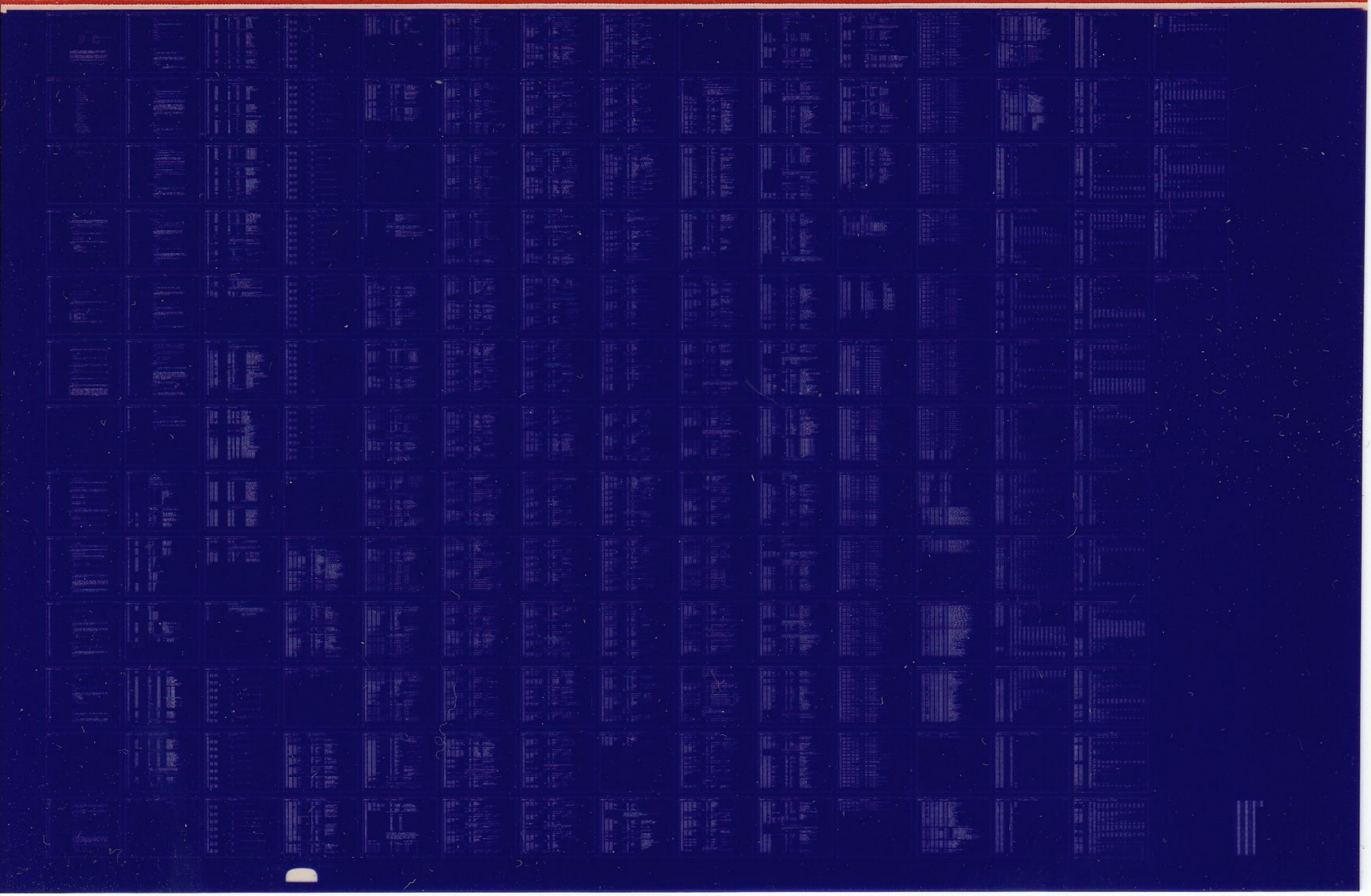
RM03/2

DISKLESS 2
CZRMKB0

AH-E377B-MC

FICHE 1 OF 1

DEC 1978
digital
MADE IN USA



```
                                .REM    \
```

IDENTIFICATION

PRODUCT CODE:	AC-E376B-MC
PRODUCT NAME:	CZRMKB0 RM03/RM02 DISKLESS DIAGNOSTIC, PART II
DATE CREATED:	1-MARCH-78
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	DOUG RIIKONEN

CONTENTS

6. XXDP COMPATIBILITY

7. OPERATING SYSTEM COMPATIBILITY

6. TEST DESCRIPTION

1.0 INTRODUCTION

1.1 ABSTRACT

THE RM03 DISKLESS DIAGNOSTIC IS A STAND ALONE PROGRAM WHICH USES FUNCTIONAL AND DIAGNOSTIC MEANS TO VERIFY THE OPERABILITY OF THE RM03 DISK SUBSYSTEM EXCLUDING AND INDEPENDENTLY OF THE STORAGE MODULE DRIVE. IN PARTICULAR, THE PROGRAM SERVES THE FOLLOWING PURPOSES:

TO DETECT ERRORS AND FAULTS IN THE RM03 MASSBUS ADAPTER;

TO RESOLVE HARDWARE FAILURES IN RM03 TO A FIELD REPLACEABLE MODULE OR MODULES.

1.2 UNIT UNDER TEST

THE UNIT UNDER TEST (UUT) IS THE RM03 DISK SUBSYSTEM, EXCLUDING THE STORAGE MODULE DISK DRIVE AND THE MASSBUS CONTROLLER.

2.0 OPERATING REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

THE FOLLOWING MINIMUM HARDWARE CONFIGURATION, ASSUMED TO BE OPERATIONAL, IS REQUIRED TO LOAD AND EXECUTE THE RM03 DISKLESS DIAGNOSTIC:

PDP-11 PROCESSOR
16 K MEMORY
KW11-L OR LW11-P CLOCK
PROGRAM LOADING DEVICE
TERMINAL
RH CONTROLLER
UNIT UNDER TEST,
WHERE THE UNIT UNDER TEST CONSISTS OF ONE TO EIGHT RM03 ADAPTERS.

2.2 MEDIA REQUIREMENTS

NONE

2.3 PREREQUISITE DIAGNOSTIC PROGRAMS

CZRMJ- ,RM03/02 DISKLESS DIAGNOSTIC PART 1

3.0 OPERATING PROCEDURE

3.1 LOADING

THE PROGRAM MAY BE LOADED BY EITHER PAPER TAPE, USING THE STANDARD PAPER TAPE LOADING PROCEDURE, OR XXDP MEDIA, USING THE APPROPRIATE LOADING DEVICE.

3.2 SWITCH OPTIONS

THE FOLLOWING SWITCH OPTIONS ARE INVOKED WHEN THE APPROPRIATE SWITCH IS ON.

SW15	HALT ON ERROR
SW14	LOOP ON TEST (CURRENTLY BEING EXECUTED)
SW13	INHIBIT ERROR TYPEOUTS
SW12	UNUSED
SW11	INHIBIT TEST ITERATIONS
SW10	BELL ON ERROR
SW09	LOOP ON ERROR
SW08	LOOP ON TEST IN SW07-00

THE LOW ORDER 8 SWITCHES ARE USED IN CONJUNCTION WITH SW08 TO SPECIFY THE OCTAL NUMBER OF THE TEST WHICH THE PROGRAM WILL LOOP ON.

3.3 STARTING

THE PROGRAM STARTS AT LOCATION 200 WHICH USES DEFAULT VALUES OF PARAMETERS AND PROVIDES MAXIMUM TESTING WITH THE SWITCH REGISTER EQUAL TO ZERO.

4.0 OPERATOR INTERFACE

4.1 PROGRAM ID

THE PROGRAM TYPES ITS NAME AND MAINDEC NUMBER THE FIRST TIME IT IS STARTED AFTER BEING LOADED.

4.2 PROGRESS REPORTS

AN END OF PASS REPORT OCCURRS EACH TIME THE PROGRAM IS EXECUTED FOR ALL ADAPTERS IN THE TEST QUE. THE END OF PASS REPORT INCLUDES A MESSAGE AND AN ERROR SUMMARY.

4.3 PERFORMANCE REPORT

NO PERFORMANCE REPORTS ARE GIVEN DURING THE EXECUTION OF THE PROGRAM.

4.4 PROGRAM HALTS

THERE ARE NO SCHEDULED HALTS DURING THE EXECUTION OF THE PROGRAM. PROCESSOR HALTS ARE DUE TO THE TRAP CATCHER.

4.5 ERROR REPORTS

THE RM03 DISKLESS DIAGNOSTIC PROVIDES COMPREHENSIVE ERROR REPORTS INTENDED TO (1) AID IN FAULT RESOLUTION AND (2) MINIMIZE REFERENCES TO PROGRAM LISTINGS.

THE FIRST LINE OF THE ERROR REPORT CONTAINS THE NUMBER OF THE UNIT BEING TESTED, THE TEST NUMBER, THE ERROR NUMBER AND THE VALUE OF THE PROGRAM COUNTER WHERE THE ERROR WAS CALLED. THIS LINE IS FOLLOWED BY THE ERROR MESSAGE: SEVERAL LINES OF TEXT WHICH GIVE A COMPREHENSIVE DESCRIPTION OF THE ERROR, AND A LIST OF FAILING MODULES IN ORDER OF DECREASING PROBABILITY. THE ERROR MESSAGE IS NORMALLY FOLLOWED BY ONE OR MORE PAIRS OF LINES CONTAINING DATA HEADERS AND DATA PERTININENT TO THE ERROR, INCLUDING EXPECTED AND ACTUAL TEST

```
265                                 RESULTS.
```

5.0 ENVIRONMENTAL SUPPORT

5.1 PROCESSOR COMPATIBILITY

THE RM03 DISKLESS DIAGNOSTIC IS EXECUTABLE ON ANY PDP-11 PROCESSOR, PROVIDING PREVIOUSLY MENTIONED HARDWARE REQUIREMENTS ARE MET.

5.2 DUAL PORT CONFIGURATIONS

THE RM03 DISKLESS DIAGNOSTIC IS NOT EXECUTABLE ON RM03 SUBSYSTEMS HAVING THE DUAL PORT OPTION UNLESS THE DUAL PORT SWITCH IS SET TO THE APPROPRIATE PORT (A OR B) AND NOT TO THE PROGRAMMABLE POSITION (A/B).

5.3 MEMORY PARITY HARDWARE

MEMORY PARITY HARDWARE WILL NOT BE USED DURING THE EXECUTION OF THE RM03 DISKLESS DIAGNGOSTIC.

5.4 MEMORY MANAGEMENT HARDWARE

MEMORY MANAGEMENT HARDWARE WILL NOT BE USED DURING THE RM03 DISKLESS DIAGNOSTIC.

5.5 ACT11, APT11 COMPATIBILITY

THE RM03 DISKLESS DIAGNOSTIC PROGRAM IS COMPATIBLE WITH ACT11 AND APT11 IN BOTH DUMP AND AUTOMATIC MODES. FURTHER, THE PROGRAM WILL EXECUTE A QUICK PASS DURING THE FIRST PASS IN SUPPORT OF QUICK VERIFY MODE.

5.6 XXDP COMPATIBILITY

THE RM03 DISKLESS DIAGNOSTIC PROGRAM IS COMPATIBLE WITH XXDP IN DUMP AND CHAIN MODES.

5.7 OPERATING SYSTEM COMPATIBILITY

THE PROGRAM IS NOT REQUIRED TO BE COMPATIBLE WITH ANY OPERATING SYSTEM.

6.0 TEST DESCRIPTION

THE PROGRAM IS DESIGNED IN A BOTTOM UP MANNER SUCH THAT EACH TEST GENERALLY USES A MORE COMPLEX SUBSET OF HARDWARE THAN THE PREVIOUS TEST.

MODULE CALLOUT IS PREDICATED ON THE ASSUMPTION THAT EARLIER TESTS HAVE BEEN COMPLETED WITHOUT ERROR AND THAT ERRORS ARE DUE TO SINGLE, NONTRANSIENT HARDWARE FAILURES.

THE "RM03 DISKLES DIAGNOSTIC" CAN BE EXECUTED USING AN RH70 OR AN RH11 MASSBUS CONTROLLER.

UNLESS SPECIFIED BY THE OPERATOR OR BY THE ENVIRONMENT TABLE THE TEST IS REPEATED FOR EACH POSSIBLE DEVICE STARTING WITH DEVICE 0.

THE MODULES WHICH MAY BE CALLED OUT DURING THE EXECUTION OF THE TEST ARE AS FOLLOWS:

IF
CS
DS
MASSBUS MODULE

THE RADIAL MODULE (RD) IS NOT TESTED BY THIS PROGRAM.

TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE RM03 CAN COMPLETE A REGISTER TRANSFER ON THE MASSBUS, AND, IN PARTICULAR, TO VERIFY THAT "TRANSFER" IS NOT STUCK IN AN INACTIVE STATE.

PROCEDURE:

THE PROGRAM WRITES AND READS REMOTE REGISTERS FOR THE SELECTED DEVICE. REGISTER CONTENTS AND PARITY ERRORS ARE IGNORED, AND THE TEST FAILS IF A "NONEXISTENT DEVICE ERROR" OR BUS TIMEOUT OCCURRS FOR EVERY REGISTER ACCESS. IF THE TEST FAILS THE PROGRAM JUMPS TO THE END OF PASS HANDLER WHICH SELECTS THE NEXT DEVICE TO BE TESTED.

PROBABLE FAULT:

THE TEST FAILS IF THE SELECTED DEVICE IS NONEXISTENT OR IS SWITCHED TO THE PROGRAMMABLE POSITION OR TO THE ALTERNATE PORT. THE FOLLOWING FAULTS ARE APPLICABLE ONLY WHEN THE DEVICE IS PRESENT AND IS SWITCHED TO THE APPROPRIATE PORT.

1. IF MODULE

2. ASYNCHRONOUS MASSBUS MODULE

3. CS MODULE

CTOD TEST

PURPOSE:

TO VERIFY THAT DATA CAN BE TRANSFERRED TO AND FROM THE RM03 USING THE CONTROL BUS AND, IN PARTICULAR, TO VERIFY THAT ''CONTROLLER TO DEVICE'' HAS NOT FAILED.

PROCEDURE:

THE TEST WRITES ONES IN REMOTE REGISTERS THEN READS EACH REGISTER WHICH WILL WRITE ZEROS IN THE REGISTER IF ''IF3 CTOD HOLD H'' IS STUCK AT ONE. THE TEST THEN READS AS MANY REMOTE REGISTERS AS ARE NECESSARY TO OBTAIN ONE OR MORE ONE BITS.

PROBABLE FAULT:

1. IF MODULE

2. ASYNCHRONOUS MASSBUS MODULE

MASSBUS INITIALIZE TEST

PURPOSE:

TO VERIFY THAT THE MASSBUS ADAPTER IS BEING INITIALIZED BY THE MASS BUS.

PROCEDURE:

USING CONTROLLER CLEAR TO INITIALIZE THE SELECTED UNIT, THIS TEST THEN READS MASSBUS ADAPTER REGISTERS TO VERIFY THAT AT LEAST ONE BIT IS CLEARED. MASSBUS ADAPTER REGISTERS ARE PRESET TO A NON ZERO VALUE PRIOR TO CONTROLLER CLEAR.

PROBABLE FAULT:

1. ASYNCHRONOUS MASSBUS MODULE

2. IF MODULE

3. CS MODULE

CLEAR STUCK ACTIVE TEST

PURPOSE:

TO VERIFY THAT 'MBA CLR L'' ON THE CS MODULE IS NOT STUCK IN AN ACTIVE STATE.

PROCEDURE:

CONTROLLER CLEAR IS USED TO INITIALIZE THE SELECTED UNIT, AFTER WHICH 1'S ARE WRITTEN IN ERROR REGISTERS 1 AND 2 AND MAINTENANCE REGISTER 1. IF ANY 1 BITS CAN BE READ BACK THE TEST IS OK, ELSE, 'MBA CLR L'' IS PROBABLY STUCK ACTIVE.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

3. ASYNCHRONOUS MASSBUS MODULE

TRISTATE TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE PATH TO AND FROM THE MASSBUS ADAPTER TRI-STATE REGISTER BUS IS NOT STUCK AT ONE OR ZERO AND THAT EACH BIT POSITION IS INDEPENDENT.

PROCEDURE:

THIS TEST PRESETS MASSBUS ADAPTER REGISTERS TO A NONZERO VALUE, THEN, ASSUMING THE REGISTERS ARE PRESET, IT CLEARS THEM USING A MOVE INSTRUCTION. THE TEST THEN READS AS MANY REGISTERS AS IS NECESSARY TO OBTAIN ONE OR MORE ZEROS FROM EACH BIT POSITION.

```
   488
```

THE TEST CLEARS MASSBUS ADAPTER REGISTERS, THEN, ASSUMING THE REGISTERS ARE CLEARED, IT LOADS THEM WITH ONES AND READS AS MANY REGISTERS AS IS NECESSARY TO OBTAIN ONE OR MORE ONE BITS IN EACH BIT POSITION.

FINALLY, THE TEST WRITES A SINGLE ONE BIT PATTERN IN BIT 0 OF SELECTED REMOTE REGISTERS AND VERIFIES THAT THE PATTERN CAN BE READ BACK. THE ONE BIT IS SHIFTED AND THE TEST REPEATED FOR ALL BIT POSITIONS.

PROBABLE FAULT:

1. ASYNCHRONOUS MASSBUS MODULE
2. IF MODULE
3. CS MODULE
4. DS MODULE

REGISTER SELECT TEST

PURPOSE:

TO VERIFY THAT THE REGISTER SELECT LINES ARE NOT IN A STUCK POSITION.

PROCEDURE:

EACH REGISTER SELECT LINE IS TESTED BY WRITING ZEROS IN THOSE DEVICE REGISTERS FOR WHICH THE LINE MUST BE ZERO, THEN WRITING ONES IN THOSE DEVICE REGISTERS FOR WHICH THE LINE MUST BE ONE. THE ZERO REGISTER IS READ BACK AND IF THE SELECT LINE IS STUCK AT ZERO, THE ZERO REGISTER WILL CONTAIN ONES. THE PROCESS IS REPEATED TO DETECT A STUCK AT ONE FAULT, EXCEPT IN THIS CASE, THE ONES REGISTER IS WRITTEN FIRST.

REGISTER SELECT LINES 1, 2, 4 AND 8 ARE TESTED IN THIS MANNER; SELECT LINE 16 IS EXPLICITLY TESTED IN THE "ILR TEST".

PROBABLE FAULT:

1. IF MODULE
2. ASYNCHRONOUS MASSBUS MODULE

```
                                DRIVE TYPE TEST

                                        PURPOSE:
                                                TO TEST THE 'DRIVE TYPE'' REGISTER, RMDT.

                                        PROCEDURE:

                                                THE PROGRAM READS RMDT AND VERIFIES THAT THE RESULT
                                        CORRESPONDS TO A SINGLE PROT OR DUAL PROT RM03 DRIVE

                                        PROBABLE FAULT:

                                        1. IF MOULE

                                DEVICE AVAILABLE TEST

                                        PRUPOSE:
                                        TO VERIFY THAT DEVICE AVAILABLE STATUS IS SET.

                                        PROCEDURE:

                                        THE PROGRAM TESTS 'DVA'',BIT 11 OF RMCS1.

                                        PROBABLE FAULT:

                                        1. IF MODULE


                        SEARCH TIMEOUT TEST

                                        PURPOSE:

                                             TO VERIFY THAT THE SEARCH TIMEOUT ONE SHOT SETS 'OPI'',
                                        EXCEPT WHEN ''SEARCH TO DISABLE'' IS ACTIVE.

                                        PROCEDURE:

                                             WITH SEARCH TIMEOUT DISABLED, THE TEST EXECUTES A DATA
                                        COMMAND TO THE POINT WHERE 'P ENABLE SEARCH'' IS ASSERTED.
                                        AFTER WAITING A SUFFICIENT PERIOD AND VERIFYING THAT OPI
                                        IS NOT SET, THE TEST ENABLES SEARCH TIMEOUT AND VERIFIES
                                        THAT OPI SETS.

                                        PROBABLE FAULT:

                                        1.  CS MODULE

                                                NOTE:   IT IS ASSUMED THAT THE ''SET OPI TEST'' HAS
                                                        ALREADY PASSED, THUS MAKING THE IF MODULE
                                                        AN IMPROBABLE FAULT.
```

```
600
601
602
603
604
605                         SET DTE TEST
606
607                                 PURPOSE:
608
609                                      IN ADDITION TO VERIFYING THAT 'DRIVE TIMING ERROR'' CAN
610                                 BE SET BY THE CS MODULE, THIS TEST ALSO VERIFIES
611
612                                 *    THAT 'MAINTENANCE SECTOR COMPARE'' IS NOT STUCK AT ONE
613                                      OR ZERO.
614
615                                 *    THAT 'ENABLE SEARCH'' IS NOT STUCK AT ONE OR ZERO.
616
617                                 PROCEDURE:
618
619                                      (1)  INITIALIZE AND VERIFY THAT 'DTE'' IS RESET, THEN
620                                 SET MAINTENANCE INDEX PULSE AND VERIFY THAT DTE IS STILL
621                                 RESET.  THIS TEST WILL INSURE THAT THE SECTOR COMPARE FLOP
622                                 IS NOT STUCK AT ONE.
623
624                                      (2)  EXECUTE A DATA COMMAND IN MAINTENANCE MODE TO
625                                 THE POINT WHERE SEARCH IS ENABLED, I.E., P EN SEARCH H IS
626                                 IS ACTIVE.  SET AND RESET THE SECTOR PULSE TO SET ''CS3 EN
627                                 SEARCH'' FLOP, AND CLOCK ''CSS SECTOR COMPARE'' FLOP WHICH
628                                 SHOULD NOT SET.  SET SECTOR PULSE AND VERIFY THAT DTE IS
629                                 RESET.  THIS TEST FAILS IF J INPUT TO SECTOR COMPARE FLOP
630                                 IS STUCK AT ONE.
631
632                                      REPEAT, BUT SET MAINTENANCE SECTOR COMPARE AND VERIFY
633                                 THAT DTE SETS.  THIS TEST FAILS IF MAINTENANCE SECTOR
634                                 COMPARE IS STUCK AT ZERO.
635
636                                 PROBABLE FAULT:
637
638                                 1.  CS MODULE
639
640                                 2.  DS MODULE
641
642
643
644
645
646                         FORMAT CHANGE TEST
647
648                                 PURPOSE:
649
650                                      TO VERIFY THAT A CHANGE IN FORMAT INHIBITS SEARCH
651                                 ENABLE UNTIL THE NEXT INDEX PULSE.
652
653                                 PROCEDURE:
654
655                                      THE TEST WILL USE 'DTE'' FOR VISIBILITY OF ''CS3 EN
```

SEARCH H''.

THE FOLLOWING STEPS ARE EXECUTED:

(1) INITIALIZE AND SET THE FORMAT TO 18 BIT MODE TO SET ''CS3 FMT CHANGE'' FLOP.

(2) EXECUTE A DATA COMMAND TO THE POINT WHERE SEARCH IS ENABLED BY THE SEQUENCER.

(3) SET 'MAINTENANCE SECTOR COMPARE'', THEN SET 'MAINTENANCE SECTOR PULSE'' TO CLOCK ''CS3 EN SEARCH'' FLOP WHICH SHOULD NOT SET BECAUSE OF THE FORMAT CHANGE.

(4) RESET SECTOR PULSE TO CLOCK ''CS5 SECTOR COMPARE'' FLOP WHICH WILL NOT SET IF ''CS3 EN SEARCH H'' IS INACTIVE.

(5) SET SECTOR PULSE AND VERIFY THAT DRIVE TIMING ERROR IS RESET.

(6) SET AND RESET INDEX PULSE TOCLEAR THE FORMAT CHANGE FLOP.

(7) SET AND RESET SECTOR PULSE TO SET ''CS3 EN SEARCH'' FLOP AND ''CS5 SECTOR COMPARE'' FLOP.

(8) SET SECTOR COMPARE AND VERIFY THAT DTE IS SET.

REPEAT THE TEST WITH A FORMAT CHANGE FROM 18 BIT MODE TO 16 BIT MODE.

PROBABLE FAULT:

1. CS MODULE

PROM STROBE TEST

PURPOSE:

TO VERIFY THAT WORD CLOCK AND PROM STROBE CAN BE MANIPULATED IN MAINTENANCE MODE.

PROCEDURE:

INITIALIZE AND SET 16 BIT MODE, THEN SEQUENCE THE MAINTENANCE CLOCK UNTIL PROM STROBE SETS. ISSUE -- MORE MAINTENANCE CLOCK PULSES AND VERIFY THAT PROM STROBE RESETS.

PROBABLE FAULT:

1. CS MODULE

2. DS MODULE

SYNC WORD COUNT INHIBIT TEST

PURPOSE:

TO VERIFY THE FOLLOWING DURING READ COMMAND:

* THAT ''CS4 P LFS'' (LOOKING FOR SYNC) GOES ACTIVE.

* THAT ''LOOKING FOR SYNC'' INHIBITS THE WORD COUNT

PROCEDURE:

A READ COMMAND IS SETUP AND EXECUTED TO THE POINT WHERE ''LOOKING FOR SYNC'' SHOULD BE ACTIVE, WITH THE PROGRAM VERIFYING THE TRANSITION OF THE SIGNAL. THE PROGRAM THEN SUPPLIES A SERIES OF BIT CLOCKS AND VERIFIES THAT ''PROM STROBE'' NEVER GOES ACTIVE.

PROBABLE FAULT:

1. CS MODULE IF LFS FAILT,

2. DS MODULE IF PROM STROBE FAILS.

SYNC DETECTION TEST

PURPOSE:

TO VERIFY THAT THE APPROPRIATE SYNC PATTERN IS RECOGNIZED.

PROCEDURE:

THE TEST EXECUTES A READ COMMAND IN MAINTENANCE MODE, USING BIT 10 OF THE MAINTENANCE REGISTER (RMMR1) TO SIMULATE THE SYNC BIT STREAM, AND USING PROM STROBE TO DETERMINE IF THE SYNC PATTERN HAS BEEN DETECTED.

THE SYNC PATTERN IS 00011001, WITH THE LEFT MOST BIT REPRESENTING THE LAST BIT OF THE STREAM.

PROBABLE FAULT:

1. DS MODULE

2. CS MODULE

ABORT SYNC DETECTION TEST

PURPOSE:

TO VERIFY THAT ''WORD COUNT INHIBIT'' IS RESET IF A ''DRIVE TIMING ERROR'' OCCURS DURING SYNC DETECTION.

PROCEDURE:

A READ COMMAND IS INITIATED IN MAINTENANCE MODE. WHEN ''LOOKING FOR SYNC'' GOES ACTIVE, THE TEST FORCES A DRIVE TIMING ERROR AND USES PROM STROBE TO VERIFY THAT ''WORD COUNT INHIBIT'' HAS BEEN RESET.

PROBABLE FAULT:

1. DS MODULE

SYNC GENERATION TEST

PURPOSE:

TO VERIFY THAT THE APPROPRIATE SYNC PATTERN IS GENERATED DURING A FORMAT OPERATION.

PROCEDURE:

THE TEST EXECUTES A WRITE HEADER AND DATA COMMAND IN MAINTENANCE MODE, AND VERIFIES THAT THE CORRECT SYNC PATTERN IS GENERATED BY MONITORING WRITE DATA AT THE MAINTENANCE REGISTER (RMMR1).

PROBABLE FAULT:

1. DS MODULE

WRITE HEADER TEST

PRUPOSE:

TO TEST THE OPERATION OF (1) THE DATA BUFFER AND SHIFT REGISTER AS WELL AS (2) THE ECC GENERATION DURING WRITE.

PROCEDURE:

A WRITE HEADER AND DATA COMMAND IS EXECUTED IN MAINTENANCE MODE. THE TEST VERIFIES HEADER WORDS ONE AND TWO AND THE CRC CHARACTER USING THE WRITE DATA BIT OF THE MAINTENANCE REGISTER.

THE TEST USES CYLINDER 822, TRACK 4, SECTOR 31 AND 16 BIT FORMAT, WHICH CORRESPONS TO THE FOLLOWING HEADER:

WORD 1 - 1101001100110110

WORD 2 - 0000010000011111

PROBABLE FAULT:

DS MODULE OR MASSBUS MODULE

HEADER COMPARE TEST

PURPOSE:

TO CHECK THE OPERATION OF (1) THE SHIFT REGISTER AND DATA BUFFER AS WELL AS THE (2) CRC GENERATOR DURING READ.

PROCEDURE:

THE TEST EXECUTES A READ HEADER AND DATA COMMAND IN MAINTENANCE MODE USING BIT 10 OF THE MAINTENANCE REGISTER (RMMR1) TO SIMULATE THE DATA BITS FOR THE FIRST AND SECOND HEADER WORDS AS WELL AS THE CRC CHARACTER. THE CRC CHARACTER IS IS FAULTED, AND THE TEST VERIFIES THAT A CRC ERROR IS DETECTED. ADDITIONALLY, THE TEST VERIFIES THAT HEADER WORDS ONE AND TWO ARE CORRECTLY TRANSFERD TO MEMORY.
THE TEST USES THE SAME HEADER AS IN THE PREVIOUS TEST EXCEPT THAT BIT 15 IS INVERTED.

PROBABLE FAULT:
DS OR IF MODULE IF CRC ERROR NOT DETECTED;
DS OR MASSBUS MODULE IF DATA INCORRECT

ECC GENERATION TEST

PURPOSE:

TO CHECK ECC OPERATION DURING WRITE.

PROCEDURE:

A WRITE DATA COMMAND IS EXECUTED IN MAINTENANCE MODE. ALL ONES DATA FIELD IS USED, AND THE TEST VERIFIES THE ECC CHARACTER VIA THE WRITE DATA BIT OF THE MAINTENANCE REGISTER. THE DATA FIELD IS NOT VERIFIED.

```
                                        PROBABLE FAULT.
                                        DS MODULE

                                ECC     DETECTION TEST

                                        PURPOSE:

                                                TO CHECK THE ECC GENERATION DURING READ.

                                        PROCEDURE:

                                                THE TEST EXECUTES A READ DATA COMMAND IN MAINTENANCE
                                        MODE, AN ALL ONES DATA FIELD IS USED, HOWEVER THE LAST DATA
                                        WORD IS FAULTED.

                                                THE TEST VERIFIES (1) THAT AN ECC ERROR IS DETECTED
                                        AND THAT (2) THE POSITION AND PATTERN REGISTERS ARE VALID.

                                        PROBABLE FAULT:

                                        DS MODULE

                                \
```

```
                                ;PROGRAM REVISION #001
                                .TITLE  CZRMKB0 RM03/2 DSKLS PRT 2
                                ;*COPYRIGHT (C) 1978
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY DOUG RIIKONEN
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
000001                          $TN=1
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                ;*         7            TN128
                                ;*         6            TN64
                                ;*         5            TN32
                                ;*         4            TN16
                                ;*         3            TN8
                                ;*         2            TN4
                                ;*         1            TN2
                                ;*         0            TN1
                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2      %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
```

```
        000004          R4=     %4              ; GENERAL REGISTER
        000005          R5=     %5              ;;GENERAL REGISTER
        000006          R6=     %6              ;;GENERAL REGISTER
        000007          R7=     %7              ;;GENERAL REGISTER
        000006          SP=     %6              ;;STACK POINTER
        000007          PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
        000000          PR0=    0               ;;PRIORITY LEVEL 0
        000040          PR1=    40              ;;PRIORITY LEVEL 1
        000100          PR2=    100             ;;PRIORITY LEVEL 2
        000140          PR3=    140             ;;PRIORITY LEVEL 3
        000200          PR4=    200             ;;PRIORITY LEVEL 4
        000240          PR5=    240             ;;PRIORITY LEVEL 5
        000300          PR6=    300             ;;PRIORITY LEVEL 6
        000340          PR7=    340             ;;PRIORITY LEVEL 7

                        ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
        100000          SW15=   100000
        040000          SW14=   40000
        020000          SW13=   20000
        010000          SW12=   10000
        004000          SW11=   4000
        002000          SW10=   2000
        001000          SW09=   1000
        000400          SW08=   400
        000200          SW07=   200
        000100          SW06=   100
        000040          SW05=   40
        000020          SW04=   20
        000010          SW03=   10
        000004          SW02=   4
        000002          SW01=   2
        000001          SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
        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
```

```
1016    000040                  BIT05=  40
1017    000020                  BIT04=  20
1018    000010                  BIT03=  10
1019    000004                  BIT02=  4
1020    000002                  BIT01=  2
1021    000001                  BIT00=  1
1022                            .EQUIV  BIT09,BIT9
1023                            .EQUIV  BIT08,BIT8
1024                            .EQUIV  BIT07,BIT7
1025                            .EQUIV  BIT06,BIT6
1026                            .EQUIV  BIT05,BIT5
1027                            .EQUIV  BIT04,BIT4
1028                            .EQUIV  BIT03,BIT3
1029                            .EQUIV  BIT02,BIT2
1030                            .EQUIV  BIT01,BIT1
1031                            .EQUIV  BIT00,BIT0
1032
1033                            ;*BASIC ''CPU'' TRAP VECTOR ADDRESSES
1034    000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
1035    000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
1036    000014                  TBITVEC=14              ;;''T'' BIT
1037    000014                  TRTVEC= 14              ;;TRACE TRAP
1038    000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
1039    000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1040    000024                  PWRVEC= 24              ;;POWER FAIL
1041    000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
1042    000034                  TRAPVEC=34              ;;''TRAP'' TRAP
1043    000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
1044    000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
1045    000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1046
1047                            .SBTTL  RM03 REGISTER BIT DEFINITIONS
1048
1049                            ;RMCS1  CONTROL STATUS REGISTER
1050
1051    004000                  DVA     =       BIT11           ;DEVICE AVAILABLE-READ ONLY
1052    000040                  F4      -       BIT05           ;FUNCTION CODE
1053    000020                  F3      =       BIT04           ;FUNCTION CODE
1054    000010                  F2      -       BIT03           ;FUNCTION CODE
1055    000004                  F1      =       BIT02           ;FUNCTION CODE
```

```
1056    000002          FO      =       BIT01           ;FUNCTION CODE
1057    000001          GO      =       BIT00           ;GO BIT
1058    000077          FNCMSK  =       000077          ;FUNCTION CODE MASK
1059
1060                    ;FUNCTION CODES (BITS 01-05 OF RMCS1)
1061
1062    000000          NOP     =       000000          ;NOP COMMAND
1063    000002          ILF02   =       000002          ;ILLEGAL COMMAND
1064    000004          SEEK    =       000004          ;SEEK COMMAND
1065    000006          RECAL   =       000006          ;RECALIBRATE COMMAND
1066    000010          DRVCLR  =       000010          ;DRIVE CLEAR COMMAND
1067    000012          RLEASE  =       000012          ;RELEASE COMMAND
1068    000014          OFFSET  =       000014          ;OFFSET COMMAND
1069    000016          RTC     =       000016          ;RETURN TO CENTERLINE COMMAND
1070    000020          RIP     =       000020          ;READ IN PRESET COMMAND
1071    000022          PAKACK  =       000022          ;PACK ACKNOWLEDGE COMMAND
1072    000022          PACACK  =       PAKACK
1073    000024          ILF24   =       000024          ;ILLEGAL COMMAND
1074    000026          ILF26   =       000026          ;ILLEGAL COMMAND
1075    000030          SEARCH  =       000030          ;SEARCH COMMAND
1076    000030          ILF30   =       000030          ;ILLEGAL COMMAND
1077    000032          ILF32   =       000032          ;ILLEGAL COMMAND
1078    000034          ILF34   =       000034          ;ILLEGAL COMMAND
1079    000036          ILF36   =       000036          ;ILLEGAL COMMAND
1080    000040          ILF40   =       000040          ;ILLEGAL COMMAND
1081    000042          ILF42   =       000042          ;ILLEGAL COMMAND
1082    000044          ILF44   =       000044          ;ILLEGAL COMMAND
1083    000046          ILF46   =       000046          ;ILLEGAL COMMAND
1084    000050          WCD     =       000050          ;WRITE CHECK DATA COMMAND
1085    000052          WCH     =       000052          ;WRITE CHECK HEADER AND DATA
1086    000054          ILF54   =       000054          ;ILLEGAL COMMAND
1087    000056          ILF56   =       000056          ;ILLEGAL COMMAND
1088    000060          WD      =       000060          ;WRITE DATA COMMAND
1089    000062          WH      =       000062          ;WRITE HEADER AND DATA COMMAND
1090    000064          ILF64   =       000064          ;ILLEGAL COMMAND
1091    000066          ILF66   =       000066          ;ILLEGAL COMMAND
1092    000070          RD      =       000070          ;READ DATA COMMAND
1093    000072          RH      =       000072          ;READ HEADER AND DATA COMMAND
1094    000074          ILF74   =       000074          ;ILLEGAL COMMAND
1095    000076          ILF76   =       000076          ;ILLEGAL COMMAND
1096
1097                    ;RMDA   DISK ADDRESS REGISTER
1098
1099    002000          TA4     =       BIT10           ;TRACK ADDRESS 4
1100    001000          TA2     =       BIT09           ;TRACK ADDRESS 2
1101    000400          TA1     =       BIT08           ;TRACK ADDRESS 1
1102    000020          SA16    =       BIT04           ;SECTOR ADDRESS 16
1103    000010          SA8     =       BIT03           ;SECTOR ADDRESS 8
1104    000004          SA4     =       BIT02           ;SECTOR ADDRESS 4
1105    000002          SA2     =       BIT01           ;SECTOR ADDRESS 2
1106    000001          SA1     =       BIT00           ;SECTOR ADDRESS 1
1107
1108                    ;TRACK,SECTOR MASKS
1109
1110    003400          TADMSK  =       003400          ;TRACK ADDRESS MASK
1111    000037          SADMSK  =       000037          ;SECTOR ADDRESS MASK
```

```
1112
1113                            ;RMDS   DRIVE STATUS REGISTER
1114
1115    100000                  ATA     =       BIT15           ;ATTENTION ACTIVE
1116    040000                  ERR     =       BIT14           ;COMPOSITE ERROR
1117    020000                  PIP     =       BIT13           ;POSITIONING IN PROGRESS
1118    010000                  MOL     -       BIT12           ;MEDIUM ON LINE
1119    004000                  WRL     =       BIT11           ;WRITE LOCK
1120    002000                  LBT             BIT10           ;LAST BLOCK TRANSFERRED
1121    001000                  PGM             BIT09           ;PROGRAMMABLE
1122    000400                  DPR     =       BIT08           ;DRIVE PRESENT
1123    000200                  DRY             BIT07           ;DRIVE READY
1124    000100                  VV      -       BIT06           ;VOLUME VALID
1125    000001                  OM      =       BIT00           ;OFFSET MODE ACTIVE
1126
1127                            ;RMER1  ERROR REGISTER #1
1128
1129    100000                  DCK             BIT15           ;DATA CHECK ERROR
1130    040000                  UNS     =       BIT14           ;DRIVE UNSAFE
1131    020000                  OPI     -       BIT13           ;OPERATION INCOMPLETE
1132    010000                  DTE     -       BIT12           ;DRIVE TIMING ERROR
1133    004000                  WLE     =       BIT11           ;WRITE LOCK ERROR
1134    002000                  IAE     -       BIT10           ;INVALID ADDRESS ERROR
1135    001000                  AOE     -       BIT09           ;ADDRESS OVERFLOW ERROR
1136    000400                  HCRC    =       BIT08           ;HEADER CRC ERROR
1137    000200                  HCE     =       BIT07           ;HEADER COMPARE ERROR
1138    000100                  ECH     -       BIT06           ;ECC 'HARD'' ERROR
1139    000040                  WCF             BIT05           ;WRITE CLOCK FAILURE
1140    000020                  FER             BIT04           ;FORMAT ERROR
1141    000010                  PAR             BIT03           ;PARITY ERROR
1142    000004                  RMR     =       BIT02           ;REGISTER MODIFICATION REFUSED
1143    000002                  ILR     =       BIT01           ;ILLEGAL REGISTER
1144    000001                  ILF     -       BIT00           ;ILLEGAL FUNCTION
1145
1146    115760                  NDTMSK  =       DCK!DTE!WLE!AOE!HCRC!HCE!ECH!WCF!FER
1147                            ;'NDTMSK'' IS USED TO MASK ERROR REGISTER 1 DURING NON - DATA
1148                            ;COMMANDS, I.E., HOUSEKEEPING AND POSITIONING COMMANDS
1149
1150                            ;RMAS   ATTENTION SUMMARY REGISTER
1151
1152    000377                  ATNMSK  -       377             ;MASK FOR ATTENTION BITS
1153
1154                            ;RMLA   LOOK AHEAD REGISTER
1155
1156    002000                  SC4     =       BIT10           ;SECTOR COUNT = 16
1157    001000                  SC3     =       BIT09           ;SECTOR COUNT = 8
1158    000400                  SC2             BIT08           ;SECTOR COUNT = 4
1159    000200                  SC1             BIT07           ;SECTOR COUNT = 2
1160    000100                  SC0             BIT06           ;SECTOR COUNT = 1
1161
1162    003700                  SCTMSK          003700          ;SECTOR COUNT MASK
1163
1164                            ;RMMR   MAINTENANCE REGISTER
1165
1166                            ;       WRITE ONLY BITS
1167
```

```
1168            100000                  DBCK    -       BIT15           ;DEBUG CLOCK
1169            040000                  DBEN    -       BIT14           ;DEBUG CLOCK ENABLE
```

```
1170            020000          DEBL    -       BIT13           ;DIAGNOSTIC END OF BLOCK
1171            010000          DTO     =       BIT12           ;DIAGNOSTIC TIMEOUT
1172            004000          MCLK    =       BIT11           ;MAINTENANCE CLOCK
1173            002000          MRD     =       BIT10           ;READ DATA
1174            001000          MUR     =       BIT09           ;UNIT READY
1175            000400          MOC     =       BIT08           ;ON CYLINDER
1176            000200          MSER    =       BIT07           ;SEEK ERROR
1177            000100          MDF     -       BIT06           ;DRIVE FAULT
1178            000040          MS      -       BIT05           ;SECTOR PULSE
1179            000010          MWP     =       BIT03           ;WRITE PROTECT
1180            000004          MI      =       BIT02           ;INDEX PULSE
1181            000002          MSC             BIT01           ;SECTOR COMPARE
1182            000001          DMD             BIT00           ;DIAGNOSTIC MODE
1183            051401          MR1AAA  =       DMD!MUR!DBEN.MOC!DTO
1184
1185                            ;       READ ONLY BITS
1186
1187            100000          OCC     =       BIT15           ;OCCUPIED
1188            040000          RG      =       BIT14           ;RUN AND GO
1189            020000          EBL     =       BIT13           ;END OF BLOCK
1190            010000          REX     =       BIT12           ;EXCEPTION
1191            004000          ESRC    =       BIT11           ;ENABLE SEARCH
1192            002000          PLFS    =       BIT10           ;LOOKING FOR SYNC
1193            001000          ECRC    =       BIT09           ;ENABLE CRC OUT
1194            000400          PDA     -       BIT08           ;DATA AREA
1195            000200          PHA     =       BIT07           ;HEADER AREA
1196            000100          CONT    =       BIT06           ;CONTINUE
1197            000040          WC      -       BIT05           ;WORD CLOCK
1198            000020          EECC    -       BIT04           ;ENABLE ECC OUT
1199            000010          MWD     -       BIT03           ;WRITE DATA BIT
1200            000004          LS      =       BIT02           ;LAST SECTOR
1201            000002          LST     -       BIT01           ;LAST SECTOR AND TRACK
1202            000001          DMD     -       BIT00           ;DIAGNOSTIC MODE
1203
1204                            ;RMDT   DRIVE TYPE REGISTER
1205
1206            100000          NSA     -       BIT15           ;NOT SECTOR ADDRESSED=0
1207            040000          TAP     -       BIT14           ;TAPE DRIVE = 0
1208            020000          MOH     -       BIT13           ;MOVING HEAD = 1
1209            004000          DRQ     -       BIT11           ;DRIVE REQUEST REQUIRED
1210
1211            020024          SNGPRT          020024          ;SINGLE PORT DRIVE TYPE
1212            024024          DULPRT  -       024024          ;DUAL PORT DRIVE TYPE
1213
1214                            ;RMOF   OFFSET REGISTER
1215
1216            010000          FMT16   -       BIT12           ;16 BIT WORD FORMAT
1217            004000          ECI     -       BIT11           ;ECC INHIBIT
1218            002000          HCI     =       BIT10           ;HEADER COMPARE INHIBIT
1219            000200          OFD     =       BIT07           ;OFFSET FORWARD
1220            161577          XNUOF   =       161577          ;UNUSED BITS OF RMOF
1221
1222
1223                            ;RMDC   DESIRED CYLINDER ADDRESS REGISTER
1224            001777          CYLMSK  =       1777            ;MASK FOR CYLINDER ADDRESS
1225            176000          XNUDC   =       176000          ;UNUSED BITS OF RMDC
```

```
1226
1227                                    ;RMMR2  MAINTENANCE REGISTER #2
1228
1229                                    ;       READ ONLY BITS
1230            100000                  RQA     =       BIT15           ;PORT A REQUEST
1231            040000                  RQB     =       BIT14           ;PORT B REQUEST
1232            020000                  TAG             BIT13           ;TAG CONTROL
1233            010000                  TST     =       BIT12           ;COMMAND SEQUENCE TEST BIT
1234            004000                  CC      -       BIT11           ;CONTROL OR CYLINDER TAG
1235            002000                  CH              BIT10           ;CONTROL OR HEAD TAG
1236            001000                  BB09    =       BIT09           ;TAG BUS
1237            000400                  BB08    -       BIT08           ;TAG BUS
1238            000200                  BB07    -       BIT07           ;TAG BUS
1239            000100                  BB06    =       BIT06           ;TAG BUS
1240            000040                  BB05    -       BIT05           ;TAG BUS
1241            000020                  BB04    =       BIT04           ;TAG BUS
1242            000010                  BB03    =       BIT03           ;TAG BUS
1243            000004                  BB02    =       BIT02           ;TAG BUS
1244            000002                  BB01    =       BIT01           ;TAG BUS
1245            000001                  BB00    =       BIT00           ;TAG BUS
1246
1247
1248                                    ;RMER2  ERROR REGISTER 2
1249
1250            100000                  BSE     =       BIT15           ;BAD SECTOR ERROR
1251            040000                  SKI     =       BIT14           ;SEEK INCOMPLETE
1252            020000                  OPE     -       BIT13           ;OPERATOR PLUG ERROR
1253            010000                  IVC     =       BIT12           ;INVALID COMMAND ERROR
1254            004000                  LSC     -       BIT11           ;LOSS OF SYSTEM CLOCK
1255            002000                  LBC     =       BIT10           ;LOSS OF BIT CLOCK
1256            000200                  DVC     =       BIT07           ;DEVICE CHECK
1257            000010                  DPE     -       BIT03           ;DATA PARITY ERROR
1258            001567                  XNUER2  =       001567          ;UNUSED BITS OF RMER2
1259
1260                                    .SBTTL  PROGRAM MNEMONICS
1261
1262            100000                  MSE     -       BIT15           ;MANUFACTURING DETECTED SECTOR ERROR
1263            040000                  USE     =       BIT14           ;USER DETECTED SECTOR ERROR
1264
1265                                    .SBTTL  RM03 REGISTER INDEX VALUES
1266
1267            000000                  RMCS1   =       00              ;CONTROL STATUS REGISTER
1268            000006                  RMDA    =       06              ;DISK ADDRESS REGISTER
1269            000012                  RMDS    =       12              ;DRIVE STATUS REGISTER
1270            000014                  RMER1   =       14              ;ERROR REGISTER 1
1271            000016                  RMAS    =       16              ;ATTENTION SUMMARY REGISTER
1272            000020                  RMLA    =       20              ;LOOK AHEAD REGISTER
1273            000024                  RMMR1   =       24              ;MAINTENANCE REGISTER
1274            000026                  RMDT    =       26              ;DRIVE TYPE REGISTER
1275            000030                  RMSN    -       30              ;SERIAL NUMBER REGISTER
1276            000032                  RMOF    =       32              ;OFFSET REGISTER
1277            000034                  RMDC            34              ;DESIRED CYLINDER REGISTER
1278            000036                  RMHR    =       36              ;HOLDING REGISTER
1279            000040                  RMMR2   =       40              ;MAINTENANCE REGISTER 2
1280            000042                  RMER2   =       42              ;ERROR REGISTER 2
1281            000044                  RMEC1   =       44              ;ECC POSITION REGISTER
```

```
1282    000046                  RMEC2   =       46              ;ECC PATTERN REGISTER
1283    000050                  ILRG50  =       50              ;ILLEGAL REGISTER 50
1284    000052                  ILRG52  =       52              ;ILLEGAL REGISTER 52
1285    000054                  ILRG54  =       54              ;ILLEGAL REGISTER 54
1286    000056                  ILRG56  =       56              ;ILLEGAL REGISTER 56
1287    000060                  ILRG60  =       60              ;ILLEGAL REGISTER 60
1288    000062                  ILRG62  =       62              ;ILLEGAL REGISTER 62
1289    000064                  ILRG64  =       64              ;ILLEGAL REGISTER 64
1290    000066                  ILRG66  =       66              ;ILLEGAL REGISTER 66
1291    000070                  ILRG70  =       70              ;ILLEGAL REGISTER 70
1292    000072                  ILRG72  =       72              ;ILLEGAL REGISTER 72
1293    000074                  ILRG74  =       74              ;ILLEGAL REGISTER 74
1294    000076                  ILRG76  =       76              ;ILLEGAL REGISTER 76
1295
1296
1297    000077                  IDXMSK  =       77              ;MASK FOR REGISTER INDEX NUMBER
1298
1299                            .SBTTL  RH CONTROLLER REGISTER BIT DEFINITIONS
1300
1301                            ;RMCS1  CONTROL STATUS REGISTER #1
1302
1303    100000                  SC      =       BIT15           ;SPECIAL CONDITION-READ ONLY
1304    040000                  TRE     =       BIT14           ;TRANSFER ERROR
1305    020000                  MCPE    =       BIT13           ;MASSBUS CONTROL BUS PARITY
1306                                                            ;ERROR-READ ONLY
1307    002000                  PSEL    =       BIT10           ;PORT B SELECT
1308    001000                  A17     =       BIT09           ;ADDRESS EXTENSION
1309    000400                  A16     =       BIT08           ;ADDRESS EXTENSION
1310    000200                  RDY     -       BIT07           ;READY-READ ONLY
1311    000100                  IE      =       BIT06           ;INTERRUPT ENABLE
1312
1313                            ;RMCS2  RH CONTROL STATUS REGISTER #2
1314
1315    100000                  DLT     =       BIT15           ;DATA LATE-READ ONLY
1316    040000                  WCE     =       BIT14           ;WRITE CHECK ERROR-READ ONLY
1317    020000                  UPE     -       BIT13           ;UNIBUS PARITY ERROR
1318    010000                  NED     -       BIT12           ;NONEXISTANT DRIVE-READ ONLY
1319    004000                  NEM     =       BIT11           ;NONEXISTANT MEMORY-READ ONLY
1320    002000                  PGE     =       BIT10           ;PROGRAM ERROR-READ ONLY
1321    001000                  MXF     -       BIT09           ;MISSED TRANSFER
1322    000400                  MDPE    =       BIT08           ;MASSBUS DATA BUS PARITY
1323                                                            ;ERROR-READ ONLY
1324    000200                  OR              BIT07           ;OUTPUT READY-READ ONLY
1325    000100                  IR      =       BIT06           ;INPUT READY-READ ONLY
1326    000040                  CLR     -       BIT05           ;CONTROLLER CLEAR
1327    000020                  PAT     -       BIT04           ;PARITY TEST
1328    000010                  BAI     =       BIT03           ;UNIBUS ADDRESS INCREMENT
1329                                                            ;INHIBIT
1330    000004                  U2      =       BIT02           ;UNIT SELECT
1331    000002                  U1      =       BIT01           ;UNIT SELECT
1332    000001                  U0      =       BIT00           ;UNIT SELECT
1333
1334                            ;UNIT SELECT MASK
1335    000007                  UNTMSK  =       7               ;UNIT SELECT MASK
1336
1337                            ;RMCS3  RH70 CONTROL STATUS REGISTER #3
```

```
1338            100000                  APE     =       BIT15           ;ADDRESS PARITY ERROR
1339            040000                  DPEHI   =       BIT14           ;DATA PARITY ERROR HIGH WORD
1340            020000                  DPELO   =       BIT13           ;DATA PARITY ERROR LOW WORD
1341            010000                  WCEHI   =       BIT12           ;WRITE CHECK ERROR HIGH WORD
1342            004000                  WCELO   =       BIT11           ;WRITE CHECK ERROR LOW WORD
1343            002000                  DBL     =       BIT10           ;DOUBLE WORD TRANSFER
1344            000100                  IE      =       BIT06           ;INTERRUPT ENABLE
1345            000010                  IPCK3   =       BIT03           ;INVERT PARITY CHECK
1346            000004                  IPCK2   =       BIT02           ;INVERT PARITY CHECK
1347            000002                  IPCK1   =       BIT01           ;INVERT PARITY CHECK
1348            000001                  IPCK0   =       BIT00           ;INVERT PARITY CHECK
1349                                    .SBTTL  RH CONTROLLER REGISTER INDEX VALUES
1350
1351            000000                  RMCS1   =       00              ;CONTROL, STATUS REGISTER
1352            000002                  RMWC    =       02              ;WORD COUNT REGISTER
1353            000004                  RMBA    =       04              ;BUS ADDRESS REGISTER
1354            000010                  RMCS2           10              ;CONTROLLER STATUS REGISTER
1355            000022                  RMDB    =       22              ;DATA BUFFER
1356            000050                  RMBAE   =       50              ;BUS ADDRESS EXTENSION
1357            000052                  RMCS3   =       52              ;CONTROL STATUS REGISTER #3
1358
1359            176700                  ABASE   =       176700          ;UNIBUS ADDRESS
1360            120254                  AVECT1  =       120254          ;UNIBUS VECTOR ADDRESS AND PRIORITY
1361
1362
1363                                    .SBTTL  TRAP CATCHER
1364
1365            000000                          .=0
1366                                    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ''.+2,HALT''
1367                                    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1368                                    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1369            000174                          .=174
1370    000174  000000                 DISPREG: .WORD  0                ;;SOFTWARE DISPLAY REGISTER
1371    000176  000000                 SWREG:   .WORD  0                ;;SOFTWARE SWITCH REGISTER
1372
1373
1374                                    .SBTTL  ACT11 HOOKS
1375
1376                                    ;;*****************************************************************
1377                                    ;HOOKS REQUIRED BY ACT11
1378            000200                          $SVPC=.                 ;SAVE PC
1379            000046                          .=46
1380    000046  021666                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1381            000052                          .=52
1382    000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
1383            000200                          .=$SVPC                 ;; RESTORE PC
1384
1385                                    .SBTTL  STARTING ADDRESS
1386
1387                                    ;THE PROGRAM STARTS AT LOCATION 200
1388            000200                  .       =       200
1389    000200  000137  002632                  JMP     START           ;JUMP TO START OF PROGRAM
1390
1391            001100                  .=1100
1392                                    .SBTTL  APT PARAMETER BLOCK
1393
```

```
1394                                    ;;*****************************************************************
1395                                    ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1396                                    ;;*****************************************************************
1397            001100                          .$X=.    ;;SAVE CURRENT LOCATION
1398            000024                          .=24     ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1399  000024    000200                          200      ;;FOR APT START UP
1400            000044                          .=44     ;;POINT TO APT INDIRECT ADDRESS PNTR.
1401  000044    001100                          $APTHDR  ;;POINT TO APT HEADER BLOCK
1402            001100                          .=.$X    ;;RESET LOCATION COUNTER
1403                                    ;;*****************************************************************
1404                                    ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1405                                    ;INTERFACE SPEC.
1406
1407  001100                            $APTHD:
1408  001100    000000                  $HIBTS: .WORD    0          ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1409  001102    001222                  $MBADR: .WORD    $MAIL      ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1410  001104    000001                  $TSTM:  .WORD    1          ;;RUN TIM OF LONGEST TEST
1411  001106    000002                  $PASTM: .WORD    2          ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1412  001110    000002                  $UNITM: .WORD    2          ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1413  001112    000042                          .WORD    $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1414            001114                  TAGADR  =        .
```

```
                                .SBTTL  COMMON TAGS

                                ;;*****************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                        001114          .=TAGADR
001114                          $CMTAG:                         ;;START OF COMMON TAGS
001114  000000                          .WORD   0
001116     000          $TSTNM: .BYTE   0                       ;;CONTAINS THE TEST NUMBER
001117     000          $ERFLG: .BYTE   0                       ;;CONTAINS ERROR FLAG
001120  000000          $ICNT:  .WORD   0                       ;;CONTAINS SUBTEST ITERATION COUNT
001122  000000          $LPADR: .WORD   0                       ;;CONTAINS SCOPE LOOP ADDRESS
001124  000000          $LPERR: .WORD   0                       ;;CONTAINS SCOPE RETURN FOR ERRORS
001126  000000          $ERTTL: .WORD   0                       ;;CONTAINS TOTAL ERRORS DETECTED
001130     000          $ITEMB: .BYTE   0                       ;;CONTAINS ITEM CONTROL BYTE
001131     001          $ERMAX: .BYTE   1                       ;;CONTAINS MAX. ERRORS PER TEST
001132  000000          $ERRPC: .WORD   0                       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001134  000000          $GDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'GOOD' DATA
001136  000000          $BDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'BAD' DATA
001140  000000          $GDDAT: .WORD   0                       ;;CONTAINS 'GOOD' DATA
001142  000000          $BDDAT: .WORD   0                       ;;CONTAINS 'BAD' DATA
001144  000000                  .WORD   0                       ;;RESERVED--NOT TO BE USED
001146  000000                  .WORD   0
001150     000          $AUTOB: .BYTE   0                       ;;AUTOMATIC MODE INDICATOR
001151     000          $INTAG: .BYTE   0                       ;;INTERRUPT MODE INDICATOR
001152  000000                  .WORD   0
001154  177570          SWR:    .WORD   DSWR                    ;;ADDRESS OF SWITCH REGISTER
001156  177570          DISPLAY: .WORD  DDISP                   ;;ADDRESS OF DISPLAY REGISTER
001160  177560          $TKS:   177560                          ;;TTY KBD STATUS
001162  177562          $TKB:   177562                          ;;TTY KBD BUFFER
001164  177564          $TPS:   177564                          ;;TTY PRINTER STATUS REG. ADDRESS
001166  177566          $TPB:   177566                          ;;TTY PRINTER BUFFER REG. ADDRESS
001170     000          $NULL.  .BYTE   0                       ;;CONTAINS NULL CHARACTER FOR FILLS
001171     002          $FILLS: .BYTE   2                       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001172     012          $FILLC: .BYTE   12                      ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
001173     000          $TPFLG: .BYTE   0                       ;;'TERMINAL AVAILABLE'' FLAG (BIT<07>=0-YES)
001174  000000          $TMP0:  .WORD   0                       ;;USER DEFINED
001176  000000          $TMP1:  .WORD   0                       ;;USER DEFINED
001200  000000          $TMP2:  .WORD   0                       ;;USER DEFINED
001202  000000          $TMP3:  .WORD   0                       ;;USER DEFINED
001204  000000          $TMP4:  .WORD   0                       ;;USER DEFINED
001206  000000          $TIMES: 0                               ;;MAX. NUMBER OF ITERATIONS
001210  000000          $ESCAPE:0                               ;;ESCAPE ON ERROR ADDRESS
001212  177607  000377    $BELL:  .ASCIZ  <207><377><377>       ;;CODE FOR BELL
001216     077          $QUES:  .ASCII  /?/                     ;;QUESTION MARK
001217     015          $CRLF:  .ASCII  <15>                    ;;CARRIAGE RETURN
001220  000012          $LF:    .ASCIZ  <12>                    ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .NLIST ME
                                ;
                                ;
```

```
1471                                    .EVEN
1472  001222                            $MAIL:                          ;;APT MAILBOX
1473  001222  000000                    $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
1474  001224  000000                    $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
1475  001226  000000                    $TESTN: .WORD   ATESTN          ;;TEST NUMBER
1476  001230  000000                    $PASS:  .WORD   APASS           ;;PASS COUNT
1477  001232  000000                    $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
1478  001234  000000                    $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
1479  001236  000000                    $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
1480  001240  000000                    $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
1481  001242                            $ETABLE:                        ;;APT ENVIRONMENT TABLE
1482  001242     000                    $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
1483  001243     000                    $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
1484  001244  000000                    $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
1485  001246  000000                    $USWR:  .WORD   AUSWR           ;;USER SWITCHES
1486  001250  000000                    $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
1487                                    ;*                              BITS 15-11=CPU TYPE
1488                                    ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1489                                    ;*                                      11/70=06,PDQ=07,Q=10
1490                                    ;*                              BIT 10=REAL TIME CLOCK
1491                                    ;*                              BIT  9=FLOATING POINT PROCESSOR
1492                                    ;*                              BIT  8=MEMORY MANAGEMENT
1493  001252     000                    $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
1494  001253     000                    $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
1495                                    ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
1496                                    ;*                                      900 NSEC CORE=001
1497                                    ;*                                      300 NSEC BIPOLAR=002
1498                                    ;*                                      500 NSEC MOS=003
1499  001254  000000                    $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
1500                                    ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE'' ABOVE
1501  001256     000                    $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
1502  001257     000                    $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
1503  001260  000000                    $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
1504  001262     000                    $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
1505  001263     000                    $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
1506  001264  000000                    $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
1507  001266     000                    $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
1508  001267     000                    $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
1509  001270  000000                    $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
1510  001272  120254                    $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1511  001274  000000                    $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1512  001276  176700                    $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1513  001300  000000                    $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
1514  001302  000000                    $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
1515  001304  000000                    $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
1516  001306  000000                    $DDW0:  .WORD   ADDW0           ;;DEVICE DESCRIPTOR WORD#0
1517  001310  000000                    $DDW1:  .WORD   ADDW1           ;;DEVICE DESCRIPTOR WORD#1
1518  001312  000000                    $DDW2:  .WORD   ADDW2           ;;DEVICE DESCRIPTOR WORD#2
1519  001314  000000                    $DDW3:  .WORD   ADDW3           ;;DEVICE DESCRIPTOR WORD#3
1520  001316  000000                    $DDW4:  .WORD   ADDW4           ;;DEVICE DESCRIPTOR WORD#4
1521  001320  000000                    $DDW5:  .WORD   ADDW5           ;;DEVICE DESCRIPTOR WORD#5
1522  001322  000000                    $DDW6:  .WORD   ADDW6           ;;DEVICE DESCRIPTOR WORD#6
1523  001324  000000                    $DDW7:  .WORD   ADDW7           ;;DEVICE DESCRIPTOR WORD#7
1524  001326                            $ETEND:
1525                                    .MEXIT
1526
```

```
1527
1528
1529                                    .SBTTL  REGISTER INPUT BUFFER
1530
1531                                    ;       THE REGISTER INPUT BUFFER IS USED FOR STORING REGISTER
1532                                    ;CONTENTS AS THEY ARE READ FROM THE DEVICE.
1533
1534  001326  000000                    RMCS1I: .WORD                   ;CONTROL,STATUS REGISTER #1
1535  001330  000000                    RMWCI: .WORD                    ;WORD COUNT REGISTER
1536  001332  000000                    RMBAI: .WORD                    ;BUS ADDRESS REGISTER
1537  001334  000000                    RMDAI: .WORD                    ;DISK ADDRESS REGISTER
1538  001336  000000                    RMCS2I: .WORD                   ;CONTROL, STATUS REGISTER #2
1539  001340  000000                    RMDSI: .WORD                    ;DRIVE STATUS REGISTER
1540  001342  000000                    RMER1I: .WORD                   ;ERROR REGISTER 1
1541  001344  000000                    RMASI: .WORD                    ;ATTENTION SUMMARY REGISTER
1542  001346  000000                    RMLAI: .WORD                    ;LOOK AHEAD REGISTER
1543  001350  000000                    RMDBI: .WORD                    ;DATA BUFFER
1544  001352  000000                    RMMR1I: .WORD                   ;MAINTENANCE REGISTER #1
1545  001354  000000                    RMDTI: .WORD                    ;DRIVE TYPE REGISTER
1546  001356  000000                    RMSNI: .WORD                    ;SERIAL NUMBER REGISTER
1547  001360  000000                    RMOFI: .WORD                    ;OFFSET REGISTER
1548  001362  000000                    RMDCI: .WORD                    ;DESIRED CYLINDER REGISTER
1549  001364  000000                    RMHRI: .WORD                    ;HOLDING REGISTER
1550  001366  000000                    RMMR2I: .WORD                   ;MAINTENANCE REGISTER #2
1551  001370  000000                    RMER2I: .WORD                   ;ERROR REGISTER 2
1552  001372  000000                    RMEC1I: .WORD                   ;ECC POSITION REGISTER
1553  001374  000000                    RMEC2I: .WORD                   ;ECC PATTERN REGISTER
1554  001376  000000                    RMBAEI: .WORD           ;BUS ADDRESS EXTENSION REGISTER
1555  001400  000000                    RMCS3I: .WORD           ;CONTROL, STATUS REGISTER #3
1556
1557
1558
1559                                    .SBTTL  REGISTER OUTPUT BUFFER
1560
1561                                    ;       THE REGISTER OUTPUT BUFFER IS USED FOR ASSEMBLING DATA TO
1562                                    ;BE WRITTEN TO THE DEVICE.
1563
1564  001402  000000                    RMCS1O: .WORD                   ;CONTROL,STATUS REGISTER #1
1565  001404  000000                    RMWCO: .WORD                    ;WORD COUNT REGISTER
1566  001406  000000                    RMBAO: .WORD                    ;BUS ADDRESS REGISTER
1567  001410  000000                    RMDAO: .WORD                    ;DISK ADDRESS REGISTER
1568  001412  000000                    RMCS2O: .WORD                   ;CONTROL, STATUS REGISTER #2
1569  001414  000000                    RMDSO: .WORD                    ;DRIVE STATUS REGISTER
1570  001416  000000                    RMER1O: .WORD                   ;ERROR REGISTER 1
1571  001420  000000                    RMASO: .WORD                    ;ATTENTION SUMMARY REGISTER
1572  001422  000000                    RMLAO: .WORD                    ;LOOK AHEAD REGISTER
1573  001424  000000                    RMDBO: .WORD                    ;DATA BUFFER
1574  001426  000000                    RMMR1O: .WORD                   ;MAINTENANCE REGISTER #1
1575  001430  000000                    RMDTO: .WORD                    ;DRIVE TYPE REGISTER
1576  001432  000000                    RMSNO: .WORD                    ;SERIAL NUMBER REGISTER
1577  001434  000000                    RMOFO: .WORD                    ;OFFSET REGISTER
1578  001436  000000                    RMDCO: .WORD                    ;DESIRED CYLINDER REGISTER
1579  001440  000000                    RMHRO: .WORD                    ;HOLDING REGISTER
1580  001442  000000                    RMMR2O: .WORD                   ;MAINTENANCE REGISTER #2
1581  001444  000000                    RMER2O: .WORD                   ;ERROR REGISTER 2
1582  001446  000000                    RMEC1O: .WORD                   ;ECC POSITION REGISTER
```

```
1583  001450  000000                  RMEC20: .WORD                   ;ECC PATTERN REGISTER
1584  001452  000000                  RMBAE0: .WORD           ;BUS ADDRESS EXTENSION REGISTER
1585  001454  000000                  RMCS30: .WORD           ;CONTROL, STATUS REGISTER #3
1586
1587                                  .SBTTL  TEST QUE
1588
1589                                  ;       EACH WORD OF THE TEST QUE CONTAINS THE DEVICE NUMBER IN
1590                                  ;THE LOW BYTE AND THE ATTENTION BIT IN THE HIGH BYTE. A ZERO
1591                                  ;WORD IS A BLANK AND REPRESENTS THE END OF THE QUE.
1592
1593  001456  000012                  TSTQUE: .BLKW   10.     ;TEST QUE
1594
1595  001502  172540                  $LPCSR: .WORD   172540          ;KW11-P CONTROL + STATUS REGISTER
1596  001504  172542                  $LPCSB. .WORD   172542          ;KW11-P COUNT SET BUFFER
1597  001506  000104                  $LPVEC: .WORD   104             ;KW11-P INTERRUPT VECTOR
1598  001510  000106                          .WORD   106
1599  001512  177546                  $LLCSR: .WORD   177546          ;KW11-L CONTROL + STATUS REGISTER
1600  001514  000100                  $LLVEC: .WORD   100             ;KW11-L INTERRUPT VECTOR
1601  001516  000102                          .WORD   102
1602  001520  000000                  $PSW:   .WORD                   ;STORAGE FOR PRIORITY
1603  001522  000000                  TIME:   .WORD                   ;STORAGE FOR ELAPSED TIME
1604  001524  000000                  WATCH:  .WORD                   ;STORAGE FOR REMAINING TIME
1605  001526  000000                  CLOCK:  .WORD                   ;ADDRESS OF START CLOCK SUB
1606  001530  000000                  STOP:   .WORD                   ;ADDRESS OF STOP CLOCK SUB
1607
1608                                  ;PUT TAGS HERE
1609
```

```
1610                                    .SBTTL  ERROR POINTER TABLE
1611
1612                                    ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1613                                    ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1614                                    ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1615                                    ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1616                                    ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1617
1618                                    ;*      EM              ;;POINTS TO THE ERROR MESSAGE
1619                                    ;*      DH              ;;POINTS TO THE DATA HEADER
1620                                    ;*      DT              ;;POINTS TO THE DATA
1621                                    ;*      DF              ;;POINTS TO THE DATA FORMAT
1622
1623
1624  001532                            $ERRTB:
1625
1626
```

```
1627
1628                                    ;ERROR  1       CANNOT CLEAR NED STATUS
1629
1630  001532  031744                              EMT1
1631  001534  037650                              EHT1
1632  001536  037760                              EDT1
1633  001540  040014                              EFT1
1634
1635
1636                                    ;ERROR  2       CANNOT READ OR WRITE ANY DEVICE REG WITHOUT NED
1637
1638  001542  031752                              EMT2
1639  001544  037654                              EHT2
1640  001546  037762                              EDT2
1641  001550  040016                              EFT2
1642
1643
1644                                    ;ERROR  3       CANNOT WRITE/READ ONES TO ANY DEVICE REGISTER
1645
1646  001552  032000                              EMT3
1647  001554  000000                              0
1648  001556  000000                              0
1649  001560  000000                              0
1650
1651
1652                                    ;ERROR  4       CANNOT CLEAR ANY DEVICE REGISTER BITS W/MASSBUS INIT
1653
1654  001562  032020                              EMT4
1655  001564  000000                              0
1656  001566  000000                              0
1657  001570  000000                              0
1658
1659
1660                                    ;ERROR  5       CANNOT WRITE/READ ZEROS TO ALL BIT POSITIONS
1661
1662  001572  032042                              EMT5
1663  001574  037660                              EHT5
1664  001576  037764                              EDT5
1665  001600  040020                              EFT5
1666
1667
1668                                    ;ERROR  6       CANNOT WRITE/READ ONES TO ALL BIT POSITIONS
1669
1670  001602  032066                              EMT6
1671  001604  037660                              EHT5
1672  001606  037764                              EDT5
1673  001610  040020                              EFT5
1674
1675
1676                                    ;ERROR  7       CANNOT WRITE/READ SHIFTING ONE BIT TO ALL BIT POSITIONS
1677                                    ;               OF DEVICE REGISTERS
1678
1679  001612  032110                              EMT7
1680  001614  037664                              EHT7
1681  001616  037764                              EDT5
1682  001620  040020                              EFT5
```

```
1683
1684
1685                                    ;ERROR  10      REGISTER SELECT 1 APPEARS S-A-0
1686
1687  001622  032132                            EMT10
1688  001624  000000                            0
1689  001626  000000                            0
1690  001630  000000                            0
1691
1692
1693                                    ;ERROR  11      REGISTER SELECT 1 APPEARS S-A-1
1694
1695  001632  032150                            EMT11
1696  001634  000000                            0
1697  001636  000000                            0
1698  001640  000000                            0
1699
1700
1701                                    ;ERROR  12      REGISTER SELECT 2 APPEARS S-A-0
1702
1703  001642  032166                            EMT12
1704  001644  000000                            0
1705  001646  000000                            0
1706  001650  000000                            0
1707
1708
1709                                    ;ERROR  13      REGISTER SELECT 2 APPEARS S-A-1
1710
1711  001652  032204                            EMT13
1712  001654  000000                            0
1713  001656  000000                            0
1714  001660  000000                            0
1715
1716
1717                                    ;ERROR  14      REGISTER SELECT 4 APPEARS S-A-0
1718
1719  001662  032222                            EMT14
1720  001664  000000                            0
1721  001666  000000                            0
1722  001670  000000                            0
1723
1724
1725                                    ;ERROR  15      REGISTER SELECT 4 APPEARS S-A-1
1726
1727  001672  032240                            EMT15
1728  001674  000000                            0
1729  001676  000000                            0
1730  001700  000000                            0
1731
1732
1733                                    ;ERROR  16      REGISTER SELECT 8 APPEARS S-A-0
1734
1735  001702  032256                            EMT16
1736  001704  000000                            0
1737  001706  000000                            0
1738  001710  000000                            0
```

```
1739
1740
1741                                    ;ERROR  17      REGISTER SELECT 8 APPEARS S-A-1
1742
1743  001712  032274                            EMT17
1744  001714  000000                            0
1745  001716  000000                            0
1746  001720  000000                            0
1747
1748
1749                                    ;ERROR  20      OPI SET WITH SEACH TIMOUT DISABLED
1750
1751  001722  032312                            EMT20
1752  001724  037650                            EHT1
1753  001726  037760                            EDT1
1754  001730  040014                            EFT1
1755
1756
1757                                    ;ERROR  21      OPI SET WITH SECTOR PULSE,SECTOR COMPARE WAS RESET
1758
1759  001732  032334                            EMT21
1760  001734  037650                            EHT1
1761  001736  037760                            EDT1
1762  001740  040014                            EFT1
1763
1764
1765                                    ;ERROR  22      DTE SET WITH INDEX PULSE,SEARCH WAS NOT ENABLED
1766
1767  001742  032356                            EMT22
1768  001744  037650                            EHT1
1769  001746  037760                            EDT1
1770  001750  040014                            EFT1
1771
1772
1773                                    ;ERROR  23      DTE SET WITH SECTOR PULSE,SECTOR COMPARE WAS RESET
1774
1775  001752  032402                            EMT23
1776  001754  037650                            EHT1
1777  001756  037760                            EDT1
1778  001760  040014                            EFT1
1779
1780
1781                                    ;ERROR  24      DTE DID NOT SET WITH SECTOR PULSE,SECTOR COMPARE WAS SET
1782
1783  001762  032426                            EMT24
1784  001764  037650                            EHT1
1785  001766  037760                            EDT1
1786  001770  040014                            EFT1
1787
1788
1789                                    ;ERROR  25      DTE SET WITH SECTOR PULSE DURING FORMAT CHANGE
1790
1791  001772  032452                            EMT25
1792  001774  037650                            EHT1
1793  001776  037760                            EDT1
1794  002000  040014                            EFT1
```

```
1795
1796
1797                                    ;ERROR  26      MBA CLR L IS STUCK ACTIVE
1798
1799  002002  032472                            EMT26
1800  002004  000000                            0
1801  002006  000000                            0
1802  002010  000000                            0
1803
1804
1805                                    ;ERROR  27      COULD NOT SET DTE AFTER FORMAT CHANGE
1806
1807  002012  032524                            EMT27
1808  002014  037650                            EHT1
1809  002016  037760                            EDT1
1810  002020  040014                            EFT1
1811
1812
1813                                    ;ERROR  30      CANNOT SET PROM STROBE WITH BIT CLOCK
1814
1815  002022  032544                            EMT30
1816  002024  037650                            EHT1
1817  002026  037760                            EDT1
1818  002030  040014                            EFT1
1819
1820
1821                                    ;ERROR  31      CANNOT CLEAR RMER1,DTE
1822
1823  002032  032564                            EMT31
1824  002034  037650                            EHT1
1825  002036  037760                            EDT1
1826  002040  040014                            EFT1
1827
1828
1829                                    ;ERROR  32      PROM STROBE RESET EARLY
1830
1831  002042  032600                            EMT32
1832  002044  037650                            EHT1
1833  002046  037760                            EDT1
1834  002050  040014                            EFT1
1835
1836
1837                                    ;ERROR  33      PROM STROBE SET EARLY
1838
1839  002052  032612                            EMT33
1840  002054  037650                            EHT1
1841  002056  037760                            EDT1
1842  002060  040014                            EFT1
1843
1844
1845                                    ;ERROR  34      LOOKING FOR SYNC SET EARLY
1846
1847  002062  032624                            EMT34
1848  002064  037650                            EHT1
1849  002066  037760                            EDT1
1850  002070  040014                            EFT1
```

```
1851
1852
1853                                    ;ERROR  35      LOOKING FOR SYNC DID NOT SET
1854
1855  002072  032636                            EMT35
1856  002074  037650                            EHT1
1857  002076  037760                            EDT1
1858  002100  040014                            EFT1
1859
1860
1861                                    ;ERROR  36      PROM STROBE  SET WHILE LOOKING FOR SYNC
1862
1863  002102  032650                            EMT36
1864  002104  037650                            EHT1
1865  002106  037760                            EDT1
1866  002110  040014                            EFT1
1867
1868
1869                                    ;ERROR  37      SYNC DETECTED WITH WRONG PATTERN
1870
1871  002112  032670                            EMT37
1872  002114  037724                            EHT115
1873  002116  040004                            EDT115
1874  002120  040032                            EFT115
1875
1876
1877                                    ;ERROR  40      SYNC NOT DETECTED
1878
1879  002122  032720                            EMT40
1880  002124  037650                            EHT1
1881  002126  037760                            EDT1
1882  002130  040014                            EFT1
1883
1884
1885                                    ;ERROR  41      DRIVE TIMING ERROR DID NOT CLEAR LOOKING FOR SYNC
1886
1887  002132  032742                            EMT41
1888  002134  037650                            EHT1
1889  002136  037760                            EDT1
1890  002140  040014                            EFT1
1891
1892
1893                                    ;ERROR  42      WRITE GATE DID NOT COME ON OR RESET EARLY
1894
1895  002142  032770                            EMT42
1896  002144  037650                            EHT1
1897  002146  037760                            EDT1
1898  002150  040014                            EFT1
1899
1900
1901                                    ;ERROR  43      INCORRECT SYNC PATTERN DURING HEADER
1902
1903  002152  033006                            EMT43
1904  002154  000000                            0
1905  002156  000000                            0
1906  002160  000000                            0
```

```
1907
1908
1909                                    ;ERROR  44      INCORRECT SYNC PATTERN DURING HEADER
1910
1911  002162  033006                              EMT43
1912  002164  037664                              EHT7
1913  002166  037764                              EDT5
1914  002170  040020                              EFT5
1915
1916
1917                                    ;ERROR  45      HEADER AREA DID NOT COME ON OR RESET EARLY
1918
1919  002172  033040                              EMT45
1920  002174  037650                              EHT1
1921  002176  037760                              EDT1
1922  002200  040014                              EFT1
1923
1924
1925                                    ;ERROR  46      CRC ENABLE DID NOT SET
1926
1927  002202  033054                              EMT46
1928  002204  037650                              EHT1
1929  002206  037760                              EDT1
1930  002210  040014                              EFT1
1931
1932
1933                                    ;ERROR  47      INCORRECT HEADER GENERATED DURING WRITE
1934
1935  002212  033070                              EMT47
1936  002214  037670                              EHT47
1937  002216  037766                              EDT47
1938  002220  040014                              EFT1
1939
1940
1941                                    ;ERROR  50      READ GATE INCORRECT DURING HEADER AREA
1942
1943  002222  033106                              EMT50
1944  002224  037650                              EHT1
1945  002226  037760                              EDT1
1946  002230  040014                              EFT1
1947
1948
1949                                    ;ERROR  51      UNEXPECTED HEADER ERROR DURING DIAGNOSTIC MODE
1950
1951  002232  033210                              EMT55
1952  002234  037650                              EHT1
1953  002236  037760                              EDT1
1954  002240  040014                              EFT1
1955
1956
1957                                    ;ERROR  52      INCORRECT HEADER READ IN DIAGNOSTIC MODE
1958
1959  002242  033140                              EMT52
1960  002244  037674                              EHT52
1961  002246  037770                              EDT52
1962  002250  040022                              EFT57
```

```
1963
1964
1965                                    ;ERROR  53      INCORRECT TAG BUS DURING DATA COMMAND
1966
1967  002252  037544                              EMT276
1968  002254  037650                              EHT1
1969  002256  037760                              EDT1
1970  002260  040014                              EFT1
1971
1972
1973                                    ;ERROR  54      DATA TIMING SEQUENCER CONTROLS INCORRECT DURING DATA COMMAND
1974
1975  002262  033172                              EMT54
1976  002264  037650                              EHT1
1977  002266  037760                              EDT1
1978  002270  040014                              EFT1
1979
1980
1981                                    ;ERROR  55      DATA AREA DID NOT COME ON OR RESET EARLY
1982
1983  002272  033210                              EMT55
1984  002274  037650                              EHT1
1985  002276  037760                              EDT1
1986  002300  040014                              EFT1
1987
1988
1989                                    ;ERROR  56      ECC ENABLE DID NOT SET
1990
1991  002302  033226                              EMT56
1992  002304  037650                              EHT1
1993  002306  037760                              EDT1
1994  002310  040014                              EFT1
1995
1996
1997                                    ;ERROR  57      DEVICE IS NOT AN RM03
1998
1999  002312  033242                              EMT57
2000  002314  037700                              EHT57
2001  002316  037772                              EDT57
2002  002320  040022                              EFT57
2003
2004
2005                                    ;ERROR  60      DEVICE AVAILABLE IS NOT SET
2006
2007  002322  033256                              EMT60
2008  002324  037650                              EHT1
2009  002326  037760                              EDT1
2010  002330  040014                              EFT1
2011
2012
2013                                    ;ERROR  61      INCORRECT ECC PATTERN GENERATED DURING WRITE
2014
2015  002332  033272                              EMT61
2016  002334  037704                              EHT61
2017  002336  037774                              EDT61
2018  002340  040022                              EFT57
```

```
2019
2020
2021                                    ;ERROR  62      CANNOT CLEAR PROM STROBE WITH BIT CLOCK
2022
2023  002342  033310                            EMT62
2024  002344  037650                            EMT1
2025  002346  037760                            EDT1
2026  002350  040014                            EFT1
2027
2028
2029                                    ;ERROR  63      DATA AREA DID NOT RESET
2030
2031  002352  033326                            EMT63
2032  002354  037650                            EHT1
2033  002356  037760                            EDT1
2034  002360  040014                            EFT1
2035
2036
2037                                    ;ERROR  64      UNEXPECTED ECC ERROR IN DIAGNOSTIC MODE
2038
2039  002362  033340                            EMT64
2040  002364  037650                            EHT1
2041  002366  037760                            EDT1
2042  002370  040014                            EFT1
2043
2044
2045                                    ;ERROR  65      INCORRECT DATA TRANSFERRED TO MEMORY
2046
2047  002372  033354                            EMT65
2048  002374  037650                            EHT1
2049  002376  037760                            EDT1
2050  002400  040014                            EFT1
2051
2052                                    ;ERROR  66
2053
2054  002402  033366                            EMT66
2055  002404  000000                            0
2056  002406  000000                            0
2057  002410  000000                            0
2058
2059                                    ;ERROR 67
2060
2061  002412  033402                            EMT67
2062  002414  000000                            0
2063  002416  000000                            0
2064  002420  000000                            0
2065
2066                                    ;ERROR 70
2067
2068  002422  033420                            EMT70
2069  002424  000000                            0
2070  002426  000000                            0
2071  002430  000000                            0
2072
2073                                    ;ERROR 71
2074
```

```
 2075  002432  033440                    EMT71
 2076  002434  000000                    0
 2077  002436  000000                    0
 2078  002440  000000                    0
 2079
 2080                              ;ERROR 72
 2081
 2082  002442  033464                    EMT72
 2083  002444  000000                    0
 2084  002446  000000                    0
 2085  002450  000000                    0
 2086
 2087                              ;ERROR 73
 2088
 2089  002452  033510                    EMT73
 2090  002454  000000                    0
 2091  002456  000000                    0
 2092  002460  000000                    0
 2093
 2094                              ;ERROR 74
 2095
 2096  002462  033530                    EMT74
 2097  002464  000000                    0
 2098  002466  000000                    0
 2099  002470  000000                    0
 2100
 2101                              ;ERROR 75
 2102
 2103  002472  033540                    EMT75
 2104  002474  000000                    0
 2105  002476  000000                    0
 2106  002500  000000                    0
 2107
 2108                              ;ERROR 76
 2109
 2110  002502  033552                    EMT76
 2111  002504  000000                    0
 2112  002506  000000                    0
 2113  002510  000000                    0
 2114
 2115                              ;ERROR 77
 2116
 2117  002512  033566                    EMT77
 2118  002514  000000                    0
 2119  002516  000000                    0
 2120  002520  000000                    0
 2121
 2122
 2123                              ;ERROR  100     CANT SET VOLUME VALID
 2124
 2125  002522  035374                    EMT170
 2126  002524  037744                    EHT150
 2127  002526  040004                    EDT115
 2128  002530  040032                    EFT115
 2129
 2130
```

```
2131                                    ;ERROR  101     RUN AND GO WONT SET
2132
2133  002532  037524                            EMT275
2134  002534  037650                            EHT1
2135  002536  037760                            EDT1
2136  002540  040014                            EFT1
2137
2138
2139                                    ;ERROR  102     SEARCH ENABLE WONT SET
2140
2141  002542  035422                            EMT172
2142  002544  037650                            EHT1
2143  002546  037760                            EDT1
2144  002550  040014                            EFT1
2145
2146
2147                                    ;ERROR  103     OCCUPIED IS INCORRECT FOR FUNCTION CODE
2148
2149  002552  035440                            EMT173
2150  002554  037744                            EHT150
2151  002556  040004                            EDT115
2152  002560  040032                            EFT115
2153
2154
2155                                    ;ERROR  104     READ IN PRESET DIDNT CLEAR RMDA, RMDC OR RMOF
2156
2157  002562  035462                            EMT174
2158  002564  000000                            0
2159  002566  000000                            0
2160  002570  000000                            0
2161
2162
2163                                    ;ERROR  105     READ IN PRESET DIDNT CLEAR RMOF
2164
2165  002572  035510                            EMT175
2166  002574  037650                            EHT1
2167  002576  037760                            EDT1
2168  002600  040014                            EFT1
2169
2170
2171                                    ;ERROR  106     READ IN PRESET DIDNT CLEAR RMDA
2172
2173  002602  035526                            EMT176
2174  002604  037650                            EHT1
2175  002606  037760                            EDT1
2176  002610  040014                            EFT1
2177
2178
2179                                    ;ERROR  107     READ IN PRESET DIDNT CLEAR RMDC
2180
2181  002612  035544                            EMT177
2182  002614  037650                            EHT1
2183  002616  037760                            EDT1
2184  002620  040014                            EFT1
2185
2186
```

```
2187                                    ;ERROR  110     CANT SET OFFSET MODE BY OFFSET COMMAND
2188
2189  002622  035562                            EMT200
2190  002624  037650                            EHT1
2191  002626  037760                            EDT1
2192  002630  040014                            EFT1
2193
2194
2195
2196
2197                                    ;PUT ERROR TABLE HERE
2198
```

```
                                        .SBTTL  START OF PROGRAM


002632                                  START:

                                        ;CLEAR AND SETUP FOR TEST EXECUTION
002632  000240                                NOP
002634  000005                                RESET                   ;INITIALIZE THE SYSTEM

                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002636  012706  001114                        MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002642  005026                              1$: CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002644  022706  001154                        CMP     #SWR,R6 ;;DONE?
002650  001374                              BNE     1$-6                ;;LOOP BACK IF NO
002652  012706  001100                        MOV     #STACK,SP       ;;SETUP THE STACK POINTER

                                        ;;INITIALIZE A FEW VECTORS
002656  012737  026076  000020                MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002664  012737  000340  000022                MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002672  012737  026460  000030                MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002700  012737  000340  000032                MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002706  012737  030220  000034                MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002714  012737  000340  000036                MOV     #340,@#TRAPVEC+2;LEVEL 7
002722  012737  030326  000024                MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002730  012737  000340  000026                MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002736  013737  021532  021524                MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
002744  005037  001206                        CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002750  005037  001210                        CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002754  112737  000001  001131                MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002762  012737  002762  001122                MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002770  012737  002770  001124                MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002776  013746  000004                        MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
003002  012737  003036  000004                MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
003010  012737  177570  001154                MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
003016  012737  177570  001156                MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
003024  022777  177777  176122                CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
003032  001012                                BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                        ;;AND  THE HARDWARE SWR IS NOT - -1
003034  000403                                BR      65$             ;;BRANCH IF NO TIMEOUT
003036  012716  003044          64$:          MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
003042  000002                                RTI
003044  012737  000176  001154  65$:          MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
003052  012737  000174  001156                MOV     #DISPREG,DISPLAY
003060  012637  000004          66$:          MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

003064  005037  001230                        CLR     $PASS           ;;CLEAR PASS COUNT
003070  132737  000200  001243                BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
003076  001403                                BEQ     67$             ;;YES,USE NON-APT SWITCH
003100  012737  001244  001154                MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
003106                          67$:
003106  012746  000140                        MOV     #PR3,-(SP)      ;;PUT NEW PS ON STACK
003112  012746  003120                        MOV     #68$,-(SP)      ;;PUT NEW PC ON  STACK
003116  000002                                RTI                     ;;POP NEW PC AND PS
003120                          68$:
```

```
2255                                        .SBTTL  TYPE PROGRAM NAME
2256                                        ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
2257  003120  005227  177777                        INC     #-1             ;;FIRST TIME?
2258  003124  001061                              BNE     69$             ;;BRANCH IF NO
2259  003126  022737  021666  000042              CMP     #$ENDAD,@#42    ;;ACT-11?
2260  003134  001455                              BEQ     69$             ;;BRANCH IF YES
2261  003136  104401  003204                      TYPE    ,70$            ;;TYPE ASCIZ STRING
2262                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2263  003142  005737  000042                      TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2264  003146  001012                              BNE     71$             ;;BRANCH IF YES
2265  003150  123727  001242  000001              CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
2266  003156  001406                              BEQ     71$             ;;BRANCH IF YES
2267  003160  023727  001154  000176              CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
2268  003166  001005                              BNE     72$             ;;BRANCH IF NO
2269  003170  104407                              GTSWR                   ;;GET SOFT-SWR SETTINGS
2270  003172  000403                              BR      72$
2271  003174  112737  000001  001150  71$:       MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
2272  003202                                    72$:
2273  003202  000432                              BR      69$             ;;GET OVER THE ASCIZ
2274                                        ;;70$:  .ASCIZ  <CRLF>@CZRMKB0, RM03/RM02 DISKLESS DIAGNOSTIC  PART II @<CRLF>
2275  003270                                    69$:
2276
2277
2278                                        ;FIND OUT IF PROGRAM IS RUNNING IN STANDALONE MODE
2279  003270  005737  000042                      TST     42              ;IS LOC 42 ZERO ??
2280  003274  001003                              BNE     10$             ;NO - NOT IN STANDALONE
2281  003276  105737  001242                      TSTB    $ENV            ;IS APT ENVIRONMENT ZERO ??
2282  003302  001451                              BEQ     STANDALONE      ;YES - PROGRAM IN STANDALONE
2283  003304                                    10$:
2284
2285                                        ;PROGRAM NOT RUNNING IN STANDALONE - SEE IF SIZING IS ALLOWED
2286  003304  132737  000200  001243  XSIZ:   BITB    #BIT7,$ENVM     ;SIZING ALLOWED ??
2287  003312  001043                              BNE     20$             ;NO
2288                                        ;       MOV     #377,$DEVM      ;YES - SET DEVICE MAP FOR ALL DEVICES
2289  003314  005037  001300                      CLR     $DEVM           ;CLEAR THE BIT MAP
2290  003320  005001                              CLR     R1              ;START FROM THE DRIVE 0
2291  003322  012704  000001                      MOV     #BIT0,R4        ;BIT MAP DOR DRIVE 0
2292  003326  013700  001276                      MOV     $BASE,R0                ;BASE ADDRESS
2293  003332  012760  000040  000010  15$:       MOV     #CLR,RMCS2(R0)  ;LOAD THE DRIVE ADDRESS
2294  003340  010160  000010                      MOV     R1,RMCS2(R0)    ;LOAD THE DRIVE ADDRESS
2295  003344  016003  000010                      MOV     RMCS2(R0),R3    ;NED BIT SET ?
2296  003350  032703  010000                      BIT     #NED,R3         ;BRANCH IF SO
2297  003354  001010                              BNE     16$             ;TO NEXT DRIVE
2298  003356  016003  000000                      MOV     RMCS1(R0),R3    ;CHECK THE DVA BIT
2299  003362  032703  004000                      BIT     #DVA,R3         ;DRIVE AVAILABLE ?
2300  003366  001403                              BEQ     16$             ;BRANCH IF DRIVE NOT AVAILABLE
2301  003370  050437  001300                      BIS     R4,$DEVM        ;SET THE BIT MAP FOR ALL AVAILABLE DRIVE
2302  003374  000405                              BR      17$             ;DONT TYPE THE NONE EXIST MESG
2303  003376                                    16$:
2304  003376  104401  031607                      TYPE    ,NOTEX          ;TYPE NOT EXIT MESSAGE
2305  003402  010146                              MOV     R1,-(SP)        ;DRIVE NUMBER
2306  003404  104403                              TYPOS
2307  003406     006                              .BYTE   6
2308  003407     000                              .BYTE   0
2309  003410  005201                            17$:  INC     R1              ;INCREMENT THE DRIVE ADDRESS
2310  003412  006304                              ASL     R4              ;SET UP THE BIT MAP FOR NEXT DRIVE
```

```
 2311  003414  022701  000007                  CMP     #7,R1           ;ALL DRIVE ARE CHECKED ?
 2312  003420  103344                          BHIS    15$             ;BRANCH IF NOT
 2313  003422                          20$:
 2314
 2315                                  ;GO TO COMMON START CODE
 2316  003422  000137  004254                  JMP     CMNSTART
```

```
 2317  003426                                  STANDALONE:
 2318
 2319  003426  004737  026670                          JSR     PC,$TKINT
 2320                                          ;SEE IF THIS IS THE FIRST START AFTER PROGRAM WAS LOADED
 2321  003432  005327  000001                          DEC     #1              ;FIRST START ??
 2322  003436  100024                                  BPL     10$             ;YES !!
 2323
 2324
 2325                                          ;SEE IF THE USER WANTS TO KEEP SAME DEVICES FOR TESTING
 2326  003440  104401  031116                  5$:     TYPE    ,CNSL00         ;MAINTAIN PREVIOUS PARAMETERS??
 2327  003444  104411                                  RDCHR                   ;GET RESPONSE
 2328  003446  012637  001176                          MOV     (SP)+,$TMP1     ;ECHO RESPONSE
 2329  003452  104401  001176                          TYPE    ,$TMP1
 2330  003456  123727  001176  000131                  CMPB    $TMP1,#'Y       ;YES RESPONSE??
 2331  003464  001002                                  BNE     6$              ;NO!!
 2332  003466  000137  004414                          JMP     READY           ;KEEP PREVIOUS PARAMETERS
 2333  003472  123727  001176  000116  6$:     CMPB    $TMP1,#'N       ;NO RESPONSE??
 2334  003500  001420                                  BEQ     20$             ;GET NEW PARAMETERS
 2335  003502  104401  030761                          TYPE    ,QSTMRK         ;NOT YES OR NO, TYPE ''?''
 2336  003506  000754                                  BR      5$              ;RETRY
 2337  003510                                  10$:
 2338
 2339                                          ;SEE IF OPERATOR WANTS HELP FILE
 2340  003510  104401  030763                          TYPE    ,HELPQST        ;WANT HELP ??
 2341  003514  104411                                  RDCHR                   ;GET RESPONSE
 2342  003516  012637  001176                          MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
 2343  003522  104401  001176                          TYPE    ,$TMP1
 2344  003526  123727  001176  000131                  CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ??
 2345  003534  001002                                  BNE     20$             ;NO - DONT TYPE HELP
 2346  003536  104401  054116                          TYPE    ,HELP           ;YES - TYPE HELP TEXT
 2347  003542                                  20$:
 2348
 2349                                          ;SEE IF USER WANTS TO CHANGE RM03 UNIBUS ADDRESS
 2350  003542  104401  031017                          TYPE    ,UBUSQST        ;WANT TO CHANGE ADDRESS ??
 2351  003546  104411                                  RDCHR                   ;GET RESPONSE
 2352  003550  012637  001176                          MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
 2353  003554  104401  001176                          TYPE    ,$TMP1
 2354  003560  104411                                  RDCHR                   ;READ S CR
 2355  003562  005726                                  TST     (SP)+           ;CLEAR THE STACK
 2356  003564  123727  001176  000131                  CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ??
 2357  003572  001137                                  BNE     90$             ;NO !!
 2358  003574                                  30$:
 2359
 2360                                          ;DIALOGUE TO CHANGE THE UNIBUS ADDRESS, AND INTERRUPT VECTOR
 2361  003574  104401  031155                          TYPE    ,CNSL01         ;TYPE CURRENT BUS ADDRESS
 2362  003600  013746  001276                          MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
 2363  003604  104402                                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2364  003606  104401  030752                          TYPE    ,CLSPRN
 2365  003612  104413                                  RDOCT                   ;GET NEW BUS ADDRESS
 2366  003614  012637  001176                          MOV     (SP)+,$TMP1     ;CARRIAGE RETURN??
 2367  003620  001412                                  BEQ     50$             ;YES-SKIP TO NEXT ENTRY
 2368  003622  022737  160000  001176                  CMP     #160000,$TMP1   ;BASE ADDRESS IN I/O PAGE??
 2369  003630  101403                                  BLOS    40$             ;YES
 2370  003632  104401  031202                          TYPE    ,CNSL02         ;TYPE WARNING MESSAGE
 2371  003636  000756                                  BR      30$             ;RETRY
 2372  003640  013737  001176  001276  40$:     MOV     $TMP1,$BASE     ;STORE NEW BUS ADDRESS
```

```
2373  003646  113737  001272  001176  50$:    MOVB    $VECT1,$TMP1    ;TYPE CURRENT VECTOR ADDRESS
2374  003654  105037  001177                  CLRB    $TMP1+1
2375  003660  104401  031263                  TYPE    ,CNSL03
2376  003664  013746  001176                  MOV     $TMP1,-(SP)     ;;SAVE $TMP1 FOR TYPEOUT
2377  003670  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
2378  003672     003                          .BYTE   3               ;;TYPE 3 DIGIT(S)
2379  003673     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
2380  003674  104401  030752                  TYPE    ,CLSPRN
2381  003700  104413                          RDOCT                   ;GET NEW VECTOR ADDRESS
2382  003702  012637  001176                  MOV     (SP)+,$TMP1     ;CARRIAGE RETURN?
2383  003706  001412                          BEQ     70$             ;YES-SKIP TO NEXT ENTRY
2384  003710  022737  001000  001176          CMP     #1000,$TMP1     ;VECTOR ADDRESS < 1000??
2385  003716  101003                          BHI     60$             ;YES!!
2386  003720  104401  031313                  TYPE    ,CNSL04         ;TYPE WARNING MESSAGE
2387  003724  000750                          BR      50$             ;RETRY
2388  003726  113737  001176  001272  60$:    MOVB    $TMP1,$VECT1    ;STORE NEW VECTOR ADDRESS
2389  003734  113737  001273  001176  70$:    MOVB    $VECT1+1,$TMP1  ;TYPE CURRENT PRIORITY
2390  003742  006237  001176                  ASR     $TMP1
2391  003746  006237  001176                  ASR     $TMP1
2392  003752  006237  001176                  ASR     $TMP1
2393  003756  006237  001176                  ASR     $TMP1
2394  003762  006237  001176                  ASR     $TMP1
2395  003766  105037  001177                  CLRB    $TMP1+1
2396  003772  104401  031367                  TYPE    ,CNSL05
2397  003776  013746  001176                  MOV     $TMP1,-(SP)     ;;SAVE $TMP1 FOR TYPEOUT
2398  004002  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
2399  004004     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
2400  004005     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
2401  004006  104401  030752                  TYPE    ,CLSPRN
2402  004012  104413                          RDOCT                   ;GET NEW PRIORITY
2403  004014  012637  001176                  MOV     (SP)+,$TMP1     ;CARRIAGE RETURN??
2404  004020  001424                          BEQ     90$             ;YES-SKIP TO NEXT ENTRY
2405  004022  023727  001176  000007          CMP     $TMP1,#7        ;LEGAL PRIORITY??
2406  004030  002403                          BLT     80$             ;YES!!
2407  004032  104401  031423                  TYPE    ,CNSL06         ;TYPE WARNING MESSAGE
2408  004036  000736                          BR      70$             ;RETRY
2409  004040                          80$:                            ;STORE NEW PRIORITY
2410  004040  006337  001176                  ASL     $TMP1
2411  004044  006337  001176                  ASL     $TMP1
2412  004050  006337  001176                  ASL     $TMP1
2413  004054  006337  001176                  ASL     $TMP1
2414  004060  006337  001176                  ASL     $TMP1
2415  004064  113737  001176  001273          MOVB    $TMP1,$VECT1+1
2416  004072                          90$:
2417
2418                                  ;DIALOGUE TO INPUT DEVICE NUMBERS
2419  004072  005037  001300                  CLR     $DEVM           ;CLEAR DEVICE MAP
2420  004076  104401  031450                  TYPE    ,CNSL07         ;TYPE INPUT INSTRUCTIONS
2421  004102  104401  030755                  TYPE    ,PROMPT         ;TYPE PROMPTING CHARACTER
2422  004106  104411                          RDCHR                   ;GET RESPONSE
2423  004110  012637  001176                  MOV     (SP)+,$TMP1     ;ECHO RESPONSE
2424  004114  104401  001176                  TYPE    ,$TMP1
2425  004120  023727  001176  000101          CMP     $TMP1,#'A       ;TEST ALL DRIVES??
2426  004126  001021                          BNE     110$            ;NO
2427  004130  000137  003304                  JMP     XSIZ            ;TO SIZE ALL POWER UP DRIVES
2428  004134  012737  000377  001300          MOV     #377,$DEVM      ;TEST ALL DEVICES
```

```
 2429  004142  000444                          BR      140$            ;SKIP TO NEXT ENTRY
 2430  004144  104401  030755          100$:   TYPE    ,PROMP          ;TYPE PROMPTING CHARACTER
 2431  004150  104411                          RDCHR                   ;GET RESPONSE
 2432  004152  012637  001176                  MOV     (SP)+,$TMP1     ;ECHO RESPONSE
 2433  004156  104401  001176                  TYPE    ,$TMP1
 2434  004162  023727  001176  000015          CMP     $TMP1,#CR       ;CARRIAGE RETURN??
 2435  004170  001431                          BEQ     140$
 2436  004172  023727  001176  000060  110$:   CMP     $TMP1,#'0       ;NUMBER < 0??
 2437  004200  002404                          BLT     120$            ;YES.!
 2438  004202  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7??
 2439  004210  003403                          BLE     130$            ;NO!!
 2440  004212  104401  030761          120$:   TYPE    ,QSTMRK         ;TYPE "?"
 2441  004216  000752                          BR      100$            ;RETRY
 2442  004220  013701  001176          130$:   MOV     $TMP1,R1        ;R1=DRIVE NUMBER
 2443  004224  042701  177770                  BIC     #^C7,R1
 2444  004230  116102  031734                  MOVB    ATNTBL(R1),R2           ;DECODE DEVICE NUMBER
 2445  004234  042702  177400                  BIC     #^C377,R2               ;CLEAR UNUSED BITS
 2446  004240  050237  001300                  BIS     R2,$DEVM                ;SET DEVICE # IN MAP
 2447  004244  122737  000377  001300          CMPB    #377,$DEVM      ;DONE ??
 2448  004252  101334                          BHI     100$            ;NO
 2449  004254                          140$:
 2450
```

```
2451  004254                                CMNSTART:
2452
2453                                        ;ASSEMBLE TEST QUE FROM DEVICE MAP
2454  004254  013700  001300                      MOV     $DEVM,R0          ;R0 = DEVICE MAP
2455  004260  012701  001460                      MOV     #TSTQUE+2,R1      ;R1 = ADDRESS OF FIRST ENTRY IN QUE
2456  004264  010137  001456                      MOV     R1,TSTQUE         ;INITIALIZE ENTRY POINTER
2457  004270  012702  000001                      MOV     #1,R2             ;R2 = DEVICE POINTER
2458  004274  005003                              CLR     R3                ;R3 = DEVICE NUMBER
2459  004276  030200                10$:    BIT     R2,R0             ;IS THIS DEVICE IN MAP ??
2460  004300  001406                              BEQ     20$               ;NO !!
2461  004302  010311                              MOV     R3,(R1)           ;YES - ENTER DEVICE NUMBER IN QUE
2462  004304  116361  031734  000001              MOVB    ATNTBL(R3),1(R1);ENTER ATTENTION BIT IN QUE
2463  004312  062701  000002                      ADD     #2,R1             ;ADVANCE ENTRY POINTER
2464  004316  006302                20$:    ASL     R2                ;ADVANCE DEVICE POINTER
2465  004320  105702                              TSTB    R2                ;DONE ALL DEVICES ??
2466  004322  001402                              BEQ     25$               ;YES
2467  004324  005203                              INC     R3                ;ADVANCE DEVICE NUMBER
2468  004326  000763                              BR      10$               ;ENTER NEXT DEVICE
2469  004330  005011                25$:    CLR     (R1)              ;TERMINATE TEST QUE
2470
2471                                        ;SIZE FOR CLOCK
2472  004332  004737  022510                      JSR     PC,SIZCLK         ;SEE IF CLOCK PRESENT
2473  004336  000403                              BR      40$               ;YES - CLOCK IS PRESENT
2474  004340  104000                30$:    ERROR                     ;NO CLOCK
2475  004342  000000                              HALT
2476  004344  000775                              BR      30$
2477  004346                        40$:
2478                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2479  004346  005737  000042                      TST     @#42              ;;ARE WE RUNNING UNDER XXDP/ACT?
2480  004352  001012                              BNE     64$               ;;BRANCH IF YES
2481  004354  123727  001242  000001              CMPB    $ENV,#1           ;;ARE WE RUNNING UNDER APT?
2482  004362  001406                              BEQ     64$               ;;BRANCH IF YES
2483  004364  023727  001154  000176              CMP     SWR,#SWREG        ;;SOFTWARE SWITCH REG SELECTED?
2484  004372  001005                              BNE     65$               ;;BRANCH IF NO
2485  004374  104407                              GTSWR                     ;;GET SOFT-SWR SETTINGS
2486  004376  000403                              BR      65$
2487  004400  112737  000001  001150  64$:     MOVB    #1,$AUTOB         ;;SET AUTO-MODE INDICATOR
2488  004406                        65$:
2489  004406  012737  000140  000032              MOV     #PR3,@#EMTVEC+2 ;DROP PRIORITY DURING ERROR TYPEOUT
2490  004414  000240                READY:  NOP                       ;READY TO START TEST
2491  004416  004737  026670                      JSR     PC,$TKINT         ;INITIALIZE TTY
2492  004422  117737  175030  001234              MOVB    @TSTQUE,$UNIT     ;LOAD UNIT NUMBER
```

```
                                .SBTTL  REGISTER AND STORAGE USAGE

                                ;REGISTER ASSIGNMENTS
```

```
2496                                    ;
2497                                    ;R0     =       UNIBUS ADDRESS OF RH CONTROLLER
2498                                    ;R1     =       ADDRESS OF ENTRY IN TEST QUE CORRESPONDING TO THE
2499                                    ;               UNIT UNDER TEST
2500                                    ;R2,R3          WORKING REGISTERS FOR TEST IN PROGRESS, MUST BE
2501                                    ;               SAVED BY SUBROUTINES
2502                                    ;R4,R5  =       GENERAL WORKING REGISTERS, ARE NOT SAVED BY
2503                                    ;               SUBROUTINES
2504                                    ;R6     =       STACK POINTER
2505                                    ;R7     =       LINKAGE REGISTER TO SUBROUTINES
2506
2507                                    ;STORAGE ASSIGNMENTS
2508                                    ;
2509                                    ;$TMP0-$TMP4    TEMPORARY STORAGE, NOT SAVED BY SUBROUTINES
2510                                    ;$GDDAT,$BDDAT  EXPECTED AND RRECEIVED STATUS FOR ERROR TYPEOUT
2511                                    ;$GDADR,$BDADR  ADDRESS OF EXPECTED AND RECEIVED STATUS IF APPLICABLE,
2512                                    ;               ALSO THE ADDRESS OF A REGISTER ERROR
2513                                    ;
2514                                    ;$TSTN          TEST NUMBER
2515                                    ;$UNIT  -       NUMBER OF DEVICE BEING TESTED
2516                                    ;RGINBF =       THE REGISTER INPUT BUFFER HAS A STORAGE LOCATION FOR
2517                                    ;               EACH REGISTER, AND IS USED WHEN READING STATUS AND
2518                                    ;               CONTROL DATA
2519                                    ;RGOTBF =       THE REGISTER OUTPUT BUFFER HAS A STORAGE LOCATION FOR
2520                                    ;               EACH REGISTER, AND IS USED FOR ASSEMBLING DATA TO BE
2521                                    ;               WRITTEN IN REGISTERS
```

```
2522                                    ;;*****************************************************************
2523                                    ;*TEST 1         TRANSFER TEST
2524
2525                                    ;;*****************************************************************
2526  004430  000004                    TST1:   SCOPE
2527  004432  012737  000001  001226            MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
2528
2529  004440  000240                            NOP
2530  004442  012737  000024  001120            MOV     #20.,$ICNT      ;20 ITERATIONS
2531  004450  112737  000001  001131            MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
2532  004456  012737  004472  001122            MOV     #T1,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2533  004464  012737  004472  001124            MOV     #T1,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2534  004472                            T1:
2535  004472  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
2536  004476  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
2537  004502  013701  001456                    MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
2538  004506  012702  000000                    MOV     #0,R2           ;R2 - REGISTER INDEX
2539  004512                            10$:
2540                                    ;CLEAR THE MASSBUS AND VERIFY THAT NONEXISTANT DEVICE ERROR IS RESET
2541  004512  012760  000040  000010            MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2542  004520  111160  000010                    MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2543
2544  004524  016037  000010  001142            MOV     RMCS2(R0),$BDDAT        ;STORE RMCS2 AT $BDDAT
2545  004532  032737  010000  001142            BIT     #NED,$BDDAT
2546  004540  001417                            BEQ     20$
2547  004542  111137  001140                    MOVB    (R1),$GDDAT
2548  004546  042737  177770  001140            BIC     #^CUNTMSK,$GDDAT
2549  004554  052737  000100  001140            BIS     #IR,$GDDAT
2550  004562  010037  001136                    MOV     R0,$BDADR
2551  004566  062737  000010  001136            ADD     #RMCS2,$BDADR
2552  004574  104001                            ERROR   1
2553  004576  000500                            BR      60$
2554  004600                            20$:
2555                                    ;READ THE REGISTER WHOSE INDEX IS IN R2 AND EXIT TEST IF THE READ
2556                                    ;DOES NOT SET 'NED' ERROR
2557  004600  010003                            MOV     R0,R3           ;R3 - REGISTER ADDRESS
2558  004602  060203                            ADD     R2,R3
2559  004604  011304                            MOV     (R3), R4                ;READ REGISTER
2560  004606  032760  010000  000010            BIT     #NED, RMCS2(R0)         ;IS 'NED" SET??
2561  004614  001473                            BEQ     70$             ;NO.!
2562  004616  012760  000040  000010            MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2563  004624  111160  000010                    MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2564
2565  004630  016037  000010  001142            MOV     RMCS2(R0),$BDDAT        ;STORE RMCS2 AT $BDDAT
2566  004636  032737  010000  001142            BIT     #NED,$BDDAT
2567  004644  001417                            BEQ     30$
2568  004646  111137  001140                    MOVB    (R1),$GDDAT
2569  004652  042737  177770  001140            BIC     #^CUNTMSK,$GDDAT
2570  004660  052737  000100  001140            BIS     #IR,$GDDAT
2571  004666  010037  001136                    MOV     R0,$BDADR
2572  004672  062737  000010  001136            ADD     #RMCS2,$BDADR
2573  004700  104001                            ERROR   1
2574  004702  000436                            BR      60$
2575  004704                            30$:
2576                                    ;WRITE THE REGISTER WHOSE INDEX IS IN R2 AND EXIT TEST IF THE WRITE
2577                                    ;DOES NOT SET 'NED" ERROR
```

```
2578  004704  012713  000000          MOV     #0,(R3)          ;WRITE REGISTER
2579  004710  032760  010000  000010  BIT     #NED, RMCS2      (R0)    ;IS 'NED' SET??
2580  004716  001432                  BEQ     70$              ;NO!!
2581
2582  004720                      40$:
2583                              ;COULD NOT READ OR WRITE THE REGISTER WITHOUT SETING 'NED' ERROR -
2584                              ;ADVANCE THE REGISTER INDEX AND REPEAT THE TEST FOR THE NEXT
2585                              ;AVAILABLE DEVICE REGISTER
2586  004720  062702  000002          ADD     #2,R2            ;ADVANCE TO NEXT REGISTER
2587  004724  022702  000002          CMP     #RMWC,R2                 ;IS THIS RMWC??
2588  004730  001773                  BEQ     40$              ;YES - TRY NEXT REGISTER
2589  004732  022702  000004          CMP     #RMBA,R2                 ;IS THIS RMBA??
2590  004736  001770                  BEQ     40$              ;YES - TRY NEXT REGISTER
2591  004740  022702  000010          CMP     #RMCS2,R2                ;IS THIS RMCS5??
2592  004744  001765                  BEQ     40$              ;YES - TRY ANOTHER REGISTER
2593  004746  022702  000016          CMP     #RMAS,R2                 ;IS THIS RMAS ??
2594  004752  001762                  BEQ     40$              ;YES - TRY ANOTHER REGISTER
2595  004754  022702  000022          CMP     #RMDB,R2                 ;IS THIS RMDB??
2596  004760  001757                  BEQ     40$              ;YES - TRY ANOTHER REGISTER
2597  004762  022702  000046          CMP     #RMEC2,R2                ;IS THIS A LEGAL REGISTER
2598  004766  103251                  BHIS    10$              ;YES - TRY THIS REGISTER
2599
2600  004770                      50$:
2601
2602                              ;GOT 'NONEXISTENT DEVICE' ERROR FOR EVERY REMOTE REGISTER ADDRESS
2603  004770  013737  001276  001136      MOV     $BASE, $BDADR            ;STORE BASE ADDRESS
2604  004776  104002                  ERROR   2                ;DEMAND OR TRANSFER FAILED
2605  005000  000137  021442      60$:    JMP     $EOSP            ;GO SELECT NEXT DEVICE
2606
2607  005004                      70$:
2608
2609                              ;;*****************************************************************
2610                              ;*TEST 2        CTOD TEST
2611
2612                              ;;*****************************************************************
2613  005004  000004              TST2:   SCOPE
2614  005006  012737  000002  001226      MOV     #2,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX
2615  005014  000240                  NOP
2616  005016  012737  000024  001120      MOV     #20.,$ICNT       ;20 ITERATIONS
2617  005024  112737  000001  001131      MOVB    #1,$ERMAX        ;ONE ERROR ALLOWED
2618  005032  012737  005046  001122      MOV     #T2,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2619  005040  012737  005046  001124      MOV     #T2,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2620  005046                      T2:
2621  005046  012706  001100          MOV     #STACK,SP        ;LOAD THE STACK POINTER
2622  005052  013700  001276          MOV     $BASE,R0         ;R0 = UNIBUS ADDRESS OF UUT
2623  005056  013701  001456          MOV     TSTQUE,R1        ;R1 = POINTER TO DEVICE
2624  005062  012760  000040  000010      MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2625  005070  111160  000010          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2626                              ;WRITE ONES IN REMOTE REGISTERS
2627
2628  005074  012760  000076  000000      MOV     #ILF76,RMCS1(R0)         ;LOAD RMCS1
2629
2630  005102  012760  177777  000006      MOV     #-1,RMDA(R0)     ;LOAD RMDA
2631
2632  005110  012760  001777  000034      MOV     #CYLMSK,RMDC(R0)         ;LOAD RMDC
2633
```

```
2634  005116  012760  016200  000032          MOV     #FMT16!ECI.HCI!OFD,RMOF(R0)     ;LOAD RMOF
2635                          ;READ REMOTE REGISTERS TWICE
2636  005124  012702  000001                  MOV     #1,R2
2637  005130                  10$:
2638
2639  005130  016037  000000  001326          MOV     RMCS1(R0),RMCS1I        ;STORE RMCS1 IN INPUT BUFFER
2640
2641  005136  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
2642
2643  005144  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
2644
2645  005152  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
2646  005160  005302                          DEC     R2
2647  005162  100362                          BPL     10$
2648                          ;SEE IF ANY ONE BITS CAME BACK
2649  005164  042737  177701  001326          BIC     #^CILF76,RMCS1I ;IS RMCS1 0??
2650  005172  001014                          BNE     20$             ;NO..
2651  005174  005737  001334                  TST     RMDAI           ;IS RMDA 0??
2652  005200  001011                          BNE     20$             ;NO!!
2653  005202  042737  176000  001362          BIC     #XNUDC,RMDCI    ;IS RMDC 0??
2654  005210  001005                          BNE     20$             ;NO!!
2655  005212  042737  161577  001360          BIC     #XNUOF,RMOFI    ;IS RMOF 0 ??
2656  005220  001001                          BNE     20$             ;NO.!
2657                          ;CANNOT READ ANY ONE BITS FROM REMOTE REGISTERS
2658  005222  104003                          ERROR   3               ;CTOD MUST BE STUCK
2659  005224                  20$:
2660
2661                          ;;*****************************************************************
2662                          ;*TEST 3        MASSBUS INITIALIZE TEST
2663
2664                          ;;*****************************************************************
2665  005224  000004          TST3:   SCOPE
2666  005226  012737  000003  001226          MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
2667
2668  005234  000240                          NOP
2669  005236  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
2670  005244  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
2671  005252  012737  005266  001122          MOV     #T3,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2672  005260  012737  005266  001124          MOV     #T3,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2673  005266                  T3:
2674  005266  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
2675  005272  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
2676  005276  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
2677  005302  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2678  005310  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2679                          ;WRITE ONES IN SELECTED REGISTERS
2680
2681  005314  012760  000076  000000          MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1
2682
2683  005322  012760  177777  000014          MOV     #-1,RMER1(R0)   ;LOAD RMER1
2684
2685  005330  012760  177777  000042          MOV     #-1,RMER2(R0)   ;LOAD RMER2
2686                          ;USING CONTROLLER CLEAR, IE., BIT 5 OF RMCS2, INITIALIZE THE MASSBUS
2687  005336  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2688  005344  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2689                          ;READ THE REGISTERS THAT WERE WRITTEN
```

```
005350  016037  000000  001326          MOV     RMCS1(R0),RMCS1I          ;STORE RMCS1 IN INPUT BUFFER
005356  016037  000014  001342          MOV     RMER1(R0),RMER1I          ;STORE RMER1 IN INPUT BUFFER
005364  016037  000042  001370          MOV     RMER2(R0),RMER2I          ;STORE RMER2 IN INPUT BUFFER
                                ;SEE IF ANY REGISTER BITS WERE CLEARED
005372  052737  177701  001326          BIS     #^CILF76,RMCS1I           ;SET ANY BIT NOT WRITTEN
005400  052737  001567  001370          BIS     #XNUER2,RMER2I
005406  022737  177777  001326          CMP     #-1,RMCS1I                ;ANY ZEROS IN RMCS1??
005414  001011                          BNE     10$             ;YES!!
005416  022737  177777  001342          CMP     #-1,RMER1I                ;ANY ZEROS IN RMERI??
005424  001005                          BNE     10$             ;YES!!
005426  022737  177777  001370          CMP     #-1,RMER2I                ;ANY ZEROS IN RMER2??
005434  001001                          BNE     10$
                                ;NONE OF THE BITS WERE CLEARED
005436  104004                          ERROR   4               ;MASSBUS INIT FAILED
005440                          10$:

                                ;;*****************************************************************
                                ;*TEST 4         CLEAR STUCK ACTIVE TEST

                                ;;*****************************************************************
005440  000004                  TST4:   SCOPE
005442  012737  000004  001226          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

005450  000240                          NOP
005452  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
005460  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
005466  012737  005502  001122          MOV     #T4,$LPADR  ;LOAD LOOP ON TEST ADDRESS
005474  012737  005502  001124          MOV     #T4,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
005502                          T4:
005502  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
005506  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
005512  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
005516  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
005524  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                                ;WRITE ONES IN TEST REGISTERS

005530  012760  177777  000014          MOV     #-1,RMER1(R0)   ;LOAD RMER1

005536  012760  177777  000042          MOV     #-1,RMER2(R0)   ;LOAD RMER2

005544  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
                                ;READ TEST REGISTERS AND SEE IF ANY BITS ARE ON

005552  016037  000014  001342          MOV     RMER1(R0),RMER1I          ;STORE RMER1 IN INPUT BUFFER

005560  016037  000042  001370          MOV     RMER2(R0),RMER2I          ;STORE RMER2 IN INPUT BUFFER

005566  016037  000024  001352          MOV     RMMR1(R0),RMMR1I          ;STORE RMMR1 IN INPUT BUFFER
005574  042737  040000  001342          BIC     #UNS,RMER1I     ;DONT ACCEPT UNSAFE
005602  001011                          BNE     10$             ;BRANCH IF ANY OTHER BITS ON
005604  042737  040200  001370          BIC     #SKI!DVC,RMER2I ;DONT ACCEPT SKI OR DVC
005612  001005                          BNE     10$             ;BRANCH IF ANY OTHER BITS ON
005614  032737  000001  001352          BIT     #DMD,RMMR1I     ;BRANCH IF DMD IS ON
```

```
                                T4      CLEAR STUCK ACTIVE TEST

2746  005622  001001                            BNE     10$
2747  005624  104026                            ERROR   26              ;MBA CLR IS STUCK ACTIVE
2748  005626                        10$:
2749
2750
2751                            ;;*****************************************************************
2752                            ;*TEST 5         TRISTATE TRANSFER TEST
2753
2754                            ;;*****************************************************************
2755  005626  000004            TST5:   SCOPE
2756  005630  012737  000005  001226            MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
2757  005636  000240                            NOP
2758  005640  012737  000024  001120            MOV     #20.,$ICNT      ;20 ITERATIONS
2759  005646  112737  000001  001131            MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
2760  005654  012737  005670  001122            MOV     #T5,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2761  005662  012737  005670  001124            MOV     #T5,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2762  005670                        T5:
2763  005670  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
2764  005674  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
2765  005700  013701  001456                    MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
2766  005704  005002                            CLR     R2              ;CLEAR ERROR FLAGS
2767  005706  012760  000040  000010            MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2768  005714  111160  000010                    MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2769                          ;WRITE ONES IN SELECTED REGISTERS
2770
2771  005720  012760  000076  000000            MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1
2772
2773  005726  012760  177777  000006            MOV     #-1,RMDA(R0)    ;LOAD RMDA
2774
2775  005734  012760  177777  000014            MOV     #-1,RMER1(R0)   ;LOAD RMER1
2776
2777  005742  012760  177777  000032            MOV     #-1,RMOF(R0)    ;LOAD RMOF
2778
2779  005750  012760  177777  000042            MOV     #-1,RMER2(R0)   ;LOAD RMER2
2780                          ;WRITE ZEROS IN SELECTED REGISTERS
2781
2782  005756  012760  000000  000000            MOV     #0,RMCS1(R0)    ;LOAD RMCS1
2783
2784  005764  012760  000000  000006            MOV     #0,RMDA(R0)     ;LOAD RMDA
2785
2786  005772  012760  000000  000014            MOV     #0,RMER1(R0)    ;LOAD RMER1
2787
2788  006000  012760  000000  000032            MOV     #0,RMOF(R0)     ;LOAD RMOF
2789
2790  006006  012760  000000  000034            MOV     #0,RMDC(R0)     ;LOAD RMDC
2791
2792  006014  012760  000000  000042            MOV     #0,RMER2(R0)    ;LOAD RMER2
2793                          ;READ BACK ALL REGISTERS
2794
2795  006022  016037  000000  001326            MOV     RMCS1(R0),RMCS1I        ;STORE RMCS1 IN INPUT BUFFER
2796
2797  006030  016037  000006  001334            MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
2798
2799  006036  016037  000014  001342            MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
2800
2801  006044  016037  000032  001360            MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
```

```
2802
2803  006052  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
2804
2805  006060  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
2806                        ;CHECK EACH REGISTER CONTENT FOR ZERO BITS WRITTEN & READ
2807  006066  012702  177777                  MOV     #-1,R2          ;ACCUMULATE ZEROS IN R2
2808  006072  052737  177701  001326          BIS     #^CILF76,RMCS1I ;SET ALL BITS NOT WRITTEN
2809  006100  052737  161577  001360          BIS     #XNUOF,RMOFI
2810  006106  052737  176000  001362          BIS     #XNUDC,RMDCI
2811  006114  052737  001567  001370          BIS     #XNUER2,RMER2I
2812  006122  005137  001326                  COM     RMCS1I          ;COMPLEMENT REGISTER CONTENTS
2813  006126  005137  001334                  COM     RMDAI
2814  006132  005137  001342                  COM     RMER1I
2815  006136  005137  001360                  COM     RMOFI
2816  006142  005137  001362                  COM     RMDCI
2817  006146  005137  001370                  COM     RMER2I
2818  006152  043702  001326                  BIC     RMCS1I,R2               ;ACCUMULATE ALL ZERO BITS
2819  006156  043702  001334                  BIC     RMDAI,R2
2820  006162  043702  001342                  BIC     RMER1I,R2
2821  006166  043702  001360                  BIC     RMOFI,R2
2822  006172  043702  001362                  BIC     RMDCI,R2
2823  006176  043702  001370                  BIC     RMER2I,R2
2824  006202  001407                          BEQ     10$             ;BRANCH IF EACH BIT IS ZERO
2825                        ;ONE OR MORE BIT POSITIONS ARE NOT ZERO
2826  006204  010237  001142                  MOV     R2,$BDDAT               ;SAVE RESULT FOR TYPE
2827  006210  005037  001140                  CLR     $GDDAT          ;LOAD EXPECTED RESULT
2828  006214  104005                          ERROR   5               ;TRISTATE BUS IS STUCK AT ONE
2829  006216  052702  000001                  BIS     #BIT0,R2        ;SET ERROR FLAG
2830  006222                          10$:
2831
2832  006222  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2833  006230  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2834                        ;PRESET SELECTED REGISTERS TO ZEROS
2835                        ;(ASSUME RMCS1, RMER1, RMER2 WERE CLEARED BY INIT)
2836
2837  006234  012760  000000  000006          MOV     #0,RMDA(R0)     ;LOAD RMDA
2838
2839  006242  012760  000000  000032          MOV     #0,RMOF(R0)     ;LOAD RMOF
2840
2841  006250  012760  000000  000034          MOV     #0,RMDC(R0)     ;LOAD RMDC
2842                        ;WRITE ONES IN SELECTED REGISTERS
2843
2844  006256  012760  000076  000000          MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1
2845
2846  006264  012760  177777  000006          MOV     #-1,RMDA(R0)    ;LOAD RMDA
2847
2848  006272  012760  016200  000032          MOV     #^CXNUOF,RMOF(R0)       ;LOAD RMOF
2849
2850  006300  012760  001777  000034          MOV     #^CXNUDC,RMDC(R0)       ;LOAD RMDC
2851
2852  006306  012760  177777  000014          MOV     #-1,RMER1(R0)   ;LOAD RMER1
2853
2854  006314  012760  176210  000042          MOV     #^CXNUER2,RMER2(R0)     ;LOAD RMER2
2855                        ;READ ALL REGISTERS
2856
2857  006322  016037  000000  001326          MOV     RMCS1(R0),RMCS1I        ;STORE RMCS1 IN INPUT BUFFER
```

```
2858
2859  006330  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
2860
2861  006336  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
2862
2863  006344  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
2864
2865  006352  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
2866
2867  006360  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
2868                          ;CHECK EACH REGISTER CONTENT FOR ONE BITS WRITTEN & READ
2869  006366  042737  177701  001326          BIC     #^CILF76,RMCS1I         ;CLEAR ALL BITS NOT WRITTEN
2870  006374  042737  161577  001360          BIC     #XNUOF,RMOFI
2871  006402  042737  176000  001362          BIC     #XNUDC,RMDCI
2872  006410  042737  001567  001370          BIC     #XNUER2,RMER2I
2873  006416  005002                          CLR     R2              ;ACCUMULATE ONES IN R2
2874  006420  053702  001326                  BIS     RMCS1I,R2               ;ACCUMULATE ALL ONE BITS
2875  006424  053702  001334                  BIS     RMDAI,R2
2876  006430  053702  001360                  BIS     RMOFI,R2
2877  006434  053702  001362                  BIS     RMDCI,R2
2878  006440  053702  001342                  BIS     RMER1I,R2
2879  006444  053702  001370                  BIS     RMER2I,R2
2880  006450  022702  177777                  CMP     #-1,R2          ;SEE IF EACH BIT POSITION WAS ONE
2881  006454  001410                          BEQ     20$             ;BRANCH IF NONE STUCK
2882                          ;ONE OR MORE BIT POSITIONS ARE NOT ONE
2883  006456  010237  001142                  MOV     R2,$BDDAT               ;SAVE RESULT FOR TYPE
2884  006462  012737  177777  001140          MOV     #-1,$GDDAT              ;EXPECTED RESULT
2885  006470  104006                          ERROR   6
2886  006472  052702  000002                  BIS     #BIT1,R2        ;SET ERROR FLAG
2887  006476                  20$:
2888
2889  006476  005702                          TST     R2              ;ANY ERRORS DETECTED ??
2890  006500  001131                          BNE     30$             ;YES - DONT DO BIT TEST
2891  006502  012702  000001                  MOV     #1,R2           ;R2=BIT POSITION
2892  006506                  25$:
2893  006506  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2894  006514  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2895                          ;WRITE THE BIT PATTERN IN SELECTED DEVICE REGISTERS
2896
2897  006520  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
2898
2899  006524  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
2900
2901  006530  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
2902
2903  006534  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
2904
2905  006540  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
2906                          ;READ BACK THE REGISTERS
2907
2908  006544  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
2909
2910  006552  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
2911
2912  006560  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
2913
```

```
2914  006566  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
2915
2916  006574  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
2917                          ;CHECK REGISTER CONTENTS FOR CORRECT PATTERN
2918  006602  005003                          CLR     R3              ;R3=ACCUMULATED ONE BIT
2919  006604  012704  177777                  MOV     #-1,R4          ;R4=ACCUMULATED ZERO BITS
2920  006610  013705  001334                  MOV     RMDAI,R5                ;GET ANY GOOD BITS FROM RMDA
2921  006614  050503                          BIS     R5,R3
2922  006616  005105                          COM     R5
2923  006620  040504                          BIC     R5,R4
2924  006622  013705  001360                  MOV     RMOFI,R5                ;GET GOOD BITS FROM RMOF
2925  006626  042705  161577                  BIC     #XNUOF,R5
2926  006632  050503                          BIS     R5,R3
2927  006634  005105                          COM     R5
2928  006636  042705  161577                  BIC     #XNUOF,R5
2929  006642  040504                          BIC     R5,R4
2930  006644  013705  001362                  MOV     RMDCI,R5                ;GET GOOD BITS FROM RMDC
2931  006650  042705  176000                  BIC     #XNUDC,R5
2932  006654  050503                          BIS     R5,R3
2933  006656  005105                          COM     R5
2934  006660  042705  176000                  BIC     #XNUDC,R5
2935  006664  040504                          BIC     R5,R4
2936  006666  013705  001342                  MOV     RMER1I,R5               ;GET GOOD BITS FROM RMER1
2937  006672  050503                          BIS     R5,R3
2938  006674  005105                          COM     R5
2939  006676  040504                          BIC     R5,R4
2940  006700  013705  001370                  MOV     RMER2I,R5               ;GET GOOD BITS FROM RMER2
2941  006704  042705  001567                  BIC     #XNUER2,R5
2942  006710  050503                          BIS     R5,R3
2943  006712  005105                          COM     R5
2944  006714  042705  001567                  BIC     #XNUER2,R5
2945  006720  040504                          BIC     R5,R4
2946  006722  010205                          MOV     R2,R5           ;RESET ALL ONES IN R3 EXCEPT
2947  006724  005105                          COM     R5              ;FOR THE TEST BIT
2948  006726  040503                          BIC     R5,R3
2949  006730  040204                          BIC     R2,R4           ;RESET TEST BIT IN R4
2950  006732  050403                          BIS     R4,R3           ;COMBINE ACCUMULATED 1'S + 0'S
2951  006734  020302                          CMP     R3,R2           ;IS PATTERN OK??
2952  006736  001406                          BEQ     26$             ;YES!!
2953  006740  010237  001140                  MOV     R2,$GDDAT               ;SAVE TEST PATTERN
2954  006744  010337  001142                  MOV     R3,$BDDAT               ;SAVE RESULT
2955  006750  104007                          ERROR   7               ;BIT INTERFERENCE IN TRISTATE BUS
2956  006752  000404                          BR      30$             ;SKIP TO NEXT
2957  006754                  26$:
2958                          ;ADVANCE R2 TO THE NEXT PATTERN AND REPEAT TEST
2959  006754  006302                  ASL     R2              ;SHIFT THE BIT
2960  006756  001402                  BEQ     30$             ;EXIT IF DONE
2961  006760  000137  006506          JMP     25$
2962  006764                  30$:
2963
2964                          ;;*******************************************************************
2965                          ;*TEST 6         REGISTER SELECT TEST
2966
2967                          ;;*******************************************************************
2968  006764  000004          TST6:   SCOPE
2969  006766  012737  000006  001226          MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
```

```
2970
2971  006774  000240                          NOP
2972  006776  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
2973  007004  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
2974  007012  012737  007026  001122          MOV     #T6,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2975  007020  012737  007026  001124          MOV     #T6,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2976  007026                          T6:
2977  007026  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
2978  007032  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
2979  007036  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
2980                                  ;THE FOLLOWING TABLE GIVES MASSBUS REGISTER SELECT VALUES FOR
2981                                  ;EACH DEVICE REGISTER
2982
2983                                  ;       REGISTER        REG SEL
2984                                  ;       NAME            (16,8,4,2,1)
2985                                  ;
2986                                  ;       RMCS1           00000
2987                                  ;       RMDS            00001
2988                                  ;       RMER1           00010
2989                                  ;       RMMR1           00011
2990                                  ;       RMAS            00100
2991                                  ;       RMDA            00101
2992                                  ;       RMDT            00110
2993                                  ;       RMLA            00111
2994                                  ;       RMSN            01000
2995                                  ;       RMOF            01001
2996                                  ;       RMDC            01010
2997                                  ;       RMHR            01011
2998                                  ;       RMMR2           01100
2999                                  ;       RMER2           01101
3000                                  ;       RMEC1           01110
3001                                  ;       RMEC2           01111
3002
3003                                  ;EACH REGISTER SELECT LINE IS TESTED FOR A STUCK AT ONE,
3004                                  ;STUCK AT ZERO FAULT. AS AN EXAMPLE, TO TEST REG SEL 1.
3005                                  ;FOR S-A-0, RMER1 IS WRITTEN WITH ZEROS. THEN THE REGISTER
3006                                  ;THAT HAS THE SAME SELECT VALUE, EXCEPT FOR THE SELECT LINE
3007                                  ;BEING TESTED, IS WRITTEN WITH ONES. IN THIS EXAMPLE,
3008                                  ;RMMR1 IS WRITTEN WITH ONES. IF SELECT LINE 1 IS S-A-0,
3009                                  ;THE ALL ONES WORD WILL BE WRITTEN IN RMER1, AND RMER1
3010                                  ;WILL NOT BE 0 WHEN READ BACK.
3011
3012  007042  005002                          CLR     R2              ;R2= ZEROS SOURCE
3013  007044  012703  177777                  MOV     #-1,R3          ;R3= ONES  SOURCE
3014                                  ;TEST REG SEL 1 FOR S-A-0
3015  007050  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3016  007056  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3017
3018  007062  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
3019
3020  007066  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
3021
3022  007072  010360  000024                  MOV     R3,RMMR1(R0)    ;LOAD RMMR1
3023
3024  007076  010360  000036                  MOV     R3,RMHR(R0)     ;LOAD RMHR
3025
```

```
3026  007102  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
3027
3028  007110  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
3029  007116  020337  001342                  CMP     R3,RMER1I
3030  007122  C01007                          BNE     10$
3031  007124  052737  176000  001362          BIS     #XNUDC,RMDCI
3032  007132  020337  001362                  CMP     R3,RMDCI
3033  007136  001001                          BNE     10$
3034  007140  104010                          ERROR   10              ;REG SEL 1 IS S-A-0
3035  007142                          10$:
3036
3037                                  ;TEST REG SEL 1 FOR S-A-1
3038  007142  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3039  007150  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3040
3041  007154  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
3042
3043  007160  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
3044
3045  007164  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
3046
3047  007170  010360  000016                  MOV     R3,RMAS(R0)     ;LOAD RMAS
3048
3049  007174  010360  000030                  MOV     R3,RMSN(R0)     ;LOAD RMSN
3050
3051  007200  010360  000040                  MOV     R3,RMMR2(R0)    ;LOAD RMMR2
3052
3053  007204  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
3054
3055  007212  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
3056
3057  007220  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
3058  007226  020337  001334                  CMP     R3,RMDAI
3059  007232  001015                          BNE     20$
3060  007234  052737  161577  001360          BIS     #XNUOF,RMOFI
3061  007242  020337  001360                  CMP     R3,RMOFI
3062  007246  001007                          BNE     20$
3063  007250  052737  001567  001370          BIS     #XNUER2,RMER2I
3064  007256  020337  001370                  CMP     R3,RMER2I
3065  007262  001001                          BNE     20$
3066  007264  104011                          ERROR   11              ;REG SEL 1 IS S-A-1
3067  007266                          20$:
3068
3069                                  ;TEST REG SEL 2 FOR S-A-0
3070  007266  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3071  007274  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3072
3073  007300  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
3074
3075  007304  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
3076
3077  007310  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
3078
3079  007314  010360  000020                  MOV     R3,RMLA(R0)     ;LOAD RMLA
3080
3081  007320  010360  000036                  MOV     R3,RMHR(R0)     ;LOAD RMHR
```

```
3082
3083  007324  010360  000046                  MOV     R3,RMEC2(R0)    ;LOAD RMEC2
3084
3085  007330  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
3086
3087  007336  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
3088
3089  007344  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
3090  007352  020337  001334                  CMP     R3,RMDAI
3091  007356  001015                          BNE     30$
3092  007360  052737  161577  001360          BIS     #XNUOF,RMOFI
3093  007366  020337  001360                  CMP     R3,RMOFI
3094  007372  001007                          BNE     30$
3095  007374  052737  001567  001370          BIS     #XNUER2,RMER2I
3096  007402  020337  001370                  CMP     R3,RMER2I
3097  007406  001001                          BNE     30$
3098  007410  104012                          ERROR   12              ;REG SEL 2 IS S-A-0
3099  007412                          30$:
3100
3101                                  ;TEST REG SEL 2 FOR S-A-1
3102  007412  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3103  007420  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3104
3105  007424  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
3106
3107  007430  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
3108
3109  007434  012760  000076  000000          MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1
3110
3111  007442  010360  000030                  MOV     R3,RMSN(R0)     ;LOAD RMSN
3112
3113  007446  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
3114
3115  007454  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
3116  007462  052737  177701  001342          BIS     #^CILF76,RMER1I
3117  007470  020337  001342                  CMP     R3,RMER1I
3118  007474  001007                          BNE     40$
3119  007476  052737  176000  001362          BIS     #XNUDC,RMDCI
3120  007504  020337  001362                  CMP     R3,RMDCI
3121  007510  001001                          BNE     40$
3122  007512  104013                          ERROR   13              ;REG SEL 2 IS S-A-1
3123  007514                          40$:
3124
3125                                  ;TEST REG SEL 4 FOR S-A-0
3126  007514  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3127  007522  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3128
3129  007526  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
3130
3131  007532  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
3132
3133  007536  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
3134
3135  007542  010360  000026                  MOV     R3,RMDT(R0)     ;LOAD RMDT
3136
3137  007546  010360  000042                  MOV     R3,RMER2(R0)    ;LOAD RMER2
```

```
3138
3139  007552  010360  000044                  MOV     R3,RMEC1(R0)    ;LOAD RMEC1
3140
3141  007556  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
3142
3143  007564  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
3144
3145  007572  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
3146  007600  020337  001342                  CMP     R3,RMER1I
3147  007604  001015                          BNE     50$
3148  007606  052737  161577  001360          BIS     #XNUOF,RMOFI
3149  007614  020337  001360                  CMP     R3,RMOFI
3150  007620  001007                          BNE     50$
3151  007622  052737  176000  001362          BIS     #XNUDC,RMDCI
3152  007630  020337  001362                  CMP     R3,RMDCI
3153  007634  001001                          BNE     50$
3154  007636  104014                          ERROR   14              ;REG SEL 4 IS S-A-0
3155  007640                          50$:
3156
3157                                  ;TEST REG SEL 4 FOR S-A-1
3158  007640  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3159  007646  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3160
3161  007652  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
3162
3163  007656  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
3164
3165  007662  010360  000012                  MOV     R3,RMDS(R0)     ;LOAD RMDS
3166
3167  007666  010360  000032                  MOV     R3,RMOF(R0)     ;LOAD RMOF
3168
3169  007672  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
3170
3171  007700  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
3172  007706  020337  001334                  CMP     R3,RMDAI
3173  007712  001007                          BNE     60$
3174  007714  052737  001567  001370          BIS     #XNUER2,RMER2I
3175  007722  020337  001370                  CMP     R3,RMER2I
3176  007726  001001                          BNE     60$
3177  007730  104015                          ERROR   15              ;REG SEL 4 IS S-A-1
3178  007732                          60$:
3179
3180                                  ;TEST REG SEL 8 FOR S-A-0
3181  007732  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3182  007740  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3183
3184  007744  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
3185
3186  007750  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
3187
3188  007754  010360  000034                  MOV     R3,RMDC(R0)     ;LOAD RMDC
3189
3190  007760  010360  000042                  MOV     R3,RMER2(R0)    ;LOAD RMER2
3191
3192  007764  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
3193
```

```
3194  007772  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
3195  010000  020337  001342                  CMP     R3,RMER1I
3196  010004  001004                          BNE     70$
3197  010006  020337  001334                  CMP     R3,RMDAI
3198  010012  001001                          BNE     70$
3199  010014  104016                          ERROR   16              ;REG SEL 8 IS S-A-0
3200  010016                          70$:
3201
3202                                  ;TEST REG SEL 8 FOR S-A-1
3203  010016  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3204  010024  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3205
3206  010030  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
3207
3208  010034  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
3209
3210  010040  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
3211
3212  010044  010360  000012                  MOV     R3,RMDS(R0)     ;LOAD RMDS
3213
3214  010050  010360  000014                  MOV     R3,RMER1(R0)    ;LOAD RMER1
3215
3216  010054  010360  000006                  MOV     R3,RMDA(R0)     ;LOAD RMDA
3217
3218  010060  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
3219
3220  010066  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
3221
3222  010074  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
3223  010102  052737  161577  001360          BIS     #XNUOF,RMOFI
3224  010110  001015                          BNE     80$
3225  010112  022737  176000  001362          CMP     #XNUDC,RMDCI
3226  010120  020337  001362                  CMP     R3,RMDCI
3227  010124  001007                          BNE     80$
3228  010126  052737  001567  001370          BIS     #XNUER2,RMER2I
3229  010134  020337  001370                  CMP     R3,RMER2I
3230  010140  001001                          BNE     80$
3231  010142  104017                          ERROR   17              ;REG SEL 8 IS S-A-1
3232  010144                          80$:
3233
3234                                  ;REGISTER SELECT 16 IS TESTED BY THE ILR TEST
3235
3236                                  ;;*****************************************************************
3237                                  ;*TEST 7         DRIVE TYPE TEST
3238
3239                                  ;;*****************************************************************
3240  010144  000004                  TST7:   SCOPE
3241  010146  012737  000007  001226          MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
3242
3243  010154  000240                          NOP
3244  010156  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
3245  010164  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3246  010172  012737  010206  001122          MOV     #T7,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3247  010200  012737  010206  001124          MOV     #T7,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3248  010206                          T7:
3249  010206  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
```

```
3250  010212  013700  001276          MOV     $BASE,R0        ;R0 - UNIBUS ADDRESS OF UUT
3251  010216  013701  001456          MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3252
3253  010222  016037  000026  001142  MOV     RMDT(R0),$BDDAT ;STORE RMDT AT $BDDAT
3254  010230  022737  020024  001142  CMP     #SNGPRT,$BDDAT  ;SINGLE PORT RM03??
3255  010236  001432                  BEQ     10$             ;YES.!
3256  010240  022737  024024  001142  CMP     #DULPRT,$BDDAT  ;DUAL PORT RM03??
3257  010246  001426                  BEQ     10$             ;YES!!
3258  010250  022737  020025  001142  CMP     #SNGPRT!BIT0,$BDDAT     ;SINGLE PROT RM02 ?
3259  010256  001422                  BEQ     10$                     ;BRANCH IF SO
3260  010260  022737  024025  001142  CMP     #DULPRT!BIT0,$BDDAT     ;DUAL PORT RM02 ?
3261  010266  001416                  BEQ     10$                     ;BRANCH IF SO
3262  010270  012737  020024  001174  MOV     #SNGPRT,$TMP0   ;LOAD ACCEPTABLE VALUES
3263  010276  012737  024024  001176  MOV     #DULPRT,$TMP1
3264  010304  010037  001136          MOV     R0,$BDADR       ;LOAD BAD ADDRESS
3265  010310  062737  000026  001136  ADD     #RMDT,$BDADR
3266  010316  104057                  ERROR   57              ;DEVICE NOT AN RM03
3267  010320  000137  021442          JMP     $EOSP           ;GO TO NEXT DEVICE
3268  010324                  10$:
3269
3270                          ;;*******************************************************************
3271                          ;*TEST 10        DEVICE AVAILABLE TEST
3272
3273                          ;;*******************************************************************
3274  010324  000004          TST10:  SCOPE
3275  010326  012737  000010  001226  MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3276
3277  010334  000240                  NOP
3278  010336  012737  000024  001120  MOV     #20.,$ICNT      ;20 ITERATIONS
3279  010344  112737  000001  001131  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3280  010352  012737  010366  001122  MOV     #T10,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3281  010360  012737  010366  001124  MOV     #T10,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3282  010366                  T10:
3283  010366  012706  001100          MOV     #STACK,SP       ;LOAD THE STACK POINTER
3284  010372  013700  001276          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3285  010376  013701  001456          MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3286  010402  012760  000040  000010  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3287  010410  111160  000010          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3288
3289  010414  016037  000000  001142  MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
3290  010422  042737  173777  001142  BIC     #^CDVA,$BDDAT   ;CLEAR ALL BUT DVA
3291  010430  001006                  BNE     10$             ;BRANCH IF DVA SET
3292  010432  012737  004000  001140  MOV     #DVA,$GDDAT     ;SETUP EXPECTED
3293  010440  010037  001136          MOV     R0,$BDADR       ;SETUP REG ADDRESS
3294  010444  104060                  ERROR   60              ;DEVICE NOT AVAILABLE
3295  010446                  10$:
3296
3297                          ;;*******************************************************************
3298                          ;*TEST 11        SEARCH TIMEOUT TEST
3299
3300                          ;;*******************************************************************
3301  010446  000004          TST11:  SCOPE
3302  010450  012737  000011  001226  MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3303  010456  000240                  NOP
3304  010460  012737  000024  001120  MOV     #20.,$ICNT      ;20 ITERATIONS
3305  010466  112737  000001  001131  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
```

```
3306  010474  012737  010510  001122          MOV     #T11,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3307  010502  012737  010510  001124          MOV     #T11,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3308  010510                              T11:
3309  010510  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3310  010514  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3311  010520  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3312                                      ;LOAD REGISTER OUTPUT BUFFER WITH COMMAND PARAMETERS
3313  010524  012737  000000  001410          MOV     #0,RMDA0        ;SECTOR=0=TRACK
3314  010532  012737  000000  001436          MOV     #0,RMDC0        ;CYLINDER=0
3315  010540  012737  010000  001434          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
3316  010546  012737  054116  001406          MOV     #BUFFER,RMBA0   ;STARTING BUFFER ADDRESS
3317  010554  012737  177777  001404          MOV     #^C1+1,RMWC0    ;WORD COUNT
3318  010562  012737  000061  001402          MOV     #WD!GO,RMCS10   ;WRITE DATA COMMAND
3319
3320                                      ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3321  010570  004737  023352                  JSR     PC,ENBSCH
3322  010574  000402                          BR      10$             ;GO TO 10$ IF NO ERROR
3323  010576  104000                          ERROR                   ;RETURN HERE IF ERROR
3324  010600  000462                          BR      50$             ;SLOE REST PF TEST
3325  010602                              10$:
3326                                      ;START THE CLOCK AND WAIT FOR 100 MS
3327  010602  012737  000144  001524          MOV     #100.,WATCH     ;SET WATCHDOG TIMER VALUE
3328  010610  004777  170712                  JSR     PC,@CLOCK       ;START THE CLOCK
3329  010614  005737  001524              20$:TST     WATCH
3330  010620  001375                          BNE     20$
3331  010622  004777  170702                  JSR     PC,@STOP        ;STOP THE CLOCK
3332                                      ;VERIFY THAT OPI IS NOT SET (SEARCH TIMEOUT IS DISABLED)
3333
3334  010626  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3335  010634  010037  001136                  MOV     R0,$BDADR       ;SET UP FOR ERROR MSG
3336  010640  062737  000014  001136          ADD     #RMER1,$BDADR
3337  010646  042737  157777  001142          BIC     #^COPI,$BDDAT
3338  010654  001404                          BEQ     30$             ;BRANCH IF NO ERROR
3339  010656  005037  001140                  CLR     $GDDAT
3340  010662  104020                          ERROR   20                      ;OPI SET W/SEARCH TIMEOUT
3341                                                                  ;DISABLED
3342  010664  000430                          BR      50$             ;SKIP
3343  010666                              30$:
3344                                      ;ENABLE SEARCH TIMEOUT, THEN WAIT 100 MS
3345
3346  010666  012760  041401  000024          MOV     #DMD!MUR!DBEN.MOC,RMMR1(R0)     ;LOAD RMMR1
3347  010674  012737  000144  001524          MOV     #100.,WATCH     ;SET WATCHDOG TIMER VALUE
3348  010702  004777  170620                  JSR     PC,@CLOCK       ;START THE CLOCK
3349  010706  005737  001524              40$:TST     WATCH
3350  010712  001375                          BNE     40$
3351  010714  004777  170610                  JSR     PC,@STOP        ;STOP THE CLOCK
3352                                      ;OPI SHOULD NOW BE SET (SEARCH TIMEOUT IS ENABLED)
3353
3354  010720  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3355  010726  042737  157777  001142          BIC     #^COPI,$BDDAT
3356  010734  001004                          BNE     50$
3357  010736  012737  020000  001140          MOV     #OPI,$GDDAT
3358  010744  104021                          ERROR   21                      ;OPI NOT SET BY SEARCH TIMEOUT
3359  010746                              50$:
3360
3361                                      ;;*****************************************************************
```

```
3362                                        ;*TEST 12       SET DTE TEST
3363
3364                                        ;;*****************************************************************
3365  010746  000004                        TST12:  SCOPE
3366  010750  012737  000012  001226                MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3367  010756  000240                                NOP
3368  010760  012737  000024  001120                MOV     #20.,$ICNT      ;20 ITERATIONS
3369  010766  112737  000001  001131                MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3370  010774  012737  011010  001122                MOV     #T12,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3371  011002  012737  011010  001124                MOV     #T12,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3372  011010                                T12:
3373  011010  012706  001100                        MOV     #STACK,SP       ;LOAD THE STACK POINTER
3374  011014  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3375  011020  013701  001456                        MOV     TSTOUE,R1       ;R1 - POINTER TO DEVICE
3376  011024  010037  001136                        MOV     R0,$BDADR       ;SETUP ERROR MSG
3377  011030  062737  000014  001136                ADD     #RMER1,$BDADR
3378  011036  005037  001140                        CLR     $GDDAT
3379                                        ;INITILAIZE AND VERIFY THAT DRIVE TIMING ERROR IS RESET
3380  011042  012760  000040  000010                MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3381  011050  111160  000010                        MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3382
3383  011054  016037  000014  001142                MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3384  011062  042737  167777  001142                BIC     #^CDTE,$BDDAT
3385  011070  001402                                BEQ     10$             ;BRANCH IF DTE=0
3386  011072  104031                                ERROR   31                      ;CANT RESET DTE
3387  011074  000517                                BR      50$             ;SKIP REST OF TEST
3388                                        ;SET MAINTENANCE INDEX PULSE AND VERIFY DTE REMAINS RESET
3389  011076                                10$:
3390
3391  011076  012760  000001  000024                MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3392
3393  011104  012760  000005  000024                MOV     #DMD!MI,RMMR1(R0)       ;LOAD RMMR1
3394
3395  011112  016037  000014  001142                MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3396  011120  042737  167777  001142                BIC     #^CDTE,$BDDAT
3397  011126  001402                                BEQ     20$
3398  011130  104000                                ERROR                   ;DTE SET WHEN SECTOR
3399  011132  000500                                BR      50$             ;COMPARE SHOULD BE RESET
3400  011134                                20$:
3401                                        ;EXECUTE DUMMY DATA COMMAND TO ENABLE SEARCH
3402  011134  012737  000000  001410                MOV     #0,RMDAO
3403  011142  012737  000005  001436                MOV     #5,RMDCO
3404  011150  012737  010000  001434                MOV     #FMT16,RMOFO
3405  011156  012737  054116  001406                MOV     #BUFFER,RMBAO
3406  011164  012737  177777  001404                MOV     #^C1+1,RMWCO
3407  011172  012737  000061  001402                MOV     #WD!GO,RMCS1O
3408
3409                                        ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3410  011200  004737  023352                        JSR     PC,ENBSCH
3411  011204  000402                                BR      30$             ;GO TO 30$ IF NO ERROR
3412  011206  104000                                ERROR                   ;RETURN HERE IF ERROR
3413  011210  000451                                BR      50$
3414                                        ;WITH SEARCH ENABLED, SET AND RESET SECTOR PULSE TO SET ENABLE
3415                                        ;SEARCH FLOP.
3416  011212                                30$:
3417
```

```
3418  011212  012760  051441  000024          MOV     #DMD!MUR.DBEN!MOC.DTO!MS,RMMR1(R0)      ;LOAD RMMR1
3419
3420  011220  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3421                          ;SET SECTOR PULSE AND VERIFY DTE DOES NOT SET
3422                          ;       PUTMR1  #DMD!MUR!DBEN!MOC.DTO!MS
3423                          ;       PUTMR1  #DMD!MUR!DBEN!MOC.DTO
3424
3425  011226  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3426  011234  042737  167777  001142          BIC     #^CDTE,$BDDAT
3427  011242  001402                          BEQ     40$
3428  011244  104023                          ERROR   23              ;DTE SET WHEN SECTOR
3429  011246  001432                          BEQ     50$             ;COMPARE SHOULD BE RESET
3430  011250                          40$:
3431                          ;FORCE SECTOR COMPARE
3432
3433  011250  012760  051403  000024          MOV     #DMD.MUR!DBEN.MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
3434
3435  011256  012760  051443  000024          MOV     #DMD!MUR.DBEN!MOC.DTO.MSC.MS,RMMR1(R0)  ;LOAD RMMR1
3436
3437  011264  012760  051403  000024          MOV     #DMD!MUR!DBEN.MOC!DTO.MSC,RMMR1(R0)     ;LOAD RMMR1
3438                          ;SET SECTOR PULSE AND VERIFY DTE SETS
3439
3440  011272  012760  051441  000024          MOV     #DMD!MUR!DBEN.MOC!DTO!MS,RMMR1(R0)      ;LOAD RMMR1
3441
3442  011300  012760  051401  000024          MOV     #DMD!MUR.DBEN!MOC.DTO,RMMR1(R0) ;LOAD RMMR1
3443
3444  011306  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3445  011314  042737  167777  001142          BIC     #^CDTE,$BDDAT
3446  011322  001004                          BNE     50$
3447  011324  012737  010000  001142          MOV     #DTE,$BDDAT
3448  011332  104024                          ERROR   24                      ;COULD NOT SET DTE WITH
3449                                                                  ;SECTOR COMPARE SET
3450  011334                          50$:
3451
3452                          ;;**************************************************************************
3453                          ;*TEST 13        FORMAT CHANGE TEST
3454
3455                          ;;**************************************************************************
3456  011334  000004          TST13:  SCOPE
3457  011336  012737  000013  001226          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3458  011344  000240                          NOP
3459  011346  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
3460  011354  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3461  011362  012737  011376  001122          MOV     #T13,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3462  011370  012737  011376  001124          MOV     #T13,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3463  011376                          T13:
3464  011376  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3465  011402  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3466  011406  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3467  011412  012702  011752                  MOV     #50$,R2         ;R2=TABLE POINTER
3468  011416                          10$:
3469                          ;;**************************************************************************
3470                          ;INITILAIZE AND SET THE FORMAT BIT, USE INDEX PULSE TO CLEAR FORMAT CHANGE
3471  011416  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3472  011424  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3473
```

```
3474  011430  011260  000032                  MOV     (R2),RMOF(R0)    ;LOAD RMOF
3475
3476  011434  012760  000001  000024          MOV     #DMD,RMMR1(R0)   ;LOAD RMMR1
3477
3478  011442  012760  000005  000024          MOV     #DMD!MI,RMMR1(R0)       ;LOAD RMMR1
3479                                  ;SETUP AND EXECUTE DUMMY DATA COMMAND USING OPPOSITE FORMAT
3480  011450  012737  000000  001410          MOV     #0,RMDAO
3481  011456  012737  000005  001436          MOV     #5,RMDCO
3482  011464  016237  000002  001434          MOV     2(R2),RMOFO
3483  011472  012737  054116  001406          MOV     #BUFFER,RMBAO
3484  011500  012737  177777  001404          MOV     #^C1+1,RMWCO
3485  011506  012737  000061  001402          MOV     #WD!GO,RMCS1O
3486
3487                                  ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3488  011514  004737  023352                  JSR     PC,ENBSCH
3489  011520  000402                          BR      20$             ;GO TO 20$ IF NO ERROR
3490  011522  104000                          ERROR                   ;RETURN HERE IF ERROR
3491  011524  000515                          BR      60$
3492  011526                          20$:
3493                                  ;FORMAT CHANGE FLOP SHOULD BE SET - VERIFY BY TRYING TO FORCE A
3494                                  ;DRIVE TIMING ERROR WHICH SHOULD NOT SET.
3495
3496  011526  012760  051403  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
3497
3498  011534  012760  051443  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MSC!MS,RMMR1(R0)  ;LOAD RMMR1
3499
3500  011542  012760  051403  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
3501
3502  011550  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3503                                  ;VERIFY THAT DRIVE TIMING ERROR DIDNT SET
3504
3505  011556  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3506  011564  042737  167777  001142          BIC     #^CDTE,$BDDAT
3507  011572  001416                          BEQ     30$
3508  011574  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
3509  011600  062737  000014  001136          ADD     #RMER1,$BDADR
3510  011606  005037  001140                  CLR     $GDDAT
3511  011612  011237  001174                  MOV     (R2),$TMP0
3512  011616  016237  000002  001176          MOV     2(R2),$TMP1
3513  011624  104025                          ERROR   25                      ;DTE SET WHEN THERE WAS
3514  011626  000454                          BR      60$             ;A FORMAT CHANGE
3515  011630                          30$:
3516                                  ;CLEAR THE FORMAT CHANGE FLOP W/INDEX PULSE
3517
3518  011630  012760  051405  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MI,RMMR1(R0)       ;LOAD RMMR1
3519                                  ;ENABLE SEARCH AND FORCE SECTOR COMPARE
3520
3521  011636  012760  051403  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
3522
3523  011644  012760  051443  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MSC!MS,RMMR1(R0)  ;LOAD RMMR1
3524
3525  011652  012760  051403  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
3526                                  ;SET DTE W/ANOTHER SECTOR PULSE - VERIFY DTE IS SET
3527
3528  011660  012760  051441  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MS,RMMR1(R0)      ;LOAD RMMR1
3529
```

```
3530  011666  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3531
3532  011674  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3533  011702  042737  167777  001142          BIC     #^CDTE,$BDDAT
3534  011710  001012                          BNE     40$
3535  011712  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
3536  011716  062737  000014  001136          ADD     #RMER1,$BDADR
3537  011724  012737  010000  001140          MOV     #DTE,$GDDAT
3538  011732  104027                          ERROR   27                      ;COULD NOT SET DTE
3539  011734  000411                          BR      60$
3540  011736                          40$:
3541                                  ;;**********************************************************************
3542                                  ;DO TEST W/18 TO 16 FORMAT CHANGE DURING SECOND EXECUTION
3543  011736  022702  011752                  CMP     #50$,R2
3544  011742  001006                          BNE     60$             ;BRANCH IF DONE TEST
3545  011744  062702  000002                  ADD     #2,R2           ;MOVE POINTER
3546  011750  000622                          BR      10$             ;REPEAT TEST
3547
3548  011752  010000                  50$:    .WORD   #FMT16          ;16-18 FORMAT CHANGE
3549  011754  000000                          .WORD   0               ;18-16 FORMAT CHANGE
3550  011756  010000                          .WORD   #FMT16
3551  011760                          60$:                            ;END OF TEST
3552
3553                                  ;;**********************************************************************
3554                                  ;*TEST 14        PROM STROBE TEST
3555
3556                                  ;;**********************************************************************
3557  011760  000004                  TST14:  SCOPE
3558  011762  012737  000014  001226          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3559  011770  000240                          NOP
3560  011772  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
3561  012000  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3562  012006  012737  012022  001122          MOV     #T14,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3563  012014  012737  012022  001124          MOV     #T14,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3564  012022                          T14:
3565  012022  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3566  012026  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3567  012032  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3568  012036  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3569  012044  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3570  012050  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
3571  012054  062737  000024  001136          ADD     #RMMR1,$BDADR
3572                                  ;SET DIAGNOSTIC MODE AND TRY TO RESET PROM STROBE
3573
3574  012062  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3575  012070  012702  000021                  MOV     #17.,R2         ;R2=MAXIMUM # CLOCK PULSES
3576  012074                          5$:
3577
3578  012074  012760  004001  000024          MOV     #DMD!MCLK,RMMR1(R0)     ;LOAD RMMR1
3579
3580  012102  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3581
3582  012110  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3583  012116  042737  177737  001142          BIC     #^CWC,$BDDAT
3584  012124  001406                          BEQ     6$
3585  012126  005302                          DEC     R2
```

```
3586  012130  001361                        BNE     5$
3587  012132  005037  001140                CLR     $GDDAT
3588  012136  104000                        ERROR                   ;CAN'T CLEAR PROM STROBE
3589  012140  000501                        BR      60$
3590  012142  012702  000021        6$:     MOV     #17.,R2
3591  012146                        10$:
3592
3593  012146  012760  004001  000024        MOV     #DMD.MCLK,RMMR1(R0)     ;LOAD RMMR1
3594
3595  012154  012760  000001  000024        MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3596
3597  012162  016037  000024  001142        MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3598  012170  042737  177737  001142        BIC     #^CWC,$BDDAT
3599  012176  001007                        BNE     20$             ;EXIT LOOP WHEN PROM STROBE ON
3600  012200  005302                        DEC     R2
3601  012202  001361                        BNE     10$
3602  012204  012737  000040  001140        MOV     #WC,$GDDAT
3603  012212  104030                        ERROR   30                      ;CANT SET PROM STROBE
3604  012214  000453                        BR      60$
3605  012216                        20$:
3606                                ;VERIFY PROM STROBE IS ON FOR 3 BIT CLOCKS
3607  012216  012702  000003                MOV     #3.,R2
3608  012222                        25$:
3609
3610  012222  012760  004001  000024        MOV     #DMD!MCLK,RMMR1(R0)     ;LOAD RMMR1
3611
3612  012230  012760  000001  000024        MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3613
3614  012236  016037  000024  001142        MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3615  012244  042737  177737  001142        BIC     #^CWC,$BDDAT
3616  012252  001005                        BNE     30$             ;BRANCH IF PROM STORBE DID NOT
3617                                                                ;DROP EARLY
3618  012254  012737  000040  001140        MOV     #WC,$GDDAT
3619  012262  104032                        ERROR   32
3620  012264  000427                        BR      60$
3621  012266  005302                30$:    DEC     R2
3622  012270  001354                        BNE     25$
3623                                ;VERIFY PROM STROBE IS OFF FOR 8 BIT CLOCKS
3624  012272  012702  000010                MOV     #8.,R2
3625  012276                        40$:
3626
3627  012276  012760  004001  000024        MOV     #DMD!MCLK,RMMR1(R0)     ;LOAD RMMR1
3628
3629  012304  012760  000001  000024        MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3630
3631  012312  016037  000024  001142        MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3632  012320  042737  177737  001142        BIC     #^CWC,$BDDAT
3633  012326  001404                        BEQ     50$             ;BRANCH IF PROM STROBE
3634  012330  005037  001140                CLR     $GDDAT          ;DID NOT SET EARLY
3635  012334  104033                        ERROR   33
3636  012336  000402                        BR      60$
3637  012340  005302                50$:    DEC     R2
3638  012342  001355                        BNE     40$
3639  012344                        60$:
3640
3641                                ;;*******************************************************************************
```

```
                                        ;*TEST 15       SYNC WORD COUNT INHIBIT TEST

                                        ;;*****************************************************************
012344  000004                          TST15:  SCOPE
012346  012737  000015  001226                  MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
012354  000240                                  NOP
012356  012737  000024  001120                  MOV     #20.,$ICNT      ;20 ITERATIONS
012364  112737  000001  001131                  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
012372  012737  012406  001122                  MOV     #T15,$LPADR  ;LOAD LOOP ON TEST ADDRESS
012400  012737  012406  001124                  MOV     #T15,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
012406                                  T15:
012406  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
012412  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
012416  013701  001456                          MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                        ;SETUP REGISTER OUTPUT BUFFER FOR SUBROUTINES
012422  012737  000000  001410                  MOV     #0,RMDAO        ;SECTOR=0, TRACK=0
012430  012737  000005  001436                  MOV     #5,RMDCO        ;CYLINDER=0
012436  012737  010000  001434                  MOV     #FMT16,RMOFO    ;16 BIT FORMAT
012444  012737  054116  001406                  MOV     #BUFFER,RMBAO   ;STARTING BUFFER ADDRESS
012452  012737  177777  001404                  MOV     #^C1+1,RMWCO    ;WORD COUNT
012460  012737  000061  001402                  MOV     #WD!GO,RMCS1O   ;WRITE DATA COMMAND
012466  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
012474  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

                                        ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
012500  004737  023352                          JSR     PC,ENBSCH
012504  000402                                  BR      10$             ;GO TO 10$ IF NO ERROR
012506  104000                                  ERROR                   ;RETURN HERE IF ERROR
012510  000562                                  BR      120$            ;SKIP IF FAILURE
012512                                  10$:
                                        ;FORCE SECTOR COMPARE USING SUBROUTINE
012512  004737  024232                          JSR     PC,SCTCMP
012516  000402                                  BR      20$             ;GO TO 20$ IF NO ERROR
012520  104000                                  ERROR                   ;RETURN HERE IF ERROR
012522  000555                                  BR      120$
                                        ;;*****************************************************************
                                        ;CAN NOW STEP DATA TIMING SEQUENCER WITH SECTOR COMPARE SET

                                        ;STEP THE SEQUENCER TO LOCATION 10(10) WHILE CHECKING 'PLFS'
012524  012702  000000                  20$:    MOV     #0,R2           ;R2=SEQUENCER ADDRESS
012530                                  30$:
                                        ;PULSE MAINTENANCE CLOCK UNTIL PROM STROBE SETS

012530  012760  055401  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1

012536  012760  051401  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
                                        ;SEE IF PROM STROBE IS SET

012544  016003  000024                          MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
012550  032703  000040                          BIT     #WC,R3
012554  001765                                  BEQ     30$             ;ISSUE NEXT BIT CLOCK
012556  005202                                  INC     R2              ;INCREMENT SEQUENCER ADDRESS
                                        ;CHECK 'LOOKING FOR SYNC' - SHOULD BE ZERO UNTIL LOCATION 11
012560  010337  001142                          MOV     R3,$BDDAT
012564  042737  175777  001142                  BIC     #^CPLFS,$BDDAT
012572  001413                                  BEQ     50$             ;BRANCH IF PLFS IS ZERO
```

```
3698  012574  005037  001140                  CLR     $GDDAT
3699  012600  010037  001136                  MOV     R0,$BDADR
3700  012604  062737  000024  001136          ADD     #RMMR1,$BDADR
3701  012612  010237  001174                  MOV     R2,$TMP0
3702  012616  104034                          ERROR   34                      ;PLFS SHOULD NOT BE SET
3703  012620  000516                          BR      120$            ;SKIP REST OF TEST
3704  012622                          50$:
3705                                  ;EXIT LOOP IF SEQUENCER NOW AT LOCATION 10
3706  012622  022702  000012                  CMP     #10.,R2
3707  012626  001414                          BEQ     70$
3708                                  ;PULSE MAINTENANCE CLOCK UNTIL PROM STROBE RESETS
3709  012630                          60$:
3710
3711  012630  012760  055401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
3712
3713  012636  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3714                                  ;SEE IF PROM STROBE IS RESET
3715
3716  012644  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
3717  012650  032703  000040                  BIT     #WC,R3
3718  012654  001365                          BNE     60$
3719  012656  000724                          BR      30$             ;GO STEP SEQUENCER TO NEXT LOC
3720  012660                          70$:
3721                                  ;;*****************************************************************
3722                                  ;THE NEXT PROM STROBE SHOULD SET 'PLFS'' AT LOCATION 11(10) OF THE
3723                                  ;DATA TIMING SEQUENCER.
3724  012660  012702  000020                  MOV     #16.,R2         ;R2=NUMBER OF BIT CLOCKS
3725                                  ;ISSUE 16 BIT CLOCKS TO GET NEXT PROM STROBE
3726  012664                          80$:
3727
3728  012664  012760  055401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
3729
3730  012672  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3731  012700  005302                          DEC     R2
3732  012702  001370                          BNE     80$
3733                                  ;VERIFY THAT 'PLFS'' IS NOW SET
3734
3735  012704  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3736  012712  042737  175777  001142          BIC     #^CPLFS,$BDDAT
3737  012720  001012                          BNE     90$             ;BRANCH IF PLFS IS SET
3738  012722  012737  002000  001140          MOV     #PLFS,$GDDAT    ;SETUP ERROR MESSAGE
3739  012730  010037  001136                  MOV     R0,$BDADR
3740  012734  062737  000024  001136          ADD     #RMMR1,$BDADR
3741  012742  104035                          ERROR   35                      ;'PLFS'' DID NOT SET
3742  012744  000444                          BR      120$            ;SKIP REST OF TEST
3743  012746                          90$:
3744
3745                                  ;ISSUE 3 MORE BIT CLOCKS TO RESET PROM STROBE
3746
3747  012746  012702  000003                  MOV     #3,R2           ;
3748  012752                          95$:
3749
3750  012752  012760  055401  000024          MOV     #DMD!MUR!DBEN!MOC.DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
3751
3752  012760  012760  051401  000024          MOV     #DMD.MUR.DBEN.MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3753  012766  005302                          DEC     R2
```

```
012770  001370                                  BNE     95$
                                ;;******************** ********************************************
                                ;WITH 'LOOKING FOR SYNC SET, FURTHER BIT CLOCKS SHOULD NOT SET
                                ;PROM STROBE
012772  012702  000040                          MOV     #32.,R2          ;R2=NUMBER OF BIT CLOCKS
012776                          100$:
                                ;PULSE BIT CLOCK AND VERIFY PROM STROBE DOES NOT SET

012776  012760  055401  000024                  MOV     #DMD.MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)     ;LOAD RMMR1

013004  012760  051401  000024                  MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1

013012  016003  000024                          MOV     RMMR1(R0),R3     ;STORE RMMR1 AT R3
013016  042703  177737                          BIC     #^CWC,R3
013022  001413                                  BEQ     110$             ;BRANCH IF PROM STROBE IS 0
013024  005037  001140                          CLR     $GDDAT           ;SETUP ERROR MESSAGE
013030  010337  001142                          MOV     R3,$BDDAT
013034  010037  001136                          MOV     R0,$BDADR
013040  062737  000024  001136                  ADD     #RMMR1,$BDADR
013046  104036                                  ERROR   36                     ;PROM STROBE WHILE LOOKING FOR SYNC
013050  000402                                  BR      120$
013052  005302                  110$:           DEC     R2               ;CONTINUE FOR 32 BIT CLOCKS
013054  001350                                  BNE     100$

013056                          120$:                                    ;END OF TEST
                                ;;*****************************************************************
                                ;*TEST 16        SYNC DETECTION TEST

                                ;;*****************************************************************
013056  000004                  TST16:  SCOPE
013060  012737  000016  001226          MOV     #16,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
013066  000240                          NOP
013070  012737  000024  001120          MOV     #20.,$ICNT       ;20 ITERATIONS
013076  112737  000001  001131          MOVB    #1,$ERMAX        ;ONE ERROR ALLOWED
013104  012737  013120  001122          MOV     #T16,$LPADR  ;LOAD LOOP ON TEST ADDRESS
013112  012737  013120  001124          MOV     #T16,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
013120                          T16:
013120  012706  001100                  MOV     #STACK,SP        ;LOAD THE STACK POINTER
013124  013700  001276                  MOV     $BASE,R0         ;R0 = UNIBUS ADDRESS OF UUT
013130  013701  001456                  MOV     TSTQUE,R1        ;R1 = POINTER TO DEVICE
                                ;SETUP REGISTER OUTPUT BUFFER FOR SUBROUTINES
013134  012737  000000  001410          MOV     #0,RMDA0         ;SECTOR 0,TRACK 0
013142  012737  000005  001436          MOV     #5,RMDC0         ;CYLINDER 5
013150  012737  010000  001434          MOV     #FMT16,RMOF0     ;16 BIT MODE
013156  012737  054116  001406          MOV     #BUFFER,RMBA0    ;BUFFER ADDRESS
013164  012737  177777  001404          MOV     #^C1+1,RMWC0     ;CORD COUNT
013172  012737  000071  001402          MOV     #RD!GO,RMCS10    ;READ DATA COMMAND

                                ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
013200  004737  023352                  JSR     PC,ENBSCH
013204  000402                          BR      10$              ;GO TO 10$ IF NO ERROR
013206  104000                          ERROR                    ;RETURN HERE IF ERROR
013210  000550                          BR      100$
013212                          10$:
                                ;FORCE SECTOR COMPARE USING SUBROUTINE
013212  004737  024232                  JSR     PC,SCTCMP
```

```
3810  013216  000402                          BR      20$             ;GO TO 20$ IF NO ERROR
3811  013220  104000                          ERROR                   ;RETURN HERE IF ERROR
3812  013222  000543                          BR      100$
3813  013224                          20$:
3814                                  ;SET 'LOOKING FOR SYNC'' USING SUBROUTINE
3815  013224  004737  024340                  JSR     PC,SETLFS
3816  013230  000402                          BR      30$
3817  013232  104000                          ERROR
3818  013234  000536                          BR      100$
3819  013236                          30$:
3820                                  ;;*****************************************************************
3821                                  ;CLOCK ALL ONES SYNC PATTERN AND VERIFY SYNC IS NOT DETECTED
3822                                  ;(USING PROM STORBE AS INDICATION)
3823
3824  013236  012702  000020                  MOV     #16.,R2         ;NUMBER OF ONE BITS
3825  013242  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
3826  013246  062737  000024  001136          ADD     #RMMR1,$BDADR
3827  013254  005037  001140                  CLR     $GDDAT
3828
3829  013260  012760  053401  000024          MOV     #MR1AAA!MRD,RMMR1(R0)   ;LOAD RMMR1
3830  013266                          40$:
3831
3832  013266  012760  057401  000024          MOV     #MR1AAA!MRD!MCLK,RMMR1(R0)      ;LOAD RMMR1
3833
3834  013274  012760  053401  000024          MOV     #MR1AAA!MRD,RMMR1(R0)   ;LOAD RMMR1
3835
3836  013302  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3837  013310  042737  177737  001142          BIC     #^CWC,$BDDAT
3838  013316  001405                          BEQ     50$
3839  013320  012737  177777  001174          MOV     #177777,$TMP0
3840  013326  104037                          ERROR   37
3841  013330  000500                          BR      100$
3842  013332                          50$:
3843  013332  005302                          DEC     R2              ;REPEAT FOR 16 BITS
3844  013334  001354                          BNE     40$
3845  013336  012702  000020                  MOV     #16.,R2         ;NUMBER OF ZERO BITS
3846
3847                                  ;;*****************************************************************
3848                                  ;CLOCK ALL ZERO SYNC PATTERN AND VERIFY SYNC IS NOT DETECTED
3849  013342                          60$:
3850
3851  013342  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
3852
3853  013350  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
3854
3855  013356  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3856  013364  042737  177737  001142          BIC     #^CWC,$BDDAT    ;MAKE SURE SYNC NOT DETECTED
3857  013372  001404                          BEQ     70$
3858  013374  005037  001174                  CLR     $TMP0
3859  013400  104037                          ERROR   37                      ;SYNC DETECTED W/ALL 0'S PATTERN
3860  013402  000453                          BR      100$            ;SKIP IF FAILURE
3861  013404                          70$:
3862  013404  005302                          DEC     R2              ;REPEAT FOR 16 BITS
3863  013406  001355                          BNE     60$
3864  013410  012702  000040                  MOV     #32.,R2         ;R2=NUMBER OF BITS
3865  013414  012703  000031                  MOV     #000031,R3      ;R3=SYNC PATTERN FOR LEFT SHIFT
```

```
3866
3867                                    ;;**********************  ********************************************
3868                                    ;CLOCK THE BINARY SYNC PATTERN (10011000) AND VERIFY THAT SYNC IS
3869                                    ;DETECTED
3870  013420  012737  051401  001426  80$:    MOV     #MR1AAA,RMMR10  ;GENERATE VALUE OF RMMR1
3871  013426  000241                          CLC                     ;CLEAR THE CARR
3872  013430  006003                          ROR     R3              ;SHIFT RIGHT
3873  013432  103003                          BCC     90$             ;BRANCH IF C BIT CLEAR
3874  013434  052737  002000  001426          BIS     #MRD,RMMR10     ;SET MRD IF PATTER BIT SETS
3875  013442                          90$:
3876
3877  013442  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
3878  013450  052737  004000  001426          BIS     #MCLK,RMMR10            ;SET THE MAINTENANCE DATA CLK
3879
3880  013456  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
3881  013464  042737  004000  001426          BIC     #MCLK,RMMR10            ;RESET BIT CLOCK
3882
3883  013472  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
3884
3885  013500  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3886  013506  042737  177737  001142          BIC     #^CWC,$BDDAT
3887  013514  001006                          BNE     100$            ;BRANCH IF PROM STROBE IS SET
3888  013516  005302                          DEC     R2              ;CONTINME FOR 16 BIT CLOCKS
3889  013520  001337                          BNE     80$
3890  013522  012737  000040  001140          MOV     #WC,$GDDAT
3891  013530  104040                          ERROR   40                      ;SYNC PATTERN NOT DETECTED
3892  013532                          100$:                           ;END OF TEST
3893
3894                                    ;;**************************************************************************
3895                                    ;*TEST 17        ABORT SYNC DETECTION TEST
3896
3897                                    ;;**************************************************************************
3898  013532  000004                  TST17:  SCOPE
3899  013534  012737  000017  001226          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3900  013542  000240                          NOP
3901  013544  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
3902  013552  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3903  013560  012737  013574  001122          MOV     #T17,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3904  013566  012737  013574  001124          MOV     #T17,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3905  013574                          T17:
3906  013574  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3907  013600  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3908  013604  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3909                                    ;SETUP REGISTER OUTPUT BUFFER FOR SUBROUTINES
3910  013610  012737  000000  001410          MOV     #0,RMDA0        ;SECTOR 0, TRACK 0
3911  013616  012737  000005  001436          MOV     #5,RMDC0        ;CYLINDER 5
3912  013624  012737  010000  001434          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
3913  013632  012737  054116  001406          MOV     #BUFFER,RMBA0   ;STARTING BUFFER ADDRESS
3914  013640  012737  177777  001404          MOV     #^C1+1,RMWC0    ;WORD COUNT
3915  013646  012737  000071  001402          MOV     #RD!GO,RMCS10   ;READ DATA COMMAND
3916
3917                                    ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3918  013654  004737  023352                  JSR     PC,ENBSCH
3919  013660  000402                          BR      10$             ;GO TO 10$ IF NO ERROR
3920  013662  104000                          ERROR                   ;RETURN HERE IF ERROR
3921  013664  000447                          BR      50$
```

```
3922  013666                                  10$:
3923                                          ;FORCE SECTOR COMPARE USING SUBROUTINE
3924  013666  004737  024232                        JSR     PC,SCTCMP
3925  013672  000402                                BR      20$             ;GO TO 20$ IF NO ERROR
3926  013674  104000                                ERROR                   ;RETURN HERE IF ERROR
3927  013676  000442                                BR      50$
3928  013700                                  20$:
3929                                          ;SET 'LOOKING FOR SYNC'' USING SUBROUTINE
3930  013700  004737  024340                        JSR     PC,SETLFS
3931  013704  000402                                BR      30$
3932  013706  104000                                ERROR
3933  013710  000435                                BR      50$
3934  013712                                  30$:
3935                                          ;;**************************************************************************
3936                                          ;LOOKING FOR SYNC INHIBITS WORD CLOCK,HOWEVER, 'DRIVE TIMING ERROR''
3937                                          ;SHOULD RESET 'LFS WORD COUNT INHIBIT'' FLOP
3938                                          ;FORCE DRIVE TIMING ERROR
3939
3940
3941  013712  012760  051441  000024                MOV     #MR1AAA!MS,RMMR1(R0)    ;LOAD RMMR1
3942                                          ;PULSE BIT CLOCK AND VERIFY PROM STROBE SETS
3943  013720  012702  000020                        MOV     #16.,R2         ;R2, NUMBER OF BIT CLOCKS
3944  013724                                  40$:
3945
3946  013724  012760  055441  000024                MOV     #MR1AAA!MS.MCLK,RMMR1(R0)       ;LOAD RMMR1
3947
3948  013732  012760  051441  000024                MOV     #MR1AAA!MS,RMMR1(R0)    ;LOAD RMMR1
3949
3950  013740  016037  000024  001142                MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3951  013746  042737  177737  001142                BIC     #^CWC,$BDDAT
3952  013754  001013                                BNE     50$
3953  013756  005302                                DEC     R2
3954  013760  001361                                BNE     40$
3955  013762  012737  000040  001140                MOV     #WC,$GDDAT
3956  013770  010037  001136                        MOV     R0,$BDADR
3957  013774  062737  000024  001136                ADD     #RMMR1,$BDADR
3958  014002  104041                                ERROR   41                      ;DTE DIDNOT CLEAR WORD COUNT INH
3959  014004                                  50$:
3960
3961
3962                                          ;;**************************************************************************
3963                                          ;*TEST 20        SYNC GENERATION TEST
3964
3965                                          ;;**************************************************************************
3966  014004  000004                          TST20:  SCOPE
3967  014006  012737  000020  001226                MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3968  014014  000240                                NOP
3969  014016  012737  000024  001120                MOV     #20.,$ICNT      ;20 ITERATIONS
3970  014024  112737  000001  001131                MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3971  014032  012737  014046  001122                MOV     #T20,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3972  014040  012737  014046  001124                MOV     #T20,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3973  014046                                  T20:
3974  014046  012706  001100                        MOV     #STACK,SP       ;LOAD THE STACK POINTER
3975  014052  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3976  014056  013701  001456                        MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3977                                          ;SETUP REGISTER OUTPUT BUFFER FOR SUBROUTINES
```

```
3978  014062  012737  000000  001410          MOV     #0,RMDAO                ;SECTOR 0, TRACK 0
3979  014070  012737  000005  001436          MOV     #5,RMDCO                ;CYL 5
3980  014076  012737  010000  001434          MOV     #FMT16,RMOFO            ;16 BIT MODE
3981  014104  012737  054116  001406          MOV     #BUFFER,RMBAO           ;BUFFER ADDRESS
3982  014112  012737  177776  001404          MOV     #^C2+1,RMWCO            ;WORD COUNT
3983  014120  012737  000063  001402          MOV     #WH.GO,RMCS1O           ;WHITE HEAD AND DATA COM
3984
3985                                  ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3986  014126  004737  023352                  JSR     PC,ENBSCH
3987  014132  000403                          BR      10$             ;GO TO 10$ IF NO ERROR
3988  014134  104000                          ERROR                   ;RETURN HERE IF ERROR
3989                                  ;       BR      160$            ;
3990  014136  000137  014526                  JMP     160$            ;
3991  014142                          10$:
3992                                  ;FORCE SECTOR COMPARE USING SUBROUTINE
3993  014142  004737  024232                  JSR     PC,SCTCMP
3994  014146  000402                          BR      20$             ;GO TO 20$ IF NO ERROR
3995  014150  104000                          ERROR                   ;RETURN HERE IF ERROR
3996  014152  000565                          BR      160$            ;
3997  014154                          20$:
3998
3999                                  ;;*****************************************************************
4000                                  ;WRITE PROM OF DATA TIMING SEQUENCER SHOULD NOW BE ENABLED
4001                                  ;FIRST, VERIFY THAT WRITE GATE IS ON, INDICATING THAT
4002                                  ;SECTOR COMPARE SET
4003
4004
4005  014154  016037  000040  001142          MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
4006  014162  042737  177776  001142          BIC     #^CBB00,$BDDAT          ;BUS BIT 0 IS WRITE GATE
4007  014170  001011                          BNE     30$             ;
4008  014172  005037  001140                  CLR     $GDDAT
4009  014176  010037  001136                  MOV     R0,$BDADR
4010  014202  062737  000040  001136          ADD     #RMMR2,$BDADR
4011  014210  104042                          ERROR   42                      ;WRITE GATE NOT ON  DURING
4012  014212  000545                          BR      160$            ;FORMAT-CS FAILURE
4013  014214                          30$:
4014                                  ;STEP THE DATA TIMING SEQUENCER TO LOCATION 16 AND VERIFY WRITE
4015                                  ;GATE STAYS ON
4016  014214  012702  000016                  MOV     #14.,R2                 ;MAXMUM NUMBER OF CLOCK
4017  014220                          40$:
4018
4019  014220  016037  000040  001142          MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
4020  014226  032737  000001  001142          BIT     #BB00,$BDDAT            ;
4021  014234  001014                          BNE     50$             ;BRANCH IF WRITE GATE ON
4022  014236  005302                          DEC     R2
4023  014240  001367                          BNE     40$
4024  014242  010037  001136                  MOV     R0,$BDADR
4025  014246  062737  000040  001136          ADD     #RMMR2,$BDADR
4026  014254  012737  000001  001140          MOV     #BB00,$GDDAT
4027  014262  104042                          ERROR   42                      ;WRITE GATE DROP DURING
4028  014264  000520                          BR      160$            ;FORMAT - CS FAILURE
4029  014266                          50$:
4030  014266                          60$:
4031
4032  014266  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4033
```

```
 4034  014274  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4035
 4036  014302  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
 4037  014306  032703  000040                  BIT     #WC,R3
 4038  014312  001765                          BEQ     60$
 4039                                  ;PROM STROBE CAME ON-DECREMENT PROM STROBE COUNT
 4040  014314  005302                          DEC     R2
 4041  014316  001414                          BEQ     80$
 4042                                  ;WAIT FOR PROM STROBE TO GO OFF, THEN REPEAT LOOP
 4043  014320                          70$:
 4044
 4045  014320  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 4046
 4047  014326  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4048
 4049  014334  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
 4050  014340  032703  000040                  BIT     #WC,R3
 4051  014344  001725                          BEQ     40$
 4052  014346  000764                          BR      70$
 4053  014350                          80$:
 4054
 4055                                  ;;*****************************************************************
 4056                                  ;VERIFY HEADER SYNC GENERATION
 4057                                  ;WRITE DATA BIT IS INVERTED AT MAINTENANCE REGISTER
 4058                                  ;FIRST,COUNT NUMBER OF ZERO BITS UNTIL FIRST ONE BIT
 4059  014350  012702  000020                  MOV     #16.,R2         ;MAX TIMES THRU LOOP
 4060  014354  005003                          CLR     R3
 4061  014356                          90$:
 4062
 4063  014356  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
 4064  014362  032704  000010                  BIT     #MWD,R4
 4065  014366  001414                          BEQ     110$            ;JUMP OUT OF LOOP WITH FIRST 1
 4066  014370  005203                          INC     R3              ;INCREMENT ZERO COUNT
 4067  014372  005302                          DEC     R2              ;DECREMENT MAX LOOP COUNT
 4068  014374  001002                          BNE     100$
 4069  014376  104043                          ERROR   43
 4070  014400  000452                          BR      160$            ;HEADER SYNC ,CANT GET FIRST 1
 4071  014402                          100$:
 4072
 4073  014402  012760  055401  000024          MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
 4074
 4075  014410  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4076  014416  000757                          BR      90$
 4077  014420                          110$:
 4078                                  ;MAKE SURE THERE WERE AT LEAST 8 ZERO BITS IN HEADER SYNC
 4079  014420  020327  000010                  CMP     R3,#8.
 4080  014424  103002                          BHIS    120$
 4081  014426  104043                          ERROR   43
 4082  014430  000436                          BR      160$            ;HEADER SYNC
 4083  014432                          120$:
 4084                                  ;SAMPLE AND STORE THE REST OF THE HEADER SYNC
 4085  014432  012702  000010                  MOV     #8.,R2          ;NUMBER OF SAMPLES
 4086  014436  005003                          CLR     R3              ;HEADER SYNC
 4087  014440                          130$:
 4088
 4089  014440  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
```

```
4090  014444  006303                          ASL     R3
4091  014446  032704  000010                  BIT     #MWD,R4         ;SHIFT SAMPLE,IF SYNC BIT IS 1
4092  014452  001002                          BNE     140$            ;THEN SET SAMPLE INPUT
4093  014454  052703  000001                  BIS     #BIT0,R3
4094  014460  005302                  140$:   DEC     R2
4095  014462  001407                          BEQ     150$
4096
4097  014464  012760  055401  000024          MOV     #MR1AAA.MCLK,RMMR*(R0)  ;LOAD RMMR1
4098
4099  014472  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4100  014500  000757                          BR      130$            ;DO LOOP TIL R2-0
4101  014502                          150$:
4102                                  ;VERIFY THE SYNC BIT STREAM IS 0000000010011000
4103  014502  012737  000230  001140          MOV     #000230,$GDDAT
4104  014510  010337  001142                  MOV     R3,$BDDAT
4105  014514  023737  001140  001142          CMP     $GDDAT,$BDDAT
4106  014522  001401                          BEQ     160$
4107  014524  104044                          ERROR   44                      ;INCORRECT HEADER SYNC
4108  014526                          160$:
4109                                  ;;*****************************************************************
4110                                  ;*TEST 21        WRITE HEADER TEST
4111
4112                                  ;;*****************************************************************
4113  014526  000004                  TST21:  SCOPE
4114  014530  012737  000021  001226          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
4115  014536  000240                          NOP
4116  014540  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
4117  014546  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
4118  014554  012737  014570  001122          MOV     #T21,$LPADR  ;LOAD LOOP ON TEST ADDRESS
4119  014562  012737  014570  001124          MOV     #T21,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
4120  014570                          T21:
4121  014570  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
4122  014574  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
4123  014600  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
4124  014604  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
4125  014612  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
4126                                  ;SETUP THE REGISTER OUTPUT BUFFER FOR SUBROUTINES
4127  014616  012737  002037  001410          MOV     #002037,RMDA0   ;TRACK 4, SECTOR 31
4128  014624  012737  001466  001436          MOV     #1466,RMDC0     ;CYLINDER 822.
4129  014632  012737  010000  001434          MOV     #FMT16,RMOF0    ;IN 16 BIT FORMAT
4130  014640  012702  054116                  MOV     #BUFFER,R2      ;BUFFER ADDRESS
4131  014644  010237  001406                  MOV     R2,RMBA0        ;BUFFER STARTS
4132  014650  012722  151466                  MOV     #151466,(R2)+   ;STORE FIRST HEADER WORD
4133  014654  012712  002037                  MOV     #002037,(R2)    ;STORE SECOND HEADER WORD
4134  014660  012737  177776  001404          MOV     #^C2+1,RMWC0    ;WORD CTR - 2
4135  014666  012737  000063  001402          MOV     #WH!GO,RMCS10   ;WRITE HEADER COMMAND
4136
4137                                  ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
4138  014674  004737  023352                  JSR     PC,ENBSCH
4139  014700  000403                          BR      10$             ;GO TO 10$ IF NO ERROR
4140  014702  104000                          ERROR                   ;RETURN HERE IF ERROR
4141  014704  000137  015506                  JMP     160$            ;
4142  014710                          10$:
4143                                  ;FORCE SECTOR COMPARE USING SUBROUTINE
4144  014710  004737  024232                  JSR     PC,SCTCMP
4145  014714  000403                          BR      20$             ;GO TO 20$ IF NO ERROR
```

```
4146  014716  104000                        ERROR                   ;RETURN HERE IF ERROR
4147  014720  000137  015506                JMP     160$
4148  014724                        20$:
4149                                ;;*****************************************************************
4150                                ;STEP THE DATA TIMING SEQUENCER UNTIL 'HEADER AREA'' COMES ON
4151  014724  012702  000017                MOV     #15.,R2         ;ALLOW 15 MORE PROM STORBES
4152  014730  012704  000041        30$:    MOV     #33.,R4         ;ALLOW 33 BIT CLOCKS PER PROM STROBE
4153
4154                                ;PULSE BIT CLOCK UNTIL PROM STROBE IS ON
4155  014734                        40$:
4156
4157  014734  012760  055401  000024        MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4158
4159  014742  012760  051401  000024        MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4160
4161  014750  016003  000024                MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4162  014754  032703  000040                BIT     #WC,R3
4163  014760  001004                        BNE     50$
4164  014762  005304                        DEC     R4
4165  014764  001363                        BNE     40$
4166  014766  000137  015506                JMP     160$            ;COUNT EXHAUSTED
4167  014772                        50$:
4168
4169                                ;SEE IF HEADER AREA CAME ON
4170  014772  032703  000200                BIT     #PHA,R3
4171  014776  001062                        BNE     80$
4172  015000  005302                        DEC     R2
4173  015002  001015                        BNE     60$
4174                                ;ERROR-HEADER AREA NEVER CAME ON
4175  015004  012737  000200  001140        MOV     #PHA,$GDDAT     ;SETUP ERROR MESSAGE
4176  015012  005037  001142                CLR     $BDDAT
4177  015016  010037  001136                MOV     R0,$BDADR
4178  015022  062737  000024  001136        ADD     #RMMR1,$BDADR
4179  015030  104045                        ERROR   45                      ;HEADER AREA WONT SET
4180  015032  000137  015506                JMP     160$            ;SKIP REST OF TEST
4181  015036                        60$:
4182                                ;WAIT FOR PROM STROBE TO GO OFF
4183
4184  015036  012760  055401  000024        MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
4185
4186  015044  012760  051401  000024        MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4187
4188  015052  016003  000024                MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4189  015056  032703  000040                BIT     #WC,R3
4190  015062  001404                        BEQ     70$
4191  015064  005304                        DEC     R4
4192  015066  001363                        BNE     60$
4193  015070  000137  015506                JMP     160$            ;COUNT EXHAUSTED
4194  015074                        70$:
4195                                ;VERIFY THE TAG BUS ONCE EVERY PROM STROBE
4196
4197  015074  016037  000040  001142        MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
4198  015102  042737  176000  001142        BIC     #^C1777,$BDDAT
4199  015110  022737  000001  001142        CMP     #BB00,$BDDAT    ;WRITE GATE SHOULD BE ON
4200  015116  001704                        BEQ     30$             ;ALL ELSE OFF
4201  015120  010037  001136                MOV     R0,$BDADR       ;
```

```
4202  015124  062737  000040  001136          ADD     #RMMR2,$BDADR
4203  015132  012737  000001  001140          MOV     #BB00,$GDDAT
4204  015140  104042                          ERROR   42                      ;TAG BUS WRONG DURING FORMAT
4205  015142  000561                          BR      160$
4206  015144                          80$:
4207                                  ;HEADER AREA IS NOW ON CAN START SAMPLE
4208                                  ;DATA AFTER 5 MORE BIT CLOCKS
4209  015144  012702  000005                  MOV     #5,R2
4210  015150                          81$:
4211
4212  015150  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4213
4214  015156  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4215  015164  005302                          DEC     R2
4216  015166  001370                          BNE     81$
4217                                  ;;*****************************************************************
4218                                  ;SAMPLE AND STORE THE TWO HEADER WORDS
4219  015170  005037  001174                  CLR     $TMP0
4220  015174  005037  001176                  CLR     $TMP1
4221  015200  012702  001174                  MOV     #$TMP0,R2       ;R2 NUMBER OF WORDS/ADDRESS
4222  015204  012703  000020          90$:    MOV     #16.,R3         ;NUMBER OF BITS
4223  015210  005004                          CLR     R4              ;WORD STORAGE
4224  015212                          100$:
4225
4226  015212  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
4227  015216  006304                          ASL     R4              ;SHIFT WORD SAMPLE
4228  015220  032705  000010                  BIT     #MWD,R5         ;SET BIT 0 IF WRITE BIT ON
4229  015224  001002                          BNE     110$
4230  015226  052704  000001                  BIS     #BIT0,R4
4231  015232                          110$:
4232
4233  015232  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4234
4235  015240  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4236  015246  005303                          DEC     R3
4237  015250  001360                          BNE     100$
4238  015252  010422                          MOV     R4,(R2)+        ;STORE WORD SAMPLE
4239  015254  022702  001200                  CMP     #$TMP0+4,R2     ;DONE ?
4240  015260  101415                          BLOS    120$            ;YES
4241                                  ;HEADER AREA  SHOULD STILL BE ON FOR SECOND HEADER WORD
4242  015262  042705  177577                  BIC     #^C<PHA>,R5
4243  015266  001346                          BNE     90$             ;BRANCH IF ON
4244  015270  010037  001136                  MOV     R0,$BDADR
4245  015274  062737  000024  001136          ADD     #RMMR1,$BDADR
4246  015302  012737  000200  001140          MOV     #PHA,$GDDAT
4247  015310  104045                          ERROR   45
4248  015312  000475                          BR      160$
4249  015314                          120$:
4250                                  ;;*****************************************************************
4251                                  ;WAIT UNTIL ECRC SETS
4252  015314  005003                          CLR     R3              ;CHECK NUMBER BITS DIFFER BET PHA AND ECRC
4253
4254  015316                          125$:
4255
4256  015316  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
4257  015322  042705  176777                  BIC     #^C<ECRC>,R5    ;CHECK OUT THE REST BITS
```

```
4258  015326  001026                          BNE     127$            ;BRANCH IF ECRC ON
4259
4260  015330  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4261
4262  015336  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4263  015344  005203                          INC     R3              ;TOTAL NUMBER OF BIT DIFF
4264  015346  022703  000020                  CMP     #16.,R3         ;OVER ONE WORD ?
4265  015352  101361                          BHI     125$            ;BRANCH IF NOT
4266  015354  010037  001136                  MOV     R0,$BDADR
4267  015360  062737  000024  001136          ADD     #RMMR1,$BDADR
4268  015366  012737  001000  001140          MOV     #ECRC,$GDDAT
4269  015374  005037  001142                  CLR     $BDDAT          ;CLEAR THE EXP
4270  015400  104046                          ERROR   46
4271  015402  000441                          BR      160$            ;EXIT
4272  015404                          127$:
4273                                  ;;**********************************************************************
4274                                  ;SAMPLE AND STORE CRC WORD
4275  015404  012703  000020                  MOV     #16.,R3
4276  015410  005004                          CLR     R4
4277  015412                          130$:
4278
4279  015412  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
4280  015416  006304                          ASL     R4              ;SHIFT WORD SAMPLE
4281  015420  032705  000010                  BIT     #MWD,R5         ;SET BIT 0 IF WRITE BIT ON
4282  015424  001002                          BNE     140$
4283  015426  052704  000001                  BIS     #BIT0,R4
4284  015432                          140$:
4285
4286  015432  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4287
4288  015440  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4289  015446  005303                          DEC     R3
4290  015450  001360                          BNE     130$
4291  015452  010422                          MOV     R4,(R2)+
4292                                  ;;**********************************************************************
4293                                  ;VERIFY 3 HEADER WORDS
4294                                  ;NOTE THAT DATA WAS STORED IN THE REVERSE
4295                                  ;ORDER FROM WHICH IT WAS WRITTEN
4296  015454  022737  066313  001174          CMP     #066313,$TMP0
4297  015462  001010                          BNE     150$            ;BRANCH IF ERROR
4298  015464  022737  174040  001176          CMP     #174040,$TMP1
4299  015472  001004                          BNE     150$            ;BRANCH IF ERROR
4300  015474  022737  167501  001200          CMP     #167501,$TMP2   ;
4301  015502  001401                          BEQ     160$            ;BRANCH IF NO ERROR
4302  015504                          150$:
4303  015504  104047                          ERROR   47
4304  015506                          160$:                           ;END OF SUB TEST
4305                                  ;;**********************************************************************
4306                                  ;*TEST 22        HEADER COMPARE TEST
4307
4308                                  ;;**********************************************************************
4309  015506  000004                  TST22:  SCOPE
4310  015510  012737  000022  001226          MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
4311  015516  000240                          NOP
4312  015520  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
4313  015526  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
```

```
                        T22     HEADER COMPARE TEST

 4314  015534  012737  015550  001122         MOV     #T22,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 4315  015542  012737  015550  001124         MOV     #T22,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 4316  015550                         T22:
 4317  015550  012706  001100                MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4318  015554  013700  001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
 4319  015560  013701  001456                MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
 4320
 4321                                 ;LOAD REGISTER OUTPUT BUFFER FOR READ HEADER COMMAND
 4322
 4323  015564  012737  001466  001436        MOV     #001466,RMDC0           ;CYL=822.
 4324  015572  012737  002037  001410        MOV     #2037,RMDA0             ;TRK=4,SEC=31.
 4325  015600  012737  054116  001406        MOV     #BUFFER,RMBA0           ;DATA ADDRESS
 4326  015606  012737  177776  001404        MOV     #^C2+1,RMWC0            ;READ HEADER ONLY
 4327  015614  012737  010000  001434        MOV     #FMT16,RMOF0            ;16 BIT MODE
 4328  015622  012737  000073  001402        MOV     #RH!GO,RMCS10           ;READ HEADER AND DATA FU'
 4329
 4330                                 ;CLEAR THE MASSBUS
 4331  015630  012760  000040  000010        MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
 4332  015636  111160  000010                MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
 4333
 4334                                 ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
 4335  015642  004737  023352                JSR     PC,ENBSCH
 4336  015646  000403                        BR      10$             ;GO TO 10$ IF NO ERROR
 4337  015650  104000                        ERROR                   ;RETURN HERE IF ERROR
 4338  015652  000137  016756                JMP     230$            ;SKIP REST OF TEST IF ERROR
 4339  015656                         10$:
 4340                                 ;FORCE SECTOR COMPARE USING SUBROUTINE
 4341  015656  004737  024232                JSR     PC,SCTCMP
 4342  015662  000403                        BR      20$             ;GO TO 20$ IF NO ERROR
 4343  015664  104000                        ERROR                   ;RETURN HERE IF ERROR
 4344  015666  000137  016756                JMP     230$
 4345  015672                         20$:
 4346                                 ;READ GATE SHOULD BE OFF FOR 3 PROM STROBES
 4347
 4348  015672  010037  001136                MOV     R0,$BDADR               ;SETUP ERROR MESSAGE
 4349  015676  062737  000040  001136        ADD     #RMMR2,$BDADR
 4350  015704  012702  000004                MOV     #4,R2                   ;R2=NUMBER OF PROM STROBES
 4351  015710  012704  000021         30$:   MOV     #17.,R4         ;MAX BIT CLOCKS
 4352
 4353  015714                         40$:
 4354
 4355                                 ;CLOCK BIT CLOCK UNTIL PROM STROBE COMES ON
 4356
 4357
 4358  015714  012760  055401  000024        MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 4359
 4360  015722  012760  051401  000024        MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4361
 4362  015730  016003  000024                MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
 4363  015734  032703  000040                BIT     #WC,R3
 4364  015740  001004                        BNE     50$             ;BRANCH WHEN PROM STROBE ON
 4365  015742  005304                        DEC     R4
 4366  015744  001363                        BNE     40$
 4367  015746  000137  016756                JMP     230$            ;***********
 4368  015752                         50$:
 4369
```

```
                                        ;VERIFY READ GATE IS OFF AND TAG BUS IS ZERO
                                
016752  016003  000040                  MOV     RMMR2(R0),R3    ;STORE RMMR2 AT R3
016756  042703  176000                  BIC     #^C1777,R3
016762  001407                          BEQ     60$             ;BRANCH IF TAG BUS ZERO
016764  010337  001142                  MOV     R3,$BDDAT
016770  005037  001140                  CLR     $GDDAT
016774  104050                          ERROR   50
016776  000137  016756                  JMP     230$            ;READ HEADER FUNCTION
016002                          60$:
                                        ;CLOCK BIT CLOCK TIL PROM STROBE GOES OFF
                                
016002  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
                                
016010  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
                                
016016  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
016022  032703  000040                  BIT     #WC,R3
016026  001404                          BEQ     70$             ;BRANCH IF PROM STROBE OFF
016030  005304                          DEC     R4
016032  001363                          BNE     60$
016034  000137  016756                  JMP     230$            ;***************
016040                          70$:
                                        ;CONTINUE FOR 3 BIT CYCLE TOTAL (LOC 0 NOT SAMPLED)
                                
016040  005302                          DEC     R2
016042  001322                          BNE     30$
                                
                                        ;;*************************************************************************
                                        ;READ GATE SHOULD COME ON WITH NEXT PROM STROBE AND STAY
                                        ;ON FOR 7 CYCLES (AND MORE)
                                
016044  012702  000006                  MOV     #6,R2
016050  012703  000021          80$:    MOV     #17.,R3
                                        ;CLOCK BIT CLOCK TIL PROM STROBE SETS
                                
016054                          90$:
                                
016054  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
                                
016062  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
                                
016070  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
016074  032704  000040                  BIT     #WC,R4
016100  001004                          BNE     100$            ;BRANCH WHEN PROM STROBE ON
016102  005303                          DEC     R3
016104  001363                          BNE     90$
016106  000137  016756                  JMP     230$            ;************
016112                          100$:
                                        ;VERIFY READ GATE ON AND REST OF TAG BUS OFF
                                
016112  016004  000040                  MOV     RMMR2(R0),R4    ;STORE RMMR2 AT R4
016116  042704  176000                  BIC     #^C1777,R4
016122  022704  000002                  CMP     #BB01,R4
```

```
4426  016126  001410                          BEQ     110$            ;BRANCH IF READ GATE ON
4427  016130  010437  001142                  MOV     R4,$BDDAT
4428  016134  012737  000002  001140          MOV     #BB01,$GDDAT
4429  016142  104050                          ERROR   50                      ;TAG BUS INCORRECT DURING
4430  016144  000137  016756                  JMP     230$            ;READ HEADER FUNCTION
4431
4432  016150                          110$:
4433                                  ;CLOCK BIT CLOCK TIL PROM STROBE GOES OFF
4434
4435
4436  016150  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4437
4438  016156  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4439
4440  016164  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
4441  016170  032704  000040                  BIT     #WC,R4
4442  016174  001404                          BEQ     120$            ;BRANCH WHEN PROM STROBE OFF
4443  016176  005303                          DEC     R3
4444  016200  001363                          BNE     110$
4445  016202  000137  016756                  JMP     230$
4446
4447  016206                          120$:
4448                                  ;CONTINUE FOR TOATL OF 7 CYCLES
4449  016206  005302                          DEC     R2
4450  016210  001317                          BNE     80$     ;*&**&&
4451                                  ;;*****************************************************************
4452                                  ;LOOKING FOR SYNC SHOULD SET WITH NEXT PROM STORBE
4453  016212  012702  000021                  MOV     #17.,R2         ;MAX BIT CLOCKS
4454  016216                          130$:
4455                                  ;CLOCK BIT CLOCK TIL PROM STROBE SETS
4456
4457
4458  016216  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4459
4460  016224  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4461
4462  016232  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4463  016236  032703  000040                  BIT     #WC,R3
4464  016242  001004                          BNE     140$            ;BRANCH WHEN PROM STROBE ON
4465  016244  005302                          DEC     R2
4466  016246  001363                          BNE     130$
4467  016250  000137  016756                  JMP     230$
4468  016254                          140$:
4469                                  ;VERIFY 'PLFS'' IS NOW ON
4470  016254  042703  175777                  BIC     #^CPLFS,R3
4471  016260  001015                          BNE     150$
4472  016262  010337  001142                  MOV     R3,$BDDAT
4473  016266  012737  002000  001140          MOV     #PLFS,$GDDAT
4474  016274  010037  001136                  MOV     R0,$BDADR
4475  016300  062737  000024  001136          ADD     #RMMR1,$BDADR
4476  016306  104035                          ERROR   35
4477  016310  000137  016756                  JMP     230$
4478  016314                          150$:
4479  016314  012703  000005                  MOV     #5,R3
4480                                  ;WITH PLFS ON, WORD COUNT INHIBIT WILL SET IN 5 BIT CLOCKS
4481  016320                          155$:
```

```
4482
4483  016320  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4484
4485  016326  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4486  016334  005303                          DEC     R3
4487  016336  001370                          BNE     155$
4488
4489                                  ;;*****************************************************************
4490                                  ;SIMULATE THE SYNC PATTERN BEING READ
4491  016340  012702  000031                  MOV     #31,R2          ;SYNC PATTERN=00011001
4492  016344  012703  000010                  MOV     #8.,R3          ;8 BITS IN PATTERN
4493  016350                          160$:
4494  016350  012737  055401  001426          MOV     #MR1AAA!MCLK,RMMR10
4495  016356  000241                          CLC
4496  016360  006002                          ROR     R2
4497  016362  103003                          BCC     165$
4498  016364  052737  002000  001426          BIS     #MRD,RMMR10
4499  016372                          165$:
4500
4501  016372  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
4502  016400  042737  004000  001426          BIC     #MCLK,RMMR10
4503
4504  016406  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
4505  016414  005303                          DEC     R3              ;CONTINUE TIL SHIFT COUNT
4506  016416  001354                          BNE     160$            ;IS ZERO
4507                                  ;;*****************************************************************
4508                                  ;SIMULATE THE HEADER BEING READ
4509  016420  012702  001174                  MOV     #$TMP0,R2
4510  016424  012737  151466  001174          MOV     #151466,$TMP0
4511  016432  012737  002037  001176          MOV     #2037,$TMP1     ;**************
4512  016440  012737  101367  001200          MOV     #101367,$TMP2           ;CRC PATTERN ***
4513  016446                          170$:
4514  016446  012703  000020                  MOV     #16.,R3         ;NUMBER OF BITS EACH WORD
4515  016452  012204                          MOV     (R2)+,R4        ;HEADER WORD 1,2 OR 3
4516  016454                          175$:
4517  016454  012737  055401  001426          MOV     #MR1AAA!MCLK,RMMR10
4518  016462  000241                          CLC
4519  016464  006004                          ROR     R4
4520  016466  103003                          BCC     180$
4521  016470  052737  002000  001426          BIS     #MRD,RMMR10
4522  016476                          180$:
4523
4524  016476  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
4525  016504  042737  004000  001426          BIC     #MCLK,RMMR10
4526
4527  016512  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
4528  016520  005303                          DEC     R3              ;SHIFT OUT 16 BITS
4529  016522  001354                          BNE     175$
4530  016524  020227  001200                  CMP     R2,#$TMP0+4     ;ALL DONE ?
4531  016530  101746                          BLOS    170$            ;BRANCH IF NOT
4532                                  ;;*****************************************************  ********
4533                                  ;LOOKING FOR SYNC SHOULD COME ON WITHIN 6 STROBES
4534  016532  012702  000006                  MOV     #6,R2
4535
4536  016536  012703  000021          185$:   MOV     #17.,R3
4537                                  ;CLOCK UNTIL PROM STROBE ON
```

```
4538  016542                                  190$:
4539
4540  016542  012760  055401  000024                MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
4541
4542  016550  012760  051401  000024                MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4543
4544  016556  016004  000024                      MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
4545  016562  032704  000040                      BIT     #WC,R4
4546  016566  001003                              BNE     195$
4547  016570  005303                              DEC     R3
4548  016572  001363                              BNE     190$
4549  016574  000470                              BR      230$
4550  016576                                  195$:
4551                                          ;SEE IF "PLFS" IS ON
4552  016576  042704  175777                      BIC     #^CPLFS,R4
4553  016602  001034                              BNE     210$
4554                                          ;CONTINUE IF LESS THAN 6 PROM STROBES
4555  016604  005302                              DEC     R2
4556  016606  001014                              BNE     200$
4557  016610  010437  001142                      MOV     R4,$BDDAT       ;SETUP ERROR
4558  016614  012737  002000  001140              MOV     #PLFS,$GDDAT
4559  016622  010037  001136                      MOV     R0,$BDADR
4560  016626  062737  000024  001136              ADD     #RMMR1,$BDADR
4561  016634  104035                              ERROR   35
4562  016636  000447                              BR      230$            ;HEADER
4563  016640                                  200$:
4564                                          ;CLOCK UNTIL PROM STROBE OFF
4565
4566  016640  012760  055401  000024                MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
4567
4568  016646  012760  051401  000024                MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4569
4570  016654  016004  000024                      MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
4571  016660  032704  000040                      BIT     #WC,R4
4572  016664  001724                              BEQ     185$
4573  016666  005303                              DEC     R3
4574  016670  001363                              BNE     200$
4575  016672  000431                              BR      230$
4576  016674                                  210$:
4577                                          ;VERIFY NO HEADER ERROR IS SET
4578
4579  016674  016037  000014  001142              MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
4580  016702  042737  177157  001142              BIC     #^C<HCRC.HCE!FER>,$BDDAT
4581  016710  001411                              BEQ     220$
4582  016712  005037  001140                      CLR     $GDDAT
4583  016716  010037  001136                      MOV     R0,$BDADR
4584  016722  062737  000014  001136              ADD     #RMER1,$BDADR
4585  016730  104051                              ERROR   51
4586  016732  000411                              BR      230$            ;HEADER ERROR SET
4587  016734                                  220$:
4588                                          ;VERIFY DATA IN MEMORY OK
4589  016734  023737  054116  001174              CMP     BUFFER,$TMP0    ;COMPARE 1 ST HEADER WORD
4590  016742  001004                              BNE     225$
4591  016744  023737  054120  001176              CMP     BUFFER+2,$TMP1  ;COMPARE 2 ND HEADER WORD
4592  016752  001401                              BEQ     230$
4593  016754                                  225$:
```

```
                                        ;REPORT ERROR IN ONE OR MORE HEADER WORDS
016754  104052                                ERROR   52
016756                                  230$:
                                        ;;*****************************************************************
                                        ;*TEST 23        ECC GENERATION TEST
                                        ;;*****************************************************************
016756  000004                          TST23:  SCOPE
016760  012737  000023  001226                  MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
016766  000240                                  NOP
016770  012737  000024  001120                  MOV     #20.,$ICNT      ;20 ITERATIONS
016776  112737  000001  001131                  MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
017004  012737  017020  001122                  MOV     #T23,$LPADR  ;LOAD LOOP ON TEST ADDRESS
017012  012737  017020  001124                  MOV     #T23,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
017020                                  T23:
017020  012706  001100                          MOV     #STACK,SP       ;LOAD THE STACK POINTER
017024  013700  001276                          MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
017030  013701  001456                          MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
017034  012760  000040  000010                  MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
017042  111160  000010                          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
                                        ;SETUP REGISTER OUTPUT BUFFER FOR WRITE DATA CAMOMMAND
017046  012737  002037  001410                  MOV     #002037,RMDA0   ;TRK 4, SEC 31
017054  012737  001466  001436                  MOV     #1466,RMDC0     ;CYL=822.
017062  012737  012000  001434                  MOV     #FMT16!HCI,RMOF0        ;16 BIT FORMAT,IGNORE HEADER
017070  012702  054116                          MOV     #BUFFER,R2
017074  010237  001406                          MOV     R2,RMBA0                ;DATA ADDRESS
017100  012703  177400                          MOV     #^C256.+1,R3
017104  010337  001404                          MOV     R3,RMWC0
017110  012737  000061  001402                  MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND ***
                                        ;FILL THE DATA BUFFER WITH ALL ONES
017116  012704  177777                          MOV     #177777,R4              ;DATA PATTERN ALL ONES
017122  012703  000400                          MOV     #400,R3         ;ONE SECTOR DATA FIELD
017126  010422                          10$:    MOV     R4,(R2)+
017130  005303                                  DEC     R3
017132  001375                                  BNE     10$

                                        ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
017134  004737  023352                          JSR     PC,ENBSCH
017140  000403                                  BR      20$             ;GO TO 20$ IF NO ERROR
017142  104000                                  ERROR                   ;RETURN HERE IF ERROR
017144  000137  017742                          JMP     220$
017150                                  20$:
                                        ;FORCE SECTOR COMPARE USING SUBROUTINE
017150  004737  024232                          JSR     PC,SCTCMP
017154  000403                                  BR      30$             ;GO TO 30$ IF NO ERROR
017156  104000                                  ERROR                   ;RETURN HERE IF ERROR
017160  000137  017742                          JMP     220$
017164                                  30$:
                                        ;SET 'LOOKING FOR SYNC' USING SUBROUTINE
017164  004737  024340                          JSR     PC,SETLFS
017170  000403                                  BR      40$
017172  104000                                  ERROR
017174  000137  017742                          JMP     220$
017200                                  40$:
                                        ;CLOCK THE SYNC PATTERN USING SUBROUTINE
017200  004737  024522                          JSR     PC,CLKSNC
```

```
4650  017204  000403                          BR      50$
4651  017206  104000                          ERROR
4652  017210  000137  017742                  JMP     220$
4653
4654                                  ;HEADER COMPARE IS INHIBITED FOR THIS TEST
4655                                  ;CLOCK THE DATA TIMING SEQUENCER AND VERIFY THE FOLLOWING WAVEFORMS
4656                                  ;FOR EACH LEADING EDGD OF PROM STROBE
4657                                  ;       HEADER ARES, READ GATE (DATA TIMING SEQUENCER 200 OCTAL)
4658                                  ;       ENABLE CRC OUT
4659                                  ;       WRITE GATE
4660
4661  017214  012702  017746          50$:    MOV     #230$,R2        ;****** POINTER TO WAVE FORM TABLE
4662
4663  017220                          70$:
4664                                  ;CLOCK BIT CLOCK (MCLK) UNTIL PROM STROBE COMES ON
4665  017220  004737  017766                  JSR     PC,300$
4666  017224                          80$:
4667                                  ;PROM STROBE IS ON - VERIFY WAVEFORM
4668
4669  017224  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
4670  017232  042737  176157  001142          BIC     #^C<ECRC!PDA!PHA.EECC>,$BDDAT
4671  017240  012237  001140                  MOV     (R2)+,$GDDAT
4672  017244  023737  001140  001142          CMP     $GDDAT,$BDDAT
4673  017252  001410                          BEQ     90$             ;BRANCH IF RMMR1 OK
4674                                  ;ERROR - THE DATA TIMING SEQUENCER OUTPUT iS ONCORRECT
4675  017254  010037  001136                  MOV     R0,$BDADR
4676  017260  062737  000024  001136          ADD     #RMMR1,$BDADR
4677  017266  104054                          ERROR   54                      ;DATA TIMING SEQUENCER WRONG
4678  017270  000137  017742                  JMP     220$            ;
4679  017274                          90$:
4680                                  ;VERIFY THE TAB BUS
4681
4682  017274  016037  000040  001142          MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
4683  017302  042737  176000  001142          BIC     #^C1777,$BDDAT
4684  017310  012237  001140                  MOV     (R2)+,$GDDAT
4685  017314  023737  001140  001142          CMP     $GDDAT,$BDDAT
4686  017322  001410                          BEQ     100$            ;BRANCH IF TAG BUS OK
4687                                  ;ERROR- TAG BUS IS WRONG
4688  017324  010037  001136                  MOV     R0,$BDADR
4689  017330  062737  000040  001136          ADD     #RMMR2,$BDADR
4690  017336  104053                          ERROR   53                      ;TAG BUS WRONG
4691  017340  000137  017742                  JMP     220$
4692  017344                          100$:
4693                                  ;CLOCK BIT CLOCK TIL PROM STROBE RESETS
4694  017344  004737  020074                  JSR     PC,350$
4695  017350                          110$:
4696                                  ;CONTINUE CHECKING THROUGH 4 PROM CYCLES
4697  017350  020227  017764                  CMP     R2,#230$+16     ;********
4698  017354  103721                          BLO     70$
4699                                  ;;*****************************************************************
4700                                  ;VERIFY THAT DATA AREA COMES ON WITHIN  9 PROM STROBES
4701  017356  012702  000011                  MOV     #9.,R2          ;R2=9 STROBES
4702  017362                          130$:
4703                                  ;CLOCK PROM STROBE ON
4704  017362  004737  017766                  JSR     PC,300$
4705  017366                          140$:
```

```
                                        ;PROM STROBE ON CKECK IF DATA AREA ON
                                                MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
                                                BIC     #^CPDA,R4
                                                BNE     160$
                                        ;CLOCK PROM STROBE OFF
                                        150$:
                                                JSR     PC,350$
                                                DEC     R2
                                                BNE     130$
                                        ;ERROR-DATA AREA DIDN'T COME ON
                                                MOV     #PDA,$GDDAT     ;SETUP ERROR MESSAGE
                                                CLR     $BDDAT
                                                MOV     R0,$BDADR
                                                ADD     #RMMR1,$BDADR
                                                ERROR   55
                                                BR      220$
                                        160$:
                                        ;;**************************************************************************
                                        ;VERIFY THAT DATA AREA IS ON FOR 256 CYCLES
                                                MOV     #256.,R2
                                        170$:
                                        ;VERIFY THAT DATA AREA IS ON AND ECC IS OFF

                                                MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
                                                BIC     #^C<PDA.EECC>,R3
                                                CMP     #PDA,R3
                                                BEQ     180$            ;DATA AREA IS ON
                                                MOV     R3,$BDDAT
                                                MOV     #PDA,$GDDAT
                                                MOV     R0,$BDADR
                                                ADD     #RMMR1,$BDADR
                                                ERROR   55                      ;DATA ERROR NOT ON
                                                BR      220$
                                        180$:
                                        ;CLOCK PROM STROBE OFF
                                                JSR     PC,350$
                                        ;CLOCK PROM STROBE ON
                                                JSR     PC,300$
                                        ;CONTINUE TIL COUNT ZERO
                                                DEC     R2
                                                BNE     170$
                                        ;ECC SHOULD  BE ENABLED WITHIN 4 CLOCK BITS

                                                CLR     R2
                                        182$:

                                                MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
                                                BIC     #^C<PDA!EECC>,R3
                                                CMP     #EECC,R3
                                                BEQ     190$
                                                INC     R2

                                                MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
```

```
 4762  017556  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4763  017564  022702  000007                  CMP     #7,R2
 4764  017570  101357                          BHI     182$
 4765  017572  010337  001142                  MOV     R3,$BDDAT
 4766  017576  012737  000020  001140          MOV     #EECC,$GDDAT
 4767  017604  010037  001136                  MOV     R0,$BDADR
 4768  017610  062737  000024  001136          ADD     #RMMR1,$BDADR
 4769  017616  104056                          ERROR   56
 4770  017620  000450                          BR      220$
 4771  017622                          190$:
 4772                                  ;SAMPLE ECC CHARACTER FOR 32 BITS
 4773  017622  012702  001174                  MOV     #$TMP0,R2
 4774  017626  005003                  195$:   CLR     R3
 4775  017630  012704  000020                  MOV     #16.,R4
 4776  017634                          200$:
 4777
 4778  017634  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
 4779  017640  006303                          ASL     R3
 4780  017642  032705  000010                  BIT     #MWD,R5
 4781  017646  001002                          BNE     205$            ;BRANCH IF ECC BIT OFF (MWD INVERTED)
 4782  017650  052703  000001                  BIS     #BIT0,R3
 4783  017654                          205$:
 4784
 4785  017654  012760  055401  000024          MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
 4786
 4787  017662  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4788  017670  005304                          DEC     R4
 4789  017672  001360                          BNE     200$
 4790  017674  010322                          MOV     R3,(R2)+
 4791  017676  022702  001200                  CMP     #$TMP0+4,R2
 4792  017702  101351                          BHI     195$
 4793                          ;;*****************************************************************
 4794                          ;VERIFY ECC CHARACTER GENERATED
 4795  017704  022737  131177  001174          CMP     #131177,$TMP0   ;********
 4796  017712  001004                          BNE     210$
 4797  017714  022737  175154  001176          CMP     #175154,$TMP1   ;*****
 4798  017722  001407                          BEQ     220$
 4799  017724  012737  131177  001200  210$:   MOV     #131177,$TMP2   ;******
 4800  017732  012737  175154  001202          MOV     #175154,$TMP3   ;******
 4801  017740  104061                          ERROR   61                      ;ECC GENERATED INCORRECT
 4802  017742                          220$:
 4803  017742  000137  020200                  JMP     400$            ;GREAT ESCAPE
 4804
 4805  017746  000200                  230$:   .WORD   PHA
 4806  017750  000002                          .WORD   BB01
 4807  017752  000200                          .WORD   PHA
 4808  017754  000002                          .WORD   BB01
 4809  017756  000000                          .WORD   0
 4810  017760  000002                          .WORD   2               ;*******
 4811  017762  000000                          .WORD   0               ;******
 4812  017764  000000                          .WORD   0               ;
 4813
 4814
 4815  017766  012737  000021  020070  300$:   MOV     #17.,320$
 4816                                  ;                               CLOCK PROM STROBE ON
 4817  017774                          305$:
```

```
4818
4819  017774  012760  055401  000024          MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
4820
4821  020002  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4822
4823  020010  016037  000024  020072          MOV     RMMR1(R0),330$  ;STORE RMMR1 AT 330$
4824  020016  042737  177737  020072          BIC     #^CWC,330$
4825  020024  001020                          BNE     310$            ;EXIT IF PROM STROBE ON
4826  020026  005337  020070                  DEC     320$
4827  020032  001360                          BNE     305$
4828  020034  013737  020070  001142          MOV     320$,$BDDAT     ,SETUP ERROR MESSAGE
4829  020042  012737  000040  001140          MOV     #WC,$GDDAT
4830  020050  010037  001136                  MOV     R0,$BDADR
4831  020054  062737  000024  001136          ADD     #RMMR1,$BDADR
4832  020062  104030                          ERROR   30
4833  020064  000445                          BR      400$            ;
4834  020066  000207              310$:       RTS     PC
4835  020070  000000              320$:       .WORD   0               ;MAX BIT COUNT
4836  020072  000000              330$:       .WORD   0               ;RMMR1 CONTENTS
4837
4838  020074  012737  000021  020174  350$:   MOV     #17.,370$
4839                                                                  ;CLOCK PROM STROBE OFF
4840  020102                      355$:
4841
4842  020102  012760  055401  000024          MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
4843
4844  020110  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4845
4846  020116  016037  000024  020176          MOV     RMMR1(R0),380$  ;STORE RMMR1 AT 380$
4847  020124  042737  177737  020176          BIC     #^CWC,380$
4848  020132  001417                          BEQ     360$
4849  020134  005337  020174                  DEC     370$
4850  020140  001360                          BNE     355$
4851  020142  013737  020176  001142          MOV     380$,$BDDAT
4852  020150  005037  001140                  CLR     $GDDAT
4853  020154  010037  001136                  MOV     R0,$BDADR
4854  020160  062737  000024  001136          ADD     #RMMR1,$BDADR
4855  020166  104062                          ERROR   62                      ;CAN'T CLEAR PROM STROBE
4856  020170  000403                          BR      400$
4857  020172  000207              360$:       RTS     PC
4858
4859  020174  000000              370$:       .WORD   0               ;MAX BIT COUNT
4860  020176  000000              380$:       .WORD   0               ;CONTENTS OF RMMR1
4861
4862  020200                      400$:                               ;SUB-TEST EXIT POINT
4863  020200  000240                          NOP                     ;****
4864                              ;;*************************************************************
4865                              ;*TEST 24        ECC DETECTION TEST
4866
4867                              ;;*************************************************************
4868  020202  000004              TST24:  SCOPE
4869  020204  012737  000024  001226          MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
4870  020212  000240                          NOP
4871  020214  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
4872  020222  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
4873  020230  012737  020244  001122          MOV     #T24,$LPADR  ;LOAD LOOP ON TEST ADDRESS
```

```
4874  020236  012737  020244  001124          MOV     #T24,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
4875  020244                              T24:
4876  020244  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
4877  020250  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
4878  020254  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
4879  020260  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
4880  020266  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
4881                              ;SETUP REGISTER OUTPUT BUFFER FOR READ DATA COMMAND
4882  020272  012737  002037  001410          MOV     #2037,RMDA0
4883  020300  012737  001466  001436          MOV     #1466,RMDC0     ;CYL-822. TRK =4, SEC 31
4884  020306  012737  012000  001434          MOV     #FMT16!HCI,RMOF0        ;INHIBIT HEADER COMPARE
4885  020314  012702  054116                  MOV     #BUFONE,R2
4886  020320  010237  001406                  MOV     R2,RMBA0
4887  020324  012737  177400  001404          MOV     #^C256.+1,RMWC0         ;256 WORDS
4888  020332  012737  000071  001402          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
4889  020340                              20$:
4890                              ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
4891  020340  004737  023352                  JSR     PC,ENBSCH
4892  020344  000403                          BR      30$             ;GO TO 30$ IF NO ERROR
4893  020346  104000                          ERROR                   ;RETURN HERE IF ERROR
4894  020350  000137  021222                  JMP     190$
4895  020354                              30$:
4896                              ;FORCE SECTOR COMPARE USING SUBROUTINE
4897  020354  004737  024232                  JSR     PC,SCTCMP
4898  020360  000403                          BR      40$             ;GO TO 40$ IF NO ERROR
4899  020362  104000                          ERROR                   ;RETURN HERE IF ERROR
4900  020364  000137  021222                  JMP     190$
4901  020370                              40$:
4902                              ;SET 'LOOKING FOR SYNC'' USING SUBROUTINE
4903  020370  004737  024340                  JSR     PC,SETLFS
4904  020374  000403                          BR      50$
4905  020376  104000                          ERROR
4906  020400  000137  021222                  JMP     190$
4907  020404                              50$:
4908                              ;CLOCK THE SYNC PATTERN USING SUBROUTINE
4909  020404  004737  024522                  JSR     PC,CLKSNC
4910  020410  000403                          BR      55$
4911  020412  104000                          ERROR
4912  020414  000137  021222                  JMP     190$
4913                              ;;*****************************************************************
4914                              ;HEADER COMPARE IS INHIBITED FOR THIS TEST
4915                              ;'LOOKING FOR SYNC'' SHOULD GO OFF WITHIN ONE CYCLE
4916  020420  012702  000002      55$:    MOV     #2,R2           ;ALLOW 2 PASSES THRU LOOP
4917  020424  000405                      BR      70$
4918  020426  004737  021334      60$:    JSR     PC,350$         ;RESET PROM STROBE
4919  020432  004737  021226              JSR     PC,300$         ;SET PROM STROBE
4920  020436  000240                      NOP
4921  020440                              70$:
4922
4923  020440  016003  000024              MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4924  020444  032703  002000              BIT     #PLFS,R3
4925  020450  001421                      BEQ     80$             ;BRANCH IF LOOKING FOR SYNC OFF
4926  020452  005302                      DEC     R2
4927  020454  001364                      BNE     60$
4928                              ;ERROR-LOOKING FOR SYNC DID NOT RESET DURING HEADER
4929  020456  010337  001142              MOV     R3,$BDDAT               ;SETUP ERROR MESSAGE
```

```
 4930  020462  042737  175777  001142          BIC     #^CPLFS,$BDDAT
 4931  020470  005037  001140                  CLR     $GDDAT
 4932  020474  010037  001136                  MOV     R0,$BDADR
 4933  020500  062737  000024  001136          ADD     #RMMR1,$BDADR
 4934  020506  104040                          ERROR   40                      ;PLFS NOT RESET AGTER HEADER
 4935  020510  000137  021222                  JMP     190$
 4936  020514                              80$:
 4937                                      ;;**************************************************************************
 4938                                      ;LOOKING FOR SYNC SHOULD COME ON FOR DATA AREA WITHIN 9 PROM CYCLES
 4939  020514  012702  000011                  MOV     #9.,R2
 4940  020520  004737  021334              85$:JSR     PC,350$             ;SET STROBE OFF
 4941  020524  004737  021226                  JSR     PC,300$             ;SET PROM STROBE
 4942
 4943  020530  016003  000024                  MOV     RMMR1(R0),R3        ;STORE RMMR1 AT R3
 4944  020534  032703  002000                  BIT     #PLFS,R3            ;
 4945  020540  001022                          BNE     90$                 ;BRANCH IF SYNC ENABLED
 4946  020542  005302                          DEC     R2
 4947  020544  001365                          BNE     85$
 4948                                      ;ERROR-CAN'T LOOKING FOR SYNC DURING DATA AREA
 4949  020546  012737  002000  001140          MOV     #PLFS,$GDDAT
 4950  020554  010337  001142                  MOV     R3,$BDDAT
 4951
 4952  020560  042737  175777  001142          BIC     #^CPLFS,$BDDAT
 4953  020566  010037  001136                  MOV     R0,$BDADR
 4954  020572  062737  000024  001136          ADD     #RMMR1,$BDADR
 4955  020600  104035                          ERROR   35                      ;CAN'T SET PLFS DURING DATA
 4956  020602  000137  021222                  JMP     190$
 4957  020606                              90$:
 4958  020606  004737  021334                  JSR     PC,350$             ;RESET PROM STROBE
 4959                                      ;CLOCK THE SYNC PATTERN USING SUBROUTINE
 4960  020612  004737  024522                  JSR     PC,CLKSNC
 4961  020616  000402                          BR      100$
 4962  020620  104000                          ERROR
 4963  020622  000577                          BR      190$
 4964  020624                             100$:
 4965                                      ;;**************************************************************************
 4966                                      ;SIMULATE READ DATA
 4967  020624  012702  000400                  MOV     #256.,R2            ;WORD COUNT
 4968  020630  012703  000020             110$:MOV     #16.,R3             ;BIT COUNT
 4969  020634                             115$:
 4970
 4971  020634  012760  057401  000024          MOV     #MR1AAA.MRD.MCLK,RMMR1(R0)      ;LOAD RMMR1
 4972
 4973  020642  012760  053'01  000024          MOV     #MR1AAA.MRD,RMMR1(R0)   ;LOAD RMMR1
 4974  020650  005303                          DEC     R3
 4975  020652  001370                          BNE     115$
 4976  020654  005302                          DEC     R2
 4977  020656  001364                          BNE     110$
 4978                                      ;;**************************************************************************
 4979                                      ;SIMULATE ECC PATTERN
 4980  020660  012737  177446  001174          MOV     #177446,$TMP0           ;FIRST ECC WORD
 4981  020666  012737  015457  001176          MOV     #015457,$TMP1           ;SECOND ECC WORD
 4982  020674  012702  001174                  MOV     #$TMP0,R2
 4983  020700  012703  000020             120$:MOV     #16.,R3
 4984  020704  012704  051401             125$:MOV     #MR1AAA,R4
 4985  020710  000241                          CLC
```

```
4986  020712  006012                          ROR     (R2)
4987  020714  103002                          BCC     130$
4988  020716  052704  002000                  BIS     #MRD,R4
4989  020722                          130$:
4990
4991  020722  010460  000024                  MOV     R4,RMMR1(R0)    ;LOAD RMMR1
4992  020726  052704  004000                  BIS     #MCLK,R4
4993
4994  020732  010460  000024                  MOV     R4,RMMR1(R0)    ;LOAD RMMR1
4995  020736  042704  004000                  BIC     #MCLK,R4
4996
4997  020742  010460  000024                  MOV     R4,RMMR1(R0)    ;LOAD RMMR1
4998  020746  005303                          DEC     R3
4999  020750  001355                          BNE     125$            ;CLOCK  A WORD
5000  020752  062702  000002                  ADD     #2,R2
5001  020756  022702  001176                  CMP     #$TMP1,R2
5002  020762  001746                          BEQ     120$            ;CLOCK THE NEXT WORD
5003                          ;VERIFY DATA AREA AND READ GATE RESET
5004  020764  012702  000005                  MOV     #5,R2
5005
5006  020770  004737  021226          140$:   JSR     PC,300$         ;SET PROM STROBE
5007  020774  004737  021334                  JSR     PC,350$         ;CLEAR PROM STROBE
5008
5009  021000  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
5010  021004  032703  000400                  BIT     #PDA,R3
5011  021010  001417                          BEQ     150$
5012  021012  005302                          DEC     R2
5013  021014  001365                          BNE     140$
5014  021016  042703  177377                  BIC     #^CPDA,R3       ;SETUP ERROR MESSAGE
5015  021022  010337  001142                  MOV     R3,$BDDAT
5016  021026  005037  001140                  CLR     $GDDAT
5017  021032  010037  001136                  MOV     R0,$BDADR
5018  021036  062737  000024  001136          ADD     #RMMR1,$BDADR
5019  021044  104063                          ERROR   63                      ;DATA AREA WON'T RESET
5020  021046  000465                          BR      190$
5021  021050                          150$:
5022
5023  021050  016003  000040                  MOV     RMMR2(R0),R3    ;STORE RMMR2 AT R3
5024  021054  042703  176000                  BIC     #^C1777,R3
5025  021060  001413                          BEQ     160$
5026  021062  010337  001142                  MOV     R3,$BDDAT
5027  021066  005037  001140                  CLR     $GDDAT
5028  021072  010037  001136                  MOV     R0,$BDADR
5029  021076  062737  000040  001136          ADD     #RMMR2,$BDADR
5030  021104  104053                          ERROR   53                      ;READ GATE WON'T RESET
5031  021106  000445                          BR      190$
5032  021110                          160$:
5033                          ;;*****************************************************************
5034                          ;VERIFY THERE ARE NO  ECC ERRORS
5035
5036  021110  016003  000014                  MOV     RMER1(R0),R3    ;STORE RMER1 AT R3
5037  021114  042703  077677                  BIC     #^C<DCK!ECH>,R3
5038  021120  012737  000000  001140          MOV     #000000,$GDDAT
5039  021126  023703  001140                  CMP     $GDDAT,R3
5040  021132  001411                          BEQ     170$
5041  021134  010337  001142                  MOV     R3,$BDDAT
```

```
5042  021140  010037  001136          MOV     R0,$BDADR
5043  021144  062737  000014  001136  ADD     #RMER1,$BDADR
5044  021152  104064                  ERROR   64
5045  021154  000422                  BR      190$
5046  021156                  170$:
5047                          ;;*****************************************************************
5048                          ;VERIFY DATA BUFFER
5049  021156  012702  054116          MOV     #BUFONE,R2      ;DATA WAS STORED HERE
5050  021162  012703  177777          MOV     #177777,R3      ;EACH WORD ALL ONES
5051  021166  012704  000400          MOV     #256.,R4
5052  021172  022203          180$:   CMP     (R2)+,R3
5053  021174  001003                  BNE     185$
5054  021176  005304                  DEC     R4
5055  021200  001374                  BNE     180$
5056  021202  000407                  BR      190$            ;EXIT IF ALL THE LOCATIONS CHECKED
5057  021204  014237  001142  185$:   MOV     -(R2),$BDDAT    ;SETUP ERROR MESSAGE
5058  021210  010337  001140          MOV     R3,$GDDAT
5059  021214  010237  001136          MOV     R2,$BDADR
5060  021220  104065                  ERROR   65                      ;DATA INCORRECT DURING READ
5061  021222  000137  021440  190$:   JMP     400$            ;GREAT ESCAPE
5062  021226  012737  000021  021330  300$:   MOV     #17.,320$
5063                          ;CLOCK PROM STROBE ON
5064  021234                  305$:
5065
5066  021234  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
5067
5068  021242  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
5069
5070  021250  016037  000024  021332          MOV     RMMR1(R0),330$  ;STORE RMMR1 AT 330$
5071  021256  042737  177737  021332          BIC     #^CWC,330$
5072  021264  001020                  BNE     310$            ;EXIT IF PROM STROBE ON
5073  021266  005337  021330          DEC     320$
5074  021272  001360                  BNE     305$
5075  021274  013737  021330  001142          MOV     320$,$BDDAT     ;SETUP ERROR MESSAGE
5076  021302  012737  000040  001140          MOV     #WC,$GDDAT
5077  021310  010037  001136          MOV     R0,$BDADR
5078  021314  062737  000024  001136          ADD     #RMMR1,$BDADR
5079  021322  104030                  ERROR   30
5080  021324  000445                  BR      400$
5081  021326  000207          310$:   RTS     PC
5082  021330  000000          320$:   .WORD   0
5083
5084  021332  000000          330$:   .WORD   0
5085
5086  021334  012737  000021  021434  350$:   MOV     #17.,370$
5087                          ;CLOCK THE STROBE OFF
5088  021342                  355$:
5089
5090  021342  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
5091
5092  021350  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
5093
5094  021356  016037  000024  021436          MOV     RMMR1(R0),380$  ;STORE RMMR1 AT 380$
5095  021364  042737  177737  021436          BIC     #^CWC,380$
5096  021372  001417                  BEQ     360$
5097  021374  005337  021434          DEC     370$
```

```
 5098  021400  001360                        BNE     355$
 5099  021402  013737  021436  001142        MOV     380$,$BDDAT
 5100  021410  005037  001140                CLR     $GDDAT
 5101  021414  010037  001136                MOV     R0,$BDADR
 5102  021420  062737  000024  001136        ADD     #RMMR1,$BDADR
 5103  021426  104062                        ERROR   62              ;CAN'T CLEAR PROM STROBE
 5104  021430  000403                        BR      400$
 5105  021432  000207              360$:     RTS     PC
 5106  021434  000000              370$:     .WORD   0
 5107  021436  000000              380$:     .WORD   0
 5108  021440  000240              400$:     NOP             ;SUB-TEST EXIT POINT
```

```
5109  021442                                  $EOSP:
5110  021442  000240                                  NOP
5111  021444  013700  001456                          MOV     TSTQUE,R0
5112  021450  062700  000002                          ADD     #2,R0
5113  021454  010037  001456                          MOV     R0,TSTQUE
5114  021460  005710                                  TST     (R0)
5115  021462  001402                                  BEQ     1$
5116  021464  000137  004414                          JMP     READY
5117  021470  012737  001460  001456  1$:             MOV     #TSTQUE+2,TSTQUE
5118                                    .SBTTL  END OF PASS ROUTINE
5119
5120                                    ;;**********************************************************************
5121                                    ;*INCREMENT THE PASS NUMBER ($PASS)
5122                                    ;*TYPE 'END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY''
5123                                    ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
5124                                    ;*IF THERES A MONITOR GO TO IT
5125                                    ;*IF THERE ISN'T JUMP TO READY
5126
5127  021476                                  $EOP:
5128  021476  000240                                  NOP
5129  021500  005037  001116                          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
5130  021504  005037  001206                          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
5131  021510  005237  001230                          INC     $PASS           ;;INCREMENT THE PASS NUMBER
5132  021514  042737  100000  001230                  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
5133  021522  005327                                  DEC     (PC)+           ;;LOOP?
5134  021524  000001                          $EOPCT: .WORD   1
5135  021526  003063                                  BGT     $DOAGN          ;;YES
5136  021530  012737                                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
5137  021532  000001                          $ENDCT: .WORD   1
5138  021534  021524                                  $EOPCT
5139  021536  104401  021544                          TYPE    ,65$            ;;TYPE ASCIZ STRING
5140  021542  000407                                  BR      64$             ;;GET OVER THE ASCIZ
5141                                    ;;65$:  .ASCIZ  <12><15>/END PASS #/
5142  021562                                  64$:
5143  021562  013746  001230                          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
5144                                                                          ;;TYPE PASS NUMBER
5145  021566  104405                                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
5146  021570  104401  021576                          TYPE    ,67$            ;;TYPE ASCIZ STRING
5147  021574  000421                                  BR      66$             ;;GET OVER THE ASCIZ
5148                                    ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
5149  021640                                  66$:
5150  021640  013746  001126                          MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
5151                                                                          ;;TOTAL NUMBER OF ERRORS
5152  021644  104405                                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
5153  021646  104401  001217                          TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
5154  021652  005037  001126                          CLR     $ERTTL          ;;CLEAR ERROR TOTAL
5155  021656  013700  000042              $GET42:     MOV     @#42,R0         ;;GET MONITOR ADDRESS
5156  021662  001405                                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
5157  021664  000005                                  RESET                   ;;CLEAR THE WORLD
5158  021666  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
5159  021670  000240                                  NOP                     ;;SAVE ROOM
5160  021672  000240                                  NOP                     ;;FOR
5161  021674  000240                                  NOP                     ;;ACT11
5162  021676                                  $DOAGN:
5163  021676  000137                                  JMP     @(PC)+          ;;RETURN
5164  021700  004414                          $RTNAD: .WORD   READY
```

```
 5165  021702      377      377      000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
 5166           021706                            .EVEN
```

```
                                        .SBTTL  SUBROUTINES

                                        ;;*******************************************************************
                                        .SBTTL  ERROR TYPEOUT ROUTINE

                                        ;*THE ERROR TYPEOUT ROUTINE ASSEMBLES AND PRINTS INFORMATION
                                        ;*REGARDING THE DETECTION OF AN ERROR AS FOLLOWS:
                                        ;*
                                        ;*      .UNIT NUMBER, TEST NUMBER, ERROR NUMBER AND PROGRAM COUNTER ARE
                                        ;*PRINTED ON THE FIRST LINE;
                                        ;*      .ERROR MESSAGE IS ASSEMBLED, FORMATTED AND PRINTED ON
                                        ;*ONE OR MORE SUCCEEDING LINES;
                                        ;*      .PAIRED LINES OF ERROR HEADERS AND ERROR DATA
                                        ;*ARE PRINTED AFTER THE ERROR MESSAGE.

021706                                  ERRTYP:
021706  104414                                SAVREG
021710  032777  020000  157236                BIT     #SW13,@SWR      ;INHIBIT TYPEOUTS??
021716  001402                                BEQ     1$              ;NO!!
021720  000137  022436                        JMP     21$             ;YES.!
                                        ;TYPE UNIT NUMBER, TEST NUMBER, ERROR NUMBER, AND PROGRAM COUNTER
021724  104401  001217                1$:     TYPE    ,$CRLF
021730  104401  022452                        TYPE    ,ERTY00         ;TYPE 'UNT#''
021734  013746  001234                        MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
                                                                        ;;TYPE UNIT NUMBER
021740  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
021742     003                                .BYTE   3               ;;TYPE 3 DIGIT(S)
021743     000                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
021744  005037  022442                        CLR     TSTNMB          ;LOAD TEST NUMBER FOR
021750  013737  001226  022442                MOV     $TESTN,TSTNMB
021756  104401  022460                        TYPE    ,ERTY01         ;TYPE 'TST#''
021762  013746  022442                        MOV     TSTNMB,-(SP)    ;;SAVE TSTNMB FOR TYPEOUT
                                                                        ;;TYPE TEST NUMBER
021766  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
021770     003                                .BYTE   3               ;;TYPE 3 DIGIT(S)
021771     000                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
021772  005037  022444                        CLR     ERRNMB          ;LOAD ERROR NUMBER FOR
021776  113737  001130  022444                MOVB    $ITEMB,ERRNMB   ;TYPEOUT
022004  001406                                BEQ     2$              ;SKIP IF NO ERROR CALLED
022006  104401  022470                        TYPE    ,ERTY02         ;TYPE 'ERR#''
022012  013746  022444                        MOV     ERRNMB,-(SP)    ;;SAVE ERRNMB FOR TYPEOUT
                                                                        ;;TYPE ERROR NUMBER
022016  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
022020     003                                .BYTE   3               ;;TYPE 3 DIGIT(S)
022021     000                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
022022  104401  022477                2$:     TYPE    ,ERTY03         ;TYPE 'PC='
022026  013746  001132                        MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                        ;;TYPE PROGRAM COUNTER
022032  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
022034     006                                .BYTE   6               ;;TYPE 6 DIGIT(S)
022035     001                                .BYTE   1               ;;TYPE LEADING ZEROS
                                        ;GENERATE POINTER TO ERROR TABLE UNLESS ERROR NUMBER IS 0
022036  005737  022444                3$:     TST     ERRNMB          ;WAS AN ERROR CALLED?
022042  001575                                BEQ     21$             ;NO!!
022044  104401  001217                        TYPE    ,$CRLF          ;YES-TYPE CRLF
022050  105037  022450                        CLRB    BOTFLG          ;CLEAR BOT FLAG
```

```
5223  022054  105037  022451          CLRB    CHRCNT          ;CLEAR CHARACTER COUNTER
5224  022060  013700  022444          MOV     ERRNMB,R0       ;R0 POINTS TO FIRST OF
5225  022064  006300                  ASL     R0              ;FOUR ENTRIES IN ERROR
5226  022066  006300                  ASL     R0              ;TABLE
5227  022070  006300                  ASL     R0
5228  022072  062700  001522          ADD     #$ERRTB-8.,R0
5229  022076  011001                  MOV     (R0),R1         ;R1 POINTS TO ERROR MESSAGE
5230                                                          ;TABLE
5231  022100  001507                  BEQ     12$             ;BRANCH IF NO ERROR MESSAGE
5232                          ;TYPE THE ERROR MESSAGE
5233  022102  012102          4$:     MOV     (R1)+,R2        ;R2=ADDRESS OF MESSAGE STRING
5234  022104  001505                  BEQ     12$             ;BRANCH IF END OF MESSAGE
5235  022106  010237  022254          MOV     R2,11$          ;LOAD ADDRESS OF STRING
5236  022112  005037  022446          CLR     BOTADR          ;CLEAR BOT ADDRESS
5237  022116  112203          5$:     MOVB    (R2)+,R3        ;END OF STRING??
5238  022120  001454                  BEQ     10$             ;YES!!
5239  022122  122703  000015          CMPB    #CR,R3          ;CARRIAGE RETURN??
5240  022126  001003                  BNE     6$              ;NO!!
5241  022130  105037  022451          CLRB    CHRCNT          ;YES-CLEAR CHAR COUNT
5242  022134  000770                  BR      5$              ;GET NEXT CHARACTER
5243  022136  122703  000012          CMPB    #LF,R3          ;LINE FEED??
5244  022142  001765                  BEQ     5$              ;YES-GET NEXT CHARACTER
5245  022144  122703  000011          CMPB    #HT,R3          ;HORIZONTAL TAB??
5246  022150  001007                  BNE     8$              ;NO.!
5247  022152  105237  022451          INCB    CHRCNT          ;ADJUST CHARACTER COUNT
5248  022156  132737  000007  022451  BITB    #7,CHRCNT
5249  022164  001372                  BNE     7$
5250  022166  000407                  BR      9$
5251  022170  105237  022451          INCB    CHRCNT          ;INCREMENT CHARACTER COUNT
5252  022174  122703  000040          CMPB    #' ,R3          ;SPACE??
5253  022200  001002                  BNE     9$              ;NO!!
5254  022202  010237  022446          MOV     R2,BOTADR       ;SAVE ADDRESS OF SPACE
5255  022206  122737  000100  022451  CMPB    #64.,CHRCNT     ;END OF LINE??
5256  022214  103340                  BHIS    5$              ;NO!!
5257  022216  013704  022446          MOV     BOTADR,R4       ;GET ADDRESS OF LAST SPACE
5258  022222  001007                  BNE     90$             ;BRANCH IF SPACE DETECTED
5259  022224  104401  001217          TYPE    ,$CRLF          ;TYPE CRLF
5260  022230  105037  022451          CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
5261  022234  013702  022254          MOV     11$,R2          ;SET UP R2 FOR TESTING
5262  022240  000726                  BR      5$
5263  022242  105044          90$:    CLRB    -(R4)           ;REPLACE SPACE
5264  022244  112737  177777  022450  MOVB    #-1,BOTFLG      ;SET BOT FLAG
5265  022252  104401          10$:    TYPE                    ;TYPE ERROR MESSAGE STRING
5266  022254  000000          11$:    .WORD                   ;STRING ADDRESS GOES HERE
5267  022256  105737  022450          TSTB    BOTFLG          ;WAS STRING TRUNCATED??
5268  022262  001707                  BEQ     4$              ;NO!!
5269  022264  104401  001217          TYPE    ,$CRLF          ;YES-TYPE CRLF
5270  022270  105037  022450          CLRB    BOTFLG          ;CLEAR BOT FLAG
5271  022274  105037  022451          CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
5272  022300  013702  022446          MOV     BOTADR,R2       ;SETUP R2 FOR TESTING
5273  022304  010237  022254          MOV     R2,11$          ;SETUP 11$ FOR TYPING
5274  022310  112742  000040          MOVB    #' ,-(R2)       ;RESTORE SPACE
5275  022314  105722                  TSTB    (R2)+           ;RESTORE R2
5276  022316  000677                  BR      5$              ;TYPE REST OF STRING
5277  022320          12$:
5278                          ;TYPE ERROR HEADER AND ERROR DATA
```

```
5279  022320                          13$:
5280  022320  016001  000002                  MOV     2(R0),R1        ;R1 POINTS TO ERROR HEADER TABLE
5281  022324  001444                          BEQ     21$             ;BRANCH IF NO HEADER
5282  022326  104401  001217                  TYPE    ,$CRLF          ;(ASSUME NO DATA)
5283  022332  016002  000004                  MOV     4(R0),R2        ;R2 POINTS TO DATA ADDRESS TABLE
5284  022336  016003  000006                  MOV     6(R0),R3        ;R3 POINTS TO FORMAT TABLE
5285  022342  012137  022352          14$:    MOV     (R1)+,15$       ;PUT HEADER ADDRESS FOR TYPE
5286  022346  001433                          BEQ     21$             ;BRANCH IF END OF HEADERS
5287                                                                  ;(ASSUME END OF DATA)
5288  022350  104401                          TYPE
5289  022352  000000          15$:    .WORD   0               ;HEADER ADDRESS GOES HERE
5290  022354  104401  001217                  TYPE    ,$CRLF
5291  022360  005702                          TST     R2              ;DATA WITH HEADER??
5292  022362  001767                          BEQ     14$             ;NO!!
5293  022364  012204                          MOV     (R2)+,R4        ;R4 POINTS TO DATA ADDRESS
5294  022366  012305                          MOV     (R3)+,R5        ;R5 POINTS TO FORMAT
5295  022370  105725          16$:    TSTB    (R5)+           ;WHAT KIND OF DATA??
5296  022372  100407                          BMI     18$             ;BINARY
5297  022374  001403                          BEQ     17$             ;OCTAL
5298  022376  013446                          MOV     @(R4)+,-(SP)    ;DECIMAL
5299  022400  104405                          TYPDS
5300  022402  000405                          BR      19$
5301  022404  013446          17$:    MOV     @(R4)+,-(SP)
5302  022406  104402                          TYPOC
5303  022410  000402                          BR      19$
5304  022412  013446          18$:    MOV     @(R4)+,-(SP)
5305  022414  104406                          TYPBN
5306  022416  005714          19$:    TST     (R4)            ;MORE DATA??
5307  022420  001403                          BEQ     20$             ;NO!.
5308  022422  104401  022505                  TYPE    ,ERTY04         ;YES-TYPE 2 SPACES
5309  022426  000760                          BR      16$             ;AND CONTINUE
5310  022430  104401  001217          20$:    TYPE    ,$CRLF          ;TYPE ONE BLANK LINE
5311  022434  000742                          BR      14$             ;BEFORE NEXT HEADER
5312  022436  104415          21$:    RESREG
5313  022440  000207                          RTS     PC
5314
5315  022442  000000          TSTNMB: .WORD   0               ;TEST NUMBER
5316  022444  000000          ERRNMB: .WORD   0               ;ERROR NUMBER
5317  022446  000000          BOTADR: .WORD   0               ;BEGINNING OF TEXT ADDRESS
5318  022450     000          BOTFLG: .BYTE   0               ;BOT FLAG
5319  022451     000          CHRCNT: .BYTE   0               ;CHARACTER COUNT
5320
5321  022452  047125  052111  000043  ERTY00: .ASCIZ  @UNIT#@
5322  022460  020054  042524  052123  ERTY01: .ASCIZ  @, TEST#@
5323  022466  000043
5324  022470  020054  051105  021522  ERTY02: .ASCIZ  @, ERR#@
5325  022476     000
5326  022477     054  050040  036503  ERTY03: .ASCIZ  @, PC @
5327  022504     000
5328  022505     040  000040          ERTY04: .ASCIZ  @ @
5329                                  .EVEN
5330
```

```
                                        .SBTTL  CLOCK SUBROUTINES

                                        ;ROUTINE TO SIZE FOR CLOCKS (KW11-L OR KW11-P)
                                        ;;*****************************************************************
022510  000240                  SIZCLK: NOP
022512  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
022516  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
022522  012737  022606  000004          MOV     #10$,ERRVEC     ;LOAD 04 TRAP VECTORS
022530  012737  000300  000006          MOV     #PR6,ERRVEC+2

                                        ;SEE IF A KW11-P CLOCK IS PRESENT - GO TO 10$ IF NOT PRESENT
022536  005777  156740                  TST     @$LPCSR         ;TEST FOR P CLOCK
022542  012737  022750  001526          MOV     #PCLOCK,CLOCK   ;LOAD SUBROUTINE ADDRESS
022550  012737  023072  001530          MOV     #PSTOP,STOP     ;LOAD STOP ADDRESS
022556  012777  023036  156722          MOV     #PCOUNT,@$LPVEC ;LOAD P CLOCK INTERRUPT VECTOR
022564  012777  000300  156716          MOV     #PR6,@$LPVEC+2
022572  013777  001516  156714          MOV     $LLVEC+2,@$LLVEC;CLEAR L CLOCK INTERRUPT VECTOR
022600  005077  156712                  CLR     @$LLVEC+2
022604  000454                          BR      30$
022606  012716  022614          10$:    MOV     #15$,(SP)       ;DUMMY RTI ADDRESS
022612  000002                          RTI                     ;RESTORE PRIORITY
022614                          15$:
                                        ;NO P CLOCK-SEE IF L CLOCK IS PRESENT-GO TO 20$ IF NOT PRESET
022614  012737  022672  000004          MOV     #20$,ERRVEC     ;CHANGE 04 TRAP VECTOR
022622  005777  156664                  TST     @$LLCSR         ;TEST FOR L CLOCK
022626  012737  022766  001526          MOV     #LCLOCK,CLOCK   ;LOAD SUBROUTINE ADDRESS
022634  012737  023100  001530          MOV     #LSTOP,STOP     ;LOAD STOP ADDRESS
022642  012777  023036  156644          MOV     #LCOUNT,@$LLVEC ;LOAD L CLOCK INTERRUPT VECTOR
022650  012777  000300  156640          MOV     #PR6,@$LLVEC+2
022656  013777  001510  156622          MOV     $LPVEC+2,@$LPVEC;CLEAR P CLOCK INTERRUPT VECTOR
022664  005077  156620                  CLR     @$LPVEC+2
022670  000422                          BR      30$
022672  012716  022700          20$:    MOV     #25$,(SP)       ;DUMMY RTI ADDRESS
022676  000002                          RTI                     ;RESTORE PRIORITY
022700                          25$:
                                        ;NO CLOCK AVAILABLE - AUGMENT RETURN ADDRESS
022700  005037  001526                  CLR     CLOCK           ;CLEAR SUBROUTINE ADDRESS
022704  012737  001510  001506          MOV     #$LPVEC+2,$LPVEC;CLEAR P CLOCK INTERRUPT VECTOR
022712  005037  001510                  CLR     $LPVEC+2
022716  012737  001516  001514          MOV     #$LLVEC+2,$LLVEC;CLEAR L CLOCK INTERRUPT VECTOR
022724  005037  001516                  CLR     $LLVEC+2
022730  062766  000002  000004          ADD     #2,4(SP)        ;CHANGE RETURN ADDRESS

022736                          30$:
022736  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
022742  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
022746  000207                          RTS     PC
                                        ;ROUTINES TO START THE CLOCK (KW11-L OR KW11-P)
                                        ;;*****************************************************************

022750  012777  177777  156526  PCLOCK: MOV     #-1,@$LPCSB     ;LOAD COUNT SET BUFFER
022756  012777  000135  156516          MOV     #135,@$LPCSR    ;LOAD CONTROL REGISTER
022764  000403                          BR      PLCLK           ;GO TO COMMON CODE

022766  012777  000100  156516  LCLOCK: MOV     #100,@$LLCSR    ;LOAD CONTROL REGISTER
```

```
5387  022774  005037  001522          PLCLK:  CLR     TIME            ;CLEAR TIMER COUNT
5388  023000  104400                          TRAP                    ;;PUSH OLD PSW AND PC ON STACK
5389  023002  012605                          MOV     (SP)+,R5        ;;SAVE THE PSW IN R5
5390  023004  010537  001520                  MOV     R5,$PSW         ;SAVE PRIORITY
5391  023010  042705  177437                  BIC     #^CPR7,R5       ;MASK X
5392  023014  022705  000300                  CMP     #PR6,R5         ;IS PRIORITY TOO HIGH??
5393  023020  101005                          BHI     40$             ;NO!!
5394  023022  012746  000240                  MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
5395  023026  012746  023034                  MOV     #30$,-(SP)      ;;PUT NEW PC ON  STACK
5396  023032  000002                          RTI                     ;;POP NEW PC AND PS
5397  023034                          30$:
5398  023034  000207                  40$:    RTS     PC
5399
5400                                  ;ROUTINES TO HANDLE CLOCK INTERRUPTS (KW11-L OR KW11-P)
5401
5402  023036                          PCOUNT:
5403  023036                          LCOUNT:
5404  023036  062737  000021  001522          ADD     #17.,TIME       ;ADD 17MS TO ELAPSED TIME
5405  023044  103003                          BCC     10$             ;BRANCH IF NO OVERFLOW
5406  023046  012737  177777  001522          MOV     #-1,TIME        ;RESTORE MAXIMUM COUNT
5407  023054  162737  000021  001524  10$:    SUB     #17.,WATCH      ;DECREMENT REMAINING TIME
5408  023062  100002                          BPL     20$             ;BRANCH IF POSITIVE
5409  023064  005037  001524                  CLR     WATCH           ;CLEAR REMAINING TIME
5410  023070  000002                  20$:    RTI                     ;RETURN TO USER
5411
5412                                  ;ROUTINES TO STOP THE CLOCK (KW11-L OR KW11-P)
5413
5414  023072  005077  156404          PSTOP:  CLR     @$LPCSR         ;STOP P CLOCK
5415  023076  000402                          BR      PLSTP           ;GO TO COMMON STOP CODE
5416
5417  023100  005077  156406          LSTOP:  CLR     @$LLCSR         ;STOP L CLOCK
5418
5419  023104                          PLSTP:
5420  023104  013746  001520                  MOV     $PSW,-(SP)      ;;PUT NEW PS ON STACK
5421  023110  012746  023116                  MOV     #10$,-(SP)      ;;PUT NEW PC ON  STACK
5422  023114  000002                          RTI                     ;;POP NEW PC AND PS
5423  023116                          10$:
5424  023116  000207                          RTS     PC
5425
5426                                  .SBTTL  SET VOLUME VALID SUBROUTINE
5427
5428                                  ;THIS SUBROUTINE INITIALIZES THE SUBSYSTEM AND SETS VOLUME VALID,
5429                                  ;RETURNING WITH THE DRIVE STILL IN DIAGNOSTIC MODE.  THE SUBROUTINE
5430                                  ;RETURNS TO THE WORD FOLLOWING THE CALL, EXCEPT WHEN AN ERROR IS
5431                                  ;DETECTED, IN WHICH CASE IT RETURNS TO THE SECOND WORD FOLLOWING THE
5432                                  ;CALL.
5433                                  ;
5434                                  ;CALL:  JSR     PC,SETVV        JUMP TO SUBROUTINE
5435                                  ;       BR      ??              RETURN HERE IF NO ERROR
5436                                  ;       ERROR                   RETURN HERE IF ERROR
5437
5438  023120                          SETVV:
5439  023120  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
5440  023126  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
5441
5442  023132  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
```

```
5443
5444  023140  012760  001001  000024          MOV     #DMD.MUR,RMMR1(R0)      ;LOAD RMMR1
5445
5446  023146  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
5447
5448  023154  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
5449
5450  023162  012760  000023  000000          MOV     #PACACK.GO,RMCS1(R0)    ;LOAD RMCS1
5451
5452  023170  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
5453  023176  042737  177677  001142          BIC     #^CVV,$BDDAT
5454  023204  001020                          BNE     10$             ;BRANCH IF VOLUME VALID SET
5455  023206  010037  001136                  MOV     R0,$BDADR       ;SETUP FOR ERROR MSG
5456  023212  062737  000012  001136          ADD     #RMDS,$BDADR
5457  023220  012737  000100  001140          MOV     #VV,$GDDAT
5458  023226  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN ADDRESS TO ERROR
5459  023232  112776  000100  000000          MOVB    #100,@(SP)      ;WRITE ERROR NUMBER
5460  023240  012737  000022  001174          MOV     #PACACK,$TMP0
5461  023246  000207                  10$:    RTS     PC              ;RETURN
5462
5463                                  .SBTTL  SET OFFSET MODE SUBROUTINE
5464
5465                                  ;THIS SUBROUTINE EXECUTES AN OFFSET COMMAND AND VERIFIES THAT OFFSET
5466                                  ;MODE SETS.  THE DRIVE SHOULD BE IN DIAGNOSTIC MODE WHEN CALLING THE
5467                                  ;SUBROUTINE, WHICH WILL LEAVE DMD ON.  THE SUBROUTINE RETURNS TO THE
5468                                  ;WORD FOLLOWING THE CALL UNLESS THERE IS AN ERROR, IN WHICH CASE IT
5469                                  ;RETURNS TO THE SECOND WORD FOLLOWING THE CALL
5470                                  ;
5471                                  ;CALL:  JSR     PC,SETOM        JUMP TO SUBROUTINE
5472                                  ;       BR      ??              RETURN HERE IF NO ERROR
5473                                  ;       ERROR                   RETURN HERE IF ERROR
5474
5475  023250                          SETOM:
5476
5477  023250  012760  001001  000024          MOV     #DMD.MUR,RMMR1(R0)      ;LOAD RMMR1
5478
5479  023256  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
5480
5481  023264  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
5482
5483  023272  012760  000015  000000          MOV     #OFFSET!GO,RMCS1(R0)    ;LOAD RMCS1
5484
5485  023300  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
5486  023306  042737  177776  001142          BIC     #^COM,$BDDAT
5487  023314  001015                          BNE     10$             ;BRANCH IF OFFSET ON
5488  023316  012737  000001  001140          MOV     #OM,$GDDAT
5489  023324  010037  001136                  MOV     R0,$BDADR
5490  023330  062737  000012  001136          ADD     #RMDS,$BDADR
5491  023336  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN ADDRESS TO ERROR
5492  023342  112776  000111  000000          MOVB    #111,@(SP)      ;WRITE ERROR NUMBER
5493  023350                          10$:
5494  023350  000207                          RTS     PC              ;RETURN TO USER
5495                                  .SBTTL  ENABLE SEARCH SUBROUTINE
5496                                  ;;*****************************************************************
5497
5498  023352                          ENBSCH:
```

```
                                        ;THE REGISTER OUTPUT BUFFER  SHOULD CONTAIN:
                                        ;       RMDA0   = DESIRED SECTOR AND TRACK ADDRESS
                                        ;       RMDC0   = DESIRED CYLINDER ADDRESS
                                        ;       RMOF0   = FORMAT, ECI, HCI
                                        ;       RMBA0   = BUFFER ADDRESS
                                        ;       RMWC0   = WORD COUNT
                                        ;       RMCS10  = FUNCTION CODE

                                        ;NOTE:  OFFSET IS NOT ENABLED WHEN USING THIS SUBROUTINE

                                        ;;*****************************************************************
                                        ;SET VOLUME VALID
                                        ;FIRST,CLEAR -SET DIAGNOSTIC MODE-SYNCHRONIZE
                                        ;PROM STROBE
023352  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
023360  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

023364  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
023372  012704  000041                  MOV     #33.,R4         ;ALLOW UP TO 33 BIT CLOCLS
                                                                ;TO SYNC PROM STROBE
                                        ;.....
023376                                  5$:

023376  012760  005001  000024          MOV     #DMD!MUR!MCLK,RMMR1(R0) ;LOAD RMMR1

023404  012760  001001  000024          MOV     #DMD!MUR,RMMR1(R0)      ;LOAD RMMR1

023412  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
023416  032705  000040                  BIT     #WC,R5  ;WAIT FOR PROM STROBE TO COME ON
023422  001023                          BNE     6$
023424  005304                          DEC     R4
023426  001363                          BNE     5$
023430  010037  001136                  MOV     R0,$BDADR       ;PROM STROBE WONT SET
023434  062737  000024  001136          ADD     #RMMR1,$BDADR
023442  005037  001142                  CLR     $BDDAT
023446  012737  000040  001140          MOV     #WC,$GDDAT
023454  062716  000002                  ADD     #2,(SP)
023460  112776  000030  000000          MOVB    #30,@(SP)
023466  000137  024230                  JMP     60$             ;REPROT ERROR
023472                          6$:

023472  012760  005001  000024          MOV     #DMD!MUR!MCLK,RMMR1(R0) ;LOAD RMMR1

023500  012760  001001  000024          MOV     #DMD!MUR,RMMR1(R0)      ;LOAD RMMR1

023506  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
023512  032705  000040                  BIT     #WC,R5          ;WAIT FOR PROM STROBE TO GO OFF
023516  001423                          BEQ     7$
023520  005304                          DEC     R4
023522  001363                          BNE     6$
023524  010037  001136                  MOV     R0,$BDADR       ;PROM STROBE WONT RESET
023530  062737  000024  001136          ADD     #RMMR1,$BDADR   ;
023536  012737  000040  001142          MOV     #WC,$BDDAT
023544  005037  001140                  CLR     $GDDAT
023550  062716  000002                  ADD     #2,(SP)
```

```
5555  023554  112776  000062  000000          MOVB    #62,@(SP)
5556  023562  000137  024230                  JMP     60$
5557  023566                          7$:
5558                                  ;SECOND, CLOCK INDEX PULSE TO 1. CLEAR FORMAT CHANGE FLOP
5559                                  ;                               2. CLEAR RTC FLOP
5560
5561  023566  012760  001005  000024          MOV     #DMD!MUR!MI,RMMR1(R0)   ;LOAD RMMR1
5562
5563  023574  012760  001001  000024          MOV     #DMD.MUR,RMMR1(R0)      ;LOAD RMMR1
5564                                  ;......
5565
5566  023602  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
5567
5568  023610  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
5569
5570  023616  012760  000023  000000          MOV     #PACACK!GO,RMCS1(R0)    ;LOAD RMCS1
5571
5572  023624  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
5573  023632  042737  177677  001142          BIC     #^CVV,$BDDAT    ;VOLUME VALID SHOULD BE SET
5574  023640  001021                          BNE     10$
5575  023642  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MSG
5576  023646  062737  000012  001136          ADD     #RMDS,$BDADR
5577  023654  012737  000100  001140          MOV     #VV,$GDDAT
5578  023662  062716  000002                  ADD     #2,(SP)         ;WRITE ERROR NUMBER
5579  023666  112776  000100  000000          MOVB    #100,@(SP)      ;
5580  023674  012737  000022  001174          MOV     #PACACK,$TMP0
5581  023702  000552                          BR      60$
5582  023704                          10$:
5583                                  ;;**************************************************************************
5584                                  ;LOAD REGISTERS
5585
5586  023704  013760  001410  000006          MOV     RMDA0,RMDA(R0)  ;LOAD RMDA
5587
5588  023712  013760  001436  000034          MOV     RMDC0,RMDC(R0)  ;LOAD RMDC
5589
5590  023720  013760  001434  000032          MOV     RMOF0,RMOF(R0)  ;LOAD RMOF
5591
5592  023726  013760  001404  000002          MOV     RMWC0,RMWC(R0)  ;LOAD RMWC
5593
5594  023734  013760  001406  000004          MOV     RMBA0,RMBA(R0)  ;LOAD RMBA
5595                                  ;;**************************************************************************
5596                                  ;ENABLE DEBUG CLOCK AND LOAD FUNCTION CODE
5597
5598  023742  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
5599
5600  023750  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
5601
5602  023756  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
5603
5604  023764  013760  001402  000000          MOV     RMCS10,RMCS1(R0)        ;LOAD RMCS1
5605                                  ;;**************************************************************************
5606                                  ;WAIT FOR 'RUN AND GO'' TO SET
5607  023772  012737  000310  001524          MOV     #200.,WATCH     ;SET WATCHDOG TIMER VALUE
5608  024000  004777  155522                  JSR     PC,@CLOCK       ;START THE CLOCK
5609  024004                          20$:
5610
```

```
 5611  024004  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
 5612  024012  042737  137777  001142          BIC     #^CRG,$BDDAT
 5613  024020  001023                          BNE     30$
 5614  024022  005737  001524                  TST     WATCH           ;ANY TIME LEFT?
 5615  024026  001366                          BNE     20$             ;YES-WAIT
 5616  024030  004777  155474                  JSR     PC,@STOP        ;STOP THE CLOCK
 5617  024034  012737  040000  001140          MOV     #RG,$GDDAT      ;SETUP ERROR MSG
 5618  024042  010037  001136                  MOV     R0,$BDADR
 5619  024046  062737  000024  001136          ADD     #RMMR1,$BDADR
 5620  024054  062716  000002                  ADD     #2,(SP)         ;WRITE ERROR NUMBER IN
 5621  024060  112776  000101  000000          MOVB    #101,@(SP)              ;USER'S CALL -
 5622
 5623  024066  000460                          BR      60$
 5624  024070                          30$:
 5625  024070  004777  155434                  JSR     PC,@STOP        ;STOP THE CLOCK
 5626                                  ;;**********************************************************************
 5627                                  ;STEP COMMAND SEQUENCER TO HEAD SEQUENCE AT LOCATION 156 (17 CLOCKS)
 5628  024074  012704  000021                  MOV     #17.,R4         ;R4=CLOCK COUNT
 5629  024100                          40$:
 5630
 5631  024100  012760  141401  000024          MOV     #DMD!MUR!DBEN!MOC!DBCK,RMMR1(R0)        ;LOAD RMMR1
 5632
 5633  024106  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
 5634  024114  005304                          DEC     R4
 5635  024116  001370                          BNE     40$
 5636                                  ;;**********************************************************************
 5637                                  ;DROP ON CYLINDER TO RESET LATCH, THEN SET ON CYLINDER TO PASS
 5638                                  ;TEST AT LOCATION 166
 5639
 5640  024120  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
 5641
 5642  024126  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
 5643                                  ;;**********************************************************************
 5644                                  ;STEP SEQUENCER (35 CLOCKS) AND VERIFY SEARCH IS ENABLED
 5645  024134  012704  000043                  MOV     #35.,R4         ;R4=CLOCK COUNT
 5646  024140                          50$:
 5647
 5648  024140  012760  151401  000024          MOV     #DMD!MUR!DBEN!MOC.DBCK.DTO,RMMR1(R0)    ;LOAD RMMR1
 5649
 5650  024146  012760  051401  000024          MOV     #DMD.MUR!DBEN!MOC.DTO,RMMR1(R0) ;LOAD RMMR1
 5651  024154  005304                          DEC     R4
 5652  024156  001370                          BNE     50$
 5653
 5654  024160  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
 5655  024166  042737  173777  001142          BIC     #^CESRC,$BDDAT  ;SEARCH SHOULD BE ENABLED
 5656  024174  001015                          BNE     60$
 5657  024176  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
 5658  024202  062737  000024  001136          ADD     #RMMR1,$BDADR
 5659  024210  012737  004000  001140          MOV     #ESRC,$GDDAT
 5660  024216  062716  000002                  ADD     #2,(SP)         ;WRITE ERROR NUMBER
 5661  024222  112776  000102  000000          MOVB    #102,@(SP)
 5662  024230                          60$:
 5663                                  ;;**********************************************************************
 5664                                  ;RETURN TO USER
 5665  024230  000207                          RTS     PC
```

```
5666                                    .SBTTL  SECTOR COMPARE SUBROUTINE
5667
5668                                    ;SECTOR COMPARE SUBROUTINE
5669
5670                                    ;THIS SUBROUTINE CONTINUES THE EXECUTION OF A DATA COMMAND
5671                                    ;FROM WHERE SEARCH HAS BEEN ENABLED TO WHERE SECTOR COMPARE
5672                                    ;IS SET.
5673
5674  024232                            SCTCMP:
5675
5676                                    ;;*****************************************************************
5677                                    ;SET INDEX PULSE TO CLEAR FORMAT CHANGE FLOP
5678
5679  024232  012760  051405  000024            MOV     #DMD.MUR!DBEN!MOC!DTO!MI,RMMR1(R0)      ;LOAD RMMR1
5680
5681  024240  012760  051401  000024            MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5682                                    ;WAIT AT LEAST 4 MS FOR "RETURN TO CENTER LINE" ONE SHOT TO SET
5683  024246  012737  000006  001524            MOV     #6,WATCH        ;SET WATCHDOG TIMER VALUE
5684  024254  004777  155246                    JSR     PC,@CLOCK       ;START THE CLOCK
5685  024260  005737  001524            10$:    TST     WATCH
5686  024264  001375                            BNE     10$
5687  024266  004777  155236                    JSR     PC,@STOP        ;STOP THE CLOCK
5688                                    ;DATA INPUT TO SEARCH ENABLE FLOP SHOULD BE ONE - CLOCK SECTOR PULSE TO
5689                                    ;SET FLOP
5690
5691  024272  012760  051441  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MS,RMMR1(R0)      ;LOAD RMMR1
5692
5693  024300  012760  051401  000024            MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5694                                    ;WITH SECTOR COMPARE HIGH, CLOCK SECTOR PULSE TO SET SECTOR COMPARE FLOP
5695
5696  024306  012760  051403  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
5697
5698  024314  012760  051443  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MSC!MS,RMMR1(R0)  ;LOAD RMMR1
5699
5700  024322  012760  051403  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
5701
5702  024330  012760  051401  000024            MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5703
5704                                    ;;*****************************************************************
5705                                    ;RETURN TO USER
5706  024336  000207                            RTS     PC
5707
5708                                    ;;*****************************************************************
5709                                    .SBTTL  SET LOOKING FOR SYNC SUBROUTINE
5710                                    ;THIS SUBROUTINE WILL SETUP THE DATA TIMING SEQUENCER
5711                                    ;ASSUMING SEARCH IS ENABLED,TO THE POINT WHERE "PLFS" IS ACTIVE
5712
5713  024340                            SETLFS:
5714                                    ;PULSE BIT CLOCK UNTIL PROM STROBE SETS
5715  024340                            10$:
5716
5717  024340  012760  055401  000024            MOV     #DMD!MUR.DBEN.MOC!DTO.MCLK,RMMR1(R0)     ;LOAD RMMR1
5718
5719  024346  012760  051401  000024            MOV     #DMD!MUR.DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5720
5721  024354  016005  000024                    MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
```

```
5722  024360  032705  000040          BIT     #WC,R5
5723  024364  001765                  BEQ     10$
5724                          ;SET UP DATA SEQUENCER TO LOCATION 11. WHERE PLFS WILL SET
5725  024366  012704  000260          MOV     #176.,R4                ;MAX NUMBER OF BIT CLOCK
5726  024372                        20$:
5727
5728  024372  012760  055401  000024  MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
5729
5730  024400  012760  051401  000024  MOV     #DMD.MUR.DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5731
5732  024406  016005  000024          MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
5733  024412  032705  002000          BIT     #PLFS,R5                ;EXIT IF PLFS IS SET
5734  024416  001010                  BNE     30$
5735  024420  005304                  DEC     R4                      ;ERROR IF COUNT EXHAUSTED
5736  024422  001363                  BNE     20$
5737  024424  062716  000002          ADD     #2,(SP)                 ;
5738  024430  112776  000035  000000  MOVB    #35,@(SP)               ;CANT SET PLFS
5739  024436  000430                  BR      50$
5740  024440                        30$:
5741                          ;STEP THE MAIN REG CLOCK UNTIL THE TRAILING EDGE OF PROM STORBE
5742                          ; IS DETECTED.
5743
5744  024440  016005  000024          MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
5745  024444  032705  000040          BIT     #WC,R5
5746  024450  001007                  BNE     40$
5747
5748  024452  012760  055401  000024  MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
5749
5750  024460  012760  051401  000024  MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
5751  024466  000414                  BR      50$
5752  024470                        40$:
5753
5754  024470  016005  000024          MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
5755  024474  032705  000040          BIT     #WC,R5
5756  024500  001407                  BEQ     50$
5757
5758  024502  012760  055401  000024  MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
5759
5760  024510  012760  051401  000024  MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
5761  024516  000764                  BR      40$
5762  024520                        50$:
5763  024520  000207                  RTS     PC                      ;EXIT
5764
5765                          .SBTTL  CLOCK SYNC SUBROUTINE
5766
5767                          ;;*****************************************************************
5768
5769                          ;THIS SUBROUTINE WILL SIMULATE THE HEADER AND DATA SYNC
5770                          ;PATTERN BEING READ
5771
5772  024522                  CLKSNC:
5773                          ;VERIFY THAT 'PLFS'' IS ON
5774
5775  024522  016004  000024          MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
5776  024526  032704  002000          BIT     #PLFS,R4                ;LOOK FOR SYN SET ?
5777  024532  001023                  BNE     10$                     ;BRANCH IF SO
```

```
5778  024534  010437  001142                  MOV     R4,$BDDAT       ;
5779  024540  042737  175777  001142          BIC     #^CPLFS,$BDDAT
5780  024546  012737  002000  001140          MOV     #PLFS,$GDDAT
5781  024554  010037  001136                  MOV     R0,$BDADR
5782  024560  062737  000024  001136          ADD     #RMMR1,$BDADR
5783  024566  062716  000002                  ADD     #2,(SP)
5784  024572  112776  000035  000000          MOVB    #35,@(SP)
5785  024600  000472                          BR      60$
5786  024602  012705  000021          10$:    MOV     #17.,R5         ;MAKE SURE PROM STROBE IS OFF
5787  024606  032704  000040          20$:    BIT     #WC,R4
5788  024612  001434                          BEQ     30$
5789
5790  024614  012760  055401  000024          MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
5791
5792  024622  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
5793
5794  024630  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
5795  024634  005305                          DEC     R5
5796  024636  001363                          BNE     20$
5797                          ;                       ERROR CAN'T RESET PROM STROBE WITH LFS ACTIVE
5798  024640  010437  001142                  MOV     R4,$BDDAT       ;SETUP ERROR FOR USER
5799  024644  042737  177737  001142          BIC     #^CWC,$BDDAT
5800  024652  005037  001140                  CLR     $GDDAT
5801  024656  010037  001136                  MOV     R0,$BDADR
5802  024662  062737  000024  001136          ADD     #RMMR1,$BDADR
5803  024670  062716  000002                  ADD     #2,(SP)
5804  024674  112776  000062  000000          MOVB    #62,@(SP)
5805  024702  000431                          BR      60$
5806  024704                          30$:
5807                          ;CLOCK THE SYNC PATTERN (00011001) THROUGH THE SHIFT REGISTER
5808  024704  012704  014400                  MOV     #014400,R4
5809  024710  012737  000020  024770          MOV     #16.,70$        ;STROBE BIT COUNT
5810  024716  012705  051401          40$:    MOV     #MR1AAA,R5      ;GENERATE REG VALUE
5811  024722  000241                          CLC                     ;WITH MAINTENANCE CLOCK ON
5812  024724  006004                          ROR     R4
5813  024726  103002                          BCC     50$
5814  024730  052705  002000                  BIS     #MRD,R5         ;SET READ BIT PER PATTERN BIT
5815
5816  024734                          50$:
5817
5818  024734  010560  000024                  MOV     R5,RMMR1(R0)    ;LOAD RMMR1
5819  024740  052705  004000                  BIS     #MCLK,R5
5820
5821  024744  010560  000024                  MOV     R5,RMMR1(R0)    ;LOAD RMMR1
5822  024750  042705  004000                  BIC     #MCLK,R5                ;CLOCK ONE BIT
5823
5824  024754  010560  000024                  MOV     R5,RMMR1(R0)    ;LOAD RMMR1
5825  024760  005337  024770                  DEC     70$
5826  024764  001354                          BNE     40$
5827                          ;CAN'T VERIFY SYNC CLOCK WAS DETECTED
5828                          ;USER CAN DO SO BY STEPING
5829                          ;BIT CLOCK AND VERIFY PROM STROBE SETS WITHIN ONE WORD TIME
5830  024766  000207          60$:    RTS     PC
5831  024770  000000          70$:    .WORD   0               ;TEMPORARY STORAGE
```

```
                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                ;;******************************************************************
                                ;*SAVE R0-R5
                                ;*CALL:
                                ;*      SAVREG
                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                ;*
                                ;*TOP---(+16)
                                ;* +2---(+18)
                                ;* +4---R5
                                ;* +6---R4
                                ;* +8---R3
                                ;*+10---R2
                                ;*+12---R1
                                ;*+14---R0

024772                          $SAVREG:
024772  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
024774  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024776  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
025000  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
025002  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
025004  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
025006  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
025012  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
025016  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
025022  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
025026  000002                          RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
025030                          $RESREG:
025030  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
025034  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
025040  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
025044  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
025050  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
025052  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
025054  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
025056  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
025060  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
025062  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
025064  000002                          RTI
                                .SBTTL  BINARY TO ASCII AND TYPE ROUTINE

                                ;;******************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 16-BIT
                                ;*BINARY-ASCII NUMBER AND TYPE IT.
                                ;*CALL:
                                ;*      MOV     NUMBER,-(SP)    ;;NUMBER TO BE TYPED
                                ;*      TYPBN                   ;;TYPE IT

025066  010146                  $TYPBN: MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
025070  016601  000006                  MOV     6(SP),R1        ;;GET THE INPUT NUMBER
```

```
 5888  025074  000261                          SEC                     ;;SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
 5889  025076  112737  000060  025140  1$:     MOVB    #'0,$BIN        ;;SET CHARACTER TO AN ASCII "0".
 5890  025104  006101                          ROL     R1              ;;GET THIS BIT
 5891  025106  001406                          BEQ     2$              ;;DONE?
 5892  025110  105537  025140                  ADCB    $BIN            ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
 5893  025114  104401  025140                  TYPE    ,$BIN           ;;GO TYPE THIS BIT
 5894  025120  000241                          CLC                     ;;CLEAR "C" SO CAN KEEP TRACK OF BITS
 5895  025122  000765                          BR      1$              ;;GO DO THE NEXT BIT
 5896  025124  012601                  2$:     MOV     (SP)+,R1        ;;POP THE STACK INTO R1
 5897  025126  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 5898  025134  012616                          MOV     (SP)+,(SP)
 5899  025136  000002                          RTI                     ;;RETURN TO USER
 5900  025140     000     000          $BIN:   .BYTE   0,0             ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
 5901                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 5902
 5903                                  ;;**************************************************************************
 5904                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 5905                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 5906                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 5907                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 5908                                  ;*REPLACED WITH SPACES.
 5909                                  ;*CALL:
 5910                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 5911                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
 5912
 5913  025142                          $TYPDS:
 5914  025142  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 5915  025144  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 5916  025146  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 5917  025150  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 5918  025152  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 5919  025154  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 5920  025160  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 5921  025164  100004                          BPL     1$              ;;BR IF INPUT IS POS.
 5922  025166  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 5923  025170  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 5924  025176  005000                  1$.     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 5925  025200  012703  025356                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 5926  025204  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 5927  025210  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 5928  025212  016001  025346                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 5929  025216  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 5930  025220  002402                          BLT     4$              ;;BR IF DONE
 5931  025222  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 5932  025224  000774                          BR      3$
 5933  025226  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 5934  025230  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
 5935  025232  001002                          BNE     5$              ;;FALL THROUGH IF 0
 5936  025234  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 5937  025236  100407                          BMI     7$              ;;BR IF YES
 5938  025240  106316                  5$:     ASLB    (SP)            ;;MSD?
 5939  025242  103003                          BCC     6$              ;;BR IF NO
 5940  025244  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 5941  025252  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 5942  025256  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 5943  025262  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
```

```
5944  025264  005720                          TST     (R0)+           ;;JUST INCREMENTING
5945  025266  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
5946  025272  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
5947  025274  003002                          BGT     8$              ;;GO TO EXIT
5948  025276  010502                          MOV     R5,R2           ;;GET THE LSD
5949  025300  000764                          BR      6$              ;;GO CHANGE TO ASCII
5950  025302  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
5951  025304  100003                          BPL     9$              ;;BR IF NO
5952  025306  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
5953  025314  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
5954  025316  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
5955  025320  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
5956  025322  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
5957  025324  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
5958  025326  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
5959  025330  104401  025356                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
5960  025334  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
5961  025342  012616                          MOV     (SP)+,(SP)
5962  025344  000002                          RTI                     ;;RETURN TO USER
5963  025346  023420                  $DTBL:  10000.
5964  025350  001750                          1000.
5965  025352  000144                          100.
5966  025354  000012                          10.
5967  025356  000004                  $DBLK:  .BLKW   4
5968                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
5969
5970                                  ;;**************************************************************************
5971                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
5972                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
5973                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
5974                                  ;*CALL:
5975                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5976                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
5977                                  ;*      .BYTE   N               ;;N 1 TO 6 FOR NUMBER OF DIGITS TO TYPE
5978                                  ;*      .BYTE   M               ;;M=1 OR 0
5979                                  ;*                                      ;;1=TYPE LEADING ZEROS
5980                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
5981                                  ;*
5982                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
5983                                  ;*$TYPOS OR $TYPOC
5984                                  ;*CALL:
5985                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5986                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
5987                                  ;*
5988                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
5989                                  ;*CALL:
5990                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5991                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
5992
5993  025366  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
5994  025372  116637  000001  025611          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
5995  025400  112637  025613                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
5996  025404  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
5997  025410  000406                          BR      $TYPON
5998  025412  112737  000001  025611  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
5999  025420  112737  000006  025613          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
```

```
 6000  025426  112737  000005  025610  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 6001  025434  010346                          MOV     R3,-(SP)        ;;SAVE R3
 6002  025436  010446                          MOV     R4,-(SP)        ;;SAVE R4
 6003  025440  010546                          MOV     R5,-(SP)        ;;SAVE R5
 6004  025442  113704  025613                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 6005  025446  005404                          NEG     R4
 6006  025450  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 6007  025454  110437  025612                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 6008  025460  113704  025611                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 6009  025464  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 6010  025470  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
 6011  025472  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO 'C'
 6012  025474  000404                          BR      3$              ;;GO DO MSB
 6013  025476  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
 6014  025500  006105                          ROL     R5
 6015  025502  006105                          ROL     R5
 6016  025504  010503                          MOV     R5,R3
 6017  025506  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 6018  025510  105337  025612                  DECB    $OMODE          ;;TYPE THIS DIGIT?
 6019  025514  100016                          BPL     7$              ;;BR IF NO
 6020  025516  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
 6021  025522  001002                          BNE     4$              ;;TEST FOR 0
 6022  025524  005704                          TST     R4              ;;SUPPRESS THIS 0?
 6023  025526  001403                          BEQ     5$              ;;BR IF YES
 6024  025530  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 6025  025532  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 6026  025536  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 6027  025542  110337  025606                  MOVB    R3,8$           ;;SAVE FOR TYPING
 6028  025546  104401  025606                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
 6029  025552  105337  025610          7$:     DECB    $OCNT           ;;COUNT BY 1
 6030  025556  003347                          BGT     2$              ;;BR IF MORE TO DO
 6031  025560  002402                          BLT     6$              ;;BR IF DONE
 6032  025562  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 6033  025564  000744                          BR      2$              ;;GO DO THE LAST DIGIT
 6034  025566  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
 6035  025570  012604                          MOV     (SP)+,R4        ;;RESTORE R4
 6036  025572  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 6037  025574  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 6038  025602  012616                          MOV     (SP)+,(SP)
 6039  025604  000002                          RTI                     ;;RETURN
 6040  025606     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 6041  025607     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 6042  025610     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 6043  025611     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 6044  025612  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 6045                          .SBTTL  TYPE ROUTINE
 6046
 6047                          ;;******************************************************************
 6048                          ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 6049                          ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 6050                          ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 6051                          ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 6052                          ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 6053                          ;*
 6054                          ;*CALL:
 6055                          ;*1) USING A TRAP INSTRUCTION
```

```
6056                                    ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
6057                                    ;*OR
6058                                    ;*      TYPE
6059                                    ;*      MESADR
6060                                    ;*
6061
6062  025614  105737  001173            $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
6063  025620  100002                            BPL     1$              ;;BR IF YES
6064  025622  000000                            HALT                    ;;HALT HERE IF NO TERMINAL
6065  025624  000430                            BR      3$              ;;LEAVE
6066  025626  010046                    1$:     MOV     R0,-(SP)        ;;SAVE R0
6067  025630  017600  000002                    MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
6068  025634  122737  000001  001242            CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
6069  025642  001011                            BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
6070  025644  132737  000100  001243            BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
6071  025652  001405                            BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
6072  025654  010037  025664                    MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
6073  025660  004737  030512                    JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
6074  025664  000000                    61$:    .WORD   0               ;;MESSAGE ADDRESS
6075  025666  132737  000040  001243    62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
6076  025674  001003                            BNE     60$             ;;YES,SKIP TYPE OUT
6077  025676  112046                    2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
6078  025700  001005                            BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
6079  025702  005726                            TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
6080  025704  012600                    60$:    MOV     (SP)+,R0        ;;RESTORE R0
6081  025706  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN PC
6082  025712  000002                            RTI                     ;;RETURN
6083  025714  122716  000011            4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
6084  025720  001430                            BEQ     8$
6085  025722  122716  000200                    CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
6086  025726  001006                            BNE     5$
6087  025730  005726                            TST     (SP)+           ;;POP  <CR><LF> EQUIV
6088  025732  104401                            TYPE                    ;;TYPE A CR AND LF
6089  025734  001217                            $CRLF
6090  025736  105037  026072                    CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
6091  025742  000755                            BR      2$              ;;GET NEXT CHARACTER
6092  025744  004737  026026            5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
6093  025750  123726  001172            6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
6094  025754  001350                            BNE     2$              ;;IF NO GO GET NEXT CHAR.
6095  025756  013746  001170                    MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
6096                                            ;;AND THE NULL CHAR.
6097  025762  105366  000001            7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
6098  025766  002770                            BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
6099  025770  004737  026026                    JSR     PC,$TYPEC       ;;GO TYPE A NULL
6100  025774  105337  026072                    DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
6101  026000  000770                            BR      7$              ;;LOOP
6102
6103                                    ;HORIZONTAL TAB PROCESSOR
6104
6105  026002  112716  000040            8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
6106  026006  004737  026026            9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
6107  026012  132737  000007  026072            BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
6108  026020  001372                            BNE     9$              ;;TAB STOP
6109  026022  005726                            TST     (SP)+           ;;POP SPACE OFF STACK
6110  026024  000724                            BR      2$              ;;GET NEXT CHARACTER
6111  026026  105777  153132            $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
```

```
6112  026032  100375                          BPL     $TYPEC
6113  026034  116677  000002  153124          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
6114  026042  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
6115  026050  001003                          BNE     1$              ;;BRANCH IF NO
6116  026052  105037  026072                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
6117  026056  000406                          BR      $TYPEX          ;;EXIT
6118  026060  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
6119  026066  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
6120  026070  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
6121  026072  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
6122  026074  000207                  $TYPEX: RTS     PC
6123
6124                                  .SBTTL  SCOPE HANDLER ROUTINE
6125
6126                                  ;;**************************************************************************
6127                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
6128                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
6129                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
6130                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
6131                                  ;*SW14=1         LOOP ON TEST
6132                                  ;*SW11=1         INHIBIT ITERATIONS
6133                                  ;*SW09=1         LOOP ON ERROR
6134                                  ;*SW08=1         LOOP ON TEST IN SWR<7:0>
6135                                  ;*CALL
6136                                  ;*      SCOPE           ;;SCOPE=IOT
6137
6138  026076                          $SCOPE:
6139  026076  104410                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
6140  026100  032777  040000  153046  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
6141  026106  001131                          BNE     $OVER           ;;YES IF SW14=1
6142                                  ;#####START OF CODE FOR THE XOR TESTER#####
6143  026110  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
6144                                                                  ;;THIS INSTRUCTION TO A ''NOP'' (NOP 240)
6145  026112  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
6146  026116  012737  026136  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
6147  026124  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
6148  026130  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
6149  026134  000500                          BR      $SVLAD          ;;GO TO THE NEXT TEST
6150  026136  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
6151  026140  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
6152  026144  000440                          BR      7$              ;;LOOP ON THE PRESENT TEST
6153  026146                          6$:;#####END OF CODE FOR THE XOR TESTER#####
6154  026146  032777  000400  153000          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
6155  026154  001421                          BEQ     2$              ;;BR IF NO
6156  026156  005046                          CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
6157  026160  117716  152770                  MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
6158  026164  001414                          BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
6159  026166  022716  000024                  CMP     #24,(SP)        ;;CHECK THE NUMBER IN THE SWR
6160  026172  002411                          BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
6161  026174  011637  001116                  MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
6162  026200  005316                          DEC     (SP)            ;;BACKUP BY ONE
6163  026202  006316                          ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
6164  026204  062716  026410                  ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
6165  026210  013637  001122                  MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
6166  026214  000466                          BR      $OVER           ;;GO LOOP ON THE TEST
6167  026216  005726                  8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
```

```
6168  026220  105737  001117          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
6169  026224  001421                          BEQ     3$              ;;BR IF NO
6170  026226  123737  001131  001117          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
6171  026234  101015                          BHI     3$              ;;BR IF NO
6172  026236  032777  001000  152710          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
6173  026244  001404                          BEQ     4$              ;;BR IF NO
6174  026246  013737  001124  001122  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
6175  026254  000446                          BR      $OVER
6176  026256  105037  001117          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
6177  026262  005037  001206                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
6178  026266  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
6179  026270  032777  004000  152656  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
6180  026276  001011                          BNE     1$              ;;BR IF YES
6181  026300  005737  001230                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
6182  026304  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
6183  026306  005237  001120                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
6184  026312  023737  001206  001120          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
6185  026320  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
6186  026322  012737  000001  001120  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
6187  026330  013737  026406  001206          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
6188  026336  105237  001116          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
6189  026342  113737  001116  001226          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
6190  026350  011637  001122                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
6191  026354  011637  001124                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
6192  026360  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
6193  026364  112737  000001  001131          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
6194  026372  013777  001116  152556  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
6195  026400  013716  001122                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
6196  026404  000002                          RTI                     ;;FIXES PS
6197  026406  000012          $MXCNT: 10.                             ;;MAX. NUMBER OF ITERATIONS
6198  026410                  $SW08TBL:
6199  026410  004432                  .WORD   TST1+2                  ;;STARTING ADDRESS OF TEST 1
6200  026412  005006                  .WORD   TST2+2                  ;;STARTING ADDRESS OF TEST 2
6201  026414  005226                  .WORD   TST3+2                  ;;STARTING ADDRESS OF TEST 3
6202  026416  005442                  .WORD   TST4+2                  ;;STARTING ADDRESS OF TEST 4
6203  026420  005630                  .WORD   TST5+2                  ;;STARTING ADDRESS OF TEST 5
6204  026422  006766                  .WORD   TST6+2                  ;;STARTING ADDRESS OF TEST 6
6205  026424  010146                  .WORD   TST7+2                  ;;STARTING ADDRESS OF TEST 7
6206  026426  010326                  .WORD   TST10+2                 ;;STARTING ADDRESS OF TEST 10
6207  026430  010450                  .WORD   TST11+2                 ;;STARTING ADDRESS OF TEST 11
6208  026432  010750                  .WORD   TST12+2                 ;;STARTING ADDRESS OF TEST 12
6209  026434  011336                  .WORD   TST13+2                 ;;STARTING ADDRESS OF TEST 13
6210  026436  011762                  .WORD   TST14+2                 ;;STARTING ADDRESS OF TEST 14
6211  026440  012346                  .WORD   TST15+2                 ;;STARTING ADDRESS OF TEST 15
6212  026442  013060                  .WORD   TST16+2                 ;;STARTING ADDRESS OF TEST 16
6213  026444  013534                  .WORD   TST17+2                 ;;STARTING ADDRESS OF TEST 17
6214  026446  014006                  .WORD   TST20+2                 ;;STARTING ADDRESS OF TEST 20
6215  026450  014530                  .WORD   TST21+2                 ;;STARTING ADDRESS OF TEST 21
6216  026452  015510                  .WORD   TST22+2                 ;;STARTING ADDRESS OF TEST 22
6217  026454  016760                  .WORD   TST23+2                 ;;STARTING ADDRESS OF TEST 23
6218  026456  020204                  .WORD   TST24+2                 ;;STARTING ADDRESS OF TEST 24
6219                          .SBTTL  ERROR HANDLER ROUTINE
6220
6221                          ;;**************************************************************************
6222                          ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
6223                          ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
```

```
                                        ;*AND GO TO ERRTYP ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1        HALT ON ERROR
                                        ;*SW13=1        INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1        BELL ON ERROR
                                        ;*SW09=1        LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N-ERROR ITEM NUMBER

026460                                  $ERROR:
026460  104410                         7$:     CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
026462  105237  001117                         INCB    $ERFLG          ;;SET THE ERROR FLAG
026466  001775                         1$:     BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
026470  013777  001116  152460                 MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
026476  032777  002000  152450                 BIT     #BIT10,@SWR     ;;BELL ON ERROR?
026504  001402                                 BEQ     1$              ;;NO - SKIP
026506  104401  001212                         TYPE    ,$BELL          ;;RING BELL
026512  005237  001126                         INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
026516  011637  001132                         MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
026522  162737  000002  001132                 SUB     #2,$ERRPC
026530  117737  152376  001130                 MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
026536  032777  020000  152410                 BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
026544  001004                                 BNE     20$             ;;SKIP TYPEOUTS
026546  004737  021706                         JSR     PC,ERRTYP       ;;GO TO USER ERROR ROUTINE
026552  104401  001217                         TYPE    ,$CRLF
026556                                  20$:
026556  122737  000001  001242                 CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
026564  001007                                 BNE     2$              ;;NO,SKIP APT ERROR REPORT
026566  113737  001130  026600                 MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
026574  004737  030522                         JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
026600     000                  21$:    .BYTE   0
026601     000                          .BYTE   0
026602  000777                  22$:    BR      22$             ;;APT ERROR LOOP
026604  005777  152344                 2$:     TST     @SWR            ;;HALT ON ERROR
026610  100002                                 BPL     3$              ;;SKIP IF CONTINUE
026612  000000                                 HALT                    ;;HALT ON ERROR!
026614  104410                                 CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
026616  032777  001000  152330          3$:    BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
026624  001402                                 BEQ     4$              ;;BR IF NO
026626  013716  001124                         MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
026632  005737  001210                 4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
026636  001402                                 BEQ     5$              ;;BR IF NONE
026640  013716  001210                         MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
026644                                  5$:
026644  022737  021666  000042                 CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
026652  001001                                 BNE     6$              ;;BRANCH IF NO
026654  000000                                 HALT                    ;;YES
026656                                  6$:
026656  000002                                 RTI                     ;;RETURN
                                        .SBTTL  TTY INPUT ROUTINE

                                        ;;**************************************************************************
                                        .ENABL  LSB
026660  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
026662  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
026664  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
```

```
6280  026666  000001                  $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
6281          026667                  $TKQEND=.
6282          026670                  .EVEN
6283
6284                                  ;*TK INITIALIZE ROUTINE
6285                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
6286                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
6287                                  ;
6288                                  ;*CALL:
6289                                  ;*      JSR     PC,$TKINT
6290                                  ;*      RETURN
6291                                  ;
6292  026670  005037  026660          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
6293  026674  012737  026666  026662          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
6294  026702  013737  026662  026664          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
6295  026710  012737  026740  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
6296  026716  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;'BR' LEVEL 4
6297  026724  005777  152232                  TST     @$TKB           ;;CLEAR DONE FLAG
6298  026730  012777  000100  152222          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
6299  026736  000207                          RTS     PC              ;;RETURN TO CALLER
6300
6301                                  ;*TK SERVICE ROUTINE
6302                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
6303                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
6304                                  ;*IT IN THE QUEUE.
6305                                  ;*IF THE CHARACTER IS A 'CONTROL-C' (^C) $TKINT IS CALLED AND
6306                                  ;*UPON RETURN EXIT IS MADE TO THE 'CONTROL-C' RESTART ADDRESS (START)
6307                                  ;
6308  026740  117746  152216          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
6309  026744  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
6310  026750  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
6311  026754  001007                          BNE     1$              ;;BRANCH IF NO
6312  026756  104401  030054                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
6313  026762  004737  026670                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
6314  026766  005726                          TST     (SP)+           ;;CLEAN UP STACK
6315  026770  000137  002632                  JMP     START           ;;CONTROL C RESTART
6316  026774  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
6317  027000  001004                          BNE     2$              ;;BRANCH IF NO
6318  027002  022737  000176  001154          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
6319  027010  001500                          BEQ     6$              ;;GO TO SWR CHANGE
6320
6321  027012                          2$:
6322  027012  022737  000001  026660          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
6323  027020  001004                          BNE     3$              ;;BRANCH IF NO
6324  027022  104401  001212                  TYPE    ,$BELL          ;;RING THE TTY BELL
6325  027026  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
6326  027030  000451                          BR      5$              ;;EXIT
6327  027032  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
6328  027036  001021                          BNE     32$             ;;BRANCH IF NO
6329  027040  005077  152114                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
6330  027044  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
6331  027046  105777  152106          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
6332  027052  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
6333  027054  117746  152102                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
6334  027060  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
6335  027064  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
```

```
 6336  027070  001366                          BNE     31$             ;;BRANCH IF NO
 6337  027072  012777  000100  152060          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
 6338  027100  000002                          RTI                     ;;RETURN
 6339  027102  005237  026660          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
 6340  027106  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
 6341  027112  002405                          BLT     4$              ;;BRANCH IF YES
 6342  027114  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
 6343  027120  003002                          BGT     4$              ;;BRANCH IF YES
 6344  027122  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
 6345  027126  112677  177530          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
 6346  027132  005237  026662                  INC     $TKQIN          ;;UPDATE THE POINTER
 6347  027136  023727  026662  026667          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
 6348  027144  001003                          BNE     5$              ;;BRANCH IF NO
 6349  027146  012737  026666  026662          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
 6350  027154  000002                  5$:     RTI                     ;;RETURN
 6351
 6352                                  ;;**********************************************************************
 6353                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 6354                                  ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
 6355                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
 6356                                  ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
 6357  027156  022737  000176  001154  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
 6358  027164  001124                          BNE     15$             ;;EXIT IF NOT
 6359  027166  105777  151766                  TSTB    @$TKS           ;;IS A CHAR WAITING?
 6360  027172  100121                          BPL     15$             ;;IF NOT, EXIT
 6361  027174  117746  151762                  MOVB    @$TKB,-(SP)     ;;YES
 6362  027200  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
 6363  027204  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
 6364  027210  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
 6365                                                                  ;;AND EXIT
 6366
 6367                                  ;;**********************************************************************
 6368                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
 6369                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
 6370                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
 6371  027212  123727  001150  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 6372  027220  001674                          BEQ     2$              ;;BRANCH IF YES
 6373  027222  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
 6374  027224  004737  026670                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
 6375  027230  005077  151724                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
 6376  027234  112737  000001  001151          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOP
 6377
 6378  027242  104401  030066                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
 6379  027246  104401  030073          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 6380  027252  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
 6381  027256  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 6382  027260  104401  030104                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
 6383  027264  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
 6384  027266  005046                          CLR     -(SP)           ;;THE NEW SWR
 6385  027270  105777  151664          7$:     TSTB    @$TKS           ;;CHAR THERE?
 6386  027274  100375                          BPL     7$              ;;IF NOT TRY AGAIN
 6387
 6388  027276  117746  151660                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 6389  027302  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 6390
 6391  027306  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
```

```
6392  027312  001015                      BNE     9$              ;;BRANCH IF NOT
6393  027314  104401  030054              TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
6394  027320  062706  000006              ADD     #6,SP           ;;CLEAN UP STACK
6395  027324  123727  001151  000001      CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
6396  027332  001003                      BNE     8$              ;;BRANCH IF NO
6397  027334  012777  000100  151616      MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
6398  027342  000137  002632          8$: JMP     START           ;;CONTROL-C RESTART
6399
6400
6401  027346  021627  000025          9$. CMP     (SP),#25        ;;IS IT A CONTROL-U?
6402  027352  001005                      BNE     10$             ;;BRANCH IF NOT
6403  027354  104401  030061              TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
6404  027360  062706  000006          20$:ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
6405  027364  000737                      BR      19$             ;;LET'S TRY IT AGAIN
6406
6407
6408  027366  021627  000015          10$:CMP     (SP),#15        ;;IS IT A <CR>?
6409  027372  001022                      BNE     16$             ;;BRANCH IF NO
6410  027374  005766  000004              TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
6411  027400  001403                      BEQ     11$             ;;BRANCH IF YES
6412  027402  016677  000002  151544      MOV     2(SP),@SWR      ;;SAVE NEW SWR
6413  027410  062706  000006          11$:ADD     #6,SP           ;;CLEAR UP STACK
6414  027414  104401  001217          14$:TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
6415  027420  123727  001151  000001      CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
6416  027426  001003                      BNE     15$             ;;BRANCH IF NOT
6417  027430  012777  000100  151522      MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
6418  027436  000002          15$:RTI                             ;;RETURN
6419  027440  004737  026026          16$:JSR     PC,$TYPEC       ;;ECHO CHAR
6420  027444  021627  000060              CMP     (SP),#60        ;;CHAR < 0?
6421  027450  002420                      BLT     18$             ;;BRANCH IF YES
6422  027452  021627  000067              CMP     (SP),#67        ;;CHAR > 7?
6423  027456  003015                      BGT     18$             ;;BRANCH IF YES
6424  027460  042726  000060              BIC     #60,(SP)+       ;;STRIP-OFF ASCII
6425  027464  005766  000002              TST     2(SP)           ;;IS THIS THE FIRST CHAR
6426  027470  001403                      BEQ     17$             ;;BRANCH IF YES
6427  027472  006316                      ASL     (SP)            ;;NO, SHIFT PRESENT
6428  027474  006316                      ASL     (SP)            ;;   CHAR OVER TO MAKE
6429  027476  006316                      ASL     (SP)            ;;   ROOM FOR NEW ONE.
6430  027500  005266  000002          17$:INC     2(SP)           ;;KEEP COUNT OF CHAR
6431  027504  056616  177776              BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
6432  027510  000667                      BR      7$              ;;GET THE NEXT ONE
6433  027512  104401  001216          18$:TYPE    ,$QUES          ;;TYPE ?<CR><LF>
6434  027516  000720                      BR      20$             ;;SIMULATE CONTROL-U
6435                          .DSABL  LSB
6436
6437
6438                          ;;*******************************************************************
6439                          ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
6440                          ;*CALL:
6441                          ;*      RDCHR                       ;;GET A CHARACTER FROM THE QUEUE
6442                          ;*      RETURN HERE                 ;;CHARACTER IS ON THE STACK
6443                          ;*                                  ;;WITH PARITY BIT STRIPPED OFF
6444                          ;
6445
6446  027520  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
6447  027522  016666  000004  000002      MOV     4(SP),2(SP)     ;;THE PS
```

```
6448  027530  005066  000004          CLR     4(SP)           ;;GET READY FOR A CHARACTER
6449  027534  005046                  CLR     -(SP)           ;;PUT NEW PS ON STACK
6450  027536  012746  027544          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
6451  027542  000002                  RTI                     ;;POP NEW PC AND PS
6452  027544                  64$:
6453  027544  005737  026660  1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
6454  027550  001775                  BEQ     1$
6455  027552  005337  026660          DEC     $TKCNT          ;;DECREMENT THE COUNTER
6456  027556  117766  177102  000004  MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
6457  027564  005237  026664          INC     $TKQOUT         ;;UPDATE THE POINTER
6458  027570  023727  026664  026667  CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
6459  027576  001003                  BNE     2$              ;;BRANCH IF NO
6460  027600  012737  026666  026664  MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
6461  027606  000002          2$:     RTI                     ;;RETURN
6462                          ;;**************************************************************
6463                          ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
6464                          ;*CALL:
6465                          ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
6466                          ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
6467                          ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
6468
6469  027610  010346  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
6470  027612  005046          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
6471  027614  012703  030044  1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
6472  027620  022703  030054  2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
6473  027624  101456                  BLOS    4$              ;;BR IF YES
6474  027626  104411                  RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
6475  027630  112613                  MOVB    (SP)+,(R3)      ;;GET CHARACTER
6476  027632  122713  000177  10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
6477  027636  001022                  BNE     5$              ;;BR IF NO
6478  027640  005716                  TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
6479  027642  001007                  BNE     6$              ;;BR IF NO
6480  027644  112737  000134  030042  MOVB    #'\,9$          ;;TYPE A BACK SLASH
6481  027652  104401  030042          TYPE    ,9$
6482  027656  012716  177777          MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
6483  027662  005303          6$:     DEC     R3              ;;BACKUP BY ONE
6484  027664  020327  030044          CMP     R3,#$TTYIN      ;;STACK EMPTY?
6485  027670  103434                  BLO     4$              ;;BR IF YES
6486  027672  111337  030042          MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
6487  027676  104401  030042          TYPE    ,9$             ;;GO TYPE
6488  027702  000746                  BR      2$              ;;GO READ ANOTHER CHAR.
6489  027704  005716          5$:     TST     (SP)            ;;RUBOUT KEY SET?
6490  027706  001406                  BEQ     7$              ;;BR IF NO
6491  027710  112737  000134  030042  MOVB    #'\,9$          ;;TYPE A BACK SLASH
6492  027716  104401  030042          TYPE    ,9$
6493  027722  005016                  CLR     (SP)            ;;CLEAR THE RUBOUT KEY
6494  027724  122713  000025  7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
6495  027730  001003                  BNE     8$              ;;BR IF NO
6496  027732  104401  030061          TYPE    ,$CNTLU         ;;TYPE A CONTROL 'U'
6497  027736  000726                  BR      1$              ;;GO START OVER
6498  027740  122713  000022  8$:     CMPB    #22,(R3)        ;;IS CHARACTER A '^R'?
6499  027744  001011                  BNE     3$              ;;BRANCH IF NO
6500  027746  105013                  CLRB    (R3)            ;;CLEAR THE CHARACTER
6501  027750  104401  001217          TYPE    ,$CRLF          ;;TYPE A 'CR' & 'LF'
6502  027754  104401  030044          TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
6503  027760  000717                  BR      2$              ;;GO PICKUP ANOTHER CHACTER
```

```
6504  027762  104401  001216          4$:     TYPE    ,$QUES          ;;TYPE A '?'
6505  027766  000712                  BR      1$              ;;CLEAR THE BUFFER AND LOOP
6506  027770  111337  030042          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
6507  027774  104401  030042                  TYPE    ,9$
6508  030000  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
6509  030004  001305                  BNE     2$              ;;LOOP IF NOT RETURN
6510  030006  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
6511  030012  104401  001220                  TYPE    ,$LF            ;;TYPE A LINE FEED
6512  030016  005726                  TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
6513  030020  012603                  MOV     (SP)+,R3        ;;RESTORE R3
6514  030022  011646                  MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
6515  030024  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
6516  030032  012766  030044  000004          MOV     #$TTYIN,4(SP)
6517  030040  000002                  RTI                     ;;RETURN
6518  030042     000          9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
6519  030043     000                  .BYTE   0               ;;TERMINATOR
6520  030044  000010          $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
6521  030054  041536  005015     000  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
6522  030061     136  006525  000012  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
6523  030066  043536  005015     000  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
6524  030073     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
6525  030100  036440  000040
6526  030104  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
6527  030112  020075     000
6528          030116                  .EVEN
6529                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
6530
6531                                  ;;**********************************************************************
6532                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
6533                                  ;*CHANGE IT TO BINARY.
6534                                  ;*CALL:
6535                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
6536                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
6537                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
6538
6539  030116  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
6540  030120  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
6541  030126  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6542  030130  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6543  030132  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6544  030134  104412          1$:     RDLIN                   ;;READ AN ASCIZ LINE
6545  030136  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
6546  030140  005001                  CLR     R1              ;;CLEAR DATA WORD
6547  030142  005002                  CLR     R2
6548  030144  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
6549  030146  001412                  BEQ     3$              ;;IF ZERO GET OUT
6550  030150  006301                  ASL     R1              ;;*2
6551  030152  006102                  ROL     R2
6552  030154  006301                  ASL     R1              ;;*4
6553  030156  006102                  ROL     R2
6554  030160  006301                  ASL     R1              ;;*8
6555  030162  006102                  ROL     R2
6556  030164  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
6557  030170  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
6558  030172  000764                  BR      2$              ;;LOOP
6559  030174  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
```

```
6560  030176  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
6561  030202  010237  030216          MOV     R2,$HIOCT
6562  030206  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
6563  030210  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
6564  030212  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
6565  030214  000002                  RTI                     ;;RETURN
6566  030216  000000          $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
6567                          .SBTTL  TRAP DECODER
6568
6569                          ;;*****************************************************************
6570                          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE ''TRAP'' INSTRUCTION
6571                          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
6572                          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
6573                          ;*GO TO THAT ROUTINE.
6574
6575  030220  016646  000002  $TRAP:  MOV     2(SP),-(SP)     ;;ASSUME THE STATUS OF
6576  030224  042716  000020          BIC     #20,(SP)        ;; THE CALLER--DO NOT ALLOW
6577  030230  012746  030236          MOV     #1$,-(SP)       ;; T-BIT TRAPS
6578  030234  000002                  RTI                     ;;SET THE NEW STATUS
6579  030236  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
6580  030240  016600  000002          MOV     2(SP),R0        ;;GET TRAP ADDRESS
6581  030244  005740                  TST     -(R0)           ;;BACKUP BY 2
6582  030246  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
6583  030250  006300                  ASL     R0              ;;POSITION FOR INDEXING
6584  030252  016000  030272          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
6585  030256  000200                  RTS     R0              ;;GO TO ROUTINE
6586
6587
6588                          ;;THIS IS USE TO HANDLE THE ''GETPRI'' MACRO
6589
6590  030260  011646          $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
6591  030262  016666  000004  000002  MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
6592  030270  000002                  RTI                     ;;RESTORE THE PSW
6593
6594                          .SBTTL  TRAP TABLE
6595
6596                          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
6597                          ;*BY THE ''TRAP'' INSTRUCTION.
6598
6599                          ;       ROUTINE
6600                          ;       -------
6601  030272  030260  $TRPAD: .WORD   $TRAP2
6602  030274  025614          $TYPE   ;;CALL-TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
6603  030276  025412          $TYPOC  ;;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
6604  030300  025366          $TYPOS  ;;CALL=TYPOS     TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
6605  030302  025426          $TYPON  ;;CALL=TYPON     TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
6606  030304  025142          $TYPDS  ;;CALL=TYPDS     TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
6607  030306  025066          $TYPBN  ;;CALL=TYPBN     TRAP+6(104406)  TYPE BINARY (ASCII) NUMBER
6608
6609  030310  027246          $GTSWR  ;;CALL=GTSWR     TRAP+7(104407)  GET SOFT-SWR SETTING
6610
6611  030312  027156          $CKSWR  ;;CALL=CKSWR     TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
6612  030314  027520          $RDCHR  ;;CALL=RDCHR     TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
6613  030316  027610          $RDLIN  ;;CALL=RDLIN     TRAP+12(104412) TTY TYPEIN STRING ROUTINE
6614  030320  030116          $RDOCT  ;;CALL=RDOCT     TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
6615  030322  024772          $SAVREG ;;CALL-SAVREG    TRAP+14(104414) SAVE R0-R5 ROUTINE
```

```
                                        $RESREG ;;CALL=RESREG   TRAP+15(104415) RESTORE R0-R5 ROUTINE
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;**********************************************************************
                                ;POWER DOWN ROUTINE
030326  012737  030466  000024  $PWRDN: MOV     #$ILLLP,@#PWRVEC ;;SET FOR FAST UP
030334  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
030342  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
030344  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
030346  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
030350  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
030352  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
030354  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
030356  017746  150572                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
030362  010637  030472                MOV     SP,$SAVR6       ;;SAVE SP
030366  012737  030400  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
030374  000000                        HALT
030376  000776                        BR      .-2             ;;HANG UP

                                ;;**********************************************************************
                                ;POWER UP ROUTINE
030400  012737  030466  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
030406  013706  030472                MOV     $SAVR6,SP       ;;GET SP
030412  005037  030472                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
030416  005237  030472          1$:   INC     $SAVR6          ;;WAIT FOR THE INC
030422  001375                        BNE     1$              ;;OF  WORD
030424  012677  150524                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
030430  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
030432  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
030434  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
030436  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
030440  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
030442  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
030444  012737  030326  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
030452  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
030460  104401                        TYPE                    ;REPORT THE POWER FAILURE
030462  030474          $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
030464  000002                        RTI
030466  000000          $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
030470  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
030472  000000          $SAVR6: 0                       ;;PUT THE SP HERE
030474  005015  047520  042527  $POWER: .ASCIZ  <15><12>'POWER''
030502  000122
                                        .EVEN
                                .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;**********************************************************************
030504  112737  000001  030750  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
030512  112737  000001  030746  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
030520  000403                        BR      $ATYC
030522  112737  000001  030750  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
030530                          $ATYC:
030530  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
030532  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
030534  105737  030746                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
030540  001450                        BEQ     5$              ;;IF NOT:  BR
```

```
6672  030542  122737  000001  001242          CMPB    #APTENV,$ENV      ;;OPERATING UNDER APT?
6673  030550  001031                          BNE     3$                ;;IF NOT:  BR
6674  030552  132737  000100  001243          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
6675  030560  001425                          BEQ     3$                ;;IF NOT:  BR
6676  030562  017600  000004                  MOV     @4(SP),R0         ;;GET MESSAGE ADDR.
6677  030566  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDR.
6678  030574  005737  001222          1$:     TST     $MSGTYPE          ;;SEE IF DONE W/ LAST XMISSION?
6679  030600  001375                          BNE     1$                ;;IF NOT:  WAIT
6680  030602  010037  001236                  MOV     R0,$MSGAD         ;;PUT ADDR IN MAILBOX
6681  030606  105720                  2$:     TSTB    (R0)+             ;;FIND END OF MESSAGE
6682  030610  001376                          BNE     2$
6683  030612  163700  001236                  SUB     $MSGAD,R0         ;;SUB START OF MESSAGE
6684  030616  006200                          ASR     R0                ;;GET MESSAGE LNGTH IN WORDS
6685  030620  010037  001240                  MOV     R0,$MSGLGT        ;;PUT LENGTH IN MAILBOX
6686  030624  012737  000004  001222          MOV     #4,$MSGTYPE       ;;TELL APT TO TAKE MSG.
6687  030632  000413                          BR      5$
6688  030634  017637  000004  030660  3$:     MOV     @4(SP),4$         ;;PUT MSG ADDR IN JSR LINKAGE
6689  030642  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDRESS
6690  030650  013746  177776                  MOV     177776,-(SP)      ;;PUSH 177776 ON STACK
6691  030654  004737  025614                  JSR     PC,$TYPE          ;;CALL TYPE MACRO
6692  030660  000000                  4$:     .WORD   0
6693  030662                          5$:
6694  030662  105737  030750          10$:    TSTB    $FFLG             ;;SHOULD REPORT FATAL ERROR?
6695  030666  001416                          BEQ     12$               ;;IF NOT:  BR
6696  030670  005737  001242                  TST     $ENV              ;;RUNNING UNDER APT?
6697  030674  001413                          BEQ     12$               ;;IF NOT:  BR
6698  030676  005737  001222          11$:    TST     $MSGTYPE          ;;FINISHED LAST MESSAGE?
6699  030702  001375                          BNE     11$               ;;IF NOT:  WAIT
6700  030704  017637  000004  001224          MOV     @4(SP),$FATAL     ;;GET ERROR #
6701  030712  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDR.
6702  030720  005237  001222                  INC     $MSGTYPE          ;;TELL APT TO TAKE ERROR
6703  030724  105037  030750          12$:    CLRB    $FFLG             ;;CLEAR FATAL FLAG
6704  030730  105037  030747                  CLRB    $LFLG             ;;CLEAR LOG FLAG
6705  030734  105037  030746                  CLRB    $MFLG             ;;CLEAR MESSAGE FLAG
6706  030740  012601                          MOV     (SP)+,R1          ;;POP STACK INTO R1
6707  030742  012600                          MOV     (SP)+,R0          ;;POP STACK INTO R0
6708  030744  000207                          RTS     PC                ;;RETURN
6709  030746     000          $MFLG:  .BYTE   0                 ;;MESSG. FLAG
6710  030747     000          $LFLG:  .BYTE   0                 ;;LOG FLAG
6711  030750     000          $FFLG:  .BYTE   0                 ;;FATAL FLAG
6712             030752                  .EVEN
6713             000200          APTSIZE=200
6714             000001          APTENV=001
6715             000100          APTSPOOL=100
6716             000040          APTCSUP=040
6717
```

```
                                        .SBTTL  CONSOLE MESSAGES

                                        .NLIST  BEX
030752    051                           CLSPRN: .ASCII  @)@
030753    075     000                   EQUALS: .ASCIZ  @-@
030755    015  025012     000           PROMPT: .ASCIZ  <CR><LF>@*@
030761    077     000                   QSTMRK: .ASCIZ  @?@
030763                                    HELPQST:
030763    015  052012  050131            .ASCIZ  <CR><LF>@TYPE HELP TEXT (Y OR N)??@
031017                                    UBUSQST:
031017    015  041412  040510            .ASCIZ  <CR><LF>@CHANGE RM03 UNIBUS ADDRESS OR VECTOR ADDRESS (Y OR N)<CR> ??@
031116 005015  051525  020105            CNSL00: .ASCIZ  <CR><LF>@USE SAME DEVICES (Y OR N) ??@
031155    015  051012  030115            CNSL01: .ASCIZ  <CR><LF>@RM03 BUS ADDRESS (@
031202 005015  047105  051124            CNSL02: .ASCII  <CR><LF>@ENTRY NOT IN I/O PAGE@
031231    015  040412  042104                    .ASCIZ  <CR><LF>@ADDRESS MUST BE >160000@
031263    015  051012  030115            CNSL03: .ASCIZ  <CR><LF>@RM03 VECTOR ADDRESS (@
031313    015  042412  052116            CNSL04: .ASCII  <CR><LF>@ENTRY OUT OF RANGE@
031337    015  040412  042104                    .ASCIZ  <CR><LF>@ADDRESS MUST BE <1000@
031367    015  051012  030115            CNSL05: .ASCIZ  <CR><LF>@RM03 INTERRUPT PRIORITY (@
031423    015  042412  052116            CNSL06: .ASCIZ  <CR><LF>@ENTRY OUT OF RANGE@
031450                                    CNSL07:
031450 005015  054524  042520            .ASCII  <CR><LF>@TYPE (A) TO TEST ALL DEVICES, OR TYPE DEVICE@
031526 047040  046525  042502            .ASCII  @ NUMBER(S)@
031540 005015  042524  046522            .ASCIZ  <CR><LF>@TERMINATE INPUT WITH CARRIAGE RETURN@
031607    015  047012  052117            NOTEX:  .ASCIZ  <CR><LF>/NOT EXIST DRIVE  /
                                        .LIST BEX

031634                                  .EVEN
```

```
6723
6724  031634                        FNCDTB:                         ;FUNCTION CODE TABLE
6725
6726  031634  020000                .WORD   OPI                     ;NOP
6727  031636  130001                .WORD   OPI!ATA.ILF.IVC         ;ILLEGAL FUNCTION (2)
6728  031640  132000                .WORD   ATA!OPI.IVC!IAE         ;SEEK
6729  031642  130000                .WORD   ATA!OPI!IVC             ;RECALIBRATE
6730  031644  020000                .WORD   OPI                     ;DRIVE CLEAR
6731  031646  030000                .WORD   OPI!IVC                 ;RELEASE
6732  031650  130000                .WORD   OPI!ATA!IVC             ;OFFSET
6733  031652  130000                .WORD   OPI!ATA!IVC             ;RETURN TO CENTERLINE
6734  031654  020000                .WORD   OPI                     ;READ IN PRESET
6735  031656  020000                .WORD   OPI                     ;PACK ACKNOWLEDGE
6736  031660  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (24)
6737  031662  130001                .WORD   OPI!ATA.ILF!IVC         ;ILLEGAL FUNCTION (26)
6738  031664  132000                .WORD   ATA!OPI!IVC!IAE         ;SEARCH
6739  031666  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (32)
6740  031670  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (34)
6741  031672  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (36)
6742  031674  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (40)
6743  031676  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (42)
6744  031700  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (44)
6745  031702  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (46)
6746  031704  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK DATA
6747  031706  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK HEADER AND DATA
6748  031710  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (54)
6749  031712  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (56)
6750  031714  037200                .WORD   OPI!.VC!WLE!IAE!AOE!HCE ;WRITE DATA
6751  031716  037000                .WORD   OPI!IVC!WLE!IAE!AOE     ;WRITE HEADER AND DATA
6752  031720  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (64)
6753  031722  130001                .WORD   OPI!ATA!ILF!IVC         ;ILLEGAL FUNCTION (66)
6754  031724  033300                .WORD   OPI!IVC!IAE!AOE.HCE!ECH ;READ DATA
6755  031726  033300                .WORD   OPI!IVC!IAE!AOE.HCE!ECH ;READ HEADER AND DATA
6756  031730  130001                .WORD   OPI!ATA.ILF!IVC         ;ILLEGAL FUNCTION (74)
6757  031732  130001                .WORD   OPI!ATA!ILF.IVC         ;ILLEGAL FUNCTION (76)
6758
6759  031734      001   ATNTBL: .BYTE   1.
6760  031735      002           .BYTE   2.
6761  031736      004           .BYTE   4.
6762  031737      010           .BYTE   8.
6763  031740      020           .BYTE   16.
6764  031741      040           .BYTE   32.
6765  031742      100           .BYTE   64.
6766  031743      200           .BYTE   128.
6767
```

```
 6768                                        .EVEN
 6769
 6770  031744  046454  040042  000000  EMT1:   .WORD   EMS300,EMS1,0
 6771  031752  046472  046515  046542  EMT2:   .WORD   EMS301,EMS302,EMS303,EMS1,EMS304
 6772  031760  040042  046553
 6773  031764  051724  051145  051305          .WORD   EMS511,EMS500,EMS501,EMS502,EMS503,0
 6774  031772  051332  051401  000000
 6775  032000  046472  046562  046515  EMT3:   .WORD   EMS301,EMS306,EMS302
 6776  032006  051724  051451  051305          .WORD   EMS511,EMS505,EMS501,EMS502,0
 6777  032014  051332  000000
 6778  032020  046454  046515  046576  EMT4:   .WORD   EMS300,EMS302,EMS307,EMS2
 6779  032026  040113
 6780  032030  051724  051332  051305          .WORD   EMS511,EMS502,EMS501,EMS503,0
 6781  032036  051401  000000
 6782  032042  046472  046637  046654  EMT5:   .WORD   EMS301,EMS310,EMS311
 6783  032050  051724  051332  051305          .WORD   EMS511,EMS502,EMS501,EMS503,EMS504
 6784  032056  051401  051425
 6785  032062  046720  000000                  .WORD   EMS312,0
 6786  032066  046472  046562  046654  EMT6:   .WORD   EMS301,EMS306,EMS311
 6787  032074  051724  051332  051305          .WORD   EMS511,EMS502,EMS501,EMS503,EMS504,0
 6788  032102  051401  051425  000000
 6789  032110  046472  046756  046515  EMT7:   .WORD   EMS301,EMS313,EMS302
 6790  032116  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,EMS504,EMS503,0
 6791  032124  051425  051401  000000
 6792  032132  047064  047105  047007  EMT10:  .WORD   EMS316,EMS317,EMS314
 6793  032140  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
 6794  032146  000000
 6795  032150  047064  047105  047036  EMT11:  .WORD   EMS316,EMS317,EMS315
 6796  032156  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
 6797  032164  000000
 6798  032166  047064  047125  047007  EMT12:  .WORD   EMS316,EMS320,EMS314
 6799  032174  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
 6800  032202  000000
 6801  032204  047064  047125  047036  EMT13:  .WORD   EMS316,EMS320,EMS315
 6802  032212  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
 6803  032220  000000
 6804  032222  047064  047145  047007  EMT14:  .WORD   EMS316,EMS321,EMS314
 6805  032230  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
 6806  032236  000000
 6807  032240  047064  047145  047036  EMT15:  .WORD   EMS316,EMS321,EMS315
 6808  032246  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
 6809  032254  000000
 6810  032256  047064  047165  047007  EMT16:  .WORD   EMS316,EMS322,EMS314
 6811  032264  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
 6812  032272  000000
 6813  032274  047064  047165  047036  EMT17:  .WORD   EMS316,EMS322,EMS315
 6814  032302  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
 6815  032310  000000
 6816  032312  044446  047404  047462  EMT20:  .WORD   EMS71,EMS335,EMS340,EMS72,EMS377,EMS372
 6817  032320  044525  050504  050365
 6818  032326  051724  051401  000000          .WORD   EMS511,EMS503,0
 6819  032334  044446  047427  047462  EMT21:  .WORD   EMS71,EMS336,EMS340,EMS72,EMS400,EMS372
 6820  032342  044525  050515  050365
 6821  032350  051724  051401  000000          .WORD   EMS511,EMS503,0
 6822  032356  044602  047707  044652  EMT22:  .WORD   EMS73,EMS352,EMS74,EMS402,EMS70,EMS406
 6823  032364  050547  044400  050646
```

```
 6824  032372  051724  051401  051425          .WORD   EMS511,EMS503,EMS504,0
 6825  032400  000000
 6826  032402  044602  047707  044725  EMT23:  .WORD   EMS73,EMS352,EMS75,EMS402,EMS77,EMS406
 6827  032410  050547  045042  050646
 6828  032416  051724  051401  051425          .WORD   EMS511,EMS503,EMS504,0
 6829  032424  000000
 6830  032426  044602  050024  044725  EMT24:  .WORD   EMS73,EMS356,EMS75,EMS402,EMS77,EMS377
 6831  032434  050547  045042  050504
 6832  032442  051724  051401  051425          .WORD   EMS511,EMS503,EMS504,0
 6833  032450  000000
 6834  032452  044602  047404  047462  EMT25:  .WORD   EMS73,EMS335,EMS340,EMS76,EMS411
 6835  032460  045001  050710
 6836  032464  051724  051401  000000          .WORD   EMS511,EMS503,0
 6837  032472  046472  046562  046044  EMT26:  .WORD   EMS301,EMS306,EMS252,EMS253,EMS327,EMS254
 6838  032500  046077  047265  046132
 6839  032506  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,EMS502
 6840  032514  051332
 6841  032516  047273  047007  000000          .WORD   EMS330,EMS314,0
 6842  032524  047446  044602  050701  EMT27:  .WORD   EMS337,EMS73,EMS410,EMS76,EMS411
 6843  032532  045001  050710
 6844  032536  051724  051401  000000          .WORD   EMS511,EMS503,0
 6845  032544  047446  045105  047473  EMT30:  .WORD   EMS337,EMS100,EMS341,EMS101
 6846  032552  045145
 6847  032554  051724  051401  051425          .WORD   EMS511,EMS503,EMS504,0
 6848  032562  000000
 6849  032564  046454  046044          EMT31:  .WORD   EMS300,EMS252
 6850  032570  051724  051305  051401          .WORD   EMS511,EMS501,EMS503,0
 6851  032576  000000
 6852  032600  045105  050321          EMT32:  .WORD   EMS100,EMS370
 6853  032604  051724  051425  000000          .WORD   EMS511,EMS504,0
 6854  032612  045105  050726          EMT33:  .WORD   EMS100,EMS412
 6855  032616  051724  051425  000000          .WORD   EMS511,EMS504,0
 6856  032624  045221  050726          EMT34:  .WORD   EMS102,EMS412
 6857  032630  051724  051401  000000          .WORD   EMS511,EMS503,0
 6858  032636  045221  047427          EMT35:  .WORD   EMS102,EMS336
 6859  032642  051724  051401  000000          .WORD   EMS511,EMS503,0
 6860  032650  045105  047404  047462  EMT36:  .WORD   EMS100,EMS335,EMS340,EMS102,EMS334
 6861  032656  045221  047374
 6862  032662  051724  051425  000000          .WORD   EMS511,EMS504,0
 6863  032670  045105  047404  047462  EMT37:  .WORD   EMS100,EMS335,EMS340,EMS102,EMS377,EMS365
 6864  032676  045221  050504  050221
 6865  032704  050742  045327  050776          .WORD   EMS413,EMS104,EMS415
 6866  032712  051724  051425  000000          .WORD   EMS511,EMS504,0
 6867  032720  045105  047427  047462  EMT40:  .WORD   EMS100,EMS336,EMS340,EMS416,EMS104,EMS415
 6868  032726  051016  045327  050776
 6869  032734  051724  051425  000000          .WORD   EMS511,EMS504,0
 6870  032742  045105  047427  047462  EMT41:  .WORD   EMS100,EMS336,EMS340,EMS73,EMS415,EMS402
 6871  032750  044602  050776  050547
 6872  032756  045221  050515                  .WORD   EMS102,EMS400
 6873  032762  051724  051425  000000          .WORD   EMS511,EMS504,0
 6874  032770  045267  050532  047265  EMT42:  .WORD   EMS103,EMS401,EMS327,EMS370
 6875  032776  050321
 6876  033000  051724  051401  000000          .WORD   EMS511,EMS503,0
 6877  033006  050756  045327  051030  EMT43:  .WORD   EMS414,EMS104,EMS417
 6878  033014  051724  051425  000000          .WORD   EMS511,EMS504,0
 6879  033022  045354  050532  047265  EMT44:  .WORD   EMS105,EMS401,EMS327,EMS370
```

```
6880  033030  050321
6881  033032  051724  051401  000000          .WORD   EMS511,EMS503,0
6882  033040  046454  046077          EMT45:  .WORD   EMS300,EMS253
6883  033044  051724  051305  051401          .WORD   EMS511,EMS501,EMS503,0
6884  033052  000000
6885  033054  045415  050532  051030  EMT46:  .WORD   EMS106,EMS401,EMS417
6886  033062  051724  051401  000000          .WORD   EMS511,EMS503,0
6887  033070  050742  045462  051047  EMT47:  .WORD   EMS413,EMS107,EMS420,EMS417
6888  033076  051030
6889  033100  051724  051425  000000          .WORD   EMS511,EMS504,0
6890  033106  045472  050742  050636  EMT50:  .WORD   EMS110,EMS413,EMS405,EMS107
6891  033114  045462
6892  033116  051724  051401  000000          .WORD   EMS511,EMS503,0
6893  033124  047534  045531  051062  EMT51:  .WORD   EMS343,EMS111,EMS421
6894  033132  051724  051425  000000          .WORD   EMS511,EMS504,0
6895  033140  050756  051016  045462  EMT52:  .WORD   EMS414,EMS416,EMS107,EMS421
6896  033146  051062
6897  033150  051724  051425  000000          .WORD   EMS511,EMS504,0
6898  033156  047312  040227  046235  EMT53:  .WORD   EMS331,EMS4,EMS256
6899  033164  051724  051305  000000          .WORD   EMS511,EMS501,0
6900  033172  045547  050742  050636  EMT54:  .WORD   EMS112,EMS413,EMS405,EMS607
6901  033200  052240
6902  033202  051724  051401  000000          .WORD   EMS511,EMS503,0
6903  033210  045605  050532  047265  EMT55:  .WORD   EMS113,EMS401,EMS327,EMS370
6904  033216  050321
6905  033220  051724  051401  000000          .WORD   EMS511,EMS503,0
6906  033226  045644  050532  051030  EMT56:  .WORD   EMS114,EMS401,EMS417
6907  033234  051724  051401  000000          .WORD   EMS511,EMS503,0
6908
6909  033242  046265  047342          EMT57:  .WORD   EMS257,EMS332
6910  033246  051724  051524  051305          .WORD   EMS511,EMS506,EMS501,0
6911  033254  000000
6912
6913  033256  040260  047360          EMT60:  .WORD   EMS5,EMS333
6914  033262  051724  051561  051305          .WORD   EMS511,EMS507,EMS501,0
6915  033270  000000
6916
6917  033272  050742  045712  051047  EMT61:  .WORD   EMS413,EMS115,EMS420,EMS417
6918  033300  051030
6919  033302  051724  051425  000000          .WORD   EMS511,EMS504,0
6920  033310  047625  045105  047473  EMT62:  .WORD   EMS346,EMS100,EMS341,EMS101
6921  033316  045145
6922  033320  051724  051425  000000          .WORD   EMS511,EMS504,0
6923  033326  045605  050336          EMT63:  .WORD   EMS113,EMS371
6924  033332  051724  051401  000000          .WORD   EMS511,EMS503,0
6925  033340  050742  045727  047723  EMT64:  .WORD   EMS413,EMS116,EMS353
6926  033346  051724  051425  000000          .WORD   EMS511,EMS504,0
6927  033354  051077  047723          EMT65:  .WORD   EMS422,EMS353
6928  033360  051724  051425  000000          .WORD   EMS511,EMS504,0
6929
6930  033366  040601  047427  047502  EMT66:  .WORD   EMS12,EMS336,EMS342
6931  033374  051724  051305  000000          .WORD   EMS511,EMS501,0
6932
6933  033402  046454  040330          EMT67:  .WORD   EMS300,EMS6
6934  033406  051724  051305                  .WORD   EMS511,EMS501
6935  033412  040374  047360  000000          .WORD   EMS7,EMS333,0
```

```
6936
6937  033420  046454  040330  047265  EMT70:  .WORD   EMS300,EMS6,EMS327,EMS7
6938  033426  040374
6939  033430  051724  051305  051425          .WORD   EMS511,EMS501,EMS504,0
6940  033436  000000
6941
6942  033440  040330  047404  047462  EMT71:  .WORD   EMS6,EMS335,EMS340,EMS10,EMS333,EMS342
6943  033446  040445  047360  047502
6944  033454  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
6945  033462  000000
6946
6947  033464  040330  047427  047462  EMT72:  .WORD   EMS6,EMS336,EMS340,EMS10,EMS334,EMS342
6948  033472  040445  047374  047502
6949  033500  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
6950  033506  000000
6951
6952  033510  046472  046321  046542  EMT73:  .WORD   EMS301,EMS260,EMS303,EMS11
6953  033516  040510
6954  033520  051724  051305  051332          .WORD   EMS511,EMS501,EMS502,0
6955  033526  000000
6956
6957  033530  047534  047550  047502  EMT74:  .WORD   EMS343,EMS344,EMS342,0
6958  033536  000000
6959
6960  033540  046454  040657          EMT75:  .WORD   EMS300,EMS13
6961  033544  051724  051401  000000          .WORD   EMS511,EMS503,0
6962
6963  033552  047625  040657  047577  EMT76:  .WORD   EMS346,EMS13,EMS345
6964  033560  051724  051401  000000          .WORD   EMS511,EMS503,0
6965
6966  033566  047446  040657  047577  EMT77:  .WORD   EMS337,EMS13,EMS345
6967  033574  051724  051401  000000          .WORD   EMS511,EMS503,0
6968
6969  033602  046472  046756  046132  EMT100: .WORD   EMS301,EMS313,EMS254,EMS347,EMS13
6970  033610  047643  040657
6971  033614  051724  051401  000000          .WORD   EMS511,EMS503,0
6972
6973  033622  047625  040726  047473  EMT101: .WORD   EMS346,EMS14,EMS341,EMS15
6974  033630  040773
6975  033632  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
6976  033640  000000
6977
6978  033642  047446  044400          EMT102: .WORD   EMS337,EMS70
6979  033646  051724  051401  000000          .WORD   EMS511,EMS503,0
6980  033654  046472  046756  046132  EMT103: .WORD   EMS301,EMS313,EMS254,EMS347,EMS15
6981  033662  047643  040773
6982  033666  051724  051401                  .WORD   EMS511,EMS503
6983  033672  040726  047342  000000          .WORD   EMS14,EMS332,0
6984
6985  033700  047625  041132  047473  EMT104: .WORD   EMS346,EMS17,EMS341,EMS16
6986  033706  041051
6987  033710  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
6988  033716  000000
6989
6990  033720  047446  041132  047473  EMT105: .WORD   EMS337,EMS17,EMS341,EMS16
6991  033726  041051
```

```
 6992  033730  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 6993  033736  000000
 6994
 6995  033740  046472  046756  046132  EMT106: .WORD   EMS301,EMS313,EMS254,EMS347,EMS16
 6996  033746  047643  041051
 6997  033752  051724  051401                  .WORD   EMS511,EMS503
 6998  033756  041132  047342  000000          .WORD   EMS17,EMS332,0
 6999
 7000  033764  047625  041173  047473  EMT107: .WORD   EMS346,EMS20,EMS341,EMS21
 7001  033772  041237
 7002  033774  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7003  034002  000000
 7004
 7005  034004  041173  047671  041237  EMT110: .WORD   EMS20,EMS351,EMS21,EMS350,EMS22,EMS315
 7006  034012  047664  041316  047036
 7007  034020  051724  051305  000000          .WORD   EMS511,EMS501,0
 7008
 7009  034026  041173  047374  047664  EMT111: .WORD   EMS20,EMS334,EMS350,EMS22,EMS333
 7010  034034  041316  047360
 7011  034040  051724  051305  000000          .WORD   EMS511,EMS501,0
 7012
 7013  034046  047446  041173  047473  EMT112: .WORD   EMS337,EMS20,EMS341,EMS21
 7014  034054  041237
 7015  034056  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7016  034064  000000
 7017
 7018  034066  047446  041173  047473  EMT113: .WORD   EMS337,EMS20,EMS341,EMS21,EMS350,EMS22,EMS334
 7019  034074  041237  047664  041316
 7020  034102  047374
 7021  034104  051724  051305  000000          .WORD   EMS511,EMS501,0
 7022
 7023  034112  041173  047707  041237  EMT114: .WORD   EMS20,EMS352,EMS21,EMS350,EMS22,EMS333
 7024  034120  047664  041316  047360
 7025  034126  051724  051305  000000          .WORD   EMS511,EMS501,0
 7026
 7027  034134  046472  046756  046132  EMT115: .WORD   EMS301,EMS313,EMS254,EMS347,EMS21
 7028  034142  047643  041237
 7029  034146  051724  051401                  .WORD   EMS511,EMS503
 7030  034152  041173  047342  000000          .WORD   EMS20,EMS332,0
 7031
 7032  034160  047625  041363  047473  EMT116: .WORD   EMS346,EMS23,EMS341,EMS24
 7033  034166  041441
 7034  034170  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7035  034176  000000
 7036
 7037  034200  047446  041363  047473  EMT117: .WORD   EMS337,EMS23,EMS341,EMS24
 7038  034206  041441
 7039  034210  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7040  034216  000000
 7041
 7042  034220  046472  046756  046132  EMT120: .WORD   EMS301,EMS313,EMS254,EMS347,EMS24
 7043  034226  047643  041441
 7044  034232  051724  051401                  .WORD   EMS511,EMS503
 7045  034236  041363  047342  000000          .WORD   EMS23,EMS332,0
 7046
 7047  034244  047625  041520  047473  EMT121: .WORD   EMS346,EMS25,EMS341,EMS26
```

```
 7048  034252  041576
 7049  034254  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7050  034262  000000
 7051
 7052  034264  047446  041520  047473  EMT122: .WORD   EMS337,EMS25,EMS341,EMS26
 7053  034272  041576
 7054  034274  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7055  034302  000000
 7056
 7057  034304  046472  046756  046132  EMT123: .WORD   EMS301,EMS313,EMS254,EMS347,EMS26
 7058  034312  047643  041576
 7059  034316  051724  051401                  .WORD   EMS511,EMS503
 7060  034322  041520  047342  000000          .WORD   EMS25,EMS332,0
 7061
 7062  034330  046454  041655  046576  EMT124: .WORD   EMS300,EMS27,EMS307,EMS2
 7063  034336  040113
 7064  034340  051724  051401  000000          .WORD   EMS511,EMS503,0
 7065
 7066  034346  047312  041655  047723  EMT125: .WORD   EMS331,EMS27,EMS353
 7067  034354  051724  051401                  .WORD   EMS511,EMS503
 7068  034360  041721  047036  000000          .WORD   EMS30,EMS315,0
 7069
 7070  034366  047446  041655  047473  EMT126: .WORD   EMS337,EMS27,EMS341,EMS30
 7071  034374  041721
 7072  034376  051724  051401  000000          .WORD   EMS511,EMS503,0
 7073
 7074  034404  046472  046756  046132  EMT127: .WORD   EMS301,EMS313,EMS254,EMS347,EMS30
 7075  034412  047643  041721
 7076  034416  051724  051401                  .WORD   EMS511,EMS503
 7077  034422  041655  047342  000000          .WORD   EMS27,EMS332,0
 7078
 7079  034430  042001  047747  045742  EMT130: .WORD   EMS31,EMS354,EMS250
 7080  034436  051724  051425  051401          .WORD   EMS511,EMS504,EMS503,0
 7081  034444  000000
 7082
 7083  034446  042052  047747  045742  EMT131: .WORD   EMS32,EMS354,EMS250
 7084  034454  051724  051425  051401          .WORD   EMS511,EMS504,EMS503,0
 7085  034462  000000
 7086
 7087  034464  050002  042132  045742  EMT132: .WORD   EMS355,EMS33,EMS250,EMS341,EMS30
 7088  034472  047473  041721
 7089  034476  051724  051425  000000          .WORD   EMS511,EMS504,0
 7090
 7091  034504  050002  042171  045742  EMT133: .WORD   EMS355,EMS34,EMS250,EMS341,EMS30
 7092  034512  047473  041721
 7093  034516  051724  051425  000000          .WORD   EMS511,EMS504,0
 7094
 7095  034524  047312  040227  046173  EMT134: .WORD   EMS331,EMS4,EMS255
 7096  034532  051724  051425  000000          .WORD   EMS511,EMS504,0
 7097
 7098  034540  042227  050044  050066  EMT135: .WORD   EMS35,EMS357,EMS360,EMS15
 7099  034546  040773
 7100  034550  051724  051305  000000          .WORD   EMS511,EMS501,0
 7101
 7102  034556  046352  050142          EMT136: .WORD   EMS261,EMS362
 7103  034562  051724  051401  000000          .WORD   EMS511,EMS503,0
```

```
 7104
 7105  034570  046454  042271  046576  EMT137: .WORD   EMS300,EMS36,EMS307,EMS2
 7106  034576  040113
 7107  034600  051724  051305  000000          .WORD   EMS511,EMS501,0
 7108
 7109  034606  050002  042321  046173  EMT140: .WORD   EMS355,EMS37,EMS255,EMS341,EMS30
 7110  034614  047473  041721
 7111  034620  051724  051425  000000          .WORD   EMS511,EMS504,0
 7112
 7113  034626  047625  042361  047577  EMT141: .WORD   EMS346,EMS40,EMS345
 7114  034634  051724  051425  000000          .WORD   EMS511,EMS504,0
 7115
 7116  034642  047446  042361  047473  EMT142: .WORD   EMS337,EMS40,EMS341,EMS30
 7117  034650  041721
 7118  034652  051724  051425  000000          .WORD   EMS511,EMS504,0
 7119
 7120  034660  050163  046637  042437  EMT143: .WORD   EMS363,EMS310,EMS41
 7121  034666  051724  051305  000000          .WORD   EMS511,EMS501,0
 7122
 7123  034674  047446  042437  047473  EMT144: .WORD   EMS337,EMS41,EMS341,EMS252,EMS327,EMS253
 7124  034702  046044  047265  046077
 7125  034710  051724  051305  000000          .WORD   EMS511,EMS501,0
 7126
 7127  034716  042437  047747  050200  EMT145: .WORD   EMS41,EMS354,EMS364,EMS252,EMS365,EMS253
 7128  034724  046044  050221  046077
 7129  034732  051724  051305  000000          .WORD   EMS511,EMS501,0
 7130
 7131  034740  046472  046562  042271  EMT146: .WORD   EMS301,EMS306,EMS36
 7132  034746  051724  051305  051401          .WORD   EMS511,EMS501,EMS503,0
 7133  034754  000000
 7134
 7135  034756  050226  042505          EMT147: .WORD   EMS366,EMS42
 7136  034762  051724  051401  000000          .WORD   EMS511,EMS503,0
 7137
 7138  034770  050251  047723  050221  EMT150: .WORD   EMS367,EMS353,EMS365,EMS42,EMS354,EMS3
 7139  034776  042505  047747  040162
 7140  035004  051724  051401  000000          .WORD   EMS511,EMS503,0
 7141
 7142  035012  047446  042271          EMT151: .WORD   EMS337,EMS36
 7143  035016  051724  051305  000000          .WORD   EMS511,EMS501,0
 7144
 7145  035024  042567  047747  042271  EMT152: .WORD   EMS43,EMS354,EMS36
 7146  035032  051724  051305  000000          .WORD   EMS511,EMS501,0
 7147
 7148  035040  050251  047723  050221  EMT153: .WORD   EMS367,EMS353,EMS365,EMS36,EMS370
 7149  035046  042271  050321
 7150  035052  051724  051401  000000          .WORD   EMS511,EMS503,0
 7151
 7152  035060  050251  047723  050221  EMT154: .WORD   EMS367,EMS353,EMS365,EMS36,EMS371
 7153  035066  042271  050336
 7154  035072  051724  051401  000000          .WORD   EMS511,EMS503,0
 7155
 7156  035100  046454  042640  046576  EMT155: .WORD   EMS300,EMS44,EMS307,EMS2
 7157  035106  040113
 7158  035110  051724  051401  000000          .WORD   EMS511,EMS503,0
 7159
```

```
 7160  035116  050251  047723  050221  EMT156: .WORD   EMS367,EMS353,EMS365,EMS44,EMS354,EMS3
 7161  035124  042640  047747  040162
 7162  035132  051724  051401  000000          .WORD   EMS511,EMS503,0
 7163
 7164  035140  046454  042701  046576  EMT157: .WORD   EMS300,EMS45,EMS307,EMS2
 7165  035146  040113
 7166  035150  051724  051401  000000          .WORD   EMS511,EMS503,0
 7167
 7168  035156  050251  047723  050221  EMT160: .WORD   EMS367,EMS353,EMS365,EMS45,EMS354,EMS3
 7169  035164  042701  047747  040162
```

```
 7170  035172  051724  051401  051305          .WORD   EMS511,EMS503,EMS501
 7171  035200  042227  047360  000000          .WORD   EMS35,EMS333,0
 7172
 7173  035206  046454  042756  046576  EMT161: .WORD   EMS300,EMS46,EMS307,EMS2
 7174  035214  040113
 7175  035216  051724  051401  000000          .WORD   EMS511,EMS503,0
 7176
 7177  035224  047446  042756  047723  EMT162: .WORD   EMS337,EMS46,EMS353
 7178  035232  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7179  035240  000000
 7180
 7181  035242  042227  047404  047446  EMT163. .WORD   EMS35,EMS335,EMS337,EMS41,EMS334,EMS372
 7182  035250  042437  047374  050365
 7183  035256  051724  051305  000000          .WORD   EMS511,EMS501,0
 7184
 7185  035264  043040  047404  047446  EMT164: .WORD   EMS47,EMS335,EMS337,EMS41,EMS335,EMS372
 7186  035272  042437  047404  050365
 7187  035300  051724  051305  000000          .WORD   EMS511,EMS501,0
 7188
 7189  035306  047446  042227  047265  EMT165: .WORD   EMS337,EMS35,EMS327,EMS47
 7190  035314  043040
 7191  035316  051724  051305                  .WORD   EMS511,EMS501
 7192  035322  042437  047360  050365          .WORD   EMS41,EMS333,EMS372,0
 7193  035330  000000
 7194
 7195  035332  046454  043040  046576  EMT166: .WORD   EMS300,EMS47,EMS307,EMS2
 7196  035340  040113
 7197  035342  051724  051305  051401          .WORD   EMS511,EMS501,EMS503,0
 7198  035350  000000
 7199
 7200  035352  043100  047404  047462  EMT167: .WORD   EMS50,EMS335,EMS340,EMS36,EMS333
 7201  035360  042271  047360
 7202  035364  051724  051305  051401          .WORD   EMS511,EMS501,EMS503,0
 7203  035372  000000
 7204
 7205  035374  047446  042227          EMT170: .WORD   EMS337,EMS35
 7206  035400  051724  051305  000000          .WORD   EMS511,EMS501,0
 7207
 7208  035406  043100  042171  040162  EMT171: .WORD   EMS50,EMS34,EMS3
 7209  035414  051724  051305  000000          .WORD   EMS511,EMS501,0
 7210
 7211  035422  046472  046562  043147  EMT172: .WORD   EMS301,EMS306,EMS5
 7212  035430  051724  051305  051425          .WORD   EMS511,EMS501,EMS504,0
 7213  035436  000000
 7214
 7215  035440  050251  047723  050221  EMT173: .WORD   EMS367,EMS353,EMS365,EMS47,EMS354,EMS3
 7216  035446  043040  047747  040162
 7217  035454  051724  051305  000000          .WORD   EMS511,EMS501,0
 7218
 7219  035462  046454  045742  047265  EMT174: .WORD   EMS300,EMS250,EMS327,EMS255,EMS327,EMS256
 7220  035470  046173  047265  046235
 7221  035476  047473  052023                  .WORD   EMS341,EMS600
 7222  035502  051724  051305  000000          .WORD   EMS511,EMS501,0
 7223
 7224  035510  046454  046235  047473  EMT175: .WORD   EMS300,EMS256,EMS341,EMS600
 7225  035516  052023
```

```
7226  035520  051724  051305  000000          .WORD   EMS511,EMS501,0
7227
7228  035526  046454  045742  047473  EMT176: .WORD   EMS300,EMS250,EMS341,EMS600
7229  035534  052023
7230  035536  051724  051425  000000          .WORD   EMS511,EMS504,0
7231
7232  035544  046454  046173  047473  EMT177: .WORD   EMS300,EMS255,EMS341,EMS600
7233  035552  052023
7234  035554  051724  051425  000000          .WORD   EMS511,EMS504,0
7235
7236  035562  047446  043216  047473  EMT200: .WORD   EMS337,EMS52,EMS341,EMS601
7237  035570  052053
7238  035572  051724  051305  000000          .WORD   EMS511,EMS501,0
7239
7240  035600  047625  043216  047473  EMT201: .WORD   EMS346,EMS52,EMS341,EMS602
7241  035606  052073
7242  035610  051724  051305  000000          .WORD   EMS511,EMS501,0
7243
7244  035616  047625  043147  047473  EMT202: .WORD   EMS346,EMS51,EMS341,EMS602
7245  035624  052073
7246  035626  051724  051305  000000          .WORD   EMS511,EMS501,0
7247
7248  035634  047625  043216  047577  EMT203: .WORD   EMS346,EMS52,EMS345,EMS373,EMS255
7249  035642  050417  046173
7250  035646  051724  051425  051305          .WORD   EMS511,EMS504,EMS501,0
7251  035654  000000
7252
7253  035656  047625  043216  047473  EMT204: .WORD   EMS346,EMS52,EMS341,EMS27
7254  035664  041655
7255  035666  051724  051425  051305          .WORD   EMS511,EMS504,EMS501,0
7256  035674  000000
7257
7258  035676  043257  047747  040162  EMT205: .WORD   EMS53,EMS354,EMS3
7259  035704  051724  051401  051332          .WORD   EMS511,EMS503,EMS502,EMS510,0
7260  035712  051651  000000
7261
7262  035716  047446  043320  050423  EMT206: .WORD   EMS337,EMS54,EMS374,EMS250,EMS327,EMS255
7263  035724  045742  047265  046173
7264  035732  047265  040162          .WORD   EMS327,EMS3
7265  035736  051724  051425  051305          .WORD   EMS511,EMS504,EMS501,0
7266  035744  000000
7267
7268  035746  043320  047747  040162  EMT207: .WORD   EMS54,EMS354,EMS3
7269  035754  051724  051305  000000          .WORD   EMS511,EMS501,0
7270
7271  035762  043320  047747  050200  EMT210: .WORD   EMS54,EMS354,EMS364,EMS250
7272  035770  045742
7273  035772  051724  051425  000000          .WORD   EMS511,EMS504,0
7274
7275  036000  043320  047747  050200  EMT211: .WORD   EMS54,EMS354,EMS364,EMS255
7276  036006  046173
7277  036010  051724  051425  000000          .WORD   EMS511,EMS504,0
7278
7279  036016  047446  043375          EMT212: .WORD   EMS337,EMS55
7280  036022  051724  051425  000000          .WORD   EMS511,EMS504,0
7281
```

```
7282  036030  043453  047374  047577  EMT213: .WORD   EMS56,EMS334,EMS345,EMS373,EMS262,EMS327,EMS251
7283  036036  050417  046411  047265
7284  036044  046000
7285  036046  051724  051305          .WORD   EMS511,EMS501
7286  036052  043453  047404  000000  .WORD   EMS56,EMS335,0
7287
7288  036060  047446  043453  EMT214: .WORD   EMS337,EMS56
7289  036064  051724  051305  000000  .WORD   EMS511,EMS501,0
7290
7291  036072  043546  047360  047664  EMT215: .WORD   EMS57,EMS333,EMS350,EMS60,EMS334
7292  036100  043632  047374
7293  036104  051724  051305  000000  .WORD   EMS511,EMS501,0
7294
7295  036112  050163  046562  040260  EMT216: .WORD   EMS363,EMS306,EMS5
7296  036120  051724  051305  000000  .WORD   EMS511,EMS501,0
7297
7298  036126  050163  046562  043704  EMT217: .WORD   EMS363,EMS306,EMS61
7299  036134  051724  051305  000000  .WORD   EMS511,EMS501,0
7300
7301  036142  043757  047360  050454  EMT220: .WORD   EMS62,EMS333,EMS375,EMS251
7302  036150  046000
7303  036152  051724  051305  000000  .WORD   EMS511,EMS501,0
7304
7305  036160  043757  047360  050470  EMT221: .WORD   EMS62,EMS333,EMS376,EMS262
7306  036166  046411
7307  036170  051724  051305  000000  .WORD   EMS511,EMS501,0
7308
7309  036176  043757  047360  050470  EMT222: .WORD   EMS62,EMS333,EMS376,EMS250
7310  036204  045742
7311  036206  051724  051305  000000  .WORD   EMS511,EMS501,0
7312
7313  036214  047625  043757  047473  EMT223: .WORD   EMS346,EMS62,EMS341,EMS603
7314  036222  052134
7315  036224  051724  051305  000000  .WORD   EMS511,EMS501,0
7316
7317  036232  047625  044040  050470  EMT224: .WORD   EMS346,EMS63,EMS376,EMS262
7318  036240  046411
7319  036242  051724  051305  051401  .WORD   EMS511,EMS501,EMS503,0
7320  036250  000000
7321
7322  036252  044040  047360  047664  EMT225: .WORD   EMS63,EMS333,EMS350,EMS363,EMS310,EMS262
7323  036260  050163  046637  046411
7324  036266  051724  051305  000000  .WORD   EMS511,EMS501,0
7325
7326  036274  044040  050044  042271  EMT226: .WORD   EMS63,EMS357,EMS36,EMS372
7327  036302  050365
7328  036304  051724  051305  000000  .WORD   EMS511,EMS501,0
7329
7330  036312  044040  050024  050066  EMT227: .WORD   EMS63,EMS356,EMS360,EMS15
7331  036320  040773
7332  036322  051724  051305  000000  .WORD   EMS511,EMS501,0
7333
```

```
 7334  036330  044040  050024  050114  EMT230: .WORD   EMS63,EMS356,EMS361,EMS15
 7335  036336  040773
 7336  036340  051724  051305  000000          .WORD   EMS511,EMS501,0
 7337
 7338  036346  044040  050024  042437  EMT231: .WORD   EMS63,EMS356,EMS41
 7339  036354  051724  051305  000000          .WORD   EMS511,EMS501,0
 7340
 7341  036362  042437  050504  047502  EMT232: .WORD   EMS41,EMS377,EMS342,EMS365,EMS63,EMS332
 7342  036370  050221  044040  047342
 7343  036376  051724  051305  000000          .WORD   EMS511,EMS501,0
 7344
 7345  036404  050251  047723  050221  EMT233: .WORD   EMS367,EMS353,EMS365,EMS63,EMS401
 7346  036412  044040  050532
 7347  036416  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7348  036424  000000
 7349
 7350  036426  041051  050504  050365  EMT234: .WORD   EMS16,EMS377,EMS372,EMS365,EMS64,EMS354,EMS3
 7351  036434  050221  044100  047747
 7352  036442  040162
 7353  036444  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7354  036452  000000
 7355
 7356  036454  046454  044150  046576  EMT235: .WORD   EMS300,EMS65,EMS307,EMS2
 7357  036462  040113
 7358  036464  051724  051332  051401          .WORD   EMS511,EMS502,EMS503,0
 7359  036472  000000
 7360
 7361  036474  043453  050504  050470  EMT236: .WORD   EMS56,EMS377,EMS376,EMS252,EMS372,EMS350
 7362  036502  046044  050365  047664
 7363  036510  044150  050532          .WORD   EMS65,EMS401
 7364  036514  051724  051332  051401  .WORD   EMS511,EMS502,EMS503,EMS501,0
 7365  036522  051305  000000
 7366
 7367  036526  046454  044211  046576  EMT237: .WORD   EMS300,EMS66,EMS307,EMS2
 7368  036534  040113
```

```
 7369  036536  051724  051305  051401          .WORD   EMS511,EMS501,EMS503,0
 7370  036544  000000
 7371
 7372  036546  040773  050515  050365  EMT240: .WORD   EMS15,EMS400,EMS372,EMS350,EMS66,EMS401
 7373  036554  047664  044211  050532
 7374  036562  051724  051401  051305          .WORD   EMS511,EMS503,EMS501,0
 7375  036570  000000
 7376
 7377  036572  044211  047427  047462  EMT241: .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS604
 7378  036600  040773  050646  050636
 7379  036606  052155
 7380  036610  051724  051401  000000          .WORD   EMS511,EMS503,0
 7381
 7382  036616  050556  052155  050547  EMT242: .WORD   EMS403,EMS604,EMS402,EMS21,EMS377
 7383  036624  041237  050504
 7384  036630  051724  051401                  .WORD   EMS511,EMS503
 7385  036634  042271  047404  000000          .WORD   EMS36,EMS335,0
 7386
 7387  036642  044211  047427  047462  EMT243: .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS604
 7388  036650  041576  050617  050636
 7389  036656  052155
 7390  036660  051724  051401  000000          .WORD   EMS511,EMS503,0
 7391
 7392  036666  044040  050532  050636  EMT244: .WORD   EMS63,FMS401,EMS405,EMS604
 7393  036674  052155
 7394  036676  051724  051401  000000          .WORD   EMS511,EMS503,0
 7395
 7396  036704  042271  050321  050636  EMT245: .WORD   EMS36,EMS370,EMS405,EMS604
 7397  036712  052155
 7398  036714  051724  051401  000000          .WORD   EMS511,EMS503,0
 7399
 7400  036722  050556  052155  050547  EMT246: .WORD   EMS403,EMS604,EMS402,EMS24,EMS377
 7401  036730  041441  050504
 7402  036734  051724  051401                  .WORD   EMS511,EMS503
 7403  036740  042271  047404  000000          .WORD   EMS36,EMS335,0
 7404
 7405  036746  044266  047342  050636  EMT247: .WORD   EMS67,EMS332,EMS405,EMS604
 7406  036754  052155
 7407  036756  051724  051401  000000          .WORD   EMS511,EMS503,0
 7408
 7409  036764  044211  047427  047462  EMT250: .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS605
 7410  036772  040773  050646  050636
 7411  037000  052202
 7412  037002  051724  051401  000000          .WORD   EMS511,EMS503,0
 7413
 7414  037010  050556  052202  050547  EMT251: .WORD   EMS403,EMS605,EMS402,EMS21,EMS377
 7415  037016  041237  050504
 7416  037022  051724  051401                  .WORD   EMS511,EMS503
 7417  037026  042271  047404  000000          .WORD   EMS36,EMS335,0
 7418
 7419  037034  044211  047427  047462  EMT252: .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS605
 7420  037042  041576  050617  050636
 7421  037050  052202
 7422  037052  051724  051401  000000          .WORD   EMS511,EMS503,0
 7423
 7424  037060  044040  050532  050636  EMT253: .WORD   EMS63,EMS401,EMS405,EMS605
```

```
 7425  037066  052202
 7426  037070  051724  051401  000000          .WORD   EMS511,EMS503,0
 7427
 7428  037076  042271  050321  050636  EMT254: .WORD   EMS36,EMS370,EMS405,EMS605
 7429  037104  052202
 7430  037106  051724  051401  000000          .WORD   EMS511,EMS503,0
 7431
 7432  037114  050556  052202  050547  EMT255: .WORD   EMS403,EMS605,EMS402,EMS24,EMS377
 7433  037122  041441  050504
 7434  037126  051724  051401                  .WORD   EMS511,EMS503
 7435  037132  042271  047404  000000          .WORD   EMS36,EMS335,0
 7436
 7437  037140  044266  047342  050636  EMT256: .WORD   EMS67,EMS332,EMS405,EMS605
 7438  037146  052202
 7439  037150  051724  051401  000000          .WORD   EMS511,EMS503,0
 7440
 7441  037156  044211  047427  047462  EMT257: .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS606
 7442  037164  040773  050646  050636
 7443  037172  052220
 7444  037174  051724  051401  000000          .WORD   EMS511,EMS503,0
 7445
 7446  037202  050556  052220  050547  EMT260: .WORD   EMS403,EMS606,EMS402,EMS21,EMS377
 7447  037210  041237  050504
 7448  037214  051724  051401                  .WORD   EMS511,EMS503
 7449  037220  042271  047404  000000          .WORD   EMS36,EMS335,0
 7450
 7451  037226  044211  047427  047462  EMT261: .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS606
 7452  037234  041576  050617  050636
 7453  037242  052220
 7454  037244  051724  051401  000000          .WORD   EMS511,EMS503,0
 7455
 7456  037252  044040  050532  050636  EMT262: .WORD   EMS63,EMS401,EMS405,EMS606
 7457  037260  052220
 7458  037262  051724  051401  000000          .WORD   EMS511,EMS503,0
 7459
 7460  037270  042271  050321  050636  EMT263: .WORD   EMS36,EMS370,EMS405,EMS606
 7461  037276  052220
 7462  037300  051724  051401  000000          .WORD   EMS511,EMS503,0
 7463
 7464  037306  050556  052220  050547  EMT264: .WORD   EMS403,EMS606,EMS402,EMS24,EMS377
 7465  037314  041441  050504
 7466  037320  051724  051401                  .WORD   EMS511,EMS503
 7467  037324  042271  047404  000000          .WORD   EMS36,EMS335,0
 7468
 7469  037332  044400  050532  050636  EMT265: .WORD   EMS70,EMS401,EMS405,EMS606
 7470  037340  052220
 7471  037342  051724  051401  000000          .WORD   EMS511,EMS503,0
 7472
 7473  037350  044266  047342  050636  EMT266: .WORD   EMS67,EMS332,EMS405,EMS606
 7474  037356  052220
 7475  037360  051724  051401  000000          .WORD   EMS511,EMS503,0
 7476
 7477  037366  044211  050024  050661  EMT267: .WORD   EMS66,EMS356,EMS407
 7478  037374  051724  051401  000000          .WORD   EMS511,EMS503,0
 7479
 7480  037402  044211  047427  047462  EMT270: .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS607
```

```
 7481  037410  040773  050646  050636
 7482  037416  052240
 7483  037420  051724  051401  000000          .WORD   EMS511,EMS503,0
 7484
 7485  037426  050556  052240  050547  EMT271: .WORD   EMS403,EMS607,EMS402,EMS21,EMS377
 7486  037434  041237  050504
 7487  037440  051724  051401          .WORD   EMS511,EMS503
 7488  037444  041655  047427  000000  .WORD   EMS27,EMS336,0
 7489
 7490  037452  041655  050321  050636  EMT272: .WORD   EMS27,EMS370,EMS405,EMS607
 7491  037460  052240
 7492  037462  051724  051401  000000          .WORD   EMS511,EMS503,0
 7493
 7494  037470  041655  050336  050636  EMT273: .WORD   EMS27,EMS371,EMS405,EMS607
 7495  037476  052240
 7496  037500  051724  051401  000000          .WORD   EMS511,EMS503,0
 7497
 7498  037506  042271  050617  050636  EMT274: .WORD   EMS36,EMS404,EMS405,EMS607
 7499  037514  052240
 7500  037516  051724  051401  000000          .WORD   EMS511,EMS503,0
 7501
 7502  037524  043257  050532  050636  EMT275: .WORD   EMS53,EMS401,EMS405,EMS607
 7503  037532  052240
 7504  037534  051724  051401  051332          .WORD   EMS511,EMS503,EMS502,0
 7505  037542  000000
 7506
 7507  037544  044266  047342  050636  EMT276: .WORD   EMS67,EMS332,EMS405,EMS607
 7508  037552  052240
 7509  037554  051724  051401  000000          .WORD   EMS511,EMS503,0
 7510
 7511  037562  044211  047427  047462  EMT277: .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS607
 7512  037570  041576  050617  050636
 7513  037576  052240
 7514  037600  051724  051401  000000          .WORD   EMS511,EMS503,0
 7515
 7516  037606  050556  052240  050547  EMT300: .WORD   EMS403,EMS607,EMS402,EMS24,EMS377
 7517  037614  041441  050504
 7518  037620  051724  051401          .WORD   EMS511,EMS503
 7519  037624  041655  047427  000000  .WORD   EMS27,EMS336,0
 7520
 7521  037632  044400  050532  050636  EMT301: .WORD   EMS70,EMS401,EMS405,EMS607
 7522  037640  052240
 7523  037642  051724  051401  000000          .WORD   EMS511,EMS503,0
 7524
 7525  037650  052256  000000          EHT1:   .WORD   EH1,0
 7526  037654  052335  000000          EHT2:   .WORD   EH2,0
 7527  037660  052353  000000          EHT5:   .WORD   EH5,0
 7528  037664  052412  000000          EHT7:   .WORD   EH7,0
 7529  037670  052431  000000          EHT47:  .WORD   EH47,0
 7530  037674  052457  000000          EHT52:  .WORD   EH52,0
 7531  037700  052556  000000          EHT57:  .WORD   EH57,0
 7532  037704  052655  000000          EHT61:  .WORD   EH61,0
 7533  037710  052713  000000          EHT65:  .WORD   EH65,0
 7534  037714  052770  000000          EHT71:  .WORD   EH71,0
 7535  037720  052344  000000          EHT74:  .WORD   EH3,0
 7536  037724  053046  000000          EHT115: .WORD   EH115,0
```

```
 7537  037730  053144  000000          EHT130: .WORD   EH130,0
 7538  037734  053263  000000          EHT132: .WORD   EH132,0
 7539  037740  053362  000000          EHT145: .WORD   EH145,0
 7540  037744  053500  000000          EHT150: .WORD   EH150,0
 7541  037750  053577  000000          EHT213: .WORD   EH213,0
 7542  037754  053675  000000          EHT220: .WORD   EH220,0
 7543
 7544  037760  053734                  EDT1:   .WORD   ED1
 7545  037762  053744                  EDT2:   .WORD   ED2
 7546  037764  053750                  EDT5:   .WORD   ED5
 7547  037766  053756                  EDT47:  .WORD   ED47
 7548  037770  053766                  EDT52:  .WORD   ED52
 7549  037772  054000                  EDT57:  .WORD   ED57
 7550  037774  054012                  EDT61:  .WORD   ED61
 7551  037776  054024                  EDT65:  .WORD   ED65
 7552  040000  054034                  EDT71:  .WORD   ED71
 7553  040002  053744                  EDT74:  .WORD   ED2
 7554  040004  054044                  EDT115: .WORD   ED115
 7555  040006  054056                  EDT130: .WORD   ED130
 7556  040010  054044                  EDT132: .WORD   ED115
 7557  040012  054072                  EDT220: .WORD   ED220
 7558
 7559  040014  054076                  EFT1:   .WORD   EF1
 7560  040016  054101                  EFT2:   .WORD   EF2
 7561  040020  054102                  EFT5:   .WORD   EF5
 7562  040022  054104                  EFT57:  .WORD   EF57
 7563  040024  054076                  EFT65:  .WORD   EF1
 7564  040026  054076                  EFT71:  .WORD   EF1
 7565  040030  054101                  EFT74:  .WORD   EF2
 7566  040032  054104                  EFT115: .WORD   EF57
 7567  040034  054110                  EFT130: .WORD   EF130
 7568  040036  054104                  EFT132: .WORD   EF57
 7569  040040  054102                  EFT220: .WORD   EF5
 7570
 7571                                  .NLIST  BEX
      040042  047516  042516  044530  EMS1:   .ASCIZ  @NONEXISTENT DEVICE 'NED'' (RMCS2,BIT 12) @
      040113     103  047117  051124  EMS2:   .ASCIZ  @CONTROLLER CLEAR ''CLR'' (RMCS2,BIT 05) @
      040162  052506  041516  044524  EMS3:   .ASCIZ  @FUNCTION CODE (RMCS1, BITS 01 - 05) @
      040227     125  052516  042523  EMS4:   .ASCIZ  @UNUSED BIT POSITIONS OF @
      040260  042504  044526  042503  EMS5:   .ASCIZ  @DEVICE AVAILABLE 'DVA'' (RMCS1, BIT 11) @
      040330  040520  052122  054511  EMS6:   .ASCIZ  @PARTIY ERROR 'PAR'' (RMER1, BIT 03) @
      040374  040504  040524  050040  EMS7:   .ASCIZ  @DATA PARITY ERROR 'DPE'' (RMER2, BIT 03) @
      040445     120  051101  052111  EMS10:  .ASCIZ  @PARITY TEST 'PAT'' (RMCS2, BIT 04) @
      040510  040515  051523  052502  EMS11:  .ASCII  @MASSBUS CONTROL BUS PARITY ERROR 'MCPE'' @
      040560  051050  041515  030523          .ASCIZ  @(RMCS1, BIT 13) @
      040601     111  046114  043505  EMS12:  .ASCIZ  @ILLEGAL REGISTER ERROR ''ILR'' (RMER1, BIT 01) @
      040657     104  040511  047107  EMS13:  .ASCIZ  @DIAGNOSTIC MODE 'DMD'' (RMMR1, BIT 00) @
      040726  042515  044504  046525  EMS14:  .ASCIZ  @MEDIUM ON LINE 'MOL'' (RMDS, BIT 12) @
      040773     115  044501  052116  EMS15:  .ASCIZ  @MAINTENANCE UNIT READY 'MUR'' (RMMR1, BIT 09) @
      041051     115  044501  052116  EMS16:  .ASCIZ  @MAINTENANCE WRITE PROTECT 'MWP'' (RMMR1, BIT 03) @
      041132  051127  052111  020105  EMS17:  .ASCIZ  @WRITE LOCK 'WRL'' (RMDS, BIT 11) @
      041173     104  053105  041511  EMS20:  .ASCIZ  @DEVICE CHECK 'DVC'' (RMER2, BIT 07) @
      041237     115  044501  052116  EMS21:  .ASCIZ  @MAINTENANCF DRIVE FAULT 'MDF'' (RMMR1, BIT 06) @
      041316  047125  040523  042506  EMS22:  .ASCIZ  @UNSAFE STATUS 'UNS'' (RMER1, BIT 14) @
      041363     123  042505  020113  EMS23:  .ASCIZ  @SEEK INCOMPLETE STATUS 'SKI'' (RMER2, BIT 14) @
      041441     115  044501  052116  EMS24:  .ASCIZ  @MAINTENANCE SEEK ERROR 'MSER'' (RMMR1, BIT 07) @
```

```
041520  047520  044523  044524  EMS25:  .ASCIZ  @POSITIONING IN PROGRESS 'PIP' (RMDS, BIT 13) @
041576  040515  047111  042524  EMS26:  .ASCIZ  @MAINTENANCE ON CYLINDER 'MOC' (RMMR1, BIT 08) @
041655     105  042116  047440  EMS27:  .ASCIZ  @END OF BLOCK 'EBL' (RMMR1, BIT 13) @
041721     104  040511  047107  EMS30:  .ASCIZ  @DIAGNOSTIC END OF BLOCK 'DEBL' (RMMR1, BIT 13) @
042001     114  051501  020124  EMS31:  .ASCIZ  @LAST SECTOR STATUS 'LS' (RMMR1, BIT 02) @
042052  040514  052123  051440  EMS32:  .ASCIZ  @LAST SECTOR/TRACK STATUS 'LST' (RMMR1, BIT 01) @
042132  042523  052103  051117  EMS33:  .ASCIZ  @SECTOR ADDRESS, BITS 00-04 OF @
042171     124  040522  045503  EMS34:  .ASCIZ  @TRACK ADDRESS, BITS 08-10 OF @
042227     126  046117  046525  EMS35:  .ASCIZ  @VOLUME VALID 'VV' (RMDS, BIT 06) @
042271     107  020117  044502  EMS36:  .ASCIZ  @GO BIT (RMCS1, BIT 00) @
042321     103  046131  047111  EMS37:  .ASCIZ  @CYLINDER ADDRESS, BIT 00-09 OF @
042361     114  051501  020124  EMS40:  .ASCIZ  @LAST BLOCK TAKEN STATUS 'LBT' (RMDS, BIT 10) @
042437     103  046517  047520  EMS41:  .ASCIZ  @COMPOSITE ERROR 'ERR' (RMDS, BIT 14) @
042505     103  046517  040515  EMS42:  .ASCIZ  @COMMAND SEQUENCER TEST BIT 'TST' (RMMR2, BIT 12) @
```

```
042567   104  044522  042526  EMS43:  .ASCIZ  @DRIVE READY STATUS 'DRY'' (RMDS, BIT 07) @
042640  047503 052116  047111  EMS44:  .ASCIZ  @CONTINUE ''CONT'' (RMMR1, BIT 06) @
042701   111  053116  046101  EMS45:  .ASCIZ  @INVALID COMMAND ERROR ''IVC'' (RMER2, BIT 12) @
042756  047514 051523  047440  EMS46:  .ASCIZ  @LOSS OF SYSTEM CLOCK ERROR 'LSC'' (RMER2, BIT 11) @
043040  041517 052503  044520  EMS47:  .ASCIZ  @OCCUPIED 'OCC'' (RMMR1, BIT 15) @
043100  046111 042514  040507  EMS50:  .ASCIZ  @ILLEGAL FUNCTION ''ILF'' (RMER1, BIT 0) @
043147   117  043106  042523  EMS51:  .ASCIZ  @OFFSET DIRECTION 'OFD'' (RMOF, BIT 07) @
043216  043117 051506  052105  EMS52:  .ASCIZ  @OFFSET MODE 'OM'' (RMDS, BIT 00) @
043257   122  047125  040440  EMS53:  .ASCIZ  @RUN AND GO 'RG'' (RMMR1, BIT 14) @
043320  047111 040526  044514  EMS54:  .ASCIZ  @INVALID ADDRESS ERROR ''IAE'' (RMER1, BIT 10) @
043375   101  042104  042522  EMS55:  .ASCIZ  @ADDRESS OVERFLOW ERROR 'AOE'' (RMER1, BIT 09) @
043453   122  043505  051511  EMS56:  .ASCII  @REGISTER MODIFICATION REFUSED ERROR @
043517   042  046522  021122          .ASCIZ  @''RMR'' (RMER1, BIT 02) @
043546  051104 053111  020105  EMS57:  .ASCIZ  @DRIVE REQUEST REQUIRED STATUS 'DRQ'' (RMDT, BIT 11) @
043632  051120 043517  040522  EMS60:  .ASCIZ  @PROGRAMMABLE STATUS 'PGM'' (RMDS, BIT 09) @
043704  051104 053111  020105  EMS61:  .ASCIZ  @DRIVE PRESENT STATUS 'DPR'' (RMDS, BIT 08) @
043757   120  051117  020124  EMS62:  .ASCIZ  @PORT REQUEST FLOP 'RQA,RQB'' (RMMR2, BITS 15,14) @
044040  052101 042524  052116  EMS63:  .ASCIZ  @ATTENTION ''ATA'' (RMDS, BIT 15) @
044100  051127 052111  020105  EMS64:  .ASCIZ  @WRITE LOCK ERROR 'WLE'' (RMER1, BIT 11) @
044150  054105 042503  052120  EMS65:  .ASCIZ  @EXCEPTION 'REX'' (RMMR1, BIT 12) @
044211   111  053116  046101  EMS66:  .ASCIZ  @INVALID COMMAND ERROR ''IVC'' (RMER2, BIT 12) @
044266  040524 020107  052502  EMS67:  .ASCIZ  @TAG BUS (RMMR2, BITS 00-09) OR TAG CONTROL @
044342  044514 042516  020123          .ASCIZ  @LINES (RMMR2, BITS 10,11,13) @
044400  042523 051101  044103  EMS70:  .ASCIZ  @SEARCH ENABLE 'ESRC'' (RMMR1, BIT 11) @

044446  050117 051105  052101  EMS71:  .ASCIZ  @OPERATION IMCOMPLETE ERROR 'OPI''(RMER1 BIT13) @
044525   104  051511  041101  EMS72:  .ASCIZ  @DISABLE SEARCH TIMEOUT 'DTO'' (RMMR1 BIT 12) @
044602  051104 053111  020105  EMS73:  .ASCIZ  @DRIVE TIMING ERROR 'DTE''(RMER1,BIT 12) @
044652  040515 047111  042524  EMS74:  .ASCIZ  @MAINTENANCE INDEX PULSE 'MI''(RMMR1 BIT 2) @
044725   115  044501  052116  EMS75:  .ASCIZ  @MAINTENANCE SECTOR PULSE 'MS''(RMMR1 BIT 5) @
045001   106  051117  040515  EMS76:  .ASCIZ  @FORMAT BIT 'FMT16'' (RMOF,BIT12) @
045042  042523 052103  051117  EMS77:  .ASCIZ  @SECTOR COMPARE 'MSC''(RMMR1 BIT 1) @
045105   120  047522  020115  EMS100: .ASCIZ  @PROM STROBE 'WC'' (RMMR1 BIT 5) @
045145   115  044501  052116  EMS101: .ASCIZ  @MAINTENANCE BIT CLOCK 'MCLK'' (RMMR1 BIT11) @
045221   114  047517  044513  EMS102: .ASCIZ  @LOOKING FOR SYNC 'PLFS''(RMMR1,BIT10) @
045267   127  044522  042524  EMS103: .ASCIZ  @WRITE GATE 'BB00'' (RMMR2 BIT0) @
045327   110  040505  042504  EMS104: .ASCIZ  @HEADER SYNC PATTERN @
045354  042510 042101  051105  EMS105: .ASCIZ  @HEADER AREA 'PHA'' (RMMR1 BIT 7) @
045415   105  040516  046102  EMS106: .ASCIZ  @ENABLE CRC OUT 'ECRC'' (RMMR1 BIT 9) @
045462  042510 042101  051105  EMS107: .ASCIZ  @HEADER @
045472  042522 042101  043440  EMS110: .ASCIZ  @READ GATE 'BB01''(RMMR2 BIT 1) @
045531   110  040505  042504  EMS111: .ASCIZ  @HEADER ERROR @
045547   104  052101  020101  EMS112: .ASCIZ  @DATA TIMING SEQUENCER OUTPUT @
045605   104  052101  020101  EMS113: .ASCIZ  @DATA AREA 'PDA'' (RMMR1 BIT 8) @
045644  047105 041101  042514  EMS114: .ASCIZ  @ENABLE ECC OUT 'EECC'' (RMMR1, BIT 4) @
045712  041505 020103  040520  EMS115: .ASCIZ  @ECC PATTERN @
045727   105  041503  042440  EMS116: .ASCIZ  @ECC ERROR @
045742  044504 045523  040440  EMS250: .ASCIZ  @DISK ADDRESS REGISTER (RMDA) @
046000  047503 052116  047522  EMS251: .ASCIZ  @CONTROL STATUS REGISTER #1 (RMCS1) @
046044  051105 047522  020122  EMS252: .ASCIZ  @ERROR REGISTER #1 (RMER1) @
046077   105  051122  051117  EMS253: .ASCIZ  @ERROR REGISTER #2 (RMER2) @
046132  040515 047111  042524  EMS254: .ASCIZ  @MAINTENANCE REGISTER #1 (RMMR1) @
046173   104  051505  051111  EMS255: .ASCIZ  @DESIRED CYLINDER REGISTER (RMDC) @
046235   117  043106  042523  EMS256: .ASCIZ  @OFFSET REGISTER (RMOF) @
046265   104  044522  042526  EMS257: .ASCIZ  @DRIVE TYPE REGISTER (RMDT) @
046321   110  046117  044504  EMS260: .ASCIZ  @HOLDING REGISTER (RMHR) @
```

```
046352 042523 044522 046101 EMS261: .ASCIZ  @SERIAL NUMBER REGISTER (RMSN) @
046411    101 052124 047105 EMS262: .ASCIZ  @ATTENTION SUMMARY REGISTER (RMAS) @

046454 040503 047116 052117 EMS300: .ASCIZ  @CANNOT CLEAR @
046472 040503 047116 052117 EMS301: .ASCIZ  @CANNOT WRITE/READ @
046515    101 054516 042040 EMS302: .ASCIZ  @ANY DEVICE REGISTER @
046542 044527 044124 052517 EMS303: .ASCIZ  @WITHOUT @
046553    105 051122 051117 EMS304: .ASCIZ  @ERROR @
046562 020101 047117 020105 EMS306: .ASCIZ  @A ONE FROM @
046576 051525 047111 020107 EMS307: .ASCIZ  @USING MASSBUS INITIALIZE, I.E., @
046637    101 055040 051105 EMS310: .ASCIZ  @A ZERO FROM @
046654 053105 051105 020131 EMS311: .ASCIZ  @EVERY DEVICE REGISTER BIT POSITION @
046720 044124 020105 047506 EMS312: .ASCIZ  @THE FOLLOWING BITS ARE STUCK @
046756 020101 044123 043111 EMS313: .ASCIZ  @A SHIFTING ONE BIT FROM @
047007    101 050120 040505 EMS314: .ASCIZ  @APPEARS STUCK AT ZERO @
047036 050101 042520 051101 EMS315: .ASCIZ  @APPEARS STUCK AT ONE @
047064 042522 044507 052123 EMS316: .ASCIZ  @REGISTER SELECT @
047105     061 024040 026061 EMS317: .ASCIZ  @1 (1,2,4,8,16) @
047125     062 024040 026061 EMS320: .ASCIZ  @2 (1,2,4,8,16) @
047145     064 024040 026061 EMS321: .ASCIZ  @4 (1,2,4,8,16) @
047165     070 024040 026061 EMS322: .ASCIZ  @8 (1,2,4,8,16) @
047205     101 046114 047440 EMS323: .ASCIZ  @ALL ONES FROM @
047224 046101 020114 042532 EMS324: .ASCIZ  @ALL ZEROS FROM @
047244 052101 055040 051105 EMS325: .ASCIZ  @AT ZERO @
047255     101 020124 047117 EMS326: .ASCIZ  @AT ONE @
047265     054 047440 020122 EMS327: .ASCIZ  @, OR @
047273     015 041412 020123 EMS330: .ASCIZ  <CR><LF>@CS MBA CLRL @
047312 040503 047116 052117 EMS331: .ASCIZ  @CANNOT READ ZEROS FROM @
047342 051511 044440 041516 EMS332: .ASCIZ  @IS INCORRECT @
047360 051511 047040 052117 EMS333: .ASCIZ  @IS NOT SET @
047374 051511 051440 052105 EMS334: .ASCIZ  @IS SET @
047404 044123 052517 042114 EMS335: .ASCIZ  @SHOULD NOT BE SET @
047427     123 047510 046125 EMS336: .ASCIZ  @SHOULD BE SET @
047446 040503 047116 052117 EMS337: .ASCIZ  @CANNOT SET @
047462 042502 040503 051525 EMS340: .ASCIZ  @BECAUSE @
047473     125 044523 043516 EMS341: .ASCIZ  @USING @
047502 052504 044522 043516 EMS342: .ASCIZ  @DURING REGISTER TRANSFER @
047534 047125 054105 042520 EMS343: .ASCIZ  @UNEXPECTED @
047550 052502 020123 044524 EMS344: .ASCIZ  @BUS TIMEOUT (04 TRAP) @
047577     102 020131 042522 EMS345: .ASCIZ  @BY REGISTER TRANSFER @
047625     103 047101 047516 EMS346: .ASCIZ  @CANNOT RESET @
047643     127 052111 047510 EMS347: .ASCIZ  @WITHOUT SETTING @
047664 052502 020124     000 EMS350: .ASCIZ  @BUT @
047671     127 051501 051040 EMS351: .ASCIZ  @WAS RESET BY @
047707     127 051501 051440 EMS352: .ASCIZ  @WAS SET BY @
047723     111 020116 044504 EMS353: .ASCIZ  @IN DIAGNOSTIC MODE @
047747     111 020123 047111 EMS354: .ASCIZ  @IS INCORRECT ACCORDING TO @
050002 040503 047116 052117 EMS355: .ASCIZ  @CANNOT INCREMENT @
050024 040527 020123 047516 EMS356: .ASCIZ  @WAS NOT SET BY @
050044 040527 020123 047516 EMS357: .ASCIZ  @WAS NOT RESET BY @
050066 020060 047524 030440 EMS360: .ASCIZ  @0 TO 1 TRANSITION OF @
050114 020061 047524 030040 EMS361: .ASCIZ  @1 TO 0 TRANSITION OF @
050142 051511 044440 041516 EMS362: .ASCIZ  @IS INCONSISTENT @
050163     103 047101 047516 EMS363: .ASCIZ  @CANNOT READ @
050200 042524 052123 050040 EMS364: .ASCIZ  @TEST PATTERN IN @
050221     101 042116 000040 EMS365: .ASCIZ  @AND @
```

```
        050226  040503  047116  052117  EMS366: .ASCIZ  @CANNOT INITIALIZE @
        050251     124  042510  041440  EMS367: .ASCIZ  @THE COMMAND SEQUENCER HAS BEEN CLOCKED @
```

```
CZRM
CZRM

EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
EMT1
```

```
050321   122  051505  052105  EMS370: .ASCIZ  @RESET EARLY @
050336  044504  020104  047516  EMS371: .ASCIZ  @DID NOT RESET ON TIME @
050365   104  051125  047111  EMS372: .ASCIZ  @DURING COMMAND EXECUTION @
050417   124  020117   000  EMS373: .ASCIZ  @TO @
050423   127  052111  020110  EMS374: .ASCIZ  @WITH ANY COMBINATION OF @
050454  054502  051040  040505  EMS375: .ASCIZ  @BY READING @
050470  054502  053440  044522  EMS376: .ASCIZ  @BY WRITING @
050504  040527  020123  042523  EMS377: .ASCIZ  @WAS SET @
050515   127  051501  047040  EMS400: .ASCIZ  @WAS NOT SET @
050532  044504  020104  047516  EMS401: .ASCIZ  @DID NOT SET @
050547   127  044510  042514  EMS402: .ASCIZ  @WHILE @
050556  047503  046515  047101  EMS403: .ASCIZ  @COMMAND SEQUENCER DID NOT ABORT @
050617   127  051501  047040  EMS404: .ASCIZ  @WAS NOT RESET @
050636  052504  044522  043516  EMS405: .ASCIZ  @DURING @
050646  040527  020123  042522  EMS406: .ASCIZ  @WAS RESET @
050661   123  040505  041522  EMS407: .ASCIZ  @SEARCH TIMEOUT @
050701   101  052106  051105  EMS410: .ASCIZ  @AFTER @
050710  053440  051501  041440  EMS411: .ASCIZ  @ WAS CHANGED @
050726  051440  052105  042440  EMS412: .ASCIZ  @ SET EARLY @
050742  044440  041516  051117  EMS413: .ASCIZ  @ INCORRECT @
050756  041440  047101  047516  EMS414: .ASCIZ  @ CANNOT DETECT @
050776  053440  051501  051440  EMS415: .ASCIZ  @ WAS SIMULATED @
051016  041440  051117  042522  EMS416: .ASCIZ  @ CORRECT @
051030  042040  051125  047111  EMS417: .ASCIZ  @ DURING WRITE @
051047   107  047105  051105  EMS420: .ASCIZ  @GENERATED @
051062  052504  044522  043516  EMS421: .ASCIZ  @DURING READ @
051077   111  041516  051117  EMS422: .ASCIZ  @INCORRECT DATA TRANSFERRED TO MEMORY @

051145   011  042504  044526  EMS500: .ASCII  @       DEVICE IS NONEXISTENT,@<CR><LF>
051176  042011  053105  041511          .ASCII  @       DEVICE IS SWITCHED TO OTHER PORT@<CR><LF>
051241   011  051124  047101          .ASCIZ  @       TRANSCEIVER ENABLE SWITCH IS OFF@<CR><LF>
051305   011  043111  046440  EMS501: .ASCIZ  @       IF MODULE, M7686,@<CR><LF>
051332  046411  051501  041123  EMS502: .ASCIZ  @       MASSBUS TRANSCEIVER,M5922 OR M5923      @<CR><LF>
051401   011  051503  046440  EMS503: .ASCIZ  @       CS MODULE,M7684,@<CR><LF>
051425   011  051504  046440  EMS504: .ASCIZ  @       DS MODULE,M7685,@<CR><LF>
051451   011  042504  044526  EMS505: .ASCIZ  @       DEVICE IS SWITCHED TO A/B PORT POSITION@<CR><LF>
051524  042011  053105  041511  EMS506: .ASCIZ  @       DEVICE IS NOT AN RM03, OR@<CR><LF>
051561   011  042504  044526  EMS507: .ASCIZ  @       DEVICE IS SWITCHED TO PROGRAMMABLE PORT POSITION, OR@<CR><LF>
051651   011  051501  052523  EMS510: .ASCIZ  @       ASSUMING THE RH CONTROLLER HAS NO FAULT@<CR><LF>
051724  005015  050011  047522  EMS511: .ASCII  <CR><LF>@       PROBABLE FAULT(S):@<CR><LF>
051753   011  047050  052117          .ASCIZ  @       (NOT INCLUDING CABLES OR CONNECTORS)@<CR><LF>

052023   122  040505  020104  EMS600: .ASCIZ  @READ IN PRESET COMMAND @
052053   117  043106  042523  EMS601: .ASCIZ  @OFFSET COMMAND @
052073   122  052105  051125  EMS602: .ASCIZ  @RETURN TO CENTER CENTER COMMAND @
052134  042522  042514  051501  EMS603: .ASCIZ  @RELEASE COMMAND @
052155   122  041505  046101  EMS604: .ASCIZ  @RECALIBRATE COMMAND @
052202  042523  045505  041440  EMS605: .ASCIZ  @SEEK COMMAND @
052220  042523  051101  044103  EMS606: .ASCIZ  @SEARCH COMMAND @
052240  040504  040524  041440  EMS607: .ASCIZ  @DATA COMMAND @

052256  054105  041520  042124  EH1:    .ASCII  @EXPCTD  RECEVD  REGSTR@<CR><LF>
052306  052123  052101  051525          .ASCIZ  @STATUS  STATUS  ADRESS@
052335   040  040502  042523  EH2:    .ASCII  @ BASE@<CR><LF>
052344  042101  042522  051523  EH3:    .ASCIZ  @ADRESS@
052353   105  050130  052103  EH5:    .ASCII  @EXPCTD   STUCK@<CR><LF>
```

```
        052373     122  051505  046125            .ASCIZ  @RESULT  BIT(S)@
        052412  054105  041520  042124  EH7:      .ASCIZ  @EXPCTD  RECEVD@
        052431     040  044506  051522  EH47:     .ASCIZ  @ FIRST  SECOND    CRC@
        052457     105  050130  052103  EH52:     .ASCII  @EXPCTD  RECEVD  EXPCTD  RECEVD@<CR><LF>
        052517     106  051111  052123            .ASCIZ  @FIRST   FIRST   SECOND  SECOND@
        052556  047123  050107  052122  EH57:     .ASCII  @SNGPRT  DULPRT  RECEVD  DRVTYP@<CR><LF>
        052616  051104  052126  050131            .ASCIZ  @DRVTYP  DRVTYP  DRVTYP  REGADR@
        052655     040  054105  042520  EH61:     .ASCIZ  @ EXPECTED ECC    RECEIVED ECC@
        052713     105  050130  052103  EH65:     .ASCII  @EXPCTD  RECEVD  TEST@<CR><LF>
        052741     123  040524  052524            .ASCIZ  @STATUS  STATUS  REGSTR@
        052770  054105  041520  042124  EH71:     .ASCII  @EXPCTD  RECEVD   TEST@<CR><LF>
        053017     123  040524  052524            .ASCIZ  @STATUS  STATUS  PATTRN@
        053046  054105  041520  042124  EH115:    .ASCII  @EXPCTD  RECEVD  REGSTR   TEST@<CR><LF>
        053105     123  040524  052524            .ASCIZ  @STATUS  STATUS  ADRESS  PATTRN@
        053144  054105  041520  042124  EH130:    .ASCII  @EXPCTD  RECEVD  REGSTR   TEST    OFFSET@<CR><LF>
        053214  052123  052101  051525            .ASCIZ  @STATUS  STATUS  ADRESS  PATTRN  REGSTR@
        053263     105  050130  052103  EH132:    .ASCII  @EXPCTD  ACTUAL  REGSTR  OFFSET@<CR><LF>
        053323     103  052517  052116            .ASCIZ  @COUNT   COUNT   ADRESS  REGSTR@
        053362  054105  041520  042124  EH145:    .ASCII  @EXPCTD  ACTUAL  REGSTR  RMER1   RMER2@<CR><LF>
        053431     103  050115  051105            .ASCIZ  @CMPERR  CMPERR  ADRESS  PATTRN  PATTRN@

        053500  054105  041520  042124  EH150:    .ASCII  @EXPCTD  ACTUAL  REGSTR FUNCTION@<CR><LF>
        053541     122  051505  046125            .ASCIZ  @RESULT  RESULT  ADRESS    CODE@
        053577     105  050130  052103  EH213:    .ASCII  @EXPCTD  ACTUAL  STATUS   TEST@<CR><LF>
        053636  042522  052523  052114            .ASCIZ  @RESULT  RESULT  ADRESS  REGSTR@
        053675     101  052103  040525  EH220:    .ASCII  @ACTUAL  REGSTR@<CR><LF>
        053715     122  051505  046125            .ASCIZ  @RESULT  ADRESS@
                                            .LIST BEX
  7572                                      .EVEN
  7573  053734  001140  001142  001136  ED1:      .WORD   $GDDAT,$BDDAT,$BDADR,0
  7574  053742  000000
  7575  053744  001136  000000          ED2:      .WORD   $BDADR,0
  7576  053750  001140  001142  000000  ED5:      .WORD   $GDDAT,$BDDAT,0
  7577  053756  001174  001176  001200  ED47:     .WORD   $TMP0,$TMP1,$TMP2,0
  7578  053764  000000
  7579  053766  001174  054116  001176  ED52:     .WORD   $TMP0,BUFFER,$TMP1,BUFFER+2,0
  7580  053774  054120  000000
  7581  054000  001174  001176  001142  ED57:     .WORD   $TMP0,$TMP1,$BDDAT,$BDADR,0
  7582  054006  001136  000000
  7583  054012  001200  001202  001174  ED61:     .WORD   $TMP2,$TMP3,$TMP0,$TMP1,0
  7584  054020  001176  000000
  7585  054024  001140  001142  001174  ED65:     .WORD   $GDDAT,$BDDAT,$TMP0,0
  7586  054032  000000
  7587  054034  001140  001142  001440  ED71:     .WORD   $GDDAT,$BDDAT,RMHR0,0
  7588  054042  000000
  7589  054044  001140  001142  001136  ED115:    .WORD   $GDDAT,$BDDAT,$BDADR,$TMP0,0
  7590  054052  001174  000000
  7591  054056  001140  001142  001136  ED130:    .WORD   $GDDAT,$BDDAT,$BDADR,$TMP0,$TMP1,0
  7592  054064  001174  001176  000000
  7593  054072  001142  001136          ED220:    .WORD   $BDDAT,$BDADR
  7594
  7595  054076     000     000     000  EF1:      .BYTE   0,0,0
  7596  054101     000                  EF2:      .BYTE   0
  7597  054102     000     000          EF5:      .BYTE   0,0
  7598  054104     000     000     000  EF57:     .BYTE   0,0,0,0
  7599  054107     000
```

```
 7600  054110      000      000      000  EF130:  .BYTE   0,0,0,0,0
 7601  054113      000      000
 7602
 7603            054116                    .EVEN
 7604  054116                              BUFFER:
 7605  054116  000402                      BUFONE: .BLKW   258.
 7606  055122  000402                      BUFTWO: .BLKW   258.
 7607            054116                    .       =       BUFFER
 7608                                      .NLIST BEX
       054116                             HELP:
       054116  005015                     .ASCII  <CR><LF>
       054120  044514  052123  047440      .ASCII  @LIST OF TESTS@<CR><LF>
       054137     055  026455  026455      .ASCII  @-------------@<CR><LF>
       054156  005015                     .ASCII  <CR><LF>
       054160  030524  052011  040522      .ASCII  @T1     TRANSFER TEST@<CR><LF>
       054202  031124  041411  047524      .ASCII  @T2     CTOD TEST@<CR><LF>
       054220  031524  046411  051501      .ASCII  @T3     MASSBUS INITIALIZE TEST@<CR><LF>
       054254  032124  041411  042514      .ASCII  @T4     CLEAR STUCK ACTIVE TEST@<CR><LF>
       054310  032524  052011  044522      .ASCII  @T5     TRISTATE TRANSFER TEST@<CR><LF>
       054343     124  004466  042522      .ASCII  @T6     REGISTER SELECT TEST@<CR><LF>
       054374  033524  042011  044522      .ASCII  @T7     DRIVE TYPE TEST@<CR><LF>
       054420  030524  004460  042504      .ASCII  @T10    DEVICE AVAILABLE TEST@<CR><LF>
       054453     124  030461  051411      .ASCII  @T11    SEARCH TIMEOUT TEST@<CR><LF>
       054504  030524  004462  042523      .ASCII  @T12    SET DTE TEST@<CR><LF>
       054526  030524  004463  047506      .ASCII  @T13    FORMAT CHANGE TEST@<CR><LF>
       054556  030524  004464  051120      .ASCII  @T14    PROM STROBE TEST@<CR><LF>
       054604  030524  004465  054523      .ASCII  @T15    SYNC WORD COUNT INHIBIT TEST@<CR><LF>
       054646  030524  004466  054523      .ASCII  @T16    SYNC DETECTION TEST@<CR><LF>
       054677     124  033461  040411      .ASCII  @T17    ABORT SYNC DETECTION TEST@<CR><LF>
       054736  031124  004460  054523      .ASCII  @T20    SYNC GENERATION TEST@<CR><LF>
       054770  031124  004461  051127      .ASCII  @T21    WRITE HEADER TEST@<CR><LF>
       055017     124  031062  044011      .ASCII  @T22    HEADER COMPARE TEST @<CR><LF>
       055051     124  031462  042411      .ASCII  @T23    ECC GENERATION TEST@<CR><LF>
       055102  031124  004464  041505      .ASCII  @T24    ECC DETECTION TEST@<CR><LF>
       055132  005015                     .ASCII  <CR><LF>
       055134  005015                     .ASCII  <CR><LF>
       055136  053523  052111  044103      .ASCII  @SWITCH                 USE@<CR><LF>
       055154  026455  026455  026455      .ASCII  @------         --------------------@<CR><LF>
       055212  020040  032461  004411      .ASCII  @  15           HALT ON ERROR@<CR><LF>
       055237     040  030440  004464      .ASCII  @  14           LOOP ON TEST@<CR><LF>
       055263     040  030440  004463      .ASCII  @  13           INHIBIT ERROR TYPEOUTS@<CR><LF>
       055321     040  030440  004461      .ASCII  @  11           INHIBIT ITERATIONS@<CR><LF>
       055353     040  030440  004460      .ASCII  @  10           BELL ON ERROR@<CR><LF>
       055400  020040  034440  004411      .ASCII  @   9           LOOP ON ERROR@<CR><LF>
       055425     040  020040  004470      .ASCII  @   8           LOOP ON TEST IN SWR<7:0>@<CR><LF>
       055465     040  020040  004467      .ASCII  @   7           TN128@<CR><LF>
       055502  020040  033040  004411      .ASCII  @   6           TN64@<CR><LF>
       055516  020040  032440  004411      .ASCII  @   5           TN32@<CR><LF>
       055532  020040  032040  004411      .ASCII  @   4           TN16@<CR><LF>
       055546  020040  031440  004411      .ASCII  @   3           TN8@<CR><LF>
       055561     040  020040  004462      .ASCII  @   2           TN4@<CR><LF>
       055574  020040  030440  004411      .ASCII  @   1           TN2@<CR><LF>
       055607     040  020040  004460      .ASCII  @   0           TN1@<CR><LF>
       055622  005015     000             .ASCIZ  <CR><LF>
                                          .LIST BEX
 7609            000001                   .END
```

```
ABASE = 176700            1359#   1471    1512
ACDW1 = 000000            1471    1514
ACDW2 = 000000            1471    1515
ACPUOP= 000000            1471    1486
ADDW0 = 000000            1471    1516
ADDW1 = 000000            1471    1517
ADDW10= 000000            1471
ADDW11  000000            1471
ADDW12= 000000            1471
ADDW13= 000000            1471
ADDW14= 000000            1471
ADDW15= 000000            1471
ADDW2 = 000000            1471    1518
ADDW3 = 000000            1471    1519
ADDW4 = 000000            1471    1520
ADDW5 = 000000            1471    1521
ADDW6 = 000000            1471    1522
ADDW7 = 000000            1471    1523
ADDW8 = 000000            1471
ADDW9 = 000000            1471
ADEVCT= 000000            1471    1477
ADEVM = 000000            1471    1513
AENV  = 000000            1471    1482
AENVM = 000000            1471    1483
AFATAL= 000000            1471    1474
AMADR1= 000000            1471    1499
AMADR2= 000000            1471    1503
AMADR3= 000000            1471    1506
AMADR4= 000000            1471    1509
AMAMS1= 000000            1471    1493
AMAMS2= 000000            1471    1501
AMAMS3= 000000            1471    1504
AMAMS4= 000000            1471    1507
AMSGAD= 000000            1471    1479
AMSGLG= 000000            1471    1480
AMSGTY= 000000            1471    1473
AMTYP1= 000000            1471    1494
AMTYP2= 000000            1471    1502
AMTYP3= 000000            1471    1505
AMTYP4= 000000            1471    1508
AOE   = 001000            1135#   1146    6746    6747    6750    6751    6754    6755
APASS = 000000            1471    1476
APE   = 100000            1338#
APRIOR- 000000            1471
APTCSU= 000040            6075    6716#
APTENV= 000001            6068    6250    6672    6714#
APTSIZ= 000200            2247    6713#
APTSPO= 000100            6070    6674    6715#
ASWREG= 000000            1471    1484
ATA   = 100000            1115#   6727    6728    6729    6732    6733    6736    6737    6738    6739    6740    6741    6742
                          6743    6744    6745    6748    6749    6752    6753    6756    6757
ATESTN= 000000            1471    1475
ATNMSK= 000377            1152#
ATNTBL  031734            2444    2462    6759#
AUNIT = 000000            1471    1478
AUSWR - 000000            1471    1485
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
AVECT1= 120254          1360#   1471    1510
AVECT2= 000000          1471    1511
A16   = 000400          1309#
A17   = 001000          1308#
BAI   = 000010          1328#
BBG0  = 000001          1245#   4006    4020    4026    4199    4203
BB01  = 000002          1244#   4425    4428    4806    4808
BB02  = 000004          1243#
BB03  = 000010          1242#
BB04  = 000020          1241#
BB05  = 000040          1240#
BB06  = 000100          1239#
BB07  = 000200          1238#
BB08  = 000400          1237#
BB09  = 001000          1236#
BIT0  = 000001          1031#   2291    2829    3258    3260    4093    4230    4283    4782
BIT00 = 000001          1021#   1031    1057    1106    1125    1144    1182    1202    1245    1332    1348
BIT01 = 000002          1020#   1030    1056    1105    1143    1181    1201    1244    1331    1347
BIT02 = 000004          1019#   1029    1055    1104    1142    1180    1200    1243    1330    1346
BIT03 = 000010          1018#   1028    1054    1103    1141    1179    1199    1242    1257    1328    1345
BIT04 = 000020          1017#   1027    1053    1102    1140    1198    1241    1327
BIT05 = 000040          1016#   1026    1052    1139    1178    1197    1240    1326
BIT06 = 000100          1015#   1025    1124    1138    1160    1177    1196    1239    1311    1325    1344
BIT07 = 000200          1014#   1024    1123    1137    1159    1176    1195    1219    1238    1256    1310    1324
BIT08 = 000400          1013#   1023    1101    1122    1136    1158    1175    1194    1237    1309    1322    6154
BIT09 = 001000          1012#   1022    1100    1121    1135    1157    1174    1193    1236    1308    1321    6172    6261
BIT1  = 000002          1030#   2886
BIT10 = 002000          1011#   1099    1120    1134    1156    1173    1192    1218    1235    1255    1307    1320    1343
                        6238
BIT11 = 004000          1010#   1051    1119    1133    1172    1191    1209    1217    1234    1254    1319    1342    6179
BIT12 = 010000          1009#   1118    1132    1171    1190    1216    1233    1253    1318    1341
BIT13 = 020000          1008#   1117    1131    1170    1189    1208    1232    1252    1305    1317    1340    6245
BIT14 = 040000          1007#   1116    1130    1169    1188    1207    1231    1251    1263    1304    1316    1339    6140
BIT15 = 100000          1006#   1115    1129    1168    1187    1206    1230    1250    1262    1303    1315    1338
BIT2  = 000004          1029#
BIT3  = 000010          1028#
BIT4  = 000020          1027#
BIT5  = 000040          1026#
BIT6  = 000100          1025#
BIT7  = 000200          1024#   2286
BIT8  = 000400          1023#
BIT9  = 001000          1022#
BOTADR  022446          5236*   5254*   5257    5272    5317#
BOTFLG  022450          5222*   5264*   5267    5270*   5318#
BPTVEC= 000014          1038#
BSE   = 100000          1250#
BUFFER  054116          3316    3405    3483    3660    3798    3913    3981    4130    4325    4589    4591    4618    7579
                        7604#   7607
BUFONE  054116          4885    5049    7605#
BUFTWO  055122          7606#
CC    = 004000          1234#
CH    = 002000          1235#
CHRCNT  022451          5223*   5241*   5247*   5248    5251*   5255    5260*   5271*   5319#
CKSWR = 104410          6139    6234    6260    6611#
CLKSNC  024522          4649    4909    4960    5772#
CLOCK   001526          1605#   3328    3348    5343*   5356*   5367*   5608    5684
```

```
CLR    = 000040  1326# 2293  2541  2562  2624  2677  2687  2725  2767  2832  2893  3015  3038
                 3070  3102  3126  3158  3181  3203  3286  3380  3471  3568  3663  4124  4331
                 4612  4879  5439  5514
CLSPRN  030752   2364  2380  2401  6720#
CMNSTA  004254   2316  2451#
CNSL00  031116   2326  6720#
CNSL01  031155   2361  6720#
CNSL02  031202   2370  6720#
CNSL03  031263   2375  6720#
CNSL04  031313   2386  6720#
CNSL05  031367   2396  6720#
CNSL06  031423   2407  6720#
CNSL07  031450   2420  6720#
CONT   = 000100  1196#
CR     = 000015   946# 2434  5239  6114  6124  6720  7571  7608
CRLF   = 000200   947# 2275  6085  6124
CYLMSK= 001777  1224# 2632
DBCK   - 100000  1168# 5631  5648
DBEN   - 040000  1169# 1 33  3346  3418  3420  3433  3435  3437  3440  3442  3496  3498  3500
                 3502  3518  3521  3523  3525  3528  3530  3685  3687  3711  3713  3728  3730
                 3750  3752  3762  3764  5598  5631  5633  5640  5642  5648  5650  5679  5681
                 5691  5693  5696  5698  5700  5702  5717  5719  5728  5730
DBL    - 002000  1343#
DCK    = 100000  1129#  1146  5037
DDISP  = 177570   953#  1443  2235
DEBL   - 020000  1170#
DISPLA  001156  1443#  2235*  2243*  6194*  6237*
DISPRE  000174  1370#  2243
DLT    = 100000  1315#
DMD    = 000001  1182#  1183  1202#  2733  2745  3346  3391  3393  3418  3420  3433  3435  3437
                 3440  3442  3476  3478  3496  3498  3500  3502  3518  3521  3523  3525  3528
                 3530  3574  3578  3580  3593  3595  3610  3612  3627  3629  3685  3687  3711
                 3713  3728  3730  3750  3752  3762  3764  5442  5444  5477  5517  5523  5525
                 5541  5543  5561  5563  5598  5631  5633  5640  5642  5648  5650  5679  5681
                 5691  5693  5696  5698  5700  5702  5717  5719  5728  5730
DPE    - 000010  1257#
DPEHI  - 040000  1339#
DPELO  = 020000  1340#
DPR    = 000400  1122#
DRQ    - 004000  1209#
DRVCLR= 000010  1066#
DRY    = 000200  1123#
DSWR   = 177570   952#  1442  2234
DTE    = 010000  1132#  1146  3384  3396  3426  3445  3447  3506  3533  3537
DTO    = 010000  1171#  1183  3418  3420  3433  3435  3437  3440  3442  3496  3498  3500  3502
                 3518  3521  3523  3525  3528  3530  3685  3687  3711  3713  3728  3730  3750
                 3752  3762  3764  5648  5650  5679  5681  5691  5693  5696  5698  5700  5702
                 5717  5719  5728  5730
DULPRT- 024024  1212#  3256  3260  3263
DVA    = 004000  1051#  2299  3290  3292
DVC    = 000200  1256#  2743
EBL    = 020000  1189#
ECH     000100  1138#  1146  5037  6746  6747  6754  6755
ECI    - 004000  1217#  2634
ECRC   = 001000  1193#  4257  4268  4670
EDT1    037760  1632  1753  1761  1769  1777  1785  1793  1809  1817  1825  1833  1841  1849
```

```
                        1857    1865    1881    1889    1897    1921    1929    1945    1953    1969    1977    1985    1993
                        2009    2025    2033    2041    2049    2135    2143    2167    2175    2183    2191    7544#
EDT115  040004          1873    2127    2151    7554#
EDT130  040006          7555#
EDT132  040010          7556#
EDT2    037762          1640    7545#
EDT220  040012          7557#
EDT47   037766          1937    7547#
EDT5    037764          1664    1672    1681    1913    7546#
EDT52   037770          1961    7548#
EDT57   037772          2001    7549#
EDT61   037774          2017    7550#
EDT65   037776          7551#
EDT71   040000          7552#
EDT74   040002          7553#
ED1     053734          7544    7573#
ED115   054044          7554    7556    7589#
ED130   054056          7555    7591#
ED2     053744          7545    7553    7575#
ED220   054072          7557    7593#
ED47    053756          7547    7577#
ED5     053750          7546    7576#
ED52    053766          7548    7579#
ED57    054000          7549    7581#
ED61    054012          7550    7583#
ED65    054024          7551    7585#
ED71    054034          7552    7587#
EECC  = 000020          1198#   4670    4732    4755    4756    4766
EFT1    040014          1633    1754    1762    1770    1778    1786    1794    1810    1818    1826    1834    1842    1850
                        1858    1866    1882    1890    1898    1922    1930    1938    1946    1954    1970    1978    1986
                        1994    2010    2026    2034    2042    2050    2136    2144    2168    2176    2184    2192    7559#
EFT115  040032          1874    2128    2152    7566#
EFT130  040034          7567#
EFT132  040036          7568#
EFT2    040016          1641    7560#
EFT220  040040          7569#
EFT5    040020          1665    1673    1682    1914    7561#
EFT57   040022          1962    2002    2018    7562#
EFT65   040024          7563#
EFT71   040026          7564#
EFT74   040030          7565#
EF1     054076          7559    7563    7564    7595#
EF130   054110          7567    7600#
EF2     054101          7560    7565    7596#
EF5     054102          7561    7569    7597#
EF57    054104          7562    7566    7568    7598#
EHT1    037650          1631    1752    1760    1768    1776    1784    1792    1808    1816    1824    1832    1840    1848
                        1856    1864    1880    1888    1896    1920    1928    1944    1952    1968    1976    1984    1992
                        2008    2024    2032    2040    2048    2134    2142    2166    2174    2182    2190    7525#
EHT115  037724          1872    7536#
EHT130  037730          7537#
EHT132  037734          7538#
EHT145  037740          7539#
EHT150  037744          2126    2150    7540#
EHT2    037654          1639    7526#
EHT213  037750          7541#
```

```
EHT220  037754            7542#
EHT47   037670            1936    7529#
EHT5    037660            1663    1671    7527#
EHT52   037674            1960    7530#
EHT57   037700            2000    7531#
EHT61   037704            2016    7532#
EHT65   037710            7533#
EHT7    037664            1680    1912    7528#
EHT71   037714            7534#
EHT74   037720            7535#
EH1     052256            7525    7571#
EH115   053046            7536    7571#
EH130   053144            7537    7571#
EH132   053263            7538    7571#
EH145   053362            7539    7571#
EH150   053500            7540    7571#
EH2     052335            7526    7571#
EH213   053577            7541    7571#
EH220   053675            7542    7571#
EH3     052344            7535    7571#
EH47    052431            7529    7571#
EH5     052353            7527    7571#
EH52    052457            7530    7571#
EH57    052556            7531    7571#
EH61    052655            7532    7571#
EH65    052713            7533    7571#
EH7     052412            7528    7571#
EH71    052770            7534    7571#
EMS1    040042            6770    6771    7571#
EMS10   040445            6942    6947    7571#
EMS100  045105            6845    6852    6854    6860    6863    6867    6870    6920    7571#
EMS101  045145            6845    6920    7571#
EMS102  045221            6856    6858    6860    6863    6872    7571#
EMS103  045267            6874    7571#
EMS104  045327            6865    6867    6877    7571#
EMS105  045354            6879    7571#
EMS106  045415            6885    7571#
EMS107  045462            6887    6890    6895    7571#
EMS11   040510            6952    7571#
EMS110  045472            6890    7571#
EMS111  045531            6893    7571#
EMS112  045547            6900    7571#
EMS113  045605            6903    6923    7571#
EMS114  045644            6906    7571#
EMS115  045712            6917    7571#
EMS116  045727            6925    7571#
EMS12   040601            5930    7571#
EMS13   040657            6960    6963    6966    6969    7571#
EMS14   040726            6973    6983    7571#
EMS15   040773            6973    6980    7098    7330    7334    7372    7377    7409    7441    7480    7571#
EMS16   041051            6985    6990    6995    7350    7571#
EMS17   041132            6985    6990    6998    7571#
EMS2    040113            6778    7062    7105    7156    7164    7173    7195    7356    7367    7571#
EMS20   041173            7000    7005    7009    7013    7018    7023    7030    7571#
EMS21   041237            7000    7005    7013    7018    7023    7027    7382    7414    7446    7485    7571#
EMS22   041316            7005    7009    7018    7023    7571#
```

```
CZRM
CZRM

RMMR

RMMR

RMMR
RMMR
RMOF

RMOF

RMOF
RMR
RMSN
RMSN
RMSN
RMWC
RMWC
RMWC
RQA
RQB
RTC
SADM
SAVR
SA1
SA16
SA2
SA4
SA8
SC
SCTC
SCTM
SCO
SC1
SC2
SC3
SC4
SEAR
SEEK
SETL
SETC
SETV
SIZC
SKI
SNGR
STAC

STAR
STAR
STKL
STOR
SWR

SWR
SWO
SWOC
SWO
```

```
EMS23   041363  7032  7037  7045  7571#
EMS24   041441  7032  7037  7042  7400  7432  7464  7516  7571#
EMS25   041520  7047  7052  7060  7571#
EMS250  045742  7079  7083  7087  7091  7219  7228  7262  7271  7309  7571#
EMS251  046000  7282  7301  7571#
EMS252  046044  6837  6849  7123  7127  7361  7571#
EMS253  046077  6837  6882  7123  7127  7571#
EMS254  046132  6837  6969  6980  6995  7027  7042  7057  7074  7571#
EMS255  046173  7095  7109  7219  7232  7248  7262  7275  7571#
EMS256  046235  6898  7219  7224  7571#
EMS257  046265  6909  7571#
EMS26   041576  7047  7052  7057  7387  7419  7451  7511  7571#
EMS260  046321  6952  7571#
EMS261  046352  7102  7571#
EMS262  046411  7282  7305  7317  7322  7571#
EMS27   041655  7062  7066  7070  7077  7253  7488  7490  7494  7519  7571#
EMS3    040162  7138  7160  7168  7208  7215  7258  7264  7268  7350  7571#
EMS30   041721  7068  7070  7074  7087  7091  7109  7116  7571#
EMS300  046454  6770  6778  6849  6882  6933  6937  6960  7062  7105  7156  7164  7173  7195
                7219  7224  7228  7232  7356  7367  7571#
EMS301  046472  6771  6775  6782  6786  6789  6817  6952  6969  6980  6995  7027  7042  7057
                7074  7131  7211  7571#
EMS302  046515  6771  6775  6778  6789  7571#
EMS303  046542  6771  6952  7571#
EMS304  046553  6771  7571#
EMS306  046562  6775  6786  6837  7131  7211  7295  7298  7571#
EMS307  046576  6778  7062  7105  7156  7164  7173  7195  7356  7367  7571#
EMS31   042001  7079  7571#
EMS310  046637  6782  7120  7322  7571#
EMS311  046654  6782  6786  7571#
EMS312  046720  6785  7571#
EMS313  046756  6789  6969  6980  6995  7027  7042  7057  7074  7571#
EMS314  047007  6792  6798  6804  6810  6841  7571#
EMS315  047036  6795  6801  6807  6813  7005  7068  7571#
EMS316  047064  6792  6795  6798  6801  6804  6807  6810  6813  7571#
EMS317  047105  6792  6795  7571#
EMS32   042052  7083  7571#
EMS320  047125  6798  6801  7571#
EMS321  047145  6804  6807  7571#
EMS322  047165  6810  6813  7571#
EMS323  047205  7571#
EMS324  047224  7571#
EMS325  047244  7571#
EMS326  047255  7571#
EMS327  047265  6837  6874  6879  6903  6937  7123  7189  7219  7262  7264  7282  7571#
EMS33   042132  7087  7571#
EMS330  047273  6841  7571#
EMS331  047312  6898  7066  7095  7571#
EMS332  047342  6909  6983  6998  7030  7045  7060  7077  7341  7405  7437  7473  7507  7571#
EMS333  047360  6913  6935  6942  7009  7023  7171  7192  7200  7291  7301  7305  7309  7322
                7571#
EMS334  047374  6860  6947  7009  7018  7181  7282  7291  7571#
EMS335  047404  6816  6834  6860  6863  6942  7181  7185  7200  7286  7385  7403  7417  7435
                7449  7467  7571#
EMS336  047427  6819  6858  6867  6870  6930  6947  7377  7387  7409  7419  7441  7451  7480
                7488  7511  7519  7571#
```

```
EMS337  047446  6842  6845  6966  6978  6990  7013  7018  7037  7052  7070  7116  7123  7142
                7177  7181  7185  7189  7205  7236  7262  7279  7288  7571#
EMS34   042171  7091  7208  7571#
EMS340  047462  6816  6819  6834  6860  6863  6867  6870  6942  6947  7200  7377  7387  7409
                7419  7441  7451  7480  7511  7571#
EMS341  047473  6845  6920  6973  6985  6990  7000  7013  7018  7032  7037  7047  7052  7070
                7087  7091  7109  7116  7123  7221  7224  7228  7232  7236  7240  7244  7253
                7313  7571#
EMS342  047502  6930  6942  6947  6957  7341  7571#
EMS343  047534  6893  6957  7571#
EMS344  047550  6957  7571#
EMS345  047577  6963  6966  7113  7248  7282  7571#
EMS346  047625  6920  6963  6973  6985  7000  7032  7047  7113  7240  7244  7248  7253  7313
                7317  7571#
EMS347  047643  6969  6980  6995  7027  7042  7057  7074  7571#
EMS35   042227  7098  7171  7181  7189  7205  7571#
EMS350  047664  7005  7009  7018  7023  7291  7322  7361  7372  7571#
EMS351  047671  7005  7571#
EMS352  047707  6822  6826  7023  7571#
EMS353  047723  6925  6927  7066  7138  7148  7152  7160  7168  7177  7215  7345  7571#
EMS354  047747  7079  7083  7127  7138  7145  7160  7168  7215  7258  7268  7271  7275  7350
                7571#
EMS355  050002  7087  7091  7109  7571#
EMS356  050024  6830  7330  7334  7338  7477  7571#
EMS357  050044  7098  7326  7571#
EMS36   042271  7105  7131  7142  7145  7148  7152  7200  7326  7385  7396  7403  7417  7428
                7435  7449  7460  7467  7498  7571#
EMS360  050066  7098  7330  7571#
EMS361  0501?4  7334  7571#
EMS362  050142  7102  7571#
EMS363  050163  7120  7295  7298  7322  7571#
EMS364  050200  7127  7271  7275  7571#
EMS365  050221  6863  7127  7138  7148  7152  7160  7168  7215  7341  7345  7350  7571#
EMS366  050226  7135  7571#
EMS367  050251  7138  7148  7152  7160  7168  7215  7345  7571#
EMS37   042321  7109  7571#
EMS370  050321  6852  6874  6879  6903  7148  7396  7428  7460  7490  7571#
EMS371  050336  6923  7152  7494  7571#
EMS372  050365  6816  6819  7181  7185  7192  7326  7350  7361  7372  7571#
EMS373  050417  7248  7282  7571#
EMS374  050423  7262  7571#
EMS375  050454  7301  7571#
EMS376  050470  7305  7309  7317  7361  7571#
EMS377  050504  6816  6830  6863  7341  7350  7361  7382  7400  7414  7432  7446  7464  7485
                7516  7571#
EMS4    040227  6898  7095  7571#
EMS40   042?61  7113  7116  7571#
EMS400  050515  6819  6872  7372  7571#
EMS401  050532  6874  6879  6885  6903  6906  7345  7363  7372  7392  7424  7456  7469  7502
                7521  7571#
EMS402  050547  6822  6826  6830  6870  7382  7400  7414  7432  7446  7464  7485  7516  7571#
EMS403  050556  7382  7400  7414  7432  7446  7464  7485  7516  7571#
EMS404  050617  7387  7419  7451  7498  7511  7571#
EMS405  050636  6890  6900  7377  7387  7392  7396  7405  7409  7419  7424  7428  7437  7441
                7451  7456  7460  7469  7473  7480  7490  7494  7498  7502  7507  7511  7521
                7571#
```

```
CZRM
CZRM

TST2
TST3
TST4
TST5
TST6
TST7
TYPB
TYPD
TYPE

TYPC
TYPC
TYPC
T1
T10
T11
T12
T13
T14
T15
T16
T17
T2
T20
T21
T22
T23
T24
T3
T4
T5
T6
T7
UBUS
UNS
UNTM
UPE
USE
U0
U1
U2
VV
WATC
WC

WCD
WCE
WCEM
WCEL
WCF
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
EMS406  050646          6822  6826  7377  7409  7441  7480  7571#
EMS407  050661          7477  7571#
EMS41   042437          7120  7123  7127  7181  7185  7192  7338  7341  7571#
EMS410  050701          6842  7571#
EMS411  050710          6834  6842  7571#
EMS412  050726          6854  6856  7571#
EMS413  050742          6865  6887  6890  6900  6917  6925  7571#
EMS414  050756          6877  6895  7571#
EMS415  050776          6865  6867  6870  7571#
EMS416  051016          6867  6895  7571#
EMS417  051030          6877  6885  6887  6906  6917  7571#
EMS42   042505          7135  7138  7571#
EMS420  051047          6887  6917  7571#
EMS421  051062          6893  6895  7571#
EMS422  051077          6927  7571#
EMS43   042567          7145  7571#
EMS44   042640          7156  7160  7571#
EMS45   042701          7164  7168  7571#
EMS46   042756          7173  7177  7571#
EMS47   043040          7185  7189  7195  7215  7571#
EMS5    040260          6913  7295  7571#
EMS50   043100          7200  7208  7571#
EMS500  051145          6773  7571#
EMS501  051305          6773  6776  6780  6783  6787  6790  6793  6796  6799  6802  6805  6808  6811
                        6814  6839  6850  6883  6899  6910  6914  6931  6934  6939  6944  6949  6954
                        6975  6987  6992  7002  7007  7011  7015  7021  7025  7034  7039  7049  7054
                        7100  7107  7121  7125  7129  7132  7143  7146  7170  7178  7183  7187  7191
                        7197  7202  7206  7209  7212  7217  7222  7226  7238  7242  7246  7250  7255
                        7265  7269  7285  7289  7293  7296  7299  7303  7307  7311  7315  7319  7324
                        7328  7332  7336  7339  7343  7347  7353  7364  7369  7374  7571#
EMS502  051332          6773  6776  6780  6783  6787  6790  6793  6796  6799  6802  6805  6808  6811
                        6814  6839  6944  6949  6954  7259  7358  7364  7504  7571#
EMS503  051401          6773  6780  6783  6787  6790  6818  6821  6824  6828  6832  6836  6839  6844
                        6847  6850  6857  6859  6876  6881  6883  6886  6892  6902  6905  6907  6924
                        6961  6964  6967  6971  6975  6979  6982  6987  6992  6997  7002  7015  7029
                        7034  7039  7044  7049  7054  7059  7064  7067  7072  7076  7080  7084  7103
                        7132  7136  7140  7150  7154  7158  7162  7166  7170  7175  7178  7197  7202
                        7259  7319  7347  7353  7358  7364  7369  7374  7380  7384  7390  7394  7398
                        7402  7407  7412  7416  7422  7426  7430  7434  7439  7444  7448  7454  7458
                        7462  7466  7471  7475  7478  7483  7487  7492  7496  7500  7504  7509  7514
                        7518  7523  7571#
EMS504  051425          6783  6787  6790  6824  6828  6832  6847  6853  6855  6862  6866  6869  6873
                        6878  6889  6894  6897  6919  6922  6926  6928  6939  7080  7084  7089  7093
                        7096  7111  7114  7118  7212  7230  7234  7250  7255  7265  7273  7277  7280
                        7571#
EMS505  051451          6776  7571#
EMS506  051524          6910  7571#
EMS507  051561          6914  7571#
EMS51   043147          7211  7244  7571#
EMS510  051651          7259  7571#
EMS511  051724          6773  6776  6780  6783  6787  6790  6793  6796  6799  6802  6805  6808  6811
                        6814  6818  6821  6824  6828  6832  6836  6839  6844  6847  6850  6853  6855
                        6857  6859  6862  6866  6869  6873  6876  6878  6881  6883  6886  6889  6892
                        6894  6897  6899  6902  6905  6907  6910  6914  6919  6922  6924  6926  6928
                        6931  6934  6939  6944  6949  6954  6961  6964  6967  6971  6975  6979  6982
                        6987  6992  6997  7002  7007  7011  7015  7021  7025  7029  7034  7039  7044
```



```
WCH
WD
WH
WLE
WRL
XNUD
XNUE
XNUC
XSIZ
$APT
$AST
$ATY
$ATY
$ATY
$ATY
$AUT
$BAS
$BDA
$BDD
$BEL
$BIN
$CDW
$CDW
$CHA
$CKS
$CMT
$CM
$CMA
$CNT
$CNT
$CNT
$CPU
$CRL
$DBL
$DDW
$DDW
$DDW
$DDW
$DDW
```

		7049	7054	7059	7064	7067	7072	7076	7080	7084	7089	7093	7096	7100
		7103	7107	7111	7114	7118	7121	7125	7129	7132	7136	7140	7143	7146
		7150	7154	7158	7162	7166	7170	7175	7178	7183	7187	7191	7197	7202
		7206	7209	7212	7217	7222	7226	7230	7234	7238	7242	7246	7250	7255
		7259	7265	7269	7273	7277	7280	7285	7289	7293	7296	7299	7303	7307
		7311	7315	7319	7324	7328	7332	7336	7339	7343	7347	7353	7358	7364
		7369	7374	7380	7384	7390	7394	7398	7402	7407	7412	7416	7422	7426
		7430	7434	7439	7444	7448	7454	7458	7462	7466	7471	7475	7478	7483
		7487	7492	7496	7500	7504	7509	7514	7518	7523	7571#			
EMS52	043216	7236	7240	7248	7253	7571#								
EMS53	043257	7258	7502	7571#										
EMS54	043320	7262	7268	7271	7275	7571#								
EMS55	043375	7279	7571#											
EMS56	043453	7282	7286	7288	7361	7571#								
EMS57	043546	7291	7571#											
EMS6	040330	6933	6937	6942	6947	7571#								
EMS60	043632	7291	7571#											
EMS600	052023	7221	7224	7228	7232	7571#								
EMS601	052053	7236	7571#											
EMS602	052073	7240	7244	7571#										
EMS603	052134	7313	7571#											
EMS604	052155	7377	7382	7387	7392	7396	7400	7405	7571#					
EMS605	052202	7409	7414	7419	7424	7428	7432	7437	7571#					
EMS606	052220	7441	7446	7451	7456	7460	7464	7469	7473	7571#				
EMS607	052240	6900	7480	7485	7490	7494	7498	7502	7507	7511	7516	7521	7571#	
EMS61	043704	7298	7571#											
EMS62	043757	7301	7305	7309	7313	7571#								
EMS63	044040	7317	7322	7326	7330	7334	7338	7341	7345	7392	7424	7456	7571#	
EMS64	044100	7350	7571#											
EMS65	044150	7356	7363	7571#										
EMS66	044211	7367	7372	7377	7387	7409	7419	7441	7451	7477	7480	7511	7571#	
EMS67	044266	7405	7437	7473	7507	7571#								
EMS7	040374	6935	6937	7571#										
EMS70	044400	6822	6978	7469	7521	7571#								
EMS71	044446	6816	6819	7571#										
EMS72	044525	6816	6819	7571#										
EMS73	044602	6822	6826	6830	6834	6842	6870	7571#						
EMS74	044652	6822	7571#											
EMS75	044725	6826	6830	7571#										
EMS76	045001	6834	6842	7571#										
EMS77	045042	6826	6830	7571#										
EMTVEC=	000030	1041#	2218*	2219*	2489*									
EMT1	031744	1630	6770#											
EMT10	032132	1687	6792#											
EMT100	033602	6969#												
EMT101	033622	6973#												
EMT102	033642	6978#												
EMT103	033654	6980#												
EMT104	033700	6985#												
EMT105	033720	6990#												
EMT106	033740	6995#												
EMT107	033764	7000#												
EMT11	032150	1695	6795#											
EMT110	034004	7005#												
EMT111	034026	7009#												
EMT112	034046	7013#												

CZRM
CZRM

$DDW
$DDW
$DDW
$DEV
$DEV
$DOA
$DTB
$END
$END
$ENU
$ENV
$ENV
$EOP
$EOP
$EOS
$ERF
$ERM

$ERR
$ERR
$ERR
$ERT
$ESC
$ETA
$ETE
$FAT
$FFL
$FIL
$FIL
$GDA
$GDD

$GET
$GTS
$HD
$HIB
$HIO
$ICN

$ILL
$INT
$ITE
$LF
$LFL
$LLC
$LLV
$LPA

$LPC
$LPC
$LPE

```
EMT113  034066  7018#
EMT114  034112  7023#
EMT115  034134  7027#
EMT116  034160  7032#
EMT117  034200  7037#
EMT12   032166  1703    6798#
EMT120  034220  7042#
EMT121  034244  7047#
EMT122  034264  7052#
EMT123  034304  7057#
EMT124  034330  7062#
EMT125  034346  7066#
EMT126  034366  7070#
EMT127  034404  7074#
EMT13   032204  1711    6801#
EMT130  034430  7079#
EMT131  034446  7083#
EMT132  034464  7087#
EMT133  034504  7091#
EMT134  034524  7095#
EMT135  034540  7098#
EMT136  034556  7102#
EMT137  034570  7105#
EMT14   032222  1719    6804#
EMT140  034606  7109#
EMT141  034626  7113#
EMT142  034642  7116#
EMT143  034660  7120#
EMT144  034674  7123#
EMT145  034716  7127#
EMT146  034740  7131#
EMT147  034756  7135#
EMT15   032240  1727    6807#
EMT150  034770  7138#
EMT151  035012  7142#
EMT152  035024  7145#
EMT153  035040  7148#
EMT154  035060  7152#
EMT155  035100  7156#
EMT156  035116  7160#
EMT157  035140  7164#
EMT16   032256  1735    6810#
EMT160  035156  7168#
EMT161  035206  7173#
EMT162  035224  7177#
EMT163  035242  7181#
EMT164  035264  7185#
EMT165  035306  7189#
EMT166  035332  7195#
EMT167  035352  7200#
EMT17   032274  1743    6813#
EMT170  035374  2125    7205#
EMT171  035406  7208#
EMT172  035422  2141    7211#
EMT173  035440  2149    7215#
EMT174  035462  2157    7219#
```

```
CZRM
CZRM

$LPV
$MADR
$MADR
$MADR
$MADR
$MAIL

$MAM
$MAM
$MAM
$MAM
$MBA
$MFL
$MNE
$MSG
$MSG
$MSG
$MSW
$MTY
$MTY
$MTY
$MTY
$MXC
$NUL
$NWT

$OCN
$OMO
$OVE
$PAS
$PAS
$POW
$PSW
$PWR
$PWR
$PWR
$QUE
$RDC
$RDD
$RDL
$RDO
$RDS
$RES
$RTN
$R2A
$SAV
$SAV
$SCO
$SET

$STU
$SVL
```

EMT175	035510	2165	7224#
EMT176	035526	2173	7228#
EMT177	035544	2181	7232#
EMT2	031752	1638	6771#
EMT20	032312	1751	6816#
EMT200	035562	2189	7236#
EMT201	035600	7240#	
EMT202	035616	7244#	
EMT203	035634	7248#	
EMT204	035656	7253#	
EMT205	035676	7258#	
EMT206	035716	7262#	
EMT207	035746	7268#	
EMT21	032334	1759	6819#
EMT210	035762	7271#	
EMT211	036000	7275#	
EMT212	036016	7279#	
EMT213	036030	7282#	
EMT214	036060	7288#	
EMT215	036072	7291#	
EMT216	036112	7295#	
EMT217	036126	7298#	
EMT22	032356	1767	6822#
EMT220	036142	7301#	
EMT221	036160	7305#	
EMT222	036176	7309#	
EMT223	036214	7313#	
EMT224	036232	7317#	
EMT225	036252	7322#	
EMT226	036274	7326#	
EMT227	036312	7330#	
EMT23	032402	1775	6826#
EMT230	036330	7334#	
EMT231	036346	7338#	
EMT232	036362	7341#	
EMT233	036404	7345#	
EMT234	036426	7350#	
EMT235	036454	7356#	
EMT236	036474	7361#	
EMT237	036526	7367#	
EMT24	032426	1783	6830#
EMT240	036546	7372#	
EMT241	036572	7377#	
EMT242	036616	7382#	
EMT243	036642	7387#	
EMT244	036666	7392#	
EMT245	036704	7396#	
EMT246	036722	7400#	
EMT247	036746	7405#	
EMT25	032452	1791	6834#
EMT250	036764	7409#	
EMT251	037010	7414#	
EMT252	037034	7419#	
EMT253	037060	7424#	
EMT254	037076	7428#	
EMT255	037114	7432#	

```
EMT256  037140            7437#
EMT257  037156            7441#
EMT26   032472            1799    6837#
EMT260  037202            7446#
EMT261  037226            7451#
EMT262  037252            7456#
EMT263  037270            7460#
EMT264  037306            7464#
EMT265  037332            7469#
EMT266  037350            7473#
EMT267  037366            7477#
EMT27   032524            1807    6842#
EMT270  037402            7480#
EMT271  037426            7485#
EMT272  037452            7490#
EMT273  037470            7494#
EMT274  037506            7498#
EMT275  037524            2133    7502#
EMT276  037544            1967    7507#
EMT277  037562            7511#
EMT3    032000            1646    6775#
EMT30   032544            1815    6845#
EMT300  037606            7516#
EMT301  037632            7521#
EMT31   032564            1823    6849#
EMT32   032600            1831    6852#
EMT33   032612            1839    6854#
EMT34   032624            1847    6856#
EMT35   032636            1855    6858#
EMT36   032650            1863    6860#
EMT37   032670            1871    6863#
EMT4    032020            1654    6778#
EMT40   032720            1879    6867#
EMT41   032742            1887    6870#
EMT42   032770            1895    6874#
EMT43   033006            1903    1911    6877#
EMT44   033022            6879#
EMT45   033040            1919    6882#
EMT46   033054            1927    6885#
EMT47   033070            1935    6887#
EMT5    032042            1662    6782#
EMT50   033106            1943    6890#
EMT51   033124            6893#
EMT52   033140            1959    6895#
EMT53   033156            6898#
EMT54   033172            1975    6900#
EMT55   033210            1951    1983    6903#
EMT56   033226            1991    6906#
EMT57   033242            1999    6909#
EMT6    032066            1670    6786#
EMT60   033256            2007    6913#
EMT61   033272            2015    6917#
EMT62   033310            2023    6920#
EMT63   033326            2031    6923#
EMT64   033340            2039    6925#
EMT65   033354            2047    6927#
```

```
EMT66   033366            2054    6930#
EMT67   033402            2061    6933#
EMT7    032110            1679    6789#
EMT70   033420            2068    6937#
EMT71   033440            2075    6942#
EMT72   033464            2082    6947#
EMT73   033510            2089    6952#
EMT74   033530            2096    6957#
EMT75   033540            2103    6960#
EMT76   033552            2110    6963#
EMT77   033566            2117    6966#
ENBSCH  023352            3321    3410    3488    3667    3803    3918    3986    4138    4335    4631    4891    5498#
EQUALS  030753            6720#
ERR   = 040000            1116#
ERRNMB  022444            5203*   5204*   5207    5219    5224    5316#
ERRTYP  021706            5182#   6247
ERRVEC= 000004            1034#   2232    2233*   2244*   5336    5337    5338*   5339*   5354*   5375*   5376*   6145    6146*
                          6148*   6151*
ERTY00  022452            5189    5321#
ERTY01  022460            5197    5322#
ERTY02  022470            5206    5324#
ERTY03  022477            5212    5326#
ERTY04  022505            5308    5328#
ESRC  = 004000            1191#   5655    5659
FER   = 000020            1140#   1146    4580
FMT16 = 010000            1216#   2634    3315    3404    3548    3550    3659    3797    3912    3980    4129    4327    4617
                          4884
FNCDTB  031634            6724#
FNCMSK= 000077            1058#
F0    = 000002            1056#
F1    = 000004            1055#
F2    = 000010            1054#
F3    - 000020            1053#
F4    = 000040            1052#
GNS   = ****** U          1369    2274    5141    5148    6602    6603    6604    6605    6606    6607    6609    6611    6612
                          6613    6614    6615    6616
GO    - 000001            1057#   3318    3407    3485    3662    3800    3915    3983    4135    4328    4622    4888    5450
                          5483    5570
GTSWR = 104407            2269    2485    6609#
HCE   = 000200            1137#   1146    4580    6746    6747    6750    6754    6755
HCI   = 002000            1218#   2634    4617    4884
HCRC  = 000400            1136#   1146    4580
HELP    054116            2346    7608#
HELPQS  030763            2340    6720#
HT    = 000011             944#   5245    6083    6124
IAE   = 002000            1134#   6728    6738    6746    6747    6750    6751    6754    6755
IDXMSK= 000077            1297#
IE    = 000100            1311#   1344#
ILF   = 000001            1144#   6727    6736    6737    6739    6740    6741    6742    6743    6744    6745    6748    6749
                          6752    6753    6756    6757
ILF02 = 000002            1063#
ILF24 = 000024            1073#
ILF26 = 000026            1074#
ILF30 = 000030            1076#
ILF32   000032            1077#
ILF34 = 000034            1078#
```

```
ILF36 = 000036           1079#
ILF40 = 000040           1080#
ILF42 = 000042           1081#
ILF44 = 000044           1082#
ILF46 = 000046           1083#
ILF54 = 000054           1086#
ILF56 = 000056           1087#
ILF64 = 000064           1090#
ILF66 = 000066           1091#
ILF74 = 000074           1094#
ILF76 = 000076           1095#  2628  2649  2681  2697  2771  2808  2844  2869  3109  3116
ILR   = 000002           1143#
ILRG50= 000050           1283#
ILRG52= 000052           1284#
ILRG54= 000054           1285#
ILRG56= 000056           1286#
ILRG60= 000060           1287#
ILRG62= 000062           1288#
ILRG64= 000064           1289#
ILRG66= 000066           1290#
ILRG70= 000070           1291#
ILRG72= 000072           1292#
ILRG74= 000074           1293#
ILRG76= 000076           1294#
IOTVEC= 000020           1039#  2216*  2217*
IPCK0 = 000001           1348#
IPCK1 = 000002           1347#
IPCK2 = 000004           1346#
IPCK3 = 000010           1345#
IR    = 000100           1325#  2549  2570
IVC   = 010000           1253#  6727  6728  6729  6731  6732  6733  6736  6737  6738  6739  6740  6741
                         6742   6743  6744  6745  6746  6747  6748  6749  6750  6751  6752  6753  6754
                         6755   6756  6757
LBC   = 002000           1255#
LBT   = 002000           1120#
LCLOCK  022766           5356   5385#
LCOUNT  023036           5358   5403#
LF    = 000012            945#  5243  6118  6124  6720  7571  7608
LS    = 000004           1200#
LSC   = 004000           1254#
LST   = 000002           1201#
LSTOP   023100           5357   5417#
MCLK    004000           1172#  3578  3593  3610  3627  3685  3711  3728  3750  3762  3832  3851  3878
                         3881   3946  4032  4045  4073  4097  4157  4184  4212  4233  4260  4286  4358
                         4384   4410  4436  4458  4483  4494  4502  4517  4525  4540  4566  4760  4785
                         4819   4842  4971  4992  4995  5066  5090  5523  5541  5717  5728  5748  5758
                         5790   5819  5822
MCPE  - 020000           1305#
MDF   - 000100           1177#
MDPE    000400           1322#
MI    = 000004           1180#  3393  3478  3518  5561  5679
MOC   = 000400           1175#  1183  3346  3418  3420  3433  3435  3437  3440  3442  3496  3498  3500
                         3502   3518  3521  3523  3525  3528  3530  3685  3687  3711  3713  3728  3730
                         3750   3752  3762  3764  5598  5631  5633  5642  5648  5650  5679  5681  5691
                         5693   5696  5698  5700  5702  5717  5719  5728  5730
MOH   = 020000           1208#
```

```
MOL    = 010000          1118#
MRD    - 002000          1173#  3829  3832  3834  3874  4498  4521  4971  4973  4988  5814
MR1AAA= 051401           1183#  3829  3832  3834  3851  3853  3870  3941  3946  3948  4032  4034  4045
                         4047   4073  4075  4097  4099  4157  4159  4184  4186  4212  4214  4233  4235
                         4260   4262  4286  4288  4358  4360  4384  4386  4410  4412  4436  4438  4458
                         4460   4483  4485  4494  4517  4540  4542  4566  4568  4760  4762  4785  4787
                         4819   4821  4842  4844  4971  4973  4984  5066  5068  5090  5092  5748  5750
                         5758   5760  5790  5792  5810
MS     - 000040          1178#  3418  3435  3440  3498  3523  3528  3941  3946  3948  5691  5698
MSC    = 000002          1181#  3433  3435  3437  3496  3498  3500  3521  3523  3525  5696  5698  5700
MSE    = 100000          1262#
MSER   = 000200          1176#
MUR    = 001000          1174#  1183  3346  3418  3420  3433  3435  3437  3440  3442  3496  3498  3500
                         3502   3518  3521  3523  3525  3528  3530  3685  3687  3711  3713  3728  3730
                         3750   3752  3762  3764  5444  5477  5523  5525  5541  5543  5561  5563  5598
                         5631   5633  5640  5642  5648  5650  5679  5681  5691  5693  5696  5698  5700
                         5702   5717  5719  5728  5730
MWD    = 000010          1199#  4064  4091  4228  4281  4780
MWP    - 000010          1179#
MXF    = 001000          1321#
NDTMSK= 115760           1146#
NED    = 010000          1318#  2296  2545  2560  2566  2579
NEM    = 004000          1319#
NOP    = 000000          1062#
NOTEX    031607          2304   6720#
NSA    - 100000          1206#
OCC    = 100000          1187#
OFD    = 000200          1219#  2634
OFFSET= 000014           1068#  5483
OM     = 000001          1125#  5486  5488
OPE    = 020000          1252#
OPI    = 020000          1131#  3337  3355  3357  6726  6727  6728  6729  6730  6731  6732  6733  6734
                         6735   6736  6737  6738  6739  6740  6741  6742  6743  6744  6745  6746  6747
                         6748   6749  6750  6751  6752  6753  6754  6755  6756  6757
OR     = 000200          1324#
PACACK= 000022           1072#  5450  5460  5570  5580
PAKACK= 000022           1071#  1072
PAR    = 000010          1141#
PAT    - 000020          1327#
PCLOCK   022750          5343   5381#
PCOUNT   023036          5345   5402#
PDA    = 000400          1194#  4670  4709  4718  4732  4733  4736  4755  5010  5014
PGE    = 002000          1320#
PGM    - 001000          1121#
PHA    = 000200          1195#  4170  4175  4242  4246  4670  4805  4807
PIP    = 020000          1117#
PIRQ   = 177772           951#
PIRQVE= 000240           1045#
PLCLK    022774          5383   5387#
PLFS   = 002000          1192#  3696  3736  3738  4470  4473  4552  4558  4924  4930  4944  4949  4952
                         5733   5776  5779  5780
PLSTP    023104          5415   5419#
PROMPT   030755          2421   2430  6720#
PR0    = 000000           968#
PR1    = 000040           969#
PR2    = 000100           970#
```

```
PR3    = 000140   971#  2251   2489
PR4    = 000200   972#
PR5    = 000240   973#  5394
PR6    = 000300   974#  5339   5346   5359   5392
PR7    = 000340   975#  5391
PS     = 177776   948#   949
PSEL   = 002000  1307#
PSTOP    023072  5344   5414#
PSW    = 177776   949#
PWRVEC= 000024  1040#  2222*  2223*  6621*  6622*  6631*  6637*  6649*  6650*
QSTMRK  030761  2335   2440   6720#
RD     = 000070  1092#  3800   3915   4888
RDCHR  = 104411  2327   2341   2351   2354   2422   2431   6474   6612#
RDLIN  = 104412  6544   6613#
RDOCT  = 104413  2365   2381   2402   6614#
RDY    = 000200  1310#
READY    004414  2332   2490#  5116   5164
RECAL  = 000006  1065#
RESREG= 104415  5312   6616#
RESVEC= 000010  1035#
REX    = 010000  1190#
RG     = 040000  1188#  5612   5617
RH     = 000072  1093#  4328
RIP    = 000020  1070#
RLEASE= 000012  1067#
RMAS   = 000016  1271#  2593   3047*
RMASI    001344  1541#
RMASO    001420  1571#
RMBA   = 000004  1353#  2589   5594*
RMBAE  = 000050  1356#
RMBAEI   001376  1554#
RMBAEO   001452  1584#
RMBAI    001332  1536#
RMBAO    001406  1566#  3316*  3405*  3483*  3660*  3798*  3913*  3981*  4131*  4325*  4619*  4886*  5594
RMCS1  = 000000  1267#  1351#  2298   2628*  2639   2681*  2691   2771*  2782*  2795   2844*  2857   3109*
                 3289   5450*  5483*  5570*  5604*
RMCS1I   001326  1534#  2639*  2649*  2691*  2697*  2699   2795*  2808*  2812*  2818   2857*  2869*  2874
RMCS1O   001402  1564#  3318*  3407*  3485*  3662*  3800*  3915*  3983*  4135*  4328*  4622*  4888*  5604
RMCS2  = 000010  1354#  2293*  2294*  2295   2541*  2542*  2544   2551   2560   2562*  2563*  2565   2572
                 2579   2591   2624*  2625*  2677*  2678*  2687*  2688*  2725*  2726*  2767*  2768*  2832*
                 2833*  2893*  2894*  3015*  3016*  3038*  3039*  3070*  3071*  3102*  3103*  3126*  3127*
                 3158*  3159*  3181*  3182*  3203*  3204*  3286*  3287*  3380*  3381*  3471*  3472*  3568*
                 3569*  3663*  3664*  4124*  4125*  4331*  4332*  4612*  4613*  4879*  4880*  5439*  5440*
                 5514*  5515*
RMCS2I   001336  1538#
RMCS2O   001412  1568#
RMCS3  = 000052  1357#
RMCS3I   001400  1555#
RMCS3O   001454  1585#
RMDA     000006  1268#  2630*  2641   2773*  2784*  2797   2837*  2846*  2859   2897*  2908   3041*  3053
                 3073*  3085   3161*  3169   3186*  3194   3216*  5586*
RMDAI    001334  1537#  2641*  2651   2797*  2813*  2819   2859*  2875   2908*  2920   3053*  3058   3085*
                 3090   3169*  3172   3194*  3197
RMDAO    001410  1567#  3313*  3402*  3480*  3657*  3795*  3910*  3978*  4127*  4324*  4615*  4882*  5586
RMDB   = 000022  1355#  2595
RMDBI    001350  1543#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
RMDB0   001424          1573#
RMDC  = 000034          1277#  2632*  2643   2790*  2803   2841*  2850*  2863   2901*  2912   3020*  3028   3107*
                        3115   3133*  3145   3188*  3208*  3220   5588*
RMDCI   001362          1548#  2643*  2653*  2803*  2810*  2816*  2822   2863*  2871*  2877   2912*  2930   3028*
                        3031*  3032   3115*  3119*  3120   3145*  3151*  3152   3220*  3225   3226
RMDCO   001436          1578#  3314*  3403*  3481*  3658*  3796*  3911*  3979*  4128*  4323*  4616*  4883*  5588
RMDS  = 000012          1269#  3165*  3212*  5452   5456   5485   5490   5572   5576
RMDSI   001340          1539#
RMDSO   001414          1569#
RMDT  = 000026          1274#  3135*  3253   3265
RMDTI   001354          1545#
RMDTO   001430          1575#
RMEC1 = 000044          1281#  3139*
RMEC1I  001372          1552#
RMEC1O  001446          1582#
RMEC2 = 000046          1282#  2597   3083*
RMEC2I  001374          1553#
RMEC2O  001450          1583#
RMER1   000014          1270#  2683*  2693   2729*  2736   2775*  2786*  2799   2852*  2865   2903*  2914   3018*
                        3026   3105*  3113   3129*  3141   3184*  3192   3214*  3334   3336   3354   3377   3383
                        3395   3425   3444   3505   3509   3532   3536   4579   4584   5036   5043   5446*  5479*
                        5566*  5600*
RMER1I  001342          1540#  2693*  2701   2736*  2741*  2799*  2814*  2820   2865*  2878   2914*  2936   3026*
                        3029   3113*  3116*  3117   3141*  3146   3192*  3195
RMER1O  001416          1570#
RMER2 - 000042          1280#  2685*  2695   2731*  2738   2779*  2792*  2805   2854*  2867   2905*  2916   3045*
                        3057   3077*  3089   3137*  3163*  3171   3190*  3210*  3222   5448*  5481*  5568*  5602*
RMER2I  001370          1551#  2695*  2698*  2703   2738*  2743*  2805*  2811*  2817*  2823   2867*  2872*  2879
                        2916*  2940   3057*  3063*  3064   3089*  3095*  3096   3171*  3174*  3175   3222*  3228*
                        3229
RMER2O  001444          1581#
RMHR  = 000036          1278#  3024*  3081*
RMHRI   001364          1549#
RMHRO   001440          1579#  7587
RMLA  = 000020          1272#  3079*
RMLAI   001346          1542#
RMLAO   001422          1572#
RMMR1   000024          1273#  2733*  2740   3022*  3346*  3391*  3393*  3418*  3420*  3433*  3435*  3437*  3440*
                        3442*  3476*  3478*  3496*  3498*  3500*  3502*  3518*  3521*  3523*  3525*  3528*  3530*
                        3571   3574*  3578*  3580*  3582   3593*  3595*  3597   3610*  3612*  3614   3627*  3629*
                        3631   3685*  3687*  3690   3700   3711*  3713*  3716   3728*  3730*  3735   3740   3750*
                        3752*  3762*  3764*  3766   3772   3826   3829*  3832*  3834*  3836   3851*  3853*  3855
                        3877*  3880*  3883*  3885   3941*  3946*  3948*  3950   3957   4032*  4034*  4036   4045*
                        4047*  4049   4063   4073*  4075*  4089   4097*  4099*  4157*  4159*  4161   4178   4184*
                        4186*  4188   4212*  4214*  4226   4233*  4235*  4245   4256   4260*  4262*  4267   4279
                        4286*  4288*  4358*  4360*  4362   4384*  4386*  4388   4410*  4412*  4414   4436*  4438*
                        4440   4458*  4460*  4462   4475   4483*  4485*  4501*  4504*  4524*  4527*  4540*  4542*
                        4544   4560   4566*  4568*  4570   4669   4676   4708   4721   4731   4738   4754   4760*
                        4762*  4768   4778   4785*  4787*  4819*  4821*  4823   4831   4842*  4844*  4846   4854
                        4923   4933   4943   4954   4971*  4973*  4991*  4994*  4997*  5009   5018   5066*  5068*
                        5070   5078   5090*  5092*  5094   5102   5442*  5444*  5477*  5517*  5523*  5525*  5527
                        5533   5541*  5543*  5545   5551   5561*  5563*  5598*  5611   5619   5631*  5633*  5640*
                        5642*  5648*  5650*  5654   5658   5679*  5681*  5691*  5693*  5696*  5698*  5700*  5702*
                        5717*  5719*  5721   5728*  5730*  5732   5744   5748*  5750*  5754   5758*  5760*  5775
                        5782   5790*  5792*  5794   5802   5818*  5821*  5824*
RMMR1I  001352          1544#  2740*  2745
```

```
RMMR10  001426  1574# 3870* 3874* 3877  3878* 3880  3881* 3883  4494* 4498* 4501  4502* 4504
                4517* 4521* 4524  4525* 4527
RMMR2 = 000040  1279# 3051* 4005  4010  4019  4025  4197  4202  4349  4373  4423  4682  4689
                5023  5029
RMMR2I  001366  1550#
RMMR20  001442  1580#
RMOF  = 000032  1276# 2634* 2645  2777* 2788* 2801  2839* 2848* 2861  2899* 2910  3043* 3055
                3075* 3087  3131* 3143  3167* 3206* 3218  3474* 5590*
RMOFI   001360  1547# 2645* 2655* 2801* 2809* 2815* 2821  2861* 2870* 2876  2910* 2924  3055*
                3060* 3061  3087* 3092* 3093  3143* 3148* 3149  3218* 3223*
RMOFO   001434  1577# 3315* 3404* 3482* 3659* 3797* 3912* 3980* 4129* 4327* 4617* 4884* 5590
RMR   = 000004  1142#
RMSN  = 000030  1275# 3049* 3111*
RMSNI   001356  1546#
RMSNO   001432  1576#
RMWC  - 000002  1352# 2587  5592*
RMWCI   001330  1535#
RMWCO   001404  1565# 3317* 3406* 3484* 3661* 3799* 3914* 3982* 4134* 4326* 4621* 4887* 5592
RQA   - 100000  1230#
RQB   = 040000  1231#
RTC   = 000016  1069#
SADMSK= 000037  1111#
SAVREG= 104414  5183  6615#
SA1   = 000001  1106#
SA16  - 000020  1102#
SA2   = 000002  1105#
SA4   = 000004  1104#
SA8   = 000010  1103#
SC    = 100000  1303#
SCTCMP  024232  3673  3809  3924  3993  4144  4341  4637  4897  5674#
SCTMSK= 003700  1162#
SC0   - 000100  1160#
SC1   = 000200  1159#
SC2   = 000400  1158#
SC3   - 001000  1157#
SC4   = 002000  1156#
SEARCH- 000030  1075#
SEEK  - 000004  1064#
SETLFS  024340  3815  3930  4643  4903  5713#
SETOM   023250  5475#
SETVV   023120  5438#
SIZCLK  022510  2472  5335#
SKI   = 040000  1251# 2743
SNGPRT= 020024  1211# 3254  3258  3262
STACK - 001100   939# 2214  2535  2621  2674  2722  2763  2977  3249  3283  3309  3373  3464
                3565  3653  3791  3906  3974  4121  4317  4609  4876
STANDA  003426  2282  2317#
START   002632  1389  2202# 6315  6398
STKLMT= 177774   950#
STOP    001530  1606# 3331  3351  5344* 5357* 5616  5625  5687
SWR     001154  1442# 2212  2234* 2236  2242* 2249* 2267  2483  5184  6140  6154  6157  6172
                6179  6238  6245  6257  6261  6318  6357  6412* 6629  6642*
SWREG   000176  1371# 2242  2267  2483  6318  6357  6380
SW0   = 000001  1003#
SW00  = 000001   993# 1003
SW01  = 000002   992# 1002
```

```
SW02  = 000004            991#   1001
SW03  = 000010            990#   1000
SW04  = 000020            989#    999
SW05  = 000040            988#    998
SW06  = 000100            987#    997
SW07  = 000200            986#    996
SW08  = 000400            985#    995
SW09  = 001000            984#    994
SW1   = 000002           1002#
SW10  = 002000            983#
SW11  = 004000            982#
SW12  = 010000            981#
SW13  - 020000            980#   5184
SW14  = 040000            979#
SW15  = 100000            978#
SW2   = 000004           1001#
SW3     000010           1000#
SW4   = 000020            999#
SW5   = 000040            998#
SW6   = 000100            997#
SW7   = 000200            996#
SW8   = 000400            995#
SW9   = 001000            994#
TADMSK= 003400           1110#
TAG   - 020000           1232#
TAGADR= 001114           1414#   1421
TAP   = 040000           1207#
TA1   = 000400           1101#
TA2   = 001000           1100#
TA4   = 002000           1099#
TBITVE= 000014           1036#
TIME    001522           1603#   5387*  5404*  5406*
TKVEC = 000060           1043#   6295*  6296*
TPVEC = 000064           1044#
TRAPVE= 000034           1042#   2220*  2221*
TRE   = 040000           1304#
TRTVEC= 000014           1037#
TST   = 010000           1233#
TSTNMB  022442           5195*   5196*  5198   5315#
TSTQUE  001456           1593#   2455   2456*  2492   2537   2623   2676   2724   2765   2979   3251   3285   3311
                         3375    3466   3567   3655   3793   3908   3976   4123   4319   4611   4878   5111   5113*
                         5117*
TST1    004430           2526#   6199
TST10   010324           3274#   6206
TST11   010446           3301#   6207
TST12   010746           3365#   6208
TST13   011334           3456#   6209
TST14   011760           3557#   6210
TST15   012344           3645#   6211
TST16   013056           3783#   6212
TST17   013532           3898#   6213
TST2    005004           2613#   6200
TST20   014004           3966#   6214
TST21   014526           4113#   6215
TST22   015506           4309#   6216
TST23   016756           4601#   6217
```

```
TST24   020202          4868#   6218
TST3    005224          2665#   6201
TST4    005440          2713#   6202
TST5    005626          2755#   6203
TST6    006764          2968#   6204
TST7    010144          3240#   6205
TYPBN = 104406          5305    6607#
TYPDS = 104405          5145    5152    5299    6606#
TYPE  = 104401          2261    2304    2326    2329    2335    2340    2343    2346    2350    2353    2361    2364    2370
                        2375    2380    2386    2396    2401    2407    2420    2421    2424    2430    2433    2440    5139
                        5146    5153    5188    5189    5197    5206    5212    5221    5259    5265    5269    5282    5288
                        5290    5308    5310    5893    5959    6028    6088    6240    6248    6312    6324    6378    6379
                        6382    6393    6403    6414    6433    6481    6487    6492    6496    6501    6502    6504    6507
                        6511    6602#   6651
TYPOC = 104402          2363    5302    6381    6603#
TYPON = 104404          6605#
TYPOS = 104403          2306    2377    2398    5192    5200    5209    5215    6604#
T1      004472          2532    2533    2534#
T10     010366          3280    3281    3282#
T11     010510          3306    3307    3308#
T12     011010          3570    3371    3372#
T13     011376          3461    3462    3463#
T14     012022          3562    3563    3564#
T15     012406          3650    3651    3652#
T16     013120          3788    3789    3790#
T17     013574          3903    3904    3905#
T2      005046          2618    2619    2620#
T20     014046          3971    3972    3973#
T21     014570          4118    4119    4120#
T22     015550          4314    4315    4316#
T23     017020          4606    4607    4608#
T24     020244          4873    4874    4875#
T3      005266          2671    2672    2673#
T4      005502          2719    2720    2721#
T5      005670          2760    2761    2762#
T6      007026          2974    2975    2976#
T7      010206          3246    3247    3248#
UBUSQS  031017          2350    6720#
UNS   = 040000          1130#   2741
UNTMSK= 000007          1335#   2548    2569
UPE   = 020000          1317#
USE     040000          1263#
U0    = 000001          1332#
U1      000002          1331#
U2    - 000004          1330#
VV    = 000100          1124#   5453    5457    5573    5577
WATCH   001524          1604#   3327*   3329    3347*   3349    5407*   5409*   5607*   5614    5683*   5685
WC      000040          1197#   3583    3598    3602    3615    3618    3632    3691    3717    3767    3837    3856    3886
                        7890    3951    3955    4037    4050    4162    4189    4363    4389    4415    4441    4463    4545
                        4571    4824    4829    4847    5071    5076    5095    5528    5535    5546    5552    5722    5745
                        5755    5787    5799
WCD   - 000050          1084#
WCE   - 040000          1316#   6746    6747
WCEHI = 010000          1341#
WCELO = 004000          1342#
WCF   = 000040          1139#   1146
```

```
WCH    = 000052          1085#
WD     = 000060          1088#  3318   3407   3485   3662   4622
WH     = 000062          1089#  3983   4135
WLE    = 004000          1133#  1146   6750   6751
WRL    = 004000          1119#
XNUDC  = 176000          1225#  2653   2810   2850   2871   2931   2934   3031   3119   3151   3225
XNUER2= 001567           1258#  2698   2811   2854   2872   2941   2944   3063   3095   3174   3228
XNUOF  = 161577          1220#  2655   2809   2848   2870   2925   2928   3060   3092   3148   3223
XSIZ     003304          2286#  2427
$APTHD   001100          1401   1407#
$ASTAT= ****** U         6694   6709
$ATYC    030530          6665   6667#
$ATY1    030504          6663#
$ATY3    030512          6073   6664#
$ATY4    030522          6253   6666#
$AUTOB   001150          1439#  2271*  2487*  6371   6528
$BASE    001276          1512#  2292   2362   2372*  2536   2603   2622   2675   2723   2764   2978   3250   3284
                         3310   3374   3465   3566   3654   3792   3907   3975   4122   4318   4610   4877
$BDADR   001136          1434#  2550*  2551*  2571*  2572*  2603*  3264*  3265*  3293*  3335*  3336*  3376*  3377*
                         3508*  3509*  3535*  3536*  3570*  3571*  3699*  3700*  3739*  3740*  3771*  3772*  3825*
                         3826*  3956*  3957*  4009*  4010*  4024*  4025*  4177*  4178*  4201*  4202*  4244*  4245*
                         4266*  4267*  4348*  4349*  4474*  4475*  4559*  4560*  4583*  4584*  4675*  4676*  4688*
                         4689*  4720*  4721*  4737*  4738*  4767*  4768*  4830*  4831*  4853*  4854*  4932*  4933*
                         4953*  4954*  5017*  5018*  5028*  5029*  5042*  5043*  5059*  5077*  5078*  5101*  5102*
                         5455*  5456*  5489*  5490*  5532*  5533*  5550*  5551*  5575*  5576*  5618*  5619*  5657*
                         5658*  5781*  5782*  5801*  5802*  7573   7575   7581   7589   7591   7593
$BDDAT   001142          1436#  2544*  2545   2565*  2566   2826*  2883*  2954*  3253*  3254   3256   3258   3260
                         3289*  3290*  3334*  3337*  3354*  3355*  3383*  3384*  3395*  3396*  3425*  3426*  3444*
                         3445*  3447*  3505*  3506*  3532*  3533*  3582*  3583*  3597*  3598*  3614*  3615*  3631*
                         3632*  3695*  3696*  3735*  3736*  3770*  3836*  3837*  3855*  3856*  3885*  3886*  3950*
                         3951*  4005*  4006*  4019*  4020   4104*  4105   4176*  4197*  4198*  4199   4269*  4376*
                         4427*  4472*  4557*  4579*  4580*  4669*  4670*  4672   4682*  4683*  4685   4719*  4735*
                         4765*  4828*  4851*  4929*  4930*  4950*  4952*  5015*  5026*  5041*  5057*  5075*  5099*
                         5452*  5453*  5485*  5486*  5534*  5552*  5572*  5573*  5611*  5612*  5654*  5655*  5778*
                         5779*  5798*  5799*  7573   7576   7581   7585   7587   7589   7591   7593
$BELL    001212          1459#  6240   6273   6324   6521
$BIN     025140          5889*  5892*  5893   5900#
$CDW1    001302          1514#
$CDW2    001304          1515#
$CHARC   026072          6090*  6100*  6107   6116*  6121#
$CKSWR   027156          6357#  6611
$CMTAG   001114          1422#  2209   2210   2218   2224   2225   2226
$CM3   = 000000          1452#
$CM4   = 000005          1452#  1453#  1454#  1455#  1456#  1457#
$CNTLC   030054          6312   6393   6521#
$CNTLG   030066          6378   6523#
$CNTLU   030061          6403   6496   6522#
$CPUOP   001250          1486#
$CRLF    001217          1461#  5153   5188   5221   5259   5269   5282   5290   5310   6089   6124   6248   6273
                         6414   6501   6521
$DBLK    025356          5925   5959   5967#
$DDW0    001306          1516#
$DDW1    001310          1517#
$DDW2    001312          1518#
$DDW3    001314          1519#
$DDW4    001316          1520#
```

```
$DDW5   001320          1521#
$DDW6   001322          1522#
$DDW7   001324          1523#
$DEVCT  001232          1477#
$DEVM   001300          1513#   2289*   2301*   2419*   2428*   2446*   2447    2454
$DOAGN  021676          5135    5156    5162#
$DTBL   025346          5928    5963#
$ENDAD  021666          1380    2259    5158#   6268
$ENDCT  021532          2224    5137#
$ENULL  021702          5165#
$ENV    001242          1482#   2265    2281    2481    6068    6250    6672    6696
$ENVM   001243          1483#   2247    2286    6070    6075    6674
$EOP    021476          5127#
$EOPCT  021524          2224*   5134#   5138
$EOSP   021442          2605    3267    5109#
$ERFLG  001117          1425#   6129    6168    6170    6176*   6198    6235*   6273
$ERMAX  001131          1431#   2227*   2531*   2617*   2670*   2718*   2759*   2973*   3245*   3279*   3305*   3369*   3460*
                        3561*   3649*   3787*   3902*   3970*   4117*   4313*   4605*   4872*   6170    6193*   6198
$ERROR  026460          2218    6233#
$ERRPC  001132          1432#   5213    6242*   6243*   6244    6273
$ERRTB  001532          1624#   5228
$ERTTL  001126          1429#   5150    5154*   6241*   6273
$ESCAP  001210          1458#   2226*   6192*   6264    6266    6273
$ETABL  001242          1481#
$ETEND  001326          1413    1524#
$FATAL  001224          1474#   6700*
$FFLG   030750          6663*   6666*   6694    6703*   6711#
$FILLC  001172          1450#   6093    6124
$FILLS  001171          1449#   6124
$GDADR  001134          1433#
$GDDAT  001140          1435#   2547*   2548*   2549*   2568*   2569*   2570*   2827*   2884*   2953*   3292*   3339*   3357*
                        3378*   3510*   3537*   3587*   3602*   3618*   3634*   3698*   3738*   3769*   3827*   3890*   3955*
                        4008*   4026*   4103*   4105    4175*   4203*   4246*   4268*   4377*   4428*   4473*   4558*   4582*
                        4671*   4672    4684*   4685    4718*   4736*   4766*   4829*   4852*   4931*   4949*   5016*   5027*
                        5038*   5039    5058*   5076*   5100*   5457*   5488*   5535*   5553*   5577*   5617*   5659*   5780*
                        5800*   7573    7576    7585    7587    7589    7591
$GET42  021656          5155#
$GTSWR  027246          6379#   6609
$HD     = 000001         916     917
$HIBTS  001100          1408#
$HIOCT  030216          6561*   6566#
$ICNT   001120          1426#   2530*   2616*   2669*   2717*   2758*   2972*   3244*   3278*   3304*   3368*   3459*   3560*
                        3648*   3786*   3901*   3969*   4116*   4312*   4604*   4871*   6183*   6184    6186*   6197
$ILLUP  030466          6621    6637    6654#
$INTAG  001151          1440#   6376*   6395    6415    6528
$ITEMB  001130          1430#   5204    6244*   6252    6273
$LF     001220          1462#   6124    6273    6511    6521
$LFLG   030747          6704*   6710#
$LLCSR  001512          1599#   5355    5385*   5417*
$LLVEC  001514          1600#   5347*   5348*   5358*   5359*   5370*   5371*
$LPADR  001122          1427#   2228*   2532*   2618*   2671*   2719*   2760*   2974*   3246*   3280*   3306*   3370*   3461*
                        3562*   3650*   3788*   3903*   3971*   4118*   4314*   4606*   4873*   6165*   6174*   6190*   6195
                        6197
$LPCSB  001504          1596#   5381*
$LPCSR  001502          1595#   5342    5382*   5414*
$LPERR  001124          1428#   2229*   2533*   2619*   2672*   2720*   2761*   2975*   3247*   3281*   3307*   3371*   3462*
```

```
                        3563*  3651*  3789*  3904*  3972*  4119*  4315*  4607*  4874*  6174   6191*  6197   6263
$LPVEC  001506          1597#  5345*  5346*  5360*  5361*  5368*  5369*
$MADR1  001254          1499#
$MADR2  001260          1503#
$MADR3  001264          1506#
$MADR4  001270          1509#
$MAIL   001222          1409   1413   1472#  2246   2265   2481   2527   2614   2666   2714   2756   2969   3241
                        3275   3302   3366   3457   3558   3646   3784   3899   3967   4114   4310   4602   4869
                        6068   6189   6250
$MAMS1  001252          1493#
$MAMS2  001256          1501#
$MAMS3  001262          1504#
$MAMS4  001266          1507#
$MBADR  001102          1409#
$MFLG   030746          6664*  6670   6705*  6709#
$MNEW   030104          6382   6526#
$MSGAD  001236          1479#  6680*  6683
$MSGLG  001240          1480#  6685*
$MSGTY  001222          1473#  6678   6686*  6698   6702*
$MSWR   030073          6379   6524#
$MTYP1  001253          1494#
$MTYP2  001257          1502#
$MTYP3  001263          1505#
$MTYP4  001267          1508#
$MXCNT  026406          6187   6197#
$NULL   001170          1448#  6095   6124
$NWTST= 000001          2522#  2524   2609#  2611   2661#  2663   2709#  2711   2751#  2753   2964#  2966   3236#
                        3238   3270#  3272   3297#  3299   3361#  3363   3452#  3454   3553#  3555   3641#  3643
                        3779#  3781   3894#  3896   3962#  3964   4109#  4111   4305#  4307   4597#  4599   4864#
                        4866
$OCNT   025610          6000*  6029*  6042#
$OMODE  025612          5995*  5999*  6004   6007*  6018*  6044#
$OVER   026372          6141   6166   6175   6185   6194#
$PASS   001230          1476#  2246*  5131*  5132*  5143   5165   6181   6198
$PASTM  001106          1411#
$POWER  030474          6652   6657#
$PSW    001520          1602#  5390*  5420
$PWRDN  030326          2222   6621#  6649
$PWRMG  030462          6652#
$PWRUP  030400          6631   6637#
$QUES   001216          1460#  6124   6273   6433   6504   6521
$RDCHR  027520          6446#  6612
$RDDEC= ****** U        6615
$RDLIN  027610          6469#  6613
$RDOCT  030116          6539#  6614
$RDSZ = 000010          6462#
$RESRE  025030          5865#  6616
$RTNAD  021700          5164#
$R2A  = ****** U        6617
$SAVRE  024772          5849#  6615
$SAVR6  030472          6630*  6638   6639*  6640*  6656#
$SCOPE  026076          2216   6138#
$SETUP= 000137          1362#  2215   2216   2218   2220   2222   2224   2225   2226   2228   2259   2262   2478
                        5129   6139   6234   6260   6268   6316   6321   6322   6352   6528
$STUP = 177777          1362#
$SVLAD  026336          6149   6188#
```

```
$SVPC  = 000200       1378#   1383
$SWR   = 167400        905#    916     921     922     923     924     925     926     927    1457    1458    1459    2225
                      2226    2228    2229    2527    2614    2666    2714    2756    2969    3241    3275    3302    3366
                      3457    3558    3646    3784    3899    3967    4114    4310    4602    4869    5124    5130    5157
                      5163    5165    6130    6131    6132    6133    6134    6140    6152    6154    6155    6168    6169
                      6170    6177    6178    6179    6191    6194    6197    6225    6226    6227    6228    6229    6238
                      6245    6257    6261    6273    6653
$SWREG   001244       1484#   2249
$SWRMK=  000000        927     928    6134    6135    6158
$SW08T   026410       6164    6198#
$TESTN   001226       1475#   2527*   2614*   2666*   2714*   2756*   2969*   3241*   3275*   3302*   3366*   3457*   3558*
                      3646*   3784*   3899*   3967*   4114*   4310*   4602*   4869*   5196    6189*
$TIMES   001206       1457#   2225*   5130*   6177*   6184    6187*   6197
$TKB     001162       1445#   6276    6297    6308    6333    6361    6388
$TKCNT   026660       6277#   6292*   6322    6339*   6453    6455*
$TKINT   026670       2319    2491    6292#   6313    6374
$TKQEN=  026667       6281#   6347    6458
$TKQIN   026662       6278#   6293*   6294    6345*   6346*   6347    6349*
$TKQOU   026664       6279#   6294*   6456    6457*   6458    6460*
$TKQSR   026666       6280#   6293    6349    6460
$TKS     001160       1444#   6276    6298*   6329*   6331    6337*   6359    6375*   6385    6397*   6417*
$TKSRV   026740       6295    6308#
$TMP0    001174       1452#   3262*   3511*   3701*   3839*   3858*   4219*   4221    4239    4296    4509    4510*   4530
                      4589    4773    4791    4795    4980*   4982    5460*   5580*   7577    7579    7581    7583    7585
                      7589    7591
$TMP1    001176       1453#   2328*   2329    2330    2333    2342*   2343    2344    2352*   2353    2356    2366*   2368
                      2372    2373*   2374*   2376    2382*   2384    2388    2389*   2390*   2391*   2392*   2393*   2394*
                      2395*   2397    2403*   2405    2410*   2411*   2412*   2413*   2414*   2415    2423*   2424    2425
                      2432*   2433    2434    2436    2438    2442    3263*   3512*   4220*   4298    4511*   4591    4797
                      4981*   5001    7577    7579    7581    7583    7591
$TMP2    001200       1454#   4300    4512*   4799*   7577    7583
$TMP3    001202       1455#   4800*   7583
$TMP4    001204       1456#
$TN    = 000025        916#   2522    2527#   2532    2609    2614#   2618    2661    2666#   2671    2709    2714#   2719
                      2751    2756#   2760    2964    2969#   2974    3236    3241#   3246    3270    3275#   3280    3297
                      3302#   3306    3361    3366#   3370    3452    3457#   3461    3553    3558#   3562    3641    3646#
                      3650    3779    3784#   3788    3894    3899#   3903    3962    3967#   3971    4109    4114#   4118
                      4305    4310#   4314    4597    4602#   4606    4864    4869#   4873    6159    6199
$TPB     001166       1447#   6113*   6124
$TPFLG   001173       1451#   6062    6124
$TPS     001164       1446#   6111    6124
$TRAP    030220       2220    6575#
$TRAP2   030260       6590#   6601
$TRP   = 000016       6594#   6603#   6604#   6605#   6606#   6607#   6608#   6609    6610#   6611    6612#   6613#   6614#
                      6615#   6616#   6617#
$TRPAD   030272       6584    6601#
$TSTM    001104       1410#
$TSTNM   001116       1424#   5129*   6129    6161*   6188*   6189    6194    6198    6237    6273
$TTYIN   030044       647[illegible] 6472 6484    6502    6516    6520#
$TYPBN   025066       5886#   6607
$TYPDS   025142       5913#   6606
$TYPE    025614       6062#   6594    6602    6691
$TYPEC   026026       6092    6099    6106    6111#   6112    6419
$TYPEX   026074       6117    6119    6122#
$TYPOC   025412       5998#   6603
$TYPON   025426       5997    6000#   6605
```

```
$TYPOS  025366            5993#   6604
$UNIT   001234            1478#   2492*   5190
$UNITM  001110            1412#
$USWR   001246            1485#
$VECT1  001272            1510#   2373    2388*   2389    2415*
$VECT2  001274            1511#
$XTSTR  026110            6143#
$$GET4= 000000            5157#
$$SW08= 000025            6198#   6199    6200#   6201#   6202#   6203#   6204#   6205#   6206#   6207#   6208#   6209#   6210#
                          6211#   6212#   6213#   6214#   6215#   6216#   6217#   6218#   6219#
$OFILL  025611            5994*   5998*   6008    6043#
$40CAT= ****** U          6140    6247
.       = 055625          1365#   1369#   1378    1379#   1381#   1383#   1388#   1391#   1397    1398#   1400#   1402#   1414
                          1421#   1463    1593#   2213    2228    2229    2275#   5142#   5165    5166#   5967#   6124    6197
                          6198    6273    6276    6280#   6281    6282#   6520#   6521    6528#   6633    6655    6712#   6722#
                          7603#   7605#   7606#   7607#
.$ASTA= ****** U          6664    6667
.$X     = 001100          1397#   1402
```

CLEAR	905#	2541	2562	2624	2677	2687	2725	2767	2832	2892	3015	3038	3070	3102	3126
	3158	3181	3203	3286	3380	3471	3568	3663	4124	4331	4612	4879	5439	5514	
CLKOFF	905#	3331	3351	5616	5624	5687									
CLKON	905#	3327	3347	5607	5683										
CLKSNC	905#	4648	4907	4959											
COMMEN	1046#														
ENBSCH	905#	3320	3409	3487	3666	3802	3917	3985	4137	4334	4630	4889			
ENDCOM	1046#														
ERR	905#	1629	1637	1645	1653	1661	1669	1678	1686	1694	1702	1710	1718	1726	1734
	1742	1750	1758	1766	1774	1782	1790	1798	1806	1814	1822	1830	1838	1846	1854
	1862	1870	1878	1886	1894	1902	1910	1918	1926	1934	1942	1950	1958	1966	1974
	1982	1990	1998	2006	2014	2022	2030	2038	2046	2053	2060	2067	2074	2081	2088
	2095	2102	2109	2116	2124	2132	2140	2148	2156	2164	2172	2180	2188		
ERROR	940#	2474	2552	2573	2604	2658	2706	2747	2828	2885	2955	3034	3066	3098	3122
	3154	3177	3199	3231	3266	3294	3323	3340	3358	3386	3398	3412	3428	3448	3490
	3513	3538	3588	3603	3619	3635	3669	3675	3702	3741	3773	3805	3811	3817	3840
	3859	3891	3920	3926	3932	3958	3988	3995	4011	4027	4069	4081	4107	4140	4146
	4179	4204	4247	4270	4303	4337	4343	4378	4429	4476	4561	4585	4595	4633	4639
	4645	4651	4677	4690	4722	4739	4769	4801	4832	4855	4893	4899	4905	4911	4934
	4955	4962	5019	5030	5044	5060	5079	5103							
ESCAPE	1046#														
GETAS	905#														
GETBA	905#														
GETBAE	905#														
GETCS1	905#	2637	2690	2794	2856	3288									
GETCS2	905#	2543	2564												
GETDA	905#	2640	2796	2858	2907	3052	3084	3168	3193						
GETDB	905#														
GETDC	905#	2642	2802	2862	2911	3027	3114	3144	3219						
GETDS	905#	5451	5484	5571											
GETDT	905#	3252													
GETEC1	905#														
GETEC2	905#														
GETER1	905#	2692	2735	2798	2864	2913	3025	3112	3140	3191	3333	3353	3382	3394	3424
	3443	3504	3531	4578	5035										
GETER2	905#	2694	2737	2804	2866	2915	3056	3088	3170	3221					
GETHR	905#														
GETLA	905#														
GETMR1	905#	2739	3581	3596	3613	3630	3689	3715	3734	3765	3835	3854	3884	3949	4035
	4048	4061	4088	4160	4187	4225	4254	4277	4361	4387	4413	4439	4461	4543	4569
	4668	4707	4730	4752	4776	4822	4845	4921	4942	5008	5069	5093	5526	5544	5609
	5653	5720	5731	5743	5752	5774	5793								
GETMR2	905#	4004	4018	4196	4372	4422	4681	5022							
GETOF	905#	2644	2800	2860	2909	3054	3086	3142	3217						
GETPRI	1046#	5388													
GETSN	905#														
GETSWR	905#	1046#	2262#	2478											
GETWC	905#														
GETX	905#														
MSG	905#	2524	2611	2663	2711	2753	2966	3238	3272	3299	3363	3454	3555	3643	3781
	3896	3964	4111	4307	4599	4866									
MULT	1046#														
NEWTST	1046#	2522	2609	2661	2709	2751	2964	3236	3270	3297	3361	3452	3553	3641	3779
	3894	3962	4109	4305	4597	4864									
POP	1046#	5374	5376	5870	5954	6562	6642	6643	6706	6707					
PUSH	1046#	5336	5337	5850	5913	6541	6623	6629	6667	6669	6690				

```
TAGS      905#  1526
TRMTRP   6594#
TSTSET    905#  2529  2615  2668  2716  2757  2971  3243  3277  3303  3367  3458  3559  3647  3785
          3900  3968  4115  4311  4603  4870
TYPBIN   1046#
TYPDEC   1046#  5143  5150
TYPNAM    905#  1046#  2255
TYPNUM   1046#
TYPOCS   1046#  2376  2397  5190  5198  5207  5213
TYPOCT   1046#  2362  6380
TYPTXT   1046#  5139  5146
$$CMRE   1415#
$$CMTM   1415#  1452  1453  1454  1455  1456
$$ECA    1046#
$$NEWT   1046#  2522  2609  2661  2709  2751  2964  3236  3270  3297  3361  3452  3553  3641  3779
          3894  3962  4109  4305  4597  4864
$$SET    6594#  6603  6604  6605  6606  6607  6609  6611  6612  6613  6614  6615  6616
$$SETM   2246#
$$SKIP   1046#
.EQUAT    905#   936
.GETPR    905#
.HEADE    905#   906
.SETUP    905#  1362
.SWRHI    905#   917
.SWRLO    905#   928#
.$ACT1    905#  1374
.$APTB    905#  1464#
.$APTH    905#  1392
.$APTY    905#  6660
.$CATC    905#  1363
.$CMTA    905#  1415
.$DIV     905#
.$EOP     905#  5118
.$ERRO    905#  6219
.$ERRT    905#
.$MULT    905#
.$POWE    905#  6617
.$RAND    905#
.$RDDE    905#
.$RDOC    905#  6529
.$READ    905#  6273
.$SAVE    905#  5832
.$SCOP    905#  6124
.$SIZE    905#
.$TRAP    905#  6567
.$TYPB    905#  5877
.$TYPD    905#  5901
.$TYPE    905#  6045
.$TYPO    905#  5968

. ABS.  056126     000

 ERRORS DETECTED:  0
```

```
DSK:CZRMKB.BIN,DSK:CZRMKB.SEQ/SOL/NL:TOC:CND:MC:MD/LI:ME/DOC/CRF DSKM:CZRMKB.P11
RUN-TIME: 35 24 3 SECONDS
RUN-TIME RATIO: 144/64=2.?
CORE USED:  32K  (63 PAGES)

DOCUMENT PAGES:  187
```