

DR11C

DEVICE REGISTER TEST
MD-11-DZDRC-F

EP-DZDRC-F-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A

The microfiche card displays a grid of frames. The left column contains frames with text, and the right column contains frames with data tables. The rest of the card is dark and mostly blank.

Frame	Content
1	Text
2	Table
3	Table
4	Table
5	Table
6	Table
7	Table
8	Table
9	Table
10	Table
11	Table
12	Table
13	Table
14	Table
15	Table
16	Table
17	Table
18	Table
19	Table
20	Table

MAIN. MAC/11 27(732) 16-JAN-76 06:56
DZORCF.P11 TABLE OF CONTENTS

320	BASIC DEFINITIONS
100	TYPE ROUTINE
100	CODER
1190	TRAP TABLE

10:08:56 27(732) 16-JAN-76 08:56 PAGE 2

1. ABSTRACT

THIS IS A LOGIC TEST OF THE DR11C. FOR THIS TEST TO OPERATE A SPECIAL MAINTENANCE CABLE MUST BE CONNECTED (BCOBR). THIS TEST WILL CHECK UP TO 32 SEQUENTIAL DR11C'S.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 STANDARD COMPUTER

DR11C

BCOBR FOR EACH DR11C

2.2 STORAGE

2.2.1 PROGRAM STORAGE - THE ROUTINE USES MEMORY FROM 0000 TO 5200.

3. LOADING PROCEDURE

3.1 METHOD

ABSOLUTE LOADER

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTING

FOR SWITCHLESS PROCESSORS, THE PROGRAM WILL USE THE CONTENTS OF LOCATION 176 AS THE VALUE OF THE SWITCHES IF NO HARDWARE SWITCH REGISTER IS FOUND. THE OPERATOR IS RESPONSIBLE FOR LOADING LOCATION 176 WITH THE DESIRED SWITCH VALUE BEFORE STARTING THE PROGRAM. LOCATION 174 WILL BE USED AS THE SOFTWARE DISPLAY REGISTER.

STARTING AT SA 200 ALL SWITCHES SHOULD BE DOWN OR ZERO. (IF NOT ZERO, BIT 0 TO 8 WILL BE STARTING VECTOR.)

4.2 STARTING ADDRESS OR ADDRESSES

- (A) 200 = START OF TEST--FOR NORMAL TESTING
- (B) 204 = SPECIAL ENTRANCE --FOR TESTING UNIQUE DR11C
- (C) 210 = RESTART--FOR STARTING AFTER SHUT DOWN

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY.
SET SWITCH REGISTER TO STARTING ADDRESS.

10
101
102
103
104
105
106
107
108
109
110
111

LOAD ADDRESS.
PRESS START.
THE PROGRAM WILL STAY IN SECTION AND LOOP.

4.3.1 FOR SPECIAL ENTRANCE - SA204

1ST HALT SET SWITCH REGISTER EQUAL TO CSR ADDRESS OF DR11C
PRESS CONTINUE
2ND HALT SET SWITCH REGISTER EQUAL TO VECTOR ADDRESS OF DR11C
PRESS CONTINUE
RAISE SWITCH 10 TO "1" TO INHIBIT SEQUENCING TO NEXT DR11C

112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 AT SA 200 ... THE INSTRUCTION AND LOGIC TEST.
WITH ALL SWITCHES DOWN THE PROGRAM WILL PRINT
OUT ON ERRORS AND CONTINUE IN TEST. (* WILL
BE PRINTED AT COMPLETION OF TESTING EACH DR11C)

5.1.2 SWITCH SETTINGS ARE

- SW15 = 1 OR UP ... HALT ON ERROR
- SW14 = 1 OR UP ... SCOPE LOOP
- SW13 = 1 OR UP ... INHIBIT PRINTOUT
- SW12 = 1 OR UP ... NOT USED
- SW11 = 1 OR UP ... INHIBIT ITERATION LOOP
- SW10 = 1 OR UP ... DO NOT ADVANCE TO NEXT DR11C
- SW09 = 1 OR UP ... INHIBIT PRINTOUT OF DEVICE TESTED.

5.1.3

5.2. SUBROUTINE ABSTRACTS

5.2.1 BEGIN SA 200

5.2.2 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST
IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING
ADDRESS OF EACH SUB-TEST AS IT IS BEING ENTERED.
IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE
START OF THE SUBTEST THAT THE SCOPE LOOP IS RE-
QUESTED FOR.

5.2.3 HALT

IS A ROUTINE THAT PRINTS-OUT AN ADDRESS THAT TAGS
THE FAILING SUBTEST, AND THE INCORRECT DATA AT
THE TIME OF THE FAILURE. SEE 6.1

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

(5. OPERATING PROCEDURE CONT'D)

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 LOADING AND STARTING AT 200 WITH ALL SWITCHES DOWN IS THE INSTRUCTION AND LOGIC TEST. IF AN ERROR IS DETECTED HERE, THERE WILL BE A PRINTOUT. WHEN AN ERROR IS DETECTED AND IT IS NECESSARY TO SCOPE ON IT, PLACE SW15 UP TO HALT ON ERROR, THEN SW14 UP TO LOOP ON ERROR, THEN SW13 UP TO DELETE PRINTOUTS.

6. ERRORS

6.1 ERROR PRINTOUT

ARE IN A FOUR WORD FORMAT. THE 1ST IS THE PC+2 OF THE DETECTED ERROR. THE 2ND IS THE PROCESSOR STATUS REGISTER. THE 3RD IS DEVICE ADDRESS, THE 4TH IS VECTOR ADDRESS.

6.2 ERROR RECOVERY

DEPRESS CONTINUE TO RESTART SECTION

179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227

- 7. RESTRICTIONS
- 7.1 STARTING RESTRICTION
- NONE
- 7.2 OPERATIONAL RESTRICTION

THE DR11C MUST HAVE THE BC08R CABLE TO RUN THIS TEST.

NOTE THAT THE DR11C HAS FLOATING VECTORS:

THE BELOW IS THE ASSIGNMENT OF FLOATING VECTORS, THE ASSIGNED SEQUENCES ARE:

1. STARTING AT 300 AND WORKING UPWARD ALL DC11'S WILL BE ASSIGNED.
2. THEN ANY EXTRA KL11 CALLED FOR (VT05, VT06, LC11)
3. THEN ANY DP11 CALLED FOR.
4. THEN ANY DM11 CALLED FOR.
5. THEN ANY DN11 CALLED FOR.
6. THEN ANY DM11BB CALLED FOR.
7. THEN ANY DR11A CALLED FOR.
8. THEN ANY DR11C CALLED FOR.

THE DR11. AND DR11C DEVICE ADDRESSES WILL BE ASSIGNED IN THE USER AREA OF 767776 TO 764000. THE ASSIGNMENT OF ADDRESSES WILL START AT THE HIGH ADDRESS LIMIT AND PROCEED DOWNWARD. USERS AND SPECIAL SYSTEMS SHOULD START THEIR ASSIGNMENTS OF SPECIAL DEVICES AT THE LOW ADDRESS LIMIT AND WORK UP. AFTER ASSIGNING ALL DR11A'S, ASSIGN DR11C'S

767776 TO 767770	DR11C #0	;ASSUMING NO DR11A'S
767766 TO 767760	DR11C #1	
:	:	
:	:	
767706 TO 767700	DR11C #7	
:	:	
:	:	
767606 TO 767600	DR11C #15	

2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300

8. MISCELLANEOUS

WHERE THERE ARE MULTIPLE DR11C OR A SYSTEM AND IT IS DESIRED TO TEST ONLY ONE OF THEM. THIS MAY BE ACHIEVED BY USING THE SPECIAL STARTING ADDRESS AND PLACING SW10 ON A ONE (UP) TO INHIBIT SEQUENCING TO THE NEXT DR11C. SEE 4.3.1.

8.1 EXECUTION TIME

FOR EACH DR11C ABOUT 1 MINUTE

8.2 UNTESTED LOGIC

SIGNALS TO USER NOT TESTED:
"NEW DATA READY"
"DATA TRANSMITTED"
"INIT" TO THE USER

9. PROGRAM DESCRIPTION

THIS PROGRAM WHEN STARTED AT 200 CHECKS THE STANDARD DR11-C'S THE PROGRAM THEN PERFORMS AN INCREMENTAL LOGIC CHECK FOR THE SELECTED DR11C.

THE DATA REGISTER IS TESTED TO SEE IF "RESET" CLEARS IT, AND IF IT WILL HOLD ALL COMBINATIONS OF NUMBERS.

THE READ/WRITE BITS OF THE STATUS REGISTER ARE ALSO TESTED.

BOTH THE "A" AND "B" INTERRUPTS ARE TESTED TO SEE IF THEY INTERRUPT AT THE CORRECT BUS REQUEST LEVEL BR-5.

AT THE END OF THE TEST AN '*' IS TYPED AND ALSO THE ADDRESSES OF THE DR11-C CONTROL STATUS REGISTER AND IT'S SIDE INTERRUPT VECTOR IS TYPED (IF SELECTED VIA SWITCH 9.). THE PROGRAM THEN RETESTS THE UNIT (IF SELECTED VIA SWITCH 10) OR GOES TO THE NEXT DR11-C. IF ANOTHER DR11-C IS ON THE SYSTEM THEN THE PROGRAM RESTARTS TESTING THE NEW DR11-C.

AFTER ALL DR11-C'S HAVE BEEN TESTED THE PROGRAM WILL TYPE '/' AND RESTART TESTING WITH THE INITIAL DR11-C.

IF NO ERRORS OCCUR AND THREE DR11-C'S ARE AVAILABLE AND SWITCH 9 IS DOWN THE PROGRAM WILL TYPE.

160000 770 *
157770 1000 *
157760 1010 *

/
ETC.

IF SWITCH 9 IS UP THEN

*
*
*

284
15
285
286
287
288
289
290
291

IF A POWER FAIL OCCURS THE PROGRAM WILL RESTART AT "START".

10. LISTING

.ENDR

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
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347

177776
104070
167770
001200

000000
000001
000002
000003
000004
000005
000006
000007

001000
002000
004000
020000
040000

000000

000000

000020
000020 004364
000022 000340
000024 004624
000026 000340
000030 003654
000032 000340
000034 005012
000036 000340
000046
000046 003636
000174
000174 000000
000176 000000
000200
000200 000137 001246
000204 000137 001334
000210 000137 001444
001200

;GENERAL REGISTER LOGIC TEST

PSW=177776
HLT=104070
CSR=167770
STKPTR=1200

;REGISTER DEFINITIONS

R0=X0
R1=X1
R2=X2
R3=X3
R4=X4
R5=X5
SP=X6
PC=X7

;SWITCHES

SW9=1000
SW10=2000
SW11=4000
SW13=20000
SW14=40000

.MCALL .STYPE, .STRAP, .EQUAT
.EQUAT

.ENABLE ABS
.=0
.REPT 200
.+2
HALT
.ENOR
.=20
.SCOPE
340
PFAIL
340
.HLT
340
TYP
340
.=46
LOGIC
.=174

.ISPRG: 0
SWREG: 0

.=200
.MP
JMP
JMP
.=1200

@#START1
@#SPEC
@#START

:INITIAL START
:TO SELECT UNIQUE ADDRESS AND VECTOR
:RESTART

;THIS TABLE CONTAINS INITIAL REGISTER AND VECTOR ADDRESSES

```

348 001200 167770
349 001202 167772
350 001204 167774
351 001206 167773
352 001210 000300
353 001212 000302
354 001214 000304
355
356
357
358 001216 177570
359 001220 177570
360 001222 167770
361 001224 167772
362 001226 167774
363 001230 167773
364
365 001232 000300
366 001234 000302
367 001236 000304
368 001240 000000
369
370 001242 000000
371 001244 000240
372
373 001246 012706 001200
374 001252 016746 176530
375 001256 016746 176522
376 001262 012767 001276 176514
377 001270 005777 177722
378 001274 000407
379 001276 012767 000176 177712
380 001304 012767 000174 177706
381 001312 022626
382 001314 012667 176464
383 001320 012667 176462
384 001324 004767 000020
385 001330 000137 001444
386 001334 012706 001200
387 001340 004767 000004
388 001344 000167 003342
389 001350 013746 000004
390 001354 012737 001430 000004
391 001362 012737 000031 177060
392 001370 012637 000004
393 001374 012737 177777 001240
394 001402 012701 160000
395 001406 004737 004742
396 001412 012701 000770
397 001416 004737 004772
398 001422 104400
399 001424 005270
400 001426 000207
401 001430 022626
402 001432 012637 000004
403 001436 005337 001240
  
```

```

RCSR: CSR
      CSR+2
      CSR+4
      CSR+3
RCSR1: 300
       302
       304
  
```

; THIS TABLE CONTAINS REGISTER AND VECTOR ADDRESSES OF THE DR11-C UNDER TEST

```

SR: 177570
DISPLA: 177570
DRCSR: 167770
DRBUF: 167772
DRIBUF: 167774
DRBHIO: 167773
  
```

```

; ADDRESS OF DR11-C STATUS REGISTER
; ADDRESS OF DR OUTPUT BUFFER REG.
; ADDRESS OF DR INPUT BUFFER REG.
; HIGH BYTE OF OUTPUT BUFFER REG.
  
```

```

DRVECA: 300
DRLVL: 302
DRVECB: 304
XORFLG: 0
  
```

```

; INTERRUPT VECTOR OF UNIT UNDER TEST
; INTERRUPT VECTOR
  
```

```

COUNT: 0
PL: 240
  
```

```

; COUNT LOCATION
; PRIORITY LEVEL
  
```

```

START1: MOV #STKPTR,%6
        MOV 6,-(SP)
        MOV 4,-(SP)
        MOV #15,4
        TST 2$SR
        BR 2$
  
```

; SAVE CURRENT VECTOR

```

1$: MOV #SWREG,SR
    MOV #DISPAC,DISPLA
    CMP (SP)+,(SP)+
    MOV (SP)+,4
    MOV (SP)+,6
    JSR PC,FIRST
    JMP @START
  
```

```

; SET UP TIME OUT VECTOR
; TRY TO REFERENCE HARDWARE SWR
; BRANCH IF NO TIME OUT OCCURS
; POINT TO SOFTWARE SWITCH 1 REGISTER
; POINT TO DISPLAY REGISTER
; RESTORE STACK
; RESTORE TIME OUT VECTOR
  
```

```

SPEC: MOV #STKPTR,%6
      JSR PC,FIRST
      JMP SPEC0
  
```

```

FIRST: MOV @4,-(%6)
        XORR @4
        MOV @31,@177060
        MOV (%6)+,@4
        MOV @-1,@XORFLG
        MOV @160000,R1
        JSR PC,@SPEC1
        MOV @1770,R1
        JSR PC,@SPEC2
  
```

```

XORA: RTS PC
      CMP (%6)+,(%6)+
      MOV (%6)+,@4
      CLR @XORFLG
  
```

```

404 001442 000207          RTS      PC
405          ;INITIALIZE ADDRESS AND VECTORS
406 001444 012700 001200  START:  MOV     #RCSR, R0      ;GET ADDRESS OF FIRST POSSIBLE DR11-C'S
407 001450 012701 001222      MOV     #DRCSA, R1
408 001454 012021          MOV     (R0)+, (R1)+    ;LOAD INITIAL TEST ADDRESSES
409 001456 012021          MOV     (R0)+, (P1)+
410 001460 012021          MOV     (R0)+, (R1)+
411 001462 012021          MOV     (R0)+, (R1)+
412 001464 012021          MOV     (R0)+, (R1)+
413 001466 012021          MOV     (R0)+, (R1)+
414 001470 012021          MOV     (R0)+, (R1)+
415 001472 012706 001200  RSTART: MOV   #STKPTR, %6    ;SET UP STACK
416 001476 012767 001524 002766  MOV     #BEGIN, RETURN    ;SET SCOPE RETURN
417 001504 005037 004470      CLR     @#SCOPEF
418
419          ;DOES RESET CLEAR REGISTER?
420 001510 032777 001000 177500  BIT     #SW9, %SR
421 001516 001002          BNE     BEGIN
422 001520 004737 004510      JSR     PC, @#MOREID
423 001524 016705 177472  BEGIN:  MOV     DRCSA, R5      ;GET ADDRESS OF STATUS REGISTER
424 001530 012777 000240 176240  MOV     #240, %PSW     ;SET PRIORITY LEVEL 6
425 001536 012737 001574 000004  MOV     #15, @#4      ;SET TIME OUT TRAP VECTOR
426 001544 012767 000010 002714  MOV     #10, ICOUNT
427 001552 012777 177777 177444  MOV     #-1, @DROBUF  ;PRESET OUTPUT BUFFER
428 001560 000005          RESET   ;CLEAR DATA REGISTER
429 001562 017700 177436      MOV     @DROBUF, %0    ;GET RESULT OF RESET
430 001566 001403          BEQ    25
431 001570 104000          HLT
432 001572 000401          BR     25
433 001574 104000          HLT
434 001576 012706 001200 15:      MOV     #STKPTR, SP    ;ERROR! TIMED OUT WHEN REFERENCING DROBUF.
435 001602 012737 000006 000004 25:      MOV     #6, @#4      ;RESET STACK POINTER
436          ;RESTORE TIME OUT TRAP
437 001610 000004          SCOPE
438 001612 012767 004000 002646  MOV     #4000, ICOUNT
439 001620 012777 177777 177376  MOV     #-1, @DROBUF  ;ALL ONES TO REGISTER
440 001626 017700 177372  MOV     @DROBUF, %0
441 001632 022700 177777      CMP    #-1, %0
442 001636 001401          BEQ    .+4
443 001640 104000          HLT
444          ;REG WILL NOT HOLD ONES
445 001642 000004          SCOPE
446 001644 012767 000010 002614  MOV     #10, ICOUNT
447 001652 012777 177777 177344  MOV     #-1, @DROBUF
448 001660 000005          RESET   ;SET DATA TO ALL ONES
449 001662 005777 177340      TST    @DROBUF        ;SHOULD CLEAR REGISTER
450 001666 001401          BEQ    .+4
451 001670 104000          HLT
452          ;REG FAILED TO CLEAR
453 001672 000004          SCOPE
454 001674 012767 004000 002564  MOV     #4000, ICOUNT
455 001702 012777 052525 177314  MOV     #52525, @DROBUF
456 001710 017700 177310  MOV     @DROBUF, %0
457 001714 022700 052525      CMP    #52525, %0
458 001720 001401          BEQ    .+4
459 001722 104000          HLT
          ;DATA NOT=52525

```



```

516 002150 001402          BEQ      35
517 002152 000764          DR       15
518 002154 000000          .WORD   0
519 002156 000004          .SCOPE
520          ;CONTROL STATUS REGISTER (DRCSR) TESTS.
521 002160 005015          CLR      (RS)
522 002162 011500          MOV      (RS),R0
523 002164 001401          BEQ     .+4
524 002166 104000          HLT
525 002170 012715 000140          MOV     #140,R5          ; INTERRUPT ENABLE FOR A+B
526 002174 011500          MOV     R5,%0
527 002176 022700 000140          CMP     #140,%0          ; ENABLE BITS
528 002202 001401          BEQ     .+4
529 002204 104000          HLT
530
531 002206 000004          SCOPE
532 002210 012767 000010 002250          MOV     #10,ICOUNT
533 002216 012715 000140          MOV     #140,R5          ; SET INTERRUPT ENABLE FLOPS
534 002222 000005          RESET          ; CLEAR THOSE FLOPS
535 002224 011500          MOV     R5,%0
536 002226 001401          BEQ     .+4
537 002230 104000          HLT          ; RESET DID NOT CLEAR INTERRUPT ENABLE BITS
538
539 002232 000004          SCOPE
540 002234 052715 000001          BIS     #1,R5          ; SHOULD SET REQ A ALSO
541 002240 021527 000201          CMP     R5,#201
542 002244 001401          BEQ     .+4
543 002246 104000          HLT
544 002250 005015          CLR     R5
545
546 002252 000004          SCOPE
547 002254 052715 000002          BIS     #2,R5          ; SHOULD SET REQ B
548 002260 021527 100002          CMP     R5,#100002
549 002264 001401          BEQ     .+4
550 002266 104000          HLT
551 002270 005015          CLR     R5
552
553 002272 000004          SCOPE
554 002274 052737 000340 177776          BIS     #340,#PSW
555 002302 052715 177777          BIS     #-1,R5
556 002306 022715 100343          CMP     #100343,(R5)
557 002312 001401          BEQ     .+4
558 002314 104000          HLT
559 002316 042715 000003          BIC     #3,R5
560 002322 022715 000140          CMP     #140,R5
561 002326 001401          BEQ     .+4
562 002330 104000          HLT          ; WRONG BITS SET
563
564 002332 000004          SCOPE
565 002334 012737 000340 177776          MOV     #340,#PSW
566 002342 052715 000003          BIS     #3,R5
567 002346 000005          RESET
568 002350 005715          TST     R5
569 002352 001401          BEQ     .+4
570 002354 104000          HLT          ; RESET DID NOT CLEAR
571
  
```

```

572 002356 000004          SCOPE
573 002360 012767 004000 002100  MOV      #4000, ICOUNT
574 002366 005315          CLR      @RS
575 002370 005215          INC      @RS
576 002372 105715          TSTB    @RS
577 002374 100401          BMI     .+4
578 002376 104000          HLT
                                     ;BIT 0 DID NOT SET BIT 7

580 002400 000004          SCOPE
581 002402 012715 000002  MOV      #2, @RS
582 002406 005715          TST     @RS
583 002410 100401          BMI     .+4
584 002412 104000          HLT
                                     ;BIT 1 DID NOT SET BIT 15

585
586
587 002414 000004          ;TEST EXTERNAL TRANSFERS - CABLE MUST BE CONNECTED
588 002416 005077 176602  SCOPE
589 002422 017777 176600 176574  CLR      @ROBUF
590 002430 001401          MOV      @ROBUF, @ROBUF ;TEST TRANSFER OF ZERO
591 002432 104000          BEQ     .+4
                                     ;NOT EQUAL TO ZERO

592
593 002434 000004          SCOPE
594 002436 012777 177777 176560  MOV      #-1, @ROBUF
595 002444 017777 176556 176552  MOV      @ROBUF, @ROBUF ;MOV ALL ONES
596 002452 022777 177777 176544  CMP      #-1, @ROBUF
597 002460 001401          BEQ     .+4
598 002462 104000          HLT
                                     ;NOT ALL ONES

599
600 002464 000004          SCOPE
601 002466 005067 001774  CLR      ICOUNT
602 002472 005000          CLR      %0
603 002474 010077 176524          TST6:  MOV      %0, @ROBUF ;TEST ALL NUMBERS
604 002500 017777 176522 176516  MOV      @ROBUF, @ROBUF
605 002506 020077 176512          CMP      %0, @ROBUF
606 002512 001401          BEQ     .+4
607 002514 104000          HLT
                                     ;ERROR - CHECK %0 FOR GOOD
608 002516 005200          INC      %0
                                     ;@ROBUF FOR BAD
609 002520 001403          BEQ     TST9
610 002522 005077 176476  CLR      @ROBUF
611 002526 000702          BR      TST6
612 002530 000004          TST9:  SCOPE
613 002532 012737 000005 004466  MOV      #5, @ICOUNT
                                     ;TEST DATA FROM BLACK BOX (NOT CONNECTED)
614
615 002540 012777 177777 176456  MOV      #-1, @ROBUF
616 002546 017777 176452 176452  MOV      @ROBUF, @ROBUF ;STATIC LINES EQUAL ONES
617 002554 017700 176446          MOV      @ROBUF, %0 ;DATA REGISTER TO %0
618 002560 022700 177777          CMP      #-1, %0
619 002564 001401          BEQ     .+4
620 002566 104000          HLT
                                     ;REG 0 SHOULD = ALL ONES

621
622
623
624
625
626 002570 000004          ;READY BIT IS IN A ONE STATE
627 002572 012715 000003  SCOPE
628 002576 011500          MOV      #3, @RS ;CSR0 AND CSR1
629 002600 022700 100203  MOV      (RS), @R0
630 002604 001401          CMP      #100203, @R0
                                     BEQ     .+4

```



```

628 002606 104000          HLT
629
630          ;CAN WE RAISE INTERUPT "A"
631 002610 000004          SCOPE
632 002612 052737 000340 177776  BIS      #340,2#PSW      ;LOCK OUT INTERRUPTS
633 002620 012706 001200          MOV      #STKPTR,%6
634 002624 012777 002646 176400  MOV      #TST4,2ORVECA ;INTERRUPT RETURN POINTER
635 002632 012715 000101          MOV      #101,2RS      ;INTERRUPT ENABLE AND CSRO
636 002636 005037 177776          CLR      2#PSW
637 002642 000240          NOP
638 002644 104000          HLT          ;NO "A" INTERRUPT
639 002646 005015          CLR      2RS
640 002650 016777 176360 176354  MOV      DRLVL,2ORVECA ;MOVE .+2 TO "A" INTERRUPT VECTOR
641
642          ;RAISE INTERRUPT "B"
643 002656 000004          SCOPE
644 002660 012706 001200          MOV      #STKPTR,%6
645 002664 052737 000340 177776  BIS      #340,2#PSW
646 002672 012777 002716 176336  MOV      #TINT1,2ORVECB
647 002700 012715 000042          MOV      #2,2RS      ;IE AND CSRI
648 002704 042737 000377 177776  BIC      #377,2#PSW
649 002712 000240          NOP
650 002714 104000          HLT          ;NO B INTERRUPT
651 002716 005015          CLR      2RS
652
653          ;TEST FOR INTERRUPT FROM DEVICE
654 002720 016777 176320 176306  MOV      PL,2DRLVL
655 002726 042737 000340 177776  BIC      #340,2#PSW      ;PROCESSOR LEVEL ZERO
656 002734 012777 002766 176270  MOV      #TINT1,2ORVECA
657 002742 012706 001200          MOV      #STKPTR,%6      ;STACK POINTER
658 002746 042777 000100 176246  BIC      #100,2ORCSR      ;CLEAR INTERRUPT ENABLE
659 002754 052777 000101 176240  BIS      #101,2ORCSR      ;SET INTERRUPT ENABLE-AND CSRO
660 002762 000240          NOP
661 002764 104000          HLT          ;NO DEVICE INTERRUPT OCCURED
662 002766 000004          TINT1: SCOPE
663
664          ;TEST FOR INTERRUPT FROM THE DEVICE
665 002770 042737 000340 177776  BIC      #340,2#PSW
666 002776 052737 000040 177776  BIS      #040,2#PSW      ;SET TO PRIORITY LEVEL 1
667 003004 012777 00303J 176220  MOV      #TINT2,2ORVECA ;INTERRUPT VECTOR ADDRESS
668 003012 012706 001200          MOV      #STKPTR,%6      ;SET UP STACK POINTER
669 003016 042777 000100 176176  BIC      #100,2ORCSR      ;CLEAR INTERRUPT ENABLE
670 003024 052777 000101 176170  BIS      #101,2ORCSR      ;SET INTERRUPT ENABLE-AND CSRO
671 003032 000240          NOP
672 003034 104000          HLT          ;NO DEVICE INTERRUPT OCCURED
673
674          TINT2: SCOPE
675 003040 042737 000340 177776  BIC      #340,2#PSW
676 003046 052737 000100 177776  BIS      #100,2#PSW      ;SET TO PRIORITY LEVEL 2
677 003054 012777 003106 176150  MOV      #TINT3,2ORVECA ;INTERRUPT VECTOR ADDRESS
678 003062 012706 001200          MOV      #STKPTR,%6      ;SET UP STACK POINTER
679 003066 042777 000100 176126  BIC      #100,2ORCSR      ;CLEAR INTERRUPT ENABLE
680 003074 052777 000101 176120  BIS      #101,2ORCSR      ;SET INTERRUPT ENABLE-AND CSRO
681 003102 000240          NOP
682 003104 104000          HLT          ;NO DEVICE INTERRUPT OCCURED
683
  
```

```

684 003106 000004          TINT3: SCOPE
685                                ;TEST FOR INTERRUPT FROM THE DEVICE
686 003110 042737 000340 177776      BIC      #340,2#PSW
687 003116 052737 000140 177776      BIS      #140,2#PSW      ;SET TO PRIORITY LEVEL 3
688 003124 012777 003156 176100      MOV      #TINT4,2ORVECA ;TINTERRUPT VECTOR ADDRESS
689 003132 012706 001200              MOV      #STKPTR,%6     ;SET UP STACK POINTER
690 003136 042777 000100 176056      BIC      #100,2ORCSR    ;CLEAR INTERRUPT ENABLE
691 003144 052777 000101 176050      BIS      #101,2ORCSR    ;SET INTERRUPT ENABLE-AND CSRC
692 003152 000240              NOP
693 003154 104000              HLT
694 003156 000004          TINT4: SCOPE
695                                ;TEST FOR INTERRUPT FROM DEVICE
696                                BIC      #340,2#PSW
697 003160 042737 000340 177776      BIS      #200,2#PSW    ;RAISE PROCESSOR PRIORITY TO LEVEL 4
698 003166 052737 000200 177776      MOV      #TINT5,2ORVECA ;IN CASE OF INTERRUPT
699 003174 012777 003236 176030      MOV      #STKPTR,%6     ;SET STACK POINTER
700 003202 012706 001200              BIC      #100,2ORCSR    ;CLEAR INTERRUPT ENABLE
701 003206 042777 000100 176006      BIS      #101,2ORCSR    ;SET INTERRUPT ENABLE AND CSRC
702 003214 052777 000101 176000      NOP
703 003222 000240              HLT
704 003224 042777 000100 175770      BIC      #100,2ORCSR    ;LET INTERRUPT OCCUR
705 003232 000240              NOP
706 003234 104000              HLT
707 003236 000004          TINT5: SCOPE
708                                ;TEST FOR NO INTERRUPT FROM DEVICE (HIGHEST PROCESSOR PRIORITY)
709                                BIS      #340,2#PSW    ;RAISE PROCESSOR PRIORITY TO HIGHEST LEVEL
710 003240 052737 000340 177776      MOV      #TINT6,2ORVECA ;IN CASE OF INTERRUPT
711 003246 012777 003306 175756      MOV      #STKPTR,%6     ;SET STACK POINTER
712 003254 012706 001200              BIC      #100,2ORCSR    ;CLEAR INTERRUPT ENABLE
713 003260 042777 000100 175734      BIS      #101,2ORCSR
714 003266 052777 000101 175726      NOP
715 003274 000240              BIC      #100,2ORCSR
716 003276 042777 000100 175716      BR      .+4
717 003304 000401              TINT6: HLT
718 003306 104000              ;WITH NO INTERRUPT, BRANCH OVER HALT
719 003310 000004          ;INTERRUPT OCCURED
720                                SCOPE
721                                ;TEST FOR NO INTERRUPT FROM DEVICE
722                                BIC      #340,2#PSW
723 003312 042737 000340 177776      BIS      #240,2#PSW    ;RAISE PROCESSOR PRIORITY TO LEVEL 5
724 003320 052737 000240 177776      MOV      #TINT7,2ORVECA ;IN CASE OF INTERRUPT
725 003326 012777 003366 175676      MOV      #STKPTR,%6     ;SET STACK POINTER
726 003334 012706 001200              BIC      #100,2ORCSR    ;CLEAR INTERRUPT ENABLE
727 003340 042777 000100 175654      BIS      #101,2ORCSR    ;SET INTERRUPT ENABLE AND CSRC
728 003346 052777 000101 175646      NOP
729 003354 000240              BIC      #100,2ORCSR
730 003356 042777 000100 175636      BR      .+4
731 003364 000401              TINT7: HLT
732 003366 104000              ;DON'T LEAVE IT SET
733 003370 000004          ;WITH NO INTERRUPT, BRANCH OVER HALT
734                                ;INTERRUPT OCCURED
735                                SCOPE
736                                ;TEST FOR NO INTERRUPT FROM DEVICE
737 003372 042737 000340 177776      BIC      #340,2#PSW
738 003400 052737 000300 177776      BIS      #300,2#PSW    ;RAISE PROCESSOR PRIORITY TO LEVEL 6
739 003406 012777 003446 175616      MOV      #TINT8,2ORVECA ;IN CASE OF INTERRUPT
740 003414 012706 001200              MOV      #STKPTR,%6     ;SET STACK POINTER
741 003420 042777 000100 175574      BIC      #100,2ORCSR    ;CLEAR INTERRUPT ENABLE

```


796	003666	012667	000160		MOV	(6)+, SAVPC	; PC OF FAILING ROUTINE
797	003672	012667	000156		MOV	(6)+, SAVCC	; CC OF ERROR CONDITION
798	003676	024646			CMP	-(6), -(6)	; REPOSITION THE STACK
799	003700	105777	000144		TSTB	@TCSR	; WAIT FOR FLAG
800	003704	100375			BPL	.-4	; IF NOT UP.
801	003706	012777	000215	000132	MOV	@215, @TDBR	; CR
802	003714	105777	000130		TSTB	@TCSR	
803	003720	100375			BPL	.-4	
804	003722	012777	000212	000116	MOV	@212, @TDBR	; LINE FEED
805	003730	105777	000114		TSTB	@TCSR	
806	003734	100375			BPL	.-4	
807	003736	010267	000076		MOV	%2, SAVR2	; SAVE R2
808	003742	010367	000074		MOV	%3, SAVR3	; SAVE R3
809	003746	010467	000072		MOV	%4, SAVR4	; SAVE R4
810	003752	016702	000074		MOV	SAVPC, %2	
811	003756	004767	000074		JSR	%7, PRTAB	; PRINT OCTAL NUMBER
812	003762	012777	000240	000056	MOV	@240, @TDBR	
813	003770	105777	000054		TSTB	@TCSR	; SPACE BETWEEN WORDS
814	003774	100375			BPL	.-4	
815	003776	016702	000052		MOV	SAVCC, %2	
816	004002	004767	000050		JSR	%7, PRTAB	; PRINT OCTAL NUMBER
817	004006	004767	0000476		JSR	%7, MOREID	; DEVICE ADDRESS AND VECTORS
818	004012	016702	000022		MOV	SAVR2, %2	; RESTORE REGISTERS
819	004016	016703	000020		MOV	SAVR3, %3	
820	004022	016704	000016		MOV	SAVR4, %4	
821	004026	005777	175164		TST	@SR	; TEST FOR HALT SWITCH
822	004032	100001			BPL	.+4	
823	004034	000000			HALT		; HALT ON ERROR SET
824	004036	000002			RTI		; RETURN TO MAIN STREAM
825	004040	000000					
826	004042	000000					
827	004044	000000					
828	004046	177566			TDBR:	177566	; DATA
829	004050	177564			TCSR:	177564	; STATUS
830	004052	000000			SAVPC:	0	
831	004054	000000			SAVCC:	0	
832							
833	004056	005067	000252		PRTAB:	CLR BINCT	
834	004062	005067	000244		CLR	WGTCT	
835	004066	012704	004340		MOV	@1, IST, %4	; GET LIST ADDRESS
836	004072	012767	000035	000236	MOV	@5, ASCNT	
837	004100	012767	000007	000220	MOV	@7, SEVEN	
838	004106	012767	000001	000214	MOV	@1, DECHL	
839	004114	105777	177730		WAIT1:	TSTB @TCSR	
840	004120	100375			BPL	WAIT1	
841	004122	005702			TST	%2	
842	004124	100404			BMI	MINUS	; NEG SIGN PRINT 1
843	004126	012777	000260	177712	MOV	@260, @TDBR	; POS SIGN PRINT 0
844	004134	000403			BR	STAR	
845	004136	012777	000261	177702	MINUS:	MOV @261, @TDBR	
846	004144	016703	000156		STAR:	MOV SEVEN, %3	; PUT MASK IN R3
847	004150	010267	000150		MOV	%2, TOODLE	; GET READY TO DOODLE NUMBER IN TOODLE
848	004154	010167	000144		COM	TOODLE	; COMPENSATES FOR COMPLEMENT DURING BIC
849	004160	016703	000140		BIC	TOODLE, %3	; AND IN OCTAL CHARACTER
850	004164	001410			BEQ	WRTOC	; ZERO, WRITE 0 IN LIST
851	004166	066767	000136	000136	MKNUM:	ADD DECHL, WGTCT	; COUNT UP TO

```

004174 005267 000134 INC BINCT ;AND RECORD
004200 026703 000126 CMP WGTCT,X3 ;SAME BINARY WEIGHT
004204 001370 BNE MKNUM ;KEEP COUNTN
004206 062767 000260 000120 WRTOC: ADD #260,BINCT ;ADD ASCII PREFIX
004214 016724 000114 MOV BINCT,(4)+ ;WRITE ASCII CHAR IN LIST
004220 066767 000102 000102 ADD SEVEN,DECM ;EXPAND BINARY WEIGHT
004236 005067 000100 CLR WGTCT
004232 005067 000076 CLR BINCT
004236 005367 000074 DEC ASCNT
004242 001410 BEQ XLIST ;5 CHAR IN LIST
004244 012703 000003 MOV #3,%3 ;SET X3 FOR ADD LOOP
004250 046767 000052 000050 MRODD: ADD SEVEN,SEVEN ;MAKING SEVENTY BY SEVEN
004256 005303 DEC X3
004260 001373 SNE MRODD
004262 000730 BR STAR ;NX SEVEN SET GET NX OCTAL
004264 012767 000005 000044 XLIST: MOV #5,ASCNT ;SEND 5 CHAR TO TTY
004272 105777 177552 WAIT2: TSTB @TCSR
004276 100375 BPL WAIT2
004300 014477 177542 MOV -(4),@TDBR
004304 005367 000026 DEC ASCNT
004310 001401 BEQ HOFHM ;FINISH PRINTING GET NXT NUM
004312 000767 BR WAIT2
004314 105777 177530 HOFHM: TSTB @TCSR
004320 100375 BR *4
004322 000707 RTS ;HEAD FOR HOME
004324 000000 TOODLE: 0
004326 000000 SEVEN: 0
004330 000000 DECM: 0
004332 000000 WGTCT: 0
004334 000000 BINCT: 0
004336 000000 ASCNT: 0
004340 000000 LIST: 0
004342 000000
004344 000000
004346 000000
004350 000000
004352 022606 ;SCOPE LOOP ROUTINE ENTERED BY USER TRAP
004354 012637 177776 SCOPEB: CMP (6)+,%6 ;REPOSITION THE STACK
004360 000177 000106 MOV (6)+,%#PSW
JMP @RETURN ;SCOPE RETURN

;SCOPE OR/AND ITERATION LOOP FOR EACH TEST 4000 TIMES
004364 032777 040000 174624 ;SCOPE: BIT #SW14,@SR ;TEST SR FOR SCOPE
004370 001367 BNE SCOPEB ;YES SCOPE
004374 005737 001240 TST @XORFLG
004400 100012 BPL IS
004402 013746 000004 MOV @4,-(%6)
004406 012737 004500 000004 MOV @XOR,@4
004410 012737 000031 177060 MOV #31,@#177060
004414 012637 000004 MOV (%6)+,@4
004418 032777 004500 174562 IS: BIT #SW11,@SR ;NO - TEST FOR ITERATION
004422 001011 BNE SCOPEA ;INHIBIT ITERATION
004426 026767 000026 000022 CMP SCOPEF,ICOUNT ;EXIT - DONE
004430 001403 BEQ SCOPEG ;INCREMENT COUNT
004434 005267 000016 INC SCOPEF ;LOOP SOME MORE
004438 000737 BR SCOPEB

```

```

908 004454 005067 000010 SCOPEG: CLR SCOPEF ;CLEAR COUNT
909 004460 011667 000006 SCOPEA: MOV %6,RETURN ;SAVE SCOPE RETURN POINTER
910 004464 007702 RTI ;RETURN INLINE-NEXT TEST
911 004466 004J00 ICOUNT: 4000
912 004470 000000 SCOPEF: 0 ;COUNT LOCATION FOR ITERATION LOOP
913 004472 001524 RETURN: BEGIN ;ADDRESS OF LAST TEST
914 004474 000167 173500 JMP 200
915
916 004500 022626 XOR: CMP (%6)+, (%6)+
917 004502 012637 000004 MOV (%6)+, @#4
918 004506 000721 BR SCOPEB
919 ;PRINT DEVICE ADDRESS AND VECTOR
920 004510 012777 000240 177330 MOREID: MOV @#240, @TDBR
921 004516 105777 177326 TSTB @TCSR
922 004522 100375 BPL -4
923 004524 013702 001222 MOV @#DRCSR, %2
924 004530 004767 177322 JSR %7, PRTAB
925 004534 012777 000240 177304 MOV @#240, @TDBR
926 004540 105777 177302 TSTB @TCSR
927 004546 100375 BPL -4
928 004550 016702 174456 MOV @#VECA, %2
929 004554 004767 177276 JSR %7, PRTAB
930 004560 012777 000215 177260 MOV @#215, @TDBR
931 004566 105777 177256 TSTB @TCSR
932 004572 100375 BPL -4
933 004574 012777 000212 177244 MOV @#212, @TDBR
934 004602 105777 177242 TSTB @TCSR
935 004606 100375 BPL -4
936 004610 005077 177232 CLR @TDBR
937 004614 105777 177230 TSTB @TCSR
938 004620 100375 BPL -4
939 004622 000207 RTS %7 ;BACK TO PRINT
940
941 ;ENTER HERE FOR POWER FAIL
942
943 PFAIL: MOV %0, -(6) ;SAVE REGISTER OFF STACK
944 004626 010146 MOV %1, -(6) ;WHEN POWERING DOWN
945 004630 010246 MOV %2, -(6)
946 004632 010346 MOV %3, -(6)
947 004634 010446 MOV %4, -(6)
948 004636 010546 MOV %5, -(6)
949 004640 016746 173160 MOV %6, -(6)
950 004644 010637 004667 MOV %6, @#SAVR6 ;STORE STACK POSITION
951 004650 012737 004662 000024 MOV @#RESTAR, @#24
952 004656 000000 HALT ;HALT ON POWER DOWN NORMAL
953 004660 000000 ;STACK IS SHAVED HERE
954 004662 016706 177772 RESTAR: MOV SAVR6, %6 ;RESTORE REGISTER OFF STACK
955 004666 012667 173132 MOV (%6)+, %4 ;WHEN POWERING UP
956 004670 012605 MOV (%6)+, %5
957 004674 012604 MOV (%6)+, %4
958 004678 012603 MOV (%6)+, %3
959 004700 012602 MOV (%6)+, %2
960 004702 012601 MOV (%6)+, %1
961 004706 000137 001472 JMP @#RESTART

```

```

964
965
966
967 004712 000000
968 004714 017701 174276
969 004720 004737 004742
970 004724 000000
971 004726 017701 174264
972 004732 004737 004772
973 004736 000137 001444
974 004742 012700 001200
975 004746 010120
976 004750 062701 000002
977 004754 010120
978 004756 062701 000002
979 004762 010120
980 004764 005301
981 004766 010120
982 004770 000207
983
984 004772 012700 001210
985 004776 010120
986 005000 005721
987 005002 010120
988 005004 005721
989 005006 010120
990 005010 000207
991
992
993 005012
994 005012
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
  
```

```

;ENTER HERE FOR UNIQUE SELECTION OF DR11C
SPEC0: HALT ;PLACE ADDRESS OF DR11-C CONTROL STATUS
        MOV   @JSR,R1
        JSR   PC,@#SPEC1
        HALT
        MOV   @JSR,R1
        JSR   PC,@#SPEC2
        JMP   @#START

SPEC1:  MOV   #RCSR,R0 ;SET TABLE ADDRESS
        MOV   R1,(R0)+ ;LOAD INTO TABLE STARTING AT RCSR
        ADD   #2,R1 ;STEP TO ADDRESS OF DROUTBUF
        MOV   R1,(R0)+ ;LOAD INTO TABLE
        ADD   #2,R1 ;STEP TO ADDRESS OF DRINBUF
        MOV   R1,(R0)+ ;LOAD INTO TABLE
        DEC   R1 ;FORM ADDRESS OF DROUTBUF+1
        MOV   R1,(R0)+ ;LOAD INTO TABLE
        RTS   PC

SPEC2:  MOV   #RCSR1,R0
        MOV   R1,(R0)+ ;LOAD INTO TABLE
        TST   (R1)+
        MOV   R1,(R0)+
        TST   (R1)+
        MOV   R1,(R0)+
        RTS   PC

.LIST ME
TYP: .STYPE
STARS
  .IF B
;*****
  .IFF
  .NLIST
  .REPT
  .LIST
;*****
  .NLIST
  .ENOR
  .LIST
  .ENDC

.SBTTL TYPE ROUTINE

;#ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
;#THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
;#NOTE1: SNULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
;#NOTE2: SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
;#NOTE3: SFILLC CONTAINS THE CHARACTER TO FILL AFTER.
;#
;#CALL:
;#1) USING A TRAP INSTRUCTION
;# TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
;#OR
;# TYPE
  
```

```

1020      ;*      MESAOR
1021      ;*
1022
1023      005012  105767  000217      $TYPE:  TSTB      $TFPLG      ;; IS THERE A TERMINAL?
1024      005016  100002                      BPL          1$          ;; BR IF YES
1025      005020  000000                      H&LT         ;; HALT HERE IF NO TERMINAL
1026      005022  000407                      BR           3$          ;; LEAVE
1027      005024  010046                      1$:  MOV      RO,-(SP)    ;; SAVE RO
1028      005026  017600  000002      MOV      @2(SP),RO      ;; GET ADDRESS OF ASCIZ STRING
1029      001
1030      .IF OF $MAIL
1031      CMPB   #APTENV,$ENV      ;; RUNNING IN APT MODE
1032      BNE    62$                ;; NO, GO CHECK FOR APT CONSOLE
1033      BITB   #APTSPOOL,$ENVM    ;; SPOOL MESSAGE TO APT
1034      BEQ    62$                ;; NO, GO CHECK FOR CONSOLE
1035      MOV    RO,61$             ;; SETUP MESSAGE ADDRESS FOR APT
1036      JSR   PC,$ATY3           ;; SPOOL MESSAGE TO APT
1037      .WORD  0                 ;; MESSAGE ADDRESS
1038      61$:  BITB   #APTCSUP,$ENVM  ;; APT CONSOLE SUPPRESSED
1039      62$:  BNE    60$                ;; YES, SKIP TYPE OUT
1040      .ENDC
1041      2$:  MOVB   (RO)+,-(SP)      ;; PUSH CHARACTER TO BE TYPED ONTO STACK
1042      BNE    4$                  ;; BR IF IT ISN'T THE TERMINATOR
1043      TST   (SP)+                ;; IF TERMINATOR POP IT OFF THE STACK
1044      60$:  MOV    (SP)+,RO        ;; RESTORE RO
1045      3$:  ADD    #2,(SP)          ;; ADJUST RETURN PC
1046      RTI
1047      4$:  CMPB   #HT,(SP)        ;; BRANCH IF <HT>
1048      BEQ   8$                  ;; BRANCH IF NOT <CRLF>
1049      6$:  CMPB   #CRLF,(SP)     ;; BRANCH IF NOT <CRLF>
1050      BNE   5$                  ;; POP <CR><LF> EQUIV
1051      TST   (SP)+                ;; TYPE A CR AND LF
1052      TYPE  $CRLF
1053      BR    2$                  ;; GET NEXT CHARACTER
1054      5$:  JSR   PC,$TYPEC        ;; GO TYPE THIS CHARACTER
1055      6$:  CMB   $FILLC,(SP)+     ;; IS IT TIME FOR FILLER CHARS.?
1056      BNE   2$                  ;; IF NO GO GET NEXT CHAR.
1057      MOV   $NULL,-(SP)         ;; GET # OF FILLER CHARS. NEEDED
1058      AND   THE NULL CHAR.
1059      7$:  DECB   1(SP)           ;; DOES A NULL NEED TO BE TYPED?
1060      BLT   6$                  ;; BR IF NO--GO POP THE NULL OFF OF STACK
1061      JSR   PC,$TYPEC        ;; GO TYPE A NULL
1062      DECB  $CHARCNT          ;; DO NOT COUNT AS A COUNT
1063      BR    7$                  ;; LOOP
1064
1065      ;HORIZONTAL TAB PROCESSOR
1066
1067      8$:  MOVB   #40,(SP)        ;; REPLACE TAB WITH SPACE
1068      9$:  JSR   PC,$TYPEC        ;; TYPE A SPACE
1069      BITB   #7,$CHARCNT        ;; BRANCH IF NOT AT
1070      BNE   9$                  ;; TAB STOP
1071      TST   (SP)+                ;; POP SPACE OFF STACK
1072      BR    2$                  ;; GET NEXT CHARACTER
1073      $TYPEC: TSTB   3$TPS        ;; WAIT UNTIL PRINTER IS READY
1074      BPL   $TYPEC
1075      MOVB   2(SP),3$TPB      ;; LOAD CHAR TO BE TYPED INTO DATA REG.

```



```

1076 005172 122766 000015 000002      CMPB    #15,2(SP)      ;; BRANCH IF
1077 005200 001003      BNE     IS             ;; NOT <CR>
1078 005202 105067 000014      CLRB    $CHARCNT      ;;
1079 005206 000406      BR      $TYPEX        ;; EXIT
1080 005210 122766 000012 000002 1$:    CMPB    #12,2(SP)      ;; BRANCH IF
1081 005216 002002      BGE     $TYPEX        ;; <LF>
1082 005220 105227      INCB   (PC)+         ;; INC SPACE
1083 005222 000000      $CHARCNT: WORD      0      ;; COUNT
1084 005224 000207      $TYPEX: RTS         PC
1085                                     ;; EQUATES
1086                                     THT=11
1087                                     TCRLF=200
1088
1089                                     .IIF NOF $STPS,$STPS:
1090 005226 177564      .IIF EQ  -$STPS,$STPS: .WORD 177564      ;; TTY PRINTER STATUS REG. ADDRESS
1091                                     .IIF NOF $STPB,$STPB:
1092 005230 177566      .IIF EQ  -$STPB,$STPB: .WORD 177566      ;; TTY PRINTER BUFFER REG. ADDRESS
1093                                     .IIF NOF $NULL,$NULL:
1094 005232      000      .IIF EQ  -$NULL,$NULL: .BYTE 0           ;; CONTAINS NULL CHARACTER FOR FILLS
1095                                     .IIF NOF $FILLS,$FILLS:
1096 005233      002      .IIF EQ  -$FILLS,$FILLS: .BYTE 2           ;; CONTAINS # OF FILLER CHARACTER
1097                                     .IIF NOF $FILLC,$FILLC:
1098 005234      012      .IIF EQ  -$FILLC,$FILLC: .BYTE 12          ;; INSERT FILL CHARS. AFTER A "LI
1099                                     .IIF NOF $STPLG,$STPLG:
1100 005235      000      .IIF EQ  -$STPLG,$STPLG: .BYTE 0           ;; "TERMINAL AVAILABLE" FLAG (BIT
1101                                     .IIF NOF $QUES,$QUES:
1102 005236      077      .IIF EQ  -$QUES,$QUES: .ASCII "?"         ;; QUESTION MARK
1103                                     .IIF NOF $CRLF,$CRLF:
1104 005237      015      000      .IIF EQ  -$CRLF,$CRLF: .ASCIZ <15>      ;; CARRAIGE RETURN
1105                                     .IIF NOF $LF,$LF:
1106 005241      012      000      .IIF EQ  -$LF,$LF: .ASCIZ <12>      ;; LINEFEED
1107 005244      005244      .IIF NE IS .EVEN
1108 005244      .STRAP
1109 005244
1110      001      STARS
1111      .IF B
1112      ;*****
1113      .IFF
1114      .MLIST
1115      .REPT
1116      .LIST
1117      ;*****
1118      .MLIST
1119      .ENDR
1120      .LIST
1121      .ENDC
1122
1123      .SBTTL TRAP DECODER
1124
1125      ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
1126      ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
1127      ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
1128      ;*GO TO THAT ROUTINE.
1129
1130 005244 010046      .IF B
1131      $TRAP: MOV     RD,-(SP)      ;; SAVE RD
1132      .IFF

```

```

1132          STRAP:  MOV  2(SP),-(SP)      :: ASSUME THE STATUS OF
1133          BIC  #20,(SP)                :: THE CALLER--DONOT ALLOW
1134          MOV  #18,-(SP)                :: T-BIT TRAPS
1135          RTI                                :: SET THE NEW STATUS
1136          IS:   MOV  RO,-(SP)            :: SAVE RO
1137          .ENOC
1138 005246      016600 000002          MOV  2(SP),RO      :: GET TRAP ADDRESS
1139 005252      005740          TST  -(RO)          :: BACKUP BY 2
1140 005254      111000          MOVB (RO),RO        :: GET RIGHT BYTE OF TRAP
1141          001
1142          .IF NB
1143          BPL  STRAP1                    :: NON-USER TRAP BELOW 200
1144          BIC  #177,RO                  :: STRIP AWAY THE JUNK
1145          JMP  (PC)                      :: USER TRAP ABOVE 177, GO TO
1146          .WORD                          :: USER TRAP HANDLER-
1147          STRAP1:
1148          .ENOC
1149          001
1150          .IF NB
1151          CMP  #$TERM,RO                 :: CHECK FOR OUT OF BOUNDS
1152          BGT  .+6                       :: BR IF OK
1153          HALT                          :: OUT OF BOUNDS
1154          BR   .-2                       :: HANGLUP
1155          .ENOC
1156 005256      006300          ASL  RO          :: POSITION FOR INDEXING
1157 005260      016000          MOV  STRPAD(RO),RO  :: INDEX TO TABLE
1158 005264      000200          RTS  RO          :: GO TO ROUTINE
1159
1160          .MACRO  SETTRAP A,B,MSG
1161          $$SET  A,B,\<TRAP+STRP>,\STRP,<MSG>
1162
1163          .NLIST
1164          STRP=STRP+1
1165          .LIST
1166          .ENDM  SETTRAP
1167          .MACRO  $$SET  A,B,C,D,COMNT
1168          .IF EQ STRP
1169
1170          .SBTTL  TRAP TABLE
1171
1172          ::*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1173          ::*BY THE "TRAP" INSTRUCTION.
1174
1175          :
1176          : ROUTINE
1177          : -----
1178          STRPAD:
1179          .ENOC
1180          .IIF NDF GNS,.NLIST
1181          A= C
1182          .IIF NDF GNS,.LIST
1183          B ;:CALL=A TRAP+D(C) COMNT
1184          .ENDM  $$SET
1185          .MACRO  TRMTRP
1186          $TERM=-STRPAD
1187          .ENDM  TRMTRP
1188          .IF DF $TYPE
1189          SETTRAP TYPE,$TYPE,↑/TTY TYPEOUT ROUTINE/
1190          $$SET  TYPE,$TYPE,\<TRAP+STRP>,\STRP,<TTY TYPEOUT ROUTINE>
1191          .IF EQ STRP

```

```

1188
1189          .SBTTL TRAP TABLE
1190
1191          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1192          ;*BY THE "TRAP" INSTRUCTION.
1193
1194          :          ROUTINE
1195          :          -----
1196          005266      $TRPAD:
1197          001          .ENDC
1198          005266      005012      $TYPE ;;CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE
1199          000          .ENDC
1200          001          .IF DF $TYPOC
1201          SETTRAP TYPOC,$TYPOC,↑/TYPE OCTAL NUMBER (WITH LEADING ZEROS)/
1202          SETTRAP TYPOS,$TYPOS,↑/TYPE OCTAL NUMBER (NO LEADING ZEROS)/
1203          SETTRAP TYPON,$TYPON,↑/TYPE OCTAL NUMBER (AS PER LAST CALL)/
1204          000          .ENDC
1205          001          .IF DF $TYPDS
1206          SETTRAP TYPDS,$TYPDS,↑/TYPE DECIMAL NUMBER (WITH SIGN)/
1207          000          .ENDC
1208          001          .IF DF $TYPBN
1209          SETTRAP TYPBN,$TYPBN,↑/TYPE BINARY (ASCII) NUMBER/
1210          000          .ENDC
1211          001          .IF DF $RDCHR
1212          SETTRAP RDCHR,$RDCHR,↑/TTY TYPEIN CHARACTER ROUTINE/
1213          000          .ENDC
1214          001          .IF DF $RDLIN
1215          SETTRAP RDLIN,$RDLIN,↑/TTY TYPEIN STRING ROUTINE/
1216          000          .ENDC
1217          001          .IF DF $RDOCT
1218          SETTRAP RDOCT,$RDOCT,↑/READ AN OCTAL NUMBER FROM TTY/
1219          000          .ENDC
1220          001          .IF DF $RDDEC
1221          SETTRAP RDDEC,$RDDEC,↑/READ A DECIMAL NUMBER FROM TTY/
1222          000          .ENDC
1223          001          .IF DF $SAVREG
1224          SETTRAP SAVREG,$SAVREG,↑/SAVE R0-R5 ROUTINE/
1225          SETTRAP RESREG,$RESREG,↑/RESTORE R0-R5 ROUTINE/
1226          000          .ENDC
1227          001          .IF DF $R2A
1228          SETTRAP R2AZ,$R2AZ
1229          SETTRAP R2AZ,$R2AZ
1230          SETTRAP R2AZ0,$R2AZ0
1231          000          .ENDC
1232          005270      005015      047531      020125      MESS1: .ASCIZ <15><12>'YOU ARE ON AN XOR TESTER'<15><12>
1233          005276      051101      020105      047117
1234          005304      040440      020116      047530
1235          005312      020122      042524      052123
1236          005320      051105      005015      000
1237          000001
          .END
  
```

ASCNT	004336	DRIBUF	001276	RCSR	001200	SW00	= 000001	TKVEC	= 000060
BEGIN	001524	DRLVL	001274	RCSR1	001210	SW01	= 000002	T000LE	004324
BINCT	004334	DROBUF	001224	RESTAR	004662	SW02	= 000004	TPVEC	= 000064
BIT0	000001	DRVECA	001232	RESVEC	= 000010	SW03	= 000010	TRAPVE	= 000034
BIT00	= 000001	DRVECB	001236	RETURN	004472	SW04	= 000020	TRTVEC	= 000014
BIT01	= 000002	DSWR	= 177570	RSTART	001472	SW05	= 000040	TST4	002646
BIT02	= 000004	EMTVEC	= 000020	R0	= %000000	SW06	= 000100	TST5	002716
BIT03	= 000010	END	003464	R1	= %000001	SW07	= 000200	TST6	002474
BIT04	= 000020	END1	003650	R2	= %000002	SW08	= 000400	TST9	002530
BIT05	= 000040	ERRVEC	= 000004	R3	= %000003	SW09	= 001000	TYP	005012
BIT06	= 000100	FIRST	001350	R4	= %000004	SW1	= 000002	TYPE	= 104400
BIT07	= 000200	HDFHM	004314	R5	= %000005	SW10	= 002000	WAIT1	004114
BIT08	= 000400	HLT	= 104000	R6	= %000006	SW11	= 004000	WAIT2	004272
BIT09	= 001000	ICOUNT	004466	R7	= %000007	SW12	= 010000	WGTCT	004332
BIT1	= 000002	IOTVEC	= 000020	SAVCC	004054	SW13	= 020000	WRTOC	004206
BIT10	= 002000	LIST	004340	SAVPC	004052	SW14	= 040000	XLIST	004264
BIT11	= 004000	LOGIC	003636	SAVR2	004040	SW15	= 100000	XOR	004500
BIT12	= 010000	MESS1	005270	SAVR3	004042	SW16	= 000004	XORA	001430
BIT13	= 020000	MINUS	004136	SAVR4	004044	SW17	= 000010	XORFLG	001240
BIT14	= 040000	MINUM	004166	SAVR6	004060	SW18	= 000020	SCHARC	005222
BIT15	= 100000	MORDD	004250	SCOPEA	004460	SW19	= 000040	SCRLF	005237
BIT2	= 000004	MOREID	004510	SCOPEB	004352	SW20	= 000100	SFILLC	005234
BIT3	= 000010	PC	= %000007	SCOPEF	004470	SW21	= 000200	SFILLS	005233
BIT4	= 000020	PFATL	004624	SCOPEG	004454	SW22	= 000400	SLF	005241
BIT5	= 000040	PIRQ	= 177772	SEVEN	004326	SW23	= 001000	\$NULL	005232
BIT6	= 000100	PIRQVE	= 000240	SP	= %000006	SW24	= 002024	\$QUES	005236
BIT7	= 000200	PL	001244	SPEC	001234	SW25	= 000014	\$TPB	005230
BIT8	= 000400	PRTAB	004056	SPECO	004712	SW26	= 000200	\$TPFLG	005235
BIT9	= 001000	PRO	= 000000	SPEC1	004742	SW27	= 004050	\$TPS	005226
BPTVEC	= 000014	PR1	= 000040	SPEC2	004772	SW28	= 004046	\$TRAP	005244
BUFTST	001760	PR2	= 000100	SR	001216	SW29	= 000011	\$TRP	= 000001
COUNT	001242	PR3	= 000140	STACK	= 001100	TINT1	002766	\$TRPAD	005266
CSR	= 167770	PR4	= 000200	STAR	004144	TINT2	003036	\$TYPE	005012
DCISP	= 177570	PR5	= 000240	START	001444	TINT3	003106	\$TYPEC	005156
DECHL	004330	PR6	= 000300	START1	001246	TINT4	003156	\$TYPEX	005224
DISPLA	001220	PR7	= 000340	STKLMT	= 177774	TINT5	003236	.HLT	003654
DISPRG	000174	PS	= 177776	SWREG	000176	TINT6	003306	.SCOPE	004364
DRBHIO	001230	PSW	= 177776			TINT7	003366	.	= 005325

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

#DZDRCF DZDRCF/SOL=DZDRCF
 RUN-TIME: 127.4 SECONDS
 RUN-TIME RATIO: 75/20=3.7
 CORE USED: 11K (21 PAGES)

