

RP11E

DISK PACK FORMATTER
MD-11-DZRP2-B

EP-DZRP2-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

Frame 1	Frame 2	Frame 3	Frame 4
Frame 5	Frame 6	Frame 7	Frame 8
Frame 9	Frame 10	Frame 11	Frame 12
Frame 13	Frame 14	Frame 15	Frame 16
Frame 17	Frame 18	Frame 19	Frame 20
Frame 21	Frame 22	Frame 23	Frame 24
Frame 25	Frame 26	Frame 27	Frame 28
Frame 29	Frame 30	Frame 31	Frame 32
Frame 33	Frame 34	Frame 35	Frame 36
Frame 37	Frame 38	Frame 39	Frame 40

DZRP2 LST

101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151

285 THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR
THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE I
NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SEPRPC).
NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS

291 EM ::POINTS TO THE ERROR MESSAGE
 DH ::POINTS TO THE DATA HEADER
 DT ::POINTS TO THE DATA
 DF ::POINTS TO THE DATA FORMAT

430 THIS ROUTINE WILL FORMAT THE ENTIRE PACK AND THEN VERIFY
THE HEADERS WITHIN EACH SECTOR. STARTING ADDRESS IS 200.

468 THIS ROUTINE ALLOWS THE OPERATOR TO VERIFY THE FORMAT OF THE
ENTIRE PACK. THIS ROUTINE IS AUTOMATICALLY INITIATED
BY THE STANDARD FORMATTER ROUTINE (TEST1). THE STARTING
ADDRESS OF THIS ROUTINE IS 204.

527 THIS ROUTINE ALLOWS THE OPERATOR TO SPECIFY A PARTICULAR
HEADER TO BE FORMATTED AND THEN VERIFIED. THE STARTING ADDR
IS 210.

623 THIS ROUTINE WILL FORMAT THE ENTIRE PACK USING THE SECTOR
ADDRESS SEQUENCE SPECIFIED BY THE OPERATOR. THE STARTING
ADDRESS IS 214.

821 TK INITIALIZE ROUTINE
THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
CALL:
 JSR PC,STKINT
 RETURN

838 TK SERVICE ROUTINE
THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
IT IN THE QUEUE.

882 SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
CALL WHEN OPERATING IN TTY INTERRUPT MODE.

897 CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SE
ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RES

899 CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SE

DZRP2 LST

02-APR-76 13:4

153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207

960 THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
CALL:
ROCHR ;:GET A CHARACTER FROM THE QUEUE
RETURN HERE ;:CHARACTER IS ON THE STACK
;:WITH PARITY BIT STRIPPED OFF

984 THIS ROUTINE WILL INPUT A STRING FROM THE TTY
CALL:
ROLIN ;:INPUT A STRING FROM THE TTY
RETURN HERE ;:ADDRESS OF FIRST CHARACTER WIL
;:TERMINATOR WILL BE A BYTE OF A

1053 THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
CHANGE IT TO BINARY.
THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPE
FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUS
THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RE
CALL:
ROOCT ;:READ AN OCTAL NUMBER
RETURN HERE ;:LOW ORDER BITS ARE ON TOP OF T
;:HIGH ORDER BITS ARE IN SHIOCT

1107 ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 B
THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE
NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
NOTE2: \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.
CALL:
1) USING A TRAP INSTRUCTION
TYPE ,MESADR ;:MESADR IS FIRST ADDRESS OF AN
OR
TYPE
MESADR

1178 THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIG
1179 OCTAL (ASCII) NUMBER AND TYPE IT.
STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS
CALL:
MOV NUM,-(SP) ;:NUMBER TO BE TYPED
TYPOS ;:CALL FOR TYPEOUT
.BYTE N ;:N=1 TO 6 FOR NUMBER OF DIGITS
.BYTE M ;:M=1 OR 0
;:1=TYPE LEADING ZEROS
;:0=SUPPRESS LEADING ZER

STYPOh----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE
STYPOS OR STYPOC
CALL:
MOV NUM,-(SP) ;:NUMBER TO BE TYPED
TYPON ;:CALL FOR TYPEOUT

FO1

MD-11-DZRP2-B, RPIIE DISK PACK FORMATTER
DZRP28.CMB

MACY11 27(732) 01-NOV-76 16:04 PAGE 6

208

DZRP2 LST

02-APR-76 13:4

209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242

STYPOC---ENTER HERE FOR TYPEOU OF A 16 BIT NUMBER
CALL:
MOV NUM,-(SP) ;;NUMBER TO BE TYPED
TYPOC ;;CALL FOR TYPEOUT

1256 SAVE RO-R5
CALL:
SAVREG
UPON RETURN FROM SSAVREG THE STACK WILL LOOK LIKE:

TOP---(+16)
+2---(+18)
+4---R5
+6---R4
+8---R3
+10---R2
+12---R1
+14---R0

1283 RESTORE RO-R5
CALL:
RESREG

1302 THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTIO
AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDR
OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
GO TO THAT ROUTINE.

1324 THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLE
BY THE "TRAP" INSTRUCTION.

1420 THIS ROUTINE CLEARS THE RP11 AND DETERMINES WHICH DRIVES ARE AVA
THE TABLE 'DRVSTA' IS LOADED TO REFLECT THE SYSTEM STATUS.

HO1

MO-11-DZRP2-8, RPIIE DISK PACK FORMATTER
DZRP28.CMB

MACY11 27(732) 01-NOV-76 16:04 PAGE 8

245

%

267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301

```
.TITLE MD-11-DZRP2-B, RP11E DISK PACK FORMATTER
;*COPYRIGHT (C) 1975, 1976
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY C. HESS/F. ROEMER
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C1), MAR 24, 1976.
;*
```

```
.SBTTL OPERATIONAL SWITCH SETTINGS
;
; SWITCH USE
;-----
; 15 HALT ON ERROR
; 14 LOOP ON TEST
; 13 INHIBIT ERROR TYPEOUTS
; 10 BELL ON ERROR
; 7 INHIBIT DRIVE STATUS TABLE PRINTOUT
```

```
.SBTTL TRAP CATCHER
```

```
000000
000174 000000
000176 000000
```

```
.=0
;#ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
;#SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
;#LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
DISPREG: .=174 .WORD 0 ;;SOFTWARE DISPLAY REGISTER
SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER
```

```
.SBTTL STARTING ADDRESSES
```

```
000200
000200 000137 001304
000204 000137 001314
000210 000137 001324
000214 000137 001334
000220 000137 001344
```

```
.=200
;#200 = FORMAT DISK AND CHECK HEADERS
JMP @#START1
;#204 = CHECK HEADERS ONLY
JMP @#START2
;#210 = FORMAT AND CHECK THE SPECIFIED HEADER
JMP @#START3
;#214 = FORMAT AND CHECK HEADERS IN THE OPERATOR SPECIFIED SEQUENCE
JMP @#START4
;#220 = CHANGE THE RP11E BUS ADDRESS, VECTOR ADDRESS, AND PRIORITY
JMP @#START5
```

```
.SBTTL BASIC DEFINITIONS
```

```
001100
```

```
;#INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
```

302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357

000011
000012
000015
000200
177776

177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007

000000
000040
000100
000140
000200
000240
000300
000340

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

```
.EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL

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

;#GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0                ;;GENERAL REGISTER
R1= %1                ;;GENERAL REGISTER
R2= %2                ;;GENERAL REGISTER
R3= %3                ;;GENERAL REGISTER
R4= %4                ;;GENERAL REGISTER
R5= %5                ;;GENERAL REGISTER
R6= %6                ;;GENERAL REGISTER
R7= %7                ;;GENERAL REGISTER
.EQUIV R6,SP          ;;STACK POINTER
.EQUIV R7,PC          ;;PROGRAM COUNTER

;#PRIORITY LEVEL DEFINITIONS
PR0= 0                ;;PRIORITY LEVEL 0
PR1= 40               ;;PRIORITY LEVEL 1
PR2= 100              ;;PRIORITY LEVEL 2
PR3= 140              ;;PRIORITY LEVEL 3
PR4= 200              ;;PRIORITY LEVEL 4
PR5= 240              ;;PRIORITY LEVEL 5
PR6= 300              ;;PRIORITY LEVEL 6
PR7= 340              ;;PRIORITY LEVEL 7

;#SWITCH REGISTER SWITCH DEFINITIONS
SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
SW06= 100
SW05= 40
SW04= 20
SW03= 10
SW02= 4
SW01= 2
SW00= 1
.EQUIV SW09,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
```


414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431

000000
000002
000004
000006
000010
000012
000014
000016
000020
000022
000024
000026

;RP11E REGISTER INDEXES

RPDS=00
RPER=02
RPCS=04
RPWC=06
RPBA=10
RPCA=12
RPDA=14
RPM1=16
RPM2=20
RPM3=22
SUCA=24
SILO=26

.SBTTL COMMON TAGS

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

432		
433		
434		
435		
436		
437		
438		001100
439	001100	000000
440	001100	000
441	001102	000
442	001103	000
443	001104	000000
444	001106	000000
445	001110	000000
446	001112	000000
447	001114	000
448	001115	001
449	001116	000000
450	001120	000000
451	001122	000000
452	001124	000000
453	001126	000000
454	001130	000000
455	001132	000000
456	001134	000
457	001135	000
458	001136	000000
459	001140	177570
460	001142	177570
461	001144	177560
462	001146	177562
463	001150	177564
464	001152	177566
465	001154	000
466	001155	002
467	001156	012
468	001157	000
469	001160	177607 000377
470	001164	077
471	001165	015
472	001166	000012
473		
474		
475	001170	000000
476	001172	000000
477		
478	001174	000000
479	001176	000000
480	001200	000000
481	001202	000000
482	001204	000000
483	001206	000000
484	001210	000000
485	001212	000000
486	001214	000000
487	001216	000000

SCMTAG:	.WORD	0	.=1100
SPASS:	.WORD	0	
STSTN:	.BYTE	00	
SERFLG:	.BYTE	00	
SICNT:	.WORD	00	
SI.PADR:	.WORD	00	
SI.PERP:	.WORD	00	
SERTTL:	.WORD	00	
SITEMB:	.BYTE	00	
SERMAX:	.BYTE	01	
SERRPC:	.WORD	00	
SGDADR:	.WORD	00	
SBDADR:	.WORD	00	
SGODAT:	.WORD	00	
SBDAT:	.WORD	00	
SAUTOB:	.BYTE	00	
SINTAG:	.BYTE	00	
SWR:	.WORD	DSWR	
DISPLAY:	.WORD	DDISP	
\$TKS:		177560	
\$TKB:		177562	
\$TPS:		177564	
\$TPB:		177566	
\$NULL:	.BYTE	0	
\$FILLS:	.BYTE	2	
\$FILLC:	.BYTE	12	
\$TFFLG:	.BYTE	0	
\$BELL:	.ASCIZ	<207><377><377>	
\$QUES:	.ASCIZ	/?	
\$CRLF:	.ASCIZ	<15>	
\$LF:	.ASCIZ	<12>	
MAXCYL:	.WORD	0	
DRVTYP:	.WORD	0	
OPTION:	.WORD	0	
BUSADR:	.WORD	00	
GOOD:	.WORD	00	
BAD:	.WORD	00	
DATA:	.WORD	00	
TOG1:	.WORD	00	
TOG2:	.WORD	00	
WORK:	.WORD	00	
WORK1:	.WORD	00	
WORK2:	.WORD	0	

START OF COMMON TAGS
CONTAINS PAGE COUNT
CONTAINS THE TEST NUMBER
CONTAINS ERROR FLAG
CONTAINS SUBTEST ITERATION COUNT
CONTAINS SCOPE LOOP ADDRESS
CONTAINS SCOPE RETURN FOR ERRORS
CONTAINS TOTAL ERRORS DETECTED
CONTAINS ITEM CONTROL BYTE
CONTAINS MAX. ERRORS PER TEST
CONTAINS PC OF LAST ERROR INSTRUCTION
CONTAINS ADDRESS OF 'GOOD' DATA
CONTAINS ADDRESS OF 'BAD' DATA
CONTAINS 'GOOD' DATA
CONTAINS 'BAD' DATA
RESERVED--NOT TO BE USED
AUTOMATIC MODE INDICATOR
INTERRUPT MODE INDICATOR
ADDRESS OF SWITCH REGISTER
ADDRESS OF DISPLAY REGISTER
TTY KBD STATUS
TTY KBD BUFFER
TTY PRINTER STATUS REG. ADDRESS
TTY PRINTER BUFFER REG. ADDRESS
CONTAINS NULL CHARACTER FOR FILLS
CONTAINS # OF FILLER CHARACTERS REQUIRED
INSERT FILL CHARS. AFTER A "LINE FEED"
"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
CODE FOR BELL
QUESTION MARK
CARRIAGE RETURN
LINE FEED

MAXIMUM CYLINDER ADDRESS
CONTAINS A BIT FOR EACH RPO3 DRIVE
BIT00 IS DRIVE 0, BIT01 IS DRIVE 1, ETC.
CONTAINS A BIT FOR THE TEST SELECTED
CHANGE RP11E ADDRESS FLAG

488 001220 000000
 489 001222 000000
 490 001224 000000
 491 001226 000000
 492 001230 000000
 493 001232 000000
 494 001234 000000
 495 001236 000000
 496 001240 000000
 497 001242 000000
 498 001244 000000
 499 001246 000000
 500 001250 000000
 501 001252 000000
 502 001254 000000
 503 001256 000000
 504 001260 000000
 505 001262 000000
 506
 507
 508
 509
 510 001264 176710
 511 001266 000254 000256
 512 001272 000240
 513
 514
 515
 516
 517
 518
 519 001274 000
 520 001275 000
 521 001276 000
 522 001277 000
 523 001300 000
 524 001301 000
 525 001302 000
 526 001303 000
 527

WORK3: .WORD 0
 UNIT: .WORD 0
 ATNB: .WORD 0
 CYL: .WORD 0
 HED: .WORD 0
 SEC: .WORD 0
 CYLR: .WORD 0
 HEDR: .WORD 0
 SECR: .WORD 0
 CYLA: .WORD 0
 HEDA: .WORD 0
 SECA: .WORD 0
 TTG: .WORD 0
 TEMP1: .WORD 0
 MASK: .WORD 0
 ROTOG: .WORD 0
 LERR: .WORD 0
 DEBUF: .WORD 0

;RPI1E ADDRESSES

RPADR: .WORD 176710 ;RPI1E BUS ADDRESS
 RPVEC: .WORD 254,256 ;RPI1E VECTOR ADDRESSES
 RPPRIO: .WORD (5*32.) ;RPI1E PRIORITY

;DRIVE STATUS TABLE

DRVSTA>0, DRIVE IS ONLINE
 DRVSTA=0, DRIVE IS OFFLINE
 DRVSTA<0, DRIVE IS UNSAFE

DRVSTA: .BYTE 0 ;DRIVE 0
 .BYTE 0 ;DRIVE 1
 .BYTE 0 ;DRIVE 2
 .BYTE 0 ;DRIVE 3
 .BYTE 0 ;DRIVE 4
 .BYTE 0 ;DRIVE 5
 .BYTE 0 ;DRIVE 6
 .BYTE 0 ;DRIVE 7

528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583

.SBTTL ERROR POINTER TABLE

;; THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;; THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;; LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;; NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
;; NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;; EM ::POINTS TO THE ERROR MESSAGE
;; DH ::POINTS TO THE DATA HEADER
;; DT ::POINTS TO THE DATA
;; DF ::POINTS TO THE DATA FORMAT

001304

SEARTB:

;; *****
; NOTE: THE ERROR TABLE IS NOT USED IN THIS PROGRAM.
;; *****

.SBTTL START OF PROGRAM

001304 012737 000001 001174 START1: MOV #1,OPTION ; SET FORMAT AND CHECK FLAG
001312 000421 BR START ; CONTINUE
001314 012737 000002 001174 START2: MOV #2,OPTION ; SET CHECK ONLY FLAG
001322 000415 BR START ; CONTINUE
001324 012737 000004 001174 START3: MOV #4,OPTION ; SET FORMAT SPECIFIED SECTOR FLAG
001332 000411 BR START ; CONTINUE
001334 012737 000010 001174 START4: MOV #10,OPTION ; SET SPECIAL FORMAT FLAG
001342 000405 BR START ; CONTINUE
001344 012737 177777 001176 START5: MOV #-1,BUSADR ; SET CHANGE ADDRESS FLAG
001352 005037 001174 CLR OPTION ; CLEAR THE OTHER START OPTION FLAGS
001356 000005 START: RESET ; CLEAR EVERYTHING
; .SBTTL INITIALIZE THE COMMON TAGS
; ; CLEAR THE COMMON TAGS (\$CHTAG) AREA
MOV # \$CHTAG,R6 ; FIRST LOCATION TO BE CLEARED
CLR (R6)+ ; CLEAR MEMORY LOCATION
CMP #SWR,R6 ; ; DONE?
BNE .-6 ; ; LOOP BACK IF NO
MOV #STACK,SP ; ; SETUP THE STACK POINTER
; ; INITIALIZE A FEW VECTORS
MOV #ERROR,@EMTVEC ; ; EMT VECTOR FOR ERROR ROUTINE
MOV #340,@EMTVEC+2 ; ; LEVEL 7
MOV #STRAP,@TRAPVEC ; ; TRAP VECTOR FOR TRAP CALLS
MOV #340,@TRAPVEC+2 ; ; LEVEL 7
; ; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
; ; EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
MOV @ERRVEC,-(SP) ; ; SAVE ERROR VECTOR
MOV #64,@ERRVEC ; ; SET UP ERROR VECTOR
MOV #DSWR,SWR ; ; SETUP FOR A HARDWARE SWITCH REGISTER
MOV #DDISP,DISPLAY ; ; AND A HARDWARE DISPLAY REGISTER
CMP #-1,@SWR ; ; BY TO REFERENCE HARDWARE SWR

```

584 001464 001012 BNE 66$ :; BRANCH IF NO TIMEOUT TRAP OCCURRED
585 :; AND THE HARDWARE SWR IS NOT = -1
586 001466 000403 BR 65$ :; BRANCH IF NO TIMEOUT
587 001470 012716 001476 64$: MOV #65$, (SP) :; SET UP FOR TRAP RETURN
588 001474 000002 RTI
589 001476 012737 000176 001140 65$: MOV #SWREG, SWR :; POINT TO SOFTWARE SWR
590 001504 012737 000174 001142 MOV #DISPREG, DISPLAY
591 001512 012637 000004 66$: MOV (SP)+, #ERRVEC :; RESTORE ERROR VECTOR
592
593 001516 005227 177777 INC #-1 :; FIRST START ?
594 001522 001035 BNE 1$ :; BR IF NOT
595 001524 104401 001532 TYPE ,68$ :; TYPE ASCIZ STRING
596 001530 000432 BR 67$ :; GET OVER THE ASCIZ
597 :; 68$: .ASCIZ <15><12><12>/MAINDEC-11-DZRP2-B, RP11E DISK PACK FORMATTER/<15><12>
598 001616 :; 67$:
599 001616 104401 001165 1$: TYPE ,SCRLF :; CR-LF
600 001622 004737 004576 JSR PC, STKINT :; SETUP THE TTY KEYBOARD
601 :; SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
602 001626 005737 000042 TST #42 :; ARE WE RUNNING UNDER XXDP/ACT?
603 001632 001006 BNE 69$ :; BRANCH IF YES
604 001634 023727 001140 000176 CMP SWR, #SWREG :; SOFTWARE SWITCH REG SELECTED?
605 001642 001005 BNE 70$ :; BRANCH IF NO
606 001644 104405 GTSWR :; GET SOFT-SWR SETTINGS
607 001646 000403 BR 70$
608 001650 112737 000001 001134 69$: MOV #1, SAUTOB :; SET AUTO-MODE INDICATOR
609 001656 70$:
610 001656 004737 006720 JSR PC, GETADR :; CHECK THE RP11E ADDRESS
611 001662 004737 007230 JSR PC, RPINIT :; CHECK THE SYSTEM
612 001666 005227 177777 INC #-1 :; FIRST START ?
613 001672 001404 BEQ SYSTAT :; BR IF IT IS
614 001674 032777 000200 177236 BIT #SW07, #SWR :; INHIBIT SUBSYSTEM STATUS TYPEOUT ?
615 001702 001070 BNE SELTST :; BR IF INHIBIT
616 001704 SYSTAT:
617 001704 104401 001712 TYPE ,65$ :; TYPE ASCIZ STRING
618 001710 000412 BR 64$ :; GET OVER THE ASCIZ
619 :; 65$: .ASCIZ <15><12>/DRIVE STATUS:/<15><12><12>
620 001736 :; 64$:
621 001736 005001 CLR R1 :; TABLE POINTER
622 001740 012702 000001 MOV #1, R2 :; DRIVE COUNTER
623 001744 1$:
624 001744 010146 MOV R1, -(SP) :; SAVE R1 FOR TYPEOUT
625 :; DRIVE NUMBER
626 001746 104403 TYPOS :; GO TYPE--OCTAL ASCII
627 001750 001 .BYTE 1 :; TYPE 1 DIGIT(S)
628 001751 000 .BYTE 0 :; SUPPRESS LEADING ZEROS
629 001752 105761 001274 TSTB DRVSTA(R1) :; CHECK THE DRIVE'S STATUS
630 001756 100405 BMI 2$ :; BR IF DRIVE UNSAFE
631 001760 001422 BEQ 5$ :; BR IF DRIVE OFFLINE
632 001762 012737 012322 002044 MOV #ONLINE, 7$ :; ONLINE MESSAGE ADDRESS
633 001770 000403 BR 3$ :; CHECK DRIVE TYPE
634 001772 012737 012341 002044 2$: MOV #UNSAFE, 7$ :; UNSAFE MESSAGE ADDRESS
635 001774 030237 001172 3$: BIT R2, DRVSTYP :; RPO3 DRIVE ?
636 001776 001004 BNE 4$ :; BR IF RPO3
637 001778 012737 012377 002050 MOV #RPO2, 8$ :; RPO2 MESSAGE ADDRESS
638 002014 000412 BR 6$ :; TYPE THE STATUS TABLE
639 002016 012737 012404 002050 4$: MOV #RPO3, 8$ :; RPO3 MESSAGE ADDRESS

```

```

640 002024 000406
641 002026 012737 012360 002044 55:  BR 65
642 002034 012737 012050 002050 55:  MOV #OFFLIN,75
643 002042 104401 65:  MOV #S1,85
644 002044 000000 75:  TYPE
645 002046 000000 75:  .WORD 0
646 002050 000000 85:  TYPE
647 002052 104401 001165 85:  .WORD 0
648 002056 005201 TYPE ,SCRLF
649 002060 106302 INC A1
650 002062 103330 ASLB R2
651 002064 013704 001264 SELTST: MOV RPAOR,R4
652 002070 032737 000001 001174 BIT #1,OPTION
653 002072 001402 BEQ 15
654 002100 000137 002236 JMP TEST1
655 002104 032737 000002 001174 15:  BIT #2,OPTION
656 002112 001402 BEQ 25
657 002114 000137 002430 JMP TEST2
658 002120 032737 000004 001174 25:  BIT #4,OPTION
659 002126 001402 BEQ 35
660 002130 000137 002740 JMP TEST3
661 002134 032737 000010 001174 35:  BIT #10,OPTION
662 002142 001402 BEQ 45
663 002144 000137 003516 JMP TEST4
664 002150 104401 001165 45:  TYPE ,SCRLF
665 002154 104401 002162 TYPE ,655
666 002160 000422 BR 645
667
668 002226 655: .ASCIZ <12>/NO TEST SELECTED - RESTART PROGRAM/
669 002226 104401 001165 645: TYPE ,SCRLF
670 002232 000000 55: HALT
671 002234 000776 BR 55
672
673 .SBTTL **** STANDARD FORMATTER ****
674
675 ;*THIS ROUTINE WILL FORMAT THE ENTIRE PACK AND THEN VERIFY
676 ;*THE HEADERS WITHIN EACH SECTOR. STARTING ADDRESS IS 200.
677
678 002236 012706 001100 TEST1: MOV #STACK,SP
679 002242 005037 177776 CLR PS
680 002246 004737 011216 JSR PC,CLRP
681 002252 004737 007426 IL1: JSR PC,UNIQ
682 002256 104401 011435 URDY1: TYPE ,SAMES
683 002262 004737 010242 JSR PC,RIN
684 002266 004737 007642 NS2: JSR PC,HONE
685 002272 012777 002366 176766 MOV #RPI,RPVEC
686 002300 012764 000001 000006 MOV #1,RPWC(R4)
687 002306 113764 001222 000005 MOVB UNIT,RPCS+1(R4)
688 002314 052764 014000 000004 BIS #14000,RPCS(R4)
689 002322 112764 000103 000004 MOVB #103,RPCS(R4)
690 002330 005037 177776 CLR PSW
691 002334 016403 000014 IL2: MOV RPA(R4),R3
692 002340 012737 177770 001206 MOV #177770,TOG1
693 002346 020364 000014 IL3: CMP R3,RPA(R4)
694 002352 001370 BNE IL2
695 002354 005337 001206 DEC TOG1

```

```

:TYPEOUT THE STATUS TABLE
:OFFLINE MESSAGE ADDRESS
:DUMMY DRIVE TYPE MESSAGE
:TYPE THE DRIVE'S STATUS
:MESSAGE ADDRESS
:TYPE THE DRIVE TYPE
:MESSAGE ADDRESS
:CR-LF
:INCREMENT THE DRIVE POINTER
:SHIFT THE DRIVE COUNTER
:BR IF NOT END
:LOAD RP11 ADDRESS
:STANDARD FORMAT AND VERIFY ?
:BR IF NOT
:DO THE TEST
:VERIFY THE PACK ?
:BR IF NOT
:DO THE TEST
:FORMAT AND VERIFY THE SPECIFIED HEADER
:BR IF NOT
:DO THE TEST
:SPECIAL FORMAT AND VERIFY
:BR IF NOT
:DO THE TEST
:TYPE ASCIZ STRING
:GET OVER THE ASCIZ
:ASCIZ <12>/NO TEST SELECTED - RESTART PROGRAM/
:SCRLF
:HALT
:INTERLOCK THE HALT

```

```

:INITIALIZE STACK POINTER
:SET THE PRIORITY TO ZERO
:CLEAR THE CONTROLLER
:GET UNIT#
:WAIT FOR KEYBOARD RESPONSE
:HOME SEEK
:RP11 VECTOR
:ALLOW DSH BREAKS
:DRIVE ADDRESS
:SET 'MODE' & 'HDR'
:WRITE WITH INTERRUPT ENABLE
:LOWER PRIORITY
:SAVE ADDRESS
:SETUP TIMEOUT
:HAS ADDRESS CHANGED?
:YES - RECYCLE
:NO - HAS 100 MSEC. TIMED OUT?

```

696	002360	001372				BNE	IL3	: NO - KEEP CHECKING
697	002362	104074				ERROR	DS!CS!ER!DA	: DISK ADDRESS NOT CHANGING IN TIME
698	002364	000740				BR	NS2	: TRY AGAIN
699	002366	032764	100000	000004	RPI:	BIT	#BIT15,RPCS(R4)	: ERROR?
700	002374	001404				BEQ	2\$: NO
701	002376	104034				ERROR	ER!CS!DS	: ERROR DURING FORMAT GENERATION
702	002400	012716	002266		1\$:	MOV	#NS2,(SP)	: CHANGE RETURN ADDRESS
703	002404	000410				BR	4\$: EXIT
704	002406	032764	000200	000004	2\$:	BIT	#BIT07,RPCS(R4)	: DONE?
705	002414	001002				BNE	3\$: YES
706	002416	104034				ERROR	DS!ER!CS	: EXTRANEIOUS INTERRUPT DURING FORMAT PASS
707	002420	000767				BR	1\$: TRY AGAIN
708	002422	012716	002450		3\$:	MOV	#STR4,(SP)	: CHANGE RETURN ADDRESS
709	002426	000002			4\$:	RTI		: EXIT TO PASS 2

.SBTTL **** PACK FORMAT VERIFY ****

```

; *THIS ROUTINE ALLOWS THE OPERATOR TO VERIFY THE FORMAT OF THE
; *ENTIRE PACK. THIS ROUTINE IS AUTOMATICALLY INITIATED
; *BY THE STANDARD FORMATTER ROUTINE (TEST1). THE STARTING
; *ADDRESS OF THIS ROUTINE IS 204.

```

718	002430	012706	001100			TEST2:	MOV	#STACK,SP	
719	002434	005037	177776				CLR	PS	: SET THE PRIORITY TO ZERO
720	002440	004737	011216				JSR	PC,CLRP	: CLEAR THE CONTROLLER
721	002444	004737	007426				JSR	PC,UNIQ	: GET THE UNIT NUMBER
722	002450	012737	000340	177776	STR4:	MOV	#PR7,PSW		: LOCKOUT INTERRUPTS
723	002456	104401	011664				TYPE	NSWRES	
724	002462	004737	010242				JSR	PC,RIN	: WAIT FOR OPERATOR
725	002466	004737	007642			P2L1:	JSR	PC,H0ME	: HOME SEEK
726	002472	005037	001226				CLR	CYL	: CYL=0
727	002476	013764	001226	000012	P2L2:	MOV	CYL,RPCA(R4)		: START AT
728	002504	005064	000014				CLR	RPDA(R4)	: SW & TAR = 0
729	002510	012764	176650	000006			MOV	#-600,RPWC(R4)	: READ ONE CYLINDER OF HEADERS INTO
730	002516	012764	012412	000010			MOV	#INBUF,RPBA(R4)	: BUFFER
731	002524	113764	001222	000005			MOVB	UNIT,RPCS+1(R4)	: DRIVE ADDRESS
732	002532	052764	014000	000004			BIS	#14000,RPCS(R4)	: 10/15 HEADER MODE
733	002540	112764	000005	000004			MOVB	#5,RPCS(R4)	: READ HEADERS
734	002546	032764	100200	000004	1\$:	BIT	#BIT15!BIT07,RPCS(R4)		: ;ERROR OR DONE ?
735	002554	001774					BEQ	1\$: NOT YET
736	002556	100004					BPL	P2N1	: DONE
737	002560	104074					ERROR	ER!DS!CS!DA	: ERROR ON READ HEADER OP.
738	002562	004737	011216				JSR	PC,CLRP	: CLEAR THE CONTROLLER
739	002566	000743					BR	P2L2	: RECYCLE
740	002570	005037	001230			P2N1:	CLR	HED	: START ADDRESS CHECKING
741	002574	005037	001232				CLR	SEC	: INITIALIZE GOOD DATA
742	002580	012703	012412				MOV	#INBUF,R3	: POINTER TO DATA READ
743	002604	004737	007760			P2L3:	JSR	PC,CHKAD	: CHECK DATA
744	002610	000240					NOP		: NO LOOPING
745	002612	005237	001232				INC	SEC	: ADVANCE SEC#
746	002616	023727	001232	000011			CMP	SEC,#9.	: OVFL0?
747	002624	101002					BHI	1\$: YES
748	002626	000137	002604				JMP	P2L3	: NO- CHECK NEXT ADDRESS
749	002632	005037	001232			1\$:	CLR	SEC	
750	002636	005237	001230				INC	HED	: ADVANCE HEAD#
751	002642	023727	001230	000023			CMP	HED,#19.	: OVFL0?

752	002650	101002			BHI	PS		:YES
753	002652	000137	002604		JMP	P2L3		:NO- CHECK NEXT ADDRESS
754	002656	005037	001230	2S:	CLR	HEL		
755	002662	001237	001226		INC	CYL		:ADVANCE CYL#
756	002666	023737	001226	001170	CMP	CYL,MAXCYL		:DONE ALL?
757	002674	101002			BHI	3S		:YES
758	002676	000137	002476		JMP	P2L2		:NO- READ ANOTHER BLOCK
759	002702	104401	012251	3S:	TYPE	ENDM1		
760	002706	032777	040000	176224	BIT	#SM14,#SWR		:DO THE TEST AGAIN
761	002714	001407			BEQ	5S		:BR IF NOT
762	002716	032737	000001	001174	BIT	#1,OPTION		:IN HERE FROM TEST 1 ?
763	002724	001402			BEQ	4S		:BR IF NOT
764	002726	000137	002256		JMP	URDY1		:DO TEST 1 AGAIN
765	002732	000555		4S:	BR	P2L1		:DO TEST 2 AGAIN
766	002734	001200		5S:	HALT			
767	002736	000775			BR	4S		:LOCK THE HALT

.SBTTL **** SINGLE HEADER FORMATTER ****

:*THIS ROUTINE ALLOWS THE OPERATOR TO SPECIFY A PARTICULAR
:*HEADER TO BE FORMATTED AND THEN VERIFIED. THE STARTING ADDR
:*IS 210.

776	002740	012706	001100		TEST3:	MOV	#STACK,SP	:SET STACK POINTER
777	002744	005037	177776			CLR	PS	:SET PRIORITY TO ZERO
778	002750	004737	011216			JSR	PC,CLRP	:CLEAR THE CONTROLLER
779	002754	004737	007426			JSR	PC,UNIQ	:GET UNIT
780	002760	104401	012010			TYPE	,HOMS	
781	002764	104401	012077		HL1:	TYPE	,OLDMS	
782	002770	013705	001170			MOV	MAXCYL,RS	:HIGHEST CYLINDER
783	002774	004737	007570			JSR	PC,PARIN	:FETCH PARAMETER
784	003000	000771				BR	HL1	:ERROR RECYCLE
785	003002	013737	001214	001242		MOV	WORK1,CYLA	:SAVE VALUE
786	003010	012705	000023			MOV	#19,RS	:HIGHEST TRACK
787	003014	004737	007570			JSR	PC,PARIN	:FETCH PARAMETER
788	003020	000761				BR	HL1	:ERROR RECYCLE
789	003022	013737	001214	001244		MOV	WORK1,HEDA	:SAVE VALUE
790	003030	004737	010122			JSR	PC,T110	:FETCH PARAMETER
791	003034	000753				BR	HL1	:ERROR RECYCLE
792	003036	020027	000011			CMP	RO,#9.	:HIGHEST SECTOR
793	003042	101350				BHI	HL1	:TOO BIG
794	003044	010027	001246			MOV	RO,SECA	:SAVE VALUE
795	003050	104401	012113		HL2:	TYPE	,NEWMS	
796	003054	013705	001170			MOV	MAXCYL,RS	:HIGHEST CYLINDER
797	003060	004737	007570			JSR	PC,PARIN	:FETCH PARAMETER
798	003064	000771				BR	HL2	:ERROR RECYCLE
799	003066	013737	001214	001226		MOV	WORK1,CYL	:SAVE VALUE
800	003074	012705	000023			MOV	#19,RS	:HIGHEST TRACK
801	003100	004737	007570			JSR	PC,PARIN	:FETCH PARAMETER
802	003104	000761				BR	HL2	:ERROR RECYCLE
803	003106	013737	001214	001230		MOV	WORK1,HED	:SAVE VALUE
804	003114	004737	010122			JSR	PC,T110	:FETCH PARAMETER
805	003120	000753				BR	HL2	:ERROR RECYCLE
806	003122	020027	000011			CMP	RO,#9.	:HIGHEST SECTOR
807	003126	101350				BHI	HL2	:TOO BIG

808	003130	010037	001232		MOV	RO, SEC	:SAVE VALUE
809	003134	005037	012412		CLR	INBUF	:HWRD1=0
810	003140	013737	001232	012416	HGO:	MOV	SEC, INBUF+4
811	003146	013737	001226	012414	MOV	CYL, INBUF+2	:HWRD3=SECTOR
812	003154	006137	012414		ROL	INBUF+2	
813	003160	006137	012414		ROL	INBUF+2	
814	003164	006137	012414		ROL	INBUF+2	
815	003170	006137	012414		ROL	INBUF+2	
816	003174	006137	012414		ROL	INBUF+2	
817	003200	006137	012414		ROL	INBUF+2	
818	003204	006137	001230		ROL	HED	
819	003210	053737	001230	012414	BIS	HED, INBUF+2	
820	003216	006037	001230		ROR	HED	
821	003222	042737	100001	012414	BIC	#100001, INBUF+2	
822	003230	004737	007642		HGL1:	JSR	PC, HOME
823	003234	012764	177775	000006	MOV	#-3, RPWC(R4)	:HOME SEEK
824	003242	012764	012412	000010	MOV	#INBUF, RPBA(R4)	:WORD COUNT
825	003250	113764	001222	000005	MOVB	UNIT, RPCS+1(R4)	:BUFFER ADDRESS
826	003256	052764	014000	000004	BIS	#14000, RPCS(R4)	:LOAD DRIVE NUMBER
827	003264	013764	001242	000012	MOV	CYLA, RPCA(R4)	:10/15 MODE HEADER OP
828	003272	113764	001244	000015	MOVB	HEDA, RPDA+1(R4)	:LOAD CYLINDER
829	003300	113764	001246	000014	MOVB	SECA, RPDA(R4)	:TRACK
830	003306	112764	000003	000004	MOVB	#3, RPCS(R4)	:AND SECTOR TO BE CHANGED
831	003314	032764	100200	000004	IS:	BIT	#BIT15:BIT07, RPCS(R4)
832	003322	001774			BEQ	IS	:REWRITE IT
833	003324	100004			BPL	HGO1	:;ERROR OR DONE?
834	003326	104074			ERROR	ER!DS!CS!DA	:NOT YET
835	003330	004737	011216		JSR	PC, CLR	:DONE
836	003334	000735			BR	HGL1	:ERROR ON WRITE HEADER OP
837	003336	005037	012412		HGO1:	CLR	:CLEAR THE CONTROLLER
838	003342	005037	012414		CLR	INBUF	:TRY AGAIN
839	003346	005037	012416		CLR	INBUF+2	:DESTROY
840	003352	004737	007642		HGL2:	JSR	:OLD
841	003356	012764	177775	000006	MOV	PC, HOME	:DATA
842	003364	012764	012412	000010	MOV	#-3, RPWC(R4)	:HOME SEEK
843	003372	013764	001242	000012	MOV	#INBUF, RPBA(R4)	:WORD COUNT
844	003400	113764	001244	000015	MOVB	CYLA, RPCA(R4)	:BUFFER
845	003406	113764	001246	000014	MOVB	HEDA, RPDA+1(R4)	:DISK
846	003414	113764	001222	000005	MOVB	SECA, RPDA(R4)	:ADDRESS
847	003422	052764	014000	000004	BIS	#14000, RPCS(R4)	:TO BE FOUND
848	003430	112764	000005	000004	MOVB	UNIT, RPCS+1(R4)	:LOAD DRIVE NUMBER
849	003436	032764	100200	000004	IS:	BIT	:10/15 MODE HEADER OP
850	003444	001774			BEQ	IS	:READ
851	003446	100004			BPL	HGO2	:;ERROR OR DONE?
852	003450	104074			ERROR	ER!DS!CS!DA	:NOT YET
853	003452	004737	011216		JSR	PC, CLR	:DONE
854	003456	000735			BR	HGL2	:ERROR ON READING NEW HEADER
855	003460	012703	012412		HGO2:	MOV	:CLEAR THE CONTROLLER
856	003464	004737	007760		JSR	PC, CHKAD	:TRY AGAIN
857	003470	000730			BR	HGL2	:R3=DATA POINTER
858	003472	104401	012251		TYPE	ENDM1	:CHECK REWRITTEN HEADER
859	003476	032777	040000	175434	BIT	#SM14, JSR	:ERROR - TRY AGAIN
860	003504	001402			BEQ	IS	:DO THE TEST AGAIN ?
861	003506	000137	003134		JMP	HGO	:BR IF NOT
862	003512	000000			IS:	HALT	:TEST 3 AGAIN
863	003514	000776			BR	IS	:LOCK THE HALT

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919

003516 012706 001100
003522 005037 177776
003526 004737 011216
003532 004737 007426
003536 104401 011251
003542 004737 004242
003546 012737 177766 001206
003554 012705 004504
003560 012703 004530
003564 012500
003566 004737 011204
003572 004737 004262
003576 000757
003600 005237 001206
003604 001367
003606 004737 004332
003612 004737 007642
003616 005037 001226
003622 013764 001226 000012
003630 005064 000014
003634 004737 004430
003640 000762
003642 012764 176650 000006
003650 012764 014712 000010
003656 113764 001222 000005
003664 052764 014000 000004
003672 112764 000003 000004
003700 032764 100200 000004
003706 001774
003710 100004
003712 104074
003714 004737 011216
003720 000740
003722 012737 177470 001206
003730 012705 014714
003734 062715 000100
003740 062705 000006
003744 005237 001206
003750 011371
003752 005237 001226
003756 023737 001226 001170
003764 101001
003766 000715
003770 012737 000340 177776
003776 004737 007642
004002 005037 001226
004006 013764 001226 000012
004014 005064 000014

.SBTTL **** SPECIAL FORMATTER ****

;*THIS ROUTINE WILL FORMAT THE ENTIRE PACK USING THE SECTOR
;*ADDRESS SEQUENCE SPECIFIED BY THE OPERATOR. THE STARTING
;*ADDRESS IS 214.

TEST4: MOV #STACK, SP ;SET STACK
CLR PS ;SET PRIORITY TO ZERO
JSR PC, CLRPC ;CLEAR THE CONTROLLER
JSR PC, UNIQ ;GET UNIT#
SFSL1: TYPE SECRES
JSR PC, CLRCS ;CLEAR FLAG STRING
MOV #-10, TOG1 ;DO BELOW 10 TIMES
MOV #NSTR, R5 ;POINTER TO MESSAGE POINTERS
MOV #SCSL1, R3
SFSL2: MOV (R5)+, R0 ;FETCH POINTER
JSR PC, TYP0UT ;FOR MESSAGE
JSR PC, SIN ;GET PARAMETER
BR SFSL1 ;ERROR
INC TOG1 ;DONE?
BNE SFSL2 ;NOT YET
SFSL3: JSR PC, MKBUF ;GENERATE A TABLE OF ADDRESSES
JSR PC, HOME ;RESET SELECTED UNIT
CLR CYL ;START AT CYL=0
SFSL4: MOV CYL, RPCA(R4) ;LOAD
CLR RPOA(R4) ;DISK ADDRESS
JSR PC, SEKE ;SEEK
BR SFSL3 ;ERROR RETURN
MOV #-600, RPWC(R4) ;GOOD RETURN
MOV #OUTBUF, RPB(R4) ;WC&CA
MOVB UNIT, RPS+1(R4) ;LOAD DRIVE NUMBER
BIS #14000, RPS(R4) ;10/15 HEADER MODE
MOVB #3, RPS(R4) ;WRITE
1\$: BIT #BIT15:BIT07, RPS(R4) ;ERROR OR DONE
BEQ 1\$;WAIT
BPL SFSN1 ;DONE
ERROR ER:DS!CS!DA ;ERROR ON WRITE HEADER OF
JSR PC, CLRPC ;CLEAR THE CONTROLLER
BR SFSL4 ;REPEAT
SFSN1: MOV #-200, TOG1
MOV #OUTBUF+2, R5
SFSL5: ADD #100, (R5) ;ADVANCE CYLINDER ADDRESS
ADD #6, R5 ;INDEX
INC TOG1 ;DONE?
BNE SFSL5 ;NO
INC CYL ;ADVANCE CYLINDER
CMP CYL, MAXCYL ;DONE?
BHI SFSCHK ;YES
BR SFSL4 ;NO
SFSCHK: MOV #PR7, PSW ;LOCKOUT INTERRUPTS
JSR PC, HOME ;HOME SEEK
CLR CYL ;CYL=0
SFSCL1: MOV CYL, RPCA(R4) ;TAR=0
CLR RPOA(R4) ;SAR=0

```

920 004020 004737 004430 JSR PC,SEKE ;SEEK
921 004024 000761 BR SFCHK ;ERROR RETURN
922 004026 012764 176650 000006 MOV #-600.,RPMC(R4) ;MC
923 004034 012764 012412 000010 MOV #INBUF,RPBA(R4) ;CA
924 004042 113764 001222 000005 MOVB UNIT,RPCS+1(R4) ;LOAD DRIVE NUMBER
925 004050 052764 014000 000004 BIS #14000,RPCS(R4) ;ID/15 HEADER OP
926 004056 112764 000005 000004 MOVB #5,RPCS(R4) ;READ
927 004064 032764 100200 000004 1$: BIT #BIT15:BIT07,RPCS(R4) ;ERROR OR DONE?
928 004072 001774 BEQ 1$ ;NOT YET
929 004074 100004 BPL SFCN1 ;JUMP IF DONE
930 004076 104074 ERROR ER:DS!CS!DA ;READ HEADER ERROR
931 004100 004737 011216 JSR PC,CLRP ;CLEAR THE CONTROLLER
932 004104 000740 BR SFCL1 ;LOOP
933 004106 012705 004530 SFCN1: MOV #SCSLT,R5 ;SETUP SECTOR LIST AND
934 004112 012737 177766 001206 MOV #-10.,TOG1 ;COUNTER
935 004120 005037 001230 CLR HED ;SETUP GOOD DATA
936 004124 012703 012412 MOV #INBUF,R3 ;SETUP BUFFER POINTER
937 004130 012537 001232 SFCL2: MOV (R5)+,SEC ;FETCH A SECTOR
938 004134 004737 007760 JSR PC,CHKAD ;CHECK ADDRESS
939 004140 000240 NOP ;DON'T USE ERROR RETURN
940 004142 005237 001206 INC TOG1 ;DONE ONE SURFACE?
941 004146 001370 BNE SFCL2 ;NO
942 004150 012705 004530 MOV #SCSLT,R5 ;YES - RESET
943 004154 012737 177766 001206 MOV #-10.,TOG1 ;SECTOR PARAMETERS
944 004162 005237 001230 INC HED ;ADVANCE TRACK
945 004166 023727 001230 000023 CMP HED,#19. ;DONE?
946 004174 101755 BLOS SFCL2 ;NO
947 004176 005037 001230 CLR HED ;YES - RESET HEAD
948 004202 005237 001226 INC CYL ;ADVANCE CYLINDER
949 004206 023737 001226 001170 CMP CYL,MAXCYL ;DONE?
950 004214 101674 BLOS SFCL1 ;NOT YET
951 004216 104401 012251 TYPE ENDM1
952 004222 032777 040000 174710 BIT #SW14,3SWR ;DO THE TEST AGAIN?
953 004230 001402 BEQ 1$ ;BR IF NOT
954 004232 000137 003606 JMP SFSL3 ;DO TEST 4 AGAIN
955 004236 000000 1$: HALT
956 004240 000776 BR 1$ ;LOCK THE HALT
957
958 ;CLEAR THE SECTOR ADDRESS SEQUENCE TABLES
959
960 004242 012705 177742 CLRS: MOV #-30.,R5 ;CLEAR
961 004246 012703 004530 MOV #SCSLT,R3 ;THE SECTOR
962 004252 105023 1$: CLRB (R3)+ ;SLOTS AND
963 004254 005205 INC R5 ;FLAG POINTERS
964 004256 001375 BNE 1$ ;EXIT
965 004260 000207 RTS PC ;WHEN DONE
966
967 ;GET THE SECTOR ADDRESS SEQUENCE
968
969 004262 004737 010242 SIN: JSR PC,RIN ;ASSEMBLE
970 004266 004737 011154 JSR PC,TTO ;ECHO
971 004272 162700 000060 SUB #60,R0 ;A LIST OF
972 004276 020027 000011 CMP R0,#9. ;LOGICAL SECTOR
973 004302 101012 BHI 1$ ;NUMBERS
974 004304 005700 TST R0 ;TOO SMALL?
975 004306 100410 BMI 1$ ;0-9 ARE LEGAL

```

```

976 004310 105760 004554      TSTB   SCFLG(R0)      ;HAS THIS NUMBER BEEN USED?
977 004314 001005          BNE    1$            ;ERROR - THIS NUMBER ALREADY USED
978 004316 105160 004554      COMB   SCFLG(R0)      ;SET FLAG
979 004322 010023          MOV    R0,(R3)+      ;STORE SEC#
980 004324 062716 000002      ADD    #2,(SP)       ;ADVANCE RETURN
981 004330 000207          1$:    RTS    PC      ;AND EXIT
982
983      ;GENERATE THE SECTOR ADDRESS BUFFER
984
985 004332 012737 177754 001206 MKBUF: MOV    #-20,TOG1      ;TOG1=COUNTER
986 004340 012705 014712          MOV    #OUTBUF,R5    ;R5=POINTER TO BUFFER
987 004344 012703 004530          MOV    #SCSLT,R3     ;R3=POINTER TO SECTOR LIST
9  8 004350 012737 177766 001210          MOV    #-10,TOG2     ;TOG2=SECTOR COUNT
989 004356 005037 001212          CLR    WORK
990 004362 005025          1$:    CLR    (R5)+        ;WORD1=0
991 004364 013725 001212          MOV    WORK,(R5)+    ;WORK2=CYL+HEAD
992 004370 012325          MOV    (R3)+,(R5)+  ;WORD3=SECTOR
993 004372 005237 001210          INC    TOG2         ;DONE 10?
994 004376 001371          BNE    1$            ;NO
995 004400 012703 004530          MOV    #SCSLT,R3     ;YES - UPDATE POINTER,
996 004404 012737 177766 001210          MOV    #-10,TOG2    ;COUNTER,
997 004412 052737 000002 001212          ADD    #2,WORK      ;AND TRACK
998 004420 005237 001206          INC    TOG1         ;FINISHED?
999 004424 001356          BNE    1$            ;NO
1000 004426 000207          RTS    PC            ;EXIT
1001
1002      ;SEEK TO THE NEXT CYLINDER
1003
1004 004430 112764 000377 000000 SEKE:  MOVB   #377,RPDS(R4) ;CLEAR ATTN
1005 004436 112764 000011 000004          MOVB   #11,RPCS(R4) ;SEEK
1006 004444 033764 001224 000000 1$:    BIT    ATNB,RPDS(R4) ;WAIT FOR
1007 004452 001774          BEQ    1$            ;ATTN
1008 004454 112764 000377 000000          MOVB   #377,RPDS(R4) ;CLEAR ATTN
1009 004462 032764 004000 000000          BIT    #BIT11,RPDS(R4) ;DONE?
1010 004470 001402          BEQ    2$            ;SEEK INCOMPLETE
1011 004472 104074          ERROR  ER!CS!DS!DA   ;ERROR EXIT
1012 004474 000207          RTS    PC
1013 004476 062716 000002 2$:    ADD    #2,(SP)      ;GOOD
1014 004502 000207          RTS    PC            ;EXIT
1015
1016      ;USER DEFINED SECTOR SEQUENCE TABLES
1017
1018 004504 011341          NSTR:  .WORD  N0
1019 004506 011347          .WORD  N1
1020 004510 011355          .WORD  N2
1021 004512 011363          .WORD  N3
1022 004514 011371          .WORD  N4
1023 004516 011377          .WORD  N5
1024 004520 011405          .WORD  N6
1025 004522 011413          .WORD  N7
1026 004524 011421          .WORD  N8
1027 004526 011427          .WORD  N9
1028
1029 004530          SCSLT: .WORD  0
1030 004530 000000          .WORD  0
1031 004532 000000          .WORD  0

```

1032 004534 000000
1033 004536 000000
1034 004540 000000
1035 004542 000000
1036 004544 000000
1037 004546 000000
1038 004550 000000
1039 004552 000000
1040
1041 004554
1042 004554 000
1043 004555 000
1044 004556 000
1045 004557 000
1046 004560 000
1047 004561 000
1048 004562 000
1049 004563 000
1050 004564 000
1051 004565 000
1052
1053
1054
1055
1056
1057
1058
1059 004566 000000
1060 004570 000000
1061 004572 000000
1062 004574 000001
1063 004575
1064 004576
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074 004576 005037 004566
1075 004602 012737 004574 004570
1076 004610 013737 004570 004572
1077 004616 012737 004646 000060
1078 004624 012737 000200 000062
1079 004632 005777 174310
1080 004636 012777 000100 174300
1081 004644 000207
1082
1083
1084
1085
1086
1087

.WORD 0
.WORD 0
.WORD 0
.WORD 0
.WORD 0
.WORD 0
.WORD 0
.WORD 0
.WORD 0

SCFLG:
.BYTE 0
.BYTE 0
.BYTE 0
.BYTE 0
.BYTE 0
.BYTE 0
.BYTE 0
.BYTE 0
.BYTE 0
.BYTE 0

.SBTTL **** SUBROUTINES ****
.SBTTL TTY INPUT ROUTINE

;ENABL LSB
\$TKCNT: .WORD 0 ;: NUMBER OF ITEMS IN QUEUE
\$TKQIN: .WORD 0 ;: INPUT POINTER
\$TKQOUT: .WORD 0 ;: OUTPUT POINTER
\$TKQSRV: .BLKB 1 ;: TTY KEYBOARD QUEUE
\$TKQEND=.
.EVEN

;*TK INITIALIZE ROUTINE
;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT

;*CALL:
;* JSR PC,\$TKINT
;* RETURN

\$TKINT: CLR \$TKCNT ;: CLEAR COUNT OF ITEMS IN QUEUE
MOV \$TKQSRV,\$TKQIN ;: MOVE THE STARTING ADDRESS OF THE
MOV \$TKQIN,\$TKQOUT ;: QUEUE INTO THE INPUT & OUTPUT POINTERS.
MOV \$TKQSRV,\$TKQVEC ;: INITIALIZE THE KEYBOARD VECTOR
MOV \$200,\$TKQVEC+2 ;: "BR" LEVEL 4
TST \$TKB ;: CLEAR DONE FLAG
MOV \$100,\$TKS ;: ENABLE TTY KEYBOARD INTERRUPT
RTS PC ;: RETURN TO CALLER

;*TK SERVICE ROUTINE
;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
;*IT IN THE QUEUE.
;

```

1088 004646 117746 174274 STKSRV: MOVB @STKB, -(SP)      ;; PICKUP THE CHARACTER
1089 004652 042716 177600      BIC #1C177, (SP)      ;; STRIP THE JUNK
1090 004656 021627 000007      1S:  CMP (SP), #7      ;; IS IT A CONTROL G?
1091 004662 001004      BNE 2S      ;; BRANCH IF NO
1092 004664 022737 000176 001140  CMP #SWREG, SWR      ;; IS SOFT-SWR SELECTED?
1093 004672 001500      BEQ 6S      ;; GO TO SWR CHANGE
1094
1095 004674      2S:
1096 004674 022737 000001 004566  CMP #1, STKCNT      ;; IS THE QUEUE FULL?
1097 004702 001004      BNE 3S      ;; BRANCH IF NO
1098 004704 104401 001160      TYPE ,SBELL      ;; RING THE TTY BELL
1099 004710 005726      TST (SP)+      ;; CLEAN CHARACTER OFF OF STACK
1100 004712 000451      BR 5S      ;; EXIT
1101 004714 021627 000023      3S:  CMP (SP), #23      ;; IS IT A CONTROL-S?
1102 004720 001021      BNE 32S      ;; BRANCH IF NO
1103 004722 005077 174216      CLR @STKS      ;; DISABLE TTY KEYBOARD INTERRUPTS
1104 004726 005726      TST (SP)+      ;; CLEAN CHAR OFF STACK
1105 004730 105777 174210      31S: TSTB @STKS      ;; WAIT FOR A CHAR
1106 004734 100375      BPL 31S      ;; LOOP UNTIL ITS THERE
1107 004736 117746 174204      MOVB @STKB, -(SP)      ;; GET THE CHARACTER
1108 004742 042716 177600      BIC #1C177, (SP)      ;; MAKE IT 7-BIT ASCII
1109 004746 022627 000021      CMP (SP)+, #21      ;; IS IT A CONTROL-Q?
1110 004752 001366      BNE 31S      ;; BRANCH IF NO
1111 004754 012777 000100 174162  MOV #100, @STKS      ;; REENABLE TTY KEYBOARD INTERRUPTS
1112 004762 000702      RTI      ;; RETURN
1113 004764 005237 004566      32S: INC STKCNT      ;; COUNT THIS CHARACTER
1114 004770 021627 000140      CMP (SP), #140      ;; IS IT UPPER CASE?
1115 004774 002405      BLT 4S      ;; BRANCH IF YES
1116 004776 021627 000175      CMP (SP), #175      ;; IS IT A SPECIAL CHAR?
1117 005002 003002      BGT 4S      ;; BRANCH IF YES
1118 005004 042716 000040      BIC #40, (SF)      ;; MAKE IT UPPER CASE
1119 005010 112677 177554      4S:  MOVB (SP)+, @STKQIN      ;; AND PUT IT IN QUEUE
1120 005014 005237 004570      INC STKQIN      ;; UPDATE THE POINTER
1121 005020 023727 004570 004575  CMP STKQIN, #STKQEND      ;; GO OFF THE END?
1122 005026 001003      BNE 5S      ;; BRANCH IF NO
1123 005030 012737 004574 004570  MOV #STKQSR, STKQIN      ;; RESET THE POINTER
1124 005036 000002      5S:  RTI      ;; RETURN
1125
1126      ;; *****
1127      ;; *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
1128      ;; *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
1129      ;; *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
1130      ;; *CALL WHEN OPERATING IN TTY INTERRUPT MODE.
1131 005040 022737 000176 001140 $CKSWR: CMP #SWREG, SWR      ;; IS THE SOFT-SWR SELECTED
1132 005046 001104      BNE 15S      ;; EXIT IF NOT
1133 005050 105777 174070      TSTB @STKS      ;; IS A CHAR WAITING?
1134 005054 100101      BPL 15S      ;; IF NOT, EXIT
1135 005056 117746 174064      MOVB @STKB, -(SP)      ;; YES
1136 005062 042716 177600      BIC #1C177, (SP)      ;; MAKE IT 7-BIT ASCII
1137 005066 021627 000007      CMP (SP), #7      ;; IS IT A CONTROL-G?
1138 005072 001300      BNE 2S      ;; IF NOT, PUT IT IN THE TTY QUEUE
1139      ;; AND EXIT
1140
1141      ;; *****
1142      ;; *CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
1143      ;; *ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A

```

```

1144                                     : *CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
1145 005074 123727 001134 000001 6S:  CMPB  $AUTOB, #1      : ARE WE RUNNING IN AUTO-MODE?
1146 005102 001674                                     : BEQ  2S              : BRANCH IF YES
1147 005104 005726                                     : TST  (SP)+          : CLEAR CONTROL-G OFF STACK
1148 005106 004737 J04576                               : JSR  PC, $TKINT     : FLUSH THE TTY INPUT QUEUE
1149 005112 005077 174026                               : CLR  @STKS          : DISABLE TTY KEYBOARD INTERRUPTS
1150 005116 112737 000001 001135                               : MOVB #1, $INTAG     : SET INTERRUPT MODE INDICATOR
1151
1152 005124 104401 005702                                     : TYPE , $CNTLG       : ECHO THE CONTROL-G (↑G)
1153 005130 104401 005707 SGTSWR: TYPE , $MSWR       : TYPE CURRENT CONTENTS
1154 005134 013746 000176                               : MOV  $SWREG, -(SP)  : SAVE SWREG FOR TYPEOUT
1155 005140 104402                                     : TYPOC              : GO TYPE--OCTAL ASCII(ALL DIGITS)
1156 005142 104401 005720                                     : TYPE , $MNEW        : PROMPT FOR NEW SWR
1157 005146 005046 19S:  CLR  -(SP)              : CLEAR COUNTER
1158 005150 005046 7S:   CLR  -(SP)              : THE NEW SWR
1159 005152 105777 173766                               : TSTB @STKS         : CHAR THERE?
1160 005156 100375                                     : BPL  7S             : IF NOT TRY AGAIN
1161
1162 005160 117746 173762                               : MOVB @STKB, -(SP)  : PICK UP CHAR
1163 005164 042716 177600                               : BIC  #1C177, (SP)  : MAKE IT 7-BIT ASCII
1164
1165
1166
1167 005170 021627 000025 9S:  CMP  (SP), #25      : IS IT A CONTROL-U?
1168 005174 001005                                     : BNE  10S            : BRANCH IF NOT
1169 005176 104401 005675                                     : TYPE , $CNTLU       : YES, ECHO CONTROL-U (↑U)
1170 005202 062706 000006 20S:  ADD  #6, SP          : IGNORE PREVIOUS INPUT
1171 005206 000757                                     : BR  19S             : LET'S TRY IT AGAIN
1172
1173
1174 005210 021627 000015 10S:  CMP  (SP), #15      : IS IT A <CR>?
1175 005214 001022                                     : BNE  16S            : BRANCH IF NO
1176 005216 005766 000004                                     : TST  4(SP)          : YES, IS IT THE FIRST CHAR?
1177 005222 001403                                     : BEQ  11S            : BRANCH IF YES
1178 005224 016677 000002 173706                               : MOV  2(SP), @SWR    : SAVE NEW SWR
1179 005232 062706 000006 11S:  ADD  #6, SP          : CLEAR UP STACK
1180 005236 104401 001165 14S:  TYPE , $CNTLF       : ECHO <CR> AND <LF>
1181 005242 123727 001135 000001                               : CMPB $INTAG, #1     : RE-ENABLE TTY KBD INTERRUPTS?
1182 005250 001003                                     : BNE  15S            : BRANCH IF NOT
1183 005252 012777 000100 173664                               : MOV  #100, @STKS    : RE-ENABLE TTY KBD INTERRUPTS
1184 005260 000002 15S:  RTI                                     : RETURN
1185 005262 004737 006242 16S:  JSR  PC, $TYPEPC    : ECHO CHAR
1186 005266 021627 000060                                     : CMP  (SP), #60      : CHAR < 0?
1187 005272 002420                                     : BLT  18S            : BRANCH IF YES
1188 005274 021627 000067                                     : CMP  (SP), #67      : CHAR > ??
1189 005300 003015                                     : BGT  18S            : BRANCH IF YES
1190 005302 042726 000060                                     : BIC  #60, (SP)+     : STRIP-OFF ASCII
1191 005306 005766 000002                                     : TST  2(SP)          : IS THIS THE FIRST CHAR
1192 005312 001403                                     : BEQ  17S            : BRANCH IF YES
1193 005314 006316                                     : ASL  (SP)           : NO, SHIFT PRESENT
1194 005316 006316                                     : ASL  (SP)           : CHAR OVER TO MAKE
1195 005320 006316                                     : ASL  (SP)           : ROOM FOR NEW ONE.
1196 005322 005266 000002 17S:  INC  2(SP)          : KEEP COUNT OF CHAR
1197 005326 056616 177776                                     : BIS  -2(SP), (SP)   : SET IN NEW CHAR
1198 005332 000707                                     : BR  7S              : GET THE NEXT ONE
1199 005334 104401 001164 18S:  TYPE , $QUES       : TYPE ?<CR><LF>

```

```

1200 005340 000720          BR      205          ;;SIMULATE CONTROL-U
1201          .DSABL  L58
1202
1203
1204          ;*****
1205          ;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
1206          ;CALL:
1207          ;      RDCHR          ;;GET A CHARACTER FROM THE QUEUE
1208          ;      RETURN HERE   ;;CHARACTER IS ON THE STACK
1209          ;                  ;;WITH PARITY BIT STRIPPED OFF
1210          ;
1211          ;
1212 005342 011646          SRDCHR: MOV      (SP),-(SP)          ;;PUSH DOWN THE PC AND
1213 005344 016666 000004 000002      MOV      4(SP),2(SP)          ;;THE PS
1214 005352 005066 000004          CLR      4(SP)          ;;GET READY FOR A CHARACTER
1215 005356 005046          CLR      -(SP)          ;;PUT NEW PS ON STACK
1216 005360 012746 005366          MOV      #645,-(SP)          ;;PUT NEW PC ON STACK
1217 005364 000002          RTI          ;;POP NEW PC AND PS
1218 005366
1219 005366 005737 004566      645:   TST      STKCNT          ;;WAIT ON A CHARACTER
1220 005372 001775          15:    BEQ      15
1221 005374 005337 004566          DEC      STKCNT          ;;DECREMENT THE COUNTER
1222 005400 117766 1771'6 000004      MOV8    #STKQOUT,4(SP)      ;;GET ONE CHARACTER
1223 005406 005237 0045.2          INC      STKQOUT          ;;UPDATE THE POINTER
1224 005412 023727 004572 004575      CMP      STKQOUT,#STKQEND  ;;DID IT GO OFF OF THE END?
1225 005420 001003          BNE      25              ;;BRANCH IF NO
1226 005422 012737 004574 004572      MOV      #STKQSRRT,STKQOUT ;;RESET THE POINTER
1227 005430 000002          25:    RTI          ;;RETURN
1228          ;*****
1229          ;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
1230          ;CALL:
1231          ;      RDLIN          ;;INPUT A STRING FROM THE TTY
1232          ;      RETURN HERE   ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
1233          ;                  ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
1234          ;
1235 005432 010346          SRDLIN: MOV      R3,-(SP)          ;;SAVE R3
1236 005434 005046          CLR      -(SP)          ;;CLEAR THE RUBOUT KEY
1237 005436 012703 005666      15:    MOV      #STTYIN,R3          ;;GET ADDRESS
1238 005442 022703 005675      25:    CMP      #STTYIN+7,R3      ;;BUFFER FULL?
1239 005446 101456          BLOS    45              ;;BR IF YES
1240 005450 104407          RDCHR          ;;GO READ ONE CHARACTER FROM THE TTY
1241 005452 112613          MOV8    (SP)+,(R3)          ;;GET CHARACTER
1242 005454 122713 000177      105:   CMP8    #177,(R3)          ;;IS IT A RUBOUT
1243 005460 001022          BNE     55              ;;BR IF NO
1244 005462 005716          TST     (SP)          ;;IS THIS THE FIRST RUBOUT?
1245 005464 001007          BNE     65              ;;BR IF NO
1246 005466 112737 000134 005664      MOV8    #'\\,95          ;;TYPE A BACK SLASH
1247 005474 104401 005664          TYPE    95
1248 005500 012716 177777          MOV     #-1,(SP)          ;;SET THE RUBOUT KEY
1249 005504 005303      65:   DEC     R3              ;;BACKUP BY ONE
1250 005506 020327 005666          CMP     R3,#STTYIN          ;;STACK EMPTY?
1251 005512 103434          BLO     45              ;;BR IF YES
1252 005514 111337 005664          MOV8    (R3),95          ;;SETUP TO TYPEOUT THE DELETED CHAR.
1253 005520 104401 005664          TYPE    95              ;;GO TYPE
1254 005524 000746          BR      25              ;;GO READ ANOTHER CHAR.
1255 005526 005716          55:   TST     (SP)          ;;RUBOUT KEY SET?

```

```

1256 005530 001406 BEQ 7$ :: BR IF NO
1257 005532 112737 000134 005664 MOVB #'\,9$ :: TYPE A BACK SLASH
1258 005540 104401 005664 TYPE 9$
1259 005544 005016 CLR (SP) :: CLEAR THE RUBOUT KEY
1260 005546 122713 000025 7$: CMPB #25,(R3) :: IS CHARACTER A CTRL U?
1261 005552 001003 BNE 8$ :: BR IF NO
1262 005554 104401 005675 TYPE $CNTLU :: TYPE A CONTROL "U"
1263 005560 000726 BR 1$ :: GO START OVER
1264 005562 122713 000022 8$: CMPB #22,(R3) :: IS CHARACTER A "R"?
1265 005566 001011 BNE 3$ :: BRANCH IF NO
1266 005570 105013 CLRB (R3) :: CLEAR THE CHARACTER
1267 005572 104401 001165 TYPE , $CRLF :: TYPE A "CR" & "LF"
1268 005576 104401 005666 TYPE $TTYIN :: TYPE THE INPUT STRING
1269 005602 000717 BR 2$ :: GO PICKUP ANOTHER CHACTER
1270 005604 104401 001164 4$: TYPE $QUES :: TYPE A '?'
1271 005610 000712 BR 1$ :: CLEAR THE BUFFER AND LOOP
1272 005612 111337 005664 3$: MOVB (R3),9$ :: ECHO THE CHARACTER
1273 005616 104401 005664 TYPE 9$
1274 005622 122723 000015 CMPB #15,(R3)+ :: CHECK FOR RETURN
1275 005626 001305 BNE 2$ :: LOOP IF NOT RETURN
1276 005630 105063 177777 CLRB -1(R3) :: CLEAR RETURN (THE 15)
1277 005634 104401 001166 TYPE $LF :: TYPE A LINE FEED
1278 005640 005726 TST (SP)+ :: CLEAN RUBOUT KEY FROM THE STACK
1279 005642 012603 MOV (SP)+,R3 :: RESTORE R3
1280 005644 011646 MOV (SP)-,(SP) :: ADJUST THE STACK AND PUT ADDRESS OF THE
1281 005646 016666 000004 000002 MOV 4(SP),2(SP) :: FIRST ASCII CHARACTER ON IT
1282 005654 012766 005666 000004 MOV #TTYIN,4(SP)
1283 005662 000002 RTI :: RETURN
1284 005664 000 9$: .BYTE 0 :: STORAGE FOR ASCII CHAR. TO TYPE
1285 005665 000 .BYTE 0 :: TERMINATOR
1286 005666 000007 $TTYIN: .BLKB 7 :: RESERVE 7 BYTES FOR TTY INPUT
1287 005675 136 006525 000012 $CNTLU: .ASCIZ /?U<15><12> :: CONTROL "U"
1288 005702 043536 005015 000 $CNTLG: .ASCIZ /?G<15><12> :: CONTROL "G"
1289 005707 015 051412 051127 $MSWR: .ASCIZ <15><12>/SWR = /
1290 005714 036440 000040 $MNEW: .ASCIZ / NEW = /
1291 005720 020040 042516 020127 .EVEN
1292 005726 020075 000
1293 005732
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309 005732 011646 $RDOCT: MOV (SP)-,(SP) :: READ AN OCTAL NUMBER
1310 005734 016666 000004 000002 MOV 4(SP),2(SP) :: LOW ORDER BITS ARE ON TOP OF THE STACK
1311 005742 010046 MOV R0,-(SP) :: HIGH ORDER BITS ARE IN $HIOCT
:: PROVIDE SPACE FOR THE
:: INPUT NUMBER
:: PUSH R0 ON STACK

```

```

*****
*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
*CHANGE IT TO BINARY.
*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
*CALL:
*   RDOCT
*   RETURN HERE
*

```

```

1312 005744 010146      MOV      R1,-(SP)      ;; PUSH R1 ON STACK
1313 005746 010246      MOV      R2,-(SP)      ;; PUSH R2 ON STACK
1314 005750 104410      15:     RDLIN          ;; READ AN ASCIZ LINE
1315 005752 012600      MOV      (SP)+,R0      ;; GET ADDRESS OF 1ST CHARACTER
1316 005754 010037 006060      MOV      R0,5$        ;; AND SAVE IT
1317 005760 005001      CLR      R1           ;; CLEAR DATA WORD
1318 005762 005002      CLR      R2
1319 005764 112046      25:     MOVB      (R0)+,-(SP)  ;; PICKUP THIS CHARACTER
1320 005766 001420      BEQ      3$          ;; IF ZERO GET OUT
1321 005770 122716 000060      C'FD      #'0,(SP)    ;; MAKE SURE THIS CHARACTER
1322 005774 003026      B'F      4$          ;; IS AN OCTAL DIGIT
1323 005776 122716 000067      C'PB      #'7,(SP)
1324 005002 002423      BLT      4$
1325 005004 006301      ASL      R1           ;; *2
1326 005006 006102      ROL      R2
1327 006010 006301      ASL      R1           ;; *4
1328 006012 006102      ROL      R2
1329 005014 006301      ASL      R1           ;; *8
1330 005016 006102      ROL      R2
1331 005020 042716 177770      BIC      #'07,(SP)    ;; STRIP THE ASCII JUNK
1332 005024 062601      ADD      (SP)+,R1     ;; ADD IN THIS DIGIT
1333 005026 000756      BR       2$          ;; LOOP
1334 005030 005726      35:     TST      (SP)+       ;; CLEAN TERMINATOR FROM STACK
1335 005032 010166 000012      MOV      R1,12(SP)    ;; SAVE THE RESULT
1336 005036 010237 006070      MOV      R2,$HI OCT
1337 005042 012602      MOV      (SP)+,R2     ;; POP STACK INTO R2
1338 005044 012601      MOV      (SP)+,R1     ;; POP STACK INTO R1
1339 005046 012600      MOV      (SP)+,R0     ;; POP STACK INTO R0
1340 006050 000002      RTI
1341 006052 005726      45:     TST      (SP)+       ;; CLEAN PARTIAL FROM STACK
1342 006054 105010      CLRB     (R0)         ;; SET A TERMINATOR
1343 006056 104401      TYPE     ;; TYPE UP THRU THE BAD CHAR.
1344 006060 000000      55:     .WORD     0
1345 006062 104401 001164      TYPE     $QUES        ;; "?" "CR" & "LF"
1346 006066 000730      BR       1$          ;; TRY AGAIN
1347 006070 000000      $HI OCT: .WORD     0  ;; HIGH ORDER BITS GO HERE

```

.SBTTL TYPE ROUTINE

```

*****
*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

```

```

*CALL:
*1) USING A TRAP INSTRUCTION
*   TYPE ,MESADR      ;; MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
*OR
*   TYPE MESADR

```

```

1365
1366 006072 105737 001157      $TYPE: TSTB     $TPFLG  ;; IS THERE A TERMINAL?
1367 006076 100002      BPL      1$          ;; BR IF YES

```

```

1368 006100 000000 HALT :: HALT HERE IF NO TERMINAL
1369 006102 000407 BR :: LEAVE
1370 006104 010046 15: MOV R0, -(SP) :: SAVE R0
1371 006106 017500 000002 MOV 2(SP), R0 :: GET ADDRESS OF ASCIZ STRING
1372 006112 112046 25: MOV#B (R0)+, -(SP) :: PUSH CHARACTER TO BE TYPED ONTO STACK
1373 006114 001005 BNE 45 :: BR IF IT ISN'T THE TERMINATOR
1374 006116 005726 TST (SP)+ :: IF TERMINATOR POP IT OFF THE STACK
1375 006120 012600 605: MOV (SP)+, R0 :: RESTORE R0
1376 006122 062716 000002 35: ADD #2, (SP) :: ADJUST RETURN PC
1377 006126 000032 RTI :: RETURN
1378 006130 122716 000011 45: CMPB #HT, (SP) :: BRANCH IF <HT>
1379 006134 001430 BEQ 85 :: BRANCH IF NOT <CR>
1380 006136 122716 000200 CMPB #CRLF, (SP) :: BRANCH IF NOT <CRLF>
1381 006142 001006 BNE 55 :: POP <CR><LF> EQUIV
1382 006144 005726 TST (SP)+ :: TYPE A CR AND LF
1383 006146 104407 TYPE $CRLF
1384 006150 001165 $CRLF
1385 006152 105037 006306 CLR#B $CHARCNT :: CLEAR CHARACTER COUNT
1386 006156 000755 BR 25 :: GET NEXT CHARACTER
1387 006160 004737 006242 55: JSR PC, $TYPEC :: GO TYPE THIS CHARACTER
1388 006164 123726 001156 65: CMPB #FILLC, (SP)+ :: IS IT TIME FOR FILLER CHARS.?
1389 006170 001350 BNE 25 :: IF NO GO GET NEXT CHAR.
1390 006172 013746 001154 MOV #NULL, -(SP) :: GET # OF FILLER CHARS. NEEDED
1391 006176 105366 000001 75: DECB 1(SP) :: AND THE NULL CHAR.
1392 006202 002770 BLT 65 :: DOES A NULL NEED TO BE TYPED?
1393 006204 004737 006242 JSR PC, $TYPEC :: BR IF NO--GO POP THE NULL OFF OF STACK
1394 006210 105337 006306 DECB $CHARCNT :: GO TYPE A NULL
1395 006214 000770 BR 75 :: DO NOT COUNT AS A COUNT
1396 006216 112716 000040 85: MOV#B #' (SP) :: REPLACE TAB WITH SPACE
1397 006222 004737 006242 95: JSR PC, $TYPEC :: TYPE A SPACE
1398 006226 132737 000007 006306 BITB #7, $CHARCNT :: BRANCH IF NOT AT
1399 006234 001372 BNE 95 :: TAB STOP
1400 006236 005726 TST (SP)+ :: POP SPACE OFF STACK
1401 006240 000724 BR 25 :: GET NEXT CHARACTER
1402 006242 105777 172702 $TYPEC: TSTB 2$TPS :: WAIT UNTIL PRINTER IS READY
1403 006246 100375 BPL $TYPEC
1404 006250 116677 000002 172674 MOV#B 2(SP), 2$TPB :: LOAD CHAR TO BE TYPED INTO DATA REG.
1405 006256 122766 000015 000002 CMPB #CR, 2(SP) :: IS CHARACTER A CARRIAGE RETURN?
1406 006264 001003 BNE 15 :: BRANCH IF NO
1407 006266 105037 006306 CLR#B $CHARCNT :: YES--CLEAR CHARACTER COUNT
1408 006272 000406 BR $TYPEX :: EXIT
1409 006274 122766 000012 000002 15: CMPB #LF, 2(SP) :: IS CHARACTER A LINE FEED?
1410 006302 001402 BEQ $TYPEX :: BRANCH IF YES
1411 006304 105227 INCB (PC)+ :: COUNT THE CHARACTER
1412 006306 000000 $CHARCNT: WORD 0 :: CHARACTER COUNT STORAGE
1413 006310 000207 $TYPEX: RTS PC
1414
1415
1416
1417
1418
1419
1420 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
1421
1422 :: *****
1423 :: THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT

```

E03

NO-11-DZRP2-B, RPL11E DISK PACK FORMATTER
 DZRP2B.CMB BINARY TO OCTAL (ASCII) AND TYPE

MACY11 27(732) 01-NOV-76 16:04 PAGE 31

1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479

006312	017646	000000	
006316	116637	000001	006535
006324	112637	006537	
006330	062716	000002	
006334	000406		
006336	112737	000001	006535
006344	112737	000006	006537
006352	112737	000005	006534
006360	010346		
006362	010446		
006364	010546		
006366	113704	006537	
006372	005404		
006374	062704	000006	
006400	110437	006536	
006404	113704	006535	
006410	010005	000012	
006414	000003		
006416	006105		18:
006420	000404		
006422	006105		28:
006424	006105		
006426	006105		
006430	010503		
006432	006103		38:
006434	105337	006536	
006440	100016		
006442	042703	177770	
006446	001002		
006450	005704		
006452	001403		
006454	005204		48:
006456	052703	000060	
006462	052703	000040	58:
006466	110337	006532	

```

: #OCTAL (ASCII) NUMBER AND TYPE IT.
: #STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
: #CALL:
:     MOV     NUM,-(SP)           ;; NUMBER TO BE TYPED
:     TYPOS   N                   ;; CALL FOR TYPEOUT
:     .BYTE   N                   ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
:     .BYTE   M                   ;; M=1 OR 0
:                                     ;; 1=TYPE LEADING ZEROS
:                                     ;; 0=SUPPRESS LEADING ZEROS
: #STYPCN---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
: #STYPOS OR STYPCN
: #CALL:
:     MOV     NUM,-(SP)           ;; NUMBER TO BE TYPED
:     TYPON   N                   ;; CALL FOR TYPEOUT
: #STYPCN---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
: #CALL:
:     MOV     NUM,-(SP)           ;; NUMBER TO BE TYPED
:     TYPCN   N                   ;; CALL FOR TYPEOUT
: #STYPOS: MOV     2(SP),-(SP)     ;; PICKUP THE MODE
:         MOV     1(SP),SOFFILL   ;; LOAD ZERO FILL SWITCH
:         MOV     (SP)+,SOMODE+1  ;; NUMBER OF DIGITS TO TYPE
:         ADD     #2,(SP)         ;; ADJUST RETURN ADDRESS
:         BR     STYPCN
: #STYPCN: MOV     #1,SOFFILL     ;; SET THE ZERO FILL SWITCH
:         MOV     #6,SOMODE+1     ;; SET FOR SIX(6) DIGITS
: #STYPCN: MOV     #5,SOCNT       ;; SET THE ITERATION COUNT
:         MOV     R3,-(SP)        ;; SAVE R3
:         MOV     R4,-(SP)        ;; SAVE R4
:         MOV     R5,-(SP)        ;; SAVE R5
:         MOV     SOMODE+1,R4     ;; GET THE NUMBER OF DIGITS TO TYPE
:         NEG     R4
:         ADD     #6,R4           ;; SUBTRACT IT FOR MAX. ALLOWED
:         MOV     R4,SOMODE       ;; SAVE IT FOR USE
:         MOV     SOFFILL,R4     ;; GET THE ZERO FILL SWITCH
:         MOV     12(SP),R5      ;; PICKUP THE INPUT NUMBER
:         CLR     R3             ;; CLEAR THE OUTPUT WORD
:         ROL     R5             ;; ROTATE MSB INTO "C"
:         BR     38             ;; GO DO MSB
:         ROL     R5             ;; FORM THIS DIGIT
:         ROL     R5
:         ROL     R5
:         ROL     R5
:         MOV     R5,R3
:         ROL     R3
:         DECB   SOMODE         ;; GET LSB OF THIS DIGIT
:         BPL   78             ;; TYPE THIS DIGIT?
:         BR     IF NO
:         BIC   #177770,R3     ;; GET RID OF JUNK
:         BNE   48             ;; TEST FOR 0
:         TST   R4             ;; SUPPRESS THIS 0?
:         BEQ   58             ;; IF YES
:         INC   R4             ;; DON'T SUPPRESS ANYMORE 0'S
:         BIS   #'0,R3         ;; MAKE THIS DIGIT ASCII
:         BIS   #' ,R3         ;; MAKE ASCII IF NOT ALREADY
:         MOV   R3,#8         ;; SAVE FOR TYPING
  
```

```

1490 006472 104401 006532
1491 006476 105337 006534
1492 006502 003347
1493 006504 002402
1494 006506 005204
1495 006510 000744
1496 00 512 012605
1497 00 514 012604
1498 006516 012603
1499 00 520 016666 000002 000004
1500 00 526 012616
1501 00 530 000002
1502 00 532 000
1503 00 533 000
1504 00 534 000
1505 00 535 000
1506 006536 000000

```

```

7S: TYPE BS
      DECB $OCNT
      BGT 2S
      BLT 6S
      INC R4
      BR 2S
6S: MOV (SP)+,R5
      MOV (SP)+,R4
      MOV (SP)+,R3
      MOV 2(SP),4(SP)
      MOV (SP)+,(SP)
      RTI
BS: .BYTE 0
     .BYTE 0
$OCNT: .BYTE 0
$OFILL: .BYTE 0
$OMODE: .WORD 0

```

```

:: GO TYPE THIS DIGIT
:: COUNT BY 1
:: BR IF MORE TO DO
:: BR IF DONE
:: INSURE LAST DIGIT ISN'T A BLANK
:: GO DO THE LAST DIGIT
:: RESTORE R5
:: RESTORE R4
:: RESTORE R3
:: SET THE STACK FOR RETURNING
:: RETURN
:: STORAGE FOR ASCII DIGIT
:: TERMINATOR FOR TYPE ROUTINE
:: OCTAL DIGIT COUNTER
:: ZERO FILL SWITCH
:: NUMBER OF DIGITS TO TYPE

```

.SBTTL SAVE AND RESTORE RO-R5 ROUTINES

```

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

```

```

1515 006540
1516 00 540 010046
1517 00 542 010146
1518 006544 010246
1519 00 546 010346
1520 00 550 010446
1521 00 552 010546
1522 006554 016646 000022
1523 00 560 016646 000022
1524 00 564 016646 000022
1525 00 570 016646 000022
1526 006574 000002

```

```

$SAVREG:
MOV R0,-(SP)
MOV R1,-(SP)
MOV R2,-(SP)
MOV R3,-(SP)
MOV R4,-(SP)
MOV R5,-(SP)
MOV 22(SP),-(SP)
MOV 22(SP),-(SP)
MOV 22(SP),-(SP)
MOV 22(SP),-(SP)
RTI

```

```

:: PUSH R0 ON STACK
:: PUSH R1 ON STACK
:: PUSH R2 ON STACK
:: PUSH R3 ON STACK
:: PUSH R4 ON STACK
:: PUSH R5 ON STACK
:: SAVE PS OF MAIN FLOW
:: SAVE PC OF MAIN FLOW
:: SAVE PS OF CALL
:: SAVE PC OF CALL

```

*RESTORE RO-R5

```

*CALL:
* RESREG

```

```

$RESREG:
MOV (SP)+,22(SP)
MOV (SP)+,22(SP)
MOV (SP)+,22(SP)
MOV (SP)+,22(SP)

```

```

:: RESTORE PC OF CALL
:: RESTORE PS OF CALL
:: RESTORE PC OF MAIN FLOW
:: RESTORE PS OF MAIN FLOW

```

```

1531 006576
1532 00 576 012666 000022
1533 00 578 012666 000022
1534 00 580 012666 000022
1535 006612 012666 000022

```

1536 006616 012605
1537 006620 012604
1538 006622 012603
1539 006624 012602
1540 006626 012601
1541 006630 012600
1542 006632 000002

MOV (SP)+,R5
MOV (SP)+,R4
MOV (SP)+,R3
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,R0
RTI

:: POP STACK INTO R5
:: POP STACK INTO R4
:: POP STACK INTO R3
:: POP STACK INTO R2
:: POP STACK INTO R1
:: POP STACK INTO R0

.SBTTL TRAP DECODER

:: *****
:: *THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
:: *AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
:: *OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
:: *GO TO THAT ROUTINE.

1551
1552 006634 010046
1553 006636 016600 000002
1554 006642 005740
1555 006644 111000
1556 006646 006300
1557 006650 016000 00670
1558 006654 000200

STRAP: MOV RO, -(SP) ; SAVE RO
MOV 2(SP), RO ; GET TRAP ADDRESS
TST -(RO) ; BACKUP BY 2
MOVB (RO), RO ; GET RIGHT BYTE OF TRAP
ASL RO ; POSITION FOR INDEXING
MOV STRPAD(RO), RO ; INDEX TO TABLE
RTS RO ; GO TO ROUTINE

:: THIS IS USE TO HANDLE THE "GETPRI" MACRO

1561
1562
1563 006656 011646
1564 006660 016666 000004 000002
1565 006666 000002

STRAP2: MOV (SP), -(SP) ; MOVE THE PC DOWN
MOV 4(SP), 2(SP) ; MOVE THE PSW DOWN
RTI ; RESTORE THE PSW

.SBTTL TRAP TABLE

:: THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
:: BY THE "TRAP" INSTRUCTION.

1570
1571
1572
1573
1574 006670 006656
1575 006672 006072
1576 006674 006336
1577 006676 006312
1578 006700 006352
1579
1580 006702 005130
1581
1582 006704 005040
1583 006706 005342
1584 006710 005432
1585 006712 005732
1586 006714 006540
1587 006716 006576

ROUTINE

STRPAD: WORD STRAP2
\$TYPE ; CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
\$TYPOC ; CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
\$TYPOS ; CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
\$TYPON ; CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
\$GTSWR ; CALL=GTSWR TRAP+5(104405) GET SOFT-SWR SETTING
\$CKSWR ; CALL=CKSWR TRAP+6(104406) TEST FOR CHANGE IN SOFT-SWR
\$RDCHR ; CALL=RDCHR TRAP+7(104407) TTY TYPEIN CHARACTER ROUTINE
\$RDLIN ; CALL=RDLIN TRAP+10(104410) TTY TYPEIN STRING ROUTINE
\$RDOCT ; CALL=RDOCT TRAP+11(104411) READ AN OCTAL NUMBER FROM TTY
\$SAVREG ; CALL=SAVREG TRAP+12(104412) SAVE RO-RS ROUTINE
\$RESREG ; CALL=RESREG TRAP+13(104413) RESTORE RO-RS ROUTINE

:: THIS ROUTINE IS USED TO ENSURE THAT THE BUS ADDRESS
:: OF THE RPI1 IS SETUP TO READ THE PROPER VALUE.
:: IT WILL ALSO READ THE ADDRESS FROM THE TTY IF

1588
1589
1590
1591

```

1592
1593
1594
1595 006720 005737 001176
1596 006724 001474
1597 006726 005037 001176
1598 006732 104401 012274
1599 006736 013746 001264
1600
1601 006742 104403
1602 006744 006
1603 006745 001
1604 006746 104401 006754
1605 006752 000401
1606
1607 006756
1608 006756 104411
1609 006760 012600
1610 006762 001402
1611 006764 010037 001264
1612 006770 104401 012303
1613 006774 013746 001266
1614
1615 007000 104403
1616 007002 006
1617 007003 001
1618 007004 104401 007012
1619 007010 000401
1620
1621 007014
1622 007014 104411
1623 007016 012600
1624 007020 001402
1625 007022 010037 001266
1626 007026 104401 012312
1627 007032 013700 001272
1628 007036 006300
1629 007040 006300
1630 007042 006300
1631 007044 000300
1632 007046 010046
1633
1634 007050 104403
1635 007052 001
1636 007053 001
1637 007054 104401 007062
1638 007060 000401
1639
1640 007064
1641 007064 104411
1642 007066 012600
1643 007070 001407
1644 007072 006300
1645 007074 006300
1646 007076 006300
1647 007100 006300

```

```

;REQUIRED.
;NOTE: THIS ROUTINE DOES NOT PROTECT RD-R4
GETADR: TST     BUSADR      ;INPUT FROM TTY REQUESTED?
        BEQ     5$          ;NO--BRANCH
        CLR     BUSADR      ;YES--CLEAR THE REQUEST FLAG
1$:      TYPE    MRPADR     ;'RPADR='
        MOV     RPADR, -(SP) ;SAVE RPADR FOR TYPEOUT
        TYPOS   ;RP11 ADDRESS
        .BYTE  6           ;GO TYPE--OCTAL ASCII
        .BYTE  1           ;TYPE 6 DIGIT(S)
        TYPE    ,65$      ;TYPE LEADING ZEROS
        BR      64$      ;TYPE ASCIZ STRING
        .ASCIZ  2/2      ;GET OVER THE ASCIZ
        64$:
RDOCT   MOV     (SP)+,RO    ;GET NEW RP11 ADDRESS
        BEQ     2$          ;GET THE NEW ADDRESS
        MOV     RO,RPADR    ;BR IF ZERO ENTRY
        TYPE    MRPVEC     ;STORE THE ADDRESS
        MOV     RPVEC, -(SP);'RPVEC='
        TYPOS   ;SAVE RPVEC FOR TYPEOUT
        .BYTE  6           ;RP11 VECTOR ADDRESS
        .BYTE  1           ;GO TYPE--OCTAL ASCII
        TYPE    ,67$      ;TYPE 6 DIGIT(S)
        BR      66$      ;TYPE LEADING ZEROS
        .ASCIZ  2/2      ;TYPE ASCIZ STRING
        66$:
RDOCT   MOV     (SP)+,RO    ;READ NEW RP11 VECTOR
        BEQ     3$          ;GET THE NEW VECTOR
        MOV     RO,RPVEC    ;BR IF ZERO ENTRY
        TYPE    MRPRI      ;SAVE NEW RP11 VECTOR ADDRESS
        MOV     RPPRIO,RO   ;'RPPRIO='
        ASL     RO          ;CONVERT PRIORITY FOR TYPEOUT
        ASL     RO
        SWAB    RO
        MOV     RO, -(SP)  ;ALIGN FOR TYPEOUT
        TYPOS   ;SAVE RO FOR TYPEOUT
        .BYTE  1           ;RP11 PRIORITY
        .BYTE  1           ;GO TYPE--OCTAL ASCII
        TYPE    ,69$      ;TYPE 1 DIGIT(S)
        BR      68$      ;TYPE LEADING ZEROS
        .ASCIZ  2/2      ;TYPE ASCIZ STRING
        68$:
RDOCT   MOV     (SP)+,RO    ;GET NEW PRIORITY LEVEL
        BEQ     4$          ;SAVE NEW PRIORITY
        ASL     RO          ;BR IF ZERO ENTRY
        ASL     RO          ;CONVERT TO PRIORITY VALUE
        ASL     RO          ;CONVERT TO PRIORITY VALUE
        ASL     RO          ;CONVERT TO PRIORITY VALUE
        ASL     RO          ;CONVERT TO PRIORITY VALUE

```

```

1648 007102 006300          ASL      R0          ;CONVERT TO PRIORITY VALUE
1649 007104 010037 001272    MOV      R0,RPPRIO   ;SAVE THE VALUE
1650 007110 013777 001272 172152 4S:  MOV      RPPRIO,RPVEC+2 ;LOAD NEW PRIORITY VALUE
1651 007116 013701 000004          MOV      ERRVEC,R1   ;SAVE THE ERROR VECTOR
1652 007122 012737 007142 000004          MOV      #65,ERRVEC  ;SETUP FOR TRAP
1653 007130 005777 172130          TST      RPADR       ;CHECK FOR RP11
1654 007134 010137 000004          MOV      R1,ERRVEC   ;RESTORE ERROR VECTOR
1655 007140 000207          RTS      PC          ;RETURN
1656 007142 010137 000004          MOV      R1,ERRVEC   ;RESTORE ERROR VECTOR
1657 007146 022626          CMP      (SP)+,(SP)+ ;CLEAN OFF THE STACK
1658 007150 104401 007156          TYPE    '715'       ;TYPE ASCIZ STRING
1659 007154 000424          BR       70S        ;GET OVER THE ASCIZ
1660          ;:715: .ASCIZ '<15><12><12>/RP11 DIDN'T RESPOND TO ADDRESSING/<15><12>'
1661 007226          70S:
1662 007226 000641          BR       1L         ;GO ASK FOR ADDRESS
1663
1664
1665          ;*THIS ROUTINE CLEARS THE RP11 AND DETERMINES WHICH DRIVES ARE AVAILABLE.
1666          ;*THE TABLE 'DRVSTA' IS LOADED TO REFLECT THE SYSTEM STATUS.
1667
1668 007230 104412          RPINIT: SAVREG      ;SAVE R0-R5
1669 007232 013701 001266          MOV      RPVEC,R1   ;VECTOR ADDRESS
1670 007236 005021          CLR      (R1)+      ;SET INTERRUPT ADDRESS TO ZERO
1671 007240 013711 001272          MOV      RPPRIO,(R1) ;RP11 PRIORITY
1672 007244 005037 001172          CLR      DRVSTYP    ;CLEAR DRIVE TYPE STORAGE
1673 007250 005037 001274          CLR      DRVSTA     ;SET DRIVE STATUS TO OFFLINE
1674 007254 005037 001276          CLR      DRVSTA+2   ;SET DRIVE STATUS TO OFFLINE
1675 007260 005037 001300          CLR      DRVSTA+4   ;SET DRIVE STATUS TO OFFLINE
1676 007264 005037 001302          CLR      DRVSTA+6   ;SET DRIVE STATUS TO OFFLINE
1677 007270 013704 001264          MOV      RPADR,R4   ;PUT RP11 ADDRESS INTO R4
1678 007274 012764 000001 000004          MOV      #1,RPCS(R4) ;CLEAR THE CONTROLLER
1679 007302 012764 000001 000004          MOV      #1,RPCS(R4) ;CLEAR THE CONTROLLER
1680 007310 005001          CLR      R1         ;TABLE POINTER
1681 007312 012702 000001          MOV      #1,R2      ;DRIVE COUNTER
1682 007316 110164 000005 1S:  MOVVB   R1,RPCS+1(R4) ;SELECT DRIVE
1683 007322 032764 001000 000000          BIT      #BIT09,RPDS(R4) ;SEE IF DRIVE UNSAFE
1684 007330 001404          BEQ      2S        ;BR IF NOT UNSAFE
1685 007332 112761 177777 001274          MOVVB   #-1,DRVSTA(R1) ;SET INDICATOR TO 'UNSAFE'
1686 007340 000417          BR       4S        ;CHECK ON DRIVE TYPE
1687 007342 032764 040000 000000 2S:  BIT      #BIT14,RPDS(R4) ;IS DRIVE ONLINE?
1688 007350 001421          BEQ      5S        ;BR IF NOT
1689 007352 032764 004000 000000          BIT      #BIT11,RPDS(R4) ;SEEK INCOMPLETE
1690 007360 001004          BNE     3S        ;BR IF SET
1691 007362 032764 100000 000000          BIT      #BIT15,RPDS(R4) ;IS DRIVE READY?
1692 007370 001411          BEQ      5S        ;BR IF NOT
1693 007372 112761 000001 001274 3S:  MOVVB   #1,DRVSTA(R1) ;SET DRIVE INDICATOR TO 'ONLINE'
1694 007400 032764 020000 000000 4S:  BIT      #BIT13,RPDS(R4) ;IS THE DRIVE AN RPO3?
1695 007406 001402          BEQ      5S        ;BR IF NOT
1696 007410 150237 001172          BISB    R2,DRVSTYP  ;SET RPO3 INDICATOR
1697 007414 005201          INC     R1         ;INCREMENT THE TABLE INDEX
1698 007416 106302          ASLB   R2          ;SHIFT THE DRIVE COUNTER
1699 007420 103336          BCC    1S        ;BR IF NOT FINISHED
1700 007422 104413          RESREG          ;RESTORE R0-R5
1701 007424 000207          RTS      PC        ;RETURN
1702
1703

```

;ROUTINE TO GET THE DRIVE NUMBER

K03

MD-11-DZRP2-B, RPI1E DISK PACK FORMATTER
DZRP28.CMB TRAP TABLE

MACY11 27(732) 01-NOV-76 16:04 PAGE 37

1760	007672	005726				TST	(SP)+	:RESTORE THE STACK POINTER
1761	007674	032764	002000	000000		BIT	#BIT10,RPDS(R4)	:IS SEEK UNDERWAY?
1762	007702	001002				BNE	2\$:YES
1763	007704	104034				ERROR	ER!DS!CS	:SEEK NOT UNDERWAY
1764	007706	000755				BR	HOME	:FIX IT
1765	007710	033764	001224	000000	2\$:	BIT	ATNB,RPDS(R4)	:WAIT FOR
1766	007716	001774				BEQ	2\$:ATTENTION TO SET
1767	007720	032764	004000	000000		BIT	#BIT11,RPDS(R4)	:SEEK INCOMPLETE?
1768	007726	001402				BEQ	3\$:NO
1769	007730	104034				ERROR	ER!DS!CS	:DEVICE STATUS ERROR
1770	007732	000743				BR	HOME	:LOOP
1771	007734	132764	000023	000001	3\$:	BITB	#23,RPDS+1(R4)	:ANY ERRORS?
1772	007742	001402				BEQ	4\$:NO
1773	007744	104034				ERROR	ER!CS!DS	:DEVICE STATUS ERROR
1774	007746	000735				BR	HOME	:LOOP
1775	007750	112764	000377	000000	4\$:	MOVB	#377,RPDS(R4)	:CLEAR ATTN BITS
1776	007756	000207				RTS	PC	:EXIT
1777								
1778								
1779								
1780	007760	062703	000002			CHKAD:	ADD	#2,R3
1781	007764	012337	001236				MOV	(R3)+,HEDR
1782	007770	013737	001236	001234			MOV	HEDR,CYLR
1783	007776	006037	001236				ROR	HEDR
1784	010002	006037	001234				ROR	CYLR
1785	010006	006037	001234				ROR	CYLR
1786	010012	006037	001234				ROR	CYLR
1787	010016	006037	001234				ROR	CYLR
1788	010022	006037	001234				ROR	CYLR
1789	010026	006037	001234				ROR	CYLR
1790	010032	012337	001240				MOV	(R3)+,SECR
1791	010036	042737	177760	001240			BIC	#1C17,SECR
1792	010044	042737	177740	001236			BIC	#1C37,HEDR
1793	010052	042737	177000	001234			BIC	#1C777,CYLR
1794	010060	023737	001226	001234			CMP	CYLR,CYLR
1795	010066	001010					BNE	1\$
1796	010070	023737	001230	001236			CMP	HED,HEDR
1797	010076	001004					BNE	1\$
1798	010100	023737	001232	001240			CMP	SEC,SECR
1799	010106	001402					BEQ	2\$
1800	010110	104100			1\$:		ERROR	NO
1801	010112	000402					BR	3\$
1802	010114	062716	000002		2\$:		ADD	#2,(SP)
1803	010120	000207			3\$:		RTS	PC
1804								
1805								
1806								
1807	010122	012737	000005	001250		TTIO:	MOV	#5,TTG
1808	010130	005037	001214				CLR	WORK1
1809	010134	004737	010242		1\$:		JSR	PC,RIN
1810	010140	020027	000015				CMP	RO,#15
1811	010144	001427					BEQ	2\$
1812	010146	004737	011154				JSR	PC,TT0
1813	010152	020027	000067				CMP	RO,#67
1814	010156	101030					BHI	3\$
1815	010160	020027	000057				CMP	RO,#57

;CHECK THE HEADER

:SKIP 1ST WORD
:FETCH HEAD AND
:CYLINDER READ
:JUSTIFY HEAD
:FETCH SECTOR READ

:IS CYL# OK?
:NO
:IS TRACK# OK
:NO
:IS SECTOR # OK?
:YES
:HEADER COMPARE ERROR
:EXIT
:INCREMENT RETURN
:RETURN

;READ AND CHECK OCTAL DATA FROM THE KEYBOARD

:INPUT UP TO 5 OCTAL DIGITS
:CLEAR WORK REGISTER
:READ TTY CHARACTER INTO RO
:IS IT CR?
:YES
:ECHO
:TOO BIG?
:YES
:TOO SMALL?

1816	010164	101425				BLOS	3\$:YES
1817	010166	005337	001250			DEC	TTC		:COUNT DIGIT
1818	010172	100422				BMI	3\$:TOO MANY?
1819	010174	042700	177770			BIC	#1C7,RO		:NO - MASK IT
1820	010200	000241				CLC			:AND
1821	010202	006137	001214			ROL	WORK1		:SHIFT
1822	010206	006137	001214			ROL	WORK1		:WORK REGISTER
1823	010212	006137	001214			ROL	WORK1		:3 PLACES
1824	010216	060037	001214			ADD	RO,WORK1		:ADD NEW NUMBER
1825	010222	000744				BR	1\$:GET NEXT ONE
1826	010224	104401	001165		2\$:	TYPE	,SCRLF		:CR-LF
1827	010230	062716	000002			ADD	#2,(SP)		:SKIP RETURN
1828	010234	013700	001214			MOV	WORK1,RO		
1829	010240	000207			3\$:	RTS	PC		
1830									
1831									
1832									
1833									
1834	010242	104407							
1835	010244	012600				RIN:	RDCHR		:READ THE KEYBOARD
1836	010246	000207				MOV	(SP)+,RO		:PUT THE CHARACTER IN RO
1837						RTS	PC		:RETURN
1838									
1839									
1840									
1841	010250	017637	000000	001256		RTL:	MOV	#2(SP),ROTOG	:GET THE SHIFT COUNT
1842	010256	062716	000002			ADD	#2,(SP)		:CORRECT RETURN ADDRESS
1843	010262	005337	001256		1\$:	DEC	ROTOG		:DECREMENT THE COUNT
1844	010266	100402				BMI	2\$:EXIT WHEN DONE
1845	010270	006300				ASL	RO		:SHIFT THE BIT
1846	010272	000773				BR	1\$:CONTINUE
1847	010274	000207			2\$:	RTS	PC		:EXIT
1848									
1849									
1850									
1851	010276	032777	002000	170634		\$ERROR:	BIT	#SW10,#SWR	:RING THE BELL?
1852	010304	001402				BEQ	1\$:BRANCH IF NO
1853	010306	104401	001160			TYPE	,\$BELL		
1854	010312	032777	020000	170620	1\$:	BIT	#SW13,#SWR		:TYPE ERROR MESSAGE ?
1855	010320	001064				BNE	2\$:EXIT IF SWITCH 13 IS SET
1856	010322	104412				SAVREG			:SAVE RO-RS
1857	010324	104401	012052			TYPE	ERMS1		
1858	010330	011605				MOV	(SP),RS		:FETCH SAVED PC
1859	010332	162705	000002			SUB	#2,RS		:FIND ADDRESS OF ERROR CALL
1860	010336	010546				MOV	RS,-(SP)		:PUT ADDRESS ON THE STACK
1861	010340	104402				TYPOC			:TYPE THE CALLING PC
1862	010342	104401	012231			TYPE	STMS		:TYPE 'PS='
1863	010346	016600	000002			MOV	#2(SP),RO		:STATUS AT TIME OF ERROR CALL
1864	010352	010046				MOV	RO,-(SP)		:PUT IT ON THE STACK
1865	010354	104402				TYPOC			:TYPE IT
1866	010356	032715	000020			BIT	#BIT04,(RS)		:RPCS?
1867	010362	001402				BEQ	+6		:NO
1868	010364	004737	010542			JSR	PC,CSTYPE		:YES
1869	010370	032715	000010			BIT	#BIT03,(RS)		:RPER?
1870	010374	001402				BEQ	+6		:NO
1871	010376	004737	010572			JSR	PC,ERTYPE		:YES

1872	010402	032715	000004	BIT	#BIT02, (R5)	:RPDS?
1873	010406	001402		BEQ	.+6	:NO
1874	010410	004737	010606	JSR	PC, DSTYPE	:YES
1875	010414	032715	000040	BIT	#BIT05, (R5)	:RPOA & RPCA?
1876	010420	001402		BEQ	.+6	:NO
1877	010422	004737	010622	JSR	PC, ADTYPE	:YES
1878	010426	032715	000100	BIT	#BIT06, (R5)	:HEADER MESSAGE?
1879	010432	001402		BEQ	.+6	:NO
1890	010434	004737	011026	JSR	PC, HCTYPE	:YES
1891	010440	032715	000001	BIT	#BIT00, (R5)	:BIT 0 FOR GOOD/BAD
1892	010444	001402		BEQ	.+6	:SKIP IF=0
1893	010446	004737	010514	JSR	PC, GBTYPE	:GOOD/BAD
1894	010452	032715	000002	BIT	#BIT01, (R5)	:BIT 1 FOR DATA
1895	010456	001402		BEQ	.+6	:SKIP IF=0
1896	010460	004737	010556	JSR	PC, DATYPE	:DATA
1897	010464	104413		RESREG		:RESTORE R0-R5
1898	010466	104401	001165	TYPE	\$CRLF	:CR-LF
1899	010472	105764	000004	25: TSTB	RPCS(R4)	:WAIT FOR DONE
1890	010476	100005		BPL	3\$	
1891	010500	032777	100000 170432	BIT	#SW15, @SWR	:HALT ON ERROR?
1892	010506	001401		BEQ	3\$	
1893	010510	000000		HALT		:ERROR HALT. CONTINUE
1894	010512	000002		3\$: RTI		:EXIT ERROR ROUTINE
1895						
1896						
1897						:TYPE GOOD/BAD DATA
1898	010514	104401	012212	GBTYPE: TYPE	GDMS	: 'GOOD'
1899	010520	013746	001200	MOV	GOOD, -(SP)	:PUT IT ON THE STACK
1900	010524	104402		TYPOC		:TYPE IT
1901	010526	104401	012223	TYPE	BDMS	:TYPE 'BAD'
1902	010532	013746	001202	MOV	BAD, -(SP)	:PUT IT ON THE STACK
1903	010536	104402		TYPOC		:TYPE IT
1904	010540	000207		RTS	PC	:EXIT
1905						
1906						:TYPE THE CONTENTS OF 'RPCS'
1907						
1908	010542	104401	012127	CSTYPE: TYPE	CSMS	: 'STATUS'
1909	010546	016446	000004	MOV	RPCS(R4), -(SP)	:PUT RPCS ON THE STACK
1910	010552	104402		TYPOC		:TYPE IT
1911	010554	000207		RTS	PC	:EXIT
1912						
1913						:TYPE THE DATA
1914						
1915	010556	104401	012240	DATYPE: TYPE	DAMS	: 'DATA='
1916	010562	013746	001204	MOV	DATA, -(SP)	:PUT DATA ON THE STACK
1917	010566	104402		TYPOC		:TYPE IT
1918	010570	000207		RTS	PC	:EXIT
1919						
1920						:TYPE THE ERROR REGISTER (RPER)
1921						
1922	010572	104401	012140	ERTYPE: TYPE	ERMS	: 'RPER='
1923	010576	016446	000002	MOV	RPER(R4), -(SP)	:PUT RPER ON THE STACK
1924	010602	104402		TYPOC		:TYPE IT
1925	010604	000207		RTS	PC	:EXIT
1926						
1927						:TYPE THE DRIVE STATUS REGISTR (RPDS)

1928									
1929	010606	104401	012151			DSTYPE: TYPE	DSMS	'RPDS:'	
1930	010612	016446	000000			MOV	RPDS(R4),-(SP)	PUT RPDS ON THE STACK	
1931	010616	104402				TYPOC		TYPE IT	
1932	010620	000207				RTS	PC	EXIT	
1933									
1934									
1935									
1936	010622	032764	000001	000002		ADTYPE: BIT	#BIT00,RPER(R4)	ADDRESS ERROR?	
1937	010630	001407				BEQ	1\$	BRANCH IF NO	
1938	010632	016437	000012	001220		MOV	RPCA(R4),WORK3	GET THE CLYINDER ADDRESS	
1939	010640	016437	000014	001216		MOV	RPDA(R4),WORK2	GET HEAD AND SECTOR ADDRESS	
1940	010646	000437				BR	ADT1		
1941	010650	016437	000012	001220	1\$:	MOV	RPCA(R4),WORK3		
1942	010656	016437	000014	001216		MOV	RPDA(R4),WORK2		
1943	010664	042737	160360	001216		BIC	#160360,WORK2		
1944	010672	032737	000017	001216		BIT	#17,WORK2		
1945	010700	001403				BEQ	DECTK		
1946	010702	005337	001216			DEC	WORK2		
1947	010706	000417				BR	ADT1		
1948	010710	132737	000037	001217	DECTK:	BITB	#37,WORK2+1		
1949	010716	001406				BEQ	DECCY		
1950	010720	105337	001217			DECB	WORK2+1		
1951	010724	052737	000011	001216		BIS	#11,WORK2		
1952	010732	000405				BR	ADT1		
1953	010734	012737	011411	001216	DECCY:	MOV	#11411,WORK2		
1954	010742	005337	001220			DEC	WORK3		
1955	010746	104401	012162		ADT1:	TYPE	CYMS	'CAR='	
1956	010752	013746	001220			MOV	WORK3, -(SP)	MPUT CAR ON THE STACK	
1957	010756	042716	177400			BIC	#1C377,(SP)		
1958	010762	104402				TYPOC		TYPE IT	
1959	010764	104401	012172			TYPE	TAMS	'TAR='	
1960	010770	005046				CLR	-(SP)	CLEAR THE STACK	
1961	010772	113716	001217			MOVW	WORK2+1,(SP)	PUT THE TRACK ON THE STACK	
1962	010776	042716	177740			BIC	#1C37,(SP)		
1963	011002	104402				TYPOC		TYPE IT	
1964	011004	104401	012202			TYPE	SEMS	'SAR='	
1965	011010	005046				CLR	-(SP)	CLEAR THE STACK	
1966	011012	113716	001216			MOVW	WORK2,(SP)	PUT THE SECTOR ADDRESS ON THE STACK	
1967	011016	042716	177760			BIC	#1C17,(SP)		
1968	011022	104402				TYPOC		TYPE IT	
1969	011024	000207				RTS	PC	EXIT	
1970									
1971									
1972									
1973	011026	104401	012010			HDTYPE: TYPE	HDMS		
1974	011032	104401	012212			TYPE	GDMS		
1975	011036	104401	012047			TYPE	S2		
1976	011042	013746	001226			MOV	CYL, -(SP)	SAVE CYL FOR TYPEOUT	
1977	011046	104402				TYPOC		GO TYPE--OCTAL ASCII(ALL DIGITS)	
1978	011050	104401	012050			TYPE	S1		
1979	011054	013746	001230			MOV	HED, -(SP)	SAVE HED FOR TYPEOUT	
1980	011060	104402				TYPOC		GO TYPE--OCTAL ASCII(ALL DIGITS)	
1981	011062	104401	012050			TYPE	S1		
1982	011066	013746	001232			MOV	SEC, -(SP)	SAVE SEC FOR TYPEOUT	
1983	011072	104402				TYPOC		GO TYPE--OCTAL ASCII(ALL DIGITS)	

1984	011074	104401	001165	TYPE	,SCLF	
1985	011100	104401	012050	TYPE	,S1	
1986	011104	104401	012223	TYPE	,BOMS	
1987	011110	104401	012047	TYPE	,S2	
1988	011114	013746	001234	MOV	CYLR,-(SP)	::SAVE CYLR FOR TYPEOUT ::GO TYPE--OCTAL ASCII(ALL DIGITS)
1989	011120	104402		TYPOC		
1990	011122	104401	012050	TYPE	,S1	
1991	011126	013746	001236	MOV	HEDR,-(SP)	::SAVE HEDR FOR TYPEOUT ::GO TYPE--OCTAL ASCII(ALL DIGITS)
1992	011132	104402		TYPOC		
1993	011134	104401	012050	TYPE	,S1	
1994	011140	013746	001240	MOV	SECR,-(SP)	::SAVE SECR FOR TYPEOUT ::GO TYPE--OCTAL ASCII(ALL DIGITS)
1995	011144	104402		TYPOC		
1996	011146	104401	001165	TYPE	,SCLF	
1997	011152	000207		RTS	PC	
1999						
2000						;TYPE THE CHARACTER IN R0
2001	011154	022700	000015	TTO:	CMP #15,R0	;PRINT A 'CR' ?
2002	011160	001405			BEO 1\$;BR IF 'CR'
2003	011162	110037	011202		MOVB R0,3\$;SETUP 'ASCIZ' FIELD
2004	011166	104401	011202		TYPE 3\$;TYPE IT
2005	011172	000402			BR 2\$;EXIT
2006	011174	104401	001165	1\$:	TYPE ,SCLF	;TYPE A CR-LF
2007	011200	000207		2\$:	RTS PC	;RETURN
2008	011202	000	000	3\$:	.BYTE 0,0	;CHARACTER STORAGE AND TERMINATOR
2009						
2010						;TYPE THE MESSAGE WHOSE ADDRESS IS IN R0
2011						
2012	011204	010037	011212	TYPOUT:	MOV R0,1\$;STORE THE ADDRESS
2013	011210	104401			TYPE	;TYPE THE MESSAGE
2014	011212	000000		1\$:	.WORD 0	;MESSAGE ADDRESS
2015	011214	000207			RTS PC	;RETURN
2016						
2017						;CLEAR THE CONTROLLER
2018						
2019	011216	012764	000001 000004	CLRP:	MOV #1,RPCS(R4)	;CLEAR THE CONTROLLER
2020	011224	012764	000001 000004		MOV #1,RPCS(R4)	;CLEAR THE CONTROLLER AGAIN
2021	011232	000207			RTS PC	;RETURN
2022						
2023						.SBTTL TELETYPE MESSAGES
2024						
2025	011234	005015	042012 044522	UMES:	.ASCIZ <15><12><12>/DRIVE #: /	
2026	011242	042526	021440 020072			
2027	011250	000				
2028	011251	015	005012 047111	SECMES:	.ASCIZ <15><12><12>/INPUT THE SECTOR NUMBERS (0-9) IN THE ORDER DESIRED./	
2029	011256	052520	020124 044124			
2030	011264	020105	042523 052103			
2031	011272	051117	047040 046525			
2032	011300	042502	051522 024040			
2033	011306	026460	024471 044440			
2034	011314	020116	044124 020105			
2035	011322	051117	042504 020122			
2036	011330	042504	044523 042522			
2037	011336	027104	000			
2038	011341	015	030012 020072	NO:	.ASCIZ <15><12>/0: /	
2039	011346	000				

0000	011347	015	030412	020072	N1:	.ASCIZ	<15><12>/1: /
0001	011354	000					
0002	011355	015	031012	020072	N2:	.ASCIZ	<15><12>/2: /
0003	011362	000					
0004	011363	015	031412	020072	N3:	.ASCIZ	<15><12>/3: /
0005	011370	000					
0006	011371	015	032012	020072	N4:	.ASCIZ	<15><12>/4: /
0007	011376	000					
0008	011377	015	032412	020072	N5:	.ASCIZ	<15><12>/5: /
0009	011404	000					
0010	011405	015	033012	020072	N6:	.ASCIZ	<15><12>/6: /
0011	011412	000					
0012	011413	015	033412	020072	N7:	.ASCIZ	<15><12>/7: /
0013	011420	000					
0014	011421	015	034012	020072	N8:	.ASCIZ	<15><12>/8: /
0015	011426	000					
0016	011427	015	034412	020072	N9:	.ASCIZ	<15><12>/9: /
0017	011434	000					
0018	011435	015	005012	042523	SWMES:	.ASCII	<15><12><12>/SET THE FORMAT ENABLE SWITCH./
0019	011442	020124	044124	020105			
0020	011450	047506	046522	052101			
0021	011456	042440	040516	046102			
0022	011464	020105	053523	052111			
0023	011472	044103	056				
0024	011475	015	051412	052105		.ASCII	<15><12>/SET THE RP11 WRITE ENABLE SWITCH./
0025	011502	052040	042510	051040			
0026	011510	030520	020061	051127			
0027	01151E	052111	020105	047105			
0028	011524	041101	042514	051440			
0029	011530	044527	041524	027110			
0030	011540	005015	042523	020124		.ASCII	<15><12>/SET THE SELECTED UNIT WRITE ENABLE SWITCH./
0031	011546	044124	020105	042523			
0032	011554	042514	052103	043105			
0033	011562	052440	044516	020124			
0034	011570	051127	052111	020105			
0035	011576	047105	041101	042514			
0036	011604	051440	044527	041524			
0037	011612	027110					
0038	011614	005015	052123	044522		.ASCIZ	<15><12>/STRIKE ANY TELETYPE KEY WHEN READY./<15><12>
0039	011622	042513	040440	054516			
0040	011630	052040	046105	052105			
0041	011636	050131	020105	042513			
0042	011644	020131	044127	047105			
0043	011652	051040	040505	054504			
0044	011660	006456	000012				
0045	011664	005015	051012	051505	MSWMS:	.ASCII	<15><12><12>/RESET THE FORMAT ENABLE SWITCH TO NORMAL./
0046	011672	052105	052040	042510			
0047	011700	043040	051117	040515			
0048	011706	020124	047105	041101			
0049	011714	042514	051440	044527			
0050	011722	041524	020110	047524			
0051	011730	047040	051117	040515			
0052	011736	027114					

2093 011740 005015 052123 044522
 2094 011746 042513 040440 054516
 2095 011754 052040 046105 052105
 2096 011762 050131 020105 042513
 2097 011770 020131 044127 047105
 2098 011776 051040 040505 054504
 2099 012004 006456 000012
 2100 012010 005015 020040 020040 HDMS: .ASCIZ <15><12>/
 2101 012016 020040 054503 044514 CYLINDER TRACK SECTOR/
 2102 012024 042116 051105 020040
 2103 012032 051124 041501 020113
 2104 012040 042523 052103 051117
 2105 012046 000
 2106 012047 040
 2107 012050 000040 S2: .ASCII //
 2108 012052 005015 042412 051122 S1: .ASCIZ //
 2109 012060 051117 040440 020124 ERMS1: .ASCIZ <15><12><12>/ERROR AT ADDRESS /
 2110 012066 042101 051104 051505
 2111 012074 020123 000
 2112 012077 015 047412 042114 OLDMS: .ASCIZ <15><12>/OLD: /
 2113 012104 020072 020040 020040
 2114 012112 000
 2115 012113 015 047012 053505 NEWS: .ASCIZ <15><12>/NEW: /
 2116 012120 020072 020040 020040
 2117 012126 000
 2118 012127 015 051012 041520 CSMS: .ASCIZ <15><12>/RPCS= /
 2119 012134 036523 000040
 2120 012140 005015 050122 051105 ERMS: .ASCIZ <15><12>/RPER= /
 2121 012146 020075 000
 2122 012151 015 051012 042120 DSMS: .ASCIZ <15><12>/RPDS= /
 2123 012156 036523 000040
 2124 012162 005015 040503 036522 CYMS: .ASCIZ <15><12>/CAR= /
 2125 012170 000040
 2126 012172 020040 040524 036522 TAMS: .ASCIZ / TAR= /
 2127 012200 000040
 2128 012202 020040 040523 036522 SEMS: .ASCIZ / SAR= /
 2129 012210 000040
 2130 012212 005015 047507 042117 GOMS: .ASCIZ <15><12>/G000= /
 2131 012220 020075 000
 2132 012223 102 042101 020075 BOMS: .ASCIZ /BA0= /
 2133 012230 000
 2134 012231 040 050040 036523 STMS: .ASCIZ / PS= /
 2135 012236 000040
 2136 012240 005015 040504 040524 DAMS: .ASCIZ <15><12>/DATA= /
 2137 012246 020075 000
 2138 012251 015 005012 042524 ENDM1: .ASCIZ <15><12><12>/TEST COMPLETE/<15><12>
 2139 012256 052123 041440 046517
 2140 012264 046120 052105 006505
 2141 012272 000012
 2142 012274 050122 042101 036522 MRPADR: .ASCIZ /RPADR=/
 2143 012302 000
 2144 012303 122 053120 041505 MRPVEC: .ASCIZ /RPVEC=/
 2145 012310 000075
 2146 012312 050122 051120 047511 MRPPRI: .ASCIZ /RPPRI0=/
 2147 012320 000075
 2148 012322 020040 020040 047117 ONLINE: .ASCIZ / ONLINE /

```

2149 012330 044514 042516 020040
2150 012336 020040 000
2151 012341 040 020040 052440 UNSAFE: .ASCIZ / UNSAFE /
2152 012346 051516 043101 020105
2153 012354 020040 000040
2154 012360 020040 020040 043117 OFFLIN: .ASCIZ / OFFLINE /
2155 012366 046106 047111 020105
2156 012374 020040 000
2157 012377 122 030120 000062 RP02: .ASCIZ /RP02/
2158 012404 050122 031460 000 RP03: .ASCIZ /RP03/
2159
2160 012412 .EVEN
2161
2162 ;;*****
2163
2164 ;BUFFERS START HERE
2165
2166 012412 INBUF=.
2167 014712 OUTBUF=INBUF+2300
2168
2169 000001 .END
2170

```


MRPVEC	012303	1612	2144#																	
NEWS	012113	795	2115#																	
NSTR	004504	879	1018#																	
NSUMES	011664	723	2085#																	
NS2	002266	684#	698	702																
NS	011341	1018	2038#																	
NI	011347	1019	2040#																	
NS	011355	1020	2042#																	
NS	011363	1021	2044#																	
NS	011371	1022	2046#																	
NS	011377	1023	2048#																	
NS	011405	1024	2050#																	
NS	011413	1025	2052#																	
NS	011421	1026	2054#																	
NS	011427	1027	2056#																	
OFFLIN	012360	641	2154#																	
OLDMS	012077	781	2112#																	
ONLINE	012322	632	2148#																	
OPTION	001174	478#	554#	556#	558#	560#	563#	652	655	658	661	762								
OUTBUF	014712	895	906	986	2168#															
PARIN	007570	783	787	797	801	1738#														
PC	=X000007	326#	600#	610#	611#	680#	681#	683#	684#	720#	721#	724#	725#	738#						
		743#	778#	779#	783#	787#	790#	797#	801#	804#	822#	835#	840#	853#						
		856#	874#	875#	877#	882#	883#	887#	888#	892#	903#	916#	920#	931#						
		938#	965#	969#	970#	981#	1000#	1012#	1014#	1081#	1148#	1185#	1387#	1394#						
		1401#	1415#	1417#	1655#	1701#	1706#	1715#	1725#	1734#	1738#	1740#	1748#	1750#						
		1754#	1776#	1803#	1809#	1812#	1829#	1836#	1847#	1868#	1871#	1874#	1877#	1880#						
		1883#	1886#	1904#	1911#	1918#	1925#	1932#	1969#	1997#	2007#	2015#	2021#							
		312#																		
PIRQ	= 177772	406#																		
PIRQVE	= 000240	329#																		
PRO	= 000000	330#																		
PR1	= 000040	331#																		
PR2	= 000100	332#																		
PR3	= 000140	333#																		
PR4	= 000200	334#																		
PR5	= 000240	335#																		
PR6	= 000300	336#																		
PR7	= 000340	722	915																	
PS	= 177776	309#	310	679#	719#	777#	873#													
PSM	= 177776	310#	690#	722#	915#															
P-RVEC	= 000024	401#																		
P2L1	001466	725#	765																	
P2L2	002476	727#	739	758																
P2L3	002604	743#	748	753																
P2N1	001570	736	740#																	
ROCHR	= 104407	1240	1583#	1834																
ROLIN	= 104410	1314	1594#																	
ROOCT	= 104411	1585#	1608	1622	1641															
RESREG	= 104413	157#	1700	1887																
RESVEC	= 000010	346#																		
RIN	010242	683	724	969	1809	1834#														
ROTOG	001256	503#	1841#	1843#																
RPROR	001264	510#	651	1599	1611#	1653	1677													
RPBA	= 000010	422#	730#	824#	842#	895#	923#													
RPCA	= 000012	423#	727#	827#	843#	890#	918#	1938	1941											
RPCS	= 000004	420#	687#	688#	689#	699	704	731#	732#	733#	734	825#	826#	830#						

USER SYMBOLS

STPFLG	001157	468#	136E	1419															
STPS	001150	463#	1406	1419															
STRAP	006634	575#	1552#																
STRAP2	006656	1563#	1574																
STRP =	000C14	1567#	1576#	1577#	1578#	1579#	1580	1581#	1582	1583#	1584#	1585#	1586#	1587#					
		1588#																	
STRPAD	006670	1557	1574#																
STSTNM	001102	441#																	
STTYIN	005666	1237	1238	1250	1268	1282	1286#												
STYPBN=	*****	1579																	
STYPOS=	*****	1579																	
STYPE	006072	1366#	1567	1575															
STYPEC	006242	1185	1387	1394	1401	1406#	1407												
STYPEX	006310	1412	1414	1417#															
STYPOC	006336	1450#	1576																
STYPON	006352	1449	1452#	1578															
STYPOS	006312	1445#	1577																
SOFILL	006535	1446#	1450*	1460	1495#														
.	= 012412	271#	275#	281#	438#	473	570	598#	620#	1058	1062#	1063	1064#	1286#					
		1287	1293#	1348	1419	1661#	1739	1867	1870	1873	1876	1879	1882	1885					
		2161#	2167																

U
U

.SPOWE	18		
.SRAND	18		
.SRODE	18		
.SROOC	18	2478	1295
.SREAO	18	2478	1055
.SR2AZ	18		
.SSAVE	18	2478	1498
.SS32D	18		
.SSB20	18		
.SSCOP	18		
.SSIZE	18		
.SSUPR	18		
.STRAP	18	2478	1544
.STYPB	18		
.STYPD	18		
.STYPE	18	2478	1349
.STYPO	18	2478	1420
.S4OCA	18		
.1170	18		

ADD	907	908	980	997	1013	1170	1179	1332	1376	1448	1458	1749	1780	1802	1824
	1827	1842													
ASL	1193	1194	1195	1325	1327	1329	1556	1628	1629	1630	1644	1645	1646	1647	1648
	1845														
ASLB	649	1698													
BCC	650	1699													
BEQ	613	631	653	656	659	662	700	735	761	763	832	850	860	900	928
	953	1007	1010	1093	1146	1177	1192	1220	1256	1320	1379	1414	1475	1596	1610
	1624	1643	1664	1688	1692	1695	1713	1720	1723	1766	1768	1772	1799	1811	1852
	1867	1870	1873	1876	1879	1882	1885	1892	1937	1945	1949	2002			
BGT	1117	1189	1322	1482											
BHI	747	752	757	793	807	913	973	1709	1744	1747	1814				
BIC	821	1089	1108	1118	1136	1163	1190	1331	1472	1791	1792	1793	1819	1943	1957
	1962	1967													
BIS	688	732	819	826	847	897	925	1197	1477	1478	1951				
BISB	1696														
BIT	614	635	652	655	658	661	699	704	734	760	762	831	849	859	899
	927	952	1006	1009	1683	1687	1689	1691	1694	1719	1722	1761	1765	1767	1851
	1854	1866	1869	1872	1875	1878	1881	1884	1891	1936	1944				
BITB	1462	1771	1948												
BLO	1251														
BLOS	946	950	1239	1816											
BLT	1115	1187	1324	1393	1483										
BMI	630	975	1818	1844											
BNE	570	584	594	603	605	615	636	694	696	705	886	910	941	964	977
	994	999	1091	1097	1102	1110	1122	1132	1138	1168	1175	1182	1225	1243	1245
	1261	1265	1275	1373	1381	1389	1403	1410	1473	1690	1733	1742	1759	1762	1795
	1797	1855													
BPL	736	833	851	901	929	1106	1134	1160	1367	1407	1471	1890			
BR	555	557	559	561	586	596	607	618	633	638	640	666	671	698	703
	707	739	765	767	784	788	791	798	802	805	836	854	857	863	884
	893	904	914	921	932	956	1100	1171	1198	1200	1254	1263	1269	1271	1333
	1346	1369	1386	1396	1405	1412	1449	1464	1485	1605	1619	1638	1659	1662	1686
	1707	1727	1739	1764	1770	1774	1801	1825	1846	1940	1947	1952	2005		
CLC	1820														
CLR	563	568	621	679	690	719	726	728	740	741	749	754	777	809	837
	838	839	873	889	891	917	919	935	947	999	990	1074	1103	1149	1157
	1158	1214	1215	1236	1259	1317	1318	1462	1597	1670	1672	1673	1674	1675	1676
	1830	1838	1940	1965											
CLRB	962	1266	1276	1342	1385	1411									
CMP	569	563	604	693	746	751	756	792	806	912	945	949	972	1090	1092
	1096	1101	1109	1114	1116	1121	1131	1137	1167	1174	1186	1189	1224	1238	1250
	1657	1708	1741	1743	1746	1794	1796	1798	1810	1813	1815	2001			
CMB	1145	1181	1242	1260	1264	1274	1321	1323	1378	1380	1388	1409	1413		
COMB	978														
DEC	695	1221	1249	1732	1758	1817	1843	1946	1954						
DEC8	1392	1395	1470	1481	1950										
EMT	301														
HALT	275	670	766	862	955	1368	1893								
INC	593	612	648	745	750	755	805	909	911	940	944	948	963	993	998
	1113	1120	1196	1223	1476	1484	1697	1745							
INCB	1415														
IOT	302														
JMP	284	287	290	293	295	654	657	660	663	748	753	758	764	861	954
JSR	600	610	611	680	681	683	684	720	721	724	725	738	743	778	779
	783	787	790	797	801	804	822	835	840	853	856	874	875	877	882

NOV	883	887	888	892	903	916	920	931	938	969	970	1148	1185	1387	1394
	1401	1706	1715	1738	1748	1754	1809	1812	1868	1871	1874	1877	1880	1883	1886
	554	556	558	560	562	567	571	573	574	575	576	579	580	581	582
	587	589	590	591	622	624	632	634	637	639	641	642	651	678	685
	696	691	632	702	708	718	722	727	729	730	742	776	782	785	786
	789	794	796	799	800	803	808	810	811	823	824	627	841	842	843
	855	872	878	879	880	881	890	894	895	905	906	915	918	922	923
	933	934	936	937	942	943	960	961	979	985	986	987	988	991	992
	995	996	1075	1076	1077	1078	1080	1111	1123	1154	1178	1183	1212	1213	1216
	1226	1235	1237	1248	1279	1280	1281	1282	1309	1310	1311	1312	1313	1315	1316
	1335	1336	1337	1338	1339	1370	1371	1375	1390	1445	1453	1454	1455	1461	1468
	1486	1487	1488	1489	1490	1516	1517	1518	1519	1520	1521	1522	1523	1524	1525
	1532	1533	1534	1535	1536	1537	1538	1539	1540	1541	1552	1553	1557	1563	1564
	1599	1609	1611	1613	1623	1625	1627	1632	1642	1649	1650	1651	1652	1654	1656
	1669	1671	1677	1678	1679	1681	1710	1711	1714	1717	1721	1724	1757	1781	1792
	1790	1807	1828	1835	1841	1858	1860	1863	1864	1899	1902	1909	1916	1923	1930
NOVB	1938	1939	1941	1942	1953	1956	1976	1979	1982	1988	1991	1994	2012	2019	2020
	608	687	689	731	733	825	828	829	830	844	845	846	848	896	898
	924	926	1004	1035	1033	1033	1107	1119	1135	1150	1162	1222	1241	1246	1252
	1257	1272	1319	1372	1400	1408	1446	1447	1450	1451	1452	1456	1459	1460	1479
	1555	1682	1685	1693	1718	1755	1756	1775	1961	1966	2003				
NEG	1457														
NOP	744	939													
RESET	564														
ROL	812	813	814	815	816	817	818	1326	1328	1330	1463	1465	1466	1467	1469
ROR	1821	1822	1823												
RTI	820	1783	1784	1785	1786	1787	1788	1789							
RTS	5 3	709	1112	1124	1184	1217	1227	1283	1340	1377	1491	1526	1542	1565	1894
	965	981	1000	1012	1014	1081	1417	1558	1655	1701	1725	1734	1740	1750	1776
	1803	1829	1836	1847	1904	1911	1918	1925	1932	1969	1997	2007	2015	2021	
SUB	971	1859													
SWAP	1631														
TRAP	1567	1576	1577	1578	1580	1582	1583	1584	1585	1586	1587				
TST	602	974	1079	1099	1104	1147	1176	1191	1219	1244	1255	1278	1334	1341	1374
TSTB	1382	1404	1474	1554	1595	1653	1712	1760							
.ASCII	629	976	1105	1133	1159	1346	1406	1889							
.ASCII2	470	471	2058	2064	2070	2035	2106								
	469	472	598	620	668	1287	1288	1289	1291	1607	1621	1640	1661	2025	2028
	2038	2040	2042	2044	2046	2048	2050	2052	2054	2056	2078	2093	2100	2107	2108
	2112	2115	2118	2120	2122	2124	2126	2128	2130	2132	2134	2136	2138	2142	2144
.BLKB	2146	2148	2151	2154	2157	2158									
.BYTE	1062	1286													
	441	442	447	448	456	457	465	466	467	468	519	520	521	522	523
	524	525	526	627	628	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051
	1234	1235	1492	1493	1494	1495	1602	1603	1616	1617	1635	1636	2008		
.DSABL	1231														
.EQU	1	247	1058												
.END	2170														
.ENDC	253	266	267	278	301	393	407	435	439	441	469	470	474	528	545
	570	551	571	572	573	575	577	593	598	604	610	620	628	629	668
	1058	1087	1090	1094	1125	1127	1142	1165	1201	1205	1216	1228	1229	1237	1239
	1242	1270	1287	1293	1298	1304	1348	1352	1372	1423	1501	1547	1553	1556	1575
	1576	1577	1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1588	1603	1604
	1607	1617	1618	1621	1636	1637	1640	1661	2164						
.EQUIV	301	302	310	325	326	355	356	357	358	359	360	361	362	363	364
	383	384	385	386	387	388	389	390	391	392					

.EVEN	598	620	668	1064	1293	1607	1621	1640	1661	2161						
.IF	249	266	267	278	299	365	393	434	438	440	469	473	474	544	549	
	551	566	571	573	575	577	593	597	601	604	619	627	628	667	1057	
	1059	1087	1090	1126	1127	1141	1165	1204	1205	1215	1228	1236	1238	1242	1243	
	1296	1267	1293	1297	1300	1316	1351	1372	1422	1500	1546	1552	1556	1567	1576	
	1577	1578	1579	1580	1582	1583	1584	1585	1586	1587	1588	1602	1603	1606	1616	
.IFF	1617	1620	1635	1636	1639	1660	2163									
	266	267	299	435	438	440	469	474	545	550	571	628	1058	1127	1142	
	1152	1205	1208	1216	1228	1229	1238	1270	1286	1298	1352	1423	1501	1547	1553	
	1603	1604	1617	1618	1636	1637	2164									
.IFT	598	620	668	1207	1212	1321	1341	1348	1607	1621	1640	1661				
.IFTF	598	620	668	1152	1205	1208	1317	1325	1347	1607	1621	1640	1661			
.IIF	248	253	258	263	264	265	266	267	268	275	473	572	577	625	1058	
	1064	1095	1096	1155	1278	1287	1293	1348	1419	1575	1576	1577	1578	1580	1582	
	1573	1584	1585	1586	1587	1600	1614	1633	1977	1980	1983	1989	1992	1995		
.IRP	474	551	1311	1337	1516	1536										
.LIST	1	247	275	407	469	551	577	598	620	668	1228	1567	1575	1576	1577	
	1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1588	1607	1621	1640	1661	
.MACRO	1	267	431	432	1567											
.FULL	247	407	577													
.MLIST	1	247	275	407	469	551	577	598	620	668	1228	1567	1575	1576	1577	
	1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1588	1607	1621	1640	1661	
.PAGE	432	528														
.P.M	1															
.FLPT	275	1029	1041	1628	1644											
.SBTTL	259	269	279	297	432	528	552	565	601	673	711	770	866	1053	1055	
	1295	1349	1420	1498	1544	1567	2023									
.TITLE	248															
.WORD	275	276	277	440	443	444	445	446	449	450	451	452	453	454	455	
	458	459	460	475	476	478	479	480	481	482	483	484	485	486	487	
	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	
	503	504	505	510	511	512	644	646	1018	1019	1020	1021	1022	1023	1024	
	1025	1026	1027	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1059	1060	
	1061	1344	1347	1416	1496	1574	1716	2014								

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* NOW, SEQ/SOL/CRF/ML:TOC/PAGNUM=DZRP2B.SML,DZRP2B.CMB
RUN-TIME: 28 35 4 SECONDS
RUN-TIME RATIO: 154/69=2.2
CORE USED: 33K (65 PAGES)

