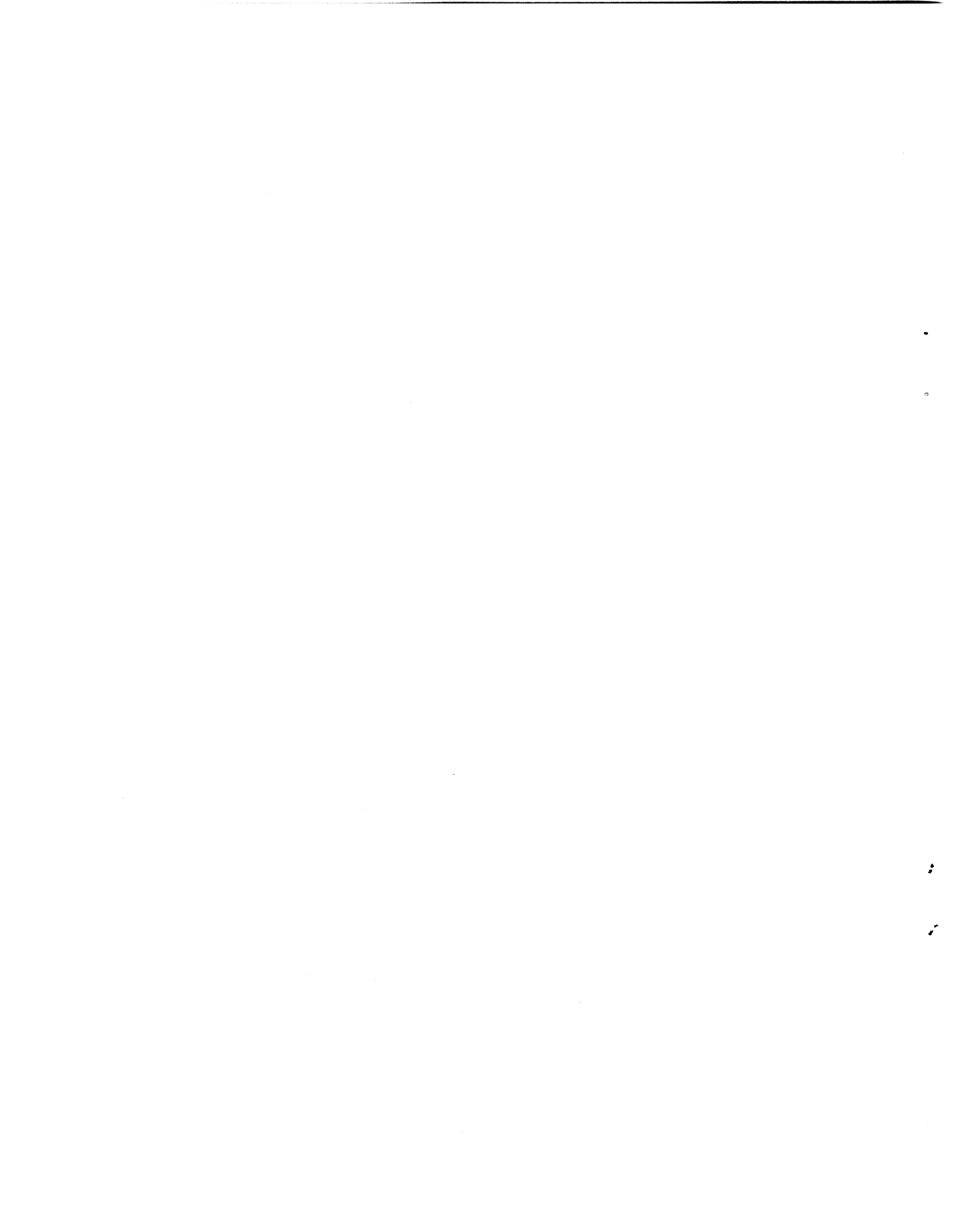


**LOTUS 700**  
**DISC CONTROLLER**  
**DIAGNOSTICS LISTINGS**

**POINT**   
**DATA CORPORATION**





```
          0          .LOC  0
0          0          0          ; PC
1          0          0          ; INTERRUPT VECTOR
2          2003       JMP @.+1      ; RESTART VECTOR
3          436        START
```

```
          50          .LOC  50
```

;\*\*\*\* CONSTANTS \*\*\*\*

```
50          2 K2:      2
51          3 K3:      3
52          4 K4:      4
53          5 K5:      5
54          7 K7:      7
55          10 K10:    10
56          17 K17:    17
57          20 K20:    20
60          27 K27:    27
61          37 K37:    37
62          70 K70:    70
63 177761 N17:      -17
64 177400 N400:     -400
65          177 RMSK:  177      ; RIGHT BYTE MASK
66          377 BMSK:  377      ; BYTE MASK
67 177700 DMSK:  177700      ; DEVICE CODE MASK
70 177000 MSKA:  177000
71          1777 CMAX:  1777
72          15 CR:     15      ; CARRIAGE RETURN
73          12 LF:     12      ; LINEFEED
74          54 COMMA:  54      ; COMMA
75          177 RUBD:  177      ; RUBOUT
76          40 SPACE:  40
77          60 ZERO:   60
100         67 SEVN:   67
101         71 NINE:   71
102        1740 SMSK:  1740      ; SECT ADDR FIELD MASK
103        76000 HMSK: 76000      ; HEAD ADDR FIELD MASK
104 176000 FIXED: 176000      ; FIXED SURFACE ADDRESS OFFSET
105 74000 CATT: 74000      ; CLEAR DRIVE ATTN BITS
106 174000 CATTs: 174000      ; CLEAR RW AND DRIVE ATTN BITS
107         40 ASECT:  40      ; A SECT INCREMENT
```

;\*\*\*\* ADDRESS POINTERS \*\*\*\*

```
110        3222 CBUF:  CBUF      ; INPUT CHARACTER BUFFER
111         450 ABORT:  IDC      ; RESTART VECTOR
112         400 INSTL:  INST      ; IO INSTRUCTION LIST
113        3242 TEMPA:  TEMP      ; SCRATCH PAD ADDRESS
114        3003 NTST:  NTS      ; NEXT TEST CONNECTOR
115        30462 WBUF:  WBUF      ; WRITE BUFFER
116        50462 RBUF:  RBUF      ; READ BUFFER
```

;\*\*\*\* DRIVE TYPE AND TEST PARAMETERS \*\*\*\*

```
117         0 DTBL:   0          ; DRIVE LOOKUP TABLE
120         0 DSKT:   0          ; DISK TYPE: 0=ALL REMOVABLE, -1=CART+FIXED
121         0 DRVF:   0          ; DRIVE NUMBER FIELD
```

```

122      0 LSFN: 0      ; LAST SURFACE NUMBER
123      0 STRK: 0      ; SECTORS PER TRACK
124      0 LCYN: 0      ; LAST CYLINDER NUMBER
125      0 DRVN: 0      ; DRIVE NUMBER
126      0 SFCN: 0      ; SURFACE NUMBER
127      0 CYLN: 0      ; CYLINDER NUMBER
130      0 SCTN: 0      ; SECTOR NUMBER
131      0 SCNT: 0      ; SECTOR COUNT
132      0 CSTA: 0      ; CONTROLLER STATUS
133      0 DSTA: 0      ; DRIVE STATUS
134      0 ATTS: 0      ; DRV ATTEN STATUS BIT

```

; \*\*\*\* VARIABLES AND TEMPORARY STORAGE \*\*\*\*

```

135      0 MFLG: 0      ; TEST MODE FLAG
136      0 TSTX: 0      ; TEST NAME TEXT ADDR
137      0 TSTV: 0      ; TEST RESTART ADDR
140      0 NEXT: 0      ; NEXT TEST ADDR
141      0 ACN:  0      ; AC DATA SAVE
142      0 SAVP: 0
143      0 PFLG: 0      ; PROGRAM FLAG
144      0 CLOK: 0      ; TIME CLOCK
145      0 TIMD: 0      ; TIMEOUT FLAG
146      0 PASS: 0      ; WR PASS COUNT
147      0 PCH:  0
150      0 LCNT: 0
151      0 DVCD: 0      ; DEVICE CODE
152      0 CNTR: 0      ; COUNTER
153      0 WBFP: 0      ; WBUFF WORD POINTER
154      0 RBFP: 0      ; RBUFF WORD POINTER

```

; \*\*\*\* GENERAL PURPOSE SUBROUTINES \*\*\*\*

```

155      2075 PRINT: PRNT ; PRINT TEXT STRING
156      2140 CHAR: ACHR  ; PRINT SINGLE CHARACTER
157      2154 INPUT: INPT ; COMMAND STRING INPUTTER
160      2314 MATCH: MTCH ; INPUT STRING MATCH
161      2146 EQM:  QM    ; INPUT ERROR RESPONSE
162      2366 ABIN: ABN   ; DECIMAL ASCII-BINARY CONVERSION
163      2473 OBIN: OBN   ; OCTAL ASCII-BINARY CONVERSION
164      2573 DECN: DEC   ; PRINT DECIMAL NUMBER
165      2644 OCTN: OCT   ; PRINT OCTAL NUMBER IN 6 DIGITS
166      1017 ERROR: ERR  ; TEST ERROR STOP, PRINTOUT AND LOOP
167      15573 DRCMD: DRC ; DRIVE COMMAND
170      15451 RWCMD: RWC ; DISK RW DRIVER
171      15670 NRDY: NRD  ; PRINT DRIVE NOT READY
172      16001 HDRG: HDG  ; HEADER GENERATOR
173      16114 HCMP: HCM  ; HEADER COMPARE
174      16175 DCM:  DCM  ; DATA COMPARE
175      2663 TIMER: TIME ; 1 MS TIME CLOCK
176      16155 PRIME: PRIM ; BUFFER PRIMER
177      2753 SET:  SETT  ; TEST STEP INITIALIZE
200      2276 SADC: SAD  ; OCTAL-ASCII CONVERSION, 2 DIGIT
201      15705 CKCK: CKK  ; CHECK CONTROL STATUS, AFTER SEEK
202      15703 CKCS: CKC  ; CHECK CONTROL STATUS, NO SEEK
203      15715 CKDS: CKD  ; CHECK DRIVE STATUS
204      15724 CKDA: CKAD ; CHECK DISK ADDRESS
205      15741 DADD: DAD  ; SET SURF-SECTOR ADDR & SECT CNT
206      16217 SKCMD: SCMD ; EMULATOR SEEK COMMAND

```

207 16262 EMCMD:ECMD ; EMULATOR WR COMMAND  
210 16317 CRBUF:CRBF ; RBUF VS REFERENCE COMPARISON

;\*\*\*\* VARIABLE DEVICE CODE IO INSTRUCTIONS \*\*\*\*

211	61000	DDOA: DOA	0,0
212	40275	STA	0,DDAO
213	1400	JMP	0,3
214	61100	DDOAS: DOAS	0,0
215	212	JMP	DDOA+1
216	61300	DDOAP: DOAP	0,0
217	212	JMP	DDOA+1
220	62000	DDOB: DOB	0,0
221	40276	STA	0,DOBO
222	1400	JMP	0,3
223	62100	DDOBS: DOBS	0,0
224	221	JMP	DDOB+1
225	63000	DDOC: DOC	0,0
226	40277	STA	0,DOCO
227	1400	JMP	0,3
230	63100	DDOCS: DOCS	0,0
231	226	JMP	DDOC+1
232	63300	DDOCP: DDCP	0,0
233	226	JMP	DDOC+1
234	60400	DDIA: DIA	0,0
235	40300	STA	0,DIAO
236	1400	JMP	0,3
237	60400	DDIA1: DIA	0,0
240	40301	STA	0,DIA1
241	1400	JMP	0,3
242	61400	DDIB: DIB	0,0
243	40302	STA	0,DIBO
244	1400	JMP	0,3
245	61400	DDIB1: DIB	0,0
246	40303	STA	0,DIB1
247	1400	JMP	0,3
250	62400	DDIC: DIC	0,0
251	40304	STA	0,DICO
252	1400	JMP	0,3
253	60100	NNIOS: NIOS	0
254	1400	JMP	0,3
255	60300	NNIOP: NIOP	0
256	1400	JMP	0,3
257	60200	NNIOC: NIOC	0
260	1400	JMP	0,3
261	63600	SKDN: SKPDN	0
262	1400	JMP	0,3
263	1401	JMP	1,3
264	63700	SKDZ: SKPDZ	0
265	1400	JMP	0,3
266	1401	JMP	1,3
267	63500	SKBZ: SKPBZ	0
270	1400	JMP	0,3
271	1401	JMP	1,3
272	63400	SKBN: SKPBN	0
273	1400	JMP	0,3
274	1401	JMP	1,3

;\*\*\*\* DISK CONTROLLER REGISTER IO DATA SAVE \*\*\*\*

275 0 DOAO: 0  
276 0 DOBO: 0  
277 0 DOCO: 0  
300 0 DIAO: 0  
301 0 DIA1: 0  
302 0 DIBO: 0  
303 0 DIB1: 0  
304 0 DICO: 0  
305 0 DIAN: 0  
306 0 DIBN: 0  
307 0 DICN: 0

400 . LOC 400

; \*\*\*\* LIST OF VARIABLE DEVICE CODE IO INSTRUCTIONS \*\*\*\*

400	211	INST:	DDOA
401	214		DDOAS
402	216		DDOAP
403	220		DDOB
404	223		DDOBS
405	225		DDOC
406	230		DDOCS
407	232		DDOCP
410	234		DDIA
411	237		DDIA1
412	242		DDIB
413	245		DDIB1
414	250		DDIC
415	253		NNIOS
416	255		NNIOP
417	257		NNIOC
420	261		SKDN
421	264		SKDZ
422	267		SKBZ
423	272		SKBN
424	4630		SRB1-1
425	4631		SRB1
426	4722		RBNOB
427	3727		CFS0+4
430	3730		CFS0+5
431	5026		RBN3A
432	5240		RBZ1A-1
433	5241		RBZ1A
434	15640		DRC1C
435	35	INSN:	.-INST ; INSTRUCTION COUNT



;\*\*\*\* PROGRAM INITIALIZATION \*\*\*\*

;PRINT PROGRAM IDENTIFICATION AND INITIALIZE

```
436 6155 START: JSR @PRINT ;PRINT TITLE
437 17612      TTL0
440 6155      JSR @PRINT ;PRINT VERSION AND DATE
441 17636      TTL1
442 102400     SUB 0,0
443 40143     STA 0,PFLG ;RESET PROGRAM FLAG
444 40135     STA 0,MFLG ;CLEAR MODE FLAG
445 40117     STA 0,DTBL ;CLEAR LOOKUP TABLE POINTER
446 40134     STA 0,ATTS ;RESET ATT BIT
447 40151     STA 0,DVCD ;CLEAR DEV CODE
```

;REQUEST ENTRY OF DEVICE CODE OTHER THAN STANDARD 27

```
450 6155 IDC: JSR @PRINT
451 17652     DEVC
452 6157      JSR @INPUT ;GET DEVICE CODE
453 465       IDC2 ;DEFAULT TO 27
454 6163      JSR @OBIN ;CONVERT TO BINARY
455 3242      TEMP
456 460       IDC1 ;NOT OCTAL EXIT
457 412       JMP IDC2A ;GOT IT
460 6160 IDC1: JSR @MATCH ;SEE IF A COMMAND
461 3162      CMDSO ;COMMAND LIST
462 1000      JMP 0,2 ;A COMMAND
463 6161 IDC1A: JSR @EGM ;INPUT ERROR, PRINT '?'
464 452       IDC+2
```

;SAVE DEVICE CODE AND ATTACH IT TO EACH IO INSTRUCTION

```
465 20060 IDC2: LDA 0,K27
466 24151     LDA 1,DVCD
467 125004    MOV 1,1,SZR ;OLD DEV CODE?
470 415       JMP PROG ;YES
471 40151 IDC2A: STA 0,DVCD ;SAVE DEV CODE
472 24743     LDA 1,INSN
473 44150     STA 1,LCNT ;SET INSTRUCTION COUNT
474 24067     LDA 1,DMSK ;MASK
475 34112     LDA 3,INSTL ;INSTRUCTION LIST
476 33400 IDC2B: LDA 2,@0,3 ;GET INSTRUCTION
477 133400    AND 1,2 ;STRIP OLD DEV CODE
500 113000    ADD 0,2 ;ATTACH NEW
501 53400     STA 2,@0,3 ;RETURN IT
502 175400    INC 3,3
503 14150     DSZ LCNT ;DONE?
504 772       JMP IDC2B ;NOT YET
```

;REQUEST ENTRY IF PROGRAM SELECTION: LOGIC, DRIVE OR TESTER

```
505 6155 PROG: JSR @PRINT ;"TEST MODE"
506 17661     TMOD
507 6157      JSR @INPUT ;GET ANSWER
510 517       PRG1 ;DEFAULT, GO CHECK SETTING
511 6160      JSR @MATCH ;SEARCH DIRECTORY
512 3064      PDIR ;PROGRAM DIRECTORY
513 1000      JMP 0,2 ;FOUND IT
```

```

514 6160 JSR @MATCH ; NOT A NAME, CHECK COMMAND
515 3166 CMDS1 ; COMMAND LIST
516 1000 JMP 0,2 ; A COMMAND
517 24143 PRG1: LDA 1,PFLG ; CHECK PROG FLAG
520 125004 MOV 1,1,SZR ; SET PREVIOUSLY?
521 411 JMP PRG3-2 ; YES, NO CHANGE
522 6161 JSR @EGM ; INPUT ERROR
523 507 PRG+2 ; EXIT VECTOR

```

```

; COME HERE TO SET PROGRAM FLAG AFTER A SELECTION HAS BEEN MADE.
; THE VALUE OF FLAG IS AS FOLLOWS:

```

```

; 1 = TEST LOGIC ONLY
; 2 = TEST WITH DRIVE
; 3 = TEST WITH TESTER

```

```

524 126520 PRG2: SUBZL 1,1
525 404 JMP PRG2B+1
526 24050 PRG2A: LDA 1,K2
527 402 JMP PRG2B+1
530 24051 PRG2B: LDA 1,K3
531 44143 STA 1,PFLG ; SET PROGRAM FLAG

```

```

; REQUEST ENTRY OF STOP OPTION AND SET MODE FLAG AS FOLLOWS:

```

```

; 1 = STOP AFTER ERROR PRINTOUT
; 2 = LOOP ON ERROR
; 4 = LOOP ON TEST

```

```

; BIT 0 OF MFLG IS USED FOR PRINT TEST NAME INDICATOR:
; 0 = PRINT, 1 = NO PRINT

```

```

532 6155 JSR @PRINT ; "STOP ON ERROR?"
533 17667 STPE
534 6157 PRG3: JSR @INPUT ; GET ANSWER
535 545 PRG3C ; DEFAULT, NO CHANGE
536 6160 JSR @MATCH ; CHECK ANSWER
537 3213 NYL ; COMMAND LIST
540 1000 JMP 0,2 ; GO SET MODE FLAG

```

```

541 126520 PRG3A: SUBZL 1,1
542 402 JMP .+2
543 126400 PRG3B: SUB 1,1
544 44135 STA 1,MFLG ; SET MODE FLAG

```

```

; INDEX TO TEST MODE AS INDICATED IN FLAG

```

```

545 20143 PRG3C: LDA 0,PFLG
546 34403 LDA 3,PROGL ; PROGRAM ADDR LIST
547 117000 ADD 0,3 ; INDEX
550 3400 JMP @0,3 ; GO THERE

```

```

551 552 PROGL: .+1
552 0 0
553 557 LOGIC
554 601 DRIVE
555 750 TESTR

```

; LOGIC TEST

; BEFORE STARTING THE TEST, CHECK TO SEE IF ANY DISK DRIVE IS ONLINE  
; TO THE CONTROLLER, PRINT DISCONNECT MESSAGE IF SO. OTHERWISE PROCEED

```
556 10000          DRDY
557 102400 LOGIC: SUB 0,0
560 4211          JSR DDOA      ; SELECT DRIVE
561 4242          JSR DDIB      ; READ DRIVE STATUS
562 24774         LDA 1, LOGIC-1
563 107405        AND 0,1, SNR   ; DRIVE READY?
564 407           JMP LTST      ; NO, START TEST
565 6155          JSR @PRINT    ; YES, PRINT MESSAGE
566 17677         DRONL
567 63610         SKPDN TTI     ; WAIT FOR START SIGNAL
570 777           JMP .-1
571 60510         DIAS 0, TTI   ; OK TO GO
572 767           JMP LOGIC+2   ; GO CHECK IT AGAIN

573 6155 LTST: JSR @PRINT    ; NEW LINE
574 20076         CRLF
575 126400        SUB 1,1
576 44146         STA 1, PASS   ; RESET PASS COUNT
577 2401          JMP @. +1
600 3300         RGRS
```

; DRIVE TEST

; REQUEST INPUT OF DRIVE NAME OR CODE NUMBER

```
601 6155 DRIVE: JSR @PRINT
602 17733         DNAME
603 6157          JSR @INPUT    ; GET DRIVE NAME
604 647           DNAMO ; DEFAULT
605 22110         LDA 0, @CBUF  ; GET 1ST INPUT WORD
606 101300        MOVS 0,0      ; WANT 1ST BYTE
607 24065         LDA 1, RMSK
610 123400        AND 1,0      ; PICK IT OFF
611 24077         LDA 1, ZERO
612 34101         LDA 3, NINE
613 162033        ADCZ# 3,0, SNC ; SKIP IF >9
614 106032        ADCZ# 0,1, SZC ; SKIP IF =>0
615 423           JMP DNAM     ; NOT A DECIMAL, MAY BE NAME
```

; FIRST CHARACTER IS A DECIMAL, CONVERT IT AND LOCATE LOOKUP TABLE

```
616 6162          JSR @ABIN    ; CALL ASCII-BINARY CONVERTER
617 3242          TEMP
620 652           DNAMO+3; ILLEGAL NUMBER
621 22113         LDA 0, @TEMPA ; GET RESULT
622 24414         LDA 1, DLIM   ; RANGE LIMIT
623 122432        SUBZ# 1,0, SZC ; OK?
624 417           JMP DNAM+3   ; OVER
625 30412         LDA 2, DLST   ; 1ST TABLE ADDR
626 103120        ADDZL 0,0     ; NUMBER X4
627 24052         LDA 1, K4
630 122400        SUB 1,0      ; OFFSET BY 4
631 113000        ADD 0,2      ; TABLE ADDR
632 25002         LDA 1,2,2    ; CHECK SECT WORD
633 125005        MOV 1,1, SNR ; A SECT COUNT?
634 407           JMP DNAM+3   ; NO, A DUMMY
```

```

635      417      JMP      DNAM1      ; YES, GO SET IT

636      117 DLIM: 117
637      16720 DLST: DT1

```

; FIRST CHARACTER IS NON-DECIMAL, SEARCH DIRECTORY FOR NAME

```

640      6160 DNAM: JSR      @MATCH      ; GO LOOK IT UP
641      16342      DDIR      ; DRIVE DIRECTORY
642      412      JMP      DNAM1      ; FOUND IT
643      6160      JSR      @MATCH      ; SEE IF A COMMAND
644      3204      CMD53      ; COMMAND LIST
645      1000      JMP      0,2      ; EXIT TO COMMAND
646      404      JMP      .+4      ; NOT FOUND

647      20117 DNAMO: LDA      0,DTBL      ; CHECK DRIVE POINTER
650      101004      MOV      0,0,SZR      ; SET?
651      414      JMP      DNUM-2      ; YES, EXIT
652      6161      JSR      @EGM      ; INPUT ERROR
653      603      DRIVE+2; RETRY

```

; HAS DRIVE LOOKUP TABLE, COPY PARAMETERS OVER

```

654      50117 DNAM1: STA      2,DTBL      ; SAVE TABLE ADDR
655      25000      LDA      1,0,2
656      44120      STA      1,DSKT      ; SET DRIVE TYPE
657      25001      LDA      1,1,2
660      44122      STA      1,LSFN      ; SET LAST SURF #
661      25002      LDA      1,2,2
662      44123      STA      1,STRK      ; SET SECT PER TRACK
663      25003      LDA      1,3,2
664      44124      STA      1,LCYN      ; SET LAST CYLINDER #

```

; GET DRIVE NUMBER INPUT

```

665      6155      JSR      @PRINT
666      17742      DNUMT
667      6157 DNUM: JSR      @INPUT
670      702      DNUA      ; DEFAULT
671      6163      JSR      @OBIN      ; CONVERT TO BINARY
672      3242      TEMP
673      700      DNUM1      ; NOT OCTAL
674      26113      LDA      1,@TEMPA      ; GET NUMBER
675      20051      LDA      0,K3
676      122432      SUBZ#      1,0,SZC      ; SKIP IF >3
677      407      JMP      DNUM2      ; OK
700      6161 DNUM1: JSR      @EGM      ; INPUT ERROR
701      667      DNUM

```

; CHECK DRIVE NUMBER SETTING FOR DEFAULT

```

702      24134 DNUA: LDA      1,ATTS
703      125004      MOV      1,1,SZR      ; SET?
704      420      JMP      DNUM4      ; YES

```

; POSITION DRIVE NUMBER FOR DOA, SET CORRESPONDING ATTENTION STATUS  
; AND CLEAR BITS

```

705      126400      SUB      1,1

```

```

706 135000 DNUM2: MOV 1,3 ; SAVE IN AC3
707 127120 ADDZL 1,1 ; SHIFT INTO FIELD POSITION
710 127120 ADDZL 1,1
711 125120 MOVZL 1,1
712 44121 STA 1,DRVF ; SAVE IT
713 20433 LDA 0,BIT1
714 175005 MOV 3,3,SNR ; DRIVE 0?
715 405 JMP DNUM3 ; YES
716 174400 NEG 3,3 ; NO, MAKE COUNT
717 101220 MOVZR 0,0 ; SHIFT ATTEN BIT
720 175404 INC 3,3,SZR
721 776 JMP -2
722 101220 DNUM3: MOVZR 0,0
723 40134 STA 0,ATTS ; SET ATTEN STATUS BIT

```

```

; BEFORE STARTING THE TEST, CHECK TO SEE IF THE DRIVE IS ONLINE, PRINT
; SET DRIVE READY MESSAGE IF NOT

```

```

724 20121 DNUM4: LDA 0,DRVF ; DRIVE NUMBER
725 4211 JSR DDOA ; SELECT IT
726 4242 JSR DDIB ; DRIVE STATUS
727 24420 LDA 1,TESTR-1
730 107414 AND# 0,1,SZR ; DRIVE READY?
731 407 JMP DTST ; YES, START
732 6155 JSR @PRINT ; NO, PRINT MESSAGE
733 17752 NRDT
734 63610 SKPDN TTI ; WAIT FOR START
735 777 JMP -1
736 60510 DIAS 0,TTI ; GOT IT
737 765 JMP DNUM4 ; GO CHECK IT AGAIN

```

```

740 6155 DTST: JSR @PRINT ; NEW LINE
741 20076 CRLF
742 126400 SUB 1,1
743 44146 STA 1,PASS ; RESET PASS COUNT
744 2401 JMP @+1
745 6400 RECAL

```

```

746 40000 BIT1: 40000

```

```

; DRIVE EMULATOR TEST

```

```

747 10000 DRDY
750 24777 TESTR: LDA 1,-1
751 30434 LDA 2,DRVR ; DRV SEL CMD LIST
752 21000 LDA 0,0,2 ; GET SEL CMD
753 101112 MOVL# 0,0,SZC ; END?
754 407 JMP TSTR ; YES
755 4211 JSR DDOA ; NO, SEL DRV
756 4242 JSR DDIB ; GET DRIVE STATUS
757 123415 AND# 1,0,SNR ; READY?
760 411 JMP TSTR1
761 151400 INC 2,2
762 770 JMP TESTR+2 ; LOOP BACK

```

```

; EMULATOR READY, GO START TEST

```

```

763 6155 TSTR: JSR @PRINT
764 20076 CRLF

```

```

765 126400      SUB    1,1
766  44146      STA    1,PASS ; RESET PASS COUNT
767   2401      JMP    @. +1
770  13000      RLDO

```

; EMULATOR NOT READY, PRINT MESSAGE THEN QUIT

```

771   6155 TSTR1: JSR    @PRINT ; "EMULATOR DRV"
772  20057      DRNM
773  25000      LDA    1,0,2
774 125220      MOVZR  1,1 ; RIGHT JUSTIFY DRV #
775 125220      MOVZR  1,1
776 125220      MOVZR  1,1
777 125220      MOVZR  1,1
1000 125220     MOVZR  1,1
1001   6164      JSR    @DECN ; PRINT DRV #
1002   6155      JSR    @PRINT ; "NOT READY"
1003 123711     @TNRD
1004   2111      JMP    @ABORT

1005   1006 DRVR:  . +1
1006     0       DRV0
1007    40       DRV1
1010   100       DRV2
1011   140       DRV3
1012 177777     -1

```

; \*\*\*\* ERROR MESSAGE PRINTOUT \*\*\*\*

; THIS SUBROUTINE IS ENTERED WHEN A TEST ERROR IS DETECTED.  
; IT WILL PROCEED ACCORDING TO THE CURRENT TEST MODE: STOP AND  
; WAIT FOR USER'S COMMAND; PROCEED AFTER PRINTOUT; OR LOOP ON  
; ERROR WITH PRINTOUT SUPPRESSED.

; CALL: JSR @ERROR  
; MESSAGE BLOCK  
; RET: JMP 1,3

; BLOCK FORMAT

; ERROR ADDRESS  
; LOOP ADDRESS  
; TEST SUBJECT TEXT  
; REGISTER INPUT DATA INDEX OR @INPUT DATA TEXT  
; REGISTER OUTPUT DATA INDEX OR @OUTPUT DATA TEXT  
; NORMAL OUTPUT TEXT

1013	0	0		
1014	0	0		
1015	0	0		
1016	0	0		
1017	54777	ERR:	STA 3, -1	; SAVE ACS
1020	50775		STA 2, ERR-2	
1021	44773		STA 1, ERR-3	
1022	40771		STA 0, ERR-4	
1023	4257		JSR NNIOC	; ISSUE CLEAR DEV CMD
1024	20135		LDA 0, MFLG	; MODE FLAG
1025	24050		LDA 1, K2	
1026	30052		LDA 2, K4	
1027	147000		ADD 2, 1	; AC1 = 6
1030	107414		AND# 0, 1, SZR	; LOOPING?
1031	526		JMP ERR4	; YES, EXIT

; PRINT PROGRAM ADDRESS WHERE ERROR IS DETECTED

1032	6155		JSR @PRINT	; PRINT REFERENCE ADDR
1033	17762		REFR	
1034	32762		LDA 2, @ERR-1	; BLOCK POINTER
1035	25000		LDA 1, 0, 2	; GET ERROR ADDR
1036	6165		JSR @OCTN	; PRINT ADDR

; PRINT TEST NAME, TOPIC AND DRIVE NUMBER IF APPLICABLE

1037	24135		LDA 1, MFLG	
1040	125122		MOVZL 1, 1, SZC	; PRINT TEST TITLE?
1041	411		JMP ERROA	; NO
1042	125240		MOVOR 1, 1	; YES, SET 1ST TIME BIT
1043	44135		STA 1, MFLG	
1044	20136		LDA 0, TSTX	
1045	40402		STA 0, +2	; SET TEXT OF TEST NAME
1046	6155		JSR @PRINT	; PRINT TEST NAME
1047	0		0	
1050	6155		JSR @PRINT	; PRINT 'TEST'
1051	117770		@TSTT	
1052	25002	ERROA:	LDA 1, 2, 2	; TOPIC TEXT
1053	44402		STA 1, +2	; SET IT
1054	6155		JSR @PRINT	
1055	0		0	

```

1056 24145 LDA 1, TIMO ; GET TIMEOUT FLAG
1057 125005 MOV 1, 1, SNR ; TIMEOUT?
1060 403 JMP . +3 ; NO
1061 6155 JSR @PRINT ; YES, PRINT "TIMEOUT"
1062 120052 @TMO
1063 24125 LDA 1, DRVN ; DRIVE NUMBER
1064 124015 COM# 1, 1, SNR ; PRINT IT
1065 412 JMP ERR1 ; NO
1066 6155 JSR @PRINT ; YES
1067 20057 DRNM
1070 24275 LDA 1, DDAO ; GET COMMAND WORD
1071 127120 ADDZL 1, 1 ; RIGHT JUSTIFY DRV #
1072 125120 MOVZL 1, 1
1073 125300 MOVS 1, 1
1074 34051 LDA 3, K3
1075 167400 AND 3, 1 ; PICK OFF DRV #
1076 6164 JSR @DECN

```

; DECODE WORDS 3 AND 4 OF BLOCK AND PRINT MESSAGE AS INDICATED

```

1077 6155 ERR1: JSR @PRINT ; "INPUT"
1100 20063 IN
1101 21003 LDA 0, 3, 2 ; GET WORD 3
1102 101113 MOVL# 0, 0, SNC ; INDEX OR TEXT?
1103 415 JMP ERR1A ; INDEX
1104 40402 STA 0, . +2 ; TEXT, SET IT
1105 6155 JSR @PRINT
1106 0 0
1107 6155 JSR @PRINT ; "OUTPUT"
1110 20067 OUT
1111 21004 LDA 0, 4, 2 ; GET WORD 4
1112 101113 MOVL# 0, 0, SNC ; INDEX OR TEXT?
1113 411 JMP ERR1B ; INDEX
1114 40402 STA 0, . +2 ; TEXT, SET IT
1115 6155 JSR @PRINT
1116 0 0
1117 410 JMP ERR2

```

; PRINT REGISTER INPUT-OUTPUT DATA

```

1120 34463 ERR1A: LDA 3, WRGL ; WRITE REG INDEX LIST
1121 117000 ADD 0, 3 ; INDEX
1122 7400 JSR @0, 3 ; PRINT INPUT DATA
1123 764 JMP ERR1+10

1124 34460 ERR1B: LDA 3, RRGL ; READ REG INDEX LIST
1125 117000 ADD 0, 3
1126 7400 JSR @0, 3 ; PRINT OUTPUT DATA

```

; PRINT NORMAL TEST DATA FOR REFERENCE

```

1127 6155 ERR2: JSR @PRINT ; PRINT 'NORMAL'
1130 17773 NORM
1131 21005 LDA 0, 5, 2
1132 101113 MOVL# 0, 0, SNC ; TEXT OR INDEX?
1133 405 JMP ERR2A ; INDEX
1134 40402 STA 0, . +2 ; TEXT, SET IT
1135 6155 JSR @PRINT
1136 0 0

```



```

1137    404      JMP      ERR3

1140  34444  ERR2A: LDA      3, RRGL  ; INDEX LIST
1141  117000      ADD      0, 3
1142    7400      JSR      @0, 3    ; GO PRINT RESULT

```

; CHECK MODE FLAG FOR STOP AND WAIT OR PROCEED

```

1143  20135  ERR3:  LDA      0, MFLG
1144  101213      MDVR#   0, 0, SNC  ; STOP?
1145    431      JMP      ERR4B  ; NO
1146   6155  ERR3A: JSR      @PRINT  ; YES, PRINT PROMPT
1147  20074      PRMT
1150   6157      JSR      @INPUT   ; GET INPUT
1151   1155      ERR3B  ; DEFAULT ERROR
1152   6160      JSR      @MATCH   ; SEARCH COMMAND LIST
1153   3107      CMDL
1154   1000      JMP      0, 2      ; GET EXECUTE COMMAND
1155   6161  ERR3B: JSR      @EQM
1156   1150      ERR3A+2

```

; IN LOOP MODE, CHECK TTI FOR QUIT INPUT. RESET MODE FLAG IF QUIT,  
; ELSE, RESTORE REGISTERS AND CONTINUE

```

1157  63610  ERR4:  SKPDN  TTI      ; STOP LOOPING?
1160    406      JMP      ERR4A  ; NO
1161  124000      COM      1, 1    ; MAKE MASK
1162  123400      AND      1, 0    ; RESET LOOP BIT
1163   40135      STA      0, MFLG
1164   74410      DIA      3, TTI  ; READ TTI
1165    761      JMP      ERR3A  ; GO TO PROMPT

1166  113414  ERR4A: AND#    0, 2, SZR ; LOOPING ON ERROR?
1167   2137      JMP      @TSTV  ; NO, EXIT TO TEST
1170   36626      LDA      3, @ERR-1 ; BLOCK ADDR
1171   35401      LDA      3, 1, 3  ; LOOP ADDR
1172   20621      LDA      0, ERR-4
1173   24621      LDA      1, ERR-3
1174   30621      LDA      2, ERR-2
1175   1400      JMP      0, 3

```

; IN NO STOP MODE, RESTORE REGISTERS THEN EXIT

```

1176  20615  ERR4B: LDA      0, ERR-4
1177  24615      LDA      1, ERR-3
1200  30615      LDA      2, ERR-2
1201  34615      LDA      3, ERR-1
1202   1401      JMP      1, 3

1203   1577  WRGL:  WRGX
1204   1616  RRGL:  RRGX

```

; EXECUTE A LOOP ON ERROR COMMAND  
; SET LOOP BIT IN MODE FLAG AND START LOOPING

```

1205  20135  LOOP:  LDA      0, MFLG
1206  24050      LDA      1, K2
1207  123000      ADD      1, 0      ; SET LOOP FLAG
1210  40135      STA      0, MFLG

```

```

-----
1211 755 JMP ERR4A
; EXECUTE A PROCEED TEST COMMAND
1212 764 PROCD: JMP ERR4B
; EXECUTE A REPEAT TEST COMMAND
1213 20135 REPT: LDA 0, MFLG
1214 24052 LDA 1, K4
1215 107000 ADD 0, 1
1216 44135 STA 1, MFLG ; SET FLAG TO REPEAT
1217 2137 JMP @TSTV ; RE-START TEST
; EXECUTE A SKIP TEST COMMAND
1220 2140 SKPT: JMP @NEXT ; GO TO NEXT TEST
; EXECUTE A CHANGE OPTION COMMAND
; CHANGE FROM STOP ON ERROR TO NO STOP, VICE VERSA
1221 126520 OPTN: SUBZL 1, 1 ; MAKE 1
1222 20135 LDA 0, MFLG ; CHECK MODE FLAG
1223 107415 AND# 0, 1, SNR ; IS STOP ON?
1224 407 JMP OPTN1 ; NO, GO SET IT
1225 124000 COM 1, 1 ; YES, MAKE MASK
1226 123400 AND 1, 0 ; RESET STOP
1227 40135 STA 0, MFLG
1230 6155 JSR @PRINT
1231 20021 NSTP
1232 714 JMP ERR3A
1233 123000 OPTN1: ADD 1, 0 ; SET STOP
1234 40135 STA 0, MFLG
1235 6155 JSR @PRINT ; "STOP"
1236 20014 STP
1237 707 JMP ERR3A

```

; EXECUTE A DUMP BUFFER COMMAND  
; DUMP THE CONTENTS OF THE PROGRAM WRITE OR READ BUFFER  
; HIT ANY KEY EXCEPT CR TO SUSPEND DUMP. CR TO QUIT DUMPING

```
1240 34115 DWBF: LDA 3,WBUF ;WRITE BUFFER ADDR
1241 402 JMP .+2
1242 34116 DRBF: LDA 3,RBUF ;READ BUFFER ADDR
1243 54441 STA 3,DMPP ;SET DUMP POINTER
1244 24441 LDA 1,BLONG
1245 44441 STA 1,DCNT ;SET DUMP LIMIT
1246 6155 DMP: JSR @PRINT ;START NEW LINE
1247 20076 CRLF
1250 20055 LDA 0,K10
1251 40436 STA 0,COLM ;SET COLUMN COUNT
```

; PRINT MEMORY ADDRESS THEN 8 DATA WORDS ON THE SAME ROW

```
1252 24432 LDA 1,DMPP
1253 6165 JSR @OCTN ;PRINT ADDR
1254 26430 DMPO: LDA 1,@DMPP
1255 6165 JSR @OCTN ;PRINT DATA WORD
1256 10426 ISZ DMPP ;BUMP POINTER
1257 14427 DSZ DCNT ;END OF BUFF?
1260 402 JMP .+2 ;NO
1261 2422 JMP @ERRR ;YES, EXIT
1262 63710 SKPDZ TTI ;STOP?
1263 404 JMP DMP2 ;YES
1264 14423 DMP1: DSZ COLM ;NO, LAST COLUMN?
1265 767 JMP DMP0 ;NO
1266 760 JMP DMP ;YES, GO START NEXT ROW
```

; GOT A TTI, SEE WHAT IS IT

```
1267 60510 DMP2: DIAS 0,TTI
1270 24412 LDA 1,CRK
1271 106405 SUB 0,1,SNR ;QUIT?
1272 2411 JMP @ERRR ;YES
1273 63610 SKPDN TTI ;NO, WAIT HERE
1274 777 JMP .-1
1275 60510 DIAS 0,TTI ;READ COMMAND AGAIN
1276 24404 LDA 1,CRK
1277 106405 SUB 0,1,SNR ;QUIT?
1300 2403 JMP @ERRR ;YES
1301 763 JMP DMP1 ;NO, RESUME
```

```
1302 215 CRK: 215
1303 1146 ERRR: ERR3A
1304 0 DMPP: 0
1305 20000 BLONG: 20000
1306 0 DCNT: 0
1307 0 COLM: 0
```

; EXECUTE A DECODE DOA WORD COMMAND  
; DECODE AND PRINT EACH FIELD OF THE DOA WORD

```
1310 20275 DAC: LDA 0,DOAO ;DOA WORD
1311 101113 MOVL# 0,0,SNC ;CLR RW DONE?
1312 403 JMP .+3 ;NO, SKIP IT
```

```

1313 6155 JSR @PRINT ; YES, PRINT "CLR RW ATT"
1314 120643 @CLRW
1315 34105 LDA 3, CATT
1316 117505 ANDL 0, 3, SNR ; PICK CLR DRV ATT BITS
1317 422 JMP DAC2 ; NO CLR DRV ATT
1320 175103 MOVL 3, 3, SNC ; CLR DRVO?
1321 404 JMP DAC1 ; NO
1322 6155 JSR @PRINT ; YES
1323 120652 @CDRO
1324 415 JMP DAC2

```

```

1325 175103 DAC1: MOVL 3, 3, SNC ; CLR DRV1?
1326 404 JMP DAC1A ; NO
1327 6155 JSR @PRINT
1330 120660 @CDR1
1331 410 JMP DAC2

```

```

1332 175103 DAC1A: MOVL 3, 3, SNC ; CLR DRV2?
1333 404 JMP DAC1B ; NO
1334 6155 JSR @PRINT
1335 120666 @CDR2
1336 403 JMP DAC2

```

```

1337 6155 DAC1B: JSR @PRINT
1340 120674 @CDR3

```

; DECODE \$ PRINT COMMAND FIELD

```

1341 30435 DAC2: LDA 2, CMS ; MASK
1342 113400 AND 0, 2 ; PICK OFF
1343 34434 LDA 3, CML ; COMMAND LIST
1344 24072 LDA 1, CR
1345 44152 STA 1, CNTR ; SET COUNT
1346 25400 DAC2A: LDA 1, 0, 3 ; GET REF
1347 146415 SUB# 2, 1, SNR ; THIS ONE?
1350 410 JMP DAC2B ; YES, GO PRINT IT
1351 175400 INC 3, 3 ; NO
1352 175400 INC 3, 3
1353 14152 DSZ CNTR ; MORE?
1354 772 JMP DAC2A ; YES
1355 6155 JSR @PRINT ; NO, PRINT "UNDEFINED CMD"
1356 120702 @UDC
1357 415 JMP DAC2D ; EXIT

```

```

1360 25401 DAC2B: LDA 1, 1, 3 ; GET TEXT
1361 44402 STA 1, .+2 ; SET IT
1362 6155 JSR @PRINT
1363 0 0

```

; CHECK AND PRINT VOLUME NUMBER

```

1364 24057 LDA 1, K20
1365 123404 AND 1, 0, SZR ; WHICH VOLUME
1366 404 JMP DAC2C ; VOL 1
1367 6155 JSR @PRINT ; VOL 0
1370 120714 @VO
1371 403 JMP DAC2D ; ALL DONE

1372 6155 DAC2C: JSR @PRINT

```

```

1373 120720 @V1
1374 2401 DAC2D: JMP @. +1
1375 1146 ERR3A

1376 3600 CMS: 3600

```

; COMMAND FIELD LIST

```

1377 1400 CML: . +1
1400 0 READ
1401 123214 @RDT
1402 200 RCAL
1403 124562 @RCLT
1404 400 SEEK
1405 120724 @SEEKT
1406 600 WHDR
1407 123117 @WHD
1410 1400 FMAT
1411 120727 @FMATT
1412 1600 RLSD
1413 123741 @RELS
1414 2200 ALT1
1415 120733 @ALTT1
1416 2600 ALT2
1417 120743 @ALTT2
1420 2600 NOOP
1421 120753 @NOP
1422 3000 DATV
1423 120757 @DATVT
1424 3200 RDBF
1425 125152 @RFFI
1426 3400 WRITE
1427 123206 @WDT
1430 3600 RHDR
1431 123140 @RHD

```

; EXECUTE A DECODE DOC WORD COMMAND  
; DECODE AND PRINT HEAD, SECTOR AND SECTOR COUNT FIELDS OF DOC WORD

```

1432 20277 DCC: LDA 0, DOC0 ; DOC WORD
1433 105300 MOVS 0, 1 ; RIGHT JUSTIFY HEAD #
1434 125220 MOVZR 1, 1
1435 125220 MOVZR 1, 1
1436 34061 LDA 3, K37
1437 167400 AND 3, 1 ; GOT HEAD #
1440 6155 JSR @PRINT ; "H"
1441 120766 @HD
1442 6164 JSR @DECN ; PRINT HEAD #
1443 24102 LDA 1, SMSK
1444 107600 ANDR 0, 1 ; RIGHT JUSTIFY SECTOR #
1445 125220 MOVZR 1, 1
1446 125220 MOVZR 1, 1
1447 125220 MOVZR 1, 1
1450 125220 MOVZR 1, 1
1451 6155 JSR @PRINT ; "S"
1452 120770 @ST
1453 6164 JSR @DECN ; PRINT SECT #
1454 24061 LDA 1, K37
1455 107400 AND 0, 1 ; GOT SECT COUNT

```

```

1456 124400      NEG      1, 1
1457   6155      JSR      @PRINT ; "SC"
1460 120772      @ESC
1461   6164      JSR      @DECN  ; PRINT SEC COUNT
1462   2401      JMP      @. +1
1463   1146      ERR3A

```

```

; EXECUTE A DECODE DIA WORD COMMAND
; DECODE AND PRINT EACH STATUS BIT WHEN SET

```

```

1464 24057 DAS:  LDA      1, K20
1465 44152      STA      1, CNTR  ; SET SHIFT COUNT
1466 30417      LDA      2, STL   ; STATUS BIT TEXT LIST
1467 20300      LDA      0, DIA0  ; DIA WORD
1470 101103 DAS1: MOVL     0, 0, SNC ; BIT SET?
1471   407      JMP      DAS2   ; NO
1472 25000      LDA      1, 0, 2   ; YES, GET TEXT
1473 125005     MOV      1, 1, SNR  ; DUMMY?
1474   404      JMP      DAS2   ; YES
1475 44402      STA      1, . +2   ; SET IT
1476 6155      JSR      @PRINT  ; PRINT STATUS TEXT
1477   0        O
1500 151400 DAS2: INC      2, 2
1501 14152      DSZ      CNTR   ; DONE?
1502 766      JMP      DAS1   ; NOT YET
1503 2401      JMP      @. +1   ; YES
1504 1146      ERR3A

```

```

; STATUS BIT TEXT LIST

```

```

1505 1506 STL:  . +1
1506 120775     @CF
1507 121005     @RWA
1510 120652     @CDRO
1511 120660     @CDR1
1512 120666     @CDR2
1513 120674     @CDR3
1514   0        O
1515 121012     @ISA
1516 121062     @ECE
1517 121022     @BS
1520 121030     @CYE
1521 121040     @SAE
1522 121053     @VE
1523 121067     @RWO
1524 121076     @DL
1525 121104     @RWF

```

```

; EXECUTE A DECODE DIB STATUS WORD
; DECODE AND PRINT EACH STATUS BIT WHEN SET

```

```

1526 20302 DBST: LDA      0, DIB0  ; DIB WORD
1527 101100     MOVL     0, 0
1530 101103     MOVL     0, 0, SNC  ; DRV RESERVED?
1531   403      JMP      . +3   ; NO
1532 6155      JSR      @PRINT  ; YES
1533 122565     @DRST
1534 101100     MOVL     0, 0
1535 101103     MOVL     0, 0, SNC  ; DRV READY?

```

```

---
1536 403 JMP . +3 ; NO
1537 6155 JSR @PRINT ; YES
1540 121111 @DR
1541 101103 MOVL 0, 0, SNC ; DRV BUSY?
1542 403 JMP . +3 ; NO
1543 6155 JSR @PRINT
1544 121117 @DB
1545 101100 MOVL 0, 0
1546 101103 MOVL 0, 0, SNC ; WRITE DISABLED?
1547 403 JMP . +3 ; NO
1550 6155 JSR @PRINT
1551 122576 @WDST
1552 101100 MOVL 0, 0
1553 101103 MOVL 0, 0, SNC ; SEEK ERROR?
1554 403 JMP . +3 ; NO
1555 6155 JSR @PRINT
1556 121125 @SE
1557 101103 MOVL 0, 0, SNC ; ILLEGAL COMMAND?
1560 403 JMP . +3 ; NO
1561 6155 JSR @PRINT
1562 121135 @IC
1563 20300 LDA 0, DIA0
1564 34062 LDA 3, K70 ; MASK
1565 117405 AND 0, 3, SNR ; DRV FAULT?
1566 403 JMP . +3 ; NO
1567 6155 JSR @PRINT
1570 121142 @DF
1571 101203 MOVR 0, 0, SNC ; DRV ERR?
1572 403 JMP . +3 ; NO
1573 6155 JSR @PRINT
1574 121150 @DER
1575 2401 JMP @. +1
1576 1146 ERR3A

```

; WRITE AND READ REGISTER DATA PRINTOUT LIST

```
1577      0 WRGX: 0
1600    1641      TDO1  ; DOA
1601    1650      TDO2  ; DOB
1602    1657      TDO3  ; DOC
1603    1665      TDO4  ; DOBS
1604    1671      TDO5  ; DOCP
1605    1675      TDO6  ; DOA, DOC, DOBS
1606    1702      TDO7  ; DOAP
1607    1706      TDO10 ; DOA, DOCP
1610    1712      TDO11 ; DOAS
1611    1716      TDO12 ; DOCS
1612    1722      TDO13 ; DOA, DOB, DOCS
1613    1727      TDO14 ; DOA, DOBS
1614    1733      TDO15 ; DOA, NIOP
1615    15753     WBAD   ; PRINT WBUFF ADDR + DATA
```

```
1616      0 RRGX: 0
1617    1742      TDI1   ; DIA(0)
1620    1751      TDI2   ; DIB(0)
1621    1760      TDI3   ; DIC(0)
1622    1767      TDI4   ; DIA(1)
1623    1776      TDI5   ; DIB(1)
1624    2004      TDI6   ; DIA(2)
1625    2010      TDI7   ; DIB(2)
1626    2014      TDI10  ; DIA(0), DIB(0)
1627    2020      TDI11  ; DIA(1), DIB(1)
1630    2024      TDI12  ; DIA(2), DIB(2)
1631    2030      TDI13  ; DIA(0), DIB(0), DIC(0)
1632    15766     RBER   ; PRINT RBUFF ADDR + DATA
1633    2036      TDI15  ; DIAN
1634    2045      TDI16  ; DIBN
1635    2054      TDI17  ; DICN
1636    7440      TIMM   ; TIMEOUT TEST ERROR
1637    2063      PWD    ; PRINT DATA WORD
```

; SUBROUTINES TO PRINT OUT THE CONTENTS OF ANY CONTROLLER REGISTER

```
1640      0
1641    54777 TDO1: STA    3, -1 ; SAVE RETURN
1642    6155      JSR    @PRINT
1643    120077     @MDOA
1644    24275      LDA    1, DOAO
1645    6165      JSR    @OCTN
1646    2772      JMP    @TDO1-1
```

```
1647      0
1650    54777 TDO2: STA    3, -1
1651    6155      JSR    @PRINT
1652    120102     @MDOB
1653    24276      LDA    1, DOBO
1654    6165      JSR    @OCTN
1655    2772      JMP    @TDO2-1
```

```
1656      0
1657    54777 TDO3: STA    3, -1
1660    6155      JSR    @PRINT
1661    120105     @MDOC
```



```

1662 24277 LDA 1, DOC0
1663 6165 JSR @OCTN
1664 2772 JMP @TDO3-1

1665 54762 TDO4: STA 3, TDO2-1
1666 6155 JSR @PRINT
1667 120110 @MDOBS
1670 763 JMP TDO2+3

1671 54765 TDO5: STA 3, TDO3-1
1672 6155 JSR @PRINT
1673 120114 @MDOCP
1674 766 JMP TDO3+3

1675 54443 TDO6: STA 3, RETA
1676 4743 JSR TDO1 ; DOA
1677 4760 JSR TDO3 ; DOC
1700 4765 JSR TDO4 ; DOBS
1701 2437 JMP @RETA

1702 54736 TDO7: STA 3, TDO1-1
1703 6155 JSR @PRINT
1704 120120 @MDOAP
1705 737 JMP TDO1+3

1706 54432 TDO10: STA 3, RETA
1707 4732 JSR TDO1 ; DOA
1710 4761 JSR TDO5 ; DOCP
1711 2427 JMP @RETA

1712 54726 TDO11: STA 3, TDO1-1
1713 6155 JSR @PRINT
1714 120124 @MDOAS
1715 727 JMP TDO1+3

1716 54740 TDO12: STA 3, TDO3-1
1717 6155 JSR @PRINT
1720 120130 @MDOCS
1721 741 JMP TDO3+3

1722 54416 TDO13: STA 3, RETA
1723 4716 JSR TDO1 ; DOA
1724 4724 JSR TDO2 ; DOB
1725 4771 JSR TDO12 ; DOCS
1726 2412 JMP @RETA

1727 54411 TDO14: STA 3, RETA
1730 4711 JSR TDO1 ; DOA
1731 4734 JSR TDO4 ; DOBS
1732 2406 JMP @RETA

1733 54405 TDO15: STA 3, RETA
1734 4705 JSR TDO1 ; DOA
1735 6155 JSR @PRINT
1736 120134 @MNIOP
1737 2401 JMP @RETA

1740 0 RETA: 0

```

```

1741      0      0
1742  54777 TDI1: STA      3, -1
1743   6155      JSR      @PRINT
1744 120137      @MDIA
1745  24300      LDA      1, DIA0
1746   6165      JSR      @OCTN
1747   2772      JMP      @TDI1-1

1750      0      0
1751  54777 TDI2: STA      3, -1
1752   6155      JSR      @PRINT
1753 120142      @MDIB
1754  24302      LDA      1, DIB0
1755   6165      JSR      @OCTN
1756   2772      JMP      @TDI2-1

1757      0      0
1760  54777 TDI3: STA      3, -1
1761   6155      JSR      @PRINT
1762 120145      @MDIC
1763  24304      LDA      1, DICO
1764   6165      JSR      @OCTN
1765   2772      JMP      @TDI3-1

1766      0      0
1767  54777 TDI4: STA      3, -1
1770   6155      JSR      @PRINT
1771 120150      @MDIA1
1772  24301      LDA      1, DIA1
1773   6165      JSR      @OCTN
1774   2772      JMP      @TDI4-1

1775      0      0
1776  54777 TDI5: STA      3, -1
1777   6155      JSR      @PRINT
2000 120155      @MDIB1
2001  24303      LDA      1, DIB1
2002   6165      JSR      @OCTN
2003   2772      JMP      @TDI5-1

2004  54762 TDI6: STA      3, TDI4-1
2005   6155      JSR      @PRINT
2006 120162      @MDIA2
2007   763      JMP      TDI4+3

2010  54765 TDI7: STA      3, TDI5-1
2011   6155      JSR      @PRINT
2012 120167      @MDIB2
2013   766      JMP      TDI5+3

2014  54724 TDI10: STA     3, RETA
2015   4725      JSR      TDI1      ; DIA(0)
2016   4733      JSR      TDI2      ; DIB(0)
2017   2721      JMP      @RETA

2020  54720 TDI11: STA     3, RETA
2021   4746      JSR      TDI4      ; DIA(1)
2022   4754      JSR      TDI5      ; DIB(1)
2023   2715      JMP      @RETA

```

```

2024 54714 TDI12: STA 3, RETA
2025 4757 JSR TDI6 ; DIA(2)
2026 4762 JSR TDI7 ; DIB(2)
2027 2711 JMP @RETA

2030 54710 TDI13: STA 3, RETA
2031 4711 JSR TDI1 ; DIA(0)
2032 4717 JSR TDI2 ; DIB(0)
2033 4725 JSR TDI3 ; DIC(0)
2034 2704 JMP @RETA

2035 0 0
2036 54777 TDI15: STA 3, -1
2037 6155 JSR @PRINT
2040 120137 @MDIA
2041 24305 LDA 1, DIAN
2042 6165 JSR @OCTN
2043 2772 JMP @TDI15-1

2044 0 0
2045 54777 TDI16: STA 3, -1
2046 6155 JSR @PRINT
2047 120142 @MDIB
2050 24306 LDA 1, DIBN
2051 6165 JSR @OCTN
2052 2772 JMP @TDI16-1

2053 0 0
2054 54777 TDI17: STA 3, -1
2055 6155 JSR @PRINT
2056 120145 @MDIC
2057 24307 LDA 1, DICN
2060 6165 JSR @OCTN
2061 2772 JMP @TDI17-1

2062 0 0
2063 54777 PWD: STA 3, -1
2064 6155 JSR @PRINT
2065 120000 @WORD
2066 24141 LDA 1, ACN
2067 6165 JSR @OCTN
2070 2772 JMP @PWD-1

```



;\*\*\*\* TEXT PRINTER \*\*\*\*

; PRINT THE CHARACTER STRING OF TEXT NAME FROM A NEW LINE.  
; IF @ SIGN PRECEDES TEXT NAME, PRINTING WILL START AT THE  
; CURRENT POSITION OF DISPLAY.

; CALL: JSR @PRINT  
; TEXT NAME  
; RET: JMP 1,3 ACO & AC2 RESTORED

2071	0	0		
2072	0	0		
2073	0	0		
2074	0	0		
2075	54777	PRNT:	STA 3,-1	; SAVE RETURN
2076	50775		STA 2,PRNT-2	
2077	44773		STA 1,PRNT-3	
2100	40771		STA 0,PRNT-4	
2101	31400		LDA 2,0,3	; TEXT VECTOR
2102	151122		MOVZL 2,2,SZC	; START CRLF?
2103	407		JMP PRNT1	; NO
2104	63511		SKPBZ TTO	; CRLF IF NOT BUSY
2105	777		JMP -1	
2106	20072		LDA 0,CR	
2107	4431		JSR ACHR	; ISSUE CR
2110	20073		LDA 0,LF	
2111	4427		JSR ACHR	; ISSUE LF
2112	155220	PRNT1:	MOVZL 2,3	; SET TXT POINTERR
2113	126400		SUB 1,1	
2114	44147		STA 1,PCH	; RESET PRINT COUNT
2115	24065		LDA 1,RMSK	; MASK
2116	21400	PRNT2:	LDA 0,0,3	; FETCH A WORD
2117	101300		MOVS 0,0	; SWAP BYTES
2120	123415		AND# 1,0,SNR	; END OF STRING?
2121	412		JMP PRNT3	; YES
2122	61111		DOAS 0,TTO	; NO, PRINT
2123	10147		ISZ PCH	; COUNT CHARACTERS
2124	151400		INC 2,2	
2125	63611		SKPDN TTO	
2126	777		JMP -1	
2127	151213		MOVR# 2,2,SNC	; BUMP WORD POINTER?
2130	175401		INC 3,3,SKP	; YES
2131	766		JMP PRNT2+1	; PRINT THE OTHER BYTE
2132	764		JMP PRNT2	
2133	34741	PRNT3:	LDA 3,PRNT-1	
2134	30737		LDA 2,PRNT-2	
2135	24735		LDA 1,PRNT-3	
2136	20733		LDA 0,PRNT-4	
2137	1401		JMP 1,3	

;\*\*\*\* PRINT SINGLE CHARACTER \*\*\*\*

; CALL: ACO = CHARACTER RIGHT JUSTIFIED  
; JSR @CHAR  
; RET: JMP 0,3

2140	61111	ACHR:	DOAS 0,TTO	
2141	63611		SKPDN TTO	

```

2142     777          JMP     .-1
2143  10147         ISZ     PCH      ; COUNT CHARS
2144   1400         JMP     0,3

```

;\*\*\*\* INPUT ERROR RESPONSE \*\*\*\*

;PRINT '?' TO INDICATE INPUT ERROR THEN EXIT  
;TO VECTORED ROUTINE.

```

2145     0          0
2146  54777 GM:    STA     3,.-1
2147   6155        JSR     @PRINT
2150  20174        GMT
2151  34774        LDA     3,GM-1
2152   3400        JMP     @0,3      ; EXIT TO VECTOR

```

;\*\*\*\* COMMAND STRING INPUTTER \*\*\*\*

;STORE INPUT CHARACTERS INTO THE CHARACTER BUFFER, RETURNS  
;WHEN CR IS INPUTTED.

```

;CALL: JSR @INPUT
;      DEFAULT VECTOR
;RET:  CR INPUTTED
;      JMP @0,3 FOR DEFAULT, AC1 = 0
;      JMP 1,3 AC1 = CHAR COUNT

```

```

2153     0          0
2154  54777 INPT:  STA     3,.-1
2155  30110        LDA     2,CBUF      ; CHAR BUFF
2156  50466        STA     2,CBP      ; SET POINTER
2157  126400       SUB     1,1
2160  44463        STA     1,CCNT     ; CLEAR CHAR COUNT
2161  20057        LDA     0,K20     ; CBUF LENGTH
2162  100400       NEG     0,0
2163  45000        STA     1,0,2     ; CLEAR OUT CBUF
2164  151400       INC     2,2
2165  101404       INC     0,0,SZR
2166   775        JMP     .-3

```

;STORE INPUT CHARACTER IF IT IS NOT CR, RUBOUT OR BACK-  
;SPACE.

```

2167  60110        NIOS    TTI
2170  63610 INP1:  SKPDN  TTI      ; WAIT FOR INPUT
2171   777        JMP     .-1
2172  60510        DIAS    0,TTI     ; GOT ONE
2173  24065        LDA     1,RMSK
2174  123400       AND     1,0      ; STRIP PARITY
2175  24072        LDA     1,CR
2176  106405       SUB     0,1,SNR    ; TERMINATE INPUT?
2177   427        JMP     INP3      ; YES
2200  24055        LDA     1,K10
2201  106405       SUB     0,1,SNR    ; BACKSPACE?
2202   434        JMP     INP5      ; YES
2203  24075        LDA     1,RUBO
2204  106405       SUB     0,1,SNR    ; RUBOUT?
2205   426        JMP     INP4
2206  6156        JSR     @CHAR     ; ECHO INPUT

```

```

2207 10434      ISZ   CCNT   ; COUNT INPUTS
2210 30433      LDA   2,CCNT
2211 34057      LDA   3,K20   ; CBUF LENGTH
2212 172432     SUBZ# 3,2,SZC  ; OVER?
2213   425      JMP   INP6   ; YES
2214 26430      LDA   1,@CBP ; FETCH LAST STORED INPUT
2215 151212     MOVR# 2,2,SZC  ; PACK TO LEFT OR RIGHT
2216   405      JMP   INP2   ; LEFT
2217 123000     ADD   1,0     ; RIGHT
2220 42424      STA   0,@CBP ; STORE WORD
2221 10423      ISZ   CBP
2222   746      JMP   INP1

```

```

2223 101300 INP2: MOVS  0,0
2224 42420      STA   0,@CBP
2225   743      JMP   INP1

```

; INPUT DONE, CONVERT CHARACTER COUNT TO WORD COUNT

```

2226 34725 INP3: LDA   3,INPT-1
2227 24414      LDA   1,CCNT  ; CHARACTER COUNT
2230 125005     MOV   1,1,SNR  ; DEFAULT?
2231   3400      JMP   @0,3   ; YES, EXIT
2232   1401      JMP   1,3    ; RETURN

```

; INPUTTTED RUBOUT, DELETE LAST CHARACTER FROM CBUF  
; AND ECHO IT.

```

2233   4414 INP4: JSR   DELC
2234   6156      JSR   @CHAR
2235   733      JMP   INP1

```

; INPUTTED BACKSPACE, DELETE LAST CHARACTER FROM CBUF

```

2236   4411 INP5: JSR   DELC
2237   731      JMP   INP1

```

; BUFFER FULL, PRINT MESSAGE THEN RESET CBUFF

```

2240   6155 INP6: JSR   @PRINT
2241 20176      CBFUL
2242   714      JMP   INPT+2

```

```

2243   0 CCNT: 0
2244   0 CBP: 0
2245 177400 LMSK: 177400

```

; A SUBROUTINE TO DELETE THE MOST RECENT INPUT FROM CBUF.  
; ON RETURN, ACO = DELETE CHAR.

```

2246   0      0
2247 24774 DELC: LDA   1,CCNT  ; CHAR COUNT
2250 125005     MOV   1,1,SNR  ; ZERO?
2251   1400      JMP   0,3    ; YES, RETURN
2252 54774      STA   3,DELC-1 ; SAVE RETURN
2253   6156      JSR   @CHAR  ; ECHO DELETE INPUT
2254 125213     MOVR# 1,1,SNC  ; ODD OR EVEN?
2255   410      JMP   DEL2   ; EVEN
2256 22766      LDA   0,@CBP  ; ODD, FETCH WORD

```

```

2257 101300      MOVS   0,0
2260 126400      SUB    1,1
2261  46763 DEL1: STA    1,@CBP ; ERASE CHAR
2262  14761      DSZ    CCNT   ; BACKUP COUNT
2263    401      JMP    .+1
2264   2762      JMP    @DELC-1 ; RETURN

2265  14757 DEL2: DSZ    CBP    ; BACKUP STORE POINTER
2266  22756      LDA    0,@CBP ; GET WORD
2267 105000      MOV    0,1
2270  30755      LDA    2,LMSK ; MASK
2271 147400      AND    2,1   ; ERASE RIGHT HALF
2272 151300      MOVS   2,2
2273 143400      AND    2,0
2274   765      JMP    DEL1

```

;\*\*\*\* SET ASCII DIGITS \*\*\*\*

```

; CONVERTS 2 OCTAL DIGITS INTO 2 ASCII BYTES FOR PRINTING
; CALL: ACO = OCTAL DIGITS
;      JSR @SADC
; RET: AC1 = ASCII BYTES PACKED L-R

```

```

2275    0      0
2276  54777 SAD: STA    3,-1
2277 105000      MOV    0,1
2300 125200      MOVR   1,1   ; RIGHT JUSTIFY
2301 125200      MOVR   1,1
2302 125200      MOVR   1,1
2303  34054      LDA    3,K7
2304 163400      AND    3,0   ; WANT 3 LSB
2305 167700      ANDS   3,1
2306 107000      ADD    0,1
2307  34403      LDA    3,ASC00
2310 167000      ADD    3,1   ; FORM ASCII DIGITS
2311   2764      JMP    @SAD-1 ; RETURN

2312  30060 ASC00: 30060

```



;\*\*\*\* CHARACTER STRING MATCH \*\*\*\*

; SEARCH THROUGH THE GIVEN DIRECTORY FOR NAME MATCH WITH  
; THAT STORED IN CBUF

; CALL: JSR @MATCH  
; DIRECTORY NAME  
; RET: JMP 1,3 FOUND, AC2 = VECTOR  
; JMP 2,3 NOT FOUND

```
2313      0          0
2314 54777 MTCH: STA    3,.-1
2315 35400      LDA    3,0,3 ; GET DIRECTORY
2316 54445      STA    3,DIRP ; SET POINTER
2317 25400      LDA    1,0,3 ; GET CHAR COUNT OF NAME
2320 125112     MOVL#  1,1,SZC ; -1?
2321      420     JMP    MTCH2 ; YES, SKIP IT
2322 125005     MOV    1,1,SNR ; END?
2323      436     JMP    MTCH5 ; YES, NO MATCH
2324 44152     STA    1,CNTR ; NO, SET CHAR COUNT
2325 35401     LDA    3,1,3 ; GET NAME TEXT ADDR
2326 54142     STA    3,SAVP ; SET POINTER
2327 30110     LDA    2,CBUF ; CHAR BUFF
2330 21000 MTCH1: LDA    0,0,2 ; FETCH INPUT WORD
2331 26142     LDA    1,@SAVP ; GET NAME CHAR
2332 34152     LDA    3,CNTR
2333 175224     MOVZR  3,3,SZR ; SINGLE CHAR?
2334      415     JMP    MTCH4 ; NO
2335 34710     LDA    3,LMSK ; YES, GET MASK
2336 163400     AND    3,0 ; WANT UPPER BYTE
2337 106415     SUB#   0,1,SNR ; MATCH?
2340      405     JMP    MTCH3 ; YES, GOT IT
```

; INDEX TO NEXT ENTRY OF DIRECTORY

```
2341 34422 MTCH2: LDA    3,DIRP
2342 24051      LDA    1,K3
2343 137000     ADD    1,3 ; INDEX
2344      752     JMP    MTCH+2 ; CONTINUE
```

; FOUND MATCH, LOAD VECTOR AND RETURN

```
2345 30416 MTCH3: LDA    2,DIRP
2346 31002      LDA    2,2,2
2347 34744      LDA    3,MTCH-1
2350 1401       JMP    1,3
```

; MATCH A WORD, COUNT CHARACTERS AND BUMP POINTERS

```
2351 106414 MTCH4: SUB#   0,1,SZR ; MATCH?
2352      767     JMP    MTCH2 ; NO HERE
2353 10142     ISZ    SAVP ; YES, NEXT WORD
2354 151400     INC    2,2
2355 14152     DSZ    CNTR
2356 14152     DSZ    CNTR
2357      751     JMP    MTCH1 ; MORE
2360      765     JMP    MTCH3 ; GOT IT
```

; NO MATCH RETURN

```

2361 34732 MTCH5:LDA 3,MTCH-1
2362 1402 JMP 2,3

```

```

2363 0 DIRP: 0

```

```

;**** ASCII DECIMAL-BINARY CONVERSION ****

```

```

; CONVERT ONE OR MORE DECIMAL STRINGS, SEPARATED BY COMMAS,
; IN CBUF AND STORE THE RESULT IN OR CONSECUTIVE LOCATIONS
; AS POINTED TO BY THE CALLING VECTOR.

```

```

; CALL: JSR @ABIN

```

```

; STORE VECTOR

```

```

; RET: JMP 2,3 DONE, ACO=LAST BINARY, AC1=ENTRY COUNT

```

```

; JMP @1,3 ILLEGAL INPUT CHARACTER

```

```

2364 0 0
2365 0 0
2366 54777 ABN: STA 3,.-1
2367 50775 STA 2,ABN-2
2370 31400 LDA 2,0,3 ; GET STORE ADDRESS
2371 50476 STA 2,BINA ; SET IT
2372 30110 LDA 2,CBUF ; CHAR BUFF
2373 50472 STA 2,IPTR ; SET CBUF POINTER
2374 126400 SUB 1,1
2375 44471 STA 1,NCNT ; CLEAR ENTRY COUNT
2376 44466 ABN1: STA 1,ICNT ; RESET CHAR COUNT
2377 44471 STA 1,BINR ; CLEAR OLD RESULT
2400 22465 ABN2: LDA 0,@IPTR ; FETCH WORD
2401 24065 LDA 1,RMSK ; MASK
2402 101300 MOVS 0,0 ; SWAP BYTES
2403 107400 AND 0,1 ; PICK OFF A BYTE
2404 30077 LDA 2,ZERO
2405 34101 LDA 3,NINE
2406 166033 ADCZ# 3,1,SNC ; SKIP IF >9
2407 132032 ADCZ# 1,2,SZC ; SKIP IF = OR >0
2410 417 JMP ABN3 ; NOT 0-9
2411 34056 LDA 3,K17
2412 167400 AND 3,1 ; WANT 4 LSBS
2413 30455 LDA 2,BINR ; BINARY
2414 155000 MOV 2,3 ; TIMES 10
2415 153120 ADDZL 2,2
2416 173120 ADDZL 3,2
2417 147000 ADD 2,1 ; ADD LAST CONVERSION
2420 44450 STA 1,BINR ; SAVE IT
2421 10443 ISZ ICNT ; COUNT DIGITS
2422 34442 LDA 3,ICNT
2423 175212 MOVR# 3,3,SZC ; ODD OR EVEN?
2424 755 JMP ABN2+1 ; ODD
2425 10440 ISZ IPTR ; EVEN, BUMP POINTER
2426 752 JMP ABN2

```

```

; NONDECIMAL CHARACTER, CHECK IT FOR NULL, COMMA OR ELSE.

```

```

2427 10437 ABN3: ISZ NCNT ; COUNT ENTRIES
2430 125014 MOV# 1,1,SZR ; END OF INPUT?
2431 407 JMP ABN3A ; NO
2432 20436 LDA 0,BINR ; GET BINARY
2433 42434 STA 0,@BINA ; STORE LAST ENTRY
2434 24432 LDA 1,NCNT ; GET ENTRY COUNT

```

```

2435 34730 LDA 3, ABN-1
2436 30726 LDA 2, ABN-2
2437 1402 JMP 2, 3

2440 30074 ABN3A: LDA 2, COMMA
2441 146415 SUB# 2, 1, SNR ; A SEPARATOR?
2442 404 JMP ABN4 ; YES
2443 34722 LDA 3, ABN-1 ; NO, BAD INPUT
2444 30720 LDA 2, ABN-2
2445 3401 JMP @1, 3 ; RETURN

2446 24422 ABN4: LDA 1, BINR ; GET BINARY
2447 46420 STA 1, @BINA ; STORE IT
2450 10417 ISZ BINA
2451 30413 LDA 2, ICNT ; GET COUNT
2452 151203 MOVR 2, 2, SNC ; ODD OR EVEN?
2453 404 JMP ABN4A ; EVEN
2454 10411 ISZ IPTR ; ODD, BUMP POINTER
2455 126400 SUB 1, 1
2456 720 JMP ABN1

2457 176400 ABN4A: SUB 3, 3
2460 54410 STA 3, BINR ; CLEAR OLD STUFF
2461 176520 SUBZL 3, 3
2462 54402 STA 3, ICNT ; SET COUNT TO ONE
2463 716 JMP ABN2+1 ; KEEP GOING

2464 0 ICNT: 0
2465 0 IPTR: 0
2466 0 NCNT: 0
2467 0 BINA: 0
2470 0 BINR: 0

```

; \*\*\*\* ASCII OCTAL-BINARY CONVERSION \*\*\*\*

; CONVERT ONE OR MORE OCTAL STRING, SEPARATED BY COMMAS,  
; STORED IN CBUFF, AND STORE THE RESULTS IN THE DESIGNATED  
; BUFFER AREA

```

; CALL: JSR @OBIN
; STORE RESULT AREA
; INPUT ERROR VECTOR
; RET: JMP @1, 3 ILLEGAL CHARACTER
; JMP 2, 3 DONE, ACO=LAST BINARY, AC1=ENTRY COUNT

```

```

2471 0 0
2472 0 0
2473 54777 OBN: STA 3, -1
2474 50775 STA 2, OBN-2
2475 31400 LDA 2, 0, 3
2476 50771 STA 2, BINA ; SET STORE POINTER
2477 30110 LDA 2, CBUF
2500 50765 STA 2, IPTR ; SET CHARACTER BUFFER POINTER
2501 126400 SUB 1, 1
2502 44764 STA 1, NCNT ; RESET ENTRY COUNT
2503 44761 OBN1: STA 1, ICNT ; RESET CHAR COUNT
2504 44764 STA 1, BINR ; CLEAR OLD STUFF
2505 22760 OBN2: LDA 0, @IPTR ; FETCH INPUT BYTES
2506 24065 LDA 1, RMSK
2507 101300 MOVS 0, 0

```

```

2510 107400      AND      0,1      ; PICK OFF 1ST BYTE
2511 30077       LDA      2,ZERO
2512 34100       LDA      3,SEVN
2513 166033      ADCZ#   3,1,SNC   ; SKIP IF >7
2514 132032      ADCZ#   1,2,SZC   ; SKIP IF =>0
2515 416         JMP      OBN3     ; NOT OCTAL
2516 34054       LDA      3,K7
2517 167400      AND      3,1      ; WANT 3 LSBS
2520 30750       LDA      2,BINR
2521 153120      ADDZL   2,2      ; LEFT SHIFT 3 PLACES
2522 151120      MOVZL   2,2
2523 147000      ADD      2,1      ; INSERT NEW DIGIT
2524 44744       STA      1,BINR   ; SAVE RESULT
2525 10737       ISZ      ICNT   ; COUNT CONVERTED DIGITS
2526 34736       LDA      3,ICNT
2527 175212      MOVR#   3,3,SZC   ; ODD OR EVEN COUNT?
2530 756         JMP      OBN2+1   ; ODD
2531 10734       ISZ      IPTR   ; EVEN, BUMP POINTER
2532 753         JMP      OBN2

```

; GOT A NONOCTAL CHARACTER, CHECK IT FOR COMMA, NULL OR ELSE

```

2533 125004 OBN3: MOV      1,1,SZR   ; A NULL?
2534 410     JMP      OBN3A   ; NO
2535 10731   ISZ      NCNT   ; YES, NO MORE INPUTS
2536 20732   LDA      0,BINR   ; GET BINARY
2537 42730   STA      0,@BINA   ; SAVE IT
2540 24726   LDA      1,NCNT   ; ENTRY COUNT
2541 30730   LDA      2,OBN-2   ; RESTORE AC2
2542 34730   LDA      3,OBN-1
2543 1402    JMP      2,3

2544 30074 OBN3A: LDA      2,COMMA
2545 146415  SUB#     2,1,SNR   ; A SEPARATOR?
2546 405     JMP      OBN4     ; YES, EXIT
2547 24717   LDA      1,NCNT   ; NO, ILLEGAL CHAR
2550 30721   LDA      2,OBN-2
2551 34721   LDA      3,OBN-1
2552 3401    JMP      @1,3

2553 24715 OBN4:  LDA      1,BINR
2554 46713   STA      1,@BINA   ; STORE AWAY BINARY
2555 10712   ISZ      BINA    ; BUMP POINTER
2556 30706   LDA      2,ICNT   ; CHECK CHAR COUNT
2557 151203  MOVR#   2,2,SNC   ; ODD OR EVEN?
2560 404     JMP      OBN4A   ; EVEN
2561 10704   ISZ      IPTR   ; ODD, BUMP CHAR POINTER
2562 126400  SUB      1,1
2563 720     JMP      OBN1

2564 176400 OBN4A: SUB      3,3
2565 54703   STA      3,BINR   ; CLEAR OLD NUMBER
2566 175140  MOVOL   3,3
2567 54675   STA      3,ICNT   ; RESET COUNT TO 1
2570 716     JMP      OBN2+1   ; CONTINUE

```

; \*\*\*\* DECIMAL PRINTOUT \*\*\*\*

; CONVERT A BINARY NUMBER TO DECIMAL AND PRINT IT. LEADING

```

; ZEROS ARE SUPPRESSED AND A SPACE AFTER.
; CALL: AC1 = BINARY NUMBER
; JSR @DECN
; RET: JMP 0,3

```

```

2571      0      0
2572      0      0
2573 54777 DEC: STA 3,.-1
2574 125004 MOV 1,1,SZR ; ZERO?
2575 404 JMP DEC1 ; NONZERO
2576 20077 LDA 0,ZERO
2577 6156 JSR @CHAR ; YES, PRINT IT
2600 430 JMP DEC3

2601 50770 DEC1: STA 2,DEC-2
2602 102400 SUB 0,0
2603 40437 STA 0,LEAD
2604 20427 LDA 0,TENA
2605 40434 STA 0,TENN ; SET BASE ADDRESS
2606 32433 DEC2: LDA 2,@TENN ; GET BASE NUMBER
2607 20077 LDA 0,ZERO ; ZERO
2610 146443 SUBO 2,1,SNC ; SKIP IF MINUS
2611 101401 INC 0,0,SKP ; ADD 1
2612 147001 ADD 2,1,SKP ; OVER, PUT IT BACK
2613 775 JMP .-3
2614 34077 LDA 3,ZERO
2615 116404 SUB 0,3,SZR ; A ZERO?
2616 404 JMP .+4 ; NO
2617 34423 LDA 3,LEAD
2620 175005 MOV 3,3,SNR ; YES, LEADING?
2621 403 JMP .+3 ; NO
2622 10420 ISZ LEAD ; YES, SET FLAG
2623 6156 JSR @CHAR ; PRINT DIGIT
2624 10415 ISZ TENN ; BUMP POINTER
2625 151223 MOVZR 2,2,SNC ; DONE?
2626 760 JMP DEC2 ; NOT YET
2627 30742 LDA 2,DEC-2 ; RESTORE AC2
2630 20076 DEC3: LDA 0,SPACE
2631 6156 JSR @CHAR ; YES, PRINT A SPACE
2632 2740 JMP @DEC-1

```

```

2633 2634 TENA: .+1
2634 23420 23420
2635 1750 1750
2636 144 144
2637 12 12
2640 1 1

```

```

2641 0 TENN: 0
2642 0 LEAD: 0

```

```

; **** OCTAL NUMBER PRINTOUT ****

```

```

; CALL: AC1 = BINARY NUMBER
; JSR @OCTN
; RET: JMP 0,3

```

```

2643      0      0
2644 54777 OCT: STA 3,.-1

```

```

2645 20414 LDA 0,SHIFT
2646 103120 ADDZL 0,0
2647 125140 MOVOL 1,1 ;BRING IN 1ST DIGIT
2650 101103 OCT1: MOVL 0,0,SNC
2651 403 JMP OCT2
2652 6156 JSR @CHAR ;PRINT DIGIT
2653 20406 LDA 0,SHIFT
2654 127024 OCT2: ADDZ 1,1,SZR ;DONE?
2655 773 JMP OCT1 ;NO
2656 20076 LDA 0,SPACE
2657 6156 JSR @CHAR ;YES, SPACE IT
2660 2763 JMP @OCT-1

```

```

2661 20006 SHIFT: 20006

```

```

;**** 1 MS SOFTWARE CLOCK ****

```

```

; INNER LOOP TIME IS 1 MILLISECOND BASED ON INSTRUCTION
; TIMES OF: DSZ = 1.1 US
;           JMP = 0.4 US
; CALL: MS COUNT SET IN CLOK
;       JSR @TIMER
; RET: JMP 0,3 WHEN DISK IS DONE OR TIMED OUT

```

```

2662 0 0
2663 54777 TIME: STA 3,-1 ;SAVE RETURN
2664 126400 SUB 1,1
2665 44145 STA 1,TIMO ;RESET TIMEOUT FLAG
2666 24412 TIME1: LDA 1,TK1
2667 44412 STA 1,TM1 ;SET INNER LOOP COUNT
2670 14411 DSZ TM1
2671 777 JMP -1
2672 4264 JSR SKDZ ;RETURN IF DONE
2673 2767 JMP @TIME-1 ;RETURN
2674 14144 DSZ CLOK ;TIMEOUT?
2675 771 JMP TIME1 ;NO, CONTINUE
2676 10145 ISZ TIMO ;YES, SET TIMEOUT FLAG
2677 2763 JMP @TIME-1 ;TIMED OUT EXIT

```

```

2700 1127 TK1: 1127
2701 0 TM1: 0

```

;\*\*\*\* ABORT OPERATION \*\*\*\*;

2702 2111 QUIT: JMP @ABORT

;\*\*\*\* GOTO COMMAND \*\*\*\*;

;EXECUTE A GOTO MEMORY LOCATION COMMAND

2703 126400 GO: SUB 1,1  
2704 44445 STA 1,GOTO ;RESET VECTOR  
2705 44152 STA 1,CNTR ;CLEAR COUNT  
2706 24110 LDA 1,CBUF  
2707 44142 STA 1,SAVP ;SET BUFF POINTER  
2710 22142 LDA 0,@SAVP ;FETCH 1ST WORD  
2711 403 JMP .+3 ;GO WORK ON IT  
2712 22142 GO1: LDA 0,@SAVP ;GET A WORD  
2713 101300 MOVS 0,0  
2714 24066 LDA 1,BMSK ;GET BYTE MASK  
2715 107404 AND 0,1,SZR ;CHECK FOR END  
2716 405 JMP GO2 ;NOT END  
2717 34152 LDA 3,CNTR ;CHECK COUNT  
2720 175005 MOV 3,3,SNR ;ADDR THERE?  
2721 2431 JMP @NOGO ;NONE, ERROR  
2722 2427 JMP @GOTO ;YES, GO THERE  
  
2723 30077 GO2: LDA 2,ZERO  
2724 34100 LDA 3,SEVN  
2725 166033 ADCZ# 3,1,SNC ;SKIP IF >7  
2726 132032 ADCZ# 1,2,SZC ;SKIP IF =>0  
2727 2423 JMP @NOGO ;NOT OCTAL  
2730 30054 LDA 2,K7  
2731 147400 AND 2,1 ;WANT 3 BITS  
2732 34417 LDA 3,GOTO  
2733 175122 MOVZL 3,3,SZC ;LEFT SHIFT 3 PLACES  
2734 2416 JMP @NOGO ;OVERFLOW  
2735 175122 MOVZL 3,3,SZC  
2736 2414 JMP @NOGO  
2737 175122 MOVZL 3,3,SZC  
2740 2412 JMP @NOGO  
2741 167000 ADD 3,1  
2742 44407 STA 1,GOTO ;ADD LAST DIGIT  
2743 10152 ISZ CNTR ;COUNT CHARS  
2744 30152 LDA 2,CNTR  
2745 151203 MOVR 2,2,SNC ;SWAP OR BUMP?  
2746 745 JMP GO1+1 ;SWAP BYTES  
2747 10142 ISZ SAVP ;BUMP POINTER  
2750 742 JMP GO1  
  
2751 0 GOTO: 0  
2752 1155 NOGO: ERR3B

;\*\*\*\* INITIALIZE TEST STEP \*\*\*\*;

;SET THE PARAMETERS OF THE CALLING BLOCK BEFORE SSTARTING TEST  
;CALL: JSR @SET  
; TEXT OF TEST TITLE  
; TEST START VECTOR  
; NEXT TEST START VECTOR  
; PRINT DRIVE NUMBER FLAG: 0 = YES, -1 = NO

```

2753 24143 SETT: LDA 1,PFLG
2754 63610 SKPDN TTI ;STOP?
2755 404 JMP .+4 ;NO
2756 20051 LDA 0,K3
2757 122415 SUB# 1,0,SNR ;EMULATOR TEST?
2760 2422 JMP @ENDM ;YES, EXIT
2761 125225 MOVZR 1,1,SNR ;LOGIC TEST?
2762 403 JMP .+3 ;YES, SKIP TTI TEST
2763 63710 SKPDZ TTI ;STOP TEST?
2764 2111 JMP @ABORT ;YES, EXIT
2765 21400 LDA 0,0,3
2766 40136 STA 0,TSTX ;SET TITLE TEXT ADDR
2767 21401 LDA 0,1,3
2770 40137 STA 0,TSTV ;SET TEST START VECTOR
2771 21402 LDA 0,2,3
2772 40140 STA 0,NEXT ;SET NEXT TEST VECTOR
2773 25403 LDA 1,3,3
2774 44125 STA 1,DRVN ;SET PRINT DRIVE # FLAG
2775 24135 LDA 1,MFLG
2776 125120 MOVZL 1,1 ;RESET PRINT TITLE FLAG
2777 125220 MOVZR 1,1
3000 44135 STA 1,MFLG
3001 2137 JMP @TSTV ;GO START TEST

```

```
3002 15440 ENDM: ENDE1
```

```
;**** TEST STEP CONNECTOR ****
```

```
; MONITOR FOR LOOP MODE AT END OF A TEST STEP, PROCEED TO THE
; NEXT STEP IF NOT IN LOOP MODE. ELSE LOOP BACK AS INDICATED IN
; MFLG.
```

```

3003 24050 NTS: LDA 1,K2
3004 30052 LDA 2,K4
3005 147000 ADD 2,1 ;COMBINE LOOP FLAG BITS
3006 20135 LDA 0,MFLG
3007 107415 AND# 0,1,SNR ;LOOPING
3010 2140 JMP @NEXT ;NO, NEXT STEP
3011 2401 JMP @.+1 ;LOOP
3012 1157 ERR4

```



; \*\*\*\* PRINT HELP MESSAGE SUBROUTINES \*\*\*\*

; DEVICE CODE ENTRY

```
3013 6155 HLP0: JSR @PRINT ; "ENTER 2 OCTAL DIGITS. DEFAULTS TO 2:
3014 20211 MSG0
3015 2401 JMP @. +1
3016 452 IDC+2
```

; TEST MODE ENTRY

```
3017 6155 HLP1: JSR @PRINT ; "LOGIC TEST LOCAL LOGIC ONLY"
3020 20233 MSG1
3021 6155 JSR @PRINT ; "DRIVE TEST WITH A DRIVE"
3022 20255 MSG2
3023 6155 JSR @PRINT ; "TESTER TEST WITH DRIVE EMULATOR"
3024 20275 MSG3
3025 2401 JMP @. +1 ; RETURN
3026 505 PROG
```

; TEST COMMAND ENTRY

```
3027 30412 HLP2: LDA 2, HCL ; MESSAGE LIST
3030 21000 LDA 0, 0, 2
3031 101005 MOV 0, 0, SNR ; END OF LIST?
3032 2406 JMP @HLP2R ; YES, EXIT
3033 40402 STA 0, . +2 ; NO, SET FOR DISPLAY
3034 6155 JSR @PRINT
3035 0 0
3036 151400 INC 2, 2
3037 771 JMP HLP2+1
```

3040 1146 HLP2R: ERR3A

```
3041 3042 HCL: . +1
3042 20320 HLNO
3043 20334 HC0
3044 20350 HC1
3045 20366 HC2
3046 20412 HC3
3047 20430 HC4
3050 20446 HC5
3051 20463 HC6
3052 20473 HC7
3053 20511 HC8
3054 20526 HC9
3055 20543 HC10
3056 20560 HC11
3057 0 0
```

; DRIVE NAME ENTRY

```
3060 6403 HLP3: JSR @DSPLA ; CALL LIST DRIVE DIRECTORY
3061 2401 JMP @. +1
3062 601 DRIVE

3063 17415 DSPLA: DSPL
```

;\*\*\*\* TEST PROGRAM DIRECTORY \*\*\*\*

;ENTRY FORMAT:

; CHARACTER COUNT OF PROGRAM NAME PLUS NULLS  
; TEXT ADDRESS OF PROGRAM NAME  
; ENTRY POINTER TO SET PROGRAM FLAG

3064	5	PDIR: 5	
3065	20575	PNAM1	; "LOGIC"
3066	524	PRG2	
3067	5	5	
3070	20600	PNAM2	; "DRIVE"
3071	526	PRG2A	
3072	6	6	
3073	20603	PNAM3	; "TESTER"
3074	530	PRG2B	
3075	1	1	
3076	20611	CHL	
3077	524	PRG2	
3100	1	1	
3101	20607	CHD	
3102	526	PRG2A	
3103	1	1	
3104	20610	CHT	
3105	530	PRG2B	
3106	0	0	

;\*\*\*\* PROGRAM COMMAND LISTS \*\*\*\*

;DIAGNOSTIC COMMANDS

3107	1	CMDL: 1	
3110	20611	CHL	
3111	1205	LOOP	
3112	1	1	
3113	20612	CHP	
3114	1212	PROCD	
3115	1	1	
3116	20613	CHR	
3117	1213	REPT	
3120	1	1	
3121	20614	CHS	
3122	1220	SKPT	
3123	3	3	
3124	20615	CHWB	
3125	1240	DWBF	
3126	3	3	
3127	20617	CHRB	
3130	1242	DRBF	

3131	1	1	
3132	20621	CHX	
3133	2702	QUIT	
3134	1	1	
3135	20622	CHH	
3136	3027	HLP2	
3137	1	1	
3140	20623	CHC	
3141	1221	OPTN	
3142	1	1	
3143	20626	CHJ	
3144	2703	GO	
3145	4	4	
3146	20627	CDDA	
3147	1310	DAC	
3150	4	4	
3151	20632	CDDC	
3152	1432	DCC	
3153	4	4	
3154	20635	CDIA	
3155	1464	DAS	
3156	4	4	
3157	20640	CDIB	
3160	1526	DBST	
3161	0	0	
3162	1	CMDS0: 1	
3163	20622	CHH	; "H"
3164	3013	HLPO	
3165	0	0	
3166	1	CMDS1: 1	
3167	20622	CHH	; "H"
3170	3017	HLP1	
3171	1	1	
3172	20621	CHX	; "X"
3173	2702	QUIT	
3174	0	0	
3175	1	CMDS2: 1	
3176	20622	CHH	
3177	3027	HLP2	
3200	1	1	
3201	20621	CHX	
3202	2702	QUIT	

3203 0 0  
3204 1 CMDS3: 1  
3205 20622 CHH  
3206 3060 HLP3  
3207 1 1  
3210 20621 CHX  
3211 2702 QUIT  
3212 0 0

;\*\*\*\* YES OR NO INPUT LIST \*\*\*\*

3213 1 NYL: 1  
3214 20624 CHY ; "Y"  
3215 541 PRG3A  
3216 1 1  
3217 20625 CHN ; "N"  
3220 543 PRG3B  
3221 0 0

;\*\*\*\* INPUT CHARACTER BUFFER \*\*\*\*

3222 20 CBUFF: .BLK 20

;\*\*\*\* TEMPORARY STORAGE AREA \*\*\*\*

3242 20 TEMP: .BLK 20

; LOTUS 700 DISK CONTROLLER DIAGNOSTICS  
; VERSION 1.7 JAN 31, 1981

; CONTROLLER LOGIC TEST SECTION

```
; * * * * *  
; *  
; *          DDLG          *  
; *  
; * * * * *
```

; \*\*\*\* CONTROLLER COMMAND EQUATES \*\*\*\*

0	READ = 0000	; READ DATA
200	RCAL = 0200	; RECAL
400	SEEK = 0400	; SEEK CYLINDER
600	WHDR = 0600	; WRITE HEADER FROM MEMORY
1400	FMAT = 1400	; FORMAT FROM CONTROL
1600	RLSD = 1600	; RELEASE DRIVE
2000	TRES = 2000	; TRESPASS
2200	ALT1 = 2200	; ALT MODE 1
2600	ALT2 = 2600	; ALT MODE 2
2600	NOOP = 2600	; NO OP
3000	DATV = 3000	; DATA VERIFY
3200	RDBF = 3200	; READ BUFFERS
3400	WRITE = 3400	; WRITE DATA
3600	RHDR = 3600	; READ HEADER

; \*\*\*\* CONTROLLER STATUS EQUATES \*\*\*\*

40000	RWDN = 40000	; RW DONE
400	ISCT = 400	; ILLEGAL SECTOR ADDR
200	ECCE = 200	; ECC ERROR
100	BSCT = 100	; BAD SECTOR
40	CADR = 40	; CYLINDER ADDR ERROR
20	SSAE = 20	; SURF-SECTOR ADDR ERROR
10	VRFE = 10	; VERIFY ERROR
4	RWTD = 4	; RW TIMEOUT
1	RWFT = 1	; RW FAULT

; \*\*\*\* DRIVE STATUS EQUATES \*\*\*\*

70	DRFL = 70	; DRIVE FAULT
100	ILCMD = 100	; ILLEGAL COMMAND
200	ISKA = 200	; ILLEGAL SEEK ADDRESS
1000	WPRT = 1000	; WRITE PROTECTED
4000	DBSY = 4000	; DRIVE BUSY
10000	DRDY = 10000	; DRIVE READY
40000	RESV = 40000	; DRIVE RESERVED

; \*\*\*\* DRIVE SELECT EQUATES \*\*\*\*

0	DRV0 = 0
40	DRV1 = 40
100	DRV2 = 100
140	DRV3 = 140

; \*\*\*\* ERROR MESSAGE BLOCK INDEXING EQUATES \*\*\*\*

6  
14  
22  
30  
36  
44  
52  
60  
66  
74  
102  
110

ONE = 6  
TWO = 14  
THREE = 22  
FOUR = 30  
FIVE = 36  
SIX = 44  
SEVEN = 52  
EIGHT = 60  
NINEE = 66  
TEN = 74  
ELEVN = 102  
TWELF = 110

3300 . LOC 3300

;\*\*\*\*\* REGISTER RESET TEST \*\*\*\*\*

; CHECK THE STATES OF VARIOUS REGISTERS AFTER A MASTER RESET

3300 6177 RGRS: JSR @SET ; SET UP TEST  
3301 24162 RRST  
3302 3305 RSTO  
3303 3443 RGWR  
3304 177777 -1

; ISSUE MASTER RESET

3305 62677 RSTO: IORST

; CHECK BUSY RESET

3306 4272 RST1: JSR SKBN ; BUSY = 0?  
3307 403 JMP RST2 ; YES  
3310 6166 JSR @ERROR ; NO  
3311 3363 RSEB

; CHECK DONE RESET

3312 4261 RST2: JSR SKDN ; DONE = 0?  
3313 403 JMP RST3 ; YES  
3314 6166 JSR @ERROR ; = 1  
3315 3371 RSEB+ONE

; CHECK DIA = 0

3316 4234 RST3: JSR DDIA ; READ CONTENTS  
3317 101005 MOV O, O, SNR ; RESET?  
3320 412 JMP RST4 ; IT DID  
3321 6166 JSR @ERROR ; HAS GARBAGE  
3322 3377 RSEB+TWO

; RESET FAILED, TRY CLEAR ATTENTION COMMAND

3323 20105 RST3A: LDA O, CATT  
3324 4211 JSR DDOA ; ISSUE CLEAR ATTEN  
3325 4234 JSR DDIA ; READ IT AGAIN  
3326 101005 MOV O, O, SNR ; ALL CLEARED?  
3327 403 JMP RST4 ; YES  
3330 6166 JSR @ERROR ; DID NOT WORK  
3331 3405 RSEB+THREE

; CHECK DIB = 0

3332 4242 RST4: JSR DDIB ; READ REG  
3333 101005 MOV O, O, SNR ; ALL ZEROS?  
3334 412 JMP RST5 ; ALL ZEROS  
3335 6166 JSR @ERROR ; NOT SO  
3336 3413 RSEB+FOUR

; RESET FAILED, TRY CLEAR ATTENTION

3337 20105 RST4A: LDA O, CATT

```

3340 4211 JSR DDOA ; ISSUE CLEAR ATTEN
3341 4242 JSR DDIB ; READ AGAIN
3342 101005 MOV O, O, SNR ; CLEARED?
3343 403 JMP RST5 ; ALL GONE
3344 6166 JSR @ERROR ; STILL NG
3345 3421 RSEB+FIVE

```

; CHECK DIC = 0

```

3346 4250 RST5: JSR DDIC ; READ CONTENTS
3347 101005 MOV O, O, SNR ; ANYTHING?
3350 2114 JMP @NTST ; ALL ZEROS, NEXT TEST
3351 6166 JSR @ERROR ; SOMETHING THERE
3352 3427 RSEB+SIX

```

; TRY AGAIN WITH CLEAR ATTENTION

```

3353 20105 RST5A: LDA O, CATT
3354 4211 JSR DDOA ; CLEAR ATTEN
3355 4250 JSR DDIC ; READ IT
3356 101005 MOV O, O, SNR ; OK?
3357 2114 JMP @NTST ; YES, EXIT
3360 6166 JSR @ERROR ; NO WORKIE
3361 3435 RSEB+SEVEN
3362 2114 JMP @NTST

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

3363 3306 RSEB: RST1 ; ERROR ADDR
3364 3305 RST0 ; LOOP ADDR
3365 21164 BSYF ; TEST SUBJECT TEXT
3366 121155 @MRST ; INPUT
3367 121264 @NRST ; OUTPUT
3370 121172 @RSET ; NORMAL

```

```

3371 3312 RST2 ; 1
3372 3305 RST0
3373 21177 DNEF
3374 121155 @MRST
3375 121264 @NRST
3376 121172 @RSET

```

```

3377 3316 RST3 ; 2
3400 3305 RST0
3401 21205 CTRG
3402 121155 @MRST
3403 1 1
3404 121445 @DATI3

```

```

3405 3323 RST3A ; 3
3406 3305 RST0
3407 21205 CTRG
3410 1 1
3411 1 1
3412 121445 @DATI3

```

```

3413 3332 RST4 ; 4
3414 3305 RST0
3415 21251 DRRG

```



3416	121155	@MRST
3417	2	2
3420	121475	@DATI7
3421	3337	RST4A ; 5
3422	3305	RSTO
3423	21251	DRRG
3424	1	1
3425	2	2
3426	121475	@DATI7
3427	3346	RST5 ; 6
3430	3305	RSTO
3431	21271	DARG
3432	121155	@MRST
3433	3	3
3434	121467	@DATI6
3435	3353	RST5A ; 7
3436	3305	RSTO
3437	21271	DARG
3440	1	1
3441	3	3
3442	121467	@DATI6

;\*\*\*\*\* WRITE-READ REGISTER TEST \*\*\*\*\*

; ADDRESS REGISTER WITH VARIED DEVICE CODES

```
3443 6177 RGWR: JSR @SET
3444 24172      RGWRT
3445 3451      RGDO
3446 3715      CFS
3447 177777    -1
```

; SET ALTERNATE MODE 1, LOAD MEMORY ADDRESS AND DISK ADDRESS REGISTERS  
; WITH ALL ONES DATA

```
3450 2200      ALT1
3451 20777 RGDO: LDA 0, RGDO-1 ; GET ALT MODE 1 COMMAND
3452 4211      JSR DDOA ; ISSUE COMMAND
3453 102000    ADC 0,0 ; MAKE ALL ONES
3454 4220      JSR DDOB ; LOAD REGISTERS
3455 4225      JSR DDOC
```

; READ CONTENTS OF REGISTERS WITH STANDARD DEVICE CODE 27

```
3456 4234 RGD1: JSR DDIA ; READ DATA
3457 126000    ADC 1,1 ; MAKE 177777
3460 122415    SUB# 1,0, SNR ; ALL ONES?
3461 403       JMP RGD1A ; IT IS
3462 6166      JSR @ERROR ; IT IS NOT
3463 3621      RGDEB
3464 4250 RGD1A: JSR DDIC ; READ THIS ONE
3465 126000    ADC 1,1
3466 125220    MOVZR 1,1 ; MAKE 077777
3467 106415    SUB# 0,1, SNR ; MATCH?
3470 403       JMP RGD2 ; YES
3471 6166      JSR @ERROR ; NO, ERROR
3472 3627      RGDEB+ONE
```

; ADDRESS REGISTERS WITH OTHER DEVICE CODES AND SEE IF THEY RESPOND

```
3473 24517 RGD2: LDA 1, DCL
3474 44142    STA 1, SAVP ; SET DEV CODE POINTER
3475 22142    LDA 0, @SAVP ; GET DEV CODE
3476 101005    MOV 0,0, SNR ; END OF LIST?
3477 430      JMP RGD3 ; YES, EXIT
```

; INSERT TEST DEVICE CODE WITH THE BASIC DIA AND DIC INSTRUCTIONS AND  
; SET THEM FOR EXECUTION

```
3500 24506    LDA 1, TDIA ; GET DIA INSTRUCTION
3501 107000    ADD 0,1 ; INSERT DC
3502 44407    STA 1, RGD2A ; SET IT FOR EXECUTION
3503 24504    LDA 1, TDIC ; DIC INSTRUCTION
3504 107000    ADD 0,1
3505 44412    STA 1, RGD2B ; SET IT
3506 6200     JSR @SADC ; GO CONVERT TO ASCII
3507 46501    STA 1, @DCXA ; SET IN TEXT
3510 46501    STA 1, @DCXB
```

; TEST DIA = 0

```

3511      0 RGD2A: 0      ; DIA 0, DC
3512 101005      MOV      0, 0, SNR ; ALL ZEROS?
3513      404      JMP      RGD2B ; YES
3514 40301      STA      0, DIA1 ; SAVE CONTENTS
3515      6166     JSR      @ERROR ; NO
3516      3635     RGDEB+TWO

```

; TEST DIC = 0

```

3517      0 RGD2B: 0      ; DIC 0, DC
3520 101005      MOV      0, 0, SNR ; WHAT'S IN THERE?
3521      404      JMP      RGD2C ; NOTHING!
3522 40304      STA      0, DICO ; SOMETHING ELSE
3523      6166     JSR      @ERROR
3524      3643     RGDEB+THREE
3525 10142 RGD2C: ISZ      SAVP ; BUMP LIST POINTER
3526      747     JMP      RGD2+2 ; CONTINUE

```

; TEST A NIOC WILL NOT CLEAR REGISTERS

```

3527      4257 RGD3: JSR      NNIOC ; ISSUE CLEAR
3530      4234     JSR      DDIA ; READ REG
3531 126000      ADC      1, 1
3532 106415      SUB#     0, 1, SNR ; STILL THE SAME?
3533      403     JMP      RGD3A ; YES
3534      6166     JSR      @ERROR ; NO, CHANGED
3535      3651     RGDEB+FOUR
3536 126000 RGD3A: ADC      1, 1
3537 125220      MOVZR   1, 1 ; MAKE 077777
3540      4250     JSR      DDIC ; READ REG
3541 106415      SUB#     0, 1, SNR ; AGREE?
3542      403     JMP      RGD3B ; THEY DO
3543      6166     JSR      @ERROR ; THEY DON'T
3544      3657     RGDEB+FIVE

```

; TEST ISSUING DOA ALL ZEROS CLEARS DIA BUT NOT DIC

```

3545 102400 RGD3B: SUB      0, 0
3546      4211     JSR      DDOA
3547      4234     JSR      DDIA ; READ REG
3550 101005      MOV      0, 0, SNR ; ALL CLEARED?
3551      410     JMP      RGD3D ; YES
3552 20105      LDA      0, CATT
3553      4211     JSR      DDOA ; NO, TRY CLEAR ATTEN
3554      4234 RGD3C: JSR      DDIA ; READ AGAIN
3555 101005      MOV      0, 0, SNR ; WORKED?
3556      403     JMP      RGD3D ; YEH
3557      6166     JSR      @ERROR ; DARN IT!
3560      3665     RGDEB+SIX
3561      4250 RGD3D: JSR      DDIC
3562 126000      ADC      1, 1
3563 125220      MOVZR   1, 1 ; 077777
3564 106405      SUB      0, 1, SNR ; DID IT CLEAR?
3565      403     JMP      RGD4 ; NO, WONDERFUL
3566      6166     JSR      @ERROR ; SORRY
3567      3673     RGDEB+SEVEN

```

; NOW ISSUE A MASTER RESET TO RESET EVERYBODY

```

3570 62677 RGD4: IORST
3571 20657 LDA 0,RGD0-1 ;GET ALT MODE 1 COMMAND
3572 4211 JSR DDOA ;SET IT
3573 4234 RGD4A: JSR DDIA ;READ
3574 101005 MOV 0,0,SNR ;RESET?
3575 403 JMP RGD4B ;YES
3576 6166 JSR @ERROR ;IT DID NOT
3577 3701 RGDEB+EIGHT
3600 4250 RGD4B: JSR DDIC
3601 101005 MOV 0,0,SNR ;HOW ABOUT THIS GUY?
3602 2114 JMP @NTST ;ALL CLEARED, EXIT
3603 6166 JSR @ERROR ;PROBLEM
3604 3707 RGDEB+NINEE
3605 2114 JMP @NTST

3606 60400 TDIA: DIA 0,0
3607 62400 TDIC: DIC 0,0
3610 21356 DCXA: AM1C+11
3611 21372 DCXB: AM1D+12

```

TEST DEVICE CODE LIST

```

3612 3613 DCL: .+1
3613 7 7
3614 23 23
3615 26 26
3616 37 37
3617 67 67
3620 0 0

```

\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

3621 3456 RGDEB: RGD1
3622 3451 RGDO
3623 21317 AM1A
3624 6 6
3625 4 4
3626 121453 @DATI4

3627 3464 RGD1A ; 1
3630 3451 RGDO
3631 21332 AM1B
3632 6 6
3633 3 3
3634 121461 @DATI5

3635 3511 RGD2A ; 2
3636 3451 RGDO
3637 21345 AM1C
3640 6 6
3641 4 4
3642 121445 @DATI3

3643 3517 RGD2B ; 3
3644 3451 RGDO
3645 21360 AM1D
3646 6 6
3647 3 3
3650 121467 @DATI6

```

3651	3527	RGD3	; 4
3652	3527	RGD3	
3653	21577	RDUN	
3654	121620	@MNIOC	
3655	1	1	
3656	121453	@DATI4	
3657	3536	RGD3A	; 5
3660	3527	RGD3	
3661	21577	RDUN	
3662	121620	@MNIOC	
3663	3	3	
3664	121461	@DATI5	
3665	3554	RGD3C	; 6
3666	3545	RGD3B	
3667	21624	CLAT	
3670	1	1	
3671	1	1	
3672	121445	@DATI3	
3673	3561	RGD3D	; 7
3674	3545	RGD3B	
3675	21641	RDUD	
3676	1	1	
3677	3	3	
3700	121461	@DATI5	
3701	3573	RGD4A	; 8
3702	3451	RGDO	
3703	21317	AM1A	
3704	121155	@MRST	
3705	4	4	
3706	121445	@DATI3	
3707	3600	RGD4B	; 9
3710	3451	RGDO	
3711	21332	AM1B	
3712	121155	@MRST	
3713	3	3	
3714	121467	@DATI6	

;\*\*\*\* CONTROL FULL STATUS TEST \*\*\*\*

; ISSUE A DRIVE COMMAND TO CAUSE CONTROL FULL STATUS TO SET. AFTER A SHORT  
; DELAY, IT RESETS

```
3715 6177 CFS: JSR @SET
3716 24204 CFST
3717 3723 CFSO
3720 3767 DRDN
3721 177777 -1
```

; ISSUE A RELEASE DRIVE COMMAND TO INITIATE CONTROL FULL AND CHECK STATUS

```
3722 1600 RLSD
3723 20777 CFS0: LDA 0,-1 ; GET COMMAND
3724 24106 LDA 1,CATTS
3725 123000 ADD 1,0 ; ADD CLEAR ATTEN
3726 40275 STA 0,DOAO ; SAVE COMMAND
3727 61300 DGAP 0,0 ; ISSUE IT
3730 60400 DIA 0,0 ; READ STATUS
3731 24070 LDA 1,MSKA
3732 123400 AND 1,0 ; MASK OUT UNWANTED BITS
3733 126620 SUBZR 1,1 ; MAKE 100000
3734 106415 CFS1: SUB# 0,1,SNR ; CF DN?
3735 403 JMP CFS1A ; YES
3736 6166 JSR @ERROR ; NO
3737 3753 CFSEB
```

; SEE IF CR DROPS TO ZERO AFTER ABOUT 10US

```
3740 24054 CFS1A: LDA 1,K7
3741 124400 NEG 1,1
3742 125404 INC 1,1,SZR ; DELAY
3743 777 JMP -1
3744 4234 JSR DDIA ; READ STATUS
3745 24070 LDA 1,MSKA
3746 107415 CFS2: AND# 0,1,SNR ; CF DROPPED?
3747 2114 JMP @NTST ; IT DID
3750 6166 JSR @ERROR ; STILL UP
3751 3761 CFSEB+ONE
3752 2114 JMP @NTST
```

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```
3753 3734 CFSEB: CFS1
3754 3723 CFSO
3755 21373 CFON
3756 7 7
3757 1 1
3760 121431 @DATI1

3761 3746 CFS2
3762 3723 CFSO
3763 21405 CFOF
3764 7 7
3765 1 1
3766 121437 @DATI2
```

;\*\*\*\* SET & RESET DRIVE DONE STATUS TEST \*\*\*\*

; DRIVE DONE STATUS IS TESTED ON ALL 4 PORTS OF CONTROLLER

```
3767 6177 DRDN: JSR @SET
3770 24216 DRDNT
3771 3774 DRD
3772 4160 CLDD
3773 0 0
```

; SET UP IO COMMAND LIST POINTER AND ISSUE COMMANDS FROM THE LIST

```
3774 30472 DRD: LDA 2, DDNC ; GET COMMAND LIST
3775 21000 LDA 0, 0, 2 ; FETCH COMMAND
3776 101005 MOV 0, 0, SNR ; END OF LIST?
3777 2114 JMP @NTST ; YES, NEXT TEST
4000 4211 JSR DDOA ; SELECT DRIVE & CLEAR DRV ATTEN
4001 20071 LDA 0, CMAX ; DUMMY CYLINDER ADDR
4002 4232 JSR DDOCP ; START SEEK
4003 34054 LDA 3, K7
4004 174400 NEG 3, 3
4005 175404 INC 3, 3, SZR ; DELAY 10US
4006 777 JMP -1
4007 4264 DRD1: JSR SKDZ ; DONE = 1?
4010 403 JMP DRD2 ; YES
4011 6166 JSR @ERROR ; NOT SO
4012 4114 DRDEB
```

; PRESET DRIVE NUMBER IN ERROR MESSAGE AND READ STATUS

```
4013 25004 DRD2: LDA 1, 4, 2
4014 44512 STA 1, DRDEB+12; SET DRV #
4015 4234 JSR DDIA ; GET STATUS
4016 34070 LDA 3, MSKA
4017 163400 AND 3, 0 ; PICK OUT THE BITS
4020 25001 LDA 1, 1, 2 ; GET REFERENCE
4021 106415 SUB# 0, 1, SNR ; OK?
4022 404 JMP DRD3 ; PERFECT
4023 44305 STA 1, DIAN ; SCREWED UP!. SAVE REF
4024 6166 JSR @ERROR
4025 4122 DRDEB+ONE
```

; CLEAR ALL OTHER DRIVE ATTENTION EXCEPT ITSELF

```
4026 21002 DRD3: LDA 0, 2, 2 ; CLEAR ATTEN BITS
4027 4211 JSR DDOA ; SEND THEM OUT
4030 4234 JSR DDIA ; READ STATUS
4031 34070 LDA 3, MSKA
4032 163400 AND 3, 0 ; PICK OFF BITS
4033 25001 LDA 1, 1, 2 ; GET REFERENCE
4034 44305 STA 1, DIAN ; SET REF
4035 106415 DRD3A: SUB# 0, 1, SNR ; CLEAR THE RIGHT ONES?
4036 405 JMP DRD3B ; IT DID
4037 25004 LDA 1, 4, 2
4040 44474 STA 1, DRDEB+20; NO, SET MESSAGE
4041 6166 JSR @ERROR
4042 4130 DRDEB+TWO
4043 4264 DRD3B: JSR SKDZ ; DONE STILL SET?
4044 403 JMP DRD4 ; SET
```

4045 6166 JSR @ERROR ; NO  
4046 4136 DRDEB+THREE

; NOW CLEAR ITS OWN DRIVE ATTENTION AND DONE FLAG

4047 21003 DRD4: LDA 0,3,2 ; GET CLEAR BIT  
4050 4211 JSR DDOA ; ISSUE IT  
4051 4234 JSR DDIA ; CHECK STATUS  
4052 24070 LDA 1,MSKA  
4053 123415 DRD4A: AND# 1,0,SNR ; CLEARED?  
4054 403 JMP DRD4B ; YES  
4055 6166 JSR @ERROR ; NO  
4056 4144 DRDEB+FOUR  
4057 4261 DRD4B: JSR SKDN ; HOW ABOUT DONE?  
4060 403 JMP DRD5 ; RESET  
4061 6166 JSR @ERROR  
4062 4152 DRDEB+FIVE  
4063 24053 DRD5: LDA 1,K5  
4064 133000 ADD 1,2 ; NEXT DRIVE TEST  
4065 710 JMP DRD+1 ; CONTINUE

; DRIVES 0-3 COMMANDS AND REFERENCE PARAMETERS LIST

4066 4067 DDNC: . +1  
4067 74400 74400 ; DOA DR 0  
4070 20000 20000 ; DIA  
4071 134000 134000 ; DOA, CLEAR ATTEN  
4072 40000 40000 ; DOA, CLEAR SELF  
4073 121517 @DATNO  
4074 74440 74440 ; DR 1  
4075 10000 10000  
4076 154000 154000  
4077 20000 20000  
4100 121525 @DATN1  
4101 74500 74500 ; DR 2  
4102 4000 4000  
4103 164000 164000  
4104 10000 10000  
4105 121533 @DATN2  
4106 74540 74540 ; DR 3  
4107 2000 2000  
4110 170000 170000  
4111 4000 4000  
4112 121541 @DATN3  
4113 0 0 ; END

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

4114 4007 DRDEB: DRD1  
4115 3775 DRD+1  
4116 21675 DNP  
4117 10 10  
4120 121661 @DNE0  
4121 121667 @DNE1  
  
4122 4013 DRD2 ; 1  
4123 3775 DRD+1  
4124 21710 DRAT  
4125 10 10



```

---
4126      1      1
4127     15     15

4130     4035    DRD3A ; 2
4131     4026    DRD3
4132    21722    CLOD
4133      1      1
4134      1      1
4135     15     15

4136     4043    DRD3B ; 3
4137     4026    DRD3
4140    21722    CLOD
4141      1      1
4142  121661    @DNE0
4143  121667    @DNE1

4144     4053    DRD4A ; 4
4145     4047    DRD4
4146    21736    CLDA
4147      1      1
4150      1      1
4151  121437    @DATI2

4152     4057    DRD4B ; 5
4153     4047    DRD4
4154    21747    CLDN
4155      1      1
4156  121667    @DNE1
4157  121661    @DNE0

```

;\*\*\*\* CLEAR DRIVE DONE WITH NIOC \*\*\*\*

;SET DRIVE DONE (ATTENTION) IN EACH DRIVE THEN CLEAR THEM ALL WITH  
;A NIOC INSTRUCTION

```
4160 6177 CLDD: JSR @SET
4161 24231 CLDDT
4162 4165 CLD
4163 4257 XTMA
4164 0 0
```

; ISSUE A SEEK COMMAND TO SET DRIVE DONE ONE DRIVE AT A TIME. COMMAND  
; AND DRIVE NUMBER WORD IS FETCHED FROM THE TEST LIST

```
4165 30440 CLD: LDA 2, DCL1 ;GET TEST LIST POINTER
4166 21000 LDA 0, 0, 2 ;FETCH COMMAND + DRIVE WORD
4167 101005 MOV 0, 0, SNR ;END OF LIST?
4170 424 JMP CLD2 ;YES, ALL SET. GO CLEAR THEM
4171 4211 JSR DDOA ;ISSUE RECAL
4172 20071 LDA 0, CMAX
4173 4232 JSR DDOCP ;SEND DUMMY ADDR AND START
```

; READ STATUS AND WAIT FOR CONTROL FULL TO DROP, CHECK DONE BITS

```
4174 4234 JSR DDIA ;GET STATUS
4175 101112 MOVL# 0, 0, SZC ;CF = 0?
4176 776 JMP -2 ;NO, LOOP TILL
4177 24070 LDA 1, MSKA
4200 123400 AND 1, 0 ;CF = 0, PICK OFF BITS
4201 25001 LDA 1, 1, 2 ;GET REFERENCE DONE BITS
4202 122415 CLD1: SUB# 1, 0, SNR ;DONE OK?
4203 406 JMP CLD1A ;OK
4204 44305 STA 1, DIAN ;SAVE REF
4205 35002 LDA 3, 2, 2
4206 54442 STA 3, CLDEB+5; NOT OK, SET ERROR MESSAGE
4207 6166 JSR @ERROR
4210 4243 CLDEB
```

; BUMP TEST LIST TO SET NEXT DONE BIT

```
4211 24051 CLD1A: LDA 1, K3
4212 133000 ADD 1, 2
4213 753 JMP CLD+1 ;LOOP BACK
```

; ALL DRIVE DONE BITS ARE SET, NOW ISSUE NIOC TO CLEAR THEM

```
4214 4257 CLD2: JSR NNIOC
4215 4234 JSR DDIA ;READ STATUS
4216 101005 MOV 0, 0, SNR ;ALL CLEARED?
4217 2114 JMP @NTST ;YES, EXIT
4220 35002 LDA 3, 2, 2
4221 54434 STA 3, CLDEB+12; NOT SO. SET ERROR MESSAGE
4222 6166 JSR @ERROR
4223 4251 CLDEB+ONE
4224 2114 JMP @NTST
```

; DRIVES 0-3 COMMANDS AND TEST REFERENCE LIST

```
4225 4226 DCL1: .+1
```

```

4226 400 400 ; DOA, SEEK DR 0
4227 20000 20000 ; DIA, DONE DR 0
4230 121517 @DATNO ; DR #
4231 440 440 ; DOA, DR 1
4232 30000 30000
4233 121547 @DATN4
4234 500 500 ; DOA, DR 2
4235 34000 34000
4236 121555 @DATN5
4237 540 540 ; DOA, DR 3
4240 36000 36000
4241 121563 @DATN6
4242 0 0 ; END

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

4243 4202 CLDEB: CLD1
4244 4166 CLD+1
4245 21710 DRAT
4246 10 10
4247 1 1
4250 15 15

4251 4216 CLD2+2
4252 4214 CLD2
4253 21736 CLDA
4254 121620 @MNI0C
4255 1 1
4256 121445 @DATI3

```

;\*\*\*\* EXTENDED MEMORY ADDRESSING TEST \*\*\*\*

; IF THE EXTENDED MEMORY ADDRESSING LOGIC HAS BEEN IMPLEMENTED IN THE  
; CONTROLLER, THEN PROCEED WITH THE TEST. IF NOT, SKIP TO NEXT TEST

```
4257 6177 XTMA: JSR @SET
4260 24242 XTMA: XTMA
4261 4265 XTMA: XTM
4262 4406 XTMA: MAR
4263 17777 XTMA: -1
```

; TEST TO SEE IF EXTENDED MEMORY ADDRESSING LOGIC IS THERE

```
4264 2200 ALT1: LDA 2,DCL2 ; LOAD TEST PARAMETR LIST
4265 30461 XTMA: LDA 0,XTM-1
4266 20776 XTMA: JSR DDOA ; SET ALT MODE 1
4267 4211 XTMA: SUB 0,0
4270 102400 XTMA: JSR DDOB ; MA = 0
4271 4220 XTMA: JSR DDOC
4272 4225 XTMA: JSR DDIB ; READ EXT'D MEM ADDR REG
4273 4242 XTMA: LDA 1,K17
4274 24056 XTMA: SUB# 0,1,SNR ; LOGIC IMPLEMENTED?
4275 106415 XTMA: JMP @NTST ; NO, SKIP TEST
4276 2114 XTMA:
```

; PROCEED WITH THE TEST.

; WRITE EXT'D MEMORY BIT TO THE REGISTER. GET TEST BITS FROM TEST LIST

```
4277 20765 XTMA: LDA 0,XTM-1 ; ALT MODE 1 COMMAND
4300 25000 XTMA: LDA 1,0,2 ; TEST BITS
4301 125005 XTMA: MOV 1,1,SNR ; END OF LIST?
4302 417 XTMA: JMP XTM2 ; YES, EXIT
4303 123000 XTMA: ADD 1,0 ; FORM ALT MODE 1 + XMA
4304 4211 XTMA: JSR DDOA ; LOAD IT
4305 102400 XTMA: SUB 0,0
4306 4220 XTMA: JSR DDOB ; MA = 0
```

; READ EXT'D MEMORY ADDRESS REGISTER AND COMPARE RESULTS

```
4307 4242 XTMA: JSR DDIB
4310 122415 XTMA: SUB# 1,0,SNR ; AGREE WITH INPUT?
4311 405 XTMA: JMP XTM1B ; AGREE
4312 35001 XTMA: LDA 3,1,2
4313 56432 XTMA: STA 3,@DTN7A ; NO, SET ERROR MESSAGE
4314 6166 XTMA: JSR @ERROR
4315 4372 XTMA: XTMEB
```

; BUMP TEST LIST POINTER THEN CONTINUE

```
4316 151400 XTMA: INC 2,2
4317 151400 XTMA: INC 2,2
4320 757 XTMA: JMP XTM1
```

; NOW RELOAD MA REGISTER WITH ALL ONES AND TEST AGAIN

```
4321 30437 XTMA: LDA 2,DCL3 ; GET TEST PARAMETER LIST
4322 20742 XTMA: LDA 0,XTM-1 ; ALT MODE 1 COMMAND
4323 25000 XTMA: LDA 1,0,2 ; TEST BITS
4324 125005 XTMA: MOV 1,1,SNR ; END?
```

```

4325 2114      JMP    @NTST    ; YES, DONE
4326 123000    ADD    1,0
4327 4211      JSR    DDOA     ; ISSUE COMMAND
4330 102000    ADC    0,0      ; MAKE ALL ONES
4331 4220      JSR    DDOB     ; MA = 177777
4332 4225      JSR    DDOC

```

; READ REGISTER AND COMPARE CONTENTS

```

4333 4242 XTM2A: JSR    DDIB
4334 122415 SUB#   1,0,SNR ; HOW'S IT LOOK?
4335 405     JMP    XTM2B  ; LOOKS GOOD
4336 35001   LDA    3,1,2
4337 56406   STA    3,@DTN7A ; LOOKS BAD, SET MESSAGE
4340 6166    JSR    @ERROR
4341 4400    XTMEB+ONE

```

; BUMP LIST POINTER AND KEEP GOING

```

4342 151400 XTM2B: INC    2,2
4343 151400     INC    2,2
4344 756     JMP    XTM2+1

```

```
4345 21576 DTN7A: DATN7+5
```

; EXTENDED MEMORY ADDRESS TEST BITS AND ERROR MESSAGE LIST

```

4346 4347 DCL2:  .+1
4347 1      1
4350 30061    30061
4351 2        2
4352 30062    30062
4353 4        4
4354 30064    30064
4355 10       10
4356 30460    30460
4357 0        0

```

```

4360 4361 DCL3:  .+1
4361 16      16
4362 30466    30466
4363 15      15
4364 30465    30465
4365 13      13
4366 30463    30463
4367 7        7
4370 30067    30067
4371 0        0

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

4372 4307 XTMEB: XTM1A
4373 4277     XTM1
4374 21772    RXMA
4375 6        6
4376 5        5
4377 121571   @DATN7

4400 4333     XTM2A

```

---  
4401 4321 XTM2  
4402 21772 RXMA  
4403 6 6  
4404 5 5  
4405 121571 @DATN7

;\*\*\*\* MEMORY ADDRESS REGISTER WRITE-READ TEST \*\*\*\*

; EACH BIT IN THE MEMORY ADDRESS REGISTER IS TESTED INDIVIDUALLY BY  
; WRITING AND READING A SINGLE BIT

```
4406 6177 MAR: JSR @SET
4407 24260 MART
4410 4414 MARO
4411 4524 SSR
4412 177777 -1
```

; WRITE A SINGLE '1' TO THE MEMORY REGISTER. AFTER THE FIRST WRITE,  
; LEFT SHIFT THE '1' BEFORE EACH WRITE

```
4413 2200 ALT1
4414 24457 MARO: LDA 1,K16
4415 44152 STA 1,CNTR ; SET SHIFT COUNT
4416 126520 SUBZL 1,1 ; STARTING POSITION
4417 20774 LDA 0,MARO-1
4420 4211 JSR DDOA ; SET ALT MODE 1
4421 102400 SUB 0,0
4422 4225 JSR DDOC
4423 121000 MAR1: MOV 1,0
4424 4220 JSR DDOB ; WRITE TO MA REG
```

; READ CONTENTS OF MEMORY REGISTER AND COMPARE RESULTS

```
4425 4234 MAR2: JSR DDIA
4426 106415 SUB# 0,1,SNR ; MATCH?
4427 403 JMP MAR2A ; PERFECTLY
4430 6166 JSR @ERROR ; DON'T MATCH
4431 4474 MAREB
4432 4250 MAR2A: JSR DDIC ; READ DISK ADDR REG
4433 101005 MOV 0,0,SNR ; ANYTHING?
4434 403 JMP MAR2B ; CLEAN
4435 6166 JSR @ERROR ; YES, SPILL OVER
4436 4502 MAREB+ONE
```

; KEEP WALKING THE '1' IF NOT AT THE END

```
4437 125120 MAR2B: MOVZL 1,1
4440 14152 DSZ CNTR ; ALL THE WAY YET?
4441 762 JMP MAR1 ; NO, LOOP BACK
```

; NOW WRITE A SINGLE '0' TO MEMORY REGISTER AND WALK IT ACROSS

```
4442 24431 MAR3: LDA 1,K16
4443 44152 STA 1,CNTR ; SET WALK COUNT
4444 20747 LDA 0,MARO-1
4445 123000 ADD 1,0
4446 4211 JSR DDOA ; SET ALT MODE 1, EXT'D MEM = 17
4447 102000 ADC 0,0 ; MAKE 177777
4450 4225 JSR DDOC ; LOAD IT
4451 126120 ADCZL 1,1 ; SET STARTING POSITION 177776
4452 121000 MAR3A: MOV 1,0
4453 4220 JSR DDOB ; LOAD MA REG
```

; READ CONTENTS OF MEMORY REGISTER AND COMPARE RESULTS

```

4454 152220 MAR4: ADCZR 2,2 ; MAKE 077777
4455 4234 JSR DDIA ; READ MA REG
4456 106415 SUB# 0,1,SNR ; GOOD BITS?
4457 403 JMP MAR4A ; YES
4460 6166 JSR @ERROR ; NO, BAD
4461 4510 MAREB+TWO
4462 4250 MAR4A: JSR DDIC ; READ THE OTHER ONE
4463 142415 SUB# 2,0,SNR ; DID IT CHANGE?
4464 403 JMP MAR4B ; NO CHANGE
4465 6166 JSR @ERROR ; CHANGED
4466 4516 MAREB+THREE

```

; STEP THE '0' ACROSS IF IT IS NOT AT THE END YET

```

4467 125140 MAR4B: MOVL 1,1
4470 14152 DSZ CNTR ; STOP?
4471 761 JMP MAR3A ; NO, KEEP GOING
4472 2114 JMP @NTST ; YES, NEXT TEST

```

```

4473 16 K16: 16

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

4474 4425 MAREB: MAR2
4475 4423 MAR1
4476 22020 WLK1
4477 6 6
4500 4 4
4501 122007 @ASDOB

4502 4432 MAR2A ; 1
4503 4423 MAR1
4504 21271 DARG
4505 6 6
4506 4 4
4507 121445 @DATI3

4510 4454 MAR4 ; 2
4511 4452 MAR3A
4512 22035 WLK0
4513 6 6
4514 4 4
4515 122007 @ASDOB

4516 4467 MAR4B ; 3
4517 4452 MAR3A
4520 124302 @SSRT
4521 6 6
4522 4 4
4523 121461 @DATI5

```



;\*\*\*\* SURFACE-SECTOR ADDRESS REGISTER WRITE-READ TEST \*\*\*\*

; EACH BIT IN THE SURFACE-SECTOR ADDRESS REGISTER IS TESTED INDIVIDUALLY  
; BY WRITING AND READ A SINGLE BIT

4524	6177	SSR:	JSR	@SET
4525	24302		SSRT	
4526	4531		SSRO	
4527	4616		SRBY	
4530	177777		-1	

; WRITE THE STARTING BIT 000001 TO THE REGISTER. AFTER THAT, LEFT SHIFT  
; IT ONCE BEFORE EACH WRITE

4531	24450	SSRO:	LDA	1,K15	
4532	44152		STA	1,CNTR	; SET SHIFT COUNT
4533	152520		SUBZL	2,2	; MAKE 000001 = TEST BITS
4534	102400		SUB	0,0	
4535	4211		JSR	DDOA	; SET NORMAL MODE
4536	4220		JSR	DDOB	; MA REG = 0
4537	141000		MOV	2,0	
4540	4225		JSR	DDOC	; SURF-SECTOR REG = TEST BITS

; READ CONTENTS OF SURFACE-SECTOR ADDRESS REGISTER AND CHECK IT

4541	4250	SSR1:	JSR	DDIC	
4542	142415		SUB#	2,0,SNR	; SAME AS LOADED
4543	403		JMP	.+3	; IT SURE IS
4544	6166		JSR	@ERROR	; WRONG
4545	4602		SSREB		

; WALK THE '1' ONCE TO THE LEFT IF NOT AT THE END ALREADY

4546	151120		MOVZL	2,2	
4547	14152		DSZ	CNTR	; FINISH?
4550	767		JMP	SSR1-2	; NOT YET

; NOW DO WALKING 0 WRITE AND READ TEST

4551	24722	SSR2:	LDA	1,K16	
4552	44152		STA	1,CNTR	; SET WALK COUNT
4553	152000		ADC	2,2	; MAKE 177777
4554	24056		LDA	1,K17	
4555	102400		SUB	0,0	
4556	123000		ADD	1,0	
4557	4211		JSR	DDOA	; SET NORMAL MODE, EXT'D MA = 17
4560	141000		MOV	2,0	
4561	4220		JSR	DDOB	; MA REG = 177777
4562	151120		MOVZL	2,2	; MAKE 177776 = TEST BITS
4563	141000	SSR2A:	MOV	2,0	
4564	126220		ADCZR	1,1	; MAKE MASK
4565	123400		AND	1,0	; MASK OUT MSB
4566	4225		JSR	DDOC	; SURF-SECT ADDR REG = TEST BITS

; READ CONTENTS OF SURFACE-SECTOR REGISTER AND CHECK IT

4567	4250	SSR2B:	JSR	DDIC	
4570	147400		AND	2,1	; MAKE REF
4571	122415		SUB#	1,0,SNR	; OK?

4572 403 JMP .+3 ; YES  
4573 6166 JSR @ERROR ; NOT OK  
4574 4610 SSREB+ONE

; LEFT SHIFT '0' ONCE, CONTINUE TEST IF NOT ALL ACROSS YET

4575 151140 MOVOL 2,2  
4576 14152 DSZ CNTR ; MORE?  
4577 764 JMP SSR2A ; YES  
4600 2114 JMP @NTST ; NO, FINISHED

4601 15 K15: 15

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

4602 4541 SSREB: SSR1  
4603 4537 SSR1-2  
4604 22020 WLK1  
4605 6 6  
4606 3 3  
4607 122052 @ASDOC

4610 4567 SSR2B  
4611 4563 SSR2A  
4612 22035 WLK0  
4613 6 6  
4614 3 3  
4615 122052 @ASDOC

;\*\*\*\* BUSY FLAG SET AND RESET TEST \*\*\*\*

; TEST THE BUSY FLAG IS SET BY A START COMMAND AND CLEARED OR RESET  
; BY NIOC INSTRUCTION OR AN IO RESET

```
4616 6177 SRBY: JSR @SET
4617 24325 SRBYT
4620 4624 SRBO
4621 4675 RBNT
4622 177777 -1
```

; SET BUSY FLAG BY ISSUING A START WRITE DATA COMMAND

```
4623 3400 WRITE
4624 20777 SRBO: LDA 0,-1 ; GET WRITE COMMAND
4625 24106 LDA 1,CATTS ; CLEAR ATTEN BITS
4626 123000 ADD 1,0
4627 40275 STA 0,DOAO ; SAVE COMMAND
4630 61100 DOAS 0,0 ; ISSUE START COMMAND
```

; TEST BUSY = 1

```
4631 63500 SRB1: SKPBZ 0 ; BUSY SET?
4632 401 JMP SRB1A ; YES, GO CLEAR IT
```

; CLEAR BUSY WITH NIOC

```
4633 4257 SRB1A: JSR NNIOC
4634 4272 JSR SKBN ; BUSY CLEARED?
4635 403 JMP SRB2 ; IT DID
4636 6166 JSR @ERROR ; DID NOT
4637 4661 SRBEB+ONE
```

; SET BUSY FLAG AGAIN

```
4640 20763 SRB2: LDA 0,SRBO-1
4641 24106 LDA 1,CATTS
4642 123000 ADD 1,0 ; FORM WRITE+CLEAR ATTEN COMMAND
4643 40275 STA 0,DOAO ; SAVE COMMAND
4644 4214 JSR DDOAS ; ISSUE START
```

; ISSUE AN IO RESET THEN CHECK FLAG

```
4645 62677 IORST
4646 4272 SRB2A: JSR SKBN ; BUSY RESET?
4647 2114 JMP @NTST ; YES, EXIT
4650 6166 JSR @ERROR ; NO
4651 4667 SRBEB+TWO
4652 2114 JMP @NTST
```

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```
4653 4631 SRBEB: SRB1
4654 4624 SRBO
4655 21303 SBSY
4656 11 11
4657 122107 @BSY0
4660 122115 @BSY1
```

4661 4633 SRB1A ; 1  
4662 4624 SRB0  
4663 22063 CLBY  
4664 121620 @MNI0C  
4665 122115 @BSY1  
4666 122107 @BSY0

4667 4646 SRB2A ; 2  
4670 4640 SRB2  
4671 22100 RSBY  
4672 121155 @MRST  
4673 122115 @BSY1  
4674 122107 @BSY0

; \*\*\*\* READ BUFFERS WITH ALL ONES DATA \*\*\*\*

; TEST THE READ BUFFERS CAPABILITY OF THE CONTROLLER WITH ALL ONES DATA.  
; THE OPERATIONS OF DONE FLAG AND INTERRUPT LOGIC ARE INCLUDED HERE.

```
4675 6177 RBNT: JSR @SET
4676 24337 RBNTT
4677 4705 RBNO
4700 5173 RBZT
4701 177777 -1
```

; PRIME THE FIRST 4 WORDS OF WBUFF WITH ALL ONES, ISSUE A DATA VERIFY  
; COMMAND TO SET BUSY

```
4702 40000 RWDN
4703 3000 DATV
4704 3200 RDBF
4705 6176 RBNO: JSR @PRIME ; PRIME WBUFF
4706 30462 WBUFF
4707 177777 -1
4710 177777 -1
4711 4 4
4712 20771 RBNOA: LDA 0, RBNO-2
4713 24106 LDA 1, CATTS
4714 123000 ADD 1, 0 ; FORM DATA VERIFY + CLEAR ATTEN WORD
4715 4211 JSR DDOA ; ISSUE COMMAND
4716 20115 LDA 0, WBUF
4717 4220 JSR DDOB ; LOAD BUFF ADDR
4720 102400 SUB 0, 0
4721 40277 STA 0, DOCO ; SAVE DATA
4722 63100 RBNOB: DOCS 0, 0 ; START IT
```

; CHECK BUSY SET AND RESET AFTER A DELAY OF 15 US

```
4723 24063 LDA 1, N17
4724 125404 INC 1, 1, SZR ; DELAY
4725 777 JMP -1
4726 4272 RBN1B: JSR SKBN ; BUSY DROPPED?
4727 401 JMP RBN1C ; YES
```

; CHECK DONE FLAG AND RW ATTENTION STATUS BIT

```
4730 4264 RBN1C: JSR SKDZ ; DONE SET?
4731 403 JMP RBN1D ; SET
4732 6166 JSR @ERROR ; DOWN
4733 5077 RBNEB+ONE
4734 4234 RBN1D: JSR DDIA ; READ STATUS
4735 24745 LDA 1, RBNO-3
4736 123414 AND# 1, 0, SZR ; RW ATTEN BIT ON?
4737 403 JMP RBN1E ; IT IS ON
4740 6166 JSR @ERROR ; IT IS OFF
4741 5105 RBNEB+TWO
```

; READ INTERRUPT ACKNOWLEDGE DEVICE CODE AND COMPARE IT WITH THAT  
; ENTERED

```
4742 61477 RBN1E: INTA 0 ; READ DEVICE CODE
4743 24151 LDA 1, DVCD ; ENTERED DEV CODE
4744 106415 SUB# 0, 1, SNR ; THE SAME?
```

```

4745 410 JMP RBN2 ; YES
4746 6200 JSR @SADC ; NO, SET ERROR MESSAGE
4747 46517 STA 1,@INTX0
4750 20151 LDA 0,DVCD
4751 6200 JSR @SADC
4752 46515 STA 1,@INTX1
4753 6166 JSR @ERROR
4754 5113 RBNEB+THREE

```

```

; SET UP THE INTERRUPT HANDLER ROUTINE IN PREPARATION FOR THE
; INTERRUPT TEST

```

```

4755 24505 RBN2: LDA 1,RBIN
4756 44001 STA 1,1 ; SET INTERRUPT VECTOR
4757 176400 SUB 3,3
4760 54505 STA 3,IFLG ; RESET INTERRUPT FLAG
4761 102000 ADC 0,0 ; CONTROLLER MASK BIT
4762 62077 MSKO 0 ; INHIBIT INTERRUPT
4763 61477 INTA 0 ; READ ACKNOWLEDGE
4764 101005 RBN2A: MOV 0,0,SNR ; ANY DEV CODE?
4765 405 JMP RBN2B ; NONE
4766 6200 JSR @SADC ; SOMETHING, ERROR
4767 46477 STA 1,@INTX0 ; SET MESSAGE
4770 6166 JSR @ERROR
4771 5121 RBNEB+FOUR

```

```

; NOW ENABLE CPU INTERRUPT AND TEST FOR CONTROLLER INTERRUPT

```

```

4772 60177 RBN2B: INTEN
4773 20472 LDA 0,IFLG ; GET INTERRUPT FLAG
4774 101005 MOV 0,0,SNR ; INTERRUPTED?
4775 403 JMP RBN2C ; NO, THAT'S GOOD
4776 6166 JSR @ERROR ; YES, MASK DID NOT WORK
4777 5127 RBNEB+FIVE

```

```

; RESET FLAG, CHANGE MASK TO ALLOW CONTROLLER TO INTERRUPT

```

```

5000 126400 RBN2C: SUB 1,1
5001 44464 STA 1,IFLG
5002 24466 LDA 1,MSKB ; CONTROLLER MASK BIT
5003 66077 MSKO 1 ; MASK OFF
5004 125000 MOV 1,1 ; NOP, SHORT DELAY
5005 20460 RBN2D: LDA 0,IFLG ; CHECK FLAG
5006 101004 MOV 0,0,SZR ; INTERRUPT?
5007 403 JMP RBN2E ; YES, MASK WORKED
5010 6166 JSR @ERROR ; SORRY, NO INTERRUPT
5011 5135 RBNEB+SIX

```

```

; END OF INTERRUPT TEST. CLEAR OUT FIRST 4 WORDS OF RBUFF AND ISSUE A
; READ BUFFERS COMMAND

```

```

5012 60277 RBN2E: INTDS
5013 6176 JSR @PRIME
5014 50462 RBUFF
5015 0 0
5016 0 0
5017 4 4
5020 20664 RBN3: LDA 0,RBNO-1 ; GET READ BUFF COMMAND
5021 126620 SUBZR 1,1

```

```

5022 123000      ADD    1,0      ; ADD CLEAR RW DONE BIT
5023  4211      JSR    DDOA     ; ISSUE COMMAND
5024  20116     LDA    0,RBUF   ;
5025  40276     STA    0,DOBO   ; SAVE ADDR
5026  62100 RBN3A: DOBS  0,0      ; LOAD BUFF ADDR & START

```

; CHECK BUSY FLAG RESET

```

5027  24056 RBN3B: LDA    1,K17
5030  124400   NEG    1,1
5031  125404   INC    1,1,SZR  ; DELAY 20 US
5032    777     JMP    .-1
5033  4272 RBN3C: JSR    SKBN    ; BUSY GONE?
5034  401      JMP    RBN3D    ; YES, WENT

```

; CHECK DONE FLAG FOR ON

```

5035  4264 RBN3D: JSR    SKDZ    ; DONE SET?
5036  403   JMP    RBN4      ; SET
5037  6166   JSR    @ERROR   ; OFF
5040  5151   RBNEB+EIGHT

```

; CLEAR DONE FLAG WITH A CLEAR RW ATTENTION BIT

```

5041  20106 RBN4: LDA    0,CATTS
5042  4211   JSR    DDOA     ; CLEAR RW ATTEN
5043  4261 RBN4A: JSR    SKDN    ; DID THAT DO IT?
5044  403   JMP    RBN4B    ; IT DID
5045  6166   JSR    @ERROR   ; DID NOT
5046  5157   RBNEB+NINEE

```

; NOW COMPARE THE FIRST WORD IN EACH THE RBUF AND WBUF TO VERIFY  
; THE READ BUFFERS OPERATION

```

5047  22116 RBN4B: LDA    0,@RBUF ; GET READ WORD
5050  26115   LDA    1,@WBUF ; GET WRITE WORD
5051  106415  SUB#    0,1,SNR  ; AGREE?
5052  2114   JMP    @NTST  ; YES, SUCCESS!
5053  24116   LDA    1,RBUF  ; NO, SET ERROR MESSAGE
5054  44154   STA    1,RBFP
5055  24115   LDA    1,WBUF
5056  44153   STA    1,WBFP
5057  6166   JSR    @ERROR
5060  5165   RBNEB+TEN
5061  2114   JMP    @NTST  ; NEXT TEST

```

; CONTROLLER INTERRUPT HANDLER

```

5062  5063 RBIN: .+1
5063  10402   ISZ    IFLG    ; SET FLAG
5064  2000   JMP    @0      ; DISMISS

5065    0 IFLG: 0
5066  22214 INTX0: INTI+4
5067  22222 INTX1: INTN+4
5070  177377 MSKB: 177377

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

5071 4726 RBNEB: RBN1B
5072 4712 RBNOA
5073 22123 BSYOD
5074 13 13
5075 122115 @BSY1
5076 122107 @BSYO

5077 4730 RBN1C ; 1
5100 4712 RBNOA
5101 22143 DNE1D
5102 13 13
5103 121661 @DNE0
5104 121667 @DNE1

5105 4734 RBN1D ; 2
5106 4712 RBNOA
5107 22163 SRWD
5110 13 13
5111 1 1
5112 121503 @DATIB

5113 4742 RBN1E ; 3
5114 4712 RBNOA
5115 22175 INTAK
5116 13 13
5117 122210 @INTI
5120 122216 @INTN

5121 4764 RBN2A ; 4
5122 4712 RBNOA
5123 22224 IMSK1
5124 122247 @MKB1
5125 122210 @INTI
5126 122340 @NINT

5127 4772 RBN2B ; 5
5130 4712 RBNOA
5131 22224 IMSK1
5132 122263 @INTRT
5133 122274 @INTDC
5134 122306 @INHI

5135 5005 RBN2D ; 6
5136 4712 RBNOA
5137 22235 IMSKO
5140 122324 @MKBO
5141 122340 @NINT
5142 122274 @INTDC

5143 5033 RBN3C ; 7
5144 5020 RBN3
5145 22347 RBFC
5146 14 14
5147 122115 @BSY1
5150 122107 @BSYO

5151 5035 RBN3D ; 8
5152 5020 RBN3
5153 22347 RBFC

```



5154 14 14  
5155 121661 @DNEO  
5156 121667 @DNE1  
  
5157 5043 RBN4A ; 9  
5160 5041 RBN4  
5161 21747 CLDN  
5162 1 1  
5163 121667 @DNE1  
5164 121661 @DNEO  
  
5165 5051 RBN4B+2; 10  
5166 5020 RBN3  
5167 22362 VRBD  
5170 14 14  
5171 14 14  
5172 120027 @AWB

- PAGE 25 -

; \*\*\*\* READ BUFFERS WITH ALL ZEROS DATA \*\*\*\*

; TEST THE READ BUFFERS CAPABILITY OF THE CONTROLLER WITH ALL ZEROS DATA

```
5173 6177 RBZT: JSR @SET
5174 24356 RBZTT
5175 5202 RBZO
5176 5331 PROMS
5177 177777 -1
```

; CLEAR THE FIRST 4 WORDS IN WBUFF AND ISSUE A START DATA VERIFY COMMAND

```
5200 3000 DATV
5201 3200 RDBF
5202 6176 RBZO: JSR @PRIME
5203 30462 WBUFF
5204 0 0
5205 0 0
5206 4 4
5207 20771 LDA 0, RBZO-2 ; GET DATA VERIFY COMMAND
5210 24106 LDA 1, CATTS
5211 123000 ADD 1, 0 ; ADD CLEAR ATTEN BITS
5212 4211 JSR DDOA ; ISSUE COMMAND
5213 20115 LDA 0, WBUF
5214 4220 JSR DDOB ; MA = WBUFF
5215 102400 SUB 0, 0
5216 4230 JSR DDOCS ; START
```

; WAIT 15 US THEN CHECK DONE FLAG = 1

```
5217 24063 LDA 1, N17
5220 125404 INC 1, 1, SZR ; DELAY
5221 777 JMP -1
5222 4264 RBZOA: JSR SKDZ ; DONE = 1?
5223 403 JMP RBZ1 ; YES
5224 6166 JSR @ERROR ; NO
5225 5273 RBZEB
```

; SET FIRST 4 WORDS IN RBUFF TO ALL ONES AND ISSUE READ BUFFERS COMMAND

```
5226 6176 RBZ1: JSR @PRIME
5227 50462 RBUFF
5230 177777 -1
5231 177777 -1
5232 4 4
5233 20746 LDA 0, RBZO-1
5234 4211 JSR DDOA ; ISSUE READ BUFFERS COMMAND
5235 20116 LDA 0, RBUF
5236 4220 JSR DDOB ; MA = RBUFF
5237 102400 SUB 0, 0
5240 63100 DOCS 0, 0
```

; CHECK DONE FLAG FOR NOT SET YET

```
5241 63600 RBZ1A: SKPDN 0 ; DONE SET?
5242 401 JMP RBZ1B ; NO, THAT'S GOOD
```

; CHECK DONE TO SET AFTER 20 US DELAY

72

```

5243 24056 RBZ1B: LDA 1, K17
5244 124400 NEG 1, 1
5245 125404 INC 1, 1, SZR ; DELAY
5246 777 JMP -1
5247 4264 RBZ1C: JSR SKDZ ; DONE?
5250 403 JMP RBZ1D ; DONE
5251 6166 JSR @ERROR ; NO DONE
5252 5307 RBZEB+TWO

```

; RESET DONE WITH IO RESET

```

5253 62677 RBZ1D: IORST
5254 4261 JSR SKDN ; RESET?
5255 403 JMP RBZ2 ; YES
5256 6166 JSR @ERROR ; NO
5257 5315 RBZEB+THREE

```

; COMPARE THE FIRST WORD IN EACH THE RBUFF AND WBUFF

```

5260 22116 RBZ2: LDA 0, @RBUF ; GET READ WORD
5261 26115 LDA 1, @WBUF ; GET WRITE WORD
5262 106415 SUB# 0, 1, SNR ; DO THEY MATCH?
5263 2114 JMP @NTST ; THEY DO, EXIT
5264 24116 LDA 1, RBUF ; NO, SET ERROR MESSAGE
5265 44154 STA 1, RBFP
5266 24115 LDA 1, WBUF
5267 44153 STA 1, WBFP
5270 6166 JSR @ERROR
5271 5323 RBZEB+FOUR
5272 2114 JMP @NTST

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

5273 5222 RBZEB: RBZOA
5274 5202 RBZO
5275 22143 DNE1D
5276 13 13
5277 121661 @DNEO
5300 121667 @DNE1

5301 5241 RBZ1A ; 1
5302 5226 RBZ1
5303 21756 SRSD
5304 13 13
5305 121667 @DNE1
5306 121661 @DNEO

5307 5247 RBZ1C ; 2
5310 5226 RBZ1
5311 22143 DNE1D
5312 13 13
5313 121661 @DNEO
5314 121667 @DNE1

5315 5254 RBZ1D+1; 3
5316 5253 RBZ1D
5317 21747 CLDN
5320 121172 @RSET
5321 121667 @DNE1

```

5322 121661

@DNEO

5323 5260

RBZ2 ; 4

5324 5226

RBZ1

5325 22362

VRBD

5326 16

16

5327 14

14

5330 122742

@ASWB

;\*\*\*\* SURFCAE-SECTOR PROMS TEST \*\*\*\*

;VERIFY THE INSTALLATIONS AND THE INTERNAL PROGRAMMING OF THE PROMS

5331	6177	PROMS:	JSR	@SET
5332	24375		PROMT	
5333	5337		PMS	
5334	5756		ENDL	
5335	0		0	

;TEST TO SEE IF PROMS ARE INSTALLED  
;CHECK STATUS AFTER ISSUING AN ILLEGAL DISK ADDRESS

5336	3000		DATV	
5337	20777	PMS:	LDA	0,-1 ;GET DATA VERIFY COMMAND
5340	24106		LDA	1,CATTS
5341	123000		ADD	1,0 ;ADD CLEAR ATTEN
5342	4211		JSR	DDOA ;ISSUE COMMAND
5343	20426		LDA	0,MAXS
5344	4230		JSR	DDOCS ;ISSUE MAX DISK ADDR
5345	4261		JSR	SKDN ;WAIT FOR DONE
5346	777		JMP	.-1

;CHECK STATUS INDICATIONS TO COMMANDS

5347	30421		LDA	2,BIT7
5350	4234	PMS1:	JSR	DDIA ;READ STATUS
5351	143414		AND#	2,0,SZR ;ILLEGAL?
5352	404		JMP	PMS1A ;YES
5353	6166		JSR	@ERROR ;NO PROMS
5354	5704		PMEB	
5355	762		JMP	PMS
5356	102400	PMS1A:	SUB	0,0
5357	4230		JSR	DDOCS ;TEST LOW END
5360	4261		JSR	SKDN
5361	777		JMP	.-1
5362	4234	PMS1B:	JSR	DDIA ;DONE, GET STATUS
5363	143415		AND#	2,0,SNR ;ILLEGAL?
5364	407		JMP	PMS2 ;NO, GOT PROMS
5365	6166		JSR	@ERROR ;NO PROMS
5366	5712		PMEB+ONE	
5367	750		JMP	PMS

5370	400	BIT7:	400
5371	77740	MAXS:	77740
5372	5663	PTBA:	PTBL

;TEST SECTOR ADDRESSING OF PROMS, A DRIVE AT A TIME

5373	34777	PMS2:	LDA	3,PTBA
5374	54502		STA	3,PTBP ;SET TEST RESULT POINTER
5375	21400		LDA	0,0,3 ;GET DRIVE SELECT
5376	4211		JSR	DDOA ;SELECT IT
5377	20102		LDA	0,SMSK ;SECTOR ADDR MAX
5400	4230		JSR	DDOCS ;LOADIT & START
5401	4261		JSR	SKDN ;WAIT FOR DONE
5402	777		JMP	.-1
5403	4234		JSR	DDIA ;READ STATUS

```

5404 113415      AND#   0, 2, SNR   ; ILLEGAL?
5405      405      JMP     PMS2A   ; NO, EXIT
5406 34107      LDA     3, ASECT
5407 20277      LDA     0, DOC0 ; SECT #
5410 162400     SUB     3, 0    ; YES, BACK DOWN A SECT
5411      767      JMP     PMS2+5  ; LOOP BACK

```

; LEGAL STATUS, SAVE MAX SECTOR ADDRESS IN TABLE

```

5412 34464 PMS2A: LDA   3, PTBP   ; TABLE POINTER
5413 20277      LDA   0, DOC0   ; GET SECT ADDR
5414 41401      STA   0, 1, 3   ; SAVE SECT ADDR
5415 34107      LDA   3, ASECT
5416 20277      LDA   0, DOC0   ; GET ADDR
5417 162400     SUB   3, 0      ; KEEP BACKING DOWN
5420 101112     MOV#  0, 0, SZC  ; MINUS?
5421      413      JMP   PMS3      ; YES, EXIT
5422 4230       JSR   DDOCS     ; NO, ISSUE IT
5423 4261       JSR   SKDN
5424      777      JMP   . -1
5425 4234 PMS2B: JSR   DDIA    ; DONE, GET STATUS
5426 143415     AND#  2, 0, SNR  ; ILLEGAL?
5427      766      JMP   PMS2A+3   ; NO, CONTINUE
5430 6166       JSR   @ERROR    ; YES, NG
5431 5720       PMEB+TWO
5432      763      JMP   PMS2A+3

```

; TEST VOLUME 0 SURFACE ADDRESSING

```

5433 2000       2000
5434 24777 PMS3: LDA   1, . -1 ; SURFACE ADDR INCREMENT
5435 102400     SUB   0, 0
5436 4230       JSR   DDOCS     ; LOAD SURF 0
5437 4261       JSR   SKDN
5440      777      JMP   . -1
5441 4234       JSR   DDIA    ; DONE, GET STATUS
5442 113414     AND#  0, 2, SZR  ; ILLEGAL?
5443      411      JMP   PMS3B     ; YES, EXIT
5444 20277      LDA   0, DOC0   ; GET ADDR
5445 123000     ADD   1, 0      ; OK, NEXT SURFACE
5446 34103      LDA   3, HMSK
5447 116414 PMS3A: SUB#  0, 3, SZR  ; AT MAX?
5450      766      JMP   PMS3+2   ; NO, KEEP GOING
5451 6166       JSR   @ERROR    ; YES, SOMETHING WRONG
5452 5726       PMEB+THREE
5453      763      JMP   PMS3+2

```

; ILLEGAL STATUS, SAVE SURFACE NUMBER IN TABLE

```

5454 34422 PMS3B: LDA   3, PTBP
5455 20277      LDA   0, DOC0   ; GET SURF ADDR
5456 41402      STA   0, 2, 3   ; SAVE IN TABLE
5457 20277      LDA   0, DOC0   ; GET IT AGAIN
5460 123000     ADD   1, 0
5461 4230       JSR   DDOCS     ; TEST MORE
5462 4261       JSR   SKDN
5463      777      JMP   . -1
5464 4234       JSR   DDIA    ; DONE, GET STATUS
5465 143414 PMS3C: AND#  2, 0, SZR  ; ILLEGAL?

```

```

5466 403 JMP PMS3D ; YES, EXIT
5467 6166 JSR @ERROR ; NO
5470 5734 PMEB+FOUR
5471 20277 PMS3D: LDA 0, DOCO
5472 34103 LDA 3, HMSK
5473 162414 SUB# 3, 0, SZR ; AT MAX?
5474 763 JMP PMS3B+3 ; NO, LOOP BACK
5475 402 JMP PMS4 ; YES, GET OUT

```

```
5476 0 PTBP: 0
```

```
; TEST VOLUME 1 SURFACE ADDRESSING
```

```

5477 34057 PMS4: LDA 3, K20
5500 20636 LDA 0, PMS-1
5501 163000 ADD 3, 0 ; DATV + VOL SELECT
5502 34774 LDA 3, PTBP
5503 35400 LDA 3, 0, 3
5504 163000 ADD 3, 0 ; INSERT DRV #
5505 4211 JSR DDOA ; SELECT DRIVE
5506 102400 SUB 0, 0
5507 4230 PMS4A: JSR DDOCS ; LOAD & START
5510 4261 JSR SKDN
5511 777 JMP .-1
5512 4234 PMS4B: JSR DDIA ; DONE, GET STATUS
5513 143414 AND# 2, 0, SZR ; ILLEGAL?
5514 412 JMP PMS5 ; YES
5515 20277 LDA 0, DOCO ; FETCH SURF ADDR
5516 34407 LDA 3, K40K
5517 162414 SUB# 3, 0, SZR ; AT 40000?
5520 403 JMP PMS4C ; NO
5521 6166 JSR @ERROR ; YES
5522 5742 PMEB+FIVE
5523 123000 PMS4C: ADD 1, 0 ; NEXT INCREMENT
5524 763 JMP PMS4A

```

```
5525 40000 K40K: 40000
```

```
; ILLEGAL STATUS, SAVE MAX VOL 1 SURFACE ADDRESS IN TABLE
```

```

5526 34750 PMS5: LDA 3, PTBP
5527 20277 LDA 0, DOCO ; GET ADDR
5530 41403 STA 0, 3, 3 ; PUT IT IN TABLE
5531 123000 ADD 1, 0
5532 4230 JSR DDOCS ; CONTINUE NEXT INCREMENT
5533 4261 JSR SKDN
5534 777 JMP .-1
5535 4234 PMS5A: JSR DDIA ; DONE, READ STATUS
5536 143414 AND# 2, 0, SZR ; ILLEGAL
5537 403 JMP PMS5B ; YES
5540 6166 JSR @ERROR ; NO, BAD PROMS
5541 5750 PMEB+SIX
5542 34103 PMS5B: LDA 3, HMSK
5543 20277 LDA 0, DOCO ; GET ADDR
5544 162414 SUB# 3, 0, SZR ; AT MAX?
5545 764 JMP PMS5+3 ; NO, CONTINUE

```

```
; COMPLETED TEST FOR ONE DRIVE, DO NEXT ONE IF NOT ALREADY DRIVE 3
```

```

5546 34730      LDA    3,PTBP    ;TABLE POINTER
5547 24052      LDA    1,K4
5550 137000     ADD    1,3      ;NEXT DRIVE
5551 21400      LDA    0,0,3
5552 101113     MOV#L# 0,0,SNC   ;ALL DONE?
5553 2507       JMP    @PMSA    ;NOT YET

```

; PRINT PROM TEST RESULTS ONLY ON THE 1ST PASS

```

5554 10146      ISZ    PASS      ;COUNT PASS
5555 24146      LDA    1,PASS
5556 125224     MOVZR  1,1,SZR   ;1ST PASS?
5557 2114       JMP    @NTST     ;NO, EXIT
5560 20052      LDA    0,K4
5561 40150      STA    0,LCNT   ;SET LINE COUNT
5562 24610      LDA    1,PTBA
5563 44713      STA    1,PTBP   ;RESET TABLE POINTER
5564 6155       JSR    @PRINT   ;"PROM TEST RESULTS"
5565 22413      PTD
5566 6155       JSR    @PRINT   ;HEADLINER
5567 22424      PLLS
5570 6155       JSR    @PRINT   ;NEW LINE
5571 20076      CRLF
5572 20076      LDA    0,SPACE
5573 6156       JSR    @CHAR    ;START WITH A SPACE
5574 126400     SUB    1,1
5575 44147      STA    1,PCH    ;RESET SPACE COUNT
5576 26700      LDA    1,@PTBP  ;DRIVE NUMBER
5577 125220     MOVZR  1,1      ;RIGHT JUSTIFY
5600 125220     MOVZR  1,1
5601 125220     MOVZR  1,1
5602 125220     MOVZR  1,1
5603 125220     MOVZR  1,1
5604 6164       JSR    @DECN    ;PRINT DRV #
5605 4441       JSR    BLANK    ;SPACE TO NEXT COLUMN
5606 6          6
5607 10667      ISZ    PTBP
5610 26666      LDA    1,@PTBP  ;GET SECTOR LIMIT
5611 125220     MOVZR  1,1      ;RIGHT JUSTIFY
5612 125220     MOVZR  1,1
5613 125220     MOVZR  1,1
5614 125220     MOVZR  1,1
5615 125220     MOVZR  1,1
5616 6164       JSR    @DECN    ;PRINT IT
5617 4427       JSR    BLANK    ;SPACE TO NEXT SPOT
5620 10         10
5621 10655      ISZ    PTBP
5622 26654      LDA    1,@PTBP  ;GET VOL 0 SURF #
5623 125300     MOV#S  1,1      ;SHIFT INTO POSITION
5624 125220     MOVZR  1,1
5625 124640     NEGOR  1,1      ;BACK UP 1
5626 124000     COM    1,1
5627 6164       JSR    @DECN    ;PRINT IT
5630 4416       JSR    BLANK
5631 7          7
5632 10644      ISZ    PTBP
5633 26643      LDA    1,@PTBP  ;GET VOL 1 SURF #
5634 125300     MOV#S  1,1
5635 125220     MOVZR  1,1

```



```

5636 124640      NEGOR  1,1      ; BACK UP 1
5637 124000      COM      1,1
5640   6164      JSR      @DECN
5641 10635       ISZ      PTBP      ; NEXT DRIVE
5642 14150       DSZ      LCNT      ; DOEN?
5643   725       JMP      PMD      ; NOT YET
5644  2114       JMP      @NTST     ; YES

```

; COLUMN SPACING

```

5645   0         0
5646 54777 BLANK: STA  3,.-1
5647 25400      LDA  1,0,3      ; SPACE COUNT
5650 20147      LDA  0,PCH     ; PRINTED COUNT
5651 106400     SUB  0,1      ; SPACES NEEDED
5652 124400     NEG  1,1
5653 20076      LDA  0,SPACE
5654   6156      JSR      @CHAR   ; SPACE IT
5655 125404     INC  1,1,SZR  ; FINISH?
5656   776      JMP      .-2      ; NOT YET
5657 44147      STA  1,PCH   ; RESET COUNT
5660 34765      LDA  3,BLANK-1
5661  1401      JMP      1,3

```

5662 5374 PMSA: PMS2+1

; PROMS TEST RESULTS SAVE TABLE

```

5663   0 PTBL: DRV0
5664   0         0      ; MAX SECT #
5665   0         0      ; MAX VOL 0 SURF
5666   0         0      ; MAX VOL 1 SURF
5667  40         DRV1
5670   0         0
5671   0         0
5672   0         0
5673 100         DRV2
5674   0         0
5675   0         0
5676   0         0
5677 140         DRV3
5700   0         0
5701   0         0
5702   0         0
5703 177777     -1

```

; \*\*\*\* ERROR MESSAGE BLOCK \*\*\*\*

```

5704 5350 PMEB: PMS1
5705 5337      PMS
5706 22457     INCK
5707   12      12
5710 122471    @NPMO
5711 121511    @DATI9

5712 5362      PMS1B ; 1
5713 5337      PMS
5714 22457     INCK
5715   12      12

```

```

5716 122510 @NPM1
5717 121445 @DATI3

5720 5425 PMS2B ; 2
5721 5373 PMS2
5722 22526 SAFD
5723 12 12
5724 1 1
5725 121445 @DATI3

5726 5447 PMS3A ; 3
5727 5434 PMS3
5730 22537 HDVO
5731 12 12
5732 1 1
5733 121511 @DATI9

5734 5465 PMS3C ; 4
5735 5434 PMS3
5736 22537 HDVO
5737 12 12
5740 1 1
5741 121511 @DATI9

5742 5512 PMS4B ; 5
5743 5477 PMS4
5744 22552 HDV1
5745 12 12
5746 1 1
5747 121445 @DATI3

5750 5535 PMS5A ; 6
5751 5477 PMS4
5752 22552 HDV1
5753 12 12
5754 1 1
5755 121511 @DATI9

```

;\*\*\*\* END OF LOGIC TEST \*\*\*\*

;PRINT TOTAL PASS COUNT THEN EXIT IF A KEY HAS BEEN STRUCK, ESLE  
;RECYCLE TEST

```

5756 10146 ENDL: ISZ PASS ;COUNT PASSES
5757 63610 SKPDN TTI ;STOP?
5760 410 JMP ENDL1 ;NO
5761 6155 JSR @PRINT
5762 20076 CRLF
5763 24146 LDA 1,PASS ;YES, GET PASS COUNT
5764 6164 JSR @DECN ;PRINT IT
5765 6155 JSR @PRINT ;"PASSES"
5766 120010 @PASSE
5767 2111 JMP @ABORT ;QUIT

5770 2401 ENDL1: JMP @. +1
5771 3300 RGRS

```

; LOTUS 700 DISK CONTROLLER DIAGNOSTICS  
; VERSION 1.7 JAN 31, 1981

; CONTROLLER TO DRIVE TEST SECTION

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

6400 . LOC 6400

;\*\*\*\* RECALIBRATE DRIVE TEST \*\*\*\*

; AFTER ISSUING A RECAL COMMAND, CHECK DRIVE STATUS BEFORE AND AFTER  
; DRIVE DONE

6400 6177 RECAL: JSR @SET ; SET IT UP  
6401 24562 RCLT ; TEST TITLE  
6402 6405 RCL ; TEST START VECTOR  
6403 6456 FMC1 ; NEXT TEST VECTOR  
6404 0 0 ; PRINT DRIVE NUMBER FLAG

; ISSUE RECAL COMMAND

6405 6167 RCL: JSR @DRCMD  
6406 200 RCAL  
6407 764 764

; CHECK PRE-DONE STATUS FOR DRIVE BUSY = 1

6410 22422 LDA 0, @DSTAP ; DRIVE STATUS  
6411 24422 LDA 1, DRBY  
6412 44306 STA 1, DIBN ; SET FOR PRINT  
6413 106415 RCL1: SUB# 0, 1, SNR ; BUSY SET?  
6414 403 JMP RCL2 ; YES  
6415 6166 JSR @ERROR ; NO  
6416 6434 RCLEB

; CHECK CONTROL STATUS FOR DRIVE DONE = 1

6417 6201 RCL2: JSR @CKCK  
6420 0 0  
6421 6424 RCL2A ; NO ERROR  
6422 6166 JSR @ERROR ; DISAGREE  
6423 6442 RCLEB+ONE

; CHECK DRIVE STATUS FOR BUSY = 0 AFTER DONE

6424 6203 RCL2A: JSR @CKDS  
6425 10000 DRDY  
6426 6431 RCL2B ; CHECKED OUT  
6427 6166 JSR @ERROR ; NOT SO  
6430 6450 RCLEB+TWO  
6431 2114 RCL2B: JMP @NTST

6432 15665 DSTAP: DSTAO  
6433 14000 DRBY: DRDY+DBSY

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

6434 6413 RCLEB: RCL1  
6435 6405 RCL  
6436 22626 DDBS  
6437 15 15  
6440 2 2  
6441 16 16  
  
6442 6417 RCL2 ; 1

---  
6443 6405 RCL  
6444 21710 DRAT  
6445 15 15  
6446 1 1  
6447 15 15  
  
6450 6424 RCL2A ; 2  
6451 6405 RCL  
6452 22642 DBY  
6453 15 15  
6454 2 2  
6455 16 16

;\*\*\*\* FORMAT ONE SECTOR \*\*\*\*

;USE THE FORMAT FUNCTION OF THE CONTROLLER TO FORMAT 1 SECTOR

6456	6177	FMC1:	JSR	@SET
6457	24570		FM1T	
6460	6463		FM1	
6461	6522		RFM1	
6462	0		0	

;SET DISK ADDRESS TO SURFACE 0, SECTOR 0 AND SECTOR COUNT 1 THEN  
;ISSUE FORMAT COMMAND

6463	6205	FM1:	JSR	@DADD
6464	0		0	
6465	0		0	
6466	1		1	
6467	6170		JSR	@RWCMD
6470	1400		FMAT	
6471	30462		WBUFF	
6472	24		24	

;DONE, CHECK CONTROL STATUS

6473	6202	FM1A:	JSR	@CKCS
6474	40000		RWDN	; RW DONE STATUS
6475	6500		FM1B	; DK VECTOR
6476	6166		JSR	@ERROR ; BAD STATUS
6477	6506		FM1EB	

;CHECK DISK ADDRESS WORD

6500	6204	FM1B:	JSR	@CKDA
6501	40		40	; SECT 1
6502	6505		FM1C	; NO ERROR VECTOR
6503	6166		JSR	@ERROR ; NOT RIGHT
6504	6514		FM1EB+ONE	
6505	2114	FM1C:	JMP	@NTST ; NEXT TEST

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

6506	6473	FM1EB:	FM1A	
6507	6463		FM1	
6510	22701		RWDS	
6511	6		6	
6512	10		10	
6513	15		15	
6514	6500		FM1B	
6515	6463		FM1	
6516	22714		RWDA	
6517	6		6	
6520	3		3	
6521	17		17	

;\*\*\*\* READ ONE SECTOR FORMAT \*\*\*\*

; READ AND VERIFY HEADER AND DATA AREA AS FORMATTED IN THE PRECEDING  
; TEST

```
6522 6177 RFM1: JSR @SET
6523 24600 RF1T
6524 6527 RF1
6525 6611 WHS1
6526 0 0
```

; PRESET RBUF WITH 6 WORDS OF ALL ONES THEN ISSUE COMMAND TO READ  
; HEADER INTO RBUF

```
6527 6176 RF1: JSR @PRIME
6530 50462 RBUF
6531 177777 -1
6532 177777 -1
6533 6 6
6534 6170 RF1A: JSR @RWCMD
6535 3600 RHDR
6536 50462 RBUF
6537 24 24
```

; DONE, CHECK CONTROL STATUS AND DISK ADDRESS WORD

```
6540 6202 RF1B: JSR @CKCS
6541 40000 RWDN
6542 6545 RF1C
6543 6166 JSR @ERROR ; BAD STATUS
6544 6567 RF1EB
6545 6204 RF1C: JSR @CKDA
6546 40 40
6547 6552 RF1D
6550 6166 JSR @ERROR ; BAD ADDR WORD
6551 6575 RF1EB+ONE
```

; CHECK FIRST 6 WORDS OF RBUF FOR ALL ZEROS

```
6552 34116 RF1D: LDA 3, RBUF ; RBUF ADDR
6553 24413 LDA 1, N6
6554 21400 RF1E: LDA 0, 0, 3
6555 101004 MOV 0, 0, SZR ; ZEROS?
6556 404 JMP RF1F ; NO, GARBAGE!
6557 125404 INC 1, 1, SZR ; MORE?
6560 774 JMP RF1E ; YES
6561 2114 JMP @NTST ; ALL DONE

6562 54154 RF1F: STA 3, RBUF ; SAVE ERROR ADDR
6563 6166 JSR @ERROR
6564 6603 RF1EB+TWO
6565 2114 JMP @NTST
```

```
6566 177772 N6: -6
```

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```
6567 6534 RF1EB: RF1A
6570 6456 FMC1
```

6571	22701	RWDS	
6572	6	6	
6573	10	10	
6574	16	16	
6575	6545	RF1C	; 1
6576	6456	FMC1	
6577	22714	RWDA	
6600	6	6	
6601	3	3	
6602	17	17	
6603	6554	RF1E	; 2
6604	6456	FMC1	
6605	22730	HWC	
6606	6	6	
6607	14	14	
6610	121221	@ALLO	



;\*\*\*\* WRITE ONE SECTOR HEADER \*\*\*\*

;WRITE A SOFTWARE GENERATED HEADER TO SURFACE 0, SECTOR 0 THEN READ  
;IT BACK FOR VERIFICATION

```
6611 6177 WHS1: JSR @SET
6612 24613 WH1T
6613 6616 WH1
6614 6726 FMC2
6615 0 0
```

;PREPARE 3 HEADER WORDS IN WBUF THEN WRITE THEM OUT

```
6616 34115 WH1: LDA 3,WBUF
6617 102400 SUB 0,0
6620 41400 STA 0,0,3 ;WORDS 0 & 1 = 0
6621 41401 STA 0,1,3
6622 100000 COM 0,0 ;MAKE -1
6623 41402 STA 0,2,3 ;WORD 2 = -1
6624 6205 JSR @DADD ;SET DISK ADDR
6625 0 0
6626 0 0
6627 1 1
6630 6170 WH1A: JSR @RWCMD
6631 600 WHDR
6632 30462 WBUF
6633 24 24
```

;DONE, CHECK CONTROL STATUS AND DISK ADDRESS WORD

```
6634 6202 WH1B: JSR @CKCS
6635 40000 RWDN
6636 6641 WH1C ;GOOD STATUS
6637 6166 JSR @ERROR ;NG STATUS
6640 6676 WHEB
6641 6204 WH1C: JSR @CKDA
6642 40 40
6643 6646 WH2 ;GOOD ADDR WORD
6644 6166 JSR @ERROR ;NG ADDR
6645 6704 WHEB+ONE
```

;PRIME RBUF WITH COMPLEMENT OF WBUF BEFORE READING HEADER BACK

```
6646 34116 WH2: LDA 3,RBUF ;RBUF ADDR
6647 102000 ADC 0,0 ;MAKE -1
6650 24716 LDA 1,N6
6651 41400 WH2A: STA 0,0,3 ;SET IT
6652 175400 INC 3,3
6653 125404 INC 1,1,SZR
6654 775 JMP WH2A
6655 100000 COM 0,0
6656 41774 STA 0,-4,3 ;WORD 2 = 0
6657 6170 JSR @RWCMD ;ISSUE READ HEADER COMMAND
6660 3600 RHDR
6661 50462 RBUF
6662 24 24
```

;DONE, CHECK CONTROL STATUS

```

6663 6202 WH2B: JSR @CKCS
6664 40000 RWDN
6665 6670 WH2C ; STATUS OK
6666 6166 JSR @ERROR ; NOT OK
6667 6712 WHEB+TWO

```

; COMPARE HEADER WORDS IN RBUFF

```

6670 6173 WH2C: JSR @HCMP
6671 1 1
6672 6675 WH2D ; NO ERROR
6673 6166 JSR @ERROR ; NOT MATCH
6674 6720 WHEB+THREE
6675 2114 WH2D: JMP @NTST

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

6676 6634 WHEB: WH1B
6677 6630 WH1A
6700 22701 RWDS
6701 6 6
6702 10 10
6703 15 15

6704 6641 WH1C ; 1
6705 6630 WH1A
6706 22714 RWDA
6707 6 6
6710 3 3
6711 17 17

6712 6663 WH2B ; 2
6713 6646 WH2
6714 22701 RWDS
6715 6 6
6716 10 10
6717 15 15

6720 6670 WH2C ; 3
6721 6646 WH2
6722 22730 HWC
6723 16 16
6724 14 14
6725 122742 @ASWB

```

;\*\*\*\* FORMAT ONE TRACK \*\*\*\*

;USE THE FORMAT FUNCTION OF THE CONTROLLER TO FORMAT THE TRACK AT  
;SURFACE 0, CYLINDER 0

6726	6177	FMC2:	JSR	@SET
6727	24626		FM2T	
6730	6733		FM2	
6731	6772		RFM2	
6732	0		0	

;SET DISK ADDRESS TO SURFACE 0, SECTOR 0 AND SECTOR COUNT 32 THEN  
;ISSUE FORMAT COMMAND

6733	6205	FM2:	JSR	@DADD
6734	0		0	
6735	0		0	
6736	100123		@STRK	
6737	6170		JSR	@RWCMD ; ISSUE COMMAND
6740	1400		FMAT	
6741	30462		WBUFF	
6742	50		50	

;DONE. CHECK CONTROL STATUS AND DISK ADDRESS WORD

6743	6202	FM2A:	JSR	@CKCS
6744	40000		RWDN	
6745	6750		FM2B	; GOOD STATUS
6746	6166		JSR	@ERROR ; BAD STATUS
6747	6756		FM2EB	
6750	6204	FM2B:	JSR	@CKDA
6751	0		0	
6752	6755		FM2C	; GOOD ADDR
6753	6166		JSR	@ERROR ; BAD ADDR
6754	6764		FM2EB+ONE	
6755	2114	FM2C:	JMP	@NTST

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

6756	6743	FM2EB:	FM2A	
6757	6733		FM2	
6760	22701		RWDS	
6761	6		6	
6762	10		10	
6763	15		15	
6764	6750		FM2B	; 1
6765	6733		FM2	
6766	22714		RWDA	
6767	6		6	
6770	3		3	
6771	17		17	

;\*\*\*\* READ FORMATTED TRACK OF HEADERS \*\*\*\*

;READ ONE TRACK OF HEADERS AS FORMATTED IN THE PRECEDING TEST INTO  
;RBUF AND COMPARE THEM WITH SOFTWARE GENERATED REFERENCE

6772	6177	RFM2:	JSR	@SET
6773	24637		RF2T	
6774	6777		RF2	
6775	7056		RDT1	
6776	0		0	

;PRIME RBUF WITH 192 WORDS OF ALL ONES THEN ISSUE READ HEADER COMMAND

6777	6176	RF2:	JSR	@PRIME
7000	50462		RBUF	
7001	177777		-1	
7002	177777		-1	
7003	300		300	
7004	6170		JSR	@RWCMD ;ISSUE COMMAND
7005	3600		RHDR	
7006	50462		RBUF	
7007	50		50	

;DONE, CHECK CONTROL STATUS AND DISK ADDRESS WORD

7010	6202	RF2A:	JSR	@CKCS
7011	40000		RWDN	
7012	7015		RF2B	;STATUS OK
7013	6166		JSR	@ERROR ;NOT OK
7014	7034		RF2EB	
7015	6204	RF2B:	JSR	@CKDA
7016	0		0	
7017	7022		RF2C	;ADDR OK
7020	6166		JSR	@ERROR ;ADDR NOT OK
7021	7042		RF2EB+ONE	

;GENERATE HEADERS IN WBUF AND COMPARE THEM WITH THOSE READ BACK IN RBUF

7022	102400	RF2C:	SUB	0,0
7023	40127		STA	0,CYLN ;SET CYLINDER 0
7024	6172		JSR	@HDRG ;CALL GENERATOR
7025	100123		@STRK	
7026	6173		JSR	@HCMP ;CALL COMPARATOR
7027	100123		@STRK	
7030	7033		RF2D	;NO ERROR
7031	6166		JSR	@ERROR ;TROUBLE!
7032	7050		RF2EB+TWO	
7033	2114	RF2D:	JMP	@NTST

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

7034	7010	RF2EB:	RF2A
7035	6726		FMC2
7036	22701		RWDS
7037	6		6
7040	10		10
7041	15		15
7042	7015	RF2B	;1

---  
7043 6726 FMC2  
7044 22714 RWDA  
7045 6 6  
7046 3 3  
7047 17 17  
  
7050 7024 RF2C+2 ; 2  
7051 6726 FMC2  
7052 22730 HWC  
7053 16 16  
7054 14 14  
7055 122742 @ASWB

;\*\*\*\* READ ONE SECTOR DATA \*\*\*\*

;READ ONE SECTOR OF DATA FROM SECTORS 1, 2, 4, 8 AND 16 RESPECTIVELY

```
7056 6177 RDT1: JSR @SET
7057 24656 RD1T
7060 7063 RD1
7061 7155 TOPM
7062 0 0
```

; INITIALIZE SECTOR ADDRESS POINTER AND DISK ADDRESS PRESET

```
7063 34446 RD1: LDA 3,SCL ;SECT ADDR TEST LIST
7064 54142 STA 3,SAVP ;SET POINTER
7065 22142 LDA 0,@SAVP ;GET TEST SECT ADDR
7066 101112 MOVL# 0,0,SZC ;END OF LIST?
7067 2114 JMP @NTST ;YES, GO TO NEXT TEST
7070 40403 STA 0,RD1A ;NO, SET IT
7071 6205 JSR @DADD ;SET DISK ADDR
7072 0 0
7073 0 RD1A: 0
7074 1 1
```

;PRIME RBUFF WITH 256 WORDS OF ALL ONES THEN ISSUE READ DATA COMMAND

```
7075 6176 RD1B: JSR @PRIME
7076 50462 RBUFF
7077 177777 -1
7100 177777 -1
7101 400 400
7102 6170 JSR @RWCMD ;ISSUE COMMAND
7103 0 READ
7104 50462 RBUFF
7105 50 50
```

; DONE, CHECK CONTROL STATUS

```
7106 6202 RD1C: JSR @CKCS
7107 40000 RWDN
7110 7113 RD1D ;LOOKS GOOD
7111 6166 JSR @ERROR ;LOOKS LIKE ERROR
7112 7141 RD1EB
```

;VERIFY DATA IN RBUFF IS 256 WORDS OF ALL ZEROS

```
7113 24064 RD1D: LDA 1,N400 ;WORD COUNT
7114 34116 LDA 3,RBUF ;RBUFF ADDR
7115 21400 LDA 0,0,3
7116 101004 MOV 0,0,SZR ;ZEROS?
7117 405 JMP RD1E ;SOMETHING ELSE!
7120 175400 INC 3,3
7121 125404 INC 1,1,SZR ;MORE?
7122 773 JMP RD1D+2 ;YES
7123 404 JMP RD1F ;NO, GO DO NEXT SECTOR

7124 54154 RD1E: STA 3,RBFP ;SAVE WORD ERROR ADDR
7125 6166 JSR @ERROR
7126 7147 RD1EB+ONE
```

---

;BUMP LIST POINTER AND LOOP BACK

7127	10142	RD1F:	ISZ	SAVP
7130	735		JMP	RD1+2

;SECTOR ADDRESS TEST LIST

7131	7132	SCL:	. +1	
7132	0		0	
7133	1		1	
7134	2		2	
7135	4		4	
7136	10		10	
7137	20		20	
7140	177777		-1	; END

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

7141	7106	RD1EB:	RD1C
7142	7075		RD1B
7143	22701		RWDS
7144	6		6
7145	10		10
7146	15		15
7147	7116		RD1D+3
7150	7075		RD1B
7151	22750		DWC
7152	16		16
7153	14		14
7154	121221		@ALLO

;\*\*\*\* MEMORY ADDRESS INCREMENT \*\*\*\*

;WRITE AND READ ONE SECTOR OF DATA FROM AND TO MEMORY STARTING ADDRESS  
;77400. WHEN DONE, CHECK MEMORY ADDRESS REGISTER FOR INCREMENTING TO THE  
;CORRECT ADDRESS

7155	6177	TOPM:	JSR	@SET
7156	24670		TOPT	
7157	7162		TOP	
7160	7301		RWTM	
7161	0		0	

;WRITE A SECTOR OF DATA FROM LOCATION 77400 TO SURF 0, SECT 0

7162	102400	TOP:	SUB	0,0
7163	40130		STA	0,SCTN ;SECT 0
7164	6170		JSR	@RWCMD
7165	3400		WRITE	
7166	77400		77400	
7167	24		24	

;DONE, CHECK CONTROL STATUS

7170	6202	TOP1:	JSR	@CKCS
7171	40000		RWDN	
7172	7175		TOP1A	;STATUS OK VECTOR
7173	6166		JSR	@ERROR ;BUM STATUS
7174	7251		TOPEB	

;READ MA REG TO CHECK IT FOR POINTING TO 100022

7175	4434	TOP1A:	JSR	RMA ;GO GET MA REG
7176	24427		LDA	1,WMAP ;GET MA REF
7177	106415		SUB#	0,1,SNR ;WHAT IS IT?
7200	404		JMP	TOP2 ;IT IS 100022
7201	44305		STA	1,DIAN
7202	6166		JSR	@ERROR ;SOMETHING ELSE
7203	7257		TOPEB+ONE	

;NOW READ ONE SECTOR OF DATA INTO LOCATION 77400

7204	6170	TOP2:	JSR	@RWCMD
7205	0		READ	
7206	77400		77400	
7207	24		24	

;DONE, CHECK CONTROL STATUS

7210	6202	TOP2A:	JSR	@CKCS
7211	40000		RWDN	
7212	7215		TOP2B	;GOOD STATUS VECTOR
7213	6166		JSR	@ERROR ;BAD ONE
7214	7265		TOPEB+TWO	

;READ MA REG AND CHECK IT FOR ADDRESS 100000

7215	4414	TOP2B:	JSR	RMA ;GET MA REG DATA
7216	126620		SUBZR	1,1 ;MAKE 100000
7217	106415		SUB#	0,1,SNR ;AT 100000?



```

7220 2114      JMP    @NTST    ; YES, BYE!
7221 44305    STA    1,DIAN  ; NO, SAVE IT
7222 6166     JSR    @ERROR
7223 7273     TOPEB+THREE
7224 2114     JMP    @NTST

```

```
7225 100022 WMAP: 100022
```

```
; READ MA REGISTER SUBROUTINE
; SET ALTERNATE MODE 1 TO ACCESS MA REGISTER, RESTORE TO NORMAL AFTERWARDS
```

```

7226 2600     NOOP
7227 2200     ALT1
7230 0        0
7231 54777 RMA: STA    3,-1
7232 24211    LDA    1,DDOA
7233 44414    STA    1,DDOAT ; SET UP DOA INSTRUCTION
7234 20121    LDA    0,DRVF ; DRIVE #
7235 24772    LDA    1,RMA-2
7236 123000   ADD    1,0 ; ADD ALT MODE 1 COMMAND
7237 4410     JSR    DDOAT ; SET IT
7240 4237     JSR    DDIA1 ; READ MA REG
7241 111000   MOV    0,2 ; SAVE IN AC2
7242 20764    LDA    0,RMA-3 ; NO-OP COMMAND
7243 123000   ADD    1,0
7244 4403     JSR    DDOAT ; RESTORE MODE
7245 141000   MOV    2,0
7246 2762     JMP    @RMA-1 ; RETURN

```

```

7247 0 DDOAT: 0
7250 1400     JMP    0,3

```

```
; **** ERROR MESSAGE BLOCKS ****
```

```

7251 7170 TOPEB: TOP1
7252 7162    TOP
7253 23206   WDT
7254 6       6
7255 1       1
7256 15      15

7257 7177    TOP1A+2; 1
7260 7162    TOP
7261 23362   IMA
7262 123403  @WD1T
7263 4       4
7264 15      15

7265 7210    TOP2A ; 2
7266 7204    TOP2
7267 23214   RDT
7270 6       6
7271 1       1
7272 15      15

7273 7217    TOP2B+2; 3
7274 7210    TOP2A
7275 23362   IMA
7276 124656  @RD1T

```

7277  
7300

4  
15

4  
15

;\*\*\*\* RW TIMEOUT \*\*\*\*

;CHECK RW TIMEOUT STATUS INDICATION AND MEASURE TIME.  
;WRITE A HEADER ADDRESS OF SURF 0, SECT 0 ONTO SECTORS 1, 2, 4, 8 AND  
;16. THE WHEN ATTEMPTING TO READ DATA FROM ONE OF THOSE SECTORS, THE  
;RESULT IS RW TIMEOUT DUE TO SECTOR 0 HEADER IS NONEXISTENT

7301	6177	RWTM:	JSR	@SET
7302	24704		RWT	
7303	7306		RWT	
7304	7506		SSAS	
7305	0		0	

;PRESET HEADER IN WBUF AND INITIALIZE SECTOR ADDRESS POINTER

7306	34115	RWT:	LDA	3,WBUF	;WBUF ADDR
7307	102400		SUB	0,0	
7310	41400		STA	0,0,3	
7311	41401		STA	0,1,3	
7312	41402		STA	0,2,3	
7313	24425		LDA	1,SCL1	;TEST SECTOR LIST
7314	44142		STA	1,SAVP	;SET POINTER
7315	26142	RWT1:	LDA	1,@SAVP	;GET SECT ADDR
7316	125112		MOVL#	1,1,SZC	;END OF LIST?
7317	422		JMP	RWT2	;YES, EXIT
7320	44403		STA	1,RWT1A	;NO, SET IT
7321	6205		JSR	@DADD	;SET DISK ADDR
7322	0		0		
7323	0	RWT1A:	0		
7324	1		1		

;WRITE HEADER TO DESIGNATED SECTOR

7325	6170		JSR	@RWCMD
7326	600		WHDR	
7327	30462		WBUF	
7330	144		144	

;DONE, CHECK CONTROL STATUS

7331	6202	RWT1B:	JSR	@CKCS
7332	40000		RWDN	
7333	7336		RWT1C	;STATUS GOOD
7334	6166		JSR	@ERROR ;STATUS BAD
7335	7450		RWTEB	

;BUMP POINTER TO DO NEXT SECTOR

7336	10142	RWT1C:	ISZ	SAVP
7337	756		JMP	RWT1

7340	7133	SCL1:	SCL+2
------	------	-------	-------

;HEADERS WRITTEN, RESET ADDRESS POINTER AND PRIME RBUF WITH ALL ONES

7341	24777	RWT2:	LDA	1,SCL1
7342	44142		STA	1,SAVP ;RESET LIST POINTER
7343	26142		LDA	1,@SAVP ;GET SECT ADDR
7344	125112		MOVL#	1,1,SZC ;END OF LIST?

```

7345     452      JMP      RWT3      ; YES
7346   44130     STA      1,SCTN   ; SET ADDR
7347   6176     RWT2A: JSR     @PRIME
7350   50462     RBUF
7351  177777     -1
7352  177777     -1
7353     400      400

```

; ISSUE READ DATA COMMAND WITH TIMER SET FOR 2 SECONDS

```

7354   6170      JSR     @RWCMD
7355     0        READ
7356   50462     RBUF
7357   3720      3720

```

; DONE, CHECK FOR RW DONE + TIMEOUT + RW FAULT STATUS

```

7360   6202     RWT2B: JSR     @CKCS
7361   40005     RWDN+RWTD+RWFT
7362   7365     RWT2C   ; CORRECT STATUS
7363   6166     JSR     @ERROR   ; INCORRECT STATUS
7364   7456     RWTEB+ONE

```

; CALCULATE AND CHECK TIMING FOR BETWEEN 0.5 TO 2 SEC

```

7365  20772     RWT2C: LDA     0,RWT2B-1; STARTING COUNT
7366  24144     LDA     1,CLOK   ; REMAINING COUNT
7367  122450     SUB     1,0       ; TIMING COUNT
7370  24445     LDA     1,LOW    ; LOWER LIMIT
7371  34445     LDA     3,HIGH   ; UPPER LIMIT
7372  116432     SUBZ#   0,3,SZC   ; SKIP IF >2 SEC
7373  106032     ADCZ#   0,1,SZC   ; SKIP IF =>.5 SEC
7374     413     JMP     RWT2F   ; OUT OF TOLERANCE

```

; VERIFY THAT NO DATA HAS BEEN READ INTO RBUF

```

7375  30116     RWT2D: LDA     2,RBUF   ; RBUF ADDR
7376  24064     LDA     1,N400   ; WORD COUNT
7377  21000     LDA     0,0,2
7400  101103     MOVL    0,0,SNC   ; ALL CLEAR?
7401     413     JMP     RWT2G   ; NO, PROBLEM!
7402  151400     INC     2,2
7403  125404     INC     1,1,SZR   ; FINISH?
7404     773     JMP     RWT2D+2 ; NOT QUITE
7405  10142     RWT2E: ISZ    SAVP   ; YES, BUMP LIST POINTER
7406     735     JMP     RWT2+2

```

; OUT OF TOLERANCE ERROR PRINTOUT

```

7407  40404     RWT2F: STA     0,TMTS
7410   6166     JSR     @ERROR
7411   7464     RWTEB+TWO
7412     773     JMP     RWT2E

```

```

7413     0      0 TMTS: 0

```

```

7414   6166     RWT2G: JSR     @ERROR
7415   7472     RWTEB+THREE
7416   767     JMP     RWT2E

```

---  
; RE-FORMAT THIS TRACK BEFORE LEAVING TEST

```
7417 6205 RWT3: JSR @DADD ; SET DISK ADDR
7420 0 0
7421 0 0
7422 100123 @STRK
7423 6170 JSR @RWCMD ; ISSUE FORMAT COMMAND
7424 1400 FMAT
7425 30462 WBUFF
7426 50 50
```

; DONE, CHECK CONTROL STATUS

```
7427 6202 RWT3A: JSR @CKCS
7430 40000 RWDN
7431 7434 RWT3B ; RIGHT ON!
7432 6166 JSR @ERROR ; OOPS!
7433 7500 RWTEB+FOUR
7434 2114 RWT3B: JMP @NTST

7435 764 LOW: 764
7436 3720 HIGH: 3720
```

; OUT OF TOLERANCE ERROR PRINTOUT

```
7437 0 0
7440 54777 TIMM: STA 3, -1
7441 6155 JSR @PRINT
7442 123015 @TOAT
7443 24750 LDA 1, TMTS ; MEASURED TIME
7444 6164 JSR @DECN ; PRINT IT
7445 6155 JSR @PRINT
7446 123277 @MS
7447 2770 JMP @TIMM-1
```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```
7450 7331 RWTEB: RWT1B
7451 7325 RWT1A+2
7452 23117 WHD
7453 6 6
7454 10 10
7455 15 15

7456 7360 RWT2B ; 1
7457 7347 RWT2A
7460 23214 RDT
7461 6 6
7462 10 10
7463 15 15

7464 7365 RWT2C ; 2
7465 7347 RWT2A
7466 22761 TMG
7467 122774 @RDNX
7470 20 20
7471 123024 @TOLR
```

7472 7375 RWT2D ; 3  
7473 7347 RWT2A  
7474 22774 RDNX  
7475 6 6  
7476 123036 @DRB  
7477 123051 @NFST

7500 7427 RWT3A ; 4  
7501 7417 RWT3  
7502 23070 RSTH  
7503 6 6  
7504 10 10  
7505 15 15

;\*\*\*\* SURF-SECT ADDRESS ERROR STATUS \*\*\*\*

;TEST SURF-SECT ADDRESS ERROR STATUS INDICATION.

;WRITE HEADER WITH A WRONG SURF-SECT ADDRESS TO SECTOR 0. THEN WHEN

;ATTEMPTING TO READ DATA FROM IT, THE RESULT IS SURF-SECT ERROR STATUS

```
7506 6177 SSAS: JSR @SET
7507 24712 SSAT
7510 7513 SSA
7511 7611 CAES
7512 0 0
```

;REFORMAT CYL 0, HEAD 0

;PRESET WBUFF WORDS 0 AND 2 TO ALL ZEROS. SET TEST ADDRESS LIST POINTER.

;FETCH THE 'WRONG' SURF-SECT ADDRESS FROM LIST AND INSERT IT IN WORD 1

```
7513 6205 SSA: JSR @DADD
7514 0 0
7515 0 0
7516 100123 @STRK
7517 6170 JSR @RWCMD
7520 1400 FMAT
7521 30462 WBUFF
7522 50 50
7523 30115 LDA 2,WBUF ; WBUFF ADDR
7524 126400 SUB 1,1
7525 45000 STA 1,0,2 ; WORD 0 = 0
7526 45002 STA 1,2,2 ; WORD 2 = 0
7527 6205 JSR @DADD ; SET DISK ADDR
7530 0 0
7531 0 0
7532 1 1
7533 34433 LDA 3,SSAL ; TEST ADDR LIST
7534 54142 STA 3,SAVP ; SET POINTER
7535 22142 SSA1: LDA 0,@SAVP ; FETCH ADDR
7536 101112 MOVL# 0,0,SZC ; END OF LIST?
7537 2114 JMP @NTST ; YES, EXIT
7540 41001 STA 0,1,2 ; NO, SET WORD 1
```

;WRITE HEADER IN WBUFF TO SURF 0, SECT 0

```
7541 6170 SSA1A: JSR @RWCMD
7542 600 WHDR
7543 30462 WBUFF
7544 24 24
```

;DONE, CHECK CONTROL STATUS

```
7545 6202 SSA1B: JSR @CKCS
7546 40000 RWDN
7547 7552 SSA1C ; LOOKS FINE
7550 6166 JSR @ERROR ; DOES NOT LOOK GOOD
7551 7575 SSAEB
```

;NOW TRY TO READ DATA FROM SECTOR 0

```
7552 6170 SSA1C: JSR @RWCMD
7553 0 READ
7554 50462 RBUFF
```

7555 24 24

; DONE, CHECK STATUS FOR SURF-SECT ADDRESS ERROR

```
7556 6202 SSA1D: JSR @CKCS
7557 40021 RWDN+SSAE+RWFT
7560 7563 SSA2 ; RIGHT STATUS
7561 6166 JSR @ERROR ; WRONG STATUS
7562 7603 SSAEB+ONE
```

; BUMP LIST POINTER TO DO NEXT SECTOR

```
7563 10142 SSA2: ISZ SAVP
7564 30115 LDA 2,WBUF
7565 750 JMP SSA1
```

; SURF-SECT ADDRESS TEST LIST

```
7566 7567 SSAL: .+1
7567 2000 2000 ; SURF 1
7570 4040 4040 ; SURF 2, SECT 1
7571 10000 10000 ; SURF 4
7572 20000 20000 ; SURF 10
7573 40000 40000 ; SURF 20
7574 177777 -1 ; END
```

; \*\*\*\* ERROR MESSAGE BLOCK \*\*\*\*

```
7575 7545 SSAEB: SSA1B
7576 7541 SSA1A
7577 23117 WHD
7600 6 6
7601 10 10
7602 15 15
```

```
7603 7556 SSA1D
7604 7541 SSA1A
7605 23101 ST11
7606 6 6
7607 10 10
7610 15 15
```



;\*\*\*\* CYLINDER ADDRESS ERROR STATUS \*\*\*\*

;TEST CYLINDER ADDRESS ERROR STATUS INDICATION  
;WRITE A HEADER WITH A WRONG CYLINDER ADDRESS TO CYLINDER 0, SURF 0,  
;SECT 0. WHEN ATTEMPTING TO READ DATA FROM IT, THE RESULT IS CYLINDER  
;ADDRESS ERROR.

7611	6177	CAES:	JSR	@SET
7612	24732		CAET	
7613	7616		CAE	
7614	7705		HUBS	
7615	0		0	

;SET UP TEST ADDRESS POINTER AND FETCH ADDRESS FROM LIST TO SET UP  
;TEST HEADER IN WBUFF

7616	24437	CAE:	LDA	1,CYTL	
7617	44142		STA	1,SAVP	;SET LIST POINTER
7620	102400		SUB	0,0	
7621	30115		LDA	2,WBUF	
7622	41001		STA	0,1,2	;WORDS 1 & 2 = 0
7623	41002		STA	0,2,2	
7624	26142	CAE1:	LDA	1,@SAVP	;FETCH 'WRONG' CYL ADDR
7625	125112		MOVL#	1,1,SZC	;END OF LIST?
7626	2114		JMP	@NTST	;YES, EXIT
7627	45000		STA	1,0,2	;NO, SET WORD 0

;WRITE HEADER TO SURF 0, SECT 0

7630	6170		JSR	@RWCMD
7631	600		WHDR	
7632	30462		WBUFF	
7633	24		24	

;DONE, CHECK CONTROL STATUS

7634	6202	CAE1A:	JSR	@CKCS
7635	40000		RWDN	
7636	7641		CAE1B	
7637	6166		JSR	@ERROR ;BAD STATUS
7640	7671		CAEEB	

;NOW TRY TO READ DATA FROM THAT SECTOR

7641	6170	CAE1B:	JSR	@RWCMD
7642	0		READ	
7643	50462		RBUFF	
7644	24		24	

;DONE, CHECK FOR RW DONE + CYLINDER ADDR ERROR + RW FAULT STATUS

7645	6202	CAE1C:	JSR	@CKCS
7646	40041		RWDN+CADR+RWFT	
7647	7652		CAE2	;STATUS OK
7650	6166		JSR	@ERROR ;STATUS INCORRECT
7651	7677		CAEEB+ONE	

;BUMP LIST POINTER THEN CONTINUE

7652 10142 CAE2: ISZ SAVP  
7653 30115 LDA 2,WBUF  
7654 750 JMP CAE1

; CYLINDER ADDRESS TEST LIST

7655	7656	CYTL:	. +1	
7656	1		1	
7657	2		2	
7660	4		4	
7661	10		10	
7662	20		20	
7663	40		40	
7664	100		100	
7665	200		200	
7666	400		400	
7667	1000		1000	
7670	177777		-1	; END

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

7671	7634	CAEEB:	CAE1A	
7672	7624		CAE1	
7673	23117		WHD	
7674	6		6	
7675	10		10	
7676	15		15	

7677	7645		CAE1C	
7700	7624		CAE1	
7701	24732		CAET	
7702	6		6	
7703	10		10	
7704	15		15	

;\*\*\*\* HEADER WORD UNUSED BITS \*\*\*\*

;VERIFY THAT THE UNUSED BITS OR FIELD IN EACH HEADER WORD HAS NO  
;EFFECT ON AN IO OPERATION

```
7705 6177 HUBS: JSR @SET
7706 24751 HUBT
7707 7712 HUB
7710 10024 FCCB
7711 0 0
```

;PUT IN WBUFF A HEADER CONTAINING ALL UNUSED BITS IN EACH WORD

```
7712 34115 HUB: LDA 3,WBUF
7713 24457 LDA 1,UNB1
7714 45400 STA 1,0,3 ;WORD 0 = 36000
7715 24456 LDA 1,UNB2
7716 45401 STA 1,1,3 ;WORD 1 = 100037
7717 102000 ADC 0,0
7720 41402 STA 0,2,3 ;WORD 2 = 177777
```

;WRITE THIS HEADER TO SURF 0, SECT 0

```
7721 6170 HUB1: JSR @RWCMD
7722 600 WHDR
7723 30462 WBUFF
7724 144 144
```

;DONE, CHECK STATUS FOR NORMAL INDICATION

```
7725 6202 HUB1A: JSR @CKCS
7726 40000 RWDN
7727 7732 HUB1B ;NORMAL STATUS
7730 6166 JSR @ERROR ;NOT NORMAL
7731 7774 HUBEB
```

;PRIME RBUFF WITH ALL ONES DATA BEFORE READING DATA

```
7732 6176 HUB1B: JSR @PRIME
7733 50462 RBUFF
7734 177777 -1
7735 177777 -1
7736 400 400
7737 6170 JSR @RWCMD
7740 0 READ
7741 50462 RBUFF
7742 74 74
```

;DONE, CHECK CONTROL STATUS AND DISK ADDRESS POINTER

```
7743 6202 HUB1C: JSR @CKCS
7744 40000 RWDN
7745 7750 HUB1D ;GOOD STATUS VECTOR
7746 6166 JSR @ERROR ;BAD STATUS
7747 10002 HUBEB+ONE
7750 6204 HUB1D: JSR @CKDA
7751 40 40
7752 7755 HUB2 ;GOOD POINTER VECTOR
7753 6166 JSR @ERROR ;NG POINTER
```

```

---
7754 10010 HUBEB+TWO
; VERIFY DATA IN RBUF IS READ IN OK

7755 24064 HUB2: LDA 1,N400
7756 34116 LDA 3,RBUF
7757 54154 STA 3,RBFP ; SET READ POINTER
7760 22154 HUB2A: LDA 0,@RBFP ; GET A WORD
7761 101004 MOV 0,0,SZR ; ALL ZEROS?
7762 405 JMP HUB2B ; NO
7763 10154 ISZ RBFP
7764 125404 INC 1,1,SZR ; ALL DONE?
7765 773 JMP HUB2A
7766 2114 JMP @NTST ; YES, EXIT

7767 6166 HUB2B: JSR @ERROR
7770 10016 HUBEB+THREE
7771 2114 JMP @NTST

7772 36000 UNB1: 36000
7773 100037 UNB2: 100037

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

7774 7725 HUBEB: HUB1A
7775 7721 HUB1
7776 23117 WHD
7777 6 6
10000 1 1
10001 15 15

10002 7743 HUB1C ; 1
10003 7732 HUB1B
10004 23214 RDT
10005 6 6
10006 1 1
10007 15 15

10010 7750 HUB1D ; 2
10011 7732 HUB1B
10012 22714 RWDA
10013 6 6
10014 3 3
10015 17 17

10016 7760 HUB2A ; 3
10017 7732 HUB1B
10020 23214 RDT
10021 16 16
10022 14 14
10023 122742 @ASWB

```

;\*\*\*\* FORMAT CROSS CYLINDER \*\*\*\*

;TEST THE ABILITY OF THE CONTROLLER TO FORMAT ACROSS CYLINDER  
;BOUNDARIES

```
10024 6177 FCCB: JSR @SET
10025 24765 FCCT
10026 10031 FCC
10027 10342 FCYL
10030 0 0
```

; SET UP A HEADER IN WBUFF, SET TO CYLINDER 1

```
10031 30115 FCC: LDA 2,WBUF
10032 102400 SUB 0,0
10033 41000 STA 0,0,2 ;WORDS 0 & 1 = 0
10034 41001 STA 0,1,2
10035 100000 COM 0,0
10036 41002 STA 0,2,2 ;WORD 2 = 177777
10037 126520 FCC1: SUBZL 1,1
10040 44127 STA 1,CYLN ;SET CYL 1
10041 6167 JSR @DRCMD
10042 400 SEEK
10043 50 50
```

; DONE, CHECK DRIVE STATUS THEN WRITE HEADER TO SURF 0, SECT 0

```
10044 6203 FCC1A: JSR @CKDS
10045 10000 DRDY
10046 10051 FCC1B ;STATUS OK
10047 6166 JSR @ERROR ;STATUS NG
10050 10232 FCCEB
10051 6170 FCC1B: JSR @RWCMD ;WRITE HEADER
10052 600 WHDR
10053 30462 WBUFF
10054 24 24
```

; DONE, CHECK CONTROL STATUS AND DISK ADDRESS WORD

```
10055 6202 FCC1C: JSR @CKCS
10056 40000 RWDN
10057 10062 FCC1D ;OK VECTOR
10060 6166 JSR @ERROR
10061 10240 FCCEB+ONE
10062 6204 FCC1D: JSR @CKDA
10063 40 40
10064 10067 FCC1E ;OK VECTOR
10065 6166 JSR @ERROR ;POINTER ERROR
10066 10246 FCCEB+TWO
```

; READ AND VERIFY HEADER JUST WROTE

```
10067 6170 FCC1E: JSR @RWCMD
10070 3600 RHDR
10071 50462 RBUFF
10072 24 24
10073 6173 FCC1F: JSR @HCMP ;CALL HEADER COMPARE
10074 1 1
10075 10100 FCC2 ;COMPARE OK
```

10076 6166 JSR @ERROR ; NOT OK  
10077 10254 FCCEB+THREE

; SEEK TO CYLINDER 0 AND CHECK DRIVE STATUS

10100 102400 FCC2: SUB 0,0  
10101 40127 STA 0,CYLN ; SET CYL 0  
10102 6167 JSR @DRCMD  
10103 400 SEEK  
10104 24 24  
10105 6203 FCC2A: JSR @CKDS  
10106 10000 DRDY  
10107 10112 FCC2B ; CORRECT STATUS VECTOR  
10110 6166 JSR @ERROR  
10111 10262 FCCEB+FOUR

; READ AND VERIFY HEADER

10112 6170 FCC2B: JSR @RWCMD  
10113 3600 RHDR  
10114 50462 RBUFF  
10115 24 24

; DONE, CHECK STATUS AND DISK ADDRESS

10116 6202 FCC2C: JSR @CKCS  
10117 40000 RWDN  
10120 10123 FCC2D ; OK  
10121 6166 JSR @ERROR ; NOT OK  
10122 10270 FCCEB+FIVE  
10123 6204 FCC2D: JSR @CKDA  
10124 40 40  
10125 10130 FCC3 ; OK  
10126 6166 JSR @ERROR ; NG  
10127 10276 FCCEB+SIX

; CHECK TYPE OF DISK FIRST THEN SET LAST REMOVABLE SURFACE ADDRESS

10130 102400 FCC3: SUB 0,0  
10131 24120 LDA 1,DSKT ; GET DRIVE TYPE  
10132 125005 MOV 1,1,SNR ; SMD?  
10133 20122 LDA 0,LSFN ; YES, GET LAST SURF #  
10134 40402 STA 0,+2 ; SET IT  
10135 6205 JSR @DADD ; SET DISK STARTING ADDR  
10136 0 0  
10137 2 2  
10140 40 40

; FORMAT 32 SECTORS STARTING FROM THE LAST REMOVABLE SURFACE, SECT 2

10141 6170 FCC3A: JSR @RWCMD  
10142 1400 FMAT  
10143 30462 WBUFF  
10144 74 74

; DONE, CHECK STATUS AND DISK ADDRESS POINTER

10145 6201 FCC3B: JSR @CKCK  
10146 40000 RWDN

```

10147 10152      FCC3C ; STATUS GOOD VECTOR
10150 6166      JSR   @ERROR
10151 10304     FCCEB+SEVEN
10152 6204 FCC3C: JSR   @CKDA
10153 100       100
10154 10157     FCC3D
10155 6166      JSR   @ERROR ; BAD ADDR
10156 10312     FCCEB+EIGHT

```

```

; HEAD IS NOW RESTING OVER CYLINDER 1, READ HEADER FROM SURFACE 0,
; SECTOR 0 INTO RBUFF

```

```

10157 6205 FCC3D: JSR   @DADD ; SET UP DISK ADDR
10160 0       0
10161 0       0
10162 1       1
10163 6170      JSR   @RWCMD ; READ HEADER
10164 3600      RHDR
10165 50462     RBUFF
10166 24       24

```

```

; DONE, CHECK STATUS AND DISK ADDRESS

```

```

10167 6202 FCC3E: JSR   @CKCS
10170 40000     RWDN
10171 10174     FCC3F ; GOOD STATUS VECTOR
10172 6166      JSR   @ERROR ; BAD STATUS
10173 10320     FCCEB+NINEE
10174 6204 FCC3F: JSR   @CKDA
10175 40       40
10176 10201     FCC3G ; ADDR OK
10177 6166      JSR   @ERROR ; ADDR DISAGREE
10200 10326     FCCEB+TEN

```

```

; CHECK HEADER IN RBUFF TO VERIFY CROSS CYLINDER FORMATTING

```

```

10201 34116 FCC3G: LDA   3,RBUF ; RBUFF ADDR
10202 21400     LDA   0,0,3 ; GET WORD 0
10203 126520    SUBZL 1,1
10204 106414    SUB#  0,1,SZR ; SAYS CYL 1?
10205 420       JMP   FCC3H ; NO
10206 175400    INC   3,3
10207 126400    SUB   1,1 ; MAKE REF
10210 21400     LDA   0,0,3 ; GET WORD 1
10211 122404    SUB   1,0,SZR ; ZERO?
10212 413       JMP   FCC3H ; NO
10213 175400    INC   3,3
10214 21400     LDA   0,0,3 ; GET WORD 2
10215 122404    SUB   1,0,SZR ; ZERO ALSO?
10216 407       JMP   FCC3H ; NO
10217 175400    INC   3,3
10220 175400    INC   3,3
10221 21400     LDA   0,0,3
10222 122404    SUB   1,0,SZR ; WORD 4 = 0?
10223 402       JMP   FCC3H ; NO
10224 2114      JMP   @NTST ; ALL OK, EXIT

10225 54154 FCC3H: STA   3,RBFP ; SAVE POINTER
10226 44141     STA   1,ACN ; SAVE REF

```

```

10227 6166 JSR @ERROR
10230 10334 FCCEB+ELEVN
10231 2114 JMP @NTST

```

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*;

```

10232 10044 FCCEB:FCC1A
10233 10037 FCC1
10234 23126 DSCK
10235 15 15
10236 2 2
10237 16 16

10240 10055 FCC1C ; 1
10241 10044 FCC1A
10242 23117 WHD
10243 6 6
10244 1 1
10245 15 15

10246 10062 FCC1D ; 2
10247 10044 FCC1A
10250 22714 RWDA
10251 6 6
10252 3 3
10253 17 17

10254 10073 FCC1F ; 3
10255 10044 FCC1A
10256 22730 HWC
10257 16 16
10260 14 14
10261 122742 @ASWB

10262 10105 FCC2A ; 4
10263 10100 FCC2
10264 23126 DSCK
10265 15 15
10266 2 2
10267 16 16

10270 10116 FCC2C ; 5
10271 10112 FCC2B
10272 23140 RHD
10273 6 6
10274 1 1
10275 15 15

10276 10123 FCC2D ; 6
10277 10112 FCC2B
10300 23140 RHD
10301 6 6
10302 3 3
10303 17 17

10304 10145 FCC3B ; 7
10305 10141 FCC3A
10306 24765 FCCT
10307 6 6

```



10310	1	1
10311	15	15
10312	10152	FCC3C ; 8
10313	10141	FCC3A
10314	24765	FCCT
10315	6	6
10316	3	3
10317	17	17
10320	10167	FCC3E ; 9
10321	10157	FCC3D
10322	23140	RHD
10323	6	6
10324	1	1
10325	15	15
10326	10174	FCC3F ; 10
10327	10157	FCC3D
10330	23140	RHD
10331	6	6
10332	3	3
10333	17	17
10334	10201	FCC3G ; 11
10335	10157	FCC3D
10336	22730	HWC
10337	124765	@FCCT
10340	14	14
10341	21	21

;\*\*\*\* FORMAT A CYLINDER \*\*\*\*

; USE THE FORMAT FUNCTION OF THE CONTROLLER TO FORMAT ALL SURFACES  
; OF CYLINDER 0 AND SURFACE 0 OF CYLINDER 1

10342	6177	FCYL:	JSR	@SET
10343	25000		FCYLT	
10344	10347		FCY	
10345	10554		FMLS	
10346	0		0	

; SEEK TO CYLINDER 0, SET STARTING ADDRESS SURFACE 0, SECTOR 0

10347	102400	FCY:	SUB	0,0
10350	40127		STA	0,CYLN
10351	6205		JSR	@DADD ; SET ADDR
10352	0		0	
10353	0		0	
10354	100123		@STRK	

; FORMAT A TRACK

10355	6167	FCY1:	JSR	@DRCMD
10356	400		SEEK	
10357	50		50	
10360	6170		JSR	@RWCMD
10361	1400		FMAT	
10362	30462		WBUFF	
10363	74		74	

; DONE, CHECK CONTROL STATUS

10364	6202	FCY1A:	JSR	@CKCS
10365	40000		RWDN	
10366	10371		FCY1B	; STATUS OK
10367	6166		JSR	@ERROR ; NOT SO
10370	10510		FCYEB	

; CHECK SURFACE NUMBER TO SEE IF DONE THE LAST ONE. GO TO CYLINDER  
; 1 IF SO. ELSE CONTINUE

10371	20126	FCY1B:	LDA	0,SFCN ; CURRENT SURF #
10372	24122		LDA	1,LSFN ; LAST SURF #
10373	106415		SUB#	0,1,SNR ; DONE CYL 0?
10374	403		JMP	FCY1C ; YES
10375	10126		ISZ	SFCN ; NO, NEXT ONE
10376	757		JMP	FCY1

; SEEK TO CYLINDER 1 TO FORMAT SURFACE 0 ONLY

10377	102520	FCY1C:	SUBZL	0,0
10400	40127		STA	0,CYLN
10401	126400		SUB	1,1
10402	44126		STA	1,SFCN ; SURF 0
10403	6167		JSR	@DRCMD
10404	400		SEEK	
10405	24		24	
10406	6170		JSR	@RWCMD
10407	1400		FMAT	

---  
10410 30462 WBUFF  
10411 74 74

; DONE, CHECK CONTROL STATUS

10412 6202 FCY1D: JSR @CKCS  
10413 40000 RWDN  
10414 10417 FCY2  
10415 6166 JSR @ERROR  
10416 10516 FCYEB+ONE

; NOW SEEK BACK TO CYLINDER 0, PRIME RBUFF WITH ALL ONES THEN READ FORMAT

10417 126400 FCY2: SUB 1, 1  
10420 44127 STA 1, CYLN ; CYL 0  
10421 44126 STA 1, SFCN ; SURF 0  
10422 6167 JSR @DRCMD  
10423 400 SEEK  
10424 24 24  
10425 6176 FCY2A: JSR @PRIME  
10426 50462 RBUFF  
10427 177777 -1  
10430 177777 -1  
10431 300 300  
10432 6170 JSR @RWCMD  
10433 3600 RHDR  
10434 50462 RBUFF  
10435 74 74

; DONE, CHECK STATUS

10436 6202 FCY2B: JSR @CKCS  
10437 40000 RWDN  
10440 10443 FCY2C ; GDD STATUS VECTOR  
10441 6166 JSR @ERROR ; BAD STATUS  
10442 10524 FCYEB+TWO

; GENERATE HEADERS IN WBUFF AND USE AS REFERENCE FOR COMPARISON

10443 6172 FCY2C: JSR @HDRG ; GENERATE HEADERS  
10444 100123 @STRK  
10445 6173 JSR @HCMP ; COMPARE THEM  
10446 100123 @STRK  
10447 10452 FCY2D ; NO ERROR RETURN  
10450 6166 JSR @ERROR  
10451 10532 FCYEB+THREE

; CHECK SURFACE NUMBER TO SEE IF DONE THE LAST SURFACE. DO NEXT ONE  
; IF NOT

10452 20126 FCY2D: LDA 0, SFCN ; CURRENT SURF #  
10453 24122 LDA 1, LSFN ; LAST SURF #  
10454 106405 SUB 0, 1, SNR ; DONE LAST ONE?  
10455 403 JMP FCY3 ; YES  
10456 10126 ISZ SFCN ; NO, BUMP IT  
10457 743 JMP FCY2+3

; SEEK TO CYLINDER 1 TO READ SURFACE 0 HEADERS

```

10460 126520 FCY3: SUBZL 1,1
10461 44127 STA 1,CYLN
10462 125220 MOVZR 1,1
10463 44126 STA 1,SFCN ; SURF 0
10464 6167 JSR @DRCMD
10465 400 SEEK
10466 24 24
10467 6170 JSR @RWCMD
10470 3600 RHDR
10471 50462 RBUFF
10472 74 74

```

; DONE, CHECK CONTROL STATUS

```

10473 6202 FCY3A: JSR @CKCS
10474 40000 RWDN
10475 10500 FCY3B ; GOOD STATUS RETURN
10476 6166 JSR @ERROR ; BAD STATUS
10477 10540 FCYEB+FOUR

```

; GENERATE HEADERS OF CYLINDER 1, SURFACE 0 AS REFERENCE

```

10500 6172 FCY3B: JSR @HDRG ; GENERATE HEADERS
10501 100123 @STRK
10502 6173 JSR @HCMP ; COMPARE THEM
10503 100123 @STRK
10504 10507 FCY3C ; NO ERROR VECTOR
10505 6166 JSR @ERROR ; BAD HEADER
10506 10546 FCYEB+FIVE
10507 2114 FCY3C: JMP @NTST

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

10510 10364 FCYEB: FCY1A
10511 10355 FCY1
10512 23146 FMKO
10513 6 6
10514 1 1
10515 15 15

10516 10412 FCY1D ; 1
10517 10377 FCY1C
10520 23155 FMK1
10521 6 6
10522 1 1
10523 15 15

10524 10436 FCY2B ; 2
10525 10425 FCY2A
10526 23164 RMKO
10527 6 6
10530 1 1
10531 15 15

10532 10445 FCY2C+2; 3
10533 10425 FCY2A
10534 22730 HWC
10535 16 16
10536 14 14

```

---  
10537 122742 @ASWB  
  
10540 10473 FCY3A ; 4  
10541 10460 FCY3  
10542 23175 RMK1  
10543 6 6  
10544 1 1  
10545 15 15  
  
10546 10502 FCY3B+2; 5  
10547 10460 FCY3  
10550 22730 HWC  
10551 16 16  
10552 14 14  
10553 122742 @ASWB

;\*\*\*\* FORMAT LAST REMOVABLE SURFACE \*\*\*\*

; USE THE FORMAT FUNCTION OF THE CONTROLLER TO FORMAT THE ENTIRE  
; SURFACE OF THE LAST REMOVABLE PLATTER

```
10554 6177 FMLS: JSR @SET
10555 25011      FMLST
10556 10561      FML
10557 10630      RMLS
10560 0          0
```

; GET LAST SURFACE NUMBER FROM LOOKUP TABLE AND SET IT

```
10561 102400 FML: SUB 0,0
10562 24120   LDA 1,DSKT ; GET DISK TYPE
10563 125005  MOV 1,1,SNR ; WHAT KIND?
10564 20122   LDA 0,LSFN ; REMOVABLE, GET LAST SURF #
10565 40402   STA 0,+2 ; SET IT
10566 6205    JSR @DADD
10567 0       0
10570 0       0
10571 100123  @STRK
```

; FORMAT ONE TRACK AT A TIME UNTIL WHOLE SURFACE IS COVERED

```
10572 102400 SUB 0,0
10573 40127   STA 0,CYLN ; STARTING CYL 0
10574 6167 FML1: JSR @DRCMD
10575 400     SEEK
10576 144     144
10577 6170   JSR @RWCMD
10600 1400   FMAT
10601 30462  WBUFF
10602 74     74
```

; DONE, CHECK STATUS WORD

```
10603 6202 FML1A: JSR @CKCS
10604 40000   RWDN
10605 10610   FML1B ; GOOD STATUS RETURN
10606 6166    JSR @ERROR
10607 10622   FMLEB
```

; CHECK CYLINDER ADDRESS TO SEE IF DONE THE LAST ONE

```
10610 20127 FML1B: LDA 0,CYLN
10611 24124   LDA 1,LCYN
10612 106415  SUB# 0,1,SNR ; FINISH?
10613 403     JMP .+3 ; YES, EXIT
10614 10127   ISZ CYLN ; NO, BUMP CYL ADDR
10615 757     JMP FML1
10616 6167   JSR @DRCMD ; DO RECAL
10617 200     RCAL
10620 1750   1750
10621 2114   JMP @NTST ; NEXT TEST
```

;\*\*\*\* ERROR MESSAGE BLOCK \*\*\*\*

```
10622 10603 FMLEB: FML1A
```

-----  
10623 10574 FML1  
10624 22701 RWDS  
10625 6 6  
10626 1 1  
10627 15 15

;\*\*\*\* READ FORMAT FROM LAST SECTORS \*\*\*\*

; READ FORMAT FROM LAST SECTOR OF EACH CYLINDER IN THE LAST REMOVABLE  
; SURFACE. CHECK SEEK TIME, DRIVE DONE STATUS ON EACH READ

```
10630 6177 RMLS: JSR @SET
10631 25030 RMLST
10632 10635 RML
10633 10751 VRFY
10634 0 0
```

; SET LAST SURFACE AND SECTOR NUMBERS FOR TEST

```
10635 102400 RML: SUB 0,0
10636 24120 LDA 1,DSKT
10637 125005 MOV 1,1,SNR ;WHAT TYPE OF DISK?
10640 20122 LDA 0,LSFN ;REMOVABLE, GET LAST SURF #
10641 40126 STA 0,SFCN ;SET IT
10642 24123 LDA 1,STRK ;SECT COUNT
10643 124400 NEG 1,1 ;SUBTRACT 1
10644 124000 COM 1,1
10645 44130 STA 1,SCTN ;SET LAST SECT ADDR
10646 102520 SUBZL 0,0
10647 40131 STA 0,SCNT ;SET SECT CNT TO 1
10650 101220 MOVZR 0,0
10651 40127 STA 0,CYLN ;STARTING CYL 0
```

; SEEK TO CYLINDER

```
10652 6167 RML1: JSR @DRCMD
10653 400 SEEK
10654 50 50
```

; DONE, COMPUTE THE SEEK TIME AND CHECK AGAINST LIMIT

```
10655 20777 LDA 0,-1 ;START TIME
10656 24144 LDA 1,CLOK ;REMAINING TIME
10657 122400 SUB 1,0 ;GET SEEK TIME
10660 34440 LDA 3,SKLM ;LIMIT
10661 116432 RML1A: SUBZ# 0,3,SZC ;SKIP IF >10 MS
10662 403 JMP RML1B ;OK
10663 6166 JSR @ERROR ;OVER LIMIT
10664 10721 RMLEB
```

; CHECK DRIVE DONE STATUS

```
10665 6201 RML1B: JSR @CKCK
10666 0 0
10667 10672 RML2 ;OK STATUS RETURN
10670 6166 JSR @ERROR ;NOT OK
10671 10727 RMLEB+ONE
```

; READ FORMAT FROM LAST SECTOR

```
10672 6170 RML2: JSR @RWCMD
10673 3600 RHDR
10674 50462 RBUFF
10675 144 144
```



---  
; DONE, CHECK STATUS AND HEADER

```
10676 6202 RML2A: JSR @CKCS
10677 40000 RWDN
10700 10703 RML2B
10701 6166 JSR @ERROR
10702 10735 RMLEB+TWO
10703 6172 RML2B: JSR @HDRG ; SET UP A HEADER
10704 1 1
10705 6173 JSR @HCMP ; COMPARE IT WITH READ
10706 1 1
10707 10712 RML2C ; NO ERROR VECTOR
10710 6166 JSR @ERROR ; DON'T MATCH
10711 10743 RMLEB+THREE
```

; DO NEXT CYLINDER IF NOT ALREADY THE LAST ONE

```
10712 20127 RML2C: LDA 0, CYLN
10713 24124 LDA 1, LCYN
10714 106405 SUB 0, 1, SNR ; AT LAST CYL?
10715 2114 JMP @NTST ; YES, END TEST
10716 10127 ISZ CYLN ; NO, DO NEXT ONE
10717 733 JMP RML1
```

```
10720 30 SKLM: 30
```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```
10721 10661 RMLEB: RML1A
10722 10652 RML1
10723 23242 MSKT
10724 10 10
10725 123260 @GT10
10726 123264 @SKMX

10727 10665 RML1B ; 1
10730 10652 RML1
10731 23126 DSCK
10732 15 15
10733 2 2
10734 16 16

10735 10676 RML2A ; 2
10736 10672 RML2
10737 23140 RHD
10740 6 6
10741 1 1
10742 15 15

10743 10705 RML2B+2; 3
10744 10672 RML2
10745 22730 HWC
10746 16 16
10747 14 14
10750 122742 @ASWB
```

;\*\*\*\* DATA VERIFY ERROR STATUS \*\*\*\*

; TEST THE DATA VERIFY FUNCTION OF THE CONTROLLER  
; AFTER WRITING A DATA PATTERN FROM WBUFF TO SURFACE 0, SECTOR 0, CHANGE  
; 1 BIT IN A DATA WORD IN MIDDLE OF WBUFF THEN DO DATA VERIFY. RESULT IS  
; DATA VERIFY ERROR STATUS

10751	6177	VERFY:	JSR	@SET
10752	25044		VERFYT	
10753	10756		VERF	
10754	11202		SECK	
10755	0		0	

; PRIME WBUFF WITH A DATA PATTERN AND WRITE IT TO SURF 0, SECT 0

10756	6176	VERF:	JSR	@PRIME
10757	30462		WBUFF	
10760	125252		125252	
10761	52525		52525	
10762	400		400	
10763	6205	JSR	@DADD	
10764	0		0	
10765	0		0	
10766	1		1	
10767	6170	VERF1:	JSR	@RWCMD
10770	3400		WRITE	
10771	30462		WBUFF	
10772	24		24	

; DONE, CHECK STATUS AND DISK ADDRESS WORD

10773	6202	VERF1A:	JSR	@CKCS
10774	40000		RWDN	
10775	11000		VERF1B	; GOOD STATUS RETURN
10776	6166	JSR	@ERROR	; BAD STATUS
10777	11114		VERFEB	
11000	6204	VERF1B:	JSR	@CKDA
11001	40		40	
11002	11005		VERF2	; ADDR OK VECTOR
11003	6166	JSR	@ERROR	; ADDR DISAGREE
11004	11122		VERFEB+ONE	

; READ HEADER INTO RBUFF

11005	6170	VERF2:	JSR	@RWCMD
11006	3600		RHDR	
11007	50462		RBUFF	
11010	24		24	

; DONE, CHECK STATUS AND ADDRESS POINTER

11011	6202	VERF2A:	JSR	@CKCS
11012	40000		RWDN	
11013	11016		VERF2B	; OK STATUS VECTOR
11014	6166	JSR	@ERROR	; STATUS NOT OK
11015	11130		VERFEB+TWO	
11016	6204	VERF2B:	JSR	@CKDA
11017	40		40	
11020	11023		VERF2C	; OK ADDR VECTOR

```

11021 6166 JSR @ERROR ; ADDR NOT OK
11022 11136 VRFEB+THREE

```

; CHECK DATA ECC VALUE

```

11023 34116 VRF2C: LDA 3, RBUF ; RBUFF ADDR
11024 24052 LDA 1, K4
11025 137000 ADD 1, 3 ; INDEX
11026 21400 LDA 0, 0, 3 ; 1ST ECC WORD
11027 24463 LDA 1, ECCH ; REFERENCE
11030 106414 SUB# 0, 1, SZR ; CORRECT?
11031 406 JMP VRF2D ; NO
11032 175400 INC 3, 3
11033 21400 LDA 0, 0, 3 ; 2ND ECC WORD
11034 24457 LDA 1, ECCL
11035 106415 SUB# 0, 1, SNR ; MATCH?
11036 405 JMP VRF3 ; PERFECTLY
11037 54154 VRF2D: STA 3, RBFP ; SET ERROR ADDR
11040 44141 STA 1, ACN ; SET REF
11041 6166 JSR @ERROR
11042 11144 VRFEB+FOUR

```

; PRIME RBUFF WITH ALL ONES THEN READ DATA FROM THAT SECTOR

```

11043 6176 VRF3: JSR @PRIME
11044 50462 RBUFF
11045 0 0
11046 0 0
11047 400 400
11050 6170 JSR @RWCMD
11051 0 READ
11052 50462 RBUFF
11053 24 24

```

; DONE, CHECK STATUS AND DISK ADDRESS

```

11054 6202 VRF3A: JSR @CKCS
11055 40000 RWDN
11056 11061 VRF3B ; STATUS OK RETURN
11057 6166 JSR @ERROR ; NOT OK
11060 11152 VRFEB+FIVE
11061 6204 VRF3B: JSR @CKDA
11062 40 40
11063 11066 VRF3C ; ADDR OK RETURN
11064 6166 JSR @ERROR ; ADDR NOT OK
11065 11160 VRFEB+SIX

```

; MAKE SURE THERE IS NO ERROR IN THE DATA THAT WAS WRITTEN TO THE DISK

```

11066 6174 VRF3C: JSR @DCMP ; CALL DATA COMPARE
11067 400 400
11070 11073 VRF4 ; COMPARE GOOD RETURN
11071 6166 JSR @ERROR ; COMPARE ERROR
11072 11166 VRFEB+SEVEN

```

; NOW CHANGE 1 DATA BIT IN 128TH WORD IN WBUFF, FROM 52525 TO 52524

```

11073 30115 VRF4: LDA 2, WBUF ; WBUFF ADDR
11074 21177 LDA 0, 177, 2 ; GET WORD

```

11075 101220 MOVZR 0,0 ; DROP LSB  
11076 101120 MOVZL 0,0  
11077 41177 STA 0,177,2 ; RETURN IT

; ISSUE A DATA VERIFY COMMAND, DISK VS WBUFF

11100 6170 VRF4A: JSR @RWCMD  
11101 3000 DATV  
11102 30462 WBUFF  
11103 24 24

; DONE, CHECK STATUS FOR VERIFY ERROR

11104 6202 VRF4B: JSR @CKCS  
11105 40011 RWDN+VRFE+RWFT  
11106 11111 VRF4C ; STATUS CORRECCT RETURN  
11107 6166 JSR @ERROR ; BUM STATUS  
11110 11174 VRFE+EIGHT  
11111 2114 VRF4C: JMP @NTST ; EXIT

11112 34464 ECCH: 34464  
11113 121535 ECCL: 121535

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

11114 10773 VRFE: VRF1A  
11115 10767 VRF1  
11116 23206 WDT  
11117 6 6  
11120 1 1  
11121 15 15

11122 11000 VRF1B ; 1  
11123 10767 VRF1  
11124 23206 WDT  
11125 6 6  
11126 3 3  
11127 17 17

11130 11011 VRF2A ; 2  
11131 11005 VRF2  
11132 23140 RHD  
11133 6 6  
11134 1 1  
11135 15 15

11136 11016 VRF2B ; 3  
11137 11005 VRF2  
11140 23140 RHD  
11141 6 6  
11142 3 3  
11143 17 17

11144 11023 VRF2C ; 4  
11145 11005 VRF2  
11146 23140 RHD  
11147 123221 @CECC  
11150 14 14  
11151 21 21

11152	11054	VRF3A	; 5
11153	11043	VRF3	
11154	23214	RDT	
11155	6	6	
11156	1	1	
11157	15	15	
11160	11061	VRF3B	; 6
11161	11043	VRF3	
11162	23214	RDT	
11163	6	6	
11164	3	3	
11165	17	17	
11166	11066	VRF3C	; 7
11167	11043	VRF3	
11170	22750	DWC	
11171	16	16	
11172	14	14	
11173	122742	@ASWB	
11174	11104	VRF4B	; 8
11175	11100	VRF4A	
11176	22701	RWDS	
11177	123226	@VBD	
11200	1	1	
11201	15	15	

;\*\*\*\* SECTOR CHECK \*\*\*\*

; USE THE SURF-SECT-SECT CNT WORD OF THE HEADER AS DATA AND WRITE IT OUT  
; TO EACH SECTOR OF SURFACE 0

```
11202 6177 SECK: JSR @SET
11203 25061 SECKT
11204 11207 SCK
11205 11354 WACY
11206 0 0
```

; INITIALIZE DISK ADDRESS AND SET UP WBUFF WITH THE UNIQUE DATA WORD

```
11207 6205 SCK: JSR @DADD
11210 0 0
11211 0 0
11212 1 1
11213 20061 LDA 0, K37
11214 40406 STA 0, SCK1 ; STARTING AT SECT 0, SECT CNT -1
11215 40406 STA 0, SCK1+1
11216 24107 LDA 1, ASECT
11217 44420 STA 1, SCK1C+1; SET DONE DISK ADDR
11220 6176 JSR @PRIME ; PRIME WBUFF
11221 30462 WBUFF
11222 0 SCK1: 0
11223 0 0
11224 400 400
```

; ISSUE A WRITE DATA COMMAND

```
11225 6170 SCK1A: JSR @RWCMD
11226 3400 WRITE
11227 30462 WBUFF
11230 24 24
```

; DONE, CHECK STATUS AND DISK ADDRESS WORD

```
11231 6202 SCK1B: JSR @CKCS
11232 40000 RWDN
11233 11236 SCK1C
11234 6166 JSR @ERRDR
11235 11324 SCKEB
11236 6204 SCK1C: JSR @CKDA
11237 0 0
11240 11243 SCK1D
11241 6166 JSR @ERROR
11242 11332 SCKEB+ONE
```

; INCREMENT ADDRESSES TO TO NEXT SECTOR IF NOT ALREADY THE LAST

```
11243 20130 SCK1D: LDA 0, SCTN
11244 24061 LDA 1, K37
11245 106415 SUB# 0, 1, SNR ; LAST SECT?
11246 415 JMP SCK2 ; YES, EXIT
11247 10130 ISZ SCTN ; NO, BUMP IT
11250 34767 LDA 3, SCK1C+1
11251 24107 LDA 1, ASECT
11252 137000 ADD 1, 3
11253 20102 LDA 0, SMSK
```

```

11254 117400      AND    0,3      ; MASK OUT SURF ADDR
11255  54762      STA    3,SCK1C+1; SEET NEXT SECT ADDR REF
11256  20744      LDA    0,SCK1
11257 123000      ADD    1,0
11260  40742      STA    0,SCK1  ; CHANGE SECT DATA
11261  40742      STA    0,SCK1+1
11262    736      JMP    SCK1-2  ; GO REPRIME WBUFF

```

```

; RESET SECTOR ADDRESS AND START READING DATA INTO RBUFF ONE SECTOR AT
; A TIME

```

```

11263 102400 SCK2: SUB    0,0
11264  40130      STA    0,SCTN  ; SECT 0
11265  24061      LDA    1,K37
11266  44435      STA    1,REF  ; SET STARTINF DATA REFERENCE
11267   6170 SCK2A: JSR    @RWCMD
11270    0        READ
11271  50462      RBUFF
11272    24        24

```

```

; DONE, CHECK STATUS

```

```

11273   6202 SCK2B: JSR    @CKCS
11274  40000      RWDN
11275  11300      SCK2C
11276   6166      JSR    @ERROR  ; BAD STATUS
11277  11340      SCKEB+TWO

```

```

; CHECK ONLY THE FIRST WORD OF RBUFF

```

```

11300  22116 SCK2C: LDA    0,@RBUF  ; GET READ WORD
11301  24422      LDA    1,REF  ; REFERENCE WORD
11302 106415      SUB#   0,1,SNR  ; AGREE?
11303    406      JMP    SCK2D  ; YES
11304  44141      STA    1,ACN  ; SAVE REF
11305  24116      LDA    1,RBUF
11306  44154      STA    1,RBFP  ; NO, SET ADDR
11307   6166      JSR    @ERROR
11310  11346      SCKEB+THREE

```

```

; GO READ NEXT SECTOR IF IT IS NOT ALREADY THE LAST SECTOR

```

```

11311  20130 SCK2D: LDA    0,SCTN  ; CURRENT SECT
11312  24061      LDA    1,K37
11313 106415      SUB#   0,1,SNR  ; THE LAST ONE?
11314   2114      JMP    @NTST  ; YES, EXIT
11315  10130      ISZ    SCTN  ; NO, BUMP IT ONCE
11316  24405      LDA    1,REF
11317  34107      LDA    3,ASECT
11320 167000      ADD    3,1
11321  44402      STA    1,REF  ; UPDATE REF
11322    745      JMP    SCK2A

```

```

11323    0 REF:  0

```

```

; **** ERROR MESSAGE BLOCKS ****

```

```

11324  11231 SCKEB: SCK1B
11325  11225      SCK1A

```

11326	23206	WDT
11327	6	6
11330	1	1
11331	15	15
11332	11236	SCK1C ; 1
11333	11225	SCK1A
11334	22714	RWDA
11335	6	6
11336	3	3
11337	17	17
11340	11273	SCK2B ; 2
11341	11267	SCK2A
11342	23214	RDT
11343	6	6
11344	1	1
11345	15	15
11346	11302	SCK2C+2; 3
11347	11267	SCK2A
11350	23214	RDT
11351	122750	@DWC
11352	14	14
11353	21	21



;\*\*\*\* WRITE HEADER \*\*\*\*

;WRITE HEADER TO A SECTOR WHICH CONTAINS DATA. VERIFY THAT WRITING THE  
;HEADER DOES NOT EFFECT THE DATA

11354	6177	WACY:	JSR	@SET
11355	25070		WACYT	
11356	11361		WAC	
11357	11510		OVLPS	
11360	0		0	

;SET DISK ADDRESS TO SURFACE 0, SECTOR 14 THEN WRITE HEADER TO IT

11361	6205	WAC:	JSR	@DADD	;PRESET ADDR
11362	0		0		
11363	16		16		
11364	1		1		
11365	34115	LDA	3,WBUF		;WBUF ADDR
11366	102400	SUB	0,0		
11367	41400	STA	0,0,3		;WORD 0 = 0
11370	24457	LDA	1,HDW		
11371	45401	STA	1,1,3		;WORD 1 = 700
11372	24456	LDA	1,HDW+1		
11373	45402	STA	1,2,3		;WORD 2 = 777
11374	6170	JSR	@RWCMD		;WRITE OUT HEADER
11375	600	WHDR			
11376	30462	WBUF			
11377	74		74		

;DONE, CHECK CONTROL STATUS

11400	6202	WAC1:	JSR	@CKCS
11401	40000		RWDN	
11402	11405		WAC1A	
11403	6166		JSR	@ERROR ;STATUS NG
11404	11452		WACEB	

;READ HEADER BACK INTO RBUF

11405	6170	WAC1A:	JSR	@RWCMD
11406	3600		RHDR	
11407	50462		RBUF	
11410	24		24	

;DONE, CHECK STATUS AND VERIFY HEADER

11411	6202	WAC1B:	JSR	@CKCS
11412	40000		RWDN	
11413	11416		WAC1C	;GOOD STATUS VECTOR
11414	6166		JSR	@ERROR ;BAD STATUS
11415	11460		WACEB+ONE	
11416	6173	WAC1C:	JSR	@HCMP ;COMPARE HEADERS
11417	1		1	
11420	11423		WAC2	;GOOD HEADER
11421	6166		JSR	@ERROR ;BAD ONE
11422	11466		WACEB+TWO	

;NOW READ DATA FROM SECTOR

11423 6170 WAC2: JSR @RWCMD  
11424 0 READ  
11425 50462 RBUFF  
11426 24 24

; DONE, CHECK FIRST WORD OF DATA IN RBUFF

11427 22116 WAC2A: LDA 0, @RBUF ; GET READ DATA  
11430 24421 LDA 1, PDAT ; REFERENCE  
11431 106405 SUB 0, 1, SNR ; DATA OK?  
11432 403 JMP WAC2B ; OK  
11433 6166 JSR @ERROR ; NOT SO  
11434 11474 WACEB+THREE

; RESTORE HEADER TO NORMAL BEFORE LEAVING

11435 6170 WAC2B: JSR @RWCMD  
11436 1400 FMAT  
11437 30462 WBUFF  
11440 24 24

; DONE, CHECK STATUS

11441 6202 WAC2C: JSR @CKCS  
11442 40000 RWDN  
11443 11446 WAC2D  
11444 6166 JSR @ERROR ; BAD STATUS  
11445 11502 WACEB+FOUR  
11446 2114 WAC2D: JMP @NTST

11447 700 HDW: 700  
11450 777 777

11451 0 PDAT: 0

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

11452 11400 WACEB: WAC1  
11453 11361 WAC  
11454 23117 WHD  
11455 6 6  
11456 1 1  
11457 15 15

11460 11411 WAC1B ; 1  
11461 11405 WAC1A  
11462 23140 RHD  
11463 6 6  
11464 1 1  
11465 15 15

11466 11416 WAC1C ; 2  
11467 11405 WAC1A  
11470 22730 HWC  
11471 16 16  
11472 14 14  
11473 122742 @ASWB

11474 11427 WAC2A ; 3

11475 11423 WAC2  
11476 22750 DWC  
11477 0 0  
11500 14 14  
11501 0 0

11502 11441 WAC2C ; 4  
11503 11435 WAC2B  
11504 23271 FMTG  
11505 6 6  
11506 1 1  
11507 15 15

;\*\*\*\* OVERLAP SEEK \*\*\*\*

;TEST THE ABILITY OF THE CONTROLLER TO ACCEPT AN OVERLAP SEEK COMMAND  
;WHILE IT IS BUSY ON AN IO OPERATION

11510	6177	OVLPS:	JSR	@SET
11511	25077		OVLPT	
11512	11515		OVL	
11513	11675		BSST	
11514	0		0	

;DO READ DATA TO WBUFF TO SYNC WITH DISK. WBUFF DATA WILL BE USED  
;FOR COMPARE REFERENCE

11515	6205	OVL:	JSR	@DADD
11516	0		0	
11517	0		0	
11520	1		1	
11521	6170		JSR	@RWCMD
11522	0		READ	
11523	30462		WBUFF	
11524	24		24	

;DONEE, CHECK CONTROL STATUS

11525	6202	OVL1:	JSR	@CKCS
11526	40000		RWDN	
11527	11532		OVL1A	; STATUS OK
11530	6166		JSR	@ERROR ; NOT OK
11531	11637		OVL1B	

;PRIME RBUFF WITH ALL ONES DATA

11532	6176	OVL1A:	JSR	@PRIME
11533	50462		RBUFF	
11534	177777		-1	
11535	177777		-1	
11536	400		400	

;ISSUE A READ COMMAND AND IMMEDIATELY FOLLOWED BY A SEEK TO CYL 1 COMMAND

11537	102400	OVL1B:	SUB	0,0	
11540	24121		LDA	1,DRVF	; DRIVE SELECT
11541	123000		ADD	1,0	
11542	4211		JSR	DDOA	; SELECT & READ
11543	20061		LDA	0,K37	
11544	4225		JSR	DDOC	; 1 SECT FROM SURF 0, SECT 0
11545	20116		LDA	0,RBUF	
11546	4223		JSR	DDOBS	; TO RBUFF, START
11547	20462		LDA	0,SEKK	
11550	123000		ADD	1,0	
11551	4211		JSR	DDOA	; SEEK ON SAME DRIVE
11552	102520		SUBZL	0,0	
11553	4232		JSR	DDOCP	; TO CYL 1

;CHECK CONTROL FULL STATUS TO STAY SET FOR ABOUT 400 US

11554	24456	OVL1C:	LDA	1,CFTM	; TIMER
11555	4234		JSR	DDIA	; READ STATUS

```

11556 101113      MOVL#  0,0,SNC  ; CF ON?
11557   405      JMP    OVL2    ; NO, EXIT
11560 125404      INC    1,1,SZR  ; TIME OUT?
11561   774      JMP    OVL1C+1 ; NO, LOOP
11562   6166     JSR    @ERROR ; YES
11563  11645     OVLEB+ONE

```

; ADD 1 TO CURRENT DRIVE NUMBER AND ISSUE SEEK TO CYL 1 ON THAT DRIVE

```

11564 20121 OVL2: LDA    0,DRVF ; CURRENT DRIVE NUMBER
11565 24446      LDA    1,ADRN
11566 123000     ADD    1,0      ; SET NEXT DRIVE
11567 34445     LDA    3,DRM   ; MASK
11570 163400     AND    3,0      ; PICK OFF DRV FIELD
11571 24440     LDA    1,SEKK
11572 123000     ADD    1,0      ; ADD SEEK COMMAND
11573  4211     JSR    DDOA   ; LOAD IT
11574 102520     SUBZL  0,0
11575  4232     JSR    DDOCP  ; SEEK TO CYL 1

```

; CHECK CONTROL FULL STATUS TO STAY ON FOR 400 US

```

11576 24434 OVL2A: LDA    1,CFTM
11577  4234     JSR    DDIA   ; GET STATUS
11600 101113     MOVL#  0,0,SNC  ; CF SET?
11601   405     JMP    OVL2B   ; NO
11602 125404     INC    1,1,SZR  ; TIME OUT?
11603   774     JMP    OVL2A+1 ; NO
11604   6166     JSR    @ERROR ; YES
11605  11653     OVLEB+TWO

```

; WAIT FOR DONE THEN CHECK DRIVE ATTENTION FLAG FOR THE PROPER DRIVE

```

11606  4261 OVL2B: JSR    SKDN
11607   777     JMP    -1
11610  4234     JSR    DDIA   ; DONE, GEET STATUS
11611 24134     LDA    1,ATTS
11612 125220     MOVZR  1,1      ; 2ND DRIVE ATTEN BIT
11613 34423     LDA    3,BIT6
11614 137415     AND#   1,3,SNR  ; OVERFLOW?
11615   402     JMP    +2      ; NO
11616 24417     LDA    1,BIT2  ; YES, SET FOR DRV 0
11617 123414     AND#   1,0,SZR  ; 2ND DRV ATTEN?
11620  403     JMP    OVL2C   ; YES
11621  6166     JSR    @ERROR ; NO
11622  11661     OVLEB+THREE

```

; VERIFY THAT THE READ DATA OPERATION IN THE FIRST DRIVE WAS SUCCESSFUL

```

11623  6174 OVL2C: JSR    @DCMP  ; CALL DATA COMPARE
11624   400     400
11625 11630     OVL2D  ; GOOD DATA VECTOR
11626  6166     JSR    @ERROR  ; NOT WORKING!
11627  11667     OVLEB+FOUR
11630  2114 OVL2D: JMP    @NTST

```

```

11631   400 SEKK: SEEK
11632 177500 CFTM: -300
11633   40  ADRN: 40

```

11634 140 DRM: 140  
11635 20000 BIT2: 20000  
11636 1000 BIT6: 1000

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*;

11637	11525	OVLEB: OVL1
11640	11515	OVL
11641	23214	RDT
11642	6	6
11643	1	1
11644	15	15
11645	11554	OVL1C ; 1
11646	11537	OVL1B
11647	23301	SAR
11650	10	10
11651	123317	@CF1
11652	123325	@CFN
11653	11576	OVL2A ; 2
11654	11564	OVL2
11655	23301	SAR
11656	10	10
11657	123317	@CF1
11660	123325	@CFN
11661	11610	OVL2B+2; 3
11662	11564	OVL2
11663	21710	DRAT
11664	10	10
11665	1	1
11666	123341	@SSDA
11667	11625	OVL2C+2; 4
11670	11515	OVL
11671	23214	RDT
11672	125077	@OVLPT
11673	123353	@USRD
11674	123415	@RDOV

;\*\*\*\* BAD SECTOR STATUS \*\*\*\*

;TEST THE ABILITY OF THE CONTROLLER TO ABORT AN WR OPERATION WHEN IT  
;DETECTS THE BAD SECTOR FLAG IS SET IN THE HEADER

```
11675 6177 BSST: JSR @SET
11676 25106 BSST
11677 11702 BSS
11700 12060 CHAIN
11701 0 0
```

;PREPARE A HEADER OF SURF 0, SECT 0 WITH BAD SECTOR BIT ON IN WBUF

```
11702 30115 BSS: LDA 2,WBUF
11703 102620 SUBZR 0,0
11704 41000 STA 0,0,2 ;WORD 0, BAD SECT BIT SET
11705 126400 SUB 1,1
11706 45001 STA 1,1,2 ;WORDS 1 & 2 = 0
11707 45002 STA 1,2,2
```

;SEEK TO CYLINDER 0 THEN WRITE HEADER

```
11710 44127 BSS1: STA 1,CYLN ;SET CYL 0
11711 6205 JSR @DADD ;SET DISK ADDR
11712 0 0
11713 0 0
11714 1 1
11715 6167 JSR @DRCMD
11716 400 SEEK
11717 144 144
11720 6170 JSR @RWCMD ;WRITE HEADER
11721 600 WHDR
11722 30462 WBUF
11723 74 74
```

;DONE, CHECK STATUS

```
11724 6202 BSS1A: JSR @CKCS
11725 40000 RWDN
11726 11731 BSS1B ;STATUS GOOD
11727 6166 JSR @ERROR ;STATUS NG
11730 12014 BSSEB
```

;READ HEADER BACK INTO RBUF FOR VERIFICATION

```
11731 6170 BSS1B: JSR @RWCMD
11732 3600 RHDR
11733 50462 RBUF
11734 50 50
```

;DONE, CHECK STATUS

```
11735 6202 BSS1C: JSR @CKCS
11736 40000 RWDN
11737 11742 BSS1D ;OK STATUS
11740 6166 JSR @ERROR ;NOT OK
11741 12022 BSSEB+ONE
```

;VERIFY THAT THE HEADER WAS CORRECTLY WRITTEN

```

11742 6173 BSS1D: JSR @HCMP ; COMPARE HEADERS
11743 1 1
11744 11747 BSS2 ; NO ERROR
11745 6166 JSR @ERROR
11746 12030 BSSEB+TWO

```

; PRIME RBUFF WITH ALL ONES THEN ATTEMPT TO READ DATA FROM BAD SECTOR

```

11747 6176 BSS2: JSR @PRIME
11750 50462 RBUFF
11751 177777 -1
11752 177777 -1
11753 400 400
11754 6170 JSR @RWCMD ; READ DATA
11755 0 READ
11756 50462 RBUFF
11757 50 50

```

; DONE, CHECK FOR RW DONE + BAD SECTOR + RW FAULT STATUS

```

11760 6202 BSS2A: JSR @CKCS
11761 40101 RWDN+BSCT+RWFT
11762 11765 BSS2B ; CORRECT STATUS
11763 6166 JSR @ERROR ; INCORRECT STATUS
11764 12036 BSSEB+THREE

```

; VERIFY THAT THE READ DATA ATTEMPT WAS ABORTED BY CHECKING NO CHANGE IN  
; DATA IN RBUFF

```

11765 34116 BSS2B: LDA 3, RBUF ; RBUFF ADDR
11766 54154 STA 3, RBFP ; SET POINTER
11767 34064 LDA 3, N400 ; WORD COUNT
11770 126000 ADC 1, 1 ; REFERENCE
11771 22154 BSS2C: LDA 0, @RBFP ; FETCH A WORD
11772 106414 SUB# 0, 1, SZR ; SAME?
11773 405 JMP BSS2D ; NO!
11774 10154 ISZ RBFP
11775 175404 INC 3, 3, SZR ; YES, MORE?
11776 773 JMP BSS2C ; LOOP BACK
11777 403 JMP BSS2E ; NO, EXIT

```

```

12000 6166 BSS2D: JSR @ERROR
12001 12044 BSSEB+FOUR

```

; RESTORE HEADER TO NORMAL BEFORE LEAVING TEST

```

12002 6170 BSS2E: JSR @RWCMD
12003 1400 FMAT
12004 30462 WBUFF
12005 50 50

```

; DONE, CHECK STATUS THEN LEAVE

```

12006 6202 BSS2F: JSR @CKCS
12007 40000 RWDN
12010 12013 BSS2G ; ALL RIGHT!
12011 6166 JSR @ERROR ; SCREWED UP
12012 12052 BSSEB+FIVE

```



12013 2114 BSS2G:JMP @NTST

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

12014 11724 BSSEB: BSS1A  
12015 11710 BSS1  
12016 23117 WHD  
12017 6 6  
12020 1 1  
12021 15 15

12022 11735 BSS1C ; 1  
12023 11731 BSS1B  
12024 23140 RHD  
12025 6 6  
12026 1 1  
12027 15 15

12030 11742 BSS1D ; 2  
12031 11731 BSS1B  
12032 22730 HWC  
12033 16 16  
12034 14 14  
12035 122742 @ASWB

12036 11760 BSS2A ; 3  
12037 11747 BSS2  
12040 22701 RWDS  
12041 123431 @RDBS  
12042 1 1  
12043 15 15

12044 11771 BSS2C ; 4  
12045 11747 BSS2  
12046 23431 RDBS  
12047 123447 @CKRB  
12050 123214 @RDT  
12051 123462 @ARDT

12052 12006 BSS2F ; 5  
12053 12002 BSS2E  
12054 23271 FMTG  
12055 6 6  
12056 1 1  
12057 15 15

;\*\*\*\* SECTOR CHAINING \*\*\*\*

;TEST THE ABILITY OF THE CONTROLLER TO READ DATA FROM A CHAINED ALTERNATE  
;SECTOR WHEN THE CHAIN BIT IS SET IN THE HEADER OF THE ORIGINAL SECTOR

12060	6177	CHAIN: JSR	@SET
12061	25117	CHNT	
12062	12065	CHNS	
12063	12331	DCHN	
12064	0	0	

;SEEK TO CYLINDER 1 TO FORMAT SURFACE 0 FOR ALTERNATE TRACK SECTOR  
;CHAINING IN THIS TEST

12065	102520	CHNS: SUBZL	0,0
12066	40127	STA	0,CYLN ; SET CYL 1
12067	6205	JSR @DADD	; SET DISK ADDR
12070	0	0	
12071	0	0	
12072	1	1	
12073	6167	JSR @DRCMD	
12074	400	SEEK	
12075	144	144	
12076	6170	JSR @RWCMD	; FORMAT SECT 0
12077	1400	FMAT	
12100	30462	WBUFF	
12101	74	74	

; DONE, CHECK STATUS

12102	6202	CHN1: JSR	@CKCS
12103	40000	RWDN	
12104	12107	CHN1A	; GOOD STATUS
12105	6166	JSR @ERROR	; BAD STATUS
12106	12243	CHNEB	

;PRIME WBUFF WITH A DATA PATTERN AND WRITE IT OUT TO THE CHAINED SECTOR  
;AS REFERENCE

12107	6176	CHN1A: JSR	@PRIME
12110	30462	WBUFF	
12111	52525	52525	
12112	125252	125252	
12113	400	400	
12114	6170	JSR @RWCMD	; WRITE DATA
12115	3400	WRITE	
12116	30462	WBUFF	
12117	50	50	

; DONE, CHECK STATUS

12120	6202	CHN1B: JSR	@CKCS
12121	40000	RWDN	
12122	12125	CHN1C	; LOOKS GOOD
12123	6166	JSR @ERROR	; DOES NOT
12124	12251	CHNEB+ONE	

;READ DATA BACK INTO RBUFF

12125 6170 CHN1C: JSR @RWCMD  
12126 0 READ  
12127 50462 RBUFF  
12130 24 24

; DONE, CHECK STATUS AND RBUFF

12131 6202 CHN1D: JSR @CKCS  
12132 40000 RWDN  
12133 12136 CHN1E ; OK  
12134 6166 JSR @ERROR  
12135 12257 CHNEB+TWO  
12136 6174 CHN1E: JSR @DCMP ; COMPARE DATA  
12137 400 400  
12140 12143 CHN2 ; GOOD DATA  
12141 6166 JSR @ERROR ; BAD DATA  
12142 12265 CHNEB+THREE

; SEEK TO CYLINDER 0 TO WRITE THE CHAIN HEADER TO SURF 0, SECT 0

12143 102400 CHN2: SUB 0,0  
12144 40127 STA 0,CYLN ; SET CYL 0  
12145 6167 JSR @DRCMD  
12146 400 SEEK  
12147 24 24  
12150 6170 JSR @RWCMD ; WRITE HEADER  
12151 600 WHDR  
12152 12240 CHDR  
12153 50 50

; DONE, CHECK STATUS

12154 6202 CHN2A: JSR @CKCS  
12155 40000 RWDN  
12156 12161 CHN2B ; GOOD STATUS  
12157 6166 JSR @ERROR ; NOT SO GOOD  
12160 12273 CHNEB+FOUR

; READ HEADER BACK INTO RBUFF

12161 6170 CHN2B: JSR @RWCMD  
12162 3600 RHDR  
12163 50462 RBUFF  
12164 24 24

; DONE, CHECK STATUS

12165 6202 CHN2C: JSR @CKCS  
12166 40000 RWDN  
12167 12172 CHN2D ; OK  
12170 6166 JSR @ERROR ; NOT OK  
12171 12301 CHNEB+FIVE

; VERIFY THAT THE CHAIN HEADER WAS WRITTEN CORRECTLY

12172 30116 CHN2D: LDA 2,RBUF  
12173 50154 STA 2,RBFP ; SET READ POINTER  
12174 24415 LDA 1,CHDRA  
12175 44153 STA 1,WBFP ; SET WRITE POINTER

```

12176 34051 LDA 3,K3
12177 174400 NEG 3,3 ; WORD COUNT
12200 22154 CHN2E: LDA 0,@RBFP ; GET READ WORD
12201 26153 LDA 1,@WBFP ; GET WRITE WORD
12202 106414 SUB# 0,1,SZR ; MATCH?
12203 407 JMP CHN2F ; NO
12204 10154 ISZ RBFP
12205 10153 ISZ WBFP
12206 175404 INC 3,3,SZR ; DONE?
12207 771 JMP CHN2E ; NOT YET
12210 404 JMP CHN3 ; YES

```

```
12211 12240 CHDRA: CHDR
```

```
12212 6166 CHN2F: JSR @ERROR
12213 12307 CHNEB+SIX
```

```
; PRIME RBUFF WITH ALLONES DATA THEN READ DATA FROM SECT 0
```

```
12214 6176 CHN3: JSR @PRIME
12215 50462 RBUFF
12216 177777 -1
12217 177777 -1
12220 400 400
12221 6170 JSR @RWCMD
12222 0 READ
12223 50462 RBUFF
12224 74 74
```

```
; DONE, CHECK STATUS
```

```
12225 6201 CHN3A: JSR @CKCK
12226 40000 RWDN
12227 12232 CHN3B ; STATUS OK
12230 6166 JSR @ERROR ; NO GOOD
12231 12315 CHNEB+SEVEN
```

```
; NOW VERIFY THAT DATA IN RBUFF IS READ FROM THE ALTERNATE SECTOR
```

```
12232 6174 CHN3B: JSR @DCMP ; COMPARE DATA
12233 400 400
12234 12237 CHN3C ; TRUE
12235 6166 JSR @ERROR ; FALSE
12236 12323 CHNEB+EIGHT
12237 2114 CHN3C: JMP @NTST
```

```
12240 40000 CHDR: 40000
12241 0 0
12242 1 1
```

```
; **** ERROR MESSAGE BLOCKS ****
```

```
12243 12102 CHNEB: CHN1
12244 12076 CHN1-4
12245 23271 FMTG
12246 6 6
12247 1 1
12250 15 15
```

12251	12120	CHN1B ; 1
12252	12114	CHN1B-4
12253	23206	WDT
12254	6	6
12255	1	1
12256	15	15
12257	12131	CHN1D ; 2
12260	12125	CHN1C
12261	23214	RDT
12262	6	6
12263	1	1
12264	15	15
12265	12136	CHN1E ; 3
12266	12125	CHN1C
12267	22750	DWC
12270	16	16
12271	14	14
12272	122742	@ASWB
12273	12154	CHN2A ; 4
12274	12145	CHN2+2
12275	23117	WHD
12276	6	6
12277	1	1
12300	15	15
12301	12165	CHN2C ; 5
12302	12161	CHN2B
12303	23140	RHD
12304	6	6
12305	1	1
12306	15	15
12307	12200	CHN2E ; 6
12310	12161	CHN2B
12311	22730	HWC
12312	16	16
12313	14	14
12314	122742	@ASWB
12315	12225	CHN3A ; 7
12316	12214	CHN3
12317	23522	RDFC
12320	6	6
12321	1	1
12322	15	15
12323	12232	CHN3B ; 8
12324	12214	CHN3
12325	23522	RDFC
12326	122750	@DWC
12327	123541	@NFCS
12330	123560	@FCS

;\*\*\*\* DOUBLE CHAIN SECTOR \*\*\*\*

;VERIFY THAT WHEN ATTEMPTING TO READ DATA FROM AN ALTERNATE CHAINED SECTOR  
;WHICH IS CHAINED BACK TO THE ORIGINAL SECTOR WILL CAUSE AN RW TIMEOUT

;THIS TEST USE THE ALTERNATE TRACK AS SET UP IN THE PRECEDING TEST

12331	6177	DCHN:	JSR	@SET
12332	25127		DCHNT	
12333	12336		DCH	
12334	12433		CBSC	
12335	0		0	

;SEEK TO CYLINDER 1 TO WRITE THE CHAIN HEADER TO THE CHAINED SECTOR

12336	102520	DCH:	SUBZL	0,0	
12337	40127		STA	0,CYLN	;SET CYL 1
12340	6167		JSR	@DRCMD	;SEEK THERE
12341	400		SEEK		
12342	144		144		
12343	6170		JSR	@RWCMD	;WRITE HEADER
12344	600		WHDR		
12345	12406		CHDR1		
12346	74		74		

;DONE, CHECK FOR RW & DRV DONE STATUS

12347	6201	DCH1:	JSR	@CKCK	
12350	40000		RWDN		
12351	12354		DCH1A		;GOOD STATUS
12352	6166		JSR	@ERROR	;BAD STATUS
12353	12411		DCHB		

;PRIME RBUFF WITH ALL ONES DATA THEN SEEK TO CYL 0

12354	6176	DCH1A:	JSR	@PRIME	
12355	50462		RBUFF		
12356	177777		-1		
12357	177777		-1		
12360	400		400		
12361	102400		SUB	0,0	
12362	40127		STA	0,CYLN	;SET CYL 0
12363	6167		JSR	@DRCMD	;SEEK TO CYL 0
12364	400		SEEK		
12365	24		24		

;NOW TRY TO READ DATA FROM SURF 0, SECT 0

12366	6170	DCH1B:	JSR	@RWCMD	
12367	0		READ		
12370	50462		RBUFF		
12371	3720		3720		

;DONE, CHECK FOR RW & DRV DONE + RW TIMEOUT + RW FAULT STATUS

12372	6201	DCH1C:	JSR	@CKCK	
12373	40005		RWDN+RWTD+RWFT		
12374	12377		DCH1D		;TIMEOUT OK
12375	6166		JSR	@ERROR	;NOT THAT

12376 12417 DCHEB+ONE

;VERIFY THAT NO DATA HAS BEEN READ INTO RBUF

```
12377 22116 DCH1D: LDA    0,@RBUF ;GET 1ST WORD
12400 126000      ADC    1,1
12401 122415      SUB#   1,0,SNR ;DIFFERENT?
12402   403      JMP    DCH2B ;NO
12403  6166      JSR    @ERROR ;NOT SUPPOSED TO
12404 12425      DCHEB+TWO
12405  2114 DCH2B: JMP    @NTST

12406 40001 CHDR1: 40001
12407   0      0
12410   0      0
```

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```
12411 12347 DCHEB: DCH1
12412 12336      DCH
12413 23117      WHD
12414   6      6
12415   1      1
12416  15      15

12417 12372      DCH1C ; 1
12420 12366      DCH1B
12421 23214      RDT
12422   6      6
12423   1      1
12424  15      15

12425 12377      DCH1D ; 2
12426 12366      DCH1B
12427 23612      VARD
12430 123627     @RDDC
12431 123214     @RDT
12432 123647     @ABRT
```

;\*\*\*\* BAD ALTERNATE SECTOR \*\*\*\*

;VERIFY THAT WHEN ATTEMPTING TO READ DATA FROM A CHAINED BAD SECTOR  
;WILL CAUSE THE OPERATION TO BE ABORTED

;THE ALTERNATE SECTOR WAS SET UP IN THE PRECEDING TEST

12433	6177	CBS:	JSR	@SET
12434	25141		CBSCT	
12435	12440		CBS	
12436	12632		RFIFO	
12437	0		0	

;SEEK TO CYLINDER 1 TO WRITE A THE HEADER WITH BAD SECTOR BIT SET

12440	126520	CBS:	SUBZL	1,1	
12441	44127		STA	1,CYLN	;SET CYL 1
12442	6167		JSR	@DRCMD	
12443	400		SEEK		
12444	144		144		
12445	6170		JSR	@RWCMD	;WRITE HEADER
12446	600		WHDR		
12447	12563		CHHD3		
12450	74		74		

;DONE, CHECK FOR RW & DRV ATTEN STATUS

12451	6201	CBS1:	JSR	@CKCK	
12452	40000		RWDN		
12453	12456		CBS1A		;GOOD STATUS
12454	6166		JSR	@ERROR	;BAD STATUS
12455	12566		CBSEB		

;READ HEADER BACK INTO RBUFF AND VERIFY IT

12456	6170	CBS1A:	JSR	@RWCMD	
12457	3600		RHDR		
12460	50462		RBUFF		
12461	24		24		
12462	24501	CBS1B:	LDA	1,CHHD3	;GET WRITE WORD 0
12463	22116		LDA	0,@RBUF	;READ WORD 0
12464	106415		SUB#	0,1,SNR	;AGREE?
12465	404		JMP	CBS2	;YES
12466	44141		STA	1,ACN	;SAVE WRITE WORD
12467	6166		JSR	@ERROR	;NO
12470	12574		CBSEB+ONE		

;PRIME RBUFF WITH ALL ONES DATA THEN SEEK TO CYLINDER 0

12471	6176	CBS2:	JSR	@PRIME	
12472	50462		RBUFF		
12473	177777		-1		
12474	177777		-1		
12475	400		400		
12476	102400		SUB	0,0	
12477	40127		STA	0,CYLN	;SET CYL 0
12500	6167		JSR	@DRCMD	
12501	400		SEEK		
12502	24		24		



---  
; NOW TRY TO READ DATA FROM CHAINED BAD SECTOR

```
12503 6170 CBS2A: JSR @RWCMD  
12504 0 READ  
12505 50462 RBUFF  
12506 74 74
```

; DONE, CHECK FOR RW & DRV DONE + BAD SECT + RW FAULT STATUS

```
12507 6201 CBS2B: JSR @CKCK  
12510 40101 RWDN+BSCT+RWFT  
12511 12514 CBS2C ; CORRECT STATUS  
12512 6166 JSR @ERROR ; INCORRECT  
12513 12602 CBSEB+TWO
```

; SEE IF READ OPERATION WAS ABORTED

```
12514 26116 CBS2C: LDA 1, @RBUF ; GET 1ST WORD  
12515 176000 ADC 3, 3  
12516 166415 SUB# 3, 1, SNR ; SAME DATA?  
12517 403 JMP CBS3 ; YES  
12520 6166 JSR @ERROR ; NO  
12521 12610 CBSEB+THREE
```

; CHECK POSITION OF RW HEADS

```
12522 6170 CBS3: JSR @RWCMD  
12523 3600 RHDR  
12524 50462 RBUFF  
12525 24 24
```

; DONE, CHECK CYLINDER WORD OF HEADER IN RBUFF

```
12526 32116 CBS3A: LDA 2, @RBUF  
12527 151213 MOVR# 2, 2, SNC ; CYL 0?  
12530 403 JMP CBS3B ; YES  
12531 6166 JSR @ERROR ; NO  
12532 12616 CBSEB+FOUR
```

; RESTORE HEADERS OF CYLINDERS 0 AND 1 BACK TO NORMAL BEFORE LEAVING

```
12533 126400 CBS3B: SUB 1, 1  
12534 44127 STA 1, CYLN ; SET CYL 0  
12535 6205 JSR @DADD ; SET DISK ADDR  
12536 0 0  
12537 0 0  
12540 100123 @STRK  
12541 6170 JSR @RWCMD ; FORMAT CYL 0  
12542 1400 FMAT  
12543 30462 WBUFF  
12544 50 50  
12545 10127 ISZ CYLN ; SET CYL 1  
12546 6167 JSR @DRCMD  
12547 400 SEEK  
12550 24 24  
12551 6170 JSR @RWCMD  
12552 1400 FMAT  
12553 30462 WBUFF
```

12554 74 74

; DONE, CHECK STATUS

12555	6202	CBS3C: JSR	@CKCS
12556	40000	RWDN	
12557	12562	CBS3D	; OK
12560	6166	JSR	@ERROR
12561	12624	CBSEB+FIVE	
12562	2114	CBS3D: JMP	@NTST
12563	100001	CHHD3: 100001	
12564	0	0	
12565	0	0	

; \*\*\*\* ERROR MESSAGE BLOCK \*\*\*\*

12566	12451	CBSEB: CBS1	
12567	12440	CBS	
12570	23117	WHD	
12571	6	6	
12572	1	1	
12573	15	15	
12574	12462	CBS1B	; 1
12575	12456	CBS1A	
12576	23140	RHD	
12577	6	6	
12600	14	14	
12601	21	21	
12602	12507	CBS2B	; 2
12603	12503	CBS2A	
12604	23214	RDT	
12605	123506	@FBCS	
12606	1	1	
12607	15	15	
12610	12514	CBS2C	; 3
12611	12507	CBS2B	
12612	23214	RDT	
12613	123506	@FBCS	
12614	123575	@FTA	
12615	123647	@ABRT	
12616	12526	CBS3A	; 4
12617	12522	CBS3	
12620	23657	HPOS	
12621	123140	@RHD	
12622	123673	@NCYO	
12623	123703	@ACYO	
12624	12555	CBS3C	; 5
12625	12533	CBS3B	
12626	23271	FMTG	
12627	6	6	
12630	1	1	
12631	15	15	

;\*\*\*\* READ FIFO BUFFER \*\*\*\*

;TEST THE ABILITY OF THE CONTROLLER TO READ CONTENTS OF ITS FIFO BUFFER

```
12632 6177 RFIFO: JSR @SET
12633 25152 RFFI
12634 12637 RFF
12635 12727 ENDD
12636 0 0
```

;PRIME WBUFF WITH A DATA PATTERN AND WRITE IT OUT TO SURF 0, SECT 0

```
12637 6176 RFF: JSR @PRIME
12640 30462 WBUFF
12641 125252 125252
12642 52525 52525
12643 422 422
12644 6205 JSR @DADD ; SET DISK ADDR
12645 0 0
12646 0 0
12647 1 1
12650 6170 RFF1: JSR @RWCMD ; WRITE DATA
12651 3400 WRITE
12652 30462 WBUFF
12653 74 74
```

;DONE, CHECK CONTROL STATUS

```
12654 6202 RFF1A: JSR @CKCS
12655 40000 RWDN
12656 12661 RFF1B ; STATUS OK
12657 6166 JSR @ERROR
12660 12705 RFFEB
```

;CLEAR RBUFF, ISSUE A CLEAR THEN A READ BUFFER COMMAND

```
12661 6176 RFF1B: JSR @PRIME
12662 50462 RBUFF
12663 0 0
12664 0 0
12665 40 40
12666 6170 JSR @RWCMD
12667 3200 RDBF
12670 50462 RBUFF
12671 24 24
```

;DONE, CHECK CONTROL STATUS WORD

```
12672 6202 RFF1C: JSR @CKCS
12673 40000 RWDN
12674 12677 RFF1D ; GOOD STATUS
12675 6166 JSR @ERROR ; BAD STATUS
12676 12713 RFFEB+ONE
```

;CHECK 16 WORDS IN RBUFF TO SEE IF SAME AS IN WBUFF

```
12677 6174 RFF1D: JSR @DCMP ; CALL DATA COMPARE
12700 22 22
12701 12704 RFF1E ; GOOD DATA
```

12702 6166 JSR @ERROR ;BUM DATA  
12703 12721 RFFEB+TWO  
12704 2114 RFF1E:JMP @NTST ;QUIT

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

12705 12654 RFFEB:RFF1A  
12706 12650 RFF1  
12707 23206 WDT  
12710 6 6  
12711 1 1  
12712 15 15

12713 12672 RFF1C ; 1  
12714 12661 RFF1B  
12715 25152 RFFI  
12716 6 6  
12717 1 1  
12720 15 15

12721 12677 RFF1D ; 2  
12722 12661 RFF1B  
12723 22750 DWC  
12724 16 16  
12725 14 14  
12726 122742 @ASWB

;\*\*\*\* END OF DRIVE TEST \*\*\*\*

;PRINT PASS NUMBER THEN RECYCLE TEST

12727 6155 ENDD: JSR @PRINT ; "PASS"  
12730 20005 PAST  
12731 10146 ISZ PASS ; COUNT PASSES  
12732 24146 LDA 1, PASS  
12733 6164 JSR @DECN ; PRINT PASS COUNT  
12734 2401 JMP @. +1  
12735 6400 RECAL

; DRIVE EMULATOR TEST SECTION

; \*\*\*\* READ OFFSET COMMAND EQUATES \*\*\*\*

1000	OF SF = 1000	; READ OFFSET FORWARD
1200	OF SR = 1200	; READ OFFSET REVERSE

; \*\*\*\* DISK ADDRESS FIELD EQUATES \*\*\*\*

20	VOL1 = 20
0	H0 = 0
2000	H1 = 2000
4000	H2 = 4000
10000	H4 = 10000
20000	H10 = 20000
40000	H20 = 40000
0	S0 = 0
40	S1 = 40
100	S2 = 100
140	S3 = 140
200	S4 = 200
300	S6 = 300
400	S10 = 400
440	S11 = 440
600	S14 = 600
740	S17 = 740
1000	S20 = 1000
1100	S22 = 1100
1240	S25 = 1240
1400	S30 = 1400
1740	S37 = 1740
37	N1 = 37
36	N2 = 36

13000 . LOC 13000

;\*\*\*\* RELEASE DRIVE \*\*\*\*

; ISSUE A RELEASE DRIVE COMMAND FOLLOWED BY A CLEAR. CHECK CONTROLLER  
; STATUS FOR ALL DONE RESET

13000	6177	RLD0:	JSR	@SET
13001	24406		RLDOT	
13002	13005		RLD	
13003	13027		DRDS	
13004	0		0	

; RELEASE DRIVE THEN CLEAR COMMAND

13005	6206	RLD:	JSR	@SKCMD
13006	1600		RLSD	
13007	0		DRVO	
13010	0		0	
13011	24		24	
13012	4257		JSR	NNIOC

; CHECK STATUS FOR ALL CLEARED

13013	6202	RLD1:	JSR	@CKCS
13014	0		0	
13015	13020		. +3	
13016	6166		JSR	@ERROR ; SOMETHING WRONG
13017	13021		RLDE0	
13020	2114		JMP	@NTST

;\*\*\*\* ERROR MESSAGE BLOCK \*\*\*\*

13021	13013	RLDE0:	RLD1
13022	13005		RLD
13023	21736		CLDA
13024	1		1
13025	1		1
13026	121445		@DATI3

; \*\*\*\* DRIVE DONE STATUS \*\*\*\*

; TEST SETTING OF EACH DRIVE DONE (ATTENTION) STATUS BY ISSUING A RECAL  
; COMMAND TO THE RESPECTIVE DRIVE ONE AT A TIME

13027	6177	DRDS:	JSR	@SET
13030	24216		DRDNT	
13031	13034		DDS	
13032	13103		REHD	
13033	0		0	

; GET DRIVE NUMBER FROM TEST LIST AND ISSUE A RECAL

13034	30427	DDS:	LDA	2, DRDL
13035	21000		LDA	0, 0, 2 ; DRV #
13036	40403		STA	0, DDS1+2 ; SET IT
13037	6206	DDS1:	JSR	@SKCMD ; ISSUE RECAL
13040	200		RCAL	
13041	0		0	
13042	0		0	
13043	50		50	

; GET REFERENCE STATUS FROM LIST AND CHECK IT AGAINST DIA

13044	21001	DDS1A:	LDA	0, 1, 2
13045	40402		STA	0, .+2 ; SET REF
13046	6202		JSR	@CKCS ; CHECK STAT
13047	0		0	
13050	13053		.+3	; ALL RIGHT
13051	6166		JSR	@ERROR ; NOT ALL RIGHT
13052	13075		DDSEO	

; CLEAR DRIVE DONE AND BUMP LIST POINTER, QUIT IF END OF LIST

13053	101120		MOVZL	0, 0
13054	4211		JSR	DDOA ; CLEAR DONE
13055	151400		INC	2, 2
13056	151400		INC	2, 2
13057	21000		LDA	0, 0, 2 ; NEXT DRV
13060	101112		MOVL#	0, 0, SZC ; END?
13061	2114		JMP	@NTST ; YES, QUIT
13062	754		JMP	DDS+2 ; NO, MORE

; DRIVE NUMBER AND DONE STATUS BIT LIST

13063	13064	DRDL:	.+1	
13064	0		DRV0	
13065	20000		20000	
13066	40		DRV1	
13067	10000		10000	
13070	100		DRV2	
13071	4000		4000	
13072	140		DRV3	
13073	2000		2000	
13074	177777		-1	; END

; \*\*\*\* ERROR MESSAGE BLOCK \*\*\*\*

13075 13046 DDSEO: DDS1A+2

---  
13076 13035 DDS+1  
13077 21710 DRAT  
13100 15 15  
13101 1 1  
13102 15 15



;\*\*\*\* READ EMULATOR HEADER \*\*\*\*

;READ HEADER FROM 6 DIFFERENT LOCATIONS IN EMULATOR

```
13103 6177 REHD: JSR @SET
13104 24426 REHDT
13105 13110 REH
13106 13265 TPRL
13107 0 0
```

; INITIALIZE TEST PARAMETER LIST POINTER, COPY ENTRIES TO THE SEEK AND  
; READ CALLING PARAMETERS OF SUBROUTINES RESPECTIVELY

```
13110 30457 REH: LDA 2,TPL
13111 50142 STA 2,SAVP ;SET LIST POINTER
13112 21000 LDA 0,0,2
13113 40415 STA 0,REH1+2 ;SET SEEK DRV#
13114 21001 LDA 0,1,2
13115 40414 STA 0,REH1+3 ;SET SEEK CYL #
13116 25002 LDA 1,2,2
13117 44422 STA 1,REH1A ;SET READ CMD
13120 25003 LDA 1,3,2
13121 44421 STA 1,REH1A+1;SET DISK ADDR
13122 155000 MOV 2,3
13123 24052 LDA 1,K4
13124 137000 ADD 1,3
13125 54426 STA 3,REH2A+1;SET HEADER REF ADDR
```

; ISSUE SEEK TO THE INITIALIZED CYLINDER

```
13126 6206 REH1: JSR @SKCMD
13127 400 SEEK
13130 0 0
13131 0 0
13132 24 24
```

; PRIME RBUFF WITH 4 WORDS OF -1 THEN READ FORMAT INTO IT

```
13133 6176 JSR @PRIME
13134 50462 RBUFF
13135 177777 -1
13136 177777 -1
13137 4 4
13140 6207 JSR @EMCMD
13141 0 REH1A: 0
13142 0 0
13143 50462 RBUFF
13144 50 50
```

; DONE, CHECK STATUS

```
13145 6201 REH2: JSR @CKCK
13146 40000 RWDN
13147 13152 REH2A ; LOOKS GOOD
13150 6166 JSR @ERROR ; LOOKS BAD
13151 13251 REHEO
```

; COMPARE HEADER IN RBUFF VS REFERENCE

```

13152 6210 REH2A:JSR @CRBUF ;CALL COMPARE
13153 0 0
13154 4 4
13155 13160 REH2B ;NO ERROR
13156 6166 JSR @ERROR
13157 13257 REHE0+ONE

```

; DO NEXT GROUP IN LIST IF NOT ALREADY THE LAST ONE

```

13160 30142 REH2B:LDA 2,SAVP
13161 24055 LDA 1,K10
13162 133000 ADD 1,2 ;INDEX
13163 21000 LDA 0,0,2 ;CHECK ENTRY
13164 101113 MOVL# 0,0,SNC ;END?
13165 724 JMP REH+1 ;NO, GO DO IT
13166 2114 JMP @NTST ;FINISH

```

; TEST PARAMETER GROUPS

; WORDS 0-1 FOR SEEK, 2-3 FOR READ, 4-7 FOR HEADER REFERENCE

```

13167 13170 TPL: .+1
13170 0 DRV0
13171 501 501
13172 3600 RHDR
13173 40077 H20+S1+N1
13174 501 501
13175 40040 H20+S1
13176 0 0
13177 0 0

13200 20 DRV0+VOL 1
13201 502 502
13202 3600 RHDR
13203 20137 H10+S2+N1
13204 2502 2502
13205 20100 H10+S2
13206 0 0
13207 0 0

13210 40 DRV1
13211 504 504
13212 3640 RHDR+DRV1
13213 10237 H4+S4+N1
13214 4504 4504
13215 10200 H4+S4
13216 0 0
13217 0 0

13220 100 DRV2
13221 510 510
13222 3700 RHDR+DRV2
13223 4437 H2+S10+N1
13224 10510 10510
13225 4400 H2+S10
13226 0 0
13227 0 0

13230 140 DRV3
13231 520 520

```

```

13232 3740 RHDR+DRV3
13233 3037 H1+S20+N1
13234 14520 14520
13235 3000 H1+S20
13236 0 0
13237 0 0

13240 160 DRV3+VOL1
13241 1240 1240
13242 3740 RHDR+DRV3
13243 1277 H0+S25+N1
13244 17240 17240
13245 1240 H0+S25
13246 0 0
13247 0 0

13250 177777 -1 ; END

```

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*;

```

13251 13145 REHEO: REH2
13252 13126 REH1
13253 24600 RF1T
13254 6 6
13255 1 1
13256 15 15

13257 13152 REH2A
13260 13126 REH1
13261 22730 HWC
13262 123716 @EHDW
13263 14 14
13264 21 21

```

;\*\*\*\* TRESPASS & RELEASE DRIVE \*\*\*\*

; CHECK EMULATOR HEADER WORD FOR DRIVE TRESPASSED AND RELEASED INDICATION  
; AFTER THE RESPECTIVE COMMAND WAS ISSUED

13265	6177	TPRL:	JSR	@SET
13266	24441		TPRLT	
13267	13272		TPR	
13270	13357		DSAB	
13271	0		0	

; ISSUE A TRESPASS COMMAND TO DRIVE 2

13272	6206	TPR:	JSR	@SKCMD
13273	2000		TRES	
13274	100		DRV2	
13275	0		0	
13276	24		24	

; READ HEADER FROM TRESPASSED DRIVE

13277	6207	TPR1:	JSR	@EMCMD
13300	3700		RHDR+DRV2	
13301	37		HO+S0+N1	
13302	50462		RBUFF	
13303	24		24	

; DONE, CHECK HEADER WORDS VS REFERENCE

13304	6210	TPR1A:	JSR	@CRBUF	; CALL COMPARE
13305	13333		TREFO		
13306	4		4		
13307	13312		TPR2		; CHECKED OUT
13310	6166		JSR	@ERROR	; CRAPPED OUT
13311	13343		TPREO		

; ISSUE A RELEASE COMMAND TO DRIVE 2

13312	6206	TPR2:	JSR	@SKCMD
13313	1600		RLSD	
13314	100		DRV2	
13315	0		0	
13316	24		24	

; DONE, READ HEADER FROM RELEASED DRIVE

13317	6207	TPR2A:	JSR	@EMCMD
13320	3700		RHDR+DRV2	
13321	37		HO+S0+N1	
13322	50462		RBUFF	
13323	24		24	

; DONE, COMPARE HEADER WORDS VS REFERENCE

13324	6210	TPR2B:	JSR	@CRBUF
13325	13337		TREF1	
13326	4		4	
13327	13332		+3	; EXCELLENT!
13330	6166		JSR	@ERROR ; @*!#\$+

13331 13351 TPREO+ONE  
13332 2114 JMP @NTST ; GET OUT OF HERE

; HEADER COMPARE REFERENCE

13333 33240 TREF0: 33240  
13334 0 0  
13335 0 0  
13336 0 0  
13337 13240 TREF1: 13240  
13340 0 0  
13341 0 0  
13342 0 0

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

13343 13304 TPREO: TPR1A  
13344 13272 TPR  
13345 23731 TPD  
13346 15 15  
13347 14 14  
13350 21 21  
  
13351 13324 TPR2B  
13352 13312 TPR2  
13353 23741 RELS  
13354 15 15  
13355 14 14  
13356 21 21

;\*\*\*\* CHECK DRIVE STATUS BITS 1,3,4 & 6 \*\*\*\*

;CHECK WRITE PROTECTED, BUSY, READY AND RESERVED STATUS IN EACH DRIVE

```
13357 6177 DSAB: JSR @SET
13360 24456 DSABT
13361 13364 DSA
13362 13607 DSBB
13363 0 0
```

;SEEK TO CYLINDER 1540

```
13364 6206 DSA: JSR @SKCMD
13365 400 SEEK
13366 0 DRVO
13367 1540 1540
13370 1750 1750
```

;CHECK DRIVE READY STATUS IN EACH DRIVE

```
13371 30421 DSA1: LDA 2, DSEL ; DRV SELECT LIST
13372 21000 LDA 0, 0, 2 ; DRV CMD
13373 101005 MOV 0, 0, SNR ; END OF LIST?
13374 430 JMP DSA2 ; YES
13375 25001 LDA 1, 1, 2 ; GET STATUS REFERENCE
13376 44405 STA 1, DSA1A+1 ; SET IT
13377 4211 JSR DDOA ; NO, SELECT DRIVE
13400 4242 JSR DDIB ; READ STATUS
13401 40133 STA 0, DSTA ; SAVE IT
13402 6203 DSA1A: JSR @CKDS ; CHECK READY ON
13403 0 0
13404 13407 .+3 ; GOOD STATUS
13405 6166 JSR @ERROR ; NOT ON
13406 13521 DSAEB
```

;BUMP LIST POINTER AND CONTINUE

```
13407 151400 INC 2, 2
13410 151400 INC 2, 2
13411 761 JMP DSA1+1
```

;DRIVE SELECT COMMANDS AND STATUS REFERENCE

```
13412 13413 DSEL: .+1
13413 2600 NOOP+DRVO
13414 11000 DRDY+WPRT
13415 2640 NOOP+DRV1
13416 14000 DRDY+DBSY
13417 2700 NOOP+DRV2
13420 0 0
13421 2740 NOOP+DRV3
13422 50000 DRDY+RESV
13423 0 0
```

;ISSUE FORMAT COMMAND TO DRV 0

```
13424 6207 DSA2: JSR @EMCMD
13425 1400 FMT+DRVO
13426 37 H0+S0+N1
```

13427 30462 WBUFF  
13430 24 24

; DONE, CHECK FOR RW FAULT IN CONTROL STATUS

13431 6202 DSA2A: JSR @CKCS  
13432 40001 RWDN+RWFT  
13433 13436 DSA2B ; CORRECT  
13434 6166 JSR @ERROR ; INCORRECT  
13435 13527 DSAEB+ONE

; CHECK FOR ILLEGAL ADDRESS + DRV ERR IN DRIVE STATUS

13436 6203 DSA2B: JSR @CKDS  
13437 11101 DRDY+WPRT+RWFT+ILCMD  
13440 13443 DSA3 ; RIGHT  
13441 6166 JSR @ERROR ; WRONG  
13442 13535 DSAEB+TWO

; ISSUE FORMAT COMMAND TO DRIVE 1

13443 6207 DSA3: JSR @EMCMD  
13444 1440 FMAT+DRV1  
13445 37 HO+SO+N1  
13446 30462 WBUFF  
13447 3720 3720

; DONE, CHECK FOR RW TIMEOUT + RW FAULT IN CONTROL STATUS

13450 6202 DSA3A: JSR @CKCS  
13451 40005 RWDN+RWTO+RWFT  
13452 13455 DSA3B ; OK  
13453 6166 JSR @ERROR ; NOT OK  
13454 13543 DSAEB+THREE

; CHECK FOR BUSY IN DRIVE STATUS

13455 6203 DSA3B: JSR @CKDS  
13456 14000 DRDY+DBSY  
13457 13462 DSA4 ; GOOD  
13460 6166 JSR @ERROR ; BAD  
13461 13551 DSAEB+FOUR

; ISSUE A FORMAT COMMAND TO DRIVE 2

13462 6207 DSA4: JSR @EMCMD  
13463 1500 FMAT+DRV2  
13464 37 HO+SO+N1  
13465 30462 WBUFF  
13466 24 24

; DONE, CHECK FOR DRIVE 2 ATTENTION IN CONTROL STATUS

13467 6202 DSA4A: JSR @CKCS  
13470 44000 RWDN+4000  
13471 13474 DSA4B ; YES  
13472 6166 JSR @ERROR ; NO  
13473 13557 DSAEB+FIVE

; CHECK FOR DRIVE NOT READY IN DRIVE STATUS

13474	6203	DSA4B: JSR	@CKDS
13475	0	0	
13476	13501	DSA5	; IT IS THAT
13477	6166	JSR	@ERROR ; NOT SO
13500	13565	DSAEB+SIX	

; ISSUE A FORMAT COMMAND TO DRIVE 3

13501	6207	DSA5: JSR	@EMCMD
13502	1540	FMAT+DRV3	
13503	37	HO+50+N1	
13504	30462	WBUF	
13505	24	24	

; DONE, CHECK FOR DRIVE 3 DONE IN CONTROL STATUS

13506	6202	DSA5A: JSR	@CKCS
13507	42000	RWDN+2000	
13510	13513	DSA5B	; OK
13511	6166	JSR	@ERROR ; NOT OK
13512	13573	DSAEB+SEVEN	

; CHECK FOR DRIVE RESERVED IN DRIVE STATUS

13513	6203	DSA5B: JSR	@CKDS
13514	50000	DRDY+RESV	
13515	13520	.+3	; RESERVED
13516	6166	JSR	@ERROR ; NO
13517	13601	DSAEB+EIGHT	
13520	2114	JMP	@NTST ; ALL DONE

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

13521	13402	DSAEB: DSA1A	
13522	13372	DSA1+1	
13523	23750	RDYS	
13524	1	1	
13525	2	2	
13526	16	16	
13527	13431	DSA2A	; 1
13530	13424	DSA2	
13531	23762	SSCK	
13532	6	6	
13533	1	1	
13534	15	15	
13535	13436	DSA2B	; 2
13536	13424	DSA2	
13537	23775	WPFT	
13540	6	6	
13541	2	2	
13542	16	16	
13543	13450	DSA3A	; 3
13544	13443	DSA3	
13545	23762	SSCK	



13546	6	6
13547	1	1
13550	15	15
13551	13455	DSA3B ; 4
13552	13443	DSA3
13553	22671	DBS
13554	6	6
13555	2	2
13556	16	16
13557	13467	DSA4A ; 5
13560	13462	DSA4
13561	21710	DRAT
13562	6	6
13563	1	1
13564	15	15
13565	13474	DSA4B ; 6
13566	13462	DSA4
13567	22655	DNST
13570	6	6
13571	2	2
13572	16	16
13573	13506	DSA5A ; 7
13574	13501	DSA5
13575	21710	DRAT
13576	6	6
13577	1	1
13600	15	15
13601	13513	DSA5B ; 8
13602	13501	DSA5
13603	22565	DRST
13604	6	6
13605	2	2
13606	16	16

- PAGE 59 -

;\*\*\*\* CHECK DRIVE STATUS BITS 0,9 & 10-12 \*\*\*\*

;CHECK SETTING OF ILLEGAL CYLINDER, ILLEGAL COMMAND, DRIVE FAULT AND  
;DRIVE ERROR STATUS

13607	6177	DSBB:	JSR	@SET
13610	24456		DSABT	
13611	13614		DSB	
13612	13655		DSCB	
13613	0		0	

;SEEK CYL 1740 TO SET ILLEGAL CYLINDER STATUS

13614	6206	DSB:	JSR	@SKCMD
13615	400		SEEK	
13616	0		DRVO	
13617	1740		1740	
13620	24		24	

;DONE, CHECK FOR ILLEGAL CYLINDER STATUS

13621	6203	DSB1:	JSR	@CKDS
13622	10201		DRDY+ISKA+RWFT	
13623	13626		DSB2 ; GOOD STATUS	
13624	6166		JSR @ERROR ; NO GOOD	
13625	13641		DSBEB	

;SEEK CYL 1147 TO SET DRIVE ERROR BIT IN STATUS

13626	6206	DSB2:	JSR	@SKCMD
13627	400		SEEK	
13630	0		DRVO	
13631	1147		1147	
13632	24		24	

;DONE, CHECK STATUS FOR ERROR

13633	6203	DSB2A:	JSR	@CKDS
13634	10071		DRDY+DRFL+RWFT	
13635	13640		. +3 ; ERROR ON	
13636	6166		JSR @ERROR ; NOT ON	
13637	13647		DSBEB+ONE	
13640	2114		JMP @NTST ; FINISH	

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

13641	13621	DSBEB:	DSB1	
13642	13614		DSB	
13643	24015		ILCS	
13644	15		15	
13645	2		2	
13646	16		16	
13647	13633		DSB2A ; 1	
13650	13626		DSB2	
13651	24046		CIDER	
13652	15		15	
13653	2		2	
13654	16		16	

;\*\*\*\* DRIVE STATUS CHECK \*\*\*\*

13655	6177	DSCB:	JSR	@SET
13656	24456		DSABT	
13657	13662		DSC	
13660	14020		FRMS	
13661	0		0	

; SEEK CYL 600 ON DRV 0 TO FORMAT A SECTOR

13662	6206	DSC:	JSR	@SKCMD
13663	400		SEEK	
13664	0		DRVO	
13665	600		600	
13666	24		24	
13667	6207		JSR	@EMCMD
13670	1400		FMAT	
13671	37		HO+S0+N1	
13672	30462		WBUFF	
13673	24		24	

; DONE, READ FORMAT INTO RBUFF

13674	6207	DSC1:	JSR	@EMCMD
13675	3600		RHDR	
13676	37		HO+S0+N1	
13677	50462		RBUFF	
13700	24		24	

; SEEK TO CYL 600 ON DRV 1

13701	6206	DSC1A:	JSR	@SKCMD
13702	400		SEEK	
13703	40		DRV1	
13704	600		600	
13705	24		24	
13706	6207		JSR	@EMCMD
13707	1440		FMAT+DRV1	
13710	37		HO+S0+N1	
13711	30462		WBUFF	
13712	24		24	

; READ FORMAT TO SET DRIVE ERROR STATUS

13713	6207	DSC1B:	JSR	@EMCMD
13714	3640		RHDR+DRV1	
13715	37		HO+S0+N1	
13716	50462		RBUFF	
13717	24		24	

; DONE, CHECK DRIVE STATUS FOR ERROR

13720	6201	DSC1C:	JSR	@CKCK
13721	40001		RWDN+RWFT	
13722	13725		DSC2	; FAULT SET
13723	6166		JSR	@ERROR ; NOT SET
13724	13776		DSCEB	

; SEEK CYL 600 ON DRV 2

```

13725 6206 DSC2: JSR @SKCMD
13726 400 SEEK
13727 100 DRV2
13730 600 600
13731 24 24
13732 6207 JSR @EMCMD
13733 1500 FMAT+DRV2
13734 37 HO+SO+N1
13735 30462 WBUFF
13736 24 24

```

; READ FORMAT TO SET STATUS

```

13737 6207 DSC2A: JSR @EMCMD
13740 3700 RHDR+DRV2
13741 37 HO+SO+N1
13742 50462 RBUFF
13743 24 24

```

; DONE, CHECK FOR SEEK CYL ERROR STATUS

```

13744 6201 DSC2B: JSR @CKCK
13745 40001 RWDN+RWFT
13746 13751 DSC3 ; OK
13747 6166 JSR @ERROR ; NOT OK
13750 14004 DSCEB+ONE

```

; SEEK CYL 600 ON DRV 3 AND FORMAT

```

13751 6206 DSC3: JSR @SKCMD
13752 400 SEEK
13753 140 DRV3
13754 600 600
13755 24 24
13756 6207 JSR @EMCMD
13757 1540 FMAT+DRV3
13760 37 HO+SO+N1
13761 30462 WBUFF
13762 24 24

```

; READ FORMAT TO SET STATUS

```

13763 6207 DSC3A: JSR @EMCMD
13764 3740 RHDR+DRV3
13765 37 HO+SO+N1
13766 50462 RBUFF
13767 24 24

```

; DONE, CHECK FOR DRV RESERVED STATUS

```

13770 6201 DSC3B: JSR @CKCK
13771 40001 RWDN+RWFT
13772 13775 . +3 ; RIGHT STATUS
13773 6166 JSR @ERROR ; WRONG
13774 14012 DSCEB+TWO
13775 2114 JMP @NTST ; BYE

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

162

13776 13720 DSCEB: DSC1C  
13777 13701 DSC1A  
14000 24063 CDRF  
14001 6 6  
14002 1 1  
14003 1 1  
  
14004 13744 DSC2B ; 1  
14005 13725 DSC2  
14006 24063 CDRF  
14007 6 6  
14010 1 1  
14011 15 15  
  
14012 13770 DSC3B ; 2  
14013 13751 DSC3  
14014 24123 CDRV  
14015 6 6  
14016 1 1  
14017 15 15

;\*\*\*\* WRITE AND READ FORMAT \*\*\*\*

; USE THE FORMAT FUNCTION OF THE CONTROLLER TO FORMAT 1 SECTOR  
; READ AND VERIFY THE OPERATION

14020	6177	FRMS:	JSR	@SET
14021	24467		FRMST	
14022	14025		FRM	
14023	14147		WRHD	
14024	0		0	

; SEEK TO CYL 0 TO FORMAT HD 0, SECT 0

14025	6206	FRM:	JSR	@SKCMD
14026	400		SEEK	
14027	0		DRVO	
14030	0		0	
14031	50		50	
14032	6207		JSR	@EMCMD
14033	1400		FMAT	
14034	37		HO+50+N1	
14035	30462		WBUFF	
14036	50		50	

; DONE, CHECK STATUS

14037	6202	FRM1:	JSR	@CKCS
14040	40000		RWDN	
14041	14044		FRM1A	; OK
14042	6166		JSR	@ERROR ; NOT OK
14043	14111		FRMEB	

; CHECK DISK ADDRESS WORD

14044	6204	FRM1A:	JSR	@CKDA
14045	40		40	
14046	14051		FRM2	; OK
14047	6166		JSR	@ERROR ; NOT OK
14050	14117		FRMEB+ONE	

; SEED RBUFF WITH -1S THEN READ FORMAT INTO IT

14051	6176	FRM2:	JSR	@PRIME
14052	50462		RBUFF	
14053	177777		-1	
14054	177777		-1	
14055	6		6	
14056	6207		JSR	@EMCMD
14057	3600		RHDR	
14060	37		HO+50+N1	
14061	50462		RBUFF	
14062	50		50	

; DONE, CHECK CONTROL STATUS

14063	6202	FRM2A:	JSR	@CKCS
14064	40000		RWDN	
14065	14070		FRM2B	; GOOD ONE
14066	6166		JSR	@ERROR ; BAD STATUS

14067 14125 FRMEB+TWO

; CHECK DISK ADDRESS INCREMENT

14070 6204 FRM2B: JSR @CKDA  
14071 40 40  
14072 14075 FRM2C ; RIGHT INCREMENT  
14073 6166 JSR @ERROR ; WRONG THING  
14074 14133 FRMEB+THREE

; VERIFY 6 WORDS OF HEADER ARE ZEROS

14075 6210 FRM2C: JSR @CRBUF  
14076 14103 FMB  
14077 14102 .+3 ; ALL IS FINE  
14100 6166 JSR @ERROR ; SOMETHING WRONG  
14101 14141 FRMEB+FOUR  
14102 2114 JMP @NTST ; EXIT

14103 0 FMB: 0  
14104 0 0  
14105 0 0  
14106 0 0  
14107 0 0  
14110 0 0

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

14111 14037 FRMEB: FRM1  
14112 14025 FRM  
14113 24570 FM1T  
14114 6 6  
14115 1 1  
14116 15 15  
  
14117 14044 FRM1A ; 1  
14120 14025 FRM  
14121 21271 DARG  
14122 6 6  
14123 3 3  
14124 17 17  
  
14125 14063 FRM2A ; 2  
14126 14051 FRM2  
14127 24600 RF1T  
14130 6 6  
14131 1 1  
14132 15 15  
  
14133 14070 FRM2B ; 3  
14134 14051 FRM2  
14135 21271 DARG  
14136 6 6  
14137 3 3  
14140 17 17  
  
14141 14075 FRM2C ; 4  
14142 14051 FRM2  
14143 24600 RF1T

14144 122730  
14145 14  
14146 21

@HWC  
14  
21



;\*\*\*\* WRITE AND READ HEADER \*\*\*\*

;WRITE HEADER FROM MEMORY  
;READ HEAD BACK AND VERIFY IT

14147	6177	WRHD:	JSR	@SET
14150	24502		WRHDT	
14151	14154		WRH	
14152	14260		BSBT	
14153	0		0	

;SET UP HEAD WORDS IN WBUFF AND WRITE IT OUT TO HD 0, SECT 0

14154	34115	WRH:	LDA	3,WBUF	;WBUFF
14155	102400		SUB	0,0	
14156	41400		STA	0,0,3	;SET WORDS 0 & 1 = 0
14157	41401		STA	0,1,3	
14160	100000		COM	0,0	
14161	41402		STA	0,2,3	;SET WORD 2 = -1
14162	6207	WRH1:	JSR	@EMCMD	
14163	600		WHDR		
14164	37		HO+S0+N1		
14165	30462		WBUFF		
14166	50		50		

;DONE, CHECK STATUS

14167	6202	WRH1A:	JSR	@CKCS
14170	40000		RWDN	
14171	14174		WRH2	;GOOD STATUS
14172	6166		JSR	@ERROR ;BAD
14173	14230		WRHEB	

;PRIME RBUFF WITH COMPLEMENT OF HEADER WORDS THEN READ HEADER INTO IT

14174	34116	WRH2:	LDA	3,RBUF	
14175	102000		ADC	0,0	
14176	41400		STA	0,0,3	;WORDS 0 & 1 = -1
14177	41401		STA	0,1,3	
14200	100000		COM	0,0	
14201	41402		STA	0,2,3	;WORD 2 = 0
14202	6207		JSR	@EMCMD	
14203	3600		RHDR		
14204	37		HO+S0+N1		
14205	50462		RBUFF		
14206	50		50		

;DONE, CHECK STATUS

14207	6202	WRH2A:	JSR	@CKCS
14210	40000		RWDN	
14211	14214		WRH2B	;OK
14212	6166		JSR	@ERROR
14213	14236		WRHEB+ONE	

;CHECK DISK ADDRESS WORD

14214	6204	WRH2B:	JSR	@CKDA
14215	40		40	

-----  
14216 14221 WRH2C ; OK  
14217 6166 JSR @ERROR  
14220 14244 WRHEB+TWO

; VERIFY HEADER IN RBUF

14221 6210 WRH2C: JSR @CRBUF  
14222 30462 WBUF  
14223 3 3  
14224 14227 . +3 ; GOOD ONE  
14225 6166 JSR @ERROR ; BAD  
14226 14252 WRHEB+THREE  
14227 2114 JMP @NTST

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

14230 14167 WRHEB: WRH1A  
14231 14162 WRH1  
14232 24613 WH1T  
14233 6 6  
14234 1 1  
14235 15 15

14236 14207 WRH2A ; 1  
14237 14174 WRH2  
14240 24600 RF1T  
14241 6 6  
14242 1 1  
14243 15 15

14244 14214 WRH2B ; 2  
14245 14174 WRH2  
14246 21271 DARG  
14247 6 6  
14250 3 3  
14251 17 17

14252 14221 WRH2C ; 3  
14253 14174 WRH2  
14254 22730 HWC  
14255 16 16  
14256 14 14  
14257 122742 @ASWB

;\*\*\*\* BAD SECTOR STATUS \*\*\*\*

;SET UP A BAD SECTOR IN HD 0, SECT 0. WHEN ATTEMPTING TO READ DATA FROM  
;IT, THE OPERATION WOULD BE ABORTED DUE TO THE BAD SECTOR BIT WAS SET IN  
;THE HEADER

```
14260 6177 BSBT: JSR @SET
14261 25106 BSST
14262 14265 BSB
14263 14362 SSDT
14264 0 0
```

;SET UP A BAD SECTOR HEADER IN WBUFF AND WRITE IT OUT

```
14265 34115 BSB: LDA 3,WBUF
14266 102620 SUBZR 0,0
14267 41400 STA 0,0,3 ;WORD 0 = BS
14270 101120 MOVZL 0,0
14271 41401 STA 0,1,3 ;WORDS 1 & 2 = 0
14272 41402 STA 0,2,3
14273 6207 BSB1: JSR @EMCMD
14274 600 WHDR
14275 37 HO+SO+N1
14276 30462 WBUFF
14277 24 24
```

;DONE, CHECK STATUS

```
14300 6202 BSB1A: JSR @CKCS
14301 40000 RWDN
14302 14305 BSB1B ;GOOD DONE
14303 6166 JSR @ERROR ;BAD DONE
14304 14340 BSBEB
```

;PRIME RBUFF WITH -1S THEN READ HEADER BACK

```
14305 6176 BSB1B: JSR @PRIME
14306 50462 RBUFF
14307 177777 -1
14310 177777 -1
14311 6 6
14312 6207 JSR @EMCMD
14313 3600 RHDR
14314 37 HO+SO+N1
14315 50462 RBUFF
14316 24 24
```

;DONE, VERIFY HEADER WAS WRITTEN CORRECTLY

```
14317 6210 BSB1C: JSR @CRBUF
14320 30462 WBUFF
14321 3 3
14322 14325 BSB2 ;GOOD HEADER
14323 6166 JSR @ERROR ;NOT SO GOOD
14324 14346 BSBEB+ONE
```

;TRY TO READ DATA FROM BAD SECTOR

```
14325 6207 BSB2: JSR @EMCMD
```

14326 0 READ  
14327 37 HO+SO+N1  
14330 50462 RBUFF  
14331 24 24

; DONE, CHECK FOR BAD SECTOR STATUS

14332 6202 BSB2A: JSR @CKCS  
14333 40101 RWDN+BSCT+RWFT  
14334 14337 . +3 ; SHOWS OK  
14335 6166 JSR @ERROR  
14336 14354 BSBE+TWO  
14337 2114 JMP @NTST

; \*\*\*\* ERROR MESSAGE BLOCK \*\*\*\*

14340 14300 BSBEB: BSB1A  
14341 14273 BSB1  
14342 23117 WHD  
14343 6 6  
14344 1 1  
14345 15 15

14346 14317 BSB1C ; 1  
14347 14305 BSB1B  
14350 22730 HWC  
14351 16 16  
14352 14 14  
14353 122742 @ASWB

14354 14332 BSB2A ; 2  
14355 14325 BSB2  
14356 25106 BSSTT  
14357 6 6  
14360 1 1  
14361 15 15

;\*\*\*\* ONE SECTOR DATA WRITE, READ & VERIFY \*\*\*\*

```
14362 6177 SSDT: JSR @SET
14363 24514      SSDTT
14364 14367      SSD
14365 14607      TSDT
14366 0          0
```

;SET DATA PATTERN POINTER AND PRIME WBUFF WITH PATTERN

```
14367 34536 SSD: LDA 3,PATN
14370 54142      STA 3,SAVP
14371 21400      LDA 0,0,3 ;GET PATTERN
14372 101005     MOV 0,0,SNR ;END?
14373 2114       JMP @NTST ;YES, EXIT
14374 25401      LDA 1,1,3 ;NO, GET 2ND PATTERN WORD
14375 40404      STA 0,SSD1 ;SET THEM
14376 44404      STA 1,SSD1+1
14377 6176       JSR @PRIME
14400 30462      WBUFF
14401 0          SSD1: 0
14402 0          0
14403 400        400
```

;SEEK TO CYL 0 AND FORMAT HO, SO

```
14404 6206 SSD1A: JSR @SKCMD
14405 400        SEEK
14406 0          DRVO
14407 0          0
14410 24         24
14411 6207      JSR @EMCMD
14412 1400      FMAT
14413 37        HO+SO+N1
14414 30462     WBUFF
14415 24        24
```

;DONE, CHECK STATUS

```
14416 6202 SSD1B: JSR @CKCS
14417 40000     RWDN
14420 14423     SSD2
14421 6166      JSR @ERROR ;BAD STATUS
14422 14543     SSDEB
```

;WRITE DATA PATTERN TO SECTOR

```
14423 6207 SSD2: JSR @EMCMD
14424 3400     WRITE
14425 37       HO+SO+N1
14426 30462     WBUFF
14427 24       24
```

;DONE, CHECK STATUS

```
14430 6202 SSD2A: JSR @CKCS
14431 40000     RWDN
14432 14435     SSD2B ;STATUS GOOD
14433 6166      JSR @ERROR ;STATUS BAD
```

14434 14551 SSDEB+ONE

; CLEAR RBUFF BEFORE READ DATA INTO IT

```
14435 6176 SSD2B: JSR @PRIME
14436 50462 RBUFF
14437 0 0
14440 0 0
14441 400 400
14442 6207 JSR @EMCMD
14443 0 READ
14444 37 HO+SO+N1
14445 50462 RBUFF
14446 24 24
```

; DONE, CHECK STATUS

```
14447 6202 SSD2C: JSR @CKCS
14450 40000 RWDN
14451 14454 SSD2D
14452 6166 JSR @ERROR ; BAD STATUS
14453 14557 SSDEB+TWO
```

; COMPARE DATA, RBUFF VS WBUFF

```
14454 6174 SSD2D: JSR @DCMP
14455 400 400
14456 14461 SSD3 ; NO ERROR
14457 6166 JSR @ERROR ; ERROR
14460 14565 SSDEB+THREE
```

; USE VERIFY FUNCTION OF CONTROLLER TO VERIFY DATA

```
14461 6207 SSD3: JSR @EMCMD
14462 3000 DATV
14463 37 HO+SO+N1
14464 30462 WBUFF
14465 24 24
```

; DONE, CHECK STATUS FOR NO VERIFY ERROR

```
14466 6202 SSD3A: JSR @CKCS
14467 40000 RWDN
14470 14473 SSD3B ; OK
14471 6166 JSR @ERROR ; NOT OK
14472 14573 SSDEB+FOUR
```

; DO READ FORMAT TO GET ECC WORDS

```
14473 6207 SSD3B: JSR @EMCMD
14474 3600 RHDR
14475 37 HO+SO+N1
14476 50462 RBUFF
14477 24 24
```

; DONE, COMPARE ECC WORDS WITH REFERENCE

```
14500 30116 SSD3C: LDA 2, RBUF
14501 24052 LDA 1, K4
```

```

14502 133000      ADD      1, 2
14503  50154      STA      2, RBFP      ; SET RBUFF POINTER
14504  34142      LDA      3, SAVP      ; REF POINTER
14505  21000      LDA      0, 0, 2      ; GET READ ECC
14506  25402      LDA      1, 2, 3      ; GET 1ST ECC REF
14507 106414      SSD3E: SUB#    0, 1, SZR      ; MATCH?
14510   406       JMP      SSD3D      ; NO
14511  10154      ISZ      RBFP      ; YES, BUMP POINTER
14512  22154      LDA      0, @RBFP      ; GET 2ND ECC
14513  25403      LDA      1, 3, 3
14514 106415      SUB#    0, 1, SNR      ; OK?
14515   404       JMP      SSD4      ; YES
14516  44141      SSD3D: STA     1, ACN      ; NO, SAVE REF
14517   6166      JSR      @ERROR
14520  14601      SSDEB+FIVE

```

; BUMP PATTERN POINTER THEN CONTINUE

```

14521  34142      SSD4: LDA      3, SAVP
14522  24052      LDA      1, K4
14523 137000      ADD      1, 3
14524   644       JMP      SSD+1      ; LOOP BACK

```

; DATA PATTERNS (WORDS 0 & 1) AND ECC REFERENCE (WORDS 2 & 3)

```

14525  14526      PATN:  . +1
14526 134270      134270
14527  56134      56134
14530  62574      62574
14531  77016      77016

14532  27056      27056
14533  13427      13427
14534 114417      114417
14535  15601      15601

14536 105613      105613
14537 142705      142705
14540 150073      150073
14541  47503      47503

14542   0         0

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

14543  14416      SSDEB: SSD1B
14544  14404      SSD1A
14545   6473      FM1A
14546   6         6
14547   1         1
14550   15        15

14551  14430      SSD2A ; 1
14552  14423      SSD2
14553  23206      WDT
14554   6         6
14555   1         1
14556   15        15

```

14557	14447	SSD2C ; 2
14560	14435	SSD2B
14561	23214	RDT
14562	6	6
14563	1	1
14564	15	15
14565	14454	SSD2D ; 3
14566	14435	SSD2B
14567	22750	DWC
14570	16	16
14571	14	14
14572	122742	@ASWB
14573	14466	SSD3A ; 4
14574	14461	SSD3
14575	23226	VBD
14576	6	6
14577	1	1
14600	15	15
14601	14507	SSD3E ; 5
14602	14473	SSD3B
14603	23221	CECC
14604	123140	@RHD
14605	14	14
14606	21	21



;\*\*\*\* TWO SECTOR DATA WRITE & READ \*\*\*\*

```
14607 6177 TSDT: JSR @SET
14610 24536 TSDTT
14611 14615 TSD
14612 14770 SCHN
14613 0 0
```

; SET UP DISK ADDRESS AT VARIOUS COMMAND LOCATIONS

```
14614 1776 S37+N2
14615 20511 TSD: LDA 0,HLAST ;LAST HD
14616 24776 LDA 1,TSD-1
14617 123000 ADD 1,0 ;FORM DISK ADDR WORD
14620 40412 STA 0,TSD1+7 ;SET IT
14621 40443 STA 0,TSD2A
14622 40466 STA 0,TSD3A
```

; SEEK TO CYL 1 TO FORMAT 2 SECTORS STARTING FROM LAST HEAD & SECT

```
14623 6206 TSD1: JSR @SKCMD
14624 400 SEEK
14625 0 DRVO
14626 1 1
14627 24 24
14630 6207 JSR @EMCMD
14631 1400 FMAT
14632 0 0
14633 30462 WBUFF
14634 120 120
```

; DONE, CHECK STATUS

```
14635 6201 TSD1A: JSR @CKCK
14636 40000 RWDN
14637 14642 TSD1B ;STATUS OK
14640 6166 JSR @ERROR ;STATUS BAD
14641 14732 TSDEB
```

; READ ONE HEADER FROM CYL 2, HEAD 0, SECT 0

```
14642 6207 TSD1B: JSR @EMCMD
14643 3600 RHDR
14644 0 0
14645 50462 RBUFF
14646 120 120
```

; VERIFY HEADER READ INTO RBUFF VS REFERENCE IN TSHD

```
14647 6210 TSD1C: JSR @CRBUF
14650 14727 TSHD
14651 3 3
14652 14655 TSD2 ;NO ERROR
14653 6166 JSR @ERROR
14654 14740 TSDEB+ONE
```

; SEEK BACK TO CYL 1 TO WRITE 2 SECTORS OF DATA STARTING FROM LAST HEAD  
; AND SECTOR

```

14655 6206 TSD2: JSR @SKCMD
14656 400 SEEK
14657 0 DRVO
14660 1 1
14661 24 24
14662 6207 JSR @EMCMD
14663 3400 WRITE
14664 0 TSD2A: 0
14665 30462 WBUFF
14666 120 120

```

; DONE, CHECK STATUS

```

14667 6201 TSD2B: JSR @CKCK
14670 40000 RWDN
14671 14674 TSD3
14672 6166 JSR @ERROR
14673 14746 TSDEB+TWO

```

; SEED RBUFF WITH -1S THEN READ DATA INTO IT

```

14674 6176 TSD3: JSR @PRIME
14675 50462 RBUFF
14676 177777 -1
14677 177777 -1
14700 1000 1000
14701 6206 JSR @SKCMD
14702 400 SEEK
14703 0 DRVO
14704 1 1
14705 24 24
14706 6207 JSR @EMCMD
14707 0 READ
14710 0 TSD3A: 0
14711 50462 RBUFF
14712 24 24

```

; DONE, CHECK STATUS

```

14713 6201 TSD3B: JSR @CKCK
14714 40000 RWDN
14715 14720 TSD3C ; GOOD STATUS
14716 6166 JSR @ERROR ; BAD ONE
14717 14754 TSDEB+THREE

```

; COMPARE DATA, RBUFF VS WBUFF

```

14720 6174 TSD3C: JSR @DCMP
14721 1000 1000
14722 14725 +3 ; GOOD DATA
14723 6166 JSR @ERROR ; BAD DATA
14724 14762 TSDEB+FOUR
14725 2114 JMP @NTST

14726 42000 HLAST: 42000
14727 2 TSHD: 2 ; REFERENCE HEADER BLOCK
14730 0 0
14731 0 0

```

;\*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*;

14732	14635	TSDEB: TSD1A	
14733	14623	TSD1	
14734	23271	FMTG	
14735	6	6	
14736	1	1	
14737	15	15	
14740	14647	TSD1C	; 1
14741	14642	TSD1B	
14742	22730	HWC	
14743	123140	@RHD	
14744	14	14	
14745	21	21	
14746	14667	TSD2B	; 2
14747	14655	TSD2	
14750	23206	WDT	
14751	6	6	
14752	1	1	
14753	15	15	
14754	14713	TSD3B	; 3
14755	14674	TSD3	
14756	23214	RDT	
14757	6	6	
14760	1	1	
14761	15	15	
14762	14720	TSD3C	; 4
14763	14674	TSD3	
14764	22750	DWC	
14765	16	16	
14766	14	14	
14767	122742	@ASWB	

;\*\*\*\* SECTOR CHAINING \*\*\*\*;

;TEST THE ABILITY OF THE CONTROLLER TO READ DATA FROM THE CHAINED SECTOR

14770	6177	SCHN:	JSR	@SET
14771	25117		CHNT	
14772	14775		SCH	
14773	15221		ROFS	
14774	0		0	

;SEEK TO CYL 1 TO FORMAT A TRACK FOR ALTERNATE SECTORS

14775	6206	SCH:	JSR	@SKCMD
14776	400		SEEK	
14777	0		DRVO	
15000	1		1	
15001	24		24	
15002	6207		JSR	@EMCMD
15003	1400		FMAT	
15004	77		HO+S1+N1	
15005	30462		WBUFF	
15006	120		120	

;DONE, CHECK STATUS

15007	6202	SCH1:	JSR	@CKCS
15010	40000		RWDN	
15011	15014		SCH1A	; ALL OK
15012	6166		JSR	@ERROR ; SOMETHING WRONG
15013	15141		SCHEB	

;SET A DATA PATTERN IN WBUFF THEN WRITE IT OUT TO SECT 1 OF ALT SECT

15014	6176	SCH1A:	JSR	@PRIME
15015	30462		WBUFF	
15016	125252		125252	
15017	52525		52525	
15020	400		400	
15021	6207		JSR	@EMCMD
15022	3400		WRITE	
15023	77		HO+S1+N1	
15024	30462		WBUFF	
15025	120		120	

;DONE, CHECK STATUS

15026	6202	SCH1B:	JSR	@CKCS
15027	40000		RWDN	
15030	15033		SCH1C	; LOOKS FINE
15031	6166		JSR	@ERROR ; PROBLEM
15032	15147		SCHEB+ONE	

;READ THAT SECTOR BACK INTO RBUFF

15033	6176	SCH1C:	JSR	@PRIME
15034	50462		RBUFF	
15035	0		0	
15036	0		0	
15037	400		400	

15040 6207 JSR @EMCMD  
15041 0 READ  
15042 77 HO+S1+N1  
15043 50462 RBUFF  
15044 120 120

; DONE, CHECK STATUS

15045 6202 SCH1D: JSR @CKCS  
15046 40000 RWDN  
15047 15052 SCH1E ; OK  
15050 6166 JSR @ERROR ; NOT OK  
15051 15155 SCHEB+TWO

; VERIFY DATA, RBUFF VS WBUFF

15052 6174 SCH1E: JSR @DCMP  
15053 400 400  
15054 15057 SCH2 ; DATA GOOD  
15055 6166 JSR @ERROR ; DATA BAD  
15056 15163 SCHEB+THREE

; SEEK TO CYL 0 TO WRITE HEADER TO SECT 0

15057 6206 SCH2: JSR @SKCMD  
15060 400 SEEK  
15061 0 DRVO  
15062 0 0  
15063 24 24  
15064 6207 JSR @EMCMD  
15065 600 WHDR  
15066 37 HO+S0+N1  
15067 15136 CCHD  
15070 120 120

; DONE, CHECK STATUS

15071 6202 SCH2A: JSR @CKCS  
15072 40000 RWDN  
15073 15076 SCH2B ; ALL RIGHT  
15074 6166 JSR @ERROR ; TROUBLE  
15075 15171 SCHEB+FOUR

; READ HEADER BACK INTO RBUFF

15076 6207 SCH2B: JSR @EMCMD  
15077 3600 RHDR  
15100 37 HO+S0+N1  
15101 50462 RBUFF  
15102 120 120

; VERIFY THAT THE CHAINED HEADER WAS WRITTEN CORRECTLY

15103 6210 SCH2C: JSR @CRBUF ; COMPARE HEADERS  
15104 15136 CCHD  
15105 3 3  
15106 15111 SCH3 ; GOOD HEADER  
15107 6166 JSR @ERROR ; NO GOOD  
15110 15177 SCHEB+FIVE

179

---  
; RESET RBUFF THEN READ DATA FROM ALT SECT. IF WORKING

15111	6176	SCH3:	JSR	@PRIME
15112	50462		RBUFF	
15113	177777		-1	
15114	177777		-1	
15115	400		400	
15116	6207		JSR	@EMCMD
15117	0		READ	
15120	37		H0+50+N1	
15121	50462		RBUFF	
15122	24		24	

; DONE, CHECK STATUS

15123	6201	SCH3A:	JSR	@CKCK
15124	40000		RWDN	
15125	15130		SCH3B	
15126	6166		JSR	@ERROR ; BAD STATUS
15127	15205		SCHEB+SIX	

; VERIFY THAT DATA IS READ FROM THE CHAINED SECTOR

15130	6174	SCH3B:	JSR	@DCMP ; COMPARE RBUFF VS WBUFF
15131	400		400	
15132	15135		. +3	
15133	6166		JSR	@ERROR ; BAD NEWS
15134	15213		SCHEB+SEVEN	
15135	2114		JMP	@NTST ; EXIT TEST

; CHAIN HEADER BLOCK

15136	40000	CCHD:	40000
15137	1		1
15140	1		1

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

15141	15007	SCHEB:	SCH1
15142	14775		SCH
15143	23271		FMTG
15144	6		6
15145	1		1
15146	15		15
15147	15026		SCH1B ; 1
15150	15014		SCH1A
15151	23206		WDT
15152	6		6
15153	1		1
15154	15		15
15155	15045		SCH1D ; 2
15156	15033		SCH1C
15157	23214		RDT
15160	6		6
15161	1		1
15162	15		15

15163	15052	SCH1E ; 3
15164	15033	SCH1C
15165	22750	DWC
15166	16	16
15167	14	14
15170	122742	@ASWB
15171	15071	SCH2A ; 4
15172	15057	SCH2
15173	23117	WHD
15174	6	6
15175	1	1
15176	15	15
15177	15103	SCH2C ; 5
15200	15076	SCH2B
15201	22730	HWC
15202	16	16
15203	14	14
15204	122742	@ASWB
15205	15123	SCH3A ; 6
15206	15111	SCH3
15207	23522	RDFC
15210	6	6
15211	1	1
15212	15	15
15213	15130	SCH3B ; 7
15214	15111	SCH3
15215	23522	RDFC
15216	122750	@DWC
15217	123541	@NFCS
15220	123560	@FCS

;\*\*\*\* READ OFFSET \*\*\*\*

15221	6177	ROFS:	JSR	@SET
15222	24554		ROFST	
15223	15226		ROF	
15224	15433		ENDE	
15225	0		0	

;SEEK TO CYL 300 TO FORMAT 1 SECTOR

15226	6206	ROF:	JSR	@SKCMD
15227	400		SEEK	
15230	0		DRVO	
15231	300		300	
15232	50		50	
15233	6207		JSR	@EMCMD
15234	1400		FMAT	
15235	37		HO+S0+N1	
15236	30462		WBUFF	
15237	50		50	

;DONE, CHECK STATUS

15240	6202	ROF1:	JSR	@CKCS
15241	40000		RWDN	
15242	15245		ROF1A	; FINE
15243	6166		JSR	@ERROR ; NO GOOD
15244	15361		ROFEB	

;SET UP A DATA PATTERN AND WRITE IT OUT

15245	6176	ROF1A:	JSR	@PRIME
15246	30462		WBUFF	
15247	123556		123556	
15250	54321		54321	
15251	400		400	
15252	6207		JSR	@EMCMD
15253	3400		WRITE	
15254	37		HO+S0+N1	
15255	30462		WBUFF	
15256	50		50	

;WRITE DONE, CHECK STATUS

15257	6202	ROF1B:	JSR	@CKCS
15260	40000		RWDN	
15261	15264		ROF1C	; OK
15262	6166		JSR	@ERROR ; NG
15263	15367		ROFEB+ONE	

;READ DATA FROM SECTOR

15264	6207	ROF1C:	JSR	@EMCMD
15265	0		READ	
15266	37		HO+S0+N1	
15267	50462		RBUFF	
15270	50		50	

;READ DONE, CHECK STATUS FOR NO ECC ERROR



15271 6202 ROF1D: JSR @CKCS  
15272 40000 RWDN  
15273 15276 ROF2 ; GOOD STATUS  
15274 6166 JSR @ERROR ; BAD STATUS  
15275 15375 ROFEB+TWO

; READ DATA FROM DRIVE 1

15276 6206 ROF2: JSR @SKCMD  
15277 400 SEEK  
15300 40 DRV1  
15301 300 300  
15302 50 50  
15303 6207 JSR @EMCMD  
15304 40 READ+DRV1  
15305 37 HO+S0+N1  
15306 50462 RBUFF  
15307 50 50

; CHECK STATUS FOR ECC ERROR

15310 6202 ROF2A: JSR @CKCS  
15311 40201 RWDN+ECCE+RWFT  
15312 15315 ROF2B ; CORRECT STATUS  
15313 6166 JSR @ERROR ; INCORRECT STATUS  
15314 15403 ROFEB+THREE

; READ OFFSET+ FROM SAME TO GET CORRECT DATA

15315 6207 ROF2B: JSR @EMCMD  
15316 1040 OFSF+DRV1  
15317 37 HO+S0+N1  
15320 50462 RBUFF  
15321 50 50

; DONE, CHECK STATUS FOR NO ECC ERROR

15322 6202 ROF2C: JSR @CKCS  
15323 40000 RWDN  
15324 15327 ROF3 ; OK  
15325 6166 JSR @ERROR ; NOT SO  
15326 15411 ROFEB+FOUR

; READ DATA FROM SAME IN DRIVE 2

15327 6206 ROF3: JSR @SKCMD  
15330 400 SEEK  
15331 100 DRV2  
15332 300 300  
15333 50 50  
15334 6207 JSR @EMCMD  
15335 100 READ+DRV2  
15336 37 HO+S0+N1  
15337 50462 RBUFF  
15340 50 50

; READ DONE, CHECK STATUS FOR ECC ERROR

```

15341 6202 ROF3A: JSR @CKCS
15342 40201 RWDN+ECCE+RWFT
15343 15346 ROF3B ; EXACTLY
15344 6166 JSR @ERROR ; NOT SO
15345 15417 ROFEB+FIVE

```

; READ OFFSET- FROM SAME FOR DATA ERROR

```

15346 6207 ROF3B: JSR @EMCMD
15347 1300 OFSR+DRV2
15350 37 HO+SO+N1
15351 50462 RBUFF
15352 24 24

```

; DONE, CHECK STATUS FOR NO ECC ERROR

```

15353 6202 ROF3C: JSR @CKCS
15354 40000 RWDN
15355 15360 . +3
15356 6166 JSR @ERROR ; NOT THAT
15357 15425 ROFEB+SIX
15360 453 JMP ENDE ; END TEST

```

; \*\*\*\* ERROR MESSAGE BLOCKS \*\*\*\*

```

15361 15240 ROFEB: ROF1
15362 15226 ROF
15363 24570 FM1T
15364 6 6
15365 1 1
15366 15 15

15367 15257 ROF1B ; 1
15370 15252 ROF1A+5
15371 23206 WDT
15372 6 6
15373 1 1
15374 15 15

15375 15271 ROF1D ; 2
15376 15264 ROF1C
15377 23214 RDT
15400 6 6
15401 1 1
15402 15 15

15403 15310 ROF2A ; 3
15404 15276 ROF2
15405 23214 RDT
15406 6 6
15407 1 1
15410 15 15

15411 15322 ROF2C ; 4
15412 15315 ROF2B
15413 24142 OFFF
15414 6 6
15415 1 1
15416 15 15

```

```

15417 15341      ROF3A ; 5
15420 15327      ROF3
15421 23214      RDT
15422      6      6
15423      1      1
15424      15     15

```

```

15425 15353      ROF3C ; 6
15426 15346      ROF3B
15427 24142      OFFF
15430      6      6
15431      1      1
15432      15     15

```

; \*\*\*\* END OF EMULATOR TEST \*\*\*\*

```

15433 10146 ENDE: ISZ   PASS      ; COUNT PASSES
15434 63710      SKPDZ  TTI      ; QUIT?
15435   403      JMP    ENDE1   ; YES
15436  2401      JMP    @. +1    ; NO
15437 13000      RLDO

```

```

15440 24146 ENDE1: LDA   1, PASS
15441  6164      JSR   @DECN   ; PRINT PASS COUNT
15442  6155      JSR   @PRINT
15443 120010     @PASSE
15444  2111      JMP   @ABORT

```



;\*\*\*\* DISK READ-WRITE DRIVER \*\*\*\*

;CHECK VARIOUS CONTROLLER AND DRIVE STATUSES THEN LOADS COMMANDS  
; INTO THE CONTROLLER AND START.

; CALL: JSR @RW CMD  
; RW COMMAND  
; MEMORY ADDRESS  
; TIME COUNT  
; RET: JMP 3,3

15445	40000	RESV		
15446	1000	WPRT		
15447	10000	DRDY		
15450	0	0		
15451	54777	RWC:	STA	3, -1
15452	171000		MOV	3, 2 ; SAVE RETURN
15453	4267		JSR	SKBZ ; WAIT TILL IF BUSY
15454	777		JMP	-1
15455	4234		JSR	DDIA ; READ STATUS
15456	101112		MOVL#	0, 0, SZC ; CONTROL FULL?
15457	776		JMP	-2 ; YES, WAIT
15460	34103		LDA	3, HMSK
15461	163520		ANDZL	3, 0 ; GET CLEAR ATTEN BITS
15462	24121		LDA	1, DRVF ; GET DRIVE NUMBER
15463	123000		ADD	1, 0 ; ADD CLEAR DONE BIT
15464	25000		LDA	1, 0, 2 ; COMMAND
15465	123000		ADD	1, 0 ; INSERT

; CHECK IF NEED TO SET VOLUME SELECT BIT IN COMMAND

15466	34120		LDA	3, DSKT ; DRIVE TYPE
15467	175005		MOV	3, 3, SNR ; FIXED OR REMOVABLE DISKS?
15470	406		JMP	RWC1 ; REMOVABLE
15471	34126		LDA	3, SFCN ; GET SURF #
15472	175005		MOV	3, 3, SNR ; ZERO?
15473	403		JMP	RWC1 ; YES
15474	24457		LDA	1, VSEL ; NO, GET VS BIT
15475	123000		ADD	1, 0 ; INSERT
15476	4211	RWC1:	JSR	DDOA ; ISSUE COMMAND & SELECT DRV

; CHECK DRIVE STATUS BEFORE PROCEED

15477	4242		JSR	DDIB ; READ DRIVE STATUS
15500	24745		LDA	1, RWC-4
15501	107404		AND	0, 1, SZR ; DRIVE RESERVED?
15502	452		JMP	DRSV ; YES, EXIT
15503	24744		LDA	1, RWC-2
15504	107405		AND	0, 1, SNR ; DRIVE READY?
15505	6171		JSR	@NRDY ; NO, PRINT MESSAGE
15506	24740		LDA	1, RWC-3
15507	107404		AND	0, 1, SZR ; WRITE DISABLED?
15510	447		JMP	WDSB ; YES, EXIT
15511	101212		MOVR#	0, 0, SZC ; ANY OTHER FAULT?
15512	450		JMP	FLT ; YES

; GET DISK ADDRESSES FROM CONTROL BLOCK TO FORM DOC WORD

15513	24120		LDA	1, DSKT ; DRIVE TYPE
-------	-------	--	-----	----------------------

```

15514 20126 LDA 0,SFCN ;GET SURFACE #
15515 101300 MOVS 0,0 ;LEFT SWAP
15516 103125 ADDZL 0,0,SNR ;LEFT SHIFT 2 PLACES
15517 405 JMP RWC2 ;EXIT, SURFACE ZERO
15520 125005 MOV 1,1,SNR ;FIXED DSIC?
15521 403 JMP RWC2 ;NO
15522 24104 LDA 1,FIXED ;>0, GET OFFSET
15523 123000 ADD 1,0 ;ADJUST IT
15524 24130 RWC2: LDA 1,SCTN ;GET SECT #
15525 127120 ADDZL 1,1 ;LEFT SHIFT 5 PLACES
15526 127120 ADDZL 1,1
15527 125120 MOVZL 1,1
15530 123000 ADD 1,0
15531 24131 LDA 1,SCNT ;SECTOR COUNT
15532 124400 NEG 1,1
15533 34061 LDA 3,K37 ;MASK
15534 167400 AND 3,1 ;EXTRACT COUNT
15535 123000 ADD 1,0 ;COMPLETE WORD
15536 4225 JSR DDOC ;LOAD IT

```

; LOAD MEMORY ADDRESS THEN START. SET NORMAL STATUS AND TIMER

```

15537 21001 LDA 0,1,2
15540 4223 JSR DDOBS ;LOAD ADDR AND START
15541 25002 LDA 1,2,2 ;GET TIME COUNT
15542 44144 STA 1,CLOK ;SET TIMER
15543 6175 JSR @TIMER ;START TIMER

```

; RW DONE, READ AND SAVE STATUSES THEN RETURN

```

15544 4242 JSR DDIB
15545 40133 STA 0,DSTA ;SAVE DRIVE STATUS
15546 105000 MOV 0,1
15547 4234 JSR DDIA
15550 40132 STA 0,CSTA ;SAVE CONTROL STATUS
15551 34677 LDA 3,RWC-1
15552 1403 JMP 3,3 ;RETURN

```

15553 20 VSEL: 20

; PRINT DRIVE ERROR MESSAGE THEN ABORT OPERATION

```

15554 6155 DRSV: JSR @PRINT ;"DRIVE RESERVED"
15555 22565 DRST
15556 2111 JMP @ABORT

15557 6155 WDSB: JSR @PRINT
15560 22576 WDST
15561 2111 JMP @ABORT

15562 6155 FLT: JSR @PRINT ;PRINT DRIVE FAULT
15563 22613 DRFT
15564 105000 MOV 0,1
15565 6165 JSR @OCTN ;PRINT DRIVE STATUS WORD
15566 2111 JMP @ABORT

```

; \*\*\*\* DRIVE OPERATION COMMAND \*\*\*\*

; ISSUE DRIVE MOVEMENT COMMAND

; CALL: JSR @DRCMD  
; DRIVE COMMAND  
; TIME LIMIT IN MS  
; RET: JMP 2,3

15567	10000	DRDY		
15570	400	SEEK		
15571	4000	DBSY		
15572	0	0		
15573	54777	DRC:	STA	3, -1
15574	171000		MOV	3, 2 ; SAVE BLOCK POINTER
15575	4234		JSR	DDIA ; GET CONTROL STATUS
15576	101112		MOVL#	0, 0, SZC ; CONTROL FULL?
15577	776		JMP	-2 ; YES, LOOP
15600	4257		JSR	NNIOC ; CLEAR STATUS
15601	34103		LDA	3, HMSK
15602	163520		ANDZL	3, 0 ; WANT CLEAR DONE BITS
15603	25000		LDA	1, 0, 2 ; GET COMMAND
15604	123000		ADD	1, 0 ; CLR DONE+CMD
15605	24121		LDA	1, DRVF ; DRIVE NUMBER
15606	123000		ADD	1, 0
15607	34120		LDA	3, DSKT ; DRIVE TYPE
15610	175005		MOV	3, 3, SNR ; FIXED?
15611	406		JMP	DRCO ; NO
15612	34126		LDA	3, SFCN ; YES, GET SURFACE #
15613	175005		MOV	3, 3, SNR ; ZERO?
15614	403		JMP	DRCO ; YES
15615	24736		LDA	1, VSEL ; NO, GET VS BIT
15616	123000		ADD	1, 0
15617	4211	DRCO:	JSR	DDOA ; LOAD COMMAND

; CHECK DRIVE STATUS FIRST BEFORE START

15620	4242		JSR	DDIB ; GET DRIVE STATUS
15621	24746		LDA	1, DRC-4
15622	107415		AND#	0, 1, SNR ; DRIVE READY?
15623	6171		JSR	@NRDY ; NO, PRINT MESSAGE
15624	21000		LDA	0, 0, 2 ; COMMAND
15625	24743		LDA	1, DRC-3 ; GET REFERENCE
15626	106404		SUB	0, 1, SZR ; DO SEEK?
15627	434		JMP	DRC1B ; NO
15630	24741		LDA	1, DRC-2
15631	4242		JSR	DDIB ; YES, READ STATUS
15632	107414		AND#	0, 1, SZR ; DRIVE BUSY?
15633	776		JMP	-2 ; YES, WAIT
15634	20127		LDA	0, CYLN ; GET CYLINDER ADDRESS
15635	4232		JSR	DDOCP ; START SEEK
15636	24733		LDA	1, DRC-2
15637	34063		LDA	3, N17 ; LOOP COUNT
15640	61400	DRC1C:	DIB	0, 0 ; READ STATUS
15641	123414		AND#	1, 0, SZR ; DRV BUSY?
15642	405		JMP	DRC1-1 ; YES, EXIT
15643	175404		INC	3, 3, SZR ; NO, TIMEOUT?
15644	774		JMP	DRC1C ; NO
15645	10145		ISZ	TIMO ; YES, SET FLAG

```

-----
15646      405      JMP      DRC1A

; SAVE DRIVE STATUS AND START TIMER

15647      40416     STA      0,DSTAO
15650      25001 DRC1: LDA      1,1,2
15651      44144     STA      1,CLOK ; SET TIMER
15652      6175      JSR      @TIMER ; START TIMER

; DRIVE COMMAND DONE, CHECK STATUS AND CLEAR DONE

15653      4234 DRC1A: JSR      DDIA ; GET CONTROL STATUS
15654      101112     MOVL#   0,0,SZC ; CONTROL FULL?
15655      776       JMP      -2 ; YES, LOOP
15656      40132     STA      0,CSTA ; SAVE IT
15657      4242      JSR      DDIB ; GET DRIVE STATUS
15660      40133     STA      0,DSTA ; SAVE IT
15661      34711 DRC2: LDA      3,DRC-1
15662      1402      JMP      2,3 ; RETURN

15663      4255 DRC1B: JSR      NNIOF ; ISSUE PULSE
15664      752       JMP      DRC1C-2

15665      0 DSTAO: 0

```



; \*\*\*\* DRIVE READY WAIT \*\*\*\*

; PRINT DRIVE NOT READY MESSAGE THEN WAIT TILL READY OR  
; EMERGENCY EXIT  
; CALL: JSR @NRDY  
; RET: JMP 0,3 DRIVE READY  
; OR EXIT TO EXEC ON A KEY STROKE

15666	10000		DRDY		
15667	0		0		
15670	54777	NRD:	STA	3, -1	
15671	40133		STA	0, DSTA	; SAVE STATUS
15672	6155		JSR	@PRINT	; "DRIVE NOT READY"
15673	17752		NRDT		
15674	24772		LDA	1, NRD-2	
15675	4242	NRD1:	JSR	DDIB	; GET STATUS AGAIN
15676	107414		AND#	0, 1, SZR	; READY?
15677	2770		JMP	@NRD-1	; YES, RETURN
15700	63610		SKPDN	T11	; EXIT?
15701	774		JMP	NRD1	; NO
15702	2111		JMP	@ABORT	; YES

; \*\*\*\* CONTROLLER STATUS COMPARE \*\*\*\*

; COMPARE THE STATUS READ FROM CONTROLLER WITH THAT OF NORMALLY  
; EXPECTED  
; CALL: JSR @CKCK SEEKED, OR @CKCS NO SEEK  
; MNORMAL STATUS  
; NO ERROR VECTOR  
; RET: JMP @1,3 NO ERROR  
; JMP 2,3 ERROR

15703	21400	CKC:	LDA	0, 0, 3	; GET REFERENCE STATUS
15704	404		JMP	CKK+3	
15705	21400	CKK:	LDA	0, 0, 3	; GET NORMAL STATUS
15706	24134		LDA	1, ATTS	; DRV ATTEN BIT
15707	123000		ADD	1, 0	; INSERT
15710	40305		STA	0, DIAN	; SET FOR PRINTOUT
15711	24132		LDA	1, CSTA	; READ STATUS
15712	122415		SUB#	1, 0, SNR	; AGREE?
15713	3401		JMP	@1, 3	; YES
15714	1402		JMP	2, 3	; ERROR

; \*\*\*\* DRIVE STATUS COMPARE \*\*\*\*

; COMPARE DRIVE STATUS WITH WAH THAT WAS EXPECTED NORMALLY EXPECTED  
; CALL: JSR @CKDS  
; NORMAL STATUS  
; NO ERROR VECTOR  
; RET: JMP @1,3 NO ERROR  
; JMP 2,3 ERROR

15715	21400	CKD:	LDA	0, 0, 3	; NORMAL STATUS
15716	40306		STA	0, DIBN	; SET IT
15717	24133		LDA	1, DSTA	; STATUS READ
15720	106415		SUB#	0, 1, SNR	; OK?
15721	3401		JMP	@1, 3	; YES
15722	1402		JMP	2, 3	; NO

;\*\*\*\* COMPARE DISK ADDRESS \*\*\*\*

; CHECK THE INCREMENT OF DISK ADDRESS AFTER AN IO OPERATION  
; CALL: JSR @CKDA  
;       NORMAL ADDR  
;       NO ERROR VECTOR  
; RET:  JMP @1,3 NO ERROR  
;       JMP 2,3 ERROR

```
15723       0       0
15724  54777 CKAD: STA     3,.-1
15725    4250       JSR     DDIC     ; READ DISK ADDR REG
15726  34775       LDA     3,CKAD-1
15727  25400       LDA     1,0,3     ; GET NORMAL ADDR
15730  44307 CKAD1: STA     1,DICN   ; SET FOR PRINTOUT
15731 106415       SNE     0,1     ; AGREE?
15732    3401       JMP     @1,3     ; YES
15733  24071       LDA     1,CMAX
15734 107414       AND#    0,1,SZR   ; AT END OF TRACK?
15735    1402       JMP     2,3     ; NO, ERROR
15736  24402       LDA     1,AHD    ; YES, TRY NEXT HEAD#
15737     771       JMP     CKAD1
```

15740  2000 AHD:  2000

;\*\*\*\* PRESET DISK ADDRESS \*\*\*\*

; COPY THE CALLING DISK ADDRESS INTO THE RESPECTIVE LOCATIONS SO THEY  
; CAN BE PROCESSED BY THE DISK DRIVER  
; CALL: JSR       @DADD  
;       SURFACE NUMBER  
;       SECTOR NUMBER  
;       SECTOR COUNT  
; RET:  JMP 3,3

```
15741  25400 DAD:  LDA     1,0,3
15742  44126       STA     1,SFCN   ; SET SURF #
15743  25401       LDA     1,1,3
15744  44130       STA     1,SCTN   ; SET SECT #
15745  25402       LDA     1,2,3
15746 125112       MOVL#  1,1,SZC   ; FULL TRACK?
15747  24123       LDA     1,STRK   ; YES
15750  44131       STA     1,SCNT   ; SET SECT CNT
15751    1403       JMP     3,3
```

;\*\*\*\* PRINT BUFFER ADDRESS AND WORD \*\*\*\*

; CALL: JSR WBAD FOR WBUF DATA  
;       JSR RBER FOR RBUF DATA

```
15752       0       0
15753  54777 WBAD: STA     3,.-1
15754    6155       JSR     @PRINT   ; "ADDR IN WBUF"
15755 120027       @AWB
15756  24153       LDA     1,WBFP   ; WORD ADDR
15757    6165       JSR     @OCTN   ; PRINT IT
15760    6155       JSR     @PRINT   ; "DATA"
15761 120045       @DATA
```

```

----
15762 26153      LDA    1,@WBFP
15763  6165      JSR    @OCTN    ;PRINT DATA WORD
15764  2766      JMP    @WBAD-1

15765          0      0
15766 54777 RBER: STA    3,-1
15767  6155      JSR    @PRINT    ;"ADDR IN RBUFF"
15770 120036      @ARB
15771  24154      LDA    1,RBFP
15772  6165      JSR    @OCTN    ;PRINT ADDR
15773  6155      JSR    @PRINT
15774 120045      @DATA
15775  26154      LDA    1,@RBFP
15776  6165      JSR    @OCTN    ;PRINT DATA WORD
15777  2766      JMP    @RBER-1

```

;\*\*\*\* HEADER GENERATOR \*\*\*\*

; GENERATE THE SPECIFIED NUMBER OF HEADERS IN SEQUENTIAL ORDER IN WBUFF.

; HEADER FORMAT:

; WORD 0 CYLINDER ADDRESS  
; WORD 1 SURF + SECT + 0 (NO ALT SECT ADDR)  
; WORD 2 ZERO (NO ALT SURF + CYLINDER)

; CALL: JSR @HDRG

; HEADER COUNT

; RET: JMP 1,3

```
16000      0      0
16001 54777 HDG: STA 3,-1
16002 21400 LDA 0,0,3
16003 101112 MOVL# 0,0,SZC ; ONE TRACK?
16004 20123 LDA 0,STRK ; YES
16005 40500 STA 0,HDC ; SET COUNT
16006 24126 LDA 1,SFCN
16007 44477 STA 1,HEAD ; SET START SURF NUMBER
16010 24130 LDA 1,SCTN
16011 44476 STA 1,SECT ; SET START SECT NUMBER
16012 20127 LDA 0,CYLN
16013 40475 STA 0,CYLD ; SET START CYL NUMBER
16014 24123 LDA 1,STRK
16015 124400 NEG 1,1
16016 124000 COM 1,1
16017 44472 STA 1,LSECT ; SET LAST SECT NUMBER
16020 34115 LDA 3,WBUF
16021 54463 STA 3,WBP ; SET BUFF POINTER
```

; GENERATE HEADERS TO WBUFF

```
16022 20465 LDA 0,SECT ; SECT NUMBER
16023 103120 ADDZL 0,0 ; SHIFT INTO POSITION
16024 103120 ADDZL 0,0
16025 101100 MOVL 0,0
16026 24460 HDG1: LDA 1,HEAD ; SURF NUMBER
16027 125300 MOVS 1,1 ; SHIFT IT INTO POSITION
16030 127125 ADDZL 1,1,SNR ; HEAD 0?
16031 406 JMP HDG1A ; YES
16032 34120 LDA 3,DSKT
16033 175005 MOV 3,3,SNR ; NO, FIXED DISK?
16034 403 JMP HDG1A ; NO
16035 34104 LDA 3,FIXED
16036 167000 ADD 3,1 ; YES, ADD OFFSET
16037 107000 HDG1A: ADD 0,1 ; SURF + SECT
16040 34450 LDA 3,CYLD ; CYL ADDR
16041 56443 STA 3,@WBP ; SET WORD 0
16042 10442 ISZ WBP
16043 46441 STA 1,@WBP ; SET WORD 1
16044 10440 ISZ WBP
16045 176400 SUB 3,3
16046 56436 STA 3,@WBP ; SET WORD 2
16047 10435 ISZ WBP
16050 14435 DSZ HDC ; MORE?
16051 403 JMP HDG2 ; YES
16052 34726 HDG1B: LDA 3,HDG-1
16053 1401 JMP 1,3 ; NO, RETURN
```

; INCREMENT SECTOR ADDRESS IF NOT AT LAST SECTOR

```
16054 20433 HDG2: LDA 0, SECT
16055 34434 LDA 3, LSECT
16056 116415 SUB# 0, 3, SNR ; AT LAST SECT?
16057 403 JMP HDG3 ; YES
16060 20432 LDA 0, OSECT ; NO, NEXT ONE
16061 756 JMP HDG1A
```

; INCREMENT CYLINDER ADDRESS IF NOT AT THE LAST ONE

```
16062 24426 HDG3: LDA 1, CYLD
16063 34124 LDA 3, LCYN
16064 136415 SUB# 1, 3, SNR ; AT LAST CYL?
16065 405 JMP HDG4 ; YES
16066 10422 ISZ CYLD ; NO, BUMP IT
16067 102400 SUB 0, 0
16070 40417 STA 0, SECT ; RESET SECT NUMBER
16071 735 JMP HDG1
```

; INCREMENT SURFACE NUMBER IF NOT AT THE LAST ONE ALREADY

```
16072 24414 HDG4: LDA 1, HEAD
16073 34117 LDA 3, DTBL ; DRIVE TABLE
16074 35401 LDA 3, 1, 3 ; GET LAST SURF NUMBER
16075 136415 SUB# 1, 3, SNR ; ALL DONE?
16076 754 JMP HDG1B ; YES, RETURN
16077 10407 ISZ HEAD ; NO, NEXT SURF
16100 102400 SUB 0, 0
16101 40406 STA 0, SECT ; RESET SECT NUMBER
16102 40406 STA 0, CYLD ; RESET CYL NUMBER
16103 723 JMP HDG1 ; CONTINUE
```

```
16104 0 WBP: 0
16105 0 HDC: 0
16106 0 HEAD: 0
16107 0 SECT: 0
16110 0 CYLD: 0
16111 0 LSECT: 0
16112 40 OSECT: 40
```

; \*\*\*\* HEADER COMPARE \*\*\*\*

```
; COMPARE THE SPECIFIED NUMBER OF HEADERS IN RBUF VS THOSE IN WBUF
; CALL: JSR @HCMP
; HEADER COUNT
; NO ERROR VECTOR
; RET: JMP @1, 3 NO ERROR
; JMP 2, 3 ERROR
```

```
16113 0 HCM: 0
16114 54777 HCM: STA 3, -1
16115 25400 LDA 1, 0, 3
16116 125112 MOVL# 1, 1, SZC ; FULL TRACK?
16117 24123 LDA 1, STRK ; YES
16120 44433 STA 1, HCC ; SET COMPARE COUNT
16121 30116 LDA 2, RBUF ; READ BUFF
16122 34115 LDA 3, WBUF ; WRITE BUFF
```

```

16123 21000 HCM1: LDA 0,0,2 ; READ WORD
16124 25400 LDA 1,0,3 ; WRITE WORD
16125 106414 SUB# 0,1,SZR ; WORD 0 AGREE?
16126 421 JMP HCM2 ; NO, ERROR
16127 21001 LDA 0,1,2
16130 25401 LDA 1,1,3
16131 106414 SUB# 0,1,SZR ; WORD 1 OK?
16132 415 JMP HCM2 ; NO
16133 21002 LDA 0,2,2
16134 25402 LDA 1,2,3
16135 106414 SUB# 0,1,SZR ; WORD 2 OK?
16136 411 JMP HCM2 ; NO
16137 24051 LDA 1,K3
16140 133000 ADD 1,2 ; NEXT SET POINTER
16141 133000 ADD 1,2
16142 137000 ADD 1,3
16143 14410 DSZ HCC ; MORE?
16144 757 JMP HCM1 ; YES
16145 34746 LDA 3,HCM-1
16146 3401 JMP @1,3 ; NO, RETURN

```

; COMPARE ERROR, SET WBUF AND RBUF ADDRESSES THEN RETURN

```

16147 50154 HCM2: STA 2,RBFP
16150 54153 STA 3,WBFP
16151 34742 LDA 3,HCM-1
16152 1402 JMP 2,3

```

```

16153 0 HCC: 0

```

;\*\*\*\* BUFFER PRIMER \*\*\*\*

; FILL THE BUFFER WITH THE SPECIFIED NUMBER OF DATA WORDS  
; CALL: JSR @PRIME  
;        BUFFER NAME  
;        DATA WORD 1  
;        DATA WORD 2  
;        WORD COUNT

16154	0	0			
16155	54777	PRIM:	STA	3, -1	
16156	31400		LDA	2, 0, 3	; BUFF POINTER
16157	21401		LDA	0, 1, 3	; WORD 1
16160	25402		LDA	1, 2, 3	; WORD 2
16161	35403		LDA	3, 3, 3	; WORD COUNT
16162	174400		NEG	3, 3	
16163	41000	PRIM1:	STA	0, 0, 2	; SET WORD
16164	175405		INC	3, 3, SNR	; DONE?
16165	406		JMP	PRIM2	; YES
16166	151400		INC	2, 2	
16167	45000		STA	1, 0, 2	; SET WORD 2
16170	151400		INC	2, 2	
16171	175404		INC	3, 3, SZR	; DONE?
16172	771		JMP	PRIM1	; NOT YET
16173	34761	PRIM2:	LDA	3, PRIM-1	; YES, RETURN
16174	1404		JMP	4, 3	

;\*\*\*\* DATA WORD COMPARE \*\*\*\*

; COMPARE DATA WORD FOR WORD RBUF VS WBUF  
; CALL: JSR @DCMP  
;        WORD COUNT  
;        NO ERROR VECTOR  
; RET:  JMP @1, 3 NO ERROR  
;        JMP 2, 3 ERROR

16175	24116	DCM:	LDA	1, RBUF	
16176	44154		STA	1, RBFP	; SET READ POINTER
16177	24115		LDA	1, WBUF	
16200	44153		STA	1, WBFP	; SET WRITE POINTER
16201	21400		LDA	0, 0, 3	
16202	40412		STA	0, DCMC	; SET COMPARE COUNT
16203	22154	DCM1:	LDA	0, @RBFP	; GET READ WORD
16204	26153		LDA	1, @WBFP	; GET WRITE WORD
16205	106414		SUB#	0, 1, SZR	; DO THEY MATCH?
16206	1402		JMP	2, 3	; NO, RETURN
16207	10154		ISZ	RBFP	
16210	10153		ISZ	WBFP	
16211	14403		DSZ	DCMC	; YES, MORE?
16212	771		JMP	DCM1	; YES
16213	3401		JMP	@1, 3	; NO
16214	0	DCMC:	0		

;\*\*\*\* EMULATOR SEEK COMMAND SUBROUTINE \*\*\*\*

; CALL: JSR @SJCMD  
;       COMMAND  
;       DRV# + VOL#  
;       CYL#  
;       TIME COUNT  
; RET:  JMP     4,3

16215	400	SEEK		
16216	0	0		
16217	54777	SCMD:	STA	3, -1
16220	4234		JSR	DDIA ; GET CONTROL STATUS
16221	34103		LDA	3, HMSK
16222	163520		ANDZL	3, 0 ; WANT CLR ATT BITS
16223	34773		LDA	3, SCMD-1
16224	25403		LDA	1, 3, 3 ; GET TIME COUNT
16225	44144		STA	1, CLOK ; SET TIMER
16226	25400		LDA	1, 0, 3
16227	123000		ADD	1, 0 ; INSERT CMD
16230	35401		LDA	3, 1, 3
16231	163000		ADD	3, 0 ; ADD DRV SEL
16232	4211		JSR	DDDA ; LOAD IT
16233	20762		LDA	0, SCMD-2
16234	106415		SUB#	0, 1, SNR ; SEEK?
16235	404		JMP	SCM1 ; YES
16236	4255		JSR	NNIOP ; NO, START IT
16237	411		JMP	SCM1A

16240	4000	DBSY		
16241	24777	SCM1:	LDA	1, -1
16242	4242		JSR	DDIB ; READ DRIVE STATUS
16243	107414		AND#	0, 1, SZR ; DRV BUSY?
16244	776		JMP	-2 ; YES, WAIT
16245	34751		LDA	3, SCMD-1
16246	21402		LDA	0, 2, 3 ; GET CYL#
16247	4232		JSR	DDOCP ; LOAD CYL# & START
16250	6175	SCM1A:	JSR	@TIMER ; START IT

; DONE, READ CONTROL & DRIVE STATUSES

16251	4242		JSR	DDIB ; GET DRIVE STAUS
16252	105000		MOV	0, 1
16253	44133		STA	1, DSTA ; SAVE IT
16254	4234		JSR	DDIA ; GET CONTROL STATUS
16255	40132		STA	0, CSTA ; SAVE IT ALSO
16256	34740		LDA	3, SCMD-1
16257	1404		JMP	4, 3

;\*\*\*\* EMULATOR IO COMMAND SUBROUTINE \*\*\*\*

; CALL: JSR @EMCMD  
;       COMMAND + DRV#  
;       H + S + N  
;       MEMORY ADDRESS  
;       TIME COUNT  
; RET:  JMP 4,3

16260       0       0



```

16261      0      0
16262  54777 ECMD: STA      3, -1
16263  50775      STA      2, ECMD-2
16264 171000      MOV      3, 2
16265   4234      JSR      DDIA      ; GET STATUS
16266  34103      LDA      3, HMSK
16267 163520      ANDZL   3, 0      ; WANT CLR ATT BITS
16270  25000      LDA      1, 0, 2
16271 123000      ADD      1, 0      ; CLR DONE + CMD + DRV#
16272   4211      JSR      DDOA      ; LOAD IT
16273  21001      LDA      0, 1, 2
16274   4225      JSR      DDOC      ; LOAD DISK ADDR
16275  21002      LDA      0, 2, 2
16276   4223      JSR      DDOBS     ; LOAD MEM ADDR & START
16277  25003      LDA      1, 3, 2
16300  44144      STA      1, CLOK     ; SET TIMER
16301  30757      LDA      2, ECMD-2 ; RESTORE AC2
16302   6175      JSR      @TIMER     ; START IT

```

; DONE, READ CONTROL AND DRIVE STATUSES

```

16303   4242 ECM: JSR      DDIB
16304 105000      MOV      0, 1
16305  44133      STA      1, DSTA     ; SAVE DRIVE STATUS
16306   4234      JSR      DDIA
16307  40132      STA      0, CSTA     ; SAVE CONTROL STATUS
16310  34405      LDA      3, ATM
16311 117400      AND      0, 3      ; WANT ATTEN BIT
16312  54134      STA      3, ATTS     ; SET IT
16313  34746      LDA      3, ECMD-1
16314   1404      JMP      4, 3

```

16315 36000 ATM: 36000

; \*\*\*\* RBUF VS REFERENCE COMPARISON \*\*\*\*

```

; CALL: JSR @CRBUF
;       REFERENCE BLOCK
;       WORD COUNT
;       NO ERROR RETURN VECTOR
; RET:  JMP @2, 3 NO ERROR
;       JMP 3, 3 ERROR

```

```

16316      0      0
16317  54777 CRBF: STA      3, -1
16320  25400      LDA      1, 0, 3
16321  44420      STA      1, REFP     ; SET REF POINTER
16322  21401      LDA      0, 1, 3
16323  40152      STA      0, CNTR     ; SET COMPARE COUNT
16324  24116      LDA      1, RBUF
16325  44154      STA      1, RBFP     ; SET RBUF POINTER
16326  22413 CRBF1: LDA      0, @REFP ; GET REF WORD
16327  26154      LDA      1, @RBFP ; GET RBUF WORD
16330 106414      SUB#     0, 1, SZR     ; OK?
16331   406      JMP      CRBF2     ; ERROR
16332  10407      ISZ     REFP     ; BUMP POINTERS
16333  10154      ISZ     RBFP
16334  14152      DSZ     CNTR     ; DONE?
16335   771      JMP      CRBF1     ; NOT YET

```

-----  
16336 3402 JMP @2,3 ; YES  
16337 40141 CRBF2: STA 0, ACN ; SAVE REF WORD  
16340 1403 JMP 3,3  
16341 0 REFP: 0

DISK DRIVE DIRECTORY

```
; * * * * *
; *
; *          DDIR          *
; *
; * * * * *
```

```
; THIS MODULE CONTAINS:
; DRIVE DIRECTORY
; DRIVE LOOKUP TABLES
; DISPLAY DIRECTORY ROUTINE
```

;\*\*\*\* DISK DRIVE DIRECTORY \*\*\*\*

; EACH DIRECTORY ENTRY CONSISTS OF 3 WORDS AS FOLLOWS:  
; CHARACTER COUNT OF DRIVE MNEMONIC, -1 = NO ENTRY  
; TEXT ADDRESS OF DRIVE MNEMONIC  
; LOOKUP TABLE VECTOR

; MANUFACTURER MNEMONICS:

; CDC CONTROL DATA CORP  
; AMP AMPEX  
; MMX MEMOREX  
; FUJ FUJITSU  
; OKI OKIDATA  
; CAL CALCOMP  
; MCD MICRODATA  
; KEN KENNEDY  
; CDS CENTURY DATA SYSTEMS

16342	5	DDIR: 5	; CDC32
16343	25163	DN1	
16344	16720	DT1	
16345	4	4	
16346	25166	DN2	
16347	16724	DT2	
16350	5	5	; CDC64
16351	25171	DN3	
16352	16730	DT3	
16353	5	5	; CDC96
16354	25174	DN4	
16355	16734	DT4	
16356	5	5	; CDC40
16357	25177	DN5	
16360	16740	DT5	
16361	5	5	; CDC80
16362	25202	DN6	
16363	16744	DT6	
16364	6	6	; CDC150
16365	25205	DN7	
16366	16750	DT7	
16367	6	6	; CDC300
16370	25211	DN8	
16371	16754	DT8	
16372	5	5	; CDC12
16373	25215	DN9	
16374	16760	DT9	
16375	5	5	; CDC25
16376	25220	DN10	
16377	16764	DT10	
16400	5	5	; CDC82
16401	25223	DN11	
16402	16770	DT11	
16403	5	5	; CDC. 9
16404	25226	DN12	
16405	16774	DT12	
16406	4	4	; CDC1
16407	25231	DN13	
16410	17000	DT13	
16411	4	4	; CDC2
16412	25234	DN14	

16413	17004	DT14	
16414	5	5	; CDC16
16415	25237	DN15	
16416	17010	DT15	
16417	6	6	; CDC16F
16420	25242	DN16	
16421	17014	DT16	
16422	6	6	; CDC48F
16423	25246	DN17	
16424	17020	DT17	
16425	6	6	; CDC80F
16426	25252	DN18	
16427	17024	DT18	
16430	177777	-1	
16431	25256	DN19	
16432	17030	DT19	
16433	177777	-1	
16434	25257	DN20	
16435	17034	DT20	
16436	177777	-1	
16437	25260	DN21	
16440	17040	DT21	
16441	177777	-1	
16442	25261	DN22	
16443	17044	DT22	
16444	177777	-1	
16445	25262	DN23	
16446	17050	DT23	
16447	177777	-1	
16450	25263	DN24	
16451	17054	DT24	
16452	5	5	; AMP40
16453	25264	DN25	
16454	17060	DT25	
16455	5	5	; AMP80
16456	25267	DN26	
16457	17064	DT26	
16460	6	6	; AMP100
16461	25272	DN27	
16462	17070	DT27	
16463	6	6	; AMP150
16464	25276	DN28	
16465	17074	DT28	
16466	6	6	; AMP160
16467	25302	DN29	
16470	17100	DT29	
16471	6	6	; AMP200
16472	25306	DN30	
16473	17104	DT30	
16474	6	6	; AMP300
16475	25312	DN31	
16476	17110	DT31	
16477	5	5	; AMP16
16500	25316	DN32	
16501	17114	DT32	
16502	6	6	; AMP16F
16503	25321	DN33	
16504	17120	DT33	
16505	5	5	; AMP48

16506	25325	DN34	
16507	17124	DT34	
16510	6	6	; AMP80C
16511	25330	DN35	
16512	17130	DT35	
16513	5	5	
16514	25334	DN36	
16515	17134	DT36	
16516	177777	-1	
16517	25335	DN37	
16520	17140	DT37	
16521	177777	-1	
16522	25336	DN38	
16523	17144	DT38	
16524	177777	-1	
16525	25337	DN39	
16526	17150	DT39	
16527	5	5	; MMX25
16530	25340	DN40	
16531	17154	DT40	
16532	5	5	; MMX50
16533	25343	DN41	
16534	17160	DT41	
16535	5	5	; MMX75
16536	25346	DN42	
16537	17164	DT42	
16540	6	6	; MMX100
16541	25351	DN43	
16542	17170	DT43	
16543	6	6	; MMX200
16544	25355	DN44	
16545	17174	DT44	
16546	177777	-1	
16547	25361	DN45	
16550	17200	DT45	
16551	177777	-1	
16552	25362	DN46	
16553	17204	DT46	
16554	177777	-1	
16555	25363	DN47	
16556	17210	DT47	
16557	5	5	; FUJ40
16560	25364	DN48	
16561	17214	DT48	
16562	177777	-1	
16563	25367	DN49	
16564	17220	DT49	
16565	5	5	; OKI12
16566	25370	DN50	
16567	17224	DT50	
16570	5	5	; OKI24
16571	25373	DN51	
16572	17230	DT51	
16573	5	5	; OKI37
16574	25376	DN52	
16575	17234	DT52	
16576	5	5	; OKI49
16577	25401	DN53	
16600	17240	DT53	

16601	5	5	; OKI61
16602	25404	DN54	
16603	17244	DT54	
16604	5	5	; OKI80
16605	25407	DN55	
16606	17250	DT55	
16607	177777	-1	
16610	25412	DN56	
16611	17254	DT56	
16612	177777	-1	
16613	25413	DN57	
16614	17260	DT57	
16615	177777	-1	
16616	25414	DN58	
16617	17264	DT58	
16620	177777	-1	
16621	25415	DN59	
16622	17270	DT59	
16623	5	5	; CAL25
16624	25416	DN60	
16625	17274	DT60	
16626	5	5	; CAL50
16627	25421	DN61	
16630	17300	DT61	
16631	5	5	; CAL80
16632	25424	DN62	
16633	17304	DT62	
16634	6	6	; CAL200
16635	25427	DN63	
16636	17310	DT63	
16637	6	6	; CAL300
16640	25433	DN64	
16641	17314	DT64	
16642	177777	-1	
16643	25437	DN65	
16644	17320	DT65	
16645	177777	-1	
16646	25440	DN66	
16647	17324	DT66	
16650	6	6	
16651	25441	DN67	
16652	17330	DT67	
16653	177777	-1	
16654	25442	DN68	
16655	17334	DT68	
16656	177777	-1	
16657	25443	DN69	
16660	17340	DT69	
16661	5	5	; MCD12
16662	25444	DN70	
16663	17344	DT70	
16664	5	5	; MCD37
16665	25447	DN71	
16666	17350	DT71	
16667	5	5	; MCD62
16670	25452	DN72	
16671	17354	DT72	
16672	177777	-1	
16673	25455	DN73	

-----  
16674 17360 DT73  
16675 177777 -1  
16676 25456 DN74  
16677 17364 DT74  
16700 5 5 ; KEN14  
16701 25457 DN75  
16702 17370 DT75  
16703 5 5 ; KEN42  
16704 25462 DN76  
16705 17374 DT76  
16706 5 5 ; KEN70  
16707 25465 DN77  
16710 17400 DT77  
16711 177777 -1  
16712 25470 DN78  
16713 17404 DT78  
16714 177777 -1  
16715 25471 DN79  
16716 17410 DT79  
16717 0 0 ; END



;\*\*\*\* DRIVE LOOKUP TABLES \*\*\*\*

;TABLE WORD ENTRIES:

; 0 DISC TYPE: 0 = REMOVABLE, -1 = FIXED  
; 1 LAST PHYSICAL WR HEAD NUMBER  
; 2 NUMBER OF SECTORS PER TRACK  
; 3 LAST CYLINDER NUMBER

16720	177777	DT1:	-1	; CDC32
16721	1		1	
16722	40		40	
16723	1466		1466	
16724	0	DT2:	0	; CDC LARK
16725	3		3	
16726	40		40	
16727	315		315	
16730	177777	DT3:	-1	; CDC64
16731	3		3	
16732	40		40	
16733	1466		1466	
16734	177777	DT4:	-1	; CDC96
16735	5		5	
16736	40		40	
16737	1466		1466	
16740	0	DT5:	0	; CDC40
16741	4		4	
16742	40		40	
16743	632		632	
16744	0	DT6:	0	; CDC80
16745	4		4	
16746	40		40	
16747	1466		1466	
16750	0	DT7:	0	; CDC150
16751	22		22	
16752	40		40	
16753	632		632	
16754	0	DT8:	0	; CDC300
16755	22		22	
16756	40		40	
16757	1466		1466	
16760	0	DT9:	0	; CDC12
16761	1		1	
16762	40		40	
16763	477		477	
16764	0	DT10:	0	; CDC25
16765	3		3	
16766	40		40	
16767	477		477	
16770	0	DT11:	0	; CDC12
16771	4		4	
16772	40		40	
16773	1466		1466	
16774	0	DT12:	0	; CDC. 9
16775	3		3	
16776	40		40	
16777	13		13	
17000	0	DT13:	0	; CDC1
17001	4		4	
17002	40		40	

17003	12	12	
17004	0 DT14:	0	; CDC2
17005	4	4	
17006	40	40	
17007	24	24	
17010	177777 DT15:	-1	; CDC16
17011	0	0	
17012	40	40	
17013	1447	1447	
17014	177777 DT16:	-1	; CDC16F
17015	0	0	
17016	40	40	
17017	1477	1477	
17020	177777 DT17:	-1	; CDC48F
17021	2	2	
17022	40	40	
17023	1477	1477	
17024	177777 DT18:	-1	; CDC80F
17025	4	4	
17026	40	40	
17027	1477	1477	
17030	0 DT19:	0	
17031	0	0	
17032	0	0	
17033	0	0	
17034	0 DT20:	0	
17035	0	0	
17036	0	0	
17037	0	0	
17040	0 DT21:	0	
17041	0	0	
17042	0	0	
17043	0	0	
17044	0 DT22:	0	
17045	0	0	
17046	0	0	
17047	0	0	
17050	0 DT23:	0	
17051	0	0	
17052	0	0	
17053	0	0	
17054	0 DT24:	0	
17055	0	0	
17056	0	0	
17057	0	0	
17060	0 DT25:	0	; AMP40
17061	4	4	
17062	40	40	
17063	632	632	
17064	0 DT26:	0	; AMP80
17065	4	4	
17066	40	40	
17067	1466	1466	
17070	0 DT27:	0	; AMP100
17071	22	22	
17072	25	25	
17073	632	632	
17074	0 DT28:	0	; AMP150
17075	15	15	

17076	40	40	
17077	1060	1060	
17100	0	DT29: 0	; AMP160
17101	4	4	
17102	40	40	
17103	3154	3154	
17104	0	DT30: 0	; AMP200
17105	22	22	
17106	25	25	
17107	1456	1456	
17110	0	DT31: 0	; AMP300
17111	22	22	
17112	40	40	
17113	1456	1456	
17114	0	DT32: 0	; AMP16
17115	0	0	
17116	40	40	
17117	1466	1466	
17120	177777	DT33: -1	; AMP16F
17121	0	0	
17122	40	40	
17123	1466	1466	
17124	177777	DT34: -1	; AMP48
17125	2	2	
17126	40	40	
17127	1466	1466	
17130	177777	DT35: -1	; AMP80C
17131	4	4	
17132	40	40	
17133	1466	1466	
17134	177777	DT36: -1	; AMP96
17135	5	5	
17136	40	40	
17137	1466	1466	
17140	0	DT37: 0	
17141	0	0	
17142	0	0	
17143	0	0	
17144	0	DT38: 0	
17145	0	0	
17146	0	0	
17147	0	0	
17150	0	DT39: 0	
17151	0	0	
17152	0	0	
17153	0	0	
17154	0	DT40: 0	; MMX25
17155	3	3	
17156	34	34	
17157	535	535	
17160	0	DT41: 0	; MMX50
17161	7	7	
17162	34	34	
17163	535	535	
17164	0	DT42: 0	; MMX75
17165	13	13	
17166	34	34	
17167	535	535	
17170	0	DT43: 0	; MMX100

17171	22	22	
17172	25	25	
17173	632	632	
17174	0 DT44:	0	; MMX200
17175	22	22	
17176	25	25	
17177	1456	1456	
17200	0 DT45:	0	
17201	0	0	
17202	0	0	
17203	0	0	
17204	0 DT46:	0	
17205	0	0	
17206	0	0	
17207	0	0	
17210	0 DT47:	0	
17211	0	0	
17212	0	0	
17213	0	0	
17214	0 DT48:	0	; FUJ40
17215	2	2	
17216	40	40	
17217	1456	1456	
17220	0 DT49:	0	
17221	0	0	
17222	0	0	
17223	0	0	
17224	0 DT50:	0	; OKI113
17225	1	1	
17226	40	40	
17227	522	522	
17230	0 DT51:	0	; OKI126
17231	3	3	
17232	40	40	
17233	522	522	
17234	0 DT52:	0	; OKI140
17235	5	5	
17236	40	40	
17237	522	522	
17240	0 DT53:	0	; OKI153
17241	7	7	
17242	40	40	
17243	522	522	
17244	0 DT54:	0	; OKI167
17245	11	11	
17246	40	40	
17247	522	522	
17250	0 DT55:	0	; OKI180
17251	13	13	
17252	40	40	
17253	522	522	
17254	0 DT56:	0	
17255	0	0	
17256	0	0	
17257	0	0	
17260	0 DT57:	0	
17261	0	0	
17262	0	0	
17263	0	0	

17264	0	DT58:	0	
17265	0		0	
17266	0		0	
17267	0		0	
17270	0	DT59:	0	
17271	0		0	
17272	0		0	
17273	0		0	
17274	0	DT60:	0	; CAL25
17275	4		4	
17276	32		32	
17277	626		626	
17300	0	DT61:	0	; CAL50
17301	4		4	
17302	32		32	
17303	1456		1456	
17304	0	DT62:	0	; CAL80
17305	4		4	
17306	40		40	
17307	1456		1456	
17310	0	DT63:	0	; CAL200
17311	22		22	
17312	40		40	
17313	1456		1456	
17314	0	DT64:	0	; CAL300
17315	22		22	
17316	40		40	
17317	1456		1456	
17320	0	DT65:	0	
17321	0		0	
17322	0		0	
17323	0		0	
17324	0	DT66:	0	
17325	0		0	
17326	0		0	
17327	0		0	
17330	0	DT67:	0	; BALL80
17331	4		4	
17332	40		40	
17333	1456		1456	
17334	0	DT68:	0	
17335	0		0	
17336	0		0	
17337	0		0	
17340	0	DT69:	0	
17341	0		0	
17342	0		0	
17343	0		0	
17344	0	DT70:	0	; MCD12
17345	1		1	
17346	34		34	
17347	535		535	
17350	0	DT71:	0	; MCD37
17351	5		5	
17352	34		34	
17353	535		535	
17354	0	DT72:	0	; MCD62
17355	11		11	
17356	34		34	

17357	535	535	
17360	0 DT73:	0	
17361	0	0	
17362	0	0	
17363	0	0	
17364	0 DT74:	0	
17365	0	0	
17366	0	0	
17367	0	0	
17370	0 DT75:	0	; KEN14
17371	0	0	
17372	40	40	
17373	1273	1273	
17374	0 DT76:	0	; KEN42
17375	2	2	
17376	40	40	
17377	1273	1273	
17400	0 DT77:	0	; KEN70
17401	4	4	
17402	40	40	
17403	1273	1273	
17404	0 DT78:	0	
17405	0	0	
17406	0	0	
17407	0	0	
17410	0 DT79:	0	
17411	0	0	
17412	0	0	
17413	0	0	

;\*\*\*\* DIRECTORY DISPLAY SUBROUTINE \*\*\*\*

; DISPLAY DRIVE CHARACTERISTICS ON CONSOLE ONE DRIVE PER LINE  
; HIT ANY KEY EXCEPT CR TO STOP OR RESUME DISPLAY. QUIT DISPLAY ON CR

```
17414      0      0
17415  54777 DSPL: STA      3,.-1
17416  20433      LDA      0,DNTL
17417  40431      STA      0,DNTP ; SET TEXT LIST POINTER
17420  26430 DSP1: LDA      1,@DNTP ; GET TEXT ADDR
17421 125113      MOVL#    1,1,SNR ; DUMMY ENTRY?
17422    403      JMP      DSP1A ; YES, SKIP IT
17423  10425      ISZ      DNTP ; BUMP POINTER
17424    774      JMP      DSP1

17425 125005 DSP1A: MOV     1,1,SNR ; END OF LIST?
17426    416      JMP      DSP3 ; YES, RETURN
17427  44402      STA      1,..+2 ; NO, SET IT
17430    6155      JSR      @PRINT ; PRINT A LINE
17431      0      0
17432  63610      SKPDN   TTI ; STOP?
17433    407      JMP      DSP2 ; NO
17434  60510      DIAS    0,TTI ; YES, READ INPUT
17435  24412      LDA      1,KCR
17436 106405      SUB     0,1,SNR ; QUIT?
17437    405      JMP      DSP3 ; YES
17440  63610      SKPDN   TTI ; WAIT FOR RESTART
17441    777      JMP      .-1
17442  10406 DSP2: ISZ      DNTP ; BUMP POINTER
17443    755      JMP      DSP1

17444    6155 DSP3: JSR      @PRINT ; SPACE A LINE
17445  20076      CRLF
17446    2746      JMP      @DSPL-1

17447    215 KCR:  215
17450      0 DNTP: 0
```

; DRIVE CHARACTERISTICS DISPLAY LIST

```
17451  17452 DNTL:  .+1
17452  25472      CDC
17453  25612      HLN1
17454  25651      DTX1
U 17455  17455      DTX2
17456  25707      DTX3
17457  25745      DTX4
17460  26003      DTX5
17461  26035      DTX6
17462  26073      DTX7
17463  26125      DTX8
17464  26163      DTX9
17465  26215      DTX10
17466  26247      DTX11
17467  26301      DTX12
17470  26333      DTX13
17471  26365      DTX14
17472  26417      DTX15
17473  26451      DTX16
```

17474	26503	DTX17
17475	26535	DTX18
17476	177777	-1
17477	177777	-1
17500	177777	-1
17501	177777	-1
17502	177777	-1
17503	177777	-1
17504	25500	AMP
17505	25612	HLN1
17506	26567	DTX25
17507	26621	DTX26
17510	26660	DTX27
17511	26712	DTX28
17512	26744	DTX29
17513	27003	DTX30
17514	27035	DTX31
17515	27074	DTX32
17516	27126	DTX33
17517	27160	DTX34
17520	27212	DTX35
U 17521	17521	DTX36
17522	177777	-1
17523	177777	-1
17524	177777	-1
17525	25507	MMX
17526	25612	HLN1
17527	27244	DTX40
17530	27301	DTX41
17531	27336	DTX42
17532	27373	DTX43
17533	27430	DTX44
17534	177777	-1
17535	177777	-1
17536	177777	-1
17537	25527	FUJ
17540	25612	HLN1
17541	27465	DTX48
17542	177777	-1
17543	25517	OKI
17544	25612	HLN1
17545	27517	DTX50
17546	27555	DTX51
17547	27613	DTX52
17550	27651	DTX53
17551	27707	DTX54
17552	27745	DTX55
17553	177777	-1
17554	177777	-1
17555	177777	-1
17556	177777	-1
17557	25552	CAL
17560	25612	HLN1
17561	30003	DTX60
17562	30040	DTX61
17563	30075	DTX62
17564	30132	DTX63
17565	30170	DTX64
17566	177777	-1



---

	17567	177777	-1	
U	17570	17570	DTX67	
	17571	177777	-1	
	17572	177777	-1	
	17573	25601	MCD	
	17574	25612	HLN1	
	17575	30226	DTX70	
	17576	30260	DTX71	
	17577	30312	DTX72	
	17600	177777	-1	
	17601	177777	-1	
	17602	25571	KEN	
	17603	25612	HLN1	
	17604	30344	DTX75	
	17605	30376	DTX76	
	17606	30430	DTX77	
	17607	177777	-1	
	17610	177777	-1	
	17611	0	0	; END OF LIST

; LOTUS 700 DISK CONTROLLER DIAGNOSTICS  
; VERSION 1.7 JAN 31, 1981

; MESSAGE TEXT FILE

```
; * * * * *  
; * * * * *  
; *          DDTX          *  
; * * * * *  
; * * * * *
```

1 .TXTM 1

17612 5114 TTLO: .TXT "<12>L  
17613 47524 OT  
17614 52523 US  
17615 20067 7  
17616 30060 00  
17617 20104 D  
17620 44523 IS  
17621 45440 K  
17622 41517 CO  
17623 47124 NT  
17624 51117 RO  
17625 46114 LL  
17626 42522 ER  
17627 20104 D  
17630 44501 IA  
17631 43516 GN  
17632 47523 OS  
17633 52111 TI  
17634 41523 CS  
17635 0 "

17636 53105 TTL1: .TXT "VE  
17637 51123 RS  
17640 44517 IO  
17641 47040 N  
17642 30456 1  
17643 33440 7  
17644 45101 JA  
17645 47040 N  
17646 31461 31  
17647 20061 1  
17650 34470 98  
17651 30400 1"

17652 42105 DEVC: .TXT "DE  
17653 53111 VI  
17654 41505 CE  
17655 20103 C  
17656 47504 OD  
17657 42472 E:  
17660 20000 "

17661 52105 TMOD: .TXT "TE  
17662 51524 ST  
17663 20115 M  
17664 47504 OD  
17665 42472 E:  
17666 20000 "

17667 51524 STPE: .TXT "ST  
17670 47520 OP  
17671 20117 0  
17672 47040 N  
17673 42522 ER  
17674 51117 RO  
17675 51077 R?  
17676 20000 "

17677 42122 DRONL: .TXT "DR

17700 44526 IV  
17701 42440 E  
17702 47516 ON  
17703 26514 -L  
17704 44516 IN  
17705 42441 E!  
17706 20120 P  
17707 46101 LA  
17710 41505 CE  
17711 20111 I  
17712 52040 T  
17713 47506 OF  
17714 43055 F-  
17715 46111 LI  
17716 47105 NE  
17717 20101 A  
17720 47104 ND  
17721 20110 H  
17722 44524 IT  
17723 20101 A  
17724 20113 K  
17725 42531 EY  
17726 20124 T  
17727 47440 O  
17730 51524 ST  
17731 40522 AR  
17732 52000 T"

17733 42122 DNAME: .TXT "DR  
17734 44526 IV  
17735 42440 E  
17736 47101 NA  
17737 46505 ME  
17740 35040 :  
17741 0 "

17742 42122 DNUMT: .TXT "DR  
17743 44526 IV  
17744 42440 E  
17745 47125 NU  
17746 46502 MB  
17747 42522 ER  
17750 35040 :  
17751 0 "

17752 42122 NRDT: .TXT "DR  
17753 44526 IV  
17754 42440 E  
17755 47117 NO  
17756 52040 T  
17757 51105 RE  
17760 40504 AD  
17761 54400 Y"

17762 51105 REFR: .TXT "RE  
17763 43105 FE  
17764 51105 RE  
17765 47103 NC  
17766 42440 E

17767 0 "  
 17770 20124 TSTT: .TXT " T  
 17771 42523 ES  
 17772 52000 T"  
 17773 47117 NORM: .TXT "NO  
 17774 51115 RM  
 17775 40514 AL  
 17776 35040 :  
 17777 0 "  
 20000 20127 WORD: .TXT " W  
 20001 47522 OR  
 20002 42040 D  
 20003 36440 =  
 20004 0 "  
 20005 50101 PAST: .TXT "PA  
 20006 51523 SS  
 20007 20000 "  
 20010 50101 PASSE: .TXT "PA  
 20011 51523 SS  
 20012 42523 ES  
 20013 5000 <12>"  
 20014 51505 STP: .TXT "SE  
 20015 52040 T  
 20016 51524 ST  
 20017 47520 OP  
 20020 0 "  
 20021 51505 NSTP: .TXT "SE  
 20022 52040 T  
 20023 47117 NO  
 20024 20123 S  
 20025 52117 TO  
 20026 50000 P"  
 20027 53502 AWB: .TXT "WB  
 20030 52506 UF  
 20031 43040 F  
 20032 40504 AD  
 20033 42122 DR  
 20034 20075 =  
 20035 20000 "  
 20036 51102 ARB: .TXT "RB  
 20037 52506 UF  
 20040 43040 F  
 20041 40504 AD  
 20042 42122 DR  
 20043 20075 =  
 20044 20000 "  
 20045 20104 DATA: .TXT " D  
 20046 40524 AT  
 20047 40440 A

```

20050 36440 =
20051 0 "

20052 20124 TMO: .TXT " T
20053 44515 IM
20054 42517 EO
20055 52524 UT
20056 0 "

20057 42122 DRNM: .TXT "DR
20060 44526 IV
20061 42440 E
20062 0 "

20063 44516 IN: .TXT "IN
20064 50125 PU
20065 52072 T:
20066 20000 "

20067 47525 OUT: .TXT "OU
20070 52120 TP
20071 52524 UT
20072 35040 :
20073 0 "

20074 5076 PRMT: .TXT "<12>>
20075 20000 "

20076 0 CRLF: .TXT "<0>"

20077 20104 MDOA: .TXT " D
20100 47501 OA
20101 20000 "

20102 20104 MDOB: .TXT " D
20103 47502 OB
20104 20000 "

20105 20104 MDOC: .TXT " D
20106 47503 OC
20107 20000 "

20110 20104 MDOBS: .TXT " D
20111 47502 OB
20112 51440 S
20113 0 "

20114 20104 MDOCP: .TXT " D
20115 47503 OC
20116 50040 P
20117 0 "

20120 20104 MDOAP: .TXT " D
20121 47501 OA
20122 50040 P
20123 0 "

20124 20104 MDOAS: .TXT " D
20125 47501 OA

```

```

-----
20126 51440 S
20127      0 "

20130 20104 MDOCS:.TXT " D
20131 47503 DC
20132 51440 S
20133      0 "

20134 20116 MNIOP:.TXT " N
20135 44517 IO
20136 50000 P"

20137 20104 MDIA: .TXT " D
20140 44501 IA
20141 20000 "

20142 20104 MDIB: .TXT " D
20143 44502 IB
20144 20000 "

20145 20104 MDIC: .TXT " D
20146 44503 IC
20147 20000 "

20150 20104 MDIA1: .TXT " D
20151 44501 IA
20152 24061 (1
20153 24440 )
20154      0 "

20155 20104 MDIB1: .TXT " D
20156 44502 IB
20157 24061 (1
20160 24440 )
20161      0 "

20162 20104 MDIA2: .TXT " D
20163 44501 IA
20164 24062 (2
20165 24440 )
20166      0 "

20167 20104 MDIB2: .TXT " D
20170 44502 IB
20171 24062 (2
20172 24440 )
20173      0 "

20174 37440 GMT: .TXT " ?
20175      0 "

20176 41125 CBFUL: .TXT "BU
20177 43106 FF
20200 42522 ER
20201 20106 F
20202 52514 UL
20203 46041 L!
20204 20122 R
20205 42455 E-

```

-----  
20206 42516 EN  
20207 52105 TE  
20210 51000 R"  
  
20211 42516 MSG0: .TXT "EN  
20212 52105 TE  
20213 51040 R  
20214 31040 2  
20215 47503 OC  
20216 52101 TA  
20217 46040 L  
20220 42111 DI  
20221 43511 GI  
20222 52123 TS  
20223 27040 .  
20224 42105 DE  
20225 43101 FA  
20226 52514 UL  
20227 52040 T  
20230 44523 IS  
20231 20062 2  
20232 33400 7"  
  
20233 46117 MSG1: .TXT "LO  
20234 43511 GI  
20235 41440 C  
20236 47522 OR  
20237 20114 L  
20240 20040  
20241 20124 T  
20242 42523 ES  
20243 52040 T  
20244 46117 LO  
20245 41501 CA  
20246 46040 L  
20247 46117 LO  
20250 43511 GI  
20251 41440 C  
20252 47516 ON  
20253 46131 LY  
20254 0 "  
  
20255 42122 MSG2: .TXT "DR  
20256 44526 IV  
20257 42440 E  
20260 47522 OR  
20261 20104 D  
20262 20040  
20263 20124 T  
20264 42523 ES  
20265 52040 T  
20266 53511 WI  
20267 52110 TH  
20270 20101 A  
20271 20104 D  
20272 51111 RI  
20273 53105 VE  
20274 0 "



20275 52105 MSG3: .TXT "TE  
20276 51524 ST  
20277 42522 ER  
20300 20117 0  
20301 51040 R  
20302 52040 T  
20303 20124 T  
20304 42523 ES  
20305 52040 T  
20306 53511 WI  
20307 52110 TH  
20310 20104 D  
20311 51111 RI  
20312 53105 VE  
20313 20105 E  
20314 46525 MU  
20315 46101 LA  
20316 52117 TO  
20317 51000 R"

20320 41517 HLNO: .TXT "CO  
20321 46515 MM  
20322 40516 AN  
20323 42040 D  
20324 20040  
20325 20040  
20326 20117 0  
20327 50105 PE  
20330 51101 RA  
20331 52111 TI  
20332 47516 ON  
20333 0 "

20334 20040 HCO: .TXT "  
20335 20114 L  
20336 20040  
20337 20040  
20340 20114 L  
20341 47517 00  
20342 50040 P  
20343 47516 ON  
20344 20105 E  
20345 51122 RR  
20346 47522 OR  
20347 0 "

20350 20040 HC1: .TXT "  
20351 20120 P  
20352 20040  
20353 20040  
20354 20120 P  
20355 51117 RO  
20356 41505 CE  
20357 42504 ED  
20360 20106 F  
20361 51117 RO  
20362 46440 M  
20363 42522 ER  
20364 51117 RO

-----  
20365 51000 R"  
  
20366 20040 HC2: .TXT "  
20367 20122 R  
20370 20040  
20371 20040  
20372 20122 R  
20373 42520 EP  
20374 42501 EA  
20375 52040 T  
20376 43122 FR  
20377 47515 OM  
20400 20123 S  
20401 52101 TA  
20402 51124 RT  
20403 20117 O  
20404 43040 F  
20405 52110 TH  
20406 44523 IS  
20407 20124 T  
20410 42523 ES  
20411 52000 T"  
  
20412 20040 HC3: .TXT "  
20413 20123 S  
20414 20040  
20415 20040  
20416 20123 S  
20417 45511 KI  
20420 50040 P  
20421 52117 TO  
20422 20116 N  
20423 42530 EX  
20424 52040 T  
20425 52105 TE  
20426 51524 ST  
20427 0 "  
  
20430 20040 HC4: .TXT "  
20431 42127 DW  
20432 41040 B  
20433 20040  
20434 20104 D  
20435 52515 UM  
20436 50040 P  
20437 53522 WR  
20440 44524 IT  
20441 42440 E  
20442 41125 BU  
20443 43106 FF  
20444 42522 ER  
20445 0 "  
  
20446 20040 HC5: .TXT "  
20447 42122 DR  
20450 41040 B  
20451 20040  
20452 20104 D  
20453 52515 UM

20454 50040 P  
 20455 51105 RE  
 20456 40504 AD  
 20457 20102 B  
 20460 52506 UF  
 20461 43105 FE  
 20462 51000 R"

20463 20040 HC6: .TXT "  
 20464 20130 X  
 20465 20040  
 20466 20040  
 20467 20101 A  
 20470 41117 BD  
 20471 51124 RT  
 20472 0 "

20473 20040 HC7: .TXT "  
 20474 20103 C  
 20475 20040  
 20476 20040  
 20477 20103 C  
 20500 44101 HA  
 20501 47107 NG  
 20502 42440 E  
 20503 51524 ST  
 20504 47520 OP  
 20505 20117 O  
 20506 50124 PT  
 20507 44517 ID  
 20510 47000 N"

20511 20104 HC8: .TXT " D  
 20512 42117 DO  
 20513 40440 A  
 20514 20040  
 20515 20104 D  
 20516 42503 EC  
 20517 47504 OD  
 20520 42440 E  
 20521 42117 DO  
 20522 40440 A  
 20523 53517 WO  
 20524 51104 RD  
 20525 0 "

20526 20104 HC9: .TXT " D  
 20527 42117 DO  
 20530 41440 C  
 20531 20040  
 20532 20104 D  
 20533 42503 EC  
 20534 47504 OD  
 20535 42440 E  
 20536 42117 DO  
 20537 41440 C  
 20540 53517 WO  
 20541 51104 RD  
 20542 0 "

20543 20104 HC10: .TXT " D  
20544 42111 DI  
20545 40440 A  
20546 20040  
20547 20104 D  
20550 42503 EC  
20551 47504 OD  
20552 42440 E  
20553 42111 DI  
20554 40440 A  
20555 53517 WO  
20556 51104 RD  
20557 0 "

20560 20104 HC11: .TXT " D  
20561 42111 DI  
20562 41040 B  
20563 20040  
20564 20104 D  
20565 42503 EC  
20566 47504 OD  
20567 42440 E  
20570 42111 DI  
20571 41040 B  
20572 53517 WO  
20573 51104 RD  
20574 0 "

20575 46117 PNAM1: .TXT "LO  
20576 43511 GI  
20577 41400 C"

20600 42122 PNAM2: .TXT "DR  
20601 44526 IV  
20602 42400 E"

20603 52105 PNAM3: .TXT "TE  
20604 51524 ST  
20605 42522 ER  
20606 0 "

20607 42000 CHD: .TXT "D"

20610 52000 CHT: .TXT "T"

20611 46000 CHL: .TXT "L"

20612 50000 CHP: .TXT "P"

20613 51000 CHR: .TXT "R"

20614 51400 CHS: .TXT "S"

20615 42127 CHWB: .TXT "DW  
20616 41000 B"

20617 42122 CHR B: .TXT "DR  
20620 41000 B"

20621 54000 CHX: .TXT "X"  
 20622 44000 CHH: .TXT "H"  
 20623 41400 CHC: .TXT "C"  
 20624 54400 CHY: .TXT "Y"  
 20625 47000 CHN: .TXT "N"  
 20626 45000 CHJ: .TXT "J"  
 20627 42104 CDDA: .TXT "DD  
 20630 47501 DA  
 20631 0 "  
 20632 42104 CDDC: .TXT "DD  
 20633 47503 DC  
 20634 0 "  
 20635 42104 CDDA: .TXT "DD  
 20636 44501 IA  
 20637 0 "  
 20640 42104 CDIB: .TXT "DD  
 20641 44502 IB  
 20642 0 "  
 20643 20103 CLRW: .TXT " C  
 20644 46122 LR  
 20645 20122 R  
 20646 53440 W  
 20647 40524 AT  
 20650 52054 T,  
 20651 0 "  
 20652 20104 CDR0: .TXT " D  
 20653 51126 RV  
 20654 20060 0  
 20655 20101 A  
 20656 52124 TT  
 20657 26000 ,"  
 20660 20104 CDR1: .TXT " D  
 20661 51126 RV  
 20662 20061 1  
 20663 20101 A  
 20664 52124 TT  
 20665 26000 ,"  
 20666 20104 CDR2: .TXT " D  
 20667 51126 RV  
 20670 20062 2  
 20671 20101 A  
 20672 52124 TT  
 20673 26000 ,"  
 20674 20104 CDR3: .TXT " D

```

---
20675 51126 RV
20676 20063 3
20677 20101 A
20700 52124 TT
20701 26000 , "

20702 20125 UDC: .TXT " U
20703 47104 ND
20704 42506 EF
20705 44516 IN
20706 42504 ED
20707 20103 C
20710 47515 DM
20711 46501 MA
20712 47104 ND
20713 0 "

20714 20126 VO: .TXT " V
20715 47514 OL
20716 20060 0
20717 0 "

20720 20126 V1: .TXT " V
20721 47514 OL
20722 20061 1
20723 0 "

20724 20123 SEEK: .TXT " S
20725 42505 EE
20726 45400 K"

20727 20106 FMATT: .TXT " F
20730 47522 OR
20731 46501 MA
20732 52000 T"

20733 20123 ALTT1: .TXT " S
20734 42524 ET
20735 20101 A
20736 46124 LT
20737 20115 M
20740 47504 OD
20741 42440 E
20742 30400 1"

20743 20123 ALTT2: .TXT " S
20744 42524 ET
20745 20101 A
20746 46124 LT
20747 20115 M
20750 47504 OD
20751 42440 E
20752 31000 2"

20753 20116 NOP: .TXT " N
20754 47440 0
20755 47520 OP
20756 0 "

```

20757 20104 DATVT: .TXT " D  
 20760 40524 AT  
 20761 40440 A  
 20762 53105 VE  
 20763 51111 RI  
 20764 43131 FY  
 20765 0 "

20766 44040 HD: .TXT "H  
 20767 0 "

20770 20123 ST: .TXT " S  
 20771 20000 "

20772 20123 SC: .TXT " S  
 20773 41440 C  
 20774 0 "

20775 20103 CF: .TXT " C  
 20776 47516 ON  
 20777 52122 TR  
 21000 47514 OL  
 21001 20106 F  
 21002 52514 UL  
 21003 46054 L,  
 21004 0 "

21005 20122 RWA: .TXT " R  
 21006 53440 W  
 21007 40524 AT  
 21010 52054 T,  
 21011 0 "

21012 20111 ISA: .TXT " I  
 21013 46114 LL  
 21014 20123 S  
 21015 42503 EC  
 21016 52040 T  
 21017 40504 AD  
 21020 42122 DR  
 21021 26000 , "

21022 20102 BS: .TXT " B  
 21023 40504 AD  
 21024 20123 S  
 21025 42503 EC  
 21026 52054 T,  
 21027 0 "

21030 20103 CYE: .TXT " C  
 21031 54514 YL  
 21032 20101 A  
 21033 42104 DD  
 21034 51040 R  
 21035 42522 ER  
 21036 51054 R,  
 21037 0 "

21040 20123 SAE: .TXT " S

21041 52522 UR  
 21042 43057 F<57>  
 21043 51505 SE  
 21044 41524 CT  
 21045 20101 A  
 21046 42104 DD  
 21047 51040 R  
 21050 42522 ER  
 21051 51054 R,  
 21052 0 "

21053 20126 VE: .TXT " V  
 21054 42522 ER  
 21055 44506 IF  
 21056 54440 Y  
 21057 42522 ER  
 21060 51054 R,  
 21061 0 "

21062 20105 ECE: .TXT " E  
 21063 41503 CC  
 21064 20105 E  
 21065 51122 RR  
 21066 26000 ,"

21067 20122 RWD: .TXT " R  
 21070 53440 W  
 21071 52111 TI  
 21072 46505 ME  
 21073 47525 OU  
 21074 52054 T,  
 21075 0 "

21076 20104 DL: .TXT " D  
 21077 40524 AT  
 21100 40440 A  
 21101 46101 LA  
 21102 52105 TE  
 21103 26000 ,"

21104 20122 RWF: .TXT " R  
 21105 53440 W  
 21106 43101 FA  
 21107 52514 UL  
 21110 52000 T"

21111 20104 DR: .TXT " D  
 21112 51126 RV  
 21113 20122 R  
 21114 42501 EA  
 21115 42131 DY  
 21116 26000 ,"

21117 20104 DB: .TXT " D  
 21120 51126 RV  
 21121 20102 B  
 21122 52523 US  
 21123 54454 Y,  
 21124 0 "



```

21125 20111 SE: .TXT " I
21126 46114 LL
21127 20123 S
21130 42505 EE
21131 45440 K
21132 40504 AD
21133 42122 DR
21134 26000 , "

21135 20111 IC: .TXT " I
21136 46114 LL
21137 20103 C
21140 46504 MD
21141 26000 , "

21142 20104 DF: .TXT " D
21143 51126 RV
21144 20106 F
21145 40525 AU
21146 46124 LT
21147 26000 , "

21150 20104 DER: .TXT " D
21151 51126 RV
21152 20105 E
21153 51122 RR
21154 0 "

21155 46501 MRST: .TXT "MA
21156 51524 ST
21157 42522 ER
21160 20122 R
21161 42523 ES
21162 42524 ET
21163 0 "

21164 23502 BSYF: .TXT " 'B
21165 52523 US
21166 54447 Y'
21167 20106 F
21170 46101 LA
21171 43400 G"

21172 44517 RSET: .TXT " ID
21173 20122 R
21174 42523 ES
21175 42524 ET
21176 0 "

21177 23504 DNEF: .TXT " 'D
21200 47516 ON
21201 42447 E'
21202 20106 F
21203 46101 LA
21204 43400 G"

21205 41517 CTRG: .TXT "CO
21206 47124 NT

```

21207 51117 RO  
 21210 46040 L  
 21211 51524 ST  
 21212 40524 AT  
 21213 52523 US  
 21214 20122 R  
 21215 42507 EG  
 21216 44523 IS  
 21217 52105 TE  
 21220 51000 R"

21221 30060 ALLO: .TXT "00  
 21222 30060 00  
 21223 30060 00  
 21224 0 "

21225 30467 ALL1: .TXT "17  
 21226 33467 77  
 21227 33467 77  
 21230 0 "

21231 47117 NCLA: .TXT "NO  
 21232 52040 T  
 21233 41514 CL  
 21234 42501 EA  
 21235 51105 RE  
 21236 42040 D  
 21237 41131 BY  
 21240 20103 C  
 21241 46105 LE  
 21242 40522 AR  
 21243 20101 A  
 21244 52124 TT  
 21245 42516 EN  
 21246 20102 B  
 21247 44524 IT  
 21250 51400 S"

21251 42122 DRRG: .TXT "DR  
 21252 44526 IV  
 21253 42440 E  
 21254 51524 ST  
 21255 40524 AT  
 21256 52523 US  
 21257 20122 R  
 21260 42507 EG  
 21261 44523 IS  
 21262 52105 TE  
 21263 51000 R"

21264 47117 NRST: .TXT "NO  
 21265 52040 T  
 21266 51105 RE  
 21267 51505 SE  
 21270 52000 T"

21271 51525 DARG: .TXT "SU  
 21272 51106 RF  
 21273 27523 <57>S

21274 42503 EC  
 21275 52040 T  
 21276 40504 AD  
 21277 42122 DR  
 21300 20122 R  
 21301 42507 EG  
 21302 0 "

21303 51505 SBSY: .TXT "SE  
 21304 52040 T  
 21305 23502 'B  
 21306 52523 US  
 21307 54447 Y'  
 21310 20106 F  
 21311 46101 LA  
 21312 43400 G"

21313 30067 SVENS: .TXT "07  
 21314 33467 77  
 21315 33467 77  
 21316 0 "

21317 46505 AM1A: .TXT "ME  
 21320 46440 M  
 21321 40504 AD  
 21322 42122 DR  
 21323 20122 R  
 21324 42507 EG  
 21325 20040  
 21326 20104 D  
 21327 41440 C  
 21330 31067 27  
 21331 0 "

21332 51525 AM1B: .TXT "SU  
 21333 51106 RF  
 21334 27523 <57>S  
 21335 42503 EC  
 21336 52040 T  
 21337 51105 RE  
 21340 43440 G  
 21341 20040  
 21342 42103 DC  
 21343 20062 2  
 21344 33400 7"

21345 46505 AM1C: .TXT "ME  
 21346 46440 M  
 21347 40504 AD  
 21350 42122 DR  
 21351 20122 R  
 21352 42507 EG  
 21353 20040  
 21354 20104 D  
 21355 41440 C  
 21356 54130 XX  
 21357 0 "

21360 51525 AM1D: .TXT "SU

21361 51106 RF  
21362 27523 <57>S  
21363 42503 EC  
21364 52040 T  
21365 51105 RE  
21366 43440 G  
21367 20040  
21370 42103 DC  
21371 20130 X  
21372 54000 X"

21373 41517 CFON: .TXT "CO  
21374 47124 NT  
21375 51117 RO  
21376 46040 L  
21377 43125 FU  
21400 46114 LL  
21401 20102 B  
21402 44524 IT  
21403 20117 O  
21404 47000 N"

21405 41517 CFOF: .TXT "CO  
21406 47124 NT  
21407 51117 RO  
21410 46040 L  
21411 43125 FU  
21412 46114 LL  
21413 20102 B  
21414 44524 IT  
21415 20117 O  
21416 43106 FF  
21417 20101 A  
21420 43124 FT  
21421 42522 ER  
21422 20061 I  
21423 30040 O  
21424 52523 US  
21425 20104 D  
21426 42514 EL  
21427 40531 AY  
21430 0 "

21431 42111 DATI1: .TXT "DI  
21432 40440 A  
21433 30460 IO  
21434 30130 OX  
21435 54130 XX  
21436 0 "

21437 42111 DATI2: .TXT "DI  
21440 40440 A  
21441 30060 OO  
21442 30130 OX  
21443 54130 XX  
21444 0 "

21445 42111 DATI3: .TXT "DI  
21446 40440 A

```

21447 30060 00
21450 30060 00
21451 30060 00
21452 0 "

21453 42111 DATI4:.TXT "DI
21454 40440 A
21455 30467 17
21456 33467 77
21457 33467 77
21460 0 "

21461 42111 DATI5:.TXT "DI
21462 41440 C
21463 30067 07
21464 33467 77
21465 33467 77
21466 0 "

21467 42111 DATI6:.TXT "DI
21470 41440 C
21471 30060 00
21472 30060 00
21473 30060 00
21474 0 "

21475 42111 DATI7:.TXT "DI
21476 41040 B
21477 30060 00
21500 30060 00
21501 30060 00
21502 0 "

21503 42111 DATI8:.TXT "DI
21504 40440 A
21505 30064 04
21506 30130 0X
21507 54130 XX
21510 0 "

21511 42111 DATI9:.TXT "DI
21512 40440 A
21513 30060 00
21514 30064 04
21515 30060 00
21516 0 "

21517 42111 DATNO:.TXT "DI
21520 40440 A
21521 30062 02
21522 30130 0X
21523 54130 XX
21524 0 "

21525 42111 DATN1:.TXT "DI
21526 40440 A
21527 30061 01
21530 30130 0X
21531 54130 XX

```

```

21532      0 "

21533  42111 DATN2: . TXT  "DI
21534  40440 A
21535  30060 00
21536  32130 4X
21537  54130 XX
21540      0 "

21541  42111 DATN3: . TXT  "DI
21542  40440 A
21543  30060 00
21544  31130 2X
21545  54130 XX
21546      0 "

21547  42111 DATN4: . TXT  "DI
21550  40440 A
21551  30063 03
21552  30130 0X
21553  54130 XX
21554      0 "

21555  42111 DATN5: . TXT  "DI
21556  40440 A
21557  30063 03
21560  32130 4X
21561  54130 XX
21562      0 "

21563  42111 DATN6: . TXT  "DI
21564  40440 A
21565  30063 03
21566  33130 6X
21567  54130 XX
21570      0 "

21571  42111 DATN7: . TXT  "DI
21572  41040 B
21573  30060 00
21574  30060 00
21575  30060 00
21576      0 "

21577  51105 RDUN: . TXT  "RE
21600  43511 GI
21601  51524 ST
21602  42522 ER
21603  20104 D
21604  40524 AT
21605  40440 A
21606  52516 UN
21607  40506 AF
21610  43105 FE
21611  41524 CT
21612  42504 ED
21613  20102 B
21614  54440 Y
21615  47111 NI

```

21616 47503 OC  
 21617 0 "

21620 47111 MNIOC: .TXT "NI  
 21621 47503 OC  
 21622 20062 2  
 21623 33400 7"

21624 41514 CLAT: .TXT "CL  
 21625 42501 EA  
 21626 51040 R  
 21627 51105 RE  
 21630 43440 G  
 21631 41131 BY  
 21632 20103 C  
 21633 46105 LE  
 21634 40522 AR  
 21635 20101 A  
 21636 52124 TT  
 21637 42516 EN  
 21640 0 "

21641 51105 RDUD: .TXT "RE  
 21642 43511 GI  
 21643 51524 ST  
 21644 42522 ER  
 21645 20104 D  
 21646 40524 AT  
 21647 40440 A  
 21650 52516 UN  
 21651 40506 AF  
 21652 43105 FE  
 21653 41524 CT  
 21654 42504 ED  
 21655 20102 B  
 21656 54440 Y  
 21657 42117 DO  
 21660 40400 A"

21661 23504 DNEO: .TXT "'D  
 21662 47516 ON  
 21663 42447 E'  
 21664 20075 =  
 21665 20060 0  
 21666 0 "

21667 23504 DNE1: .TXT "'D  
 21670 47516 ON  
 21671 42447 E'  
 21672 20075 =  
 21673 20061 1  
 21674 0 "

21675 42117 DNP: .TXT "DO  
 21676 47105 NE  
 21677 20101 A  
 21700 52040 T  
 21701 42117 DO  
 21702 41520 CP

21703 20053 +  
 21704 20061 1  
 21705 30040 0  
 21706 52523 US  
 21707 0 "

21710 51505 DRAT: .TXT "SE  
 21711 52040 T  
 21712 42122 DR  
 21713 44526 IV  
 21714 42440 E  
 21715 40524 AT  
 21716 52105 TE  
 21717 47040 N  
 21720 41111 BI  
 21721 52000 T"

21722 41514 CLOD: .TXT "CL  
 21723 42501 EA  
 21724 51040 R  
 21725 47524 OT  
 21726 44105 HE  
 21727 51040 R  
 21730 42122 DR  
 21731 44526 IV  
 21732 42440 E  
 21733 40524 AT  
 21734 52105 TE  
 21735 47000 N"

21736 41514 CLDA: .TXT "CL  
 21737 42501 EA  
 21740 51040 R  
 21741 42122 DR  
 21742 44526 IV  
 21743 42440 E  
 21744 40524 AT  
 21745 52105 TE  
 21746 47000 N"

21747 41514 CLDN: .TXT "CL  
 21750 42501 EA  
 21751 51040 R  
 21752 23504 'D  
 21753 47516 ON  
 21754 42447 E'  
 21755 0 "

21756 23523 SRSD: .TXT "'S  
 21757 52101 TA  
 21760 51124 RT  
 21761 23440 '  
 21762 52117 TO  
 21763 20122 R  
 21764 42523 ES  
 21765 42524 ET  
 21766 20047 '  
 21767 41125 BU  
 21770 51531 SY



21771 23400 ' "  
 21772 51105 RXMA: .TXT "RE  
 21773 40504 AD  
 21774 20105 E  
 21775 54124 XT  
 21776 23504 'D  
 21777 20115 M  
 22000 42515 EM  
 22001 20101 A  
 22002 42104 DD  
 22003 51040 R  
 22004 41111 BI  
 22005 52123 TS  
 22006 0 "  
  
 22007 40523 ASDOB: .TXT "AS  
 22010 20111 I  
 22011 47040 N  
 22012 42117 DO  
 22013 41040 B  
 22014 41517 CO  
 22015 46515 MM  
 22016 40516 AN  
 22017 42000 D"  
  
 22020 53522 WLK1: .TXT "WR  
 22021 44524 IT  
 22022 42455 E-  
 22023 51105 RE  
 22024 40504 AD  
 22025 20101 A  
 22026 20127 W  
 22027 40514 AL  
 22030 45511 KI  
 22031 47107 NG  
 22032 20047 '  
 22033 30447 1'  
 22034 0 "  
  
 22035 53522 WLK0: .TXT "WR  
 22036 44524 IT  
 22037 42455 E-  
 22040 51105 RE  
 22041 40504 AD  
 22042 20101 A  
 22043 20127 W  
 22044 40514 AL  
 22045 45511 KI  
 22046 47107 NG  
 22047 20047 '  
 22050 30047 0'  
 22051 0 "  
  
 22052 40523 ASDOC: .TXT "AS  
 22053 20111 I  
 22054 47040 N  
 22055 42117 DO  
 22056 41440 C

22057 41517 CD  
 22060 46515 MM  
 22061 40516 AN  
 22062 42000 D"

22063 41514 CLBY: .TXT "CL  
 22064 42501 EA  
 22065 51040 R  
 22066 23502 'B  
 22067 52523 US  
 22070 54447 Y'  
 22071 0 "

22072 47117 NCLR: .TXT "NO  
 22073 52040 T  
 22074 41514 CL  
 22075 42501 EA  
 22076 51105 RE  
 22077 42000 D"

22100 51105 RSBY: .TXT "RE  
 22101 51505 SE  
 22102 52040 T  
 22103 23502 'B  
 22104 52523 US  
 22105 54447 Y'  
 22106 0 "

22107 23502 BSYO: .TXT "'B  
 22110 52523 US  
 22111 54447 Y'  
 22112 20075 =  
 22113 20060 0  
 22114 0 "

22115 23502 BSY1: .TXT "'B  
 22116 52523 US  
 22117 54447 Y'  
 22120 20075 =  
 22121 20061 1  
 22122 0 "

22123 23502 BSYOD: .TXT "'B  
 22124 52523 US  
 22125 54447 Y'  
 22126 20075 =  
 22127 20060 0  
 22130 20061 1  
 22131 32440 5  
 22132 52523 US  
 22133 20101 A  
 22134 43124 FT  
 22135 42522 ER  
 22136 20047 '  
 22137 51524 ST  
 22140 40522 AR  
 22141 52047 T'  
 22142 0 "

22143 23504 DNE1D: .TXT " 'D  
22144 47516 ON  
22145 42447 E'  
22146 20075 =  
22147 20061 1  
22150 20061 1  
22151 32440 5  
22152 52523 US  
22153 20101 A  
22154 43124 FT  
22155 42522 ER  
22156 20047 '  
22157 51524 ST  
22160 40522 AR  
22161 52047 T'  
22162 0 "

22163 51505 SRWD: .TXT "SE  
22164 52040 T  
22165 51127 RW  
22166 20104 D  
22167 47516 ON  
22170 42440 E  
22171 51524 ST  
22172 40524 AT  
22173 52523 US  
22174 0 "

22175 44516 INTAK: .TXT " IN  
22176 52105 TE  
22177 51122 RR  
22200 52520 UP  
22201 52040 T  
22202 40503 AC  
22203 45516 KN  
22204 47527 OW  
22205 46105 LE  
22206 42107 DG  
22207 42400 E"

22210 44516 INTI: .TXT " IN  
22211 52101 TA  
22212 20075 =  
22213 20060 0  
22214 30060 00  
22215 0 "

22216 44516 INTN: .TXT " IN  
22217 52101 TA  
22220 20075 =  
22221 20060 0  
22222 30060 00  
22223 0 "

22224 44516 IMSK1: .TXT " IN  
22225 52105 TE  
22226 51122 RR  
22227 52520 UP  
22230 52040 T

22231 46501 MA  
 22232 51513 SK  
 22233 20117 0  
 22234 47000 N"

22235 44516 IMSKO: .TXT " IN  
 22236 52105 TE  
 22237 51122 RR  
 22240 52520 UP  
 22241 52040 T  
 22242 46501 MA  
 22243 51513 SK  
 22244 20117 0  
 22245 43106 FF  
 22246 0 "

22247 44516 MKB1: .TXT " IN  
 22250 52105 TE  
 22251 51122 RR  
 22252 52520 UP  
 22253 52040 T  
 22254 46501 MA  
 22255 51513 SK  
 22256 20075 =  
 22257 20060 0  
 22260 30060 00  
 22261 32060 40  
 22262 30000 0"

22263 44516 INTRT: .TXT " IN  
 22264 52105 TE  
 22265 51122 RR  
 22266 52520 UP  
 22267 52040 T  
 22270 42516 EN  
 22271 40502 AB  
 22272 46105 LE  
 22273 42000 D"

22274 44516 INTOC: .TXT " IN  
 22275 52105 TE  
 22276 51122 RR  
 22277 52520 UP  
 22300 52040 T  
 22301 47503 OC  
 22302 41525 CU  
 22303 51122 RR  
 22304 42504 ED  
 22305 0 "

22306 46501 INHI: .TXT " MA  
 22307 51513 SK  
 22310 20124 T  
 22311 47440 0  
 22312 44516 IN  
 22313 44110 HH  
 22314 44502 IB  
 22315 44524 IT  
 22316 20111 I

22317 47124 NT  
 22320 42522 ER  
 22321 51125 RU  
 22322 50124 PT  
 22323 0 "

22324 44516 MKBO: .TXT "IN  
 22325 52105 TE  
 22326 51122 RR  
 22327 52520 UP  
 22330 52040 T  
 22331 46501 MA  
 22332 51513 SK  
 22333 20075 =  
 22334 20060 0  
 22335 30060 00  
 22336 30060 00  
 22337 30000 0"

22340 47117 NINT: .TXT "NO  
 22341 20111 I  
 22342 47124 NT  
 22343 42522 ER  
 22344 51125 RU  
 22345 50124 PT  
 22346 0 "

22347 51105 RBFC: .TXT "RE  
 22350 40504 AD  
 22351 20102 B  
 22352 52506 UF  
 22353 43105 FE  
 22354 51123 RS  
 22355 20103 C  
 22356 47515 OM  
 22357 46501 MA  
 22360 47104 ND  
 22361 0 "

22362 53105 VRBD: .TXT "VE  
 22363 51111 RI  
 22364 43131 FY  
 22365 20122 R  
 22366 42501 EA  
 22367 42040 D  
 22370 41125 BU  
 22371 43106 FF  
 22372 42522 ER  
 22373 51440 S  
 22374 42101 DA  
 22375 52101 TA  
 22376 0 "

22377 51102 ADB: .TXT "RB  
 22400 52506 UF  
 22401 43040 F  
 22402 42101 DA  
 22403 52101 TA  
 22404 20075 =

22405 20127 W  
22406 41125 BU  
22407 43106 FF  
22410 20104 D  
22411 40524 AT  
22412 40400 A"  
  
22413 50122 PTD: .TXT "PR  
22414 47515 OM  
22415 20124 T  
22416 42523 ES  
22417 52040 T  
22420 51105 RE  
22421 51525 SU  
22422 46124 LT  
22423 51400 S"  
  
22424 42122 PLLS: .TXT "DR  
22425 53040 V  
22426 20040  
22427 46101 LA  
22430 51524 ST  
22431 20040  
22432 20114 L  
22433 40523 AS  
22434 52040 T  
22435 51525 SU  
22436 51106 RF  
22437 40503 AC  
22440 42415 E<15>  
22441 5040 <12>  
22442 20040  
22443 20040  
22444 51505 SE  
22445 41524 CT  
22446 47522 OR  
22447 20040  
22450 53117 VD  
22451 46040 L  
22452 30040 O  
22453 20126 V  
22454 47514 OL  
22455 20061 1  
22456 5000 <12>"  
  
22457 44516 INCK: .TXT "IN  
22460 51524 ST  
22461 40514 AL  
22462 46101 LA  
22463 52111 TI  
22464 47516 ON  
22465 20103 C  
22466 44105 HE  
22467 41513 CK  
22470 0 "  
  
22471 51525 NPMO: .TXT "SU  
22472 51106 RF  
22473 27523 <57>S

22474 42503 EC  
 22475 52040 T  
 22476 50122 PR  
 22477 47515 OM  
 22500 51440 S  
 22501 47117 NO  
 22502 52040 T  
 22503 44516 IN  
 22504 51524 ST  
 22505 40514 AL  
 22506 46105 LE  
 22507 42000 D"

22510 51525 NPM1: .TXT "SU  
 22511 51106 RF  
 22512 40503 AC  
 22513 42440 E  
 22514 50122 PR  
 22515 47515 OM  
 22516 20116 N  
 22517 47524 OT  
 22520 20111 I  
 22521 47123 NS  
 22522 52101 TA  
 22523 46114 LL  
 22524 42504 ED  
 22525 0 "

22526 51505 SAFD: .TXT "SE  
 22527 41524 CT  
 22530 47522 DR  
 22531 20101 A  
 22532 42104 DD  
 22533 51040 R  
 22534 50122 PR  
 22535 47515 OM  
 22536 0 "

22537 51525 HDV0: .TXT "SU  
 22540 51106 RF  
 22541 20101 A  
 22542 42104 DD  
 22543 51040 R  
 22544 50122 PR  
 22545 47515 OM  
 22546 20126 V  
 22547 47514 OL  
 22550 20060 0  
 22551 0 "

22552 51525 HDV1: .TXT "SU  
 22553 51106 RF  
 22554 20101 A  
 22555 42104 DD  
 22556 51040 R  
 22557 50122 PR  
 22560 47515 OM  
 22561 20126 V  
 22562 47514 OL

22563 20061 1  
22564 0 "

22565 42122 DRST: .TXT "DR  
22566 44526 IV  
22567 42440 E  
22570 44523 IS  
22571 20122 R  
22572 42523 ES  
22573 42522 ER  
22574 53105 VE  
22575 42000 D"

22576 42122 WDST: .TXT "DR  
22577 44526 IV  
22600 42440 E  
22601 44523 IS  
22602 20127 W  
22603 51111 RI  
22604 52105 TE  
22605 20120 P  
22606 51117 RO  
22607 52105 TE  
22610 41524 CT  
22611 42504 ED  
22612 0 "

22613 42122 DRFT: .TXT "DR  
22614 44526 IV  
22615 42440 E  
22616 43101 FA  
22617 52514 UL  
22620 52040 T  
22621 51524 ST  
22622 40524 AT  
22623 52523 US  
22624 20075 =  
22625 20000 "

22626 42122 DDBS: .TXT "DR  
22627 53040 V  
22630 40524 AT  
22631 52105 TE  
22632 47040 N  
22633 23040 &  
22634 41125 BU  
22635 51531 SY  
22636 20123 S  
22637 52101 TA  
22640 52125 TU  
22641 51400 S"

22642 42122 DBY: .TXT "DR  
22643 44526 IV  
22644 42440 E  
22645 47117 NO  
22646 52040 T  
22647 41125 BU



22650 51531 SY  
 22651 20123 S  
 22652 52101 TA  
 22653 52125 TU  
 22654 51400 S"

22655 42122 DNST: .TXT "DR  
 22656 44526 IV  
 22657 42440 E  
 22660 47117 NO  
 22661 52040 T  
 22662 51105 RE  
 22663 40504 AD  
 22664 54440 Y  
 22665 51524 ST  
 22666 40524 AT  
 22667 52523 US  
 22670 0 "

22671 42122 DBS: .TXT "DR  
 22672 53040 V  
 22673 41125 BU  
 22674 51531 SY  
 22675 20123 S  
 22676 52101 TA  
 22677 52125 TU  
 22700 51400 S"

22701 51127 RWDS: .TXT "RW  
 22702 20104 D  
 22703 47516 ON  
 22704 42440 E  
 22705 51524 ST  
 22706 40524 AT  
 22707 52523 US  
 22710 20103 C  
 22711 44105 HE  
 22712 41513 CK  
 22713 0 "

22714 51127 RWDA: .TXT "RW  
 22715 20104 D  
 22716 47516 ON  
 22717 42440 E  
 22720 42111 DI  
 22721 51513 SK  
 22722 20101 A  
 22723 42104 DD  
 22724 51040 R  
 22725 41510 CH  
 22726 42503 EC  
 22727 45400 K"

22730 44105 HWC: .TXT "HE  
 22731 40504 AD  
 22732 42522 ER  
 22733 20127 W  
 22734 47522 OR  
 22735 42040 D

22736 41517 CO  
 22737 46520 MP  
 22740 40522 AR  
 22741 42400 E"

22742 40523 ASWB: .TXT "AS  
 22743 20111 I  
 22744 47040 N  
 22745 53502 WB  
 22746 52506 UF  
 22747 43000 F"

22750 42101 DWC: .TXT "DA  
 22751 52101 TA  
 22752 20127 W  
 22753 47522 OR  
 22754 42040 D  
 22755 41517 CO  
 22756 46520 MP  
 22757 40522 AR  
 22760 42400 E"

22761 46505 TMG: .TXT "ME  
 22762 40523 AS  
 22763 52522 UR  
 22764 42440 E  
 22765 52111 TI  
 22766 46505 ME  
 22767 47525 OU  
 22770 52040 T  
 22771 52111 TI  
 22772 46505 ME  
 22773 0 "

22774 51105 RDNX: .TXT "RE  
 22775 40504 AD  
 22776 20104 D  
 22777 40524 AT  
 23000 40440 A  
 23001 43122 FR  
 23002 47516 ON  
 23003 20116 N  
 23004 47505 OE  
 23005 54111 XI  
 23006 51524 ST  
 23007 42516 EN  
 23010 52040 T  
 23011 51505 SE  
 23012 41524 CT  
 23013 47522 OR  
 23014 0 "

23015 52111 TOAT: .TXT "TI  
 23016 46505 ME  
 23017 42040 D  
 23020 47525 OU  
 23021 52040 T  
 23022 40524 AT  
 23023 20000 "

23024 36476 TOLR: .TXT   "=<76>  
 23025 20060 0  
 23026 27065 .5  
 23027 20124 T  
 23030 47440 0  
 23031 36474 =<74>  
 23032 20062 2  
 23033 20123 S  
 23034 42503 EC  
 23035 0 "

23036 42101 DRB: .TXT   "DA  
 23037 52101 TA  
 23040 20122 R  
 23041 42501 EA  
 23042 42040 D  
 23043 44516 IN  
 23044 52117 TO  
 23045 20122 R  
 23046 41125 BU  
 23047 43106 FF  
 23050 0 "

23051 51510 NFST: .TXT   "SH  
 23052 47525 DU  
 23053 46104 LD  
 23054 20116 N  
 23055 47524 OT  
 23056 20110 H  
 23057 40526 AV  
 23060 42440 E  
 23061 43117 FO  
 23062 52516 UN  
 23063 42040 D  
 23064 51505 SE  
 23065 41524 CT  
 23066 47522 OR  
 23067 0 "

23070 51105 RSTH: .TXT   "RE  
 23071 51524 ST  
 23072 47522 OR  
 23073 44516 IN  
 23074 43440 G  
 23075 44105 HE  
 23076 40504 AD  
 23077 42522 ER  
 23100 0 "

23101 51525 ST11: .TXT   "SU  
 23102 51106 RF  
 23103 27523 <57>S  
 23104 42503 EC  
 23105 52040 T  
 23106 40504 AD  
 23107 42122 DR  
 23110 20105 E  
 23111 51122 RR

23112 47522 OR  
 23113 20123 S  
 23114 52101 TA  
 23115 52125 TU  
 23116 51400 S"

23117 53522 WHD: .TXT "WR  
 23120 44524 IT  
 23121 42440 E  
 23122 44105 HE  
 23123 40504 AD  
 23124 42522 ER  
 23125 0 "

23126 42122 DSCK: .TXT "DR  
 23127 44526 IV  
 23130 42440 E  
 23131 51524 ST  
 23132 40524 AT  
 23133 52523 US  
 23134 20103 C  
 23135 44105 HE  
 23136 41513 CK  
 23137 0 "

23140 51105 RHD: .TXT "RE  
 23141 40504 AD  
 23142 20110 H  
 23143 42501 EA  
 23144 42105 DE  
 23145 51000 R"

23146 43117 FMKO: .TXT "FO  
 23147 51115 RM  
 23150 40524 AT  
 23151 20103 C  
 23152 54514 YL  
 23153 20060 O  
 23154 0 "

23155 43117 FMK1: .TXT "FO  
 23156 51115 RM  
 23157 40524 AT  
 23160 20103 C  
 23161 54514 YL  
 23162 20061 1  
 23163 0 "

23164 51105 RMKO: .TXT "RE  
 23165 40504 AD  
 23166 20110 H  
 23167 42501 EA  
 23170 42105 DE  
 23171 51040 R  
 23172 41531 CY  
 23173 46040 L  
 23174 30000 O"

23175 51105 RMK1: .TXT "RE

23176 40504 AD  
 23177 20110 H  
 23200 42501 EA  
 23201 42105 DE  
 23202 51040 R  
 23203 41531 CY  
 23204 46040 L  
 23205 30400 1"

23206 53522 WDT: .TXT "WR  
 23207 44524 IT  
 23210 42440 E  
 23211 42101 DA  
 23212 52101 TA  
 23213 0 "

23214 51105 RDT: .TXT "RE  
 23215 40504 AD  
 23216 20104 D  
 23217 40524 AT  
 23220 40400 A"

23221 42503 CECC: .TXT "EC  
 23222 41440 C  
 23223 41510 CH  
 23224 42503 EC  
 23225 45400 K"

23226 53105 VBD: .TXT "VE  
 23227 51111 RI  
 23230 43131 FY  
 23231 20113 K  
 23232 47117 ND  
 23233 53516 WN  
 23234 20104 D  
 23235 40524 AT  
 23236 40440 A  
 23237 42522 ER  
 23240 51117 RD  
 23241 51000 R"

23242 46505 MSKT: .TXT "ME  
 23243 40523 AS  
 23244 52522 UR  
 23245 42440 E  
 23246 51505 SE  
 23247 42513 EK  
 23250 20124 T  
 23251 44515 IM  
 23252 42400 E"

23253 51505 SKTM: .TXT "SE  
 23254 42513 EK  
 23255 20124 T  
 23256 44515 IM  
 23257 42400 E"

23260 37061 GT10: .TXT ">1  
 23261 30040 0

23262 46523 MS  
 23263 0 "

23264 46501 SKMX: .TXT "MA  
 23265 54040 X  
 23266 30460 10  
 23267 20115 M  
 23270 51400 S"

23271 43117 FMTG: .TXT "FO  
 23272 51115 RM  
 23273 40524 AT  
 23274 52111 TI  
 23275 47107 NG  
 23276 0 "

23277 46523 MS: .TXT "MS  
 23300 0 "

23301 44523 SAR: .TXT "IS  
 23302 51525 SU  
 23303 42440 E  
 23304 51505 SE  
 23305 42513 EK  
 23306 20106 F  
 23307 47514 OL  
 23310 46117 LD  
 23311 53511 WI  
 23312 47107 NG  
 23313 20101 A  
 23314 20122 R  
 23315 42501 EA  
 23316 42000 D"

23317 41506 CF1: .TXT "CF  
 23320 20123 S  
 23321 52101 TA  
 23322 54523 YS  
 23323 20117 0  
 23324 47000 N"

23325 41506 CFN: .TXT "CF  
 23326 20104 D  
 23327 51117 RD  
 23330 50123 PS  
 23331 20124 T  
 23332 47440 0  
 23333 30040 0  
 23334 44516 IN  
 23335 20064 4  
 23336 30060 00  
 23337 20125 U  
 23340 51400 S"

23341 51505 SSDA: .TXT "SE  
 23342 52040 T  
 23343 31116 2N  
 23344 42040 D  
 23345 42122 DR

23346 44526 IV  
 23347 42440 E  
 23350 40524 AT  
 23351 52105 TE  
 23352 47000 N"

23353 52516 USRD: .TXT "UN  
 23354 51525 SU  
 23355 41503 CC  
 23356 42523 ES  
 23357 51506 SF  
 23360 52514 UL  
 23361 0 "

23362. 46501 IMA: .TXT "MA  
 23363 20122 R  
 23364 42507 EG  
 23365 20111 I  
 23366 47103 NC  
 23367 51105 RE  
 23370 46505 ME  
 23371 47124 NT  
 23372 20067 7  
 23373 33464 74  
 23374 30060 00  
 23375 20124 T  
 23376 47440 0  
 23377 30460 10  
 23400 30060 00  
 23401 30060 00  
 23402 0 "

23403 53522 WDIT: .TXT "WR  
 23404 44524 IT  
 23405 42440 E  
 23406 30440 1  
 23407 51505 SE  
 23410 41524 CT  
 23411 47522 OR  
 23412 20104 D  
 23413 40524 AT  
 23414 40400 A"

23415 51105 RDOV: .TXT "RE  
 23416 40504 AD  
 23417 20127 W  
 23420 44111 HI  
 23421 46105 LE  
 23422 20123 S  
 23423 42505 EE  
 23424 45440 K  
 23425 47526 OV  
 23426 42522 ER  
 23427 46101 LA  
 23430 50000 P"

23431 51105 RDBS: .TXT "RE  
 23432 40504 AD  
 23433 20104 D

23434 40524 AT  
 23435 40440 A  
 23436 43122 FR  
 23437 47515 OM  
 23440 20101 A  
 23441 20102 B  
 23442 40504 AD  
 23443 20123 S  
 23444 42503 EC  
 23445 52117 TD  
 23446 51000 R"

23447 41510 CKRB: .TXT "CH  
 23450 42503 EC  
 23451 45440 K  
 23452 51102 RB  
 23453 52506 UF  
 23454 43040 F  
 23455 41517 CD  
 23456 47124 NT  
 23457 42516 EN  
 23460 52123 TS  
 23461 0 "

23462 40502 ARDT: .TXT "AB  
 23463 47522 OR  
 23464 52040 T  
 23465 51105 RE  
 23466 40504 AD  
 23467 20101 A  
 23470 52124 TT  
 23471 42515 EM  
 23472 50124 PT  
 23473 0 "

23474 41510 CHHD: .TXT "CH  
 23475 40511 AI  
 23476 47040 N  
 23477 44105 HE  
 23500 40504 AD  
 23501 42522 ER  
 23502 20127 W  
 23503 47522 OR  
 23504 42123 DS  
 23505 0 "

23506 43122 FBCS: .TXT "FR  
 23507 47515 OM  
 23510 20103 C  
 23511 44101 HA  
 23512 44516 IN  
 23513 42504 ED  
 23514 20102 B  
 23515 40504 AD  
 23516 20123 S  
 23517 42503 EC  
 23520 52117 TD  
 23521 51000 R"



23522 51105 RDFC: .TXT "RE  
23523 40504 AD  
23524 20104 D  
23525 40524 AT  
23526 40440 A  
23527 43122 FR  
23530 47515 OM  
23531 20103 C  
23532 44101 HA  
23533 44516 IN  
23534 42504 ED  
23535 20123 S  
23536 42503 EC  
23537 52117 TO  
23540 51000 R"

23541 42101 NFCS: .TXT "DA  
23542 52101 TA  
23543 20116 N  
23544 47524 OT  
23545 20106 F  
23546 51117 RO  
23547 46440 M  
23550 41510 CH  
23551 40511 AI  
23552 47105 NE  
23553 42040 D  
23554 51505 SE  
23555 41524 CT  
23556 47522 OR  
23557 0 "

23560 51105 FCS: .TXT "RE  
23561 40504 AD  
23562 20106 F  
23563 51117 RO  
23564 46440 M  
23565 41510 CH  
23566 40511 AI  
23567 47105 NE  
23570 42040 D  
23571 51505 SE  
23572 41524 CT  
23573 47522 OR  
23574 0 "

23575 43101 FTA: .TXT "FA  
23576 44514 IL  
23577 42504 ED  
23600 20124 T  
23601 47440 O  
23602 40502 AB  
23603 47522 OR  
23604 52040 T  
23605 47520 OP  
23606 42522 ER  
23607 40524 AT  
23610 44517 IO  
23611 47000 N"

23612 53105 VARD: .TXT "VE  
23613 51111 RI  
23614 43131 FY  
23615 20101 A  
23616 41117 BO  
23617 51124 RT  
23620 20122 R  
23621 42501 EA  
23622 42040 D  
23623 40524 AT  
23624 52105 TE  
23625 46520 MP  
23626 52000 T"

23627 51105 RDDC: .TXT "RE  
23630 40504 AD  
23631 20106 F  
23632 51117 RO  
23633 46440 M  
23634 42117 DO  
23635 52502 UB  
23636 46105 LE  
23637 20103 C  
23640 44101 HA  
23641 44516 IN  
23642 42504 ED  
23643 20123 S  
23644 42503 EC  
23645 52117 TO  
23646 51000 R"

23647 40502 ABRT: .TXT "AB  
23650 47522 OR  
23651 52040 T  
23652 47520 OP  
23653 42522 ER  
23654 40524 AT  
23655 44517 IO  
23656 47000 N"

23657 41510 HPOS: .TXT "CH  
23660 42503 EC  
23661 45440 K  
23662 51127 RW  
23663 20110 H  
23664 42501 EA  
23665 42040 D  
23666 50117 PQ  
23667 51511 SI  
23670 52111 TI  
23671 47516 ON  
23672 0 "

23673 47117 NCYO: .TXT "NO  
23674 52040 T  
23675 47526 OV  
23676 42522 ER  
23677 20103 C

23700 54514 YL  
 23701 20060 0  
 23702 0 "

23703 47526 ACYO: .TXT "OV  
 23704 42522 ER  
 23705 20103 C  
 23706 54514 YL  
 23707 20060 0  
 23710 0 "

23711 47117 TNRD: .TXT "NO  
 23712 52040 T  
 23713 51105 RE  
 23714 40504 AD  
 23715 54400 Y"

23716 42515 EHDW: .TXT "EM  
 23717 52514 UL  
 23720 40524 AT  
 23721 47522 OR  
 23722 20110 H  
 23723 42501 EA  
 23724 42105 DE  
 23725 51040 R  
 23726 53517 WO  
 23727 51104 RD  
 23730 0 "

23731 52122 TPD: .TXT "TR  
 23732 42523 ES  
 23733 50101 PA  
 23734 51523 SS  
 23735 20104 D  
 23736 51111 RI  
 23737 53105 VE  
 23740 0 "

23741 51105 RELS: .TXT "RE  
 23742 46105 LE  
 23743 40523 AS  
 23744 42440 E  
 23745 42122 DR  
 23746 44526 IV  
 23747 42400 E"

23750 42122 RDYS: .TXT "DR  
 23751 44526 IV  
 23752 42440 E  
 23753 51105 RE  
 23754 40504 AD  
 23755 54440 Y  
 23756 51524 ST  
 23757 40524 AT  
 23760 52523 US  
 23761 0 "

23762 41517 SSCK: .TXT "CO

23763 47124 NT  
23764 51117 RO  
23765 46040 L  
23766 51524 ST  
23767 40524 AT  
23770 52523 US  
23771 20103 C  
23772 44105 HE  
23773 41513 CK  
23774 0 "

23775 53522 WPFT: .TXT "WR  
23776 44524 IT  
23777 42440 E  
24000 50122 PR  
24001 47524 OT  
24002 42503 EC  
24003 52105 TE  
24004 42040 D  
24005 25440 +  
24006 43101 FA  
24007 52514 UL  
24010 52040 T  
24011 51524 ST  
24012 40524 AT  
24013 52523 US  
24014 0 "

24015 44514 ILCS: .TXT "IL  
24016 46105 LE  
24017 43501 GA  
24020 46040 L  
24021 41531 CY  
24022 46111 LI  
24023 47104 ND  
24024 42522 ER  
24025 20123 S  
24026 52101 TA  
24027 52125 TU  
24030 51400 S"

24031 41510 HCAD: .TXT "CH  
24032 42503 EC  
24033 45440 K  
24034 41531 CY  
24035 46111 LI  
24036 47104 ND  
24037 42522 ER  
24040 20101 A  
24041 42104 DD  
24042 51040 R  
24043 53517 WO  
24044 51104 RD  
24045 0 "

24046 41510 CDER: .TXT "CH  
24047 42503 EC  
24050 45440 K  
24051 42122 DR

```

24052 53040 V
24053 42522 ER
24054 51040 R
24055 51524 ST
24056 40524 AT
24057 52523 US
24060 20102 B
24061 44524 IT
24062 0 "

24063 41510 CDRF: .TXT "CH
24064 42503 EC
24065 45440 K
24066 42122 DR
24067 53040 V
24070 43101 FA
24071 52514 UL
24072 52040 T
24073 51524 ST
24074 40524 AT
24075 52523 US
24076 20102 B
24077 44524 IT
24100 51400 S"

24101 41510 CSKE: .TXT "CH
24102 42503 EC
24103 45440 K
24104 44514 IL
24105 46105 LE
24106 43501 GA
24107 46040 L
24110 51505 SE
24111 42513 EK
24112 20101 A
24113 42104 DD
24114 51040 R
24115 51524 ST
24116 40524 AT
24117 52523 US
24120 20102 B
24121 44524 IT
24122 0 "

24123 41510 CDRV: .TXT "CH
24124 42503 EC
24125 45440 K
24126 42122 DR
24127 53040 V
24130 51105 RE
24131 51505 SE
24132 51126 RV
24133 42504 ED
24134 20123 S
24135 52101 TA
24136 52125 TU
24137 51440 S
24140 41111 BI
24141 52000 T"

```

24142 47506 OFFF: .TXT "OF  
24143 43123 FS  
24144 42524 ET  
24145 20106 F  
24146 47522 OR  
24147 53501 WA  
24150 51104 RD  
24151 0 "

24152 47506 OFFR: .TXT "OF  
24153 43123 FS  
24154 42524 ET  
24155 20122 R  
24156 42526 EV  
24157 42522 ER  
24160 51505 SE  
24161 0 "

;\*\*\*\* LOGIC TEST TITLES \*\*\*\*;

24162 51105 RRST: .TXT "RE  
24163 43511 GI  
24164 51524 ST  
24165 42522 ER  
24166 20122 R  
24167 42523 ES  
24170 42524 ET  
24171 0 "

24172 51105 RGWRT: .TXT "RE  
24173 43511 GI  
24174 51524 ST  
24175 42522 ER  
24176 20127 W  
24177 51111 RI  
24200 52105 TE  
24201 26522 -R  
24202 42501 EA  
24203 42000 D"

24204 41517 CFST: .TXT "CO  
24205 47124 NT  
24206 51117 RD  
24207 46040 L  
24210 43125 FU  
24211 46114 LL  
24212 20123 S  
24213 52101 TA  
24214 52125 TU  
24215 51400 S"

24216 42122 DRDNT: .TXT "DR  
24217 44526 IV  
24220 42440 E  
24221 42117 DO  
24222 47105 NE  
24223 20123 S

24224 42524 ET  
 24225 27522 <57>R  
 24226 42523 ES  
 24227 42524 ET  
 24230 0 "

24231 41514 CLDDT: .TXT "CL  
 24232 42501 EA  
 24233 51040 R  
 24234 42122 DR  
 24235 44526 IV  
 24236 42440 E  
 24237 42117 DO  
 24240 47105 NE  
 24241 0 "

24242 42530 XTMAT: .TXT "EX  
 24243 52105 TE  
 24244 47104 ND  
 24245 42504 ED  
 24246 20115 M  
 24247 42515 EM  
 24250 47522 OR  
 24251 54440 Y  
 24252 40504 AD  
 24253 42122 DR  
 24254 42523 ES  
 24255 51511 SI  
 24256 47107 NG  
 24257 0 "

24260 46505 MART: .TXT "ME  
 24261 46517 MO  
 24262 51131 RY  
 24263 20101 A  
 24264 42104 DD  
 24265 51105 RE  
 24266 51523 SS  
 24267 20122 R  
 24270 42507 EG  
 24271 44523 IS  
 24272 52105 TE  
 24273 51040 R  
 24274 53522 WR  
 24275 44524 IT  
 24276 42457 E<57>  
 24277 51105 RE  
 24300 40504 AD  
 24301 0 "

24302 51525 SSRT: .TXT "SU  
 24303 51106 RF  
 24304 27523 <57>S  
 24305 42503 EC  
 24306 52040 T  
 24307 40504 AD  
 24310 42122 DR  
 24311 42523 ES  
 24312 51440 S

24313 51105 RE  
 24314 43511 GI  
 24315 51524 ST  
 24316 42522 ER  
 24317 20127 W  
 24320 51111 RI  
 24321 52105 TE  
 24322 27522 <57>R  
 24323 42501 EA  
 24324 42000 D"

24325 41125 SRBYT: .TXT "BU  
 24326 51531 SY  
 24327 20106 F  
 24330 46101 LA  
 24331 43440 G  
 24332 51505 SE  
 24333 52057 T<57>  
 24334 51105 RE  
 24335 51505 SE  
 24336 52000 T"

24337 51105 RBNTT: .TXT "RE  
 24340 40504 AD  
 24341 20102 B  
 24342 52506 UF  
 24343 43105 FE  
 24344 51123 RS  
 24345 20050 (  
 24346 40514 AL  
 24347 46040 L  
 24350 47516 ON  
 24351 42523 ES  
 24352 20104 D  
 24353 40524 AT  
 24354 40451 A)  
 24355 0 "

24356 51105 RBZTT: .TXT "RE  
 24357 40504 AD  
 24360 20102 B  
 24361 52506 UF  
 24362 43105 FE  
 24363 51123 RS  
 24364 20050 (  
 24365 40514 AL  
 24366 46040 L  
 24367 55105 ZE  
 24370 51117 RO  
 24371 51440 S  
 24372 42101 DA  
 24373 52101 TA  
 24374 24400 )"

24375 50122 PROMT: .TXT "PR  
 24376 47515 OM  
 24377 20120 P  
 24400 51117 RO  
 24401 43522 GR



24402 40515 AM  
24403 46511 MI  
24404 47107 NG  
24405 0 "

;\*\*\*\* DRIVE EMULATOR TEST TITLES \*\*\*\*

24406 51105 RLDOT: .TXT "RE  
24407 46105 LE  
24410 40523 AS  
24411 42440 E  
24412 42122 DR  
24413 44526 IV  
24414 42400 E"

24415 42122 DDST: .TXT "DR  
24416 44526 IV  
24417 42440 E  
24420 42117 DO  
24421 47105 NE  
24422 20123 S  
24423 52101 TA  
24424 52125 TU  
24425 51400 S"

24426 51105 REHDT: .TXT "RE  
24427 40504 AD  
24430 20105 E  
24431 46525 MU  
24432 46101 LA  
24433 52117 TO  
24434 51040 R  
24435 44105 HE  
24436 40504 AD  
24437 42522 ER  
24440 0 "

24441 52122 TPRLT: .TXT "TR  
24442 42523 ES  
24443 50101 PA  
24444 51523 SS  
24445 20046 &  
24446 20122 R  
24447 42514 EL  
24450 42501 EA  
24451 51505 SE  
24452 20104 D  
24453 51111 RI  
24454 53105 VE  
24455 0 "

24456 42122 DSABT: .TXT "DR  
24457 44526 IV  
24460 42440 E  
24461 51524 ST  
24462 40524 AT  
24463 52523 US  
24464 20102 B

24465 44524 IT  
 24466 51400 S"

24467 43117 FRMST: . TXT "FO  
 24470 51115 RM  
 24471 40524 AT  
 24472 20046 &  
 24473 20122 R  
 24474 42501 EA  
 24475 42040 D  
 24476 43117 FO  
 24477 51115 RM  
 24500 40524 AT  
 24501 0 "

24502 53522 WRHDT: . TXT "WR  
 24503 44524 IT  
 24504 42440 E  
 24505 23040 &  
 24506 51105 RE  
 24507 40504 AD  
 24510 20110 H  
 24511 42501 EA  
 24512 42105 DE  
 24513 51000 R"

24514 30440 SSDTT: . TXT "1  
 24515 51505 SE  
 24516 41524 CT  
 24517 47522 OR  
 24520 20104 D  
 24521 40524 AT  
 24522 40440 A  
 24523 53522 WR  
 24524 44524 IT  
 24525 42454 E,  
 24526 20122 R  
 24527 42501 EA  
 24530 42040 D  
 24531 23040 &  
 24532 53105 VE  
 24533 51111 RI  
 24534 43131 FY  
 24535 0 "

24536 31040 TSDTT: . TXT "2  
 24537 51505 SE  
 24540 41524 CT  
 24541 47522 OR  
 24542 20104 D  
 24543 40524 AT  
 24544 40440 A  
 24545 53522 WR  
 24546 44524 IT  
 24547 42440 E  
 24550 23040 &  
 24551 51105 RE  
 24552 40504 AD  
 24553 0 "

24554 51105 ROFST: .TXT "RE  
24555 40504 AD  
24556 20117 O  
24557 43106 FF  
24560 51505 SE  
24561 52000 T"

;\*\*\*\* DRIVE TEST TITLES \*\*\*\*;

24562 51105 RCLT: .TXT "RE  
24563 41501 CA  
24564 46040 L  
24565 42122 DR  
24566 44526 IV  
24567 42400 E"

24570 43117 FM1T: .TXT "FO  
24571 51115 RM  
24572 40524 AT  
24573 20061 1  
24574 20123 S  
24575 42503 EC  
24576 52117 TO  
24577 51000 R"

24600 51105 RF1T: .TXT "RE  
24601 40504 AD  
24602 20061 1  
24603 20123 S  
24604 42503 EC  
24605 52117 TO  
24606 51040 R  
24607 43117 FO  
24610 51115 RM  
24611 40524 AT  
24612 0 "

24613 53522 WH1T: .TXT "WR  
24614 44524 IT  
24615 42440 E  
24616 30440 1  
24617 51505 SE  
24620 41524 CT  
24621 47522 OR  
24622 20110 H  
24623 42501 EA  
24624 42105 DE  
24625 51000 R"

24626 43117 FM2T: .TXT "FO  
24627 51115 RM  
24630 40524 AT  
24631 20117 O  
24632 47105 NE  
24633 20124 T  
24634 51101 RA  
24635 41513 CK

24636 0 "  
 24637 51105 RF2T: .TXT "RE  
 24640 40504 AD  
 24641 20106 F  
 24642 47522 OR  
 24643 46501 MA  
 24644 52124 TT  
 24645 42504 ED  
 24646 20124 T  
 24647 51101 RA  
 24650 41513 CK  
 24651 20110 H  
 24652 42501 EA  
 24653 42105 DE  
 24654 51123 RS  
 24655 0 "  
 24656 51105 RD1T: .TXT "RE  
 24657 40504 AD  
 24660 20061 1  
 24661 20123 S  
 24662 42503 EC  
 24663 52117 TO  
 24664 51040 R  
 24665 42101 DA  
 24666 52101 TA  
 24667 0 "  
 24670 46505 TOPT: .TXT "ME  
 24671 46440 M  
 24672 40504 AD  
 24673 42122 DR  
 24674 20122 R  
 24675 42507 EG  
 24676 20111 I  
 24677 47103 NC  
 24700 51105 RE  
 24701 46505 ME  
 24702 47124 NT  
 24703 0 "  
 24704 51127 RWTT: .TXT "RW  
 24705 20124 T  
 24706 44515 IM  
 24707 42517 EO  
 24710 52524 UT  
 24711 0 "  
 24712 51525 SSAT: .TXT "SU  
 24713 51106 RF  
 24714 27523 <57>S  
 24715 42503 EC  
 24716 52040 T  
 24717 40504 AD  
 24720 42122 DR  
 24721 42523 ES  
 24722 51440 S  
 24723 42522 ER

24724 51117 RO  
24725 51040 R  
24726 51524 ST  
24727 40524 AT  
24730 52523 US  
24731 0 "

24732 41531 CAET: .TXT "CY  
24733 46111 LI  
24734 47104 ND  
24735 42522 ER  
24736 20101 A  
24737 42104 DD  
24740 51105 RE  
24741 51523 SS  
24742 20105 E  
24743 51122 RR  
24744 47522 UR  
24745 20123 S  
24746 52101 TA  
24747 52125 TU  
24750 51400 S"

24751 52516 HUBT: .TXT "UN  
24752 52523 US  
24753 42504 ED  
24754 20110 H  
24755 42501 EA  
24756 42105 DE  
24757 51040 R  
24760 53517 WD  
24761 51104 RD  
24762 20102 B  
24763 44524 IT  
24764 51400 S"

24765 43117 FCCT: .TXT "FO  
24766 51115 RM  
24767 40524 AT  
24770 20103 C  
24771 51117 RO  
24772 51523 SS  
24773 20103 C  
24774 54514 YL  
24775 44516 IN  
24776 42105 DE  
24777 51000 R"

25000 43117 FCYLT: .TXT "FO  
25001 51115 RM  
25002 40524 AT  
25003 20101 A  
25004 20103 C  
25005 54514 YL  
25006 44516 IN  
25007 42105 DE  
25010 51000 R"

25011 43117 FMLST: .TXT "FO

25012 51115 RM  
25013 40524 AT  
25014 20114 L  
25015 40523 AS  
25016 52040 T  
25017 51105 RE  
25020 46517 MD  
25021 53101 VA  
25022 41114 BL  
25023 42440 E  
25024 51525 SU  
25025 51106 RF  
25026 40503 AC  
25027 42400 E"

25030 51105 RMLST: .TXT "RE  
25031 40504 AD  
25032 20114 L  
25033 40523 AS  
25034 52040 T  
25035 51505 SE  
25036 41524 CT  
25037 47522 OR  
25040 20106 F  
25041 47522 OR  
25042 46501 MA  
25043 52000 T"

25044 42101 VRFYT: .TXT "DA  
25045 52101 TA  
25046 20126 V  
25047 42522 ER  
25050 44506 IF  
25051 54440 Y  
25052 42522 ER  
25053 51117 RD  
25054 51040 R  
25055 51524 ST  
25056 40524 AT  
25057 52523 US  
25060 0 "

25061 51505 SECKT: .TXT "SE  
25062 41524 CT  
25063 47522 OR  
25064 20103 C  
25065 44105 HE  
25066 41513 CK  
25067 0 "

25070 53522 WACYT: .TXT "WR  
25071 44524 IT  
25072 42440 E  
25073 44105 HE  
25074 40504 AD  
25075 42522 ER  
25076 0 "

25077 47526 OVLPT: .TXT "OV

25100 42522 ER  
 25101 46101 LA  
 25102 50040 P  
 25103 51505 SE  
 25104 42513 EK  
 25105 0 "

25106 41101 BSSTT: .TXT "BA  
 25107 42040 D  
 25110 51505 SE  
 25111 41524 CT  
 25112 47522 OR  
 25113 20123 S  
 25114 52101 TA  
 25115 52125 TU  
 25116 51400 S"

25117 51505 CHNT: .TXT "SE  
 25120 41524 CT  
 25121 47522 OR  
 25122 20103 C  
 25123 44101 HA  
 25124 44516 IN  
 25125 44516 IN  
 25126 43400 G"

25127 42117 DCHNT: .TXT "DO  
 25130 52502 UB  
 25131 46105 LE  
 25132 20103 C  
 25133 44101 HA  
 25134 44516 IN  
 25135 20123 S  
 25136 42503 EC  
 25137 52117 TO  
 25140 51000 R"

25141 41510 CBSCT: .TXT "CH  
 25142 40511 AI  
 25143 47040 N  
 25144 41101 BA  
 25145 42040 D  
 25146 51505 SE  
 25147 41524 CT  
 25150 47522 OR  
 25151 0 "

25152 51105 RFFI: .TXT "RE  
 25153 40504 AD  
 25154 20106 F  
 25155 44506 IF  
 25156 47440 O  
 25157 41125 BU  
 25160 43106 FF  
 25161 42522 ER  
 25162 0 "

; \*\*\*\* DRIVE MNEMONIC TEXT \*\*\*\*

25163	41504	DN1:	. TXT	"CD
25164	41463	C3		
25165	31000	2"		
25166	41504	DN2:	. TXT	"CD
25167	41464	C4		
25170	34000	8"		
25171	41504	DN3:	. TXT	"CD
25172	41466	C6		
25173	32000	4"		
25174	41504	DN4:	. TXT	"CD
25175	41471	C9		
25176	33000	6"		
25177	41504	DN5:	. TXT	"CD
25200	41464	C4		
25201	30000	0"		
25202	41504	DN6:	. TXT	"CD
25203	41470	C8		
25204	30000	0"		
25205	41504	DN7:	. TXT	"CD
25206	41461	C1		
25207	32460	50		
25210	0	"		
25211	41504	DN8:	. TXT	"CD
25212	41463	C3		
25213	30060	00		
25214	0	"		
25215	41504	DN9:	. TXT	"CD
25216	41461	C1		
25217	31000	2"		
25220	41504	DN10:	. TXT	"CD
25221	41462	C2		
25222	32400	5"		
25223	41504	DN11:	. TXT	"CD
25224	41470	C8		
25225	31000	2"		
25226	41504	DN12:	. TXT	"CD
25227	41456	C.		
25230	34400	9"		
25231	41504	DN13:	. TXT	"CD
25232	41461	C1		
25233	0	"		
25234	41504	DN14:	. TXT	"CD
25235	41462	C2		
25236	0	"		



25237	41504	DN15:	.TXT	"CD
25240	41461	C1		
25241	33000	6"		
25242	41504	DN16:	.TXT	"CD
25243	41461	C1		
25244	33106	6F		
25245	0	"		
25246	41504	DN17:	.TXT	"CD
25247	41464	C4		
25250	34106	8F		
25251	0	"		
25252	41504	DN18:	.TXT	"CD
25253	41470	C8		
25254	30106	0F		
25255	0	"		
25256	0	DN19:	.TXT	"<0>"
25257	0	DN20:	.TXT	"<0>"
25260	0	DN21:	.TXT	"<0>"
25261	0	DN22:	.TXT	"<0>"
25262	0	DN23:	.TXT	"<0>"
25263	0	DN24:	.TXT	"<0>"
25264	40515	DN25:	.TXT	"AM
25265	50064	P4		
25266	30000	0"		
25267	40515	DN26:	.TXT	"AM
25270	50070	P8		
25271	30000	0"		
25272	40515	DN27:	.TXT	"AM
25273	50061	P1		
25274	30060	00		
25275	0	"		
25276	40515	DN28:	.TXT	"AM
25277	50061	P1		
25300	32460	50		
25301	0	"		
25302	40515	DN29:	.TXT	"AM
25303	50061	P1		
25304	33060	60		
25305	0	"		
25306	40515	DN30:	.TXT	"AM
25307	50062	P2		
25310	30060	00		
25311	0	"		

```

25312 40515 DN31: .TXT "AM
25313 50063 P3
25314 30060 00
25315 0 "

25316 40515 DN32: .TXT "AM
25317 50061 P1
25320 33000 6"

25321 40515 DN33: .TXT "AM
25322 50061 P1
25323 33106 6F
25324 0 "

25325 40515 DN34: .TXT "AM
25326 50064 P4
25327 34000 8"

25330 40515 DN35: .TXT "AM
25331 50070 P8
25332 30103 0C
25333 0 "

25334 0 DN36: .TXT "<0>"
25335 0 DN37: .TXT "<0>"
25336 0 DN38: .TXT "<0>"
25337 0 DN39: .TXT "<0>"

25340 46515 DN40: .TXT "MM
25341 54062 X2
25342 32400 5"

25343 46515 DN41: .TXT "MM
25344 54065 X5
25345 30000 0"

25346 46515 DN42: .TXT "MM
25347 54067 X7
25350 32400 5"

25351 46515 DN43: .TXT "MM
25352 54061 X1
25353 30060 00
25354 0 "

25355 46515 DN44: .TXT "MM
25356 54062 X2
25357 30060 00
25360 0 "

25361 0 DN45: .TXT "<0>"
25362 0 DN46: .TXT "<0>"
25363 0 DN47: .TXT "<0>"

```

25364	43125	DN48:	.TXT	"FU
25365	45064	J4		
25366	30000	0"		
25367	0	DN49:	.TXT	"<0>"
25370	47513	DN50:	.TXT	"OK
25371	44461	I1		
25372	31400	3"		
25373	47513	DN51:	.TXT	"OK
25374	44462	I2		
25375	33000	6"		
25376	47513	DN52:	.TXT	"OK
25377	44464	I4		
25400	30000	0"		
25401	47513	DN53:	.TXT	"OK
25402	44465	I5		
25403	30000	0"		
25404	47513	DN54:	.TXT	"OK
25405	44466	I6		
25406	33400	7"		
25407	47513	DN55:	.TXT	"OK
25410	44470	I8		
25411	30000	0"		
25412	0	DN56:	.TXT	"<0>"
25413	0	DN57:	.TXT	"<0>"
25414	0	DN58:	.TXT	"<0>"
25415	0	DN59:	.TXT	"<0>"
25416	41501	DN60:	.TXT	"CA
25417	46062	L2		
25420	32400	5"		
25421	41501	DN61:	.TXT	"CA
25422	46065	L5		
25423	30000	0"		
25424	41501	DN62:	.TXT	"CA
25425	46070	L8		
25426	30000	0"		
25427	41501	DN63:	.TXT	"CA
25430	46062	L2		
25431	30060	00		
25432	0	"		
25433	41501	DN64:	.TXT	"CA
25434	46063	L3		
25435	30060	00		

```

25436      0 "
25437      0 DN65: .TXT "<0>"
25440      0 DN66: .TXT "<0>"
25441      0 DN67: .TXT "<0>"
25442      0 DN68: .TXT "<0>"
25443      0 DN69: .TXT "<0>"
25444  46503 DN70: .TXT "MC
25445  42061 D1
25446  31000 2"
25447  46503 DN71: .TXT "MC
25450  42063 D3
25451  33400 7"
25452  46503 DN72: .TXT "MC
25453  42066 D6
25454  31000 2"
25455      0 DN73: .TXT "<0>"
25456      0 DN74: .TXT "<0>"
25457  45505 DN75: .TXT "KE
25460  47061 N1
25461  32000 4"
25462  45505 DN76: .TXT "KE
25463  47064 N4
25464  31000 2"
25465  45505 DN77: .TXT "KE
25466  47067 N7
25467  30000 0"
25470      0 DN78: .TXT "<0>"
25471      0 DN79: .TXT "<0>"
25472  5103 CDC: .TXT "<12>C
25473  42103 DC
25474  20104 D
25475  51111 RI
25476  53105 VE
25477  51400 S"
25500  5101 AMP: .TXT "<12>A
25501  46520 MP
25502  42530 EX
25503  20104 D
25504  51111 RI
25505  53105 VE
25506  51400 S"

```

25507 5115 MMX: .TXT "<12>M  
25510 42515 EM  
25511 47522 OR  
25512 42530 EX  
25513 20104 D  
25514 51111 RI  
25515 53105 VE  
25516 51400 S"  
  
25517 5117 OKI: .TXT "<12>O  
25520 45511 KI  
25521 42101 DA  
25522 52101 TA  
25523 20104 D  
25524 51111 RI  
25525 53105 VE  
25526 51400 S"  
  
25527 5106 FUJ: .TXT "<12>F  
25530 52512 UJ  
25531 44524 IT  
25532 51525 SU  
25533 20104 D  
25534 51111 RI  
25535 53105 VE  
25536 51400 S"  
  
25537 5103 CDD: .TXT "<12>C  
25540 42516 EN  
25541 52125 TU  
25542 51131 RY  
25543 20104 D  
25544 40524 AT  
25545 40440 A  
25546 42122 DR  
25547 44526 IV  
25550 42523 ES  
25551 0 "  
  
25552 5103 CAL: .TXT "<12>C  
25553 40514 AL  
25554 41517 CO  
25555 46520 MP  
25556 27503 <57>C  
25557 42516 EN  
25560 52125 TU  
25561 51131 RY  
25562 20104 D  
25563 40524 AT  
25564 40440 A  
25565 42122 DR  
25566 44526 IV  
25567 42523 ES  
25570 0 "  
  
25571 5113 KEN: .TXT "<12>K  
25572 42516 EN  
25573 47105 NE

25574 42131 DY  
25575 20104 D  
25576 51111 RI  
25577 53105 VE  
25600 51400 S"

25601 5115 MCD: .TXT "<12>M  
25602 44503 IC  
25603 51117 RD  
25604 42101 DA  
25605 52101 TA  
25606 20104 D  
25607 51111 RI  
25610 53105 VE  
25611 51400 S"

;\*\*\*\* DRIVE CHARACTERISTICS TEXT \*\*\*\*

25612 41517 HLN1: .TXT "CO  
25613 42105 DE  
25614 20040  
25615 46516 MN  
25616 42515 EM  
25617 47516 ON  
25620 44503 IC  
25621 20040  
25622 52131 TY  
25623 50105 PE  
25624 20040  
25625 41501 CA  
25626 50101 PA  
25627 41511 CI  
25630 52131 TY  
25631 20040  
25632 41531 CY  
25633 46047 L'  
25634 51440 S  
25635 20110 H  
25636 42501 EA  
25637 42123 DS  
25640 20040  
25641 51505 SE  
25642 41524 CT  
25643 23523 'S  
25644 20040  
25645 46517 MO  
25646 42105 DE  
25647 46012 L<12>  
25650 0 "

25651 20040 DTX1: .TXT "  
25652 30440 1  
25653 20040  
25654 20103 C  
25655 42103 DC  
25656 31462 32  
25657 20040

25660 20040  
 25661 41515 CM  
 25662 42040 D  
 25663 20040  
 25664 20040  
 25665 31462 32  
 25666 46502 MB  
 25667 20040  
 25670 20040  
 25671 20070 8  
 25672 31063 23  
 25673 20040  
 25674 20040  
 25675 20062 2  
 25676 20040  
 25677 20040  
 25700 20040  
 25701 31462 32  
 25702 20040  
 25703 20040  
 25704 34464 94  
 25705 32070 48  
 25706 0 "

25707 20040 DTX3: .TXT "  
 25710 31440 3  
 25711 20040  
 25712 20103 C  
 25713 42103 DC  
 25714 33064 64  
 25715 20040  
 25716 20040  
 25717 41515 CM  
 25720 42040 D  
 25721 20040  
 25722 20040  
 25723 33064 64  
 25724 46502 MB  
 25725 20040  
 25726 20040  
 25727 20070 8  
 25730 31063 23  
 25731 20040  
 25732 20040  
 25733 20064 4  
 25734 20040  
 25735 20040  
 25736 20040  
 25737 31462 32  
 25740 20040  
 25741 20040  
 25742 34464 94  
 25743 32070 48  
 25744 0 "

25745 20040 DTX4: .TXT "  
 25746 32040 4  
 25747 20040  
 25750 20103 C

25751 42103 DC  
 25752 34466 96  
 25753 20040  
 25754 20040  
 25755 41515 CM  
 25756 42040 D  
 25757 20040  
 25760 20040  
 25761 34466 96  
 25762 46502 MB  
 25763 20040  
 25764 20040  
 25765 20070 8  
 25766 31063 23  
 25767 20040  
 25770 20040  
 25771 20066 6  
 25772 20040  
 25773 20040  
 25774 20040  
 25775 31462 32  
 25776 20040  
 25777 20040  
 26000 34464 94  
 26001 32070 48  
 26002 0 "  
  
 26003 20040 DTX5: .TXT "  
 26004 32440 5  
 26005 20040  
 26006 20103 C  
 26007 42103 DC  
 26010 32060 40  
 26011 20040  
 26012 20040  
 26013 51515 SM  
 26014 42040 D  
 26015 20040  
 26016 20040  
 26017 32060 40  
 26020 46502 MB  
 26021 20040  
 26022 20040  
 26023 20064 4  
 26024 30461 11  
 26025 20040  
 26026 20040  
 26027 20065 5  
 26030 20040  
 26031 20040  
 26032 20040  
 26033 31462 32  
 26034 0 "  
  
 26035 20040 DTX6: .TXT "  
 26036 33040 6  
 26037 20040  
 26040 20103 C  
 26041 42103 DC



26042 34060 80  
 26043 20040  
 26044 20040  
 26045 51515 SM  
 26046 42040 D  
 26047 20040  
 26050 20040  
 26051 34060 80  
 26052 46502 MB  
 26053 20040  
 26054 20040  
 26055 20070 8  
 26056 31063 23  
 26057 20040  
 26060 20040  
 26061 20065 5  
 26062 20040  
 26063 20040  
 26064 20040  
 26065 31462 32  
 26066 20040  
 26067 20040  
 26070 34467 97  
 26071 33062 62  
 26072 0 "

26073 20040 DTX7: .TXT "  
 26074 33440 7  
 26075 20040  
 26076 20103 C  
 26077 42103 DC  
 26100 30465 15  
 26101 30040 0  
 26102 20040  
 26103 51515 SM  
 26104 42040 D  
 26105 20040  
 26106 20061 1  
 26107 32460 50  
 26110 46502 MB  
 26111 20040  
 26112 20040  
 26113 20064 4  
 26114 30461 11  
 26115 20040  
 26116 20040  
 26117 30471 19  
 26120 20040  
 26121 20040  
 26122 20040  
 26123 31462 32  
 26124 0 "

26125 20040 DTX8: .TXT "  
 26126 34040 8  
 26127 20040  
 26130 20103 C  
 26131 42103 DC  
 26132 31460 30

26133 30040 0  
 26134 20040  
 26135 51515 SM  
 26136 42040 D  
 26137 20040  
 26140 20063 3  
 26141 30060 00  
 26142 46502 MB  
 26143 20040  
 26144 20040  
 26145 20070 8  
 26146 31063 23  
 26147 20040  
 26150 20040  
 26151 30471 19  
 26152 20040  
 26153 20040  
 26154 20040  
 26155 31462 32  
 26156 20040  
 26157 20040  
 26160 34467 97  
 26161 33066 66  
 26162 0 "

26163 20040 DTX9: .TXT "

26164 34440 9  
 26165 20040  
 26166 20103 C  
 26167 42103 DC  
 26170 30462 12  
 26171 20040  
 26172 20040  
 26173 46515 MM  
 26174 42040 D  
 26175 20040  
 26176 20040  
 26177 30462 12  
 26200 46502 MB  
 26201 20040  
 26202 20040  
 26203 20063 3  
 26204 31060 20  
 26205 20040  
 26206 20040  
 26207 20062 2  
 26210 20040  
 26211 20040  
 26212 20040  
 26213 31462 32  
 26214 0 "

26215 20061 DTX10: .TXT " 1

26216 30040 0  
 26217 20040  
 26220 20103 C  
 26221 42103 DC  
 26222 31065 25  
 26223 20040

```

26224 20040
26225 46515 MM
26226 42040 D
26227 20040
26230 20040
26231 31065 25
26232 46502 MB
26233 20040
26234 20040
26235 20063 3
26236 31060 20
26237 20040
26240 20040
26241 20064 4
26242 20040
26243 20040
26244 20040
26245 31462 32
26246 0 "

26247 20061 DTX11:.TXT " 1
26250 30440 1
26251 20040
26252 20103 C
26253 42103 DC
26254 34062 82
26255 20040
26256 20040
26257 46515 MM
26260 42040 D
26261 20040
26262 20040
26263 34062 82
26264 46502 MB
26265 20040
26266 20040
26267 20070 8
26270 31063 23
26271 20040
26272 20040
26273 20065 5
26274 20040
26275 20040
26276 20040
26277 31462 32
26300 0 "

26301 20061 DTX12:.TXT " 1
26302 31040 2
26303 20040
26304 20103 C
26305 42103 DC
26306 27071 9
26307 20040
26310 20040
26311 46515 MM
26312 42040 D
26313 20040
26314 20060 0

```



26315 27071 .9  
26316 46502 MB  
26317 20040  
26320 20040  
26321 20040  
26322 30462 12  
26323 20040  
26324 20040  
26325 20064 4  
26326 20040  
26327 20040  
26330 20040  
26331 31462 32  
26332 0 "

26333 20061 DTX13: .TXT " 1  
26334 31440 3  
26335 20040  
26336 20103 C  
26337 42103 DC  
26340 30440 1  
26341 20040  
26342 20040  
26343 46515 MM  
26344 42040 D  
26345 20040  
26346 20040  
26347 20061 1  
26350 46502 MB  
26351 20040  
26352 20040  
26353 20040  
26354 30460 10  
26355 20040  
26356 20040  
26357 20065 5  
26360 20040  
26361 20040  
26362 20040  
26363 31462 32  
26364 0 "

26365 20061 DTX14: .TXT " 1  
26366 32040 4  
26367 20040  
26370 20103 C  
26371 42103 DC  
26372 31040 2  
26373 20040  
26374 20040  
26375 46515 MM  
26376 42040 D  
26377 20040  
26400 20040  
26401 20062 2  
26402 46502 MB  
26403 20040  
26404 20040  
26405 20040

282 / 283



26406 31060 20  
26407 20040  
26410 20040  
26411 20065 5  
26412 20040  
26413 20040  
26414 20040  
26415 31462 32  
26416 0 "

26417 20061 DTX15: .TXT " 1  
26420 32440 5  
26421 20040  
26422 20103 C  
26423 42103 DC  
26424 30466 16  
26425 20040  
26426 20040  
26427 41515 CM  
26430 42040 D  
26431 20040  
26432 20040  
26433 30466 16  
26434 46502 MB  
26435 20040  
26436 20040  
26437 20070 8  
26440 30070 08  
26441 20040  
26442 20040  
26443 20061 1  
26444 20040  
26445 20040  
26446 20040  
26447 31462 32  
26450 0 "

26451 20061 DTX16: .TXT " 1  
26452 33040 6  
26453 20040  
26454 20103 C  
26455 42103 DC  
26456 30466 16  
26457 43040 F  
26460 20040  
26461 43115 FM  
26462 42040 D  
26463 20040  
26464 20040  
26465 30466 16  
26466 46502 MB  
26467 20040  
26470 20040  
26471 20070 8  
26472 30070 08  
26473 20040  
26474 20040  
26475 20061 1  
26476 20040





26477 20040  
 26500 20040  
 26501 31462 32  
 26502 0 "

26503 20061 DTX17: .TXT " 1  
 26504 33440 7  
 26505 20040  
 26506 20103 C  
 26507 42103 DC  
 26510 32070 48  
 26511 43040 F  
 26512 20040  
 26513 43115 FM  
 26514 42040 D  
 26515 20040  
 26516 20040  
 26517 32070 48  
 26520 46502 MB  
 26521 20040  
 26522 20040  
 26523 20070 8  
 26524 30070 08  
 26525 20040  
 26526 20040  
 26527 20063 3  
 26530 20040  
 26531 20040  
 26532 20040  
 26533 31462 32  
 26534 0 "

26535 20061 DTX18: .TXT " 1  
 26536 34040 8  
 26537 20040  
 26540 20103 C  
 26541 42103 DC  
 26542 34060 80  
 26543 43040 F  
 26544 20040  
 26545 43115 FM  
 26546 42040 D  
 26547 20040  
 26550 20040  
 26551 34060 80  
 26552 46502 MB  
 26553 20040  
 26554 20040  
 26555 20070 8  
 26556 30070 08  
 26557 20040  
 26560 20040  
 26561 20065 5  
 26562 20040  
 26563 20040  
 26564 20040  
 26565 31462 32  
 26566 0 "

26567 20062 DTX25: .TXT " 2  
26570 32440 5  
26571 20040  
26572 20101 A  
26573 46520 MP  
26574 32060 40  
26575 20040  
26576 20040  
26577 51515 SM  
26600 42040 D  
26601 20040  
26602 20040  
26603 32060 40  
26604 46502 MB  
26605 20040  
26606 20040  
26607 20064 4  
26610 30461 11  
26611 20040  
26612 20040  
26613 20065 5  
26614 20040  
26615 20040  
26616 20040  
26617 31462 32  
26620 0 "

26621 20062 DTX26: .TXT " 2  
26622 33040 6  
26623 20040  
26624 20101 A  
26625 46520 MP  
26626 34060 80  
26627 20040  
26630 20040  
26631 51515 SM  
26632 42040 D  
26633 20040  
26634 20040  
26635 34060 80  
26636 46502 MB  
26637 20040  
26640 20040  
26641 20070 8  
26642 31063 23  
26643 20040  
26644 20040  
26645 20065 5  
26646 20040  
26647 20040  
26650 20040  
26651 31462 32  
26652 20040  
26653 20040  
26654 42115 DM  
26655 26471 -9  
26656 34060 80  
26657 20000 "

26660 20062 DTX27: .TXT " 2  
26661 33440 7  
26662 20040  
26663 20101 A  
26664 46520 MP  
26665 30460 10  
26666 30040 0  
26667 20040  
26670 51515 SM  
26671 42040 D  
26672 20040  
26673 20061 1  
26674 30060 00  
26675 46502 MB  
26676 20040  
26677 20040  
26700 20064 4  
26701 30461 11  
26702 20040  
26703 20040  
26704 30471 19  
26705 20040  
26706 20040  
26707 20040  
26710 31061 21  
26711 0 "

26712 20062 DTX28: .TXT " 2  
26713 34040 8  
26714 20040  
26715 20101 A  
26716 46520 MP  
26717 30465 15  
26720 30040 0  
26721 20040  
26722 51515 SM  
26723 42040 D  
26724 20040  
26725 20061 1  
26726 32460 50  
26727 46502 MB  
26730 20040  
26731 20040  
26732 20065 5  
26733 33061 61  
26734 20040  
26735 20040  
26736 30464 14  
26737 20040  
26740 20040  
26741 20040  
26742 31462 32  
26743 0 "

26744 20062 DTX29: .TXT " 2  
26745 34440 9  
26746 20040  
26747 20101 A  
26750 46520 MP

26751	30466	16	
26752	30040	0	
26753	20040		
26754	51515	SM	
26755	42040	D	
26756	20040		
26757	20061	1	
26760	33060	60	
26761	46502	MB	
26762	20040		
26763	20040		
26764	30466	16	
26765	32065	45	
26766	20040		
26767	20040		
26770	20065	5	
26771	20040		
26772	20040		
26773	20040		
26774	31462	32	
26775	20040		
26776	20040		
26777	42115	DM	
27000	34461	91	
27001	33060	60	
27002	0	"	
27003	20063	DTX30: .TXT	" 3
27004	30040	0	
27005	20040		
27006	20101	A	
27007	46520	MP	
27010	31060	20	
27011	30040	0	
27012	20040		
27013	51515	SM	
27014	42040	D	
27015	20040		
27016	20062	2	
27017	30060	00	
27020	46502	MB	
27021	20040		
27022	20040		
27023	20070	8	
27024	30465	15	
27025	20040		
27026	20040		
27027	30471	19	
27030	20040		
27031	20040		
27032	20040		
27033	31061	21	
27034	0	"	
27035	20063	DTX31: .TXT	" 3
27036	30440	1	
27037	20040		
27040	20101	A	
27041	46520	MP	

27042	31460	30		
27043	30040	0		
27044	20040			
27045	51515	SM		
27046	42040	D		
27047	20040			
27050	20063	3		
27051	30060	00		
27052	46502	MB		
27053	20040			
27054	20040			
27055	20070	8		
27056	30465	15		
27057	20040			
27060	20040			
27061	30471	19		
27062	20040			
27063	20040			
27064	20040			
27065	31462	32		
27066	20040			
27067	20040			
27070	42115	DM		
27071	34463	93		
27072	30060	00		
27073	0	"		
27074	20063	DTX32: .TXT	"	3
27075	31040	2		
27076	20040			
27077	20101	A		
27100	46520	MP		
27101	30466	16		
27102	20040			
27103	20040			
27104	41515	CM		
27105	42040	D		
27106	20040			
27107	20040			
27110	30466	16		
27111	46502	MB		
27112	20040			
27113	20040			
27114	20070	8		
27115	31063	23		
27116	20040			
27117	20040			
27120	20061	1		
27121	20040			
27122	20040			
27123	20040			
27124	31462	32		
27125	0	"		
27126	20063	DTX33: .TXT	"	3
27127	31440	3		
27130	20040			
27131	20101	A		
27132	46520	MP		

27133 30466 16  
 27134 43040 F  
 27135 20040  
 27136 43115 FM  
 27137 42040 D  
 27140 20040  
 27141 20040  
 27142 30466 16  
 27143 46502 MB  
 27144 20040  
 27145 20040  
 27146 20070 8  
 27147 31063 23  
 27150 20040  
 27151 20040  
 27152 20061 1  
 27153 20040  
 27154 20040  
 27155 20040  
 27156 31462 32  
 27157 0 "

27160 20063 DTX34: .TXT " 3  
 27161 32040 4  
 27162 20040  
 27163 20101 A  
 27164 46520 MP  
 27165 32070 48  
 27166 20040  
 27167 20040  
 27170 43115 FM  
 27171 42040 D  
 27172 20040  
 27173 20040  
 27174 32070 48  
 27175 46502 MB  
 27176 20040  
 27177 20040  
 27200 20070 8  
 27201 31063 23  
 27202 20040  
 27203 20040  
 27204 20063 3  
 27205 20040  
 27206 20040  
 27207 20040  
 27210 31462 32  
 27211 0 "

27212 20063 DTX35: .TXT " 3  
 27213 32440 5  
 27214 20040  
 27215 20101 A  
 27216 46520 MP  
 27217 34060 80  
 27220 43040 F  
 27221 20040  
 27222 43115 FM  
 27223 42040 D

27224 20040  
 27225 20040  
 27226 34060 80  
 27227 46502 MB  
 27230 20040  
 27231 20040  
 27232 20070 8  
 27233 31063 23  
 27234 20040  
 27235 20040  
 27236 20065 5  
 27237 20040  
 27240 20040  
 27241 20040  
 27242 31462 32  
 27243 0 "

27244 20064 DTX40: .TXT " 4  
 27245 30040 0  
 27246 20040  
 27247 20115 M  
 27250 46530 MX  
 27251 31065 25  
 27252 20040  
 27253 20040  
 27254 51515 SM  
 27255 42040 D  
 27256 20040  
 27257 20040  
 27260 31065 25  
 27261 46502 MB  
 27262 20040  
 27263 20040  
 27264 20063 3  
 27265 32460 50  
 27266 20040  
 27267 20040  
 27270 20064 4  
 27271 20040  
 27272 20040  
 27273 20040  
 27274 31070 28  
 27275 20040  
 27276 20040  
 27277 33060 60  
 27300 30400 1"

27301 20064 DTX41: .TXT " 4  
 27302 30440 1  
 27303 20040  
 27304 20115 M  
 27305 46530 MX  
 27306 32460 50  
 27307 20040  
 27310 20040  
 27311 51515 SM  
 27312 42040 D  
 27313 20040  
 27314 20040

27315 32460 50  
27316 46502 MB  
27317 20040  
27320 20040  
27321 20063 3  
27322 32460 50  
27323 20040  
27324 20040  
27325 20070 8  
27326 20040  
27327 20040  
27330 20040  
27331 31070 28  
27332 20040  
27333 20040  
27334 33060 60  
27335 30400 1"  
  
27336 20064 DTX42: .TXT " 4  
27337 31040 2  
27340 20040  
27341 20115 M  
27342 46530 MX  
27343 33465 75  
27344 20040  
27345 20040  
27346 51515 SM  
27347 42040 D  
27350 20040  
27351 20040  
27352 33465 75  
27353 46502 MB  
27354 20040  
27355 20040  
27356 20063 3  
27357 32460 50  
27360 20040  
27361 20040  
27362 30462 12  
27363 20040  
27364 20040  
27365 20040  
27366 31070 28  
27367 20040  
27370 20040  
27371 33060 60  
27372 30400 1"  
  
27373 20064 DTX43: .TXT " 4  
27374 31440 3  
27375 20040  
27376 20115 M  
27377 46530 MX  
27400 30460 10  
27401 30040 0  
27402 20040  
27403 51515 SM  
27404 42040 D  
27405 20040



27406 20061 1  
27407 30060 00  
27410 46502 MB  
27411 20040  
27412 20040  
27413 20064 4  
27414 30461 11  
27415 20040  
27416 20040  
27417 30471 19  
27420 20040  
27421 20040  
27422 20040  
27423 31061 21  
27424 20040  
27425 20040  
27426 33067 67  
27427 33400 7"  
  
27430 20064 DTX44: .TXT " 4  
27431 32040 4  
27432 20040  
27433 20115 M  
27434 46530 MX  
27435 31060 20  
27436 30040 0  
27437 20040  
27440 51515 SM  
27441 42040 D  
27442 20040  
27443 20062 2  
27444 30060 00  
27445 46502 MB  
27446 20040  
27447 20040  
27450 20070 8  
27451 30465 15  
27452 20040  
27453 20040  
27454 30471 19  
27455 20040  
27456 20040  
27457 20040  
27460 31061 21  
27461 20040  
27462 20040  
27463 33067 67  
27464 33400 7"  
  
27465 20064 DTX48: .TXT " 4  
27466 34040 8  
27467 20040  
27470 20106 F  
27471 52512 UJ  
27472 32060 40  
27473 20040  
27474 20040  
27475 51515 SM  
27476 42040 D

```

-----
27477 20040
27500 20040
27501 32060 40
27502 46502 MB
27503 20040
27504 20040
27505 20070 8
27506 30465 15
27507 20040
27510 20040
27511 20063 3
27512 20040
27513 20040
27514 20040
27515 31462 32
27516 0 "

27517 20065 DTX50:.TXT " 5
27520 30040 0
27521 20040
27522 20117 0
27523 45511 KI
27524 30463 13
27525 20040
27526 20040
27527 20040
27530 20040
27531 20040
27532 20040
27533 30463 13
27534 46502 MB
27535 20040
27536 20040
27537 20063 3
27540 31471 39
27541 20040
27542 20040
27543 20062 2
27544 20040
27545 20040
27546 20040
27547 31462 32
27550 20040
27551 20040
27552 31463 33
27553 30061 01
27554 0 "

27555 20065 DTX51:.TXT " 5
27556 30440 1
27557 20040
27560 20117 0
27561 45511 KI
27562 31066 26
27563 20040
27564 20040
27565 20040
27566 20040
27567 20040

```

```

27570 20040
27571 31066 26
27572 46502 MB
27573 20040
27574 20040
27575 20063 3
27576 31471 39
27577 20040
27600 20040
27601 20064 4
27602 20040
27603 20040
27604 20040
27605 31462 32
27606 20040
27607 20040
27610 31463 33
27611 30062 02
27612 0 "

27613 20065 DTX52: .TXT " 5
27614 31040 2
27615 20040
27616 20117 0
27617 45511 KI
27620 32060 40
27621 20040
27622 20040
27623 20040
27624 20040
27625 20040
27626 20040
27627 32060 40
27630 46502 MB
27631 20040
27632 20040
27633 20063 3
27634 31471 39
27635 20040
27636 20040
27637 20066 6
27640 20040
27641 20040
27642 20040
27643 31462 32
27644 20040
27645 20040
27646 31463 33
27647 30063 03
27650 0 "

27651 20065 DTX53: .TXT " 5
27652 31440 3
27653 20040
27654 20117 0
27655 45511 KI
27656 32460 50
27657 20040
27660 20040

```

27661	20040		
27662	20040		
27663	20040		
27664	20040		
27665	32460	50	
27666	46502	MB	
27667	20040		
27670	20040		
27671	20063	3	
27672	31471	39	
27673	20040		
27674	20040		
27675	20070	8	
27676	20040		
27677	20040		
27700	20040		
27701	31462	32	
27702	20040		
27703	20040		
27704	31463	33	
27705	30064	04	
27706	0	"	
27707	20065	DTX54: .TXT	" 5
27710	32040	4	
27711	20040		
27712	20117	0	
27713	45511	KI	
27714	33067	67	
27715	20040		
27716	20040		
27717	20040		
27720	20040		
27721	20040		
27722	20040		
27723	33067	67	
27724	46502	MB	
27725	20040		
27726	20040		
27727	20063	3	
27730	31471	39	
27731	20040		
27732	20040		
27733	30460	10	
27734	20040		
27735	20040		
27736	20040		
27737	31462	32	
27740	20040		
27741	20040		
27742	31463	33	
27743	30065	05	
27744	0	"	
27745	20065	DTX55: .TXT	" 5
27746	32440	5	
27747	20040		
27750	20117	0	
27751	45511	KI	

27752 34060 B0  
 27753 20040  
 27754 20040  
 27755 43115 FM  
 27756 42040 D  
 27757 20040  
 27760 20040  
 27761 34060 B0  
 27762 46502 MB  
 27763 20040  
 27764 20040  
 27765 20063 3  
 27766 31471 39  
 27767 20040  
 27770 20040  
 27771 30462 12  
 27772 20040  
 27773 20040  
 27774 20040  
 27775 31462 32  
 27776 20040  
 27777 20040  
 30000 31463 33  
 30001 30066 06  
 30002 0 "

30003 20066 DTX60: .TXT " 6  
 30004 30040 0  
 30005 20040  
 30006 20103 C  
 30007 40514 AL  
 30010 31065 25  
 30011 20040  
 30012 20040  
 30013 20040  
 30014 20040  
 30015 20040  
 30016 20040  
 30017 31065 25  
 30020 46502 MB  
 30021 20040  
 30022 20040  
 30023 20064 4  
 30024 30067 07  
 30025 20040  
 30026 20040  
 30027 20065 5  
 30030 20040  
 30031 20040  
 30032 20040  
 30033 31063 23  
 30034 20040  
 30035 20040  
 30036 52062 T2  
 30037 32400 5"

30040 20066 DTX61: .TXT " 6  
 30041 30440 1  
 30042 20040

-----  
30043 20103 C  
30044 40514 AL  
30045 32460 50  
30046 20040  
30047 20040  
30050 20040  
30051 20040  
30052 20040  
30053 20040  
30054 32460 50  
30055 46502 MB  
30056 20040  
30057 20040  
30060 20070 8  
30061 30465 15  
30062 20040  
30063 20040  
30064 20065 5  
30065 20040  
30066 20040  
30067 20040  
30070 31063 23  
30071 20040  
30072 20040  
30073 52065 T5  
30074 30000 0"  
  
30075 20066 DTX62: .TXT " 6  
30076 31040 2  
30077 20040  
30100 20103 C  
30101 40514 AL  
30102 34060 80  
30103 20040  
30104 20040  
30105 20040  
30106 20040  
30107 20040  
30110 20040  
30111 34060 80  
30112 46502 MB  
30113 20040  
30114 20040  
30115 20070 8  
30116 30465 15  
30117 20040  
30120 20040  
30121 20065 5  
30122 20040  
30123 20040  
30124 20040  
30125 31462 32  
30126 20040  
30127 20040  
30130 52070 T8  
30131 31000 2"  
  
30132 20066 DTX63: .TXT " 6  
30133 31440 3

30134 20040  
30135 20103 C  
30136 40514 AL  
30137 31060 20  
30140 30040 0  
30141 20040  
30142 20040  
30143 20040  
30144 20040  
30145 20062 2  
30146 30060 00  
30147 46502 MB  
30150 20040  
30151 20040  
30152 20070 8  
30153 30465 15  
30154 20040  
30155 20040  
30156 30471 19  
30157 20040  
30160 20040  
30161 20040  
30162 31063 23  
30163 20040  
30164 20040  
30165 52062 T2  
30166 30060 00  
30167 0 "  
  
30170 20066 DTX64: .TXT " 6  
30171 32040 4  
30172 20040  
30173 20103 C  
30174 40514 AL  
30175 31460 30  
30176 30040 0  
30177 20040  
30200 20040  
30201 20040  
30202 20040  
30203 20063 3  
30204 30060 00  
30205 46502 MB  
30206 20040  
30207 20040  
30210 20070 8  
30211 30465 15  
30212 20040  
30213 20040  
30214 30471 19  
30215 20040  
30216 20040  
30217 20040  
30220 31462 32  
30221 20040  
30222 20040  
30223 52063 T3  
30224 30062 02  
30225 0 "

30226 20067 DTX70: .TXT " 7  
30227 30040 0  
30230 20040  
30231 20115 M  
30232 41504 CD  
30233 30462 12  
30234 20040  
30235 20040  
30236 20040  
30237 20040  
30240 20040  
30241 20040  
30242 30462 12  
30243 46502 MB  
30244 20040  
30245 20040  
30246 20063 3  
30247 32460 50  
30250 20040  
30251 20040  
30252 20062 2  
30253 20040  
30254 20040  
30255 20040  
30256 31070 28  
30257 0 "

30260 20067 DTX71: .TXT " 7  
30261 30440 1  
30262 20040  
30263 20115 M  
30264 41504 CD  
30265 31467 37  
30266 20040  
30267 20040  
30270 20040  
30271 20040  
30272 20040  
30273 20040  
30274 31467 37  
30275 46502 MB  
30276 20040  
30277 20040  
30300 20063 3  
30301 32460 50  
30302 20040  
30303 20040  
30304 20066 6  
30305 20040  
30306 20040  
30307 20040  
30310 31070 28  
30311 0 "

30312 20067 DTX72: .TXT " 7  
30313 31040 2  
30314 20040  
30315 20115 M



30316	41504	CD	
30317	33062	62	
30320	20040		
30321	20040		
30322	20040		
30323	20040		
30324	20040		
30325	20040		
30326	33062	62	
30327	46502	MB	
30330	20040		
30331	20040		
30332	20063	3	
30333	32460	50	
30334	20040		
30335	20040		
30336	30460	10	
30337	20040		
30340	20040		
30341	20040		
30342	31070	28	
30343	0	"	
30344	20067	DTX75: .TXT	" 7
30345	32440	5	
30346	20040		
30347	20113	K	
30350	42516	EN	
30351	30464	14	
30352	20040		
30353	20040		
30354	20040		
30355	20040		
30356	20040		
30357	20040		
30360	30464	14	
30361	46502	MB	
30362	20040		
30363	20040		
30364	20067	7	
30365	30060	00	
30366	20040		
30367	20040		
30370	20061	1	
30371	20040		
30372	20040		
30373	20040		
30374	31462	32	
30375	0	"	
30376	20067	DTX76: .TXT	" 7
30377	33040	6	
30400	20040		
30401	20113	K	
30402	42516	EN	
30403	32062	42	
30404	20040		
30405	20040		
30406	20040		

```

30407 20040
30410 20040
30411 20040
30412 32062 42
30413 46502 MB
30414 20040
30415 20040
30416 20067 7
30417 30060 00
30420 20040
30421 20040
30422 20063 3
30423 20040
30424 20040
30425 20040
30426 31462 32
30427 0 "

30430 20067 DTX77:.TXT " 7
30431 33440 7
30432 20040
30433 20113 K
30434 42516 EN
30435 33460 70
30436 20040
30437 20040
30440 20040
30441 20040
30442 20040
30443 20040
30444 33460 70
30445 46502 MB
30446 20040
30447 20040
30450 20067 7
30451 30060 00
30452 20040
30453 20040
30454 20065 5
30455 20040
30456 20040
30457 20040
30460 31462 32
30461 0 "

```

\*\*\*\* WRITE-READ BUFFER AREAS \*\*\*\*

```

30462 20000 WBUFF: .BLK 20000 ; BK
50462 20000 RBUFF: .BLK 20000

436 .END START

```

ABIN	162	ABN	2366	ABN1	2376	ABN2	2400	ABN3	2427
ABN3A	2440	ABN4	2446	ABN4A	2457	ABORT	111	ABRT	23647
ACHR	2140	ACN	141	ACYO	23703	ADB	22377	ADRN	11633
AHD	15740	ALLO	21221	ALL1	21225	ALT1	2200	ALT2	2600
ALTT1	20733	ALTT2	20743	AM1A	21317	AM1B	21332	AM1C	21345
AM1D	21360	AMP	25500	ARB	20036	ARDT	23462	ASCOO	2312
ASDOB	22007	ASDOC	22052	ASECT	107	ASWB	22742	ATM	16315
ATTS	134	AWB	20027	BINA	2467	BINR	2470	BIT1	746
BIT2	11635	BIT6	11636	BIT7	5370	BLANK	5646	BLONG	1305
BMSK	66	BS	21022	BSB	14265	BSB1	14273	BSB1A	14300
BSB1B	14305	BSB1C	14317	BSB2	14325	BSB2A	14332	BSBEB	14340
BSBT	14260	BSCT	100	BSS	11702	BSS1	11710	BSS1A	11724
BSS1B	11731	BSS1C	11735	BSS1D	11742	BSS2	11747	BSS2A	11760
BSS2B	11765	BSS2C	11771	BSS2D	12000	BSS2E	12002	BSS2F	12006
BSS2G	12013	BSS2E	12014	BSST	11675	BSSTT	25106	BSY0	22107
BSYOD	22123	BSY1	22115	BSYF	21164	CADR	40	CAE	7616
CAE1	7624	CAE1A	7634	CAE1B	7641	CAE1C	7645	CAE2	7652
CAEEB	7671	CAES	7611	CAET	24732	CAL	25552	CATT	105
CATTS	106	CBFUL	20176	CBP	2244	CBS	12440	CBS1	12451
CBS1A	12456	CBS1B	12462	CBS2	12471	CBS2A	12503	CBS2B	12507
CBS2C	12514	CBS3	12522	CBS3A	12526	CBS3B	12533	CBS3C	12555
CBS3D	12562	CBSC	12433	CBSC	25141	CBSEB	12566	CBUF	110
CBUFF	3222	CCHD	15136	CCNT	2243	CDC	25472	CDD	25537
CDER	24046	CDIA	20635	CDIB	20640	CDDA	20627	CDDC	20632
CDRO	20652	CDR1	20660	CDR2	20666	CDR3	20674	CDRF	24063
CDRV	24123	CECC	23221	CF	20775	CF1	23317	CFN	23325
CFDF	21405	CFON	21373	CFS	3715	CFSO	3723	CFS1	3734
CFS1A	3740	CFS2	3746	CFSEB	3753	CFST	24204	CFTM	11632
CHAIN	12060	CHAR	156	CHC	20623	CHD	20607	CHDR	12240
CHDR1	12406	CHDRA	12211	CHH	20622	CHHD	23474	CHHD3	12563
CHJ	20626	CHL	20611	CHN	20625	CHN1	12102	CHN1A	12107
CHN1B	12120	CHN1C	12125	CHN1D	12131	CHN1E	12136	CHN2	12143
CHN2A	12154	CHN2B	12161	CHN2C	12165	CHN2D	12172	CHN2E	12200
CHN2F	12212	CHN3	12214	CHN3A	12225	CHN3B	12232	CHN3C	12237
CHNEB	12243	CHNS	12065	CHNT	25117	CHP	20612	CHR	20613
CHRB	20617	CHS	20614	CHT	20610	CHWB	20615	CHX	20621
CHY	20624	CKAD	15724	CKAD1	15730	CKC	15703	CKCK	201
CKCS	202	CKD	15715	CKDA	204	CKDS	203	CKK	15705
CKRB	23447	CLAT	21624	CLBY	22063	CLD	4165	CLD1	4202
CLD1A	4211	CLD2	4214	CLDA	21736	CLDD	4160	CLDDT	24231
CLDEB	4243	CLDN	21747	CLOD	21722	CLOK	144	CLRW	20643
CMAX	71	CMDL	3107	CMDSO	3162	CMDS1	3166	CMDS2	3175
CMDS3	3204	CML	1377	CMS	1376	CNTR	152	COLM	1307
COMMMA	74	CR	72	CRBF	16317	CRBF1	16326	CRBF2	16337
CRBUF	210	CRK	1302	CRI.F	20076	CSKE	24101	CSTA	132
CTRG	21205	CYE	21030	CYLD	16110	CYLN	127	CYTL	7655
DAC	1310	DAC1	1325	DAC1A	1332	DAC1B	1337	DAC2	1341
DAC2A	1346	DAC2B	1360	DAC2C	1372	DAC2D	1374	DAD	15741
DADD	205	DARG	21271	DAS	1464	DAS1	1470	DAS2	1500
DATA	20045	DATI1	21431	DATI2	21437	DATI3	21445	DATI4	21453
DATI5	21461	DATI6	21467	DATI7	21475	DATI8	21503	DATI9	21511
DATNO	21517	DATN1	21525	DATN2	21533	DATN3	21541	DATN4	21547
DATN5	21555	DATN6	21563	DATN7	21571	DATV	3000	DATVT	20757
DB	21117	DBS	22671	DBST	1526	DBSY	4000	DBY	22642
DCC	1432	DCH	12336	DCH1	12347	DCH1A	12354	DCH1B	12366
DCH1C	12372	DCH1D	12377	DCH2B	12405	DCHEB	12411	DCHN	12331
DCHNT	25127	DCL	3612	DCL1	4225	DCL2	4346	DCL3	4360
DCM	16175	DCM1	16203	DCMC	16214	DCMP	174	DCNT	1306

DCXA	3610	DCXB	3611	DDBS	22626	DDIA	234	DDIA1	237
DDIB	242	DDIB1	245	DDIC	250	DDIR	16342	DDNC	4066
DDGA	211	DDOAP	216	DDOAS	214	DDOAT	7247	DDOB	220
DDOBS	223	DDOC	225	DDOCP	232	DDOCS	230	DDS	13034
DDS1	13037	DDS1A	13044	DDSEO	13075	DDST	24415	DEC	2573
DEC1	2601	DEC2	2606	DEC3	2630	DECN	164	DEL1	2261
DEL2	2265	DELC	2247	DER	21150	DEVC	17652	DF	21142
DIA0	300	DIA1	301	DIAN	305	DIB0	302	DIB1	303
DIBN	306	DICO	304	DICN	307	DIRP	2363	DL	21076
DLIM	636	DLST	637	DMP	1246	DMP0	1254	DMP1	1264
DMP2	1267	DMPP	1304	DMSK	67	DN1	25163	DN10	25220
DN11	25223	DN12	25226	DN13	25231	DN14	25234	DN15	25237
DN16	25242	DN17	25246	DN18	25252	DN19	25256	DN2	25166
DN20	25257	DN21	25260	DN22	25261	DN23	25262	DN24	25263
DN25	25264	DN26	25267	DN27	25272	DN28	25276	DN29	25302
DN3	25171	DN30	25306	DN31	25312	DN32	25316	DN33	25321
DN34	25325	DN35	25330	DN36	25334	DN37	25335	DN38	25336
DN39	25337	DN4	25174	DN40	25340	DN41	25343	DN42	25346
DN43	25351	DN44	25355	DN45	25361	DN46	25362	DN47	25363
DN48	25364	DN49	25367	DN5	25177	DN50	25370	DN51	25373
DN52	25376	DN53	25401	DN54	25404	DN55	25407	DN56	25412
DN57	25413	DN58	25414	DN59	25415	DN6	25202	DN60	25416
DN61	25421	DN62	25424	DN63	25427	DN64	25433	DN65	25437
DN66	25440	DN67	25441	DN68	25442	DN69	25443	DN7	25205
DN70	25444	DN71	25447	DN72	25452	DN73	25455	DN74	25456
DN75	25457	DN76	25462	DN77	25465	DN78	25470	DN79	25471
DN8	25211	DN9	25215	DNAM	640	DNAM0	647	DNAM1	654
DNAME	17733	DNE0	21661	DNE1	21667	DNE1D	22143	DNEF	21177
DNP	21675	DNST	22655	DNTL	17451	DNTP	17450	DNUA	702
DNUM	667	DNUM1	700	DNUM2	706	DNUM3	722	DNUM4	724
DNUMT	17742	DOA0	275	DOB0	276	DOCO	277	DR	21111
DRAT	21710	DRB	23036	DRBF	1242	DRBY	6433	DRC	15573
DRC0	15617	DRC1	15650	DRC1A	15653	DRC1B	15663	DRC1C	15640
DRC2	15661	DRCMD	167	DRD	3774	DRD1	4007	DRD2	4013
DRD3	4026	DRD3A	4035	DRD3B	4043	DRD4	4047	DRD4A	4053
DRD4B	4057	DRD5	4063	DRDEB	4114	DRDL	13063	DRDN	3767
DRDNT	24216	DRDS	13027	DRDY	10000	DRFL	70	DRFT	22613
DRIVE	601	DRM	11634	DRNM	20057	DRONL	17677	DRRG	21251
DRST	22565	DRSV	15554	DRVO	0	DRV1	40	DRV2	100
DRV3	140	DRVF	121	DRVN	125	DRVR	1005	DSA	13364
DSA1	13371	DSA1A	13402	DSA2	13424	DSA2A	13431	DSA2B	13436
DSA3	13443	DSA3A	13450	DSA3B	13455	DSA4	13462	DSA4A	13467
DSA4B	13474	DSA5	13501	DSA5A	13506	DSA5B	13513	DSAB	13357
DSABT	24456	DSAEB	13521	DSB	13614	DSB1	13621	DSB2	13626
DSB2A	13633	DSBB	13607	DSBEB	13641	DSC	13662	DSC1	13674
DSC1A	13701	DSC1B	13713	DSC1C	13720	DSC2	13725	DSC2A	13737
DSC2B	13744	DSC3	13751	DSC3A	13763	DSC3B	13770	DSCB	13655
DSCEB	13776	DSCK	23126	DSEL	13412	DSKT	120	DSP1	17420
DSP1A	17425	DSP2	17442	DSP3	17444	DSPL	17415	DSPLA	3063
DSTA	133	DSTA0	15665	DSTAP	6432	DT1	16720	DT10	16764
DT11	16770	DT12	16774	DT13	17000	DT14	17004	DT15	17010
DT16	17014	DT17	17020	DT18	17024	DT19	17030	DT2	16724
DT20	17034	DT21	17040	DT22	17044	DT23	17050	DT24	17054
DT25	17060	DT26	17064	DT27	17070	DT28	17074	DT29	17100
DT3	16730	DT30	17104	DT31	17110	DT32	17114	DT33	17120
DT34	17124	DT35	17130	DT36	17134	DT37	17140	DT38	17144
DT39	17150	DT4	16734	DT40	17154	DT41	17160	DT42	17164
DT43	17170	DT44	17174	DT45	17200	DT46	17204	DT47	17210

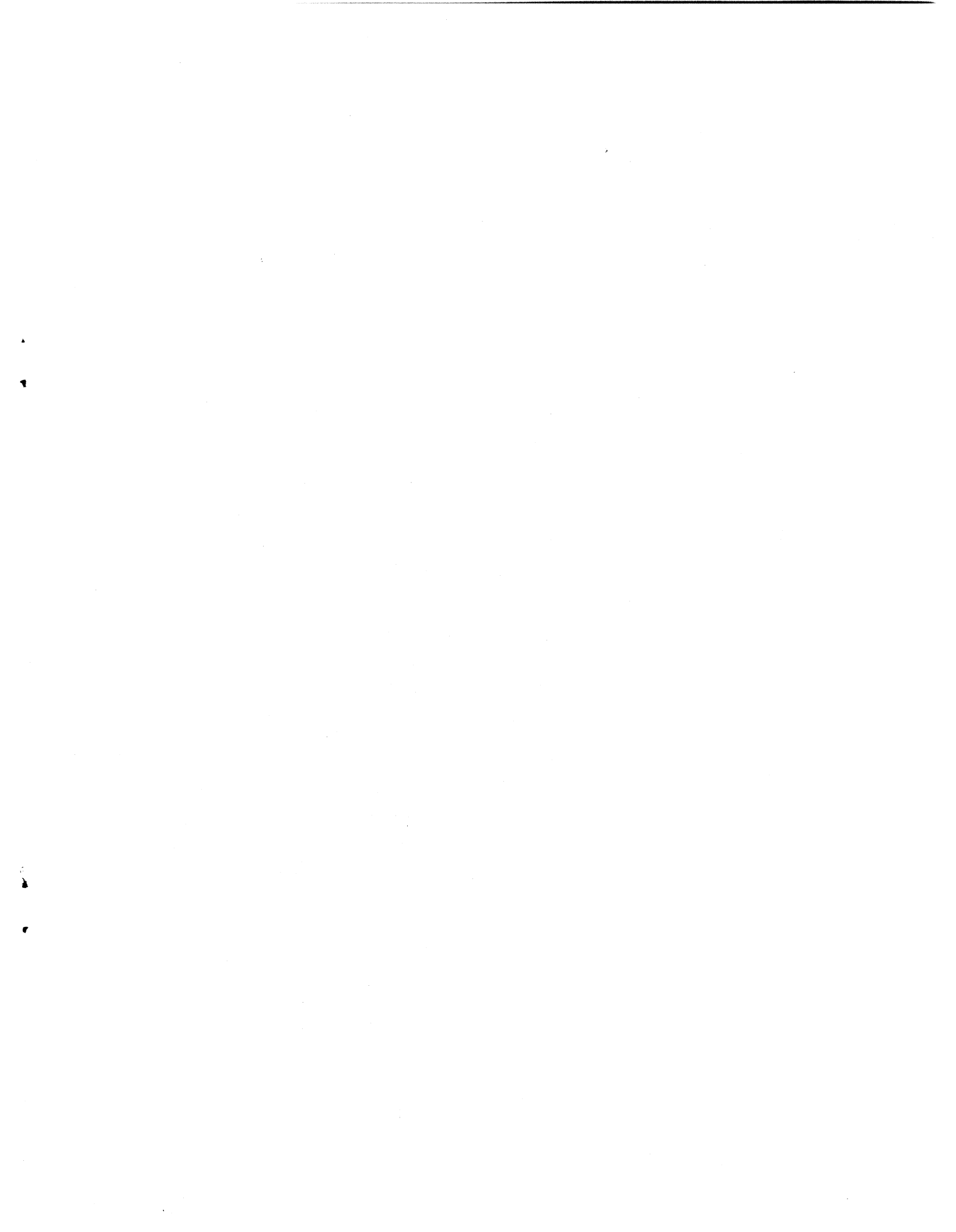
DT48	17214	DT49	17220	DT5	16740	DT50	17224	DT51	17230
DT52	17234	DT53	17240	DT54	17244	DT55	17250	DT56	17254
DT57	17260	DT58	17264	DT59	17270	DT6	16744	DT60	17274
DT61	17300	DT62	17304	DT63	17310	DT64	17314	DT65	17320
DT66	17324	DT67	17330	DT68	17334	DT69	17340	DT7	16750
DT70	17344	DT71	17350	DT72	17354	DT73	17360	DT74	17364
DT75	17370	DT76	17374	DT77	17400	DT78	17404	DT79	17410
DT8	16754	DT9	16760	DTBL	117	DTN7A	4345	DTST	740
DTX1	25651	DTX10	26215	DTX11	26247	DTX12	26301	DTX13	26333
DTX14	26365	DTX15	26417	DTX16	26451	DTX17	26503	DTX18	26535
DTX2	17455	DTX25	26567	DTX26	26621	DTX27	26660	DTX28	26712
DTX29	26744	DTX3	25707	DTX30	27003	DTX31	27035	DTX32	27074
DTX33	27126	DTX34	27160	DTX35	27212	DTX36	17521	DTX4	25745
DTX40	27244	DTX41	27301	DTX42	27336	DTX43	27373	DTX44	27430
DTX48	27465	DTX5	26003	DTX50	27517	DTX51	27555	DTX52	27613
DTX53	27651	DTX54	27707	DTX55	27745	DTX6	26035	DTX60	30003
DTX61	30040	DTX62	30075	DTX63	30132	DTX64	30170	DTX67	17570
DTX7	26073	DTX70	30226	DTX71	30260	DTX72	30312	DTX75	30344
DTX76	30376	DTX77	30430	DTX8	26125	DTX9	26163	DVCD	151
DWBF	1240	DWC	22750	ECCE	200	ECCH	11112	ECCL	11113
ECE	21062	ECM	16303	ECMD	16262	EHDW	23716	EIGHT	60
ELEVN	102	EMCMD	207	ENDD	12727	ENDE	15433	ENDE1	15440
ENDL	5756	ENDL1	5770	ENDM	3002	EQM	161	ERR	1017
ERR0A	1052	ERR1	1077	ERR1A	1120	ERR1B	1124	ERR2	1127
ERR2A	1140	ERR3	1143	ERR3A	1146	ERR3B	1155	ERR4	1157
ERR4A	1166	ERR4B	1176	ERROR	166	ERRR	1303	FBCS	23506
FCC	10031	FCC1	10037	FCC1A	10044	FCC1B	10051	FCC1C	10055
FCC1D	10062	FCC1E	10067	FCC1F	10073	FCC2	10100	FCC2A	10105
FCC2B	10112	FCC2C	10116	FCC2D	10123	FCC3	10130	FCC3A	10141
FCC3B	10145	FCC3C	10152	FCC3D	10157	FCC3E	10167	FCC3F	10174
FCC3G	10201	FCC3H	10225	FCCB	10024	FCCEB	10232	FCCT	24765
FCS	23560	FCY	10347	FCY1	10355	FCY1A	10364	FCY1B	10371
FCY1C	10377	FCY1D	10412	FCY2	10417	FCY2A	10425	FCY2B	10436
FCY2C	10443	FCY2D	10452	FCY3	10460	FCY3A	10473	FCY3B	10500
FCY3C	10507	FCYEB	10510	FCYL	10342	FCYLT	25000	FIVE	36
FIXED	104	FLT	15562	FM1	6463	FM1A	6473	FM1B	6500
FM1C	6505	FM1EB	6506	FM1T	24570	FM2	6733	FM2A	6743
FM2R	6750	FM2C	6755	FM2EB	6756	FM2T	24626	FMAT	1400
FMATT	20727	FMB	14103	FMC1	6456	FMC2	6726	FMKO	23146
FMK1	23155	FML	10561	FML1	10574	FML1A	10603	FML1B	10610
FMLEB	10622	FMLS	10554	FMLST	25011	FMTG	23271	FOUR	30
FRM	14025	FRM1	14037	FRM1A	14044	FRM2	14051	FRM2A	14063
FRM2B	14070	FRM2C	14075	FRMEB	14111	FRMS	14020	FRMST	24467
FTA	23575	FUJ	25527	GO	2703	GO1	2712	GO2	2723
GOTO	2751	GT10	23260	HO	0	H1	2000	H10	20000
H2	4000	H20	40000	H4	10000	HCO	20334	HC1	20350
HC10	20543	HC11	20560	HC2	20366	HC3	20412	HC4	20430
HC5	20446	HC6	20463	HC7	20473	HCB	20511	HC9	20526
HCAD	24031	HCC	16153	HCL	3041	HCM	16114	HCM1	16123
HCM2	16147	HCMP	173	HD	20766	HDC	16105	HDC	16001
HDG1	16026	HDG1A	16037	HDG1B	16052	HDG2	16054	HDG3	16062
HDG4	16072	HDRG	172	HDVO	22537	HDV1	22552	HDW	11447
HEAD	16106	HIGH	7436	HLAST	14726	HLNO	20320	HLN1	25612
HLPO	3013	HLP1	3017	HLP2	3027	HLP2R	3040	HLP3	3060
HMSK	103	HPOS	23657	HUB	7712	HUB1	7721	HUB1A	7725
HUB1B	7732	HUB1C	7743	HUB1D	7750	HUB2	7755	HUB2A	7760
HUB2B	7767	HUBEB	7774	HUBS	7705	HURT	24751	HWC	22730
IC	21135	ICNT	2464	IDC	450	IDC1	460	IDC1A	463

IDC2	465	IDC2A	471	IDC2B	476	IFLG	5065	ILCMD	100
ILCS	24015	IMA	23362	IMSKO	22235	IMSK1	22224	IN	20063
INCK	22457	INH1	22306	INP1	2170	INP2	2223	INP3	2226
INP4	2233	INP5	2236	INP6	2240	INPT	2154	INPUT	157
INSN	435	INST	400	INSTL	112	INTAK	22175	INTI	22210
INTN	22216	INTOC	22274	INTRT	22263	INTX0	5066	INTX1	5067
IPTR	2465	ISA	21012	ISCT	400	ISKA	200	K10	55
K15	4601	K16	4473	K17	56	K2	50	K20	57
K27	60	K3	51	K37	61	K4	52	K40K	5525
K5	53	K7	54	K70	62	KCR	17447	KEN	25571
LCNT	150	LCYN	124	LEAD	2642	LF	73	LMSK	2245
LOGIC	557	LOOP	1205	LOW	7435	LSECT	16111	LSFN	122
LTST	573	MAR	4406	MARO	4414	MAR1	4423	MAR2	4425
MAR2A	4432	MAR2B	4437	MAR3	4442	MAR3A	4452	MAR4	4454
MAR4A	4462	MAR4B	4467	MAREB	4474	MART	24260	MATCH	160
MAXS	5371	MCD	25601	MDIA	20137	MDIA1	20150	MDIA2	20162
MDIB	20142	MDIB1	20155	MDIB2	20167	MDIC	20145	MDOA	20077
MDOAP	20120	MDOAS	20124	MDOB	20102	MDOBS	20110	MDOC	20105
MDOCP	20114	MDOCS	20130	MFLG	135	MKBO	22324	MKB1	22247
MMX	25507	MNI0C	21620	MNI0P	20134	MRST	21155	MS	23277
MSG0	20211	MSG1	20233	MSG2	20255	MSG3	20275	MSKA	70
MSKB	5070	MSKT	23242	MTCH	2314	MTCH1	2330	MTCH2	2341
MTCH3	2345	MTCH4	2351	MTCH5	2361	N1	37	N17	63
N2	36	N400	64	N6	6566	NCLA	21231	NCLR	22072
NCNT	2466	NCY0	23673	NEXT	140	NFCS	23541	NFST	23051
NINE	101	NINEE	66	NINT	22340	NNI0C	257	NNI0P	255
NNI0S	253	NOG0	2752	N00P	2600	N0P	20753	NORM	17773
NPM0	22471	NPM1	22510	NRD	15670	NRD1	15675	NRDT	17752
NRDY	171	NRST	21264	NSTP	20021	NTS	3003	NTST	114
NYL	3213	OBIN	163	OBN	2473	OBN1	2503	OBN2	2505
OBN3	2533	OBN3A	2544	OBN4	2553	OBN4A	2564	OCT	2644
OCT1	2650	OCT2	2654	OCTN	165	OFFF	24142	OFFR	24152
OF5F	1000	OF5R	1200	OKI	25517	ONE	6	OPTN	1221
OPTN1	1233	OSECT	16112	OUT	20067	OVL	11515	OVL1	11525
OVL1A	11532	OVL1B	11537	OVL1C	11554	OVL2	11564	OVL2A	11576
OVL2B	11606	OVL2C	11623	OVL2D	11630	OVLEB	11637	OVLPS	11510
OVLPT	25077	PASS	146	PASSE	20010	PAST	20005	PATN	14525
PCH	147	PDAT	11451	PDIR	3064	PFLG	143	PLLS	22424
PMD	5570	PMEB	5704	PMS	5337	PMS1	5350	PMS1A	5356
PMS1B	5362	PMS2	5373	PMS2A	5412	PMS2B	5425	PMS3	5434
PMS3A	5447	PMS3B	5454	PMS3C	5465	PMS3D	5471	PMS4	5477
PMS4A	5507	PMS4B	5512	PMS4C	5523	PMS5	5526	PMS5A	5535
PMS5B	5542	PMSA	5662	PNAM1	20575	PNAM2	20600	PNAM3	20603
PRG1	517	PRG2	524	PRG2A	526	PRG2B	530	PRG3	534
PRG3A	541	PRG3B	543	PRG3C	545	PRIM	16155	PRIM1	16163
PRIM2	16173	PRIME	176	PRINT	155	PRMT	20074	PRNT	2075
PRNT1	2112	PRNT2	2116	PRNT3	2133	PROCD	1212	PROG	505
PROGL	551	PROMS	5331	PROMT	24375	PTBA	5372	PTBL	5663
PTBP	5476	PTD	22413	PWD	2063	QM	2146	GMT	20174
QUIT	2702	RBEB	15766	RBFC	22347	RBFP	154	RBIN	5062
RBNO	4705	RBNOA	4712	RBN0B	4722	RBN1B	4726	RBN1C	4730
RBN1D	4734	RBN1E	4742	RBN2	4755	RBN2A	4764	RBN2B	4772
RBN2C	5000	RBN2D	5005	RBN2E	5012	RBN3	5020	RBN3A	5026
RBN3B	5027	RBN3C	5033	RBN3D	5035	RBN4	5041	RBN4A	5043
RBN4B	5047	RBNEB	5071	RBNT	4675	RBNTT	24337	RBUF	116
RBUFF	50462	RBZO	5202	RBZ0A	5222	RBZ1	5226	RBZ1A	5241
RBZ1B	5243	RBZ1C	5247	RBZ1D	5253	RBZ2	5260	RBZEB	5273
RBZT	5173	RBZTT	24356	RCAL	200	RCL	6405	RCL1	6413

RCL2	6417	RCL2A	6424	RCL2B	6431	RCLEB	6434	RCLT	24562
RD1	7063	RD1A	7073	RD1B	7075	RD1C	7106	RD1D	7113
RD1E	7124	RD1EB	7141	RD1F	7127	RD1T	24656	RDBF	3200
RDBS	23431	RDDC	23627	RDFC	23522	RDNX	22774	RDOV	23415
RDT	23214	RDT1	7056	RDUD	21641	RDUN	21577	RDYS	23750
READ	0	RECAL	6400	REF	11323	REFP	16341	REFR	17762
REH	13110	REH1	13126	REH1A	13141	REH2	13145	REH2A	13152
REH2B	13160	REHD	13103	REHDT	24426	REHEO	13251	RELS	23741
REPT	1213	RESV	40000	RETA	1740	RF1	6527	RF1A	6534
RF1B	6540	RF1C	6545	RF1D	6552	RF1E	6554	RF1EB	6567
RF1F	6562	RF1T	24600	RF2	6777	RF2A	7010	RF2B	7015
RF2C	7022	RF2D	7033	RF2EB	7034	RF2T	24637	RFF	12637
RFF1	12650	RFF1A	12654	RFF1B	12661	RFF1C	12672	RFF1D	12677
RFF1E	12704	RFFEB	12705	RFFI	25152	RFIFO	12632	RFM1	6522
RFM2	6772	RGDO	3451	RGD1	3456	RGD1A	3464	RGD2	3473
RGD2A	3511	RGD2B	3517	RGD2C	3525	RGD3	3527	RGD3A	3536
RGD3B	3545	RGD3C	3554	RGD3D	3561	RGD4	3570	RGD4A	3573
RGD4B	3600	RGDEB	3621	RGRS	3300	RGWR	3443	RGWRT	24172
RHD	23140	RHDR	3600	RLD	13005	RLDO	13000	RLDOT	24406
RLD1	13013	RLDEO	13021	RLSD	1600	RMA	7231	RMKO	23164
RMK1	23175	RML	10635	RML1	10652	RML1A	10661	RML1B	10665
RML2	10672	RML2A	10676	RML2B	10703	RML2C	10712	RMLEB	10721
RMLS	10630	RMLST	25030	RMSK	65	ROF	15226	ROF1	15240
ROF1A	15245	ROF1B	15257	ROF1C	15264	ROF1D	15271	ROF2	15276
ROF2A	15310	ROF2B	15315	ROF2C	15322	ROF3	15327	ROF3A	15341
ROF3B	15346	ROF3C	15353	ROFEB	15361	ROFS	15221	ROFST	24554
RRGL	1204	RRGX	1616	RRST	24162	RSBY	22100	RSEB	3363
RSET	21172	RSTO	3305	RST1	3306	RST2	3312	RST3	3316
RST3A	3323	RST4	3332	RST4A	3337	RST5	3346	RST5A	3353
RSTH	23070	RUBO	75	RWA	21005	RWC	15451	RWC1	15476
RWC2	15524	RWCMD	170	RWDA	22714	RWDN	40000	RWDS	22701
RWF	21104	RWFT	1	RWD	21067	RWT	7306	RWT1	7315
RWT1A	7323	RWT1B	7331	RWT1C	7336	RWT2	7341	RWT2A	7347
RWT2B	7360	RWT2C	7365	RWT2D	7375	RWT2E	7405	RWT2F	7407
RWT2G	7414	RWT3	7417	RWT3A	7427	RWT3B	7434	RWTEB	7450
RWTM	7301	RWTD	4	RWTT	24704	RXMA	21772	SO	0
S1	40	S10	400	S11	440	S14	600	S17	740
S2	100	S20	1000	S22	1100	S25	1240	S3	140
S30	1400	S37	1740	S4	200	S6	300	SAD	2276
SADC	200	SAE	21040	SAFD	22526	SAR	23301	SAVP	142
SBSY	21303	SC	20772	SCH	14775	SCH1	15007	SCH1A	15014
SCH1B	15026	SCH1C	15033	SCH1D	15045	SCH1E	15052	SCH2	15057
SCH2A	15071	SCH2B	15076	SCH2C	15103	SCH3	15111	SCH3A	15123
SCH3B	15130	SCHEB	15141	SCHN	14770	SCK	11207	SCK1	11222
SCK1A	11225	SCK1B	11231	SCK1C	11236	SCK1D	11243	SCK2	11263
SCK2A	11267	SCK2B	11273	SCK2C	11300	SCK2D	11311	SCKEB	11324
SCL	7131	SCL1	7340	SCM1	16241	SCM1A	16250	SCMD	16217
SCNT	131	SCTN	130	SE	21125	SECK	11202	SECKT	25061
SECT	16107	SEEK	400	SEEKT	20724	SEKK	11631	SET	177
SETT	2753	SEVEN	52	SEVN	100	SFCN	126	SHIFT	2661
SIX	44	SKBN	272	SKBZ	267	SKCMD	206	SKDN	261
SKDZ	264	SKLM	10720	SKMX	23264	SKPT	1220	SKTM	23253
SMSK	102	SPACE	76	SRBO	4624	SRB1	4631	SRB1A	4633
SRB2	4640	SRB2A	4646	SRBEB	4653	SRBY	4616	SRBYT	24325
SRSD	21756	SRWD	22163	SSA	7513	SSA1	7535	SSA1A	7541
SSA1B	7545	SSA1C	7552	SSA1D	7556	SSA2	7563	SSAE	20
SSAEB	7575	SSAL	7566	SSAS	7506	SSAT	24712	SSCK	23762
SSD	14367	SSD1	14401	SSD1A	14404	SSD1B	14416	SSD2	14423

SSD2A	14430	SSD2B	14435	SSD2C	14447	SSD2D	14454	SSD3	14461
SSD3A	14466	SSD3B	14473	SSD3C	14500	SSD3D	14516	SSD3E	14507
SSD4	14521	SSDA	23341	SSDEB	14543	SSDT	14362	SSDTT	24514
SSR	4524	SSRO	4531	SSR1	4541	SSR2	4551	SSR2A	4563
SSR2B	4567	SSREB	4602	SSRT	24302	ST	20770	ST11	23101
START	436	STL	1505	STP	20014	STPE	17667	STRK	123
SVENG	21313	TDI1	1742	TDI10	2014	TDI11	2020	TDI12	2024
TDI13	2030	TDI15	2036	TDI16	2045	TDI17	2054	TDI2	1751
TDI3	1760	TDI4	1767	TDI5	1776	TDI6	2004	TDI7	2010
TDIA	3606	TDIC	3607	TDO1	1641	TDO10	1706	TDO11	1712
TDO12	1716	TDO13	1722	TDO14	1727	TDO15	1733	TDO2	1650
TDO3	1657	TDO4	1665	TDO5	1671	TDO6	1675	TDO7	1702
TEMP	3242	TEMPA	113	TEN	74	TENA	2633	TENN	2641
TESTR	750	THREE	22	TIME	2663	TIME1	2666	TIMER	175
TIMM	7440	TIMO	145	TK1	2700	TM1	2701	TMG	22761
TMO	20052	TMOD	17661	TMTS	7413	TNRD	23711	TOAT	23015
TOLR	23024	TOP	7162	TOP1	7170	TOP1A	7175	TOP2	7204
TOP2A	7210	TOP2B	7215	TOPEB	7251	TOPM	7155	TOPT	24670
TPD	23731	TPL	13167	TPR	13272	TPR1	13277	TPR1A	13304
TPR2	13312	TPR2A	13317	TPR2B	13324	TPREO	13343	TPRL	13265
TPRLT	24441	TREFO	13333	TREF1	13337	TRES	2000	TSD	14615
TSD1	14623	TSD1A	14635	TSD1B	14642	TSD1C	14647	TSD2	14655
TSD2A	14664	TSD2B	14667	TSD3	14674	TSD3A	14710	TSD3B	14713
TSD3C	14720	TSDEB	14732	TSDT	14607	TSDTT	24536	TSHD	14727
TSTR	763	TSTR1	771	TSTT	17770	TSTV	137	TSTX	136
TTL0	17612	TTL1	17636	TWELF	110	TWO	14	UDC	20702
UNB1	7772	UNB2	7773	USRD	23353	VO	20714	V1	20720
VARD	23612	VBD	23226	VE	21053	VOL1	20	VRBD	22362
VRF	10756	VRF1	10767	VRF1A	10773	VRF1B	11000	VRF2	11005
VRF2A	11011	VRF2B	11016	VRF2C	11023	VRF2D	11037	VRF3	11043
VRF3A	11054	VRF3B	11061	VRF3C	11066	VRF4	11073	VRF4A	11100
VRF4B	11104	VRF4C	11111	VRFE	10	VRFEB	11114	VRFY	10751
VRFYT	25044	VSEL	15553	WAC	11361	WAC1	11400	WAC1A	11405
WAC1B	11411	WAC1C	11416	WAC2	11423	WAC2A	11427	WAC2B	11435
WAC2C	11441	WAC2D	11446	WACEB	11452	WACY	11354	WACYT	25070
WBAD	15753	WBFP	153	WBP	16104	WBUF	115	WBUFF	30462
WD1T	23403	WDSB	15557	WDST	22576	WDT	23206	WH1	6616
WH1A	6630	WH1B	6634	WH1C	6641	WH1T	24613	WH2	6646
WH2A	6651	WH2B	6663	WH2C	6670	WH2D	6675	WHD	23117
WHDR	600	WHEB	6676	WHS1	6611	WLKO	22035	WLK1	22020
WMAP	7225	WORD	20000	WPFT	23775	WPRT	1000	WRGL	1203
WRGX	1577	WRH	14154	WRH1	14162	WRH1A	14167	WRH2	14174
WRH2A	14207	WRH2B	14214	WRH2C	14221	WRHD	14147	WRHDT	24502
WRHEB	14230	WRITE	3400	XTM	4265	XTM1	4277	XTM1A	4307
XTM1B	4316	XTM2	4321	XTM2A	4333	XTM2B	4342	XTMA	4257
XTMAT	24242	XTMEB	4372	ZERO	77				





**POINT 4 DATA CORPORATION**

2569 McCabe Way / Irvine, California 92714 / (714) 863-1111