

RP04

LOGIC TEST PART 2
MD-11-DERPQ-A

EP-DERPQ-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN US

This microfiche card contains a grid of 100 frames of logic test data, arranged in 10 rows and 10 columns. Each frame displays a complex circuit diagram or test procedure, including various logic gates, flip-flops, and timing diagrams. The text within the frames is small and dense, typical of microfiche data. The frames are separated by thin white lines, and the overall layout is a standard microfiche format.

.REM%

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DERPQ-A-0
PRODUCT NAME: RPO4 DUAL CONTROLLER LOGIC TEST - PART 2
DATE CREATED: DECEMBER 21, 1974
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: C. HESS

COPYRIGHT (C) 1974, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THE INFORMATION IN THIS STATEMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

ACTUAL DISTRIBUTION OF THE SOFTWARE DESCRIBED IN THIS DOCUMENT WILL BE SUBJECT TO TERMS AND CONDITIONS TO BE ANNOUNCED ON SOME FUTURE DATE BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

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

NO-11-DERPQ-A, RPO4 DUAL CONTROLLER LOGIC TEST - PART 2 MACY11 27(732) 05-OCT-76 14:30 PAGE 2
DERPQ.P11

CONTENTS

- 1. ABSTRACT
- 2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 PRELIMINARY PROGRAMS
 - 2.3 OTHER PROGRAMS
- 3. LOADING PROCEDURES
- 4. STARTING PROCEDURES
 - 4.1 STARTING ADDRESSES
 - 4.2 UNIBUS & VECTOR ADDRESSES
 - 4.3 OPERATOR ACTION
- 5. OPERATING PROCEDURES
 - 5.1 OPERATIONAL SWITCH SETTINGS
 - 5.2 TEST SELECTION
 - 5.3 DUAL PORT TEST CABLE CONNECTION
- 6. ERRORS
- 7. MISCELLANEOUS
 - 7.1 RESTRICTIONS
 - 7.2 LIMITATIONS
 - 7.3 EXECUTION TIME
 - 7.4 STACK POINTER
 - 7.5 SUBROUTINE CALLS
 - 7.6 REQUIRED TESTS
 - 7.7 DISK SURFACE USAGE
 - 7.8 LOOP ON ERROR OPTION
 - 7.9 TEST ITERATION
- 8. TEST DESCRIPTIONS
- 9. PROGRAM LISTING

55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
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

1. ABSTRACT

THE RPO4 DUAL CONTROLLER OPTION LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RPO4 DUAL CONTROLLER LOGIC IS FUNCTIONING PROPERLY. ONLY THE CONTROL LOGIC IS TESTED BY THIS PROGRAM; DATA HANDLING IN THE DUAL CONTROLLER MODE IS NOT TESTED BY THIS PROGRAM.

BOTH PORTS OF THE RPO4 ARE CABLED TO THE SAME MASSBUS BY A SPECIAL ADAPTER CABLE. THIS ARRANGEMENT ALLOWS THE DUAL CONTROLLER LOGIC TO BE TESTED FROM ONE PDP-11/RH70.

THIS PROGRAM IS THE SECOND PART OF THE RPO4 DUAL CONTROLLER OPTION LOGIC TEST. ONLY THE CONTROL LOGIC ASSOCIATED WITH THE UNLOAD COMMAND AND 'CONTROLLER SELECT' SWITCH IS TEST BY THIS PROGRAM.

2. REQUIREMENTS

2.1 EQUIPMENT

- PDP-11 PROCESSOR
- 16K OF MEMORY
- KW11-L OR KW11-P CLOCK
- TELETYPE
- RH70 WITH AN RPO4
- RPO4 DUAL CONTROLLER OPTION TEST CABLE

2.2 PRELIMINARY PROGRAMS

- A. RPO4 DISKLESS CONTROLLER TEST
 - PART 1 (MAINDEC-11-DERPS)
 - PART 2 (MAINDEC-11-DERPT)
- B. RPO4 FUNCTIONAL CONTROLLER TEST
 - PART 1 (MAINDEC-11-DERPU)
 - PART 2 (MAINDEC-11-DERPV)

(THE ABOVE PROGRAMS MUST BE RUN TWICE: ONCE FROM EACH CONTROLLER (PORT).)

- C. RPO4 DUAL CONTROLLER LOGIC TEST
 - PART 1 (MAINDEC-11-DERPP)

2.3 OTHER PROGRAMS

- A. DYNAMIC OPERATION OF THE DUAL CONTROLLER OPTION IS TESTED BY THE RPO4 MULTIDRIVE EXERCISER (MAINDEC-11-DERPQ-B). NOTE THAT THE RPO4 EXERCISER MUST BE PROGRAM REVISION 'B' OR LATER. REVISION 'A' OF THE RPO4 EXERCISER DOES NOT SUPPORT DUAL CONTROLLER OPERATION.

E01

143
143
144
145

3. LOADING PROCEDURES

1
2

146
147
148
149
150
151
152
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

THE PROGRAM MAY BE LOADED BY THE ABSOLUTE PAPER TAPE LOADER OR IT MAY BE LOADED FROM THE APPROPRIATE MEDIA USING THE ASSOCIATED 'XXDP' LOADER.

4. STARTING PROCEDURES

4.1 STARTING ADDRESSES

- A. THE NORMAL STARTING ADDRESS OF THE PROGRAM IS LOCATION 200(8). STARTING AT THIS ADDRESS ALLOWS THE OPERATOR TO SELECT (OR RESELECT) THE DRIVE ADDRESS OF THE RPO4 TO BE TESTED.
- B. THE RESTART ADDRESS IS LOCATION 204(8). THE PROGRAM WILL USE THE CURRENT DRIVE ADDRESS.
- C. THE PROGRAM CAN BE STARTED AT LOCATION 210(8) TO ALLOW THE RH70 ADDRESS TO BE CHANGED.

4.2 UNIBUS & VECTOR ADDRESSES

THE PROGRAM ASSURES THE FOLLOWING UNIBUS AND VECTOR ADDRESSES. THESE ADDRESSES MAY BE CHANGED PRIOR TO INITIATING A PROGRAM START AT ANY OF THE STARTING LOCATIONS.

MEMORY LOCATION	CONTENTS	FUNCTION
-----	-----	-----
1136	177560	TTY KEYBOARD STATUS REG
1140	177562	TTY KEYBOARD BUFFER REG
1142	177564	TTY PRINTER STATUS REG
1144	177566	TTY PRINTER BUFFER REG
1204	172540	KW11-P STATUS REG
1206	172542	KW11-L COUNTER BUFFER
1210	104	KW11-P VECTOR ADDRESS
1212	177546	KW11-L STATUS REGISTER
1214	100	KW11-L VECTOR ADDRESS
1262	176700	RH70/RPO4 ADDRESS
1264	254	RH70 INTERRUPT VECTOR ADDRESS

4.3 OPERATOR ACTION

- A. CONNECT THE DUAL CONTROLLER TEST CABLE BETWEEN BUS A & BUS B ON THE RPO4 BEING TESTED. (SEE SECTION 5.3)
- B. LOAD THE PROGRAM INTO MEMORY IN THE PROPER PROCESSOR.
- C. SWITCH THE 'CONTROLLER SELECT' SWITCH ON THE RPO4 TO BE TESTED TO THE 'A/B' POSITION. CYCLE THE DRIVE UP.
- D. LOAD THE APPROPRIATE STARTING ADDRESS (200(8) OR 210(8)) INTO THE SWITCH REGISTER.
- E. PRESS START.
- F. ENTER THE DRIVE NUMBER. (THIS MUST BE THE NUMBER

GO1

MD-11-DERPQ-A, RPO4 DUAL CONTROLLER LOGIC TEST - PART 2 MACY11 27(732) 05-OCT-76 14:30 PAGE 7
DERPQA.P11

202
203
204
205

DISPLAYED IN THE DRIVE NUMBER LED ON THE CONTROL
PANEL.)
G. ENTER THE NUMBER OF THE TEST TO BE RUN.
H. THE PROGRAM CAN BE STOPPED AT ANY TIME AND RESTARTED

FROM LOCATION 204.

206
207
208
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
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261

5. OPERATING PROCEDURES

5.1 OPERATIONAL SWITCH SETTINGS

WITH ALL SWITCHES SET TO ZERO, THE PROGRAM WILL TYPE ALL ERRORS AND CONTINUE TESTING.

THE SWITCH SETTINGS ARE:

- SW<15>=1...HALT ON ERROR
- SW<14>=1...LOOP ON TEST
- SW<13>=1...INHIBIT ERROR TYPEOUTS
- SW<11>=1...INHIBIT TEST ITERATIONS
- SW<10>=1...RING TTY BELL ON ERROR
- SW<09>=1...LOOP ON ERROR

5.2 TEST SELECTION

INDIVIDUAL TESTS ARE SELECTED IN RESPONSE TO THE 'ENTER TEST NUMBER:' MESSAGE. ANY VALID TEST NUMBER CAN BE ENTERED. EACH ENTRY MUST BE TERMINATED BY A CARRIAGE RETURN (CR). THE LOOP ON TEST SWITCH, SW<15>, MUST BE SET TO ALL CONTINUOUS EXECUTION OF THE SELECTED TEST.

TO RUN ALL TESTS IN SEQUENCE, ENTER EITHER A '0' FOLLOWED BY A CARRIAGE RETURN, OR A CARRIAGE RETURN BY ITSELF. THE PROGRAM WILL THEN EXECUTE ALL TESTS IN SEQUENCE UNTIL IT IS HALTED.

THE 'RUBOUT KEY' (RO) CAN BE USED TO DELETE THE LAST CHARACTER ENTERED. SUCCESSIVE STUCKING AT THE RO KEY WILL DELETE CHARACTERS UNTIL THE PREVIOUS CHARACTERS HAVE BEEN DELETED. CHARACTERS DELETED BY THE RO KEY WILL BE TYPED AND WILL BE SEPARATED BY '\ ' FROM THE CHARACTERS ENTERED BY THE OPERATOR.

THE OPERATOR CAN DELETE THE ENTIRE ENTRY BY TYPING A 'CONTROL U' (↑U).

5.3 TEST CABLE CONNECTION

TO TEST THE RPO4 DUAL CONTROLLER OPTION WITH THIS PROGRAM, A SPECIAL TEST CABLE MUST BE USED. (THE TEST CABLE IS P/N 7010507-02). THE TEST CABLE CONNECTS MASSBUS A & MASSBUS B TOGETHER AT THE RPO4 BEING TESTED AND IS CONSTRUCTED SO THAT BIT 0 OF THE MASSBUS UNIT SELECT LINES IS COMPLEMENTED.

WITH THE TEST CABLE CONNECTED TO THE RPO4 UNDER TEST, THE DRIVE APPEARS AS TWO UNITS ON THE MASSBUS: EACH PORT OF THE RPO4 WILL RESPOND TO A DIFFERENT MASSBUS ADDRESS. THE ADDRESS OF EACH PORT WILL DEPEND UPON THE DRIVE'S ADDRESS (THE ADDRESS SELECTED BY THE SWITCHES ON THE

I01

MD-11-DERPO-A, RPO4 DUAL CONTROLLER LOGIC TEST - PART 2 MACY!1 27(732) 05-OCT-76 14:30 PAGE 9
DERPOA.P11

262
263
264
265

'UP' BOARD - MODULE M7775.)

THE PROGRAM WILL TYPEOUT THE APPARENT ADDRESSES OF BOTH
PORTS. (ONE PORT WILL HAVE THE ADDRESS OF THE DRIVE; THE

OTHER PORT WILL HAVE THE ADDRESS DEVELOPED BY THE CABLE).

* ANY DEVICE ON THE SYSTEM (RPO4 OR NON-RPO4 DEVIC%

.TITLE MD-11-DERPQ-A, RPO4 DUAL CONTROLLER LOGIC TEST - PART 2
;*COPYRIGHT (C) 1974
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*PROGRAM BY C. HESS
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-A2).
;*

.SBTTL OPERATIONAL SWITCH SETTINGS

SWITCH	USE
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
11	INHIBIT ITERATIONS
10	BELL ON ERROR
9	LOOP ON ERROR

.SBTTL BASIC DEFINITIONS

001100	STACK= 1100	;; INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
177776	.EQUIV EMT, ERROR	;; BASIC DEFINITION OF ERROR CALL
177774	.EQUIV IOT, SCOPE	;; BASIC DEFINITION OF SCOPE CALL
177772	PS= 177776	;; PROCESSOR STATUS WORD
177570	.EQUIV PS, PSW	
177570	STKLM= 177774	;; STACK LIMIT REGISTER
177570	PIRQ= 177772	;; PROGRAM INTERRUPT REQUEST REGISTER
177570	SWR= 177570	;; SWITCH REGISTER
	DISPLAY=SWR	

.GENERAL PURPOSE REGISTER DEFINITIONS

000000	R0= %0	;; GENERAL REGISTER
000001	R1= %1	;; GENERAL REGISTER
000002	R2= %2	;; GENERAL REGISTER
000003	R3= %3	;; GENERAL REGISTER
000004	R4= %4	;; GENERAL REGISTER
000005	R5= %5	;; GENERAL REGISTER
000006	R6= %6	;; GENERAL REGISTER
000007	R7= %7	;; GENERAL REGISTER
	.EQUIV R6, SP	;; STACK POINTER
	.EQUIV R7, PC	;; PROGRAM COUNTER

266
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
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321

322
323
324 000000
325 000040
326 000100
327 000140
328 000200
329 000240
330 000300
331 000340
332
333
334 100000
335 040000
336 020000
337 010000
338 004000
339 002000
340 001000
341 000400
342 000200
343 000100
344 000040
345 000020
346 000010
347 000004
348 000002
349 000001
350
351
352
353
354
355
356
357
358
359
360
361
362 100000
363 040000
364 020000
365 010000
366 004000
367 002000
368 001000
369 000400
370 000200
371 000100
372 000040
373 000020
374 000010
375 000004
376 000002
377 000001

.*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
.EQUIV SW06, SW6
.EQUIV SW05, SW5
.EQUIV SW04, SW4
.EQUIV SW03, SW3
.EQUIV SW02, SW2
.EQUIV SW01, SW1
.EQUIV SW00, SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)
BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1

5/15
2/26/76

```

378 .EQUIV BIT09,BIT9
379 .EQLIV BIT08,BIT8
380 .EQUIV BIT07,BIT7
381 .EQUIV BIT06,BIT6
382 .EQUIV BIT05,BIT5
383 .EQUIV BIT04,BIT4
384 .EQUIV BIT03,BIT3
385 .EQUIV BIT02,BIT2
386 .EQUIV BIT01,BIT1
387 .EQUIV BIT00,BIT0

```

```

388
389 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
390 000004 ERRVEC= 4 ;:TIME OUT AND OTHER ERRORS
391 000010 RESVEC= 10 ;:RESERVED AND ILLEGAL INSTRUCTIONS
392 000014 TBITVEC=14 ;:"T" BIT
393 000014 TRTVEC= 14 ;:TRACE TRAP
394 000014 BPTVEC= 14 ;:BREAKPOINT TRAP (BPT)
395 000020 IOTVEC= 20 ;:INPUT/OUTPUT TRAP (IOT) **SCOPE**
396 000024 PWFVEC= 24 ;:POWER FAIL
397 000030 EMTVEC= 30 ;:EMULATOR TRAP (EMT) **ERROR**
398 000034 TRAPVEC=34 ;:"TRAP" TRAP
399 000060 TKVEC= 60 ;:TTY KEYBOARD VECTOR
400 000064 TPVEC= 64 ;:TTY PRINTER VECTOR
401 000240 PIRQVEC=240 ;:PROGRAM INTERRUPT REQUEST VECTOR

```

402 ;:*****

403 .SBTTL RH11 REGISTERS

404 ;:*****

```

405 ;WORD COUNT REGISTER (RHWC)
406 ;EACH BIT IS CALLED BY BIT NUMBER

```

```

407 ;BUS ADDRESS REGISTER (RHBA)
408 ;EACH BIT IS CALLED BY BIT NUMBER

```

409 ;CONTROL AND STATUS REGISTER 2 (RHCS2)

```

410
411
412
413
414
415
416
417 000001 US1= 1 ;:UNIT SELECT (BIT #0)
418 000002 US2= 2 ;:UNIT SELECT (BIT #1)
419 000004 US4= 4 ;:UNIT SELECT (BIT #2)
420 000010 BAI= 10 ;:BUS ADDRESS INCREMENT INHIBIT (BIT #3)
421 000020 PAT= 20 ;:MASSBUS PARITY TEST (BIT #4)
422 000040 CLR= 40 ;:CLEAR (BIT #5)
423 000100 IR= 100 ;:INPUT READY (BIT #6)
424 000200 OR= 200 ;:OUTPUT READY (BIT #7)
425 000400 MPE= 400 ;:MASS BUS PARITY ERROR (BIT #8)
426 001000 MXF= 1000 ;:MISSED TRANSFER ERROR (BIT #9)
427 002000 PGE= 2000 ;:PROGRAM ERROR (BIT #10)
428 004000 NEM= 4000 ;:NON EXISTANT MEMORY (BIT #11)
429 010000 NED= 10000 ;:NON EXISTANT DRIVE (BIT #12)
430 020000 UPE= 20000 ;:UNIBUS PARITY ERROR (BIT #13)
431 040000 WCE= 40000 ;:WRITE CHECK ERROR (BIT #14)
432 100000 DLT= 100000 ;:DATA LATE (BIT #15)
433

```

434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489

;DATA BUFFER REGISTER (RHDB)
;EACH BIT IS CALLED BY BIT NUMBER

;*****

.SBTTL RPO4 REGISTERS

;*****

;CONTROL AND STATUS 1 REGISTER. (#00)

000001
000100
000200
000400
001000
002000
004000
020000
040000
100000

GO= 1
IE= 100
RDY= 200
A16= 400
A17= 1000
PSEL= 2000
DVA= 4000
MCPE= 20000
TRE= 40000
SC= 100000

;GO (BIT #0)
;INTERRUPT ENABLE (BIT #6)
;READY (BIT #7)
;HIGH ORDER UNIBUS BITS (BIT #8)
;HIGH ORDER UNIBUS BITS (BIT #9)
;PORT SELECT (BIT #10)
;DEVICE AVAILABLE (BIT #11)
;MASSBUSS PARITY ERROR (BIT #13)
;TRANSFER ERROR (BIT #14)
;SPECIAL CONDITION (BIT #15)

;STATUS REGISTER (RHDS1) (#01)

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

:DFS= 1
OFF20= 2
DIGB= 4
GRV= 10
DL64= 20
DE1= 40
VV= 100
DRY= 200
DPR= 400
PGM= 1000
LST= 2000
WRL= 4000
MOL= 10000
PIP= 20000
ERR= 40000
ATA= 100000

DRIVE FORWARD 5"/SEC. (BIT #0)
;DRIVE FORWARD 20"/SEC. (BIT #1)
;DRIVE TO INNER GUARD BAND (BIT #2)
;GO REVERSE (BIT #3)
;DIFFERENCE LESS THAN 64 (BIT #4)
;DIFFERENCE EQUALS 1 (BIT #5)
;VOLUME VALID (BIT #6)
;DRIVE READY (BIT #7)
;DRIVE PRESENT (BIT #8)
;PROGRAMABLE (BIT #9)
;LAST SECTOR TRANSFERRED (BIT #10)
;WRITE LOCK (BIT #11)
;MEDIUM ON-LINE (BIT #12)
;POSITIONING OPERATION IN PROGRESS (BIT #13)
;COMPOSIT ERROR. (BIT #14)
;ATTENTION ACTIVE (BIT #15)

;ERROR REGISTER #01 (RHER1) (#02)

000001
000002
000004
000010
000020
000040
000100
000200
000400
001000
002000
004000

ILF= 1
ILR= 2
RMR= 4
PAR= 10
FER= 20
WCF= 40
ECH= 100
HCE= 200
HCRC= 400
AOE= 1000
IAE= 2000
WLE= 4000

;ILLEGAL FUNCTION (BIT #0)
;ILLEGAL REGISTER (BIT #1)
;REGISTER MODIFICATION REFUSED (BIT #2)
;PARITY ERROR (BIT #3)
;FORMAT ERROR (BIT #4)
;WRITE CLOCK FAIL (BIT #5)
;ECC HARD ERROR (BIT #6)
;HEADER COMPARE ERROR (BIT #7)
;HEADER CRC ERROR (BIT #8)
;ADDRESS OVERFLOW ERROR (BIT #9)
;INVALID ADDRESS ERROR (BIT #10)
;WRITE LOCK ERROR (BIT #11)

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
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601

000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
100000

000001
000002
000004
000010
000020
000040
000200
002000
004000
010000

000001
000002
000010
000020
000040
000100
004000
100000

CSU= 10
MSE= 20
TDF= 40
TUF= 100
FEN= 200
WRU= 400
MHS= 1000
NHS= 2000
IXE= 4000
VJ30= 10000
PLU= 20000
ACU= 100000

;CURRENT SWITCH UNSAFE (BIT #2)
;MOTOR SEQUENCE ERROR (BIT #4)
;TRANSITIONS DETECTOR FAILURE (BIT #5)
;TRANSITIONS UNSAFE (BIT #6)
;FAILSAFE ENABLED (BIT #7)
;WRITE READY UNSAFE (BIT #8)
;MULTIPLE HEAD SELECT (BIT #9)
;NO HEAD SELECTION (BIT #10)
;INDEX ERROR (BIT #11)
;30VOLT UNSAFE (BIT #12)
;PLO UNSAFE (BIT #13)
;AC UNSAFE (BIT #15)

;OFFSET REGISTER (RHOF) (#11)

OF25= 1
OF50= 2
OF100= 4
OF200= 10
OF400= 20
OF800= 40
OFREV= 200
HCI= 2000
ECI= 4000
FMT22= 10000

;OFFSET 25 MICRO INCHES (BIT #0)
;OFFSET 50 MICRO INCHES (BIT #1)
;OFFSET 100 MICRO INCHES (BIT #2)
;OFFSET 200 MICRO INCHES (BIT #3)
;OFFSET 400 MICRO INCHES (BIT #4)
;OFFSET 800 MICRO INCHES (BIT #5)
;OFFSET NEGATIVE (REVERSE) (BIT #5)
;HEADER COMPARE INHIBIT (BIT #10)
;ERROR CORRECTION CODE INHIBIT (BIT #11)
;FORMAT BIT (BIT #12)

;DESIRED CYLINDER ADDRESS (RHCA) (#12)
;EACH BIT IS CALLED BY BIT NUMBER.

;CURRENT CYLINDER ADDRESS (RHCC) (#13)
;EACH BIT IS CALLED BY BIT NUMBER

;SERIAL NUMBER REGISTER (RHSN) (#14)
;EACH IS CALLED BY BIT NUMBER

;ERROR REGISTER #03 (RHER3) (#15)

PSU= 1
VUF= 2
UWR= 10
PRE= 20
ACL= 40
DCL= 100
SKI= 40000
OCYL= 100000

;PACK SPEED UNSAFE (BIT #0)
;VELOCITY UNSAFE (BIT #1)
;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
;DISK PCK ROTATION ERROR (BIT #4)
;AC LOW (BIT #5)
;DC LOW (BIT #6)
;SEEK INCOMPLETE (BIT #14)
;OFF CYLINDER (BIT #15)

;ECC POSITION REGISTER (RHEC1) (#16)
;EACH BIT IS CALLED BY BIT NUMBER

;ECC PATTERN REGISTER (RHEC2) (#17)
;EACH BIT IS CALLED BY BIT NUMBER

;*****

.SBTTL DEFINITIONS OF THE RH11/RPO4 ADDRESS INDEXES

```

602          ;:*****
603
604          000000          RHCS1=0          ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
605          000002          RHWC=2          ;WORD COUNT REGISTER (NOT A DRIVE REG)
606          000004          RHBA=4          ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
607          000006          RHDA=6          ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
608          000010          RHCS2=10         ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
609          000012          RHDS1=12         ;DRIVE STATUS REGISTER (DRIVE REG 01)
610          000014          RHER1=14         ;ERROR REGISTER #1 (DRIVE REG. 02)
611          000016          RHAS=16         ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
612          000020          RHLA=20         ;LOOK AHEAD REGISTER (DRIVE REG. 07)
613          000022          RHDB=22         ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
614          000024          RHMR=24         ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
615          000026          RHDT=26         ;DRIVE TYPE REGISTER (DRIVE REG. 06)
616          000030          RHSN=30         ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
617          000032          RHOF=32         ;OFFSET REGISTER (DRIVE REG. 11)
618          000034          RHCA=34         ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
619          000036          RHCC=36         ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
620          000040          RHER2=40         ;ERROR REGISTER #2 (DRIVE REG. 14)
621          000042          RHER3=42         ;ERROR REGISTER #3 (DRIVE REG. 15)
622          000044          RHEC1=44         ;ECC POSITION REGISTER (DRIVE REG. 16)
623          000046          RHEC2=46         ;ECC PATTERN REGISTER (DRIVE REG. 17)

```

.SBTTL TRAP CATCHER

```

624
625          .=0
626          000000          ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
627
628          ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
629          ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
630
631          .SBTTL STARTING ADDRESS(ES)
632          .=200
633
634          000200          JMP      J#START          ;: JUMP TO STARTING ADDRESS OF PROGRAM
635          000200 000137 001676          ;*STARTING ADDRESS IS LOCATION 200
636
637          000204          JMP      EXEC          ;: RESTART
638          000204 000137 002262          ;*RESTART ADDRESS IS LOCATION 204
639
640          000210          JMP      CHANGE          ;: CHANGE RH11 ADDRESS AND START
641          000210 000137 002430          ;*START AT LOCATION 210 TO CHANGE THE RH11 ADDRESS FROM 176700
642
643
644
645
646

```

647					:*****
648					
649					.SBTTL COMMON TAGS
650					
651					:*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
652					:*USED IN THE PROGRAM.
653					
654		000046		.=46	
655	000046	017726		\$ENDAD	;;LOGICAL END OF PROGRAM
656					
657		001100		.=1100	
658					
659	001100			\$CMTAG:	;; START OF COMMON TAGS
660	001100	000000		\$PASS: .WORD 0	;; CONTAINS PASS COUNT
661	001102	000		\$TSTNM: .BYTE 0	;; CONTAINS THE TEST NUMBER
662	001103	000		\$ERFLG: .BYTE 0	;; CONTAINS ERROR FLAG
663	001104	000000		\$ICNT: .WORD 0	;; CONTAINS SUBTEST ITERATION COUNT
664	001106	000000		\$LPADR: .WORD 0	;; CONTAINS SCOPE LOOP
665	001110	000000		\$LPERR: .WORD 0	;; CONTAINS SCOPE RETURN FOR ERRORS
666	001112	000000		\$ERTTL: .WORD 0	;; CONTAINS TOTAL ERRORS DETECTED
667	001114	000		\$ITEMB: .BYTE 0	;; CONTAINS ITEM CONTROL BYTE
668	001115	001		\$ERMAX: .BYTE 1	;; CONTAINS MAX. ERRORS PER TEST
669	001116	000000		\$ERRPC: .WORD 0	;; CONTAINS PC OF LAST ERROR INSTRUCTION
670	001120	000000		\$GDADR: .WORD 0	;; CONTAINS OF 'GOOD' DATA
671	001122	000000		\$BDADR: .WORD 0	;; CONTAINS OF 'BAD' DATA
672	001124	000000		\$GDDAT: .WORD 0	;; CONTAINS 'GOOD' DATA
673	001126	000000		\$BDDAT: .WORD 0	;; CONTAINS 'BAD' DATA
674	001130	000000	000000 000000	.WORD 0,0,0	;; RESERVED--NOT TO BE USED
675	001136	177560		\$TKS: 177560	;; TTY KBD STATUS
676	001140	177562		\$TKB: 177562	;; TTY KBD BUFFER
677	001142	177564		\$TPS: 177564	;; TTY PRINTER STATUS REG.
678	001144	177566		\$TPB: 177566	;; TTY PRINTER BUFFER REG.
679	001146	000		\$NULL: .BYTE 0	;; CONTAINS NULL CHARACTER FOR FILLS
680	001147	002		\$FILLS: .BYTE 2	;; CONTAINS # OF FILLER CHARACTERS REQUIRED
681	001150	012		\$FILLC: .BYTE 12	;; INSERT FILL CHARS. AFTER A "LINE FEED"
682	001151	000		\$TPFLG: .BYTE 0	;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
683	001152	000000		\$REGAD: .WORD 0	;; CONTAINS THE FROM
684					;; WHICH (\$REGO) WAS OBTAINED
685	001154	000000		\$REGO: .WORD 0	;; CONTAINS ((\$REGAD)+0)
686	001156	000000		\$TMP0: .WORD 0	;; USER DEFINED
687	001160	000000		\$TMP1: .WORD 0	;; USER DEFINED
688	001162	000000		\$TMP2: .WORD 0	;; USER DEFINED
689	001164	000000		\$TMP3: .WORD 0	;; USER DEFINED
690	001166	000000		\$TMP4: .WORD 0	;; USER DEFINED
691	001170	000000		\$TIMES: 0	;; MAX. NUMBER OF ITERATIONS
692	001172	000000		\$ESCAPE: 0	;; ESCAPE ON ERROR
693	001174	177607	000377	\$BELL: .ASCIZ <207><377><377>	;; CODE FOR BELL
694	001200	077		\$QUES: .ASCII /?/	;; QUESTION MARK
695	001201	015		\$CRLF: .ASCII <15>	;; CARRIAGE RETURN
696	001202	000012		\$LF: .ASCIZ <12>	;; LINE FEED
697	001204	172540		\$LKCSR: .WORD 172540	;; ADDR OF KW11-P STATUS REGISTER
698	001206	172542		\$LKCSB: .WORD 172542	;; ADDR OF KW11-P COUNTER BUFFER
699	001210	000104		\$LPVEC: .WORD 104	;; ADDR OF KW11-P VECTOR
700	001212	177546		\$LKS: .WORD 177546	;; ADDR OF KW11-L STATUS REGISTER
701	001214	000100		\$LLVEC: .WORD 100	;; ADDR OF KW11-L VECTOR
702	001216	000000		PORTA: .WORD 0	;; ADDRESS OF PORT A

703	001220	000000	PORTB: .WORD	0	; ADDRESS OF PORT B
704	001222	000000	PORTC: .WORD	0	; ADDRESS OF DIFFERENT DRIVE
705	001224	000000	ASR1: .WORD	0	; ATA-A OR ATA-B = 1
706	001226	000000	PTNBR: .WORD	0	; CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
707	001230	000000	SEIZPT: .WORD	0	; CONTAINS THE ADDRESS OF THE SEIZING PORT
708	001232	000000	OPPR: .WORD	0	; CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
709	001234	000000	TSTNUM: .WORD	0	; NUMBER OF THE CURRENT TEST
710	001236	000000	CKERR: .WORD	0	; IF -1, A REGISTER MISCOMPARISON OCCURED
711	001240	000000	NOSEIZ: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
712	001242	000000	RELERR: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
713	001244	000000	TIME: .WORD	0	; ELAPSED TIME COUNTER
714	001246	000000	WATCH: .WORD	0	; WATCH DOG TIMER LOCATION
715	001250	000000	TIMEA: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
716	001252	000000	TIMEAP: .WORD	0	; PORT A TIMEOUT VALUE + 25%
717	001254	000000	TIMEB: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
718	001256	000000	TIMEBP: .WORD	0	; PORT B TIMEOUT VALUE + 25%
719	001260	000000	KYBCTL: .WORD	0	; SINGLE TEST INDICATOR
720					
721			;*****		
722			.SBTTL RH11/RPO4 UNIBUS AND VECTOR ADDRESSES		
723					
724			;*****		
725					
726					
727	001262	176700	\$RPADR: .WORD	176700	; RH11/RPO4 UNIBUS ADDRESS
728	001264	000254	\$RPVEC: .WORD	254	; RH11 INTERRUPT VECTOR ADDRESS
729					

730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785

.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 \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$FRRPC).
;*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

\$ERRTB:

;ERROR 1

EM1 ;DRIVE IS NON-EXISTENT ('NED' BIT SET)
DH1
DT1
DF1

;ERROR 2

EM2 ;WRONG DRIVE TYPE
DH2
DT2
DF2

;ERROR 3

EM3 ;CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
DH1
DT1
DF1

;ERROR 4

EM4 ;DRIVE NOT ON LINE
DH2
DT2
DF2

;ERROR 5

EM5 ;SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME
DH5
DT5
DF5

;ERROR 6

001266

001266 023331
001270 026036
001272 027410
001274 027624

001276 023377
001300 026107
001302 027424
001304 027631

001306 023420
001310 026136
001312 027410
001314 027624

001316 023477
001320 026107
001322 027424
001324 027631

001326 023521
001330 026163
001332 027442
001334 027637

786	001336	023603	EM6	;TIMEOUT HAS NOT OCCURED WITHIN 2 SECONDS
787	001340	026232	DH6	
788	001342	027456	DT6	
789	001344	027644	DF6	
790				
791				;ERROR 7
792				
793	001346	023654	EM7	;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
794	001350	026261	DH7	
795	001352	027466	DT7	
796	001354	027647	DF7	
797				
798				;ERROR 10
799				
800	001356	023721	EM10	;READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT
801	001360	026036	DH1	
802	001362	027410	DT1	
803	001364	027624	DF1	
804				
805				;ERROR 11
806				
807	001366	024006	EM11	; 'GO' BIT RESET DURING UNLOAD COMMAND
808	001370	026036	DH1	
809	001372	027410	DT1	
810	001374	027624	DF1	
811				
812				;ERROR 12
813				
814	001376	024053	EM12	;INCORRECT STATUS DURING UNLOAD COMMAND
815	001400	026036	DH1	
816	001402	027410	DT1	
817	001404	027624	DF1	
818				
819				;ERROR 13
820				
821	001406	024122	EM13	;DRIVE DID NOT RETURN TO NEUTRAL AFTER UNLOAD COMMAND
822	001410	026326	DH13	
823	001412	027500	DT13	
824	001414	027644	DF6	
825				
826				;ERROR 14
827				
828	001416	024207	EM14	;ATTENTION BIT SET ON 'OPPOSITE PORT' AFTER UNLOAD
829	001420	026404	DH14	
830	001422	027510	DT14	
831	001424	027653	DF14	
832				
833				;ERROR 15
834				
835	001426	024271	EM15	;ATTENTION BIT NOT SET ON PORT WHICH ISSUED 'UNLOAD'
836	001430	026036	DH1	
837	001432	027410	DT1	
838	001434	027624	DF1	
839				
840				;ERROR 16
841				

H02

842	001436	024355	EM16	;DRIVE NOT IN NEUTRAL AFTER UNLOAD WITH 'CONTROLLER
843				;SELECT' SWITCH MOVED FROM 'A/B'
844	001440	026326	DH13	
845	001442	027500	DT13	
846	001444	027644	DF6	
847				;ERROR 17
848				
849	001446	024501	EM17	;DRIVE LOCKED ON PORT 'A' BY SWITCH WHILE CYCLED UP
850	001450	026524	DH17	
851	001452	027526	DT17	
852	001454	027661	DF17	
853				
854				;ERROR 20
855				
856	001456	024564	EM20	;DRIVE LOCKED ON PORT 'B' BY SWITCH WHILE CYCLED UP
857	001460	026524	DH17	
858	001462	027526	DT17	
859	001464	027661	DF17	
860				
861				;ERROR 21
862				
863	001466	024647	EM21	;STATUS INCORRECT FOR PORT AFTER CYCLE UP
864	001470	026036	DH1	
865	001472	027410	DT1	
866	001474	027624	DF1	
867				
868				;ERROR 22
869				
870	001476	024720	EM22	;REGISTER CONTENTS SEEN WHEN DRIVE SWITCHED ON 'OPPOSITE' PORT
871	001500	026036	DH1	
872	001502	027410	DT1	
873	001504	027624	DF1	
874				
875				;ERROR 23
876				
877	001506	025016	EM23	; 'NED' SET WHEN RHDS1 ACCESSED THROUGH PORT NOT SWITCHED
878	001510	026036	DH1	
879	001512	027410	DT1	
880	001514	027624	DF1	
881				
882				;ERROR 24
883				
884	001516	025106	EM24	;DRIVE SWITCHED TO LOCKED OUT PORT WHEN RELEASED/
885	001520	026543	DH24	
886	001522	027534	DT24	
887	001524	027647	DF7	
888				
889				;ERROR 25
890				
891	001526	000000	0	;UNUSED ERROR MESSAGES
892	001530	000000	0	
893	001532	000000	0	
894	001534	000000	0	
895				
896				;ERROR 26
897				

898	001536	000000	0	;UNUSED ERROR MESSAGES
899	001540	000000	0	
900	001542	000000	0	
901	001544	000000	0	
902				
903				;ERROR 27
904				
905	001546	000000	0	;UNUSED ERROR MESSAGES
906	001550	000000	0	
907	001552	000000	0	
908	001554	000000	0	
909				
910				;ERROR 30
911				
912	001556	025166	EM30	;DRIVE NOT SEIZED BY PORT 'N'
913	001560	026647	DH30	
914	001562	027546	DT30	
915	001564	027663	DF30	
916				
917				;ERROR 31
918				
919	001566	025217	EM31	;WRONG STATUS SEEN BY THE SEIZING PORT
920	001570	026107	DH2	
921	001572	027424	DT2	
922	001574	027631	DF2	
923				
924				;ERROR 32
925				
926	001576	025265	EM32	;REGISTER CONTENTS INCORRECT
927	001600	026107	DH2	
928	001602	027424	DT2	
929	001604	027631	DF2	
930				
931				;ERROR 33
932				
933	001606	025315	EM33	;CONTROL BUS PARITY ERROR WHILE READING REGISTER
934	001610	026036	DH1	
935	001612	027410	DT1	
936	001614	027624	DF1	
937				
938				;ERROR 34
939				
940	001616	025401	EM34	;CAN'T ACCESS DRIVE THROUGH EITHER PORT
941	001620	026772	DH34	
942	001622	027566	DT34	
943	001624	027672	DF34	
944				
945				;ERROR 35
946				
947	001626	025450	EM35	;DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
948	001630	027070	DH35	
949	001632	027600	DT35	
950	001634	027647	DF7	
951				
952				;ERROR 36
953				

954	001636	025535	EM36	;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
955	001640	027070	DH35	
956	001642	027600	DT35	
957	001644	027647	DF7	
958				
959				;ERROR 37
960				
961	001646	025622	EM37	;REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
962	001650	027166	DH37	
963	001652	027546	DT30	
964	001654	027663	DF30	
965				
966				;ERROR 40
967				
968	001656	025703	EM40	;DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SET
969	001660	027311	DH40	
970	001662	027612	DT40	
971	001664	027647	DF7	
972				
973				;ERROR 41
974				
975	001666	025760	EM41	;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
976	001670	026647	DH30	
977	001672	027546	DT30	
978	001674	027663	DF30	
979				

980									
981									
982	001676			START:					
983	001676	012737	000340	177776	MOV	#340, @#PS	;; LOCK OUT ALL INTERRUPTS		
984	001704	012706	001100		MOV	#\$CMTAG, R6	;; FIRST LOCATION TO BE CLEARED		
985	001710	005026			CLR	(R6)+	;; CLEAR MEMORY LOCATION		
986	001712	022706	001136		CMP	#\$STKS, R6	;; DONE?		
987	001716	001374			BNE	.-6	;; LOOP BACK IF NO		
988	001720	012706	001100		MOV	#\$STACK, SP	;; SETUP THE STACK POINTER		
989	001724	012737	017762	000020	MOV	#\$SCOPE, @#IOTVEC	;; IOT VECTOR FOR SCOPE ROUTINE		
990	001732	012737	000340	000022	MOV	#340, @#IOTVEC+2	;; LEVEL 7		
991	001740	012737	020140	000030	MOV	#\$ERROR, @#EMTVEC	;; EMT VECTOR FOR ERROR ROUTINE		
992	001746	012737	000340	000032	MOV	#340, @#EMTVEC+2	;; LEVEL 7		
993	001754	012737	021750	000034	MOV	#\$TRAP, @#TRAPVEC	;; TRAP VECTOR FOR TRAP CALLS		
994	001762	012737	000340	000036	MOV	#340, @#TRAPVEC+2	;; LEVEL 7		
995	001770	013737	017656	017650	MOV	SENDCT, SEOPCT	;; SETUP END-OF-PROGRAM COUNTER		
996	001776	005037	001170		CLR	\$TIMES	;; INITIALIZE NUMBER OF ITERATIONS		
997	002072	005037	001172		CLR	\$ESCAPE	;; CLEAR THE ESCAPE ON ERROR ADDRESS		
998	002006	012737	000001	001115	MOVB	#1, \$ERMAX	;; ALLOW ONE ERROR PER TEST		
999	002014	012737	002014	001106	MOV	#, \$SLPADR	;; INITIALIZE THE LOOP ADDRESS FOR SCOPE		
1000	002022	012737	002022	001110	MOV	#, \$SLPERR	;; SETUP THE ERROR LOOP ADDRESS		
1001	002030	000005			RESET		;; CLEAR THE SYSTEM		
1002	002032	104400	022014		START1: TYPE	TITLE	;; TYPE PROGRAM NAME		
1003	002036	012737	000240	002032	MOV	#\$NOP, START1	;; DISABLE TITLE TYPEOUT AFTER INITIAL START		
1004	002044	012737	000240	002034	MOV	#\$NOP, START1+2	;; FROM LOCATION 200 OR 210		
1005	002052	104400	022117		IS: TYPE	, ENTERA	;; ENTER DRIVE ADDRESS		
1006	002056	104416			RDOCT		;; GET THE ADDRESS		
1007	002060	012637	001216		MOV	(SP)+, PORTA	;; STORE THE ADDRESS		
1008	002064	023727	001216	000007	CMP	PORTA, #7	;; SEE IF ADDRESS TOO LARGE		
1009	002072	011403			3LOS	2\$;; BR IF NOT		


```

1010 002074 104400 022147          TYPE      ADRERR          ;TYPE ADDRESS ERROR MESSAGE
1011 002100 000764          BR          1$          ;TRY AGAIN
1012 002102 013737 001216 001220 2$: MOV      PORTA,PORTB    ;GENERATE THE PORT B ADDRESS
1013 002110 005237 001220          INC      PORTB        ;INCREMENT THE ADDRESS
1014 002114 042737 000006 001220          BIC      #6,PORTB     ;LEAVE BIT 0
1015 002122 013746 001216          MOV      PORTA,-(SP)   ;PUT PORT A ADDRESS ON THE STACK
1016 002126 042716 177771          SIC      #1C6,(SP)    ;SAVE BITS 1 & 2
1017 002132 052637 001220          BIS      (SP)+,PORTB  ;SET BITS 1 & 2 IN PORT B ADDRESS
1018 002136 104400 022171          TYPE      PORTA IS    ;'PORT A ADDRESS IS '
1019 002142 013746 001216          MOV      PORTA,-(SP)  ;PUT THE ADDRESS ON THE STACK
1020 002146 104410          TYPDS     PORTA       ;TYPE PORT A ADDRESS
1021 002150 104400 022217          TYPE      PORTB IS    ;'PORT B ADDRESS IS '
1022 002154 013746 001220          MOV      PORTB,-(SP)  ;PUT ADDRESS ON THE STACK
1023 002160 104410          TYPDS     PORTB       ;TYPE PORT B ADDRESS
1024 002162 104400 001201          TYPE      $CRLF       ;ANOTHER CR-LF
1025 002166 013737 001216 001222          MOV      PORTA,PORTC  ;GENERATE ADDRESS OF DRIVE NOT TESTED
1026 002174 062737 000006 001222          ADD      #6,PORTC     ;COMPLEMENT SOME BITS
1027 002202 042737 177770 001222          BIC      #1C7,PORTC  ;SAVE ONLY LOWER BITS
1028 002210 013701 001216          MOV      PORTA,R1     ;USE PORT A ADDRESS AS INDEX
1029 002214 116137 027726 001224          MOVB     ATABIT(R1),ASR1 ;GET ATTENTION BIT FOR DRIVE
1030 002222 005037 001250          CLR      TIMEA        ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
1031 002226 005037 001252          CLR      TIMEAP       ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
1032 002232 005037 001254          CLR      TIMEB        ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
1033 002236 005037 001256          CLR      TIMEBP       ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
1034 002242 004737 017376          JSR      PC,CKCLK     ;SETUP CLOCK
1035 002246 000137 002262          JMP      EXEC         ;CLOCK HAS BEEN STARTED
1036 002252 104400 022245          TYPE      ,NOCLOCK    ;NO CLOCK ON SYSTEM
1037 002256 000000          HALT                    ;FATAL ERROR
1038 002260 000776          BR          .-2        ;INTERLOCK HALT
1039
1040          ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
1041
1042 002262 000005          EXEC:  RESET          ;CLEAR EVERYTHING
1043 002264 005037 177776          CLR      PS          ;CLEAR THE PROCESSOR STATUS WORD
1044 002270 104400 001201          TYPE      $CRLF       ;CR-LF
1045 002274 013700 001262          MOV      $RPADR,R0    ;RH11 ADDRESS FOR INDEXING
1046 002300 012706 0011C0          MOV      #STACK,R6   ;LOAD STACK POINTER
1047 002304 004737 017376          JSR      PC,CKCLK     ;START THE CLOCK
1048 002310 000240          NOP                    ;RETURN IF NO CLOCK
1049 002312 005037 001260          CLR      KYBCTL       ;CLEAR SINGLE TEST INDICATOR
1050 002316 005037 001100          CLR      $PASS        ;CLEAR THE PASS COUNT
1051 002322 112737 000001 001115          MOVB     #1,$ERMAX    ;SET ERROR MAX TO 1
1052 002330 012737 002330 001106          MOV      #,$SLPADR    ;INITIAL SETTING FOR LOOP ADDRESS
1053 002336 012737 002336 001110          MOV      #,$SLPERR    ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
1054 002344 104400 022314          TYPE      ,TESTNO    ;ASK FOR TEST NUMBER
1055 002350 104416          RDOCT          ;GET THE NUMBER
1056 002352 012601          MOV      (SP)+,R1     ;PUT ENTRY INTO R1
1057 002354 001002          BNE      .+6          ;BR IF NOT ZERO
1058 002356 000137 002560          JMP      TST1        ;ENTER ZERO - PERFORM ALL TESTS
1059 002362 020137 027736          CMP      R1,MAXTN    ;SEE IF NUMBER GREATER THAN MAXIMUM
1060 002366 003403          BLE      1$          ;BR IF LESS OR EQUAL
1061 002370 104400 022334          TYPE      ,BADNO     ;BAD ENTRY
1062 002374 000732          BR          EXEC      ;TRY AGAIN
1063 002376 005301          1$:  DEC      R1      ;DECREMENT ENTRY
1064 002400 006301          ASL      R1          ;SHIFT IT LEFT
1065 002402 016137 027676 002426          MOV      TSTADR(R1),2$ ;GET THE TEST ADDRESS

```

```

1066 002410 005237 001260          INC    KYBCTL          ;SET SINGLE TEST INDICATOR
1067 002414 012737 000001 001104    MOV    #1,$ICNT       ;PRESET ITERATION COUNT
1068 002422 000177 000000          JMP    @2$           ;GO TO THE SELECTED TEST
1069 002426 000000          2$: .WORD 0          ;TEST ADDRESS GOES HERE
1070
1071          ;CHANGE THE RH11 UNIBUS ADDRESS USED BY THE PROGRAM
1072
1073 002410 000005          CHANGE: RESET        ;CLEAR THE SYSTEM
1074 002412 012737 000340 177776    MOV    #340,@#PS     ;LOCK OUT ALL INTERRUPTS
1075 002410 012706 001100          MOV    #STACK,SP    ;LOAD THE STACK POINTER
1076 002444 012737 021750 000034    MOV    #STRAP,@#TRAPVEC ;LOAD TRAP VECTOR
1077 002452 012737 000340 000036    MOV    #340,@#TRAPVEC+2 ;LEVEL 7
1078 002460 104400 022362          TYPE   ADDRIS       ;TYPE OUT WHAT THE PRESENT ADDRESS IS
1079 002464 013746 001262          MOV    $RPADR,-(SP) ;PUT THE ADDRESS ON THE STACK
1080 002470 104402          TYPOC              ;TYPE THE ACTUAL ADDRESS
1081 002472 104400 001201          TYPE   ,$CRLF       ;CR-LF
1082 002476 104400 022442          TYPE   ,NTRH1       ;ASK FOR NEW ADDRESS
1083 002502 104416          RDOCT
1084 002504 005716          TST    (SP)         ;0 OR 'CR' ENTERED ?
1085 002506 001402          BEQ    1$          ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
1086 002510 011637 001262          MOV    (SP),$RPADR  ;NEW RH11 ADDRESS
1087 002514 012737 002536 000004 1$: MOV    #2$,4        ;LOAD TRAP ADDRESS
1088 002522 013700 001262          MOV    $RPADR,RO   ;RH11 ADDRESS
1089 002526 062700 000002          ADD    #2,RO       ;FORM ADDRESS OF RHC
1090 002532 005710          TST    (RO)        ;SEE IF RH11 RESPONDS AT THAT ADDRESS
1091 002534 000405          BR     3$          ;BR, RH11 ALIVE AT PRESENT ADDRESS
1092 002536 104400 022474          2$: TYPE   ,NORESP  ;REPORT NO RESPONSE
1093 002542 010046          MOV    RO,-(SP)    ;SETUP TO CONVERT THE ADDRESS
1094 002544 104402          TYPOC              ;TYPE THE ADDRESS
1095 002546 000730          BR     CHANGE      ;GET ADDRESS AGAIN
1096 002550 000137 001676          3$: JMP    START    ;GO TO THE STARTING ADDRESS
1097
1098          ;*****
1099
1100          .SBTTL *** TESTS ***
1101
1102          ;*****
1103
1104
1105 002554 013700 001262          TST1AA: MOV    $RPADR,RO ;;RESTORE RO AFTER END OF PASS
1106
1107          ;*****
1108          ;*TEST 1 DRIVE ACCESS TEST
1109          ;*
1110          ;*VERIFY THAT THE DRIVE CAN BE ACCESSED THROUGH BOTH PORTS
1111          ;*
1112          ;* A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE
1113          ;* DRIVE IS A DUAL PORT RPO4, THAT THE DRIVE IS ONLINE (RHDS1 HAS
1114          ;* 'MOL' 'PGM' 'DPR' & 'DRY' BITS SET), AND THE THE DRIVE SERIAL
1115          ;* NUMBER READ THROUGH BOTH PORTS IS THE SAME.
1116          ;*
1117          ;* B. THE TEST IS REPEATED THROUGH BOTH PORTS.
1118          ;*
1119          ;*****
1120          TST1:
1121          SCOPE          ;INITIALIZE THE SCOPE HANDLER

```

```

1122 002562 005737 001260      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
1123 002566 001406              BEQ      2$         ;BR IF NOT
1124 002570 100002              BPL      1$         ;BR IF JUST ENTERED TEST
1125 002572 000137 002262      JMP      EXEC       ;RETURN & GET NEXT TEST NUMBER
1126 002576 012737 177777 001260 1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
1127 002604 112737 000001 001102 2$:      MOVB     #1,$STNM   ;TEST NUMBER
1128 002612 012737 002634 001106      MOV      #TEST1,$LPADR ;LOAD LOOP ON TEST ADDRESS
1129 002620 012737 002634 001110      MOV      #TEST1,$LPERR ;LOAD LOOP ON ERROR ADDRESS
1130 002626 012737 000001 001170      MOV      #1,$TIMES   ;DO 1 ITERATION
1131
1132 ;*****
1133 ;END OF 'SCOPE' SETUP - START OF MAIN TEST
1134
1135 002634      TEST1:
1136
1137 ;*****
1138 ;VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B
1139
1140 002634 113760 001216 000010      MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
1141 002642 013737 001216 001226      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1142 002650 005760 000012              TST      RHDS1(RO)    ;SEE IF DRIVE (PORT A) PRESENT
1143 002654 005037 001236              CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
1144 002660 016037 000010 001126      MOV      RHCS2(RO), $BDDAT ;GET CONTENTS OF RHCS2
1145 002666 012737 000010 001122      MOV      #RHCS2,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
1146 002674 060037 001122              ADD      RO,$B0ADR    ;ADD RH11 BASE ADDRESS
1147 002700 005037 001124              CLR      $GDDAT       ;WHAT REGISTER SHOULD BE
1148 002704 013737 001126 001156      MOV      $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
1149 002712 042737 167777 001156      BIC      #1CNED,$TMP0 ;SAVE SPECIFIED BITS
1150 002720 023737 001124 001156      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
1151 002726 001414              BEQ      64$         ;BR IF OK
1152 002730 013737 001126 001166      MOV      $BDDAT,$TMP4  ;COPY 'BAD DATA'
1153 002736 042737 010000 001166      BIC      #NED,$TMP4   ;CLEAR THE MASKED BITS
1154 002744 053737 001166 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
1155 002752 104001              ERROR    1           ;TYPE MESSAGE 1
1156 002754 005137 001236              COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
1157 002760 000240              NOP
1158 002762 005737 001236      64$:      TST      CKERR        ;WAS 'NED' SET ?
1159 002766 001403              BEQ      .+10        ;BR IF NOT
1160 002770 012760 000040 000010      MOV      #CLR,RHCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
1161 002776 113760 001220 000010      MOVB     PORTB,RHCS2(RO) ;SELECT PORT B
1162 003004 013737 001220 001226      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1163 003012 005760 000012              TST      RHDS1(RO)    ;SEE IF DRIVE (PORT B) PRESENT
1164 003016 005037 001236              CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
1165 003022 016037 000010 001126      MOV      RHCS2(RO), $BDDAT ;GET CONTENTS OF RHCS2
1166 003030 012737 000010 001122      MOV      #RHCS2,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
1167 003036 060037 001122              ADD      RO,$B0ADR    ;ADD RH11 BASE ADDRESS
1168 003042 005037 001124              CLR      $GDDAT       ;WHAT REGISTER SHOULD BE
1169 003046 013737 001126 001156      MOV      $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
1170 003054 042737 167777 001156      BIC      #1CNED,$TMP0 ;SAVE SPECIFIED BITS
1171 003062 023737 001124 001156      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
1172 003070 001414              BEQ      65$         ;BR IF OK
1173 003072 013737 001126 001166      MOV      $BDDAT,$TMP4  ;COPY 'BAD DATA'
1174 003100 042737 010000 001166      BIC      #NED,$TMP4   ;CLEAR THE MASKED BITS
1175 003106 053737 001166 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
1176 003114 104001              ERROR    1           ;TYPE MESSAGE 1
1177 003116 005137 001236              COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR

```

```

1178 003122 000240          65$:  NOP
1179 003124 005737 001236    TST    CKERR          ;WAS 'NED' SET ?
1180 003130 001403          BEQ    .+10           ;BR IF NOT
1181 003132 012760 000040 000010  MOV    #CLR,RHCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
1182
1183 ;:*****
1184 ;CONFIRM THAT DRIVE IS AN RPO4 AND IS DUAL PORT
1185
1186 003140 113760 001216 000010  MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
1187 003146 013737 001216 001226  MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1188 003154 005037 001236          CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
1189 003160 016037 000026 001126  MOV    RHDT(RO), $BDDAT ;GET CONTENTS OF RHDT
1190 003166 012737 000026 001122  MOV    #RHDT,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
1191 003174 060037 001122          ADD    RO,$B0ADR      ;ADD RH11 BASE ADDRESS
1192 003200 012737 024020 001124  MOV    #24020,$GDDAT ;WHAT REGISTER SHOULD BE
1193 003206 023737 001124 001126  CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
1194 003214 001403          BEQ    66$           ;BR IF OK
1195 003216 104002          ERROR  2            ;TYPE MESSAGE 2
1196 003220 005137 001236          COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
1197 003224 000240          66$:  NOP
1198 003226 113760 001220 000010  MOVB   PORTB,RHCS2(RO) ;SELECT PORT B
1199 003234 013737 001220 001226  MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1200 003242 005037 001236          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
1201 003246 016037 000026 001126  MOV    RHDT(RO), $BDDAT ;GET CONTENTS OF RHDT
1202 003254 012737 000026 001122  MOV    #RHDT,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
1203 003262 060037 001122          ADD    RO,$B0ADR      ;ADD RH11 BASE ADDRESS
1204 003266 012737 024020 001124  MOV    #24020,$GDDAT ;WHAT REGISTER SHOULD BE
1205 003274 023737 001124 001126  CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
1206 003302 001403          BEQ    67$           ;BR IF OK
1207 003304 104002          ERROR  2            ;TYPE MESSAGE 2
1208 003306 005137 001236          COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
1209 003312 000240          67$:  NOP
1210
1211 ;:*****
1212 ;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
1213
1214 003314 113760 001216 000010  MOVB   PORTA,RHCS2(RO) ;SELECT PORT A
1215 003322 013737 001216 001226  MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1216 003330 005037 001236          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
1217 003334 016037 000012 001126  MOV    RHDS1(RO), $GDDAT ;GET CONTENTS OF RHDS1
1218 003342 012737 000012 001122  MOV    #RHDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
1219 003350 060037 001122          ADD    RO,$B0ADR      ;ADD RH11 BASE ADDRESS
1220 003354 012737 001000 001124  MOV    #PGM,$GDDAT ;WHAT REGISTER SHOULD BE
1221 003362 013737 001126 001156  MOV    $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
1222 003370 042737 176777 001156  BIC    #1CPGM,$TMP0 ;SAVE SPECIFIED BITS
1223 003376 023737 001124 001156  CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
1224 003404 001414          BEQ    68$           ;BR IF OK
1225 003406 013737 001126 001166  MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
1226 003414 042737 001000 001166  BIC    #PGM,$TMP4 ;CLEAR THE MASKED BITS
1227 003422 053737 001166 001124  BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
1228 003430 104003          ERROR  3            ;TYPE MESSAGE 3
1229 003432 005137 001236          COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
1230 003436 000240          68$:  NOP
1231 003440 005037 001236          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
1232 003444 016037 000012 001126  MOV    RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
1233 003452 012737 000012 001122  MOV    #RHDS1,$B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE

```

```

1234 003460 060037 001122
1235 003464 012737 010600 001124
1236 003472 013737 001126 001156
1237 003500 042737 167177 001156
1238 003506 023737 001124 001156
1239 003514 001414
1240 003516 013737 001126 001166
1241 003524 042737 010600 001166
1242 003532 053737 001166 001124
1243 003540 104004
1244 003542 005137 001236
1245 003546 000240
1246 003550 113760 001220 000010
1247 003556 013737 001220 001226
1248 003564 005037 001236
1249 003570 016037 000012 001126
1250 003576 012737 000012 001122
1251 003604 060037 001122
1252 003610 012737 001000 001124
1253 003616 013737 001126 001156
1254 003624 042737 176777 001156
1255 003632 023737 001124 001156
1256 003640 001414
1257 003642 013737 001126 001166
1258 003650 042737 001000 001166
1259 003656 053737 001166 001124
1260 003664 104003
1261 003666 005137 001236
1262 003672 000240
1263 003674 005037 001236
1264 003700 016037 000012 001126
1265 003706 012737 000012 001122
1266 003714 060037 001122
1267 003720 012737 010600 001124
1268 003726 013737 001126 001156
1269 003734 042737 167177 001156
1270 003742 023737 001124 001156
1271 003750 001414
1272 003752 013737 001126 001166
1273 003760 042737 010600 001166
1274 003766 053737 001166 001124
1275 003774 104004
1276 003776 005137 001236
1277 004002 000240
1278
1279
1280
1281
1282 004004 113760 001216 000010
1283 004012 016037 000030 001124
1284 004020 113760 001220 000010
1285 004026 016037 000030 001126
1286 00404 023737 001124 001126
1287 004042 001406
1288 004044 104005
1289 004046 032737 100000 177570

```

```

ADD R0,$BDADR ;ADD RH11 BASE ADDRESS
MOV #MOL!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1C10600,$STMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$STMP0 ;COMPARE THE BITS
BEQ 69$ ;BR IF OK
MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
BIC #10600,$STMP4 ;CLEAR THE MASKED BITS
BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 4 ;TYPE MESSAGE 4
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
69$:
MOV PORTB,RHCS2(R0) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHDS1(R0),$BDDAT ;GET CONTENTS OF RHDS1
MOV #RHDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0,$BDADR ;ADD RH11 BASE ADDRESS
MOV #PGM,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1CPGM,$STMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$STMP0 ;COMPARE THE BITS
BEQ 70$ ;BR IF OK
MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
BIC #PGM,$STMP4 ;CLEAR THE MASKED BITS
BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 3 ;TYPE MESSAGE 3
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
70$:
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHDS1(R0),$BDDAT ;GET CONTENTS OF RHDS1
MOV #RHDS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0,$BDADR ;ADD RH11 BASE ADDRESS
MOV #MOL!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
BIC #1C10600,$STMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$STMP0 ;COMPARE THE BITS
BEQ 71$ ;BR IF OK
MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
BIC #10600,$STMP4 ;CLEAR THE MASKED BITS
BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 4 ;TYPE MESSAGE 4
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP
71$:
;*****
;VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME
MOV PORTA,RHCS2(R0) ;SELECT PORT A
MOV RSHN(R0),$GDDAT ;STORE THE PORT A SERIAL NUMBER
MOV PORTB,RHCS2(R0) ;SELECT PORT B
MOV RSHN(R0),$BDDAT ;STORE THE PORT B SERIAL NUMBER
CMP $GDDAT,$BDDAT ;ARE THEY THE SAME?
BEQ 1$ ;BR IF THEY ARE
ERROR 5 ;REPORT THE ERROR
BIT #SW15,$SWR ;HALT ON ERROR?

```



```

1290 004054 001001          BNE      IS          ;BR IF SET - PROGRAM HAS ALREADY HALTED
1291 004056 000000          HALT          ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
1292 004060                IS:
1293
1294                ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
1295
1296 004060 105737 001103      TSTB     SERFLG      ;DID AN ERROR OCCUR ?
1297 004064 001412          BEQ      TST2        ;:BR IF NOT
1298 004066 032737 001000 177570 BIT      #SW09,SWR    ;SEE IF LOOP ON ERROR SET (SWR9=1)
1299 004074 001406          BEQ      TST2        ;:BR IF NOT
1300 004076 105037 001103      CLRB     SERFLG      ;CLEAR THE ERROR FLAG
1301 004102 005037 001170      CLR      $TIMES      ;CLEAR THE MAX ITERATION COUNT
1302 004106 000177 174776      JMP      $SLPERR     ;GO TO THE LOOP ADDRESS
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320

```

```

*****
*TEST 2      SET 'VV' FOR PORT A
*
*SET VOLUME VALID
*
* A.  ISSUE A DRIVE CLEAR COMMAND THROUGH PORT A.
*
* B.  ISSUE A READIN PRESET COMMAND THROUGH PORT A.  VERIFY
*      THAT THE 'VV' BIT IS SET FOR PORT A.
*
* C.  ISSUE A RELEASE COMMAND THROUGH PORT A.  VERIFY THAT
*      THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
*      BIT IS SET.
*****

```

```

1321 004112                TST2:
1322 004112 000004          SCOPE          ;INITIALIZE THE SCOPE HANDLER
1323 004114 005737 001260      TST      KYBCTL      ;PERFORMING ONLY SINGLE TESTS ?
1324 004120 001406          BEQ      2$          ;BR IF NOT
1325 004122 100002          BPL      IS          ;BR IF JUST ENTERED TEST
1326 004124 000137 002262      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
1327 004130 012737 177777 001260 1$: MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
1328 004136 112737 000002 001102 2$: MOVB     #2,$ST$NM  ;TEST NUMBER
1329 004144 012737 004166 001106      MOV      #TEST1,$LPADR ;LOAD LOOP ON TEST ADDRESS
1330 004152 012737 004166 001110      MOV      #TEST2,$LPERR ;LOAD LOOP ON ERROR ADDRESS
1331 004160 012737 000001 001170      MOV      #1,$TIMES   ;DO 1 ITERATION
1332
1333
1334
1335
1336

```

```

*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
*****

```

```

1337 004166                TEST2:
1338 004166 113760 001216 000010      MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
1339 004174 013737 001216 001226      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1340
1341
1342                ;*****
1343                ;SET VOLUME VALUE FOR PORT
1344 004202 012760 000011 000000      MOV      #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR
1345 004210 012760 000021 000000      MOV      #21,RHCS1(RO) ;ISSUE A READIN PRESET

```

```

1346 004216 012760 010000 000032      MOV      #FMT22,RHOF(RO) ;SET FMT22
1347
1348                                     ;:*****
1349                                     ;:VERIFY THAT THE DRIVE STATUS IS CORRECT
1350
1351 004224 005037 001236      CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
1352 004230 016037 000012 001126      MOV      RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
1353 004236 012737 000012 001122      MOV      #RHDS1,$BDAOR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
1354 004244 060037 001122      ADD      RO,$BDAOR      ;ADD RH11 BASE ADDRESS
1355 004250 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
1356 004256 013737 001126 001156      MOV      $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
1357 004264 042737 106077 001156      BIC      #1C71700,$TMP0  ;SAVE SPECIFIED BITS
1358 004272 023737 001124 001156      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
1359 004300 001414      BEQ      64$           ;BR IF OK
1360 004302 013737 001126 001166      MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
1361 004310 042737 071700 001166      BIC      #71700,$TMP4   ;CLEAR THE MASKED BITS
1362 004316 053737 001166 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
1363 004324 104010      ERROR   10           ;TYPE MESSAGE 10
1364 004326 005137 001236      COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
1365 004332 000240      64$: NOP
1366
1367                                     ;:*****
1368                                     ;:RELEASE THE DRIVE FROM PORT A
1369
1370
1371 004334 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A
1372 004342 013737 001216 001226      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1373 004350 012760 000013 000000      MOV      #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT A
1374
1375                                     ;:VERIFY THAT THE DRIVE IS IN NEUTRAL
1376
1377 004356 005037 001242      CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
1378 004362 012737 000012 001122      MOV      #RHDS1,$BDAOR  ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
1379 004370 060037 001122      ADD      RO,$BDAOR     ;ADD THE I/O BASE ADDRESS
1380 004374 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
1381 004402 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
1382 004410 016037 000012 001162      MOV      RHDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
1383 004416 013737 001162 001156      MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
1384 004424 042737 100100 001156      BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
1385 004432 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
1386 004440 016037 000012 001164      MOV      RHDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
1387 004446 013737 001164 001160      MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
1388 004454 042737 100100 001160      BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
1389 004462 023737 001156 001160      CMP      $TMP2,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
1390 004470 001006      BNE      65$         ;BR IF NOT
1391 004472 005737 001156      TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
1392 004476 001037      BNE      67$         ;BR IF NOT
1393 004500 104034      ERROR   34          ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
1394 004502 000137 004666      JMP      69$         ;BYPASS THE REST OF THE CHECKS
1395 004506 013737 001162 001126      65$: MOV      $TMP2,$BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
1396 004514 013737 001220 001226      MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1397 004522 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B.
1398 004530 005737 001156      TST      $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
1399 004534 001414      BEQ      66$         ;BR IF ZERO
1400 004536 013737 001216 001226      MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1401 004544 013737 001164 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT

```

E03

```

1402 004552 113760 001216 000010      MOVB   PORTA,RHCS2(R0) ;SELECT PORT A.
1403 004560 005737 001160              TST    $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
1404 004564 001004              BNE    67$            ;BR IF NOT
1405 004566 012737 177777 001242 66$:  MOV    #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
1406 004574 104036              ERROR  36            ;TYPE ERROR MESSAGE 36
1407 004576 013737 001162 001126 67$:  MOV    $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
1408 004604 013737 001216 001226              MOV    PORTA,PTNBR    ;CHANGE PORT NUMBER
1409 004612 042737 100100 001162              BIC    #ATA!VV,$TMP2  ;DON'T CHECK ATTN BIT OR VV BIT
1410 004620 023737 001124 001162              CMP    $GDDAT,$TMP2  ;ALL BITS OK ?
1411 004626 001401              BEQ    68$            ;BR IF OK FROM PORT A.
1412 004630 104037              ERROR  37            ;REPORT ERROR
1413 004632 013737 001164 001126 68$:  MOV    $TMP3,$BDDAT   ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
1414 004640 013737 001220 001226              MOV    PORTB,PTNBR    ;CHANGE PORT NUMBER
1415 004646 042737 100100 001164              BIC    #ATA!VV,$TMP3  ;DON'T CHECK ATTN BIT OR VV BIT
1416 004654 023737 001124 001164              CMP    $GDDAT,$TMP3  ;SEE IF READ OK FROM PORT B.
1417 004662 001401              BEQ    69$            ;BR IF OK
1418 004664 104037              ERROR  37            ;REPORT THE ERROR
1419 004666 000240 69$:  NOP
1420
1421                                     ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
1422
1423 004670 105737 001103              TSTB   $ERFLG         ;DID AN ERROR OCCUR ?
1424 004674 001412              BEQ    TST3           ;BR IF NOT
1425 004676 032737 001000 177570              BIT    #SW09,$SWR     ;SEE IF LOOP ON ERROR SET (SWR9=1)
1426 004704 001406              BEQ    TST3           ;BR IF NOT
1427 004706 105037 001103              CLRB   $ERFLG         ;CLEAR THE ERROR FLAG
1428 004712 005037 001170              CLR    $TIMES         ;CLEAR THE MAX ITERATION COUNT
1429 004716 000177 174166              JMP    $SLPERR        ;GO TO THE LOOP ADDRESS
1430
1431                                     ;*****
1432                                     ;*TEST 3      SET 'VV' FOR PORT B
1433                                     ;*
1434                                     ;*SET VOLUME VALID
1435                                     ;*
1436                                     ;*  A.  ISSUE A DRIVE CLEAR COMMAND THROUGH PORT B.
1437                                     ;*
1438                                     ;*  B.  ISSUE A READIN PRESET COMMAND THROUGH PORT B.  VERIFY
1439                                     ;*      THAT THE 'VV' BIT IS SET FOR PORT B.
1440                                     ;*
1441                                     ;*  C.  ISSUE A RELEASE COMMAND THROUGH PORT B.  VERIFY THAT
1442                                     ;*      THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
1443                                     ;*      BIT IS SET.
1444                                     ;*
1445                                     ;*****
1446                                     ;*TST3:
1447 004722 000004              SCOPE  KYBCTL         ;INITIALIZE THE SCOPE HANDLER
1448 004724 005737 001260              TST    KYBCTL         ;PERFORMING ONLY SINGLE TESTS ?
1449 004730 001406              BEQ    2$            ;BR IF NOT
1450 004732 100002              BPL    1$            ;BR IF JUST ENTERED TEST
1451 004734 000137 002262              JMP    EXEC          ;RETURN & GET NEXT TEST NUMBER
1452 004740 012737 177777 001260 1$:  MOV    #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
1453 004746 112737 000003 001102 2$:  MOVB   #3,$STNM      ;TEST NUMBER
1454 004754 012737 004776 001106              MOV    #TEST3,$LPAOR ;LOAD LOOP ON TEST ADDRESS
1455 004762 012737 004776 001110              MOV    #TEST3,$LPERR ;LOAD LOOP ON ERROR ADDRESS
1456 004770 012737 000001 001170              MOV    #1,$TIMES     ;DO 1 ITERATION
1457

```

```

1458
1459
1460
1461
1462 004776
1463 004776 113760 001220 000010
1464 005004 013737 001220 001226
1465
1466
1467
1468
1469 005012 012760 000011 000000
1470 005020 012760 000021 000000
1471 005026 012760 010000 000032
1472
1473
1474
1475
1476 005034 005037 001236
1477 005040 016037 000012 001126
1478 005046 012737 000012 001122
1479 005054 060037 001122
1480 005060 012737 011700 001124
1481 005066 013737 001126 001156
1482 005074 042737 106077 001156
1483 005102 023737 001124 001156
1484 005110 001414
1485 005112 013737 001126 001166
1486 005120 042737 071700 001166
1487 005126 053737 001166 001124
1488 005134 104010
1489 005136 005137 001236
1490 005142 000240
1491
1492
1493
1494
1495
1496 005144 113760 001220 000010
1497 005152 013737 001220 001226
1498 005160 012760 000013 000000
1499
1500
1501
1502 005166 005037 001242
1503 005172 012737 000012 001122
1504 005200 060037 001122
1505 005204 012737 011600 001124
1506 005212 113760 001216 000010
1507 005220 016037 000012 001162
1508 005226 013737 001162 001156
1509 005234 042737 100100 001156
1510 005242 113760 001220 000010
1511 005250 016037 000012 001164
1512 005256 013737 001164 001160
1513 005264 042737 100100 001160

;*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST

TEST3:
MOV  PORTB,RHCS2(RO) ;SELECT PORT B
MOV  PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;*****
;SET VOLUME VALUE FOR PORT

MOV  #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR
MOV  #21,RHCS1(RO) ;ISSUE A READIN PRESET
MOV  #FMT22,RHOF(RO) ;SET FMT22

;*****
;VERIFY THAT THE DRIVE STATUS IS CORRECT

CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV  RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
MOV  #RHDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD  RO,$BDDADR ;ADD RH11 BASE ADDRESS
MOV  #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
MOV  $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
BIC  #1C71700,$TMP0 ;SAVE SPECIFIED BITS
CMP  $GDDAT,$TMP0 ;COMPARE THE BITS
BEQ  645 ;BR IF OK
MOV  $BDDAT,$TMP4 ;COPY 'BAD DATA'
BIC  #71700,$TMP4 ;CLEAR THE MASKED BITS
BIS  $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
ERROR 10 ;TYPE MESSAGE 10
COM  CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
NOP

645:

;*****
;RELEASE THE DRIVE FROM PORT B

MOV  PORTB,RHCS2(RO) ;SELECT PORT B
MOV  PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV  #13,RHCS1(RO) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS IN NEUTRAL

CLR  RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV  #RHDS1,$BDDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
ADD  RO,$BDDADR ;ADD THE I/O BASE ADDRESS
MOV  #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
MOV  PORTA,RHCS2(RO) ;SELECT PORT A.
MOV  RHDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
MOV  $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
BIC  #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOV  PORTB,RHCS2(RO) ;SELECT PORT B.
MOV  RHDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
MOV  $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
BIC  #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY

```

```

1514 005272 023737 001156 001160      CMP      $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
1515 005300 001006                        BNE      65$              ;BR IF NOT
1516 005302 005737 001156      TST      $TMP0              ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
1517 005306 001037                        BNE      67$              ;BR IF NOT
1518 005310 104034      ERROR    34                ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
1519 005312 000137 005476      JMP      69$              ;BYPASS THE REST OF THE CHECKS
1520 005316 013737 001162 001126 65$:    MOV      $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
1521 005324 013737 001220 001226      MOV      PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1522 005332 113760 001220 000010      MOV      PORTB,RHCS2(RO)  ;SELECT PORT B.
1523 005340 005737 001156      TST      $TMP0              ;SEE IF STATUS EQ 0 FROM PORT A.
1524 005344 001414                        BEQ      66$              ;BR IF ZERO
1525 005346 013737 001216 001226      MOV      PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1526 005354 013737 001164 001126      MOV      $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
1527 005362 113760 001216 000010      MOV      PORTA,RHCS2(RO)  ;SELECT PORT A.
1528 005370 005737 001160      TST      $TMP1              ;SEE IF STATUS EQ ZERO FROM PORT B.
1529 005374 001004                        BNE      67$              ;BR IF NOT
1530 005376 012737 177777 001242 66$:    MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
1531 005404 104036      ERROR    36                ;TYPE ERROR MESSAGE 36
1532 005406 013737 001162 001126 67$:    MOV      $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
1533 005414 013737 001216 001226      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
1534 005422 042737 100100 001162      BIC      #ATA!VV,$TMP2     ;DON'T CHECK ATTN BIT OR VV BIT
1535 005430 023737 001124 001162      CMP      $GDDAT,$TMP2     ;ALL BITS OK ?
1536 005436 001401                        BEQ      68$              ;BR IF OK FROM PORT A.
1537 005440 104037      ERROR    37                ;REPORT ERROR
1538 005442 013737 001164 001126 68$:    MOV      $TMP3,$BDDAT      ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
1539 005450 013737 001220 001226      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
1540 005456 042737 100100 001164      BIC      #ATA!VV,$TMP3     ;DON'T CHECK ATTN BIT OR VV BIT
1541 005464 023737 001124 001164      CMP      $GDDAT,$TMP3     ;SEE IF READ OK FROM PORT B.
1542 005472 001401                        BEQ      69$              ;BR IF OK
1543 005474 104037      ERROR    37                ;REPORT THE ERROR
1544 005476 000240 69$:    NOP
1545
1546                                ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
1547
1548 005500 105737 001103      TSTB     $ERFLG            ;DID AN ERROR OCCUR ?
1549 005504 001412                        BEQ      TST4              ;:BR IF NOT
1550 005506 032737 001000 177570      BIT      #SW09,SWR        ;SEE IF LOOP ON ERROR SET (SWR9=1)
1551 005514 001406                        BEQ      TST4              ;:BR IF NOT
1552 005516 105037 001103      CLRB     $ERFLG            ;CLEAR THE ERROR FLAG
1553 005522 005037 001170      CLR      $TIMES           ;CLEAR THE MAX ITERATION COUNT
1554 005526 000177 173356      JMP      $SLPERR          ;GO TO THE LOOP ADDRESS
1555
1556
1557
1558
1559                                ;*****
1560                                ;*TEST 4      MEASURE THE TIMEOUT ONE-SHOT THROUGH PORT A
1561                                ;*
1562                                ;*MEASURE THE TIMEOUT ONE-SHOT VALUE THROUGH PORT A
1563                                ;*
1564                                ;* A.  WRITE 0'S INTO RHDS1 THROUGH PORT A AND VERIFY THAT THE
1565                                ;*      DRIVE HAS BEEN SEIZED.
1566                                ;*
1567                                ;* B.  WAIT FOR TIMEOUT TO OCCUR.  MEASURE THE DURATION OF THE TIMEOUT
1568                                ;*      ONE-SHOT AND SAVE THE VALUE FOR LATER USE.
1569                                ;*

```

```

1570      ;* C. VERIFY THAT THE TIMEOUT OCCURED AND THAT THE DRIVE RETURNS
1571      ;* TO NEUTRAL
1572      ;*
1573      ;*****
1574 005532          TST4:
1575 005532 000004          SCOPE          ;INITIALIZE THE SCOPE HANDLER
1576 005534 005737 001260  TST          KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
1577 005540 001406          BEQ          2$          ;BR IF NOT
1578 005542 100002          BPL          1$          ;BR IF JUST ENTERED TEST
1579 005544 000137 002262  JMP          EXEC          ;RETURN & GET NEXT TEST NUMBER
1580 005550 012737 177777 001260 1$: MOV          #-1,KYBCTL          ;SET SINGLE TEST INDICATOR
1581 005556 112737 000004 001102 2$: MOV          #4,$TSTNM          ;TEST NUMBER
1582 005564 012737 005606 001106  MOV          #TEST4,$LPADR          ;LOAD LOOP ON TEST ADDRESS
1583 005572 012737 005606 001110  MOV          #TEST4,$LPERR          ;LOAD LOOP ON ERROR ADDRESS
1584 005600 012737 000001 001170  MOV          #1,$TIMES          ;DO 1 ITERATION
1585
1586
1587      ;*****
1588      ;END OF 'SCOPE' SETUP - START OF MAIN TEST
1589
1590 TEST4:
1591 005606 005037 001250          CLR          TIMEA          ;CLEAR THE TIMEOUT VALUE STORAGE LOCATION
1592 005612 005037 001252          CLR          TIMEAP          ;CLEAR THE + 25% TOLERANCE LOCATION
1593
1594      ;*****
1595      ;START THE TIMER
1596
1597 005616 005037 001244          CLR          TIME          ;CLEAR THE ELAPSED TIME COUNTER
1598 005622 012737 003720 001246  MOV          #2000.,WATCH          ;SET WATCH TO 2000 MS
1599
1600      ;*****
1601
1602      ;SEIZE THE DRIVE THROUGH PORT A
1603
1604 005630 113760 001216 000010  MOV          PORTA,RHCS2(RO) ;SELECT PORT A
1605 005636 013737 001216 001230  MOV          PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
1606 005644 005060 000012          CLR          RHDS1(RO) ;WRITE RHDS1
1607 005650 113750 001220 000010  MOV          PORTB,RHCS2(RO) ;SELECT PORT B
1608 005656 013737 001220 001226  MOV          PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1609 005664 013737 001220 001232  MOV          PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
1610 005672 016037 000012 001126  MOV          RHDS1(RO),SBDDAT ;SEE IF DRIVE SEIZED BY PORT A
1611 005700 010037 001122          MOV          RO,$BDAOR ;RH11 BASE ADDRESS
1612 005704 062737 000012 001122  ADD          #RHDS1,$BDAOR ;GENERATE BAD REGISTER ADDRESS
1613 005712 005037 001124          CLR          $GDDAT ;REGISTER SHOULD BE ZERO
1614 005716 023737 001124 001126  CMP          $GDDAT,$BDDAT ;IS THE REGISTER ZERO
1615 005724 001403          BEQ          .+10 ;BR IF IT IS
1616 005726 104030          ERROR          30 ;REPORT THE ERROR
1617 005730 000137 006442          JMP          4$ ;BYPASS REST OF THE SUBTEST
1618 005734 113760 001216 000010  MOV          PORTA,RHCS2(RO) ;SELECT PORT A
1619 005742 013737 001216 001226  MOV          PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1620 005750 016037 000012 001126  MOV          RHDS1(RO),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
1621 005756 012737 011700 001124  MOV          #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
1622 005764 013737 001124 001160  MOV          $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
1623 005772 005137 001160          COM          $TMP1 ;COMPLEMENT THE EXPECTED STATUS
1624 005776 013737 001126 001156  MOV          $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
1625 006004 043737 001160 001156  BIC          $TMP1,$TMP0 ;CLEAR UNWANTED BITS

```

```

1626 006012 023737 001124 001156      CMP      $GDDAT,$STMP0      ;ARE THE EXPECTED STATUS BITS SET ?
1627 006020 001401                      BEQ      .+4                ;BR IF THEY ARE
1628 006022 104031                      ERROR   31                  ;REPORT THE ERROR
1629
1630                                     ;:*****
1631                                     ;WAIT FOR PORT A TO TIMEOUT
1632
1633 006024 113760 001220 000010      MOV      PORTB,RHCS2(RO)    ;SELECT PORT B
1634 006032 013737 001220 001226      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1635 006040 005760 000012      1$:    TST      RHDS1(RO)      ;WAIT FOR THE DRIVE TO TIMEOUT
1636 006044 001006                      BNE     2$                ;BR WHEN TIMEOUT OCCURS
1637 006046 005737 001246      TST      WATCH           ;CHECK WATCH
1638 006052 001372                      BNE     1$                ;BR IF NOT ZERO
1639 006054 104006                      ERROR   6                  ;NO TIMEOUT WITHIN 2 SECONDS
1640 006056 000137 006114                      JMP     3$                ;BYPASS THE REST OF THE TEST
1641 006062 013737 001244 001250 2$:    MOV      TIME,TIMEA       ;SAVE THE ELAPSED TIME FOR PORT A
1642 006070 004537 017562      JSR      RS,TOLER        ;CALCULATE THE TOLERANCE
1643 006074 001250                      .WORD   TIMEA             ;TIMEOUT VALUE FOR PORT A
1644 006076 012637 001252      MOV      (SP)+,TIMEAP     ;+25% TOLERANCE
1645
1646                                     ;:*****
1647                                     ;VERIFY THAT THE TIMEOUT ONE-SHOT VALUE IS AT LEAST 500 MS
1648
1649 006102 023727 001250 000764      CMP      TIMEA,#500.      ;IS TIMEOUT VALUE AT LEAST 500 MS ?
1650 006110 103001                      BHS     3$                ;BR IF IT IS
1651 006112 104007                      ERROR   7                  ;TIMEOUT LESS THAN 500 MS
1652
1653                                     ;:*****
1654                                     ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT A TIMED OUT
1655
1656 006114      3$:
1657
1658                                     ;VERIFY THAT THE DRIVE IS IN NEUTRAL
1659
1660 006114 005037 001242                      CLR     RELERR            ;CLEAR THE 'RELEASE ERROR' INDICATOR
1661 006120 012737 000012 001122      MOV     #RHDS1,$BDADR     ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
1662 006126 060037 001122                      ADD     RO,$BDADR        ;ADD THE I/O BASE ADDRESS
1663 006132 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
1664 006140 113760 001216 000010      MOV     PORTA,RHCS2(RO)   ;SELECT PORT A.
1665 006146 016037 000012 001162      MOV     RHCS1(RO),$STMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
1666 006154 013737 001162 001156      MOV     $STMP2,$STMP0     ;COPY IT INTO '$STMP0'
1667 006162 042737 100100 001156      BIC     #ATA!VV,$STMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
1668 006170 113760 001220 000010      MOV     PORTB,RHCS2(RO)   ;SELECT PORT B.
1669 006176 016037 000012 001164      MOV     RHDS1(RO),$STMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
1670 006204 013737 001164 001160      MOV     $STMP3,$STMP1     ;COPY IT INTO '$STMP1'
1671 006212 042737 100100 001160      BIC     #ATA!VV,$STMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
1672 006220 023737 001156 001160      CMP     $STMP0,$STMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
1673 006226 001006                      BNE     64$              ;BR IF NOT
1674 006230 005737 001156                      TST     $STMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
1675 006234 001045                      BNE     66$              ;BR IF NOT
1676 006236 104034                      ERROR   34               ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
1677 006240 000137 006440                      JMP     68$              ;BYPASS THE REST OF THE CHECKS
1678 006244 013737 001162 001126 64$:    MOV     $STMP2,$BDADR     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
1679 006252 013737 001220 001226      MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1680 006260 113760 001220 000010      MOV     PORTB,RHCS2(RO)   ;SELECT PORT B.
1681 006266 005737 001156                      TST     $STMP0           ;SEE IF STATUS EQ 0 FROM PORT A.

```



```

1682 006272 001414      BEQ      65$      ;BR IF ZERO
1683 006274 013737 001216 001226      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1684 006302 013737 001164 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
1685 006310 113760 001216 000010      MOV      PORTA,RHCS2(RO) ;SELECT PORT A.
1686 006316 005737 001160      TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
1687 006322 001012      BNE      66$      ;BR IF NOT
1688 006324 012737 177777 001242 65$:      MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
1689 006332 012760 000011 000000      MOV      #11,RHCS1(RO) ;CLEAR THE DRIVE
1690 006340 012760 000013 000000      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
1691 006346 104035      ERROR    35      ;TYPE ERROR MESSAGE 35
1692 006350 013737 001162 001126 66$:      MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
1693 006356 013737 001216 001226      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
1694 006364 042737 100000 001162      BIC      #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
1695 006372 023737 001124 001162      CMP      $GDDAT,$TMP2 ;ALL BITS OK ?
1696 006400 001401      BEQ      67$      ;BR IF OK FROM PORT A.
1697 006402 104037      ERROR    37      ;REPORT ERROR
1698 006404 013737 001164 001126 67$:      MOV      $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
1699 006412 013737 001220 001226      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
1700 006420 042737 100000 001164      BIC      #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
1701 006426 023737 001124 001164      CMP      $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
1702 006434 001401      BEQ      68$      ;BR IF OK
1703 006436 104037      ERROR    37      ;REPORT THE ERROR
1704 006440 000240 68$:      NOP
1705 006442 4$:

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

1709 006442 105737 001103      TSTB     $ERFLG     ;DID AN ERROR OCCUR ?
1710 006446 001412      BEQ      TST5      ;BR IF NOT
1711 006450 032737 001000 177570      BIT      #SW09,SWR  ;SEE IF LOOP ON ERROR SET (SWR9=1)
1712 006456 001406      BEQ      TST5      ;BR IF NOT
1713 006460 105037 001103      CLRB     $ERFLG     ;CLEAR THE ERROR FLAG
1714 006464 005037 001170      CLR      $TIMES     ;CLEAR THE MAX ITERATION COUNT
1715 006470 000177 172414      JMP      $SLPERR    ;GO TO THE LOOP ADDRESS

```

```

*****
*TEST 5      MEASURE THE TIMEOUT ONE-SHOT THROUGH PORT B
*
*MEASURE THE TIMEOUT ONE-SHOT VALUE THROUGH PORT B
*
* A.  WRITE 0'S INTO RHDS1 THROUGH PORT B AND VERIFY THAT THE
*      DRIVE HAS BEEN SEIZED.
*
* B.  WAIT FOR TIMEOUT TO OCCUR.  MEASURE THE DURATION OF THE TIMEOUT
*      ONE-SHOT AND SAVE THE VALUE FOR LATER USE.
*
* C.  VERIFY THAT THE TIMEOUT OCCURED AND THAT THE DRIVE RETURNS
*      TO NEUTRAL
*
*****

```

```

1733 006474      TST5:      SCOPE
1734 006474 000004      TST      KYBCTL    ;INITIALIZE THE SCOPE HANDLER
1735 006476 005737 001260      BEQ      2$      ;PERFORMING ONLY SINGLE TESTS ?
1736 006502 001406      BPL      1$      ;BR IF NOT
1737 006504 100002      BPL      1$      ;BR IF JUST ENTERED TEST

```


K03

```

1738 006506 000137 002262          JMP      EXEC          ;RETURN & GET NEXT TEST NUMBER
1739 006512 012737 177777 001260 15:  MOV      #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
1740 006520 112737 000C05 001102 25:  MOVB     #5,$STSTM    ;TEST NUMBER
1741 006526 012737 006550 001106      MOV      #TEST5,$LPADR ;LOAD LOOP ON TEST ADDRESS
1742 006534 012737 0C6550 001110      MOV      #TEST5,$LPERR ;LOAD LOOP ON ERROR ADDRESS
1743 006542 012737 000001 001170      MOV      #1,$TIMES    ;DO 1 ITERATION
1744
1745
1746 ;:*****
1747 ;END OF 'SCOPE' SETUP - START OF MAIN TEST
1748
1749 TESTS:
1750 006550 0C5037 001254          CLR      TIMEB        ;CLEAR THE TIMEOUT VALUE STORAGE LOCATION
1751 006554 005037 001256          CLR      TIMEBP       ;CLEAR THE + 25% TOLERANCE LOCATION
1752
1753 ;:*****
1754 ;START THE TIMER
1755
1756 006560 005037 001244          CLR      TIME         ;CLEAR THE ELAPSED TIME COUNTER
1757 006564 012737 003720 001246      MOV      #2000.,WATCH ;SET WATCH TO 2000 MS
1758
1759 ;:*****
1760 ;SEIZE THE DRIVE THROUGH PORT B
1761
1762
1763 006572 113760 001220 000010      MOVB     PORTB,RHCS2(RO) ;SELECT PORT B
1764 006600 013737 001220 001230      MOV      PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
1765 006606 005060 000012          CLR      RHDS1(RO)     ;WRITE RHDS1
1766 006612 113760 001216 000010      MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
1767 006620 013737 001216 001226      MOV      PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1768 006626 013737 001216 001232      MOV      PORTA,OPPRT  ;'OPPOSITE' PORT ADDRESS
1769 006634 016037 000012 001126      MOV      RHDS1(RO),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
1770 006642 010037 001122          MOV      RO,$BDADR    ;RH11 BASE ADDRESS
1771 006646 062737 000012 001122      ADD      #RHDS1,$BDADR ;GENERATE BAD REGISTER ADDRESS
1772 006654 005037 001124          CLR      $GDDAT       ;REGISTER SHOULD BE ZERO
1773 006660 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER ZERO
1774 006666 001403          BEQ      .+10         ;BR IF IT IS
1775 006670 104030          ERROR   30           ;REPORT THE ERROR
1776 006672 000137 007404          JMP      45           ;BYPASS REST OF THE SUBTEST
1777 006676 113760 001220 000010      MOVB     PORTB,RHCS2(RO) ;SELECT PORT B
1778 006704 013737 001220 001226      MOV      PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1779 006712 016037 000012 001126      MOV      RHDS1(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
1780 006720 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
1781 006726 013737 001124 001160      MOV      $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
1782 006734 005137 001160          COM      $TMP1        ;COMPLEMENT THE EXPECTED STATUS
1783 006740 013737 001126 001156      MOV      $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
1784 006746 043737 001160 001156      BIC      $TMP1,$TMP0  ;CLEAR UNWANTED BITS
1785 006754 023737 001124 001156      CMP      $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
1786 006762 001401          BEQ      .+4         ;BR IF THEY ARE
1787 006764 104031          ERROR   31           ;REPORT THE ERROR
1788
1789 ;:*****
1790 ;WAIT FOR PORT B TO TIMEOUT
1791
1792 006766 113760 001216 000010      MOVB     PORTA,RHCS2(RO) ;SELECT PORT A
1793 006774 013737 001216 001226      MOV      PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

1794 007002 005760 000012 1$: TST RHDS1(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
1795 007006 001006 BNE 2$ ;BR WHEN TIMEOUT OCCURS
1796 007010 005737 001246 TST WATCH ;CHECK WATCH
1797 007014 001372 BNE 1$ ;BR IF NOT ZERO
1798 007016 104006 ERROR 6 ;NO TIMEOUT WITHIN 2 SECONDS
1799 007020 000137 007056 JMP 3$ ;BYPASS THE REST OF THE TEST
1800 007024 013737 001244 001254 2$: MOV TIME,TIMEB ;SAVE THE ELAPSED TIME FOR PORT B
1801 007032 004537 017562 JSR R5,TOLER ;CALCULATE THE TOLERANCE
1802 007036 001254 .WORD TIMEB ;TIMEOUT VALUE FOR PORT B
1803 007040 012637 001256 MOV (SP)+,TIMEBP ;+25% TOLERANCE
1804
1805 ;:*****
1806 ;VERIFY THAT THE TIMEOUT ONE-SHOT VALUE IS AT LEAST 500 MS
1807
1809 007044 023727 001254 000764 CMP TIMEB,#500. ;IS TIMEOUT VALUE AT LEAST 500 MS ?
1809 007052 103001 BHS 3$ ;BR IF IT IS
1810 007054 104007 ERROR 7 ;TIMEOUT LESS THAN 500 MS
1811
1812 ;:*****
1813 ;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT B TIMED OUT
1814
1815 007056 3$:
1816
1817 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
1818
1819 007056 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
1820 007062 012737 000012 001122 MOV #RHDS1,$BDDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
1821 007070 060037 001122 ADD RO,$BDDADR ;ADD THE I/O BASE ADDRESS
1822 007074 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
1823 007102 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
1824 007110 016037 000012 001162 MOV RHDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
1825 007116 013737 001162 001156 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
1826 007124 042737 100100 001156 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
1827 007132 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
1828 007140 016037 000012 001164 MOV RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
1829 007146 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
1830 007154 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
1831 007162 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
1832 007170 001006 BNE 64$ ;BR IF NOT
1833 007172 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
1834 007176 001045 BNE 66$ ;BR IF NOT
1835 007200 104034 ERROR 34 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
1836 007202 000137 007402 JMP 68$ ;BYPASS THE REST OF THE CHECKS
1837 007206 013737 001162 001126 64$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
1838 007214 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1839 007222 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B.
1840 007230 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
1841 007234 001414 BEQ 65$ ;BR IF ZERO
1842 007236 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
1843 007244 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
1844 007252 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A.
1845 007260 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
1846 007264 001012 BNE 66$ ;BR IF NOT
1847 007266 012737 177777 001242 65$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
1848 007274 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
1849 007302 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE

```

```

1850 007310 104035          ERROR 35          ;TYPE ERROR MESSAGE 35
1851 007312 013737 001162 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
1852 007320 013737 001216 001226      MOV PORTA,PTNBR ;CHANGE PORT NUMBER
1853 007326 042737 100000 001162      BIC #ATA,$TMP2 ;DON'T CHECK THE ATTN BIT
1854 007334 023737 001124 001162      CMP $GDDAT,$TMP2 ;ALL BITS OK ?
1855 007342 001401          BEQ 67$          ;BR IF OK FROM PORT A.
1856 007344 104037          ERROR 37          ;REPORT ERROR
1857 007346 013737 001164 001126 67$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
1858 007354 013737 001220 001226      MOV PORTB,PTNBR ;CHANGE PORT NUMBER
1859 007362 042737 100000 001164      BIC #ATA,$TMP3 ;DON'T CHECK THE ATTN BIT
1860 007370 023737 001124 001164      CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
1861 007376 001401          BEQ 68$          ;BR IF OK
1862 007400 104037          ERROR 37          ;REPORT THE ERROR
1863 007402 000240          68$: NOP
1864 007404          4$:
1865
1866          ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
1867
1868 007404 105737 001103          TSTB $ERFLG      ;DID AN ERROR OCCUR ?
1869 007410 001412          BEQ TST6         ;BR IF NOT
1870 007412 032737 001000 177570      BIT #SW09,SWR   ;SEE IF LOOP ON ERROR SET (SWR9=1)
1871 007420 001406          BEQ TST6         ;BR IF NOT
1872 007422 105037 001103          CLRB $ERFLG     ;CLEAR THE ERROR FLAG
1873 007426 005037 001170          CLR $TIMES      ;CLEAR THE MAX ITERATION COUNT
1874 007432 000177 171452          JMP $SLPERR     ;GO TO THE LOOP ADDRESS

```

1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905

```

*****
*TEST 6      TEST UNLOAD COMMAND THROUGH PORT A
*
*
*VERIFY THAT THE UNLOAD COMMAND FUNCTIONS PROPERLY AND THAT A PORT
*TIMEOUT WILL NOT OCCUR WHILE THE 'GO' BIT IS SET.
*
* A.  ISSUE AN UNLOAD COMMAND THROUGH PORT A; VERIFY THAT THE
*      DRIVE IS SEIZED.
*
* B.  WAIT THE MEASURED TIMEOUT INTERVAL + 25%; VERIFY THAT THE DRIVE
*      DOES NOT TIME OUT.  VERIFY THAT THE 'GO' BIT IS STILL SET AND
*      THAT 'DRY' AND 'PIP' ARE NOT SET.
*
* C.  REQUEST THAT THE OPERATOR PRESS THE 'STANDBY' BUTTON ON THE DRIVE.
*
* D.  WHEN THE DRIVE CYCLES UP, VERIFY THAT THE DRIVE IS STILL SEIZED
*      BY PORT A, THAT THE 'VV' BIT FOR PORT A IS RESET, THAT THE
*      ATTENTION BIT FOR PORT A IS SET, AND THAT THE ATTENTION BIT
*      FOR PORT B IS NOT SET.
*
* E.  WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE.  WHEN THE TIMEOUT
*      OCCURS, VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, THAT THE
*      ATTENTION BIT FOR PORT A IS STILL SET, AND THAT THE ATTENTION
*      BIT FOR PORT B IS NOT SET.
*
* F.  VERIFY THAT THE 'VV' BIT FOR PORT B IS NOT SET.
*
* G.  ISSUE A PACK ACKNOWLEDGE INSTRUCTION THROUGH BOTH PORTS.
*

```

```

1906                                     ;*****
1907 007436                               TST6:
1908 007436 000004                         SCOPE                ;INITIALIZE THE SCOPE HANDLER
1909 007440 005737 001260                 TST      KYBCTL          ;PERFORMING ONLY SINGLE TESTS ?
1910 007444 001406                         BEQ      2$              ;BR IF NOT
1911 007446 100002                         BPL      1$              ;BR IF JUST ENTERED TEST
1912 007450 000137 002262                 JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
1913 007454 012737 177777 001260 1$:    MOV      #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
1914 007462 112737 000006 001102 2$:    MOV      #6,$STSTNM     ;TEST NUMBER
1915 007470 012737 007512 001106         MOV      #TEST6,$LPADR  ;LOAD LOOP ON TEST ADDRESS
1916 007476 012737 007512 001110         MOV      #TEST6,$LPERR ;LOAD LOOP ON ERROR ADDRESS
1917 007504 012737 000001 001170         MOV      #1,$TIMES     ;DO 1 ITERATION
1918
1919
1920                                     ;*****
1921                                     ;END OF 'SCOPE' SETUP - START OF MAIN TEST
1922
1923 007512                               TEST6:
1924
1925                                     ;CLEAR ATTENTION BITS FOR BOTH PORTS
1926
1927 007512 113760 001216 000010           MOV      PORTA,RHCS2(RO) ;SELECT PORT #A
1928 007520 005060 000012                   CLR      RHDS1(RO)       ;SEIZE THE DRIVE
1929 007524 012760 000011 000030           MOV      #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
1930 007532 012760 000013 000000           MOV      #13,RHCS1(RO)  ;RELEASE THE DRIVE
1931 007540 113760 001220 000010           MOV      PORTB,RHCS2(RO) ;SELECT PORT #B
1932 007546 005060 000012                   CLR      RHDS1(RO)       ;SEIZE THE DRIVE THROUGH PORT 'B'
1933 007552 012760 000011 000000           MOV      #11,RHCS1(RO)  ;ISSUE DRIVE CLEAR
1934 007560 012760 000013 000000           MOV      #13,RHCS1(RO)  ;RELEASE THE DRIVE
1935 007566 113760 001216 000010           MOV      PORTA,RHCS2(RO) ;SELECT PORT A
1936 007574 013737 001216 001226           MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1937 007602 013737 001216 001230           MOV      PORTA,SEIZPT   ;SEIZING PORT'S ADDRESS
1938
1939                                     ;*****
1940                                     ;DO AN UNLOAD COMMAND THROUGH PORT A
1941
1942 007610 012760 000003 000000           MOV      #3,RHCS1(RO)   ;ISSUE AN UNLOAD COMMAND THROUGH PORT A
1943 007616 013737 001252 001246           MOV      TIMEAP,WATCH   ;TIMEOUT ONESHOT VALUE + 25%
1944 007624 005737 001246                   1$:    TST      WATCH         ;FINISHED TIMEOUT ?
1945 007630 001375                             BNE     1$              ;BR IF NOT
1946
1947                                     ;*****
1948                                     ;IS THE STATUS OK ?
1949
1950 007632 005037 001236                   CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
1951 007636 016037 000000 001126           MOV      RHCS1(RO), $BDDAT ;GET CONTENTS OF RHCS1
1952 007644 012737 000000 001122           MOV      #RHCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
1953 007652 060037 001122                   ADD      RO,$BDADR      ;ADD RH11 BASE ADDRESS
1954 007656 012737 004001 001124           MOV      #DVA!GO,$GDDAT ;WHAT REGISTER SHOULD BE
1955 007664 013737 001126 001156           MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
1956 007672 042737 173776 001156           BIC      #1C4001,$TMP0  ;SAVE SPECIFIED BITS
1957 007700 023737 001124 001156           CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
1958 007706 001414                             BEQ     64$             ;BR IF OK
1959 007710 013737 001126 001166           MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
1960 007716 042737 004001 001166           BIC      #4001,$TMP4    ;CLEAR THE MASKED BITS
1961 007724 053737 001166 001124           BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
  
```

```

1962 007732 104011          ERROR 11          ;TYPE MESSAGE 11
1963 007734 005137 001236    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
1964 007740 000240          64$:  NOP
1965 007742 005037 001236    CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
1966 007746 016037 000012 001126    MOV      RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
1967 007754 012737 000012 001122    MOV      @RHDS1, $B0ADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
1968 007752 060037 001122    ADD      RO, $B0ADR      ;ADD RH11 BASE ADDRESS
1969 007756 012737 021500 001124    MOV      @PIP!PGM!DPR!VV, $GDDAT ;WHAT REGISTER SHOULD BE
1970 007774 013737 001126 001156    MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO 'TMP0'
1971 010002 042737 000077 001156    BIC      @I177700, $TMP0 ;SAVE SPECIFIED BITS
1972 010010 023737 001124 001156    CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
1973 010016 001414          BEQ      65$           ;BR IF OK
1974 010020 013737 001126 001166    MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
1975 010026 042737 177700 001166    BIC      @177700, $TMP4  ;CLEAR THE MASKED BITS
1976 010034 053737 001166 001124    BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
1977 010042 104012          ERROR 12          ;TYPE MESSAGE 12
1978 010044 005137 001236    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
1979 010050 000240          65$:  NOP
1980 010052 104400 022564    TYPE     STANDBY        ;TYPE THE STANDBY MESSAGE
1981 010056 013746 001216    MOV      PORTA, -(SP)   ;SAVE PORTA FOR TYPEOUT
1982                                ;TYPE THE PORT NUMBER
1983 010062 104410          TYPDS
1984 010064 104400 001201    TYPE     , $CRLF        ;GO TYPE--DECIMAL ASCII WITH SIGN
1985                                ;CR-LF
1986                                ;*****
1987                                ;WAIT FOR 'MOL' TO SET
1988
1989 010070 032760 010000 000012 2$:  BIT      @MCL, RHDS1(RO) ;WAIT FOR MOL TO SET
1990 010076 001774          BEQ      2$           ;LOOP UNTIL 'MOL' SETS
1991 010100 012737 003720 001246    MOV      @2000, WATCH   ;SETUP A 2 SECOND STALL
1992 010106 113760 001220 000010    MOV      PORTB, RHCS2(RO) ;SELECT PORT B
1993 010114 013737 001220 001226    MOV      PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
1994 010122 005760 000012          3$:  TST      RHDS1(RO)   ;DRIVE TIMEOUT ?
1995 010126 001004          BNE      4$           ;BR IF IT HAS
1996 010130 005737 001246    TST      WATCH         ;2 SECONDS ELAPSED ?
1997 010134 001372          BNE      3$           ;BR IF NOT
1998 010136 104006          ERROR 6           ;NO TIMEOUT AFTER 2 SECONDS
1999
2000                                ;*****
2001 010140          4$:
2002
2003                                ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2004
2005 010140 005037 001242          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
2006 010144 012737 000012 001122    MOV      @RHDS1, $B0ADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
2007 010152 060037 001122    ADD      RO, $B0ADR     ;ADD THE I/O BASE ADDRESS
2008 010156 012737 011600 001124    MOV      @MOL!PGM!DPR!DRY, $GDDAT ;COMPARISON CONSTANT
2009 010164 113760 001216 000010    MOV      PORTA, RHCS2(RO) ;SELECT PORT A.
2010 010172 016037 000012 001162    MOV      RHDS1(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2011 010200 013737 001162 001156    MOV      $TMP2, $TMP~   ;COPY IT INTO 'TMP0'
2012 010206 042737 100100 001156    BIC      @ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2013 010214 113760 001220 000010    MOV      PORTB, RHCS2(RO) ;SELECT PORT B.
2014 010222 016037 000012 001164    MOV      RHDS1(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2015 010230 013737 001164 001160    MOV      $TMP3, $TMP1   ;COPY IT INTO 'TMP1'
2016 010236 042737 100100 001160    BIC      @ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2017 010244 023737 001156 001160    CMP      $TMP0, $TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?

```

2018	010252	001006				BNE	66\$:BR IF NOT
2019	010254	005737	001156			TST	\$TMP0		:REGISTERS ARE THE SAME: ARE THEY ZERO ?
2020	010260	001037				BNE	68\$:BR IF NOT
2021	010262	104034				ERROR	34		:REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2022	010264	000137	010450			JMP	70\$:BYPASS THE REST OF THE CHECKS
2023	010270	013737	001162	001126	66\$:	MOV	\$TMP2,\$BDDAT		:SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2024	010276	013737	001220	001226		MOV	PORTB,PTNBR		:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2025	010304	113760	001220	000010		MOVB	PORTB,RHCS2(RO)		:SELECT PORT B.
2026	010312	005737	001156			TST	\$TMP0		:SEE IF STATUS EQ 0 FROM PORT A.
2027	010316	001414				BEQ	67\$:BR IF ZERO
2028	010320	013737	001216	001226		MOV	PORTA,PTNBR		:SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2029	010326	013737	001164	001126		MOV	\$TMP3,\$BDDAT		: 'BAD DATA' FOR ERROR TYPE OUT
2030	010334	113760	001216	000010		MOVB	PORTA,RHCS2(RO)		:SELECT PORT A.
2031	010342	005737	001160			TST	\$TMP1		:SEE IF STATUS EQ ZERO FROM PORT B.
2032	010346	001004				BNE	68\$:BR IF NOT
2033	010350	012737	177777	001242	67\$:	MOV	#-1,RELERR		:SET 'RELEASE ERROR' INDICATOR
2034	010356	104013				ERROR	13		:TYPE ERROR MESSAGE 13
2035	010360	013737	001162	001126	68\$:	MOV	\$TMP2,\$BDDAT		:LOOK FOR BIT FAILURES WHEN RHDS1 READ
2036	010366	013737	001216	001226		MOV	PORTA,PTNBR		:CHANGE PORT NUMBER
2037	010374	042737	100100	001162		BIC	#ATA!VV,\$TMP2		:DON'T CHECK ATTN BIT OR VV BIT
2038	010402	023737	001124	001162		CMP	\$GDDAT,\$TMP2		:ALL BITS OK ?
2039	010410	001401				BEQ	69\$:BR IF OK FROM PORT A.
2040	010412	104037				ERROR	37		:REPORT ERROR
2041	010414	013737	001164	001126	69\$:	MOV	\$TMP3,\$BDDAT		:CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
2042	010422	013737	001220	001226		MOV	PORTB,PTNBR		:CHANGE PORT NUMBER
2043	010430	042737	100100	001164		BIC	#ATA!VV,\$TMP3		:DON'T CHECK ATTN BIT OR VV BIT
2044	010436	023737	001124	001164		CMP	\$GDDAT,\$TMP3		:SEE IF READ OK FROM PORT B.
2045	010444	001401				BEQ	70\$:BR IF OK
2046	010446	104037				ERROR	37		:REPORT THE ERROR
2047	010450	000240			70\$:	NOP			

2048									
2049									
2050									:*****
2051	010452	113760	001220	000010		MOVB	PORTB,RHCS2(RO)		:SELECT PORT B
2052	010460	013737	001220	001226		MOV	PORTB,PTNBR		:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2053	010466	005037	001236			CLR	CKERR		:CLEAR THE 'CHECK ERROR' INDICATOR
2054	010472	016037	000012	001126		MOV	RHDS1(RO),\$BDDAT		:GET CONTENTS OF RHDS1
2055	010500	012737	000012	001122		MOV	#RHDS1,\$BDAOR		:FORM REGISTER ADDRESS OF ERROR MESSAGE
2056	010506	060037	001122			ADD	RO,\$BDAOR		:ADD RH11 BASE ADDRESS
2057	010512	005037	001124			CLR	\$GDDAT		:WHAT REGISTER SHOULD BE
2058	010516	013737	001126	001156		MOV	\$BDDAT,\$TMP0		:MOVE REGISTER CONTENTS TO 'TMP0'
2059	010524	042737	077777	001156		BIC	#ATA,\$TMP0		:SAVE SPECIFIED BITS
2060	010532	023737	001124	001156		CMP	\$GDDAT,\$TMP0		:COMPARE THE BITS
2061	010540	001414				BEQ	71\$:BR IF OK
2062	010542	013737	001126	001156		MOV	\$BDDAT,\$TMP4		:COPY 'BAD DATA'
2063	010550	042737	100000	001166		BIC	#ATA,\$TMP4		:CLEAR THE MASKED BITS
2064	010556	053737	001166	001124		BIS	\$TMP4,\$GDDAT		: 'OR' WITH GOOD DATA FOR TYPEOUT
2065	010564	104014				ERROR	14		:TYPE MESSAGE 14
2066	010566	005137	001236			COM	CKERR		:SET THE REGISTER COMPARE ERROR INDICATOR
2067	010572	000240			71\$:	NOP			
2068	010574	113760	001216	000010		MOVB	PORTA,RHCS2(RO)		:SELECT PORT A
2069	010602	013737	001216	001226		MOV	PORTA,PTNBR		:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2070	010610	005037	001236			CLR	CKERR		:CLEAR THE 'CHECK ERROR' INDICATOR
2071	010614	016037	000012	001126		MOV	RHDS1(RO),\$BDDAT		:GET CONTENTS OF RHDS1
2072	010622	012737	000012	001122		MOV	#RHDS1,\$BDAOR		:FORM REGISTER ADDRESS OF ERROR MESSAGE
2073	010630	060037	001122			ADD	RO,\$BDAOR		:ADD RH11 BASE ADDRESS


```

2074 010634 012737 100000 001124      MOV      #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
2075 010642 013737 001126 001156      MOV      $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
2076 010650 042737 077777 001156      BIC      #1CATA,$STMP0 ;SAVE SPECIFIED BITS
2077 010656 023737 001124 001156      CMP      $GDDAT,$STMP0 ;COMPARE THE BITS
2078 010664 001414                BEQ      725 ;BR IF OK
2079 010666 013737 001126 001166      MOV      $BDDAT,$STMP4 ;COPY 'BAD DATA'
2080 010674 042737 100000 001166      BIC      #ATA,$STMP4 ;CLEAR THE MASKED BITS
2081 010702 053737 001166 001124      BIS      $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2082 010710 104015                ERROR    15 ;TYPE MESSAGE 15
2083 010712 005137 001236                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2084 010716 000240                725:    NOP

```

```

;*****
;SET 'VV' FOR EACH PORT

```

```

2089 010720 012760 000011 000000      MOV      #11,RHCS1(RO) ;CLEAR THE DRIVE
2090 010726 012760 000021 000000      MOV      #21,RHCS1(RO) ;DO A READIN PRESET THROUGH PORT A
2091 010734 012760 000013 000000      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE
2092 010742 113760 001220 000010      MOV      PORTB,RHCS2(RO) ;SELECT PORT B
2093 010750 012760 000021 000000      MOV      #21,RHCS1(RO) ;DO A READIN PRESET THROUGH PORT B
2094 010756 012760 010000 000032      MOV      #FM122,RHOF(RO) ;SET 'FM122'
2095 010764 012760 000013 000000      MOV      #13,RHCS1(RO) ;RELEASE THE DRIVE

```

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

```

2099 010772 105737 001103                TSTB     $FRFLG ;DID AN ERROR OCCUR ?
2100 010776 001412                BEQ      TST7 ;BR IF NOT
2101 011000 032737 001000 177570      BIT      #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
2102 011006 001406                BEQ      TST7 ;BR IF NOT
2103 011010 105037 001103                CLRB     $ERFLG ;CLEAR THE ERROR FLAG
2104 011014 005037 001170                CLR      $TIMES ;CLEAR THE MAX ITERATION COUNT
2105 011020 000177 170064                JMP      $SLPERR ;GO TO THE LOOP ADDRESS

```

```

;*****
;TEST 7 TEST UNLOAD COMMAND THROUGH PORT B

```

- ```

*
*VERIFY THAT THE UNLOAD COMMAND FUNCTIONS PROPERLY AND THAT A PORT
*TIMEOUT WILL NOT OCCUR WHILE THE 'GO' BIT IS SET.
*
* A. ISSUE AN UNLOAD COMMAND THROUGH PORT B; VERIFY THAT THE
* DRIVE IS SEIZED.
*
* B. WAIT THE MEASURED TIMEOUT INTERVAL + 25%; VERIFY THAT THE DRIVE
* DOES NOT TIME OUT. VERIFY THAT THE 'GO' BIT IS STILL SET AND
* THAT 'DRY' AND 'PIP' ARE NOT SET.
*
* C. REQUEST THAT THE OPERATOR PRESS THE 'STANDBY' BUTTON ON THE DRIVE.
*
* D. WHEN THE DRIVE CYCLES UP, VERIFY THAT THE DRIVE IS STILL SEIZED
* BY PORT B, THAT THE 'VV' BIT FOR PORT B IS RESET, THAT THE
* ATTENTION BIT FOR PORT B IS SET, AND THAT THE ATTENTION BIT
* FOR PORT A IS NOT SET.
*
* E. WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE. WHEN THE TIMEOUT
* OCCURS, VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, THAT THE
* ATTENTION BIT FOR PORT B IS STILL SET, AND THAT THE ATTENTION

```

```

2130
2131
2132
2133
2134
2135
2136
2137 011024
2138 011024 000004
2139 011026 005737 001260
2140 011032 001406
2141 011034 100002
2142 011036 000137 002262
2143 011042 012737 177777 001260
2144 011050 112737 000007 00102
2145 011056 012737 011100 001106
2146 011064 012737 011100 001110
2147 011072 012737 000001 001170
2148
2149
2150
2151
2152
2153 011100
2154
2155
2156
2157 011100 113760 001216 000010
2158 011106 005060 000012
2159 011112 012760 000011 000000
2160 011120 012760 000013 000000
2161 011126 113760 001220 000010
2162 011134 005060 000012
2163 011140 012760 000011 000000
2164 011146 012760 000013 000000
2165 011154 113760 001220 000010
2166 011162 013737 001220 001226
2167 011170 013737 001220 001230
2168
2169
2170
2171
2172 011176 012760 000003 000000
2173 011204 013737 001256 001246
2174 011212 005737 001246
2175 011216 001375
2176
2177
2178
2179
2180 011220 005037 001236
2181 011224 016037 000000 001126
2182 011232 012737 000000 001122
2183 011240 060037 001122
2184 011244 012737 004001 001124
2185 011252 013737 001126 001156

```

```

;* BIT FOR PORT A IS NOT SET.
;*
;* F. VERIFY THAT THE 'VV' BIT FOR PORT A IS NOT SET.
;*
;* G. ISSUE A PACK ACKNOWLEDGE INSTRUCTION THROUGH BOTH PORTS.
;*

TST7:
SCOPE ; INITIALIZE THE SCOPE HANDLER
TST KYBCTL ; PERFORMING ONLY SINGLE TESTS ?
BEQ 25 ; BR IF NOT
BPL 15 ; BR IF JUST ENTERED TEST
JMP EXEC ; RETURN & GET NEXT TEST NUMBER
15: MOV #-1,KYBCTL ; SET SINGLE TEST INDICATOR
25: MOVB #7,$TSTNM ; TEST NUMBER
MOV #TEST7,$LPADR ; LOAD LOOP ON TEST ADDRESS
MOV #TEST7,$LPERR ; LOAD LOOP ON ERROR ADDRESS
MOV #1,$TIMES ; DO 1 ITERATION

;END OF 'SCOPE' SETUP - START OF MAIN TEST
TEST7:
;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RHCS2(RO) ; SELECT PORT #A
CLR RHDS1(RO) ; SEIZE THE DRIVE
MOV #11,RHCS1(RO) ; ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ; RELEASE THE DRIVE
MOVB PORTB,RHCS2(RO) ; SELECT PORT #B
CLR RHDS1(RO) ; SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RHCS1(RO) ; ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ; RELEASE THE DRIVE
MOVB PORTB,RHCS2(RO) ; SELECT PORT B
MOV PORTB,PTNBR ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB,SEIZPT ; SEIZING PORT'S ADDRESS

;DO AN UNLOAD COMMAND THROUGH PORT B
MOV #3,RHCS1(RO) ; ISSUE AN UNLOAD COMMAND THROUGH PORT B
MOV TIMEBP,WATCH ; TIMEOUT ONESHOT VALUE + 25%
15: TST WATCH ; FINISHED TIMEOUT ?
BNE 15 ; BR IF NOT

;IS THE STATUS OK ?
CLR CKERR ; CLEAR THE 'CHECK ERROR' INDICATOR
MOV RHCS1(RO),$BDDAT ; GET CONTENTS OF RHCS1
MOV #RHCS1,$B0ADR ; FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,$B0ADR ; ADD RH11 BASE ADDRESS
MOV #DVA!GO,$GDDAT ; WHAT REGISTER SHOULD BE
MOV $BDDAT,$TMPO ; MOVE REGISTER CONTENTS TO 'TMPO'

```



```

2186 011260 042737 173776 001156 BIC #1C4001,$STMP0 ;SAVE SPECIFIED BITS
2187 011266 023737 001124 001156 CMP $GDDAT,$STMP0 ;COMPARE THE BITS
2188 011274 001414 BEQ 64$;BR IF OK
2189 011276 013737 001126 001166 MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
2190 011304 042737 0C4001 001166 BIC #4001,$STMP4 ;CLEAR THE MASKED BITS
2191 011312 053737 001166 001124 BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2192 011320 104011 ERROR 11 ;TYPE MESSAGE 11
2193 011322 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2194 011326 000240 NOP
2195 011330 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2196 011334 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
2197 011342 012737 000012 001122 MOV #RHDS1,$BDAOR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2198 011350 060037 001122 ADD RO,$BDAOR ;ADD RHI1 BASE ADDRESS
2199 011354 012737 021500 001124 MOV #PIP!PGM!DPR!VV,$GDDAT ;WHAT REGISTER SHOULD BE
2200 011362 013737 001126 001156 MOV $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
2201 011370 042737 000077 001156 BIC #1C177700,$STMP0 ;SAVE SPECIFIED BITS
2202 011376 023737 001124 001156 CMP $GDDAT,$STMP0 ;COMPARE THE BITS
2203 011404 001414 BEQ 65$;BR IF OK
2204 011406 013737 001126 001166 MOV $BDDAT,$STMP4 ;COPY 'BAD DATA'
2205 011414 042737 177700 001166 BIC #177700,$STMP4 ;CLEAR THE MASKED BITS
2206 011422 053737 001166 001124 BIS $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2207 011430 104012 ERROR 12 ;TYPE MESSAGE 12
2208 011432 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2209 011436 000240 NOP
2210 011440 104400 022564 TYPE STANDBY ;TYPE THE STANDBY MESSAGE
2211 011444 013746 001220 MOV PORTB,-(SP) ;SAVE PORTB FOR TYPEOUT
2212 ;TYPE THE PORT NUMBER
2213 011450 104410 TYPDS , ;GO TYPE--DECIMAL ASCII WITH SIGN
2214 011452 104400 001201 TYPE ,SCLF ;CR-LF
2215
2216 ;*****
2217 ;WAIT FOR 'MOL' TO SET
2218
2219 011456 032760 010000 000012 2$: BIT #MOL,RHDS1(RO) ;WAIT FOR MOL TO SET
2220 011464 001774 BEQ 2$;LOOP UNTIL 'MOL' SETS
2221 011466 012737 003720 001246 MOV #2000,WATCH ;SETUP A 2 SECOND STALL
2222 011474 113760 001216 000010 MOV PORTA,RHCS2(RO) ;SELECT PORT A
2223 011502 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2224 011510 005760 000012 3$: TST RHDS1(RO) ;DRIVE TIMEOUT ?
2225 011514 001004 BNE 4$;BR IF IT HAS
2226 011516 005737 001246 TST WATCH ;2 SECONDS ELAPSED ?
2227 011522 001372 BNE 3$;BR IF NOT
2228 011524 104006 ERROR 6 ;NO TIMEOUT AFTER 2 SECONDS
2229
2230 ;*****
2231 011526 4$:
2232
2233 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2234
2235 011526 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
2236 011532 012737 000012 001122 MOV #RHDS1,$BDAOR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
2237 011540 060037 001122 ADD RO,$BDAOR ;ADD THE I/O BASE ADDRESS
2238 011544 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
2239 011552 113760 001216 000010 MOV PORTA,RHCS2(RO) ;SELECT PORT A
2240 011560 016037 000012 001162 MOV RHDS1(RO),$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2241 011566 013737 001162 001156 MOV $STMP2,$STMP0 ;COPY IT INTO 'STMP0'

```

```

2242 011574 042737 100100 001156 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2243 011602 113760 001220 000010 MOV PORTB,RHCS2(RO) ;SELECT PORT B.
2244 011610 016037 000012 001164 MOV RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2245 011616 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
2246 011624 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2247 011632 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2248 011640 001006 BNE 66$;BR IF NOT
2249 011642 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2250 011646 001037 BNE 68$;BR IF NOT
2251 011650 104034 ERROR 34 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2252 011652 000137 012036 JMP 70$;BYPASS THE REST OF THE CHECKS
2253 011656 013737 001162 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2254 011664 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2255 011672 113760 001220 000010 MOV PORTB,RHCS2(RO) ;SELECT PORT B.
2256 011700 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
2257 011704 001414 BEQ 67$;BR IF ZERO
2258 011706 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2259 011714 013737 001164 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
2260 011722 113760 001216 000010 MOV PORTA,RHCS2(RO) ;SELECT PORT A.
2261 011730 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
2262 011734 001004 BNE 68$;BR IF NOT
2263 011736 012737 177777 001242 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
2264 011744 104013 ERROR 13 ;TYPE ERROR MESSAGE 13
2265 011746 013737 001162 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RHDS1 READ
2266 011754 013737 001216 001226 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
2267 011762 042737 100100 001162 BIC #ATA!VV,$TMP2 ;DON'T CHECK ATTN BIT OR VV BIT
2268 011770 023737 001124 001162 CMP $GDDAT,$TMP2 ;ALL BITS OK ?
2269 011776 001401 BEQ 69$;BR IF OK FROM PORT A.
2270 012000 104037 ERROR 37 ;REPORT ERROR
2271 012002 013737 001164 001126 69$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
2272 012010 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
2273 012016 042737 100100 001164 BIC #ATA!VV,$TMP3 ;DON'T CHECK ATTN BIT OR VV BIT
2274 012024 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
2275 012032 001401 BEQ 70$;BR IF OK
2276 012034 104037 ERROR 37 ;REPORT THE ERROR
2277 012036 000240 70$: NOP
2278
2279
2280 ;:*****
;CHECK THE ATTENTION BITS
2281 012040 113760 001216 000010 MOV PORTA,RHCS2(RO) ;SELECT PORT A
2282 012046 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2283 012054 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2284 012060 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
2285 012066 012737 000012 001122 MOV #RHDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2286 012074 060037 001122 ADD RO,$BDDADR ;ADD RH11 BASE ADDRESS
2287 012100 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
2288 012104 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
2289 012112 042737 077777 001156 BIC #!CATA,$TMP0 ;SAVE SPECIFIED BITS
2290 012120 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
2291 012126 001414 BEQ 71$;BR IF OK
2292 012130 013737 001126 001166 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
2293 012136 042737 100000 001166 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
2294 012144 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2295 012152 104014 ERROR 14 ;TYPE MESSAGE 14
2296 012154 001137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2297 012160 000240 71$: NOP

```

```

2298 012162 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
2299 012170 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2300 012176 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2301 012202 016037 000012 001126 MOV RHDS1(RO),SBDDAT ;GET CONTENTS OF RHDS1
2302 012210 012737 000012 001122 MOV #RHDS1,SBADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2303 012216 060037 001122 ADD RO,SBADR ;ADD RH11 BASE ADDRESS
2304 012222 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
2305 012230 013737 001126 001156 MOV SBDDAT,$TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'
2306 012236 042737 077777 001156 BIC #+CATA,$TMPD ;SAVE SPECIFIED BITS
2307 012244 023737 001124 001156 CMP $GDDAT,$TMPD ;COMPARE THE BITS
2308 012252 001414 BEQ 72$;BR IF OK
2309 012254 013737 001126 001166 MOV SBDDAT,$TMP4 ;COPY 'BAD DATA'
2310 012262 042737 100000 001166 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
2311 012270 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2312 012276 104015 ERROR 15 ;TYPE MESSAGE 15
2313 012300 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2314 012304 000240 72$: NOP

```

```

;SET 'VV' FOR EACH PORT

```

```

2319 012306 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
2320 012314 012760 000021 000000 MOV #21,RHCS1(RO) ;DO A READIN PRESET THROUGH PORT B
2321 012322 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
2322 012330 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
2323 012336 012760 000021 000000 MOV #21,RHCS1(RO) ;DO A READIN PRESET THROUGH PORT A
2324 012344 012760 010000 000032 MOV #FMT22,RHOF(RO) ;SET 'FMT22'
2325 012352 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
2326
2327 ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
2328
2329 012360 105737 001103 TSTB $ERFLG ;DID AN ERROR OCCUR ?
2330 012364 001412 BEQ TST10 ;BR IF NOT
2331 012366 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
2332 012374 001406 BEQ TST10 ;BR IF NOT
2333 012376 105037 001103 CLRB $ERFLG ;CLEAR THE ERROR FLAG
2334 012402 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
2335 012406 000177 166476 JMP $SLPERR ;GO TO THE LOOP ADDRESS
2336
2337
2338
2339

```

```

*TEST 10 TEST 'CONTROLLER SELECT' SWITCH THROUGH PORT A DURING UNLOAD
*
*VERIFY THAT THE 'CONTROLLER SELECT' SWITCH IS INHIBITED WHEN THE
* RPO4 IS CYCLED DOWN BY AN UNLOAD COMMAND.
*
* A. ISSUE AN UNLOAD COMMAND THROUGH PORT A.
*
* B. REQUEST THAT THE OPERATOR SWITCH THE 'CONTROLLER SELECT' SWITCH
* TO A AND THEN PRESS THE 'STANDBY' BUTTON.
*
* C. WAIT FOR 'MOL' TO SET BY MONITORING RHDS1 THROUGH PORT
* A. WHEN THE DRIVE HAS CYCLED UP ('MOL' HAS SET), ISSUE
* RELEASE COMMAND THROUGH PORT A.
*

```

```

2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353

```

```

2354
2355
2356
2357
2358
2359
2360
2361 012412
2362 012412 000004
2363 012414 005737 001260
2364 012420 001406
2365 012422 100002
2366 012424 000137 002262
2367 012430 012737 177777 001260 1S:
2368 012436 112737 000010 001102 2S:
2369 012444 012737 012466 001106
2370 012452 012737 012466 001110
2371 012460 012737 000001 001170
2372
2373
2374
2375
2376
2377 012466
2378
2379
2380
2381 012466 113760 001216 000010
2382 012474 005060 000012
2383 012500 012760 000011 000000
2384 012506 012760 000013 000000
2385 012514 113760 001220 000010
2386 012522 005060 000012
2387 012526 012760 000011 000000
2388 012534 012760 000013 000000
2389 012542 113760 001216 000010
2390 012550 013737 001216 001226
2391 012556 013737 001216 001230
2392 012564 012760 000003 000000
2393 012572 104400 022776
2394 012576 104400 022564
2395 012602 013746 001216
2396
2397 012606 104410
2398 012610 104400 001201
2399 012614 032760 010000 000012 1S:
2400 012622 001774
2401 012624 012760 000011 000000
2402 012632 012760 000021 000000
2403 012640 012760 000013 000000
2404
2405
2406
2407 012646 005037 001242
2408 012652 012737 000012 001122
2409 012660 060037 001122

```

```

;* D. VERIFY THAT THE DRIVE CAN BE ACCESSED BY BOTH PORTS AND THAT
;* THE DRIVE IS STILL IN NEUTRAL.
;*
;* E. REQUEST THAT THE OPERATOR RETURN THE 'CONTROLLER SELECT' SWITCH
;* TO A/B'.

TEST10:
SCOPE ;INITIALIZE THE SCOPE HANDLER
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
BEQ 2S ;BR IF NOT
BPL 1S ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1S: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2S: MOVB #10,$STNM ;TEST NUMBER
MOV #TEST10,$LPADR ;LOAD LOOP ON TEST ADDRESS
MOV #TEST10,$LPERR ;LOAD LOOP ON ERROR ADDRESS
MOV #1,$TIMES ;DO 1 ITERATION

;*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
TEST10:
;CLEAR ATTENTION BITS FOR BOTH PORTS
MOVB PORTA,RHCS2(RO) ;SELECT PORT #A
CLR RHDS1(RO) ;SEIZE THE DRIVE
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
MOVB PORTB,RHCS2(RO) ;SELECT PORT #B
CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
MOVB PORTA,RHCS2(RO) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,SEIZPT ;SEIZING PORT'S ADDRESS
MOV #3,RHCS1(RO) ;ISSUE AN UNLOAD COMMAND THROUGH PORT A
TYPE ,SWTCHA ;SWITCH TO PORT A MESSAGE
TYPE ,STANDBY ;TYPE PRESS STANDBY
MOV PORTA,-(SP) ;SAVE PORTA FOR TYPEOUT
;TYPE THE PORT NUMBER
;GO TYPE--DECIMAL ASCII WITH SIGN
TYPDS
TYPE ,SCRLF ;CR-LF
1S: BIT #MOL,RHDS1(RO) ;TEST 'MOL'
BEQ 1S ;BR IF NOT SET
MOV #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR
MOV #21,RHCS1(RO) ;ISSUE A READIN PRESET
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE

;VERIFY THAT THE DRIVE IS IN NEUTRAL
CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS

```

|      |        |        |        |        |       |       |                          |                                                    |
|------|--------|--------|--------|--------|-------|-------|--------------------------|----------------------------------------------------|
| 2410 | 012664 | 012737 | 011600 | 001124 |       | MOV   | #MOL!PGM!DPR!DRY,\$GDDAT | :COMPARISON CONSTANT                               |
| 2411 | 012672 | 113760 | 001216 | 000C10 |       | MOV   | PORTA,RHCS2(RO)          | :SELECT PORT A.                                    |
| 2412 | 012700 | 016037 | 000012 | 001162 |       | MOV   | RHDS1(RO),STMP2          | :GET THE DRIVE STATUS REGISTER FROM PORT A.        |
| 2413 | 012706 | 013737 | 001162 | 001156 |       | MOV   | STMP2,STMP0              | :COPY IT INTO 'STMP0'                              |
| 2414 | 012714 | 042737 | 100100 | 001156 |       | BIC   | #ATA!VV,STMP0            | :CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 2415 | 012722 | 113760 | 001220 | 000010 |       | MOV   | PORTB,RHCS2(RO)          | :SELECT PORT B.                                    |
| 2416 | 012730 | 016037 | 000012 | 001164 |       | MOV   | RHDS1(RO),STMP3          | :GET THE DRIVE STATUS REGISTER FROM PORT B.        |
| 2417 | 012736 | 013737 | 001164 | 001160 |       | MOV   | STMP3,STMP1              | :COPY IT INTO 'STMP1'                              |
| 2418 | 012744 | 042737 | 100100 | 001160 |       | BIC   | #ATA!VV,STMP1            | :CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 2419 | 012752 | 023737 | 001156 | 001160 |       | CMP   | STMP0,STMP1              | :IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| 2420 | 012760 | 001006 |        |        |       | BNE   | 64\$                     | :BR IF NOT                                         |
| 2421 | 012762 | 005737 | 001156 |        |       | TST   | STMP0                    | :REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| 2422 | 012766 | 001037 |        |        |       | BNE   | 66\$                     | :BR IF NOT                                         |
| 2423 | 012770 | 104034 |        |        |       | ERPOR | 34                       | :REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| 2424 | 012772 | 000137 | 013156 |        |       | JMP   | 68\$                     | :BYPASS THE REST OF THE CHECKS                     |
| 2425 | 012776 | 013737 | 001162 | 001126 | 64\$: | MOV   | STMP2,\$BDDAT            | :SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE        |
| 2426 | 013004 | 013737 | 001220 | 001226 |       | MOV   | PORTB,PTNBR              | :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 2427 | 013012 | 113760 | 001220 | 000010 |       | MOV   | PORTB,RHCS2(RO)          | :SELECT PORT B.                                    |
| 2428 | 013020 | 005737 | 001156 |        |       | TST   | STMP0                    | :SEE IF STATUS EQ 0 FROM PORT A.                   |
| 2429 | 013024 | 001414 |        |        |       | BEQ   | 65\$                     | :BR IF ZERO                                        |
| 2430 | 013026 | 013737 | 001216 | 001226 |       | MOV   | PORTA,PTNBR              | :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 2431 | 013034 | 013737 | 001164 | 001126 |       | MOV   | STMP3,\$BDDAT            | : 'BAD DATA' FOR ERROR TYPE OUT                    |
| 2432 | 013042 | 113760 | 001216 | 000010 |       | MOV   | PORTA,RHCS2(RO)          | :SELECT PORT A.                                    |
| 2433 | 013050 | 005737 | 001160 |        |       | TST   | STMP1                    | :SEE IF STATUS EQ ZERO FROM PORT B.                |
| 2434 | 013054 | 001004 |        |        |       | BNE   | 66\$                     | :BR IF NOT                                         |
| 2435 | 013056 | 012737 | 177777 | 001242 | 65\$: | MOV   | #-1,RELERR               | :SET 'RELEASE ERROR' INDICATOR                     |
| 2436 | 013064 | 104016 |        |        |       | ERROR | 16                       | :TYPE ERROR MESSAGE 16                             |
| 2437 | 013066 | 013737 | 001162 | 001126 | 66\$: | MOV   | STMP2,\$BDDAT            | :LOOK FOR BIT FAILURES WHEN RHDS1 READ             |
| 2438 | 013074 | 013737 | 001216 | 001226 |       | MOV   | PORTA,PTNBR              | :CHANGE PORT NUMBER                                |
| 2439 | 013102 | 042737 | 000100 | 001162 |       | BIC   | #VV,STMP2                | :DON'T CHECK THE VV BIT                            |
| 2440 | 013110 | 023737 | 001124 | 001162 |       | CMP   | \$GDDAT,STMP2            | :ALL BITS OK ?                                     |
| 2441 | 013116 | 001401 |        |        |       | BEQ   | 67\$                     | :BR IF OK FROM PORT A.                             |
| 2442 | 013120 | 104037 |        |        |       | ERROR | 37                       | :REPORT ERROR                                      |
| 2443 | 013122 | 013737 | 001164 | 001126 | 67\$: | MOV   | STMP3,\$BDDAT            | :CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.       |
| 2444 | 013130 | 013737 | 001220 | 001226 |       | MOV   | PORTB,PTNBR              | :CHANGE PORT NUMBER                                |
| 2445 | 013136 | 042737 | 000100 | 001164 |       | BIC   | #VV,STMP3                | :DON'T CHECK THE VV BIT                            |
| 2446 | 013144 | 023737 | 001124 | 001164 |       | CMP   | \$GDDAT,STMP3            | :SEE IF READ OK FROM PORT B.                       |
| 2447 | 013152 | 001401 |        |        |       | BEQ   | 68\$                     | :BR IF OK                                          |
| 2448 | 013154 | 104037 |        |        |       | ERROR | 37                       | :REPORT THE ERROR                                  |
| 2449 | 013156 | 000240 |        |        | 68\$: | NOP   |                          |                                                    |
| 2450 |        |        |        |        |       |       |                          |                                                    |
| 2451 |        |        |        |        |       |       |                          | :IF ERROR OCCURED, CHECK FOR LOOP ON TEST          |
| 2452 |        |        |        |        |       |       |                          |                                                    |
| 2453 | 013160 | 105737 | 001103 |        |       | TSTB  | \$ERFLG                  | :DID AN ERROR OCCUR                                |
| 2454 | 013164 | 001412 |        |        |       | BEQ   | 2\$                      | :BR IF NOT                                         |
| 2455 | 013166 | 032737 | 001000 | 177570 |       | BIT   | #SW09,SWR                | :SEE IF LOOP ON ERROR (SWR9=1)                     |
| 2456 | 013174 | 001406 |        |        |       | BEQ   | 2\$                      | :BR IF NOT                                         |
| 2457 | 013176 | 105037 | 001103 |        |       | CLRB  | \$ERFLG                  | :CLEAR THE ERROR FLAG                              |
| 2458 | 013202 | 005037 | 001170 |        |       | CLR   | \$TIMES                  | :CLEAR THE MAX ITERATION COUNT                     |
| 2459 | 013206 | 000177 | 165676 |        |       | JMP   | \$JLPERR                 | :GO TO THE LOOP ADDRESS                            |
| 2460 | 013212 | 005737 | 001260 |        | 2\$:  | TST   | KYBCTL                   | :SINGLE TEST MODE ?                                |
| 2461 | 013216 | 001406 |        |        |       | BEQ   | 3\$                      | :BR IF NOT                                         |
| 2462 | 013220 | 032737 | 040000 | 177570 |       | BIT   | #SW14,SWR                | :LOOP ON TEST ?                                    |
| 2463 | 013226 | 001002 |        |        |       | BNE   | 3\$                      | :BR IF LOOPING                                     |
| 2464 | 013230 | 104400 | 022707 |        |       | TYPE  | ,SWTCHN                  | :RETURN CONTROLLER SELECT SWITCH TO 'A/B'          |
| 2465 | 013234 |        |        |        | 3\$:  |       |                          |                                                    |

K04

2466 013234 113760 001220 000010  
2467 013242 013737 001220 001226  
2468 013250 012760 000011 000000  
2469 013256 012760 000021 000000  
2470 013264 012760 010000 000032  
2471 013272 012760 000013 000000  
2472 013300 000400

MOV B PORTB,RHCS2(RO) ;SELECT PORT B  
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
MOV #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR  
MOV #21,RHCS1(RO) ;ISSUE A READIN PRESET  
MOV #FMT22,RHOF(RO) ;SET 'FMT22'  
MOV #13,RHCS1(RO) ;RELEASE PORT B  
BR TST11 ;GO TO NEXT TEST

2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480  
2481  
2482  
2483  
2484  
2485  
2486  
2487  
2488  
2489  
2490  
2491  
2492  
2493  
2494  
2495

\*\*\*\*\*  
;TEST 11 TEST 'CONTROLLER SELECT' SWITCH THROUGH PORT B DURING UNLOAD  
;  
;VERIFY THAT THE 'CONTROLLER SELECT' SWITCH IS INHIBITED WHEN THE  
; RPO4 IS CYCLED DOWN BY AN UNLOAD COMMAND.  
;  
; A. ISSUE AN UNLOAD COMMAND THROUGH PORT B.  
;  
; B. REQUEST THAT THE OPERATOR SWITCH THE 'CONTROLLER SELECT' SWITCH  
; TO B AND THEN PRESS THE 'STANDBY' BUTTON.  
;  
; C. WAIT FOR 'MOL' TO SET BY MONITORING RHDS1 THROUGH PORT  
; B. WHEN THE DRIVE HAS CYCLED UP ('MOL' HAS SET), ISSUE  
; RELEASE COMMAND THROUGH PORT B.  
;  
; D. VERIFY THAT THE DRIVE CAN BE ACCESSED BY BOTH PORTS AND THAT  
; THE DRIVE IS STILL IN NEUTRAL.  
;  
; E. REQUEST THAT THE OPERATOR RETURN THE 'CONTROLLER SELECT' SWITCH  
; TO 'A/B'.  
\*\*\*\*\*

2496 013302  
2497 013302 000004  
2498 013304 005737 001260  
2499 013310 001406  
2500 013312 100002  
2501 013314 000137 002262  
2502 013320 012737 177777 001260  
2503 013326 112737 000011 001102  
2504 013334 012737 013356 001106  
2505 013342 012737 013356 001110  
2506 013350 012737 000001 001170  
2507  
2508  
2509  
2510  
2511  
2512 013356  
2513  
2514  
2515  
2516 013356 113760 001216 000010  
2517 013364 005060 000012  
2518 013370 012760 000011 000000  
2519 013376 012760 000013 000000  
2520 013404 113760 001220 000010  
2521 013412 005060 000012

TST11:  
SCOPE ;INITIALIZE THE SCOPE HANDLER  
TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?  
BEQ 25 ;BR IF NOT  
BPL 15 ;BR IF JUST ENTERED TEST  
JMP EXEC ;RETURN & GET NEXT TEST NUMBER  
15: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR  
25: MOV B #11,\$STSTM ;TEST NUMBER  
MOV #TEST11,\$LPADR ;LOAD LOOP ON TEST ADDRESS  
MOV #TEST11,\$LPERR ;LOAD LOOP ON ERROR ADDRESS  
MOV #1,\$TIMES ;DO 1 ITERATION

\*\*\*\*\*  
;END OF 'SCOPE' SETUP - START OF MAIN TEST

TEST11:  
;CLEAR ATTENTION BITS FOR BOTH PORTS  
MOV B PORTA,RHCS2(RO) ;SELECT PORT #A  
CLR RHDS1(RO) ;SEIZE THE DRIVE  
MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR  
MOV #13,RHCS1(RO) ;RELEASE THE DRIVE  
MOV B PORTB,RHCS2(RO) ;SELECT PORT #B  
CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'



|      |        |        |        |        |       |       |                          |                                                    |
|------|--------|--------|--------|--------|-------|-------|--------------------------|----------------------------------------------------|
| 2522 | 013416 | 012760 | 000011 | 000000 |       | MOV   | #11,RHCS1(RO)            | ;ISSUE DRIVE CLEAR                                 |
| 2523 | 013424 | 012760 | 000013 | 000000 |       | MOV   | #13,RHCS1(RO)            | ;RELEASE THE DRIVE                                 |
| 2524 | 013432 | 113760 | 001220 | 000010 |       | MOVB  | PORTB,RHCS2(RO)          | ;SELECT PORT B                                     |
| 2525 | 013440 | 013737 | 001220 | 001226 |       | MOV   | PORTB,PTNBR              | ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT         |
| 2526 | 013446 | 013737 | 001220 | 001230 |       | MOV   | PORTB,SEIZPT             | ;SEIZING PORT'S ADDRESS                            |
| 2527 | 013454 | 012760 | 000003 | 000000 |       | MOV   | #3,RHCS1(RO)             | ;ISSUE AN UNLOAD COMMAND THROUGH PORT B            |
| 2528 | 013462 | 104400 | 023061 |        |       | TYPE  | ,SWTCHB                  | ;SWITCH TO PORT B MESSAGE                          |
| 2529 | 013466 | 104400 | 022564 |        |       | TYPE  | STANDBY                  | ;TYPE PRESS STANDBY                                |
| 2530 | 013472 | 013746 | 001220 |        |       | MOV   | PORTB,-(SP)              | ;SAVE PORTB FOR TYPEOUT                            |
| 2531 |        |        |        |        |       |       |                          | ;TYPE THE PORT NUMBER                              |
| 2532 | 013476 | 104410 |        |        |       | TYPDS |                          | ;GO TYPE--DECIMAL ASCII WITH SIGN                  |
| 2533 | 013500 | 104400 | 001201 |        |       | TYPE  | ,\$CRLF                  | ;CR-LF                                             |
| 2534 | 013504 | 032760 | 010000 | 000012 | 1\$:  | BIT   | #MOL,RHDS1(RO)           | ;TEST 'MOL'                                        |
| 2535 | 013512 | 001774 |        |        |       | BEQ   | 1\$                      | ;BR IF NOT SET                                     |
| 2536 | 013514 | 012760 | 000011 | 000000 |       | MOV   | #11,RHCS1(RO)            | ;ISSUE A DRIVE CLEAR                               |
| 2537 | 013522 | 012760 | 000021 | 000000 |       | MOV   | #21,RHCS1(RO)            | ;ISSUE A READIN PRESET                             |
| 2538 | 013530 | 012760 | 000013 | 000000 |       | MOV   | #13,RHCS1(RO)            | ;RELEASE THE DRIVE                                 |
| 2539 |        |        |        |        |       |       |                          |                                                    |
| 2540 |        |        |        |        |       |       |                          | ;VERIFY THAT THE DRIVE IS IN NEUTRAL               |
| 2541 |        |        |        |        |       |       |                          |                                                    |
| 2542 | 013536 | 005037 | 001242 |        |       | CLR   | RELERR                   | ;CLEAR THE 'RELEASE ERROR' INDICATOR               |
| 2543 | 013542 | 012737 | 000012 | 001122 |       | MOV   | #RHDS1,\$BDDADR          | ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT             |
| 2544 | 013550 | 060037 | 001122 |        |       | ADD   | RO,\$BDDADR              | ;ADD THE I/O BASE ADDRESS                          |
| 2545 | 013554 | 012737 | 011600 | 001124 |       | MOV   | #MOL!PGM!DPR!DRY,\$GDDAT | ;COMPARISON CONSTANT                               |
| 2546 | 013562 | 113760 | 001216 | 000010 |       | MOVB  | PORTA,RHCS2(RO)          | ;SELECT PORT A.                                    |
| 2547 | 013570 | 016037 | 000012 | 001162 |       | MOV   | RHDS1(RO),\$TMP2         | ;GET THE DRIVE STATUS REGISTER FROM PORT A.        |
| 2548 | 013576 | 013737 | 001162 | 001156 |       | MOV   | \$TMP2,\$TMP0            | ;COPY IT INTO 'TMP0'                               |
| 2549 | 013604 | 042737 | 100100 | 001156 |       | BIC   | #ATA!VV,\$TMP0           | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 2550 | 013612 | 113760 | 001220 | 000010 |       | MOVB  | PORTB,RHCS2(RO)          | ;SELECT PORT B.                                    |
| 2551 | 013620 | 016037 | 000012 | 001164 |       | MOV   | RHDS1(RO),\$TMP3         | ;GET THE DRIVE STATUS REGISTER FROM PORT B.        |
| 2552 | 013626 | 013737 | 001164 | 001160 |       | MOV   | \$TMP3,\$TMP1            | ;COPY IT INTO 'TMP1'                               |
| 2553 | 013634 | 042737 | 100100 | 001160 |       | BIC   | #ATA!VV,\$TMP1           | ;CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 2554 | 013642 | 023737 | 001156 | 001160 |       | CMP   | \$TMP0,\$TMP1            | ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| 2555 | 013650 | 001006 |        |        |       | BNE   | 64\$                     | ;BR IF NOT                                         |
| 2556 | 013652 | 005737 | 001156 |        |       | TST   | \$TMP0                   | ;REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| 2557 | 013656 | 001037 |        |        |       | BNE   | 66\$                     | ;BR IF NOT                                         |
| 2558 | 013660 | 104034 |        |        |       | ERROR | 34                       | ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| 2559 | 013662 | 000137 | 014046 |        |       | JMP   | 68\$                     | ;BYPASS THE REST OF THE CHECKS                     |
| 2560 | 013666 | 013737 | 001162 | 001126 | 64\$: | MOV   | \$TMP2,\$BDDAT           | ;SET UP POSSIBLE JAD DATA FOR ERROR MESSAGE        |
| 2561 | 013674 | 013737 | 001220 | 001226 |       | MOV   | PORTB,PTNBR              | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 2562 | 013702 | 113760 | 001220 | 000010 |       | MOVB  | PORTB,RHCS2(RO)          | ;SELECT PORT B.                                    |
| 2563 | 013710 | 005737 | 001156 |        |       | TST   | \$TMP0                   | ;SEE IF STATUS EQ 0 FROM PORT A.                   |
| 2564 | 013714 | 001414 |        |        |       | BEQ   | 65\$                     | ;BR IF ZERO                                        |
| 2565 | 013716 | 013737 | 001216 | 001226 |       | MOV   | PORTA,PTNBR              | ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 2566 | 013724 | 013737 | 001164 | 001126 |       | MOV   | \$TMP3,\$BDDAT           | ; 'BAD DATA' FOR ERROR TYPE OUT                    |
| 2567 | 013732 | 113760 | 001216 | 000010 |       | MOVB  | PORTA,RHCS2(RO)          | ;SELECT PORT A.                                    |
| 2568 | 013740 | 005737 | 001160 |        |       | TST   | \$TMP1                   | ;SEE IF STATUS EQ ZERO FROM PORT B.                |
| 2569 | 013744 | 001004 |        |        |       | BNE   | 66\$                     | ;BR IF NOT                                         |
| 2570 | 013746 | 012737 | 177777 | 001242 | 65\$: | MOV   | #-1,RELERR               | ;SET 'RELEASE ERROR' INDICATOR                     |
| 2571 | 013754 | 104016 |        |        |       | ERROR | 16                       | ;TYPE ERROR MESSAGE 16                             |
| 2572 | 013756 | 013737 | 001162 | 001126 | 66\$: | MOV   | \$TMP2,\$BDDAT           | ;LOOK FOR BIT FAILURES WHEN RHDS1 READ             |
| 2573 | 013764 | 013737 | 001216 | 001226 |       | MOV   | PORTA,PTNBR              | ;CHANGE PORT NUMBER                                |
| 2574 | 013772 | 042737 | 000100 | 001162 |       | BIC   | #VV,\$TMP2               | ;DON'T CHECK THE VV BIT                            |
| 2575 | 014000 | 023737 | 001124 | 001162 |       | CMP   | \$GDDAT,\$TMP2           | ;ALL BITS OK ?                                     |
| 2576 | 014006 | 001401 |        |        |       | BEQ   | 67\$                     | ;BR IF OK FROM PORT A.                             |
| 2577 | 014010 | 104037 |        |        |       | ERROR | 37                       | ;REPORT ERROR                                      |

```

2578 014012 013737 001164 001126 67$: MOV $TMP3,$BDDAT ;CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.
2579 014020 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
2580 014026 042737 000100 001164 BIC #VV,$TMP3 ;DON'T CHECK THE VV BIT
2581 014034 023737 001124 001164 CMP $GDDAT,$TMP3 ;SEE IF READ OK FROM PORT B.
2582 014042 001401 BEQ 68$;BR IF OK
2583 014044 104037 ERROR 37 ;REPORT THE ERROR
2584 014046 000240 68$: NOP

```

;IF ERROR OCCURED, CHECK FOR LOOP ON TEST

```

2588 014050 105737 001103 TSTB $ERFLG ;DID AN ERROR OCCUR
2589 014054 001412 BEQ 2$;BR IF NOT
2590 014056 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR (SWR9=1)
2591 014064 001406 BEQ 2$;BR IF NOT
2592 014066 105037 001103 CLRB $ERFLG ;CLEAR THE ERROR FLAG
2593 014072 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
2594 014076 000177 165006 JMP @SLPERR ;GO TO THE LOOP ADDRESS
2595 014102 032737 040000 177570 2$: BIT #SW14,SWR ;LOOP ON TEST ?
2596 014110 001002 BNE 3$;BR IF LOOPING
2597 014112 104400 022707 TYPE ,SWTCHN ;RETURN CONTROLLER SELECT SWITCH TO 'A/B'
2598 014116 3$:
2599 014116 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
2600 014124 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2601 014132 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR
2602 014140 012760 000021 000000 MOV #21,RHCS1(RO) ;ISSUE A READIN PRESET
2603 014146 012760 010000 000032 MOV #FMT22,RHOF(RO) ;SET 'FMT22'
2604 014154 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE PORT A
2605 014162 000400 BR TST12 ;GO TO NEXT TEST

```

```

*TEST 12 TEST 'CONTROLLER SELECT' SWITCH, DRIVE CYCLED UP
*
*TEST THE OPERATION OF THE 'CONTROLLER SELECT' SWITCH (DRIVE CYCLED UP).
*
* A. SWITCH TO CONTROLLER 'A' POSITION. VERIFY THAT THE DRIVE IS IN
* NEUTRAL AND THAT THE STATUS BITS IN RHDS1, AS READ THROUGH BOTH
* PORTS, ARE CORRECT.
*
* B. SWITCH TO CONTROLLER 'B' POSITION. VERIFY THAT THE DRIVE IS IN
* NEUTRAL AND THAT THE STATUS BITS IN RHDS1, AS READ THROUGH BOTH
* PORTS, ARE CORRECT.
*
* C. RETURN THE 'CONTROLLER SELECT' SWITCH TO THE 'A/B' POSITION. VERIFY
* THE DRIVE STATE.
*

```

```

2625 014164 000004 TST12: SCOPE ;INITIALIZE THE SCOPE HANDLER
2626 014164 005737 001260 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
2627 014166 001406 BEQ 2$;BR IF NOT
2628 014172 001406 BPL 1$;BR IF JUST ENTERED TEST
2629 014174 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
2630 014176 000137 002262 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2631 014202 012737 177777 001260 2$: MOVB #12,$TSTNM ;TEST NUMBER
2632 014210 112737 000012 001102 MOV #TEST12,$LPADR ;LOAD LOOP ON TEST ADDRESS
2633 014216 012737 014240 001106

```



```

2634 014224 012737 014240 001110 MOV #TEST12,$LPERR ;LOAD LOOP ON ERROR ADDRESS
2635 014232 012737 000001 001170 MOV #1,$TIMES ;;DO 1 ITERATION
2636
2637
2638 ;:*****
2639 ;END OF 'SCOPE' SETUP - START OF MAIN TEST
2640
2641 014240 TEST12:
2642
2643 ;CLEAR ATTENTION BITS FOR BOTH PORTS
2644
2645 014240 113760 001216 000010 MOV PORTA,RHCS2(RO) ;SELECT PORT #A
2646 014246 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE
2647 014252 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
2648 014260 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
2649 014266 113760 001220 000010 MOV PORTB,RHCS2(RO) ;SELECT PORT #B
2650 014274 005060 000012 CLR RHDS1(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
2651 014300 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE DRIVE CLEAR
2652 014306 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE
2653 014314 104400 022776 TYPE ,SWTCHA ;SWITCH TO 'A'
2654 014320 104400 023144 TYPE ,CONTUE ;PRESS 'CONTINUE'
2655 014324 000000 HALT
2656
2657 ;VERIFY THAT THE DRIVE IS IN NEUTRAL
2658
2659 014326 005037 001242 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
2660 014332 012737 000012 001122 MOV #RHDS1,$BDADR ;FORM THE ADDRESS OF RHDS1 FOR TYPEOUT
2661 014340 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
2662 014344 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
2663 014352 113760 001216 000010 MOV PORTA,RHCS2(RO) ;SELECT PORT A.
2664 014360 016037 000012 001162 MOV RHDS1(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
2665 014366 013737 001162 001156 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
2666 014374 042737 100100 001156 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2667 014402 113760 001220 000010 MOV PORTB,RHCS2(RO) ;SELECT PORT B.
2668 014410 016037 000012 001164 MOV RHDS1(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
2669 014416 013737 001164 001160 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
2670 014424 042737 100100 001160 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
2671 014432 023737 001156 001160 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
2672 014440 001006 BNE 64$;BR IF NOT
2673 014442 005737 001156 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
2674 014446 001045 BNE 66$;BR IF NOT
2675 014450 104034 ERROR 34 ;REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED
2676 014452 000137 014636 JMP 68$;BYPASS THE REST OF THE CHECKS
2677 014456 013737 001162 001126 64$: MOV $TMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
2678 014464 013737 001220 001226 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2679 014472 113760 001220 000010 MOV PORTB,RHCS2(RO) ;SELECT PORT B.
2680 014500 005737 001156 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
2681 014504 001414 BEQ 65$;BR IF ZERO
2682 014506 013737 001216 001226 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
2683 014514 013737 001164 001126 MOV $TMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
2684 014522 113760 001216 000010 MOV PORTA,RHCS2(RO) ;SELECT PORT A.
2685 014530 005737 001160 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
2686 014534 001012 BNE 66$;BR IF NOT
2687 014536 012737 177777 001242 65$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
2688 014544 012760 000011 000000 MOV #11,RHCS1(RO) ;CLEAR THE DRIVE
2689 014552 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE THE DRIVE

```

|      |        |        |        |        |       |       |                             |  |                                                    |
|------|--------|--------|--------|--------|-------|-------|-----------------------------|--|----------------------------------------------------|
| 2690 | 014560 | 104017 |        |        |       | ERROR | 17                          |  | :TYPE ERROR MESSAGE 17                             |
| 2691 | 014562 | 013737 | 001162 | 001126 | 66\$: | MOV   | \$TMP2,\$BDDAT              |  | :LOOK FOR BIT FAILURES WHEN RHDS1 READ             |
| 2692 | 014570 | 013737 | 001216 | 001226 |       | MOV   | PORTA,PTNBR                 |  | :CHANGE PORT NUMBER                                |
| 2693 | 014576 | 023737 | 001124 | 001162 |       | CMP   | \$GDDAT,\$TMP2              |  | :ALL BITS OK ?                                     |
| 2694 | 014604 | 001401 |        |        |       | BEQ   | 67\$                        |  | :BR IF OK FROM PORT A.                             |
| 2695 | 014606 | 104037 |        |        |       | ERROR | 37                          |  | :REPORT ERROR                                      |
| 2696 | 014610 | 013737 | 001164 | 001126 | 67\$: | MOV   | \$TMP3,\$BDDAT              |  | :CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.       |
| 2697 | 014616 | 013737 | 001220 | 001226 |       | MOV   | PORTB,PTNBR                 |  | :CHANGE PORT NUMBER                                |
| 2698 | 014624 | 023737 | 001124 | 001164 |       | CMP   | \$GDDAT,\$TMP3              |  | :SEE IF READ OK FROM PORT B.                       |
| 2699 | 014632 | 001401 |        |        |       | BEQ   | 68\$                        |  | :BR IF OK                                          |
| 2700 | 014634 | 104037 |        |        |       | ERROR | 37                          |  | :REPORT THE ERROR                                  |
| 2701 | 014636 | 000240 |        |        | 68\$: | NOP   |                             |  |                                                    |
| 2702 | 014640 | 104400 | 023061 |        |       | TYPE  | ,SWTCHB                     |  | :SWITCH TO 'B'                                     |
| 2703 | 014644 | 104400 | 023144 |        |       | TYPE  | ,CONTUE                     |  | :PRESS 'CONTINUE'                                  |
| 2704 | 014650 | 000000 |        |        |       | HALT  |                             |  |                                                    |
| 2705 |        |        |        |        |       |       |                             |  |                                                    |
| 2706 |        |        |        |        |       |       |                             |  | :VERIFY THAT THE DRIVE IS IN NEUTRAL               |
| 2707 |        |        |        |        |       |       |                             |  |                                                    |
| 2708 | 014652 | 005037 | 001242 |        |       | CLR   | RELERR                      |  | :CLEAR THE 'RELEASE ERROR' INDICATOR               |
| 2709 | 014656 | 012737 | 000012 | 001122 |       | MOV   | #RHDS1,\$BDAOR              |  | :FORM THE ADDRESS OF RHDS1 FOR TYPEOUT             |
| 2710 | 014664 | 060037 | 001122 |        |       | ADD   | RO,\$BDAOR                  |  | :ADD THE I/O BASE ADDRESS                          |
| 2711 | 014670 | 012737 | 011700 | 001124 |       | MOV   | #MOL!PGM!DPR!DRY!VV,\$GDDAT |  | :COMPARISON CONSTANT                               |
| 2712 | 014676 | 113760 | 001216 | 000010 |       | MOVB  | PORTA,RHCS2(RO)             |  | :SELECT PORT A.                                    |
| 2713 | 014704 | 016037 | 000012 | 001162 |       | MOV   | RHDS1(RO),\$TMP2            |  | :GET THE DRIVE STATUS REGISTER FROM PORT A.        |
| 2714 | 014712 | 013737 | 001162 | 001156 |       | MOV   | \$TMP2,\$TMP0               |  | :COPY IT INTO 'TMP0'                               |
| 2715 | 014720 | 042737 | 100100 | 001156 |       | BIC   | #ATA!VV,\$TMP0              |  | :CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 2716 | 014726 | 113760 | 001220 | 000010 |       | MOVB  | PORTB,RHCS2(RO)             |  | :SELECT PORT B.                                    |
| 2717 | 014734 | 016037 | 000012 | 001164 |       | MOV   | RHDS1(RO),\$TMP3            |  | :GET THE DRIVE STATUS REGISTER FROM PORT B.        |
| 2718 | 014742 | 013737 | 001164 | 001160 |       | MOV   | \$TMP3,\$TMP1               |  | :COPY IT INTO 'TMP1'                               |
| 2719 | 014750 | 042737 | 100100 | 001160 |       | BIC   | #ATA!VV,\$TMP1              |  | :CLEAR PORT DEPENDENT BITS FROM THE COPY           |
| 2720 | 014756 | 023737 | 001156 | 001160 |       | CMP   | \$TMP0,\$TMP1               |  | :IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ? |
| 2721 | 014764 | 001006 |        |        |       | BNE   | 69\$                        |  | :BR IF NOT                                         |
| 2722 | 014766 | 005737 | 001156 |        |       | TST   | \$TMP0                      |  | :REGISTERS ARE THE SAME: ARE THEY ZERO ?           |
| 2723 | 014772 | 001045 |        |        |       | BNE   | 71\$                        |  | :BR IF NOT                                         |
| 2724 | 014774 | 104034 |        |        |       | ERROR | 34                          |  | :REPORT DRIVE NOT IN NEUTRAL OR NOT SEIZED         |
| 2725 | 014776 | 000137 | 015162 |        |       | JMP   | 73\$                        |  | :BYPASS THE REST OF THE CHECKS                     |
| 2726 | 015002 | 013737 | 001162 | 001126 | 69\$: | MOV   | \$TMP2,\$BDDAT              |  | :SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE        |
| 2727 | 015010 | 013737 | 001220 | 001226 |       | MOV   | PORTB,PTNBR                 |  | :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 2728 | 015016 | 113760 | 001220 | 000010 |       | MOVB  | PORTB,RHCS2(RO)             |  | :SELECT PORT B.                                    |
| 2729 | 015024 | 005737 | 001156 |        |       | TST   | \$TMP0                      |  | :SEE IF STATUS EQ 0 FROM PORT A.                   |
| 2730 | 015030 | 001414 |        |        |       | BEQ   | 70\$                        |  | :BR IF ZERO                                        |
| 2731 | 015032 | 013737 | 001216 | 001226 |       | MOV   | PORTA,PTNBR                 |  | :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL   |
| 2732 | 015040 | 013737 | 001164 | 001126 |       | MOV   | \$TMP3,\$BDDAT              |  | : 'BAD DATA' FOR ERROR TYPE OUT                    |
| 2733 | 015046 | 113760 | 001216 | 000010 |       | MOVB  | PORTA,RHCS2(RO)             |  | :SELECT PORT A.                                    |
| 2734 | 015054 | 005737 | 001160 |        |       | TST   | \$TMP1                      |  | :SEE IF STATUS EQ ZERO FROM PORT B.                |
| 2735 | 015060 | 001012 |        |        |       | BNE   | 71\$                        |  | :BR IF NOT                                         |
| 2736 | 015062 | 012737 | 177777 | 001242 | 70\$: | MOV   | #-1,RELERR                  |  | :SET 'RELEASE ERROR' INDICATOR                     |
| 2737 | 015070 | 012760 | 000011 | 000000 |       | MOV   | #11,RHCS1(RO)               |  | :CLEAR THE DRIVE                                   |
| 2738 | 015076 | 012760 | 000013 | 000000 |       | MOV   | #13,RHCS1(RO)               |  | :RELEASE THE DRIVE                                 |
| 2739 | 015104 | 104020 |        |        |       | ERROR | 20                          |  | :TYPE ERROR MESSAGE 20                             |
| 2740 | 015106 | 013737 | 001162 | 001126 | 71\$: | MOV   | \$TMP2,\$BDDAT              |  | :LOOK FOR BIT FAILURES WHEN RHDS1 READ             |
| 2741 | 015114 | 013737 | 001216 | 001226 |       | MOV   | PORTA,PTNBR                 |  | :CHANGE PORT NUMBER                                |
| 2742 | 015122 | 023737 | 001124 | 001162 |       | CMP   | \$GDDAT,\$TMP2              |  | :ALL BITS OK ?                                     |
| 2743 | 015130 | 001401 |        |        |       | BEQ   | 72\$                        |  | :BR IF OK FROM PORT A.                             |
| 2744 | 015132 | 104037 |        |        |       | ERROR | 37                          |  | :REPORT ERROR                                      |
| 2745 | 015134 | 013737 | 001164 | 001126 | 72\$: | MOV   | \$TMP3,\$BDDAT              |  | :CHECK RHDS1 FOR BIT FAILURES - FROM PORT B.       |

```

2746 015142 013737 001220 001226 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
2747 015150 023737 001124 001164 CMP %GD0AT,%STMP3 ;SEE IF READ OK FROM PORT B.
2748 015156 001401 BEQ 735 ;BR IF OK
2749 015160 104037 ERROR 37 ;REPORT THE ERROR
2750 015162 000240 NOP
2751 015164 104400 022707 TYPE ,SWTCHN ;RETURN SWITCH TO 'A/B'
2752
2753 ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
2754
2755 015170 105737 001103 TSTB %ERFLG ;DID AN ERROR OCCUR ?
2756 015174 001412 BEQ TST13 ;BR IF NOT
2757 015176 032737 001000 177570 BIT %SW09,SWR ;SEE IF LOOP ON ERROR SET (SWR9=1)
2758 015204 001406 BEQ TST13 ;BR IF NOT
2759 015206 105037 001103 CLRB %ERFLG ;CLEAR THE ERROR FLAG
2760 015212 005037 001170 CLR %STIMES ;CLEAR THE MAX ITERATION COUNT
2761 015216 000177 163666 JMP %$LPERR ;GO TO THE LOOP ADDRESS
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792

```

```

*TEST 13 TEST 'CONTROLLER SELECT' SWITCH LOCKED ON PORT A
*
*TEST THE OPERATION OF THE 'CONTROLLER SELECT' SWITCH (DRIVE CYCLED DOWN).
*
* A. CYCLE THE DRIVE DOWN.
*
* B. SWITCH TO CONTROLLER A POSITION. VERIFY THAT THE DRIVE IS IN
* NEUTRAL AND THAT THE STATUS BITS IN RHDS1, AS READ THROUGH BOTH
* PORTS, ARE CORRECT.
*
* C. SWITCH THE 'CONTROLLER SELECT' SWITCH TO A; CYCLE THE DRIVE UP.
*
* D. WHEN THE DRIVE CYCLES UP, VERIFY THAT 'VV-A IS RESET, AND
* THAT 'ATA-A IS SET.
*
* E. ISSUE A DRIVE CLEAR COMMAND AND A READIN PRESET COMMAND THROUGH
* PORT A.
*
* F. VERIFY THAT THE DRIVE CANNOT BE ACCESSED THROUGH PORT B AND
* 'MED' DOES NOT SET WHEN ATEMPTING TO ACCESS THE DRIVE THROUGH
* PORT B. ATTEMPT TO SET PORT REQUEST BY WRITING 0'S
* INTO RHDS1 THROUGH PORT B.
*
* G. ISSUE A RELEASE COMMAND THROUGH PORT A. VERIFY THAT THE
* DRIVE REMAINS LOCKED ON PORT A.
*

```

```

2793 015222 000004 TST13: SCOPE ;INITIALIZE THE SCOPE HANDLER
2794 015222 000004 TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
2795 015224 005737 001260 BEQ 25 ;BR IF NOT
2796 015230 001406 BPL 15 ;BR IF JUST ENTERED TEST
2797 015232 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
2798 015234 000137 002262 MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2799 015240 012737 177777 001260 15: MOV #13,%STSTNM ;TEST NUMBER
2800 015246 112737 000013 001102 25: MOV
2801 015254 012737 015276 001106 MOV %TEST13,%$LPADR ;LOAD LOOP ON TEST ADDRESS

```

```

2802 015262 012737 015276 001110 MOV #TEST13,$LPERR ;LOAD LOOP ON ERROR ADDRESS
2803 015270 012737 000001 001170 MOV #1,$TIMES ;;DO 1 ITERATION
2804
2805
2806 ;:*****
2807 ;END OF 'SCOPE' SETUP - START OF MAIN TEST
2808
2809 TEST13:
2810 015276 113760 001216 000010 MOV PORTA,RHCS2(RO) ;SELECT PORT A
2811 015304 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2812 015312 104400 023213 TYPE ,CYCLED ;'CYCLE DOWN THE DRIVE'
2813
2814 ;:*****
2815 ;WAIT FOR 'MOL' TO RESET
2816
2817 015316 032760 010000 000012 15: BIT #MOL,RHDS1(RO) ;TEST 'MOL'
2818 015324 001374 BNE 15 ;BR IF IT IS STILL SET
2819 015326 104400 022776 TYPE ,SWTCHA ;SWITCH TO 'A'
2820 015332 104400 023235 TYPE ,CYCLEU ;'CYCLE UP THE DRIVE'
2821
2822 ;:*****
2823 ;WAIT FOR DRIVE TO CYCLE UP AFTER SWITCH CHANGED
2824
2825 015336 032760 010000 000012 25: BIT #MOL,RHDS1(RO) ;TEST 'MOL' AGAIN
2826 015344 001774 BEQ 25 ;BR IF NOT SET
2827
2828 ;:*****
2829 ;DRIVE IS CYCLED UP, CHECK STATUS THROUGH PORT A
2830
2831 015346 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2832 015352 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
2833 015360 012737 000012 001122 MOV #RHDS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2834 015366 060037 001122 ADD RO,$BDDADR ;ADD RH11 BASE ADDRESS
2835 015372 012737 110600 001124 MOV #ATA!MOL!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
2836 015400 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
2837 015406 042737 066077 001156 BIC #111700,$TMP0 ;SAVE SPECIFIED BITS
2838 015414 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
2839 015422 001414 BEQ 645 ;BR IF OK
2840 015424 013737 001126 001166 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
2841 015432 042737 111700 001166 BIC #111700,$TMP4 ;CLEAR THE MASKED BITS
2842 015440 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2843 015446 104021 ERROR 21 ;TYPE MESSAGE 21
2844 015450 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2845 015454 000240 645: NOP
2846
2847 ;:*****
2848 ;SET VOLUME VALID FOR PORT A
2849
2850 015456 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR
2851 015464 012760 000021 000000 MOV #21,RHCS1(RO) ;ISSUE A READIN PRESET
2852 015472 012760 010000 000032 MOV #FMT22,RHOF(RO) ;SET FMT22
2853
2854 ;:*****
2855 ;CHECK THE DRIVE STATUS THROUGH PORT B; VERIFY THAT 'NED' DOES NOT
2856 ; SET WHEN THE DRIVE IS ACCESSED THROUGH PORT B.
2857

```

E05

```

2858 015500 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
2859 015506 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2860 015514 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2861 015520 016037 000012 001126 MOV RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
2862 015526 012737 000012 001122 MOV #RHDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2863 015534 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
2864 015540 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
2865 015544 013737 001126 001156 MOV $BDDAT, $TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'
2866 015552 042737 000077 001156 BIC #1C17700, $TMPD ;SAVE SPECIFIED BITS
2867 015560 023737 001124 001156 CMP $GDDAT, $TMPD ;COMPARE THE BITS
2868 015566 001414 BEQ 65$;BR IF OK
2869 015570 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
2870 015576 042737 177700 001166 BIC #177700, $TMP4 ;CLEAR THE MASKED BITS
2871 015604 053737 001166 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2872 015612 104022 ERROR 22 ;TYPE MESSAGE 22
2873 015614 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2874 015620 000240 NOP
2875 015622 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2876 015626 016037 000010 001126 MOV RHCS2(RO), $BDDAT ;GET CONTENTS OF RHCS2
2877 015634 012737 000010 001122 MOV #RHCS2, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2878 015642 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
2879 015646 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
2880 015652 013737 001126 001156 MOV $BDDAT, $TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'
2881 015660 042737 167777 001156 BIC #1CND, $TMPD ;SAVE SPECIFIED BITS
2882 015666 023737 001124 001156 CMP $GDDAT, $TMPD ;COMPARE THE BITS
2883 015674 001414 BEQ 66$;BR IF OK
2884 015676 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
2885 015704 042737 010000 001166 BIC #ND, $TMP ;CLEAR THE MASKED BITS
2886 015712 053737 001166 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
2887 015720 104023 ERROR 23 ;TYPE MESSAGE 23
2888 015722 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2889 015726 000240 NOP
2890 015730 005060 000012 CLR RHDS1(RO) ;TRY TO SET REQUEST BY WRITING THROUGH
;THE LOCKED OUT PORT (PORT 'B')
2891
2892
2893 ;*****
2894 ;VERIFY THAT DRIVE STAYS LOCKED ON PORT A
2895
2896 015734 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
2897 015742 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2898 015750 012760 000013 000012 MOV #13, RHDS1(RO) ;ISSUE A RELEASE THROUGH PORT A
2899 015756 013737 001216 001230 MOV PORTA, SEIZPT ;ADDRESS OF 'LOCKED ON' PORT
2900 015764 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
2901 015772 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2902 016000 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
2903 016004 016037 000012 001126 MOV RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
2904 016012 012737 000012 001122 MOV #RHDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
2905 016020 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
2906 016024 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
2907 016030 013737 001126 001156 MOV $BDDAT, $TMPD ;MOVE REGISTER CONTENTS TO '$TMPD'
2908 016036 042737 000077 001156 BIC #1C17700, $TMPD ;SAVE SPECIFIED BITS
2909 016044 023737 001124 001156 CMP $GDDAT, $TMPD ;COMPARE THE BITS
2910 016052 001414 BEQ 67$;BR IF OK
2911 016054 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
2912 016062 042737 177700 001166 BIC #177700, $TMP4 ;CLEAR THE MASKED BITS
2913 016070 053737 001166 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT

```

# F05

```

2914 016076 104024 ERROR 24 ;TYPE MESSAGE 24
2915 016100 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
2916 016104 000240 67$: NOP
2917
2918 ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
2919
2920 016106 105737 001103 TSTB $ERFLG ;DID AN ERROR OCCUR
2921 016112 001412 BEQ 3$;BR IF NOT
2922 016114 032737 001000 177570 BIT #SW09,SWR ;SEE IF LOOP ON ERROR (SWR9=1)
2923 016122 001406 BEQ 3$;BR IF NOT
2924 016124 105037 001103 CLRB $ERFLG ;CLEAR THE ERROR FLAG
2925 016130 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
2926 016134 000177 162750 JMP $SLPERR ;GO TO THE LOOP ADDRESS
2927 016140 005737 001260 3$: TST KYBCTL ;IN SINGLE TEST MODE ?
2928 016144 001460 BEQ 6$;BR IF NOT
2929 016146 032737 040000 177570 BIT #SW14,SWR ;LOOP ON TEST ?
2930 016154 001054 BNE 6$;BR IF LOOPING
2931 016156 104400 023213 TYPE ,CYCLED ;TYPE 'CYCLE DOWN'
2932 016162 104400 022707 TYPE ,SWTCHN ;'SWITCH TO A/B'
2933 016166 104400 023235 TYPE ,CYCLEU ;'CYCLE THE DRIVE UP'
2934 016172 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
2935 016200 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2936 016206 032760 010000 000012 4$: BIT #MOL,RHDS1(RO) ;CHECK 'MOL'
2937 016214 001374 BNE 4$;BR IF SET (DRIVE NOT CYCLED DOWN)
2938 016216 032760 010000 000012 5$: BIT #MOL,RHDS1(RO) ;CHECK 'MOL' AGAIN
2939 016224 001774 BEQ 5$;BR IF NOT SET (DRIVE NOT CYCLED UP)
2940
2941 ;*****
2942 ;SET VOLUME VALID FOR BOTH PORTS
2943
2944 016226 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR THROUGH PORT A
2945 016234 012760 000021 000000 MOV #21,RHCS1(RO) ;ISSUE A READIN PRESET THROUGH PORT A
2946 016242 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE PORT A
2947 016250 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
2948 016256 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
2949 016264 012760 000021 000000 MOV #21,RHCS1(RO) ;ISSUE A READIN PRESET THROUGH PORT B
2950 016272 012760 010000 000032 MOV #FMT22,RHOF(RO) ;SET FMT22
2951 016300 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE PORT B
2952 016306 104400 001201 6$: TYPE $CRLF ;CR-LF
2953 016312 000400 BR $TST14 ;;GO TO NEXT TEST
2954
2955 ;*****
2956 *TEST 14 TEST 'CONTROLLER SELECT' SWITCH LOCKED ON PORT B
2957 *
2958 *TEST THE OPERATION OF THE 'CONTROLLER SELECT' SWITCH (DRIVE CYCLED DOWN).
2959 *
2960 * A. CYCLE THE DRIVE DOWN.
2961 *
2962 * B. SWITCH TO CONTROLLER B POSITION. VERIFY THAT THE DRIVE IS IN
2963 * NEUTRAL AND THAT THE STATUS BITS IN RHDS1, AS READ THROUGH BOTH
2964 * PORTS, ARE CORRECT.
2965 *
2966 * C. SWITCH THE 'CONTROLLER SELECT' SWITCH TO B; CYCLE THE DRIVE UP.
2967 *
2968 * D. WHEN THE DRIVE CYCLES UP, VERIFY THAT 'VV-B IS RESET, AND
2969 * THAT 'ATA-B IS SET.

```

# G05

2970  
2971  
2972  
2973  
2974  
2975  
2976  
2977  
2978  
2979  
2980  
2981  
2982  
2983  
2984  
2985  
2986  
2987  
2988  
2989  
2990  
2991  
2992  
2993  
2994  
2995  
2996  
2997  
2998  
2999  
3000  
3001  
3002  
3003  
3004  
3005  
3006  
3007  
3008  
3009  
3010  
3011  
3012  
3013  
3014  
3015  
3016  
3017  
3018  
3019  
3020  
3021  
3022  
3023  
3024  
3025

016314  
016314 000004  
016316 005737 001260  
016322 001406  
016324 100002  
016326 000137 002262  
016332 012737 177777 001260  
016340 112737 000014 001102  
016346 012737 016370 001106  
016354 012737 016370 001110  
016362 012737 000001 001170  
  
016370  
016370 113760 001220 000010  
016376 013737 001220 001226  
016404 104400 023213  
  
016410 032760 010000 000012  
016416 001374  
016420 104400 023061  
016424 104400 023235  
  
016430 032760 010000 000012  
016436 001774

```

;*
;* E. ISSUE A DRIVE CLEAR COMMAND AND A READIN PRESET COMMAND THROUGH
;* PORT B.
;*
;* F. VERIFY THAT THE DRIVE CANNOT BE ACCESSED THROUGH PORT A AND
;* 'NED' DOES NOT SET WHEN ATTEMPTING TO ACCESS THE DRIVE THROUGH
;* PORT A. ATTEMPT TO SET PORT REQUEST BY WRITING 0'S
;* INTO RHDS1 THROUGH PORT A.
;*
;* G. ISSUE A RELEASE COMMAND THROUGH PORT B. VERIFY THAT THE
;* DRIVE REMAINS LOCKED ON PORT B.
;*
;* H. CYCLE THE DRIVE DOWN. CHANGE THE 'CONTROLLER SELECT' SWITCH TO
;* A/B; CYCLE THE DRIVE UP.
;*
;* I. VERIFY THAT BOTH PORTS CAN ACCESS THE DRIVE, THAT BOTH ATTENTION
;* BITS ARE SET, AND THAT BOTH 'VV' BITS ARE RESET.
;*
;*****
;ST14:
;SCOPE ;INITIALIZE THE SCOPE HANDLER
;TST KYBCTL ;PERFORMING ONLY SINGLE TESTS ?
;BEQ 2$;BR IF NOT
;BPL 1$;BR IF JUST ENTERED TEST
;JMP EXEC ;RETURN & GET NEXT TEST NUMBER
;MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
;MOV #14,$TSTNM ;TEST NUMBER
;MOV #TEST14,$LPADR ;LOAD LOOP ON TEST ADDRESS
;MOV #TEST14,$LPERR ;LOAD LOOP ON ERROR ADDRESS
;MOV #1,$TIMES ;;DO 1 ITERATION
;*****
;END OF 'SCOPE' SETUP - START OF MAIN TEST
;*****
;TEST14:
;MOVB PORTB,RHCS2(RO) ;SELECT PORT B
;MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
;TYPE ,CYCLED ;'CYCLE DOWN THE DRIVE'
;*****
;WAIT FOR 'MOL' TO RESET
;*****
;1$: BIT #MOL,RHDS1(RO) ;TEST 'MOL'
;BNE 1$;BR IF IT IS STILL SET
;TYPE ,SWTCHB ;SWITCH TO 'B'
;TYPE ,CYCLEU ;'CYCLE UP THE DRIVE'
;*****
;WAIT FOR DRIVE TO CYCLE UP AFTER SWITCH CHANGED
;*****
;2$: BIT #MOL,RHDS1(RO) ;TEST 'MOL' AGAIN
;BEQ 2$;BR IF NOT SET
;*****
;DRIVE IS CYCLED UP, CHECK STATUS THROUGH PORT B

```



# H05

```
3026
3027 016440 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3028 016444 016037 000012 001126 MOV RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
3029 016452 012737 000012 001122 MOV #RHDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3030 016460 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
3031 016464 012737 110600 001124 MOV #ATA!MOL!DPR!DRY, $GDDAT ;WHAT REGISTER SHOULD BE
3032 016472 013737 001126 001156 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3033 016500 042737 066077 001156 BIC #1C111700, $TMP0 ;SAVE SPECIFIED BITS
3034 016506 023737 001124 001156 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
3035 016514 001414 BEQ 64$;BR IF OK
3036 016516 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
3037 016524 042737 111700 001166 BIC #111700, $TMP4 ;CLEAR THE MASKED BITS
3038 016532 053737 001166 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3039 016540 104021 ERROR 21 ;TYPE MESSAGE 21
3040 016542 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3041 016546 000240 64$: NOP
3042
3043 ;*****
3044 ;SET VOLUME VALID FOR PORT B
3045
3046 016550 012760 000011 000000 MOV #11, RHCS1(RO) ;ISSUE A DRIVE CLEAR
3047 016556 012760 000021 000000 MOV #21, RHCS1(RO) ;ISSUE A READIN PRESET
3048 016564 012760 010000 000032 MOV #FMT22, RHOF(RO) ;SET FMT22
3049
3050 ;*****
3051 ;CHECK THE DRIVE STATUS THROUGH PORT A; VERIFY THAT 'NED' DOES NOT
3052 ;SET WHEN THE DRIVE IS ACCESSED THROUGH PORT A.
3053
3054 016572 113760 001216 000010 MOV PORTA, RHCS2(RO) ;SELECT PORT A
3055 016600 013737 001216 001226 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3056 016606 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3057 016612 016037 000012 001126 MOV RHDS1(RO), $BDDAT ;GET CONTENTS OF RHDS1
3058 016620 012737 000012 001122 MOV #RHDS1, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3059 016626 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
3060 016632 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
3061 016636 013737 001126 001156 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3062 016644 042737 000077 001156 BIC #1C177700, $TMP0 ;SAVE SPECIFIED BITS
3063 016652 023737 001124 001156 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
3064 016660 001414 BEQ 65$;BR IF OK
3065 016662 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
3066 016670 042737 177700 001166 BIC #177700, $TMP4 ;CLEAR THE MASKED BITS
3067 016676 053737 001166 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3068 016704 104022 ERROR 22 ;TYPE MESSAGE 22
3069 016706 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3070 016712 000240 65$: NOP
3071 016714 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3072 016720 016037 000010 001126 MOV RHCS2(RO), $BDDAT ;GET CONTENTS OF RHCS2
3073 016726 012737 000010 001122 MOV #RHCS2, $B0ADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3074 016734 060037 001122 ADD RO, $B0ADR ;ADD RH11 BASE ADDRESS
3075 016740 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
3076 016744 013737 001126 001156 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3077 016752 042737 167777 001156 BIC #1CNED, $TMP0 ;SAVE SPECIFIED BITS
3078 016760 023737 001124 001156 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
3079 016766 001414 BEQ 66$;BR IF OK
3080 016770 013737 001126 001166 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
3081 016776 042737 010000 001166 BIC #NED, $TMP4 ;CLEAR THE MASKED BITS
```



```

3082 017004 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3083 017012 104023 ERROR 23 ;TYPE MESSAGE 23
3084 017014 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3085 017020 000240 66$: NOP
3086 017022 005060 000012 CLR RHDS1(RO) ;TRY TO SET REQUEST BY WRITING THROUGH
 ;THE LOCKED OUT PORT (PORT 'A')
3087
3088
3089
3090 ;:*****X*****
3091 ;VERIFY THAT DRIVE STAYS LOCKED ON PORT B
3092 017026 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
3093 017034 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3094 017042 012760 000013 000012 MOV #13,RHDS1(RO) ;ISSUE A RELEASE THROUGH PORT B
3095 017050 013737 001220 001230 MOV PORTB,SEIZPT ;ADDRESS OF 'LOCKED ON' PORT
3096 017056 113760 001216 000010 MOVB PORTA,RHCS2(RO) ;SELECT PORT A
3097 017064 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3098 017072 005037 001236 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
3099 017076 016037 000012 001126 MOV RHDS1(RO),$BDDAT ;GET CONTENTS OF RHDS1
3100 017104 012737 000012 001122 MOV #RHDS1,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
3101 017112 060037 001122 ADD RO,$BADDR ;ADD RH11 BASE ADDRESS
3102 017116 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
3103 017122 013737 001126 001156 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
3104 017130 042737 000077 001156 BIC #177700,$TMP0 ;SAVE SPECIFIED BITS
3105 017136 023737 001124 001156 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
3106 017144 001414 BEQ 67$;BR IF OK
3107 017146 013737 001126 001166 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
3108 017154 042737 177700 001166 BIC #177700,$TMP4 ;CLEAR THE MASKED BITS
3109 017162 053737 001166 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
3110 017170 104024 ERROR 24 ;TYPE MESSAGE 24
3111 017172 005137 001236 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
3112 017176 000240 67$: NOP
3113
3114 ;IF ERROR OCCURED, CHECK FOR LOOP ON TEST
3115
3116 017200 105737 001103 TSTB $ERFLG ;DID AN ERROR OCCUR
3117 017204 001412 BEQ 3$;BR IF NOT
3118 017206 032737 001000 177570 BIT #SW09,$SWR ;SEE IF LOOP ON ERROR (SWR9=1)
3119 017214 001406 BEQ 3$;BR IF NOT
3120 017216 105037 001103 CLRB $ERFLG ;CLEAR THE ERROR FLAG
3121 017222 005037 001170 CLR $TIMES ;CLEAR THE MAX ITERATION COUNT
3122 017226 000177 161656 JMP @SLPERR ;GO TO THE LOOP ADDRESS
3123 017232 032737 040000 177570 3$: BIT #SW14,$SWR ;LOOP ON TEST ?
3124 017240 001054 BNE 6$;BR IF LOOPING
3125 017242 104400 023213 TYPE ,CYCLED ;TYPE 'CYCLE DOWN'
3126 017246 104400 022707 TYPE ,SWTCHN ;'SWITCH TO A/B'
3127 017252 104400 023235 TYPE ,CYCLEU ;'CYCLE THE DRIVE UP'
3128 017256 113760 001220 000010 MOVB PORTB,RHCS2(RO) ;SELECT PORT B
3129 017264 013737 001220 001226 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3130 017272 032760 010000 000012 4$: BIT #MOL,RHDS1(RO) ;CHECK 'MOL'
3131 017300 001374 BNE 4$;BR IF SET (DRIVE NOT CYCLED DOWN)
3132 017302 032760 010000 000012 5$: BIT #MOL,RHDS1(RO) ;CHECK 'MOL' AGAIN
3133 017310 001774 BEQ 5$;BR IF NOT SET (DRIVE NOT CYCLED UP)
3134
3135 ;:*****X*****
3136 ;SET VOLUME VALID FOR BOTH PORTS
3137

```

```

3138 017312 012760 000011 000000 MOV #11,RHCS1(RO) ;ISSUE A DRIVE CLEAR THROUGH PORT B
3139 017320 012760 000021 000000 MOV #21,RHCS1(RO) ;ISSUE A READIN PRESET THROUGH PORT B
3140 017326 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE PORT B
3141 017334 113760 001216 000010 MOV PORTA,RHCS2(RO) ;SELECT PORT A
3142 017342 013737 001216 001226 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
3143 017350 012760 000021 000000 MOV #21,RHCS1(RO) ;ISSUE A READIN PRESET THRU'GH PORT A
3144 017356 012760 010000 000032 MOV #FMT22,RHOF(RO) ;SET FMT22
3145 017364 012760 000013 000000 MOV #13,RHCS1(RO) ;RELEASE PORT A
3146 017372 000137 017610 6S: JMP $EOP ;GO TO THE END OF PASS ROUTINE
3147
3148 ;*****
3149
3150 .SBTTL *** SUBROUTINES ***
3151
3152 ;*****
3153
3154 ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
3155
3156
3157 017376 012737 017446 000004 CKCLK: MOV #CKCLK1,@#ERRVEC ;SET UP VECTOR FOR CLOCK CHECK
3158 017404 005037 000006 CLR @#ERRVEC+2 ;NEW PSW
3159 017410 005777 161570 TST @SLKCSR ;CHECK FOR KW11-P
3160 017414 013701 001210 MOV SLPVEC,R1 ;KW11-P VECTOR ADDRESS
3161 017420 012721 017530 MOV #CLOCK,(R1)+ ;SET UP KW11-P VECTOR
3162 017424 012711 000300 MOV #300,(R1) ;PSW - PRI 6
3163 017430 012777 177777 161550 MOV #-1,@SLKCSB ;LOAD COUNTER BUFFER WITH 1'S
3164 017436 012777 000135 161540 MOV #135,@SLKCSR ;SET CLOCK - CNT UP, 16MS, CONT INT
3165 017444 000425 BR CKCLK3
3166 017446 062706 000004 CKCLK1: ADD #4,SP ;RESTORE THE STACK POINTER
3167 017452 012737 017510 000004 MOV #CKCLK2,@#ERRVEC ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
3168 017460 005777 161526 TST @SLKS ;LOOK FOR KW11-L
3169 017464 013701 001214 MOV SLLVEC,R1 ;KW11-L VECTOR ADDRESS
3170 017470 012721 017530 MOV #CLOCK,(R1)+ ;SET UP KW11-L VECTOR
3171 017474 012711 000300 MOV #300,(R1) ;PSW - PRI 6
3172 017500 012777 000100 161504 MOV #100,@SLKS ;SET KW11-L INTERRUPT
3173 017506 000404 BR CKCLK3
3174 017510 062706 000004 CKCLK2: ADD #4,SP ;RESTORE THE STACK POINTER
3175 017514 062716 000002 ADD #2,(SP) ;INCREMENT RETURN, NO CLOCK
3176 017520 012737 000006 000004 CKCLK3: MOV #6,@#ERRVEC ;RESTORE THE ERROR VECTOR
3177 017526 000207 RTS PC
3178
3179 ;ROUTINE TO COUNT CLOCK TICKS
3180
3181 017530 062737 000021 001244 CLOCK: ADD #17.,TIME ;ADD 17 MS TO ELAPSED TIME COUNTER
3182 017536 005737 001246 TST WATCH ;IS WATCH ALREADY ZERO ?
3183 017542 001406 BEQ IS ;BR IF IT IS
3184 017544 162737 000021 001246 SUB #17.,WATCH ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
3185 017552 100002 BPL IS ;BR IF NOT MINUS
3186 017554 005037 001246 CLR WATCH ;CLEAR WATCH DOG COUNTER
3187 017560 000002 1S: RTI ;RETURN
3188
3189 ;ROUTINE TO CALCULATE + 25% TIME TOLERANCE VALUES
3190
3191 017562 005746 TOLER: TST -(SP) ;MAKE ROOM ON THE STACK
3192 017564 016616 000002 MOV 2(SP),(SP) ;SAVE STACK
3193 017570 013546 MOV @R5+,-(SP) ;GET TIME VALUE

```

```

3194 017572 011666 000004 MOV (SP),4(SP) ;MOVE TIME VALUE
3195 017576 006216 ASR (SP) ;DIVIDE BY 2
3196 017600 006216 ASR (SP) ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
3197 017602 062666 000002 ADD (SP)+,2(SP) ;CALCULATE UPPER LIMIT FOR TIMEOUT
3198 017606 000205 RTS R5 ;RETURN WITH TOLERANCES ON THE STACK
3199
3200 ;*****
3201 .SBTTL 'SYSMAC' UTILITY ROUTINES
3202
3203 ;*****
3204
3205 ;*****
3206
3207 .SBTTL END OF PASS ROUTINE
3208
3209 ;*INCREMENT THE PASS NUMBER ($PASS)
3210 ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
3211 ;*TYPE "END PASS #XXXX" (WHERE XXXX IS A DECIMAL NUMBER)
3212 ;*IF THERES A MONITOR GO TO IT
3213 ;*IF THERE ISN'T JUMP TO TST1AA
3214
3215 SEOP:
3216 017610 000004 SCOPE
3217 017610 005737 001260 TST KYBCTL ;ENTERED TEST VIA KEYBOARD COMMAND ?
3218 017612 001402 BEQ .+6 ;BR IF NOT
3219 017616 000137 002262 JMP EXEC ;RETURN TO KEYBOARD CONTROL
3220 017620 005037 001102 CLR $TSTNM ;ZERO THE TEST NUMBER
3221 017624 005037 001170 CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
3222 017630 005237 001100 INC $PASS ;INCREMENT THE PASS NUMBER
3223 017634 042737 100000 001100 BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
3224 017640 005327 DEC (PC)+ ;LOOP?
3225 017646 000001 SEOPCT: .WORD 1
3226 017650 003031 BGT $DOAGN ;YES
3227 017652 012737 MOV (PC)+,2(PC)+ ;RESTORE COUNTER
3228 017654 000001 SENDCT: .WORD 1
3229 017656 104400 017742 $EOPCT TYPE $SENDMG ;TYPE "END PASS #"
3230 017660 013746 001100 MOV $PASS,-(SP) ;SAVE $PASS FOR TYPEOUT
3231 017662 104410 017757 TYPDS TYPE $NULL ;GO TYPE--DECIMAL ASCII WITH SIGN
3232 017666 104400 000042 $GET42: MOV #42,R0 ;TYPE A NULL CHARACTER
3233 017672 001414 017726 BEQ $DOAGN ;GET MONITOR ADDRESS
3234 017674 022700 177777 000002 CMP #SENDAD,R0 ;BRANCH IF NO MONITOR
3235 017700 004710 000240 BNE $SENDAD ;IS MONITOR ACT11?
3236 017704 000240 000240 CMP #-1,2(R0) ;NO--BRANCH
3237 017706 000240 000240 BNE $DOAGN ;YES--IS THIS THE LAST PASS?
3238 017712 000240 000240 RESET ;NO--MAKE ANOTHER PASS
3239 017714 000240 000240 JSR PC,(R0) ;CLEAR THE WORLD
3240 017722 000240 000240 NOP ;GO TO MONITOR
3241 017724 000240 000240 NOP ;SAVE ROOM
3242 017726 000137 002554 NOP ;FOR
3243 017730 000137 047105 020104 JMP #TST1AA ;ACT11
3244 017732 000137 051523 021440 .ASCIZ <15><12>/END PASS #/ ;RETURN
3245 017734 000137
3246 017736 000137
3247 017742 000137
3248 017750 000137
3249 017756 000137

```

```

3250 017757 377 377 000 $ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
3251 ;*****
3252
3253 .SBTTL SCOPE HANDLER ROUTINE
3254
3255 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
3256 ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
3257 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
3258 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3259 ;*SW14=1 LOOP ON TEST
3260 ;*SW11=1 INHIBIT ITERATIONS
3261 ;*CALL
3262 ;* SCOPE ;;SCOPE=IOT
3263
3264 $$SCOPE:
3265 017762 006137 177570 ROL @#SWR ;;LOOP ON PRESENT TEST?
3266 017766 100455 BMI $OVER ;;YES IF SW14=1
3267 ;*****START OF CODE FOR THE XOR TESTER*****
3268 017770 000416 $XTSTR: BR 6$;;IF RUNNING ON THE "XOR" TESTER CHANGE
3269 ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
3270 017772 013746 000004 MOV @#ERRVEC, -(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
3271 017776 012737 020016 000004 MOV $$, @#ERRVEC ;;SET FOR TIMEOUT
3272 020004 005737 177060 TST @#177060 ;;TIME OUT ON XOR?
3273 020010 012637 000004 MOV (SP)+, @#ERRVEC ;;RESTORE THE ERROR VECTOR
3274 020014 000436 BR $$VLAD ;;GO TO THE NEXT TEST
3275 020016 022626 $$: CMP (SP)+, (SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
3276 020020 012637 000004 MOV (SP)+, @#ERRVEC ;;RESTORE THE ERROR VECTOR
3277 020024 000436 BR $OVER ;;LOOP ON THE PRESENT TEST
3278 020026 6$;*****END OF CODE FOR THE XOR TESTER*****
3279 020026 105737 001103 2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
3280 020032 001404 BEQ 3$;;BR IF NO
3281 020034 105037 001103 4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG
3282 020040 005037 001170 CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
3283 020044 032737 004000 177570 3$: BIT #BIT11, @#SWR ;;INHIBIT ITERATIONS?
3284 020052 001011 BNE 1$;;BR IF YES
3285 020054 005737 001100 TST $PASS ;;IF FIRST PASS OF PROGRAM
3286 020060 001406 BEQ 1$;;INHIBIT ITERATIONS
3287 020062 005237 001104 INC $ICNT ;;INCREMENT ITERATION COUNT
3288 020066 023737 001170 001104 CMP $TIMES, $ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
3289 020074 002012 BGE $OVER ;;BR IF MORE ITERATION REQUIRED
3290 020076 012737 000001 001104 1$: MOV #1, $ICNT ;;REINITIALIZE THE ITERATION COUNTER
3291 020104 013737 020136 001170 MOV $MXCNT, $TIMES ;;SET NUMBER OF ITERATIONS TO DO
3292 020112 105237 001102 $$VLAD: INCB $TSTNM ;;COUNT TEST NUMBERS
3293 020116 011637 001106 MOV (SP), $LPADR ;;SAVE SCOPE LOOP ADDRESS
3294 020122 013737 001102 177570 $OVER: MOV $TSTNM, @#DISPLAY ;;DISPLAY TEST NUMBER
3295 020130 013716 001106 MOV $LPADR, (SP) ;;FUDGE RETURN ADDRESS
3296 020134 000002 RTI ;;FIXES PS
3297 020136 000004 $MXCNT: 4 ;;MAX. NUMBER OF ITERATIONS
3298 ;*****
3299
3300 .SBTTL ERROR HANDLER ROUTINE
3301
3302 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3303 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3304 ;*AND GO TO $ERRTYP ON ERROR
3305 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:

```

```

3306 ;*SW15=1 HALT ON ERROR
3307 ;*SW13=1 INHIBIT ERROR TYPEOUTS
3308 ;*SW10=1 BELL ON ERROR
3309 ;*CALL
3310 ;* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3311
3312 020140 SERROR:
3313 020140 113737 001102 001234 MOV B $TSTNM,TSTNUM
3314 020146 105237 001103 7$: INC B $ERFLG ;;SET THE ERROR FLAG
3315 020152 001775 BEQ 7$;;DON'T LET THE FLAG GO TO ZERO
3316 020154 013737 001102 177570 MOV $TSTNM,@#DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
3317 020162 032737 002000 177570 BIT #BIT10,@#SWR ;;BELL ON ERROR?
3318 020170 001402 BEQ 1$;;NO - SKIP
3319 020172 104400 001174 TYPE $BELL ;;RING BELL
3320 020176 005237 001112 1$: INC $ERTTL ;;COUNT THE NUMBER OF ERRORS
3321 020202 011637 001116 MOV (SP),$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
3322 020206 162737 000002 001116 SUB #2,$ERRPC
3323 020214 117737 160676 001114 MOV B @#ERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
3324 020222 032737 020000 177570 BIT #BIT13,@#SWR ;;SKIP TYPEOUT IF SET
3325 020230 001004 BNE 2$;;SKIP TYPEOUTS
3326 020232 004737 020254 JSR PC,@#$ERRPC ;;GO TO USER ERROR ROUTINE
3327 020236 104400 001201 TYPE $CRLF
3328 020242 005737 177570 2$: TST @#SWR ;;HALT ON ERROR
3329 020246 100001 BPL 3$;;SKIP IF CONTINUE
3330 020250 000000 HALT ;;HALT ON ERROR!
3331
3332 020252 000002 3$: RTI ;;RETURN
3333 ;*****
3334
3335 .SBTTL ERROR MESSAGE TYPEOUT ROUTINE
3336
3337 ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
3338 ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS FROM THE "ERROR TABLE" ($ERRTB),
3339 ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
3340
3341 020254 SERRTYP:
3342 020254 104400 001201 TYPE $CRLF ;; "CARRIAGE RETURN" & "LINE FEED"
3343 020260 010046 MOV RO,-(SP) ;;SAVE RO
3344 020262 005000 CLR RO ;;PICKUP THE ITEM INDEX
3345 020264 153700 001114 BIS B @#$ITEMB,RO
3346 020270 001004 BNE 1$;; IF ITEM NUMBER IS ZERO, JUST
3347 TYPE THE PC OF THE ERROR
3348 020272 013746 001116 MOV $ERRPC,-(SP) ;;SAVE $ERRPC FOR TYPEOUT
3349 ERROR ADDRESS
3350 020276 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3351 020300 000445 BR 10$;;GET OUT
3352 020302 005300 1$: DEC RO ;;ADJUST THE INDEX SO THAT IT WILL
3353 020304 006300 ASL RO ;; WORK FOR THE ERROR TABLE
3354 020306 006300 ASL RO
3355 020310 006300 ASL RO
3356 020312 062700 001266 ADD #SERRTB,RO ;;FORM TABLE POINTER
3357 020316 012037 020326 MOV (RO)+,2$;;PICKUP "ERROR MESSAGE" POINTER
3358 020322 001404 BEQ 3$;;SKIP TYPEOUT IF NO POINTER
3359 020324 104400 TYPE ;;TYPE THE "ERROR MESSAGE"
3360 020326 000000 2$: .WORD 0 ;; "ERROR MESSAGE" POINTER GOES HERE
3361 020330 104400 001201 TYPE $CRLF ;; "CARRIAGE RETURN" & "LINE FEED"

```

```

3362 020334 012037 020344 3$: MOV (R0)+,4$;; PICKUP "DATA HEADER" POINTER
3363 020340 001404 BEQ 5$;; SKIP TYPEOUT IF 0
3364 020342 104400 TYPE ;; TYPE THE "DATA HEADER"
3365 020344 000000 4$: .WORD 0 ;; "DATA HEADER" POINTER GOES HERE
3366 020346 104400 0C1201 TYPE $CF - ;; "CARRIAGE RETURN" & "LINE FEED"
3367 020352 010146 5$: MOV R1,-(R0) ;; SAVE R1
3368 020354 012001 MOV (R0)-,R1 ;; PICKUP "DATA TABLE" POINTER
3369 020356 001415 BEQ 9$;; BR IF NO DATA TO BE TYPED
3370 020360 012000 MOV (R0)+,R0 ;; PICKUP "DATA FORMAT" POINTER
3371 020362 105720 6$: TSTB (R0)+ ;; "CARRIAGE RETURN" & "LINE FEED"
3372 020364 001003 BNE 7$;; BR IF DECIMAL
3373 020366 013146 MOV @ (R1)+,-(SP) ;; SAVE @ (R1)+ FOR TYPEOUT
3374 020370 104402 TYPOC ;; GO TYPE--OCTAL ASCII (ALL DIGITS)
3375 020372 000402 BR 8$
3376 020374 7$:
3377 020374 013146 MOV @ (R1)+,-(SP) ;; SAVE @ (R1)+ FOR TYPEOUT
3378 020376 104410 TYPDS ;; GO TYPE--DECIMAL ASCII WITH SIGN
3379 020400 005711 8$: TST (R1) ;; IS THERE ANOTHER NUMBER?
3380 020402 001403 BEQ 9$;; BR IF NO
3381 020404 104400 020424 TYPE ,11$;; TYPE TWO(2) SPACES
3382 020410 000764 BR 6$;; LOOP
3383
3384 020412 012601 9$: MOV (SP)+,R1 ;; RESTORE R1
3385 020414 012600 10$: MOV (SP)+,R0 ;; RESTORE R0
3386 020416 104400 001201 TYPE $CRLF ;; "CARRIAGE RETURN" & "LINE FEED"
3387 020422 000207 RTS PC ;; RETURN
3388 020424 020040 000 11$: .ASCIZ / / ;; TWO(2) SPACES
3389 020430 020430 .EVEN
3390
;*****
3391
3392 .SBTTL TYPE ROUTINE
3393
3394 ;; *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
3395 ;; *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
3396 ;; *NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
3397 ;; *NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
3398 ;; *NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
3399 ;; *
3400 ;; *CALL:
3401 ;; *1) USING A TRAP INSTRUCTION
3402 ;; * TYPE ,MESADR ;; MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
3403 ;; *OR
3404 ;; * TYPE
3405 ;; * MESADR
3406 ;; *
3407 ;; *2) USING A JSR INSTRUCTION
3408 ;; * MOV PS,-(SP) ;; PUSH PROCESSOR STATUS WORD ON THE STACK
3409 ;; * JSR PC,$TYPE ;; CALL TYPE ROUTINE
3410 ;; * MESADDR ;; FIRST ADDRESS OF MESSAGE
3411
3412 020430 105737 001151 $TYPE: TSTB $TPFLG ;; IS THERE A TERMINAL?
3413 020434 100002 BPL 1$;; BR IF YES
3414 020436 000000 HALT ;; HALT HERE IF NO TERMINAL
3415 020440 000407 BR 3$;; LEAVE
3416 020442 010046 1$: MOV R0,-(SP) ;; SAVE R0
3417 020444 017600 000002 MOV @2(SP),R0 ;; GET ADDRESS OF ASCIZ STRING

```

```

3418 020450 112046 2S: MOVB (RO)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
3419 020452 001005 BNE 4S ;; BR IF IT ISN'T THE TERMINATOR
3420 020454 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
3421 020456 012600 MOV (SP)+,R0 ;; RESTORE R0
3422 020460 062716 0C0002 3S: ADD #2,(SP) ;; ADJUST RETURN PC
3423 020464 000002 RTI ;; RETURN
3424 020466 004737 020520 4S: JSR PC,$TYPEC ;; GO TYPE THIS CHARACTER
3425 020472 123726 0C1150 5S: CMPB $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
3426 020476 001364 BNE 2S ;; IF NO GO GET NEXT CHAR.
3427 020500 013746 0C1146 MOV $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
3428 ;; AND THE NULL CHAR.
3429 020504 105366 000001 6S: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
3430 020510 002770 BLT 5S ;; BR IF NO--GO POP THE NULL OFF OF STACK
3431 020512 004737 020520 JSR PC,$TYPEC ;; GO TYPE A NULL
3432 020516 000772 BR 6S ;; LOOP
3433 020520 105777 160416 $TYPEC: TSTB $STPS ;; WAIT UNTIL PRINTER IS READY
3434 020524 100375 BPL $TYPEC
3435 020526 116677 000002 160410 MOVB 2(SP),$STPB ;; LOAD CHAR TO BE TYPED INTO DATA REG.
3436 020534 C00207 RTS PC
;*****
.SBTTL BINARY TO OCTAL (ASCII) AND TYPE
; *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
; *OCTAL (ASCII) NUMBER AND TYPE IT.
; *$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
; *CALL:
; * MOV NUM,-(SP) ;; NUMBER TO BE TYPED
; * TYPOS ;; CALL FOR TYPEOUT
; * .BYTE N ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
; * .BY ? M ;; M=1 OR 0
; * ;; 1=TYPE LEADING ZEROS
; * ;; 0=SUPPRESS LEADING ZEROS
; *
; *$STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
; *$TYPOS OR $TYPOC
; *CALL:
; * MOV NUM,-(SP) ;; NUMBER TO BE TYPED
; * TYPON ;; CALL FOR TYPEOUT
; *
; *$STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
; *CALL:
; * MOV NUM,-(SP) ;; NUMBER TO BE TYPED
; * TYPOC ;; CALL FOR TYPEOUT
3463 020536 017646 000000 $TYPOS: MOV 2(SP),-(SP) ;; PICKUP THE MODE
3464 020542 116637 000001 020761 MOVB 1(SP),$OFILL ;; LOAD ZERO FILL SWITCH
3465 020550 112637 020763 MOVB (SP)+,$SOMODE+1 ;; NUMBER OF DIGITS TO TYPE
3466 020554 062716 000002 ADD #2,(SP) ;; ADJUST RETURN ADDRESS
3467 020560 000406 BR $STYPON
3468 020562 112737 000001 020761 $STYPOC: MOVB #1,$OFILL ;; SET THE ZERO FILL SWITCH
3469 020570 112737 000006 020763 MOVB #6,$SOMODE+1 ;; SET FOR SIX(6) DIGITS
3470 020576 112737 000005 020760 $STYPON: MOVB #5,$SOCNT ;; SET THE ITERATION COUNT
3471 020604 010346 MOV R3,-(SP) ;; SAVE R3
3472 020606 010446 MOV R4,-(SP) ;; SAVE R4
3473 020610 010546 MOV R5,-(SP) ;; SAVE R5

```



```

3474 020612 113704 020763 MOVB $OMODE+1,R4 ;;GET THE NUMBER OF DIGITS TO TYPE
3475 020616 005404 NEG R4
3476 020620 062704 000006 ADD #6,R4 ;;SUBTRACT IT FOR MAX. ALLOWED
3477 020624 110437 020762 MOVB R4,$OMODE ;;SAVE IT FOR USE
3478 020630 113704 020761 MOVB $OFILL,R4 ;;GET THE ZERO FILL SWITCH
3479 020634 016605 000012 MOV !2(SP),R5 ;;PICKUP THE INPUT NUMBER
3480 020640 005003 CLR R3 ;;CLEAR THE OUTPUT WORD
3481 020642 006105 1$: ROL R5 ;;ROTATE MSB INTO "C"
3482 020644 000404 BR 3$;;GO DO MSB
3483 020646 006105 2$: ROL R5 ;;FORM THIS DIGIT
3484 020650 006105 ROL R5
3485 020652 006105 ROL R5
3486 020654 010503 MOV R5,R3
3487 020656 006103 3$: ROL R3 ;;GET LSB OF THIS DIGIT
3488 020660 105337 020762 DECB $OMODE ;;TYPE THIS DIGIT?
3489 020664 100016 BPL 7$;;BR IF NO
3490 020666 042703 177770 BIC #177770,R3 ;;GET RID OF JUNK
3491 020672 001002 BNE 4$;;TEST FOR 0
3492 020674 005704 TST R4 ;;SUPPRESS THIS 0?
3493 020676 001403 BEQ 5$;;BR IF YES
3494 020700 005204 4$: INC R4 ;;DON'T SUPPRESS ANYMORE 0'S
3495 020702 052703 000060 BIS #'0,R3 ;;MAKE THIS DIGIT ASCII
3496 020706 052703 000040 5$: BIS #' ,R3 ;;MAKE ASCII IF NOT ALREADY
3497 020712 110337 020756 MOVB R3,$$;;SAVE FOR TYPING
3498 020716 104400 020756 TYPE 8$;;GO TYPE THIS DIGIT
3499 020722 105337 020760 7$: DECB $OCNT ;;COUNT BY 1
3500 020726 003347 BGT 2$;;BR IF MORE TO DO
3501 020730 002402 BLT 6$;;BR IF DONE
3502 020732 005204 INC R4 ;;INSURE LAST DIGIT ISN'T A BLANK
3503 020734 000744 BR 2$;;GO DO THE LAST DIGIT
3504 020736 012605 6$: MOV (SP)+,R5 ;;RESTORE R5
3505 020740 012604 MOV (SP)+,R4 ;;RESTORE R4
3506 020742 012603 MOV (SP)+,R3 ;;RESTORE R3
3507 020744 016666 000002 000004 MOV 2(SP),4(SP) ;;SET THE STACK FOR RETURNING
3508 020752 012616 MOV (SP)+,(SP)
3509 020754 000002 RTI
3510 020756 8$: .BYTE 0 ;;RETURN
3511 020757 .BYTE 0 ;;STORAGE FOR ASCII DIGIT
3512 020760 .BYTE 0 ;;TERMINATOR FOR TYPE ROUTINE
3513 020761 .BYTE 0 ;;OCTAL DIGIT COUNTER
3514 020762 000000 .WORD 0 ;;ZERO FILL SWITCH
3515 ;*****
3516 .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
3517
3518
3519 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
3520 ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
3521 ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
3522 ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
3523 ;*REPLACED WITH SPACES.
3524 ;*CALL:
3525 ;* MOV NUM,-(SP) ;;PUT THE BINARY NUMBER ON THE STACK
3526 ;* TYPDS ;;GO TO THE ROUTINE
3527
3528 $TYPDS:
3529 MOV R0,-(SP) ;;PUSH R0 ON STACK

```

```

3530 020766 010146 MOV R1,-(SP) ;; PUSH R1 ON STACK
3531 020770 010246 MOV R2,-(SP) ;; PUSH R2 ON STACK
3532 020772 010346 MOV R3,-(SP) ;; PUSH R3 ON STACK
3533 020774 010546 MOV R5,-(SP) ;; PUSH R5 ON STACK
3534 020776 012746 020200 MOV #20200,-(SP) ;; SET BLANK SWITCH AND SIGN
3535 021002 016605 000020 MOV 20(SP),R5 ;; GET THE INPUT NUMBER
3536 021006 100004 BPL 1$;; BR IF INPUT IS POS.
3537 021010 005405 NEG R5 ;; MAKE THE BINARY NUMBER POS.
3538 021012 112766 000055 000001 MOVB #'-,1(SP) ;; MAKE THE ASCII NUMBER NEG.
3539 021020 005000 CLR R0 ;; ZERO THE CONSTANTS INDEX
3540 021022 012703 021200 MOV #SDBLK,R3 ;; SETUP THE OUTPUT POINTER
3541 021026 112723 000040 MOVB #' ,(R3)+ ;; SET THE FIRST CHARACTER TO A BLANK
3542 021032 005002 CLR R2 ;; CLEAR THE BCD NUMBER
3543 021034 016001 021170 MOV $DTBL(R0),R1 ;; GET THE CONSTANT
3544 021040 160105 SUB R1,R5 ;; FORM THIS BCD DIGIT
3545 021042 002402 BLT 4$;; BR IF DONE
3546 021044 005202 INC R2 ;; INCREASE THE BCD DIGIT BY 1
3547 021046 000774 BR 3$;;
3548 021050 060105 ADD R1,R5 ;; ADD BACK THE CONSTANT
3549 021052 005702 TST R2 ;; CHECK IF BCD DIGIT=0
3550 021054 001002 BNE 5$;; FALL THROUGH IF 0
3551 021056 105716 TSTB (SP) ;; STILL DOING LEADING 0'S?
3552 021060 100407 BMI 7$;; BR IF YES
3553 021062 106316 ASLB (SP) ;; MSD?
3554 021064 103003 BCC 6$;; BR IF NO
3555 021066 116663 000001 177777 MOVB 1(SP),-1(R3) ;; YES--SET THE SIGN
3556 021074 052702 000060 6$: BIS #'0,R2 ;; MAKE THE BCD DIGIT ASCII
3557 021100 052702 000040 7$: BIS #' ,R2 ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
3558 021104 110223 MOVB R2,(R3)+ ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
3559 021106 005720 TST (R0)+ ;; JUST INCREMENTING
3560 021110 020027 000010 CMP R0,#10 ;; CHECK THE TABLE INDEX
3561 021114 002746 BLT 2$;; GO DO THE NEXT DIGIT
3562 021116 003002 BGT 8$;; GO TO EXIT
3563 021120 010502 MOV R5,R2 ;; GET THE LSD
3564 021122 000764 BR 6$;; GO CHANGE TO ASCII
3565 021124 105726 8$: TSTB (SP)+ ;; WAS THE LSD THE FIRST NON-ZERO?
3566 021126 100003 BPL 9$;; BR IF NO
3567 021130 116663 177777 177776 9$: MOVB -1(SP),-2(R3) ;; YES--SET THE SIGN FOR TYPING
3568 021136 105013 CLRB (R3) ;; SET THE TERMINATOR
3569 021140 012605 MOV (SP)+,R5 ;; POP STACK INTO R5
3570 021142 012603 MOV (SP)+,R3 ;; POP STACK INTO R3
3571 021144 012602 MOV (SP)+,R2 ;; POP STACK INTO R2
3572 021146 012601 MOV (SP)+,R1 ;; POP STACK INTO R1
3573 021150 012600 MOV (SP)+,R0 ;; POP STACK INTO R0
3574 021152 104400 021200 TYPE $SDBLK ;; NOW TYPE THE NUMBER
3575 021156 016666 000002 000004 MOV 2(SP),4(SP) ;; ADJUST THE STACK
3576 021164 012616 MOV (SP)+,(SP) ;;
3577 021166 000002 RTI ;; RETURN TO USER
3578 021170 023420 SOTBL: 10000.
3579 021172 001750 1000.
3580 021174 000144 100.
3581 021176 000012 10.
3582 021200 000004 SDBLK: .BLKW 4
3583 ;*****
3584
3585 .SBT: TTY INPUT ROUTINE

```

```

3586
3587 ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
3588 ;*CALL:
3589 ;* RDCHR ;: INPUT A SINGLE CHARACTER FROM THE TTY
3590 ;* RETURN HERE ;: CHARACTER IS ON THE STACK
3591 ;
3592
3593 021210 011646 $RDCHR: MOV (SP), -(SP) ;: PUSH DOWN THE PC
3594 021212 016666 000004 000002 MOV 4(SP), 2(SP) ;: SAVE THE PS
3595 021220 105777 157712 1$: TSTB 2$TKS ;: WAIT FOR
3596 021224 100375 BPL 1$;: A CHARACTER
3597 021226 117766 157706 000004 MOV 2$TKB, 4(SP) ;: READ THE TTY
3598 021234 042766 177600 000004 BIC #1C(177), 4(SP) ;: GET RID OF JUNK IF ANY
3599 021242 000C02 RTI ;: GO BACK TO USER
3600 ;*****
3601 ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
3602 ;*CALL:
3603 ;* RDLIN ;: INPUT A STRING FROM THE TTY
3604 ;* RETURN HERE ;: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3605 ;* ;: TERMINATOR WILL BE A BYTE OF ALL D'S
3606 ;
3607 021244 010346 $RDLIN: MOV R3, -(SP) ;: SAVE R3
3608 021246 005046 CLR -(SP) ;: CLEAR THE RUBOUT KEY
3609 021250 012703 021505 1$: MOV #ST(YIN, R3) ;: GET ADDRESS
3610 021254 022703 021514 2$: CMP #STTYIN+7, R3 ;: BUFFER FULL?
3611 021260 101456 BLOS 4$;: BR IF YES
3612 021262 104412 RDCHR ;: GO READ ONE CHARACTER FROM THE TTY
3613 021264 112613 MOV (SP)+, (R3) ;: GET CHARACTER
3614 021266 122713 000177 CMPB #177, (R3) ;: IS IT A RUBOUT
3615 021272 001022 BNE 5$;: BR IF NO
3616 021274 005716 TST (SP) ;: IS THIS THE FIRST RUBOUT?
3617 021276 001007 BNE 6$;: BR IF NO
3618 021300 112737 000134 021476 MOV #'\, 9$;: TYPE A BACK SLASH
3619 021306 104400 021476 TYPE 9$
3620 021312 012716 177777 MOV 1-1, (SP) ;: SET THE RUBOUT KEY
3621 021316 005303 6$: DEC R3 ;: BACKUP BY ONE
3622 021320 020327 021505 CMP R3, #STTYIN ;: STACK EMPTY?
3623 021324 103434 BLO 4$;: BR IF YES
3624 021326 111337 021476 MOV (R3), 9$;: SETUP TO TYPEOUT THE DELETED CHAR.
3625 021332 104400 021476 TYPE 9$;: GO TYPE
3626 021336 000746 BR 2$;: GO READ ANOTHER CHAR.
3627 021340 005716 5$: TST (SP) ;: RUBOUT KEY SET?
3628 021342 001406 BEQ 7$;: BR IF NO
3629 021344 112737 000134 021476 MOV #'\, 9$;: TYPE A BACK SLASH
3630 021352 104400 021476 TYPE 9$
3631 021356 005016 CLR (SP) ;: CLEAR THE RUBOUT KEY
3632 021360 122713 000025 7$: CMPB #25, (R3) ;: IS CHARACTER A CTRL U?
3633 021364 001003 BNE 8$;: BR IF NO
3634 021366 104400 021500 TYPE , SCNTLU ;: TYPE A CONTROL "U"
3635 021372 000726 BR 1$;: GO START OVER
3636 021374 122713 000012 8$: CMPB #12, (R3) ;: IS CHARACTER A "LF"?
3637 021400 001011 BNE 3$;: BRANCH IF NO
3638 021402 105013 CLRB (R3) ;: CLEAR THE CHARACTER
3639 021404 104400 001201 TYPE , SCRLF ;: TYPE A "CR" & "LF"
3640 021410 104400 021505 TYPE , STTYIN ;: TYPE THE INPUT STRING
3641 021414 000717 BR 2$;: GO PICKUP ANOTHER CHARACTER

```

```

3642 021416 104400 001200 4$: TYPE $QUES ;;TYPE A '?'
3643 021422 000712 BR 1$;;CLEAR THE BUFFER AND LOOP
3644 021424 111337 021476 3$: MOVB (R3),9$;;ECHO THE CHARACTER
3645 021430 104400 021476 TYPE 9$
3646 021434 122723 000015 CMPB #15,(R3)+ ;;CHECK FOR RETURN
3647 021440 001305 BNE 2$;;LOOP IF NOT RETURN
3648 021442 105063 177777 CLRB -1(R3) ;;CLEAR RETURN (THE 15)
3649 021446 104400 001202 TYPE $LF ;;TYPE A LINE FEED
3650 021452 005726 TST (SP)+ ;;CLEAN RUBOUT KEY FROM THE STACK
3651 021454 012603 MOV (SP)+,R3 ;;RESTORE R3
3652 021456 011646 MOV (SP)-,(SP) ;;ADJUST THE STACK AND PUT ADDRESS OF THE
3653 021460 016666 000004 000002 MOV 4(SP),2(SP) ;; FIRST ASCII CHARACTER ON IT
3654 021466 012766 021505 000004 MOV #TTYIN,4(SP)
3655 021474 000002 RTI ;;RETURN
3656 021476 000 9$: .BYTE 0 ;;STORAGE FOR ASCII CHAR. TO TYPE
3657 021477 000 .BYTE 0 ;;TERMINATOR
3658 021500 052536 005015 000 $CNTLU: .ASCIZ /!U/<15><12> ;;CONTROL "U"
3659 021505 000007 $TTYIN: .BLKB 7 ;;RESERVE 7 BYTES FOR TTY INPUT
;*****
.SBTTL READ AN OCTAL NUMBER FROM THE TTY
;
;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 ;;READ AN OCTAL NUMBER
;* RETURN HERE ;;LOW ORDER BITS ARE ON TOP OF THE STACK
;* ;;HIGH ORDER BITS ARE IN $HIOCT
$RDOCT: MOV (SP)-,(SP) ;;PROVIDE SPACE FOR THE
MOV 4(SP),2(SP) ;;INPUT NUMBER
MOV R0,-(SP) ;;PUSH R0 ON STACK
MOV R1,-(SP) ;;PUSH R1 ON STACK
MOV R2,-(SP) ;;PUSH R2 ON STACK
1$: RDLIN ;;READ AN ASCIZ LINE
MOV (SP)+,R0 ;;GET ADDRESS OF 1ST CHARACTER
MOV R0,$$;;AND SAVE IT
CLR R1 ;;CLEAR DATA WORD
CLR R2
2$: MOVB (R0)+,-(SP) ;;PICKUP THIS CHARACTER
BEQ 3$;;IF ZERO GET OUT
CMPB #'0,(SP) ;;MAKE SURE THIS CHARACTER
BGT 4$;;IS AN OCTAL DIGIT
CMPB #'7,(SP)
BLT 4$
3$: ASL R1 ;;*2
ROL R2
4$: ASL R1 ;;*4
ROL R2
5$: ASL R1 ;;*8
ROL R2
6$: BIC #!C7,(SP) ;;STRIP THE ASCII JUNK

```

```

3698 021606 062601 AUD (SP)+,R1 ;; ADD IN THIS DIGIT
3699 021610 000756 BR 2$;; LOOP
3700 021612 005726 3$: TST (SP)+ ;; CLEAN TERMINATOR FROM STACK
3701 021614 010166 000012 MOV R1,12(SP) ;; SAVE THE RESULT
3702 021620 010237 021652 MOV R2,$SHIOCT
3703 021624 012602 MOV (SP)+,R2 ;; POP STACK INTO R2
3704 021626 012601 MOV (SP)+,R1 ;; POP STACK INTO R1
3705 021630 012600 MOV (SP)+,R0 ;; POP STACK INTO R0
3706 021632 000002 RTI ;; RETURN
3707 021634 005726 4$: TST (SP)+ ;; CLEAN PARTIAL FROM STACK
3708 021636 105010 CLR (R0) ;; SET A TERMINATOR
3709 021640 104400 TYPE ;; TYPE UP THRU THE BAD CHAR.
3710 021642 000000 5$: .WORD 0
3711 021644 104400 001200 TYPE $QUES ;; "?" "CR" & "LF"
3712 021650 000730 BR 1$;; TRY AGAIN
3713 021652 000000 $SHIOCT: .WORD 0 ;; HIGH ORDER BITS GO HERE
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733 021654 010046 $SBTTL SAVE AND RESTORE R0-R5 ROUTINES
3734 021656 010146 ;; *SAVE R0-R5
3735 021660 010246 ;; *CALL:
3736 021662 010346 ;; * SAVREG
3737 021664 010446 ;; *UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
3738 021666 010546 ;; *
3739 021670 016646 000022 ;; *TOP---(+16)
3740 021674 016646 000022 ;; * +2---(+18)
3741 021700 016646 000022 ;; * +4---R5
3742 021704 016646 000022 ;; * +6---R4
3743 021710 000002 ;; * +8---R3
3744
3745
3746
3747
3748 021712 ;; *+10---R2
3749 021712 012666 000022 ;; *+12---R1
3750 021716 012666 000022 ;; *+14---R0
3751 021722 012666 000022 $SAVREG:
3752 021726 012666 000022 MOV R0,-(SP) ;; PUSH R0 ON STACK
3753 021732 012605 MOV R1,-(SP) ;; PUSH R1 ON STACK
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000

```

3754 021734 012604  
3755 021736 012603  
3756 021740 012602  
3757 021742 012601  
3758 021744 012600  
3759 021746 000002

MOV (SP)+,R4 ;:POP STACK INTO R4  
MOV (SP)+,R3 ;:POP STACK INTO R3  
MOV (SP)+,R2 ;:POP STACK INTO R2  
MOV (SP)+,R1 ;:POP STACK INTO R1  
MOV (SP)+,R0 ;:POP STACK INTO R0  
RTI

\*\*\*\*\*

.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.

3769 021750 010046  
3770 021752 016600 000002  
3771 021756 005740  
3772 021760 111000  
3773 021762 016000 021770  
3774 021766 000200

\$TRAP: MOV R0, -(SP) ;:SAVE R0  
MOV 2(SP),R0 ;:GET TRAP ADDRESS  
TST -(R0) ;:BACKUP BY 2  
MOVB (R0),R0 ;:GET RIGHT BYTE OF TRAP  
MOV \$TRPAD(R0),R0 ;:INDEX TO TABLE  
RTS R0 ;:GO TO ROUTINE

.SBTTL TRAP TABLE

;\*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
;\*BY THE "TRAP" INSTRUCTION.

ROUTINE  
-----

3784 021770  
3785 021770 020430  
3786 021772 020562  
3787 021774 020536  
3788 021776 020576  
3789 022000 020764  
3790 022002 021210  
3791 022004 021244  
3792 022006 021514  
3793 022010 021654  
3794 022012 021712

\$TRPAD: \$TYPE ;:CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE  
\$TYPOC ;:CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)  
\$TYPOS ;:CALL=TYPOS TRAP+4(104404) TYPE OCTAL NUMBER (NO LEADING ZEROS)  
\$TYPON ;:CALL=TYPON TRAP+6(104406) TYPE OCTAL NUMBER (AS PER LAST CALL)  
\$TYPDS ;:CALL=TYPDS TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)  
\$RDCHR ;:CALL=RDCHR TRAP+12(104412) TTY TYPEIN CHARACTER ROUTINE  
\$RDLIN ;:CALL=RDLIN TRAP+14(104414) TTY TYPEIN STRING ROUTINE  
\$RDOCT ;:CALL=RDOCT TRAP+16(104416) READ AN OCTAL NUMBER FROM TTY  
\$SAVREG ;:CALL=SAVREG TRAP+20(104420) SAVE RO-R5 ROUTINE  
\$RESREG ;:CALL=RESREG TRAP+22(104422) RESTORE RO-R5 ROUTINE

\*\*\*\*\*

.SBTTL TELETYPE MESSAGES

\*\*\*\*\*

3801  
3802 022014 005015 040515 047111  
3803 022022 042504 026503 030461  
3804 022030 042055 051105 050520  
3805 022036 040455 005015 012  
3806 022043 122 030120 020064  
3807 022050 052504 046101 041440  
3808 022056 047117 051124 046117  
3809 022064 042514 020122 047514

TITLE: .ASCII <15><12>/MAINDEC-11-DERPQ-A/<15><12><12>

.ASCIZ /RPO4 DUAL CONTROLLER LOGIC TEST - PART 2/<15><12><12>

|      |        |        |        |        |                                                                                |
|------|--------|--------|--------|--------|--------------------------------------------------------------------------------|
| 3810 | 022072 | 044507 | 020103 | 042524 |                                                                                |
| 3811 | 022100 | 052123 | 026440 | 050040 |                                                                                |
| 3812 | 022106 | 051101 | 020124 | 006462 |                                                                                |
| 3813 | 022114 | 005012 | 000    |        |                                                                                |
| 3814 | 022117 | 015    | 042412 | 052116 | ENTERA: .ASCIZ <15><12>/ENTER DRIVE ADDRESS: /                                 |
| 3815 | 022124 | 051105 | 042040 | 044522 |                                                                                |
| 3816 | 022132 | 042526 | 040440 | 042104 |                                                                                |
| 3817 | 022140 | 042522 | 051523 | 020072 |                                                                                |
| 3818 | 022146 | 000    |        |        |                                                                                |
| 3819 | 022147 | 111    | 053116 | 046101 | ADRERR: .ASCIZ /INVALID ADDRESS/<15><12>                                       |
| 3820 | 022154 | 042111 | 040440 | 042104 |                                                                                |
| 3821 | 022162 | 042522 | 051523 | 005015 |                                                                                |
| 3822 | 022170 | 000    |        |        |                                                                                |
| 3823 | 022171 | 015    | 050012 | 051117 | PORTAIS: .ASCIZ <15><12>/PORT A ADDRESS IS: /                                  |
| 3824 | 022176 | 020124 | 020101 | 042101 |                                                                                |
| 3825 | 022204 | 051104 | 051505 | 020123 |                                                                                |
| 3826 | 022212 | 051511 | 020072 | 000    |                                                                                |
| 3827 | 022217 | 015    | 050012 | 051117 | PORTBIS: .ASCIZ <15><12>/PORT B ADDRESS IS: /                                  |
| 3828 | 022224 | 020124 | 020102 | 042101 |                                                                                |
| 3829 | 022232 | 051104 | 051505 | 020123 |                                                                                |
| 3830 | 022240 | 051511 | 020072 | 000    |                                                                                |
| 3831 | 022245 | 015    | 051412 | 051531 | NOCLOCK: .ASCIZ <15><12>/SYSTEM MUST HAVE 'L' OR 'P' CLOCK/<15><12><12>        |
| 3832 | 022252 | 042524 | 020115 | 052515 |                                                                                |
| 3833 | 022260 | 052123 | 044040 | 053101 |                                                                                |
| 3834 | 022266 | 020105 | 046047 | 020047 |                                                                                |
| 3835 | 022274 | 051117 | 023440 | 023520 |                                                                                |
| 3836 | 022302 | 041440 | 047514 | 045503 |                                                                                |
| 3837 | 022310 | 005015 | 000012 |        |                                                                                |
| 3838 | 022314 | 042412 | 052116 | 051105 | TESTNO: .ASCIZ <12>/ENTER TEST #: /                                            |
| 3839 | 022322 | 052040 | 051505 | 020124 |                                                                                |
| 3840 | 022330 | 035043 | 000040 |        |                                                                                |
| 3841 | 022334 | 047111 | 040526 | 044514 | BADNO: .ASCIZ /INVALID TEST NUMBER/<15><12>                                    |
| 3842 | 022342 | 020104 | 042524 | 052123 |                                                                                |
| 3843 | 022350 | 047040 | 046525 | 042502 |                                                                                |
| 3844 | 022356 | 006522 | 000012 |        |                                                                                |
| 3845 | 022362 | 005015 | 052012 | 042510 | ADDRIS: .ASCIZ <15><12><12>/THE PRESENT ADDRESS OF THE RH11 (RHCS1) IS: /      |
| 3846 | 022370 | 050040 | 042522 | 042523 |                                                                                |
| 3847 | 022376 | 052116 | 040440 | 042104 |                                                                                |
| 3848 | 022404 | 042522 | 051523 | 047440 |                                                                                |
| 3849 | 022412 | 020106 | 044124 | 020105 |                                                                                |
| 3850 | 022420 | 044122 | 030461 | 024040 |                                                                                |
| 3851 | 022426 | 044122 | 051503 | 024461 |                                                                                |
| 3852 | 022434 | 044440 | 035123 | 000040 |                                                                                |
| 3853 | 022442 | 042412 | 052116 | 051105 | NTRH11: .ASCIZ <12>/ENTER NEW RH11 ADDRESS: /                                  |
| 3854 | 022450 | 047040 | 053505 | 051040 |                                                                                |
| 3855 | 022456 | 030510 | 020061 | 042101 |                                                                                |
| 3856 | 022464 | 051104 | 051505 | 035123 |                                                                                |
| 3857 | 022472 | 000040 |        |        |                                                                                |
| 3858 | 022474 | 005015 | 044124 | 020105 | NORESP: .ASCIZ <15><12>/THE RH11 DID NOT RESPOND WHEN RHWC ACCESSED AT ADDR: / |
| 3859 | 022502 | 044122 | 030461 | 042040 |                                                                                |
| 3860 | 022510 | 042111 | 047040 | 052117 |                                                                                |
| 3861 | 022516 | 051040 | 051505 | 047520 |                                                                                |
| 3862 | 022524 | 042116 | 053440 | 042510 |                                                                                |
| 3863 | 022532 | 020116 | 044122 | 041527 |                                                                                |
| 3864 | 022540 | 040440 | 041503 | 051505 |                                                                                |
| 3865 | 022546 | 042523 | 020104 | 052101 |                                                                                |



|      |        |        |        |        |                                                                                 |
|------|--------|--------|--------|--------|---------------------------------------------------------------------------------|
| 3866 | 022554 | 040440 | 042104 | 035122 |                                                                                 |
| 3867 | 022562 | 000040 |        |        |                                                                                 |
| 3868 | 022564 | 005015 | 050012 | 042522 | STANDBY: .ASCII <15><12><12>/PRESS 'STANDBY' ON DRIVE/                          |
| 3869 | 022572 | 051523 | 023440 | 052123 |                                                                                 |
| 3870 | 022600 | 047101 | 041104 | 023531 |                                                                                 |
| 3871 | 022606 | 047440 | 020116 | 051104 |                                                                                 |
| 3872 | 022614 | 053111 | 105    |        |                                                                                 |
| 3873 | 022617 | 015    | 050012 | 047522 | .ASCIZ <15><12>/PROGRAM WILL LOOP WAITING FOR 'MOL' TO SET FROM PORT /          |
| 3874 | 022624 | 051107 | 046501 | 053440 |                                                                                 |
| 3875 | 022632 | 046111 | 020114 | 047514 |                                                                                 |
| 3876 | 022640 | 050117 | 053440 | 044501 |                                                                                 |
| 3877 | 022646 | 044524 | 043516 | 043040 |                                                                                 |
| 3878 | 022654 | 051117 | 023440 | 047515 |                                                                                 |
| 3879 | 022662 | 023514 | 052040 | 020117 |                                                                                 |
| 3880 | 022670 | 042523 | 020124 | 051106 |                                                                                 |
| 3881 | 022676 | 046517 | 050040 | 051117 |                                                                                 |
| 3882 | 022704 | 020124 | 000    |        |                                                                                 |
| 3883 | 022707 | 015    | 005012 | 042522 | SWTCHN: .ASCIZ <15><12><12>RETURN 'CONTROLLER SELECT' SWITCH ON DRIVE TO 'A/B'@ |
| 3884 | 022714 | 052524 | 047122 | 023440 |                                                                                 |
| 3885 | 022722 | 047503 | 052116 | 047522 |                                                                                 |
| 3886 | 022730 | 046114 | 051105 | 051440 |                                                                                 |
| 3887 | 022736 | 046105 | 041505 | 023524 |                                                                                 |
| 3888 | 022744 | 051440 | 044527 | 041524 |                                                                                 |
| 3889 | 022752 | 020110 | 047117 | 042040 |                                                                                 |
| 3890 | 022760 | 044522 | 042526 | 052040 |                                                                                 |
| 3891 | 022766 | 020117 | 040447 | 041057 |                                                                                 |
| 3892 | 022774 | 000047 |        |        |                                                                                 |
| 3893 | 022776 | 005015 | 052012 | 051125 | SWTCHA: .ASCIZ <15><12><12>/TURN 'CONTROLLER SELECT' SWITCH ON DRIVE TO 'A'/'   |
| 3894 | 023004 | 020116 | 041447 | 047117 |                                                                                 |
| 3895 | 023012 | 051124 | 046117 | 042514 |                                                                                 |
| 3896 | 023020 | 020122 | 042523 | 042514 |                                                                                 |
| 3897 | 023026 | 052103 | 020047 | 053523 |                                                                                 |
| 3898 | 023034 | 052111 | 044103 | 047440 |                                                                                 |
| 3899 | 023042 | 020116 | 051104 | 053111 |                                                                                 |
| 3900 | 023050 | 020105 | 047524 | 023440 |                                                                                 |
| 3901 | 023056 | 023501 | 000    |        |                                                                                 |
| 3902 | 023061 | 015    | 005012 | 052524 | SWTCHB: .ASCIZ <15><12><12>/TURN 'CONTROLLER SELECT' SWITCH ON DRIVE TO 'B'/'   |
| 3903 | 023066 | 047122 | 023440 | 047503 |                                                                                 |
| 3904 | 023074 | 052116 | 047522 | 046114 |                                                                                 |
| 3905 | 023102 | 051105 | 051440 | 046105 |                                                                                 |
| 3906 | 023110 | 041505 | 023524 | 051440 |                                                                                 |
| 3907 | 023116 | 044527 | 041524 | 020110 |                                                                                 |
| 3908 | 023124 | 047117 | 042040 | 044522 |                                                                                 |
| 3909 | 023132 | 042526 | 052040 | 020117 |                                                                                 |
| 3910 | 023140 | 041047 | 000047 |        |                                                                                 |
| 3911 | 023144 | 005015 | 044124 | 047105 | CONTUE: .ASCIZ <15><12>/THEN PRESS 'CONT' ON THE PROCESSOR/<15><12>             |
| 3912 | 023152 | 050040 | 042522 | 051523 |                                                                                 |
| 3913 | 023160 | 023440 | 047503 | 052116 |                                                                                 |
| 3914 | 023166 | 020047 | 047117 | 052040 |                                                                                 |
| 3915 | 023174 | 042510 | 050040 | 047522 |                                                                                 |
| 3916 | 023202 | 042503 | 051523 | 051117 |                                                                                 |
| 3917 | 023210 | 005015 | 000    |        |                                                                                 |
| 3918 | 023213 | 015    | 005012 | 052123 | CYCLED: .ASCIZ <15><12><12>/STOP THE DRIVE/                                     |
| 3919 | 023220 | 050117 | 052040 | 042510 |                                                                                 |
| 3920 | 023226 | 042040 | 044522 | 042526 |                                                                                 |
| 3921 | 023234 | 000    |        |        |                                                                                 |

3922 023235 015 005012 052123  
 3923 023242 051101 020124 044124  
 3924 023250 020105 051104 053111  
 3925 023256 020105 020055 044124  
 3926 023264 020105 051120 043517  
 3927 023272 040522 020115 044527  
 3928 023300 046114 053440 044501  
 3929 023306 020124 047506 020122  
 3930 023314 046447 046117 020047  
 3931 023322 047524 051440 052105  
 3932 023330 000  
 3933  
 3934  
 3935  
 3936  
 3937  
 3938  
 3939  
 3940 023331 104 044522 042526  
 3941 023336 044440 020123 047516  
 3942 023344 026516 054105 051511  
 3943 023352 040524 052116 024040  
 3944 023360 047047 042105 020047  
 3945 023366 044502 020124 042523  
 3946 023374 024524 000  
 3947  
 3948 023377 127 047522 043516  
 3949 023404 042040 044522 042526  
 3950 023412 052040 050131 000105  
 3951  
 3952 023420 047503 052116 047522  
 3953 023426 046114 051105 051440  
 3954 023434 046105 041505 020124  
 3955 023442 053523 052111 044103  
 3956 023450 047440 020116 051104  
 3957 023456 053111 020105 047516  
 3958 023464 020124 047111 023440  
 3959 023472 027501 023502 000  
 3960  
 3961 023477 104 044522 042526  
 3962 023504 047040 052117 047440  
 3963 023512 020116 044514 042516  
 3964 023520 000  
 3965  
 3966 023521 123 051105 040511  
 3967 023526 020114 052516 041115  
 3968 023534 051105 051040 040505  
 3969 023542 020104 044124 047522  
 3970 023550 043525 020110 040505  
 3971 023556 044103 050040 051117  
 3972 023564 020124 047516 020124  
 3973 023572 044124 020105 040523  
 3974 023600 042515 000  
 3975  
 3976 023603 124 046511 047505  
 3977 023610 052125 044040 051501

CYCLEU: .ASCIZ <15><12><12>/START THE DRIVE - THE PROGRAM WILL WAIT FOR 'MOL' TO SET/

;;\*\*\*\*\*

.SBTTL TEST ERROR MESSAGES

;;\*\*\*\*\*

EM1: .ASCIZ /DRIVE IS NON-EXISTANT ('NED' BIT SET)/

EM2: .ASCIZ /WRONG DRIVE TYPE/

EM3: .ASCIZ @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@

EM4: .ASCIZ /DRIVE NOT ON LINE/

EM5: .ASCIZ /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/

EM6: .ASCIZ /TIMEOUT HAS NOT OCCURED WITHIN 2 SECONDS/

|      |        |        |        |        |                                                                     |
|------|--------|--------|--------|--------|---------------------------------------------------------------------|
| 3978 | 023616 | 047040 | 052117 | 047440 |                                                                     |
| 3979 | 023624 | 041503 | 051125 | 042105 |                                                                     |
| 3980 | 023632 | 053440 | 052111 | 044510 |                                                                     |
| 3981 | 023640 | 020116 | 020062 | 042523 |                                                                     |
| 3982 | 023646 | 047503 | 042116 | 000123 |                                                                     |
| 3983 |        |        |        |        |                                                                     |
| 3984 | 023654 | 044524 | 042515 | 052517 | EM7: .ASCIZ /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/                  |
| 3985 | 023662 | 020124 | 047117 | 026505 |                                                                     |
| 3986 | 023670 | 044123 | 052117 | 044440 |                                                                     |
| 3987 | 023676 | 020123 | 042514 | 051523 |                                                                     |
| 3988 | 023704 | 052040 | 040510 | 020116 |                                                                     |
| 3989 | 023712 | 030065 | 020060 | 051515 |                                                                     |
| 3990 | 023720 | 000    |        |        |                                                                     |
| 3991 |        |        |        |        |                                                                     |
| 3992 | 023721 | 122    | 040505 | 044504 | EM10: .ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/ |
| 3993 | 023726 | 020116 | 051120 | 051505 |                                                                     |
| 3994 | 023734 | 052105 | 042040 | 042517 |                                                                     |
| 3995 | 023742 | 020123 | 047516 | 020124 |                                                                     |
| 3996 | 023750 | 042523 | 020124 | 047526 |                                                                     |
| 3997 | 023756 | 052514 | 042515 | 053040 |                                                                     |
| 3998 | 023764 | 046101 | 042111 | 043040 |                                                                     |
| 3999 | 023772 | 051117 | 052040 | 042510 |                                                                     |
| 4000 | 024000 | 050040 | 051117 | 000124 |                                                                     |
| 4001 |        |        |        |        |                                                                     |
| 4002 | 024006 | 043447 | 023517 | 041040 | EM11: .ASCIZ /'GO' BIT RESET DURING UNLOAD COMMAND/                 |
| 4003 | 024014 | 052111 | 051040 | 051505 |                                                                     |
| 4004 | 024022 | 052105 | 042040 | 051125 |                                                                     |
| 4005 | 024030 | 047111 | 020107 | 047125 |                                                                     |
| 4006 | 024036 | 047514 | 042101 | 041440 |                                                                     |
| 4007 | 024044 | 046517 | 040515 | 042116 |                                                                     |
| 4008 | 024052 | 000    |        |        |                                                                     |
| 4009 |        |        |        |        |                                                                     |
| 4010 | 024053 | 111    | 041516 | 051117 | EM12: .ASCIZ /INCORRECT STATUS DURING UNLOAD COMMAND/               |
| 4011 | 024060 | 042522 | 052103 | 051440 |                                                                     |
| 4012 | 024066 | 040524 | 052524 | 020123 |                                                                     |
| 4013 | 024074 | 052504 | 044522 | 043516 |                                                                     |
| 4014 | 024102 | 052440 | 046116 | 040517 |                                                                     |
| 4015 | 024110 | 020104 | 047503 | 046515 |                                                                     |
| 4016 | 024116 | 047101 | 000104 |        |                                                                     |
| 4017 |        |        |        |        |                                                                     |
| 4018 | 024122 | 051104 | 053111 | 020105 | EM13: .ASCIZ /DRIVE DID NOT RETURN TO NEUTRAL AFTER UNLOAD COMMAND/ |
| 4019 | 024130 | 044504 | 020104 | 047516 |                                                                     |
| 4020 | 024136 | 020124 | 042522 | 052524 |                                                                     |
| 4021 | 024144 | 047122 | 052040 | 020117 |                                                                     |
| 4022 | 024152 | 042516 | 052125 | 040522 |                                                                     |
| 4023 | 024160 | 020114 | 043101 | 042524 |                                                                     |
| 4024 | 024166 | 020122 | 047125 | 047514 |                                                                     |
| 4025 | 024174 | 042101 | 041440 | 046517 |                                                                     |
| 4026 | 024202 | 040515 | 042116 | 000    |                                                                     |
| 4027 |        |        |        |        |                                                                     |
| 4028 | 024207 | 101    | 052124 | 047105 | EM14: .ASCIZ /ATTENTION BIT SET ON 'OPPOSITE PORT' AFTER UNLOAD/    |
| 4029 | 024214 | 044524 | 047117 | 041040 |                                                                     |
| 4030 | 024222 | 052111 | 051440 | 052105 |                                                                     |
| 4031 | 024230 | 047440 | 020116 | 047447 |                                                                     |
| 4032 | 024236 | 050120 | 051517 | 052111 |                                                                     |
| 4033 | 024244 | 020105 | 047520 | 052122 |                                                                     |

|      |        |        |        |        |                                                                           |
|------|--------|--------|--------|--------|---------------------------------------------------------------------------|
| 4034 | 024252 | 020047 | 043101 | 042524 |                                                                           |
| 4035 | 024260 | 020122 | 047125 | 047514 |                                                                           |
| 4036 | 024266 | 042101 | 000    |        |                                                                           |
| 4037 |        |        |        |        |                                                                           |
| 4038 | 024271 | 101    | 052124 | 047105 | EM15: .ASCIZ /ATTENTION BIT NOT SET ON PORT WHICH ISSUED 'UNLOAD'/'       |
| 4039 | 024276 | 044524 | 047117 | 041040 |                                                                           |
| 4040 | 024304 | 052111 | 047040 | 052117 |                                                                           |
| 4041 | 024312 | 051440 | 052105 | 047440 |                                                                           |
| 4042 | 024320 | 020116 | 047520 | 052122 |                                                                           |
| 4043 | 024326 | 053440 | 044510 | 044103 |                                                                           |
| 4044 | 024334 | 044440 | 051523 | 042525 |                                                                           |
| 4045 | 024342 | 020104 | 052447 | 046116 |                                                                           |
| 4046 | 024350 | 040517 | 023504 | 000    |                                                                           |
| 4047 |        |        |        |        |                                                                           |
| 4048 | 024355 | 104    | 044522 | 042526 | EM16: .ASCII /DRIVE NOT IN NEUTRAL AFTER UNLOAD WITH 'CONTROLLER/<15><12> |
| 4049 | 024362 | 047040 | 052117 | 044440 |                                                                           |
| 4050 | 024370 | 020116 | 042516 | 052125 |                                                                           |
| 4051 | 024376 | 040522 | 020114 | 043101 |                                                                           |
| 4052 | 024404 | 042524 | 020122 | 047125 |                                                                           |
| 4053 | 024412 | 047514 | 042101 | 053440 |                                                                           |
| 4054 | 024420 | 052111 | 020110 | 041447 |                                                                           |
| 4055 | 024426 | 047117 | 051124 | 046117 |                                                                           |
| 4056 | 024434 | 042514 | 006522 | 012    |                                                                           |
| 4057 | 024441 | 123    | 046105 | 041505 | .ASCIZ @SELECT' SWITCH MOVED FROM 'A/B'@                                  |
| 4058 | 024446 | 023524 | 051440 | 044527 |                                                                           |
| 4059 | 024454 | 041524 | 020110 | 047515 |                                                                           |
| 4060 | 024462 | 042526 | 020104 | 051106 |                                                                           |
| 4061 | 024470 | 046517 | 023440 | 027501 |                                                                           |
| 4062 | 024476 | 023502 | 000    |        |                                                                           |
| 4063 |        |        |        |        |                                                                           |
| 4064 | 024501 | 104    | 044522 | 042526 | EM17: .ASCIZ /DRIVE LOCKED ON PORT 'A' BY SWITCH WHILE CYCLED UP/         |
| 4065 | 024506 | 046040 | 041517 | 042513 |                                                                           |
| 4066 | 024514 | 020104 | 047117 | 050040 |                                                                           |
| 4067 | 024522 | 051117 | 020124 | 040447 |                                                                           |
| 4068 | 024530 | 020047 | 054502 | 051440 |                                                                           |
| 4069 | 024536 | 044527 | 041524 | 020110 |                                                                           |
| 4070 | 024544 | 044127 | 046111 | 020105 |                                                                           |
| 4071 | 024552 | 054503 | 046103 | 042105 |                                                                           |
| 4072 | 024560 | 052440 | 000120 |        |                                                                           |
| 4073 |        |        |        |        |                                                                           |
| 4074 | 024564 | 051104 | 053111 | 020105 | EM20: .ASCIZ /DRIVE LOCKED ON PORT 'B' BY SWITCH WHILE CYCLED UP/         |
| 4075 | 024572 | 047514 | 045503 | 042105 |                                                                           |
| 4076 | 024600 | 047440 | 020116 | 047520 |                                                                           |
| 4077 | 024606 | 052122 | 023440 | 023502 |                                                                           |
| 4078 | 024614 | 041040 | 020131 | 053523 |                                                                           |
| 4079 | 024622 | 052111 | 044103 | 053440 |                                                                           |
| 4080 | 024630 | 044510 | 042514 | 041440 |                                                                           |
| 4081 | 024636 | 041531 | 042514 | 020104 |                                                                           |
| 4082 | 024644 | 050125 | 000    |        |                                                                           |
| 4083 |        |        |        |        |                                                                           |
| 4084 | 024647 | 123    | 040524 | 052524 | EM21: .ASCIZ /STATUS INCORRECT FOR PORT AFTER CYCLE UP/                   |
| 4085 | 024654 | 020123 | 047111 | 047503 |                                                                           |
| 4086 | 024662 | 051122 | 041505 | 020124 |                                                                           |
| 4087 | 024670 | 047506 | 020122 | 047520 |                                                                           |
| 4088 | 024676 | 052122 | 040440 | 052106 |                                                                           |
| 4089 | 024704 | 051105 | 041440 | 041531 |                                                                           |

|      |        |        |        |        |                                                                              |
|------|--------|--------|--------|--------|------------------------------------------------------------------------------|
| 4090 | 024712 | 042514 | 052440 | 000120 |                                                                              |
| 4091 |        |        |        |        |                                                                              |
| 4092 | 024720 | 042522 | 044507 | 052123 | EM22: .ASCIZ /REGISTER CONTENTS SEEN WHEN DRIVE SWITCHED ON 'OPPOSITE' PORT/ |
| 4093 | 024726 | 051105 | 041440 | 047117 |                                                                              |
| 4094 | 024734 | 042524 | 052116 | 020123 |                                                                              |
| 4095 | 024742 | 042523 | 047105 | 053440 |                                                                              |
| 4096 | 024750 | 042510 | 020116 | 051104 |                                                                              |
| 4097 | 024756 | 053111 | 020105 | 053523 |                                                                              |
| 4098 | 024764 | 052111 | 044103 | 042105 |                                                                              |
| 4099 | 024772 | 047440 | 020116 | 047447 |                                                                              |
| 4100 | 025000 | 050120 | 051517 | 052111 |                                                                              |
| 4101 | 025006 | 023505 | 050040 | 051117 |                                                                              |
| 4102 | 025014 | 000124 |        |        |                                                                              |
| 4103 |        |        |        |        |                                                                              |
| 4104 | 025016 | 047047 | 042105 | 020047 | EM23: .ASCIZ /'NED' SET WHEN RHDS1 ACCESSED THROUGH PORT NOT SWITCHED/       |
| 4105 | 025024 | 042523 | 020124 | 044127 |                                                                              |
| 4106 | 025032 | 047105 | 051040 | 042110 |                                                                              |
| 4107 | 025040 | 030523 | 040440 | 041503 |                                                                              |
| 4108 | 025046 | 051505 | 042523 | 020104 |                                                                              |
| 4109 | 025054 | 044124 | 047522 | 043525 |                                                                              |
| 4110 | 025062 | 020110 | 047520 | 052122 |                                                                              |
| 4111 | 025070 | 047040 | 052117 | 051440 |                                                                              |
| 4112 | 025076 | 044527 | 041524 | 042510 |                                                                              |
| 4113 | 025104 | 000104 |        |        |                                                                              |
| 4114 |        |        |        |        |                                                                              |
| 4115 | 025106 | 051104 | 053111 | 020105 | EM24: .ASCIZ /DRIVE SWITCHED TO LOCKED OUT PORT WHEN RELEASED/               |
| 4116 | 025114 | 053523 | 052111 | 044103 |                                                                              |
| 4117 | 025122 | 042105 | 052040 | 020117 |                                                                              |
| 4118 | 025130 | 047514 | 045503 | 042105 |                                                                              |
| 4119 | 025136 | 047440 | 052125 | 050040 |                                                                              |
| 4120 | 025144 | 051117 | 020124 | 044127 |                                                                              |
| 4121 | 025152 | 047105 | 051040 | 046105 |                                                                              |
| 4122 | 025160 | 040505 | 042523 | 000104 |                                                                              |
| 4123 |        |        |        |        |                                                                              |
| 4124 | 025166 | 051104 | 053111 | 020105 | EM30: .ASCIZ /DRIVE NOT SEIZED BY PORT/                                      |
| 4125 | 025174 | 047516 | 020124 | 042523 |                                                                              |
| 4126 | 025202 | 055111 | 042105 | 041040 |                                                                              |
| 4127 | 025210 | 020131 | 047520 | 052122 |                                                                              |
| 4128 | 025216 | 000    |        |        |                                                                              |
| 4129 |        |        |        |        |                                                                              |
| 4130 | 025217 | 127    | 047522 | 043516 | EM31: .ASCIZ /WRONG STATUS SEEN BY THE SEIZING PORT/                         |
| 4131 | 025224 | 051440 | 040524 | 052524 |                                                                              |
| 4132 | 025232 | 020123 | 042523 | 047105 |                                                                              |
| 4133 | 025240 | 041040 | 020131 | 044124 |                                                                              |
| 4134 | 025246 | 020105 | 042523 | 055111 |                                                                              |
| 4135 | 025254 | 047111 | 020107 | 047520 |                                                                              |
| 4136 | 025262 | 052122 | 000    |        |                                                                              |
| 4137 |        |        |        |        |                                                                              |
| 4138 | 025265 | 122    | 043505 | 051511 | EM32: .ASCIZ /REGISTER CONTENTS WRONG/                                       |
| 4139 | 025272 | 042524 | 020122 | 047503 |                                                                              |
| 4140 | 025300 | 052116 | 047105 | 051524 |                                                                              |
| 4141 | 025306 | 053440 | 047522 | 043516 |                                                                              |
| 4142 | 025314 | 000    |        |        |                                                                              |
| 4143 |        |        |        |        |                                                                              |
| 4144 | 025315 | 103    | 047117 | 051124 | EM33: .ASCIZ /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/           |
| 4145 | 025322 | 046117 | 041040 | 051525 |                                                                              |

|      |        |        |        |        |                                                                     |
|------|--------|--------|--------|--------|---------------------------------------------------------------------|
| 4146 | 025330 | 050040 | 051101 | 052111 |                                                                     |
| 4147 | 025336 | 020131 | 051105 | 047522 |                                                                     |
| 4148 | 025344 | 020122 | 042522 | 042101 |                                                                     |
| 4149 | 025352 | 047111 | 020107 | 047111 |                                                                     |
| 4150 | 025360 | 044504 | 040503 | 042524 |                                                                     |
| 4151 | 025366 | 020104 | 042522 | 044507 |                                                                     |
| 4152 | 025374 | 052123 | 051105 | 000    |                                                                     |
| 4153 |        |        |        |        |                                                                     |
| 4154 | 025401 | 103    | 047101 | 052047 | EM34: .ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT/               |
| 4155 | 025406 | 040440 | 041503 | 051505 |                                                                     |
| 4156 | 025414 | 020123 | 051104 | 053111 |                                                                     |
| 4157 | 025422 | 020105 | 044124 | 047522 |                                                                     |
| 4158 | 025430 | 043525 | 020110 | 044505 |                                                                     |
| 4159 | 025436 | 044124 | 051105 | 050040 |                                                                     |
| 4160 | 025444 | 051117 | 000124 |        |                                                                     |
| 4161 |        |        |        |        |                                                                     |
| 4162 | 025450 | 051104 | 053111 | 020105 | EM35: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/ |
| 4163 | 025456 | 047516 | 020124 | 047111 |                                                                     |
| 4164 | 025464 | 047040 | 052505 | 051124 |                                                                     |
| 4165 | 025472 | 046101 | 040440 | 052106 |                                                                     |
| 4166 | 025500 | 051105 | 051040 | 046105 |                                                                     |
| 4167 | 025506 | 040505 | 042523 | 026440 |                                                                     |
| 4168 | 025514 | 051040 | 050505 | 042525 |                                                                     |
| 4169 | 025522 | 052123 | 047040 | 052117 |                                                                     |
| 4170 | 025530 | 051440 | 052105 | 000    |                                                                     |
| 4171 |        |        |        |        |                                                                     |
| 4172 | 025535 | 104    | 044522 | 042526 | EM36: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/ |
| 4173 | 025542 | 047040 | 052117 | 044440 |                                                                     |
| 4174 | 025550 | 020116 | 042516 | 052125 |                                                                     |
| 4175 | 025555 | 040522 | 020114 | 043101 |                                                                     |
| 4176 | 025564 | 042524 | 020122 | 044524 |                                                                     |
| 4177 | 025572 | 042515 | 052517 | 020124 |                                                                     |
| 4178 | 025600 | 020055 | 042522 | 052521 |                                                                     |
| 4179 | 025606 | 051505 | 020124 | 047516 |                                                                     |
| 4180 | 025614 | 020124 | 042523 | 000124 |                                                                     |
| 4181 |        |        |        |        |                                                                     |
| 4182 | 025622 | 042522 | 044507 | 052123 | EM37: .ASCIZ /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/     |
| 4183 | 025630 | 051105 | 041440 | 047117 |                                                                     |
| 4184 | 025636 | 042524 | 052116 | 020123 |                                                                     |
| 4185 | 025644 | 051127 | 047117 | 020107 |                                                                     |
| 4186 | 025652 | 043101 | 042524 | 020122 |                                                                     |
| 4187 | 025660 | 042522 | 042514 | 051501 |                                                                     |
| 4188 | 025666 | 020105 | 051117 | 052040 |                                                                     |
| 4189 | 025674 | 046511 | 047505 | 052125 |                                                                     |
| 4190 | 025702 | 000    |        |        |                                                                     |
| 4191 |        |        |        |        |                                                                     |
| 4192 | 025703 | 104    | 044522 | 042526 | EM40: .ASCIZ /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/         |
| 4193 | 025710 | 044440 | 020116 | 042516 |                                                                     |
| 4194 | 025716 | 052125 | 040522 | 020114 |                                                                     |
| 4195 | 025724 | 043101 | 042524 | 020122 |                                                                     |
| 4196 | 025732 | 042522 | 042514 | 051501 |                                                                     |
| 4197 | 025740 | 020105 | 020055 | 042522 |                                                                     |
| 4198 | 025746 | 052521 | 051505 | 020124 |                                                                     |
| 4199 | 025754 | 042523 | 000124 |        |                                                                     |
| 4200 |        |        |        |        |                                                                     |
| 4201 | 025760 | 042522 | 044507 | 052123 | EM41: .ASCIZ /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/        |

|      |        |        |        |        |
|------|--------|--------|--------|--------|
| 4202 | 025766 | 051105 | 053440 | 047522 |
| 4203 | 025774 | 043516 | 040440 | 052106 |
| 4204 | 026002 | 051105 | 051040 | 046105 |
| 4205 | 026010 | 040505 | 042523 | 053440 |
| 4206 | 026016 | 052111 | 020110 | 042522 |
| 4207 | 026024 | 052521 | 051505 | 020124 |
| 4208 | 026032 | 042523 | 000124 |        |
| 4209 |        |        |        |        |
| 4210 |        |        |        |        |
| 4211 | 026036 | 042524 | 052123 | 021440 |
| 4212 | 026044 | 020040 | 051105 | 020122 |
| 4213 | 026052 | 041520 | 020040 | 047520 |
| 4214 | 026060 | 052122 | 021440 | 020040 |
| 4215 | 026066 | 042522 | 020107 | 042101 |
| 4216 | 026074 | 020122 | 047503 | 052116 |
| 4217 | 026102 | 047105 | 051524 | 000    |
| 4218 | 026107 | 124    | 051505 | 020124 |
| 4219 | 026114 | 020043 | 042440 | 051122 |
| 4220 | 026122 | 050040 | 020103 | 050040 |
| 4221 | 026130 | 051117 | 020124 | 020043 |
| 4222 | 026136 | 051040 | 043505 | 040440 |
| 4223 | 026144 | 051104 | 043440 | 047517 |
| 4224 | 026152 | 020104 | 020040 | 041040 |
| 4225 | 026160 | 042101 | 000    |        |
| 4226 | 026163 | 124    | 051505 | 020124 |
| 4227 | 026170 | 020043 | 042440 | 051122 |
| 4228 | 026176 | 050040 | 020103 | 051040 |
| 4229 | 026204 | 043505 | 040440 | 051104 |
| 4230 | 026212 | 050040 | 051117 | 020124 |
| 4231 | 026220 | 020101 | 050040 | 051117 |
| 4232 | 026226 | 020124 | 000102 |        |
| 4233 | 026232 | 042524 | 052123 | 021440 |
| 4234 | 026240 | 020040 | 051105 | 020122 |
| 4235 | 026246 | 041520 | 020040 | 047520 |
| 4236 | 026254 | 052122 | 021440 | 000    |
| 4237 | 026261 | 124    | 051505 | 020124 |
| 4238 | 026266 | 020043 | 042440 | 051122 |
| 4239 | 026274 | 050040 | 020103 | 050040 |
| 4240 | 026302 | 051117 | 020124 | 020043 |
| 4241 | 026310 | 052040 | 046511 | 020105 |
| 4242 | 026316 | 044450 | 020116 | 051515 |
| 4243 | 026324 | 000051 |        |        |
| 4244 | 026326 | 020040 | 020040 | 020040 |
| 4245 | 026334 | 020040 | 020040 | 020040 |
| 4246 | 026342 | 020040 | 020040 | 042523 |
| 4247 | 026350 | 055111 | 006505 | 012    |
| 4248 | 026355 | 124    | 051505 | 020124 |
| 4249 | 026362 | 020043 | 042440 | 051122 |
| 4250 | 026370 | 050040 | 020103 | 050040 |
| 4251 | 026376 | 051117 | 020124 | 000043 |
| 4252 | 026404 | 020040 | 020040 | 020040 |
| 4253 | 026412 | 020040 | 020040 | 020040 |
| 4254 | 026420 | 020040 | 020040 | 042523 |
| 4255 | 026426 | 055111 | 020105 | 020040 |
| 4256 | 026434 | 051105 | 047522 | 006522 |
| 4257 | 026442 | 012    |        |        |

DH1: .ASCIZ /TEST # ERR PC PORT # REG ADR CONTENTS/

DH2: .ASCIZ /TEST # ERR PC PORT # REG ADR GOOD BAD/

DH5: .ASCIZ /TEST # ERR PC REG ADR PORT A PORT B/

DH6: .ASCIZ /TEST # ERR PC PORT #/

DH7: .ASCIZ /TEST # ERR PC PORT # TIME (IN MS)/

DH13: .ASCII / SEIZE/<15><12>

.ASCIZ /TEST # ERR PC PORT #/

DH14: .ASCII / SEIZE ERROR/<15><12>



| ADDR | DATA1  | DATA2  | DATA3  | DATA4  | DATA5 | DATA6  | DATA7 | DATA8   | DATA9  | DATA10       |                   |
|------|--------|--------|--------|--------|-------|--------|-------|---------|--------|--------------|-------------------|
| 4258 | 026443 | 124    | 051505 | 020124 |       | .ASCIZ | /TEST | ERR PC  | PORT   | PORT         | REG ADR CONTENTS/ |
| 4259 | 026450 | 020043 | 042440 | 051122 |       |        |       |         |        |              |                   |
| 4260 | 026456 | 050040 | 020103 | 050040 |       |        |       |         |        |              |                   |
| 4261 | 026464 | 051117 | 020124 | 020043 |       |        |       |         |        |              |                   |
| 4262 | 026472 | 050040 | 051117 | 020124 |       |        |       |         |        |              |                   |
| 4263 | 026500 | 020043 | 051040 | 043505 |       |        |       |         |        |              |                   |
| 4264 | 026506 | 040440 | 051104 | 041440 |       |        |       |         |        |              |                   |
| 4265 | 026514 | 047117 | 042524 | 052116 |       |        |       |         |        |              |                   |
| 4266 | 026522 | 000123 |        |        |       |        |       |         |        |              |                   |
| 4267 | 026524 | 042524 | 052123 | 021440 | DH17: | .ASCIZ | /TEST | ERR PC/ |        |              |                   |
| 4268 | 026532 | 020040 | 051105 | 020122 |       |        |       |         |        |              |                   |
| 4269 | 026540 | 041520 | 000    |        |       |        |       |         |        |              |                   |
| 4270 | 026543 | 040    | 020040 | 020040 | DH24: | .ASCII | /     |         | LOCKED | SWITCHED TO/ | <15><12>          |
| 4271 | 026550 | 020040 | 020040 | 020040 |       |        |       |         |        |              |                   |
| 4272 | 026556 | 020040 | 020040 | 046040 |       |        |       |         |        |              |                   |
| 4273 | 026564 | 041517 | 042513 | 020104 |       |        |       |         |        |              |                   |
| 4274 | 026572 | 051440 | 044527 | 041524 |       |        |       |         |        |              |                   |
| 4275 | 026600 | 042510 | 020104 | 047524 |       |        |       |         |        |              |                   |
| 4276 | 026606 | 005015 |        |        |       |        |       |         |        |              |                   |
| 4277 | 026610 | 042524 | 052123 | 021440 |       | .ASCIZ | /TEST | ERR PC  | PORT   | PORT         | /                 |
| 4278 | 026616 | 020040 | 051105 | 020122 |       |        |       |         |        |              |                   |
| 4279 | 026624 | 041520 | 020040 | 047520 |       |        |       |         |        |              |                   |
| 4280 | 026632 | 052122 | 021440 | 020040 |       |        |       |         |        |              |                   |
| 4281 | 026640 | 047520 | 052122 | 021440 |       |        |       |         |        |              |                   |
| 4282 | 026646 | 000    |        |        |       |        |       |         |        |              |                   |
| 4283 | 026647 | 040    | 020040 | 020040 | DH30: | .ASCII | /     |         | SEIZE  | ERROR/       | <15><12>          |
| 4284 | 026654 | 020040 | 020040 | 020040 |       |        |       |         |        |              |                   |
| 4285 | 026662 | 020040 | 020040 | 051440 |       |        |       |         |        |              |                   |
| 4286 | 026670 | 044505 | 042532 | 020040 |       |        |       |         |        |              |                   |
| 4287 | 026676 | 042440 | 051122 | 051117 |       |        |       |         |        |              |                   |
| 4288 | 026704 | 005015 |        |        |       |        |       |         |        |              |                   |
| 4289 | 026706 | 042524 | 052123 | 021440 |       | .ASCIZ | /TEST | ERR PC  | PORT   | PORT         | REG ADR GOOD BAD/ |
| 4290 | 026714 | 020040 | 051105 | 020122 |       |        |       |         |        |              |                   |
| 4291 | 026722 | 041520 | 020040 | 047520 |       |        |       |         |        |              |                   |
| 4292 | 026730 | 052122 | 021440 | 020040 |       |        |       |         |        |              |                   |
| 4293 | 026736 | 047520 | 052122 | 021440 |       |        |       |         |        |              |                   |
| 4294 | 026744 | 020040 | 042522 | 020107 |       |        |       |         |        |              |                   |
| 4295 | 026752 | 042101 | 051122 | 047507 |       |        |       |         |        |              |                   |
| 4296 | 026760 | 042117 | 020040 | 020040 |       |        |       |         |        |              |                   |
| 4297 | 026766 | 040502 | 000104 |        |       |        |       |         |        |              |                   |
| 4298 | 026772 | 020040 | 020040 | 020040 | DH34: | .ASCII | /     |         | PORT A | PORT B/      | <15><12>          |
| 4299 | 027000 | 020040 | 020040 | 020040 |       |        |       |         |        |              |                   |
| 4300 | 027006 | 020040 | 020040 | 047520 |       |        |       |         |        |              |                   |
| 4301 | 027014 | 052122 | 040440 | 020040 |       |        |       |         |        |              |                   |
| 4302 | 027022 | 047520 | 052122 | 041040 |       |        |       |         |        |              |                   |
| 4303 | 027030 | 005015 |        |        |       |        |       |         |        |              |                   |
| 4304 | 027032 | 042524 | 052123 | 021440 |       | .ASCIZ | /TEST | ERR PC  | RHDS1  | RHDS1/       |                   |
| 4305 | 027040 | 020040 | 051105 | 020122 |       |        |       |         |        |              |                   |
| 4306 | 027046 | 041520 | 020040 | 044122 |       |        |       |         |        |              |                   |
| 4307 | 027054 | 051504 | 020061 | 020040 |       |        |       |         |        |              |                   |
| 4308 | 027062 | 044122 | 051504 | 000061 |       |        |       |         |        |              |                   |
| 4309 | 027070 | 020040 | 020040 | 020040 | DH35: | .ASCII | /     |         | RELSNG | ERROR/       | <15><12>          |
| 4310 | 027076 | 020040 | 020040 | 020040 |       |        |       |         |        |              |                   |
| 4311 | 027104 | 020040 | 020040 | 042522 |       |        |       |         |        |              |                   |
| 4312 | 027112 | 051514 | 043516 | 020040 |       |        |       |         |        |              |                   |
| 4313 | 027120 | 051105 | 047522 | 006522 |       |        |       |         |        |              |                   |



```

4370 027546 001234 001116 001230 DT30: .WORD TSTNUM, $ERRPC, SEIZPT, PTNBR, $SDADR, $GDDAT, $BDDAT, 0
4371 027554 001226 001122 001124
4372 027562 001126 000000
4373 027566 001234 001116 001162 DT34: .WORD TSTNUM, $ERRPC, $TMP2, $TMP3, 0
4374 027574 001164 000000
4375 027600 001234 001116 001230 DT35: .WORD TSTNUM, $ERRPC, SEIZPT, PTNBR, 0
4376 027606 001226 000000
4377 027612 001234 001116 001230 DT40: .WORD TSTNUM, $ERRPC, SEIZPT, OPPRT, 0
4378 027620 001232 000000
4379
4380 027624 000 000 001 DF1: .BYTE 0,0,1,0,0
4381 027627 000 000
4382 027631 000 000 001 DF2: .BYTE 0,0,1,0,0,0
4383 027634 000 000 000
4384 027637 000 000 000 DF5: .BYTE 0,0,0,0,0
4385 027642 000 000
4386 027644 000 000 001 DF6: .BYTE 0,0,1
4387 027647 000 000 001 DF7: .BYTE 0,0,1,1
4388 027652 001
4389 027653 000 000 001 DF14: .BYTE 0,0,1,1,0,0
4390 027656 001 000 000
4391 027661 000 000 DF17: .BYTE 0,0
4392 027663 000 000 001 DF30: .BYTE 0,0,1,1,0,0,0
4393 027666 001 000 000
4394 027671 000
4395 027672 000 000 000 DF34: .BYTE 0,0,0,0
4396 027675 000

```

.EVEN

;;\*\*\*\*\*

.SBTTL CONSTANTS, TABLES, ETC

;;\*\*\*\*\*

;TABLE OF TEST STARTING ADDRESSES

```

4409 027676 002562 TSTADR: .WORD TST1+2 ;STARTING ADDRESS OF TEST 1
4410 027700 004114 .WORD TST2+2 ;STARTING ADDRESS OF TEST 2
4411 027702 004724 .WORD TST3+2 ;STARTING ADDRESS OF TEST 3
4412 027704 005534 .WORD TST4+2 ;STARTING ADDRESS OF TEST 4
4413 027706 006476 .WORD TST5+2 ;STARTING ADDRESS OF TEST 5
4414 027710 007440 .WORD TST6+2 ;STARTING ADDRESS OF TEST 6
4415 027712 011026 .WORD TST7+2 ;STARTING ADDRESS OF TEST 7
4416 027714 012414 .WORD TST10+2 ;STARTING ADDRESS OF TEST 10
4417 027716 013304 .WORD TST11+2 ;STARTING ADDRESS OF TEST 11
4418 027720 014166 .WORD TST12+2 ;STARTING ADDRESS OF TEST 12
4419 027722 015224 .WORD TST13+2 ;STARTING ADDRESS OF TEST 13
4420 027724 016316 .WORD TST14+2 ;STARTING ADDRESS OF TEST 14

```

;ATTENTION BIT TABLE

```

4424 027726 001 ATABIT: .BYTE 1 ;ATTENTION BIT FOR DRIVE 0
4425 027727 002 .BYTE 2 ;ATTENTION BIT FOR DRIVE 1

```

|      |        |        |        |       |        |  |                            |
|------|--------|--------|--------|-------|--------|--|----------------------------|
| 4426 | 027730 | 004    |        | .BYTE | 4      |  | ;ATTENTION BIT FOR DRIVE 2 |
| 4427 | 027731 | 010    |        | .BYTE | 10     |  | ;ATTENTION BIT FOR DRIVE 3 |
| 4428 | 027732 | 020    |        | .BYTE | 20     |  | ;ATTENTION BIT FOR DRIVE 4 |
| 4429 | 027733 | 040    |        | .BYTE | 40     |  | ;ATTENTION BIT FOR DRIVE 5 |
| 4430 | 027734 | 100    |        | .BYTE | 100    |  | ;ATTENTION BIT FOR DRIVE 6 |
| 4431 | 027735 | 200    |        | .BYTE | 200    |  | ;ATTENTION BIT FOR DRIVE 7 |
| 4432 |        |        |        |       |        |  |                            |
| 4433 | 027736 | 000014 | MAXTN: | .WORD | \$TN-1 |  | ;MAXIMUM TEST NUMBER       |
| 4434 |        |        |        |       |        |  |                            |
| 4435 |        | 000001 |        | .END  |        |  |                            |













|                |       |       |       |       |       |       |       |       |       |       |       |       |       |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | 1933* | 1934* | 1942* | 1951  | 1952  | 2089* | 2090* | 2091* | 2093* | 2095* | 2159* | 2160* | 2163* |
|                | 2164* | 2172* | 2181  | 2182  | 2319* | 2320* | 2321* | 2323* | 2325* | 2383* | 2384* | 2387* | 2388* |
|                | 2392* | 2401* | 2402* | 2403* | 2468* | 2469* | 2471* | 2518* | 2519* | 2522* | 2523* | 2527* | 2536* |
|                | 2537* | 2538* | 2601* | 2602* | 2604* | 2647* | 2648* | 2651* | 2652* | 2688* | 2689* | 2737* | 2738* |
|                | 2850* | 2851* | 2944* | 2945* | 2946* | 2949* | 2951* | 3046* | 3047* | 3138* | 3139* | 3140* | 3143* |
| RHCS2 = 000010 | 3145* |       |       |       |       |       |       |       |       |       |       |       |       |
|                | 608#  | 1140* | 1144  | 1145  | 1160* | 1161* | 1165  | 1166  | 1181* | 1186* | 1198* | 1214* | 1246* |
|                | 1282* | 1284* | 1338* | 1371* | 1381* | 1385* | 1397* | 1402* | 1463* | 1496* | 1506* | 1510* | 1522* |
|                | 1527* | 1604* | 1607* | 1618* | 1633* | 1664* | 1668* | 1680* | 1685* | 1763* | 1766* | 1777* | 1792* |
|                | 1823* | 1827* | 1839* | 1844* | 1927* | 1931* | 1935* | 1992* | 2009* | 2013* | 2025* | 2030* | 2051* |
|                | 2068* | 2092* | 2157* | 2161* | 2165* | 2222* | 2239* | 2243* | 2255* | 2260* | 2281* | 2298* | 2322* |
|                | 2381* | 2385* | 2389* | 2411* | 2415* | 2427* | 2432* | 2466* | 2516* | 2520* | 2524* | 2546* | 2550* |
|                | 2562* | 2567* | 2599* | 2645* | 2649* | 2663* | 2667* | 2679* | 2684* | 2712* | 2716* | 2728* | 2733* |
|                | 2810* | 2858* | 2876  | 2877  | 2896* | 2900* | 2934* | 2947* | 3006* | 3054* | 3072  | 3073  | 3092* |
|                | 3096* | 3128* | 3141* |       |       |       |       |       |       |       |       |       |       |
| RHDA = 000006  | 607#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RHDB = 000022  | 613#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RHDS1 = 000012 | 609#  | 1142  | 1163  | 1217  | 1218  | 1232  | 1233  | 1249  | 1250  | 1264  | 1265  | 1352  | 1353  |
|                | 1378  | 1382  | 1386  | 1477  | 1478  | 1503  | 1507  | 1511  | 1606* | 1610  | 1612  | 1620  | 1635  |
|                | 1661  | 1665  | 1669  | 1765* | 1769  | 1771  | 1779  | 1794  | 1820  | 1824  | 1828  | 1928* | 1932* |
|                | 1966  | 1967  | 1989  | 1994  | 2006  | 2010  | 2014  | 2054  | 2055  | 2071  | 2072  | 2158* | 2162* |
|                | 2196  | 2197  | 2219  | 2224  | 2236  | 2240  | 2244  | 2284  | 2285  | 2301  | 2302  | 2382* | 2386* |
|                | 2399  | 2408  | 2412  | 2416  | 2517* | 2521* | 2534  | 2543  | 2547  | 2551  | 2646* | 2650* | 2660  |
|                | 2664  | 2668  | 2709  | 2713  | 2717  | 2817  | 2825  | 2832  | 2833  | 2861  | 2862  | 2890* | 2898* |
|                | 2903  | 2904  | 2936  | 2938  | 3013  | 3021  | 3028  | 3029  | 3057  | 3058  | 3086* | 3094* | 3099  |
|                | 3100  | 3130  | 3132  |       |       |       |       |       |       |       |       |       |       |
| RHDT = 000026  | 615#  | 1189  | 1190  | 1201  | 1202  |       |       |       |       |       |       |       |       |
| RHEC1 = 000044 | 622#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RHEC2 = 000046 | 623#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RHER1 = 000014 | 610#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RHER2 = 000040 | 620#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RHER3 = 000042 | 621#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RHLA = 000020  | 612#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RHMR = 000024  | 614#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RHOF = 000032  | 617#  | 1346* | 1471* | 2094* | 2324* | 2470* | 2603* | 2852* | 2950* | 3048* | 3144* |       |       |
| RHSN = 000030  | 616#  | 1283  | 1285  |       |       |       |       |       |       |       |       |       |       |
| RHWC = 000002  | 605#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RMR = 000004   | 480#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RP6 = 000300   | 330#  |       |       |       |       |       |       |       |       |       |       |       |       |
| RO = %000000   | 312#  | 1045* | 1088* | 1089* | 1090  | 1093  | 1105* | 1140* | 1142  | 1144  | 1146  | 1160* | 1161* |
|                | 1163  | 1165  | 1167  | 1181* | 1186* | 1189  | 1191  | 1198* | 1201  | 1203  | 1214* | 1217  | 1219  |
|                | 1232  | 1234  | 1246* | 1249  | 1251  | 1264  | 1266  | 1282* | 1283  | 1284* | 1285  | 1338* | 1344* |
|                | 1345* | 1346* | 1352  | 1354  | 1371* | 1373* | 1379  | 1381* | 1382  | 1385* | 1386  | 1397* | 1402* |
|                | 1463* | 1469* | 1470* | 1471* | 1477  | 1479  | 1496* | 1498* | 1504  | 1506* | 1507  | 1510* | 1511  |
|                | 1522* | 1527* | 1604* | 1606* | 1607* | 1610  | 1611  | 1618* | 1620  | 1633* | 1635  | 1662  | 1664* |
|                | 1665  | 1668* | 1669  | 1680* | 1685* | 1689* | 1690* | 1763* | 1765* | 1766* | 1769  | 1770  | 1777* |
|                | 1779  | 1792* | 1794  | 1821  | 1823* | 1824  | 1827* | 1828  | 1839* | 1844* | 1848* | 1849* | 1927* |
|                | 1928* | 1929* | 1930* | 1931* | 1932* | 1933* | 1934* | 1935* | 1942* | 1951  | 1953  | 1966  | 1968  |
|                | 1989  | 1992* | 1994  | 2007  | 2009* | 2010  | 2013* | 2014  | 2025* | 2030* | 2051* | 2054  | 2056  |
|                | 2068* | 2071  | 2073  | 2089* | 2090* | 2091* | 2092* | 2093* | 2094* | 2095* | 2157* | 2158* | 2159* |
|                | 2160* | 2161* | 2162* | 2163* | 2164* | 2165* | 2172* | 2181  | 2183  | 2196  | 2198  | 2219  | 2222* |
|                | 2224  | 2237  | 2239* | 2240  | 2243* | 2244  | 2255* | 2260* | 2281* | 2284  | 2286  | 2298* | 2301  |
|                | 2303  | 2319* | 2320* | 2321* | 2322* | 2323* | 2324* | 2325* | 2381* | 2382* | 2383* | 2384* | 2385* |
|                | 2386* | 2387* | 2388* | 2389* | 2392* | 2399  | 2401* | 2402* | 2403* | 2409  | 2411* | 2412  | 2415* |
|                | 2416  | 2427* | 2432* | 2466* | 2468* | 2469* | 2470* | 2471* | 2516* | 2517* | 2518* | 2519* | 2520* |
|                | 2521* | 2522* | 2523* | 2524* | 2527* | 2534  | 2536* | 2537* | 2538* | 2544  | 2546* | 2547  | 2550* |



|                 |      |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
|-----------------|------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|--|--|--|--|--|--|--|--|
| SW00 = DCC001   | 349# | 359   |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW01 = 000002   | 348# | 358   |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW02 = 000004   | 347# | 357   |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW03 = 000010   | 346# | 356   |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW04 = 000020   | 345# | 355   |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW05 = 000040   | 344# | 354   |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW06 = 000100   | 343# | 353   |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW07 = 000200   | 342# | 352   |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW08 = 000400   | 341# | 351   |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW09 = 001000   | 340# | 350   | 1298  | 1425  | 1550  | 1711  | 1870 | 2101 | 2331 | 2455 | 2590 | 2757 | 2922 |  |  |  |  |  |  |  |  |
|                 | 3118 |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW1 = 000002    | 358# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW10 = 002000   | 339# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW11 = 004000   | 338# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW12 = 010000   | 337# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW13 = 020000   | 336# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW14 = 040000   | 335# | 2462  | 2595  | 2929  | 3123  |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW15 = 100000   | 334# | 1289  |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW16 = 000004   | 337# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW17 = 000010   | 336# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW18 = 000020   | 335# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW19 = 000040   | 334# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW20 = 000100   | 333# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW21 = 000200   | 332# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW22 = 000400   | 331# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| SW23 = 001000   | 330# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TBITVE = 000014 | 392# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TOP = 000040    | 548# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TESTNO = 022314 | 1054 | 3838# |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST1 = 002634  | 1128 | 1129  | 1135# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST10 = 012466 | 2369 | 2370  | 2377# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST11 = 013356 | 2504 | 2505  | 2512# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST12 = 014240 | 2633 | 2634  | 2641# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST13 = 015276 | 2801 | 2802  | 2809# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST14 = 016370 | 2997 | 2998  | 3005# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST2 = 004166  | 1329 | 1330  | 1337# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST3 = 004776  | 1454 | 1455  | 1462# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST4 = 005606  | 1582 | 1583  | 1590# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST5 = 006550  | 1741 | 1742  | 1749# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST6 = 007512  | 1915 | 1916  | 1923# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TEST7 = 011100  | 2145 | 2146  | 2153# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TIME = 001244   | 713# | 1597* | 1641  | 1756* | 1800  | 3181* | 4360 |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TIMEA = 001250  | 715# | 1030* | 1591* | 1641* | 1643  | 1649  |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TIMEAP = 001252 | 716# | 1031* | 1592* | 1644* | 1943  |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TIMEB = 001254  | 717# | 1032* | 1750* | 1800* | 1802  | 1808  |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TIMEBP = 001256 | 718# | 1033* | 1751* | 1803* | 2173  |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TITLE = 022014  | 1002 | 3802# |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TKVEC = 000060  | 399# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TOLER = 017562  | 1642 | 1801  | 3191# |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TPVEC = 000064  | 400# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TRAPVE = 000034 | 398# | 993*  | 994*  | 1076* | 1077* |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TRE = 040000    | 454# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TRK1 = 004000   | 535# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TRK10 = 040000  | 538# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TRK2 = 010000   | 536# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |
| TRK20 = 100000  | 539# |       |       |       |       |       |      |      |      |      |      |      |      |  |  |  |  |  |  |  |  |

|          |        |       |       |       |       |       |       |       |       |       |       |       |       |       |
|----------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| TRK4 =   | 020000 | 537#  |       |       |       |       |       |       |       |       |       |       |       |       |
| TRTVEC = | 000014 | 393#  |       |       |       |       |       |       |       |       |       |       |       |       |
| TSTADR   | 027676 | 1065  | 4409# |       |       |       |       |       |       |       |       |       |       |       |
| TSTNUM   | 001234 | 709#  | 3313# | 4351  | 4353  | 4356  | 4358  | 4360  | 4362  | 4364  | 4367  | 4368  | 4370  | 4373  |
|          |        | 4375  | 4377  |       |       |       |       |       |       |       |       |       |       |       |
| TST1     | 002560 | 1058  | 1120# | 4409  |       |       |       |       |       |       |       |       |       |       |
| TST1AA   | 002554 | 1105# | 3246  |       |       |       |       |       |       |       |       |       |       |       |
| TST10    | 012412 | 2330  | 2332  | 2361# | 4416  |       |       |       |       |       |       |       |       |       |
| TST11    | 013302 | 2472  | 2496# | 4417  |       |       |       |       |       |       |       |       |       |       |
| TST12    | 014164 | 2605  | 2625# | 4418  |       |       |       |       |       |       |       |       |       |       |
| TST13    | 015222 | 2756  | 2758  | 2793# | 4419  |       |       |       |       |       |       |       |       |       |
| TST14    | 016314 | 2953  | 2989# | 4420  |       |       |       |       |       |       |       |       |       |       |
| TST2     | 004112 | 1297  | 1299  | 1321# | 4410  |       |       |       |       |       |       |       |       |       |
| TST3     | 004722 | 1424  | 1426  | 1446# | 4411  |       |       |       |       |       |       |       |       |       |
| TST4     | 005532 | 1549  | 1551  | 1574# | 4412  |       |       |       |       |       |       |       |       |       |
| TST5     | 006474 | 1710  | 1712  | 1733# | 4413  |       |       |       |       |       |       |       |       |       |
| TST6     | 007436 | 1869  | 1871  | 1907# | 4414  |       |       |       |       |       |       |       |       |       |
| TST7     | 011024 | 2100  | 2102  | 2137# | 4415  |       |       |       |       |       |       |       |       |       |
| TUF =    | 000100 | 549#  |       |       |       |       |       |       |       |       |       |       |       |       |
| TYPOS =  | 104410 | 1020  | 1023  | 1983  | 2213  | 2397  | 2532  | 3233  | 3378  | 3789# |       |       |       |       |
| TYPE =   | 104400 | 1002  | 1005  | 1010  | 1018  | 1021  | 1024  | 1036  | 1044  | 1054  | 1061  | 1078  | 1081  | 1082  |
|          |        | 1092  | 1980  | 1984  | 2210  | 2214  | 2393  | 2394  | 2398  | 2464  | 2528  | 2529  | 2533  | 2597  |
|          |        | 2653  | 2654  | 2702  | 2703  | 2751  | 2812  | 2819  | 2820  | 2931  | 2932  | 2933  | 2952  | 3008  |
|          |        | 3015  | 3016  | 3125  | 3126  | 3127  | 3231  | 3234  | 3319  | 3327  | 3342  | 3359  | 3361  | 3364  |
|          |        | 3366  | 3381  | 3386  | 3498  | 3574  | 3619  | 3625  | 3630  | 3634  | 3639  | 3640  | 3642  | 3645  |
|          |        | 3649  | 3709  | 3711  | 3785# |       |       |       |       |       |       |       |       |       |
| TYPOC =  | 104402 | 1080  | 1094  | 3350  | 3374  | 3786# |       |       |       |       |       |       |       |       |
| TYPON =  | 104406 | 3788# |       |       |       |       |       |       |       |       |       |       |       |       |
| TYPOS =  | 104404 | 3787# |       |       |       |       |       |       |       |       |       |       |       |       |
| UNS =    | 040000 | 492#  |       |       |       |       |       |       |       |       |       |       |       |       |
| LPE =    | 020000 | 430#  |       |       |       |       |       |       |       |       |       |       |       |       |
| US1 =    | 000001 | 417#  |       |       |       |       |       |       |       |       |       |       |       |       |
| US2 =    | 000002 | 418#  |       |       |       |       |       |       |       |       |       |       |       |       |
| US4 =    | 000004 | 419#  |       |       |       |       |       |       |       |       |       |       |       |       |
| UMR =    | 000010 | 585#  |       |       |       |       |       |       |       |       |       |       |       |       |
| VUF =    | 000002 | 584#  |       |       |       |       |       |       |       |       |       |       |       |       |
| VU30 =   | 010000 | 555#  |       |       |       |       |       |       |       |       |       |       |       |       |
| VV =     | 000100 | 465#  | 1355  | 1384  | 1388  | 1409  | 1415  | 1480  | 1509  | 1513  | 1534  | 1540  | 1621  | 1663  |
|          |        | 1667  | 1671  | 1780  | 1822  | 1826  | 1830  | 1969  | 2012  | 2016  | 2037  | 2043  | 2199  | 2242  |
|          |        | 2246  | 2267  | 2273  | 2414  | 2418  | 2439  | 2445  | 2549  | 2553  | 2574  | 2580  | 2662  | 2666  |
|          |        | 2670  | 2711  | 2715  | 2719  |       |       |       |       |       |       |       |       |       |
| VVSET =  | 000001 | 982#  | 1380  | 1406  | 1409  | 1415  | 1505  | 1531  | 1534  | 1540  | 1556# | 1621  | 1663  | 1689  |
|          |        | 1691  | 1694  | 1695  | 1700  | 1701  | 1780  | 1822  | 1848  | 1850  | 1853  | 1854  | 1859  | 1860  |
|          |        | 1999# | 2008  | 2034  | 2037  | 2043  | 2096# | 2229# | 2238  | 2264  | 2267  | 2273  | 2326# | 2404# |
|          |        | 2410  | 2436  | 2439  | 2445  | 2472# | 2539# | 2545  | 2571  | 2574  | 2580  | 2605# | 2662  | 2688  |
|          |        | 2693  | 2698  | 2711  | 2737  | 2742  | 2747  |       |       |       |       |       |       |       |
| WATCH    | 001246 | 714#  | 1598# | 1637  | 1757* | 1796  | 1943* | 1944  | 1991* | 1996  | 2173* | 2174  | 2221* | 2226  |
|          |        | 3182  | 3184* | 3186* |       |       |       |       |       |       |       |       |       |       |
| WCE =    | 040000 | 431#  |       |       |       |       |       |       |       |       |       |       |       |       |
| WCF =    | 000040 | 483#  |       |       |       |       |       |       |       |       |       |       |       |       |
| WCU =    | 000001 | 543#  |       |       |       |       |       |       |       |       |       |       |       |       |
| WLE =    | 004000 | 489#  |       |       |       |       |       |       |       |       |       |       |       |       |
| WPL =    | 004000 | 470#  |       |       |       |       |       |       |       |       |       |       |       |       |
| WRU =    | 000400 | 551#  |       |       |       |       |       |       |       |       |       |       |       |       |
| WSU =    | 000004 | 545#  |       |       |       |       |       |       |       |       |       |       |       |       |
| WBOADR   | 001122 | 671#  | 1145* | 1146* | 1166* | 1167* | 1190* | 1191* | 1202* | 1203* | 1218* | 1219* | 1233* | 1234* |

|        |          |       |       |       |       |       |       |       |       |       |       |       |       |       |
|--------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|        |          | 1250* | 1251* | 1265* | 1266* | 1353* | 1354* | 1378* | 1379* | 1478* | 1479* | 1503* | 1504* | 1611* |
|        |          | 1612* | 1661* | 1662* | 1770* | 1771* | 1820* | 1821* | 1952* | 1953* | 1967* | 1968* | 2006* | 2007* |
|        |          | 2055* | 2056* | 2072* | 2073* | 2182* | 2183* | 2197* | 2198* | 2236* | 2237* | 2285* | 2286* | 2302* |
|        |          | 2303* | 2408* | 2409* | 2543* | 2544* | 2660* | 2661* | 2709* | 2710* | 2833* | 2834* | 2862* | 2863* |
|        |          | 2877* | 2878* | 2904* | 2905* | 3029* | 3030* | 3058* | 3059* | 3073* | 3074* | 3100* | 3101* | 4351  |
|        |          | 4353  | 4356  | 4364  | 4370  |       |       |       |       |       |       |       |       |       |
| SEDDAT | 001126   | 673*  | 1144* | 1148  | 1152  | 1165* | 1169  | 1173  | 1189* | 1193  | 1201* | 1205  | 1217* | 1221  |
|        |          | 1225  | 1232* | 1236  | 1240  | 1249* | 1253  | 1257  | 1264* | 1268  | 1272  | 1285* | 1286  | 1352* |
|        |          | 1356  | 1360  | 1395* | 1401* | 1407* | 1413* | 1477* | 1481  | 1485  | 1520* | 1526* | 1532* | 1538* |
|        |          | 1610* | 1614  | 1620* | 1624  | 1678* | 1684* | 1692* | 1698* | 1769* | 1773  | 1779* | 1783  | 1837* |
|        |          | 1843* | 1851* | 1857* | 1951* | 1955  | 1959  | 1966* | 1970  | 1974  | 2023* | 2029* | 2035* | 2041* |
|        |          | 2054* | 2058  | 2062  | 2071* | 2075  | 2079  | 2181* | 2185  | 2189  | 2196* | 2200  | 2204  | 2253* |
|        |          | 2259* | 2265* | 2271* | 2284* | 2288  | 2292  | 2301* | 2305  | 2309  | 2425* | 2431* | 2437* | 2443* |
|        |          | 2560* | 2566* | 2572* | 2578* | 2677* | 2683* | 2691* | 2696* | 2726* | 2732* | 2740* | 2745* | 2832* |
|        |          | 2836  | 2840  | 2861* | 2865  | 2869  | 2876* | 2880  | 2884  | 2903* | 2907  | 2911  | 3028* | 3032  |
|        |          | 3036  | 3057* | 3061  | 3065  | 3072* | 3076  | 3080  | 3099* | 3103  | 3107  | 4351  | 4353  | 4356  |
|        |          | 4364  | 4370  |       |       |       |       |       |       |       |       |       |       |       |
| SBELL  | 001174   | 693*  | 3319  | 3333  |       |       |       |       |       |       |       |       |       |       |
| SCHTAG | 001100   | 659*  | 983   | 984   | 991   | 995   | 996   | 997   |       |       |       |       |       |       |
| SCH1   | = 000001 | 685*  | 686*  |       |       |       |       |       |       |       |       |       |       |       |
| SCH2   | = 000002 | 685*  | 686*  |       |       |       |       |       |       |       |       |       |       |       |
| SCH3   | = 000001 | 683*  | 685   |       |       |       |       |       |       |       |       |       |       |       |
| SCH4   | = 000005 | 686*  | 687*  | 688*  | 689*  | 690*  | 691*  |       |       |       |       |       |       |       |
| SCNTLU | 021500   | 3634  | 3658* |       |       |       |       |       |       |       |       |       |       |       |
| SCRFL  | 001201   | 695*  | 1024  | 1044  | 1081  | 1984  | 2214  | 2398  | 2533  | 2952  | 3327  | 3333  | 3342  | 3361  |
|        |          | 3366  | 3386  | 3639  | 3660  | 3714  |       |       |       |       |       |       |       |       |
| SOBLK  | 021200   | 3540  | 3574  | 3582* |       |       |       |       |       |       |       |       |       |       |
| SDCAGN | 017736   | 3227  | 3236  | 3240  | 3246* |       |       |       |       |       |       |       |       |       |
| SDTFL  | 021170   | 3543  | 3578* |       |       |       |       |       |       |       |       |       |       |       |
| SENDAD | 017726   | 653   | 655   | 3237  | 3238  | 3242* |       |       |       |       |       |       |       |       |
| SENDCT | 017656   | 995   | 3229* |       |       |       |       |       |       |       |       |       |       |       |
| SENDMG | 017742   | 3231  | 3247* |       |       |       |       |       |       |       |       |       |       |       |
| SENULL | 017757   | 3234  | 3250* |       |       |       |       |       |       |       |       |       |       |       |
| SEOP   | 017610   | 3146  | 3216* |       |       |       |       |       |       |       |       |       |       |       |
| SEOPCT | 017650   | 995*  | 3226* | 3230  |       |       |       |       |       |       |       |       |       |       |
| SERFLG | 001103   | 662*  | 1296  | 1300* | 1423  | 1427* | 1548  | 1552* | 1709  | 1713* | 1868  | 1872* | 2099  | 2103* |
|        |          | 2329  | 2333* | 2453  | 2457* | 2588  | 2592* | 2755  | 2759* | 2920  | 2924* | 3116  | 3120* | 3257  |
|        |          | 3279  | 3281* | 3298  | 3314* | 3333  |       |       |       |       |       |       |       |       |
| SERMAX | 001115   | 668*  | 998*  | 1051* | 3298  |       |       |       |       |       |       |       |       |       |
| SERROR | 020140   | 991   | 3312* |       |       |       |       |       |       |       |       |       |       |       |
| SERRPC | 001116   | 669*  | 3321* | 3322* | 3323  | 3333  | 3348  | 4351  | 4353  | 4356  | 4358  | 4360  | 4362  | 4364  |
|        |          | 4367  | 4368  | 4370  | 4373  | 4375  | 4377  |       |       |       |       |       |       |       |
| SERRTB | 001266   | 746*  | 3356  |       |       |       |       |       |       |       |       |       |       |       |
| SERTY  | 020254   | 3326  | 3341* |       |       |       |       |       |       |       |       |       |       |       |
| SERTTL | 001112   | 666*  | 3320* | 3333  |       |       |       |       |       |       |       |       |       |       |
| SESCAP | 001172   | 692*  | 997*  |       |       |       |       |       |       |       |       |       |       |       |
| SFILLC | 001150   | 681*  | 3425  | 3437  |       |       |       |       |       |       |       |       |       |       |
| SFILLS | 001147   | 680*  | 3437  |       |       |       |       |       |       |       |       |       |       |       |
| SGADR  | 001120   | 670*  |       |       |       |       |       |       |       |       |       |       |       |       |
| SGOAT  | 001124   | 672*  | 1147* | 1150  | 1154* | 1168* | 1171  | 1175* | 1192* | 1193  | 1204* | 1205  | 1220* | 1223  |
|        |          | 1227* | 1235* | 1238  | 1242* | 1252* | 1255  | 1259* | 1267* | 1270  | 1274* | 1283* | 1286  | 1355* |
|        |          | 1358  | 1362* | 1380* | 1410  | 1416  | 1480* | 1483  | 1487* | 1505* | 1535  | 1541  | 1613* | 1614  |
|        |          | 1621* | 1622  | 1626  | 1663* | 1695  | 1701  | 1772* | 1773  | 1780* | 1781  | 1785  | 1822* | 1854  |
|        |          | 1860  | 1954* | 1957  | 1961* | 1969* | 1972  | 1976* | 2008* | 2038  | 2044  | 2057* | 2060  | 2064* |
|        |          | 2074* | 2077  | 2081* | 2184* | 2187  | 2191* | 2199* | 2202  | 2206* | 2238* | 2268  | 2274  | 2287* |
|        |          | 2290  | 2294* | 2304* | 2307  | 2311* | 2410* | 2440  | 2446  | 2545* | 2575  | 2581  | 2662* | 2693  |



|         |           |       |       |       |       |       |       |       |       |       |       |       |       |       |
|---------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|         |           | 2698  | 2711* | 2742  | 2747  | 2835* | 2838  | 2842* | 2864* | 2867  | 2871* | 2879* | 2882  | 2886* |
|         |           | 2906* | 2909  | 2913* | 3031* | 3034  | 3038* | 3060* | 3063  | 3067* | 3075* | 3078  | 3082* | 3102* |
|         |           | 3105  | 3109* | 4353  | 4356  | 4370  |       |       |       |       |       |       |       |       |
| \$GET42 | 017700    | 3235# |       |       |       |       |       |       |       |       |       |       |       |       |
| \$HD    | = 000000  | 283   |       |       |       |       |       |       |       |       |       |       |       |       |
| \$HIOCT | 021652    | 3702* | 3713# |       |       |       |       |       |       |       |       |       |       |       |
| \$ICNT  | 001104    | 663#  | 1067* | 3287* | 3288  | 3290* | 3297  |       |       |       |       |       |       |       |
| \$ITEMB | 001114    | 667#  | 3323* | 3333  | 3345  |       |       |       |       |       |       |       |       |       |
| \$LF    | 001202    | 696#  | 3333  | 3649  | 3660  | 3714  |       |       |       |       |       |       |       |       |
| \$LKCSB | 001206    | 698#  | 3163* |       |       |       |       |       |       |       |       |       |       |       |
| \$LKCSR | 001204    | 697#  | 3159  | 3164* |       |       |       |       |       |       |       |       |       |       |
| \$LKS   | 001212    | 700#  | 3168  | 3172* |       |       |       |       |       |       |       |       |       |       |
| \$LVEC  | 001214    | 701#  | 3169  |       |       |       |       |       |       |       |       |       |       |       |
| \$LPADR | 001106    | 664#  | 999*  | 1052* | 1128* | 1329* | 1454* | 1582* | 1741* | 1915* | 2145* | 2369* | 2504* | 2633* |
|         |           | 2801* | 2997* | 3293* | 3295  | 3297  |       |       |       |       |       |       |       |       |
| \$LPERR | 001110    | 665#  | 1000* | 1053* | 1129* | 1302  | 1330* | 1429  | 1455* | 1554  | 1583* | 1715  | 1742* | 1874  |
|         |           | 1916* | 2105  | 2146* | 2335  | 2370* | 2459  | 2505* | 2594  | 2634* | 2761  | 2802* | 2926  | 2998* |
|         |           | 3122  |       |       |       |       |       |       |       |       |       |       |       |       |
| \$LPVEC | 001210    | 699#  | 3160  |       |       |       |       |       |       |       |       |       |       |       |
| \$MXCNT | 020136    | 3291  | 3297# |       |       |       |       |       |       |       |       |       |       |       |
| \$NULL  | 001146    | 679#  | 3427  | 3437  |       |       |       |       |       |       |       |       |       |       |
| \$NWTST | = 000001  | 1107# | 1109  | 1306# | 1308  | 1431# | 1433  | 1559# | 1561  | 1718# | 1720  | 1877# | 1879  | 2107# |
|         |           | 2109  | 2339# | 2341  | 2474# | 2476  | 2608# | 2610  | 2765# | 2767  | 2955# | 2957  |       |       |
| \$OCNT  | 020760    | 3470* | 3499# | 3512# |       |       |       |       |       |       |       |       |       |       |
| \$OMODE | 020762    | 3465* | 3469# | 3474  | 3477* | 3488* | 3514# |       |       |       |       |       |       |       |
| \$OVER  | 020122    | 3266  | 3277  | 3289  | 3294# |       |       |       |       |       |       |       |       |       |
| \$PASS  | 001100    | 660#  | 1050* | 3223* | 3224* | 3232  | 3247  | 3285  | 3298  |       |       |       |       |       |
| \$QUES  | 001200    | 694#  | 3333  | 3642  | 3660  | 3711  | 3714  |       |       |       |       |       |       |       |
| \$ROCHR | 021210    | 3593# | 3790  |       |       |       |       |       |       |       |       |       |       |       |
| \$RODEC | = ##### U | 3793  |       |       |       |       |       |       |       |       |       |       |       |       |
| \$ROLIN | 021244    | 3607# | 3791  |       |       |       |       |       |       |       |       |       |       |       |
| \$ROOCT | 021514    | 3675# | 3792  |       |       |       |       |       |       |       |       |       |       |       |
| \$ROSZ  | = 000007  | 3600# |       |       |       |       |       |       |       |       |       |       |       |       |
| \$REGAD | 001152    | 683#  |       |       |       |       |       |       |       |       |       |       |       |       |
| \$REGO  | 001154    | 685#  |       |       |       |       |       |       |       |       |       |       |       |       |
| \$RESRE | 021712    | 3748# | 3794  |       |       |       |       |       |       |       |       |       |       |       |
| \$RPADR | 001262    | 727#  | 1045  | 1079  | 1086* | 1088  | 1105  |       |       |       |       |       |       |       |
| \$RPVEC | 001264    | 728#  |       |       |       |       |       |       |       |       |       |       |       |       |
| \$SAVRE | 021654    | 3732# | 3793  |       |       |       |       |       |       |       |       |       |       |       |
| \$SCOPE | 017762    | 989   | 3264# |       |       |       |       |       |       |       |       |       |       |       |
| \$SETUP | = 000027  | 982#  | 989   | 991   | 993   | 995   | 996   | 997   | 999   | 3221  |       |       |       |       |
| \$STUP  | = 177777  | 982#  |       |       |       |       |       |       |       |       |       |       |       |       |
| \$SVLAD | 020112    | 3274  | 3292# |       |       |       |       |       |       |       |       |       |       |       |
| \$SMR   | = 166000  | 272#  | 283   | 289   | 290   | 291   | 292   | 293   | 294   | 295   | 691   | 692   | 693   | 996   |
|         |           | 997   | 999   | 1000  | 1130  | 1331  | 1456  | 1584  | 1743  | 1917  | 2147  | 2371  | 2506  | 2635  |
|         |           | 2803  | 2999  | 3213  | 3222  | 3235  | 3247  | 3251# | 3258  | 3259  | 3260  | 3261  | 3265  | 3277  |
|         |           | 3279  | 3280  | 3281  | 3282  | 3283  | 3294  | 3297  | 3305  | 3306  | 3307  | 3308  | 3309  | 3317  |
|         |           | 3324  | 3328  | 3331  | 3333  |       |       |       |       |       |       |       |       |       |
|         |           | 3261  |       |       |       |       |       |       |       |       |       |       |       |       |
| \$SMRK  | = 000000  | 691#  | 996*  | 1130* | 1301* | 1331* | 1428* | 1456* | 1553* | 1584* | 1714* | 1743* | 1873* | 1917* |
| \$TIMES | 001170    | 2104* | 2147* | 2334* | 2371* | 2458* | 2506* | 2593* | 2635* | 2760* | 2803* | 2925* | 2999* | 3121* |
|         |           | 3222* | 3282* | 3288  | 3291* | 3297  |       |       |       |       |       |       |       |       |
|         |           | 676#  | 3587  | 3597  |       |       |       |       |       |       |       |       |       |       |
| \$TKB   | 001140    | 675#  | 986   | 3587  | 3595  |       |       |       |       |       |       |       |       |       |
| \$TKS   | 001136    | 686#  | 1148* | 1149* | 1150  | 1169* | 1170* | 1171  | 1221* | 1222* | 1223  | 1236* | 1237* | 1238  |
| \$TMPO  | 001156    | 1253* | 1254* | 1255  | 1268* | 1269* | 1270  | 1356* | 1357* | 1358  | 1383* | 1384* | 1389  | 1391  |

|         |          |       |       |       |       |       |       |       |       |       |       |       |       |       |
|---------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|         |          | 1398  | 1481* | 1482* | 1483  | 1508* | 1509* | 1514  | 1516  | 1523  | 1624* | 1625* | 1626  | 1666* |
|         |          | 1667* | 1672  | 1674  | 1681  | 1783* | 1784* | 1785  | 1825* | 1826* | 1831  | 1833  | 1840  | 1955* |
|         |          | 1956* | 1957  | 1970* | 1971* | 1972  | 2011* | 2012* | 2017  | 2019  | 2026  | 2058* | 2059* | 2060  |
|         |          | 2075* | 2076* | 2077  | 2185* | 2186* | 2187  | 2200* | 2201* | 2202  | 2241* | 2242* | 2247  | 2249  |
|         |          | 2256  | 2288* | 2289* | 2290  | 2305* | 2306* | 2307  | 2413* | 2414* | 2419  | 2421  | 2428  | 2548* |
|         |          | 2549* | 2554  | 2556  | 2563  | 2665* | 2666* | 2671  | 2673  | 2680  | 2714* | 2715* | 2720  | 2722  |
|         |          | 2729  | 2836* | 2837* | 2838  | 2865* | 2866* | 2867  | 2880* | 2881* | 2882  | 2907* | 2908* | 2909  |
|         |          | 3032* | 3033* | 3034  | 3061* | 3062* | 3063  | 3076* | 3077* | 3078  | 3103* | 3104* | 3105  |       |
| STMP1   | 001160   | 687*  | 1387* | 1388* | 1389  | 1403  | 1512* | 1513* | 1514  | 1528  | 1622* | 1623* | 1625  | 1670* |
|         |          | 1671* | 1672  | 1686  | 1781* | 1782* | 1784  | 1829* | 1830* | 1831  | 1845  | 2015* | 2016* | 2017  |
|         |          | 2031  | 2245* | 2246* | 2247  | 2261  | 2417* | 2418* | 2419  | 2433  | 2552* | 2553* | 2554  | 2568  |
| STMP2   | 001162   | 2669* | 2670* | 2671  | 2685  | 2718* | 2719* | 2720  | 2734  |       |       |       |       |       |
|         |          | 688*  | 1382* | 1383  | 1395  | 1407  | 1409* | 1410  | 1507* | 1508  | 1520  | 1532  | 1534* | 1535  |
|         |          | 1665* | 1666  | 1678  | 1692  | 1694* | 1695  | 1824* | 1825  | 1837  | 1851  | 1853* | 1854  | 2010* |
|         |          | 2011  | 2023  | 2035  | 2037* | 2038  | 2240* | 2241  | 2253  | 2265  | 2267* | 2268  | 2412* | 2413  |
|         |          | 2425  | 2437  | 2439* | 2440  | 2547* | 2548  | 2560  | 2572  | 2574* | 2575  | 2664* | 2665  | 2677  |
| STMP3   | 001164   | 2691  | 2693  | 2713* | 2714  | 2726  | 2740  | 2742  | 4373  |       |       |       |       |       |
|         |          | 689*  | 1386* | 1387  | 1401  | 1413  | 1415* | 1416  | 1511* | 1512  | 1526  | 1538  | 1540* | 1541  |
|         |          | 1669* | 1670  | 1684  | 1698  | 1700* | 1701  | 1828* | 1829  | 1843  | 1857  | 1859* | 1860  | 2014* |
|         |          | 2015  | 2029  | 2041  | 2043* | 2044  | 2244* | 2245  | 2259  | 2271  | 2273* | 2274  | 2416* | 2417  |
|         |          | 2431  | 2443  | 2445* | 2446  | 2551* | 2552  | 2566  | 2578  | 2580* | 2581  | 2668* | 2669  | 2683  |
| STMP4   | 001166   | 2696  | 2698  | 2717* | 2718  | 2732  | 2745  | 2747  | 4373  |       |       |       |       |       |
|         |          | 690*  | 1152* | 1153* | 1154  | 1173* | 1174* | 1175  | 1225* | 1226* | 1227  | 1240* | 1241* | 1242  |
|         |          | 1257* | 1258* | 1259  | 1272* | 1273* | 1274  | 1360* | 1361* | 1362  | 1485* | 1486* | 1487  | 1959* |
|         |          | 1960* | 1961  | 1974* | 1975* | 1976  | 2062* | 2063* | 2064  | 2079* | 2080* | 2081  | 2189* | 2190* |
|         |          | 2191  | 2204* | 2205* | 2206  | 2292* | 2293* | 2294  | 2309* | 2310* | 2311  | 2840* | 2841* | 2842  |
|         |          | 2869* | 2870* | 2871  | 2884* | 2885* | 2886  | 2911* | 2912* | 2913  | 3036* | 3037* | 3038  | 3065* |
| STN     | = 000015 | 3066* | 3067  | 3080* | 3081* | 3082  | 3107* | 3108* | 3109  |       |       |       |       |       |
|         |          | 272*  | 283   | 1107  | 1120  | 1130* | 1131  | 1297  | 1299  | 1306  | 1321  | 1331* | 1333  | 1424  |
|         |          | 1426  | 1431  | 1446  | 1456* | 1458  | 1549  | 1551  | 1559  | 1574  | 1584* | 1586  | 1710  | 1712  |
|         |          | 1718  | 1733  | 1743* | 1745  | 1869  | 1871  | 1877  | 1907  | 1917* | 1919  | 2100  | 2102  | 2107  |
|         |          | 2137  | 2147* | 2149  | 2330  | 2332  | 2339  | 2361  | 2371* | 2373  | 2472  | 2474  | 2496  | 2506* |
|         |          | 2508  | 2605  | 2608  | 2625  | 2635* | 2637  | 2756  | 2758  | 2765  | 2793  | 2803* | 2805  | 2953  |
|         |          | 2955  | 2989  | 2999* | 3001  | 4433  |       |       |       |       |       |       |       |       |
| STPB    | 001144   | 678*  | 3435* | 3437  |       |       |       |       |       |       |       |       |       |       |
| STPFLG  | 001151   | 682*  | 3412  | 3437  |       |       |       |       |       |       |       |       |       |       |
| STPS    | 001142   | 677*  | 3433  | 3437  |       |       |       |       |       |       |       |       |       |       |
| STRAP   | 021750   | 993   | 1076  | 3769* |       |       |       |       |       |       |       |       |       |       |
| STRP    | = 000024 | 3776* | 3786* | 3787* | 3788* | 3789* | 3790* | 3791* | 3792* | 3793* | 3794* | 3795* |       |       |
| STRPAD  | 021770   | 3773  | 3784* |       |       |       |       |       |       |       |       |       |       |       |
| STSTNM  | 001102   | 661*  | 1127* | 1328* | 1453* | 1581* | 1740* | 1914* | 2144* | 2368* | 2503* | 2632* | 2800* | 2996* |
|         |          | 3221* | 3257  | 3292* | 3294  | 3298  | 3313  | 3316  | 3333  |       |       |       |       |       |
| STTYIN  | 021505   | 3609  | 3610  | 3622  | 3640  | 3654  | 3659* |       |       |       |       |       |       |       |
| STYPBN= | ***** U  | 3790  |       |       |       |       |       |       |       |       |       |       |       |       |
| STYPOS  | 020764   | 3528* | 3789  |       |       |       |       |       |       |       |       |       |       |       |
| STYPE   | 020430   | 3412* | 3776  | 3785  |       |       |       |       |       |       |       |       |       |       |
| STYPEC  | 020520   | 3424  | 3431  | 3433* | 3434  |       |       |       |       |       |       |       |       |       |
| STYPOC  | 020562   | 3468* | 3786  |       |       |       |       |       |       |       |       |       |       |       |
| STYPON  | 020576   | 3467  | 3470* | 3788  |       |       |       |       |       |       |       |       |       |       |
| STYPOS  | 020536   | 3463* | 3787  |       |       |       |       |       |       |       |       |       |       |       |
| EXTSTR  | 017770   | 3268* |       |       |       |       |       |       |       |       |       |       |       |       |
| SSTRP   | = 000002 | 3775* | 3786  | 3787  | 3788  | 3789  | 3790  | 3791  | 3792  | 3793  | 3794  | 3795  |       |       |
| SOFILL  | 020761   | 3464* | 3468* | 3478  | 3513* |       |       |       |       |       |       |       |       |       |
| .       | = 027740 | 628*  | 632   | 634*  | 654*  | 657*  | 697   | 987   | 999   | 1000  | 1038  | 1052  | 1053  | 1057  |
|         |          | 1159  | 1180  | 1615  | 1627  | 1774  | 1786  | 3219  | 3247  | 3251  | 3297  | 3298  | 3333  | 3389* |
|         |          | 3437  | 3582* | 3587  | 3659* | 3660  | 3714  |       |       |       |       |       |       |       |







|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| CMP  | 986  | 1008 | 1059 | 1150 | 1171 | 1193 | 1205 | 1223 | 1238 | 1255 | 1270 | 1286 | 1358 | 1389 | 1410 |
|      | 1416 | 1483 | 1514 | 1535 | 1541 | 1614 | 1626 | 1649 | 1672 | 1695 | 1701 | 1773 | 1785 | 1808 | 1831 |
|      | 1854 | 1860 | 1957 | 1972 | 2017 | 2038 | 2044 | 2060 | 2077 | 2187 | 2202 | 2247 | 2268 | 2274 | 2290 |
|      | 2307 | 2419 | 2440 | 2446 | 2554 | 2575 | 2581 | 2671 | 2693 | 2698 | 2720 | 2742 | 2747 | 2838 | 2867 |
|      | 2822 | 2909 | 3034 | 3063 | 3078 | 3105 | 3237 | 3239 | 3275 | 3288 | 3560 | 3610 | 3622 |      |      |
| CMPB | 3425 | 3614 | 3632 | 3636 | 3646 | 3687 | 3689 |      |      |      |      |      |      |      |      |
| COM  | 1156 | 1177 | 1196 | 1208 | 1229 | 1244 | 1261 | 1276 | 1364 | 1489 | 1623 | 1782 | 1963 | 1978 | 2066 |
|      | 2083 | 2193 | 2208 | 2296 | 2313 | 2844 | 2873 | 2888 | 2915 | 3040 | 3069 | 3084 | 3111 |      |      |
| DEC  | 1063 | 3225 | 3352 | 3621 |      |      |      |      |      |      |      |      |      |      |      |
| DECB | 3429 | 3488 | 3499 |      |      |      |      |      |      |      |      |      |      |      |      |
| EMT  | 302  |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| HALT | 632  | 1037 | 1291 | 2655 | 2704 | 3330 | 3414 |      |      |      |      |      |      |      |      |
| INC  | 1013 | 1066 | 3223 | 3287 | 3320 | 3494 | 3502 | 3546 |      |      |      |      |      |      |      |
| INCB | 3292 | 3314 |      |      |      |      |      |      |      |      |      |      |      |      |      |
| IOT  | 303  |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| JMP  | 636  | 639  | 642  | 1035 | 1058 | 1068 | 1096 | 1125 | 1302 | 1326 | 1394 | 1429 | 1451 | 1519 | 1554 |
|      | 1579 | 1617 | 1640 | 1677 | 1715 | 1738 | 1776 | 1799 | 1836 | 1874 | 1912 | 2022 | 2105 | 2142 | 2252 |
|      | 2335 | 2366 | 2424 | 2459 | 2501 | 2559 | 2594 | 2630 | 2676 | 2725 | 2761 | 2798 | 2926 | 2994 | 3122 |
|      | 3146 | 3220 | 3246 |      |      |      |      |      |      |      |      |      |      |      |      |
| JSR  | 1034 | 1047 | 1642 | 1801 | 3242 | 3326 | 3424 | 3431 |      |      |      |      |      |      |      |
| MOV  | 983  | 984  | 988  | 989  | 990  | 991  | 992  | 993  | 994  | 995  | 999  | 1000 | 1003 | 1004 | 1007 |
|      | 1012 | 1015 | 1019 | 1022 | 1025 | 1028 | 1045 | 1046 | 1052 | 1053 | 1056 | 1065 | 1067 | 1074 | 1075 |
|      | 1076 | 1077 | 1079 | 1086 | 1087 | 1088 | 1093 | 1105 | 1126 | 1128 | 1129 | 1130 | 1141 | 1144 | 1145 |
|      | 1148 | 1152 | 1160 | 1162 | 1165 | 1166 | 1169 | 1173 | 1181 | 1187 | 1189 | 1190 | 1192 | 1199 | 1201 |
|      | 1202 | 1204 | 1215 | 1217 | 1218 | 1220 | 1221 | 1225 | 1232 | 1233 | 1235 | 1236 | 1240 | 1247 | 1249 |
|      | 1250 | 1252 | 1253 | 1257 | 1264 | 1265 | 1267 | 1268 | 1272 | 1283 | 1285 | 1327 | 1329 | 1330 | 1331 |
|      | 1339 | 1344 | 1345 | 1346 | 1352 | 1353 | 1355 | 1356 | 1360 | 1372 | 1373 | 1378 | 1380 | 1382 | 1383 |
|      | 1386 | 1387 | 1395 | 1396 | 1400 | 1401 | 1405 | 1407 | 1408 | 1413 | 1414 | 1452 | 1454 | 1455 | 1456 |
|      | 1464 | 1469 | 1470 | 1471 | 1477 | 1478 | 1480 | 1481 | 1485 | 1497 | 1498 | 1503 | 1505 | 1507 | 1508 |
|      | 1511 | 1512 | 1520 | 1521 | 1525 | 1526 | 1530 | 1532 | 1533 | 1538 | 1539 | 1580 | 1582 | 1583 | 1584 |
|      | 1598 | 1605 | 1608 | 1609 | 1610 | 1611 | 1619 | 1620 | 1621 | 1622 | 1624 | 1634 | 1641 | 1644 | 1661 |
|      | 1663 | 1665 | 1666 | 1669 | 1670 | 1678 | 1679 | 1683 | 1684 | 1688 | 1689 | 1690 | 1692 | 1693 | 1698 |
|      | 1699 | 1739 | 1741 | 1742 | 1743 | 1757 | 1764 | 1767 | 1768 | 1769 | 1770 | 1778 | 1779 | 1780 | 1781 |
|      | 1783 | 1793 | 1800 | 1803 | 1820 | 1822 | 1824 | 1825 | 1828 | 1829 | 1837 | 1838 | 1842 | 1843 | 1847 |
|      | 1848 | 1849 | 1851 | 1852 | 1857 | 1858 | 1913 | 1915 | 1916 | 1917 | 1929 | 1930 | 1933 | 1934 | 1936 |
|      | 1937 | 1942 | 1943 | 1951 | 1952 | 1954 | 1955 | 1959 | 1966 | 1967 | 1969 | 1970 | 1974 | 1981 | 1991 |
|      | 1993 | 2006 | 2008 | 2010 | 2011 | 2014 | 2015 | 2023 | 2024 | 2028 | 2029 | 2033 | 2035 | 2036 | 2041 |
|      | 2042 | 2052 | 2054 | 2055 | 2058 | 2062 | 2069 | 2071 | 2072 | 2074 | 2075 | 2079 | 2089 | 2090 | 2091 |
|      | 2093 | 2094 | 2095 | 2143 | 2145 | 2146 | 2147 | 2159 | 2160 | 2163 | 2164 | 2166 | 2167 | 2172 | 2173 |
|      | 2181 | 2182 | 2184 | 2185 | 2189 | 2196 | 2197 | 2199 | 2200 | 2204 | 2211 | 2221 | 2223 | 2236 | 2238 |
|      | 2240 | 2241 | 2244 | 2245 | 2253 | 2254 | 2258 | 2259 | 2263 | 2265 | 2266 | 2271 | 2272 | 2282 | 2284 |
|      | 2285 | 2288 | 2292 | 2299 | 2301 | 2302 | 2304 | 2305 | 2309 | 2319 | 2320 | 2321 | 2323 | 2324 | 2325 |
|      | 2367 | 2369 | 2370 | 2371 | 2383 | 2384 | 2387 | 2388 | 2390 | 2391 | 2392 | 2395 | 2401 | 2402 | 2403 |
|      | 2408 | 2410 | 2412 | 2413 | 2416 | 2417 | 2425 | 2426 | 2430 | 2431 | 2435 | 2437 | 2438 | 2443 | 2444 |
|      | 2467 | 2468 | 2469 | 2470 | 2471 | 2502 | 2504 | 2505 | 2506 | 2518 | 2519 | 2522 | 2523 | 2525 | 2526 |
|      | 2527 | 2530 | 2536 | 2537 | 2538 | 2543 | 2545 | 2547 | 2548 | 2551 | 2552 | 2560 | 2561 | 2565 | 2566 |
|      | 2570 | 2572 | 2573 | 2578 | 2579 | 2600 | 2601 | 2602 | 2603 | 2604 | 2631 | 2633 | 2634 | 2635 | 2647 |
|      | 2648 | 2651 | 2652 | 2660 | 2662 | 2664 | 2665 | 2668 | 2669 | 2677 | 2678 | 2682 | 2683 | 2687 | 2688 |
|      | 2689 | 2691 | 2692 | 2696 | 2697 | 2709 | 2711 | 2713 | 2714 | 2717 | 2718 | 2726 | 2727 | 2731 | 2732 |
|      | 2736 | 2737 | 2738 | 2740 | 2741 | 2745 | 2746 | 2799 | 2801 | 2802 | 2803 | 2811 | 2832 | 2833 | 2835 |
|      | 2836 | 2840 | 2850 | 2851 | 2852 | 2859 | 2861 | 2862 | 2865 | 2869 | 2876 | 2877 | 2880 | 2884 | 2897 |
|      | 2898 | 2899 | 2901 | 2903 | 2904 | 2907 | 2911 | 2935 | 2944 | 2945 | 2946 | 2948 | 2949 | 2950 | 2951 |
|      | 2995 | 2997 | 2998 | 2999 | 3007 | 3028 | 3029 | 3031 | 3032 | 3036 | 3046 | 3047 | 3048 | 3055 | 3057 |
|      | 3058 | 3061 | 3065 | 3072 | 3073 | 3076 | 3080 | 3093 | 3094 | 3095 | 3097 | 3099 | 3100 | 3103 | 3107 |
|      | 3129 | 3138 | 3139 | 3140 | 3142 | 3143 | 3144 | 3145 | 3157 | 3160 | 3161 | 3162 | 3163 | 3164 | 3167 |
|      | 3169 | 3170 | 3171 | 3172 | 3176 | 3192 | 3193 | 3194 | 3228 | 3232 | 3235 | 3270 | 3271 | 3273 | 3276 |

|        |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|        | 3290 | 3291 | 3293 | 3294 | 3295 | 3316 | 3321 | 3343 | 3348 | 3357 | 3362 | 3367 | 3368 | 3370 | 3373 |
|        | 3377 | 3384 | 3385 | 3416 | 3417 | 3421 | 3427 | 3463 | 3471 | 3472 | 3473 | 3479 | 3486 | 3504 | 3505 |
|        | 3506 | 3507 | 3508 | 3529 | 3530 | 3531 | 3532 | 3533 | 3534 | 3535 | 3540 | 3543 | 3563 | 3569 | 3570 |
|        | 3571 | 3572 | 3573 | 3575 | 3576 | 3593 | 3594 | 3607 | 3609 | 3620 | 3651 | 3652 | 3653 | 3654 | 3675 |
|        | 3676 | 3677 | 3678 | 3679 | 3681 | 3682 | 3701 | 3702 | 3703 | 3704 | 3705 | 3733 | 3734 | 3735 | 3736 |
|        | 3737 | 3738 | 3739 | 3740 | 3741 | 3742 | 3749 | 3750 | 3751 | 3752 | 3753 | 3754 | 3755 | 3756 | 3757 |
|        | 3758 | 3769 | 3770 | 3773 |      |      |      |      |      |      |      |      |      |      |      |
| MOV8   | 998  | 1029 | 1051 | 1127 | 1140 | 1161 | 1186 | 1198 | 1214 | 1246 | 1282 | 1284 | 1328 | 1338 | 1371 |
|        | 1381 | 1385 | 1397 | 1402 | 1453 | 1463 | 1496 | 1506 | 1510 | 1522 | 1527 | 1581 | 1604 | 1607 | 1618 |
|        | 1633 | 1664 | 1668 | 1680 | 1685 | 1740 | 1763 | 1766 | 1777 | 1792 | 1823 | 1827 | 1839 | 1844 | 1914 |
|        | 1927 | 1931 | 1935 | 1992 | 2009 | 2013 | 2025 | 2030 | 2051 | 2068 | 2092 | 2144 | 2157 | 2161 | 2165 |
|        | 2222 | 2239 | 2243 | 2255 | 2260 | 2281 | 2298 | 2322 | 2368 | 2381 | 2385 | 2389 | 2411 | 2415 | 2427 |
|        | 2432 | 2466 | 2503 | 2516 | 2520 | 2524 | 2546 | 2550 | 2562 | 2567 | 2599 | 2632 | 2645 | 2649 | 2663 |
|        | 2667 | 2679 | 2684 | 2712 | 2716 | 2728 | 2733 | 2800 | 2810 | 2858 | 2896 | 2900 | 2934 | 2947 | 2996 |
|        | 3006 | 3054 | 3092 | 3096 | 3128 | 3141 | 3313 | 3323 | 3418 | 3435 | 3464 | 3465 | 3468 | 3469 | 3470 |
|        | 3474 | 3477 | 3478 | 3497 | 3538 | 3541 | 3555 | 3558 | 3567 | 3597 | 3613 | 3618 | 3624 | 3629 | 3644 |
|        | 3685 | 3772 |      |      |      |      |      |      |      |      |      |      |      |      |      |
| NEG    | 3475 | 3537 |      |      |      |      |      |      |      |      |      |      |      |      |      |
| NOP    | 1003 | 1004 | 1048 | 1157 | 1178 | 1197 | 1209 | 1230 | 1245 | 1262 | 1277 | 1365 | 1419 | 1490 | 1544 |
|        | 1704 | 1863 | 1964 | 1979 | 2047 | 2067 | 2084 | 2194 | 2209 | 2277 | 2297 | 2314 | 2449 | 2584 | 2701 |
|        | 2750 | 2845 | 2874 | 2889 | 2916 | 3041 | 3070 | 3085 | 3112 | 3243 | 3244 | 3245 |      |      |      |
| RESET  | 1001 | 1042 | 1073 | 3241 |      |      |      |      |      |      |      |      |      |      |      |
| ROL    | 3265 | 3481 | 3483 | 3484 | 3485 | 3487 | 3692 | 3694 | 3696 |      |      |      |      |      |      |
| RTI    | 3187 | 3296 | 3332 | 3423 | 3509 | 3577 | 3599 | 3655 | 3706 | 3743 | 3759 |      |      |      |      |
| RTS    | 3177 | 3198 | 3387 | 3436 | 3774 |      |      |      |      |      |      |      |      |      |      |
| SUB    | 3184 | 3322 | 3544 |      |      |      |      |      |      |      |      |      |      |      |      |
| TRAP   | 3776 | 3786 | 3787 | 3788 | 3789 | 3790 | 3791 | 3792 | 3793 | 3794 |      |      |      |      |      |
| TST    | 1084 | 1090 | 1122 | 1142 | 1158 | 1163 | 1179 | 1323 | 1391 | 1398 | 1403 | 1448 | 1516 | 1523 | 1528 |
|        | 1576 | 1635 | 1637 | 1674 | 1681 | 1686 | 1735 | 1794 | 1796 | 1833 | 1840 | 1845 | 1909 | 1944 | 1994 |
|        | 1396 | 2019 | 2026 | 2031 | 2139 | 2174 | 2224 | 2226 | 2249 | 2256 | 2261 | 2363 | 2421 | 2428 | 2433 |
|        | 2460 | 2498 | 2556 | 2563 | 2568 | 2627 | 2673 | 2680 | 2685 | 2722 | 2729 | 2734 | 2795 | 2927 | 2991 |
|        | 3159 | 3168 | 3182 | 3191 | 3218 | 3272 | 3285 | 3328 | 3379 | 3420 | 3492 | 3549 | 3559 | 3616 | 3627 |
|        | 3650 | 3700 | 3707 | 3771 |      |      |      |      |      |      |      |      |      |      |      |
| TSTB   | 1296 | 1423 | 1548 | 1709 | 1868 | 2099 | 2329 | 2453 | 2588 | 2755 | 2920 | 3116 | 3279 | 3371 | 3412 |
|        | 3433 | 3551 | 3565 | 3595 |      |      |      |      |      |      |      |      |      |      |      |
| .ASCII | 694  | 695  | 3802 | 3868 | 4048 | 4244 | 4252 | 4270 | 4283 | 4298 | 4309 | 4321 | 4336 |      |      |
| .ASCIZ | 693  | 696  | 3247 | 3388 | 3658 | 3806 | 3814 | 3819 | 3823 | 3827 | 3831 | 3838 | 3841 | 3845 | 3853 |
|        | 3858 | 3873 | 3883 | 3893 | 3902 | 3911 | 3918 | 3922 | 3940 | 3948 | 3952 | 3961 | 3966 | 3976 | 3984 |
|        | 3992 | 4002 | 4010 | 4018 | 4028 | 4038 | 4057 | 4064 | 4074 | 4084 | 4092 | 4104 | 4115 | 4124 | 4130 |
|        | 4138 | 4144 | 4154 | 4162 | 4172 | 4182 | 4192 | 4201 | 4211 | 4218 | 4226 | 4233 | 4237 | 4248 | 4258 |
|        | 4267 | 4277 | 4289 | 4304 | 4315 | 4327 | 4342 |      |      |      |      |      |      |      |      |
| .BLKB  | 3659 |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| .BLKW  | 3582 |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| .BYTE  | 661  | 662  | 667  | 668  | 679  | 680  | 681  | 682  | 3250 | 3510 | 3511 | 3512 | 3513 | 3656 | 3657 |
|        | 4380 | 4382 | 4384 | 4386 | 4387 | 4389 | 4391 | 4392 | 4395 | 4424 | 4425 | 4426 | 4427 | 4428 | 4429 |
|        | 4430 | 4431 |      |      |      |      |      |      |      |      |      |      |      |      |      |
| .ENABL | 1    | 272  |      |      |      |      |      |      |      |      |      |      |      |      |      |
| .END   | 4435 |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| .ENDC  | 278  | 292  | 294  | 295  | 302  | 388  | 402  | 637  | 648  | 656  | 658  | 683  | 686  | 691  | 692  |
|        | 693  | 694  | 730  | 731  | 982  | 988  | 989  | 991  | 993  | 995  | 996  | 997  | 999  | 1001 | 1108 |
|        | 1109 | 1119 | 1120 | 1130 | 1131 | 1144 | 1148 | 1151 | 1155 | 1156 | 1158 | 1165 | 1169 | 1172 | 1176 |
|        | 1177 | 1179 | 1189 | 1193 | 1194 | 1195 | 1196 | 1198 | 1201 | 1205 | 1206 | 1207 | 1208 | 1210 | 1217 |
|        | 1221 | 1224 | 1228 | 1229 | 1231 | 1232 | 1236 | 1239 | 1243 | 1244 | 1246 | 1249 | 1253 | 1256 | 1260 |
|        | 1261 | 1263 | 1264 | 1268 | 1271 | 1275 | 1276 | 1278 | 1298 | 1300 | 1307 | 1308 | 1320 | 1321 | 1331 |
|        | 1332 | 1352 | 1356 | 1359 | 1363 | 1364 | 1366 | 1374 | 1381 | 1406 | 1407 | 1410 | 1416 | 1420 | 1425 |
|        | 1427 | 1432 | 1433 | 1445 | 1446 | 1456 | 1457 | 1477 | 1481 | 1484 | 1488 | 1489 | 1491 | 1499 | 1506 |



|        |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|        | 1531 | 1532 | 1535 | 1541 | 1545 | 1550 | 1552 | 1560 | 1561 | 1573 | 1574 | 1584 | 1585 | 1599 | 1607 |
|        | 1616 | 1618 | 1620 | 1622 | 1629 | 1664 | 1691 | 1692 | 1695 | 1701 | 1711 | 1713 | 1719 | 1720 | 1732 |
|        | 1733 | 1743 | 1744 | 1758 | 1766 | 1775 | 1777 | 1779 | 1781 | 1788 | 1823 | 1850 | 1851 | 1854 | 1860 |
|        | 1870 | 1872 | 1878 | 1879 | 1906 | 1907 | 1917 | 1918 | 1935 | 1951 | 1955 | 1958 | 1962 | 1963 | 1965 |
|        | 1966 | 1970 | 1973 | 1977 | 1978 | 1980 | 2009 | 2034 | 2035 | 2038 | 2044 | 2054 | 2058 | 2061 | 2065 |
|        | 2066 | 2068 | 2071 | 2075 | 2078 | 2082 | 2083 | 2085 | 2101 | 2103 | 2108 | 2109 | 2136 | 2137 | 2147 |
|        | 2148 | 2165 | 2181 | 2185 | 2188 | 2192 | 2193 | 2195 | 2196 | 2200 | 2203 | 2207 | 2208 | 2210 | 2239 |
|        | 2264 | 2265 | 2268 | 2274 | 2284 | 2288 | 2291 | 2295 | 2296 | 2298 | 2301 | 2305 | 2308 | 2312 | 2313 |
|        | 2315 | 2331 | 2333 | 2340 | 2341 | 2360 | 2361 | 2371 | 2372 | 2389 | 2411 | 2436 | 2437 | 2439 | 2440 |
|        | 2445 | 2446 | 2463 | 2473 | 2475 | 2476 | 2495 | 2496 | 2506 | 2507 | 2524 | 2546 | 2571 | 2572 | 2574 |
|        | 2575 | 2580 | 2581 | 2596 | 2606 | 2609 | 2610 | 2624 | 2625 | 2635 | 2636 | 2653 | 2663 | 2690 | 2691 |
|        | 2693 | 2698 | 2712 | 2739 | 2740 | 2742 | 2747 | 2757 | 2759 | 2766 | 2767 | 2792 | 2793 | 2803 | 2804 |
|        | 2832 | 2836 | 2839 | 2843 | 2844 | 2846 | 2861 | 2865 | 2868 | 2872 | 2873 | 2875 | 2876 | 2880 | 2883 |
|        | 2887 | 2888 | 2890 | 2903 | 2907 | 2910 | 2914 | 2915 | 2917 | 2930 | 2954 | 2956 | 2957 | 2988 | 2989 |
|        | 2999 | 3000 | 3028 | 3032 | 3035 | 3039 | 3040 | 3042 | 3057 | 3061 | 3064 | 3068 | 3069 | 3071 | 3072 |
|        | 3076 | 3079 | 3083 | 3084 | 3086 | 3099 | 3103 | 3106 | 3110 | 3111 | 3113 | 3124 | 3147 | 3207 | 3210 |
|        | 3212 | 3213 | 3215 | 3221 | 3227 | 3230 | 3231 | 3235 | 3247 | 3250 | 3251 | 3252 | 3258 | 3261 | 3265 |
|        | 3267 | 3278 | 3279 | 3281 | 3283 | 3287 | 3292 | 3294 | 3297 | 3298 | 3299 | 3305 | 3314 | 3321 | 3327 |
|        | 3328 | 3332 | 3333 | 3334 | 3352 | 3390 | 3391 | 3438 | 3516 | 3584 | 3587 | 3600 | 3601 | 3609 | 3611 |
|        | 3642 | 3659 | 3660 | 3661 | 3670 | 3714 | 3715 | 3761 | 3770 | 3773 | 3775 | 3785 | 3786 | 3787 | 3788 |
|        | 3789 | 3790 | 3791 | 3792 | 3793 | 3794 | 3795 |      |      |      |      |      |      |      |      |
| .EQUIV | 302  | 303  | 305  | 320  | 321  | 350  | 351  | 352  | 353  | 354  | 355  | 356  | 357  | 358  | 359  |
|        | 378  | 379  | 380  | 381  | 382  | 383  | 384  | 385  | 386  | 387  |      |      |      |      |      |
| .EVEN  | 3389 | 4349 | 4398 |      |      |      |      |      |      |      |      |      |      |      |      |
| .IF    | 274  | 292  | 293  | 294  | 295  | 300  | 360  | 388  | 632  | 647  | 653  | 657  | 683  | 686  | 691  |
|        | 692  | 693  | 697  | 730  | 982  | 983  | 988  | 989  | 991  | 993  | 995  | 996  | 997  | 999  | 1107 |
|        | 1109 | 1119 | 1130 | 1131 | 1144 | 1147 | 1148 | 1152 | 1155 | 1157 | 1165 | 1168 | 1169 | 1173 | 1176 |
|        | 1178 | 1189 | 1192 | 1193 | 1195 | 1197 | 1201 | 1204 | 1205 | 1207 | 1209 | 1217 | 1220 | 1221 | 1225 |
|        | 1228 | 1230 | 1232 | 1235 | 1236 | 1240 | 1243 | 1245 | 1249 | 1252 | 1253 | 1257 | 1260 | 1262 | 1264 |
|        | 1267 | 1268 | 1272 | 1275 | 1277 | 1297 | 1299 | 1306 | 1308 | 1320 | 1331 | 1332 | 1352 | 1355 | 1356 |
|        | 1360 | 1363 | 1365 | 1374 | 1380 | 1406 | 1409 | 1410 | 1415 | 1416 | 1420 | 1424 | 1426 | 1431 | 1433 |
|        | 1445 | 1456 | 1457 | 1477 | 1480 | 1481 | 1485 | 1488 | 1490 | 1499 | 1505 | 1531 | 1534 | 1535 | 1540 |
|        | 1541 | 1545 | 1549 | 1551 | 1559 | 1561 | 1573 | 1584 | 1585 | 1598 | 1606 | 1607 | 1615 | 1617 | 1621 |
|        | 1663 | 1689 | 1691 | 1694 | 1695 | 1700 | 1701 | 1710 | 1712 | 1718 | 1720 | 1732 | 1743 | 1744 | 1757 |
|        | 1765 | 1766 | 1774 | 1776 | 1780 | 1822 | 1848 | 1850 | 1853 | 1854 | 1859 | 1860 | 1869 | 1871 | 1877 |
|        | 1879 | 1906 | 1917 | 1918 | 1925 | 1951 | 1954 | 1955 | 1959 | 1962 | 1964 | 1966 | 1969 | 1970 | 1974 |
|        | 1977 | 1979 | 2008 | 2034 | 2035 | 2037 | 2038 | 2043 | 2044 | 2054 | 2057 | 2058 | 2062 | 2065 | 2067 |
|        | 2071 | 2074 | 2075 | 2079 | 2082 | 2084 | 2100 | 2102 | 2107 | 2109 | 2136 | 2147 | 2148 | 2155 | 2181 |
|        | 2184 | 2185 | 2189 | 2192 | 2194 | 2196 | 2199 | 2200 | 2204 | 2207 | 2209 | 2238 | 2264 | 2265 | 2267 |
|        | 2268 | 2273 | 2274 | 2284 | 2287 | 2288 | 2292 | 2295 | 2297 | 2301 | 2304 | 2305 | 2309 | 2312 | 2314 |
|        | 2330 | 2332 | 2339 | 2341 | 2360 | 2371 | 2372 | 2379 | 2410 | 2436 | 2437 | 2439 | 2445 | 2460 | 2472 |
|        | 2474 | 2476 | 2495 | 2506 | 2507 | 2514 | 2545 | 2571 | 2572 | 2574 | 2580 | 2595 | 2605 | 2608 | 2610 |
|        | 2624 | 2635 | 2636 | 2643 | 2662 | 2688 | 2690 | 2691 | 2693 | 2698 | 2711 | 2737 | 2739 | 2740 | 2742 |
|        | 2747 | 2756 | 2758 | 2765 | 2767 | 2792 | 2803 | 2804 | 2832 | 2835 | 2836 | 2840 | 2843 | 2845 | 2861 |
|        | 2864 | 2865 | 2869 | 2872 | 2874 | 2876 | 2879 | 2880 | 2884 | 2887 | 2889 | 2899 | 2903 | 2906 | 2911 |
|        | 2914 | 2916 | 2927 | 2952 | 2953 | 2955 | 2957 | 2982 | 2988 | 2999 | 3000 | 3028 | 3031 | 3032 | 3036 |
|        | 3039 | 3041 | 3057 | 3060 | 3061 | 3065 | 3068 | 3070 | 3072 | 3075 | 3076 | 3080 | 3083 | 3085 | 3099 |
|        | 3102 | 3103 | 3107 | 3110 | 3112 | 3123 | 3146 | 3206 | 3210 | 3211 | 3212 | 3213 | 3214 | 3215 | 3217 |
|        | 3226 | 3229 | 3231 | 3235 | 3246 | 3247 | 3251 | 3257 | 3261 | 3265 | 3277 | 3279 | 3280 | 3281 | 3283 |
|        | 3285 | 3294 | 3296 | 3297 | 3298 | 3304 | 3313 | 3317 | 3324 | 3326 | 3327 | 3328 | 3331 | 3332 | 3333 |
|        | 3351 | 3367 | 3390 | 3437 | 3515 | 3583 | 3587 | 3600 | 3608 | 3610 | 3615 | 3658 | 3659 | 3660 | 3666 |
|        | 3682 | 3714 | 3760 | 3769 | 3773 | 3775 | 3776 | 3786 | 3787 | 3788 | 3789 | 3790 | 3791 | 3792 | 3793 |
|        | 3794 |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| .IFF   | 292  | 294  | 295  | 300  | 648  | 657  | 683  | 731  | 988  | 1107 | 1108 | 1109 | 1120 | 1131 | 1148 |
|        | 1151 | 1156 | 1157 | 1169 | 1172 | 1177 | 1178 | 1192 | 1193 | 1196 | 1197 | 1204 | 1205 | 1208 | 1209 |
|        | 1220 | 1224 | 1229 | 1230 | 1235 | 1239 | 1244 | 1245 | 1252 | 1256 | 1261 | 1262 | 1267 | 1271 | 1276 |



MD-11-DERPQ-A, RPO4 DUAL CONTROLLER LOGIC TEST - PART 2 MACY11 27(732) 05-OCT-76 14:30 PAGE 108  
DERPQA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

\* DERPQA.SEQ/SOL/CRF/PAGNUM/NL:TOC/NL:MC=DERPGA.SML, DERPQA.P11  
RUN-TIME: 41 58 8 SECONDS  
RUN-TIME RATIO: 352/108=3.2  
CORE USED: 30K (59 PAGES)

