                                  ;* ERROR. AT THE END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 0, INDICATING
 8713                                  ;* AN ERROR.
 8714                                  ;*
 8715                                  ;* - CRC-CCITT - 1'S PRESET :
 8716                                  ;* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT THE
 8717                                  ;* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
 8718                                  ;*
 8719                                  ;* - CRC-CCITT - 0'S PRESET :
 8720                                  ;* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT THE
 8721                                  ;* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
 8722                                  ;;*****************************************************************************
 8723  027252                          BGNTST
  (3)  027252                                                                  T14::
 8724                                  ;-----------------------------------------
 8725                                  ; CRC 16, CHAR MODE
 8726                                  ;-----------------------------------------
```

```
8727  027252  012737  027654  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
8728  027260  004737  005212                  JSR     PC,INITRN       ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
8729  027264  000226                          SYNCH
8730  027266  000011                          STRIP!DDCMP
8731  027270  004737  005574                  JSR     PC,TXCHAR       ;LOAD 000 CHAR, TX 1ST SYNCH
8732  027274  000000                          000
8733  027276  100010                          CHPCHK!8.
8734  027300  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
8735  027304  002676                          MSG1+6
8736  027306  000006                          6
8737  027310  004737  004726                  JSR     PC,STPLU        ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
8738  027314  000010                          8.
8739  027316  004737  007164                  JSR     PC,STPERR       ;MAKE 1ST BIT = 1 INSTEAD OF 0
8740  027322  000011                          STRIP!DDCMP
8741  027324  000001                          1
8742  027326  004737  004726                  JSR     PC,STPLU        ;CLOCK REST OF MESSAGE
8743  027332  000122                          82.
8744  027334  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
8745  027340  000001                          001
8746  027342  000000                          0
8747  027344  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
8748  027350  000125                          125
8749  027352  000000                          0
8750  027354  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
8751  027360  000252                          252
8752  027362  000000                          0
8753  027364  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
8754  027370  000377                          377
8755  027372  000000                          0
8756  027374  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
8757  027400  100000                          CRCCHK!000
8758  027402  000000                          0
8759
8760                                  ;-----------------------------------------
8761                                  ; CRC-CCITT-1'S PRESET, BIT MODE
8762                                  ;-----------------------------------------
8763  027404  004737  005212                  JSR     PC,INITRN       ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
8764  027410  000000                          000
8765  027412  000000                          000
8766  027414  004737  005574                  JSR     PC,TXCHAR       ;LOAD 000 CHAR, TX 1ST FLAG
8767  027420  000000                          000
8768  027422  100010                          CHPCHK!8.
8769  027424  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
8770  027430  002676                          MSG1+6
8771  027432  000006                          6
8772  027434  004737  004726                  JSR     PC,STPLU        ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
8773  027440  000010                          8.
8774  027442  004737  007164                  JSR     PC,STPERR       ;MAKE 1ST BIT = 1 INSTEAD OF 0
8775  027446  000000                          000
8776  027450  000001                          1
8777  027452  004737  004726                  JSR     PC,STPLU        ;CLOCK REST OF MESSAGE
8778  027456  000122                          82.
8779  027460  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
8780  027464  000001                          001
8781  027466  000000                          0
8782  027470  004737  007266                  JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
```

```
8783  027474  000125                      125
8784  027476  000000                      0
8785  027500  004737  007266              JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
8786  027504  000252                      252
8787  027506  000000                      0
8788  027510  004737  007266              JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
8789  027514  000377                      377
8790  027516  000000                      0
8791  027520  004737  007266              JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 000, CHK BCC = 1
8792  027524  101400                      CRCCHK!1400
8793  027526  000000                      0
8794
8795                                  ;----------------------------------------
8796                                  ; CRC-CCITT-0'S PRESET, BIT MODE
8797                                  ;----------------------------------------
8798  027530  004737  005212              JSR     PC,INITRN       ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
8799  027534  000000                      000
8800  027536  000100                      CRC1
8801  027540  004737  005574              JSR     PC,TXCHAR       ;LOAD 000 CHAR, TX 1ST FLAG
8802  027544  000000                      000
8803  027546  100010                      CHPCHK!8.
8804  027550  004737  010476              JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
8805  027554  002676                      MSG1+6
8806  027556  000006                      6
8807  027560  004737  004726              JSR     PC,STPLU        ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
8808  027564  000010                      8.
8809  027566  004737  007164              JSR     PC,STPERR       ;MAKE 1ST BIT = 1 INSTEAD OF 0
8810  027572  000100                      CRC1
8811  027574  000001                      1
8812  027576  004737  004726              JSR     PC,STPLU        ;CLOCK REST OF MESSAGE
8813  027602  000122                      82.
8814  027604  004737  007266              JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
8815  027610  000001                      001
8816  027612  000000                      0
8817  027614  004737  007266              JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 125
8818  027620  000125                      125
8819  027622  000000                      0
8820  027624  004737  007266              JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 252
8821  027630  000252                      252
8822  027632  000000                      0
8823  027634  004737  007266              JSR     PC,CKDATA       ;READ AND COMPARE CHAR TO 377
8824  027640  000377                      377
8825  027642  000000                      0
8826  027644  004737  007266              JSR     PC,CKDATA       ,READ AND COMPARE CHAR TO 000, CHK BCC = 1
8827  027650  101400                      CRCCHK!1400
8828  027652  000000                      0
8829
8830  027654  004737  003276  24$:        JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
8831  027660                  ENDTST
 (3)  027660                                                              L10046:
 (3)  027660  104401                                                              TRAP    C$ETST
8832
8833
8834
8835
8836
```

```
                                ;;*****************************************************************************
                                .SBTTL      TEST 15 - VRC PARITY GENERATION TEST
                                ;*
                                ;* SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
                                ;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC AND 7-BIT CHARS SELECTED.
                                ;* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
                                ;* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
                                ;* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1 AND FOR THE
                                ;* LAST 4 CHARS IT SHOULD = 0.
                                ;*
                                ;* SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
                                ;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
                                ;* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
                                ;* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
                                ;* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0 AND FOR THE
                                ;* LAST 4 CHARS IT SHOULD = 1.
                                ;*
                                ;*    DATA PATTERN Q = 000,120,125,137,040,052,057,177
                                ;;*****************************************************************************
027662                          BGNTST
(3) 027662                                                                      T15::
                                ;-------------------------------------------------------------
                                ; TEST ODD VRC GENERATION
                                ;-------------------------------------------------------------
027662  012737  000006  002366          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
027670  012737  000017  002364          MOV     #17,REGNUM      ;SET REG NO. = 17
027676                                  BGNSUB
(3) 027676                                                                      T15.1:
(3) 027676  104402                                                                      TRAP    C$BSUB
027700  012737  030102  002346          MOV     #8$,RETADR      ;SET SUBTEST EXIT ADDRESS FOR ERRORS
027706  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
027712  000026                          026
027714  000111                          CRC1!STRIP!DDCMP        ;CHAR MODE, ODD VRC
027716  004737  010554                  JSR     PC,LDBYTS       ;LOAD DATA INTO TX SILO
027722  002613                          PATQ
027724  000010                          8.
027726  004737  010716                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
027732  001000                          TXEOM
027734  000002                          2
027736  005037  002360                  CLR     WAX15
027742  012737  000347  002362          MOV     #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0,WAX16
027750  004737  003764                  JSR     PC,WRITAX       ;SET TX AND RCV LENGTHS = 7
027754  004737  004726                  JSR     PC,STPLU        ;CLOCK FIRST SYNCH
027760  000010                          8.
027762  004737  004726                  JSR     PC,STPLU        ;CLOCK 2ND SYNCH
027766  000010                          8.
027770  005001                          CLR     R1              ;INIT CHAR COUNT
027772  004737  004726          2$:     JSR     PC,STPLU        ;CLOCK A CHAR
027776  000010                          8.
030000  004737  003344                  JSR     PC,READLU       ;READ REG 17
030004  005201                          INC     R1              ;INCR CHAR COUNT
030006  020127  000004                  CMP     R1,#4           ;SEE IF 4 CHARS CLKD YET
030012  003014                          BGT     4$              ;BR IF YES
030014  132737  000040  002350          BITB    #TXDATA,REDBYT  ;SEE IF PARITY BIT IS SET
030022  001024                          BNE     6$              ;BR IF YES
030024  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
```

```
8890                                    ;REPORT ODD VRC PARITY BIT NOT SET
8891  030030                                    ERRDF   48,EM48,ERR7
 (4)  030030  104455                                                        TRAP    C$ERDF
 (5)  030032  000060                                                        .WORD   48
 (5)  030034  014263                                                        .WORD   EM48
 (5)  030036  020224                                                        .WORD   ERR7
8892  030040                                    ESCAPE  SUB
 (3)  030040  104410                                                        TRAP    C$ESCAPE
 (3)  030042  000040                                                        .WORD   L10050-.
8893  030044  132737  000040  002350  4$:      BITB    #TXDATA,REDBYT  ;SEE IF PARITY BIT IS CLEARED
8894  030052  001410                          BEQ     6$              ;BR IF YES
8895  030054  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
8896                                    ;REPORT ODD VRC PARITY BIT NOT CLEARED
8897  030060                                    ERRDF   49,EM49,ERR7
 (4)  030060  104455                                                        TRAP    C$ERDF
 (5)  030062  000061                                                        .WORD   49
 (5)  030064  014316                                                        .WORD   EM49
 (5)  030066  020224                                                        .WORD   ERR7
8898  030070                                    ESCAPE  SUB
 (3)  030070  104410                                                        TRAP    C$ESCAPE
 (3)  030072  000010                                                        .WORD   L10050-.
8899  030074  020127  000010            6$:     CMP     R1,#8.          ;SEE IF ALL CHARS TESTED YET
8900  030100  002734                          BLT     2$              ;BR IF NOT
8901  030102                            8$:
8902  030102                                    ENDSUB
 (3)  030102                                                                L10050:
 (3)  030102  104403                                                        TRAP    C$ESUB
8903                                    ;-----------------------------------------------------------
8904                                    ; TEST EVEN VRC GENERATION
8905                                    ;-----------------------------------------------------------
8906  030104  012737  000006  002366            MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
8907  030112  012737  000017  002364            MOV     #17,REGNUM      ;SET REG NO. = 17
8908  030120                                    BGNSUB
 (3)  030120                                                                T15.2:
 (3)  030120  104402                                                        TRAP    C$BSUB
8909  030122  012737  030324  002346            MOV     #18$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
8910  030130  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
8911  030134  000026                          026
8912  030136  000211                          CRC2!STRIP!DDCMP        ;CHAR MODE, EVEN VRC
8913  030140  004737  010554                  JSR     PC,LDBYTS       ;LOAD DATA INTO TX SILO
8914  030144  002613                          PATQ
8915  030146  000010                          8.
8916  030150  004737  010716                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
8917  030154  001000                          TXEOM
8918  030156  000002                          2
8919  030160  005037  002360                  CLR     WAX15
8920  030164  012737  000347  002362            MOV     #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0,WAX16
8921  030172  004737  003764                  JSR     PC,WRITAX       ;SET TX AND RCV LENGTHS = 7
8922  030176  004737  004726                  JSR     PC,STPLU        ;CLOCK FIRST SYNCH
8923  030202  000010                          8.
8924  030204  004737  004726                  JSR     PC,STPLU        ;CLOCK 2ND SYNCH
8925  030210  000010                          8.
8926  030212  005001                          CLR     R1              ;INIT CHAR COUNT
8927  030214  004737  004726            12$:  JSR     PC,STPLU        ;CLOCK A CHAR
8928  030220  000010                          8.
8929  030222  004737  003344                  JSR     PC,READLU       ;READ REG 17
```

```
 8930  030226  005201                          INC     R1              ;INCR CHAR COUNT
 8931  030230  020127  000004                  CMP     R1,#4           ;SEE IF 4 CHARS CLKD YET
 8932  030234  003014                          BGT     14$             ;BR IF YES
 8933  030236  132737  000040  002350          BITB    #TXDATA,REDBYT  ;SEE IF PARITY BIT IS CLEARED
 8934  030244  001424                          BEQ     16$             ;BR IF YES
 8935  030246  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8936                                  ;REPORT EVEN VRC PARITY BIT NOT CLEARED
 8937  030252                                  ERRDF   51,EM51,ERR7
   (4) 030252  104455                                                              TRAP    C$ERDF
   (5) 030254  000063                                                              .WORD   51
   (5) 030256  014411                                                              .WORD   EM51
   (5) 030260  020224                                                              .WORD   ERR7
 8938  030262                                  ESCAPE  SUB
   (3) 030262  104410                                                              TRAP    C$ESCAPE
   (3) 030264  000040                                                              .WORD   L10051-.
 8939  030266  132737  000040  002350  14$:     BITB    #TXDATA,REDBYT  ;SEE IF PARITY BIT IS SET
 8940  030274  001010                          BNE     16$             ;BR IF YES
 8941  030276  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 8942                                  ;REPORT EVEN VRC PARITY BIT NOT SET
 8943  030302                                  ERRDF   50,EM50,ERR7
   (4) 030302  104455                                                              TRAP    C$ERDF
   (5) 030304  000062                                                              .WORD   50
   (5) 030306  014355                                                              .WORD   EM50
   (5) 030310  020224                                                              .WORD   ERR7
 8944  030312                                  ESCAPE  SUB
   (3) 030312  104410                                                              TRAP    C$ESCAPE
   (3) 030314  000010                                                              .WORD   L10051-.
 8945  030316  020127  000010          16$:     CMP     R1,#8.          ;SEE IF ALL CHARS TESTED YET
 8946  030322  002734                          BLT     12$             ;BR IF NOT
 8947  030324                          18$:
 8948  030324                                  ENDSUB
   (3) 030324                                                                    L10051:
   (3) 030324  104403                                                              TRAP    C$ESUB
 8949  030326  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 8950  030332                          ENDTST
   (3) 030332                                                                    L10047:
   (3) 030332  104401                                                              TRAP    C$ETST
 8951
 8952
 8953
 8954
 8955
 8956                                  ;;*************************************************************************************
 8957                                  .SBTTL      TEST 16 - VRC ERROR DETECTION TEST
 8958                                  ;*
 8959                                  ;* SUBTEST 1 - FORCING OF BCC USING ODD VRC
 8960                                  ;* THE LINE UNIT IS PLACED IN CHAR MODE WITH ODD VRC AND 7-BIT CHARS SELECTED.
 8961                                  ;* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
 8962                                  ;* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
 8963                                  ;* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
 8964                                  ;* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
 8965                                  ;* RECEIVED (INDICATING AN ERROR).
 8966                                  ;*
 8967                                  ;* SUBTEST 2 - FORCING OF BCC USING EVEN VRC
 8968                                  ;* THE LINE UNIT IS PLACED IN CHAR MODE WITH EVEN VRC AND 7-BIT CHARS SELECTED.
 8969                                  ;* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
```

```
                        ;* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
                        ;* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
                        ;* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
                        ;* RECEIVED (INDICATING AN ERROR).
                        ;*
                        ;*   DATA PATTERN R = 000,100,120,124,164,172,176,177,000,100,120,124,164,
                        ;*                    172,176.
                        ;;*****************************************************************************
030334                  BGNTST
(3)  030334                                                                     T16::
                        ;-------------------------------------------------------------------
                        ; TEST ODD VRC ERROR DETECTION
                        ;-------------------------------------------------------------------
030334  012737  000006  002366          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
030342  012737  000012  002364          MOV     #12,REGNUM      ;SET REG NO.
030350                                  BGNSUB
(3)  030350                                                                     T16.1:
(3)  030350  104402                                                                     TRAP    C$BSUB
030352  012737  030570  002346          MOV     #10$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
030360  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
030364  000026                          026
030366  000111                          CRC1!STRIP!DDCMP        ;CHAR MODE, ODD VRC
030370  004737  010716                  JSR     PC,LODSIL       ;LOAD A THIRD SOM INTO TX SILO
030374  000400                          TXSOM
030376  000001                          1
030400  004737  010554                  JSR     PC,LDBYTS       ;LOAD DATA INTO TX BUFFER
030404  002623                          PATR
030406  000017                          15.
030410  004737  010716                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
030414  001000                          TXEOM
030416  000002                          2
030420  005037  002360                  CLR     WAX15
030424  012737  000347  002362          MOV     #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0,WAX16
030432  013737  002362  002424          MOV     WAX16,SAVLEN    ;STORE LENGTH 7
030440  004737  003764                  JSR     PC,WRITAX       ;SET TX AND RCV LENGTHS = 7
030444  004737  004726                  JSR     PC,STPLU        ;CLOCK 1ST 8 CHARS, WITH NO ERRORS
030450  000130                          88.
030452  012701  000007                  MOV     #7,R1           ;INIT COUNTER FOR LAST 7 CHARS
030456  004737  007164          3$:     JSR     PC,STPERR       ;ASSERT IERR BIT FOR 1 TIME
030462  000111                          CRC1!STRIP!DDCMP
030464  000001                          1
030466  004737  004726                  JSR     PC,STPLU        ;CLOCK REST OF CHAR
030472  000007                          7
030474  005301                          DEC     R1              ;DECR COUNTER
030476  001367                          BNE     3$              ;BR IF NOT DONE TRANSMITTING YET
030500  004737  004726                  JSR     PC,STPLU        ;CLOCK 2 TERMINATING SYNCHS
030504  000020                          16.
030506  012701  000010                  MOV     #8.,R1          ;INIT COUNTER FOR ERROR-FREE CHARS
030512  012703  002623                  MOV     #PATR,R3        ;INIT DATA PATTERN POINTER
030516  112337  030526          5$:     MOVB    (R3)+,6$        ;GET AN EXPECTED DATA CHAR
030522  004737  007266                  JSR     PC,CKDATA       ;GO CHECK CHAR, CHK BCC=0
030526  100000          6$:     BCCCHK!000
030530  000000                          0
030532  005301                          DEC     R1              ;DECR COUNTER
030534  001370                          BNE     5$              ;BR IF NOT DONE YET
030536  012701  000007                  MOV     #7,R1           ;INIT COUNTER FOR ERROR CHARS
```

```
 9023  030542  112337  030560          8$:     MOVB    (R3)+,9$        ;GET EXPECTED DATA CHAR
 9024  030546  052737  000001  030560          BIS     #BIT0,9$        ;EXPECT ERROR BIT 0 SET
 9025  030554  004737  007266                  JSR     PC,CKDATA       ;CHECK DATA, CHK BCC=1
 9026  030560  100400                  9$:     BCCCHK!RXBCC!000
 9027  030562  000000                          0
 9028  030564  005301                          DEC     R1              ;DECR COUNTER
 9029  030566  001365                          BNE     8$              ;BR IF NOT DONE YET
 9030  030570                          10$:
 9031  030570                                  ENDSUB
   (3) 030570                                                                  L10053:
   (3) 030570  104403                                                            TRAP    C$ESUB
 9032                                  ;-------------------------------------------------------------
 9033                                  ; TEST EVEN VRC ERROR DETECTION
 9034                                  ;-------------------------------------------------------------
 9035  030572  012737  000006  002366          MOV     #6,AXNUM        ;SET AX BYTE NO. FOR AX3
 9036  030600  012737  000012  002364          MOV     #12,REGNUM      ;SET REG NO.
 9037  030606                                  BGNSUB
   (3) 030606                                                                  T16.2:
   (3) 030606  104402                                                            TRAP    C$BSUB
 9038  030610  012737  031026  002346          MOV     #30$,RETADR     ;SET SUBTEST EXIT ADRS FOR ERRORS
 9039  030616  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
 9040  030622  000026                          026
 9041  030624  000211                          CRC2!STRIP!DDCMP        ;CHAR MODE, EVEN VRC
 9042  030626  004737  010716                  JSR     PC,LODSIL       ;LOAD A THIRD SOM INTO TX SILO
 9043  030632  000400                          TXSOM
 9044  030634  000001                          1
 9045  030636  004737  010554                  JSR     PC,LDBYTS       ;LOAD DATA INTO TX BUFFER
 9046  030642  002623                          PATR
 9047  030644  000017                          15.
 9048  030646  004737  010716                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
 9049  030652  001000                          TXEOM
 9050  030654  000002                          2
 9051  030656  005037  002360                  CLR     WAX15
 9052  030662  012737  000347  002362          MOV     #TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0,WAX16
 9053  030670  013737  002362  002424          MOV     WAX16,SAVLEN    ;STORE LENGTH 7
 9054  030676  004737  003764                  JSR     PC,WRITAX       ;SET TX AND RCV LENGTHS = 7
 9055  030702  004737  004726                  JSR     PC,STPLU        ;CLOCK 1ST 8 CHARS, WITH NO ERRORS
 9056  030706  000130                          88.
 9057  030710  012701  000007                  MOV     #7,R1           ;INIT COUNTER FOR LAST 7 CHARS
 9058  030714  004737  007164          23$:    JSR     PC,STPERR       ;ASSERT IERR BIT FOR 1 TIME
 9059  030720  000211                          CRC2!STRIP!DDCMP
 9060  030722  000001                          1
 9061  030724  004737  004726                  JSR     PC,STPLU        ;CLOCK REST OF CHAR
 9062  030730  000007                          7
 9063  030732  005301                          DEC     R1              ;DECR COUNTER
 9064  030734  001367                          BNE     23$             ;BR IF NOT DONE TRANSMITTING YET
 9065  030736  004737  004726                  JSR     PC,STPLU        ;CLOCK 2 TERMINATING SYNCHS
 9066  030742  000020                          16.
 9067  030744  012701  000010                  MOV     #8.,R1          ;INIT COUNTER FOR ERROR-FREE CHARS
 9068  030750  012703  002623                  MOV     #PATR,R3        ;INIT DATA PATTERN POINTER
 9069  030754  112337  030764          25$:    MOVB    (R3)+,26$       ;GET EXPECTED DATA CHAR
 9070  030760  004737  007266                  JSR     PC,CKDATA       ;CHK DATA, CHECK BCC=0
 9071  030764  100000                  26$:    BCCCHK!000
 9072  030766  000000                          0
 9073  030770  005301                          DEC     R1              ;DECR COUNTER
 9074  030772  001370                          BNE     25$             ;BR IF NOT DONE YET
```

```
 9075  030774  012701  000007                 MOV     #7,R1           ;INIT COUNTER FOR ERROR CHARS
 9076  031000  112337  031016          28$:   MOVB    (R3)+,29$       ;GET EXPECTED DATA CHAR
 9077  031004  052737  000001  031016        BIS     #BIT0,29$       ;SET EXPECTED ERROR BIT 0
 9078  031012  004737  007266                JSR     PC,CKDATA       ;CHK DATA, CHK BCC=1
 9079  031016  100400                 29$:   BCCCHK!RXBCC!000
 9080  031020  000000                        0
 9081  031022  005301                        DEC     R1              ;DECR COUNTER
 9082  031024  001365                        BNE     28$             ;BR IF NOT DONE YET
 9083  031026                         30$:
 9084  031026                                ENDSUB
   (3) 031026                                                                      L10054:
   (3) 031026  104403                                                                    TRAP    C$ESUB
 9085  031030  004737  003276                JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 9086  031034                         ENDTST
   (3) 031034                                                                      L10052:
   (3) 031034  104401                                                                    TRAP    C$ETST
 9087
 9088
 9089
 9090
 9091
 9092                                ;;*************************************************************************************
 9093                                .SBTTL      TEST 17 - INTEGRAL MODEM INTERFACE TEST - CHAR MODE, CRC
 9094                                ;*
 9095                                ;* THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM IN AX3-15, AND A
 9096                                ;* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
 9097                                ;* ON THE LINE UNIT OR AT THE CABLE. THE MESSAGE CONSISTS OF
 9098                                ;* 2 SYNCHS, 000,125,252,377,000, AND  1 SYNCH. IF THE P-TABLE FOR THE CURRENT
 9099                                ;* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
 9100                                ;* SKIPPED FOR THAT UNIT.
 9101                                ;;*************************************************************************************
 9102  031036                        BGNTST
   (3) 031036                                                                      T17::
 9103  031036  012737  000021  002434        MOV     #17.,TSTNUM     ;SET TEST NO.
 9104  031044  012737  031316  002346        MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 9105  031052  004737  003276                MOV     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9106  031056  004737  010766                JSR     PC,CKLPBK       ;CHECK LOOPBACK -
 9107  031062  000010                        INTGRL                  ;    SEE IF TEST SHOULD BE RUN
 9108  031064  012737  000323  031102        MOV     #I422!XYZ!V35!OP!TEST,6$ ;SET UP TO SELECT INTEGRAL MODEM
 9109  031072  004737  010304                JSR     PC,SETUP        ;PROGRAM THE USYRT
 9110  031076  000226                        SYNCH
 9111  031100  000011                        STRIP!DDCMP
 9112  031102  000000                 6$:    .WORD   0
 9113  031104  000000                        000
 9114  031106  004737  010636                JSR     PC,LDMSG1       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 9115  031112  142777  000010  151320        BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
 9116  031120  012737  000012  002364        MOV     #12,REGNUM      ;SET LU REG NO. = 12
 9117  031126  012703  002762                MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 9118  031132  013702  002264          9$:   MOV     TCOUNT,R2               ;INIT TIMER
 9119  031136  004737  003344          10$:  JSR     PC,READLU       ;READ REG 12
 9120  031142  132737  000020  002350        BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 9121  031150  001011                        BNE     12$             ;BR IF YES
 9122  031152  005202                        INC     R2              ;INCREMENT TIMER
 9123  031154  001370                        BNE     10$             ;BR IF NO TIME-OUT YET
 9124  031156  004737  004200                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9125                                ;REPORT IRDY NOT SET
```

```
 9126  031162                                          ERRDF   17,EM17,ERR7
  (4)  031162  104455                                                          TRAP    C$ERDF
  (5)  031164  000021                                                          .WORD   17
  (5)  031166  013436                                                          .WORD   EM17
  (5)  031170  020224                                                          .WORD   ERR7
 9127  031172  000451                                  BR      24$             ;ESCAPE TO END OF TEST
 9128  031174  012337  031204                  12$:    MOV     (R3)+,16$
 9129  031200  004737  007266                          JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 9130  031204  000000                          16$:    0
 9131  031206  000000                                  0
 9132  031210  020327  003000                          CMP     R3,#RCVBUF+14.  ;SEE IF ALL CHARS CHECKED YET
 9133  031214  103746                                  BLO     9$              ;BR IF NOT YET
 9134  031216  004737  004620                          JSR     PC,WAIT50       ;STALL FOR 50 MICRO-SEC
 9135  031222  004737  005024                          JSR     PC,OACTIV       ;CHECK OACT = 0
 9136  031226  000000                                  0
 9137  031230  004737  006232                          JSR     PC,IACTIV       ;CHECK IACT = 0
 9138  031234  000000                                  0
 9139  031236  012737  000013  002364                  MOV     #13,REGNUM      ;SET REG NO. = 13
 9140  031244  004737  003344                          JSR     PC,READLU       ;READ REG 13
 9141  031250  042737  000232  002350                  BIC     #RING!HDX!MODR!STBY,REDBYT      ;CLR UNUSED BITS
 9142  031256  023727  002350  000000                  CMP     REDBYT,#0       ;CHECK REG 13 FOR 000 (MODEM SIGNALS SHOULD BE CLEARED)
 9143  031264  001414                                  BEQ     24$             ;BR IF CLEARED
 9144  031266  012737  000000  002370                  MOV     #0,GOODAT       ;SET EXPECTED DATA = 0
 9145  031274  013737  002350  002372                  MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
 9146  031302  004737  004200                          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9147                                          ;REPORT REG MISCOMPARE
 9148  031306                                          ERRDF   3,EM3,ERR3
  (4)  031306  104455                                                          TRAP    C$ERDF
  (5)  031310  000003                                                          .WORD   3
  (5)  031312  013265                                                          .WORD   EM3
  (5)  031314  015674                                                          .WORD   ERR3
 9149  031316                                  24$:
 9150  031316                                  ENDTST
  (3)  031316                                                          L10055:
  (3)  031316  104401                                                          TRAP    C$ETST
 9151
 9152
 9153
 9154
 9155
 9156                                          ;;*****************************************************************************
 9157                                          .SBTTL      TEST 18 - V.35 MODEM INTERFACE TEST - CHAR MODE, CRC
 9158                                          ;*
 9159                                          ;* THE V.35 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
 9160                                          ;* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
 9161                                          ;* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
 9162                                          ;* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
 9163                                          ;* 2 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
 9164                                          ;* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
 9165                                          ;* SKIPPED FOR THAT UNIT.
 9166                                          ;;*****************************************************************************
 9167  031320                                  BGNTST
  (3)  031320                                                          T18::
 9168  031320  012737  000022  002434                  MOV     #18.,TSTNUM     ;SET TEST NO.
 9169  031326  012737  031570  002346                  MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 9170  031334  004737  003276                          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
```

```
 9171  031340  004737  010766                    JSR     PC,CKLPBK       ;CHECK LOOPBACK -
 9172  031344  000020                            V35                     ;   SEE IF TEST SHOULD BE RUN
 9173  031346  012737  000313  031364            MOV     #I422!XYZ!INTGRL!OP!TEST,6$ ;SET UP TO SELECT V35
 9174  031354  004737  010304                    JSR     PC,SETUP        ;PROGRAM THE USYRT
 9175  031360  000226                            SYNCH
 9176  031362  000011                            STRIP!DDCMP
 9177  031364  000000                    6$:     .WORD   0
 9178  031366  000000                            000
 9179  031370  142777  000010  151042            BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
 9180  031376  012737  000013  002364            MOV     #13,REGNUM      ;SET LU REG NO. = 13
 9181  031404  004737  003344                    JSR     PC,READLU       ;READ REG 13
 9182  031410  132737  000001  002350            BITB    #CARR,REDBYT    ;CHECK FOR CARRIER FALSELY SET
 9183  031416  001415                            BEQ     8$              ;BR IF NOT SET
 9184  031420  012737  000000  002370            MOV     #000,GOODAT     ;SET EXPECTED DATA
 9185  031426  013737  002350  002372            MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
 9186  031434  004737  004200                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9187                                    ;REPORT CARRIER NOT CLEARED
 9188  031440                                    ERRDF   66,EM66,ERR7
  (4)  031440  104455                                                                    TRAP    C$ERDF
  (5)  031442  000102                                                                    .WORD   66
  (5)  031444  015000                                                                    .WORD   EM66
  (5)  031446  020224                                                                    .WORD   ERR7
 9189  031450  000447                            BR      24$
 9190  031452  152777  000010  150760    8$:     BISB    #LULOOP,@BSEL1  ;SET LULOOP AGAIN
 9191  031460  004737  010636                    JSR     PC,LDMSG1       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 9192  031464  142777  000010  150746            BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
 9193  031472  012737  000012  002364            MOV     #12,REGNUM      ;SET LU REG NO. = 12
 9194  031500  012703  002762                    MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 9195  031504  013702  002264            9$:     MOV     TCOUNT,R2               ;INIT TIMER
 9196  031510  004737  003344            10$:    JSR     PC,READLU       ;READ REG 12
 9197  031514  132737  000020  002350            BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 9198  031522  001011                            BNE     12$             ;BR IF YES
 9199  031524  005202                            INC     R2              ;INCREMENT TIMER
 9200  031526  001370                            BNE     10$             ;BR IF NO TIME-OUT YET
 9201  031530  004737  004200                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9202                                    ;REPORT IRDY NOT SET
 9203  031534                                    ERRDF   17,EM17,ERR7
  (4)  031534  104455                                                                    TRAP    C$ERDF
  (5)  031536  000021                                                                    .WORD   17
  (5)  031540  013436                                                                    .WORD   EM17
  (5)  031542  020224                                                                    .WORD   ERR7
 9204  031544  000411                            BR      24$             ;ESCAPE TO END OF TEST
 9205  031546  012337  031556            12$:    MOV     (R3)+,16$
 9206  031552  004737  007266                    JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 9207  031556  000000                    16$:    0
 9208  031560  000000                            0
 9209  031562  020327  003000                    CMP     R3,#RCVBUF+14.  ;SEE IF ALL CHARS CHECKED YET
 9210  031566  103746                            BLO     9$              ;BR IF NOT YET
 9211  031570                            24$:
 9212  031570                            ENDTST
  (3)  031570                                                                    L10056:
  (3)  031570  104401                                                                    TRAP    C$ETST
 9213
 9214
 9215
 9216
```

```
                        ;;*************************************************************************
                        .SBTTL      TEST 19 - RS 232C AND RS 423 MODEM INTERFACE TEST - CHAR MODE, CRC
                        ;*
                        ;* THE RS232C & RS423 (XYZ) MODEM INTERFACE IS SELECTED BY THE PROGRAM IN
                        ;* AX3-15, AND A MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURN-
                        ;* AROUND CONNECTOR ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
                        ;* OR A MODEM TEST MODE. THE MESSAGE CONSISTS
                        ;* OF 2 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE
                        ;* P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS
                        ;* PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
                        ;;*************************************************************************
031572                  BGNTST
(3) 031572                                                                      T19::
031572  012737  000023  002434          MOV     #19.,TSTNUM     ;SET TEST NO.
031600  012737  031752  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
031606  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
031612  004737  010766                  JSR     PC,CKLPBK       ;CHECK LOOPBACK -
031616  000100                          XYZ                     ;   SEE IF TEST SHOULD BE RUN
031620  012737  000233  031636          MOV     #I422!V35!INTGRL!OP!TEST,6$ ;SET UP TO SELECT XYZ
031626  004737  010304                  JSR     PC,SETUP        ;PROGRAM THE USYRT
031632  000226                          SYNCH
031634  000011                          STRIP!DDCMP
031636  000000                  6$:     .WORD   0
031640  000000                          000
031642  004737  010636                  JSR     PC,LDMSG1       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
031646  142777  000010  150564          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
031654  012737  000012  002364          MOV     #12,REGNUM      ;SET LU REG NO. = 12
031662  012703  002762                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
031666  013702  002264          9$:     MOV     TCOUNT,R2               ;INIT TIMER
031672  004737  003344          10$:    JSR     PC,READLU       ;READ REG 12
031676  132737  000020  002350          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
031704  001011                          BNE     12$             ;BR IF YES
031706  005202                          INC     R2              ;INCREMENT TIMER
031710  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
031712  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT IRDY NOT SET
031716                          ERRDF   17,EM17,ERR7
(4) 031716  104455                                                      TRAP    C$ERDF
(5) 031720  000021                                                      .WORD   17
(5) 031722  013436                                                      .WORD   EM17
(5) 031724  020224                                                      .WORD   ERR7
031726  000411                          BR      24$             ;ESCAPE TO END OF TEST
031730  012337  031740          12$:    MOV     (R3)+,16$
031734  004737  007266                  JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
031740  000000                  16$:    0
031742  000000                          0
031744  020327  003000                  CMP     R3,#RCVBUF+14.  ;SEE IF ALL CHARS CHECKED YET
031750  103746                          BLO     9$              ;BR IF NOT YET
031752                          24$:
031752                          ENDTST
(3) 031752                                                              L10057:
(3) 031752  104401                                                              TRAP    C$ETST
```

```
                                ;;*****************************************************************************
                                .SBTTL      TEST 20 - RS 422 MODEM INTERFACE TEST - CHAR MODE, CRC
                                ;*
                                ;* THE RS 422 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
                                ;* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
                                ;* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
                                ;* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
                                ;* 2 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
                                ;* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
                                ;* SKIPPED FOR THAT UNIT.
                                ;;*****************************************************************************
031754                          BGNTST
(3)  031754                                                                     T20::
031754  012737  000024  002434          MOV     #20.,TSTNUM     ;SET TEST NO.
031762  012737  032134  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
031770  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
031774  004737  010766                  JSR     PC,CKLPBK       ;CHECK LOOPBACK -
032000  000200                          I422                    ;   SEE IF TEST SHOULD BE RUN
032002  012737  000133  032020          MOV     #XYZ!V35!INTGRL!OP!TEST,6$ ;SET UP TO SELECT 422
032010  004737  010304                  JSR     PC,SETUP        ;PROGRAM THE USYRT
032014  000226                          SYNCH
032016  000011                          STRIP!DDCMP
032020  000000                  6$:     .WORD   0
032022  000000                          000
032024  004737  010636                  JSR     PC,LDMSG1       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
032030  142777  000010  150402          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
032036  012737  000012  002364          MOV     #12,REGNUM      ;SET LU REG NO. = 12
032044  012703  002762                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
032050  013702  002264          9$:     MOV     TCOUNT,R2               ;INIT TIMER
032054  004737  003344          10$:    JSR     PC,READLU       ;READ REG 12
032060  132737  000020  002350          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
032066  001011                          BNE     12$             ;BR IF YES
032070  005202                          INC     R2              ;INCREMENT TIMER
032072  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
032074  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT IRDY NOT SET
032100                          ERRDF   17,EM17,ERR7
(4)  032100  104455                                                             TRAP    C$ERDF
(5)  032102  000021                                                             .WORD   17
(5)  032104  013436                                                             .WORD   EM17
(5)  032106  020224                                                             .WORD   ERR7
032110  000411                          BR      24$             ;ESCAPE TO END OF TEST
032112  012337  032122          12$:    MOV     (R3)+,16$
032116  004737  007266                  JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
032122  000000                  16$:    0
032124  000000                          0
032126  020327  003000                  CMP     R3,#RCVBUF+14.  ;SEE IF ALL CHARS CHECKED YET
032132  103746                          BLO     9$              ;BR IF NOT YET
032134                          24$:
032134                          ENDTST
(3)  032134                                                                 L10060:
(3)  032134  104401                                                             TRAP    C$ETST
```

```
                        ;;*************************************************************************
                        .SBTTL      TEST 21 - HALF-DUPLEX BIT (HALF DUPX) TEST
                        ;*
                        ;* THIS TEST VERIFIES THAT SETTING HALF-DUPLEX BIT IN REG 13 INHIBITS
                        ;* LOADING OF THE USYRT TRANSMITTER FROM THE TRANSMITTER SILO.
                        ;* A MASTER CLEAR IS ISSUED, DDCMP MODE IS ENTERED, AND THE HALF DUPX
                        ;* BIT IN REG 13 IS SET. A MESSAGE IS LOADED INTO THE TX SILO
                        ;* CONSISTING OF 2 SYNCHS, 000,125,252,377,000, AND 2 MORE SYNCHS.
                        ;* THE LINE UNIT IS THEN CLOCKED EXTENSIVELY, AND THE TX SILO IS CHECKED TO
                        ;* BE STILL LOADED (NO CHARS SHOULD HAVE BEEN REMOVED) AND THE RECEIVER
                        ;* IS MONITORED TO INSURE THAT NO RCV FLAGS ARE GENERATED.
                        ;;*************************************************************************
032136                  BGNTST
                                                                                    T21::
032136  012737  032226  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
032144  012737  000013  002364          MOV     #13,REGNUM      ;SET REG NO. = 13
032152  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
032156  000226                          SYNCH
032160  000011                          STRIP!DDCMP
032162  112737  000020  002352          MOVB    #HDX,WRIBYT
032170  004737  003422                  JSR     PC,WRITLU       ;SET HDX BIT IN REG 13
032174  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG INTO TX SILO
032200  002674                          MSG1+4
032202  000007                          7
032204  004737  004726                  JSR     PC,STPLU        ;CLK MORE THAN ENTIRE MSG
032210  000136                          94.
032212  004737  004334                  JSR     PC,OSIRDY       ;CHK ORDY = 1, OCOR = 1
032216  000003                          3
032220  004737  005746                  JSR     PC,ISIRDY       ;CHK ICIR = 1, IRDY = 0
032224  000001                          1
032226  004737  003276          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
032232                          ENDTST
032232                                                                  L10061:
032232  104401                                                                  TRAP    C$ETST


                        ;;*************************************************************************
                        .SBTTL      TEST 22 - HALF-DUPLEX RCV DISABLED TEST WITH SILOS DISABLED
                        ;*
                        ;* THIS TEST SENDS A MESSAGE IN HDX, CHAR MODE, WITH NO ERROR DETECTION, AND
                        ;* THE SILOS DISABLED. THE MSG CONSISTS OF 2 SYNCHS AND 2 000 CHARS.
                        ;* THE DATA IS SENT WITH LULOOP SET FOR TTL DATA LOOPBACK. THE PROGRAM CHECKS
                        ;* THAT THE RECEIVER NEVER BECOMES ACTIVE, BECAUSE THE RCV CLOCK IS INHIBITED
                        ;* WHEN THE HDX BIT IS SET.
                        ;;*************************************************************************
032234                  BGNTST
                                                                                    T22::
032234  012737  032432  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
032242  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
032246  004737  010304                  JSR     PC,SETUP        ;PROGRAM USYRT FOR CHAR MODE, NO CRC
```

```
 9367  032252  000226                        SYNCH
 9368  032254  000301                        CRC2!CRC1!DDCMP
 9369  032256  000000                        000
 9370  032260  000000                        000
 9371  032262  012737  000013  002364        MOV     #13,REGNUM      ;SET LU REG NO. = 13
 9372  032270  112737  000020  002352        MOVB    #HDX,WRIBYT     ;SET HDX BIT IN REG 13
 9373  032276  004737  003422                JSR     PC,WRITLU
 9374  032302  012737  000014  002364        MOV     #14,REGNUM      ;SET LU REG NO. = 14
 9375  032310  112737  000140  002352        MOVB    #TXEN!DISSI,WRIBYT ;DISABLE SILOS
 9376  032316  004737  003422                JSR     PC,WRITLU
 9377  032322  012737  000140  002416        MOV     #TXEN!DISSI,DISILO
 9378  032330  012737  000002  002366        MOV     #2,AXNUM        ;SET AX BYTE NO FOR AX1
 9379  032336  112737  000000  002360        MOVB    #000,WAX15      ;SET TSOM IN USYRT
 9380  032344  112737  000001  002362        MOVB    #TSOM,WAX16
 9381  032352  004737  003764                JSR     PC,WRITAX
 9382  032356  004737  004726                JSR     PC,STPLU        ;CLOCK FIRST SYNCH OUT
 9383  032362  000013                        11.
 9384  032364  112737  000000  002360        MOVB    #000,WAX15      ;LOAD FIRST 000 DATA CHAR INTO USYRT
 9385  032372  112737  000000  002362        MOVB    #000,WAX16
 9386  032400  004737  003764                JSR     PC,WRITAX
 9387  032404  004737  004726                JSR     PC,STPLU        ;CLOCK SECOND SYNCH
 9388  032410  000010                        8.
 9389  032412  004737  003764                JSR     PC,WRITAX       ;LOAD SECOND 000 CHAR
 9390  032416  004737  004726                JSR     PC,STPLU        ;CLOCK FIRST 000 CHAR OUT
 9391  032422  000013                        11.
 9392  032424  004737  006232                JSR     PC,IACTIV       ;CHK FOR IACT = 0 (RECEIVER NOT ACTIVE)
 9393  032430  000000                        0
 9394  032432  005037  002416        24$:    CLR     DISILO          ;CLEAR DISABLE SILO FLAG
 9395  032436  004737  003276                JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
 9396  032442                        ENDTST
   (3) 032442                                                                L10062:
   (3) 032442  104401                                                              TRAP    C$ETST
 9397
 9398
 9399
 9400
 9401
 9402                    ;;*************************************************************************
 9403                    .SBTTL      TEST 23 - INTERACTION OF MODEM CONTROL BITS
 9404                    ;*
 9405                    ;* THIS TEST WILL BE RUN ONLY IF THE P-TABLE FOR THIS UNIT INDICATES THAT THE
 9406                    ;* H3254 AND H3255 TEST CONNECTORS ARE INSTALLED. OTHERWISE, THE TEST WILL BE
 9407                    ;* SKIPPED FOR THE UNIT.
 9408                    ;* THE FOLLOWING SUBTESTS ARE PERFORMED:
 9409                    ;* - A MASTER CLEAR IS DONE AND REG 13 IS READ AND CHECKED FOR INITIALIZED
 9410                    ;*   STATE, WITH LULOOP SET TO 1. THEN, LULOOP IS CLEARED AND REG 13 IS READ
 9411                    ;*   AND CHECKED FOR THE PROPER STATE, WITH LULOOP CLEARED.
 9412                    ;*   REG 13 IS THEN LOADED WITH 0'S, AND READ AND CHECKED FOR THE INITIALIZED
 9413                    ;*   STATE.
 9414                    ;*   REG 17 IS THEN READ AND CHECKED FOR INITIALIZED STATE.
 9415                    ;* - RUN IS SET IN BSEL1, AND REG 13 IS READ AND CHECKED FOR RING SET.
 9416                    ;* - POLL IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGQ SET.
 9417                    ;* - BPOLL IS SET IN REG 12, ONLY TO LIGHT THE LED FOR THIS SIGNAL.
 9418                    ;* - DTR IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR DTR AND MODR SET.
 9419                    ;* - SELFR IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGR SET.
 9420                    ;* - HDX IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR HDX SET.
```

```
                        ;* - MAINT1 IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR TEST MODE SET.
                        ;* - SELSBY IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR STBY SET.
                        ;* - A MASTER CLEAR IS DONE, 2 TSOM'S ARE LOADED INTO THE TX SILO, THE LINE
                        ;*   UNIT IS CLOCKED UNTIL THE TRANSMITTER IS ACTIVE, AND REG 13 IS READ AND
                        ;*   CHECKED FOR RTS, CS, CARR SET.
                        ;;*****************************************************************************
 9427  032444           BGNTST
  (3)  032444                                                                    T23::
 9428  032444  012737  000027  002434          MOV     #23.,TSTNUM     ;SET TEST NO.
 9429  032452  012737  033730  002346          MOV     #A12,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
 9430  032460  004737  010766                  JSR     PC,CKLPBK       ;SEE IF H3254,5 INSTALLED - SKIP TEST IF NOT
 9431  032464  100000                          TCCHEK
 9432
 9433                          ;-------------------------------------------------------------------------
 9434                          ; DO MASTER CLEAR, CHK REGS 13,17 FOR INITIALIZED STATES
 9435                          ;-------------------------------------------------------------------------
 9436  032466                  BGNSUB
  (3)  032466                                                                    T23.1:
  (3)  032466  104402                                                                    TRAP    C$BSUB
 9437  032470  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9438  032474  012737  000013  002364          MOV     #13,REGNUM      ;SET REG NO. = 13
 9439  032502  004737  003344                  JSR     PC,READLU       ;READ REG 13
 9440  032506  023727  002350  000210          CMP     REDBYT,#RING!MODR       ;CHECK REG 13 FOR INIT'D STATE
 9441  032514  001416                          BEQ     6$              ;BR IF REG 13 INIT'D
 9442  032516  012737  000210  002370          MOV     #RING!MODR,GOODAT       ;SET EXPECTED DATA
 9443  032524  013737  002350  002372          MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
 9444  032532  004737  004200          3$:     JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9445                          ;REPORT REG MISCOMPARE
 9446  032536                  ERRDF   3,EM3,ERR2
  (4)  032536  104455                                                                    TRAP    C$ERDF
  (5)  032540  000003                                                                    .WORD   3
  (5)  032542  013265                                                                    .WORD   EM3
  (5)  032544  015366                                                                    .WORD   ERR2
 9447  032546                  ESCAPE  SUB
  (3)  032546  104410                                                                    TRAP    C$ESCAPE
  (3)  032550  000170                                                                    .WORD   L10064-.
 9448  032552  142777  000010  147660  6$:     BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
 9449  032560  004737  003344                  JSR     PC,READLU       ;READ REG 13
 9450  032564  023727  002350  000000          CMP     REDBYT,#0       ;CHECK FOR INITIALIZED STATE
 9451  032572  001416                          BEQ     8$              ;BR IF OK
 9452  032574  012737  000000  002370          MOV     #0,GOODAT       ;GET EXPECTED DATA
 9453  032602  013737  002350  002372          MOV     REDBYT,BADDAT   ;GET ACTUAL DATA
 9454  032610  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9455                          ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
 9456  032614                  ERRDF   2,EM2,ERR2
  (4)  032614  104455                                                                    TRAP    C$ERDF
  (5)  032616  000002                                                                    .WORD   2
  (5)  032620  013226                                                                    .WORD   EM2
  (5)  032622  015366                                                                    .WORD   ERR2
 9457  032624                  ESCAPE  SUB
  (3)  032624  104410                                                                    TRAP    C$ESCAPE
  (3)  032626  000112                                                                    .WORD   L10064-.
 9458  032630  005037  002352          8$:     CLR     WRIBYT          ;SET DATA = 0 TO BE WRITTEN
 9459  032634  004737  003422                  JSR     PC,WRITLU       ;LOAD 0'S INTO REG 13
 9460  032640  004737  003344                  JSR     PC,READLU       ;READ REG 13
 9461  032644  023727  002350  000000          CMP     REDBYT,#000     ;CHECK FOR REG 13 CLEARED
```

```
 9462  032652  001407                          BEQ     9$                  ;BR IF CLEARED
 9463  032654  012737  000000  002370          MOV     #000,GOODAT         ;SET EXPECTED DATA
 9464  032662  013737  002350  002372          MOV     REDBYT,BADDAT       ;SET ACTUAL DATA
 9465  032670  000720                          BR      3$                  ;GO PRINT ERROR
 9466  032672  012737  000017  002364  9$:     MOV     #17,REGNUM          ;SET REG NO. = 17
 9467  032700  004737  003344                  JSR     PC,READLU           ;READ REG 17
 9468  032704  042737  000002  002350          BIC     #MCLK,REDBYT        ;IGNORE MCLK BIT
 9469  032712  123727  002350  000051          CMPB    REDBYT,#TXDATA!ICIR!DDCMP ;CHK REG 17 FOR INIT'D STATE
 9470  032720  001407                          BEQ     10$                 ;BR IF REG 17 INITIALIZED
 9471  032722  012737  000051  002370          MOV     #TXDATA!ICIR!DDCMP,GOODAT ;SET EXPECTED DATA
 9472  032730  013737  002350  002372          MOV     REDBYT,BADDAT       ;SET ACTUAL DATA
 9473  032736  000675                          BR      3$                  ;GO REPORT ERROR
 9474  032740                          10$:
 9475  032740                                  ENDSUB
   (3) 032740                                                                        L10064:
   (3) 032740  104403                                                                      TRAP    C$ESUB
 9476
 9477                                  ;-------------------------------------------------------------------------
 9478                                  ; SET RUN IN BSEL1, CHECK FOR RING SET IN REG 13
 9479                                  ;-------------------------------------------------------------------------
 9480  032742                                  BGNSUB
   (3) 032742                                                                        T23.2:
   (3) 032742  104402                                                                      TRAP    C$BSUB
 9481  032744  004737  003276                  JSR     PC,MSTCLR           ;ISSUE MASTER CLEAR
 9482  032750  105077  147464                  CLRB    @BSEL1              ;CLEAR LULOOP
 9483  032754  112777  000200  147456          MOVB    #RUN,@BSEL1         ;SET RUN BIT IN BSEL1
 9484  032762  112777  000010  147450          MOVB    #LULOOP,@BSEL1      ;CLEAR RUN, SET LULOOP
 9485  032770  012737  000013  002364          MOV     #13,REGNUM          ;SET REG NO. = 13
 9486  032776  004737  003344                  JSR     PC,READLU           ;READ REG 13
 9487  033002  132737  000200  002350          BITB    #RING,REDBYT        ;SEE IF RING = 1
 9488  033010  001010                          BNE     9$                  ;BR IF RING = 1
 9489  033012  004737  004200                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
 9490
                                  ;REPORT RING NOT SET
 9491  033016                                  ERRDF   56,EM56,ERR7
   (4) 033016  104455                                                                TRAP    C$ERDF
   (5) 033020  000070                                                                .WORD   56
   (5) 033022  014577                                                                .WORD   EM56
   (5) 033024  020224                                                                .WORD   ERR7
 9492  033026                                  ESCAPE  SUB
   (3) 033026  104410                                                                TRAP    C$ESCAPE
   (3) 033030  000002                                                                .WORD   L10065-.
 9493  033032                          9$:
 9494  033032                                  ENDSUB
   (3) 033032                                                                        L10065:
   (3) 033032  104403                                                                      TRAP    C$ESUB
 9495
 9496                                  ;-------------------------------------------------------------------------
 9497                                  ; SET POLL IN REG 13, CHK FOR SIGQ SET IN REG 17
 9498                                  ;-------------------------------------------------------------------------
 9499  033034                                  BGNSUB
   (3) 033034                                                                        T23.3:
   (3) 033034  104402                                                                      TRAP    C$BSUB
 9500  033036  004737  003276                  JSR     PC,MSTCLR           ;ISSUE MASTER CLEAR
 9501  033042  112737  000200  002352          MOVB    #POLL,WRIBYT
 9502  033050  012737  000013  002364          MOV     #13,REGNUM          ;SET REG NO. = 13
 9503  033056  004737  003422                  JSR     PC,WRITLU           ;SET POLL IN REG 13
```

```
 9504  033062  012737  000017  002364          MOV     #17,REGNUM      ;SET REG NO. = 17
 9505  033070  004737  003344                  JSR     PC,READLU       ;READ REG 17
 9506  033074  132737  000100  002350          BITB    #SIGQ,REDBYT    ;SEE IF SIGQ = 1
 9507  033102  001006                          BNE     6$              ;BR IF SIGQ = 1
 9508  033104  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9509                                  ;REPORT SIGQ NOT SET
 9510  033110                                  ERRDF   63,EM63,ERR7
   (4) 033110  104455                                                              TRAP    C$ERDF
   (5) 033112  000077                                                              .WORD   63
   (5) 033114  014726                                                              .WORD   EM63
   (5) 033116  020224                                                              .WORD   ERR7
 9511  033120                              6$:
 9512  033120                                  ENDSUB
   (3) 033120                                                                  L10066:
   (3) 033120  104403                                                              TRAP    C$ESUB
 9513
 9514                                  ;-------------------------------------------------------------------------------
 9515                                  ; SET BPOLL IN REG 12, TO LIGHT LED ONLY
 9516                                  ;-------------------------------------------------------------------------------
 9517  033122                                  BGNSUB
   (3) 033122                                                                  T23.4:
   (3) 033122  104402                                                              TRAP    C$BSUB
 9518  033124  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9519  033130  012737  000012  002364          MOV     #12,REGNUM      ;SET LU REG NO. = 12
 9520  033136  112737  000100  002352          MOVB    #BPOLL,WRIBYT   ;SET BPOLL IN LU REG 12
 9521  033144  004737  003422                  JSR     PC,WRITLU
 9522  033150                                  ENDSUB
   (3) 033150                                                                  L10067:
   (3) 033150  104403                                                              TRAP    C$ESUB
 9523
 9524                                  ;-------------------------------------------------------------------------------
 9525                                  ; SET DTR IN REG 13, CHECK FOR DTR AND MODR SET IN REG 13
 9526                                  ;-------------------------------------------------------------------------------
 9527  033152                                  BGNSUB
   (3) 033152                                                                  T23.5:
   (3) 033152  104402                                                              TRAP    C$BSUB
 9528  033154  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9529  033160  012737  000013  002364          MOV     #13,REGNUM      ;SET REG NO. = 13
 9530  033166  112737  000100  002352          MOVB    #DTR,WRIBYT
 9531  033174  004737  003422                  JSR     PC,WRITLU       ;SET DTR IN REG 13
 9532  033200  004737  003344                  JSR     PC,READLU       ;READ REG 13
 9533  033204  132737  000100  002350          BITB    #DTR,REDBYT     ;SEE IF DTR = 1
 9534  033212  001010                          BNE     6$              ;BR IF DTR = 1
 9535  033214  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9536                                  ;REPORT DTR NOT SET
 9537  033220                                  ERRDF   55,EM55,ERR7
   (4) 033220  104455                                                              TRAP    C$ERDF
   (5) 033222  000067                                                              .WORD   55
   (5) 033224  014563                                                              .WORD   EM55
   (5) 033226  020224                                                              .WORD   ERR7
 9538  033230                                  ESCAPE  SUB
   (3) 033230  104410                                                              TRAP    C$ESCAPE
   (3) 033232  000026                                                              .WORD   L10070-.
 9539  033234  132737  000010  002350  6$:     BITB    #MODR,REDBYT    ;SEE IF MODR = 1
 9540  033242  001006                          BNE     12$             ;BR IF MODR = 1
 9541  033244  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
```

```
 9542                                   ;REPORT MODR NOT SET
 9543  033250                                  ERRDF   57,EM57,ERR7
  (4)  033250  104455                                                    TRAP    C$ERDF
  (5)  033252  000071                                                    .WORD   57
  (5)  033254  014614                                                    .WORD   EM57
  (5)  033256  020224                                                    .WORD   ERR7
 9544  033260                          12$:
 9545  033260                                  ENDSUB
  (3)  033260                                                       L10070:
  (3)  033260  104403                                                    TRAP    C$ESUB
 9546
 9547                                   ;-------------------------------------------------------------------------------
 9548                                   ; SET SELFR IN REG 13, CHK FOR SIGR SET IN REG 17
 9549                                   ;-------------------------------------------------------------------------------
 9550  033262                                  BGNSUB
  (3)  033262                                                       T23.6:
  (3)  033262  104402                                                    TRAP    C$BSUB
 9551  033264  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9552  033270  112737  000040  002352          MOVB    #SELFR,WRIBYT
 9553  033276  012737  000013  002364          MOV     #13,REGNUM      ;SET REG NO. = 13
 9554  033304  004737  003422                  JSR     PC,WRITLU       ;SET SELFR IN REG 13
 9555  033310  012737  000017  002364          MOV     #17,REGNUM      ;SET REG NO. = 17
 9556  033316  004737  003344                  JSR     PC,READLU       ;READ REG 17
 9557  033322  132737  000200  002350          BITB    #SIGR,REDBYT    ;SEE IF SIGR = 1
 9558  033330  001006                          BNE     6$              ;BR IF SIGR = 1
 9559  033332  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9560                                  ;REPORT SIGR NOT SET
 9561  033336                                  ERRDF   64,EM64,ERR7
  (4)  033336  104455                                                    TRAP    C$ERDF
  (5)  033340  000100                                                    .WORD   64
  (5)  033342  014743                                                    .WORD   EM64
  (5)  033344  020224                                                    .WORD   ERR7
 9562  033346                          6$:
 9563  033346                                  ENDSUB
  (3)  033346                                                       L10071:
  (3)  033346  104403                                                    TRAP    C$ESUB
 9564
 9565                                   ;-------------------------------------------------------------------------------
 9566                                   ; SET HDX IN REG 13, CHK FOR HDX SET IN REG 13
 9567                                   ;-------------------------------------------------------------------------------
 9568  033350                                  BGNSUB
  (3)  033350                                                       T23.7:
  (3)  033350  104402                                                    TRAP    C$BSUB
 9569  033352  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9570  033356  112737  000020  002352          MOVB    #HDX,WRIBYT
 9571  033364  012737  000013  002364          MOV     #13,REGNUM      ;SET REG NO. = 13
 9572  033372  004737  003422                  JSR     PC,WRITLU       ;SET HDX IN REG 13
 9573  033376  004737  003344                  JSR     PC,READLU       ;READ REG 13
 9574  033402  132737  000020  002350          BITB    #HDX,REDBYT     ;SEE IF HDX = 1
 9575  033410  001006                          BNE     6$              ;BR IF HDX = 1
 9576  033412  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9577                                  ;REPORT HDX NOT SET
 9578  033416                                  ERRDF   58,EM58,ERR7
  (4)  033416  104455                                                    TRAP    C$ERDF
  (5)  033420  000072                                                    .WORD   58
  (5)  033422  014631                                                    .WORD   EM58
```

```
  (5)  033424  020224                                                                  .WORD   ERR7
9579   033426                          6$:
9580   033426                                  ENDSUB
  (3)  033426                                                                L10072:
  (3)  033426  104403                                                               TRAP    C$ESUB
9581
9582                                    ;-----------------------------------------------------------------------
9583                                    ; SET MAINT1 IN REG 13, CHK FOR TEST MODE SET IN REG 17
9584                                    ;-----------------------------------------------------------------------
9585   033430                                   BGNSUB
  (3)  033430                                                                T23.8:
  (3)  033430  104402                                                               TRAP    C$BSUB
9586   033432  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
9587   033436  112737  000010  002352          MOVB    #MAINT1,WRIBYT
9588   033444  012737  000013  002364          MOV     #13,REGNUM      ;SET REG NO. = 13
9589   033452  004737  003422                  JSR     PC,WRITLU       ;SET MAINT1 IN REG 13
9590   033456  012737  000017  002364          MOV     #17,REGNUM      ;SET REG NO. = 17
9591   033464  004737  003344                  JSR     PC,READLU       ;READ REG 17
9592   033470  132737  000004  002350          BITB    #TESTMD,REDBYT  ;SEE IF TESTMD = 1
9593   033476  001006                          BNE     6$              ;BR IF TESTMD = 1
9594   033500  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
9595                          ;REPORT TEST MODE NOT SET BY MAINT1
9596   033504                                  ERRDF   52,EM52,ERR7
  (4)  033504  104455                                                               TRAP    C$ERDF
  (5)  033506  000064                                                               .WORD   52
  (5)  033510  014451                                                               .WORD   EM52
  (5)  033512  020224                                                               .WORD   ERR7
9597   033514                          6$:
9598   033514                                  ENDSUB
  (3)  033514                                                                L10073:
  (3)  033514  104403                                                               TRAP    C$ESUB
9599
9600                                    ;-----------------------------------------------------------------------
9601                                    ; SET SELSBY IN REG 13, CHK FOR STBY SET IN REG 13
9602                                    ;-----------------------------------------------------------------------
9603   033516                                   BGNSUB
  (3)  033516                                                                T23.9:
  (3)  033516  104402                                                               TRAP    C$BSUB
9604   033520  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
9605   033524  112737  000002  002352          MOVB    #SELSBY,WRIBYT
9606   033532  012737  000013  002364          MOV     #13,REGNUM      ;SET REG NO. = 13
9607   033540  004737  003422                  JSR     PC,WRITLU       ;SET SELSBY IN REG 13
9608   033544  004737  003344                  JSR     PC,READLU       ;READ REG 13
9609   033550  132737  000002  002350          BITB    #STBY,REDBYT    ;SEE IF STBY = 1
9610   033556  001006                          BNE     6$              ;BR IF STBY = 1
9611   033560  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
9612                          ;REPORT STBY NOT SET
9613   033564                                  ERRDF   59,EM59,ERR7
  (4)  033564  104455                                                               TRAP    C$ERDF
  (5)  033566  000073                                                               .WORD   59
  (5)  033570  014645                                                               .WORD   EM59
  (5)  033572  020224                                                               .WORD   ERR7
9614   033574                          6$:
9615   033574                                  ENDSUB
  (3)  033574                                                                L10074:
  (3)  033574  104403                                                               TRAP    C$ESUB
```

```
                        ;-------------------------------------------------------------------------------
                        ; DO MASTER CLEAR, LOAD 2 TSOM'S INTO TX SILO, CLOCK TRANSMITTER UNTIL ACTIVE,
                        ;    CHECK FOR RTS, CS, CARR SET IN REG 13
                        ;-------------------------------------------------------------------------------
                              BGNSUB
                                                                                        T23.10:
                                                                                              TRAP    C$BSUB
                              JSR     PC,INITRN       ;MST CLR, LOAD SOM'S, CLK TRANSMITTER
                              000
                              000
                              MOV     #13,REGNUM      ;SET REG NO. = 13
                              JSR     PC,READLU       ;READ REG 13
                              BITB    #RTS,REDBYT     ;SEE IF RTS = 1
                              BNE     6$              ;BR IF RTS = 1
                              JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT RTS NOT SET
                              ERRDF   60,EM60,ERR7
                                                                                              TRAP    C$ERDF
                                                                                              .WORD   60
                                                                                              .WORD   EM60
                                                                                              .WORD   ERR7
                              ESCAPE  SUB
                                                                                              TRAP    C$ESCAPE
                                                                                              .WORD   L10075-.
6$:                           BITB    #CS,REDBYT      ;SEE IF CS = 1
                              BNE     9$              ;BR IF CS = 1
                              JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT CS NOT SET
                              ERRDF   61,EM61,ERR7
                                                                                              TRAP    C$ERDF
                                                                                              .WORD   61
                                                                                              .WORD   EM61
                                                                                              .WORD   ERR7
                              ESCAPE  SUB
                                                                                              TRAP    C$ESCAPE
                                                                                              .WORD   L10075-.
9$:                           BITB    #CARR,REDBYT    ;SEE IF CARR = 1
                              BNE     12$             ;BR IF CARR = 1
                              JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT CARR NOT SET
                              ERRDF   62,EM62,ERR7
                                                                                              TRAP    C$ERDF
                                                                                              .WORD   62
                                                                                              .WORD   EM62
                                                                                              .WORD   ERR7
12$:
                              ENDSUB
                                                                                    L10075:
                                                                                              TRAP    C$ESUB

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

```
 9650
 9651
 9652
 9653
 9654                                   ;;*************************************************************************
 9655                                   .SBTTL      TEST 24 - DATA TEST - BIT MODE, NO ERR DET
 9656                                   ;*
 9657                                   ;* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH ERROR DETECTION
 9658                                   ;* INHIBITED. THE MESSAGE CONSISTS OF 2 FLAGS, PAT A REPEATED 3 TIMES,
 9659                                   ;* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
 9660                                   ;* THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
 9661                                   ;* SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
 9662                                   ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
 9663                                   ;* TEST WILL NOT BE RUN.
 9664                                   ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
 9665                                   ;*               375,373,367,357,337,277,177
 9666                                   ;* 8-BIT CHARACTERS ARE USED.
 9667                                   ;;*************************************************************************
 9668  033736                           BGNTST
  (3)  033736                                                                          T24::
 9669  033736  012737  000030  002434           MOV     #24.,TSTNUM     ;SET TEST NO.
 9670  033744  012737  034124  002346           MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 9671  033752  004737  003276                   JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9672  033756  004737  010766                   JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
 9673  033762  000000                           0
 9674  033764  013737  002422  034002           MOV     MODINT,6$       ;SET MODEM SELECTION
 9675  033772  004737  010304                   JSR     PC,SETUP        ;PROGRAM THE USYRT
 9676  033776  000000                           000
 9677  034000  000300                           CRC2!CRC1               ;BIT MODE, NO ERR DET
 9678  034002  000000                   6$:     .WORD   0               ;MODEM SELECTION GOES HERE
 9679  034004  000000                           000
 9680  034006  004737  010156                   JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 9681  034012  012737  001177  003150           MOV     #RXEBL!177,RCVBUF+118. ; SET LAST DATA CHAR IN BUFFER
 9682  034020  142777  000010  146412           BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
 9683  034026  012737  000012  002364           MOV     #12,REGNUM      ;SET LU REG NO. = 12
 9684  034034  012703  002762                   MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 9685  034040  013702  002264           9$:     MOV     TCOUNT,R2               ;INIT TIMER
 9686  034044  004737  003344           10$:    JSR     PC,READLU       ;READ REG 12
 9687  034050  132737  000020  002350           BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 9688  034056  001011                           BNE     12$             ;BR IF YES
 9689  034060  005202                           INC     R2              ;INCREMENT TIMER
 9690  034062  001370                           BNE     10$             ;BR IF NO TIME-OUT YET
 9691  034064  004737  004200                   JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9692                                   ;REPORT IRDY NOT SET
 9693  034070                                   ERRDF   17,EM17,ERR7
  (4)  034070  104455                                                                  TRAP    C$ERDF
  (5)  034072  000021                                                                  .WORD   17
  (5)  034074  013436                                                                  .WORD   EM17
  (5)  034076  020224                                                                  .WORD   ERR7
 9694  034100  000411                           BR      24$             ;ESCAPE TO END OF TEST
 9695  034102  012337  034112           12$:    MOV     (R3)+,16$
 9696  034106  004737  007266                   JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 9697  034112  000000                   16$:    0
 9698  034114  000000                           0
 9699  034116  020327  003152                   CMP     R3,#RCVBUF+120. ;SEE IF ALL CHARS CHECKED YET
 9700  034122  103746                           BLO     9$              ;BR IF NOT YET
```

```
 9701  034124                          24$:
 9702  034124                          ENDTST
  (3)  034124                                                              L10076:
  (3)  034124  104401                                                              TRAP    C$ETST
 9703
 9704
 9705
 9706
 9707
 9708                                  ;;*****************************************************************************
 9709                                  .SBTTL      TEST 25 - DATA TEST - CHAR MODE, NO ERR DET
 9710                                  ;*
 9711                                  ;* A MESSAGE IS INITIATED IN CHAR MODE, WITH ERROR DETECTION
 9712                                  ;* INHIBITED. THE MESSAGE CONSISTS OF 2 SYNCHS, PAT A REPEATED 3 TIMES,
 9713                                  ;* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
 9714                                  ;* THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
 9715                                  ;* SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
 9716                                  ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
 9717                                  ;* TEST WILL NOT BE RUN.
 9718                                  ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
 9719                                  ;*               375,373,367,357,337,277,177
 9720                                  ;* 8-BIT CHARACTERS ARE USED.
 9721                                  ;;*****************************************************************************
 9722  034126                          BGNTST
  (3)  034126                                                              T25::
 9723  034126  012737  000031  002434          MOV     #25.,TSTNUM     ;SET TEST NO.
 9724  034134  012737  034314  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 9725  034142  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9726  034146  004737  010766                  JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
 9727  034152  000000                          0
 9728  034154  013737  002422  034172          MOV     MODINT,6$       ;SET MODEM SELECTION
 9729  034162  004737  010304                  JSR     PC,SETUP        ;PROGRAM THE USYRT
 9730  034166  000226                          SYNCH
 9731  034170  000311                          CRC2!CRC1!STRIP!DDCMP   ;CHAR MODE, NO ERR DET
 9732  034172  000000                  6$:     .WORD   0               ;MODEM SELECTION GOES HERE
 9733  034174  000000                          000
 9734  034176  004737  010156                  JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 9735  034202  012737  000177  003150          MOV     #177,RCVBUF+118. ;SET LAST DATA CHAR IN BUFFER
 9736  034210  142777  000010  146222          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
 9737  034216  012737  000012  002364          MOV     #12,REGNUM      ;SET LU REG NO. = 12
 9738  034224  012703  002762                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 9739  034230  013702  002264          9$:     MOV     TCOUNT,R2               ;INIT TIMER
 9740  034234  004737  003344          10$:    JSR     PC,READLU       ;READ REG 12
 9741  034240  132737  000020  002350          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 9742  034246  001011                          BNE     12$             ;BR IF YES
 9743  034250  005202                          INC     R2              ;INCREMENT TIMER
 9744  034252  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
 9745  034254  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9746                                  ;REPORT IRDY NOT SET
 9747  034260                          ERRDF   17,EM17,ERR7
  (4)  034260  104455                                                              TRAP    C$ERDF
  (5)  034262  000021                                                              .WORD   17
  (5)  034264  013436                                                              .WORD   EM17
  (5)  034266  020224                                                              .WORD   ERR7
 9748  034270  000411                          BR      24$             ;ESCAPE TO END OF TEST
 9749  034272  012337  034302          12$:    MOV     (R3)+,16$
```

```
 9750  034276  004737  007266          JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 9751  034302  000000          16$:    0
 9752  034304  000000                  0
 9753  034306  020327  003152          CMP     R3,#RCVBUF+120. ;SEE IF ALL CHARS CHECKED YET
 9754  034312  103746                  BLO     9$              ;BR IF NOT YET
 9755  034314                  24$:
 9756  034314                  ENDTST
   (3) 034314                                                          L10077:
   (3) 034314  104401                                                          TRAP    C$ETST
 9757
 9758
 9759
 9760
 9761
 9762                          ;;*************************************************************************
 9763                          .SBTTL      TEST 26 - DATA TEST - BIT MODE, CRC-CCITT-1
 9764                          ;*
 9765                          ;* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-1 ERROR
 9766                          ;* DETECTION. THE MESSAGE CONSISTS OF 2 FLAGS, PAT A REPEATED 3 TIMES,
 9767                          ;* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
 9768                          ;* THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
 9769                          ;* SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
 9770                          ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
 9771                          ;* TEST WILL NOT BE RUN.
 9772                          ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
 9773                          ;*               375,373,367,357,337,277,177
 9774                          ;* 8-BIT CHARACTERS ARE USED.
 9775                          ;;*************************************************************************
 9776  034316                  BGNTST
   (3) 034316                                                                  T26::
 9777  034316  012737  000032  002434          MOV     #26.,TSTNUM     ;SET TEST NO.
 9778  034324  012737  034504  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 9779  034332  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9780  034336  004737  010766                  JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
 9781  034342  000000                          0
 9782  034344  013737  002422  034362          MOV     MODINT,6$       ;SET MODEM SELECTION
 9783  034352  004737  010304                  JSR     PC,SETUP        ;PROGRAM THE USYRT
 9784  034356  000000                          000
 9785  034360  000000                          000                     ;BIT MODE CRC-CCITT-1
 9786  034362  000000          6$:     .WORD   0                       ;MODEM SELECTION GOES HERE
 9787  034364  000000                          000
 9788  034366  004737  010156                  JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 9789  034372  012737  101177  003150          MOV     #CRCCHK!RXEBL!177,RCVBUF+118. ;SET LAST DATA CHAR IN BUFFER
 9790  034400  142777  000010  146032          BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
 9791  034406  012737  000012  002364          MOV     #12,REGNUM      ;SET LU REG NO. = 12
 9792  034414  012703  002762                  MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 9793  034420  013702  002264          9$:     MOV     TCOUNT,R2               ;INIT TIMER
 9794  034424  004737  003344          10$:    JSR     PC,READLU       ;READ REG 12
 9795  034430  132737  000020  002350          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 9796  034436  001011                          BNE     12$             ;BR IF YES
 9797  034440  005202                          INC     R2              ;INCREMENT TIMER
 9798  034442  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
 9799  034444  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9800                          ;REPORT IRDY NOT SET
 9801  034450                  ERRDF   17,EM17,ERR7
   (4) 034450  104455                                                          TRAP    C$ERDF
```

```
  (5)  034452  000021                                                    .WORD   17
  (5)  034454  013436                                                    .WORD   EM17
  (5)  034456  020224                                                    .WORD   ERR7
 9802  034460  000411                    BR      24$             ;ESCAPE TO END OF TEST
 9803  034462  012337  034472    12$:    MOV     (R3)+,16$
 9804  034466  004737  007266            JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 9805  034472  000000    16$:    0
 9806  034474  000000            0
 9807  034476  020327  003152            CMP     R3,#RCVBUF+120. ;SEE IF ALL CHARS CHECKED YET
 9808  034502  103746            BLO     9$              ;BR IF NOT YET
 9809  034504                    24$:
 9810  034504                    ENDTST
  (3)  034504                                                    L10100:
  (3)  034504  104401                                                    TRAP    C$ETST
 9811
 9812
 9813
 9814
 9815
 9816                    ;;*************************************************************************
 9817                    .SBTTL      TEST 27 - DATA TEST - BIT MODE, CRC-CCITT-0
 9818                    ;*
 9819                    ;* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-0 ERROR
 9820                    ;* DETECTION. THE MESSAGE CONSISTS OF 2 FLAGS, PAT A REPEATED 3 TIMES,
 9821                    ;* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
 9822                    ;* THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
 9823                    ;* SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
 9824                    ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
 9825                    ;* TEST WILL NOT BE RUN.
 9826                    ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
 9827                    ;*               375,373,367,357,337,277,177
 9828                    ;* 8-BIT CHARACTERS ARE USED.
 9829                    ;;*************************************************************************
 9830  034506            BGNTST
  (3)  034506                                                            T27::
 9831  034506  012737  000033  002434    MOV     #27.,TSTNUM     ;SET TEST NO.
 9832  034514  012737  034674  002346    MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 9833  034522  004737  003276            JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9834  034526  004737  010766            JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
 9835  034532  000000                    0
 9836  034534  013737  002422  034552    MOV     MODINT,6$       ;SET MODEM SELECTION
 9837  034542  004737  010304            JSR     PC,SETUP        ;PROGRAM THE USYRT
 9838  034546  000000                    000
 9839  034550  000100                    CRC1                    ;BIT MODE, CRC-CCITT-0
 9840  034552  000000    6$:     .WORD   0               ;MODEM SELECTION GOES HERE
 9841  034554  000000                    000
 9842  034556  004737  010156            JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 9843  034562  012737  101177  003150    MOV     #CRCCHK!RXEBL!177,RCVBUF+118. ;SET LAST DATA CHAR IN BUFFER
 9844  034570  142777  000010  145642    BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
 9845  034576  012737  000012  002364    MOV     #12,REGNUM      ;SET LU REG NO. = 12
 9846  034604  012703  002762            MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 9847  034610  013702  002264    9$:     MOV     TCOUNT,R2               ;INIT TIMER
 9848  034614  004737  003344    10$:    JSR     PC,READLU       ;READ REG 12
 9849  034620  132737  000020  002350    BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 9850  034626  001011                    BNE     12$             ;BR IF YES
 9851  034630  005202                    INC     R2              ;INCREMENT TIMER
```

```
 9852  034632  001370                        BNE     10$             ;BR IF NO TIME-OUT YET
 9853  034634  004737  004200                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9854                                  ;REPORT IRDY NOT SET
 9855  034640                                ERRDF   17,EM17,ERR7
   (4) 034640  104455                                                        TRAP    C$ERDF
   (5) 034642  000021                                                        .WORD   17
   (5) 034644  013436                                                        .WORD   EM17
   (5) 034646  020224                                                        .WORD   ERR7
 9856  034650  000411                        BR      24$             ;ESCAPE TO END OF TEST
 9857  034652  012337  034662        12$:    MOV     (R3)+,16$
 9858  034656  004737  007266                JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 9859  034662  000000        16$:    0
 9860  034664  000000                        0
 9861  034666  020327  003152                CMP     R3,#RCVBUF+120. ;SEE IF ALL CHARS CHECKED YET
 9862  034672  103746                        BLO     9$              ;BR IF NOT YET
 9863  034674                        24$:
 9864  034674                                ENDTST
   (3) 034674                                                              L10101:
   (3) 034674  104401                                                        TRAP    C$ETST
 9865
 9866
 9867
 9868
 9869
 9870                                  ;;*************************************************************************
 9871                                  .SBTTL      TEST 28 - DATA TEST - CHAR MODE, CRC-16
 9872                                  ;*
 9873                                  ;* A MESSAGE IS INITIATED IN CHAR MODE, WITH CRC-16 ERROR
 9874                                  ;* DETECTION. THE MESSAGE CONSISTS OF 2 SYNCHS, PAT A REPEATED 3 TIMES,
 9875                                  ;* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
 9876                                  ;* THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
 9877                                  ;* SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
 9878                                  ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
 9879                                  ;* TEST WILL NOT BE RUN.
 9880                                  ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
 9881                                  ;*               375,373,367,357,337,277,177
 9882                                  ;* 8-BIT CHARACTERS ARE USED.
 9883                                  ;*************************************************************************
 9884  034676                        BGNTST
   (3) 034676                                                              T28::
 9885  034676  012737  000034  002434        MOV     #28.,TSTNUM     ;SET TEST NO.
 9886  034704  012737  035064  002346        MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 9887  034712  004737  003276                JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9888  034716  004737  010766                JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
 9889  034722  000000                        0
 9890  034724  013737  002422  034742        MOV     MODINT,6$       ;SET MODEM SELECTION
 9891  034732  004737  010304                JSR     PC,SETUP        ;PROGRAM THE USYRT
 9892  034736  000226                        SYNCH
 9893  034740  000011                        STRIP!DDCMP
 9894  034742  000000        6$:     .WORD   0               ;MODEM SELECTION GOES HERE
 9895  034744  000000                        000
 9896  034746  004737  010156                JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 9897  034752  012737  100577  003150        MOV     #CRCCHK!RXBCC!177,RCVBUF+118. ;SET LAST DATA CHAR IN BUFFER
 9898  034760  142777  000010  145452        BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
 9899  034766  012737  000012  002364        MOV     #12,REGNUM      ;SET LU REG NO. = 12
 9900  034774  012703  002762                MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
```

```
 9901  035000  013702  002264          9$:     MOV     TCOUNT,R2               ;INIT TIMER
 9902  035004  004737  003344          10$:    JSR     PC,READLU       ;READ REG 12
 9903  035010  132737  000020  002350          BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 9904  035016  001011                          BNE     12$             ;BR IF YES
 9905  035020  005202                          INC     R2              ;INCREMENT TIMER
 9906  035022  001370                          BNE     10$             ;BR IF NO TIME-OUT YET
 9907  035024  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9908                                  ;REPORT IRDY NOT SET
 9909  035030                                  ERRDF   17,EM17,ERR7
   (4) 035030  104455                                                              TRAP    C$ERDF
   (5) 035032  000021                                                              .WORD   17
   (5) 035034  013436                                                              .WORD   EM17
   (5) 035036  020224                                                              .WORD   ERR7
 9910  035040  000411                          BR      24$             ;ESCAPE TO END OF TEST
 9911  035042  012337  035052          12$:    MOV     (R3)+,16$
 9912  035046  004737  007266                  JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 9913  035052  000000                  16$:    0
 9914  035054  000000                          0
 9915  035056  020327  003152                  CMP     R3,#RCVBUF+120. ;SEE IF ALL CHARS CHECKED YET
 9916  035062  103746                          BLO     9$              ;BR IF NOT YET
 9917  035064                          24$:
 9918  035064                          ENDTST
   (3) 035064                                                                    L10102:
   (3) 035064  104401                                                              TRAP    C$ETST
 9919
 9920
 9921
 9922
 9923
 9924                                  ;;*************************************************************************
 9925                                  .SBTTL      TEST 29 - DATA TEST - CHAR MODE, ODD VRC
 9926                                  ;*
 9927                                  ;* A MESSAGE IS INITIATED IN CHAR MODE, WITH ODD VRC ERROR DETECTION
 9928                                  ;* SELECTED. THE MESSAGE CONSISTS OF 2 SYNCHS, PAT A REPEATED 3 TIMES,
 9929                                  ;* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
 9930                                  ;* THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
 9931                                  ;* SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
 9932                                  ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
 9933                                  ;* TEST WILL NOT BE RUN.
 9934                                  ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
 9935                                  ;*               375,373,367,357,337,277,177
 9936                                  ;* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
 9937                                  ;*************************************************************************
 9938  035066                          BGNTST
   (3) 035066                                                                    T29::
 9939  035066  012737  000035  002434          MOV     #29.,TSTNUM     ;SET TEST NO.
 9940  035074  012737  035250  002346          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 9941  035102  004737  003276                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9942  035106  004737  010766                  JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
 9943  035112  000000                          0
 9944  035114  013737  002422  035132          MOV     MODINT,6$       ;SET MODEM SELECTION
 9945  035122  004737  010304                  JSR     PC,SETUP        ;PROGRAM THE USYRT
 9946  035126  000026                          026
 9947  035130  000111                          CRC1!STRIP!DDCMP
 9948  035132  000000                  6$:     .WORD   0               ;MODEM SELECTION GOES HERE
 9949  035134  000347                          TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
```

```
 9950  035136  004737  010156                    JSR     PC,LODATA       ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 9951  035142  142777  000010  145270            BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP, CLOCK MSG
 9952  035150  012737  000012  002364            MOV     #12,REGNUM      ;SET LU REG NO. = 12
 9953  035156  012703  002762                    MOV     #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
 9954  035162  013702  002264            9$:     MOV     TCOUNT,R2               ;INIT TIMER
 9955  035166  004737  003344            10$:    JSR     PC,READLU       ;READ REG 12
 9956  035172  132737  000020  002350            BITB    #IRDY,REDBYT    ;SEE IF IRDY IS SET YET
 9957  035200  001011                            BNE     12$             ;BR IF YES
 9958  035202  005202                            INC     R2              ;INCREMENT TIMER
 9959  035204  001370                            BNE     10$             ;BR IF NO TIME-OUT YET
 9960  035206  004737  004200                    JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
 9961                                    ;REPORT IRDY NOT SET
 9962  035212                                    ERRDF   17,EM17,ERR7
   (4) 035212  104455                                                                      TRAP    C$ERDF
   (5) 035214  000021                                                                      .WORD   17
   (5) 035216  013436                                                                      .WORD   EM17
   (5) 035220  020224                                                                      .WORD   ERR7
 9963  035222  000412                            BR      24$             ;ESCAPE TO END OF TEST
 9964  035224  112337  035236            12$:    MOVB    (R3)+,16$       ;GET AN EXPECTED DATA BYTE
 9965  035230  005203                            INC     R3              ;INCREMENT POINTER
 9966  035232  004737  007266                    JSR     PC,CKDATA       ;COMPARE RCV'D DATA CHAR TO EXPECTED
 9967  035236  100000            16$:    BCCCHK
 9968  035240  000000                            0
 9969  035242  020327  003152                    CMP     R3,#RCVBUF+120. ;SEE IF ALL CHARS CHECKED YET
 9970  035246  103745                            BLO     9$              ;BR IF NOT YET
 9971  035250                            24$:
 9972  035250                            ENDTST
   (3) 035250                                                                      L10103:
   (3) 035250  104401                                                                      TRAP    C$ETST
 9973
 9974
 9975
 9976
 9977
 9978                                    ;;*************************************************************************
 9979                                    .SBTTL      TEST 30 - DATA TEST - CHAR MODE, EVEN VRC
 9980                                    ;*
 9981                                    ;* A MESSAGE IS INITIATED IN CHAR MODE, WITH EVEN VRC ERROR DETECTION
 9982                                    ;* SELECTED. THE MESSAGE CONSISTS OF 2 SYNCHS, PAT A REPEATED 3 TIMES,
 9983                                    ;* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
 9984                                    ;* THE TEST WILL BE RUN WITH THE INTEGRAL MODEM SELECTED (EITHER HI OR LO
 9985                                    ;* SPEED). IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
 9986                                    ;* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
 9987                                    ;* TEST WILL NOT BE RUN.
 9988                                    ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
 9989                                    ;*               375,373,367,357,337,277,177
 9990                                    ;* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
 9991                                    ;;*************************************************************************
 9992  035252                            BGNTST
   (3) 035252                                                                      T30::
 9993  035252  012737  000036  002434            MOV     #30.,TSTNUM     ;SET TEST NO.
 9994  035260  012737  035434  002346            MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
 9995  035266  004737  003276                    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 9996  035272  004737  010766                    JSR     PC,CKLPBK       ;CHECK LOOPBACK, GET MODEM SELECTION
 9997  035276  000000                            0
 9998  035300  013737  002422  035316            MOV     MODINT,6$       ;SET MODEM SELECTION
```

```
 9999  035306  004737  010304                    JSR     PC,SETUP          ;PROGRAM THE USYRT
10000  035312  000026                            026
10001  035314  000211                            CRC2!STRIP!DDCMP
10002  035316  000000                    6$:     .WORD   0                 ;MODEM SELECTION GOES HERE
10003  035320  000347                            TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
10004  035322  004737  010156                    JSR     PC,LODATA         ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
10005  035326  142777  000010  145104            BICB    #LULOOP,@BSEL1    ;CLEAR LULOOP, CLOCK MSG
10006  035334  012737  000012  002364            MOV     #12,REGNUM        ;SET LU REG NO. = 12
10007  035342  012703  002762                    MOV     #RCVBUF,R3        ;GET POINTER TO RCV MSG BUF
10008  035346  013702  002264            9$:     MOV     TCOUNT,R2                 ;INIT TIMER
10009  035352  004737  003344            10$:    JSR     PC,READLU         ;READ REG 12
10010  035356  132737  000020  002350            BITB    #IRDY,REDBYT      ;SEE IF IRDY IS SET YET
10011  035364  001011                            BNE     12$               ;BR IF YES
10012  035366  005202                            INC     R2                ;INCREMENT TIMER
10013  035370  001370                            BNE     10$               ;BR IF NO TIME-OUT YET
10014  035372  004737  004200                    JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
10015                                    ;REPORT IRDY NOT SET
10016  035376                                    ERRDF   17,EM17,ERR7
  (4)  035376  104455                                                                       TRAP    C$ERDF
  (5)  035400  000021                                                                       .WORD   17
  (5)  035402  013436                                                                       .WORD   EM17
  (5)  035404  020224                                                                       .WORD   ERR7
10017  035406  000412                            BR      24$               ;ESCAPE TO END OF TEST
10018  035410  112337  035422            12$:    MOVB    (R3)+,16$         ;GET AN EXPECTED DATA CHAR
10019  035414  005203                            INC     R3                ;INCREMENT POINTER
10020  035416  004737  007266                    JSR     PC,CKDATA         ;COMPARE RCV'D DATA CHAR TO EXPECTED
10021  035422  100000                    16$:    BCCCHK
10022  035424  000000                            0
10023  035426  020327  003152                    CMP     R3,#RCVBUF+120.   ;SEE IF ALL CHARS CHECKED YET
10024  035432  103745                            BLO     9$                ;BR IF NOT YET
10025  035434                            24$:
10026  035434                            ENDTST
  (3)  035434                                                                       L10104:
  (3)  035434  104401                                                                       TRAP    C$ETST
10027
10028
10029
10030
10031
10032                                     ;;*************************************************************************
10033                                     .SBTTL      TEST 31 - CONTIGUOUS ONES IN SEC. STA. ADRS. MODE, BIT MODE
10034                                     ;*
10035                                     ;* IN THIS TEST, A MESSAGE CONSISTING OF 5 ONES CHARS (377 OCT)
10036                                     ;* IS SENT IN SECONDARY STATION ADDRESS MODE, WITH THE STATION ADRS
10037                                     ;* FOR THIS LINE = 377. THE PROGRAM CHECKS FOR CORRECT RECEPTION OF
10038                                     ;* THE FIRST CHARACTER (STATION ADDRESS) AND THE REMAINING 4
10039                                     ;* ONES CHARACTERS (DATA). THIS TEST EXERCISES THE SECONDARY STATION
10040                                     ;* ADDRESS LOGIC, AND CHECKS THAT THE SEC. STA. ADRS. CAN BE BIT-STUFFED
10041                                     ;* AND TRANSMITTED AND RECEIVED CORRECTLY.
10042                                     ;;*************************************************************************
10043  035436                             BGNTST
  (3)  035436                                                                       T31::
10044  035436  012737  035542  002346            MOV     #24$,RETADR       ;SET TEST EXIT ADDRESS FOR ERRORS
10045  035444  004737  005212                    JSR     PC,INITRN         ;MST CLR, LOAD 2 SOM'S
10046  035450  000377                            377
10047  035452  000320                            CRC2!CRC1!SECA            ;BIT MODE, NO ERROR DET, SEC ADR MODE
```

```
10048  035454  004737  010716                  JSR     PC,LODSIL       ;LOAD 5 377-CHARS INTO TX SILO
10049  035460  000377                          377
10050  035462  000005                          5
10051  035464  004737  010716                  JSR     PC,LODSIL       ;LOAD 2 EOM'S INTO TX SILO
10052  035470  001000                          TXEOM
10053  035472  000002                          2
10054  035474  004737  004726                  JSR     PC,STPLU        ;CLOCK MORE THAN ENTIRE MSG
10055  035500  000240                          160.
10056  035502  004737  007266                  JSR     PC,CKDATA       ;RCV SEC ADRS = 377
10057  035506  000377                          377
10058  035510  000000                          0
10059  035512  012701  000003                  MOV     #3,R1           ;RCV 3 MORE 377 CHARS
10060  035516  004737  007266          6$:     JSR     PC,CKDATA
10061  035522  000377                          377
10062  035524  000000                          0
10063  035526  005301                          DEC     R1
10064  035530  001372                          BNE     6$
10065  035532  004737  007266                  JSR     PC,CKDATA       ;RCV LAST 377 CHAR, CHK EBLK = 1
10066  035536  001377                          1377
10067  035540  000000                          0
10068  035542  004737  003276          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
10069  035546                          ENDTST
  (3)  035546                                                                   L10105:
  (3)  035546  104401                                                                   TRAP    C$ETST
10070
10071
10072
10073
10074
10075                                   ;;*************************************************************************
10076                                   .SBTTL      TEST 32 - DDCMP MESSAGE TEST - CHAR MODE
10077                                   ;*
10078                                   ;* IN THIS TEST, THREE USYRT MESSAGES ARE SENT TO SIMULATE A DDCMP HEADER,
10079                                   ;* DDCMP DATA MESSAGE, AND THE START OF A NEW DDCMP HEADER.
10080                                   ;* FIRST, THE DATA IN PATTERN A IS TRANSMITTED AND RECEIVED
10081                                   ;* AND THEN CRC (CRC-16) IS SENT, FOLLOWED BY THE DATA IN PATTERN A
10082                                   ;* AGAIN AND THE CRC ON THAT DATA, AND FINALLY THE DATA IN 'MSG1'' IS
10083                                   ;* SENT WITH ITS CORRESPONDING CRC.
10084                                   ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
10085                                   ;*                 375,373,367,357,337,277,177
10086                                   ;*   MSG1 = SYNCH,SYNCH,SYNCH,SYNCH,000,125,252,377,000,SYNCH,SYNCH
10087                                   ;;*************************************************************************
10088  035550                           BGNTST
  (3)  035550                                                                   T32::
10089  035550  012737  036326  002346           MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
10090                                  ;-------------------------------------------------------------------
10091                                  ; TRANSMIT AND RCV ENTIRE MSG
10092                                  ;-------------------------------------------------------------------
10093  035556  004737  005212                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
10094  035562  000226                          SYNCH
10095  035564  000011                          STRIP!DDCMP
10096  035566  004737  010554                  JSR     PC,LDBYTS       ;LOAD 20 WORDS OF PAT A INTO TX SILO
10097  035572  002557                          PATA
10098  035574  000024                          20.
10099  035576  004737  010716                  JSR     PC,LODSIL       ;LOAD AN EOM INTO TX SILO
10100  035602  001000                          TXEOM
```

```
10101  035604  000001                          1
10102  035606  004737  010554                  JSR     PC,LDBYTS       ;LOAD 20 WORDS OF PAT A INTO TX SILO
10103  035612  002557                          PATA
10104  035614  000024                          20.
10105  035616  004737  010716                  JSR     PC,LODSIL       ;LOAD 1 EOM INTO TX SILO
10106  035622  001000                          TXEOM
10107  035624  000001                          1
10108  035626  004737  010716                  JSR     PC,LODSIL       ;LOAD 3 SOM'S INTO TX SILO
10109  035632  000400                          TXSOM
10110  035634  000003                          3
10111  035636  004737  010476                  JSR     PC,LODMSG       ;LOAD MSG1 INTO TX SILO
10112  035642  002670                          MSG1
10113  035644  000013                          11.
10114  035646  004737  004726                  JSR     PC,STPLU        ;CLOCK HDR MSG AND CRC CHARS
10115  035652  000300                          192.
10116  035654  012737  000013  002364          MOV     #13,REGNUM      ;SET REG. NO. = 13
10117  035662  004737  003344                  JSR     PC,READLU       ;READ REG 13
10118  035666  032737  000040  002350          BIT     #RTS,REDBYT     ;SEE IF RTS SET
10119  035674  001010                          BNE     2$              ;BR IF RTS SET
10120  035676  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
10121                                  ;REPORT RTS NOT SET
10122  035702                                  ERRDF   60,EM60,ERR7
   (4) 035702  104455                                                          TRAP    C$ERDF
   (5) 035704  000074                                                          .WORD   60
   (5) 035706  014662                                                          .WORD   EM60
   (5) 035710  020224                                                          .WORD   ERR7
10123  035712  000137  036326                  JMP     24$             ;EXIT TEST
10124  035716  004737  004726          2$:     JSR     PC,STPLU        ;CLK DATA MSG AND FIRST CRC CHAR
10125  035722  000250                          168.
10126  035724  012703  000040                  MOV     #32.,R3         ;SET COUNTER FOR CHECKING RTS
10127  035730  004737  004726          4$:     JSR     PC,STPLU        ;CLOCK LINE UNIT FOR 1 CYCLE
10128  035734  000001                          1
10129  035736  004737  003344                  JSR     PC,READLU       ;READ REG 13
10130  035742  032737  000040  002350          BIT     #RTS,REDBYT     ;CHK FOR RTS SET
10131  035750  001007                          BNE     5$              ;BR IF RTS SET
10132  035752  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
10133                                  ;REPORT RTS NOT SET
10134  035756                                  ERRDF   60,EM60,ERR7
   (4) 035756  104455                                                          TRAP    C$ERDF
   (5) 035760  000074                                                          .WORD   60
   (5) 035762  014662                                                          .WORD   EM60
   (5) 035764  020224                                                          .WORD   ERR7
10135  035766  000557                          BR      24$
10136  035770  005303          5$:     DEC     R3              ;DECR COUNTER
10137  035772  001356                          BNE     4$              ;BR IF NOT DONE YET
10138                                  ;-----------------------------------------------------------
10139                                  ; READ AND CHK HEADER AND CRC
10140                                  ;-----------------------------------------------------------
10141  035774  012701  002557                  MOV     #PATA,R1        ;INIT PATTERN A POINTER
10142  036000  112137  036010          7$:     MOVB    (R1)+,8$        ;GET AN EXPECTED CHAR
10143  036004  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK A CHAR
10144  036010  000000                  8$:     .WORD   0
10145  036012  000000                          0
10146  036014  020127  002601                  CMP     R1,#PATB-2      ;SEE IF CHKING NEXT-TO-LAST CHAR YET
10147  036020  103767                          BLO     7$              ;BR IF NOT YET
10148  036022  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK CHAR, BCC=0
```

```
10149  036026  100277                          CRCCHK!277
10150  036030  000000                          0
10151  036032  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK LAST CHAR, BCC=1
10152  036036  100577                          CRCCHK!RXBCC!177
10153  036040  000000                          0
10154  036042  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK HI CRC BYTE
10155  036046  000156                          156
10156  036050  000000                          0
10157  036052  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK LO CRC BYTE
10158  036056  000236                          236
10159  036060  000000                          0
10160                                  ;-------------------------------------------------------------------
10161                                  ; READ AND CHK DATA MSG AND CRC
10162                                  ;-------------------------------------------------------------------
10163  036062  012701  002557                  MOV     #PATA,R1        ;INIT PATTERN A POINTER
10164  036066  112137  036076          9$:     MOVB    (R1)+,12$       ;GET AN EXPECTED CHAR
10165  036072  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK A CHAR
10166  036076  000000                  12$:    .WORD   0
10167  036100  000000                          0
10168  036102  020127  002601                  CMP     R1,#PATB-2      ;SEE IF CHKING NEXT-TO-LAST CHAR YET
10169  036106  103767                          BLO     9$              ;BR IF NOT YET
10170  036110  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK CHAR, BCC=0
10171  036114  100277                          CRCCHK!277
10172  036116  000000                          0
10173  036120  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK LAST CHAR, BCC=1
10174  036124  100577                          CRCCHK!RXBCC!177
10175  036126  000000                          0
10176  036130  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK HI CRC BYTE
10177  036134  000156                          156
10178  036136  000000                          0
10179  036140  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK LO CRC BYTE
10180  036144  000236                          236
10181  036146  000000                          0
10182                                  ;-------------------------------------------------------------------
10183                                  ; CLOCK 3RD MESSAGE ('MSG1'' DATA)
10184                                  ;-------------------------------------------------------------------
10185  036150  012737  000012  002364          MOV     #12,REGNUM      ;SET REG NO. = 12
10186  036156  112737  000200  002352          MOVB    #IC,WRIBYT      ;SET IC TO CLEAR RECEIVER FOR NEW MSG
10187  036164  004737  003422                  JSR     PC,WRITLU
10188  036170  012737  000013  002364          MOV     #13,REGNUM      ;RESTORE REG NO. TO 13
10189  036176  004737  004726                  JSR     PC,STPLU        ;CLOCK THE REST OF MSG
10190  036202  000150                          104.
10191  036204  004737  003344                  JSR     PC,READLU       ;READ REG 13
10192  036210  032737  000040  002350          BIT     #RTS,REDBYT     ;SEE IF RTS IS CLEARED
10193  036216  001407                          BEQ     14$             ;BR IF RTS CLEARED
10194  036220  004737  004200                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
10195                                  ;REPORT RTS NOT CLEARED
10196  036224                                  ERRDF   65,EM65,ERR7
  (4)  036224  104455                                                   TRAP    C$ERDF
  (5)  036226  000101                                                   .WORD   65
  (5)  036230  014760                                                   .WORD   EM65
  (5)  036232  020224                                                   .WORD   ERR7
10197  036234  000434                          BR      24$
10198                                  ;-------------------------------------------------------------------
10199                                  ;READ AND CHECK 3RD MESSAGE AND CRC
10200                                  ;-------------------------------------------------------------------
```

```
10201  036236  004737  007266          14$:    JSR     PC,CKDATA       ;READ AND CHECK 000 DATA CHAR
10202  036242  000000                          000
10203  036244  000000                          0
10204  036246  004737  007266                  JSR     PC,CKDATA       ;READ AND CHECK 125 DATA CHAR
10205  036252  000125                          125
10206  036254  000000                          0
10207  036256  004737  007266                  JSR     PC,CKDATA       ;READ AND CHECK 252 DATA CHAR
10208  036262  000252                          252
10209  036264  000000                          0
10210  036266  004737  007266                  JSR     PC,CKDATA       ;READ AND CHECK 377 DATA CHAR, AND BCC=0
10211  036272  100377                          CRCCHK!377
10212  036274  000000                          0
10213  036276  004737  007266                  JSR     PC,CKDATA       ;READ AND CHECK 000 DATA CHAR, AND BCC=1
10214  036302  100400                          CRCCHK!RXBCC!000
10215  036304  000000                          0
10216  036306  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK HI CRC BYTE
10217  036312  000160                          160
10218  036314  000000                          0
10219  036316  004737  007266                  JSR     PC,CKDATA       ;READ AND CHK LO CRC BYTE
10220  036322  000034                          034
10221  036324  000000                          0
10222  036326  004737  003276          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
10223  036332                          ENDTST
  (3)  036332                                                                  L10106:
  (3)  036332  104401                                                                  TRAP    C$ETST
10224
10225
10226
10227
10228
```

```
10230                           .SBTTL  HARDWARE PARAMETER CODING SECTION
10231
10232
10233
10234                           ;////////////////////////////////////////////////////////////////////////
10235                           ;/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
10236                           ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
10237                           ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
10238                           ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
10239                           ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
10240                           ;/ WITH THE OPERATOR.
10241                           ;////////////////////////////////////////////////////////////////////////
10242
10243  036334                           BGNHRD
   (3)  036334  000046                                                    .WORD L10107-L$HARD/2
   (3)  036336                                                          L$HARD::
10244
10245  036336                           GPRMA   ADDRES,2,0,160000,177776,YES
   (4)  036336  001031                                                    .WORD   T$CODE
   (4)  036340  036452                                                    .WORD   ADDRES
   (4)  036342  160000                                                    .WORD   T$LOLIM
   (4)  036344  177776                                                    .WORD   T$HILIM
10246  036346                           GPRMA   VECTOR,4,0,0,674,YES
   (4)  036346  002031                                                    .WORD   T$CODE
   (4)  036350  036500                                                    .WORD   VECTOR
   (4)  036352  000000                                                    .WORD   T$LOLIM
   (4)  036354  000674                                                    .WORD   T$HILIM
10247  036356                           GPRMD   PRIRTY,6,0,7000,4,7,YES
   (4)  036356  003032                                                    .WORD   T$CODE
   (4)  036360  036531                                                    .WORD   PRIRTY
   (4)  036362  007000                                                    .WORD   7000
   (4)  036364  000004                                                    .WORD   T$LOLIM
   (4)  036366  000007                                                    .WORD   T$HILIM
10248  036370                           GPRMD   SWPAC1,12,0,377,0,056,YES
   (4)  036370  005032                                                    .WORD   T$CODE
   (4)  036372  036562                                                    .WORD   SWPAC1
   (4)  036374  000377                                                    .WORD   377
   (4)  036376  000000                                                    .WORD   T$LOLIM
   (4)  036400  000056                                                    .WORD   T$HILIM
10249  036402                           GPRMD   SWPAC2,14,0,377,0,377,YES
   (4)  036402  006032                                                    .WORD   T$CODE
   (4)  036404  036623                                                    .WORD   SWPAC2
   (4)  036406  000377                                                    .WORD   377
   (4)  036410  000000                                                    .WORD   T$LOLIM
   (4)  036412  000377                                                    .WORD   T$HILIM
10250  036414                           GPRMD   SWPAC3,16,0,377,0,377,YES
   (4)  036414  007032                                                    .WORD   T$CODE
   (4)  036416  036664                                                    .WORD   SWPAC3
   (4)  036420  000377                                                    .WORD   377
   (4)  036422  00C000                                                    .WORD   T$LOLIM
   (4)  036424  000377                                                    .WORD   T$HILIM
10251  036426                           GPRMD   LOOPBK,20,0,7,0,4,YES
   (4)  036426  010032                                                    .WORD   T$CODE
   (4)  036430  036725                                                    .WORD   LOOPBK
   (4)  036432  000007                                                    .WORD   7
   (4)  036434  000000                                                    .WORD   T$LOLIM
```

```
  (4)  036436  000004                                                          .WORD   T$HILIM
10252  036440                          GPRMD   BAUDRT,22,0,7,0,7,YES
  (4)  036440  011032                                                          .WORD   T$CODE
  (4)  036442  037042                                                          .WORD   BAUDRT
  (4)  036444  000007                                                          .WORD   7
  (4)  036446  000000                                                          .WORD   T$LOLIM
  (4)  036450  000007                                                          .WORD   T$HILIM
10253
10254  036452                          ENDHRD
  (2)                                                                          .EVEN
  (3)  036452                                                          L10107:
10255
10256  036452  042504  044526  042503  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
       036460  041440  051123  040440
       036466  042104  042522  051523
       036474  035040  000040
10257  036500  042504  044526  042503  VECTOR: .ASCIZ  /DEVICE VECTOR ADDRESS : /
       036506  053040  041505  047524
       036514  020122  042101  051104
       036522  051505  020123  020072
       036530     000
10258  036531     104  053105  041511  PRIRTY: .ASCIZ  /DEVICE PRIORITY LEVEL : /
       036536  020105  051120  047511
       036544  044522  054524  046040
       036552  053105  046105  035040
       036560  000040
10259  036562  034115  030062  020063  SWPAC1: .ASCIZ  /M8203 SWITCH PACK #1 (REG 11) : /
       036570  053523  052111  044103
       036576  050040  041501  020113
       036604  030443  024040  042522
       036612  020107  030461  020051
       036620  020072     000
10260  036623     115  031070  031460  SWPAC2: .ASCIZ  /M8203 SWITCH PACK #2 (REG 15) : /
       036630  051440  044527  041524
       036636  020110  040520  045503
       036644  021440  020062  051050
       036652  043505  030440  024465
       036660  035040  000040
10261  036664  034115  030062  020063  SWPAC3: .ASCIZ  /M8203 SWITCH PACK #3 (REG 16) : /
       036672  053523  052111  044103
       036700  050040  041501  020113
       036706  031443  024040  042522
       036714  020107  033061  020051
       036722  020072     000
10262  036725     124  051125  040516  LOOPBK: .ASCII  /TURNAROUND TYPE -/<15><12>
       036732  047522  047125  020104
       036740  054524  042520  026440
       036746  005015
10263  036750  030050  044075  031063          .ASCIZ  /(0=H3254&H3255, 1=CABLE, 2=MOD LOC, 3=MOD REM, 4=NONE) : /
       036756  032065  044046  031063
       036764  032465  020054  036461
       036772  040503  046102  026105
       037000  031040  046475  042117
       037006  046040  041517  020054
       037014  036463  047515  020104
       037022  042522  026115  032040
```

```
                037030  047075  047117  024505
                037036  035040  000040
10264           037042  046120  040505  042523  BAUDRT: .ASCII  /PLEASE SELECT BAUD RATE; TYPE '0' FOR 2.4K; '1' FOR 4.8K;/<15><12>
                037050  051440  046105  041505
                037056  020124  040502  042125
                037064  051040  052101  035505
                037072  052040  050131  020105
                037100  030047  020047  047506
                037106  020122  027062  045464
                037114  020073  030447  020047
                037122  047506  020122  027064
                037130  045470  006473     012
10265           037135     047  023462  043040          .ASCII  /'2' FOR 9.6K; '3' FOR 19.2K; '4' FOR 56K; '5' FOR 250K;/<15><12>
                037142  051117  034440  033056
                037150  035513  023440  023463
                037156  043040  051117  030440
                037164  027071  045462  020073
                037172  032047  020047  047506
                037200  020122  033065  035513
                037206  023440  023465  043040
                037214  051117  031040  030065
                037222  035513  005015
10266           037226  033047  020047  047506          .ASCIZ  /'6' FOR 500K; OR '7' FOR 1 MEG BAUD : /
                037234  020122  030065  045460
                037242  020073  051117  023440
                037250  023467  043040  051117
                037256  030440  046440  043505
                037264  041040  052501  020104
                037272  020072     000
10267                   037276                  .EVEN
10268
10269
10270
10271
10272
10273
```

```
10275                                   .SBTTL  SOFTWARE PARAMETER CODING SECTION
10276
10277
10278                                   ;////////////////////////////////////////////////////////////////////
10279                                   ;/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
10280                                   ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
10281                                   ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
10282                                   ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
10283                                   ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
10284                                   ;/ WITH THE OPERATOR.
10285                                   ;////////////////////////////////////////////////////////////////////
10286
10287  037276                           BGNSFT
  (3)  037276  000016                                                    .WORD L10110-L$SOFT/2
  (3)  037300                                                          L$SOFT::
10288
10289  037300                           GPRML   ISMANI,0,1,YES
  (4)  037300  000130                                                    .WORD   T$CODE
  (4)  037302  037334                                                    .WORD   ISMANI
  (4)  037304  000001                                                    .WORD   1
10290  037306                           GPRML   ISPRNT,2,1,YES
  (4)  037306  001130                                                    .WORD   T$CODE
  (4)  037310  037422                                                    .WORD   ISPRNT
  (4)  037312  000001                                                    .WORD   1
10291  037314                           GPRML   ISWPAK,4,1,YES
  (4)  037314  002130                                                    .WORD   T$CODE
  (4)  037316  037505                                                    .WORD   ISWPAK
  (4)  037320  000001                                                    .WORD   1
10292  037322                           GPRMD   TIMCNT,6,0,177777,0,177777,YES
  (4)  037322  003032                                                    .WORD   T$CODE
  (4)  037324  037552                                                    .WORD   TIMCNT
  (4)  037326  177777                                                    .WORD   177777
  (4)  037330  000000                                                    .WORD   T$LOLIM
  (4)  037332  177777                                                    .WORD   T$HILIM
10293
10294  037334                           ENDSFT
  (2)
  (3)  037334                                                            .EVEN
                                                                       L10110:
10295
10296  037334  051511  046440  047101  ISMANI: .ASCIZ  /IS MAN. INTERVEN. DESIRED TO MOUNT TEST CONNECTOR(S) /
       037342  020056  047111  042524
       037350  053122  047105  020056
       037356  042504  044523  042522
       037364  020104  047524  046440
       037372  052517  052116  052040
       037400  051505  020124  047503
       037406  047116  041505  047524
       037414  024122  024523  000040
10297  037422  044123  052517  042114  ISPRNT: .ASCIZ  /SHOULD SWITCH PACK AND AX3-15 PRINTOUT BE ALLOWED /
       037430  051440  044527  041524
       037436  020110  040520  045503
       037444  040440  042116  040440
       037452  031530  030455  020065
       037460  051120  047111  047524
       037466  052125  041040  020105
       037474  046101  047514  042527
```

```
                037502  020104  000
10298           037505  123     047510  046125  ISWPAK: .ASCIZ  /SHOULD SWITCH PACK TESTS BE ALLOWED /
                037512  020104  053523  052111
                037520  044103  050040  041501
                037526  020113  042524  052123
                037534  020123  042502  040440
                037542  046114  053517  042105
                037550  000040
10299           037552  051515  020107  044524  TIMCNT: .ASCIZ  /MSG TIMER VALUE (0-177777), 0 = LONGEST TIME-OUT : /
                037560  042515  020122  040526
                037566  052514  020105  030050
                037574  030455  033467  033467
                037602  024467  020054  020060
                037610  020075  047514  043516
                037616  051505  020124  044524
                037624  042515  047455  052125
                037632  035040  000040
10300                                   .EVEN
10301
10302
10303
10304
10305
10306
10307
10308
10309                                   ;****** PATCH AREA FOR DEBUG ******
10310   037636                          PATCH:
10311           037736                          .=.+100
10312   037736  000240                  NOP
10313   037740  000240                  NOP
10314   037742  000240                  NOP
10315                                   ;************************************
10316
10317
10318
10319   037744                          ENDMOD
10320
10321   037744                          LASTAD
  (2)
  (4)   037744  000000                                          .EVEN
  (4)   037746  000000                                          .WORD   0
  (3)   037750                          L$LAST::                .WORD   0
10322
10323           000001                  .END
```

```
ABORT = 000004          5364#
ADDRES  036452          7371    7377    7392    7412    7430    7451    7469    7488    7507    10245   10256#
ADR   = 000020 G        5323#
ANBITS  002547          5763#
APA   = 000200          5542#
ASBC0 = 000020          5499#
ASBC1 = 000040          5498#
ASBC2 = 000100          5497#
ASSEMB= 000010          5176
AXNUM   002366          5683#   6099    6137    6150    6175    6177    6183*   6189*   6190    6192*   6194    6385    6393*
                        6436*   6600    6606*   6635*   6641*   6918    6920*   6928*   6939*   7063    7077*   7227*   7242
                        7245*   7258*   7262*   8157*   8185*   8188*   8200*   8569*   8860*   8906*   8982*   9035*   9378*
AX0.15= 002306          5610#   6182    7401    7420    7440    7459    7477    7497    7515
AX0.16= 002310          5611#   7401    7420    7440    7459    7477    7497    7515
AX1   = 000001          5395#   5467#
AX1.15= 002312          5612#   7401    7420    7440    7459    7477    7497    7515
AX1.16= 002314          5613#   7401    7420    7440    7459    7477    7497    7515
AX2   = 000002          5394#   5466#
AX2.15= 002316          5614#   7403    7422    7442    7461    7479    7499    7517
AX2.16= 002320          5615#   7403    7422    7442    7461    7479    7499    7517
AX3.15= 002322          5616#   7403    7422    7442    7461    7479    7499    7517
AX3.16= 002324          5617#   7403    7422    7442    7461    7479    7499    7517
AX315U= 000332          5560#
A11     025434          8292#   8322
A12     033730          9429    9646#
BADDAT  002372          5685#   6772*   6773*   6794*   6795*   7380    7395    7434    7491    8106*   8115*   8222*   8499*
                        8523*   8547*   9145*   9185*   9443*   9453*   9464*   9472*
BAUDRT  037042          10252   10264#
BCC   = 000001          5443#   6798    6800    6806
BCCCHK= 100000          5625#   6796    9018    9026    9071    9079    9967    10021
BDRATE  002464          5717#   7097    7154    7636*
BIT0  = 000001 G        5323#   5345    5357    5366    5395    5407    5419    5431    5443    5455    5467    5479    5491
                        5503    5515    5524    5536    5549    5559    5570    5643    6177    6218    6347    6527    6568
                        6608    7606    9024    9077
BIT00 = 000001 G        5323#
BIT01 = 000002 G        5323#
BIT02 = 000004 G        5323#
BIT03 = 000010 G        5323#
BIT04 = 000020 G        5323#
BIT05 = 000040 G        5323#
BIT06 = 000100 G        5323#
BIT07 = 000200 G        5323#
BIT08 = 000400 G        5323#
BIT09 = 001000 G        5323#
BIT1  = 000002 G        5323#   5344    5356    5365    5384    5394    5406    5418    5430    5442    5454    5466    5478
                        5490    5502    5514    5523    5535    5548    5558    5569    5644    6234    6511    6622
BIT10 = 002000 G        5323#   5580    5592
BIT11 = 004000 G        5323#   5579    5591
BIT12 = 010000 G        5323#
BIT13 = 020000 G        5323#
BIT14 = 040000 G        5323#
BIT15 = 100000 G        5323#   5623    5625    5626    5628    6315    6469
BIT2  = 000004 G        5323#   5343    5355    5364    5383    5393    5405    5417    5429    5441    5453    5465    5477
                        5489    5501    5513    5522    5534    5547    5568
BIT3  = 000010 G        5323#   5342    5354    5363    5382    5392    5404    5416    5428    5440    5452    5464    5476
                        5488    5500    5512    5521    5533    5546    5557
```

```
BIT4   = 000020 G   5323# 5341  5353  5381  5391  5403  5415  5427  5439  5451  5463  5475  5487
                    5499  5511  5532  5545  5556
BIT5   = 000040 G   5323# 5352  5373  5380  5390  5402  5414  5426  5438  5450  5462  5474  5486
                    5498  5510  5531  5544  5567
BIT6   = 000100 G   5323# 5340  5351  5372  5379  5389  5401  5413  5425  5437  5449  5461  5473
                    5485  5497  5509  5530  5543  5555  5566
BIT7   = 000200 G   5323# 5339  5350  5362  5371  5378  5400  5412  5424  5436  5448  5460  5472
                    5484  5496  5508  5520  5529  5542  5554  5565  6766  6767
BIT8   = 000400 G   5323# 5582  5594
BIT9   = 001000 G   5323# 5581  5593
BOE    = 000400 G   5323#
BPOLL  = 000100     5372# 9520
BSEL1    002440    5706# 5968* 5970* 5971* 5984* 5985* 5990* 6078* 6080* 6181* 6319* 6320* 6322*
                    6412* 6413* 6418* 6431* 6934* 6936* 7076* 7229* 7621* 7622* 8568* 9115* 9179*
                    9190* 9192* 9242* 9292* 9448* 9482* 9483* 9484* 9682* 9736* 9790* 9844* 9898*
                    9951* 10005*
BSEL4    002442    5707# 6016  6037*
CARR   = 000001     5455# 9182  9638
CHKABT   012136     7242#
CHPCHK= 100000      5623# 8426  8733  8768  8803
CHPTYP   002420     5696# 6316  6426* 6430* 6470  7574* 8190  8196
CKDATA   007266     6757# 7789  7796  7840  7845  7848  7881  7886  7889  7940  7943  7950  7964
                    7967  7992  7995  8031  8034  8064  8067  8312  8315  8390  8393  8442  8445
                    8620  8623  8626  8629  8632  8650  8653  8656  8659  8662  8681  8684  8687
                    8690  8693  8744  8747  8750  8753  8756  8779  8782  8785  8788  8791  8814
                    9750  9804  9858  9912  9966  10020 10056 10060 10065 10143 10148 10151 10154
                    10157 10165 10170 10173 10176 10179 10201 10204 10207 10210 10213 10216 10219
CKLPBK   010766     7062# 9106  9171  9233  9283  9430  9672  9726  9780  9834  9888  9942  9996
CRCCHK= 100000      5626# 6900  7012  8316  8394  8633  8663  8694  8757  8792  8827  9789  9843
                    9897  10149 10152 10171 10174 10211 10214
CRCTY0= 000001      5549#
CRCTY1= 000002      5548#
CRCTY2= 000004      5547#
CRC1   = 000100     5401# 7778  7826  7867  8147  8673  8800  8810  8866  8988  9006  9368  9677
                    9731  9839  9947  10047
CRC2   = 000200     5400# 7778  7826  7867  8147  8912  9041  9059  9368  9677  9731  10001 10047
CS     = 000004     5453# 9632
C$AU   = 000052     5176# 7747
C$AUTO= 000061      5176# 7692
C$BRK  = 000022     5176# 7647
C$BSEG= 000004      5176#
C$BSUB= 000002      5176# 8264  8293  8347  8374  8480  8508  8532  8862  8908  8984  9037  9436
                    9480  9499  9517  9527  9550  9568  9585  9603  9620
C$CEFG= 000045      5176#
C$CLCK= 000062      5176#
C$CLEA= 000012      5176# 7709
C$CLOS= 000035      5176#
C$CLP1= 000006      5176#
C$CVEC= 000036      5176#
C$DCLN= 000044      5176#
C$DODU= 000051      5176# 7689
C$DRPT= 000024      5176#
C$DU   = 000053     5176# 7728
C$EDIT= 000000      5176# 5216
C$ERDF= 000055      5176# 6224  6230  6240  6246  6353  6359  6517  6523  6533  6539  6574  6580
```

```
                    6614    6620    6628    6634    6780    6785    6804    6810    6819    6825    6834    6840    6849
                    6855    7196    7212    7251    7257    7859    7899    7978    8005    8118    8173    8212    8225
                    8502    8526    8550    8891    8897    8937    8943    9126    9148    9188    9203    9253    9303
                    9446    9456    9491    9510    9537    9543    9561    9578    9596    9613    9630    9636    9642
                    9693    9747    9801    9855    9909    9962   10016   10122   10134   10196
C$ERHR= 000056       5176#
C$ERRO= 000060       5176#
C$ERSF= 000054       5176#
C$ERSO= 000057       5176#
C$ESCA= 000010       5176#   8892    8898    8938    8944    9447    9457    9492    9538    9631    9637
C$ESEG= 000005       5176#
C$ESUB= 000003       5176#   8287    8319    8370    8397    8504    8528    8552    8902    8948    9031    9084    9475
                     9494    9512    9522    9545    9563    9580    9598    9615    9644
C$ETST= 000001       5176#   7800    7901    8007    8038    8071    8120    8234    8324    8398    8449    8553    8581
                     8698    8831    8950    9086    9150    9212    9262    9312    9348    9396    9648    9702    9756
                     9810    9864    9918    9972   10026   10069   10223
C$EXIT= 000032       5176#
C$GETB= 000026       5176#
C$GETW= 000027       5176#
C$GMAN= 000043       5176#   7655
C$GPHR= 000042       5176#   7613
C$GPLO= 000030       5176#
C$GPRI= 000040       5176#
C$INIT= 000011       5176#   7660
C$INLP= 000020       5176#
C$MANI= 000050       5176#   7642
C$MEM = 000031       5176#
C$MSG = 000023       5176#   7372    7385    7404    7423    7443    7462    7480    7500    7518
C$OPEN= 000034       5176#
C$PNTB= 000014       5176#   7371    7377    7378    7379    7380    7392    7393    7394    7395    7411    7412    7413
                     7414    7430    7431    7432    7433    7434    7450    7451    7452    7453    7469    7470    7471
                     7487    7488    7489    7490    7491    7507    7508    7509
C$PNTF= 000017       5176#   7105    7110    7119    7646    7651    7727    8491    8493    8517    8541    8577    8579
C$PNTS= 000016       5176#
C$PNTX= 000015       5176#   7381    7382    7383    7384    7396    7397    7398    7399    7400    7401    7402    7403
                     7415    7416    7417    7418    7419    7420    7421    7422    7435    7436    7437    7438    7439
                     7440    7441    7442    7454    7455    7456    7457    7458    7459    7460    7461    7472    7473
                     7474    7475    7476    7477    7478    7479    7492    7493    7494    7495    7496    7497    7498
                     7499    7510    7511    7512    7513    7514    7515    7516    7517
C$QIO = 000377       5176#
C$RDBU= 000007       5176#
C$REFG= 000047       5176#   7586    7589    7592    7595
C$RESE= 000033       5176#   7725
C$REVI= 000003       5176#   5216
C$RFLA= 000021       5176#
C$RPT = 000025       5176#   7535
C$SEFG= 000046       5176#
C$SPRI= 000041       5176#   7682
C$SVEC= 000037       5176#
C$TPRI= 000013       5176#
DDC   = 000100       5543#
DDCMP = 000001       5407#   5479#   7826    7867    8421    8428    8433    8438    8612    8730    8740    8866    8912
                     8988    9006    9041    9059    9111    9176    9238    9288    9335    9368    9469    9471    9731
                     9893    9947   10001   10095
DEVMAP  002426       5699#   7601*   7617*
DEVPTR  002430       5700#   7606*   7615*   7617    7618*
```

```
DFPTBL  002226 G        5261#
DH1     015021          7354#   7379    7414    7471    7490
DH2     015043          7355#   7381    7396    7415    7435    7454    7472    7492    7510
DH3     015072          7356#   7381    7396    7415    7435    7454    7472    7492    7510
DH4     015130          7357#   7383    7398    7417    7437    7456    7474    7494    7512
DH5     015172          6176    7358#
DH6     015175          6179    7359#
DH7     015200          7360#   7400    7419    7439    7458    7476    7496    7514
DH8     015232          7361#   7400    7419    7439    7458    7476    7496    7514
DH9     015271          7362#   7402    7421    7441    7460    7478    7498    7516
DIAGMC= 000000          5176
DISILO  002416          5695#   6102    6139    6153    7573*   8058*   8153*   8232*   9377*   9394*
DISSI = 000040          5390#   5462#   8151    8153    9375    9377
DTR   = 000100          5379#   5449#   7219    9530    9533
EBLK  = 000002          5442#   6813    6815    6821
EF.CON= 000036 G        5323#   7595
EF.NEW= 000035 G        5323#   7592
EF.PWR= 000034 G        5323#
EF.RES= 000037 G        5323#   7589
EF.STA= 000040 G        5323#   7586
EM10    013357          6246    7305#
EM11    013400          6353    7306#   8173
EM12    013415          6359    7307#
EM17    013436          6517    7308#   9126    9203    9253    9303    9693    9747    9801    9855    9909    9962    10016
EM18    013453          6523    7309#
EM19    013474          6533    7310#
EM2     013226          7300#   9456
EM20    013511          6539    7311#
EM21    013532          6574    7312#
EM22    013547          6580    7313#
EM28    013570          6620    7314#
EM29    013611          6614    7315#
EM3     013265          7301#   9148    9446
EM30    013626          6634    7316#
EM31    013647          6628    7257    7317#   8212
EM34    013664          6785    7318#
EM35    013712          6804    7319#
EM36    013732          6810    7320#
EM37    013746          6819    7321#
EM38    013767          6825    7322#
EM39    014004          6834    7323#   7978    8005
EM40    014024          6840    7251    7324#
EM41    014040          6849    7325#   7859    7899
EM42    014061          6855    7326#
EM43    014076          7327#   8502
EM44    014123          7328#   8526
EM45    014150          7329#   8550
EM46    014175          7330#   8118
EM47    014230          7331#   8225
EM48    014263          7332#   8891
EM49    014316          7333#   8897
EM50    014355          7334#   8943
EM51    014411          7335#   8937
EM52    014451          7196    7336#   9596
EM53    014505          7212    7337#
EM54    014541          6780    7338#
```

```
EM55     014563              7339#   9537
EM56     014577              7340#   9491
EM57     014614              7341#   9543
EM58     014631              7342#   9578
EM59     014645              7343#   9613
EM60     014662              7344#   9630   10122   10134
EM61     014676              7345#   9636
EM62     014711              7346#   9642
EM63     014726              7347#   9510
EM64     014743              7348#   9561
EM65     014760              7349#  10196
EM66     015000              7350#   9188
EM7      013304              6224    7302#
EM8      013321              6230    7303#
EM9      013342              6240    7304#
ENAX   = 000004              5393#   5465#   6101    6152
ENDIT    022520              7596    7659#
ENDPAT   002667              5860#   8321
EOM    = 000002              5365#.
ERRFLG   002342              5673#
ERROR1   002410              5692#   6097*   6110*   6135*   6161*   7575*
ERR1     015334 G            7370#
ERR10    021364 G            7506#   8212
ERR2     015366 G            7376#   8118    8502    8526    8550    9446    9456
ERR3     015674 G            7391#   8225    9148
ERR4     016356 G.           6224    6230    6240    6246    6353    6359    6517    6523    6533    6539    6574    6580    6780
                             7196    7212    7410#
ERR5     017034 G            7429#
ERR6     017546 G            6614    6620    6628    6634    7251    7257    7449#
ERR7     020224 G            7468#   7859    7899    7978    8005    8173    8891    8897    8937    8943    9126    9188    9203
                             9253    9303    9491    9510    9537    9543    9561    9578    9596    9613    9630    9636    9642
                             9693    9747    9801    9855    9909    9962   10016   10122   10134   10196
ERR8     020656 G            6785    6804    6810    6819    6825    6834    6840    6849    6855    7486#
EVL    = 000004 G            5323#
E$END  = 002100              5176#
E$LOAD = 000035              5176#   5216
FMT1     012264              7278#   7371    7377    7392    7412    7430    7451    7469    7488    7507
FMT10    012527              7287#   7411    7450    7487
FMT11    012560              7288#   7431
FMT12    012617              7289#   8493
FMT13    012663              7290#   8517
FMT14    012727              7291#   8541
FMT15    012773              7292#   8579
FMT16    022522              7646    7662#
FMT17    022644              7651    7664#
FMT18    013042              7293#   8491    8577
FMT19    013104              7110    7294#
FMT2     012274              7279#   7378    7393    7413    7432    7452    7470    7489    7508
FMT25    013135              7105    7295#
FMT26    013165              7119    7296#
FMT27    023112              7727    7730#
FMT3     012316              7280#   7380    7395    7434    7491
FMT4     012360              7281#   7381    7396    7400    7415    7419    7435    7439    7454    7458    7472    7476    7492
                             7496    7510    7514
FMT5     012373              7282#   7382    7397    7401    7416    7420    7436    7440    7455    7459    7473    7477    7493
                             7497    7511    7515
```

```
FMT6    012423  7283# 7384  7399  7403  7418  7422  7438  7442  7457  7461  7475  7479  7495
                7499  7513  7517
FMT7    012456  7284# 7379  7414  7471  7490
FMT8    012466  7285# 7394  7433  7453  7509
FMT9    012522  7286# 7383  7398  7402  7417  7421  7437  7441  7456  7460  7474  7478  7494
                7498  7512  7516
FRSPAS  002400  5688# 7604* 8485  8512  8536  8571
FRSTIM  002376  5687# 7577  7584*
F$AU  = 000015  5176# 7746  7747
F$AUTO= 000020  5176# 7680  7692
F$BGN = 000040  5176# 5182  7370  7376  7391  7410  7429  7449  7468  7486  7506  7533  7551
                7569  7680  7706  7723  7746  7774  7800  7819  7901  7927  8007  8021  8038
                8053  8071  8086  8120  8138  8234  8260  8264  8287  8293  8319  8324  8343
                8347  8370  8374  8397  8398  8417  8449  8476  8480  8504  8508  8528  8532
                8552  8553  8566  8581  8604  8698  8723  8831  8856  8862  8892  8898  8902
                8908  8938  8944  8948  8950  8978  8984  9031  9037  9084  9086  9102  9150
                9167  9212  9229  9262  9279  9312  9330  9348  9363  9396  9427  9436  9447
                9457  9475  9480  9492  9494  9499  9512  9517  9522  9527  9538  9545  9550
                9563  9568  9580  9585  9598  9603  9615  9620  9631  9637  9644  9648  9668
                9702  9722  9756  9776  9810  9830  9864  9884  9918  9938  9972  9992  10026
                10043 10069 10088 10223 10243 10287 10319
F$CLEA= 000007  5176# 7706  7709
F$DU  = 000016  5176# 7723  7728
F$END = 000041  5176# 5182  7372  7385  7404  7423  7443  7462  7480  7500  7518  7535  7660
                7692  7709  7728  7747  7774  7800  7819  7901  7927  8007  8021  8038  8053
                8071  8086  8120  8138  8234  8260  8264  8287  8293  8319  8324  8343  8347
                8370  8374  8397  8398  8417  8449  8476  8480  8504  8508  8528  8532  8552
                8553  8566  8581  8604  8698  8723  8831  8856  8862  8892  8898  8902  8908
                8938  8944  8948  8950  8978  8984  9031  9037  9084  9086  9102  9150  9167
                9212  9229  9262  9279  9312  9330  9348  9363  9396  9427  9436  9447  9457
                9475  9480  9492  9494  9499  9512  9517  9522  9527  9538  9545  9550  9563
                9568  9580  9585  9598  9603  9615  9620  9631  9637  9644  9648  9668  9702
                9722  9756  9776  9810  9830  9864  9884  9918  9938  9972  9992  10026 10043
                10069 10088 10223 10254 10294 10319
F$HARD= 000004  5176# 10243 10254
F$HW  = 000013  5176# 5261  5275
F$INIT= 000006  5176# 7569  7660
F$JMP = 000050  5176#
F$MOD = 000000  5176# 5182  10319
F$MSG = 000011  5176# 7370  7372  7376  7385  7391  7404  7410  7423  7429  7443  7449  7462
                7468  7480  7486  7500  7506  7518
F$PROT= 000021  5176# 7551  7555
F$PWR = 000017  5176#
F$RPT = 000012  5176# 7533  7535
F$SEG = 000003  5176#
F$SOFT= 000005  5176# 10287 10294
F$SRV = 000010  5176#
F$SUB = 000002  5176# 8264  8287  8293  8319  8347  8370  8374  8397  8480  8504  8508  8528
                8532  8552  8862  8902  8908  8948  8984  9031  9037  9084  9436  9475  9480
                9494  9499  9512  9517  9522  9527  9545  9550  9563  9568  9580  9585  9598
                9603  9615  9620  9644
F$SW  = 000014  5176# 5289  5296
F$TEST= 000001  5176# 7774  7800  7819  7901  7927  8007  8021  8038  8053  8071  8086  8120
                8138  8234  8260  8324  8343  8398  8417  8449  8476  8553  8566  8581  8604
                8698  8723  8831  8856  8950  8978  9086  9102  9150  9167  9212  9229  9262
                9279  9312  9330  9348  9363  9396  9427  9648  9668  9702  9722  9756  9776
```

```
                    9810    9830    9864    9884    9918    9938    9972    9992   10026   10043   10069   10088   10223
GETALL  004200      6174#   6222    6228    6238    6244    6351    6357    6515    6521    6531    6537    6572    6578
                    6612    6618    6626    6632    6778    6783    6802    6808    6817    6823    6832    6838    6847
                    6853    7194    7210    7249    7255    7857    7897    7976    8003    8116    8171    8210    8223
                    8500    8524    8548    8889    8895    8935    8941    9124    9146    9186    9201    9251    9301
                    9444    9454    9489    9508    9535    9541    9559    9576    9594    9611    9628    9634    9640
                    9691    9745    9799    9853    9907    9960   10014   10120   10132   10194
GETPRM  022222       7597    7609#   7616
GETREG  003470       6051#   6180
GOAH    = 000010     5363#
GOODAT  002370       5684#   6770*   6771*   6792*   6793*   7380    7395    7434    7491    8105*   8219*   8220*   8221*
                     8498*   8522*   8546*   9144*   9184*   9442*   9452*   9463*   9471*
G$CNTO= 000200       5176#
G$DISP= 000003       5176#
G$EXCP= 000400       5176#
G$HILI= 000002       5176#
G$LOLI= 000001       5176#
G$NO    = 000000     5176#   7655
G$OFFS= 000400       5176#   7655    10245   10246   10247   10248   10249   10250   10251   10252   10289   10290   10291
                    10292
G$OFSI= 000376       5176#   7655    10245   10246   10247   10248   10249   10250   10251   10252   10289   10290   10291
                    10292
G$PRMA= 000001       5176#   10245   10246
G$PRMD= 000002       5176#   10247   10248   10249   10250   10251   10252   10292
G$PRML= 000000       5176#    7655   10289   10290   10291
G$RADA= 000140       5176#
G$RADB= 000000       5176#
G$RADD= 000040       5176#
G$RADL= 000120       5176#    7655   10289   10290   10291
G$RADO= 000020       5176#   10245   10246   10247   10248   10249   10250   10251   10252   10292
G$XFER= 000004       5176#
G$YES   = 000010     5176#   10245   10246   10247   10248   10249   10250   10251   10252   10289   10290   10291   10292
HDX     = 000020     5381#    5451#   9141    9336    9372    9570    9574
HELP    = 000001     5162#    5208    5218    5241    5941
HOE     = 100000 G   5323#
IACT    = 000100     5437#    6570    6576
IACTIV  006232       6560#    6687    6708    7946    8101    8280    8284    8363    8367    9137    9392
IBE     = 010000 G   5323#
IC      = 000200     5371#    5436#   7852    7971    8099   10186
ICIR    = 000010     5476#    6529    6535    9469    9471
IDL     = 000010     5546#
IDLE    = 000040     5402#    8025
IDU     = 000040 G   5323#
IER     = 020000 G   5323#
IERR    = 000002     5406#    6732    6738
INITRN  005212       6383#    7776    7824    7865    7932    7956    7984    8023    8055    8088    8140    8266    8298
                     8349     8376    8419    8728    8763    8798    8864    8910    8986    9039    9333    9621   10045
                    10093
INTFLG  002340       5671#
INTGRL= 000010       5557#    5560    7067    7082    7099    7113    7162    9107    9173    9235    9285
IRDY    = 000020     5439#    6513    6519    6698    9120    9197    9247    9297    9687    9741    9795    9849    9903
                     9956    10010
ISIRDY  005746       6503#    6689    6702    6706    6710    8103    9345
ISMANI  037334      10289    10296#
ISPRNT  037422      10290    10297#
ISR     = 000100 G   5323#
```

```
ISWPAK  037505          10291   10298#
IXE    = 004000 G        5323#
I$AU   = 000041          5176#   7746#   7747#
I$AUTO= 000041           5176#   7680#   7692#
I$CLN  = 000041          5176#   7706#   7709#
I$DU   = 000041          5176#   7723#   7728#
I$HRD  = 000041         10243#  10254#
I$INIT= 000041           5176#   7569#   7660#
I$MOD  = 000041          5176#   5182#  10319#
I$MSG  = 000041          5176#   7370#   7372#   7376#   7385#   7391#   7404#   7410#   7423#   7429#   7443#   7449#   7462#
                         7468#   7480#   7486#   7500#   7506#   7518#
I$PROT= 000040           5176#   7551#
I$PTAB= 000041           5176#
I$PWR  = 000041          5176#
I$RPT  = 000041          5176#   7533#   7535#
I$SEG  = 000041          5176#   7774    7819    7927    8021    8053    8086    8138    8260    8264    8293    8343    8347
                         8374    8417    8476    8480    8508    8532    8566    8604    8723    8856    8862    8908    8978
                         8984    9037    9102    9167    9229    9279    9330    9363    9427    9436    9480    9499    9517
                         9527    9550    9568    9585    9603    9620    9668    9722    9776    9830    9884    9938    9992
                        10043   10088
I$SETU= 000041           5176#
I$SFT  = 000041         10287#  10294#
I$SRV  = 000041          5176#
I$SUB  = 000041          5176#   7774    7819    7927    8021    8053    8086    8138    8260    8264#   8287#   8293#   8319#
                         8343    8347#   8370#   8374#   8397#   8417    8476    8480#   8504#   8508#   8528#   8532#   8552#
                         8566    8604    8723    8856    8862#   8892    8898    8902#   8908#   8938    8944    8948#   8978
                         8984#   9031#   9037#   9084#   9102    9167    9229    9279    9330    9363    9427    9436#   9447
                         9457    9475#   9480#   9492    9494#   9499#   9512#   9517#   9522#   9527#   9538    9545#   9550#
                         9563#   9568#   9580#   9585#   9598#   9603#   9615#   9620#   9631    9637    9644#   9668    9722
                         9776    9830    9884    9938    9992   10043   10088
I$TST  = 000041          5176#   7774#   7800#   7819#   7901#   7927#   8007#   8021#   8038#   8053#   8071#   8086#   8120#
                         8138#   8234#   8260#   8264    8293    8324#   8343#   8347    8374    8398#   8417#   8449#   8476#
                         8480    8508    8532    8553#   8566#   8581#   8604#   8698#   8723#   8831#   8856#   8862    8908
                         8950#   8978#   8984    9037    9086#   9102#   9150#   9167#   9212#   9229#   9262#   9279#   9312#
                         9330#   9348#   9363#   9396#   9427#   9436    9480    9499    9517    9527    9550    9568    9585
                         9603    9620    9648#   9668#   9702#   9722#   9756#   9776#   9810#   9830#   9864#   9884#   9918#
                         9938#   9972#   9992#  10026#  10043#  10069#  10088#  10223#
I422   = 000200          5554#   5560    7067    7086    7123    7141    7143    7148    7168    9108    9173    9235    9284
J$JMP  = 000167          5176#
LDBYTS  010554           6977#   7779    8867    8913    8992    9045   10096   10102
LDMSG1  010636           7002#   9114    9191    9241    9291
LDTXSI  004646           6290#   6403    6404    6462    6887    6888    6895    6902    6903    6959    6985    7032    7783
                         7784
LOADAT  002374           5686#   7431
LODATA  010156           6883#   9680    9734    9788    9842    9896    9950   10004
LODMSG  010476           6952#   7004    7935    7959    7987    8026    8422    8615    8645    8676    8734    8769    8804
                         9338   10111
LODSIL  010716           7028#   7827    7834    7868    7875    8059    8091    8269    8272    8275    8301    8304    8307
                         8352    8355    8358    8379    8382    8385    8870    8916    8989    8995    9042    9048   10048
                        10051   10099   10105   10108
LOE    = 040000 G        5323#
LOGDEV  002330           5667#   7603*   7610*   7611    7613    7689    7727
LOOPBK  036725          10251   10262#
LOOPIN  003546           6077#
LOT    = 000010 G        5323#
LULOOP= 000010           5342#   5990    6181    6319    6412    6934    6936    7076    7229    8568    9115    9179    9190
```

```
                    9192   9242   9292   9448   9484   9682   9736   9790   9844   9898   9951  10005
LULP  = 000040      5373#  5438#  8155
LUREG   002266      5602   5603   5604   5605   5606   5607   5608   5609   5610   5611   5612   5613   5614
                    5615   5616   5617   5661#
LUR10 = 002266      5602#  6053   7382   7397   7416   7436   7455   7473   7493   7511
LUR11 = 002270      5603#  7382   7397   7416   7436   7455   7473   7493   7511
LUR12 = 002272      5604#  7382   7397   7416   7436   7455   7473   7493   7511
LUR13 = 002274      5605#  7382   7397   7416   7436   7455   7473   7493   7511
LUR14 = 002276      5606#  7384   7399   7418   7438   7457   7475   7495   7513
LUR15 = 002300      5607#  7384   7399   7418   7438   7457   7475   7495   7513
LUR16 = 002302      5608#  7384   7399   7418   7438   7457   7475   7495   7513
LUR17 = 002304      5609#  7384   7399   7418   7438   7457   7475   7495   7513
LUSWI1  002454      5713#  7632*  8496   8498
LUSWI2  002456      5714#  7633*  8520   8522
LUSWI3  002460      5715#  7634*  8544   8546
LU2MOD  002000 G    5182#
L$ACP   002110 G    5216#
L$APT   002036 G    5216#
L$AU    023144 G    5216   7746#
L$AUT   002070 G    5216#
L$AUTO  023000 G    5216   7680#
L$CCP   002106 G    5216#
L$CLEA  023060 G    5216   7706#
L$CO    002032 G    5216#
L$DEPO  002011 G    5216#
L$DESC  003170 G    5216   5934#
L$DESP  002076 G    5216#
L$DEVP  002060 G    5216#
L$DISP  002124 G    5216   5239#
L$DLY   002116 G    5216#
L$DTP   002040 G    5216#
L$DTYP  002034 G    5216#
L$DU    023062 G    5216   7723#
L$DUT   002072 G    5216#
L$DVTY  003162 G    5216   5929#
L$EF    002052 G    5216#
L$ETP   002102 G    5216#
L$EXP1  002042 G    5216#
L$EXP2  002044 G    5216#
L$EXP3  002046 G    5216#
L$EXP4  002064 G    5216#
L$EXP5  002066 G    5216#
L$HARD  036336 G    5216  10243#
L$HIME  002120 G    5216#
L$HPCP  002016 G    5216#
L$HPTP  002022 G    5216#
L$HW    002226 G    5216   5261#
L$ICP   002104 G    5216#
L$INIT  022026 G    5216   7569#
L$LADP  002026 G    5216#
L$LAST  037750 G    5216  10321#
L$LOAD  002100 G    5216#
L$LUN   002074 G    5216#
L$MREV  002050 G    5216#
L$NAME  002000 G    5216#
L$PROT  022020 G    5216   7551#
```

```
L$PRT   002112 G        5216#
L$REPP  002062 G        5216#
L$REV   002010 G        5216#
L$RPT   022016 G        7533#
L$SOFT  037300 G        5216   10287#
L$SPC   002056 G        5216#
L$SPCP  002020 G        5216#
L$SPTP  002024 G        5216#
L$STA   002030 G        5216#
L$SW    002256 G        5216    5289#
L$TEST  002114 G        5216#
L$TIML  002014 G        5216#
L$UNIT  002012 G        5216#   7611
L10000  002254          5261    5275#
L10001  002266          5289    5296#
L10002  015364          7372#
L10003  015672          7385#
L10004  016354          7404#
L10005  017032          7423#
L10006  017544          7443#
L10007  020222          7462#
L10010  020654          7480#
L10011  021362          7500#
L10012  022014          7518#
L10013  022016          7535#
L10015  022520          7660#
L10016  023056          7692#
L10017  023060          7709#
L10020  023110          7728#
L10021  023144          7747#
L10022  023274          7800#
L10023  023650          7901#
L10024  024206          8007#
L10025  024270          8038#
L10026  024360          8071#
L10027  024554          8120#
L10030  025324          8234#
L10031  025564          8324#
L10032  025426          8287#
L10033  025546          8319#
L10034  025770          8398#
L10035  025666          8370#
L10036  025766          8397#
L10037  026124          8449#
L10040  026576          8553#
L10041  026310          8504#
L10042  026442          8528#
L10043  026574          8552#
L10044  026712          8581#
L10045  027250          8698#
L10046  027660          8831#
L10047  030332          8950#
L10050  030102          8892    8898    8902#
L10051  030324          8938    8944    8948#
L10052  031034          9086#
L10053  030570          9031#
```

```
L10054  031026            9084#
L10055  031316            9150#
L10056  031570            9212#
L10057  031752            9262#
L10060  032134            9312#
L10061  032232            9348#
L10062  032442            9396#
L10063  033734            9648#
L10064  032740            9447    9457    9475#
L10065  033032            9492    9494#
L10066  033120            9512#
L10067  033150            9522#
L10070  033260            9538    9545#
L10071  033346            9563#
L10072  033426            9580#
L10073  033514            9598#
L10074  033574            9615#
L10075  033726            9631    9637    9644#
L10076  034124            9702#
L10077  034314            9756#
L10100  034504            9810#
L10101  034674            9864#
L10102  035064            9918#
L10103  035250            9972#
L10104  035434           10026#
L10105  035546           10069#
L10106  036332           10223#
L10107  036452           10243   10254#
L10110  037334           10287   10294#
MAINT1= 000010            5382#   7185    9587
MAINT2= 000004            5383#   7139    7201
MCLK  = 000002            5478#   9468
MCLR  = 000100            5340#   5984    5985
MIFLAG  002256            5291#   7639
MODINT  002422            5697#   7067*   7099*   7128*   7141*   7157*   9674    9728    9782    9836    9890    9944    9998
MODR  = 000010            5452#   9141    9440    9442    9539
MPCSR   002436            5705#   7371    7377    7392    7412    7430    7451    7469    7488    7507    7620*   7651    7685
                          8491    8577
MPIVEC  002446            5710#   7627*
MPOVEC  002450            5711#   7628*   7629*
MPRIOR  002452            5712#   7630*
MSG1    002670            5869#   7005    7007    7010    8616    8646    8677    8735    8770    8805    9339   10112
MSG2    002716            5881#   8423
MSG3    002732            5888#   7936    7960    7988    8027
MSTCLR  003276            5982#   6388    7799    7893    7900    7999    8006    8037    8070    8119    8144    8233    8323
                          8448    8481    8509    8533    8567    8609    8639    8670    8697    8830    8949    9085    9105
                          9170    9232    9282    9347    9365    9395    9437    9481    9500    9518    9528    9551    9569
                          9586    9604    9647    9671    9725    9779    9833    9887    9941    9995   10068   10222
MVIOX = 021000            5637#   6012
MVIXO = 122000            5638#   6034
NEWST   022176            7593    7602#   7612
OACT  = 000100            5425#   6349    6355    6416    8166
OACTIV  005024            6339#   6391    6408    6423    6473    9135
OC    = 000200            5362#   5424#
OCOR  = 000020            5475#   6236    6242    6428
OR    = 000002            5558#   5560    7067    9108    9173    9235    9285
```

```
ORDY  = 000020           5427#  6220  6226
OSIRDY  004334           6210#  6389  6406  6433  6477  6483  9343
OVRR  = 000010           5440#  6843  6845  6851  7855  7895
O$APTS= 000000           5176#  5216
O$AU  = 000001           5176#  5206#  5216
O$BGNR= 000000           5176#  5216
O$BGNS= 000001           5176#  5206#  5216
O$DU  = 000001           5176#  5206#  5216
O$ERRT= 000000           5176#  5216
O$GNSW= 000001           5176#  5206#  5216
O$POIN= 000001           5176#  5206#  5216
O$SETU= 000000           5176#  5216  10321
PATA    002557           5774#  6891  10097  10103  10141  10163
PATB    002603           5797#  6896  10146  10168
PATCH   037636          10310#
PATQ    002613           5809#  8868  8914
PATR    002623           5819#  8993  9015  9046  9068
PATS    002642           5836#  7780  7787  7792
PATT    002662           5854#  8291
PNT   = 001000 G         5323#
POLL  = 000200           5378#  9501
PRI   = 002000 G         5323#
PRIOR   002334           5669#
PRIRTY  036531          10247  10258#
PRI00 = 000000 G         5323#
PRI01 = 000040 G         5323#
PRI02 = 000100 G         5323#
PRI03 = 000140 G         5323#
PRI04 = 000200 G         5323#
PRI05 = 000240 G         5323#
PRI06 = 000300 G         5323#
PRI07 = 000340 G         5323#  7682  7684
PRNFLG  002260           5292#  8487  8514  8538  8573
PSTACK  002332           5668#  6248  6361  6541  6582  6636  6864  7259  7571*
RAB   = 000004           5441#  6828  6830  6836  7974  8001
RABT  = 000004           5501#  7247
RAX15   002354           5678#  6114*  6115*  6185  7113  7123  7130  7143  7148  7151  7162  7165  7168
                         7174  8579
RAX16   002356           5679#  6118*  6119*  6187  6610  6616  6624  6630  7247  7253  8203  8214  8222
RCVBUF  002762           5906#  6890  7008  9117  9132  9194  9209  9244  9259  9294  9309  9681*  9684
                         9699  9735*  9738  9753  9789*  9792  9807  9843*  9846  9861  9897*  9900  9915
                         9953  9969  10007  10023
RCV1ST  006752           6676#  7830  7871  7938  7948  7962  7990  8029  8062  8094  8310  8388  8440
RDALL = 000004           5405#  8378
RDAX  = 000020           5391#  5463#  6101
RDRXSI  006672           6652#  6763  7854  7894  7973  8000  8107
READAX  003576           6095#  6184  6607  7078  7246  8202  8570
READLU  003344           6004#  6055  6105  6113  6117  6156  6217  6233  6346  6415  6427  6510  6526
                         6567  6654  6658  6697  6775  7189  7205  7216  8165  8483  8511  8535  8883
                         8929  9119  9140  9181  9196  9246  9296  9439  9449  9460  9467  9486  9505
                         9532  9556  9573  9591  9608  9625  9686  9740  9794  9848  9902  9955  10009
                        10117  10129  10191
READY = 000200           5460#  6106  6157
REDBYT  002350           5676#  6016*  6017*  6056  6106  6114  6118  6157  6220  6226  6236  6242  6349
                         6355  6416  6428  6513  6519  6529  6535  6570  6576  6655  6659  6698  6776
                         7190  7206  7217  8166  8484*  8493  8496  8499  8517  8520  8523  8541  8544
```

```
                    8547   8887   8893   8933   8939   9120   9141*  9142   9145   9182   9185   9197   9247
                    9297   9440   9443   9450   9453   9461   9464   9468*  9469   9472   9487   9506   9533
                    9539   9557   9574   9592   9609   9626   9632   9638   9687   9741   9795   9849   9903
                    9956   10010  10118  10130  10192
REDDAT  002466      5720#
REGNUM  002364      5682#  6006   6032   6052   6054*  6058*  6059   6061*  6096   6098*  6112*  6116*  6120*
                    6134   6136*  6141*  6145*  6149*  6162*  6210   6216*  6232*  6247*  6253*  6290   6292*
                    6295*  6298*  6339   6345*  6360*  6366*  6384   6399*  6411*  6425*  6437*  6503   6509*
                    6525*  6540*  6546*  6560   6566*  6581*  6587*  6652   6653*  6657*  6660*  6678   6681*
                    6713*  6729   6730*  6741*  6758   6774*  6782*  6788*  6863*  6868*  6919   6924*  6938*
                    7062   7138*  7182*  7187*  7203*  7215*  7228*  7379   7414   7431   7471   7490   7851*
                    7892*  7970*  7998*  8098*  8110*  8114*  8150*  8154*  8162*  8482*  8510*  8534*  8861*
                    8907*  8983*  9036*  9116*  9139*  9180*  9193*  9243*  9293*  9332*  9371*  9374*  9438*
                    9466*  9485*  9502*  9504*  9519*  9529*  9553*  9555*  9571*  9588*  9590*  9606*  9624*
                    9683*  9737*  9791*  9845*  9899*  9952*  10006*  10116*  10185*  10188*
REG0    002476      5730#
REG1    002500      5731#
REG2    002502      5732#  7652*  7655   7656
REG3    002504      5733#
REG4    002506      5734#
REG5    002510      5735#
REG6    002512      5736#
REG7    002514      5737#
REOM  = 000002      5502#  6624   6630   7253   8203   8221
RERR  = 000200      5496#
RETADR  002346      5675#  6249   6362   6542   6583   6637   6865   7111   7260   7775*  7820*  7928*  8022*
                    8054*  8087*  8139*  8265*  8294*  8348*  8375*  8418*  8608*  8727*  8863*  8909*  8985*
                    9038*  9104*  9169*  9231*  9281*  9331*  9364*  9429*  9670*  9724*  9778*  9832*  9886*
                    9940*  9994*  10044*  10089*
RING  = 000200      5448#  9141   9440   9442   9487
ROMI  = 000002      5344#  5968   5970   5971   6078   6080
ROMO  = 000004      5343#  5968   5970   5971   6078   6080
ROR   = 000010      5500#
RRDYTO= 000001      5643#  6097   6110
RSEOM   006420      6600#  6691
RSOM  = 000001      5503#  6610   6616
RTS   = 000040      5450#  9626   10118  10130  10192
RUN   = 000200      5339#  5985   6080   9483
RXABT = 002000      5592#  7944   7968   7996   8068
RXBCC = 000400      5594#  6900   7012   9026   9079   9897   10152  10174  10214
RXEBL = 001000      5593#  7795   7944   7968   7996   8035   8068   8316   8394   9681   9789   9843
RXLEN0= 000001      5570#  6764   8874   8920   8999   9052   9949   10003
RXLEN1= 000002      5569#  6764   8874   8920   8999   9052   9949   10003
RXLEN2= 000004      5568#  6764   8874   8920   8999   9052   9949   10003
RXOVR = 004000      5591#
RXWORD  002414      5694#  6655*  6656*  6659*  6767*  6768   6773   6789   6795   6800   6806   6815   6821
                    6830   6836   6845   6851   7855   7895   7974   8001   8108   8112   8115
RX0   = 000001      5419#  5491#
RX1   = 000002      5418#  5490#
RX2   = 000004      5417#  5489#
RX3   = 000010      5416#  5488#
RX4   = 000020      5415#  5487#
RX5   = 000040      5414#  5486#
RX6   = 000100      5413#  5485#
RX7   = 000200      5412#  5484#
R14NRW  002546      5760#
```

```
SAVE4   002404          5690# 7579* 7582  7690
SAVE6   002406          5691# 7580* 7583  7691
SAVLEN  002424          5698# 5992* 6764  6933* 7576* 9000* 9053*
SCRACH  002326          5666#
SEC     = 000020        5545#
SECA    = 000020        5403# 8268  8300  8351  8378  10047
SELFR   = 000040        5380# 9552
SELSBY= 000002          5384# 9605
SEL4    002442          5708# 7623* 7624*
SEL6    002444          5709# 5969* 6079* 7625* 7626*
SETUP   010304          6918# 8145  8610  8640  8671  9109  9174  9236  9286  9366  9675  9729  9783
                        9837  9891  9945  9999
SFPTBL  002256 G        5289#
SIGQ    = 000100        5473# 9506
SIGR    = 000200        5472# 9557
SOM     = 000001        5366#
STALL   004636          6277# 6321  6323  6414  6419  6432
STARES  002402          5689# 7103  7108  7117  7599* 7605*
STARST  022166          7587  7590  7598#
STBY    = 000002        5454# 9141  9609
STEPLU= 000020          5341# 6320  6322  6413  6418  6431
STEPMP= 000001          5345# 5970  5971
STPCLK  003240          5967# 6014  6038
STPERR  007164          6729# 8427  8432  8437  8739  8774  8809  9005  9058
STPLU   004726          6311# 6479  6693  6736  6860  7785  7832  7837  7873  7878  8096  8163  8176
                        8180  8183  8194  8205  8278  8282  8361  8365  8425  8430  8435  8618  8648
                        8679  8737  8742  8772  8777  8807  8812  8876  8878  8881  8922  8924  8927
                        9002  9008  9012  9055  9061  9065  9341  9382  9387  9390  10054 10114 10124
                        10127 10189
STR     = 000040        5544#
STRIP   = 000010        5404# 7826  7867  8421  8428  8433  8438  8612  8730  8740  8866  8912  8988
                        9006  9041  9059  9111  9176  9238  9288  9335  9731  9893  9947  10001 10095
SUBRPC  002336          5670# 6211  6212  6214* 6215* 6252* 6340  6341  6343* 6344* 6365* 6386* 6387*
                        6439* 6459* 6460* 6486* 6504  6505  6507* 6508* 6545* 6561  6562  6564* 6565*
                        6586* 6601  6602  6604* 6605* 6640* 6679* 6680* 6716* 6759* 6760* 6870* 6940*
                        7065* 7066* 7230* 7243* 7244* 7411  7450  7487  7572*
SVCGBL= 000000          5176# 5182  5190# 5216  5239  5261  5289  5929  5934  7370  7376  7391  7410
                        7429  7449  7468  7486  7506  7533  7551  7569  7680  7706  7723  7746  10243
                        10287 10321#
SVCINS= 000001          5176# 5187# 5216  5239  5261  5289  5929  5934  6224  6230  6240  6246  6353
                        6359  6517  6523  6533  6539  6574  6580  6614  6620  6628  6634  6780  6785
                        6804  6810  6819  6825  6834  6840  6849  6855  7105  7110  7119  7196  7212
                        7251  7257  7371  7372  7377  7378  7379  7380  7381  7382  7383  7384  7385
                        7392  7393  7394  7395  7396  7397  7398  7399  7400  7401  7402  7403  7404
                        7411  7412  7413  7414  7415  7416  7417  7418  7419  7420  7421  7422  7423
                        7430  7431  7432  7433  7434  7435  7436  7437  7438  7439  7440  7441  7442
                        7443  7450  7451  7452  7453  7454  7455  7456  7457  7458  7459  7460  7461
                        7462  7469  7470  7471  7472  7473  7474  7475  7476  7477  7478  7479  7480
                        7487  7488  7489  7490  7491  7492  7493  7494  7495  7496  7497  7498  7499
                        7500  7507  7508  7509  7510  7511  7512  7513  7514  7515  7516  7517  7518
                        7535  7586  7587  7589  7590  7592  7593  7595  7596  7613  7614  7642  7644
                        7646  7647  7651  7655  7660  7682  7689  7692  7709  7725  7727  7728  7747
                        7800  7859  7899  7901  7978  8005  8007  8038  8071  8118  8120  8173  8212
                        8225  8234  8264  8287  8293  8319  8324  8347  8370  8374  8397  8398  8449
                        8480  8491  8493  8502  8504  8508  8517  8526  8528  8532  8541  8550  8552
                        8553  8577  8579  8581  8698  8831  8862  8891  8892  8897  8898  8902  8908
```

```
                    8937    8938    8943    8944    8948    8950    8984    9031    9037    9084    9086    9126    9148
                    9150    9188    9203    9212    9253    9262    9303    9312    9348    9396    9436    9446    9447
                    9456    9457    9475    9480    9491    9492    9494    9499    9510    9512    9517    9522    9527
                    9537    9538    9543    9545    9550    9561    9563    9568    9578    9580    9585    9596    9598
                    9603    9613    9615    9620    9630    9631    9636    9637    9642    9644    9648    9693    9702
                    9747    9756    9801    9810    9855    9864    9909    9918    9962    9972   10016   10026   10069
                   10122   10134   10196   10223   10243   10245   10246   10247   10248   10249   10250   10251   10252
                   10254   10287   10289   10290   10291   10292   10294   10321
SVCSUB= 000001       5176#   5189#   8264    8293    8347    8374    8480    8508    8532    8862    8908    8984    9037
                     9436    9480    9499    9517    9527    9550    9568    9585    9603    9620
SVCTAG= 000001       5176#   5191#   5275    5296    7372    7385    7404    7423    7443    7462    7480    7500    7518
                     7535    7655    7660    7692    7709    7728    7747    7800    7901    8007    8038    8071    8120
                     8234    8287    8319    8324    8370    8397    8398    8449    8504    8528    8552    8553    8581
                     8698    8831    8902    8948    8950    9031    9084    9086    9150    9212    9262    9312    9348
                     9396    9475    9494    9512    9522    9545    9563    9580    9598    9615    9644    9648    9702
                     9756    9810    9864    9918    9972   10026   10069   10223   10254   10294
SVCTST= 000001       5176#   5188#   7774    7819    7927    8021    8053    8086    8138    8260    8343    8417    8476
                     8566    8604    8723    8856    8978    9102    9167    9229    9279    9330    9363    9427    9668
                     9722    9776    9830    9884    9938    9992   10043   10088
SWIFLG  002262       5293#   8494    8518    8542
SWPAC1  036562      10248   10259#
SWPAC2  036623      10249   10260#
SWPAC3  036664      10250   10261#
SW0    = 000002      5430#
SW1    = 000004      5429#
SW2    = 000010      5428#
SW3    = 000040      5426#
SYNCH  = 000226      5537#   7825    7866    8420    8611    8729    9110    9175    9237    9287    9334    9367    9730
                     9892   10094
SYN0   = 000001      5536#
SYN1   = 000002      5535#
SYN2   = 000004      5534#
SYN3   = 000010      5533#
SYN4   = 000020      5532#
SYN5   = 000040      5531#
SYN6   = 000100      5530#
SYN7   = 000200      5529#
S$LSYM= 010000       5176#   5275#   5296#   7372#   7385#   7404#   7423#   7443#   7462#   7480#   7500#   7518#   7535#
                     7655#   7660#   7692#   7709#   7728#   7747#   7800#   7901#   8007#   8038#   8071#   8120#   8234#
                     8287#   8319#   8324#   8370#   8397#   8398#   8449#   8504#   8528#   8552#   8553#   8581#   8698#
                     8831#   8902#   8948#   8950#   9031#   9084#   9086#   9150#   9212#   9262#   9312#   9348#   9396#
                     9475#   9494#   9512#   9522#   9545#   9563#   9580#   9598#   9615#   9644#   9648#   9702#   9756#
                     9810#   9864#   9918#   9972#  10026#  10069#  10223#  10254#  10294#
TCCHEK= 100000       5628#   7068    9431
TCOUNT  002264       5294#   9118    9195    9245    9295    9685    9739    9793    9847    9901    9954   10008
TEOM   = 000002      5523#   8192    8198
TERR   = 000200      5520#
TEST   = 000001      5559#   7067    9108    9173    9235    9285
TESTMD= 000004       5477#   7190    7206    9592
TIMCNT  037552      10292   10299#
TIMFLG  002344       5674#
TMP0    002516       5740#   6176*   6179*   7394    7433    7453    7509
TMP1    002520       5741#   6194*   6195*   7394    7433    7453    7509
TMP2    002522       5742#
TMP3    002524       5743#
TMP4    002526       5744#
```

```
TMP5    002530          5745#
TMP6    002532          5746#
TMP7    002534          5747#
TSOM  = 000001          5524#   8159    9380
TSTCON  002462          5716#   7070    7074    7095    7125    7136    7171    7183    7199    7635*
TSTNUM  002434          5702#   7110    9103*   9168*   9230*   9280*   9428*   9669*   9723*   9777*   9831*   9885*   9939*
                        9993*
TXAB  = 000004          5522#
TXABT = 002000          5580#   5892
TXCHAR  005574          6457#   8731    8766    8801
TXDATA= 000040          5474#   8887    8893    8933    8939    9469    9471
TXEN  = 000100          5389#   5461#   8058    8151    8153    9375    9377
TXEOM = 001000          5581#   5876    5877    5878    5879    5885    5886    6901    7782    8276    8308    8359    8386
                        8871    8917    8996    9049    10052   10100   10106
TXGA  = 000010          5521#
TXGOA = 004000          5579#
TXLEN0= 000040          5567#   6764    8229    8874    8920    8999    9052    9949    10003
TXLEN1= 000100          5566#   6764    8143    8874    8920    8999    9052    9949    10003
TXLEN2= 000200          5565#   6764    8874    8920    8999    9052    9949    10003
TXSOM = 000400          5582#   5869    5870    5881    5882    5893    5894    6395    6886    8990    9043    10109
TXWORD  002412          5693#   6291*   6293    6296    6395*   6396*   6461*   6886*   6892*   6893*   6894    6901*   6958*
                        6983*   6984*   7029*   7782*
TX0   = 000001          5357#   5515#
TX1   = 000002          5356#   5514#
TX2   = 000004          5355#   5513#
TX3   = 000010          5354#   5512#
TX4   = 000020          5353#   5511#
TX5   = 000040          5352#   5510#
TX6   = 000100          5351#   5509#
TX7   = 000200          5350#   5508#
TYPEY   022733          7655    7665#
T$ARGC= 000002          5216#   7105#   7110#   7119#   7371#   7377#   7378#   7379#   7380#   7381#   7382#   7383#   7384#
                        7392#   7393#   7394#   7395#   7396#   7397#   7398#   7399#   7400#   7401#   7402#   7403#   7411#
                        7412#   7413#   7414#   7415#   7416#   7417#   7418#   7419#   7420#   7421#   7422#   7430#   7431#
                        7432#   7433#   7434#   7435#   7436#   7437#   7438#   7439#   7440#   7441#   7442#   7450#   7451#
                        7452#   7453#   7454#   7455#   7456#   7457#   7458#   7459#   7460#   7461#   7469#   7470#   7471#
                        7472#   7473#   7474#   7475#   7476#   7477#   7478#   7479#   7487#   7488#   7489#   7490#   7491#
                        7492#   7493#   7494#   7495#   7496#   7497#   7498#   7499#   7507#   7508#   7509#   7510#   7511#
                        7512#   7513#   7514#   7515#   7516#   7517#   7646#   7651#   7727#   8491#   8493#   8517#   8541#
                        8577#   8579#
T$CODE= 003032          7655#   10245#  10246#  10247#  10248#  10249#  10250#  10251#  10252#  10289#  10290#  10291#  10292#
T$ERRN= 000101          5176#   6224#   6230#   6240#   6246#   6353#   6359#   6517#   6523#   6533#   6539#   6574#   6580#
                        6614#   6620#   6628#   6634#   6780#   6785#   6804#   6810#   6819#   6825#   6834#   6840#   6849#
                        6855#   7196#   7212#   7251#   7257#   7859#   7899#   7978#   8005#   8118#   8173#   8212#   8225#
                        8502#   8526#   8550#   8891#   8897#   8937#   8943#   9126#   9148#   9188#   9203#   9253#   9303#
                        9446#   9456#   9491#   9510#   9537#   9543#   9561#   9578#   9596#   9613#   9630#   9636#   9642#
                        9693#   9747#   9801#   9855#   9909#   9962#   10016#  10122#  10134#  10196#
T$EXCP= 000000          10245#  10246#  10247#  10248#  10249#  10250#  10251#  10252#  10292#
T$FLAG= 000040          8892#   8898#   8938#   8944#   9447#   9457#   9492#   9538#   9631#   9637#
T$GMAN= 000000          5176#
T$HILI= 177777          10245#  10246#  10247#  10248#  10249#  10250#  10251#  10252#  10292#
T$LAST= 000001          5176#   10321#
T$LOLI= 000000          10245#  10246#  10247#  10248#  10249#  10250#  10251#  10252#  10292#
T$LSYM= 010000          5176#   5275    5296    7372    7385    7404    7423    7443    7462    7480    7500    7518    7535
                        7660    7692    7709    7728    7747    7800    7901    8007    8038    8071    8120    8234    8287
                        8319    8324    8370    8397    8398    8449    8504    8528    8552    8553    8581    8698    8831
```

CZDMSA M8203 STATIC TESTS #2 MACY11 30A(1052) 17-JUL-79 15:33 PAGE 10-16
CZDMSA.P11 25-JUN-79 14:01 CROSS REFERENCE TABLE -- USER SYMBOLS SEQ 0190

```
                    8902   8948   8950   9031   9084   9086   9150   9212   9262   9312   9348   9396   9475
                    9494   9512   9522   9545   9563   9580   9598   9615   9644   9648   9702   9756   9810
                    9864   9918   9972  10026  10069  10223  10254  10294
T$LTNO= 000040     10321#
T$NEST= 177777      5176#  5182#  5261#  5275#  5289#  5296#  7370#  7372#  7376#  7385#  7391#  7404#  7410#
                    7423#  7429#  7443#  7449#  7462#  7468#  7480#  7486#  7500#  7506#  7518#  7533#  7535#
                    7551#  7555#  7569#  7660#  7680#  7692#  7706#  7709#  7723#  7728#  7746#  7747#  7774#
                    7800#  7819#  7901#  7927#  8007#  8021#  8038#  8053#  8071#  8086#  8120#  8138#  8234#
                    8260#  8264#  8287#  8293#  8319#  8324#  8343#  8347#  8370#  8374#  8397#  8398#  8417#
                    8449#  8476#  8480#  8504#  8508#  8528#  8532#  8552#  8553#  8566#  8581#  8604#  8698#
                    8723#  8831#  8856#  8862#  8902#  8908#  8948#  8950#  8978#  8984#  9031#  9037#  9084#
                    9086#  9102#  9150#  9167#  9212#  9229#  9262#  9279#  9312#  9330#  9348#  9363#  9396#
                    9427#  9436#  9475#  9480#  9494#  9499#  9512#  9517#  9522#  9527#  9545#  9550#  9563#
                    9568#  9580#  9585#  9598#  9603#  9615#  9620#  9644#  9648#  9668#  9702#  9722#  9756#
                    9776#  9810#  9830#  9864#  9884#  9918#  9938#  9972#  9992#  10026#  10043#  10069#  10088#
                   10223#  10243#  10254#  10287#  10294#  10319#
T$NS0 = 000000      5182#  10319
T$NS1 = 000005      5261#  5275   5289#  5296   7370#  7372   7376#  7385   7391#  7404   7410#  7423   7429#
                    7443   7449#  7462   7468#  7480   7486#  7500   7506#  7518   7533#  7535   7551#  7555
                    7569#  7660   7680#  7692   7706#  7709   7723#  7728   7746#  7747   7774#  7800   7819#
                    7901   7927#  8007   8021#  8038   8053#  8071   8086#  8120   8138#  8234   8260#  8324
                    8343#  8398   8417#  8449   8476#  8553   8566#  8581   8604#  8698   8723#  8831   8856#
                    8950   8978#  9086   9102#  9150   9167#  9212   9229#  9262   9279#  9312   9330#  9348
                    9363#  9396   9427#  9648   9668#  9702   9722#  9756   9776#  9810   9830#  9864   9884#
                    9918   9938#  9972   9992#  10026  10043#  10069  10088#  10223  10243#  10254  10287#  10294
T$NS2 = 000002      8264#  8287   8293#  8319   8347#  8370   8374#  8397   8480#  8504   8508#  8528   8532#
                    8552   8862#  8902   8908#  8948   8984#  9031   9037#  9084   9436#  9475   9480#  9494
                    9499#  9512   9517#  9522   9527#  9545   9550#  9563   9568#  9580   9585#  9598   9603#
                    9615   9620#  9644
T$PTNU= 000000      5176#
T$SAVL= 177777      5176#
T$SEGL= 177777      5176#
T$SUBN= 000000      5176#  7774#  7819#  7927#  8021#  8053#  8086#  8138#  8260#  8264#  8293#  8343#  8347#
                    8374#  8417#  8476#  8480#  8508#  8532#  8566#  8604#  8723#  8856#  8862#  8908#  8978#
                    8984#  9037#  9102#  9167#  9229#  9279#  9330#  9363#  9427#  9436#  9480#  9499#  9517#
                    9527#  9550#  9568#  9585#  9603#  9620#  9668#  9722#  9776#  9830#  9884#  9938#  9992#
                   10043#  10088#
T$TAGL= 177777      5176#
T$TAGN= 010111      5176#  5261#  5289#  7370#  7376#  7391#  7410#  7429#  7449#  7468#  7486#  7506#  7533#
                    7551#  7569#  7680#  7706#  7723#  7746#  7774#  7819#  7927#  8021#  8053#  8086#  8138#
                    8260#  8264#  8293#  8343#  8347#  8374#  8417#  8476#  8480#  8508#  8532#  8566#  8604#
                    8723#  8856#  8862#  8908#  8978#  8984#  9037#  9102#  9167#  9229#  9279#  9330#  9363#
                    9427#  9436#  9480#  9499#  9517#  9527#  9550#  9568#  9585#  9603#  9620#  9668#  9722#
                    9776#  9830#  9884#  9938#  9992#  10043#  10088#  10243#  10287#
T$TEMP= 000000      5239#  5275#  5296#  7372#  7385#  7404#  7423#  7443#  7462#  7480#  7500#  7518#  7535#
                    7555#  7655#  7660#  7692#  7709#  7728#  7747#  7800#  7901#  8007#  8038#  8071#  8120#
                    8234#  8287#  8319#  8324#  8370#  8397#  8398#  8449#  8504#  8528#  8552#  8553#  8581#
                    8698#  8831#  8892#  8898#  8902#  8938#  8944#  8948#  8950#  9031#  9084#  9086#  9150#
                    9212#  9262#  9312#  9348#  9396#  9447#  9457#  9475#  9492#  9494#  9512#  9522#  9538#
                    9545#  9563#  9580#  9598#  9615#  9631#  9637#  9644#  9648#  9702#  9756#  9810#  9864#
                    9918#  9972#  10026#  10069#  10223#  10245#  10246#  10247#  10248#  10249#  10250#  10251#  10252#
                   10254#  10289#  10290#  10291#  10292#  10294#  10319#
T$TEST= 000040      5176#  7774#  7819#  7927#  8021#  8053#  8086#  8138#  8260#  8264   8293   8343#  8347
                    8374   8417#  8476#  8480   8508   8532   8566#  8604#  8723#  8856#  8862   8908   8978#
                    8984   9037   9102#  9167#  9229#  9279#  9330#  9363#  9427#  9436   9480   9499   9517
                    9527   9550   9568   9585   9603   9620   9668#  9722#  9776#  9830#  9884#  9938#  9992#
```

```
                    10043#  10088#  10321
T$TSTM= 177777       5176#   6224    6230    6240    6246    6353    6359    6517    6523    6533    6539    6574    6580
                     6614    6620    6628    6634    6780    6785    6804    6810    6819    6825    6834    6840    6849
                     6855    7105    7110    7119    7196    7212    7251    7257    7371    7372    7377    7378    7379
                     7380    7381    7382    7383    7384    7385    7392    7393    7394    7395    7396    7397    7398
                     7399    7400    7401    7402    7403    7404    7411    7412    7413    7414    7415    7416    7417
                     7418    7419    7420    7421    7422    7423    7430    7431    7432    7433    7434    7435    7436
                     7437    7438    7439    7440    7441    7442    7443    7450    7451    7452    7453    7454    7455
                     7456    7457    7458    7459    7460    7461    7462    7469    7470    7471    7472    7473    7474
                     7475    7476    7477    7478    7479    7480    7487    7488    7489    7490    7491    7492    7493
                     7494    7495    7496    7497    7498    7499    7500    7507    7508    7509    7510    7511    7512
                     7513    7514    7515    7516    7517    7518    7535    7586    7589    7592    7595    7613    7642
                     7646    7647    7651    7655    7660    7682    7689    7692    7709    7725    7727    7728    7747
                     7800    7859    7899    7901    7978    8005    8007    8038    8071    8118    8120    8173    8212
                     8225    8234    8264    8287    8293    8319    8324    8347    8370    8374    8397    8398    8449
                     8480    8491    8493    8502    8504    8508    8517    8526    8528    8532    8541    8550    8552
                     8553    8577    8579    8581    8698    8831    8862    8891    8892    8897    8898    8902    8908
                     8937    8938    8943    8944    8948    8950    8984    9031    9037    9084    9086    9126    9148
                     9150    9188    9203    9212    9253    9262    9303    9312    9348    9396    9436    9446    9447
                     9456    9457    9475    9480    9491    9492    9494    9499    9510    9512    9517    9522    9527
                     9537    9538    9543    9545    9550    9561    9563    9568    9578    9580    9585    9596    9598
                     9603    9613    9615    9620    9630    9631    9636    9637    9642    9644    9648    9693    9702
                     9747    9756    9801    9810    9855    9864    9909    9918    9962    9972   10016   10026   10069
                    10122   10134   10196   10223
T$TSTS= 000001        5176#   7774#   7819#   7927#   8021#   8053#   8086#   8138#   8260#   8343#   8417#   8476#   8566#
                      8604#   8723#   8856#   8978#   9102#   9167#   9229#   9279#   9330#   9363#   9427#   9668#   9722#
                      9776#   9830#   9884#   9938#   9992#  10043#  10088#
T$$AU = 010021        7746#   7747
T$$AUT= 010016        7680#   7692
T$$CLE= 010017        7706#   7709
T$$DU = 010020        7723#   7728
T$$HAR= 010107       10243#  10254
T$$HW = 010000        5261#   5275
T$$INI= 010015        7569#   7660
T$$MSG= 010012        7370#   7372    7376#   7385    7391#   7404    7410#   7423    7429#   7443    7449#   7462    7468#
                      7480    7486#   7500    7506#   7518
T$$PRO= 010014        7551#
T$$RPT= 010013        7533#   7535
T$$SOF= 010110       10287#  10294
T$$SUB= 010075        8264#   8287    8293#   8319    8347#   8370    8374#   8397    8480#   8504    8508#   8528    8532#
                      8552    8862#   8892    8898    8902    8908#   8938    8944    8948    8984#   9031    9037#   9084
                      9436#   9447    9457    9475    9480#   9492    9494    9499#   9512    9517#   9522    9527#   9538
                      9545    9550#   9563    9568#   9580    9585#   9598    9603#   9615    9620#   9631    9637    9644
T$$SW = 010001        5289#   5296
T$$TES= 010106        7774#   78C0    7819#   7901    7927#   8007    8021#   8038    8053#   8071    8086#   8120    8138#
                      8234    8260#   8324    8343#   8398    8417#   8449    8476#   8553    8566#   8581    8604#   8698
                      8723#   8831    8856#   8950    8978#   9086    9102#   9150    9167#   9212    9229#   9262    9279#
                      9312    9330#   9348    9363#   9396    9427#   9648    9668#   9702    9722#   9756    9776#   9810
                      9830#   9864    9884#   9918    9938#   9972    9992#  10026  10043#  10069  10088#  10223
T1      023146 G      5239    7774#
T10     025772 G      5239    8417#
T11     026126 G      5239    8476#
T11.1   026126        8480#
T11.2   026312        8508#
T11.3   026444        8532#
T12     026600 G      5239    8566#
```

```
T13      026714 G          5239    8604#
T14      027252 G          5239    8723#
T15      027662 G          5239    8856#
T15.1    027676            8862#
T15.2    030120            8908#
T16      030334 G          5239    8978#
T16.1    030350            8984#
T16.2    030606            9037#
T17      031036 G          5239    9102#
T18      031320 G          5239    9167#
T19      031572 G          5239    9229#
T2       023276 G          5239    7819#
T20      031754 G          5239    9279#
T21      032136 G          5239    9330#
T22      032234 G          5239    9363#
T23      032444 G          5239    9427#
T23.1    032466            9436#
T23.10   033576            9620#
T23.2    032742            9480#
T23.3    033034            9499#
T23.4    033122            9517#
T23.5    033152            9527#
T23.6    033262            9550#
T23.7    033350            9568#
T23.8    033430            9585#
T23.9    033516            9603#
T24      033736 G          5239    9668#
T25      034126 G          5239    9722#
T26      034316 G          5239    9776#
T27      034506 G          5239    9830#
T28      034676 G          5239    9884#
T29      035066 G          5239    9938#
T3       023652 G          5239    7927#
T30      035252 G          5239    9992#
T31      035436 G          5239   10043#
T32      035550 G          5239   10088#
T4       024210 G          5239    8021#
T5       024272 G          5239    8053#
T6       024362 G          5239    8086#
T7       024556 G          5239    8138#
T8       025326 G          5239    8260#
T8.1     025326            8264#
T8.2     025434            8293#
T9       025566 G          5239    8343#
T9.1     025566            8347#
T9.2     025670            8374#
UAM   =  000200 G          5323#
UNIT     002432            5701#
UNRR  =  000001            5431#   6776
UPBITS   002536            5750#
VECTOR   036500           10246   10257#
V35   =  000020            5556#   5560    7067    7084    7128    7130    7165    9108    9172    9235    9285
WAIT50   004620            6263#   6405    6463    6695    6962    6988    7035    9134
WAX   =  000010            5392#   5464#   6152
WAX15    002360            5680#   6142*   6143    6394*   6396    6921*   6930*   8158*   8873*   8919*   8998*   9051*   9379*
                           9384*
```

```
WAX16   002362          5681#  6146*  6147   6397*  6922*  6932*  6933   8159*  8178*  8186*  8189*  8192*  8198*
                        8874*  8920*  8999*  9000   9052*  9053   9380*  9385*
WRDYTO= 000002          5644#  6135   6161
WRIBYT  002352          5677#  6036*  6037   6099*  6100*  6101*  6102*  6137*  6138*  6139*  6143*  6147*  6150*
                        6151*  6152*  6153*  6293*  6296*  6401*  6731*  6732*  6738*  6926*  7139*  7185*  7201*
                        7217*  7218*  7219*  7852*  7971*  8099*  8151*  8155*  9336*  9372*  9375*  9458*  9501*
                        9520*  9530*  9552*  9570*  9587*  9605*  10186*
WRITAX  003764          6133#  6398   6923   6935   8160   8175   8179   8182   8187   8193   8199   8875   8921
                        9001   9054   9381   9386   9389
WRITLU  003422          6030#  6103   6140   6144   6148   6154   6294   6297   6402   6733   6739   6927   7140
                        7186   7202   7220   7853   7972   8100   8152   8156   9337   9373   9376   9459   9503
                        9521   9531   9554   9572   9589   9607   10187
XYZ     = 000100        5555#  5560   7067   7089   7151   7157   7174   9108   9173   9234   9285
X$ALWA= 000000          5176#
X$FALS= 000040          5176#
X$OFFS= 000400          5176#
X$TRUE= 000020          5176#
$LSTIN= 000001          5185#
$LSTTA= 000001          5186#
.       = 037750        5168#  5661#  5861#  5906#  5934#  7666#  7731#  8892   8898   8938   8944   9447   9457
                        9492   9538   9631   9637   10267#  10311#
```

```
BCOMPL    102#  5176#  7587  7590  7593  7596  7614  7644
BERROR    110#  5176#
BGNAU     118#  5176#  7746
BGNAUT    132#  5176#  7680
BGNCLN    146#  5176#  7706
BGNDU     159#  5176#  7723
BGNHRD    172#  5176#  10243
BGNHW     187#  5176#  5261
BGNINI    203#  5176#  7569
BGNMOD    215#  5176#  5182
BGNMSG    232#  5176#  7370  7376  7391  7410  7429  7449  7468  7486  7506
BGNPRO    245#  5176#  7551
BGNPTA    259#  5176#
BGNRPT    297#  5176#  7533
BGNSEG    311#  5176#
BGNSET    325#  5176#
BGNSFT    352#  5176#  10287
BGNSRV    367#  5176#
BGNSUB    380#  5176#  8264  8293  8347  8374  8480  8508  8532  8862  8908  8984  9037  9436  9480
          9499  9517   9527  9550  9568  9585  9603  9620
BGNSW     408#  5176#  5289
BGNTST    423#  5176#  7774  7819  7927  8021  8053  8086  8138  8260  8343  8417  8476  8566  8604
          8723  8856   8978  9102  9167  9229  9279  9330  9363  9427  9668  9722  9776  9830  9884
          9938  9992   10043 10088
BNCOMP    457#  5176#
BNERRO    465#  5176#
BREAK     473#  5176#  7647
BRESET    481#  5176#  7725
CKLOOP    491#  5176#
CLOCK     503#  5176#
CLOSE     514#  5176#
CLRVEC    528#  5176#
COMMEN    537#  5176#
DELAY     568#  5176#
DESCRI    558#  5176#  5934
DEVTYP    588#  5176#  5929
DISPAT    597#  5176#  5239
DISPLA    615#  5176#
DOCLN     635#  5176#
DODU      643#  5176#  7689
DORPT     651#  5176#
ENDAU     659#  5176#  7747
ENDAUT    675#  5176#  7692
ENDCLN    692#  5176#  7709
ENDCOM    708#  5176#
ENDDU     730#  5176#  7728
ENDHRD    749#  5176#  10254
ENDHW     765#  5176#  5275
ENDINI    779#  5176#  7660
ENDMOD    796#  5176#  10319
ENDMSG    813#  5176#  7372  7385  7404  7423  7443  7462  7480  7500  7518
ENDPRO    830#  5176#  7555
ENDPTA    844#  5176#
ENDRPT    857#  5176#  7535
ENDSEG    876#  5176#
ENDSET    895#  5176#
```

```
ENDSFT   915#  5176#  10294
ENDSRV   932#  5176#
ENDSUB   952#  5176#  8287  8319  8370  8397  8504  8528  8552  8902  8948  9031  9084  9475  9494
        9512   9522   9545  9563  9580  9598  9615  9644
ENDSW    974#  5176#  5296
ENDTST   988#  5176#  7800  7901  8007  8038  8071  8120  8234  8324  8398  8449  8553  8581  8698
        8831   8950   9086  9150  9212  9262  9312  9348  9396  9648  9702  9756  9810  9864  9918
        9972  10026  10069 10223
EQUALS  1009#  5176#  5323
ERRDF   1087#  5176#  6224  6230  6240  6246  6353  6359  6517  6523  6533  6539  6574  6580  6614
        6620   6628   6634  6780  6785  6804  6810  6819  6825  6834  6840  6849  6855  7196  7212
        7251   7257   7859  7899  7978  8005  8118  8173  8212  8225  8502  8526  8550  8891  8897
        8937   8943   9126  9148  9188  9203  9253  9303  9446  9456  9491  9510  9537  9543  9561
        9578   9596   9613  9630  9636  9642  9693  9747  9801  9855  9909  9962 10016 10122 10134
       10196
ERRHRD  1099#  5176#
ERROR   1109#  5176#
ERRSF   1118#  5176#
ERRSOF  1130#  5176#
ERRTBL  1140#  5176#
ESCAPE  1156#  5176#  8892  8898  8938  8944  9447  9457  9492  9538  9631  9637
EXIT    1186#  5176#
FEQUAL  1228#  5176#
GETBYT  1246#  5176#
GETPRI  1264#  5176#
GETWOR  1256#  5176#
GMANIA  1286#  5176#
GMANID  1299#  5176#
GMANIL  1315#  5176#  7655
GPHARD  1328#  5176#  7613
GPRMA   1340#  5176#  10245 10246
GPRMD   1372#  5176#  10247 10248 10249 10250 10251 10252 10292
GPRML   1407#  5176#  7655# 10289 10290 10291
HEADER  1432#  5176#  5216
INLOOP  1446#  5176#
IOSETU  1453#  5176#
IOSTAR  1466#  5176#
KT11    1488#  5176#
LASTAD  1659#  5176#  10321
MANUAL  1677#  5176#  7642
MEMORY  1685#  5176#
M$BYTE  2901#  5176#  5216#
M$CHEC  3206#  5176#
M$CNTO  3279#  5176#  7655# 10245# 10246# 10247# 10248# 10249# 10250# 10251# 10252# 10289# 10290# 10291# 10292#
M$COUN  3124#  5176#  7105#  7110#  7119#  7371#  7377#  7378#  7379#  7380#  7381#  7382#  7383#  7384#  7392#
        7393#  7394#  7395#  7396#  7397#  7398#  7399#  7400#  7401#  7402#  7403#  7411#  7412#  7413#  7414#
        7415#  7416#  7417#  7418#  7419#  7420#  7421#  7422#  7430#  7431#  7432#  7433#  7434#  7435#  7436#
        7437#  7438#  7439#  7440#  7441#  7442#  7450#  7451#  7452#  7453#  7454#  7455#  7456#  7457#  7458#
        7459#  7460#  7461#  7469#  7470#  7471#  7472#  7473#  7474#  7475#  7476#  7477#  7478#  7479#  7487#
        7488#  7489#  7490#  7491#  7492#  7493#  7494#  7495#  7496#  7497#  7498#  7499#  7507#  7508#  7509#
        7510#  7511#  7512#  7513#  7514#  7515#  7516#  7517#  7646#  7651#  7727#  8491#  8493#  8517#  8541#
        8577#  8579#
M$DATA  2614#  5176#  5216#  5929#  5934#
M$DECR  3063#  5176#  5275#  5296#  7372#  7385#  7404#  7423#  7443#  7462#  7480#  7500#  7518#  7535#  7555#
        7660#  7692#  7709#  7728#  7747#  7800#  7901#  8007#  8038#  8071#  8120#  8234#  8287#  8319#  8324#
        8370#  8397#  8398#  8449#  8504#  8528#  8552#  8553#  8581#  8698#  8831#  8902#  8948#  8950#  9031#
```

```
          9084#  9086#  9150#  9212#  9262#  9312#  9348#  9396#  9475#  9494#  9512#  9522#  9545#  9563#  9580#
          9598#  9615#  9644#  9648#  9702#  9756#  9810#  9864#  9918#  9972#  10026#  10069#  10223#  10254#  10294#
         10319#
M$DEFA    3263#  5176#  7655#  10245#  10246#  10247#  10248#  10249#  10250#  10251#  10252#  10289#  10290#  10291#  10292#
M$ENDE    3145#  5176#  5275#  5296#  7372#  7385#  7404#  7423#  7443#  7462#  7480#  7500#  7518#  7535#  7660#
          7692#  7709#  7728#  7747#  7800#  7901#  8007#  8038#  8071#  8120#  8234#  8287#  8319#  8324#  8370#
          8397#  8398#  8449#  8504#  8528#  8552#  8553#  8581#  8698#  8831#  8902#  8948#  8950#  9031#  9084#
          9086#  9150#  9212#  9262#  9312#  9348#  9396#  9475#  9494#  9512#  9522#  9545#  9563#  9580#  9598#
          9615#  9644#  9648#  9702#  9756#  9810#  9864#  9918#  9972#  10026#  10069#  10223#  10254#  10294#  10319#
M$ERRI    2365#  5176#  6224#  6230#  6240#  6246#  6353#  6359#  6517#  6523#  6533#  6539#  6574#  6580#  6614#
          6620#  6628#  6634#  6780#  6785#  6804#  6810#  6819#  6825#  6834#  6840#  6849#  6855#  7196#  7212#
          7251#  7257#  7859#  7899#  7978#  8005#  8118#  8173#  8212#  8225#  8502#  8526#  8550#  8891#  8897#
          8937#  8943#  9126#  9148#  9188#  9203#  9253#  9303#  9446#  9456#  9491#  9510#  9537#  9543#  9561#
          9578#  9596#  9613#  9630#  9636#  9642#  9693#  9747#  9801#  9855#  9909#  9962#  10016#  10122#  10134#
         10196#
M$ESCA    2921#  5176#  8892#  8898#  8938#  8944#  9447#  9457#  9492#  9538#  9631#  9637#
M$ESCS    2932#  5176#  8892#  8898#  8938#  8944#  9447#  9457#  9492#  9538#  9631#  9637#
M$EXCP    3186#  5176#  10245#  10246#  10247#  10248#  10249#  10250#  10251#  10252#  10292#
M$EXIT    2943#  5176#
M$EXSE    2965#  5176#
M$EXTJ    2954#  5176#
M$GEN     3087#  5176#  5182#  5216#  5239#  5261#  5275#  5289#  5296#  5929#  5934#  7370#  7372#  7376#  7385#
          7391#  7404#  7410#  7423#  7429#  7443#  7449#  7462#  7468#  7480#  7486#  7500#  7506#  7518#  7533#
          7535#  7551#  7569#  7655#  7660#  7680#  7692#  7706#  7709#  7723#  7728#  7746#  7747#  7774#  7800#
          7819#  7901#  7927#  8007#  8021#  8038#  8053#  8071#  8086#  8120#  8138#  8234#  8260#  8264#  8287#
          8293#  8319#  8324#  8343#  8347#  8370#  8374#  8397#  8398#  8417#  8449#  8476#  8480#  8504#  8508#
          8528#  8532#  8552#  8553#  8566#  8581#  8604#  8698#  8723#  8831#  8856#  8862#  8902#  8908#  8948#
          8950#  8978#  8984#  9031#  9037#  9084#  9086#  9102#  9150#  9167#  9212#  9229#  9262#  9279#  9312#
          9330#  9348#  9363#  9396#  9427#  9436#  9475#  9480#  9494#  9499#  9512#  9517#  9522#  9527#  9545#
          9550#  9563#  9568#  9580#  9585#  9598#  9603#  9615#  9620#  9644#  9648#  9668#  9702#  9722#  9756#
          9776#  9810#  9830#  9864#  9884#  9918#  9938#  9972#  9992#  10026#  10043#  10069#  10088#  10223#  10243#
         10254#  10287#  10294#  10321#
M$GENB    2764#  5176#  7655#
M$GETS    3079#  5176#  5275#  5296#  7372#  7385#  7404#  7423#  7443#  7462#  7480#  7500#  7518#  7535#  7555#
          7660#  7692#  7709#  7728#  7747#  7800#  7901#  8007#  8038#  8071#  8120#  8234#  8287#  8319#  8324#
          8370#  8397#  8398#  8449#  8504#  8528#  8552#  8553#  8581#  8698#  8831#  8902#  8948#  8950#  9031#
          9084#  9086#  9150#  9212#  9262#  9312#  9348#  9396#  9475#  9494#  9512#  9522#  9545#  9563#  9580#
          9598#  9615#  9644#  9648#  9702#  9756#  9810#  9864#  9918#  9972#  10026#  10069#  10223#  10254#  10294#
         10319#
M$GETT    2634#  5176#  8892#  8898#  8938#  8944#  9447#  9457#  9492#  9538#  9631#  9637#
M$GNGB    2689#  5176#  5182#  5216#  5239#  5261#  5289#  5929#  5934#  7370#  7376#  7391#  7410#  7429#  7449#
          7468#  7486#  7506#  7533#  7551#  7569#  7680#  7706#  7723#  7746#  10243#  10287#  10321#
M$GNIN    3101#  5176#  5216#  5239#  5261#  5289#  5929#  5934#  6224#  6230#  6240#  6246#  6353#  6359#  6517#
          6523#  6533#  6539#  6574#  6580#  6614#  6620#  6628#  6634#  6780#  6785#  6804#  6810#  6819#  6825#
          6834#  6840#  6849#  6855#  7105#  7110#  7119#  7196#  7212#  7251#  7257#  7371#  7372#  7377#  7378#
          7379#  7380#  7381#  7382#  7383#  7384#  7385#  7392#  7393#  7394#  7395#  7396#  7397#  7398#  7399#
          7400#  7401#  7402#  7403#  7404#  7411#  7412#  7413#  7414#  7415#  7416#  7417#  7418#  7419#  7420#
          7421#  7422#  7423#  7430#  7431#  7432#  7433#  7434#  7435#  7436#  7437#  7438#  7439#  7440#  7441#
          7442#  7443#  7450#  7451#  7452#  7453#  7454#  7455#  7456#  7457#  7458#  7459#  7460#  7461#  7462#
          7469#  7470#  7471#  7472#  7473#  7474#  7475#  7476#  7477#  7478#  7479#  7480#  7487#  7488#  7489#
          7490#  7491#  7492#  7493#  7494#  7495#  7496#  7497#  7498#  7499#  7500#  7507#  7508#  7509#  7510#
          7511#  7512#  7513#  7514#  7515#  7516#  7517#  7518#  7535#  7586#  7587#  7589#  7590#  7592#  7593#
          7595#  7596#  7613#  7614#  7642#  7644#  7646#  7647#  7651#  7655#  7660#  7682#  7689#  7692#  7709#
          7725#  7727#  7728#  7747#  7800#  7859#  7899#  7901#  7978#  8005#  8007#  8038#  8071#  8118#  8120#
          8173#  8212#  8225#  8234#  8264#  8287#  8293#  8319#  8324#  8347#  8370#  8374#  8397#  8398#  8449#
          8480#  8491#  8493#  8502#  8504#  8508#  8517#  8526#  8528#  8532#  8541#  8550#  8552#  8553#  8577#
```

M$PRIN 2349# 5176# 7105# 7110# 7119# 7371# 7377# 7378# 7379# 7380# 7381# 7382# 7383# 7384# 7392#
7393# 7394# 7395# 7396# 7397# 7398# 7399# 7400# 7401# 7402# 7403# 7411# 7412# 7413# 7414#
7415# 7416# 7417# 7418# 7419# 7420# 7421# 7422# 7430# 7431# 7432# 7433# 7434# 7435# 7436#
7437# 7438# 7439# 7440# 7441# 7442# 7450# 7451# 7452# 7453# 7454# 7455# 7456# 7457# 7458#
7459# 7460# 7461# 7469# 7470# 7471# 7472# 7473# 7474# 7475# 7476# 7477# 7478# 7479# 7487#
7488# 7489# 7490# 7491# 7492# 7493# 7494# 7495# 7496# 7497# 7498# 7499# 7507# 7508# 7509#
7510# 7511# 7512# 7513# 7514# 7515# 7516# 7517# 7646# 7651# 7727# 8491# 8493# 8517# 8541#
8577# 8579#

M$PUSH 2337# 5176# 5182# 5261# 5289# 7370# 7376# 7391# 7410# 7429# 7449# 7468# 7486# 7506# 7533#
7551# 7569# 7680# 7706# 7723# 7746# 7774# 7819# 7927# 8021# 8053# 8086# 8138# 8260# 8264#
8293# 8343# 8347# 8374# 8417# 8476# 8480# 8508# 8532# 8566# 8604# 8723# 8856# 8862# 8908#
8978# 8984# 9037# 9102# 9167# 9229# 9279# 9330# 9363# 9427# 9436# 9480# 9499# 9517# 9527#
9550# 9568# 9585# 9603# 9620# 9668# 9722# 9776# 9830# 9884# 9938# 9992# 10043# 10088# 10243#
10287#

M$PUT 2819# 5176# 7105# 7110# 7119# 7371# 7377# 7378# 7379# 7380# 7381# 7382# 7383# 7384# 7392#
7393# 7394# 7395# 7396# 7397# 7398# 7399# 7400# 7401# 7402# 7403# 7411# 7412# 7413# 7414#
7415# 7416# 7417# 7418# 7419# 7420# 7421# 7422# 7430# 7431# 7432# 7433# 7434# 7435# 7436#
7437# 7438# 7439# 7440# 7441# 7442# 7450# 7451# 7452# 7453# 7454# 7455# 7456# 7457# 7458#
7459# 7460# 7461# 7469# 7470# 7471# 7472# 7473# 7474# 7475# 7476# 7477# 7478# 7479# 7487#
7488# 7489# 7490# 7491# 7492# 7493# 7494# 7495# 7496# 7497# 7498# 7499# 7507# 7508# 7509#
7510# 7511# 7512# 7513# 7514# 7515# 7516# 7517# 7646# 7651# 7727# 8491# 8493# 8517# 8541#
8577# 8579#

M$PUT1 2842# 5176# 7105# 7110# 7119# 7371# 7377# 7378# 7379# 7380# 7381# 7382# 7383# 7384# 7392#
7393# 7394# 7395# 7396# 7397# 7398# 7399# 7400# 7401# 7402# 7403# 7411# 7412# 7413# 7414#
7415# 7416# 7417# 7418# 7419# 7420# 7421# 7422# 7430# 7431# 7432# 7433# 7434# 7435# 7436#
7437# 7438# 7439# 7440# 7441# 7442# 7450# 7451# 7452# 7453# 7454# 7455# 7456# 7457# 7458#
7459# 7460# 7461# 7469# 7470# 7471# 7472# 7473# 7474# 7475# 7476# 7477# 7478# 7479# 7487#
7488# 7489# 7490# 7491# 7492# 7493# 7494# 7495# 7496# 7497# 7498# 7499# 7507# 7508# 7509#
7510# 7511# 7512# 7513# 7514# 7515# 7516# 7517# 7646# 7651# 7727# 8491# 8493# 8517# 8541#
8577# 8579#

M$RADI 3151# 5176# 7655# 10245# 10246# 10247# 10248# 10249# 10250# 10251# 10252# 10289# 10290# 10291# 10292#

M$RBRO 2787# 5176#

M$RNRO 2802# 5176# 7613#

M$SETS 3071# 5176# 5182# 5261# 5289# 7370# 7376# 7391# 7410# 7429# 7449# 7468# 7486# 7506# 7533#
7551# 7569# 7680# 7706# 7723# 7746# 7774# 7819# 7927# 8021# 8053# 8086# 8138# 8260# 8264#
8293# 8343# 8347# 8374# 8417# 8476# 8480# 8508# 8532# 8566# 8604# 8723# 8856# 8862# 8908#
8978# 8984# 9037# 9102# 9167# 9229# 9279# 9330# 9363# 9427# 9436# 9480# 9499# 9517# 9527#
9550# 9568# 9585# 9603# 9620# 9668# 9722# 9776# 9830# 9884# 9938# 9992# 10043# 10088# 10243#
10287#

M$STAR 2468# 5176#

M$SVC 2746# 5176# 6224 6230 6240 6246 6353 6359 6517 6523 6533 6539 6574 6580 6614
6620 6628 6634 6780 6785 6804 6810 6819 6825 6834 6840 6849 6855 7105# 7110#
7119# 7196 7212 7251 7257 7371# 7372# 7377# 7378# 7379# 7380# 7381# 7382# 7383# 7384#
7385# 7392# 7393# 7394# 7395# 7396# 7397# 7398# 7399# 7400# 7401# 7402# 7403# 7404# 7411#
7412# 7413# 7414# 7415# 7416# 7417# 7418# 7419# 7420# 7421# 7422# 7423# 7430# 7431# 7432#
7433# 7434# 7435# 7436# 7437# 7438# 7439# 7440# 7441# 7442# 7443# 7450# 7451# 7452# 7453#
7454# 7455# 7456# 7457# 7458# 7459# 7460# 7461# 7462# 7469# 7470# 7471# 7472# 7473# 7474#
7475# 7476# 7477# 7478# 7479# 7480# 7487# 7488# 7489# 7490# 7491# 7492# 7493# 7494# 7495#
7496# 7497# 7498# 7499# 7500# 7507# 7508# 7509# 7510# 7511# 7512# 7513# 7514# 7515# 7516#
7517# 7518# 7535# 7586# 7589# 7592# 7595# 7613# 7642# 7646# 7647# 7651# 7655# 7660# 7682#
7689# 7692# 7709# 7725# 7727# 7728# 7747# 7800# 7859 7899 7901# 7978 8005 8007# 8038#
8071# 8118 8120# 8173 8212 8225 8234# 8264# 8287# 8293# 8319# 8324# 8347# 8370# 8374#
8397# 8398# 8449# 8480# 8491# 8493# 8502 8504# 8508# 8517# 8526 8528# 8532# 8541# 8550
8552# 8553# 8577# 8579# 8581# 8698# 8831# 8862# 8891 8892# 8897 8898# 8902# 8908# 8937
8938# 8943 8944# 8948# 8950# 8984# 9031# 9037# 9084# 9086# 9126 9148 9150# 9188 9203
9212# 9253 9262# 9303 9312# 9348# 9396# 9436# 9446 9447# 9456 9457# 9475# 9480# 9491

```
          9492#  9494#  9499#  9510   9512#  9517#  9522#  9527#  9537   9538#  9543   9545#  9550#  9561   9563#
          9568#  9578   9580#  9585#  9596   9598#  9603#  9613   9615#  9620#  9630   9631#  9636   9637#  9642
          9644#  9648#  9693   9702#  9747   9756#  9801   9810#  9855   9864#  9909   9918#  9962   9972#  10016
         10026#  10069#  10122  10134  10196  10223#
M$TLAB    2739#  5176#  6224#  6230#  6240#  6246#  6353#  6359#  6517#  6523#  6533#  6539#  6574#  6580#  6614#
          6620#  6628#  6634#  6780#  6785#  6804#  6810#  6819#  6825#  6834#  6840#  6849#  6855#  7105#  7110#
          7119#  7196#  7212#  7251#  7257#  7371#  7372#  7377#  7378#  7379#  7380#  7381#  7382#  7383#  7384#
          7385#  7392#  7393#  7394#  7395#  7396#  7397#  7398#  7399#  7400#  7401#  7402#  7403#  7404#  7411#
          7412#  7413#  7414#  7415#  7416#  7417#  7418#  7419#  7420#  7421#  7422#  7423#  7430#  7431#  7432#
          7433#  7434#  7435#  7436#  7437#  7438#  7439#  7440#  7441#  7442#  7443#  7450#  7451#  7452#  7453#
          7454#  7455#  7456#  7457#  7458#  7459#  7460#  7461#  7462#  7469#  7470#  7471#  7472#  7473#  7474#
          7475#  7476#  7477#  7478#  7479#  7480#  7487#  7488#  7489#  7490#  7491#  7492#  7493#  7494#  7495#
          7496#  7497#  7498#  7499#  7500#  7507#  7508#  7509#  7510#  7511#  7512#  7513#  7514#  7515#  7516#
          7517#  7518#  7535#  7586#  7589#  7592#  7595#  7613#  7642#  7646#  7647#  7651#  7655#  7660#  7682#
          7689#  7692#  7709#  7725#  7727#  7728#  7747#  7800#  7859#  7899#  7901#  7978#  8005#  8007#  8038#
          8071#  8118#  8120#  8173#  8212#  8225#  8234#  8264#  8287#  8293#  8319#  8324#  8347#  8370#  8374#
          8397#  8398#  8449#  8480#  8491#  8493#  8502#  8504#  8508#  8517#  8526#  8528#  8532#  8541#  8550#
          8552#  8553#  8577#  8579#  8581#  8698#  8831#  8862#  8891#  8892#  8897#  8898#  8902#  8908#  8937#
          8938#  8943#  8944#  8948#  8950#  8984#  9031#  9037#  9084#  9086#  9126#  9148#  9150#  9188#  9203#
          9212#  9253#  9262#  9303#  9312#  9348#  9396#  9436#  9446#  9447#  9456#  9457#  9475#  9480#  9491#
          9492#  9494#  9499#  9510#  9512#  9517#  9522#  9527#  9537#  9538#  9543#  9545#  9550#  9561#  9563#
          9568#  9578#  9580#  9585#  9596#  9598#  9603#  9613#  9615#  9620#  9630#  9631#  9636#  9637#  9642#
          9644#  9648#  9693#  9702#  9747#  9756#  9801#  9810#  9855#  9864#  9909#  9918#  9962#  9972#  10016#
         10026#  10069#  10122#  10134#  10196#  10223#
M$TSTL    2728#  5176#  6224#  6230#  6240#  6246#  6353#  6359#  6517#  6523#  6533#  6539#  6574#  6580#  6614#
          6620#  6628#  6634#  6780#  6785#  6804#  6810#  6819#  6825#  6834#  6840#  6849#  6855#  7105#  7110#
          7119#  7196#  7212#  7251#  7257#  7371#  7372#  7377#  7378#  7379#  7380#  7381#  7382#  7383#  7384#
          7385#  7392#  7393#  7394#  7395#  7396#  7397#  7398#  7399#  7400#  7401#  7402#  7403#  7404#  7411#
          7412#  7413#  7414#  7415#  7416#  7417#  7418#  7419#  7420#  7421#  7422#  7423#  7430#  7431#  7432#
          7433#  7434#  7435#  7436#  7437#  7438#  7439#  7440#  7441#  7442#  7443#  7450#  7451#  7452#  7453#
          7454#  7455#  7456#  7457#  7458#  7459#  7460#  7461#  7462#  7469#  7470#  7471#  7472#  7473#  7474#
          7475#  7476#  7477#  7478#  7479#  7480#  7487#  7488#  7489#  7490#  7491#  7492#  7493#  7494#  7495#
          7496#  7497#  7498#  7499#  7500#  7507#  7508#  7509#  7510#  7511#  7512#  7513#  7514#  7515#  7516#
          7517#  7518#  7535#  7586#  7589#  7592#  7595#  7613#  7642#  7646#  7647#  7651#  7655#  7660#  7682#
          7689#  7692#  7709#  7725#  7727#  7728#  7747#  7800#  7859#  7899#  7901#  7978#  8005#  8007#  8038#
          8071#  8118#  8120#  8173#  8212#  8225#  8234#  8264#  8287#  8293#  8319#  8324#  8347#  8370#  8374#
          8397#  8398#  8449#  8480#  8491#  8493#  8502#  8504#  8508#  8517#  8526#  8528#  8532#  8541#  8550#
          8552#  8553#  8577#  8579#  8581#  8698#  8831#  8862#  8891#  8892#  8897#  8898#  8902#  8908#  8937#
          8938#  8943#  8944#  8948#  8950#  8984#  9031#  9037#  9084#  9086#  9126#  9148#  9150#  9188#  9203#
          9212#  9253#  9262#  9303#  9312#  9348#  9396#  9436#  9446#  9447#  9456#  9457#  9475#  9480#  9491#
          9492#  9494#  9499#  9510#  9512#  9517#  9522#  9527#  9537#  9538#  9543#  9545#  9550#  9561#  9563#
          9568#  9578#  9580#  9585#  9596#  9598#  9603#  9613#  9615#  9620#  9630#  9631#  9636#  9637#  9642#
          9644#  9648#  9693#  9702#  9747#  9756#  9801#  9810#  9855#  9864#  9909#  9918#  9962#  9972#  10016#
         10026#  10069#  10122#  10134#  10196#  10223#
M$WORD    2888#  5176#  5216#  5239#  6224#  6230#  6240#  6246#  6353#  6359#  6517#  6523#  6533#  6539#  6574#
          6580#  6614#  6620#  6628#  6634#  6780#  6785#  6804#  6810#  6819#  6825#  6834#  6840#  6849#  6855#
          7196#  7212#  7251#  7257#  7655#  7859#  7899#  7978#  8005#  8118#  8173#  8212#  8225#  8502#  8526#
          8550#  8891#  8897#  8937#  8943#  9126#  9148#  9188#  9203#  9253#  9303#  9446#  9456#  9491#  9510#
          9537#  9543#  9561#  9578#  9596#  9613#  9630#  9636#  9642#  9693#  9747#  9801#  9855#  9909#  9962#
         10016#  10122#  10134#  10196#  10245#  10246#  10247#  10248#  10249#  10250#  10251#  10252#  10289#  10290#  10291#
         10292#  10321
M$XFER    2410#  5176#
OPEN      1694#  5176#
POINTE    1702#  5176#  5206
PRINTB    1768#  5176#  7371   7377   7378   7379   7380   7392   7393   7394   7395   7411   7412   7413   7414
          7430   7431   7432   7433   7434   7450   7451   7452   7453   7469   7470   7471   7487   7488   7489
```

```
        7490    7491    7507    7508    7509
PRINTF  1811#   5176#   7105    7110    7119    7646    7651    7727    8491    8493    8517    8541    8577    8579
PRINTS  1854#   5176#
PRINTX  1897#   5176#   7381    7382    7383    7384    7396    7397    7398    7399    7400    7401    7402    7403    7415
        7416    7417    7418    7419    7420    7421    7422    7435    7436    7437    7438    7439    7440    7441    7442
        7454    7455    7456    7457    7458    7459    7460    7461    7472    7473    7474    7475    7476    7477    7478
        7479    7492    7493    7494    7495    7496    7497    7498    7499    7510    7511    7512    7513    7514    7515
        7516    7517
READBU  1940#   5176#
READEF  1949#   5176#   7586    7589    7592    7595
RFLAGS  1967#   5176#
SETPRI  1977#   5176#   7682
SETVEC  1986#   5176#
SLASH   1998#   5176#
STARS   2015#   5176#
SVC     2036#   5175#   5176
XFER    2299#   5176#
XFERF   2310#   5176#
XFERT   2319#   5176#


. ABS.  037750     000


? ERRORS DETECTED:  1

 SAIL:CZDMSA,CZDMSA/CRF/NL:TOC=CZDMP.MLB,CZDMSA.P11
 RUN-TIME: 153 184 15 SECONDS
 RUN-TIME RATIO: 2054/353=5.8
 CORE USED:  19K  (37 PAGES)
```