DMP-11 M8203 STATIC DIAG.#1 AH-E232D-MC

DMR-11 CZDMRD0 FICHE 1 OF 2

DMP-11 M8203 STATIC DIAG.#1 AH-E232D-MC
DMR-11 CZDMRD0 FICHE 2 OF 2

NOV 1980
COPYRIGHT © 79-80
MADE IN USA

```
                        .REM @

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

                        PRODUCT CODE:   AC-E231D-MC

                        PRODUCT NAME:   CZDMRDO M8203 STATIC DIAG #1

                        PRODUCT DATE:   AUGUST 1980

                        MAINTAINER:     DIAGNOSTIC ENGINEERING

                        AUTHOR:         DAVID HOFFMAN
```

CONTENTS

1.0 INTRODUCTION

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

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

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

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

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

2.0 HARDWARE REQUIREMENTS

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

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

M8203 LINE UNIT AND BC08S-1 CABLE AND BERG CONNECTORS

3.0 PRELIMINARY PROGRAM REQUIREMENTS

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

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

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

4.2 EXECUTION TIME

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

4.3 XXDP+

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

4.4 ACT/SLIDE

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

4.5 APT

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

4.6 MEMORY MANAGEMENT

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

4.7 MEMORY PARITY OPTION

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

4.8 ERROR LOGGING

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

5.0 PROGRAM LOAD MEDIA

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

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

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

6.1.2 STARTING PROCEDURES

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

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

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

6.2 INITIAL DIALOGUE

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

```
DRS LOADED
DIAG. RUN-TIME SERVICES
CZDMR-D-0
M8203 STATIC LOGIC TESTS - PART 1 OF 2
UNIT IS M8203
DR>
```

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

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

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

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

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

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

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

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

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

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

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

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

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

6.3.1.5 EFFECT OF START COMMAND

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

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION "# UNITS?" TO WHICH THE OPERATOR REPLIES WITH A DECIMAL NUMBER N FROM 1 TO 16. THE TERM "UNIT" REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE

AFTER THE PARENTHESES.

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

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

EXAMPLE:

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

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

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

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

6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT

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

6.3.3 CONTINUE COMMAND

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

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

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

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

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

6.3.3.3 EFFECT OF CONTINUE COMMAND

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

6.3.4 PROCEED COMMAND

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

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

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

6.3.4.2 EFFECT OF PROCEED COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND

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

6.3.5 ADD COMMAND

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

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

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

6.3.5.2 EFFECT OF ADD COMMAND

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

6.3.6 DROP COMMAND

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

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

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

6.3.6.2 EFFECT OF DROP COMMAND

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

6.3.7 PRINT COMMAND

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

6.3.7.1 EFFECT OF PRINT COMMAND

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

6.3.8 DISPLAY COMMAND

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

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

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

6.3.8.2 EFFECT OF DISPLAY COMMAND

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

6.3.9 FLAGS COMMAND

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

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

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

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

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

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR DIALOGUES- HARD CORE QUESTIONS (SEE 6.2),HARDWARE DIALOGUE

(SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

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

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 4 QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

1. DEVICE CSR ADDRESS : (O) 160170?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SEL0) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000-177776 (OCTAL), AND THE DEFAULT VALUE IS 160170.

2. DEVICE VECTOR ADDRESS : (O) 300 ?

THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE DEFAULT VALUE IS 300.

3. DEVICE PRIORITY LEVEL : (O) 5 ?

THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF THIS DEVICE WILL BE EXECUTED. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS 5.

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

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

6.3.13 SOFTWARE PARAMETERS

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

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

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

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

QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

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

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

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

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

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

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D) ? 16
UNIT 0
<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

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

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

SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

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

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

```
7.0 DEVICE INFORMATION TABLES


;*************************************************************************
* MAINTENANCE REGISTER - BSEL1
;*************************************************************************
RUN     = BIT7
MCLR    = BIT6
STEPLU  = BIT4
LULOOP  = BIT3
ROMO    = BIT2
ROMI    = BIT1
STEPMP  = BIT0

;*************************************************************************
* OBUS REG 10 - TRANSMITTER BUFFER
;*************************************************************************
TX7     = BIT7
TX6     = BIT6
TX5     = BIT5
TX4     = BIT4
TX3     = BIT3
TX2     = BIT2
TX1     = BIT1
TX0     = BIT0

;*************************************************************************
* OBUS REG 11
;*************************************************************************
OC      = BIT7
GOAH    = BIT3
ABORT   = BIT2
EOM     = BIT1
SOM     = BIT0

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

;*************************************************************************
* OBUS REG 13
;*************************************************************************
POLL    = BIT7
DTR     = BIT6
SELFR   = BIT5
HDX     = BIT4
MAINT1  = BIT3
MAINT2  = BIT2
SELSBY  = BIT1

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

```
* OBUS REG 14
;***************************************************************************************
TXEN    = BIT6
DISSI   = BIT5
RDAX    = BIT4
WAX     = BIT3
ENAX    = BIT2
AX2     = BIT1
AX1     = BIT0

;***************************************************************************************
* OBUS REG 17
;***************************************************************************************
CRC2    = BIT7
CRC1    = BIT6
IDLE    = BIT5
SECA    = BIT4
STRIP   = BIT3
RDALL   = 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
;*************************************************************************
TERR    = BIT7
TXGA    = BIT3
TXAB    = BIT2
TEOM    = BIT1
TSOM    = BIT0

;*************************************************************************
* AX2-15 - USYRT REG 4
;*************************************************************************
SYN7    = BIT7
SYN6    = BIT6
SYN5    = BIT5
SYN4    = BIT4
SYN3    = BIT3
SYN2    = BIT2
SYN1    = BIT1
SYN0    = BIT0
SYNCH   = 226

;*************************************************************************
* AX2-16 - USYRT REG 5
;*************************************************************************
APA     = BIT7
DDC     = BIT6
STR     = BIT5
SEC     = BIT4
IDL     = BIT3
CRCTY2  = BIT2
CRCTY1  = BIT1
CRCTY0  = BIT0

;*************************************************************************
* AX3-15 - USYRT REG 6
;*************************************************************************
I422    = BIT7
XYZ     = BIT6
C32BCC  = BIT5
V35     = BIT4
```

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

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

```
8.0 TEST DESCRIPTIONS



;*****************************************************************************
;    TEST 1 - MICROPROCESSOR CSR ADDRESSING TEST (SEL0)
;*
;* THIS TEST ADDRESSES THE FIRST MICROPROCESSOR CSR (SEL0), TO MAKE SURE
;* THAT A NON-EXISTENT MEMORY TIME-OUT TRAP DOES NOT OCCUR WHILE
;* ATTEMPTING TO ADDRESS THE MICROPROCESSOR.
;*****************************************************************************




;*****************************************************************************
;    TEST 2 - INBUS/OUTBUS REG 14 INITIALIZATION TEST
;*
;* MASTER CLEAR (MCLR) IS SET IN THE MICROPROCESSOR, IBUS REG 14 IS READ
;* AND COMPARED TO 200.
;*****************************************************************************




;*****************************************************************************
;    TEST 3 - INBUS/OUTBUS REG 14 READ/WRITE BIT TEST
;*
;* WRITE, READ, AND COMPARE ALL WORDS OF DATA PATTERN A INTO REG 14,
;* A BYTE AT A TIME. NON-R/W BITS ARE MASKED OFF TO 0 BEFORE WRITING AND
;* READING.
;*    DATA PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
;*                     375,373,367,357,337,277,177.
;*****************************************************************************




;*****************************************************************************
;    TEST 4 - REG 14 MASTER CLEAR TEST
;* WRITE 377 INTO REG 14, ISSUE MASTER CLEAR, READ REG 14 AND COMPARE
;* TO 200.
;*****************************************************************************
```

```
;*****************************************************************************
        TEST 5 - REG 14 UNIBUS RESET (INIT) TEST
* WRITE 377 INTO REG 14, ISSUE UNIBUS RESET (INIT), READ REG 14 AND COMPARE
* TO 200.
;*****************************************************************************






;*****************************************************************************
        TEST 6 - LINE UNIT FALSE SELECTION TEST
*
* FIRST, A MASTER CLEAR IS PERFORMED. THEN, THE PROGRAM SINGLE-STEPS THE
* MICROPROCESSOR THROUGH AN INSTRUCTION WHICH LOADS 041 (OCT) INTO THE MAR
* REGISTER (OBUS* ADRS 14). THEN, THE LINE UNIT REGISTER 14 IS READ AND CHECKED
* TO BE UNAFFECTED (STILL = 0). THIS TEST IS INTENDED TO DETECT A FALSE
* SELECTION OF THE LINE UNIT REGISTERS, WHEN THE LINE UNIT IS NOT BEING
* ACCESSED.
;*****************************************************************************





;*****************************************************************************
        TEST 7 - INBUS REG MASTER CLEAR TEST
*
* FIRST, ALL READ/WRITE BITS OF REGS 10-17 ARE SET BY LOADING A
* DIFFERENT WORD OF PATTERN G INTO EACH REG.  THEN,
* A MASTER CLEAR IS ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF
* PATTERN M, WHICH CONTAINS THE INITIALIZED STATES OF THE REGS. (UNPREDICTABLE
* BITS ARE MASKED OFF TO 0 BEFORE COMPARISON).
*       PATTERN G = 000,000,240,120,177,000,000,001
*       PATTERN M = 000,020,000,000,200,000,000,051
;*****************************************************************************





;*****************************************************************************
        TEST 8 - REGISTER 10-17 ADDRESSING TEST
*
* FIRST, A MASTER CLEAR IS ISSUED. THEN,
* WRITE A DIFFERENT WORD OF DATA PATTERN B INTO EACH OF REGS 10-17,
* AND AFTER EACH WRITE, READ AND COMPARE ALL REGS TO EXPECTED VALUES.
* UNPREDICTABLE BITS ARE MASKED OFF TO 0 WHEN READ FOR COMPARISON.
*        PATTERN B = 000,000,040,100,220,000,000,051
;*****************************************************************************





;*****************************************************************************
        TEST 9 - REG 11 READ/WRITE BIT TEST
```

```
*
* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN C INTO REG 11 :
*      DATA PATTERN C = 020,020,020.
;*****************************************************************************

;*****************************************************************************
       TEST 10 - REG 12 READ/WRITE BIT TEST
*
* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN D INTO REG 12 :
*      DATA PATTERN D = 000,040,000.
;*****************************************************************************

;*****************************************************************************
       TEST 11 - REG 13 READ/WRITE BIT TEST
*
* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN E INTO REG 13 :
*      DATA PATTERN E = 000,120,020,100,120,000.
;*****************************************************************************

;*****************************************************************************
       TEST 12 - REG 17 READ/WRITE BIT TEST
*
* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN F INTO REG 17 :
*      DATA PATTERN F = 050,051,050.
;*****************************************************************************

;*****************************************************************************
       TEST 13 - MAINTENANCE CLOCK BIT TEST
*
* FIRST, A MASTER CLEAR IS ISSUED TO INIT ALL REGS. THEN, THE MICROPROCESSOR
* IS PLACED IN A LOOP ON AN INSTRUCTION, BY SETTING THE INSTRUCTION IN SEL6
* AND SETTING ROMI AND RUN IN BSEL1. THE INSTRUCTION IS ONE WHICH REPETITIVELY
* READS LINE UNIT REG 17 INTO BSEL2. THE PDP-11 CAN THEN SCAN BSEL2 TO MONITOR
* THE MAINTENANCE CLOCK BIT, MCLK. THE FOLLOWING SEQUENCE IS THEN PERFORMED
* TO MONITOR MCLK :
* - THE PROGRAM REPEATEDLY CHECKS THE MCLK BIT FOR THE 1 STATE, AND IF IT IS
*     NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC (DEPENDING ON THE PROCESSOR)
*     AN ERROR IS REPORTED. (THE MAINTENANCE CLOCK HAS A PERIOD OF 41.6 MICRO-
*     SEC).
* - THE PROGRAM NEXT REPEATEDLY CHECKS THE MCLK BIT FOR THE 0 STATE, AND IF
*     IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC AN ERROR IS REPORTED.
```

```
* - THE PROGRAM NEXT REPEATEDLY CHECKS MCLK BIT FOR THE 1 STATE AGAIN, AND
*    IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC, AN ERROR IS REPORTED.
*
* IF THE P-TABLE FOR THIS UNIT INDICATES THAT THE M8207 RUN SWITCH (E28 SW7)
* IS OFF, THE TEST WILL BE SKIPPED.
;*****************************************************************************





;*****************************************************************************
     TEST 14 - EXTENDED REGISTER MASTER CLEAR TEST
*
* FIRST, ALL READ/WRITE BITS OF EXTENDED REGS AX0-AX3 ARE SET BY LOADING
* A DIFFERENT WORD OF PATTERN H INTO EACH REG. THEN, A MASTER CLEAR IS
* ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF PATTERN I, WHICH
* CONTAINS THE INITIALIZED STATES OF ALL THE EXTENDED REGS.
*     PATTERN H = 000,000,377,017,377,377,375,377
*     PATTERN I = 000,000,000,000,000,103,000,000
;*****************************************************************************





;*****************************************************************************
     TEST 15 - EXTENDED REGISTER ADDRESSING TEST
*
* FIRST, ISSUE A MASTER CLEAR TO PUT REGS INTO INITIALIZED STATES SHOWN IN
* PATTERN I. THEN, WRITE A DIFFERENT WORD OF PATTERN J INTO EACH EXTENDED (AX)
* REG, AND AFTER EACH WRITE, READ AND COMPARE ALL EXTENDED REGS TO EXPECTED
* VALUES.
*    PATTERN I = 000,000,000,000,000,103,000,000
*    PATTERN J = 000,000,001,002,004,103,040,100
;*****************************************************************************





;*****************************************************************************
     TEST 16 - REGS 15,16 / AX2-15,AX2-16 READ/WRITE BIT TEST
*
* USING REGS 15,16, THE INDIRECT REGS AX2-15,AX2-16 (USYRT REGS 4,5) ARE
* WRITTEN AND READ USING EACH WORD OF PATTERN K. AX2-15 IS COMPARED
* TO THE WORD WRITTEN, AND AX2-16 IS ALWAYS COMPARED TO 103. (AX2-16 IS NOT
* WRITEABLE).
*     PATTERN K =
*       FOR REG 15: 000,377,125,252,001,002,004,010,020,040,100,200,000,000,
*                   000,000,000,000,000,000,376,375,373,367,357,337,277,177,
*                   377,377,377,377,377,377,377,377
*       FOR REG 16: 000,377,125,252,000,000,000,000,000,000,000,000,001,002,
*                   004,010,020,040,100,200,377,377,377,377,377,377,377,377,
*                   376,375,373,367,357,337,277,177.
;*****************************************************************************
```

```
;************************************************************************************
    TEST 17 - AX0-15,AX0-16 READ/WRITE BIT TEST
*
* IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
* ARE PERFORMED IN REGS AX0-15,AX0-16 USING EACH WORD OF PATTERN L.
* ANY BITS IN AX0-15,AX0-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
* IN THE EXPECTED VALUE BEFORE COMPARISON.
*     PATTERN L =
*         FOR REG 15: 000,377,000
*         FOR REG 16: 000,377,000.
;************************************************************************************


;************************************************************************************
    TEST 18 - AX1-15,AX1-16 READ/WRITE BIT TEST
*
* IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
* ARE PERFORMED IN REGS AX1-15,AX1-16 USING EACH WORD OF PATTERN K.
* ANY BITS IN AX1-15,AX1-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
* IN THE EXPECTED VALUE BEFORE COMPARISON.
;************************************************************************************


;************************************************************************************
    TEST 19 - AX3-15,AX3-16 READ/WRITE BIT TEST
*
* IN THIS TEST A MASTER CLEAR IS DONE AND THEN A WRITE, READ, AND COMPARE ARE
* PERFORMED IN REGS AX3-15,AX3-16 USING EACH WORD OF PATTERN V FOR WRITING,
* AND PATTERN U FOR COMPARING.
* ANY BITS IN AX3-15,AX3-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
* IN THE EXPECTED VALUE BEFORE COMPARISON.
*   PATTERN V =
*       FOR REG 15 : 000,333,331,323,313,233,133,000,000,000,000,
*                    000,000,000,000,000,000,000,000
*       FOR REG 16 : 000,000,000,000,000,000,000,001,002,004,040,
*                    100,200,346,345,343,307,247,147
*   PATTERN U =
*       FOR REG 15 : 000,001,013,011,021,101,301,000,000,000,000,
*                    000,000,000,000,000,000,000,000
*       FOR REG 16 : 000,000,000,000,000,000,000,001,002,004,040,
*                    100,200,346,345,343,307,247,147
;************************************************************************************
```

```
;*************************************************************************************
	TEST 20 - REG 17 - AX2-16 READ/WRITE, MASTER CLEAR TEST
*
* THIS TEST CONSISTS OF 2 SUBTESTS. IN THE FIRST SUBTEST, EACH BYTE OF PAT O
* IS WRITTEN INTO REG 17 AND AFTER EACH WRITE, AX2-16 IS READ AND COMPARED
* TO A BYTE OF PAT P.
*   PATTERN O = 000,041,004,010,020,040,100,101,200,201,300,111,301,375
*   PATTERN P = 000,113,200,040,020,010,001,104,007,105,007,144,107,157
* IN THE SECOND SUBTEST, REG 17 IS LOADED WITH 375, A MASTER CLEAR IS ISSUED,
* AND AX2-16 IS COMPARED TO ITS INITIALIZED STATE (103).
;*************************************************************************************






;*************************************************************************************
	TEST 21 - TRANSMITTER BUFFER DATA TEST
* A MASTER CLEAR IS DONE FIRST, AND THEN A BYTE OF PATTERN N IS LOADED INTO
* REG 11 AND THE NEXT BYTE IS LOADED TWICE INTO REG 10. THE PROGRAM THEN WAITS
* AT LEAST 50 MICRO-SEC, AND THEN IT READS AND COMPARES AX1-15 TO THE BYTE WHICH
* WAS LOADED INTO REG 10, AND IT READS AND COMPARES AX1-16 TO THE BYTE WHICH WAS
* LOADED INTO REG 11. THIS PROCESS IS REPEATED (INCLUDING THE MASTER CLEAR)
* FOR EACH PAIR OF BYTES IN PATTERN N.
*   PATTERN N =
*     FOR REG 10: 000,125,252,377,000,000,000
*     FOR REG 11: 000,000,000,000,005,012,017
;*************************************************************************************






;*************************************************************************************
	TEST 22 - TRANSMITTER BUFFER SEQUENCING TEST
*
* FIRST, A MASTER CLEAR IS DONE, AND THE PROGRAM CHECKS FOR ORDY=1, OCOR=0.
* THEN, 2 TSOM CHARS ARE LOADED INTO THE TX SILO, AND ALLOWED TO RIPPLE
* DOWN TO THE OUTPUT. THE PROGRAM CHECKS FOR ORDY=1, OCOR=1.
* NEXT, THE PROGRAM CYCLES THE STEPLU BIT UNTIL OCOR=0 AGAIN, AND CHECKS FOR
* THIS TO OCCUR WITHIN 3 CYCLES.
* THE SILO IS THEN FILLED WITH 64 BYTES OF A 256-BYTE BINARY COUNT PATTERN
* (000-377) AND THE PROGRAM CHECKS FOR ORDY=0 AFTER THE 64TH CHAR IS LOADED.
* THE PROGRAM CYCLES STEPLU FOR 8 CYCLES AND CHECKS THAT AFTER THE 8TH, ORDY=1.
* AX1-15 IS READ AND COMPARED TO EXPECTED DATA.
* THE REST OF THE BINARY COUNT DATA BYTES ARE LOADED, CYCLED 8 CLOCKS, READ AND
* COMPARED, A BYTE AT A TIME. UPON COMPLETION, THE SILO IS CHECKED TO BE EMPTY
* WITH ORDY=1, OCOR=0.
;*************************************************************************************






;*************************************************************************************
	TEST 23 - TX MSG TIMING TEST, CHAR MODE, WITH CRC
*
```

```
* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-16.
* THE FOLLOWING STEPS ARE DONE:
* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
* THEN 2 TERMINATING SYNCHS ARE SENT.
* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
* CLEARED STATE AFTER THE 3RD SYNCH COMPLETES.
;*****************************************************************************






;*****************************************************************************
        TEST 24 - TX MSG TIMING TEST, BIT MODE, WITH CRC
*
* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
* BIT ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-CCITT-1.
* THE FOLLOWING STEPS ARE DONE:
* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN BIT MODE.
* SOM IS SET TWICE TO SEND 2 FLAG CHARS. THEN, 2 000 CHARS ARE SENT, AND
* THEN 2 TERMINATING FLAGS ARE SENT.
* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
* CLEARED STATE.
;*****************************************************************************





;*****************************************************************************
        TEST 25 - TX MSG TIMING TEST, CHAR MODE, WITH NO CRC
*
* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND NO ERROR CHECKING.
* THE FOLLOWING STEPS ARE DONE:
* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
* THEN 2 TERMINATING SYNCHS ARE SENT.
* THE TEST IS PERFORMED WITH TXEN (REG 14, BIT6) SET, AND THE PROGRAM CHECKS
* THAT THIS HOLDS RTS HIGH PAST THE END OF THE MESSAGE.
* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
* CLEARED STATE.
;*****************************************************************************





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

```
	TEST 26 - TX UNDERRUN SET AND CLEAR TEST - CHAR MODE
*
* IN THIS TEST, A TX UNDERRUN ERROR IS FORCED IN EACH OF 2 SITUATIONS,
* AND THEN CLEARED DIFFERENTLY IN EACH.
* IN THE FIRST, A MESSAGE IS INITIATED, A 000 CHAR IS SENT, AND THE TX
* BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG,
* WHICH CAUSES UNRR TO SET IN REG 11. THEN, SOM IS SET TO CLEAR THE ERROR,
* AND THIS IS VERIFIED.
* IN THE SECOND SITUATION, A MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX
* BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH
* AGAIN CAUSES UNRR TO SET. THEN, A MASTER CLEAR IS DONE, AND THE UNRR BIT
* IS CHECKED TO BE CLEARED.
;*******************************************************************************





;*******************************************************************************
	TEST 27 - TRANSMIT CHAR LENGTH TIMING TEST - CHAR MODE, CRC
*
* THE LINE UNIT IS PLACED IN CHAR MODE (DDCMP) AND A MESSAGE IS INITIATED
* WITH AN 8-BIT SYNCH AND A 5-BIT SYNCH CHAR. NEXT, A 000 CHAR IS SENT WITH
* EACH OF THE FOLLOWING TX CHAR LENGTHS : 5 BITS, 6 BITS, 7 BITS, 8 BITS.
* (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED). TWO
* TERMINATING SYNCHS ARE SENT AFTER THE DATA.
;*******************************************************************************





;*******************************************************************************
	TEST 28 - TRANSMIT CHAR LENGTH TIMING TEST - BIT MODE, CRC
*
* THE LINE UNIT IS PLACED IN BIT MODE AND A MESSAGE IS INITIATED
* WITH 2 FLAG CHARS. NEXT, 2 8-BIT 000 CHARS ARE SENT, FOLLOWED BY 000 CHARS
* WITH EACH OF THE FOLLOWING TRANSMITTER CHAR LENGTHS:
*	1 BIT, 2 BITS, 3 BITS, 4 BITS, 5 BITS, 6 BITS, 7 BITS, AND 8 BITS.
* (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED).
* TWO TERMINATING FLAGS ARE SENT AFTER THE DATA.
;*******************************************************************************





;*******************************************************************************
	TEST 29 - TXDATA BIT TEST - CHAR MODE, CRC
*
* THE LINE UNIT IS INITIALIZED AND A MSG IS INITIATED (USING STEPLU) WITH CRC-
* 16 SELECTED IN CHAR MODE. TWO SYNCHS, 000,125,252,377,000, AND 2 TERMINATING
* SYNCHS ARE THEN SENT. THE PROGRAM CHECKS EACH BIT OF THE TRANSMITTED
* DATA CHARS, BY MONITORING TXDATA (REG 17) AS THE DATA IS CLOCKED OUT OF
* THE USYRT TRANSMITTER.
;*******************************************************************************
```

```
;*************************************************************************************
	TEST 30 - USYRT RECEIVER MSG TEST - CHAR MODE, CRC
*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-16
* SELECTED. TWO SYNCHS, 000,125,252,377,000, AND FOUR TERMINATING SYNCHS ARE
* SENT. THE PROGRAM MONITORS IACT, AND THE RCV'D CHARS AND CRC BYTES ARE READ
* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
*  STILL SET AFTER THE MESSAGE.
;*************************************************************************************





;*************************************************************************************
	TEST 31 - USYRT RECEIVER MSG TEST - BIT MODE, CRC
*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-
* CCITT-1. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS ARE THEN
* SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE READ
* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
;*************************************************************************************





;*************************************************************************************
	TEST 32 - USYRT RECEIVER MSG TEST - CHAR MODE, NO CRC
*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH NO
* ERROR DETECTION. TWO SYNCHS, 000,125,252,377,000, AND TWO SYNCHS ARE
* THEN SENT. THE PROGRAM MONITORS IACT, AND THE RECEIVED CHARS ARE READ FROM
* AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
* STILL = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT = 0.
;*************************************************************************************





;*************************************************************************************
	TEST 33 - USYRT RECEIVER MSG TEST - BIT MODE, NO CRC
*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH ERROR
* DETECTION INHIBITED. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS
* ARE THEN SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE
* READ FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
```

```
* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
;*******************************************************************************







;*******************************************************************************
    TEST 34 - SILO-DISABLED TRANSMITTER LOAD TEST
*
* THIS TEST DISABLES THE SILOS, LOADS A 125 CHARACTER INTO THE TX SILO, AND
* READS AX1-15 AND CHECKS THAT THE DATA DID NOT GET LOADED INTO THE USYRT TX
* BUFFER.
;*******************************************************************************






;*******************************************************************************
    TEST 35 - SILO-DISABLED MESSAGE TEST - BIT MODE, NO CRC
*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU)
* WITH LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, WITH SILO
* DISABLE SET, AND WITH NO ERROR DETECTION. TWO FLAGS, 000,125,252, AND
* TERMINATING FLAGS ARE THEN SENT BY LOADING THE TRANSMITTED CHARS INTO
* REG AX1. THE PROGRAM MONITORS OACT, IACT, RSOM, REOM, ORDY, OCOR, ICIR,
* IRDY, AND THE RECEIVED CHARS ARE READ FROM AXO AND COMPARED TO EXPECTED
* VALUES.
;*******************************************************************************






;*******************************************************************************
    TEST 36 - RECEIVER BUFFER TEST - CHAR MODE, CRC
*
* FIRST, A MASTER CLEAR IS DONE AND THE PROGRAM CHECKS FOR ICIR = 1 AND IRDY
* = 0. THEN, 2 SOM CHARS ARE LOADED AND CLOCKED INTO THE USYRT, AND 64
* BYTES OF A 256-BYTE BINARY COUNT DATA PATTERN (000-377) ARE LOADED INTO
* THE TX SILO.
* THE LINE UNIT IS THEN CLOCKED UNTIL IRDY = 1, AND THE PROGRAM CHECKS FOR
* THIS TO OCCUR WITHIN 40-43 CYCLES. THE PROGRAM READS THE RCV SILO, CHECKS THE
* CHAR FOR 000, AND CHECKS FOR IRDY = 0 AGAIN.
* THE LINE UNIT IS THEN CLOCKED IN GROUPS OF 8 CYCLES, AND AFTER EACH, THE
* PROGRAM CHECKS FOR ICIR = 1, IRDY = 1, UNTIL THE 64TH GROUP, AFTER WHICH
* IT CHECKS FOR ICIR = 0, IRDY = 1. THE SECOND DATA CHAR IS READ FROM THE
* RECEIVER SILO AND COMPARED TO 001. THEN, THE PROGRAM CHECKS FOR ICIR = 1,
* IRDY = 1 AGAIN.
* THE REST OF THE BINARY COUNT DATA BYTES ARE CYCLED 8 CLOCKS AND READ AND
* COMPARED A BYTE AT A TIME.
;*******************************************************************************
```

```
;*******************************************************************************
        TEST 37 - RECEIVER CHAR LENGTH TIMING TEST - CHAR MODE, NO CRC
*
* THE LINE UNIT IS PLACED IN CHAR MODE, WITH NO ERROR DETECTION, AND A MSG IS
* INITIATED WITH 2 SYNCH CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
* SET, WHILE THE RECEIVER CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 5,6,7,8.
* FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THET USYRT RECEIVER
* FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV CHAR LENGTH.
* (FOR EXAMPLE A 5-BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED). A MASTER
* CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
;*******************************************************************************






;*******************************************************************************
        TEST 38 - RECEIVER CHAR LENGTH TIMING TEST - BIT MODE, NO CRC
*
* THE LINE UNIT IS PLACED IN BIT MODE WITH NO ERROR DETECTION, AND A MESSAGE IS
* INITIATED  WITH 2 FLAG CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
* SET, WHILE THE RCV CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 8,8,8,7,6,5,
* 4,3,2,1. FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THAT THE
* USYRT RECEIVER FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV
* CHAR LENGTH. (FOR EXAMPLE, A 5 BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED).
* A MASTER CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
;*******************************************************************************






;*******************************************************************************
        TEST 39 - TRANSMITTER UNDERRUN ERROR, IDLE MARKING, CHAR MODE,NO CRC

* THE LINE UNIT IS PLACED IN CHAR MODE, AND THE IDLE BIT IS SET. THEN, A
* MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX BUFFER IS NOT SERVICED
* IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH CAUSES A TX UNDERRUN
* ERROR. THEN, THE RECEIVER IS CLOCKED AND CHECKED FOR TWO 377 CHARS TO BE
* RECEIVED (LINE MARKING) BEFORE SHUTTING DOWN WITH A MASTER CLEAR.
;*******************************************************************************






;***************************************************************************
        TEST 40 - MSG TERMINATION WITH GA CHARS - BIT MODE,NO CRC
*
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
* INITIATED IN BIT MODE.
* 2 FLAG CHARACTERS ARE SENT, FOLLOWED BY
* THE FOLLOWING DATA CHARACTERS : 000, 125, 252, 377, 000. THEN THE LOOP
* MODE BIT (STRIP) IS SET AND 2 TERMINATING GO-AHEAD CHARACTERS ARE
```

```
* SENT. EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 5 DATA
* WORDS ARE READ AND COMPARED TO EXPECTED VALUES.
* ALSO, THE FIRST GA CHAR IS CHECKED BY SCANNING THE TXDATA BIT AS THE GA
* IS BEING TRANSMITTED (GA = 376 OCTAL).
* THE TEST ALSO CHECKS FOR SETTING OF RAB AND EBLK
* IN LOOP MODE.
;*******************************************************************





;*******************************************************************
    TEST 41 - IDLE SYNCHS TEST - CHAR MODE
*
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
* INITIATED IN CHAR MODE. 24(DEC) SYNCHS ARE SENT.
* EACH SYNCH IS TIMED AS IT IS RECEIVED, AND THE BITS ARE CHECKED
* FOR A VALID SYNCH CHAR FOR EACH OF THE 22 SYNCHS WHICH FOLLOW
* THE FIRST TWO (THESE PERFORM SYNCHRONIZATION, AND ARE NOT READ).
* WHILE THE LAST SYNCH IS BEING TRANSMITTED, OC IS SET, AND THE
* NEXT CHAR RCV'D AFTER THE SYNCH IS CHECKED TO BE 377 (LINE MARKING).
* THEN, A MASTER CLEAR IS ISSUED.
* THE SYNCH CHAR USED IS 226 (OCTAL).
;*******************************************************************





;*******************************************************************
    TEST 42 - STRIP SYNCH TEST
*
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
* INITIATED IN CHAR MODE AND WITH THE STRIP SYNCH
* BIT SET. THEN 24 (DEC) SYNCHS ARE SENT
* FOLLOWED BY THE FOLLOWING DATA CHARACTERS : 377, 000, 125, 252,
* AND THEN 2 TERMINATING SYNCHS.
* EACH OF THE 23 SYNCHS AFTER THE FIRST ARE CHECKED AT THE TRANSMITTER OUTPUT,
* BY SCANNING THE TXDATA BIT.
* EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 4 DATA WORDS
* ARE READ AND COMPARED TO EXPECTED VALUES.
* FINALLY, THE LINE UNIT IS CLOCKED FOR SEVERAL CHAR TIMES, AND A CHECK
* IS MADE FOR OACT = 0 (TEOM SHOULD CAUSE TX ENABLE TO DROP).
* THE ABOVE TEST IS REPEATED FOR EACH OF THE FOLLOWING SYNCH CHAR DATA
* PATTERNS : 226,000,125,252,376,177.
;*******************************************************************





8.1 DATA PATTERNS USED



***** 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 B *****
PATB:
        .BYTE   000
        .BYTE   000
        .BYTE   040
        .BYTE   100
        .BYTE   220
        .BYTE   000
        .BYTE   000
        .BYTE   051

***** DATA PATTERN C *****
PATC:
        .BYTE   020
        .BYTE   020
        .BYTE   020

***** DATA PATTERN D *****
PATD:
        .BYTE   000
        .BYTE   040
        .BYTE   000

***** DATA PATTERN E *****
PATE:
        .BYTE   000
        .BYTE   120
        .BYTE   020
        .BYTE   100
        .BYTE   120
        .BYTE   000

***** DATA PATTERN F *****
PATF:
        .BYTE   050
```

```
                .BYTE   051
                .BYTE   050

***** DATA PATTERN G *****
PATG:
                .BYTE   000
                .BYTE   000
                .BYTE   240
                .BYTE   120
                .BYTE   177
                .BYTE   000
                .BYTE   000
                .BYTE   001

***** DATA PATTERN H *****
PATH:
                .BYTE   000
                .BYTE   000
                .BYTE   377
                .BYTE   017
                .BYTE   377
                .BYTE   377
                .BYTE   375
                .BYTE   377

***** DATA PATTERN I *****
PATI:
                .BYTE   000
                .BYTE   000
                .BYTE   000
                .BYTE   000
                .BYTE   000
                .BYTE   103
                .BYTE   000
                .BYTE   000

***** DATA PATTERN J *****
PATJ:
                .BYTE   000
                .BYTE   000
                .BYTE   010
                .BYTE   002
                .BYTE   004
                .BYTE   103
                .BYTE   001
                .BYTE   100

***** DATA PATTERN K *****
PATK:           .BYTE   000
                .BYTE   000
                .BYTE   377
                .BYTE   377
                .BYTE   125
                .BYTE   125
                .BYTE   252
                .BYTE   252
                .BYTE   001
```

```
        .BYTE   000
        .BYTE   002
        .BYTE   000
        .BYTE   004
        .BYTE   000
        .BYTE   010
        .BYTE   000
        .BYTE   020
        .BYTE   000
        .BYTE   040
        .BYTE   000
        .BYTE   100
        .BYTE   000
        .BYTE   200
        .BYTE   000
        .BYTE   000
        .BYTE   001
        .BYTE   000
        .BYTE   002
        .BYTE   000
        .BYTE   004
        .BYTE   000
        .BYTE   010
        .BYTE   000
        .BYTE   020
        .BYTE   000
        .BYTE   040
        .BYTE   000
        .BYTE   100
        .BYTE   000
        .BYTE   200
        .BYTE   376
        .BYTE   377
        .BYTE   375
        .BYTE   377
        .BYTE   373
        .BYTE   377
        .BYTE   367
        .BYTE   377
        .BYTE   357
        .BYTE   377
        .BYTE   337
        .BYTE   377
        .BYTE   277
        .BYTE   377
        .BYTE   177
        .BYTE   377
        .BYTE   377
        .BYTE   376
        .BYTE   377
        .BYTE   375
        .BYTE   377
        .BYTE   373
        .BYTE   377
        .BYTE   367
        .BYTE   377
        .BYTE   357
```

```
                        .BYTE   377
                        .BYTE   337
                        .BYTE   377
                        .BYTE   277
                        .BYTE   377
                        .BYTE   177

        ***** DATA PATTERN L *****
        PATL:
                        .BYTE   000
                        .BYTE   000
                        .BYTE   377
                        .BYTE   377
                        .BYTE   000
                        .BYTE   000

        ***** DATA PATTERN M *****
        PATM:
                        .BYTE   000
                        .BYTE   020
                        .BYTE   000
                        .BYTE   000
                        .BYTE   200
                        .BYTE   000
                        .BYTE   000
                        .BYTE   051

        ***** DATA PATTERN N *****
        PATN:
                        .BYTE   000
                        .BYTE   000
                        .BYTE   000
                        .BYTE   125
                        .BYTE   000
                        .BYTE   252
                        .BYTE   000
                        .BYTE   377
                        .BYTE   005
                        .BYTE   000
                        .BYTE   012
                        .BYTE   000
                        .BYTE   017
                        .BYTE   000

        ***** DATA PATTERN O *****
        PATO:
                        .BYTE   000
                        .BYTE   041
                        .BYTE   004
                        .BYTE   010
                        .BYTE   020
                        .BYTE   040
                        .BYTE   100
                        .BYTE   101
                        .BYTE   200
                        .BYTE   201
                        .BYTE   300
```

```
                .BYTE   111
                .BYTE   301
                .BYTE   375

        ***** DATA PATTERN P *****
        PATP:
                .BYTE   000
                .BYTE   113
                .BYTE   200
                .BYTE   040
                .BYTE   020
                .BYTE   010
                .BYTE   001
                .BYTE   104
                .BYTE   007
                .BYTE   105
                .BYTE   007
                .BYTE   144
                .BYTE   107
                .BYTE   157

        ***** DATA PATTERN U *****
        PATU:   .BYTE   000
                .BYTE   000
                .BYTE   001
                .BYTE   000
                .BYTE   013
                .BYTE   000
                .BYTE   011
                .BYTE   000
                .BYTE   021
                .BYTE   000
                .BYTE   101
                .BYTE   000
                .BYTE   301
                .BYTE   000
                .BYTE   000
                .BYTE   001
                .BYTE   000
                .BYTE   002
                .BYTE   000
                .BYTE   004
                .BYTE   000
                .BYTE   040
                .BYTE   000
                .BYTE   100
                .BYTE   000
                .BYTE   200
                .BYTE   000
                .BYTE   346
                .BYTE   000
                .BYTE   345
                .BYTE   000
                .BYTE   343
                .BYTE   000
                .BYTE   307
                .BYTE   000
```

```
                .BYTE   247
                .BYTE   000
                .BYTE   147

        ***** DATA PATTERN V *****
        PATV:   .BYTE   000
                .BYTE   000
                .BYTE   333
                .BYTE   000
                .BYTE   331
                .BYTE   000
                .BYTE   323
                .BYTE   000
                .BYTE   313
                .BYTE   000
                .BYTE   233
                .BYTE   000
                .BYTE   133
                .BYTE   000
                .BYTE   000
                .BYTE   001
                .BYTE   000
                .BYTE   002
                .BYTE   000
                .BYTE   004
                .BYTE   000
                .BYTE   040
                .BYTE   000
                .BYTE   100
                .BYTE   000
                .BYTE   200
                .BYTE   000
                .BYTE   346
                .BYTE   000
                .BYTE   345
                .BYTE   000
                .BYTE   343
                .BYTE   000
                .BYTE   307
                .BYTE   000
                .BYTE   247
                .BYTE   000
                .BYTE   147
```

9.0 ERROR INFORMATION

9.1 ERROR REPORTING

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

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

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

FAILING REG:  INBUS/OUTBUS REG 12

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

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

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

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

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

FAILING REG:  INBUS/OUTBUS REG 12
```

```
						@
```

```
                                   .TITLE CZDMRD M8203 STATIC DIAG #1
             002000                         .=2000

                                            .MCALL  SVC
      002000                                SVC                                ; INITIALIZE SUPERVISOR MACROS

      002000                                BGNMOD  LU1MOD

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

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

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

 002000                                   POINTER BGNAU,BGNDU

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

 002000                                   HEADER  CZDMR,D,0,45.,0
 002000                                                                            L$NAME::
 002000      103                                                                         .ASCII /C/
 002001      132                                                                         .ASCII /Z/
 002002      104                                                                         .ASCII /D/
 002003      115                                                                         .ASCII /M/
 002004      122                                                                         .ASCII /R/
 002005      000                                                                         .BYTE   0
 002006      000                                                                         .BYTE   0
 002007      000                                                                         .BYTE   0
 002010                                                                            L$REV::
 002010      104                                                                         .ASCII  /D/
 002011                                                                            L$DEPO::
 002011      060                                                                         .ASCII  /0/
 002012                                                                            L$UNIT::
 002012   000000                                                                         .WORD   0
 002014                                                                            L$TIML::
 002014   000055                                                                         .WORD   45.
 002016                                                                            L$HPCP::
 002016   035554                                                                         .WORD   L$HARD
 002020                                                                            L$SPCP::
 002020   000000                                                                         .WORD   0
 002022                                                                            L$HPTP::
 002022   002252                                                                         .WORD   L$HW
 002024                                                                            L$SPTP::
 002024   000000                                                                         .WORD   0
 002026                                                                            L$LADP::
 002026   036234                                                                         .WORD   L$LAST
 002030                                                                            L$STA::
 002030   000000                                                                         .WORD   0
 002032                                                                            L$CO::
 002032   000000                                                                         .WORD   0
 002034                                                                            L$DTYP::
 002034   000000                                                                         .WORD   0
 002036                                                                            L$APT::
 002036   000000                                                                         .WORD   0
 002040                                                                            L$DTP::
 002040   002124                                                                         .WORD   L$DISPATCH
 002042                                                                            L$PRIO::
 002042   000000                                                                         .WORD   0
 002044                                                                            L$ENVI::
 002044   000000                                                                         .WORD   0
 002046                                                                            L$EXP1::
```

```
002046  000000                          .WORD   0
002050                          L$MREV::
002050     003                          .BYTE   C$REVISION
002051     003                          .BYTE   C$EDIT
002052                          L$EF::
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::
002060  003462                          .WORD   L$DVTYP
002062                          L$REPP::
002062  000000                          .WORD   0
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::
002070  021632                          .WORD   L$AU
002072                          L$DUT::
002072  021550                          .WORD   L$DU
002074                          L$LUN::
002074  000000                          .WORD   0
002076                          L$DESP::
002076  003470                          .WORD   L$DESC
002100                          L$LOAD::
002100  104035                          EMT     E$LOAD
002102                          L$ETP::
002102  000000                          .WORD   0
002104                          L$ICP::
002104  021116                          .WORD   L$INIT
002106                          L$CCP::
002106  021546                          .WORD   L$CLEAN
002110                          L$ACP::
002110  021466                          .WORD   L$AUTO
002112                          L$PRT::
002112  021110                          .WORD   L$PROT
002114                          L$TEST::
002114  000000                          .WORD   0
002116                          L$DLY::
002116  000000                          .WORD   0
002120                          L$HIME::
002120  000000                          .WORD   0

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

                .EVEN
```

```
                                        .SBTTL  DISPATCH TABLE

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

  002122                                        DISPATCH 42
  002122  000052                                                        .WORD   42
  002124                                                        L$DISPATCH::
  002124  021634                                                        .WORD   T1
  002126  021716                                                        .WORD   T2
  002130  022002                                                        .WORD   T3
  002132  022126                                                        .WORD   T4
  002134  022230                                                        .WORD   T5
  002136  022356                                                        .WORD   T6
  002140  022476                                                        .WORD   T7
  002142  022652                                                        .WORD   T8
  002144  023106                                                        .WORD   T9
  002146  023224                                                        .WORD   T10
  002150  023342                                                        .WORD   T11
  002152  023460                                                        .WORD   T12
  002154  023576                                                        .WORD   T13
  002156  024226                                                        .WORD   T14
  002160  024560                                                        .WORD   T15
  002162  025050                                                        .WORD   T16
  002164  025260                                                        .WORD   T17
  002166  025464                                                        .WORD   T18
  002170  025670                                                        .WORD   T19
  002172  026126                                                        .WORD   T20
  002174  026360                                                        .WORD   T21
  002176  026632                                                        .WORD   T22
  002200  027310                                                        .WORD   T23
  002202  027430                                                        .WORD   T24
  002204  027540                                                        .WORD   T25
  002206  027734                                                        .WORD   T26
  002210  030300                                                        .WORD   T27
  002212  030510                                                        .WORD   T28
  002214  031040                                                        .WORD   T29
  002216  031140                                                        .WORD   T30
  002220  031510                                                        .WORD   T31
  002222  032052                                                        .WORD   T32
  002224  032322                                                        .WORD   T33
  002226  032664                                                        .WORD   T34
  002230  033014                                                        .WORD   T35
  002232  033726                                                        .WORD   T36
  002234  034210                                                        .WORD   T37
  002236  034410                                                        .WORD   T38
  002240  034710                                                        .WORD   T39
  002242  035000                                                        .WORD   T40
  002244  035142                                                        .WORD   T41
  002246  035310                                                        .WORD   T42

                                        ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        ;       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.
                                  ;//////////////////////////////////////////////////////////////////////////////

 002250                                BGNHW   DFPTBL
 002250  000013                                                                  .WORD   L10000-L$HW/2
 002252                                                                  L$HW::
 002252                                                                  DFPTBL::

 002252  000007                        .WORD        7          ;MICROPROCESSOR TYPE = M8207
 002254  160170                        .WORD   160170          ;M8207 CSR UNIBUS ADDRESS
 002256  000300                        .WORD      300          ;M8207 INTERRUPT VECTOR
 002260  005000                        .WORD     5000          ;M8207 INTERRUPT PRIORITY LEVEL = 5
 002262  000003                        .WORD        3          ;LINE UNIT = M8203
 002264  000000                        .WORD      000          ;M8203 REG 11 (E121 SW10,9 , E134 SW9,10)
 002266  000000                        .WORD      000          ;M8203 REG 15 (E134 SW1-8)
 002270  000000                        .WORD      000          ;M8203 REG 16 (E121 SW1-8)
 002272  000000                        .WORD        0          ;H3254&H3255 USED
 002274  000004                        .WORD        4          ;BAUD RATE = 56 K
 002276  000001                        .WORD        1          ;M8207 RUN SWITCH (E28 SW7) IS ON

 002300                                ENDHW
 002300                                                                  L10000:
```

```
                                        .SBTTL  SOFTWARE P-TABLE

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

  002300                                BGNSW   SFPTBL
  002300  000000                                                                  .WORD   L10001-L$SW/2
  002302                                                                  L$SW::
  002302                                                                  SFPTBL::


  002302                                ENDSW
  002302                                                                  L10001:
```

```
                                        .SBTTL  GLOBAL EQUATES SECTION

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

   002302                                       EQUALS
                                        ;
                                        ; BIT DIFINITIONS
                                        ;
           100000                       BIT15== 100000
           040000                       BIT14== 40000
           020000                       BIT13== 20000
           010000                       BIT12== 10000
           004000                       BIT11== 4000
           002000                       BIT10== 2000
           001000                       BIT09== 1000
           000400                       BIT08== 400
           000200                       BIT07== 200
           000100                       BIT06== 100
           000040                       BIT05== 40
           000020                       BIT04== 20
           000010                       BIT03== 10
           000004                       BIT02== 4
           000002                       BIT01== 2
           000001                       BIT00== 1
                                        ;
           001000                       BIT9==  BIT09
           000400                       BIT8==  BIT08
           000200                       BIT7==  BIT07
           000100                       BIT6==  BIT06
           000040                       BIT5==  BIT05
           000020                       BIT4==  BIT04
           000010                       BIT3==  BIT03
           000004                       BIT2==  BIT02
           000002                       BIT1==  BIT01
           000001                       BIT0==  BIT00
                                        ;
                                        ; EVENT FLAG DEFINITIONS
                                        ;    EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                        ;
           000040                       EF.START==      32.                     ; START COMMAND WAS ISSUED
           000037                       EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
           000036                       EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
```

```
        000035                 EF.NEW==        29.                       ; A NEW PASS HAS BEEN STARTED
        000034                 EF.PWR==        28.                       ; A POWER-FAIL/POWER-UP OCCURRED
                               ;
                               ;
                               ; PRIORITY LEVEL DEFINITIONS
                               ;
        000340                 PRI07== 340
        000300                 PRI06== 300
        000240                 PRI05== 240
        000200                 PRI04== 200
        000140                 PRI03== 140
        000100                 PRI02== 100
        000040                 PRI01== 40
        000000                 PRI00== 0
                               ;
                               ;OPERATOR FLAG BITS
                               ;
        000004                 EVL==        4
        000010                 LOT==       10
        000020                 ADR==       20
        000040                 IDU==       40
        000100                 ISR==      100
        000200                 UAM==      200
        000400                 BOE==      400
        001000                 PNT==     1000
        002000                 PRI==     2000
        004000                 IXE==     4000
        010000                 IBE==    10000
        020000                 IER==    20000
        040000                 LOE==    40000
        100000                 HOE==   100000




                               ;;*******************************************************************
                               ;* PROGRAM EVENT FLAG DEFINITIONS
                               ;;*******************************************************************






                               ;;***********************************************************************
                               ;* MAINTENANCE REGISTER - BSEL1
                               ;;************************************************************************
        000200                 RUN     = BIT7
        000100                 MCLR    = BIT6
        000020                 STEPLU  = BIT4
        000010                 LULOOP  = BIT3
        000004                 ROMO    = BIT2
        000002                 ROMI    = BIT1
        000001                 STEPMP  = BIT0

                               ;;************************************************************************
                               ;* 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
000010          STRIP   = BIT3
000004          RDALL   = BIT2
000002          IERR    = BIT1
```

```
        000001                  DDCMP   = BIT0

                                ;;*********************************************************************
                                ;* IBUS REG 10 - RECEIVER BUFFER
                                ;;*********************************************************************
        000200                  RX7     = BIT7
        000100                  RX6     = BIT6
        000040                  RX5     = BIT5
        000020                  RX4     = BIT4
        000010                  RX3     = BIT3
        000004                  RX2     = BIT2
        000002                  RX1     = BIT1
        000001                  RX0     = BIT0

                                ;;*********************************************************************
                                ;* IBUS REG 11
                                ;;*********************************************************************
        000200                  OC      = BIT7
        000100                  OACT    = BIT6
        000040                  SW3     = BIT5
        000020                  ORDY    = BIT4
        000010                  SW2     = BIT3
        000004                  SW1     = BIT2
        000002                  SW0     = BIT1
        000001                  UNRR    = BIT0

                                ;;*********************************************************************
                                ;* IBUS REG 12
                                ;;*********************************************************************
        000200                  IC      = BIT7
        000100                  IACT    = BIT6
        000040                  LULP    = BIT5
        000020                  IRDY    = BIT4
        000010                  OVRR    = BIT3
        000004                  RAB     = BIT2
        000002                  EBLK    = BIT1
        000001                  BCC     = BIT0

                                ;;*********************************************************************
                                ;* IBUS REG 13
                                ;;*********************************************************************
        000200                  RING    = BIT7
        000100                  DTR     = BIT6
        000040                  RTS     = BIT5
        000020                  HDX     = BIT4
        000010                  MODR    = BIT3
        000004                  CS      = BIT2
        000002                  STBY    = BIT1
        000001                  CARR    = BIT0

                                ;;*********************************************************************
                                ;* IBUS REG 14
                                ;;*********************************************************************
        000200                  READY   = BIT7
        000100                  TXEN    = BIT6
        000040                  DISSI   = BIT5
        000020                  RDAX    = BIT4
```

```
        000010          WAX     = BIT3
        000004          ENAX    = BIT2
        000002          AX2     = BIT1
        000001          AX1     = BIT0

                        ;;***********************************************************************
                        ;* IBUS REG 17
                        ;;***********************************************************************
        000200          SIGR    = BIT7
        000100          SIGQ    = BIT6
        000040          TXDATA  = BIT5
        000020          CCOR    = BIT4
        000010          ICIR    = BIT3
        000004          TESTMD  = BIT2
        000002          MCLK    = BIT1
        000001          DDCMP   = BIT0

                        ;;***********************************************************************
                        ;* AX0-15 - USYRT REG 0 (READ ONLY)
                        ;;***********************************************************************
        000200          RX7     = BIT7
        000100          RX6     = BIT6
        000040          RX5     = BIT5
        000020          RX4     = BIT4
        000010          RX3     = BIT3
        000004          RX2     = BIT2
        000002          RX1     = BIT1
        000001          RX0     = BIT0

                        ;;***********************************************************************
                        ;* AX0-16 - USYRT REG 1 (READ ONLY)
                        ;;***********************************************************************
        000200          RERR    = BIT7
        000100          ASBC2   = BIT6
        000040          ASBC1   = BIT5
        000020          ASBC0   = BIT4
        000010          ROR     = BIT3
        000004          RABT    = BIT2
        000002          REOM    = BIT1
        000001          RSOM    = BIT0

                        ;;***********************************************************************
                        ;* AX1-15 - USYRT REG 2
                        ;;***********************************************************************
        000200          TX7     = BIT7
        000100          TX6     = BIT6
        000040          TX5     = BIT5
        000020          TX4     = BIT4
        000010          TX3     = BIT3
        000004          TX2     = BIT2
        000002          TX1     = BIT1
        000001          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
        000040          C32BCC  = BIT5
        000020          V35     = BIT4
        000010          INTGRL  = BIT3
        000004          C32ENB  = BIT2
        000002          OP      = BIT1
        000001          TEST    = BIT0
        000372          AX315U  = I422!XYZ!C32BCC!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
```

```
                                ;*****************************************************************************
                                ;* TX CONTROL BITS DEFINED ON WORD BASIS
                                ;*****************************************************************************
        004000                  TXGOA   = BIT11
        002000                  TXABT   = BIT10
        001000                  TXEOM   = BIT9
        000400                  TXSOM   = BIT8





                                ;******************************************************************
                                ;* RCV CONTROL BITS DEFINED ON WORD BASIS
                                ;******************************************************************
        004000                  RXOVR   = BIT11
        002000                  RXABT   = BIT10
        001000                  RXEBL   = BIT9
        000400                  RXBCC   = BIT8




                                ;*****************************************************************************
                                ;* ADDRESS EQUATES FOR REGISTER STORAGE TABLE (LUREG:)
                                ;*****************************************************************************
        002302                  LUR10   = LUREG+0       ;LINE UNIT IBUS REG 10
        002304                  LUR11   = LUREG+2       ;LINE UNIT IBUS REG 11
        002306                  LUR12   = LUREG+4       ;LINE UNIT IBUS REG 12
        002310                  LUR13   = LUREG+6       ;LINE UNIT IBUS REG 13
        002312                  LUR14   = LUREG+10      ;LINE UNIT IBUS REG 14
        002314                  LUR15   = LUREG+12      ;LINE UNIT IBUS REG 15
        002316                  LUR16   = LUREG+14      ;LINE UNIT IBUS REG 16
        002320                  LUR17   = LUREG+16      ;LINE UNIT IBUS REG 17
        002322                  AX0.15  = LUREG+20      ;USYRT REG 0
        002324                  AX0.16  = LUREG+22      ;USYRT REG 1
        002326                  AX1.15  = LUREG+24      ;USYRT REG 2
        002330                  AX1.16  = LUREG+26      ;USYRT REG 3
        002332                  AX2.15  = LUREG+30      ;USYRT REG 4
        002334                  AX2.16  = LUREG+32      ;USYRT REG 5
        002336                  AX3.15  = LUREG+34      ;USYRT REG 6
        002340                  AX3.16  = LUREG+36      ;USYRT REG 7





        100000                  CHPCHK  = BIT15

        100000                  BCCCHK  = BIT15
        100000                  CRCCHK  = BIT15






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

```
                        ;* MICROINSTRUCTION DEFINITIONS
                        ;*********************************************************************************
        021000          MVIOX   = 021000                ;MOVE IBUS TO OBUS*
        122000          MVIXO   = 122000                ;MOVE IBUS* TO OBUS
        121000          MVIXOX  = 121000                ;MOVE IBUS* TO OBUS*



                        ;***** ERROR1 BIT FLAG DEFINITIONS *****
        000001          RRDYTO  = BIT0
        000002          WRDYTO  = BIT1
```

```
                            .SBTTL  GLOBAL DATA SECTION

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

                            ;;*************************************************************************
                            ;* STORAGE FOR DEVICE REGISTERS
                            ;;*************************************************************************
002302                      LUREG:  .BLKW   16.

                            ;*************************************************************************
                            ;* MISCELLANEOUS STORAGE
                            ;*************************************************************************
002342  000000              SCRACH: .WORD   0               ;GEN'L PURPOSE SCRATCH WORD
002344  000000              LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
002346  000000              PSTACK: .WORD   0               ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
002350  000000              PRIOR:  .WORD   0               ;CPU PRIORITY FOR PRINTOUT
002352  000000              SUBRPC: .WORD   0               ;PC OF SUBR CALL FOR ERROR REPORTS
002354  000000              INTFLG: .WORD   0               ;INTERRUPT RECEIVED FLAGS
                                                            ;  BIT 0 FOR TX, BIT 1 FOR RCV
002356  000000              ERRFLG: .WORD   0               ;SUBROUTINE ERROR FLAG
002360  000000              TIMFLG: .WORD   0               ;EVENT TIME-OUT FLAG
002362  000000              RETADR: .WORD   0               ;SUBR ERROR RETURN ADDRESS
002364  000000              REDBYT: .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM LU REG
002366  000000              WRIBYT: .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
002370  000000              RAX15:  .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM REG 15
002372  000000              RAX16:  .WORD   0               ;LO BYTE CONTAINS BYTE READ FROM REG 16
002374  000000              WAX15:  .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 15
002376  000000              WAX16:  .WORD   0               ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 16
002400  000000              REGNUM: .WORD   0               ;NUMBER (10-17) OF LINE UNIT REG BEING TESTED
002402  000000              AXNUM:  .WORD   0               ;NUMBER (0-7) OF EXTENDED REG BYTE BEING TESTED
002404  000000              GOODAT: .WORD   0               ;STORAGE FOR EXPECTED DATA
002406  000000              BADDAT: .WORD   0               ;STORAGE FOR ACTUAL DATA
002410  000000              LOADAT: .WORD   0               ;CONTAINS TEST DATA LOADED INTO REG
002412  000000              FRSTIM: .WORD   0               ;FLAG=0 IF PROGRAM JUST LOADED
002414  000000              SAVE4:  .WORD   0               ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
002416  000000              SAVE6:  .WORD   0               ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
002420  000000              ERROR1: .WORD   0               ;SUBR ERROR BIT FLAGS (DEF'D IN GLOBAL EQUATES)
002422  000000              TXWORD: .WORD   0               ;BITS 0-11 CONTAIN DATA TO LOAD INTO TX SILO
002424  000000              RXWORD: .WORD   0               ;BITS 0-11 CONTAIN DATA READ FROM RCV SILO
002426  000000              DISILO: .WORD   0               ;CONTAINS CURRENT STATE OF DISSI IN BIT 5
002430  000000              CHPTYP: .WORD   0               ;USYRT CHIP TYPE, =0 FOR SIG, ELSE =1
002432  000000              SAVLEN: .WORD   0               ;SAVED TX AND RCV CHAR LENGTHS
002434  000000              DEVMAP: .WORD   0               ;BIT MAP OF ACTIVE DEVICES
002436  000000              DEVPTR: .WORD   0               ;DEVICE MAP BIT POINTER
002440  000000              UNIT:   .WORD   0               ;CONTAINS UNIT NO. (1 TO N)
002442  000000              TSTNUM: .WORD   0               ;CONTAINS TEST NUMBER FOR SOME TESTS
002444  000000              STARES: .WORD   0               ;FLAG=0 IF FIRST PASS AFTER STA OR RES

                            ;;*********  CURRENT DEVICE PARAMETERS  *************
002446  160170              MPCSR:  .WORD   160170          ;POINTER TO MICROPROCESSOR CSR'S
002450  160171              BSEL1:  .WORD   160171          ;POINTER TO BSEL1
002452  160172              BSEL2:  .WORD   160172          ;POINTER TO BSEL2
002454                      BSEL4:
002454  160174              SEL4:   .WORD   160174          ;POINTER TO SEL4
```

```
002456  160176          SEL6:   .WORD   160176          ;POINTER TO SEL6
002460  000300          MPIVEC: .WORD      300          ;MICROPROCESSOR INPUT INTERRUPT VECTOR
002462  000304          MPOVEC: .WORD      304          ;MICROPROCESSOR OUTPUT INTERRUPT VECTOR
002464  000240          MPRIOR: .WORD      240          ;MICROPROCESSOR DEVICE PRIORITY
002466  000000          LUSWI1: .WORD        0          ;LINE UNIT SWITCH PACK #1
002470  000000          LUSWI2: .WORD        0          ;LINE UNIT SWITCH PACK #2
002472  000000          LUSWI3: .WORD        0          ;LINE UNIT SWITCH PACK #3
002474  000000          TSTCON: .WORD        0          ;TEST CONNECTOR INDICATOR
002476  000000          RUNINH: .WORD        0          ;RUN SWITCH INDICATOR


                        ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
002500     000          REDDAT: .BYTE   0
002501     000                  .BYTE   0
002502     000                  .BYTE   0
002503     000                  .BYTE   0
002504     000                  .BYTE   0
002505     000                  .BYTE   0
002506     000                  .BYTE   0
002507     000                  .BYTE   0


                        ;;****  GEN'L PURPOSE SCRATCH STORAGE  *****
002510  000000          REG0:   .WORD   0
002512  000000          REG1:   .WORD   0
002514  000000          REG2:   .WORD   0
002516  000000          REG3:   .WORD   0
002520  000000          REG4:   .WORD   0
002522  000000          REG5:   .WORD   0
002524  000000          REG6:   .WORD   0
002526  000000          REG7:   .WORD   0


                        ;;****  SCRATCH STORAGE FOR MESSAGE REPORTING   *****
002530  000000          TMP0:   .WORD   0
002532  000000          TMP1:   .WORD   0
002534  000000          TMP2:   .WORD   0
002536  000000          TMP3:   .WORD   0
002540  000000          TMP4:   .WORD   0
002542  000000          TMP5:   .WORD   0
002544  000000          TMP6:   .WORD   0
002546  000000          TMP7:   .WORD   0


                        ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
002550                  UPBITS:
002550     000                  .BYTE   000             ;MASK FOR REG 10
002551     056                  .BYTE   056             ;MASK FOR REG 11
002552     000                  .BYTE   000             ;MASK FOR REG 12
002553     257                  .BYTE   257             ;MASK FOR REG 13
002554     100                  .BYTE   100             ;MASK FOR REG 14
002555     377                  .BYTE   377             ;MASK FOR REG 15
002556     377                  .BYTE   377             ;MASK FOR REG 16
002557     306                  .BYTE   306             ;MASK FOR REG 17

002560     200          R14NRW: .BYTE   200             ;REG 14 NON-R/W BITS


                        ;***** MASKS FOR EXTENDED REGISTER NON-READ/WRITE BITS *****
002561                  ANBITS:
002561     377                  .BYTE   377             ;MASK FOR AX0-15
002562     377                  .BYTE   377             ;MASK FOR AX0-16
```

```
002563    000                .BYTE   000             ;MASK FOR AX1-15
002564    360                .BYTE   360             ;MASK FOR AX1-16
002565    000                .BYTE   000             ;MASK FOR AX2-15
002566    000                .BYTE   000             ;MASK FOR AX2-16
002567    004                .BYTE   004             ;MASK FOR AX3-15
002570    030                .BYTE   030             ;MASK FOR AX3-16

                     ;***** DATA PATTERN A *****
002571               PATA:
002571    125                .BYTE   125
002572    252                .BYTE   252
002573    000                .BYTE   000
002574    377                .BYTE   377
002575    001                .BYTE   001
002576    002                .BYTE   002
002577    004                .BYTE   004
002600    010                .BYTE   010
002601    020                .BYTE   020
002602    040                .BYTE   040
002603    100                .BYTE   100
002604    200                .BYTE   200
002605    376                .BYTE   376
002606    375                .BYTE   375
002607    373                .BYTE   373
002610    367                .BYTE   367
002611    357                .BYTE   357
002612    337                .BYTE   337
002613    277                .BYTE   277
002614    177                .BYTE   177

                     ;***** DATA PATTERN B *****
002615               PATB:
002615    000                .BYTE   000
002616    000                .BYTE   000
002617    040                .BYTE   040
002620    100                .BYTE   100
002621    220                .BYTE   220
002622    000                .BYTE   000
002623    000                .BYTE   000
002624    051                .BYTE   051

                     ;***** DATA PATTERN C *****
002625               PATC:
002625    020                .BYTE   020
002626    020                .BYTE   020
002627    020                .BYTE   020

                     ;***** DATA PATTERN D *****
002630               PATD:
002630    000                .BYTE   000
002631    040                .BYTE   040
002632    000                .BYTE   000

                     ;***** DATA PATTERN E *****
002633               PATE:
002633    000                .BYTE   000
002634    120                .BYTE   120
```

```
002635  020                  .BYTE   020
002636  100                  .BYTE   100
002637  120                  .BYTE   120
002640  000                  .BYTE   000

                     ;***** DATA PATTERN F *****
002641               PATF:
002641  050                  .BYTE   050
002642  051                  .BYTE   051
002643  050                  .BYTE   050

                     ;***** DATA PATTERN G *****
002644               PATG:
002644  000                  .BYTE   000
002645  000                  .BYTE   000
002646  240                  .BYTE   240
002647  120                  .BYTE   120
002650  177                  .BYTE   177
002651  000                  .BYTE   000
002652  000                  .BYTE   000
002653  001                  .BYTE   001

                     ;***** DATA PATTERN H *****
002654               PATH:
002654  000                  .BYTE   000
002655  000                  .BYTE   000
002656  377                  .BYTE   377
002657  017                  .BYTE   017
002660  377                  .BYTE   377
002661  377                  .BYTE   377
002662  375                  .BYTE   375
002663  377                  .BYTE   377

                     ;***** DATA PATTERN I *****
002664               PATI:
002664  000                  .BYTE   000
002665  000                  .BYTE   000
002666  000                  .BYTE   000
002667  000                  .BYTE   000
002670  000                  .BYTE   000
002671  103                  .BYTE   103
002672  000                  .BYTE   000
002673  000                  .BYTE   000

                     ;***** DATA PATTERN J *****
002674               PATJ:
002674  000                  .BYTE   000
002675  000                  .BYTE   000
002676  010                  .BYTE   010
002677  002                  .BYTE   002
002700  004                  .BYTE   004
002701  103                  .BYTE   103
002702  001                  .BYTE   001
002703  100                  .BYTE   100

                     ;***** DATA PATTERN K *****
002704  000          PATK:   .BYTE   000
```

```
002705   000          .BYTE   000
002706   377          .BYTE   377
002707   377          .BYTE   377
002710   125          .BYTE   125
002711   125          .BYTE   125
002712   252          .BYTE   252
002713   252          .BYTE   252
002714   001          .BYTE   001
002715   000          .BYTE   000
002716   002          .BYTE   002
002717   000          .BYTE   000
002720   004          .BYTE   004
002721   000          .BYTE   000
002722   010          .BYTE   010
002723   000          .BYTE   000
002724   020          .BYTE   020
002725   000          .BYTE   000
002726   040          .BYTE   040
002727   000          .BYTE   000
002730   100          .BYTE   100
002731   000          .BYTE   000
002732   200          .BYTE   200
002733   000          .BYTE   000
002734   000          .BYTE   000
002735   001          .BYTE   001
002736   000          .BYTE   000
002737   002          .BYTE   002
002740   000          .BYTE   000
002741   004          .BYTE   004
002742   000          .BYTE   000
002743   010          .BYTE   010
002744   000          .BYTE   000
002745   020          .BYTE   020
002746   000          .BYTE   000
002747   040          .BYTE   040
002750   000          .BYTE   000
002751   100          .BYTE   100
002752   000          .BYTE   000
002753   200          .BYTE   200
002754   376          .BYTE   376
002755   377          .BYTE   377
002756   375          .BYTE   375
002757   377          .BYTE   377
002760   373          .BYTE   373
002761   377          .BYTE   377
002762   367          .BYTE   367
002763   377          .BYTE   377
002764   357          .BYTE   357
002765   377          .BYTE   377
002766   337          .BYTE   337
002767   377          .BYTE   377
002770   277          .BYTE   277
002771   377          .BYTE   377
002772   177          .BYTE   177
002773   377          .BYTE   377
002774   377          .BYTE   377
002775   376          .BYTE   376
```

```
002776     377                .BYTE   377
002777     375                .BYTE   375
003000     377                .BYTE   377
003001     373                .BYTE   373
003002     377                .BYTE   377
003003     367                .BYTE   367
003004     377                .BYTE   377
003005     357                .BYTE   357
003006     377                .BYTE   377
003007     337                .BYTE   337
003010     377                .BYTE   377
003011     277                .BYTE   277
003012     377                .BYTE   377
003013     177                .BYTE   177

                      ;***** DATA PATTERN L *****
003014                PATL:
003014     000                .BYTE   000
003015     000                .BYTE   000
003016     377                .BYTE   377
003017     377                .BYTE   377
003020     000                .BYTE   000
003021     000                .BYTE   000

                      ;***** DATA PATTERN M *****
003022                PATM:
003022     000                .BYTE   000
003023     020                .BYTE   020
003024     000                .BYTE   000
003025     000                .BYTE   000
003026     200                .BYTE   200
003027     000                .BYTE   000
003030     000                .BYTE   000
003031     051                .BYTE   051

                      ;***** DATA PATTERN N *****
003032                PATN:
003032     000                .BYTE   000
003033     000                .BYTE   000
003034     000                .BYTE   000
003035     125                .BYTE   125
003036     000                .BYTE   000
003037     252                .BYTE   252
003040     000                .BYTE   000
003041     377                .BYTE   377
003042     005                .BYTE   005
003043     000                .BYTE   000
003044     012                .BYTE   012
003045     000                .BYTE   000
003046     017                .BYTE   017
003047     000                .BYTE   000

                      ;***** DATA PATTERN O *****
003050                PATO:
003050     000                .BYTE   000
003051     041                .BYTE   041
003052     004                .BYTE   004
```

```
003053  010               .BYTE  010
003054  020               .BYTE  020
003055  040               .BYTE  040
003056  100               .BYTE  100
003057  101               .BYTE  101
003060  200               .BYTE  200
003061  201               .BYTE  201
003062  300               .BYTE  300
003063  111               .BYTE  111
003064  301               .BYTE  301
003065  375               .BYTE  375

                  ;***** DATA PATTERN P *****
003066            PATP:
003066  000               .BYTE  000
003067  113               .BYTE  113
003070  200               .BYTE  200
003071  040               .BYTE  040
003072  020               .BYTE  020
003073  010               .BYTE  010
003074  001               .BYTE  001
003075  104               .BYTE  104
003076  007               .BYTE  007
003077  105               .BYTE  105
003100  007               .BYTE  007
003101  144               .BYTE  144
003102  107               .BYTE  107
003103  157               .BYTE  157

                  ;***** DATA PATTERN U *****
003104  000       PATU:   .BYTE  000
003105  000               .BYTE  000
003106  001               .BYTE  001
003107  000               .BYTE  000
003110  013               .BYTE  013
003111  000               .BYTE  000
003112  011               .BYTE  011
003113  000               .BYTE  000
003114  021               .BYTE  021
003115  000               .BYTE  000
003116  101               .BYTE  101
003117  000               .BYTE  000
003120  301               .BYTE  301
003121  000               .BYTE  000
003122  000               .BYTE  000
003123  001               .BYTE  001
003124  000               .BYTE  000
003125  002               .BYTE  002
003126  000               .BYTE  000
003127  004               .BYTE  004
003130  000               .BYTE  000
003131  040               .BYTE  040
003132  000               .BYTE  000
003133  100               .BYTE  100
003134  000               .BYTE  000
003135  200               .BYTE  200
003136  000               .BYTE  000
```

```
003137	346			.BYTE	346
003140	000			.BYTE	000
003141	345			.BYTE	345
003142	000			.BYTE	000
003143	343			.BYTE	343
003144	000			.BYTE	000
003145	307			.BYTE	307
003146	000			.BYTE	000
003147	247			.BYTE	247
003150	000			.BYTE	000
003151	147			.BYTE	147

				;***** PATTERN V *****
003152	000		PATV:	.BYTE	000
003153	000			.BYTE	000
003154	333			.BYTE	333
003155	000			.BYTE	000
003156	331			.BYTE	331
003157	000			.BYTE	000
003160	323			.BYTE	323
003161	000			.BYTE	000
003162	313			.BYTE	313
003163	000			.BYTE	000
003164	233			.BYTE	233
003165	000			.BYTE	000
003166	133			.BYTE	133
003167	000			.BYTE	000
003170	000			.BYTE	000
003171	001			.BYTE	001
003172	000			.BYTE	000
003173	002			.BYTE	002
003174	000			.BYTE	000
003175	004			.BYTE	004
003176	000			.BYTE	000
003177	040			.BYTE	040
003200	000			.BYTE	000
003201	100			.BYTE	100
003202	000			.BYTE	000
003203	200			.BYTE	200
003204	000			.BYTE	000
003205	346			.BYTE	346
003206	000			.BYTE	000
003207	345			.BYTE	345
003210	000			.BYTE	000
003211	343			.BYTE	343
003212	000			.BYTE	000
003213	307			.BYTE	307
003214	000			.BYTE	000
003215	247			.BYTE	247
003216	000			.BYTE	000
003217	147			.BYTE	147

003220			ENDPAT:
			.EVEN
```

```


                                ;***  TEST MESSAGES TO BE TRANSMITTED  ***

 003220  000400                 MSG1:   TXSOM
 003222  000400                         TXSOM
 003224  000000                         000
 003226  000125                         125
 003230  000252                         252
 003232  000377                         377
 003234  000000                         000
 003236  001000                         TXEOM
 003240  001000                         TXEOM
 003242  001000                         TXEOM
 003244  001000                         TXEOM

 003246  000377                 MSG4:   377
 003250  000000                         000
 003252  000125                         125
 003254  000252                         252
 003256  001000                         TXEOM
 003260  001000                         TXEOM






                                ;*** RECEIVED DATA BUFFER (64. WORDS) ***
 003262                         RCVBUF: .BLKW   64.
```

```
                                         .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
                                         ;;*********************************************************************
 003462                                         DEVTYP  <M8203>
 003462                                                                             L$DVTYP::
 003462      115      070      062                                                         .ASCIZ  /M8203/
 003465      060      063      000
                                                                                           .EVEN

                                         ;;**********************************************************************
                                         ;* TITLE OF PROGRAM
                                         ;;**********************************************************************
 003470                                         DESCRIPT        <M8203 STATIC LOGIC TESTS - PART 1 OF 2>
 003470                                                                             L$DESC::
 003470      115      070      062                                                         .ASCIZ  /M8203 STATIC LOGIC
 003473      060      063      040
 003476      123      124      101
 003501      124      111      103
 003504      040      114      117
 003507      107      111      103
 003512      040      124      105
 003515      123      124      123
 003520      040      055      040
 003523      120      101      122
 003526      124      040      061
 003531      040      117      106
 003534      040      062      000
                                                                                           .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.
                                                ;*************************************************************************
003540                                          STPCLK:
003540  152777  000006  176702                          BISB    #ROMO!ROMI,@BSEL1          ;SET ROMO, ROMI BITS IN BSEL1
003546  017677  000000  176702                          MOV     @(SP),@SEL6       ;PUT INSTRUCTION INTO SEL6
003554  152777  000007  176666                          BISB    #ROMO!ROMI!STEPMP,@BSEL1 ;SET ROMO, ROMI, STEPMP IN BSEL1
003562  142777  000007  176660                          BICB    #ROMO!ROMI!STEPMP,@BSEL1 ;CLEAR ROMO, ROMI, STEPMP IN BSEL1
003570  062716  000002                                  ADD     #2,(SP)           ;FIX UP RETURN PC
003574  000207                                          RTS     PC                ;RETURN

                                                ;*************************************************************************
                                                ;* MSTCLR - THIS SUBROUTINE ISSUES A MASTER CLEAR AND SETS LULOOP
                                                ;*************************************************************************
003576                                          MSTCLR:
003576  010146                                          MOV     R1,-(SP)          ;SAVE R1
003600  013746  002400                                  MOV     REGNUM,-(SP)      ;SAVE LU REG NO.
003604  112777  000100  176636                          MOVB    #MCLR,@BSEL1      ;SET MASTER CLEAR BIT
003612  142777  000300  176630                          BICB    #RUN!MCLR,@BSEL1  ;CLEAR RUN AND MCLR BITS
003620  012701  000024                                  MOV     #20.,R1           ;INITIALIZE STALL COUNTER
003624  000240                                  2$:     NOP                       ;STALL IN LOOP FOR SEVERAL MICRO-SEC
003626  005301                                          DEC     R1
003630  001375                                          BNE     2$
003632  152777  000010  176610                          BISB    #LULOOP,@BSEL1    ;SET LU LOOP
003640  012737  000013  002400                          MOV     #13,REGNUM        ;SET LU REG NO. = 13
003646  005037  002366                                  CLR     WRIBYT
003652  004737  003750                                  JSR     PC,WRITLU         ;CLEAR REG 13
003656  012637  002400                                  MOV     (SP)+,REGNUM      ;RESTORE LU REG NO.
003662  012601                                          MOV     (SP)+,R1          ;RESTORE R1
003664  005037  002432                                  CLR     SAVLEN            ;CLEAR CHAR LENGTH FROM SETUP
003670  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.
                                                ;*************************************************************************
003672                                          READLU:
```

```
003672  010146                              MOV     R1,-(SP)        ;SAVE R1
003674  013701  002400                      MOV     REGNUM,R1       ;GET LINE UNIT REG NUMBER
003700  006301                              ASL     R1              ;SHIFT INTO SOURCE BITS 4-7
003702  006301                              ASL     R1
003704  006301                              ASL     R1
003706  006301                              ASL     R1
003710  052701  000004                      BIS     #4,R1           ;SET DESTINATION = BSEL4
003714  052701  021000                      BIS     #MVIOX,R1       ;SET REST OF MOVE INSTRUCTION
003720  010137  003730                      MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
003724  004737  003540                      JSR     PC,STPCLK       ;EXECUTE MOVE INSTRUCTION
003730  000000                      2$:     .WORD   0               ;INSTRUCTION GOES HERE
003732  117737  176516  002364              MOVB    @BSEL4,REDBYT   ;GET LU REG CONTENTS INTO REDBYT
003740  105037  002365                      CLRB    REDBYT+1        ;CLR HI BYTE OF STORAGE
003744  012601                              MOV     (SP)+,R1        ;RESTORE R1
003746  000207                              RTS     PC              ;RETURN

                                    ;*******************************************************************
                                    ;* WRITLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
                                    ;*        EXECUTE AN INSTRUCTION WHICH LOADS THE BYTE CONTAINED IN WRIBYT
                                    ;*        INTO THE LU REG WHOSE NUMBER IS PASSED IN REGNUM,
                                    ;*******************************************************************
003750                              WRITLU:
003750  010146                              MOV     R1,-(SP)        ;SAVE R1
003752  013701  002400                      MOV     REGNUM,R1       ;GET LINE UNIT REG NUMBER
003756  052701  000100                      BIS     #100,R1         ;SET SOURCE = BSEL4
003762  052701  122000                      BIS     #MVIXO,R1       ;SET REST OF MOVE INSTRUCTION
003766  010137  004010                      MOV     R1,2$           ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
003772  105037  002367                      CLRB    WRIBYT+1        ;CLR HI BYTE OF STORAGE
003776  113777  002366  176450              MOVB    WRIBYT,@BSEL4   ;LOAD BYTE INTO BSEL4
004004  004737  003540                      JSR     PC,STPCLK       ;EXECUTE MOVE INSTRUCTION
004010  000000                      2$:     .WORD   0
004012  012601                              MOV     (SP)+,R1        ;RESTORE R1
004014  000207                              RTS     PC              ;RETURN

                                    ;*******************************************************************
                                    ;* GETREG - THIS SUBROUTINE READS THE LINE UNIT REGISTERS 10-17 INTO THE
                                    ;*        REGISTER STORAGE TABLE (LUREG:).
                                    ;*******************************************************************
004016  010146                      GETREG: MOV     R1,-(SP)        ;SAVE R1
004020  013746  002400                      MOV     REGNUM,-(SP)    ;SAVE CURRENT REG NO.
004024  012701  002302                      MOV     #LUR10,R1       ;INIT POINTER TO REG STORAGE TABLE
004030  012737  000010  002400              MOV     #10,REGNUM      ;INIT LU REG NO. TO 10
004036  004737  003672              3$:     JSR     PC,READLU       ;READ A LINE UNIT REG
004042  113721  002364                      MOVB    REDBYT,(R1)+    ;PUT BYTE READ INTO TABLE
004046  105021                              CLRB    (R1)+           ;CLEAR UPPER BYTE OF TABLE ENTRY
004050  005237  002400                      INC     REGNUM          ;INCREMENT REG NO.
004054  023727  002400  000020              CMP     REGNUM,#20      ;SEE IF ALL REGS READ YET
004062  002765                              BLT     3$              ;BR IF NOT
004064  012637  002400                      MOV     (SP)+,REGNUM    ;RESTORE CURRENT REG NO.
```

```
004070  012601                              MOV     (SP)+,R1            ;RESTORE R1
004072  000207                              RTS     PC                  ;RETURN

                                    ;*******************************************************************************
                                    ;* LOOPIN - THIS SUBROUTINE PLACES THE MICROPROCESSOR IN A LOOP ON AN
                                    ;*       INSTRUCTION, BY MOVING THE INSTRUCTION FROM THE WORD FOLLOWING THE CALL
                                    ;*       INTO SEL6, AND SETTING RUN AND ROMI IN BSEL1. THE SUBROUTINE RETURNS
                                    ;*       WITH THE MICROPROCESSOR STUCK IN THE LOOP, AND IF IT IS DESIRED TO
                                    ;*       TERMINATE THE LOOP, THE PDP-11 PROGRAM MUST CLEAR THE RUN BIT IN
                                    ;*       BSEL1, OR CALL SUBROUTINE MSTCLR TO DO THIS.
                                    ;*******************************************************************************
004074                              LOOPIN:
004074  152777  000006  176346              BISB    #ROMO!ROMI,@BSEL1       ;SET ROMO, ROMI BITS IN BSEL1
004102  017677  000000  176346              MOV     @(SP),@SEL6         ;PUT MICROINSTRUCTION INTO SEL6
004110  152777  000206  176332              BISB    #RUN!ROMO!ROMI,@BSEL1 ;SET RUN, ROMO, ROMI IN BSEL1
004116  062716  000002                      ADD     #2,(SP)             ;FIX UP RETURN PC
004122  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.
                                    ;*******************************************************************************
004124  010146                      READAX: MOV     R1,-(SP)            ;SAVE R1
004126  013746  002400                      MOV     REGNUM,-(SP)        ;STORE CURRENT REG NO.
004132  042737  000001  002420              BIC     #RRDYTO,ERROR1      ;CLEAR ERROR BIT
004140  012737  000014  002400              MOV     #14,REGNUM          ;SET LU REG NO. = 14
004146  113737  002402  002366              MOVB    AXNUM,WRIBYT        ;SET UP AX REG NO. BITS
004154  006237  002366                      ASR     WRIBYT
004160  152737  000024  002366              BISB    #RDAX!ENAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
004166  053737  002426  002366              BIS     DISILO,WRIBYT       ;SET PROPER STATE OF DISSI BIT
004174  004737  003750                      JSR     PC,WRITLU           ;SET RDAX AND ENAX IN REG 14
004200  005001                              CLR     R1                  ;INIT TIMER
004202  004737  003672              6$:     JSR     PC,READLU           ;READ REG 14
004206  132737  000200  002364              BITB    #READY,REDBYT       ;SEE IF READY BIT SET IN REG 14 YET
004214  001006                              BNE     9$                  ;BR IF READY SET
004216  005201                              INC     R1                  ;INCR TIMER
004220  001370                              BNE     6$                  ;BR IF TIMER DIDN'T TIME OUT YET
004222  052737  000001  002420              BIS     #RRDYTO,ERROR1      ;SET ERROR FLAG FOR TIME OUT ON READ RDY
004230  000424                              BR      12$                 ;BR TO RETURN
004232  012737  000015  002400      9$:     MOV     #15,REGNUM          ;SET REG NO. = 15
004240  004737  003672                      JSR     PC,READLU           ;READ REG 15
004244  113737  002364  002370              MOVB    REDBYT,RAX15        ;STORE REG AX-15
004252  105037  002371                      CLRB    RAX15+1             ;CLR HI BYTE OF STORAGE
004256  012737  000016  002400              MOV     #16,REGNUM          ;SET REG NO. = 16
004264  004737  003672                      JSR     PC,READLU           ;READ REG 16
004270  113737  002364  002372              MOVB    REDBYT,RAX16        ;STORE REG AX-16
```

```
004276  105037  002373                  CLRB    RAX16+1             ;CLR HI BYTE OF STORAGE
004302  012637  002400          12$:    MOV     (SP)+,REGNUM        ;RESTORE CURRENT REG NO.
004306  012601                          MOV     (SP)+,R1            ;RESTORE R1
004310  000207                          RTS     PC                  ;RETURN

                                ;***********************************************************************
                                ;* WRITAX - THIS SUBROUTINE WRITES THE USYRT REG PAIR WHOSE NUMBER (0-3) IS
                                ;*      PASSED IN BITS 1,2 OF AXNUM ON ENTRY, WITH THE DATA FROM WAX15 AND
                                ;*      WAX16. IF LINE UNIT DOES NOT RESPOND WITH READY IN REG 14, WRDYTO BIT
                                ;*      IS SET IN ERROR1 ON RETURN.
                                ;***********************************************************************
004312  010146                  WRITAX: MOV     R1,-(SP)            ;SAVE R1
004314  013746  002400                  MOV     REGNUM,-(SP)        ;SAVE CURRENT REG NO.
004320  042737  000002  002420          BIC     #WRDYTO,ERROR1      ;CLEAR ERROR BIT
004326  012737  000014  002400          MOV     #14,REGNUM          ;SET LU REG NO. = 14
004334  113737  002402  002366          MOVB    AXNUM,WRIBYT        ;SET AX REG NO. BITS
004342  006237  002366                  ASR     WRIBYT
004346  053737  002426  002366          BIS     DISILO,WRIBYT       ;SET PROPER STATE OF DISSI BIT
004354  004737  003750                  JSR     PC,WRITLU           ;SET AX NO. BITS IN REG 14
004360  012737  000015  002400          MOV     #15,REGNUM          ;SET REG NO. = 15
004366  105037  002375                  CLRB    WAX15+1             ;CLR HI BYTE OF STORAGE
004372  113737  002374  002366          MOVB    WAX15,WRIBYT        ;SET UP BYTE TO WRITE INTO REG 15
004400  004737  003750                  JSR     PC,WRITLU           ;WRITE BYTE INTO REG 15
004404  005237  002400                  INC     REGNUM              ;SET REG NO. = 16
004410  105037  002377                  CLRB    WAX16+1             ;CLR HI BYTE OF STORAGE
004414  113737  002376  002366          MOVB    WAX16,WRIBYT        ;SET UP BYTE TO WRITE INTO REG 16
004422  004737  003750                  JSR     PC,WRITLU           ;WRITE BYTE INTO REG 16
004426  012737  000014  002400          MOV     #14,REGNUM          ;SET REG NO. = 14
004434  113737  002402  002366          MOVB    AXNUM,WRIBYT        ;SET AX REG NO. BITS
004442  006237  002366                  ASR     WRIBYT
004446  152737  000014  002366          BISB    #ENAX!WAX,WRIBYT    ;SET UP BITS TO LOAD INTO REG 14
004454  053737  002426  002366          BIS     DISILO,WRIBYT       ;SET PROPER STATE OF DISSI BIT
004462  004737  003750                  JSR     PC,WRITLU           ;SET ENAX AND WAX IN REG 14
004466  005001                          CLR     R1                  ;INIT PROGRAM TIMER
004470  004737  003672          6$:     JSR     PC,READLU           ;READ REG 14
004474  132737  000200  002364          BITB    #READY,REDBYT       ;SEE IF READY BIT SET IN REG 14 YET
004502  001005                          BNE     9$                  ;BR IF READY SET
004504  005201                          INC     R1                  ;INCR TIMER
004506  001370                          BNE     6$                  ;BR IF TIMER DIDN'T TIME OUT YET
004510  052737  000002  002420          BIS     #WRDYTO,ERROR1      ;SET ERROR FLAG BIT FOR TIME OUT ON WRITE RDY
004516  012637  002400          9$:     MOV     (SP)+,REGNUM        ;RESTORE CURRENT REG NO.
004522  012601                          MOV     (SP)+,R1            ;RESTORE R1
004524  000207                          RTS     PC                  ;RETURN

                                ;***********************************************************************
                                ;* GETALL - THIS SUBROUTINE READS THE LINE UNIT REGS 10-17 AND THE EXTENDED
                                ;*      REGISTERS AX0-AX3 INTO REGISTER STORAGE TABLE (LUREG:).
                                ;***********************************************************************
004526  010146                  GETALL: MOV     R1,-(SP)            ;SAVE R1
004530  013746  002402                  MOV     AXNUM,-(SP)         ;SAVE CURRENT AX REG BYTE NO.
```

```
004534  012737  014435  002530          MOV     #DH5,TMPO          ;SET AX LO BYTE NO.
004542  032737  000001  002402          BIT     #BIT0,AXNUM        ;SEE IF LO OR HI BYTE
004550  001403                          BEQ     1$                 ;BR IF LO BYTE
004552  012737  014440  002530          MOV     #DH6,TMPO          ;SET AX HI BYTE NO.
004560  004737  004016          1$:     JSR     PC,GETREG          ;READ AND STORE REGS 10-17
004564  142777  000010  175656          BICB    #LULOOP,@BSEL1     ;CLEAR LULOOP
004572  012701  002322                  MOV     #AX0.15,R1         ;INIT POINTER TO REG STORAGE TABLE
004576  005037  002402                  CLR     AXNUM              ;INIT AX REG BYTE NO. TO 0
004602  004737  004124          3$:     JSR     PC,READAX          ;READ 2 AX REG BYTES
004606  113721  002370                  MOVB    RAX15,(R1)+        ;PUT LO BYTE READ INTO TABLE
004612  105021                          CLRB    (R1)+              ;CLEAR UPPER BYTE OF TABLE ENTRY
004614  113721  002372                  MOVB    RAX16,(R1)+        ;PUT HI BYTE READ INTO TABLE
004620  105021                          CLRB    (R1)+              ;CLEAR UPPER BYTE OF TABLE ENTRY
004622  062737  000002  002402          ADD     #2,AXNUM           ;INCR AX REG BYTE NO.
004630  023727  002402  000010          CMP     AXNUM,#10          ;SEE IF ALL REGS READ YET
004636  002761                          BLT     3$                 ;BR IF NOT
004640  012637  002402                  MOV     (SP)+,AXNUM        ;RESTORE CURRENT AX REG BYTE NO.
004644  012601                          MOV     (SP)+,R1           ;RESTORE R1
004646  013737  002402  002532          MOV     AXNUM,TMP1
004654  006237  002532                  ASR     TMP1               ;GET EXTENDED REG NO. FOR PRINTOUT
004660  000207                          RTS     PC                 ;RETURN





                                ;*********************************************************************
                                ;* OSIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ORDY (REG 11)
                                ;*       AND OCOR (REG 17) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
                                ;*       AS PASSED IN BIT 0 (ORDY) AND BIT 1 (OCOR) OF THE WORD FOLLOWING THE
                                ;*       CALL.
                                ;*       IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS IN
                                ;*       RETADR.
                                ;*********************************************************************
004662  013746  002400          OSIRDY: MOV     REGNUM,-(SP)       ;SAVE LU REG NO.
004666  013746  002352                  MOV     SUBRPC,-(SP)
004672  005737  002352                  TST     SUBRPC             ;SEE IF THIS IS A NESTED CALL
004676  001006                          BNE     1$                 ;BR IF YES
004700  016637  000004  002352          MOV     4(SP),SUBRPC
004706  162737  000004  002352          SUB     #4,SUBRPC          ;GET PC OF SUBROUTINE CALL
004714  012737  000011  002400  1$:     MOV     #11,REGNUM         ;SET REG NO. TO 11
004722  004737  003672                  JSR     PC,READLU          ;READ REG 11
004726  032776  000001  000004          BIT     #BIT0,@4(SP)       ;GET EXPECTED STATE OF ORDY
004734  001413                          BEQ     3$                 ;BR IF EXPECTED ORDY = 0
004736  132737  000020  002364          BITB    #ORDY,REDBYT       ;SEE IF ORDY = 1
004744  001022                          BNE     9$                 ;BR IF ORDY = 1
004746  004737  004526                  JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                ;REPORT ORDY NOT SET
004752                                  ERRDF   7,EM7,ERR4
004752  104455                                                                  TRAP    C$ERDF
004754  000007                                                                  .WORD   7
004756  012374                                                                  .WORD   EM7
004760  015622                                                                  .WORD   ERR4
004762  000451                          BR      16$                ;TAKE ERROR RETURN
004764  132737  000020  002364  3$:     BITB    #ORDY,REDBYT       ;SEE IF ORDY = 0
004772  001407                          BEQ     9$                 ;BR IF ORDY = 0
004774  004737  004526                  JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
```

```
                                          ;REPORT ORDY NOT CLEARED
     005000                                       ERRDF   8,EM8,ERR4
     005000  104455                                                                        TRAP    C$ERDF
     005002  000010                                                                        .WORD   8
     005004  012411                                                                        .WORD   EM8
     005006  015622                                                                        .WORD   ERR4
     005010  000436                               BR      16$             ;TAKE ERROR RETURN
     005012  012737  000017  002400   9$:         MOV     #17,REGNUM      ;SET REG NO. = 17
     005020  004737  003672                       JSR     PC,READLU       ;READ LU REG 17
     005024  132776  000002  000004               BITB    #BIT1,@4(SP)    ;GET EXPECTED STATE OF OCOR
     005032  001413                               BEQ     12$             ;BR IF EXPECTED OCOR = 0
     005034  132737  000020  002364               BITB    #OCOR,REDBYT    ;SEE IF OCOR = 1
     005042  001031                               BNE     20$             ;BR IF OCOR = 1
     005044  004737  004526                       JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                          ;REPORT OCOR NOT SET
     005050                                       ERRDF   9,EM9,ERR4
     005050  104455                                                                        TRAP    C$ERDF
     005052  000011                                                                        .WORD   9
     005054  012432                                                                        .WORD   EM9
     005056  015622                                                                        .WORD   ERR4
     005060  000412                               BR      16$             ;TAKE ERROR RETURN
     005062  132737  000020  002364   12$:        BITB    #OCOR,REDBYT    ;SEE IF OCOR = 0
     005070  001416                               BEQ     20$             ;BR IF OCOR = 0
     005072  004737  004526                       JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                          ;REPORT OCOR NOT CLEARED
     005076                                       ERRDF   10,EM10,ERR4
     005076  104455                                                                        TRAP    C$ERDF
     005100  000012                                                                        .WORD   10
     005102  012447                                                                        .WORD   EM10
     005104  015622                                                                        .WORD   ERR4
     005106  016637  000002  002400   16$:        MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
     005114  013706  002346                       MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
     005120  013746  002362                       MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
     005124  000407                               BR      23$
     005126  062766  000002  000004   20$:        ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
     005134  012637  002352                       MOV     (SP)+,SUBRPC
     005140  012637  002400                       MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
     005144  000207                   23$:        RTS     PC              ;RETURN





                                  ;*******************************************************************************
                                  ;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC. AND THEN RETURNS.
                                  ;*******************************************************************************
     005146  010146                   WAIT50: MOV     R1,-(SP)        ;SAVE R1
     005150  012701  000310                   MOV     #200.,R1        ;INIT COUNTER
     005154  005301                   3$:     DEC     R1              ;DECREMENT COUNTER
     005156  001376                           BNE     3$              ;BR IF NOT DONE YET
     005160  012601                           MOV     (SP)+,R1        ;RESTORE R1
     005162  000207                           RTS     PC              ;RETURN
```

```
                                                ;*****************************************************************************
                                                ;* STALL - THIS SUBROUTINE STALLS FOR ABOUT A MICRO-SEC.
                                                ;*****************************************************************************
005164  000240                                  STALL:  NOP
005166  000240                                          NOP
005170  000240                                          NOP
005172  000207                                          RTS     PC

                                                ;*****************************************************************************
                                                ;* LDTXSI - THIS SUBROUTINE LOADS THE TX SILO (REGS 10,11) WITH THE DATA PASSED
                                                ;*        IN BITS 0-11 OF TXWORD.
                                                ;*****************************************************************************
005174  013746  002400                          LDTXSI: MOV     REGNUM,-(SP)       ;SAVE LU REG NO.
005200  042737  170000  002422                          BIC     #170000,TXWORD     ;CLEAR UNUSED BITS
005206  012737  000011  002400                          MOV     #11,REGNUM         ;SET REG NO. = 11
005214  113737  002423  002366                          MOVB    TXWORD+1,WRIBYT    ;SET DATA TO BE WRITTEN INTO REG 11
005222  004737  003750                                  JSR     PC,WRITLU          ;LOAD DATA INTO REG 11
005226  012737  000010  002400                          MOV     #10,REGNUM         ;SET REG NO. = 10
005234  113737  002422  002366                          MOVB    TXWORD,WRIBYT      ;SET DATA TO BE WRITTEN INTO REG 10
005242  004737  003750                                  JSR     PC,WRITLU          ;LOAD DATA INTO REG 10
005246  012637  002400                                  MOV     (SP)+,REGNUM       ;RESTORE LU REG NO.
005252  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.
                                                ;*****************************************************************************
005254  010146                                  STPLU:  MOV     R1,-(SP)           ;SAVE R1
005256  017601  000002                                  MOV     @2(SP),R1          ;GET DESIRED NO. OF CYCLES
005262  001426                                          BEQ     6$                 ;IF DESIRED CYCLES = 0, RETURN
005264  100006                                          BPL     2$                 ;BR IF CHIP TYPE CHECK NOT NECESSARY
005266  042701  100000                                  BIC     #BIT15,R1          ;CLEAR FLAG BIT
005272  005737  002430                                  TST     CHPTYP             ;SEE IF SIG USYRT
005276  001401                                          BEQ     2$                 ;BR IF YES
005300  005301                                          DEC     R1                 ;DECREMENT CYCLE COUNT
005302  152777  000010  175140          2$:             BISB    #LULOOP,@BSEL1     ;SET LU LOOP BIT
005310  152777  000020  175132          3$:             BISB    #STEPLU,@BSEL1     ;SET THE STEPLU BIT (CLOCK THE TRANSMITTER)
005316  004737  005164                                  JSR     PC,STALL           ;STALL
005322  142777  000020  175120                          BICB    #STEPLU,@BSEL1     ;CLEAR THE STEPLU BIT (CLOCK THE RECEIVER)
005330  004737  005164                                  JSR     PC,STALL           ;STALL
005334  005301                                          DEC     R1                 ;DECREMENT CYCLE COUNTER
005336  001364                                          BNE     3$                 ;BR IF NOT DONE YET
005340  062766  000002  000002          6$:             ADD     #2,2(SP)           ;FIX UP RETURN PC
005346  012601                                          MOV     (SP)+,R1           ;RESTORE R1
005350  000207                                          RTS     PC                 ;RETURN
```

```



                                        ;*************************************************************************
                                        ;* OACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF OACT (REG 11) AND
                                        ;*       REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
                                        ;*       WORD FOLLOWING THE CALL.
                                        ;*************************************************************************
005352  013746  002400                  OACTIV: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
005356  013746  002352                          MOV     SUBRPC,-(SP)
005362  005737  002352                          TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
005366  001006                                  BNE     1$              ;BR IF YES
005370  016637  000004  002352                  MOV     4(SP),SUBRPC
005376  162737  000004  002352                  SUB     #4,SUBRPC       ;GET PC OF SUBROUTINE CALL
005404  012737  000011  002400  1$:             MOV     #11,REGNUM      ;SET REG NO. = 11
005412  004737  003672                          JSR     PC,READLU       ;READ REG 11
005416  032776  000001  000004                  BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF OACT
005424  001413                                  BEQ     3$              ;BR IF EXPECTED OACT = 0
005426  132737  000100  002364                  BITB    #OACT,REDBYT    ;SEE IF OACT = 1
005434  001031                                  BNE     9$              ;BR IF OACT = 1
005436  004737  004526                          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                        ;REPORT OACT NOT SET
005442                                          ERRDF   11,EM11,ERR4
005442  104455                                                                          TRAP    C$ERDF
005444  000013                                                                          .WORD   11
005446  012470                                                                          .WORD   EM11
005450  015622                                                                          .WORD   ERR4
005452  000412                                  BR      6$              ;TAKE ERROR RETURN
005454  132737  000100  002364  3$:             BITB    #OACT,REDBYT    ;SEE IF OACT = 0
005462  001416                                  BEQ     9$              ;BR IF OACT = 0
005464  004737  004526                          JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                        ;REPORT OACT NOT CLEARED
005470                                          ERRDF   12,EM12,ERR4
005470  104455                                                                          TRAP    C$ERDF
005472  000014                                                                          .WORD   12
005474  012505                                                                          .WORD   EM12
005476  015622                                                                          .WORD   ERR4
005500  016637  000002  002400  6$:             MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
005506  013706  002346                          MOV     PSTACK,SP       ;RESTORE PROGRAM STACK TO BASE LEVEL
005512  013746  002362                          MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
005516  000407                                  BR      12$
005520  062766  000002  000004  9$:             ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
005526  012637  002352                          MOV     (SP)+,SUBRPC
005532  012637  002400                          MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
005536  000207                          12$:    RTS     PC              ;RETURN





                                        ;*************************************************************************
                                        ;* INITRN - THIS SUBROUTINE INITIATES TRANSMISSION OF A MESSAGE, BY DOING A
                                        ;*       MASTER CLEAR, LOADING AX2-15 AND REG 17 WITH THE DATA PASSED IN THE 2
                                        ;*       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.
                                         ;*******************************************************************************
005540  010146                          INITRN: MOV     R1,-(SP)          ;SAVE R1
005542  013746  002400                          MOV     REGNUM,-(SP)      ;SAVE LU REG NO.
005546  013746  002402                          MOV     AXNUM,-(SP)       ;SAVE AX BYTE NO.
005552  016637  000006  002352                  MOV     6(SP),SUBRPC
005560  162737  000004  002352                  SUB     #4,SUBRPC         ;GET PC OF SUBR CALL
005566  004737  003576                          JSR     PC,MSTCLR         ;ISSUE A MASTER CLEAR
005572  004737  004662                          JSR     PC,OSIRDY         ;CHECK ORDY=1, OCOR=0
005576  000001                                  1
005600  004737  005352                          JSR     PC,OACTIV         ;CHK OACT=0
005604  000000                                  0
005606  012737  000004  002402                  MOV     #4,AXNUM          ;SET AX BYTE NO. = 4 FOR AX2
005614  117637  000006  002374                  MOVB    @6(SP),WAX15      ;SET DATA BYTE TO LOAD INTO AX2-15
005622  012737  000400  002422                  MOV     #TXSOM,TXWORD     ;SET TSOM BIT
005630  113737  002374  002422                  MOVB    WAX15,TXWORD      ;SET SYNCH CHAR
005636  005037  002376                          CLR     WAX16
005642  004737  004312                          JSR     PC,WRITAX         ;LOAD AX2
005646  012737  000017  002400                  MOV     #17,REGNUM        ;SET REG NO. = 17
005654  062766  000002  000006                  ADD     #2,6(SP)          ;INCR POINTER TO NEXT DATA BYTE
005662  117637  000006  002366                  MOVB    @6(SP),WRIBYT     ;SET DATA BYTE TO LOAD INTO REG 17
005670  004737  003750                          JSR     PC,WRITLU         ;LOAD REG 17
005674  004737  005174                          JSR     PC,LDTXSI         ;LOAD THE SILO WITH SOM CHAR
005700  004737  005174                          JSR     PC,LDTXSI         ;LOAD ANOTHER SOM INTO SILO
005704  004737  005146                          JSR     PC,WAIT50         ;WAIT FOR DATA TO RIPPLE
005710  004737  004662                          JSR     PC,OSIRDY         ;CHK ORDY=1, OCOR=1
005714  000003                                  3
005716  004737  005352                          JSR     PC,OACTIV         ;CHK FOR OACT = 0
005722  000000                                  0
005724  005001                                  CLR     R1                ;INIT CYCLE COUNTER
005726  012737  000011  002400                  MOV     #11,REGNUM        ;SET LU REG NO. = 11
005734  152777  000010  174506  6$:             BISB    #LULOOP,@BSEL1    ;SET LINE UNIT LOOP BIT
005742  152777  000020  174500                  BISB    #STEPLU,@BSEL1    ;SET CLOCK BIT
005750  004737  005164                          JSR     PC,STALL          ;STALL FOR MICRO-SEC
005754  004737  003672                          JSR     PC,READLU         ;READ REG 11
005760  132737  000100  002364                  BITB    #OACT,REDBYT      ;SEE IF OACT = 1 YET
005766  001014                                  BNE     9$                ;BR IF OACT = 1
005770  142777  000020  174452                  BICB    #STEPLU,@BSEL1    ;CLEAR CLOCK BIT
005776  004737  005164                          JSR     PC,STALL          ;STALL FOR A MICRO-SEC
006002  005201                                  INC     R1                ;INCR CYCLE COUNT
006004  020127  000003                          CMP     R1,#3             ;SEE IF 3 CYCLES DONE YET
006010  002751                                  BLT     6$                ;BR IF NOT
006012  004737  005352                          JSR     PC,OACTIV         ;CHK FOR OACT = 1
006016  000001                                  1
006020  012737  000017  002400  9$:             MOV     #17,REGNUM        ;SET REG NO. = 17
006026  005037  002430                          CLR     CHPTYP            ;CLEAR USYRT CHIP INDICATOR
006032  004737  003672                          JSR     PC,READLU         ;READ REG 17
006036  132737  000020  002364                  BITB    #OCOR,REDBYT      ;CHK FOR OCOR CLEARED YET
006044  001403                                  BEQ     12$               ;BR IF YES - IT IS SIG CHIP
006046  012737  000001  002430                  MOV     #1,CHPTYP         ;SET INDICATOR FOR OTHER CHIP TYPE
006054  142777  000020  174366  12$:            BICB    #STEPLU,@BSEL1    ;CLEAR CLOCK BIT
006062  004737  005164                          JSR     PC,STALL          ;STALL FOR MICRO-SEC
006066  004737  004662                          JSR     PC,OSIRDY         ;CHK FOR ORDY = 1, OCOR = 0
006072  000001                                  1
006074  062766  000002  000006                  ADD     #2,6(SP)          ;FIX UP RETURN PC
006102  012637  002402                          MOV     (SP)+,AXNUM       ;RESTORE AX BYTE NO.
```

```
006106  012637  002400                          MOV     (SP)+,REGNUM    ;RESTORE LU REG NO.
006112  012601                                  MOV     (SP)+,R1        ;RESTORE R1
006114  005037  002352                          CLR     SUBRPC          ;CLEAR SUBR CALL PC
006120  000207                                  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.
                                        ;*************************************************************************
006122  010146                          TXCHAR: MOV     R1,-(SP)        ;SAVE R1
006124  010246                                  MOV     R2,-(SP)        ;SAVE R2
006126  016637  000004  002352                  MOV     4(SP),SUBRPC
006134  162737  000004  002352                  SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
006142  017637  000004  002422                  MOV     @4(SP),TXWORD   ;GET DATA TO BE TRANSMITTED
006150  004737  005174                          JSR     PC,LDTXSI       ;LOAD THE TX SILO WITH THE DATA
006154  004737  005146                          JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE DOWN SILO
006160  062766  000002  000004                  ADD     #2,4(SP)        ;INCR POINTER
006166  005001                                  CLR     R1              ;INIT CYCLE COUNT
006170  017602  000004                          MOV     @4(SP),R2       ;GET DESIRED NO. OF CYCLES
006174  005702                                  TST     R2              ;SEE IF CHIP TYPE CHK SHOULD BE MADE
006176  100006                                  BPL     9$              ;BR IF NOT
006200  042702  100000                          BIC     #BIT15,R2       ;CLEAR FLAG BIT
006204  005737  002430                          TST     CHPTYP          ;SEE IF SIG USYRT
006210  001401                                  BEQ     9$              ;BR IF YES
006212  005302                                  DEC     R2              ;DECREMENT NO. OF CYCLES
006214  004737  005352                  9$:     JSR     PC,OACTIV       ;CHK OACT = 1
006220  000001                                  1
006222  020102                                  CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
006224  001410                                  BEQ     12$             ;BR IF YES
006226  004737  004662                          JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
006232  000003                                  3
006234  004737  005254                          JSR     PC,STPLU        ;STEP LU ONE CYCLE
006240  000001                                  1
006242  005201                                  INC     R1              ;INCR CYCLE COUNT
006244  000763                                  BR      9$
006246  004737  004662                  12$:    JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
006252  000001                                  1
006254  062766  000002  000004                  ADD     #2,4(SP)        ;FIX UP RETURN PC
006262  005037  002352                          CLR     SUBRPC          ;CLEAR SUBR CALL PC
006266  012602                                  MOV     (SP)+,R2        ;RESTORE R2
006270  012601                                  MOV     (SP)+,R1        ;RESTORE R1
006272  000207                                  RTS     PC              ;RETURN
```

```

                                         ;*******************************************************************************
                                         ;* ENDTRN - THIS SUBROUTINE CLEARS THE TRANSMITTER BY SETTING OC. THE PROGRAM
                                         ;*        WAITS FOR 50 US, AND CHECKS FOR ORDY=1, OCOR=0, OACT=0, RTS=0.
                                         ;*******************************************************************************
006274  010146                           ENDTRN: MOV     R1,-(SP)            ;SAVE R1
006276  013746  002400                           MOV     REGNUM,-(SP)        ;SAVE LU REG NO.
006302  016637  000004  002352                   MOV     4(SP),SUBRPC
006310  162737  000004  002352                   SUB     #4,SUBRPC           ;GET PC OF SUBROUTINE CALL
006316  012737  000011  002400                   MOV     #11,REGNUM          ;SET LU REG NO. = 11
006324  012737  000200  002366                   MOV     #OC,WRIBYT          ;SET OC IN DATA
006332  004737  003750                           JSR     PC,WRITLU           ;SET OC IN REG 11
006336  004737  005146                           JSR     PC,WAIT50           ;STALL FOR >50 US.
006342  004737  004662                           JSR     PC,OSIRDY           ;CHK ORDY=1, OCOR=0
006346  000001                                   1
006350  004737  005352                           JSR     PC,OACTIV           ;CHK OACT = 0
006354  000000                                   0
006356  012737  000013  002400                   MOV     #13,REGNUM          ;SET REG NO. = 13
006364  004737  003672                           JSR     PC,READLU           ;READ REG 13
006370  032737  000040  002364                   BIT     #RTS,REDBYT         ;CHK FOR RTS = 0
006376  001406                                   BEQ     3$                  ;BR IF RTS = 0
006400  004737  004526                           JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
                                         ;REPORT RTS NOT CLEARED
006404                                           ERRDF   65,EM65,ERR4
006404  104455                                                                                       TRAP    C$ERDF
006406  000101                                                                                       .WORD   65
006410  014244                                                                                       .WORD   EM65
006412  015622                                                                                       .WORD   ERR4
006414  005037  002352                   3$:     CLR     SUBRPC              ;CLEAR SUBR CALL PC
006420  012637  002400                           MOV     (SP)+,REGNUM        ;RESTORE LU REG NO.
006424  012601                                   MOV     (SP)+,R1            ;RESTORE R1
006426  000207                                   RTS     PC                  ;RETURN





                                         ;*******************************************************************************
                                         ;* ISIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ICIR (REG 17)
                                         ;*        AND IRDY (REG 12) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
                                         ;*        AS PASSED IN BIT 0 (ICIR) AND BIT 1 (IRDY) OF THE WORD FOLLOWING THE
                                         ;*        CALL.
                                         ;*        IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS
                                         ;*        IN RETADR.
                                         ;*******************************************************************************
006430  013746  002400                   ISIRDY: MOV     REGNUM,-(SP)        ;SAVE LU REG NO.
006434  013746  002352                           MOV     SUBRPC,-(SP)
006440  005737  002352                           TST     SUBRPC              ;SEE IF THIS IS A NESTED CALL
006444  001006                                   BNE     1$                  ;BR IF YES
006446  016637  000004  002352                   MOV     4(SP),SUBRPC
006454  162737  000004  002352                   SUB     #4,SUBRPC           ;GET PC OF SUBR CALL
006462  012737  000012  002400           1$:     MOV     #12,REGNUM          ;SET REG NO. TO 12
006470  004737  003672                           JSR     PC,READLU           ;READ REG 12
006474  032776  000002  000004                   BIT     #BIT1,@4(SP)        ;GET EXPECTED STATE OF IRDY
006502  001413                                   BEQ     3$                  ;BR IF EXPECTED IRDY = 0
006504  132737  000020  002364                   BITB    #IRDY,REDBYT        ;SEE IF IRDY = 1
006512  001022                                   BNE     9$                  ;BR IF IRDY = 1
```

```
006514  004737  004526                    JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                  ;REPORT IRDY NOT SET
006520                                    ERRDF   17,EM17,ERR4
006520  104455                                                                   TRAP    C$ERDF
006522  000021                                                                   .WORD   17
006524  012643                                                                   .WORD   EM17
006526  015622                                                                   .WORD   ERR4
006530  000451                            BR      16$                ;TAKE ERROR EXIT
006532  132737  000020  002364  3$:       BITB    #IRDY,REDBYT       ;SEE IF IRDY = 0
006540  001407                            BEQ     9$                 ;BR IF IRDY = 0
006542  004737  004526                    JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                  ;REPORT IRDY NOT CLEARED
006546                                    ERRDF   18,EM18,ERR4
006546  104455                                                                   TRAP    C$ERDF
006550  000022                                                                   .WORD   18
006552  012660                                                                   .WORD   EM18
006554  015622                                                                   .WORD   ERR4
006556  000436                            BR      16$                ;TAKE ERROR RETURN
006560  012737  000017  002400  9$:       MOV     #17,REGNUM         ;SET REG NO. = 17
006566  004737  003672                    JSR     PC,READLU          ;READ REG 17
006572  132776  000001  000004            BITB    #BIT0,@4(SP)       ;GET EXPECTED STATE OF ICIR
006600  001413                            BEQ     12$                ;BR IF EXPECTED ICIR = 0
006602  132737  000010  002364            BITB    #ICIR,REDBYT       ;SEE IF ICIR = 1
006610  001031                            BNE     20$                ;BR IF ICIR = 1
006612  004737  004526                    JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                  ;REPORT ICIR NOT SET
006616                                    ERRDF   19,EM19,ERR4
006616  104455                                                                   TRAP    C$ERDF
006620  000023                                                                   .WORD   19
006622  012701                                                                   .WORD   EM19
006624  015622                                                                   .WORD   ERR4
006626  000412                            BR      16$                ;TAKE ERROR RETURN
006630  132737  000010  002364  12$:      BITB    #ICIR,REDBYT       ;SEE IF ICIR = 0
006636  001416                            BEQ     20$                ;BR IF ICIR = 0
006640  004737  004526                    JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                  ;REPORT ICIR NOT CLEARED
006644                                    ERRDF   20,EM20,ERR4
006644  104455                                                                   TRAP    C$ERDF
006646  000024                                                                   .WORD   20
006650  012716                                                                   .WORD   EM20
006652  015622                                                                   .WORD   ERR4
006654  016637  000002  002400  16$:      MOV     2(SP),REGNUM       ;RESTORE LU REG NO.
006662  013706  002346                    MOV     PSTACK,SP          ;RESTORE STACK POINTER TO BASE LEVEL
006666  013746  002362                    MOV     RETADR,-(SP)       ;FIX ERROR RETURN PC
006672  000407                            BR      23$
006674  062766  000002  000004  20$:      ADD     #2,4(SP)           ;FIX UP ERROR-FREE RETURN PC
006702  012637  002352                    MOV     (SP)+,SUBRPC
006706  012637  002400                    MOV     (SP)+,REGNUM       ;RESTORE LU REG NO.
006712  000207                  23$:      RTS     PC                 ;RETURN

                                  ;*******************************************************************************
                                  ;* IACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF IACT (REG 12) AND
                                  ;*        REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
```

```
                                        ;*      WORD FOLLOWING THE CALL.
                                        ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN
                                        ;*      RETADR.
                                        ;*************************************************************************
006714  013746  002400                  IACTIV: MOV     REGNUM,-(SP)            ;SAVE LU REG NO.
006720  013746  002352                          MOV     SUBRPC,-(SP)
006724  005737  002352                          TST     SUBRPC                  ;SEE IF THIS IS A NESTED CALL
006730  001006                                  BNE     1$                      ;BR IF YES
006732  016637  000004  002352                  MOV     4(SP),SUBRPC
006740  162737  000004  002352                  SUB     #4,SUBRPC               ;GET PC OF SUBR CALL
006746  012737  000012  002400  1$:             MOV     #12,REGNUM              ;SET REG NO. = 12
006754  004737  003672                          JSR     PC,READLU               ;READ REG 12
006760  032776  000001  000004                  BIT     #BIT0,@4(SP)            ;GET EXPECTED STATE OF IACT
006766  001413                                  BEQ     3$                      ;BR IF EXPECTED IACT = 0
006770  132737  000100  002364                  BITB    #IACT,REDBYT            ;SEE IF IACT = 1
006776  001031                                  BNE     9$                      ;BR IF IACT = 1
007000  004737  004526                          JSR     PC,GETALL               ;GET REGS FOR PRINTOUT
                                        ;REPORT IACT NOT SET
007004                                          ERRDF   21,EM21,ERR4
007004  104455                                                                                  TRAP    C$ERDF
007006  000025                                                                                  .WORD   21
007010  012737                                                                                  .WORD   EM21
007012  015622                                                                                  .WORD   ERR4
007014  000412                                  BR      6$                      ;TAKE ERROR EXIT
007016  132737  000100  002364  3$:             BITB    #IACT,REDBYT            ;SEE IF IACT = 0
007024  001416                                  BEQ     9$                      ;BR IF IACT = 0
007026  004737  004526                          JSR     PC,GETALL               ;GET REGS FOR PRINTOUT
                                        ;REPORT IACT NOT CLEARED
007032                                          ERRDF   22,EM22,ERR4
007032  104455                                                                                  TRAP    C$ERDF
007034  000026                                                                                  .WORD   22
007036  012754                                                                                  .WORD   EM22
007040  015622                                                                                  .WORD   ERR4
007042  016637  000002  002400  6$:             MOV     2(SP),REGNUM            ;RESTORE LU REG NO.
007050  013706  002346                          MOV     PSTACK,SP               ;RESTORE PROGRAM STACK TO BASE LEVEL
007054  013746  002362                          MOV     RETADR,-(SP)            ;FIX UP ERROR RETURN PC
007060  000407                                  BR      12$
007062  062766  000002  000004  9$:             ADD     #2,4(SP)                ;FIX UP ERROR-FREE RETURN PC
007070  012637  002352                          MOV     (SP)+,SUBRPC
007074  012637  002400                          MOV     (SP)+,REGNUM            ;RESTORE LU REG NO.
007100  000207                          12$:    RTS     PC                      ;RETURN






                                        ;*************************************************************************
                                        ;* RSEOM - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF RSOM AND REOM IN
                                        ;* AX0-16, AND REPORTS AN ERROR IF EITHER IS NOT SET TO THE STATE PASSED IN BITS
                                        ;* 0,1, RESPECTIVELY, OF THE WORD FOLLOWING THE CALL.
                                        ;* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN RETADR.
                                        ;*************************************************************************
007102  013746  002402                  RSEOM:  MOV     AXNUM,-(SP)             ;SAVE AX BYTE NO.
007106  013746  002352                          MOV     SUBRPC,-(SP)
007112  005737  002352                          TST     SUBRPC                  ;SEE IF THIS IS A NESTED CALL
007116  001006                                  BNE     1$                      ;BR IF YES
007120  016637  000004  002352                  MOV     4(SP),SUBRPC
```

```
007126  162737  000004  002352          SUB     #4,SUBRPC       ;GET PC OF SUBR CALL
007134  012737  000001  002402  1$:     MOV     #1,AXNUM        ;SET AX BYTE NO. FOR AX0-16
007142  004737  004124                  JSR     PC,READAX       ;READ AX0
007146  032776  000001  000004          BIT     #BIT0,@4(SP)    ;GET EXPECTED STATE OF RSOM
007154  001413                          BEQ     3$              ;BR IF EXPECTED RSOM = 0
007156  132737  000001  002372          BITB    #RSOM,RAX16     ;SEE IF RSOM = 1
007164  001022                          BNE     9$              ;BR IF RSOM = 1
007166  004737  004526                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT RSOM NOT SET
007172                                  ERRDF   29,EM29,ERR6
007172  104455                                                                  TRAP    C$ERDF
007174  000035                                                                  .WORD   29
007176  013173                                                                  .WORD   EM29
007200  017012                                                                  .WORD   ERR6
007202  000444                          BR      16$             ;TAKE ERROR EXIT
007204  132737  000001  002372  3$:     BITB    #RSOM,RAX16     ;SEE IF RSOM = 0
007212  001407                          BEQ     9$              ;BR IF RSOM = 0
007214  004737  004526                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT RSOM NOT CLEARED
007220                                  ERRDF   28,EM28,ERR6
007220  104455                                                                  TRAP    C$ERDF
007222  000034                                                                  .WORD   28
007224  013152                                                                  .WORD   EM28
007226  017012                                                                  .WORD   ERR6
007230  000431                          BR      16$             ;TAKE ERROR RETURN
007232  132776  000002  000004  9$:     BITB    #BIT1,@4(SP)    ;GET EXPECTED STATE OF REOM
007240  001413                          BEQ     12$             ;BR IF EXPECTED REOM = 0
007242  132737  000002  002372          BITB    #REOM,RAX16     ;SEE IF REOM = 1
007250  001031                          BNE     20$             ;BR IF REOM = 1
007252  004737  004526                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT REOM NOT SET
007256                                  ERRDF   31,EM31,ERR6
007256  104455                                                                  TRAP    C$ERDF
007260  000037                                                                  .WORD   31
007262  013231                                                                  .WORD   EM31
007264  017012                                                                  .WORD   ERR6
007266  000412                          BR      16$             ;TAKE ERROR RETURN
007270  132737  000002  002372  12$:    BITB    #REOM,RAX16     ;SEE IF REOM = 0
007276  001416                          BEQ     20$             ;BR IF REOM = 0
007300  004737  004526                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT REOM NOT CLEARED
007304                                  ERRDF   30,EM30,ERR6
007304  104455                                                                  TRAP    C$ERDF
007306  000036                                                                  .WORD   30
007310  013210                                                                  .WORD   EM30
007312  017012                                                                  .WORD   ERR6
007314  016637  000002  002402  16$:    MOV     2(SP),AXNUM     ;RESTORE AX BYTE NO.
007322  013706  002346                  MOV     PSTACK,SP       ;RESTORE STACK POINTER TO BASE LEVEL
007326  013746  002362                  MOV     RETADR,-(SP)    ;FIX ERROR RETURN PC
007332  000407                          BR      23$
007334  062766  000002  000004  20$:    ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
007342  012637  002352                  MOV     (SP)+,SUBRPC
007346  012637  002402                  MOV     (SP)+,AXNUM     ;RESTORE AX BYTE NO.
007352  000207                  23$:    RTS     PC              ;RETURN
```

```
                                         ;*****************************************************************************
                                         ;* RDRXSI - THIS SUBROUTINE READS THE RCV SILO (REGS 10,12) AND RETURNS THE
                                         ;*        SILO ENTRY IN BITS 0-11 OF RXWORD.
                                         ;*****************************************************************************
007354  013746  002400                   RDRXSI: MOV     REGNUM,-(SP)      ;SAVE LU REG NO.
007360  012737  000012  002400                   MOV     #12,REGNUM        ;SET REG NO. = 12
007366  004737  003672                           JSR     PC,READLU         ;READ LU REG 12
007372  113737  002364  002425                   MOVB    REDBYT,RXWORD+1   ;GET HI BITS OF SILO ENTRY
007400  042737  170000  002424                   BIC     #170000,RXWORD    ;CLEAR UNUSED BITS
007406  012737  000010  002400                   MOV     #10,REGNUM        ;SET REG NO. = 10
007414  004737  003672                           JSR     PC,READLU         ;READ REG 10
007420  113737  002364  002424                   MOVB    REDBYT,RXWORD     ;GET LOW BITS OF SILO ENTRY
007426  012637  002400                           MOV     (SP)+,REGNUM      ;RESTORE LU REG NO.
007432  000207                                   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.
                                         ;*        IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
                                         ;*        CONTAINED IN RETADR.
                                         ;*****************************************************************************
007434  010146                           RCV1ST: MOV     R1,-(SP)          ;SAVE R1
007436  010346                                   MOV     R3,-(SP)          ;SAVE R3
007440  013746  002400                           MOV     REGNUM,-(SP)      ;SAVE LU REG NO.
007444  016637  000006  002352                   MOV     6(SP),SUBRPC
007452  162737  000004  002352                   SUB     #4,SUBRPC         ;GET PC OF SUBROUTINE CALL
007460  012737  000012  002400                   MOV     #12,REGNUM        ;SET LU REG NO. = 12
007466  005001                                   CLR     R1                ;INIT CYCLE COUNT TO 0
007470  017603  000006                           MOV     @6(SP),R3         ;GET CYCLE COUNT LIMIT
007474  062703  000003                           ADD     #3,R3
007500  005776  000006                           TST     @6(SP)            ;SEE IF DESIRED CYCLES = 0
007504  001414                                   BEQ     8$                ;BR IF YES
007506  004737  006714                           JSR     PC,IACTIV         ;CHK FOR IACT = 0
007512  000000                                   0
007514  004737  006430                           JSR     PC,ISIRDY         ;CHK FOR ICIR = 1, IRDY = 0
007520  000001                                   1
007522  004737  007102                           JSR     PC,RSEOM          ;CHK RSOM = 0, REOM = 0 IN AX0-16
007526  000000                                   0
007530  004737  005254                   6$:     JSR     PC,STPLU          ;CLOCK LU FOR 1 CYCLE
007534  000001                                   1
007536  004737  005146                   8$:     JSR     PC,WAIT50         ;ALLOW SILO DATA TO RIPPLE
007542  005201                                   INC     R1                ;INCREMENT CYCLE COUNT
007544  004737  003672                           JSR     PC,READLU         ;READ REG 12
007550  132737  000020  002364                   BITB    #IRDY,REDBYT      ;SEE IF IRDY = 1 YET
007556  001005                                   BNE     9$                ;BR IF IRDY = 1
007560  020103                                   CMP     R1,R3             ;SEE IF LIMIT EXCEEDED
007562  002762                                   BLT     6$                ;BR IF NOT YET
007564  004737  006430                           JSR     PC,ISIRDY         ;CHK FOR ICIR = 1, IRDY = 1
```

```
007570  000003                                   3
007572  020176  000006          9$:      CMP     R1,@6(SP)        ;SEE IF LESS THAN REQUIRED CYCLES
007576  002003                           BGE     12$              ;BR IF NOT
007600  004737  006430                   JSR     PC,ISIRDY        ;CHK FOR ICIR = 1, IRDY = 0
007604  000001                           1
007606  004737  006714          12$:     JSR     PC,IACTIV        ;CHK FOR IACT = 1
007612  000001                           1
007614  004737  006430                   JSR     PC,ISIRDY        ;CHK FOR ICIR = 1, IRDY = 1
007620  000003                           3
007622  062766  000002  000006           ADD     #2,6(SP)         ;FIX UP RETURN PC
007630  012637  002400                   MOV     (SP)+,REGNUM     ;RESTORE LU REG NO.
007634  012603                           MOV     (SP)+,R3         ;RESTORE R3
007636  012601                           MOV     (SP)+,R1         ;RESTORE R1
007640  005037  002352                   CLR     SUBRPC           ;CLEAR SUBR CALL PC
007644  000207                           RTS     PC               ;RETURN

                                ;*************************************************************************
                                ;* STPERR - THIS SUBROUTINE LOADS THE CONTENTS OF THE FIRST WORD FOLLOWING THE
                                ;*       CALL INTO REG 17, AND SETS THE IERR BIT, AND CLOCKS THE LINE UNIT
                                ;*       FOR THE NO. OF CYCLES PASSED IN THE 2ND WORD FOLLOWING THE CALL. THEN,
                                ;*       IT RESTORES REG 17 TO ITS ORIGINAL CONTENTS, CLEARING THE IERR BIT.
                                ;*************************************************************************
007646  013746  002400          STPERR:  MOV     REGNUM,-(SP)     ;SAVE LU REG NO.
007652  012737  000017  002400           MOV     #17,REGNUM       ;SET LU REG NO. = 17
007660  017637  000002  002366           MOV     @2(SP),WRIBYT
007666  152737  000002  002366           BISB    #IERR,WRIBYT
007674  004737  003750                   JSR     PC,WRITLU        ;SET IERR BIT IN REG 17
007700  062766  000002  000002           ADD     #2,2(SP)         ;INCREMENT SUBR ARGUMENT POINTER
007706  017637  000002  007720           MOV     @2(SP),3$        ;GET DESIRED NO. OF CYCLES
007714  004737  005254                   JSR     PC,STPLU         ;CLOCK LU FOR DESIRED NO. OF CYCLES
007720  000000                  3$:      .WORD   0                ;NO. OF CYCLES GOES HERE
007722  142737  000002  002366           BICB    #IERR,WRIBYT
007730  004737  003750                   JSR     PC,WRITLU        ;CLEAR IERR BIT IN REG 17
007734  062766  000002  000002           ADD     #2,2(SP)         ;FIX UP RETURN PC
007742  012637  002400                   MOV     (SP)+,REGNUM     ;RESTORE LU REG NO.
007746  000207                           RTS     PC               ;RETURN

                                ;*************************************************************************
                                ;* CKDATA - THIS SUBROUTINE READS THE RCV SILO AND COMPARES THE SILO ENTRY
                                ;*       TO BITS 0-11 OF THE FIRST WORD FOLLOWING THE CALL. IF THERE IS A
                                ;*       MISMATCH, THE ERROR IS REPORTED AND A RETURN IS MADE TO THE TEST AT THE
                                ;*       ADDRESS CONTAINED IN RETADR. IF BIT 15 = 0 IN THE FIRST WORD
                                ;*       FOLLOWING THE CALL, THE SUBROUTINE WILL NOT CHECK THE BCC BIT (SILO
                                ;*       BIT 8). IF THERE ARE NO ERRORS, THE LINE UNIT IS CLOCKED FOR THE
                                ;*       NUMBER OF CYCLES PASSED IN THE SECOND WORD FOLLOWING THE CALL.
                                ;*************************************************************************
007750  010146                  CKDATA:  MOV     R1,-(SP)         ;SAVE R1
007752  013746  002400                   MOV     REGNUM,-(SP)     ;SAVE LU REG NO.
007756  016637  000004  002352           MOV     4(SP),SUBRPC
```

```
007764  162737  000004  002352            SUB     #4,SUBRPC         ;GET PC OF SUBR CALL
007772  017601  000004                    MOV     @4(SP),R1         ;GET EXPECTED SILO ENTRY
007776  042701  170000                    BIC     #170000,R1        ;CLEAR UNUSED BITS FOR COMPARE
010002  004737  007354                    JSR     PC,RDRXSI         ;READ RCV SILO
010006  023727  002432  000347            CMP     SAVLEN,#TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
010014  001005                            BNE     4$                ;BR IF CHAR LENGTH NOT = 7
010016  042701  000200                    BIC     #BIT7,R1          ;MASK OFF BIT 8TH BIT
010022  042737  000200  002424            BIC     #BIT7,RXWORD
010030  120137  002424            4$:     CMPB    R1,RXWORD         ;COMPARE EXPECTED BITS 0-7 TO ACTUAL
010034  001445                            BEQ     6$                ;BR IF MATCH
010036  005037  002404                    CLR     GOODAT
010042  110137  002404                    MOVB    R1,GOODAT         ;GET EXPECTED DATA
010046  005037  002406                    CLR     BADDAT
010052  113737  002424  002406            MOVB    RXWORD,BADDAT     ;GET ACTUAL DATA
010060  012737  000011  002400            MOV     #11,REGNUM        ;SET REG NO. = 11
010066  004737  003672                    JSR     PC,READLU         ;READ REG 11
010072  132737  000001  002364            BITB    #UNRR,REDBYT      ;SEE IF TX UNDERRUN ERROR
010100  001410                            BEQ     5$                ;BR IF NOT
010102  004737  004526                    JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                  ;REPORT TX UNDERRUN ERROR
010106                                    ERRDF   54,EM54,ERR4
010106  104455                                                                          TRAP    C$ERDF
010110  000066                                                                          .WORD   54
010112  014206                                                                          .WORD   EM54
010114  015622                                                                          .WORD   ERR4
010116  000137  010600                    JMP     36$               ;TAKE ERROR EXIT
010122  012737  000010  002400    5$:     MOV     #10,REGNUM        ;SET REG NO. = 10
010130  004737  004526                    JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                  ;REPORT RCV'D DATA MISCOMPARE
010134                                    ERRDF   34,EM34,ERR8
010134  104455                                                                          TRAP    C$ERDF
010136  000042                                                                          .WORD   34
010140  013320                                                                          .WORD   EM34
010142  020122                                                                          .WORD   ERR8
010144  000137  010600                    JMP     36$               ;TAKE ERROR EXIT
010150  000301                    6$:     SWAB    R1
010152  012737  000012  002400            MOV     #12,REGNUM        ;SET LU REG NO. FOR ERROR REPORTS
010160  120137  002425                    CMPB    R1,RXWORD+1       ;COMPARE EXPECTED SILO BITS 8-11 TO ACTUAL
010164  001002                            BNE     7$                ;BR IF MISMATCH
010166  000137  010554                    JMP     22$               ;CONTINUE
010172  005037  002404            7$:     CLR     GOODAT
010176  110137  002404                    MOVB    R1,GOODAT         ;SET EXPECTED DATA
010202  005037  002406                    CLR     BADDAT
010206  113737  002425  002406            MOVB    RXWORD+1,BADDAT   ;SET ACTUAL DATA
010214  032776  100000  000004            BIT     #BCCCHK,@4(SP)    ;SEE IF BCC SHOULD BE IGNORED
010222  001433                            BEQ     10$               ;BR IF YES
010224  132701  000001                    BITB    #BCC,R1           ;SEE IF EXPECTED BIT = 1
010230  001014                            BNE     8$                ;BR IF YES
010232  132737  000001  002425            BITB    #BCC,RXWORD+1     ;SEE IF ACTUAL BIT = 0
010240  001424                            BEQ     10$               ;BR IF YES
010242  004737  004526                    JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                  ;REPORT BCC NOT CLEARED
010246                                    ERRDF   35,EM35,ERR8
010246  104455                                                                          TRAP    C$ERDF
010250  000043                                                                          .WORD   35
010252  013346                                                                          .WORD   EM35
010254  020122                                                                          .WORD   ERR8
```

```
890 010256  000137  010600                  JMP     36$                 ;TAKE ERROR EXIT
891 010262  132737  000001  002425  8$:     BITB    #BCC,RXWORD+1       ;SEE IF ACTUAL BIT = 1
892 010270  001010                          BNE     10$                 ;BR IF YES
893 010272  004737  004526                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
894                                 ;REPORT BCC NOT SET
895 010276                                  ERRDF   36,EM36,ERR8
    010276  104455                                                                          TRAP    C$ERDF
    010300  000044                                                                          .WORD   36
    010302  013366                                                                          .WORD   EM36
    010304  020122                                                                          .WORD   ERR8
896 010306  000137  010600                  JMP     36$                 ;TAKE ERROR EXIT
897 010312                          10$:
898 010312  132701  000002                  BITB    #EBLK,R1            ;SEE IF EXPECTED BIT = 1
899 010316  001014                          BNE     12$                 ;BR IF YES
900 010320  132737  000002  002425          BITB    #EBLK,RXWORD+1      ;SEE IF ACTUAL BIT = 0
901 010326  001424                          BEQ     14$                 ;BR IF YES
902 010330  004737  004526                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
903                                 ;REPORT EBLK NOT CLEARED
904 010334                                  ERRDF   37,EM37,ERR8
    010334  104455                                                                          TRAP    C$ERDF
    010336  000045                                                                          .WORD   37
    010340  013402                                                                          .WORD   EM37
    010342  020122                                                                          .WORD   ERR8
905 010344  000137  010600                  JMP     36$                 ;TAKE ERROR EXIT
906 010350  132737  000002  002425  12$:    BITB    #EBLK,RXWORD+1      ;SEE IF ACTUAL BIT = 1
907 010356  001010                          BNE     14$                 ;BR IF YES
908 010360  004737  004526                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
909                                 ;REPORT EBLK NOT SET
910 010364                                  ERRDF   38,EM38,ERR8
    010364  104455                                                                          TRAP    C$ERDF
    010366  000046                                                                          .WORD   38
    010370  013423                                                                          .WORD   EM38
    010372  020122                                                                          .WORD   ERR8
911 010374  000137  010600                  JMP     36$                 ;TAKE ERROR EXIT
912 010400                          14$:
913 010400  132701  000004                  BITB    #RAB,R1             ;SEE IF EXPECTED BIT = 1
914 010404  001014                          BNE     16$                 ;BR IF YES
915 010406  132737  000004  002425          BITB    #RAB,RXWORD+1       ;SEE IF ACTUAL BIT = 0
916 010414  001424                          BEQ     18$                 ;BR IF YES
917 010416  004737  004526                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
918                                 ;REPORT RAB NOT CLEARED
919 010422                                  ERRDF   39,EM39,ERR8
    010422  104455                                                                          TRAP    C$ERDF
    010424  000047                                                                          .WORD   39
    010426  013440                                                                          .WORD   EM39
    010430  020122                                                                          .WORD   ERR8
920 010432  000137  010600                  JMP     36$                 ;TAKE ERROR EXIT
921 010436  132737  000004  002425  16$:    BITB    #RAB,RXWORD+1       ;SEE IF ACTUAL BIT = 1
922 010444  001010                          BNE     18$                 ;BR IF YES
923 010446  004737  004526                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
924                                 ;REPORT RAB NOT SET
925 010452                                  ERRDF   40,EM40,ERR8
    010452  104455                                                                          TRAP    C$ERDF
    010454  000050                                                                          .WORD   40
    010456  013460                                                                          .WORD   EM40
    010460  020122                                                                          .WORD   ERR8
926 010462  000137  010600                  JMP     36$                 ;TAKE ERROR EXIT
```

```
010466                                 18$:
010466  132701  000010                         BITB    #OVRR,R1           ;SEE IF EXPECTED BIT = 1
010472  001014                                 BNE     20$                ;BR IF YES
010474  132737  000010  002425                 BITB    #OVRR,RXWORD+1     ;SEE IF ACTUAL BIT = 0
010502  001424                                 BEQ     22$                ;BR IF YES
010504  004737  004526                         JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                       ;REPORT OVRR NOT CLEARED
010510                                         ERRDF   41,EM41,ERR8
010510  104455                                                                              TRAP    C$ERDF
010512  000051                                                                              .WORD   41
010514  013474                                                                              .WORD   EM41
010516  020122                                                                              .WORD   ERR8
010520  000137  010600                         JMP     36$                ;TAKE ERROR EXIT
010524  132737  000010  002425         20$:    BITB    #OVRR,RXWORD+1     ;SEE IF ACTUAL BIT = 1
010532  001010                                 BNE     22$                ;BR IF YES
010534  004737  004526                         JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                       ;REPORT OVRR NOT SET
010540                                         ERRDF   42,EM42,ERR8
010540  104455                                                                              TRAP    C$ERDF
010542  000052                                                                              .WORD   42
010544  013515                                                                              .WORD   EM42
010546  020122                                                                              .WORD   ERR8
010550  000137  010600                         JMP     36$                ;TAKE ERROR EXIT
010554                                 22$:
010554  062766  000002  000004                 ADD     #2,4(SP)           ;INCR SUBROUTINE ARGUMENT POINTER
010562  017637  000004  010574                 MOV     @4(SP),24$         ;GET DESIRED CYCLE COUNT
010570  004737  005254                         JSR     PC,STPLU           ;CLOCK LU FOR DESIRED CYCLES
010574  000000                         24$:    .WORD   0
010576  000407                                 BR      38$                ;TAKE ERROR-FREE EXIT
010600  011637  002400                 36$:    MOV     (SP),REGNUM        ;RESTORE LU REG NO.
010604  013706  002346                         MOV     PSTACK,SP          ;RESTORE PROGRAM STACK TO BASE LEVEL
010610  013746  002362                         MOV     RETADR,-(SP)       ;FIX UP ERROR RETURN PC
010614  000406                                 BR      40$
010616  062766  000002  000004         38$:    ADD     #2,4(SP)           ;FIX UP ERROR-FREE RETURN PC
010624  012637  002400                         MOV     (SP)+,REGNUM       ;RESTORE LU REG NO.
010630  012601                                 MOV     (SP)+,R1           ;RESTORE R1
010632  005037  002352                 40$:    CLR     SUBRPC             ;CLEAR SUBROUTINE PC
010636  000207                                 RTS     PC                 ;RETURN





                                       ;*****************************************************************************
                                       ;* SETUP - THIS SUBROUTINE LOADS THE FIRST WORD AFTER THE CALL INTO AX2-15
                                       ;*        (SYNCH CHAR), LOADS THE SECOND WORD AFTER THE CALL INTO REG 17
                                       ;*        LOADS THE THIRD WORD INTO AX3-15, AND LOADS THE FOURTH INTO AX3-16.
                                       ;*****************************************************************************
010640  013746  002402                 SETUP:  MOV     AXNUM,-(SP)        ;SAVE AX BYTE NO.
010644  013746  002400                         MOV     REGNUM,-(SP)       ;SAVE LU REG NO.
010650  012737  000004  002402                 MOV     #4,AXNUM           ;SET AX BYTE NO. FOR AX2
010656  017637  000004  002374                 MOV     @4(SP),WAX15
010664  005037  002376                         CLR     WAX16
010670  004737  004312                         JSR     PC,WRITAX          ;SET SYNCH CHAR IN AX2-15, CLEAR AX2-16
010674  012737  000017  002400                 MOV     #17,REGNUM         ;SET LU REG NO. = 17
010702  062766  000002  000004                 ADD     #2,4(SP)           ;INCREMENT ARGUMENT POINTER
010710  017637  000004  002366                 MOV     @4(SP),WRIBYT
```

```
010716  004737  003750                    JSR    PC,WRITLU         ;LOAD REG 17
010722  012737  000006  002402            MOV    #6,AXNUM          ;SET AX BYTE NO. FOR AX3
010730  062766  000002  000004            ADD    #2,4(SP)          ;INCREMENT ARGUMENT POINTER
010736  017637  000004  002374            MOV    @4(SP),WAX15
010744  062766  000002  000004            ADD    #2,4(SP)          ;INCR ARGUMENT POINTER
010752  017637  000004  002376            MOV    @4(SP),WAX16
010760  013737  002376  002432            MOV    WAX16,SAVLEN      ;STORE TX AND RCV CHAR LENGTH
010766  004737  004312                    JSR    PC,WRITAX         ;LOAD AX3-15, AX3-16
010772  062766  000002  000004            ADD    #2,4(SP)          ;FIX RETURN PC
011000  012637  002400                    MOV    (SP)+,REGNUM      ;RESTORE LU REG NO.
011004  012637  002402                    MOV    (SP)+,AXNUM       ;RESTORE AX BYTE NO.
011010  005037  002352                    CLR    SUBRPC            ;CLEAR SUBROUTINE PC STORAGE
011014  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.
                                  ;*******************************************************************************
011016  010146                    LODMSG: MOV    R1,-(SP)          ;SAVE R1
011020  010246                            MOV    R2,-(SP)          ;SAVE R2
011022  017601  000004                    MOV    @4(SP),R1         ;GET MSG POINTER INTO R1
011026  062766  000002  000004            ADD    #2,4(SP)          ;INCR ARG POINTER
011034  017602  000004                    MOV    @4(SP),R2         ;GET WORD COUNT INTO R2
011040  062766  000002  000004            ADD    #2,4(SP)          ;FIX UP RETURN PC
011046  012137  002422            6$:     MOV    (R1)+,TXWORD      ;GET NEXT MSG WORD
011052  004737  005174                    JSR    PC,LDTXSI         ;LOAD A WORD INTO TX SILO
011056  005302                            DEC    R2                ;DECR COUNT
011060  001372                            BNE    6$                ;BR IF NOT DONE YET
011062  004737  005146                    JSR    PC,WAIT50         ;WAIT FOR SILO TO RIPPLE
011066  012602                            MOV    (SP)+,R2          ;RESTORE R2
011070  012601                            MOV    (SP)+,R1          ;RESTORE R1
011072  000207                            RTS    PC                ;RETURN

                                  ;*******************************************************************************
                                  ;* RXCHAR - THIS SUBROUTINE READS THE RCV SILO AND CLOCKS THE LINE UNIT.
                                  ;*         FIRST, IT READS THE CHAR FROM THE RCV SILO, AND CHECKS FOR ICIR
                                  ;*         = 1, IRDY = 0. IT THEN CLOCKS THE LINE UNIT FOR THE NO. OF CYCLES
                                  ;*         PASSED IN THE WORD FOLLOWING THE CALL, AND THEN CHECKS FOR ICIR
                                  ;*         = 1, IRDY = 1.
                                  ;*******************************************************************************
011074  010146                    RXCHAR: MOV    R1,-(SP)          ;SAVE R1
011076  016637  000002  002352            MOV    2(SP),SUBRPC
011104  162737  000004  002352            SUB    #4,SUBRPC         ;GET PC OF SUBR CALL
011112  004737  007354                    JSR    PC,RDRXSI         ;READ RCV SILO
011116  004737  005146                    JSR    PC,WAIT50         ;ALLOW SILO TO RIPPLE
011122  005001                            CLR    R1                ;INIT CYCLE COUNT
011124  020176  000002            9$:     CMP    R1,@2(SP)         ;SEE IF REQUIRED CYCLES DONE YET
011130  001410                            BEQ    12$               ;BR IF YES
```

```
011132  004737  006430                      JSR     PC,ISIRDY         ;CHK ICIR = 1, IRDY = 0
011136  000001                              1
011140  004737  005254                      JSR     PC,STPLU          ;CLK LU 1 CYCLE
011144  000001                              1
011146  005201                              INC     R1                ;INCR CYCLE COUNT
011150  000765                              BR      9$
011152  004737  006430          12$:        JSR     PC,ISIRDY         ;CHK ICIR = 1 IRDY = 1
011156  000003                              3
011160  062766  000002  000002              ADD     #2,2(SP)          ;FIX RETURN PC
011166  005037  002352                      CLR     SUBRPC            ;CLEAR SUBR CALL PC
011172  012601                              MOV     (SP)+,R1          ;RESTORE R1
011174  000207                              RTS     PC                ;RETURN





                                ;*************************************************************************
                                ;* CKTBIT - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR 8 CYCLES, AND AFTER EACH
                                ;*          CYCLE, THE TXDATA BIT IN REG 17 IS EXAMINED, AND COMPARED TO A BIT OF
                                ;*          THE CHAR PASSED IN THE WORD FOLLOWING THE CALL.
                                ;*          IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN
                                ;*          RETADR.
                                ;*************************************************************************
011176  013746  002400          CKTBIT: MOV     REGNUM,-(SP)      ;SAVE LU REG NO.
011202  010146                          MOV     R1,-(SP)          ;SAVE R1
011204  016637  000004  002352          MOV     4(SP),SUBRPC
011212  162737  000004  002352          SUB     #4,SUBRPC         ;GET PC OF SUBROUTINE CALL
011220  012701  000001                  MOV     #BIT0,R1          ;INIT BIT POINTER
011224  012737  000017  002400          MOV     #17,REGNUM        ;SET REG NO. = 17
011232  004737  005254          4$:     JSR     PC,STPLU          ;CLOCK LINE UNIT 1 CYCLE
011236  000001                          1
011240  004737  003672                  JSR     PC,READLU         ;READ REG 17
011244  030176  000004                  BIT     R1,@4(SP)         ;SEE IF EXPECTED BIT = 1
011250  001013                          BNE     9$                ;BR IF YES
011252  132737  000040  002364          BITB    #TXDATA,REDBYT    ;SEE IF ACTUAL BIT = 0
011260  001422                          BEQ     12$               ;BR IF YES
011262  004737  004526                  JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                ;REPORT TXDATA NOT CLEARED
011266                                  ERRDF   32,EM32,ERR4
011266  104455                                                                          TRAP    C$ERDF
011270  000040                                                                          .WORD   32
011272  013246                                                                          .WORD   EM32
011274  015622                                                                          .WORD   ERR4
011276  000420                          BR      20$               ;TAKE ERROR RETURN
011300  132737  000040  002364  9$:     BITB    #TXDATA,REDBYT    ;SEE IF ACTUAL BIT = 1
011306  001007                          BNE     12$               ;BR IF YES
011310  004737  004526                  JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                ;REPORT TXDATA BIT NOT SET
011314                                  ERRDF   33,EM33,ERR4
011314  104455                                                                          TRAP    C$ERDF
011316  000041                                                                          .WORD   33
011320  013275                                                                          .WORD   EM33
011322  015622                                                                          .WORD   ERR4
011324  000405                          BR      20$               ;TAKE ERROR EXIT
011326  006301                  12$:    ASL     R1                ;SHIFT BIT POINTER
011330  020127  000400                  CMP     R1,#400           ;SEE IF 8 BITS SCANNED YET
```

```
011334  001336                          BNE     4$              ;BR IF NO
011336  000405                          BR      22$
011340  013706  002346          20$:    MOV     PSTACK,SP       ;RESTORE PROGRAM STACK POINTER TO BASE LEVEL
011344  013746  002362                  MOV     RETADR,-(SP)    ;FIX UP RETURN PC
011350  000406                          BR      40$
011352  062766  000002  000004  22$:    ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
011360  012601                          MOV     (SP)+,R1        ;RESTORE R1
011362  012637  002400                  MOV     (SP)+,REGNUM    ;RESTORE REG NO.
011366  005037  002352          40$:    CLR     SUBRPC          ;CLEAR SUBR CALL PC
011372  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.
                                ;*****************************************************************************
011374  010146                  LDMSG1: MOV     R1,-(SP)        ;SAVE R1
011376  010246                          MOV     R2,-(SP)        ;SAVE R2
011400  004737  011016                  JSR     PC,LODMSG       ;LOAD MSG1 INTO TX SILO
011404  003220                          MSG1
011406  000011                          9.
011410  012701  003224                  MOV     #MSG1+4,R1      ;GET POINTER TO MSG1
011414  012702  003262                  MOV     #RCVBUF,R2      ;GET POINTER TO MSG BUF
011420  012122                  3$:     MOV     (R1)+,(R2)+     ;LOAD A CHAR INTO MSG BUF
011422  020127  003236                  CMP     R1,#MSG1+14.    ;SEE IF DID LAST DATA CHAR YET
011426  103774                          BLO     3$              ;BR IF NOT
011430  052762  100400  177776          BIS     #CRCCHK!RXBCC,-2(R2) ;SET EXPECTED BCC
011436  012726  000160                  MOV     #160,(SP)+      ;LOAD HI CRC BYTE
011442  012726  000034                  MOV     #034,(SP)+      ;LOAD LO CRC BYTE
011446  012602                          MOV     (SP)+,R2        ;RESTORE R2
011450  012601                          MOV     (SP)+,R1        ;RESTORE R1
011452  000207                          RTS     PC              ;RETURN
```

```
                                          .SBTTL  GLOBAL ERROR REPORT SECTION

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


  011454    045     124     045   FMT1:   .ASCIZ  /%T%06%N/
  011457    117     066     045
  011462    116     000
  011464    045     116     045   FMT2:   .ASCIZ  /%N%AFAILING REG: /
  011467    101     106     101
  011472    111     114     111
  011475    116     107     040
  011500    122     105     107
  011503    072     040     000
  011506    045     101     105   FMT3:   .ASCIZ  /%AEXPECTED: %03%S5%AACTUAL: %03%N/
  011511    130     120     105
  011514    103     124     105
  011517    104     072     040
  011522    045     117     063
  011525    045     123     065
  011530    045     101     101
  011533    103     124     125
  011536    101     114     072
  011541    040     045     117
  011544    063     045     116
  011547    000
  011550    045     116     045   FMT4:   .ASCIZ  /%N%T%N%T%N/
  011553    124     045     116
  011556    045     124     045
  011561    116     000
  011563    045     117     063   FMT5:   .ASCIZ  /%03%S5%03%S5%03%S5%03%N/
  011566    045     123     065
  011571    045     117     063
  011574    045     123     065
  011577    045     117     063
  011602    045     123     065
  011605    045     117     063
  011610    045     116     000
  011613    045     123     064   FMT6:   .ASCIZ  /%S4%03%S5%03%S5%03%S5%03%N/
  011616    045     117     063
  011621    045     123     065
  011624    045     117     063
  011627    045     123     065
  011632    045     117     063
  011635    045     123     065
  011640    045     117     063
  011643    045     116     000
  011646    045     124     045   FMT7:   .ASCIZ  /%T%02%N/
  011651    117     062     045
  011654    116     000
  011656    045     101     105   FMT8:   .ASCIZ  /%AEXTENDED REG AX%01%A-%T%N/
  011661    130     124     105
  011664    116     104     105
  011667    104     040     122
```

L 7

```
        011672       105      107      040
        011675       101      130      045
        011700       117      061      045
        011703       101      055      045
        011706       124      045      116
        011711       000
     17 011712       045      124      045  FMT9:   .ASCIZ  /%T%N/
        011715       116      000
     18 011717       045      101      120  FMT10:  .ASCIZ  /%APC OF SUBR CALL: %06%N/
        011722       103      040      117
        011725       106      040      123
        011730       125      102      122
        011733       040      103      101
        011736       114      114      072
        011741       040      045      117
        011744       066      045      116
        011747       000
     19 011750       045      101      122  FMT11:  .ASCIZ  /%AREG %O2%A LOADED WITH: %O3%N/
        011753       105      107      040
        011756       045      117      062
        011761       045      101      040
        011764       114      117      101
        011767       104      105      104
        011772       040      127      111
        011775       124      110      072
        012000       040      045      117
        012003       063      045      116
        012006       000
     20 012007       045      116      045  FMT19:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
        012012       101      124      105
        012015       123      124      040
        012020       045      104      062
        012023       045      101      040
        012026       116      117      124
        012031       040      122      125
        012034       116      045      116
        012037       000
     21 012040       045      116      045  FMT24:  .ASCIZ  /%N%APLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON%N/
        012043       101      120      114
        012046       105      101      123
        012051       105      040      111
        012054       116      123      125
        012057       122      105      040
        012062       115      070      062
        012065       060      067      040
        012070       122      125      116
        012073       040      123      127
        012076       111      124      103
        012101       110      040      050
        012104       105      062      070
        012107       040      123      127
        012112       067      051      040
        012115       111      123      040
        012120       117      116      045
        012123       116      000
     22
     23
```

```
24
25 012125   103   123   122  EM1:    .ASCIZ  /CSR ADDRESS TIME-OUT (SEL0)/
   012130   040   101   104
   012133   104   122   105
   012136   123   123   040
   012141   124   111   115
   012144   105   055   117
   012147   125   124   040
   012152   050   123   105
   012155   114   060   051
   012160   000
26 012161   122   105   107  EM2:    .ASCIZ  /REG NOT INITIALIZED BY MST CLR/
   012164   040   116   117
   012167   124   040   111
   012172   116   111   124
   012175   111   101   114
   012200   111   132   105
   012203   104   040   102
   012206   131   040   115
   012211   123   124   040
   012214   103   114   122
   012217   000
27 012220   122   105   107  EM3:    .ASCIZ  /REG MISCOMPARE/
   012223   040   115   111
   012226   123   103   117
   012231   115   120   101
   012234   122   105   000
28 012237   122   105   107  EM4:    .ASCIZ  /REG NOT INITIALIZED BY UNIBUS RESET (INIT)/
   012242   040   116   117
   012245   124   040   111
   012250   116   111   124
   012253   111   101   114
   012256   111   132   105
   012261   104   040   102
   012264   131   040   125
   012267   116   111   102
   012272   125   123   040
   012275   122   105   123
   012300   105   124   040
   012303   050   111   116
   012306   111   124   051
   012311   000
29 012312   115   101   111  EM5:    .ASCIZ  /MAINT CLK BIT STUCK AT 0/
   012315   116   124   040
   012320   103   114   113
   012323   040   102   111
   012326   124   040   123
   012331   124   125   103
   012334   113   040   101
   012337   124   040   060
   012342   000
30 012343   115   101   111  EM6:    .ASCIZ  /MAINT CLK BIT STUCK AT 1/
   012346   116   124   040
   012351   103   114   113
   012354   040   102   111
   012357   124   040   123
   012362   124   125   103
```

```
        012365   113   040   101
        012370   124   040   061
        012373   000
     31 012374   117   122   104  EM7:    .ASCIZ  /ORDY NOT SET/
        012377   131   040   116
        012402   117   124   040
        012405   123   105   124
        012410   000
     32 012411   117   122   104  EM8:    .ASCIZ  /ORDY NOT CLEARED/
        012414   131   040   116
        012417   117   124   040
        012422   103   114   105
        012425   101   122   105
        012430   104   000
     33 012432   117   103   117  EM9:    .ASCIZ  /OCOR NOT SET/
        012435   122   040   116
        012440   117   124   040
        012443   123   105   124
        012446   000
     34 012447   117   103   117  EM10:   .ASCIZ  /OCOR NOT CLEARED/
        012452   122   040   116
        012455   117   124   040
        012460   103   114   105
        012463   101   122   105
        012466   104   000
     35 012470   117   101   103  EM11:   .ASCIZ  /OACT NOT SET/
        012473   124   040   116
        012476   117   124   040
        012501   123   105   124
        012504   000
     36 012505   117   101   103  EM12:   .ASCIZ  /OACT NOT CLEARED/
        012510   124   040   116
        012513   117   124   040
        012516   103   114   105
        012521   101   122   105
        012524   104   000
     37 012526   125   116   122  EM13:   .ASCIZ  /UNRR NOT CLEARED BY SOM/
        012531   122   040   116
        012534   117   124   040
        012537   103   114   105
        012542   101   122   105
        012545   104   040   102
        012550   131   040   123
        012553   117   115   000
     38 012556   125   116   122  EM14:   .ASCIZ  /UNRR NOT SET/
        012561   122   040   116
        012564   117   124   040
        012567   123   105   124
        012572   000
     39 012573   125   116   122  EM15:   .ASCIZ  /UNRR NOT CLEARED BY OC/
        012576   122   040   116
        012601   117   124   040
        012604   103   114   105
        012607   101   122   105
        012612   104   040   102
        012615   131   040   117
        012620   103   000
```

```
    40 012622     125     116     122  EM16:   .ASCIZ  /UNRR NOT CLEARED/
       012625     122     040     116
       012630     117     124     040
       012633     103     114     105
       012636     101     122     105
       012641     104     000
    41 012643     111     122     104  EM17:   .ASCIZ  /IRDY NOT SET/
       012646     131     040     116
       012651     117     124     040
       012654     123     105     124
       012657     000
    42 012660     111     122     104  EM18:   .ASCIZ  /IRDY NOT CLEARED/
       012663     131     040     116
       012666     117     124     040
       012671     103     114     105
       012674     101     122     105
       012677     104     000
    43 012701     111     103     111  EM19:   .ASCIZ  /ICIR NOT SET/
       012704     122     040     116
       012707     117     124     040
       012712     123     105     124
       012715     000
    44 012716     111     103     111  EM20:   .ASCIZ  /ICIR NOT CLEARED/
       012721     122     040     116
       012724     117     124     040
       012727     103     114     105
       012732     101     122     105
       012735     104     000
    45 012737     111     101     103  EM21:   .ASCIZ  /IACT NOT SET/
       012742     124     040     116
       012745     117     124     040
       012750     123     105     124
       012753     000
    46 012754     111     101     103  EM22:   .ASCIZ  /IACT NOT CLEARED/
       012757     124     040     116
       012762     117     124     040
       012765     103     114     105
       012770     101     122     105
       012773     104     000
    47 012775     104     123     123  EM23:   .ASCIZ  /DSSI NOT CLEARED/
       013000     111     040     116
       013003     117     124     040
       013006     103     114     105
       013011     101     122     105
       013014     104     000
    48 013016     104     123     123  EM24:   .ASCIZ  /DSSI NOT SET/
       013021     111     040     116
       013024     117     124     040
       013027     123     105     124
       013032     000
    49 013033     104     123     123  EM25:   .ASCIZ  /DSSI NOT CLEARED BY MST CLR/
       013036     111     040     116
       013041     117     124     040
       013044     103     114     105
       013047     101     122     105
       013052     104     040     102
       013055     131     040     115
```

```
        013060  123   124   040
        013063  103   114   122
        013066  000
     50 013067  111   116   103  EM26:   .ASCIZ  /INCORRECT DATA CHAR RCV'D/
        013072  117   122   122
        013075  105   103   124
        013100  040   104   101
        013103  124   101   040
        013106  103   110   101
        013111  122   040   122
        013114  103   126   047
        013117  104   000
     51 013121  111   116   103  EM27:   .ASCIZ  /INCORRECT CRC BYTE RCV'D/
        013124  117   122   122
        013127  105   103   124
        013132  040   103   122
        013135  103   040   102
        013140  131   124   105
        013143  040   122   103
        013146  126   047   104
        013151  000
     52 013152  122   123   117  EM28:   .ASCIZ  /RSOM NOT CLEARED/
        013155  115   040   116
        013160  117   124   040
        013163  103   114   105
        013166  101   122   105
        013171  104   000
     53 013173  122   123   117  EM29:   .ASCIZ  /RSOM NOT SET/
        013176  115   040   116
        013201  117   124   040
        013204  123   105   124
        013207  000
     54 013210  122   105   117  EM30:   .ASCIZ  /REOM NOT CLEARED/
        013213  115   040   116
        013216  117   124   040
        013221  103   114   105
        013224  101   122   105
        013227  104   000
     55 013231  122   105   117  EM31:   .ASCIZ  /REOM NOT SET/
        013234  115   040   116
        013237  117   124   040
        013242  123   105   124
        013245  000
     56 013246  124   130   104  EM32:   .ASCIZ  /TXDATA BIT NOT CLEARED/
        013251  101   124   101
        013254  040   102   111
        013257  124   040   116
        013262  117   124   040
        013265  103   114   105
        013270  101   122   105
        013273  104   000
     57 013275  124   130   104  EM33:   .ASCIZ  /TXDATA BIT NOT SET/
        013300  101   124   101
        013303  040   102   111
        013306  124   040   116
        013311  117   124   040
        013314  123   105   124
```

```
        013317   000
     58 013320   122     103     126   EM34:   .ASCIZ  /RCV'D DATA MISCOMPARE/
        013323   047     104     040
        013326   104     101     124
        013331   101     040     115
        013334   111     123     103
        013337   117     115     120
        013342   101     122     105
        013345   000
     59 013346   102     103     103   EM35:   .ASCIZ  /BCC NOT CLEARED/
        013351   040     116     117
        013354   124     040     103
        013357   114     105     101
        013362   122     105     104
        013365   000
     60 013366   102     103     103   EM36:   .ASCIZ  /BCC NOT SET/
        013371   040     116     117
        013374   124     040     123
        013377   105     124     000
     61 013402   105     102     114   EM37:   .ASCIZ  /EBLK NOT CLEARED/
        013405   113     040     116
        013410   117     124     040
        013413   103     114     105
        013416   101     122     105
        013421   104     000
     62 013423   105     102     114   EM38:   .ASCIZ  /EBLK NOT SET/
        013426   113     040     116
        013431   117     124     040
        013434   123     105     124
        013437   000
     63 013440   122     101     102   EM39:   .ASCIZ  /RAB NOT CLEARED/
        013443   040     116     117
        013446   124     040     103
        013451   114     105     101
        013454   122     105     104
        013457   000
     64 013460   122     101     102   EM40:   .ASCIZ  /RAB NOT SET/
        013463   040     116     117
        013466   124     040     123
        013471   105     124     000
     65 013474   117     126     122   EM41:   .ASCIZ  /OVRR NOT CLEARED/
        013477   122     040     116
        013502   117     124     040
        013505   103     114     105
        013510   101     122     105
        013513   104     000
     66 013515   117     126     122   EM42:   .ASCIZ  /OVRR NOT SET/
        013520   122     040     116
        013523   117     124     040
        013526   123     105     124
        013531   000
     67 013532   123     127     040   EM43:   .ASCIZ  /SW PACK #1 INCORRECT/
        013535   120     101     103
        013540   113     040     043
        013543   061     040     111
        013546   116     103     117
        013551   122     122     105
```

```
     013554   103   124   000
  68 013557   123   127   040   EM44:   .ASCIZ  /SW PACK #2 INCORRECT/
     013562   120   101   103
     013565   113   040   043
     013570   062   040   111
     013573   116   103   117
     013576   122   122   105
     013601   103   124   000
  69 013604   123   127   040   EM45:   .ASCIZ  /SW PACK #3 INCORRECT/
     013607   120   101   103
     013612   113   040   043
     013615   063   040   111
     013620   116   103   117
     013623   122   122   105
     013626   103   124   000
  70 013631   122   103   126   EM46:   .ASCIZ  /RCV SILO NOT CLEARED BY IC/
     013634   040   123   111
     013637   114   117   040
     013642   116   117   124
     013645   040   103   114
     013650   105   101   122
     013653   105   104   040
     013656   102   131   040
     013661   111   103   000
  71 013664   101   123   123   EM47:   .ASCIZ  /ASSEMB BIT COUNT INCORRECT/
     013667   105   115   102
     013672   040   102   111
     013675   124   040   103
     013700   117   125   116
     013703   124   040   111
     013706   116   103   117
     013711   122   122   105
     013714   103   124   000
  72 013717   117   104   104   EM48:   .ASCIZ  /ODD VRC PARITY BIT NOT SET/
     013722   040   126   122
     013725   103   040   120
     013730   101   122   111
     013733   124   131   040
     013736   102   111   124
     013741   040   116   117
     013744   124   040   123
     013747   105   124   000
  73 013752   117   104   104   EM49:   .ASCIZ  /ODD VRC PARITY BIT NOT CLEARED/
     013755   040   126   122
     013760   103   040   120
     013763   101   122   111
     013766   124   131   040
     013771   102   111   124
     013774   040   116   117
     013777   124   040   103
     014002   114   105   101
     014005   122   105   104
     014010   000
  74 014011   105   126   105   EM50:   .ASCIZ  /EVEN VRC PARITY BIT NOT SET/
     014014   116   040   126
     014017   122   103   040
     014022   120   101   122
```

```
   014025   111   124   131
   014030   040   102   111
   014033   124   040   116
   014036   117   124   040
   014041   123   105   124
   014044   000
   014045   105   126   105   EM51:   .ASCIZ  /EVEN VRC PARITY BIT NOT CLEARED/
   014050   116   040   126
   014053   122   103   040
   014056   120   101   122
   014061   111   124   131
   014064   040   102   111
   014067   124   040   116
   014072   117   124   040
   014075   103   114   105
   014100   101   122   105
   014103   104   000
   014105   122   105   101   EM52:   .ASCIZ  /READY NOT SET AFTER AX REG WRITE/
   014110   104   131   040
   014113   116   117   124
   014116   040   123   105
   014121   124   040   101
   014124   106   124   105
   014127   122   040   101
   014132   130   040   122
   014135   105   107   040
   014140   127   122   111
   014143   124   105   000
   014146   122   105   101   EM53:   .ASCIZ  /READY NOT SET AFTER AX REG READ/
   014151   104   131   040
   014154   116   117   124
   014157   040   123   105
   014162   124   040   101
   014165   106   124   105
   014170   122   040   101
   014173   130   040   122
   014176   105   107   040
   014201   122   105   101
   014204   104   000
   014206   124   130   040   EM54:   .ASCIZ  /TX UNDERRUN ERROR/
   014211   125   116   104
   014214   105   122   122
   014217   125   116   040
   014222   105   122   122
   014225   117   122   000
   014230   122   124   123   EM60:   .ASCIZ  /RTS NOT SET/
   014233   040   116   117
   014236   124   040   123
   014241   105   124   000
   014244   122   124   123   EM65:   .ASCIZ  /RTS NOT CLEARED/
   014247   040   116   117
   014252   124   040   103
   014255   114   105   101
   014260   122   105   104
   014263   000
```

```
 83
 84 014264    111    116    102  DH1:     .ASCIZ  &INBUS/OUTBUS REG &
    014267    125    123    057
    014272    117    125    124
    014275    102    125    123
    014300    040    122    105
    014303    107    040    000
 85 014306    114    111    116  DH2:     .ASCIZ  /LINE UNIT INBUS REGS :/
    014311    105    040    125
    014314    116    111    124
    014317    040    111    116
    014322    102    125    123
    014325    040    122    105
    014330    107    123    040
    014333    072    000
 86 014335    122    105    107  DH3:     .ASCIZ  /REG10   REG11   REG12   REG13/
    014340    061    060    040
    014343    040    040    122
    014346    105    107    061
    014351    061    040    040
    014354    040    122    105
    014357    107    061    062
    014362    040    040    040
    014365    122    105    107
    014370    061    063    000
 87 014373    040    040    040  DH4:     .ASCIZ  /    REG14   REG15   REG16   REG17/
    014376    040    122    105
    014401    107    061    064
    014404    040    040    040
    014407    122    105    107
    014412    061    065    040
    014415    040    040    122
    014420    105    107    061
    014423    066    040    040
    014426    040    122    105
    014431    107    061    067
    014434    000
 88 014435    061    065    000  DH5:     .ASCIZ  /15/
 89 014440    061    066    000  DH6:     .ASCIZ  /16/
 90 014443    114    111    116  DH7:     .ASCIZ  /LINE UNIT EXTENDED REGS :/
    014446    105    040    125
    014451    116    111    124
    014454    040    105    130
    014457    124    105    116
    014462    104    105    104
    014465    040    122    105
    014470    107    123    040
    014473    072    000
 91 014475    101    130    060  DH8:     .ASCIZ  /AX0-15  AX0-16  AX1-15  AX1-16/
    014500    055    061    065
    014503    040    040    101
    014506    130    060    055
    014511    061    066    040
    014514    040    101    130
    014517    061    055    061
    014522    065    040    040
    014525    101    130    061
```

```
      014530    055     061     066
      014533    000
   92 014534    040     040     040  DH9:     .ASCIZ  /     AX2-15  AX2-16  AX3-15  AX3-16/
      014537    040     101     130
      014542    062     055     061
      014545    065     040     040
      014550    101     130     062
      014553    055     061     066
      014556    040     040     101
      014561    130     063     055
      014564    061     065     040
      014567    040     101     130
      014572    063     055     061
      014575    066     000
   93
   94                                .EVEN
   95
   96
   97
   98
   99
  100 014600                                  BGNMSG  ERR1
      014600                                                                  ERR1::
  101 014600                                  PRINTB  #FMT1,#ADDRES,MPCSR
      014600  013746  002446                                                          MOV     MPCSR,-(SP)
      014604  012746  035620                                                          MOV     #ADDRES,-(SP)
      014610  012746  011454                                                          MOV     #FMT1,-(SP)
      014614  012746  000003                                                          MOV     #3,-(SP)
      014620  010600                                                                  MOV     SP,R0
      014622  104414                                                                  TRAP    C$PNTB
      014624  062706  000010                                                          ADD     #10,SP
  102 014630                                  ENDMSG
      014630                                                                  L10002:
      014630  104423                                                                  TRAP    C$MSG
  103
  104
  105
  106 014632                                  BGNMSG  ERR2
      014632                                                                  ERR2::
  107 014632                                  PRINTB  #FMT1,#ADDRES,MPCSR
      014632  013746  002446                                                          MOV     MPCSR,-(SP)
      014636  012746  035620                                                          MOV     #ADDRES,-(SP)
      014642  012746  011454                                                          MOV     #FMT1,-(SP)
      014646  012746  000003                                                          MOV     #3,-(SP)
      014652  010600                                                                  MOV     SP,R0
      014654  104414                                                                  TRAP    C$PNTB
      014656  062706  000010                                                          ADD     #10,SP
  108 014662                                  PRINTB  #FMT2
      014662  012746  011464                                                          MOV     #FMT2,-(SP)
      014666  012746  000001                                                          MOV     #1,-(SP)
      014672  010600                                                                  MOV     SP,R0
      014674  104414                                                                  TRAP    C$PNTB
      014676  062706  000004                                                          ADD     #4,SP
  109 014702                                  PRINTB  #FMT7,#DH1,REGNUM
      014702  013746  002400                                                          MOV     REGNUM,-(SP)
      014706  012746  014264                                                          MOV     #DH1,-(SP)
      014712  012746  011646                                                          MOV     #FMT7,-(SP)
```

```
    014716  012746  000003                                        MOV     #3,-(SP)
    014722  010600                                                MOV     SP,R0
    014724  104414                                                TRAP    C$PNTB
    014726  062706  000010                                        ADD     #10,SP
110 014732                      PRINTB  #FMT3,GOODAT,BADDAT
    014732  013746  002406                                        MOV     BADDAT,-(SP)
    014736  013746  002404                                        MOV     GOODAT,-(SP)
    014742  012746  011506                                        MOV     #FMT3,-(SP)
    014746  012746  000003                                        MOV     #3,-(SP)
    014752  010600                                                MOV     SP,R0
    014754  104414                                                TRAP    C$PNTB
    014756  062706  000010                                        ADD     #10,SP
111 014762                      PRINTX  #FMT4,#DH2,#DH3
    014762  012746  014335                                        MOV     #DH3,-(SP)
    014766  012746  014306                                        MOV     #DH2,-(SP)
    014772  012746  011550                                        MOV     #FMT4,-(SP)
    014776  012746  000003                                        MOV     #3,-(SP)
    015002  010600                                                MOV     SP,R0
    015004  104415                                                TRAP    C$PNTX
    015006  062706  000010                                        ADD     #10,SP
112 015012                      PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
    015012  013746  002310                                        MOV     LUR13,-(SP)
    015016  013746  002306                                        MOV     LUR12,-(SP)
    015022  013746  002304                                        MOV     LUR11,-(SP)
    015026  013746  002302                                        MOV     LUR10,-(SP)
    015032  012746  011563                                        MOV     #FMT5,-(SP)
    015036  012746  000005                                        MOV     #5,-(SP)
    015042  010600                                                MOV     SP,R0
    015044  104415                                                TRAP    C$PNTX
    015046  062706  000014                                        ADD     #14,SP
113 015052                      PRINTX  #FMT9,#DH4
    015052  012746  014373                                        MOV     #DH4,-(SP)
    015056  012746  011712                                        MOV     #FMT9,-(SP)
    015062  012746  000002                                        MOV     #2,-(SP)
    015066  010600                                                MOV     SP,R0
    015070  104415                                                TRAP    C$PNTX
    015072  062706  000006                                        ADD     #6,SP
114 015076                      PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
    015076  013746  002320                                        MOV     LUR17,-(SP)
    015102  013746  002316                                        MOV     LUR16,-(SP)
    015106  013746  002314                                        MOV     LUR15,-(SP)
    015112  013746  002312                                        MOV     LUR14,-(SP)
    015116  012746  011613                                        MOV     #FMT6,-(SP)
    015122  012746  000005                                        MOV     #5,-(SP)
    015126  010600                                                MOV     SP,R0
    015130  104415                                                TRAP    C$PNTX
    015132  062706  000014                                        ADD     #14,SP
115 015136                      ENDMSG
    015136                                              L10003:
    015136  104423                                                TRAP    C$MSG
116
117
118
119
120
121 015140                      BGNMSG  ERR3
    015140                                              ERR3::
```

```
    122 015140                              PRINTB  #FMT1,#ADDRES,MPCSR
        015140  013746  002446                                              MOV     MPCSR,-(SP)
        015144  012746  035620                                              MOV     #ADDRES,-(SP)
        015150  012746  011454                                              MOV     #FMT1,-(SP)
        015154  012746  000003                                              MOV     #3,-(SP)
        015160  010600                                                      MOV     SP,R0
        015162  104414                                                      TRAP    C$PNTB
        015164  062706  000010                                              ADD     #10,SP
    123 015170                              PRINTB  #FMT2
        015170  012746  011464                                              MOV     #FMT2,-(SP)
        015174  012746  000001                                              MOV     #1,-(SP)
        015200  010600                                                      MOV     SP,R0
        015202  104414                                                      TRAP    C$PNTB
        015204  062706  000004                                              ADD     #4,SP
    124 015210                              PRINTB  #FMT8,TMP1,TMP0
        015210  013746  002530                                              MOV     TMP0,-(SP)
        015214  013746  002532                                              MOV     TMP1,-(SP)
        015220  012746  011656                                              MOV     #FMT8,-(SP)
        015224  012746  000003                                              MOV     #3,-(SP)
        015230  010600                                                      MOV     SP,R0
        015232  104414                                                      TRAP    C$PNTB
        015234  062706  000010                                              ADD     #10,SP
    125 015240                              PRINTB  #FMT3,GOODAT,BADDAT
        015240  013746  002406                                              MOV     BADDAT,-(SP)
        015244  013746  002404                                              MOV     GOODAT,-(SP)
        015250  012746  011506                                              MOV     #FMT3,-(SP)
        015254  012746  000003                                              MOV     #3,-(SP)
        015260  010600                                                      MOV     SP,R0
        015262  104414                                                      TRAP    C$PNTB
        015264  062706  000010                                              ADD     #10,SP
    126 015270                              PRINTX  #FMT4,#DH2,#DH3
        015270  012746  014335                                              MOV     #DH3,-(SP)
        015274  012746  014306                                              MOV     #DH2,-(SP)
        015300  012746  011550                                              MOV     #FMT4,-(SP)
        015304  012746  000003                                              MOV     #3,-(SP)
        015310  010600                                                      MOV     SP,R0
        015312  104415                                                      TRAP    C$PNTX
        015314  062706  000010                                              ADD     #10,SP
    127 015320                              PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
        015320  013746  002310                                              MOV     LUR13,-(SP)
        015324  013746  002306                                              MOV     LUR12,-(SP)
        015330  013746  002304                                              MOV     LUR11,-(SP)
        015334  013746  002302                                              MOV     LUR10,-(SP)
        015340  012746  011563                                              MOV     #FMT5,-(SP)
        015344  012746  000005                                              MOV     #5,-(SP)
        015350  010600                                                      MOV     SP,R0
        015352  104415                                                      TRAP    C$PNTX
        015354  062706  000014                                              ADD     #14,SP
    128 015360                              PRINTX  #FMT9,#DH4
        015360  012746  014373                                              MOV     #DH4,-(SP)
        015364  012746  011712                                              MOV     #FMT9,-(SP)
        015370  012746  000002                                              MOV     #2,-(SP)
        015374  010600                                                      MOV     SP,R0
        015376  104415                                                      TRAP    C$PNTX
        015400  062706  000006                                              ADD     #6,SP
    129 015404                              PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
        015404  013746  002320                                              MOV     LUR17,-(SP)
```

```
        015410  013746  002316                                          MOV     LUR16,-(SP)
        015414  013746  002314                                          MOV     LUR15,-(SP)
        015420  013746  002312                                          MOV     LUR14,-(SP)
        015424  012746  011613                                          MOV     #FMT6,-(SP)
        015430  012746  000005                                          MOV     #5,-(SP)
        015434  010600                                                  MOV     SP,R0
        015436  104415                                                  TRAP    C$PNTX
        015440  062706  000014                                          ADD     #14,SP
130     015444                  PRINTX  #FMT4,#DH7,#DH8
        015444  012746  014475                                          MOV     #DH8,-(SP)
        015450  012746  014443                                          MOV     #DH7,-(SP)
        015454  012746  011550                                          MOV     #FMT4,-(SP)
        015460  012746  000003                                          MOV     #3,-(SP)
        015464  010600                                                  MOV     SP,R0
        015466  104415                                                  TRAP    C$PNTX
        015470  062706  000010                                          ADD     #10,SP
131     015474                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
        015474  013746  002330                                          MOV     AX1.16,-(SP)
        015500  013746  002326                                          MOV     AX1.15,-(SP)
        015504  013746  002324                                          MOV     AX0.16,-(SP)
        015510  013746  002322                                          MOV     AX0.15,-(SP)
        015514  012746  011563                                          MOV     #FMT5,-(SP)
        015520  012746  000005                                          MOV     #5,-(SP)
        015524  010600                                                  MOV     SP,R0
        015526  104415                                                  TRAP    C$PNTX
        015530  062706  000014                                          ADD     #14,SP
132     015534                  PRINTX  #FMT9,#DH9
        015534  012746  014534                                          MOV     #DH9,-(SP)
        015540  012746  011712                                          MOV     #FMT9,-(SP)
        015544  012746  000002                                          MOV     #2,-(SP)
        015550  010600                                                  MOV     SP,R0
        015552  104415                                                  TRAP    C$PNTX
        015554  062706  000006                                          ADD     #6,SP
133     015560                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
        015560  013746  002340                                          MOV     AX3.16,-(SP)
        015564  013746  002336                                          MOV     AX3.15,-(SP)
        015570  013746  002334                                          MOV     AX2.16,-(SP)
        015574  013746  002332                                          MOV     AX2.15,-(SP)
        015600  012746  011613                                          MOV     #FMT6,-(SP)
        015604  012746  000005                                          MOV     #5,-(SP)
        015610  010600                                                  MOV     SP,R0
        015612  104415                                                  TRAP    C$PNTX
        015614  062706  000014                                          ADD     #14,SP
134     015620                  ENDMSG
        015620                                                  L10004:
        015620  104423                                                  TRAP    C$MSG
135
136
137
138
139
140     015622                  BGNMSG  ERR4
        015622                                                  ERR4::
141     015622                  PRINTB  #FMT10,SUBRPC
        015622  013746  002352                                          MOV     SUBRPC,-(SP)
        015626  012746  011717                                          MOV     #FMT10,-(SP)
        015632  012746  000002                                          MOV     #2,-(SP)
```

```
     015636  010600                                              MOV     SP,R0
     015640  104414                                              TRAP    C$PNTB
     015642  062706  000006                                      ADD     #6,SP
 142 015646                        PRINTB  #FMT1,#ADDRES,MPCSR
     015646  013746  002446                                      MOV     MPCSR,-(SP)
     015652  012746  035620                                      MOV     #ADDRES,-(SP)
     015656  012746  011454                                      MOV     #FMT1,-(SP)
     015662  012746  000003                                      MOV     #3,-(SP)
     015666  010600                                              MOV     SP,R0
     015670  104414                                              TRAP    C$PNTB
     015672  062706  000010                                      ADD     #10,SP
 143 015676                        PRINTB  #FMT2
     015676  012746  011464                                      MOV     #FMT2,-(SP)
     015702  012746  000001                                      MOV     #1,-(SP)
     015706  010600                                              MOV     SP,R0
     015710  104414                                              TRAP    C$PNTB
     015712  062706  000004                                      ADD     #4,SP
 144 015716                        PRINTB  #FMT7,#DH1,REGNUM
     015716  013746  002400                                      MOV     REGNUM,-(SP)
     015722  012746  014264                                      MOV     #DH1,-(SP)
     015726  012746  011646                                      MOV     #FMT7,-(SP)
     015732  012746  000003                                      MOV     #3,-(SP)
     015736  010600                                              MOV     SP,R0
     015740  104414                                              TRAP    C$PNTB
     015742  062706  000010                                      ADD     #10,SP
 145 015746                        PRINTX  #FMT4,#DH2,#DH3
     015746  012746  014335                                      MOV     #DH3,-(SP)
     015752  012746  014306                                      MOV     #DH2,-(SP)
     015756  012746  011550                                      MOV     #FMT4,-(SP)
     015762  012746  000003                                      MOV     #3,-(SP)
     015766  010600                                              MOV     SP,R0
     015770  104415                                              TRAP    C$PNTX
     015772  062706  000010                                      ADD     #10,SP
 146 015776                        PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
     015776  013746  002310                                      MOV     LUR13,-(SP)
     016002  013746  002306                                      MOV     LUR12,-(SP)
     016006  013746  002304                                      MOV     LUR11,-(SP)
     016012  013746  002302                                      MOV     LUR10,-(SP)
     016016  012746  011563                                      MOV     #FMT5,-(SP)
     016022  012746  000005                                      MOV     #5,-(SP)
     016026  010600                                              MOV     SP,R0
     016030  104415                                              TRAP    C$PNTX
     016032  062706  000014                                      ADD     #14,SP
 147 016036                        PRINTX  #FMT9,#DH4
     016036  012746  014373                                      MOV     #DH4,-(SP)
     016042  012746  011712                                      MOV     #FMT9,-(SP)
     016046  012746  000002                                      MOV     #2,-(SP)
     016052  010600                                              MOV     SP,R0
     016054  104415                                              TRAP    C$PNTX
     016056  062706  000006                                      ADD     #6,SP
 148 016062                        PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
     016062  013746  002320                                      MOV     LUR17,-(SP)
     016066  013746  002316                                      MOV     LUR16,-(SP)
     016072  013746  002314                                      MOV     LUR15,-(SP)
     016076  013746  002312                                      MOV     LUR14,-(SP)
     016102  012746  011613                                      MOV     #FMT6,-(SP)
     016106  012746  000005                                      MOV     #5,-(SP)
```

```
    016112  010600                                                  MOV     SP,R0
    016114  104415                                                  TRAP    C$PNTX
    016116  062706  000014                                          ADD     #14,SP
    016122                  PRINTX  #FMT4,#DH7,#DH8
    016122  012746  014475                                          MOV     #DH8,-(SP)
    016126  012746  014443                                          MOV     #DH7,-(SP)
    016132  012746  011550                                          MOV     #FMT4,-(SP)
    016136  012746  000003                                          MOV     #3,-(SP)
    016142  010600                                                  MOV     SP,R0
    016144  104415                                                  TRAP    C$PNTX
    016146  062706  000010                                          ADD     #10,SP
    016152                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
    016152  013746  002330                                          MOV     AX1.16,-(SP)
    016156  013746  002326                                          MOV     AX1.15,-(SP)
    016162  013746  002324                                          MOV     AX0.16,-(SP)
    016166  013746  002322                                          MOV     AX0.15,-(SP)
    016172  012746  011563                                          MOV     #FMT5,-(SP)
    016176  012746  000005                                          MOV     #5,-(SP)
    016202  010600                                                  MOV     SP,R0
    016204  104415                                                  TRAP    C$PNTX
    016206  062706  000014                                          ADD     #14,SP
    016212                  PRINTX  #FMT9,#DH9
    016212  012746  014534                                          MOV     #DH9,-(SP)
    016216  012746  011712                                          MOV     #FMT9,-(SP)
    016222  012746  000002                                          MOV     #2,-(SP)
    016226  010600                                                  MOV     SP,R0
    016230  104415                                                  TRAP    C$PNTX
    016232  062706  000006                                          ADD     #6,SP
    016236                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
    016236  013746  002340                                          MOV     AX3.16,-(SP)
    016242  013746  002336                                          MOV     AX3.15,-(SP)
    016246  013746  002334                                          MOV     AX2.16,-(SP)
    016252  013746  002332                                          MOV     AX2.15,-(SP)
    016256  012746  011613                                          MOV     #FMT6,-(SP)
    016262  012746  000005                                          MOV     #5,-(SP)
    016266  010600                                                  MOV     SP,R0
    016270  104415                                                  TRAP    C$PNTX
    016272  062706  000014                                          ADD     #14,SP
    016276                  ENDMSG
    016276                                                  L10005:
    016276  104423                                                  TRAP    C$MSG
    016300                  BGNMSG  ERR5
    016300                                                  ERR5::
    016300                  PRINTB  #FMT1,#ADDRES,MPCSR
    016300  013746  002446                                          MOV     MPCSR,-(SP)
    016304  012746  035620                                          MOV     #ADDRES,-(SP)
    016310  012746  011454                                          MOV     #FMT1,-(SP)
    016314  012746  000003                                          MOV     #3,-(SP)
    016320  010600                                                  MOV     SP,R0
    016322  104414                                                  TRAP    C$PNTB
    016324  062706  000010                                          ADD     #10,SP
    016330                  PRINTB  #FMT11,REGNUM,LOADAT
```

```
      016330  013746  002410                                          MOV     LOADAT,-(SP)
      016334  013746  002400                                          MOV     REGNUM,-(SP)
      016340  012746  011750                                          MOV     #FMT11,-(SP)
      016344  012746  000003                                          MOV     #3,-(SP)
      016350  010600                                                  MOV     SP,R0
      016352  104414                                                  TRAP    C$PNTB
      016354  062706  000010                                          ADD     #10,SP
  162 016360                          PRINTB  #FMT2
      016360  012746  011464                                          MOV     #FMT2,-(SP)
      016364  012746  000001                                          MOV     #1,-(SP)
      016370  010600                                                  MOV     SP,R0
      016372  104414                                                  TRAP    C$PNTB
      016374  062706  000004                                          ADD     #4,SP
  163 016400                          PRINTB  #FMT8,TMP1,TMP0
      016400  013746  002530                                          MOV     TMP0,-(SP)
      016404  013746  002532                                          MOV     TMP1,-(SP)
      016410  012746  011656                                          MOV     #FMT8,-(SP)
      016414  012746  000003                                          MOV     #3,-(SP)
      016420  010600                                                  MOV     SP,R0
      016422  104414                                                  TRAP    C$PNTB
      016424  062706  000010                                          ADD     #10,SP
  164 016430                          PRINTB  #FMT3,GOODAT,BADDAT
      016430  013746  002406                                          MOV     BADDAT,-(SP)
      016434  013746  002404                                          MOV     GOODAT,-(SP)
      016440  012746  011506                                          MOV     #FMT3,-(SP)
      016444  012746  000003                                          MOV     #3,-(SP)
      016450  010600                                                  MOV     SP,R0
      016452  104414                                                  TRAP    C$PNTB
      016454  062706  000010                                          ADD     #10,SP
  165 016460                          PRINTX  #FMT4,#DH2,#DH3
      016460  012746  014335                                          MOV     #DH3,-(SP)
      016464  012746  014306                                          MOV     #DH2,-(SP)
      016470  012746  011550                                          MOV     #FMT4,-(SP)
      016474  012746  000003                                          MOV     #3,-(SP)
      016500  010600                                                  MOV     SP,R0
      016502  104415                                                  TRAP    C$PNTX
      016504  062706  000010                                          ADD     #10,SP
  166 016510                          PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
      016510  013746  002310                                          MOV     LUR13,-(SP)
      016514  013746  002306                                          MOV     LUR12,-(SP)
      016520  013746  002304                                          MOV     LUR11,-(SP)
      016524  013746  002302                                          MOV     LUR10,-(SP)
      016530  012746  011563                                          MOV     #FMT5,-(SP)
      016534  012746  000005                                          MOV     #5,-(SP)
      016540  010600                                                  MOV     SP,R0
      016542  104415                                                  TRAP    C$PNTX
      016544  062706  000014                                          ADD     #14,SP
  167 016550                          PRINTX  #FMT9,#DH4
      016550  012746  014373                                          MOV     #DH4,-(SP)
      016554  012746  011712                                          MOV     #FMT9,-(SP)
      016560  012746  000002                                          MOV     #2,-(SP)
      016564  010600                                                  MOV     SP,R0
      016566  104415                                                  TRAP    C$PNTX
      016570  062706  000006                                          ADD     #6,SP
  168 016574                          PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
      016574  013746  002320                                          MOV     LUR17,-(SP)
      016600  013746  002316                                          MOV     LUR16,-(SP)
```

```
        016604  013746  002314                                          MOV     LUR15,-(SP)
        016610  013746  002312                                          MOV     LUR14,-(SP)
        016614  012746  011613                                          MOV     #FMT6,-(SP)
        016620  012746  000005                                          MOV     #5,-(SP)
        016624  010600                                                  MOV     SP,R0
        016626  104415                                                  TRAP    C$PNTX
        016630  062706  000014                                          ADD     #14,SP
    169 016634                  PRINTX  #FMT4,#DH7,#DH8
        016634  012746  014475                                          MOV     #DH8,-(SP)
        016640  012746  014443                                          MOV     #DH7,-(SP)
        016644  012746  011550                                          MOV     #FMT4,-(SP)
        016650  012746  000003                                          MOV     #3,-(SP)
        016654  010600                                                  MOV     SP,R0
        016656  104415                                                  TRAP    C$PNTX
        016660  062706  000010                                          ADD     #10,SP
    170 016664                  PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
        016664  013746  002330                                          MOV     AX1.16,-(SP)
        016670  013746  002326                                          MOV     AX1.15,-(SP)
        016674  013746  002324                                          MOV     AX0.16,-(SP)
        016700  013746  002322                                          MOV     AX0.15,-(SP)
        016704  012746  011563                                          MOV     #FMT5,-(SP)
        016710  012746  000005                                          MOV     #5,-(SP)
        016714  010600                                                  MOV     SP,R0
        016716  104415                                                  TRAP    C$PNTX
        016720  062706  000014                                          ADD     #14,SP
    171 016724                  PRINTX  #FMT9,#DH9
        016724  012746  014534                                          MOV     #DH9,-(SP)
        016730  012746  011712                                          MOV     #FMT9,-(SP)
        016734  012746  000002                                          MOV     #2,-(SP)
        016740  010600                                                  MOV     SP,R0
        016742  104415                                                  TRAP    C$PNTX
        016744  062706  000006                                          ADD     #6,SP
    172 016750                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
        016750  013746  002340                                          MOV     AX3.16,-(SP)
        016754  013746  002336                                          MOV     AX3.15,-(SP)
        016760  013746  002334                                          MOV     AX2.16,-(SP)
        016764  013746  002332                                          MOV     AX2.15,-(SP)
        016770  012746  011613                                          MOV     #FMT6,-(SP)
        016774  012746  000005                                          MOV     #5,-(SP)
        017000  010600                                                  MOV     SP,R0
        017002  104415                                                  TRAP    C$PNTX
        017004  062706  000014                                          ADD     #14,SP
    173 017010                  ENDMSG
        017010                                                  L10006:
        017010  104423                                                  TRAP    C$MSG
    174
    175
    176
    177
    178
    179 017012                  BGNMSG  ERR6
        017012                                                  ERR6::
    180 017012                  PRINTB  #FMT10,SUBRPC
        017012  013746  002352                                          MOV     SUBRPC,-(SP)
        017016  012746  011717                                          MOV     #FMT10,-(SP)
        017022  012746  000002                                          MOV     #2,-(SP)
        017026  010600                                                  MOV     SP,R0
```

```
     017030  104414                                                          TRAP    C$PNTB
     017032  062706  000006                                                  ADD     #6,SP
 181 017036                             PRINTB  #FMT1,#ADDRES,MPCSR
     017036  013746  002446                                                  MOV     MPCSR,-(SP)
     017042  012746  035620                                                  MOV     #ADDRES,-(SP)
     017046  012746  011454                                                  MOV     #FMT1,-(SP)
     017052  012746  000003                                                  MOV     #3,-(SP)
     017056  010600                                                          MOV     SP,R0
     017060  104414                                                          TRAP    C$PNTB
     017062  062706  000010                                                  ADD     #10,SP
 182 017066                             PRINTB  #FMT2
     017066  012746  011464                                                  MOV     #FMT2,-(SP)
     017072  012746  000001                                                  MOV     #1,-(SP)
     017076  010600                                                          MOV     SP,R0
     017100  104414                                                          TRAP    C$PNTB
     017102  062706  000004                                                  ADD     #4,SP
 183 017106                             PRINTB  #FMT8,TMP1,TMP0
     017106  013746  002530                                                  MOV     TMP0,-(SP)
     017112  013746  002532                                                  MOV     TMP1,-(SP)
     017116  012746  011656                                                  MOV     #FMT8,-(SP)
     017122  012746  000003                                                  MOV     #3,-(SP)
     017126  010600                                                          MOV     SP,R0
     017130  104414                                                          TRAP    C$PNTB
     017132  062706  000010                                                  ADD     #10,SP
 184 017136                             PRINTX  #FMT4,#DH2,#DH3
     017136  012746  014335                                                  MOV     #DH3,-(SP)
     017142  012746  014306                                                  MOV     #DH2,-(SP)
     017146  012746  011550                                                  MOV     #FMT4,-(SP)
     017152  012746  000003                                                  MOV     #3,-(SP)
     017156  010600                                                          MOV     SP,R0
     017160  104415                                                          TRAP    C$PNTX
     017162  062706  000010                                                  ADD     #10,SP
 185 017166                             PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
     017166  013746  002310                                                  MOV     LUR13,-(SP)
     017172  013746  002306                                                  MOV     LUR12,-(SP)
     017176  013746  002304                                                  MOV     LUR11,-(SP)
     017202  013746  002302                                                  MOV     LUR10,-(SP)
     017206  012746  011563                                                  MOV     #FMT5,-(SP)
     017212  012746  000005                                                  MOV     #5,-(SP)
     017216  010600                                                          MOV     SP,R0
     017220  104415                                                          TRAP    C$PNTX
     017222  062706  000014                                                  ADD     #14,SP
 186 017226                             PRINTX  #FMT9,#DH4
     017226  012746  014373                                                  MOV     #DH4,-(SP)
     017232  012746  011712                                                  MOV     #FMT9,-(SP)
     017236  012746  000002                                                  MOV     #2,-(SP)
     017242  010600                                                          MOV     SP,R0
     017244  104415                                                          TRAP    C$PNTX
     017246  062706  000006                                                  ADD     #6,SP
 187 017252                             PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
     017252  013746  002320                                                  MOV     LUR17,-(SP)
     017256  013746  002316                                                  MOV     LUR16,-(SP)
     017262  013746  002314                                                  MOV     LUR15,-(SP)
     017266  013746  002312                                                  MOV     LUR14,-(SP)
     017272  012746  011613                                                  MOV     #FMT6,-(SP)
     017276  012746  000005                                                  MOV     #5,-(SP)
     017302  010600                                                          MOV     SP,R0
```

```
      017304  104415                                                           TRAP    C$PNTX
      017306  062706  000014                                                   ADD     #14,SP
  188 017312                          PRINTX  #FMT4,#DH7,#DH8
      017312  012746  014475                                                   MOV     #DH8,-(SP)
      017316  012746  014443                                                   MOV     #DH7,-(SP)
      017322  012746  011550                                                   MOV     #FMT4,-(SP)
      017326  012746  000003                                                   MOV     #3,-(SP)
      017332  010600                                                           MOV     SP,R0
      017334  104415                                                           TRAP    C$PNTX
      017336  062706  000010                                                   ADD     #10,SP
  189 017342                          PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
      017342  013746  002330                                                   MOV     AX1.16,-(SP)
      017346  013746  002326                                                   MOV     AX1.15,-(SP)
      017352  013746  002324                                                   MOV     AX0.16,-(SP)
      017356  013746  002322                                                   MOV     AX0.15,-(SP)
      017362  012746  011563                                                   MOV     #FMT5,-(SP)
      017366  012746  000005                                                   MOV     #5,-(SP)
      017372  010600                                                           MOV     SP,R0
      017374  104415                                                           TRAP    C$PNTX
      017376  062706  000014                                                   ADD     #14,SP
  190 017402                          PRINTX  #FMT9,#DH9
      017402  012746  014534                                                   MOV     #DH9,-(SP)
      017406  012746  011712                                                   MOV     #FMT9,-(SP)
      017412  012746  000002                                                   MOV     #2,-(SP)
      017416  010600                                                           MOV     SP,R0
      017420  104415                                                           TRAP    C$PNTX
      017422  062706  000006                                                   ADD     #6,SP
  191 017426                          PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
      017426  013746  002340                                                   MOV     AX3.16,-(SP)
      017432  013746  002336                                                   MOV     AX3.15,-(SP)
      017436  013746  002334                                                   MOV     AX2.16,-(SP)
      017442  013746  002332                                                   MOV     AX2.15,-(SP)
      017446  012746  011613                                                   MOV     #FMT6,-(SP)
      017452  012746  000005                                                   MOV     #5,-(SP)
      017456  010600                                                           MOV     SP,R0
      017460  104415                                                           TRAP    C$PNTX
      017462  062706  000014                                                   ADD     #14,SP
  192 017466                          ENDMSG
      017466                                                         L10007:
      017466  104423                                                           TRAP    C$MSG
  193
  194
  195
  196
  197
  198 017470                          BGNMSG  ERR7
      017470                                                         ERR7::
  199 017470                          PRINTB  #FMT1,#ADDRES,MPCSR
      017470  013746  002446                                                   MOV     MPCSR,-(SP)
      017474  012746  035620                                                   MOV     #ADDRES,-(SP)
      017500  012746  011454                                                   MOV     #FMT1,-(SP)
      017504  012746  000003                                                   MOV     #3,-(SP)
      017510  010600                                                           MOV     SP,R0
      017512  104414                                                           TRAP    C$PNTB
      017514  062706  000010                                                   ADD     #10,SP
  200 017520                          PRINTB  #FMT2
      017520  012746  011464                                                   MOV     #FMT2,-(SP)
```

```
     017524  012746  000001                                   MOV     #1,-(SP)
     017530  010600                                           MOV     SP,R0
     017532  104414                                           TRAP    C$PNTB
     017534  062706  000004                                   ADD     #4,SP
 201 017540                   PRINTB  #FMT7,#DH1,REGNUM
     017540  013746  002400                                   MOV     REGNUM,-(SP)
     017544  012746  014264                                   MOV     #DH1,-(SP)
     017550  012746  011646                                   MOV     #FMT7,-(SP)
     017554  012746  000003                                   MOV     #3,-(SP)
     017560  010600                                           MOV     SP,R0
     017562  104414                                           TRAP    C$PNTB
     017564  062706  000010                                   ADD     #10,SP
 202 017570                   PRINTX  #FMT4,#DH2,#DH3
     017570  012746  014335                                   MOV     #DH3,-(SP)
     017574  012746  014306                                   MOV     #DH2,-(SP)
     017600  012746  011550                                   MOV     #FMT4,-(SP)
     017604  012746  000003                                   MOV     #3,-(SP)
     017610  010600                                           MOV     SP,R0
     017612  104415                                           TRAP    C$PNTX
     017614  062706  000010                                   ADD     #10,SP
 203 017620                   PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
     017620  013746  002310                                   MOV     LUR13,-(SP)
     017624  013746  002306                                   MOV     LUR12,-(SP)
     017630  013746  002304                                   MOV     LUR11,-(SP)
     017634  013746  002302                                   MOV     LUR10,-(SP)
     017640  012746  011563                                   MOV     #FMT5,-(SP)
     017644  012746  000005                                   MOV     #5,-(SP)
     017650  010600                                           MOV     SP,R0
     017652  104415                                           TRAP    C$PNTX
     017654  062706  000014                                   ADD     #14,SP
 204 017660                   PRINTX  #FMT9,#DH4
     017660  012746  014373                                   MOV     #DH4,-(SP)
     017664  012746  011712                                   MOV     #FMT9,-(SP)
     017670  012746  000002                                   MOV     #2,-(SP)
     017674  010600                                           MOV     SP,R0
     017676  104415                                           TRAP    C$PNTX
     017700  062706  000006                                   ADD     #6,SP
 205 017704                   PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
     017704  013746  002320                                   MOV     LUR17,-(SP)
     017710  013746  002316                                   MOV     LUR16,-(SP)
     017714  013746  002314                                   MOV     LUR15,-(SP)
     017720  013746  002312                                   MOV     LUR14,-(SP)
     017724  012746  011613                                   MOV     #FMT6,-(SP)
     017730  012746  000005                                   MOV     #5,-(SP)
     017734  010600                                           MOV     SP,R0
     017736  104415                                           TRAP    C$PNTX
     017740  062706  000014                                   ADD     #14,SP
 206 017744                   PRINTX  #FMT4,#DH7,#DH8
     017744  012746  014475                                   MOV     #DH8,-(SP)
     017750  012746  014443                                   MOV     #DH7,-(SP)
     017754  012746  011550                                   MOV     #FMT4,-(SP)
     017760  012746  000003                                   MOV     #3,-(SP)
     017764  010600                                           MOV     SP,R0
     017766  104415                                           TRAP    C$PNTX
     017770  062706  000010                                   ADD     #10,SP
 207 017774                   PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
     017774  013746  002330                                   MOV     AX1.16,-(SP)
```

```
020000  013746  002326                                          MOV     AX1.15,-(SP)
020004  013746  002324                                          MOV     AX0.16,-(SP)
020010  013746  002322                                          MOV     AX0.15,-(SP)
020014  012746  011563                                          MOV     #FMT5,-(SP)
020020  012746  000005                                          MOV     #5,-(SP)
020024  010600                                                  MOV     SP,R0
020026  104415                                                  TRAP    C$PNTX
020030  062706  000014                                          ADD     #14,SP
020034                  PRINTX  #FMT9,#DH9
020034  012746  014534                                          MOV     #DH9,-(SP)
020040  012746  011712                                          MOV     #FMT9,-(SP)
020044  012746  000002                                          MOV     #2,-(SP)
020050  010600                                                  MOV     SP,R0
020052  104415                                                  TRAP    C$PNTX
020054  062706  000006                                          ADD     #6,SP
020060                  PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
020060  013746  002340                                          MOV     AX3.16,-(SP)
020064  013746  002336                                          MOV     AX3.15,-(SP)
020070  013746  002334                                          MOV     AX2.16,-(SP)
020074  013746  002332                                          MOV     AX2.15,-(SP)
020100  012746  011613                                          MOV     #FMT6,-(SP)
020104  012746  000005                                          MOV     #5,-(SP)
020110  010600                                                  MOV     SP,R0
020112  104415                                                  TRAP    C$PNTX
020114  062706  000014                                          ADD     #14,SP
020120                  ENDMSG
020120                                                  L10010:
020120  104423                                                  TRAP    C$MSG





020122                  BGNMSG  ERR8
020122                                                  ERR8::
020122                  PRINTB  #FMT10,SUBRPC
020122  013746  002352                                          MOV     SUBRPC,-(SP)
020126  012746  011717                                          MOV     #FMT10,-(SP)
020132  012746  000002                                          MOV     #2,-(SP)
020136  010600                                                  MOV     SP,R0
020140  104414                                                  TRAP    C$PNTB
020142  062706  000006                                          ADD     #6,SP
020146                  PRINTB  #FMT1,#ADDRES,MPCSR
020146  013746  002446                                          MOV     MPCSR,-(SP)
020152  012746  035620                                          MOV     #ADDRES,-(SP)
020156  012746  011454                                          MOV     #FMT1,-(SP)
020162  012746  000003                                          MOV     #3,-(SP)
020166  010600                                                  MOV     SP,R0
020170  104414                                                  TRAP    C$PNTB
020172  062706  000010                                          ADD     #10,SP
020176                  PRINTB  #FMT2
020176  012746  011464                                          MOV     #FMT2,-(SP)
020202  012746  000001                                          MOV     #1,-(SP)
020206  010600                                                  MOV     SP,R0
020210  104414                                                  TRAP    C$PNTB
020212  062706  000004                                          ADD     #4,SP
020216                  PRINTB  #FMT7,#DH1,REGNUM
```

```
     020216  013746  002400                                      MOV     REGNUM,-(SP)
     020222  012746  014264                                      MOV     #DH1,-(SP)
     020226  012746  011646                                      MOV     #FMT7,-(SP)
     020232  012746  000003                                      MOV     #3,-(SP)
     020236  010600                                              MOV     SP,R0
     020240  104414                                              TRAP    C$PNTB
     020242  062706  000010                                      ADD     #10,SP
221  020246                      PRINTB  #FMT3,GOODAT,BADDAT
     020246  013746  002406                                      MOV     BADDAT,-(SP)
     020252  013746  002404                                      MOV     GOODAT,-(SP)
     020256  012746  011506                                      MOV     #FMT3,-(SP)
     020262  012746  000003                                      MOV     #3,-(SP)
     020266  010600                                              MOV     SP,R0
     020270  104414                                              TRAP    C$PNTB
     020272  062706  000010                                      ADD     #10,SP
222  020276                      PRINTX  #FMT4,#DH2,#DH3
     020276  012746  014335                                      MOV     #DH3,-(SP)
     020302  012746  014306                                      MOV     #DH2,-(SP)
     020306  012746  011550                                      MOV     #FMT4,-(SP)
     020312  012746  000003                                      MOV     #3,-(SP)
     020316  010600                                              MOV     SP,R0
     020320  104415                                              TRAP    C$PNTX
     020322  062706  000010                                      ADD     #10,SP
223  020326                      PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
     020326  013746  002310                                      MOV     LUR13,-(SP)
     020332  013746  002306                                      MOV     LUR12,-(SP)
     020336  013746  002304                                      MOV     LUR11,-(SP)
     020342  013746  002302                                      MOV     LUR10,-(SP)
     020346  012746  011563                                      MOV     #FMT5,-(SP)
     020352  012746  000005                                      MOV     #5,-(SP)
     020356  010600                                              MOV     SP,R0
     020360  104415                                              TRAP    C$PNTX
     020362  062706  000014                                      ADD     #14,SP
224  020366                      PRINTX  #FMT9,#DH4
     020366  012746  014373                                      MOV     #DH4,-(SP)
     020372  012746  011712                                      MOV     #FMT9,-(SP)
     020376  012746  000002                                      MOV     #2,-(SP)
     020402  010600                                              MOV     SP,R0
     020404  104415                                              TRAP    C$PNTX
     020406  062706  000006                                      ADD     #6,SP
225  020412                      PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
     020412  013746  002320                                      MOV     LUR17,-(SP)
     020416  013746  002316                                      MOV     LUR16,-(SP)
     020422  013746  002314                                      MOV     LUR15,-(SP)
     020426  013746  002312                                      MOV     LUR14,-(SP)
     020432  012746  011613                                      MOV     #FMT6,-(SP)
     020436  012746  000005                                      MOV     #5,-(SP)
     020442  010600                                              MOV     SP,R0
     020444  104415                                              TRAP    C$PNTX
     020446  062706  000014                                      ADD     #14,SP
226  020452                      PRINTX  #FMT4,#DH7,#DH8
     020452  012746  014475                                      MOV     #DH8,-(SP)
     020456  012746  014443                                      MOV     #DH7,-(SP)
     020462  012746  011550                                      MOV     #FMT4,-(SP)
     020466  012746  000003                                      MOV     #3,-(SP)
     020472  010600                                              MOV     SP,R0
     020474  104415                                              TRAP    C$PNTX
```

```
020476  062706  000010                                          ADD     #10,SP
020502                          PRINTX  #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
020502  013746  002330                                          MOV     AX1.16,-(SP)
020506  013746  002326                                          MOV     AX1.15,-(SP)
020512  013746  002324                                          MOV     AX0.16,-(SP)
020516  013746  002322                                          MOV     AX0.15,-(SP)
020522  012746  011563                                          MOV     #FMT5,-(SP)
020526  012746  000005                                          MOV     #5,-(SP)
020532  010600                                                  MOV     SP,R0
020534  104415                                                  TRAP    C$PNTX
020536  062706  000014                                          ADD     #14,SP
020542                          PRINTX  #FMT9,#DH9
020542  012746  014534                                          MOV     #DH9,-(SP)
020546  012746  011712                                          MOV     #FMT9,-(SP)
020552  012746  000002                                          MOV     #2,-(SP)
020556  010600                                                  MOV     SP,R0
020560  104415                                                  TRAP    C$PNTX
020562  062706  000006                                          ADD     #6,SP
020566                          PRINTX  #FMT6,AX2.15,AX2.16,AX3.15,AX3.16
020566  013746  002340                                          MOV     AX3.16,-(SP)
020572  013746  002336                                          MOV     AX3.15,-(SP)
020576  013746  002334                                          MOV     AX2.16,-(SP)
020602  013746  002332                                          MOV     AX2.15,-(SP)
020606  012746  011613                                          MOV     #FMT6,-(SP)
020612  012746  000005                                          MOV     #5,-(SP)
020616  010600                                                  MOV     SP,R0
020620  104415                                                  TRAP    C$PNTX
020622  062706  000014                                          ADD     #14,SP
020626                          ENDMSG
020626                                                  L10011:
020626  104423                                                  TRAP    C$MSG




020630                          BGNMSG  ERR9
020630                                                  ERR9::
020630                          PRINTB  #FMT1,#ADDRES,MPCSR
020630  013746  002446                                          MOV     MPCSR,-(SP)
020634  012746  035620                                          MOV     #ADDRES,-(SP)
020640  012746  011454                                          MOV     #FMT1,-(SP)
020644  012746  000003                                          MOV     #3,-(SP)
020650  010600                                                  MOV     SP,R0
020652  104414                                                  TRAP    C$PNTB
020654  062706  000010                                          ADD     #10,SP
020660                          PRINTB  #FMT2
020660  012746  011464                                          MOV     #FMT2,-(SP)
020664  012746  000001                                          MOV     #1,-(SP)
020670  010600                                                  MOV     SP,R0
020672  104414                                                  TRAP    C$PNTB
020674  062706  000004                                          ADD     #4,SP
020700                          PRINTB  #FMT7,#DH1,REGNUM
020700  013746  002400                                          MOV     REGNUM,-(SP)
020704  012746  014264                                          MOV     #DH1,-(SP)
020710  012746  011646                                          MOV     #FMT7,-(SP)
020714  012746  000003                                          MOV     #3,-(SP)
```

```
 020720  010600                                                          MOV     SP,R0
 020722  104414                                                          TRAP    C$PNTB
 020724  062706  000010                                                  ADD     #10,SP
 020730                          PRINTX  #FMT4,#DH2,#DH3
 020730  012746  014335                                                  MOV     #DH3,-(SP)
 020734  012746  014306                                                  MOV     #DH2,-(SP)
 020740  012746  011550                                                  MOV     #FMT4,-(SP)
 020744  012746  000003                                                  MOV     #3,-(SP)
 020750  010600                                                          MOV     SP,R0
 020752  104415                                                          TRAP    C$PNTX
 020754  062706  000010                                                  ADD     #10,SP
 020760                          PRINTX  #FMT5,LUR10,LUR11,LUR12,LUR13
 020760  013746  002310                                                  MOV     LUR13,-(SP)
 020764  013746  002306                                                  MOV     LUR12,-(SP)
 020770  013746  002304                                                  MOV     LUR11,-(SP)
 020774  013746  002302                                                  MOV     LUR10,-(SP)
 021000  012746  011563                                                  MOV     #FMT5,-(SP)
 021004  012746  000005                                                  MOV     #5,-(SP)
 021010  010600                                                          MOV     SP,R0
 021012  104415                                                          TRAP    C$PNTX
 021014  062706  000014                                                  ADD     #14,SP
 021020                          PRINTX  #FMT9,#DH4
 021020  012746  014373                                                  MOV     #DH4,-(SP)
 021024  012746  011712                                                  MOV     #FMT9,-(SP)
 021030  012746  000002                                                  MOV     #2,-(SP)
 021034  010600                                                          MOV     SP,R0
 021036  104415                                                          TRAP    C$PNTX
 021040  062706  000006                                                  ADD     #6,SP
 021044                          PRINTX  #FMT6,LUR14,LUR15,LUR16,LUR17
 021044  013746  002320                                                  MOV     LUR17,-(SP)
 021050  013746  002316                                                  MOV     LUR16,-(SP)
 021054  013746  002314                                                  MOV     LUR15,-(SP)
 021060  013746  002312                                                  MOV     LUR14,-(SP)
 021064  012746  011613                                                  MOV     #FMT6,-(SP)
 021070  012746  000005                                                  MOV     #5,-(SP)
 021074  010600                                                          MOV     SP,R0
 021076  104415                                                          TRAP    C$PNTX
 021100  062706  000014                                                  ADD     #14,SP
 021104                          ENDMSG
 021104                                                         L10012:
 021104  104423                                                          TRAP    C$MSG
```

```
                                        .SBTTL  REPORT CODING SECTION


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

 021106                                       BGNRPT
 021106                                                                    L$RPT::

 021106                                       ENDRPT
 021106                                                                    L10013:
 021106  104425                                                                    TRAP    C$RPT
```

```
                                .SBTTL  LOAD DEVICE PROTECTION TABLE

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

021110                          BGNPROT
021110                                                                  L$PROT::
021110  177777                  .WORD   -1      ;DON'T CHK CSR ADRS
021112  177777                  .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
021114  177777                  .WORD   -1      ;DON'T CHK DRIVE NO.
021116                          ENDPROT
```

```
                                                .SBTTL  INITIALIZE SECTION

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

021116                                                  BGNINIT
021116                                                                                          L$INIT::

021116  010637  002346                                  MOV     SP,PSTACK       ;SAVE BASE-LEVEL STACK POINTER
021122  005037  002352                                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
021126  005037  002426                                  CLR     DISILO          ;CLEAR CURRENT STATE OF DISSI
021132  005037  002430                                  CLR     CHPTYP          ;CLEAR USYRT CHIP TYPE INDICATOR
021136  005037  002420                                  CLR     ERROR1          ;CLEAR ERROR FLAG
021142  005037  002432                                  CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
021146  005737  002412                                  TST     FRSTIM          ;SEE IF FIRST TIME THROUGH AFTER LOAD
021152  001007                                          BNE     6$              ;BR IF NOT
021154  013737  000004  002414                          MOV     @#4,SAVE4       ;SAVE ERROR TRAP VECTOR
021162  013737  000006  002416                          MOV     @#6,SAVE6
021170  000406                                          BR      9$
021172  013737  002414  000004          6$:             MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
021200  013737  002416  000006                          MOV     SAVE6,@#6
021206  012737  000001  002412          9$:             MOV     #1,FRSTIM       ;MARK FLAG FOR NEXT TIME THROUGH
                                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
021214                                                  READEF  #EF.START
021214  012700  000040                                                                          MOV     #EF.START,R0
021220  104447                                                                                  TRAP    C$REFG
021222                                                  BCOMPLETE       STARST
021222  103415                                                                                  BCS     STARST
                                                ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
021224                                                  READEF  #EF.RESTART
021224  012700  000037                                                                          MOV     #EF.RESTART,R0
021230  104447                                                                                  TRAP    C$REFG
021232                                                  BCOMPLETE       STARST
021232  103411                                                                                  BCS     STARST
                                                ;SEE IF THIS IS A NEW PASS, BR IF YES
021234                                                  READEF  #EF.NEW
021234  012700  000035                                                                          MOV     #EF.NEW,R0
021240  104447                                                                                  TRAP    C$REFG
021242                                                  BCOMPLETE       NEWST
021242  103411                                                                                  BCS     NEWST
                                                ;SEE IF PROGRAM WAS JUST CONTINUED
021244                                                  READEF  #EF.CONTINUE
021244  012700  000036                                                                          MOV     #EF.CONTINUE,R0
021250  104447                                                                                  TRAP    C$REFG
021252                                                  BCOMPLETE       ENDIT
021252  103504                                                                                  BCS     ENDIT
021254  000414                                          BR      GETPRM
021256                                          STARST:
021256  005037  002444                                  CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
                                                ;CLEAR DEVICE MAP
021262  005037  002434                                  CLR     DEVMAP
021266                                          NEWST:
021266  012737  177777  002344                          MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
021274  005237  002444                                  INC     STARES          ;INCR NO. OF PASSES SINCE STA OR RES
021300  012737  000001  002436                          MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
```

```
                                        ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
                                        ;    CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
021306                                  GETPRM:
021306  005237  002344                          INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
021312  023737  002344  002012                  CMP     LOGDEV,L$UNIT   ;SEE IF MAXIMUM UNIT NO. EXCEEDED
021320  002362                                  BGE     NEWST           ;BR IF YES
021322                                          GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
021322  013700  002344                                                                          MOV     LOGDEV,R0
021326  104442                                                                                  TRAP    C$GPHRD
021330  010001                                                                                  MOV     R0,R1
021332                                          BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
021332  103403                                                                                  BCS     10$
021334  006337  002436                          ASL     DEVPTR          ;SHIFT DEVICE POINTER
021340  000762                                  BR      GETPRM          ;SKIP THIS DEVICE
021342  053737  002436  002434  10$:            BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE
021350  006337  002436                          ASL     DEVPTR          ;SHIFT BIT POINTER
021354  062701  000002                          ADD     #2,R1           ;INCREMENT R1 PAST MICROPROCESSOR TYPE
021360  011137  002446                          MOV     (R1),MPCSR      ;STORE POINTER TO MICROPROCESSOR CSR'S
021364  011137  002450                          MOV     (R1),BSEL1
021370  005237  002450                          INC     BSEL1           ;GET POINTER TO BSEL1 (MAINTENANCE REGISTER)
021374  013737  002450  002452                  MOV     BSEL1,BSEL2
021402  005237  002452                          INC     BSEL2           ;GET POINTER TO BSEL2
021406  011137  002454                          MOV     (R1),SEL4
021412  062737  000004  002454                  ADD     #4,SEL4         ;GET POINTER TO SEL4
021420  012137  002456                          MOV     (R1)+,SEL6
021424  062737  000006  002456                  ADD     #6,SEL6         ;STORE POINTER TO SEL6
021432  011137  002460                          MOV     (R1),MPIVEC     ;GET MICROPROCESSOR INPUT INTRPT VECTOR
021436  012137  002462                          MOV     (R1)+,MPOVEC
021442  062737  000004  002462                  ADD     #4,MPOVEC       ;GET MICROPROCESSOR OUTPUT INTRPT VECTOR
021450  012137  002464                          MOV     (R1)+,MPRIOR    ;GET MICROPROCESSOR DEVICE PRIORITY
021454  062701  000014                          ADD     #14,R1          ;POINT R1 TO RUN SWITCH INDICATOR
021460  011137  002476                          MOV     (R1),RUNINH     ;GET STATE OF MICROPROCESSOR RUN SWITCH
021464                                  ENDIT:
021464                                          ENDINIT
021464                                                                                  L10015:
021464  104411                                                                                  TRAP    C$INIT
```

```
                                        .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.
                                        ;////////////////////////////////////////////////////////////////////////////////
 021466                                         BGNAUTO
 021466                                                                                          L$AUTO::
                                        ;ESTABLISH CPU PRIORITY = 7
 021466                                         SETPRI  #PRI07
 021466  012700  000340                                                                                  MOV     #PRI07,R0
 021472  104441                                                                                          TRAP    C$SPRI
 021474  012737  021516  000004                 MOV     #6$,@#4         ;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR
 021502  012737  000340  000006                 MOV     #PRI07,@#6
 021510  005777  160732                         TST     @MPCSR          ;ADDRESS SEL0
 021514  000405                                 BR      9$              ;TAKE THIS BR IF DEVICE RESPONDS
                                        ;COME HERE IF DEVICE CSR IS NON-EXISTENT
 021516  062706  000004                 6$:     ADD     #4,SP           ;CLEAN UP THE STACK POINTER
 021522                                         DODU    LOGDEV          ;DROP THIS UNIT FROM TESTING
 021522  013700  002344                                                                                  MOV     LOGDEV,R0
 021526  104451                                                                                          TRAP    C$DODU
 021530  013737  002414  000004         9$:     MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
 021536  013737  002416  000006                 MOV     SAVE6,@#6
 021544                                         ENDAUTO
 021544                                                                                          L10016:
 021544  104461                                                                                          TRAP    C$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.
                                        ;//////////////////////////////////////////////////////////////////////////////

 021546                                         BGNCLN
 021546                                                                                L$CLEAN::


 021546                                         ENDCLN
 021546                                                                                L10017:
 021546  104412                                                                                 TRAP    C$CLEAN
```

```
                                        .SBTTL  DROP UNIT SECTION

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

021550                                          BGNDU
021550                                                                                  L$DU::
                                        ;ISSUE UNIBUS RESET TO CLEAN UP
021550                                          BRESET
021550  104433                                                                                  TRAP    C$RESET
                                        ;PRINT "UNIT XX DROPPED"
021552                                          PRINTF  #FMT27,LOGDEV
021552  013746  002344                                                                          MOV     LOGDEV,-(SP)
021556  012746  021600                                                                          MOV     #FMT27,-(SP)
021562  012746  000002                                                                          MOV     #2,-(SP)
021566  010600                                                                                  MOV     SP,R0
021570  104417                                                                                  TRAP    C$PNTF
021572  062706  000006                                                                          ADD     #6,SP
021576                                          ENDDU
021576                                                                                  L10020:
021576  104453                                                                                  TRAP    C$DU

021600     045     116     045          FMT27:  .ASCIZ  /%N%AUNIT %D2%A DROPPED%N/
021603     101     125     116
021606     111     124     040
021611     045     104     062
021614     045     101     040
021617     104     122     117
021622     120     120     105
021625     104     045     116
021630     000
                                        .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.
                               ;//////////////////////////////////////////////////////////////////////

021632                                  BGNAU
021632                                                                  L$AU::
021632                                  ENDAU
021632                                                                  L10021:
021632  104452                                                                   TRAP    C$AU
```

```
                                        .SBTTL  HARDWARE TESTS






                                        ;;*********************************************************************
                                        .SBTTL       TEST 1 - MICROPROCESSOR CSR ADDRESSING TEST (SELO)
                                        ;*
                                        ;* THIS TEST ADDRESSES THE FIRST MICROPROCESSOR CSR (SELO), TO MAKE SURE
                                        ;* THAT A NON-EXISTENT MEMORY TIME-OUT TRAP DOES NOT OCCUR WHILE
                                        ;* ATTEMPTING TO ADDRESS THE MICROPROCESSOR.
                                        ;;*********************************************************************
021634                                  BGNTST
021634                                                                                  T1::
                                        ;ESTABLISH CPU PRIORITY = 7
021634                                          SETPRI  #PRI07
021634  012700  000340                                                                          MOV     #PRI07,RO
021640  104441                                                                                  TRAP    C$SPRI
021642  012737  021664  000004                  MOV     #6$,@#4         ;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR
021650  012737  000340  000006                  MOV     #PRI07,@#6
021656  005777  160564                          TST     @MPCSR          ;ADDRESS SELO
021662  000406                                  BR      9$              ;TAKE THIS BR IF DEVICE RESPONDS
                                        ;COME HERE IF DEVICE CSR IS NON-EXISTENT
021664  062706  000004                  6$:     ADD     #4,SP           ;CLEAN UP THE STACK POINTER
                                        ;REPORT CSR ADDRESS TIME-OUT
021670                                          ERRDF   1,EM1,ERR1
021670  104455                                                                                  TRAP    C$ERDF
021672  000001                                                                                  .WORD   1
021674  012125                                                                                  .WORD   EM1
021676  014600                                                                                  .WORD   ERR1
021700  013737  002414  000004  9$:     MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
021706  013737  002416  000006          MOV     SAVE6,@#6
021714                                  ENDTST
021714                                                                                  L10022:
021714  104401                                                                                  TRAP    C$ETST





                                        ;;*********************************************************************
                                        .SBTTL       TEST 2 - INBUS/OUTBUS REG 14 INITIALIZATION TEST
                                        ;*
                                        ;* MASTER CLEAR (MCLR) IS SET IN THE MICROPROCESSOR, IBUS REG 14 IS READ
                                        ;* AND COMPARED TO 200.
                                        ;;*********************************************************************
021716                                  BGNTST
021716                                                                                  T2::
021716  012737  000014  002400                  MOV     #14,REGNUM      ;SET LU REG NO. = 14
021724  004737  003576                          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
021730  004737  003672                          JSR     PC,READLU       ;READ REG 14
021734  123737  002364  003026                  CMPB    REDBYT,PATM+4   ;CHK FOR INITIALIZED STATE
021742  001416                                  BEQ     6$              ;BR IF YES
021744  005037  002404                          CLR     GOODAT          ;SET EXPECTED REG CONTENTS = 000
021750  113737  003026  002404                  MOVB    PATM+4,GOODAT   ;SET EXPECTED DATA
021756  013737  002364  002406                  MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
```

```
021764  004737  004016                 JSR     PC,GETREG        ;GET REGS FOR PRINTOUT
                               ;REPORT REG NOT CLEARED BY MASTER CLEAR
021770                                 ERRDF   2,EM2,ERR2
021770  104455                                                                   TRAP    C$ERDF
021772  000002                                                                   .WORD   2
021774  012161                                                                   .WORD   EM2
021776  014632                                                                   .WORD   ERR2
022000                         6$:
022000                         ENDTST
022000                                                                   L10023:
022000  104401                                                                   TRAP    C$ETST





                               ;;*********************************************************************
                               .SBTTL       TEST 3 - INBUS/OUTBUS REG 14 READ/WRITE BIT TEST
                               ;*
                               ;* WRITE, READ, AND COMPARE ALL WORDS OF DATA PATTERN A INTO REG 14,
                               ;* A BYTE AT A TIME. NON-R/W BITS ARE MASKED OFF TO 0 BEFORE WRITING AND
                               ;* READING.
                               ;*     DATA PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
                               ;*                        375,373,367,357,337,277,177.
                               ;;*********************************************************************
022002                         BGNTST
022002                                                                   T3::
022002  004737  003576                 JSR     PC,MSTCLR        ;ISSUE MASTER CLEAR
022006  012737  000014  002400         MOV     #14,REGNUM       ;SET LU REG NO. = 14
022014  012701  002571                 MOV     #PATA,R1         ;GET POINTER TO DATA PAT IN R1
022020                         3$:
022020                                 BGNSEG
022020  104404                                                                   TRAP    C$BSEG
022022  111137  002366                 MOVB    (R1),WRIBYT      ;GET A BYTE OF PAT A
022026  143737  002560  002366         BICB    R14NRW,WRIBYT    ;MASK OFF NON-READ/WRITE BITS
022034  004737  003750                 JSR     PC,WRITLU        ;WRITE DATA BYTE INTO REG 14
022040  004737  003672                 JSR     PC,READLU        ;READ DATA BYTE FROM REG 14
022044  143737  002560  002364         BICB    R14NRW,REDBYT    ;MASK OFF NON-READ/WRITE BITS
022052  123737  002364  002366         CMPB    REDBYT,WRIBYT    ;COMPARE BYTE READ TO BYTE WRITTEN
022060  001414                         BEQ     6$               ;BR IF BYTES MATCH
022062  013737  002366  002404         MOV     WRIBYT,GOODAT    ;SET EXPECTED REG CONTENTS
022070  013737  002364  002406         MOV     REDBYT,BADDAT    ;SET ACTUAL REG CONTENTS
022076  004737  004016                 JSR     PC,GETREG        ;GET REGS FOR PRINTOUT
                               ;REPORT LINE UNIT REG MISCOMPARE
022102                                 ERRDF   3,EM3,ERR2
022102  104455                                                                   TRAP    C$ERDF
022104  000003                                                                   .WORD   3
022106  012220                                                                   .WORD   EM3
022110  014632                                                                   .WORD   ERR2
022112                         6$:
022112                                 ENDSEG
022112                                                                   10000$:
022112  104405                                                                   TRAP    C$ESEG
022114  005201                         INC     R1               ;INCREMENT DATA PATTERN POINTER
022116  020127  002615                 CMP     R1,#PATB         ;SEE IF ALL WORDS OF PATTERN A USED YET
022122  103736                         BLO     3$               ;BR IF NOT DONE YET
022124                         ENDTST
```

```
 022124                                                              L10024:
 022124  104401                                                               TRAP    C$ETST
                              ;;************************************************************************
                              .SBTTL      TEST 4 - REG 14 MASTER CLEAR TEST
                              ;* WRITE 377 INTO REG 14, ISSUE MASTER CLEAR, READ REG 14 AND COMPARE
                              ;* TO 200.
                              ;;************************************************************************
 022126                       BGNTST
 022126                                                              T4::
 022126  004737  003576                 JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR
 022132  012737  000014  002400         MOV     #14,REGNUM      ;SET LU REG NO. = 14
 022140  112737  000377  002366         MOVB    #377,WRIBYT     ;SET DATA BYTE = 377
 022146  004737  003750                 JSR     PC,WRITLU       ;WRITE 377 INTO REG 14
 022152  004737  003576                 JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
 022156  004737  003672                 JSR     PC,READLU       ;READ REG 14
 022162  123737  002364  003026         CMPB    REDBYT,PATM+4   ;CHK FOR INIT'D STATE
 022170  001416                         BEQ     6$              ;BR IF REG GOT CLEARED
 022172  005037  002404                 CLR     GOODAT
 022176  113737  003026  002404         MOVB    PATM+4,GOODAT   ;SET EXPECTED DATA
 022204  013737  002364  002406         MOV     REDBYT,BADDAT   ;SET ACTUAL DATA
 022212  004737  004016                 JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                              ;REPORT REG NOT CLEARED BY MASTER CLEAR
 022216                                 ERRDF   2,EM2,ERR2
 022216  104455                                                               TRAP    C$ERDF
 022220  000002                                                               .WORD   2
 022222  012161                                                               .WORD   EM2
 022224  014632                                                               .WORD   ERR2
 022226                       6$:
 022226                       ENDTST
 022226                                                              L10025:
 022226  104401                                                               TRAP    C$ETST
                              ;;************************************************************************
                              .SBTTL      TEST 5 - REG 14 UNIBUS RESET (INIT) TEST
                              ;* WRITE 377 INTO REG 14, ISSUE UNIBUS RESET (INIT), READ REG 14 AND COMPARE
                              ;* TO 200.
                              ;;************************************************************************
 022230                       BGNTST
 022230                                                              T5::
 022230  004737  003576                 JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR
 022234  012737  000014  002400         MOV     #14,REGNUM      ;SET LU REG NO. = 14
 022242  112737  000377  002366         MOVB    #377,WRIBYT     ;SET DATA BYTE = 377
 022250  004737  003750                 JSR     PC,WRITLU       ;WRITE 377 INTO REG 14
 022254                                 BRESET                  ;ISSUE UNIBUS RESET (INIT)
 022254  104433                                                               TRAP    C$RESET
 022256  142777  000200  160164         BICB    #RUN,@BSEL1     ;CLEAR RUN BIT
 022264  012701  000024                 MOV     #20.,R1         ;INIT LOOP COUNTER
 022270  000240               2$:       NOP
```

```
 022272  005301                          DEC     R1                 ;DECR COUNTER
 022274  001375                          BNE     2$                 ;BR TO STALL
 022276  004737  003672                  JSR     PC,READLU          ;READ REG 14
 022302  142737  000100  002364          BICB    #TXEN,REDBYT       ;CLEAR UNPREDICTABLE BIT
 022310  123737  002364  003026          CMPB    REDBYT,PATM+4      ;CHK FOR INIT'D STATE
 022316  001416                          BEQ     6$                 ;BR IF REG GOT CLEARED
 022320  005037  002404                  CLR     GOODAT
 022324  113737  003026  002404          MOVB    PATM+4,GOODAT      ;SET EXPECTED DATA
 022332  013737  002364  002406          MOV     REDBYT,BADDAT      ;SET ACTUAL DATA
 022340  004737  004016                  JSR     PC,GETREG          ;GET REGS FOR PRINTOUT
                                 ;REPORT REG NOT CLEARED BY UNIBUS RESET (INIT)
 022344                                  ERRDF   4,EM4,ERR2
 022344  104455                                                                          TRAP    C$ERDF
 022346  000004                                                                          .WORD   4
 022350  012237                                                                          .WORD   EM4
 022352  014632                                                                          .WORD   ERR2
 022354                          6$:
 022354                          ENDTST
 022354                                                                          L10026:
 022354  104401                                                                          TRAP    C$ETST





                                 ;;*************************************************************************
                                 .SBTTL       TEST 6 - LINE UNIT FALSE SELECTION TEST
                                 ;*
                                 ;* FIRST, A MASTER CLEAR IS PERFORMED. THEN, THE PROGRAM SINGLE-STEPS THE
                                 ;* MICROPROCESSOR THROUGH AN INSTRUCTION WHICH LOADS 041 (OCT) INTO THE MAR
                                 ;* REGISTER (OBUS* ADRS 14). THEN, THE LINE UNIT REGISTER 14 IS READ AND CHECKED
                                 ;* TO BE UNAFFECTED (STILL = 0). THIS TEST IS INTENDED TO DETECT A FALSE
                                 ;* SELECTION OF THE LINE UNIT REGISTERS, WHEN THE LINE UNIT IS NOT BEING
                                 ;* ACCESSED.
                                 ;;*************************************************************************
 022356                          BGNTST
 022356                                                                  T6::
 022356  004737  003576                  JSR     PC,MSTCLR          ;ISSUE A MASTER CLEAR
 022362  012737  000014  002400          MOV     #14,REGNUM         ;SET LU REG NO. = 14
 022370  013701  002400                  MOV     REGNUM,R1          ;SET DESTINATION = OBUS* REG 14
 022374  052701  000100                  BIS     #100,R1            ;SET SOURCE = BSEL4
 022400  052701  121000                  BIS     #MVIXOX,R1         ;SET REST OF MOVE INSTRUCTION
 022404  010137  022422                  MOV     R1,2$              ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
 022410  112777  000041  160036          MOVB    #041,@BSEL4        ;SET DATA BYTE = 041
 022416  004737  003540                  JSR     PC,STPCLK          ;EXECUTE MOVE INSTRUCTION
 022422  000000                  2$:     .WORD   0                  ;INSTRUCTION GOES HERE
 022424  004737  003672                  JSR     PC,READLU          ;READ LU REG 14
 022430  123737  002364  003026          CMPB    REDBYT,PATM+4      ;CHECK FOR LU REG 14 UNCHANGED
 022436  001416                          BEQ     4$                 ;BR IF LU REG 14 UNCHANGED
 022440  005037  002404                  CLR     GOODAT             ;SET EXPECTED DATA
 022444  113737  003026  002404          MOVB    PATM+4,GOODAT
 022452  013737  002364  002406          MOV     REDBYT,BADDAT      ;SET ACTUAL DATA
 022460  004737  004016                  JSR     PC,GETREG          ;GET REGS FOR PRINTOUT
                                 ;REPORT REGISTER MISCOMPARE
 022464                                  ERRDF   3,EM3,ERR2
 022464  104455                                                                          TRAP    C$ERDF
 022466  000003                                                                          .WORD   3
```

```
022470  012220                                                                          .WORD   EM3
022472  014632                                                                          .WORD   ERR2
022474                                  4$:
022474                                  ENDTST
022474                                                                      L10027:
022474  104401                                                                          TRAP    C$ETST
                                ;;*****************************************************************
                                .SBTTL      TEST 7 - INBUS REG MASTER CLEAR TEST
                                ;*
                                ;* FIRST, ALL READ/WRITE BITS OF REGS 10-17 ARE SET BY LOADING A
                                ;* DIFFERENT WORD OF PATTERN G INTO EACH REG.  THEN,
                                ;* A MASTER CLEAR IS ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF
                                ;* PATTERN M, WHICH CONTAINS THE INITIALIZED STATES OF THE REGS. (UNPREDICTABLE
                                ;* BITS ARE MASKED OFF TO 0 BEFORE COMPARISON).
                                ;*   PATTERN G = 000,000,240,120,177,000,000,001
                                ;*   PATTERN M = 000,020,000,000,200,000,000,051
                                ;;*****************************************************************
022476                          BGNTST
022476                                                                          T7::
022476  004737  003576                  JSR     PC,MSTCLR           ;ISSUE MASTER CLEAR
022502  012737  000010  002400          MOV     #10,REGNUM          ;INIT REG NO. TO 10
022510  012701  002644                  MOV     #PATG,R1            ;INIT DATA PATTERN POINTER
022514  112137  002366          2$:     MOVB    (R1)+,WRIBYT        ;SET DATA PATTERN BYTE TO BE WRITTEN
022520  004737  003750                  JSR     PC,WRITLU           ;WRITE BYTE INTO REG
022524  005237  002400                  INC     REGNUM              ;INCREMENT REG NO. FOR WRITING
022530  020127  002654                  CMP     R1,#PATH            ;SEE IF ALL BYTES WRITTEN YET
022534  103767                          BLO     2$                  ;BR IF NOT DONE WRITING YET
022536  004737  003576                  JSR     PC,MSTCLR           ;ISSUE MASTER CLEAR
022542  012737  000010  002400          MOV     #10,REGNUM          ;INIT LU REG NO. TO 10
022550  012702  003022                  MOV     #PATM,R2            ;INIT DATA PATTERN POINTER
022554  012701  002550                  MOV     #UPBITS,R1          ;INIT POINTER TO UNPREDICTABLE BITS
022560                          3$:
022560  004737  003672                  JSR     PC,READLU           ;READ A LINE UNIT REG
022564  142137  002364                  BICB    (R1)+,REDBYT        ;MASK OUT UNPREDICTABLE BITS FOR THIS REG
022570  123712  002364                  CMPB    REDBYT,(R2)         ;COMPARE MASKED DATA TO EXPECTED
022574  001417                          BEQ     6$                  ;BR IF DATA READ IS OK
022576  005037  002404                  CLR     GOODAT
022602  111237  002404                  MOVB    (R2),GOODAT         ;SET EXPECTED DATA
022606  013737  002364  002406          MOV     REDBYT,BADDAT       ;SET ACTUAL DATA
022614  004737  004016                  JSR     PC,GETREG           ;GET REGS FOR PRINTOUT
                                ;REPORT REG NOT CLEARED BY MASTER CLEAR
022620                                  ERRDF   2,EM2,ERR2
022620  104455                                                                          TRAP    C$ERDF
022622  000002                                                                          .WORD   2
022624  012161                                                                          .WORD   EM2
022626  014632                                                                          .WORD   ERR2
022630                                  ESCAPE  TST
022630  104410                                                                          TRAP    C$ESCAPE
022632  000016                                                                          .WORD   L10030-.
022634  005237  002400          6$:     INC     REGNUM              ;INCREMENT REG NO.
022640  005202                          INC     R2                  ;INCR DATA PATTERN POINTER
022642  020227  003032                  CMP     R2,#PATN            ;SEE IF ALL DONE YET
```

```
022646  103744                          BLO     3$                 ;BR IF NOT DONE YET
022650                          ENDTST
022650                                                                       L10030:
022650  104401                                                                          TRAP    C$ETST





                                ;;*************************************************************************
                                .SBTTL       TEST 8 - REGISTER 10-17 ADDRESSING TEST
                                ;*
                                ;* FIRST, A MASTER CLEAR IS ISSUED. THEN,
                                ;* WRITE A DIFFERENT WORD OF DATA PATTERN B INTO EACH OF REGS 10-17,
                                ;* AND AFTER EACH WRITE, READ AND COMPARE ALL REGS TO EXPECTED VALUES.
                                ;* UNPREDICTABLE BITS ARE MASKED OFF TO 0 WHEN READ FOR COMPARISON.
                                ;*    PATTERN B = 000,000,040,100,220,000,000,051
                                ;;*************************************************************************
022652                          BGNTST
022652                                                                       T8::
022652  004737  003576                  JSR     PC,MSTCLR          ;ISSUE MASTER CLEAR
022656  012701  002500                  MOV     #REDDAT,R1         ;INIT POINTER TO EXPECTED DATA AREA
022662  012702  003022                  MOV     #PATM,R2           ;GET POINTER TO PATTERN M
022666  012703  000010                  MOV     #8.,R3             ;SET COUNTER
022672  112221                  3$:     MOVB    (R2)+,(R1)+        ;LOAD BYTE OF PATRN INTO EXPECTED DATA AREA
022674  005303                          DEC     R3                 ;DECR COUNTER
022676  001375                          BNE     3$                 ;BR IF NOT DONE LOADING YET
022700  005001                          CLR     R1                 ;INIT DATA PATTERN INDEX FOR WRITING
022702  010137  002400          6$:     MOV     R1,REGNUM
022706  062737  000010  002400          ADD     #10,REGNUM         ;GET REG NO. FOR WRITING
022714  116137  002615  002366          MOVB    PATB(R1),WRIBYT ;SET DATA BYTE TO BE WRITTEN
022722  113761  002366  002500          MOVB    WRIBYT,REDDAT(R1)  ;SET EXPECTED DATA FOR READ
022730  146161  002550  002500          BICB    UPBITS(R1),REDDAT(R1)  ;MASK OUT UNPREDICTABLE BITS
022736  004737  003750                  JSR     PC,WRITLU          ;WRITE DATA BYTE INTO REG
022742  005003                          CLR     R3                 ;INIT DATA PAT INDEX FOR READS
022744  010337  002400          9$:     MOV     R3,REGNUM
022750  062737  000010  002400          ADD     #10,REGNUM         ;GET REG NO. FOR READING
022756  004737  003672                  JSR     PC,READLU          ;READ A LINE UNIT REG
022762  146337  002550  002364          BICB    UPBITS(R3),REDBYT  ;MASK OUT UNPREDICTABLE BITS
022770  023727  002400  000011          CMP     REGNUM,#11         ;SEE IF READING REG 11
022776  001006                          BNE     10$                ;BR IF NOT
023000  142737  000020  002364          BICB    #ORDY,REDBYT       ;MASK ORDY BIT IN ACTUAL BYTE
023006  142763  000020  002500          BICB    #ORDY,REDDAT(R3)   ;MASK ORDY BIT IN EXPECTED BYTE
023014  123763  002364  002500  10$:    CMPB    REDBYT,REDDAT(R3)  ;COMPARE BYTE READ TO EXPECTED
023022  001420                          BEQ     12$                ;BR IF DATA MATCHES
023024  005037  002404                  CLR     GOODAT
023030  116337  002500  002404          MOVB    REDDAT(R3),GOODAT  ;SET EXPECTED DATA
023036  013737  002364  002406          MOV     REDBYT,BADDAT      ;SET ACTUAL DATA
023044  004737  004016                  JSR     PC,GETREG          ;READ AND STORE REGS 10-17 FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
023050                                  ERRDF   3,EM3,ERR2
023050  104455                                                                          TRAP    C$ERDF
023052  000003                                                                          .WORD   3
023054  012220                                                                          .WORD   EM3
023056  014632                                                                          .WORD   ERR2
023060                                  ESCAPE  TST
023060  104410                                                                          TRAP    C$ESCAPE
```

```
023062  000022                                                                      .WORD   L10031-.
023064  005203                          12$:    INC     R3              ;INCREMENT DATA PATTERN INDEX FOR READING
023066  020327  000010                          CMP     R3,#10          ;SEE IF ALL REGS READ YET
023072  002724                                  BLT     9$              ;BR IF NOT
023074  005201                                  INC     R1              ;INCREMENT DATA PAT INDEX FOR WRITING
023076  020127  000010                          CMP     R1,#10          ;SEE IF ALL REGS WRITTEN YET
023102  002677                                  BLT     6$              ;BR IF NOT
023104                                  ENDTST
023104                                                                          L10031:
023104  104401                                                                      TRAP    C$ETST

                                        ;;*******************************************************************
                                        .SBTTL          TEST 9 - REG 11 READ/WRITE BIT TEST
                                        ;*
                                        ;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN C INTO REG 11 :
                                        ;*      DATA PATTERN C = 020,020,020.
                                        ;;*******************************************************************
023106                                  BGNTST
023106                                                                          T9::
023106  004737  003576                          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
023112  012737  000011  002400                  MOV     #11,REGNUM      ;SET LU REG NO. = 11
023120  012701  002625                          MOV     #PATC,R1        ;GET POINTER TO DATA PAT IN R1
023124                                  3$:
023124                                          BGNSEG
023124  104404                                                                      TRAP    C$BSEG
023126  111137  002366                          MOVB    (R1),WRIBYT     ;GET A BYTE OF PAT C
023132  004737  003750                          JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG 11
023136  004737  003672                          JSR     PC,READLU       ;READ DATA BYTE FROM REG 11
023142  143737  002551  002364                  BICB    UPBITS+1,REDBYT ;MASK OUT UNPREDICTABLE BITS
023150  123737  002364  002366                  CMPB    REDBYT,WRIBYT   ;COMPARE BYTE READ TO BYTE WRITTEN
023156  001414                                  BEQ     6$              ;BR IF BYTES MATCH
023160  013737  002366  002404                  MOV     WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
023166  013737  002364  002406                  MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
023174  004737  004016                          JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                                        ;REPORT LINE UNIT REG MISCOMPARE
023200                                          ERRDF   3,EM3,ERR2
023200  104455                                                                      TRAP    C$ERDF
023202  000003                                                                      .WORD   3
023204  012220                                                                      .WORD   EM3
023206  014632                                                                      .WORD   ERR2
023210                                  6$:
023210                                          ENDSEG
023210                                                                          10000$:
023210  104405                                                                      TRAP    C$ESEG
023212  005201                                  INC     R1              ;INCREMENT DATA PATTERN POINTER
023214  020127  002630                          CMP     R1,#PATD        ;SEE IF ALL WORDS OF PATTERN C USED YET
023220  103741                                  BLO     3$              ;BR IF NOT DONE YET
023222                                  ENDTST
023222                                                                          L10032:
023222  104401                                                                      TRAP    C$ETST
```

```
                                    ;;*******************************************************************
                                    .SBTTL      TEST 10 - REG 12 READ/WRITE BIT TEST
                                    ;*
                                    ;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN D INTO REG 12 :
                                    ;*      DATA PATTERN D = 000,040,000.
                                    ;;*******************************************************************
023224                              BGNTST
023224                                                                              T10::
023224  004737  003576                      JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
023230  012737  000012  002400              MOV     #12,REGNUM      ;SET LU REG NO. = 12
023236  012701  002630                      MOV     #PATD,R1        ;GET POINTER TO DATA PAT IN R1
023242                              3$:
023242                                      BGNSEG
023242  104404                                                                          TRAP    C$BSEG
023244  111137  002366                      MOVB    (R1),WRIBYT     ;GET A BYTE OF PAT D
023250  004737  003750                      JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG 12
023254  004737  003672                      JSR     PC,READLU       ;READ DATA BYTE FROM REG 12
023260  143737  002552  002364              BICB    UPBITS+2,REDBYT ;MASK OUT UNPREDICTABLE BITS
023266  123737  002364  002366              CMPB    REDBYT,WRIBYT   ;COMPARE BYTE READ TO BYTE WRITTEN
023274  001414                              BEQ     6$              ;BR IF BYTES MATCH
023276  013737  002366  002404              MOV     WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
023304  013737  002364  002406              MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
023312  004737  004016                      JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                                    ;REPORT LINE UNIT REG MISCOMPARE
023316                                      ERRDF   3,EM3,ERR2
023316  104455                                                                          TRAP    C$ERDF
023320  000003                                                                          .WORD   3
023322  012220                                                                          .WORD   EM3
023324  014632                                                                          .WORD   ERR2
023326                              6$:
023326                                      ENDSEG
023326                                                                              10000$:
023326  104405                                                                          TRAP    C$ESEG
023330  005201                              INC     R1              ;INCREMENT DATA PATTERN POINTER
023332  020127  002633                      CMP     R1,#PATE        ;SEE IF ALL WORDS OF PATTERN D USED YET
023336  103741                              BLO     3$              ;BR IF NOT DONE YET
023340                              ENDTST
023340                                                                              L10033:
023340  104401                                                                          TRAP    C$ETST





                                    ;;*******************************************************************
                                    .SBTTL      TEST 11 - REG 13 READ/WRITE BIT TEST
                                    ;*
                                    ;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN E INTO REG 13 :
                                    ;*      DATA PATTERN E = 000,120,020,100,120,000.
                                    ;;*******************************************************************
023342                              BGNTST
023342                                                                              T11::
023342  004737  003576                      JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
023346  012737  000013  002400              MOV     #13,REGNUM      ;SET LU REG NO. = 13
023354  012701  002633                      MOV     #PATE,R1        ;GET POINTER TO DATA PAT IN R1
```

```
023360                                  3$:
023360                                          BGNSEG
023360  104404                                                                                  TRAP    C$BSEG
023362  111137  002366                          MOVB    (R1),WRIBYT     ;GET A BYTE OF PAT E
023366  004737  003750                          JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG 13
023372  004737  003672                          JSR     PC,READLU       ;READ DATA BYTE FROM REG 13
023376  143737  002553  002364                  BICB    UPBITS+3,REDBYT ;MASK OUT UNPREDICTABLE BITS
023404  123737  002364  002366                  CMPB    REDBYT,WRIBYT   ;COMPARE BYTE READ TO BYTE WRITTEN
023412  001414                                  BEQ     6$              ;BR IF BYTES MATCH
023414  013737  002366  002404                  MOV     WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
023422  013737  002364  002406                  MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
023430  004737  004016                          JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                                        ;REPORT LINE UNIT REG MISCOMPARE
023434                                          ERRDF   3,EM3,ERR2
023434  104455                                                                                  TRAP    C$ERDF
023436  000003                                                                                  .WORD   3
023440  012220                                                                                  .WORD   EM3
023442  014632                                                                                  .WORD   ERR2
023444                                  6$:
023444                                          ENDSEG
023444                                                                                  10000$:
023444  104405                                                                                  TRAP    C$ESEG
023446  005201                                  INC     R1              ;INCREMENT DATA PATTERN POINTER
023450  020127  002641                          CMP     R1,#PATF        ;SEE IF ALL WORDS OF PATTERN E USED YET
023454  103741                                  BLO     3$              ;BR IF NOT DONE YET
023456                                  ENDTST
023456                                                                                  L10034:
023456  104401                                                                                  TRAP    C$ETST






                                        ;;*********************************************************************
                                        .SBTTL          TEST 12 - REG 17 READ/WRITE BIT TEST
                                        ;*
                                        ;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN F INTO REG 17 :
                                        ;*      DATA PATTERN F = 050,051,050.
                                        ;;*********************************************************************
023460                                  BGNTST
023460                                                                                  T12::
023460  004737  003576                          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
023464  012737  000017  002400                  MOV     #17,REGNUM      ;SET LU REG NO. = 17
023472  012701  002641                          MOV     #PATF,R1        ;GET POINTER TO DATA PAT IN R1
023476                                  3$:
023476                                          BGNSEG
023476  104404                                                                                  TRAP    C$BSEG
023500  111137  002366                          MOVB    (R1),WRIBYT     ;GET A BYTE OF PAT F
023504  004737  003750                          JSR     PC,WRITLU       ;WRITE DATA BYTE INTO REG 17
023510  004737  003672                          JSR     PC,READLU       ;READ DATA BYTE FROM REG 17
023514  143737  002557  002364                  BICB    UPBITS+7,REDBYT ;MASK OUT UNPREDICTABLE BITS
023522  123737  002364  002366                  CMPB    REDBYT,WRIBYT   ;COMPARE BYTE READ TO BYTE WRITTEN
023530  001414                                  BEQ     6$              ;BR IF BYTES MATCH
023532  013737  002366  002404                  MOV     WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
023540  013737  002364  002406                  MOV     REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
023546  004737  004016                          JSR     PC,GETREG       ;GET REGS FOR PRINTOUT
                                        ;REPORT LINE UNIT REG MISCOMPARE
```

```
    023552                                  ERRDF   3,EM3,ERR2
    023552  104455                                                                         TRAP    C$ERDF
    023554  000003                                                                         .WORD   3
    023556  012220                                                                         .WORD   EM3
    023560  014632                                                                         .WORD   ERR2
    023562                          6$:
    023562                                  ENDSEG
    023562                                                                          10000$:
    023562  104405                                                                         TRAP    C$ESEG
    023564  005201                          INC     R1              ;INCREMENT DATA PATTERN POINTER
    023566  020127  002644                  CMP     R1,#PATG        ;SEE IF ALL WORDS OF PATTERN F USED YET
    023572  103741                          BLO     3$              ;BR IF NOT DONE YET
    023574                          ENDTST
    023574                                                                          L10035:
    023574  104401                                                                         TRAP    C$ETST




                                    ;;*************************************************************************
                                    .SBTTL      TEST 13 - MAINTENANCE CLOCK BIT TEST
                                    ;*
                                    ;* FIRST, A MASTER CLEAR IS ISSUED TO INIT ALL REGS. THEN, THE MICROPROCESSOR
                                    ;* IS PLACED IN A LOOP ON AN INSTRUCTION, BY SETTING THE INSTRUCTION IN SEL6
                                    ;* AND SETTING ROMI AND RUN IN BSEL1. THE INSTRUCTION IS ONE WHICH REPETITIVELY
                                    ;* READS LINE UNIT REG 17 INTO BSEL2. THE PDP-11 CAN THEN SCAN BSEL2 TO MONITOR
                                    ;* THE MAINTENANCE CLOCK BIT, MCLK. THE FOLLOWING SEQUENCE IS THEN PERFORMED
                                    ;* TO MONITOR MCLK :
                                    ;* - THE PROGRAM REPEATEDLY CHECKS THE MCLK BIT FOR THE 1 STATE, AND IF IT IS
                                    ;*    NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC (DEPENDING ON THE PROCESSOR)
                                    ;*    AN ERROR IS REPORTED. (THE MAINTENANCE CLOCK HAS A PERIOD OF 41.6 MICRO-
                                    ;*    SEC).
                                    ;* - THE PROGRAM NEXT REPEATEDLY CHECKS THE MCLK BIT FOR THE 0 STATE, AND IF
                                    ;*    IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC AN ERROR IS REPORTED.
                                    ;* - THE PROGRAM NEXT REPEATEDLY CHECKS MCLK BIT FOR THE 1 STATE AGAIN, AND
                                    ;*    IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC, AN ERROR IS REPORTED.
                                    ;*
                                    ;* IF THE P-TABLE FOR THIS UNIT INDICATES THAT THE M8207 RUN SWITCH (E28 SW7)
                                    ;* IS OFF, THE TEST WILL BE SKIPPED.
                                    ;;*************************************************************************
    023576                          BGNTST
    023576                                                                  T13::
    023576  004737  003576                  JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR
    023602  012737  000015  002442          MOV     #13.,TSTNUM     ;SET TEST NO.
    023610  005737  002476                  TST     RUNINH          ;SEE IF RUN SWITCH IS SET
    023614  001020                          BNE     1$              ;BR IF YES, TO RUN TEST
    023616  023727  002444  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
    023624  001012                          BNE     40$             ;BR IF NOT, TO SKIP PRINTING
    023626                                  PRINTF  #FMT19,TSTNUM   ;PRINT MSG TO SAY TEST NOT RUN
    023626  013746  002442                                                                 MOV     TSTNUM,-(SP)
    023632  012746  012007                                                                 MOV     #FMT19,-(SP)
    023636  012746  000002                                                                 MOV     #2,-(SP)
    023642  010600                                                                         MOV     SP,R0
    023644  104417                                                                         TRAP    C$PNTF
    023646  062706  000006                                                                 ADD     #6,SP
    023652  000137  024220          40$:    JMP     A1              ;GO TO SKIP TEST
```

```
023656  012737  000017  002400  1$:     MOV     #17,REGNUM          ;SET REG NO. = 17
023664  012701  000360                  MOV     #360,R1             ;SET INSTRUCTION SOURCE = INBUS REG 17
023670  052701  000002                  BIS     #2,R1               ;SET INSTRUCTION DESTINATION = BSEL2
023674  052701  021000                  BIS     #MVIOX,R1           ;SET REST OF MOVE INSTRUCTION
023700  010137  023710                  MOV     R1,2$               ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
023704  004737  004074                  JSR     PC,LOOPIN           ;GET MICROPROCESSOR LOOPING ON MOVE INSTRUCTION
023710  000000                  2$:     .WORD   0                   ;INSTRUCTION GOES HERE

                                ;-----------------------------------------------------------------------------
                                ; WAIT FOR MCLK BIT TO BE SET TO 1
                                ;-----------------------------------------------------------------------------
023712  005037  002510                  CLR     REG0                ;INIT PROGRAM TIMER
023716  117737  156530  002364  3$:     MOVB    @BSEL2,REDBYT       ;GET REG 17 INTO REDBYT
023724  132737  000002  002364          BITB    #MCLK,REDBYT        ;SEE IF MCLK BIT = 1 YET
023732  001031                          BNE     6$                  ;BR IF MCLK = 1
023734  005237  002510                  INC     REG0                ;INCREMENT TIMER
023740  001366                          BNE     3$                  ;BR IF PROGRAM TIMER DID NOT TIME OUT YET
                                                                    ; (TIME OUT = SEVERAL HUNDRED MILLI-SEC)
023742  012737  000002  002404          MOV     #MCLK,GOODAT        ;SET EXPECTED REG CONTENTS
023750  013737  002364  002406          MOV     REDBYT,BADDAT       ;SET ACTUAL REG CONTENTS
023756  004737  004016                  JSR     PC,GETREG           ;GET REGS FOR PRINTOUT
                                ;REPORT MCLK BIT STUCK AT 0
023762                                  ERRDF   5,EM5,ERR2
023762  104455                                                                          TRAP    C$ERDF
023764  000005                                                                          .WORD   5
023766  012312                                                                          .WORD   EM5
023770  014632                                                                          .WORD   ERR2
                                ;TYPE "PLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON"
023772                                  PRINTF  #FMT24
023772  012746  012040                                                                  MOV     #FMT24,-(SP)
023776  012746  000001                                                                  MOV     #1,-(SP)
024002  010600                                                                          MOV     SP,R0
024004  104417                                                                          TRAP    C$PNTF
024006  062706  000004                                                                  ADD     #4,SP
024012  000137  024220                  JMP     A1                  ;ESCAPE TO END OF TEST

                                ;-----------------------------------------------------------------------------
                                ; WAIT FOR MCLK BIT TO BE CLEARED TO 0
                                ;-----------------------------------------------------------------------------
024016                          6$:
024016  005037  002510                  CLR     REG0                ;INIT PROGRAM TIMER
024022  117737  156424  002364  8$:     MOVB    @BSEL2,REDBYT       ;GET REG 17 INTO REDBYT
024030  132737  000002  002364          BITB    #MCLK,REDBYT        ;SEE IF MCLK BIT = 0 YET
024036  001430                          BEQ     10$                 ;BR IF MCLK = 0
024040  005237  002510                  INC     REG0                ;INCREMENT TIMER
024044  001366                          BNE     8$                  ;BR IF TIMER DID NOT TIME OUT YET
024046  005037  002404                  CLR     GOODAT              ;SET EXPECTED REG CONTENTS
024052  013737  002364  002406          MOV     REDBYT,BADDAT       ;SET ACTUAL REG CONTENTS
024060  004737  004016                  JSR     PC,GETREG           ;GET REGS FOR PRINTOUT
                                ;REPORT MCLK BIT STUCK AT 1
024064                                  ERRDF   6,EM6,ERR2
024064  104455                                                                          TRAP    C$ERDF
024066  000006                                                                          .WORD   6
024070  012343                                                                          .WORD   EM6
024072  014632                                                                          .WORD   ERR2
                                ;TYPE "PLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON"
024074                                  PRINTF  #FMT24
```

```
024074  012746  012040                                                                      MOV     #FMT24,-(SP)
024100  012746  000001                                                                      MOV     #1,-(SP)
024104  010600                                                                              MOV     SP,R0
024106  104417                                                                              TRAP    C$PNTF
024110  062706  000004                                                                      ADD     #4,SP
024114  000137  024220                  JMP     A1                  ;ESCAPE TO END OF TEST

                                ;------------------------------------------------------------------------
                                ; WAIT FOR MCLK BIT TO BE SET TO 1 AGAIN
                                ;------------------------------------------------------------------------
024120                          10$:
024120  005037  002510                  CLR     REG0                ;INIT PROGRAM TIMER
024124  117737  156322  002364  12$:    MOVB    @BSEL2,REDBYT       ;GET REG 17 INTO REDBYT
024132  132737  000002  002364          BITB    #MCLK,REDBYT        ;SEE IF MCLK BIT = 1 YET
024140  001027                          BNE     A1                  ;BE IF MCLK = 1
024142  005237  002510                  INC     REG0                ;INCREMENT TIMER
024146  001366                          BNE     12$                 ;BR IF TIMER DID NOT TIME OUT YET
024150  012737  000002  002404          MOV     #MCLK,GOODAT        ;SET EXPECTED REG CONTENTS
024156  013737  002364  002406          MOV     REDBYT,BADDAT       ;SET ACTUAL REG CONTENTS
024164  004737  004016                  JSR     PC,GETREG           ;GET REGS FOR PRINTOUT
                                ;REPORT MCLK BIT STUCK AT 0
024170                                  ERRDF   5,EM5,ERR2
024170  104455                                                                              TRAP    C$ERDF
024172  000005                                                                              .WORD   5
024174  012312                                                                              .WORD   EM5
024176  014632                                                                              .WORD   ERR2
                                ;TYPE ''PLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON''
024200                                  PRINTF  #FMT24
024200  012746  012040                                                                      MOV     #FMT24,-(SP)
024204  012746  000001                                                                      MOV     #1,-(SP)
024210  010600                                                                              MOV     SP,R0
024212  104417                                                                              TRAP    C$PNTF
024214  062706  000004                                                                      ADD     #4,SP
024220                          A1:
024220  004737  003576                  JSR     PC,MSTCLR           ;ISSUE MASTER CLEAR
024224                          ENDTST
024224                                                                              L10036:
024224  104401                                                                              TRAP    C$ETST





                                ;;**********************************************************************
                                .SBTTL      TEST 14 - EXTENDED REGISTER MASTER CLEAR TEST
                                ;*
                                ;* FIRST, ALL READ/WRITE BITS OF EXTENDED REGS AX0-AX3 ARE SET BY LOADING
                                ;* A DIFFERENT WORD OF PATTERN H INTO EACH REG. THEN, A MASTER CLEAR IS
                                ;* ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF PATTERN I, WHICH
                                ;* CONTAINS THE INITIALIZED STATES OF ALL THE EXTENDED REGS.
                                ;*      PATTERN H = 000,000,377,017,377,377,375,377
                                ;*      PATTERN I = 000,000,000,000,000,103,000,000
                                ;;**********************************************************************
024226                          BGNTST
024226                                                                              T14::
024226  004737  003576                  JSR     PC,MSTCLR           ;ISSUE AN INITIAL MASTER CLEAR
024232  005037  002402                  CLR     AXNUM               ;INIT AX REG BYTE NO. TO 0
```

```
542 024236  012701  002654                    MOV    #PATH,R1          ;INIT DATA PATTERN POINTER
543 024242  112137  002374           1$:      MOVB   (R1)+,WAX15       ;SET BITS TO LOAD INTO LO BYTE
544 024246  112137  002376                    MOVB   (R1)+,WAX16       ;SET BITS TO LOAD INTO HI BYTE
545 024252  004737  004312                    JSR    PC,WRITAX         ;WRITE EXTENDED REG
546 024256  032737  000002  002420            BIT    #WRDYTO,ERROR1    ;SEE IF READY FAILED TO SET
547 024264  001413                            BEQ    8$                ;BR IF NOT
548 024266  012737  000014  002400            MOV    #14,REGNUM        ;SET LU REG NO = 14
549 024274  004737  004016                    JSR    PC,GETREG         ;GET REGS FOR PRINTOUT
550                                  ;REPORT READY NOT SET AFTER AX REG WRITE
551 024300                                    ERRDF  52,EM52,ERR9
    024300  104455                                                                         TRAP    C$ERDF
    024302  000064                                                                         .WORD   52
    024304  014105                                                                         .WORD   EM52
    024306  020630                                                                         .WORD   ERR9
552 024310                                    ESCAPE TST
    024310  104410                                                                         TRAP    C$ESCAPE
    024312  000244                                                                         .WORD   L10037-.
553 024314  062737  000002  002402   8$:      ADD    #2,AXNUM          ;INCR REG BYTE NO.
554 024322  020127  002664                    CMP    R1,#PATI          ;SEE IF ALL REGS WRITTEN YET
555 024326  103745                            BLO    1$                ;BR IF NOT DONE WRITING YET
556 024330  004737  003576                    JSR    PC,MSTCLR         ;ISSUE MASTER CLEAR
557 024334  005037  002402                    CLR    AXNUM             ;INIT EXTENDED REG BYTE NO. TO 0
558 024340  012701  002664                    MOV    #PATI,R1          ;INIT DATA PAT POINTER FOR READS
559 024344  004737  004124           2$:      JSR    PC,READAX         ;READ AN EXTENDED REG
560 024350  032737  000001  002420            BIT    #RRDYTO,ERROR1    ;SEE IF READY FAILED TO SET
561 024356  001413                            BEQ    10$               ;BR IF NOT
562 024360  012737  000014  002400            MOV    #14,REGNUM        ;SET LU REG NO. = 14
563 024366  004737  004016                    JSR    PC,GETREG         ;GET REGS FOR PRINTOUT
564                                  ;REPORT READY NOT SET AFTER AX REG READ
565 024372                                    ERRDF  53,EM53,ERR9
    024372  104455                                                                         TRAP    C$ERDF
    024374  000065                                                                         .WORD   53
    024376  014146                                                                         .WORD   EM53
    024400  020630                                                                         .WORD   ERR9
566 024402                                    ESCAPE TST
    024402  104410                                                                         TRAP    C$ESCAPE
    024404  000152                                                                         .WORD   L10037-.
567 024406  023727  002402  000006   10$:     CMP    AXNUM,#6          ;SEE IF AX3-15
568 024414  001003                            BNE    3$                ;BR IF NOT
569 024416  142737  000372  002370            BICB   #AX315U,RAX15     ;MASK OFF UNPREDICTABLE BITS
570 024424  123711  002370           3$:      CMPB   RAX15,(R1)        ;COMPARE LO BYTE TO EXPECTED VALUE
571 024430  001417                            BEQ    4$                ;BR IF DATA MATCHES
572 024432  005037  002404                    CLR    GOODAT
573 024436  111137  002404                    MOVB   (R1),GOODAT       ;GET EXPECTED DATA BYTE
574 024442  013737  002370  002406            MOV    RAX15,BADDAT      ;GET ACTUAL DATA BYTE
575 024450  004737  004526                    JSR    PC,GETALL         ;GET REGS FOR PRINTOUT
576                                  ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
577 024454                                    ERRDF  2,EM2,ERR3
    024454  104455                                                                         TRAP    C$ERDF
    024456  000002                                                                         .WORD   2
    024460  012161                                                                         .WORD   EM2
    024462  015140                                                                         .WORD   ERR3
578 024464                                    ESCAPE TST
    024464  104410                                                                         TRAP    C$ESCAPE
    024466  000070                                                                         .WORD   L10037-.
579 024470  005237  002402           4$:      INC    AXNUM             ;INCREMENT AX BYTE NO.
580 024474  005201                            INC    R1                ;INCREMENT PAT POINTER
```

```
024476  123711  002372                          CMPB    RAX16,(R1)          ;COMPARE HI BYTE TO EXPECTED VALUE
024502  001417                                  BEQ     6$                  ;BR IF DATA MATCHES
024504  005037  002404                          CLR     GOODAT
024510  111137  002404                          MOVB    (R1),GOODAT         ;GET EXPECTED DATA BYTE
024514  013737  002372  002406                  MOV     RAX16,BADDAT        ;GET ACTUAL DATA BYTE
024522  004737  004526                          JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
                                        ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
024526                                          ERRDF   2,EM2,ERR3
024526  104455                                                                                  TRAP    C$ERDF
024530  000002                                                                                  .WORD   2
024532  012161                                                                                  .WORD   EM2
024534  015140                                                                                  .WORD   ERR3
024536                                          ESCAPE  TST
024536  104410                                                                                  TRAP    C$ESCAPE
024540  000016                                                                                  .WORD   L10037-.
024542  005237  002402                  6$:     INC     AXNUM               ;INCR AX BYTE NO.
024546  005201                                  INC     R1                  ;INCR PAT POINTER
024550  020127  002674                          CMP     R1,#PATJ            ;SEE IF ALL REGS READ YET
024554  103673                                  BLO     2$                  ;BR IF NOT DONE READING YET
024556                                  ENDTST
024556                                                                                  L10037:
024556  104401                                                                                  TRAP    C$ETST






                                        ;;*****************************************************************************
                                        .SBTTL       TEST 15 - EXTENDED REGISTER ADDRESSING TEST
                                        ;*
                                        ;* FIRST, ISSUE A MASTER CLEAR TO PUT REGS INTO INITIALIZED STATES SHOWN IN
                                        ;* PATTERN I. THEN, WRITE A DIFFERENT WORD OF PATTERN J INTO EACH EXTENDED (AX)
                                        ;* REG, AND AFTER EACH WRITE, READ AND COMPARE ALL EXTENDED REGS TO EXPECTED
                                        ;* VALUES.
                                        ;*   PATTERN I = 000,000,000,000,000,103,000,000
                                        ;*   PATTERN J = 000,000,010,002,004,103,001,100
                                        ;;*****************************************************************************
024560                                  BGNTST
024560                                                                                  T15::
024560  004737  003576                          JSR     PC,MSTCLR           ;ISSUE MASTER CLEAR
024564  012701  002664                          MOV     #PATI,R1            ;INIT POINTER TO PAT I
024570  012702  002500                          MOV     #REDDAT,R2          ;INIT POINTER TO EXPECTED DATA AREA
024574  112122                          3$:     MOVB    (R1)+,(R2)+         ;MOVE PAT I INTO REDDAT TABLE
024576  020127  002674                          CMP     R1,#PATJ
024602  103774                                  BLO     3$
024604  005001                                  CLR     R1                  ;INIT INDEX FOR WRITING
024606  010137  002402                  6$:     MOV     R1,AXNUM            ;GET AX BYTE NO. FOR WRITING
024612  116137  002674  002374                  MOVB    PATJ(R1),WAX15      ;SET LO AX BYTE TO BE WRITTEN
024620  116137  002675  002376                  MOVB    PATJ+1(R1),WAX16    ;SET HI AX BYTE TO BE WRITTEN
024626  113761  002374  002500                  MOVB    WAX15,REDDAT(R1)    ;SET EXPECTED LO BYTE IN TABLE
024634  113761  002376  002501                  MOVB    WAX16,REDDAT+1(R1)  ;SET EXPECTED HI BYTE IN TABLE
024642  004737  004312                          JSR     PC,WRITAX           ;WRITE DATA BYTES INTO EXTENDED REG
024646  005003                                  CLR     R3                  ;INIT INDEX FOR READING
024650  010337  002402                  9$:     MOV     R3,AXNUM            ;GET AX BYTE NO. FOR READING
024654  004737  004124                          JSR     PC,READAX           ;READ 2 AX BYTES
024660  023727  002402  000006                  CMP     AXNUM,#6            ;SEE IF AX3-15
024666  001003                                  BNE     10$                 ;BR IF NOT
```

```
024670  142737  000372  002370          BICB    #AX315U,RAX15     ;MASK OFF UNPREDICTABLE BITS
024676  123763  002370  002500  10$:    CMPB    RAX15,REDDAT(R3)  ;COMPARE LO BYTE READ TO EXPECTED
024704  001420                          BEQ     12$               ;BR IF LO BYTE MATCHES EXPECTED
024706  005037  002404                  CLR     GOODAT
024712  116337  002500  002404          MOVB    REDDAT(R3),GOODAT ;GET EXPECTED LO BYTE
024720  013737  002370  002406          MOV     RAX15,BADDAT      ;GET ACTUAL DATA
024726  004737  004526                  JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
024732                                  ERRDF   3,EM3,ERR3
024732  104455                                                                      TRAP    C$ERDF
024734  000003                                                                      .WORD   3
024736  012220                                                                      .WORD   EM3
024740  015140                                                                      .WORD   ERR3
024742                                  ESCAPE  TST
024742  104410                                                                      TRAP    C$ESCAPE
024744  000102                                                                      .WORD   L10040-.
024746  005237  002402          12$:    INC     AXNUM             ;INCR AX BYTE NO.
024752  123763  002372  002501          CMPB    RAX16,REDDAT+1(R3) ;COMPARE HI BYTE READ TO EXPECTED
024760  001420                          BEQ     15$               ;BR IF HI BYTE MATCHES EXPECTED
024762  005037  002404                  CLR     GOODAT
024766  116337  002501  002404          MOVB    REDDAT+1(R3),GOODAT ;GET EXPECTED HI BYTE
024774  013737  002372  002406          MOV     RAX16,BADDAT      ;GET ACTUAL DATA
025002  004737  004526                  JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
025006                                  ERRDF   3,EM3,ERR3
025006  104455                                                                      TRAP    C$ERDF
025010  000003                                                                      .WORD   3
025012  012220                                                                      .WORD   EM3
025014  015140                                                                      .WORD   ERR3
025016                                  ESCAPE  TST
025016  104410                                                                      TRAP    C$ESCAPE
025020  000026                                                                      .WORD   L10040-.
025022  062703  000002          15$:    ADD     #2,R3             ;INCR INDEX FOR READS
025026  020327  000010                  CMP     R3,#10            ;SEE IF ALL AX BYTES READ YET
025032  002706                          BLT     9$                ;BR IF NOT DONE READING YET
025034  062701  000002                  ADD     #2,R1             ;INCR INDEX FOR WRITES
025040  020127  000010                  CMP     R1,#10            ;SEE IF ALL AX BYTES WRITTEN YET
025044  002660                          BLT     6$                ;BR IF NOT DONE WRITING YET
025046                          ENDTST
025046                                                                      L10040:
025046  104401                                                                      TRAP    C$ETST

                                ;;*************************************************************************
                                .SBTTL      TEST 16 - REGS 15,16 / AX2-15,AX2-16 READ/WRITE BIT TEST
                                ;*
                                ;* USING REGS 15,16, THE INDIRECT REGS AX2-15,AX2-16 (USYRT REGS 4,5) ARE
                                ;* WRITTEN AND READ USING EACH WORD OF PATTERN K. AX2-15 IS COMPARED
                                ;* TO THE WORD WRITTEN, AND AX2-16 IS ALWAYS COMPARED TO 103. (AX2-16 IS NOT
                                ;* WRITEABLE).
                                ;*     PATTERN K =
                                ;*       FOR REG 15: 000,377,125,252,001,002,004,010,020,040,100,200,000,000,
                                ;*                   000,000,000,000,000,000,376,375,373,367,357,337,277,177,
                                ;*                   377,377,377,377,377,377,377,377
```

```
                                        ;*      FOR REG 16: 000,377,125,252,000,000,000,000,000,000,000,000,001,002,
                                        ;*                  004,010,020,040,100,200,377,377,377,377,377,377,377,377,
                                        ;*                  376,375,373,367,357,337,277,177.
                                        ;;************************************************************************
025050                                  BGNTST
025050                                                                                          T16::
025050  004737  003576                          JSR     PC,MSTCLR               ;ISSUE MASTER CLEAR
025054  012701  002704                          MOV     #PATK,R1                ;INIT DATA PATTERN POINTER
025060                                  3$:
025060                                          BGNSEG
025060  104404                                                                                  TRAP    C$BSEG
025062  012737  000004  002402                  MOV     #4,AXNUM                ;SET BYTE NO. = 4
025070  111137  002374                          MOVB    (R1),WAX15              ;SET DATA TO WRITE INTO LO BYTE
025074  116137  000001  002376                  MOVB    1(R1),WAX16             ;SET DATA TO WRITE INTO HI BYTE
025102  143737  002565  002374                  BICB    ANBITS+4,WAX15          ;MASK OFF NON-READ/WRITE BITS IN LO BYTE
025110  143737  002566  002376                  BICB    ANBITS+5,WAX16          ;MASK OFF NON-READ/WRITE BITS IN HI BYTE
025116  004737  004312                          JSR     PC,WRITAX               ;LOAD DATA INTO AX2-15,AX2-16
025122  004737  004124                          JSR     PC,READAX               ;READ AX2-15 AND AX2-16
025126  123737  002370  002374                  CMPB    RAX15,WAX15             ;COMPARE LO BYTE DATA READ
025134  001416                                  BEQ     6$                      ;BR IF DATA MATCHES
025136  013737  002374  002404                  MOV     WAX15,GOODAT            ;SET EXPECTED DATA
025144  013737  002370  002406                  MOV     RAX15,BADDAT            ;SET ACTUAL DATA
025152  004737  004526                          JSR     PC,GETALL               ;GET REGS FOR PRINTOUT
                                        ;REPORT REG MISCOMPARE
025156                                          ERRDF   3,EM3,ERR3
025156  104455                                                                                  TRAP    C$ERDF
025160  000003                                                                                  .WORD   3
025162  012220                                                                                  .WORD   EM3
025164  015140                                                                                  .WORD   ERR3
025166                                          ESCAPE  SEG
025166  104410                                                                                  TRAP    C$ESCAPE
025170  000052                                                                                  .WORD   10000$-.
025172  005237  002402                  6$:     INC     AXNUM                   ;SET AX BYTE NO. = 5
025176  123737  002372  002671                  CMPB    RAX16,PATI+5            ;COMPARE HI BYTE DATA READ
025204  001416                                  BEQ     9$                      ;BR IF DATA MATCHES
025206  005037  002404                          CLR     GOODAT
025212  113737  002671  002404                  MOVB    PATI+5,GOODAT           ;SET EXPECTED DATA
025220  013737  002372  002406                  MOV     RAX16,BADDAT            ;SET ACTUAL DATA
025226  004737  004526                          JSR     PC,GETALL               ;GET REGS FOR PRINTOUT
                                        ;REPORT REG MISCOMPARE
025232                                          ERRDF   3,EM3,ERR3
025232  104455                                                                                  TRAP    C$ERDF
025234  000003                                                                                  .WORD   3
025236  012220                                                                                  .WORD   EM3
025240  015140                                                                                  .WORD   ERR3
025242                                  9$:
025242                                          ENDSEG
025242                                                                                  10000$:
025242  104405                                                                                  TRAP    C$ESEG
025244  062701  000002                          ADD     #2,R1                   ;INCR PATTERN POINTER
025250  020127  003014                          CMP     R1,#PATL                ;SEE IF ALL DATA WRITTEN YET
025254  103701                                  BLO     3$                      ;BR IF NOT DONE YET
025256                                  ENDTST
025256                                                                                  L10041:
025256  104401                                                                                  TRAP    C$ETST
```

```
                                        ;;*****************************************************************************
                                        .SBTTL       TEST 17 - AX0-15,AX0-16 READ/WRITE BIT TEST
                                        ;*
                                        ;* IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
                                        ;* ARE PERFORMED IN REGS AX0-15,AX0-16 USING EACH WORD OF PATTERN L.
                                        ;* ANY BITS IN AX0-15,AX0-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
                                        ;* IN THE EXPECTED VALUE BEFORE COMPARISON.
                                        ;*     PATTERN L =
                                        ;*        FOR REG 15: 000,377,000
                                        ;*        FOR REG 16: 000,377,000.
                                        ;;*****************************************************************************
025260                                  BGNTST
025260                                                                                      T17::
025260  004737  003576                          JSR     PC,MSTCLR          ;ISSUE MASTER CLEAR
025264  012701  003014                          MOV     #PATL,R1           ;INIT DATA PATTERN POINTER
025270                                  3$:
025270                                          BGNSEG
025270  104404                                                                                  TRAP    C$BSEG
025272  012737  000000  002402                  MOV     #0,AXNUM           ;SET BYTE NO. = 0
025300  111137  002374                          MOVB    (R1),WAX15         ;SET DATA TO WRITE INTO LO BYTE
025304  116137  000001  002376                  MOVB    1(R1),WAX16        ;SET DATA TO WRITE INTO HI BYTE
025312  143737  002561  002374                  BICB    ANBITS+0,WAX15     ;MASK OFF NON-READ/WRITE BITS IN LO BYTE
025320  143737  002562  002376                  BICB    ANBITS+1,WAX16     ;MASK OFF NON-READ/WRITE BITS IN HI BYTE
025326  004737  004312                          JSR     PC,WRITAX          ;LOAD DATA INTO AX0-15,AX0-16
025332  004737  004124                          JSR     PC,READAX          ;READ AX0-15 AND AX0-16
025336  123737  002370  002374                  CMPB    RAX15,WAX15        ;COMPARE LO BYTE DATA READ
025344  001416                                  BEQ     6$                 ;BR IF DATA MATCHES
025346  013737  002374  002404                  MOV     WAX15,GOODAT       ;SET EXPECTED DATA
025354  013737  002370  002406                  MOV     RAX15,BADDAT       ;SET ACTUAL DATA
025362  004737  004526                          JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                        ;REPORT REG MISCOMPARE
025366                                          ERRDF   3,EM3,ERR3
025366  104455                                                                                  TRAP    C$ERDF
025370  000003                                                                                  .WORD   3
025372  012220                                                                                  .WORD   EM3
025374  015140                                                                                  .WORD   ERR3
025376                                          ESCAPE  SEG
025376  104410                                                                                  TRAP    C$ESCAPE
025400  000046                                                                                  .WORD   10000$-.
025402  005237  002402                  6$:     INC     AXNUM              ;SET AX BYTE NO. = 1
025406  123737  002372  002376                  CMPB    RAX16,WAX16        ;COMPARE HI BYTE DATA READ
025414  001414                                  BEQ     9$                 ;BR IF DATA MATCHES
025416  013737  002376  002404                  MOV     WAX16,GOODAT       ;SET EXPECTED DATA
025424  013737  002372  002406                  MOV     RAX16,BADDAT       ;SET ACTUAL DATA
025432  004737  004526                          JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                        ;REPORT REG MISCOMPARE
025436                                          ERRDF   3,EM3,ERR3
025436  104455                                                                                  TRAP    C$ERDF
025440  000003                                                                                  .WORD   3
025442  012220                                                                                  .WORD   EM3
025444  015140                                                                                  .WORD   ERR3
025446                                  9$:
025446                                          ENDSEG
025446                                                                                  10000$:
```

```
025446  104405                                                                          TRAP    C$ESEG
025450  062701  000002                  ADD     #2,R1           ;INCR PATTERN POINTER
025454  020127  003022                  CMP     R1,#PATM        ;SEE IF ALL DATA WRITTEN YET
025460  103703                          BLO     3$              ;BR IF NOT DONE YET
025462                          ENDTST
025462                                                                          L10042:
025462  104401                                                                          TRAP    C$ETST





                        ;;*************************************************************************
                        .SBTTL          TEST 18 - AX1-15,AX1-16 READ/WRITE BIT TEST
                        ;*
                        ;* IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
                        ;* ARE PERFORMED IN REGS AX1-15,AX1-16 USING EACH WORD OF PATTERN K.
                        ;* ANY BITS IN AX1-15,AX1-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
                        ;* IN THE EXPECTED VALUE BEFORE COMPARISON.
                        ;;*************************************************************************
025464                  BGNTST
025464                                                                          T18::
025464  004737  003576                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
025470  012701  002704                  MOV     #PATK,R1        ;INIT DATA PATTERN POINTER
025474                  3$:
025474                                  BGNSEG
025474  104404                                                                          TRAP    C$BSEG
025476  012737  000002  002402          MOV     #2,AXNUM        ;SET BYTE NO. = 2
025504  111137  002374                  MOVB    (R1),WAX15      ;SET DATA TO WRITE INTO LO BYTE
025510  116137  000001  002376          MOVB    1(R1),WAX16     ;SET DATA TO WRITE INTO HI BYTE
025516  143737  002563  002374          BICB    ANBITS+2,WAX15  ;MASK OFF NON-READ/WRITE BITS IN LO BYTE
025524  143737  002564  002376          BICB    ANBITS+3,WAX16  ;MASK OFF NON-READ/WRITE BITS IN HI BYTE
025532  004737  004312                  JSR     PC,WRITAX       ;LOAD DATA INTO AX1-15,AX1-16
025536  004737  004124                  JSR     PC,READAX       ;READ AX1-15 AND AX1-16
025542  123737  002370  002374          CMPB    RAX15,WAX15     ;COMPARE LO BYTE DATA READ
025550  001416                          BEQ     6$              ;BR IF DATA MATCHES
025552  013737  002374  002404          MOV     WAX15,GOODAT    ;SET EXPECTED DATA
025560  013737  002370  002406          MOV     RAX15,BADDAT    ;SET ACTUAL DATA
025566  004737  004526                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT REG MISCOMPARE
025572                                  ERRDF   3,EM3,ERR3
025572  104455                                                                          TRAP    C$ERDF
025574  000003                                                                          .WORD   3
025576  012220                                                                          .WORD   EM3
025600  015140                                                                          .WORD   ERR3
025602                                  ESCAPE  SEG
025602  104410                                                                          TRAP    C$ESCAPE
025604  000046                                                                          .WORD   10000$-.
025606  005237  002402          6$:     INC     AXNUM           ;SET AX BYTE NO. = 3
025612  123737  002372  002376          CMPB    RAX16,WAX16     ;COMPARE HI BYTE DATA READ
025620  001414                          BEQ     9$              ;BR IF DATA MATCHES
025622  013737  002376  002404          MOV     WAX16,GOODAT    ;SET EXPECTED DATA
025630  013737  002372  002406          MOV     RAX16,BADDAT    ;SET ACTUAL DATA
025636  004737  004526                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT REG MISCOMPARE
025642                                  ERRDF   3,EM3,ERR3
025642  104455                                                                          TRAP    C$ERDF
```

```
    025644  000003                                                                           .WORD   3
    025646  012220                                                                           .WORD   EM3
    025650  015140                                                                           .WORD   ERR3
    025652                                  9$:
    025652                                          ENDSEG
    025652                                                                           10000$:
    025652  104405                                                                           TRAP    C$ESEG
    025654  062701  000002                          ADD     #2,R1           ;INCR PATTERN POINTER
    025660  020127  003014                          CMP     R1,#PATL        ;SEE IF ALL DATA WRITTEN YET
    025664  103703                                  BLO     3$              ;BR IF NOT DONE YET
    025666                                  ENDTST
    025666                                                                           L10043:
    025666  104401                                                                           TRAP    C$ETST





                                        ;;*************************************************************************
                                        .SBTTL          TEST 19 - AX3-15,AX3-16 READ/WRITE BIT TEST
                                        ;*
                                        ;* IN THIS TEST A MASTER CLEAR IS DONE AND THEN A WRITE, READ, AND COMPARE ARE
                                        ;* PERFORMED IN REGS AX3-15,AX3-16 USING EACH WORD OF PATTERN V FOR WRITING,
                                        ;* AND PATTERN U FOR COMPARING.
                                        ;* ANY BITS IN AX3-15,AX3-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
                                        ;* IN THE EXPECTED VALUE BEFORE COMPARISON.
                                        ;*    PATTERN V =
                                        ;*        FOR REG 15 : 000,333,331,323,313,233,133,000,000,000,000,
                                        ;*                     000,000,000,000,000,000,000,000
                                        ;*        FOR REG 16 : 000,000,000,000,000,000,000,001,002,004,040,
                                        ;*                     100,200,346,345,343,307,247,147
                                        ;*    PATTERN U =
                                        ;*        FOR REG 15 : 000,001,013,011,021,101,301,000,000,000,000,
                                        ;*                     000,000,000,000,000,000,000,000
                                        ;*        FOR REG 16 : 000,000,000,000,000,000,000,001,002,004,040,
                                        ;*                     100,200,346,345,343,307,247,147
                                        ;;*************************************************************************
    025670                              BGNTST
    025670                                                                           T19::
    025670  004737  003576                          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
    025674  142777  000010  154546                  BICB    #LULOOP,@BSEL1  ;CLEAR LULOOP
    025702  012702  003152                          MOV     #PATV,R2        ;INIT PATTERN V POINTER
    025706  012701  003104                          MOV     #PATU,R1        ;INIT PATTERN U POINTER
    025712                              3$:
    025712                                          BGNSEG
    025712  104404                                                                           TRAP    C$BSEG
    025714  012737  000006  002402                  MOV     #6,AXNUM        ;SET BYTE NO. = 6
    025722  111237  002374                          MOVB    (R2),WAX15      ;SET DATA TO WRITE INTO LO BYTE
    025726  116237  000001  002376                  MOVB    1(R2),WAX16     ;SET DATA TO WRITE INTO HI BYTE
    025734  004737  004312                          JSR     PC,WRITAX       ;LOAD DATA INTO AX3-15,AX3-16
    025740  004737  004124                          JSR     PC,READAX       ;READ AX3-15 AND AX3-16
    025744  132737  000001  002374                  BITB    #TEST,WAX15     ;SEE IF AN INTERFACE IS SELECTED
    025752  001003                                  BNE     4$              ;BR IF YES
    025754  142737  000372  002370                  BICB    #AX315U,RAX15   ;MASK OFF UNPREDICTABLE BITS
    025762  142737  000040  002370  4$:             BICB    #C32BCC,RAX15   ;CLEAR CRC32 BCC BIT
    025770  123711  002370                          CMPB    RAX15,(R1)      ;COMPARE LO BYTE DATA READ
    025774  001417                                  BEQ     6$              ;BR IF DATA MATCHES
```

```
 025776  005037  002404                           CLR     GOODAT
 026002  111137  002404                           MOVB    (R1),GOODAT        ;SET EXPECTED DATA
 026006  013737  002370  002406                   MOV     RAX15,BADDAT       ;SET ACTUAL DATA
 026014  004737  004526                           JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                          ;REPORT REG MISCOMPARE
 026020                                           ERRDF   3,EM3,ERR3
 026020  104455                                                                                   TRAP    C$ERDF
 026022  000003                                                                                   .WORD   3
 026024  012220                                                                                   .WORD   EM3
 026026  015140                                                                                   .WORD   ERR3
 026030                                           ESCAPE  SEG
 026030  104410                                                                                   TRAP    C$ESCAPE
 026032  000052                                                                                   .WORD   10000$-.
 026034  005237  002402                   6$:     INC     AXNUM              ;SET AX BYTE NO. = 7
 026040  123761  002372  000001                   CMPB    RAX16,1(R1)        ;COMPARE HI BYTE DATA READ
 026046  001416                                   BEQ     9$                 ;BR IF DATA MATCHES
 026050  005037  002404                           CLR     GOODAT
 026054  116137  000001  002404                   MOVB    1(R1),GOODAT       ;SET EXPECTED DATA
 026062  013737  002372  002406                   MOV     RAX16,BADDAT       ;SET ACTUAL DATA
 026070  004737  004526                           JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                          ;REPORT REG MISCOMPARE
 026074                                           ERRDF   3,EM3,ERR3
 026074  104455                                                                                   TRAP    C$ERDF
 026076  000003                                                                                   .WORD   3
 026100  012220                                                                                   .WORD   EM3
 026102  015140                                                                                   .WORD   ERR3
 026104                                   9$:
 026104                                           ENDSEG
 026104                                                                                   10000$:
 026104  104405                                                                                   TRAP    C$ESEG
 026106  062702  000002                           ADD     #2,R2              ;INCR PATTERN V POINTER
 026112  062701  000002                           ADD     #2,R1              ;INCR PATTERN U POINTER
 026116  020127  003152                           CMP     R1,#PATV           ;SEE IF ALL DATA WRITTEN YET
 026122  103673                                   BLO     3$                 ;BR IF NOT DONE YET
 026124                                   ENDTST
 026124                                                                                   L10044:
 026124  104401                                                                                   TRAP    C$ETST






                                          ;;*********************************************************************
                                          .SBTTL        TEST 20 - REG 17 - AX2-16 READ/WRITE, MASTER CLEAR TEST
                                          ;*
                                          ;* THIS TEST CONSISTS OF 2 SUBTESTS. IN THE FIRST SUBTEST, EACH BYTE OF PAT O
                                          ;* IS WRITTEN INTO REG 17 AND AFTER EACH WRITE, AX2-16 IS READ AND COMPARED
                                          ;* TO A BYTE OF PAT P.
                                          ;*   PATTERN O = 000,041,004,010,020,040,100,101,200,201,300,111,301,375
                                          ;*   PATTERN P = 000,113,200,040,020,010,001,104,007,105,007,144,107,157
                                          ;* IN THE SECOND SUBTEST, REG 17 IS LOADED WITH 375, A MASTER CLEAR IS ISSUED,
                                          ;* AND AX2-16 IS COMPARED TO ITS INITIALIZED STATE (103).
                                          ;;*********************************************************************
 026126                                   BGNTST
 026126                                                                                   T20::
 026126  004737  003576                           JSR     PC,MSTCLR          ;ISSUE MASTER CLEAR
 026132  012701  003050                           MOV     #PATO,R1           ;INIT PAT O POINTER
```

```
026136  012702  003066                  MOV     #PATP,R2          ;INIT PAT P POINTER
026142  012737  000017  002400          MOV     #17,REGNUM        ;SET LU REG NO. = 17
026150  012737  000005  002402          MOV     #5,AXNUM          ;SET AX BYTE NO. = 5 FOR AX2-16
                                ;-----------------------------------------------------------------
                                ; WRITE REG 17, READ AND COMPARE AX2-16
                                ;-----------------------------------------------------------------
026156                                  BGNSUB
026156                                                                             T20.1:
026156  104402                                                                             TRAP    C$BSUB
026160                          3$:
026160                                  BGNSEG
026160  104404                                                                             TRAP    C$BSEG
026162  111137  002366                  MOVB    (R1),WRIBYT       ;SET BYTE TO BE WRITTEN
026166  004737  003750                  JSR     PC,WRITLU         ;WRITE DATA BYTE INTO REG 17
026172  004737  004124                  JSR     PC,READAX         ;READ AX2
026176  123712  002372                  CMPB    RAX16,(R2)        ;COMPARE AX2-16 TO EXPECTED DATA
026202  001421                          BEQ     6$                ;BR IF DATA MATCHES
026204  005037  002410                  CLR     LOADAT
026210  111137  002410                  MOVB    (R1),LOADAT       ;SET DATA WHICH WAS WRITTEN
026214  005037  002404                  CLR     GOODAT
026220  111237  002404                  MOVB    (R2),GOODAT       ;SET EXPECTED DATA READ
026224  013737  002372  002406          MOV     RAX16,BADDAT      ;SET ACTUAL DATA READ
026232  004737  004526                  JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
026236                                  ERRDF   3,EM3,ERR5
026236  104455                                                                             TRAP    C$ERDF
026240  000003                                                                             .WORD   3
026242  012220                                                                             .WORD   EM3
026244  016300                                                                             .WORD   ERR5
026246                          6$:
026246                                  ENDSEG
026246                                                                             10000$:
026246  104405                                                                             TRAP    C$ESEG
026250  005201                          INC     R1                ;INCR PAT O POINTER
026252  005202                          INC     R2                ;INCR PAT P POINTER
026254  020127  003066                  CMP     R1,#PATP          ;SEE IF ALL BYTES LOADED YET
026260  103737                          BLO     3$                ;BR IF NOT
026262                                  ENDSUB
026262                                                                             L10046:
026262  104403                                                                             TRAP    C$ESUB
                                ;-----------------------------------------------------------------
                                ; LOAD REG 17, DO MASTER CLEAR, READ AND COMPARE AX2-16
                                ;-----------------------------------------------------------------
026264                                  BGNSUB
026264                                                                             T20.2:
026264  104402                                                                             TRAP    C$BSUB
026266  112737  000375  002366          MOVB    #375,WRIBYT       ;SET DATA TO BE LOADED
026274  004737  003750                  JSR     PC,WRITLU         ;LOAD DATA INTO REG 17
026300  004737  003576                  JSR     PC,MSTCLR         ;PERFORM MASTER CLEAR
026304  004737  004124                  JSR     PC,READAX         ;READ AX2-15,AX2-16
026310  123737  002372  002671          CMPB    RAX16,PATI+5      ;SEE IF AX2-16 WAS INIT'D CORRECTLY
026316  001416                          BEQ     6$                ;BR IF YES
026320  005037  002404                  CLR     GOODAT
026324  113737  002671  002404          MOVB    PATI+5,GOODAT     ;SET EXPECTED DATA
026332  013737  002372  002406          MOV     RAX16,BADDAT      ;SET ACTUAL DATA
026340  004737  004526                  JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
```

```
026344                                  ERRDF   2,EM2,ERR3
026344  104455                                                                  TRAP    C$ERDF
026346  000002                                                                  .WORD   2
026350  012161                                                                  .WORD   EM2
026352  015140                                                                  .WORD   ERR3
026354                          6$:
026354                                  ENDSUB
026354                                                                  L10047:
026354  104403                                                                  TRAP    C$ESUB
026356                          ENDTST
026356                                                                  L10045:
026356  104401                                                                  TRAP    C$ETST




                                ;;*********************************************************************
                                .SBTTL       TEST 21 - TRANSMITTER BUFFER DATA TEST
                                ;* A MASTER CLEAR IS DONE FIRST, AND THEN A BYTE OF PATTERN N IS LOADED INTO
                                ;* REG 11 AND THE NEXT BYTE IS LOADED TWICE INTO REG 10. THE PROGRAM THEN WAITS
                                ;* AT LEAST 50 MICRO-SEC, AND THEN IT READS AND COMPARES AX1-15 TO THE BYTE
                                ;* WHICH WAS LOADED INTO REG 10, AND IT READS AND COMPARES AX1-16 TO THE BYTE
                                ;* WHICH WAS LOADED INTO REG 11. THIS PROCESS IS REPEATED (INCLUDING THE MASTER
                                ;* CLEAR) FOR EACH PAIR OF BYTES IN PATTERN N.
                                ;*    PATTERN N =
                                ;*       FOR REG 10: 000,125,252,377,000,000,000
                                ;*       FOR REG 11: 000,000,000,000,005,012,017
                                ;;*********************************************************************
026360                          BGNTST
026360                                                                  T21::
026360  012701  003032                  MOV     #PATN,R1        ;INIT PATTERN POINTER
026364  012737  026626  002362          MOV     #A2,RETADR      ;SET SUBROUTINE ERROR RETURN ADDRESS
026372                                  BGNSUB
026372                                                                  T21.1:
026372  104402                                                                  TRAP    C$BSUB
026374  004737  003576          3$:     JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
026400  004737  004662                  JSR     PC,OSIRDY       ;CHECK ORDY AND OCOR FOR EXPECTED STATES
026404  000001                          1
026406  012737  000011  002400          MOV     #11,REGNUM      ;SET LU REG NO. = 11
026414  111137  002366                  MOVB    (R1),WRIBYT     ;SET DATA BYTE TO BE WRITTEN
026420  004737  003750                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 11
026424  012737  000010  002400          MOV     #10,REGNUM      ;SET LU REG NO. = 10
026432  116137  000001  002366          MOVB    1(R1),WRIBYT    ;SET DATA BYTE TO BE WRITTEN
026440  004737  003750                  JSR     PC,WRITLU       ;WRITE BYTE INTO REG 10
026444  004737  003750                  JSR     PC,WRITLU       ;WRITE IT AGAIN (SO 2 ENTRIES ARE IN SILO)
026450  004737  005146                  JSR     PC,WAIT50       ;WAIT FOR SILO DATA TO RIPPLE
026454  004737  004662                  JSR     PC,OSIRDY       ;CHECK ORDY AND OCOR FOR EXPECTED STATES
026460  000003                          3
026462  012737  000002  002402          MOV     #2,AXNUM        ;SET BYTE NO. FOR AX1-15
026470  004737  004124                  JSR     PC,READAX       ;READ AX1-15, AX1-16
026474  123761  002370  000001          CMPB    RAX15,1(R1)     ;COMPARE AX1-15 TO EXPECTED
026502  001420                          BEQ     6$              ;BR IF MATCH
026504  005037  002404                  CLR     GOODAT
026510  116137  000001  002404          MOVB    1(R1),GOODAT    ;SET EXPECTED DATA
026516  013737  002370  002406          MOV     RAX15,BADDAT    ;SET ACTUAL DATA
026524  004737  004526                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
```

```
                                              ;REPORT REG MISCOMPARE
     026530                                           ERRDF   3,EM3,ERR3
     026530   104455                                                                            TRAP    C$ERDF
     026532   000003                                                                            .WORD   3
     026534   012220                                                                            .WORD   EM3
     026536   015140                                                                            .WORD   ERR3
     026540                                           ESCAPE  SUB
     026540   104410                                                                            TRAP    C$ESCAPE
     026542   000064                                                                            .WORD   L10051-.
     026544   005237   002402              6$:        INC     AXNUM          ;INCR AX BYTE NO.
     026550   123711   002372                         CMPB    RAX16,(R1)     ;COMPARE AX1-16 TO EXPECTED
     026554   001417                                  BEQ     9$             ;BR IF MATCH
     026556   005037   002404                         CLR     GOODAT
     026562   111137   002404                         MOVB    (R1),GOODAT    ;SET EXPECTED DATA
     026566   013737   002372   002406                MOV     RAX16,BADDAT   ;SET ACTUAL DATA
     026574   004737   004526                         JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
                                              ;REPORT REG MISCOMPARE
     026600                                           ERRDF   3,EM3,ERR3
     026600   104455                                                                            TRAP    C$ERDF
     026602   000003                                                                            .WORD   3
     026604   012220                                                                            .WORD   EM3
     026606   015140                                                                            .WORD   ERR3
     026610                                           ESCAPE  SUB
     026610   104410                                                                            TRAP    C$ESCAPE
     026612   000014                                                                            .WORD   L10051-.
     026614   062701   000002              9$:        ADD     #2,R1          ;INCR DATA PATTERN POINTER
     026620   020127   003050                         CMP     R1,#PATO       ;SEE IF ALL DATA BYTES WRITTEN YET
     026624   103663                                  BLO     3$             ;BR IF NOT DONE YET
     026626                                A2:
     026626                                           ENDSUB
     026626                                                                           L10051:
     026626   104403                                                                            TRAP    C$ESUB
     026630                                ENDTST
     026630                                                                           L10050:
     026630   104401                                                                            TRAP    C$ETST





                                       ;;*******************************************************************************
                                       .SBTTL      TEST 22 - TRANSMITTER BUFFER SEQUENCING TEST
                                       ;*
                                       ;* FIRST, A MASTER CLEAR IS DONE, AND THE PROGRAM CHECKS FOR ORDY=1, OCOR=0.
                                       ;* THEN, 2 TSOM CHARS ARE LOADED INTO THE TX SILO, AND ALLOWED TO RIPPLE
                                       ;* DOWN TO THE OUTPUT. THE PROGRAM CHECKS FOR ORDY=1, OCOR=1.
                                       ;* NEXT, THE PROGRAM CYCLES THE STEPLU BIT UNTIL OCOR=0 AGAIN, AND CHECKS FOR
                                       ;* THIS TO OCCUR WITHIN 3 CYCLES.
                                       ;* THE SILO IS THEN FILLED WITH 64 BYTES OF A 256-BYTE BINARY COUNT PATTERN
                                       ;* (000-377) AND THE PROGRAM CHECKS FOR ORDY=0 AFTER THE 64TH CHAR IS LOADED.
                                       ;* THE PROGRAM CYCLES STEPLU FOR 8 CYCLES AND CHECKS THAT AFTER THE 8TH, ORDY=1.
                                       ;* AX1-15 IS READ AND COMPARED TO EXPECTED DATA.
                                       ;* THE REST OF THE BINARY COUNT DATA BYTES ARE LOADED, CYCLED 8 CLOCKS, READ AND
                                       ;* COMPARED, A BYTE AT A TIME. UPON COMPLETION, THE SILO IS CHECKED TO BE EMPTY
                                       ;* WITH ORDY=1, OCOR=0.
                                       ;;*******************************************************************************
     026632                            BGNTST
```

```
026632                                                        T22::
026632  012737  027306  002362          MOV     #A3,RETADR      ;SET SUBR ERROR RETURN ADDR
                                ;----------------------------------------------------------------
                                ; SET MASTER CLEAR, CHECK FOR ORDY=1, OCOR=0
                                ;----------------------------------------------------------------
026640  004737  003576                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
026644  004737  004662                  JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
026650  000001                          1
                                ;----------------------------------------------------------------
                                ; LOAD 2 SOM CHARS, ALLOW SILO TO RIPPLE, CHK ORDY=1, OCOR=1
                                ;----------------------------------------------------------------
026652  012737  000400  002422          MOV     #TXSOM,TXWORD   ;SET DATA TO WRITE INTO SILO
026660  004737  005174                  JSR     PC,LDTXSI       ;LOAD THE SILO WITH SOM
026664  004737  005174                  JSR     PC,LDTXSI       ;LOAD ANOTHER SOM
026670  004737  005146                  JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE
026674  004737  004662                  JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=1
026700  000003                          3
                                ;----------------------------------------------------------------
                                ; CLOCK LINE UNIT, CHK FOR OCOR = 0 WITHIN 3 CYCLES
                                ;----------------------------------------------------------------
026702  005001                          CLR     R1              ;INIT CYCLE COUNTER TO 0
026704  012737  000017  002400          MOV     #17,REGNUM      ;SET REG NO. = 17
026712  004737  005254          3$:     JSR     PC,STPLU        ;STEP LU 1 CYCLE
026716  000001                          1
026720  004737  003672                  JSR     PC,READLU       ;READ REG 17
026724  132737  000020  002364          BITB    #OCOR,REDBYT    ;SEE IF OCOR = 0 YET
026732  001404                          BEQ     6$              ;BR IF OCOR = 0
026734  005201                          INC     R1              ;INCR CYCLE COUNT
026736  020127  000003                  CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
026742  002763                          BLT     3$              ;BR IF NO
026744  004737  004662          6$:     JSR     PC,OSIRDY       ;CHK ORDY=1, OCOR=0
026750  000001                          1
                                ;----------------------------------------------------------------
                                ; LOAD 64 BINARY COUNT CHARS INTO SILO, CHK ORDY=0
                                ;----------------------------------------------------------------
026752  005003                          CLR     R3              ;INIT PATTERN FOR WRITING
026754  010337  002422          8$:     MOV     R3,TXWORD       ;SET DATA TO BE WRITTEN
026760  004737  005174                  JSR     PC,LDTXSI       ;LOAD DATA CHAR INTO TX SILO
026764  004737  005146                  JSR     PC,WAIT50       ;WAIT FOR DATA TO RIPPLE IN SILO
026770  020327  000077                  CMP     R3,#63.         ;SEE IF 64TH CHAR JUST LOADED
026774  001004                          BNE     9$              ;BR IF NO
026776  012737  000002  027020          MOV     #2,14$          ;SET UP TO CHK ORDY=0,OCOR=1
027004  000403                          BR      12$
027006  012737  000003  027020  9$:     MOV     #3,14$          ;SET UP TO CHK ORDY=1,OCOR=1
027014  004737  004662          12$:    JSR     PC,OSIRDY       ;CHK ORDY, OCOR
027020  000000                  14$:    .WORD   0
027022  005203                          INC     R3              ;INCR PATTERN FOR WRITES
027024  020327  000100                  CMP     R3,#64.         ;SEE IF 64 CHARS LOADED YET
027030  002751                          BLT     8$              ;BR IF NO
                                ;----------------------------------------------------------------
                                ; CLOCK LINE UNIT, CHECK ORDY = 1 WITHIN 8 CYCLES
                                ;----------------------------------------------------------------
027032  012737  000011  002400          MOV     #11,REGNUM      ;SET REG NO. = 11
027040  005001                          CLR     R1              ;INIT CYCLE COUNT
027042  004737  005254          16$:    JSR     PC,STPLU        ;CLOCK LU FOR 1 CYCLE
027046  000001                          1
027050  004737  003672                  JSR     PC,READLU       ;READ REG 11
```

```
027054  132737  000020  002364          BITB    #ORDY,REDBYT        ;SEE IF ORDY = 1 YET
027062  001004                          BNE     19$                 ;BR IF YES
027064  005201                          INC     R1                  ;INCR CYCLE COUNT
027066  020127  000010                  CMP     R1,#8.              ;SEE IF 8 CYCLES YET
027072  002763                          BLT     16$                 ;BR IF NOT YET
027074  004737  004662          19$:    JSR     PC,OSIRDY           ;CHK ORDY = 1, OCOR = 1
027100  000003                          3
                                ;-----------------------------------------------------------------
                                ; READ AND COMPARE FIRST CHARACTER IN AX1-15
                                ;-----------------------------------------------------------------
027102  005004                          CLR     R4                  ;INIT PATTERN FOR READING
027104  012737  000002  002402          MOV     #2,AXNUM            ;SET AX BYTE NO. FOR AX1-15
027112  004737  004124                  JSR     PC,READAX           ;READ AX1-15
027116  123704  002370                  CMPB    RAX15,R4            ;COMPARE AX1-15 TO EXPECTED
027122  001415                          BEQ     20$                 ;BR IF MATCH
027124  010437  002404                  MOV     R4,GOODAT           ;SET EXPECTED DATA
027130  013737  002370  002406.         MOV     RAX15,BADDAT        ;SET ACTUAL DATA
027136  004737  004526                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
027142                                  ERRDF   3,EM3,ERR3
027142  104455                                                                          TRAP    C$ERDF
027144  000003                                                                          .WORD   3
027146  012220                                                                          .WORD   EM3
027150  015140                                                                          .WORD   ERR3
027152                                  ESCAPE  TST
027152  104410                                                                          TRAP    C$ESCAPE
027154  000132                                                                          .WORD   L10052-.
027156                          20$:
                                ;-----------------------------------------------------------------
                                ; LOAD AND COMPARE REST OF CHARS, MONITOR ORDY, OCOR
                                ;-----------------------------------------------------------------
027156  020327  000377          24$:    CMP     R3,#255.            ;SEE IF ALL CHARS LOADED YET
027162  003010                          BGT     26$                 ;BR IF YES
027164  010337  002422                  MOV     R3,TXWORD           ;SET DATA TO BE WRITTEN
027170  004737  005174                  JSR     PC,LDTXSI           ;LOAD DATA CHAR INTO TX SILO
027174  005203                          INC     R3                  ;INCR DATA TO BE WRITTEN
027176  004737  004662                  JSR     PC,OSIRDY           ;CHK ORDY=0, OCOR=1
027202  000002                          2
027204  005204                  26$:    INC     R4                  ;INCR PAT FOR READING
027206  004737  005254                  JSR     PC,STPLU            ;CLOCK LINE UNIT FOR 8 CYCLES
027212  000010                          8.
027214  004737  004124                  JSR     PC,READAX           ;READ AX1-15
027220  123704  002370                  CMPB    RAX15,R4            ;COMPARE AX1-15 TO EXPECTED
027224  001415                          BEQ     27$                 ;BR IF MATCH
027226  010437  002404                  MOV     R4,GOODAT           ;SET EXPECTED DATA
027232  013737  002370  002406          MOV     RAX15,BADDAT        ;SET ACTUAL DATA
027240  004737  004526                  JSR     PC,GETALL           ;GET REGS FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
027244                                  ERRDF   3,EM3,ERR3
027244  104455                                                                          TRAP    C$ERDF
027246  000003                                                                          .WORD   3
027250  012220                                                                          .WORD   EM3
027252  015140                                                                          .WORD   ERR3
027254                                  ESCAPE  TST
027254  104410                                                                          TRAP    C$ESCAPE
027256  000030                                                                          .WORD   L10052-.
027260  020427  000377          27$:    CMP     R4,#255.            ;SEE IF WE READ LAST CHAR YET
```

```
027264  001004                          BNE     29$                     ;BR IF NOT
027266  004737  004662                  JSR     PC,OSIRDY               ;CHK ORDY=1, OCOR=0
027272  000001                          1
027274  000404                          BR      A3
027276  004737  004662          29$:    JSR     PC,OSIRDY               ;CHK ORDY=1, OCOR=1
027302  000003                          3
027304  000724                          BR      24$                     ;CONTINUE
027306                          A3:
027306                          ENDTST
027306                                                                          L10052:
027306  104401                                                                          TRAP    C$ETST





                                ;;*********************************************************************
                                .SBTTL          TEST 23 - TX MSG TIMING TEST, CHAR MODE, WITH CRC
                                ;*
                                ;* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
                                ;* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
                                ;* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
                                ;* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-16.
                                ;* THE FOLLOWING STEPS ARE DONE:
                                ;* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
                                ;* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
                                ;* THEN 2 TERMINATING SYNCHS ARE SENT.
                                ;* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
                                ;* CLEARED STATE AFTER THE 3RD SYNCH COMPLETES.
                                ;;*********************************************************************
027310                          BGNTST
027310                                                                  T23::
027310  012737  027422  002362          MOV     #A4,RETADR              ;SET TEST EXIT ADDRESS FOR ERRORS
027316  004737  005540                  JSR     PC,INITRN               ;DO MASTER CLR, LOAD 2 SOM'2
027322  000226                          SYNCH
027324  000011                          STRIP!DDCMP
027326  004737  006122                  JSR     PC,TXCHAR               ;LOAD A 000 CHAR, TX FIRST SYNCH (226)
027332  000000                          000
027334  100010                          CHPCHK!8.
027336  004737  006122                  JSR     PC,TXCHAR               ;LOAD 2ND 000 CHAR, TX 2ND SYNCH
027342  000000                          000
027344  000010                          8.
027346  004737  006122                  JSR     PC,TXCHAR               ;LOAD EOM CHAR, TX FIRST 000 CHAR
027352  001000                          TXEOM
027354  000010                          8.
027356  004737  006122                  JSR     PC,TXCHAR               ;LOAD EOM CHAR, TX 2ND 000 CHAR
027362  001000                          TXEOM
027364  000010                          8.
027366  004737  006122                  JSR     PC,TXCHAR               ;LOAD EOM, TX CRC-16 CHAR
027372  001000                          TXEOM
027374  000020                          16.
027376  004737  006122                  JSR     PC,TXCHAR               ;LOAD EOM, TX FIRST TERMINATING SYNCH
027402  001000                          TXEOM
027404  000010                          8.
027406  004737  006122                  JSR     PC,TXCHAR               ;LOAD EOM, TX 2ND TERMINATING SYNCH
027412  001000                          TXEOM
027414  000010                          8.
```

```
027416  004737  006274                  JSR     PC,ENDTRN       ;SET OC, MONITOR OCOR
027422                          A4:
027422  004737  003576                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
027426                          ENDTST
027426                                                          L10053:
027426  104401                                                          TRAP    C$ETST





                                ;;*************************************************************************
                                .SBTTL      TEST 24 - TX MSG TIMING TEST, BIT MODE, WITH CRC
                                ;*
                                ;* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
                                ;* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
                                ;* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
                                ;* BIT ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-CCITT-1.
                                ;* THE FOLLOWING STEPS ARE DONE:
                                ;* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN BIT MODE.
                                ;* SOM IS SET TWICE TO SEND 2 FLAG CHARS. THEN, 2 000 CHARS ARE SENT, AND
                                ;* THEN 2 TERMINATING FLAGS ARE SENT.
                                ;* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
                                ;* CLEARED STATE.
                                ;;*************************************************************************
027430                          BGNTST
027430                                                          T24::
027430  012737  027532  002362          MOV     #A5,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
027436  004737  005540                  JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'2
027442  000000                          000
027444  000000                          000
027446  004737  006122                  JSR     PC,TXCHAR       ;LOAD A 000 CHAR, TX FIRST FLAG
027452  000000                          000
027454  100010                          CHPCHK!8.
027456  004737  006122                  JSR     PC,TXCHAR       ;LOAD 2ND 000 CHAR, TX 2ND FLAG
027462  000000                          000
027464  000010                          8.
027466  004737  006122                  JSR     PC,TXCHAR       ;LOAD EOM CHAR, TX FIRST 000 CHAR
027472  001000                          TXEOM
027474  000010                          8.
027476  004737  006122                  JSR     PC,TXCHAR       ;LOAD EOM, TX 2ND 000 CHAR AND CRC-CCITT-1 CHAR
027502  001000                          TXEOM
027504  000030                          24.
027506  004737  006122                  JSR     PC,TXCHAR       ;LOAD EOM, TX FIRST TERMINATING FLAG
027512  001000                          TXEOM
027514  000010                          8.
027516  004737  006122                  JSR     PC,TXCHAR       ;LOAD EOM, TX 2ND TERMINATING FLAG
027522  001000                          TXEOM
027524  000010                          8.
027526  004737  006274                  JSR     PC,ENDTRN       ;SET OC, MONITOR OCOR
027532                          A5:
027532  004737  003576                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
027536                          ENDTST
027536                                                          L10054:
027536  104401                                                          TRAP    C$ETST
```

```
                                      ;;*********************************************************************
                                      .SBTTL      TEST 25 - TX MSG TIMING TEST, CHAR MODE, WITH NO CRC
                                      ;*
                                      ;* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
                                      ;* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
                                      ;* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
                                      ;* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND NO ERROR CHECKING.
                                      ;* THE FOLLOWING STEPS ARE DONE:
                                      ;* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
                                      ;* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
                                      ;* THEN 2 TERMINATING SYNCHS ARE SENT.
                                      ;* THE TEST IS PERFORMED WITH TXEN (REG14, BIT6) SET, AND THE PROGRAM CHECKS
                                      ;* THAT THIS HOLDS RTS HIGH PAST THE END OF THE MESSAGE.
                                      ;* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
                                      ;* CLEARED STATE.
                                      ;;*********************************************************************
027540                                BGNTST
027540                                                                        T25::
027540 012737 027726 002362                  MOV     #A6,RETADR       ;SET TEST EXIT ADDRESS FOR ERRORS
027546 004737 005540                         JSR     PC,INITRN        ;DO MASTER CLR, LOAD 2 SOM'2
027552 000226                                SYNCH
027554 000311                                CRC2!CRC1!STRIP!DDCMP
027556 012737 000014 002400                  MOV     #14,REGNUM       ;SET REG NO. = 14
027564 012737 000100 002366                  MOV     #TXEN,WRIBYT     ;SET TXEN BIT
027572 004737 003750                         JSR     PC,WRITLU        ;LOAD TXEN BIT IN REG 14
027576 004737 006122                         JSR     PC,TXCHAR        ;LOAD A 000 CHAR, TX FIRST SYNCH (226)
027602 000000                                000
027604 100010                                CHPCHK!8.
027606 004737 006122                         JSR     PC,TXCHAR        ;LOAD 2ND 000 CHAR, TX 2ND SYNCH
027612 000000                                000
027614 000010                                8.
027616 004737 006122                         JSR     PC,TXCHAR        ;LOAD EOM CHAR, TX FIRST 000 CHAR
027622 001000                                TXEOM
027624 000010                                8.
027626 004737 006122                         JSR     PC,TXCHAR        ;LOAD EOM CHAR, TX 2ND 000 CHAR
027632 001000                                TXEOM
027634 000010                                8.
027636 004737 006122                         JSR     PC,TXCHAR        ;LOAD EOM, TX FIRST TERMINATING SYNCH
027642 001000                                TXEOM
027644 000010                                8.
027646 004737 006122                         JSR     PC,TXCHAR        ;LOAD EOM, TX 2ND TERMINATING SYNCH
027652 001000                                TXEOM
027654 000010                                8.
027656 004737 005254                         JSR     PC,STPLU         ;CLK PAST END OF MSG
027662 000030                                24.
027664 012737 000013 002400                  MOV     #13,REGNUM       ;SET REG NO. = 13
027672 004737 003672                         JSR     PC,READLU        ;READ REG 13
027676 032737 000040 002364                  BIT     #RTS,REDBYT      ;CHK FOR RTS STILL SET
027704 001006                                BNE     3$               ;BR IF RTS SET
027706 004737 004526                         JSR     PC,GETALL        ;GET REGS FOR PRINTOUT
                                      ;REPORT RTS NOT SET
027712                                       ERRDF   60,EM60,ERR7
027712 104455                                                                 TRAP    C$ERDF
027714 000074                                                                 .WORD   60
```

```
027716 014230                                                                      .WORD   EM60
027720 017470                                                                      .WORD   ERR7
027722 004737 006274            3$:     JSR     PC,ENDTRN       ;SET OC, MONITOR OCOR
027726                          A6:
027726 004737 003576                    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
027732                          ENDTST
027732                                                                 L10055:
027732 104401                                                                      TRAP    C$ETST




                        ;;*****************************************************************************
                        .SBTTL      TEST 26 - TX UNDERRUN SET AND CLEAR TEST - CHAR MODE
                        ;*
                        ;* IN THIS TEST, A TX UNDERRUN ERROR IS FORCED IN EACH OF 2 SITUATIONS,
                        ;* AND THEN CLEARED DIFFERENTLY IN EACH.
                        ;* IN THE FIRST, A MESSAGE IS INITIATED, A 000 CHAR IS SENT, AND THE TX
                        ;* BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG,
                        ;* WHICH CAUSES UNRR TO SET IN REG 11. THEN, SOM IS SET TO CLEAR THE ERROR,
                        ;* AND THIS IS VERIFIED.
                        ;* IN THE SECOND SITUATION, A MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX
                        ;* BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH
                        ;* AGAIN CAUSES UNRR TO SET. THEN, A MASTER CLEAR IS DONE, AND THE UNRR BIT
                        ;* IS CHECKED TO BE CLEARED.
                        ;;*****************************************************************************
027734                  BGNTST
027734                                                          T26::
                        ;------------------------------------------------------------------------------
                        ; CAUSE TX UNDERRUN, CHK UNRR = 1; SET SOM, CHK UNRR = 0
                        ;------------------------------------------------------------------------------
027734 012737 030272 002362     MOV     #A7,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
027742 004737 005540            JSR     PC,INITRN       ;DO MASTER CLEAR, LOAD 2 SOM'S
027746 000226                   SYNCH
027750 000011                   STRIP!DDCMP
027752 004737 006122            JSR     PC,TXCHAR       ;LOAD A 000 CHAR, TX FIRST SYNCH
027756 000000                   000
027760 100010                   CHPCHK!8.
027762 004737 005254            JSR     PC,STPLU        ;CLOCK THE TRANSMITTER UNTIL 7 BITS OF
027766 000016                   14.                     ;     THE 000 CHAR HAVE BEEN TRANSMITTED
027770 012737 000011 002400     MOV     #11,REGNUM      ;SET LU REG NO. = 11
027776 004737 003672            JSR     PC,READLU       ;READ REG 11
030002 132737 000001 002364     BITB    #UNRR,REDBYT    ;CHK FOR UNRR = 0
030010 001410                   BEQ     6$              ;BR IF UNRR = 0
030012 004737 004526            JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                        ;REPORT UNRR NOT CLEARED
030016                          ERRDF   16,EM16,ERR7
030016 104455                                                                      TRAP    C$ERDF
030020 000020                                                                      .WORD   16
030022 012622                                                                      .WORD   EM16
030024 017470                                                                      .WORD   ERR7
030026 000137 030272            JMP     A7              ;SKIP TO END OF TEST
030032 004737 005254    6$:     JSR     PC,STPLU        ;CLOCK LAST BIT OF 000 CHAR
030036 000003                   3
030040 004737 003672            JSR     PC,READLU       ;READ REG 11
030044 132737 000001 002364     BITB    #UNRR,REDBYT    ;CHK FOR UNRR = 1
```

```
030052  001010                        BNE     9$              ;BR IF UNRR = 1
030054  004737  004526                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                              ;REPORT UNRR NOT SET
030060                                ERRDF   14,EM14,ERR7
030060  104455                                                                      TRAP    C$ERDF
030062  000016                                                                      .WORD   14
030064  012556                                                                      .WORD   EM14
030066  017470                                                                      .WORD   ERR7
030070  000137  030272                JMP     A7              ;SKIP TO END OF TEST
030074  012737  000400  002422  9$:   MOV     #TXSOM,TXWORD   ;SET SOM CHAR TO BE WRITTEN
030102  004737  005174                JSR     PC,LDTXSI       ;LOAD SOM CHAR INTO TX SILO
030106  004737  005146                JSR     PC,WAIT50       ;WAIT FOR SILO DATA TO RIPPLE
030112  004737  005254                JSR     PC,STPLU        ;CLOCK LU FOR 2 CYCLES
030116  000002                        2
030120  004737  003672                JSR     PC,READLU       ;READ REG 11
030124  132737  000001  002364        BITB    #UNRR,REDBYT    ;CHK FOR UNRR = 0
030132  001410                        BEQ     12$             ;BR IF UNRR = 0
030134  004737  004526                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                              ;REPORT UNRR NOT CLEARED BY SOM
030140                                ERRDF   13,EM13,ERR7
030140  104455                                                                      TRAP    C$ERDF
030142  000015                                                                      .WORD   13
030144  012526                                                                      .WORD   EM13
030146  017470                                                                      .WORD   ERR7
030150  000137  030272                JMP     A7              ;SKIP TO END OF TEST
030154                        12$:
                              ;------------------------------------------------------------------------------
                              ;CAUSE TX UNDERRUN, CHK UNRR = 1; DO MASTER CLR, CHK UNRR = 0
                              ;------------------------------------------------------------------------------
030154  004737  005540                JSR     PC,INITRN       ;DO MASTER CLEAR, LOAD 2 SOM'S
030160  000226                        SYNCH
030162  000051                        IDLE!STRIP!DDCMP
030164  004737  006122                JSR     PC,TXCHAR       ;LOAD A 000 CHAR, TX FIRST SYNCH
030170  000000                        000
030172  100010                        CHPCHK!8.
030174  004737  005254                JSR     PC,STPLU        ;STEP THE LU UNTIL 000 HAS BEN TRANSMITTED
030200  000021                        17.
030202  004737  003672                JSR     PC,READLU       ;READ REG 11
030206  132737  000001  002364        BITB    #UNRR,REDBYT    ;CHK FOR UNR = 1
030214  001010                        BNE     16$             ;BR IF UNRR = 1
030216  004737  004526                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                              ;REPORT UNRR NOT SET
030222                                ERRDF   14,EM14,ERR7
030222  104455                                                                      TRAP    C$ERDF
030224  000016                                                                      .WORD   14
030226  012556                                                                      .WORD   EM14
030230  017470                                                                      .WORD   ERR7
030232  000137  030272                JMP     A7              ;SKIP TO END OF TEST
030236  004737  003576        16$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
030242  004737  003672                JSR     PC,READLU       ;READ REG 11
030246  132737  000001  002364        BITB    #UNRR,REDBYT    ;CHK FOR UNRR = 0
030254  001406                        BEQ     A7              ;BR IF UNRR = 0
030256  004737  004526                JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                              ;REPORT UNRR NOT CLEARED BY OC
030262                                ERRDF   15,EM15,ERR7
030262  104455                                                                      TRAP    C$ERDF
030264  000017                                                                      .WORD   15
```

```
030266  012573                                                                 .WORD   EM15
030270  017470                                                                 .WORD   ERR7
030272  004737  003576                A7:     JSR     PC,MSTCLR       ;ISSUE CLEAN-UP MASTER CLEAR
030276                                ENDTST
030276                                                                 L10056:
030276  104401                                                                 TRAP    C$ETST





                                      ;;*******************************************************************
                                      .SBTTL      TEST 27 - TRANSMIT CHAR LENGTH TIMING TEST - CHAR MODE, CRC
                                      ;;*
                                      ;;* THE LINE UNIT IS PLACED IN CHAR MODE (DDCMP) AND A MESSAGE IS INITIATED
                                      ;;* WITH AN 8-BIT SYNCH AND A 5-BIT SYNCH CHAR. NEXT, A 000 CHAR IS SENT WITH
                                      ;;* EACH OF THE FOLLOWING TX CHAR LENGTHS : 5 BITS, 6 BITS, 7 BITS, 8 BITS.
                                      ;;* (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED). TWO
                                      ;;* TERMINATING SYNCHS ARE SENT AFTER THE DATA.
                                      ;;*******************************************************************
030300                                BGNTST
030300                                                         T27::
030300  012737  030506  002362                MOV     #A8,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
030306  004737  005540                        JSR     PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'S
030312  000000                                000
030314  000041                                IDLE!DDCMP
030316  012737  000006  002402                MOV     #6,AXNUM        ;SET BYTE NO. = 6 FOR AX3
030324  012737  000000  002374                MOV     #000,WAX15      ;SET DATA FOR AX3-15 = 0
030332  012737  000240  002376                MOV     #TXLEN2!TXLEN0,WAX16 ;SET TX LENGTH = 5 FOR AX3-16
030340  004737  004312                        JSR     PC,WRITAX       ;LOAD AX3-15,AX3-16
030344  004737  006122                        JSR     PC,TXCHAR       ;LOAD 5-BIT 000 CHAR, TX 8-BIT SYNCH
030350  000000                                000
030352  100010                                CHPCHK!8.
030354  004737  006122                        JSR     PC,TXCHAR       ;LOAD 6-BIT 000 CHAR, TX 5-BIT SYNCH
030360  000000                                000
030362  000005                                5
030364  012737  000300  002376                MOV     #TXLEN2!TXLEN1,WAX16
030372  004737  004312                        JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 6
030376  004737  006122                        JSR     PC,TXCHAR       ;LOAD 7-BIT 000 CHAR, TX 5-BIT 000 CHAR
030402  000000                                000
030404  000005                                5
030406  012737  000340  002376                MOV     #TXLEN2!TXLEN1!TXLEN0,WAX16
030414  004737  004312                        JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 7
030420  004737  006122                        JSR     PC,TXCHAR       ;LOAD 8-BIT 000 CHAR, TX 6-BIT 000 CHAR
030424  000000                                000
030426  000006                                6
030430  012737  000000  002376                MOV     #000,WAX16
030436  004737  004312                        JSR     PC,WRITAX       ;SET TX CHAR LENGTH = 8
030442  004737  006122                        JSR     PC,TXCHAR       ;LOAD EOM, TX 7-BIT 000 CHAR
030446  001000                                TXEOM
030450  000007                                7
030452  004737  006122                        JSR     PC,TXCHAR       ;LOAD EOM, TX 8-BIT 000 CHAR
030456  001000                                TXEOM
030460  000010                                8.
030462  004737  006122                        JSR     PC,TXCHAR       ;LOAD EOM, TX CRC-16 CHAR
030466  001000                                TXEOM
030470  000020                                16.
```

```
030472  004737  006122                   JSR     PC,TXCHAR        ;LOAD EOM, TX FIRST TERMINATING SYNCH
030476  001000                           TXEOM
030500  000010                           8.
030502  004737  006274                   JSR     PC,ENDTRN        ;CLEAR TRANSMITTER
030506                           A8:
030506                           ENDTST
030506                                                                          L10057:
030506  104401                                                                          TRAP    C$ETST





                                 ;;*******************************************************************************
                                 .SBTTL      TEST 28 - TRANSMIT CHAR LENGTH TIMING TEST - BIT MODE, CRC
                                 ;*
                                 ;* THE LINE UNIT IS PLACED IN BIT MODE AND A MESSAGE IS INITIATED
                                 ;* WITH 2 FLAG CHARS. NEXT, 2 8-BIT 000 CHARS ARE SENT, FOLLOWED BY 000 CHARS
                                 ;* WITH EACH OF THE FOLLOWING TRANSMITTER CHAR LENGTHS:
                                 ;*     1 BIT, 2 BITS, 3 BITS, 4 BITS, 5 BITS, 6 BITS, 7 BITS, AND 8 BITS.
                                 ;* (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED).
                                 ;* TWO TERMINATING FLAGS ARE SENT AFTER THE DATA.
                                 ;;*******************************************************************************
030510                           BGNTST
030510                                                                          T28::
030510  012737  031036  002362           MOV     #A9,RETADR       ;SET TEST EXIT ADDRESS FOR ERRORS
030516  004737  005540                   JSR     PC,INITRN        ;DO MASTER CLR, LOAD 2 SOM'S
030522  000000                           000
030524  000000                           000
030526  004737  006122                   JSR     PC,TXCHAR        ;LOAD FIRST 8-BIT 000 CHAR, TX FIRST FLAG
030532  000000                           000
030534  100010                           CHPCHK!8.
030536  004737  006122                   JSR     PC,TXCHAR        ;LOAD 2ND 8-BIT 000 CHAR, TX 2ND FLAG
030542  000000                           000
030544  000010                           8.
030546  004737  006122                   JSR     PC,TXCHAR        ;LOAD 1-BIT 000 CHAR, TX FIRST 8-BIT 000 CHAR
030552  000000                           000
030554  000010                           8.
030556  012737  000006  002402           MOV     #6,AXNUM         ;SET BYTE NO. = 6 FOR AX3
030564  012737  000000  002374           MOV     #000,WAX15       ;SET DATA FOR AX3-15 = 0
030572  012737  000040  002376           MOV     #TXLEN0,WAX16    ;SET TX CHAR LENGTH = 1 FOR AX3-16
030600  004737  004312                   JSR     PC,WRITAX        ;LOAD AX3-15,AX3-16
030604  004737  006122                   JSR     PC,TXCHAR        ;LOAD 2-BIT 000 CHAR, TX 2ND 8-BIT 000 CHAR
030610  000000                           000
030612  000010                           8.
030614  012737  000100  002376           MOV     #TXLEN1,WAX16
030622  004737  004312                   JSR     PC,WRITAX        ;SET TX CHAR LENGTH = 2
030626  004737  006122                   JSR     PC,TXCHAR        ;LOAD 3-BIT 000 CHAR, TX 1-BIT 000 CHAR
030632  000000                           000
030634  000001                           1
030636  012737  000140  002376           MOV     #TXLEN1!TXLEN0,WAX16
030644  004737  004312                   JSR     PC,WRITAX        ;SET TX CHAR LENGTH = 3
030650  004737  006122                   JSR     PC,TXCHAR        ;LOAD 4-BIT 000 CHAR, TX 2-BIT 000 CHAR
030654  000000                           000
030656  000002                           2
030660  012737  000200  002376           MOV     #TXLEN2,WAX16
030666  004737  004312                   JSR     PC,WRITAX        ;SET TX CHAR LENGTH = 4
```

```
030672  004737  006122                   JSR     PC,TXCHAR         ;LOAD 5-BIT 000 CHAR, TX 3-BIT 000 CHAR
030676  000000                           000
030700  000003                           3
030702  012737  000240  002376           MOV     #TXLEN2!TXLEN0,WAX16
030710  004737  004312                   JSR     PC,WRITAX         ;SET TX CHAR LENGTH = 5
030714  004737  006122                   JSR     PC,TXCHAR         ;LOAD 6-BIT 000 CHAR, TX 4-BIT 000 CHAR
030720  000000                           000
030722  000004                           4
030724  012737  000300  002376           MOV     #TXLEN2!TXLEN1,WAX16
030732  004737  004312                   JSR     PC,WRITAX         ;SET TX CHAR LENGTH = 6
030736  004737  006122                   JSR     PC,TXCHAR         ;LOAD 7-BIT 000 CHAR, TX 5-BIT 000 CHAR
030742  000000                           000
030744  000005                           5
030746  012737  000340  002376           MOV     #TXLEN2!TXLEN1!TXLEN0,WAX16
030754  004737  004312                   JSR     PC,WRITAX         ;SET TX CHAR LENGTH = 7
030760  004737  006122                   JSR     PC,TXCHAR         ;LOAD 8-BIT 000 CHAR, TX 6-BIT 000 CHAR
030764  000000                           000
030766  000006                           6
030770  012737  000000  002376           MOV     #000,WAX16
030776  004737  004312                   JSR     PC,WRITAX         ;SET TX CHAR LENGTH = 8
031002  004737  006122                   JSR     PC,TXCHAR         ;LOAD EOM, TX 7-BIT 000 CHAR
031006  001000                           TXEOM
031010  000007                           7
031012  004737  006122                   JSR     PC,TXCHAR         ;LOAD EOM, TX 8-BIT 000 CHAR, CRC-CCITT-1 CHAR
031016  001000                           TXEOM
031020  000031                           25.
031022  004737  006122                   JSR     PC,TXCHAR         ;LOAD EOM, TX FIRST TERMINATING FLAG
031026  001000                           TXEOM
031030  000010                           8.
031032  004737  006274                   JSR     PC,ENDTRN         ;CLEAR TRANSMITTER
031036                           A9:
031036                           ENDTST
031036                                                                         L10060:
031036  104401                                                                         TRAP    C$ETST




                                 ;;*************************************************************************
                                 .SBTTL       TEST 29 - TXDATA BIT TEST - CHAR MODE, CRC
                                 ;*
                                 ;* THE LINE UNIT IS INITIALIZED AND A MSG IS INITIATED (USING STEPLU) WITH CRC-
                                 ;* 16 SELECTED IN CHAR MODE. TWO SYNCHS, 000,125,252,377,000, AND 2 TERMINATING
                                 ;* SYNCHS ARE THEN SENT. THE PROGRAM CHECKS EACH BIT OF THE TRANSMITTED
                                 ;* DATA CHARS, BY MONITORING TXDATA (REG 17) AS THE DATA IS CLOCKED OUT OF
                                 ;* THE USYRT TRANSMITTER.
                                 ;;*************************************************************************
031040                           BGNTST
031040                                                                         T29::
031040  004737  005540                   JSR     PC,INITRN         ;DO MASTER CLR, LOAD 2 SOM'S
031044  000226                           SYNCH
031046  000011                           STRIP!DDCMP
031050  012701  003224                   MOV     #MSG1+4,R1        ;GET POINTER TO DATA
031054  010103                           MOV     R1,R3
031056  012137  002422           3$:     MOV     (R1)+,TXWORD
031062  004737  005174                   JSR     PC,LDTXSI         ;LOAD A DATA CHAR INTO TX SILO
```

```
031066  020127  003242                  CMP     R1,#MSG1+18.    ;SEE IF ALL CHARS LOADED YET
031072  103771                          BLO     3$              ;BR IF NOT YET
031074  004737  005146                  JSR     PC,WAIT50       ;WAIT FOR SILO TO RIPPLE
031100  004737  005254                  JSR     PC,STPLU        ;CLOCK LU UNTIL SYNCHS ARE TX'D
031104  100020                          CHPCHK!16.
031106  011337  031116          6$:     MOV     (R3),8$         ;GET EXPECTED DATA CHAR
031112  004737  011176                  JSR     PC,CKTBIT       ;CHECK TXDATA FOR CHAR BITS
031116  000000                  8$:     .WORD   0               ;EXPECTED CHAR GOES HERE
031120  062703  000002                  ADD     #2,R3           ;INCR PATTERN POINTER
031124  020327  003236                  CMP     R3,#MSG1+14.    ;SEE IF ALL CHARS CHECKED YET
031130  103766                          BLO     6$              ;BR IF NOT YET
031132  004737  003576          15$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
031136                          ENDTST
031136                                                                  L10061:
031136  104401                                                                  TRAP    C$ETST

                                ;;*******************************************************************
                                .SBTTL          TEST 30 - USYRT RECEIVER MSG TEST - CHAR MODE, CRC
                                ;*
                                ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
                                ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-16
                                ;* SELECTED. TWO SYNCHS, 000,125,252,377,000, AND FOUR TERMINATING SYNCHS ARE
                                ;* SENT. THE PROGRAM MONITORS IACT, AND THE RCV'D CHARS AND CRC BYTES ARE READ
                                ;* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
                                ;* STILL SET AFTER THE MESSAGE.
                                ;;*******************************************************************
031140                          BGNTST
031140                                                                  T30::
031140  012737  031502  002362          MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
031146  004737  003576                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
031152  004737  010640                  JSR     PC,SETUP        ;PROGRAM THE USYRT
031156  000226                          SYNCH
031160  000013                          STRIP!IERR!DDCMP
031162  000000                          000
031164  000000                          000
031166  012737  000012  002400          MOV     #12,REGNUM      ;SET LU REG NO. = 12
031174  005037  002402                  CLR     AXNUM           ;SET AX BYTE NO. = 0 FOR AX0
031200  112737  000040  002366          MOVB    #LULP,WRIBYT
031206  004737  003750                  JSR     PC,WRITLU       ;SET LULP IN REG 12
031212  012701  003220                  MOV     #MSG1,R1        ;GET POINTER TO MSG DATA TABLE
031216  012137  002422          3$:     MOV     (R1)+,TXWORD    ;GET CHAR TO BE LOADED
031222  004737  005174                  JSR     PC,LDTXSI       ;LOAD CHAR INTO TX SILO
031226  020127  003246                  CMP     R1,#MSG1+22.    ;SEE IF ALL MSG CHARS LOADED YET
031232  103771                          BLO     3$              ;BR IF NOT YET
031234  004737  005146                  JSR     PC,WAIT50       ;ALLOW DATA TO RIPPLE IN SILO
031240  004737  005254                  JSR     PC,STPLU        ;CLOCK LU FOR 40 CYCLES (UNTIL FIRST
031244  000050                          40.                     ;   DATA CHAR IS ABOUT TO BE RECEIVED)
031246  004737  006714                  JSR     PC,IACTIV       ;CHK IACT = 0
031252  000000                          0
031254  004737  005254                  JSR     PC,STPLU        ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
031260  000006                          6
031262  012701  003224                  MOV     #MSG1+4,R1
031266  004737  006714          10$:    JSR     PC,IACTIV       ;CHK IACT = 1
```

```
031272  000001                                        1
031274  004737  004124                                JSR     PC,READAX          ;READ AXO
031300  023721  002370                                CMP     RAX15,(R1)+        ;COMPARE RCV'D CHAR TO EXPECTED
031304  001415                                        BEQ     12$                ;BR IF RCV'D DATA OK
031306  016137  177776  002404                        MOV     -2(R1),GOODAT      ;GET EXPECTED DATA
031314  013737  002370  002406                        MOV     RAX15,BADDAT       ;GET ACTUAL DATA
031322  004737  004526                                JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                              ;REPORT INCORRECT DATA CHAR RCV'D
031326                                                ERRDF   26,EM26,ERR3
031326  104455                                                                              TRAP    C$ERDF
031330  000032                                                                              .WORD   26
031332  013067                                                                              .WORD   EM26
031334  015140                                                                              .WORD   ERR3
031336  000461                                        BR      24$
031340  004737  005254                        12$:    JSR     PC,STPLU           ;CLOCK LU 8 CYCLES
031344  000010                                        8.
031346  020127  003236                                CMP     R1,#MSG1+14.       ;SEE IF CHECKING HI CRC BYTE YET
031352  103745                                        BLO     10$                ;BR IF NOT YET
031354  004737  006714                                JSR     PC,IACTIV          ;CHK IACT = 1
031360  000001                                        1
031362  004737  004124                                JSR     PC,READAX          ;READ AXO
031366  123727  002370  000160                        CMPB    RAX15,#160         ;CMP RCV'D CHAR TO EXPECTED HI CRC BYTE
031374  001415                                        BEQ     16$                ;BR IF HI CRC BYTE RCV'D OK
031376  012737  000160  002404                        MOV     #160,GOODAT        ;GET EXPECTED DATA
031404  013737  002370  002406                14$:    MOV     RAX15,BADDAT       ;SET ACTUAL DATA
031412  004737  004526                                JSR     PC,GETALL          ;GET REGS FOR PRINTOUT
                                              ;REPORT INCORRECT CRC BYTE RCV'D
031416                                                ERRDF   27,EM27,ERR3
031416  104455                                                                              TRAP    C$ERDF
031420  000033                                                                              .WORD   27
031422  013121                                                                              .WORD   EM27
031424  015140                                                                              .WORD   ERR3
031426  000425                                        BR      24$
031430  004737  005254                        16$:    JSR     PC,STPLU           ;CLOCK LU FOR 8 CYCLES
031434  000010                                        8.
031436  004737  006714                                JSR     PC,IACTIV          ;CHK IACT = 1
031442  000001                                        1
031444  012737  000034  002404                        MOV     #034,GOODAT        ;GET EXPECTED LO CRC BYTE
031452  004737  004124                                JSR     PC,READAX          ;READ AXO
031456  123727  002370  000034                        CMPB    RAX15,#034         ;CMP RCV'D CHAR TO EXPECTED LO CRC BYTE
031464  001347                                        BNE     14$                ;BR IF LO CRC INCORRECT
031466  004737  005254                                JSR     PC,STPLU           ;CLOCK LU 8 CYCLES
031472  000010                                        8.
031474  004737  006714                                JSR     PC,IACTIV          ;CHK IACT STILL = 1
031500  000001                                        1
031502  004737  003576                        24$:    JSR     PC,MSTCLR          ;ISSUE CLEAN-UP MASTER CLEAR
031506                                        ENDTST
031506                                                                              L10062:
031506  104401                                                                              TRAP    C$ETST





                                              ;;*******************************************************************************
                                              .SBTTL        TEST 31 - USYRT RECEIVER MSG TEST - BIT MODE, CRC
                                              ;*
```

```
                                        ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
                                        ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-
                                        ;* CCITT-1. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS ARE THEN
                                        ;* SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE READ
                                        ;* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
                                        ;* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
                                        ;;*******************************************************************************
031510                                  BGNTST
031510                                                                                  T31::
031510  012737  032044  002362                  MOV     #24$,RETADR     ;SET TEST EXIT ADDRESS FOR ERRORS
031516  004737  003576                          JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
031522  004737  010640                          JSR     PC,SETUP        ;PROGRAM THE USYRT
031526  000000                                  000
031530  000002                                  IERR
031532  000000                                  000
031534  000000                                  000
031536  012737  000012  002400                  MOV     #12,REGNUM      ;SET LU REG NO. = 12
031544  005037  002402                          CLR     AXNUM           ;SET AX BYTE NO. = 0 FOR AX0
031550  112737  000040  002366                  MOVB    #LULP,WRIBYT
031556  004737  003750                          JSR     PC,WRITLU       ;SET LULP IN REG 12
031562  012701  003220                          MOV     #MSG1,R1        ;GET POINTER TO MSG DATA TABLE
031566  012137  002422                  3$:     MOV     (R1)+,TXWORD    ;GET CHAR TO BE LOADED
031572  004737  005174                          JSR     PC,LDTXSI       ;LOAD CHAR INTO TX SILO
031576  020127  003242                          CMP     R1,#MSG1+18.    ;SEE IF ALL MSG CHARS LOADED YET
031602  103771                                  BLO     3$              ;BR IF NOT YET
031604  004737  005146                          JSR     PC,WAIT50       ;ALLOW DATA TO RIPPLE IN SILO
031610  004737  005254                          JSR     PC,STPLU        ;CLOCK LU FOR 50 CYCLES (UNTIL FIRST
031614  000062                                  50.                     ;    DATA CHAR IS ABOUT TO BE RECEIVED)
031616  004737  006714                          JSR     PC,IACTIV       ;CHK IACT = 0
031622  000000                                  0
031624  004737  007102                          JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
031630  000000                                  0
031632  004737  005254                          JSR     PC,STPLU        ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
031636  000006                                  6
031640  012701  003224                          MOV     #MSG1+4,R1
031644  020127  003224                  5$:     CMP     R1,#MSG1+4      ;SEE IF 1ST CHAR RCV'D
031650  001007                                  BNE     6$              ;BR IF NO
031652  004737  006714                          JSR     PC,IACTIV       ;CHK IACT = 1
031656  000001                                  1
031660  004737  007102                          JSR     PC,RSEOM        ;CHK RSOM = 1, REOM = 0
031664  000001                                  1
031666  000420                                  BR      9$
031670  020127  003234                  6$:     CMP     R1,#MSG1+12.    ;SEE IF LAST CHAR RCV'D
031674  001007                                  BNE     8$              ;BR IF NO
031676  004737  006714                          JSR     PC,IACTIV       ;CHK FOR IACT = 0
031702  000000                                  0
031704  004737  007102                          JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
031710  000000                                  0
031712  000406                                  BR      9$
031714  004737  006714                  8$:     JSR     PC,IACTIV       ;CHK FOR IACT = 1
031720  000001                                  1
031722  004737  007102                          JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
031726  000000                                  0
031730  004737  004124                  9$:     JSR     PC,READAX       ;READ AX0
031734  023721  002370                          CMP     RAX15,(R1)+     ;COMPARE RCV'D CHAR TO EXPECTED
031740  001415                                  BEQ     12$             ;BR IF RCV'D DATA OK
031742  016137  177776  002404                  MOV     -2(R1),GOODAT   ;GET EXPECTED DATA
```

```
031750  013737  002370  002406            MOV     RAX15,BADDAT      ;GET ACTUAL DATA
031756  004737  004526                    JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                  ;REPORT INCORRECT DATA CHAR RCV'D
031762                                    ERRDF   26,EM26,ERR3
031762  104455                                                                          TRAP    C$ERDF
031764  000032                                                                          .WORD   26
031766  013067                                                                          .WORD   EM26
031770  015140                                                                          .WORD   ERR3
031772  000424                            BR      24$
031774  004737  005254            12$:    JSR     PC,STPLU          ;CLOCK LU 8 CYCLES
032000  000010                            8.
032002  020127  003236                    CMP     R1,#MSG1+14.      ;SEE IF ALL DATA CHARS CHECKED YET
032006  103716                            BLO     5$                ;BR IF NOT YET
032010  004737  006714                    JSR     PC,IACTIV         ;CHK IACT = 0
032014  000000                            0
032016  004737  007102                    JSR     PC,RSEOM          ;CHK RSOM = 0, REOM = 0
032022  000000                            0
032024  012737  000200  002366            MOV     #IC,WRIBYT
032032  004737  003750                    JSR     PC,WRITLU         ;SET IC (INPUT CLEAR) IN REG 12
032036  004737  006714                    JSR     PC,IACTIV         ;CHK IACT = 0
032042  000000                            0
032044  004737  003576            24$:    JSR     PC,MSTCLR         ;ISSUE CLEAN-UP MASTER CLEAR
032050                            ENDTST
032050                                                                          L10063:
032050  104401                                                                          TRAP    C$ETST






                                  ;;*******************************************************************************
                                  .SBTTL      TEST 32 - USYRT RECEIVER MSG TEST - CHAR MODE, NO CRC
                                  ;*
                                  ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
                                  ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH NO
                                  ;* ERROR DETECTION. TWO SYNCHS, 000,125,252,377,000, AND TWO SYNCHS ARE
                                  ;* THEN SENT. THE PROGRAM MONITORS IACT, AND THE RECEIVED CHARS ARE READ FROM
                                  ;* AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
                                  ;* STILL = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT = 0.
                                  ;;*******************************************************************************
032052                            BGNTST
032052                                                                  T32::
032052  012737  032314  002362            MOV     #24$,RETADR       ;SET TEST EXIT ADDRESS FOR ERRORS
032060  004737  003576                    JSR     PC,MSTCLR         ;ISSUE MASTER CLEAR
032064  004737  010640                    JSR     PC,SETUP          ;PROGRAM THE USYRT
032070  000226                            SYNCH
032072  000313                            CRC2!CRC1!STRIP!IERR!DDCMP
032074  000000                            000
032076  000000                            000
032100  012737  000012  002400            MOV     #12,REGNUM        ;SET LU REG NO. = 12
032106  005037  002402                    CLR     AXNUM             ;SET AX BYTE NO. = 0 FOR AX0
032112  112737  000040  002366            MOVB    #LULP,WRIBYT
032120  004737  003750                    JSR     PC,WRITLU         ;SET LULP IN REG 12
032124  012701  003220                    MOV     #MSG1,R1          ;GET POINTER TO MSG DATA TABLE
032130  012137  002422            3$:     MOV     (R1)+,TXWORD      ;GET CHAR TO BE LOADED
032134  004737  005174                    JSR     PC,LDTXSI         ;LOAD CHAR INTO TX SILO
032140  020127  003242                    CMP     R1,#MSG1+18.      ;SEE IF ALL MSG CHARS LOADED YET
```

```
032144  103771                          BLO     3$              ;BR IF NOT YET
032146  004737  005146                  JSR     PC,WAIT50       ;ALLOW DATA TO RIPPLE IN SILO
032152  004737  005254                  JSR     PC,STPLU        ;CLOCK LU FOR 24 CYCLES (UNTIL FIRST
032156  000030                          24.                     ;   DATA CHAR IS ABOUT TO BE RECEIVED)
032160  004737  006714                  JSR     PC,IACTIV       ;CHK IACT = 0
032164  000000                          0
032166  004737  005254                  JSR     PC,STPLU        ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
032172  000006                          6
032174  012701  003224                  MOV     #MSG1+4,R1
032200  004737  006714          10$:    JSR     PC,IACTIV       ;CHK IACT = 1
032204  000001                          1
032206  004737  004124                  JSR     PC,READAX       ;READ AX0
032212  023721  002370                  CMP     RAX15,(R1)+     ;COMPARE RCV'D CHAR TO EXPECTED
032216  001415                          BEQ     12$             ;BR IF RCV'D DATA OK
032220  016137  177776  002404          MOV     -2(R1),GOODAT   ;GET EXPECTED DATA
032226  013737  002370  002406          MOV     RAX15,BADDAT    ;GET ACTUAL DATA
032234  004737  004526                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT INCORRECT DATA CHAR RCV'D
032240                                  ERRDF   26,EM26,ERR3
032240  104455                                                                  TRAP    C$ERDF
032242  000032                                                                  .WORD   26
032244  013067                                                                  .WORD   EM26
032246  015140                                                                  .WORD   ERR3
032250  000421                          BR      24$
032252  004737  005254          12$:    JSR     PC,STPLU        ;CLOCK LU 8 CYCLES
032256  000010                          8.
032260  020127  003236                  CMP     R1,#MSG1+14.    ;SEE IF ALL 5 DATA CHARS RCV'D YET
032264  103745                          BLO     10$             ;BR IF NOT YET
032266  004737  006714                  JSR     PC,IACTIV       ;CHK FOR IACT STILL = 1
032272  000001                          1
032274  012737  000200  002366          MOV     #IC,WRIBYT
032302  004737  003750                  JSR     PC,WRITLU       ;SET IC (INPUT CLEAR) IN REG 12
032306  004737  006714                  JSR     PC,IACTIV       ;CHK IACT = 0
032312  000000                          0
032314  004737  003576          24$:    JSR     PC,MSTCLR       ;ISSUE CLEAN-UP MASTER CLEAR
032320                          ENDTST
032320                                                                  L10064:
032320  104401                                                                  TRAP    C$ETST

                                ;;*****************************************************************************
                                .SBTTL          TEST 33 - USYRT RECEIVER MSG TEST - BIT MODE, NO CRC
                                ;*
                                ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
                                ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH ERROR
                                ;* DETECTION INHIBITED. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS
                                ;* ARE THEN SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE
                                ;* READ FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
                                ;* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
                                ;;*****************************************************************************
032322                          BGNTST
032322                                                                  T33::
032322  012737  032656  002362          MOV     #24$,RETADR
032330  004737  003576                  JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
```

```
032334  004737  010640                   JSR     PC,SETUP          ;PROGRAM THE USYRT
032340  000000                           000
032342  000302                           CRC2!CRC1!IERR
032344  000000                           000
032346  000000                           000
032350  012737  000012  002400           MOV     #12,REGNUM        ;SET LU REG NO. = 12
032356  005037  002402                   CLR     AXNUM             ;SET AX BYTE NO. = 0 FOR AX0
032362  112737  000040  002366           MOVB    #LULP,WRIBYT
032370  004737  003750                   JSR     PC,WRITLU         ;SET LULP IN REG 12
032374  012701  003220                   MOV     #MSG1,R1          ;GET POINTER TO MSG DATA TABLE
032400  012137  002422          3$:      MOV     (R1)+,TXWORD      ;GET CHAR TO BE LOADED
032404  004737  005174                   JSR     PC,LDTXSI         ;LOAD CHAR INTO TX SILO
032410  020127  003242                   CMP     R1,#MSG1+18.      ;SEE IF ALL MSG CHARS LOADED YET
032414  103771                           BLO     3$                ;BR IF NOT YET
032416  004737  005146                   JSR     PC,WAIT50         ;ALLOW DATA TO RIPPLE IN SILO
032422  004737  005254                   JSR     PC,STPLU          ;CLOCK LU FOR 33 CYCLES (UNTIL FIRST
032426  000041                           33.                       ;    DATA CHAR IS ABOUT TO BE RECEIVED)
032430  004737  006714                   JSR     PC,IACTIV         ;CHK IACT = 0
032434  000000                           0
032436  004737  007102                   JSR     PC,RSEOM          ;CHK RSOM = 0, REOM = 0
032442  000000                           0
032444  004737  005254                   JSR     PC,STPLU          ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
032450  000006                           6
032452  012701  003224                   MOV     #MSG1+4,R1
032456  020127  003224          5$:      CMP     R1,#MSG1+4        ;SEE IF 1ST CHAR RCV'D
032462  001007                           BNE     6$                ;BR IF NO
032464  004737  006714                   JSR     PC,IACTIV         ;CHK IACT = 1
032470  000001                           1
032472  004737  007102                   JSR     PC,RSEOM          ;CHK RSOM = 1, REOM = 0
032476  000001                           1
032500  000420                           BR      9$
032502  020127  003234          6$:      CMP     R1,#MSG1+12.      ;SEE IF LAST CHAR RCV'D
032506  001007                           BNE     8$                ;BR IF NO
032510  004737  006714                   JSR     PC,IACTIV         ;CHK FOR IACT = 0
032514  000000                           0
032516  004737  007102                   JSR     PC,RSEOM          ;CHK RSOM = 0, REOM = 0
032522  000000                           0
032524  000406                           BR      9$
032526  004737  006714          8$:      JSR     PC,IACTIV         ;CHK FOR IACT = 1
032532  000001                           1
032534  004737  007102                   JSR     PC,RSEOM          ;CHK RSOM = 0, REOM = 0
032540  000000                           0
032542  004737  004124          9$:      JSR     PC,READAX         ;READ AX0
032546  023721  002370                   CMP     RAX15,(R1)+       ;COMPARE RCV'D CHAR TO EXPECTED
032552  001415                           BEQ     12$               ;BR IF RCV'D DATA OK
032554  016137  177776  002404           MOV     -2(R1),GOODAT     ;GET EXPECTED DATA
032562  013737  002370  002406           MOV     RAX15,BADDAT      ;GET ACTUAL DATA
032570  004737  004526                   JSR     PC,GETALL         ;GET REGS FOR PRINTOUT
                                ;REPORT INCORRECT DATA CHAR RCV'D
032574                                   ERRDF   26,EM26,ERR3
032574  104455                                                                     TRAP    C$ERDF
032576  000032                                                                     .WORD   26
032600  013067                                                                     .WORD   EM26
032602  015140                                                                     .WORD   ERR3
032604  000424                           BR      24$
032606  004737  005254          12$:     JSR     PC,STPLU          ;CLOCK LU 8 CYCLES
032612  000010                           8.
```

```
032614  020127  003236                  CMP     R1,#MSG1+14.     ;SEE IF ALL DATA CHARS CHECKED YET
032620  103716                          BLO     5$               ;BR IF NOT YET
032622  004737  006714                  JSR     PC,IACTIV        ;CHK IACT = 0
032626  000000                          0
032630  004737  007102                  JSR     PC,RSEOM         ;CHK RSOM = 0, REOM = 0
032634  000000                          0
032636  012737  000200  002366          MOV     #IC,WRIBYT
032644  004737  003750                  JSR     PC,WRITLU        ;SET IC (INPUT CLEAR) IN REG 12
032650  004737  006714                  JSR     PC,IACTIV        ;CHK IACT = 0
032654  000000                          0
032656  004737  003576          24$:    JSR     PC,MSTCLR        ;ISSUE CLEAN-UP MASTER CLEAR
032662                          ENDTST
032662                                                                   L10065:
032662  104401                                                                   TRAP    C$ETST




                                ;;*************************************************************************
                                .SBTTL      TEST 34 - SILO-DISABLED TRANSMITTER LOAD TEST
                                ;*
                                ;* THIS TEST DISABLES THE SILOS, LOADS A 125 CHARACTER INTO THE TX SILO, AND
                                ;* READS AX1-15 AND CHECKS THAT THE DATA DID NOT GET LOADED INTO THE USYRT TX
                                ;* BUFFER.
                                ;;*************************************************************************
032664                          BGNTST
032664                                                                   T34::
032664  004737  003576                  JSR     PC,MSTCLR        ;ISSUE MASTER CLEAR
032670  012737  000014  002400          MOV     #14,REGNUM       ;SET REG NO. = 14
032676  012737  000040  002366          MOV     #DISSI,WRIBYT    ;SET DISSI BIT
032704  004737  003750                  JSR     PC,WRITLU
032710  012737  000040  002426          MOV     #DISSI,DISILO    ;SET DISABLE SILO FLAG
032716  012737  000125  002422          MOV     #125,TXWORD      ;LOAD 125 INTO TX SILO
032724  004737  005174                  JSR     PC,LDTXSI
032730  012737  000002  002402          MOV     #2,AXNUM         ;SET REG NO. FOR AX1
032736  004737  004124                  JSR     PC,READAX        ;READ AX1-15, AX1-16
032742  123727  002370  000000          CMPB    RAX15,#000       ;CHECK FOR AX1-15 UNCHANGED
032750  001414                          BEQ     3$               ;BR IF UNCHANGED
032752  012737  000000  002404          MOV     #000,GOODAT      ;GET EXPECTED DATA
032760  013737  002370  002406          MOV     RAX15,BADDAT     ;GET ACTUAL DATA
032766  004737  004526                  JSR     PC,GETALL        ;GET REGS FOR PRINTOUT
                                ;REPORT REG MISCOMPARE
032772                                  ERRDF   3,EM3,ERR3
032772  104455                                                                   TRAP    C$ERDF
032774  000003                                                                   .WORD   3
032776  012220                                                                   .WORD   EM3
033000  015140                                                                   .WORD   ERR3
033002  005037  002426          3$:     CLR     DISILO           ;CLEAR DISABLE SILO FLAG
033006  004737  003576                  JSR     PC,MSTCLR        ;ISSUE MASTER CLEAR TO CLEAN UP
033012                          ENDTST
033012                                                                   L10066:
033012  104401                                                                   TRAP    C$ETST
```

```

                                     ;;*********************************************************************
                                     .SBTTL     TEST 35 - SILO-DISABLED MESSAGE TEST - BIT MODE, NO CRC
                                     ;*
                                     ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU)
                                     ;* WITH LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, WITH SILO
                                     ;* DISABLE SET, AND WITH NO ERROR DETECTION. TWO FLAGS, 000,125,252, AND
                                     ;* TERMINATING FLAGS ARE THEN SENT BY LOADING THE TRANSMITTED CHARS INTO
                                     ;* REG AX1. THE PROGRAM MONITORS OACT, IACT, RSOM, REOM, ORDY, OCOR, ICIR,
                                     ;* IRDY, AND THE RECEIVED CHARS ARE READ FROM AXO AND COMPARED TO EXPECTED
                                     ;* VALUES.
                                     ;;*********************************************************************
033014                               BGNTST
033014                                                                          T35::
033014  012737  033714  002362               MOV     #18$,RETADR       ;SET TEST EXIT ADDRESS FOR ERRORS
033022  004737  005540                       JSR     PC,INITRN         ;FIND OUT WHICH USYRT CHIP
033026  000000                               0
033030  000000                               0
033032  004737  003576                       JSR     PC,MSTCLR         ;ISSUE MASTER CLEAR
033036  004737  010640                       JSR     PC,SETUP          ;PROGRAM THE USYRT
033042  000000                               000
033044  000302                               CRC2!CRC1!IERR
033046  000000                               000
033050  000000                               000
033052  012737  000014  002400               MOV     #14,REGNUM        ;SET REG NO. = 14
033060  012737  000140  002366               MOV     #TXEN!DISSI,WRIBYT
033066  004737  003750                       JSR     PC,WRITLU         ;SET TXEN AND DISSI IN REG 14
033072  012737  000140  002426               MOV     #TXEN!DISSI,DISILO      ;SET DISABLE SILO FLAG
033100  012737  000012  002400               MOV     #12,REGNUM        ;SET LU REG NO. = 12
033106  112737  000040  002366               MOVB    #LULP,WRIBYT
033114  004737  003750                       JSR     PC,WRITLU         ;SET LULP IN REG 12
033120  012701  003220                       MOV     #MSG1,R1          ;GET POINTER TO MSG
033124  004737  004662                       JSR     PC,OSIRDY         ;CHK ORDY = 1
033130  000001                               1
033132  012737  000002  002402               MOV     #2,AXNUM          ;SET AX BYTE NO. FOR AX1
033140  112137  002374                       MOVB    (R1)+,WAX15       ;GET A CHAR
033144  112137  002376                       MOVB    (R1)+,WAX16
033150  004737  004312                       JSR     PC,WRITAX         ;LOAD CHAR INTO USYRT TX BUFFER
033154  004737  004662                       JSR     PC,OSIRDY         ;CHK ORDY = 0
033160  000000                               0
033162  004737  005352                       JSR     PC,OACTIV         ;CHK OACT = 0
033166  000000                               0
033170  004737  005254                       JSR     PC,STPLU          ;CLOCK LU FOR 3 CYCLES
033174  000003                               3
033176  004737  005352                       JSR     PC,OACTIV         ;CHK OACT = 1
033202  000001                               1
033204  012703  000004                       MOV     #4,R3             ;INIT COUNTER
033210  112137  002374               4$:     MOVB    (R1)+,WAX15       ;GET ANOTHER CHAR
033214  112137  002376                       MOVB    (R1)+,WAX16
033220  020327  000001                       CMP     R3,#1             ;SEE IF LOADING LAST DATA CHAR YET
033224  001006                               BNE     5$                ;BR IF NOT
033226  005737  002430                       TST     CHPTYP            ;SEE IF SIG USYRT
033232  001403                               BEQ     5$                ;BR IF YES
033234  112737  000002  002376               MOVB    #TEOM,WAX16       ;SET TEOM WITH LAST DATA CHAR
033242  004737  004662               5$:     JSR     PC,OSIRDY         ;CHK ORDY = 1
033246  000001                               1
033250  004737  004312                       JSR     PC,WRITAX         ;LOAD ANOTHER CHAR INTO USYRT TX BUFFER
```

```
033254  004737  004662                  JSR     PC,OSIRDY       ;CHK ORDY = 0
033260  000000                          0
033262  004737  006714                  JSR     PC,IACTIV       ;CHK IACT = 0
033266  000000                          0
033270  004737  007102                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
033274  000000                          0
033276  004737  005254                  JSR     PC,STPLU        ;CLOCK LU FOR 8 CYCLES
033302  000010                          8.
033304  004737  005352                  JSR     PC,OACTIV       ;CHK OACT = 1
033310  000001                          1
033312  004737  004662                  JSR     PC,OSIRDY       ;CHK ORDY = 1
033316  000001                          1
033320  005303                          DEC     R3              ;DECR COUNTER
033322  001332                          BNE     4$              ;BR IF NOT DONE YET
033324  004737  006430                  JSR     PC,ISIRDY       ;CHK IRDY = 0
033330  000000                          0
033332  005737  002430                  TST     CHPTYP          ;SEE IF SIG USYRT
033336  001007                          BNE     11$             ;BR IF NOT
033340  105037  002374                  CLRB    WAX15
033344  112737  000002  002376          MOVB    #TEOM,WAX16     ;LOAD EOM CHAR
033352  004737  004312                  JSR     PC,WRITAX       ;LOAD ANOTHER CHAR INTO USYRT TX BUFFER
033356  004737  005254          11$:    JSR     PC,STPLU        ;CLOCK LU FOR 3 CYCLES
033362  000003                          3
033364  004737  006430                  JSR     PC,ISIRDY       ;CHK IRDY = 1
033370  000002                          2
033372  004737  005352                  JSR     PC,OACTIV       ;CHK OACT = 1
033376  000001                          1
033400  004737  006714                  JSR     PC,IACTIV       ;CHK IACT = 1
033404  000001                          1
033406  004737  007102                  JSR     PC,RSEOM        ;CHK RSOM = 1, REOM = 0
033412  000001                          1
033414  004737  006430                  JSR     PC,ISIRDY       ;CHK IRDY = 0
033420  000000                          0
033422  012737  000000  002402          MOV     #0,AXNUM        ;SET AX BYTE NO. FOR AX0
033430  123727  002370  000000          CMPB    RAX15,#000      ;COMPARE RCV'D CHAR TO 000
033436  001415                          BEQ     9$              ;BR IF MATCH
033440  012737  000000  002404          MOV     #0,GOODAT       ;SET EXPECTED DATA
033446  013737  002370  002406  6$:     MOV     RAX15,BADDAT    ;SET ACTUAL DATA
033454  004737  004526                  JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
                                ;REPORT INCORRECT DATA CHAR RCV'D
033460                                  ERRDF   26,EM26,ERR3
033460  104455                                                                  TRAP    C$ERDF
033462  000032                                                                  .WORD   26
033464  013067                                                                  .WORD   EM26
033466  015140                                                                  .WORD   ERR3
033470  000511                          BR      18$
033472  004737  005254          9$:     JSR     PC,STPLU        ;CLOCK LU FOR 8 CYCLES
033476  000010                          8.
033500  004737  006430                  JSR     PC,ISIRDY       ;CHK IRDY = 1
033504  000002                          2
033506  004737  005352                  JSR     PC,OACTIV       ;CHK OACT = 1
033512  000001                          1
033514  004737  006714                  JSR     PC,IACTIV       ;CHK IACT = 1
033520  000001                          1
033522  004737  007102                  JSR     PC,RSEOM        ;CHK RSOM = 0, REOM = 0
033526  000000                          0
033530  004737  006430                  JSR     PC,ISIRDY       ;CHK IRDY = 0
```

```
033534  000000                           0
033536  123727  002370  000125           CMPB    RAX15,#125       ;COMPARE 2ND RCV'D CHAR TO 125
033544  001404                           BEQ     12$              ;BR IF MATCH
033546  012737  000125  002404           MOV     #125,GOODAT      ;SET EXPECTED DATA
033554  000734                           BR      6$               ;BR TO REPORT ERROR
033556  004737  005254          12$:     JSR     PC,STPLU         ;CLOCK LU FOR 8 CYCLES
033562  000010                           8.
033564  004737  006430                   JSR     PC,ISIRDY        ;CHK IRDY = 1
033570  000002                           2
033572  004737  005352                   JSR     PC,OACTIV        ;CHK OACT = 1
033576  000001                           1
033600  004737  006714                   JSR     PC,IACTIV        ;CHK IACT = 0
033604  000000                           0
033606  004737  007102                   JSR     PC,RSEOM         ;CHK RSOM = 0, REOM = 1
033612  000002                           2
033614  004737  006430                   JSR     PC,ISIRDY        ;CHK IRDY = 0
033620  000000                           0
033622  123727  002370  000252           CMPB    RAX15,#252       ;COMPARE 3RD RCV'D CHAR TO 252
033630  001404                           BEQ     14$              ;BR IF MATCH
033632  012737  000252  002404           MOV     #252,GOODAT      ;SET EXPECTED DATA
033640  000702                           BR      6$               ;BR TO REPORT ERROR
033642  012737  000014  002400  14$:     MOV     #14,REGNUM       ;SET REG NO. = 14
033650  012737  000040  002366           MOV     #DISSI,WRIBYT
033656  004737  003750                   JSR     PC,WRITLU        ;CLEAR TX ENABLE
033662  012737  000011  002400           MOV     #11,REGNUM       ;SET REG NO. = 11
033670  012737  000200  002366           MOV     #OC,WRIBYT
033676  004737  003750                   JSR     PC,WRITLU        ;SET OC TO SHUT DOWN TRANSMITTER
033702  004737  005146                   JSR     PC,WAIT50        ;WAIT FOR SHUTDOWN
033706  004737  005352                   JSR     PC,OACTIV        ;CHK FOR OACT = 0
033712  000000                           0
033714  005037  002426          18$:     CLR     DISILO           ;CLEAR DISABLE SILO FLAG
033720  004737  003576                   JSR     PC,MSTCLR        ;ISSUE MASTER CLEAR TO CLEAN UP
033724                                   ENDTST
033724                                                                    L10067:
033724  104401                                                                      TRAP    C$ETST





                                ;;*******************************************************************************
                                .SBTTL      TEST 36 - RECEIVER BUFFER TEST - CHAR MODE, CRC
                                ;*
                                ;* FIRST, A MASTER CLEAR IS DONE AND THE PROGRAM CHECKS FOR ICIR = 1 AND IRDY
                                ;* = 0. THEN, 2 SOM CHARS ARE LOADED AND CLOCKED INTO THE USYRT, AND 64
                                ;* BYTES OF A 256-BYTE BINARY COUNT DATA PATTERN (000-377) ARE LOADED INTO
                                ;* THE TX SILO.
                                ;* THE LINE UNIT IS THEN CLOCKED UNTIL IRDY = 1, AND THE PROGRAM CHECKS FOR
                                ;* THIS TO OCCUR WITHIN 40-43 CYCLES. THE PROGRAM READS THE RCV SILO, CHECKS THE
                                ;* CHAR FOR 000, AND CHECKS FOR IRDY = 0 AGAIN.
                                ;* THE LINE UNIT IS THEN CLOCKED IN GROUPS OF 8 CYCLES, AND AFTER EACH, THE
                                ;* PROGRAM CHECKS FOR ICIR = 1, IRDY = 1, UNTIL THE 64TH GROUP, AFTER WHICH
                                ;* IT CHECKS FOR ICIR = 0, IRDY = 1. THE SECOND DATA CHAR IS READ FROM THE
                                ;* RECEIVER SILO AND COMPARED TO 001. THEN, THE PROGRAM CHECKS FOR ICIR = 1,
                                ;* IRDY = 1 AGAIN.
                                ;* THE REST OF THE BINARY COUNT DATA BYTES ARE CYCLED 8 CLOCKS AND READ AND
                                ;* COMPARED A BYTE AT A TIME.
```

```
                                        ;;***********************************************************************
     033726                             BGNTST
     033726                                                                          T36::
     033726  012737  034202  002362             MOV     #A10,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
                                        ;-------------------------------------------------------------------
                                        ; DO MASTER CLR, CHK FOR ICIR = 1, IRDY = 0
                                        ;-------------------------------------------------------------------
     033734  004737  003576                     JSR     PC,MSTCLR        ;ISSUE MASTER CLEAR
     033740  004737  006430                     JSR     PC,ISIRDY        ;CHK ICIR = 1, IRDY = 0
     033744  000001                             1
                                        ;-----------------------------------------------------------------------
                                        ; LOAD AND CLOCK 2 SOM'S, LOAD 64 BYTES OF BINARY COUNT PATTERN INTO TX SILO,
                                        ; CLOCK LINE UNIT, CHK FOR IRDY = 1 WITHIN 40-43 CYCLES
                                        ;-----------------------------------------------------------------------
     033746  004737  005540                     JSR     PC,INITRN        ;LOAD 2 SOM'S, CLOCK THEM INTO USYRT
     033752  000226                             SYNCH
     033754  000011                             STRIP!DDCMP
     033756  005003                             CLR     R3               ;INIT BINARY COUNT DATA FOR WRITING
     033760  010337  002422             2$:     MOV     R3,TXWORD
     033764  004737  005174                     JSR     PC,LDTXSI        ;LOAD A DATA BYTE INTO TX SILO
     033770  005203                             INC     R3               ;INCR DATA
     033772  020327  000100                     CMP     R3,#64.          ;SEE IF 64 BYTES LOADED YET
     033776  002770                             BLT     2$               ;BR IF NOT YET
     034000  004737  007434                     JSR     PC,RCV1ST        ;RECEIVE AND TIME FIRST CHARACTER
     034004  000050                             40.
                                        ;-----------------------------------------------------------------
                                        ; READ RCV SILO, COMPARE FIRST CHAR TO 000, CHK FOR ICIR = 1, IRDY = 0
                                        ;-----------------------------------------------------------------
     034006  005004                     9$:     CLR     R4               ;INIT PATTERN FOR READING
     034010  004737  007750                     JSR     PC,CKDATA        ;READ RCV SILO, COMPARE DATA
     034014  000000                             0                        ;EXPECTED DATA = 000
     034016  000000                             0                        ;DON'T CLOCK LINE UNIT
     034020  005204                     16$:    INC     R4               ;INCR DATA FOR READING
     034022  004737  006430                     JSR     PC,ISIRDY        ;CHK FOR ICIR = 1, IRDY = 0
     034026  000001                             1
                                        ;-----------------------------------------------------------------
                                        ; CLOCK 63 CHARS INTO RCV SILO, CHK ICIR = 1, IRDY = 1
                                        ;-----------------------------------------------------------------
     034030  004737  005254             18$:    JSR     PC,STPLU         ;CLOCK LU FOR 8 CYCLES
     034034  000010                             8.
     034036  010337  002422                     MOV     R3,TXWORD
     034042  004737  005174                     JSR     PC,LDTXSI        ;LOAD ANOTHER WORD INTO TX SILO
     034046  005203                             INC     R3               ;INCR PATTERN FOR WRITING
     034050  004737  006430                     JSR     PC,ISIRDY        ;CHK ICIR = 1, IRDY = 1
     034054  000003                             3
     034056  020327  000177                     CMP     R3,#127.         ;SEE IF 63 MORE CHARS CLOCKED YET
     034062  002762                             BLT     18$              ;BR IF NOT YET
                                        ;-----------------------------------------------------------------
                                        ; CLOCK 1 MORE CHAR INTO RCV SILO, CHK ICIR = 0, IRDY = 1
                                        ;-----------------------------------------------------------------
     034064  004737  005254                     JSR     PC,STPLU         ;CLOCK LU FOR 8 CYCLES
     034070  000010                             8.
     034072  004737  006430                     JSR     PC,ISIRDY        ;CHK ICIR = 0, IRDY = 1
     034076  000002                             2
                                        ;-----------------------------------------------------------------
                                        ;READ, COMPARE, CLOCK REST OF DATA CHARS
                                        ;-----------------------------------------------------------------
```

```
034100  010437  034110            20$:    MOV     R4,21$            ;SET EXPECTED DATA
034104  004737  007750                    JSR     PC,CKDATA         ;READ AND COMPARE DATA
034110  000000                    21$:    0                         ;EXPECTED SILO ENTRY GOES HERE
034112  000000                            0                         ;DON'T CLOCK LINE UNIT
034114  005204                    22$:    INC     R4                ;INCR DATA PATTERN FOR READS
034116  020427  000400                    CMP     R4,#400           ;SEE IF ALL DONE READING YET
034122  001427                            BEQ     32$               ;BR IF DONE READING
034124  004737  006430                    JSR     PC,ISIRDY         ;CHK ICIR = 1, IRDY = 1
034130  000003                            3
034132  004737  005254                    JSR     PC,STPLU          ;CLOCK LU FOR 8 CYCLES
034136  000010                            8.
034140  020327  000377                    CMP     R3,#377           ;SEE IF ALL CHARS LOADED INTO TX SILO YET
034144  003007                            BGT     24$               ;BR IF YES
034146  010337  002422                    MOV     R3,TXWORD
034152  004737  005174                    JSR     PC,LDTXSI         ;LOAD ANOTHER CHAR INTO TX SILO
034156  005203                            INC     R3                ;INCR DATA PATTERN FOR WRITING
034160  000137  034100                    JMP     20$
034164  012737  001000  002422    24$:    MOV     #TXEOM,TXWORD
034172  004737  005174                    JSR     PC,LDTXSI         ;LOAD EOM INTO TX SILO
034176  000137  034100                    JMP     20$
034202                            32$:
034202  004737  003576            A10:    JSR     PC,MSTCLR         ;ISSUE MASTER CLEAR TO CLEAN UP
034206                                    ENDTST
034206                                                                          L10070:
034206  104401                                                                          TRAP    C$ETST

                                  ;;*************************************************************************
                                  .SBTTL      TEST 37 - RECEIVER CHAR LENGTH TIMING TEST - CHAR MODE, NO CRC
                                  ;*
                                  ;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH NO ERROR DETECTION, AND A MSG IS
                                  ;* INITIATED WITH 2 SYNCH CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
                                  ;* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
                                  ;* SET, WHILE THE RECEIVER CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 5,6,7,8.
                                  ;* FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THET USYRT RECEIVER
                                  ;* FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV CHAR LENGTH.
                                  ;* (FOR EXAMPLE A 5-BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED). A MASTER
                                  ;* CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
                                  ;;*************************************************************************
034210                            BGNTST
034210                                                                          T37::
034210  012737  034402  002362            MOV     #24$,RETADR       ;SET TEST EXIT ADRS FOR ERRORS
034216  004737  005540                    JSR     PC,INITRN         ;DO MASTER CLR, LOAD 2 SOM'S
034222  000000                            000
034224  000341                            CRC2!CRC1!IDLE!DDCMP
034226  012701  000017                    MOV     #15.,R1           ;INIT COUNTER
034232  005037  002422                    CLR     TXWORD
034236  004737  005174            3$:     JSR     PC,LDTXSI         ;LOAD A 000 CHAR INTO TX SILO
034242  005301                            DEC     R1                ;DECR COUNTER
034244  001374                            BNE     3$                ;BR IF NOT DONE LOADING YET
034246  004737  005146                    JSR     PC,WAIT50         ;WAIT FOR SILO TO RIPPLE
034252  012737  000006  002402            MOV     #6,AXNUM          ;SET BYTE NO. = 6 FOR AX3
034260  012737  000000  002374            MOV     #000,WAX15        ;SET DATA FOR AX3-15 = 0
034266  012737  000005  002376            MOV     #RXLEN2!RXLEN0,WAX16 ;SET RCV LEN = 5
```

```
034274  004737  004312                    JSR     PC,WRITAX          ;LOAD AX3
034300  004737  005254                    JSR     PC,STPLU           ;CLK LU UNTIL TX'ING 1ST DATA CHAR
034304  100012                            CHPCHK!10.
034306  012737  000006  002376            MOV     #RXLEN2!RXLEN1,WAX16 ;SET RCV LEN = 6
034314  004737  004312                    JSR     PC,WRITAX          ;LOAD AX3
034320  004737  007434                    JSR     PC,RCV1ST          ;CLOCK 5-BIT DATA CHAR
034324  000005                            5
034326  012737  000007  002376            MOV     #RXLEN2!RXLEN1!RXLEN0,WAX16 ;SET RCV LEN = 7
034334  004737  004312                    JSR     PC,WRITAX          ;LOAD AX3
034340  004737  011074                    JSR     PC,RXCHAR          ;RCV 5-BIT DATA CHAR, CLK 6-BIT
034344  000006                            6
034346  012737  000000  002376            MOV     #0,WAX16           ;SET RCV LEN = 8
034354  004737  004312                    JSR     PC,WRITAX          ;LOAD AX3
034360  004737  011074                    JSR     PC,RXCHAR          ;RCV 6-BIT DATA CHAR, CLK 7-BIT
034364  000007                            7
034366  004737  011074                    JSR     PC,RXCHAR          ;RCV 7-BIT DATA CHAR, CLK 8-BIT
034372  000010                            8.
034374  004737  011074                    JSR     PC,RXCHAR          ;RCV 8-BIT DATA CHAR
034400  000010                            8.
034402  004737  003576            24$:    JSR     PC,MSTCLR          ;ISSUE MASTER CLEAR TO CLEAN UP
034406                            ENDTST
034406                                                                          L10071:
034406  104401                                                                          TRAP    C$ETST





                                  ;;*****************************************************************************
                                  .SBTTL      TEST 38 - RECEIVER CHAR LENGTH TIMING TEST - BIT MODE, NO CRC
                                  ;*
                                  ;* THE LINE UNIT IS PLACED IN BIT MODE WITH NO ERROR DETECTION, AND A MESSAGE IS
                                  ;* INITIATED  WITH 2 FLAG CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
                                  ;* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
                                  ;* SET, WHILE THE RCV CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 8,8,8,7,6,5,
                                  ;* 4,3,2,1. FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THAT THE
                                  ;* USYRT RECEIVER FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV
                                  ;* CHAR LENGTH. (FOR EXAMPLE, A 5 BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED).
                                  ;* A MASTER CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
                                  ;;*****************************************************************************
034410                            BGNTST
034410                                                                          T38::
034410  012737  034702  002362            MOV     #24$,RETADR        ;SET TEST EXIT ADRS FOR ERRORS
034416  004737  005540                    JSR     PC,INITRN          ;DO MASTER CLR, LOAD 2 SOM'S
034422  000000                            000
034424  000300                            CRC2!CRC1
034426  012701  000017                    MOV     #15.,R1            ;INIT COUNTER
034432  005037  002422                    CLR     TXWORD
034436  004737  005174            3$:     JSR     PC,LDTXSI          ;LOAD A 000 CHAR INTO TX SILO
034442  005301                            DEC     R1                 ;DECR COUNTER
034444  001374                            BNE     3$                 ;BR IF NOT DONE LOADING YET
034446  004737  005146                    JSR     PC,WAIT50          ;WAIT FOR SILO TO RIPPLE
034452  012737  000006  002402            MOV     #6,AXNUM           ;SET BYTE NO. = 6 FOR AX3
034460  012737  000000  002374            MOV     #000,WAX15         ;SET DATA FOR AX3-15 = 0
034466  004737  007434                    JSR     PC,RCV1ST          ;CLOCK FIRST 8-BIT DATA CHAR
034472  000040                            32.
034474  012737  000000  002376            MOV     #0,WAX16           ;SET RCV LEN = 8
```

```
034502  004737  004312                  JSR     PC,WRITAX          ;LOAD AX3
034506  004737  011074                  JSR     PC,RXCHAR          ;RCV FIRST 8-BIT DATA CHAR, CLK SECOND 8-BIT
034512  000010                          8.
034514  012737  000007  002376          MOV     #RXLEN2!RXLEN1!RXLEN0,WAX16 ;SET RCV LEN = 7
034522  004737  004312                  JSR     PC,WRITAX          ;LOAD AX3
034526  004737  011074                  JSR     PC,RXCHAR          ;RCV SECOND 8-BIT DATA CHAR, CLK 3RD 8-BIT
034532  000010                          8.
034534  012737  000006  002376          MOV     #RXLEN2!RXLEN1,WAX16    ;SET RCV LEN = 6
034542  004737  004312                  JSR     PC,WRITAX          ;LOAD AX3
034546  004737  011074                  JSR     PC,RXCHAR          ;RCV 3RD 8-BIT DATA CHAR, CLK 7-BIT
034552  000007                          7
034554  012737  000005  002376          MOV     #RXLEN2!RXLEN0,WAX16    ;SET RCV LEN = 5
034562  004737  004312                  JSR     PC,WRITAX          ;LOAD AX3
034566  004737  011074                  JSR     PC,RXCHAR          ;RCV 7-BIT DATA CHAR, CLK 6-BIT
034572  000006                          6
034574  012737  000004  002376          MOV     #RXLEN2,WAX16      ;SET RCV LEN = 4
034602  004737  004312                  JSR     PC,WRITAX          ;LOAD AX3
034606  004737  011074                  JSR     PC,RXCHAR          ;RCV 6-BIT DATA CHAR, CLK 5-BIT
034612  000005                          5
034614  012737  000003  002376          MOV     #RXLEN1!RXLEN0,WAX16    ;SET RCV LEN = 3
034622  004737  004312                  JSR     PC,WRITAX          ;LOAD AX3
034626  004737  011074                  JSR     PC,RXCHAR          ;RCV 5-BIT DATA CHAR, CLR 4-BIT
034632  000004                          4
034634  012737  000002  002376          MOV     #RXLEN1,WAX16      ;SET RCV LEN = 2
034642  004737  004312                  JSR     PC,WRITAX          ;LOAD AX3
034646  004737  011074                  JSR     PC,RXCHAR          ;RCV 4-BIT DATA CHAR, CLK 3-BIT
034652  000003                          3
034654  012737  000001  002376          MOV     #RXLEN0,WAX16      ;SET RCV LEN = 1
034662  004737  004312                  JSR     PC,WRITAX          ;LOAD AX3
034666  004737  011074                  JSR     PC,RXCHAR          ;RCV 3-BIT DATA CHAR, CLK 2-BIT
034672  000002                          2
034674  004737  011074                  JSR     PC,RXCHAR          ;RCV 2-BIT DATA CHAR, CLK 1-BIT
034700  000001                          1
034702  004737  003576          24$:    JSR     PC,MSTCLR          ;ISSUE MASTER CLEAR TO CLEAN UP
034706                          ENDTST
034706                                                                     L10072:
034706  104401                                                                     TRAP    C$ETST





                                ;;*****************************************************************
                                .SBTTL      TEST 39 - TRANSMITTER UNDERRUN ERROR, IDLE MARKING, CHAR MODE,NO CRC
                                ;*
                                ;* THE LINE UNIT IS PLACED IN CHAR MODE, AND THE IDLE BIT IS SET. THEN, A
                                ;* MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX BUFFER IS NOT SERVICED
                                ;* IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH CAUSES A TX UNDERRUN
                                ;* ERROR. THEN, THE RECEIVER IS CLOCKED AND CHECKED FOR TWO 377 CHARS TO BE
                                ;* RECEIVED (LINE MARKING) BEFORE SHUTTING DOWN WITH A MASTER CLEAR.
                                ;;*****************************************************************
034710                          BGNTST
034710                                                                     T39::
034710  012737  034772  002362          MOV     #24$,RETADR        ;SET TEST EXIT ADDRESS FOR ERRORS
034716  004737  005540                  JSR     PC,INITRN          ;MST CLR, LOAD 2 SOM'S
034722  000226                          SYNCH
034724  000341                          CRC2!CRC1!IDLE!DDCMP
```

```
034726  012737  000000  002422          MOV     #000,TXWORD
034734  004737  005174                  JSR     PC,LDTXSI       ;LOAD 000 CHAR INTO TX SILO
034740  004737  005254                  JSR     PC,STPLU        ;CLOCK LINE UNIT UNTIL LINE GOES MARKING
034744  000063                          51.
034746  004737  007354                  JSR     PC,RDRXSI       ;READ 000 CHAR
034752  004737  007750                  JSR     PC,CKDATA       ;READ AND CHECK FOR MARK CHAR (377)
034756  000377                          377
034760  000000                          000
034762  004737  007750                  JSR     PC,CKDATA       ;READ AND CHECK FOR ANOTHER MARK CHAR
034766  000377                          377
034770  000000                          000
034772  004737  003576          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
034776                          ENDTST
034776                                                                          L10073:
034776  104401                                                                          TRAP    C$ETST





                                ;;*********************************************************************
                                .SBTTL      TEST 40 - MSG TERMINATION WITH GA CHARS - BIT MODE,NO CRC
                                ;*
                                ;* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
                                ;* INITIATED IN BIT MODE.
                                ;* 2 FLAG CHARACTERS ARE SENT, FOLLOWED BY
                                ;* THE FOLLOWING DATA CHARACTERS : 000, 125, 252, 377, 000. THEN THE LOOP
                                ;* MODE BIT (STRIP) IS SET AND 2 TERMINATING GO-AHEAD CHARACTERS ARE
                                ;* SENT. EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 5 DATA
                                ;* WORDS ARE READ AND COMPARED TO EXPECTED VALUES.
                                ;* ALSO, THE FIRST GA CHAR IS CHECKED BY SCANNING THE TXDATA BIT AS THE GA
                                ;* IS BEING TRANSMITTED (GA = 376 OCTAL).
                                ;* THE TEST ALSO CHECKS FOR SETTING OF RAB AND EBLK
                                ;* IN LOOP MODE.
                                ;;*********************************************************************
035000                          BGNTST
035000                                                                  T40::
035000  012737  035134  002362          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
035006  004737  005540                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
035012  000000                          000
035014  000310                          CRC2!CRC1!STRIP
035016  004737  011016                  JSR     PC,LODMSG       ;LOAD DATA CHARS INTO TX SILO
035022  003224                          MSG1+4
035024  000005                          5
035026  012737  005000  002422          MOV     #TXGOA!TXEOM,TXWORD
035034  004737  005174                  JSR     PC,LDTXSI       ;LOAD A GA CHAR INTO TX SILO
035040  004737  005174                  JSR     PC,LDTXSI       ;LOAD ANOTHER GA
035044  004737  005146                  JSR     PC,WAIT50       ;ALLOW SILO TO RIPPLE
035050  004737  005254                  JSR     PC,STPLU        ;CLOCK LU UNTIL GA CHAR IS TX'ING
035054  100071                          CHPCHK!57.
035056  004737  011176                  JSR     PC,CKTBIT       ;SCAN TXDATA BIT FOR GA CHAR
035062  000376                          376
035064  004737  007750                  JSR     PC,CKDATA       ;RCV 000 CHAR, CLK 125
035070  000000                          000
035072  000000                          0
035074  004737  007750                  JSR     PC,CKDATA       ;RCV 125 CHAR, CLK 252
035100  000125                          125
```

```
035102  000000                               0
035104  004737  007750                       JSR     PC,CKDATA       ;RCV 252 CHAR, CLK 377
035110  000252                               252
035112  000000                               0
035114  004737  007750                       JSR     PC,CKDATA       ;RCV 377 CHAR
035120  000377                               377
035122  000010                               8.
035124  004737  007750                       JSR     PC,CKDATA       ;RCV 000 CHAR, CHK RAB = 1, EBLK = 1
035130  003000                               3000
035132  000010                               8.
035134  004737  003576               24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR
035140                               ENDTST
035140                                                                               L10074:
035140  104401                                                                               TRAP    C$ETST





                                     ;;*********************************************************************
                                     .SBTTL          TEST 41 - IDLE SYNCHS TEST - CHAR MODE
                                     ;*
                                     ;* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
                                     ;* INITIATED IN CHAR MODE. 24(DEC) SYNCHS ARE SENT.
                                     ;* EACH SYNCH IS TIMED AS IT IS RECEIVED, AND THE BITS ARE CHECKED
                                     ;* FOR A VALID SYNCH CHAR FOR EACH OF THE 22 SYNCHS WHICH FOLLOW
                                     ;* THE FIRST TWO (THESE PERFORM SYNCHRONIZATION, AND ARE NOT READ).
                                     ;* WHILE THE LAST SYNCH IS BEING TRANSMITTED, OC IS SET, AND THE
                                     ;* NEXT CHAR RCV'D AFTER THE SYNCH IS CHECKED TO BE 377 (LINE MARKING).
                                     ;* THEN, A MASTER CLEAR IS ISSUED.
                                     ;* THE SYNCH CHAR USED IS 226 (OCTAL).
                                     ;;*********************************************************************
035142                               BGNTST
035142                                                                               T41::
035142  012737  035302  002362               MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
035150  004737  005540                       JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
035154  000226                               SYNCH
035156  000341                               CRC2!CRC1!IDLE!DDCMP
035160  012701  000026                       MOV     #22.,R1         ;INIT COUNTER
035164  012737  000226  002422               MOV     #SYNCH,TXWORD
035172  004737  005174               6$:     JSR     PC,LDTXSI       ;LOAD AN SOM INTO TX SILO
035176  005301                               DEC     R1              ;DECR COUNTER
035200  001374                               BNE     6$              ;BR IF MORE TO LOAD
035202  004737  005146                       JSR     PC,WAIT50       ;ALLOW SILO TO RIPPLE
035206  004737  007434                       JSR     PC,RCV1ST       ;CLK LU UNTIL 3RD SYNCH RCV'D (RCVR IS ACTIVE)
035212  000030                               24.
035214  012701  000025                       MOV     #21.,R1         ;INIT COUNTER
035220  004737  007750               8$:     JSR     PC,CKDATA       ;READ A SYNCH, CLK NEXT ONE
035224  000226                               SYNCH
035226  000010                               8.
035230  005301                               DEC     R1              ;DERC COUNTER
035232  001372                               BNE     8$              ;BR IF NOT ALL CHECKED
035234  004737  007750                       JSR     PC,CKDATA       ;CHECK LAST SYNCH
035240  000226                               SYNCH
035242  000004                               4
035244  012737  000011  002400               MOV     #11,REGNUM      ;SET REG NO. = 11
035252  112737  000200  002366               MOVB    #OC,WRIBYT
```

```
035260  004737  003750                  JSR     PC,WRITLU       ;SET OC IN REG 11
035264  004737  005254                  JSR     PC,STPLU        ;FINISH CLOCKING CHAR, AND THEN SOME
035270  000014                          12.
035272  004737  007750                  JSR     PC,CKDATA       ;RCV A MARK CHAR, CHK IT
035276  000377                          377
035300  000000                          0
035302  004737  003576          24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
035306                                  ENDTST
035306                                                                  L10075:
035306  104401                                                                  TRAP    C$ETST

                                ;;***********************************************************************
                                .SBTTL          TEST 42 - STRIP SYNCH TEST
                                ;*
                                ;* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
                                ;* INITIATED IN CHAR MODE AND WITH THE STRIP SYNCH
                                ;* BIT SET. THEN 24 (DEC) SYNCHS ARE SENT
                                ;* FOLLOWED BY THE FOLLOWING DATA CHARACTERS : 377, 000, 125, 252,
                                ;* AND 2 TERMINATING SYNCHS.
                                ;* EACH OF THE 23 SYNCHS AFTER THE FIRST ARE CHECKED AT THE TRANSMITTER OUTPUT,
                                ;* BY SCANNING THE TXDATA BIT.
                                ;* EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 4 DATA WORDS
                                ;* ARE READ AND COMPARED TO EXPECTED VALUES.
                                ;* FINALLY, THE LINE UNIT IS CLOCKED FOR SEVERAL CHAR TIMES, AND A CHECK
                                ;* IS MADE FOR OACT = 0 (TEOM SHOULD CAUSE TX ENABLE TO DROP).
                                ;* THE ABOVE TEST IS REPEATED FOR EACH OF THE FOLLOWING SYNCH CHAR DATA
                                ;* PATTERNS : 226,000,125,252,376,177.
                                ;;***********************************************************************
035310                          BGNTST
035310                                                          T42::
035310  012737  035530  002362          MOV     #24$,RETADR     ;SET TEST EXIT ADRS FOR ERRORS
035316  012701  035536                  MOV     #SYNPAT,R1      ;GET POINTER TO DATA
035322  011137  035332          2$:     MOV     (R1),3$         ;GET A SYNCH PATTERN
035326  004737  005540                  JSR     PC,INITRN       ;MST CLR, LOAD 2 SOM'S
035332  000000                  3$:     .WORD   0               ;SYNCH PATTERN GOES HERE
035334  000311                          CRC2!CRC1!STRIP!DDCMP
035336  012737  000400  002422          MOV     #TXSOM,TXWORD
035344  012702  000026                  MOV     #22.,R2         ;LOAD 22 SOM'S INTO TX SILO
035350  004737  005174          6$:     JSR     PC,LDTXSI
035354  005302                          DEC     R2
035356  001374                          BNE     6$
035360  004737  011016                  JSR     PC,LODMSG       ;LOAD DATA CHARS INTO TX SILO
035364  003246                          MSG4
035366  000006                          6
035370  012737  001000  002422          MOV     #TXEOM,TXWORD
035376  004737  005174                  JSR     PC,LDTXSI       ;LOAD A TEOM
035402  004737  005174                  JSR     PC,LDTXSI       ;LOAD ANOTHER TEOM
035406  004737  005146                  JSR     PC,WAIT50       ;ALLOW SILO TO RIPPLE
035412  011137  035434                  MOV     (R1),16$        ;GET CURRENT SYNCH PATTERN
035416  012702  000027                  MOV     #23.,R2         ;INIT COUNTER
035422  004737  005254                  JSR     PC,STPLU        ;CLOCK OUT FIRST SYNCH
035426  100010                          CHPCHK!8.
035430  004737  011176          14$:    JSR     PC,CKTBIT       ;CHECK TX'D SYNCH
```

```
035434  000000                   16$:    .WORD   0               ;SYNCH PATTERN GOES HERE
035436  005302                           DEC     R2              ;DECR COUNTER
035440  001373                           BNE     14$             ;BR IF NOT DONE CHECKING LAST 23 SYNCHS
035442  004737  007434                   JSR     PC,RCV1ST       ;CLOCK UNTIL 000 CHAR RCV'D
035446  000010                           8.
035450  004737  007750                   JSR     PC,CKDATA       ;RCV 377, CLOCK 000
035454  000377                           377
035456  000010                           8.
035460  004737  007750                   JSR     PC,CKDATA       ;RCV 000, CLK 125
035464  000000                           000
035466  000010                           8.
035470  004737  007750                   JSR     PC,CKDATA       ;RCV 125, CLK 252
035474  000125                           125
035476  000010                           8.
035500  004737  007750                   JSR     PC,CKDATA       ;RCV 252, CLK END OF MSG
035504  000252                           252
035506  000040                           32.
035510  004737  005352                   JSR     PC,OACTIV       ;CHK FOR OACT = 0
035514  000000                           0
035516  062701  000002                   ADD     #2,R1           ;INIT SYNCH PATTERN POINTER
035522  020127  035552                   CMP     R1,#SYNPAT+12.  ;SEE IF ALL PATTERNS CHECKED YET
035526  103675                           BLO     2$              ;BR IF NOT YET
035530  004737  003576           24$:    JSR     PC,MSTCLR       ;ISSUE MASTER CLEAR TO CLEAN UP
035534                                   ENDTST
035534                                                           L10076:
035534  104401                                                           TRAP    C$ETST

035536  000226                   SYNPAT: 226
035540  000000                           000
035542  000125                           125
035544  000252                           252
035546  000376                           376
035550  000177                           177
```

```
                                        .SBTTL  HARDWARE PARAMETER CODING SECTION



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

035552                                         BGNHRD
035552  000022                                                                          .WORD L10077-L$HARD/2
035554                                                                         L$HARD::

035554                                         GPRMA   ADDRES,2,0,160000,177776,YES
035554  001031                                                                          .WORD   T$CODE
035556  035620                                                                          .WORD   ADDRES
035560  160000                                                                          .WORD   T$LOLIM
035562  177776                                                                          .WORD   T$HILIM
035564                                         GPRMA   VECTOR,4,0,0,674,YES
035564  002031                                                                          .WORD   T$CODE
035566  035646                                                                          .WORD   VECTOR
035570  000000                                                                          .WORD   T$LOLIM
035572  000674                                                                          .WORD   T$HILIM
035574                                         GPRMD   PRIRTY,6,0,7000,4,7,YES
035574  003032                                                                          .WORD   T$CODE
035576  035677                                                                          .WORD   PRIRTY
035600  007000                                                                          .WORD   7000
035602  000004                                                                          .WORD   T$LOLIM
035604  000007                                                                          .WORD   T$HILIM
035606                                         GPRMD   ISRUN,24,0,7,0,7,YES
035606  012032                                                                          .WORD   T$CODE
035610  035730                                                                          .WORD   ISRUN
035612  000007                                                                          .WORD   7
035614  000000                                                                          .WORD   T$LOLIM
035616  000007                                                                          .WORD   T$HILIM

035620                                         ENDHRD
                                                                                        .EVEN
035620                                                                         L10077:

035620     104     105     126  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
035623     111     103     105
035626     040     103     123
035631     122     040     101
035634     104     104     122
035637     105     123     123
035642     040     072     040
035645     000
035646     104     105     126  VECTOR: .ASCIZ  /DEVICE VECTOR ADDRESS : /
035651     111     103     105
035654     040     126     105
035657     103     124     117
035662     122     040     101
```

```
   035665    104    104    122
   035670    105    123    123
   035673    040    072    040
   035676    000
   035677    104    105    126  PRIRTY: .ASCIZ  /DEVICE PRIORITY LEVEL : /
   035702    111    103    105
   035705    040    120    122
   035710    111    117    122
   035713    111    124    131
   035716    040    114    105
   035721    126    105    114
   035724    040    072    040
   035727    000
   035730    115    070    062  ISRUN:  .ASCIZ  /M8207 RUN SWITCH (E28 SW7) - TYPE 0 IF OFF, 1 IF ON : /
   035733    060    067    040
   035736    122    125    116
   035741    040    123    127
   035744    111    124    103
   035747    110    040    050
   035752    105    062    070
   035755    040    123    127
   035760    067    051    040
   035763    055    040    124
   035766    131    120    105
   035771    040    060    040
   035774    111    106    040
   035777    117    106    106
   036002    054    040    061
   036005    040    111    106
   036010    040    117    116
   036013    040    072    040
   036016    000
                                  .EVEN
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION


                                ;//////////////////////////////////////////////////////////////////////
                                ;/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ;/ WITH THE OPERATOR.
                                ;//////////////////////////////////////////////////////////////////////

036020                                  BGNSFT
036020  000000                                                                  .WORD L10100-L$SOFT/2
036022                                                                  L$SOFT::


036022                                  ENDSFT
                                                                                .EVEN
036022                                                                  L10100:

                                .EVEN








                                ;****** PATCH AREA FOR DEBUG ******
036022                          PATCH:
        036222                          .=.+200
036222  000240                          NOP
036224  000240                          NOP
036226  000240                          NOP
                                ;**********************************
        .


036230                                  ENDMOD

036230                                  LASTAD
                                                                                .EVEN
036230  000000                                                                  .WORD   0
036232  000000                                                                  .WORD   0
036234                          L$LAST::

        000001                  .END
```

SYMBOL TABLE

```
ABORT = 000004
ADDRES  035620
ADR   = 000020 G
ANBITS  002561
APA   = 000200
ASBC0 = 000020
ASBC1 = 000040
ASBC2 = 000100
ASSEMB= 000010
AXNUM   002402
AX0.15= 002322
AX0.16= 002324
AX1   = 000001
AX1.15= 002326
AX1.16= 002330
AX2   = 000002
AX2.15= 002332
AX2.16= 002334
AX3.15= 002336
AX3.16= 002340
AX315U= 000372
A1      024220
A10     034202
A2      026626
A3      027306
A4      027422
A5      027532
A6      027726
A7      030272
A8      030506
A9      031036
BADDAT  002406
BCC   = 000001
BCCCHK= 100000
BIT0  = 000001 G
BIT00 = 000001 G
BIT01 = 000002 G
BIT02 = 000004 G
BIT03 = 000010 G
BIT04 = 000020 G
BIT05 = 000040 G
BIT06 = 000100 G
BIT07 = 000200 G
BIT08 = 000400 G
BIT09 = 001000 G
BIT1  = 000002 G
BIT10 = 002000 G
BIT11 = 004000 G
BIT12 = 010000 G
BIT13 = 020000 G
BIT14 = 040000 G
BIT15 = 100000 G
BIT2  = 000004 G
BIT3  = 000010 G
BIT4  = 000020 G
BIT5  = 000040 G
BIT6  = 000100 G
BIT7  = 000200 G
BIT8  = 000400 G
BIT9  = 001000 G
BOE   = 000400 G
BPOLL = 000100
BSEL1   002450
BSEL2   002452
BSEL4   002454
CARR  = 000001
CHPCHK= 100000
CHPTYP  002430
CKDATA  007750
CKTBIT  011176
CRCCHK= 100000
CRCTY0= 000001
CRCTY1= 000002
CRCTY2= 000004
CRC1  = 000100
CRC2  = 000200
CS    = 000004
C$AU  = 000052
C$AUTO= 000061
C$BRK = 000022
C$BSEG= 000004
C$BSUB= 000002
C$CEFG= 000045
C$CLCK= 000062
C$CLEA= 000012
C$CLOS= 000035
C$CLP1= 000006
C$CVEC= 000036
C$DCLN= 000044
C$DODU= 000051
C$DRPT= 000024
C$DU  = 000053
C$EDIT= 000003
C$ERDF= 000055
C$ERHR= 000056
C$ERPO= 000060
C$ERSF= 000054
C$ERSO= 000057
C$ESCA= 000010
C$ESEG= 000005
C$ESUB= 000003
C$ETST= 000001
C$EXIT= 000032
C$GETB= 000026
C$GETW= 000027
C$GMAN= 000043
C$GPHR= 000042
C$GPLO= 000030
C$GPRI= 000040
C$INIT= 000011
C$INLP= 000020
C$MANI= 000050
C$MEM = 000031
C$MSG = 000023
C$OPEN= 000034
C$PNTB= 000014
C$PNTF= 000017
C$PNTS= 000016
C$PNTX= 000015
C$QIO = 000377
C$RDBU= 000007
C$REFG= 000047
C$RESE= 000033
C$REVI= 000003
C$RFLA= 000021
C$RPT = 000025
C$SEFG= 000046
C$SPRI= 000041
C$SVEC= 000037
C$TPRI= 000013
C32BCC= 000040
C32ENB= 000004
DDC   = 000100
DDCMP = 000001
DEVMAP  002434
DEVPTR  002436
DFPTBL  002252 G
DH1     014264
DH2     014306
DH3     014335
DH4     014373
DH5     014435
DH6     014440
DH7     014443
DH8     014475
DH9     014534
DIAGMC= 000000
DISILO  002426
DISSI = 000040
DTR   = 000100
EBLK  = 000002
EF.CON= 000036 G
EF.NEW= 000035 G
EF.PWR= 000034 G
EF.RES= 000037 G
EF.STA= 000040 G
EM1     012125
EM10    012447
EM11    012470
EM12    012505
EM13    012526
EM14    012556
EM15    012573
EM16    012622
EM17    012643
EM18    012660
EM19    012701
EM2     012161
EM20    012716
EM21    012737
EM22    012754
EM23    012775
EM24    013016
EM25    013033
EM26    013067
EM27    013121
EM28    013152
EM29    013173
EM3     012220
EM30    013210
EM31    013231
EM32    013246
EM33    013275
EM34    013320
EM35    013346
EM36    013366
EM37    013402
EM38    013423
EM39    013440
EM4     012237
EM40    013460
EM41    013474
EM42    013515
EM43    013532
EM44    013557
EM45    013604
EM46    013631
EM47    013664
EM48    013717
EM49    013752
EM5     012312
EM50    014011
EM51    014045
EM52    014105
EM53    014146
EM54    014206
EM6     012343
EM60    014230
EM65    014244
EM7     012374
EM8     012411
EM9     012432
ENAX  = 000004
ENDIT   021464
ENDPAT  003220
ENDTRN  006274
EOM   = 000002
ERRFLG  002356
ERROR1  002420
ERR1    014600 G
ERR2    014632 G
ERR3    015140 G
ERR4    015622 G
ERR5    016300 G
ERR6    017012 G
ERR7    017470 G
ERR8    020122 G
ERR9    020630 G
EVL   = 000004 G
E$END = 002100
E$LOAD= 000035
FMT1    011454
FMT10   011717
FMT11   011750
FMT19   012007
FMT2    011464
FMT24   012040
FMT27   021600
FMT3    011506
FMT4    011550
FMT5    011563
FMT6    011613
FMT7    011646
FMT8    011656
FMT9    011712
FRSTIM  002412
F$AU  = 000015
F$AUTO= 000020
F$BGN = 000040
F$CLEA= 000007
F$DU  = 000016
F$END = 000041
F$HARD= 000004
F$HW  = 000013
F$INIT= 000006
F$JMP = 000050
F$MOD = 000000
F$MSG = 000011
F$PROT= 000021
F$PWR = 000017
F$RPT = 000012
F$SEG = 000003
F$SOFT= 000005
F$SRV = 000010
F$SUB = 000002
F$SW  = 000014
F$TEST= 000001
GETALL  004526
GETPRM  021306
GETREG  004016
GOAH  = 000010
GOODAT  002404
G$CNTO= 000200
G$DELM= 000372
G$DISP= 000003
G$EXCP= 000400
G$HILI= 000002
G$LOLI= 000001
G$NO  = 000000
G$OFFS= 000400
G$OFSI= 000376
G$PRMA= 000001
G$PRMD= 000002
G$PRML= 000000
G$RADA= 000140
```

```
G$RADB= 000000
G$RADD= 000040
G$RADL= 000120
G$RADO= 000020
G$XFER= 000004
G$YES = 000010
HDX   = 000020
HELP  = 000001
HOE   = 100000 G
IACT  = 000100
IACTIV  006714
IBE   = 010000 G
IC    = 000200
ICIR  = 000010
IDL   = 000010
IDLE  = 000040
IDU   = 000040 G
IER   = 020000 G
IERR  = 000002
INITRN  005540
INTFLG  002354
INTGRL= 000010
IRDY  = 000020
ISIRDY  006430
ISR   = 000100 G
ISRUN   035730
IXE   = 004000 G
I$AU  = 000041
I$AUTO= 000041
I$CLN = 000041
I$DU  = 000041
I$HRD = 000041
I$INIT= 000041
I$MOD = 000041
I$MSG = 000041
I$PROT= 000040
I$PTAB= 000041
I$PWR = 000041
I$RPT = 000041
I$SEG = 000041
I$SETU= 000041
I$SFT = 000041
I$SRV = 000041
I$SUB = 000041
I$TST = 000041
I422  = 000200
J$JMP = 000167
LDMSG1  011374
LDTXSI  005174
LOADAT  002410
LODMSG  011016
LOE   = 040000 G
LOGDEV  002344
LOOPIN  004074
LOT   = 000010 G
LULOOP= 000010
LULP  = 000040
LUREG   002302
LUR10 = 002302
LUR11 = 002304
LUR12 = 002306
LUR13 = 002310
LUR14 = 002312
LUR15 = 002314
LUR16 = 002316
LUR17 = 002320
LUSWI1  002466
LUSWI2  002470
LUSWI3  002472
LU1MOD  002000 G
L$ACP   002110 G
L$APT   002036 G
L$AU    021632 G
L$AUT   002070 G
L$AUTO  021466 G
L$CCP   002106 G
L$CLEA  021546 G
L$CO    002032 G
L$DEPO  002011 G
L$DESC  003470 G
L$DESP  002076 G
L$DEVP  002060 G
L$DISP  002124 G
L$DLY   002116 G
L$DTP   002040 G
L$DTYP  002034 G
L$DU    021550 G
L$DUT   002072 G
L$DVTY  003462 G
L$EF    002052 G
L$ENVI  002044 G
L$ETP   002102 G
L$EXP1  002046 G
L$EXP4  002064 G
L$EXP5  002066 G
L$HARD  035554 G
L$HIME  002120 G
L$HPCP  002016 G
L$HPTP  002022 G
L$HW    002252 G
L$ICP   002104 G
L$INIT  021116 G
L$LADP  002026 G
L$LAST  036234 G
L$LOAD  002100 G
L$LUN   002074 G
L$MREV  002050 G
L$NAME  002000 G
L$PRIO  002042 G
L$PROT  021110 G
L$PRT   002112 G
L$REPP  002062 G
L$REV   002010 G
L$RPT   021106 G
L$SOFT  036022 G
L$SPC   002056 G
L$SPCP  002020 G
L$SPTP  002024 G
L$STA   002030 G
L$SW    002302 G
L$TEST  002114 G
L$TIML  002014 G
L$UNIT  002012 G
L10000  002300
L10001  002302
L10002  014630
L10003  015136
L10004  015620
L10005  016276
L10006  017010
L10007  017466
L10010  020120
L10011  020626
L10012  021104
L10013  021106
L10015  021464
L10016  021544
L10017  021546
L10020  021576
L10021  021632
L10022  021714
L10023  022000
L10024  022124
L10025  022226
L10026  022354
L10027  022474
L10030  022650
L10031  023104
L10032  023222
L10033  023340
L10034  023456
L10035  023574
L10036  024224
L10037  024556
L10040  025046
L10041  025256
L10042  025462
L10043  025666
L10044  026124
L10045  026356
L10046  026262
L10047  026354
L10050  026630
L10051  026626
L10052  027306
L10053  027426
L10054  027536
L10055  027732
L10056  030276
L10057  030506
L10060  031036
L10061  031136
L10062  031506
L10063  032050
L10064  032320
L10065  032662
L10066  033012
L10067  033724
L10070  034206
L10071  034406
L10072  034706
L10073  034776
L10074  035140
L10075  035306
L10076  035534
L10077  035620
L10100  036022
MAINT1= 000010
MAINT2= 000004
MCLK  = 000002
MCLR  = 000100
MODR  = 000010
MPCSR   002446
MPIVEC  002460
MPOVEC  002462
MPRIOR  002464
MSG1    003220
MSG4    003246
MSTCLR  003576
MVIOX = 021000
MVIXO = 122000
MVIXOX= 121000
NEWST   021266
OACT  = 000100
OACTIV  005352
OC    = 000200
OCOR  = 000020
OP    = 000002
ORDY  = 000020
OSIRDY  004662
OVRR  = 000010
O$APTS= 000000
O$AU  = 000001
O$BGNR= 000000
O$BGNS= 000000
O$DU  = 000001
O$ERRT= 000000
O$GNSW= 000000
O$POIN= 000001
O$SETU= 000000
PATA    002571
PATB    002615
PATC    002625
PATCH   036022
PATD    002630
PATE    002633
PATF    002641
PATG    002644
PATH    002654
PATI    002664
PATJ    002674
PATK    002704
PATL    003014
PATM    003022
PATN    003032
PATO    003050
PATP    003066
PATU    003104
PATV    003152
PNT   = 001000 G
POLL  = 000200
PRI   = 002000 G
PRIOR   002350
PRIRTY  035677
PRI00 = 000000 G
PRI01 = 000040 G
PRI02 = 000100 G
PRI03 = 000140 G
PRI04 = 000200 G
PRI05 = 000240 G
PRI06 = 000300 G
PRI07 = 000340 G
PSTACK  002346
RAB   = 000004
RABT  = 000004
RAX15   002370
RAX16   002372
RCVBUF  003262
RCV1ST  007434
RDALL = 000004
RDAX  = 000020
RDRXSI  007354
READAX  004124
READLU  003672
READY = 000200
REDBYT  002364
REDDAT  002500
REGNUM  002400
REG0    002510
REG1    002512
REG2    002514
REG3    002516
REG4    002520
REG5    002522
REG6    002524
REG7    002526
REOM  = 000002
RERR  = 000200
RETADR  002362
RING  = 000200
ROMI  = 000002
ROMO  = 000004
ROR   = 000010
RRDYTO= 000001
RSEOM   007102
```

```
RSOM   = 000001     STPCLK   003540     TXABT  = 002000     T$TEMP= 000000      T29        031040 G
RTS    = 000040     STPERR   007646     TXCHAR   006122     T$TEST= 000052      T3         022002 G
RUN    = 000200     STPLU    005254     TXDATA= 000040      T$TSTM= 177777      T30        031140 G
RUNINH   002476     STR    = 000040     TXEN   = 000100     T$TSTS= 000001      T31        031510 G
RXABT  = 002000     STRIP  = 000010     TXEOM  = 001000     T$$AU  = 010021     T32        032052 G
RXBCC  = 000400     SUBRPC   002352     TXGA   = 000010     T$$AUT= 010016      T33        032322 G
RXCHAR   011074     SVCGBL= 000000      TXGOA  = 004000     T$$CLE= 010017      T34        032664 G
RXEBL  = 001000     SVCINS= 000001      TXLEN0= 000040      T$$DU  = 010020     T35        033014 G
RXLEN0= 000001      SVCSUB= 000001      TXLEN1= 000100      T$$HAR= 010077      T36        033726 G
RXLEN1= 000002      SVCTAG= 000001      TXLEN2= 000200      T$$HW  = 010000     T37        034210 G
RXLEN2= 000004      SVCTST= 000001      TXSOM  = 000400     T$$INI= 010015      T38        034410 G
RXOVR  = 004000     SW0    = 000002     TXWORD   002422     T$$MSG= 010012      T39        034710 G
RXWORD   002424     SW1    = 000004     TX0    = 000001     T$$PRO= 010014      T4         022126 G
RX0    = 000001     SW2    = 000010     TX1    = 000002     T$$RPT= 010013      T40        035000 G
RX1    = 000002     SW3    = 000040     TX2    = 000004     T$$SEG= 010000      T41        035142 G
RX2    = 000004     SYNCH  = 000226     TX3    = 000010     T$$SOF= 010100      T42        035310 G
RX3    = 000010     SYNPAT   035536     TX4    = 000020     T$$SUB= 010051      T5         022230 G
RX4    = 000020     SYN0   = 000001     TX5    = 000040     T$$SW  = 010001     T6         022356 G
RX5    = 000040     SYN1   = 000002     TX6    = 000100     T$$TES= 010076      T7         022476 G
RX6    = 000100     SYN2   = 000004     TX7    = 000200     T1         021634 G T8         022652 G
RX7    = 000200     SYN3   = 000010     T$ARGC= 000001      T10        023224 G T9         023106 G
R14NRW   002560     SYN4   = 000020     T$CODE= 012032      T11        023342 G UAM    = 000200 G
SAVE4    002414     SYN5   = 000040     T$ERRN= 000032      T12        023460 G UNIT       002440
SAVE6    002416     SYN6   = 000100     T$EXCP= 000000      T13        023576 G UNRR   = 000001
SAVLEN   002432     SYN7   = 000200     T$FLAG= 000040      T14        024226 G UPBITS     002550
SCRACH   002342     S$LSYM= 010000      T$GMAN= 000000      T15        024560 G VECTOR     035646
SEC    = 000020     TEOM   = 000002     T$HILI= 000007      T16        025050 G V35    = 000020
SECA   = 000020     TERR   = 000200     T$LAST= 000001      T17        025260 G WAIT50     005146
SELFR  = 000040     TEST   = 000001     T$LOLI= 000000      T18        025464 G WAX    = 000010
SELSBY= 000002      TESTMD= 000004      T$LSYM= 010000      T19        025670 G WAX15      002374
SEL4     002454     TIMFLG   002360     T$LTNO= 000052      T2         021716 G WAX16      002376
SEL6     002456     TMP0     002530     T$NEST= 177777      T20        026126 G WRDYTO= 000002
SETUP    010640     TMP1     002532     T$NS0  = 000000     T20.1      026156   WRIBYT     002366
SFPTBL   002302 G   TMP2     002534     T$NS1  = 000005     T20.2      026264   WRITAX     004312
SIGQ   = 000100     TMP3     002536     T$NS2  = 000002     T21        026360 G WRITLU     003750
SIGR   = 000200     TMP4     002540     T$NS3  = 000003     T21.1      026372   XYZ    = 000100
SOM    = 000001     TMP5     002542     T$PTNU= 000000      T22        026632 G X$ALWA= 000000
STALL    005164     TMP6     002544     T$SAVL= 177777      T23        027310 G X$FALS= 000040
STARES   002444     TMP7     002546     T$SEGL= 177777      T24        027430 G X$OFFS= 000400
STARST   021256     TSOM   = 000001     T$SEKO= 010000      T25        027540 G X$TRUE= 000020
STBY   = 000002     TSTCON   002474     T$SUBN= 000000      T26        027734 G $LSTIN= 000001
STEPLU= 000020      TSTNUM   002442     T$TAGL= 177777      T27        030300 G $LSTTA= 000001
STEPMP= 000001      TXAB   = 000004     T$TAGN= 010101      T28        030510 G

. ABS.  036234      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  21669 WORDS  ( 85 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  69 PAGES
CZDMRD.BIC,CZDMRD.SEQ/C/N:TOC=SVC34R.MLB,CZDMRD.P11
```

```
$LSTIN    7-27#
$LSTTA    7-28#
A1       24-457    24-483    24-502    24-511    24-521#
A10      24-D59    24-E34#
A2       24-959    24-998#
A3       24-:23    24-;27    24-;31#
A4       24-;53    24-;79#
A5       24-<02    24-<25#
A6       24-<50    24-<85#
A7       24-=11    24-=27    24-=36    24-=48    24-=67    24-=71    24-=75#
A8       24-=92    24->31#
A9       24->49    24-?11#
ABORT    12-61#
ADDRES   16-101    16-107    16-122    16-142    16-160    16-181    16-199    16-218    16-237    25-16     25-23#
ADR      12-20#
ANBITS   13-112#   24-684    24-685    24-735    24-736    24-782    24-783
APA      12-239#
ASBC0    12-196#
ASBC1    12-195#
ASBC2    12-194#
ASSEMB    7-18      7-18
AX0.15   12-309#   15-235    16-131    16-150    16-170    16-189    16-207    16-227
AX0.16   12-310#   16-131    16-150    16-170    16-189    16-207    16-227
AX1      12-92#    12-164#
AX1.15   12-311#   16-131    16-150    16-170    16-189    16-207    16-227
AX1.16   12-312#   16-131    16-150    16-170    16-189    16-207    16-227
AX2      12-91#    12-163#
AX2.15   12-313#   16-133    16-152    16-172    16-191    16-209    16-229
AX2.16   12-314#   16-133    16-152    16-172    16-191    16-209    16-229
AX3.15   12-315#   16-133    16-152    16-172    16-191    16-209    16-229
AX3.16   12-316#   16-133    16-152    16-172    16-191    16-209    16-229
AX315U   12-259#   24-569    24-629    24-846
AXNUM    13-33#    15-152    15-190    15-203    15-228    15-230    15-236*   15-242*   15-243    15-245*   15-247    15-438    15-446*   15-489*
         15-685    15-691*   15-720*   15-726*   15-967    15-969*   15-977*   15-986*   24-541*   24-553*   24-557*   24-567    24-579*   24-590*
         24-618*   24-625*   24-627    24-639*   24-681*   24-696*   24-732*   24-747*   24-779*   24-794*   24-839*   24-857*   24-894*   24-974*
         24-985*   24-:90*   24-=96*   24->62*   24-?72*   24-@52*   24-A35*   24-A98*   24-B77*   24-C26*   24-C82*   24-E64*   24-F16*
BADDAT   13-35#    15-857*   15-858*   15-879*   15-880*   16-110    16-125    16-164    16-221    24-47*    24-81*    24-112*   24-145*   24-181*
         24-223*   24-275*   24-311*   24-345*   24-379*   24-413*   24-477*   24-496*   24-515*   24-574*   24-585*   24-634*   24-644*   24-691*
         24-701*   24-742*   24-751*   24-789*   24-798*   24-852*   24-862*   24-910*   24-933*   24-980*   24-990*   24-:95*   24-;18*   24-?94*
         24-@09*   24-@92*   24-A57*   24-B38*   24-B82*   24-C86*
BCC      12-140#   15-883    15-885    15-891
BCCCHK   12-324#   15-881
BIT0     12-20#    12-42     12-54     12-63     12-92     12-104    12-116    12-128    12-140    12-152    12-164    12-176    12-188    12-200
         12-212    12-221    12-233    12-246    12-258    12-269    12-341    15-230    15-271    15-400    15-612    15-653    15-693    15-:61
         19-44
BIT00    12-20     12-20#
BIT01    12-20     12-20#
BIT02    12-20     12-20#
BIT03    12-20     12-20#
BIT04    12-20     12-20#
BIT05    12-20     12-20#
BIT06    12-20     12-20#
BIT07    12-20     12-20#
BIT08    12-20     12-20#
BIT09    12-20     12-20#
BIT1     12-20#    12-41     12-53     12-62     12-81     12-91     12-103    12-115    12-127    12-139    12-151    12-163    12-175    12-187
```

```
         12-199   12-211   12-220   12-232   12-245   12-257   12-268   12-342   15-287   15-596   15-707
BIT10    12-20#   12-279   12-291
BIT11    12-20#   12-278   12-290
BIT12    12-20#
BIT13    12-20#
BIT14    12-20#
BIT15    12-20#   12-322   12-324   12-325   15-368   15-522
BIT2     12-20#   12-40    12-52    12-61    12-80    12-90    12-102   12-114   12-126   12-138   12-150   12-162   12-174   12-186
         12-198   12-210   12-219   12-231   12-244   12-256   12-267
BIT3     12-20#   12-39    12-51    12-60    12-79    12-89    12-101   12-113   12-125   12-137   12-149   12-161   12-173   12-185
         12-197   12-209   12-218   12-230   12-243   12-255
BIT4     12-20#   12-38    12-50    12-78    12-88    12-100   12-112   12-124   12-136   12-148   12-160   12-172   12-184   12-196
         12-208   12-229   12-242   12-254
BIT5     12-20#   12-49    12-70    12-77    12-87    12-99    12-111   12-123   12-135   12-147   12-159   12-171   12-183   12-195
         12-207   12-228   12-241   12-253   12-266
BIT6     12-20#   12-37    12-48    12-69    12-76    12-86    12-98    12-110   12-122   12-134   12-146   12-158   12-170   12-182
         12-194   12-206   12-227   12-240   12-252   12-265
BIT7     12-20#   12-36    12-47    12-59    12-68    12-75    12-97    12-109   12-121   12-133   12-145   12-157   12-169   12-181
         12-193   12-205   12-217   12-226   12-239   12-251   12-264   15-851   15-852
BIT8     12-20#   12-281   12-293
BIT9     12-20#   12-280   12-292
BOE      12-20#
BPOLL    12-69#
BSEL1    13-54#   15-16*   15-18*   15-19*   15-33*   15-34*   15-39*   15-131*  15-133*  15-234*  15-372*  15-373*  15-375*  15-465*
         15-466*  15-471*  15-484*  19-59*   19-60*   19-61    24-134*  24-834*
BSEL2    13-55#   19-61*   19-62*   24-470   24-490   24-509
BSEL4    13-56#   15-69    15-90*   24-173*
C$AU      7-18#   23-10
C$AUTO    7-18#   20-19
C$BRK     7-18#
C$BSEG    7-18#   24-72    24-303   24-337   24-371   24-405   24-680   24-731   24-778   24-838   24-900
C$BSUB    7-18#   24-898   24-924   24-960
C$CEFG    7-18#
C$CLCK    7-18#
C$CLEA    7-18#   21-11
C$CLOS    7-18#
C$CLP1    7-18#
C$CVEC    7-18#
C$DCLN    7-18#
C$DODU    7-18#   20-16
C$DRPT    7-18#
C$DU      7-18#   22-13
C$EDIT    7-18#    8-17
C$ERDF    7-18#   15-277   15-283   15-293   15-299   15-406   15-412   15-570   15-602   15-608   15-618   15-624   15-659   15-665
         15-699   15-705   15-713   15-719   15-865   15-870   15-889   15-895   15-904   15-910   15-919   15-925   15-934   15-940
         15-:72   15-:78   24-24    24-50    24-84    24-115   24-148   24-184   24-226   24-278   24-314   24-348   24-382   24-416
         24-480   24-499   24-518   24-551   24-565   24-577   24-588   24-637   24-647   24-694   24-704   24-745   24-754   24-792
         24-801   24-855   24-865   24-913   24-936   24-983   24-993   24-:98   24-;21   24-<83   24-=26   24-=35   24-=47   24-=66
         24-=74   24-?97   24-@12   24-@95   24-A60   24-B41   24-B85   24-C89
C$ERHR    7-18#
C$ERRO    7-18#
C$ERSF    7-18#
C$ERSO    7-18#
C$ESCA    7-18#   24-227   24-279   24-552   24-566   24-578   24-589   24-638   24-648   24-695   24-746   24-793   24-856   24-984
         24-994   24-:99   24-;22
C$ESEG    7-18#   24-86    24-316   24-350   24-384   24-418   24-706   24-756   24-803   24-867   24-915
C$ESUB    7-18#   24-920   24-938   24-999
```

```
C$ETST   7-18#    24-27    24-52    24-90    24-117   24-150   24-186   24-232   24-286   24-320   24-354   24-388   24-422   24-523
         24-594   24-655   24-710   24-760   24-807   24-872   24-939   24-:00   24-;32   24-;81   24-<27   24-<87   24-=76   24->32
         24-?12   24-?47   24-@27   24-A10   24-A73   24-B56   24-B88   24-D34   24-E35   24-E87   24-F55   24-F87   24-G40   24-G90
         24-H60
C$EXIT   7-18#
C$GETB   7-18#
C$GETW   7-18#
C$GMAN   7-18#
C$GPHR   7-18#    19-51
C$GPLO   7-18#
C$GPRI   7-18#
C$INIT   7-18#    19-74
C$INLP   7-18#
C$MANI   7-18#
C$MEM    7-18#
C$MSG    7-18#    16-102   16-115   16-134   16-153   16-173   16-192   16-210   16-230   16-244
C$OPEN   7-18#
C$PNTB   7-18#    16-101   16-107   16-108   16-109   16-110   16-122   16-123   16-124   16-125   16-141   16-142   16-143   16-144
         16-160   16-161   16-162   16-163   16-164   16-180   16-181   16-182   16-183   16-199   16-200   16-201   16-217   16-218
         16-219   16-220   16-221   16-237   16-238   16-239
C$PNTF   7-18#    22-12    24-456   24-482   24-501   24-520
C$PNTS   7-18#
C$PNTX   7-18#    16-111   16-112   16-113   16-114   16-126   16-127   16-128   16-129   16-130   16-131   16-132   16-133   16-145
         16-146   16-147   16-148   16-149   16-150   16-151   16-152   16-165   16-166   16-167   16-168   16-169   16-170   16-171
         16-172   16-184   16-185   16-186   16-187   16-188   16-189   16-190   16-191   16-202   16-203   16-204   16-205   16-206
         16-207   16-208   16-209   16-222   16-223   16-224   16-225   16-226   16-227   16-228   16-229   16-240   16-241   16-242
         16-243
C$QIO    7-18#
C$RDBU   7-18#
C$REFG   7-18#    19-25    19-28    19-31    19-34
C$RESE   7-18#     7-18#   22-10    24-133
C$REVI   7-18#     8-17
C$RFLA   7-18#
C$RPT    7-18#    17-11
C$SEFG   7-18#
C$SPRI   7-18#    20-9     24-16
C$SVEC   7-18#
C$TPRI   7-18#
C32BCC   12-253#  12-259   24-847
C32ENB   12-256#
CARR     12-152#
CHPCHK   12-322#  24-;59   24-<08   24-<59   24-=17   24-=58   24->02   24->55   24-?39   24-E69   24-G21   24-H35
CHPTYP   13-44#   15-369   15-479*  15-483*  15-523   19-13*   24-C43   24-C65
CKDATA   15-842#  24-D85   24-E14   24-F80   24-F83   24-G24   24-G27   24-G30   24-G33   24-G36   24-G73   24-G78   24-G86   24-H42
         24-H45   24-H48   24-H51
CKTBIT   15-:57#  24-?41   24-G22   24-H36
CRC1     12-98#   24-<53   24-A31   24-A94   24-C13   24-E57   24-F09   24-F74   24-G12   24-G63   24-H19
CRC2     12-97#   24-<53   24-A31   24-A94   24-C13   24-E57   24-F09   24-F74   24-G12   24-G63   24-H19
CRCCHK   12-325#  15-;12
CRCTY0   12-246#
CRCTY1   12-245#
CRCTY2   12-244#
CS       12-150#
DDC      12-240#
DDCMP    12-104#  12-176#  24-;56   24-<53   24-=14   24-=55   24-=95   24-?30   24-?68   24-A31   24-D72   24-E57   24-F74   24-G63
         24-H19
DEVMAP   13-46#   19-40*   19-55*
```

```
DEVPTR   13-47#    19-44*    19-53*    19-55     19-56*
DFPTBL   10-9#
DH1      16-84#    16-109    16-144    16-201    16-220    16-239
DH2      16-85#    16-111    16-126    16-145    16-165    16-184    16-202    16-222    16-240
DH3      16-86#    16-111    16-126    16-145    16-165    16-184    16-202    16-222    16-240
DH4      16-87#    16-113    16-128    16-147    16-167    16-186    16-204    16-224    16-242
DH5      15-229    16-88#
DH6      15-232    16-89#
DH7      16-90#    16-130    16-149    16-169    16-188    16-206    16-226
DH8      16-91#    16-130    16-149    16-169    16-188    16-206    16-226
DH9      16-92#    16-132    16-151    16-171    16-190    16-208    16-228
DIAGMC    7-18      7-18
DISILO   13-43#    15-155    15-192    15-206    19-12*    24-B74*   24-B86*   24-C19*   24-D32*
DISSI    12-87#    12-159#   24-B72    24-B74    24-C17    24-C19    24-D24
DTR      12-76#    12-146#
E$END     7-18#
E$LOAD    7-18#     8-17
EBLK     12-139#   15-898    15-900    15-906
EF.CON   12-20#    19-34
EF.NEW   12-20#    19-31
EF.PWR   12-20#
EF.RES   12-20#    19-28
EF.STA   12-20#    19-25
EM1      16-25#    24-24
EM10     15-299    16-34#
EM11     15-406    16-35#
EM12     15-412    16-36#
EM13     16-37#    24-=47
EM14     16-38#    24-=35    24-=66
EM15     16-39#    24-=74
EM16     16-40#    24-=26
EM17     15-602    16-41#
EM18     15-608    16-42#
EM19     15-618    16-43#
EM2      16-26#    24-50     24-115    24-226    24-577    24-588    24-936
EM20     15-624    16-44#
EM21     15-659    16-45#
EM22     15-665    16-46#
EM23     16-47#
EM24     16-48#
EM25     16-49#
EM26     16-50#    24-?97    24-@95    24-A60    24-B41    24-C89
EM27     16-51#    24-@12
EM28     15-705    16-52#
EM29     15-699    16-53#
EM3      16-27#    24-84     24-184    24-278    24-314    24-348    24-382    24-416    24-637    24-647    24-694    24-704    24-745    24-754
         24-792    24-801    24-855    24-865    24-913    24-983    24-993    24-:98    24-;21    24-B85
EM30     15-719    16-54#
EM31     15-713    16-55#
EM32     15-:72    16-56#
EM33     15-:78    16-57#
EM34     15-870    16-58#
EM35     15-889    16-59#
EM36     15-895    16-60#
EM37     15-904    16-61#
EM38     15-910    16-62#
EM39     15-919    16-63#
```

```
EM4      16-28#   24-148
EM40     15-925   16-64#
EM41     15-934   16-65#
EM42     15-940   16-66#
EM43     16-67#
EM44     16-68#
EM45     16-69#
EM46     16-70#
EM47     16-71#
EM48     16-72#
EM49     16-73#
EM5      16-29#   24-480   24-518
EM50     16-74#
EM51     16-75#
EM52     16-76#   24-551
EM53     16-77#   24-565
EM54     15-865   16-78#
EM6      16-30#   24-499
EM60     16-79#   24-<83
EM65     15-570   16-80#
EM7      15-277   16-31#
EM8      15-283   16-32#
EM9      15-293   16-33#
ENAX     12-90#   12-162#  15-154   15-205
ENDIT    19-35    19-73#
ENDPAT   13-452#
ENDTRN   15-552#  24-;78   24-<24   24-<84   24->30   24-?10
EOM      12-62#
ERR1     16-100#  24-24
ERR2     16-106#  24-50    24-84    24-115   24-148   24-184   24-226   24-278   24-314   24-348   24-382   24-416   24-480   24-499
         24-518
ERR3     16-121#  24-577   24-588   24-637   24-647   24-694   24-704   24-745   24-754   24-792   24-801   24-855   24-865   24-936
         24-983   24-993   24-:98   24-;21   24-?97   24-@12   24-@95   24-A60   24-B41   24-B85   24-C89
ERR4     15-277   15-283   15-293   15-299   15-406   15-412   15-570   15-602   15-608   15-618   15-624   15-659   15-665   15-865
         15-:72   15-:78   16-140#
ERR5     16-159#  24-913
ERR6     15-699   15-705   15-713   15-719   16-179#
ERR7     16-198#  24-<83   24-=26   24-=35   24-=47   24-=66   24-=74
ERR8     15-870   15-889   15-895   15-904   15-910   15-919   15-925   15-934   15-940   16-216#
ERR9     16-236#  24-551   24-565
ERRFLG   13-23#
ERROR1   13-40#   15-150*  15-163*  15-188*  15-214*  19-14*   24-546   24-560
EVL      12-20#
F$AU      7-18#   23-9     23-10
F$AUTO    7-18#   20-7     20-19
F$BGN     7-18#    7-24    16-100   16-106   16-121   16-140   16-159   16-179   16-198   16-216   16-236   17-9     18-8     19-8
         20-7     21-8     22-8     23-9     24-14    24-27    24-39    24-52    24-67    24-72    24-90    24-101   24-117   24-128
         24-150   24-166   24-186   24-203   24-227   24-232   24-247   24-279   24-286   24-298   24-303   24-320   24-332   24-337
         24-354   24-366   24-371   24-388   24-400   24-405   24-422   24-449   24-523   24-539   24-552   24-566   24-578   24-589
         24-594   24-610   24-638   24-648   24-655   24-676   24-680   24-695   24-710   24-727   24-731   24-746   24-760   24-774
         24-778   24-793   24-807   24-832   24-838   24-856   24-872   24-889   24-898   24-898   24-900   24-920   24-924   24-924
         24-938   24-939   24-957   24-960   24-960   24-984   24-994   24-999   24-:00   24-:22   24-:99   24-;22   24-;32   24-;52
         24-;81   24-<01   24-<27   24-<49   24-<87   24-=07   24-=76   24-=91   24->32   24->48   24-?12   24-?27   24-?47   24-?63
         24-@27   24-@43   24-A10   24-A26   24-A73   24-A89   24-B56   24-B69   24-B88   24-C05   24-D34   24-D58   24-E35   24-E53
         24-E87   24-F05   24-F55   24-F70   24-F87   24-G08   24-G40   24-G59   24-G90   24-H13   24-H60   25-14    26-13    26-37
F$CLEA    7-18#   21-8     21-11
F$DU      7-18#   22-8     22-13
```

F$END	7-18	7-18	7-18	7-18	7-18	7-18	7-18	7-18	7-18	7-18	7-18	7-18	7-18	7-18
	7-18	7-18	7-18#	7-24	16-102	16-115	16-134	16-153	16-173	16-192	16-210	16-230	16-244	17-11
	19-74	20-19	21-11	22-13	23-10	24-14	24-14	24-14	24-27	24-27	24-39	24-39	24-39	24-52
	24-52	24-67	24-67	24-67	24-86	24-90	24-90	24-101	24-101	24-101	24-117	24-117	24-128	24-128
	24-128	24-150	24-150	24-166	24-166	24-166	24-186	24-186	24-203	24-203	24-203	24-227	24-232	24-232
	24-247	24-247	24-247	24-279	24-286	24-286	24-298	24-298	24-298	24-316	24-320	24-320	24-332	24-332
	24-332	24-350	24-354	24-354	24-366	24-366	24-366	24-384	24-388	24-388	24-400	24-400	24-400	24-418
	24-422	24-422	24-449	24-449	24-449	24-523	24-523	24-539	24-539	24-539	24-552	24-566	24-578	24-589
	24-594	24-594	24-610	24-610	24-610	24-638	24-648	24-655	24-655	24-676	24-676	24-676	24-695	24-706
	24-710	24-710	24-727	24-727	24-727	24-746	24-756	24-760	24-760	24-774	24-774	24-774	24-793	24-803
	24-807	24-807	24-832	24-832	24-832	24-856	24-867	24-872	24-872	24-889	24-889	24-889	24-898	24-898
	24-915	24-920	24-920	24-924	24-924	24-938	24-938	24-939	24-939	24-957	24-957	24-957	24-960	24-960
	24-984	24-994	24-999	24-999	24-:00	24-:00	24-:22	24-:22	24-:22	24-:99	24-;22	24-;32	24-;32	24-;52
	24-;52	24-;52	24-;81	24-;81	24-<01	24-<01	24-<01	24-<27	24-<27	24-<49	24-<49	24-<49	24-<87	24-<87
	24-=07	24-=07	24-=07	24-=76	24-=76	24-=91	24-=91	24-=91	24->32	24->32	24->48	24->48	24->48	24-?12
	24-?12	24-?27	24-?27	24-?27	24-?47	24-?47	24-?63	24-?63	24-?63	24-@27	24-@27	24-@43	24-@43	24-@43
	24-A10	24-A10	24-A26	24-A26	24-A26	24-A73	24-A73	24-A89	24-A89	24-A89	24-B56	24-B56	24-B69	24-B69
	24-B69	24-B88	24-B88	24-C05	24-C05	24-C05	24-D34	24-D34	24-D58	24-D58	24-D58	24-E35	24-E35	24-E53
	24-E53	24-E53	24-E87	24-E87	24-F05	24-F05	24-F05	24-F55	24-F55	24-F70	24-F70	24-F70	24-F87	24-F87
	24-G08	24-G08	24-G08	24-G40	24-G40	24-G59	24-G59	24-G59	24-G90	24-G90	24-H13	24-H13	24-H13	24-H60
	24-H60	25-21	26-16	26-37										
F$HARD	7-18#	25-14	25-21											
F$HW	7-18#	10-9	10-23											
F$INIT	7-18#	19-8	19-74											
F$JMP	7-18#													
F$MOD	7-18#	7-24	26-37											
F$MSG	7-18#	16-100	16-102	16-106	16-115	16-121	16-134	16-140	16-153	16-159	16-173	16-179	16-192	16-198
	16-210	16-216	16-230	16-236	16-244									
F$PROT	7-18#	18-8	18-12											
F$PWR	7-18#													
F$RPT	7-18#	17-9	17-11											
F$SEG	7-18#	24-72	24-86	24-303	24-316	24-337	24-350	24-371	24-384	24-405	24-418	24-680	24-706	24-731
	24-756	24-778	24-803	24-838	24-867	24-900	24-915							
F$SOFT	7-18#	26-13	26-16											
F$SRV	7-18#													
F$SUB	7-18#	24-898	24-920	24-924	24-938	24-960	24-999							
F$SW	7-18#	11-8	11-11											
F$TEST	7-18#	24-14	24-27	24-39	24-52	24-67	24-90	24-101	24-117	24-128	24-150	24-166	24-186	24-203
	24-232	24-247	24-286	24-298	24-320	24-332	24-354	24-366	24-388	24-400	24-422	24-449	24-523	24-539
	24-594	24-610	24-655	24-676	24-710	24-727	24-760	24-774	24-807	24-832	24-872	24-889	24-939	24-957
	24-:00	24-:22	24-;32	24-;52	24-;81	24-<01	24-<27	24-<49	24-<87	24-=07	24-=76	24-=91	24->32	24->48
	24-?12	24-?27	24-?47	24-?63	24-@27	24-@43	24-A10	24-A26	24-A73	24-A89	24-B56	24-B69	24-B88	24-C05
	24-D34	24-D58	24-E35	24-E53	24-E87	24-F05	24-F55	24-F70	24-F87	24-G08	24-G40	24-G59	24-G90	24-H13
	24-H60													
FMT1	16-9#	16-101	16-107	16-122	16-142	16-160	16-181	16-199	16-218	16-237				
FMT10	16-18#	16-141	16-180	16-217										
FMT11	16-19#	16-161												
FMT19	16-20#	24-456												
FMT2	16-10#	16-108	16-123	16-143	16-162	16-182	16-200	16-219	16-238					
FMT24	16-21#	24-482	24-501	24-520										
FMT27	22-12	22-15#												
FMT3	16-11#	16-110	16-125	16-164	16-221									
FMT4	16-12#	16-111	16-126	16-130	16-145	16-149	16-165	16-169	16-184	16-188	16-202	16-206	16-222	16-226
	16-240													
FMT5	16-13#	16-112	16-127	16-131	16-146	16-150	16-166	16-170	16-185	16-189	16-203	16-207	16-223	16-227
	16-241													
FMT6	16-14#	16-114	16-129	16-133	16-148	16-152	16-168	16-172	16-187	16-191	16-205	16-209	16-225	16-229

```
          16-243
FMT7      16-15#    16-109    16-144    16-201    16-220    16-239
FMT8      16-16#    16-124    16-163    16-183
FMT9      16-17#    16-113    16-128    16-132    16-147    16-151    16-167    16-171    16-186    16-190    16-204    16-208    16-224    16-228
          16-242
FRSTIM    13-37#    19-16     19-23*
G$CNTO    7-18#
G$DELM    7-18#
G$DISP    7-18#
G$EXCP    7-18#
G$HILI    7-18#
G$LOLI    7-18#
G$NO      7-18#
G$OFFS    7-18#     25-16     25-17     25-18     25-19
G$OFSI    7-18#     25-16     25-17     25-18     25-19
G$PRMA    7-18#     25-16     25-17
G$PRMD    7-18#     25-18     25-19
G$PRML    7-18#
G$RADA    7-18#
G$RADB    7-18#
G$RADD    7-18#
G$RADL    7-18#
G$RADO    7-18#     25-16     25-17     25-18     25-19
G$XFER    7-18#
G$YES     7-18#     25-16     25-17     25-18     25-19
GETALL    15-227#   15-275    15-281    15-291    15-297    15-404    15-410    15-568    15-600    15-606    15-616    15-622    15-657    15-663
          15-697    15-703    15-711    15-717    15-863    15-868    15-887    15-893    15-902    15-908    15-917    15-923    15-932    15-938
          15-:70    15-:76    24-575    24-586    24-635    24-645    24-692    24-702    24-743    24-752    24-790    24-799    24-853    24-863
          24-911    24-934    24-981    24-991    24-:96    24-;19    24-<81    24-=24    24-=33    24-=45    24-=64    24-=72    24-?95    24-@10
          24-@93    24-A58    24-B39    24-B83    24-C87
GETPRM    19-36     19-47#    19-54
GETREG    15-104#   15-233    24-48     24-82     24-113    24-146    24-182    24-224    24-276    24-312    24-346    24-380    24-414    24-478
          24-497    24-516    24-549    24-563
GOAH      12-60#
GOODAT    13-34#    15-855*   15-856*   15-877*   15-878*   16-110    16-125    16-164    16-221    24-45*    24-46*    24-80*    24-110*   24-111*
          24-143*   24-144*   24-179*   24-180*   24-221*   24-222*   24-273*   24-274*   24-310*   24-344*   24-378*   24-412*   24-476*   24-495*
          24-514*   24-572*   24-573*   24-583*   24-584*   24-632*   24-633*   24-642*   24-643*   24-690*   24-699*   24-700*   24-741*   24-750*
          24-788*   24-797*   24-850*   24-851*   24-860*   24-861*   24-908*   24-909*   24-931*   24-932*   24-978*   24-979*   24-988*   24-989*
          24-:94*   24-;17*   24-?93*   24-@08*   24-@18*   24-@91*   24-A56*   24-B37*   24-B81*   24-C85*   24-D05*   24-D21*
HDX       12-78#    12-148#
HELP      7-4#      8-9       8-19      9-10      14-24
HOE       12-20#
I$AU      7-18#     23-9#     23-10#
I$AUTO    7-18#     20-7#     20-19#
I$CLN     7-18#     21-8#     21-11#
I$DU      7-18#     22-8#     22-13#
I$HRD     25-14#    25-21#
I$INIT    7-18#     19-8#     19-74#
I$MOD     7-18#     7-24      7-24#     26-37     26-37#
I$MSG     7-18#     16-100#   16-102#   16-106#   16-115#   16-121#   16-134#   16-140#   16-153#   16-159#   16-173#   16-179#   16-192#   16-198#
          16-210#   16-216#   16-230#   16-236#   16-244#
I$PROT    7-18#     18-8#
I$PTAB    7-18#
I$PWR     7-18#
I$RPT     7-18#     17-9#     17-11#
I$SEG     7-18#     24-14     24-39     24-67     24-72#    24-86#    24-101    24-128    24-166    24-203    24-247    24-298    24-303#   24-316#
          24-332    24-337#   24-350#   24-366    24-371#   24-384#   24-400    24-405#   24-418#   24-449    24-539    24-610    24-676    24-680#
```

```
         24-695   24-706#  24-727   24-731#  24-746   24-756#  24-774   24-778#  24-793   24-803#  24-832   24-838#  24-856   24-867#
         24-889   24-898   24-900#  24-915#  24-924   24-957   24-960   24-:22   24-;52   24-<01   24-<49   24-=07   24-=91   24->48
         24-?27   24-?63   24-@43   24-A26   24-A89   24-B69   24-C05   24-D58   24-E53   24-F05   24-F70   24-G08   24-G59   24-H13
I$SETU    7-18#
I$SFT    26-13#   26-16#
I$SRV     7-18#
I$SUB     7-18#   24-14    24-39    24-67    24-101   24-128   24-166   24-203   24-247   24-298   24-332   24-366   24-400   24-449
         24-539   24-610   24-676   24-727   24-774   24-832   24-889   24-898   24-898#  24-920   24-920#  24-920#  24-924   24-924#
         24-938   24-938#  24-938#  24-957   24-960   24-960#  24-984   24-994   24-999   24-999#  24-999#  24-:22   24-;52   24-<01
         24-<49   24-=07   24-=91   24->48   24-?27   24-?63   24-@43   24-A26   24-A89   24-B69   24-C05   24-D58   24-E53   24-F05
         24-F70   24-G08   24-G59   24-H13
I$TST     7-18#   24-14    24-14#   24-27    24-27#   24-27#   24-39    24-39#   24-52    24-52#   24-52#   24-67    24-67#   24-90
         24-90#   24-90#   24-101   24-101#  24-117   24-117#  24-117#  24-128   24-128#  24-150   24-150#  24-150#  24-166   24-166#
         24-186   24-186#  24-186#  24-203   24-203#  24-227   24-232   24-232#  24-232#  24-247   24-247#  24-279   24-286   24-286#
         24-286#  24-298   24-298#  24-320   24-320#  24-320#  24-332   24-332#  24-354   24-354#  24-354#  24-366   24-366#  24-388
         24-388#  24-388#  24-400   24-400#  24-422   24-422#  24-422#  24-449   24-449#  24-523   24-523#  24-523#  24-539   24-539#
         24-552   24-566   24-578   24-589   24-594   24-594#  24-594#  24-610   24-610#  24-638   24-648   24-655   24-655#  24-655#
         24-676   24-676#  24-710   24-710#  24-710#  24-727   24-727#  24-760   24-760#  24-760#  24-774   24-774#  24-807   24-807#
         24-807#  24-832   24-832#  24-872   24-872#  24-872#  24-889   24-889#  24-898   24-924   24-939   24-939#  24-939#  24-957
         24-957#  24-960   24-:00   24-:00#  24-:00#  24-:22   24-:22#  24-:99   24-;22   24-;32   24-;32#  24-;32#  24-;52   24-;52#
         24-;81   24-;81#  24-;81#  24-<01   24-<01#  24-<27   24-<27#  24-<27#  24-<49   24-<49#  24-<87   24-<87#  24-<87#  24-=07
         24-=07#  24-=76   24-=76#  24-=76#  24-=91   24-=91#  24->32   24->32#  24->32#  24->48   24->48#  24-?12   24-?12#  24-?12#
         24-?27   24-?27#  24-?47   24-?47#  24-?47#  24-?63   24-?63#  24-@27   24-@27#  24-@27#  24-@43   24-@43#  24-A10   24-A10#
         24-A10#  24-A26   24-A26#  24-A73   24-A73#  24-A73#  24-A89   24-A89#  24-B56   24-B56#  24-B56#  24-B69   24-B69#  24-B88
         24-B88#  24-B88#  24-C05   24-C05#  24-D34   24-D34#  24-D34#  24-D58   24-D58#  24-E35   24-E35#  24-E35#  24-E53   24-E53#
         24-E87   24-E87#  24-E87#  24-F05   24-F05#  24-F55   24-F55#  24-F55#  24-F70   24-F70#  24-F87   24-F87#  24-F87#  24-G08
         24-G08#  24-G40   24-G40#  24-G40#  24-G59   24-G59#  24-G90   24-G90#  24-G90#  24-H13   24-H13#  24-H60   24-H60#  24-H60#
I422     12-251#  12-259
IACT     12-134#  15-655   15-661
IACTIV   15-645#  15-772   15-793   24-?83   24-?88   24-@03   24-@16   24-@24   24-@63   24-@72   24-@79   24-@84   24-A01   24-A07
         24-A46   24-A51   24-A66   24-A70   24-B09   24-B18   24-B25   24-B30   24-B47   24-B53   24-C51   24-C76   24-C97   24-D13
IBE      12-20#
IC       12-68#   12-133#  24-A05   24-A68   24-B51
ICIR     12-173#  15-614   15-620
IDL      12-243#
IDLE     12-99#   24-=55   24-=95   24-E57   24-F74   24-G63
IDU      12-20#
IER      12-20#
IERR     12-103#  15-817   15-823   24-?68   24-@48   24-A31   24-A94   24-C13
INITRN   15-436#  24-;54   24-<03   24-<51   24-=12   24-=53   24-=93   24->50   24-?28   24-C07   24-D70   24-E55   24-F07   24-F72
         24-G10   24-G61   24-H17
INTFLG   13-21#
INTGRL   12-255#  12-259
IRDY     12-136#  15-598   15-604   15-783
ISIRDY   15-588#  15-774   15-787   15-791   15-795   15-:33   15-:39   24-C63   24-C72   24-C80   24-C93   24-D01   24-D09   24-D17
         24-D64   24-D89   24-D99   24-E08   24-E20
ISR      12-20#
ISRUN    25-19    25-26#
IXE      12-20#
J$JMP     7-18#
L$ACP     8-17#
L$APT     8-17#
L$AU      8-17    23-9#
L$AUT     8-17#
L$AUTO    8-17    20-7#
L$CCP     8-17#
L$CLEA    8-17    21-8#
```

L$CO	8-17#		
L$DEPO	8-17#		
L$DESC	8-17	14-17#	
L$DESP	8-17#		
L$DEVP	8-17#		
L$DISP	8-17	9-8#	
L$DLY	8-17#		
L$DTP	8-17#		
L$DTYP	8-17#		
L$DU	8-17	22-8#	
L$DUT	8-17#		
L$DVTY	8-17	14-12#	
L$EF	8-17#		
L$ENVI	8-17#		
L$ETP	8-17#		
L$EXP1	8-17#		
L$EXP4	8-17#		
L$EXP5	8-17#		
L$HARD	8-17	25-14	25-14#
L$HIME	8-17#		
L$HPCP	8-17#		
L$HPTP	8-17#		
L$HW	8-17	10-9	10-9#
L$ICP	8-17#		
L$INIT	8-17	19-8#	
L$LADP	8-17#		
L$LAST	8-17	26-39#	
L$LOAD	8-17#		
L$LUN	8-17#		
L$MREV	8-17#		
L$NAME	8-17#		
L$PRIO	8-17#		
L$PROT	8-17	18-8#	
L$PRT	8-17#		
L$REPP	8-17#		
L$REV	8-17#		
L$RPT	17-9#		
L$SOFT	26-13	26-13#	
L$SPC	8-17#		
L$SPCP	8-17#		
L$SPTP	8-17#		
L$STA	8-17#		
L$SW	11-8	11-8#	
L$TEST	8-17#		
L$TIML	8-17#		
L$UNIT	8-17#	19-49	
L10000	10-9	10-23#	
L10001	11-8	11-11#	
L10002	16-102#		
L10003	16-115#		
L10004	16-134#		
L10005	16-153#		
L10006	16-173#		
L10007	16-192#		
L10010	16-210#		
L10011	16-230#		
L10012	16-244#		

```
L10013   17-11#
L10015   19-74#
L10016   20-19#
L10017   21-11#
L10020   22-13#
L10021   23-10#
L10022   24-27#
L10023   24-52#
L10024   24-90#
L10025   24-117#
L10026   24-150#
L10027   24-186#
L10030   24-227    24-232#
L10031   24-279    24-286#
L10032   24-320#
L10033   24-354#
L10034   24-388#
L10035   24-422#
L10036   24-523#
L10037   24-552    24-566    24-578    24-589    24-594#
L10040   24-638    24-648    24-655#
L10041   24-710#
L10042   24-760#
L10043   24-807#
L10044   24-872#
L10045   24-939#
L10046   24-920#
L10047   24-938#
L10050   24-:00#
L10051   24-984    24-994    24-999#
L10052   24-:99    24-;22    24-;32#
L10053   24-;81#
L10054   24-<27#
L10055   24-<87#
L10056   24-=76#
L10057   24->32#
L10060   24-?12#
L10061   24-?47#
L10062   24-@27#
L10063   24-A10#
L10064   24-A73#
L10065   24-B56#
L10066   24-B88#
L10067   24-D34#
L10070   24-E35#
L10071   24-E87#
L10072   24-F55#
L10073   24-F87#
L10074   24-G40#
L10075   24-G90#
L10076   24-H60#
L10077   25-14     25-21#
L10100   26-13     26-16#
LDMSG1   15-;02#
LDTXSI   15-343#   15-456    15-457    15-515    15-:06    24-:34    24-:35    24-:59    24-;07    24-=38    24-?34    24-?77    24-@57    24-A40
         24-B03    24-B76    24-D75    24-D97    24-E27    24-E31    24-E60    24-F12    24-F76    24-G17    24-G18    24-G66    24-H22    24-H29
         24-H30
```

```
LOADAT  13-36#   16-161   24-906*  24-907*
LODMSG  15-999#  15-;04   24-G13   24-H25
LOE     12-20#
LOGDEV  13-17#   19-42*   19-48*   19-49    19-51    20-16    22-12
LOOPIN  15-130#  24-463
LOT     12-20#
LU1MOD   7-24#
LULOOP  12-39#   15-39    15-234   15-372   15-465   24-834
LULP    12-70#   12-135#  24-?73   24-@53   24-A36   24-A99   24-C21
LUR10   12-301#  15-106   16-112   16-127   16-146   16-166   16-185   16-203   16-223   16-241
LUR11   12-302#  16-112   16-127   16-146   16-166   16-185   16-203   16-223   16-241
LUR12   12-303#  16-112   16-127   16-146   16-166   16-185   16-203   16-223   16-241
LUR13   12-304#  16-112   16-127   16-146   16-166   16-185   16-203   16-223   16-241
LUR14   12-305#  16-114   16-129   16-148   16-168   16-187   16-205   16-225   16-243
LUR15   12-306#  16-114   16-129   16-148   16-168   16-187   16-205   16-225   16-243
LUR16   12-307#  16-114   16-129   16-148   16-168   16-187   16-205   16-225   16-243
LUR17   12-308#  16-114   16-129   16-148   16-168   16-187   16-205   16-225   16-243
LUREG   12-301   12-302   12-303   12-304   12-305   12-306   12-307   12-308   12-309   12-310   12-311   12-312   12-313   12-314
        12-315   12-316   13-11#
LUSWI1  13-62#
LUSWI2  13-63#
LUSWI3  13-64#
MAINT1  12-79#
MAINT2  12-80#
MCLK    12-175#  24-471   24-476   24-491   24-510   24-514
MCLR    12-37#   15-33    15-34
MODR    12-149#
MPCSR   13-53#   16-101   16-107   16-122   16-142   16-160   16-181   16-199   16-218   16-237   19-58*   20-12    24-19
MPIVEC  13-59#   19-67*
MPOVEC  13-60#   19-68*   19-69*
MPRIOR  13-61#   19-70*
MSG1    13-461#  15-;05   15-;07   15-;10   24-?31   24-?35   24-?44   24-?75   24-?78   24-?87   24-@01   24-@55   24-@58   24-@69
        24-@70   24-@77   24-@99   24-A38   24-A41   24-A50   24-A64   24-B01   24-B04   24-B15   24-B16   24-B23   24-B45   24-C23
        24-G14
MSG4    13-473#  24-H26
MSTCLR  15-30#   15-441   24-41    24-68    24-102   24-106   24-129   24-167   24-204   24-212   24-248   24-299   24-333   24-367
        24-401   24-450   24-522   24-540   24-556   24-611   24-677   24-728   24-775   24-833   24-890   24-927   24-961   24-:27
        24-;80   24-<26   24-<86   24-=68   24-=75   24-?46   24-?65   24-@26   24-@45   24-A09   24-A28   24-A72   24-A91   24-B55
        24-B70   24-B87   24-C10   24-D33   24-D63   24-E34   24-E86   24-F54   24-F86   24-G39   24-G89   24-H59
MVIOX   12-334#  15-65    24-461
MVIXO   12-335#  15-87
MVIXOX  12-336#  24-171
NEWST   19-32    19-41#   19-50
O$APTS   7-18#    8-17
O$AU     7-18#    8-7#     8-17
O$BGNR   7-18#    8-17
O$BGNS   7-18#    8-17
O$DU     7-18#    8-7#     8-17
O$ERRT   7-18#    8-17
O$GNSW   7-18#    8-17
O$POIN   7-18#    8-7      8-7#     8-7#     8-17
O$SETU   7-18#    8-17    26-39
OACT    12-122#  15-402   15-408   15-469
OACTIV  15-392#  15-444   15-461   15-476   15-526   15-562   24-C32   24-C36   24-C57   24-C74   24-C95   24-D11   24-D30   24-H54
OC      12-59#   12-121#  15-557   24-D27   24-G82
OCOR    12-172#  15-289   15-295   15-481   24-:47
OP      12-257#  12-259
```

```
ORDY     12-124#  15-273   15-279   24-269   24-270   24-:79
OSIRDY   15-263#  15-442   15-459   15-486   15-530   15-536   15-560   24-962   24-972   24-:28   24-:37   24-:52   24-:66   24-:84
         24-;09   24-;25   24-;28   24-C24   24-C30   24-C46   24-C49   24-C59
OVRR     12-137#  15-928   15-930   15-936
PATA     13-123#  24-70
PATB     13-146#  24-88    24-258
PATC     13-157#  24-301
PATCH    26-28#
PATD     13-163#  24-318   24-335
PATE     13-169#  24-352   24-369
PATF     13-178#  24-386   24-403
PATG     13-184#  24-206   24-420
PATH     13-195#  24-210   24-542
PATI     13-206#  24-554   24-558   24-612   24-697   24-700   24-929   24-932
PATJ     13-217#  24-592   24-615   24-619   24-620
PATK     13-228#  24-678   24-776
PATL     13-302#  24-708   24-729   24-805
PATM     13-311#  24-43    24-46    24-108   24-111   24-141   24-144   24-177   24-180   24-214   24-250   24-758
PATN     13-322#  24-230   24-958
PATO     13-339#  24-891   24-996
PATP     13-356#  24-892   24-918
PATU     13-373#  24-836
PATV     13-413#  24-835   24-870
PNT      12-20#
POLL     12-75#
PRI      12-20#
PRI00    12-20#
PRI01    12-20#
PRI02    12-20#
PRI03    12-20#
PRI04    12-20#
PRI05    12-20#
PRI06    12-20#
PRI07    12-20#   20-9     20-11    24-16    24-18
PRIOR    13-19#
PRIRTY   25-18    25-25#
PSTACK   13-18#   15-301   15-414   15-626   15-667   15-721   15-949   15-:84   19-10*
R14NRW   13-109#  24-74    24-77
RAB      12-138#  15-913   15-915   15-921
RABT     12-198#
RAX15    13-28#   15-167*  15-168*  15-238   24-569*  24-570   24-574   24-629*  24-630   24-634   24-688   24-691   24-739   24-742
         24-786   24-789   24-846*  24-847*  24-848   24-852   24-976   24-980   24-:92   24-:95   24-;15   24-;18   24-?91   24-?94
         24-@06   24-@09   24-@20   24-@89   24-@92   24-A54   24-A57   24-B35   24-B38   24-B79   24-B82   24-C83   24-C86   24-D03
         24-D19
RAX16    13-29#   15-171*  15-172*  15-240   15-695   15-701   15-709   15-715   24-581   24-585   24-640   24-644   24-697   24-701
         24-748   24-751   24-795   24-798   24-858   24-862   24-904   24-910   24-929   24-933   24-986   24-990
RCV1ST   15-761#  24-D79   24-E72   24-F18   24-G70   24-H40
RCVBUF   13-485#  15-;08
RDALL    12-102#
RDAX     12-88#   12-160#  15-154
RDRXSI   15-737#  15-848   15-:28   24-F79
READAX   15-148#  15-237   15-692   24-559   24-626   24-687   24-738   24-785   24-843   24-903   24-928   24-975   24-:91   24-;14
         24-?90   24-@05   24-@19   24-@88   24-A53   24-B34   24-B78
READLU   15-57#   15-108   15-158   15-166   15-170   15-209   15-270   15-286   15-399   15-468   15-480   15-565   15-595   15-611
         15-652   15-739   15-743   15-782   15-860   15-:65   24-42    24-76    24-107   24-139   24-176   24-217   24-265   24-306
         24-340   24-374   24-408   24-:46   24-:78   24-<78   24-=21   24-=30   24-=42   24-=61   24-=69
READY    12-157#  15-159   15-210
```

```
REDBYT  13-26#   15-69*   15-70*   15-109   15-159   15-167   15-171   15-210   15-273   15-279   15-289   15-295   15-402   15-408
        15-469   15-481   15-566   15-598   15-604   15-614   15-620   15-655   15-661   15-740   15-744   15-783   15-861   15-:68
        15-:74   24-43    24-47    24-77*   24-78    24-81    24-108   24-112   24-140*  24-141   24-145   24-177   24-181   24-218*
        24-219   24-223   24-266*  24-269*  24-271   24-275   24-307*  24-308   24-311   24-341*  24-342   24-345   24-375*  24-376
        24-379   24-409*  24-410   24-413   24-470*  24-471   24-477   24-490*  24-491   24-496   24-509*  24-510   24-515   24-:47
        24-:79   24-<79   24-=22   24-=31   24-=43   24-=62   24-=70
REDDAT  13-69#   24-249   24-259*  24-260*  24-270*  24-271   24-274   24-613   24-621*  24-622*  24-630   24-633   24-640   24-643
REG0    13-79#   24-469*  24-473*  24-489*  24-493*  24-508*  24-512*
REG1    13-80#
REG2    13-81#
REG3    13-82#
REG4    13-83#
REG5    13-84#
REG6    13-85#
REG7    13-86#
REGNUM  13-32#   15-32    15-40*   15-43*   15-59    15-85    15-105   15-107*  15-111*  15-112   15-114*  15-149   15-151*  15-165*
        15-169*  15-173*  15-187   15-189*  15-194*  15-198*  15-202*  15-215*  15-263   15-269*  15-285*  15-300*  15-306*  15-343
        15-345*  15-348*  15-351*  15-392   15-398*  15-413*  15-419*  15-437   15-452*  15-464*  15-478*  15-490*  15-553   15-556*
        15-564*  15-572*  15-588   15-594*  15-610*  15-625*  15-631*  15-645   15-651*  15-666*  15-672*  15-737   15-738*  15-742*
        15-745*  15-763   15-766*  15-798*  15-814   15-815*  15-826*  15-843   15-859*  15-867*  15-873*  15-948*  15-953*  15-968
        15-973*  15-985*  15-:57   15-:62*  15-:89*  16-109   16-144   16-161   16-201   16-220   16-239   24-40*   24-69*   24-103*
        24-130*  24-168*  24-169   24-205*  24-209*  24-213*  24-228*  24-256*  24-257*  24-263*  24-264*  24-267   24-300*  24-334*
        24-368*  24-402*  24-458*  24-548*  24-562*  24-893*  24-964*  24-967*  24-:43*  24-:74*  24-<54*  24-<77*  24-=20*  24-?71*
        24-@51*  24-A34*  24-A97*  24-B71*  24-C16*  24-C20*  24-D23*  24-D26*  24-G81*
REOM    12-199#  15-709   15-715
RERR    12-193#
RETADR  13-25#   15-302   15-415   15-627   15-668   15-722   15-950   15-:85   24-959*  24-:23*  24-;53*  24-<02*  24-<50*  24-=11*
        24-=92*  24->49*  24-?64*  24-@44*  24-A27*  24-A90*  24-C06*  24-D59*  24-E54*  24-F06*  24-F71*  24-G09*  24-G60*  24-H14*
RING    12-145#
ROMI    12-41#   15-16    15-18    15-19    15-131   15-133
ROMO    12-40#   15-16    15-18    15-19    15-131   15-133
ROR     12-197#
RRDYTO  12-341#  15-150   15-163   24-560
RSEOM   15-685#  15-776   24-@65   24-@74   24-@81   24-@86   24-A03   24-B11   24-B20   24-B27   24-B32   24-B49   24-C53   24-C78
        24-C99   24-D15
RSOM    12-200#  15-695   15-701
RTS     12-147#  15-566   24-<79
RUN     12-36#   15-34    15-133   24-134
RUNINH  13-66#   19-72*   24-452
RX0     12-116#  12-188#
RX1     12-115#  12-187#
RX2     12-114#  12-186#
RX3     12-113#  12-185#
RX4     12-112#  12-184#
RX5     12-111#  12-183#
RX6     12-110#  12-182#
RX7     12-109#  12-181#
RXABT   12-291#
RXBCC   12-293#  15-;12
RXCHAR  15-:25#  24-E76   24-E80   24-E82   24-E84   24-F22   24-F26   24-F30   24-F34   24-F38   24-F42   24-F46   24-F50   24-F52
RXEBL   12-292#
RXLEN0  12-269#  15-849   24-E66   24-E74   24-F24   24-F32   24-F40   24-F48
RXLEN1  12-268#  15-849   24-E70   24-E74   24-F24   24-F28   24-F40   24-F44
RXLEN2  12-267#  15-849   24-E66   24-E70   24-E74   24-F24   24-F28   24-F32   24-F36
RXOVR   12-290#
RXWORD  13-42#   15-740*  15-741*  15-744*  15-852*  15-853   15-858   15-874   15-880   15-885   15-891   15-900   15-906   15-915
        15-921   15-930   15-936
```

```
S$LSYM   7-18#    10-23#   11-11#   16-102#  16-115#  16-134#  16-153#  16-173#  16-192#  16-210#  16-230#  16-244#  17-11#   19-74#
         20-19#   21-11#   22-13#   23-10#   24-27#   24-52#   24-72    24-72    24-72#   24-90#   24-117#  24-150#  24-186#  24-232#
         24-286#  24-303   24-303   24-303#  24-320#  24-337   24-337   24-337#  24-354#  24-371   24-371   24-371#  24-388#  24-405
         24-405   24-405#  24-422#  24-523#  24-594#  24-655#  24-680   24-680   24-680#  24-710#  24-731   24-731   24-731#  24-760#
         24-778   24-778   24-778#  24-807#  24-838   24-838   24-838#  24-872#  24-900   24-900   24-900#  24-920#  24-938#  24-939#
         24-999#  24-:00#  24-;32#  24-;81#  24-<27#  24-<87#  24-=76#  24->32#  24-?12#  24-?47#  24-@27#  24-A10#  24-A73#  24-B56#
         24-B88#  24-D34#  24-E35#  24-E87#  24-F55#  24-F87#  24-G40#  24-G90#  24-H60#  25-21#   26-16#
SAVE4    13-38#   19-18*   19-21    20-17    24-25
SAVE6    13-39#   19-19*   19-22    20-18    24-26
SAVLEN   13-45#   15-45*   15-849   15-982*  19-15*
SCRACH   13-16#
SEC      12-242#
SECA     12-100#
SEL4     13-57#   19-63*   19-64*
SEL6     13-58#   15-17*   15-132*  19-65*   19-66*
SELFR    12-77#
SELSBY   12-81#
SETUP    15-967#  24-?66   24-@46   24-A29   24-A92   24-C11
SFPTBL   11-8#
SIGQ     12-170#
SIGR     12-169#
SOM      12-63#
STALL    15-330#  15-374   15-376   15-467   15-472   15-485
STARES   13-50#   19-38*   19-43*   24-454
STARST   19-26    19-29    19-37#
STBY     12-151#
STEPLU   12-38#   15-373   15-375   15-466   15-471   15-484
STEPMP   12-42#   15-18    15-19
STPCLK   15-15#   15-67    15-91    24-174
STPERR   15-814#
STPLU    15-364#  15-532   15-778   15-821   15-945   15-:35   15-:63   24-:44   24-:76   24-;12   24-<75   24-=18   24-=28   24-=40
         24-=59   24-?38   24-?81   24-?85   24-?99   24-@14   24-@22   24-@61   24-@67   24-@97   24-A44   24-A48   24-A62   24-B07
         24-B13   24-B43   24-C34   24-C55   24-C70   24-C91   24-D07   24-D94   24-E06   24-E22   24-E68   24-F77   24-G20   24-G84
         24-H34
STR      12-241#
STRIP    12-101#  24-;56   24-<53   24-=14   24-=55   24-?30   24-?68   24-A31   24-D72   24-G12   24-H19
SUBRPC   13-20#   15-264   15-265   15-267*  15-268*  15-305*  15-393   15-394   15-396*  15-397*  15-418*  15-439*  15-440*  15-492*
         15-512*  15-513*  15-539*  15-554*  15-555*  15-571*  15-589   15-590   15-592*  15-593*  15-630*  15-646   15-647   15-649*
         15-650*  15-671*  15-686   15-687   15-689*  15-690*  15-725*  15-764*  15-765*  15-801*  15-844*  15-845*  15-955*  15-987*
         15-:26*  15-:27*  15-:42*  15-:59*  15-:60*  15-:90*  16-141   16-180   16-217   19-11*
SVCGBL   7-18#    7-24     7-24     7-24     7-32#    8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     9-8      10-9     10-9     11-8     11-8     14-12    14-17    16-100   16-106   16-121   16-140   16-159   16-179
         16-198   16-216   16-236   17-9     18-8     19-8     20-7     21-8     22-8     23-9     25-14    26-13    26-39    26-39
         26-39    26-39#
SVCINS   7-18#    7-29#    8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
         8-17     8-17     8-17     9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8
```

9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	10-9	10-9	10-9	11-8	11-8	11-8	14-12	14-12
14-12	14-12	14-12	14-12	14-17	14-17	14-17	14-17	14-17	14-17	15-277	15-277	15-277	15-277
15-277	15-277	15-277	15-277	15-277	15-277	15-277	15-277	15-283	15-283	15-283	15-283	15-283	15-283
15-283	15-283	15-283	15-283	15-283	15-283	15-293	15-293	15-293	15-293	15-293	15-293	15-293	15-293
15-293	15-293	15-293	15-293	15-299	15-299	15-299	15-299	15-299	15-299	15-299	15-299	15-299	15-299
15-299	15-299	15-406	15-406	15-406	15-406	15-406	15-406	15-406	15-406	15-406	15-406	15-406	15-406
15-412	15-412	15-412	15-412	15-412	15-412	15-412	15-412	15-412	15-412	15-412	15-412	15-570	15-570
15-570	15-570	15-570	15-570	15-570	15-570	15-570	15-570	15-570	15-570	15-602	15-602	15-602	15-602
15-602	15-602	15-602	15-602	15-602	15-602	15-602	15-602	15-608	15-608	15-608	15-608	15-608	15-608
15-608	15-608	15-608	15-608	15-608	15-608	15-618	15-618	15-618	15-618	15-618	15-618	15-618	15-618
15-618	15-618	15-618	15-618	15-624	15-624	15-624	15-624	15-624	15-624	15-624	15-624	15-624	15-624
15-624	15-624	15-659	15-659	15-659	15-659	15-659	15-659	15-659	15-659	15-659	15-659	15-659	15-659
15-665	15-665	15-665	15-665	15-665	15-665	15-665	15-665	15-665	15-665	15-665	15-665	15-699	15-699
15-699	15-699	15-699	15-699	15-699	15-699	15-699	15-699	15-699	15-699	15-705	15-705	15-705	15-705
15-705	15-705	15-705	15-705	15-705	15-705	15-705	15-705	15-713	15-713	15-713	15-713	15-713	15-713
15-713	15-713	15-713	15-713	15-713	15-713	15-719	15-719	15-719	15-719	15-719	15-719	15-719	15-719
15-719	15-719	15-719	15-719	15-865	15-865	15-865	15-865	15-865	15-865	15-865	15-865	15-865	15-865
15-865	15-865	15-870	15-870	15-870	15-870	15-870	15-870	15-870	15-870	15-870	15-870	15-870	15-870
15-889	15-889	15-889	15-889	15-889	15-889	15-889	15-889	15-889	15-889	15-889	15-889	15-895	15-895
15-895	15-895	15-895	15-895	15-895	15-895	15-895	15-895	15-895	15-895	15-904	15-904	15-904	15-904
15-904	15-904	15-904	15-904	15-904	15-904	15-904	15-904	15-910	15-910	15-910	15-910	15-910	15-910
15-910	15-910	15-910	15-910	15-910	15-910	15-919	15-919	15-919	15-919	15-919	15-919	15-919	15-919
15-919	15-919	15-919	15-919	15-925	15-925	15-925	15-925	15-925	15-925	15-925	15-925	15-925	15-925
15-925	15-925	15-934	15-934	15-934	15-934	15-934	15-934	15-934	15-934	15-934	15-934	15-934	15-934
15-940	15-940	15-940	15-940	15-940	15-940	15-940	15-940	15-940	15-940	15-940	15-940	15-:72	15-:72
15-:72	15-:72	15-:72	15-:72	15-:72	15-:72	15-:72	15-:72	15-:72	15-:72	15-:78	15-:78	15-:78	15-:78
15-:78	15-:78	15-:78	15-:78	15-:78	15-:78	15-:78	15-:78	16-101	16-101	16-101	16-101	16-101	16-101
16-101	16-101	16-101	16-101	16-101	16-101	16-101	16-101	16-101	16-101	16-101	16-101	16-101	16-101
16-101	16-102	16-102	16-102	16-107	16-107	16-107	16-107	16-107	16-107	16-107	16-107	16-107	16-107
16-107	16-107	16-107	16-107	16-107	16-107	16-107	16-107	16-107	16-107	16-107	16-108	16-108	16-108
16-108	16-108	16-108	16-108	16-108	16-108	16-108	16-108	16-108	16-108	16-108	16-108	16-109	16-109
16-109	16-109	16-109	16-109	16-109	16-109	16-109	16-109	16-109	16-109	16-109	16-109	16-109	16-109
16-109	16-109	16-109	16-109	16-109	16-110	16-110	16-110	16-110	16-110	16-110	16-110	16-110	16-110
16-110	16-110	16-110	16-110	16-110	16-110	16-110	16-110	16-110	16-110	16-110	16-110	16-111	16-111
16-111	16-111	16-111	16-111	16-111	16-111	16-111	16-111	16-111	16-111	16-111	16-111	16-111	16-111
16-111	16-111	16-111	16-111	16-111	16-112	16-112	16-112	16-112	16-112	16-112	16-112	16-112	16-112
16-112	16-112	16-112	16-112	16-112	16-112	16-112	16-112	16-112	16-112	16-112	16-112	16-112	16-112
16-112	16-112	16-112	16-112	16-113	16-113	16-113	16-113	16-113	16-113	16-113	16-113	16-113	16-113
16-113	16-113	16-113	16-113	16-113	16-113	16-113	16-113	16-114	16-114	16-114	16-114	16-114	16-114
16-114	16-114	16-114	16-114	16-114	16-114	16-114	16-114	16-114	16-114	16-114	16-114	16-114	16-114
16-114	16-114	16-114	16-114	16-114	16-114	16-114	16-115	16-115	16-115	16-122	16-122	16-122	16-122
16-122	16-122	16-122	16-122	16-122	16-122	16-122	16-122	16-122	16-122	16-122	16-122	16-122	16-122
16-122	16-122	16-122	16-123	16-123	16-123	16-123	16-123	16-123	16-123	16-123	16-123	16-123	16-123
16-123	16-123	16-123	16-123	16-124	16-124	16-124	16-124	16-124	16-124	16-124	16-124	16-124	16-124
16-124	16-124	16-124	16-124	16-124	16-124	16-124	16-124	16-124	16-124	16-124	16-125	16-125	16-125
16-125	16-125	16-125	16-125	16-125	16-125	16-125	16-125	16-125	16-125	16-125	16-125	16-125	16-125
16-125	16-125	16-125	16-125	16-126	16-126	16-126	16-126	16-126	16-126	16-126	16-126	16-126	16-126
16-126	16-126	16-126	16-126	16-126	16-126	16-126	16-126	16-126	16-126	16-126	16-127	16-127	16-127
16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127

16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-128	16-128	16-128	16-128
16-128	16-128	16-128	16-128	16-128	16-128	16-128	16-128	16-128	16-128	16-128	16-128	16-128	16-128
16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129
16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-130
16-130	16-130	16-130	16-130	16-130	16-130	16-130	16-130	16-130	16-130	16-130	16-130	16-130	16-130
16-130	16-130	16-130	16-130	16-130	16-130	16-131	16-131	16-131	16-131	16-131	16-131	16-131	16-131
16-131	16-131	16-131	16-131	16-131	16-131	16-131	16-131	16-131	16-131	16-131	16-131	16-131	16-131
16-131	16-131	16-131	16-131	16-131	16-132	16-132	16-132	16-132	16-132	16-132	16-132	16-132	16-132
16-132	16-132	16-132	16-132	16-132	16-132	16-132	16-132	16-132	16-133	16-133	16-133	16-133	16-133
16-133	16-133	16-133	16-133	16-133	16-133	16-133	16-133	16-133	16-133	16-133	16-133	16-133	16-133
16-133	16-133	16-133	16-133	16-133	16-133	16-133	16-133	16-134	16-134	16-134	16-141	16-141	16-141
16-141	16-141	16-141	16-141	16-141	16-141	16-141	16-141	16-141	16-141	16-141	16-141	16-141	16-141
16-141	16-142	16-142	16-142	16-142	16-142	16-142	16-142	16-142	16-142	16-142	16-142	16-142	16-142
16-142	16-142	16-142	16-142	16-142	16-142	16-142	16-142	16-143	16-143	16-143	16-143	16-143	16-143
16-143	16-143	16-143	16-143	16-143	16-143	16-143	16-143	16-143	16-144	16-144	16-144	16-144	16-144
16-144	16-144	16-144	16-144	16-144	16-144	16-144	16-144	16-144	16-144	16-144	16-144	16-144	16-144
16-144	16-144	16-145	16-145	16-145	16-145	16-145	16-145	16-145	16-145	16-145	16-145	16-145	16-145
16-145	16-145	16-145	16-145	16-145	16-145	16-145	16-145	16-145	16-146	16-146	16-146	16-146	16-146
16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146
16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-147	16-147	16-147	16-147	16-147	16-147
16-147	16-147	16-147	16-147	16-147	16-147	16-147	16-147	16-147	16-147	16-147	16-147	16-148	16-148
16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148
16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-149	16-149	16-149
16-149	16-149	16-149	16-149	16-149	16-149	16-149	16-149	16-149	16-149	16-149	16-149	16-149	16-149
16-149	16-149	16-149	16-149	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150
16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150
16-150	16-150	16-150	16-151	16-151	16-151	16-151	16-151	16-151	16-151	16-151	16-151	16-151	16-151
16-151	16-151	16-151	16-151	16-151	16-151	16-151	16-152	16-152	16-152	16-152	16-152	16-152	16-152
16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152
16-152	16-152	16-152	16-152	16-152	16-152	16-153	16-153	16-153	16-160	16-160	16-160	16-160	16-160
16-160	16-160	16-160	16-160	16-160	16-160	16-160	16-160	16-160	16-160	16-160	16-160	16-160	16-160
16-160	16-160	16-161	16-161	16-161	16-161	16-161	16-161	16-161	16-161	16-161	16-161	16-161	16-161
16-161	16-161	16-161	16-161	16-161	16-161	16-161	16-161	16-161	16-162	16-162	16-162	16-162	16-162
16-162	16-162	16-162	16-162	16-162	16-162	16-162	16-162	16-162	16-162	16-163	16-163	16-163	16-163
16-163	16-163	16-163	16-163	16-163	16-163	16-163	16-163	16-163	16-163	16-163	16-163	16-163	16-163
16-163	16-163	16-163	16-164	16-164	16-164	16-164	16-164	16-164	16-164	16-164	16-164	16-164	16-164
16-164	16-164	16-164	16-164	16-164	16-164	16-164	16-164	16-164	16-164	16-165	16-165	16-165	16-165
16-165	16-165	16-165	16-165	16-165	16-165	16-165	16-165	16-165	16-165	16-165	16-165	16-165	16-165
16-165	16-165	16-165	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166
16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166
16-166	16-166	16-167	16-167	16-167	16-167	16-167	16-167	16-167	16-167	16-167	16-167	16-167	16-167
16-167	16-167	16-167	16-167	16-167	16-167	16-168	16-168	16-168	16-168	16-168	16-168	16-168	16-168
16-168	16-168	16-168	16-168	16-168	16-168	16-168	16-168	16-168	16-168	16-168	16-168	16-168	16-168
16-168	16-168	16-168	16-168	16-168	16-169	16-169	16-169	16-169	16-169	16-169	16-169	16-169	16-169
16-169	16-169	16-169	16-169	16-169	16-169	16-169	16-169	16-169	16-169	16-169	16-169	16-170	16-170
16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170
16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-171	16-171	16-171
16-171	16-171	16-171	16-171	16-171	16-171	16-171	16-171	16-171	16-171	16-171	16-171	16-171	16-171
16-171	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172
16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172
16-173	16-173	16-173	16-180	16-180	16-180	16-180	16-180	16-180	16-180	16-180	16-180	16-180	16-180
16-180	16-180	16-180	16-180	16-180	16-180	16-180	16-181	16-181	16-181	16-181	16-181	16-181	16-181
16-181	16-181	16-181	16-181	16-181	16-181	16-181	16-181	16-181	16-181	16-181	16-181	16-181	16-181
16-182	16-182	16-182	16-182	16-182	16-182	16-182	16-182	16-182	16-182	16-182	16-182	16-182	16-182
16-182	16-183	16-183	16-183	16-183	16-183	16-183	16-183	16-183	16-183	16-183	16-183	16-183	16-183
16-183	16-183	16-183	16-183	16-183	16-183	16-183	16-183	16-184	16-184	16-184	16-184	16-184	16-184
16-184	16-184	16-184	16-184	16-184	16-184	16-184	16-184	16-184	16-184	16-184	16-184	16-184	16-184

16-184	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185
16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185	16-185
16-186	16-186	16-186	16-186	16-186	16-186	16-186	16-186	16-186	16-186	16-186	16-186	16-186	16-186
16-186	16-186	16-186	16-186	16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187
16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187	16-187
16-187	16-187	16-187	16-188	16-188	16-188	16-188	16-188	16-188	16-188	16-188	16-188	16-188	16-188
16-188	16-188	16-188	16-188	16-188	16-188	16-188	16-188	16-188	16-188	16-189	16-189	16-189	16-189
16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-189
16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-189	16-190	16-190	16-190	16-190	16-190
16-190	16-190	16-190	16-190	16-190	16-190	16-190	16-190	16-190	16-190	16-190	16-190	16-190	16-191
16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191
16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-191	16-192	16-192
16-192	16-199	16-199	16-199	16-199	16-199	16-199	16-199	16-199	16-199	16-199	16-199	16-199	16-199
16-199	16-199	16-199	16-199	16-199	16-199	16-199	16-199	16-200	16-200	16-200	16-200	16-200	16-200
16-200	16-200	16-200	16-200	16-200	16-200	16-200	16-200	16-200	16-201	16-201	16-201	16-201	16-201
16-201	16-201	16-201	16-201	16-201	16-201	16-201	16-201	16-201	16-201	16-201	16-201	16-201	16-201
16-201	16-201	16-202	16-202	16-202	16-202	16-202	16-202	16-202	16-202	16-202	16-202	16-202	16-202
16-202	16-202	16-202	16-202	16-202	16-202	16-202	16-202	16-202	16-203	16-203	16-203	16-203	16-203
16-203	16-203	16-203	16-203	16-203	16-203	16-203	16-203	16-203	16-203	16-203	16-203	16-203	16-203
16-203	16-203	16-203	16-203	16-203	16-203	16-203	16-203	16-204	16-204	16-204	16-204	16-204	16-204
16-204	16-204	16-204	16-204	16-204	16-204	16-204	16-204	16-204	16-204	16-204	16-204	16-205	16-205
16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205
16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-205	16-206	16-206	16-206
16-206	16-206	16-206	16-206	16-206	16-206	16-206	16-206	16-206	16-206	16-206	16-206	16-206	16-206
16-206	16-206	16-206	16-206	16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207
16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207	16-207
16-207	16-207	16-207	16-208	16-208	16-208	16-208	16-208	16-208	16-208	16-208	16-208	16-208	16-208
16-208	16-208	16-208	16-208	16-208	16-208	16-208	16-209	16-209	16-209	16-209	16-209	16-209	16-209
16-209	16-209	16-209	16-209	16-209	16-209	16-209	16-209	16-209	16-209	16-209	16-209	16-209	16-209
16-209	16-209	16-209	16-209	16-209	16-209	16-210	16-210	16-210	16-217	16-217	16-217	16-217	16-217
16-217	16-217	16-217	16-217	16-217	16-217	16-217	16-217	16-217	16-217	16-217	16-217	16-217	16-218
16-218	16-218	16-218	16-218	16-218	16-218	16-218	16-218	16-218	16-218	16-218	16-218	16-218	16-218
16-218	16-218	16-218	16-218	16-218	16-218	16-219	16-219	16-219	16-219	16-219	16-219	16-219	16-219
16-219	16-219	16-219	16-219	16-219	16-219	16-219	16-220	16-220	16-220	16-220	16-220	16-220	16-220
16-220	16-220	16-220	16-220	16-220	16-220	16-220	16-220	16-220	16-220	16-220	16-220	16-220	16-220
16-221	16-221	16-221	16-221	16-221	16-221	16-221	16-221	16-221	16-221	16-221	16-221	16-221	16-221
16-221	16-221	16-221	16-221	16-221	16-221	16-221	16-222	16-222	16-222	16-222	16-222	16-222	16-222
16-222	16-222	16-222	16-222	16-222	16-222	16-222	16-222	16-222	16-222	16-222	16-222	16-222	16-222
16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223
16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-224
16-224	16-224	16-224	16-224	16-224	16-224	16-224	16-224	16-224	16-224	16-224	16-224	16-224	16-224
16-224	16-224	16-224	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225
16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225
16-225	16-225	16-226	16-226	16-226	16-226	16-226	16-226	16-226	16-226	16-226	16-226	16-226	16-226
16-226	16-226	16-226	16-226	16-226	16-226	16-226	16-226	16-226	16-227	16-227	16-227	16-227	16-227
16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227
16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-228	16-228	16-228	16-228	16-228	16-228
16-228	16-228	16-228	16-228	16-228	16-228	16-228	16-228	16-228	16-228	16-228	16-228	16-229	16-229
16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229
16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-230	16-230	16-230
16-237	16-237	16-237	16-237	16-237	16-237	16-237	16-237	16-237	16-237	16-237	16-237	16-237	16-237
16-237	16-237	16-237	16-237	16-237	16-237	16-237	16-238	16-238	16-238	16-238	16-238	16-238	16-238
16-238	16-238	16-238	16-238	16-238	16-238	16-238	16-238	16-239	16-239	16-239	16-239	16-239	16-239
16-239	16-239	16-240	16-240	16-240	16-240	16-240	16-240	16-240	16-240	16-240	16-240	16-240	16-240
16-240	16-240	16-240	16-240	16-240	16-240	16-240	16-240	16-241	16-241	16-241	16-241	16-241	16-241
16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241

16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-242	16-242	16-242	16-242	16-242	16-242	16-242
16-242	16-242	16-242	16-242	16-242	16-242	16-242	16-242	16-242	16-242	16-242	16-243	16-243	16-243
16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243
16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-244	16-244	16-244	17-11
17-11	17-11	19-25	19-25	19-25	19-25	19-25	19-25	19-26	19-26	19-26	19-28	19-28	19-28
19-28	19-28	19-28	19-29	19-29	19-29	19-31	19-31	19-31	19-31	19-31	19-31	19-32	19-32
19-32	19-34	19-34	19-34	19-34	19-34	19-34	19-35	19-35	19-35	19-51	19-51	19-51	19-51
19-51	19-51	19-51	19-51	19-51	19-52	19-52	19-52	19-74	19-74	19-74	20-9	20-9	20-9
20-9	20-9	20-9	20-16	20-16	20-16	20-16	20-16	20-16	20-19	20-19	20-19	21-11	21-11
21-11	22-10	22-10	22-10	22-12	22-12	22-12	22-12	22-12	22-12	22-12	22-12	22-12	22-12
22-12	22-12	22-12	22-12	22-12	22-12	22-12	22-12	22-13	22-13	22-13	23-10	23-10	23-10
24-16	24-16	24-16	24-16	24-16	24-16	24-24	24-24	24-24	24-24	24-24	24-24	24-24	24-24
24-24	24-24	24-24	24-24	24-27	24-27	24-27	24-50	24-50	24-50	24-50	24-50	24-50	24-50
24-50	24-50	24-50	24-50	24-50	24-52	24-52	24-52	24-72	24-72	24-72	24-84	24-84	24-84
24-84	24-84	24-84	24-84	24-84	24-84	24-84	24-84	24-84	24-86	24-86	24-86	24-90	24-90
24-90	24-115	24-115	24-115	24-115	24-115	24-115	24-115	24-115	24-115	24-115	24-115	24-115	24-117
24-117	24-117	24-133	24-133	24-133	24-148	24-148	24-148	24-148	24-148	24-148	24-148	24-148	24-148
24-148	24-148	24-148	24-150	24-150	24-150	24-184	24-184	24-184	24-184	24-184	24-184	24-184	24-184
24-184	24-184	24-184	24-184	24-186	24-186	24-186	24-226	24-226	24-226	24-226	24-226	24-226	24-226
24-226	24-226	24-226	24-226	24-226	24-227	24-227	24-227	24-227	24-227	24-227	24-232	24-232	24-232
24-278	24-278	24-278	24-278	24-278	24-278	24-278	24-278	24-278	24-278	24-278	24-278	24-279	24-279
24-279	24-279	24-279	24-279	24-286	24-286	24-286	24-303	24-303	24-303	24-314	24-314	24-314	24-314
24-314	24-314	24-314	24-314	24-314	24-314	24-314	24-314	24-316	24-316	24-316	24-320	24-320	24-320
24-337	24-337	24-337	24-348	24-348	24-348	24-348	24-348	24-348	24-348	24-348	24-348	24-348	24-348
24-348	24-350	24-350	24-350	24-354	24-354	24-354	24-371	24-371	24-371	24-382	24-382	24-382	24-382
24-382	24-382	24-382	24-382	24-382	24-382	24-382	24-382	24-384	24-384	24-384	24-388	24-388	24-388
24-405	24-405	24-405	24-416	24-416	24-416	24-416	24-416	24-416	24-416	24-416	24-416	24-416	24-416
24-416	24-418	24-418	24-418	24-422	24-422	24-422	24-456	24-456	24-456	24-456	24-456	24-456	24-456
24-456	24-456	24-456	24-456	24-456	24-456	24-456	24-456	24-456	24-456	24-456	24-480	24-480	24-480
24-480	24-480	24-480	24-480	24-480	24-480	24-480	24-480	24-480	24-482	24-482	24-482	24-482	24-482
24-482	24-482	24-482	24-482	24-482	24-482	24-482	24-482	24-482	24-482	24-499	24-499	24-499	24-499
24-499	24-499	24-499	24-499	24-499	24-499	24-499	24-499	24-501	24-501	24-501	24-501	24-501	24-501
24-501	24-501	24-501	24-501	24-501	24-501	24-501	24-501	24-501	24-518	24-518	24-518	24-518	24-518
24-518	24-518	24-518	24-518	24-518	24-518	24-518	24-520	24-520	24-520	24-520	24-520	24-520	24-520
24-520	24-520	24-520	24-520	24-520	24-520	24-520	24-520	24-523	24-523	24-523	24-551	24-551	24-551
24-551	24-551	24-551	24-551	24-551	24-551	24-551	24-551	24-551	24-552	24-552	24-552	24-552	24-552
24-552	24-565	24-565	24-565	24-565	24-565	24-565	24-565	24-565	24-565	24-565	24-565	24-565	24-566
24-566	24-566	24-566	24-566	24-566	24-577	24-577	24-577	24-577	24-577	24-577	24-577	24-577	24-577
24-577	24-577	24-577	24-578	24-578	24-578	24-578	24-578	24-578	24-588	24-588	24-588	24-588	24-588
24-588	24-588	24-588	24-588	24-588	24-588	24-588	24-589	24-589	24-589	24-589	24-589	24-589	24-594
24-594	24-594	24-637	24-637	24-637	24-637	24-637	24-637	24-637	24-637	24-637	24-637	24-637	24-637
24-638	24-638	24-638	24-638	24-638	24-638	24-647	24-647	24-647	24-647	24-647	24-647	24-647	24-647
24-647	24-647	24-647	24-647	24-648	24-648	24-648	24-648	24-648	24-648	24-655	24-655	24-655	24-680
24-680	24-680	24-694	24-694	24-694	24-694	24-694	24-694	24-694	24-694	24-694	24-694	24-694	24-694
24-695	24-695	24-695	24-695	24-695	24-695	24-704	24-704	24-704	24-704	24-704	24-704	24-704	24-704
24-704	24-704	24-704	24-704	24-706	24-706	24-706	24-710	24-710	24-710	24-731	24-731	24-731	24-745
24-745	24-745	24-745	24-745	24-745	24-745	24-745	24-745	24-745	24-745	24-745	24-746	24-746	24-746
24-746	24-746	24-746	24-754	24-754	24-754	24-754	24-754	24-754	24-754	24-754	24-754	24-754	24-754
24-754	24-756	24-756	24-756	24-760	24-760	24-760	24-778	24-778	24-778	24-792	24-792	24-792	24-792
24-792	24-792	24-792	24-792	24-792	24-792	24-792	24-792	24-793	24-793	24-793	24-793	24-793	24-793
24-801	24-801	24-801	24-801	24-801	24-801	24-801	24-801	24-801	24-801	24-801	24-801	24-803	24-803
24-803	24-807	24-807	24-807	24-838	24-838	24-838	24-855	24-855	24-855	24-855	24-855	24-855	24-855
24-855	24-855	24-855	24-855	24-855	24-856	24-856	24-856	24-856	24-856	24-856	24-865	24-865	24-865
24-865	24-865	24-865	24-865	24-865	24-865	24-865	24-865	24-865	24-867	24-867	24-867	24-872	24-872
24-872	24-898	24-898	24-898	24-900	24-900	24-900	24-913	24-913	24-913	24-913	24-913	24-913	24-913
24-913	24-913	24-913	24-913	24-913	24-915	24-915	24-915	24-920	24-920	24-920	24-924	24-924	24-924
24-936	24-936	24-936	24-936	24-936	24-936	24-936	24-936	24-936	24-936	24-936	24-936	24-938	24-938

```
        24-938  24-939  24-939  24-939  24-960  24-960  24-960  24-983  24-983  24-983  24-983  24-983  24-983  24-983
        24-983  24-983  24-983  24-983  24-983  24-984  24-984  24-984  24-984  24-984  24-984  24-993  24-993  24-993
        24-993  24-993  24-993  24-993  24-993  24-993  24-993  24-993  24-993  24-994  24-994  24-994  24-994  24-994
        24-994  24-999  24-999  24-999  24-:00  24-:00  24-:00  24-:98  24-:98  24-:98  24-:98  24-:98  24-:98  24-:98
        24-:98  24-:98  24-:98  24-:98  24-:98  24-:99  24-:99  24-:99  24-:99  24-:99  24-:99  24-;21  24-;21  24-;21
        24-;21  24-;21  24-;21  24-;21  24-;21  24-;21  24-;21  24-;21  24-;21  24-;22  24-;22  24-;22  24-;22  24-;22
        24-;22  24-;32  24-;32  24-;32  24-;81  24-;81  24-;81  24-<27  24-<27  24-<27  24-<83  24-<83  24-<83  24-<83
        24-<83  24-<83  24-<83  24-<83  24-<83  24-<83  24-<83  24-<83  24-<87  24-<87  24-<87  24-=26  24-=26  24-=26
        24-=26  24-=26  24-=26  24-=26  24-=26  24-=26  24-=26  24-=26  24-=26  24-=35  24-=35  24-=35  24-=35  24-=35
        24-=35  24-=35  24-=35  24-=35  24-=35  24-=35  24-=35  24-=47  24-=47  24-=47  24-=47  24-=47  24-=47  24-=47
        24-=47  24-=47  24-=47  24-=47  24-=47  24-=66  24-=66  24-=66  24-=66  24-=66  24-=66  24-=66  24-=66  24-=66
        24-=66  24-=66  24-=66  24-=74  24-=74  24-=74  24-=74  24-=74  24-=74  24-=74  24-=74  24-=74  24-=74  24-=74
        24-=74  24-=76  24-=76  24-=76  24->32  24->32  24->32  24-?12  24-?12  24-?12  24-?47  24-?47  24-?47  24-?97
        24-?97  24-?97  24-?97  24-?97  24-?97  24-?97  24-?97  24-?97  24-?97  24-?97  24-?97  24-@12  24-@12  24-@12
        24-@12  24-@12  24-@12  24-@12  24-@12  24-@12  24-@12  24-@12  24-@12  24-@27  24-@27  24-@27  24-@95  24-@95
        24-@95  24-@95  24-@95  24-@95  24-@95  24-@95  24-@95  24-@95  24-@95  24-@95  24-A10  24-A10  24-A10  24-A60
        24-A60  24-A60  24-A60  24-A60  24-A60  24-A60  24-A60  24-A60  24-A60  24-A60  24-A60  24-A73  24-A73  24-A73
        24-B41  24-B41  24-B41  24-B41  24-B41  24-B41  24-B41  24-B41  24-B41  24-B41  24-B41  24-B41  24-B56  24-B56
        24-B56  24-B85  24-B85  24-B85  24-B85  24-B85  24-B85  24-B85  24-B85  24-B85  24-B85  24-B85  24-B85  24-B88
        24-B88  24-B88  24-C89  24-C89  24-C89  24-C89  24-C89  24-C89  24-C89  24-C89  24-C89  24-C89  24-C89  24-C89
        24-D34  24-D34  24-D34  24-E35  24-E35  24-E35  24-E87  24-E87  24-E87  24-F55  24-F55  24-F55  24-F87  24-F87
        24-F87  24-G40  24-G40  24-G40  24-G90  24-G90  24-G90  24-H60  24-H60  24-H60  25-14   25-14   25-14   25-16
        25-16   25-16   25-16   25-16   25-16   25-16   25-16   25-16   25-16   25-16   25-16   25-17   25-17   25-17
        25-17   25-17   25-17   25-17   25-17   25-17   25-17   25-17   25-17   25-18   25-18   25-18   25-18   25-18
        25-18   25-18   25-18   25-18   25-18   25-18   25-18   25-18   25-18   25-18   25-19   25-19   25-19   25-19
        25-19   25-19   25-19   25-19   25-19   25-19   25-19   25-19   25-19   25-19   25-19   25-21   25-21   25-21
        26-13   26-13   26-13   26-16   26-16   26-16   26-39   26-39   26-39   26-39   26-39   26-39   26-39   26-39
        26-39
SVCSUB   7-18#   7-31#  24-898  24-924  24-960
SVCTAG   7-18#   7-33#  10-23   11-11   16-102  16-115  16-134  16-153  16-173  16-192  16-210  16-230  16-244  17-11
        19-74   20-19   21-11   22-13   23-10   24-27   24-52   24-86   24-90   24-117  24-150  24-186  24-232  24-286
        24-316  24-320  24-350  24-354  24-384  24-388  24-418  24-422  24-523  24-594  24-655  24-706  24-710  24-756
        24-760  24-803  24-807  24-867  24-872  24-915  24-920  24-938  24-939  24-999  24-:00  24-;32  24-;81  24-<27
        24-<87  24-=76  24->32  24-?12  24-?47  24-@27  24-A10  24-A73  24-B56  24-B88  24-D34  24-E35  24-E87  24-F55
        24-F87  24-G40  24-G90  24-H60  25-21   26-16
SVCTST   7-18#   7-30#  24-14   24-39   24-67   24-101  24-128  24-166  24-203  24-247  24-298  24-332  24-366  24-400
        24-449  24-539  24-610  24-676  24-727  24-774  24-832  24-889  24-957  24-:22  24-;52  24-<01  24-<49  24-=07
        24-=91  24->48  24-?27  24-?63  24-@43  24-A26  24-A89  24-B69  24-C05  24-D58  24-E53  24-F05  24-F70  24-G08
        24-G59  24-H13
SW0     12-127#
SW1     12-126#
SW2     12-125#
SW3     12-123#
SYN0    12-233#
SYN1    12-232#
SYN2    12-231#
SYN3    12-230#
SYN4    12-229#
SYN5    12-228#
SYN6    12-227#
SYN7    12-226#
SYNCH   12-234#  24-;55  24-<52  24-=13  24-=54  24-?29  24-?67  24-A30  24-D71  24-F73  24-G62  24-G65  24-G74  24-G79
SYNPAT  24-H15   24-H57  24-H62#
T$$AU   23-9#    23-10
T$$AUT  20-7#    20-19
T$$CLE  21-8#    21-11
T$$DU   22-8#    22-13
```

```
T$$HAR  25-14     25-14#    25-21
T$$HW   10-9      10-9#     10-23
T$$INI  19-8#     19-74
T$$MSG  16-100#   16-102    16-106#   16-115    16-121#   16-134    16-140#   16-153    16-159#   16-173    16-179#   16-192    16-198#   16-210
        16-216#   16-230    16-236#   16-244
T$$PRO  18-8#
T$$RPT  17-9#     17-11
T$$SEG  24-72     24-72#    24-86     24-86#    24-303    24-303#   24-316    24-316#   24-337    24-337#   24-350    24-350#   24-371    24-371#
        24-384    24-384#   24-405    24-405#   24-418    24-418#   24-680    24-680#   24-695    24-706    24-706#   24-731    24-731#   24-746
        24-756    24-756#   24-778    24-778#   24-793    24-803    24-803#   24-838    24-838#   24-856    24-867    24-867#   24-900    24-900#
        24-915    24-915#
T$$SOF  26-13     26-13#    26-16
T$$SUB  24-898#   24-920    24-924#   24-938    24-960#   24-984    24-994    24-999
T$$SW   11-8      11-8#     11-11
T$$TES  24-14#    24-27     24-39#    24-52     24-67#    24-90     24-101#   24-117    24-128#   24-150    24-166#   24-186    24-203#   24-227
        24-232    24-247#   24-279    24-286    24-298#   24-320    24-332#   24-354    24-366#   24-388    24-400#   24-422    24-449#   24-523
        24-539#   24-552    24-566    24-578    24-589    24-594    24-6:0#   24-638    24-648    24-655    24-676#   24-710    24-727#   24-760
        24-774#   24-807    24-832#   24-872    24-889#   24-939    24-957#   24-:00    24-:2     24-:99    24-;22    24-;32    24-;52#   24-;81
        24-<01#   24-<27    24-<49#   24-<87    24-=07#   24-=76    24-=91#   24->32    24->48#   24-?12    24-?27#   24-?47    24-?63#   24-@27
        24-@43#   24-A10    24-A26#   24-A73    24-A89#   24-B56    24-B69#   24-B88    24-C05#   24-D34    24-D58#   24-E35    24-E53#   24-E87
        24-F05#   24-F55    24-F70#   24-F87    24-G08#   24-G40    24-G59#   24-G90    24-H13#   24-H60
T$ARGC   8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17#     8-17#     8-17#
         8-17#     8-17#     8-17#    16-101    16-101    16-101    16-101    16-101#   16-101#   16-101#   16-107    16-107    16-107    16-107
        16-107#   16-107#   16-107#   16-108    16-108    16-108#   16-109    16-109    16-109    16-109    16-109#   16-109#   16-109#   16-110
        16-110    16-110    16-110    16-110#   16-110#   16-110#   16-111    16-111    16-111    16-111    16-111#   16-111#   16-111#   16-112
        16-112    16-112    16-112    16-112    16-112    16-112#   16-112#   16-112#   16-112#   16-112#   16-113    16-113    16-113    16-113#
        16-113#   16-114    16-114    16-114    16-114    16-114    16-114    16-114#   16-114#   16-114#   16-114#   16-114#   16-122    16-122
        16-122    16-122    16-122#   16-122#   16-122#   16-123    16-123    16-123#   16-124    16-124    16-124    16-124    16-124#   16-124#
        16-124#   16-125    16-125    16-125    16-125    16-125#   16-125#   16-125#   16-126    16-126    16-126    16-126    16-126#   16-126#
        16-126#   16-127    16-127    16-127    16-127    16-127    16-127    16-127#   16-127#   16-127#   16-127#   16-127#   16-128    16-128
        16-128    16-128#   16-128#   16-129    16-129    16-129    16-129    16-129    16-129    16-129#   16-129#   16-129#   16-129#   16-129#
        16-130    16-130    16-130    16-130    16-130#   16-130#   16-130#   16-131    16-131    16-131    16-131    16-131    16-131    16-131#
        16-131#   16-131#   16-131#   16-131#   16-132    16-132    16-132    16-132#   16-132#   16-133    16-133    16-133    16-133    16-133
        16-133    16-133#   16-133#   16-133#   16-133#   16-133#   16-141    16-141    16-141    16-141#   16-141#   16-142    16-142    16-142
        16-142    16-142#   16-142#   16-142#   16-143    16-143    16-143#   16-144    16-144    16-144    16-144    16-144#   16-144#   16-144#
        16-145    16-145    16-145    16-145    16-145#   16-145#   16-145#   16-146    16-146    16-146    16-146    16-146    16-146    16-146#
        16-146#   16-146#   16-146#   16-146#   16-147    16-147    16-147    16-147#   16-147#   16-148    16-148    16-148    16-148    16-148
        16-148    16-148#   16-148#   16-148#   16-148#   16-148#   16-149    16-149    16-149    16-149    16-149#   16-149#   16-149#   16-150
        16-150    16-150    16-150    16-150    16-150    16-150#   16-150#   16-150#   16-150#   16-150#   16-151    16-151    16-151    16-151#
        16-151#   16-152    16-152    16-152    16-152    16-152    16-152    16-152#   16-152#   16-152#   16-152#   16-152#   16-160    16-160
        16-160    16-160    16-160#   16-160#   16-160#   16-161    16-161    16-161    16-161    16-161#   16-161#   16-161#   16-162    16-162
        16-162#   16-163    16-163    16-163    16-163    16-163#   16-163#   16-163#   16-164    16-164    16-164    16-164    16-164#   16-164#
        16-164#   16-165    16-165    16-165    16-165    16-165#   16-165#   16-165#   16-166    16-166    16-166    16-166    16-166    16-166
        16-166#   16-166#   16-166#   16-166#   16-166#   16-167    16-167    16-167    16-167#   16-167#   16-168    16-168    16-168    16-168
        16-168    16-168    16-168#   16-168#   16-168#   16-168#   16-168#   16-169    16-169    16-169    16-169    16-169#   16-169#   16-169#
        16-170    16-170    16-170    16-170    16-170    16-170    16-170#   16-170#   16-170#   16-170#   16-170#   16-171    16-171    16-171
        16-171#   16-171#   16-172    16-172    16-172    16-172    16-172    16-172    16-172#   16-172#   16-172#   16-172#   16-172#   16-180
        16-180    16-180    16-180#   16-180#   16-181    16-181    16-181    16-181    16-181#   16-181#   16-181#   16-182    16-182    16-182#
        16-183    16-183    16-183    16-183    16-183#   16-183#   16-183#   16-184    16-184    16-184    16-184    16-184#   16-184#   16-184#
        16-185    16-185    16-185    16-185    16-185    16-185    16-185#   16-185#   16-185#   16-185#   16-185#   16-186    16-186    16-186
        16-186#   16-186#   16-187    16-187    16-187    16-187    16-187    16 187    16-187#   16-187#   16-187#   16-187#   16-187#   16-188
        16-188    16-188    16-188    16-188#   16-188#   16-188#   16-189    1   189   16-189    16-189    16-189    16-189    16-189#   16-189#
        16-189#   16-189#   16-189#   16-190    16-190    16-190    16-190#   1 -190#   16-191    16-191    16-191    16-191    16-191    16-191
        16-191#   16-191#   16-191#   16-191#   16-191#   16-199    16-199    16-199    16-199    16-199#   16-199#   16-199#   16-200    16-200
        16-200#   16-201    16-201    16-201    16-201    16-201#   16-201#   16-201#   16-202    16-202    16-202    16-202    16-202#   16-202#
        16-202#   16-203    16-203    16-203    16-203    16-203    16-203    16-203#   16-203#   16-203#   16-203#   16-203#   16-204    16-204
        16-204    16-204#   16-204#   16-205    16-205    16-205    16-205    16-205    16-205    16-205#   16-205#   16-205#   16-205#   16-205#
```

	16-206	16-206	16-206	16-206	16-206#	16-206#	16-206#	16-207	16-207	16-207	16-207	16-207	16-207	16-207#
	16-207#	16-207#	16-207#	16-207#	16-208	16-208	16-208	16-208#	16-208#	16-209	16-209	16-209	16-209	16-209
	16-209	16-209#	16-209#	16-209#	16-209#	16-209#	16-217	16-217	16-217	16-217#	16-217#	16-218	16-218	16-218
	16-218	16-218#	16-218#	16-218#	16-219	16-219	16-219#	16-220	16-220	16-220	16-220	16-220#	16-220#	16-220#
	16-221	16-221	16-221	16-221	16-221#	16-221#	16-221#	16-222	16-222	16-222	16-222	16-222#	16-222#	16-222#
	16-223	16-223	16-223	16-223	16-223	16-223	16-223#	16-223#	16-223#	16-223#	16-223#	16-224	16-224	16-224
	16-224#	16-224#	16-225	16-225	16-225	16-225	16-225	16-225	16-225#	16-225#	16-225#	16-225#	16-225#	16-226
	16-226	16-226	16-226	16-226#	16-226#	16-226#	16-227	16-227	16-227	16-227	16-227	16-227	16-227#	16-227#
	16-227#	16-227#	16-227#	16-228	16-228	16-228	16-228#	16-228#	16-229	16-229	16-229	16-229	16-229	16-229
	16-229#	16-229#	16-229#	16-229#	16-229#	16-237	16-237	16-237	16-237	16-237#	16-237#	16-237#	16-238	16-238
	16-238#	16-239	16-239	16-239	16-239	16-239#	16-239#	16-239#	16-240	16-240	16-240	16-240	16-240#	16-240#
	16-240#	16-241	16-241	16-241	16-241	16-241	16-241	16-241#	16-241#	16-241#	16-241#	16-241#	16-242	16-242
	16-242	16-242#	16-242#	16-243	16-243	16-243	16-243	16-243	16-243	16-243#	16-243#	16-243#	16-243#	16-243#
	22-12	22-12	22-12	22-12#	22-12#	24-456	24-456	24-456	24-456#	24-456#	24-482	24-482	24-482#	24-501
	24-501	24-501#	24-520	24-520	24-520#									
T$CODE	25-16	25-16	25-16	25-16#	25-16#	25-16#	25-17	25-17	25-17	25-17#	25-17#	25-17#	25-18	25-18
	25-18	25-18#	25-18#	25-18#	25-19	25-19	25-19	25-19#	25-19#	25-19#				
T$ERRN	7-18#	15-277	15-277#	15-283	15-283#	15-293	15-293#	15-299	15-299#	15-406	15-406#	15-412	15-412#	15-570
	15-570#	15-602	15-602#	15-608	15-608#	15-618	15-618#	15-624	15-624#	15-659	15-659#	15-665	15-665#	15-699
	15-699#	15-705	15-705#	15-713	15-713#	15-719	15-719#	15-865	15-865#	15-870	15-870#	15-889	15-889#	15-895
	15-895#	15-904	15-904#	15-910	15-910#	15-919	15-919#	15-925	15-925#	15-934	15-934#	15-940	15-940#	15-:72
	15-:72#	15-:78	15-:78#	24-24	24-24#	24-50	24-50#	24-84	24-84#	24-115	24-115#	24-148	24-148#	24-184
	24-184#	24-226	24-226#	24-278	24-278#	24-314	24-314#	24-348	24-348#	24-382	24-382#	24-416	24-416#	24-480
	24-480#	24-499	24-499#	24-518	24-518#	24-551	24-551#	24-565	24-565#	24-577	24-577#	24-588	24-588#	24-637
	24-637#	24-647	24-647#	24-694	24-694#	24-704	24-704#	24-745	24-745#	24-754	24-754#	24-792	24-792#	24-801
	24-801#	24-855	24-855#	24-865	24-865#	24-913	24-913#	24-936	24-936#	24-983	24-983#	24-993	24-993#	24-:98
	24-:98#	24-;21	24-;21#	24-<83	24-<83#	24-=26	24-=26#	24-=35	24-=35#	24-=47	24-=47#	24-=66	24-=66#	24-=74
	24-=74#	24-?97	24-?97#	24-@12	24-@12#	24-@95	24-@95#	24-A60	24-A60#	24-B41	24-B41#	24-B85	24-B85#	24-C89
	24-C89#													
T$EXCP	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#						
T$FLAG	24-227	24-227#	24-227#	24-279	24-279#	24-279#	24-552	24-552#	24-552#	24-566	24-566#	24-566#	24-578	24-578#
	24-578#	24-589	24-589#	24-589#	24-638	24-638#	24-638#	24-648	24-648#	24-648#	24-695	24-695#	24-695#	24-746
	24-746#	24-746#	24-793	24-793#	24-793#	24-856	24-856#	24-856#	24-984	24-984#	24-984#	24-994	24-994#	24-994#
	24-:99	24-:99#	24-:99#	24-;22	24-;22#	24-;22#								
T$GMAN	7-18#													
T$HILI	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#						
T$LAST	7-18#	26-39#												
T$LOLI	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#						
T$LSYM	7-18	7-18#	10-23	11-11	16-102	16-115	16-134	16-153	16-173	16-192	16-210	16-230	16-244	17-11
	19-74	20-19	21-11	22-13	23-10	24-27	24-52	24-90	24-117	24-150	24-186	24-232	24-286	24-320
	24-354	24-388	24-422	24-523	24-594	24-655	24-710	24-760	24-807	24-872	24-920	24-938	24-939	24-999
	24-:00	24-:32	24-;81	24-<27	24-<87	24-=76	24->32	24-?12	24-?47	24-@27	24-A10	24-A73	24-B56	24-B88
	24-D34	24-E35	24-E87	24-F55	24-F87	24-G40	24-G90	24-H60	25-21	26-16				
T$LTNO	26-39#													
T$NEST	7-18#	7-24	7-24	7-24#	10-9	10-9	10-9#	10-23	10-23	10-23	10-23#	11-8	11-8	11-8#
	11-11	11-11	11-11	11-11#	16-100	16-100	16-100#	16-102	16-102	16-102	16-102#	16-106	16-106	16-106#
	16-115	16-115	16-115	16-115#	16-121	16-121	16-121#	16-134	16-134	16-134	16-134#	16-140	16-140	16-140#
	16-153	16-153	16-153	16-153#	16-159	16-159	16-159#	16-173	16-173	16-173	16-173#	16-179	16-179	16-179#
	16-192	16-192	16-192	16-192#	16-198	16-198	16-198#	16-210	16-210	16-210	16-210#	16-216	16-216	16-216#
	16-230	16-230	16-230	16-230#	16-236	16-236	16-236#	16-244	16-244	16-244	16-244#	17-9	17-9	17-9#
	17-11	17-11	17-11	17-11#	18-8	18-8	18-8#	18-12	18-12	18-12	18-12#	19-8	19-8	19-8#
	19-74	19-74	19-74	19-74#	20-7	20-7	20-7#	20-19	20-19	20-19	20-19#	21-8	21-8	21-8#
	21-11	21-11	21-11	21-11#	22-8	22-8	22-8#	22-13	22-13	22-13	22-13#	23-9	23-9	23-9#
	23-10	23-10	23-10	23-10#	24-14	24-14	24-14#	24-27	24-27	24-27	24-27#	24-39	24-39	24-39#
	24-52	24-52	24-52	24-52#	24-67	24-67	24-67#	24-72	24-72	24-72#	24-86	24-86	24-86	24-86#
	24-90	24-90	24-90	24-90#	24-101	24-101	24-101#	24-117	24-117	24-117	24-117#	24-128	24-128	24-128#
	24-150	24-150	24-150	24-150#	24-166	24-166	24-166#	24-186	24-186	24-186	24-186#	24-203	24-203	24-203#

```
          24-232  24-232  24-232  24-232# 24-247  24-247  24-247# 24-286  24-286  24-286  24-286# 24-298  24-298  24-298#
          24-303  24-303  24-303# 24-316  24-316  24-316  24-316# 24-320  24-320  24-320  24-320# 24-332  24-332  24-332#
          24-337  24-337  24-337# 24-350  24-350  24-350  24-350# 24-354  24-354  24-354  24-354# 24-366  24-366  24-366#
          24-371  24-371  24-371# 24-384  24-384  24-384  24-384# 24-388  24-388  24-388  24-388# 24-400  24-400  24-400#
          24-405  24-405  24-405# 24-418  24-418  24-418  24-418# 24-422  24-422  24-422  24-422# 24-449  24-449  24-449#
          24-523  24-523  24-523  24-523# 24-539  24-539  24-539# 24-594  24-594  24-594  24-594# 24-610  24-610  24-610#
          24-655  24-655  24-655  24-655# 24-676  24-676  24-676# 24-680  24-680  24-680# 24-706  24-706  24-706  24-706#
          24-710  24-710  24-710  24-710# 24-727  24-727  24-727# 24-731  24-731  24-731# 24-756  24-756  24-756  24-756#
          24-760  24-760  24-760  24-760# 24-774  24-774  24-774# 24-778  24-778  24-778# 24-803  24-803  24-803  24-803#
          24-807  24-807  24-807  24-807# 24-832  24-832  24-832# 24-838  24-838  24-838# 24-867  24-867  24-867  24-867#
          24-872  24-872  24-872  24-872# 24-889  24-889  24-889# 24-898  24-898  24-898# 24-900  24-900  24-900# 24-915
          24-915  24-915  24-915# 24-920  24-920  24-920  24-920# 24-924  24-924  24-924# 24-938  24-938  24-938  24-938#
          24-939  24-939  24-939  24-939# 24-957  24-957  24-957# 24-960  24-960  24-960# 24-999  24-999  24-999  24-999#
          24-:00  24-:00  24-:00  24-:00# 24-:22  24-:22  24-:22# 24-;32  24-;32  24-;32  24-;32# 24-;52  24-;52  24-;52#
          24-;81  24-;81  24-;81  24-;81# 24-<01  24-<01  24-<01# 24-<27  24-<27  24-<27  24-<27# 24-<49  24-<49  24-<49#
          24-<87  24-<87  24-<87  24-<87# 24-=07  24-=07  24-=07# 24-=76  24-=76  24-=76  24-=76# 24-=91  24-=91  24-=91#
          24->32  24->32  24->32  24->32# 24->48  24->48  24->48# 24-?12  24-?12  24-?12  24-?12# 24-?27  24-?27  24-?27#
          24-?47  24-?47  24-?47  24-?47# 24-?63  24-?63  24-?63# 24-@27  24-@27  24-@27  24-@27# 24-@43  24-@43  24-@43#
          24-A10  24-A10  24-A10  24-A10# 24-A26  24-A26  24-A26# 24-A73  24-A73  24-A73  24-A73# 24-A89  24-A89  24-A89#
          24-B56  24-B56  24-B56  24-B56# 24-B69  24-B69  24-B69# 24-B88  24-B88  24-B88  24-B88# 24-C05  24-C05  24-C05#
          24-D34  24-D34  24-D34  24-D34# 24-D58  24-D58  24-D58# 24-E35  24-E35  24-E35  24-E35# 24-E53  24-E53  24-E53#
          24-E87  24-E87  24-E87  24-E87# 24-F05  24-F05  24-F05# 24-F55  24-F55  24-F55  24-F55# 24-F70  24-F70  24-F70#
          24-F87  24-F87  24-F87  24-F87# 24-G08  24-G08  24-G08# 24-G40  24-G40  24-G40  24-G40# 24-G59  24-G59  24-G59#
          24-G90  24-G90  24-G90  24-G90# 24-H13  24-H13  24-H13# 24-H60  24-H60  24-H60  24-H60# 25-14   25-14   25-14#
          25-21   25-21   25-21   25-21#  26-13   26-13   26-13#  26-16   26-16   26-16   26-16#  26-37   26-37   26-37
          26-37#
T$NS0      7-24#  26-37
T$NS1     10-9#   10-23   11-8#   11-11   16-100# 16-102  16-106# 16-115  16-121# 16-134  16-140# 16-153  16-159# 16-173
          16-179# 16-192  16-198# 16-210  16-216# 16-230  16-236# 16-244  17-9#   17-11   18-8#   18-12   19-8#   19-74
          20-7#   20-19   21-8#   21-11   22-8#   22-13   23-9#   23-10   24-14#  24-27   24-39#  24-52   24-67#  24-90
          24-101# 24-117  24-128# 24-150  24-166# 24-186  24-203# 24-232  24-247# 24-286  24-298# 24-320  24-332# 24-354
          24-366# 24-388  24-400# 24-422  24-449# 24-523  24-539# 24-594  24-610# 24-655  24-676# 24-710  24-727# 24-760
          24-774# 24-807  24-832# 24-872  24-889# 24-939  24-957# 24-:00  24-:22# 24-;32  24-;52# 24-;81  24-<01# 24-<27
          24-<49# 24-<87  24-=07# 24-=76  24-=91# 24->32  24->48# 24-?12  24-?27# 24-?47  24-?63# 24-@27  24-@43# 24-A10
          24-A26# 24-A73  24-A89# 24-B56  24-B69# 24-B88  24-C05# 24-D34  24-D58# 24-E35  24-E53# 24-E87  24-F05# 24-F55
          24-F70# 24-F87  24-G08# 24-G40  24-G59# 24-G90  24-H13# 24-H60  25-14#  25-21   26-13#  26-16
T$NS2     24-72#  24-86   24-303# 24-316  24-337# 24-350  24-371# 24-384  24-405# 24-418  24-680# 24-706  24-731# 24-756
          24-778# 24-803  24-838# 24-867  24-898# 24-920  24-924# 24-938  24-960# 24-999
T$NS3     24-900# 24-915
T$PTNU     7-18#
T$SAVL     7-18#
T$SEGL     7-18#  24-72   24-72   24-72#  24-86   24-86   24-86   24-86   24-86#  24-303  24-303  24-303# 24-316  24-316
          24-316  24-316  24-316# 24-337  24-337  24-337# 24-350  24-350  24-350  24-350  24-350# 24-371  24-371  24-371#
          24-384  24-384  24-384  24-384  24-384# 24-405  24-405  24-405# 24-418  24-418  24-418  24-418  24-418# 24-680
          24-680  24-680# 24-695  24-706  24-706  24-706  24-706  24-706# 24-731  24-731  24-731# 24-746  24-756  24-756
          24-756  24-756  24-756# 24-778  24-778  24-778# 24-793  24-803  24-803  24-803  24-803  24-803# 24-838  24-838
          24-838# 24-856  24-867  24-867  24-867  24-867  24-867# 24-900  24-900  24-900# 24-915  24-915  24-915  24-915
          24-915#
T$SEKO    24-72#  24-86   24-303# 24-316  24-337# 24-350  24-371# 24-384  24-405# 24-418  24-680# 24-695  24-706  24-731#
          24-746  24-756  24-778# 24-793  24-803  24-838# 24-856  24-867  24-900# 24-915
T$SUBN     7-18#  24-14#  24-39#  24-67#  24-101# 24-128# 24-166# 24-203# 24-247# 24-298# 24-332# 24-366# 24-400# 24-449#
          24-539# 24-610# 24-676# 24-727# 24-774# 24-832# 24-889# 24-898  24-898  24-898# 24-924  24-924  24-924# 24-957#
          24-960  24-960  24-960# 24-:22# 24-;52# 24-<01# 24-<49# 24-=07# 24-=91# 24->48# 24-?27# 24-?63# 24-@43# 24-A26#
          24-A89# 24-B69# 24-C05# 24-D58# 24-E53# 24-F05# 24-F70# 24-G08# 24-G59# 24-H13#
T$TAGL     7-18#
T$TAGN     7-18#  10-9    10-9    10-9#   11-8    11-8    11-8#   16-100  16-100  16-100# 16-106  16-106  16-106# 16-121
          16-121  16-121# 16-140  16-140  16-140# 16-159  16-159  16-159# 16-179  16-179  16-179# 16-198  16-198  16-198#
```

```
         16-216   16-216   16-216#  16-236   16-236   16-236#  17-9     17-9     17-9#    18-8     18-8     18-8#    19-8     19-8
         19-8#    20-7     20-7     20-7#    21-8     21-8     21-8#    22-8     22-8     22-8#    23-9     23-9     23-9#    24-14
         24-14    24-14#   24-39    24-39    24-39#   24-67    24-67    24-67#   24-101   24-101   24-101#  24-128   24-128   24-128#
         24-166   24-166   24-166#  24-203   24-203   24-203#  24-247   24-247   24-247#  24-298   24-298   24-298#  24-332   24-332
         24-332#  24-366   24-366   24-366#  24-400   24-400   24-400#  24-449   24-449   24-449#  24-539   24-539   24-539#  24-610
         24-610   24-610#  24-676   24-676   24-676#  24-727   24-727   24-727#  24-774   24-774   24-774#  24-832   24-832   24-832#
         24-889   24-889   24-889#  24-898   24-898   24-898#  24-924   24-924   24-924#  24-957   24-957   24-957#  24-960   24-960
         24-960#  24-:22   24-:22   24-:22#  24-;52   24-;52   24-;52#  24-<01   24-<01   24-<01#  24-<49   24-<49   24-<49#  24-=07
         24-=07   24-=07#  24-=91   24-=91   24-=91#  24->48   24->48   24->48#  24-?27   24-?27   24-?27#  24-?63   24-?63   24-?63#
         24-@43   24-@43   24-@43#  24-A26   24-A26   24-A26#  24-A89   24-A89   24-A89#  24-B69   24-B69   24-B69#  24-C05   24-C05
         24-C05#  24-D58   24-D58   24-D58#  24-E53   24-E53   24-E53#  24-F05   24-F05   24-F05#  24-F70   24-F70   24-F70#  24-G08
         24-G08   24-G08#  24-G59   24-G59   24-G59#  24-H13   24-H13   24-H13#  25-14    25-14    25-14#   26-13    26-13    26-13#
T$TEMP    9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8
          9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8
          9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8
          9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8
          9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8
          9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8      9-8
          9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#
          9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#
          9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#     9-8#
          9-8#    10-23    10-23#   11-11    11-11#   16-102   16-102#  16-115   16-115#  16-134   16-134#  16-153   16-153#  16-173
         16-173#  16-192   16-192#  16-210   16-210#  16-230   16-230#  16-244   16-244#  17-11    17-11#   18-12    18-12#   19-74
         19-74#   20-19    20-19#   21-11    21-11#   22-13    22-13#   23-10    23-10#   24-27    24-27#   24-52    24-52#   24-86
         24-86#   24-90    24-90#   24-117   24-117#  24-150   24-150#  24-186   24-186#  24-227   24-227#  24-232   24-232#  24-279
         24-279#  24-286   24-286#  24-316   24-316#  24-320   24-320#  24-350   24-350#  24-354   24-354#  24-384   24-384#  24-388
         24-388#  24-418   24-418#  24-422   24-422#  24-523   24-523#  24-552   24-552#  24-566   24-566#  24-578   24-578#  24-589
         24-589#  24-594   24-594#  24-638   24-638#  24-648   24-648#  24-655   24-655#  24-695   24-695#  24-695#  24-706   24-706#
         24-710   24-710#  24-746   24-746#  24-746#  24-756   24-756#  24-760   24-760#  24-793   24-793#  24-793#  24-803   24-803#
         24-807   24-807#  24-856   24-856#  24-856#  24-867   24-867#  24-872   24-872#  24-915   24-915#  24-920   24-920#  24-938
         24-938#  24-939   24-939#  24-984   24-984#  24-994   24-994#  24-999   24-999#  24-:00   24-:00#  24-:99   24-:99#  24-;22
         24-;22#  24-;32   24-;32#  24-;81   24-;81#  24-<27   24-<27#  24-<87   24-<87#  24-=76   24-=76#  24->32   24->32#  24-?12
         24-?12#  24-?47   24-?47#  24-@27   24-@27#  24-A10   24-A10#  24-A73   24-A73#  24-B56   24-B56#  24-B88   24-B88#  24-D34
         24-D34#  24-E35   24-E35#  24-E87   24-E87#  24-F55   24-F55#  24-F87   24-F87#  24-G40   24-G40#  24-G90   24-G90#  24-H60
         24-H60#  25-16    25-16    25-16    25-16#   25-16#   25-16#   25-17    25-17    25-17    25-17#   25-17#   25-17#   25-18
         25-18    25-18    25-18#   25-18#   25-18#   25-19    25-19    25-19    25-19#   25-19#   25-19#   25-21    25-21#   26-16
         26-16#   26-37    26-37#
T$TEST    7-18#   24-14    24-14    24-14#   24-39    24-39    24-39#   24-67    24-67    24-67#   24-101   24-101   24-101#  24-128
         24-128   24-128#  24-166   24-166   24-166#  24-203   24-203   24-203#  24-247   24-247   24-247#  24-298   24-298   24-298#
         24-332   24-332   24-332#  24-366   24-366   24-366#  24-400   24-400   24-400#  24-449   24-449   24-449#  24-539   24-539
         24-539#  24-610   24-610   24-610#  24-676   24-676   24-676#  24-727   24-727   24-727#  24-774   24-774   24-774#  24-832
         24-832   24-832#  24-889   24-889   24-889#  24-898   24-924   24-957   24-957   24-957#  24-960   24-:22   24-:22   24-:22#
         24-;52   24-;52   24-;52#  24-<01   24-<01   24-<01#  24-<49   24-<49   24-<49#  24-=07   24-=07   24-=07#  24-=91   24-=91
         24-=91#  24->48   24->48   24->48#  24-?27   24-?27   24-?27#  24-?63   24-?63   24-?63#  24-@43   24-@43   24-@43#  24-A26
         24-A26   24-A26#  24-A89   24-A89   24-A89#  24-B69   24-B69   24-B69#  24-C05   24-C05   24-C05#  24-D58   24-D58   24-D58#
         24-E53   24-E53   24-E53#  24-F05   24-F05   24-F05#  24-F70   24-F70   24-F70#  24-G08   24-G08   24-G08#  24-G59   24-G59
         24-G59#  24-H13   24-H13   24-H13#  26-39
T$TSTM    7-18#   15-277   15-283   15-293   15-299   15-406   15-412   15-570   15-602   15-608   15-618   15-624   15-659   15-665
         15-699   15-705   15-713   15-719   15-865   15-870   15-889   15-895   15-904   15-910   15-919   15-925   15-934   15-940
         15-:72   15-:78   16-101   16-102   16-107   16-108   16-109   16-110   16-111   16-112   16-113   16-114   16-115   16-122
         16-123   16-124   16-125   16-126   16-127   16-128   16-129   16-130   16-131   16-132   16-133   16-134   16-141   16-142
         16-143   16-144   16-145   16-146   16-147   16-148   16-149   16-150   16-151   16-152   16-153   16-160   16-161   16-162
         16-163   16-164   16-165   16-166   16-167   16-168   16-169   16-170   16-171   16-172   16-173   16-180   16-181   16-182
         16-183   16-184   16-185   16-186   16-187   16-188   16-189   16-190   16-191   16-192   16-199   16-200   16-201   16-202
         16-203   16-204   16-205   16-206   16-207   16-208   16-209   16-210   16-217   16-218   16-219   16-220   16-221   16-222
         16-223   16-224   16-225   16-226   16-227   16-228   16-229   16-230   16-237   16-238   16-239   16-240   16-241   16-242
         16-243   16-244   17-11    19-25    19-28    19-31    19-34    19-51    19-74    20-9     20-16    20-19    21-11    22-10
```

```
          22-12     22-13     23-10     24-16     24-24     24-27     24-50     24-52     24-72     24-84     24-86     24-90     24-115    24-117
          24-133    24-148    24-150    24-184    24-186    24-226    24-227    24-232    24-278    24-279    24-286    24-303    24-314    24-316
          24-320    24-337    24-348    24-350    24-354    24-371    24-382    24-384    24-388    24-405    24-416    24-418    24-422    24-456
          24-480    24-482    24-499    24-501    24-518    24-520    24-523    24-551    24-552    24-565    24-566    24-577    24-578    24-588
          24-589    24-594    24-637    24-638    24-647    24-648    24-655    24-680    24-694    24-695    24-704    24-706    24-710    24-731
          24-745    24-746    24-754    24-756    24-760    24-778    24-792    24-793    24-801    24-803    24-807    24-838    24-855    24-856
          24-865    24-867    24-872    24-898    24-900    24-913    24-915    24-920    24-924    24-936    24-938    24-939    24-960    24-983
          24-984    24-993    24-994    24-999    24-:00    24-:98    24-:99    24-;21    24-;22    24-;32    24-;81    24-<27    24-<83    24-<87
          24-=26    24-=35    24-=47    24-=66    24-=74    24-=76    24->32    24-?12    24-?47    24-?97    24-@12    24-@27    24-@95    24-A10
          24-A60    24-A73    24-B41    24-B56    24-B85    24-B88    24-C89    24-D34    24-E35    24-E87    24-F55    24-F87    24-G40    24-G90
          24-H60
T$TSTS     7-18#    24-14#    24-39#    24-67#    24-101#   24-128#   24-166#   24-203#   24-247#   24-298#   24-332#   24-366#   24-400#   24-449#
          24-539#   24-610#   24-676#   24-727#   24-774#   24-832#   24-889#   24-957#   24-:22#   24-;52#   24-<01#   24-<49#   24-=07#   24-=91#
          24->48#   24-?27#   24-?63#   24-@43#   24-A26#   24-A89#   24-B69#   24-C05#   24-D58#   24-E53#   24-F05#   24-F70#   24-G08#   24-G59#
          24-H13#
T1          9-8     24-14#
T10         9-8     24-332#
T11         9-8     24-366#
T12         9-8     24-400#
T13         9-8     24-449#
T14         9-8     24-539#
T15         9-8     24-610#
T16         9-8     24-676#
T17         9-8     24-727#
T18         9-8     24-774#
T19         9-8     24-832#
T2          9-8     24-39#
T20         9-8     24-889#
T20.1     24-898#
T20.2     24-924#
T21         9-8     24-957#
T21.1     24-960#
T22         9-8     24-:22#
T23         9-8     24-;52#
T24         9-8     24-<01#
T25         9-8     24-<49#
T26         9-8     24-=07#
T27         9-8     24-=91#
T28         9-8     24->48#
T29         9-8     24-?27#
T3          9-8     24-67#
T30         9-8     24-?63#
T31         9-8     24-@43#
T32         9-8     24-A26#
T33         9-8     24-A89#
T34         9-8     24-B69#
T35         9-8     24-C05#
T36         9-8     24-D58#
T37         9-8     24-E53#
T38         9-8     24-F05#
T39         9-8     24-F70#
T4          9-8     24-101#
T40         9-8     24-G08#
T41         9-8     24-G59#
T42         9-8     24-H13#
T5          9-8     24-128#
T6          9-8     24-166#
```

```
T7      9-8      24-203#
T8      9-8      24-247#
T9      9-8      24-298#
TEOM    12-220#  24-C45   24-C68
TERR    12-217#
TEST    12-258#  24-844
TESTMD  12-174#
TIMFLG  13-24#
TMP0    13-89#   15-229*  15-232*  16-124   16-163   16-183
TMP1    13-90#   15-247*  15-248*  16-124   16-163   16-183
TMP2    13-91#
TMP3    13-92#
TMP4    13-93#
TMP5    13-94#
TMP6    13-95#
TMP7    13-96#
TSOM    12-221#
TSTCON  13-65#
TSTNUM  13-49#   24-451*  24-456
TX0     12-54#   12-212#
TX1     12-53#   12-211#
TX2     12-52#   12-210#
TX3     12-51#   12-209#
TX4     12-50#   12-208#
TX5     12-49#   12-207#
TX6     12-48#   12-206#
TX7     12-47#   12-205#
TXAB    12-219#
TXABT   12-279#
TXCHAR  15-510#  24-;57   24-;60   24-;63   24-;66   24-;69   24-;72   24-;75   24-<06   24-<09   24-<12   24-<15   24-<18   24-<21
        24-<57   24-<60   24-<63   24-<66   24-<69   24-<72   24-=15   24-=56   24->00   24->03   24->08   24->13   24->18   24->21
        24->24   24->27   24->53   24->56   24->59   24->66   24->71   24->76   24->81   24->86   24->91   24->96   24-?01   24-?04
        24-?07
TXDATA  12-171#  15-:68   15-:74
TXEN    12-86#   12-158#  24-140   24-<55   24-C17   24-C19
TXEOM   12-280#  13-468   13-469   13-470   13-471   13-477   13-478   24-;64   24-;67   24-;70   24-;73   24-;76   24-<13   24-<16
        24-<19   24-<22   24-<64   24-<67   24-<70   24-<73   24->19   24->22   24->25   24->28   24-?02   24-?05   24-?08   24-E30
        24-G16   24-H28
TXGA    12-218#
TXGOA   12-278#  24-G16
TXLEN0  12-266#  15-849   24-=98   24->11   24->64   24->74   24->84   24->94
TXLEN1  12-265#  15-849   24->06   24->11   24->69   24->74   24->89   24->94
TXLEN2  12-264#  15-849   24-=98   24->06   24->11   24->79   24->84   24->89   24->94
TXSOM   12-281#  13-461   13-462   15-448   24-:33   24-=37   24-H20
TXWORD  13-41#   15-344*  15-346   15-349   15-448*  15-449*  15-514*  15-:05*  24-:33*  24-:58*  24-;06*  24-=37*  24-?33*  24-?76*
        24-@56*  24-A39*  24-B02*  24-B75*  24-D74*  24-D96*  24-E26*  24-E30*  24-E59*  24-F11*  24-F75*  24-G16*  24-G65*  24-H20*
        24-H28*
UAM     12-20#
UNIT    13-48#
UNRR    12-128#  15-861   24-=22   24-=31   24-=43   24-=62   24-=70
UPBITS  13-99#   24-215   24-260   24-266   24-307   24-341   24-375   24-409
V35     12-254#  12-259
VECTOR  25-17    25-24#
WAIT50  15-316#  15-458   15-516   15-559   15-780   15-:09   15-:29   24-971   24-:36   24-:60   24-=39   24-?37   24-?80   24-@60
        24-A43   24-B06   24-D29   24-E63   24-F15   24-G19   24-G69   24-H31
WAX     12-89#   12-161#  15-205
WAX15   13-30#   15-195*  15-196   15-447*  15-449   15-970*  15-979*  24-543*  24-619*  24-621   24-682*  24-684*  24-688   24-690
```

```
         24-733*  24-735*  24-739   24-741   24-780*  24-782*  24-786   24-788   24-840*  24-844   24-=97*  24->63*  24-C27*  24-C39*
         24-C67*  24-E65*  24-F17*
WAX16    13-31#   15-199*  15-200   15-450*  15-971*  15-981*  15-982   24-544*  24-620*  24-622   24-683*  24-685*  24-734*  24-736*
         24-748   24-750   24-781*  24-783*  24-795   24-797   24-841*  24-=98*  24->06*  24->11*  24->16*  24->64*  24->69*  24->74*
         24->79*  24->84*  24->89*  24->94*  24->99*  24-C28*  24-C40*  24-C45*  24-C68*  24-E66*  24-E70*  24-E74*  24-E78*  24-F20*
         24-F24*  24-F28*  24-F32*  24-F36*  24-F40*  24-F44*  24-F48*
WRDYTO   12-342#  15-188   15-214   24-546
WRIBYT   13-27#   15-41*   15-89*   15-90    15-152*  15-153*  15-154*  15-155*  15-190*  15-191*  15-192*  15-196*  15-200*  15-203*
         15-204*  15-205*  15-206*  15-346*  15-349*  15-454*  15-557*  15-816*  15-817*  15-823*  15-975*  24-73*   24-74*   24-78
         24-80    24-104*  24-131*  24-207*  24-258*  24-259   24-304*  24-308   24-310   24-338*  24-342   24-344   24-372*  24-376
         24-378   24-406*  24-410   24-412   24-901*  24-925*  24-965*  24-968*  24-<55*  24-?73*  24-@53*  24-A05*  24-A36*  24-A68*
         24-A99*  24-B51*  24-B72*  24-C17*  24-C21*  24-D24*  24-D27*  24-G82*
WRITAX   15-186#  15-451   15-972   15-983   24-545   24-623   24-686   24-737   24-784   24-842   24-=99   24->07   24->12   24->17
         24->65   24->70   24->75   24->80   24->85   24->90   24->95   24-?00   24-C29   24-C48   24-C69   24-E67   24-E71   24-E75
         24-E79   24-F21   24-F25   24-F29   24-F33   24-F37   24-F41   24-F45   24-F49
WRITLU   15-42    [illegible]3#  15-156   15-193   15-197   15-201   15-207   15-347   15-350   15-455   15-558   15-818   15-824   15-976
         24-75    24-105   24-132   24-208   24-261   24-305   24-339   24-373   24-407   24-902   24-926   24-966   24-969   24-970
         24-<56   24-?74   24-@54   24-A06   24-A37   24-A69   24-B00   24-B52   24-B73   24-C18   24-C22   24-D25   24-D28   24-G83
X$ALWA    7-18#
X$FALS    7-18#
X$OFFS    7-18#
X$TRUE    7-18#
XYZ      12-252#  12-259
```

```
BCOMPL   1-15#    7-18#    19-26    19-29    19-32    19-35    19-52
BERROR   1-19#    7-18#
BGNAU    1-23#    7-18#    23-9
BGNAUT   1-31#    7-18#    20-7
BGNCLN   1-39#    7-18#    21-8
BGNDU    1-47#    7-18#    22-8
BGNHRD   1-55#    7-18#    25-14
BGNHW    1-66#    7-18#    10-9
BGNINI   1-77#    7-18#    19-8
BGNMOD   1-85#    7-18#     7-24
BGNMSG   1-98#    7-18#    16-100   16-106   16-121   16-140   16-159   16-179   16-198   16-216   16-236
BGNPRO   1-106#   7-18#    18-8
BGNPTA   1-114#   7-18#
BGNRPT   1-144#   7-18#    17-9
BGNSEG   1-152#   7-18#    24-72    24-303   24-337   24-371   24-405   24-680   24-731   24-778   24-838   24-900
BGNSET   1-161#   7-18#
BGNSFT   1-182#   7-18#    26-13
BGNSRV   1-193#   7-18#
BGNSUB   1-201#   7-18#    24-898   24-924   24-960
BGNSW    1-225#   7-18#    11-8
BGNTST   1-236#   7-18#    24-14    24-39    24-67    24-101   24-128   24-166   24-203   24-247   24-298   24-332   24-366   24-400
         24-449   24-539   24-610   24-676   24-727   24-774   24-832   24-889   24-957   24-:22   24-;52   24-<01   24-<49   24-=07
         24-=91   24->48   24-?27   24-?63   24-@43   24-A26   24-A89   24-B69   24-C05   24-D58   24-E53   24-F05   24-F70   24-G08
         24-G59   24-H13
BNCOMP   1-266#   7-18#
BNERRO   1-270#   7-18#
BREAK    1-274#   7-18#
BRESET   1-278#   7-18#    22-10    24-133
CKLOOP   1-282#   7-18#
CLOCK    1-286#   7-18#
CLOSE    1-292#   7-18#
CLRVEC   1-296#   7-18#
COMMEN   1-301#   7-18#
DELAY    1-322#   7-18#
DESCRI   1-317#   7-18#    14-17
DEVTYP   1-341#   7-18#    14-12
DISPAT   1-346#   7-18#     9-8
DISPLA   1-360#   7-18#
DOCLN    1-376#   7-18#
DODU     1-380#   7-18#    20-16
DORPT    1-385#   7-18#
ENDAU    1-389#   7-18#    23-10
ENDAUT   1-401#   7-18#    20-19
ENDCLN   1-413#   7-18#    21-11
ENDCOM   1-425#   7-18#
ENDDU    1-441#   7-18#    22-13
ENDHRD   1-453#   7-18#    25-21
ENDHW    1-465#   7-18#    10-23
ENDINI   1-475#   7-18#    19-74
ENDMOD   1-487#   7-18#    26-37
ENDMSG   1-500#   7-18#    16-102   16-115   16-134   16-153   16-173   16-192   16-210   16-230   16-244
ENDPRO   1-512#   7-18#    18-12
ENDPTA   1-520#   7-18#
ENDRPT   1-529#   7-18#    17-11
ENDSEG   1-541#   7-18#    24-86    24-316   24-350   24-384   24-418   24-706   24-756   24-803   24-867   24-915
ENDSET   1-555#   7-18#
```

```
ENDSFT   1-568#   7-18#    26-16
ENDSRV   1-580#   7-18#
ENDSUB   1-596#   7-18#    24-920   24-938   24-999
ENDSW    1-614#   7-18#    11-11
ENDTST   1-624#   7-18#    24-27    24-52    24-90    24-117   24-150   24-186   24-232   24-286   24-320   24-354   24-388   24-422
         24-523   24-594   24-655   24-710   24-760   24-807   24-872   24-939   24-:00   24-;32   24-;81   24-<27   24-<87   24-=76
         24->32   24-?12   24-?47   24-@27   24-A10   24-A73   24-B56   24-B88   24-D34   24-E35   24-E87   24-F55   24-F87   24-G40
         24-G90   24-H60
EQUALS   1-642#   7-18#    12-20
ERRDF    1-714#   7-18#    15-277   15-283   15-293   15-299   15-406   15-412   15-570   15-602   15-608   15-618   15-624   15-659
         15-665   15-699   15-705   15-713   15-719   15-865   15-870   15-889   15-895   15-904   15-910   15-919   15-925   15-934
         15-940   15-:72   15-:78   24-24    24-50    24-84    24-115   24-148   24-184   24-226   24-278   24-314   24-348   24-382
         24-416   24-480   24-499   24-518   24-551   24-565   24-577   24-588   24-637   24-647   24-694   24-704   24-745   24-754
         24-792   24-801   24-855   24-865   24-913   24-936   24-983   24-993   24-:98   24-;21   24-<83   24-=26   24-=35   24-=47
         24-=66   24-=74   24-?97   24-@12   24-@95   24-A60   24-B41   24-B85   24-C89
ERRHRD   1-718#   7-18#
ERROR    1-722#   7-18#
ERRSF    1-726#   7-18#
ERRSOF   1-730#   7-18#
ERRTBL   1-734#   7-18#
ESCAPE   1-744#   7-18#    24-227   24-279   24-552   24-566   24-578   24-589   24-638   24-648   24-695   24-746   24-793   24-856
         24-984   24-994   24-:99   24-;22
EXIT     1-771#   7-18#
FEQUAL   1-810#   7-18#
GETBYT   1-824#   7-18#
GETPRI   1-834#   7-18#
GETWOR   1-829#   7-18#
GMANIA   1-839#   7-18#
GMANID   1-848#   7-18#
GMANIL   1-859#   7-18#
GPHARD   1-868#   7-18#    19-51
GPRMA    1-874#   7-18#    25-16    25-17
GPRMD    1-903#   7-18#    25-18    25-19
GPRML    1-934#   7-18#
HEADER   1-954#   7-18#    8-17
INLOOP   1-962#   7-18#
IOSETU   1-966#   7-18#
IOSTAR   1-974#   7-18#
KT11     1-982#   7-18#
LASTAD   1-;47#   7-18#    26-39
MSBYTE   1-D00#   7-18#    8-17     8-17     8-17     8-17#
MSCHEC   1-E18#   7-18#
MSCNTO   1-E82#   7-18#    25-16    25-16#   25-17    25-17#   25-18    25-18#   25-19    25-19#
MSCOUN   1-D66#   7-18#    16-101   16-101   16-101#  16-107   16-107   16-107#  16-108   16-108#  16-109   16-109   16-109#  16-110
         16-110   16-110#  16-111   16-111   16-111#  16-112   16-112   16-112   16-112   16-112#  16-113   16-113#  16-114   16-114
         16-114   16-114   16-114#  16-122   16-122   16-122#  16-123   16-123#  16-124   16-124   16-124#  16-125   16-125   16-125#
         16-126   16-126   16-126#  16-127   16-127   16-127   16-127   16-127#  16-128   16-128#  16-129   16-129   16-129   16-129
         16-129#  16-130   16-130   16-130#  16-131   16-131   16-131   16-131   16-131#  16-132   16-132#  16-133   16-133   16-133
         16-133   16-133#  16-141   16-141#  16-142   16-142   16-142#  16-143   16-143#  16-144   16-144   16-144#  16-145   16-145
         16-145#  16-146   16-146   16-146   16-146   16-146#  16-147   16-147#  16-148   16-148   16-148   16-148   16-148#  16-149
         16-149   16-149#  16-150   16-150   16-150   16-150   16-150#  16-151   16-151#  16-152   16-152   16-152   16-152   16-152#
         16-160   16-160   16-160#  16-161   16-161   16-161#  16-162   16-162#  16-163   16-163   16-163#  16-164   16-164   16-164#
         16-165   16-165   16-165#  16-166   16-166   16-166   16-166   16-166#  16-167   16-167#  16-168   16-168   16-168   16-168
         16-168#  16-169   16-169   16-169#  16-170   16-170   16-170   16-170   16-170#  16-171   16-171#  16-172   16-172   16-172
         16-172   16-172#  16-180   16-180#  16-181   16-181   16-181#  16-182   16-182#  16-183   16-183   16-183#  16-184   16-184
         16-184#  16-185   16-185   16-185   16-185   16-185#  16-186   16-186#  16-187   16-187   16-187   16-187   16-187#  16-188
         16-188   16-188#  16-189   16-189   16-189   16-189   16-189#  16-190   16-190#  16-191   16-191   16-191   16-191   16-191#
```

```
         16-199    16-199    16-199#   16-200    16-200#   16-201    16-201    16-201#   16-202    16-202    16-202#   16-203    16-203    16-203
         16-203    16-203#   16-204    16-204#   16-205    16-205    16-205    16-205    16-205#   16-206    16-206    16-206#   16-207    16-207
         16-207    16-207    16-207#   16-208    16-208#   16-209    16-209    16-209    16-209    16-209#   16-217    16-217#   16-218    16-218
         16-218#   16-219    16-219#   16-220    16-220    16-220#   16-221    16-221    16-221#   16-222    16-222    16-222#   16-223    16-223
         16-223    16-223    16-223#   16-224    16-224#   16-225    16-225    16-225    16-225    16-225#   16-226    16-226    16-226#   16-227
         16-227    16-227    16-227    16-227#   16-228    16-228#   16-229    16-229    16-229    16-229    16-229#   16-237    16-237    16-237#
         16-238    16-238#   16-239    16-239    16-239#   16-240    16-240    16-240#   16-241    16-241    16-241    16-241    16-241#   16-242
         16-242#   16-243    16-243    16-243    16-243    16-243#   22-12     22-12#    24-456    24-456#   24-482    24-482#   24-501    24-501#
         24-520    24-520#
M$DATA    1-B67#    7-18#     8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17
          8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17
          8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17#     8-17#     14-12
         14-12#    14-17     14-17#
M$DECR    1-D29#    7-18#    10-23     10-23#    11-11     11-11#    16-102    16-102#   16-115    16-115#   16-134    16-134#   16-153    16-153#
         16-173    16-173#   16-192    16-192#   16-210    16-210#   16-230    16-230#   16-244    16-244#   17-11     17-11#    18-12     18-12#
         19-74     19-74#    20-19     20-19#    21-11     21-11#    22-13     22-13#    23-10     23-10#    24-27     24-27#    24-52     24-52#
         24-86     24-86     24-86#    24-86#    24-90     24-90#    24-117    24-117#   24-150    24-150#   24-186    24-186#   24-232    24-232#
         24-286    24-286#   24-316    24-316    24-316#   24-316#   24-320    24-320#   24-350    24-350    24-350#   24-350#   24-354    24-354#
         24-384    24-384    24-384#   24-384#   24-388    24-388#   24-418    24-418    24-418#   24-418#   24-422    24-422#   24-523    24-523#
         24-594    24-594#   24-655    24-655#   24-706    24-706    24-706#   24-706#   24-710    24-710#   24-756    24-756    24-756#   24-756#
         24-760    24-760#   24-803    24-803    24-803#   24-803#   24-807    24-807#   24-867    24-867    24-867#   24-867#   24-872    24-872#
         24-915    24-915    24-915#   24-915#   24-920    24-920#   24-938    24-938#   24-939    24-939#   24-999    24-999#   24-:00    24-:00#
         24-;32    24-;32#   24-;81    24-;81#   24-<27    24-<27#   24-<87    24-<87#   24-=76    24-=76#   24->32    24->32#   24-?12    24-?12#
         24-?47    24-?47#   24-@27    24-@27#   24-A10    24-A10#   24-A73    24-A73#   24-B56    24-B56#   24-B88    24-B88#   24-D34    24-D34#
         24-E35    24-E35#   24-E87    24-E87#   24-F55    24-F55#   24-F87    24-F87#   24-G40    24-G40#   24-G90    24-G90#   24-H60    24-H60#
         25-21     25-21#    26-16     26-16#    26-37     26-37#
M$DEFA    1-E70#    7-18#    25-16     25-16#    25-17     25-17#    25-18     25-18#    25-19     25-19#
M$ENDE    1-D74#    7-18#    10-23#    11-11#    16-102#   16-115#   16-134#   16-153#   16-173#   16-192#   16-210#   16-230#   16-244#   17-11#
         19-74#    20-19#    21-11#    22-13#    23-10#    24-27#    24-52#    24-86#    24-90#    24-117#   24-150#   24-186#   24-232#   24-286#
         24-316#   24-320#   24-350#   24-354#   24-384#   24-388#   24-418#   24-422#   24-523#   24-594#   24-655#   24-706#   24-710#   24-756#
         24-760#   24-803#   24-807#   24-867#   24-872#   24-915#   24-920#   24-938#   24-939#   24-999#   24-:00#   24-;32#   24-;81#   24-<27#
         24-<87#   24-=76#   24->32#   24-?12#   24-?47#   24-@27#   24-A10#   24-A73#   24-B56#   24-B88#   24-D34#   24-E35#   24-E87#   24-F55#
         24-F87#   24-G40#   24-G90#   24-H60#   25-21#    26-16#    26-37#
M$ERRI    1-@49#    7-18#    15-277    15-277#   15-283    15-283#   15-293    15-293#   15-299    15-299#   15-406    15-406#   15-412    15-412#
         15-570    15-570#   15-602    15-602#   15-608    15-608#   15-618    15-618#   15-624    15-624#   15-659    15-659#   15-665    15-665#
         15-699    15-699#   15-705    15-705#   15-713    15-713#   15-719    15-719#   15-865    15-865#   15-870    15-870#   15-889    15-889#
         15-895    15-895#   15-904    15-904#   15-910    15-910#   15-919    15-919#   15-925    15-925#   15-934    15-934#   15-940    15-940#
         15-:72    15-:72#   15-:78    15-:78#   24-24     24-24#    24-50     24-50#    24-84     24-84#    24-115    24-115#   24-148    24-148#
         24-184    24-184#   24-226    24-226#   24-278    24-278#   24-314    24-314#   24-348    24-348#   24-382    24-382#   24-416    24-416#
         24-480    24-480#   24-499    24-499#   24-518    24-518#   24-551    24-551#   24-565    24-565#   24-577    24-577#   24-588    24-588#
         24-637    24-637#   24-647    24-647#   24-694    24-694#   24-704    24-704#   24-745    24-745#   24-754    24-754#   24-792    24-792#
         24-801    24-801#   24-855    24-855#   24-865    24-865#   24-913    24-913#   24-936    24-936#   24-983    24-983#   24-993    24-993#
         24-:98    24-:98#   24-;21    24-;21#   24-<83    24-<83#   24-=26    24-=26#   24-=35    24-=35#   24-=47    24-=47#   24-=66    24-=66#
         24-=74    24-=74#   24-?97    24-?97#   24-@12    24-@12#   24-@95    24-@95#   24-A60    24-A60#   24-B41    24-B41#   24-B85    24-B85#
         24-C89    24-C89#
M$ESCA    1-D06#    7-18#    24-227    24-227#   24-279    24-279#   24-552    24-552#   24-566    24-566#   24-578    24-578#   24-589    24-589#
         24-638    24-638#   24-648    24-648#   24-695#   24-746#   24-793#   24-856#   24-984    24-984#   24-994    24-994#   24-:99    24-:99#
         24-;22    24-;22#
M$ESCS    1-D10#    7-18#    24-227#   24-279#   24-552#   24-566#   24-578#   24-589#   24-638#   24-648#   24-695    24-695#   24-746    24-746#
         24-793    24-793#   24-856    24-856#   24-984#   24-994#   24-:99#   24-;22#
M$EXCP    1-E01#    7-18#    25-16     25-16     25-16#    25-17     25-17     25-17#    25-18     25-18     25-18#    25-19     25-19     25-19#
M$EXIT    1-D14#    7-18#
M$EXSE    1-D22#    7-18#
M$EXTJ    1-D18#    7-18#
M$GEN     1-D38#    7-18#     7-24      7-24#     8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17
          8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17
          8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17      8-17
```

```
          8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#
          8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#
          8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    9-8      9-8#     10-9     10-9
          10-9#    10-9#    10-23    10-23#   11-8     11-8     11-8#    11-8#    11-11    11-11#   14-12    14-12#   14-17    14-17#
          16-100   16-100#  16-102   16-102#  16-106   16-106#  16-115   16-115#  16-121   16-121#  16-134   16-134#  16-140   16-140#
          16-153   16-153#  16-159   16-159#  16-173   16-173#  16-179   16-179#  16-192   16-192#  16-198   16-198#  16-210   16-210#
          16-216   16-216#  16-230   16-230#  16-236   16-236#  16-244   16-244#  17-9     17-9#    17-11    17-11#   18-8     18-8#
          19-8     19-8#    19-74    19-74#   20-7     20-7#    20-19    20-19#   21-8     21-8#    21-11    21-11#   22-8     22-8#
          22-13    22-13#   23-9     23-9#    23-10    23-10#   24-14    24-14#   24-27    24-27#   24-39    24-39#   24-52    24-52#
          24-67    24-67#   24-86    24-86#   24-90    24-90#   24-101   24-101#  24-117   24-117#  24-128   24-128#  24-150   24-150#
          24-166   24-166#  24-186   24-186#  24-203   24-203#  24-232   24-232#  24-247   24-247#  24-286   24-286#  24-298   24-298#
          24-316   24-316#  24-320   24-320#  24-332   24-332#  24-350   24-350#  24-354   24-354#  24-366   24-366#  24-384   24-384#
          24-388   24-388#  24-400   24-400#  24-418   24-418#  24-422   24-422#  24-449   24-449#  24-523   24-523#  24-539   24-539#
          24-594   24-594#  24-610   24-610#  24-655   24-655#  24-676   24-676#  24-706   24-706#  24-710   24-710#  24-727   24-727#
          24-756   24-756#  24-760   24-760#  24-774   24-774#  24-803   24-803#  24-807   24-807#  24-832   24-832#  24-867   24-867#
          24-872   24-872#  24-889   24-889#  24-898   24-898#  24-915   24-915#  24-920   24-920#  24-924   24-924#  24-938   24-938#
          24-939   24-939#  24-957   24-957#  24-960   24-960#  24-999   24-999#  24-:00   24-:00#  24-:22   24-:22#  24-;32   24-;32#
          24-;52   24-;52#  24-;81   24-;81#  24-<01   24-<01#  24-<27   24-<27#  24-<49   24-<49#  24-<87   24-<87#  24-=07   24-=07#
          24-=76   24-=76#  24-=91   24-=91#  24->32   24->32#  24->48   24->48#  24-?12   24-?12#  24-?27   24-?27#  24-?47   24-?47#
          24-?63   24-?63#  24-@27   24-@27#  24-@43   24-@43#  24-A10   24-A10#  24-A26   24-A26#  24-A73   24-A73#  24-A89   24-A89#
          24-B56   24-B56#  24-B69   24-B69#  24-B88   24-B88#  24-C05   24-C05#  24-D34   24-D34#  24-D58   24-D58#  24-E35   24-E35#
          24-E53   24-E53#  24-E87   24-E87#  24-F05   24-F05#  24-F55   24-F55#  24-F70   24-F70#  24-F87   24-F87#  24-G08   24-G08#
          24-G40   24-G40#  24-G59   24-G59#  24-G90   24-G90#  24-H13   24-H13#  24-H60   24-H60#  25-14    25-14#   25-21    25-21#
          26-13    26-13#   26-16    26-16#   26-39    26-39#
MSGENB    1-C38#   7-18#
MSGETS    1-D35#   7-18#    10-23    10-23#   11-11    11-11#   16-102   16-102#  16-115   16-115#  16-134   16-134#  16-153   16-153#
          16-173   16-173#  16-192   16-192#  16-210   16-210#  16-230   16-230#  16-244   16-244#  17-11    17-11#   18-12    18-12#
          19-74    19-74#   20-19    20-19#   21-11    21-11#   22-13    22-13#   23-10    23-10#   24-27    24-27#   24-52    24-52#
          24-86    24-86    24-86#   24-86#   24-90    24-90#   24-117   24-117#  24-150   24-150#  24-186   24-186#  24-232   24-232#
          24-286   24-286#  24-316   24-316   24-316#  24-316#  24-320   24-320#  24-350   24-350   24-350#  24-350#  24-354   24-354#
          24-384   24-384   24-384#  24-384#  24-388   24-388#  24-418   24-418   24-418#  24-418#  24-422   24-422#  24-523   24-523#
          24-594   24-594#  24-655   24-655#  24-695   24-695#  24-706   24-706   24-706#  24-706#  24-710   24-710#  24-746   24-746#
          24-756   24-756   24-756#  24-756#  24-760   24-760#  24-793   24-793#  24-803   24-803   24-803#  24-803#  24-807   24-807#
          24-856   24-856#  24-867   24-867   24-867#  24-867#  24-872   24-872#  24-915   24-915   24-915#  24-915#  24-920   24-920#
          24-938   24-938#  24-939   24-939#  24-999   24-999#  24-:00   24-:00#  24-;32   24-;32#  24-;81   24-;81#  24-<27   24-<27#
          24-<87   24-<87#  24-=76   24-=76#  24->32   24->32#  24-?12   24-?12#  24-?47   24-?47#  24-@27   24-@27#  24-A10   24-A10#
          24-A73   24-A73#  24-B56   24-B56#  24-B88   24-B88#  24-D34   24-D34#  24-E35   24-E35#  24-E87   24-E87#  24-F55   24-F55#
          24-F87   24-F87#  24-G40   24-G40#  24-G90   24-G90#  24-H60   24-H60#  25-21    25-21#   26-16    26-16#   26-37    26-37#
MSGETT    1-B77#   7-18#    24-227#  24-279#  24-552#  24-566#  24-578#  24-589#  24-638#  24-648#  24-695   24-695#  24-746   24-746#
          24-793   24-793#  24-856   24-856#  24-984#  24-994#  24-:99#  24-;22#
MSGNGB    1-C02#   7-18#    7-24     7-24#    8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
          8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
          8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
          8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#
          8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#
          8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    9-8      9-8#     10-9     10-9
          10-9#    11-8     11-8     11-8#    14-12    14-12#   14-17    14-17#   16-100   16-100#  16-106   16-106#  16-121   16-121#
          16-140   16-140#  16-159   16-159#  16-179   16-179#  16-198   16-198#  16-216   16-216#  16-236   16-236#  17-9     17-9#
          18-8     18-8#    19-8     19-8#    20-7     20-7#    21-8     21-8#    22-8     22-8#    23-9     23-9#    25-14    25-14#
          26-13    26-13#   26-39    26-39#
MSGNIN    1-D49#   7-18#    8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
          8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
          8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17
          8-17     8-17     8-17     8-17     8-17     8-17     8-17     8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#
          8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#
          8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#
          8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    8-17#    9-8      9-8      9-8      9-8      9-8      9-8
```

9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8#	9-8#	9-8#	9-8#	9-8#
9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#
9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#
9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	10-9	10-9#	11-8	11-8#
14-12	14-12	14-12#	14-12#	14-17	14-17	14-17#	14-17#	15-277	15-277	15-277	15-277	15-277#	15-277#
15-277#	15-277#	15-277#	15-283	15-283	15-283	15-283	15-283#	15-283#	15-283#	15-283#	15-283#	15-293	15-293
15-293	15-293	15-293#	15-293#	15-293#	15-293#	15-293#	15-299	15-299	15-299	15-299	15-299#	15-299#	15-299#
15-299#	15-299#	15-406	15-406	15-406	15-406	15-406#	15-406#	15-406#	15-406#	15-406#	15-412	15-412	15-412
15-412	15-412#	15-412#	15-412#	15-412#	15-412#	15-570	15-570	15-570	15-570	15-570#	15-570#	15-570#	15-570#
15-570#	15-602	15-602	15-602	15-602	15-602#	15-602#	15-602#	15-602#	15-602#	15-608	15-608	15-608	15-608
15-608#	15-608#	15-608#	15-608#	15-608#	15-618	15-618	15-618	15-618	15-618#	15-618#	15-618#	15-618#	15-618#
15-624	15-624	15-624	15-624	15-624#	15-624#	15-624#	15-624#	15-624#	15-659	15-659	15-659	15-659	15-659#
15-659#	15-659#	15-659#	15-659#	15-665	15-665	15-665	15-665	15-665#	15-665#	15-665#	15-665#	15-665#	15-699
15-699	15-699	15-699	15-699#	15-699#	15-699#	15-699#	15-699#	15-705	15-705	15-705	15-705	15-705#	15-705#
15-705#	15-705#	15-705#	15-713	15-713	15-713	15-713	15-713#	15-713#	15-713#	15-713#	15-713#	15-719	15-719
15-719	15-719	15-719#	15-719#	15-719#	15-719#	15-719#	15-865	15-865	15-865	15-865	15-865#	15-865#	15-865#
15-865#	15-865#	15-870	15-870	15-870	15-870	15-870#	15-870#	15-870#	15-870#	15-870#	15-889	15-889	15-889
15-889	15-889#	15-889#	15-889#	15-889#	15-889#	15-895	15-895	15-895	15-895	15-895#	15-895#	15-895#	15-895#
15-895#	15-904	15-904	15-904	15-904	15-904#	15-904#	15-904#	15-904#	15-904#	15-910	15-910	15-910	15-910
15-910#	15-910#	15-910#	15-910#	15-910#	15-919	15-919	15-919	15-919	15-919#	15-919#	15-919#	15-919#	15-919#
15-925	15-925	15-925	15-925	15-925#	15-925#	15-925#	15-925#	15-925#	15-934	15-934	15-934	15-934	15-934#
15-934#	15-934#	15-934#	15-934#	15-940	15-940	15-940	15-940	15-940#	15-940#	15-940#	15-940#	15-940#	15-:72
15-:72	15-:72	15-:72	15-:72#	15-:72#	15-:72#	15-:72#	15-:72#	15-:78	15-:78	15-:78	15-:78	15-:78#	15-:78#
15-:78#	15-:78#	15-:78#	16-101	16-101	16-101	16-101	16-101	16-101	16-101	16-101#	16-101#	16-101#	16-101#
16-101#	16-101#	16-102	16-102#	16-107	16-107	16-107	16-107	16-107	16-107	16-107	16-107#	16-107#	16-107#
16-107#	16-107#	16-107#	16-108	16-108	16-108	16-108	16-108	16-108#	16-108#	16-108#	16-108#	16-109	16-109
16-109	16-109	16-109	16-109	16-109	16-109#	16-109#	16-109#	16-109#	16-109#	16-109#	16-110	16-110	16-110
16-110	16-110	16-110	16-110	16-110#	16-110#	16-110#	16-110#	16-110#	16-110#	16-111	16-111	16-111	16-111
16-111	16-111	16-111	16-111#	16-111#	16-111#	16-111#	16-111#	16-111#	16-112	16-112	16-112	16-112	16-112
16-112	16-112	16-112	16-112	16-112#	16-112#	16-112#	16-112#	16-112#	16-112#	16-112#	16-112#	16-113	16-113
16-113	16-113	16-113	16-113	16-113#	16-113#	16-113#	16-113#	16-113#	16-114	16-114	16-114	16-114	16-114
16-114	16-114	16-114	16-114	16-114#	16-114#	16-114#	16-114#	16-114#	16-114#	16-114#	16-114#	16-115	16-115#
16-122	16-122	16-122	16-122	16-122	16-122	16-122	16-122#	16-122#	16-122#	16-122#	16-122#	16-122#	16-123
16-123	16-123	16-123	16-123	16-123#	16-123#	16-123#	16-123#	16-124	16-124	16-124	16-124	16-124	16-124
16-124	16-124#	16-124#	16-124#	16-124#	16-124#	16-124#	16-125	16-125	16-125	16-125	16-125	16-125	16-125
16-125#	16-125#	16-125#	16-125#	16-125#	16-125#	16-126	16-126	16-126	16-126	16-126	16-126	16-126	16-126#
16-126#	16-126#	16-126#	16-126#	16-126#	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127	16-127
16-127#	16-127#	16-127#	16-127#	16-127#	16-127#	16-127#	16-127#	16-128	16-128	16-128	16-128	16-128	16-128
16-128#	16-128#	16-128#	16-128#	16-128#	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129	16-129
16-129#	16-129#	16-129#	16-129#	16-129#	16-129#	16-129#	16-129#	16-130	16-130	16-130	16-130	16-130	16-130
16-130	16-130#	16-130#	16-130#	16-130#	16-130#	16-130#	16-131	16-131	16-131	16-131	16-131	16-131	16-131
16-131	16-131	16-131#	16-131#	16-131#	16-131#	16-131#	16-131#	16-131#	16-131#	16-132	16-132	16-132	16-132
16-132	16-132	16-132#	16-132#	16-132#	16-132#	16-132#	16-133	16-133	16-133	16-133	16-133	16-133	16-133
16-133	16-133	16-133#	16-133#	16-133#	16-133#	16-133#	16-133#	16-133#	16-133#	16-134	16-134#	16-141	16-141
16-141	16-141	16-141	16-141	16-141#	16-141#	16-141#	16-141#	16-141#	16-142	16-142	16-142	16-142	16-142
16-142	16-142	16-142#	16-142#	16-142#	16-142#	16-142#	16-142#	16-143	16-143	16-143	16-143	16-143	16-143#
16-143#	16-143#	16-143#	16-144	16-144	16-144	16-144	16-144	16-144	16-144	16-144#	16-144#	16-144#	16-144#
16-144#	16-144#	16-145	16-145	16-145	16-145	16-145	16-145	16-145	16-145#	16-145#	16-145#	16-145#	16-145#
16-145#	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146	16-146#	16-146#	16-146#	16-146#
16-146#	16-146#	16-146#	16-146#	16-147	16-147	16-147	16-147	16-147	16-147	16-147#	16-147#	16-147#	16-147#
16-147#	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148	16-148#	16-148#	16-148#	16-148#
16-148#	16-148#	16-148#	16-148#	16-149	16-149	16-149	16-149	16-149	16-149	16-149	16-149#	16-149#	16-149#
16-149#	16-149#	16-149#	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150	16-150#	16-150#
16-150#	16-150#	16-150#	16-150#	16-150#	16-150#	16-151	16-151	16-151	16-151	16-151	16-151	16-151#	16-151#
16-151#	16-151#	16-151#	16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152	16-152#	16-152#

16-152#	16-152#	16-152#	16-152#	16-152#	16-152#	16-153	16-153#	16-160	16-160	16-160	16-160	16-160	16-160
16-160	16-160#	16-160#	16-160#	16-160#	16-160#	16-160#	16-161	16-161	16-161	16-161	16-161	16-161	16-161
16-161#	16-161#	16-161#	16-161#	16-161#	16-161#	16-162	16-162	16-162	16-162	16-162	16-162#	16-162#	16-162#
16-162#	16-163	16-163	16-163	16-163	16-163	16-163	16-163	16-163#	16-163#	16-163#	16-163#	16-163#	16-163#
16-164	16-164	16-164	16-164	16-164	16-164	16-164	16-164#	16-164#	16-164#	16-164#	16-164#	16-164#	16-165
16-165	16-165	16-165	16-165	16-165	16-165	16-165#	16-165#	16-165#	16-165#	16-165#	16-165#	16-166	16-166
16-166	16-166	16-166	16-166	16-166	16-166	16-166	16-166#	16-166#	16-166#	16-166#	16-166#	16-166#	16-166#
16-166#	16-167	16-167	16-167	16-167	16-167	16-167	16-167#	16-167#	16-167#	16-167#	16-167#	16-168	16-168
16-168	16-168	16-168	16-168	16-168	16-168	16-168	16-168#	16-168#	16-168#	16-168#	16-168#	16-168#	16-168#
16-168#	16-169	16-169	16-169	16-169	16-169	16-169	16-169	16-169#	16-169#	16-169#	16-169#	16-169#	16-169#
16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170	16-170#	16-170#	16-170#	16-170#	16-170#
16-170#	16-170#	16-170#	16-171	16-171	16-171	16-171	16-171	16-171	16-171#	16-171#	16-171#	16-171#	16-171#
16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172	16-172#	16-172#	16-172#	16-172#	16-172#
16-172#	16-172#	16-172#	16-173	16-173#	16-180	16-180	16-180	16-180	16-180	16-180	16-180#	16-180#	16-180#
16-180#	16-180#	16-181	16-181	16-181	16-181	16-181	16-181	16-181	16-181#	16-181#	16-181#	16-181#	16-181#
16-181#	16-182	16-182	16-182	16-182	16-182	16-182#	16-182#	16-182#	16-182#	16-183	16-183	16-183	16-183
16-183	16-183	16-183	16-183#	16-183#	16-183#	16-183#	16-183#	16-183#	16-184	16-184	16-184	16-184	16-184
16-184	16-184	16-184#	16-184#	16-184#	16-184#	16-184#	16-184#	16-185	16-185	16-185	16-185	16-185	16-185
16-185	16-185	16-185	16-185#	16-185#	16-185#	16-185#	16-185#	16-185#	16-185#	16-185#	16-186	16-186	16-186
16-186	16-186	16-186	16-186#	16-186#	16-186#	16-186#	16-186#	16-187	16-187	16-187	16-187	16-187	16-187
16-187	16-187	16-187	16-187#	16-187#	16-187#	16-187#	16-187#	16-187#	16-187#	16-187#	16-188	16-188	16-188
16-188	16-188	16-188	16-188	16-188#	16-188#	16-188#	16-188#	16-188#	16-188#	16-189	16-189	16-189	16-189
16-189	16-189	16-189	16-189	16-189	16-189#	16-189#	16-189#	16-189#	16-189#	16-189#	16-189#	16-189#	16-190
16-190	16-190	16-190	16-190	16-190	16-190#	16-190#	16-190#	16-190#	16-190#	16-191	16-191	16-191	16-191
16-191	16-191	16-191	16-191	16-191	16-191#	16-191#	16-191#	16-191#	16-191#	16-191#	16-191#	16-191#	16-192
16-192#	16-199	16-199	16-199	16-199	16-199	16-199	16-199	16-199#	16-199#	16-199#	16-199#	16-199#	16-199#
16-200	16-200	16-200	16-200	16-200	16-200#	16-200#	16-200#	16-200#	16-201	16-201	16-201	16-201	16-201
16-201	16-201	16-201#	16-201#	16-201#	16-201#	16-201#	16-201#	16-202	16-202	16-202	16-202	16-202	16-202
16-202	16-202#	16-202#	16-202#	16-202#	16-202#	16-202#	16-203	16-203	16-203	16-203	16-203	16-203	16-203
16-203	16-203	16-203#	16-203#	16-203#	16-203#	16-203#	16-203#	16-203#	16-203#	16-204	16-204	16-204	16-204
16-204	16-204	16-204#	16-204#	16-204#	16-204#	16-204#	16-205	16-205	16-205	16-205	16-205	16-205	16-205
16-205	16-205	16-205#	16-205#	16-205#	16-205#	16-205#	16-205#	16-205#	16-205#	16-206	16-206	16-206	16-206
16-206	16-206	16-206	16-206#	16-206#	16-206#	16-206#	16-206#	16-206#	16-207	16-207	16-207	16-207	16-207
16-207	16-207	16-207	16-207	16-207#	16-207#	16-207#	16-207#	16-207#	16-207#	16-207#	16-207#	16-208	16-208
16-208	16-208	16-208	16-208	16-208#	16-208#	16-208#	16-208#	16-208#	16-209	16-209	16-209	16-209	16-209
16-209	16-209	16-209	16-209	16-209#	16-209#	16-209#	16-209#	16-209#	16-209#	16-209#	16-209#	16-210	16-210#
16-217	16-217	16-217	16-217	16-217	16-217	16-217#	16-217#	16-217#	16-217#	16-217#	16-218	16-218	16-218
16-218	16-218	16-218	16-218	16-218#	16-218#	16-218#	16-218#	16-218#	16-218#	16-219	16-219	16-219	16-219
16-219	16-219#	16-219#	16-219#	16-219#	16-220	16-220	16-220	16-220	16-220	16-220	16-220	16-220#	16-220#
16-220#	16-220#	16-220#	16-220#	16-221	16-221	16-221	16-221	16-221	16-221	16-221	16-221#	16-221#	16-221#
16-221#	16-221#	16-221#	16-222	16-222	16-222	16-222	16-222	16-222	16-222	16-222#	16-222#	16-222#	16-222#
16-222#	16-222#	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223	16-223#	16-223#	16-223#
16-223#	16-223#	16-223#	16-223#	16-223#	16-224	16-224	16-224	16-224	16-224	16-224	16-224#	16-224#	16-224#
16-224#	16-224#	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225	16-225#	16-225#	16-225#
16-225#	16-225#	16-225#	16-225#	16-225#	16-226	16-226	16-226	16-226	16-226	16-226	16-226	16-226#	16-226#
16-226#	16-226#	16-226#	16-226#	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227#
16-227#	16-227#	16-227#	16-227#	16-227#	16-227#	16-227#	16-228	16-228	16-228	16-228	16-228	16-228	16-228#
16-228#	16-228#	16-228#	16-228#	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229	16-229#
16-229#	16-229#	16-229#	16-229#	16-229#	16-229#	16-229#	16-230	16-230#	16-237	16-237	16-237	16-237	16-237
16-237	16-237	16-237#	16-237#	16-237#	16-237#	16-237#	16-237#	16-238	16-238	16-238	16-238	16-238	16-238#
16-238#	16-238#	16-238#	16-239	16-239	16-239	16-239	16-239	16-239	16-239	16-239#	16-239#	16-239#	16-239#
16-239#	16-239#	16-240	16-240	16-240	16-240	16-240	16-240	16-240	16-240#	16-240#	16-240#	16-240#	16-240#
16-240#	16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241	16-241#	16-241#	16-241#	16-241#
16-241#	16-241#	16-241#	16-241#	16-242	16-242	16-242	16-242	16-242	16-242	16-242#	16-242#	16-242#	16-242#
16-242#	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243	16-243#	16-243#	16-243#	16-243#
16-243#	16-243#	16-243#	16-243#	16-244	16-244#	17-11	17-11#	19-25	19-25	19-25#	19-25#	19-26	19-26#
19-28	19-28	19-28#	19-28#	19-29	19-29#	19-31	19-31	19-31#	19-31#	19-32	19-32#	19-34	19-34

```
        19-34#   19-34#   19-35    19-35#   19-51    19-51    19-51    19-51#   19-51#   19-51#   19-52    19-52#   19-74    19-74#
        20-9     20-9     20-9#    20-9#    20-16    20-16    20-16#   20-16#   20-19    20-19#   21-11    21-11#   22-10    22-10#
        22-12    22-12    22-12    22-12    22-12    22-12    22-12#   22-12#   22-12#   22-12#   22-12#   22-13    22-13#   23-10
        23-10#   24-16    24-16    24-16#   24-16#   24-24    24-24    24-24    24-24    24-24#   24-24#   24-24#   24-24#   24-24#
        24-27    24-27#   24-50    24-50    24-50    24-50    24-50#   24-50#   24-50#   24-50#   24-50#   24-52    24-52#   24-72
        24-72#   24-84    24-84    24-84    24-84    24-84#   24-84#   24-84#   24-84#   24-84#   24-86    24-86#   24-90    24-90#
        24-115   24-115   24-115   24-115   24-115#  24-115#  24-115#  24-115#  24-115#  24-117   24-117#  24-133   24-133#  24-148
        24-148   24-148   24-148   24-148#  24-148#  24-148#  24-148#  24-148#  24-150   24-150#  24-184   24-184   24-184   24-184
        24-184#  24-184#  24-184#  24-184#  24-184#  24-186   24-186#  24-226   24-226   24-226   24-226   24-226#  24-226#  24-226#
        24-226#  24-226#  24-227   24-227   24-227#  24-227#  24-232   24-232#  24-278   24-278   24-278   24-278   24-278#  24-278#
        24-278#  24-278#  24-278#  24-279   24-279   24-279#  24-279#  24-286   24-286#  24-303   24-303#  24-314   24-314   24-314
        24-314   24-314#  24-314#  24-314#  24-314#  24-314#  24-316   24-316#  24-320   24-320#  24-337   24-337#  24-348   24-348
        24-348   24-348   24-348#  24-348#  24-348#  24-348#  24-348#  24-350   24-350#  24-354   24-354#  24-371   24-371#  24-382
        24-382   24-382   24-382   24-382#  24-382#  24-382#  24-382#  24-382#  24-384   24-384#  24-388   24-388#  24-405   24-405#
        24-416   24-416   24-416   24-416   24-416#  24-416#  24-416#  24-416#  24-416#  24-418   24-418#  24-422   24-422#  24-456
        24-456   24-456   24-456   24-456   24-456   24-456#  24-456#  24-456#  24-456#  24-456#  24-480   24-480   24-480   24-480
        24-480#  24-480#  24-480#  24-480#  24-480#  24-482   24-482   24-482   24-482   24-482   24-482#  24-482#  24-482#  24-482#
        24-499   24-499   24-499   24-499   24-499#  24-499#  24-499#  24-499#  24-499#  24-501   24-501   24-501   24-501   24-501
        24-501#  24-501#  24-501#  24-501#  24-518   24-518   24-518   24-518   24-518#  24-518#  24-518#  24-518#  24-518#  24-520
        24-520   24-520   24-520   24-520   24-520#  24-520#  24-520#  24-520#  24-523   24-523#  24-551   24-551   24-551   24-551
        24-551#  24-551#  24-551#  24-551#  24-551#  24-552   24-552   24-552#  24-552#  24-565   24-565   24-565   24-565   24-565#
        24-565#  24-565#  24-565#  24-565#  24-566   24-566   24-566#  24-566#  24-577   24-577   24-577   24-577   24-577#  24-577#
        24-577#  24-577#  24-577#  24-578   24-578   24-578#  24-578#  24-588   24-588   24-588   24-588   24-588#  24-588#  24-588#
        24-588#  24-588#  24-589   24-589   24-589#  24-589#  24-594   24-594#  24-637   24-637   24-637   24-637   24-637#  24-637#
        24-637#  24-637#  24-637#  24-638   24-638   24-638#  24-638#  24-647   24-647   24-647   24-647   24-647#  24-647#  24-647#
        24-647#  24-647#  24-648   24-648   24-648#  24-648#  24-655   24-655#  24-680   24-680#  24-694   24-694   24-694   24-694
        24-694#  24-694#  24-694#  24-694#  24-694#  24-695   24-695   24-695#  24-695#  24-704   24-704   24-704   24-704   24-704#
        24-704#  24-704#  24-704#  24-704#  24-706   24-706#  24-710   24-710#  24-731   24-731#  24-745   24-745   24-745   24-745
        24-745#  24-745#  24-745#  24-745#  24-745#  24-746   24-746   24-746#  24-746#  24-754   24-754   24-754   24-754   24-754#
        24-754#  24-754#  24-754#  24-754#  24-756   24-756#  24-760   24-760#  24-778   24-778#  24-792   24-792   24-792   24-792
        24-792#  24-792#  24-792#  24-792#  24-792#  24-793   24-793   24-793#  24-793#  24-801   24-801   24-801   24-801   24-801#
        24-801#  24-801#  24-801#  24-801#  24-803   24-803#  24-807   24-807#  24-838   24-838#  24-855   24-855   24-855   24-855
        24-855#  24-855#  24-855#  24-855#  24-855#  24-856   24-856   24-856#  24-856#  24-865   24-865   24-865   24-865   24-865#
        24-865#  24-865#  24-865#  24-865#  24-867   24-867#  24-872   24-872#  24-898   24-898#  24-900   24-900#  24-913   24-913
        24-913   24-913   24-913#  24-913#  24-913#  24-913#  24-913#  24-915   24-915#  24-920   24-920#  24-924   24-924#  24-936
        24-936   24-936   24-936   24-936#  24-936#  24-936#  24-936#  24-936#  24-938   24-938#  24-939   24-939#  24-960   24-960#
        24-983   24-983   24-983   24-983   24-983#  24-983#  24-983#  24-983#  24-983#  24-984   24-984   24-984#  24-984#  24-993
        24-993   24-993   24-993   24-993#  24-993#  24-993#  24-993#  24-993#  24-994   24-994   24-994#  24-994#  24-999   24-999#
        24-:00   24-:00#  24-:98   24-:98   24-:98   24-:98   24-:98#  24-:98#  24-:98#  24-:98#  24-:98#  24-:99   24-:99   24-:99#
        24-:99#  24-;21   24-;21   24-;21   24-;21   24-;21#  24-;21#  24-;21#  24-;21#  24-;21#  24-;22   24-;22   24-;22#  24-;22#
        24-;32   24-;32#  24-;81   24-;81#  24-<27   24-<27#  24-<83   24-<83   24-<83   24-<83   24-<83#  24-<83#  24-<83#  24-<83#
        24-<83#  24-<87   24-<87#  24-=26   24-=26   24-=26   24-=26   24-=26#  24-=26#  24-=26#  24-=26#  24-=26#  24-=35   24-=35
        24-=35   24-=35   24-=35#  24-=35#  24-=35#  24-=35#  24-=35#  24-=47   24-=47   24-=47   24-=47   24-=47#  24-=47#  24-=47#
        24-=47#  24-=47#  24-=66   24-=66   24-=66   24-=66   24-=66#  24-=66#  24-=66#  24-=66#  24-=66#  24-=74   24-=74   24-=74
        24-=74   24-=74#  24-=74#  24-=74#  24-=74#  24-=74#  24-=76   24-=76#  24->32   24->32#  24-?12   24-?12#  24-?47   24-?47#
        24-?97   24-?97   24-?97   24-?97   24-?97#  24-?97#  24-?97#  24-?97#  24-?97#  24-@12   24-@12   24-@12   24-@12   24-@12#
        24-@12#  24-@12#  24-@12#  24-@12#  24-@27   24-@27#  24-@95   24-@95   24-@95   24-@95   24-@95#  24-@95#  24-@95#  24-@95#
        24-@95#  24-A10   24-A10#  24-A60   24-A60   24-A60   24-A60   24-A60#  24-A60#  24-A60#  24-A60#  24-A60#  24-A73   24-A73#
        24-B41   24-B41   24-B41   24-B41   24-B41#  24-B41#  24-B41#  24-B41#  24-B41#  24-B56   24-B56#  24-B85   24-B85   24-B85
        24-B85   24-B85#  24-B85#  24-B85#  24-B85#  24-B85#  24-B88   24-B88#  24-C89   24-C89   24-C89   24-C89   24-C89#  24-C89#
        24-C89#  24-C89#  24-C89#  24-D34   24-D34#  24-E35   24-E35#  24-E87   24-E87#  24-F55   24-F55#  24-F87   24-F87#  24-G40
        24-G40#  24-G90   24-G90#  24-H60   24-H60#  25-14    25-14#   25-16    25-16    25-16    25-16    25-16#   25-17    25-17
        25-17    25-17    25-17#   25-18    25-18    25-18    25-18    25-18    25-18#   25-19    25-19    25-19    25-19    25-19
        25-19#   25-21    25-21#   26-13    26-13#   26-16    26-16#   26-39    26-39    26-39    26-39#   26-39#   26-39#
M$GNLS   1-C13#   7-18#   24-86    24-86#   24-316   24-316#  24-350   24-350#  24-384   24-384#  24-418   24-418#  24-706   24-706#
        24-756   24-756#  24-803   24-803#  24-867   24-867#  24-915   24-915#
M$GNSU   1-B98#   7-18#   24-898   24-898#  24-924   24-924#  24-960   24-960#
```

```
MSGNTA    1-B90#    7-18#     10-23     10-23#    11-11     11-11#    16-102    16-102#   16-115    16-115#   16-134    16-134#   16-153    16-153#
          16-173    16-173#   16-192    16-192#   16-210    16-210#   16-230    16-230#   16-244    16-244#   17-11     17-11#    19-74     19-74#
          20-19     20-19#    21-11     21-11#    22-13     22-13#    23-10     23-10#    24-27     24-27#    24-52     24-52#    24-90     24-90#
          24-117    24-117#   24-150    24-150#   24-186    24-186#   24-232    24-232#   24-286    24-286#   24-320    24-320#   24-354    24-354#
          24-388    24-388#   24-422    24-422#   24-523    24-523#   24-594    24-594#   24-655    24-655#   24-710    24-710#   24-760    24-760#
          24-807    24-807#   24-872    24-872#   24-920    24-920#   24-938    24-938#   24-939    24-939#   24-999    24-999#   24-:00    24-:00#
          24-;32    24-;32#   24-;81    24-;81#   24-<27    24-<27#   24-<87    24-<87#   24-=76    24-=76#   24->32    24->32#   24-?12    24-?12#
          24-?47    24-?47#   24-@27    24-@27#   24-A10    24-A10#   24-A73    24-A73#   24-B56    24-B56#   24-B88    24-B88#   24-D34    24-D34#
          24-E35    24-E35#   24-E87    24-E87#   24-F55    24-F55#   24-F87    24-F87#   24-G40    24-G40#   24-G90    24-G90#   24-H60    24-H60#
          25-21     25-21#    26-16     26-16#
MSGNTE    1-B94#    7-18#     24-14     24-14#    24-39     24-39#    24-67     24-67#    24-101    24-101#   24-128    24-128#   24-166    24-166#
          24-203    24-203#   24-247    24-247#   24-298    24-298#   24-332    24-332#   24-366    24-366#   24-400    24-400#   24-449    24-449#
          24-539    24-539#   24-610    24-610#   24-676    24-676#   24-727    24-727#   24-774    24-774#   24-832    24-832#   24-889    24-889#
          24-957    24-957#   24-:22    24-:22#   24-;52    24-;52#   24-<01    24-<01#   24-<49    24-<49#   24-=07    24-=07#   24-=91    24-=91#
          24->48    24->48#   24-?27    24-?27#   24-?63    24-?63#   24-@43    24-@43#   24-A26    24-A26#   24-A89    24-A89#   24-B69    24-B69#
          24-C05    24-C05#   24-D58    24-D58#   24-E53    24-E53#   24-F05    24-F05#   24-F70    24-F70#   24-G08    24-G08#   24-G59    24-G59#
          24-H13    24-H13#
MSHAPT    1-A39#    7-18#     8-17      8-17#
MSHNAP    1-B24#    7-18#     8-17      8-17#
MSINCR    1-D26#    7-18#     7-24      7-24#     10-9      10-9      10-9#     10-9#     11-8      11-8      11-8#     11-8#     15-277#   15-283#
          15-293#   15-299#   15-406#   15-412#   15-570#   15-602#   15-608#   15-618#   15-624#   15-659#   15-665#   15-699#   15-705#   15-713#
          15-719#   15-865#   15-870#   15-889#   15-895#   15-904#   15-910#   15-919#   15-925#   15-934#   15-940#   15-:72#   15-:78#   16-100
          16-100    16-100#   16-100#   16-101#   16-102#   16-106    16-106    16-106#   16-106#   16-107#   16-108#   16-109#   16-110#   16-111#
          16-112#   16-113#   16-114#   16-115#   16-121    16-121    16-121#   16-121#   16-122#   16-123#   16-124#   16-125#   16-126#   16-127#
          16-128#   16-129#   16-130#   16-131#   16-132#   16-133#   16-134#   16-140    16-140    16-140#   16-140#   16-141#   16-142#   16-143#
          16-144#   16-145#   16-146#   16-147#   16-148#   16-149#   16-150#   16-151#   16-152#   16-153#   16-159    16-159    16-159#   16-159#
          16-160#   16-161#   16-162#   16-163#   16-164#   16-165#   16-166#   16-167#   16-168#   16-169#   16-170#   16-171#   16-172#   16-173#
          16-179    16-179    16-179#   16-179#   16-180#   16-181#   16-182#   16-183#   16-184#   16-185#   16-186#   16-187#   16-188#   16-189#
          16-190#   16-191#   16-192#   16-198    16-198    16-198#   16-198#   16-199#   16-200#   16-201#   16-202#   16-203#   16-204#   16-205#
          16-206#   16-207#   16-208#   16-209#   16-210#   16-216    16-216    16-216#   16-216#   16-217#   16-218#   16-219#   16-220#   16-221#
          16-222#   16-223#   16-224#   16-225#   16-226#   16-227#   16-228#   16-229#   16-230#   16-236    16-236    16-236#   16-236#   16-237#
          16-238#   16-239#   16-240#   16-241#   16-242#   16-243#   16-244#   17-9      17-9      17-9#     17-9#     17-11#    18-8      18-8
          18-8#     18-8#     19-8      19-8      19-8#     19-8#     19-25#    19-28#    19-31#    19-34#    19-51#    19-74#    20-7      20-7
          20-7#     20-7#     20-9#     20-16#    20-19#    21-8      21-8      21-8#     21-8#     21-11#    22-8      22-8      22-8#     22-8#
          22-10#    22-12#    22-13#    23-9      23-9      23-9#     23-9#     23-10#    24-14     24-14     24-14     24-14#    24-14#    24-14#
          24-16#    24-24#    24-27#    24-39     24-39     24-39     24-39#    24-39#    24-39#    24-50#    24-52#    24-67     24-67     24-67
          24-67#    24-67#    24-67#    24-72     24-72     24-72     24-72#    24-72#    24-72#    24-72#    24-84#    24-86#    24-90#    24-101
          24-101    24-101    24-101#   24-101#   24-101#   24-115#   24-117#   24-128    24-128    24-128    24-128#   24-128#   24-128#   24-133#
          24-148#   24-150#   24-166    24-166    24-166    24-166#   24-166#   24-166#   24-184#   24-186#   24-203    24-203    24-203    24-203#
          24-203#   24-203#   24-226#   24-227#   24-232#   24-247    24-247    24-247    24-247#   24-247#   24-247#   24-278#   24-279#   24-286#
          24-298    24-298    24-298    24-298#   24-298#   24-298#   24-303    24-303    24-303    24-303#   24-303#   24-303#   24-303#   24-314#
          24-316#   24-320#   24-332    24-332    24-332    24-332#   24-332#   24-332#   24-337    24-337    24-337    24-337#   24-337#   24-337#
          24-337#   24-348#   24-350#   24-354#   24-366    24-366    24-366    24-366#   24-366#   24-366#   24-371    24-371    24-371    24-371#
          24-371#   24-371#   24-371#   24-382#   24-384#   24-388#   24-400    24-400    24-400    24-400#   24-400#   24-400#   24-405    24-405
          24-405    24-405#   24-405#   24-405#   24-405#   24-416#   24-418#   24-422#   24-449    24-449    24-449    24-449#   24-449#   24-449#
          24-456#   24-480#   24-482#   24-499#   24-501#   24-518#   24-520#   24-523#   24-539    24-539    24-539    24-539#   24-539#   24-539#
          24-551#   24-552#   24-565#   24-566#   24-577#   24-578#   24-588#   24-589#   24-594#   24-610    24-610    24-610    24-610#   24-610#
          24-610#   24-637#   24-638#   24-647#   24-648#   24-655#   24-676    24-676    24-676    24-676#   24-676#   24-676#   24-680    24-680
          24-680    24-680#   24-680#   24-680#   24-680#   24-694#   24-695#   24-704#   24-706#   24-710#   24-727    24-727    24-727    24-727#
          24-727#   24-727#   24-731    24-731    24-731    24-731#   24-731#   24-731#   24-731#   24-745#   24-746#   24-754#   24-756#   24-760#
          24-774    24-774    24-774    24-774#   24-774#   24-774#   24-778    24-778    24-778    24-778#   24-778#   24-778#   24-778#   24-792#
          24-793#   24-801#   24-803#   24-807#   24-832    24-832    24-832    24-832#   24-832#   24-832#   24-838    24-838    24-838    24-838#
          24-838#   24-838#   24-838#   24-855#   24-856#   24-865#   24-867#   24-872#   24-889    24-889    24-889    24-889#   24-889#   24-889#
          24-898    24-898    24-898    24-898#   24-898#   24-898#   24-900    24-900    24-900    24-900#   24-900#   24-900#   24-900#   24-913#
          24-915#   24-920#   24-924    24-924    24-924    24-924#   24-924#   24-924#   24-936#   24-938#   24-939#   24-957    24-957    24-957
          24-957#   24-957#   24-957#   24-960    24-960    24-960    24-960#   24-960#   24-960#   24-983#   24-984#   24-993#   24-994#   24-999#
          24-:00#   24-:22    24-:22    24-:22    24-:22#   24-:22#   24-:22#   24-:98#   24-:99#   24-;21#   24-;22#   24-:32#   24-;52    24-;52
```

```
          24-;52    24-;52#   24-;52#   24-;52#   24-;81#   24-<01    24-<01    24-<01    24-<01#   24-<01#   24-<01#   24-<27#   24-<49    24-<49
          24-<49    24-<49#   24-<49#   24-<49#   24-<83#   24-<87#   24-=07    24-=07    24-=07    24-=07#   24-=07#   24-=07#   24-=26#   24-=35#
          24-=47#   24-=66#   24-=74#   24-=76#   24-=91    24-=91    24-=91    24-=91#   24-=91#   24-=91#   24->32#   24->48    24->48    24->48
          24->48#   24->48#   24->48#   24-?12#   24-?27    24-?27    24-?27    24-?27#   24-?27#   24-?27#   24-?47#   24-?63    24-?63    24-?63
          24-?63#   24-?63#   24-?63#   24-?97#   24-@12#   24-@27#   24-@43    24-@43    24-@43    24-@43#   24-@43#   24-@43#   24-@95#   24-A10#
          24-A26    24-A26    24-A26    24-A26#   24-A26#   24-A26#   24-A60#   24-A73#   24-A89    24-A89    24-A89    24-A89#   24-A89#   24-A89#
          24-B41#   24-B56#   24-B69    24-B69    24-B69    24-B69#   24-B69#   24-B69#   24-B85#   24-B88#   24-C05    24-C05    24-C05    24-C05#
          24-C05#   24-C05#   24-C89#   24-D34#   24-D58    24-D58    24-D58    24-D58#   24-D58#   24-D58#   24-E35#   24-E53    24-E53    24-E53
          24-E53#   24-E53#   24-E53#   24-E87#   24-F05    24-F05    24-F05    24-F05#   24-F05#   24-F05#   24-F55#   24-F70    24-F70    24-F70
          24-F70#   24-F70#   24-F70#   24-F87#   24-G08    24-G08    24-G08    24-G08#   24-G08#   24-G08#   24-G40#   24-G59    24-G59    24-G59
          24-G59#   24-G59#   24-G59#   24-G9C#   24-H13    24-H13    24-H13    24-H13#   24-H13#   24-H13#   24-H60#   25-14     25-14     25-14#
          25-14#    26-13     26-13     26-13#    26-13#
M$IOSE     1-A00#    7-18#
M$LDRO     1-C42#    7-18#    19-25     19-25#    19-28     19-28#    19-31     19-31#    19-34     19-34#    19-51     19-51#    20-9      20-9#
          20-16     20-16#    24-16     24-16#
M$MASK     1-@71#    7-18#
M$MCHI     1-4#      7-18      7-18#     7-18#
M$MCLO     1-@24#    7-18      7-18#     7-18#
M$MSK1     1-@77#    7-18#
M$POP      1-B81#    7-18#    10-23     10-23#    11-11     11-11#    16-102    16-102#   16-115    16-[illegible] 16-134 16-134#   16-153    16-153#
          16-173    16-173#   16-192    16-192#   16-210    16-210#   16-230    16-230#   16-244    16-[illegible] 17-11  17-11#    18-12     18-12#
          19-74     19-74#    20-19     20-19#    21-11     21-11#    22-13     22-13#    23-10     23-10#    24-27     24-27#    24-52     24-52#
          24-86     24-86     24-86#    24-90     24-90#    24-117    24-117#   24-150    24-150#   24-186    24-186#   24-232    24-232#   24-286
          24-286#   24-316    24-316    24-316#   24-320    24-320#   24-350    24-350    24-350#   24-354    24-354#   24-384    24-384    24-384#
          24-388    24-388#   24-418    24-418    24-418#   24-422    24-422#   24-523    24-523#   24-594    24-594#   24-655    24-655#   24-706
          24-706    24-706#   24-710    24-710#   24-756    24-756    24-756#   24-760    24-760#   24-803    24-803    24-803#   24-807    24-807#
          24-867    24-867    24-867#   24-872    24-872#   24-915    24-915    24-915#   24-920    24-920#   24-938    24-938#   24-939    24-939#
          24-999    24-999#   24-:00    24-:00#   24-;32    24-;32#   24-;81    24-;81#   24-<27    24-<27#   24-<87    24-<87#   24-=76    24-=76#
          24->32    24->32#   24-?12    24-?12#   24-?47    24-?47#   24-@27    24-@27#   24-A10    24-A10#   24-A73    24-A73#   24-B56    24-B56#
          24-B88    24-B88#   24-D34    24-D34#   24-E35    24-E35#   24-E87    24-E87#   24-F55    24-F55#   24-F87    24-F87#   24-G40    24-G40#
          24-G90    24-G90#   24-H60    24-H60#   25-21     25-21#    26-16     26-16#    26-37     26-37#
M$PRIN     1-@36#    7-18#    16-101    16-101#   16-107    16-107#   16-108    16-108#   16-109    16-109#   16-110    16-110#   16-111    16-111#
          16-112    16-112#   16-113    16-113#   16-114    16-114#   16-122    16-122#   16-123    16-123#   16-124    16-124#   16-125    16-125#
          16-126    16-126#   16-127    16-127#   16-128    16-128#   16-129    16-129#   16-130    16-130#   16-131    16-131#   16-132    16-132#
          16-133    16-133#   16-141    16-141#   16-142    16-142#   16-143    16-143#   16-144    16-144#   16-145    16-145#   16-146    16-146#
          16-147    16-147#   16-148    16-148#   16-149    16-149#   16-150    16-150#   16-151    16-151#   16-152    16-152#   16-160    16-160#
          16-161    16-161#   16-162    16-162#   16-163    16-163#   16-164    16-164#   16-165    16-165#   16-166    16-166#   16-167    16-167#
          16-168    16-168#   16-169    16-169#   16-170    16-170#   16-171    16-171#   16-172    16-172#   16-180    16-180#   16-181    16-181#
          16-182    16-182#   16-183    16-183#   16-184    16-184#   16-185    16-185#   16-186    16-186#   16-187    16-187#   16-188    16-188#
          16-189    16-189#   16-190    16-190#   16-191    16-191#   16-199    16-199#   16-200    16-200#   16-201    16-201#   16-202    16-202#
          16-203    16-203#   16-204    16-204#   16-205    16-205#   16-206    16-206#   16-207    16-207#   16-208    16-208#   16-209    16-209#
          16-217    16-217#   16-218    16-218#   16-219    16-219#   16-220    16-220#   16-221    16-221#   16-222    16-222#   16-223    16-223#
          16-224    16-224#   16-225    16-225#   16-226    16-226#   16-227    16-227#   16-228    16-228#   16-229    16-229#   16-237    16-237#
          16-238    16-238#   16-239    16-239#   16-240    16-240#   16-241    16-241#   16-242    16-242#   16-243    16-243#   22-12     22-12#
          24-456    24-456#   24-482    24-482#   24-501    24-501#   24-520    24-520#
M$PUSH     1-@31#    7-18#     7-24      7-24#    10-9      10-9#     11-8      11-8#    16-100    16-100#   16-106    16-106#   16-121    16-121#
          16-140    16-140#   16-159    16-159#   16-179    16-179#   16-198    16-198#   16-216    16-216#   16-236    16-236#   17-9      17-9#
          18-8      18-8#     19-8      19-8#     20-7      20-7#     21-8      21-8#     22-8      22-8#     23-9      23-9#     24-14     24-14#
          24-39     24-39#    24-67     24-67#    24-72     24-72     24-72#    24-101    24-101#   24-128    24-128#   24-166    24-166#   24-203
          24-203#   24-247    24-247#   24-298    24-298#   24-303    24-303    24-303#   24-332    24-332#   24-337    24-337    24-337#   24-366
          24-366#   24-371    24-371    24-371#   24-400    24-400#   24-405    24-405    24-405#   24-449    24-449#   24-539    24-539#   24-610
          24-610#   24-676    24-676#   24-680    24-680    24-680#   24-727    24-727#   24-731    24-731    24-731#   24-774    24-774#   24-778
          24-778    24-778#   24-832    24-832#   24-838    24-838    24-838#   24-889    24-889#   24-898    24-898#   24-900    24-900    24-900#
          24-924    24-924#   24-957    24-957#   24-960    24-960#   24-:22    24-:22#   24-;52    24-;52#   24-<01    24-<01#   24-<49    24-<49#
          24-=07    24-=07#   24-=91    24-=91#   24->48    24->48#   24-?27    24-?27#   24-?63    24-?63#   24-@43    24-@43#   24-A26    24-A26#
          24-A89    24-A89#   24-B69    24-B69#   24-C05    24-C05#   24-D58    24-D58#   24-E53    24-E53#   24-F05    24-F05#   24-F70    24-F70#
          24-G08    24-G08#   24-G59    24-G59#   24-H13    24-H13#   25-14     25-14#    26-13     26-13#
```

MSPUT	1-C72#	7-18#	16-101	16-101	16-101	16-101	16-101#	16-107	16-107	16-107	16-107	16-107#	16-108	16-108
	16-108#	16-109	16-109	16-109	16-109	16-109#	16-110	16-110	16-110	16-110	16-110#	16-111	16-111	16-111
	16-111	16-111#	16-112	16-112	16-112	16-112	16-112	16-112	16-112#	16-113	16-113	16-113	16-113#	16-114
	16-114	16-114	16-114	16-114	16-114	16-114#	16-122	16-122	16-122	16-122	16-122#	16-123	16-123	16-123#
	16-124	16-124	16-124	16-124	16-124#	16-125	16-125	16-125	16-125	16-125#	16-126	16-126	16-126	16-126
	16-126#	16-127	16-127	16-127	16-127	16-127	16-127	16-127#	16-128	16-128	16-128	16-128#	16-129	16-129
	16-129	16-129	16-129	16-129	16-129#	16-130	16-130	16-130	16-130	16-130#	16-131	16-131	16-131	16-131
	16-131	16-131	16-131#	16-132	16-132	16-132	16-132#	16-133	16-133	16-133	16-133	16-133	16-133	16-133#
	16-141	16-141	16-141	16-141#	16-142	16-142	16-142	16-142	16-142#	16-143	16-143	16-143#	16-144	16-144
	16-144	16-144	16-144#	16-145	16-145	16-145	16-145	16-145#	16-146	16-146	16-146	16-146	16-146	16-146
	16-146#	16-147	16-147	16-147	16-147#	16-148	16-148	16-148	16-148	16-148	16-148	16-148#	16-149	16-149
	16-149	16-149	16-149#	16-150	16-150	16-150	16-150	16-150	16-150	16-150#	16-151	16-151	16-151	16-151#
	16-152	16-152	16-152	16-152	16-152	16-152	16-152#	16-160	16-160	16-160	16-160	16-160#	16-161	16-161
	16-161	16-161	16-161#	16-162	16-162	16-162#	16-163	16-163	16-163	16-163	16-163#	16-164	16-164	16-164
	16-164	16-164#	16-165	16-165	16-165	16-165	16-165#	16-166	16-166	16-166	16-166	16-166	16-166	16-166#
	16-167	16-167	16-167	16-167#	16-168	16-168	16-168	16-168	16-168	16-168	16-168#	16-169	16-169	16-169
	16-169	16-169#	16-170	16-170	16-170	16-170	16-170	16-170	16-170#	16-171	16-171	16-171	16-171#	16-172
	16-172	16-172	16-172	16-172	16-172	16-172#	16-180	16-180	16-180	16-180#	16-181	16-181	16-181	16-181
	16-181#	16-182	16-182	16-182#	16-183	16-183	16-183	16-183	16-183#	16-184	16-184	16-184	16-184	16-184#
	16-185	16-185	16-185	16-185	16-185	16-185	16-185#	16-186	16-186	16-186	16-186#	16-187	16-187	16-187
	16-187	16-187	16-187	16-187#	16-188	16-188	16-188	16-188	16-188#	16-189	16-189	16-189	16-189	16-189
	16-189	16-189#	16-190	16-190	16-190	16-190#	16-191	16-191	16-191	16-191	16-191	16-191	16-191#	16-199
	16-199	16-199	16-199	16-199#	16-200	16-200	16-200#	16-201	16-201	16-201	16-201	16-201#	16-202	16-202
	16-202	16-202	16-202#	16-203	16-203	16-203	16-203	16-203	16-203	16-203#	16-204	16-204	16-204	16-204#
	16-205	16-205	16-205	16-205	16-205	16-205	16-205#	16-206	16-206	16-206	16-206	16-206#	16-207	16-207
	16-207	16-207	16-207	16-207	16-207#	16-208	16-208	16-208	16-208#	16-209	16-209	16-209	16-209	16-209
	16-209	16-209#	16-217	16-217	16-217	16-217#	16-218	16-218	16-218	16-218	16-218#	16-219	16-219	16-219#
	16-220	16-220	16-220	16-220	16-220#	16-221	16-221	16-221	16-221	16-221#	16-222	16-222	16-222	16-222
	16-222#	16-223	16-223	16-223	16-223	16-223	16-223	16-223#	16-224	16-224	16-224	16-224#	16-225	16-225
	16-225	16-225	16-225	16-225	16-225#	16-226	16-226	16-226	16-226	16-226#	16-227	16-227	16-227	16-227
	16-227	16-227	16-227#	16-228	16-228	16-228	16-228#	16-229	16-229	16-229	16-229	16-229	16-229	16-229#
	16-237	16-237	16-237	16-237	16-237#	16-238	16-238	16-238#	16-239	16-239	16-239	16-239	16-239#	16-240
	16-240	16-240	16-240	16-240#	16-241	16-241	16-241	16-241	16-241	16-241	16-241#	16-242	16-242	16-242
	16-242#	16-243	16-243	16-243	16-243	16-243	16-243	16-243#	22-12	22-12	22-12	22-12#	24-456	24-456
	24-456	24-456#	24-482	24-482	24-482#	24-501	24-501	24-501#	24-520	24-520	24-520#			
MSPUT1	1-C81#	7-18#	16-101	16-101	16-101	16-101	16-101#	16-101#	16-101#	16-101#	16-107	16-107	16-107	16-107
	16-107#	16-107#	16-107#	16-107#	16-108	16-108	16-108#	16-108#	16-109	16-109	16-109	16-109	16-109#	16-109#
	16-109#	16-109#	16-110	16-110	16-110	16-110	16-110#	16-110#	16-110#	16-110#	16-111	16-111	16-111	16-111
	16-111#	16-111#	16-111#	16-111#	16-112	16-112	16-112	16-112	16-112	16-112	16-112#	16-112#	16-112#	16-112#
	16-112#	16-112#	16-113	16-113	16-113	16-113#	16-113#	16-113#	16-114	16-114	16-114	16-114	16-114	16-114
	16-114#	16-114#	16-114#	16-114#	16-114#	16-114#	16-122	16-122	16-122	16-122	16-122#	16-122#	16-122#	16-122#
	16-123	16-123	16-123#	16-123#	16-124	16-124	16-124	16-124	16-124#	16-124#	16-124#	16-124#	16-125	16-125
	16-125	16-125	16-125#	16-125#	16-125#	16-125#	16-126	16-126	16-126	16-126	16-126#	16-126#	16-126#	16-126#
	16-127	16-127	16-127	16-127	16-127	16-127	16-127#	16-127#	16-127#	16-127#	16-127#	16-127#	16-128	16-128
	16-128	16-128#	16-128#	16-128#	16-129	16-129	16-129	16-129	16-129	16-129	16-129#	16-129#	16-129#	16-129#
	16-129#	16-129#	16-130	16-130	16-130	16-130	16-130#	16-130#	16-130#	16-130#	16-131	16-131	16-131	16-131
	16-131	16-131	16-131#	16-131#	16-131#	16-131#	16-131#	16-131#	16-132	16-132	16-132	16-132#	16-132#	16-132#
	16-133	16-133	16-133	16-133	16-133	16-133	16-133#	16-133#	16-133#	16-133#	16-133#	16-133#	16-141	16-141
	16-141	16-141#	16-141#	16-141#	16-142	16-142	16-142	16-142	16-142#	16-142#	16-142#	16-142#	16-143	16-143
	16-143#	16-143#	16-144	16-144	16-144	16-144	16-144#	16-144#	16-144#	16-144#	16-145	16-145	16-145	16-145
	16-145#	16-145#	16-145#	16-145#	16-146	16-146	16-146	16-146	16-146	16-146	16-146#	16-146#	16-146#	16-146#
	16-146#	16-146#	16-147	16-147	16-147	16-147#	16-147#	16-147#	16-148	16-148	16-148	16-148	16-148	16-148
	16-148#	16-148#	16-148#	16-148#	16-148#	16-148#	16-149	16-149	16-149	16-149	16-149#	16-149#	16-149#	16-149#
	16-150	16-150	16-150	16-150	16-150	16-150	16-150#	16-150#	16-150#	16-150#	16-150#	16-150#	16-151	16-151
	16-151	16-151#	16-151#	16-151#	16-152	16-152	16-152	16-152	16-152	16-152	16-152#	16-152#	16-152#	16-152#
	16-152#	16-152#	16-160	16-160	16-160	16-160	16-160#	16-160#	16-160#	16-160#	16-161	16-161	16-161	16-161
	16-161#	16-161#	16-161#	16-161#	16-162	16-162	16-162#	16-162#	16-163	16-163	16-163	16-163	16-163#	16-163#

```
         16-163#  16-163#  16-164   16-164   16-164   16-164   16-164#  16-164#  16-164#  16-164#  16-165   16-165   16-165   16-165
         16-165#  16-165#  16-165#  16-165#  16-166   16-166   16-166   16-166   16-166   16-166   16-166#  16-166#  16-166#  16-166#
         16-166#  16-166#  16-167   16-167   16-167   16-167#  16-167#  16-167#  16-168   16-168   16-168   16-168   16-168   16-168
         16-168#  16-168#  16-168#  16-168#  16-168#  16-168#  16-169   16-169   16-169   16-169   16-169#  16-169#  16-169#  16-169#
         16-170   16-170   16-170   16-170   16-170   16-170   16-170#  16-170#  16-170#  16-170#  16-170#  16-170#  16-171   16-171
         16-171   16-171#  16-171#  16-171#  16-172   16-172   16-172   16-172   16-172   16-172   16-172#  16-172#  16-172#  16-172#
         16-172#  16-172#  16-180   16-180   16-180   16-180#  16-180#  16-180#  16-181   16-181   16-181   16-181   16-181#  16-181#
         16-181#  16-181#  16-182   16-182   16-182#  16-182#  16-183   16-183   16-183   16-183   16-183#  16-183#  16-183#  16-183#
         16-184   16-184   16-184   16-184   16-184#  16-184#  16-184#  16-184#  16-185   16-185   16-185   16-185   16-185   16-185
         16-185#  16-185#  16-185#  16-185#  16-185#  16-185#  16-186   16-186   16-186   16-186#  16-186#  16-186#  16-187   16-187
         16-187   16-187   16-187   16-187   16-187#  16-187#  16-187#  16-187#  16-187#  16-187#  16-188   16-188   16-188   16-188
         16-188#  16-188#  16-188#  16-188#  16-189   16-189   16-189   16-189   16-189   16-189   16-189#  16-189#  16-189#  16-189#
         16-189#  16-189#  16-190   16-190   16-190   16-190#  16-190#  16-190#  16-191   16-191   16-191   16-191   16-191   16-191
         16-191#  16-191#  16-191#  16-191#  16-191#  16-191#  16-199   16-199   16-199   16-199   16-199#  16-199#  16-199#  16-199#
         16-200   16-200   16-200#  16-200#  16-201   16-201   16-201   16-201   16-201#  16-201#  16-201#  16-201#  16-202   16-202
         16-202   16-202   16-202#  16-202#  16-202#  16-202#  16-203   16-203   16-203   16-203   16-203   16-203   16-203#  16-203#
         16-203#  16-203#  16-203#  16-203#  16-204   16-204   16-204   16-204#  16-204#  16-204#  16-205   16-205   16-205   16-205
         16-205   16-205   16-205#  16-205#  16-205#  16-205#  16-205#  16-205#  16-206   16-206   16-206   16-206   16-206#  16-206#
         16-206#  16-206#  16-207   16-207   16-207   16-207   16-207   16-207   16-207#  16-207#  16-207#  16-207#  16-207#  16-207#
         16-208   16-208   16-208   16-208#  16-208#  16-208#  16-209   16-209   16-209   16-209   16-209   16-209   16-209#  16-209#
         16-209#  16-209#  16-209#  16-209#  16-217   16-217   16-217   16-217#  16-217#  16-217#  16-218   16-218   16-218   16-218
         16-218#  16-218#  16-218#  16-218#  16-219   16-219   16-219#  16-219#  16-220   16-220   16-220   16-220   16-220#  16-220#
         16-220#  16-220#  16-221   16-221   16-221   16-221   16-221#  16-221#  16-221#  16-221#  16-222   16-222   16-222   16-222
         16-222#  16-222#  16-222#  16-222#  16-223   16-223   16-223   16-223   16-223   16-223   16-223#  16-223#  16-223#  16-223#
         16-223#  16-223#  16-224   16-224   16-224   16-224#  16-224#  16-224#  16-225   16-225   16-225   16-225   16-225   16-225
         16-225#  16-225#  16-225#  16-225#  16-225#  16-225#  16-226   16-226   16-226   16-226   16-226#  16-226#  16-226#  16-226#
         16-227   16-227   16-227   16-227   16-227   16-227   16-227#  16-227#  16-227#  16-227#  16-227#  16-227#  16-228   16-228
         16-228   16-228#  16-228#  16-228#  16-229   16-229   16-229   16-229   16-229   16-229   16-229#  16-229#  16-229#  16-229#
         16-229#  16-229#  16-237   16-237   16-237   16-237   16-237#  16-237#  16-237#  16-237#  16-238   16-238   16-238#  16-238#
         16-239   16-239   16-239   16-239   16-239#  16-239#  16-239#  16-239#  16-240   16-240   16-240   16-240   16-240#  16-240#
         16-240#  16-240#  16-241   16-241   16-241   16-241   16-241   16-241   16-241#  16-241#  16-241#  16-241#  16-241#  16-241#
         16-242   16-242   16-242   16-242#  16-242#  16-242#  16-243   16-243   16-243   16-243   16-243   16-243   16-243#  16-243#
         16-243#  16-243#  16-243#  16-243#  22-12    22-12    22-12    22-12#   22-12#   22-12#   24-456   24-456   24-456   24-456#
         24-456#  24-456#  24-482   24-482   24-482#  24-482#  24-501   24-501   24-501#  24-501#  24-520   24-520   24-520#  24-520#
MSRADI    1-D77#   7-18#   25-16    25-16#   25-17    25-17#   25-18    25-18#   25-19    25-19#
MSRBRO    1-C52#   7-18#
MSRNRO    1-C62#   7-18#   19-51    19-51#
MSSETS    1-D32#   7-18#    7-24     7-24#   10-9     10-9#    11-8     11-8#    16-100   16-100#  16-106   16-106#  16-121   16-121#
         16-140   16-140#  16-159   16-159#  16-179   16-179#  16-198   16-198#  16-216   16-216#  16-236   16-236#  17-9     17-9#
         18-8     18-8#    19-8     19-8#    20-7     20-7#    21-8     21-8#    22-8     22-8#    23-9     23-9#    24-14    24-14#
         24-39    24-39#   24-67    24-67#   24-72    24-72    24-72#   24-72#   24-101   24-101#  24-128   24-128#  24-166   24-166#
         24-203   24-203#  24-247   24-247#  24-298   24-298#  24-303   24-303   24-303#  24-303#  24-332   24-332#  24-337   24-337
         24-337#  24-337#  24-366   24-366#  24-371   24-371   24-371#  24-371#  24-400   24-400#  24-405   24-405   24-405#  24-405#
         24-449   24-449#  24-539   24-539#  24-610   24-610#  24-676   24-676#  24-680   24-680   24-680#  24-680#  24-727   24-727#
         24-731   24-731   24-731#  24-731#  24-774   24-774#  24-778   24-778   24-778#  24-778#  24-832   24-832#  24-838   24-838
         24-838#  24-838#  24-889   24-889#  24-898   24-898#  24-900   24-900   24-900#  24-900#  24-924   24-924#  24-957   24-957#
         24-960   24-960#  24-:22   24-:22#  24-;52   24-;52#  24-<01   24-<01#  24-<49   24-<49#  24-=07   24-=07#  24-=91   24-=91#
         24->48   24->48#  24-?27   24-?27#  24-?63   24-?63#  24-@43   24-@43#  24-A26   24-A26#  24-A89   24-A89#  24-B69   24-B69#
         24-C05   24-C05#  24-D58   24-D58#  24-E53   24-E53#  24-F05   24-F05#  24-F70   24-F70#  24-G08   24-G08#  24-G59   24-G59#
         24-H13   24-H13#  25-14    25-14#   26-13    26-13#
MSSTAR    1-A33#   7-18#
MSSVC     1-C33#   7-18#   15-277   15-283   15-293   15-299   15-406   15-412   15-570   15-602   15-608   15-618   15-624   15-659
         15-665   15-699   15-705   15-713   15-719   15-865   15-870   15-889   15-895   15-904   15-910   15-919   15-925   15-934
         15-940   15-:72   15-:78   16-101   16-101#  16-102   16-102#  16-107   16-107#  16-108   16-108#  16-109   16-109#  16-110
         16-110#  16-111   16-111#  16-112   16-112#  16-113   16-113#  16-114   16-114#  16-115   16-115#  16-122   16-122#  16-123
         16-123#  16-124   16-124#  16-125   16-125#  16-126   16-126#  16-127   16-127#  16-128   16-128#  16-129   16-129#  16-130
         16-130#  16-131   16-131#  16-132   16-132#  16-133   16-133#  16-134   16-134#  16-141   16-141#  16-142   16-142#  16-143
```

```
        16-143#  16-144   16-144#  16-145   16-145#  16-146   16-146#  16-147   16-147#  16-148   16-148#  16-149   16-149#  16-150
        16-150#  16-151   16-151#  16-152   16-152#  16-153   16-153#  16-160   16-160#  16-161   16-161#  16-162   16-162#  16-163
        16-163#  16-164   16-164#  16-165   16-165#  16-166   16-166#  16-167   16-167#  16-168   16-168#  16-169   16-169#  16-170
        16-170#  16-171   16-171#  16-172   16-172#  16-173   16-173#  16-180   16-180#  16-181   16-181#  16-182   16-182#  16-183
        16-183#  16-184   16-184#  16-185   16-185#  16-186   16-186#  16-187   16-187#  16-188   16-188#  16-189   16-189#  16-190
        16-190#  16-191   16-191#  16-192   16-192#  16-199   16-199#  16-200   16-200#  16-201   16-201#  16-202   16-202#  16-203
        16-203#  16-204   16-204#  16-205   16-205#  16-206   16-206#  16-207   16-207#  16-208   16-208#  16-209   16-209#  16-210
        16-210#  16-217   16-217#  16-218   16-218#  16-219   16-219#  16-220   16-220#  16-221   16-221#  16-222   16-222#  16-223
        16-223#  16-224   16-224#  16-225   16-225#  16-226   16-226#  16-227   16-227#  16-228   16-228#  16-229   16-229#  16-230
        16-230#  16-237   16-237#  16-238   16-238#  16-239   16-239#  16-240   16-240#  16-241   16-241#  16-242   16-242#  16-243
        16-243#  16-244   16-244#  17-11    17-11#   19-25    19-25#   19-28    19-28#   19-31    19-31#   19-34    19-34#   19-51
        19-51#   19-74    19-74#   20-9     20-9#    20-16    20-16#   20-19    20-19#   21-11    21-11#   22-10    22-10#   22-12
        22-12#   22-13    22-13#   23-10    23-10#   24-16    24-16#   24-24    24-27    24-27#   24-50    24-52    24-52#   24-72
        24-72#   24-84    24-86    24-86#   24-90    24-90#   24-115   24-117   24-117#  24-133   24-133#  24-148   24-150   24-150#
        24-184   24-186   24-186#  24-226   24-227   24-227#  24-232   24-232#  24-278   24-279   24-279#  24-286   24-286#  24-303
        24-303#  24-314   24-316   24-316#  24-320   24-320#  24-337   24-337#  24-348   24-350   24-350#  24-354   24-354#  24-371
        24-371#  24-382   24-384   24-384#  24-388   24-388#  24-405   24-405#  24-416   24-418   24-418#  24-422   24-422#  24-456
        24-456#  24-480   24-482   24-482#  24-499   24-501   24-501#  24-518   24-520   24-520#  24-523   24-523#  24-551   24-552
        24-552#  24-565   24-566   24-566#  24-577   24-578   24-578#  24-588   24-589   24-589#  24-594   24-594#  24-637   24-638
        24-638#  24-647   24-648   24-648#  24-655   24-655#  24-680   24-680#  24-694   24-695   24-695#  24-704   24-706   24-706#
        24-710   24-710#  24-731   24-731#  24-745   24-746   24-746#  24-754   24-756   24-756#  24-760   24-760#  24-778   24-778#
        24-792   24-793   24-793#  24-801   24-803   24-803#  24-807   24-807#  24-838   24-838#  24-855   24-856   24-856#  24-865
        24-867   24-867#  24-872   24-872#  24-898   24-898#  24-900   24-900#  24-913   24-915   24-915#  24-920   24-920#  24-924
        24-924#  24-936   24-938   24-938#  24-939   24-939#  24-960   24-960#  24-983   24-984   24-984#  24-993   24-994   24-994#
        24-999   24-999#  24-:00   24-:00#  24-:98   24-:99   24-:99#  24-;21   24-;22   24-;22#  24-;32   24-;32#  24-;81   24-;81#
        24-<27   24-<27#  24-<83   24-<87   24-<87#  24-=26   24-=35   24-=47   24-=66   24-=74   24-=76   24-=76#  24->32   24->32#
        24-?12   24-?12#  24-?47   24-?47#  24-?97   24-@12   24-@27   24-@27#  24-@95   24-A10   24-A10#  24-A60   24-A73   24-A73#
        24-B41   24-B56   24-B56#  24-B85   24-B88   24-B88#  24-C89   24-D34   24-D34#  24-E35   24-E35#  24-E87   24-E87#  24-F55
        24-F55#  24-F87   24-F87#  24-G40   24-G40#  24-G90   24-G90#  24-H60   24-H60#
MSTLAB   1-C29#   7-18#   15-277#  15-283#  15-293#  15-299#  15-406#  15-412#  15-570#  15-602#  15-608#  15-618#  15-624#  15-659#
        15-665#  15-699#  15-705#  15-713#  15-719#  15-865#  15-870#  15-889#  15-895#  15-904#  15-910#  15-919#  15-925#  15-934#
        15-940#  15-:72#  15-:78#  16-101#  16-102#  16-107#  16-108#  16-109#  16-110#  16-111#  16-112#  16-113#  16-114#  16-115#
        16-122#  16-123#  16-124#  16-125#  16-126#  16-127#  16-128#  16-129#  16-130#  16-131#  16-132#  16-133#  16-134#  16-141#
        16-142#  16-143#  16-144#  16-145#  16-146#  16-147#  16-148#  16-149#  16-150#  16-151#  16-152#  16-153#  16-160#  16-161#
        16-162#  16-163#  16-164#  16-165#  16-166#  16-167#  16-168#  16-169#  16-170#  16-171#  16-172#  16-173#  16-180#  16-181#
        16-182#  16-183#  16-184#  16-185#  16-186#  16-187#  16-188#  16-189#  16-190#  16-191#  16-192#  16-199#  16-200#  16-201#
        16-202#  16-203#  16-204#  16-205#  16-206#  16-207#  16-208#  16-209#  16-210#  16-217#  16-218#  16-219#  16-220#  16-221#
        16-222#  16-223#  16-224#  16-225#  16-226#  16-227#  16-228#  16-229#  16-230#  16-237#  16-238#  16-239#  16-240#  16-241#
        16-242#  16-243#  16-244#  17-11#   19-25#   19-28#   19-31#   19-34#   19-51#   19-74#   20-9#    20-16#   20-19#   21-11#
        22-10#   22-12#   22-13#   23-10#   24-16#   24-24#   24-27#   24-50#   24-52#   24-72#   24-84#   24-86#   24-90#   24-115#
        24-117#  24-133#  24-148#  24-150#  24-184#  24-186#  24-226#  24-227#  24-232#  24-278#  24-279#  24-286#  24-303#  24-314#
        24-316#  24-320#  24-337#  24-348#  24-350#  24-354#  24-371#  24-382#  24-384#  24-388#  24-405#  24-416#  24-418#  24-422#
        24-456#  24-480#  24-482#  24-499#  24-501#  24-518#  24-520#  24-523#  24-551#  24-552#  24-565#  24-566#  24-577#  24-578#
        24-588#  24-589#  24-594#  24-637#  24-638#  24-647#  24-648#  24-655#  24-680#  24-694#  24-695#  24-704#  24-706#  24-710#
        24-731#  24-745#  24-746#  24-754#  24-756#  24-760#  24-778#  24-792#  24-793#  24-801#  24-803#  24-807#  24-838#  24-855#
        24-856#  24-865#  24-867#  24-872#  24-898#  24-900#  24-913#  24-915#  24-920#  24-924#  24-936#  24-938#  24-939#  24-960#
        24-983#  24-984#  24-993#  24-994#  24-999#  24-:00#  24-:98#  24-:99#  24-;21#  24-;22#  24-;32#  24-;81#  24-<27#  24-<83#
        24-<87#  24-=26#  24-=35#  24-=47#  24-=66#  24-=74#  24-=76#  24->32#  24-?12#  24-?47#  24-?97#  24-@12#  24-@27#  24-@95#
        24-A10#  24-A60#  24-A73#  24-B41#  24-B56#  24-B85#  24-B88#  24-C89#  24-D34#  24-E35#  24-E87#  24-F55#  24-F87#  24-G40#
        24-G90#  24-H60#
MSTSTL   1-C21#   7-18#   15-277   15-277#  15-277#  15-283   15-283#  15-283#  15-293   15-293#  15-293#  15-299   15-299#  15-299#
        15-406   15-406#  15-406#  15-412   15-412#  15-412#  15-570   15-570#  15-570#  15-602   15-602#  15-602#  15-608   15-608#
        15-608#  15-618   15-618#  15-618#  15-624   15-624#  15-624#  15-659   15-659#  15-659#  15-665   15-665#  15-665#  15-699
        15-699#  15-699#  15-705   15-705#  15-705#  15-713   15-713#  15-713#  15-719   15-719#  15-719#  15-865   15-865#  15-865#
        15-870   15-870#  15-870#  15-889   15-889#  15-889#  15-895   15-895#  15-895#  15-904   15-904#  15-904#  15-910   15-910#
        15-910#  15-919   15-919#  15-919#  15-925   15-925#  15-925#  15-934   15-934#  15-934#  15-940   15-940#  15-940#  15-:72
        15-:72#  15-:72#  15-:78   15-:78#  15-:78#  16-101   16-101#  16-102   16-102#  16-107   16-107#  16-108   16-108#  16-109
```

```
        16-109# 16-110  16-110# 16-111  16-111# 16-112  16-112# 16-113  16-113# 16-114  16-114# 16-115  16-115# 16-122
        16-122# 16-123  16-123# 16-124  16-124# 16-125  16-125# 16-126  16-126# 16-127  16-127# 16-128  16-128# 16-129
        16-129# 16-130  16-130# 16-131  16-131# 16-132  16-132# 16-133  16-133# 16-134  16-134# 16-141  16-141# 16-142
        16-142# 16-143  16-143# 16-144  16-144# 16-145  16-145# 16-146  16-146# 16-147  16-147# 16-148  16-148# 16-149
        16-149# 16-150  16-150# 16-151  16-151# 16-152  16-152# 16-153  16-153# 16-160  16-160# 16-161  16-161# 16-162
        16-162# 16-163  16-163# 16-164  16-164# 16-165  16-165# 16-166  16-166# 16-167  16-167# 16-168  16-168# 16-169
        16-169# 16-170  16-170# 16-171  16-171# 16-172  16-172# 16-173  16-173# 16-180  16-180# 16-181  16-181# 16-182
        16-182# 16-183  16-183# 16-184  16-184# 16-185  16-185# 16-186  16-186# 16-187  16-187# 16-188  16-188# 16-189
        16-189# 16-190  16-190# 16-191  16-191# 16-192  16-192# 16-199  16-199# 16-200  16-200# 16-201  16-201# 16-202
        16-202# 16-203  16-203# 16-204  16-204# 16-205  16-205# 16-206  16-206# 16-207  16-207# 16-208  16-208# 16-209
        16-209# 16-210  16-210# 16-217  16-217# 16-218  16-218# 16-219  16-219# 16-220  16-220# 16-221  16-221# 16-222
        16-222# 16-223  16-223# 16-224  16-224# 16-225  16-225# 16-226  16-226# 16-227  16-227# 16-228  16-228# 16-229
        16-229# 16-230  16-230# 16-237  16-237# 16-238  16-238# 16-239  16-239# 16-240  16-240# 16-241  16-241# 16-242
        16-242# 16-243  16-243# 16-244  16-244# 17-11   17-11#  19-25   19-25#  19-28   19-28#  19-31   19-31#  19-34
        19-34#  19-51   19-51#  19-74   19-74#  20-9    20-9#   20-16   20-16#  20-19   20-19#  21-11   21-11#  22-10
        22-10#  22-12   22-12#  22-13   22-13#  23-10   23-10#  24-16   24-16#  24-24   24-24#  24-24#  24-27   24-27#
        24-50   24-50#  24-50#  24-52   24-52#  24-72   24-72#  24-84   24-84#  24-84#  24-86   24-86#  24-90   24-90#
        24-115  24-115# 24-115# 24-117  24-117# 24-133  24-133# 24-148  24-148# 24-148# 24-150  24-150# 24-184  24-184#
        24-184# 24-186  24-186# 24-226  24-226# 24-226# 24-227  24-227# 24-232  24-232# 24-278  24-278# 24-278# 24-279
        24-279# 24-286  24-286# 24-303  24-303# 24-314  24-314# 24-314# 24-316  24-316# 24-320  24-320# 24-337  24-337#
        24-348  24-348# 24-348# 24-350  24-350# 24-354  24-354# 24-371  24-371# 24-382  24-382# 24-382# 24-384  24-384#
        24-388  24-388# 24-405  24-405# 24-416  24-416# 24-416# 24-418  24-418# 24-422  24-422# 24-456  24-456# 24-480
        24-480# 24-480# 24-482  24-482# 24-499  24-499# 24-499# 24-501  24-501# 24-518  24-518# 24-518# 24-520  24-520#
        24-523  24-523# 24-551  24-551# 24-551# 24-552  24-552# 24-565  24-565# 24-565# 24-566  24-566# 24-577  24-577#
        24-577# 24-578  24-578# 24-588  24-588# 24-588# 24-589  24-589# 24-594  24-594# 24-637  24-637# 24-637# 24-638
        24-638# 24-647  24-647# 24-647# 24-648  24-648# 24-655  24-655# 24-680  24-680# 24-694  24-694# 24-694# 24-695
        24-695# 24-704  24-704# 24-704# 24-706  24-706# 24-710  24-710# 24-731  24-731# 24-745  24-745# 24-745# 24-746
        24-746# 24-754  24-754# 24-754# 24-756  24-756# 24-760  24-760# 24-778  24-778# 24-792  24-792# 24-792# 24-793
        24-793# 24-801  24-801# 24-801# 24-803  24-803# 24-807  24-807# 24-838  24-838# 24-855  24-855# 24-855# 24-856
        24-856# 24-865  24-865# 24-865# 24-867  24-867# 24-872  24-872# 24-898  24-898# 24-900  24-900# 24-913  24-913#
        24-913# 24-915  24-915# 24-920  24-920# 24-924  24-924# 24-936  24-936# 24-936# 24-938  24-938# 24-939  24-939#
        24-960  24-960# 24-983  24-983# 24-983# 24-984  24-984# 24-993  24-993# 24-993# 24-994  24-994# 24-999  24-999#
        24-:00  24-:00# 24-:98  24-:98# 24-:98# 24-:99  24-:99# 24-;21  24-;21# 24-;21# 24-;22  24-;22# 24-;32  24-;32#
        24-;81  24-;81# 24-<27  24-<27# 24-<83  24-<83# 24-<83# 24-<87  24-<87# 24-=26  24-=26# 24-=26# 24-=35  24-=35#
        24-=35# 24-=47  24-=47# 24-=47# 24-=66  24-=66# 24-=66# 24-=74  24-=74# 24-=74# 24-=76  24-=76# 24->32  24->32#
        24-?12  24-?12# 24-?47  24-?47# 24-?97  24-?97# 24-?97# 24-@12  24-@12# 24-@12# 24-@27  24-@27# 24-@95  24-@95#
        24-@95# 24-A10  24-A10# 24-A60  24-A60# 24-A60# 24-A73  24-A73# 24-B41  24-B41# 24-B41# 24-B56  24-B56# 24-B85
        24-B85# 24-B85# 24-B88  24-B88# 24-C89  24-C89# 24-C89# 24-D34  24-D34# 24-E35  24-E35# 24-E87  24-E87# 24-F55
        24-F55# 24-F87  24-F87# 24-G40  24-G40# 24-G90  24-G90# 24-H60  24-H60#
MSWORD  1-C94#  7-18#   8-17    8-17#   9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8
        9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8
        9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8     9-8
        9-8     9-8     9-8     9-8     9-8     9-8#    15-277  15-277  15-277  15-277# 15-283  15-283  15-283  15-283#
        15-293  15-293  15-293  15-293# 15-299  15-299  15-299  15-299# 15-406  15-406  15-406  15-406# 15-412  15-412
        15-412  15-412# 15-570  15-570  15-570  15-570# 15-602  15-602  15-602  15-602# 15-608  15-608  15-608  15-608#
        15-618  15-618  15-618  15-618# 15-624  15-624  15-624  15-624# 15-659  15-659  15-659  15-659# 15-665  15-665
        15-665  15-665# 15-699  15-699  15-699  15-699# 15-705  15-705  15-705  15-705# 15-713  15-713  15-713  15-713#
        15-719  15-719  15-719  15-719# 15-865  15-865  15-865  15-865# 15-870  15-870  15-870  15-870# 15-889  15-889
        15-889  15-889# 15-895  15-895  15-895  15-895# 15-904  15-904  15-904  15-904# 15-910  15-910  15-910  15-910#
        15-919  15-919  15-919  15-919# 15-925  15-925  15-925  15-925# 15-934  15-934  15-934  15-934# 15-940  15-940
        15-940  15-940# 15-:72  15-:72  15-:72  15-:72# 15-:78  15-:78  15-:78  15-:78# 24-24   24-24   24-24   24-24#
        24-50   24-50   24-50   24-50#  24-84   24-84   24-84   24-84#  24-115  24-115  24-115  24-115# 24-148  24-148
        24-148  24-148# 24-184  24-184  24-184  24-184# 24-226  24-226  24-226  24-226# 24-278  24-278  24-278  24-278#
        24-314  24-314  24-314  24-314# 24-348  24-348  24-348  24-348# 24-382  24-382  24-382  24-382# 24-416  24-416
        24-416  24-416# 24-480  24-480  24-480  24-480# 24-499  24-499  24-499  24-499# 24-518  24-518  24-518  24-518#
        24-551  24-551  24-551  24-551# 24-565  24-565  24-565  24-565# 24-577  24-577  24-577  24-577# 24-588  24-588
        24-588  24-588# 24-637  24-637  24-637  24-637# 24-647  24-647  24-647  24-647# 24-694  24-694  24-694  24-694#
```

```
        24-704   24-704   24-704   24-704#  24-745   24-745   24-745   24-745#  24-754   24-754   24-754   24-754#  24-792   24-792
        24-792   24-792#  24-801   24-801   24-801   24-801#  24-855   24-855   24-855   24-855#  24-865   24-865   24-865   24-865#
        24-913   24-913   24-913   24-913#  24-936   24-936   24-936   24-936#  24-983   24-983   24-983   24-983#  24-993   24-993
        24-993   24-993#  24-:98   24-:98   24-:98   24-:98#  24-;21   24-;21   24-;21   24-;21#  24-<83   24-<83   24-<83   24-<83#
        24-=26   24-=26   24-=26   24-=26#  24-=35   24-=35   24-=35   24-=35#  24-=47   24-=47   24-=47   24-=47#  24-=66   24-=66
        24-=66   24-=66#  24-=74   24-=74   24-=74   24-=74#  24-?97   24-?97   24-?97   24-?97#  24-@12   24-@12   24-@12   24-@12#
        24-@95   24-@95   24-@95   24-@95#  24-A60   24-A60   24-A60   24-A60#  24-B41   24-B41   24-B41   24-B41#  24-B85   24-B85
        24-B85   24-B85#  24-C89   24-C89   24-C89   24-C89#  25-16    25-16#   25-17    25-17#   25-18    25-18#   25-19    25-19#
        26-39    26-39
M$XFER   1-@82#   7-18#
MANUAL   1-;62#   7-18#
MEMORY   1-;66#   7-18#
OPEN     1-;71#   7-18#
POINTE   1-;76#   7-18#   8-7
PRINTB   1-<39#   7-18#   16-101   16-107   16-108   16-109   16-110   16-122   16-123   16-124   16-125   16-141   16-142   16-143
        16-144   16-160   16-161   16-162   16-163   16-164   16-180   16-181   16-182   16-183   16-199   16-200   16-201   16-217
        16-218   16-219   16-220   16-221   16-237   16-238   16-239
PRINTF   1-<79#   7-18#   22-12    24-456   24-482   24-501   24-520
PRINTS   1-=19#   7-18#
PRINTX   1-=59#   7-18#   16-111   16-112   16-113   16-114   16-126   16-127   16-128   16-129   16-130   16-131   16-132   16-133
        16-145   16-146   16-147   16-148   16-149   16-150   16-151   16-152   16-165   16-166   16-167   16-168   16-169   16-170
        16-171   16-172   16-184   16-185   16-186   16-187   16-188   16-189   16-190   16-191   16-202   16-203   16-204   16-205
        16-206   16-207   16-208   16-209   16-222   16-223   16-224   16-225   16-226   16-227   16-228   16-229   16-240   16-241
        16-242   16-243
READBU   1-=99#   7-18#
READEF   1->03#   7-18#   19-25    19-28    19-31    19-34
RFLAGS   1->08#   7-18#
SETPRI   1->13#   7-18#   20-9     24-16
SETVEC   1->18#   7-18#
SLASH    1->24#   7-18#
STARS    1->38#   7-18#
SVC      1->52#   7-17#   7-18
XFER     1-@12#   7-18#
XFERF    1-@16#   7-18#
XFERT    1-@20#   7-18#
```