

RXV21/RX02

FORMATTER PROGRAM
CZR XEA0

AH-E623A-MC
COPYRIGHT © 1978
FICHE 1 OF 1

DEC 1978
digital
MADE IN USA

This microfiche card contains a grid of frames. The frames are arranged in approximately 14 rows and 8 columns. The content of the frames includes:

- Textual data, possibly code or configuration parameters.
- Tables with multiple columns and rows of data.
- Diagrams, including what appears to be a signal waveform or a graph in one of the frames.

The frames are separated by thin white lines, and the overall card has a dark background.

5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601

; <PRUCHA>RX2FMT.P11.7, 22-AUG-78 13:28:06, EDIT BY SOUSA
.REM &

IDENTIFICATION

PRODUCT CODE: AC-E662A-MC
PRODUCT NAME: CZRXEAO RX02 FMTR PROG
DATE CREATED: 25 JUL 78
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: L.S. PRUCHA

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1978, BY DIGITAL EQUIPMENT CORPORATION

5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635

TABLE OF CONTENTS

| | |
|-----|-------------------------------------|
| 1.0 | ABSTRACT |
| 2.0 | REQUIREMENTS |
| 2.1 | HARDWARE |
| 2.2 | STORAGE |
| 3.0 | LOADING PROCEDURE |
| 4.0 | STARTING PROCEDURE |
| 5.0 | CONSOLE OR SOFTWARE SWITCH SETTINGS |
| 6.0 | OPERATION |
| 6.1 | CONDITIONS OF OPERATION |
| 7.0 | PROGRAM DESCRIPTION |
| 7.1 | FLOW |
| 7.2 | TEST DESCRIPTIONS |
| 8.0 | DEVICE REGISTERS |
| 9.0 | LISTING INDEX |
| 9.1 | LISTING |

5637 1.0 ABSTRACT
5638 -----
5639
5640 THIS PROGRAM IS INTENDED TO FORMAT A FLOPPY DISK TO EITHER SINGLE
5641 OR DOUBLE DENSITY ON A DRIVE CAPABLE OF SETTING DENSITY ON A FLOPPY
5642 DISK. IT WILL RUN UNDER APT, BUT WILL RUN IN A USER MODE WITH
5643 SEVERAL SWITCHABLE OPTIONS TO ENABLE A TESTING AND DEBUGGING.
5644
5645
5646 2.0 REQUIREMENTS
5647 -----
5648
5649
5650 2.1 HARDWARE
5651 -----
5652
5653 1. ANY PDP-11 PROCESSOR
5654 2. RX02,XX SUBSYSTEM
5655
5656
5657 2.2 STORAGE
5658 -----
5659
5660 THIS PROGRAM REQUIRES AT LEAST 4K WORDS OF CORE. IF LOADING
5661 VIA XXDP MORE STORAGE WILL BE NEEDED FOR THE XXDP MONITOR.
5662
5663
5664 3.0 LOADING PROCEDURE
5665 -----
5666
5667 USE OF STANDARD BINARY LOADING PROCEDURE OR DOWN LINE LOAD VIA
5668 APT.
5669
5670
5671 4.0 STARTING PROCEDURE
5672 -----
5673
5674 1. USER MODE - START AT LOCATION 200(8) TO INITIALIZE
5675 ----- REGISTERS.
5676 - START AT LOCATION 210(8) TO RESTART TEST.
5677 - START AT LOCATION 220(8) TO ENTER DEBUG MODE.
5678
5679 2. APT MODE - ALWAYS START AT LOCATION 200(8).
5680 -----

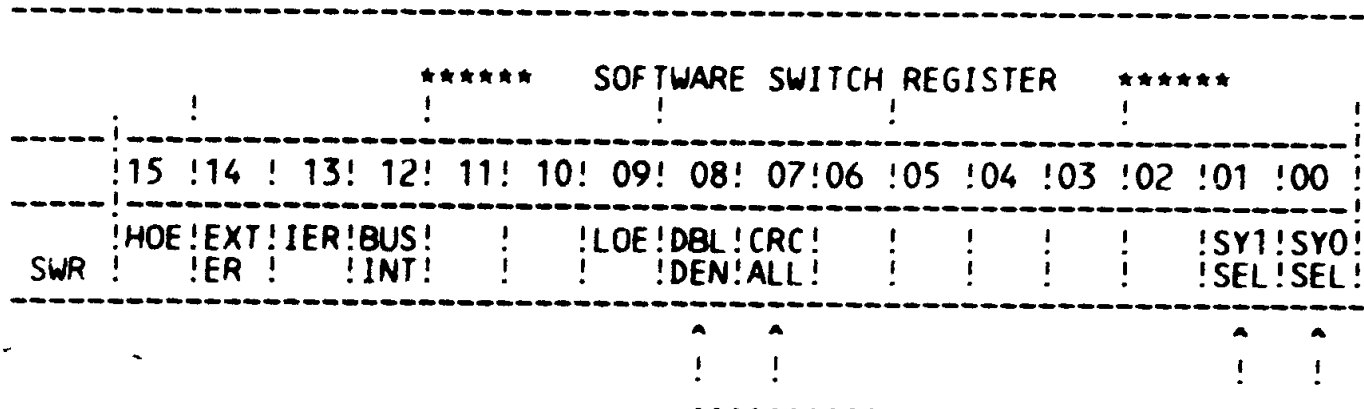
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729

5.0 SOFTWARE SWITCH SETTINGS

1. USER MODE - THE SWITCH REGISTER IS SETUP FOR THE USER BY THE PROGRAM VIA USER DIALOG.

- SW15: 1 = HALT ON ERROR
0 = CONTINUE
- SW14: 1 = EXTENDED ERROR REPORTS
0 = NORMAL ERROR REPORTS
- SW13: 1 = INHIBIT ERROR REPORTS
0 = PRINT ERRORS
- SW12: 1 = BUS INITIALIZE ON ERROR IF LOOPING
0 = NO BUS INITIALIZE ON ERROR
- SW09: 1 = LOOP ON ERROR
0 = CONTINUE

THESE SWITCH SETTINGS ARE AVAILABLE, BUT ARE INTENDED FOR HARDWARE DEBUG ONLY.



THESE BITS ARE SET BY PROGRAM INTERFACE TO USER - THEY ARE SHOWN HERE TO AID IN PROGRAM MAINTENANCE.

2. APT MODE - THE APT SWITCH REGISTER '\$SWREG' MUST BE SET UP AS FOLLOWS: SY0SEL=1 (BIT0) AND DBLDEN=1 (BIT8) FOR SETTING TO DOUBLE DENSITY. IF CRC CHECK ON ALL TRACKS IS DESIRED THEN SET BIT CRCALL=1 (BIT7) SEE SWITCH REGISTER ABOVE.

5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770

6.0 OPERATION

THE PROGRAM OPERATION IN 'USER MODE' IS SIMPLE. AFTER INITIAL START THE PROGRAM WILL ASK THE OPERATOR IF HELP IS WANTED AND WILL TYPE OUT A SHORT DESCRIPTION OF THE PROGRAM IF ANSWERED YES. THE PROGRAM WILL ALSO ASK IF A FULL DISKETTE SCAN IS WANTED TO VERIFY CRC. IF ADDRESS MODIFICATION IS SELECTED THEN THE PROGRAM WILL THEN ASK THE OPERATOR TO ENTER THE BUS ADDRESS AND VECTOR ADDRESS.

6.1 CONDITIONS OF OPERATION

THE PROGRAM EXPECTS TO RUN ON AN RX02 INTERFACE AND RX02 DRIVE SUBSYSTEM THAT HAVE PASSED ALL DIAGNOSTIC TESTS. ERRORS ARE REPORTED, BUT PRINTOUTS ARE NOT EXTENSIVE ENOUGH TO DIAGNOSE HARDWARE FAILURES.

7.0 PROGRAM DESCRIPTION

THE PROGRAM IS ORGANIZED AS FOLLOWS:

START
DETERMINE IF LSI PROCESSOR
IF NOT APT MODE, CALL OPERATOR INTERFACE
SET UP REGISTERS
CALL SYSTEM SCHEDULER
IF NOT APT MODE-CALL OUTPUT SYSTEM DONE (GET FURTHER OPERATOR INSTRUCTIONS)
PRINT END OF PASS
INCREMENT PASS COUNTER
CHECK IF ACT MODE
BUS RESET
IF IDLE MODE SET THEN IDLE IN LOOP
JUMP BACK TO 'CALL SYSTEM SCHEDULER'

```
5772          7.1  FLOW
5773          ----
5774          BEGINROUTINE (MOD PROGRAM START + CONTROL)
5775          : INITIALIZE STACK POINTER
5776          : IF RO IS FALSE
5777          :   THEN
5778          :     PRINT PROGRAM IDENTIFICATION MSG
5779          :     IF APT MODE IS TRUE
5780          :       THEN
5781          :         GET APT SWITCH REGISTER
5782          :         IF APT_PROCESSOR_Q-BUS TRUE
5783          :           THEN
5784          :             SET LSI PROC PRI
5785          :           ELSE
5786          :             SET PDP11 PROC PRI
5787          :           ENDIF
5788          :         GET APT VECTOR
5789          :         SET SYSTEM VECTOR
5790          :         GET APT BASE ADDRESS OF UNIT UNDER TEST
5791          :         CALL ADDRESSING TEST
5792          :       ELSE
5793          :         SETUP BUS TRAP FOR LSI PROC TEST
5794          :         SET PROCESSOR STATUS WORD
5795          :         IF LSI_PROCESSOR_STATUS_WORD_TRAP TRUE
5796          :           THEN
5797          :             SET LSI PROCESSOR WORD STATUS
5798          :             SET LSI PROCESSOR FLAG
5799          :           ENDIF
5800          :         CALL OPERATOR INTERFACE (MOD 1.0)
5801          :         SET VECTOR INTERRUPT ADDRESS
5802          :       ENDIF
5803          :     SETUP DENSITY PER SWITCH REGISTER
5804          :     CLEAR TABLES + FLAGS
5805          :     CALL MOD-TEST ADDRESS
5806          :   ENDIF
5807          : CLEAR PASS COUNTER
5808          : BGNDO
5809          :   CALL SYSTEM SCHEDULER (MOD 2.0)
5810          :   IF APT MODE FALSE
5811          :     THEN
5812          :       CALL OUTPUT DONE (MOD 3.0)
5813          :     ENDIF
5814          :   SET UP/PRINT END OF PASS + PASS COUNT
5815          :   INCREMENT PASS COUNTER
5816          :   IF ACT MODE TRUE
5817          :     THEN
5818          :       RESET
5819          :       CALL ACT HOOKS
5820          :     ENDIF
5821          :   IF SYSTEM_FLAG-IDLE
5822          :     THEN
5823          :       BEGINDO
5824          :     ENDDO
5825          :   ENDIF
5826          : ENDDO
5827          ENDROUTINE
```

5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844

7.2 PROGRAM FUNCTIONAL DESCRIPTIONS

THE PROGRAM WILL ACCOMPLISH THE FOLLOWING FUNCTIONAL ELEMENTS:

1. PROGRAM INITIALIZE SELECTED SYSTEMS + CHECK STATUS
2. READ MAINTENANCE STATUS ON SELECTED DRIVES + CHECK STATUS
3. SET DENSITY ON ALL SELECTED DRIVES + CHECK STATUS
4. CHECK CRC ON TRACKS #76 AND #0 OR ALL TRACKS, IF SELECTED BY USER. REPORT STATUS, IF ANY ERRORS.
5. REPORT WHEN DONE.

5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883

8.0 DEVICE REGISTERS

| CODE | FUNCTION |
|------|--|
| 0 | = FILL BUFFER |
| 1 | = EMPTY BUFFER |
| 2 | = WRITE SECTOR |
| 3 | = READ SECTOR |
| 4 | = SET DENSITY (RX02) ** BE CAREFUL TAKES 15 SECONDS ** |
| 5 | = READ STATUS ** MAINTENANCE MODE ** |
| 6 | = WRITE SECTOR WITH DELISTED DATA |
| 7 | = READ ERROR CODE |

***** RX02 REGISTER BITS *****

| | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|-------|--------------|-----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| RXCS: | ERR | INT | XM | XM | RX2 | | SID | DEN | TR | IE | DON | DRV | FUN | FUN | FUN | GO |
| WC: | WORD COUNT | | | | | | | | | | | | | | | |
| BA: | BASE ADDRESS | | | | | | | | | | | | | | | |
| RXES: | X | X | X | X | NXM | WC | SID | DRV | DRV | DEL | DSK | DEN | AC | INT | SID | CRC |
| | | | | | | OVF | #1 | #1 | RDY | DAT | DEN | ERR | LOW | DON | RDY | ERR |
| RXDB: | DATA BUFFER | | | | | | | | | | | | | | | |

5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940

9.0 LISTING INDEX

6042 BASIC DEFINITIONS
6069 ACT11 HOOKS
6128 APT MAILBOX-ETABLE
6130 APT PARAMETER BLOCK
6186 MODULE 0.0 - PROGRAM START AND CONTROL
6269 MODULE 1.0 - OPERATOR INTERFACE
6376 MODULE 1.1 - GET SYSTEM BUS ADDRESS + TEST
6432 ADDRESSING TEST
6452 ERROR 1 UNIT NOT RESPONDING TO ADDRESS
6459 MODULE 1.2 - GET SOFTWARE SWITCH REGISTER
6476 MODULE 1.3 - CHECK SYSTEM ADDRESS
6512 MODULE 2.0 - SYSTEM SCHEDULER
6618 MODULE 2.1 - INITIALIZE CHECK
6677 ERROR 2 NO DONE BIT AFTER INITIALIZE
6690 ERROR 3 ERROR BIT SET ON INITIALIZE
6705 MODULE 2.2 - SYSTEM DRIVER
6818 MODULE 2.2.1 - GET COMMAND
6911 MODULE 2.2.2 - TRACK + SECTOR UPDATE
7040 MODULE 2.2.3 - DRIVE DRIVER
7083 MODULE 2.2.3.1 - OUTPUT DRIVE COMMAND
7149 MODULE 2.2.3.1.1 - OUTPUT SINGLE WORD
7161 MODULE 2.2.5 - ERROR CHECK
7237 ERROR 4 READ MAINT STATUS COMMAND ERROR
7253 ERROR 5 SET DENSITY COMMAND ERROR
7270 ERROR 6 READ SECTOR COMMAND ERROR
7279 MODULE 2.2.5.1 - DRIVE DROP
7385 MODULE 2.2.5.2 - DRIVES AVAILABLE CHECK
7481 MODULE 2.2.5.3 - SET TEST BITS IN CSR + ESR
7524 MODULE 2.2.5.4 - LOOPING MODULE
7539 ERROR 7 LOOPING MOD-CSR ERR
7543 ERROR 10 LOOPING MOD-ESR ERR
7548 MODULE 2.3 - SYSTEM STATUS
7659 MODULE 2.3.1 - PRINT SYSTEM STATUS
7728 MODULE 2.3.2 - SYSTEM DROP
7758 MODULE 2.3.3 - SYSTEM TIMEOUT CHECK
7803 MODULE 2.4 - WATCH DOG SYSTEM#0
7826 MODULE 2.5 - WATCH DOG SYSTEM#1
7853 MODULE 3.0 - OUTPUT SYSTEM DONE
7893 MODULE U.PRIHI - SET PROCESSOR PRIORITY HIGH
7901 MODULE U.PRILO - SET PROCESSOR PRIORITY LOW
7909 MODULE U.PROPRI - SET PROCESSOR PRI
7922 MODULE U.SADR - SET SYSTEM BUS ADDRESS
7932 MODULE U.SUCO - SET SYSTEM UNDER CONTROL OFFSET
7940 MODULE U.DL - DELAY FOR 'TR' OR 'DONE'
7963 TIME OUT ERROR PRINT
7996 RXCS ERROR CHECK
8037 RXES ERROR CHECK
8078 TEST HEADER CHECK + PRINT
8098 MODULE U.PRYSYS - PRINT SYSTEM IDENTIFICATION
8106 MODULE U.PRTRV - PRINT DRIVE IDENTIFICATION
8124 MODULE U.PRTKSC - PRINT TRACK + SECTOR ERROR IDENT
8142 CHECK BITS SET + NOT SET
8255 ERROR SET SUBROUTINE

| | | |
|------|------|--|
| 5941 | 8264 | BUS INITIALIZE SUBROUTINE |
| 5942 | 8294 | RX02 INTERRUPT HANDLER #0 |
| 5943 | 8301 | RX02 INTERRUPT HANDLER #1 |
| 5944 | 8307 | TTY INTERRUPT HANDLER |
| 5945 | 8323 | BUS ADDRESS TRAP HANDLER |
| 5946 | 8346 | APT ERROR HANDLER |
| 5947 | 8358 | MODULE U.OPRANS - SETUP + GET OPERATOR ANSWERS |
| 5948 | 8371 | TTY ENTRY SUBROUTINE |
| 5949 | 8418 | TTY ENTRY ERROR SUBROUTINE |
| 5950 | 8427 | TTY ANSWER ENTRY SUBROUTINE |
| 5951 | 8458 | TTY READ SUBROUTINE |
| 5952 | 8474 | TTY ASCII OUTPUT SUBROUTINE |
| 5953 | 8497 | OCTAL OUTPUT SUBROUTINE |
| 5954 | 8560 | DATA CHARACTER OUTPUT SUBROUTINE |
| 5955 | 8587 | TTY OUTPUT |
| 5956 | 8601 | MESSAGE TABLE |
| 5957 | 8666 | TEST HEADERS |
| 5958 | 8677 | ERROR CODE STORAGE |
| 5959 | 8691 | PATCH AREA |
| 5960 | | |
| 5961 | | |
| 5962 | | |
| 5963 | | |
| 5964 | | |
| 5965 | | |
| 5966 | | |

5968
 5969
 5970
 5971
 5972
 5973
 5974
 5975
 5976
 5977
 5978
 5979
 5980
 5981
 5982
 5983
 5984
 5985
 5986
 5987
 5988
 5989
 5990
 5991
 5992
 5993
 5994
 5995
 5996
 5997
 5998
 5999
 6000
 6001
 6002
 6003
 6004
 6005
 6006
 6007
 6008
 6027
 6034

9.1 LISTING

```
&
.TITLE CZRXEAO RX02 FMTR PROG
:RX02-APT
:14-MARCH-78
:L. PRUCHA
.ENABLE ABS,AMA
.LIST ME
.NLIST BEX,CND,MC,MD
```

:***** PROGRAM EQUIVALENTS *****

| | | | | |
|--------|---------|---|--------|--|
| 000040 | DNBIT | = | 40 | :DONE BIT-----<CSR> |
| 000200 | TRBIT | = | 200 | :TR BIT-----<CSR> |
| 000400 | DENBIT | = | 400 | :DENSITY BIT-----<CSR> |
| 001000 | SIDE1 | = | 1000 | :SIDE #1 BIT-----<ESR> & <CSR> |
| 000200 | DRVRDY | = | 200 | :DRIVE READY BIT-<ESR> |
| 000010 | ACLOW | = | 10 | :AC LOW BIT-----<ESR> |
| 000004 | INITDN | = | 4 | :INITIALIZE DONE BIT-<ESR> |
| 000002 | SIDRDY | = | 2 | :SIDE READY BIT--<ESR> |
| 000400 | DRIVE1 | = | 400 | :DRIVE #1 BIT-----<ESR> |
| 000020 | DENERR | = | BIT4 | :DENSITY ERROR----<ESR> |
| 000040 | DRV DEN | = | BIT5 | :DRIVE DENSITY----<ESR> |
| 000001 | CRCERR | = | BIT0 | :CRC ERROR BIT----<ESR> |
| 100000 | ERRBIT | = | 100000 | :ERROR BIT-----<CSR> |
| 000020 | DRV1 | = | 20 | :DRIVE 1-----<CSR> |
| 040000 | RXINIT | = | 40000 | :RXINIT BIT-----<CSR> |
| 004000 | RX2BIT | = | 4000 | :RX02 BIT-----<CSR> |
| 000001 | APTENV | = | 1 | :APT ENVIRONMENT TEST WORD |
| 000040 | APTCSP | = | 40 | :APT CONSOLE SUPPRESSION WORD |
| 000200 | CRCALL | = | SW7 | :CRC ALL TRACKS & SECTORS FLAG |
| 000006 | RDSEC | = | 6 | :READ SECTOR COMMAND CODE (LEFT SHIFTED) |
| 000012 | RDMNST | = | 12 | :READ MAINT STATUS COMMAND CODE (LS) |
| 000010 | SETDEN | = | 10 | :SET DENSITY COMMAND CODE (LS) |
| 000400 | DBLDEN | = | BIT8 | :DOUBLE DENSITY BIT |
| 000001 | X=1 | | | |
| 000001 | N=1 | | | |


```

(1)      000304      SW02= 4
(1)      000002      SW01= 2
(1)      000001      SW00= 1
(1)      .EQUIV SW09,SW9
(1)      .EQUIV SW08,SW8
(1)      .EQUIV SW07,SW7
(1)      .EQUIV SW06,SW6
(1)      .EQUIV SW05,SW5
(1)      .EQUIV SW04,SW4
(1)      .EQUIV SW03,SW3
(1)      .EQUIV SW02,SW2
(1)      .EQUIV SW01,SW1
(1)      .EQUIV SW00,SW0

(1)      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)      100000      BIT15= 100000
(1)      040000      BIT14= 40000
(1)      020000      BIT13= 20000
(1)      010000      BIT12= 10000
(1)      004000      BIT11= 4000
(1)      002000      BIT10= 2000
(1)      001000      BIT09= 1000
(1)      000400      BIT08= 400
(1)      000200      BIT07= 200
(1)      000100      BIT06= 100
(1)      000040      BIT05= 40
(1)      000020      BIT04= 20
(1)      000010      BIT03= 10
(1)      000004      BIT02= 4
(1)      000002      BIT01= 2
(1)      000001      BIT00= 1
(1)      .EQUIV BIT09,BIT9
(1)      .EQUIV BIT08,BIT8
(1)      .EQUIV BIT07,BIT7
(1)      .EQUIV BIT06,BIT6
(1)      .EQUIV BIT05,BIT5
(1)      .EQUIV BIT04,BIT4
(1)      .EQUIV BIT03,BIT3
(1)      .EQUIV BIT02,BIT2
(1)      .EQUIV BIT01,BIT1
(1)      .EQUIV BIT00,BIT0

(1)      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)      000004      ERRVEC= 4      ;;TIME OUT AND OTHER ERRORS
(1)      000010      RESVEC= 10     ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)      000014      TBITVEC=14     ;;'T' BIT
(1)      000014      TRTVEC= 14     ;;TRACE TRAP
(1)      000014      EPTVEC= 14     ;;BREAKPOINT TRAP (BPT)
(1)      000020      IOTVEC= 20     ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)      000024      PWRVEC= 24     ;;POWER FAIL
(1)      000030      EMTVEC= 30     ;;EMULATOR TRAP (EMT) **ERROR**
(1)      000034      TRAPVEC=34     ;;'TRAP' TRAP
(1)      000060      TKVEC= 60      ;;TTY KEYBOARD VECTOR
(1)      000064      TPVEC= 64      ;;TTY PRINTER VECTOR
(1)      000240      PIRQVEC-240    ;;PROGRAM INTERRUPT REQUEST VECTOR
  
```



```

6044
6045
6046          000000          .-0
6047 000000 000000          .WORD 0
6048 000002 000000          .WORD 0          ;UNASSIGNED TRAP
6049 000004 000000          .WORD 0
6050 000006 000000          .WORD 0          ;TIME OUT, BUS TRAP
6051 000010 000000          .WORD 0
6052 000012 000000          .WORD 0          ;RESERVED INSTRUCTION
6053 000014 000000          .WORD 0
6054 000016 000000          .WORD 0          ;TRACE TRAP
6055 000020 000000          .WORD 0
6056 000022 000000          .WORD 0
6057 000024 000000          .WORD 0
6058 000026 000000          .WORD 0
6059 000030 000000          .WORD 0
6060 000032 000000          .WORD 0
6061 000034 000000          .WORD 0
6062 000036 000000          .WORD 0
6063 000040 000000          .WORD 0
6064 000042 000000          .WORD 0
6065 000044 000000          .WORD 0
6066
6067
6068
6069          .SBTTL ACT11 HOOKS
(1)
(2)
(1)          ;*****
(1)          ;HOOKS REQUIRED BY ACT11
(1)          $SVPC=.          ;SAVE PC
(1)          .=46
(1) 000046 001422          $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD
(1)          .=52
(1) 000052 000000          .WORD 0          ;;2)SET LOC.52 TO ZERO
(1)          .-$VPC          ;; RESTORE PC
  
```

```
6071 ;***** TTY INTERRUPT VECTOR *****
6072
6073      000060      000060      .=60
6074      000060      012420      ITINT      ;TTY INTERRUPT HEADER ADDRESS
6075      000062      000000      0
6076
6077 ;***** START ADDRESS *****
6078
6079      000200      000200      .=200
6080      000200      005000      CLR      R0
6081      000202      000137      001000      JMP      START      ;PROGRAM START
6082
6083      000210      000210      .=210
6084      000210      012700      000001      MOV      #1,R0
6085      000214      000137      001000      JMP      START      ;SET NO HEADER FLAG
6086
6087      000220      000220      .=220
6088      000220      005237      000516      INC      DBGFLG      ;SET DEBUG FLAG
6089      000224      005000      CLR      R0      ;SET SO HEADER WILL PRINT
6090      000226      000137      001000      JMP      START
6091
6092 ;***** RX02 INTERRUPT VECTORS *****
6093
6094      000264      000264      .=264
6095      000264      012404      RXINT0      ;RX02 INTERRUPT HANDLER #0 ADDRESS
6096      000266      000340      340
6097
6098      000270      000270      .=270
6099      000270      012412      RXINT1      ;RX02 INTERRUPT HANDLER #1 ADDRESS
6100      000272      000340      340
6101
6102      000300      000300      .=300
6103
6104 ;***** CONSTANTS *****
6105
6106      000300      177560      TKS:      177560      ;TTY READER STATUS
6107      000302      177562      TKB:      177562      ;TTY READ BUFFER
6108      000304      177564      TPS:      177564      ;TTY PUNCH STATUS
6109      000306      177566      TPB:      177566      ;TTY PUNCH BUFFER
6110
6111 ;***** INITIAL CONSTANTS *****
6112
6113
6114      000310      000264      VECT0:    264      ; INTERRUPT VECTOR SYS #0
6115      000312      000270      VECT1:    270      ; INTERRUPT VECTOR SYS #1
6116      000314      177170      REGS0:    177170     ; STARTING REGISTER ADDRESS SYS #0
6117      000316      177200      REGS1:    177200     ; STARTING REGISTER ADDRESS SYS #1
6118      000320      000004      BTRP:     4          ; BUS TRAP ADDRESS
6119      000322      000006      BTRP2:    6          ; BUS TRAP PRIORITY LEVEL
6120      000324      000401      SWREG:    401       ; SOFTWARE SWITCH REGISTER - PRESET-> DEN-DBL,SYSO
6121
```

```

6123
6124      000400
6125      000401      .=400
6126      000264      ASWREG=401
6127      177170      AVECT1=264      :APT VECTOR 264
6128      .SBTTL APT MAILBOX-ETABLE      ABASE=177170      :APT UNIBUS ADDRESS
(1)
(2)      ::*****
(1)      .EVEN
(1) 000400      $MAIL:      ::APT MAILBOX
(1) 000400 000000      $MSGTY: .WORD   AMSGTY      ::MESSAGE TYPE CODE
(1) 000402 000000      $FATAL: .WORD   AFATAL      ::FATAL ERROR NUMBER
(1) 000404 000000      $TESTN: .WORD   ATESTN     ::TEST NUMBER
(1) 000406 000000      $PASS:  .WORD   APASS      ::PASS COUNT
(1) 000410 000000      $DEVCT: .WORD   ADEVCT     ::DEVICE COUNT
(1) 000412 000000      $UNIT:  .WORD   AUNIT      ::I/O UNIT NUMBER
(1) 000414 000000      $MSGAD: .WORD   AMSGAD     ::MESSAGE ADDRESS
(1) 000416 000000      $MSGLG: .WORD   AMSGLG     ::MESSAGE LENGTH
(1) 000420      $ETABLE:      ::APT ENVIRONMENT TABLE
(1) 000420      000      $ENV:  .BYTE   AENV        ::ENVIRONMENT BYTE
(1) 000421      000      $ENVM: .BYTE   AENVM       ::ENVIRONMENT MODE BITS
(1) 000422 000401      $SWREG: .WORD   ASWREG     ::APT SWITCH REGISTER
(1) 000424 000000      $USWR:  .WORD   AUSWR      ::USER SWITCHES
(1) 000426 000000      $CPUOP: .WORD   ACPUOP     ::CPU TYPE,OPTIONS
(1)      :*      BITS 15-11=CPU TYPE
(1)      :*      11/04=01,11/05=02,11/20=03,11/40 04,11/45 05
(1)      :*      11/70=06,PDQ=07,Q=10
(1)      :*      BIT 10=REAL TIME CLOCK
(1)      :*      BIT 9=FLOATING POINT PROCESSOR
(1)      :*      BIT 8=MEMORY MANAGEMENT
(1) 000430      000      $MAMS1: .BYTE   AMAMS1     ::HIGH ADDRESS,M.S. BYTE
(1) 000431      000      $MTYP1: .BYTE   AMTYP1     ::MEM. TYPE,BLK#1
(1)      :*      MEM.TYPE BYTE -- (HIGH BYTE)
(1)      :*      900 NSEC CORE=001
(1)      :*      300 NSEC BIPOLAR=002
(1)      :*      500 NSEC MOS=003
(1) 000432 000000      $MADR1: .WORD   AMADR1     ::HIGH ADDRESS,BLK#1
(1)      :*      MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
(1) 000434      000      $MAMS2: .BYTE   AMAMS2     ::HIGH ADDRESS,M.S. BYTE
(1) 000435      000      $MTYP2: .BYTE   AMTYP2     ::MEM. TYPE,BLK#2
(1) 000436 000000      $MADR2: .WORD   AMADR2     ::MEM.LAST ADDRESS,BLK#2
(1) 000440      000      $MAMS3: .BYTE   AMAMS3     ::HIGH ADDRESS,M.S.BYTE
(1) 000441      000      $MTYP3: .BYTE   AMTYP3     ::MEM. TYPE,BLK#3
(1) 000442 000000      $MADR3: .WORD   AMADR3     ::MEM.LAST ADDRESS,BLK#3
(1) 000444      000      $MAMS4: .BYTE   AMAMS4     ::HIGH ADDRESS,M.S.BYTE
(1) 000445      000      $MTYP4: .BYTE   AMTYP4     ::MEM. TYPE,BLK#4
(1) 000446 000000      $MADR4: .WORD   AMADR4     ::MEM.LAST ADDRESS,BLK#4
(1) 000450 000264      $VECT1: .WORD   AVECT1     ::INTERRUPT VECTOR#1,BUS PRIORITY#1
(1) 000452 000000      $VECT2: .WORD   AVECT2     ::INTERRUPT VECTOR#2BUS PRIORITY#2
(1) 000454 177170      $BASE:  .WORD   ABASE      ::BASE ADDRESS OF EQUIPMENT UNDER TEST
(1) 000456
(1)      .MEXIT
    
```

6130
(1)
(2)
(1)
(2)
(1) 000456
(1) 000024
(1) 000024 000200
(1) 000044
(1) 000044 000456
(1) 000456
(2)
(1)
(1)
(1)
(1) 000456
(1) 000456 000000
(1) 000460 000400
(1) 000462 000120
(1) 000464 000120
(1) 000466 000120
(1) 000470 000027

.SBTTL APT PARAMETER BLOCK

```
*****  
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT  
*****  
.$X=. ;;SAVE CURRENT LOCATION  
.=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM  
200 ;;FOR APT START UP  
.=44 ;;POINT TO APT INDIRECT ADDRESS PNTR.  
$APTHDR ;;POINT TO APT HEADER BLOCK  
.=.$X ;;RESET LOCATION COUNTER  
*****  
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC  
:INTERFACE SPEC.
```

```
$APTHD:  
$HIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.  
$MADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)  
$STMT: .WORD 120 ;;RUN TIME OF LONGEST TEST  
$PASTM: .WORD 120 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)  
$UNITM: .WORD 120 ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT  
          .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
6132 ;***** FLAGS AND COUNTERS *****
6133
6134 000472 177170 RXCSAD: 177170 ;RX02 CSR ADDRESS
6135 000474 177172 RXDBAD: 177172 ;RX02 DBR ADDRESS
6136 000476 000264 RXVECT: 264 ;RX02 VECTOR ADDRESS
6137 000500 000001 SECTR: 1 ;SECTOR ADDRESS
6138 000502 000000 TRACK: 0 ;TRACK ADDRESS
6139 000504 000324 SWR: #SWREG ;SWITCH REG ADDRESS
6140 000506 000000 DENS: 0 ;DENSITY WORD
6141 000510 000000 REGS: 0 ;REGISTER ADDRESS BUFFER
6142 000512 000000 VECT: 0 ;VECTOR ADDRESS BUFFER
6143 000514 000000 INITPG: 0 ;PROGRAM INITIALIZE FLAG
6144 000516 000000 DBGFLG: 0 ;DEBUG FLAG
6145 000520 000000 LSIFLG: 0 ;LSI OR SWITCHLESS PROCESSOR FLAG
6146 000522 000000 PCNTR: 0 ;PASS COUNTER
6147 000524 000000 TOB: 0 ;TYPE OUTPUT BUFFER
6148 000526 000000 TIB: 0 ;TYPE INPUT BUFFER
6149 000530 000000 ANSWER: 0 ;ANSWER TO QUESTION BUFFER
6150 000532 000000 TTWAIT: 0 ;TTY WAIT FLAG
6151 000534 000000 CSR: 0 ;RXCS TEMP STORE
6152 000536 000000 ESR: 0 ;RXES TEMP STORE
6153 000540 000000 WDOT: 0 ;WORD FOR OUTPUT
6154 000542 000000 CMD: 0 ;RX COMMAND
6155 000544 000000 CSRMSK: 0 ;RXCS MASK WORD
6156 000546 000000 CSRCMP: 0 ;RXCS COMPARE WORD
6157 000550 000000 ESRMSK: 0 ;RXES MASK WORD
6158 000552 000000 ESRCMP: 0 ;RXES COMPARE WORD
6159 000554 000000 TSTERR: 0 ;TEST ERROR FLAG
6160 000556 000000 SOWMLT: 0 ;WATCHDOG MULTIPLIER SYSTEM#0
6161 000560 000000 SIWMLT: 0 ;WATCHDOG MULTIPLIER SYSTEM#1
6162 000562 000000 TEMP1: 0
6163 000564 000000 TEMP2: 0
6164 000566 000000 TEMP3: 0
6165 000570 000000 BTRPFL: 0 ;BUS TRAP FLAG
6166 000572 000000 SYMSG: 0 ;SYSTEM MESSAGE BUFFER
6167 000574 000000 SYSFLG: 0 ;SYSTEM FLAGS
6168 000576 000000 STFLG: 0 ; - START PROGRAM FLAG
6169 000600 000000 TMSGAD: 0 ;MESSAGE ADDRESS-HEADER
6170 000602 000000 HDRFLG: 0 ;HEADER FLAG
6171 000604 000000 ERRTP: 0 ;ERROR TYPE
6172 000606 000000 SYSERR: 0 ;SYSTEM ERROR
6173 000610 000000 SUC: 0 ;SYSTEM UNDER CONTROL
6174 000612 000000 SOAV: 0 ;SYSTEM#0 AVAILABLE
6175 000614 000000 SIAV: 0 ;SYSTEM#1 AVAILABLE
6176 000616 000000 SORDY: 0 ;SYSTEM#0 READY
6177 000620 000000 SIRDY: 0 ;SYSTEM#1 READY
6178 000622 000000 SOCMD: 0 ;SYSTEM#0 COMMAND WORD
6179 000624 000000 SICMD: 0 ;SYSTEM#1 COMMAND WORD
6180 000626 000000 SODN: 0 ;SYSTEM#0 DONE FLAG
6181 000630 000000 SIDN: 0 ;SYSTEM#1 DONE FLAG
6182 000632 000000 TTITFG: 0 ;TTY INTERRUPT FLAG
6183 000634 000000 ENDFTB: 0 ;*** DO NOT REMOVE THIS LABEL ***
```



```

6241 001324 005237 000514      INC      INITPG      ;SET PROGRAM INITIALIZE FLAG
6242 001330 004737 002514      CALL     SYSCHD      ;GO FORMAT SCHEDULED SYSTEMS      (MOD 2.0)
6243 001334 132737 000001 000420 IG00:  BITB     #APTENV,$ENV ;IF APT MODE
6244 001342 001002                BNE     EG00         ;FALSE, THEN
6245 001344 004737 007676      CALL     OTSYDN      ;CALL OUTPUT SYSTEM DONE          (MOD 3.0)
6246 001350 012704 014164      EG00:  MOV     #MSG6,R4  ;SET END OF PASS MSG
6247 001354 004737 013232      JSR     PC,TTOUT     ;PRINT END OF PASS
6248 001360 005237 000532      INC     TTWAIT      ;SET TTY WAIT FLAG
6249 001364 013703 000522      MOV     PCNTR,R3
6250 001370 004737 013346      JSR     PC,OCTP      ;PRINT PASS NUMBER
6251 001374 005237 000522      INC     PCNTR        ;BUMP PASS COUNTER
6252 001400 013737 000522 000406      MOV     PCNTR,$PASS ;BUMP APT PASS COUNTER
6253 001406 013700 000042      IH00:  MOV     @#42,R0  ;IF ACT
6254 001412 001407                BEQ     II00         ;MODE, THEN
6255 001414 005037 000042      CLR     @#42        ;ABORT THIS PROGRAM
6256 001420 000005                RESET
6257 001422 004710                $ENDAD: JSR     PC,(R0) ;ACT HOOKS
6258 001424 000240                NOP
6259 001426 000240                NOP
6260 001430 000240                NOP
6261 001432 032737 000100 000574      II00:  BIT     #100,SYSFLG ;IF IDLE MODE      THESE NOP'S *
6262 001440 001401                BEQ     EI00         ;SET, THEN
6263 001442 000777                BR     1$           ;IDLE LOOP
6264 001444 000005                EI00:  RESET
6265 001446 005037 000532      CLR     TTWAIT      ;CLEAR TTY WAIT FLAG
6266 001452 000137 001306      END00: JMP     BE00         ;DO NEXT PASS
6267

```

```
6269 .SBTTL MODULE 1.0 - OPERATOR INTERFACE
6270
6271 ;BEGINROUTINE (MODULE 1.0 - GET USER INPUT)
6272 : SET SOFTWARE SWITCH REG ADDRESS
6273 : SETUP & PRINT 'HELP?'
6274 : CALL GET ANSWER
6275 : IF ANSWER=YES [A]
6276 : : THEN
6277 : : SETUP & PRINT HELP FILE
6278 : : ENDF
6279 : : SETUP & PRINT 'SET DISKETTE TO SINGLE DENSITY? (Y OR N)''
6280 : : CALL GET ANSWER
6281 : : IF ANSWER=YES [B]
6282 : : : THEN
6283 : : : SET DENS & SOFT SWITCH REG DENSITY BIT=SINGLE DENSITY
6284 : : : ELSE
6285 : : : SET DENS & SOFT SWITCH REG DENSITY BIT=DOUBLE DENSITY
6286 : : : ENDF
6287 : : SETUP & PRINT 'VERIFY DISKETTE CRC (ALL TRACKS)? (Y OR N)''
6288 : : CALL GET ANSWER
6289 : : IF ANSWER=YES [C]
6290 : : : THEN
6291 : : : SET SOFT SWITCH REG=CRC ALL BIT
6292 : : : ELSE
6293 : : : CLEAR SOFT SWITCH REG=CRC ALL BIT
6294 : : : ENDF
6295 : : SET <SUC>=0
6296 : : SETUP & PRINT 'FLOPPY SYS #0 ADDRESS CHANGE? (Y OR N)''
6297 : : CALL GET ANSWER
6298 : : IF ANSWER=YES [D]
6299 : : : THEN
6300 : : : SET YES ANSWER FLAG
6301 : : : ELSE
6302 : : : CLEAR YES ANSWER FLAG
6303 : : : ENDF
6304 : : CALL GET SYSTEM BUS ADDRESS
6305 : : CALL CHECK SYSTEM BUS ADDRESS (SEE IF BOOT SYSTEM)
6306 : : SET SOFT SWITCH REG=SYS #0 AVAILABLE
6307 : : SETUP & PRINT 'IS SECOND FLOPPY SYSTEM AVAILABLE? (Y OR N)''
6308 : : CALL GET ANSWER
6309 : : IF ANSWER=YES [E]
6310 : : : THEN
6311 : : : SET <SUC>=1
6312 : : : SET YES ANSWER FLAG
6313 : : : CALL GET SYSTEM BUS ADDRESS
6314 : : : CALL CHECK SYSTEM BUS ADDRESS (SEE IF BOT MEDIA)
6315 : : : SET SOFT SWITCH REG=SYS #1 AVAILABLE
6316 : : : ENDF
6317 : : IF DEBUG_FLAG=SET [F]
6318 : : : THEN
6319 : : : CALL GET SOFT SWITCH REG
6320 : : : ENDF
6321 ;ENDROUTINE
```

```

6323
6324 001456 012737 000324 000504 OPINT: MOV #SWREG,SWR ;SET SOFTSWITCH REG ADR
6325 001464 012704 015726 MOV #MSG43,R4 ;SETUP MSG 'HELP?'
6326 001470 004737 013232 CALL TTOUT ;PRINT
6327 001474 004737 012624 CALL GETANS ;CALL GET OPERATOR ANSWER
6328 001500 122737 000131 000530 IA10: CMPB #'Y,ANSWER ;IF ANSWER
6329 001506 001004 BNE EA10 ;IS YES, THEN
6330 001510 012704 015300 MOV #MSG39,R4 ;SETUP MSG --> HELP FILE
6331 001514 004737 013232 CALL TTOUT ;PRINT
6332 001520 012704 016045 EA10: MOV #MSG45,R4 ;SETUP MSG 'SET DISKETTE TO SINGLE DENSITY: (Y OR N)?'
6333 001524 004737 013232 CALL TTOUT ;PRINT
6334 001530 004737 012624 CALL GETANS ;CALL GET OPERATOR ANSWER
6335 001534 122737 000131 000530 IB10: CMPB #'Y,ANSWER ;IF ANSWER
6336 001542 001004 BNE LB10 ;IS YES, (SET SINGLE DENSITY), THEN
6337 001544 042777 000400 176732 BIC #DENBIT,@SWR ;CLEAR DENSITY BIT IN SOFT SWITCH REG
6338 001552 000403 BR EB10 ;BR TO END 'B'
6339 001554 052777 000400 176722 LB10: BIS #DENBIT,@SWR ;SET DENSITY BIT IN SOFT SWITCH REG
6340 001562 012704 016125 EB10: MOV #MSG46,R4 ;SETUP MSG 'VERIFY DISKETTE CRC (ALL TRACKS): (Y OR N)?'
6341 001566 004737 013232 CALL TTOUT ;PRINT
6342 001572 004737 012624 CALL GETANS ;CALL GET OPERATOR ANSWER
6343 001576 122737 000131 000530 IC10: CMPB #'Y,ANSWER ;IF ANSWER
6344 001604 001004 BNE LC10 ;IS YES, THEN
6345 001606 052777 000200 176670 BIS #CRCALL,@SWR ;SET CRC ALL FLAG IN SOFT SWITCH REG
6346 001614 000403 BR EC10 ;BR TO END 'C'
6347 001616 042777 000200 176660 LC10: BIC #CRCALL,@SWR ;CLEAR CRC ALL FLAG IN SOFT SWITCH REG
6348 001624 005037 000610 EC10: CLR SUC ;SET <SUC> = 0
6349 001630 012704 016206 MOV #MSG47,R4 ;SETUP MSG 'FLOPPY SYS 0 ADDRESS MODIFICATION (Y OR N)?'
6350 001634 004737 013232 CALL TTOUT ;PRINT
6351 001640 004737 012624 CALL GETANS ;CALL GET OPERATOR ANSWER
6352 001644 122737 000131 000530 ID10: CMPB #'Y,ANSWER ;IF ANSWER
6353 001652 001002 BNE LD10 ;IS YES, THEN
6354 001654 005202 INC R2 ;SET YES ANSWER FLAG
6355 001656 000401 BR ED10 ;BR TO END 'D'
6356 001660 005002 LD10: CLR R2 ;CLEAR YES ANSWER FLAG
6357 001662 004737 001766 ED10: CALL GTSYAD ;CALL GET SYSTEM BUS ADR & TEST
6358 001666 004737 002422 CALL CKSYAD ;CALL CHECK SYSTEM ADR - MEDIA PROTECT
6359 001672 052777 000001 176604 BIS #1,@SWR ;SET SYSTEM#0 AVAIL
6360 001700 012704 016271 MOV #MSG48,R4 ;SETUP MSG 'SECOND FLOPPY SUBSYSTEM (Y OR N)?'
6361 001704 004737 013232 CALL TTOUT ;PRINT
6362 001710 004737 012624 CALL GETANS ;CALL GET OPERATOR ANSWER
6363 001714 122737 000131 000530 IE10: CMPB #'Y,ANSWER ;IF ANSWER
6364 001722 001013 BNE IF10 ;IS YES THEN
6365 001724 012737 000001 000610 MOV #1,SUC ;SET <SUC> = 1
6366 001732 005202 INC R2 ;SET YES ANSWER FLAG
6367 001734 004737 001766 CALL GTSYAD ;CALL GET SYSTEM BUS ADR & TEST
6368 001740 004737 002422 CALL CKSYAD ;CALL CHECK SYSTEM ADR - MEDIA PROTECT
6369 001744 052777 000002 176532 BIS #2,@SWR ;SET SYSTEM#1 AVAIL
6370 001752 005737 000516 IF10: TST DBGFLG ;IF DEBUG FLAG
6371 001756 001402 BEQ X10 ;SET, THEN
6372 001760 004737 002346 CALL GTSWR ;GET SOFTWARE SWITCH REG
6373 001764 000207 X10: RETURN ;RETURN
6374

```

```

6376 .SBTTL MODULE 1.1 - GET SYSTEM BUS ADDRESS & TEST
6377
6378 :BEGIN ROUTINE (MODULE 1.1. GET SYSTEM BUS ADDRESS & TEST)
6379 : BEGIN DO
6380 :     GET <SUC> REGISTER & VECTOR ADDRESSES
6381 :     SETUP & PRINT BUS ADDRESS
6382 :     GET NEW BUS ADDRESS
6383 :     SET UP BUS ADDRESS
6384 :     SETUP & PRINT VECTOR ADDRESS
6385 :     GET NEW VECTOR
6386 :     SETUP VECTOR
6387 :     CLEAR BUS-TRAP-FLAG
6388 :     CALL ADDRESS TEST
6389 : DO UNTIL BUS TRAP FLAG=0
6390 :     SET <SUC> REGISTER & VECTOR ADDRESSES
6391 :ENDROUTINE
6392
6393 -----
    
```

| | | | | | | | |
|------|--------|--------|--------|--------|-------------|----------------|----------------------------|
| 6394 | 001766 | 000240 | | | GTSYAD: NOP | | |
| 6395 | 001770 | 000240 | | | BA11: NOP | | |
| 6396 | 001772 | 004737 | 010126 | | CALL | SSUCOF | : CALL SET <SUC> OFFSET |
| 6397 | 001776 | 016137 | 000314 | 000510 | MOV | REGS0(R1),REGS | : GET <SUC> REGISTER ADR |
| 6398 | 002004 | 016137 | 000310 | 000512 | MOV | VECT0(R1),VECT | : GET <SUC> VECTOR ADR |
| 6399 | 002012 | 005702 | | | IB11: TST | R2 | : IF YES FLAG |
| 6400 | 002014 | 001450 | | | BEQ | EB11 | : SET, THEN |
| 6401 | 002016 | 004737 | 011132 | | CALL | PRTSYS | : CALL PRINT SYSTEM IDENT |
| 6402 | 002022 | 012704 | 014072 | | MOV | #MSG4,R4 | : USER MODE |
| 6403 | 002026 | 004737 | 013232 | | JSR | PC,TTOUT | : REQUEST REGISTER ADDRESS |
| 6404 | 002032 | 013703 | 000510 | | MOV | REGS,R3 | |
| 6405 | 002036 | 004737 | 013346 | | JSR | PC,OCTP | : PRINT CURRENT ADDRESS |
| 6406 | 002042 | 012705 | 000510 | | MOV | #REGS,R5 | : SET ADDRESS SAVE LOC |
| 6407 | 002046 | 012701 | 000006 | | MOV | #6,R1 | : SET SIZE OF RESPONSE |
| 6408 | 002052 | 012702 | 177500 | | MOV | #177500,R2 | : SET UPPER LIMIT |
| 6409 | 002056 | 012703 | 177100 | | MOV | #177100,R3 | : SET LOWER LIMIT |
| 6410 | 002062 | 004737 | 012652 | | JSR | PC,TTR | : GO GET RESPONSE |
| 6411 | 002066 | 012704 | 014127 | | MOV | #MSG5,R4 | |
| 6412 | 002072 | 004737 | 013232 | | JSR | PC,TTOUT | : REQUEST VECTOR |
| 6413 | 002076 | 013703 | 000512 | | MOV | VECT,R3 | |
| 6414 | 002102 | 004737 | 013346 | | JSR | PC,OCTP | : PRINT CURRENT VECTOR |
| 6415 | 002106 | 012705 | 000512 | | MOV | #VECT,R5 | : SET ADDRESS SAVE LOC |
| 6416 | 002112 | 012701 | 000003 | | MOV | #3,R1 | : SET SIZE OF RESPONSE |
| 6417 | 002116 | 012702 | 000300 | | MOV | #300,R2 | : SET UPPER LIMIT |
| 6418 | 002122 | 012703 | 000250 | | MOV | #250,R3 | : SET LOWER LIMIT |
| 6419 | 002126 | 004737 | 012652 | | JSR | PC,TTR | : GO GET RESPONSE |
| 6420 | 002132 | 013700 | 000512 | | MOV | VECT,R0 | : GET VECTOR |
| 6421 | 002136 | 005037 | 000570 | | EB11: CLR | BTRPFL | : CLEAR BUS TRAP FLAG |
| 6422 | 002142 | 013737 | 000510 | 000472 | MOV | REGS,RXCSAD | : SET REGISTER ADR |
| 6423 | 002150 | 004737 | 002204 | | CALL | ADTST | : CALL ADDRESSING TEST |
| 6424 | 002154 | 005737 | 000570 | | UA11: TST | BTRPFL | : DO UNTIL BUS TRAP FLAG |
| 6425 | 002160 | 001303 | | | BNE | BA11 | : EQUALS 0 |
| 6426 | 002162 | 004737 | 010126 | | CALL | SSUCOF | : CALL SET <SUC> OFFSET |
| 6427 | 002166 | 013761 | 000510 | 000314 | MOV | REGS,REGS0(R1) | : SET <SUC> REGISTER ADR |
| 6428 | 002174 | 013761 | 000512 | 000310 | MOV | VECT,VECT0(R1) | : SET <SUC> VECTOR ADR |
| 6429 | 002202 | 000207 | | | X11: RETURN | | : RETURN |
| 6430 | | | | | | | |

```

6432          .SBTTL ADDRESSING TEST
6433          ;-----
6434
6435 002204 000005          ADTST: RESET          ;INITIALIZE BUS
6436 002206 012737 014367 000600          MOV #MSG12,TMSGAD ;SET TEST MSG HEADER
6437 002214 012777 012474 176076          MOV #TRAP,@BTRP   ;SET TRAP HANDLER ADR
6438 002222 012777 000340 176072          MOV #340,@BTRP2  ;
6439 002230 013701 000472          MOV RXCSAD,R1    ;GET CSR ADDRESS
6440 002234 005711          AD1: TST (R1)    ;REF CS REG
6441 002236 000240          NOP              ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR
6442 002240 062701 000002          ADD #2,R1        ;SET ADDRESS OF DB
6443 002244 005711          TST (R1)        ;REF DB REG
6444 002246 000240          NOP              ;
6445 002250 005737 000570          TST BTRPFL      ;IF INITIAL ADDRESS TEST
6446 002254 001427          BEQ AD2          ;DID NOT PASS, THEN
6447 002256 032777 020000 176220          BIT #BIT13,@SWR ;IF INHIBIT ERROR REPORT
6448 002264 001004          BNE 2$          ;NOT SET THEN
6449 002266 012704 014326          MOV #MSG11,R4   ;SET UNIT NOT RESPONDING
6450 002272 004737 013232          JSR PC,TTOUT    ;PRINT MSG
6451 002276 000240          2$: NOP          ;
6452 002300 004737 012262          JSR PC,BSINIT   ;CHECK SWR FOR BUS INITIALIZE
(1) 002304 032777 001000 176172          BIT #SW09,@SWR ;CHECK FOR LOOP ON ERROR
(1) 002312 001350          BNE AD1         ;GO TO LOOP ERROR
(1) 002314 004737 012240          JSR PC,ERRSET   ;
(2) 002320 000001          .WORD #1       ;FATAL ERR # 1 - UNIT NOT RESPONDING TO ADDRESS
(1) 002322 005777 176156          TST @SWR        ;CHECK FOR HALT ON ERROR
(1) 002326 100001          BPL 1$         ;HALT IF SET
(1) 002330 000000          HALT           ;<UNIT NOT RESPONDING TO ADDRESS>
(1) 002332 000240          1$: NOP         ;
6453 002334 005077 175760          AD2: CLR @BTRP   ;CLEAR BUS TRAP
6454 002340 005077 175756          CLR @BTRP2     ;HANDLER
6455 002344 000207          ADTSTX: RTS PC ;RETURN
6456
6457

```

```

6459      .SBTTL MODULE 1.2 - GET SOFTWARE SWITCH REGISTER
6460      -----
6461
6462      002346 012704 014002      GTSWR:  MOV      #MSG2,R4      ;REQUEST SWITCH REG MSG
6463      002352 004737 013232      JSR      PC,TTOUT           ;
6464      002356 013703 000324      MOV      SWREG,R3          ;SET CURRENT SWITCH REG
6465      002362 004737 013334      JSR      PC,OCTPE          ;PRINT CURRENT SWITCH REG
6466      002366 012737 000324      MOV      #SWREG,SWR        ;SET SOFTWARE SWITCH REG ADR
6467      002374 012705 000324      MOV      #SWREG,R5         ;SET ADDRESS OF SWITCH REG
6468      002400 012701 000006      MOV      #6,R1             ;SET SIZE OR RESPONSE
6469      002404 012702 177777      MOV      #177777,R2        ;SET UPPER LIMIT
6470      002410 012703 000000      MOV      #0,R3             ;SET LOWER LIMIT
6471      002414 004737 012652      JSR      PC,TTR            ;GO GET RESPONSE
6472      002420 000207      X12:    RETURN             ;RETURN
6473      -----
6474
6475      .SBTTL MODULE 1.3 - CHECK SYSTEM ADDRESS
6476      -----
6477
6478      ;BEGINROUTINE (MODULE 1.3 - CHECK SYSTEM ADDRESS)
6479      ; IF LOAD MEDIA PROTECT BYTE=RX02 [A]
6480      ; THEN
6481      ; IF <SUC> ADDRESS=RX02 STANDARD ADDRESS [B]
6482      ; THEN
6483      ; CALL SYSTEM IDENTIFICATION
6484      ; SETUP & PRINT 'REMOVE XXDP MEDIA FROM THIS SYSTEM'
6485      ; SETUP & PRINT '-TYPE 'CR' WHEN READY'
6486      ; BEGINDO [C]
6487      ; CALL TTY INPUT-GET TTY CHARACTER
6488      ; DOUNTIL CHARACTER='CR'
6489      ; CLEAR MEDIA PROTECT BYTE
6490      ; ENDDO
6491      ; ENDF
6492      ; ENDRROUTINE
6493      -----
6494
6495      002422 000240      CKSYAD: NOP
6496      002424 122737 000015 000041      IA13:  CMPB     #15,@#41      ;IF RX02
6497      002432 001026      BNE     EA13              ;WAS LOAD MEDIA, THEN
6498      002434 022761 177170 000314      IB13:  CMP     #177170,REGS0(R1);IF SYSTEM ADDRESS
6499      002442 001022      BNE     EA13              ;SET = RX02 STANDARD ADDRESS, THEN
6500      002444 004737 011132      CALL    PRTSYS           ;CALL PRINT SYSTEM IDENT
6501      002450 012704 015140      MOV     #MSG34,R4        ;SET MSG-> 'REMOVE XXDP MEDIA FROM BOOT SYSTEM'
6502      (1) 002454 004737 013232      CALL    TTOUT            ;PRINT MSG
6503      002460 012704 017065      MOV     #MSG58,R4        ;SET MSG-> '-TYPE 'CR' WHEN READY'
6504      (1) 002464 004737 013232      CALL    TTOUT            ;PRINT MSG
6505      002470 004737 013156      BC13:  CALL    TTIN          ;CALL TTY INPUT-GET CHARACTER
6506      002502 001372      CMPB   #15,TIB           ;DO UNTIL
6507      002504 105037 000041      BNE    BC13              ;CHARACTER = 'CR'
6508      002510 000240      CLRB  @#41              ;CLEAR RX02 BOOT MEDIA PROTECTION
6509      002512 000207      EA13:  NOP
6510      X13:  RETURN             ;RETURN
  
```


6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533
6534
6535
6536
6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
6563

```
.SBTTL MODULE 2.0 - SYSTEM SCHEDULER  
;BEGINROUTINE (MOD 2.0 SYSTEM SCHEDULER)  
: SET SYS#0 & SYS#1 READY BITS  
: SET SYS#0 SEL BIT  
: SET <SUC> = 0  
: SET SYSOAVL = ALL  
: CALL INITIALIZE CHECK (MOD 2.1)  
: IF SWR = SYS#1 SELECTED [A]  
:   THEN  
:     SET <SUC>=1  
:     SET SYSTAVL = ALL  
:     CALL INITIALIZE CHECK (MOD 2.1)  
:   ENDIF  
: IF NO SYSTEM AVAIL [B]  
:   THEN  
:     SETUP SYMSG 'NO SYSTEM AVAILABLE'  
:   ELSE  
:     BEGINDO [C]  
:     : IF SYSO SELECTED [D]  
:     :   THEN  
:     :     IF SYO DONE=NOTSET [H]  
:     :     :   THEN  
:     :     :     SET <SUC> = 0  
:     :     :     IF SYSO RDY SET [E]  
:     :     :     :   THEN  
:     :     :     :     CALL SYSTEM DRIVER (MOD 2.2)  
:     :     :     :     ELSE  
:     :     :     :     CALL WATCH DOG SYS#0 (MOD 2.4)  
:     :     :     :     ENDIF  
:     :     :   ENDIF  
:     :   ENDIF  
:     : IF SYST SELECTED [F]  
:     :   THEN  
:     :     IF SY1 DSONE=NOTSET [I]  
:     :     :   THEN  
:     :     :     SET <SUC> = 1  
:     :     :     IF SYST RDY SET [G]  
:     :     :     :   THEN  
:     :     :     :     CALL SYSTEM DRIVER (MOD 2.2)  
:     :     :     :     ELSE  
:     :     :     :     CALL WATCH DOG SYS#1 (MOD 2.5)  
:     :     :     :     ENDIF  
:     :     :   ENDIF  
:     :   ENDIF  
:     : CLEAR PROGRAM INITIALIZE FLAG  
:     : CALL UPDATE SYSTEM STATUS (MOD 2.3)  
:     DO UNTIL ALL SYSTEMS DONE OR NO SYSTEM AVAIL  
:     SETUP SYSTEM MSG  
:   ENDIF  
: PRINT PRINT SYSTEM MSG  
;ENDROUTINE
```

```

6565 ;-----
6566
6567 002514 005237 000616 SYSCHD: INC SORDY ;SET SYS0 RDY
6568 002520 005237 000620 INC S1RDY ;SET SYS1 RDY
6569 002524 052777 000001 175752 BIS #1,@SWR ;SET SYS1 SELECT IN SOFT SWITCH REG
6570 002532 005037 000610 CLR SUC ;SET SYS UNDER CONTROL=#0
6571 002536 012737 000017 000612 MOV #17,SOAV ;SET SYSTEM#0 AVAIL=ALL
6572 002544 004737 003002 CALL INITCK ;CALL INITIALIZE CHECK (MOD 2.1)
6573 002550 032777 000002 175726 IA20: BIT #2,@SWR ;IF SYS #1
6574 002556 001410 BEQ IB20 ;SELECTED, THEN
6575 002560 012737 000001 000610 MOV #1,SUC ;SET SYS UNDER CONTROL=#1
6576 002566 012737 000017 000614 MOV #17,S1AV ;SET SYSTEM#1 AVAIL=ALL
6577 002574 004737 003002 CALL INITCK ;CALL INITIALIZE CHECK (MOD 2.1)
6578 002600 005737 000612 IB20: TST SOAV ;IF SYSTEM#0
6579 002604 001007 BNE ID20 ;NOT AVAIL AND
6580 002606 005737 000614 TST S1AV ;IF SYSTEM#1
6581 002612 001004 BNE ID20 ;NOT AVAIL, THEN
6582 002614 012704 016357 MOV #MSG49,R4 ;SET MSG='NO SYS AVAIL TO THIS PROG'
6583 002620 000465 BR EB20 ;BR TO END 'B'
6584 002622 000240 BC20: NOP ;BEGIN DO UNTIL 'C'
6585 002624 032777 000001 175652 ID20: BIT #1,@SWR ;IF SYS #0
6586 002632 001421 BEQ IF20 ;SELECTED, THEN
6587 002634 005737 000626 IH20: TST SODN ;IF SYS0 DONE
6588 002640 001016 BNE IF20 ;NOT SET, THEN
6589 002642 005037 000610 CLR SUC ;SET SYSTEM UNDER CONTROL SYS #0
6590 002646 005737 000616 IE20: TST SORDY ;IF SYS#0 READY FLAG
6591 002652 001407 BEQ LE20 ;IS SET, THEN
6592 002654 005037 000616 CLR SORDY ;CLEAR SYS #0 READY
6593 002660 005237 007670 INC WATINO ;SET WATCHDOG INIT #0 FLAG
6594 002664 004737 003272 CALL SYSDVR ;CALL SYS DRIVER (MOD 2.2)
6595 002670 000402 BR IF20 ;BR TO IF 'F'
6596 002672 004737 007474 LE20: CALL WATCHO ;CALL WATCH DOG SYS#0 (MOD 2.4)
6597 002676 032777 000002 175600 IF20: BIT #2,@SWR ;IF SYS #1
6598 002704 001422 BEQ EF20 ;SELECTED, THEN
6599 002706 005737 000630 II20: TST S1DN ;IF SYS1 DONE
6600 002712 001017 BNE EF20 ;NOT SET, THEN
6601 002714 012737 000001 000610 MOV #1,SUC ;SET SYSTEM UNDER CONTROL SYS #1
6602 002722 005737 000620 IG20: TST S1RDY ;IF SYS #1 READY FLAG
6603 002726 001407 BEQ LG20 ;IS SET, THEN
6604 002730 005037 000620 CLR S1RDY ;CLEAR SYS #1 READY
6605 002734 005237 007672 INC WATIN1 ;SET WATCHDOG INIT #1 FLAG
6606 002740 004737 003272 CALL SYSDVR ;CALL SYS DRIVER (MOD 2.2)
6607 002744 000402 BR OF END 'G' ;BR OT END 'G'
6608 002746 004737 007572 LG20: CALL WATCH1 ;CALL WATCH DOG SYS #1 (MOD 2.5)
6609 002752 005037 000514 EF20: CLR INITPG ;CLEAR PROGRAM INITIALIZE FLAG
6610 002756 004737 006570 CALL SYSTAT ;CALL SYSTEM STATUS (MOD 2.3)
6611 002762 005737 000574 DC20: TST SYSFLG ;DO UNTIL SYSTEM FLAG
6612 002766 001715 BEQ BC20 ;SET (EITHER SYSTEM DONE OR NO SYS AVAIL)
6613 002770 013704 000572 MOV SYMSG,R4 ;SETUP SYSTEM MSG
6614 002774 004737 013232 EB20: CALL TTOUT ;PRINT SYSTEM MSG
6615 003000 000207 X20: RETURN ;RETURN
6616 ;-----

```

```
6618 .SBTTL MODULE 2.1 - INITIALIZE CHECK
6619
6620 ;BEGINROUTINE (MOD 2.1 - INITIALIZE CHECK
6621 : CLEAR INITIALIZE DROP FLAG
6622 : SET <SUC> ADDRESS & OFFSET
6623 : SET <SUC> COMMAND=INITIALIZE
6624 : INITIALIZE SYSTEM AT ADDRESS
6625 : SETUP DELAY FOR DONE
6626 : CALL DELAY ROUTINE
6627 : SET <SUC> OFFSET
6628 : IF <SUC> CSR = NO DONE BIT SET [A]
6629 : : THEN
6630 : : SET INITIALIZE DROP FLAG
6631 : : CALL PRINT SYSTEM IDENTIFICATION
6632 : : SET APT ERROR #
6633 : ELSE
6634 : : IF <SUC> CSR=ERR BIT SET [B]
6635 : : : THEN
6636 : : : SET INITIALIZE DROP FLAG
6637 : : : CALL PRINT SYSTEM IDENTIFICATION
6638 : : : SETUP & PRINT 'ERROR BIT SET AFTER INITIALIZE'
6639 : : : IF <SUC> ESR = AC LOW BIT SET [C]
6640 : : : : THEN
6641 : : : : SETUP & PRINT 'AC-LOW-IS FLOPPY SYSTEM POWERED UP?'
6642 : : : : ENDIF
6643 : : : IF <SUC> ESR = INITIALIZE DONE NOT SET [D]
6644 : : : : THEN
6645 : : : : SETUP & PRINT 'INITIALIZE NOT DONE - RUN DIAG UNLESS ERROR'
6646 : : : : ENDIF
6647 : : : SET APT ERROR #
6648 : : : ENDIF
6649 : : IF <SUC> CSR NOT=DOUBLE DENSITY TYPE SYSTEM [E]
6650 : : : THEN
6651 : : : SET INITIALIZE DROP FLAG
6652 : : : SETUP & PRINT 'THIS SYSTEM NOT CAPABLE OF DOUBLE DEN OPS'
6653 : : : ENDIF
6654 : : ENDIF
6655 : IF INITIALIZE_DROP FLAG SET [F]
6656 : : THEN
6657 : : CLEAR <SUC> AVAIL
6658 : : ENDIF
6659 :ENDROUTINE
```

```

6661 ;-----
6662 ;
6663 003002 005037 003270 INITCK: CLR INITDP ;CLEAR INITIALIZE DROP FLAG
6664 003006 004737 010076 CALL SSUCAD ;CALL SET <SUC> ADDRESS
6665 003012 012761 040000 000622 MOV #RXINIT,SOCMD(R1) ;SET <SUC> COMMAND = INITIALIZE
6666 003020 012777 040000 175444 MOV #RXINIT,@RXCSAD ;INITIALIZE SYSTEM AT <SUC> CSR ADDRESS
6667 003026 013737 000472 010230 MOV RXCSAD,CSRADR ;SET ADDRESS FOR DELAY TEST
6668 003034 012737 000040 010226 MOV #DNMBIT,RDYWD ;SET DONE BIT FOR DELAY TEST
6669 003042 012737 000015 010222 MOV #15,RYDX ;SET DELAY MULTIPLIER
6670 003050 004737 010136 CALL DELAY ;DELAY FOR DONE BIT
6671 003054 004737 010126 CALL SSUCOF ;CALL SET <SUC> OFFSET
6672 003060 032777 000040 175404 IA21: BIT #DNMBIT,@RXCSAD ;IF <SUC> CSR DONE BIT
6673 003066 001014 BNE IB21 ;NOT SET, THEN
6674 003070 005237 003270 INC INITDP ;SET INITIALIZE DROP FLAG
6675 003074 004737 011132 CALL PRSYS ;CALL PRINT SYSTEM IDENT
6676 003100 012704 016436 MOV #MSG52,R4 ;SET MSG-> 'NO DONE BIT AFTER INITIALIZE'
(1) 003104 004737 013232 CALL TTOUT ;PRINT MSG
6677 003110 004737 012240 JSR PC,ERRSET
(2) 003114 000002 .WORD #2 ;FATAL ERR # 2 - NO DONE BIT AFTER INITIALIZE
6678 003116 000453 BR EA21 ;BR TO MOD EXIT
6679 003120 032777 100000 175344 IB21: BIT #ERRBIT,@RXCSAD ;IF <SUC> CSR ERROR BIT
6680 003126 001433 BEQ IE21 ;SET, THEN
6681 003130 005237 003270 INC INITDP ;SET INITIALIZE DROP FLAG
6682 003134 004737 011132 CALL PRSYS ;CALL PRINT SYSTEM IDENT
6683 003140 012704 014776 MOV #MSG29,R4 ;SET MSG-> 'ERROR BIT SET AFTER INITIALIZE'
(1) 003144 004737 013232 CALL TTOUT ;PRINT MSG
6684 003150 032777 000010 175316 IC21: BIT #ACLOW,@RXDBAD ;IF <SUC> ESR AC LOW BIT
6685 003156 001404 BEQ ID21 ;IS SET THEN
6686 003160 012704 016530 MOV #MSG54,R4 ;SET MSG-> 'AC LOW ERROR-IS DISK SYSTEM POWERED UP?'
(1) 003164 004737 013232 CALL TTOUT ;PRINT MSG
6687 003170 032777 000004 175276 ID21: BIT #INITDN,@RXDBAD ;IF <SUC> ESR INITIALIZE DONE BIT
6688 003176 001004 BNE ED21 ;NOT SET, THEN
6689 003200 012704 015752 MOV #MSG44,R4 ;SET MSG-> 'INITIALIZE NOT DONE-RUN DIAG UNLESS ERR OBVI'
(1) 003204 004737 013232 CALL TTOUT ;PRINT MSG
6690 003210 ED21:
(1) 003210 004737 012240 JSR PC,ERRSET
(2) 003214 000003 .WORD #3 ;FATAL ERR # 3 - ERROR BIT SET ON INITIALIZE
6691 003216 032777 004000 175246 IE21: BIT #RX2BIT,@RXCSAD ;IF <SUC>-DOUBLE DENSITY SYSTEM TYPE
6692 003224 001010 BNE EA21 ;BIT NOT SET, THEN
6693 003226 005237 003270 INC INITDP ;SET INITIALIZE DROP FLAG
6694 003232 004737 011132 CALL PRSYS ;CALL PRINT SYSTEM IDENT
6695 003236 012704 014610 MOV #MSG23,R4 ;SET MSG-> 'THIS SYS NOT CAPABLE OF DOUBLE DEN OPS'
(1) 003242 004737 013232 CALL TTOUT ;PRINT MSG
6696 003246 012761 000017 000612 EA21: MOV #17,SOAV(R1) ;SET <SUC> SYSTEM AVAIL
6697 003254 005737 003270 IF21: TST INITDP ;IF INITIALIZE DROP FLAG
6698 003260 001402 BEQ X21 ;SET, THEN
6699 003262 005061 000612 CLR SOAV(R1) ;CLEAR <SUC> SYSTEM AVAILABLE
6700 003266 000207 X21: RETURN ;RETURN
6701 ;-----
6702 003270 000000 INITDP: 0 ;INITIALIZE DROP FLAG
6703 ;-----

```

```
6705 .SBTTL MODULE 2.2 - SYSTEM DRIVER
6706
6707 :BEGINROUTINE (MOD 2.2 SYSTEM DRIVER)
6708 :   CLEAR WATCHDOG INITIALIZE FLAG FOR <SUC>
6709 :   CALL ERROR CHECK ROUTINE (MOD 2.2.5)
6710 :   IF THIS_SYSTEM_AVAILABLE NOT = 0 [A]
6711 :   THEN
6712 :   BEGINDO [C]
6713 :   :   SET <SUC> DRIVE_DONE=GET NEXT COMMAND FLAG
6714 :   :   CLEAR GET NEXT COMMAND FLAG
6715 :   :   CALL GET NEXT COMMAND (MOD 2.2.1)
6716 :   :   IF COMMAND = DONE [D]
6717 :   :   THEN
6718 :   :   SET SUC->SYS_DONE_FLAG
6719 :   :   ELSE
6720 :   :   GET SYSTEM UNDER CONTROL OFFSET
6721 :   :   IF COMMAND = DRIVE #1 SELECTED [E]
6722 :   :   THEN
6723 :   :   IF SUC-> DRIVE #1 AVAIL [F]
6724 :   :   THEN
6725 :   :   IF COMMAND=SIDE #1 SELECTED [G]
6726 :   :   THEN
6727 :   :   IF SUC-> SIDE#1_AVAIL NOT SET [H]
6728 :   :   THEN
6729 :   :   SET GET NEXT COMMAND FLAG
6730 :   :   ENDIF
6731 :   :   ENDIF
6732 :   :   ELSE
6733 :   :   SET GET NEXT COMMAND FLAG
6734 :   :   ENDIF
6735 :   :   ELSE
6736 :   :   IF SUC-> DRIVE #0_AVIAL SET [I]
6737 :   :   THEN
6738 :   :   IF COMMAND = SIDE #1 SELECTED [J]
6739 :   :   THEN
6740 :   :   IF SUC-> SIDE #1_AVIAL NOT SET [K]
6741 :   :   THEN
6742 :   :   SET GET NEXT COMMAND FLAG
6743 :   :   ENDIF
6744 :   :   ENDIF
6745 :   :   ELSE
6746 :   :   SET GET NEXT COMMAND FLAG
6747 :   :   ENDIF
6748 :   :   ENDIF
6749 :   :   ENDIF
6750 :   DOUNTIL GET NEXT_COMMAND=0
6751 :   IF COMMAND NOT = "DONE" [L]
6752 :   THEN
6753 :   :   IF COMMAND = READ SECTOR [M]
6754 :   :   THEN
6755 :   :   CALL TRK & SECTOR (MOD 2.2.2)
6756 :   :   ENDIF
6757 :   :   CALL DRVR (MOD 2.2.3)
6758 :   :   ENDIF
6759 :   ENDIF
6760 :ENDROUTINE
```

```

6762
6763 003272 053737 000514 004360 SYSVDR: BIS INITPG,INITTS ;PASS INITIALIZE FLAG TO TRACK & SECTOR MOD
6764 003300 005037 003556 CLR GTCMD ;CLEAR GET COMMAND FLAG
6765 003304 004737 010126 CALL SSUCOF ;CALL SET <SUC> OFFSET
6766 003310 005261 007670 INC WATINO(R1) ;INCREMENT SYSTEM WATCH DOG INITIALIZE
6767 003314 004737 005164 CALL ERRCHK ;CALL ERROR CHECK (MOD 2.2.5)
6768 003320 004737 010126 CALL SSUCOF ;CALL SET <SUC> OFFSET
6769 003324 005761 000612 IA22: TST SOAV(R1) ;IF SYSTEM AVAILABLE
6770 003330 001511 BEQ X22 ;NOT = 0, THEN
6771 003332 053761 003556 003560 BC22: BIS GTCMD,SODVDN(R1) ;SET <SUC> DRIVE DONE =GET COMMAND FLAG
6772 003340 005037 003556 CLR GTCMD ;CLEAR GET COMMAND FLAG
6773 003344 004737 003564 CALL GETCMD ;CALL GET COMMAND (MOD 2.2.1)
6774 003350 005737 000542 ID22: TST CMD ;IF COMMAND
6775 003354 002005 BGE LD22 ;EQUALS DONE (= -1), THEN
6776 003356 004737 010126 CALL SSUCOF ;CALL SET <SUC> OFFSET
6777 003362 005261 000626 INC SODN(R1) ;SET SYSTEM DONE FLAG
6778 003366 000451 BR UC22 ;BR TO UNTIL 'C'
6779 003370 004737 010126 LD22: CALL SSUCOF ;CALL SET <SUC> OFFSET
6780 003374 032737 000020 000542 IE22: BIT #DRV1,CMD ;IF COMMAND
6781 003402 001422 BEQ I122 ;HAS DRIVE #1 SELECTED
6782 003404 032761 000012 000612 IF22: BIT #12,SOAV(R1) ;IF SYSTEM DRIVE #1 AVAILABLE
6783 003412 001413 BEQ LF22 ;SET, THEN
6784 003414 032737 001000 000542 IG22: BIT #SIDE1,CMD ;IF COMMAND
6785 003422 001433 BEQ UC22 ;HAS SIDE # SELECTED
6786 003424 032761 000010 000612 IH22: BIT #10,SOAV(R1) ;IF <SUC> SIDE #1 AVAILABLE
6787 003432 001027 BNE UC22 ;NOT SET, THEN
6788 003434 005237 003556 INC GTCMD ;SET GET COMMAND FLAG
6789 003440 000424 BR UC22 ;BR TO ENDIF 'E'
6790 003442 005237 003556 LF22: INC GTCMD ;ELSE SET GET COMMAND FLAG
6791 003446 000421 BR UC22 ;BR TO END IF 'E'
6792 003450 032761 000005 000612 I122: BIT #5,SOAV(R1) ;IF SYSTEM DRIVE #0 AVAILABLE
6793 003456 001413 BEQ L122 ;SET, THEN
6794 003460 032737 001000 000542 IJ22: BIT #SIDE1,CMD ;IF COMMAND
6795 003466 001411 BEQ UC22 ;HAS SIDE #1 SELECTED, THEN
6796 003470 032761 000004 000612 IK22: BIT #4,SOAV(R1) ;IF SYSTEM SIDE #1 AVAILABLE
6797 003476 001005 BNE UC22 ;NOT SET, THEN
6798 003500 005237 003556 INC GTCMD ;SET GET COMMAND FLAG
6799 003504 000402 BR UC22 ;BR TO ENDIF 'E'
6800 003506 005237 003556 LI22: INC GTCMD ;SET GET COMMAND FLAG
6801 003512 005737 003556 UC22: TST GTCMD ;DO UNTIL GET NEXT CMD FLAG
6802 003516 001305 BNE BC22 ;EQUALS 0
6803 003520 005737 000542 IL22: TST CMD ;IF COMMAND
6804 003524 100413 BMI X22 ;NOT=DONE(-1), THEN
6805 003526 013702 000542 MOV CMD,R2 ;GET COMMAND TO TEST
6806 003532 042702 177761 BIC #177761,R2 ;CLEAR ALL BUT ACTUAL COMMAND
6807 003536 022702 000006 IM22: CMP #RDSEC,R2 ;IF COMMAND
6808 003542 001002 BNE EM22 ;SET TO READ SECTOR, THEN
6809 003544 004737 004016 CALL TRKSEC ;CALL TRACK & SECTOR UPDATE (MOD 2.2.2)
6810 003550 004737 004402 EM22: CALL DRVR ;CALL UNIT DRIVER (MOD 2.2.3)
6811 003554 000207 X22: RETURN ;RETURN TO CALLER
6812
6813 003556 000000 GTCMD: 0 ;GET COMMAND FLAG
6814 003560 000000 SODVDN: 0 ;SYSTEM#0 DRIVE DONE FLAG
6815 003562 000000 SIDVDN: 0 ;SYSTEM#1 DRIVE DONE FLAG
6816

```


6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852

```
.SBTTL MODULE 2.2.1 - GET COMMAND  
:BEGINROUTINE (MOD 2.2.1 - GET COMMAND)  
: INITIALIZE SYSTEM COMMAND TABLE POINTERS, IF INIT SET  
: GET SYSTEM UNDER CONTROL <SUC> COMMAND TABLE POINTER.  
: GET COMMAND AT POINTER  
: IF COMMAND = DONE [A]  
: THEN  
: RESET <SUC> COMMAND POINTER  
: ELSE  
: IF COMMAND = READ SECTOR [B]  
: THEN  
: GET <SUC> DRIVE DONE FLAG  
: IF DRIVE_DONE_FLAG SET [C]  
: THEN  
: CLEAR <SUC> DRIVE DONE FLAG  
: IF <SUC> NEW COMMAND FLAG NOT SET [E]  
: THEN  
: INCREMENT <SUC> COMMAND TABLE POINTER  
: GET COMMAND AT POINTER  
: IF COMMAND = DONE [D]  
: THEN  
: RESET <SUC> COMMAND TABLE POINTER  
: ENDIF  
: ENDIF  
: CLEAR <SUC> NEW COMMAND FLAG  
: ENDIF  
: ELSE  
: INCREMENT <SUC> COMMAND TABLE POINTER  
: SET <SUC> NEW COMMAND FLAG  
: ENDIF  
: ENDIF  
: CLEAR <SUC> DRIVE DONE FLAG  
: SET SELECTED DENSITY ONTO COMMAND  
:ENDROUTINE
```

```

6854
6855 003564 005737 000514 GETCMD: TST INITPG ;IF INITIALIZE
6856 003570 001404 BEQ 1$ ;SET, THEN
6857 003572 005037 003762 CLR SYOTPT ;CLEAR SYS #0 TABLE PTR
6858 003576 005037 003764 CLR SY1TPT ;CLEAR SYS #1 TABLE PTR
6859 003602 004737 010126 1$: CALL SSUCOF ;CALL SET <SUC> OFFSET
6860 003606 016102 003762 MOV SYOTPT(R1),R2 ;GET <SUC> CMD TABLE PTR
6861 003612 016237 003772 000542 MOV CMDTBL(R2),CMD ;GET COMMAND AT POINTER
6862 003620 013703 000542 MOV CMD,R3 ;SAVE COMMAND TEMP #3
6863 003624 005703 IA221: TST R3 ;IF COMMAND
6864 003626 002003 BGE LA221 ;EQUALS 'DONE', THEN
6865 003630 005061 003762 CLR SYOTPT(R1) ;RESET <SUC> CMD TABLE PTR
6866 003634 000444 BR EA221 ;BR TO EXIT MOD
6867 003636 042703 177761 LA221: BIC #177761,R3 ;CLEAR ALL BUT COMMAND
6868 003642 022703 000006 IB221: CMP #6,R3 ;IF COMMAND
6869 003646 001032 BNE LB221 ;EQUALS 'READ SECTOR', THEN
6870 003650 016104 003560 MOV SODVDN(R1),R4 ;GET <SUC> DRIVE DONE FLAG
6871 003654 005704 IC221: TST R4 ;IF <SUC> DRIVE DONE FLAG
6872 003656 001433 BEQ EA221 ;SET, THEN
6873 003660 005061 003560 CLR SODVDN(R1) ;CLEAR <SUC> DRIVE DONE FLAG
6874 003664 005761 003766 IE221: TST SOCMDF(R1) ;IF <SUC> NEW COMMAND FLAG
6875 003670 001016 BNE EE221 ;NOT SET, THEN
6876 003672 062761 000002 003762 ADD #2,SYOTPT(R1) ;INCREMENT <SUC> CMD TABLE PTR
6877 003700 016102 003762 MOV SYOTPT(R1),R2 ;GET <SUC> CMD TABLE PTR
6878 003704 016237 003772 000542 MOV CMDTBL(R2),CMD ;GET COMMAND AT POINTER
6879 003712 013703 000542 MOV CMD,R3 ;SAVE COMMAND IN TEMP#3
6880 003716 005703 ID221: TST R3 ;IF COMMAND
6881 003720 002012 BGE EA221 ;EQUALS 'DONE', THEN
6882 003722 005061 003762 CLR SYOTPT(R1) ;RESET CMD TABLE POINTER
6883 003726 005061 003766 EE221: CLR SOCMDF(R1) ;CLEAR <SUC> NEW COMMAND FLAG
6884 003732 000405 BR EA221 ;BR TO EXIT MOD
6885 003734 062761 000002 003762 LB221: ADD #2,SYOTPT(R1) ;INCREMENT <SUC> CMD TABLE PTR
6886 003742 005261 003766 INC SOCMDF(R1) ;SET <SUC> NEW COMMAND FLAG
6887 003746 053737 000506 000542 EA221: BIS DENS,CMD ;SET SELECTED DENSITY
6888 003754 005061 003560 CLR SODVDN(R1) ;CLEAR <SUC> DRIVE DONE FLAG
6889 003760 000207 X221: RETURN ;RETURN
6890
6891

```

```

6892 003762 000000 SYOTPT: 0 ;SYSTEM#0 COMMAND TABLE POINTER
6893 003764 000000 SY1TPT: 0 ;SYSTEM#1 COMMAND TABLE POINTER
6894 003766 000000 SOCMDF: 0 ;SYS#0 NEW COMMAND FLAG
6895 003770 000000 SICMDF: 0 ;SYS#1 NEW COMMAND FLAG
6896

```

```

6897
6898
6899 003772 000113 CMDTBL: .WORD 0113 ;DRV#/SIDEN/COMMAND /PTR
6900 003774 000133 .WORD 0133 ; 0 / 0 / READ MAINT STATUS /0
6901 003776 000111 .WORD 0111 ; 1 / 0 / READ MAINT STATUS /2
6902 004000 000131 .WORD 0131 ; 0 / 0 / SET DENSITY /4
6903 004002 000107 .WORD 0107 ; 1 / 0 / SET DENSITY /6
6904 004004 000127 .WORD 0127 ; 0 / 0 / READ SECTOR /10
6905 004006 001107 .WORD 1107 ; 1 / 0 / READ SECTOR /12
6906 004010 001127 .WORD 1127 ; 0 / 1 / READ SECTOR /14
6907 004012 177777 .WORD -1 ; 1 / 1 / READ SECTOR /16
6908 004014 177777 .WORD -1 ;DONE TERMINATOR /20
6909 .WORD -1 ;DONE TERMINATOR /22

```

```
6911 .SBTTL MODULE 2.2.2 - TRACK & SECTOR UPDATE
6912
6913 ;BEGINROUTINE (MOD 2.2.2 TRACK & SECTOR UPDATE)
6914 ; INITIALIZE MODULE, IF INITIALIZE PROGRAM SET
6915 ; IF SOFT_SWITCH_REG->CRC ALL TRACKS SET [A]
6916 ; THEN
6917 ; IF <SUC> DRIVE SECTOR DONE [B]
6918 ; THEN
6919 ; IF <SUC> DRIVE TRACK DONE [C]
6920 ; THEN
6921 ; CLEAR <SUC> DRIVE TRACK DONE
6922 ; SET <SUC> DRIVE TRACK = 76
6923 ; SET <SUC> DRIVE DONE
6924 ; ELSE
6925 ; DECREMENT <SUC> DRIVE TRACK
6926 ; SET TRACK=<SUC> DRIVE TRACK
6927 ; IF <SUC> DRIVE TRACK=0 [D]
6928 ; THEN
6929 ; SET <SUC> DRIVE TRACK DONE
6930 ; ENDF
6931 ; SET SECTOR = <SUC> DRIVE SECTOR
6932 ; ENDF
6933 ; CLEAR <SUC> DRIVE SECTOR DONE
6934 ; ELSE
6935 ; SET <SUC> DRIVE SECTOR=INTERLEAVE + <SUC> DRIVE SECTOR
6936 ; SET SECTOR=<SUC> DRIVE SECTOR
6937 ; IF <SUC> DRIVE SECTOR=26 [E]
6938 ; THEN
6939 ; SET <SUC> DRIVE SECTOR DONE FLAG
6940 ; SET <SUC> DRIVE SECTOR=1
6941 ; ELSE
6942 ; IF SECTOR >26 [F]
6943 ; THEN
6944 ; SET <SUC> DRIVE SECTOR & SECTOR 2
6945 ; ENDF
6946 ; ENDF
6947 ; SET TRACK = <SUC> DRIVE TRACK
6948 ; ENDF
6949 ; ELSE
6950 ; IF <SUC> DRIVE TRACK DONE FLAG SET [G]
6951 ; THEN
6952 ; SET TRACK=0
6953 ; SET <SUC> DRIVE TRACK=0
6954 ; CLEAR <SUC> DRIVE TRACK DONE
6955 ; SET <SUC> DRIVE DONE
6956 ; ELSE
6957 ; SET TRACK =76
6958 ; SET <SUC> DRIVE TRACK DONE FLAG
6959 ; ENDF
6960 ; SET <SUC> DRIVE SECTOR=1
6961 ; SET SECTOR= <SUC> DRIVE SECTOR
6962 ; ENDF
6963 ;ENDROUTINE
```

CZRXEAO RX02 FMTR PROG MACY11 30(1046) 20-SEP-78 10:48 PAGE 67-24
CZRXEAO.P11 20-SEP-78 10:48 MODULE 2.2.2 - TRACK & SECTOR UPDATE

SEQ 0035

6965
6966 004016 005737 004360

TRKSEC: TST INITTS ;IF INITIALIZE PROGRAM

```

6968 004022 001426 BEQ 1$ ;SET, THEN
6969 004024 005037 004360 CLR INITTS ;CLEAR INITIALIZE FLAG
6970 004030 005037 004362 CLR DVTKDN ;CLEAR SYS#0 DRIVE TRK DONE
6971 004034 005037 004364 CLR DVTKDN+2 ;CLEAR SYS#1 DRIVE TRK DONE
6972 004040 005037 004366 CLR DVSCDN ;CLEAR SYS#0 DRIVE SEC DONE
6973 004044 005037 004370 CLR DVSCDN+2 ;CLEAR SYS#1 DRIVE SEC DONE
6974 004050 012737 000114 004376 MOV #76.,DVTRK ;SET SYS#0 DRIVE TRK = 76
6975 004056 012737 000114 004400 MOV #76.,DVTRK+2 ;SET SYS#1 DRIVE TRK = 76
6976 004064 012737 000001 004372 MOV #1,DVSEC ;SET SYSTEM#0 DRVSEC=1
6977 004072 012737 000001 004374 MOV #1,DVSEC+2 ;SET SYSTEM#1 DRVSEC=1
6978 004100 004737 010126 1$: CALL SSUCOF ;CALL SET <SUC> OFFSET
6979 004104 032777 000200 174372 IA222: BIT #CRCALL,@SWR ;IF SWR -> CRC ALL TRACKS
6980 004112 001474 BEQ IG222 ;SET, THEN
6981 004114 005761 004366 IB222: TST DVSCDN(R1) ;IF <SUC> DRIVE SECTOR DONE FLAG
6982 004120 001433 BEQ LB222 ;SET, THEN
6983 004122 005761 004362 IC222: TST DVTKDN(R1) ;IF <SUC> DRIVE TRACK DONE FLAG
6984 004126 001410 BEQ LC222 ;SET, THEN
6985 004130 005061 004362 CLR DVTKDN(R1) ;CLEAR <SUC> DRIVE TRACK DONE FLAG
6986 004134 012761 000114 004376 MOV #76.,DVTRK(R1) ;SET <SUC> DRIVE TRACK =76
6987 004142 005261 003560 INC SODVDN(R1) ;SET <SUC> DRIVE DONE FLAG
6988 004146 000415 BR EC222 ;BR TO ENDF 'C'
6989 004150 016137 004376 000502 LC222: MOV DVTRK(R1),TRACK ;SET TRACK = <SUC> DRIVE TRACK
6990 004156 005361 004376 DEC DVTRK(R1) ;DECREMENT <SUC> DRIVE TRACK
6991 004162 005761 004376 ID222: TST DVTRK(R1) ;IF <SUC> DRIVE TRACK
6992 004166 001002 BNE ED222 ;EQUALS ZERO, THEN
6993 004170 005261 004362 INC DVTKDN(R1) ;SET <SUC> DRIVE TRACK DONE FLAG
6994 004174 012761 000001 004372 ED222: MOV #1,DVSEC(R1) ;SET <SUC> DRIVE SECTOR=1
6995 004202 005061 004366 EC222: CLR DVSCDN(R1) ;CLEAR <SUC> DRIVE SECTOR DONE FLAG
6996 004206 000463 BR X222 ;BR TO MOD EXIT
6997 004210 062761 000002 004372 LB222: ADD #INTLV,DVSEC(R1) ;SET <SUC> DRV SEC=INTERLEAVE + OLD DRV SEC
6998 004216 016137 004372 000500 MOV DVSEC(R1),SECTR ;SET SECTOR=<SUC> DRIVE SECTOR
6999 004224 026127 004372 000032 IE222: CMP DVSEC(R1),#26. ;IF <SUC> DRVE SECTOR
7000 004232 001006 BNE IF222 ;EQUALS 26, THEN
7001 004234 005261 004366 INC DVSCDN(R1) ;SET <SUC> DRIVE SECTOR DONE FLAG
7002 004240 012761 000001 004372 MOV #1,DVSEC(R1) ;SET <SUC> DRIVE SECTOR=1
7003 004246 000412 BR EE222 ;BR TO END 'E'
7004 004250 023727 000500 000032 IF222: CMP SECTR,#26. ;IF SECTOR
7005 004256 103406 BLO EE222 ;>26, THEN
7006 004260 012737 000002 000500 MOV #2,SECTR ;SET SECTOR=2
7007 004266 012761 000002 004372 MOV #2,DVSEC(R1) ;SET <SUC> DRIVE SECTOR = 2
7008 004274 016137 004376 000502 EE222: MOV DVTRK(R1),TRACK ;SET TRACK = <SUC> DRIVE TRACK
7009 004302 000425 BR X222 ;BR TO MOD EXIT
7010 004304 005761 004362 IG222: TST DVTKDN(R1) ;IF <SUC> DRIVE TRACK DONE FLAG
7011 004310 001407 BEQ LG222 ;SET, THEN
7012 004312 005061 004362 CLR DVTKDN(R1) ;CLEAR <SUC> DRIVE TRACK DONE FLAG
7013 004316 005037 000502 CLR TRACK ;SET TRACK=0
7014 004322 005261 003560 INC SODVDN(R1) ;SET <SUC> DRIVE DONE FLAG
7015 004326 000405 BR EG222 ;BR TO ENDF 'G'
7016 004330 012737 000114 000502 LG222: MOV #76.,TRACK ;SET TRACK=76
7017 004336 005261 004362 INC DVTKDN(R1) ;SET <SUC> DRIVE TRACK DONE FLAG
7018 004342 012761 000001 004372 EG222: MOV #1,DVSEC(R1) ;SET <SUC> DRIVE SECTOR=1
7019 004350 016137 004372 000500 MOV DVSEC(R1),SECTR ;SET SECTOR=<SUC> DRIVE SECTOR
7020 004356 000207 X222: RETURN ;RETURN - MOD EXIT
7021

```

```
7023  
7024          000002  
7025  
7026  
7027 004360 000000  
7028  
7029  
7030 004362 000000  
7031 004364 000000  
7032 004366 000000  
7033 004370 000000  
7034 004372 000001  
7035 004374 000001  
7036 004376 000114  
7037 004400 000114  
7038  
7039
```

```
-----  
;          INTLV = 2  
-----  
;INITTS: 0          ;INITIALIZE TRACK & SECTOR FLAG  
-----  
DVTKDN: .WORD 0          ;DRIVE TRACK DONE (DVTKDN)  SYS #0  
          .WORD 0          ;DRIVE TRACK DONE (DVTKDN)  SYS #1  
DVSCDN: .WORD 0          ;DRIVE SECTOR DONE (DVSCDN) SYS #0  
          .WORD 0          ;DRIVE SECTOR DONE (DVSCDN) SYS #1  
DVSEC:  .WORD 1          ;DRIVE SECTOR (DVSEC)     SYS #0  
          .WORD 1          ;DRIVE SECTOR (DVSEC)     SYS #1  
DVTRK:  .WORD 76.        ;DRIVE TRACK (DVTRK)     SYS #0  
          .WORD 76.        ;DRIVE TRACK (DVTRK)     SYS #1  
-----
```

7041
7042
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082

```

.SBTTL MODULE 2.2.3 - DRIVE DRIVER

:BEGINROUTINE (MOD 2.2.3 - DRIVE DRIVER)
: CALL SET <SUC> ADDRESS AND OFFSET
: SAVE COMMAND IN <SUC> COMMAND
: IF COMMAND=SET DENSITY
:   THEN
:     SET <SUC> WATCHDOG MULTIPLIER LIMIT=400
:   ELSE
:     SET <SUC> WATCHDOG MULTIPLIER LIMIT=20
: ENDIF
: SETUP SECTOR FOR PRINT
: SETUP TRACK FOR PRINT
: SETUP SECTOR ADR FOR OUTPUT
: SETUP TRACK ADR FOR OUTPUT
: SETUP COMMAND FOR OUTPUT
: CALL SET PRIORITY HIGH
: CALL OUTPUT COMMAND
: CALL SET PRIORITY LOW
:ENDROUTINE

-----
DRVR:  CALL  SSUCAD      ;CALL SET <SUC> ADDRESS & OFFSET
      MOV   CMD,SOCMD(R1) ;SAVE COMMAND IN <SUC> COMMAND
      MOV   CMD,R2       ;GET COMMAND
      BIC   #177761,R2   ;CLEAR ALL BUT COMMAND
IA223: CMP   #SETDEN,R2  ;IF COMMAND
      BNE   LA223        ;EQUALS 'SET DENSITY'
      MOV   #400,SOwMLT(R1) ;SET <SUC> WATCHDOG MULT LIMIT=400
      BR    EA223        ;BR TO END 'A'
LA223: MOV   #20,SOwMLT(R1) ;SET <SUC> WATCHDOG MULT LIMIT=20
EA223: MOV   SECTR,SOSEC(R1) ;SETUP SECTOR FOR PRINT
      MOV   TRACK,SOTRK(R1) ;SETUP TRACK FOR PRINT
      MOV   SECTR,SECADR   ;SETUP SECTOR ADR FOR OUTPUT
      MOV   TRACK,TRKADR  ;SETUP TRACK ADR FOR OUTPUT
      MOV   CMD,WDOT      ;SETUP COMMAND FOR OUTPUT
      CALL  PRIHI         ;CALL SET PRIORITY HIGH
      CALL  OUTCMD        ;CALL OUTPUT COMMAND
      CALL  PRILO         ;CALL SET PRIORITY LOW
X223:  RETURN            ;RETURN
-----

```

```

7084 .SBTTL MODULE 2.2.3.1 - OUTPUT DRIVE COMMAND
7085 -----
7086 004524 012737 000040 010226 OUTCMD: MOV #DNBIT,RDYWD ;READY TEST WD (PASS TO 2.2.3.1.1)
7087 004532 013737 000540 005102 MOV WDOT,WRDS ;WORD FOR OUTPUT (PASS TO 2.2.3.1.1)
7088 004540 013737 000472 005104 MOV RXCSAD,ADRS ;ADDRESS OF OUTPUT (PASS TO 2.2.3.1.1)
7089 004546 004737 005124 JSR PC,OUTSWD ;OUTPUT FUNCTION WD (FW DO 2.2.3.1.1)
7090 004552 032737 000010 000540 IB2231: BIT #10,WDOT ;THEN, IF FUNCTION IS
7091 004560 001043 BNE IC2231 ;'READ, WRITE, FILL, EMPTY' (FW BIT #3=0)
7092 004562 032737 000004 000540 IH2231: BIT #4,WDOT ;AND THEN IF FUNCTION IS
7093 004570 001047 BNE LH2231 ;'EMPTY, FILL' (FW BIT#2=0)
7094 004572 012737 000200 010226 MOV #TRBIT,RDYWD ;THEN SET OUTPUT READY TEST WORD (PASS TO 2.2.3.1.1)
7095 004600 013737 005122 005102 MOV WDCT,WRDS ;AND SET WORD FOR OUTPUT (PASS TO 2.2.3.1.1)
7096 004606 013737 000474 005104 MOV RXDBAD,ADRS ;AND SET ADDRESS OF OUTPUT (PASS TO 2.2.3.1.1)
7097 004614 004737 005124 JSR PC,OUTSWD ;OUTPUT BASE ADDRESS WORD DO 2.2.3.1.1
7098 004620 032737 000002 000540 IK2231: BIT #2,WDOT ;IF 'FILL' (FW BIT#1=0)
7099 004626 001004 BNE LK2231 ;THEN
7100 004630 013737 005116 005102 MOV BAFILL,WRDS ;SET DATA FILL BUFFER ADR (PASS TO 2.2.3.1.1)
7101 004636 000403 BR EK2231 ;BR TO END IF 'K'
7102 004640 013737 005120 005102 LK2231: MOV BAEMPTY,WRDS ;SET DATA EMPTY BUFFER ADR (PASS TO 2.2.3.1.1)
7103 004646 012737 000200 010226 EK2231: MOV #TRBIT,RDYWD ;SET OUTPUT READY TEST WORD (PASS TO 2.2.3.1.1)
7104 004654 013737 000474 005104 MOV RXDBAD,ADRS ;ADDRESS OF OUTPUT (PASS TO 2.2.3.1.1)
7105 004662 004737 005124 JSR PC,OUTSWD ;OUTPUT WORD COUNT WORD DO 2.2.3.1.1
7106 004666 000444 BR EH2231 ;BRANCH TO END IF 'H'
7107 004670 032737 000004 000540 IC2231: BIT #4,WDOT ;IF FUNCTION WORD IS
7108 004676 001455 BEQ IE2231 ;'WRITE D.D' OR 'READ E.C' (FW BIT #2=1)
7109 004700 032737 000002 000540 ID2231: BIT #2,WDOT ;THEN, IF FUNCTION IS
7110 004706 001035 BNE LD2231 ;'WRITE D.D', THEN (FW BIT#1=0)
7111 004710 012737 000200 010226 LH2231: MOV #TRBIT,RDYWD ;SET OUTPUT READY TEST WORD
7112 004716 013737 005114 005102 MOV SECADR,WRDS ;MOVE TRACK AND SECTOR ADDRESS
7113 004724 042737 177700 005102 BIC #177700,WRDS ;FORMAT TO SECTOR ADDRESS
7114 004732 013737 000474 005104 MOV RXDBAD,ADRS ;ADDRESS OF OUTPUT
7115 004740 004737 005124 JSR PC,OUTSWD ;OUTPUT SECTOR ADDRESS
7116 004744 013737 005112 005102 MOV TRKADR,WRDS ;MOVE TRACK AND SECTOR ADDRESS
7117 004752 042737 177600 005102 BIC #177600,WRDS ;FORMAT TRACK ADDRESS
7118 004760 012737 000200 010226 MOV #TRBIT,RDYWD ;SET OUTPUT READY TEST WORD
7119 004766 013737 000474 005104 MOV RXDBAD,ADRS ;ADDRESS OF OUTPUT
7120 004774 004737 005124 JSR PC,OUTSWD ;OUTPUT TRACK ADDRESS
7121 005000 000437 BR EH2231 ;ENDIF H -DONE
7122 005002 012737 000200 010226 LD2231: MOV #TRBIT,RDYWD ;SET READY WD TO TR MODE
7123 005010 012737 017446 005102 MOV #XER,WRDS ;EXT ERR. CODE TABLE ADD
7124 005016 013737 000474 005104 MOV RXDBAD,ADRS ;ADDRESS OF OUTPUT, RXDB
7125 005024 004737 005124 JSR PC,OUTSWD ;O/P BASE ADD FOR ERR. CODE
7126 005030 000423 BR END2231 ;DONE
7127 005032 032737 000002 000540 IE2231: BIT #2,WDOT ;IF FUNCTION IS
7128 005040 001404 BEQ LE2231 ;'READ STATUS' (FW BIT#1=1)
7129 005042 012737 000001 005106 TE2231: MOV #1,ERSTAT ;THEN-SET ERR STATUS FLAG
7130 005050 000413 BR END2231 ;DONE
7131 005052 012737 000200 010226 LE2231: MOV #TRBIT,RDYWD ;SET OUTPUT READY TEST WD
7132 005060 013737 005110 005102 MOV VALWD,WRDS ;VALIDATION WORD
7133 005066 013737 000474 005104 MOV RXDBAD,ADRS ;ADDRESS OF OUTPUT, RXDB
7134 005074 004737 005124 JSR PC,OUTSWD ;OUTPUT VALIDATION WORD
7135 005100 000207 END2231: RTS PC ;RETURN TO MOD 2.3
7136 -----
  
```



```

7138 ;-----
7139 005102 000000 WRDS: 0 ;MODULE 2.2.3.1.1 OUTPUT WORD
7140 005104 000000 ADRS: 0 ;MODULE 2.2.3.1.1 OUTPUT ADDRESS
7141 005106 000000 ERSTAT: 0 ;MODULE 0.0 ERR STATUS READ FLAG
7142 005110 000111 VALWD: 111 ;VALIDATION WD (SET DENS-ASCII 'I')
7143 005112 000000 TRKADR: 0 ;TRACK ADDRESS
7144 005114 000000 SECADR: 0 ;SECTOR ADDRESS
7145 005116 000000 BAFILL: 0 ;BASE ADDRESS FILL BUFFER
7146 005120 000000 BAEMPT: 0 ;BASE ADDRESS FMPTY BUFFER
7147 005122 000000 WDCT: 0 ;WORD COUNT
7148 ;MOD 2.2.3.1 ----- END MODULE -----
7149
7150 .SBTTL MODULE 2.2.3.1.1 - OUTPUT SINGLE WORD
7151 ;-----
7152
7153 005124 013737 000472 010230 OUTSWD: MOV RXCSAD,CSRADR ;SET C&S REG ADR
7154 005132 013737 010226 010226 MOV RDYWD,RDYWD ;OUTPUT READY WORD (PASS TO DELAY)
7155 005140 004737 010136 JSR PC,DELAY ;DELAY FOR READY DO DELAY
7156 005144 033777 010226 173320 BIT RDYWD,@RXCSAD ;IF READY,
7157 005152 001403 BEQ X22311 ;THEN
7158 005154 013777 005102 177722 MOV WRDS,@ADRS ;MOV WORD TO ADDRESS
7159 005162 000207 X22311: RTS PC ;RETURN
7160 ;MOD 2.2.3.1.1 ----- END MODULE -----

```

```
7162 .SBTTL MODULE 2.2.5 - ERROR CHECK
7163
7164 ;BEGINROUTINE (MOD 2.2.5 - ERROR CHECK)
7165 : CLEAR PRINT HEADER FLAG & ERROR FLAG
7166 : CALL SET <SUC> ADDRESS & OFFSET
7167 : GET <SUC> COMMAND (FROM LAST OPERATION)
7168 : IF <SUC> COMMAND=READ MAINT STATUS [A]
7169 : : THEN
7170 : : SET COMMAND HEADER='READ MAINT STATUS'
7171 : : SETUP CSR & ESR CHECK BITS & DON'T CARE BITS
7172 : : CALL CSR ERROR CHECK & CALL ESR ERROR CHECK
7173 : : SET ERROR FLAG, IF ANY ERRORS
7174 : : IF ERROR_FLAG SET [B]
7175 : : : THEN
7176 : : : SET ERROR=<READ MAINT STATUS COMMAND ERROR>
7177 : : : CALL DRIVES DROP (MOD 2.2.5.1)
7178 : : : CALL LOOP (MOD 2.2.5.4)
7179 : : : ELSE
7180 : : : CALL DRIVES AVAILABLE CHECK (MOD 2.2.5.2)
7181 : : : ENDF
7182 : : ELSE
7183 : : : IF <SUC> COMMAND=SET DENSITY [C]
7184 : : : : THEN
7185 : : : : SET COMMAND HEADER='SET DENSITY ERROR'
7186 : : : : CALL SET TEST BITS (MOD 2.2.5.3)
7187 : : : : CALL CSR ERROR CHECK & CALL ESR ERROR CHECK
7188 : : : : SET ERROR FLAG, IF ANY ERRORS
7189 : : : : IF ERROR_FLAG SET [D]
7190 : : : : : THEN
7191 : : : : : SET ERROR=<SET DENSITY COMMAND ERROR>
7192 : : : : : CALL DRIVES DROP (MOD 2.2.5.1)
7193 : : : : : CALL LOOP (MOD 2.2.5.4)
7194 : : : : : ENDF
7195 : : : : ELSE
7196 : : : : : IF <SUC> COMMAND=READ SECTOR [F]
7197 : : : : : : THEN
7198 : : : : : : SET COMMAND HEADER='READ SECTOR COMMAND ERROR'
7199 : : : : : : CALL SET TEST BITS (MOD 2.2.5.3)
7200 : : : : : : CALL CSR ERROR CHECK & CALL ESR ERROR CHECK
7201 : : : : : : SET ERROR FLAG, IF ANY ERRORS
7202 : : : : : : IF ERROR_FLAG SET [E]
7203 : : : : : : : THEN
7204 : : : : : : : IF RXDB=CRC ERROR [G]
7205 : : : : : : : : THEN
7206 : : : : : : : : CALL TRACK & SECTOR IDENT
7207 : : : : : : : : ENDF
7208 : : : : : : : SET ERROR=<READ SECTOR COMMAND ERROR>
7209 : : : : : : : CALL DRIVES DROP (MOD 2.2.5.1)
7210 : : : : : : : CALL LOOP (MOD 2.2.5.4)
7211 : : : : : : : ENDF
7212 : : : : : : ENDF
7213 : : : : ENDF
7214 : : ENDF
7215 :ENDROUTINE
```

```

7217 ;-----
7218
7219 005164 005037 000602 ERRCHK: CLR HDRFLG ;CLEAR HEADER FLAG
7220 005170 005037 005526 CLR ERRFLG ;CLEAR ERROR FLAGS
7221 005174 004737 010076 CALL SSUCAD ;CALL SET <SUC> ADDRESS & OFFSET
7222 005200 016102 000622 MOV SOCMD(R1),R2 ;GET <SUC> LAST COMMAND
7223 005204 042702 177761 BIC #177761,R2 ;CLEAR ALL BUT COMMAND
7224 005210 022702 000012 IA225: CMP #RDMNST,R2 ;IF COMMAND
7225 005214 001047 BNE IC225 ;EQUALS 'READ MAINT STATUS', THEN
7226 005216 012737 017346 000600 MOV #MSGCD5,TMSGAD ;SETUP COMMAND HEADER
7227 005224 012737 000040 000546 MOV #DNBIT,CSRMP ;SET DONE BIT CK-CSR COMPARE
7228 005232 052737 004000 000546 BIS #RX2BIT,CSRMP ;SET DOUBLE DENSITY BIT CK-CSR COMPARE
7229 005240 012737 173537 000544 MOV #173537,CSRMSK ;SET CSR DON'T CARE BITS
7230 005246 005037 000552 CLR ESRMP ;SET DRV RDY-ESR COMPARE
7231 005252 012737 171662 000550 MOV #171662,ESRMSK ;SET-ESR DON'T CARE BITS
7232 005260 004737 010404 CALL CSRCHK ;CALL CSR ERROR CHECK
7233 005264 010037 005526 MOV RO,ERRFLG ;SET ERROR FLAG, IF ANY ERRORS
7234 005270 004737 010624 CALL ESRCHK ;CALL ESR ERROR CHECK
7235 005274 050037 005526 BIS RO,ERRFLG ;SET ERROR FLAG, IF ANY ERRORS
7236 005300 005737 005526 IB225: TST ERRFLG ;IF ERROR FLAG
7237 005304 001410 BEQ LB225 ;SET, THEN
7238 005306 004737 012240 JSR PC,ERRSET
(2) 005312 000004 .WORD #4 ;FATAL ERR # 4 - READ MAINT STATUS COMMAND ERROR
7239 005314 004737 005530 CALL DRVDRP ;CALL DRIVES DROP (MOD 2.2.5.1)
7240 005320 004737 006406 CALL LOOP ;CALL LOOPING MODULE (MOD 2.2.5.4)
7241 005324 000476 BR EA225 ;BR TO MOD EXIT
7242 005326 004737 006040 LB225: CALL DRVAVL ;CALL DRIVES AVAIL CHK (MOD 2.2.5.2)
7243 005332 000473 BR EA225 ;BR TO MOD EXIT
7244 005334 022702 000010 IC225: CMP #SETDEN,R2 ;IF COMMAND
7245 005340 001030 BNE IF225 ;EQUALS 'SET DENSITY', THEN
7246 005342 012737 017311 000600 MOV #MSGCD4,TMSGAD ;SETUP COMMAND HEADER
7247 005350 004737 006266 CALL STSTBT ;CALL SET TEST BITS (MOD 2.2.5.3)
7248 005354 004737 010404 CALL CSRCHK ;CALL CSR ERROR CHECK
7249 005360 010037 005526 MOV RO,ERRFLG ;SET ERROR FLAG, IF ANY ERRORS
7250 005364 004737 010624 CALL ESRCHK ;CALL ESR ERROR CHECK
7251 005370 050037 005526 BIS RO,ERRFLG ;SET ERROR FLAG, IF ANY ERRORS
7252 005374 005737 005526 ID225: TST ERRFLG ;IF ERROR FLAG
7253 005400 001450 BEQ EA225 ;SET, THEN
7254 005402 004737 012240 JSR PC,ERRSET
(2) 005406 000005 .WORD #5 ;FATAL ERR # 5 - SET DENSITY COMMAND ERROR
7255 005410 004737 005530 CALL DRVDRP ;CALL DRIVES DROP (MOD 2.2.5.1)
7256 005414 004737 006406 CALL LOOP ;CALL LOOPING MODULE (MOD 2.2.5.4)
7257 005420 000440 BR EA225 ;
7258 005422 022702 000006 IF225: CMP #RDSEC,R2 ;IF COMMAND
7259 005426 001035 BNE EA225 ;EQUALS 'READ SECTOR', THEN
7260 005430 012737 017254 000600 MOV #MSGCD3,TMSGAD ;SET COMMAND HEADER
7261 005436 004737 006266 CALL STSTBT ;CALL SET TEST BITS (MOD 2.2.5.3)
7262 005442 004737 010404 CALL CSRCHK ;CALL CSR ERROR CHECK
7263 005446 010037 005526 MOV RO,ERRFLG ;SET ERROR FLAG, IF ANY ERRORS
7264 005452 004737 010624 CALL ESRCHK ;CALL ESR ERROR CHECK
7265 005456 050037 005526 BIS RO,ERRFLG ;SET ERROR FLAG, IF ANY ERRORS
7266 005462 005737 005526 IE225: TST ERRFLG ;IF ERROR FLAG
7267 005466 001415 BEQ EA225 ;SET, THEN
7268 005470 032777 000001 172776 IG225: BIT #CRCERR,@RXDBAD ;IF CRC ERROR
7269 005476 001402 BEQ EG225 ;SET, THEN
7270 005500 004737 011232 CALL PRTKSC ;CALL PRINT TRACK/SECTOR
  
```

7271 005504
(1) 005504 004737 012240
(2) 005510 000006
7272 005512 004737 005530
7273 005516 004737 006406
7274 005522 000240
7275 005524 000207
7276
7277 005526 000000
7278

```
EG225:      JSR      PC,ERRSET  
            .WORD   #6           ;FATAL ERR # 6 - READ SECTOR COMMAND ERROR  
            CALL   DRVDRP       ;CALL DRIVES DROP      (MOD 2.2.5.1)  
            CALL   LOOP        ;CALL LOOPING MODULE   (MOD 2.2.5.4)  
EA225:      NOP  
X225:      RETURN              ;RETURN  
-----  
ERRFLG: 0                      ;ERROR FLAG  
-----
```

7280
7281
7282
7283
7284
7285
7286
7287
7288
7289
7290
7291
7292
7293
7294
7295
7296
7297
7298
7299
7300
7301
7302
7303
7304
7305
7306
7307
7308
7309
7310
7311
7312
7313
7314
7315
7316
7317
7318
7319
7320
7321
7322
7323
7324
7325
7326
7327
7328
7329
7330
7331

```
.SBTTL MODULE 2.2.5.1 - DRIVE DROP  
:BEGINROUTINE (MOD 2.2.5.1 - DRIVE DROP)  
: CLEAR DROP DRIVE CONTROL WORD  
: CLEAR PRINT SIDE FLAG  
: CALL SET <SUC> OFFSET  
: IF <SUC> COMMAND=DRIVE #1 SELECTED [A]  
: THEN  
: : IF <SUC> COMMAND=SIDE #1 SELECTED [B]  
: : THEN  
: : : IF <SUC> SYS_AVAIL=SIDE #1 [C]  
: : : THEN  
: : : : SETUP PRINT SIDE=#1, DRIVE=#1  
: : : : SET DROP_DRIVE=SIDE 1/DRIVE 1  
: : : : SET PRINT SIDE FLAG  
: : : : ENDFIF  
: : : ELSE  
: : : : SET DROP_DRIVE=SIDE 0/DRIVE 1  
: : : : IF <SUC> SYS_AVAIL=SIDE #1 [D]  
: : : : THEN  
: : : : : SETUP PRINT SIDE=#0, DRIVE=#1  
: : : : : SET PRINT_SIDE_FLAG  
: : : : : ELSE  
: : : : : : SETUP PRINT DRIVE=#1  
: : : : : ENDFIF  
: : : : ENDFIF  
: : : ELSE  
: : : : IF <SUC> COMMAND=SIDE #1 SELECTED [E]  
: : : : THEN  
: : : : : IF <SUC> SYS_AVAIL=SIDE #1 [F]  
: : : : : THEN  
: : : : : : SETUP PRINT SIDE=#1, DRIVE=#0  
: : : : : : SET DROP_DRIVE=SIDE 1/DRIVE 0  
: : : : : : ENDFIF  
: : : : : ELSE  
: : : : : : SET DROP_DRIVE=SIDE 0/ DRIVE 0  
: : : : : : IF <SUC> SYS_AVAIL=SIDE #1 [G]  
: : : : : : THEN  
: : : : : : : SETUP PRINT SIDE=#0, DRIVE=#0  
: : : : : : : SET PRINT_SIDE_FLAG  
: : : : : : : ELSE  
: : : : : : : : SETUP PRINT DRIVE=#0  
: : : : : : : ENDFIF  
: : : : : : ENDFIF  
: : : : : ENDFIF  
: : : : IF DROP_DRIVE NOT=0 [H]  
: : : : : THEN  
: : : : : : CALL PRINT SYSTEM IDENTIFICATION  
: : : : : : CALL PRINT DRIVE IDENTIFICATION  
: : : : : : DROP DRIVES SET IN DROP_DRIVE FROM <SUC> AVAIL  
: : : : : ENDFIF  
: ENDFIF  
: ENDRROUTINE
```

```

7333
7334
7335 005530 005037 006036      DRVDRP: CLR      DROPDV      :CLEAR DROP DRIVES CONTROL WORD
7336 005534 005037 011230      CLR      PTSIDF      :CLEAR PRINT SIDE FLAG
7337 005540 004737 010126      CALL     SSUCOF      :CALL SET <SUC> OFFSET
7338 005544 032761 000020 000622 IA2251: BIT      #DRV1,SOCMD(R1) :IF <SUC> COMMAND=DRIVE #1
7339 005552 001447      BEQ      IE2251      :SELECTED, THEN
7340 005554 032761 001000 000622 IB2251: BIT      #SIDE1,SOCMD(R1) :IF <SUC> COMMAND=SIDE #1
7341 005562 001420      BEQ      LB2251      :SELECTED, THEN
7342 005564 032761 000010 000612 IC2251: BIT      #10,SOAV(R1) :IF <SUC> SYS AVAIL=SIDE #1
7343 005572 001502      BEQ      IH2251      :SET, THEN
7344 005574 012737 000001 011226 MOV      #1,SIDE      :SETUP PRINT SIDE=#1
7345 005602 012737 000001 011224 MOV      #1,DRIVE     :SETUP PRINT DRIVE=#1
7346 005610 012737 000010 006036 MOV      #10,DROPDV   :SET DROP DRIVE=SIDE 1/DRIVE 1
7347 005616 005237 011230      INC      PTSIDF      :SET PRINT SIDE FLAG
7348 005622 000466      BR       IH2251      :BR TO IF 'H'
7349 005624 012737 000002 006036 LB2251: MOV      #2,DROPDV   :SET DROP DRIVE=SIDE 0/DRIVE 1
7350 005632 032761 000010 000612 ID2251: BIT      #10,SOAV(R1) :IF <SUC> SYS AVAIL=SIDE #1
7351 005640 001410      BEQ      LD2251      :SET, THEN
7352 005642 005037 011226      CLR      SIDE        :SETUP PRINT SIDE=#0
7353 005646 012737 000001 011224 MOV      #1,DRIVE     :SETUP PRINT DRIVE=#1
7354 005654 005237 011230      INC      PTSIDF      :SET PRINT SIDE FLAG
7355 005660 000447      BR       IH2251      :BR TO IF 'H'
7356 005662 012737 000001 011224 LD2251: MOV      #1,DRIVE     :SETUP PRINT DRIVE=#1
7357 005670 000443      BR       IH2251      :BR TO IF 'H'
7358 005672 032761 001000 000622 IE2251: BIT      #SIDE1,SOCMD(R1) :IF <SUC> COMMAND=SIDE #1
7359 005700 001417      BEQ      LE2251      :SELECTED, THEN
7360 005702 032761 000001 000612 IF2251: BIT      #1,SOAV(R1)  :IF <SUC> AVAIL=SIDE #1/DRIVE #0
7361 005710 001433      BEQ      IH2251      :SET, THEN
7362 005712 012737 000001 011226 MOV      #1,SIDE        :SETUP PRINT SIDE=#1
7363 005720 005037 011224      CLR      DRIVE       :SETUP PRINT DRIVE=#0
7364 005724 012737 000004 006036 MOV      #4,DROPDV   :SET DROP DRIVE=SIDE 1/DRIVE 0
7365 005732 005237 011230      INC      PTSIDF      :SET PRINT SIDE FLAG
7366 005736 000420      BR       IH2251      :BR TO IF 'H'
7367 005740 012737 000001 006036 LE2251: MOV      #1,DROPDV   :SET DROP DRIVE=SIDE 0/DRIVE 0
7368 005746 032761 000004 000612 IG2251: BIT      #4,SOAV(R1)  :IF <SUC> SYS AVAIL=SIDE #1/DRIVE #0
7369 005754 001407      BEQ      LG2251      :SET, THEN
7370 005756 005037 011226      CLR      SIDE        :SETUP PRINT SIDE=#0
7371 005762 005037 011224      CLR      DRIVE       :SETUP PRINT DRIVE=#0
7372 005766 005237 011230      INC      PTSIDF      :SET PRINT SIDE FLAG
7373 005772 000402      BR       IH2251      :BR TO IF 'H'
7374 005774 005037 011224      LG2251: CLR      DRIVE       :SETUP PRINT DRIVE=#0
7375 006000 005737 006036      IH2251: TST      DROPDV     :IF DROP DRIVE CONTROL WORD
7376 006004 001413      BEQ      X2251      :SET, THEN
7377 006006 004737 011132      CALL     PRSYS       :CALL PRINT SYSTEM IDENT
7378 006012 004737 011154      CALL     PRDRV       :CALL PRINT DRIVE IDENT
7379 006016 043761 006036 000612 BIC      DROPDV,SOAV(R1) :DROP DRIVES SET IN <SUC> AVAIL
7380 006024 012704 015121      MOV      #MSG33,R4   :SET MSG-> 'DROPPED'
7381 006034 000207      CALL     TTOUT       :PRINT MSG
7382      X2251: RETURN      :RETURN
7383 006036 000000      DROPDV: 0          :DROP DRIVE CONTROL WORD
7384

```

```
7386 .SBTTL MODULE 2.2.5.2 - DRIVES AVAILABLE CHECK
7387
7388 :BEGINROUTINE (MOD 2.2.5.2 - DRIVES AVAILABLE CHECK)
7389 : CLEAR DROP AVAILABLE DRIVES CONTROL
7390 : IF <SUC> COMMAND=DRIVE #1 SELECTED [A]
7391 : THEN
7392 : IF RXES=DRIVE NOT RDY [B]
7393 : THEN
7394 : IF <SUC> AVAIL=DRIVE #1 SET [C]
7395 : THEN
7396 : SET DRIVE=1
7397 : SET DROP_AVAIL_DRIVE=#1
7398 : ENDF
7399 : ENDF
7400 : IF RXES=SIDE NOT RDY [D]
7401 : THEN
7402 : DROP <SUC> AVAIL-SIDE #1/DRIVE #1
7403 : ENDF
7404 : ELSE
7405 : IF RXES=DRIVE NOT RDY [E]
7406 : THEN
7407 : IF <SUC> AVAIL=DRIVE #0 SET [F]
7408 : THEN
7409 : SET DRIVE=0
7410 : SET DROP_AVAIL_DRIVE=#0
7411 : ENDF
7412 : ENDF
7413 : IF RXES=SIDE NOT RDY [G]
7414 : THEN
7415 : DROP <SUC> AVAIL-SIDE #1/DRIVE #1
7416 : ENDF
7417 : ENDF
7418 : IF DROP_AVAIL_DRIVE NOT=0 [H]
7419 : THEN
7420 : CALL PRINT SYSTEM IDENT
7421 : SETUP PRINT 'DRIVE NOT READY-IS DISK IN DRIVE B'
7422 : CALL PRINT SYSTEM IDENT
7423 : CLEAR SIDE PRINT FLAG
7424 : CALL PRINT DRIVE IDENT
7425 : DROP THE DRIVE SET IN DROP_AVAIL_DRIVE IN <SUC> AVAIL
7426 : SETUP & PRINT 'DROPPED'
7427 : ENDF
7428 :ENDROUTINE
```


7482
7483
7484
7485
7486
7487
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498
7499
7500
7501
7502
7503
7504
7505
7506
7507
7508
7509
7510
7511
7512
7513
7514
7515
7516
7517
7518
7519
7520
7521
7522
7523

```
.SBTTL MODULE 2.2.5.3 - SET TEST BITS IN CSR & ESR
;BEGINROUTINE (MOD 2.2.5.3 - SET TEST BITS IN CSR & ESR)
:  SETUP CSR_COMPARE=DONE BIT
:  SETUP CSR_COMPARE=DOUBLE DENSITY BIT
:  GET <SUC> LAST COMMAND
:  SAVE ALL READ/WRITE BITS
:  SETUP CSR_COMPARE=SET ALL READ/WRITE BITS
:  SETUP CSR_MASK=DON'T CARE BITS
:  CLEAR ALL BUT SIDE BIT FROM <SUC> LAST COMMAND
:  SETUP ESR_COMPARE=SIDE BIT AS SET FROM <SUC> LAST CMD
:  IF <SUC> LAST COMMAND=DRIVE#1 SELECTED [A]
:  : THEN
:  :   SETUP ESR_COMPARE=DRIVE#1
:  ENDIF
:  IF <SUC> LAST COMMAND=DOUBLE DENSITY BIT SET [B]
:  : THEN
:  :   SETUP ESR_COMPARE=DRIVE DOUBLE DENSITY BIT SET
:  ENDIF
:  SETUP ESR_MASK=DON'T CARE BITS
:ENDROUTINE
```

```
-----
7505 006266 012737 000040 000546 STSTBT: MOV #DNBIT,CSRCMP ;SET DONE BIT CK-CSR MODULE
7506 006274 052737 004000 000546 BIS #RX2BIT,CSRCMP ;SET DOUBLE DENSITY BIT CK-CSR COMPARE
7507 006302 016103 000622 MOV SOCMD(R1),R3 ;SET <SUC> COMMAND
7508 006306 042703 176257 BIC #176257,R3 ;SAVE ALL R/W BITS
7509 006312 050337 000546 BIS R3,CSRCMP ;SET R/W BITS FOR-CSR COMPARE
7510 006316 012737 070017 000544 MOV #70017,CSRMSK ;SET-CSR DON'T CARE BIT MASK
7511 006324 016103 000622 MOV SOCMD(R1),R3 ;GET <SUC> COMMAND
7512 006330 042703 176777 BIC #^CSIDE1,R3 ;CLEAR ALL BIT SIDE BIT
7513 006334 010337 000552 MOV R3,ESRCMP ;SET SIDE BIT (IF SELECTED)-ESR COMAPRE
7514 006340 032761 000020 000622 IA2253: BIT #DRV1,SOCMD(R1) ;IF DRIVE #1
7515 006346 001403 BEQ IB2253 ;SELECTED IN COMMAND, THEN
7516 006350 052737 000400 000552 BIS #DRIVE1,ESRCMP ;SET DRIVE #1 BIT-ESR COMPARE
7517 006356 032761 000400 000622 IB2253: BIT #DNBIT,SOCMD(R1) ;IF DOUBLE DENSITY
7518 006364 001403 BEQ EB2253 ;SELECTED IN COMMAND, THEN
7519 006366 052737 000040 000552 BIS #DRV1DEN,ESRCMP ;SET DRIVE DENSITY BIT-ESR COMPARE
7520 006374 012737 170000 000550 EB2253: MOV #170000,ESRMSK ;SET ESR DON'T CARE BIT MASK
7521 006402 000240 NOP
7522 006404 000207 X2253: RETURN ;RETURN
-----
```

```

7525          .SBTTL  MODULE 2.2.5.4 - LOOPING MODULE
7526          ;-----
7527
7528 006406 000240          LOOP:  NOP
7529 006410 005737 000516    TST   DBGFLG          ;IF DEBUG FLAG
7530 006414 001464          BEQ   X2254          ;SET, THEN
7531 006416 032777 101000 172060 BIT   #101000,@SWR   ;IF HALT ON ERR OR LOOP ON ERR
7532 006424 001460          BEQ   X2254          ;SET, THEN
7533 006426 005037 000602    CLR   HDRFLG        ;CLEAR HEADER FLAG
7534 006432 012737 017417 000600 MOV   #MSGLP,TMSGAD ;SET MODULE MESSAGE
7535 006440 016137 000622 000542 MOV   SOCMD(R1),CMD ;GET <SUC> COMMAND
7536 006446 013737 000542 000540 LOOP1: MOV  CMD,WDOT   ;PASS COMMAND TO OUTPUT MODULE
7537 006454 004737 010404    CALL  CSRCHK        ;CALL CSR CHECK
7538 006460 005700          TST   R0            ;IF RXCS ERROR
7539 006462 001416          BEQ   LOOP2        ;SET, THEN
7540 006464 004737 012262    JSR   PC,BSINIT    ;CHECK SWR FOR BUS INITIALIZE
(1) 006470 032777 001000 172006 BIT   #SW09,@SWR   ;CHECK FOR LOOP ON ERROR
(1) 006476 001363          BNE   LOOP1        ;GO TO LOOP ERROR
(1) 006500 004737 012240    JSR   PC,ERRSET    ;FATAL ERR # 7 - LOOPING MOD-CSR ERR
(2) 006504 000007          .WORD #7
(1) 006506 005777 171772    TST   @SWR         ;CHECK FOR HALT ON ERROR
(1) 006512 100001          BPL   1$           ;HALT IF SET
(1) 006514 000000          HALT
(1) 006516 000240          1$:  NOP           ;<LOOPING MOD-CSR ERR>
7541 006520 004737 010624    LOOP2: CALL  ESRCHK   ;CALL ESR CHECK
7542 006524 005700          TST   R0            ;IF RXES ERROR
7543 006526 001416          BEQ   LOOP3        ;SET, THEN
7544 006530 004737 012262    JSR   PC,BSINIT    ;CHECK SWR FOR BUS INITIALIZE
(1) 006534 032777 001000 171742 BIT   #SW09,@SWR   ;CHECK FOR LOOP ON ERROR
(1) 006542 001341          BNE   LOOP1        ;GO TO LOOP ERROR
(1) 006544 004737 012240    JSR   PC,ERRSET    ;FATAL ERR # 10 - LOOPING MOD-ESR ERR
(2) 006550 000010          .WORD #10
(1) 006552 005777 171726    TST   @SWR         ;CHECK FOR HALT ON ERROR
(1) 006556 100001          BPL   1$           ;HALT IF SET
(1) 006560 000000          HALT
(1) 006562 000240          1$:  NOP           ;<LOOPING MOD-ESR ERR>
7545 006564 000240    LOOP3: NOP
7546 006566 000207    X2254: RETURN
7547          ;-----

```

```
7549 .SBTTL MODULE 2.3 - SYSTEM STATUS
7550
7551 :BEGINROUTINE (MOD 2.3 - SYSTEM STATUS)
7552 : IF SYSTEM#0_AVAIL=0 & SYSTEM#0=SELECTED [A]
7553 : : THEN
7554 : : SET <SUC>=0
7555 : : CALL DROP SYSTEM (MOD 2.3.2)
7556 : : ENDIF
7557 : IF SYSTEM#1_AVAIL=0 & SYSTEM#1=SELECTED [B]
7558 : : THEN
7559 : : SET <SUC>=1
7560 : : CALL DROP SYSTEM (MOD 2.3.2)
7561 : : ENDIF
7562 : IF SYSTEM#0 & SYSTEM#1=NOT SELECTED [C]
7563 : : THEN
7564 : : SET SYSTEM FLAG
7565 : : SET MSG 'NO SYSTEM AVAILABLE TO FORMATER''
7566 : : ELSE
7567 : : IF SYSTEM#1=SELECTED [D]
7568 : : : THEN
7569 : : : IF SYSTEM#1_DONE=1 [E]
7570 : : : : THEN
7571 : : : : IF SYSTEM#0 SELECTED [F]
7572 : : : : : THEN
7573 : : : : : IF SYSTEM#0_DONE=1 [G]
7574 : : : : : : THEN
7575 : : : : : : SET SYSTEM FLAG
7576 : : : : : : CALL PRINT SYSTEM STATUS (MOD 2.3.1)
7577 : : : : : : CLEAR SYSTEM#0 & SYSTEM#1 DONE
7578 : : : : : : SETUP FORMATTER MSG
7579 : : : : : : ELSE
7580 : : : : : : CALL SYSTEM TIMEOUT CHECK (MOD 2.3.3)
7581 : : : : : : ENDIF
7582 : : : : : : ELSE
7583 : : : : : : SET SYSTEM FLAG
7584 : : : : : : CALL PRINT SYSTEM STATUS (MOD 2.3.1)
7585 : : : : : : CLEAR SYSTEM#1 DONE
7586 : : : : : : SETUP FORMATTER MSG
7587 : : : : : : ENDIF
7588 : : : : : : ELSE
7589 : : : : : : CALL SYSTEM TIMEOUT CHECK (MOD 2.3.3)
7590 : : : : : : ENDIF
7591 : : : : : : ELSE
7592 : : : : : : IF SYSTEM#0_DONE=1 [H]
7593 : : : : : : : THEN
7594 : : : : : : : SET SYSTEM FLAG
7595 : : : : : : : CALL PRINT SYSTEM STATUS (MOD 2.3.1)
7596 : : : : : : : CLEAR SYSTEM#0 DONE
7597 : : : : : : : SETUP FORMATTER MSG
7598 : : : : : : : ELSE
7599 : : : : : : : CALL SYSTEM TIMEOUT CHECK (MOD 2.3.3)
7600 : : : : : : : ENDIF
7601 : : : : : : : ENDIF
7602 : : : : : : : ENDIF
7603 : : : : : : : ENDROUTINE
```

```

7605 ;-----
7606
7607 006570 000240 SYSTAT: NOP
7608 006572 005737 000612 IA23: TST SOAV ; IF SYSTEM#0 AVAIL
7609 006576 001010 BNE IB23 ; EQUALS ZERO AND
7610 006600 032777 000001 171676 BIT #1,@SWR ; SYSTEM#0
7611 006606 001404 BEQ IB23 ; SELECTED, THEN
7612 006610 005037 000610 CLR SUC ; SET <SUC>=0
7613 006614 004737 007264 CALL SYSDRP ; CALL SYSTEM DROP (MOD 2.3.2)
7614 006620 005737 000614 IB23: TST S1AV ; IF SYSTEM#1 AVAIL
7615 006624 001011 BNE IC23 ; EQUALS ZERO AND
7616 006626 032777 000002 171650 BIT #2,@SWR ; SYSTEM#1
7617 006634 001405 BEQ IC23 ; SELECTED THEN
7618 006636 012737 000001 000610 MOV #1,SUC ; SET <SUC>=1
7619 006644 004737 007264 CALL SYSDRP ; CALL SYSTEM DROP (MOD 2.3.2)
7620 006650 032777 000003 171626 IC23: BIT #3,@SWR ; IF SYS#0 & SYS#1
7621 006656 001006 BNE ID23 ; NOT SELECTED OR DESELECTED, THEN
7622 006660 005237 000574 INC SYSFLG ; SET SYSTEM FLAG
7623 006664 012737 014206 000572 MOV #MSG8,SYSMSG ; SET SYSTEM MSG 'NO SYSTEMS TO FORMAT'
7624 006672 000471 BR EC23 ; BR TO END 'C'
7625 006674 032777 000002 171602 ID23: BIT #2,@SWR ; IF SYSTEM#1
7626 006702 001446 BEQ IH23 ; SELECTED, THEN
7627 006704 005737 000630 IE23: TST S1DN ; IF SYSTEM#1 DONE
7628 006710 001440 BEQ LE23 ; SET, THEN
7629 006712 032777 000001 171564 IF23: BIT #1,@SWR ; IF SYSTEM#0
7630 006720 001422 BEQ LF23 ; SELECTED, THEN
7631 006722 005737 000626 IG23: TST SODN ; IF SYSTEM#0 DONE
7632 006726 001414 BEQ LG23 ; SET, THEN
7633 006730 005237 000574 INC SYSFLG ; SET SYSTEM FLAG
7634 006734 004737 007062 CALL PTSYST ; CALL PRINT SYSTEM STATUS (MOD 2.3.1)
7635 006740 005037 000626 CLR SODN ; CLEAR SYSTEM#0 DONE
7636 006744 005037 000630 CLR S1DN ; CLEAR SYSTEM#1 DONE
7637 006750 012737 014246 000572 MOV #MSG9,SYSMSG ; SET SYSTEM MSG 'FORMAT COMPLETED'
7638 006756 000437 BR EC23 ; BR TO END 'C'
7639 006760 004737 007342 LG23: CALL SYTMCK ; CALL SYSTEM TIMEOUT CHECK (MOD 2.3.3)
7640 006764 000434 BR EC23 ; BR TO END 'C'
7641 006766 005237 000574 LF23: INC SYSFLG ; SET SYSTEM FLAG
7642 006772 004737 007062 CALL PTSYST ; CALL PRINT SYSTEM STATUS (MOD 2.3.1)
7643 006776 005037 000630 CLR S1DN ; CLEAR SYSTEM#1 DONE
7644 007002 012737 014246 000572 MOV #MSG9,SYSMSG ; SET SYSTEM MSG 'FORMAT COMPLETED'
7645 007010 000422 BR EC23 ; BR TO END 'C'
7646 007012 004737 007342 LE23: CALL SYTMCK ; CALL SYSTEM TIMEOUT CHECK (MOD 2.3.3)
7647 007016 000417 BR EC23 ; BR TO END 'C'
7648 007020 005737 000626 IH23: TST SODN ; IF SYSTEM#0 DONE
7649 007024 001412 BEQ LH23 ; SET, THEN
7650 007026 005237 000574 INC SYSFLG ; SET SYSTEM FLAG
7651 007032 004737 007062 CALL PTSYST ; CALL PRINT SYSTEM STATUS (MOD 2.3.1)
7652 007036 005037 000626 CLR SODN ; CLEAR SYSTEM#0 DONE
7653 007042 012737 014246 000572 MOV #MSG9,SYSMSG ; SET SYSTEM MSG 'FORMAT COMPLETED'
7654 007050 000402 BR EC23 ; BR TO END 'C'
7655 007052 004737 007342 LH23: CALL SYTMCK ; CALL SYSTEM TIMEOUT CHECK (MOD 2.3.3)
7656 007056 000240 EC23: NOP
7657 007060 000207 X23: RETURN ; RETURN
7658 ;-----

```

7660
7661
7662
7663
7664
7665
7666
7667
7668
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690

```
.SBTTL MODULE 2.3.1 - PRINT SYSTEM STATUS  
:BEGINROUTINE (MOD 2.3.1 - PRINT SYSTEM STATUS)  
: SET <SUC>=0  
: BEGINDO  
: : INITIALIZE DRIVE_PTR=1, DRIVE_CTR=0 [A]  
: : IF <SUC> DONE=SET [B]  
: : : THEN  
: : : PRINT 'SYSTEM: <SUC> FORMAT DONE ON FOLLOWING:'  
: : : BEGINDO [C]  
: : : : IF <SUC> AVAIL=BIT SET AT SYSTEM AVAIL PTR [D]  
: : : : : THEN  
: : : : : IF <SUC> AVAIL=EITHER SIDE AVAIL [E]  
: : : : : : THEN  
: : : : : : SET SIDE FLAG  
: : : : : : ELSE  
: : : : : : CLEAR SIDE FLAG  
: : : : : : ENDIF  
: : : : : : SET DRIVE #  
: : : : : : SET SIDE #  
: : : : : : CALL PRINT DRIVES IDENT  
: : : : : : ENDIF  
: : : : : : INCREMENT DRIVE COUNTER  
: : : : : : SHIFT LEFT DRIVE POINTER  
: : : : : : DO UNTIL DRIVE_COUNTER=1  
: : : : : ENDDO  
: : : : ENDIF  
: : : INCREMENT <SUC> '  
: : : DO UNTIL <SUC>=2  
: : ENDDO  
:ENDROUTINE
```

```

7692
7693
7694 007062
(1) 007062 012704 014452
(1) 007066 004737 013232
7695 007072 005037 000610
7696 007076 012737 000001 007262 BA231: MOV #MSG14,R4 ;SET MSG-> 'FORMAT DONE ON FOLLOWING'
7697 007104 005037 007260 CLR TTOUT ;PRINT MSG
7698 007110 004737 010126 CLR SUC ;SET <SUC>=0
7699 007114 005761 000626 007262 BA231: MOV #1,DRVPTR ;INITIALIZE DRIVE POINTER=1
7700 007120 001450 IB231: CLR DRVCNT ;INITIALIZE DRIVE COUNTER=0
7701 007122 004737 011132 CALL SSUCOF ;CALL SET <SUC> OFFSET
7702 007126 000240 IB231: TST SODN(R1) ;IF <SUC> DONE
7703 007130 033761 007262 000612 BC231: BEQ EB231 ;SET, THEN
7704 007136 001431 ID231: CALL PRTSYS ;CALL PRINT SYSTEM IDENT
7705 007140 032761 000014 000612 IE231: NOP ;
7706 007146 001403 IE231: BIT DRVPTR,SOAV(R1) ;IF DRIVE POINTER=<SUC> AVAIL BIT
7707 007150 005237 011230 BEQ ED231 ;SET, THEN
7708 007154 000402 IE231: BIT #14,SOAV(R1) ;IF EITHER SIDE IN <SUC> AVAIL
7709 007156 005037 011230 BEQ LE231 ;SET, THEN
7710 007162 013737 007260 011224 LE231: INC PTSIDF ;SET PRINT SIDE FLAG
7711 007170 042737 000002 011224 EE231: BR EE231 ;BR TO END 'E'
7712 007176 013737 007260 011226 EE231: CLR PTSIDF ;CLEAR PRINT SIDE FLAG
7713 007204 006237 011226 EE231: MOV DRVCNT,DRIVE ;GET DRIVE COUNT
7714 007210 042737 177776 011226 BIC #2,DRIVE ;CLEAR SIDE BIT (DRIVE=DRIVE#) PRINT
7715 007216 004737 011154 MOV DRVCNT,SIDE ;GET DRIVE COUNT
7716 007222 005237 007260 ASR SIDE ;SHIFT IT RIGHT TO SHOW SIDE BIT
7717 007226 006337 007262 BIC #177776,SIDE ;CLEAR ANY JUNK BITS
7718 007232 022737 000004 007260 UC231: CALL PRTDRV ;CALL PRINT DRIVE IDENT
7719 007240 001332 UC231: INC DRVCNT ;INCREMENT DRIVE COUNTER
7720 007242 005237 000610 EB231: ASL DRVPTR ;SHIFT LEFT DRIVE POINTER
7721 007246 022737 000002 000610 UA231: CMP #4,DRVCNT ;DO UNTIL DRIVE COUNT
7722 007254 001310 UA231: BNE BC231 ;EQUALS 4
7723 007256 000207 X231: INC SUC ;INCREMENT <SUC> TO NEXT SYSTEM
7724 ; CMP #2,SUC ;DO UNTIL <SUC>
7725 007260 000000 ; BNE BA231 ;EQUALS 2
7726 007262 000000 ; RETURN ;RETURN
7727 ;
DRVCNT: 0 ;DRIVE COUNTER
DRVPTR: 0 ;DRIVE POINTER

```

```

7729 .SBTTL MODULE 2.3.2 - SYSTEM DROP
7730
7731 ;BEGINROUTINE (MOD 2.3.2 SYSTEM DROP)
7732 ;   SETUP & PRINT MSG 'SYSTEM'
7733 ;   SETUP & PRINT SYSTEM#
7734 ;   IF SUC = 0
7735 ;       THEN
7736 ;           DESELECT SYSTEM#0
7737 ;       ELSE
7738 ;           DESELECT SYSTEM#1
7739 ;   ENDIF
7740 ;   SETUP & PRINT MSG 'NO DRIVES AVAIL - DROPPED'
7741 ;ENDROUTINE
7742
7743
7744
  
```

```

7745 007264 012704 016425 SYSDRP: MOV #MSG51,R4 ;SETUP MSG 'SYSTEM:'
7746 007270 004737 013232 ;CALL TTOUT ;PRINT IT!
7747 007274 013703 000610 ;MOV SUC,R3 ;SET SYSTE #
7748 007300 004737 013346 ;CALL OCTP ;PRINT IT!
7749 007304 005737 000610 IA232: TST SUC ;IF SYSTEM UNDER CONTROL
7750 007310 001004 ;BNE LA232 ;IS ZERO, THEN
7751 007312 042777 000001 171164 ;BIC #1,@SWR ;DESELECT SYSTEM ZERO
7752 007320 000403 ;BR EA232 ;BR TO ENDIF 'A'
7753 007322 042777 000002 171154 LA232: BIC #2,@SWR ;ELSE DESELECT SYSTEM ONE
7754 007330 012704 016475 EA232: MOV #MSG53,R4 ;SETUP MSG 'DROPPED FROM FORMATTER'
7755 007334 004737 013232 ;CALL TTOUT ;PRINT IT!
7756 007340 000207 X232: RETURN ;RETURN
7757
  
```

```

7759 .SBTTL MODULE 2.3.3 - SYSTEM TIMEOUT CHECK
7760
7761 ;BEGINROUTINE (MOD 2.3.3 - SYSTEM TIMEOUT CHECK)
7762 ; IF SYSTEM_ERROR NOT=0 [A]
7763 ; THEN
7764 ; IF SYSTEM_ERROR=SYS#0 TIME OUT ERROR [B]
7765 ; THEN
7766 ; SET <SUC>=0
7767 ; CALL SYSTEM IDENT
7768 ; SETUP & PRINT TIME OUT ERROR
7769 ; CLEAR SYS #0 AVAILABLE WORD
7770 ; ENDF
7771 ; IF SYSTEM_ERROR=SYS#1 TIMEOUT ERROR [C]
7772 ; THEN
7773 ; SET <SUC>=1
7774 ; CALL SYSTEM IDENT
7775 ; SETUP & PRINT TIMEOUT ERROR
7776 ; CLEAR SYS#1 AVAILABLE WORD
7777 ; ENDF
7778 ; ENDF
7779 ;ENDROUTINE
7780
7781 -----
  
```

```

7782 007342 000240 SYTMCK: NOP ;
7783 007344 005737 000606 IA233: TST SYSERR ;IF SYSTEM ERROR
7784 007350 001446 BEQ EA233 ;NOT = 0, THEN
7785 007352 032737 000010 000606 IB233: BIT #10,SYSERR ;IF SYSTEM ERROR
7786 007360 001417 BEQ IC233 ;EQUALS SYS#0 TIME OUT ERR
7787 007362 005037 000610 CLR SUC ;SET <SUC> = 0
7788 007366 004737 011132 CALL PRSYS ;CALL PRINT SYSTEM IDENT
7789 007372 012704 014272 MOV #MSG10,R4 ;SET MSG-> 'INTERRUPT DID NOT OCCUR'
(1) 007376 004737 013232 CALL TTOUT ;PRINT MSG
7790 007402 012704 015702 MOV #MSG41,R4 ;SET MSG-> '-TIME OUT ERROR'
(1) 007406 004737 013232 CALL TTOUT ;PRINT MSG
7791 007412 005037 000612 CLR SOAV ;CLEAR SYS#0 AVIALABLE
7792 007416 000423 BR EA233 ;BR TO END 'A'
7793 007420 032737 000020 000606 IC233: BIT #20,SYSERR ;IF SYSTEM ERROR
7794 007426 001417 BEQ EA233 ;EQUALS SYS#1 TIME OUT ERR , THEN
7795 007430 012737 000001 000610 MOV #1,SUC ;SET <SUC> = 1
7796 007436 004737 011132 CALL PRSYS ;CALL PRINT SYSTEM IDENT
7797 007442 012704 014272 MOV #MSG10,R4 ;SET MSG-> 'INTERRUPT DID NOT OCCUR'
(1) 007446 004737 013232 CALL TTOUT ;PRINT MSG
7798 007452 012704 015702 MOV #MSG41,R4 ;SET MSG-> '-TIME OUT ERROR'
(1) 007456 004737 013232 CALL TTOUT ;PRINT MSG
7799 007462 005037 000614 CLR S1AV ;CLEAR SYS#1 AVIALABLE
7800 007466 005037 000606 EA233: CLR SYSERR ;CLEAR SYSTEM ERROR
7801 007472 000207 RETURN ;RETURN
7802
  
```


7804
 7805
 7806 007474 005737 007670
 7807 007500 001406
 7808 007502 005037 007670
 7809 007506 005037 007566
 7810 007512 005037 007570
 7811 007516 005237 007566
 7812 007522 023737 007566 007674
 7813 007530 103415
 7814 007532 005037 007566
 7815 007536 005237 007570
 7816 007542 023737 007570 000556
 7817 007550 103405
 7818 007552 005037 007570
 7819 007556 052737 000010 000606
 7820 007564 000207
 7821
 7822 007566 000000
 7823 007570 000000
 7824
 7825
 7826
 7827
 7828
 7829 007572 005737 007672
 7830 007576 001406
 7831 007600 005037 007672
 7832 007604 005037 007664
 7833 007610 005037 007666
 7834 007614 005237 007664
 7835 007620 023737 007664 007674
 7836 007626 103415
 7837 007630 005037 007664
 7838 007634 005237 007666
 7839 007640 023737 007666 000560
 7840 007646 103405
 7841 007650 005037 007666
 7842 007654 052737 000020 000606
 7843 007662 000207
 7844
 7845 007664 000000
 7846 007666 000000
 7847
 7848 007670 000000
 7849 007672 000000
 7850 007674 177777
 7851
 7852

```

.SBTTL MODULE 2.4 - WATCH DOG SYSTEM#0
-----
WATCH0: TST      WATINO      ;IF WATCH DOG INIT0
        BEQ      1$          ;NOT = 0, THEN
        CLR      WATINO      ;CLEAR WATCHDOG INIT #0 FLAG
        CLR      WCNTRO      ;CLEAR WATCH DOG COUNTER #0
        CLR      WMLTO       ;CLEAR WATCH DOG MULTIPLIER #0
1$:     INC      WCNTRO      ;INCREMENT WATCH DOG COUNTER #0
        CMP      WCNTRO,WCNTMX ;IF COUNTER
        BLO      XWAT0       ;IS = TO COUNT MAX
        CLR      WCNTRO      ;CLEAR WATCH DOG COUNTER #0
        INC      WMLTO       ;INCREMENT WATCH DOG MULTIPLIER #0
        CMP      WMLTO,SOWMLT ;IF MULTIPLIER
        BLO      XWAT0       ;IS = TO MULTIPLIER MAX
        CLR      WMLTO       ;CLEAR WATCH DOG MULTIPLIER #0
        BIS      #10,SYSERR  ;SET SYSTEM ERR = TIME OUT
XWAT0:  RETURN              ;RETURN
-----
WCNTRO: 0                    ;WATCH DOG COUNTER #0
WMLTO:  0                    ;WATCH DOG MULTIPLIER #0
-----

```

```

.SBTTL MODULE 2.5 - WATCH DOG SYSTEM#1
-----
WATCH1: TST      WATIN1      ;IF WATCH DOG INIT1
        BEQ      1$          ;NOT = 1, THEN
        CLR      WATIN1      ;CLEAR WATCH DOG INIT #1 FLAG
        CLR      WCNTR1      ;CLEAR WATCH DOG COUNTER #1
        CLR      WMLT1       ;CLEAR WATCH DOG MULTIPLIER #1
1$:     INC      WCNTR1      ;INCREMENT WATCH DOG COUNTER #1
        CMP      WCNTR1,WCNTMX ;IF COUNTER
        BLO      XWAT1       ;IS = TO COUNT MAX
        CLR      WCNTR1      ;CLEAR WATCH DOG COUNTER #1
        INC      WMLT1       ;INCREMENT WATCH DOG MULTIPLIER #1
        CMP      WMLT1,S1WMLT ;IF MULTIPLIER
        BLO      XWAT1       ;IS = TO MULTIPLIER MAX
        CLR      WMLT1       ;CLEAR WATCH DOG MULTIPLIER #1
        BIS      #20,SYSERR  ;SET SYSTEM ERR = TIME OUT
XWAT1:  RETURN              ;RETURN
-----
WCNTR1: 0                    ;WATCH DOG COUNTER #1
WMLT1:  0                    ;WATCH DOG MULTIPLIER #1
-----
WATINO: 0                    ;WATCH DOG INITIALIZE FLAG #0
WATIN1: 0                    ;WATCH DOG INITIALIZE FLAG #1
WCNTMX: -1                   ;WATCH DOGS MAX COUNT LIMIT
-----

```

```

7854 .SBTTL MODULE 3.0 - OUTPUT SYSTEM DONE
7855
7856 ;BEGINROUTINE (MOD 3.0 - OUTPUT SYSTEM DONE)
7857 ; SETUP & PRINT 'DO YOU WANT TO FORMAT MORE DISKETTES?'
7858 ; CALL GET ANSWER
7859 ; IF ANSWER=YES [A]
7860 ; THEN
7861 ; : SETUP & PRINT 'REMOVE FORMATTED DISKETTS & INSERT DISKETTS TO BE FORMATTED''
7862 ; : SETUP & PRINT '-->TYPE 'CR' WHEN READY''
7863 ; : BGND0 [B]
7864 ; : GET OPERATOR RESPONSE-CALL TTY INPUT
7865 ; : DO UNTIL TTY CTR='CR''
7866 ; : ENDD0
7867 ; ELSE
7868 ; : SETUP & PRINT 'FORMATTER DONE-RESTART MONITOR, UPDATE PROGRAM-->TYPE CTRL-C TO S
7869 ; : SET SYSTEM FLAG=IDLE
7870 ; : ENABLE TTY INTERRUPTS & SET TTY GO BIT
7871 ; ENDF
7872 ;ENDROUTINE
7873
7874 -----
7875

```

```

7876 007676 OTSYDN:
(1) 007676 012704 016703 MOV #MSG56,R4 ;SET MSG-> DO YOU WANT TO FORMAT MORE DISKETTES?
(1) 007702 004737 013232 CALL TTOUT ;PRINT MSG
7877 007706 004737 012624 CALL GETANS ;CALL GET OPERATOR ANSWER
7878 007712 122737 000131 000530 IA30: CMPB #'Y,ANSWER ;IF ANSWER
7879 007720 001022 BNE LA30 ;IS YES, THEN
7880 007722 012704 016766 MOV #MSG57,R4 ;SET MSG-> REMOVE FORMATTED DISKETTES & INSERT DISKETTES
(1) 007726 004737 013232 CALL TTOUT ;PRINT MSG
7881 007732 012704 017065 MOV #MSG58,R4 ;SET MSG-> TYPE 'CR' WHEN READY
(1) 007736 004737 013232 CALL TTOUT ;PRINT MSG
7882 007742 004737 013156 BB30: CALL TTIN ;CALL TTY INPUT
7883 007746 122737 000015 000526 CMPB #15,TIB ;DO UNTIL
7884 007754 001372 BNE BB30 ;CHARACTER='CR''
7885 007756 012737 000002 000574 MOV #2,SYSFLG ;SET SYSTEM FLAG=RESTART
7886 007764 000414 BR X30 ;BR TO MODULE EXIT
7887 007766 LA30:
(1) 007766 012704 017117 MOV #MSG59,R4 ;SET MSG-> FORMATTER DONE-RESTART MONITOR, UPDATE PROGRA
(1) 007772 004737 013232 CALL TTOUT ;PRINT MSG
7888 007776 012737 000100 000574 MOV #100,SYSFLG ;SET SYSTEM FLAG=IDLE
7889 010004 052777 000100 170266 BIS #BIT6,@TKS ;SET TTY KEYBOARD INTERRUPT BIT
7890 010012 005277 170262 INC @TKS ;SET TTY KEYBOARD GO BIT
7891 010016 000207 X30: RETURN ;RETURN
7892 -----

```

```
7894 .SBTTL MODULE U.PRIHI - SET PROCESSOR PRIORITY HIGH
7895 ;-----
7896
7897 010020 012737 000340 010074 PRIHI: MOV #PR7,NEWPRI ;SETUP FOR PROCESSOR PRI LEV-7
7898 010026 004737 010050 CALL SETPRI ;SET PROCESSOR FLAG
7899 010032 000207 RETURN ;RETURN
7900 ;-----
7901
7902 .SBTTL MODULE U.PRILO - SET PROCESSOR PRIORITY LOW
7903 ;-----
7904
7905 010034 012737 000000 010074 PRILO: MOV #PRO,NEWPRI ;SETUP FOR PROCESSOR PRI LEV=0
7906 010042 004737 010050 CALL SETPRI ;SET PROCESSOR PRI
7907 010046 000207 RETURN ;RETURN
7908 ;-----
7909
7910 .SBTTL MODULE U.PROPRI - SET PROCESSOR PRI
7911 ;-----
7912
7913 010050 005737 000520 SETPRI: TST LSIFLG ;IF PROCESSOR IS
7914 010054 001403 BEQ 1$ ;LSI, THEN
7915 010056 106437 010074 MTPS NEWPRI ;SET PROCESSOR PRI
7916 010062 000403 BR SETPIX ;BR TO END
7917 010064 013777 010074 167704 1$: MOV NEWPRI,@PSW ;SET PROCESSOR PRI
7918 010072 000207 SETPIX: RTS PC ;RETURN
7919 ;-----
7920 010074 000000 NEWPRI: 0 ;NEW PROCESSOR PRIORITY
7921 ;-----
7922
7923 .SBTTL MODULE U.SADR - SET SYSTEM BUS ADDRESS
7924 ;-----
7925
7926 010076 004737 010126 SSUCAD: CALL SSUCOF ;CALL SET <SUC> OFFSET
7927 010102 016137 000314 000472 MOV REGS0(R1),RXCSAD ;SET NEW SYSTEM ADDRESS
7928 010110 016137 000314 000474 MOV REGS0(R1),RXDBAD ;SET NEW SYSTEM DB ADDR
7929 010116 062737 000002 000474 ADD #2,RXDBAD ;BUMP ADDRESS
7930 010124 000207 RETURN ;RETURN
7931 ;-----
7932
7933 .SBTTL MODULE U.SUCO - SET SYSTEM UNDER CONTROL OFFSET
7934 ;-----
7935
7936 010126 013701 000610 SSUCOF: MOV SUC,R1 ;GET SYSTEM UNDER CONTROL
7937 010132 006301 ASL R1 ;DOUBLE IT! FOR WORD OFFSET ADDRESSING
7938 010134 000207 RETURN ;RETURN
7939 ;-----
```

```

7941      .SBTTL  MODULE U.DL - DELAY FOR 'TR' OR 'DONE'
7942      :-----
7943 010136 013704 010222      DELAY:  MOV    RYDX,R4      ;SET READY DELAY MULT
7944 010142 013703 010224      BDAUDL: MOV    RYDLY,R3     ;SET READY DELAY
7945 010146 033777 010226 000054 BDBUDL: BIT    RDYWD,@CSRADR ;IF READY
7946 010154 001021              BNE    XUDL              ;EQUAL TO '1' ,THEN BR TO EXIT
7947 010156 005303              DEC    R3                ;ELSE DECREMENT DELAY
7948 010160 001372              BNE    BDBUDL           ;DO UNTIL R3=0
7949 010162 012737 000007 000524      MOV    #7,TOB           ;SET UP BELL
7950 010170 004737 013700              JSR    PC,TOG           ;PRINT BELL
7951 010174 005304              DEC    R4                ;DECREMENT DELAY MULT.
7952 010176 001361              BNE    BDAUDL           ;DO UNTIL R4=0
7953 010200 052737 040000 000606      BIS    #40000,SYSERR    ;SET TIME OUT ERR
7954 010206 013737 010226 010402      MOV    RDYWD,TOERBT    ;SET T.O. WORD
7955 010214 004737 010232              JSR    PC,TOPT         ;GO PRINT TIMEOUT ERRORS
7956 010220 000207              XUDL:  RTS    PC        ;RETURN TO CALLING MOD
7957      :-----
7958 010222 000015      RYDX:   15              ;READY MULTIPLIER
7959 010224 100000      RYDLY: 100000          ;READY DELAY
7960 010226 000000      RDYWD:  0              ;READY WORD - TEST FOR DEVICE READY
7961 010230 000000      CSRADR: 0             ;C&S REG OF UNIT- WAITING FOR
7962      :-----
7963      :----- END MODULE -----
7964      .SBTTL  TIME OUT ERROR PRINT
7965      :-----
7966 010232 032777 020000 170244      TOPRT:  BIT    #BIT13,@SWR ;IF INHIBIT ERROR RESULTS
  
```

| | | | | | | | | |
|------|--------|--------|--------|--------|------|-------------|---------------|-----------------------------|
| 7968 | 010240 | 001057 | | | | BNE | TOPRTX | : IS NOT SET, THEN |
| 7969 | 010242 | 004737 | 011042 | | | JSR | PC,TSHDCK | : CALL TEST HEADER CHECK |
| 7970 | 010246 | 032737 | 000040 | 010402 | 1\$: | BIT | #DNBIT,TOERBT | : IF READY WORD WAS SET TO |
| 7971 | 010254 | 001405 | | | | BEQ | 2\$ | : TEST DONE BIT, THEN |
| 7972 | 010256 | 012704 | 015031 | | | MOV | #MSG30,R4 | : SET DONE NOT SET MSG |
| 7973 | 010262 | 004737 | 013232 | | | JSR | PC,TTOUT | : PRINT MSG |
| 7974 | 010266 | 000404 | | | | BR | 3\$ | : GO PRINT MSG |
| 7975 | 010270 | 012704 | 015055 | | 2\$: | MOV | #MSG31,R4 | : SET TR BIT NOT SET MSG |
| 7976 | 010274 | 004737 | 013232 | | | JSR | PC,TTOUT | : PRINT MSG |
| 7977 | 010300 | 012704 | 015702 | | 3\$: | MOV | #MSG41,R4 | : SET TIME OUT MSG |
| 7978 | 010304 | 004737 | 013232 | | | JSR | PC,TTOUT | : PRINT MSG |
| 7979 | 010310 | 032777 | 040000 | 170166 | | BIT | #SW14,@SWR | : IF EXTENDED ERROR REPORTS |
| 7980 | 010316 | 001430 | | | | BEQ | TOPRTX | : SET, THEN |
| 7981 | 010320 | 012704 | 013742 | | | MOV | #MSG1,R4 | : SET REG MSG |
| 7982 | 010324 | 004737 | 013232 | | | JSR | PC,TTOUT | : PRINT MSG |
| 7983 | 010330 | 017703 | 170136 | | | MOV | @RXCSAD,R3 | : SET TO PRINT RXCS |
| 7984 | 010334 | 004737 | 013334 | | | JSR | PC,OCTPE | : PRINT |
| 7985 | 010340 | 012704 | 014564 | | | MOV | #MSG21,R4 | : SET SPACES MSG |
| 7986 | 010344 | 004737 | 013232 | | | JSR | PC,TTOUT | : PRINT |
| 7987 | 010350 | 017703 | 170120 | | | MOV | @RXDBAD,R3 | : SET TO PRINT RXES |
| 7988 | 010354 | 004737 | 013334 | | | JSR | PC,OCTPE | : PRINT |
| 7989 | 010360 | 012704 | 014564 | | | MOV | #MSG21,R4 | : SET SPACES |
| 7990 | 010364 | 004737 | 013232 | | | JSR | PC,TTOUT | : PRINT |
| 7991 | 010370 | 013703 | 000542 | | | MOV | CMD,R3 | : SET COMMAND |
| 7992 | 010374 | 004737 | 013346 | | | JSR | PC,OCTP | : PRINT COMMAND |
| 7993 | 010400 | 000207 | | | | TOPRTX: RTS | PC | : RETURN |
| 7994 | | | | | | ----- | | |
| 7995 | 010402 | 000000 | | | | TOERBT: 0 | | |
| 7996 | | | | | | ----- | | |

7998
 7999
 8000
 8001
 8002
 8003
 8004
 8005
 8006
 8007
 8008
 8009
 8010
 8011
 8012
 8013
 8014
 8015
 8016
 8017
 8018
 8019
 8020
 8021
 8022
 8023
 8024
 8025
 8026
 8027
 8028
 8029
 8030
 8031
 8032
 8033
 8034
 8035
 8036
 8037

010404 017701 170062
 010410 043701 000544
 010414 023701 000546
 010420 001471
 010422 010137 011540
 010426 013737 000546 011536
 010434 012737 011544 011542
 010442 012737 000004 011534
 010450 012737 000014 011532
 010456 032777 020000 170020
 010464 001044
 010466 004737 011042
 010472 012704 014722
 010476 004737 013232
 010502 032777 040000 167774
 010510 001430
 010512 012704 014411
 010516 004737 013232
 010522 013703 000546
 010526 004737 013334
 010532 012704 014564
 010536 004737 013232
 010542 013703 011540
 010546 004737 013334
 010552 012704 014564
 010556 004737 013232
 010562 017703 167704
 010566 004737 013334
 010572 004737 011314
 010576 012700 177777
 010602 000405
 010604 005000
 010606 053700 000606
 010612 005037 000606
 010616 050037 000554
 010622 000207

```

.SBTTL RXCS ERROR CHECK
-----
CSRCHK: MOV @RXCSAD,R1 ;GET RXCS
        BIC CSRMSK,R1 ;MASK OFF BITS DON'T CARE ABOUT
        CMP CSRCMP,R1 ;IF RXCS CONTAINS
        BEQ 5$ ;ERRORS, THEN
1$: MOV R1,BADWRD ;SET BAD WORD
    MOV CSRCMP,CMPWRD ;SET COMPARE WORD
    MOV #CMSTB,TABADR ;SET MSG TABLE ADDRESS
    MOV #4,BITOFF ;SET # BITS TO OFFSET WORD
    MOV #12,BITLIM ;SET # BITS TO CHECK
    BIT #SW13,@SWR ;IF INHIBIT ERROR REPORTS
    BNE 4$ ;NOT SET, THEN
    CALL TSHDCK ;CALL TEST HEADER CHECK
2$: MOV #MSG26,R4 ;SET RXCS MSG HEADER
    JSR PC,TTOUT ;PRINT MSG
    BIT #SW14,@SWR ;IF EXTENDED ERROR REPORTS
    BEQ 3$ ;SET, THEN
    MOV #MSG13,R4 ;SET REG FORMAT MSG
    JSR PC,TTOUT ;PRINT MSG
    MOV CSRCMP,R3 ;SET GOOD RXCS
    JSR PC,OCTPE ;PRINT GOOD RXCS
    MOV #MSG21,R4 ;SET SPACES MSG
    JSR PC,TTOUT ;PRINT SPACES
    MOV BADWRD,R3 ;SET COMPARED RXCS
    JSR PC,OCTPE ;PRINT COMPARED RXCS
    MOV #MSG21,R4 ;SET SPACES MSG
    JSR PC,TTOUT ;PRINT MSG
    MOV @RXCSAD,R3 ;SET FULL RXCS
    JSR PC,OCTPE ;PRINT FULL RXCS
3$: JSR PC,CKBITS ;REPORT BAD BITS
4$: MOV #-1,R0 ;SET ERR
    BR CSRCKX ;BR TO END
5$: CLR R0 ;CLEAR ERRORS
    BIS SYSERR,R0 ;SET ANY SYSTEM ERRORS
    CLR SYSERR ;CLEAR SYS ERROR
CSRCKX: BIS R0,TSTERR ;SET TEST ERROR FLAG, IF ERRORS
        RTS PC ;RETURN
-----

```

```

8039      .SBTTL  RXES ERROR CHECK
8040      :-----:
8041
8042 010624 017701 167644      ESRCHK: MOV      @RXDBAD,R1      .GET RXES
8043 010630 043701 000550      BIC      ESRMSK,R1      ;MASK OFF BITS DON'T CARE ABOUT
8044 010634 023701 000552      CMP      ESRCMP,R1      ;IF RXES CONTAINS
8045 010640 001470      BEQ      5$              ;ERRORS, THEN
8046 010642 010137 011540      1$:  MOV      R1,BADWRD      ;SET BAD WORD
8047 010646 013737 000552 011536  MOV      ESRCMP,CMPWRD      ;SET COMPARE WORD
8048 010654 012737 011574 011542  MOV      #EMSGT8,TABADR      ;SET MSG TABLE ADR
8049 010662 005037 011534      CLR      BITOFF          ;SET BIT OFFSET
8050 010666 012737 000014 011532  MOV      #12,BITLIM        ;SET # BITS TO CHECK
8051 010674 032777 020000 167602  BIT      #SW13,@SWR        ;IF INHIBIT ERROR REPORTS
8052 010702 001044      BNE      4$              ;NOT SET, THEN
8053 010704 004737 011042      CALL     TSHDCK          ;CALL TEST HEADER CHECK
8054 010710 012704 014736      2$:  MOV      #MSG27,R4      ;SET RXES MSG HEADER
8055 010714 004737 013232      JSR      PC,TTOUT        ;PRINT MSG
8056 010720 032777 040000 167556  BIT      #SW14,@SWR        ;IF EXTENDED ERROR REPORTS
8057 010726 001430      BEQ      3$              ;SET, THEN
8058 010730 012704 014411      MOV      #MSG13,R4      ;SET REG FORMAT MSG
8059 010734 004737 013232      JSR      PC,TTOUT        ;PRINT MSG
8060 010740 013703 000552      MOV      ESRCMP,R3      ;SET GOOD RXES
8061 010744 004737 013334      JSR      PC,OCTPE        ;PRINT GOOD RXES
8062 010750 012704 014564      MOV      #MSG21,R4      ;SET SPACES MSG
8063 010754 004737 013232      JSR      PC,TTOUT        ;PRINT SPACES
8064 010760 013703 011540      MOV      BADWRD,R3      ;SET COMPARED RXES
8065 010764 004737 013334      JSR      PC,OCTPE        ;PRINT COMPARED RXCS
8066 010770 012704 014564      MOV      #MSG21,R4      ;SET SPACES MSG
8067 010774 004737 013232      JSR      PC,TTOUT        ;PRINT MSG
8068 011000 017703 167470      MOV      @RXDBAD,R3      ;SET FULL RXCS
8069 011004 004737 013334      JSR      PC,OCTPC        ;PRINT FULL RXCS
8070 011010 004737 011314      3$:  JSR      PC,CKBITS      ;REPORT BAD BITS
8071 011014 012700 177777      4$:  MOV      #-1,R0        ;SET ERR
8072 011020 000405      BR       ESRCKX          ;BR TO END
8073 011022 005000      5$:  CLR      R0              ;CLEAR ERRORS
8074 011024 053700 000606      BIS      SYSERR,R0      ;SET ANY SYSTEM ERRORS
8075 011030 005037 000606      CLR      SYSERR        ;CLEAR SYSTEM ERRORS
8076 011034 050037 000554      ESRCKX: BIS      R0,TSTERR ;SET TEST ERROR FLAG, IF ERRORS
8077 011040 000207      RTS      PC              ;RETURN
8078      :-----:

```

```
8080      .SBTTL TEST HEADER CHECK & PRINT
8081      :-----
8082
8083 011042 005737 000602      TSHDCK: TST      HDRFLG      ;IF HEADER FLAG
8084 011046 001030              BNE      ENDTSH      ;NOT SET, THEN
8085 011050 005237 000602      INC      HDRFLG      ;SET TEST HEADER FLAG
8086 011054 013704 000600      MOV      TMSGAD,R4   ;SET TEST MSG #
8087 011060 004737 013232      JSR      PC,TTOUT    ;PRINT TEST MSG
8088 011064 004737 010126      CALL    SSUCOF       ;CALL SET <SUC> OFFSET
8089 011070 004737 011132      CALL    PRSYS       ;CALL PRINT SYSTEM IDENTIFICATION
8090 011074 032761 000020 000622 BIT      #DRV1,SOCMD(R1) ;IF DRIVE #1
8091 011102 001404              BEQ      1$          ;SELECTED, THEN
8092 011104 012737 000001 011224 MOV      #1,DRIVE     ;SET DRIVE=#1
8093 011112 000402              BR       2$          ;BR TO 2$
8094 011114 005037 011224      1$:      CLR      DRIVE ;SET DRIVE=#0
8095 011120 005037 011230      2$:      CLR      PTSIDF ;CLEAR PRINT SIDE FLAG
8096 011124 004737 011154      CALL    PRDRV       ;CALL PRINT DRIVE
8097 011130 000207              ENDTSH: RTS      PC ;RETURN
8098      :-----
```


8100
8101
8102 011132
 (1) 011132 012704 016425
 (1) 011136 004737 013232
8103 011142 013703 000610
8104 011146 004737 013346
8105 011152 000207
8106
8107
8108
8109
8110
8111 011154
 (1) 011154 012704 014547
 (1) 011160 004737 013232
8112 011164 013703 011224
8113 011170 004737 013346
8114 011174 005737 011230
8115 011200 001410
8116 011202 012704 014556
 (1) 011206 004737 013232
8117 011212 013703 011226
8118 011216 004737 013346
8119 011222 000207
8120
8121 011224 000000
8122 011226 000000
8123 011230 000000
8124
8125
8126
8127
8128
8129 011232
 (1) 011232 012704 014570
 (1) 011236 004737 013232
8130 011242 012704 014506
 (1) 011246 004737 013232
8131 011252 016103 011304
8132 011256 004737 013346
8133 011262 012704 014526
 (1) 011266 004737 013232
8134 011272 016103 011310
8135 011276 004737 013346
8136 011302 000207
8137
8138 011304 000000
8139 011306 000000
8140 011310 000000
8141 011312 000000
8142

.SBTTL MODULE U.PRYSYS - PRINT SYSTEM IDENTIFICATION

PRYSYS:

```
MOV    #MSG51,R4      ;SET MSG-> 'SYSTEM:'
CALL   TTOUT          ;PRINT MSG
MOV    SUC,R3         ;SET SYSTEM# FOR PRINT
CALL   OCTP           ;PRINT IT!
RETURN ;RETURN
```

.SBTTL MODULE U.PRTRV - PRINT DRIVE IDENTIFICATION

PRTRV:

```
MOV    #MSG17,R4     ;SET MSG-> 'DRIVE: '
CALL   TTOUT        ;PRINT MSG
MOV    DRIVE,R3     ;GET DRIVE #
CALL   OCTP         ;PRINT IT!
TST   PTSIDF       ;IF SIDE FLAG
BEQ   XUPTDV       ;SET, THEN
MOV    #MSG20,R4    ;SET MSG-> 'SIDE: '
CALL   TTOUT        ;PRINT MSG
MOV    SIDE,R3      ;GET SIDE #
CALL   OCTP         ;PRINT IT!
XUPTDV: RETURN      ;RETURN
```

```
DRIVE: 0 ;DRIVE # TO PRINT
SIDE: 0 ;SIDE # TO PRINT
PTSIDF: 0 ;SIDE PRINT CONTROL FLAG
```

.SBTTL MODULE U.PRTRKSC - PRINT RACK & SECTOR ERROR IDENT

PRTRKSC:

```
MOV    #MSG22,R4     ;SET MSG-> 'CRC ERROR-'
CALL   TTOUT        ;PRINT MSG
MOV    #MSG15,R4     ;SET MSG-> 'TRACK='
CALL   TTOUT        ;PRINT MSG
MOV    SOTRK(R1),R3 ;GET TRACK #
CALL   OCTP         ;PRINT IT!
MOV    #MSG16,R4     ;SET MSG-> 'SECTOR='
CALL   TTOUT        ;PRINT MSG
MOV    SOSEC(R1),R3 ;GET SECTOR #
CALL   OCTP         ;PRINT IT!
RETURN ;RETURN
```

```
SOTRK: 0 ;SYS#0 TRACK
S1TRK: 0 ;SYS#1 TRACK
SOSEC: 0 ;SYS#0 SECTOR
S1SEC: 0 ;SYS#1 SECTOR
```

```

8144      .SBTTL CHECK BITS SET & NOT SET
8145      ;-----
8146
8147 011314 005037 011524      CKBITS: CLR      BITPAS      ;CLEAR BIT PASS COUNT
8148 011320 012737 012222 011530  MOV      #SETMSG,BITMSG ;SET, SET BITS MSG ADR
8149 011326 013700 011536      MOV      CMPWRD,R0      ;GET COMPARE WORD
8150 011332 013701 011540      MOV      BADWRD,R1     ;GET BAD WORD
8151 011336 040001      BIC      R0,R1         ;R1 = BITS THAT SHOULDN'T BE SET
8152 011340 005100      COM      R0            ;COMPLIMENT COMPARE WORD
8153 011342 053700 011540      BIS      BADWRD,R0     ;SET BITS
8154 011346 005100      COM      R0            ;R0 = BITS THAT SHOULD BE SET
8155 011350 005737 011534      TST      BITOFF       ;IF BIT OFFSET
8156 011354 001411      BEQ      2$           ;NOT=0, THEN
8157 011356 005337 011534      1$: DEC      BITOFF
8158 011362 000241      CLC
8159 011364 006000      ROR      R0
8160 011366 000241      CLC
8161 011370 006001      ROR      R1
8162 011372 005737 011534      TST      BITOFF       ;IF BIT OFFSET
8163 011376 001367      BNE      1$           ;EQUALS 0, THEN
8164 011400 005037 011526      2$: CLR      BITCNT     ;CLEAR BIT COUNTER
8165 011404 032701 000001      3$: BIT      #1,R1
8166 011410 001420      BEQ      4$
8167 011412 012704 015210      MOV      #MSG35,R4     ;SET UP '-' MSG
8168 011416 004737 013232      JSR      PC,TTOUT      ;PRINT MSG
8169 011422 013704 011526      MOV      BITCNT,R4     ;GET BIT COUNT
8170 011426 006304      ASL      R4            ;DOUBLE FOR WORD ADDRESSING
8171 011430 063704 011542      ADD      TABADR,R4     ;ADD TABLE ADDRESS
8172 011434 011404      MOV      (R4),R4       ;SET MSG TO PRINT
8173 011436 004737 013232      JSR      PC,TTOUT      ;PRINT MSG
8174 011442 013704 011530      MOV      BITMSG,R4     ;SET SET BITS MSG
8175 011446 004737 013232      JSR      PC,TTOUT      ;PRINT MSG
8176 011452 005237 011526      4$: INC      BITCNT     ;INCREMENT BIT# COUNTER
8177 011456 000241      CLC
8178 011460 006001      ROR      R1            ;CLEAR CARRY BIT
8179 011462 023737 011532 011526  CMP      BITLIM,BITCNT ;SHIFT NEXT BIT FOR TEST
8180 011470 101345      BHI      3$           ;IF ALL BITS SPECIFIED
8181 011472 005037 011526      CLR      BITCNT       ;DONE, THEN
8182 011476 005737 011524      TST      BITPAS       ;RESET BIT COUNT
8183 011502 001007      BNE      5$
8184 011504 005237 011524      INC      BITPAS       ;SET BITPASS TO GET OUT NEXT PASS
8185 011510 012737 012200 011530  MOV      #NSMSG,BITMSG ;SET NOT SET BITS MSG ADR
8186 011516 010001      MOV      R0,R1         ;GET NOT SET BITS FOR TEST
8187 011520 000731      BR      3$
8188 011522 000207      5$: RTS      PC        ;DO NOT SET BITS
8189      ;RETURN
8190 011524 000000      BITPAS: 0
8191 011526 000000      BITCNT: 0
8192 011530 000000      BITMSG: 0
8193 011532 000000      BITLIM: 0
8194 011534 000000      BITOFF: 0
8195 011536 000000      CMPWRD: 0
8196 011540 000000      BADWRD: 0
8197 011542 000000      TABADR: 0
8198      ;-----

```



```

8257      .SBTTL  ERROR SET SUBROUTINE
8258      ;-----
8259
8260 012240 017637 000000 000402 ERRSET: MOV    @ (SP), $FATAL ;MOVE ERR# TO $FATAL MAILBOX
8261 012246 012737 000001 000400      MOV    #1, $MSGTY ;SET MAIL BOX FLAG
8262 012254 062716 000002      ADD    #2, (SP) ;ADJUST STACK RETURN ADDRESS
8263 012260 000207      ERSETX: RTS    PC ;RETURN
8264      ;-----
8265
8266      .SBTTL  BUS INITIALIZE SUBROUTINE
8267      ;-----
8268
8269 012262 032777 041000 166214 BSINIT: BIT    #41000, @SWR ;IF LOOP
8270 012270 001407      BEQ    1$ ;IS SET, THEN
8271 012272 004737 011042      CALL   TSHDCK ;PRINT TEST HEADER
8272 012276 032777 010000 166200      BIT    #SW12, @SWR ;IF BUS INITIALIZE
8273 012304 001401      BEQ    1$ ;BIT SET, THEN
8274 012306 000005      RESET ;INITIALIZE BUS
8275 012310 005737 012402      1$: TST    _PPRT ;IF LOOP MESSAGE
8276 012314 001026      BNE    5$ ;NOT PRINTED, THEN
8277 012316 032777 001000 166160 2$: BIT    #SW09, @SWR ;IF LOOP ON
8278 012324 001405      BEQ    3$ ;ERROR, THEN
8279 012326 012704 014752      MOV    #MSG28, R4 ;SET LOOP ON ERROR MSG
8280 012332 004737 013232      CALL   TTOUT ;PRINT MSG
8281 012336 000407      BR     4$ ;
8282 012340 005777 166140 3$: TST    @SWR ;IF HALT ON ERROR
8283 012344 100012      BPL    5$ ;IS SET, THEN
8284 012346 012704 015223      MOV    #MSG37, R4 ;SET HALT ON ERROR MSG
8285 012352 004737 013232      CALL   TTOUT ;PRINT MSG
8286 012356 011603 4$: MOV    (SP), R3 ;GET RETURN ADR
8287 012360 062703 000014      ADD    #14, R3 ;ADJ. ADDRESS FOR ERR NO.
8288 012364 011303      MOV    (R3), R3 ;GET ERR NO.
8289 012366 004737 013346      CALL   OCTP ;PRINT ERR #
8290 012372 012737 000001 012402 5$: MOV    #1, LPPRT ;SET LOOP MSG PRINTED FLAG
8291 012400 000207      BSINTX: RTS    PC ;RETURN
8292      ;-----
8293 012402 000000      LPPRT: 0 ;LOOP FLAG
8294      ;-----

```

```
8296      .SBTTL RX02 INTERRUPT HANDLER #0
8297      ;-----
8298
8299 012404 005237 000616  RXINT0: INC      SORDY      ;INCREMENT SYS #0 READY
8300 012410 000002          RTI          ;RETURN TO CALLER
8301      ;-----
8302
8303      .SBTTL RX02 INTERRUPT HANDLER #1
8304      ;-----
8305
8306 012412 005237 000620  RXINT1: INC      S1RDY      ;INCREMENT SYS #1 READY
8307 012416 000002          RTI          ;RETURN TO CALLER
8308      ;-----
8309      .SBTTL TTY INTERRUPT HANDLER
8310      ;-----
8311
8312 012420 004737 013156  TTINT:  CALL      TTIN          ;CALL TTY INPUT ROUTINE
8313 012424 113737 000520 000632  MOVB     T1B,TTITFG        ;SAVE TTY INPUT CHARACTER
8314 012432 022737 000003 000632  CMP      #3,TTITFG        ;IF TTY INTERRUPT
8315 012440 001007          BNE      1$                ;WAS CTRL C, THEN
8316 012442 012704 014715  MOV      #MSG25,R4        ;SET MSG->'^C'
(1) 012446 004737 013232  CALL     TTOUT            ;PRINT MSG
8317 012452 005000          CLR      R0                ;
8318 012454 000137 001000  JMP      START            ;JUMP TO START OVER
8319 012460 052777 000100 165612 1$:  BIS      #BIT6,@TKS      ;SET TTY INTERRUPT BIT
8320 012466 005277 165606  INC      @TKS              ;SET TTY GO BIT
8321 012472 000002          RTI
8322      ;-----
8323
```

```

8325      .SBTTL  BUS ADDRESS TRAP HANDLER
8326      ;-----
8327
8328 012474 005237 000570      TRAP:  INC      BTRPFL      ;SET BUS TRAP FLAG
8329 012500 032777 020000 165776  BIT      #20000,@SWR    ;SEE IF SHOULD PRINT ERRORS
8330 012506 001020                BNE      TRAP2        ;IF NOT, BR
8331 012510 005737 000602      TST      HDRFLG      ;SEE IF DONE HEADER
8332 012514 001006                BNE      TRAP1        ;IF SO, BR
8333 012516 005237 000602      INC      HDRFLG      ;ELSE SET HEADER FLAG
8334 012522 013704 000600      MOV      TMSGAD,R4
8335 012526 004737 013232      JSR      PC,TTOUT    ;PRINT HEADER
8336 012532 012704 014701      TRAP1: MOV      #MSG24,R4
8337 012536 004737 013232      JSR      PC,TTOUT    ;PRINT ERROR
8338 012542 010103                MOV      R1,R3
8339 012544 004737 013346      JSR      PC,OCTP     ;PRINT ADDRESS OF TRAP
8340 012550 004737 012570      TRAP2: JSR      PC,APTER
8341 012554 000055                .WORD   '55          ;BUS TRAP ERROR
8342 012556 005777 165722      TST      @SWR        ;SEE IF HALT ON ERROR
8343 012562 100001                BPL      TRAPX       ;IF NOT, BR
8344 012564 000000                HALT
8345 012566 000002      TRAPX: RTI          ;RETURN FROM INTERRUPT
8346      ;-----
8347
8348      .SBTTL  APT ERROR HANDLER
8349      ;-----
8350
8351 012570 132737 000001 000420  APTER: BITB     #APTENV,$ENV ;IF IN
8352 012576 001407                BEQ     2$           ;APT MODE, THEN
8353 012600 017637 000000 000402  MOV      @(SP),$FATAL ;MOVE ERR # TO $FATAL MAILBOX
8354 012606 012737 000001 000400  MOV      #1,$MSGTY   ;SET MAIL BOX FLAG
8355 012614 000777                BR      1$           ;APT ERROR LOOP
8356 012616 062716 000002      2$:  ADD     #2,(SP)
8357 012622 000207                RTS      PC          ;RETURN
8358      ;-----
8359
8360      .SBTTL  MODULE U.OPRANS - SETUP & GET OPERATOR ANSWERS
8361      ;-----
8362
8363 012624 012737 000116 012646  GETANS: MOV      #'N,ANSDEF ;SET DEFAULT ANSWER-'NO'
8364 012632 012737 000131 012650  MOV      #'Y,ANSCHG ;SET CHANGE ANSWER-'YES'
8365 012640 004737 013036                CALL     TTAR        ;GET ANSWER-CALL TTY ANSWER
8366 012644 000207                RETURN              ;RETURN
8367      ;-----
8368 012646 000000      ANSDEF: 0          ;DEFAULT ANSWER BUFFER
8369 012650 000000      ANSCHG: 0          ;CHANGE ANSWER BUFFER
8370      ;-----
  
```

```

8372
8373      .SBTTL TTY ENTRY SUBROUTINE
8374      ;-----
8375
8376      ;THIS SUBROUTINE IS USED BY THE TEST CONDITION ENTRY ROUTINE TO READ
8377      ;THE RESPONSE ENTERED AT THE TTY AND CHECK THEM FOR LEGALITY AND LIMITS.
8378      ;ALL RESPONSE MUST BE TYPED IN OCTAL (0 - 7) AND MUST FALL WITHIN THE
8379      ;LIMITS SET BY THE CALLING ROUTINE. IF AN ENTRY IS ILLEGAL OR OUTSIDE
8380      ;THE LIMITS, A QUESTION MARK (?) IS TYPED AND THE RESPONSE MAY BE RE-
8381      ;ENTERED. ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND MAY BE TERMIN-
8382      ;ATED AT LESS THAN SIX BY TYPING A CARRIAGE RETURN.
8383
8384      012652 005037 000562      TTR:   CLR      TEMP1      ;CLEAR FIRST CHARACTER FLAG
8385      012656 005000              CLR      RO
8386      012660 004737 013156      TTR0:  JSR      PC,TTIN    ;GO READ CHARACTER
8387      012664 122737 000015 000526      CMPB    #15,TIB      ;SEE IF CR
8388      012672 001005              BNE     TTR1         ;IF NOT, BR
8389      012674 005737 000562              TST     TEMP1       ;SEE IF FIRST CHARACTER
8390      012700 001446              BEQ     TTR5         ;IF SO, BR
8391      012702 000137 012774              JMP     TTR2         ;ELSE, GO LOAD VALUE
8392      012706 122737 000060 000526      TTR1:  CMPB    #60,TIB    ;SEE IF CHAR IS LESS THAN 0
8393      012714 101402              BLOS   TTR1A        ;IF NOT, BR
8394      012716 000137 013020              JMP     TTR1A        ;ELSE, GO TO ERROR
8395      012722 122737 000070 000526      TTR1A: CMPB    #70,TIB    ;SEE IF CHAR IS GREATER THAN 7
8396      012730 101002              BHI     TTR1B        ;IF NOT, BR
8397      012732 000137 013020              JMP     TTR1B        ;ELSE, GO TO ERROR
8398      012736 005237 000562      TTR1B: INC     TEMP1     ;SET FIRST CHARACTER FLAG
8399      012742 000241              CLC
8400      012744 006100              ROL     RO
8401      012746 000241              CLC
8402      012750 006100              ROL     RO           ;SHIFT 3 LEFT
8403      012752 000241              CLC
8404      012754 006100              ROL     RO
8405      012756 042737 177770 000526      BIC     #177770,TIB  ;STRIP ASCII
8406      012764 053700 000526      BIS     TIB,RO       ;LOAD CHARACTER
8407      012770 005301              DEC     R1           ;SEE IF DONE
8408      012772 001332              BNE     TTR0         ;IF NOT, BR
8409      012774 020002              TTR2:  CMP     R0,R2    ;SEE IF EXCEEDED MAXIMUM LIMIT
8410      012776 101402              BLOS   TTR3         ;IF NOT, BR
8411      013000 000137 013020              JMP     TTR3         ;ELSE, GO TO ERROR
8412      013004 020300              TTR3:  CMP     R3,R0    ;SEE IF BELOW MINIMUM LIMIT
8413      013006 101402              BLOS   TTR4         ;IF NOT, BR
8414      013010 000137 013020              JMP     TTR4         ;ELSE, GO TO ERROR
8415      013014 010015      TTR4:  MOV     R0,(R5)  ;LOAD VALUE
8416      013016 000207      TTR5:  RTS     PC       ;EXIT
  
```

```

8417      ;-----
8418
8419      .SBTTL TTY ENTRY ERROR SUBROUTINE
8420      ;-----
8421
8422
8423      013020 012704 014202      TINER: MOV     #MSG7,R4
8424      013024 004737 013232              JSR     PC,TTOUT    ;PRINT?
8425      013030 1627'6 000020              SUB     #20,(SP)    ;RESET SP TO START OF VALUE ROUTINE
8426      013034 000207              RTS     PC           ;REDO VALUE ENTRY
8427
  
```

8428
 8429
 8430
 8431
 8432
 8433
 8434
 8435
 8436
 8437 013036 105037 000530
 8438 013042 005000
 8439 013044 004737 013156
 8440 013050 122737 000032 000526
 8441 013056 001003
 8442 013060 005237 000576
 8443 013064 000433
 8444 013066 122737 000015 000526
 8445 013074 001004
 8446 013076 113737 012646 000530
 8447 013104 000423
 8448 013106 042737 000040 000526
 8449 013114 123737 012650 000526
 8450 013122 001411
 8451 013124 123737 012646 000526
 8452 013132 001405
 8453 013134 012704 014202
 8454 013140 004737 013232
 8455 013144 000734
 8456 013146 113737 000526 000530
 8457 013154 000207
 8458
 8459
 8460
 8461
 8462
 8463 013156 005077 165116
 8464 013162 005037 000526
 8465 013166 005277 165106
 8466 013172 105777 165102
 8467 013176 100375
 8468 013200 017737 165076 000526
 8469 013206 105777 165072
 8470 013212 100375
 8471 013214 113777 000526 165064
 8472 013222 042737 000200 000526
 8473 013230 000207
 8474

.SBTTL TTY ANSWER ENTRY SUBROUTINE

 : THIS SUBROUTINE IS USED BY THE TEST CONDITION ENTRY ROUTINE TO
 : READ THE RESPONSE ENTERED AT THE TTY AND CHECK THEM FOR LEGALITY.
 : ALL RESPONSES MUST BE EQUAL TO RETURN THE DEFAULT ANSWER, CHANGED ANSWER
 : OR A CARRIAGE RETURN, IF ENTRY IS ILLEGAL A QUESTION MARK IS TYPED AND
 : THE RESPONSE MAY BE REENTERED.

```

TTAR:  CLR      ANSWER      ;CLEAR ANSWER
        CLR      RO
        CALL     TTIN        ;GO READ A CHARACTER
        CMPB    #32,TIB      ;IF 'L'
        BNE     TTAR1        ;THEN
        INC     STFLG        ;SET START PROGRAM
        BR      XTTAR        ;BR TO END
TTAR1:  CMPB    #15,TIB      ;IF 'CR'
        BNE     TTAR2        ;THEN
        MOVB   ANSDEF,ANSWER ;SET DEFAULT ANSWER
        BR      XTTAR        ;BR TO END
TTAR2:  BIC     #40,TIB      ;CLEAR LOWER CASE BIT
        CMPB   ANSCHG,TIB    ;IF ANSWER NOT OK WITH CHANGE ANS
        BEQ    TTAR3        ;THEN
        CMPB   ANSDEF,TIB    ;IF ANSWER NOT OK WITH DEFAULT ANS
        BEQ    TTAR3        ;THEN
        MOV    #MSG7,R4     ;SETUP MSG '?'
        CALL   TTOUT        ;PRINT MSG
        BR     TTAR         ;START OVER
TTAR3:  MOVB   TIB,ANSWER    ;SAVE ANSWER
XTTAR:  RETURN              ;RETURN
  
```

.SBTTL TTY READ SUBROUTINE

```

TTIN:   CLR     @TKS
        CLR     TIB
        INC     @TKS
TTIN1:  TSTB   @TKS
        BPL    TTIN1
        MOV    @TKB,TIB
TTIN2:  TSTB   @TPS
        BPL    TTIN2
        MOVB   TIB,@TPB
        BIC    #200,TIB      ;STRIP OFF TOP BIT OF BYTE
        RTS    PC
  
```



```

8476
8477
8478
8479 013232 112437 000524
8480 013236 122737 000043 000524
8481 013244 001432
8482 013246 122737 000045 000524
8483 013254 001403
8484 013256 004737 013700
8485 013262 000763
8486 013264 112737 000015 000524
8487 013272 004737 013700
8488 013276 012703 000001
8489 013302 005037 000524
8490 013306 004737 013700
8491 013312 005303
8492 013314 001372
8493 013316 112737 000012 000524
8494 013324 004737 013700
8495 013330 000740
8496 013332 000207
8497

```

```

.SBTTL TTY ASCII OUTPUT SUBROUTINE
-----
TTOUT:  MOVB  (R4)+,TOB
        CMPB  #43,TOB      ;IF TOB= '#'
        BEQ   TCEX
        CMPB  #45,TOB      ;IF TOB= '%'
        BEQ   TCRLF
        JSR   PC,TOG
        BR    TTOUT
TCRLF:  MOVB  #15,TOB      ;SET TOB= 'CR'
        JSR   PC,TOG
        MOV   #1,R3
TCRLFA: CLR   TOB
        JSR   PC,TOG
        DEC   R3
        BNE   TCRLFA      ;DO FILLERS
        MOVB  #12,TOB      ;SET TOB= 'LF'
        JSR   PC,TOG
        BR    TTOUT
TCEX:   RTS   PC          ;RETURN
-----

```

```

8499          .SBTTL OCTAL OUTPUT SUBROUTINE
8500          ;-----
8501
8502 013334 012737 000001 013570 OCTPE: MOV #1,OFL
8503 013342 010304          MOV R3,R4
8504 013344 000410          BR OCTP0
8505 013346 005037 013570 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
8506 013352 010304          OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
8507 013354 001004          BNE OCTP0 ;IF NOT ZERO, BR
8508 013356 004737 013550 JSR PC,OCTPG1 ;ELSE PRINT ZERO
8509 013362 000137 013512 JMP OCTP3 ;SPACE AND EXIT
8510 013366 032704 100000 OCTP0: BIT #100000,R4 ;SEE IF MSG=1
8511 013372 001406          BEQ OCTP1 ;IF NOT, BR
8512 013374 012704 000001 MOV #1,R4
8513 013400 004737 013526 JSR PC,OCTPG ;PRINT 1
8514 013404 000137 013416 JMP OCTP2
8515 013410 005004          OCTP1: CLR R4
8516 013412 004737 013526 JSR PC,OCTPG ;PRINT 0
8517 013416 010304          OCTP2: MOV R3,R4
8518 013420 006004          ROR R4
8519 013422 006004          ROR R4
8520 013424 006004          ROR R4
8521 013426 006004          ROR R4 ;POSITION DIGIT
8522 013430 000304          SWAB R4
8523 013432 004737 013526 JSR PC,OCTPG ;PRINT DIGIT 2
8524 013436 010304          MOV R3,R4
8525 013440 006004          ROR R4
8526 013442 000304          SWAB R4
8527 013444 004737 013526 JSR PC,OCTPG ;PRINT DIGIT 3
8528 013450 010304          MOV R3,R4
8529 013452 006104          ROL R4
8530 013454 006104          ROL R4
8531 013456 000304          SWAB R4
8532 013460 004737 013526 JSR PC,OCTPG ;PRINT DIGIT 4
8533 013464 010304          MOV R3,R4
8534 013466 006004          ROR R4
8535 013470 006004          ROR R4
8536 013472 006004          ROR R4
8537 013474 004737 013526 JSR PC,OCTPG ;PRINT DIGIT 5
8538 013500 005237 013526 INC OFL ;SET FLAG TO PRINT LSD
8539 013504 010304          MOV R3,R4
8540 013506 004737 013526 JSR PC,OCTPG ;PRINT DIGIT 6
8541 013512 012737 000240 000524 OCTP3: MOV #240,TOB ;PRINT SPACE
8542 013520 004737 013700 JSR PC,TOG ;EXIT
8543 013524 000207          OCTPX: RTS PC ;EXIT
8544          ;-----
  
```

8546
8547 013526 042704 177770
8548 013532 001004
8549 013534 005737 013570
8550 013540 001001
8551 013542 000207
8552 013544 005237 013570
8553 013550 052704 000260
8554 013554 010437 000524
8555 013560 004737 013700
8556 013564 010304
8557 013566 000207
8558
8559 013570 000000
8560

OCTPG: BIC #177770,R4
BNE OCTPG0
TST OFL
BNE OCTPG0
RTS PC
OCTPG0: INC OFL
OCTPG1: BIS #260,R4
MOV R4,TOB
JSR PC,TOG
MOV R3,R4
OCTPGX: RTS PC

OFL: 0 ;FIRST CHAR FLAG

```

8562      .SBTTL DATA CHARACTER OUTPUT SUBROUTINE
8563      ;-----
8564
8565
8566      013572  J05037  000524      DOUT:  CLR      TOB
8567      013576  012704  000010      MOV      #10,R4      ;SET NUMBER TO PRINT
8568      013602  110337  000524      MOV      R3,TOB
8569      013606  105777  164472      DOUT1:  TSTB     @TPS
8570      013612  100375      BPL      DOUT1
8571      013614  132737  000200  000524      BITB     #200,TOB
8572      013622  001404      BEQ      DOUT2
8573      013624  012777  000061  164454      MOV      #061,@TPB
8574      013632  000403      BR       DOUT3
8575      013634  012777  000060  164444      DOUT2:  MOV      #060,@TPB
8576      013642  006137  000524      DOUT3:  ROL      TOB
8577      013646  005304      DEC      R4
8578      013650  001356      BNE      DOUT1
8579      013652  000207      RTS      PC
8580      ;-----
8581      013654  013703  000566      DOUTD:  MOV      TEMP3,R3
8582      013660  000303      SWAB     R3
8583      013662  004737  013572      JSR      PC,DOUT
8584      013666  013703  000566      MOV      TEMP3,R3
8585      013672  004737  013572      JSR      PC,DOUT
8586      013676  000207      RTS      PC
8587      ;-----
8588
8589      .SBTTL TTY OUTPUT
8590      ;-----
8591      013700  132737  000040  000421      TOG:    BITB     #APTCSP,$ENVM ;SEE IF CONSOLE SUPPRESS MODE
8592      013706  001014      BNE      TEX          ;IF SO, BR
8593      013710  105777  164370      TSTB     @TPS
8594      013714  100371      BPL      TOG
8595      013716  113777  000524  164362      MOV      TOB,@TPB
8596      013724  005737  000532      TST      TTWAIT      ;IS TTY WAIT FLAG SET
8597      013730  001403      BEQ      TEX          ;THEN
8598      013732  105777  164346      1$:     TSTB     @TPS      ;WAIT FOR CHARACTER
8599      013736  100375      BPL      1$          ;TO BE PRINTED
8600      013740  000207      TEX:    RTS      PC
8601      ;-----
    
```

| Line No | Address | Offset | Length | Value | Label | Text |
|---------|---------|--------|--------|--------|--------|--|
| 8603 | | | | | .SBTTL | MESSAGE TABLE |
| 8604 | | | | | | |
| 8605 | | | | | | |
| 8606 | 013742 | 020045 | 020040 | 054122 | MSG1: | .ASCII /% RXCS RXDB CMD% #/ |
| 8607 | 014002 | 051445 | 043117 | 053524 | MSG2: | .ASCII /%SOFTWARE SWITCH REG (OCT) = #/ |
| 8608 | 014040 | 022445 | 055103 | 054122 | MSG3: | .ASCII /%CZRXEAO RX02 FMTR PROG%#/ |
| 8609 | 014072 | 020045 | 020040 | 020040 | MSG4: | .ASCII /% RXCS ADDRESS (OCT) = #/ |
| 8610 | 014127 | 045 | 020040 | 020040 | MSG5: | .ASCII /% VECTOR (OCT) = #/ |
| 8611 | 014164 | 042445 | 042116 | 047440 | MSG6: | .ASCII /%END OF PASS #/ |
| 8612 | 014202 | 037440 | 021440 | | MSG7: | .ASCII / ? #/ |
| 8613 | 014206 | 047045 | 020117 | 054523 | MSG8: | .ASCII /%NO SYSTEM AVAILABLE TO FORMAT #/ |
| 8614 | 014246 | 022445 | 047506 | 046522 | MSG9: | .ASCII /%FORMAT COMPLETED #/ |
| 8615 | 014272 | 020045 | 044440 | 052116 | MSG10: | .ASCII /% INTERRUPT DID NOT OCCUR #/ |
| 8616 | 014326 | 052445 | 044516 | 020124 | MSG11: | .ASCII /%UNIT NOT RESPONDING TO ADDRESS #/ |
| 8617 | 014367 | 045 | 042101 | 051104 | MSG12: | .ASCII /%ADDRESSING TEST #/ |
| 8618 | 014411 | 045 | 020040 | 054105 | MSG13: | .ASCII /% EXPECT RCVD ACTUAL% #/ |
| 8619 | 014452 | 022445 | 047506 | 046522 | MSG14: | .ASCII /%FORMAT DONE ON FOLLOWING #/ |
| 8620 | 014506 | 020040 | 051124 | 041501 | MSG15: | .ASCII / TRACK (OCT) =#/ |
| 8621 | 014526 | 020040 | 042523 | 052103 | MSG16: | .ASCII / SECTOR (OCT) =#/ |
| 8622 | 014547 | 104 | 044522 | 042526 | MSG17: | .ASCII /DRIVE: #/ |
| 8623 | 014556 | 044523 | 042504 | 021472 | MSG20: | .ASCII /SIDE: #/ |
| 8624 | 014564 | 020040 | 021440 | | MSG21: | .ASCII / #/ |
| 8625 | 014570 | 020045 | 051103 | 020103 | MSG22: | .ASCII /% CRC ERROR -> #/ |
| 8626 | 014610 | 020045 | 044124 | 051511 | MSG23: | .ASCII /% THIS SYSTEM NOT CAPABLE OF DOUBLE DENSITY OPERATIONS #/ |
| 8627 | 014701 | 045 | 052502 | 020123 | MSG24: | .ASCII /%BUS TRAP: #/ |
| 8628 | 014715 | 045 | 041536 | 021440 | MSG25: | .ASCII /% ^C #/ |
| 8629 | 014722 | 020045 | 054122 | 051503 | MSG26: | .ASCII /% RXCS ERR #/ |
| 8630 | 014736 | 020045 | 054122 | 051505 | MSG27: | .ASCII /% RXES ERR #/ |
| 8631 | 014752 | 020040 | 046055 | 047517 | MSG28: | .ASCII / -LOOPING ON ERR: #/ |
| 8632 | 014776 | 042445 | 051122 | 051117 | MSG29: | .ASCII /%ERROR BIT SET AFTER INIT #/ |
| 8633 | 015031 | 045 | 042040 | 047117 | MSG30: | .ASCII /% DONE BIT NOT SET #/ |
| 8634 | 015055 | 045 | 052040 | 020122 | MSG31: | .ASCII /% TR BIT NOT SET #/ |
| 8635 | 015077 | 045 | 020040 | 020040 | MSG32: | .ASCII /% = #/ |
| 8636 | 015121 | 040 | 026440 | 020076 | MSG33: | .ASCII / -> DROPPED #/ |
| 8637 | 015140 | 026445 | 051076 | 046505 | MSG34: | .ASCII /% ->REMOVE XXDP MEDIA FROM THIS SYSTEM.!#/ |
| 8638 | 015210 | 020045 | 020040 | 020055 | MSG35: | .ASCII /% - #/ |
| 8639 | 015217 | 045 | 037040 | 043 | MSG36: | .ASCII /% > #/ |
| 8640 | 015223 | 040 | 026440 | 040510 | MSG37: | .ASCII / -HALT ON ERR: #/ |
| 8641 | 015244 | 042445 | 052116 | 051105 | MSG38: | .ASCII /%ENTER CONDITIONS IN OCTAL #/ |
| 8642 | 015300 | 052045 | 044510 | 020123 | MSG39: | .ASCII /%THIS PROGRAM FORMATS DISKETTES TO SINGLE OR DOUBLE(DEFAULT) DENSITY/ |
| 8643 | 015404 | 047445 | 020116 | 047502 | MSG40: | .ASCII /%ON BOTH DRIVES OF A FLOPPY DISK SUBSYSTEM CAPABLE OF DOUBLE DENSITY/ |
| 8644 | 015510 | 047445 | 042520 | 040522 | | .ASCII /%OPERATIONS. A CRC VERIFY IS PERFORMED ON TRACKS: 0 & 76 (DEFAULT) OR/ |
| 8645 | 015615 | 045 | 046101 | 020114 | | .ASCII /%ALL TRACKS OF THE FORMATTED DISKETTE, IF SELECTED.%#/ |
| 8646 | 015702 | 052055 | 046511 | 020105 | MSG41: | .ASCII /-TIME OUT ERROR #/ |
| 8647 | 015723 | 045 | 021440 | | MSG42: | .ASCII /% #/ |
| 8648 | 015726 | 044045 | 046105 | 037520 | MSG43: | .ASCII /%HELP? (Y OR N) N #/ |
| 8649 | 015752 | 004445 | 047111 | 052111 | MSG44: | .ASCII /% INITIALIZE NOT DONE - RUN DIAGNOSTICS UNLESS OBVIOUS ERR#/ |
| 8650 | 016045 | 045 | 042523 | 020124 | MSG45: | .ASCII /%SET DISKETTE TO SINGLE DENSITY? (Y OR N) N #/ |
| 8651 | 016125 | 045 | 042526 | 044522 | MSG46: | .ASCII /%VERIFY DISKETTE CRC (ALL TRACKS)? (Y OR N) N #/ |
| 8652 | 016206 | 043045 | 047514 | 050120 | MSG47: | .ASCII /%FLOPPY DISK SYSTEM: 0 ADDRESS CHANGE? (Y OR N) N #/ |
| 8653 | 016271 | 045 | 051511 | 040440 | MSG48: | .ASCII /%IS ANOTHER FLOPPY DISK SYSTEM AVAILABLE? (Y OR N) N #/ |
| 8654 | 016357 | 045 | 047516 | 051440 | MSG49: | .ASCII /%NO SYSTEM AVAILABLE TO FORMATTER #/ |
| 8655 | 016422 | 004445 | 011 | | MSG50: | .ASCII /% / |
| 8656 | 016425 | 045 | 054523 | 052123 | MSG51: | .ASCII /%SYSTEM: #/ |
| 8657 | 016436 | 047045 | 020117 | 047504 | MSG52: | .ASCII /%NO DONE BIT AFTER INITIALIZE #/ |
| 8658 | 016475 | 040 | 037055 | 051104 | MSG53: | .ASCII / ->DROPPED FROM FORMATTER%#/ |

8659 016530 004445 041501 046040 MSG54: .ASCII /% AC LOW ERROR - IS FLOPPY DISK SYSTEM POWERED UP?#/

8660 016613 045 042011 044522 MSG55: .ASCII /% DRIVE NOT READY - IS DISKETTE IN DRIVE & DOOR CLOSED?#/

8661 016703 045 047504 054440 MSG56: .ASCII /%DO YOU WANT TO FORMAT MORE DISKETTES? (Y OR N) N #/

8662 016766 051045 046505 053117 MSG57: .ASCII /%REMOVE FORMATTED DISKETTES & INSERT DISKETTES TO BE FORMATTED#/

8663 017065 045 026455 020076 MSG58: .ASCII /%--> TYPE 'CR' WHEN READY#/

8664 017117 045 047506 046522 MSG59: .ASCII /%FORMATTER DONE-RESTART MONITOR OR UPDATE PROGRAM-->TYPE CTRL C TO/

8665 017221 045 052123 051101 .ASCII /%START THIS PROGRAM AGAIN #/

: .SBTTL TEST HEADERS
:-----

8671 017254 022445 037055 042522 MSGCD3: .ASCII /%%-->READ SECTOR COMMAND ERR #/

8672 017311 045 026445 051476 MSGCD4: .ASCII /%%-->SET DENSITY COMMAND ERR #/

8673 017346 022445 037055 042522 MSGCD5: .ASCII /%%-->READ MAINTENANCE STATUS COMMAND ERR #/

8674 017417 045 026445 037055 MSGLP: .ASCII /%%-->LOOP MODULE-ERR #/

8675 017446 .EVEN

: .SBTTL ERROR CODE STORAGE
:-----

8681 017446 000 XER: .BYTE 0 ;ERR CODE - EXTENDED

8682 017447 000 .BYTE 0 ;

8683 017450 000 .BYTE 0 ;

8684 017451 000 .BYTE 0 ;

8685 017452 000 .BYTE 0 ;

8686 017453 000 .BYTE 0 ;

8687 017454 000 .BYTE 0 ;

8688 017455 000 .BYTE 0 ;

: .SBTTL PATCH AREA
:-----

8696 017456 000000 PATCH: 0

8697 017660 017660 .+.200

8698 017660 000000 LASTAD: 0 ;LAST ADDRESS NOT TO EXCEED 17776

8700
8701 000001 .END

| | | | | | | | |
|----------------|-------|-------|-------|-------|-------|-------|-------|
| AUNIT = 000000 | 6128 | | | | | | |
| AUSWR = 000000 | 6128 | | | | | | |
| AVECT1= 000264 | 6126# | 6128 | | | | | |
| AVECT2= 000000 | 6128 | | | | | | |
| BADWRD 011540 | 8005* | 8023 | 8046* | 8064 | 8150 | 8153 | 8196# |
| BAEMPT 005120 | 7102 | 7146# | | | | | |
| BAFILL 005116 | 7100 | 7145# | | | | | |
| BA11 001770 | 6395# | 6425 | | | | | |
| BA231 007076 | 7696# | 7722 | | | | | |
| BB30 007742 | 7882# | 7884 | | | | | |
| BC13 002470 | 6503# | 6505 | | | | | |
| BC20 002622 | 6584# | 6612 | | | | | |
| BC22 003332 | 6771# | 6802 | | | | | |
| BC231 007126 | 7702# | 7719 | | | | | |
| BDAUDL 010142 | 7944# | 7952 | | | | | |
| BDBUDL 010146 | 7945# | 7948 | | | | | |
| BE00 001306 | 6234 | 6236# | 6266 | | | | |
| BF00 001316 | 6238# | 6240 | | | | | |
| BITCNT 011526 | 8164* | 8169 | 8176* | 8179 | 8181* | 8191# | |
| BITLIM 011532 | 8009* | 8050* | 8179 | 8193# | | | |
| BITMSG 011530 | 8148* | 8174 | 8185* | 8192# | | | |
| BITOFF 011534 | 8008* | 8049* | 8155 | 8157* | 8162 | 8194# | |
| BITPAS 011524 | 8147* | 8182 | 8184* | 8190# | | | |
| BIT0 = 000001 | 5994 | 6042# | | | | | |
| BIT00 = 000001 | 6042# | | | | | | |
| BIT01 = 000002 | 6042# | | | | | | |
| BIT02 = 000004 | 6042# | | | | | | |
| BIT03 = 000010 | 6042# | | | | | | |
| BIT04 = 000020 | 6042# | | | | | | |
| BIT05 = 000040 | 6042# | | | | | | |
| BIT06 = 000100 | 6042# | | | | | | |
| BIT07 = 000200 | 6042# | | | | | | |
| BIT08 = 000400 | 6042# | | | | | | |
| BIT09 = 001000 | 6042# | | | | | | |
| BIT1 = 000002 | 6042# | | | | | | |
| BIT10 = 002000 | 6042# | | | | | | |
| BIT11 = 004000 | 6042# | | | | | | |
| BIT12 = 010000 | 6042# | | | | | | |
| BIT13 = 020000 | 6042# | 6447 | 7966 | | | | |
| BIT14 = 040000 | 6042# | | | | | | |
| BIT15 = 100000 | 6042# | | | | | | |
| BIT2 = 000004 | 6042# | | | | | | |
| BIT3 = 000010 | 6042# | | | | | | |
| BIT4 = 000020 | 5992 | 6042# | | | | | |
| BIT5 = 000040 | 5993 | 6042# | | | | | |
| BIT6 = 000100 | 6042# | 6214 | 7889 | 8319 | | | |
| BIT7 = 000200 | 6042# | | | | | | |
| BIT8 = 000400 | 6005 | 6042# | 6231 | | | | |
| BIT9 = 001000 | 6042# | | | | | | |
| BPTVEC= 000014 | 6042# | | | | | | |
| BSINIT 012262 | 6452 | 7540 | 7544 | 8269# | | | |
| BSINTX 012400 | 8291# | | | | | | |
| BTRP 000320 | 6118# | 6215* | 6221* | 6437* | 6453* | | |
| BTRPFL 000570 | 6165# | 6192* | 6421* | 6424 | 6445 | 8328* | |
| BTRP2 000322 | 6119# | 6216* | 6222* | 6438* | 6454* | | |
| CKBITS 011314 | 8029 | 8070 | 8147# | | | | |

| | | | | | | | | | | | | | | | | | | | |
|---------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|--|--|--|--|--|
| CKSYAD | 002422 | 6358* | 6368* | 6495# | | | | | | | | | | | | | | | |
| CMD | 000542 | 6154# | 6774 | 6780 | 6784 | 6794 | 6803 | 6805 | 6861* | 6862 | 6878* | 6879 | 6887* | 7055 | | | | | |
| | | 7066 | 7077 | 7535* | 7536 | 7991 | | | | | | | | | | | | | |
| CMDTBL | 003772 | 6861 | 6878 | 6899# | | | | | | | | | | | | | | | |
| CMPWRD | 011536 | 8006* | 8047* | 8149 | 8195# | | | | | | | | | | | | | | |
| CMSGTB | 011544 | 8007 | 8201# | | | | | | | | | | | | | | | | |
| CMSG10 | 011704 | 8207 | 8233# | | | | | | | | | | | | | | | | |
| CMSG11 | 011713 | 8208 | 8234# | | | | | | | | | | | | | | | | |
| CMSG12 | 011750 | 8209 | 8210 | 8235# | | | | | | | | | | | | | | | |
| CMSG15 | 011770 | 8212 | 8236# | | | | | | | | | | | | | | | | |
| CMSG4 | 011624 | 8201 | 8227# | | | | | | | | | | | | | | | | |
| CMSG5 | 011634 | 8202 | 8228# | | | | | | | | | | | | | | | | |
| CMSG6 | 011641 | 8203 | 8229# | | | | | | | | | | | | | | | | |
| CMSG7 | 011662 | 8204 | 8230# | | | | | | | | | | | | | | | | |
| CMSG8 | 011665 | 8205 | 8231# | | | | | | | | | | | | | | | | |
| CMSG9 | 011675 | 8206 | 8232# | | | | | | | | | | | | | | | | |
| CR | = 000015 | 6042# | | | | | | | | | | | | | | | | | |
| CRCALL | = 000200 | 6001# | 6345 | 6347 | 6979 | | | | | | | | | | | | | | |
| CRCERR | = 000001 | 5994# | 7268 | | | | | | | | | | | | | | | | |
| CRLF | = 000200 | 6042# | | | | | | | | | | | | | | | | | |
| CSR | 000534 | 6151# | | | | | | | | | | | | | | | | | |
| CSRADR | 010230 | 6667* | 7153* | 7945 | 7961# | | | | | | | | | | | | | | |
| CSRCHK | 010404 | 7232* | 7248* | 7262* | 7537* | 8001# | | | | | | | | | | | | | |
| CSRCKX | 010616 | 8031 | 8035# | | | | | | | | | | | | | | | | |
| CSRCMP | 000546 | 6156# | 7227* | 7228* | 7505* | 7506* | 7509* | 8003 | 8006 | 8019 | | | | | | | | | |
| CSRMSK | 000544 | 6155# | 7229* | 7510* | 8002 | | | | | | | | | | | | | | |
| DBGFLG | 000516 | 6088* | 6144# | 6190* | 6370 | 7529 | | | | | | | | | | | | | |
| DBLDEN | = 000400 | 6005# | 6233 | | | | | | | | | | | | | | | | |
| DC20 | 002762 | 6611# | | | | | | | | | | | | | | | | | |
| DDISP | = 177570 | 6042# | | | | | | | | | | | | | | | | | |
| DELAY | 010136 | 6670* | 7155 | 7943# | | | | | | | | | | | | | | | |
| DENBIT | = 000400 | 5985# | 6337 | 6339 | 7517 | | | | | | | | | | | | | | |
| DENERR | = 000020 | 5992# | | | | | | | | | | | | | | | | | |
| DENS | 000506 | 6140# | 6233* | 6235* | 6887 | | | | | | | | | | | | | | |
| DNBIT | = 000040 | 5983# | 6668 | 6672 | 7086 | 7227 | 7505 | 7970 | | | | | | | | | | | |
| DOUT | 013572 | 8566# | 8583 | 8585 | | | | | | | | | | | | | | | |
| DOUTD | 013654 | 8581# | | | | | | | | | | | | | | | | | |
| DOUT1 | 013606 | 8569# | 8570 | 8578 | | | | | | | | | | | | | | | |
| DOUT2 | 013634 | 8572 | 8575# | | | | | | | | | | | | | | | | |
| DOUT3 | 013642 | 8574 | 8576# | | | | | | | | | | | | | | | | |
| DRIVE | 011224 | 7345* | 7353* | 7356* | 7363* | 7371* | 7374* | 7440* | 7450* | 7710* | 7711* | 8092* | 8094* | 8112 | | | | | |
| | | 8121# | | | | | | | | | | | | | | | | | |
| DRIVE1 | = 000400 | 5991# | 7516 | | | | | | | | | | | | | | | | |
| DROPAV | 006264 | 7432* | 7441* | 7451* | 7455 | 7462 | 7467# | | | | | | | | | | | | |
| DROPDV | 006036 | 7335* | 7346* | 7349* | 7364* | 7367* | 7375 | 7379 | 7383# | | | | | | | | | | |
| DRVAVL | 006040 | 7242* | 7432# | | | | | | | | | | | | | | | | |
| DRVCNT | 007260 | 7697* | 7710 | 7712 | 7716* | 7718 | 7725# | | | | | | | | | | | | |
| DRV DEN | = 000040 | 5993# | 7519 | | | | | | | | | | | | | | | | |
| DRVDRP | 005530 | 7239* | 7255* | 7272* | 7335# | | | | | | | | | | | | | | |
| DRV PTR | 007262 | 7696* | 7703 | 7717* | 7726# | | | | | | | | | | | | | | |
| DRV R | 004402 | 6810* | 7064# | | | | | | | | | | | | | | | | |
| DRV RLY | = 000200 | 5987# | 7436 | 7446 | | | | | | | | | | | | | | | |
| DRV1 | = 000020 | 5996# | 6780 | 7338 | 7434 | 7514 | 8090 | | | | | | | | | | | | |
| DSWR | = 177570 | 6042# | | | | | | | | | | | | | | | | | |
| DVSCDN | 004366 | 6972* | 6973* | 6981 | 6945* | 7001* | 7032# | | | | | | | | | | | | |
| DVSEC | 004372 | 6976* | 6977* | 6994* | 6997* | 6998 | 6999 | 7002* | 7007* | 7018* | 7019 | 7034# | | | | | | | |

| | | | | | | | | | | |
|---------|--------|-------|-------|-------|-------|-------|------|-------|-------|-------|
| DVTKDN | 004362 | 6970* | 6971* | 6983 | 6985* | 6993* | 7010 | 7012* | 7017* | 7030# |
| DVTRK | 00437c | 6974* | 6975* | 6986* | 6989 | 6990* | 6991 | 7008 | 7036# | |
| EA00 | 00125o | 6194 | 6213 | 6230# | | | | | | |
| EA10 | 001520 | 6329 | 6332# | | | | | | | |
| EA13 | 002510 | 6497 | 6499 | 6507# | | | | | | |
| EA21 | 003246 | 6678 | 6692 | 6696# | | | | | | |
| EA221 | 003746 | 6866 | 6872 | 6881 | 6884 | 6887# | | | | |
| EA223 | 004450 | 7071 | 7073# | | | | | | | |
| EA225 | 005522 | 7241 | 7243 | 7253 | 7257 | 7259 | 7267 | 7274# | | |
| EA232 | 007330 | 7752 | 7754# | | | | | | | |
| EA233 | 007466 | 7784 | 7792 | 7794 | 7800# | | | | | |
| EB10 | 001562 | 6338 | 6340# | | | | | | | |
| EB11 | 002136 | 6400 | 6421# | | | | | | | |
| EB20 | 002774 | 6583 | 6614# | | | | | | | |
| EB2253 | 006374 | 7518 | 7520# | | | | | | | |
| EB231 | 007242 | 7700 | 7720# | | | | | | | |
| EC00 | 001076 | 6203 | 6205# | | | | | | | |
| EC10 | 001624 | 6346 | 6348# | | | | | | | |
| EC222 | 004202 | 6988 | 6995# | | | | | | | |
| EC23 | 007056 | 7624 | 7638 | 7640 | 7645 | 7647 | 7654 | 7656# | | |
| ED00 | 001212 | 6218 | 6221# | | | | | | | |
| ED10 | 001662 | 6355 | 6357# | | | | | | | |
| ED21 | 003210 | 6688 | 6690# | | | | | | | |
| ED222 | 004174 | 6992 | 6994# | | | | | | | |
| ED231 | 007222 | 7704 | 7716# | | | | | | | |
| EE221 | 003726 | 6875 | 6883# | | | | | | | |
| EE222 | 004274 | 7003 | 7005 | 7008# | | | | | | |
| EE231 | 007162 | 7708 | 7710# | | | | | | | |
| EF20 | 002752 | 6598 | 6600 | 6607 | 6609# | | | | | |
| EG00 | 001350 | 6244 | 6246# | | | | | | | |
| EG222 | 004342 | 7015 | 7018# | | | | | | | |
| EG225 | 005504 | 7269 | 7271# | | | | | | | |
| EH2231 | 005000 | 7106 | 7121# | | | | | | | |
| EH2252 | 006260 | 7456 | 7464# | | | | | | | |
| E100 | 001444 | 6262 | 6264# | | | | | | | |
| EK2231 | 004646 | 7101 | 7103# | | | | | | | |
| EMSGTB | 011574 | 8048 | 8214# | | | | | | | |
| EMSG0 | 011777 | 8214 | 8238# | | | | | | | |
| EMSG1 | 012003 | 8215 | 8239# | | | | | | | |
| EMSG10 | 012134 | 8224 | 8248# | | | | | | | |
| EMSG11 | 012156 | 8225 | 8249# | | | | | | | |
| EMSG2 | 012014 | 8216 | 8240# | | | | | | | |
| EMSG3 | 012026 | 8217 | 8241# | | | | | | | |
| EMSG4 | 012035 | 8218 | 8242# | | | | | | | |
| EMSG5 | 012051 | 8219 | 8243# | | | | | | | |
| EMSG6 | 012066 | 8220 | 8244# | | | | | | | |
| EMSG7 | 012103 | 8221 | 8245# | | | | | | | |
| EMSG8 | 012115 | 8222 | 8246# | | | | | | | |
| EMSG9 | 012125 | 8223 | 8247# | | | | | | | |
| EMTVEC= | 000030 | 6042# | | | | | | | | |
| EM22 | 003550 | 6808 | 6810# | | | | | | | |
| ENDFTB | 000634 | 6183# | 6237 | | | | | | | |
| ENDTSH | 011130 | 8084 | 8097# | | | | | | | |
| END00 | 001452 | 6266# | | | | | | | | |
| END223 | 005100 | 7121 | 7126 | 7130 | 7135# | | | | | |
| ERRBIT= | 100000 | 5995# | 6679 | | | | | | | |

| | | | | | | | | | |
|--------|--------|-------|-------|-------|-------|------|------|------|-------|
| IC222 | 004122 | 6983# | | | | | | | |
| IC2231 | 004670 | 7091 | 7107# | | | | | | |
| IC225 | 005334 | 7225 | 7244# | | | | | | |
| IC2251 | 005564 | 7342# | | | | | | | |
| IC2252 | 006070 | 7438# | | | | | | | |
| IC23 | 006650 | 7615 | 7617 | 1620# | | | | | |
| IC233 | 007420 | 7786 | 7793# | | | | | | |
| ID00 | 001170 | 6217# | | | | | | | |
| ID10 | 001644 | 6352# | | | | | | | |
| ID20 | 002624 | 6579 | 6581 | 6585# | | | | | |
| ID21 | 003170 | 6685 | 6687# | | | | | | |
| ID22 | 003350 | 6774# | | | | | | | |
| ID221 | 003716 | 6880# | | | | | | | |
| ID222 | 004162 | 6991# | | | | | | | |
| ID2231 | 004700 | 7109# | | | | | | | |
| ID225 | 005374 | 7252# | | | | | | | |
| ID2251 | 005632 | 7350# | | | | | | | |
| ID2252 | 006114 | 7437 | 7439 | 7442# | | | | | |
| ID23 | 006674 | 7621 | 7625# | | | | | | |
| ID231 | 007130 | 7703# | | | | | | | |
| IE10 | 001714 | 6363# | | | | | | | |
| IE20 | 002646 | 6590# | | | | | | | |
| IE21 | 003216 | 6680 | 6691# | | | | | | |
| IE22 | 003374 | 6780# | | | | | | | |
| IE221 | 003664 | 6874# | | | | | | | |
| IE222 | 004224 | 6999# | | | | | | | |
| IE2231 | 005032 | 7108 | 7127# | | | | | | |
| IE225 | 005462 | 7266# | | | | | | | |
| IE2251 | 005672 | 7339 | 7358# | | | | | | |
| IE2252 | 006134 | 7435 | 7446# | | | | | | |
| IE23 | 006704 | 7627# | | | | | | | |
| IE231 | 007140 | 7705# | | | | | | | |
| IF10 | 001752 | 6364 | 6370# | | | | | | |
| IF20 | 002676 | 6586 | 6588 | 6595 | 6597# | | | | |
| IF21 | 003254 | 6697# | | | | | | | |
| IF22 | 003404 | 6782# | | | | | | | |
| IF222 | 004250 | 7000 | 7004# | | | | | | |
| IF225 | 005422 | 7245 | 7258# | | | | | | |
| IF2251 | 005702 | 7360# | | | | | | | |
| IF2252 | 006144 | 7448# | | | | | | | |
| IF23 | 006712 | 7629# | | | | | | | |
| IG00 | 001334 | 6243# | | | | | | | |
| IG20 | 002722 | 6602# | | | | | | | |
| IG22 | 003414 | 6784# | | | | | | | |
| IG222 | 004304 | 6980 | 7010# | | | | | | |
| IG225 | 005470 | 7268# | | | | | | | |
| IG2251 | 005746 | 7368# | | | | | | | |
| IG2252 | 006166 | 7447 | 7449 | 7452# | | | | | |
| IG23 | 006722 | 7631# | | | | | | | |
| IH00 | 001406 | 6253# | | | | | | | |
| IH20 | 002634 | 6587# | | | | | | | |
| IH22 | 003424 | 6786# | | | | | | | |
| IH2231 | 004562 | 7092# | | | | | | | |
| IH2251 | 006000 | 7343 | 7348 | 7355 | 7357 | 7361 | 7366 | 7373 | 7375# |
| IH2252 | 006204 | 7443 | 7445 | 7453 | 7455# | | | | |
| IH23 | 007020 | 7626 | 7648# | | | | | | |

| | | | | | | | |
|---------|--------|-------|-------|-------|-------|------|-------|
| I100 | 001432 | 6254 | 6261# | | | | |
| I120 | 002706 | 6599# | | | | | |
| I122 | 003450 | 6781 | 6792# | | | | |
| IJ00 | 001262 | 6231# | | | | | |
| IJ22 | 003460 | 6794# | | | | | |
| IK22 | 003470 | 6796# | | | | | |
| IK2231 | 004620 | 7098# | | | | | |
| IL22 | 003520 | 6803# | | | | | |
| IM22 | 003536 | 6807# | | | | | |
| INITCK | 003002 | 6572* | 6577* | 6663# | | | |
| INITDN= | 000004 | 5989# | 6687 | | | | |
| INITDP | 003270 | 6663* | 6674* | 6681* | 6693* | 6697 | 6702# |
| INITPG | 000514 | 6143# | 6241* | 6609* | 6763 | 6855 | |
| INITTS | 004360 | 6763* | 6966 | 6969* | 7027# | | |
| INTLV = | 000002 | 5997 | 7024# | | | | |
| IOTVEC= | 000020 | 6042# | | | | | |
| LASTAD | 017660 | 8698# | | | | | |
| LA221 | 003636 | 6864 | 6867# | | | | |
| LA223 | 004442 | 7069 | 7072# | | | | |
| LA232 | 007322 | 7750 | 7753# | | | | |
| LA30 | 007766 | 7879 | 7887# | | | | |
| LB00 | 001146 | 6198 | 6214# | | | | |
| LB10 | 001554 | 6336 | 6339# | | | | |
| LB221 | 003734 | 6869 | 6885# | | | | |
| LB222 | 004210 | 6982 | 6997# | | | | |
| LB225 | 005326 | 7237 | 7242# | | | | |
| LB2251 | 005624 | 7341 | 7349# | | | | |
| LC00 | 001070 | 6200 | 6204# | | | | |
| LC10 | 001616 | 6344 | 6347# | | | | |
| LC222 | 004150 | 6984 | 6989# | | | | |
| LD10 | 001660 | 6353 | 6356# | | | | |
| LD22 | 003370 | 6775 | 6779# | | | | |
| LD2231 | 005002 | 7110 | 7122# | | | | |
| LD2251 | 005662 | 7351 | 7356# | | | | |
| LE20 | 002672 | 6591 | 6596# | | | | |
| LE2231 | 005052 | 7128 | 7131# | | | | |
| LE2251 | 005740 | 7359 | 7367# | | | | |
| LE23 | 007012 | 7628 | 7646# | | | | |
| LE231 | 007156 | 7706 | 7709# | | | | |
| LF = | 000012 | 6042# | | | | | |
| LF22 | 003442 | 6783 | 6790# | | | | |
| LF23 | 006766 | 7630 | 7641# | | | | |
| LG20 | 002746 | 6603 | 6608# | | | | |
| LG222 | 004330 | 7011 | 7016# | | | | |
| LG2251 | 005774 | 7369 | 7374# | | | | |
| LG23 | 006760 | 7632 | 7639# | | | | |
| LH2231 | 004710 | 7093 | 7111# | | | | |
| LH23 | 007052 | 7649 | 7655# | | | | |
| LI22 | 003506 | 6793 | 6800# | | | | |
| LJ00 | 001302 | 6232 | 6235# | | | | |
| LK2231 | 004640 | 7099 | 7102# | | | | |
| LOOP | 006406 | 7240* | 7256* | 7273* | 7528# | | |
| LOOP1 | 006446 | 7536# | 7540 | 7544 | | | |
| LOOP2 | 006520 | 7539 | 7541# | | | | |
| LOOP3 | 006564 | 7543 | 7545# | | | | |
| LPPRT | 012402 | 8275 | 8290* | 8293# | | | |

| | | | | | | | | | |
|--------|--------|-------|-------|-------|-------|------|------|-------|--|
| LSIFLG | 000520 | 6145# | 6201* | 6219* | 7913 | | | | |
| LSITRP | 001200 | 6215 | 6219# | | | | | | |
| MSGCD3 | 017254 | 7260 | 8671# | | | | | | |
| MSGCD4 | 017311 | 7246 | 8672# | | | | | | |
| MSGCD5 | 017346 | 7226 | 8673# | | | | | | |
| MSGLP | 017417 | 7534 | 8674# | | | | | | |
| MSG1 | 013742 | 7981 | 8606# | | | | | | |
| MSG10 | 014272 | 7789 | 7797 | 8615# | | | | | |
| MSG11 | 014326 | 6449 | 8616# | | | | | | |
| MSG12 | 014367 | 6436 | 8617# | | | | | | |
| MSG13 | 014411 | 8017 | 8058 | 8618# | | | | | |
| MSG14 | 014452 | 7694 | 8619# | | | | | | |
| MSG15 | 014506 | 8130 | 8620# | | | | | | |
| MSG16 | 014526 | 8133 | 8621# | | | | | | |
| MSG17 | 014547 | 8111 | 8622# | | | | | | |
| MSG2 | 014002 | 6462 | 8607# | | | | | | |
| MSG20 | 014556 | 8116 | 8623# | | | | | | |
| MSG21 | 014564 | 7985 | 7989 | 8021 | 8025 | 8062 | 8066 | 8624# | |
| MSG22 | 014570 | 8129 | 8625# | | | | | | |
| MSG23 | 014610 | 6695 | 8626# | | | | | | |
| MSG24 | 014701 | 8336 | 8627# | | | | | | |
| MSG25 | 014715 | 8316 | 8628# | | | | | | |
| MSG26 | 014722 | 8013 | 8629# | | | | | | |
| MSG27 | 014736 | 8054 | 8630# | | | | | | |
| MSG28 | 014752 | 8279 | 8631# | | | | | | |
| MSG29 | 014776 | 6683 | 8632# | | | | | | |
| MSG3 | 014040 | 6195 | 8608# | | | | | | |
| MSG30 | 015031 | 7972 | 8633# | | | | | | |
| MSG31 | 015055 | 7975 | 8634# | | | | | | |
| MSG32 | 015077 | 8635# | | | | | | | |
| MSG33 | 015121 | 7380 | 7463 | 8636# | | | | | |
| MSG34 | 015140 | 6501 | 8637# | | | | | | |
| MSG35 | 015210 | 8167 | 8638# | | | | | | |
| MSG36 | 015217 | 8639# | | | | | | | |
| MSG37 | 015223 | 8284 | 8640# | | | | | | |
| MSG38 | 015244 | 8641# | | | | | | | |
| MSG39 | 015300 | 6330 | 8642# | | | | | | |
| MSG4 | 014072 | 6402 | 8609# | | | | | | |
| MSG40 | 015404 | 8643# | | | | | | | |
| MSG41 | 015702 | 7790 | 7798 | 7977 | 8646# | | | | |
| MSG42 | 015723 | 8647# | | | | | | | |
| MSG43 | 015726 | 6325 | 8648# | | | | | | |
| MSG44 | 015752 | 6689 | 8649# | | | | | | |
| MSG45 | 016045 | 6332 | 8650# | | | | | | |
| MSG46 | 016125 | 6340 | 8651# | | | | | | |
| MSG47 | 016206 | 6349 | 8652# | | | | | | |
| MSG48 | 016271 | 6360 | 8653# | | | | | | |
| MSG49 | 016357 | 6582 | 8654# | | | | | | |
| MSG5 | 014127 | 6411 | 8610# | | | | | | |
| MSG50 | 016422 | 8655# | | | | | | | |
| MSG51 | 016425 | 7745 | 8102 | 8656# | | | | | |
| MSG52 | 016436 | 6676 | 8657# | | | | | | |
| MSG53 | 016475 | 7754 | 8658# | | | | | | |
| MSG54 | 016530 | 6686 | 8659# | | | | | | |
| MSG55 | 016613 | 7458 | 8660# | | | | | | |
| MSG56 | 016703 | 7876 | 8661# | | | | | | |

.SRDOC 3723#
.\$READ 3328#
.\$R2AZ 4858#
.\$SAVE 3889#
.\$SB2D 4675#
.\$SB2O 4776#
.\$SLOP 2397#
.\$SIZE 4271#
.\$SUPR 4814#
.\$TRAP 3991#
.\$TYPB 3221#
.\$TYPD 3144#
.\$TYPE 2925#
.\$TYPO 3048#
.\$4OCA 944#
.1170 498#

. ABS. 017662 000

ERRORS DETECTED: 0

CZR XEA,CZR XEA/CRF=CZR XEA.SML,CZR XEA.P11
RUN-TIME: 16 14 1 SECONDS
RUN-TIME RATIO: 38/31 1.2
CORE USED: 35K (70 PAGES)