

DZ11

8-LINE ASYNCHMUX TESTS MD-11-DZDZA-B

EP-DZDZA-B-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains 120 frames of test data, arranged in a 10x12 grid. Each frame displays a small, high-contrast image, likely a test pattern or waveform, used for digital logic testing. The frames are organized into 10 rows and 12 columns. The data is presented in a structured, grid-like format, typical of digital logic testing equipment output.

78
79
80
81
82
83
84
85
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

2.2 STORAGE

PROGRAM WILL USE ALL 8K OF MEMORY EXCEPT WHERE ABL AND BOOTSTRAP LOADER RESIDE. LOCATION 1500 THRU 2000 ARE ESPECIALLY TO BE NOTED AND TO BE UNTOUCHED BY OPERATOR AFTER PARAMETERS HAVE BEEN INPUT FROM CONSOLE (SW00=1); OR AFTER THE 'AUTO SIZING' HAS BEEN DONE. THESE LOCATIONS MAY BE CHANGED IF THE USER UNDERSTANDS THEIR MEANING AND DIFFERENT PARAMETERS ARE REQUIRED.

3. LOADING PROCEEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE ABSOLUTE LOADER. NOTE: IF THE DIAGNOSTICS ARE ON A MEDIA SUCH AS DISK, MAGTAPE, DECTAPE, OR CASSETTE; FOLLOW INSTRUCTIONS FOR THE MONITOR WHICH HAS BEEN PROVIDED ON THAT SPECIFIC MEDIA.

ABSOLUTE LOADER STARTING ADDRESS *500

MEMORY * SIZE

4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

3.1.1 PLACE ADDRESS OF ABS LOADER INTO SWITCH REGISTER.
(ALSO PLACE 'HALT' SW UP)

3.1.2 DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE AND RELEASE.

3.1.3 DEPRESS 'START KEY' ON CONSOLE AND RELEASE (PROGRAM SHOULD NOW BE LOADING INTO CPU)

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
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171

4. STARTING PROCEEDURE

- A. SET SWITCH REGISTER TO 000200
- B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
- C. SET SWR TO ZERO FOR 'AUTO SIZING' OR LEAVE OR SET SW00=1 FOR USER INPUT FROM CONSOLE TERMINAL.
- D. DEPRESS 'START KEY' AND RELEASE, THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM OR PARAMETERS WERE CHANGED BY SW00=1) AND ALSO THE FOLLOWING:

'MAP OF DZ11 STATUS'

1500	160010
1502	000300
1504	000005
1506	000377
1510	017470
1512	000000

THE ABOVE IS ONLY AN EXAMPLE! THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD. 1500 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS TABLE SEE SECTION 8.4 FOR HELP.

THE PROGRAM WILL TYPE "RUNNING" AND PROCEED TO RUN THE DIAGNOSTIC

4.1 CONTROL SWITCH SETTINGS

NOTE: IF THERE IS NO REAL SWR (177570); SWR MAY BE MODIFIED AT LOC:176 OR BY HITTING CONTROL "G" (↑G) ON CONSOLE TERMINAL.

- SW 15 SET: HALT ON ERROR
- SW 14 SET: LOOP ON CURRENT TEST
- SW 13 SET: INHIBIT ERROR PRINT OUT
- SW 12 SET: INHIBIT **ALL** TYPE OUT/BELL ON ERROR.
- SW 11 SET: INHIBIT ITERATIONS. (QUICK PASS)
- SW 10 SET: ESCAPE TO NEXT TEST
- SW 09 SET: LOOP WITH CURRENT DATA
- SW 08 SET: CATCH ERROR AND LOOP ON IT
- SW 07 SET: NO AUTO SIZE; CLR-DO AUTO SIZE. IF 1ST START OF PROGRAM AFTER LOADING.
- SW 06 SET: RESELECT DZ11'S DESIRED ACTIVE
- SW 05 SET: RESERVED
- SW 04 SET: SELECT DELAY PARAMETER(DEFAULT = 36)
- SW 03 SET: EXTRA PARAMETER INPUT
- SW 02 SET: LOCK ON SELECTED TEST
- SW 01 SET: RESTART PROGRAM AT SELECTED TEST
- SW 00 SET: GET USERS PARAMETERS FROM CONSOLE

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 06 RESELECT DZ11'S DESIRED ACTIVE. PLEASE NOTE THAT A MESSAGE IS TYPED OUT FOR SETTING THE SWITCH REGISTER EQUAL TO DZ11'S ACTIVE. THIS MEANS IF THE SYSTEM HAS FOUR DZ11S; BITS 00,01,02,03 WILL BE SET IN LOC 'DZACTV' FROM THE SWITCH REGISTER. USING THIS SWITCH(SW06) ALTERS THAT LOCATION; THEREFORE IF FOUR DZ11S ARE IN THE SYSTEM ***DO NOT*** SET SWITCHS GREATER THAN SW 03 IN THE UP POSITION. THIS WOULD BE A FATAL ERROR. DO NOT SELECT MORE ACTIVE DZ11S THAN HAS BEEN GIVEN INFORMATION ABOUT IN PARAMETER INPUT (SW00=1)

METHOD: A: LOAD ADDRESS 200
B: START WITH SW 06=1
C: PROGRAM WILL TYPE MESSAGE
D: SET THE BINARY NUMBER OF DZ11S DESIRED ACTIVE EXAMPLE: 1=1
DZ11; 3=2 DZ11; 7=3 DZ11; 17=4 DZ11 37=5 DZ11 ETC/AA PRESS CONTINUE.
E: NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05)
F: SET WITH ANY OTHER SWITCH SETTINGS DESIRED. PRESS CONTINUE.

SW 01 RESTART PROGRAM AT SELECTED TEST IT IS STRONGLY SUGGESTED THAT AT LEAST ONE PASS HASS BEEN MADE BEFORE TRYING TO SELECT A TEST THAT IS NOT IN THE ORDER OF SEQUENCE THE REASON BEING IS THAT THE PROGRAM HAS TO CLEAR AREAS AND SET UP PARAMETERS. NOTE: IF RUNNING MULTIPLE DZ11'S; THE DZ11 YOU DESIRE TO BE UNDER TEST MUST BE SELECTED BY THE USE OF SW06 BEFORE LOCKING ON THE TEST. IN OTHER WORDS; EACH TIME THE PROGRAM IS STARTED; THE FIRST DZ11 WILL BE SELECTED TO BE UNDER TEST UNLESS SW06 IS USED TO SELECT ONLY ONE.

SW 09 LOOP ON CURRENT DATA: THIS SWITCH WILL ONLY WORK IF CALL 'SCOPI' IS IN THAT TEST. THE REASON BEING THAT MOST TESTS DEAL WITH BLOCKS OF DIFFERENT DATA TO BE SENT OR RECEIVED ALL AT ONCE THUS IN BLOCK DATA; ONE PATTERN CANN'T BE SINGLED OUT.

SW 04 SELECT DELAY PARAMETER: THIS SWITCH SHOULD BE USED WITH CARE AS TOO SHORT A DELAY WILL CAUSE VALID TESTS TO FAIL ON CERTAIN PROCESSORS. IT IS RECCOMENDED THAT THIS SWITCH ONLY BE IN CONJUNCTION WITH SCOPE LOOPS, E.G. SW 14,9,4,1 SET; SW 9,4,2,1 SET. THE SHORTEST PARAMETER IS 1; THE LONGEST ACCEPTED IS 177776.

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

4.1.3 SWITCH REGISTER PRIORITYS

ERROR SWITCHES

1. SW 12 DELETE PRINT OUT/BELL ON ERROR.
2. SW 13 DELETE ERROR PRINTOUT.
3. SW 15 HALT ON THE ERROR.
4. SW 08 GOTO BEGINNING OF THE TEST(ON ERROR).
5. SW 10 GOTO NEXT TEST(ON ERROR).

SCOPE SWITCHES

1. SW 09 (IF ENABLED BY 'SCOPI') ON AN ERROR; IF AN '*' IS PRINTED IN FRONT OF THE TEST NO. (EX. *TEST NO. 10) SW09 IS INCORPORATED IN THAT TEST AND THEREFORE SW09 IS *USUALLY* THE BEST SWITCH FOR THE SCOPE LOOP (SW14=0, SW10=0, SW09=1, SW08=0). IF SW09 IS NOT ENABELED; AND THERE IS A *HARD* ERROR (CONSTANT); SW08 IS BEST.
(SW14=1, 0, SW10=0, SW09=0, SW08=1). FOR INTERMITTENT ERRORS; SW14=1 WILL LOOP ON TEST REGARDLESS OF ERROR OR NOT ERROR.
(SW14=1, SW10=0, SW09=0, SW08=1,0)
2. SW 14
3. SW 11

4.2 STARTING ADDRESS

SA 200 - ADDRESS 200 IS FOR NORMAL EXECUTION OF THE DIAGNOSTIC. THIS WILL DO THE MAJOR TESTING NECESSARY FOR VERIFICATION OF HARDWARE.

SA 210 - CABLE/ECHO - TERMINAL TESTS. STARTING AT ADDRESS 210 WILL GIVE THE USER THE OPTION TO VERIFY THE EIA CABLES AT THE DIST PNL OR VERIFY A TRUE LINK TO ANY DEC SUPPORTED EIA TERMINAL SUPPORTED BY THE DZ11.

NOTE: IF ADDRESS 000042 IS NON-ZERO THE PROGRAM ASSUMES IT IS UNDER ACT11 OR XXDP CONTROL AND WILL ACT ACCORDINGLY AFTER *ALL* AVAILABLE DZ11'S ARE TESTED THE PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'.

5. OPERATING PROCEDURE

WHEN PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION FOUR WILL BE PRINTED AND PROGRAM WILL BEGIN RUNNING THE DIAGNOSTIC.

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

5.1 NORMAL START OF DIAGNOSTIC

ON THE FIRST START OF THE DIAGNOSTIC AT ADDRESS 200; IF AUTO SIZING IS NOT USED OR WHENEVER SW00=1; THE FOLLOWING QUESTIONS ARE ASKED AND MUST BE ANSWERED.

"1ST CSR ADDRESS (160000:163700): "

YOU MUST TYPE IN THE FIRST DZ11 CSR IN THE SYSTEM YOU WISH TESTING TO BEGIN AT. RANGE: 160000:163700

"1ST VECTOR ADDRESS (300:770): "

YOU MUST TYPE IN THE VECTOR OF THE FIRST DZ11 IN THE SYSTEM UNDER TEST. RANGE 300:770

"BR LEVEL (4:6): "

TYPE IN THE PRIORITY LEVEL OF THE DZ11 THAT THE ABOVE INFORMATION HAS BEEN GIVEN ABOUT. RANGE 4 OR 5 OR 6.

"TYPE "A" FOR EIA MODULE OR "B" FOR 20MA (A:B): "

TYPE "A" IF RUNNING A DZ11-A,B,E (EIA).
TYPE "B" IF RUNNING A DZ11-C,D,F (20MA).
TYPING A <CR> DEFAULTS TO EIA MODULES.

"MAINTAINCE MODE
[EXTERNAL <H325> (E)]
[INTERNAL <DZCSR03=1>(I)]
[STAGGERED <H327> (S)]

TYPE "E" OR "I" OR "S" DEPENDING ON WHICH MODE YOU WISH TO RUN IN. IF RUNNING "EXTERNAL"; ALL SELECTED LINES MUST BE TERMINATED BY A H325 TEST CONNECTOR.

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

"# OF DZ11'S <IN OCTAL> (1:20): "

TYPE TOTAL NUMBER OF DZ11'S TO BE TESTED IN THE SYSTEM. RANGE IS 1 THRU 20 IN OCTAL.

***** IF SW03=1 THEN *****
IF SW03=1 THE FOLLOWING WILL BE PRINTED.

"LINES ACTIVE BY BIT <IN OCTAL> (001:377):"

EACH BIT REPRESENTS A LINE AND ANY COMBINATION OF LINES MAY BE SELECTED (HOWEVER IN STAGGERED MODE TWO ADJACENT LINES MUST BE SELECTED (0-1, 2-3, 4-5, 6-7))..

"DEFAULT BAUD RATE <IN OCTAL> (00:17): "

THIS GIVES THE USER A CHANCE TO CHANGE THE DEFAULT BAUD RATE USED IN APP. 90% OF THE TEST. NORMAL OPERATION IS A "17" (19.2K) OR "16" (9.6K). "00"(50 BAUD)- NOT ADVISED.

IT IS IMPORTANT TO NOTE THAT ALL DZ11'S IN THE SYSTEM MUST BE CONTIGIOUS FOR BOTH ADDRESS AND VECTORS. ALSO ALL THE EXTRA PARAMETERS OTHER THAN CSR AND VECTORS ARE GIVEN TO THE EXISTING DZ11'S IN THE SYSTEM. IF NOT ALL DZ11'S ARE SAME PRIORITY OR IF THE MODE OF OPERATION IS DIFFERENT FOR EACH DZ11; THIS MUST BE "PATCHED" INTO THE CORRECT STATUS MAP ENTRY WHICH IS PRINTED AT START TIME. AN ALTERNATIVE IS TO PUT SW00=1 AT START TIME; ANSWER QUESTIONS ABOUT DZ11 UNDER TEST AND INDICATE ONLY 1 DZ11 IN THE SYSTEM. IF THE STATUS MAP IS TO BE "PATCHED" IT MUST BE DONE AFTER THE QUESTIONS ARE ANSWERED OR AFTER THE AUTO SIZE.

339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392

5.2 HOW TO RUN THE "CABLE/ECHO" TESTS.

NORMAL STARTING FOR THE FIRST TIME WOULD BE: LOAD ADDRESS 210; START WITH THE SWR EQUAL TO 213.

NOTE: SW00=1 ASKS FOR "VECTOR" AND "CSR"
 SW01=1 ASKS FOR "WHICH TEST ECHO OR CABLE", "BAUD RATE", "LINE" UNDER TEST. PROGRAM WILL PRINT OUT:

"VECTOR ADDRESS-"

YOU TYPE VECTOR WITH A <CR>.

"CONTROL REGISTER ADDRESS-"

YOU TYPE IN DZCSR UNDER TEST.

"WHICH TEST ? ECHO OR CABLE (E OR C)"

LETS DO THE CABLE TEST FIRST. **THIS TEST IS ONLY TO BE DONE ON THE EIA VERSION OF THE DZ11 NOT THE 20MA VERSION". TYPE "C" <CR>

"BAUD RATE- "

TYPE EITHER 50, 110, 135, 150, 300, 600, 1200 1800, 2000, 2400, 3600, 4800, 7200, 9600 FOLLOWED BY <CR>

"LINE: "

YOU TYPE THE LINE WHICH HAS THE H325 TEST CONNECTOR. (TYPE EITHER 0, 1, 2, 3, 4, 5, 6, 7) PROGRAM WILL THEN PRINT:

"CABLE TEST"

AND IF EVERYTHING IS WORKING; THE FOLLOWING WILL BE PRINTED:

"PASS DONE."

"PASS DONE."

ETC.

TO CHANGE LINES; HIT ANY PRINTING KEY ON YOUR CONSOLE TERMINAL WHILE THE PROGRAM IS RUNNING AND THE FOLLOWING WILL BE PRINTED:

"LINE: "

NOW CHANGE THE H325 TEST CONNECTOR TO ANOTHER LINE AND TYPE THE NEW LINE. PROGRAM WILL THEN PRINT:

"CABLE TEST"

"PASS DONE."

"PASS DONE."

CONTINUE THIS OPERATION UNTIL ALL LINES ARE TESTED.

393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440

5.3 ECHO TEST

IF PROGRAM HAS ALREADY BEEN STARTED AT 210 AND THE VECTOR AND ADDRESS HAVE BEEN TYPED IN: JUST LOAD ADDRESS 210 AND START WITH SWR EQUAL TO 212. PROGRAM WILL PRINT:

"WHICH TEST ? ECHO OR CABLE (E OR C)"

NOW TYPE AN "E" TO DO THE ECHO TEST. PROGRAM WILL PRINT:

"BAUD RATE--"

TYPE BAUD RATE AT WHICH THE TERMINAL IS SET THAT IS CONNECTED TO THE DZ11 DIST PNL. PROGRAM WILL PRINT:

LINE: "

TYPE THE LINE THE TERMINAL IS CONNECTED TO AT THE DIST PNL THEN THE PROGRAM WILL PRINT:

"TERMINAL ECHO TEST"

*** AT THIS POINT THE MESSAGE:

"THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789"

SHOULD BE PRINTED ON THE TERMINAL CONNECTED TO THE DZ11. IF THIS MESSAGE IS DESIRED TO BE CONTINUOUSLY OUTPUT; SET THE SWR TO 377 (SWR=377) WHILE IT IS BEING OUTPUT OR WHEN PROGRAM IS STARTED AT 210. WHEN THIS MESSAGE IS DONE AND THE SWR IS NOT EQUAL TO 377; THE CONSOLE WILL PRINT:

"TYPE A CHAR. ON DZ11 TERMINAL"

ANY PRINTABLE CHAR HIT ON DZ11 TERMINAL SHOULD BE ECHOED BACK ON THE TERMINAL. **IF YOU HIT CNTRL C (^C) ON THE DZ11 TERMINAL THE PROGRAM WILL PRINT:

"PASS DONE."

ON THE CONSOLE TERMINAL AND THE "QUICK BROWN FOX" WILL BE PRINTED ON DZ11 TERMINAL AGAIN AND THE ECHO TEST WILL BE RUNNING. TO CHANGE LINES; DO LIKE CABLE TEST. HIT PRINTABLE KEY ON CONSOLE TERMINAL. AND CHANGE THE LINE ON WHICH THE TERMINAL IS CONNECTED. AND ENTER THE NEW LINE TO THE PROGRAM.

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

5.4 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

1. HALT ON ERROR (VIA SW 15=1) WHEN EVER AN ERROR OCCURS.
2. CLEAR SW 15.
3. SET SW 14: (LOOP ON THIS TEST)
4. SET SW 13: (INHIBIT ERROR PRINT OUT)

THE TEST NUMBER AND PC WILL BE TYPED OUT AND POSSIBLY AN ERROR MESSAGE (THIS DEPENDS ON THE TEST) TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE OF THE PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT; LOOK IN THE LISTING FOR THAT TEST NUMBER WHICH WAS TYPED OUT AND THEN NOTE THE PC OF THE ERROR REPORT THIS WAY THE EXACT FUNCTIONING OF THE TEST CAN BE INTERPETED.

6. ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC TYPED OUT AT THE TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 12=0). IN MOST CASES ADDITIONAL INFORMATION WILL BE SUPPLIED TO THE THE ERROR MESSAGE WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.

6.2 ERROR RECOVERY

IF FOR SOME REASON THE DZ11 SHOULD 'HANG THE BUS' (GAIN CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU. IF THIS SHOULD HAPPEN; LOOK IN LOCATION 'TSTNO' (ADDRESS 1216) FOR THE NUMBER OF THE TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR. IN THIS WAY THE OPERATOR WILL HAVE AN IDEA AS TO WHAT THE DZ11 WAS DOING AT THE TIME OF THE ERROR.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)
STATUS TABLE SHOULD BE VERIFIED REGARDLESS OF HOW PROGRAM WAS STARTED. ALSO IT IS IMPORTANT TO USE THIS LISTING ALONG WITH THE INFORMATION PRINTED ON THE TTY TO COMPLETLY ISOLATE PROBLEMS.

486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515

7.2 OPERATING RESTRICTIONS

PARAMETER MUST BE INPUT FROM USER OR APT IF "AUTO SIZING" IS NOT USED.

8. MISCELLANEOUS

8.1 EXECUTION TIME

ALL DZ11 DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS' MESSAGE (PROVIDING NO ERRORS AND SW12=0) WITHIN 2 MIN. THIS IS ASSUMING SW11=2 (DELETE ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION. THE ACTUAL EXECUTION TIME DEPENDS GREATLY ON THE PDP11 CPU CONFIGURATION.

8.2 PASS COMPLETE

NOTE: *EVERY* TIME THE PROGRAM IS STARTED; THE TESTS WILL RUN AS IF SW11 (DELETE ITERATIONS) WAS UP (=1). THIS IS TO 'VERIFY NO *HARD* ERRORS' AS SOON AS POSSIBLE. THEREFORE THE FIRST PASS -EACH TIME PROGRAM IS STARTED- WILL BE A 'QUICK PASS' UNTIL ALL DZ11'S IN SYSTEM ARE TESTED. WHEN THE DIAGNOSTIC HAS COMPLETED A PASS THE FOLLOWING IS AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.

END PASS DZDZA-B CSR: 160010 VEC: 300 PASSES: 000001 ERRORS: 000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE NOT NECESSARILY THE VALUES FOR THE DEVICE. THEY ARE ONLY FOR THIS EXAMPLE.

516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544

8.4 KEY LOCATIONS

\$LPADR (1126)	CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS ASSERTED.
NEXT (1360)	CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PERFORMED.
\$STNM (1122)	CONTAINS THE NUMBER OF THE TEST NOW BEING PERFORMED.
RUN (1406)	THE BIT IN 'RUN' ALWAYS POINTS ONE PAST THE DZ11 CURRENTLY BEING TESTED. EXAMPLE: (RUN) 1304/0000000001000000 MEANS THAT DZ11 NO.05 IS THE DZ11 NOW RUNNING.
STATUS MAP (1500)-(2000)	THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 16 (DECIMAL) DZ11S SEQUENTIALY. THEY CONTAIN THE CSR, VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DZ11.
DZACTV (1404)	EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DZ11 WILL BE TESTED IN TURN. EXAMPLE: (DZACTV) 1300/0000000000011111 MEANS THAT DZ11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DZACTV) 1300/0000000000010001 MEANS THAT DZ11 NO. 00,04 WILL BE TESTED.
\$BASE (1310)	CONTAINS THE RECEIVER CSR OF THE CURRENT DZ11 UNDER TEST.

545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586

B.4A MORE ON THAT 'STATUS TABLE' (1500-2000)

'MAP OF DZ11 STATUS'

1500	160010
1502	000300
1504	000005
1506	000377
1510	017470
1512	000000

THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 8 DZ11'S IN THE SYSTEM (THESE WILL FOLLOW UNDER THIS TABLE). EXPLANATION:

- 1500 160010 THIS IS THE SYSTEM CONTROL REGISTER FOR THE 1ST DZ11 IN THE SYSTEM.
- 1502 000300 THIS IS VECTOR 'A' FOR THE FIRST DZ11 IN THE SYSTEM.
- 1504 000005 THIS REPRESENTS THE BUS INTERRUPT PRIORITY LEVEL OF THE DZ11. BIT15 OF THIS LOCATION INDICATES EITHER EIA OR 20MA. IF BIT15=0 MODULE SHOULD BE EIA; IF BIT15=1 MODULE SHOULD BE 20MA.
- 1506 000377 THIS IS THE BINARY REPRESENTATION OF WHAT LINES ARE TO BE TESTED.
- 1510 017470 THIS IS THE PARAMETER LOCATION USED IS MOST OF THE TESTS. IT INDICATED PARAMETERS OF: RX ON, SPEED SELECT 17 (19.2K BAUD) EIGHT BITS PER CHAR, AND TWO STOP BITS. THE USER MAY ALTER THE STOP BITS AND THE SPEED, BUT THE REMAINING PARAMETERS SHOULD BE LEFT ALONE.
- 1512 000000 THIS LOCATION WILL CONTAIN EITHER ALL ZEROS INDICATING THAT INTERNAL LOOP WAS SELECTED AS MODE OF OPERATION OR IT WILL CONTAIN 10000 INDICATING THAT "STAGGERED MODE" WAS SELECTED OR IT WILL CONTAIN 000200 INDICATING THAT "EXTERNAL" WAS THE MODE SELECTED.

THE ABOVE IS REPEATED FOR EACH DZ11 IN THE SYSTEM. THE TABLE IS FILLED BY AUTO SIZING OR BY THE MANUAL PARAMETER INPUT PROGRAM AS DESCRIBED PREVIOUSLY. ALSO IF DESIRED BY USER; THE LOCATIONS MAY BE ALTERED BY HAND (TOGGLED IN) TO SUIT THE SPECIFIC CONFIGURATION.

587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

THE PROGRAM WILL START AT ADDRESS 160000 AND START 'REFERENCEING' THE ADDRESS IN THE POINTER. IF A NON-EX MEMORY TRAP OCCURS, THE POINTER (HOLDING 160000) IS UPDATED BY 10 AND THE ABOVE IS REPEATED UNTIL ADDRESS 163700 IS REACHED. IF A 'SLAVE SYNC RESPONSE' WAS ISSUED BY THE DZ11 (OR ANY OTHER DEVICE) (NO NXM TRAP), "MASTER SCAN ENABLE" IS ATTEMPTED TO BE SET AND THE "TCR" BIT FOR LINE 7 IS SET. "TRDY" IS THEN TESTED TO BE SET AND BOTH "TCR07" AND "MASTER SCAN ENABLE" ARE TESTED TO BE STILL SET. IF ALL OF THIS WORKED, THEN A "DEVICE CLEAR" IS ISSUED TESTING THAT THE BIT CAN BE READ BACK AND THAT AFTER SOME TIME IT SELF CLEARS. IF ALL OF THE ABOVE WORKED, THIS DEVICE IS ASSUMED TO BE A DZ11. IF ANY OF THE ABOVE FAILED, UPDATING OF THE POINTER IS DONE AND THE SEQUENCE IS REPEATED.

NOTE: IF THE PROGRAM DOES NOT FIND YOUR DZ11; SOMETHING IS WRONG AND AUTO SIZING SHOULD NOT BE DONE.

8.5.2 FINDING THE VECTOR

THE VECTOR AREA (ADDRESS 300-776) IS FILLED WITH THE INSTRUCTION IOT AND '+2' (NEXT ADDRESS). BIT14 AND BITS (TX INTERRUPT ENABLE AND MSTSCAN ENABLE) ARE SET INTO THE DZCSR. "TCR07" IS THEN SET. A DELAY IS MADE AND IF NO INTERRUPT OCCURS (BECAUSE OF A BAD DZ11) THE PROGRAM ASSUMES VECTOR ADDRESS 300 AND THE PROBLEM SHOULD BE FIXED IN THE DIAGNOSTIC. ONCE THE PROBLEM IS FIXED, THE PROGRAM SHOULD BE RE-SETUP AGAIN TO GET CORRECT VECTOR. IF AN INTERRUPT OCCURRED, THE ADDRESS TO WHICH THE DZ11 INTERRUPTED TO IS PICKED UP AND REPORTED AS THE VECTOR. NOTE: IF THE VECTOR REPORTED IS NOT THE VECTOR SET UP BY YOU; THERE IS A PROBLEM AND AUTO SIZING SHOULD NOT BE DONE.

8.5.3 PARAMETER ASSUMPTIONS.

SINCE TOO MUCH HARDWARE WOULD NEED TO BE TURNED ON TO SIZE THE REST OF THE PARAMETERS; THE PROGRAM MUST ASSUME THE REMAINING VARIATIONS. THE RESULT IF NOT TO YOUR SPECIFIC CONFIGURATION MAY BE ALTERED BY HAND (TOGGLE IN) IF DESIRED. IN THIS WAY 95% OF THE PARAMETER SETUP WAS DONE BY THE PROGRAM AND 5% BY YOU.

THEREFORE:

- 1) BUS PRIORITY IS SET TO LEVELS.
- 2) ALL EIGHT LINES ARE ASSUMED TO BE TESTED.
- 3) DEFAULT BAUD RATE IS SET TO 17 (19.2 K).
- 4) MODE OF OPERATION IS "INTERNAL MODE".
- 5) MODULE IS ASSUMED TO BE "EIA" VERSION.
SET BIT 15 IN PRIORITY ENTRY OF MAP IF YOU HAVE A 20MA MODULE.

IN ALL ADJUSTMENTS PLEASE REFER TO SECTION 8.4A FOR GREATER DETAIL.

.TITLE MD-11-DZDZA-B
: *COPYRIGHT (C) 1976
: *DIGITAL EQUIPMENT CORP.
: *MAYNARD, MASS. 01754

643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698

000001

001120

000011
000012
000015
000200
177776

177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007

000000
000040
000100
000140
000200
000240
000300
000340

```

;*
;*PROGRAM BY JERRYL PAYNE,JOHN EGOLF
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-CO),MAR 21, 1976.
;*
$TN=1
:STARTING PROCEDURE
:LOAD PROGRAM
:LOAD ADDRESS 000200
:PRESS START
:PROGRAM WILL TYPE "MAINDEC-11-DZDZAB/<200>/EIGHT LINE ASYNC MUX TESTS"
:PROGRAM WILL TYPE "RUNNING" TO INDICATE THAT TESTING HAS STARTED
:AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
:AND THEN RESUME TESTING
  
```

.SBTTL BASIC DEFINITIONS

;*INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***

```

STACK= 1120
.EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
  
```

;*MISCELLANEOUS DEFINITIONS

```

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

;*GENERAL PURPOSE REGISTER DEFINITIONS

```

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

;*PRIORITY LEVEL DEFINITIONS

```

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

MD-11-DZDZA-B
DZDZAB.P11MACY11 27(732) 03-NOV-76 15:03 PAGE 18
BASIC DEFINITIONS

```

699
700
701      100000
702      040000
703      020000
704      010000
705      004000
706      002000
707      001000
708      000400
709      000200
710      000100
711      000040
712      000020
713      000010
714      000004
715      000002
716      000001
717
718
719
720
721
722
723
724
725
726
727
728
729      100000
730      040000
731      020000
732      010000
733      004000
734      002000
735      001000
736      000400
737      000200
738      000100
739      000040
740      000020
741      000010
742      000004
743      000002
744      000001
745
746
747
748
749
750
751
752
753
754

```

```

.*"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
.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1
.EQUIV BIT00,BIT0

```

```

755
756
757      000004      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
758      000010      ERRVEC= 4      ;: TIME OUT AND OTHER ERRORS
759      000014      RESVEC= 10     ;: RESERVED AND ILLEGAL INSTRUCTIONS
760      000014      TBITVEC=14    ;: "T" BIT
761      000014      TRTVEC= 14    ;: TRACE TRAP
762      000020      BPTVEC= 14    ;: BREAKPOINT TRAP (BPT)
763      000024      IOTVEC= 20    ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
764      000030      PWRVEC= 24    ;: POWER FAIL
765      000034      EMTVEC= 30    ;: EMULATOR TRAP (EMT) **ERROR**
766      000060      TRAPVEC=34    ;: "TRAP" TRAP
767      000064      TKVEC= 60     ;: TTY KEYBOARD VECTOR
768      000064      TPVEC= 64     ;: TTY PRINTER VECTOR
769      000240      PIRQVEC=240   ;: PROGRAM INTERRUPT REQUEST VECTOR

```

; INSTRUCTION DEFINITIONS

```

770
771 -----
772
773
774      005746      PUSH1SP=5746  ;: DECREMENT PROCESSOR STACK 1 WORD
775      005726      POP1SP=5726   ;: INCREMENT PROCESSOR STACK 1 WORD
776      010046      PUSHRO=10046   ;: SAVE RO ON STACK
777      012600      POPRO=12600    ;: RESTORE RO FROM STACK
778      024646      PUSH2SP=24646 ;: DECREMENT STACK TWICE
779      022626      POP2SP=22626  ;: INCREMENT STACK TWICE

```

; DZ11 CONTROL AND STATUS REGISTER DEFINITIONS
(DZCSR) BIT DEFINITIONS

```

780
781 -----
782
783
784
785      000010      MAINT = BIT3   ;: MAINTENANCE MODE ENABLE
786      000020      DCLR=BIT4     ;: DEVICE CLEAR
787      000040      MSENAB=BITS   ;: MASTER SCAN ENABLE
788      000100      RIE=BIT6      ;: RECEIVER INTERRUPT ENABLE
789      000200      RDONE=BIT7    ;: RECEIVER DONE
790      010000      SILOEN= BIT12  ;: SILO ALARM ENABLE
791      020000      SILOAL = BIT13 ;: SILO ALARM
792      040000      TIE=BIT14     ;: TRANSMITTER INTERRUPT ENABLE
793      100000      TRDY=BIT15    ;: TRANSMITTER READY

```

; DZCSR WORD DEFINITIONS

```

794
795 -----
796
797      000000      TL0=0          ;: TRANSMIT LINE 0
798      000400      TL1=BIT8      ;: TRANSMIT LINE 1
799      001000      TL2=BIT9      ;: TRANSMIT LINE 2
800      001400      TL3=BIT9!BIT8 ;: TRANSMIT LINE 3
801      002000      TL4=BIT10     ;: TRANSMIT LINE 4
802      002400      TL5=BIT10!BIT8 ;: TRANSMIT LINE 5
803      003000      TL6=BIT10!BIT9 ;: TRANSMIT LINE 6
804      003400      TL7=BIT10!BIT9!BIT8 ;: TRANSMIT LINE 7

```

; DZRBUF BIT DEFINITIONS

```

805
806 -----
807
808
809
810      010000      PARER=BIT12    ;: PARITY ERROR

```

811	020000	FRMERR=BIT13	:FRAME ERROR
812	040000	OVRRUN=BIT14	:OVERRUN ERROR
813	100000	DVALID=BIT15	:DATA VALID
814			
815			
816			
817			
818	000000	RL0=0	:RECEIVER LINE 0
819	000400	RL1=BIT8	:RECEIVER LINE 1
820	001000	RL2=BIT9	:RECEIVER LINE 2
821	001400	RL3=BIT9!BIT8	:RECEIVER LINE 3
822	002000	RL4=BIT10	:RECEIVER LINE 4
823	002400	RL5=BIT10!BIT8	:RECEIVER LINE 5
824	003000	RL6=BIT10!BIT9	:RECEIVER LINE 6
825	003400	RL7=BIT10!BIT9!BIT8	:RECEIVER LINE 7
826			
827			
828			
829			
830	000000	LP0=0	:LINE PARAMETER 0
831	000001	LP1=BIT0	:LINE PARAMETER 1
832	000002	LP2=BIT1	:LINE PARAMETER 2
833	000003	LP3=BIT1!BIT0	:LINE PARAMETER 3
834	000004	LP4=BIT2	:LINE PARAMETER 4
835	000005	LP5=BIT2!BIT0	:LINE PARAMETER 5
836	000006	LP6=BIT2!BIT1	:LINE PARAMETER 6
837	000007	LP7=BIT2!BIT1!BIT0	:LINE PARAMETER 7
838			
839	000000	FIVE=0	:FIVE BITS/CHAR, 1 STOP BIT
840	000010	SIX=BIT3	:SIX BITS/CHAR, 1 STOP BIT
841	000020	SEVEN=BIT4	:SEVEN BITS/CHAR, 1 STOP BIT
842	000030	EIGHT=BIT4!BIT3	:EIGHT BITS/CHAR, 1 STOP BIT
843	000040	FIVES=BIT5	:FIVE BITS/CHAR, 2 STOP BITS
844	000050	SIXS=BIT5!BIT3	:SIX BITS/CHAR, 2 STOP BITS
845	000060	SEVENS=BIT5!BIT4	:SEVEN BITS/CHAR, 2 STOP BITS
846	000070	EIGHTS=BIT5!BIT4!BIT3	:EIGHT BITS/CHAR, 2 STOP BITS
847			
848	000100	PARITY=BIT6	:PARITY ENABLED
849	000200	ODDPAR=BIT7	:ODD PARITY ENABLED
850	000000	ONESTOP=0	:ONE STOP BIT ENABLED
851	000040	TWOSTOP=BIT5	:TWO STOP BITS ENABLED
852	000000	EVEPAR=0	:EVEN PARITY ENABLED
853	010000	RCVON=BIT12	:ENABLE RECEIVER (RECEIVER ON)
854			
855	000000	S50=0	:SPEED 50 BAUD
856	000400	S75=BIT8	:SPEED 75 BAUD
857	001000	S110=BIT9	:SPEED 110 BAUD
858	001400	S134=BIT9!BIT8	:SPEED 134.5 BAUD
859	002000	S150=BIT10	:SPEED 150 BAUD
860	002400	S300=BIT10!BIT8	:SPEED 300 BAUD
861	003000	S600=BIT10!BIT9	:SPEED 600 BAUD
862	003400	S1200=BIT10!BIT9!BIT8	:SPEED 1200 BAUD
863	004000	S1800=BIT11	:SPEED 1800 BAUD
864	004400	S2000=BIT11!BIT8	:SPEED 2000 BAUD
865	005000	S2400=BIT11!BIT9	:SPEED 2400 BAUD
866	005400	S3600=BIT11!BIT9!BIT8	:SPEED 3600 BAUD

867	006000	S4800=BIT11!BIT10	:SPEED 4800 BAUD
868	006400	S7200=BIT11!BIT10!BIT8	:SPEED 7200 BAUD
869	007000	S9600=BIT11!BIT10!BIT9	:SPEED 9600 BAUD
870	007400	S19200=BIT11!BIT10!BIT9!BIT8	:SPEED 19200 BAUD

;DZTCR BIT DEFINITIONS

873			
874	000001	TCR0=BIT0	:TCR0
875	000002	TCR1=BIT1	:TCR1
876	000004	TCR2=BIT2	:TCR2
877	000010	TCR3=BIT3	:TCR3
878	000020	TCR4=BIT4	:TCR4
879	000040	TCR5=BIT5	:TCR5
880	000100	TCR6=BIT6	:TCR6
881	000200	TCR7=BIT7	:TCR7
882	000400	DTR0=BIT8	:DTR0
883	001000	DTR1=BIT9	:DTR1
884	002000	DTR2=BIT10	:DTR2
885	004000	DTR3=BIT11	:DTR3
886	010000	DTR4=BIT12	:DTR4
887	020000	DTR5=BIT13	:DTR5
888	040000	DTR6=BIT14	:DTR6
889	100000	DTR7=BIT15	:DTR7

;DZMSR BIT DEFINITIONS

892			
893	000001	RING0=BIT0	:RING INDICATED ON LINE 0
894	000002	RING1=BIT1	:RING INDICATED ON LINE 1
895	000004	RING2=BIT2	:RING INDICATED ON LINE 2
896	000010	RING3=BIT3	:RING INDICATED ON LINE 3
897	000020	RING4=BIT4	:RING INDICATED ON LINE 4
898	000040	RING5=BIT5	:RING INDICATED ON LINE 5
899	000100	RING6=BIT6	:RING INDICATED ON LINE 6
900	000200	RING7=BIT7	:RING INDICATED ON LINE 7
901	000400	C00=BIT8	:CARRIER PRESENT ON LINE 0
902	001000	C01=BIT9	:CARRIER PRESENT ON LINE 1
903	002000	C02=BIT10	:CARRIER PRESENT ON LINE 2
904	004000	C03=BIT11	:CARRIER PRESENT ON LINE 3
905	010000	C04=BIT12	:CARRIER PRESENT ON LINE 4
906	020000	C05=BIT13	:CARRIER PRESENT ON LINE 5
907	040000	C06=BIT14	:CARRIER PRESENT ON LINE 6
908	100000	C07=BIT15	:CARRIER PRESENT ON LINE 7

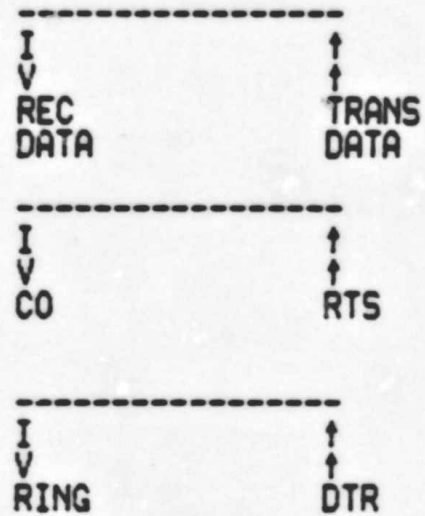
;DZTDR BIT DEFINITIONS

912			
913	000400	BRK0=BIT8	:BREAK FOR LINE 0
914	001000	BRK1=BIT9	:BREAK FOR LINE 1
915	002000	BRK2=BIT10	:BREAK FOR LINE 2
916	004000	BRK3=BIT11	:BREAK FOR LINE 3
917	010000	BRK4=BIT12	:BREAK FOR LINE 4
918	020000	BRK5=BIT13	:BREAK FOR LINE 5
919	040000	BRK6=BIT14	:BREAK FOR LINE 6
920	100000	BRK7=BIT15	:BREAK FOR LINE 7

921
922

923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940

TABLE OF LOOP AROUND FUNCTIONS (H325)



```

941      ;*****
942      ;-----
943      ; TRAPCATCHER FOR ILLEGAL INTERRUPTS
944      ; THE STANDARD "TRAP CATCHER" IS PLACED
945      ; BETWEEN ADDRESS 0 TO ADDRESS 776.
946      ; IT LOOKS LIKE "PC+2 HALT".
947      ;-----
948      ;*****
949
950      000000      .=0
951      ; STANDARD INTERRUPT VECTORS
952      ;-----
953
954      000010      .=10
955      000010      010620      SET.PS      ; FAKE "MTPS" INSTRUCTION TRAP
956      000012      000340      PR7          ; MAKE SURE PS IS PRIORITY 7
957
958      000020      .=20
959      000020      004642      .SCOPE      ; SCOPE LOOP HANDLER
960      000022      000340      PR7          ; HANDLE AT PRIORITY 7
961      000024      007500      $PWRDN     ; POWER FAIL HANDLER
962      000026      000340      340         ; SERVICE AT PRIORITY LEVEL 7
963      000030      006570      $ERROR    ; ERROR HANDLER
964      000032      000340      340         ; SERVICE AT PRIORITY LEVEL 7
965      000034      006462      .TRPSRV    ; GENERAL HANDLER DISPATCH SERVICE
966      000036      000340      340         ; SERVICE AT PRIORITY LEVEL 7
967
968      .SBTTL      ACT11 HOOKS
969
970      ;*****
971      ; HOOKS REQUIRED BY ACT11
972      ;-----
973      000040      $SVPC=.      ; SAVE PC
974      000046      .=46        ; ;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
975      000052      .SENDAD     ; ;2)SET LOC.52 TO ZERO
976      000052      .WORD 0     ; ; RESTORE PC
977      000052      .=$SVPC
978
979      000174      .=174
980      000176      000000      DISPREG:0 ; SOFTWARE DISPLAY REGISTER FOR SWITCHLESS 11S
981      000176      000000      SWREG: 0    ; SOFTWARE SWITCH REGISTER FOR SWITCHLESS 11S
982      000200      000137      002150      .=200      JMP      .START ; GO TO START OF PROGRAM
983      000210      000210      023114      .=210      JMP      XSTART  ; GOTO CABLE TEST/ECHO TEST
984
985
986
987      001000      001000      .=1000
988      001000      005200      040515      047111      MTITLE: .ASCIZ <200><12>/MAINDEC-11-DZDZAB/<200>/EIGHT LINE ASYNC MUX TESTS/<200>
(2)

```

```

989
990
991
992
993
994
995
996 001120
997 001120 000000
998 001122 000
999 001123 000
1000 001124 000000
1001 001126 000000
1002 001130 000000
1003 001132 000000
1004 001134 000
1005 001135 001
1006 001136 000000
1007 001140 000000
1008 001142 000000
1009 001144 000000
1010 001146 000000
1011 001150 000000
1012 001152 000000
1013 001154 000
1014 001155 000
1015 001156 000000
1016 001160 177570
1017 001162 177570
1018 001164 177560
1019 001166 177562
1020 001170 177564
1021 001172 177566
1022 001174 000
1023 001175 002
1024 001176 012
1025 001177 000
1026 001200 000000
1027
1028 001202 000000
1029 001204 000000
1030 001206 000000
1031 001210 000000
1032 001212 000000
1033 001214 000000
1034 001216 000000
1035 001220 000000
1036 001222 000000
1037 001224 000000
1038 001226 000000
1039 001230 077
1040 001231 015
1041 001232 000012
1042
1043
1044

```

.SBTTL COMMON TAGS

```

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

```

SCMTAG: .-1120

;; START OF COMMON TAGS

```

.SWORD 0
STSTNM: .BYTE 0
SERFLG: .BYTE 0
SICNT: .WORD 0
SLPADR: .WORD 0
SLPERR: .WORD 0
SERTTL: .WORD 0
SITEMB: .BYTE 0
SERMAX: .BYTE 1
SERRPC: .WORD 0
SGDADR: .WORD 0
SBDADR: .WORD 0
SGDDAT: .WORD 0
SBDDAT: .WORD 0
SAUTOB: .BYTE 0
SINTAG: .BYTE 0
SWR: .WORD DSWR
DISPLAY: .WORD DDISP
STKS: 177560
STKB: 177562
STPS: 177564
STPB: 177566
SNULL: .BYTE 0
SFILLS: .BYTE 2
SFILLC: .BYTE 12
STPFLG: .BYTE 0
SREGAD: .WORD 0
SREG0: .WORD 0
SREG1: .WORD 0
SREG2: .WORD 0
SREG3: .WORD 0
SREG4: .WORD 0
SREG5: .WORD 0
STMP0: .WORD 0
STMP1: .WORD 0
STMP2: .WORD 0
STMP3: .WORD 0
STIMES: 0
SQUES: .ASCII /?/
SCRLF: .ASCII <15>
SLF: .ASCIZ <12>

```

```

CONTAINS THE TEST NUMBER
CONTAINS ERROR FLAG
CONTAINS SUBTEST ITERATION COUNT
CONTAINS SCOPE LOOP ADDRESS
CONTAINS SCOPE RETURN FOR ERRORS
CONTAINS TOTAL ERRORS DETECTED
CONTAINS ITEM CONTROL BYTE
CONTAINS MAX. ERRORS PER TEST
CONTAINS PC OF LAST ERROR INSTRUCTION
CONTAINS ADDRESS OF 'GOOD' DATA
CONTAINS ADDRESS OF 'BAD' DATA
CONTAINS 'GOOD' DATA
CONTAINS 'BAD' DATA
RESERVED--NOT TO BE USED
AUTOMATIC MODE INDICATOR
INTERRUPT MODE INDICATOR
ADDRESS OF SWITCH REGISTER
ADDRESS OF DISPLAY REGISTER
TTY KBD STATUS
TTY KBD BUFFER
TTY PRINTER STATUS REG. ADDRESS
TTY PRINTER BUFFER REG. ADDRESS
CONTAINS NULL CHARACTER FOR FILLS
CONTAINS # OF FILLER CHARACTERS REQUIRED
INSERT FILL CHARS. AFTER A "LINE FEED"
"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
CONTAINS THE ADDRESS FROM WHICH ($REG0) WAS OBTAINED
CONTAINS (($REGAD)+0)
CONTAINS (($REGAD)+2)
CONTAINS (($REGAD)+4)
CONTAINS (($REGAD)+6)
CONTAINS (($REGAD)+10)
CONTAINS (($REGAD)+12)
USER DEFINED
USER DEFINED
USER DEFINED
USER DEFINED
MAX. NUMBER OF ITERATIONS
QUESTION MARK
CARRIAGE RETURN
LINE FEED

```

```

*****
.SBTTL APT MAILBOX-ETABLE

```


Line No.	Address	Value	Field Name	Field Type	Description
1045			;	*****	
1046			.EVEN		
1047	001234		\$MAIL:	.WORD	APT MAILBOX
1048	001234	000000	\$MSGTY:	.WORD	MESSAGE TYPE CODE
1049	001236	000000	\$FATAL:	.WORD	FATAL ERROR NUMBER
1050	001240	000000	\$TESTN:	.WORD	TEST NUMBER
1051	001242	000000	\$PASS:	.WORD	PASS COUNT
1052	001244	000000	\$DEVCT:	.WORD	DEVICE COUNT
1053	001246	000000	\$UNIT:	.WORD	I/O UNIT NUMBER
1054	001250	000000	\$MSGAD:	.WORD	MESSAGE ADDRESS
1055	001252	000000	\$MSGLG:	.WORD	MESSAGE LENGTH
1056	001254		\$ETABLE:		APT ENVIRONMENT TABLE
1057	001254	000	\$ENV:	.BYTE	ENVIRONMENT BYTE
1058	001255	000	\$ENVM:	.BYTE	ENVIRONMENT MODE BITS
1059	001256	000000	\$SWREG:	.WORD	APT SWITCH REGISTER
1060	001260	000000	\$USWR:	.WORD	USER SWITCHES
1061	001262	000000	\$CPUOP:	.WORD	CPU TYPE, OPTIONS
1062			;	*	BITS 15-11=CPU TYPE
1063			;	*	11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
1064			;	*	11/70=06, PDQ=07, Q=10
1065			;	*	BIT 10=REAL TIME CLOCK
1066			;	*	BIT 9=FLOATING POINT PROCESSOR
1067			;	*	BIT 8=MEMORY MANAGEMENT
1068	001264	000	\$MAMS1:	.BYTE	HIGH ADDRESS, M.S. BYTE
1069	001265	000	\$MTYP1:	.BYTE	MEM. TYPE, BLK#1
1070			;	*	MEM. TYPE BYTE -- (HIGH BYTE)
1071			;	*	900 NSEC CORE=001
1072			;	*	300 NSEC BIPOLAR=002
1073			;	*	500 NSEC MOS=003
1074	001266	000000	\$MADR1:	.WORD	HIGH ADDRESS, BLK#1
1075			;	*	MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
1076	001270	000	\$MAMS2:	.BYTE	HIGH ADDRESS, M.S. BYTE
1077	001271	000	\$MTYP2:	.BYTE	MEM. TYPE, BLK#2
1078	001272	000000	\$MADR2:	.WORD	MEM. LAST ADDRESS, BLK#2
1079	001274	000	\$MAMS3:	.BYTE	HIGH ADDRESS, M.S. BYTE
1080	001275	000	\$MTYP3:	.BYTE	MEM. TYPE, BLK#3
1081	001276	000000	\$MADR3:	.WORD	MEM. LAST ADDRESS, BLK#3
1082	001300	000	\$MAMS4:	.BYTE	HIGH ADDRESS, M.S. BYTE
1083	001301	000	\$MTYP4:	.BYTE	MEM. TYPE, BLK#4
1084	001302	000000	\$MADR4:	.WORD	MEM. LAST ADDRESS, BLK#4
1085	001304	000000	\$VECT1:	.WORD	INTERRUPT VECTOR#1, BUS PRIORITY#1
1086	001306	000000	\$VECT2:	.WORD	INTERRUPT VECTOR#2, BUS PRIORITY#2
1087	001310	160010	\$BASE:	.WORD	BASE ADDRESS OF EQUIPMENT UNDER TEST
1088	001312	000000	\$DEVN:	.WORD	DEVICE MAP
1089	001314	000000	\$CDW1:	.WORD	CONTROLLER DESCRIPTION WORD#1
1090	001316	000000	\$CDW2:	.WORD	CONTROLLER DESCRIPTION WORD#2
1091	001320	000000	\$DDW0:	.WORD	DEVICE DESCRIPTOR WORD#0
1092	001322	000000	\$DDW1:	.WORD	DEVICE DESCRIPTOR WORD#1
1093	001324	000000	\$DDW2:	.WORD	DEVICE DESCRIPTOR WORD#2
1094	001326	000000	\$DDW3:	.WORD	DEVICE DESCRIPTOR WORD#3
1095	001330	000000	\$DDW4:	.WORD	DEVICE DESCRIPTOR WORD#4
1096	001332	000000	\$DDW5:	.WORD	DEVICE DESCRIPTOR WORD#5
1097	001334	000000	\$DDW6:	.WORD	DEVICE DESCRIPTOR WORD#6
1098	001336	000000	\$DDW7:	.WORD	DEVICE DESCRIPTOR WORD#7
1099	001340	000000	\$DDW8:	.WORD	DEVICE DESCRIPTOR WORD#8
1100	001342	000000	\$DDW9:	.WORD	DEVICE DESCRIPTOR WORD#9


```

1111      .SBTTL  ERROR POINTER TABLE
1112
1113      ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1114      ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1115      ;*LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1116      ;*NOTE1:      IF SITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1117      ;*NOTE2:      EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1118
1119      ;*      EM      ;;POINTS TO THE ERROR MESSAGE
1120      ;*      DH      ;;POINTS TO THE DATA HEADER
1121      ;*      DT      ;;POINTS TO THE DATA
1122      ;*      DF      ;;POINTS TO THE DATA FORMAT
1123
1124
1125      001360      SERRTB:
1126
1127      ;PROGRAM CONTROL PARAMETERS
1128      ;-----
1129
1130      001360      000000      NEXT:      0      ;ADDRESS OF NEXT TEST TO BE EXECUTED
1131      001362      000000      LOCK:      0      ;ADDRESS FOR LOCK ON CURRENT DATA
1132
1133      ;PROGRAM VARIABLES
1134      ;-----
1135
1136      001364      000377      LINE:      377      ;DEFAULT ALL EIGHT LINES RUNNING
1137      001366      017470      PAR:      17470      ;PARAMETERS: 8 BITS/CHAR, 2 STOP BITS, 19200 BAUD, NO PARIT
1138      001370      000000      MODE:      0      ;DEFAULT MAINTENANCE MODE
1139      001372      000000      SAVLIN: 0      ;LINE NUMBER
1140      001374      000000      XMTLIN: 0      ;TRANSMISSION LINE NUMBER
1141      001376      000000      XMTCNT: 0      ;COUNT OF WORDS IN A TRANSMISSION PATTERN
1142      001400      000000      REGIST: 0      ;DEVICE ADDRESS STORAGE LOCATION
1143      001402      000000      SAVPC: 0      ;PROGRAM COUNTER STORAGE
1144      001404      000001      DZACTV: .BLKW 1      ;*DZ11'S SELECTED ACTIVE.
1145      001406      000001      RUN:      1      ;*POINTER ONE PAST RUNNING DEVICE.
1146      001410      000001      DZNUM:  .BLKB 1      ;*OCTAL NUMBER OF DZ11'S.
1147      001411      001      SAVNUM: .BYTE 1      ;*WORKABLE NUMBER.
1148
1149      001412      001500      .EVEN
      ACTIVE: DZ.MAP      ;TABLE POINTER.

```

```

1150
1151
1152
1153
1154 001414 000
1155 001415 000
1156 001416 000
1157 001417 000
1158 001420 000
1159 001422
1160
1161 001422 000000
1162 001424 000000
1163 001426 000000
1164 001430 000000
1165 001432 000000
1166 001434 000000
1167 001436 000000
1168 001440 000000
1169 001442 000000
1170 001444 000000
1171 001446 000000
1172 001450 000000
1173 001452 000000
1174 001454 000000
1175 001456 000000
1176 001460 000000
1177 001462
1178
1179
1180
1181
1182
1183 001462
1184 000024
1185 000024 000200
1186 000044 000044
1187 000044 001462
1188 001462
1189
1190
1191
1192
1193 001462
1194 001462 000000
1195 001464 001234
1196 001466 000132
1197 001470 000137
1198 001472 000137
1199 001474 000052
1200
1201
1202
1203 001500
1204 001500
1205

```

```

:PROGRAM CONTROL FLAGS
:-----
EIAFLG: .BYTE 0 ;0=EIA 100000=20MA
INIFLG: .BYTE 00 ;PROGRAM INITIALIZATION FLAG
HDRFLG: .BYTE 00 ;PROGRAM INITIALIZATION FLAG FOR HEADER MAP
MNTFLG: .BYTE 00 ;MAINTENANCE BIT SET FLAG
DONFLG: .BYTE 0 ;TRANSMISSION COMPLETION FLAG
.EVEN
:DATA VARIABLES
TD0: .WORD 0
TD1: .WORD 000000
TD2: .WORD 000000
TD3: .WORD 000000
TD4: .WORD 000000
TD5: .WORD 000000
TD6: .WORD 000000
TD7: .WORD 000000
TR0: .WORD 000000
TR1: .WORD 000000
TR2: .WORD 000000
TR3: .WORD 000000
TR4: .WORD 000000
TR5: .WORD 000000
TR6: .WORD 000000
TR7: .WORD 0
STOP:
.SBTTL APT PARAMETER BLOCK
:*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
:*****
.SX= ;SAVE CURRENT LOCATION
.=24 ;SET POWER FAIL TO POINT TO START OF PROGRAM
200 ;FOR APT START UP
.=44 ;POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR ;POINT TO APT HEADER BLOCK
.=.SX ;RESET LOCATION COUNTER
:*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.
$APTHD:
$HIBTS: .WORD 0 ;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MADR: .WORD $MAIL ;ADDRESS OF APT MAILBOX (BITS 0-15)
$STMT: .WORD 90 ;RUN TIM OF LONGEST TEST
$PASTM: .WORD 95 ;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$SUNIT: .WORD 95 ;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
.WORD $ETEND-$MAIL/2 ;LENGTH MAILBOX-ETABLE(WORDS)
:DZ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
:-----
.=1500
DZ.MAP:

```

1206	001500	000001	DZCR0:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 0
1207	001502	000001	DZVC0:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 0
1208	001504	000001	DZLV0:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1209	001506	000001	LINE0:	.BLKW	1	:ALL LINES SELECTED
1210	001510	000001	PAR0:	.BLKW	1	:PARAMETERS
1211	001512	000001	MANT0:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1212						
1213	001514	000001	DZCR1:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 1
1214	001516	000001	DZVC1:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 1
1215	001520	000001	DZLV1:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1216	001522	000001	LINE1:	.BLKW	1	:ALL LINES SELECTED
1217	001524	000001	PAR1:	.BLKW	1	:PARAMETERS
1218	001526	000001	MANT1:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1219						
1220	001530	000001	DZCR2:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 2
1221	001532	000001	DZVC2:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 2
1222	001534	000001	DZLV2:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1223	001536	000001	LINE2:	.BLKW	1	:ALL LINES SELECTED
1224	001540	000001	PAR2:	.BLKW	1	:PARAMETERS
1225	001542	000001	MANT2:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1226						
1227	001544	000001	DZCR3:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 3
1228	001546	000001	DZVC3:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 3
1229	001550	000001	DZLV3:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1230	001552	000001	LINE3:	.BLKW	1	:ALL LINES SELECTED
1231	001554	000001	PAR3:	.BLKW	1	:PARAMETERS
1232	001556	000001	MANT3:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1233						
1234	001560	000001	DZCR4:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 4
1235	001562	000001	DZVC4:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 4
1236	001564	000001	DZLV4:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1237	001566	000001	LINE4:	.BLKW	1	:ALL LINES SELECTED
1238	001570	000001	PAR4:	.BLKW	1	:PARAMETERS
1239	001572	000001	MANT4:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1240						
1241	001574	000001	DZCR5:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 5
1242	001576	000001	DZVC5:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 5
1243	001600	000001	DZLV5:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1244	001602	000001	LINE5:	.BLKW	1	:ALL LINES SELECTED
1245	001604	000001	PAR5:	.BLKW	1	:PARAMETERS
1246	001606	000001	MANT5:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1247						
1248	001610	000001	DZCR6:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 6
1249	001612	000001	DZVC6:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 6
1250	001614	000001	DZLV6:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1251	001616	000001	LINE6:	.BLKW	1	:ALL LINES SELECTED
1252	001620	000001	PAR6:	.BLKW	1	:PARAMETERS
1253	001622	000001	MANT6:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1254						
1255	001624	000001	DZCR7:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 7
1256	001626	000001	DZVC7:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 7
1257	001630	000001	DZLV7:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1258	001632	000001	LINE7:	.BLKW	1	:ALL LINES SELECTED
1259	001634	000001	PAR7:	.BLKW	1	:PARAMETERS
1260	001636	000001	MANT7:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1261						

1262	001640	000001	DZCR10: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 10
1263	001642	000001	DZVC10: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 10
1264	001644	000001	DZLV10: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1265	001646	000001	LINE10: .BLKW	1	:ALL LINES SELECTED
1266	001650	000001	PAR10: .BLKW	1	:PARAMETERS
1267	001652	000001	MANT10: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1268					
1269	001654	000001	DZCR11: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 11
1270	001656	000001	DZVC11: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 11
1271	001660	000001	DZLV11: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1272	001662	000001	LINE11: .BLKW	1	:ALL LINES SELECTED
1273	001664	000001	PAR11: .BLKW	1	:PARAMETERS
1274	001666	000001	MANT11: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1275					
1276	001670	000001	DZCR12: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 12
1277	001672	000001	DZVC12: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 12
1278	001674	000001	DZLV12: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1279	001676	000001	LINE12: .BLKW	1	:ALL LINES SELECTED
1280	001700	000001	PAR12: .BLKW	1	:PARAMETERS
1281	001702	000001	MANT12: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1282					
1283	001704	000001	DZCR13: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 13
1284	001706	000001	DZVC13: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 13
1285	001710	000001	DZLV13: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1286	001712	000001	LINE13: .BLKW	1	:ALL LINES SELECTED
1287	001714	000001	PAR13: .BLKW	1	:PARAMETERS
1288	001716	000001	MANT13: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1289					
1290	001720	000001	DZCR14: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 14
1291	001722	000001	DZVC14: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 14
1292	001724	000001	DZLV14: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1293	001726	000001	LINE14: .BLKW	1	:ALL LINES SELECTED
1294	001730	000001	PAR14: .BLKW	1	:PARAMETERS
1295	001732	000001	MANT14: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1296					
1297	001734	000001	DZCR15: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 15
1298	001736	000001	DZVC15: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 15
1299	001740	000001	DZLV15: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1300	001742	000001	LINE15: .BLKW	1	:ALL LINES SELECTED
1301	001744	000001	PAR15: .BLKW	1	:PARAMETERS
1302	001746	000001	MANT15: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1303					
1304	001750	000001	DZCR16: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 16
1305	001752	000001	DZVC16: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 16
1306	001754	000001	DZLV16: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1307	001756	000001	LINE16: .BLKW	1	:ALL LINES SELECTED
1308	001760	000001	PAR16: .BLKW	1	:PARAMETERS
1309	001762	000001	MANT16: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1310					
1311	001764	000001	DZCR17: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 17
1312	001766	000001	DZVC17: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 17
1313	001770	000001	DZLV17: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1314	001772	000001	LINE17: .BLKW	1	:ALL LINES SELECTED
1315	001774	000001	PAR17: .BLKW	1	:PARAMETERS
1316	001776	000001	MANT17: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1317					

E03

MD-11-DZDZA-B MACY11 27(732) 03-NOV-76 15:03 PAGE 31
DZDZAB.P11 APT PARAMETER BLOCK

1318 002000 177777

DZ.END: 177777

;DEFINITIONS FOR TRAP SUBROUTINE CALLS
;POINTERS TO SUBROUTINES CAN BE FOUND
;IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS

1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360

002002 104400
002002 006556
104400
002004 005108
104402
002006 005132
104403
002010 005662
104404
002012 005766
104405
002014 006006
104406
002016 010332
104407
002020 006206
104410
002022 006246
104411
002024 006300
104412
002026 006304
104413
002030 006504
104414
002032 006536
104415
002034 024626
104416
002036 025022
104417
002040 006524

```

:*****
-----
:TRPTAB:
ADVANCE=TRAP+0 ;CALL TO ADVANCE TO NEXT TEST( OR SCOPE THIS ONE)
      .ADVANCE
SCOPI=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
      .SCOPI
TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
      .TYPE
INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
      .INSTR
INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER
      .INSTER
PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
      .PARAM
SETFLG=TRAP+6 ;CALL TO SET FLAG ROUTINE
      .SETFLG
SAVOS=TRAP+7 ;CALL TO REGISTER SAVE ROUTINE
      .SAVOS
RESOS=TRAP+10 ;CALL TO REGISTER RESTORE ROUTINE
      .RESOS
CONVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE
      .CONVRT
CNVRT=TRAP+12 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
      .CNVRT
DEVICE.CLR=TRAP+13 ;CALL TO ISSUE A DEVICE CLEAR
      .DEVICE.CLR
DELAY=TRAP+14 ;CALL TO DELAY FOR FAST CPU'S
      .DELAY
PARMD=TRAP+15 ;CONVERT DECIMAL STRING TO OCTAL
      .PARMD
PAWCH=TRAP+16 ;SET FLAG ECHO OR CABLE
      .PAWCH
DCLASM=TRAP+17 ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
      .DCLASM
:*****

```



```

1361                                     ;DZ11 VECTOR AND REGISTER INDIRECT POINTERS
1362                                     ;WORKING AREA
1363
1364 002042 160040 DZCSR: 160040 ;R/W
1365 002044 160041 HDZCSR: 160041 ;R/W
1366 002046 160042 DZRBUF: 160042 ;READ ONLY
1367 002050 160043 HDZRBUF: 160043 ;READ ONLY
1368 002052 160042 DZLPR: 160042 ;WRITE ONLY
1369 002054 160043 HDZLPR: 160043 ;WRITE ONLY
1370 002056 160044 DZTCR: 160044 ;R/W
1371 002060 160045 HDZTCR: 160045 ;R/W
1372 002062 160046 DZMSR: 160046 ;READ ONLY
1373 002064 160047 HDZMSR: 160047 ;READ ONLY
1374 002066 160046 DZTDR: 160046 ;WRITE ONLY
1375 002070 160047 HDZTDR: 160047 ;WRITE ONLY
1376                                     ;DEFAULT DZ VECTORS
1377 002072 000300 DZRIV: 300 ;REC INTR VECTOR
1378 002074 000302 DZRIS: 302 ;REC INTR STATUS
1379 002076 000304 DZTIV: 304 ;XMIT INTR VECTOR
1380 002100 000306 DZTIS: 306 ;XMIT INTR STATUS
1381
1382

```

1383
1384
1385
1386
1387 002102
1388 002102 000000
1389 002104 000000
1390 002106 000000
1391 002110 000000
1392 002112 000000
1393 002114 000000
1394 002116 000000
1395 002120 000000
1396 002122 000000
1397 002124 000000
1398 002126 000000
1399 002130 000000
1400 002132 000000
1401 002134 000000
1402 002136 000000
1403 002140 000000
1404 002142 000000
1405 002144 000000
1406 002146 000000

; TIME TABLE FOR RELATIVE TIMING TESTS

TMTBL:
T50: 0
T75: 0
T110: 0
T134: 0
T150: 0
T300: 0
T600: 0
T1200: 0
T1800: 0
T2000: 0
T2400: 0
T3600: 0
T4800: 0
T7200: 0
T9600: 0
TEIGHT: 0
TSEVEN: 0
TSIX: 0
TFIVE: 0

```

1407
1408
1409
1410
1411
1412
1413
1414
1415 002150
1416 002150 000005
1417 002152 012706 001120
1418 002156 106427 000340
1419 002162 012737 007500 000024
1420 002170 113737 001410 001411
1421 002176 005037 001242
1422 002202 105037 001123
1423 002206 012737 001500 001412
1424 002214 012737 000001 001406
1425 002222 005037 001132
1426 002226 005037 001136
1427 002232 005037 001122
1428 002236 012737 002150 001126
1429
1430
1431 002244 013746 000006
1432 002250 013746 000004
1433 002254 012737 002274 000004
1434 002262 022777 177777 176670
1435 002270 001402
1436 002272 000407
1437 002274 022626 20$:
1438 002276 012737 000176 001160 22$:
1439 002304 012737 000174 001162
1440 002312 012637 000004 21$:
1441 002316 012637 000006
1442 002322 005737 000042
1443 002326 001402
1444 002330 000137 004114
1445 002334 105737 001415 31$:
1446 002340 001004
1447 002342 104402 001000
1448 002346 105337 001415
1449 002352 105737 001255 29$:
1450 002356 100004
1451 002360 004737 011260
1452 002364 000137 004140
1453 002370 032777 000001 176562 30$:
1454 002376 001011
1455 002400 122737 000377 001415
1456 002406 001003
1457 002410 105777 176544
1458 002414 100402
1459 002416 000137 003104
1460 002422 012700 001500 32$:
1461 002426 105037 001416
1462 002432 005020 65$:

```

```

;PROGRAM INITIALIZATION
;LOCK OUT INTERRUPTS
;SET UP PROCESSOR STACK
;SET UP POWER FAIL VECTOR
;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
;TYPE TITLE MESSAGE

.START:
RESET
MOV #STACK,SP
MTPS #PR7
MOV #SPWRDN,2#24
MOVVB DZNUM,SAVNUM
CLR SPASS
CLRB SERFLG
MOV #DZ.MAP,ACTIVE
MOV #1,RUN
CLR SERTTL
CLR SERRPC
CLR STSTNM
MOV #.START,$LPADR

;CLEAR THE WORLD. START NEW ENVIRONMENT
;SET UP STACK
;LOCK OUT INTERRUPTS
;SET UP POWER FAIL VECTOR
;SAVE NUMBER OF DEVICES IN SYSTEM.
;CLEAR PASS COUNT
;CLEAR ERROR FLAG
;GET MAP POINTER.
;POINT POINTER TO FIRST DEVICE.
;CLEAR ERROR COUNT
;CLEAR LAST ERROR POINTER
;SET UP FOR TEST 1
;SET UP FOR POWER FAIL BEFORE
;TESTING STARTS

;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY
MOV 6,-(SP)
MOV 4,-(SP)
MOV #20$,4
CMP #-1,$SWR
BEQ 22$
BR 21$
POP2SP
MOV #SWREG,$SWR
MOV #DISPREG,$DISPLAY
MOV (SP)+,4
MOV (SP)+,6
TST 42
BEQ 31$
JMP 63$
TSTB INIFLG
BNE 29$
TYPE $MTITLE
DECB INIFLG
TSTB $ENVM
BPL 30$
JSR PC,$SETAPT
JMP 16$
BIT #SW00,$SWR
BNE 32$
CMPB #377,$INIFLG
BNE ,+10
TSTB $SWR
BMI 32$
JMP 73$
MOV #DZ.MAP,$RO
CLRB HDRFLG
CLR (RO)+

;SAVE BUS ERROR PS
;SAVE BUS ERROR PC
;SET UP TO TRAP TO THIS ROUTINE
;CAN 177570 BE REFERENCED?
;IF SO AND IT IS -1, TREAT LIKE SWITCHLESS
;IF YES, SKIP AROUND THE SETUP
;REMOVE THE TRAP FROM THE STACK
;IF NO TRAP COMES HERE. POINT TO SOFTWARE SWR
;POINT TO SOFTWARE DISPLAY REGISTER
;RESTORE THE BUS ERROR VECTOR

;WORKING UNDER A MONITOR ?
;NO
;IF YES, SKIP THE TERMINAL INTERROGATION
;HAVE WE ALREADY BEEN HERE TODAY?
;IF SO, SKIP PRINTING THE TITLE
;PRINT THE DIAGNOSTIC'S TITLE
;SET THE ONCE ONLY FLAG
;DETERMINE WHETHER APT SIZING SHOULD BE DONE
;IF NOT, GO CHECK FOR AUTO-SIZING
;OTHERWISE, GO DO APT SIZING FROM ETABLE
;GO PRINT DZ STATUS TABLE
;RESELECT ?
;IF YES, GO SET UP THE INFORMATION
;ON 1ST START; MUST ANSWER QUESTION
;IF NOT ANSWERING QUESTIONS
;ARE U AUTO SIZING?
;NO AUTO SIZE! NO SW00=1 ON 1ST START!
;IF NO, SKIP THE INTERROGATION
;POINT TO THE BEGINNING OF THE MAP TABLE
;MAKE SURE A MAP GETS PRINTED
;CLEAR A TABLE LOCATION

```

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 36
PROGRAM INITIALIZATION AND START UP.

```

1463 002434 020027 002000          CMP      RD,#DZ.END      ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
1464 002440 001374                    BNE      65$            ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE
1465 002442 105337 001415          DECB     INIFLG         ;INSURE NO AUTO SIZING IF QUESTIONS ANSWERED!
1466
1467
1468
1469
1470
1471
1472 002446                    33$:
1473 002446 104403          INSTR     ;CALL THE STRING INPUT ROUTINE
1474 002450 003312          66$      ;POINTER TO MESSAGE TO BE PRINTED
1475 002452 104405          PARAM     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1476 002454 160000          160000   ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1477 002456 163770          163770   ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1478 002460 001500          DZCRO    ;POINTER TO MAP LOCATION TO BE FILLED
1479 002462 007           .BYTE    7      ;MASK OF INVALID BITS FOR THIS PARAMETER
1480 002463 001           .BYTE    1      ;NUMBER OF PARAMETERS TO STORE
1481 002464 013737 001500 001310  MOV      DZCRO,$BASE ;COPY BASE ADDRESS TO ETABLE
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498 002516 104403          34$:
1499 002520 003417          INSTR     ;CALL THE STRING INPUT ROUTINE
1500 002522 104405          67$      ;POINTER TO MESSAGE TO BE PRINTED
1501 002524 000004          PARAM     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1502 002526 000007          300      ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1503 002530 001502          776      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1504 002532 000           DZVCO    ;POINTER TO MAP LOCATION TO BE FILLED
1505 002533 001           .BYTE    3      ;MASK OF INVALID BITS FOR THIS PARAMETER
1506 002534 113737 001502 001304  .BYTE    1      ;NUMBER OF PARAMETERS TO STORE
1507 002542 106337 001305          MOV      DZVCO,$VECT1 ;COPY VECTOR TO ETABLE
1508 002546 106337 001305          ;GET THE BUS REQUEST LEVEL
1509 002552 106337 001305          INSTR     ;CALL THE STRING INPUT ROUTINE
1510 002556 106337 001305          68$      ;POINTER TO MESSAGE TO BE PRINTED
1511 002562 106337 001305          PARAM     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1512
1513
1514
1515 002566 104402 004000          4        ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1516 002572 005037 001220          7        ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1517 002576 105777 176362          DZLVD    ;POINTER TO MAP LOCATION TO BE FILLED
1518 002602 100375          .BYTE    0      ;MASK OF INVALID BITS FOR THIS PARAMETER
                                .BYTE    1      ;NUMBER OF PARAMETERS TO STORE
                                MOVVB   DZLVD,$VECT1+1 ;GET BUS REQUEST LEVEL INTO ETABLE
                                ASLB    $VECT1+1 ;ALIGN THE BITS PROPERLY
                                ASLB    $VECT1+1 ;ALIGN THE BITS PROPERLY
                                ASLB    $VECT1+1 ;ALIGN THE BITS PROPERLY
                                ASLB    $VECT1+1 ;ALIGN THE BITS PROPERLY
                                ASLB    $VECT1+1 ;ALIGN THE BITS PROPERLY
                                ASLB    $VECT1+1 ;ALIGN THE BITS PROPERLY
                                ;FIND OUT IF MODULE IS EIA OR 20 MA.
                                TYPE     74$
                                CLR      $TMP1 ;PRINT EIA MESSAGE
                                TSTB   $STKS ;USE $TMP1
                                BPL     80$    ;IS KEYBOARD DONE?
                                ;IF NOT, WAIT FOR IT

```

```

1519 002604 017746 176356          MOV      @STKB,-(SP)      ;IF YES, PUT CHARACETR ON STACK
1520 002610 042716 000240          BIC      #240,(SP)      ;STRIP DOWN CHARACTER
1521 002614 122726 000015          CMPB    #15,(SP)+      ;IS IT ?
1522 002620 001414                   BEQ      81$           ;IF SO, GET OUT
1523 002622 014677 176344          MOV      -(SP),@STPB    ;IF NOT, PRINT CHARACTER
1524 002626 042737 100000 001504    BIC      #BIT15,DZLVO   ;CLEAR EIA FLAG
1525 002634 122726 000102          CMPB    #102,(SP)+     ;IS IT A B?
1526 002640 001356                   BNE     80$           ;IF NOT, GO BACK FOR INPUT
1527 002642 052737 100000 001504    BIS      #BIT15,DZLVO   ;IF SO, SET FLAG
1528 002650 000752                   BR       80$           ;GET MORE INPUT
1529 002652                   81$:
1530
1531                               ;GET THE MODE OF OPERATION (E,I,S)
1532
1533 002652 104403          INSTR    ;CALL THE STRING INPUT ROUTINE
1534 002654 003630          72$     ;POINTER TO THE MESSAGE TO BE PRINTED
1535 002656 104406          SETFLG   ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
1536 002660 001512          MANTO    ;THIS IS THE FLAG BEING SETUP
1537
1538                               ;GET THE NUMBER OF DZ11'S RUNNING
1539
1540 002662 104403          INSTR    ;CALL THE STRING INPUT ROUTINE
1541 002664 003566          71$     ;POINTER TO MESSAGE TO BE PRINTED
1542 002666 104405          PARAM   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1543 002670 000001          1       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1544 002672 000020          16.     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1545 002674 001220          $TMP1   ;POINTER TO MAP LOCATION TO BE FILLED
1546 002676 000       .BYTE 0 ;MASK OF INVALID BITS FOR THIS PARAMETER
1547 002677 001       .BYTE 1 ;NUMBER OF PARAMETERS TO STORE
1548
1549 002700 012737 000377 001506    MOV      #377,LINED    ;SET UP DEFAULT LINES
1550 002706 012737 017470 001510    MOV      #17470,PARO   ;SET UP DEFAULT LPR PARAMETER
1551
1552 002714 032777 000010 176236    BIT      #SW03,JSWR    ;RECEIVER ON; 19.2 KBAUD; 2STOP BITS; 8 BIT/CHAR
1553 002722 001402                   BEQ     40$           ;DO YOU WANT PARAMETERS?
1554 002724 004737 003134                   JSR     PC,23$       ;IF NO, SKIP THE PARAMETER CALL
1555 002730 012737 000001 001312 40$:   MOV      #1,$DEVN     ;GET PARAMETERS
1556 002736 113737 001220 001410     MOVVB   $TMP1,DZNUM   ;INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
1557 002744 113737 001220 001411     MOVVB   $TMP1,SAVNUM  ;COPY THE NUMBER OF DEVICES
1558 002752 005337 001220 62$:   DEC     $TMP1        ;COPY A BACKUP NUMBER
1559 002756 001404                   BEQ     61$           ;$TMP1 CONTAINS THE COUNT OF UNINITIALIZED
1560 002760 000261                   SEC     ;SELECTED DEVICES
1561 002762 006137 001312                   ROL     $DEVN        ;SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
1562 002766 000771                   BR      62$         ;POINT TO THE NEXT DEVICE
1563 002770 013737 001312 001222 61$:   MOV     $DEVN,$TMP2  ;GO DO THIS PROCEDURE AGAIN
1564 002776 013737 001312 001404     MOV     $DEVN,DZACTV ;# OF TIMES
1565 003004 012700 001500                   MOV     #DZCR0,R0    ;COPY THE ACTIVE DEVICE PARAMETER
1566 003010 012701 001514                   MOV     #DZCR1,R1    ;SET A POINTER TO THE SPECIFIED INFORMATION
1567 003014 012702 001320                   MOV     #SDDWD,R2    ;POINT R1 TO THE REST OF THE MAP TABLE
1568 003020 000241                   CLC     ;POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
1569 003022 006037 001222                   ROR     $TMP2        ;INITIALIZE THE "C" BIT FOR A ROTATION
1570 003026 006237 001222 64$:   ASR     $TMP2        ;SKIP MAPPING SETUP FOR DEVICE 0- IT'S DONE
1571 003032 103404                   BCS    41$           ;ISOLATE A SELECTION FLAG IN THE "C" BIT
1572 003034 012711 177777                   MOV     #-1,(R1)     ;IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
1573 003040 000137 004114                   JMP     63$         ;TERMINATE THE LIST
1574 003044 012011 41$:   MOV     (R0)+,(R1)   ;GO TO THE NEXT BLOCK
;ADDRESS

```

```

1575 003046 062721 000010      ADD      #10,(R1)+      ;POINT TO THE NEXT DZ11 ADDRESS VALUE
1576 003052 012011      MOV      (R0)+,(R1)    ;VECTOR
1577 003054 062721 000010      ADD      #10,(R1)+      ;POINT TO THE NEXT VECTOR VALUE
1578 003060 012021      MOV      (R0)+,(R1)+    ;LEVEL
1579 003062 012021      MOV      (R0)+,(R1)+    ;LINES
1580 003064 016012 177774      MOV      -4(R0),(R2)    ;GET THE EIA FLAG FROM THE PRIORITY WORD
1581 003070 042712 077777      BIC      #77777,(R2)    ;ISOLATE THAT FLAG
1582 003074 051022      BIS      (R0),(R2)+     ;ADD PARAMETERS TO DEVICE DESCRIPTOR WORD
1583 003076 012021      MOV      (R0)+,(R1)+    ;PARAMETERS
1584 003100 012021      MOV      (R0)+,(R1)+    ;MAINTENANCE MODE
1585 003102 000751      BR       64$           ;
1586 003104 032777 000010 176046 73$:    BIT      #SW03,DSWR     ;ASK PARAMETERS ?
1587 003112 001002      BNE      42$           ;IF NO, GO DO AUTO SIZING
1588 003114 000137 004114      JMP      63$           ;GO SET UP FOR AUTO SIZING
1589 003120 004737 003134      JSR      PC,23$        ;GO ASK PARAMETERS
1590 003124 105337 001415      DECB     INIFLG        ;INSURE NO AUTO SIZE IF QUESTIONS ANSWERED
1591 003130 000137 004140      JMP      16$           ;GO TO THE NEXT BLOCK
1592
1593                               ;GET THE ACTIVE LINES PARAMETER
1594
1595                               23$:
1596 003134      104403      INSTR     ;CALL THE STRING INPUT ROUTINE
1597 003136      003442      69$      ;POINTER TO MESSAGE TO BE PRINTED
1598 003140      104405      PARAM    ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1599 003142      000001      1        ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1600 003144      000377      377      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1601 003146      001506      LINEO    ;POINTER TO MAP LOCATION TO BE FILLED
1602 003150      000      .BYTE    0        ;MASK OF INVALID BITS FOR THIS PARAMETER
1603 003151      001      .BYTE    1        ;NUMBER OF PARAMETERS TO STORE
1604 003152      105037 001416      CLRB     HDRFLG       ;MAKE SURE THE CHANGES ARE PRINTED
1605
1606                               ;THIS SEGMENT CHECKS TO MAKE SURE THE LINE PARAMETER JUST ENTERED
1607                               ;IS LEGITIMATE IN STAGGERED MODE OPERATION IF THAT MODE WAS SELECTED
1608
1609 003156 005737 001512      TST      MANTO        ;IS STAGGERED THE MODE OF OPERATION?
1610 003162 100021      BPL      26$          ;IF NOT, SKIP THIS SEGMENT
1611 003164 013703 001506      MOV      LINEO,R3     ;GET A SCRATCH COPY OF THE ACTIVE LINES
1612 003170 006003      ROR      R3           ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
24$: 1613 003172 103410      BCS      25$          ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
1614 003174 001414      BEQ      26$          ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
1615 003176 006203      ASR      R3           ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
1616 003200 103373      BCC      24$          ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
27$: 1617 003202 104402 001230      TYPE     ,SQUES       ;THIS IS AN INCORRECT PARAMETER
1618 003206 104402 010256      TYPE     ,MBADLN      ;LET THE USER KNOW ABOUT IT
25$: 1619 003212 000750      BR       23$          ;GO GET THE CORRECT PARAMETER
1620 003214 001772      BEQ      27$          ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
1621 003216 006203      ASR      R3           ;GET THE NEXT FLAG
1622 003220 103370      BCC      27$          ;IF IT ISN'T SET, THERE'S AN ERROR
1623 003222 000241      CLC      ;INITIALIZE THE "C" BIT FOR TESTING OF THE NEXT PAIR
1624 003224 000761      BR       24$          ;GO TEST THE NEXT PAIR OF FLAGS
1625
1626                               ;GET THE LINE PARAMETER REGISTER ARGUMENT
1627
26$: 1628 003226      104403      INSTR     ;CALL THE STRING INPUT ROUTINE
1629 003226      003516      70$      ;POINTER TO MESSAGE TO BE PRINTED
1630

```



```

1674 004244 101404          BLOS      2$
1675 004246 104402 010024      TYPE      ,MERR3
1676 004252 000000          9$:      HALT
1677 004254 000776          BR        9$
1678 004256 017737 174676 001404 2$:      MOV      JSWR,DZACTV
1679 004264 013700 001404      MOV      DZACTV,R0
1680 004270 000000          HALT
1681 004272 032777 000020 174660 3$:      BIT      #SW04,JSWR
1682 004300 001407          BEQ      18$
1683 004302 104403          INSTR
1684 004304 004062          75$
1685 004306 104405          PARAM
1686 004310 000001          1
1687 004312 177777          177777
1688 004314 006554          DLYCNT
1689 004316 000          .BYTE    0
1690 004317 001          .BYTE    1
1691 004320 012700 000300 18$:      MOV      #300,R0
1692 004324 012701 000302      MOV      #302,R1
1693 004330 010120 4$:      MOV      R1,(R0)+
1694 004332 005021          CLR      (R1)+
1695 004334 022021          CMP      (R0)+,(R1)+
1696 004336 022700 001000      CMP      #1000,R0
1697 004342 001372          BNE      4$
1698
1699          ;TEST START AND RESTART
1700          -----
1701
1702 004344 012706 001120      .BEGIN:  MOV      #STACK,SP
1703 004350 106427 000340      MTPS     #PR7
1704 004354 005737 000042      TST     JS#42
1705 004360 001015          BNE     2$
1706 004362 032777 000004 174570      BIT     #BIT2,JSWR
1707 004370 001406          BEQ     1$
1708 004372 104402 010050          TYPE     ,MLOCK
1709 004376 012737 000240 004660      MOV     #NOP,TTST
1710 004404 000403          BR     2$
1711 004406 013737 005102 004660 1$:      MOV     BRW,TTST
1712 004414 012737 010722 001126 2$:      MOV     #CYCLE,SLPADR
1713 004422 104402 007741          TYPE     MR
1714 004426 000177 174474          JMP     JSLPADR

```

```

;BR IF NUMBER IS OK.
;TELL USER OF INVALID NUMBER.
;STOP EVERY THING.
;RESTART THE PROGRAM AGAIN.
;GET NEW DEVICE PATTERN
;SHOW THE USER WHAT HE SELECTED.
;CONTINUE DYNAMIC SWITCHES.
;CHECK TO SEE IF DELAY COUNT CHANGES
;IF NOT, GO CLEAR VECTOR AREA
;CALL THE STRING INPUT ROUTINE
;POINTER TO MESSAGE TO BE PRINTED
;CALL THE OCTAL TO ASCII CONVERT ROUTINE
;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
;POINTER TO MAP LOCATION TO BE FILLED
;MASK OF INVALID BITS FOR THIS PARAMETER
;NUMBER OF PARAMETERS TO STORE
;PREPARE TO CLEAR THE FLOATING
;VECTOR AREA. 300-776
;START PUTTING "PC+2 - HALT"
;IN VECTOR AREA.
;POP POINTERS
;ALL DONE??
;BR IF NO.

```

```

;SET UP STACK
;LOCK OUT INTERRUPTS
;IS PROGRAM UNDER MONITOR CONTROL
;BR IF YES
;CHECK FOR LOCK ON TEST
;BR IF NO LOCK DESIRED.
;TYPE LOCK SELECTED.
;ADJUST SCOPE ROUTINE.
;CONTINUE ALONG.
;PREPARE NORMAL SCOPE ROUTINE
;START AT "CYCLE" FIND WHICH DEVICE TO TEST
;TYPE "RUNNING"
;START TESTING

```


1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770

004432
004432 000004
004434 005037 001136
004440 105037 001123
004444 104402 007715
004450 104402 010077
004454 104412 004612
004460 104402 010105
004464 104412 004620
004470 005237 001242
004474 104402 010113
004500 104412 004626
004504 005337 001242
004510 104402 010124
004514 104412 004634
004520 105337 001411
004524 001030
004526 113737 001410 001411
004534 005037 001226
004540 005237 001242
004544 042737 100000 001242
004552 005327
004554 000001
004556 003013
004560 012737
004562 000001
004564 004554 000042
004566 013700
004572 001405
004574 000005
004576 004710
004600 000240
004602 000240
004604 000240
004606
004606 000137
004610 010722
004612 000001
004614 006 002
004616 002042
004620 000001
004622 003 002
004624 002072

```

:END OF PASS
:TYPE NAME OF TEST
:UPDATE PASS COUNT
:CHECK FOR EXIT TO ACT-11
:RESTART TEST
.SBTTL END OF PASS ROUTINE

:*****
:INCREMENT THE PASS NUMBER ($PASS)
:*IF THERES A MONITOR GO TO IT
:*IF THERE ISN'T JUMP TO CYCLE

SEOP:
SCOPE
CLR SERRPC :CLEAR LAST ERROR PC
CLRB SERFLG :CLEAR ERROR FLAG
TYPE ,MEPASS :TYPE END PASS
TYPE ,MCSRX :TYPE CSR
CNVRT ,XCSR :SHOW IT
TYPE ,MVECX :TYPE VECTOR
CNVRT ,XVEC :SHOW IT
INC $PASS :RAISE PASS COUNT
TYPE ,MPASSX :TYPE PASSES
CNVRT ,XPASS :SHOW IT
DEC $PASS :RESTORE PASS COUNT
TYPE ,MERRX :TYPE ERRORS
CNVRT ,XERR :SHOW IT
DECB SAVNUM :ARE ALL DEVICES TESTED?
BNE SDOAGN :BR IF NO.
MOVB DZNUM, SAVNUM :RESTORE THE COUNT
CLR $TIMES :ZERO THE NUMBER OF ITERATIONS
INC $PASS :INCREMENT THE PASS NUMBER
BIC #100000, $PASS :DON'T ALLOW A NEG. NUMBER
DEC (PC)+ :LOOP?

SEOPCT: .WORD 1 :YES
BGT SDOAGN :RESTORE COUNTER
MOV (PC)+, 2(PC)+

SENDCT: .WORD 1

SGET42: MOV 2#42, R0 :GET MONITOR ADDRESS
BEQ SDOAGN :BRANCH IF NO MONITOR
RESET :CLEAR THE WORLD
SENDAD: JSR PC, (R0) :GO TO MONITOR
NOP :SAVE ROOM
NOP :FOR
NOP :ACT11

SDOAGN: JMP 2(PC)+ :RETURN
SRTNAD: .WORD CYCLE

XCSR: 1
.BYTE 6,2
DZCSR

XVEC: 1
.BYTE 3,2
DZRIV
```

```

1771 004626 000001
1772 004630 006 002
1773 004632 001242
1774 004634 000001
1775 004636 006 002
1776 004640 001132
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793 004642
1794 004642 004737 007212
1795 004646 005037 001136
1796 004652 022716 012170
1797 004656 001413
1798 004660 000406
1799 004662 105777 174276
1800 004666 100067
1801 004670 017766 174272 177776
1802 004676 032777 040000 174254
1803 004704 001060
1804
1805 004706 000416
1806
1807 004710 013746 000004
1808 004714 012737 004734 000004
1809 004722 005737 177060
1810 004726 012637 000004
1811 004732 000436
1812 004734 022626
1813 004736 012637 000004
1814 004742 000441
1815 004744
1816 004744 105737 001123
1817 004750 001404
1818 004752 105037 001123
1819 004756 005037 001226
1820 004762 032777 004000 174170
1821 004770 001011
1822 004772 005737 001242
1823 004776 001406
1824 005000 005237 001124
1825 005004 023737 001226 001124
1826 005012 002015

```

```

XPASS: 1
        .BYTE 6,2
        $PASS
XERR: 1
        .BYTE 6,2
        $ERTTL
;SCOPE LOOP AND ITERATION HANDLER
-----
.SBTTL SCOPE HANDLER ROUTINE
;*****
;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
;AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SW14=1 LOOP ON TEST
;SW11=1 INHIBIT ITERATIONS
;CALL
;* SCOPE ;;SCOPE=10T
$SCOPE:
.SCOPE: JSR PC,SERV.G ;FIND OUT IF <?G> WAS HIT
        CLR $ERRPC ;CLEAR LAST ERROR PC.
        CMP #TST1+2,(SP) ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
        BEQ $XTSTR ;IF SO, DON'T LOOP ON IT
        BR 1$ ;GOTO 1$ (IF LOCK SMD2=1; THIS LOC =240)
        TSTB $STKS ;KEYBOARD DONE?
        BPL $OVER ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
        MOV $STKB,-2(SP) ;CLEAR DONE BIT
        BIT #BIT14,$SWR ;LOOP ON PRESENT TEST?
        BNE $OVER ;YES IF SW14=1
;*****START OF CODE FOR THE XOR TESTER*****
$XTSTR: BR 6$ ;IF RUNNING ON THE "XOR" TESTER CHANGE
        MOV $ERRVEC,-(SP) ;THIS INSTRUCTION TO A "NOP" (NOP=240)
        MOV #5,$ERRVEC ;SAVE THE CONTENTS OF THE ERROR VECTOR
        TST #177060 ;SET FOR TIMEOUT
        MOV (SP)+,$ERRVEC ;TIME OUT ON XOR?
        BR $SVLAD ;RESTORE THE ERROR VECTOR
        CMP (SP)+,(SP)+ ;GO TO THE NEXT TEST
        MOV (SP)+,$ERRVEC ;CLEAR THE STACK AFTER A TIME OUT
        BR $OVER ;RESTORE THE ERROR VECTOR
        ;LOOP ON THE PRESENT TEST
6$:;*****END OF CODE FOR THE XOR TESTER*****
2$: TSTB $ERFLG ;HAS AN ERROR OCCURRED?
        BEQ 3$ ;BR IF NO
        CLRB $ERFLG ;ZERO THE ERROR FLAG
        CLR $TIMES ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
        BIT #BIT11,$SWR ;INHIBIT ITERATIONS?
        BNE 1$ ;BR IF YES
        TST $PASS ;IF FIRST PASS OF PROGRAM
        BEQ 1$ ;INHIBIT ITERATIONS
        INC $ICNT ;INCREMENT ITERATION COUNT
        CMP $TIMES,$ICNT ;CHECK THE NUMBER OF ITERATIONS MADE
        BGE $OVER ;;BR IF MORE ITERATION REQUIRED

```

```

1827 005014 012737 000001 001124 1S:  MOV  #1,SICNT      ;;REINITIALIZE THE ITERATION COUNTER
1828 005022 013737 005104 001226      MOV  SMXCNT,STIMES ;;SET NUMBER OF ITERATIONS TO DO
1829 005030 105237 001122      $SVLAD: INCB  STSTNM      ;;COUNT TEST NUMBERS
1830 005034 113737 001122 001240      MOV  STSTNM,STESTN ;;SET TEST NUMBER IN APT MAILBOX
1831 005042 011637 001126      MOV  (SP),SLPADR  ;;SAVE SCOPE LOOP ADDRESS
1832 005046 013777 001122 174106 $OVER: MOV  STSTNM,DISP  ;;DISPLAY TEST NUMBER
1833 005054 013716 001126      MOV  SLPADR,(SP)  ;;FUDGE RETURN ADDRESS
1834 005060 105037 001417      3S:  CLRB  MNTFLG    ;;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TEST
1835 005064 005737 001370      TST  MODE        ;;HAS THE MODE BEEN CHANGED?
1836 005070 001003      BNE  4S          ;;IF NOT INTERNAL, GO DO A TEST
1837 005072 112737 000010 001417      MOV  #MAINT,MNTFLG ;;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
1838 005100 000002      4S:  RTI          ;;GO DO THE TEST
1839 005102 000406      BRW: 406
1840 005104 000005      SMXCNT: 5          ;;MAX. NUMBER OF ITERATIONS
1841
1842      ;;CHECK FOR FREEZE ON CURRENT DATA
1843      -----
1844
1845 005106 032777 001000 174044 .SCOPI: BIT  #SW09,DSWR  ;;IS SW09=1(SET)?
1846 005114 001405      BEQ  1S          ;;BR IF NOT SET.
1847 005116 005737 001362      TST  LOCK        ;;IS THER A TIGHT LOOP SPECIFIED?
1848 005122 001402      BEQ  1S          ;;IF NO, RETURN
1849 005124 013716 001362      MOV  LOCK,(SP)   ;;IF YES, GOTO THE ADDRESS IN LOCK.
1850 005130 000002      1S:  RTI          ;;GO BACK.
1851
1852 005132      .TYPE:
1853      .SBTTL  TYPE ROUTINE
1854
1855      ;;*****
1856      ;;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1857      ;;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1858      ;;NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1859      ;;NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1860      ;;NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1861      ;;
1862      ;;CALL:
1863      ;;#1) USING A TRAP INSTRUCTION
1864      ;;      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
1865      ;;OR
1866      ;;      TYPE
1867      ;;      MESADR
1868      ;;
1869
1870 005132 105737 001177      STYPE: TSTB  STPFLG  ;;IS THERE A TERMINAL?
1871 005136 100002      BPL  1S          ;;BR IF YES
1872 005140 000000      HALT  ;;HALT HERE IF NO TERMINAL
1873 005142 000430      BR  3S          ;;LEAVE
1874 005144 010046      1S:  MOV  RO,-(SP)  ;;SAVE RO
1875 005146 017600 000002      MOV  #22(SP),RO  ;;GET ADDRESS OF ASCIZ STRING
1876 005152 122737 000001 001254      CMPB #APTENV,SENV ;;RUNNING IN APT MODE
1877 005160 001011      BNE  62S        ;;NO GO CHECK FOR APT CONSOLE
1878 005162 132737 000100 001255      BITB #APTPOOL,SENV ;;SPOOL MESSAGE TO APT
1879 005170 001405      BEQ  62S        ;;NO GO CHECK FOR CONSOLE
1880 005172 010037 005202      MOV  RO,61S    ;;SETUP MESSAGE ADDRESS FOR APT
1881 005176 004737 005422      JSR  PC,SATY3  ;;SPOOL MESSAGE TO APT
1882 005202 000000      61S: .WORD  0   ;;MESSAGE ADDRESS

```

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 44
TYPE ROUTINE

```

1883 005204 132737 000040 001255 62$: BITB #APTCSUP,SENVM ;;APT CONSOLE SUPPRESSED
1884 005212 001003 BNE 60$ ;;YES,SKIP TYPE OUT
1885 005214 112046 2$: MOVB (RO)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK
1886 005216 001005 BNE 4$ ;;BR IF IT ISN'T THE TERMINATOR
1887 005220 005726 TST (SP)+ ;;IF TERMINATOR POP IT OFF THE STACK
1888 005222 012600 60$: MOV (SP)+,RO ;;RESTORE RO
1889 005224 062716 000002 3$: ADD #2,(SP) ;;ADJUST RETURN PC
1890 005230 000002 RTI ;;RETURN
1891 005232 122716 000011 4$: CMPB #HT,(SP) ;;BRANCH IF <HT>
1892 005236 001430 BEQ 8$
1893 005240 122716 000200 CMPB #CRLF,(SP) ;;BRANCH IF NOT <CRLF>
1894 005244 001006 BNE 5$
1895 005246 005726 TST (SP)+ ;;POP <CR><LF> EQUIV
1896 005250 104402 TYPE ;;TYPE A CR AND LF
1897 005252 001231 SCRLF
1898 005254 105037 005410 CLRB $CHARCNT ;;CLEAR CHARACTER COUNT
1899 005260 000755 BR 2$ ;;GET NEXT CHARACTER
1900 005262 004737 005344 5$: JSR PC,$TYPEC ;;GO TYPE THIS CHARACTER
1901 005266 123726 001176 6$: CMPB $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
1902 005272 001350 BNE 2$ ;;IF NO GO GET NEXT CHAR.
1903 005274 013746 001174 MOV $NULL,-(SP) ;;GET # OF FILLER CHARS. NEEDED
1904 AND THE NULL CHAR.
1905 005300 105366 000001 7$: DECB 1(SP) ;;DOES A NULL NEED TO BE TYPED?
1906 005304 002770 BLT 6$ ;;BR IF NO--GO POP THE NULL OFF OF STACK
1907 005306 004737 005344 JSR PC,$TYPEC ;;GO TYPE A NULL
1908 005312 105337 005410 DECB $CHARCNT ;;DO NOT COUNT AS A COUNT
1909 005316 000770 BR 7$ ;;LOOP
1910
1911 ;HORIZONTAL TAB PROCESSOR
1912
1913 005320 112716 000040 8$: MOVB #' (SP) ;;REPLACE TAB WITH SPACE
1914 005324 004737 005344 9$: JSR PC,$TYPEC ;;TYPE A SPACE
1915 005330 132737 000007 005410 BITB #7,$CHARCNT ;;BRANCH IF NOT AT
1916 005336 001372 BNE 9$ ;;TAB STOP
1917 005340 005726 TST (SP)+ ;;POP SPACE OFF STACK
1918 005342 000724 BR 2$ ;;GET NEXT CHARACTER
1919 005344 105777 173620 $TYPEC: TSTB $STPS ;;WAIT UNTIL PRINTER IS READY
1920 005350 100375 BPL $TYPEC
1921 005352 116677 000002 173612 MOVB 2(SP),$STPB ;;LOAD CHAR TO BE TYPED INTO DATA REG.
1922 005360 122766 000015 000002 CMPB #CR,2(SP) ;;IS CHARACTER A CARRIAGE RETURN?
1923 005366 001003 BNE 1$ ;;BRANCH IF NO
1924 005370 105037 005410 CLRB $CHARCNT ;;YES--CLEAR CHARACTER COUNT
1925 005374 000406 BR $TYPEX ;;EXIT
1926 005376 122766 000012 000002 1$: CMPB #LF,2(SP) ;;IS CHARACTER A LINE FEED?
1927 005404 001402 BEQ $TYPEX ;;BRANCH IF YES
1928 005406 105227 INCB (PC)+ ;;COUNT THE CHARACTER
1929 005410 000000 $CHARCNT: .WORD 0 ;;CHARACTER COUNT STORAGE
1930 005412 000207 $TYPEX: RTS PC
1931
1932 .SBTTL APT COMMUNICATIONS ROUTINE
1933
1934 ;;*****
1935 005414 112737 000001 005660 $ATY1: MOVB #1,$FFLG ;;TO REPORT FATAL ERROR
1936 005422 112737 000001 005656 $ATY3: MOVB #1,$MFLG ;;TO TYPE A MESSAGE
1937 005430 000403 BR $ATYC
1938 005432 112737 000001 005660 $ATY4: MOVB #1,$FFLG ;;TO ONLY REPORT FATAL ERROR

```

```

1939 005440          SATYC:
1940 005440 010046      MOV      RO,-(SP)      ;; PUSH RO ON STACK
1941 005442 010146      MOV      R1,-(SP)      ;; PUSH R1 ON STACK
1942 005444 105737 005656  TSTB     $MFLG        ;; SHOULD TYPE A MESSAGE?
1943 005450 001450      BEQ      5$           ;; IF NOT: BR
1944 005452 122737 000001 001254  CMPB     #APTENV,SENV  ;; OPERATING UNDER APT?
1945 005460 001031      BNE      3$           ;; IF NOT: BR
1946 005462 132737 000100 001255  BITB     #APTPOOL,SENV  ;; SHOULD SPOOL MESSAGES?
1947 005470 001425      BEQ      3$           ;; IF NOT: BR
1948 005472 017600 000004      MOV      24(SP),RO    ;; GET MESSAGE ADDR.
1949 005476 062766 000002 000004  ADD      #2,4(SP)     ;; BUMP RETURN ADDR.
1950 005504 005737 001234      1$: TST     $MSGTYPE     ;; SEE IF DONE W/ LAST XMISSION?
1951 005510 001375      BNE      1$          ;; IF NOT: WAIT
1952 005512 010037 001250      MOV      RO,$MSGAD    ;; PUT ADDR IN MAILBOX
1953 005516 105720      2$: TSTB     (RO)+      ;; FIND END OF MESSAGE
1954 005520 001376      BNE      2$
1955 005522 163700 001250      SUB      $MSGAD,RO    ;; SUB START OF MESSAGE
1956 005526 006200      ASR      RO          ;; GET MESSAGE LGTH IN WORDS
1957 005530 010037 001252      MOV      RO,$MSGLGT   ;; PUT LENGTH IN MAILBOX
1958 005534 012737 000004 001234  MOV      #4,$MSGTYPE  ;; TELL APT TO TAKE MSG.
1959 005542 000413      BR       5$
1960 005544 017637 000004 005570  3$: MOV      24(SP),4$   ;; PUT MSG ADDR IN JSR LINKAGE
1961 005552 062766 000002 000004  ADD      #2,4(SP)     ;; BUMP RETURN ADDRESS
1962 005560 013746 177776      MOV      177776,-(SP) ;; PUSH 177776 ON STACK
1963 005564 004737 005132      JSR     PC,$TYPE     ;; CALL TYPE MACRO
1964 005570 000000      4$: .WORD    0
1965 005572      5$:
1966 005572 105737 005660      10$: TSTB     $FFLG     ;; SHOULD REPORT FATAL ERROR?
1967 005576 001416      BEQ      12$        ;; IF NOT: BR
1968 005600 005737 001254      TST     $SENV        ;; RUNNING UNDER APT?
1969 005604 001413      BEQ      12$        ;; IF NOT: BR
1970 005606 005737 001234      11$: TST     $MSGTYPE   ;; FINISHED LAST MESSAGE?
1971 005612 001375      BNE      11$        ;; IF NOT: WAIT
1972 005614 017637 000004 001236  MOV      24(SP),$FATAL ;; GET ERROR #
1973 005622 062766 000002 000004  ADD      #2,4(SP)     ;; BUMP RETURN ADDR.
1974 005630 005237 001234      INC     $MSGTYPE     ;; TELL APT TO TAKE ERROR
1975 005634 105037 005660      12$: CLRB     $FFLG     ;; CLEAR FATAL FLAG
1976 005640 105037 005657      CLRB     $LFLG     ;; CLEAR LOG FLAG
1977 005644 105037 005656      CLRB     $MFLG     ;; CLEAR MESSAGE FLAG
1978 005650 012601      MOV      (SP)+,R1    ;; POP STACK INTO R1
1979 005652 012600      MOV      (SP)+,RO    ;; POP STACK INTO RO
1980 005654 000207      RTS     PC          ;; RETURN
1981 005656      000      $MFLG: .BYTE    0      ;; MESSG. FLAG
1982 005657      000      $LFLG: .BYTE    0      ;; LOG FLAG
1983 005660      000      $FFLG: .BYTE    0      ;; FATAL FLAG
1984      005662      .EVEN
1985      000200  APTSIZE=200
1986      000001  APTENV=001
1987      000100  APTPOOL=100
1988      000040  APTCSUP=040
1989
1990      ;STRING INPUT ROUTINE
1991      -----
1992
1993 005662 010346      .INSTR: MOV      R3,-(SP)  ;; SAVE R3 ON STACK
1994 005664 010446      MOV      R4,-(SP)  ;; SAVE R4 ON STACK

```

1995	005666	017637	000004	005704		MOV	04(SP),.MSG	:GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
1996	005674	062766	000002	000004		ADD	#2,4(SP)	:POINT TO INSTRUCTION AFTER ADDRESS POINTER
1997	005702	104402			.INST1:	TYPE		:PRINT THE MESSAGE
1998	005704	000000			.MSG:	0		:MESSAGE IS POINTED TO FROM HERE
1999	005706	012704	010452			MOV	#INBUF,R4	:POINT R4 TO THE INPUT BUFFER
2000	005712	012703	000007			MOV	#7,R3	:SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
2001	005716	105777	173242		15:	TSTB	0STKS	:HAS A CHARACTER BEEN RECEIVED?
2002	005722	100375				BPL	15	:IF NO, KEEP WAITING FOR IT
2003	005724	117714	173236			MOVB	0STKB,(R4)	:IF YES, SAVE IT IN THE INPUT BUFFER
2004	005730	142714	000200			BICB	#200,(R4)	:KEEP ONLY THE 7-BIT ASCII INFORMATION
2005	005734	122427	000015			CMPB	(R4)+,#15	:IS THIS CHARACTER A LINE FEED?
2006	005740	001417				BEQ	INSTR2	:IF SO, TERMINATE THE INPUT SEQUENCE
2007	005742	105777	173222		25:	TSTB	0STPS	:IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
2008	005746	100375				BPL	25	:IF WE CAN'T, WAIT UNTIL WE CAN
2009	005750	017777	173212	173214		MOV	0STKB,0STPB	:ECHO THE CHARACTER BACK
2010	005756	005303				DEC	R3	:REDUCE THE NUMBER OF CHARACTERS RECEIVED
2011	005760	001356				BNE	15	:IF WE DON'T HAVE 7, GO GET SOME MORE
2012	005762	012604				MOV	(SP)+,R4	:IF WE HAVE 7, RESTORE R4
2013	005764	012603				MOV	(SP)+,R3	:RESTORE R3
2014	005766	010346			.INSTE:	MOV	R3,-(SP)	:SAVE R3 ON THE STACK
2015	005770	010446				MOV	R4,-(SP)	:SAVE R4 ON THE STACK
2016	005772	104402	001230			TYPE	,SQUES	:PRINT A QUESTION MARK... WHAT'S GOING ON?
2017	005776	000741				BR	.INST1	:GO PRINT THE MESSAGE AGAIN
2018	006000	012604			INSTR2:	MOV	(SP)+,R4	:RESTORE R4
2019	006002	012603				MOV	(SP)+,R3	:RESTORE R3
2020	006004	000002				RTI		:RETURN TO THE MAIN PROCEDURE
2021								
2022								
2023								
2024								
2025	006006	010546			.PARAM:	MOV	R5,-(SP)	:SAVE R5 ON THE STACK
2026	006010	010446				MOV	R4,-(SP)	:SAVE R4 ON THE STACK
2027	006012	016605	000004			MOV	4(SP),R5	:GET THE SETUP INFORMATION POINTER
2028	006016	012537	006176			MOV	(R5)+,LOLIM	:SET THE LOW LIMIT FOR THE INPUT
2029	006022	012537	006200			MOV	(R5)+,HILIM	:SET THE HIGH LIMIT FOR THE INPUT
2030	006026	012537	006202			MOV	(R5)+,DEVADR	:SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
2031	006032	112537	006204			MOVB	(R5)+,LOBIT'S	:GET THE MASK OF THE INCORRECT BITS
2032	006036	112537	006205			MOVB	(R5)+,ADRCNT	:GET THE COUNT OF THE ITEMS TO BE STORED
2033	006042	010566	000004			MOV	R5,4(SP)	:POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
2034	006046	005005			PARAM1:	CLR	R5	:INITIALIZE THE ASCII TO OCTAL RESULT WORD
2035	006050	012704	010452			MOV	#INBUF,R4	:POINT TO THE INPUT BUFFER
2036	006054	122714	000015			CMPB	#15,(R4)	:IS THIS CHARACTER A CARRIAGE RETURN?
2037	006060	001420				BEQ	PARERR	:IF SO, PRINT THE MESSAGE AGAIN
2038	006062	121427	000060		15:	CMPB	(R4),#60	:IS THIS CHARACTER BELOW THE NUMERIC RANGE?
2039	006066	002415				BLT	PARERR	:IF SO, GO PRINT THE MESSAGE AGAIN
2040	006070	121427	000067			CMPB	(R4),#67	:IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
2041	006074	003012				BGT	PARERR	:IF SO, GO PRINT THE MESSAGE AGAIN
2042	006076	142714	000060			BICB	#60,(R4)	:ISOLATE THE NUMBER THE CHARACTER REPRESENTS
2043	006102	152405				BISB	(R4)+,R5	:CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
2044	006104	122714	000015			CMPB	#15,(R4)	:IS THE NEXT CHARACTER A CARRIAGE RETURN?
2045	006110	001406				BEQ	LIMITS	:IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
2046	006112	006305				ASL	R5	:CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
2047	006114	006305				ASL	R5	:CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
2048	006116	006305				ASL	R5	:MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
2049								:NEXT THREE BITS
2050	006120	000760				BR	15	:GO GET THE NEXT CHARACTER

:CONVERT ASCII STRING TO OCTAL

```

2051 006122 104404          PARERR: INSTER          ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
2052 006124 000750          BR          PARAM1          ;TRY GETTING THE PARAMETERS AGAIN
2053
2054                          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2055                          -----
2056
2057 006126 020537 006200    LIMITS: CMP          R5,HILIM          ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
2058 006132 101373          BHI          PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
2059 006134 020537 006176    CMP          R5,LOLIM          ;IS THE RESULT LOWER THAN ALLOWED?
2060 006140 103770          BLO          PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
2061 006142 133705 006204    BITB        LOBITS,R5          ;ARE ANY INCORRECT BITS SET IN THE RESULT?
2062 006146 001365          BNE          PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
2063
2064                          ;STORE NUMBER AT SPECIFIED ADDRESS
2065
2066 006150 013704 006202    1$: MOV          DEVADR,R4          ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
2067 006154 010524          MOV          R5,(R4)+          ;STORE THE RESULT
2068 006156 062705 000002    ADD          #2,R5              ;CALCULATE THE NEXT DATUM
2069 006162 105337 006205    DECB        ADCNT              ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
2070 006166 001372          BNE          1$                ;IF NOT, GO STORE THE NEXT DATUM
2071 006170 012604          MOV          (SP)+,R4          ;RESTORE R4
2072 006172 012605          MOV          (SP)+,R5          ;RESTORE R5
2073 006174 000002          RTI                          ;RETURN TO THE MAIN PROGRAM
2074
2075 006176 000000          LOLIM: 0                      ;LOWEST ACCEPTABLE VALUE
2076 006200 000000          HILIM: 0                      ;HIGHEST ACCEPTABLE
2077 006202 000000          DEVADR: 0                     ;LOCATION WHERE RESULT WILL BE STORED
2078 006204          000                          ;INCORRECT BITS MASK
2079 006205          000                          ;COUNT OF ITEMS TO BE STORED
2080
2081                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
2082                          -----
2083
2084 006206 016637 000004 001402 .SAV05: MOV          4(SP),SAVPC          ;SAVE R7 (PC)
2085
2086                          ;SAVE R0-R5
2087
2088 006214 010537 001214    SV05: MOV          R5,$REG5          ;SAVE R5
2089 006220 010437 001212    MOV          R4,$REG4          ;SAVE R4
2090 006224 010337 001210    MOV          R3,$REG3          ;SAVE R3
2091 006230 010237 001206    MOV          R2,$REG2          ;SAVE R2
2092 006234 010137 001204    MOV          R1,$REG1          ;SAVE R1
2093 006240 010037 001202    MOV          R0,$REG0          ;SAVE R0
2094 006244 000002          RTI                          ;LEAVE.
2095
2096                          ;RESTORE R0-R5
2097
2098 006246 013700 001202    .RES05: MOV          $REG0,R0          ;RESTORE R0
2099 006252 013701 001204    MOV          $REG1,R1          ;RESTORE R1
2100 006256 013702 001206    MOV          $REG2,R2          ;RESTORE R2
2101 006262 013703 001210    MOV          $REG3,R3          ;RESTORE R3
2102 006266 013704 001212    MOV          $REG4,R4          ;RESTORE R4
2103 006272 013705 001214    MOV          $REG5,R5          ;RESTORE R5
2104 006276 000002          RTI                          ;LEAVE
2105
2106                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

```

```

2107
2108
2109 006300 104402 001231 .CONVR: TYPE SCLRF ;PRINT A CARRIAGE RETURN
2110 006304 010046 .CNVRT: MOV R0,-(SP) ;SAVE R0
2111 006306 010146 MOV R1,-(SP) ;SAVE R1
2112 006310 010346 MOV R3,-(SP) ;SAVE R3
2113 006312 010446 MOV R4,-(SP) ;SAVE R4
2114 006314 010546 MOV R5,-(SP) ;SAVE R5
2115 006316 017601 000012 MOV @12(SP),R1 ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
2116 006322 062766 000002 000012 ADD #2,12(SP) ;POINT TO WHERE MAIN PROGRAM WILL RESUME
2117 006330 012137 006454 MOV (R1)+,WRDCNT ;GET NUMBER OF WORDS TO BE PRINTED
2118 006334 112105 1S: MOV (R1)+,R5 ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
2119 006336 112100 MOV (R1)+,R0 ;GET THE NUMBER OF SPACES TO PRINT
2120 006340 013104 MOV @12(SP),R4 ;COPY THE WORD TO BE CONVERTED
2121 006342 110537 006456 MOV R5,CHRCNT ;COPY THE CHARACTER COUNT
2122 006346 010403 3S: MOV R4,R3 ;COPY THE ARGUMENT WORD AGAIN
2123 006350 042703 177770 BIC #1C<7>,R3 ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
2124 006354 062703 000060 ADD #060,R3 ;MAKE AN ASCII CHARACTER OUT OF THEM
2125 006360 110346 MOV R3,-(SP) ;SAVE THAT CHARACTER
2126 006362 006004 ROR R4 ;MOVE THE NEXT THREE BITS INTO PLACE
2127 006364 006204 ASR R4 ;MOVE THEM AGAIN
2128 006366 006204 ASR R4 ;AND FINALLY A THIRD TIME
2129 006370 005305 DEC R5 ;REDUCE CHARACTER COUNT. ARE ALL CHARACTERS
2130 BNE 3S ;BUILT?
2131 006372 001365 BNE 3S ;IF NO, GO BUILD THE NEXT ONE.
2132 006374 012703 010556 MOV #MDATA,R3 ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
2133 006400 112623 4S: MOV (SP)+,(R3)+ ;STORE THE CHARACTER, STARTING WITH THE MOST
2134 006402 105337 006456 DECB CHRCNT ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
2135 006406 001374 BNE 4S ;IF NO, GO TRANSFER ANOTHER
2136 006410 105700 TSTB R0 ;ARE ANY SPACES TO BE PRINTED?
2137 006412 001404 BEQ 6S ;IF NO, DON'T SET UP ANY
2138 006414 112723 000040 5S: MOV #040,(R3)+ ;ADD A SPACE TO THE OUTPUT BUFFER
2139 006420 105300 DECB R0 ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
2140 006422 001374 BNE 5S ;IF YES, GO ADD ANOTHER SPACE
2141 006424 105013 6S: CLRB (R3) ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
2142 006426 104402 010556 TYPE ,MDATA ;PRINT THE STRING WE JUST BUILT
2143 006432 005337 006454 DEC WRDCNT ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
2144 006436 001336 BNE 1S ;IF YES, GO CONVERT THEM
2145 006440 012605 MOV (SP)+,R5 ;RESTORE R5
2146 006442 012604 MOV (SP)+,R4 ;RESTORE R4
2147 006444 012603 MOV (SP)+,R3 ;RESTORE R3
2148 006446 012601 MOV (SP)+,R1 ;RESTORE R1
2149 006450 012600 MOV (SP)+,R0 ;RESTORE R0
2150 006452 000002 RTI ;RETURN TO THE MAIN PROGRAM
2151 006454 000000 WRDCNT: 0 ;NUMBER OF CHARACTERS TO PRINT
2152 006456 000 CHRCNT: .BYTE ;NUMBER OF SPACES TO PRINT
2153 006457 000 SPACNT: .BYTE 0
2154
2155 006460 000000 BINWRD: 0
2156
2157
2158 ;TRAP DISPATCH SERVICE
2159 ;ARGUMENT OF TRAP IS EXTRACTED
2160 ;AND USED AS OFFSET TO OBTAIN POINTER
2161 ;TO SELECTED SUBROUTINE
2162

```



```

2163 006462 010046      .TRPSR: MOV      RO,-(SP)      ;SAVE RO, USE RO TO FIND TRAP ROUTINE
2164 006464 016600 000002      MOV      2(SP),RO      ;GET TRAP ADDRESS
2165 006470 005740      TST      -(RO)        ;GET TRAP
2166 006472 111000      MOV      (RO),RO      ;GET RIGHT BYTE OF TRAP (TRAP OFFSET)
2167 006474 006300      ASL      RO          ;POSITION OFFSET FOR TABLE INDEXING
2168 006476 016000 002002      MOV      .TRPTAB(RO),RO ;PLACE INDEXED ADDRESS OF TABLE IN RO
2169 006502 000200      RTS      RO          ;TRANSFER TO THAT ADDRESS AND RESTORE OLD RO
2170
2171      ;DEVICE CLEAR ROUTINE
2172      ;ISSUE A DEVICE CLEAR
2173      -----
2174 006504      .DEVICE.CLR:
2175 006504 052777 000020 173330      BIS      #DCLR, @DZCSR ;SET DCLR
2176 006512 032777 000020 173322      IS:     BIT      #DCLR, @DZCSR ;DID IT CLEAR?
2177 006520 001374      BNE      IS          ;BR IF NO
2178 006522 000002      RTI          ;EXIT ROUTINE
2179
2180      ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
2181      -----
2182 006524 104413      .DCLASM: DEVICE.CLR ;ISSUE A DEVICE CLEAR
2183 006526 153777 001417 173306      BIS      MNTFLG, @DZCSR ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
2184 006534 000002      RTI          ;RETURN TO CALLING ROUTINE
2185
2186 006536      .DELAY:
2187 006536 010046      MOV      RO,-(SP)      ;SAVE RO
2188 006540 013700 006554      MOV      DLYCNT,RO     ;SET COUNT
2189 006544 005300      IS:     DEC      RO      ;DELAY
2190 006546 001376      BNE      IS          ;
2191 006550 012600      MOV      (SP)+,RO     ;RESTORE RO
2192 006552 000002      RTI          ;LEAVE ROUTINE
2193 006554 000036      DLYCNT: .WORD 30.     ;PATCHABLE LOC FOR MORE TIME
2194
2195      ;ADVANCE TO NEXT TEST HANDLER
2196      -----
2197
2198 006556 013716 001360      .ADVANCE: MOV      NEXT,(SP) ;CRUNCH STACK WITH ADDRESS OF SCOPE CALL
2199 006562 005037 001362      CLR      LOCK        ;RESET TIGHT LOOP ADDRESS
2200 006566 000002      RTI          ;CHECK TO SEE IF OLD TEST GETS REPEATED
2201
2202      ;ERROR HANDLER
2203      -----
2204
2205 006570 004737 007212      $ERROR: JSR      PC, SERV.G ;FIND OUT IF <↑G> WAS HIT
2206 006574 032777 010000 172356      BIT      #SW12, @SWR   ;BELL ON ERROR?
2207 006602 001406      BEQ      XBX         ;BR IF NO BELL
2208 006604 105777 172360      TSTB    @STPS        ;TTY READY.
2209 006610 100003      BPL      XBX         ;DON'T WAIT IF TTY NOT READY.
2210 006612 112777 000207 172352      MOV      #207, @STPB  ;PUSH A BELL AT THE TTY.
2211 006620 032777 020000 172332      XBX:     BIT      #SW13, @SWR ;DELETE ERROR PRINT OUT?
2212 006626 001113      BNE      HALTS       ;BR IF NO PRINT OUT WANTED.
2213 006630 021637 001136      CMP      (SP), $ERRPC ;WAS THIS ERROR FOUND LAST TIME?
2214 006634 001404      BEQ      IS          ;BR IF YES
2215 006636 011637 001136      MOV      (SP), $ERRPC ;RECORD BEING HERE
2216 006642 105037 001123      CLRB    $ERFLG      ;PREPARE HEADER
2217 006646 104407      IS:     SAVOS      ;SAVE ALL PROC REGISTERS
2218 006650 011605      MOV      (SP), R5    ;GET THE PC OF ERROR

```

2219	006652	162705	000002		SUB	#2,R5		;GET ADDRESS OF TRAP CALL
2220	006656	011504			MOV	(R5),R4		;GET ERROR INSTRUCTION
2221	006660	110437	001134		MOVB	R4,\$ITEMB		;COPY TEST NUMBER FOR APT HANDLING
2222	006664	006304			ASL	R4		;MULT BY TWO
2223	006666	061504			ADD	(R5),R4		;DOUBLE IT
2224	006670	006304			ASL	R4		;MULT AGAIN
2225	006672	042704	177001		BIC	#177001,R4		;CLEAR JUNK
2226	006676	062704	026174		ADD	#.ERRTAB,R4		;GET POINTER
2227	006702	012437	007026		MOV	(R4)+,ERRMSG		;GET ERROR MESSAGE
2228	006706	012437	007040		MOV	(R4)+,DATAHD		;GET DATA HEADRER
2229	006712	011437	007052		MOV	(R4),DATABP		;GET DATA TABLE
2230	006716	105737	001123		TSTB	SERFLG		;TYPE HEADER
2231	006722	001403			BEQ	TYPMSG		;BR IF YES
2232	006724	005737	007052		TST	DATABP		;DOES DATA TABLE EXIST?
2233	006730	001044			BNE	TYPDAT		;BR IF YES.
2234	006732	104402	001231	TYPMSG:	TYPE	,SCRLF		;TYPE A CARRIAGE RETURN
2235	006736	104402	001231		TYPE	,SCRLF		;AND TYPE ANOTHER
2236	006742	005737	001362		TST	LOCK		
2237	006746	001402			BEQ	1\$		
2238	006750	104402	010147		TYPE	,MASTEK		
2239	006754	104402	010135	1\$:	TYPE	,MTSTN		
2240	006760	104412	007204		CNVRT	,XTSTN		;SHOW IT
2241	006764	104402	010224		TYPE	,MERRPC		;TYPE PC.
2242	006770	104412	007176		CNVRT	,ERTABO		;SHOW IT
2243	006774	104402	010077		TYPE	,MCSRX		
2244	007000	104412	004612		CNVRT	,XCSR		
2245	007004	104402	001231		TYPE	,SCRLF		;GIVE A CR/LF
2246	007010	112737	177777	001123	MOVB	#-1,SERFLG		;NO MORE HEADER UNLESS NO DATA TABLE.
2247	007016	005737	007026		TST	ERRMSG		;IS THERE AN ERROR MESSAGE?
2248	007022	001402			BEQ	WTBS.FM		;BR IF NO.
2249	007024	104402			TYPE			;TYPE
2250	007026	000000		ERRMSG:	0			;ERROR MESSAGE
2251	007030			WTBS.FM:				
2252	007030	005737	007040		TST	DATAHD		;DATA HEADER?
2253	007034	001402			BEQ	TYPDAT		;BR IF NO
2254	007036	104402			TYPE			;TYPE
2255	007040	000000		DATAHD:	0			;DATA HEADER
2256	007042	005737	007052	TYPDAT:	TST	DATABP		;DATA TABLE?
2257	007046	001402			BEQ	RESREG		;BR IF NO.
2258	007050	104411			CONVRT			;SHOW
2259	007052	000000		DATABP:	0			;DATA TABLE
2260	007054	104410		RESREG:	RES05			;RESTORE PROC REGISTERS
2261	007056	122737	000001	001254	HALTS:	CMPB	#APTENV,SENV	;IS APT RUNNING?
2262	007064	001007			BNE	2\$;SKIP APT CALL IF NOT
2263	007066	113737	001134	007100	MOVB	\$ITEMB,7\$;COPY ERROR NUMBER
2264	007074	004737	005432		JSR	PC,\$ATY4		;CALL APT SERVICE
2265	007100	000000		7\$:	.WORD	0		;ERROR NUMBER STUCK HERE
2266	007102	000777		8\$:	BR	8\$;LOCK UP HERE
2267	007104	022737	004576	000042	2\$:	CMP	#SENDAD,\$#42	;CHECK TO SEE IF IN ACT-11 MODE
2268	007112	001403			BEQ	1\$;IF SO, HANDLE ACCORDINGLY
2269	007114	005777	172040		TST	\$SWR		;HALT ON ERROR?
2270	007120	100004			BPL	EXITER		;BR IF NO HALT ON ERROR
2271	007122	016677	000002	172032	1\$:	MOV	2(SP),\$DISPLAY	;SHOW ERROR PC IN DATA DISPLAY
2272	007130	000000			HALT			;HALT
2273	007132	005237	001132		EXITER:	INC	SERTTL	;UPDATE ERROR COUNT
2274	007136	032777	000400	172014	BIT	#SW08,\$SWR		;GOTO TOP OF TEST?


```

2331 007452 020200 051450 051127 89$: .ASCIZ <200>? (SWR)=/?
2332 007460 036451 000057
2333
2334 007464 000001 .EVEN
2335 007466 006 000 88$: 1
2336 007470 007472 .BYTE 6,0
2337 007472 000000 90$: .WORD 0
2338 007474 036457 000057 91$: .ASCIZ ?/=/?
2339
2340 .EVEN
2341 .SBTTL POWER DOWN AND UP ROUTINES
2342
2343 ::*****
2344 :POWER DOWN ROUTINE
2345 $PWRDN: MOV $SILLUP, @#PWRVEC ;; SET FOR FAST UP
2346 MOV #340, @#PWRVEC+2 ;; PRIO:7
2347 MOV RO, -(SP) ;; PUSH RO ON STACK
2348 MOV R1, -(SP) ;; PUSH R1 ON STACK
2349 MOV R2, -(SP) ;; PUSH R2 ON STACK
2350 MOV R3, -(SP) ;; PUSH R3 ON STACK
2351 MOV R4, -(SP) ;; PUSH R4 ON STACK
2352 MOV R5, -(SP) ;; PUSH R5 ON STACK
2353 MOV @SWR, -(SP) ;; PUSH @SWR ON STACK
2354 MOV SP, $SAVR6 ;; SAVE SP
2355 MOV $SPWRUP, @#PWRVEC ;; SET UP VECTOR
2356 HALT
2357 BR -.2 ;; HANG UP
2358
2359 ::*****
2360 :POWER UP ROUTINE
2361 $PWRUP: MOV $SILLUP, @#PWRVEC ;; SET FOR FAST DOWN
2362 MOV $SAVR6, SP ;; GET SP
2363 CLR $SAVR6 ;; WAIT LOOP FOR THE TTY
2364 1$: INC $SAVR6 ;; WAIT FOR THE INC
2365 BNE 1$ OF WORD
2366 MOV (SP)+, @SWR ;; POP STACK INTO @SWR
2367 MOV (SP)+, R5 ;; POP STACK INTO R5
2368 MOV (SP)+, R4 ;; POP STACK INTO R4
2369 MOV (SP)+, R3 ;; POP STACK INTO R3
2370 MOV (SP)+, R2 ;; POP STACK INTO R2
2371 MOV (SP)+, R1 ;; POP STACK INTO R1
2372 MOV (SP)+, RO ;; POP STACK INTO RO
2373 MOV $SPWRDN, @#PWRVEC ;; SET UP THE POWER DOWN VECTOR
2374 MOV #340, @#PWRVEC+2 ;; PRIO:7
2375 TYPE REPORT THE POWER FAILURE
2376 SPWRMG: .WORD MPFAIL ;; POWER FAIL MESSAGE POINTER
2377 MOV (PC)+, (SP) ;; RESTART AT RESTART
2378 SPWRAD: .WORD RESTART ;; RESTART ADDRESS
2379 RTI
2380 $SILLUP: HALT ;; THE POWER UP SEQUENCE WAS STARTED
2381 BR -.2 ;; BEFORE THE POWER DOWN WAS COMPLETE
2382 $SAVR6: 0 ;; PUT THE SP HERE
2383 MPFAIL: .ASCIZ <200>/PWR FAILED. RESTART AT LAST TEST /
(2) 007715 200 047105 020104 MEPASS: .ASCIZ <200>/END PASS DZDZA-B /
(2) 007741 200 052522 047116 MR: .ASCIZ <200>/RUNNING /
(2) 007755 200 051120 043517 MERR2: .ASCIZ <200>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 010024 044600 051516 043125 MERR3: .ASCIZ <200>/INSUFFICIENT DATA!/

```

```

(2) 010050 046200 041517 020113 MLOCK: .ASCIZ <200>/LOCK ON SELECTED TEST/
(2) 010077 103 051123 020072 MCSRX: .ASCIZ /CSR: /
(2) 010105 126 041505 020072 MVECX: .ASCIZ /VEC: /
(2) 010113 120 051501 042523 MPASSX: .ASCIZ /PASSES: /
(2) 010124 051105 047522 051522 MERRX: .ASCIZ /ERRORS: /
(2) 010135 124 051505 020124 MTSTN: .ASCIZ /TEST NO: /
(2) 010147 052 000040 MASTEK: .ASCIZ /* /
(2) 010152 051600 052105 051440 MNEW: .ASCIZ <200>/SET SWITCH REG TO DZ11'S DESIRED ACTIVE./
(2) 010224 041520 020072 000 MERRPC: .ASCIZ /PC: /
(2) 010231 200 040515 020120 XHEAD: .ASCIZ <200>/MAP OF DZ11 STATUS/<200>
(2) 010256 044600 046114 043505 MBADLN: .ASCIZ <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
(2) .EVEN
(2) 010320 000002 XSTATQ: 2
2383 010322 006 003 .BYTE 6,3
2384 010324 001220 $TMP1
2385 010326 006 002 .BYTE 6,2
2386 010330 001222 $TMP2
2387 .EVEN
2388 ;THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
2389 -----
2390 ;E=EXTERNAL LOOP BACK
2391 ;I=INTERNAL LOOP BACK
2392 ;S=STAGGERED LOOP BACK
2393 010332 017605 000000 .SETFLG: MOV 0(SP),R5 ;PICK UP ADDRESS OF TAG
2394 010336 042737 000040 010452 BIC #40,INBUF ;STRIP LOWER CASE
2395 010344 122737 000105 010452 CMPB #'E,INBUF ;IS IT EXTERNAL LOOP BACK ?
2396 010352 001005 BNE 4$ ;NO
2397 010354 013715 010444 MOV 15,(R5) ;YES STORE INFO
2398 010360 105037 001417 CLRB MNTFLG ;SET MAINT BIT =0
2399 010364 000422 BR 7$ ;GET OUT
2400 010366 122737 000111 010452 4$: CMPB #'I,INBUF ;IS IT INTERNAL LOOP BACK ?
2401 010374 001006 BNE 5$ ;NO
2402 010376 013715 010446 MOV 25,(R5) ;YES STORE INFO
2403 010402 112737 000010 001417 MOVB #MAINT,MNTFLG ;SET UP THE MAINTENANCE FLAG LOADER
2404 010410 000410 BR 7$ ;GET OUT
2405 010412 122737 000123 010452 5$: CMPB #'S,INBUF ;IS IT STAGGERED LOOP BACK ?
2406 010420 001007 BNE 6$ ;WHAT ?
2407 010422 013715 010450 MOV 35,(R5) ;YES STORE INFO
2408 010426 105037 001417 CLRB MNTFLG ;ZERO BITS
2409 010432 062716 000002 7$: ADD #2,(SP) ;POP AROUND
2410 010436 000002 RTI
2411 010440 104404 6$: INSTER ;RETRY
2412 010442 000733 BR .SETFLG ;DITTO
2413 010444 000200 1$: .WORD 200 ;EXTERNAL = E
2414 010446 000000 2$: .WORD 0 ;INTERNAL = I
2415 010450 100000 3$: .WORD 100000 ;STAGGERED = S
2416
2417 ;BUFFERS FOR INPUT-OUTPUT
2418
2419 010452 000000 INBUF: 0
2420 010514 .=. +40
2421 010514 000000 TEMP: 0
2422 010556 .=. +40
2423 010556 000000 MDATA: 0
2424 010620 .=. +40
2425

```

20	010620	011637	010716		SET.PS:	MOV	(SP),3S	
21	010624	162737	000002	010716		SUB	#2,3S	
22	010632	017737	000060	010720		MOV	23S,4S	
23	010640	022737	106427	010720		CMP	#106427,4S	
24	010646	001003				BNE	1S	
25	010650	011637	010716			MOV	(SP),3S	
26	010654	000412				BR	2S	
27	010656	022737	106437	010720	1S:	CMP	#106437,4S	
28	010664	001401				BEQ	.+4	
29	010666	000000				HALT		;RESERVED INSTRUCTION NOT "MTPS"
30	010670	011637	010716			MOV	(SP),3S	
31	010674	017737	000016	010716		MOV	23S,3S	
32	010702	062716	000002		2S:	ADD	#2,(SP)	
33	010706	017766	000004	000002		MOV	23S,2(SP)	
34	010714	000002				RTI		
35	010716	000000			3S:	O		
36	010720	000000			4S:	O		

473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498

```

010722 005737 001404          CYCLE: TST     DZACTV
010726 001004                BNE     IS
010730 104402 007755          TYPE   ,MERR2
010734 000000                HALT
010736 000776                BR     -2
010740 013737 005104 001226 1S:  MOV     $MXCNT,STIMES
010746 033737 001406 001404  BIT     RUN,DZACTV
010754 001017                BNE     2S
010756 006137 001406      ROL     RUN
010762 005537 001406      ADC     RUN
010766 062737 000014 001412  ADD     #14,ACTIVE
010774 022737 002000 001412  CMP     #DZ.END,ACTIVE
011002 001356                BNE     IS
011004 012737 001500 001412  MOV     #DZ.MAP,ACTIVE
011012 000752                BR     IS
011014 006137 001406      ROL     RUN
011020 005537 001406      ADC     RUN
011024 013700 001412      MOV     ACTIVE,RO
011030 062737 000014 001412  ADD     #14,ACTIVE
011036 022737 002000 001412  CMP     #DZ.END,ACTIVE

011044 001003                BNE     3S
011046 012737 001500 001412  MOV     #DZ.MAP,ACTIVE
011054 012037 001310      MOV     (RO)+,$BASE
011060 012037 002072      MOV     (RO)+,DZRIV
011064 012037 026170      MOV     (RO)+,DZPRT
011070 113737 026171 001414  MOVB   DZPRT+1,EIAFLG
011076 042737 100000 026170  BIC    #BIT15,DZPRT
011104 012037 001364      MOV     (RO)+,LINE
011110 012037 001366      MOV     (RO)+,PAR
011114 012037 001370      MOV     (RO)+,MODE
011120 004737 025762      JSR    PC,DZLEV
011124 005737 000042      TST    #42
011130 001046                BNE     4S
011132 032777 000002 170020  BIT    #SMD1,$SWR
011140 001442                BEQ    4S
011142 104402 001231      7S:  TYPE   ,SCLF
011146 104403                INSTR
011150 010135                HTSTN
011152 104405                PARAM
011154 000001                1
011156 001000                1000
011160 001122                $STNM
011162 000                .BYTE 0
011163 001                .BYTE 1
011164 012700 012166      MOV     #TST1,RO
011170 022710 000004      5S:  CMP     #4,(RO)
  
```

```

:ROUTINE USED TO "CYCLE" THROUGH UP TO SIXTEEN DZ11'S
:THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
:AND RUNS THE SPECIFIED DZ11'S. THIS ROUTINE *MUST*
:BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
:SETUP NECESSARY.
:
  
```

```

:ARE ANY DZ11'S TO BE TESTED?
:BR IF OK.
:NO DZ11'S SELECTED!!
:STOP THE SHOW.
:DISQUALIFY CONT. SW.
:RESTORE THE NUMBER OF ITERATIONS TO MAKE
:IS THIS ONE "ACTIVE"
:BR IF GOOD ONE FOUND.
:UPDATE POINTER
:CATCH CARRY FROM RUN
:UPDATE ADDRESS POINTER.
:HAVE WE PASSED THE END OF THE MAP?
:IF NO, KEEP GOING; NOT ALL TESTED FOR.
:RESET ADDRESS POINTER.
:KEEP LOOKING FOR ACTIVE DZ11
:UPDATE POINTER.
:CATCH CARRY.
:GET ADDRESS POINTER.
:UPDATE.

:ALL DONE?
:BR IF NO.
:RESTORE POINTER.
:LOAD SYSTEM CTRL. REG
:LOAD VECTOR
:LOAD PRIORITY
:EIA OR 20MA
:CLEAR FLAG
:SET UP LINE DZ LINES ACTIVE
:SET UP PARAMETERIZATION
:SET UP MAINTENANCE MODE
:SET UP
:ARE WE UNDER MONITOR CONTROL?
:IF YES, SKIP THIS SETUP
:IF SMD1=1, GET STARTING TEST #
:BR IF NO TEST IS TO BE INPUTTED

:CALL THE STRING INPUT ROUTINE
:POINTER TO MESSAGE TO BE PRINTED
:CALL THE OCTAL TO ASCII CONVERT ROUTINE
:LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
:HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
:POINTER TO MAP LOCATION TO BE FILLED
:MASK OF INVALID BITS FOR THIS PARAMETER
:NUMBER OF PARAMETERS TO STORE
  
```

2499	011174	001015			BNE	6S	
2500	011176	022760	012737	000002	CMP	#12737,2(RO)	
2501	011204	001011			BNE	6S	
2502	011206	023760	001122	000004	CMP	\$TSTNM,4(RO)	: IS THIS THE TEST ?
2503	011214	001005			BNE	6S	: IF NOT, DON'T PROCESS NUMBER
2504	011216	010037	001126		MOV	RO,\$LPADR	: SAVE PC
2505	011222	104402	001231		TYPE	\$CRLF	
2506	011226	000412			BR	8S	
2507	011230	005720		6S:	TST	(RO)+	
2508	011232	020027	022130		CMP	RO,#TLAST+10	
2509	011236	001354			BNE	5S	
2510	011240	104402	001230		TYPE	\$QUES	
2511	011244	000736			BR	7S	
2512	011246	012737	012166	001126	4S: MOV	#TST1,\$LPADR	: PREPARE TEST ADDRESS
2513	011254				8S:		
2514	011254	000177	167646		RESTART: JMP	\$LPADR	: GO START TESTING.***WARNING!***
2515							: THIS JUMP IS USED BY POWER UP ROUTINE!!!!
2516							


```

2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572

```

```

;--ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.
; IF BIT7 IN THE ENVIRONMENT MODE ($ENVM) BYTE IS SET,
; THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.

SETAPT: MOV    #DZ.MAP,R0      ;POINT TO THE DEVICE MAP TABLE
        MOV    $BASE,R1      ;BUILD DEVICE ADDRESSES IN R1
        MOV    $VECT1,R2     ;BUILD DEVICE VECTORS IN R2
        BIC    #1C<770>,R2   ;STRIP AWAY OTHER INFORMATION

        MOV    $VECT1+1,R3   ;LOAD THE INTERRUPT PRIORITY FROM R3
        RORB   R3            ;ALIGN THE NUMBER
        RORB   R3            ;ALIGN THE NUMBER
        RORB   R3            ;ALIGN THE NUMBER
        RORB   R3            ;ALIGN THE NUMBER
        RORB   R3            ;ALIGN THE NUMBER
        BIC    #1C<7>,R3     ;REMOVE ALL BUT BUS LEVEL NUMBER
        MOV    #SDDWD,R4     ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
        MOV    $DEVN,R5     ;GET THE MAP OF ACTIVE DEVICES
        MOV    R5,DZACTV    ;SAVE THE BIT MAP
1S:     ROR    R5            ;GET A DEVICE SELECTION BIT
        BCS   3S            ;IF IT IS SELECTED, GO SET UP A MAP
        BEQ   5S            ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
        TST   (R4)+         ;POINT TO NEXT DEVICE DESCRIPTOR
2S:     ADD   #10,R1        ;SET UP THE NEXT ADDRESS
        ADD   #10,R2        ;SET UP THE NEXT VECTOR GROUP
        BR    1S           ;GO SEE IF MORE DEVICES REMAIN
3S:     MOV   R1,(R0)+      ;LOAD DEVICE ADDRESS
        MOV   R2,(R0)+      ;LOAD THE VECTOR ADDRESS
        MOV   R3,(R0)+      ;LOAD THE INTERRUPT PRIORITY LEVEL
        MOV   $CDW1,(R0)+   ;GET THE NUMBER OF LINES IN OPERATION
        MOV   (R4)+,(R0)+   ;LOAD DEVICE PARAMETERS
        BMI   4S           ;IF 20MA MODE SELECTED, SET IT UP
        BIS   #100000,-6(R0);SET THE 20MA FLAG IN DZLVN
        BIC   #100000,-2(R0);CLEAR THE FLAG IN DZPARN
4S:     CLR   (R0)+        ;DEFAULT OPERATION TO INTERNAL MAINTENANCE MODE
        BR    2S           ;GO BUILD THE NEXT ADDRESS
5S:     MOV   #-1,(R0)      ;TERMINATE THE DEVICE MAP
        MOV   #$$WREG,SWR   ;SET TO SOFTWARE APT SWITCH REGISTER
        RTS   PC           ;RETURN TO PRINT STATUS TABLE

```

```

;#ROUTINE USED TO "AUTO SIZE" THE DZ11
;CSR AND VECTOR.
;NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
;      ADDRESS RANGE (160000:163700)
;      AND THE VECTOR MAY BE ANY WHERE IN THE
;      FLOATING VECTOR RANGE (300:770)
;

```

```

2566 011432 000005
2567 011432 105337 001415
2568 011434 012702 001500
2569 011440 012703 001320
2570 011444 005022
2571 011450 022702 002000
2572 011452

```

```

AUTO.SIZE:
        RESET
CSRMAP: DECB   INIFLG
        MOV   #DZ.MAP,R2
        MOV   #SDDWD,R3
1S:     CLR   (R2)+
        CMP   #DZ.END,R2
;INSURE A BUS INIT.
;SHOW THAT I WAS HERE
;LOAD MAP POINTER.
;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
;ZERO ENTIRE MAP
;ALL DONE?

```

2573	011456	001374			BNE	1\$;BR IF NO
2574	011460	105037	001410		CLRB	DZNUM		;SET OCTAL NUMBER OF DZ11'S TO 0
2575	011464	012702	001500		MOV	#DZ.MAP,R2		
2576	011470	012701	160000		MOV	#160000,R1		;SET FOR FIRST ADDRESS TO BE TESTED
2577	011474	012737	011770	000004	MOV	#6\$,2#4		;SET FOR NON-EXISTENT DEVICE TIME OUT
2578	011502	052711	000040		BIS	#BIT5,(R1)	2\$:	;TRY TO SET MASTER SCAN ENABLE
2579	011506	052761	000200	000004	BIS	#BIT7,4(R1)		;TRY TO TRANSMIT ON LINE 7
2580	011514	005000			CLR	RO		;USE RO AS A COUNTER
2581	011516	005711			TST	(R1)	7\$:	;HAS TRANSMITTER READY COME UP?
2582	011520	100403			BMI	8\$;IF SO, GO GET A FINAL CHECK
2583	011522	005300			DEC	RO		;REDUCE COUNT. TIME UP?
2584	011524	001374			BNE	7\$;IF NOT, KEEP WAITING
2585	011526	000451			BR	3\$;ASSUME IT'S NOT A DZ11
2586	011530	032761	000200	000004	BIT	#BIT7,4(R1)	8\$:	;IS LINE 7 ENABLE STILL SET? IT SHOULD BE
2587	011536	001445			BEQ	3\$;IF IT'S NOT, ASSUME IT'S NOT A DZ11
2588	011540	032711	000040		BIT	#BIT5,(R1)		;IS MASTER SCAN ENABLE STILL SET?
2589	011544	001442			BEQ	3\$;IF NOT, ASSUME IT'S NOT A DZ11
2590	011546	005000			CLR	RO		
2591	011550	052711	000020		BIS	#20,(R1)		;SET DEVICE CLEAR
2592	011554	032711	000020		BIT	#20,(R1)		;SHOULD STAY SET FOR A WHILE IF DZ
2593	011560	001434			BEQ	3\$;BR IF NOT DZ11
2594	011562	032711	000020		BIT	#20,(R1)		;WAIT FOR BIT TO CLEAR
2595	011566	001404			BEQ	.+12		;BR WHEN CLEARED
2596	011570	104414			DELAY			
2597	011572	005200			INC	RO		
2598	011574	001372			BNE	.-12		
2599	011576	000425			BR	3\$;BIT NOT CLEARED! MUST NOT BE DZ11
2600	011600	005011			CLR	(R1)		;GET RID OF MASTER SCAN ENABLE
2601	011602	005061	000004		CLR	4(R1)		;GET RID OF LINE 7 ENABLE
2602								;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DZ11 CSR ADDRESS.
2603	011606	010122			MOV	R1,(R2)+		;STORE CSR IN CORE TABLE.
2604	011610	005722			TST	(R2)+		;POP OVER VECTOR STORE AREA
2605	011612	012722	000005		MOV	#5,(R2)+		;SET THE DEFAULT BUS LEVEL
2606	011616	012722	000377		MOV	#377,(R2)+		;SET THE DEFAULT LINE SELECTION PARAMETER
2607	011622	012712	017470		MOV	#17470,(R2)		;SET THE DEFAULT PARAMETERS
2608	011626	012223			MOV	(R2)+,(R3)+		;COPY PARAMETERS INTO ETABLE DESCRIPTOR
2609	011630	005022			CLR	(R2)+		;SET THE DEFAULT MODE OF OPERATION
2610	011632	012712	177777		MOV	#-1,(R2)		;TERMINATE LIST
2611	011636	105237	001410		INCB	DZNUM		;UPDATE DEVICE COUNTER
2612	011642	122737	000020	001410	CMPB	#20,DZNUM		;ARE MAX. NO. OF DEV FOUND?
2613	011650	001405			BEQ	100\$;YES DON'T LOOK FOR ANY MORE.
2614	011652	062701	000010		ADD	#10,R1	3\$:	;UPDATE CSR POINTER ADDRESS
2615	011656	022701	163700		CMP	#163700,R1		
2616	011662	001307			BNE	2\$;BR IF MORE ADDRESS TO CHECK.
2617	011664						100\$:	
2618	011664	105737	001410		TSTB	DZNUM		;WERE ANY DZ11'S FOUND AT ALL?
2619	011670	001432			BEQ	5\$;ERROR AUTO SIZER FOUND NO DZ11'S IN THIS SYS.
2620	011672	113701	001410		MOV	DZNUM,R1		
2621	011676	110137	001411		MOV	R1,SAVNUM		;SAVE NUMBER OF DEVICES
2622	011702	012737	000001	001404	MOV	#1,DZACTV		
2623	011710	005301			DEC	R1	4\$:	
2624	011712	001404			BEQ	98\$		
2625	011714	000261			SEC			
2626	011716	006137	001404		ROL	DZACTV		
2627	011722	000772			BR	4\$		
2628	011724	013737	001500	001310	MOV	DZCRO,\$BASE	98\$:	;POINT TO THE ADDRESS OF FIRST DEVICE

H05

MO-11-DZDZA-8
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 60
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729

012166	000004		
012170	012737	000001	001122
012176	012737	012356	001360
012204	012737	012344	000004
012212	012737	000340	000006
012220	012737	012226	001362
012226	013700	002042	
012232	011001		
012234	000240		
012236	005010		
012240	000240		
012242	012737	012250	001362
012250	013700	002046	
012254	011001		
012256	000240		
012260	005010		
012262	000240		
012264	012737	012272	001362
012272	013700	002056	
012276	011001		
012300	000240		
012302	005010		
012304	000240		
012306	012737	012314	001362
012314	013700	002062	
012320	011001		
012322	000240		
012324	005010		
012326	000240		
012330	012737	000006	000004
012336	005037	000006	
012342	104400		
012344	011601		
012346	022626		
012350	104001		
012352	104401		
012354	000111		

```
***** TEST 1 *****
*THIS TEST PROVES THE SLAVE SYNC RESPONSE
*DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
* DZCSR, DZRBUF, DZTCR, DZMSR
::* TEST 1
*****
TST1: SCOPE
MOV #1,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST2,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV #55,4 ;SET TRAP VECTOR
MOV #PR7,6 ;SET PRIORITY TO LEVEL 7
MOV #1$,LOCK ;SET RETURN IF SW09=11
1$: MOV DZCSR,RO ;SET ADDRESS TO TEST
MOV (RO),R1 ;READ THE ADDRESS
NOP ;WASTE TIME
CLR (RO) ;WRITE THE ADDRESS
MOV #2$,LOCK ;SET RETURN ADDRESS FOR SW09
2$: MOV DZRBUF,RO ;SET ADDRESS TO TEST
MOV (RO),R1 ;READ THE ADDRESS
NOP ;
CLR (RO) ;WRITE THE ADDRESS
MOV #3$,LOCK ;SET RETURN ADDRESS FOR SW09
3$: MOV DZTCR,RO ;SET ADDRESS TO TEST
MOV (RO),R1 ;READ THE ADDRESS
NOP ;
CLR (RO) ;WRITE THE ADDRESS
MOV #4$,LOCK ;SET RETURN ADDRESS
4$: MOV DZMSR,RO ;SET ADDRESS TO TEST
MOV (RO),R1 ;READ FROM ADDRESS
NOP ;
CLR (RO) ;WRITE THE ADDRESS
MOV #6,4 ;SET TRAP CATCHER BACK TO NORMAL
CLR 6 ;
ADVANCE ;SCOPE THIS TEST
5$: MOV (SP),R1 ;SAVE PC OF TRAP
CMP (SP)+,(SP)+ ;POP TRAP OFF STACK
ERROR 1 ;*NO SLAVE SYNC RESPONSE.
SCOP1 ;SW09=1?
JMP (R1) ;RTI
***** TEST 2 *****
*THIS TEST PROVES THAT BIT "DCLR"
*CAN BE SET AND THAT IT WILL CLEAR
*BY ITSELF AFTER A PERIOD OF TIME.
::* TEST 2
*****
TST2: SCOPE
MOV #2,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST3,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DZCSR,RO ;SET POINTER
MOV #DCLR,R5 ;SET DCLR
MOV R5,(RO) ;WRITE DCLR INTO DZCSR
```



```

2786                                     ;:* TEST 4
2787                                     ;:*****
2788 012534 000004                       TST4: SCOPE
2789 012536 012737 000004 001122        MOV #4,STSTNM ;LOAD THE NUMBER OF THIS TEST
2790 012544 012737 012626 001360        MOV #TST5,NEXT ;POINT TO THE START OF THE NEXT TEST
2791 012552 013700 002042              MOV DZCSR,R0 ;GET BASE ADDRESS
2792 012556 012705 000040              MOV #MSENAB,R5 ;SET BIT
2793 012562 010510                      MOV R5,(R0) ;SET SET IN DEVICE
2794 012564 011004                      MOV (R0),R4 ;READ THE BIT FROM DEVICE
2795 012566 020504                      CMP R5,R4 ;WAS BIT SET?
2796 012570 001401                      BEQ 1$ ;BR IF YES
2797 012572 104002                      ERROR 2 ;*BIT R/W FAILURE
2798 012574 040510 1$: BIC R5,(R0) ;CLEAR THE BIT.
2799 012576 011004                      MOV (R0),R4 ;READ DEVICE
2800 012600 001404                      BEQ 2$ ;BR IF BITS WERE CLEARED.
2801 012602 010546                      MOV R5,-(SP) ;SAVE THE BIT
2802 012604 005005                      CLR R5 ;SET EXPECTED RESULTS TO 0
2803 012606 104002                      ERROR 2 ;*BIT FAILED TO CLEAR
2804 012610 012605 2$: MOV (SP)+,R5 ;RESTORE THE BIT.
2805 012612 010510                      MOV R5,(R0) ;SET THE BIT AGAIN
2806 012614 104413                      DEVICE.CLR ;ISSUE DEVICE CLEAR
2807 012616 011004                      MOV (R0),R4 ;READ THE BIT.
2808 012620 001402                      BEQ 3$ ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
2809 012622 005005                      CLR R5 ;SET EXPECTED TO ZERO
2810 012624 104002                      ERROR 2 ;*BIT NOT CLEARED BY DEVICE CLEAR
2811 012626 3$:
2812                                     ;:***** TEST 5 *****
2813                                     ;:*TEST TO VERIFY THAT BIT "SILOEN" CAN
2814                                     ;:*BE SET. THEN VERIFY THAT BIT "SILOEN" CAN
2815                                     ;:*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
2816                                     ;:*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
2817                                     ;:*CLEARED BY A "DEVICE CLEAR"
2818
2819                                     ;:* TEST 5
2820                                     ;:*****
2820 012626 000004                       TST5: SCOPE
2821 012630 012737 000005 001122        MOV #5,STSTNM ;LOAD THE NUMBER OF THIS TEST
2822 012636 012737 012720 001360        MOV #TST6,NEXT ;POINT TO THE START OF THE NEXT TEST
2823 012644 013700 002042              MOV DZCSR,R0 ;GET BASE ADDRESS
2824 012650 012705 010000              MOV #SILOEN,R5 ;SET BIT
2825 012654 010510                      MOV R5,(R0) ;SET SET IN DEVICE
2826 012656 011004                      MOV (R0),R4 ;READ THE BIT FROM DEVICE
2827 012660 020504                      CMP R5,R4 ;WAS BIT SET?
2828 012662 001401                      BEQ 1$ ;BR IF YES
2829 012664 104002                      ERROR 2 ;*BIT R/W FAILURE
2830 012666 040510 1$: BIC R5,(R0) ;CLEAR THE BIT.
2831 012670 011004                      MOV (R0),R4 ;READ DEVICE
2832 012672 001404                      BEQ 2$ ;BR IF BITS WERE CLEARED.
2833 012674 010546                      MOV R5,-(SP) ;SAVE THE BIT
2834 012676 005005                      CLR R5 ;SET EXPECTED RESULTS TO 0
2835 012700 104002                      ERROR 2 ;*BIT FAILED TO CLEAR
2836 012702 012605 2$: MOV (SP)+,R5 ;RESTORE THE BIT.
2837 012704 010510                      MOV R5,(R0) ;SET THE BIT AGAIN
2838 012706 104413                      DEVICE.CLR ;ISSUE DEVICE CLEAR
2839 012710 011004                      MOV (R0),R4 ;READ THE BIT.
2840 012712 001402                      BEQ 3$ ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
2841 012714 005005                      CLR R5 ;SET EXPECTED TO ZERO

```

2842 012716 104002
 2843 012720
 2844
 2845
 2846
 2847
 2848
 2849
 2850
 2851
 2852 012720 000004
 2853 012722 012737 000006 001122
 2854 012730 012737 013012 001360
 2855 012736 013700 002042
 2856 012742 012705 000100
 2857 012746 010510
 2858 012750 011004
 2859 012752 020504
 2860 012754 001401
 2861 012756 104002
 2862 012760 040510
 2863 012762 011004
 2864 012764 001404
 2865 012766 010546
 2866 012770 005005
 2867 012772 104002
 2868 012774 012605
 2869 012776 010510
 2870 013000 104413
 2871 013002 011004
 2872 013004 001402
 2873 013006 005005
 2874 013010 104002
 2875 013012

```

ERROR 2 ;*BIT NOT CLEARED BY DEVICE CLEAR
3$:
;***** TEST 6 *****
;*TEST TO VERIFY THAT BIT "RIE" CAN
;*BE SET. THEN VERIFY THAT BIT "RIE" CAN
;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
;*CLEARED BY A "DEVICE CLEAR"
;:* TEST 6
;*****
TST6: SCOPE
MOV #6,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST7,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DZCSR,R0 ;GET BASE ADDRESS
MOV #RIE,R5 ;SET BIT
MOV R5,(R0) ;SET SET IN DEVICE
MOV (R0),R4 ;READ THE BIT FROM DEVICE
CMP R5,R4 ;WAS BIT SET?
BEQ 1$ ;BR IF YES
ERROR 2 ;*BIT R/W FAILURE
1$: BIC R5,(R0) ;CLEAR THE BIT.
MOV (R0),R4 ;READ DEVICE
BEQ 2$ ;BR IF BITS WERE CLEARED.
MOV R5,-(SP) ;SAVE THE BIT
CLR R5 ;SET EXPECTED RESULTS TO 0
ERROR 2 ;*BIT FAILED TO CLEAR
MOV (SP)+,R5 ;RESTORE THE BIT.
2$: MOV R5,(R0) ;SET THE BIT AGAIN
DEVICE.CLR ;ISSUE DEVICE CLEAR
MOV (R0),R4 ;READ THE BIT.
BEQ 3$ ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
CLR R5 ;SET EXPECTED TO ZERO
ERROR 2 ;*BIT NOT CLEARED BY DEVICE CLEAR
3$:

```

2876
 2877
 2878
 2879
 2880
 2881
 2882
 2883
 2884 013012 000004
 2885 013014 012737 000007 001122
 2886 013022 012737 013104 001360
 2887 013030 013700 002042
 2888 013034 012705 040000
 2889 013040 010510
 2890 013042 011004
 2891 013044 020504
 2892 013046 001401
 2893 013050 104002
 2894 013052 040510
 2895 013054 011004
 2896 013056 001404
 2897 013060 010546

```

;***** TEST 7 *****
;*TEST TO VERIFY THAT BIT "TIE" CAN
;*BE SET. THEN VERIFY THAT BIT "TIE" CAN
;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
;*CLEARED BY A "DEVICE CLEAR"
;:* TEST 7
;*****
TST7: SCOPE
MOV #7,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST10,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DZCSR,R0 ;GET BASE ADDRESS
MOV #TIE,R5 ;SET BIT
MOV R5,(R0) ;SET SET IN DEVICE
MOV (R0),R4 ;READ THE BIT FROM DEVICE
CMP R5,R4 ;WAS BIT SET?
BEQ 1$ ;BR IF YES
ERROR 2 ;*BIT R/W FAILURE
1$: BIC R5,(R0) ;CLEAR THE BIT.
MOV (R0),R4 ;READ DEVICE
BEQ 2$ ;BR IF BITS WERE CLEARED.
MOV R5,-(SP) ;SAVE THE BIT

```

```

2898 013062 005005          CLR      R5          ;SET EXPECTED RESULTS TO 0
2899 013064 104002          ERROR    2          ;*BIT FAILED TO CLEAR
2900 013066 012605          MOV      (SP)+,R5    ;RESTORE THE BIT.
2901 013070 010510          2S:     MOV      R5,(R0) ;SET THE BIT AGAIN
2902 013072 104413          DEVICE.CLR ;ISSUE DEVICE CLEAR
2903 013074 011004          MOV      (R0),R4    ;READ THE BIT.
2904 013076 001402          BEQ      3S         ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
2905 013100 005005          CLR      R5          ;SET EXPECTED TO ZERO
2906 013102 104002          ERROR    2          ;*BIT NOT CLEARED BY DEVICE CLEAR
2907 013104
2908
2909
2910
2911
2912
2913
2914
2915 013104 000004          ;***** TEST 10 *****
2916 013106 012737 000010 001122 ;*THIS TESTS THAT ALL OF THE FOLLOWING
2917 013114 012737 013242 001360 ;*BITS CAN BE: SET, CLEARED, CLEARED BY "DEVICE CLEAR "
2918 013122 013700 002056 ;*BITS TESTED ARE:
2919 013126 012705 000001 ;* TCR0, TCR1, TCR2, TCR3, TCR4, TCR5, TCR6, TCR7
2920 013132 012737 013140 001362 ;:* TEST 10
2921 013140 010510          ;*****
2922 013142 011004          TST10: SCOPE
2923 013144 042704 177400  MOV      #10,STSTNM ;LOAD THE NUMBER OF THIS TEST
2924 013150 020504          MOV      #TST11,NEXT ;POINT TO THE START OF THE NEXT TEST
2925 013152 001401          MOV      DZTCR,R0    ;SET DEVICE ADDRESS
2926 013154 104002          MOV      #TCR0,R5    ;SET EXPECTED RESULTS
2927 013156 040510          MOV      #1$,LOCK    ;SET FOR SW09
2928 013160 011004          1S:     MOV      R5,(R0)    ;SET THE BIT
2929 013162 042704 177400  MOV      (R0),R4     ;READ THE BIT FROM THE DEVICE
2930 013166 005704          BIC      #1C<377>,R4 ;CLEAR HIGH BYTE
2931 013170 001404          CMP      R5,R4       ;WAS BIT OK?
2932 013172 010546          BEQ      2S         ;BR IF YES
2933 013174 005005          ERROR    2          ;*BIT FAILED TO SET.
2934 013176 104002          BIC      R5,(R0)     ;CLEAR THE BIT
2935 013200 012605          MOV      (R0),R4     ;READ THE REGISTER
2936 013202 010510          BIC      #1C<377>,R4 ;CLEAR HIGH BYTE
2937 013204 104413          TST      R4          ;BITS CLEAR?
2938 013206 011004          BEQ      3S         ;BR IF YES
2939 013210 042704 177400  MOV      R5,-(SP)    ;SAVE GOOD RESULTS
2940 013214 005704          CLR      R5          ;SET EXPECTED TO 0
2941 013216 001404          ERROR    2          ;*REPORT BIT NOT CLEAR
2942 013220 010546          MOV      (SP)+,R5    ;RESTORE R5
2943 013222 005005          3S:     MOV      R5,(R0)    ;SET THE BIT AGAIN.
2944 013224 104002          DEVICE.CLR ;ISSUE DEVICE CLEAR
2945 013226 012605          MOV      (R0),R4     ;READ THE REGISTER
2946 013230 104401          BIC      #1C<377>,R4 ;CLEAR HIGH BYTE
2947 013232 106305          TST      R4          ;BITS CLEAR?
2948 013234 001341          BEQ      4S         ;BR IF YES
2949 013236 005037 001362  MOV      R5,-(SP)    ;SAVE GOOD RESULTS
2950
2951
2952
2953
          CLR      R5          ;SET EXPECTED TO 0
          ERROR    2          ;*REPORT BIT NOT CLEAR
          MOV      (SP)+,R5    ;RESTORE R5
          4S:     SCOPI    R5 ;LOCK ON BIT? SET SW09=1
          ASLB    R5          ;CHANGE TO NEXT BIT
          BNE    1$         ;CONTINUE TESTING
          CLR    LOCK        ;MAKE SURE TIGHT LOOP IS CLEANED UP
          ;***** TEST 11 *****
          ;*THIS TESTS THAT ALL OF THE FOLLOWING
          ;*BITS CAN BE: SET, CLEARED, CLEARED BY "RESET INSTR *NOT* DEVICE CLEAR "
          ;*BITS TESTED ARE:

```


M05

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 65
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

```

2954                                     ;* DTR0, DTR1, DTR2, DTR3, DTR4, DTR5, DTR6, DTR7
2955                                     ;*THIS TEST IS NOT DONE IF MODULE IS 20MA VERSION
2956                                     ;::* TEST 11
2957                                     ;*****
2958 013242 000004 TST11: SCOPE
2959 013244 012737 000011 001122 MOV #11,STSTNM ;LOAD THE NUMBER OF THIS TEST
2960 013252 012737 013416 001360 MOV #TST12,NEXT ;POINT TO THE START OF THE NEXT TEST
2961 013260 013700 002056 MOV DZTCR,R0 ;SET DEVICE ADDRESS
2962 013264 012705 000400 MOV #DTR0,R5 ;SET EXPECTED RESULTS
2963 013270 012737 013306 001362 MOV #1$,LOCK ;SET FOR SW09
2964 013276 105737 001414 TSTB EIAFLG ;20MA OR EIA
2965 013302 100001 BPL 1$ ;BR IF EIA
2966 013304 104400 ADVANCE ;EXIT TEST
2967 013306 010510 1$: MOV R5,(R0) ;SET THE BIT
2968 013310 011004 MOV (R0),R4 ;READ THE BIT FROM THE DEVICE
2969 013312 105004 CLRB R4 ;CLEAR LOW BYTE
2970 013314 020504 CMP R5,R4 ;WAS BIT OK?
2971 013316 001401 BEQ 2$ ;BR IF YES
2972 013320 104002 ERROR 2 ;*BIT FAILED TO SET.
2973 013322 040510 2$: BIC R5,(R0) ;CLEAR THE BIT
2974 013324 011004 MOV (R0),R4 ;READ THE REGISTER
2975 013326 105004 CLRB R4 ;CLEAR LOW BYTE
2976 013330 005704 TST R4 ;BITS CLEAR?
2977 013332 001404 BEQ 3$ ;BR IF YES
2978 013334 010546 MOV R5,-(SP) ;SAVE GOOD RESULTS
2979 013336 005005 CLR R5 ;SET EXPECTED TO 0
2980 013340 104002 ERROR 2 ;*REPORT BIT NOT CLEAR
2981 013342 012605 MOV (SP)+,R5 ;RESTORE R5
2982 013344 010510 3$: MOV R5,(R0) ;SET THE BIT AGAIN.
2983 013346 104413 DEVICE.CLR ;ISSUE DEVICE CLEAR
2984 013350 011004 MOV (R0),R4 ;READ THE REGISTER
2985 013352 105004 CLRB R4 ;CLEAR LOW BYTE
2986 013354 030510 BIT R5,(R0) ;WAS BIT CLEARED BY DEVICE.CLR?
2987 013356 001001 BNE .+4 ;BR IF NO (IT SHOULDN'T BE CLEAR)
2988 013360 104002 ERROR 2 ;*BIT CLEARED BY DEVICE.CLR
2989 013362 000005 RESET ;ISSUE A BUS INIT
2990 013364 011004 MOV (R0),R4 ;READ REGISTER
2991 013366 105004 CLRB R4 ;CLEAR LOW BYTE
2992 013370 005704 TST R4 ;BITS CLEAR?
2993 013372 001404 BEQ 4$ ;BR IF YES
2994 013374 010546 MOV R5,-(SP) ;SAVE GOOD RESULTS
2995 013376 005005 CLR R5 ;SET EXPECTED TO 0
2996 013400 104002 ERROR 2 ;*REPORT BIT NOT CLEAR
2997 013402 012605 MOV (SP)+,R5 ;RESTORE R5
2998 013404 104401 4$: SCOP1 ;LOCK ON BIT? SET SW09=1
2999 013406 106305 ASLB R5 ;CHANGE TO NEXT BIT
3000 013410 001336 BNE 1$ ;CONTINUE TESTING
3001 013412 005037 001362 CLR LOCK ;MAKE SURE TIGHT LOOP IS CLEANED UP
3002                                     ;***** TEST 12 *****
3003                                     ;*THIS TEST PERFORMS RESET TESTING &
3004                                     ;*TESTING OF WRITE ONLY OR READ ONLY BIT
3005                                     ;* TEST BITS "RDONE, BIT11, BIT10, BIT9, BIT8, BIT2, BIT1
3006                                     ;* BIT0, SILOAL" ARE READ ONLY AND THAT TRDY IS
3007                                     ;* ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.
3008                                     ;*
3009                                     ;::* TEST 12

```

N05

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 66
DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

```

3010
3011 013416 000004
3012 013420 012737 000012 001122
3013 013426 012737 013534 001360
3014 013434 013700 002042
3015 013440 005005
3016 013442 012710 027607
3017
3018 013446 011004
3019 013450 001401
3020 013452 104002
3021 013454 012710 100000 1S:
3022 013460 011004
3023 013462 001401
3024 013464 104002
3025 013466 012705 100000 2S:
3026 013472 005077 166354
3027 013476 052777 000001 166352
3028 013504 052710 000040
3029 013510 052705 000040
3030 013514 005002
3031 013516 011004 3S:
3032 013520 020504
3033 013522 001404
3034 013524 104414
3035 013526 005202
3036 013530 001372
3037 013532 104002
3038 013534 4S:
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048 013534 000004
3049 013536 012737 000013 001122
3050 013544 012737 013620 001360
3051 013552 104413
3052 013554 013700 002042
3053 013560 012710 177757
3054 013564 012705 050150
3055 013570 011004
3056 013572 020405
3057 013574 001401
3058 013576 104002
3059 013600 012705 000020 1S:
3060
3061
3062 013604 052710 000020
3063 013610 011004
3064 013612 020405
3065 013614 001401

```

```

*****
TST12: SCOPE
MOV #12,$STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST13,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DZCSR,R0 ;SET ADDRESS TO R0
CLR R5 ;SET EXPECTED TO 0
MOV #RDONE+BIT11+BIT10+BIT9+BIT8+BIT2+BIT1+BIT0+SILOAL,(R0) ;WRITE THE BITS
MOV (R0),R4 ;READ BACK THE BITS
BEQ 1S ;BR IF NONE ARE SET.
ERROR 2 ;*BITS WERE SET.
MOV #TRDY,(R0) ;ATTEMPT TO WRITE TRDY
MOV (R0),R4 ;READ TRDY
BEQ 2S ;BR IF NOT SET
ERROR 2 ;*
MOV #TRDY,R5 ;SET EXPECTED BIT
CLR @DZLPR ;LOAD LINE 0
BIS #TCRD,@DZTCR ;SET TCR BIT
BIS #MSENAB,(R0) ;
BIS #MSENAB,R5 ;SET SCAN ENABLE
CLR R2 ;SET COUNTER TO ZERO
MOV (R0),R4 ;READ THE REGISTER
CMP R5,R4 ;BIT SET?
BEQ 4S ;BR IF YES
DELAY ;STALL TIME
INC R2 ;UPDATE COUNTER
BNE 3S ;BR IF COUNTER NOT DONE.
ERROR 2 ;*TRDY NOT SET!
*****
;***** TEST 13 *****
;THIS TEST PERFORMS RESET TESTING AND
;TESTING OF READ ONLY AND WRITE ONLY BITS
;* IN REGISTER DZCSR
;VERIFY THAT "TIE", "SILOEN", "RIE", "MSENAB", "MAINT"
;ARE THE ONLY R/W BITS IN THE DZCSR.
;THEN SET "DCLR" AND VERIFY THEY ARE CLEARED
;:* TEST 13
*****
TST13: SCOPE
MOV #13,$STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST14,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DEVICE.CLR
MOV DZCSR,R0 ;SET UP FOR ERROR MESSAGE
MOV #TC<DCLR>,(R0) ;TRY TO WRITE
MOV #TIE!SILOEN!RIE!MSENAB!MAINT,R5 ;MAKE EXPECTED
MOV (R0),R4 ;ACTUAL
CMP R4,R5 ;CMP EXPECTED VS ACTUAL
BEQ 1S ;YES
ERROR 2 ;*NO
MOV #DCLR,R5 ;EXPECTED...NOTE THAT DCLR REMAINS
;SET LONG ENOUGH TO READ IT....HOWEVER
;IF YOU EXAMINE THIS BIT IT SHOULD BE CLEAR.
BIS #DCLR,(R0) ;DEVICE MASTER RESET
MOV (R0),R4 ;ACTUAL
CMP R4,R5 ;CMP ACTUAL VS EXPECTED
BEQ 2S ;YES

```

B06

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 67
DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

3066	013616	104002		
3067	013620			
3068				
3069				
3070				
3071				
3072				
3073				
3074	013620	000004		
3075	013622	012737	000014	001122
3076	013630	012737	013710	001360
3077	013636	104413		
3078	013640	013700	002046	
3079	013644	012777	177777	166200
3080	013652	011004		
3081	013654	010405		
3082	013656	042705	104000	
3083	013662	020405		
3084	013664	001401		
3085	013666	104002		
3086	013670	010403		
3087	013672	005103		
3088	013674	010377	166152	
3089	013700	011004		
3090	013702	020405		
3091	013704	001401		
3092	013706	104002		
3093	013710			
3094				
3095				
3096				
3097				
3098				
3099				
3100	013710	000004		
3101	013712	012737	000015	001122
3102	013720	012737	013774	001360
3103	013726	104413		
3104	013730	013700	002062	
3105	013734	012777	177777	166124
3106	013742	011004		
3107	013744	010405		
3108	013746	020405		
3109	013750	001401		
3110	013752	104002		
3111	013754	010403		
3112	013756	005103		
3113	013760	010377	166102	
3114	013764	011004		
3115	013766	020405		
3116	013770	001401		
3117	013772	104002		
3118	013774			
3119				
3120				
3121				

```

25: ERROR 2 ;#NO
;***** TEST 14 *****
;THIS TEST PERFORMS RESET TESTING AND
;TESTING OF READ ONLY REGISTER DZRBUF
;AND TESTING OF WRITE ONLY REGISTER DZLPR
;: TEST 14
;*****
TST14: SCOPE
MOV #14,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST15,NEXT ;POINT TO THE START OF THE NEXT TEST
DEVICE.CLR ;CLEAR DZ11
MOV DZRBUF,R0 ;SET UP FOR ERROR MESSAGE
MOV #1,DZLPR ;TRY TO WRITE ALL 1'S
MOV (R0),R4 ;ACTUAL
MOV R4,R5 ;MAKE EXPECTED
BIC #DVALID:BIT11,R5 ;DITTO
CMP R4,R5 ;CMP ACTUAL VS EXPECTED
BEQ 15 ;IF YES,GO CONTINUE PROCESSING
ERROR 2 ;ERROR- BIT PATTERN NOT CORRECT
15: MOV R4,R3 ;GET A COPY OF THE ACTUAL BIT PATTERN
COM R3 ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
MOV R3,DZLPR ;TRY TO WRITE
MOV (R0),R4 ;ACTUAL
CMP R4,R5 ;CMP ACTUAL VS EXPECTED
BEQ 25 ;IF YES, GET OUT OF THIS TEST
ERROR 2 ;#NO

25: ;***** TEST 15 *****
;THIS TEST PERFORMS RESET TESTING AND
;TESTING OF READ ONLY REGISTER DZMSR
;AND TESTING OF WRITE ONLY REGISTER DZTDR
;: TEST 15
;*****
TST15: SCOPE
MOV #15,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST16,NEXT ;POINT TO THE START OF THE NEXT TEST
DEVICE.CLR ;CLEAR DZ11
MOV DZMSR,R0 ;SET UP FOR ERROR MESSAGE
MOV #1,DZTDR ;TRY TO WRITE ALL 1'S
MOV (R0),R4 ;ACTUAL
MOV R4,R5 ;MAKE EXPECTED
CMP R4,R5 ;CMP ACTUAL VS EXPECTED
BEQ 15 ;IF YES,GO CONTINUE PROCESSING
ERROR 2 ;ERROR- BIT PATTERN NOT CORRECT
15: MOV R4,R3 ;GET A COPY OF THE ACTUAL BIT PATTERN
COM R3 ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
MOV R3,DZTDR ;TRY TO WRITE
MOV (R0),R4 ;ACTUAL
CMP R4,R5 ;CMP ACTUAL VS EXPECTED
BEQ 25 ;IF YES, GET OUT OF THIS TEST
ERROR 2 ;#NO

25: ;***** TEST 16 *****
;VERIFY THAT IF WE ARE IN "STAGGERED" MODE

```

3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177

013774 000004
013776 012737 000016 001122
014004 012737 014164 001360
014012 012737 014064 001362
014020 105737 001414
014024 100005
014026 104400
014030 013700 002062
014034 104413
014036 005003
014040 012702 000001
014044 005737 001370
014050 100405
014052 013737 001360 001126
014060 000177 165042
014064 130237 001364
014070 001004
014072 005203
014074 106302
014076 103372
014100 104400
014102 010204
014104 032703 000001
014110 001402
014112 006204
014114 000401
014116 006304
014120 005005
014122 150405
014124 000305
014126 150405
014130 150277 165724
014134 011004
014136 020504
014140 001401
014142 104002
014144 140277 165710
014150 011004
014152 001402
014154 005005
014156 104002
014160 104401
014162 000743

```

: * THAT SETTING "DTR" FOR A LINE WILL
: * BRING UP "RING" AND "CARRIER" FOR THE
: * ASSOCIATED LINE IN WHICH WE ARE STAGGERED!
: * LINE0 DTR= LINE1 RING AND CARRIER
: * LINE1 DTR= LINE2 RING AND CARRIER
: * LINE2 DTR= LINE3 RING AND CARRIER
: * LINE3 DTR= LINE4 RING AND CARRIER
: *
: * ETC...

: : * TEST 16
: : *****
TST16: SCOPE
MOV #16, STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST17, NEXT ;POINT TO THE START OF THE NEXT TEST
MOV #15, LOCK ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
TSTB EIAFLG ;EIA OR 20MA?
BPL .+14 ;BR IF EIA
ADVANCE ;EXIT TEST
MOV DZMSR, R0 ;SET REGISTER
DEVICE. CLR ;INIT DZ11
CLR R3 ;ZERO LINE NUMBER
MOV #1, R2 ;SET POINTER
TST MODE ;ARE WE IN STAGGERED MODE?
BMI 15 ;YES WE ARE!
MOV NEXT, SLPADR ;LEAVE THIS TEST! NOT STAGGERED
JMP @SLPADR ;EXIT
15: BITB R2, LINE ;TEST THIS LINE?
BNE 35 ;YES
25: INC R3 ;LINE #
ASLB R2 ;GET NEXT LINE
BCC 15 ;KEEP TESTING
ADVANCE ;ADVANCE THIS TEST
35: MOV R2, R4 ;SAVE BINARY BIT FOR LINE #
BIT #BIT0, R3 ;GET STAGGERED COMPANION LINE
BEQ 45 ;BR IF LINE EVEN
ASR R4 ;ADJUST LINE
BR 55 ;ADJUST LINE
45: ASL R4 ;ADJUST LINE
55: CLR R5 ;SET EXPECTED
BISB R4, R5
SWAB R5
BISB R4, R5
BISB R2, @DZTCR ;SET DTR
MOV (R0), R4 ;READ MSR REGISTER
CMP R5, R4 ;OK?
BEQ 65 ;YES
ERROR 2 ;*ERROR IN RING OR CARRIER
65: BICB R2, @DZTCR ;CLEAR DTR
MOV (R0), R4 ;READ MSR
BEQ 75 ;BR IF THEY CLEARED
CLR R5 ;SET EXPECTED TO 0
ERROR 2 ;*BITS NOT CLEARED
75: SCOP1 ;LOCK ON SIGNAL?
BR 25 ;CONTINUE TEST
: : ***** TEST 17 *****

```

```

3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189 014164 000004
3190 014166 012737 000017 001122
3191 014174 012737 014322 001360
3192 014202 012737 014236 001362
3193 014210 105737 001370
3194 014214 100401
3195 014216 104400
3196 014220 105737 001414
3197 014224 100774
3198 014226 013700 002062
3199 014232 012702 000001
3200 014236 130237 001364
3201 014242 001003
3202 014244 106302
3203 014246 103373
3204 014250 104400
3205 014252 005005
3206 014254 150205
3207 014256 000305
3208 014260 150205
3209 014262 150277 165572
3210 014266 104414
3211 014270 011004
3212 014272 020504
3213 014274 001401
3214 014276 104002
3215 014300 140277 165554
3216 014304 104414
3217 014306 011004
3218 014310 001402
3219 014312 005005
3220 014314 104002
3221 014316 104401
3222 014320 000751

```

```

:*TEST TO VERIFY THAT IF IN "EXTERNAL"
:*MODE; SETTING DTR FOR SELECTED LINES
:*WILL BRING UP "CARRIER" AND "RING"
:*FOR THAT SAME LINE. NOTE: IF YOU HAVE
:*SELECTED MODE AS "EXTERNAL"; THE H325 TEST CONNECTER
:*MUST BE USED ON ALL SPECIFIED LINES.
:*LINES MAY BE SPECIFIED BY SWRO3=1
:*AND SWRO0=1 AT START TIME OR ALTERING
:*STATUS MAP.

```

::* TEST 17

```

TST17: SCOPE
MOV #17,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST20,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV #35,LOCK ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
TSTB MODE ;EXTERNAL?
BMI 25 ;BR IF YES
ADVANCE ;EXIT TEST
TSTB EIAFLG ;YOU BETTER BE IN
BMI 15 ;EIA MODE FOR THIS TEST.
MOV DZMSR,R0 ;SET REGISTER
MOV #1,R2 ;SET LINE POINTER
BITB R2,LINE ;LINE SELECTED?
BNE 55 ;BR IF YES
ASLB R2 ;NEXT LINE
BCC 35 ;CONTINUE TEST
ADVANCE ;ADVANCE THIS TEST
CLR R5 ;SET EXPECTED
BISB R2,R5
SWAB R5
BISB R2,R5
BISB R2,ANDZTCR ;SET DTR
DELAY ;CABLE DELAY
MOV (R0),R4 ;READ MSR
CMP R5,R4 ;BITS OK?
BEQ 65 ;BR IF YES
ERROR 2 ;CARRIER OR RING ERROR
BICB R2,ANDZTCR ;CLEAR DTR
DELAY ;CABLE DELAY
MOV (R0),R4 ;READ MSR
BEQ 75 ;BR IF BITS CLEARED
CLR R5 ;CLEAR EXPECTED LOC.
ERROR 2 ;BITS NOT CLEARED.
SCOPI ;LOCK ON LINE?
BR 45 ;CONTINUE TEST

```

```

***** TEST 20 *****
:* THIS TEST VERIFIES THAT TROY IS SET WHEN A LINE
:* IS READY TO BE LOADED, AND THAT THE LINE SPECI-
:* FIED IN BITS 8-10 OF DZCSR CORRESPOND
:* TO THE LINE SELECTED IN DZTCR.

```

::* TEST 20

```

3223
3224
3225
3226
3227
3228
3229
3230
3231 014322 000004
3232 014324 012737 000020 001122
3233 014332 012737 014436 001360

```

```

TST20: SCOPE
MOV #20,STSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST21,NEXT ;POINT TO THE START OF THE NEXT TEST

```

E06

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 70
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

```

3234 014340 104413          DEVICE.CLR          ;ISSUE A "DEVICE CLEAR" (RESET)
3235 014342 013700 002042  MOV          DZCSR,R0          ;SET POINTER
3236 014346 012705 100040  MOV          #MSENAB!TRDY,R5 ;START THE EXPECTED LINE NUMBER AT 0
3237 014352 012702 000001  MOV          #1,R2          ;USING R2 AS A BIT POINTER, POINT TO LINE 0
3238 014356 130237 001364          BITB          R2,LINE          ;IS THIS LINE SELECTED?
3239 014362 001420          BEQ          5$             ;IF NO, SKIP THE STARTUP
3240 014364 050277 165466          BIS          R2,JDZTCR       ;SET THE GO BIT FOR THIS LINE
3241 014370 052710 000040          BIS          #MSENAB,(R0)    ;START THE SCANNER
3242 014374 005004          CLR          R4             ;SET FOR DELAY
3243 014376 032710 100000          BIT          #TRDY,(R0)     ;TX READY?
3244 014402 001004          BNE          4$             ;BR IF YES
3245 014404 104414          DELAY          ;DELAY
3246 014406 005204          INC          R4             ;COUNTER
3247 014410 001372          BNE          3$             ;BR IF (>)0!
3248 014412 104002          ERROR        2             ;*TX NOT READY!
3249 014414 011004          MOV          (R0),R4        ;GET THE LINE POINTED TO BY THE SCANNER
3250 014416 020405          CMP          R4,R5          ;IS THE LINE NUMBER WHAT IT SHOULD BE?
3251 014420 001401          BEQ          5$             ;IF YES,GO WORK ON THE NEXT LINE
3252 014422 104002          ERROR        2             ;*LINE NUMBER DID NOT MATCH TCR BIT
3253 014424 062705 000400          ADD          #400,R5        ;POINT TO THE NEXT EXPECTED LINE
3254 014430 104413          DEVICE.CLR          ;ISSUE A "DEVICE CLEAR" (RESET)
3255 014432 106302          ASLB         R2             ;POINT TO THE NEXT LINE.ARE ALL LINES TESTED?
3256 014434 103350          BCC          1$             ;IF NOT, GO DO THE NEXT LINE
3257 014436
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267 014436 000004          ;***** TEST 21 *****
3268 014440 012737 000021 001122          ;*TEST TO TRANSMIT ONE CHAR AND
3269 014446 012737 014754 001360          ;*RECEIVE ONE CHAR ON ONE LINE
3270 014454 012737 014732 001362          ;*AT A TIME. THE CHAR IS "252" AND
3271 014462 104417          ;*ALL SELECTED LINES WILL BE TURNED ON
3272 014464 013701 001366          ;*ONE AT A TIME. THIS IS THE FIRST TIME ANY
3273 014470 012702 000001          ;*DATA IS CHECKED IN THE RECEIVER.
3274 014474 030237 001364          ;:# TEST 21
3275 014500 001402          ;:*****
3276 014502 010177 165344          TST21: SCOPE
3277 014506 005201          MOV          #21,$STSTNM     ;LOAD THE NUMBER OF THIS TEST
3278 014510 106302          MOV          #TST22,NEXT     ;POINT TO THE START OF THE NEXT TEST
3279 014512 103370          MOV          #16$,$LOCK      ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3280 014514 005037 001372          DCLASH          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3281 014520 012702 000001          MOV          PAR,R1          ;PICK UP PARAMETERS
3282 014524 052777 000040 165310          MOV          #1,R2          ;PICK UP INIT POINTER
3283 014532 030237 001364          BIT          R2,LINE          ;SHOULD THIS LINE BE SET UP ?
3284 014536 001463          BEQ          2$             ;NO
3285 014540 010277 165312          MOV          R1,$DZLPR       ;SET UP LINE PARAMETERS
3286 014544 032777 000200 165270          INC          R1             ;POSITION POINTER TO THE NEXT LINE
3287 014552 001401          ASLB         R2             ;GOT 'EM ALL ?
3288 014554 104020          BCC          1$             ;IF NO, GO SET UP THE NEXT LINE
3289 014556 005005          CLR          SAVLIN          ;CLEAR LINE # INDICATOR
3290          MOV          #1,R2          ;LINE POINTER
3291          BIS          #MSENAB,$DZCSR ;START SCANNER
3292          BIT          R2,LINE          ;VALID LINE ?
3293          BEQ          14$          ;NO SET UP NEXT LINE
3294          MOV          R2,$DZTCR       ;SET TCR BIT
3295          BIT          #RDONE,$DZCSR ;IS REC DONE = 0 ?
3296          BEQ          5$             ;IF YES, ALLOW TIME FOR TRDY TO SET
3297          ERROR        20          ;*REC DONE SHOULD = 0
3298          CLR          R5

```

F06

```

3290 014560 032777 100000 165254 6S: BIT #TRDY,JDZCSR
3291 014566 001004 BNE 7S
3292 014570 104414 DELAY
3293 014572 105205 INCB R5
3294 014574 001371 BNE 6S
3295 014576 104003 ERROR 3 ;*TRDY FAILED TO SET!
3296 014600 112777 000252 165260 7S: MOVB #252,JDZTDR ;LOAD CHARACTER
3297 014606 013705 001372 MOV SAVLIN,R5 ;MAKE EXPECTED LINE #
3298 014612 105737 001371 TSTB MODE+1 ;IS THIS TEST IN STAGGERED MODE?
3299 014616 001406 BEQ 10S ;IF NOT, SKIP STAGGERED SETUP
3300
3302 ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3303 014620 006205 ASR R5 ;GET THE LAST BIT INTO THE CARRY BIT
3304 014622 103402 BCS 8S ;IF IT IS SET, GO CLEAR IT
3305 014624 000261 SEC ;IF IT IS CLEAR SET IT HERE
3306 014626 000401 BR 9S ;SKIP THE CLEARING
3307 014630 000241 CLC ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3308 014632 006105 8S: ROL R5 ;GET THE NEW BIT BACK INTO R5
3309 014634 000305 9S: SWAB R5 ;MOVE THE LINE NUMBER TO THE UPPER BYTE
3310 014636 152705 000252 10S: BISB #252,R5 ;ADD CHARACTER
3311 014642 104400 ADVANCE ;ADVANCE TO THE NEXT TEST
3312 014644 052705 100000 BIS #DVALID,R5 ;ADD DATA VALID
3313 014650 005003 CLR R3
3314 014652 032777 000200 165162 11S: BIT #RDONE,JDZCSR
3315 014660 001004 BNE 12S
3316 014662 104414 DELAY
3317 014664 105203 INCB R3
3318 014666 001371 BNE 11S
3319 014670 104004 ERROR 4 ;*RDONE FAILED TO SET!
3320 014672 017704 165150 12S: MOV JDZRBUF,R4 ;LOAD THE VALUE ACTUALLY RECEIVED
3321 014676 020405 CMP R4,R5 ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
3322 014700 001401 BEQ 13S ;IF YES, GO DO THE NEXT LINE
3323 014702 104006 ERROR 6 ;*NO DATA/CONTENTS DID NOT COMPARE
3324 014704 104401 13S: SCOP1 ;CHECK TO SEE IF SWITCH NINE IS SET
3325 014706 040277 165144 14S: BIC R2,JDZTCR ;CLEAR TCR BIT FOR THAT LINE.
3326 014712 005237 001372 15S: INC SAVLIN ;INC EXPECTED LINE
3327 014716 013700 001372 MOV SAVLIN,R0 ;SET UP CHARACTER OFFSET
3328 014722 006300 ASL R0 ;MAKE THE OFFSET A POWER OF TWO
3329 014724 106302 ASLB R2 ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
3330 014726 103301 BCC 3S ;IF NO, GO AROUND AGAIN FOR NEXT LINE
3331 014730 104400 ADVANCE ;GO TO NEXT TEST
3332
3333 ;TIGHT SCOPE LOOP FOR THIS TEST. LOOP TRANSMITS CHARACTERS ONLY
3334 ;THANKS TO JOHN MCINTYRE AND WESTMINSTER STAFF...
3335
3336 014732 032777 100000 165102 16S: BIT #TRDY,JDZCSR ;IS TRANSMITTER READY?
3337 014740 001774 BEQ 16S ;IF NOT, WAIT FOR IT
3338 014742 112777 000252 165116 MOVB #252,JDZTDR ;LOAD THE CHARACTER
3339 014750 104401 SCOP1 ;LOOP AGAIN IF SW09=1
3340 014752 000755 BR 14S ;OTHERWISE, GO PICK UP THE TEST NORMALLY
3341
3342 ;***** TEST 22 *****
3343 ;* THIS TEST PROVES THAT THE TRANSMITTER TRANSMITS
3344 ;*CHARACTERS (FLAG MODE)AND THE RECEIVER RECEIVES (FLAG MODE)
3345 ;*(ONE LINE AT A TIME BASED UPON VALID LINES)

```

G06

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 72
DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

```

3346                                     ;*THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED
3347                                     ;::* TEST 22
3348                                     ;*****
3349 014754 000004 TST22: SCOPE
3350 014756 012737 000022 001122 MOV #22,STSTNM ;LOAD THE NUMBER OF THIS TEST
3351 014764 012737 015302 001360 MOV #TST23,NEXT ;POINT TO THE START OF THE NEXT TEST
3352 014772 012737 015106 001362 MOV #45,LOCK ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3353 015000 104417 DCLASM ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3354 015002 013701 001366 MOV PAR,R1 ;PICK UP PARAMETERS
3355 015006 012702 000001 MOV #1,R2 ;PICK UP INIT POINTER
3356 015012 030237 001364 1S: BIT R2,LINE ;SHOULD THIS LINE BE SET UP ?
3357 015016 001402 BEQ 2S ;NO
3358 015020 010177 165026 MOV R1,JDZLPR ;SET UP LINE PARAMETERS
3359 015024 005201 2S: INC R1 ;POSITION POINTER TO THE NEXT LINE
3360 015026 106302 ASLB R2 ;GOT 'EM ALL ?
3361 015030 103370 BCC 1S ;IF NO, GO SET UP THE NEXT LINE
3362 015032 005037 001372 CLR SAVLIN ;CLEAR LINE # INDICATOR
3363 015036 012700 001422 MOV #TDO,R0 ;POINT TO THE DATA AREA
3364 015042 005020 CLR (R0)+ ;CLEAR A DATA WORD
3365 015044 022700 001462 CMP #STOP,R0 ;FINISHED ?
3366 015050 001374 BNE -6 ;NO
3367 015052 005000 CLR R0 ;CLEAR OFFSET
3368 015054 013737 002046 001400 MOV DZRBUF,REGIST ;SAVE FOR ERROR MSG
3369 015062 012702 000001 MOV #1,R2 ;LINE POINTER
3370 015066 052777 000040 164746 BIS #MSENAB,JDZCSR ;START SCANNER
3371 015074 030237 001364 3S: BIT R2,LINE ;VALID LINE ?
3372 015100 001465 BEQ 14S ;NO SET UP NEXT LINE
3373 015102 010277 164750 MOV R2,JDZTCR ;SET TCR BIT
3374 015106 032777 000200 164726 4S: BIT #RDONE,JDZCSR ;IS REC DONE = 0 ?
3375 015114 001401 BEQ 5S ;IF YES, ALLOW TIME FOR TRDY TO SET
3376 015116 104020 ERROR 20 ;*REC DONE SHOULD = 0
3377 015120 005005 5S: CLR R5
3378 015122 032777 100000 164712 6S: BIT #TRDY,JDZCSR
3379 015130 001004 BNE 7S
3380 015132 104414 DELAY
3381 015134 105205 INCB R5
3382 015136 001371 BNE 6S
3383 015140 104003 ERROR 3
3384 015142 116077 001422 164716 7S: MOVB TDO(R0),JDZTDR ;*TRDY FAILED TO SET!
3385 015150 013705 001372 MOV SAVLIN,R5 ;LOAD CHARACTER
3386 015154 105737 001371 TSTB MODE+1 ;MAKE EXPECTED LINE #
3387 015160 001406 BEQ 10S ;IS THIS TEST IN STAGGERED MODE?
3388                                     ;IF NOT, SKIP STAGGERED SETUP
3389                                     ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3390
3391 015162 006205 ASR R5 ;GET THE LAST BIT INTO THE CARRY BIT
3392 015164 103402 BCS 8S ;IF IT IS SET, GO CLEAR IT
3393 015166 000261 SEC ;IF IT IS CLEAR SET IT HERE
3394 015170 000401 BR 9S ;SKIP THE CLEARING
3395 015172 000241 8S: CLC ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3396 015174 006105 9S: ROL R5 ;GET THE NEW BIT BACK INTO R5
3397 015176 000305 10S: SWAB R5 ;MOVE THE LINE NUMBER TO THE UPPER BYTE
3398 015200 156005 001422 BISB TDO(R0),R5 ;ADD CHARACTER
3399 015204 052705 100000 BIS #DVALID,R5 ;ADD DATA VALID
3400 015210 005003 CLR R3
3401 015212 032777 000200 164622 11S: BIT #RDONE,JDZCSR

```


H06

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 73
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

3402	015220	001004			BNE	12\$		
3403	015222	104414			DELAY			
3404	015224	005204			INC	R4		
3405	015226	001371			BNE	11\$		
3406	015230	104004			ERROR	4		;
3407	015232	017704	164610	12\$:	MOV	JDZRBUF,R4		RDONE FAILED TO SET!
3408	015236	020405			CMP	R4,R5		LOAD THE VALUE ACTUALLY RECEIVED
3409	015240	001401			BEQ	13\$		COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
3410	015242	104006			ERROR	6		IF YES, GO DO THE NEXT LINE
3411	015244	104401		13\$:	SCOPI			NO DATA/CONTENTS DID NOT COMPARE
3412	015246	105260	001422		INCB	TDO(RO)		CHECK TO SEE IF SWITCH NINE IS SET
3413	015252	001315			BNE	4\$		INCREMENT BINARY PATTERN FOR THIS LINE
3414	015254	040277	164576	14\$:	BIC	R2,JDZTCR		GO 'ROUND AGAIN FOR NEXT CHARACTER
3415	015260	005237	001372	15\$:	INC	SAVLIN		CLEAR TCR BIT FOR THAT LINE.
3416	015264	013700	001372		MOV	SAVLIN,RO		INC EXPECTED LINE
3417	015270	006300			ASL	RO		SET UP CHARACTER OFFSET
3418	015272	106302			ASLB	R2		MAKE THE OFFSET A POWER OF TWO
3419	015274	103277			BCC	3\$		SHIFT THE LINE POINTER. ARE WE ALL DONE?
3420	015276	005037	001362		CLR	LOCK		IF NO, GO AROUND AGAIN FOR NEXT LINE

3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457

```
***** TEST 23 *****  
:THIS TEST WILL PROVE THAT:  
: 1) THE TRANSMITTER "BREAK BIT" WORKS  
: 2) THE RECEIVER CAN FLAG "FRAMING ERRORS"  
: 3) THE RECEIVER CAN FLAG "PARITY ERRORS"  
: ONLY ONE LINE AT A TIME WILL BE EXERCISED.  
: THIS TEST WILL NOT BE EXERCISED UNLESS  
: CONNECTED BY EXTERNAL PLUG.
```

```
::* TEST 23  
:*****  
:*****
```

3433	015302	000004			SCOPE			
3434	015304	012737	000023	001122	MOV	#23,\$TSTNM		LOAD THE NUMBER OF THIS TEST
3435	015312	012737	015560	001360	MOV	#TST24,NEXT		POINT TO THE START OF THE NEXT TEST
3436	015320	012737	015416	001362	MOV	#3\$,LOCK		SET FOR LOOP
3437	015326	005737	001370		TST	MODE		ARE WE RUNNING IN INTERNAL MODE?
3438	015332	001510			BEQ	12\$		IF SO, SKIP THIS TEST
3439	015334	104417			DCLASM			CLEAR DEVICE AND SET MAINT BIT IF I MODE
3440	015336	013701	001366		MOV	PAR,R1		PICK UP PARAMETERS
3441	015342	052701	000300		BIS	#ODDPAR!PARITY,R1		FORCE ODD PARITY
3442	015346	012700	000001		MOV	#1,RO		PICK UP INIT POINTER
3443	015352	030037	001364	1\$:	BIT	RO,LINE		SHOULD THIS LINE BE SET UP ?
3444	015356	001402			BEQ	2\$		IF NOT DON'T SET IT UP
3445	015360	010177	164466		MOV	R1,JDZLPR		OTHERWISE, SET UP LINE PARAMETERS
3446	015364	005201		2\$:	INC	R1		
3447	015366	106300			ASLB	RO		GOT 'EM ALL ?
3448	015370	103370			BCC	1\$		NO
3449	015372	005037	001372		CLR	SAVLIN		CLEAR LINE #
3450	015376	012702	000001		MOV	#1,R2		LINE POINTER
3451	015402	052777	000040	164432	BIS	#MSENAB,JDZCSR		SET MASTER SCAN ENABLE
3452	015410	013737	002046	001400	MOV	DZRBUF,REGIST		SAVE FOR ERRR MESSAGE
3453	015416	030237	001364	3\$:	BIT	R2,LINE		
3454	015422	001443			BEQ	10\$		
3455	015424	010277	164426		MOV	R2,JDZTCR		SET TCR BIT
3456	015430	110277	164434		MOVB	R2,JDZTDR		SET BREAK BIT
3457	015434	112777	000377	164424	MOVB	#377,JDZTDR		LOAD CHARACTER

```

3458 015442 013705 001372      MOV     SAVLIN,R5      ;MAKE EXPECTED DATA
3459 015446 105737 001371      TSTB   MODE+1        ;IS THIS TEST IN STAGGERED MODE?
3460 015452 001406                BEQ     7$            ;IF NOT, SKIP STAGGERED SETUP
3461
3462                                ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3463
3464 015454 006205      ASR     R5            ;GET THE LAST BIT INTO THE CARRY BIT
3465 015456 103402      BCS     5$            ;IF IT IS SET, GO CLEAR IT
3466 015460 000261      SEC                ;IF IT IS CLEAR SET IT HERE
3467 015462 000401      BR      6$            ;SKIP THE CLEARING
3468 015464 000241      CLC                ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3469 015466 006105      5$:    ROL     R5            ;GET THE NEW BIT BACK INTO R5
3470 015470 000305      6$:    SWAB   R5            ;PUT LINE NUMBER IN UPPER BYTE
3471 015472 052705 130000      BIS     #DVALID:PARER:FMERR,R5 ;ADD EXPECTED
3472 015476 005004      CLR     R4
3473 015500 032777 000200 164334 8$:    BIT     #RDONE,ADZCSR
3474 015506 001004      BNE     9$
3475 015510 104414      DELAY
3476 015512 005204      INC     R4
3477 015514 001371      BNE     8$
3478 015516 104004      ERROR  4             ;*RDONE FAILED TO SET!
3479 015520 017704 164322 9$:    MOV     ADZRBUF,R4    ;ACTUAL
3480 015524 020405      CMP     R4,R5        ;CMP ACTUAL VS EXPECTED. DO THEY MATCH?
3481 015526 001401      BEQ     10$         ;IF YES, GO CLEAN UP
3482 015530 104006      ERROR  6             ;*DATA/CONTENTS FAILED TO COMPARE
3483 015532 105077 164332 10$:   CLRB   ADZTDR        ;CLEAR BREAK BITS
3484 015536 104401      SCOP1
3485 015540 005237 001372 11$:   INC     SAVLIN        ;INC LINE #
3486 015544 040277 164306      BIC     R2,ADZTCR    ;CLEAR TCR BIT
3487 015550 106302      ASLB   R2
3488 015552 103321      BCC    3$
3489 015554 005037 001362 12$:   CLR     LOCK         ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST
3490
3491                                ;***** TEST 24 *****
3492                                ;* THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
3493                                ;*WHILE THE PROCESSOR STATUS IS SET EXACTLY
3494                                ;*TO WHAT THE DZ11 PRIORITY IS SET TO.
3495                                ;*DEFAULT PRIORITY IS AT 5 (240).
3496                                ;::* TEST 24
3497                                ;*****
3498 015560 000004      TST24: SCOPE
3499 015562 012737 000024 001122      MOV     #24,STSTNM   ;LOAD THE NUMBER OF THIS TEST
3500 015570 012737 016066 001360      MOV     #TST25,NEXT ;POINT TO THE START OF THE NEXT TEST
3501 015576 104417      DCLASM
3502 015600 013701 001366      MOV     R0,R1        ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3503 015604 012702 000001      MOV     #1,R2        ;PICK UP PARAMETERS
3504 015610 030237 001364 1$:    BIT     R2,LINE      ;PICK UP INIT POINTER
3505 015614 001402      BEQ     2$            ;SHOULD THIS LINE BE SET UP ?
3506 015616 010177 164230      MOV     R1,ADZLPR    ;NO
3507 015622 005201 2$:    INC     R1            ;SET UP LINE PARAMETERS
3508 015624 106302      ASLB   R2            ;POSITION POINTER TO THE NEXT LINE
3509 015626 103370      BCC    1$            ;GOT 'EM ALL ?
3510 015630 005037 001372      CLR     SAVLIN        ;IF NO, GO SET UP THE NEXT LINE
3511 015634 106437 026170      MTPS   ADZPRT        ;CLEAR LINE # INDICATOR
3512 015640 113777 001364 164210 3$:    MOVB   LINE,ADZTCR   ;SET CPU STATUS TO DZ11 PRIO,
3513 015646 012777 015734 164222      MOV     #6$,ADZTIV   ;ENABLE THE VALID LINES
;SET UP THE TRANSMITTER INTERRUPT VECTOR

```

```

3514 015654 012777 015742 164210 MOV #75,ADZRIV ;SET UP THE RECEIVER INTERRUPT VECTOR
3515 015662 013777 026170 164204 MOV DZPRT,ADZRI5 ;SET THE INTERRUPT VECTOR STATUS
3516 015670 013777 026170 164202 MOV DZPRT,ADZTI5 ;SET TRANSMITTER INTERRUPT PRIORITY
3517 015676 052777 040040 164136 BIS #TIE!MSENAB,ADZCSR ;ENABLE THE DEVICE
3518 015704 005005 CLR R5
3519 015706 032777 100000 164126 4$: BIT #TRDY,ADZCSR
3520 015714 001403 BEQ 5$
3521 015716 000240 NOP
3522 015720 000240 NOP
3523 015722 000411 BR 8$
3524 015724 104414 5$: DELAY
3525 015726 005205 INC R5
3526 015730 001366 BNE 4$
3527 015732 104003 ERROR 3 ;*TRDY NOT SET!
3528 015734 104010 6$: ERROR 10 ;*TRANSMITTER SHOULD NOT INTERRUPT
3529 015736 022626 CMP (SP)+,(SP)+ ;POP FOR FAKE RTI
3530 015740 000402 BR 8$ ;CONTINUE TEST
3531 015742 104012 7$: ERROR 12 ;*RECEIVER SHOULD NOT INTERRUPT
3532 015744 022626 CMP (SP)+,(SP)+ ;POP FOR FAKE RTI
3533 015746 042777 040000 164066 8$: BIC #TIE,ADZCSR ;RESET TRANSMITTER INTERRUPT ENABLE
3534 015754 113777 001422 164104 MOV#B TDO,ADZTDR ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
3535 015762 012777 016052 164106 MOV #11$,ADZTIV ;SET UP THE TRANSMITTER INTERRUPT VECTOR
3536 015770 012777 016060 164074 MOV #12$,ADZRIV ;SET UP THE RECEIVER INTERRUPT VECTOR
3537 015776 013777 026170 164070 MOV DZPRT,ADZRI5 ;SET THE INTERRUPT VECTOR STATUS
3538 016004 013777 026170 164066 MOV DZPRT,ADZTI5 ;SET TRANSMITTER INTERRUPT PRIORITY
3539 016012 052777 000140 164022 BIS #RIE!MSENAB,ADZCSR ;ENABLE THE DEVICE
3540 016020 005005 CLR R5
3541 016022 032777 000200 164012 9$: BIT #RDONE,ADZCSR
3542 016030 001403 BEQ 10$
3543 016032 000240 NOP
3544 016034 000240 NOP
3545 016036 000412 BR 13$
3546 016040 104414 10$: DELAY
3547 016042 005205 INC R5
3548 016044 001366 BNE 9$
3549 016046 104004 ERROR 4 ;*NO RX DONE! (NOT SET)
3550 016050 000405 BR 13$ ;CONTINUE TEST
3551 016052 104010 11$: ERROR 10 ;*TRANSMITTER SHOULD NOT INTERRUPT
3552 016054 022626 CMP (SP)+,(SP)+ ;POP FOR FAKE RTI
3553 016056 000402 BR 13$ ;CONT TEST
3554 016060 104012 12$: ERROR 12 ;*RECEIVER SHOULD NOT INTERRUPT
3555 016062 022626 CMP (SP)+,(SP)+ ;POP FOR FAKE RTI
3556 016064 13$:
3557 016064 104413 DEVICE.CLR ;ISSUE DEVICE CLEAR (RESET)
3558 ;***** TEST 25 *****
3559 ;* THIS TEST VERIFIES THAT THE DEVICE DOES INTERRUPT
3560 ;*WHILE THE PROCESSOR STATUS IS SET TO EXACTLY
3561 ;*ONE LEVEL LOWER THAN THE DZ11. DZ11 PRIORITY
3562 ;*DEFAULT TO LEVEL 5 MINUS ONE LEVEL IS LEVEL 4.
3563 ;::* TEST 25
3564 ;*****
3565 016066 000004 TST25: SCOPE
3566 016070 012737 000025 001122 MOV #25,$TSTNM ;LOAD THE NUMBER OF THIS TEST
3567 016076 012737 016422 001360 MOV #TST26,NEXT ;POINT TO THE START OF THE NEXT TEST
3568 016104 104417 DCLASM ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3569 016106 013701 001366 MOV PAR,R1 ;PICK UP PARAMETERS

```

K06

MD-11-DZDZA-B MACY11 27(732) 03-NOV-76 15:03 PAGE 76
 DZDZAB.P11 DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

3570	016112	012702	000001		MOV	#1,R2	;PICK UP INIT POINTER
3571	016116	030237	001364	1S:	BIT	R2,LINE	;SHOULD THIS LINE BE SET UP ?
3572	016122	001402			BEQ	2S	;NO
3573	016124	010177	163722		MOV	R1,ADZLPR	;SET UP LINE PARAMETERS
3574	016130	005201		2S:	INC	R1	;POSITION POINTER TO THE NEXT LINE
3575	016132	106302			ASLB	R2	;GOT 'EM ALL ?
3576	016134	103370			BCC	1S	;IF NO, GO SET UP THE NEXT LINE
3577	016136	005037	001372		CLR	SAVLIN	;CLEAR LINE # INDICATOR
3578	016142	106437	026170		MTPS	ADZPRT	;SET CPU STATUS TO DZ11 PRIO.
3579	016146	106437	026172		MTPS	ADZLESS1	;MAKE CPU ONE LEVEL LOWER THAN DZ11
3580	016152	113777	001364	163676	MOVW	LINE,ADZTCR	;ENABLE THE VALID LINES
3581	016160			3S:			
3582	016160	012777	016250	163710	MOV	#6S,ADZTIV	;SET UP THE TRANSMITTER INTERRUPT VECTOR
3583	016166	012777	016272	163676	MOV	#7S,ADZRIV	;SET UP THE RECEIVER INTERRUPT VECTOR
3584	016174	013777	026170	163672	MOV	DZPRT,ADZRS	;SET THE INTERRUPT VECTOR STATUS
3585	016202	013777	026170	163670	MOV	DZPRT,ADZTIS	;SET TRANSMITTER INTERRUPT PRIORITY
3586	016210	052777	040040	163624	BIS	#TIE!#SENAB,ADZCSR	;ENABLE THE DEVICE
3587	016216	005277			CLR	R5	
3588	016220	032777	100000	163614	4S:	BIT	#TRDY,ADZCSR
3589	016226	001404			BEQ	5S	
3590	016230	000240			NOP		
3591	016232	000240			NOP		
3592	016234	104007			ERROR	7	;*TRANSMITTER FAILED TO INTERRUPT
3593	016236	000417			BR	8S	
3594	016240	104414		5S:	DELAY		
3595	016242	005205			INC	R5	
3596	016244	001365			BNE	4S	
3597	016246	104003			ERROR	3	;*TRDY NOT SET!
3598	016250	022626		6S:	POP2SP		;REMOVE THE INTERRUPT FROM THE STACK
3599	016252	042777	040000	163562	BIC	#TIE,ADZCSR	;DON'T LET ANY MORE INTERRUPTS OCCUR
3600	016260	106437	026170		MTPS	ADZPRT	;SET CPU STATUS TO DZ11 PRIORITY
3601	016264	106437	026172		MTPS	ADZLESS1	;MAKE CPU ONE LEVEL LOWER THAN DZ11
3602	016270	000402			BR	8S	;RETURN TO THE NORMAL FLOW
3603	016272	104012		7S:	ERROR	12	;*RECEIVER SHOULD NOT INTERRUPT
3604	016274	022626			CMP	(SP)+,(SP)+	;POP FOR FAKE RTI
3605	016276	042777	040000	163536	8S:	BIC	#TIE,ADZCSR
3606	016304	113777	001422	163554	MOVW	TDO,ADZTCR	;RESET TRANSMITTER INTERRUPT ENABLE
3607	016312	012777	016404	163556	MOV	#11S,ADZTIV	;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
3608	016320	012777	016412	163544	MOV	#12S,ADZRIV	;SET UP THE TRANSMITTER INTERRUPT VECTOR
3609	016326	013777	026170	163540	MOV	DZPRT,ADZRS	;SET UP THE RECEIVER INTERRUPT VECTOR
3610	016334	013777	026170	163536	MOV	DZPRT,ADZTIS	;SET THE INTERRUPT VECTOR STATUS
3611	016342	052777	000140	163472	MOV	DZPRT,ADZTIS	;SET TRANSMITTER INTERRUPT PRIORITY
3612	016350	005005			BIS	#RIE!#SENAB,ADZCSR	;ENABLE THE DEVICE
3613	016352	032777	000200	163462	9S:	CLR	R5
3614	016360	001404			BIT	#RDONE,ADZCSR	
3615	016362	000240			BEQ	10S	
3616	016364	000240			NOP		
3617	016366	104011			NOP		
3618	016370	000413			ERROR	11	;*RECEIVER FAILED TO INTERRUPT
3619	016372	104414		10S:	BR	13S	
3620	016374	005205			DELAY		
3621	016376	001365			INC	R5	
3622	016400	104004			BNE	9S	
3623	016402	000406			ERROR	4	;*NO RX DONE! (NOT SET)
3624	016404	104010		11S:	BR	13S	;CONTINUE TEST
3625	016406	022626			ERROR	10	;*TRANSMITTER SHOULD NOT INTERRUPT
					CMP	(SP)+,(SP)+	;POP FOR FAKE RTI

L06

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 77
DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

3626	016410	000403			BR 13\$;CONT TEST
3627	016412	022626		12\$:	POP2SP	;REMOVE THE INTERRUPT FROM THE STACK
3628	016414	005077	163422		CLR JDZCSR	;DON'T ALLOW ANY MORE INTERRUPTS
3629	016420			13\$:		
3630	016420	104413			DEVICE.CLR	;ISSUE DEVICE CLEAR (RESET)
3631						
3632						;***** TEST 26 *****
3633						;*THIS TEST VERIFIES THAT THE RECEIVER WILL
3634						;*INTERRUPT BEFORE THE TRANSMITTER EVEN
3635						;*THOUGH THE TRANSMITTER WAS ENABLED
3636						;*FIRST, SET PS TO LEVEL 7;
3637						;*GET RDONE AND TRDY TO SET;
3638						;*SET TX IE AND RX IE;
3639						;*CLEAR PS AND EXPECT RX TO INTERRUPT FIRST
3640						
3641					::* TEST 26	
3642	016422	000004			TST26: SCOPE	;*****
3643	016424	012737	000026 001122		MOV #26,\$TSTNM	;LOAD THE NUMBER OF THIS TEST
3644	016432	012737	017054 01360		MOV #TST27,NEXT	;POINT TO THE START OF THE NEXT TEST
3645	016440	104417			DCLASM	;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3646	016442	013701	001366		MOV PAR,R1	;PICK UP PARAMETERS
3647	016446	012702	000001		MOV #1,R2	;PICK UP INIT POINTER
3648	016452	030237	001364	1\$:	BIT R2,LINE	;SHOULD THIS LINE BE SET UP ?
3649	016456	001402			BEQ 2\$;NO
3650	016460	010177	163366		MOV R1,JDZLPR	;SET UP LINE PARAMETERS
3651	016464	005201		2\$:	INC R1	;POSITION POINTER TO THE NEXT LINE
3652	016466	106302			ASLB R2	;GOT 'EM ALL ?
3653	016470	103370			BCC 1\$;IF NO, GO SET UP THE NEXT LINE
3654	016472	005037	001372		CLR SAVLIN	;CLEAR LINE # INDICATOR
3655	016476	012777	016726 163366		MOV #8\$,JDZRIV	;SETUP INTERRUPT STUFF
3656	016504	013777	026170 163362		MOV DZPRT,JDZRIS	
3657	016512	012777	017016 163356		MOV #12\$,JDZTIV	
3658	016520	013777	026170 163352		MOV DZPRT,JDZTIS	
3659	016526	052777	000040 163306		BIS #MSENAB,JDZCSR	
3660	016534	012702	000001		MOV #1,R2	;LINE POINTER
3661	016540	030237	001364	3\$:	BIT R2,LINE	;VALID LINE ?
3662	016544	001004			BNE 4\$	
3663	016546	005237	001372		INC SAVLIN	
3664	016552	106302			ASLB R2	
3665	016554	000771			BR 3\$	
3666	016556	106427	000340	4\$:	MTPS #PR7	
3667	016562	000240			NOP	
3668	016564	000240			NOP	
3669	016566	110277	163264		MOVB R2,JDZTCR	;SET TCR BIT
3670	016572	005777	163250		TST JDZRBUF	;VALID DATA?
3671	016576	100001			BPL .+4	;IT BETTER NOT BE SET
3672	016600	104017			ERROR 17	;DATA VALID SHOULD NOT BE SET
3673	016602	105777	163234	5\$:	TSTB JDZCSR	;RECEIVER DONE ?
3674	016606	100001			BPL .+4	
3675	016610	104020			ERROR 20	;RECEIVER DONE BIT SHOULD NOT BE SET
3676	016612	005005			CLR R5	
3677	016614	005004			CLR R4	
3678	016616	005777	163220	99\$:	TST JDZCSR	;WAIT FOR TRDY
3679	016622	100404			BMI 100\$;BR IF READY
3680	016624	104414			DELAY	;STALL TIME
3681	016626	005204			INC R4	

MO6

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 78
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

```

3682 016630 001372      BNE      99$
3683 016632 104003      ERROR    3          ;TRDY FAILED TO SET
3684 016634 105077 163226 100$: CLR      @DZTDR
3685 016640 005004      CLR      R4
3686 016642 032777 000200 163172 6$: BIT      #RDONE,@DZCSR
3687 016650 001004      BNE      7$
3688 016652 104414      DELAY
3689 016654 005204      INC      R4
3690 016656 001371      BNE      6$
3691 016660 104004      ERROR    4          ;*RDONE FAILED TO SET!
3692 016662 005777 163154 7$: TST      @DZCSR      ;TRANS DONE BIT = 1 ?
3693 016666 100401      BMI      +4          ;YES
3694 016670 104003      ERROR    3          ;*NO TRANS DONE FAILED TO SET
3695                          ;NOW THAT BOTH TRANSMITTER AND RECEIVER DONE BIT =1
3696                          ;SET INTERRUPT ENABLES AND WATCH THE FUR FLY
3697 016672 052777 040000 163142  BIS      #TIE,@DZCSR
3698 016700 052777 000100 163134  BIS      #RIE,@DZCSR
3699 016706 106427 000000      MTPS     #0
3700 016712 000240      NOP
3701 016714 000240      NOP
3702 016716 104007      ERROR    7          ;*TRANSMITTER FAILED TO INTERRUPT
3703 016720 104011      ERROR    11         ;*RECEIVER FAILED TO INTERRUPT
3704                          ;CHECK BR LEVEL
3705 016722 000137 017022      JMP      13$      ;GET OUT
3706
3707                          ;RECEIVER INTERRUPT ROUTINE
3708 016726 017704 163114 8$: MOV      @DZRBUF,R4          ;ACTUAL
3709 016732 010403      MOV      R4,R3
3710 016734 000303      SWAB     R3
3711 016736 042703 177770      BIC      #1C<7>,R3      ;STRIP JUNK
3712 016742 105737 001371      TSTB     MODE+1          ;IS THIS TEST IN STAGGERED MODE?
3713 016746 001406      BEQ      11$          ;IF NOT, SKIP STAGGERED SETUP
3714
3715                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3716
3717 016750 006203      ASR      R3          ;GET THE LAST BIT INTO THE CARRY BIT
3718 016752 103402      BCS      9$          ;IF IT IS SET, GO CLEAR IT
3719 016754 000261      SEC
3720 016756 000401      BR       10$         ;IF IT IS CLEAR SET IT HERE
3721 016760 000241      CLC
3722 016762 006103      ROL      R3          ;SKIP THE CLEARING
3723 016764 020337 001372 10$: CMP      R3,SAVLIN      ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3724 016770 001401      BEQ      +4          ;GET THE NEW BIT BACK INTO R3
3725 016772 104015      ERROR    15         ;IS THIS A VALID LINE
3726 016774 042704 177400      BIC      #1C<377>,R4      ;YES
3727 017000 120504      CMPB     R5,R4          ;*INVALID LINE
3728 017002 001401      BEQ      +4          ;STRIP JUNK
3729 017004 104005      ERROR    5          ;DATA COMPARE ?
3730 017006 040277 163044      BIC      R2,@DZTCR      ;YES
3731 017012 022626      POP2SP
3732 017014 000402      BR       13$         ;*DATA DOES NOT COMPARE
3733                          ;CLEAR TCR BIT
3734 017016 104011 12$: ERROR    11         ;REMOVE HE INTERRUPT VECTOR FROM THE STACK
3735                          ;GO GET OUT OF INTERRUPT MODE
3736 017020 022626      POP2SP
3737 017022 042777 040100 163012 13$: BIC      #TIE!RIE,@DZCSR ;THE RECEIVER INTERRUPT FAILED
;TO OVERRIDE THE TRANSMITTER
;REMOVE THE INTERRUPT VECTOR FROM THE STACK
;CLEAR INTERRUPT ENABLES

```

```

3738 017030 013777 002074 163034      MOV    DZRIS,ADZRIV    ;RESTORE TRAPCATCHER
3739 017036 005077 163032              CLR    ADZRIS
3740 017042 013777 002100 163026      MOV    DZTIS,ADZTIV
3741 017050 005077 163024              CLR    ADZTIS
3742                                     ;***** TEST 27 *****
3743                                     ;*THIS TEST VERIFYS OVERRUN AND SILO ALARM
3744                                     ;*ONE LINE AT A TIME - BASED UPON VALID LINES
3745                                     ;*AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
3746                                     ;*TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN
3747                                     ;*EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
3748                                     ;*SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
3749                                     ;*CHAR PULLED OUT OUT THE SILO.
3750                                     ;::* TEST 27
3751                                     ;*****
3752 017054 000004      TST27: SCOPE
3753 017056 012737 000027 001122      MOV    #27,STSTNM    ;LOAD THE NUMBER OF THIS TEST
3754 017064 012737 017602 001360      MOV    #TST30,NEXT  ;POINT TO THE START OF THE NEXT TEST
3755 017072 012737 017506 001362      MOV    #18$,LOCK    ;SET FOR LOOP
3756 017100 104417      DCLASM              ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3757 017102 013701 001366      MOV    PAR,R1       ;PICK UP PARAMETERS
3758 017106 012702 000001      MOV    #1,R2        ;PICK UP INIT POINTER
3759 017112 030237 001364      1$: BIT    R2,LINE   ;SHOULD THIS LINE BE SET UP ?
3760 017116 001402      BEQ    2$           ;NO
3761 017120 010177 162726      MOV    R1,ADZLPR    ;SET UP LINE PARAMETERS
3762 017124 005201      2$: INC    R1        ;POSITION POINTER TO THE NEXT LINE
3763 017126 106302      ASLB   R2           ;GOT 'EM ALL ?
3764 017130 103370      BCC   1$           ;IF NO, GO SET UP THE NEXT LINE
3765 017132 005037 001372      CLR    SAVLIN       ;CLEAR LINE # INDICATOR
3766 017136 012700 001422      MOV    #TDO,RO      ;POINT TO THE DATA AREA
3767 017142 005020      CLR    (RO)+        ;CLEAR A DATA WORD
3768 017144 022700 001462      CMP    #STOP,RO     ;FINISHED ?
3769 017150 001374      BNE   -6            ;NO
3770 017152 005000      CLR    RO           ;CLEAR OFFSET
3771 017154 012702 000001      MOV    #1,R2        ;LINE POINTER
3772 017160 052777 010040 162654      BIS    #MSENAB!SILOEN,ADZCSR ;START SCANNER & SET SILO ENABLE
3773 017166 030237 001364      3$: BIT    R2,LINE   ;VALID LINE?
3774 017172 001002      BNE   +6            ;YES
3775 017174 000137 017462      JMP    17$         ;TRY NEXT LINE
3776 017200 013700 001372      MOV    SAVLIN,RO   ;MAKE OFFSET
3777 017204 006300      ASL   RO           ;MAKE POWER OF TWO
3778 017206 010277 162644      MOV    R2,ADZTCR   ;SET TCR BIT
3779 017212 105777 162624      4$: TSTB   ADZCSR   ;REC DONE = 1 ?
3780 017216 100001      BPL   +4            ;NO
3781 017220 104020      ERROR 20           ;REC DONE SHOULD NOT = 1
3782 017222 005003      CLR    R3           ;SET CHARACTER COUNT
3783 017224 005004      5$: CLR    R4
3784 017226 032777 100000 162606 6$: BIT    #TRDY,ADZCSR
3785 017234 001004      BNE   7$           ;NO
3786 017236 104414      DELAY              ;
3787 017240 105204      INCB   R4          ;
3788 017242 001371      BNE   6$          ;
3789 017244 104003      ERROR 3           ;*TRDY FAILED TO SET
3790 017246 116077 001422 162612 7$: MOVB   TDO(RO),ADZTOR ;LOAD A CHARACTER
3791 017254 005260 001422      INC    TDO(RO)     ;SET UP NEXT CHARACTER
3792 017260 020327 000017      CMP    R3,#15.    ;16 CHARACTERS ?
3793 017264 103006      BHIS  8$

```

```

3794 017266 032777 020000 162546 BIT #SILOAL, DZCSR ;SILO ALARM = 0 ?
3795 017274 001401 BEQ +4 ;YES
3796 017276 104013 ERROR 13 ;*SILO ALARM SHOULD NOT = 1
3797 ;UNTIL 16. DATA CHARACTERS
3798 017300 000411 BR 10$
3799 017302 005004 CLR R4
3800 017304 032777 020000 162530 8$: BIT #SILOAL, DZCSR
3801 017312 001004 BNE 10$
3802 017314 104414 DELAY
3803 017316 005204 INC R4
3804 017320 001371 BNE 9$
3805 017322 104014 ERROR 14 ;*SILO ALARM FAILED TO SET!
3806 ;SILO ALARM SHOULD =1 AFTER 16.
3807 ;DATA CHARACTERS
3808 017324 005203 10$: INC R3
3809 017326 022703 000102 CMP #66., R3
3810 017332 001334 BNE 5$
3811 017334 005004 CLR R4
3812 017336 104414 DELAY
3813 017340 105204 INCB R4
3814 017342 001375 BNE -4
3815 ;NOW LETS READ THE SILO
3816 017344 013705 001372 MOV SAVLIN, R5 ;MAKE EXPECTED LINE #
3817 017350 105737 001371 TSTB MODE+1 ;IS THIS TEST IN STAGGERED MODE?
3818 017354 001406 BEQ 13$ ;IF NOT, SKIP STAGGERED SETUP
3819
3820 ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3821
3822 017356 006205 ASR R5 ;GET THE LAST BIT INTO THE CARRY BIT
3823 017360 103402 BCS 11$ ;IF IT IS SET, GO CLEAR IT
3824 017362 000261 SEC ;IF IT IS CLEAR SET IT HERE
3825 017364 000401 BR 12$ ;SKIP THE CLEARING
3826 017366 000241 11$: CLC ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3827 017370 006105 12$: ROL R5 ;GET THE NEW BIT BACK INTO R5
3828 017372 000305 13$: SWAB R5 ;PUT IN UPPER BYTE
3829 017374 052705 100000 BIS #DVALID, R5 ;ADD DATA VALID
3830 017400 017704 162442 14$: MOV DZRBUF, R4 ;ACTUAL
3831 017404 020405 CMP R4, R5 ;ACTUAL VS. EXPECTED
3832 017406 001401 BEQ 15$ ;YES
3833 017410 104006 ERROR 6 ;*DATA/CONTENTS DID NOT COMPARE
3834 017412 032777 020000 162422 15$: BIT #SILOAL, DZCSR ;SILO ALARM= 0 ?
3835 017420 001401 BEQ 16$ ;YES
3836 017422 104016 ERROR 16 ;READING DZRBUF DID NOT CLEAR SILO ALARM
3837 017424 005205 16$: INC R5 ;UP CHARACTER
3838 017426 120527 000077 CMPB R5, #63. ;LAST SILO CHAR ?....64TH CHAR
3839 017432 101762 BLOS 14$
3840 017434 005205 INC R5 ;ADD 1 MORE FOR THE CLOBBERED CHAR
3841 017436 052705 040000 BIS #OVRUN, R5 ;ADD OVERRUN TO EXPECTED
3842 017442 120527 000101 CMPB R5, #65. ;LAST CHARACTER ?
3843 017446 001754 BEQ 14$
3844 017450 017704 162372 MOV DZRBUF, R4 ;FOR GOOD MEASURE
3845 017454 005704 TST R4 ;DATA VALID SHOULD = 0
3846 017456 100001 BPL 17$ ;YES
3847 017460 104017 ERROR 17 ;DATA VALID SHOULD = 0
3848 017462 040277 162370 17$: BIC R2, DZTCR ;CLR TCR BIT
3849 017466 104401 SCOP1 ;LOOP?

```



```

3850 017470 005237 001372          INC      SAVLIN      ;INC EXPECTED LINE
3851 017474 106302          ASLB     R2           ;NEXT LINE
3852 017476 103402          BCS     .+6          ;NO
3853 017500 000137 017166          JMP      35          ;YES
3854 017504 104400          ADVANCE          ;GO TO NEXT TEST
3855
3856
3857 ;TIGHT SCOPE LOOP FOR THIS TEST. SENDS 20. CHARACTERS
3858 ;ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
3859 ;USED TO SCOPE SILO ALARM PULSES, ETC.
3860 ;SPECIAL THANKS TO WALT BYSKO FOR THIS TEST
3861 017506 052777 010040 162326 18$:  BIS     #MSENAB!SILOEN,0DZCSR ;SETUP DEVICE
3862 017514 012777 017572 162354      MOV     #20$,0DZTIV ;SETUP TRANSMITTER VECTOR
3863 017522 012737 000024 001216      MOV     #20$,STMP0 ;TEMPORARY COUNT OF CHARACTER BURST
3864 017530 050277 162322          BIS     R2,0DZTCR ;ENABLE LINE
3865 017534 052777 040000 162300      BIS     #TIE,0DZCSR ;ENABLE INTERRUPTS
3866 017542 106427 000000          MTPS    #0          ;LOWER PRIORITY
3867 017546 000001          19$:  WAIT          ;ALLOW INTERRUPTS
3868 017550 005337 001216      DEC     STMP0        ;REDUCE COUNT. ALL CHARACTERS SENT?
3869 017554 001374          BNE     19$         ;IF NO, WAIT FOR MORE
3870 017556 042777 050040 162256      BIC     #SILOEN!MSENAB!TIE,0DZCSR ;RESET SILO COUNTER, CLEAR STROBE
3871 017564 104401          SCOP1          ;LOOP AGAIN?
3872 017566 000137 017462          JMP     17$         ;IF NOT, RETURN TO WHERE YOU LEFT OFF
3873 017572 112777 000252 162266 20$:  MOVB   #252,0DZTDR ;SEND A CHARACTER
3874 017600 000002          RTI          ;ALLOW MORE CHARACTERS TO COME
3875 ;***** TEST 30 *****
3876 ;*THIS TEST THAT "SILO ENABLE" WILL INHIBIT
3877 ;*RECEIVER INTERRUPTS AND THAT ON THE
3878 ;*16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
3879 ;*INTERRUPT WITH "RIE" SET.
3880 ;*THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
3881 ;::* TEST 30
3882 ;*****
3883 017602 000004          TST30: SCOPE
3884 017604 012737 000030 001122      MOV     #30$,STSTNM ;LOAD THE NUMBER OF THIS TEST
3885 017612 012737 020164 001360      MOV     #TST31,NEXT ;POINT TO THE START OF THE NEXT TEST
3886 017620 012737 017714 001362      MOV     #35$,LOCK ;SET FOR LOOP
3887 017626 104417          DCLASM          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3888 017630 013701 001366          MOV     PAR,R1     ;PICK UP PARAMETERS
3889 017634 012702 000001          MOV     #1,R2     ;PICK UP INIT POINTER
3890 017640 030237 001364          1$:  BIT     R2,LINE ;SHOULD THIS LINE BE SET UP ?
3891 017644 001402          BEQ     25$       ;NO
3892 017646 010177 162200          MOV     R1,0DZLPR ;SET UP LINE PARAMETERS
3893 017652 005201          2$:  INC     R1     ;POSITION POINTER TO THE NEXT LINE
3894 017654 106302          ASLB   R2         ;GOT 'EM ALL ?
3895 017656 103370          BCC    1$        ;IF NO, GO SET UP THE NEXT LINE
3896 017660 005037 001372          CLR     SAVLIN    ;CLEAR LINE # INDICATOR
3897 017664 012700 001422          MOV     #TDO,RO  ;POINT TO THE DATA AREA
3898 017670 005020          CLR     (RO)+    ;CLEAR A DATA WORD
3899 017672 022700 001462          CMP     #STOP,RO ;FINISHED ?
3900 017676 001374          BNE     .-6       ;NO
3901 017700 005000          CLR     RO       ;CLEAR OFFSET
3902 017702 012702 000001          MOV     #1,R2    ;LINE POINTER
3903 017706 052777 010140 162126      BIS     #MSENAB!SILOEN!RIE,0DZCSR ;START SCANNER & SET SILO ENABLE
3904
3905 017714 030237 001364          3$:  BIT     R2,LINE ;VALID LINE?

```

3906	017720	001002				BNE	+6		:YES
3907	017722	000137	020136			JMP	17\$:TRY NEXT LINE
3908	017726	005777	162114			TST	ADZRBUF		:EMPTY THE SILO
3909	017732	100775				BMI	-4		:BR IF DATA VALID IS SET!
3910	017734	106427	000000			MTPS	#0		:SET PROCESSOR PRIORITY TO 0
3911	017740	012777	020126	162124		MOV	#11\$,ADZRIV		:SET FOR UNEXPECTED INTER.
3912	017746	012777	000340	162120		MOV	#PR7,ADZRI5		:SET PRIO.
3913	017754	013700	001372			MOV	SAVLIN,R0		:MAKE OFFSET
3914	017760	006300				ASL	R0		:MAKE POWER OF TWO
3915	017762	010277	162070			MOV	R2,ADZTCR		:SET TCR BIT
3916	017766	005004			5\$:	CLR	R4		
3917	017770	032777	100000	162044	6\$:	BIT	#TRDY,ADZCSR		
3918	017776	001004				BNE	7\$		
3919	020000	104414				DELAY			
3920	020002	005204				INC	R4		
3921	020004	001371				BNE	6\$		
3922	020006	104003				ERROR	3		:*TRDY FAILED TO SET
3923	020010	116077	001422	162050	7\$:	MOVB	TDO(R0),ADZTDR		:LOAD A CHARACTER
3924	020016	005260	001422			INC	TDO(R0)		:SET UP NEXT CHARACTER
3925	020022	022760	000017	001422		CMP	#15.,TDO(R0)		:15 CHARS YET?
3926	020030	001406				BEQ	8\$		
3927	020032	032777	020000	162002		BIT	#SILOAL,ADZCSR		:SILO ALARM = 0 ?
3928	020040	001401				BEQ	+4		:YES
3929	020042	104013				ERROR	13		:*SILO ALARM SHOULD NOT = 1
3930									:UNTIL 16. DATA CHARACTERS
3931	020044	000751				BR	6\$		
3932	020046	012777	020134	162016	8\$:	MOV	#12\$,ADZRIV		:SET NEW VECTOR
3933	020054	032777	100000	161760		BIT	#TRDY,ADZCSR		:READY FOR 16TH CHAR
3934	020062	001774				BEQ	-6		
3935	020064	016077	001422	161774		MOV	TDO(R0),ADZTDR		:LOAD THE 16TH CHAR.
3936	020072	005004				CLR	R4		
3937	020074	032777	020000	161740	9\$:	BIT	#SILOAL,ADZCSR		
3938	020102	001005				BNE	10\$		
3939	020104	104414				DELAY			
3940	020106	005204				INC	R4		
3941	020110	001371				BNE	9\$		
3942	020112	104014				ERROR	14		:*SILO ALARM FAILED TO SET!
3943	020114	000410				BR	17\$:SILO ALARM SHOULD =1 AFTER 16.
3944									:DATA CHARACTERS
3945	020116	000240			10\$:	.P			:STALL
3946	020120	000240				NOP			
3947	020122	104000				ERROR			:SILO ALARM NOT INTERRUPTING.
3948	020124	000404				BR	17\$:CONTINUE TEST.
3949	020126	022626			11\$:	CMP	(SP)+,(SP)+		:FAKE RTI
3950	020130	104012				ERROR	12		:RX SHOULD NOT INTERRUPT
3951	020132	000401				BR	17\$:CONTINUE
3952	020134	022626			12\$:	CMP	(SP)+,(SP)+		:GOOD INTERRUPT TO HERE.
3953	020136	040277	161714		17\$:	BIC	R2,ADZTCR		:CLR TCR BIT
3954	020142	104401				SCOPI			:LOOP?
3955	020144	005237	001372			INC	SAVLIN		:INC EXPECTED LINE
3956	020150	106302				ASLB	R2		:NEXT LINE
3957	020152	103402				BCS	+6		:NO
3958	020154	000137	017714			JMP	3\$:YES
3959	020160	005037	001362			CLR	LOCK		:CLEAR TIGHT LOOP FOR NEXT TEST

F07

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 84
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

4016	020430	042703	177770		BIC	#1C<7>,R3	;STRIP JUNK
4017	020434	010304			MOV	R3,R4	;SAVE
4018	020436	012702	000001		MOV	#1,R2	;SET UP POSITION POINTER
4019	020442	105303		3\$:	DECB	R3	;IS IT THIS LINE ?
4020	020444	100402			BMI	4\$;YES
4021	020446	006302			ASL	R2	;UP THE LINE #
4022	020450	000774			BR	3\$;GO 'ROUND AGAIN
4023	020452	030237	001364	4\$:	BIT	R2,LINE	;VALID LINE?
4024	020456	001001			BNE	.+4	;YES
4025	020460	104011			ERROR	11	;NO, INVALID LINE!!!!
4026	020462	042704	177770		BIC	#1C<7>,R4	;STRIP JUNK
4027	020466	006304			ASL	R4	;MAKE POWER OF 2
4028	020470	116477	001422	161370	MOVB	TDO(R4),DZTDR	;LOAD CHARACTER
4029	020476	105264	001422		INCB	TDO(R4)	;SET UP NEXT CHARACTER
4030	020502	001002			BNE	5\$;LAST CHARACTER ?
4031	020504	040277	161346		BIC	R2,DZTCR	;YES, CLEAR TCR BIT
4032	020510	005200		5\$:	INC	R0	;INCR RECEIVER TIMER
4033	020512	000002			RTI		
4034							
4035							
4036							
4037	020514	105777	161322		.REC INTR SVC	ROUTINE	
4038	020520	100401			RXSVC: TSTB	DZCSR	;REC DONE ?
4039	020522	104004			BMI	.+4	;YES
4040	020524	032777	020000	161310	ERROR	4	;FALSE INTERRUPT
4041	020532	001401			BIT	#SILOAL,DZCSR	;SILO ALARM?
4042	020534	104000			BEQ	.+4	;NO
4043	020536	017704	161304		ERROR		;SILO ALARM SHOULD NOT =1
4044	020542	100401			MOV	DZRBUF,R4	;SAVE IT
4045	020544	104000			BMI	.+4	;YES
4046	020546	032704	070000		ERROR		;YOU LOSE ...DATA VALID WAS'NT SET
4047	020552	001401			BIT	#OVRUN!FRMERR!PARER,R4	
4048	020554	104000			BEQ	.+4	
4049	020556	010403			ERROR		;RECEIVER ERROR FLAG/S WERE SET
4050	020560	000303			MOV	R4,R3	
4051	020562	042703	177770		SWAB	R3	
4052	020566	010337	001372		BIC	#1C<7>,R3	;STRIP JUNK
4053	020572	012702	000001		MOV	R3,SAVLIN	;SAVE LINE NUMBER
4054	020576	105303		5\$:	MOV	#1,R2	;SET UP POSITION POINTER
4055	020600	100402			DECB	R3	
4056	020602	006302			BMI	6\$	
4057	020604	000774			ASL	R2	;RE POSITION POINTER
4058	020606	030237	001364	6\$:	BR	5\$;GO 'ROUND AGAIN
4059	020612	001001			BIT	R2,LINE	;LINE VALID ?
4060	020614	104011			BNE	.+4	;YES
4061	020616	013703	001372		ERROR	11	;INVALID LINE #
4062	020622	006303			MOV	SAVLIN,R3	;GET THE LINE NUMBER AGAIN
4063	020624	126304	001442		ASL	R3	;USE R3 AS A POINTER IN THE DATA TABLE
4064	020630	001405			CMPB	TRO(R3),R4	;DOES THE DATA CHARACTER COMPARE ?
4065	020632	016305	001442		BEQ	2\$;YES
4066	020636	042704	177400		MOV	TRO(R3),R5	;SAVE EXPECTED
4067					BIC	#1C<377>,R4	;CLEAR JUNK
4068							;R2 = LINE # BY BIT POSITION
4069							;R4 = ACTUAL DATA
4070	020642	104005					;R5 = EXPECTED DATA
4071	020644	005263	001442	2\$:	ERROR	5	;*NO, DATA DOES NOT COMPARE
					INC	TRO(R3)	;SET UP FOR NEXT CHARACTER

G07

```

4072 020650 105763 001442      TSTB   TR0(R3) ;ALL CHARS DONE?
4073 020654 001002              BNE     +6
4074 020656 040237 020770      BIC     R2,RXTCR ;ZERO LINE DONE INDICATOR.
4075 020662 012716 020342      MOV     #SNAP,(SP) ;RESET THE BACKGROUND TIMING LOOP
4076 020666 000002              RTI
4077
4078
4079
4080 020670 106427 000340      OUT:   MTPS   #PR7 ;STOP ALL INTERRUPTS
4081 020674 104413              DEVICE.CLR ;CLEAR ALL INTERRUPTS AWAY
4082 020676 005003              CLR     R3
4083 020700 005037 001372      CLR     SAVLIN
4084 020704 012702 000001      MOV     #1,R2
4085 020710 030237 001364      1$:    BIT     R2,LINE ;VALID LINE ?
4086 020714 001405              BEQ     2$ ;NO
4087 020716 022763 000400 001442      CMP     #400,TR0(R3) ;RECEIVED A BINARY COUNT PATTERN ?
4088 020724 001401              BEQ     +4 ;YES
4089 020726 104000              ERROR   0 ;THE LINE FAILED TO RECEIVE A FULL
4090 ;BINARY COUNT PATTERN
4091 020730 005237 001372      2$:    INC     SAVLIN ;SET UP FOR NEXT LINE
4092 020734 005723              TST     (R3)+ ;ADD 2
4093 020736 106302              ASLB   R2 ;SET UP NEXT LINE POINTER
4094 020740 103363              BCC    1$ ;FINISHED ?
4095 020742
4096 020742 013777 002074 161122      FINI:  MOV     DZCRIS,DZCRIV ;RESTORE TRAPCATCHER
4097 020750 005077 161120              CLR     DZCRIS
4098 020754 013777 002100 161114      MOV     DZTRIS,DZTRIV
4099 020762 005077 161112              CLR     DZTRIS
4100 020766 104400              ADVANCE
4101 020770 000000      RXTCR: 0 ;GO TO THE NEXT TEST
         ;RX IMAGE OF TCR BITS

```

```

***** TEST 32 *****
;*DZ11 RELATIVE TIMING TEST.
;*EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
;*AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
;*WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
;*DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
;*THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
;* AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
;*PARAMETERS ARE:
;* EIGHT BITS/PER/CHAR - TWO STOP BITS AT
;* 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
;* 2400, 3600, 4800, 7200, 9600 BAUD.
;* 19.2 K BAUD - TWO STOP BITS AT
;* SEVEN, SIX, FIVE BITS/PER/CHAR.
;*AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
;*THE NEXT SELECTED LINE IS THE TESTED.

```

```

::* TEST 32
*****
TST32: SCOPE
MOV     #32,$TSTNM ;LOAD THE NUMBER OF THIS TEST
MOV     #2,$TIMES
MOV     #TST33,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV     #3,$LOCK ;SET FOR LOOP
CLR     OFFSET ;RESET THIS VARIABLE

```

```

4122 020772 000004
4123 020774 012737 000032 001122
4124 021002 012737 000002 001226
4125 021010 012737 021466 001360
4126 021016 012737 021142 001362
4127 021024 005037 023112

```

H07

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 86
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

4129	021030	005037	001372				CLR	SAVLIN	:RESET LINE NUMBER INDICATOR
4129	021034	005037	001374				CLR	XMTLIN	:USE THIS WORD TO TELL WHAT LINE TRANSMITTED
4130	021040	012737	000001	001216			MOV	#1,STMP0	:USE STMP0 AS A BIT POINTER
4131	021046	012737	010070	021464			MOV	#RCVON!550!EIGHT	:TWO STOP,7S :BUILD TEMPORARY PARAMETERS
4132	021054	033737	001216	001364	1S:		BIT	STMP0,LINE	:IS THIS LINE ACTIVE?
4133	021062	001027					BNE	3S	:IF SO, GO GET STARTED
4134	021064	012737	010070	021464	2S:		MOV	#RCVON!550!EIGHT	:TWO STOP,7S :LOAD PARAMETERS TEMPORARILY
4135	021072	012700	001422				MOV	#TDO,RO	:POINT TO THE DATA AREA
4136	021076	005020					CLR	(RO)+	:CLEAR A DATA WORD
4137	021100	022700	001462				CMP	#STOP,RO	:FINISHED ?
4138	021104	001374					BNE	.-6	:NO
4139	021106	005237	001374				INC	XMTLIN	:POINT TO THE NEXT LINE TO TRANSMIT
4140	021112	042737	000007	021464			BIC	#7,7S	:MAKE SURE TEMPORARY PARAMETERS POINT TO 0
4141	021120	053737	001374	021464			BIS	XMTLIN,7S	:ADD DESIRED LINE NUMBER
4142	021126	005037	023112				CLR	OFFSET	
4143	021132	106337	001216				ASLB	STMP0	:POINT TO THE NEXT LINE
4144	021136	103346					BCC	1S	:PROCESS THE NEXT LINE
4145	021140	104400					ADVANCE		:TEST TO SEE IF THIS TEST GETS REPEATED
4146	021142				3S:				
4147	021142	104417					DCLASH		:CLEAR DEVICE AND SET MAINT BIT IF I MODE
4148	021144	042737	010000	021464			BIC	#RCVON,7S	:ZERO PARAMTERS FOR TX LINE
4149	021152	013777	021464	160672			MOV	7S,ADZLPR	:LOAD PARAMTERS FOR TX
4150	021160	005737	001370				TST	MODE	:STAGGERED?
4151	021164	100011					BPL	100S	:BR IF NO
4152	021166	000241					CLC		:SET UP LINE
4153	021170	006037	021464				ROR	7S	
4154	021174	103002					BCC	98S	:BR IF LINE WAS EVEN
4155	021176	000241					CLC		:PREPARE TO MKE LINE EVEN
4156	021200	000401					BR	99S	:CONTINUE
4157	021202	000261			98S:		SEC		:PREPARE TO MAKE LINE ODD
4158	021204	006137	021464		99S:		ROL	7S	:SET ALTERED LINE
4159	021210	052737	010000	021464	100S:		BIS	#RCVON,7S	:SET RX ON
4160	021216	013777	021464	160626			MOV	7S,ADZLPR	:LOAD RX PARAMETERS
4161	021224	042737	000007	021464			BIC	#7,7S	:CLEAR OLD LINE #
4162	021232	053737	001374	021464			BIS	XMTLIN,7S	:SET LINE UP AGAIN
4163	021240	013737	021464	001400			MOV	7S,REGIST	:SAVE PARAMETERS FOR PRINTOUT
4164	021246	012700	001422				MOV	#TDO,RO	:POINT TO THE DATA AREA
4165	021252	005020					CLR	(RO)+	:CLEAR A DATA WORD
4166	021254	022700	001462				CMP	#STOP,RO	:FINISHED ?
4167	021260	001374					BNE	.-6	:NO
4168	021262	005002					CLR	R2	:USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
4169	021264	005003					CLR	R3	:USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
4170	021266	005037	001220				CLR	STMP1	:INITIALIZE THE TIMER
4171	021272	005037	001224				CLR	STMP3	:INITIALIZE THESE BITS ALSO
4172	021276	012737	000020	001376			MOV	#20,XMTCNT	:SET HOW MANY CHARACTERS TO TRANSMIT
4173	021304	012777	022552	160564			MOV	#XMTSRV,ADZTIV	
4174	021312	012777	022676	160552			MOV	#RXISR1,ADZRIV	
4175	021320	013777	026170	160546			MOV	DZPRT,ADZRIS	
4176	021326	013777	026170	160544			MOV	DZPRT,ADZTIS	
4177	021334	113777	001216	160514			MOV	STMP0,ADZTCR	:START THE VALID LINE
4178	021342	052777	040140	160472			BIS	#TIE!RIE!MSENAB,ADZCSR	
4179	021350	106427	000000				MTPS	#0	:LOWER THE PRIORITY TO ALLOW INTERRUPTS
4180	021354	032777	000100	160460	4S:		BIT	#RIE,ADZCSR	:IS ROUTINE DONE?
4181	021362	001407					BEQ	5S	:WHEN ALL IS DONE RX IE IS CLEARED IN ISR.
4182	021364	005237	001220				INC	STMP1	:COUNT TIME
4183	021370	001371					BNE	4S	:CONTINUE TEST


```

4240 021616 022700 001462      CMP      #STOP,R0      ;FINISHED ?
4241 021622 001374              BNE      .-6          ;NO
4242 021624 005000              CLR      R0          ;CLEAR OFFSET
4243 021626 012777 022552 160242  MOV      #XMTSRV, @DZTIV ;SET UP THE TRANSMITTER INTERRUPT VECTOR
4244 021634 012777 021742 160230  MOV      #9$, @DZRIV   ;SET UP THE RECEIVER INTERRUPT VECTOR
4245 021642 013777 026170 160224  MOV      DZPRT, @DZRRIS ;SET THE INTERRUPT VECTOR STATUS
4246 021650 013777 026170 160222  MOV      DZPRT, @DZTIS  ;SET TRANSMITTER INTERRUPT PRIORITY
4247 021656 052777 040140 160156  BIS      #RIE!TIE!MSENAB, @DZCSR ;ENABLE THE DEVICE
4248 021664 113777 001364 160164  MOVVB   LINE, @DZTCR   ;ENABLE ALL SELECTED LINES
4249 021672 106427 000000              MTPS     #0          ;ALLOW INTERRUPTS
4250 021676 005037 021736      4$:     CLR      7$
4251 021702 005037 021740              CLR      8$
4252 021706 032777 000100 160126  5$:     BIT      #RIE, @DZCSR   ;WHEN RX DONE; RIE WILL =0
4253 021714 001407              BEQ      6$          ;BR IF ALL DONE
4254 021716 005237 021736              INC      7$
4255 021722 001371              BNE      5$
4256 021724 105237 021740              INCB    8$
4257 021730 100366              BPL     5$
4258 021732 104011              ERROR   11          ;*RX FAILED TO FINISH (INTERRUPT)
4259 021734 104400              6$:     ADVANCE ;ADVANCE LOOP
4260 021736 000000              7$:     0
4261 021740 000000              8$:     0
4262
4263
4264
4265
4266 021742 017704 160100      9$:     MOV      @DZRBUF, R4   ;GET THE CHARACTER
4267 021746 100401              BMI     10$          ;IF IT WAS VALID, CONTINUE TESTING
4268 021750 104000              ERROR  ;ERROR- ILLEGAL CHAR... DATA VALID NOT SET
4269 021752 010401              10$:    MOV      R4, R1     ;COPY THE RECEIVED INFORMATION
4270 021754 000301              SWAB   R1           ;GET THE LINE NUMBER IN THE LOWER BYTE
4271 021756 042701 177770      BIC     #1C<7>, R1   ;ISOLATE THE LINE NUMBER
4272 021762 006301              ASL    R1           ;ALIGN IT ON A WORD BOUNDARY
4273 021764 032704 010000      BIT     #PARER, R4   ;PARITY ERROR SHOULD BE SET. IS IT?
4274 021770 001013              BNE    11$          ;IF SO, GO CHECK CHARACTER
4275 021772 013737 002046 001400  MOV     DZRBUF, REGIST ;SET UP FOR THE ERROR MESSAGE
4276 022000 010405              MOV     R4, R5
4277 022002 042705 000377      BIC     #377, R5
4278 022006 156105 001442      BISB   TRO(R1), R5   ;GET THE CORRECT CHARACTER
4279 022012 052705 110000      BIS    #DVALID!PARER, R5 ;BUILD WHAT WAS EXPECTED
4280 022016 104006              ERROR  6           ;*ERROR- DID NOT GET CORRECT INFORMATION
4281 022020 126104 001442      11$:   CMPB   TRO(R1), R4   ;CHECK THE CHARACTER. IS IT CORRECT?
4282 022024 001413              BEQ    12$          ;IF SO, GO SET UP NEXT CHARACTER
4283 022026 116105 001442      MOVVB  TRO(R1), R5   ;LOAD THE CHARACTER FOR ERROR REPORTING
4284 022032 042705 177400      BIC    #1C<377>, R5 ;CLEAR SIGN EXTEND
4285 022036 010137 001372      MOV    R1, SAVLIN   ;GET THE LINE NUMBER FOR REPORTING
4286 022042 006237 001372      ASR    SAVLIN        ;ALIGN IT CORRECTLY
4287 022046 042704 177400      BIC    #1C<377>, R4 ;REMOVE THE JUNK FROM R4, THE ACTUAL CHARACTER
4288 022052 104005              ERROR  5           ;DATA ERROR
4289 022054 005261 001442      12$:   INC    TRO(R1)     ;SET UP THE NEXT CHARACTER
4290 022060 005203              INC    R3           ;ADD TO THE TOTAL RECEIVED COUNT
4291 022062 032777 040000 157752  BIT     #TIE, @DZCSR  ;ARE TRANSMISSIONS DONE?
4292 022070 001010              BNE    13$          ;IF NO, GO RECEIVE SOME MORE
4293 022072 020203              CMP    R2, R3       ;ARE ALL CHARACTERS RECEIVED?
4294 022074 001006              BNE    13$          ;IF NO, GO RECEIVE SOME MORE
4295 022076 042777 000100 157736  BIC    #RIE, @DZCSR  ;DISABLE RECEIVER INTERRUPTS

```

;RECEIVER SERVICE ROUTINE

K07

```

4296 022104 012716 021734      MOV      #6$, (SP)      ;CRUNCH THE STACK
4297 022110 000002              RTI                    ;RETURN AND FINISH
4298 022112 012716 021676      13$:  MOV      #4$, (SP)      ;CRUNCH THE STACK
4299 022116 000002              RTI                    ;GO BACK TO RECEIVER WAIT LOOP
4300                                     ;***** TEST 34 *****
4301                                     ;*THIS TEST VERIFIES THAT ODD PARITY WORKS FOR ALL ODD LINES
4302                                     ;* SELECTED AND THAT EVEN PARITY WORKS FOR ALL EVEN LINES SELECTED
4303                                     ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
4304                                     ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
4305                                     ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
4306                                     ;*YOU ARE IN "STAGGERED" MODE.
4307                                     ;*40(8) CHARS ARE USED FOR THIS TEST.
4308                                     ;*ALL SELECTED LINES WILL BE ENABLED
4309                                     ;*AT THE SAME TIME!
4310
4311                                     ;::* TEST 34
4312                                     ;:*****
4313 022120 000004      TST34: SCOPE
4314 022122 012737 000034 001122  MOV      #34, $STSTNM   ;LOAD THE NUMBER OF THIS TEST
4315 022130 012737 004432 001360  MOV      #SEOP, NEXT    ;POINT TO THE END-OF-PASS HANDLER
4316 022136 005737 001370      TST      MODE          ;IS THIS STAGGERED MODE?
4317 022142 100111      BPL      6$            ;IF NOT, DON'T DO THIS TEST
4318 022144 104417      DCLASM              ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4319 022146 013701 001366  MOV      PAR, R1        ;USE R1 TO BUILD PARAMETERS TO BE LOADED
4320 022152 042701 000200  BIC      #ODDPAR, R1    ;MAKE SURE ODD PARITY ISN'T SET
4321 022156 052701 000100  BIS      #PARITY, R1    ;MAKE SURE PARITY IS TURNED ON
4322 022162 012702 000001  MOV      #1, R2         ;USE R2 AS A LINE POINTER
4323 022166 030237 001364      1$:  BIT      R2, LINE     ;IS THIS A VALID LINE?
4324 022172 001411      BEQ      3$            ;IF NOT, SKIP TO THE NEXT LINE
4325 022174 032701 000001  BIT      #BIT0, R1      ;IS THIS LINE AN ODD LINE?
4326 022200 001402      BEQ      2$            ;IF IT'S EVEN, USE EVEN PARITY
4327 022202 052701 000200  BIS      #ODDPAR, R1    ;IF IT'S ODD, USE ODD PARITY
4328 022206 010177 157640      2$:  MOV      R1, JDZLPR   ;LOAD THE LINE PARAMETER REGISTER
4329 022212 042701 000200  BIC      #ODDPAR, R1    ;SET UP THE NEXT PARITY TO EVEN
4330 022216 005201      3$:  INC      R1           ;POINT TO THE NEXT LINE
4331 022220 106302      ASLB     R2            ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
4332 022222 103361      BCC     1$            ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
4333 022224 005037 001372      CLR     SAVLIN        ;CLEAR THE LINE NUMBER INDICATOR
4334 022230 005002      CLR     R2           ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
4335 022232 005003      CLR     R3           ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
4336 022234 012737 000040 001376  MOV      #40, XMTCNT    ;TRANSMIT A BINARY COUNT PATTERN(00-40)
4337 022242 012700 001422      MOV      #TDD, RO      ;POINT TO THE DATA AREA
4338 022246 005020      CLR     (RO)+         ;CLEAR A DATA WORD
4339 022250 022700 001462      CMP     #STOP, RO      ;FINISHED ?
4340 022254 001374      BNE     -6            ;NO
4341 022256 005000      CLR     RO            ;CLEAR OFFSET
4342 022260 012777 022552 157610  MOV      #XMTSRV, JDZTIV ;SET UP THE TRANSMITTER INTERRUPT VECTOR
4343 022266 012777 022374 157576  MOV      #9$, JDZRIV    ;SET UP THE RECEIVER INTERRUPT VECTOR
4344 022274 013777 026170 157572  MOV      DZPRT, JDZRI5 ;SET THE INTERRUPT VECTOR STATUS
4345 022302 013777 026170 157570  MOV      DZPRT, JDZTIS  ;SET TRANSMITTER INTERRUPT PRIORITY
4346 022310 052777 040140 157524  BIS      #RIE!TIE!MSENAB, JDZCSR ;ENABLE THE DEVICE
4347 022316 113777 001364 157532  MOV     B, LINE, JDZTCR ;ENABLE ALL SELECTED LINES
4348 022324 106427 000000      MTPS     #0           ;ALLOW INTERRUPTS
4349 022330 005037 022370      4$:  CLR     7$           ;
4350 022334 005037 022372      CLR     8$           ;
4351 022340 032777 000100 157474  5$:  BIT     #RIE, JDZCSR  ;WHEN RX DONE; RIE WILL =0
4352 022346 001407      BEQ     6$           ;BR IF ALL DONE

```

MD-11-DZDZA-B
DZDZAB.P11MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.15:03 PAGE 90
COPYRIGHT 1976

DIGITAL EQUIP. CORP.

4352	022350	005237	022370		INC	7\$	
4353	022354	001371			BNE	5\$	
4354	022356	105237	022372		INCB	8\$	
4355	022362	100366			BPL	5\$	
4356	022364	104011			ERROR	11	; *RX FAILED TO FINISH (INTERRUPT)
4357	022366	104400		6\$:	ADVANCE		; ADVANCE LOOP
4358	022370	000000		7\$:	0		
4359	022372	000000		8\$:	0		
4360							
4361							
4362							
4363							
4364	022374	017704	157446	9\$:	MOV	ADZBUF, R4	; GET THE CHARACTER
4365	022400	100401			BMI	10\$; IF IT WAS VALID, CONTINUE TESTING
4366	022402	104000			ERROR		; ERROR- ILLEGAL CHAR... DATA VALID NOT SET
4367	022404	010401		10\$:	MOV	R4, R1	; COPY THE RECEIVED INFORMATION
4368	022406	000301			SWAB	R1	; GET THE LINE NUMBER IN THE LOWER BYTE
4369	022410	042701	177770		BIC	#1C<7>, R1	; ISOLATE THE LINE NUMBER
4370	022414	006301			ASL	R1	; ALIGN IT ON A WORD BOUNDARY
4371	022416	032704	010000		BIT	#PARER, R4	; PARITY ERROR SHOULD BE SET. IS IT?
4372	022422	001013			BNE	11\$; IF SO, GO CHECK CHARACTER
4373	022424	013737	002046 001400		MOV	DZBUF, REGIST	; SET UP FOR THE ERROR MESSAGE
4374	022432	010405			MOV	R4, R5	
4375	022434	042705	000377		BIC	#377, R5	
4376	022440	156105	001442		BISB	TRO(R1), R5	; GET THE CORRECT CHARACTER
4377	022444	052705	110000		BIS	#DVALID, PARER, R5	; BUILD WHAT WAS EXPECTED
4378	022450	104006			ERROR	6	; ERROR- DID NOT GET CORRECT INFORMATION
4379	022452	126104	001442	11\$:	CMPB	TRO(R1), R4	; CHECK THE CHARACTER. IS IT CORRECT?
4380	022456	001413			BEG	12\$; IF SO, GO SET UP NEXT CHARACTER
4381	022460	116105	001442		MOVB	TRO(R1), R5	; LOAD THE CHARACTER FOR ERROR REPORTING
4382	022464	042705	177400		BIC	#1C<377>, R5	; CLEAR SIGN EXTEND
4383	022470	010137	001372		MOV	R1, SAVLIN	; GET THE LINE NUMBER FOR REPORTING
4384	022474	006237	001372		ASR	SAVLIN	; ALIGN IT CORRECTLY
4385	022500	042704	177400		BIC	#1C<377>, R4	; REMOVE THE JUNK FROM R4, THE ACTUAL CHARACTER
4386	022504	104005			ERROR	5	; DATA ERROR
4387	022506	005261	001442	12\$:	INC	TRO(R1)	; SET UP THE NEXT CHARACTER
4388	022512	005203			INC	R3	; ADD TO THE TOTAL RECEIVED COUNT
4389	022514	032777	040000 157320		BIT	#TIE, ADZCSR	; ARE TRANSMISSIONS DONE?
4390	022522	001010			BNE	13\$; IF NO, GO RECEIVE SOME MORE
4391	022524	020203			CMP	R2, R3	; ARE ALL CHARACTERS RECEIVED?
4392	022526	001006			BNE	13\$; IF NO, GO RECEIVE SOME MORE
4393	022530	042777	000100 157304		BIC	#RIE, ADZCSR	; DISABLE RECEIVER INTERRUPTS
4394	022536	012716	022366		MOV	#6\$, (SP)	; CRUNCH THE STACK
4395	022542	000002			RTI		; RETURN AND FINISH
4396	022544	012716	022330	13\$:	MOV	#4\$, (SP)	; CRUNCH THE STACK
4397	022550	000002			RTI		; GO BACK TO RECEIVER WAIT LOOP

; RECEIVER SERVICE ROUTINE

M07

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 91
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

```

4398
4399
4400
4401
4402 022552 117701 157266 XMTSRV: MOVB   @HDZCSR,R1      ;GET THE LINE NUMBER. IS THE TRANSMITTER
4403 022556 100401          BMI    1$          ;REALLY READY? IF SO, GO LOAD THE CHARACTER
4404 022560 104003          ERROR  3          ;*TRANSMITTER NOT READY- FALSE INTERRUPT
4405 022562 042701 177770 1$:  BIC    #1C<7>,R1    ;ISOLATE THE LINE NUMBER
4406 022566 006301          ASL    R1          ;MAKE SURE IT REFERENCES A WORD BOUNDARY
4407 022570 116177 001422 157270 MOVB   TDO(R1),@DZTDR ;LOAD THE CURRENT CHARACTER FOR THIS LINE
4408 022576 005261 001422      INC    TDO(R1)      ;SET UP NEXT CHARACTER FOR THIS LINE
4409 022602 005202          INC    R2          ;UP THE NUMBER OF TRANSMISSIONS
4410 022604 023761 001376 001422 CMP    XMTCNT,TDO(R1) ;HAVE WE DONE ALL PATTERNS ON THIS LINE?
4411 022612 001015          BNE    4$          ;IF NOT, KEEP ON TRANSMITTING
4412 022614 012700 000001      MOV    #1,R0       ;SET UP A DESELECTION POINTER
4413 022620 006201          ASR    R1          ;GET THE LINE NUMBER AGAIN
4414 022622 005301          2$:  DEC    R1          ;REDUCE THE COUNT. WAS THIS THE LINE?
4415 022624 100402          BMI    3$          ;IF SO, GO DISABLE THE ENABLE BIT FOR IT
4416 022626 006300          ASL    R0          ;MOVE THE POINTER TO THE NEXT LINE
4417 022630 000774          BR     2$          ;GO CHECK THE NEXT LINE
4418 022632 140077 157220 3$:  BICB   R0,@DZTCR    ;DISABLE THE LINE POINTED TO BY R0
4419 022636 001003          BNE    4$          ;IF MORE LINES ARE ACTIVE, GO CONTINUE TRANSMIT
4420 022640 042777 040000 157174 BIC    #TIE,@DZCSR  ;IF NOT, DISABLE TRANSMITTER INTERRUPTS
4421 022646 000002          4$:  RTI          ;RETURN TO THE TIMING LOOP
4422
4423
4424
4425
4426 022650 012737 000004 001222 BUILD: MOV    #4,STMP2    ;ROTATE 4 BITS BACK INTO STMP1
4427 022656 006037 001224 1$:  ROR    STMP3      ;GET THE BITS FROM STMP3, THE HIGH BYTE
4428 022662 006037 001220      ROR    STMP1      ;OF THE RELATIVE TIME COUNTER. PUT THEM BACK
4429 022666 005337 001222      DEC    STMP2      ;INTO STMP1 USING THE CARRY BIT WITH
4430          ;ROTATE INSTRUCTIONS
4431 022672 001371          BNE    1$          ;REDUCE COUNT. ALL BITS BACK? IF NOT, GET MORE
4432 022674 000207          RTS    PC          ;RETURN TO CALLING TEST
4433

```

; RELATIVE TIME BUILDING ROUTINE

```

;RECEIVER SERVICE ROUTINE
4434
4435
4436 022676 105777 157140      RXISR1: TSTB   JDZCSR      ; IS THE RECEIVER REALLY READY?
4437 022702 100401             BMI     1$          ; IF SO, GO SERVICE IT
4438 022704 104004             ERROR   4           ; *ERROR- RECEIVER DONE FLAG ISN'T SET
4439 022706 017704 157134      1$:   MOV     JDZRBUF,R4  ; SAVE THE RECEIVER INFORMATION
4440 022712 100401             BMI     2$          ; IF IT WAS VALID, GO PROCESS IT
4441 022714 104000             ERROR   2           ; ERROR- DATA VALID WASN'T SET
4442 022716 032704 070000      2$:   BIT     #OVRUN!FRMERR!PARER,R4 ; ARE ANY ERROR FLAGS SET?
4443 022722 001404             BEQ     3$          ; IF NOT, GO CONTINUE PROCESSING
4444 022724 013737 002046 001400  MOV     DZRBUF,REGIST ; SET UP FOR ERROR REPORTING
4445 022732 104002             ERROR   2           ; ERROR- RECEIVER ERROR FLAG SET
4446 022734 010401             3$:   MOV     R4,R1      ; COPY THE RECEIVER INFORMATION
4447 022736 000301             SWAB   R1          ; GET THE LINE NUMBER IN THE LOWER BYTE
4448 022740 042701 177770      BIC     #1C<7>,R1   ; ISOLATE THE LINE NUMBER
4449 022744 006301             ASL    R1          ; ALIGN IT ON A WORD BOUNDARY
4450 022746 120461 001442      CMPB   R4,TR0(R1)  ; IS THE CHARACTER WHAT IT SHOULD BE?
4451 022752 001413             BEQ     4$          ; IF SO, GO CONTINUE PROCESSING
4452 022754 116105 001442      MOVB   TR0(R1),R5  ; GET WHAT WAS EXPECTED FOR ERROR REPORTING
4453 022760 042705 177400      BIC     #1C<377>,R5 ; ELIMINATE PROPAGATED SIGN
4454 022764 042704 177400      BIC     #1C<377>,R4 ; ISOLATE THE ACTUAL CHARACTER
4455 022770 010137 001372      MOV     R1,SAVLIN  ; GET THE LINE NUMBER OF THE RECEIVER ERROR
4456 022774 006237 001372      ASR    SAVLIN     ; ALIGN IT CORRECTLY FOR REPORTING
4457 023000 104005             ERROR   5           ; *DATA ERROR
4458 023002 005261 001442      4$:   INC     TR0(R1)   ; SET UP THE NEXT EXPECTED CHARACTER
4459 023006 005203             INC    R3          ; INCREMENT THE COUNT OF RECEIVED CHARACTERS
4460 023010 032761 000020 001442  BIT     #20,TR0(R1) ; HAVE ALL CHARACTERS BEEN RECEIVED?
4461 023016 001402             BEQ     5$          ; IF NOT, GO RECEIVE SOME MORE
4462 023020 020203             CMP    R2,R3      ; HAVE WE RECEIVED ALL CHARACTERS?
4463 023022 001401             BEQ     6$          ; IF SO, GO DETERMINE THE TIMING
4464 023024 000002             5$:   RTI                    ; GO CONTINUE TIMING AND ALLOW INTERRUPTS
4465 023026 004737 022650      6$:   JSR     PC,BUILD   ; GET THE RELATIVE TIME (SIGNIFICANT BITS)
4466
4467 023032 013700 023112      MOV     OFFSET,R0  ; GET POPINTER
4468 023036 013760 001220 002102  MOV     $TMP1,TMTBL(R0) ; SAVE THIS TEST'S TIME
4469 023044 005737 023112      TST    OFFSET     ; FIRST TEST?
4470 023050 001414             BEQ     7$          ; IF NOT, GO CHECK THE TIME
4471 023052 005740             TST    -(R0)       ; POINT TO THE PREVIOUS TIME TAKEN
4472 023054 026037 002102 001220  CMP     TMTBL(R0),$TMP1 ; IS THIS TIME WHAT IT SHOULD BE?
4473 023062 101007             BHI    7$          ; IF SO, GO TO THE NEXT TEST
4474 023064 016005 002102      MOV     TMTBL(R0),R5 ; PLACE WHAT WAS EXPECTED IN R5
4475 023070 010137 001372      MOV     R1,SAVLIN  ; GET THE LINE NUMBER OF THE RECEIVER
4476 023074 006237 001372      ASR    SAVLIN     ; MAKE SURE IT'S THE LINE NUMBER
4477 023100 104021             ERROR   21         ; TIMING ERROR
4478 023102 042777 000140 156732 7$:   BIC     #RIE!MSENAB,JDZCSR ; DISABLE THE DEVICE
4479 023110 000002             RTI                    ; RETURN TO THE PROGRAM
4480 023112 000000      OFFSET: 0

```

:DZ11 ECHO/CABLE TEST
:COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536

:#STARTING PROCEDURE
:#LOAD PROGRAM
:#LOAD ADDRESS 000210
:#PRESS START
:#PROGRAM WILL TYPE DZ11 ECHO/CABLE TEST
:#PROGRAM WILL TYPE WHICH TEST- ECHO OR CABLE
:#TYPE IN E OR C RESPECTIVELY
:#PROGRAM WILL TYPE "VECTOR ADDRESS-"
:#TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR
:#FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
:#PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-"
:#TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER
:#FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
:#PROGRAM WILL TYPE "LINE NUMBER-"
:#TYPE IN THE LINE NUMBER TO BE TESTED (IN OCTAL)
:# FOLLOWED BY <CARRIAGE RETURN>
:#PROGRAM WILL TYPE "BAUD RATE-"
:#TYPE IN THE BAUD RATE OF THE DZ11 TERMINAL
:#, FOLLOWED BY <CARRIAGE RETURN>
;#THE FOLLOWING BAUD RATES ARE ACCEPTED IN DECIMAL

- * 50
- * 75
- * 110
- * 135 (ROUNDED OFF 134.5)
- * 150
- * 300
- * 600
- * 1200
- * 1800
- * 2000
- * 2400
- * 3600
- * 4800
- * 7200
- * 9600

:#ALL OTHERS ARE REJECTED

:#PROGRAM WILL TYPE "ECHO" OR "CABLE TEST" TO INDICATE THAT TESTING HAS STARTE

:PROGRAM INITIALIZATION
:LOCK OUT INTERRUPTS
:SET UP PROCESSOR STACK
:SET UP POWER FAIL VECTOR
:CLEAR PROGRAM FLAGS AND COUNTS

023114 012706 001120
023120 106427 000340
023124 012737 023114
023132 005037 025274
023136 005037 001242
023142 005037 001132
023146 105037 001123

001126

XSTART: MOV #STACK, SP ;SET UP PROCESSOR STACK
MTPS #R7 ;LOCK OUT INTERRUPTS
MOV #XSTART, SLPADR ;SET UP IN CASE OF POWER FAIL
CLR STFLG ;CLEAR TEST START FLAG
CLR SPASS ;CLEAR PASS COUNT
CLR SERTTL ;CLEAR ERROR COUNT
CLRB SERFLG ;CLEAR ERROR FLAG

4537	023152	005037	025300			CLR	LAST		:CLEAR LAST ERROR PC
4538	023156	032777	000001	155774	VEC1:	BIT	#SW00,JSWR		:IF SW00=1, GET NEW VECTOR
4539	023164	001465				BEG	OTHER		:AND CSR
4540	023166	012701	000300		VEC2:	MOV	#300,R1		
4541	023172	012702	000302			MOV	#302,R2		
4542	023176	010221			IS:	MOV	R2,(R1)+		:RESTORE TRAPCATCHER
4543	023200	005022				CLR	(R2)+		:IN FLOATING VECTOR AREA
4544	023202	022122				CMP	(R1)+,(R2)+		:UPDATE THE POINTERS
4545	023204	020127	001000			CMP	R1,#1000		
4546	023210	001372				BNE	IS		
4547	023212	104403				INSTR			:INPUT ADDRESS OF DEVICE VECTOR
4548	023214	025326				MVECTOR			:MESSAGE "VECTOR ADDRESS--"
4549	023216	104405				PARAM			:CONVERT STRING TO OCTAL
4550	023220	000300				300			:LOW LIMIT
4551	023222	000770				770			:HIGH LIMIT
4552	023224	002072				DZRIV			:LOCATIONS TO BE FILLED
4553	023226	003			.BYTE	3			:LSB MASK
4554	023227	004			.BYTE	4			:NUMBER OF LOCATIONS
4555	023230	104403				INSTR			:INPUT ADDRESS OF DEVICE CSR
4556	023232	025350				MREGAD			:MESSAGE "CONTROL REGISTER ADDRESS--"
4557	023234	104405				PARAM			:CONVERT STRING TO OCTAL
4558	023236	160000				160000			:LOW LIMIT
4559	023240	163700				163700			:HIGH LIMIT
4560	023242	002042				DZCSR			:LOCATIONS TO BE FILLED
4561	023244	007			.BYTE	7			:LSB MASK
4562	023245	001			.BYTE	1			:NUMBER OF LOCATIONS
4563	023246	013737	002042	002046		MOV	DZCSR,DZRBUF		:BEGIN BUILDING DEVICE ADDRESSES
4564	023254	062737	000002	002046		ADD	#2,DZRBUF		:FORM THE READ BUFFER ADDRESS
4565	023262	013737	002046	002052		MOV	DZRBUF,DZLPR		:REMEMBER THAT THIS IS ALSO LINE PARAMETER REG.
4566	023270	013737	002046	002056		MOV	DZRBUF,DZTCR		:BEGIN BUILDING TRANSMITTER CONTROL REGISTER
4567	023276	062737	000002	002056		ADD	#2,DZTCR		:FORM THE TRANSMITTER CONTROL REGISTER POINTER
4568	023304	013737	002056	002060		MOV	DZTCR,HDZTCR		
4569	023312	005237	002060			INC	HDZTCR		
4570	023316	013737	002056	002066		MOV	DZTCR,DZTDR		:BEGIN FORMING TRANSMITTER DATA REGISTER
4571	023324	062737	000002	002066		ADD	#2,DZTDR		:FORM THE TRANSMITTER DATA REGISTER
4572	023332	013737	002066	002062		MOV	DZTDR,DZMSR		
4573	023340	032777	000002	155612	OTHER:	BIT	#SW01,JSWR		:RESELECT OF TEST?
4574	023346	001427				BEG	XBEGIN		:IF NOT, SKIP ASKING WHICH ONE
4575	023350	104403				INSTR			:INPUT WHICH TEST YOU ARE RUNNING
4576	023352	025534				MWHICH			:ECHO OR CABLE
4577	023354	104416				PARAM			:SET FLAG
4578	023356	025272				WCHFLG			:THIS FLAG
4579	023360	104403			BAUD:	INSTR			:INPUT BAUD RATE
4580	023362	025456				MSPEED			:MESSAGE "BAUD RATE--"
4581	023364	104415				PARAM			:CONVERT DECIMAL STRING TO OCTAL
4582	023366	000062				50.			:LOW LIMIT
4583	023370	022600				9600.			:HIGH LIMIT
4584	023372	025310				LINESP			:LOCATION TO BE FILLED
4585	023374	000			.BYTE	0			:LSB MASK
4586	023375	001			.BYTE	1			:NUMBER OF LOCATIONS
4587	023376	104413			LINEX:	DEVICE.CLR			:CLEAR DEVICE
4588	023400	005037	025274			CLR	STFLG		:CLEAR PROGRAM START FLAG
4589	023404	104403				INSTR			:INPUT LINE NUMBER
4590	023406	025446				MLINE			:MESSAGE "LINE NUMBER--"
4591	023410	104405				PARAM			:CONVERT STRING TO OCTAL
4592	023412	000000				0			:LOW LIMIT

F08

NO-11-DZDZA-B MACY11 27(732) 03-NOV-76 15:03 PAGE 97
 DZDZAB.P11 DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

4705	024172	012737	024200	001362		MOV	#18,LOCK	:LOOP
4706	024200	113777	025316	155652	1S:	MOVB	NUMTCR,JDZTCR	:SET DTR
4707	024206	005005				CLR	R5	
4708	024210	153705	025316			BISB	NUMTCR,R5	:BUILD EXPECTED
4709	024214	000305				SWAB	R5	:PUT IN HIGH BYTE
4710	024216	153705	025316			BISB	NUMTCR,R5	
4711	024222	104414				DELAY		:WAIT FOR CABLE DELAY
4712	024224	017704	155632			MOV	JDZMSR,R4	:READY MODEM BITS
4713	024230	020504				CMP	R5,R4	:ARE THEY OK?
4714	024232	001401				BEG	25	:BR IF YES
4715	024234	104022				ERROR	22	:IS THE TEST CONNECTOR ON?
4716								:HAS RIGHT LINE BEEN SELECTED?
4717								:IF SO- YOU HAVE A PROBLEM!
4718								:MODEM BITS NOT RIGHT
4719	024236	104401			2S:	SCOPI		:LOOP
4720	024240	104413			3S:	DEVICE.CLR		:INIT DZ11
4721	024242	013737	025312	001366		MOV	SPEED,PAR	:SET LINE SPEED
4722	024250	053737	025314	001366		BIS	NUMLIN,PAR	:SELECT LINE # & REC. INTERRUPT ENABLE
4723	024256	052737	010000	001366		BIS	#RCVON,PAR	:ENABLE THE RECEIVER FOR THIS LINE
4724	024264	052777	040140	155550		BIS	#TIE!R!E!MSENAB,JDZCSR	:SET TRANSMITTER INTERRUPT ENABLE
4725	024272	012777	024406	155572		MOV	#INTREC,JDZRIV	:SET UP INTR SERVICE
4726	024300	013777	025320	155566		MOV	PRI0,JDZRI5	:SET UP LEVEL
4727	024306	012777	024566	155562		MOV	#INTRAN,JDZTIV	:SET UP INTR SERVICE
4728	024314	013777	025320	155556		MOV	PRI0,JDZTI5	:SET UP LEVEL
4729	024322	005001				CLR	R1	:RX DATA POINTER- SET TO 0
4730	024324	005002				CLR	R2	:TX DATA POINTER- SET TO 0
4731	024326	013777	001366	155516		MOV	PAR,JDZLPR	:SET THE PARAMETERS AND TURN ON RECEIVER
4732	024334	106437	026172			MTPS	JD#LESS1	:ALLOW INTERRUPTS
4733	024340	013777	025316	155510		MOV	NUMTCR,JDZTCR	:SET UP TCR BIT
4734								
4735								:YOU RETURN HERE AFTER EVERY RECEIVER INTERRUPT
4736	024346	105777	154612		SPIN:	TSTB	JDSTKS	:IF SOMEBODY HITS A KEY- GET A NEW LINE #
4737	024352	100006				BPL	15	:BR IF NO KEY HIT
4738	024354	005777	154606			TST	JDSTKB	:CLEAR CHAR
4739	024360	004737	007212			JSR	PC,SERV.G	:MAKE SURE IT WASN'T <1G>
4740	024364	000137	023376			JMP	LINEX	:SWD2=1
4741	024370	005237	025276		1S:	INC	LOCKUP	:INC TIMEOUT FLAG
4742	024374	001364				BNE	SPIN	:IF NOT 0 RETURN SPINNING
4743	024376	104011				ERROR	11	:*RECEIVER FAILED TO INTERRUPT CHECK CABLE/TERMINATOR
4744	024400	000004			QUITS:	SCOPE		
4745	024402	000137	024606			JMP	XEOP	:CALL FOR END OF PASS
4746	024406	005037	025276		INTREC:	CLR	LOCKUP	:CLEAR TIMEOUT FLAG
4747	024412	105777	155424			TSTB	JDZCSR	:TEST REC DONE
4748	024416	100401				BMI	.+4	:YES
4749	024420	104004				ERROR	4	:*FALSE INTERRUPT
4750	024422	017737	155420	025322		MOV	JDZRBUFF,RECDAT	:SAVE WORD
4751	024430	100401				BMI	.+4	
4752	024432	104023				ERROR	23	:*NON VALID CHARACTER
4753	024434	032737	040000	025322		BIT	#BIT14,RECDAT	:DATA OVERRUN ?
4754	024442	001401				BEG	.+4	:NO
4755	024444	104024				ERROR	24	:*YES
4756	024446	032737	020000	025322		BIT	#BIT13,RECDAT	:FRAMING ERROR ?
4757	024454	001401				BEG	.+4	:NO
4758	024456	104025				ERROR	25	:*YES
4759	024460	032737	010000	025322		BIT	#BIT12,RECDAT	:PARITY ERROR ?
4760	024466	001401				BEG	.+4	:NO

F08

NO-11-DZDZA-B MACY11 27(732) 03-NOV-76 15:03 PAGE 97
 DZDZAB.P11 DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

```

4705 024172 012737 024200 001362      MOV      #15, LOCK      ; LOOP
4706 024200 113777 025316 155652 15:   MOVB    NUMTCR, @HDZTCR ; SET DTR
4707 024206 005005                CLR      R5              ;
4708 024210 153705 025316                BISB    NUMTCR, R5       ; BUILD EXPECTED
4709 024214 000305                SWAB    R5                ; PUT IN HIGH BYTE
4710 024216 153705 025316                BISB    NUMTCR, R5       ;
4711 024222 104414                DELAY                     ; WAIT FOR CABLE DELAY
4712 024224 017704 155632                MOV     @DZMSR, R4       ; READY MODEM BITS
4713 024230 020504                CMP     R5, R4           ; ARE THEY OK?
4714 024232 001401                BEQ     25                ; BR IF YES
4715 024234 104022                ERROR   22                ; IS THE TEST CONNECTOR ON?
4716                                     ; HAS RIGHT LINE BEEN SELECTED?
4717                                     ; IF SO- YOU HAVE A PROBLEM!
4718                                     ; MODEM BITS NOT RIGHT
4719 024236 104401                25:    SCOP1              ; LOOP
4720 024240 104413                35:    DEVICE. CLR         ; INIT DZ11
4721 024242 013737 025312 001366                MOV     SPEED, PAR       ; SET LINE SPEED
4722 024250 053737 025314 001366                BIS     NUMLIN, PAR      ; SELECT LINE # & REC. INTERRUPT ENABLE
4723 024256 052737 010000 001366                BIS     #RCVON, PAR      ; ENABLE THE RECEIVER FOR THIS LINE
4724 024264 052777 040140 155550                BIS     #TIE!R!E!MSENAB, @DZCSR ; SET TRANSMITTER INTERRUPT ENABLE
4725 024272 012777 024406 155572                MOV     #INTREC, @DZRIV ; SET UP INTR SERVICE
4726 024300 013777 025320 155566                MOV     PRIO, @DZRI5     ; SET UP LEVEL
4727 024306 012777 024566 155562                MOV     #INTRAN, @DZTIV ; SET UP INTR SERVICE
4728 024314 013777 025320 155556                MOV     PRIO, @DZTI5     ; SET UP LEVEL
4729 024322 005001                CLR     R1                ; RX DATA POINTER- SET TO 0
4730 024324 005002                CLR     R2                ; TX DATA POINTER- SET TO 0
4731 024326 013777 001366 155516                MOV     PAR, @DZLPR      ; SET THE PARAMETERS AND TURN ON RECEIVER
4732 024334 106437 026172                MTPS   @#LESS1           ; ALLOW INTERRUPTS
4733 024340 013777 025316 155510                MOV     NUMTCR, @DZTCR  ; SET UP TCR BIT
4734
4735                                     ; YOU RETURN HERE AFTER EVERY RECEIVER INTERRUPT
4736 024346 105777 154612                SPIN:   TSTB   @STKS       ; IF SOMEBODY HITS A KEY- GET A NEW LINE #
4737 024352 100006                BPL     15                ; BR IF NO KEY HIT
4738 024354 005777 154606                TST     @STKB            ; CLEAR CHAR
4739 024360 004737 007212                JSR     PC, SERV.G       ; MAKE SURE IT WASN'T <TG>
4740 024364 000137 023376                JMP     LINEX            ; SWD=1
4741 024370 005237 025276                15:    INC     LOCKUP      ; INC TIMEOUT FLAG
4742 024374 001364                BNE     SPIN             ; IF NOT 0 RETURN SPINNING
4743 024376 104011                ERROR   11                ; *RECEIVER FAILED TO INTERRUPT CHECK CABLE/TERMINATOR
4744 024400 000004                QUITs: SCOPE
4745 024402 000137 024606                JMP     XEOP              ; CALL FOR END OF PASS
4746 024406 005037 025276                INTREC: CLR     LOCKUP    ; CLEAR TIMEOUT FLAG
4747 024412 105777 155424                TSTB   @DZCSR           ; TEST REC DONE
4748 024416 100401                BMI     .+4              ; YES
4749 024420 104004                ERROR   4                ; *FALSE INTERRUPT
4750 024422 017737 155420 025322                MOV     @DZRBUF, RECDAT ; SAVE WORD
4751 024430 100401                BMI     .+4              ;
4752 024432 104023                ERROR   23                ;
4753 024434 032737 040000 025322                BIT     #BIT14, RECDAT  ; *NON VALID CHARACTER
4754 024442 001401                BEQ     .+4              ; DATA OVERRUN ?
4755 024444 104024                ERROR   24                ; NO
4756 024446 032737 020000 025322                BIT     #BIT13, RECDAT  ; *YES
4757 024454 001401                BEQ     .+4              ; FRAMING ERROR ?
4758 024456 104025                ERROR   25                ; NO
4759 024460 032737 010000 025322                BIT     #BIT12, RECDAT  ; *YES
4760 024466 001401                BEQ     .+4              ; PARITY ERROR ?
4760                                     ; NO
    
```

H08

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 99
DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

4788				
4789				:END OF PASS
4790				:RESTART TEST
4791				
4792	024606	104402		XEOP: TYPE ;TYPE NAME OF TEST
4793	024610	025404		MPASS
4794	024612	005037	025300	CLR LAST ;CLEAR LAST ERROR PC
4795	024616	105037	001123	CLRB SERFLG ;CLEAR ERROR FLAG
4796	024622	000137	023426	RSTRT: JMP XBEGIN
4797				
4798				
4799	024626	011605		.PARMD: ;CONVERT DECIMAL ASCII STRING TO OCTAL
4800	024630	012537	025012	MOV (SP),R5
4801	024634	012537	025014	MOV (R5)+,6\$
4802	024640	012537	025016	MOV (R5)+,7\$
4803	024644	112537	025020	MOV (R5)+,8\$
4804	024650	112537	025021	MOVB (R5)+,9\$
4805	024654	010516		MOVB (R5)+,10\$
4806	024656	005005		MOV R5,(SP)
4807	024660	012704	010452	2\$: CLR R5
4808	024664	122714	000015	MOV #INBUF,R4
4809	024670	001424		CMPB #15,(R4)
4810	024672	121427	000060	1\$: BEQ 3\$
4811	024676	002421		CMPB (R4),#'0
4812	024700	121427	000071	BLT 3\$
4813	024704	003016		CMPB (R4),#'9
4814	024706	142714	000060	BGT 3\$
4815	024712	005002		BICB #'0,(R4)
4816	024714	152402		CLR R2
4817	024716	060205		BISB (R4)+,R2
4818	024720	122714	000015	ADD R2,R5
4819	024724	001410		CMPB #15,(R4)
4820	024726	006305		BEQ 4\$
4821	024730	010502		ASL R5 ;X2
4822	024732	006305		MOV R5,R2 ;SAVE X2
4823	024734	006305		ASL R5 ;X4
4824	024736	060205		ASL R5 ;X8
4825	024740	000754		ADD R2,R5 ;TIMES 10
4826	024742	104404		BR 1\$
4827	024744	000744		3\$: INSTER
4828				BR 2\$
4829				
4830				:TEST TO SEE IF NUMBER IS WITHIN LIMITS
4831	024746	020537	025014	4\$: CMP R5,7\$
4832	024752	101373		BHI 3\$
4833	024754	020537	025012	CMP R5,6\$
4834	024760	103770		BLO 3\$
4835	024762	133705	025020	BITB 9\$,R5
4836	024766	001365		BNE 3\$
4837				
4838				:STORE NUMBER AT SPECIFIED ADDRESS
4839				
4840	024770	013704	025016	5\$: MOV 8\$,R4
4841	024774	010524		MOV R5,(R4)+
4842	024776	062705	000002	ADD #2,R5
4843	025002	105337	025021	DECB 10\$

4844 025006 001372
4845 025010 000002
4846 025012 000000
4847 025014 000000
4848 025016 000000
4849 025020 000
4850 025021 000

BNE 55
RTI
6S: 0
7S: 0
8S: 0
9S: .BYTE 0
10S: .BYTE 0

4851
4852
4853
4854
4855
4856
4857

;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
;BUFFER TO THE CHARACTERS "E" AND "C"
;IF THE CHARACTER IS "E" CLEAR THE FLAG
;IF THE CHARACTER IS "C" SET THE FLAG

4858 025022 017605 000000
4859 025026 142737 000040 010452
4860 025034 122737 000105 010452
4861 025042 001002
4862 025044 105015
4863 025046 000406
4864 025050 122737 000103 010452
4865 025056 001005
4866 025060 112715 177777
4867 025064 062716 000002
4868 025070 000002
4869 025072 104404
4870 025074 000752

.PAWCH:MOV 2(SP),R5
BICB #40,INBUF ;SET FOR LOWER CASE INPUT
CMPB #'E,INBUF ;IS IT "E" ?
BNE 1\$
CLRB (R5) ;000
BR 2\$
1\$: CMPB #'C,INBUF ;IS IT "C" ?
BNE 3\$
MOVB #-1,(R5) ;3177
2\$: ADD #2,(SP)
RTI
3\$: INSTER ;RETRY
BR .PAWCH

4871
4872
4873
4874
4875
4876
4877

;THIS ROUTINE CONVERTS LINE SPEED (LINESP) AND
;LINE NUMBER (SAVLIN) FOR DZLPR, DZTCR AND DZCSR
;REGISTER USAGE.

4878 025076 013737 001372 025314
4879 025104 013700 001372
4880 025110 005037 025316
4881 025114 012702 000001
4882 025120 005300
4883 025122 100402
4884 025124 006302
4885 025126 000774
4886 025130 012701 025172
4887 025134 010237 025316
4888 025140 022137 025310
4889 025144 001407
4890 025146 005721
4891 025150 001373
4892 025152 104402 025420
4893 025156 012705 023360
4894 025162 000402
4895 025164 011137 025312
4896 025170 000205

SET: MOV SAVLIN,NUMLIN ;SAVE SAVLIN
XTCRO: MOV SAVLIN,R0 ;COPY THE LINE NUMBER FOR LOOP CONTROL
CLR NUMTCR ;SET A DEFAULT OF LINE 0 OR NO LINES
MOV #1,R2 ;SET A BIT POINTER TO THE FIRST LINE
XTCR1: DEC R0 ;REDUCE THE INDICATOR, IS IT MINUS YET?
BMI SET1 ;IF SO, R2 POINTS TO THE RIGHT LINE
ASL R2 ;IF NOT, MOVE THE POINTER TO THE NEXT LINE
BR XTCR1 ;GO SEE IF THIS LINE IS THE ONE
SET1: MOV #TABLE2,R1
MOV R2,NUMTCR ;COPY THE CORRECT BIT POINTER
1\$: CMP (R1)+,LINESP
BEQ 2\$
TST (R1)+ ;IS IT THE END OF TABLE?
BNE 1\$;NO
TYPE MINVAL ;INVALID BAUD RATE, BEGIN AGAIN
MOV #BAUD,R5 ;JUMP TO BAUD THRU R5
BR 3\$
2\$: MOV (R1),SPEED ;SET UP BAUD RATE
3\$: RTS R5

4897
4898
4899

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 100
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

4844 025006 001372
4845 025010 000002
4846 025012 000000
4847 025014 000000
4848 025016 000000
4849 025020 000
4850 025021 000

BNE 55
RTI
6S: 0
7S: 0
8S: 0
9S: .BYTE 0
10S: .BYTE 0

4851
4852
4853
4854
4855
4856
4857

;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
;BUFFER TO THE CHARACTERS "E" AND "C"
;IF THE CHARACTER IS "E" CLEAR THE FLAG
;IF THE CHARACTER IS "C" SET THE FLAG

4858 025022 017605 000000
4859 025026 142737 000040 010452
4860 025034 122737 000105 010452
4861 025042 001002
4862 025044 105015
4863 025046 000406
4864 025050 122737 000103 010452 1S:
4865 025056 001005
4866 025060 112715 177777
4867 025064 062716 000002 2S:
4868 025070 000002
4869 025072 104404 3S:
4870 025074 000752
4871
4872
4873
4874
4875
4876
4877

.PAWCH:MOV 2(SP),R5
BICB #40,INBUF ;SET FOR LOWER CASE INPUT
CMPB #'E,INBUF ;IS IT "E" ?
BNE 1S
CLRB (R5) ;000
BR 2S
CMPB #'C,INBUF ;IS IT "C" ?
BNE 3S
MOVB #-1,(R5) ;3177
ADD #2,(SP)
RTI
INSTER ;RETRY
BR .PAWCH

4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899

;THIS ROUTINE CONVERTS LINE SPEED (LINESP) AND
;LINE NUMBER (SAVLIN) FOR DZLPR, DZTCR AND DZCSR
;REGISTER USAGE.

4878 025076 013737 001372 025314
4879 025104 013700 001372
4880 025110 005037 025316
4881 025114 012702 000001
4882 025120 005300
4883 025122 100402
4884 025124 006302
4885 025126 000774
4886 025130 012701 025172
4887 025134 010237 025316
4888 025140 022137 025310
4889 025144 001407
4890 025146 005721
4891 025150 001373
4892 025152 104402 025420
4893 025156 012705 023360
4894 025162 000402
4895 025164 011137 025312
4896 025170 000205

SET: MOV SAVLIN,NUMLIN ;SAVE SAVLIN
XTCRO: MOV SAVLIN,R0 ;COPY THE LINE NUMBER FOR LOOP CONTROL
CLR NUMTCR ;SET A DEFAULT OF LINE 0 OR NO LINES
MOV #1,R2 ;SET A BIT POINTER TO THE FIRST LINE
XTCR1: DEC R0 ;REDUCE THE INDICATOR. IS IT MINUS YET?
BMI SET1 ;IF SO, R2 POINTS TO THE RIGHT LINE
ASL R2 ;IF NOT, MOVE THE POINTER TO THE NEXT LINE
BR XTCR1 ;GO SEE IF THIS LINE IS THE ONE
SET1: MOV #TABLE2,R1
MOV R2,NUMTCR ;COPY THE CORRECT BIT POINTER
1S: CMP (R1)+,LINESP
BEQ 2S
TST (R1)+ ;IS IT THE END OF TABLE?
BNE 1S ;NO
TYPE MINVAL ;INVALID BAUD RATE, BEGIN AGAIN
MOV #BAUD,R5 ;JUMP TO BAUD THRU R5
BR 3S
2S: MOV (R1),SPEED ;SET UP BAUD RATE
3S: RTS R5

JOB

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 101
DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

```

4900
4901 025172 000062
4902 025174 010070
4903 025176 000113
4904 025200 010470
4905 025202 000156
4906 025204 011070
4907 025206 000207
4908 025210 011470
4909 025212 000226
4910 025214 012070
4911 025216 000454
4912 025220 012470
4913 025222 001130
4914 025224 013070
4915 025226 002260
4916 025230 013470
4917 025232 003410
4918 025234 014070
4919 025236 003720
4920 025240 014470
4921 025242 004540
4922 025244 015070
4923 025246 007020
4924 025250 015470
4925 025252 011300
4926 025254 016070
4927 025256 016040
4928 025260 016470
4929 025262 022600
4930 025264 017070
4931 025266 177777 000000
4932
4933
4934 025272 000000
4935 025274 000000
4936 025276 000000
4937 025300 000000
4938 025302 000000
4939 025304 000000
4940 025306 000000
4941 025310 000156
4942 025312 006307
4943
4944 025314 000100
4945
4946 025316 000001
4947 025320 000240
4948 025322 000000
4949 025324 000000
4950 025326 053200 041505 047524
      025350 041600 047117 051124
      025404 050200 051501 020123
      025420 044600 053116 046101
      025446 046200 047111 035105
      025456 041200 052501 020104
  
```

```

TABLE2: ; THE FOLLOWING IS A TABLE OF LEGAL BAUD RATES (8 BITS/CHAR)
          .WORD 50. ; 50 BAUD
          .WORD 10070 ;
          .WORD 75. ; 75 BAUD
          .WORD 10470 ;
          .WORD 110. ; 110 BAUD
          .WORD 11070 ; TWO STOP BITS
          .WORD 135. ; 134.5 BAUD
          .WORD 11470 ; TWO STOP BITS
          .WORD 150. ; 150 BAUD
          .WORD 12070 ; TWO STOP BITS
          .WORD 300. ; 300 BAUD
          .WORD 12470 ; ONE STOP BIT
          .WORD 600. ; 600 BAUD
          .WORD 13070 ; ONE STOP BIT
          .WORD 1200. ; 1200 BAUD
          .WORD 13470 ; ONE STOP BIT
          .WORD 1800. ; 1800 BAUD
          .WORD 14070 ; ONE STOP BIT
          .WORD 2000. ; 2000 BAUD
          .WORD 14470 ; ONE STOP BIT
          .WORD 2400. ; 2400 BAUD
          .WORD 15070 ; ONE STOP BIT
          .WORD 3600. ; 3600 BAUD
          .WORD 15470 ; ONE STOP BIT
          .WORD 4800. ; 4800 BAUD
          .WORD 16070 ; ONE STOP BIT
          .WORD 7200. ; 7200 BAUD
          .WORD 16470 ; ONE STOP BIT
          .WORD 9600. ; 9600 BAUD
          .WORD 17070 ;
          .WORD -1,0 ; TABLE TERMINATOR
  
```

```

WCHFLG: 0 ; ECHO OR CABLE FLAG
STFLG: 0 ; PROGRAM START FLAG
LOCKUP: 0 ; TIMEOUT FLAG
LAST: 0 ; LAST ERROR PC
TDATA: 0
RDATA: 0
BYTCNT: 0 1
LINE: 110 ; DEFAULT BAUD RATE
SPEED: 6307 ; DEFAULT 110 BAUD, 8 BITS/CHAR,
; FD, 2 STOP BITS
; DEFAULT VALUE, REC. INTERRUPT ENABLED
NUMLIN: 100
NUMTCR: 1 ; DEFAULT VALUE, TCR BIT 0
PRIO: 240 ; DEFAULT DEVICE PRIORITY 5
RECDAT: 0
TBUF: 0
MVECTO: .ASCIZ <200>/VECTOR ADDRESS- /
MREGAD: .ASCIZ <200>/CONTROL REGISTER ADDRESS- /
MPASS: .ASCIZ <200>/PASS DONE./
MINVAL: .ASCIZ <200>/INVALID BAUD RATE - /
MLINE: .ASCIZ <200>/LINE: /
MSPEED: .ASCIZ <200>/BAUD RATE - /
  
```

K08

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 102
DZ11 DEVICE DIAGNOSTICS. COPYRIGHT 1976 DIGITAL EQUIP. CORP.

```

025474 052200 050131 020105 MCHAR: .ASCIZ <200>/TYPE A CHAR. ON DZ11 TERMINAL /
025534 053600 044510 044103 MWHICH: .ASCIZ <200>/WHICH TEST ? ECHO OR CABLE (E OR C) /
025602 052200 051105 044515 MTERM: .ASCIZ <200>/TERMINAL ECHO TEST /
025627 200 040503 046102 MCABLE: .ASCIZ <200>/CABLE TEST /
025644 006777 177777 177412 MQUICK: .ASCII <377><15><377><377><12><377><377>
025653 124 042510 050440 .ASCII /THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789/
025750 006777 177777 177412 .ASCII <377><15><377><377><12><377><377><377><0>
025762
    
```

```

.EVEN
;*****
;UTILITIES
;*****
    
```

4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996

```

;THIS UTILITY CALCULATES PRIORITY LEVEL, SETS UP CSR'S, SETS UP VECTORS.
DZLEV: ASL DZPRT ;BUILD PRIORITY IN THIS LOCATION
        ASL DZPRT ;USING ARITHMETIC SHIFTS, ROTATE
        ASL DZPRT ; THE PRIORITY LEVEL PAST
        ASL DZPRT ; THE BIT POSITIONS CORRE-
        ASL DZPRT ;SPONDING TO THE CONDITION CODES
        MOV DZPRT, LESS1 ;MOVE THIS TO LESS1
        SUB #1, LESS1 ;CREATE THE NEXT LOWEST PRIORITY
        BIC #37, LESS1 ;INSURE THAT THE TNZVC BITS ARE CLEAR
        MOV DZRIV, RO ;PLACE THE BASE VECTOR ADDRESS IN RO
        ADD #2, RO ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
        MOV RO, DZ RIS ;STORE IT HERE
        ADD #2, RO ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
        MOV RO, DZTIV ;STORE IT HERE
        ADD #2, RO ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
        MOV RO, DZTIS ;STORE IT HERE
    
```

```

;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN DZ11. $BASE IS THE BASE ADDRESS
;OF THE DEVICE
    
```

```

MOV $BASE, RO ;COPY THE ADDRESS BEING LOADED
MOV RO, DZCSR ;XXX0
INC RO
MOV RO, HDZCSR ;XXX1
INC RO
MOV RO, DZRBUF ;XXX2
MOV RO, DZLPR ;XXX2
INC RO
MOV RO, HDZRBUF ;XXX3
MOV RO, HDZLPR ;XXX3
INC RO
MOV RO, DZTCR ;XXX4
INC RO
MOV RO, HDZTCR ;XXX5
INC RO
MOV RO, DZMSR ;XXX6
MOV RO, DZTDR ;XXX6
INC RO
MOV RO, HDZMSR ;XXX7
MOV RO, HDZTDR ;XXX7
RTS PC
    
```

```

DZPRT: PR5
LESS1: PR4 ;LEVEL TO ALLOW INTERRUPTS
    
```

L08

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 103
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

			;ERROR ERROR TABLE	
			.ERRTAB:	;ERROR 0
4997				
4998	026174	000000	0	
4999	026176	000000	0	
5000	026200	000000	0	
5001				
5002	026202	026406	EM1	;ERROR
5003	026204	027606	DH1	
5004	026206	030004	DT1	
5005				
5006	026210	026461	EM2	;ERROR 2
5007	026212	027631	DH2	
5008	026214	030016	DT2	
5009				
5010	026216	026507	EM3	;ERROR 3
5011	026220	027664	DH3	
5012	026222	030034	DT3	
5013				
5014	026224	026546	EM4	;ERROR 4
5015	026226	027664	DH3	
5016	026230	030034	DT3	
5017				
5018	026232	026575	EM5	;ERROR 5
5019	026234	027676	DH4	
5020	026236	030042	DT4	
5021				
5022	026240	026624	EM6	;ERROR 6
5023	026242	027676	DH4	
5024	026244	030042	DT4	
5025				
5026	026246	026662	EM7	;ERROR 7
5027	026250	027664	DH3	
5028	026252	030034	DT3	
5029				
5030	026254	026723	EM8	;ERROR 10
5031	026256	027664	DH3	
5032	026260	030034	DT3	
5033				
5034	026262	026765	EM9	;ERROR 11
5035	026264	027664	DH3	
5036	026266	030034	DT3	
5037				
5038	026270	027023	EM10	;ERROR 12
5039	026272	027664	DH3	
5040	026274	030034	DT3	
5041				
5042	026276	027062	EM13	;ERROR 13
5043	026300	027664	DH3	
5044	026302	030034	DT3	
5045				
5046	026304	027113	EM14	;ERROR 14
5047	026306	027664	DH3	
5048	026310	030034	DT3	
5049				
5050	026312	027145	EM15	;ERROR 15
5051	026314	000000	0	
5052	026316	000000	0	

5089
5090

```

026406 047200 020117 046123 EM1: .ASCIZ <200>/NO SLAVE SYNC RESPONSE FROM DZ11 REGISTER/
026461      200 042522 044507 EM2: .ASCIZ <200>/REGISTER R/W FAILURE?
026507      200 051124 047101 EM3: .ASCIZ <200>/TRANSMIT READY (TRDY) NOT SET/
026546 051200 041505 044505 EM4: .ASCIZ <200>/RECEIVER DONE NOT SET/
026575      200 040504 040524 EM5: .ASCIZ <200>/DATA COMPARISON ERROR/
026624 042200 030532 020061 EM6: .ASCIZ <200>/DZ11 *RECEIVER BUFFER* ERROR/
026662 052200 040522 051516 EM7: .ASCIZ <200>/TRANSMITTER FAILED TO INTERRUPT/
026723      200 047125 054105 EM8: .ASCIZ <200>/UNEXPECTED TRANSMITTER INTERRUPT/
026765      200 042522 042503 EM9: .ASCIZ <200>/RECEIVER FAILED TO INTERRUPT/
027023      200 047125 054105 EM10: .ASCIZ <200>/UNEXPECTED RECEIVER INTERRUPT/
027062 051600 046111 020117 EM13: .ASCIZ <200>/SILO ALARM SET TOO SOON/
027113      200 044523 047514 EM14: .ASCIZ <200>/SILO ALARM FAILED TO SET/
027145      200 041501 044524 EM15: .ASCIZ <200>/ACTION DETECTED ON INVALID LINE./
027207      200 042522 042101 EM16: .ASCIZ <200>/READING DZRBUF DID NOT CLEAR SILO ALARM/
027260 042200 052101 020101 EM17: .ASCIZ <200>/DATA VALID SHOULD NOT BE SET/
027316 051200 041505 044505 EM20: .ASCIZ <200>/RECEIVER DONE SHOULD NOT BE SET/
027357      200 042522 040514 EM21: .ASCIZ <200>/RELATIVE TIMING ERROR./
027407      200 047515 042504 EM22: .ASCIZ <200>/MODEM SIGNAL ERROR ON CABLE TEST/
027451      200 040504 040524 EM23: .ASCIZ <200>/DATA VALID IS NOT SET!/
027501      200 040504 040524 EM24: .ASCIZ <200>/DATA OVERRUN IS SET!/
027527      200 051106 046501 EM25: .ASCIZ <200>/FRAMING ERROR OCCURRED/
027557      200 040520 044522 EM26: .ASCIZ <200>/PARITY ERROR OCCURRED/

027606 052200 040522 020120 DH1: .ASCIZ <200>/TRAP PC DZ11 REG/
027631      200 054105 042520 DH2: .ASCIZ <200>/EXPECTED FOUND REGISTER/
027664 046200 047111 020105 DH3: .ASCIZ <200>/LINE NO./
027676 042600 050130 041505 DH4: .ASCIZ <200>/EXPECTED FOUND LINE/
027725      200 054124 046040 DH5: .ASCIZ <200>/TX LINE PREVIOUS TIME ACTUAL TIME PARAMETER/
    
```

.EVEN

```

030004 000002          DT1: 2 .DATA TABLES FOR ERROR MESSAGES
030006      006      003      .BYTE 6,3
030010 001204          $REG1
030012      006      001      .BYTE 6,1
030014 001202          $REG0

030016 000003          DT2: 3
030020      006      004      .BYTE 6,4
030022 001214          $REG5
030024      006      001      .BYTE 6,1
030026 001212          $REG4
030030      006      001      .BYTE 6,1
030032 001202          $REG0

030034 000001          DT3: 1
030036      003      001      .BYTE 3,1
030040 001372          SAVL,IN

030042 000003          DT4: 3
030044      006      004      .BYTE 6,4
030046 001214          $REG5
030050      006      001      .BYTE 6,1
030052 001212          $REG4
030054      003      001      .BYTE 3,1
    
```

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76
DZ11 DEVICE DIAGNOSTICS.

15:03 PAGE 106
COPYRIGHT 1976 DIGITAL EQUIP. CORP.

030056 001372

SAVLIN

030060 000004

DTS: 4

030062 003 005

.BYTE 3,5

030064 001372

SAVLIN

030066 006 011

.BYTE 6,9.

030070 001214

\$REGS

030072 006 007

.BYTE 6,7

030074 001220

\$TMP1

030076 006 001

.BYTE 6,1

030100 001400

REGIST

030102 000001

CORMAX:
.END

RDONE = 000200	789#	3016	3286	3314	3374	3401	3473	3541	3613	3686	4759	4764	4769*
RECDAT 025322	4666*	4669	4673	4674	4676*	4677*	4678	4750*	4753	4756			
REGIST 001400	4770*	4771	4948#										
RESREG 007054	1142#	3368*	3452*	4163*	4275*	4373*	4444*	5090					
RESTAR 011254	2257	2260#											
RESVEC= 000010	2377	2514#											
RESOS = 104410	758#												
RIE = 000100	1342#	2260											
	788#	2856	3054	3539	3611	3698	3737	3903	3989	4178	4180	4247	4252
	4295	4345	4350	4393	4478	4653	4690	4724					
RINGO = 000001	893#												
RING1 = 000002	894#												
RING2 = 000004	895#												
RING3 = 000010	896#												
RING4 = 000020	897#												
RING5 = 000040	898#												
RING6 = 000100	899#												
RING7 = 000200	900#												
RL0 = 000000	818#												
RL1 = 000400	819#												
RL2 = 001000	820#												
RL3 = 001400	821#												
RL4 = 002000	822#												
RL5 = 002400	823#												
RL6 = 003000	824#												
RL7 = 003400	825#												
RSTART 020212	3972#												
RSTRT 024622	4796#												
RUN 001406	1145#	1424*	2458	2460*	2461*	2467*	2468*						
RXISR1 022676	4174	4436#											
RXSVC 020514	3984	4037#											
RXTCR 020770	3971#	4000	4074*	4101#									
RO =%000000	679#	1460*	1462*	1463	1565*	1574	1576	1578	1579	1580	1582	1583	1584
	1658#	1659	1660	1671*	1679*	1691*	1693*	1695	1696	1754*	1757	1874	1875*
	1880	1885	1888*	1940	1948*	1952	1953	1955*	1956*	1957	1979*	2093	2098*
	2110	2119*	2136	2139*	2149*	2163	2164*	2165	2166*	2167*	2168*	2169*	2187
	2188#	2189*	2191*	2346	2371*	2469*	2475	2476	2477	2480	2481	2482	2497*
	2498	2500	2502	2504	2507	2508	2521*	2543*	2544*	2545*	2546*	2547*	2549*
	2550*	2551*	2553*	2580*	2583*	2590*	2597*	2634*	2643*	2645*	2647	2658*	2687*
	2688	2690*	2693*	2694	2696*	2699*	2700	2702*	2705*	2706	2708*	2727*	2729*
	2730	2738	2759*	2761*	2762	2766*	2767	2773*	2775	2791*	2793*	2794	2798*
	2799	2805*	2807	2823*	2825*	2826	2830*	2831	2837*	2839	2855*	2857*	2858
	2862*	2863	2869*	2871	2887*	2889*	2890	2894*	2895	2901*	2903	2918*	2921*
	2922	2927*	2928	2936*	2938	2961*	2967*	2968	2973*	2974	2982*	2984	2986
	2990	3014*	3016*	3018	3021*	3022	3028*	3031	3052*	3053*	3055	3062*	3063
	3078*	3080	3089	3104*	3106	3114	3140*	3165	3170	3198*	3211	3217	3235*
	3241*	3243	3249	3327*	3328*	3363*	3364*	3365	3367*	3384	3398	3412*	3416*
	3417*	3442*	3443	3447*	3766*	3767*	3768	3770*	3776*	3777*	3790	3791*	3897*
	3898*	3899	3901*	3913*	3914*	3923	3924*	3925	3935	3973*	3974	3978*	3990*
	3981*	3982	4032*	4135*	4136*	4137	4164*	4165*	4166	4190*	4191*	4192	4238*
	4239*	4240	4242*	4336*	4337*	4338	4340*	4412*	4416*	4418	4467*	4468*	4471
	4472	4474	4879*	4882*	4963*	4964*	4965	4966*	4967	4968*	4969	4973*	4974
	4975*	4976	4977*	4978	4979	4980*	4981	4982	4983*	4984	4985*	4986	4987*
	4988	4989	4990*	4991	4992								
R1 =%000001	680#	1566*	1572*	1574*	1575*	1576*	1577*	1578*	1579*	1583*	1584*	1692*	1693
	1694*	1695	1941	1978*	2092	2099*	2111	2115*	2117	2118	2119	2120	2148*

R2 =%000002

2347	2370*	2522*	2540*	2543	2576*	2578*	2579*	2581	2586	2588	2591*	2592
2594	2600*	2601*	2603	2614*	2615	2620*	2621	2623*	2644*	2645	2646*	2647
2648	2655*	2656*	2688*	2694*	2700*	2706*	2713*	2717	3272*	3276	3277*	3354*
3358	3359*	3440*	3441*	3445	3446*	3501*	3505	3506*	3569*	3573	3574*	3646*
3650	3651*	3757*	3761	3762*	3888*	3892	3893*	3972*	3976	3977*	4220*	4221*
4222*	4226	4228*	4229	4230*	4231*	4269*	4270*	4271*	4272*	4278	4281	4283
4285	4289*	4318*	4319*	4320*	4324	4326*	4327	4328*	4329*	4367*	4368*	4369*
4370*	4376	4379	4381	4383	4387*	4402*	4405*	4406*	4407	4408*	4410	4413*
4414*	4446*	4447*	4448*	4449*	4450	4452	4455	4458*	4460	4475	4540*	4542*
4544	4545	4729*	4762	4774	4778*	4886*	4888	4890	4895			
681*	1567*	1580*	1581*	1582*	1637*	1641*	1643*	2091	2100*	2348	2369*	2523*
2524*	2541*	2544	2569*	2571*	2572	2575*	2603*	2604	2605*	2606*	2607*	2608
2609*	2610*	2642*	2654*	2655	2660*	2662*	2664*	2665*	2666*	2735*	2744*	3030*
3035*	3143*	3148	3151*	3154	3164	3169	3199*	3200	3202*	3206	3208	3209
3215	3237*	3238	3240	3255*	3273*	3274	3278*	3281*	3283	3285	3325	3329*
3355*	3356	3360*	3369*	3371	3373	3414	3418*	3450*	3453	3455	3456	3486
3487*	3502*	3503	3507*	3570*	3571	3575*	3647*	3648	3652*	3660*	3661	3664*
3669	3730	3758*	3759	3763*	3771*	3773	3778	3848	3851*	3864	3889*	3890
3894*	3902*	3905	3915	3953	3956*	4018*	4021*	4023	4031	4053*	4056*	4058
4074	4084*	4085	4093*	4168*	4223*	4224	4232*	4235*	4293	4321*	4322	4330*
4333*	4391	4409*	4462	4541*	4542	4543*	4544	4730*	4785	4786*	4815*	4816*
4817	4821*	4824	4881*	4884*	4887							

R3 =%000003

682*	1611*	1612*	1615*	1621*	1638*	1639*	1640*	1642*	1644*	1645	1993	2000*
2010*	2013*	2014	2019*	2090	2101*	2112	2122*	2123*	2124*	2125	2132*	2133*
2138*	2141*	2147*	2349	2368*	2526*	2527*	2528*	2529*	2530*	2531*	2532*	2545
2570*	2608*	2737*	2740*	3086*	3087*	3088	3111*	3112*	3113	3142*	3150*	3155
3313*	3317*	3400*	3709*	3710*	3711*	3717*	3722*	3723	3782*	3792	3808*	3809
4014*	4016*	4017	4019*	4049*	4050*	4051*	4052	4054*	4061*	4062*	4063	4065
4071*	4072	4082*	4087	4092	4169*	4236*	4290*	4293	4334*	4388*	4391	4459*
4462												

R4 =%000004

683*	1994	1999*	2003*	2004*	2005	2012*	2015	2018*	2026	2035*	2036	2038
2040	2042*	2043	2044	2066*	2067*	2071*	2089	2102*	2113	2120*	2122	2126*
2127*	2128*	2146*	2220*	2221	2222*	2223*	2224*	2225*	2226*	2227	2228	2229
2350	2367*	2533*	2539	2547	2650*	2651*	2730*	2731	2738*	2762*	2763	2767*
2775*	2794*	2795	2799*	2807*	2826*	2827	2831*	2839*	2858*	2859	2863*	2871*
2890*	2891	2895*	2903*	2922*	2923*	2924	2928*	2929*	2930	2938*	2939*	2940
2968*	2969*	2970	2974*	2975*	2976	2984*	2985*	2990*	2991*	2992	3018*	3022*
3031*	3032	3055*	3056	3063*	3064	3080*	3081	3083	3086	3089*	3090	3106*
3107	3108	3111	3114*	3115	3154*	3157*	3159*	3161	3163	3165*	3166	3170*
3211*	3212	3217*	3242*	3246*	3249*	3250	3320*	3321	3404*	3407*	3408	3472*
3476*	3479*	3480	3677*	3681*	3685*	3689*	3708*	3709	3726*	3727	3783*	3787*
3799*	3803*	3811*	3813*	3830*	3831	3844*	3845	3916*	3920*	3936*	3940*	4017*
4026*	4027*	4028	4029*	4043*	4046	4049	4063	4066*	4266*	4269	4273	4276
4281	4287*	4364*	4367	4371	4374	4379	4385*	4439*	4442	4446	4450	4454*
4627*	4632*	4635*	4712*	4713	4764*	4765*	4766	4807*	4808	4810	4812	4814*
4816	4818	4840*	4841*									

R5 =%000005

684*	2025	2027*	2028	2029	2030	2031	2032	2033	2034*	2043*	2046*	2047*
2048*	2057	2059	2061	2067	2068*	2072*	2088	2103*	2114	2118*	2121	2129*
2145*	2218*	2219*	2220	2223	2351	2366*	2393*	2397*	2402*	2407*	2534*	2535
2536*	2728*	2729	2731	2736*	2760*	2761	2763	2766	2769	2770*	2772*	2773
2777*	2792*	2793	2795	2798	2801	2802*	2804*	2805	2809*	2824*	2825	2827
2830	2833	2834*	2836*	2837	2841*	2856*	2857	2859	2862	2865	2866*	2868*
2869	2873*	2888*	2889	2891	2894	2897	2898*	2900*	2901	2905*	2919*	2921
2924	2927	2932	2933*	2935*	2936	2942	2943*	2945*	2947*	2962*	2967	2970
2973	2978	2979*	2981*	2982	2986	2994	2995*	2997*	2999*	3015*	3025*	3029*
3032	3054*	3056	3059*	3064	3081*	3082*	3083	3090	3107*	3108	3115	3160*

R6 =%000006
R7 =%000007
SAVLIN 001372

SAVNUM 001411
SAVPC 001402
SAVOS = 104407
SCOP1 = 104401

SERV.G 007212
SET 025076
SETAPT 011260
SETFLG= 104406
SET.PS 010620
SET1 025130
SEVEN = 000020
SEVENS= 000060
SILOAL= 026000
SILOEN= 010000
SIX = 000010
SIXS = 000050
SNAP 020342
SP =%000006

SPACNT 006457
SPEED 025312
SPIN 024346
STACK = 001120
STFLG 025274
STKLMT= 177774
STOP 001462
SVOS 006214
SWR 001160

3161*	3162*	3163*	3166	3172*	3205*	3206*	3207*	3208*	3212	3219*	3236*	3250
3253*	3289*	3293*	3297*	3303*	3308*	3309*	3310*	3312*	3321	3377*	3381*	3385*
3391*	3396*	3397*	3398*	3399*	3408	3458*	3464*	3469*	3470*	3471*	3480	3518*
3525*	3540*	3547*	3587*	3595*	3612*	3620*	3676*	3727	3816*	3822*	3827*	3828*
3829*	3831	3837*	3838	3840*	3841*	3842	4065*	4276*	4277*	4278*	4279*	4283*
4284*	4374*	4375*	4376*	4377*	4381*	4382*	4452*	4453*	4474*	4597*	4628*	4636
4707*	4708*	4709*	4710*	4713	4762*	4763*	4766	4799*	4800	4801	4802	4803
4804	4805	4806*	4817*	4820*	4821	4822*	4823*	4824*	4831	4833	4835	4841
4842*	4858*	4862*	4866*	4893*	4896*							
685#	687											
686#	688											
1139#	3280*	3297	3326*	3327	3362*	3385	3415*	3416	3449*	3458	3485*	3509*
3577*	3654*	3663*	3723	3765*	3776	3816	3850*	3896*	3913	3955*	4052*	4061
4083*	4091*	4128*	4234*	4285*	4286*	4332*	4383*	4384*	4455*	4456*	4475*	4476*
4594	4678	4771	4878	4879	5090							
1147#	1420*	1557*	1742*	1744*	2621*							
1143#	2084*	2284										
1340#	2217											
1328#	2716	2946	2998	3174	3221	3324	3339	3411	3484	3849	3871	3954
4188	4719											
1794	2205	2288#	2295	4187	4638	4658	4739					
4597	4878#											
1451	2521#											
1338#	1535											
955	2426#											
4883	4886#											
841#												
845#												
791#	3016	3794	3800	3834	3927	3937	4040					
790#	2824	3054	3772	3861	3870	3903						
840#												
844#												
3995#	4075											
687#	1417*	1431*	1432*	1440	1441	1519*	1520*	1521	1523	1525	1702*	1796
1801*	1807*	1810	1812	1813	1831	1833*	1849*	1874*	1875	1885*	1887	1888
1889*	1891	1893	1895	1901	1903*	1905*	1913*	1917	1921	1922	1926	1940*
1941*	1948	1949*	1960	1961*	1962*	1972	1973*	1978	1979	1993*	1994*	1995
1996*	2012	2013	2014*	2015*	2018	2019	2025*	2026*	2027	2033*	2071	2072
2084	2110*	2111*	2112*	2113*	2114*	2115	2116*	2125*	2133	2145	2146	2147
2148	2149	2163*	2164	2187*	2191	2198*	2213	2215	2218	2271	2279*	2290*
2291*	2292	2303*	2304*	2305	2310	2316	2318	2320*	2321	2324*	2325*	2326
2346*	2347*	2348*	2349*	2350*	2351*	2352*	2353	2361*	2365	2366	2367	2368
2369	2370	2371	2376*	2393	2409*	2426	2431	2436	2438*	2439*	2637*	2664
2668*	2713	2714	2769*	2772	2801*	2804	2833*	2836	2865*	2868	2897*	2900
2932*	2935	2942*	2945	2978*	2981	2994*	2997	3529	3532	3552	3555	3604
3625	3949	3952	4075*	4296*	4298*	4394*	4396*	4530*	4600*	4686*	4776*	4779*
4799	4805*	4858	4867*									
2153#												
4624	4646	4721	4895*	4942#								
4736#	4742	4779										
662#	1417	1702	2279	4530	4600							
4533*	4588*	4605	4607*	4611	4613*	4935#						
673#												
1177#	3365	3768	3899	3982	4137	4166	4240	4338				
2088#												
1016#	1434	1438*	1453	1457	1552	1586	1651	1668	1673	1678	1681	1706

H10

MD-11-DZDZA-B MACY11 27(732) 03-NOV-76 15:03 PAGE 126
DZDZAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

.SASTA= ***** U	1936	1939
.SX = 001462	1183*	1188

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 130
CROSS REFERENCE TABLE -- MACRO NAMES

.SSCOP	18	639#	1781
.SSIZE	18		
.SSUPR	18		
.STRAP	18	-639#	
.STYPB	18		
.STYPD	18		
.STYPE	18	639#	1852
.STYPO	18		
.S4OCA	18		
.1170	18		

ADC	2461	2468													
ADD	1575	1577	1643	1644	1889	1949	1961	1973	1996	2068	2116	2124	2223	2226	2409
	2438	2462	2470	2540	2541	2614	2662	3253	4189	4198	4564	4567	4571	4817	4824
ASL	4842	4867	4964	4966	4968										
	2046	2047	2048	2167	2222	2224	3159	3328	3417	3777	3914	4021	4027	4056	4062
ASLB	4272	4370	4406	4416	4449	4820	4822	4823	4884	4955	4956	4957	4958	4959	
	1507	1508	1509	1510	1511	2947	2999	3151	3202	3255	3278	3329	3360	3418	3447
ASR	3487	3507	3575	3652	3664	3763	3851	3894	3956	3978	4093	4143	4232	4330	
	1570	1615	1621	1956	2127	2128	3157	3303	3391	3464	3717	3822	4286	4384	4413
BCC	4456	4476													
	1616	1622	2652	3152	3203	3256	3279	3330	3361	3419	3448	3488	3508	3576	3653
BCC	3764	3895	3979	4094	4144	4154	4233	4331							
BCS	1571	1613	2315	2537	3304	3392	3465	3718	3823	3852	3957				
BEQ	1435	1443	1522	1553	1559	1614	1620	1662	1669	1682	1707	1755	1797	1817	1823
	1846	1848	1879	1892	1927	1943	1947	1967	1969	2006	2037	2045	2137	2207	2214
	2231	2237	2248	2253	2257	2268	2277	2289	2306	2434	2487	2538	2587	2589	2593
	2595	2613	2619	2624	2732	2739	2764	2768	2776	2796	2800	2808	2828	2832	2840
	2860	2864	2872	2892	2896	2904	2925	2931	2941	2971	2977	2993	3019	3023	3033
	3057	3065	3084	3091	3109	3116	3156	3167	3171	3213	3218	3239	3251	3275	3284
	3287	3299	3322	3337	3357	3372	3375	3387	3409	3438	3444	3454	3460	3481	3504
	3520	3542	3572	3589	3614	3649	3713	3724	3728	3760	3795	3818	3832	3835	3843
	3891	3926	3928	3934	3975	4041	4047	4064	4086	4088	4181	4195	4225	4253	4282
	4323	4325	4351	4380	4443	4451	4461	4463	4470	4539	4574	4603	4640	4670	4679
	4714	4754	4757	4760	4767	4772	4809	4819	4889						
BGE	1826														
BGT	1750	2041	2319	4813											
BHI	2058	4473	4832												
BHIS	3793														
BIC	1520	1524	1581	1747	2123	2225	2291	2304	2320	2325	2394	2479	2524	2532	2550
	2666	2766	2798	2830	2862	2894	2923	2927	2929	2939	2973	3082	3325	3414	3486
	3533	3599	3605	3711	3726	3730	3737	3848	3870	3953	4016	4026	4031	4051	4066
	4074	4140	4148	4161	4191	4221	4230	4271	4277	4284	4287	4295	4319	4328	4369
	4375	4382	4385	4393	4405	4420	4448	4453	4454	4478	4675	4676	4763	4765	4769
	4962														
BICB	2004	2042	3169	3215	4418	4814	4859								
BIS	1527	1582	1640	2175	2321	2549	2578	2579	2591	3027	3028	3029	3062	3240	3241
	3282	3312	3370	3399	3441	3451	3471	3517	3539	3586	3611	3659	3697	3698	3772
	3829	3841	3861	3864	3865	3903	3988	3989	3990	4141	4159	4162	4178	4222	4228
	4247	4279	4320	4326	4345	4377	4624	4646	4648	4722	4723	4724			
BISB	2043	2183	3161	3163	3164	3206	3208	3209	3310	3398	4278	4376	4708	4710	4816
BIT	1453	1552	1586	1651	1668	1681	1706	1802	1820	1845	2176	2206	2211	2274	2276
	2294	2458	2486	2586	2588	2592	2594	2986	3155	3243	3274	3283	3286	3290	3314
	3336	3356	3371	3374	3378	3401	3443	3453	3473	3503	3519	3541	3571	3588	3613
	3648	3661	3686	3759	3773	3784	3794	3800	3834	3890	3905	3917	3927	3933	3937
	3974	4023	4040	4046	4058	4085	4132	4180	4194	4224	4226	4252	4273	4291	4322
	4324	4350	4371	4389	4442	4460	4538	4573	4669	4753	4756	4759			
BITB	1878	1883	1915	1946	2061	3148	3200	3238	4835						
BLO	2060	4834													
BLOS	1674	2649	3839												
BLT	1906	2039	2317	4811											
BMI	1458	2548	2582	3145	3194	3197	3679	3693	3909	4012	4020	4038	4044	4055	4267
	4365	4403	4415	4437	4440	4630	4664	4667	4748	4751	4783	4883			
BNE	1446	1454	1456	1464	1526	1587	1646	1650	1652	1655	1667	1697	1705	1743	1803
	1821	1836	1877	1884	1886	1894	1902	1916	1923	1945	1951	1954	1971	2011	2062
	2070	2131	2135	2140	2144	2177	2190	2212	2233	2262	2275	2293	2295	2327	2364
	2396	2401	2406	2430	2453	2459	2464	2473	2485	2499	2501	2503	2509	2573	2584

M10

MD-11-DZDZA-B
DZDZAB.P11

MACY11 27(732) 03-NOV-76 15:03 PAGE 133
CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	2598	2616	2659	2743	2745	2948	2987	3000	3036	3149	3201	3244	3247	3291	3294
	3315	3318	3366	3379	3382	3402	3405	3413	3474	3477	3526	3548	3596	3621	3662
	3682	3687	3690	3769	3774	3785	3788	3801	3804	3810	3814	3869	3900	3906	3918
	3921	3938	3941	3983	3999	4001	4004	4024	4030	4059	4073	4133	4138	4167	4183
	4185	4193	4227	4241	4255	4274	4292	4294	4339	4353	4372	4390	4392	4411	4419
	4431	4546	4606	4612	4633	4637	4684	4742	4775	4836	4844	4651	4865	4891	
BPL	1450	1518	1610	1800	1871	1920	2002	2008	2209	2270	2302	2309	2323	2965	3138
	3671	3674	3780	3846	4151	4218	4257	4316	4355	4656	4737				
BR	1436	1528	1562	1585	1619	1624	1665	1677	1710	1798	1805	1811	1814	1873	1899
	1909	1918	1925	1937	1959	2017	2050	2052	2266	2356	2380	2399	2404	2412	2432
	2456	2466	2506	2511	2542	2552	2585	2599	2627	2632	2636	2663	3158	3175	3222
	3306	3340	3394	3467	3523	3530	3545	3550	3553	3593	3602	3618	3623	3626	3665
	3720	3732	3798	3825	3931	3943	3948	3951	4022	4057	4156	4197	4199	4417	4609
	4777	4825	4827	4863	4870	4885	4894								
CLC	1568	1623	2311	3307	3395	3468	3721	3826	4152	4155					
CLR	1421	1425	1426	1427	1462	1516	1671	1694	1729	1745	1795	1819	2034	2199	2307
	2362	2551	2571	2580	2590	2600	2601	2609	2634	2690	2696	2702	2708	2711	2735
	2736	2737	2770	2777	2802	2809	2834	2841	2866	2873	2898	2905	2933	2943	2949
	2979	2995	3001	3015	3026	3030	3142	3160	3172	3205	3219	3242	3280	3289	3313
	3362	3364	3367	3377	3400	3420	3449	3472	3489	3509	3518	3540	3577	3587	3612
	3628	3654	3676	3677	3685	3739	3741	3765	3767	3770	3782	3783	3799	3811	3896
	3898	3901	3916	3936	3959	3981	3995	4082	4083	4097	4099	4127	4128	4129	4136
	4142	4165	4168	4169	4170	4171	4234	4235	4236	4239	4242	4250	4251	4332	4333
	4334	4337	4340	4348	4349	4533	4534	4535	4537	4543	4588	4601	4627	4635	4707
	4729	4730	4746	4794	4806	4815	4880								
CLRB	1422	1461	1604	1730	1818	1834	1898	1924	1975	1976	1977	2141	2216	2398	2408
	2574	2969	2975	2985	2991	3483	3684	4536	4795	4862					
CMP	1434	1463	1645	1661	1673	1695	1696	1796	1812	1825	2057	2059	2213	2267	2288
	2316	2318	2429	2433	2463	2471	2498	2500	2502	2508	2572	2615	2647	2648	2714
	2731	2763	2795	2827	2859	2891	2924	2970	3032	3056	3064	3083	3090	3108	3115
	3166	3212	3250	3321	3365	3408	3480	3529	3532	3552	3555	3604	3625	3723	3768
	3792	3809	3831	3899	3925	3949	3952	3982	4087	4137	4166	4192	4240	4293	4338
	4391	4410	4462	4472	4544	4545	4678	4713	4766	4771	4831	4833	4888		
CMPB	1455	1521	1525	1649	1876	1891	1893	1901	1922	1926	1944	2005	2036	2038	2040
	2044	2261	2292	2305	2326	2395	2400	2405	2612	3727	3838	3842	4063	4281	4379
	4450	4639	4683	4774	4808	4810	4812	4818	4860	4864					
COM	3087	3112	4607	4613											
DEC	1558	1739	1748	2010	2129	2143	2189	2583	2623	2744	3868	3998	4414	4429	4632
	4882														
DECB	1448	1465	1590	1656	1742	1905	1908	2069	2134	2139	2568	4019	4054	4843	
EMT	663														
HALT	951	1672	1676	1680	1872	2272	2355	2379	2435	2455	2635				
INC	1736	1746	1824	1974	2273	2363	2597	2658	2740	2455	2635				
	3404	3415	3446	3476	3485	3506	3525	3547	3574	3595	3620	3246	3277	3326	3359
	3762	3791	3803	3808	3837	3840	3850	3893	3920	3924	3940	3651	3663	3681	3689
	4071	4091	4139	4182	4231	4254	4289	4290	4329	4352	4387	3955	3977	4003	4032
	4459	4569	4741	4975	4977	4980	4983	4985	4987	4990		4388	4408	4409	4458
	4829	1928	2611	3293	3317	3381	3412	3787	3813	4029	4184	4256	4354	4778	4786
INCB	664														
IOT	982	984	1444	1452	1459	1573	1588	1591	1714	1762	2280	2514	2717	3147	3705
JMP	3775	3853	3872	3907	3958	4002	4007	4615	4659	4740	4745	4796			
JSR	1451	1554	1589	1653	1757	1794	1881	1900	1907	1914	1963	2205	2264	2483	4187
	4465	4597	4638	4658	4739										
MOV	1417	1419	1423	1424	1428	1431	1432	1433	1438	1439	1440	1441	1460	1481	1494
	1519	1523	1549	1550	1555	1563	1564	1565	1566	1567	1572	1574	1576	1578	1579
	1580	1583	1584	1611	1637	1638	1641	1642	1658	1659	1660	1678	1679	1691	1692

RTI	1838	1850	1890	2020	2073	2094	2104	2150	2178	2184	2192	2200	2281	2378	2410
	2440	2638	2669	3874	4033	4076	4297	4299	4395	4397	4421	4464	4479	4687	4691
	4780	4787	4845	4868											
RTS	1647	1930	1980	2169	2329	2555	2672	4432	4896	4993					
SEC	1560	2625	3305	3393	3466	3479	3824	4157							
SUB	1955	2219	2427	2665	4196	4196									
SWAB	1639	3162	3207	3309	3397	3470	3710	3828	4050	4270	4368	4447	4677	4709	4770
TRAP	1326	1328	1330	1332	1334	1336	1338	1340	1342	1344	1346	1348	1350	1352	1354
	1356														
TST	1442	1609	1666	1704	1809	1822	1835	1847	1887	1895	1917	1950	1968	1970	2165
	2232	2236	2247	2252	2256	2269	2452	2484	2507	2539	2581	2604	2930	2940	2976
	2992	3144	3437	3670	3678	3692	3845	3908	4011	4092	4150	4217	4315	4469	4471
	4602	4605	4611	4629	4657	4738	4782	4890							
TSTB	1445	1449	1457	1517	1654	1799	1816	1870	1919	1942	1953	1966	2001	2007	2136
	2208	2230	2301	2308	2322	2618	2964	3137	3193	3196	3298	3386	3459	3673	3712
	3779	3817	4000	4037	4072	4436	4655	4663	4736	4747					
WAIT	3867														
.ASCII	1039	1040	1648	4950											
.ASCIZ	988	1041	1648	2331	2338	2382	4950	5090							
.BLKB	1146														
.BLKW	1144	1206	1207	1208	1209	1210	1211	1213	1214	1215	1216	1217	1218	1220	1221
	1222	1223	1224	1225	1227	1228	1229	1230	1231	1232	1234	1235	1236	1237	1238
	1239	1241	1242	1243	1244	1245	1246	1248	1249	1250	1251	1252	1253	1255	1256
	1257	1258	1259	1260	1262	1263	1264	1265	1266	1267	1269	1270	1271	1272	1273
	1274	1276	1277	1278	1279	1280	1281	1283	1284	1285	1286	1287	1288	1290	1291
	1292	1293	1294	1295	1297	1298	1299	1300	1301	1302	1304	1305	1306	1307	1308
	1309	1311	1312	1313	1314	1315	1316								
.BYTE	998	999	1004	1005	1013	1014	1022	1023	1024	1025	1057	1058	1068	1069	1076
	1077	1079	1080	1082	1083	1147	1154	1155	1156	1157	1158	1479	1480	1492	1493
	1504	1505	1546	1547	1602	1603	1635	1636	1689	1690	1766	1769	1772	1775	1981
	1982	1983	2078	2079	2152	2153	2283	2286	2335	2383	2385	2495	2495	4553	4554
	4561	4562	4585	4586	4595	4596	4849	4850	5090						
.ENABL	1	639													
.END	5090														
.ENDC	644	663	755	769	970	974	976	992	996	998	1026	1034	1038	1039	1043
	1046	1068	1076	1079	1082	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094
	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1111	1181
	1183	1190	1203	1723	1724	1726	1745	1750	1753	1754	1756	1762	1764	1784	1787
	1790	1802	1804	1815	1816	1818	1820	1824	1829	1831	1832	1840	1841	1856	1885
	1935	1936	1939	1966	1981	2343	2352	2353	2359	2365	2366	2376	2378	2382	2443
	2674	2676	2679	2681	2682	2684	2719	2722	2724	2725	2727	2749	2754	2756	2757
	2759	2781	2786	2788	2789	2791	2813	2818	2820	2821	2823	2845	2850	2852	2853
	2855	2877	2882	2884	2885	2887	2909	2913	2915	2916	2918	2921	2949	2951	2956
	2958	2959	2961	2967	3001	3003	3009	3011	3012	3014	3040	3046	3048	3049	3051
	3069	3072	3074	3075	3077	3095	3098	3100	3101	3103	3121	3130	3133	3134	3136
	3178	3187	3189	3190	3192	3225	3229	3231	3232	3234	3259	3265	3267	3268	3270
	3272	3343	3347	3349	3350	3352	3354	3424	3431	3433	3434	3436	3440	3491	3495
	3497	3498	3500	3501	3531	3556	3558	3559	3563	3565	3566	3568	3569	3603	3629
	3631	3633	3640	3642	3643	3645	3646	3743	3750	3752	3753	3755	3757	3876	3881
	3883	3884	3886	3888	3961	3965	3967	3968	3970	3971	4105	4120	4122	4123	4126
	4148	4202	4205	4212	4214	4215	4217	4220	4229	4301	4303	4310	4312	4313	4315
	4318	4327	5090												
.EQUIV	663	664	672	687	688	717	718	719	720	721	722	723	724	725	726
	745	746	747	748	749	750	751	752	753	754					
.EVEN	1046	1148	1159	1648	1984	2333	2339	2382	2387	4950	5090				
.IF	639	640	661	727	755	969	972	974	991	995	997	1026	1034	1038	1039

	1042	1043	1045	1068	1076	1079	1082	1085	1086	1087	1088	1089	1090	1091	1092
	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107
	1111	1180	1182	1189	1203	1722	1723	1724	1725	1726	1728	1749	1752	1754	1756
	1762	1764	1783	1786	1790	1794	1802	1814	1816	1817	1818	1820	1822	1830	1832
	1834	1840	1841	1855	1876	1934	1936	1939	1966	1981	2342	2352	2353	2358	2365
	2366	2374	2376	2378	2382	2426	2675	2679	2680	2682	2683	2718	2722	2723	2725
	2726	2748	2754	2755	2757	2758	2780	2786	2787	2789	2790	2812	2818	2819	2821
	2822	2844	2850	2851	2853	2854	2876	2882	2883	2885	2886	2908	2913	2914	2916
	2917	2921	2923	2950	2956	2957	2959	2960	2964	2969	3002	3009	3010	3012	3013
	3039	3046	3047	3049	3050	3068	3072	3073	3075	3076	3094	3098	3099	3101	3102
	3120	3130	3132	3134	3135	3177	3187	3188	3190	3191	3224	3229	3230	3232	3233
	3258	3265	3266	3268	3269	3271	3342	3347	3348	3350	3351	3353	3423	3431	3432
	3434	3435	3439	3490	3491	3495	3496	3498	3499	3500	3528	3554	3558	3559	3563
	3564	3566	3567	3568	3598	3627	3631	3632	3640	3641	3643	3644	3645	3742	3750
	3751	3753	3754	3756	3875	3881	3882	3884	3885	3887	3960	3965	3966	3968	3969
	3970	4104	4120	4121	4123	4125	4147	4201	4202	4212	4213	4215	4216	4219	4227
	4300	4301	4310	4311	4313	4314	4317	4325	5090						
.IFF	663	970	974	976	992	996	998	1026	1043	1046	1181	1183	1190	1723	1725
	1728	1750	1753	1764	1784	1814	1816	1818	1834	1840	1856	1935	2343	2359	2374
	2426	2676	2679	2681	2682	2683	2684	2719	2722	2724	2725	2726	2727	2749	2754
	2756	2757	2758	2759	2781	2786	2788	2789	2790	2791	2813	2818	2820	2821	2822
	2823	2845	2850	2852	2853	2854	2855	2877	2882	2884	2885	2886	2887	2909	2913
	2915	2916	2917	2918	2924	2930	2940	2951	2956	2958	2959	2960	2961	2969	2975
	2985	3003	3009	3011	3012	3013	3014	3040	3046	3048	3049	3050	3051	3069	3072
	3074	3075	3076	3077	3095	3098	3100	3101	3102	3103	3121	3130	3133	3134	3135
	3136	3178	3187	3189	3190	3191	3192	3225	3229	3231	3232	3233	3234	3259	3265
	3267	3268	3269	3270	3343	3347	3349	3350	3351	3352	3424	3431	3433	3434	3435
	3436	3491	3495	3497	3498	3499	3500	3528	3554	3559	3563	3565	3566	3567	3568
	3603	3627	3633	3640	3642	3643	3644	3645	3743	3750	3752	3753	3754	3755	3876
	3881	3883	3884	3885	3886	3961	3965	3967	3968	3969	3970	4105	4120	4122	4123
	4125	4126	4202	4205	4212	4214	4215	4216	4217	4227	4301	4310	4312	4313	4314
4327															
.IFT	639	1820	2674	2675	2679	2718	2722	2748	2754	2780	2786	2812	2818	2844	2850
	2876	2882	2908	2913	2923	2929	2939	2950	2956	2969	2975	2985	3002	3009	3039
	3046	3068	3072	3094	3098	3120	3130	3177	3187	3224	3229	3258	3265	3342	3347
	3423	3431	3490	3495	3528	3556	3558	3559	3563	3598	3627	3632	3640	3742	3750
.IFTF	3875	3881	3960	3965	4104	4120	4201	4202	4212	4227	4300	4301	4310	4325	
.IIF	639	1818	2924	2930	2940	2970	2976	2992							
	639	644	649	650	1042	1046	1745	1764	1787	1788	1789	1790	1794	1819	1832
	1840	1841	1932	2683	2684	2726	2727	2758	2759	2790	2791	2822	2823	2854	2855
	2886	2887	2913	2917	2918	2955	2960	2961	3013	3014	3050	3051	3076	3077	3082
	3102	3103	3108	3135	3136	3191	3192	3233	3234	3269	3270	3271	3351	3352	3353
	3435	3436	3439	3499	3500	3511	3523	3545	3557	3567	3568	3579	3592	3601	3617
	3630	3644	3645	3754	3755	3756	3885	3886	3887	3969	3970	4124	4126	4147	4216
	4217	4219	4314	4315	4317										
.IRP	1326	1328	1330	1332	1334	1336	1338	1340	1342	1344	1346	1348	1350	1352	1354
	1356	1728	1794	1834	1940	1941	1962	1978	1979	2346	2352	2365	2366	2675	2679
	2680	2683	2718	2722	2723	2726	2748	2754	2755	2758	2780	2786	2787	2790	2812
	2818	2819	2822	2844	2850	2851	2854	2876	2882	2883	2886	2908	2913	2914	2917
	2950	2956	2957	2960	3002	3009	3010	3013	3039	3046	3047	3050	3062	3072	3073
	3076	3094	3098	3099	3102	3120	3131	3132	3135	3177	3187	3188	3191	3224	3229
	3230	3233	3258	3265	3266	3269	3342	3347	3348	3351	3423	3431	3432	3435	3490
	3495	3496	3499	3558	3563	3564	3567	3632	3640	3641	3644	3742	3750	3751	3754
	3875	3881	3882	3885	3960	3965	3966	3969	4104	4120	4121	4125	4201	4212	4213
	4216	4300	4310	4311	4314	5090									
.LIST	1	639	650	659	769	941	951	986	988	1025	1026	1029	1030	1031	1032



	1033	1034	1035	1036	1037	1038	1043	1046	1205	1212	1219	1226	1233	1240	1247
	1254	1261	1268	1275	1282	1289	1296	1303	1310	1317	1319	1328	1330	1332	1334
	1336	1338	1340	1342	1344	1346	1348	1350	1352	1354	1356	1358	1407	1648	1715
	1745	1756	1778	1790	1989	2382	2674	2675	2679	2680	2682	2684	2718	2722	2723
	2725	2727	2748	2754	2755	2757	2759	2780	2786	2787	2789	2791	2812	2818	2819
	2821	2823	2844	2850	2851	2853	2855	2876	2882	2883	2885	2887	2908	2913	2914
	2916	2918	2950	2956	2957	2959	2961	3002	3009	3010	3012	3014	3039	3046	3047
	3049	3051	3068	3072	3073	3075	3077	3094	3098	3099	3101	3103	3120	3130	3132
	3134	3137	3177	3187	3188	3190	3193	3224	3229	3230	3232	3234	3258	3265	3266
	3268	3271	3342	3347	3348	3350	3353	3423	3431	3432	3434	3436	3490	3495	3496
	3498	3500	3558	3563	3564	3566	3568	3632	3640	3641	3643	3645	3742	3750	3751
	3753	3755	3875	3881	3882	3884	3886	3960	3965	3966	3968	3970	4104	4120	4121
	4123	4126	4201	4212	4213	4215	4217	4300	4310	4311	4313	4315	4522	4950	5090
.MACRO	1	639	989												
.MCALL	639	769	1043												
.NLIST	1	639	650	659	769	941	951	986	988	1026	1028	1029	1030	1031	1032
	1033	1034	1035	1036	1037	1038	1043	1046	1205	1212	1219	1226	1233	1240	1247
	1254	1261	1268	1275	1282	1289	1296	1303	1310	1317	1319	1328	1330	1332	1334
	1336	1338	1340	1342	1344	1346	1348	1350	1352	1354	1356	1358	1407	1648	1715
	1745	1756	1778	1790	1989	2382	2674	2675	2679	2680	2682	2684	2718	2722	2723
	2725	2727	2748	2754	2755	2757	2759	2780	2786	2787	2789	2791	2812	2818	2819
	2821	2823	2844	2850	2851	2853	2855	2876	2882	2883	2885	2887	2908	2913	2914
	2916	2918	2950	2956	2957	2959	2961	3002	3009	3010	3012	3014	3039	3046	3047
	3049	3051	3068	3072	3073	3075	3077	3094	3098	3099	3101	3103	3120	3130	3132
	3134	3137	3177	3187	3188	3190	3193	3224	3229	3230	3232	3234	3258	3265	3266
	3268	3271	3342	3347	3348	3350	3353	3423	3431	3432	3434	3436	3490	3495	3496
	3498	3500	3558	3563	3564	3566	3568	3632	3640	3641	3643	3645	3742	3750	3751
	3753	3755	3875	3881	3882	3884	3886	3960	3965	3966	3968	3970	4104	4120	4121
	4123	4126	4201	4212	4213	4215	4217	4300	4310	4311	4313	4315	4522	4950	5090
.PAGE	941	989	1111	1150	1319	1361	1383	1407	1715	2443	2517	2674	4398	4434	4997
.REM	1														
.REPT	650	659	951	1028	1034	1205	1989								
.SBTTL	650	659	769	941	967	986	989	1043	1111	1178	1407	1715	1720	1778	1781
	1853	1932	2340	2674											
.TITLE	639														
.WORD	975	997	1000	1001	1002	1003	1006	1007	1008	1009	1010	1011	1012	1015	1016
	1017	1026	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1048	1049	1050
	1051	1052	1053	1054	1055	1059	1060	1061	1074	1078	1081	1084	1085	1086	1087
	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102
	1103	1104	1105	1106	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171
	1172	1173	1174	1175	1176	1194	1195	1196	1197	1198	1199	1749	1752	1763	1882
	1929	1964	2193	2265	2337	2375	2377	2413	2414	2415	4901	4902	4903	4904	4905
	4906	4907	4908	4909	4910	4911	4912	4913	4914	4915	4916	4917	4918	4919	4920
	4921	4922	4923	4924	4925	4926	4927	4928	4929	4930	4931				

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DZDZAB.SEG/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.CO,DZDZAB.P11
RUN-TIME: 57 76 11 SECONDS
RUN-TIME RATIO: 323/146=2.2
CORE USED: 51K (101 PAGES)

