

DZ11

DZ11 LN ASYNC MUX TST AH-8783F-MC  
CZDZAFO FICHE 1 OF 1

MAR 1980  
COPYRIGHT © 76 80  
MADE IN USA



A large grid of data tables, likely representing a test log or configuration data. The grid consists of approximately 15 columns and 15 rows of smaller tables. Each cell in the grid contains a small table with multiple columns and rows of text, which is mostly illegible due to the low resolution and high contrast of the scan. The data appears to be organized in a structured, tabular format.





1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44

.RtM %

IDENTIFICATION  
-----

PRODUCT CODE: AC-8781F-MC  
PRODUCT NAME: CZDZAF0 DZ11 LN ASYNC MUX TSTS  
PRODUCT DATE: NOVEMBER, 1979  
MAINTAINER: DIAGNOSTIC ENGINEERING

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1976,1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL            PDP            UNIBUS            MASSBUS  
DEC                DECUS            DECTAPE

45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84

1. ABSTRACT

THE FUNCTION OF THE DZ11 DIAGNOSTICS IS TO VERIFY THE OPTION OPERATES ACCORDING TO SPECIFICATIONS. THE DIAGNOSTICS ALSO VERIFY THAT THE DZ11 OPERATES IN ITS ENVIRONMENT SUCH AS THE SYSTEM IN WHICH IT IS INSTALLED.

PARAMETERS MAY BE SUPPLIED TO THE PROGRAM BY EITHER 'AUTO SIZING' OR INPUT FROM THE USER ON THE CONSOLE BY HAVING SW00=1 AT START TIME. AUTO SIZING WILL BE DONE ONLY THE FIRST TIME THE PROGRAM IS STARTED AND SW07=0 AND SW00=0 AND SW03=0. THE AUTOSIZER IS DESIGNED TO DETECT DZ11 DEVICE ADDRESSES AND VECTORS AND TO DETERMINE WHETHER THE DZ11 THAT IS DETECTED IS AN EIA OR ZOMA BOARD. ALL REMAINING PARAMETERS DEFAULT TO CERTAIN VALUES (SEE SEC.8.5). CONSOLE INPUT MAY BE CONTROLLED AT ANY START TIME THROUGH THE USE OF SW00, SW03, SW04, AND SW06 (SEE SEC. 4.1.1 FOR A DETAILED DESCRIPTION OF THESE SWITCHES).

CURRENTLY THERE IS ONE STANDALONE DIAGNOSTIC (CZDZA), ONE SYSTEM FOR DEC X/11 (DZAA), AND AN ONLINE OVERLAY FOR DZITA (ITEP) - DZDZB. (ITEP) - DZDZB.

CZDZA WILL TEST ALL PARTS OF THE DZ11 SUCH AS CABLES, DIST PNL., INTERFACE MODULE ITSELF.

2. REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 8K MEMORY)  
ASR 33 (OR EQUIVALENT FOR CONSOLE)  
DZ11 INTERFACE MODULE (M7819(EIA), M7814(ZOMA))  
H3271 STAGGERED TURNAROUND CONNECTOR FOR EIA MODULE.  
H3190 STAGGERED TURNAROUND CONNECTOR FOR ZOMA MODULE.  
H325 CABLE TURNAROUND AND DIST PNL TESTING FOR EIA MODULE.  
H315 THIS MAY BE SUBSTITUTED FOR H325.

NOTE: A STAGGERED TURNAROUND CONNECTOR IS NEEDED IN ORDER TO TEST THE PARITY AND BREAK LOGIC.

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  
121  
122  
123  
124  
125  
126

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 INSTRUCTION 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)

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  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182

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 SET SW00=1 FOR USER PARAMETER INPUT FROM CONSOLE TERMINAL. ON FIRST START IF SW07=1 AND SW00=0 THE PROGRAM WILL DEFAULT TO CONSOLE PARAMETER INPUT (SW00=1).
- 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 160100  
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. 1 00 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. IF 1ST START OF PROGRAM AFTER LOADING THE OPERATOR MUST INPUT ADDRESS AND VECTOR FROM CONSOLE.
- SW 06 SET: RESELECT DZ11'S DESIRED ACTIVE
- SW 05 SET: RESERVED
- SW 04 SET: SELECT DELAY PARAMETER (SEE SEC. 4.1.1)
- SW 03 SET: EXTRA PARAMETER INPUT (SEE SEC. 4.1.1)
- SW 02 SET: LOCK ON SELECTED TEST
- \*SW 01 SET: RESTART PROGRAM AT SELECTED TEST
- \*SW 00 SET: GET USERS PARAMETERS FROM CONSOLE

\* FOR ECHO OR CABLE TESTS (PROGRAM STARTED AT LOC. 210) THIS SWITCH SET TO 1 ALLOWS THE USER TO TYPE IN THE VECTOR AND THE CSR FOR THE DZ11 UNDER TEST.  
\*\* FOR ECHO OR CABLE TEST THIS SWITCH SET TO 1 ALLOWS THE SELECTION OF EITHER THE ECHO OR CABLE TEST, BAUD RATE, AND THE LINE NUMBER UNDER TEST.

183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228

4.1.1 SWITCH REGISTER CONTROL OF PARAMETER INPUT FROM CONSOLE

SW 00 GET USERS PARAMETERS FROM CONSOLE. SETTING THIS SWITCH AT START UP TIME ALLOWS THE USER TO INPUT AT THE CONSOLE TERMINAL THE FOLLOWING PARAMETERS: BASE DEVICE ADDRESS, BASE VECTOR ADDRESS, BUS REQUEST LEVEL, DECLARE EIA OR 20MA MODULE, MODE OF OPERATION (EXTERNAL, INTERNAL, OR STAGGERED), AND THE NUMBER OF DZ11'S THAT ARE RUNNING. USING THIS SWITCH ALONE DEFAULTS THE FOLLOWING PARAMETERS: ALL 8 LINES ARE SET TO BE TESTED ON EACH DZ11, THE DEFAULT BAUD RATE IS SET AT 19.2 KBAUD, AND THE CHARACTER LENGTH FOR THE MAJORITY OF TESTING IS SET AT EIGHT BITS PER CHARACTER WITH TWO STOP BITS.

SW 03 EXTRA PARAMETER INPUT. SETTING THIS SWITCH AT START UP TIME PROVIDES THE USER WITH THE ABILITY TO SET THE LINES ACTIVE FOR TESTING AND TO SET THE DEFAULT BAUD RATE USED FOR THE MAJORITY OF THE DIAGNOSTIC TESTS. THE DELAY PARAMETER IS AUTOMATICALLY ADJUSTED TO THE BAUD RATE GIVEN BY THE USER.

SW 04 SELECT DELAY PARAMETER. THE DELAY PARAMETER THIS SWITCH CONTROLS DETERMINES THE LENGTH OF TIME THE PROGRAM STALLS WAITING FOR A CHARACTER TO BE COMPLETELY TRANSMITTED OR RECEIVED. THIS DELAY COUNT IS AUTOMATICALLY SET TO PROVIDE ENOUGH DELAY TIME FOR THE DEFAULT BAUD RATE SPECIFIED WHEN RUNNING THE PROGRAM ON AN 11/45 WITH BIPOLAR MEMORY. WHEN RUNNING THIS PROGRAM ON A FASTER PROCESSOR THE DELAY PARAMETER SHOULD BE ADJUSTED PROPORTIONALLY HIGHER THAN THE FOLLOWING DEFAULTED VALUES:

2450	:TIME FOR	50 BAUD
1560	:TIME FOR	75 BAUD
1120	:TIME FOR	110 BAUD
0750	:TIME FOR	134 BAUD
0660	:TIME FOR	150 BAUD
0330	:TIME FOR	300 BAUD
0150	:TIME FOR	600 BAUD
0060	:TIME FOR	1200 BAUD
0040	:TIME FOR	1800 BAUD
0030	:TIME FOR	2000 BAUD
0020	:TIME FOR	2400 BAUD
0010	:TIME FOR	3600 BAUD
0001	:TIME FOR	4800 BAUD
0001	:TIME FOR	7200 BAUD
0001	:TIME FOR	9600 BAUD
0001	:TIME FOR	19.2 KBAUD

229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275

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 SWITCHES 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.  
F: 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 HAS 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 CAN'T BE SINGLED OUT. THIS SWITCH IS DESIGNED TO PROVIDE AN AID FOR A TRAINED TROUBLE-SHOOTER TO SAMPLE VARIOUS SIGNALS ON THE MODULE AND IS NOT MEANT TO BE USED AS A GENERAL USER CONTROL SWITCH.

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 RECOMMENDED THAT THIS SWITCH ONLY BE USED 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. (SEE SEC. 4.1.1)

276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325

#### 4.1.3 SWITCH REGISTER PRIORITIES

##### 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'). IF AN '\*' IS PRINTED IN FRONT OF THE TEST NO. ON AN ERROR REPORT (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 THE PROGRAM USER IS TECHNICALLY TRAINED TO ELECTRONICALLY ISOLATE SIGNAL PROBLEMS ON THE DZ11 MODULE. IF SW09 IS NOT ENABELED; AND THERE IS A \*HARD\* ERROR (CONSTANT); SW08 IS BEST.
2. FOR INTERMITTENT ERRORS EITHER START THE PROGRAM WITH SW01 AND SW02 SET WHICH WILL ALLOW THE USER TO LOCK ON A SELECTED TEST, OR ELSE SET SW14 AS AN ERROR IS BEING TYPED OUT ON THE TERMINAL. SW14 WILL CONTINUE TO LOOP ON THAT TEST REGARDLESS OF WHETHER AN ERROR OCCURS.
3. SW 14 LOOP ON CURRENT TEST.

#### 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 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.



326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364

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.

'MAINTENANCE MODE

[EXTERNAL <H325>-EIA ONLY (E)]  
[INTERNAL <DZCSR03=1> (I)]  
[STAGGERED <H3271>-EIA ONLY (S)]  
[STAGGERED <H3190>-20MA ONLY (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 AN H325 TEST CONNECTOR.

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  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402

'# 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 PERCENT OF THE TEST. BAUD RATE CHOICES ARE:  
'00' ( 50 BAUD), '01' ( 75 BAUD), '02' ( 110 BAUD), '03' ( 134 BAUD),  
'04' ( 150 BAUD), '05' ( 300 BAUD), '06' ( 600 BAUD), '07' (1200 BAUD),  
'10' (1800 BAUD), '11' (2000 BAUD), '12' (2400 BAUD), '13' (3600 BAUD),  
'14' (4800 BAUD), '15' (7200 BAUD), '16' (9600 BAUD), '17' (19.2 KBAUD)  
LOW DEFAULT BAUD RATES ARE NOT SUGGESTED SINCE THEY LENGTHEN THE TIME TO COMPLETE A PROGRAM PASS DRAMATICALLY.

\*\*\*\*\*

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.

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  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455

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 003.

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.



456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505

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 002. 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. BAUD RATE CHOICES ARE: 50, 75, 110, 135, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600. THE PROGRAM WILL THEN 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 THE LINE NO. IS REQUESTED ABOVE. 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: TYPE ANY PRINTABLE CHARACTER ON THE CONSOLE TERMINAL (NOT THE DZ11 TERMINAL). THE PROGRAM WILL AGAIN TYPE 'LINE: '' AND WAIT FOR A RESPONSE.

506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
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  
545  
546  
547  
548

5.4 PROGRAM AND/OR OPERATOR ACTION

THE VARIETY OF PROGRAM CONTROL SWITCHES PROVIDED IN THIS DIAGNOSTIC PACKAGE IS DESIGNED TO PROVIDE THE USER WITH A WIDE RANGE OF TROUBLE-SHOOTING TECHNIQUES. BEFORE THE USER ATTEMPTS TO RUN THIS DIAGNOSTIC HE SHOULD BECOME FAMILIAR WITH THE USE OF THESE CONTROL SWITCHES AND THEIR RESTRICTIONS. (SEE SEC. 4.1, 4.1.1, 4.1.2, 4.1.3)

WHEN THE PROGRAM DETECTS AN ERROR THE TEST NUMBER AND PC WILL BE TYPED OUT AND POSSIBLY AN ERROR MESSAGE (DEPENDING ON THE PARTICULAR ERROR). IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT THEN LOOK IN THE PROGRAM LISTING FOR THAT TEST NUMBER AND THEN NOTE THE PC OF THE ERROR REPORT. THE REASON FOR THE ERROR REPORT WILL BECOME CLEARER WHEN READING THE COMMENTS IN THE PROGRAM LISTING.

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 4.1.2  
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 COMPLETELY ISOLATE PROBLEMS.

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

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=1 (INHIBIT ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION. THE ACTUAL EXECUTION TIME DEPENDS GREATLY ON THE PDP11 CPU CONFIGURATION. AN 11/40 WITH CORE MEMORY WILL TAKE AROUND 100 SECONDS TO EXECUTE A PASS WITH NO ITERATIONS AND ABOUT 400 SECONDS TO EXECUTE A FULLY ITERATED PASS. ANY OTHER PDP11 CPU TYPE WILL EXECUTE A PASS IN TIME PROPORTIONAL TO THE EXECUTION SPEED OF THE CPU 'S MEMORY IN RELATION TO THAT OF AN 11/40.

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 CZDZA-F CSR: 160010 VEC: 300 PASSES: 000001 ERRORS:

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



582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609

8.4 KEY LOCATIONS

SLPADR (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.

STSTNM (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) 1304/00000000100000 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 SEQUENTIALLY. 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) 1300/000000000011111 MEANS THAT DZ11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DZACTV) 1300/000000000010001 MEANS THAT DZ11 NO. 00,04 WILL BE TESTED.

SBASE (1310)

CONTAINS THE RECEIVER CSR OF THE CURRENT DZ11 UNDER TEST.

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  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654

8.4A MORE ON THAT 'STATUS TABLE' (1500-2000)

'MAP OF DZ11 STATUS'	
1500	160100
1502	000300
1504	000005
1506	000377
1510	017470
1512	000000

THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 16 DZ11'S IN THE SYSTEM (THESE WILL FOLLOW UNDER THIS TABLE). EXPLANATION:

1500	160100	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 AN M7819, IF BIT15=1 MODULE SHOULD BE AN M7814.
1506	000377	THIS IS THE BINARY REPRESENTATION OF WHAT LINES ARE TO BE TESTED.
1510	017470	THIS IS THE PARAMETER LOCATION USED IN MOST OF THE TESTS. IT INDICATES 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.
		THIS LOCATION IS USED TO LOAD THE DZ11 LINE PARAMETER REGISTER FOR EACH LINE. THE MEANING OF THE BITS SET IN THIS LOCATION IS THE SAME AS THE FUNCTION OF THE RELATED BITS IN THE DEVICE LINE PARAMETER REGISTER.
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 !N) TO SUIT THE SPECIFIC CONFIGURATION.

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  
699  
700  
701  
702  
703  
704  
705  
706

8.5 \*\*\* METHOD OF AUTO SIZING \*\*\*

8.5.1 FINDING THE CONTROL STATUS REGISTER.

THE PROGRAM WILL START AT ADDRESS 160000 AND START 'REFERENCING' 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 D211 (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 D211. 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 D211; SOMETHING IS WRONG AND AUTO SIZING SHOULD NOT BE DONE. AFTER IDENTIFYING A D211 THE PROGRAM THEN ATTEMPTS TO SET ALL DTR BITS IN DEVICE REGISTER 4. IF ANY DTR BITS DID SET THE MODULE IS ASSUMED TO BE AN EIA MODULE (M7819) OTHERWISE THE STATUS MAP ENTRY IS SET FOR 20MA (M7814).

8.5.2 FINDING THE VECTOR

THE VECTOR AREA (ADDRESS 300-776) IS FILLED WITH THE INSTRUCTION IOT AND '+2' (NEXT ADDRESS). BIT14 AND BIT5 (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 D211) 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 D211 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 PERCENT OF THE PARAMETER SETUP WAS DONE BY THE PROGRAM, AND 5 PERCENT BY YOU.

THEREFORE:

- 1) BUS PRIORITY IS SET TO LEVEL5.
- 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".

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



707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762

9.0 RUNNING THE DZ11 DIAGNOSTIC UNDER APT

9.1.1 THE APT INTERFACE

CZDZA HAS BEEN REDESIGNED TO BE COMPATIBLE WITH THE AP -  
AUTOMATED PRODUCT TEST SYSTEM. IT CAN BE RUN AS A STANDALONE  
DIAGNOSTIC OR IN EITHER OF THE APT MODES. CERTAIN VARIABLES  
IN THE ORIGINAL APT MODULE WERE REASSIGNED TO THE AREAS SET  
ASIDE FOR APT INTERFACING. THESE NEW VARIABLES GENERALLY  
BEGIN WITH A DOLLAR SIGN (\$), E.G., \$DEVM, \$BASE.

9.1.2 SETTING UP THE DIAGNOSTIC USING APT

THE DIAGNOSTIC USES SEVERAL VARIABLES IN THE REGION SUBTITLED  
'APT MAILBOX-ETABLE'. THESE VARIABLES ARE:

\$SWREG - USED IF A SOFTWARE SWITCH REGISTER IS DESIRED WHILE  
UNDER APT

\$VECT1 - USED TO SPECIFY THE INTERRUPT LEVEL AND THE FIRST  
VECTOR ADDRESS

\$BASE - USED TO INDICATE BOTTOM ADDRESS OF DZ11 UNDER TEST

\$DEVM - A BIT MAP REPRESENTING WHICH DZ11'S WILL BE TESTED

\$CDW1 - USED TO INDICATE WHICH LINES TO RUN ON ALL DZ11'S

\$DDW0 - EACH OF THE \$DDW WORDS DESCRIBES THE PARAMETERS  
(LPR) FOR A PARTICULAR DZ11, GOING UP TO 16 DZ11'S

9.1.3 RUNNING UNDER APT

THE USER SHOULD BE FAMILIAR WITH THE APT SYSTEM. THE APT  
TIMING PARAMETERS FOR THE DZ11 DIAGNOSTIC WERE BASED ON AN  
11/40 PROCESSOR. IT MAY BE NECESSARY TO ADD A FEW MORE  
SECONDS IF THE DIAGNOSTIC IS OUT ON AN 11/05 PROCESSOR.

ALL OF THE VARIABLES MENTIONED IN SECTION 9.1.2 SHOULD BE SET  
UP PRIOR TO RUNNING THE DIAGNOSTIC UNDER APT.

NOTE

BE SURE \$BASE POINTS TO THE FIRST DZ11 BEFORE RUNNING

BASED ON THESE VALUES, THE DIAGNOSTIC WILL SET UP THE STATUS

CZDZA-FO  
CZDZAF.P11

MACY11 30A(1052) 26-NOV-79 11:06 PAGE 19  
26-NOV-79 11:03

F 2

SEQ 0018

763

TABLE. THE USER IS THEN FREE TO MGNITOR UNDER APT AS NORMAL.

764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819

10.0 CHANGE HISTORY

NOTE: HISTORY STARTS WITH REV. F0

REV CHANGE DESCRIPTION

---

F0: ALTER TRANSMITTER INTERRUPT SERVICE ROUTINE  
TO ALLOW MORE TIME FOR THE TCR BIT TO CLEAR BEFORE  
LOWERING THE BUS PRIORITY TO ENABLE DZ11 INTERRUPTS.

%

.TITLE CZDZA-F0

;\*COPYRIGHT (C) 1976,1979

;\*DIGITAL EQUIPMENT CORP.

;\*MAYNARD, MASS. 01754

.\*

.\*

.\*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC

.\*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.

.\*

\$TN=1

.; STARTING PROCEDURE

.; LOAD PROGRAM

.; LOAD ADDRESS 000200

.; PRESS START

.; PROGRAM WILL TYPE 'CZDZA-F0/<200>/CZDZAF0 DZ11 LN ASYNC MUX TSTS ''

.; 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

DSWR= 177570 ;;HARDWARE SWITCH REGISTER

DDISP= 177570 ;;HARDWARE DISPLAY REGISTER

.\*GENERAL PURPOSE REGISTER DEFINITIONS

000001

001120

000011

000012

000015

000200

177776

177774

177772

177570

177570



```
820      000000      R0=      %0          ;;GENERAL REGISTER
821      000001      R1=      %1          ;;GENERAL REGISTER
822      000002      R2=      %2          ;;GENERAL REGISTER
823      000003      R3=      %3          ;;GENERAL REGISTER
824      000004      R4=      %4          ;;GENERAL REGISTER
825      000005      R5=      %5          ;;GENERAL REGISTER
826      000006      R6=      %6          ;;GENERAL REGISTER
827      000007      R7=      %7          ;;GENERAL REGISTER
828      000006      SP=      %6          ;;STACK POINTER
829      000007      PC=      %7          ;;PROGRAM COUNTER
830
831      ;*PRIORITY LEVEL DEFINITIONS
832      000000      PR0=      0          ;;PRIORITY LEVEL 0
833      000040      PR1=      40         ;;PRIORITY LEVEL 1
834      000100      PR2=      100        ;;PRIORITY LEVEL 2
835      000140      PR3=      140        ;;PRIORITY LEVEL 3
836      000200      PR4=      200        ;;PRIORITY LEVEL 4
837      000240      PR5=      240        ;;PRIORITY LEVEL 5
838      000300      PR6=      300        ;;PRIORITY LEVEL 6
839      000340      PR7=      340        ;;PRIORITY LEVEL 7
840
841      ;*'SW'ITCH REGISTER' SWITCH DEFINITIONS
842      100000      SW15=     100000
843      040000      SW14=     40000
844      020000      SW13=     20000
845      010000      SW12=     10000
846      004000      SW11=     4000
847      002000      SW10=     2000
848      001000      SW09=     1000
849      000400      SW08=     400
850      000200      SW07=     200
851      000100      SW06=     100
852      000040      SW05=     40
853      000020      SW04=     20
854      000010      SW03=     10
855      000004      SW02=     4
856      000002      SW01=     2
857      000001      SW00=     1
858      .EQUIV     SW09,SW9
859      .EQUIV     SW08,SW8
860      .EQUIV     SW07,SW7
861      .EQUIV     SW06,SW6
862      .EQUIV     SW05,SW5
863      .EQUIV     SW04,SW4
864      .EQUIV     SW03,SW3
865      .EQUIV     SW02,SW2
866      .EQUIV     SW01,SW1
867      .EQUIV     SW00,SW0
868
869      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
870      100000      BIT15=    100000
871      040000      BIT14=    40000
872      020000      BIT13=    20000
873      010000      BIT12=    10000
874      004000      BIT11=    4000
875      002000      BIT10=    2000
```

```
876      001000      BIT09= 1000
877      000400      BIT08= 400
878      000200      BIT07= 200
879      000100      BIT06= 100
880      000040      BIT05= 40
881      000020      BIT04= 20
882      000010      BIT03= 10
883      000004      BIT02= 4
884      000002      BIT01= 2
885      000001      BIT00= 1
886      .EQUIV BIT09,BIT9
887      .EQUIV BIT08,BIT8
888      .EQUIV BIT07,BIT7
889      .EQUIV BIT06,BIT6
890      .EQUIV BIT05,BIT5
891      .EQUIV BIT04,BIT4
892      .EQUIV BIT03,BIT3
893      .EQUIV BIT02,BIT2
894      .EQUIV BIT01,BIT1
895      .EQUIV BIT00,BIT0
896
897      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
898      000004      ERRVEC= 4          ;;TIME OUT AND OTHER ERRORS
899      000010      RESVEC= 10         ;;RESERVED AND ILLEGAL INSTRUCTIONS
900      000014      TBIVVEC=14        ;;'T' BIT
901      000014      TRTVEC= 14        ;;TRACE TRAP
902      000014      BPTVEC= 14        ;;BREAKPOINT TRAP (BPT)
903      000020      IOTVEC= 20        ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
904      000024      PWRVEC= 24        ;;POWER FAIL
905      000030      EMTVEC= 30        ;;EMULATOR TRAP (EMT) **ERROR**
906      000034      TRAPVEC=34        ;;'TRAP' TRAP
907      000060      TKVEC= 60         ;;TTY KEYBOARD VECTOR
908      000064      TPVEC= 64         ;;TTY PRINTER VECTOR
909      000240      PIRQVEC=240       ;;PROGRAM INTERRUPT REQUEST VECTOR
910
911
912      ;INSTRUCTION DEFINITIONS
913      ;-----
914
915      005746      PUSH1SP=5746      ;DECREMENT PROCESSOR STACK 1 WORD
916      005726      POP1SP=5726      ;INCREMENT PROCESSOR STACK 1 WORD
917      010046      PUSHRO=10046     ;SAVE R0 ON STACK
918      012600      POPRO=12600      ;RESTORE R0 FROM STACK
919      024646      PUSH2SP=24646    ;DECREMENT STACK TWICE
920      022626      POP2SP=22626     ;INCREMENT STACK TWICE
921
922      ;DZ11 CONTROL AND STATUS REGISTER DEFINITIONS
923      ;(DZCSR) BIT DEFINITIONS
924      ;-----
925
926      000010      MAINT - BIT3      ;MAINTENANCE MODE ENABLE
927      000020      DCLR=BIT4        ;DEVICE CLEAR
928      000040      MSENAB=BIT5      ;MASTER SCAN ENABLE
929      000100      RIE=BIT6         ;RECEIVER INTERRUPT ENABLE
930      000200      RDONE=BIT7       ;RECEIVER DONE
931      010000      SILOEN= BIT12     ;SILO ALARM ENABLE
```

```

932      020000      SILOAL = BIT13      ;SILO ALARM
933      040000      TIE=BIT14       ;TRANSMITTER INTERRUPT ENABLE
934      100000      TRDY=BIT15      ;TRANSMITTER READY
935
936              ;DZCSR WORD DEFINITIONS
937              -----
938      000000      TL0=0           ;TRANSMIT LINE 0
939      000400      TL1=BIT8       ;TRANSMIT LINE 1
940      001000      TL2=BIT9       ;TRANSMIT LINE 2
941      001400      TL3=BIT9!BIT8  ;TRANSMIT LINE 3
942      002000      TL4=BIT10      ;TRANSMIT LINE 4
943      002400      TL5=BIT10!BIT8 ;TRANSMIT LINE 5
944      003000      TL6=BIT10!BIT9 ;TRANSMIT LINE 6
945      003400      TL7=BIT10!BIT9!BIT8 ;TRANSMIT LINE 7
946
947
948              ;DZRBUF BIT DEFINITIONS
949              -----
950
951      010000      PARER=BIT12     ;PARITY ERROR
952      020000      FRMERR=BIT13   ;FRAME ERROR
953      040000      OVRRUN=BIT14   ;OVERRUN ERROR
954      100000      DVALID=BIT15   ;DATA VALID
955
956              ;DZRBUF WORD DEFINITIONS
957              -----
958
959      000000      RL0=0           ;RECEIVER LINE 0
960      000400      RL1=BIT8       ;RECEIVER LINE 1
961      001000      RL2=BIT9       ;RECEIVER LINE 2
962      001400      RL3=BIT9!BIT8  ;RECEIVER LINE 3
963      002000      RL4=BIT10      ;RECEIVER LINE 4
964      002400      RL5=BIT10!BIT8 ;RECEIVER LINE 5
965      003000      RL6=BIT10!BIT9 ;RECEIVER LINE 6
966      003400      RL7=BIT10!BIT9!BIT8 ;RECEIVER LINE 7
967
968              ;DZLPR WORD DEFINITIONS
969              -----
970
971      000000      LP0=0           ;LINE PARAMETER 0
972      000001      LP1=BIT0        ;LINE PARAMETER 1
973      000002      LP2=BIT1        ;LINE PARAMETER 2
974      000003      LP3=BIT1!BIT0   ;LINE PARAMETER 3
975      000004      LP4=BIT2        ;LINE PARAMETER 4
976      000005      LP5=BIT2!BIT0   ;LINE PARAMETER 5
977      000006      LP6=BIT2!BIT1   ;LINE PARAMETER 6
978      000007      LP7=BIT2!BIT1!BIT0 ;LINE PARAMETER 7
979
980      000000      FIVE=0          ;FIVE BITS/CHAR,1 STOP BIT
981      000010      SIX=BIT3        ;SIX BITS/CHAR,1 STOP BIT
982      000020      SEVEN=BIT4      ;SEVEN BITS/CHAR,1 STOP BIT
983      000030      EIGHT=BIT4!BIT3 ;EIGHT BITS/CHAR,1 STOP BIT
984      000040      FIVES=BIT5      ;FIVE BITS/CHAR,2 STOP BITS
985      000050      SIXS=BIT5!BIT3  ;SIX BITS/CHAR,2 STOP BITS
986      000060      SEVENS=BIT5!BIT4 ;SEVEN BITS/CHAR, 2 STOP BITS
987      000070      EIGHTS=BIT5!BIT4!BIT3 ;EIGHT BITS/CHAR, 2 STOP BITS

```

988			
989	000100	PARITY=BIT6	;PARITY ENABLED
990	000200	ODDPAR=BIT7	;ODD PARITY ENABLED
991	000000	ONESTOP=0	;ONE STOP BIT ENABLED
992	000040	TWOSTOP=BIT5	;TWO STOP BITS ENABLED
993	000000	EVEPAR=0	;EVEN PARITY ENABLED
994	010000	RCVON=BIT12	;ENABLE RECEIVER (RECEIVER ON)
995			
996	000000	S50=0	;SPEED 50 BAUD
997	000400	S75=BIT8	;SPEED 75 BAUD
998	001000	S110=BIT9	;SPEED 110 BAUD
999	001400	S134=BIT9!BIT8	;SPEED 134.5 BAUD
1000	002000	S150=BIT10	;SPEED 150 BAUD
1001	002400	S300=BIT10!BIT8	;SPEED 300 BAUD
1002	003000	S600=BIT10!BIT9	;SPEED 600 BAUD
1003	003400	S1200=BIT10!BIT9!BIT8	;SPEED 1200 BAUD
1004	004000	S1800=BIT11	;SPEED 1800 BAUD
1005	004400	S2000=BIT11!BIT8	;SPEED 2000 BAUD
1006	005000	S2400=BIT11!BIT9	;SPEED 2400 BAUD
1007	005400	S3600=BIT11!BIT9!BIT8	;SPEED 3600 BAUD
1008	006000	S4800=BIT11!BIT10	;SPEED 4800 BAUD
1009	006400	S7200=BIT11!BIT10!BIT8	;SPEED 7200 BAUD
1010	007000	S9600=BIT11!BIT10!BIT9	;SPEED 9600 BAUD
1011	007400	S19200=BIT11!BIT10!BIT9!BIT8	;SPEED 19200 BAUD
1012			

;DZTCR BIT DEFINITIONS

1013			
1014			
1015	000001	TCR0=BIT0	;TCR0
1016	000002	TCR1=BIT1	;TCR1
1017	000004	TCR2=BIT2	;TCR2
1018	000010	TCR3=BIT3	;TCR3
1019	000020	TCR4=BIT4	;TCR4
1020	000040	TCR5=BIT5	;TCR5
1021	000100	TCR6=BIT6	;TCR6
1022	000200	TCR7=BIT7	;TCR7
1023	000400	DTR0=BIT8	;DTR0
1024	001000	DTR1=BIT9	;DTR1
1025	002000	DTR2=BIT10	;DTR2
1026	004000	DTR3=BIT11	;DTR3
1027	010000	DTR4=BIT12	;DTR4
1028	020000	DTR5=BIT13	;DTR5
1029	040000	DTR6=BIT14	;DTR6
1030	100000	DTR7=BIT15	;DTR7
1031			

;DZMSR BIT DEFINITIONS

1032			
1033			
1034	000001	RING0=BIT0	;RING INDICATED ON LINE 0
1035	000002	RING1=BIT1	;RING INDICATED ON LINE 1
1036	000004	RING2=BIT2	;RING INDICATED ON LINE 2
1037	000010	RING3=BIT3	;RING INDICATED ON LINE 3
1038	000020	RING4=BIT4	;RING INDICATED ON LINE 4
1039	000040	RING5=BIT5	;RING INDICATED ON LINE 5
1040	000100	RING6=BIT6	;RING INDICATED ON LINE 6
1041	000200	RING7=BIT7	;RING INDICATED ON LINE 7
1042	000400	CO0=BIT8	;CARRIER PRESENT ON LINE 0
1043	001000	CO1=BIT9	;CARRIER PRESENT ON LINE 1

CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 25

GENERAL DEFINITIONS AND EQUIVALENCES

SEQ 0024

1044	002000	C02=BIT10	:CARRIER PRESENT ON LINE 2
1045	004000	C03=BIT11	:CARRIER PRESENT ON LINE 3
1046	010000	C04=BIT12	:CARRIER PRESENT ON LINE 4
1047	020000	C05=BIT13	:CARRIER PRESENT ON LINE 5
1048	040000	C06=BIT14	:CARRIER PRESENT ON LINE 6
1049	100000	C07=BIT15	:CARRIER PRESENT ON LINE 7

:DZTDR BIT DEFINITIONS

:-----

1054	000400	BRK0=BIT8	:BREAK FOR LINE 0
1055	001000	BRK1=BIT9	:BREAK FOR LINE 1
1056	002000	BRK2=BIT10	:BREAK FOR LINE 2
1057	004000	BRK3=BIT11	:BREAK FOR LINE 3
1058	010000	BRK4=BIT12	:BREAK FOR LINE 4
1059	020000	BRK5=BIT13	:BREAK FOR LINE 5
1060	040000	BRK6=BIT14	:BREAK FOR LINE 6
1061	100000	BRK7=BIT15	:BREAK FOR LINE 7

:TABLE OF LOOP AROUND FUNCTIONS (H325)

```

:-----
: I ^
: V ^
: REC TRANS
: DATA DATA
:-----
: I ^
: V ^
: CO RTS
:-----
: I ^
: V ^
: RING DTR
:

```

1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081



CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 26

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

SEQ 0025

```

1082 ;:*****
1083 ;-----
1084 ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
1085 ;THE STANDARD 'TRAP CATCHER' IS PLACED
1086 ;BETWEEN ADDRESS 0 TO ADDRESS 776.
1087 ;IT LOOKS LIKE 'PC+2 HALT'.
1088 ;-----
1089 ;:*****
1090
1091      000000      .=0
1092      ;STANDARD INTERRUPT VECTORS
1093      ;-----
1094
1095      000010      .=10
1096 000010 010766      SET,PS      ;FAKE 'MTPS' INSTRUCTION TRAP
1097 000012 000340      PR7          ;MAKE SURE PS IS PRIORITY 7
1098
1099      000020      .=20
1100 000020 004772      .SCOPE      ;SCOPE LOOP HANDLER
1101 000022 000340      PR7          ;HANDLE AT PRIORITY 7
1102 000024 007646      $PWRDN     ;POWER FAIL HANDLER
1103 000026 000340      340        ;SERVICE AT PRIORITY LEVEL 7
1104 000030 006736      $ERROR     ;ERROR HANDLER
1105 000032 000340      340        ;SERVICE AT PRIORITY LEVEL 7
1106 000034 006630      .TRPSRV    ;GENERAL HANDLER DISPATCH SERVICE
1107 000036 000340      340        ;SERVICE AT PRIORITY LEVEL 7
1108      .SBTTL ACT11 HOOKS
1109
1110 ;:*****
1111 ;HOOKS REQUIRED BY ACT11
1112      000040      $SVPC=.      ;SAVE PC
1113      000046      .=46
1114 000046 004726      $ENDAD     ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1115      000052      .=52
1116 000052 000000      .WORD 0    ;;2)SET LOC.52 TO ZERO
1117      000040      .=$SVPC     ;; RESTORE PC
1118
1119      000174      .=174
1120 000174 000000      DISPREG:0   ;SOFTWARE DISPLAY REGISTER FOR SWITCHLESS 11S
1121 000176 000000      SWREG: 0    ;SOFTARE SWITCH REGISTER FOR SWITCHLESS 11S
1122      000200      .=200
1123 000200 000137 002150      JMP .START ;GO TO START OF PROGRAM
1124      000210      .=210
1125 000210 000137 023770      JMP XSTART ;GOTO CABLE TEST/ECHO TEST
1126
1127
1128      001000      .=1000
1129 001000 005200 055103 055104 MTITLE: .ASCIZ <200><12>/CZDZA-F0/<200>/CZDZAF0 DZ11 LN ASYNC MUX TSTS /<200>
(2)

```

1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185

001120  
001120 000000  
001122 000  
001123 000  
001124 000000  
001126 000000  
001130 000000  
001132 000000  
001134 000  
001135 001  
001136 000000  
001140 000000  
001142 000000  
001144 000000  
001146 000000  
001150 000000  
001152 000000  
001154 000  
001155 000  
001156 000000  
001160 177570  
001162 177570  
001164 177560  
001166 177562  
001170 177564  
001172 177566  
001174 000  
001175 002  
001176 012  
001177 000  
001200 000000  
001202 000000  
001204 000000  
001206 000000  
001210 000000  
001212 000000  
001214 000000  
001216 000000  
001220 000000  
001222 000000  
001224 000000  
001226 000000  
001230 077  
001231 015  
001232 000012

.SBTTL COMMON TAGS

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

SCMTAG: .=1120

:::START OF COMMON TAGS  
\$STNM: .WORD 0 ::CONTAINS THE TEST NUMBER  
\$ERFLG: .BYTE 0 ::CONTAINS ERROR FLAG  
\$ICNT: .WORD 0 ::CONTAINS SUBTEST ITERATION COUNT  
\$LPADR: .WORD 0 ::CONTAINS SCOPE LOOP ADDRESS  
\$LPERR: .WORD 0 ::CONTAINS SCOPE RETURN FOR ERRORS  
\$ERTTL: .WORD 0 ::CONTAINS TOTAL ERRORS DETECTED  
\$ITEMB: .BYTE 0 ::CONTAINS ITEM CONTROL BYTE  
\$ERMAX: .BYTE 1 ::CONTAINS MAX. ERRORS PER TEST  
\$ERRPC: .WORD 0 ::CONTAINS PC OF LAST ERROR INSTRUCTION  
\$GDADR: .WORD 0 ::CONTAINS ADDRESS OF 'GOOD' DATA  
\$BDADR: .WORD 0 ::CONTAINS ADDRESS OF 'BAD' DATA  
\$GDDAT: .WORD 0 ::CONTAINS 'GOOD' DATA  
\$BDDAT: .WORD 0 ::CONTAINS 'BAD' DATA  
          .WORD 0 ::RESERVED--NOT TO BE USED  
\$AUTOB: .BYTE 0 ::AUTOMATIC MODE INDICATOR  
\$INTAG: .BYTE 0 ::INTERRUPT MODE INDICATOR  
          .WORD 0  
SWR: .WORD DSWR ::ADDRESS OF SWITCH REGISTER  
DISPLAY: .WORD DDISP ::ADDRESS OF DISPLAY REGISTER  
\$TKS: 177560 ::TTY KBD STATUS  
\$TKB: 177562 ::TTY KBD BUFFER  
\$TPS: 177564 ::TTY PRINTER STATUS REG. ADDRESS  
\$TPB: 177566 ::TTY PRINTER BUFFER REG. ADDRESS  
\$NULL: .BYTE 0 ::CONTAINS NULL CHARACTER FOR FILLS  
\$FILLS: .BYTE 2 ::CONTAINS # OF FILLER CHARACTERS REQUIRED  
\$FILLC: .BYTE 12 ::INSERT FILL CHARS. AFTER A 'LINE FEED'  
\$STPFLG: .BYTE 0 ::'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)  
\$REGAD: .WORD 0 ::CONTAINS THE ADDRESS FROM  
          WHICH (\$REGO) WAS OBTAINED  
\$REG0: .WORD 0 ::CONTAINS ((\$REGAD)+0)  
\$REG1: .WORD 0 ::CONTAINS ((\$REGAD)+2)  
\$REG2: .WORD 0 ::CONTAINS ((\$REGAD)+4)  
\$REG3: .WORD 0 ::CONTAINS ((\$REGAD)+6)  
\$REG4: .WORD 0 ::CONTAINS ((\$REGAD)+10)  
\$REG5: .WORD 0 ::CONTAINS ((\$REGAD)+12)  
\$TMP0: .WORD 0 ::USER DEFINED  
\$TMP1: .WORD 0 ::USER DEFINED  
\$TMP2: .WORD 0 ::USER DEFINED  
\$TMP3: .WORD 0 ::USER DEFINED  
\$TIMES: 0 ::MAX. NUMBER OF ITERATIONS  
\$QUES: .ASCII /?/ ::QUESTION MARK  
\$CRLF: .ASCII <15> ::CARRIAGE RETURN  
\$LF: .ASCII <12> ::LINE FEED

::\*\*\*\*\*  
.SBTTL APT MAILBOX-ETABLE

1186			*****		
1187			.EVEN		
1188	001234		SMAIL:	APT MAILBOX	
1189	001234	000000	SMSGTY: .WORD	AMSGTY	MESSAGE TYPE CODE
1190	001236	000000	SFATAL: .WORD	AFATAL	FATAL ERROR NUMBER
1191	001240	000000	STESTN: .WORD	ATESTN:	TEST NUMBER
1192	001242	000000	SPASS: .WORD	APASS	PASS COUNT
1193	001244	000000	SDEVCT: .WORD	ADEVCT	DEVICE COUNT
1194	001246	000000	SUNIT: .WORD	AUNIT	I/O UNIT NUMBER
1195	001250	000000	SMSGAD: .WORD	AMSGAD	MESSAGE ADDRESS
1196	001252	000000	SMSGLG: .WORD	AMSGLG	MESSAGE LENGTH
1197	001254		SETABLE:	APT ENVIRONMENT TABLE	
1198	001254	000	SENV: .BYTE	AENV	ENVIRONMENT BYTE
1199	001255	000	SEVM: .BYTE	AENVM	ENVIRONMENT MODE BITS
1200	001256	000000	SSWREG: .WORD	ASWREG	APT SWITCH REGISTER
1201	001260	000000	SUSWR: .WORD	AUSWR	USER SWITCHES
1202	001262	000000	SCPUOP: .WORD	ACPUOP	CPU TYPE, OPTIONS
1203			.*	BITS 15-11-CPU TYPE	
1204			.*	11/04=01,11/05=02,11/20=03,11/40=04,11/45 05	
1205			.*	11/70=06,PDQ=07,Q=10	
1206			.*	BIT 10=REAL TIME CLOCK	
1207			.*	BIT 9=FLOATING POINT PROCESSOR	
1208			.*	BIT 8=MEMORY MANAGEMENT	
1209	001264	000	SMAMS1: .BYTE	AMAMS1	HIGH ADDRESS, M.S. BYTE
1210	001265	000	SMTYP1: .BYTE	AMTYP1	MEM. TYPE, BLK #1
1211			.*	MEM. TYPE BYTE -- (HIGH BYTE)	
1212			.*	900 NSEC CORE=001	
1213			.*	300 NSEC BIPOLAR=002	
1214			.*	500 NSEC MOS=003	
1215	001266	000000	SMADR1: .WORD	AMADR1	HIGH ADDRESS, BLK#1
1216			.*	MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF 'TYPE' ABOVE	
1217	001270	000	SMAMS2: .BYTE	AMAMS2	HIGH ADDRESS, M.S. BYTE
1218	001271	000	SMTYP2: .BYTE	AMTYP2	MEM. TYPE, BLK#2
1219	001272	000000	SMADR2: .WORD	AMADR2	MEM. LAST ADDRESS, BLK#2
1220	001274	000	SMAMS3: .BYTE	AMAMS3	HIGH ADDRESS, M.S. BYTE
1221	001275	000	SMTYP3: .BYTE	AMTYP3	MEM. TYPE, BLK#3
1222	001276	000000	SMADR3: .WORD	AMADR3	MEM. LAST ADDRESS, BLK#3
1223	001300	000	SMAMS4: .BYTE	AMAMS4	HIGH ADDRESS, M.S. BYTE
1224	001301	000	SMTYP4: .BYTE	AMTYP4	MEM. TYPE, BLK#4
1225	001302	000000	SMADR4: .WORD	AMADR4	MEM. LAST ADDRESS, BLK#4
1226	001304	000000	SVECT1: .WORD	AVECT1	INTERRUPT VECTOR#1, BUS PRIORITY#1
1227	001306	000000	SVECT2: .WORD	AVECT2	INTERRUPT VECTOR#2, BUS PRIORITY#2
1228	001310	160010	SBASE: .WORD	ABASE	BASE ADDRESS OF EQUIPMENT UNDER TEST
1229	001312	000000	SDEVN: .WORD	ADEVN	DEVICE MAP
1230	001314	000000	SCDW1: .WORD	ACDW1	CONTROLLER DESCRIPTION WORD#1
1231	001316	000000	SCDW2: .WORD	ACDW2	CONTROLLER DESCRIPTION WORD#2
1232	001320	000000	SDDW0: .WORD	ADDW0	DEVICE DESCRIPTOR WORD#0
1233	001322	000000	SDDW1: .WORD	ADDW1	DEVICE DESCRIPTOR WORD#1
1234	001324	000000	SDDW2: .WORD	ADDW2	DEVICE DESCRIPTOR WORD#2
1235	001326	000000	SDDW3: .WORD	ADDW3	DEVICE DESCRIPTOR WORD#3
1236	001330	000000	SDDW4: .WORD	ADDW4	DEVICE DESCRIPTOR WORD#4
1237	001332	000000	SDDW5: .WORD	ADDW5	DEVICE DESCRIPTOR WORD#5
1238	001334	000000	SDDW6: .WORD	ADDW6	DEVICE DESCRIPTOR WORD#6
1239	001336	000000	SDDW7: .WORD	ADDW7	DEVICE DESCRIPTOR WORD#7
1240	001340	000000	SDDW8: .WORD	ADDW8	DEVICE DESCRIPTOR WORD#8
1241	001342	000000	SDDW9: .WORD	ADDW9	DEVICE DESCRIPTOR WORD#9

CZDZA-FO  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 29  
APT MAILBOX-ETABLE

SEG 0028

1242	001344	000000	\$DDW10:	.WORD	ADDW10	::DEVICE	DESCRIPTOR	WORD#10
1243	001346	000000	\$DDW11:	.WORD	ADDW11	::DEVICE	DESCRIPTOR	WORD#11
1244	001350	000000	\$DDW12:	.WORD	ADDW12	::DEVICE	DESCRIPTOR	WORD#12
1245	001352	000000	\$DDW13:	.WORD	ADDW13	::DEVICE	DESCRIPTOR	WORD#13
1246	001354	000000	\$DDW14:	.WORD	ADDW14	::DEVICE	DESCRIPTOR	WORD#14
1247	001356	000000	\$DDW15:	.WORD	ADDW15	::DEVICE	DESCRIPTOR	WORD#15
1248								
1249								
1250	001360		\$cTEND:					
1251								

1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266 001360  
1267  
1268  
1269  
1270  
1271 001360 000000  
1272 001362 000000  
1273  
1274  
1275  
1276  
1277 001364 000377  
1278 001366 017470  
1279 001370 WWWW  
1280 001372 000000  
1281 001374 000000  
1282 001376 000000  
1283 001400 000000  
1284 001402 000000  
1285 001404 000001  
1286 001406 000001  
1287 001410 000001  
1288 001411 001  
1289  
1290 001412 001500

.SBTTL ERROR POINTER TABLE

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

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

\$ERRTB:

;;PROGRAM CONTROL PARAMETERS  
;-----

NEXT: 0 ;ADDRESS OF NEXT TEST TO BE EXECUTED  
LOCK: 0 ;ADDRESS FOR LOCK ON CURRENT DATA

;;PROGRAM VARIABLES  
;-----

LINE: 377 ;DEFAULT ALL EIGHT LINES RUNNING  
PAR: 17470 ;PARAMETERS: 8 BITS/CHAR,2 STOP BITS,19200 BAUD,NO PARIT  
MODE: 0 ;DEFAULT MAINTENANCE MODE  
SAVLIN: 0 ;LINE NUMBER  
XMTLIN: 0 ;TRANSMISSION LINE NUMBER  
XMTCNT: 0 ;COUNT OF WORDS IN A TRANSMISSION PATTERN  
REGIST: 0 ;DEVICE ADDRESS STORAGE LOCATION  
SAVPC: 0 ;PROGRAM COUNTER STORAGE  
DZACTV: .BLKW 1 ;\*DZ11'S SELECTED ACTIVE.  
RUN: 1 ;\*POINTER ONE PAST RUNNING DEVICE.  
DZNUM: .BLKB 1 ;\*OCTAL NUMBER OF DZ11'S.  
SAVNUM: .BYTE 1 ;\*WORKABLE NUMBER.  
.EVEN  
ACTIVE: DZ.MAP ;TABLE POINTER.

```

1291
1292
1293
1294
1295 001414 000
1296 001415 000
1297 001416 000
1298 001417 000
1299 001420 000
1300 001422
1301
1302 001422 000000
1303 001424 000000
1304 001426 000000
1305 001430 000000
1306 001432 000000
1307 001434 000000
1308 001436 000000
1309 001440 000000
1310 001442 000000
1311 001444 000000
1312 001446 000000
1313 001450 000000
1314 001452 000000
1315 001454 000000
1316 001456 000000
1317 001460 000000
1318 001462
1319
1320
1321
1322
1323
1324 001462
1325 000024 000200
1326 000024 000200
1327 000044 000044
1328 000044 001462
1329 001462
1330
1331
1332
1333
1334 001462
1335 001462 000000
1336 001464 001234
1337 001466 000132
1338 001470 000137
1339 001472 000137
1340 001474 000052
1341
1342
1343
1344 001500
1345 001500
1346

```

```

:PROGRAM CONTROL FLAGS
:-----
EIAFLG: .BYTE 0 ;0=EIA 100000=20MA
INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
HDRFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG FOR HEADER MAP
MNTFLG: .BYTE 0 ;MAINTENANCE BIT SET FLAG
DONFLG: .BYTE 0 ;TRANSMISSION COMPLETION FLAG
.EVEN
:DATA VARIABLES
TDO: .WORD 0
TD1: .WORD 0
TD2: .WORD 0
TD3: .WORD 0
TD4: .WORD 0
TD5: .WORD 0
TD6: .WORD 0
TD7: .WORD 0
TRC: .WORD 0
TR1: .WORD 0
TR2: .WORD 0
TR3: .WORD 0
TR4: .WORD 0
TR5: .WORD 0
TR6: .WORD 0
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.
$APTHDR ;POINT TO APT HEADER BLOCK
--.$X ;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 ^-15)
$STMT: .WORD 90 ;RUN TIM OF LONGEST TEST
$PASTM: .WORD 95 ;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .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:

```



1347	001500	000001	DZCR0:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 0
1348	001502	000001	DZVC0:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 0
1349	001504	000001	DZLV0:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1350	001506	000001	LINE0:	.BLKW	1	:ALL LINES SELECTED
1351	001510	000001	PAR0:	.BLKW	1	:PARAMETERS
1352	001512	000001	MANT0:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1353						
1354	001514	000001	DZCR1:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 1
1355	001516	000001	DZVC1:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 1
1356	001520	000001	DZLV1:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1357	001522	000001	LINE1:	.BLKW	1	:ALL LINES SELECTED
1358	001524	000001	PAR1:	.BLKW	1	:PARAMETERS
1359	001526	000001	MANT1:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1360						
1361	001530	000001	DZCR2:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 2
1362	001532	000001	DZVC2:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 2
1363	001534	000001	DZLV2:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1364	001536	000001	LINE2:	.BLKW	1	:ALL LINES SELECTED
1365	001540	000001	PAR2:	.BLKW	1	:PARAMETERS
1366	001542	000001	MANT2:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1367						
1368	001544	000001	DZCR3:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 3
1369	001546	000001	DZVC3:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 3
1370	001550	000001	DZLV3:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1371	001552	000001	LINE3:	.BLKW	1	:ALL LINES SELECTED
1372	001554	000001	PAR3:	.BLKW	1	:PARAMETERS
1373	001556	000001	MANT3:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1374						
1375	001560	000001	DZCR4:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 4
1376	001562	000001	DZVC4:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 4
1377	001564	000001	DZLV4:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1378	001566	000001	LINE4:	.BLKW	1	:ALL LINES SELECTED
1379	001570	000001	PAR4:	.BLKW	1	:PARAMETERS
1380	001572	000001	MANT4:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1381						
1382	001574	000001	DZCR5:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 5
1383	001576	000001	DZVC5:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 5
1384	001600	000001	DZLV5:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1385	001602	000001	LINE5:	.BLKW	1	:ALL LINES SELECTED
1386	001604	000001	PAR5:	.BLKW	1	:PARAMETERS
1387	001606	000001	MANT5:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1388						
1389	001610	000001	DZCR6:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 6
1390	001612	000001	DZVC6:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 6
1391	001614	000001	DZLV6:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1392	001616	000001	LINE6:	.BLKW	1	:ALL LINES SELECTED
1393	001620	000001	PAR6:	.BLKW	1	:PARAMETERS
1394	001622	000001	MANT6:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1395						
1396	001624	000001	DZCR7:	.BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 7
1397	001626	000001	DZVC7:	.BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 7
1398	001630	000001	DZLV7:	.BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1399	001632	000001	LINE7:	.BLKW	1	:ALL LINES SELECTED
1400	001634	000001	PAR7:	.BLKW	1	:PARAMETERS
1401	001636	000001	MANT7:	.BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1402						

1403	001640	000001	DZCR10: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 10
1404	001642	000001	DZVC10: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 10
1405	001644	000001	DZLV10: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1406	001646	000001	LINE10: .BLKW	1	:ALL LINES SELECTED
1407	001650	000001	PAR10: .BLKW	1	:PARAMETERS
1408	001652	000001	MANT10: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1409					
1410	001654	000001	DZCR11: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 11
1411	001656	000001	DZVC11: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 11
1412	001660	000001	DZLV11: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1413	001662	000001	LINE11: .BLKW	1	:ALL LINES SELECTED
1414	001664	000001	PAR11: .BLKW	1	:PARAMETERS
1415	001666	000001	MANT11: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1416					
1417	001670	000001	DZCR12: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 12
1418	001672	000001	DZVC12: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 12
1419	001674	000001	DZLV12: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1420	001676	000001	LINE12: .BLKW	1	:ALL LINES SELECTED
1421	001700	000001	PAR12: .BLKW	1	:PARAMETERS
1422	001702	000001	MANT12: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1423					
1424	001704	000001	DZCR13: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 13
1425	001706	000001	DZVC13: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 13
1426	001710	000001	DZLV13: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1427	001712	000001	LINE13: .BLKW	1	:ALL LINES SELECTED
1428	001714	000001	PAR13: .BLKW	1	:PARAMETERS
1429	001716	000001	MANT13: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1430					
1431	001720	000001	DZCR14: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 14
1432	001722	000001	DZVC14: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 14
1433	001724	000001	DZLV14: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1434	001726	000001	LINE14: .BLKW	1	:ALL LINES SELECTED
1435	001730	000001	PAR14: .BLKW	1	:PARAMETERS
1436	001732	000001	MANT14: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1437					
1438	001734	000001	DZCR15: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 15
1439	001736	000001	DZVC15: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 15
1440	001740	000001	DZLV15: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1441	001742	000001	LINE15: .BLKW	1	:ALL LINES SELECTED
1442	001744	000001	PAR15: .BLKW	1	:PARAMETERS
1443	001746	000001	MANT15: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1444					
1445	001750	000001	DZCR16: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 16
1446	001752	000001	DZVC16: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 16
1447	001754	000001	DZLV16: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1448	001756	000001	LINE16: .BLKW	1	:ALL LINES SELECTED
1449	001760	000001	PAR16: .BLKW	1	:PARAMETERS
1450	001762	000001	MANT16: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1451					
1452	001764	000001	DZCR17: .BLKW	1	:CONTROL STATUS REGISTER FOR DZ11 NUMBER 17
1453	001766	000001	DZVC17: .BLKW	1	:RECEIVER AND BASE VECTOR FOR DZ11 NUMBER 17
1454	001770	000001	DZLV17: .BLKW	1	:PRIORITY LEVEL AND EIA FLAG SELECTOR
1455	001772	000001	LINE17: .BLKW	1	:ALL LINES SELECTED
1456	001774	000001	PAR17: .BLKW	1	:PARAMETERS
1457	001776	000001	MANT17: .BLKW	1	:MAINTENANCE MODE FOR THIS DEVICE
1458					

CZDZA-FO MACY11 30A(1052) 26-NOV-79 11:06 PAGE 34  
CZDZAF.P11 26-NOV-79 11:03 APT PARAMETER BLOCK

SEQ 0033

1459 002000 177777

DZ.END: 177777

1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501

002002 104400  
002002 006724  
002004 104401  
002006 005236  
002006 104402  
002010 005262  
002010 104403  
002012 006030  
002012 104404  
002012 006134  
002014 104405  
002014 006154  
002016 104406  
002016 010500  
002020 104407  
002020 006354  
002022 104410  
002022 006414  
002024 104411  
002024 006446  
002026 104412  
002026 006452  
002030 104413  
002030 006652  
002032 104414  
002032 006704  
002034 104415  
002034 025516  
002036 104416  
002036 025712  
002040 104417  
002040 006672

```

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

```

```

1502                                     :DZ11 VECTOR AND REGISTER INDIRECT POINTERS
1503                                     :WORKING AREA
1504
1505 002042 160040      DZCSR: 160040  :R/W
1506 002044 160041      HDZCSR: 160041  :R/W
1507 002046 160042      DZRBUF: 160042  :READ ONLY
1508 002050 160043      HDZRBUF: 160043  :READ ONLY
1509 002052 160042      DZLPR: 160042  :WRITE ONLY
1510 002054 160043      HDZLPR: 160043  :WRITE ONLY
1511 002056 160044      DZTCR: 160044  :R/W
1512 002060 160045      HDZTCR: 160045  :R/W
1513 002062 160046      DZMSR: 160046  :READ ONLY
1514 002064 160047      HDZMSR: 160047  :READ ONLY
1515 002066 160046      DZTDR: 160046  :WRITE ONLY
1516 002070 160047      HDZTDR: 160047  :WRITE ONLY
1517                                     :DEFAULT DZ VECTORS
1518 002072 000300      DZRIV: 300    :REC INTR VECTOR
1519 002074 000302      DZRIS: 302    :REC INTR STATUS
1520 002076 000304      DZTIV: 304    :XMIT INTR VECTOR
1521 002100 000306      DZTIS: 306    :XMIT INTR STATUS
1522
1523

```

1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547

002102  
002102 000000  
002104 000000  
002106 000000  
002110 000000  
002112 000000  
002114 000000  
002116 000000  
002120 000000  
002122 000000  
002124 000000  
002126 000000  
002130 000000  
002132 000000  
002134 000000  
002136 000000  
002140 000000  
002142 000000  
002144 000000  
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



```

1548
1549
1550
1551
1552
1553
1554
1555
1556 002150          .START:
1557 002150 000005    RESET          ;CLEAR THE WORLD. START NEW ENVIRONMENT
1558 002152 012706 001120    MOV          #STACK,SP      ;SET UP STACK
1559 002156 106427 000340    MTPS        #PR7           ;LOCK OUT INTERRUPTS
1560 002162 012737 007646 000024    MOV          #SPWRDN,@#24   ;SET UP POWER FAIL VECTOR
1561 002170 113737 001410 001411    MOVVB       DZNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
1562 002176 005037 001242          CLR          $PASS         ;CLEAR PASS COUNT
1563 002202 105037 001123          CLRB        $ERFLG        ;CLEAR ERROR FLAG
1564 002206 012737 001500 001412    MOV          #DZ.MAP,ACTIVE ;GET MAP POINTER.
1565 002214 012737 000001 001406    MOV          #1,RUN        ;POINT POINTER TO FIRST DEVICE.
1566 002222 005037 001132          CLR          $ERTTL       ;CLEAR ERROR COUNT
1567 002226 005037 001136          CLR          $ERRPC       ;CLEAR LAST ERROR POINTER
1568 002232 005037 001122          CLR          $STSTM       ;SET UP FOR TEST 1
1569 002236 012737 002150 001126    MOV          #.START,$LPADR ;SET UP FOR POWER FAIL BEFORE
1570
1571
1572 002244 013746 000006          ;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY
1573 002250 013746 000004          MOV          6,-(SP)        ;SAVE BUS ERROR PS
1574 002254 012737 002274 000004          MOV          4,-(SP)        ;SAVE BUS ERROR PC
1575 002262 022777 177777 176670          MOV          #20$,4        ;SET UP TO TRAP TO THIS ROUTINE
1576 002270 001402          CMP          #-1,@SWR      ;CAN 177570 BE REFERENCED?
1577 002272 000407          BEQ         22$           ;IF SO AND IT IS -1, TREAT LIKE SWITCHLESS
1578 002274 022626          BR          21$           ;IF YES, SKIP AROUND THE SETUP
1579 002276 012737 000176 001160 20$: POP2SP      ;REMOVE THE TRAP FROM THE STACK
1580 002304 012737 000174 001162 22$: MOV          #SWREG,SWR    ;IF NO, TRAP COMES HERE. POINT TO SOFTWARE SWR
1581 002312 012637 000004          MOV          #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REGISTER
1582 002316 012637 000006          MOV          (SP)+,4        ;RESTORE THE BUS ERROR VECTOR
1583 002322 105737 001415          MOV          (SP)+,6
1584 002326 001010          TSTB       INIFLG        ;TITLE ALREADY PRINTED?
1585 002330 023727 000042 004726          BNE         29$           ;BRANCH IF YES
1586 002336 001402          CMP          @#42,$SENDAD  ;RUNNING UNDER ACT?
1587 002340 104402 001000          BEQ         31$           ;IF YES DON'T PRINT TITLE
1588 002344 105337 001415          TYPE       ,MTITLE       ;PRINT THE DIAGNOSTIC'S TITLE
1589 002350 105737 001255          DECB       INIFLG        ;SET THE ONCE ONLY FLAG
1590 002354 100006          TSTB       $ENVM         ;DETERMINE WHETHER APT SIZING SHOULD BE DONE
1591 002356 004737 011440          BPL        30$           ;IF NOT, GO CHECK FOR AUTO-SIZING
1592 002362 105037 001416          JSR        PC,SETAPT     ;OTHERWISE, GO DO APT SIZING FROM ETABLE
1593 002366 000137 004270          CLRB       HDRFLG       ;MAKE SURE STATUS TABLE IS PRINTED
1594 002372 032777 000001 176560 30$: JMP         16$           ;GO PRINT DZ STATUS TABLE
1595 002400 001011          BIT        #SW00,@SWR    ;RESELECT ?
1596 002402 122737 000377 001415          BNE        32$           ;IF YES, GO SET UP THE INFORMATION
1597 002410 001003          CMPB      #377,INIFLG   ;ON 1ST START; MUST ANSWER QUESTION
1598 002412 105777 176542          BNE        .+10         ;IF NOT ANSWERING QUESTIONS
1599 002416 100402          TSTB      @SWR          ;ARE U AUTO SIZING?
1600 002420 000137 003114          BMI        32$           ;NO AUTO SIZE! NO SW00=1 ON 1ST START.
1601 002424 012700 001500          JMP        73$           ;IF NO, SKIP THE INTERROGATION
1602 002430 105037 001416          MOV        #DZ.MAP,RO    ;POINT TO THE BEGINNING OF THE MAP TABLE
1603 002434 005020          CLRB      HDRFLG       ;MAKE SURE A MAP GETS PRINTED
                                CLR        (RO)+          ;CLEAR A TABLE LOCATION

```

```

1604 002436 020027 002000      CMP      R0,#DZ.END      ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
1605 002442 001374              BNE      65$             ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE
1606 002444 105337 001415      DECB     INIFLG          ;INSURE NO AUTO SIZING IF QUESTIONS ANSWERED!
1607
1608                               ;THE FOLLOWING ARE PARAMETERS USED TO FILL IN THE MAP
1609                               ;TABLE AND SET UP THE DIAGNOSTIC.
1610
1611                               ;GET THE BASE ADDRESS OF THE DZ11'S
1612
1613 002450      33$:
1614 002450 104403      INSTR     ;CALL THE STRING INPUT ROUTINE
1615 002452 003334      66$      ;POINTER TO MESSAGE TO BE PRINTED
1616 002454 104405      PARAM     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1617 002456 160000      160000  ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1618 002460 163770      163770  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1619 002462 001500      DZCRO   ;POINTER TO MAP LOCATION TO BE FILLED
1620 002464      007     .BYTE    7      ;MASK OF INVALID BITS FOR THIS PARAMETER
1621 002465      001     .BYTE    1      ;NUMBER OF PARAMETERS TO STORE
1622 002466 013737 001500 001310  MOV      DZCRO,$BASE   ;COPY BASE ADDRESS TO ETABLE
1623
1624                               ;GET THE BASE VECTOR ADDRESS
1625
1626 002474      34$:
1627 002474 104403      INSTR     ;CALL THE STRING INPUT ROUTINE
1628 002476 003400      67$      ;POINTER TO MESSAGE TO BE PRINTED
1629 002500 104405      PARAM     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1630 002502 000300      300     ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1631 002504 000776      776     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1632 002506 001502      DZVCO   ;POINTER TO MAP LOCATION TO BE FILLED
1633 002510      003     .BYTE    3      ;MASK OF INVALID BITS FOR THIS PARAMETER
1634 002511      001     .BYTE    1      ;NUMBER OF PARAMETERS TO STORE
1635 002512 013737 001502 001304  MOV      DZVCO,$VECT1 ;COPY VECTOR TO ETABLE
1636
1637                               ;GET THE BUS REQUEST LEVEL
1638
1639 002520 104403      INSTR     ;CALL THE STRING INPUT ROUTINE
1640 002522 003441      68$      ;POINTER TO MESSAGE TO BE PRINTED
1641 002524 104405      PARAM     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1642 002526 000004      4       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1643 002530 000007      7       ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1644 002532 001504      DZLVO   ;POINTER TO MAP LOCATION TO BE FILLED
1645 002534      000     .BYTE    0      ;MASK OF INVALID BITS FOR THIS PARAMETER
1646 002535      001     .BYTE    1      ;NUMBER OF PARAMETERS TO STORE
1647 002536 113737 001504 001305  MOV&    DZLVO,$VECT1+1 ;GET BUS REQUEST LEVEL INTO ETABLE
1648 002544 106337 001305      ASLB    $VECT1+1      ;ALIGN THE BITS PROPERLY
1649 002550 106337 001305      ASLB    $VECT1+1      ;ALIGN THE BITS PROPERLY
1650 002554 106337 001305      ASLB    $VECT1+1      ;ALIGN THE BITS PROPERLY
1651 002560 106337 001305      ASLB    $VECT1+1      ;ALIGN THE BITS PROPERLY
1652 002564 106337 001305      ASLB    $VECT1+1      ;ALIGN THE BITS PROPERLY
1653
1654                               ;FIND OUT IF MODULE IS EIA OR 20 MA.
1655
1656 002570 104402 004130      TYPE     ,74$          ;PRINT EIA MESSAGE
1657 002574 005037 001220      CLR      $TMP1        ;USE $TMP1
1658 002600 105777 176360      80$:  TSTB    @$TKS      ;IS KEYBOARD DONE?
1659 002604 100375      BPL      80$          ;IF NOT, WAIT FOR IT

```

```
1660 002606 017746 176354      MOV    @STKB,-(SP)      ;IF YES, PUT CHARACETR ON STACK
1661 002612 042716 000240      BIC    #240,(SP)      ;STRIP DOWN CHARACTER
1662 002616 122726 000015      CMPB   #15,(SP)+     ;IS IT ?
1663 002622 001414              BEQ    81$            ;IF SO, GET OUT
1664 002624 014677 176342      MOV    -(SP),@STPB    ;IF NOT, PRINT CHARACTER
1665 002630 042737 100000 001504  BIC    #BIT15,DZLVO   ;CLEAR EIA FLAG
1666 002636 122726 000102      CMPB   #102,(SP)+    ;IS IT A B?
1667 002642 001356              BNE    80$            ;IF NOT, GO BACK FOR INPUT
1668 002644 052737 100000 001504  BIS    #BIT15,DZLVO   ;IF SO, SET FLAG
1669 002652 000752              BR     80$            ;GET MORE INPUT
1670 002654              81$:
1671
1672              ;GET THE MODE OF OPERATION (E,I,S)
1673
1674 002654 104403      INSTR              ;CALL THE STRING INPUT ROUTINE
1675 002656 003652      72$                ;POINTER TO THE MESSAGE TO BE PRINTED
1676 002660 104406      SETFLG             ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
1677 002662 001512      MANTO             ;THIS IS THE FLAG BEING SETUP
1678
1679              ;GET THE NUMBER OF DZ11'S RUNNING
1680
1681 002664 104403      INSTR              ;CALL THE STRING INPUT ROUTINE
1682 002666 003610      71$                ;POINTER TO MESSAGE TO BE PRINTED
1683 002670 104405      PARAM             ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1684 002672 000001      1                 ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1685 002674 000020      16.               ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1686 002676 001220      $TMP1             ;POINTER TO MAP LOCATION TO BE FILLED
1687 002700 000                .BYTE 0             ;MASK OF INVALID BITS FOR THIS PARAMETER
1688 002701 001                .BYTE 1             ;NUMBER OF PARAMETERS TO STORE
1689
1690 002702 012737 000377 001506  MOV    #377,LINE0    ;SET UP DEFAULT LINES
1691 002710 012737 017470 001510  MOV    #17470,PAR0   ;SET UP DEFAULT LPR PARAMETER
1692
1693 002716 012737 000001 006722  MOV    #1,DLYCNT     ;RECEIVER ON; 19.2 KBAUD; 2STOP BITS; 8 BIT/CHAR
1694 002724 032777 000010 176226  BIT    #SW03,@SWR    ;INITIALIZE DELAY COUNT
1695 002732 001402              BEQ    40$            ;DO YOU WANT PARAMETERS?
1696 002734 004737 003144      JSR    PC,23$        ;IF NO, SKIP THE PARAMETER CALL
1697 002740 012737 000001 001312 40$:  MOV    #1,$DEVMS    ;GET PARAMETERS
1698 002746 113737 001220 001410  MOV    $TMP1,DZNUM   ;INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
1699 002754 113737 001220 001411  MOV    $TMP1,SAVNUM  ;COPY THE NUMBER OF DEVICES
1700 002762 005337 001220      62$:  DEC    $TMP1         ;COPY A BACKUP NUMBER
1701 002766 001404              REQ    61$           ;$TMP1 CONTAINS THE COUNT OF UNINITIALIZED
1702 002770 000261              SEC                    ;SELECTED DEVICES
1703 002772 006137 001312      ROL    $DEVMS        ;SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
1704 002776 000771              BR     62$           ;POINT TO THE NEXT DEVICE
1705 003000 013737 001312 001222 61$:  MOV    $DEVMS,$TMP2  ;GO DO THIS PROCEDURE AGAIN
1706 003006 013737 001312 001404  MOV    $DEVMS,DZACTV ;# OF TIMES
1707 003014 012700 001500      MOV    #DZCR0,R0     ;COPY THE ACTIVE DEVICE PARAMETER
1708 003020 012701 001514      MOV    #DZCR1,R1     ;SET A POINTER TO THE SPECIFIED INFORMATION
1709 003024 012702 001320      MOV    #SDDW0,R2     ;POINT R1 TO THE REST OF THE MAP TABLE
1710 003030 000241              CLC                    ;POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
1711 003032 006037 001222      ROR    $TMP2         ;INITIALIZE THE 'C' BIT FOR A ROTATION
1712 003036 006237 001222      64$:  ASR    $TMP2         ;SKIF MAPPING SETUP FOR DEVICE 0- IT'S DONE
1713 003042 103404              BCS    41$           ;ISOLATE A SELECTION FLAG IN THE 'C' BIT
1714 003044 012711 177777      MOV    #-1,(R1)      ;IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
1715 003050 000137 004244      JMP    63$           ;TERMINATE THE LIST
                                ;GO TO THE NEXT BLOCK
```

```

1716 003054 012011          41$:  MOV      (R0)+,(R1)      :ADDRESS
1717 003056 062721 000010    ADD      #10,(R1)+      :POINT TO THE NEXT DZ11 ADDRESS VALUE
1718 003062 012011          MOV      (R0)+,(R1)      :VECTOR
1719 003064 062721 000010    ADD      #10,(R1)+      :POINT TO THE NEXT VECTOR VALUE
1720 003070 012021          MOV      (R0)+,(R1)+    :LEVEL
1721 003072 012021          MOV      (R0)+,(R1)+    :LINES
1722 003074 016012 177774    MOV      -4(R0),(R2)    :GET THE EIA FLAG FROM THE PRIORITY WORD
1723 003100 042712 077777    BIC      #77777,(R2)    :ISOLATE THAT FLAG
1724 003104 051022          BIS      (R0),(R2)+    :ADD PARAMETERS TO DEVICE DESCRIPTOR WORD
1725 003106 012021          MOV      (R0)+,(R1)+    :PARAMETERS
1726 003110 012021          MOV      (R0)+,(R1)+    :MAINTENANCE MODE
1727 003112 000751          BR       64$
1728 003114 032177 000010 176036 73$: BIT      #SW03,@SWR      :ASK PARAMETERS ?
1729 003122 001002          BNE     42$
1730 003124 000137 004244    JMP     63$
1731 003130 004737 003144    42$:  JSR     PC,23$
1732 003134 105337 001415    DECB   INIFLG
1733 003140 000137 004270    JMP     16$
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
1771

003144          23$:
003144 104403    INSTR      :CALL THE STRING INPUT ROUTINE
003146 003464    69$
003150 104405    PARAM     :POINTER TO MESSAGE TO BE PRINTED
003152 000001    1         :CALL THE OCTAL TO ASCII CONVERT ROUTINE
003154 000377    377      :LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003156 001506    LINE0    :HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003160          .BYTE 0    :POINTER TO MAP LOCATION TO BE FILLED
003161          .BYTE 1    :MASK OF INVALID BITS FOR THIS PARAMETER
003162 105037 001416    CLRB     HDRFLG      :NUMBER OF PARAMETERS TO STORE
                                :MAKE SURE THE CHANGES ARE PRINT.

:THIS SEGMENT CHECKS TO MAKE SURE THE LINE PARAMETER JUST ENTERED
:IS LEGITIMATE IN STAGGERED MODE OPERATION IF THAT MODE WAS SELECTED

003166 005737 001512    TST     MANTO      :IS STAGGERED THE MODE OF OPERATION?
003172 100021    BPL     26$
003174 013703 001506    MOV     LINE0,R3   :IF NOT, SKIP THIS SEGMENT
003200 006003    24$:  ROR     R3         :GET A SCRATCH COPY OF THE ACTIVE LINES
003202 103410    BCS     25$
003204 001414    BEQ     26$
003206 006203    ASR     R3         :GET A LINE SELECTION BIT(EVEN NUMBER LINE)
003210 103373    BCC     24$
003212 104402 001250    TYPE   ,SQUES     :IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
003216 104402 010423    TYPE   ,MBADLN    :IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
003222 000750    BR     23$
003224 001772    25$:  BEQ     27$
003226 006203    ASR     R3         :IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
003230 103370    BCC     27$
003232 000241    CLC
003234 000761    BR     24$
                                :IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
                                :THIS IS AN INCORRECT PARAMETER
                                :LET THE USER KNOW ABOUT IT
                                :GO GET THE CORRECT PARAMETER
                                :IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
                                :GET THE NEXT FLAG
                                :IF IT ISN'T SET, THERE'S AN ERROR
                                :INITIALIZE THE 'C' BIT FOR TESTING OF THE NEXT PAIR
                                :GO TEST THE NEXT PAIR OF FLAGS

:GET THE LINE PARAMETER REGISTER ARGUMENT

003236          26$:
003236 104403    INSTR      :CALL THE STRING INPUT ROUTINE

```

1772	003240	003540				70\$		: POINTER TO MESSAGE TO BE PRINTED
1773	003242	104405				PARAM		: CALL THE OCTAL TO ASCII CONVERT ROUTINE
1774	003244	000000				0		: LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1775	003246	000017				17		: HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1776	003250	001510				PAR0		: POINTER TO MAP LOCATION TO BE FILLED
1777	003252	000				.BYTE 0		: MASK OF INVALID BITS FOR THIS PARAMETER
1778	003253	001				.BYTE 1		: NUMBER OF PARAMETERS TO STORE
1779	003254	012702	001506			MOV #LINE0,R2		: POINT TO THE LINE SELECTION PARAMETER
1780	003260	012703	001510			MOV #PAR0,R3		: POINT TO THE CHOSEN PARAMETERS
1781	003264	011304				MOV (R3),R4		: USE BAUD RATE AS AN INDEX IN DELAY TABLE
1782	003266	006304				ASL R4		: ALIGN INDEX ON WORD BOUNDARY
1783	003270	016437	031050	006722		MOV DLYTBL(R4),DLYCNT		: SET THE DELAY COUNT FOR THIS BAUD RATE
1784	003276	000313				SWAB (R3)		: PLACE IN HIGH BYTE
1785	003300	052713	010070			BIS #10070,(R3)		: PLACE EXTRA PARAMETERS INTO LOC
1786	003304	011262	000014	28\$		MOV (R2),14(R2)		: LOAD THE LINES
1787	003310	011363	000014			MOV (R3),14(R3)		: LOAD THE PARAMETERS
1788	003314	062702	000014			ADD #14,R2		: POINT TO THE NEXT SET
1789	003320	062703	000014			ADD #14,R3		: .. OF BOTH PARAMETERS
1790	003324	020327	001774			CMP R3,#PAR17		: HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
1791	003330	001365				BNE 28\$		: IF NOT, GO LOAD SOME MORE PARAMETERS
1792	003332	000207				RTS PC		: RETURN TO CALLING BLOCK
1793	003334	030600	052123	041440	66\$:	.ASCIZ <200>/1ST CSR ADDRESS (160000:163700): /		
(1)	003400	030600	052123	053040	67\$:	.ASCIZ <200>/1ST VECTOR ADDRESS (300:770): /		
(1)	003441	200	051102	046040	68\$:	.ASCIZ <200>/BR LEVEL (4:6): /		
(1)	003464	046200	047111	051505	69\$:	.ASCIZ <200>/LINES ACTIVE BY BIT <IN OCTAL>(001:377): /		
(1)	003540	042200	043105	052501	70\$:	.ASCIZ <200>/DEFAULT BAUD RATE <IN OCTAL>(00:17): /		
(1)	003610	021600	047440	020106	71\$:	.ASCIZ <200>/# OF DZ11'S <IN OCTAL> (1:20): /		
(1)	003652	046600	044501	052116	72\$:	.ASCII <200>/MAINTENANCE MODE/		
(1)	003673	200	055440	054105		.ASCII <200>/ [EXTERNAL <H325>-EIA ONLY (E)]/		
(1)	003741	200	055440	047111		.ASCII <200>/ [INTERNAL <DZCSR03=1> (I)]/		
(1)	004007	200	055440	052123		.ASCII <200>/ [STAGGERED <H3271>-EIA ONLY (S)]: /		
(1)	004057	200	055440	052123		.ASCII <200>/ [STAGGERED <H3190>-20MA ONLY (S)]: /		
(1)	004130	052200	050131	020105	74\$:	.ASCIZ <200>/TYPE 'A' FOR EIA MODULE OR 'B' FOR 20 MA (A:B): /		
(1)	004212	042600	052116	051105	75\$:	.ASCIZ <200>/ENTER DELAY PARAMETER: /		
(1)	004244	004244				.EVEN		
(1)	004244				65\$:			
1794	004244	122737	000377	001415		CMPB #577,INIFLG		: ONLY DO AUTO SIZE ON 1ST START
1795	004252	001006				BNE 16\$		
1796	004254	032777	000200	174676		BIT #BIT7,@SWR		: BIT7=1??
1797	004262	001002				BNE 16\$		: BR IF NO AUTO SIZE
1798	004264	004737	011612			SR PC,AUTO.SIZE		: GO DO THE AUTO SIZE
1799	004270	105737	001416	16\$:		TSTB HDRFLG		: HAS THE TABLE BEEN TYPED YET?
1800	004274	001021				BNE 1\$		: IF SO, DON'T TYPE IT AGAIN
1801	004276	105337	001416			DECB HDRFLG		: INDICATE THAT THE TABLE WILL BE TYPED
1802	004302	104402	010376			TYPE ,XHEAD		: TYPE MAP HEADER
1803	004306	012700	001500			MOV #DZ.MAP,R0		: SET POINTER
1804	004312	010037	001220	5\$:		MOV R0,\$TMP1		: POINT TO THE MAP LOCATION
1805	004316	012037	001222			MOV (R0)+,\$TMP2		: SET DATA
1806	004322	022737	177777	001222		CMP #-1,\$TMP2		: END OF LIST?
1807	004330	001403				BEQ 1\$		: BR IF YES
1808	004332	104411			17\$:	CONVRT		: CALL THE OCTAL TO ASCII CONVERSION ROUTINE
1809	004334	010466				XSTAT0		: CONVERT THE DATA AT THIS ADDRESS
1810	004336	000765				BR 5\$		: GO PRINT THE NEXT PARAMETER
1811	004340	005737	000042	1\$:		TST @42		: IS PROGRAM RUNNING UNDER MONITOR
1812	004344	001026				BNE 3\$		: YES
1813	004346	032777	000100	174604		BIT #SW06,@SWR		: DESELECT SPECIFIC DEVICES??

CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 43  
PROGRAM INITIALIZATION AND START UP.

SEQ 0042

```

1814 004354 001422          BEQ      3$          ;BR IF NO.
1815 004356 104402 010317  TYPE     ,MNEW      ;TYPE THE MESSAGE.
1816 004362 005000          CLR      R0         ;ZERO DATA DISPLAY
1817 004364 000000          HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1818 004366 027737 174566 001312  CMP      @SWR,$DEV   ;IS THE NUMBER VALID?
1819 004374 101404          BLOS    2$         ;BR IF NUMBER IS OK.
1820 004376 104402 010171          TYPE     ,MERR3     ;TELL USER OF INVALID NUMBER.
1821 004402 000000          9$: HALT                    ;STOP EVERY THING.
1822 004404 000776          BR      9$         ;RESTART THE PROGRAM AGAIN.
1823 004406 017737 174546 001404  2$: MOV    @SWR,DZACTV ;GET NEW DEVICE PATTERN
1824 004414 013700 001404          MOV    DZACTV,R0   ;SHOW THE USER WHAT HE SELECTED.
1825 004420 000000          HALT                    ;CONTINUE DYNAMIC SWITCHES.
1826 004422 032777 000020 174530  3$: BIT    #SW04,@SWR ;CHECK TO SEE IF DELAY COUNT CHANGES
1827 004430 001407          BEQ    18$        ;IF NOT, GO CLEAR VECTOR AREA
1828 004432 104403          INSTR                   ;CALL THE STRING INPUT ROUTINE
1829 004434 004212          75$                    ;POINTER TO MESSAGE TO BE PRINTED
1830 004436 104405          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1831 004440 000001          1                        ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1832 004442 177777          177777                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1833 004444 006722          DLYCNT                   ;POINTER TO MAP LOCATION TO BE FILLED
1834 004446 000          .BYTE  0              ;MASK OF INVALID BITS FOR THIS PARAMETER
1835 004447 001          .BYTE  1              ;NUMBER OF PARAMETERS TO STORE
1836 004450 012700 000300 18$: MOV    #300,R0     ;PREPARE TO CLEAR THE FLOATING
1837 004454 012701 000302          MOV    #302,R1     ;VECTOR AREA. 300-776
1838 004460 010120 4$: MOV    R1,(R0)+ ;START PUTTING 'PC+2 - HALT'
1839 004462 005021          CLR    (R1)+       ;IN VECTOR AREA.
1840 004464 022021          CMP    (R0)+,(R1)+ ;POP POINTERS
1841 004466 022700 001000          CMP    #1000,R0   ;ALL DONE??
1842 004472 001372          BNE   4$          ;BR IF NO.
1843
1844          ;TEST START AND RESTART
1845          ;-----
1846
1847 004474 012706 001120 .BEGIN: MOV    #STACK,SP ;SET UP STACK
1848 004500 106427 000340 MTPS    #PR7         ;LOCK OUT INTERRUPTS
1849 004504 005737 000042 TST     @#42        ;IS PROGRAM UNDER MONITOR CONTROL
1850 004510 001015          BNE    2$         ;BR IF YES
1851 004512 032777 000004 174440 BIT     #BIT2,@SWR  ;CHECK FOR LOCK ON TEST
1852 004520 001406          BEQ    1$         ;BR IF NO LOCK DESIRED.
1853 004522 104402 010215          TYPE     ,MLOCK   ;TYPE LOCK SELECTED.
1854 004526 012737 000240 005010 MOV    #NOP,TTST   ;ADJUST SCOPE ROUTINE.
1855 004534 000403          BR     2$         ;CONTINUE ALONG.
1856 004536 013737 005232 005010 1$: MOV    BRW,TTST   ;PREPARE NORMAL SCOPE ROUTINE
1857 004544 012737 011070 001126 2$: MOV    #CYCLE,$LPADR ;START AT 'CYCLE' FIND WHICH DEVICE TO TEST
1858 004552 104402 010106          TYPE     ,MR      ;TYPE 'RUNNING'
1859 004556 000177 174344          JMP    @SLPADR    ;START TESTING

```



```

1860                                     ;END OF PASS
1861                                     ;TYPE NAME OF TEST
1862                                     ;UPDATE PASS COUNT
1863                                     ;CHECK FOR EXIT TO ACT-11
1864                                     ;RESTART TEST
1865 .SBTTL END OF PASS ROUTINE
1866
1867 ::*****
1868 ;*INCREMENT THE PASS NUMBER ($PASS)
1869 ;*IF THERES A MONITOR GO TO IT
1870 ;*IF THERE ISN'T JUMP TO CYCLE
1871
1872 004562 $EOP:
1873 004562 000004 SCOPE
1874 004564 005037 001136 CLR $ERRPC ;CLEAR LAST ERROR PC
1875 004570 105037 001123 CLR $ERFLG ;CLEAR ERROR FLAG
1876 004574 104402 010063 TYPE ,MEPASS ;TYPE END PASS
1877 004600 104402 010244 TYPE ,MCSRX ;TYPE CSR
1878 004604 104412 004742 CNVRT ,XCSR ;SHOW IT
1879 004610 104402 010252 TYPE ,MVECX ;TYPE VECTOR
1880 004614 104412 004750 CNVRT ,XVEC ;SHOW IT
1881 004620 005237 001242 INC $PASS ;RAISE PASS COUNT
1882 004624 104402 010260 TYPE ,MPASSX ;TYPE PASSES
1883 004630 104412 004756 CNVRT ,XPASS ;SHOW IT
1884 004634 005337 001242 DEC $PASS ;RESTORE PASS COUNT
1885 004640 104402 010271 TYPE ,MERRX ;TYPE ERRORS
1886 004644 104412 004764 CNVRT ,XERR ;SHOW IT
1887 004650 105337 001411 DECB SAVNUM ;ARE ALL DEVICES TESTED?
1888 004654 001030 BNE $DOAGN ;BR IF NO.
1889 004656 113737 001410 001411 MOV $DZNUM, SAVNUM ;RESTORE THE COUNT
1890 004664 005037 001226 CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
1891 004670 005237 001242 INC $PASS ;INCREMENT THE PASS NUMBER
1892 004674 042737 100000 001242 BIC #100000, $PASS ;DON'T ALLOW A NEG. NUMBER
1893 004702 005327 DEC (PC)+ ;LOOP?
1894 004704 000001 $EOPCT: .WORD 1
1895 004706 003C13 BGT $DOAGN ;:YES
1896 004710 012737 MOV (PC)+, @ (PC)+ ;:RESTORE COUNTER
1897 004712 000001 $ENDCT: .WORD 1
1898 004714 004704 $EOPCT
1899 004716 013700 000042 $GET42: MOV @#42, R0 ;:GET MONITOR ADDRESS
1900 004722 001405 BEQ $DOAGN ;:BRANCH IF NO MONITOR
1901 004724 000005 RESET ;:CLEAR THE WORLD
1902 004726 004710 $ENDAD: JSR PC, (R0) ;:GO TO MONITOR
1903 004730 000240 NOP ;:SAVE ROOM
1904 004732 000240 NOP ;:FOR
1905 004734 000240 NOP ;:ACT11
1906 004736 $DOAGN:
1907 004736 000137 JMP @ (PC)+ ;:RETURN
1908 004740 011070 $RTNAD: .WORD CYCLE
1909
1910 004742 000001 XCSR: 1
1911 004744 006 002 .BYTE 6,2
1912 004746 002042 DZCSR
1913 004750 000001 XVEC: 1
1914 004752 003 002 .BYTE 3,2
1915 004754 002072 DZRIV

```

1916 004756 000001  
 1917 004760 006 002  
 1918 004762 001242  
 1919 004764 000001  
 1920 004766 006 002  
 1921 004770 001132  
 1922  
 1923  
 1924  
 1925  
 1926  
 1927  
 1928  
 1929  
 1930  
 1931  
 1932  
 1933  
 1934  
 1935  
 1936  
 1937  
 1938 004772  
 1939 004772 004737 007360  
 1940 004776 005037 001136  
 1941 005002 022716 012376  
 1942 005006 001413  
 1943 005010 000406  
 1944 005012 105777 174146  
 1945 005016 100067  
 1946 005020 017766 174142 177776  
 1947 005026 032777 040000 174124  
 1948 005034 001060  
 1949  
 1950 005036 000416  
 1951  
 1952 005040 013746 000004  
 1953 005044 012737 005064 000004  
 1954 005052 005737 177060  
 1955 005056 012637 000004  
 1956 005062 000436  
 1957 005064 022626  
 1958 005066 012637 000004  
 1959 005072 000441  
 1960 005074  
 1961 005074 105737 001123  
 1962 005100 001404  
 1963 005102 105037 001123  
 1964 005106 005037 001226  
 1965 005112 032777 004000 174040  
 1966 005120 001011  
 1967 005122 005737 001242  
 1968 005126 001406  
 1969 005130 005237 001124  
 1970 005134 023737 001226 001124  
 1971 005142 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=IOT

$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 $XTSTR ;IF SO, DON'T LOOP ON IT
TTST: BR 1$ ;GOTO 1$ (IF LOCK SW02=1; THIS LOC =240)
        TSTB @STKS ;KEYBOARD DONE?
        BPL $OVER ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
        MOV @STKB,-2(SP) ;CLEAR DONE BIT
1$: BIT #BIT14,@SWR ;:LOOP ON PRESENT TEST?
        BNE $OVER ;:YES IF SW14=1
;*****START OF CODE FOR THE XOR TESTER*****
$XTSTR: BR 6$

        MOV @WERRVEC,-(SP) ;:IF RUNNING ON THE 'XOR' TESTER CHANGE
        MOV #5$,@WERRVEC ;:THIS INSTRUCTION TO A 'NOP' (NOP=240)
        TST @W177060 ;:SAVE THE CONTENTS OF THE ERROR VECTOR
        MOV (SP)+,@WERRVEC ;:SET FOR TIMEOUT
        BR $SVLAD ;:TIME OUT ON XOR?
5$: CMP (SP)+,(SP)+ ;:RESTORE THE ERROR VECTOR
        MOV (SP)+,@WERRVEC ;:GO TO THE NEXT TEST
        BR $OVER ;:CLEAR THE STACK AFTER A TIME OUT
6$: ;*****END OF CODE FOR THE XOR TESTER*****
2$: TSTB $ERFLG ;:HAS AN ERROR OCCURRED?
        BEQ 3$ ;:BR IF NO
4$: CLRB $ERFLG ;:ZERO THE ERROR FLAG
        CLR $TIMES ;:CLEAR THE NUMBER OF ITERATIONS TO MAKE
3$: 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

```

```

1972 005144 012737 000001 001124 1$:   MOV    #1,$ICNT      ;;REINITIALIZE THE ITERATION COUNTER
1973 005152 013737 005234 001226      MOV    $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
1974 005160 105237 001122      $SVLAD: INCB   $STNM      ;;COUNT TEST NUMBERS
1975 005164 113737 001122 001240      MOV    $STNM,$TESTN  ;;SET TEST NUMBER IN APT MAILBOX
1976 005172 011637 001126      MOV    (SP),$LPADR   ;;SAVE SCOPE LOOP ADDRESS
1977 005176 013777 001122 173756 $OVER: MOV    $STNM,@DISPLAY ;;DISPLAY TEST NUMBER
1978 005204 013716 001126      MOV    $LPADR,(SP)   ;;FUDGE RETURN ADDRESS
1979 005210 105037 001417      3$:   CLRB   MNTFLG    ;;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TEST
1980 005214 005737 001370      TST    MODE         ;;HAS THE MODE BEEN CHANGED?
1981 005220 001003      BNE    4$           ;;IF NOT INTERNAL , GO DO A TEST
1982 005222 112737 000010 001417      MOV    #MAINT,MNTFLG ;;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
1983 005230 000002      4$:   RTI           ;;GO DO THE TEST
1984 005232 000406      BRW:   406
1985 005234 000005      $MXCNT: 5           ;;MAX. NUMBER OF ITERATIONS
1986
1987      ;CHECK FOR FREEZE ON CURRENT DATA
1988      ;-----
1989
1990 005236 032777 001000 173714 .SCOPI: BIT    #SW09,@SWR   ;;IS SW09=1(SET)?
1991 005244 001405      BEQ    1$          ;;BR IF NOT SET.
1992 005246 005737 001362      TST    LOCK        ;;IS THER A TIGHT LOOP SPECIFIED?
1993 005252 001402      BEQ    1$          ;;IF NO, RETURN
1994 005254 013716 001362      MOV    LOCK,(SP)   ;;IF YES, GOTO THE ADDRESS IN LOCK.
1995 005260 000002      1$:   RTI           ;;GO BACK.
1996
1997 005262 032777 010000 173670 .TYPE: BIT    #SW12,@SWR   ;;INHIBIT ALL PRINTOUT??
1998 005270 001403      BEQ    1$          ;;IF NOT, GO TYPE
1999 005272 062716 000002      ADD    #2,(SP)     ;;SKIP OVER MESSAGE POINTER
2000 005276 000002      RTI           ;;RETURN TO WHERE PROCEDURE WAS INVOKED
2001 005300
2002      .SBTTL  TYPE ROUTINE
2003
2004      ;*****
2005      ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2006      ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2007      ;*NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
2008      ;*NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
2009      ;*NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2010      ;*
2011      ;*CALL:
2012      ;*1) USING A TRAP INSTRUCTION
2013      ;*      TYPE    ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2014      ;*OR
2015      ;*      TYPE
2016      ;*      MESADR
2017      ;*
2018
2019 005300 105737 001177      $TYPE: TSTB   $TPFLG    ;;IS THERE A TERMINAL?
2020 005304 100002      BPL    1$          ;;BR IF YES
2021 005306 000000      HALT                    ;;HALT HERE IF NO TERMINAL
2022 005310 000430      BR     3$          ;;LEAVE
2023 005312 010046      1$:   MOV    RO,-(SP)   ;;SAVE RO
2024 005314 017600 000002      MOV    @2(SP),RO    ;;GET ADDRESS OF ASCIZ STRING
2025 005320 122737 000001 001254      CMPB   #APTENV,$ENV  ;;RUNNING IN APT MODE
2026 005326 001011      BNE    62$         ;;NO,GO CHECK FOR APT CONSOLE
2027 005330 132737 000100 001255      BITB   #APTPOOL,$ENVM ;;SPOOL MESSAGE TO APT

```

```

2028 005336 001405          BEQ      62$          ::NO,GO CHECK FOR CONSOLE
2029 005340 010037 005350    MOV      R0,61$      ::SETUP MESSAGE ADDRESS FOR APT
2030 005344 004737 005570    JSR      PC,$ATY3    ::SPOOL MESSAGE TO APT
2031 005350 000000          .WORD    0           ::MESSAGE ADDRESS
2032 005352 132737 000040 001255 62$:    BITB     #APT$SUP,$ENVM ::APT CONSOLE SUPPRESSED
2033 005360 001003          BNE      60$          ::YES,SKIP TYPE OUT
2034 005362 112046          MOVB     (R0)+,-(SP)  ::PUSH CHARACTER TO BE TYPED ONTO STACK
2035 005364 001005          BNE      4$           ::BR IF IT ISN'T THE TERMINATOR
2036 005366 005726          TST     (SP)+        ::IF TERMINATOR POP IT OFF THE STACK
2037 005370 012600          60$:    MOV      (SP)+,R0  ::RESTORE R0
2038 005372 062716 000002    3$:    ADD      #2,(SP)   ::ADJUST RETURN PC
2039 005376 000002          RTI                      ::RETURN
2040 005400 122716 000011    4$:    CMPB     #HT,(SP)   ::BRANCH IF <HT>
2041 005404 001430          BEQ      8$           ::BRANCH IF NOT <CRLF>
2042 005406 122716 000200    CMPB     #CRLF,(SP)
2043 005412 001006          BNE      5$           ::POP <CR><LF> EQUIV
2044 005414 005726          TST     (SP)+        ::TYPE A CR AND LF
2045 005416 104402          TYPE
2046 005420 001231          $CRLF
2047 005422 105037 005556    CLRB     $CHARCNT    ::CLEAR CHARACTER COUNT
2048 005426 000755          BR       2$           ::GET NEXT CHARACTER
2049 005430 004737 005512    5$:    JSR      PC,$TYPEC   ::GO TYPE THIS CHARACTER
2050 005434 123726 001176    6$:    CMPB     $FILLC,(SP)+ ::IS IT TIME FOR FILLER CHARS.?
2051 005440 001350          BNE      2$           ::IF NO GO GET NEXT CHAR.
2052 005442 013746 001174    MOV      $NULL,-(SP)  ::GET # OF FILLER CHARS. NEEDED
2053                                ::AND THE NULL CHAR.
2054 005446 105366 000001    7$:    DECB     1(SP)       ::DOES A NULL NEED TO BE TYPED?
2055 005452 002770          BLT     6$           ::BR IF NO--GO POP THE NULL OFF OF STACK
2056 005454 004737 005512    JSR      PC,$TYPEC   ::GO TYPE A NULL
2057 005460 105337 005556    DECB     $CHARCNT    ::DO NOT COUNT AS A COUNT
2058 005464 000770          BR       7$           ::LOOP
2059
2060                                ;HORIZONTAL TAB PROCESSOR
2061
2062 005466 112716 000040    8$:    MOVB     #' ,(SP)   ::REPLACE TAB WITH SPACE
2063 005472 004737 005512    9$:    JSR      PC,$TYPEC   ::TYPE A SPACE
2064 005476 132737 000007 005556    C1TB     #7,$CHARCNT  ::BRANCH IF NOT AT
2065 005504 001372          BNE      9$           ::TAB STOP
2066 005506 005726          TST     (SP)+        ::POP SPACE OFF STACK
2067 005510 000724          BR       2$           ::GET NEXT CHARACTER
2068 005512 105777 173452    $TYPEC: TSTB     @STPS   ::WAIT UNTIL PRINTER IS READY
2069 005516 100375          BPL     $TYPEC
2070 005520 116677 000002 173444    MOVB     2(SP),@STPB  ::LOAD CHAR TO BE TYPED INTO DATA REG.
2071 005526 122766 000015 000002    CMPB     #CR,2(SP)   ::IS CHARACTER A CARRIAGE RETURN?
2072 005534 001003          BNE      1$           ::BRANCH IF NO
2073 005536 105037 005556    CLRB     $CHARCNT    ::YES--CLEAR CHARACTER COUNT
2074 005542 000406          BR       $TYPEX
2075 005544 122766 000012 000002 1$:    CMPB     #LF,2(SP)  ::IS CHARACTER A LINE FEED?
2076 005552 001402          BEQ     $TYPEX       ::BRANCH IF YES
2077 005554 105227          INCB     (PC)+       ::COUNT THE CHARACTER
2078 005556 000000          $CHARCNT: .WORD    0  ::CHARACTER COUNT STORAGE
2079 005560 000207          $TYPEX: RTS      PC
2080
2081                                .SBTTL  APT COMMUNICATIONS ROUTINE
2082
2083                                ;:*****

```

CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 48  
APT COMMUNICATIONS ROUTINE

SEQ 0047

```

2084 005562 112737 000001 006026 $ATY1:  MOVB  #1,$FFLG  ;;TO REPORT FATAL ERROR
2085 005570 112737 000001 006024 $ATY3:  MOVB  #1,$MFLG  ;;TO TYPE A MESSAGE
2086 005576 000403                BR      $ATYC
2087 005600 112737 000001 006026 $ATY4:  MOVB  #1,$FFLG  ;;TO ONLY REPORT FATAL ERROR
2088 005606                $ATYC:
2089 005606 010046                MOV     R0,-(SP)  ;;PUSH R0 ON STACK
2090 005610 010146                MOV     R1,-(SP)  ;;PUSH R1 ON STACK
2091 005612 105737 006024                TSTB   $MFLG     ;;SHOULD TYPE A MESSAGE?
2092 005616 001450                BEQ    5$        ;;IF NOT: BR
2093 005620 122737 000001 001254                CMPB   #APTENV,$ENV  ;;OPERATING UNDER APT?
2094 005626 001031                BNE    3$        ;;IF NOT: BR
2095 005630 132737 000100 001255                BITB   #APTSPOOL,$ENVM  ;;SHOULD SPOOL MESSAGES?
2096 005636 001425                BEQ    3$        ;;IF NOT: BR
2097 005640 017600 000004                MCV    @4(SP),R0  ;;GET MESSAGE ADDR.
2098 005644 062766 000002 000004                ADD    #2,4(SP)  ;;BUMP RETURN ADDR.
2099 005652 005737 001234                1$:    TST    $MSGTYPE  ;;SEE IF DONE W/ LAST XMISSION?
2100 005656 001375                BNE    1$        ;;IF NOT: WAIT
2101 005660 010037 001250                MOV    R0,$MSGAD  ;;PUT ADDR IN MAILBOX
2102 005664 105720                2$:    TSTB   (R0)+    ;;FIND END OF MESSAGE
2103 005666 001376                BNE    2$
2104 005670 163700 001250                SUB    $MSGAD,R0  ;;SUB START OF MESSAGE
2105 005674 006200                ASR    R0        ;;GET MESSAGE LNGTH IN WORDS
2106 005676 010037 001252                MOV    R0,$MSGGLT  ;;PUT LENGTH IN MAILBOX
2107 005702 012737 000004 001234                MOV    #4,$MSGTYPE  ;;TELL APT TO TAKE MSG.
2108 005710 000413                BR     5$
2109 005712 017637 000004 005736 3$:    MOV    @4(SP),4$  ;;PUT MSG ADDR IN JSR LINKAGE
2110 005720 062766 000002 000004                ADD    #2,4(SP)  ;;BUMP RETURN ADDRESS
2111 005726 013746 177776                MOV    177776,-(SP)  ;;PUSH 177776 ON STACK
2112 005732 004737 005300                JSR    PC,$TYPE  ;;CALL TYPE MACRO
2113 005736 000000                4$:    .WORD  0
2114 005740                5$:
2115 005740 105737 006026                10$:   TSTB   $FFLG     ;;SHOULD REPORT FATAL ERROR?
2116 005744 001416                BEQ    12$      ;;IF NOT: BR
2117 005746 005737 001254                TST    $ENV     ;;RUNNING UNDER APT?
2118 005752 001413                BEQ    12$      ;;IF NOT: BR
2119 005754 005737 001234                11$:   TST    $MSGTYPE  ;;FINISHED LAST MESSAGE?
2120 005760 001375                BNE    11$     ;;IF NOT: WAIT
2121 005762 017637 000004 001236                MOV    @4(SP),$FATAL  ;;GET ERROR #
2122 005770 062766 000002 000004                ADD    #2,4(SP)  ;;BUMP RETURN ADDR.
2123 005776 005237 001234                INC    $MSGTYPE  ;;TELL APT TO TAKE ERROR
2124 006002 105037 006026                12$:   CLRB   $FFLG     ;;CLEAR FATAL FLAG
2125 006006 105037 006025                CLRB   $LFLG     ;;CLEAR LOG FLAG
2126 006012 105037 006024                CLRB   $MFLG     ;;CLEAR MESSAGE FLAG
2127 006016 012601                MOV    (SP)+,R1  ;;POP STACK INTO R1
2128 006020 012600                MOV    (SP)+,R0  ;;POP STACK INTO R0
2129 006022 000207                RTS     PC       ;;RETURN
2130 006024 000                $MFLG: .BYTE  0  ;;MESSG. FLAG
2131 006025 000                $LFLG: .BYTE  0  ;;LOG FLAG
2132 006026 000                $FFLG: .BYTE  0  ;;FATAL FLAG
2133 006030                .EVEN
2134 000200                APTSIZE=200
2135 000001                APTENV=001
2136 000100                APTSPOOL=100
2137 000040                APTCSUP=040
2138
2139                ;STRING INPUT ROUTINE

```

```
2140
2141
2142 006030 010346
2143 006032 010446
2144 006034 017637 000004 006052
2145 006042 062766 000002 000004
2146 006050 104402
2147 006052 000000
2148 006054 012704 010620
2149 006050 012703 000007
2150 006050 105777 173074
2151 006070 100375
2152 006072 117714 173070
2153 006076 142714 000200
2154 006102 122427 000015
2155 006106 001417
2156 006110 105777 173054
2157 006114 100375
2158 006116 017777 173044 173046
2159 006124 005303
2160 006126 001356
2161 006130 012604
2162 006132 012603
2163 006134 010346
2164 006136 010446
2165 006140 104402 001230
2166 006144 000741
2167 006146 012604
2168 006150 012603
2169 006152 000002
2170
2171
2172
2173
2174 006154 010546
2175 006156 010446
2176 006160 016605 000004
2177 006164 012537 006344
2178 006170 012537 006346
2179 006174 012537 006350
2180 006200 112537 006352
2181 006204 11 37 006353
2182 006210 010566 000004
2183 006214 005005
2184 006216 012704 010620
2185 006222 122714 000015
2186 006226 001420
2187 006230 121427 000060
2188 006234 002415
2189 006236 121427 000067
2190 006242 003012
2191 006244 142714 000060
2192 006250 152405
2193 006252 122714 000015
2194 006256 001406
2195 006260 006305

:-----
.INSTR: MOV R3,-(SP) ;SAVE R3 ON STACK
MOV R4,-(SP) ;SAVE R4 ON STACK
MOV @4(SP),.MSG ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
ADD #2,4(SP) ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
.INST1: TYPE ;PRINT THE MESSAGE
.MSG: 0 ;MESSAGE IS POINTED TO FROM HERE
MOV #INBUF,R4 ;POINT R4 TO THE INPUT BUFFER
MOV #7,R3 ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
1$: TSTB @STKS ;HAS A CHARACTER BEEN RECEIVED?
BPL 1$ ;IF NO, KEEP WAITING FOR IT
MOV @STKB,(R4) ;IF YES, SAVE IT IN THE INPUT BUFFER
BICB #200,(R4) ;KEEP ONLY THE 7-BIT ASCII INFORMATION
CMPB (R4)+,#15 ;IS THIS CHARACTER A LINE FEED?
BEQ INSTR2 ;IF SO, TERMINATE THE INPUT SEQUENCE
2$: TSTB @STPS ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
BPL 2$ ;IF WE CAN'T, WAIT UNTIL WE CAN
MOV @STKB,@STPB ;ECHO THE CHARACTER BACK
DEC R3 ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
BNE 1$ ;IF WE DON'T HAVE 7, GO GET SOME MORE
MOV (SP)+,R4 ;IF WE HAVE 7, RESTORE R4
MOV (SP)+,R3 ;RESTORE R3
.INSTE: MOV R3,-(SP) ;SAVE R3 ON THE STACK
MOV R4,-(SP) ;SAVE R4 ON THE STACK
TYPE ,SQUES ;PRINT A QUESTION MARK... WHAT'S GOING ON?
BR .INST1 ;GO PRINT THE MESSAGE AGAIN
INSTR2: MOV (SP)+,R4 ;RESTORE R4
MOV (SP)+,R3 ;RESTORE R3
RTI ;RETURN TO THE MAIN PROCEDURE

:CONVERT ASCII STRING TO OCTAL
:-----
.PARAM: MOV R5,-(SP) ;SAVE R5 ON THE STACK
MOV R4,-(SP) ;SAVE R4 ON THE STACK
MOV 4(SP),R5 ;GET THE SETUP INFORMATION POINTER
MOV (R5)+,LOLIM ;SET THE LOW LIMIT FOR THE INPUT
MOV (R5)+,HILIM ;SET THE HIGH LIMIT FOR THE INPUT
MOV (R5)+,DEVADR ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
MOV (R5)+,LOBITS ;GET THE MASK OF THE INCORRECT BITS
MOV (R5)+,ADRCNT ;GET THE COUNT OF ITEMS TO BE STORED
PARAM1: MOV R5,4(SP) ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
CLR R5 ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
MOV #INBUF,R4 ;POINT TO THE INPUT BUFFER
CMPB #15,(R4) ;IS THIS CHARACTER A CARRIAGE RETURN?
BEQ PARERR ;IF SO, PRINT THE MESSAGE AGAIN
1$: CMPB (R4),#60 ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
BLT PARERR ;IF SO, GO PRINT THE MESSAGE AGAIN
CMPB (R4),#67 ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
BGT PARERR ;IF SO, GO PRINT THE MESSAGE AGAIN
BICB #60,(R4) ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
BISB (R4)+,R5 ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
CMPB #15,(R4) ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
BEQ LIMITS ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
ASL R5 ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
```



```

2196 006262 006305          ASL      R5          ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
2197 006264 006305          ASL      R5          ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
2198                                ;NEXT THREE BITS
2199 006266 000760          BR        1$          ;GO GET THE NEXT CHARACTER
2200 006270 104404          PARERR: INSTER       ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
2201 006272 000750          BR        PARAM1     ;TRY GETTING THE PARAMETERS AGAIN
2202
2203                                ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2204                                ;-----
2205
2206 006274 020537 006346          LIMITS: CMP      R5,HILIM      ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
2207 006300 101373          BHI      PARERR       ;IF YES, GO PRINT THE MESSAGE AGAIN
2208 006302 020537 006344          CMP      R5,LOLIM     ;IS THE RESULT LOWER THAN ALLOWED?
2209 006306 103770          BLO      PARERR       ;IF YES, GO PRINT THE MESSAGE AGAIN
2210 006310 133705 006352          BITB    LOBITS,R5    ;ARE ANY INCORRECT BITS SET IN THE RESULT?
2211 006314 001365          BNE      PARERR       ;IF SO, GO PRINT THE MESSAGE AGAIN
2212
2213                                ;STORE NUMBER AT SPECIFIED ADDRESS
2214
2215 006316 013704 006350          1$:      MOV      DEVADR,R4    ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
2216 006322 010524          MOV      R5,(R4)+    ;STORE THE RESULT
2217 006324 062705 000002          ADD      #2,R5       ;CALCULATE THE NEXT DATUM
2218 006330 105337 006353          DECB    ADRCNT       ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
2219 006334 001372          BNE      1$          ;IF NOT, GO STORE THE NEXT DATUM
2220 006336 012604          MOV      (SP)+,R4    ;RESTORE R4
2221 006340 012605          MOV      (SP)+,R5    ;RESTORE R5
2222 006342 000002          RTI                    ;RETURN TO THE MAIN PROGRAM
2223
2224 006344 000000          LOLIM:   0            ;LOWEST ACCEPTABLE VALUE
2225 006346 000000          HILIM:   0            ;HIGHEST ACCEPTABLE
2226 006350 000000          DEVADR:  0            ;LOCATION WHERE RESULT WILL BE STORED
2227 006352 000          LOBITS:  .BYTE 0      ;INCORRECT BITS MASK
2228 006353 000          ADRCNT:  .BYTE 0      ;COUNT OF ITEMS TO BE STORED
2229
2230                                ;SAVE PC OF TEST THAT FAILED AND R0-R5
2231                                ;-----
2232
2233 006354 016637 000004 001402 .SAV05: MOV      4(SP),SAVPC    ;SAVE R7 (PC)
2234
2235                                ;SAVE R0-R5
2236
2237 006362 010537 001214          SV05:   MOV      R5,$REG5    ;SAVE R5
2238 006366 010437 001212          MOV      R4,$REG4    ;SAVE R4
2239 006372 010337 001210          MOV      R3,$REG3    ;SAVE R3
2240 006376 010237 001206          MOV      R2,$REG2    ;SAVE R2
2241 006402 010137 001204          MOV      R1,$REG1    ;SAVE R1
2242 006406 010037 001202          MOV      R0,$REG0    ;SAVE R0
2243 006412 000002          RTI                    ;LEAVE.
2244
2245                                ;RESTORE R0-R5
2246
2247 006414 013700 001202          .RES05: MOV      $REG0,R0    ;RESTORE R0
2248 006420 013701 001204          MOV      $REG1,R1    ;RESTORE R1
2249 006424 013702 001206          MOV      $REG2,R2    ;RESTORE R2
2250 006430 013703 001210          MOV      $REG3,R3    ;RESTORE R3
2251 006434 013704 001212          MOV      $REG4,R4    ;RESTORE R4

```

```

2252 00644C 013705 001214      MOV    $REG5,R5      ;RESTORE R5
2253 006444 000002              RTI                  ;LEAVE
2254
2255                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2256                          ;-----
2257
2258 006446 104402 001231      .CONVR: TYPE        , $CRLF          ;PRINT A CARRIAGE RETURN
2259 006452 010046      .CNVRT: MOV    R0,-(SP)      ;SAVE R0
2260 006454 010146      MOV    R1,-(SP)      ;SAVE R1
2261 006456 010346      MOV    R3,-(SP)      ;SAVE R3
2262 006460 010446      MOV    R4,-(SP)      ;SAVE R4
2263 006462 010546      MOV    R5,-(SP)      ;SAVE R5
2264 006464 017601 000012      MOV    @12(SP),R1     ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
2265 006470 062766 000002 000012      ADD    #2,12(SP)      ;POINT TO WHERE MAIN PROGRAM WILL RESUME
2266 006476 012137 006622      MOV    (R1)+,WRDCNT   ;GET NUMBER OF WORDS TO BE PRINTED
2267 006502 112105      1$:   MOV    (R1)+,R5   ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
2268 006504 112100      MOV    (R1)+,R0      ;GET THE NUMBER OF SPACES TO PRINT
2269 006506 013104      MOV    @ (R1)+,R4     ;COPY THE WORD TO BE CONVERTED
2270 006510 110537 006624      MOV    R5,CHRCNT     ;COPY THE CHARACTER COUNT
2271 006514 010403      3$:   MOV    R4,R3        ;COPY THE ARGUMENT WORD AGAIN
2272 006516 042703 177770      BIC    #^C<7>,R3     ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
2273 006522 062703 000060      ADD    #060,R3       ;MAKE AN ASCII CHARACTER OUT OF THEM
2274 006526 110346      MOV    R3,-(SP)      ;SAVE THAT CHARACTER
2275 006530 006004      ROR    R4             ;MOVE THE NEXT THREE BITS INTO PLACE
2276 006532 006204      ASR    R4             ;MOVE THEM AGAIN
2277 006534 006204      ASR    R4             ;AND FINALLY A THIRD TIME
2278 006536 005305      DEC    R5             ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
2279                          ;BUILT?
2280 006540 001365      BNE    3$            ;IF NO, GO BUILD THE NEXT ONE.
2281 006542 012703 010724      MOV    #MDATA,R3     ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
2282 006546 112623      4$:   MOV    (SP)+,(R3)+  ;STORE THE CHARACTER, STARTING WITH THE MOST
2283 006550 105337 006624      DECB  CHRCNT         ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
2284 006554 001374      BNE    4$           ;IF NO, GO TRANSFER ANOTHER
2285 006556 105700      TSTB  R0             ;ARE ANY SPACES TO BE PRINTED?
2286 006560 001404      BEQ   6$           ;IF NO, DON'T SET UP ANY
2287 006562 112723 000040      5$:   MOV    #040,(R3)+  ;ADD A SPACE TO THE OUTPUT BUFFER
2288 006566 105300      DECB  R0             ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
2289 006570 001374      BNE    5$           ;IF YES, GO ADD ANOTHER SPACE
2290 006572 105013      6$:   CLRB  (R3)        ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
2291 006574 104402 010724      TYPE  ,MDATA        ;PRINT THE STRING WE JUST BUILT
2292 006600 005337 006622      DEC   WRDCNT        ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
2293 006604 001336      BNE   1$           ;IF YES, GO CONVERT THEM
2294 006606 012605      MOV   (SP)+,R5      ;RESTORE R5
2295 006610 012604      MOV   (SP)+,R4      ;RESTORE R4
2296 006612 012603      MOV   (SP)+,R3      ;RESTORE R3
2297 006614 012601      MOV   (SP)+,R1      ;RESTORE R1
2298 006616 012600      MOV   (SP)+,R0      ;RESTORE R0
2299 006620 000002      RTI                  ;RETURN TO THE MAIN PROGRAM
2300 006622 000000      WRDCNT: 0
2301 006624 000      CHRCNT: .BYTE
2302 006625 000      SPACNT: .BYTE 0      ;NUMBER OF CHARACTERS TO PRINT
2303
2304 006626 000000      BINWRD: 0           ;NUMBER OF SPACES TO PRINT
2305
2306
2307                          ;TRAP DISPATCH SERVICE

```

```

2308                                     ;ARGUMENT OF TRAP IS EXTRACTED
2309                                     ;AND USED AS OFFSET TO OBTAIN POINTER
2310                                     ;TO SELECTED SUBROUTINE
2311
2312 006630 010046 .TRPSR: MOV      RO,-(SP)      ;SAVE RO. USE RO TO FIND TRAP ROUTINE
2313 006632 016600 000002      MOV      2(SP),RO      ;GET TRAP ADDRESS
2314 006636 005740      TST      -(RO)        ;GET TRAP
2315 006640 111000      MOVVB   (RO),RO       ;GET RIGHT BYTE OF TRAP(TRAP OFFSET)
2316 006642 006300      ASL      RO          ;POSITION OFFSET FOR TABLE INDEXING
2317 006644 016000 002002      MOV      .TRPTAB(RO),RO ;PLACE INDEXED ADDRESS OF TABLE IN RO
2318 006650 000200      RTS      RO          ;TRANSFER TO THAT ADDRESS AND RESTORE OLD RO
2319
2320                                     ;DEVICE CLEAR ROUTINE
2321                                     ;ISSUE A DEVICE CLEAR
2322                                     ;-----
2323 006652 .DEVICE.CLR:
2324 006652 052777 000020 173162 BIS      #DCLR,@DZCSR    ;SET DCLR
2325 006660 032777 000020 173154 1$: BIT      #DCLR,@DZCSR  ;DID IT CLEAR?
2326 006666 001374      BNE      1$          ;BR IF NO
2327 006670 000002      RTI          ;EXIT ROUTINE
2328
2329                                     ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
2330                                     ;-----
2331 006672 104413 .DCLASM: DEVICE.CLR    ;ISSUE A DEVICE CLEAR
2332 006674 153777 001417 173140 BISE    MNTFLG,@DZCSR  ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
2333 006702 000002      RTI          ;RETURN TO CALLING ROUTINE
2334
2335 .DELAY:
2336 006704 010046      MOV      RO,-(SP)      ;SAVE RO
2337 006706 013700 006722      MOV      DLYCNT,RO    ;SET COUNT
2338 006712 005300      1$: DEC      RO        ;DELAY
2339 006714 001376      BNE      1$          ;
2340 006716 012600      MOV      (SP)+,RO     ;RESTORE RO
2341 006720 000002      RTI          ;LEAVE ROUTINE
2342 006722 000001      DLYCNT: .WORD    1    ;PATCHABLE LOC FOR MORE TIME
2343
2344                                     ;ADVANCE TO NEXT TEST HANDLER
2345                                     ;-----
2346
2347 006724 013716 001360 .ADVANCE: MOV      NEXT,(SP) ;CRUNCH STACK WITH ADDRESS OF SCOPE CALL
2348 006730 005037 001362      CLR      LOCK        ;RESET TIGHT LOOP ADDRESS
2349 006734 000002      RTI          ;CHECK TO SEE IF OLD TEST GETS REPEATED
2350
2351                                     ;ERROR HANDLFR
2352                                     ;-----
2353
2354 006736 004737 007360 $ERROR: JSR      PC,SERV.G ;FIND OUT IF <^G> WAS HIT
2355 006742 032777 010000 172210 BIT      #SW12,@SWR    ;BELL ON ERROR?
2356 006750 001406      BEQ      XBX        ;BR IF NO BELL
2357 006752 105777 172212      TSTB   @$TPS        ;TTY READY.
2358 006756 100003      BPL      XBX        ;DON'T WAIT IF TTY NOT READY.
2359 006760 112777 000207 172204 MOVVB   #207,@$TPB    ;PUSH A BELL AT THE TTY.
2360 006766 032777 020000 172164 XBX: BIT      #SW13,@SWR ;DELETE ERROR PRINT OUT?
2361 006774 001113      BNE      HALTS      ;BR IF NO PRINT OUT WANTED.
2362 006776 021637 001136      CMP      (SP),$ERRPC ;WAS THIS ERROR FOUND LAST TIME?
2363 007002 001404      BEQ      1$        ;BR IF YES

```

```

2364 007004 011637 001136      MOV      (SP), $ERRPC      ;RECORD BEING HERE
2365 007010 105037 001123      CLRB     $ERFLG           ;PREPARE HEADER
2366 007014 104407      1$: SAVO5                ;SAVE ALL PROC REGISTERS
2367 007016 011605      MOV      (SP), R5         ;GET THE PC OF ERROR
2368 007020 162705 000002      SUB      #2, R5          ;GET ADDRESS OF TRAP CALL
2369 007024 011504      MOV      (R5), R4        ;GET ERROR INSTRUCTION
2370 007026 110437 001134      MOVB    R4, $ITEMB       ;COPY TEST NUMBER FOR APT HANDLING
2371 007032 006304      ASL      R4              ;MULT BY TWO
2372 007034 061504      ADD      (R5), R4        ;DOUBLE IT
2373 007036 006304      ASL      R4              ;MULT AGAIN
2374 007040 042704 177001      BIC      #177001, R4     ;CLEAR JUNK
2375 007044 062704 027064      ADD      #.ERRTAB, R4    ;GET POINTER
2376 007050 012437 007174      MOV      (R4)+, ERRMSG   ;GET ERROR MESSAGE
2377 007054 012437 007206      MOV      (R4)+, DATAHD ;GET DATA HEADRER
2378 007060 011437 007220      MOV      (R4), DATABP   ;GET DATA TABLE
2379 007064 105737 001123      TSTB    $ERFLG          ;TYPE HEADER
2380 007070 001403      BEQ      TYPMSG         ;BR IF YES
2381 007072 005737 007220      TST     DATABP          ;DOES DATA TABLE EXIST?
2382 007076 001044      BNE      TYPDAT         ;BR IF YES.
2383 007100 104402 001231      TYPMSG: TYPE , $CRLF     ;TYPE A CARRIAGE RETURN
2384 007104 104402 001231      TYPE    , $CRLF         ;AND TYPE ANOTHER
2385 007110 005737 001362      TST     LOCK            ;
2386 007114 001402      BEQ      1$             ;
2387 007116 104402 010314      TYPE    , MASTEK        ;
2388 007122 104402 010302      1$:  TYPE    , MTSTN     ;
2389 007126 104412 007352      CNVRT   , XTSTN         ;SHOW IT
2390 007132 104402 010371      TYPE    , MERRPC       ;TYPE PC.
2391 007136 104412 007344      CNVRT   , ERTABO       ;SHOW IT
2392 007142 104402 010244      TYPE    , MCSRX        ;
2393 007146 104412 004742      CNVRT   , XCSR         ;
2394 007152 104402 001231      TYPE    , $CRLF        ;GIVE A CR/LF
2395 007156 112737 177777 001123      MOVB    #-1, $ERFLG     ;NO MORE HEADER UNLESS NO DATA TABLE.
2396 007164 005737 007174      TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
2397 007170 001402      BEQ      WTBS.FM        ;BR IF NO.
2398 007172 104402      TYPE    ;
2399 007174 000000      ERRMSG: 0               ;ERROR MESSAGE
2400 007176      WTBS.FM: ;
2401 007176 005737 007206      TST     DATAHD         ;DATA HEADER?
2402 007202 001402      BEQ      TYPDAT         ;BR IF NO
2403 007204 104402      TYPE    ;
2404 007206 000000      DATAHD: 0              ;DATA HEADER
2405 007210 005737 007220      TYPDAT: TST DATABP      ;DATA TABLE?
2406 007214 001402      BEQ      RESREG         ;BR IF NO.
2407 007216 104411      CONVRT ;SHOW
2408 007220 000000      DATABP: 0               ;DATA TABLE
2409 007222 104410      RESREG: RESO5           ;RESTORE PROC REGISTERS
2410 007224 122737 000001 001254      HALTS:  CMPB #APTENV, $ENV ;IS APT RUNNING?
2411 007232 001007      BNE     2$              ;SKIP APT CALL IF NOT
2412 007234 113737 001134 007246      MOVB    $ITEMB, 7$      ;COPY ERROR NUMBER
2413 007242 004737 005600      JSR     PC, $ATY4       ;CALL APT SERVICE
2414 007246 000000      7$:   .WORD 0           ;ERROR NUMBER STUCK HERE
2415 007250 000777      8$:   BR 8$              ;LOCK UP HERE
2416 007252 022737 004726 000042      2$:   CMP #SENDAD, @#42 ;CHECK TO SEE IF IN ACT-11 MODE
2417 007260 001403      BEQ     1$              ;IF SO, HANDLE ACCORDINGLY
2418 007262 005777 171672      TST     @SWR            ;HALT ON ERROR?
2419 007266 100004      BPL     EXITER          ;BR IF NO HALT ON ERROR

```

```

2420 007270 016677 000002 171664 1S:  MOV      2(SP),@DISPLAY  :SHOW ERROR PC IN DATA DISPLAY
2421 007276 000000                HALT          :HALT
2422 007300 005237 001132          EXITER:  INC      $ERTTL      :UPDATE ERROR COUNT
2423 007304 032777 000400 171646          BIT      #SW08,@SWR      :GOTO TOP OF TEST?
2424 007312 001007                BNE        1$          :BR IF YES
2425 007314 032777 002000 171636          BIT      #SW10,@SWR      :GOTO NEXT TEST?
2426 007322 001407                BEQ        2$          :BR IF NO
2427 007324 013737 001360 001126          MOV      NEXT,$LPADR     :SET FOR NEXT TEST
2428 007332 012706 001120 1S:  MOV      #STACK,SP      :RESET SP
2429 007336 000177 171564          JMP      @SLPADR        :GOTO SPECIFIED TEST
2430 007342 000002                2S:  RTI                   :RETURN
2431 007344 000001          ERTAB0: 1
2432 007346          006          002          .BYTE      6,2
2433 007350 001402                SAVPC
2434 007352 000001          XTSTN: 1
2435 007354          002          002          .BYTE      2,2
2436 007356 001122                $TSTNM
2437 007360 022737 177570 001160  SERV.G:  CMP      #177570,$WR      :IS THE SWITCH REGISTER HARDWIRED?
2438 007366 001513                BEQ        6$          :IF SO, IGNORE ^G
2439 007370 017746 171572          MOV      @STKB,-(SP)     :OTHERWISE, GET THE LAST CHARACTER TYPED
2440 007374 042716 000200          BIC      #BIT7,(SP)      :STRIP PARITY(EIGHTH) BIT
2441 007400 122726 000007          CMPB     #7,(SP)+        :IS IT ^G?
2442 007404 001104                BNE        6$          :IF NOT, IGNORE INPUT
2443 007406 032777 004000 171550          BIT      #4000,@STKS     :RX BUSY?
2444 007414 001361                BNE        SERV.G       :BR IF YES
2445 007416 017737 171536 007640          MOV      @SWR,90$        :SAVE (SWR).
2446 007424 013777 007640 171526 1S:  MOV      90$,@SWR
2447 007432 104402 007620          TYPE     .89$
2448 007436 104412 007632          CNVRT    .88$
2449 007442 104402 007642          TYPE     .91$
2450 007446 105777 171512          TSTB     @STKS          :WAIT FOR DONE.
2451 007452 100375                BPL        -4
2452 007454 017746 171506          MOV      @STKB,-(SP)
2453 007460 042716 000200          BIC      #BIT7,(SP)
2454 007464 122726 000015          CMPB     #15,(SP)+
2455 007470 001450                BEQ        5$
2456 007472 005077 171462          CLR      @SWR
2457 007476 105777 171466          2S:  TSTB     @STPS
2458 007502 100375                BPL        -4
2459 007504 016677 177776 171460          MOV      -2(SP),@STPB
2460 007512 000241                CLC
2461 007514 006177 171440          ROL      @SWR
2462 007520 006177 171434          ROL      @SWR
2463 007524 006177 171430          ROL      @SWR
2464 007530 103735                BCS        1$          :ERROR
2465 007532 026627 177776 000060          CMP      -2(SP),#60
2466 007540 002731                BLT        1$
2467 007542 026627 177776 000067          CMP      -2(SP),#67
2468 007550 003325                BGT        1$
2469 007552 042766 177770 177776          BIC      #^C<7>,-2(SP)
2470 007560 056677 177776 171372          BIS      -2(SP),@SWR
2471 007566 105777 171372          TSTB     @STKS
2472 007572 100375                BPL        -4
2473 007574 017746 171366          MOV      @STKB,-(SP)
2474 007600 042716 000200          BIC      #BIT7,(SP)
2475 007604 122726 000015          CMPB     #15,(SP)+

```

```

2476 007610 001332      BNE      2$      :
2477 007612 104402 001231 5$:      TYPE      ,SCLRF  :
2478 007616 000207      6$:      RTS      PC      :
2479
2480 007620 020200 051450 051127 89$:      .ASCIZ  <200>? (SWR)=/?
2481 007626 03645 000057
2482
2483 007632 000001      .EVEN
2484 007634 006 000      88$:      1
2485 007636 007640      .BYTE  6,0
2486 007640 000000      90$:      .WORD  0
2487 007642 036457 000057 91$:      .ASCIZ  ?/=/?
2488
2489      .EVEN
2490      .SBTTL  POWER DOWN AND UP ROUTINES
2491
2492      ::*****
2493      :POWER DOWN ROUTINE
2493 007646 012737 010012 000024 $PWRDN: MOV      #SILLUP,@PWRVEC  ::SET FOR FAST UP
2494 007654 012737 000340 000026      MOV      #340,@PWRVEC+2  ::PRIO:7
2495 007662 010046      MOV      R0,-(SP)      ::PUSH R0 ON STACK
2496 007664 010146      MOV      R1,-(SP)      ::PUSH R1 ON STACK
2497 007666 010246      MOV      R2,-(SP)      ::PUSH R2 ON STACK
2498 007670 010346      MOV      R3,-(SP)      ::PUSH R3 ON STACK
2499 007672 010446      MOV      R4,-(SP)      ::PUSH R4 ON STACK
2500 007674 010546      MOV      R5,-(SP)      ::PUSH R5 ON STACK
2501 007676 017746 171256      MOV      @SWR,-(SP)     ::PUSH @SWR ON STACK
2502 007702 010637 010016      MOV      SP,$SAVR6     ::SAVE SP
2503 007706 012737 007720 000024      MOV      #SPWRUP,@PWRVEC  ::SET UP VECTOR
2504 007714 000000      HALT
2505 007716 000776      BR      -2      ::HANG UP
2506
2507      ::*****
2508      :POWER UP ROUTINE
2509 007720 012737 010012 000024 $PWRUP: MOV      #SILLUP,@PWRVEC  ::SET FOR FAST DOWN
2510 007726 013706 010016      MOV      $SAVR6,SP     ::GET SP
2511 007732 005037 010016      CLR      $SAVR6      ::WAIT LOOP FOR THE TTY
2512 007736 005237 010016      1$:      INC      $SAVR6      ::WAIT FOR THE INC
2513 007742 001375      BNE      1$      ::OF WORD
2514 007744 012677 171210      MOV      (SP)+,@SWR    ::POP STACK INTO @SWR
2515 007750 012605      MOV      (SP)+,R5     ::POP STACK INTO R5
2516 007752 012604      MOV      (SP)+,R4     ::POP STACK INTO R4
2517 007754 012603      MOV      (SP)+,R3     ::POP STACK INTO R3
2518 007756 012602      MOV      (SP)+,R2     ::POP STACK INTO R2
2519 007760 012601      MOV      (SP)+,R1     ::POP STACK INTO R1
2520 007762 012600      MOV      (SP)+,R0     ::POP STACK INTO R0
2521 007764 012737 007646 000024      MOV      #SPWRDN,@PWRVEC  ::SET UP THE POWER DOWN VECTOR
2522 007772 012737 000340 000026      MOV      #340,@PWRVEC+2  ::PRIO:7
2523 010000 104402      TYPE      MPFAIL      ::REPORT THE POWER FAILURE
2524 010002 010020      $PWRMG: .WORD      MPFAIL  ::POWER FAIL MESSAGE POINTER
2525 010004 012716      MOV      (PC)+,(SP)   ::RESTART AT RESTART
2526 010006 011434      $PWRAD: .WORD      RESTART  ::RESTART ADDRESS
2527 010010 000002      RTI
2528 010012 000000      $ILLUP: HALT      ::THE POWER UP SEQUENCE WAS STARTED
2529 010014 000776      BR      -2      ::BEFORE THE POWER DOWN WAS COMPLETE
2530 010016 000000      $SAVR6: 0      ::PUT THE SP HERE
2531 010020 050200 051127 043040 MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /

```

CZDZA-FO  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 56  
POWER DOWN AND UP ROUTINES

SEQ 0055

```

(2) 010063 200 047105 020104 MEPASS: .ASCIZ <200>/END PASS CZDZA-F /
(2) 010106 051200 047125 044516 MR: .ASCIZ <200>/RUNNING /
(2) 010122 050200 047522 051107 MERR2: .ASCIZ <200>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 010171 200 047111 052523 MERR3: .ASCIZ <200>/INSUFFICIENT DATA!/
(2) 010215 200 047514 045503 MLOCK: .ASCIZ <200>/LOCK ON SELECTED TEST/
(2) 010244 051503 035122 000040 MCSRX: .ASCIZ /CSR: /
(2) 010252 042526 035103 000040 MVECX: .ASCIZ /VEC: /
(2) 010260 040520 051523 051505 MPASSX: .ASCIZ /PASSES: /
(2) 010271 105 051122 051117 MERRX: .ASCIZ /ERRORS: /
(2) 010302 042524 052123 047040 MTSTN: .ASCIZ /TEST NO: /
(2) 010314 020052 000 MASTEK: .ASCIZ /* /
(2) 010317 200 042523 020124 MNEW: .ASCIZ <200>/SET SWITCH REG TO DZ11'S DESIRED ACTIVE./
(2) 010371 120 035103 000040 MERRPC: .ASCIZ /PC: /
(2) 010376 046600 050101 047440 XHEAD: .ASCIZ <200>/MAP OF DZ11 STATUS/<200>
(2) 010423 200 046111 042514 MBADLN: .ASCIZ <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
(2) 010466 010466 .EVEN
(2) 010466 000002 XSTATQ: 2
2532 010470 006 003 .BYTE 6,3
2533 010472 001220 $TMP1
2534 010474 006 002 .BYTE 6,2
2535 010476 001222 $TMP2
2536 .EVEN
2537 :THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
2538 :-----
2539 :E=EXTERNAL LOOP BACK
2540 :I=INTERNAL LOOP BACK
2541 :S=STAGGERED LOOP BACK
2542 010500 017605 000000 .SETFLG:MOV @ (SP),R5 ;PICK UP ADDRESS OF TAG
2543 010504 042737 000040 010620 BIC #40,INBUF ;STRIP LOWER CASE
2544 010512 122737 000105 010620 CMPB #'E,INBUF ;IS IT EXTERNAL LOOP BACK ?
2545 010520 001005 BNE 4$ ;NO
2546 010522 013715 010612 MOV 1$, (R5) ;YES STORE INFO
2547 010526 105037 001417 CLRB MNTFLG ;SET MAINT BIT =0
2548 010532 000422 BR 7$ ;GET OUT
2549 010534 122737 000111 010620 4$: CMPB #'I,INBUF ;IS IT INTERNAL LOOP BACK ?
2550 010542 001006 BNE 5$ ;NO
2551 010544 013715 010614 MOV 2$, (R5) ;YES STORE INFO
2552 010550 112737 000010 001417 MOVB #MAINT,MNTFLG ;SET UP THE MAINTENANCE FLAG LOADER
2553 010556 000410 BR 7$ ;GET OUT
2554 010560 122737 000123 010620 5$: CMPB #'S,INBUF ;IS IT STAGGERED LOOP BACK ?
2555 010566 001007 BNE 6$ ;WHAT ?
2556 010570 013715 010616 MOV 3$, (R5) ;YES STORE INFO
2557 010574 105037 001417 CLRB MNTFLG ;ZERO BITS
2558 010600 062716 000002 7$: ADD #2, (SP) ;POP AROUND
2559 010604 000002 RTI
2560 010606 104404 6$: INSTER ;RETRY
2561 010610 000733 BR .SETFLG ;DITTO
2562 010612 000200 1$: .WORD 200 ;EXTERNAL = E
2563 010614 000000 2$: .WORD 0 ;INTERNAL = I
2564 010616 100000 3$: .WORD 100000 ;STAGGERED S
2565 :
2566 :BUFFERS FOR INPUT-OUTPUT
2567 :
2568 010620 000000 INBUF: 0
2569 010662 010662 .+40
2570 010662 000000 TEMP: 0

```

2571		010724				. = .+40					
2572	010724	000000				MDATA: 0					
2573		010766				. = .+40					
2574											
2575	010766	011637	011064			SET.PS: MOV	(SP),3\$				
2576	010772	162737	000002	011064		SUB	#2,3\$				
2577	011000	017737	000060	011066		MOV	@3\$,4\$				
2578	011006	022737	106427	011066		CMP	#106427,4\$				
2579	011014	001003				BNE	1\$				
2580	011016	011637	011064			MOV	(SP),3\$				
2581	011022	000412				BR	2\$				
2582	011024	022737	106437	011066	1\$:	CMP	#106437,4\$				
2583	011032	001401				BEQ	.+4				
2584	011034	000000				HALT					:RESERVED INSTRUCTION NOT 'MTPS'
2585	011036	011637	011064			MOV	(SP),3\$				
2586	011042	017737	000016	011064		MOV	@3\$,3\$				
2587	011050	062716	000002		2\$:	ADD	#2,(SP)				
2588	011054	017766	000004	000002		MOV	@3\$,2(SP)				
2589	011062	000002				RTI					
2590	011064	000000			3\$:	0					
2591	011066	000000			4\$:	0					



```

2592
2593
2594
2595
2596
2597
2598
2599
2600
2601 011070 005737 001404          CYCLE: TST      DZACTV      ;ARE ANY DZ11'S TO BE TESTED?
2602 011074 001004                    BNE      1$          ;BR IF OK.
2603 011076 104402 010122          TYPE      ,MERR2     ;NO DZ11'S SELEC LVL
2604 011102 000000                    HALT                     ;STOP THE SHOW.
2605 011104 000776                    BR                          ;DISQUALIFY CONT. SW.
2606 011106 013737 005234 001226 1$: MOV      $MXCNT,$TIMES ;RESTORE THE NUMBER OF ITERATIONS TO MAKE
2607 011114 033737 001406 001404 BIR      RUN,DZACTV  ;IS THIS ONE 'ACTIVE'
2608 011122 001020                    BNE      2$          ;BR IF GOOD ONE FOUND.
2609 011124 000241                    CLC
2610 011126 006137 001406          ROL      RUN          ;UPDATE POINTER
2611 011132 005537 001406          ADC      RUN          ;CATCH CARRY FROM RUN
2612 011136 062737 000014 001412 ADD      #14,ACTIVE    ;UPDATE ADDRESS POINTER.
2613 011144 022737 002000 001412 CMP      #DZ.END,ACTIVE ;HAVE WE PASSED THE END OF THE MAP?
2614 011152 001355                    BNE      1$          ;IF NO, KEEP GOING; NOT ALL TESTED FOR.
2615 011154 012737 001500 001412 MOV      #DZ.MAP,ACTIVE ;RESET ADDRESS POINTER.
2616 011162 000751                    BR       1$          ;KEEP LOOKING FOR ACTIVE DZ11
2617 011164 000241                    2$: CLC
2618 011166 006137 001406          ROL      RUN          ;UPDATE POINTER.
2619 011172 005537 001406          ADC      RUN          ;CATCH CARRY.
2620 011176 013700 001412          MOV      ACTIVE,R0    ;GET ADDRESS POINTER.
2621 011202 062737 000014 001412 ADD      #14,ACTIVE    ;UPDATE.
2622 011210 022737 002000 001412 CMP      #DZ.END,ACTIVE
2623
2624 011216 001003                    BNE      3$          ;ALL DONE?
2625 011220 012737 001500 001412 MOV      #DZ.MAP,ACTIVE ;BR IF NO.
2626 011226 012037 001310                    3$: MOV      (R0)+,$BASE ;RESTORE POINTER.
2627 011232 012037 002072          MOV      (R0)+,DZRIV ;LOAD SYSTEM CTRL. REG
2628 011236 012037 027060          MOV      (R0)+,DZPRT ;LOAD VECTOR
2629 011242 113737 027061 001414 MOV      (R0)+,DZPRT  ;LOAD PRIORITY
2630 011250 042737 100000 027060 MOVB    DZPRT+1,EIAFLG ;EIA OR ZOMA
2631 011256 012037 001364          BIC      #BIT15,DZPRT ;CLEAR FLAG
2632 011262 012037 001366          MOV      (R0)+,LINE  ;SET UP LINE DZ LINES ACTIVE
2633 011266 012037 001370          MOV      (R0)+,PAR   ;SET UP PARAMETERIZATION
2634 011272 004737 026652          MOV      (R0)+,MODE  ;SET UP MAINTENANCE MODE
2635 011276 005737 000042          JSR      PC,DZLEV    ;SET UP
2636 011302 001051                    TST      @#42        ;ARE WE UNDER MONITOR CONTROL?
2637 011304 032777 000002 167646 BNE      4$          ;IF YES, SKIP THIS SETUP
2638 011312 001445                    BIT      #SW01,@SWR  ;IF SW01=1, GET STARTING TEST #
2639 011314 104402 001231                    BEQ      4$          ;BR IF NO TEST IS TO BE INPUTTED
2640 011320 104403                    7$: TYPE      ,SCLF
2641 011322 010302          INSTR
2642 011324 104405          MTSTN
2643 011326 000001          PARAM
2644 011330 001000          1
2645 011332 001122          1000
2646 011334 000          $STNM
2647 011335 001          .BYTE 0
          .BYTE 1
;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

```

```

2648 011336 012700 012374      MOV      #TST1,R0
2649 011342 022710 000004      5$:    (MP      #4,(R0)
2650 011346 001020      BNE      6$
2651 011350 022760 012737 000002  (MP      #12737,2(R0)
2652 011356 001014      BNE      6$
2653 011360 023760 001122 000004  (MP      $TSTNM,4(R0) ;IS THIS THE TEST ?
2654 011366 001010      BNE      6$ ;IF NOT, DON'T PROCESS NUMBER
2655 011370 010037 001126      MOV      R0,$LPADR ;SAVE PC
2656 011374 062737 000002 001126  (ADD     #2,$LPADR ;POP OVER SCOPE
2657 011402 104402 001231      TYPE    , $CRLF
2658 011406 000412      BR       8$
2659 011410 005720      6$:    TST      (R0)+
2660 011412 020027 023002  (CMP     R0,#TLAST+10
2661 011416 001351      BNE      5$
2662 011420 104402 001230      TYPE    , $QUES
2663 011424 000733      BR       7$
2664 011426 012737 012374 001126  4$:    MOV      #TST1,$LPADR ;PREPARE TEST ADDRESS
2665 011434      8$:
2666 011434 000177 167466  (RESTART:JMP @ $LPADR ;GO START TESTING.***WARNING!***
2667 ;THIS JUMP IS USED BY POWER UP ROUTINE
2668

```

```

2669                                     :-ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.
2670                                     ; IF BIT7 IN THE ENVIRONMENT MODE ($ENVM) BYTE IS SET,
2671                                     ; THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.
2672
2673 011440 012700 001500          SETAPT: MOV    #DZ.MAP,R0      ;POINT TO THE DEVICE MAP TABLE
2674 011444 013701 001310          MOV    $BASE,R1      ;BUILD DEVICE ADDRESSES IN R1
2675 011450 013702 001304          MOV    $VECT1,R2     ;BUILD DEVICE VECTORS IN R2
2676 011454 042702 177007          BIC    #^C<770>,R2  ;STRIP AWAY OTHER INFORMATION
2677
2678 011460 113703 001305          MOVB   $VECT1+1,R3   ;LOAD THE INTERRUPT PRIORITY FROM R3
2679 011464 106003                 RORB   R3            ;ALIGN THE NUMBER
2680 011466 106003                 RORB   R3            ;ALIGN THE NUMBER
2681 011470 106003                 RORB   R3            ;ALIGN THE NUMBER
2682 011472 106003                 RORB   R3            ;ALIGN THE NUMBER
2683 011474 106003                 RORB   R3            ;ALIGN THE NUMBER
2684 011476 042703 177770          BIC    #^C<7>,R3    ;REMOVE ALL BUT BUS LEVEL NUMBER
2685 011502 012704 001320          MOV    #SDDW0,R4    ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
2686 011506 013705 001312          MOV    $DEVM,R5     ;GET THE MAP OF ACTIVE DEVICES
2687 011512 010537 001404          MOV    R5,DZACTV    ;SAVE THE BIT MAP
2688 011516 006005          1$:  ROR    R5            ;GET A DEVICE SELECTION BIT
2689 011520 103407          BCS   3$            ;IF IT IS SELECTED, GO SET UP A MAP
2690 011522 001425          BEQ   5$            ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
2691 011524 005724          TST   (R4)+         ;POINT TO NEXT DEVICE DESCRIPTOR
2692 011526 062701 000010          2$:  ADD   #10,R1      ;SET UP THE NEXT ADDRESS
2693 011532 062702 000010          ADD   #10,R2      ;SET UP THE NEXT VECTOR GROUP
2694 011536 000767          BR    1$            ;GO SEE IF MORE DEVICES REMAIN
2695 011540 010120          3$:  MOV    R1,(R0)+     ;LOAD DEVICE ADDRESS
2696 011542 010220          MOV    R2,(R0)+     ;LOAD THE VECTOR ADDRESS
2697 011544 010320          MOV    R3,(R0)+     ;LOAD THE INTERRUPT PRIORITY LEVEL
2698 011546 013720 001314          MOV    $CDW1,(R0)+  ;GET THE NUMBER OF LINES IN OPERATION
2699 011552 012420          MOV    (R4)+,(R0)+  ;LOAD DEVICE PARAMETERS
2700 011554 100006          BPL   4$            ;IF 20MA MODE SELECTED, SET IT UP
2701 011556 052760 100000 17777;  BIS   #100000,-6(R0) ;SET THE 20MA FLAG IN DZLVN
2702 011564 042760 100000 17777;  BIC   #100000,-2(R0) ;CLEAR THE FLAG IN DZPARN
2703 011572 005020          CLR   (R0)+         ;DEFAULT OPERATION TO INTERNAL MAINTENANCE MODE
2704 011574 000754          BR    2$            ;GO BUILD THE NEXT ADDRESS
2705 011576 012710 177777          5$:  MOV    #-1,(R0)     ;TERMINATE THE DEVICE MAP
2706 011602 012737 001256 001160  MOV    #SSWREG,SWR   ;SET TO SOFTWARE APT SWITCH REGISTER
2707 011610 000207          RTS    PC           ;RETURN TO PRINT STATUS TABLE
2708
2709
2710                                     ;*ROUTINE USED TO 'AUTO SIZE' THE DZ11
2711                                     ;*CSR AND VECTOR.
2712                                     ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
2713                                     ;*      ADDRESS RANGE (160000:163700)
2714                                     ;*      AND THE VECTOR MAY BE ANY WHERE IN THE
2715                                     ;*      FLOATING VECTOR RANGE (300:770)
2716                                     ;*
2717                                     ;*
2718
2718 011612          AUTO.SIZE:
2719 011612 000005          .RESET
2720 011614 105337 001415          DECB  INIFLG        ;INSURE A BUS INIT.
2721 011620 012702 001500          CSRMAP: MOV #DZ.MAP,R2 ;SHOW THAT I WAS HERE
2722 011621 012703 001320          MOV   #SDDW0,R3    ;LOAD MAP POINTER.
2723 011630 005022          1$:  CLR   (R2)+         ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
2724 011632 022702 002000          CMP   #DZ.END,R2  ;ZERO ENTIRE MAP
                                     ;ALL DONE?

```

CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 61  
POWER DOWN AND UP ROUTINES

SEQ 0060

```

2725 011636 001374          BNE      1$          :BR IF NO
2726 011640 105037 001410   CLR      DZNUM      :SET OCTAL NUMBER OF DZ11'S TO 0
2727 011644 012702 001500   MOV      #DZ.MAP,R2
2728 011650 012701 160000   MOV      #160000,R1  :SET FOR FIRST ADDRESS TO BE TESTED
2729 011654 012737 012174 000004  MOV      #6$,@#4     :SET FOR NON-EXISTENT DEVICE TIME OUT
2730 011662 052711 000040 2$:   BIS      #BIT5,(R1)  :TRY TO SET MASTER SCAN ENABLE
2731 011666 05276' 000200 000004  BIS      #BIT7,4(R1) :TRY TO TRANSMIT ON LINE 7
2732 011674 005000          CLR      R0         :USE R0 AS A COUNTER
2733 011676 005711          7$:   TST      (R1)      :HAS TRANSMITTER READY COME UP?
2734 011700 100403          BMI      8$         :IF SO, GO GET A FINAL CHECK
2735 011702 005300          DEC      R0         :REDUCE COUNT. TIME UP?
2736 011704 001374          BNE      7$         :IF NOT, KEEP WAITING
2737 011706 000463          BR       3$         :ASSUME IT'S NOT A DZ11
2738 011710 032761 000200 000004 8$:   BIT      #BIT7,4(R1) :IS LINE 7 ENABLE STILL SET? IT SHOULD BE
2739 011716 001457          BEQ      3$         :IF IT'S NOT, ASSUME IT'S NOT A DZ11
2740 011720 032711 000040   BIT      #BIT5,(R1) :IS MASTER SCAN ENABLE STILL SET?
2741 011724 001454          BEQ      3$         :IF NOT, ASSUME IT'S NOT A DZ11
2742 011726 005000          CLR      R0
2743 011730 052711 000020   BIS      #20,(R1)   :SET DEVICE CLEAR
2744 011734 032711 000020   BIT      #20,(R1)   :SHOULD STAY SET FOR A WHILE IF DZ
2745 011740 001446          BEQ      3$         :BR IF NOT DZ11
2746 011742 032711 000020   BIT      #20,(R1)   :WAIT FOR BIT TO CLEAR
2747 011746 001404          BEQ      .+12       :BR WHEN CLEARED
2748 011750 104414          DELAY
2749 011752 005200          INC      R0
2750 011754 001372          BNE      .-12
2751 011756 000437          BR       3$         :BIT NOT CLEARED! MUST NOT BE DZ11
2752 011760 005011          CLR      (R1)      :GET RID OF MASTER SCAN ENABLE
2753 011762 005061 000004  CLR      4(R1)     :GET RID OF LINE 7 ENABLE
2754          ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DZ11 CSR ADDRESS.
2755 011766 010122          MOV      R1,(R2)+  :STORE CSR IN CORE TABLE.
2756 011770 005722          TST      (R2)+     :POP OVER VECTOR STORE AREA
2757 011772 012722 000005   MOV      #5,(R2)+  :SET THE DEFAULT BUS LEVEL
2758 011776 052761 177400 000004  BIS      #177400,4(R1) :TRY TO SET ALL DTR BITS
2759 012004 032761 177400 000004  BIT      #177400,4(R1) :IF ANY SET ASSUME EIA BOARD
2760 012012 001003          BNE      9$         :IF NONE SET ASSUME BOARD IS
2761 012014 052762 100000 177776 9$:   BIS      #BIT15,-2(R2) :20 MA, SET 20 MA FLAG
2762 012022 012722 000377   MOV      #377,(R2)+ :SET THE DEFAULT LINE SELECTION PARAMETER
2763 012026 012712 017470   MOV      #17470,(R2) :SET THE DEFAULT PARAMETERS
2764 012032 012223          MOV      (R2)+,(R3)+ :COPY PARAMETERS INTO ETABLE DESCRIPTOR
2765 012034 005022          CLR      (R2)+     :SET THE DEFAULT MODE OF OPERATION
2766 012036 012712 177777   MOV      #-1,(R2)  :TERMINATE LIST
2767 012042 105237 001410   INCB    DZNUM      :UPDATE DEVICE COUNTER
2768 012046 122737 000020 001410  CMPB    #20,DZNUM  :ARE MAX. NO. OF DEV FOUND?
2769 012054 001405          BEQ      100$      :YES DON'T LOOK FOR ANY MORE.
2770 012056 062701 000010 3$:   ADD      #10,R1    :UPDATE CSR POINTER ADDRESS
2771 012062 022701 163700   CMP     #163700,R1
2772 012066 001275          BNE      2$         :BR IF MORE ADDRESS TO CHECK.
2773 012070          100$:
2774 012070 105737 001410   TSTB   DZNUM      :WERE ANY DZ11'S FOUND AT ALL?
2775 012074 001432          BEQ     5$         :ERROR AUTO SIZER FOUND NO DZ11'S IN THIS SYS.
2776 012076 113701 001410   MOVB   DZNUM,R1
2777 012102 110137 001411   MOVB   R1,SAVNUM  :SAVE NUMBER OF DEVICES
2778 012106 012737 000001 001404  MOV     #1,DZACTV
2779 012114 005301          4$:   DEC     R1
2780 012116 001404          BEQ     98$

```

```

2781 012120 000261          SEC
2782 012122 006137 001404  ROL      DZACTV
2783 012126 000772          BR      4$
2784 012130 013737 001500 001310 48$:  MOV    DZCRO,$BASE      ;POINT TO THE ADDRESS OF FIRST DEVICE
2785 012136 013737 001512 001314      MOV    MANTO,$CDW1     ;INDICATE TO ETABLE WHAT MODE IS BEING USED
2786 012144 012737 000006 000004 99$:  MOV    #6,@#4         ;RESTORE TRAP VECTOR
2787 012152 013737 001404 001312      MOV    DZACTV,$DEVMS  ;SAVE ACTIVE REGISTER
2788 012160 000410          BR      VECMAP        ;GO FIND THE VECTOR NOW.
2789 012162 104402 010122      5$:  TYPE  ,MERR2         ;NOTIFY OPR THAT NO DZ11'S FOUND.
2790 012166 005000          CLR      R0           ;MAKE DATA DISPLAY ZERO
2791 012170 000000          HALT
2792 012172 000776          BR      -2           ;STOP THE SHOW
2793 012174 012716 012056      6$:  MOV    #3$, (SP)     ;DISABLE CONT. SW.
2794 012200 000002          RTI                ;ENTERED BY NON-EXISTENT TIME-OUT
2795                                     ;RETURN TO MAINSTREAM
2796 012202 012737 000340 000022  VECMAP: MOV    #340,@#22     ;SET IOT TRAP PRIORITY TO 7
2797 012210 012737 012326 000020      MOV    #4$,@#20     ;SET IOT TRAP VECTOR
2798 012216 012702 001500          MOV    #DZ.MAP,R2   ;SET SOFTWARE POINTER
2799 012222 012700 000300          MOV    #300,R0      ;FLOATING VECTORS START HERE.
2800 012226 012701 000302          MOV    #302,R1      ;PC OF IOT INSTR.
2801 012232 010120          1$:  MOV    R1,(R0)+     ;START FILLING VECTOR AREA
2802 012234 012721 000004          MOV    #4,(R1)+    ;WITH .+2; IOT
2803 012240 022021          CMP     (R0)+,(R1)+ ;ADD 2 TO R0+R1
2804 012242 020127 001000          CMP     R1,#1000   ;HAS THE VECTOR AREA BEEN EXCFEDED?
2805 012246 101771          BLOS   1$          ;BR IF MORE TO FILL
2806 012250 013704 001404          MOV    DZACTV,R4   ;STORE TEMPORARILY
2807 012254 000241          2$:  CLC
2808 012256 006004          ROR     R4          ;BRING OUT A BIT
2809 012260 103036          BCC     5$          ;BR IF ALL DONE
2810 012262 106427 000000          MTPS   #0          ;ZERO CPU PRIO
2811 012266 012772 040040 000000      MOV    #BIT14+BITS,@(R2)
2812 012274 011201          MOV    (R2),R1     ;GET CSR
2813 012276 112761 000200 000004      MOV    #BIT7,4(R1) ;SET THE ICR BIT!
2814                                     ;ATTEMPT TO FORCE AN INTERRUPT
2815                                     ;STALL
2816 012304 005200          INC     R0
2817 012306 001376          BNE     -2          ;FOR TIME TO INTERRUPT
2818 012310 012762 000300 000002      3$:  MOV    #300,2(R2)  ;NO INTERRUPT ASSUME 300 AND FIX DZ11 LATER
2819 012316 000005          RESET
2820 012320 062702 000014          ADD     #14,R2     ;POP SOFTWARE POINTER
2821 012324 000753          BR      2$          ;KEEP GOING
2822 012326 011662 000002 4$:  MOV    (SP),2(R2)  ;GET VECTOR ADDRESS
2823 012332 162762 000010 000002      SUB     #10,2(R2)  ;POINT BACK TO THE CORRECT VECTOR
2824 012340 042762 000007 000002      BIC     #7,2(R2)   ;CLEAR JUNK
2825 012346 022626          POP2SP
2826 012350 012716 012316          MOV    #3$, (SP)   ;POP IOT JUNK OFF STACK
2827 012354 000002          RTI                ;SET FOR RETURN
2828 012356 013737 001502 001304 5$:  MOV    DZVCO,$VECT1 ;COPY VECTOR OF FIRST DEVICE INTO ETABLE
2829 012364 012737 004772 000020      MOV    #.SCOPE,IOTVEC ;RESTORE THE SCOPE TRAP
2830 012372 000207          RTS                ;ALL DONE WITH 'AUTO SIZING'

```

```

2831
2832
2833
2834
2835
2836
2837
2838 012374 000004
2839 012376 012737 000001 001122
2840 012404 012737 012564 001360
2841 012412 012737 012552 000004
2842 012420 012737 000340 000006
2843 012426 012737 012434 001362
2844 012434 013700 002042
2845 012440 011001
2846 012442 000240
2847 012444 005010
2848 012446 000240
2849 012450 012737 012456 001362
2850 012456 013700 002046
2851 012462 011001
2852 012464 000240
2853 012466 005010
2854 012470 000240
2855 012472 012737 012500 001362
2856 012500 013700 002056
2857 012504 011001
2858 012506 000240
2859 012510 005010
2860 012512 000240
2861 012514 012737 012522 001362
2862 012522 013700 002062
2863 012526 011001
2864 012530 000240
2865 012532 005010
2866 012534 000240
2867 012536 012737 000006 000004
2868 012544 005037 000006
2869 012550 104400
2870 012552 011601
2871 012554 022626
2872 012556 104001
2873 012560 104401
2874 012562 000111
2875
2876
2877
2878
2879
2880
2881 012564 000004
2882 012566 012737 000002 001122
2883 012574 012737 012650 001360
2884 012602 013700 002042
2885 012606 012705 000020
2886 012612 010510

```

```

***** 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,$STSTM ;LOAD THE NUMBER OF THIS TEST
MOV #TST2,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV #5$,4 ;SET TRAP VECTOR
MOV #PR7,6 ;SET PRIORITY TO LEVEL 7
MOV #1$,LOCK ;SET RETURN IF SW09=11
1$: MOV DZCSR,R0 ;SET ADDRESS TO TEST
MOV (R0),R1 ;READ THE ADDRESS
NOP ;WASTE TIME
CLR (R0) ;WRITE THE ADDRESS
NOP ;WASTE TIME
MOV #2$,LOCK ;SET RETURN ADDRESS FOR SW09
2$: MOV DZRBUF,R0 ;SET ADDRESS TO TEST
MOV (R0),R1 ;READ THE ADDRESS
NOP ;
CLR (R0) ;WRITE THE ADDRESS
NOP ;WASTE TIME
MOV #3$,LOCK ;SET RETURN ADDRESS FOR SW09
3$: MOV DZTCR,R0 ;SET ADDRESS TO TEST
MOV (R0),R1 ;READ THE ADDRESS
NOP ;
CLR (R0) ;WRITE THE ADDRESS
NOP ;
MOV #4$,LOCK ;SET RETURN ADDRESS
4$: MOV DZMSR,R0 ;SET ADDRESS TO TEST
MOV (R0),R1 ;READ FROM ADDRESS
NOP ;
CLR (R0) ;WRITE THE ADDRESS
NOP ;
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.
SCOPE1 ;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,$STSTM ;LOAD THE NUMBER OF THIS TEST
MOV #TST3,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DZCSR,R0 ;SET POINTER
MOV #DCLR,R5 ;SET DCLR
MOV R5,(R0) ;WRITE DCLR INTO DZCSR

```



```
2943                :::* TEST 4
2944                :*****
2945 012742 000004    TST4: SCOPE
2946 012744 012737 000004 001122   MOV #4,$STSTM ;LOAD THE NUMBER OF THIS TEST
2947 012752 012737 013034 001360   MOV #TST5,NEXT ;POINT TO THE START OF THE NEXT TEST
2948 012760 013700 002042           MOV DZCSR,R0 ;GET BASE ADDRESS
2949 012764 012705 000040           MOV #MSENAB,R5 ;SET BIT
2950 012770 010510           MOV R5,(R0) ;SET SET IN DEVICE
2951 012772 011004           MOV (R0),R4 ;READ THE BIT FROM DEVICE
2952 012774 020504           CMP R5,R4 ;WAS BIT SET?
2953 012776 001401           BEQ 1$ ;BR IF YES
2954 013000 104002           ERROR 2 ;*BIT R/W FAILURE
2955 013002 040510 1$: BIC R5,(R0) ;CLEAR THE BIT.
2956 013004 011004           MOV (R0),R4 ;READ DEVICE
2957 013006 001404           BEQ 2$ ;BR IF BITS WERE CLEARED.
2958 013010 010546           MOV R5,-(SP) ;SAVE THE BIT
2959 013012 005005           CLR R5 ;SET EXPECTED RESULTS TO 0
2960 013014 104002           ERROR 2 ;*BIT FAILED TO CLEAR
2961 013016 012605           MOV (SP)+,R5 ;RESTORE THE BIT.
2962 013020 010510 2$: MOV R5,(R0) ;SET THE BIT AGAIN
2963 013022 104413           DEVICE.CLR ;ISSUE DEVICE CLEAR
2964 013024 011004           MOV (R0),R4 ;READ THE BIT.
2965 013026 001402           BEQ 3$ ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
2966 013030 005005           CLR R5 ;SET EXPECTED TO ZERO
2967 013032 104002           ERROR 2 ;*BIT NOT CLEARED BY DEVICE CLEAR
2968 013034
2969                :***** TEST 5 *****
2970                :*TEST TO VERIFY THAT BIT 'SILOEN' CAN
2971                :*BE SET. THEN VERIFY THAT BIT 'SILOEN' CAN
2972                :*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
2973                :*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
2974                :*CLEARED BY A 'DEVICE CLEAR'
2975                :::* TEST 5
2976                :*****
2977 013034 000004    TST5: SCOPE
2978 013036 012737 000005 001122   MOV #5,$STSTM ;LOAD THE NUMBER OF THIS TEST
2979 013044 012737 013126 001360   MOV #TST6,NEXT ;POINT TO THE START OF THE NEXT TEST
2980 013052 013700 002042           MOV DZCSR,R0 ;GET BASE ADDRESS
2981 013056 012705 010000           MOV #SILOEN,R5 ;SET BIT
2982 013062 010510           MOV R5,(R0) ;SET SET IN DEVICE
2983 013064 011004           MOV (R0),R4 ;READ THE BIT FROM DEVICE
2984 013066 020504           CMP R5,R4 ;WAS BIT SET?
2985 013070 001401           BEQ 1$ ;BR IF YES
2986 013072 104002           ERROR 2 ;*BIT R/W FAILURE
2987 013074 040510 1$: BIC R5,(R0) ;CLEAR THE BIT.
2988 013076 011004           MOV (R0),R4 ;READ DEVICE
2989 013100 001404           BEQ 2$ ;BR IF BITS WERE CLEARED.
2990 013102 010546           MOV R5,-(SP) ;SAVE THE BIT
2991 013104 005005           CLR R5 ;SET EXPECTED RESULTS TO 0
2992 013106 104002           ERROR 2 ;*BIT FAILED TO CLEAR
2993 013110 012605           MOV (SP)+,R5 ;RESTORE THE BIT.
2994 013112 010510 2$: MOV R5,(R0) ;SET THE BIT AGAIN
2995 013114 104413           DEVICE.CLR ;ISSUE DEVICE CLEAR
2996 013116 011004           MOV (R0),R4 ;READ THE BIT.
2997 013120 001402           BEQ 3$ ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
2998 013122 005005           CLR R5 ;SET EXPECTED TO ZERO
```



2999 013124 104002  
 3000 013126  
 3001  
 3002  
 3003  
 3004  
 3005  
 3006  
 3007  
 3008  
 3009 013126 000004  
 3010 013130 012737 000006 001122  
 3011 013136 012737 013220 001360  
 3012 013144 013700 002042  
 3013 013150 012705 000100  
 3014 013154 010510  
 3015 013156 011004  
 3016 013160 020504  
 3017 013162 001401  
 3018 013164 104002  
 3019 013166 040510  
 3020 013170 011004  
 3021 013172 001404  
 3022 013174 010546  
 3023 013176 005005  
 3024 013200 104002  
 3025 013202 012605  
 3026 013204 010510  
 3027 013206 104413  
 3028 013210 011004  
 3029 013212 001402  
 3030 013214 005005  
 3031 013216 104002  
 3032 013220

```

ERROR 2 ;*BIT NOT CLEARED BY DEVICE CLEAR
3$:
;***** TEST 6 *****
; *TEST TO VERIFY THAT BIT 'RIE' CAN
; *BE SET. THEN VERIFY THAT BIT 'RIE' CAN
; *BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
; *VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
; *CLEARED BY A 'DEVICE CLEAR'
;:* TEST 6
;*****
TST6: SCOPE
MOV #6,$STSTM ;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
2$: MOV (SP)+,R5 ;RESTORE THE BIT.
MOV R5,(R0) ;SET THE BIT AGAIN
DEVICE.CLR ;ISSUE DEVICE CLEAR
MOV (R0),R4 ;READ THE BIT.
BEQ 3$ ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
CLR R5 ;SET EXPECTED TO ZERO
ERROR 2 ;*BIT NOT CLEARED BY DEVICE CLEAR
3$:

```

3033  
 3034  
 3035  
 3036  
 3037  
 3038  
 3039  
 3040  
 3041 013220 000004  
 3042 013222 012737 000007 001122  
 3043 013230 012737 013312 001360  
 3044 013236 013700 002042  
 3045 013242 012705 040000  
 3046 013246 010510  
 3047 013250 011004  
 3048 013252 020504  
 3049 013254 001401  
 3050 013256 104002  
 3051 013260 040510  
 3052 013262 011004  
 3053 013264 001404  
 3054 013266 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,$STSTM ;LOAD THE NUMBER OF THIS TEST
MOV #TST10,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DZCSR,R0 ;GET BASE ADDRESS
MOV #TIE,R5 ;SET BIT
MOV R5,(R0) ;SET SET IN DEVICE
MOV (R0),R4 ;READ THE BIT FROM DEVICE
CMP R5,R4 ;WAS BIT SET?
BEQ 1$ ;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

```

```

3055 013270 005005          CLR      R5          ;SET EXPECTED RESULTS .0 0
3056 013272 104002          ERROR    2          ;*BIT FAILED TO CLEAR
3057 013274 012605          MOV      (SP)+,R5   ;RESTORE THE BIT.
3058 013276 010510          2$:     MOV      R5,(R0) ;SET THE BIT AGAIN
3059 013300 104413          DEVICE.CLR        ;ISSUE DEVICE CLEAR
3060 013302 011004          MOV      (R0),R4   ;READ THE BIT.
3061 013304 001402          BEQ     3$         ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
3062 013306 005005          CLR      R5          ;SET EXPECTED TO ZERO
3063 013310 104002          ERROR    2          ;*BIT NOT CLEARED BY DEVICE CLEAR
3064 013312
3065
3066          ;***** TEST 10 *****
3067          ;*THIS TESTS THAT ALL OF THE FOLLOWING
3068          ;*BITS CAN BE: SET, CLEARED, CLEARED BY 'DEVICE CLEAR ''
3069          ;*BITS TESTED ARE:
3070          ;* TCR0, TCR1, TCR2, TCR3, TCR4, TCR5, TCR6, TCR7
3071          ;:* TEST 10
3072          ;*****
3072 013312 000004          TST10: SCOPE
3073 013314 012737 000010 001122          MOV      #10,$TSTNM ;LOAD THE NUMBER OF THIS TEST
3074 013322 012737 013450 001360          MOV      #TST11,NEXT ;POINT TO THE START OF THE NEXT TEST
3075 013330 013700 002056          MOV      DZTCR,R0  ;SET DEVICE ADDRESS
3076 013334 012705 000001          MOV      #TCR0,R5  ;SET EXPECTED RESULTS
3077 013340 012737 013346 001362          MOV      #1$,LOCK  ;SET FOR SW09
3078 013346 010510          1$:     MOV      R5,(R0)  ;SET THE BIT
3079 013350 011004          MOV      (R0),R4   ;READ THE BIT FROM THE DEVICE
3080 013352 042704 177400          BIC      #^C<377>,R4 ;CLEAR HIGH BYTE
3081 013356 020504          CMP     R5,R4      ;WAS BIT OK?
3082 013360 001401          BEQ     2$         ;BR IF YES
3083 013362 104002          ERROR    2          ;*BIT FAILED TO SET.
3084 013364 040510          2$:     BIC      R5,(R0) ;CLEAR THE BIT
3085 013366 011004          MOV      (R0),R4   ;READ THE REGISTER
3086 013370 042704 177400          BIC      #^C<377>,R4 ;CLEAR HIGH BYTE
3087 013374 005704          TST     R4         ;BITS CLEAR?
3088 013376 001404          BEQ     3$         ;BR IF YES
3089 013400 010546          MOV      R5,-(SP)  ;SAVE GOOD RESULTS
3090 013402 005005          CLR      R5          ;SET EXPECTED TO 0
3091 013404 104002          ERROR    2          ;*REPORT BIT NOT CLEAR
3092 013406 012605          MOV      (SP)+,R5  ;RESTORE R5
3093 013410 010510          3$:     MOV      R5,(R0)  ;SET THE BIT AGAIN.
3094 013412 104413          DEVICE.CLR        ;ISSUE DEVICE CLEAR
3095 013414 011004          MOV      (R0),R4   ;READ THE REGISTER
3096 013416 042704 177400          BIC      #^C<377>,R4 ;CLEAR HIGH BYTE
3097 013422 005704          TST     R4         ;BITS CLEAR?
3098 013424 001404          BEQ     4$         ;BR IF YES
3099 013426 010546          MOV      R5,-(SP)  ;SAVE GOOD RESULTS
3100 013430 005005          CLR      R5          ;SET EXPECTED TO 0
3101 013432 104002          ERROR    2          ;*REPORT BIT NOT CLEAR
3102 013434 012605          MOV      (SP)+,R5  ;RESTORE R5
3103 013436 104401          4$:     SCOP1   ;LOCK ON BI,? SET SW09=1
3104 013440 106305          ASLB    R5          ;CHANGE TO NEXT BIT
3105 013442 001341          BNE     1$         ;CONTINUE TESTING
3106 013444 005037 001362          CLR      LOCK      ;MAKE SURE TIGHT LOOP IS CLEANED UP
3107
3108          ;***** TEST 11 *****
3109          ;*THIS TESTS THAT ALL OF THE FOLLOWING
3110          ;*BITS CAN BE: SET, CLEARED, CLEARED BY 'RESET INSTR *NOT* DEVICE CLEAR ''
          ;*BITS TESTED ARE:

```

```

3111          : * DTR0, DTR1, DTR2, DTR3, DTR4, DTR5, DTR6, DTR7
3112          : * THIS TEST IS NOT DONE IF MODULE IS ZOMA VERSION
3113          : : * TEST 11
3114          : : *****
3115 013450 000004 TST11: SCOPE
3116 013452 012737 000011 001122 MOV #11, $STSTNM ;LOAD THE NUMBER OF THIS TEST
3117 013460 012737 013632 001360 MOV #TST12, NEXT ;POINT TO THE START OF THE NEXT TEST
3118 013466 013700 002056 MOV DZTCR, R0 ;SET DEVICE ADDRESS
3119 013472 012705 000400 MOV #DTR0, R5 ;SET EXPECTED RESULTS
3120 013476 012737 013514 001362 MOV #1$, LOCK ;SET FOR SW09
3121 013504 105737 001414 TSTB EIAFLG ;ZOMA OR EIA
3122 013510 100001 BPL 1$ ;BR IF EIA
3123 013512 104400 ADVANCE ;EXIT TEST
3124 013514 010510 1$: MOV R5, (R0) ;SET THE BIT
3125 013516 011004 MOV (R0), R4 ;READ THE BIT FROM THE DEVICE
3126 013520 105004 CLRB R4 ;CLEAR LOW BYTE
3127 013522 020504 CMP R5, R4 ;WAS BIT OK?
3128 013524 001401 BEQ 2$ ;BR IF YES
3129 013526 104002 ERROR 2 ;*BIT FAILED TO SET.
3130 013530 040510 2$: BIC R5, (R0) ;CLEAR THE BIT
3131 013532 011004 MOV (R0), R4 ;READ THE REGISTER
3132 013534 105004 CLRB R4 ;CLEAR LOW BYTE
3133 013536 005704 TST R4 ;BITS CLEAR?
3134 013540 001404 BEQ 3$ ;BR IF YES
3135 013542 010546 MOV R5, -(SP) ;SAVE GOOD RESULTS
3136 013544 005005 CLR R5 ;SET EXPECTED TO 0
3137 013546 104002 ERROR 2 ;*REPORT BIT NOT CLEAR
3138 013550 012605 MOV (SP)+, R5 ;RESTORE R5
3139 013552 010510 3$: MOV R5, (R0) ;SET THE BIT AGAIN.
3140 013554 104413 DEVICE.CLR ;ISSUE DEVICE CLEAR
3141 013556 011004 MOV (R0), R4 ;READ THE REGISTER
3142 013560 105004 CLRB R4 ;CLEAR LOW BYTE
3143 013562 030510 BIT R5, (R0) ;WAS BIT CLEARED BY DEVICE.CLR?
3144 013564 001001 BNE 4$ ;BR IF NO (IT SHOULDN'T BE CLEAR)
3145 013566 104002 ERROR 2 ;*BIT CLEARED BY DEVICE.CLR
3146 013570 104401 4$: SCOP1 ;LOCK ON BIT? SW09=1
3147 013572 006305 ASL R5 ;CHANGE TO NEXT BIT
3148 013574 001347 BNE 1$ ;IF NOT DONE LOOP
3149 013576 012710 177400 MOV #177400, (R0) ;SET ALL DTR BITS
3150 013602 005005 CLR R5 ;CLEAR LOCATION FOR ERROR PRINTOUT
3151 013604 005227 000000 5$: INC #0 ;ACT DELAY LOOP FOR
3152 013610 001375 BNE 5$ ;RESET INSTRUCTION
3153 013612 000005 RESET ;ISSUE A BUS INIT
3154 013614 011004 MOV (R0), R4 ;READ REGISTER
3155 013616 105004 CLRB R4 ;CLEAR LOW BYTE
3156 013620 005704 TST R4 ;DTR BITS CLEAR?
3157 013622 001401 BEQ .+4 ;IF YES CONTINUE
3158 013624 104002 ERROR 2 ;IF NO PRINT FROR
3159 013626 005037 001362 CLR LOCK ;MAKE SURE TIGHT LOOP IS CLEANED UP
3160          : ***** TEST 12 *****
3161          : * THIS TEST PERFORMS RESET TESTING &
3162          : * TESTING OF WRITE ONLY OR READ ONLY BIT
3163          : * TEST BITS 'RDONE, BIT11, BIT10, BIT9, BIT8, BIT2, BIT1
3164          : * BIT0, SILOAL' ARE READ ONLY AND THAT TRDY IS
3165          : * ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.
3166          : *

```

CZDZA-FO  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 69  
CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0068

```

3167
3168
3169 013632 000004
3170 013634 012737 000012 001122
3171 013642 012737 013750 001360
3172 013650 013700 002042
3173 013654 005005
3174 013656 012710 027607
3175
3176 013662 011004
3177 013664 001401
3178 013666 104002
3179 013670 012710 100000
3180 013674 011004
3181 013676 001401
3182 013700 104002
3183 013702 012705 100000
3184 013706 005077 166140
3185 013712 052777 000001 166136
3186 013720 052710 000040
3187 013724 052705 000040
3188 013730 005002
3189 013732 011004
3190 013734 020504
3191 013736 001404
3192 013740 104414
3193 013742 005202
3194 013744 001372
3195 013746 104002
3196 013750
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207 013750 000004
3208 013752 012737 000013 001122
3209 013760 012737 014100 001360
3210 013766 104413
3211 013770 013700 002042
3212 013774 012710 177757
3213 014000 012705 050150
3214 014004 011004
3215 014006 020405
3216 014010 001401
3217 014012 104002
3218 014014 105010
3219 014016 105005
3220 014020 011004
3221 014022 020405
3222 014024 001401

```

```

::* TEST 12
:*****
TST12: SCOPE
MOV #12,$STSTM ;LOAD THE NUMBER OF THIS TEST
MOV #TST13,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV DZCSR,R0 ;SET ADDRESS TO R0
CLR R5 ;SET EXPECTED TO 0
MOV #RDONE+BIT11+BIT10+BIT9+BIT8+BIT2+BIT1+BIT0+SILOAL,(R0) ;WRITE THE BITS
MOV (R0),R4 ;READ BACK THE BITS
BEQ 1$ ;BR IF NONE ARE SET.
ERROR 2 ;*BITS WERE SET.
1$: MOV #TRDY,(R0) ;ATTEMPT TO WRITE TRDY
MOV (R0),R4 ;READ TRDY
BEQ 2$ ;BR IF NOT SET
ERROR 2 ;*
2$: MOV #TRDY,R5 ;SET EXPECTED BIT
CLR @DZLPR ;LOAD LINE 0
BIS #TCRO,@DZTCR ;SET TCR BIT
BIS #MSENAB,(R0) ;
BIS #MSENAB,R5 ;SET SCAN ENABLE
CLR R2 ;SET COUNTER TO ZERO
3$: MOV (R0),R4 ;READ THE REGISTER
CMP R5,R4 ;BIT SET?
BEQ 4$ ;BR IF YES
DELAY ;STALL TIME
INC R2 ;UPDATE COUNTER
BNE 3$ ;BR IF COUNTER NOT DONE.
4$: 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 VERIFY THAT A RESET WILL CLEAR THESE BITS
:THIS TEST ALSO CHECKS BYTE OPERATIONS ON THE CSR
::* TEST 13
:*****
TST13: SCOPE
MOV #13,$STSTM ;LOAD THE NUMBER OF THIS TEST
MOV #TST14,NEXT ;POINT TO THE START OF THE NEXT TEST
DEVICE.CLR
MOV DZCSR,R0 ;SET UP FOR ERROR MESSAGE
MOV #^C<DCLR>,(R0) ;TRY TO WRITE
MOV #TIE!SILOEN!RIE!MSENAB!MAINT,R5 ;MAKE EXPECTED
MOV (R0),R4 ;ACTUAL
CMP R4,R5 ;CMP EXPECTED VS ACTUAL
BEQ 1$ ;YES
ERROR 2 ;*NO
1$: CLRB (R0) ;CLEAR LOWER BYTE OF CSR
CLRB R5 ;SET EXPECTED
MOV (R0),R4 ;READ CSR BITS
CMP R4,R5 ;COMPARE ACTUAL TO EXPECTED
BEQ 3$ ;BRANCH IF SAME

```

```

3223 014026 104002          ERROR 2          ;OTHERWISE PRINT ERROR
3224 014030 012710 177757 3$:  MOV #*C<DCLR>,(R0) ;RESET CSR BITS
3225 014034 105077 166004  CLRB @HDZCSR      ;CLEAR HIGH BYTE OF CSR
3226 014040 012705 000150  MOV #RIE!MSENAB!MAINT,R5 ;SET R5 TO EXPECTED RESULTS
3227          ;READ CSR
3228 014044 011004          MOV (R0),R4        ;ACTUAL = EXPECTED?
3229 014046 020405          CMP R4,R5         ;BRANCH IF SAME
3230 014050 001401          BEQ 4$           ;OTHERWISE PRINTOUT ERROR
3231 014052 104002          ERROR 2          ;RESET CSR BITS
3232 014054 012710 177757 4$:  MOV #*C<DCLR>,(R0) ;SET R5 TO EXPECTED RESULTS
3233 014060 005005          CLR R5           ;DELAY TIMER FOR
3234 014062 005227 000000 5$:  INC #0          ;ACT-11 COMPATIBILITY
3235 014066 001375          BNE 5$          ;ISSUE BUS INIT
3236 014070 000005          RESET          ;READ CSR REGISTER
3237 014072 011004          MOV (R0),R4        ;BRANCH IF CSR IS CLEAR
3238 014074 001401          BEQ 2$           ;IF NOT PRINT ERROR
3239 014076 104002          ERROR 2
3240 014100          2$:
3241          ;***** TEST 14 *****
3242          ;*THIS TEST PERFORMS RESET TESTING AND
3243          ;*TESTING OF READ ONLY REGISTER DZRBUF
3244          ;*AND TESTING OF WRITE ONLY REGISTER DZLPR
3245          ;:* TEST 14
3246          ;*****
3247 014100 000004          TST14: SCOPE
3248 014102 012737 000014 001122  MOV #14,$STNM      ;LOAD THE NUMBER OF THIS TEST
3249 014110 012737 014170 001360  MOV #TST15,NEXT    ;POINT TO THE START OF THE NEXT TEST
3250 014116 104413          DEVICE.CLR        ;CLEAR DZ11
3251 014120 013700 002046          MOV DZRBUF,R0      ;SET UP FOR ERROR MESSAGE
3252 014124 011005          MOV (R0),R5        ;SET EXPECTED
3253 014126 012777 177777 165716  MOV #-1,@DZLPR     ;TRY TO WRITE ALL 1'S
3254 014134 011004          MOV (R0),R4        ;ACTUAL
3255 014136 042705 104000          BIC #DVALID!BIT11,R5 ;DITTO
3256 014142 020405          CMP R4,R5         ;CMP ACTUAL VS EXPECTED
3257 014144 001401          BEQ 1$           ;IF YES,GO CONTINUE PROCESSING
3258 014146 104002          ERROR 2          ;*ERROR- BIT PATTERN NOT CORRECT
3259 014150 010403          1$:  MOV R4,R3         ;GET A COPY OF THE ACTUAL BIT PATTERN
3260 014152 005103          COM R3           ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
3261 014154 010377 165672          MOV R3,@DZLPR     ;TRY TO WRITE
3262 014160 011004          MOV (R0),R4        ;ACTUAL
3263 014162 020405          CMP R4,R5         ;CMP ACTUAL VS EXPECTED
3264 014164 001401          BEQ 2$           ;IF YES, GET OUT OF THIS TEST
3265 014166 104002          ERROR 2          ;*NO
3266 014170          2$:
3267          ;***** TEST 15 *****
3268          ;*THIS TEST PERFORMS RESET TESTING AND
3269          ;*TESTING OF READ ONLY REGISTER DZMSR
3270          ;*AND TESTING OF WRITE ONLY REGISTER DZTDR
3271          ;:* TEST 15
3272          ;*****
3273 014170 000004          TST15: SCOPE
3274 014172 012737 000015 001122  MOV #15,$STNM      ;LOAD THE NUMBER OF THIS TEST
3275 014200 012737 014254 001360  MOV #TST16,NEXT    ;POINT TO THE START OF THE NEXT TEST
3276 014206 104413          DEVICE.CLR        ;CLEAR DZ11
3277 014210 013700 002062          MOV DZMSR,R0      ;SET UP FOR ERROR MESSAGE
3278 014214 011005          MOV (R0),R5        ;SET EXPECTED

```



```

3335 014402 150405 BISB R4,R5 :
3336 014404 000305 SWAB R5 :
3337 014406 150405 BISB R4,R5 :
3338 014410 150277 165444 BISB R2,@HDZTCR :SET DTR
3339 014414 104414 DELAY :CABLE DELAY
3340 014416 011004 MOV (R0),R4 :READ MSR REGISTER
3341 014420 020504 CMP R5,R4 :OK?
3342 014422 001401 BEQ 6$ :YES
3343 014424 104002 ERROR 2 :*ERROR IN RING OR CARRIER
3344 014426 140277 165426 6$: BICB R2,@HDZTCR :CLEAR DTR
3345 014432 104414 DELAY :CABLE DELAY
3346 014434 011004 MOV (R0),R4 :READ MSR
3347 014436 001402 BEQ 7$ :BR IF THEY CLEARED
3348 014440 005005 CLR R5 :SET EXPECTED TO 0
3349 014442 104002 ERROR 2 :*BITS NOT CLEARED
3350 014444 104401 7$: SCOP1 :LOCK ON SIGNAL?
3351 014446 000741 BR 2$ :CONTINUE TEST
3352
3353 :***** TEST 17 *****
3354 :*TEST TO VERIFY THAT IF IN 'EXTERNAL'
3355 :*MODE; SETTING DTR FOR SELECTED LINES
3356 :*WILL BRING UP 'CARRIER' AND 'RING'
3357 :*FOR THAT SAME LINE. NOTE: IF YOU HAVE
3358 :*SELECTED MODE AS 'EXTERNAL'; THE H325 TEST CONNECTOR
3359 :*MUST BE USED ON ALL SPECIFIED LINES.
3360 :*LINES MAY BE SPECIFIED BY SWR03=1
3361 :*AND SWR00=1 AT START TIME OR ALTERING
3362 :*STATUS MAP.
3363 :::* TEST 17
3364 :*****
3365 014450 000004 TST17: SCOPE
3366 014452 012737 000017 001122 MOV #17,$TSTNM ;LOAD THE NUMBER OF THIS TEST
3367 014460 012737 014606 001360 MOV #TST20,NEXT ;POINT TO THE START OF THE NEXT TEST
3368 014466 012737 014522 001362 MOV #3$,LOCK ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3369 014474 105737 001370 TSTB MODE ;EXTERNAL?
3370 014500 100401 BMI 2$ ;BR IF YES
3371 014502 104400 1$: ADVANCE ;EXIT TEST
3372 014504 105737 001414 2$: TSTB EIAFLG ;YOU BETTER BE IN
3373 014510 100774 BMI 1$ ;EIA MODE FOR THIS TEST.
3374 014512 013700 002062 MOV DZMSR,R0 ;SET REGISTER
3375 014516 012702 000001 MOV #1,R2 ;SET LINE POINTER
3376 014522 130237 001354 3$: BITB R2,LINE ;LINE SELECTED?
3377 014526 001003 BNE 5$ ;BR IF YES
3378 014530 106302 4$: ASLB R2 ;NEXT LINE
3379 014532 103373 BCC 3$ ;CONTINUE TEST
3380 014534 104400 ADVANCE ;ADVANCE THIS TEST
3381 014536 005005 5$: CLR R5 ;SET EXPECTED
3382 014540 150205 BISB R2,R5 :
3383 014542 000305 SWAB R5 :
3384 014544 150205 BISB R2,R5 :
3385 014546 150277 165306 BISB R2,@HDZTCR :SET DTR
3386 014552 104414 DELAY :CABLE DELAY
3387 014554 011004 MOV (R0),R4 :READ MSR
3388 014556 020504 CMP R5,R4 :BITS OK?
3389 014560 001401 BEQ 6$ ;BR IF YES
3390 014562 104002 ERROR 2 :CARRIER OR RING ERROR

```

3391	014564	140277	165270
3392	014570	104414	
3393	014572	011004	
3394	014574	001402	
3395	014576	005005	
3396	014600	104002	
3397	014602	104401	
3398	014604	000751	
3399			

```

6$: BICB R2,@HDZTCR ;CLEAR DTR
    DELAY ;CABLE DELAY
    MOV (R0),R4 ;READ MSR
    BEQ 7$ ;BR IF BITS CLEARED
    CLR R5 ;CLEAR EXPECTED LOC.
    ERROR 2 ;BITS NOT CLEARED.
7$: SCOP1 ;LOCK ON LINE?
    BR 4$ ;CONTINUE TEST

```

3400				
3401				
3402				
3403				
3404				
3405				
3406				
3407	014606	000004		
3408	014610	012737	000020	001122
3409	014616	012737	014732	001360
3410	014624	104413		
3411	014626	013700	002042	
3412	014632	012705	100040	
3413	014636	005037	001372	
3414	014642	012702	000001	
3415	014646	130237	001364	
3416	014652	001420		
3417	014654	050277	165176	
3418	014660	052710	000040	
3419	014664	005004		
3420	014666	032710	100000	
3421	014672	001004		
3422	014674	104414		
3423	014676	005204		
3424	014700	001372		
3425	014702	104003		
3426	014704	011004		
3427	014706	020405		
3428	014710	001401		
3429	014712	104002		
3430	014714	062705	000400	
3431	014720	104413		
3432	014722	005237	001372	
3433	014726	106302		
3434	014730	103346		
3435	014732			

```

:***** TEST 20 *****
:* THIS TEST VERIFIES THAT TRDY 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
:*****
TST20: SCOPE
    MOV #20,$TSTNM ;LOAD THE NUMBER OF THIS TEST
    MOV #TST21,NEXT ;POINT TO THE START OF THE NEXT TEST
    DEVICE.CLR ;ISSUE A 'DEVICE CLEAR' (RESET)
    MOV DZCSR,R0 ;SET POINTER
    MOV #MSENAB!TRDY,R5 ;START THE EXPECTED LINE NUMBER AT 0
    CLR SAVLIN ;SET UP FOR ERROR PRINTOUTS
    MOV #1,R2 ;USING R2 AS A BIT POINTER, POINT TO LINE 0
1$: BITB R2,LINE ;IS THIS LINE SELECTED?
    BEQ 5$ ;IF NO, SKIP THE STARTUP
2$: BIS R2,@DZTCR ;SET THE GO BIT FOR THIS LINE
    BIS #MSENAB,(R0) ;START THE SCANNER
    CLR R4 ;SET FOR DELAY
3$: BIT #TRDY,(R0) ;TX READY?
    BNE 4$ ;BR IF YES
    DELAY ;DELAY
    INC R4 ;COUNTER
    BNE 3$ ;BR IF <>0!
    ERROR 3 ;*TX NOT READY!
4$: MOV (R0),R4 ;GET THE LINE POINTED TO BY THE SCANNER
    CMP R4,R5 ;IS THE LINE NUMBER WHAT IT SHOULD BE?
    BEQ 5$ ;IF YES,GO WORK ON THE NEXT LINE
    ERROR 2 ;*LINE NUMBER DID NOT MATCH TCR BIT
5$: ADD #400,R5 ;POINT TO THE NEXT EXPECTED LINE
    DEVICE.CLR ;ISSUE A 'DEVICE CLEAR' (RESET)
    INC SAVLIN ;ADJUST FOR NEXT LINE
    ASLB R2 ;POINT TO THE NEXT LINE.ARE ALL LINES TESTED?
    BCC 1$ ;IF NOT, GO DO THE NEXT LINE
6$:

```

3436				
3437				
3438				
3439				
3440				
3441				
3442				
3443				
3444				
3445				
3446				

```

:***** TEST 21 *****
:*TEST TO TRANSMIT ONE CHAR AND
:*RECEIVE ONE CHAR ON ONE LINE
:*AT A TIME. THE CHAR IS '252' AND
:*ALL SELECTED LINES WILL BE TURNED ON
:*ONE AT A TIME. THIS IS THE FIRST TIME ANY
:*DATA IS CHECKED IN THE RECEIVER.
:*USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
:*WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.
:;* TEST 21
:*****

```





```

3503 015176 104401          13$: SCOP1          ;CHECK TO SEE IF SWITCH NINE IS SET
3504 015200 040277 164652  14$: BIC      R2,@DZTCR ;CLEAR TCR BIT FOR THAT LINE.
3505 015204 005237 001372  15$: INC      SAVLIN    ;INC EXPECTED LINE
3506 015210 013700 001372  MOV      SAVLIN,R0    ;SET UP CHARACTER OFFSET
3507 015214 006300          ASL      R0           ;MAKE THE OFFSET A POWER OF TWO
3508 015216 106302          ASLB     R2           ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
3509 015220 103302          BCC     3$           ;IF NO, GO AROUND AGAIN FOR NEXT LINE
3510 015222 005003          CLR     R3           ;THIS CODE HAS BEEN INSERTED
3511 015224 104414          17$: DELAY          ;TO DETECT A PROBLEM FOUND IN FAULT
3512 015226 105203          INCB    R3           ;INSERTION. IF AN ERROR OCCURS MORE
3513 015230 001375          BNE     17$         ;THAN ONE WORD WAS RECIEVED ON
3514 015232 032777 000200 164602 BIT      #RDONE,@DZCSR ;LINE 7.
3515 015240 001401          BEQ     18$         ;
3516 015242 104020          ERROR  20           ;
3517 015244 104400          18$: ADVANCE          ;GO TO NEXT TEST
3518
3519 ;TIGHT SCOPE LOOP FOR THIS TEST. LOOP TRANSMITS CHARACTERS ONLY
3520
3521 015246 032777 100000 164566 16$: BIT      #TRDY,@DZCSR ;IS TRANSMITTER READY?
3522 015254 001774          BEQ     16$         ;IF NOT, WAIT FOR IT
3523 015256 112777 000252 164602 MOVB     #252,@DZTDR ;LOAD THE CHARACTER
3524 015264 104401          SCOP1          ;LOOP AGIN IF SW09=1
3525 015266 000744          BR      14$         ;OTHERWISE, GO PICK UP THE TEST NORMALLY
3526
3527 ;***** TEST 22 *****
3528 ;* THIS TEST PROVES THAT THE TRANSMITTER TRANSMITS
3529 ;*CHARACTERS (FLAG MODE)AND THE RECEIVER RECEIVES (FLAG MODE)
3530 ;*(ONE LINE AT A TIME BASED UPON VALID LINES)
3531 ;*THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED
3532 ;:* TEST 22
3533 ;*****
3534 015270 000004          TST22: SCOPE
3535 015272 012737 000022 001122 MOV      #22,$TSTNM ;LOAD THE NUMBER OF THIS TEST
3536 015300 012737 015616 001360 MOV      #TST23,NEXT ;POINT TO THE START OF THE NEXT TEST
3537 015306 012737 015422 001362 MOV      #4$,LOCK    ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3538 015314 104417          DCLASm          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3539 015316 013701 001366          MOV      PAR,R1     ;PICK UP PARAMETERS
3540 015322 012702 000001          MOV      #1,R2     ;PICK UP INIT POINTER
3541 015326 030237 001364          1$: BIT      R2,LINE ;SHOULD THIS LINE BE SET UP ?
3542 015332 001402          BEQ     2$         ;NO
3543 015334 010177 164512          MOV      R1,@DZLPR ;SET UP LINE PARAMETERS
3544 015340 005201          2$: INC     R1     ;POSITION POINTER TO THE NEXT LINE
3545 015342 106302          ASLB    R2         ;GOT 'EM ALL ?
3546 015344 103370          BCC     1$         ;IF NO, GO SET UP THE NEXT LINE
3547 015346 005037 001372          CLR     SAVLIN    ;CLEAR LINE # INDICATOR
3548 015352 012700 001422          MOV      #TDO,R0   ;POINT TO THE DATA AREA
3549 015356 005020          CLR     (R0)+     ;CLEAR A DATA WORD
3550 015360 022700 001462          CMP     #STOP,R0  ;FINISHED ?
3551 015364 001374          BNE     -6         ;NO
3552 015366 005000          CLR     R0        ;CLEAR OFFSET
3553 015370 013737 002046 001400 MOV      DZRBUF,REGIST ;SAVE FOR ERROR MSG
3554 015376 012702 000001          MOV      #1,R2     ;LINE POINTER
3555 015402 052777 000040 164432 BIS      #MSENAB,@DZCSR ;START SCANNER
3556 015410 030237 001364          3$: BIT      R2,LINE ;VALID LINE ?
3557 015414 001465          BEQ     14$         ;NO SET UP NEXT LINE
3558 015416 010277 164434          MOV      R2,@DZTCR ;SET TCR BIT

```

```

3559 015422 032777 000200 164412 4$: BIT #RDONE,@DZCSR ;IS REC DONE = 0 ?
3560 015430 001401 BEQ 5$ ;IF YES, ALLOW TIME FOR TRDY TO SET
3561 015432 104020 ERROR 20 ;*REC DONE SHOULD = 0
3562 015434 005005 5$: CLR R5
3563 015436 032777 100000 164376 6$: BIT #TRDY,@DZCSR
3564 015444 001004 BNE 7$
3565 015446 104414 DELAY
3566 015450 105205 INCB R5
3567 015452 001371 BNE 6$
3568 015454 104003 ERROR 3 ;*TRDY FAILED TO SET.
3569 015456 116077 001422 164402 7$: MOVB TD0(R0),@DZTDR ;LOAD CHARACTER
3570 015464 013705 001372 MOV SAVLIN,R5 ;MAKE EXPECTED LINE #
3571 015470 105737 001371 TSTB MODE+1 ;IS THIS TEST IN STAGGERED MODE.
3572 015474 001406 BEQ 10$ ;IF NOT, SKIP STAGGERED SETUP
3573
3574 ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3575
3576 015476 006205 ASR R5 ;GET THE LAST BIT INTO THE CARRY BIT
3577 015500 103402 BCS 8$ ;IF IT IS SET, GO CLEAR IT
3578 015502 000261 SEC ;IF IT IS CLEAR SET IT HERE
3579 015504 000401 BR 9$ ;SKIP THE CLEARING
3580 015506 000241 8$: CLC ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3581 015510 006105 9$: ROL R5 ;GET THE NEW BIT BACK INTO R5
3582 015512 000305 10$: SWAB R5 ;MOVE THE LINE NUMBER TO THE UPPER BYTE
3583 015514 156005 001422 BISB TD0(R0),R5 ;ADD CHARACTER
3584 015520 052705 100000 BIS #DVALID,R5 ;ADD DATA VALID
3585 015524 005003 CLR R3
3586 015526 032777 000200 164306 11$: BIT #RDONE,@DZCSR
3587 015534 001004 BNE 12$
3588 015536 104414 DELAY
3589 015540 005203 INC R3
3590 015542 001371 BNE 11$
3591 015544 104004 ERROR 4 ;*RDONE FAILED TO SET!
3592 015546 017704 164274 12$: MOV @DZRBUF,R4 ;LOAD THE VALUE ACTUALLY RECEIVED
3593 015552 020405 CMP R4,R5 ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
3594 015554 001401 BEQ 13$ ;IF YES, GO DO THE NEXT LINE
3595 015556 104006 ERROR 6 ;*NO DATA/CONTENTS DID NOT COMPARE
3596 015560 104401 13$: SCOP1 ;CHECK TO SEE IF SWITCH NINE IS SET
3597 015562 105260 001422 INCB TD0(R0) ;INCREMENT BINARY PATTERN FOR THIS LINE
3598 015566 001315 BNE 4$ ;GO 'ROUND AGAIN FOR NEXT CHARACTER
3599 015570 040277 164262 14$: BIC R2,@DZTCR ;CLEAR TCR BIT FOR THAT LINE.
3600 015574 005237 001372 15$: INC SAVLIN ;INC EXPECTED LINE
3601 015600 013700 001372 MOV SAVLIN,R0 ;SET UP CHARACTER OFFSET
3602 015604 006300 ASL R0 ;MAKE THE OFFSET A POWER OF TWO
3603 015606 106302 ASLB R2 ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
3604 015610 103277 BCC 3$ ;IF NO, GO AROUND AGAIN FOR NEXT LINE
3605 015612 005037 001362 CLR LOCK ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST
3606
3607
3608 ;***** TEST 23 *****
3609 ;*THIS TEST WILL PROVE THAT EACH RECEIVING LINE CAN
3610 ;*BE DISABLED BY SETTING THE RCVON BIT TO ZERO
3611 ;*FOR EACH LINE IN THE LPR REGISTER. IT ALSO
3612 ;*VERIFIES THAT MASTER CLEAR WILL ZERO DVALID FOR
3613 ;*CHARACTERS STORED IN THE SILO.
3614 ;:* TEST 23

```

CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052) 26-NOV-79 11:03

26-NOV-79 11:06 PAGE 77

CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0076

```

36:5
3616 015616 000004
3617 015620 012737 000023 001122
3618 015626 012737 016150 001360
3619 015634 105037 001420
3620 015640 005037 001372
3621 015644 104417
3622 015646 013701 001366
3623 015652 042701 010000
3624 015656 012702 000001 1$:
3625 015662 010177 164164 2$:
3626 015666 005201
3627 015670 106302
3628 015672 103373
3629 015674 012701 000252
3630 015700 013702 001364
3631 015704 010277 164146
3632 015710 052777 000040 164124
3633 015716 005005 3$:
3634 015720 005777 164116 4$:
3635 015724 100404
3636 015726 104414
3637 015730 005205
3638 015732 001372
3639 015734 104003
3640 015736 117705 164102 5$:
3641 015742 012703 000001
3642 015746 042705 177770
3643 015752 001403
3644 015754 106303 20$:
3645 015756 005305
3646 015760 001375
3647 015762 030302 21$:
3648 015764 001007
3649 015766 140377 164064
3650 015772 001351
3651 015774 105737 001420
3652 016000 001040
3653 016002 000404
3654 016004 110177 164056 6$:
3655 016010 040302
3656 016012 000741
3657 016014 005077 164036 7$:
3658 016020 005005
3659 016022 104414 8$:
3660 016024 005205
3661 016026 001375
3662 016030 105777 164006
3663 016034 100003
3664 016036 005037 001372
3665 016042 104020
3666 016044 017704 163776 10$:
3667 016050 100007
3668 016052 000304
3669 016054 042704 177770
3670 016060 010437 001372

```

```

:*****
TST23: SCOPE
MOV #23,$TSTNM ;LOAD THE NUMBER OF THIS TEST
MOV #TST24,NEXT ;POINT TO THE START OF THE NEXT TEST
CLRB DONFLG ;INITIALIZE FOR FIRST TEST LOOP
CLR SAVLIN ;ZERO LINE NO. FOR ERROR REPORT
DCLASM ;EXECUTE MASTER CLEAR
MOV PAR,R1 ;STORE DEFAULT PARAMETERS
BIC #RCVON,R1 ;CLEAR RCVON BIT
MOV #1,R2 ;INIT LINE POINTER
MOV R1,@DZLPR ;LOAD LINE PARAMETER REGISTER
INC R1 ;SET R1 FOR NEXT LINE
ASLB R2 ;SHIFT R2 TO NEXT LINE
BCC 2$ ;ALL LINES LOADED?
MOV #252,R1 ;LOAD TRANSMITTING CHARACTER
MOV LINE,R2 ;COPY ACTIVE LINE BITS
MOV R2,@DZTCR ;LOAD TCR BITS
BIS #MSENAB,@DZCSR ;SET SCANNER
CLR R5 ;INIT DELAY COUNTER
TST @DZCSR ;TRDY SET?
BMI 5$ ;IF YES BRANCH
DELAY ;IF NOT THEN WAIT
INC R5 ;INCREMENT DELAY COUNTER
BNE 4$ ;DELAY DONE?
ERROR 3 ;IF YES TRDY FAILED TO SET
MOV @HDZCSR,R5 ;MOVE LINE NO. INTO R5
MOV #1,R3 ;INIT TCR POINTER
BIC #^C<?>,R5 ;ISOLATE LINE NO.
BEQ 21$ ;IF LINE 0 GO TEST TRANSM. FLAG
ASLB R3 ;POINT R3 TO NEXT TCR BIT
DEC R5 ;DECREMENT R5 UNTIL R3 POINTS
BNE 20$ ;TO CORRECT TCR BIT
BIT R3,R2 ;HAS THIS LINE BEEN SERVICED?
BNE 6$ ;IF NOT GO SEND CHARACTER
BICB R3,@DZTCR ;IF YES CLEAR TCR BIT
BNE 3$ ;IF MORE LINES SET BRANCH
TSTB DONFLG ;IF ALL LOADED IS THIS SECOND PASS
BNE 12$ ;IF YES BRANCH TO SECOND PART OF TEST
BR 7$ ;OTHERWISE CONTINUE WITH FIRST PART
MOV R1,@DZTDR ;TRANSMIT CHARACTER
BIC R3,R2 ;CLEAR FLAG FOR THIS LINE
BR 3$ ;GO WAIT FOR NEXT LINE
CLR @DZTCR ;CLEAR TCR BITS
CLR R5 ;CLEAR DELAY COUNTER
DELAY ;WAIT FOR LAST CHARACTER
INC R5 ;INCREMENT DELAY COUNTER
BNE 8$ ;IF NOT FINISHED CONTINUE WAITING
TSTB @DZCSR ;RDONE BIT SET?
BPL 10$ ;IF NO CONTINUE
CLR SAVLIN ;IF YES SET LINE NO. TO ZERO
ERROR 20 ;AND PRINT ERROR
MOV @DZRBUF,R4 ;READ SILO
BPL 11$ ;IF DVALID IS ZERO BRANCH
SWAB R4 ;IF SET THEN
BIC #^C<?>,R4 ;ISOLATE LINE NO. IN R4
MOV R4,SAVLIN ;SET SAVLIN FOR ERROR REPORT

```

```

3671 016064 104017          ERROR 17          ;DATA VALID SHOULD NOT BE SET
3672 016066 000766          BR      10$       ;GO READ SILO AGAIN
3673 016070 105237 001420 11$: INCB   DONFLG    ;PREPARE FOR SECOND PART OF TEST
3674 016074 013701 001366    MOV    PAR,R1     ;MOVE DEFAULT PARAMETERS TO R1
3675 016100 000666          BR      1$        ;GO LOAD LPR REGISTER
3676 016102 005005          CLR     R5        ;INIT DELAY LO ER
3677 016104 104414          12$:  DELAY      ;WAIT FOR LAST CHARACTER
3678 016106 005205          13$:  INC      R5        ;TO BE RECEIVED
3679 016110 001375          BNE    13$       ;DELAY FINISHED?
3680 016112 104413          DEVICE.CLR     ;IF YES EXECUTE MASTER CLEAR
3681 016114 000240          NOP
3682 016116 000240          NOP
3683 016120 105777 163716    TSTB   @DZCSR    ;R DONE SET?
3684 016124 100003          BPL    14$      ;IF NOT BRANCH
3685 016126 005037 001372    CLR    SAVLIN   ;IF YES THEN PRINT OUT
3686 016132 104020          ERROR 20        ;REPORT
3687 016134 017704 163706 14$:  MOV    @DZRBUF,R4 ;READ SILO
3688 016140 100003          BPL    15$      ;DATA VALID SET?
3689 016142 005037 001372    CLR    SAVLIN   ;IF YES THEN PRINT OUT
3690 016146 104017          ERROR 17        ;ERROR REPORT
3691 016150
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703 016150 000004          ;***** TEST 24 *****
3704 016152 012737 000024 001122 TST24: SCOPE
3705 016160 012737 016426 001360    MOV    #24,$STNM ;LOAD THE NUMBER OF THIS TEST
3706 016166 012737 016264 001362    MOV    #TST25,NEXT ;POINT TO THE START OF THE NEXT TEST
3707 016174 005737 001370    MOV    #3$,LOCK   ;SET FOR LOOP
3708 016200 001510          TST    MODE      ;ARE WE RUNNING IN INTERNAL MODE?
3709 016202 104417          BEQ    12$       ;IF SO, SKIP THIS TEST
3710 016204 013701 001366    DCLASM          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3711 016210 052701 000300    MOV    PAR,R1    ;PICK UP PARAMETERS
3712 016214 012700 000001    BIS    #ODDPAR!PARITY,R1 ;FORCE ODD PARITY
3713 016220 030037 001364    MOV    #1,R0     ;PICK UP INIT POINTER
3714 016224 001402          1$:  BIT    R0,LINE ;SHOULD THIS LINE BE SET UP ?
3715 016226 010177 163620    BEQ    2$        ;IF NOT,DON'T SET IT UP
3716 016232 005201          2$:  MOV    R1,@DZLPR ;OTHERWISE, SET UP LINE PARAMETERS
3717 016234 106300          INC    R1
3718 016236 103370          ASLB   R0        ;GOT 'EM ALL ?
3719 016240 005037 001372    BCC   1$        ;NO
3720 016244 012702 000001    CLR    SAVLIN   ;CLEAR LINE #
3721 016250 052777 000040 163564    MOV    #1,R2    ;LINE POINTER
3722 016256 013737 002046 001400    BIS    #MSENAB,@DZCSR ;SET MASTER SCAN ENABLE
3723 016264 030237 001364          3$:  MOV    DZRBUF,REGIST ;SAVE FOR ERRR MESSAGE
3724 016270 001446          BIT    R2,LINE
3725 016272 010277 163560    BEQ   11$
3726 016276 110277 163566    MOV    R2,@DZTCR ;SET TCR BIT
          MOVB   R2,@HDZTDR ;SET BREAK BIT

```

```

3727 016302 112777 000377 163556 4$:   MOVB   #377,@DZTDR   ;LOAD CHARACTER
3728 016310 013705 001372           MOV    SAVLIN,R5    ;MAKE EXPECTED DATA
3729 016314 105737 001371           TSTB  MODE+1       ;IS THIS TEST IN STAGGERED MODE?
3730 016320 001406           BEQ   7$           ;IF NOT, SKIP STAGGERED SETUP
3731
3732                               ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3733
3734 016322 006205           ASR   R5           ;GET THE LAST BIT INTO THE CARRY BIT
3735 016324 103402           BCS  5$           ;IF IT IS SET, GO CLEAR IT
3736 016326 000261           SEC                    ;IF IT IS CLEAR SET IT HERE
3737 016330 000401           BR   6$           ;SKIP THE CLEARING
3738 016332 000241           5$:  CLC                    ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3739 016334 006105           6$:  ROL   R5       ;GET THE NEW BIT BACK INTO R5
3740 016336 000305           7$:  SWAB  R5       ;PUT LINE NUMBER IN UPPER BYTE
3741 016340 052705 130000           BIS  #DVALID.PARER!FRMERR,R5 ;ADD EXPECTED
3742 016344 005004           CLR  R4
3743 016346 032777 000200 163466 8$:  BIT  #RDONE,@DZCSR
3744 016354 001004           BNE  9$
3745 016356 104414           DELAY
3746 016360 005204           INC  R4
3747 016362 001371           BNE  8$
3748 016364 104004           ERROR 4           ;*RDONE FAILED TO SET.
3749 016366 017704 163454 9$:  MOV  @DZRBUF,R4   ;ACTUAL
3750 016372 020405           CMP  R4,R5       ;CMP ACTUAL VS EXPECTED. DO THEY MATCH?
3751 016374 001401           BEQ  10$        ;IF YES, GO CLEAN UP
3752 016376 104006           ERROR 6           ;*DATA/CONTENTS FAILED TO COMPARE
3753 016400 105077 163464 10$: CLRB @HDZTDR   ;CLEAR BREAK BITS
3754 016404 104401           SCOP1           ;LOOP?
3755 016406 005237 001372 11$: INC  SAVLIN     ;INC LINE #
3756 016412 040277 163440           BIC  R2,@DZTCR  ;CLEAR TCR BIT
3757 016416 106302           ASLB R2
3758 016420 103321           BCC  3$
3759 016422 005037 001362 12$: CLR  LOCK      ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST
3760                               ;***** TEST 25 *****
3761                               ;* THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
3762                               ;*WHILE THE PROCESSOR STATUS IS SET EXACTLY
3763                               ;*TO WHAT THE DZ11 PRIORITY IS SET TO.
3764                               ;*DEFAULT PRIORITY IS AT 5 (240).
3765                               ;:* TEST 25
3766                               ;*****
3767 016426 000004           TS:25: SCOPE
3768 016430 012737 000025 001122           MOV  #25,$STNM   ;LOAD THE NUMBER OF THIS TEST
3769 016436 012737 016736 001360           MOV  #TST26,NEXT ;POINT TO THE START OF THE NEXT TEST
3770 016444 104417           DCLASM           ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3771 016446 013701 001366           MOV  PAR,R1     ;PICK UP PARAMETERS
3772 016452 012702 000001           MOV  #1,R2     ;PICK UP INIT POINTER
3773 016456 030237 001364 1$:  BIT  R2,LINE   ;SHOULD THIS LINE BE SET UP ?
3774 016462 001402           BEQ  2$
3775 016464 010177 163362           MOV  R1,@DZLPR ;SET UP LINE PARAMETERS
3776 016470 005201 2$:  INC  R1       ;POSITION POINTER TO THE NEXT LINE
3777 016472 106302           ASLB R2       ;GOT 'EM ALL ?
3778 016474 103370           BCC  1$
3779 016476 005037 001372           CLR  SAVLIN    ;CLEAR LINE # INDICATOR
3780 016502 106437 027060           MTPS @DZPRT    ;SET CPU STATUS TO DZ11 PRIO,
3781 016506 113777 001364 163342 3$: MOVB LINE,@DZTCR ;ENABLE THE VALID LINES
3782 016514

```

```

3783 016514 012777 016604 163354      MOV      #6$,@DZTIV      ;SET UP THE TRANSMITTER INTERRUPT VECTOR
3784 016522 012777 016612 163342      MOV      #7$,@DZRIV      ;SET UP THE RECEIVER INTERRUPT VECTOR
3785 016530 013777 027060 163336      MOV      DZPRT,@DZ RIS    ;SET THE INTERRUPT VECTOR STATUS
3786 016536 013777 027060 163334      MOV      DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
3787 016544 052777 040040 163270      BIS      #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
3788 016552 005005                CLR      R5
3789 016554 032777 100000 163260 4$:  BIT      #TRDY,@DZCSR
3790 016562 001403                BEQ      5$
3791 016564 000240                NOP
3792 016566 000240                NOP
3793 016570 000412                BR       8$
3794 016572 104414                5$:  DELAY
3795 016574 005205                INC      R5
3796 016576 001366                BNE     4$
3797 016600 104003                ERROR   3                ;*TRDY NOT SET!
3798 016602 000405                BR       8$
3799 016604 104010                6$:  ERROR   10            ;*TRANSMITTER SHOULD NOT INTERRUPT
3800 016606 022626                CMP     (SP)+,(SP)+      ;POP FOR FAKE RTI
3801 016610 000402                BR       8$                ;CONTINUE TEST
3802 016612 104012                7$:  ERROR   12            ;*RECEIVER SHOULD NOT INTERRUPT
3803 016614 022626                CMP     (SP)+,(SP)+      ;POP FOR FAKE RTI
3804 016616 042777 040000 163216 8$:  BIC      #TIE,@DZCSR      ;RESET TRANSMITTER INTERRUPT ENABLE
3805 016624 012777 016722 163244      MOV      #11$,@DZTIV     ;SET UP THE TRANSMITTER INTERRUPT VECTOR
3806 016632 012777 016730 163232      MOV      #12$,@DZRIV     ;SET UP THE RECEIVER INTERRUPT VECTOR
3807 016640 013777 027060 163226      MOV      DZPRT,@DZ RIS    ;SET THE INTERRUPT VECTOR STATUS
3808 016646 013777 027060 163224      MOV      DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
3809 016654 052777 000140 163160      BIS      #RIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
3810 016662 113777 001422 163176      MOV     TD0,@DZTDR       ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
3811 016670 005005                CLR      R5
3812 016672 032777 000200 163142 9$:  BIT      #RDONE,@DZCSR
3813 016700 001403                BEQ     10$
3814 016702 000240                NOP
3815 016704 000240                NOP
3816 016706 000412                BR       13$
3817 016710 104414                10$: DELAY
3818 016712 005205                INC      R5
3819 016714 001366                BNE     9$
3820 016716 104004                ERROR   4                ;*NO RX DONE! (NOT SLT)
3821 016720 000405                BR       13$              ;CONTINUE TEST
3822 016722 104010                11$: ERROR   10            ;*TRANSMITTER SHOULD NOT INTERRUPT
3823 016724 022626                CMP     (SP)+,(SP)+      ;POP FOR FAKE RTI
3824 016726 000402                BR       13$              ;CONT TEST
3825 016730 104012                12$: ERROR   12            ;*RECEIVER SHOULD NOT INTERRUPT
3826 016732 022626                CMP     (SP)+,(SP)+      ;POP FOR FAKE RTI
3827 016734                13$:
3828 016734 104413                DEVICE.CLR                ;ISSUE DEVICE CLEAR (RESET)
3829                ;***** TEST 26 *****
3830                ;* THIS TEST VERIFIES THAT THE DEVICE DOES INTERRUPT
3831                ;*WHILE THE PROCESSOR STATUS IS SET TO EXACTLY
3832                ;*ONE LEVEL LOWER THAN THE DZ11. DZ11 PRIORITY
3833                ;*DEFAULT TO LEVEL 5 MINUS ONE LEVEL IS LEVEL 4.
3834                ;:* TEST 26
3835                ;*****
3836 016736 000004                TST26: SCOPE
3837 016740 012737 000026 001122      MOV     #26,$STNM        ;LOAD THE NUMBER OF THIS TEST
3838 016746 012737 017264 001360      MOV     #TST27,NEXT      ;POINT TO THE START OF THE NEXT TEST

```



```
3839 016754 104417          DCLASM          :CLEAR DEVICE AND SET MAINT BIT IF I MODE
3840 016756 013701 001366    MOV      PAR,R1  :PICK UP PARAMETERS
3841 016762 012702 000001    MOV      #1,R2  :PICK UP INIT POINTER
3842 016766 030237 001364    1$: BIT      R2,LINE :SHOULD THIS LINE BE SET UP ?
3843 016772 001402          BEQ      2$     :NO
3844 016774 010177 163052    MOV      R1,@DZLPR :SET UP LINE PARAMETERS
3845 017000 005201          2$: INC      R1     :POSITION POINTER TO THE NEXT LINE
3846 017002 106302          ASLB     R2     :GOT 'EM ALL ?
3847 017004 103370          BCC     1$     :IF NO, GO SET UP THE NEXT LINE
3848 017006 005037 001372    CLR     SAVLIN  :CLEAR LINE # INDICATOR
3849 017012 106437 027062    MTPS   @#LESS1 :MAKE CPU ONE LEVEL LOWER THAN DZ11
3850 017016 113777 001364 163032    MOVB   LINE,@DZTCR :ENABLE THE VALID LINES
3851 017024          3$:
3852 017024 012777 017116 163044    MOV     #6$,@DZTIV :SET UP THE TRANSMITTER INTERRUPT VECTOR
3853 017032 012777 017134 163032    MOV     #7$,@DZRIV :SET UP THE RECEIVER INTERRUPT VECTOR
3854 017040 013777 027060 163026    MOV     DZPRT,@DZCRIS :SET THE INTERRUPT VECTOR STATUS
3855 017046 013777 027060 163024    MOV     DZPRT,@DZTIS  :SET TRANSMITTER INTERRUPT PRIORITY
3856 017054 052777 040040 162760    BIS     #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
3857 017062 005005          CLR     R5
3858 017064 032777 100000 162750 4$: BIT     #TRDY,@DZCSR
3859 017072 001404          BEQ     5$
3860 017074 000240          NOP
3861 017076 000240          NOP
3862 017100 104007          ERROR   7      :*TRANSMITTER FAILED TO INTERUPT
3863 017102 000416          BR     8$
3864 017104 104414          5$: DELAY
3865 017106 005205          INC     R5
3866 017110 001365          BNE     4$
3867 017112 104003          ERROR   3      :*TRDY NOT SET!
3868 017114 000411          BR     8$
3869 017116 022626          6$: POP2SP :REMOVE THE INTERRUPT FROM THE STACK
3870 017120 042777 040000 162714    BIC     #TIE,@DZCSR :DON'T LET ANY MORE INTERRUPTS OCCUR
3871 017126 106437 027062    MTPS   @#LESS1  :MAKE CPU ONE LEVEL LOWER THAN DZ11
3872 017132 000402          BR     8$
3873 017134 104012          7$: ERROR   12   :*RECEIVER SHOULD NOT INTERRUPT
3874 017136 022626          CMP     (SP)+,(SP)+ :POP FOR FAKE RTI
3875 017140 042777 040000 162674 8$: BIC     #TIE,@DZCSR :RESET TRANSMITTER INTERRUPT ENABLE
3876 017146 012777 017246 162722    MOV     #11$,@DZTIV :SET UP THE TRANSMITTER INTERRUPT VECTOR
3877 017154 012777 017254 162710    MOV     #12$,@DZRIV :SET UP THE RECEIVER INTERRUPT VECTOR
3878 017162 013777 027060 162704    MOV     DZPRT,@DZCRIS :SET THE INTERRUPT VECTOR STATUS
3879 017170 013777 027060 162702    MOV     DZPRT,@DZTIS  :SET TRANSMITTER INTERRUPT PRIORITY
3880 017176 052777 000140 162636    BIS     #RIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
3881 017204 113777 001422 162654    MOVB   TD0,@DZTDR  :PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
3882 017212 005005          CLR     R5
3883 017214 032777 000200 162620 9$: BIT     #RDONE,@DZCSR
3884 017222 001404          BEQ     10$
3885 017224 000240          NOP
3886 017226 000240          NOP
3887 017230 104011          ERROR   11   :*RECEIVER FAILED TO INTERRUPT
3888 017232 000413          BR     13$
3889 017234 104414          10$: DELAY
3890 017236 005205          INC     R5
3891 017240 001365          BNE     9$
3892 017242 104004          ERROR   4      :*NO RX DONE. (NOT SET)
3893 017244 000406          BR     13$
3894 017246 104010          11$: ERROR   10   :*TRANSMITTER SHOULD NOT INTERRUPT
```



CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 82  
CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0081

```

3895 017250 022626          CMP      (SP)+,(SP)+      ;POP FOR FAKE RTI
3896 017252 000403          BR       13$              ;CONT TEST
3897 017254 022626          12$:    POP2SP            ;REMOVE THE INTERRUPT FROM THE STACK
3898 017256 005077 162560   CLR      @DZCSR           ;DON'T ALLOW ANY MORE INTERRUPTS
3899 017262                    13$:
3900 017262 104413          DEVICE.CLR                ;ISSUE DEVICE CLEAR (RESET)
3901
3902                          ;***** TEST 27 *****
3903                          ;*THIS TEST VERIFIES THAT THE RECEIVER WILL
3904                          ;*INTERRUPT BEFORE THE TRANSMITTER EVEN
3905                          ;*THOUGH THE TRANSMITTER WAS ENABLED
3906                          ;*FIRST. SET PS TO LEVEL 7;
3907                          ;*GET RDONE AND TRDY TO SET;
3908                          ;*SET TX IE AND RX IE;
3909                          ;*CLEAR PS AND EXPECT RX TO INTERRUPT FIRST
3910                          ;:* TEST 27
3911                          ;*****
3912 017264 000004          TST27: SCOPE
3913 017266 012737 000027 001122  MOV      #27,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
3914 017274 012737 017716 001360  MOV      #TST30,NEXT     ;POINT TO THE START OF THE NEXT TEST
3915 017302 104417          DCLASM                    ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3916 017304 013701 001366  MOV      PAR,R1           ;PICK UP PARAMETERS
3917 017310 012702 000001  MOV      #1,R2            ;PICK UP INIT POINTER
3918 017314 030237 001364          1$:    BIT      R2,LINE       ;SHOULD THIS LINE BE SET UP ?
3919 017320 001402          BEQ      2$              ;NO
3920 017322 010177 162524  MOV      R1,@DZLPR        ;SET UP LINE PARAMETERS
3921 017326 005201          2$:    INC      R1          ;POSITION POINTER TO THE NEXT LINE
3922 017330 106302          ASLB     R2              ;GOT 'EM ALL ?
3923 017332 103370          BCC      1$              ;IF NO, GO SET UP THE NEXT LINE
3924 017334 005037 001372  CLR      SAVLIN           ;CLEAR LINE # INDICATOR
3925 017340 012777 017570 162524  MOV      #8$,@DZRIV       ;SETUP INTERRUPT STUFF
3926 017346 013777 027060 162520  MOV      DZPRT,@DZRI$
3927 017354 012777 017660 162514  MOV      #12$,@DZTI$
3928 017362 013777 027060 162510  MOV      DZPRT,@DZTI$
3929 017370 052777 000040 162444  BIS      #MSENAB,@DZCSR
3930 017376 012702 000001  MOV      #1,R2            ;LINE POINTER
3931 017402 030237 001364          3$:    BIT      R2,LINE       ;VALID LINE ?
3932 017406 001004          BNE      4$              ;NO
3933 017410 005237 001372  INC      SAVLIN
3934 017414 106302          ASLB     R2
3935 017416 000771          BR       3$
3936 017420 106427 000340          4$:    MTPS     #PR7
3937 017424 000240          NOP
3938 017426 000240          NOP
3939 017430 110277 162422  MOVB     R2,@DZTCR        ;SET TCR BIT
3940 017434 005777 162406  TST      @DZRBUF         ;VALID DATA?
3941 017440 100001          BPL      .+4             ;IT BETTER NOT BE SET
3942 017442 104017          ERROR   17              ;DATA VALID SHOULD NOT BE SET
3943 017444 105777 162372          5$:    TSTB     @DZCSR       ;RECEIVER DONE ?
3944 017450 100001          BPL      .+4
3945 017452 104020          ERROR   20              ;RECEIVER DONE BIT SHOULD NOT BE SET
3946 017454 005005          CLR     R5
3947 017456 005004          CLR     R4
3948 017460 005777 162356          99$:   TST      @DZCSR         ;WAIT FOR TRDY
3949 017464 100404          BMI     100$           ;BR IF READY
3950 017466 104414          DELAY

```

```

3951 017470 005204      INC      R4      ;
3952 017472 001372      JNE      99$
3953 017474 104003      ERROR   3      ;TRDY FAILED TO SET
3954 017476 105777 162364 100$:  CLR     @DZTDR
3955 017502 005004      CLR     R4
3956 017504 032777 000200 162330 6$:  BIT     #RDONE,@DZCSR
3957 017512 001004      BNE     7$
3958 017514 104414      DELAY
3959 017516 005204      INC     R4
3960 017520 001371      BNE     6$
3961 017522 104004      ERROR   4      ;*RDONE FAILED TO SET!
3962 017524 005777 162312 7$:  TST     @DZCSR      ;TRANS DONE BIT = 1 ?
3963 017530 100401      BMI     .+4      ;YES
3964 017532 104003      ERROR   3      ;*NO TRANS DONE FAILED TO SET
3965      ;NOW THAT BOTH TRANSMITTER AND RECEIVER DONE BIT -1
3966      ;SET INTERRUPT ENABLES AND WATCH THE FUR FLY
3967 017534 052777 040000 162300  BIS     #TIE,@DZCSR
3968 017542 052777 000100 162272  BIS     #RIE,@DZCSR
3969 017550 106427 000000  MTPS   #0
3970 017554 000240      NOP
3971 017556 000240      NOP
3972 017560 104007      ERROR   7      ;*TRANSMITTER FAILED TO INTERRUPT
3973 017562 104011      ERROR   11     ;*RECEIVER FAILED TO INTERRUPT
3974      ;CHECK BR LEVEL
3975 017564 000137 017664  JMP     13$      ;GET OUT
3976
3977      ;RECEIVER INTERRUPT ROUTINE
3978 017570 017704 162252 8$:  MOV     @DZRBUF,R4      ;ACTUAL
3979 017574 010403      MOV     R4,R3
3980 017576 000303      SWAB   R3
3981 017600 042703 177770  BIC     #^C<?>,R3      ;STRIP JUNK
3982 017604 105737 021371  TSTB   MODE+1      ;IS THIS TEST IN STAGGERED MODE?
3983 017610 001406      BEQ    11$      ;IF NOT, SKIP STAGGERED SETUP
3984
3985      ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3986
3987 017612 006203      ASR    R3      ;GET THE LAST BIT INTO THE CARRY BIT
3988 017614 103402      BCS    9$      ;IF IT IS SET, GO CLEAR IT
3989 017616 000261      SEC
3990 017620 000401      BR     10$     ;IF IT IS CLEAR SET IT HERE
3991 017622 000241 9$:  CLC
3992 017624 006103 10$:  ROL    R3      ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3993 017626 020337 001372 11$:  CMP    R3,SAVLIN  ;GET THE NEW BIT BACK INTO R3
3994 017632 001401      BEQ    .+4      ;IS THIS A VALID LINE
3995 017634 104015      BEQ    .+4      ;YES
3996 017636 042704 177400  ERROR   15     ;*INVALID LINE
3997 017642 120504      BIC    #^C<377>,R4 ;STRIP JUNK
3998 017644 001401      CMPB   R5,R4      ;DATA COMPARE ?
3999 017646 104005      BEQ    .+4      ;YES
4000 017650 040277 162202  ERROR   5      ;*DATA DOES NOT COMPARE
4001 017654 022626      BIC    R2,@DZTCR  ;CLEAR TCR BIT
4002 017656 000402      POP2SP ;REMOVE HE INTERRUPT VECTOR FROM THE STACK
4003      BR     13$      ;GO GET OUT OF INTERRUPT MODE
4004 017660 104011 12$:  ;TRANSMITTER INTERRUPT SVC ROUTINE
4005      ERROR   11     ;THE RECEIVER INTERRUPT FAILED
4006 017662 022626      POP2SP ;TO OVERRIDE THE TRANSMITTER
;REMOVE THE INTERRUPT VECTOR FROM THE STACK

```

```

4007 017664 042777 040100 162150 13$: BIC #TIE!RIE,@DZCSR ;CLEAR INTERRUPT ENABLES
4008 017672 013777 002074 162172     MOV DZRI5,@DZRIV ;RESTORE TRAPCATCHER
4009 017700 005077 162170     CLR @DZRI5
4010 017704 013777 002100 162164     MOV DZTI5,@DZTIV
4011 017712 005077 162162     CLR @DZTI5
4012                                     :***** TEST 30 *****
4013                                     :*TEST TO VERIFY THAT 'RDONE DOES NOT SET
4014                                     :*IF THE SCANNER IS DISABLED.
4015                                     :*TURN ON SCANNER, WAIT FOR TRDY.
4016                                     :*TURN OFF SCANNER, TRANSMIT A CHARACTER
4017                                     :*'RDONE SHOULD NOT SET.
4018                                     ;:* TEST 30
4019                                     :*****
4020 017716 000004 TST30: SCOPE
4021 017720 012737 000030 001122     MOV #30,$TSTNM ;LOAD THE NUMBER OF THIS TEST
4022 017726 012737 020104 001360     MOV #TST31,NEXT ;POINT TO THE START OF THE NEXT TEST
4023 017734 104417     DCLASM ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4024 017736 013701 001366     MOV PAR,R1 ;PICK UP PARAMETERS
4025 017742 012702 000001     MOV #1,R2 ;PICK UP INIT POINTER
4026 017746 030237 001364 1$: BIT R2,LINE ;SHOULD THIS LINE BE SET UP ?
4027 017752 001402     BEQ 2$ ;NO
4028 017754 010177 162072     MOV R1,@DZLPR ;SET UP LINE PARAMETERS
4029 017760 005201 2$: INC R1 ;POSITION POINTER TO THE NEXT LINE
4030 017762 106302     ASLB R2 ;GOT 'EM ALL ?
4031 017764 103370     BCC 1$ ;IF NO, GO SET UP THE NEXT LINE
4032 017766 005037 001372     CLR SAVLIN ;CLEAR LINE # INDICATOR
4033 017772 052777 000040 162042     BIS #MSENAB,@DZCSR ;TURN ON SCANNER
4034 020000 012702 000001     MOV #1,R2 ;INIT LINE COUNTER
4035 020004 030237 001364 3$: BIT R2,LINE ;FIND A VALID LINE
4036 020010 001004     BNE 4$ ;IF WE FOUND ONE GO TO TEST
4037 020012 005237 001372     INC SAVLIN ;IF NOT
4038 020016 106302     ASLB R2 ;KEEP LOOKING
4039 020020 000771     BR 3$
4040 020022 110277 162030 4$: MOVB R2,@DZTCR ;SET TCR BIT
4041 020026 005005     CLR R5
4042 020030 005777 162006 5$: TST @DZCSR ;IS TRDY SET
4043 020034 100404     BMI 6$ ;CON'T TESTING IF IT IS
4044 020036 104414     DELAY ;IF IT NOT WAIT A WHILE
4045 020040 005205     INC R5
4046 020042 001372     BNE 5$
4047 020044 104003     ERROR 3 ;WE WAITED LONG ENOUGH-ERROR
4048 020046 042777 000040 161766 6$: BIC #MSENAB,@DZCSR ;TURN OFF SCANNER
4049 020054 105077 162006     CLRB @DZTDR ;TRANSMIT A CHARACTER
4050 020060 005005     CLR R5 ;CLEAR COUNTER
4051 020062 104414 7$: DELAY ;WAIT SUFFICIENT TIME FOR
4052 020064 005205     INC R5 ;RDONE TO SET
4053 020066 001375     BNE 7$
4054 020070 032777 000200 161744     BIT #RDONE,@DZCSR ;RDONE SET
4055 020076 001401     BEQ 8$ ;IT SHOULDN'T BE-CONTINUE
4056 020100 104020     ERROR 20 ;IF IT IS THERE'S AN ERROR
4057 020102 104400 8$: ADVANCE
4058                                     :***** TEST 31 *****
4059                                     :*THIS TEST VERIFIES OVERRUN AND SILO ALARM
4060                                     :*ONE LINE AT A TIME - BASED UPON VALID LINES
4061                                     :*AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
4062                                     :*TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN

```

4063 : \*EXPECTS SILO ALARM TO SET. THEN THE ENTIRE  
 4064 : \*SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH  
 4065 : \*CHAR PULLED OUT OUT THE SILO.  
 4066 : \*USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS  
 4067 : \*ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.  
 4068 : \*USED TO SCOPE SILO ALARM PULSES, ETC.

::\* TEST 31

::\*\*\*\*\*

4071	020104	000004				TST31: SCOPE		
4072	020106	012737	000031	001122		MOV #31,\$TSTNM	:LOAD THE NUMBER OF THIS TEST	
4073	020114	012737	020632	001360		MOV #TST32,NEXT	:POINT TO THE START OF THE NEXT TEST	
4074	020122	012737	020536	001362		MOV #18\$,LOCK	:SET FOR LOOP	
4075	020130	104417				DCLASM	:CLEAR DEVICE AND SET MAINT BIT IF I MODE	
4076	020132	013701	001366			MOV PAR,R1	:PICK UP PARAMETERS	
4077	020136	012702	000001			MOV #1,R2	:PICK UP INIT POINTER	
4078	020142	030237	001364		1\$:	BIT R2,LINE	:SHOULD THIS LINE BE SET UP ?	
4079	020146	001402				BEQ 2\$	:NO	
4080	020150	010177	161676			MOV R1,@DZLPR	:SET UP LINE PARAMETERS	
4081	020154	005201			2\$:	INC R1	:POSITION POINTER TO THE NEXT LINE	
4082	020156	106302				ASLB R2	:GOT 'EM ALL ?	
4083	020160	103370				BCC 1\$	:IF NO, GO SET UP THE NEXT LINE	
4084	020162	005037	001372			CLR SAVLIN	:CLEAR LINE # INDICATOR	
4085	020166	012700	001422			MOV #TDO,R0	:POINT TO THE DATA AREA	
4086	020172	005020				CLR (R0)+	:CLEAR A DATA WORD	
4087	020174	022700	001462			CMP #STOP,R0	:FINISHED ?	
4088	020200	001374				BNE .-6	:NO	
4089	020202	005000				CLR R0	:CLEAR OFFSET	
4090	020204	012702	000001			MOV #1,R2	:LINE POINTER	
4091	020210	052777	010040	161624		BIS #MSENAB!SILOEN,@DZCSR	:START SCANNER & SET SILO ENABLE	
4092	020216	030237	001364		3\$:	BIT R2,LINE	:VALID LINE?	
4093	020222	001002				BNE .+6	:YES	
4094	020224	000137	020520			JMP 22\$	:TRY NEXT LINE	
4095	020230	013700	001372			MOV SAVLIN,R0	:MAKE OFFSET	
4096	020234	006300				ASL R0	:MAKE POWER OF TWO	
4097	020236	010277	161614			MOV R2,@DZTCR	:SET TCR BIT	
4098	020242	105777	161574		4\$:	TSTB @DZCSR	:REC DONE = 1 ?	
4099	020246	100001				BPL .+4		
4100	020250	104020				ERROR 20	:REC DONE SHOULD NOT - 1	
4101	020252	005003				CLR R3	:SET CHARACTER COUNT	
4102	020254	005004			5\$:	CLR R4		
4103	020256	032777	100000	161556	6\$:	BIT #TRDY,@DZCSR		
4104	020264	001004				BNE 7\$		
4105	020266	104414				DELAY		
4106	020270	105204				INCB R4		
4107	020272	001371				BNE 6\$		
4108	020274	104003				ERROR 3	: *TRDY FAILED TO SET	
4109	020276	116077	001422	161562	7\$:	MOVB TDO(R0),@DZTDR	:LOAD A CHARACTER	
4110	020304	005260	001422			INC TDO(R0)	:SET UP NEXT CHARACTER	
4111	020310	020327	000017			CMP R3,#15.	:16 CHARACTERS ?	
4112	020314	103006				BHIS 8\$		
4113	020316	032777	020000	161516		BIT #SILOAL,@DZCSR	:SILO ALARM = 0 ?	
4114	020324	001401				BEQ .+4	:YES	
4115	020326	104013				ERROR 13	: *SILO ALARM SHOULD NOT	
4116							:UNTIL 16. DATA CHARACTERS	
4117	020330	000411				BR 10\$		
4118	020332	005004			8\$:	CLR R4		

4119	020354	032777	020000	161500	9\$:	BIT	#SILOAL,@DZCSR	
4120	020342	001004				BNE	10\$	
4121	020344	104414				DELAY		
4122	020346	005204				INC	R4	
4123	020350	001371				BNE	9\$	
4124	020352	104014				ERROR	14	:*SILO ALARM FAILED TO SET.
4125								:SILO ALARM SHOULD =1 AFTER 16.
4126								:DATA CHARACTERS
4127	020354	005203			10\$:	INC	R3	:INC CHAR COUNT
4128	020356	022703	000102			CMP	#66.,R3	:FINISHED SENDING CHARACTERS ?
4129	020362	001334				BNE	5\$	:NO
4130	020364	005004				CLR	R4	
4131	020366	104414				DELAY		
4132	020370	105204				INCB	R4	
4133	020372	001375				BNE	-4	
4134								:NOW LETS READ THE SILO
4135	020374	013705	001372			MOV	SAVLIN,R5	:MAKE EXPECTED LINE #
4136	020400	105737	001371			TSTB	MODE+1	:IS THIS TEST IN STAGGERED MODE?
4137	020404	001406				BEQ	13\$	:IF NOT, SKIP STAGGERED SETUP
4138								
4139								:WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
4140								
4141	020406	006205				ASR	R5	:GET THE LAST BIT INTO THE CARRY BIT
4142	020410	103402				BCS	11\$	:IF IT IS SET, GO CLEAR IT
4143	020412	000261				SEC		:IF IT IS CLEAR SET IT HERE
4144	020414	000401				BR	12\$	:SKIP THE CLEARING
4145	020416	000241			11\$:	CLC		:CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
4146	020420	006105			12\$:	ROL	R5	:GET THE NEW BIT BACK INTO R5
4147	020422	000305			13\$:	SWAB	R5	:PUT IN UPPER BYTE
4148	020424	052705	100000			BIS	#DVALID,R5	:ADD DATA VALID
4149	020430	017704	161412		14\$:	MOV	@DZRBUF,R4	:ACTUAL
4150	020434	020405				CMP	R4,R5	:ACTUAL VS. EXPECTED
4151	020436	001401				BEQ	15\$	:YES
4152	020440	104006				ERROR	6	:*DATA/CONTENTS DID NOT COMPARE
4153	020442	032777	020000	161372	15\$:	BIT	#SILOAL,@DZCSR	:SILO ALARM= 0 ?
4154	020450	001401				BEQ	16\$	:YES
4155	020452	104016				ERROR	16	:READING DZRBUF DID NOT CLEAR SILO ALARM
4156	020454	005205			16\$:	INC	R5	:UP CHARACTER
4157	020456	120527	000077			CMPB	R5,#63.	:LAST SILO CHAR ?....64TH CHAR
4158	020462	101762				BLOS	14\$	
4159	020464	005205				INC	R5	:ADD 1 MORE FOR THE CLOBBERED CHAR
4160	020466	052705	040000			BIS	#OVRUN,R5	:ADD OVERRUN TO EXPECTED
4161	020472	120527	000101			CMPB	R5,#65.	:LAST CHARACTER ?
4162	020476	001754				BEQ	14\$	
4163	020500	017704	161342			MOV	@DZRBUF,R4	:FOR GOOD MEASURE
4164	020504	005704				TST	R4	:DATA VALID SHOULD = 0
4165	020506	100001				BPL	17\$	:YES
4166	020510	104017				ERROR	17	:DATA VALID SHOULD = 0
4167	020512	040277	161340		17\$:	BIC	R2,@DZTCR	:CLR TCR BIT
4168	020516	104401				SCOP1		:LOOP?
4169	020520	005237	001372		22\$:	INC	SAVLIN	:INC EXPECTED LINE
4170	020524	106302				ASLB	R2	:NEXT LINE
4171	020526	103402				BCS	+6	:NO
4172	020530	000137	020216			JMP	3\$	:YES
4173	020534	104400				ADVANCE		:GO TO NEXT TEST
4174								

```

4175 ;TIGHT SCOPE LOOP FOR THIS TEST. SENDS 20. CHARACTERS
4176 ;ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09-1.
4177 ;USED TO SCOPE SILO ALARM PULSES, ETC.
4178
4179 020536 052777 010040 161276 18$: BIS #MSENAB!SILOEN,@DZCSR ;SETUP DEVICE
4180 020544 012777 020622 161324 MOV #20$,@DZTIV ;SETUP TRANSMITTER VECTOR
4181 020552 012737 000024 001216 MOV #20,$TMP0 ;TEMPORARY COUNT OF CHARACTER BURST
4182 020560 050277 161272 BIS R2,@DZTCR ;ENABLE LINE
4183 020564 052777 040000 161250 BIS #TIE,@DZCSR ;ENABLE INTERRUPTS
4184 020572 106427 000000 MTPS #0 ;LOWER PRIORITY
4185 020576 000001 19$: WAIT ;ALLOW INTERRUPTS
4186 020600 005337 001216 DEC $TMP0 ;REDUCE COUNT. ALL CHARACTERS SENT?
4187 020604 001374 BNE 19$ ;IF NO, WAIT FOR MORE
4188 020606 042777 050040 161226 BIC #SILOEN!MSENAB!TIE,@DZCSR ;RESET SILO COUNTER, CLEAR STROBE
4189 020614 104401 SCOP1 ;LOOP AGAIN?
4190 020616 000137 020512 JMP 17$ ;IF NOT, RETURN TO WHERE YOU LEFT OFF
4191 020622 112777 000252 161236 20$: MOVB #252,@DZTDR ;SEND A CHARACTER
4192 020630 000002 RTI ;ALLOW MORE CHARACTERS TO COME
4193 ;***** TEST 32 *****
4194 ;*THIS TEST THAT 'SILO ENABLE' WILL INHIBIT
4195 ;*RECEIVER INTERRUPTS AND THAT ON THE
4196 ;*16TH CHAR THAT 'SILO ALARM' WILL CAUSE AN
4197 ;*INTERRUPT WITH 'RIE' SET.
4198 ;*THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
4199
4200 ;:* TEST 32
4201 ;*****
4201 020632 000004 TST32: SCOPE
4202 020634 012737 000032 001122 MOV #32,$STNM ;LOAD THE NUMBER OF THIS TEST
4203 020642 012737 021214 001360 MOV #TST33,NEAT ;POINT TO THE START OF THE NEXT TEST
4204 020650 012737 020736 001362 MOV #3$,LOCK ;SET FOR LOOP
4205 020656 104417 DCLASM ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4206 020660 013701 001366 MOV PAR,R1 ;PICK UP PARAMETERS
4207 020664 012702 000001 MOV #1,R2 ;PICK UP INIT POINTER
4208 020670 030237 001364 1$: BIT R2,LINE ;SHOULD THIS LINE BE SET UP ?
4209 020674 001402 BEQ 2$ ;NO
4210 020676 010177 161150 MOV R1,@DZLPR ;SET UP LINE PARAMETERS
4211 020702 005201 2$: INC R1 ;POSITION POINTER TO THE NEXT LINE
4212 020704 106302 ASLB R2 ;GOT 'EM ALL ?
4213 020706 103370 BCC 1$ ;IF NO, GO SET UP THE NEXT LINE
4214 020710 005037 001372 CLR SAVLIN ;CLEAR LINE # INDICATOR
4215 020714 012700 001422 MOV #TDO,R0 ;POINT TO THE DATA AREA
4216 020720 005020 CLR (R0)+ ;CLEAR A DATA WORD
4217 020722 022700 001462 CMP #STOP,F0 ;FINISHED ?
4218 020726 001374 BNE -6 ;NO
4219 020730 005000 CLR R0 ;CLEAR OFFSET
4220 020732 012702 000001 MOV #1,R2 ;LINE POINTER
4221 020736 012777 021156 161126 3$: MOV #11$,@DZRIV ;SET FOR UNEXPECTED INTER.
4222 020744 012777 000340 161122 MOV #PR7,@DZCRIS ;SET PRIO.
4223 020752 052777 010140 161062 BIS #MSENAB!SILOEN!RIE,@DZCSR ;START SCANNER & SET SILO ENABLE
4224
4225 020760 030237 001364 BIT R2,LINE ;VALID LINE?
4226 020764 001002 BNE +6 ;YES
4227 020766 000137 021174 JMP 22$ ;TRY NEXT LINE
4228 020772 005777 161150 TST @DZRBUFF ;EMPTY THE SILO
4229 020776 100775 BMI -4 ;BR IF DATA VALID IS SET!
4230 021000 106427 000000 MTPS #0 ;SET PROCESSOR PRIORITY TO 0

```

CZDZA-FO  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 88  
CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0087

4231	021004	013700	001372		MOV	SAVLIN,R0	:MAKE OFFSET	
4232	021010	006300			ASL	R0	:MAKE POWER OF TWO	
4233	021012	010277	161040		MOV	R2,@DZTCR	:SET TCR BIT	
4234	021016	005004		5\$:	CLR	R4		
4235	021020	032777	100000	161014	6\$:	BIT	#TRDY,@DZCSR	
4236	021026	001004			BNE	7\$		
4237	021030	104414			DELAY			
4238	021032	005204			INC	R4		
4239	021034	001371			BNE	6\$		
4240	021036	104003			ERROR	3	:*TRDY FAILED TO SET	
4241	021040	116077	001422	161020	7\$:	MOVB	TDO(R0),@DZTDR	:LOAD A CHARACTER
4242	021046	005260	001422		INC	TDC(R0)	:SET UP NEXT CHARACTER	
4243	021052	022760	000017	001422	CMP	#15.,TDO(R0)	:15 CHARS YET?	
4244	021060	001406			BEQ	8\$		
4245	021062	032777	020000	160752	BIT	#SILOAL,@DZCSR	:SILO ALARM = 0 ?	
4246	021070	001401			BEQ	+.4	:YES	
4247	021072	104013			ERROR	13	:*SILO ALARM SHOULD NOT - 1	
4248							:UNTIL 16. DATA CHARACTERS	
4249	021074	000750			BR	5\$		
4250	021076	012777	021164	160766	8\$:	MOV	#12\$,@DZRIV	:SET NEW VECTOR
4251	021104	032777	100000	160730	BIT	#TRDY,@DZCSR	:READY FOR 16TH CHAR	
4252	021112	001774			BEQ	.-6		
4253	021114	016077	001422	160744	MOV	TDO(R0),@DZTDR	:LOAD THE 16TH CHAR.	
4254	021122	005004			CLR	R'		
4255	021124	032777	020000	160710	9\$:	BIT	#SILOAL,@DZCSR	
4256	021132	001005			BNE	10\$		
4257	021134	104414			DELAY			
4258	021136	005204			INC	R4		
4259	021140	001371			BNE	9\$		
4260	021142	104014			ERROR	14	:*SILO ALARM FAILED TO SET!	
4261	021144	000410			BR	17\$	:SILO ALARM SHOULD -1 AFTER 16.	
4262							:DATA CHARACTERS	
4263	021146	000240		10\$:	NOP		:STALL	
4264	021150	000240			NOP			
4265	021152	104000			ERROR		:SILO ALARM NOT INTERRUPTING.	
4266	021154	000404			BR	17\$	:CONTINUE TEST.	
4267	021156	022626		11\$:	CMP	(SP)+,(SP)+	:FAKE RTI	
4268	021160	104012			ERROR	12	:RX SHOULD NOT INTERRUPT	
4269	021162	000401			BR	17\$	:CONTINUE	
4270	021164	022626		12\$:	CMP	(SP)+,(SP)+	:GOOD INTERRUPT TO HERE.	
4271	021166	040277	160664	17\$:	BIC	R2,@DZTCR	:CLR TCR BIT	
4272	021172	104401			SCOP1		:LOOP?	
4273	021174	005237	001372	22\$:	INC	SAVLIN	:INC EXPECTED LINE	
4274	021200	106302			ASLB	R2	:NEXT LINE	
4275	021202	103402			BCS	+.6	:NO	
4276	021204	000137	020736		JMP	3\$	:YES	
4277	021210	005037	001362		CLR	LOCK	:CLEAR TIGHT LOOP FOR NEXT TEST	

```

4278                                     :***** TEST 33 *****
4279                                     :*THIS TEST RUNS ALL LINES FULL BORE
4280                                     :*BASED UPON QUALIFIED LINES
4281                                     :*. THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
4282                                     :*TRANSMITTER
4283                                     ::* TEST 33
4284                                     :*****
4285 021214 000004 TST33: SCOPE
4286 021216 012737 000033 001122 MOV #33,$TSTNM ;LOAD THE NUMBER OF THIS TEST
4287 021224 012737 022022 001360 MOV #TST34,NEXT ;POINT TO THE START OF THE NEXT TEST
4288 021232 104417 DCLASM ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4289 021234 013737 001364 022020 MOV LINE,RXTCR ;SET IMAGE OF TCR BITS
4290 021242 013701 001366 RSTART: MOV PAR,R1 ;PICK UP PARAMETER
4291 021246 012700 000001 MOV #1,R0 ;PICK UP INIT POINTER
4292 021252 030037 001364 INIT: BIT R0,LINE ;SHOULD THIS LINE BE SET UP
4293 021256 001402 BEQ 1$ ;NO
4294 021260 010177 160566 MOV R1,@DZLPR ;SET UP LINE PARAM REGISTER
4295 021264 005201 1$: INC R1
4296 021266 106300 ASLB R0 ;GOT 'EM ALL ?
4297 021270 103370 BCC INIT ;NO
4298 021272 012700 001422 MOV #TDO,R0 ;CLEAR TRANS DATA POINTER & REC POINTERS
4299 021276 005020 INIT1: CLR (R0)+
4300 021300 022700 001462 CMP #STOP,R0 ;FINISHED ?
4301 021304 001374 BNE INIT1 ;NO, CONTINUE CLEARING
4302 021306 012777 021542 160556 MOV #RXSVC,@DZRIV ;SET UP REC INTR VECTOR
4303 021314 012777 000340 160552 MOV #PR7,@DZ RIS ;STATUS
4304 021322 012777 021444 160546 MOV #TXSVC,@DZTIV ;SET UP TRANS INTR VECTOR
4305 021330 012777 000340 160542 MOV #PR7,@DZTIS ;STATUS
4306 021336 052777 000100 160476 BIS #RIE,@DZCSR ;SET REC INTR ENABLE
4307 021344 052777 040000 160470 BIS #TIE,@DZCSR ;SET TRANS INTR ENABLE
4308 021352 052777 000040 160462 BIS #MSENAB,@DZCSR ;SET MASTER SCAN ENABLE
4309 021360 113777 001364 160470 MOVB LINE,@DZTCR ;SET TCR BITS...UP UP AND AWAY
4310 021366 106437 027062 MTPS @#LESS1 ;ALLOW INTERRUPTS
4311
4312
4313 021372 005037 021442 SNAP: CLR 66$
4314 021376 013727 006722 67$: MOV DLYCNT,(PC)+ ;SET FOR DELAY
4315 021402 000000 68$: 0
4316 021404 005337 021402 DEC 68$
4317 021410 001375 BNE .-4
4318 021412 105737 022020 TSTB RXTCR ;WAIT FOR ALL RECIEVERS TO FINISH
4319 021416 001002 BNE 3$
4320 021420 000137 021720 JMP OUT
4321 021424 005237 021442 3$: INC 66$
4322 021430 001362 BNE 67$
4323 021432 104007 ERROR 7 ;*TRANSMITTER FAILED TO INTERRUPT
4324 021434 104011 ERROR 11 ;*RECEIVER FAILED TO INTERRUPT
4325 021436 000137 021772 JMP FINI
4326 021442 000000 66$: 0
4327
4328 ;TRANS INTR SVC ROUTINE
4329 021444 005777 160372 TXSVC: TST @DZCSR ;TRANS INTR ?
4330 021450 100401 BMI .+4
4331 021452 104003 ERROR 3 ;*TRANSMITTER FAILED
4332 021454 117703 160364 MOVB @HDZCSR,R3 ;SAVE IT
4333 ;NOW TEST FOR LINE # ETC

```



```

4334 021460 042703 177770      BIC      #^C<7>,R3      ;STRIP JUNK
4335 021464 010304      MOV      R3,R4          ;SAVE
4336 021466 010337 001372      MOV      R3,SAVLIN     ;ADJUST LOCATION FOR ERROR PRINTOUT
4337 021472 012702 000001      MOV      #1,R2         ;SET UP POSITION POINTER
4338 021476 105303      3$:      DECB      R3          ;IS IT THIS LINE ?
4339 021500 100402      BMI      4$            ;YES
4340 021502 006302      ASL      R2            ;UP THE LINE #
4341 021504 000774      BR       3$            ;GO 'ROUND AGAIN
4342 021506 030237 001364      4$:      BIT      R2,LINE     ;VALID LINE?
4343 021512 001001      BNE      .+4           ;YES
4344 021514 104010      ERROR    10           ;NO,INVALID LINE!!!
4345 021516 006304      ASL      R4            ;MAKE POWER OF 2
4346 021520 116477 001422 160340      MOVB     TD0(R4),@DZTDR ;LOAD CHARACTER
4347 021526 105264 001422      INCB     TD0(R4)       ;SET UP NEXT CHARACTER
4348 021532 001002      BNE      5$            ;LAST CHARACTER ?
4349 021534 040277 160316      BIC      R2,@DZTCR     ;YES ,CLEAR TCR BIT
4350 021540 000002      5$:      RTI

4351
4352
4353      ;REC INTR SVC ROUTINE
RXSVC: TSTB     @DZCSR     ;REC DONE ?
4354 021542 105777 160274      BMI      .+4           ;YES
4355 021546 100401      ERROR    4            ;FALSE INTERRUPT
4356 021550 104004      MOV      @DZRBUF,R4    ;SAVE IT
4357 021552 017704 160270      MOV      R4,R3
4358 021556 010403      SWAB     R3
4359 021560 000303      BIC      #^C<7>,R3     ;STRIP JUNK
4360 021562 042703 177770      MOV      R3,SAVLIN     ;SAVE LINE NUMBER
4361 021566 010337 001372      BIT      #SILOAL,@DZCSR ;SILO ALARM?
4362 021572 032777 020000 160242      BEQ      .+4           ;NO
4363 021600 001401      ERROR    11           ;SILO ALARM SHOULD NOT =1
4364 021602 104000      TST      R4            ;DATA VALID SET?
4365 021604 005704      BMI      .+4           ;YES
4366 021606 100401      ERROR    23           ;YOU LOSE ...DATA VALID WAS'NT SET
4367 021610 104023      BIT      #OVRRUN!FRMERR!PARER,R4
4368 021612 032704 070000      BEQ      .+4
4369 021616 001401      ERROR    5            ;RECEIVER ERROR FLAG/S WERE SET
4370 021620 104000      MOV      #1,R2         ;SET UP POSITION POINTER
4371 021622 012702 000001      5$:      DECB      R5
4372 021626 105303      BMI      6$            ;RE POSITION POINTER
4373 021630 100402      ASL      R2            ;GO 'ROUND AGAIN
4374 021632 006302      BR       5$            ;LINE VALID ?
4375 021634 000774      6$:      BIT      R2,LINE     ;LINE VALID ?
4376 021636 030237 001364      BNE      .+4           ;YES
4377 021642 001001      ERROR    11           ;INVALID LINE #
4378 021644 104011      MOV      SAVLIN,R3     ;GET THE LINE NUMBER AGAIN
4379 021646 013703 001372      ASL      R3            ;USE R3 AS A POINTER IN THE DATA TABLE
4380 021652 006303      CMPB     TR0(R3),R4    ;DOES THE DATA CHARACTER COMPARE ?
4381 021654 126304 001442      BEQ      2$            ;YES
4382 021660 001405      MOV      TR0(R3),R5    ;SAVE EXPECTED
4383 021662 016305 001442      BIC      #^C<377>,R4   ;CLEAR JUNK
4384 021666 042704 177400      ;R2 = LINE # BY BIT POSITION
4385      ;R4 = ACTUAL DATA
4386      ;R5 = EXPECTED DATA
4387      ;*NO, DATA DOES NOT COMPARE
4388 021672 104005      2$:      INC      TR0(R3)      ;SET UP FOR NEXT CHARACTER
4389 021674 005263 001442

```

```

4390 021700 105763 001442      TSTB   TRO(R3) ;ALL CHARS DONE?
4391 021704 001002      BNE    .+6
4392 021706 040237 022020      BIC    R2,RXTCR ;ZERO LINE DONE INDICATOR.
4393 021712 012716 021372      MOV    #SNAP,(SP) ;RESET THE BACKGROUND TIMING LOOP
4394 021716 000002      RTI
4395
4396
4397      ;FINISH UP ROUTINE
4398 021720 106427 000340      OUT:   MTPS   #PR7 ;STOP ALL INTERRUPTS
4399 021724 104413      DEVICE.CLR ;CLEAR ALL INTERRUPTS AWAY
4400 021726 005003      CLR    R3
4401 021730 005037 001372      CLR    SAVLIN
4402 021734 012702 000001      MOV    #1,R2
4403 021740 030237 001364      1$:   BIT    R2,LINE ;VALID LINE ?
4404 021744 001405      BEQ    2$ ;NO
4405 021746 022763 000400 001442      CMP    #400,TRO(R3) ;RECEIVED A BINARY COUNT PATTERN ?
4406 021754 001401      BEQ    .+4 ;YES
4407 021756 104027      ERROR  27 ;THE LINE FAILED TO RECEIVE A FULL
4408 ;BINARY COUNT PATTERN
4409 021760 005237 001372      2$:   INC    SAVLIN ;SET UP FOR NEXT LINE
4410 021764 005723      TST    (R3)+ ;ADD 2
4411 021766 106302      ASLB   R2 ;SET UP NEXT LINE POINTER
4412 021770 103363      BCC    1$ ;FINISHED ?
4413 021772
4414 021772 013777 002074 160072      FINI: MOV    DZRRIS,@DZRIV ;RESTORE TRAPCATCHER
4415 022000 005077 160070      CLR    @DZRRIS
4416 022004 013777 002100 160064      MOV    DZTRIS,@DZTRIV
4417 022012 005077 160062      CLR    @DZTRIS
4418 022016 104400      ADVANCE ;GO TO THE NEXT TEST
4419 022020 000000      RXTCR: 0 ;RX IMAGE OF TCR BITS
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439

```

```

:***** TEST 34 *****
:*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 34

```

:*****
TST34: SCOPE
MOV    #34,$STSTM ;LOAD THE NUMBER OF THIS TEST
MOV    #2,$TIMES
MOV    #TST35,NEXT ;POINT TO THE START OF THE NEXT TEST
MOV    #3$,LOCK ;SET FOR LOOP
CLR    OFFSET ;RESET THIS VARIABLE

```

```

4440 022022 000004
4441 022024 012737 000034 001122
4442 022032 012737 000002 001226
4443 022040 012737 022532 001360
4444 022046 012737 022172 001362
4445 022054 005037 023766

```

CZDZA-FO  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 92  
CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0091

```

4446 022060 005037 001372          CLR SAVLIN          ;RESET LINE NUMBER INDICATOR
4447 022064 005037 001374          CLR XMTLIN         ;USE THIS WORD TO TELL WHAT LINE TRANSMITTED
4448 022070 012737 000001 001216  MOV #1,$TMP0       ;USE $TMP0 AS A BIT POINTER
4449 022076 012737 010070 022530  MOV #RCVON!$S0!EIGHT!TWO$STOP,7$ ;BUILD TEMPORARY PARAMETERS
4450 022104 033737 001216 001364 1$: BIT $TMP0,LINE    ;IS THIS LINE ACTIVE?
4451 022112 001027          BNE 3$           ;IF SO, GO GET STARTED
4452 022114 012737 010070 022530 2$: MOV #RCVON!$S0!EIGHT!TWO$STOP,7$ ;LOAD PARAMETERS TEMPORARILY
4453 022122 012700 001422          MOV #TDO,RO        ;POINT TO THE DATA AREA
4454 022126 005020          CLR (RO)+         ;CLEAR A DATA WORD
4455 022130 022700 001462          CMP #STOP,RO      ;FINISHED ?
4456 022134 001374          BNE .-6          ;NO
4457 022136 005237 001374          INC XMTLIN        ;POINT TO THE NEXT LINE TO TRANSMIT
4458 022142 042737 000007 022530  BIC #7,7$        ;MAKE SURE TEMPORARY PARAMETERS POINT TO 0
4459 022150 053737 001374 022530  BIS XMTLIN,7$    ;ADD DESIRED LINE NUMBER
4460 022156 005037 023766          CLR OFFSET       ;
4461 022162 106337 001216          ASLB $TMP0       ;POINT TO THE NEXT LINE
4462 022166 103346          BCC 1$           ;PROCESS THE NEXT LINE
4463 022170 104400          ADVANCE          ;TEST TO SEE IF THIS TEST GETS REPEATED
4464 022172          3$:
4465 022172 104417          DCLASM           ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4466 022174 042737 010000 022530  BIC #RCVON,7$    ;ZERO PARAMTERS FOR TX LINE
4467 022202 013777 022530 157642  MOV 7$,@DZLPR    ;LOAD PARAMTERS FOR TX
4468 022210 005737 001370          TSI MODE         ;STAGGERED?
4469 022214 100011          BPL 100$        ;BR IF NO
4470 022216 000241          CLC             ;SET UP LINE
4471 022220 006037 022530          ROR 7$         ;
4472 022224 103002          BCC 98$        ;BR IF LINE WAS EVEN
4473 022226 000241          CLC             ;PREPARE TO MKE LINE EVEN
4474 022230 000401          BR 99$        ;CONTINUE
4475 022232 000261          98$: SEC       ;PREPARE TO MAKE LINE ODL
4476 022234 006137 022530 99$: ROL 7$      ;SET ALTERED LINE
4477 022240 052737 010000 022530 100$: BIS #RCVON,7$ ;SET RX ON
4478 022246 013777 022530 157576  MOV 7$,@DZLPR    ;LOAD RX PARAMETERS
4479 022254 013737 022530 001372  MOV 7$,SAVLIN   ;ADJUST LOCATION FOR ERROR PRINTOUT
4480 022262 042737 177770 001372  BIC #^C<7>,SAVLIN ;STRIP JUNK
4481 022270 042737 000007 022530  BIC #7,7$      ;CLEAR OLD LINE #
4482 022276 053737 001374 022530  BIS XMTLIN,7$  ;SET LINE UP AGAIN
4483 022304 013737 022530 001400  MOV 7$,REGIST  ;SAVE PARAMETERS FOR PRINTOUT
4484 022312 012700 001422          MOV #TDO,RO     ;POINT TO THE DATA AREA
4485 022316 005020          CLR (RO)+      ;CLEAR A DATA WORD
4486 022320 022700 001462          CMP #STOP,RO   ;FINISHED ?
4487 022324 001374          BNE .-6        ;NO
4488 022326 005002          CLR R2        ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
4489 022330 005003          CLR R3        ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
4490 022332 005037 001220          CLR $TMP1     ;INITIALIZE THE TIMER
4491 022336 005037 001224          CLR $TMP3     ;INITIALIZE THESE BITS ALSO
4492 022342 012737 000020 001376  MOV #20,XMTCNT ;SET HOW MANY CHARACTERS TO TRANSMIT
4493 022350 012777 023410 157520  MOV #XMTSRV,@DZTIV
4494 022356 012777 023554 157506  MOV #RXISR1,@DZRIV
4495 022364 013777 027060 157502  MOV DZPRT,@DZTIS
4496 022372 013777 027060 157500  MOV DZPRT,@DZTIS
4497 022400 113777 001216 157450  MOV $TMP0,@DZTCR ;START THE VALID LINE
4498 022406 052777 040140 157426  BIS #TIE!RIE!MSENAB,@DZCSR
4499 022414 106427 000000          MTPS #0       ;LOWER THE PRIORITY TO ALLOW INTERRUPTS
4500 022420 032777 000100 157414 4$: BIT #RIE,@DZCSR ;IS ROUTINE DONE?
4501 022426 001407          BEQ 5$        ;WHEN ALL IS DONE RX IE IS CLEARED IN ISR.

```

```

4502 022430 005237 001220      INC      $TMP1      ;COUNT TIME
4503 022434 001371              BNE      4$        ;CONTINUE TEST
4504 022436 105237 001224      INCB     $TMP3     ;DOUBLE COUNT
4505 022442 001366              BNE      4$        ;CONTINUE TEST
4506 022444 104011              ERROR    11        ;INTERRUPTS NOT FINISHED
4507 022446 004737 007360      JSR      PC,SERV.G ;<^G>?
4508 022452 104401              SCOP1                    ;LOOP?
4509 022454 062737 000002 023766  ADD      #2,OFFSET
4510 022462 013700 022530      MOV      7$,R0
4511 022466 042700 170377      RLC      #^C<17*400>,R0
4512 022472 022700 007400      CMP      #<17*400>,R0
4513 022476 001010              BNE      6$
4514 022500 032737 000030 022530  BIT      #BIT4+BIT3,7$
4515 022506 001602              BEQ      2$
4516 022510 162737 000010 022530  SUB      #BIT3,7$
4517 022516 000625              BR       3$
4518 022520 062737 000400 022530  6$: ADD      #400,7$
4519 022526 000621              BR       3$
4520 022530 000000      7$: 0
4521
4522      ;***** TEST 35 *****
4523      ;* THIS TEST VERIFIES THAT EVEN PARITY WORKS
4524      ;* FOR ALL ODD LINES SELECTED AND THAT ODD PARITY WORKS FOR ALL
4525      ;* EVEN LINES SELECTED.
4526      ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
4527      ;*THAT 'PE' (PARITY ERROR) CAN BE FLAGGED BY
4528      ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
4529      ;*YOU ARE IN 'STAGGERED' MODE.
4530      ;*40(8) CHARS ARE USED FOR THIS TEST.
4531      ;*ALL SELECTED LINES WILL BE ENABLED
4532      ;*AT THE SAME TIME!
4533      ;:* TEST 35
4534      ;*****
4535 022532 000004      TST35: SCOPE
4536 022534 012737 000035 001122  MOV      #35,$STNM ;LOAD THE NUMBER OF THIS TEST
4537 022542 012737 022772 001360  MOV      #TST36,NEXT ;POINT TO THE START OF THE NEXT TEST
4538 022550 005737 001370      TST     MODE      ;IS THIS STAGGERED MODE?
4539 022554 100105      BPL     6$        ;IF NOT, DON'T DO THIS TEST
4540 022556 104417      DCLASM ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4541 022560 013701 001366      MOV     PAR,R1    ;USE R1 TO BUILD PARAMETERS TO BE LOADED
4542 022564 042701 000200      BIC     #ODDPAR,R1 ;MAKE SURE ODD PARITY ISN'T SET
4543 022570 042701 000100      BIS     #PARITY,R1 ;MAKE SURE PARITY IS TURNED ON
4544 022574 042702 000001      MOV     #1,R2    ;USE R2 AS A LINE POINTER
4545 022580 001411      1$: BIT     R2,LINE ;IS THIS A VALID LINE?
4546 022584 001411      BEQ     3$        ;IF NOT, SKIP TO THE NEXT LINE
4547 022590 032701 000001      BIT     #BIT0,R1 ;IS THIS LINE AN ODD LINE?
4548 022594 001002      BNE     2$        ;IF IT'S ODD, USE EVEN PARITY
4549 022600 052701 000200      BIS     #ODDPAR,R1 ;IF IT'S EVEN, USE ODD PARITY
4550 022604 010177 157226      2$: MOV     R1,@DZLPR ;LOAD THE LINE PARAMETER REGISTER
4551 022608 042701 000200      BIC     #ODDPAR,R1 ;SET UP THE NEXT PARITY TO EVEN
4552 022612 001002      3$: INC     R1     ;POINT TO THE NEXT LINE
4553 022616 106302      ASLB    R2        ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
4554 022620 103361      BCC     1$        ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
4555 022624 005037 001372      CLR     SAVLIN   ;CLEAR THE LINE NUMBER INDICATOR
4556 022628 005002      CLR     R2        ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
4557 022632 005003      CLR     R3        ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
4558 022636 012737 000040 001376  MOV     #40,XMTCNT ;TRANSMIT A BINARY COUNT PATTERN(00-40)

```

```

4558 022654 012700 001422      MOV      #TDO,RO      ;POINT TO THE DATA AREA
4559 022660 005020      CLR      (RO)+        ;CLEAR A DATA WORD
4560 022662 022700 001462      CMP      #STOP,RO    ;FINISHED ?
4561 022666 001374      BNE     -.6          ;NO
4562 022670 005000      CLR      RO          ;CLEAR OFFSET
4563 022672 012777 023410 157176    MOV      #XMTSRV,@DZTIV ;SET UP THE TRANSMITTER INTERRUPT VECTOR
4564 022700 012777 023232 157164    MOV      #PARESE,@DZRIV ;SET UP THE RECEIVER INTERRUPT VECTOR
4565 022706 013777 027060 157160    MOV      DZPRT,@DZ RIS ;SET THE INTERRUPT VECTOR STATUS
4566 022714 013777 027060 157156    MOV      DZPRT,@DZTIS ;SET TRANSMITTER INTERRUPT PRIORITY
4567 022722 052777 040140 157112    BIS      #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
4568 022730 113777 001364 157120    MOVSB   LINE,@DZTCR  ;ENABLE ALL SELECTED LINES
4569 022736 106427 000000      MTPS    #0           ;ALLOW INTERRUPTS
4570 022742 032777 000100 157072 5$:    BIT      #RIE,@DZCSR  ;WHEN RX DONE; RIE WILL =0
4571 022750 001407      BEQ     6$          ;BR IF ALL DONE
4572 022752 005237 023404      INC     COUNT0
4573 022756 102771      BVS     5$
4574 022760 105237 023406      INCB   COUNT1
4575 022764 100366      BPL     5$
4576 022766 104011      ERROR  11           ;*RX FAILED TO FINISH (INTERRUPT)
4577 022770 104400      6$:    ADVANCE          ;ADVANCE LOOP
4578      ;***** TEST 36 *****
4579      ;*THIS TEST VERIFIES THAT ODD PARITY WORKS FOR ALL ODD LINES
4580      ;* SELECTED AND THAT EVEN PARITY WORKS FOR ALL EVEN LINES SELECTED
4581      ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
4582      ;*THAT 'PE' (PARITY ERROR) CAN BE FLAGGED BY
4583      ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
4584      ;*YOU ARE IN 'STAGGERED' MODE.
4585      ;*40(8) CHARS ARE USED FOR THIS TEST.
4586      ;*ALL SELECTED LINES WILL BE ENABLED
4587      ;*AT THE SAME TIME!
4588      ;:* TEST 36
4589      ;:*****
4590      ;:*****
4591 022772 000004      TST36: SCOPE
4592 022774 012737 000036 001122    MOV      #36,$STSTNM ;LOAD THE NUMBER OF THIS TEST
4593 023002 012737 004562 001360    MOV      #SEOP,NEXT  ;POINT TO THE END-OF-PASS HANDLER
4594 023010 005737 001370      TST     MODE         ;IS THIS STAGGERED MODE?
4595 023014 100105      BPL     6$          ;IF NOT, DON'T DO THIS TEST
4596 023016 104417      DCLASM          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4597 023020 013701 001366      MOV     PAR,R1       ;USE R1 TO BUILD PARAMETERS TO BE LOADED
4598 023024 042701 000200      BIC     #ODDPAR,R1   ;MAKE SURE ODD PARITY ISN'T SET
4599 023030 052701 000100      BIS     #PARITY,R1   ;MAKE SURE PARITY IS TURNED ON
4600 023034 012702 000001      MOV     #1,R2        ;USE R2 AS A LINE POINTER
4601 023040 030237 001364      1$:    BIT     R2,LINE     ;IS THIS A VALID LINE?
4602 023044 001411      BEQ     3$          ;IF NOT, SKIP TO THE NEXT LINE
4603 023046 032701 000001      BIT     #BIT0,R1     ;IS THIS LINE AN ODD LINE?
4604 023052 001402      BEQ     2$          ;IF IT'S EVEN, USE EVEN PARITY
4605 023054 052701 000200      BIS     #ODDPAR,R1   ;IF IT'S ODD, USE ODD PARITY
4606 023060 010177 156766      2$:    MOV     R1,@DZLPR ;LOAD THE LINE PARAMETER REGISTER
4607 023064 042701 000200      BIC     #ODDPAR,R1   ;SET UP THE NEXT PARITY TO EVEN
4608 023070 005201      3$:    INC     R1          ;POINT TO THE NEXT LINE
4609 023072 106302      ASLB   R2           ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
4610 023074 103361      BCC    1$          ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
4611 023076 005037 001372      CLR     SAVLIN       ;CLEAR THE LINE NUMBER INDICATOR
4612 023102 005002      CLR     R2          ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
4613 023104 005003      CLR     R3          ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
4613 023106 012737 000040 001376    MOV     #40,XMTCNT   ;TRANSMIT A BINARY COUNT PATTERN(00-40)

```

CZDZA-FO  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 95  
CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0094

```

4614 023114 012700 001422      MOV    #TDO,RO      ;POINT TO THE DATA AREA
4615 023120 005020              CLR    (RO)+        ;CLEAR A DATA WORD
4616 023122 022700 001462      CMP    #STOP,RO    ;FINISHED ?
4617 023126 001374              BNE    .-6         ;NO
4618 023130 005000              CLR    RO          ;CLEAR OFFSET
4619 023132 012777 023410 156736 MOV    #XMTSRV,@DZTIV ;SET UP THE TRANSMITTER INTERRUPT VECTOR
4620 023140 012777 023232 156724 MOV    #PARESE,@DZRIV ;SET UP THE RECEIVER INTERRUPT VECTOR
4621 023146 013777 027060 156720 MOV    DZPRT,@DZ RIS ;SET THE INTERRUPT VECTOR STATUS
4622 023154 013777 027060 156716 MOV    DZPRT,@DZTIS ;SET TRANSMITTER INTERRUPT PRIORITY
4623 023162 052777 040140 156652 BIS    #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
4624 023170 113777 001364 156660 MOV    LINE,@DZTCR  ;ENABLE ALL SELECTED LINES
4625 023176 106427 000000      MTPS   #0          ;ALLOW INTERRUPTS
4626 023202 032777 000100 156632 5$: BIT    #RIE,@DZCSR  ;WHEN RX DONE; RIE WILL =0
4627 023210 001407              BEQ    6$         ;BR IF ALL DONE
4628 023212 005237 023404      INC    COUNT0
4629 023216 102771              BVS    5$
4630 023220 105237 023406      INCB   COUNT1
4631 023224 100366              BPL    5$
4632 023226 104011              ERROR  11        ;*RX FAILED TO FINISH (INTERRUPT)
4633 023230 104400              6$:  ADVANCE    ;ADVANCE LOOP

```

```

4634
4635
4636
4637 023232 017704 156610
4638 023236 010401
4639 023240 000301
4640 023242 042701 177770
4641 023246 010137 001372
4642 023252 005704
4643 023254 100401
4644 023256 104023
4645 023260 006301
4646 023262 032704 010000
4647 023266 001013
4648 023270 013737 002046 001400
4649 023276 010405
4650 023300 042705 000377
4651 023304 156105 001442
4652 023310 052705 110000
4653 023314 104006
4654 023316 126104 001442
4655 023322 001407
4656 023324 116105 001442
4657 023330 042705 177400
4658 023334 042704 177400
4659 023340 104005
4660 023342 005261 001442
4661 023346 005203
4662 023350 005037 023404
4663 023354 005037 023406
4664 023360 032777 040000 156454
4665 023366 001005
4666 023370 020203
4667 023372 001003
4668 023374 042777 000100 156440
4669 023402 000002
4670 023404 000000
4671 023406 000000

```

```

;RECEIVER SERVICE ROUTINE(PARITY TEST ONLY)
PARESE: MOV @DZRBUF,R4 ;GET THE CHARACTER
MOV R4,R1 ;COPY THE RECEIVED INFORMATION
SWAB R1 ;GET THE LINE NUMBER IN THE LOWER BYTE
BIC #^C<7>,R1 ;ISOLATE THE LINE NUMBER
MOV R1,SAVLIN ;FILL LOC. FOR ERROR PRINTOUT
TST R4 ;WAS DATA VALID?
BMI 10$ ;BRANCH IF YES
ERROR 23 ;ERROR - DATA VALID NOT SET!
10$: ASL R1 ;ALIGN IT ON A WORD BOUNDARY
BIT #PARER,R4 ;PARITY ERROR SHOULD BE SET. IS IT?
BNE 11$ ;IF SO, GO CHECK CHARACTER
MOV DZRBUF,REGIST ;SET UP FOR THE ERROR MESSAGE
MOV R4,R5
BIC #377,R5
BISB TRO(R1),R5 ;GET THE CORRECT CHARACTER
BIS #DVALID!PARER,R5 ;BUILD WHAT WAS EXPECTED
ERROR 6 ;*ERROR- DID NOT GET CORRECT INFORMATION
11$: CMPB TRO(R1),R4 ;CHECK THE CHARACTER. IS IT CORRECT?
BEQ 12$ ;IF SO, GO SET UP NEXT CHARACTER
MOVB TRO(R1),R5 ;LOAD THE CHARACTER FOR ERROR REPORTING
BIC #^C<377>,R5 ;CLEAR SIGN EXTEND
BIC #^C<377>,R4 ;REMOVE THE JUNK FROM R4, THE ACTUAL CHARACTER
ERROR 5 ;DATA ERROR
12$: INC TRO(R1) ;SET UP THE NEXT CHARACTER
INC R3 ;ADD TO THE TOTAL RECEIVED COUNT
CLR COUNT0 ;RESET COUNTERS TO NEXT
CLR COUNT1 ;RECEIVER INTERRUPT
BIT #TIE,@DZCSR ;ARE TRANSMISSIONS DONE?
BNE 13$ ;IF NO, GO RECEIVE SOME MORE
CMP R2,R3 ;ARE ALL CHARACTERS RECEIVED?
BNE 13$ ;IF NO, GO RECEIVE SOME MORE
BIC #RIE,@DZCSR ;DISABLE RECEIVER INTERRUPTS
13$: RTI ;GO BACK TO RECEIVER WAIT LOOP
COUNT0: 0
COUNT1: 0

```

```

4672
4673
4674 ;TRANSMITTER INTERRUPT SERVICE
4675 ;-----
4676

```

```

4677 023410 117701 156430
4678 023414 100411
4679 023416 013700 001372
4680 023422 042701 177770
4681 023426 010137 001372
4682 023432 104003
4683 023434 010037 001372
4684 023440 042701 177770
4685 023444 006301
4686 023446 116177 001422 156412
4687 023454 005261 001422
4688
4689

```

```

XMTSRV: MOVB @HDZCSR,R1 ;GET THE LINE NUMBER. IS THE TRANSMITTER
BMI 1$ ;REALLY READY? IF SO, GO LOAD THE CHARACTER
MOV SAVLIN,R0 ;ADJUST LOCATION SAVLIN
BIC #^C<7>,R1 ;ISOLATE THE LINE NUMBER
MOV R1,SAVLIN ;FOR ERROR PRINTOUT
ERROR 3 ;*TRANSMITTER NOT READY- FALSE INTERRUPT
1$: MOV R0,SAVLIN ;RESET SAVLIN TO PREVIOUS VALUE
BIC #^C<7>,R1 ;ISOLATE THE LINE NUMBER
ASL R1 ;MAKE SURE IT REFERENCES A WORD BOUNDARY
MOVB TDO(R1),@DZTDR ;LOAD THE CURRENT CHARACTER FOR THIS LINE
INC TDO(R1) ;SET UP NEXT CHARACTER FOR THIS LINE

```

\*\*\*\*\*

CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052) 26-NOV-79 11:03

26-NOV-79 11:06 PAGE 97  
CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0096

```

4690 ;*****
4691 023460 023761 001376 001422      CMP      XMTCNT,TD0(R1)  ;HAVE WE DONE ALL PATTERNS ON THIS LINE?
4692 023466 001015                      BNE      4$             ;IF NOT, KEEP ON TRANSMITTING
4693 023470 012700 000001              MOV      #1,R0         ;SET UP A DESELECTION POINTER
4694 023474 006201                      ASR      R1            ;GET THE LINE NUMBER AGAIN
4695 023476 005301                      2$:     DEC      R1     ;REDUCE THE COUNT. WAS THIS THE LINE?
4696 023500 100402                      BMI      3$           ;IF SO, GO DISABLE THE ENABLE BIT FOR IT
4697 023502 006300                      ASL      R0            ;MOVE THE POINTER TO THE NEXT LINE
4698 023504 000774                      BR       2$           ;GO CHECK THE NEXT LINE
4699 023506 140077 156344              3$:     BICB     R0,@DZTCR ;DISABLE THE LINE POINTED TO BY R0
4700 023512 001003                      BNE      4$           ;IF MORE LINES ARE ACTIVE, GO CONTINUE TRANSMIT
4701 023514 042777 040000 156320      BIC      #TIE,@DZCSR  ;IF NOT, DISABLE TRANSMITTER INTERRUPTS
4702 ;*****
4703 023522 005202                      4$:     INC      R2     ;UP THE NUMBER OF TRANSMISSIONS (REV. F0)
4704 ;*****
4705 023524 000002                      RTI              ;RETURN TO THE TIMING LOOP
4706
4707 ; RELATIVE TIME BUILDING ROUTINE
4708 ; -----
4709
4710 023526 012737 000004 001222 BUILD: MOV      #4,$TMP2    ;ROTATE 4 BITS BACK INTO $TMP1
4711 023534 006037 001224 1$:     ROR      $TMP3    ;GET THE BITS FROM $TMP3, THE HIGH BYTE
4712 023540 006037 001220              ROR      $TMP1    ;OF THE RELATIVE TIME COUNTER. PUT THEM BACK
4713 023544 005337 001222              DEC      $TMP2    ;INTO $TMP1 USING THE CARRY BIT WITH
4714 ; ROTATE INSTRUCTIONS
4715 023550 001371                      BNE      1$         ;REDUCE COUNT. ALL BITS BACK? IF NOT, GET MORE
4716 023552 000207                      RTS      PC         ;RETURN TO CALLING TEST
4717

```



```

4718                                     ;RECEIVER SERVICE ROUTINE
4719
4720 023554 105777 156262          RXISR1: TSTB   @DZCSP           ;IS THE RECEIVER REALLY READY?
4721 023560 100401                BMI     1$           ;IF SO, GO SERVICE IT
4722 023562 104004                ERROR   4            ;*ERROR- RECEIVER DONE FLAG ISN'T SET
4723 023564 017704 156256          1$:   MOV    @DZRBUL,R4       ;SAVE THE RECEIVER INFORMATION
4724 023570 100401                BMI     2$           ;IF IT WAS VALID, GO PROCESS IT
4725 023572 104023                ERROR   23           ;ERROR- DATA VALID WASN'T SET
4726 023574 032704 070000          2$:   BIT    #OVRRUN.FRMERR!PARER,R4 ;ARE ANY ERROR FLAGS SET?
4727 023600 001403                BEQ     3$           ;IF NOT, GO CONTINUE PROCESSING
4728 023602 013700 002046          MOV    DZRBUL,R0       ;SET UP FOR ERROR REPORTING
4729 023606 104002                ERROR   2            ;ERROR- RECEIVER ERROR FLAG SET
4730 023610 010401                3$:   MOV    R4,R1         ;COPY THE RECEIVER INFORMATION
4731 023612 000301                SWAB   R1             ;GET THE LINE NUMBER IN THE LOWER BYTE
4732 023614 042701 177770          BIC    #^C<7>,R1       ;ISOLATE THE LINE NUMBER
4733 023620 006301                ASL    R1             ;ALIGN IT ON A WORD BOUNDARY
4734 023622 120461 001442          CMPB   R4,TR0(R1)      ;IS THE CHARACTER WHAT IT SHOULD BE?
4735 023626 001413                BEQ     4$           ;IF SO,GO CONTINUE PROCESSING
4736 023630 116105 001442          MOVB   TR0(R1),R5      ;GET WHAT WAS EXPECTED FOR ERROR REPORTING
4737 023634 042705 177400          BIC    #^C<377>,R5     ;ELIMINATE PROPAGATED SIGN
4738 023640 042704 177400          BIC    #^C<377>,R4     ;ISOLATE THE ACTUAL CHARACTER
4739 023644 010137 001372          MOV    R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER ERROR
4740 023650 006237 001372          ASR    SAVLIN          ;ALIGN IT CORRECTLY FOR REPORTING
4741 023654 104005                ERROR   5            ;*DATA ERROR
4742 023656 005261 001442          4$:   INC    TR0(R1)       ;SET UP THE NEXT EXPECTED CHARACTER
4743 023662 005203                INC    R3             ;INCREMENT THE COUNT OF RECEIVED CHARACTERS
4744 023664 032761 000020 001442  BIT    #20,TR0(R1)     ;HAVE ALL CHARACTERS BEEN RECEIVED?
4745 023672 001402                BEQ     5$           ;IF NOT, GO RECEIVE SOME MORE
4746 023674 020203                CMP    R2,R3          ;HAVE WE RECEIVED ALL CHARACTERS?
4747 023676 001401                BEQ     6$           ;IF SO,GO DETERMINE THE TIMING
4748 023700 000002                5$:   RTI                    ;GO CONTINUE TIMING AND ALLOW INTERRUPTS
4749 023702 004737 023526          6$:   JSR    PC,BUILD      ;GET THE RELATIVE TIME (SIGNIFICANT BITS)
4750
4751 023706 013700 023766          MOV    OFFSET,R0       ;GET POINTER
4752 023712 013760 001220 002102  MOV    $TMP1,TMTBL(R0) ;SAVE THIS TEST'S TIME
4753 023720 005737 023766          TST    OFFSET         ;FIRST TEST?
4754 023724 001414                BEQ     7$           ;IF NOT, GO CHECK THE TIME
4755 023726 005740                TST    -(R0)         ;POINT TO THE PREVIOUS TIME TAKEN
4756 023730 026037 002102 001220  CMP    TMTBL(R0),$TMP1 ;IS THIS TIME WHAT IT SHOULD BE?
4757 023736 101007                BHI    7$           ;IF SO, GO TO THE NEXT TEST
4758 023740 016005 002102          MOV    TMTBL(R0),R5    ;PLACE WHAT WAS EXPECTED IN R5
4759 023744 010137 001372          MOV    R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER
4760 023750 006237 001372          ASR    SAVLIN          ;MAKE SURE IT'S THE LINE NUMBER
4761 023754 104021                ERROR   21           ;TIMING ERROR
4762 023756 042777 000140 156056  7$:   BIC    #RIE.MSENAB,@DZCSR ;DISABLE THE DEVICE
4763 023764 000002                RTI                    ;RETURN TO THE PROGRAM
4764 023766 000000                OFFSET: 0

```

4765  
4766  
4767  
4768  
4769  
4770  
4771  
4772  
4773  
4774  
4775  
4776  
4777  
4778  
4779  
4780  
4781  
4782  
4783  
4784  
4785  
4786  
4787  
4788  
4789  
4790  
4791  
4792  
4793  
4794  
4795  
4796  
4797  
4798  
4799  
4800  
4801  
4802  
4803  
4804  
4805  
4806  
4807  
4808  
4809  
4810  
4811  
4812  
4813  
4814  
4815  
4816  
4817  
4818  
4819  
4820

:DZ11 ECHO/CABLE TEST

:\*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 STARTED

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

XSTART: MOV #STACK,SP ;SET UP PROCESSOR STACK  
MTPS #PR7 ;LOCK OUT INTERRUPTS  
MOV #XSTART,\$LPADR ;SET UP IN CASE OF POWER FAIL  
CLR STFLG ;CLEAR TEST START FLAG  
CLR \$PASS ;CLEAR PASS COUNT  
CLR \$ERTTL ;CLEAR ERROR COUNT  
CLRB \$ERFLG ;CLEAR ERRCR FLAG  
CLR LAST ;CLEAR LAST ERROR PC

001126

023770 012706 001120  
023774 106427 000340  
024000 012737 023770  
024006 005037 026164  
024012 005037 001242  
024016 005037 001132  
024022 105037 001123  
024026 005037 026170

```

4821 024032 032777 000001 155120 VEC1: BIT #SW00,@SWR ;IF SW00-1, GET NEW VECTOR
4822 024040 001465 BEQ OTHER ;AND CSR
4823 024042 012701 000300 VEC2: MOV #300,R1
4824 024046 012702 000302 MOV #302,R2
4825 024052 010221 1$: MOV R2,(R1)+ ;RESTORE TRAPCATCHER
4826 024054 005022 CLR (R2)+ ;IN FLOATING VECTOR AREA
4827 024056 022122 CMP (R1)+,(R2)+ ;UPDATE THE POINTERS
4828 024060 020127 001000 CMP R1,#1000
4829 024064 001372 BNE 1$
4830 024066 104403 INSTR ;INPUT ADDRESS OF DEVICE VECTOR
4831 024070 026216 MVECTOR ;MESSAGE 'VECTOR ADDRESS-'
4832 024072 104405 PARAM ;CONVERT STRING TO OCTAL
4833 024074 000300 300 ;LOW LIMIT
4834 024076 000770 770 ;HIGH LIMIT
4835 024100 002072 DZRIV ;LOCATIONS TO BE FILLED
4836 024102 003 .BYTE 3 ;LSB MASK
4837 024103 004 .BYTE 4 ;NUMBER OF LOCATIONS
4838 024104 104403 INSTR ;INPUT ADDRESS OF DEVICE CSR
4839 024106 026240 MREGAD ;MESSAGE 'CONTROL REGISTER ADDRESS-'
4840 024110 104405 PARAM ;CONVERT STRING TO OCTAL
4841 024112 160000 160000 ;LOW LIMIT
4842 024114 163700 163700 ;HIGH LIMIT
4843 024116 002042 DZCSR ;LOCATIONS TO BE FILLED
4844 024120 007 .BYTE 7 ;LSB MASK
4845 024121 001 .BYTE 1 ;NUMBER OF LOCATIONS
4846 024122 013737 002042 002046 MOV DZCSR,DZRBUF ;BEGIN BUILDING DEVICE ADDRESSES
4847 024130 062737 000002 002046 ADD #2,DZRBUF ;FORM THE READ BUFFER ADDRESS
4848 024136 013737 002046 002052 MOV DZRBUF,DZLPR ;REMEMBER THAT THIS IS ALSO LINE PARAMETER REG.
4849 024144 013737 002046 002056 MOV DZRBUF,DZTCR ;BEGIN BUILDING TRANSMITTER CONTROL REGISTER
4850 024152 062737 000002 002056 ADD #2,DZTCR ;FORM THE TRANSMITTER CONTROL REGISTER POINTER
4851 024160 013737 002056 002060 MOV DZTCR,HDZTCR
4852 024166 005237 002060 INC HDZTCR
4853 024172 013737 002056 002066 MOV DZTCR,DZTDR ;BEGIN FORMING TRANSMITTER DATA REGISTER
4854 024200 062737 000002 002066 ADD #2,DZTDR ;FORM THE TRANSMITTER DATA REGISTER
4855 024206 013737 002066 002062 MOV DZTDR,DZMSR
4856 024214 032777 000002 154736 OTHER: BIT #SW01,@SWR ;RESELECT OF TEST?
4857 024222 001427 BEQ XBEGIN ;IF NOT, SKIP ASKING WHICH ONE
4858 024224 104403 INSTR ;INPUT WHICH TEST YOU ARE RUNNING
4859 024226 026424 MWHICH ;ECHO OR CABLE
4860 024230 104416 PAWCH ;SET FLAG
4861 024232 026162 WCHFLG ;THIS FLAG
4862 024234 104403 BAUD: INSTR ;INPUT BAUD RATE
4863 024236 026346 MSPEED ;MESSAGE 'BAUD RATE-'
4864 024240 104415 PARM ;CONVERT DECIMAL STRING TO OCTAL
4865 024242 000062 50. ;LOW LIMIT
4866 024244 022600 9600. ;HIGH LIMIT
4867 024246 026200 LINESP ;LOCATION TO BE FILLED
4868 024250 000 .BYTE 0 ;LSB MASK
4869 024251 001 .BYTE 1 ;NUMBER OF LOCATIONS
4870 024252 104413 LINEX: DEVICE.CLR ;CLEAR DEVICE
4871 024254 005037 026164 CLR STFLG ;CLEAR PROGRAM START FLAG
4872 024260 104403 INSTR ;INPUT LINE NUMBER
4873 024262 026336 MLINE ;MESSAGE 'LINE NUMBER-'
4874 024264 104405 PARAM ;CONVERT STRING TO OCTAL
4875 024266 000000 0 ;LOW LIMIT
4876 024270 000007 7 ;HIGH LIMIT

```

```

4877 024272 001372          SAVLIN          ;LOCATION TO BE FILLED
4878 024274      000          .BYTE 0          ;LSB MASK
4879 024275      001          .BYTE 1          ;NUMBER OF LOCATIONS
4880 024276 004537 025766    JSR      R5,SET
4881
4882 024302 106427 000340    XBEGIN: MTPS      #PR7          ;LOCK OUT INTERRUPTS
4883 024306 012706 001120    MOV      #STACK,SP ;SET UP PROCESSOR STACK
4884 024312 005037 026166    CLR      LOCKUP     ;CLEAR TIMEOUT
4885 024316 005737 026162    TST      WCHFLG     ;ECHO OR CABLE TEST ?
4886 024322 001413          BEQ      2$         ;ECHO
4887 024324 012737 025040 001126 MOV      #TEST2,$LPADR ;CABLE TEST
4888 024332 005737 026164    TST      STFLG      ;ARE YOU LOOPING ?
4889 024336 001017          BNE      1$         ;YES
4890 024340 005137 026164    COM      STFLG      ;NO
4891 024344 104402 026517    TYPE     ,MCABLE   ;TYPE CABLE TEST
4892 024350 000412          BR       1$
4893 024352 012737 024402 001126 2$: MOV      #TEST1,$LPADR ;SFT UP ECHO TEST
4894 024360 005737 026164    TST      STFLG      ;ARE YOU LOOPING ?
4895 024364 001004          BNE      1$         ;YES
4896 024366 005137 026164    COM      STFLG      ;NO
4897 024372 104402 026472    TYPE     ,MTERM    ;TYPE ECHO TEST
4898 024376 000177 154524    1$: JMP      @SLPADR  ;START TESTING
4899
4900          ;THIS TEST WILL ACCEPT 1 CHARACTER AT A TIME
4901          ;:(IN INTERRUPT MODE) AND TRANSMIT THAT SAME CHARACTER,
4902          ;ONE LINE AT A TIME, ANY LINE 0 THRU 7 (OCTAL)
4903 024402 104413          TEST1: DEVICE.CLR  ;CLEAR DZ11
4904 024404 012737 000001 001122 MOV      #1,$STNM   ;
4905 024412 013777 026206 155436 MOV      NUMTCR,@DZTCR ;SET TCR BIT
4906 024420 013737 026204 001366 MOV      NUMLIN,PAR  ;SET PARAMETERS
4907 024426 053737 026202 001366 BIS      SPEED,PAR  ;SET BAUD RATE
4908 024434 013777 001366 155410 MOV      PAR,@DZLPR ;LOAD PARAM.
4909 024442 012777 000040 155372 MOV      #MSENAB,@DZCSR ;SET SCANN ENABLE
4910 024450 005004          CLR      R4
4911 024452 012705 026534          MOV      #MQUICK,R5 ;SET MESSAGE BUFFER
4912 024456 005777 155360    3$: TST      @DZCSR  ;TRDY?
4913 024462 100404          BMI      2$         ;BR IF YES
4914 024464 104414          DELAY
4915 024466 005304          DEC      R4
4916 024470 001372          BNE      3$
4917 024472 104003          ERROR    3
4918 024474 005004          2$: CLR      R4
4919 024476 112577 155364    MOV      (R5)+,@DZTDR ;LOAD CHAR
4920 024502 001365          BNE      3$
4921 024504 004737 007360    JSR      PC,SERV.G  ;<^G>?
4922 024510 122777 000377 154442 CMPB     #377,@SWR  ;RE-DO QUICK BROWN?
4923 024516 001731          BEQ      TEST1     ;BR IF REPEAT PATTERN
4924 024520 104413          DEVICE.CLR
4925 024522 106427 000340    MTPS     #PR7
4926 024526 012737 025476 001360 MOV      #XEOP,NEXT ;LOCK OUT INTERRUPTS
4927 024534 104413          DEVICE.CLR
4928 024536 013737 026204 001366 MOV      NUMLIN,PAR  ;SELECT LINE # & SET INTERRUPT ENABLE
4929 024544 053737 026202 001366 BIS      SPEED,PAR  ;SET LINE SPEED AND
4930          ;CHARACTER LENGTH (TRANS. & REC.)
4931 024552 052737 010000 001366 BIS      #RCVON,PAR  ;MAKE SURE RECEIVER IS TURNED ON
4932 024560 013777 001366 155264 MOV      PAR,@DZLPR ;LOAD THE LINE PARAMETER REGISTER

```

```

4933 024566 012777 024642 155276      MOV    #INTSVC,@DZRIV ;SET UP INTERRUPT SERVICE
4934 024574 013777 026210 155272      MOV    PRIO,@DZ RIS  ;AND LEVEL
4935 024602 106437 027062                MTPS   @#LESS1      ;ALLOW INTERRUPTS
4936 024606 012777 000140 155226      MOV    #RIE!MSENAB,@DZCSR ;SET RECEIVER INTERRUPT ENABLE
4937 024614 104402 026364                TYPE  ,MCHAR        ;TYPE 'ANY CHARACTER'
4938 024620 105777 154340                1$:   TSTB   @STKS    ;IF SOMEBODY HITS A KEY- GET NEW LINE #
4939 024624 100375                BPL    1$           ;LOOP HERE
4940 024626 005777 154334                TST   @STKB        ;CLEAR CHAR
4941 024632 004737 007360                JSR   PC,SERV.G    ;MAKE SURE IT WASN'T <^G>
4942 024636 000137 024252                JMP   LINEX        ;
4943
4944
4945

```

```

4946 024642 105777 155174      INTSVC: TSTB   @DZCSR      ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
4947 024646 100401                BMI   .+4           ;TEST REC. FLAG
4948 024650 104004                ERROR 4            ;ERROR - INTERRUPT NOT CAUSED BY FLAG
4949 024652 017737 155170 026212      MOV    @DZRBUF,RECDAT
4950 024660 100401                BMI   .+4
4951 024662 104023                ERROR 23           ;NON- VALID CHARACTER
4952 024664 032737 020000 026212      BIT    #BIT13,RECDAT ;CHECK FOR FRAMING ERROR
4953 024672 001401                BEQ   .+4           ;BR IF NO ERROR
4954 024674 104025                ERROR 25           ;EITHER SOMEBODY HIT THE
4955                                ;'BREAK KEY' OR YOU HAVE AN ERROR.
4956 024676 113737 026212 026214      MOVSB RECDAT,TBUF   ;MOVE CHARACTER TO OUTPUT AREA
4957 024704 113737 026212 010620      MOVSB RECDAT,INBUF  ;MOVE CHARACTER TO CHECK FOR ^C
4958 024712 042737 177600 010620      BIC   #^C<177>,INBUF ;STRIP JUNK PLUS PARITY
4959 024720 042737 174377 026212      BIC   #174377,RECDAT ;SAVE ONLY LINE NUMBER
4960 024726 000337 026212                SWAB  RECDAT
4961 024732 023737 001372 026212      CMP   SAVLIN,RECDAT ;DOES THE LINE # COMPARE?
4962 024740 001401                BEQ   .+4
4963 024742 104015                ERROR 15           ;*WRONG LINE NUMBER
4964 024744 012777 000040 155070      MOV    #MSENAB,@DZCSR ;START THE TRANSMITTERS SCANNER
4965 024752 123727 010620 000003      CMPB  INBUF,#3     ;IS IT A ^C ?
4966 024760 001004                BNE   1$           ;NO
4967 024762 104413                DEVICE.CLR
4968 024764 012716 025476                MOV   #XEOP,(SP)   ;CRUNCH STACK
4969 024770 000002                RTI
4970 024772 005003                1$:   CLR    R3      ;INITIALIZE DELAY
4971 024774 013777 026206 155054      MOV    NUMTCR,@DZTCR ;ENABLE THE LINE
4972 025002 005777 155034                10$:  TST   @DZCSR     ;TRANSMITTER READY?
4973 025006 100403                BMI   2$           ;IF YES BRANCH
4974 025010 005203                INC   R3           ;INCREMENT DELAY
4975 025012 001373                BNE   10$         ;DELAY DONE?
4976 025014 104003                ERROR 3            ;TRANSMIT READY NOT SET!
4977 025016 113777 026214 155042      MOVSB TBUF,@DZTDR   ;TRANSMIT THE CHARACTER
4978 025024 012777 000140 155010      MOV    #RIE.MSENAB,@DZCSR ;RESTART THE RECEIVER
4979 025032 005077 155020                CLR   @DZTCR      ;CLEAR TCR BIT
4980 025036 000002                RTI
4981
4982
4983

```

```

4984                                ;THIS TEST TRANSMITS A BINARY COUNT PATTERN
4985                                ;VIA INTERRUPT MODE TO THE RECEIVER
4986                                ;...THE LINE UNDER TEST MUST BE TERMINATED WITH THE TEST CONNECTOR
4986 025040 106427 000340      TEST2: MTPS   #PR7   ;DISABLE INTERRUPTS
4987 025044 012737 000002 001122      MOV    #2,$STSNM
4988 025052 012737 025476 001360      MOV    #XEOP,NEXT

```

CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 103  
CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0102

```

4989 025060 104413          DEVICE.CLR
4990                               ;*TEST TO VERIFY THAT SETTING DTR FOR A GIVEN LINE
4991                               ;*WILL BRING UP 'CO' AND 'RING' FOR THE SAME LINE
4992                               ;*THE DIST PNL MUST HAVE JUMPER FROM DTR TO ROST TO SEND
4993                               ;*IN ORDER FOR THIS TEST TO WORK!
4994 025062 012737 025070 001362  MOV    #1$,LOCK           ;LOOP
4995 025070 113777 026206 154762 1$:  MOVB   NUMTCR,@HDZTCR     ;SET DTR
4996 025076 005005                CLR    R5                 ;
4997 025100 153705 026206        BISB   NUMTCR,R5          ;BUILD EXPECTED
4998 025104 000305                SWAB  R5                 ;PUT IN HIGH BYTE
4999 025106 153705 026206        BISB   NUMTCR,R5          ;
5000 025112 104414                DELAY                ;WAIT FOR CABLE DELAY
5001 025114 017704 154742        MOV    @DZMSR,R4       ;READY MODEM BITS
5002 025120 020504                CMP   R5,R4           ;ARE THEY OK?
5003 025122 001401                BEQ   2$              ;BR IF YES
5004 025124 104022                ERROR  22              ;IS THE TEST CONNECTOR ON?
5005                               ;HAS RIGHT LINE BEEN SELECTED?
5006                               ;IF SO- YOU HAVE A PROBLEM!
5007                               ;MODEM BITS NOT RIGHT
5008 025126 104401                2$:  SCOP1                ;LOOP
5009 025130 104413                3$:  DEVICE.CLR          ;INIT DZ11
5010 025132 013737 026202 001366  MOV    SPEED,PAR       ;SET LINE SPEED
5011 025140 053737 026204 001366  BIS    NUMLIN,PAR      ;SELECT LINE # & REC. INTERRUPT ENABLE
5012 025146 052737 010000 001366  BIS    #RCVON,PAR      ;ENABLE THE RECEIVER FOR THIS LINE
5013 025154 052777 040140 154660  BIS    #TIE!RIE!MSENAB,@DZCSR ;SET TRANSMITTER INTERRUPT ENABLE
5014 025162 012777 025276 154702  MOV    #INTREC,@DZRIV  ;SET UP INTR SERVICE
5015 025170 013777 026210 154676  MOV    PRIO,@DZRIIS    ;SET UP LEVEL
5016 025176 012777 025456 154672  MOV    #INTRAN,@DZTIV  ;SET UP INTR SERVICE
5017 025204 013777 026210 154666  MOV    PRIO,@DZTIS     ;SET UP LEVEL
5018 025212 005001                CLR    R1              ;RX DATA POINTER- SET TO 0
5019 025214 005002                CLR    R2              ;TX DATA POINTER- SET TO 0
5020 025216 013777 001366 154626  MOV    PAR,@DZLPR      ;SET THE PARAMETERS AND TURN ON RECEIVER
5021 025224 106437 027062                MTPS  @WLESS1         ;ALLOW INTERRUPTS
5022 025230 013777 026206 154620  MOV    NUMTCR,@DZTCR   ;SET UP TCR BIT
5023
5024                               ;YOU RETURN HERE AFTER EVERY RECEIVER INTERRUPT
5025 025236 105777 153722  SPIN:  TSTB   @STKS           ;IF SOMEBODY HITS A KEY- GET A NEW LINE #
5026 025242 100006                BPL   1$               ;BR IF NO KEY HIT
5027 025244 005777 153716                TST   @STKB           ;CLEAR CHAR
5028 025250 004737 007360                JSR   PC,SERV.G       ;MAKE SURE IT WASN'T <^G>
5029 025254 000137 024252                JMP   LINEX           ;SW02=1
5030 025260 005237 026166                1$:  INC    LOCKUP        ;INC TIMEOUT FLAG
5031 025264 001364                BNE   SPIN            ;IF NOT 0 RETURN SPINNING
5032 025266 104011                ERROR  11             ;*RECEIVER FAILED TO INTERRUPT CHECK CABLE/TERMINATOR
5033 025270 104413                QUILTS: DEVICE.CLR    ;
5034 025272 000137 025476                JMP   XEOP             ;CALL FOR END OF PASS
5035 025276 005037 026166                INTREC: CLR    LOCKUP   ;CLEAR TIMEOUT FLAG
5036 025302 105777 154534                TSTB  @DZCSR          ;TEST REC DONE
5037 025306 100401                BMI   .+4             ;YES
5038 025310 104004                ERROR  4              ;*FALSE INTERRUPT
5039 025312 017737 154530 026212        MOV    @DZRBUF,RECDAT ;SAVE WORD
5040 025320 100401                BMI   .+4             ;
5041 025322 104023                ERROR  23             ;*NON VALID CHARACTER
5042 025324 032737 040000 026212        BIT   #BIT14,RECDAT   ;DATA OVERRUN ?
5043 025332 001401                BEQ   .+4             ;NO
5044 025334 104024                ERROR  24             ;*YES

```

CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 104  
CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0103

5045	025336	032737	020000	026212	BIT	#BIT13,RECDAT	:FRAMING ERROR ?
5046	025344	001401			BEQ	.+4	:NO
5047	025346	104025			ERROR	25	:*YES
5048	025350	032737	010000	026212	BIT	#BIT12,RECDAT	:PARITY ERROR ?
5049	025356	001401			BEQ	.+4	:NO
5050	025360	104026			ERROR	26	:*YES
5051	025362	110105			MOVB	R1,R5	:SET EXPECTED
5052	025364	042705	177400		BIC	#^C<377>,R5	:CLEAR HIGH BYTE
5053	025370	113704	026212		MOVB	RECDAT,R4	:GET FOUND
5054	025374	042704	177400		BIC	#^C<377>,R4	:CLEAR HIGH BYTE
5055	025400	020504			CMP	R5,R4 ;OK?	
5056	025402	001401			BEQ	.+4	
5057	025404	104005			ERROR	5	:DATA ERROR
5058	025406	042737	174377	026212	BIC	#174377,RECDAT	:SAVE ONLY LINE NUMBER
5059	025414	000337	026212		SWAB	RECDAT	
5060	025420	023737	001372	026212	CMP	SAVLIN,RECDAT	:DOES THE LINE # COMPARE ?
5061	025426	001401			BEQ	.+4	:YES
5062	025430	104015			ERROR	15	:*WRONG LINE #
5063	025432	120127	000377		CMPB	R1,#377	:LAST CHARACTER ?
5064	025436	001003			BNE	1\$	:NO
5065	025440	012716	025270		MOV	#QUITS,(SP)	:CRUNCH STACK
5066	025444	000403			BR	2\$	
5067	025446	105201			1\$:	INCB R1	:UPDATE EXPECTED DATA
5068	025450	012716	025236		MOV	#SPIN,(SP)	:CRUNCH STACK
5069	025454	000002			2\$:	RTI	
5070							
5071	025456	005777	154360		INTRAN:	TST @DZCSR ;TEST TRANSMIT FLAG	
5072	025462	100401			BMI	.+4	
5073	025464	104003			ERROR	3	:*FALSE INTERRUPT
5074	025466	110277	154374		MOVB	R2,@DZTDR	:TRANSMIT A CHARACTER
5075	025472	105202			INCB	R2	:UPDATE TX DATA
5076	025474	000002			RTI	;RETURN	

```

5077
5078                                     :END OF PASS
5079                                     :RESTART TEST
5080
5081 025476 104402 XEOP: TYPE ;TYPE NAME OF TEST
5082 025500 026274 MPASS
5083 025502 005037 026170 CLR LAST ;CLEAR LAST ERROR PC
5084 025506 105037 001123 CLRB $ERFLG ;CLEAR ERROR FLAG
5085 025512 000137 024302 RSTRT: JMP XBEGIN
5086
5087                                     :CONVERT DECIMAL ASCII STRING TO OCTAL
5088 025516 011605 .PARMD: MOV (SP),R5
5089 025520 012537 025702 MOV (R5)+,6$
5090 025524 012537 025704 MOV (R5)+,7$
5091 025530 012537 025706 MOV (R5)+,8$
5092 025534 112537 025710 MOV (R5)+,9$
5093 025540 112537 025711 MOV (R5)+,10$
5094 025544 010516 MOV R5,(SP)
5095 025546 005005 2$: CLR R5
5096 025550 012704 010620 MOV #INBUF,R4
5097 025554 122714 000015 CMPB #15,(R4)
5098 025560 001424 BEQ 3$
5099 025562 121427 000060 1$: CMPB (R4),#'0
5100 025566 002421 BLT 3$
5101 025570 121427 000071 CMPB (R4),#'9
5102 025574 003016 BGT 3$
5103 025576 142714 000060 BICB #'0,(R4)
5104 025602 005002 CLR R2
5105 025604 152402 BISB (R4)+,R2
5106 025606 060205 ADD R2,R5
5107 025610 122714 000015 CMPB #15,(R4)
5108 025614 001410 BEQ 4$
5109 025616 006305 ASL R5 ;X2
5110 025620 010502 MOV R5,R2 ;SAVE X2
5111 025622 006305 ASL R5 ;X4
5112 025624 006305 ASL R5 ;X8
5113 025626 060205 ADD R2,R5 ;TIMES 10
5114 025630 000754 BR 1$
5115 025632 104404 3$: INSTER
5116 025634 000744 BR 2$
5117
5118                                     :TEST TO SEE IF NUMBER IS WITHIN LIMITS
5119
5120 025636 020537 025704 4$: CMP R5,7$
5121 025642 101373 BHI -,3$
5122 025644 020537 025702 CMP R5,6$
5123 025650 103770 BLO 3$
5124 025652 133705 025710 BITB 9$,R5
5125 025656 001365 BNE 3$
5126
5127                                     :STORE NUMBER AT SPECIFIED ADDRESS
5128
5129 025660 013704 025706 5$: MOV 8$,R4
5130 025664 010524 MOV R5,(R4)+
5131 025666 062705 000002 ADD #2,R5
5132 025672 105337 025711 DECB 10$

```



```

5133 025676 001372          BNE      5$
5134 025700 000002          RTI
5135 025702 000000          6$:      0
5136 025704 000000          7$:      0
5137 025706 000000          8$:      0
5138 025710          000          9$:      .BYTE 0
5139 025711          000          10$:     .BYTE 0
5140
5141
5142
5143          ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
5144          ;BUFFER TO THE CHARACTERS 'E' AND 'C'.
5145          ;IF THE CHARACTER IS 'E' CLEAR THE FLAG
5146          ;IF THE CHARACTER IS 'C' SET THE FLAG
5147 025712 017605 000000      .PAWCH:MOV  @ (SP),R5
5148 025716 142737 000040 010620  BICB    #40,INBUF      ;SET FOR LOWER CASE INPUT
5149 025724 122737 000105 010620  CMPB    #'E',INBUF     ;IS IT 'E' ?
5150 025732 001002          BNE     1$
5151 025734 105015          CLRB    (R5)           ;000
5152 025736 000406          BR      2$
5153 025740 122737 000103 010620  1$:    CMPB    #'C',INBUF     ;IS IT 'C' ?
5154 025746 001005          BNE     3$
5155 025750 112715 177777      MOVB    #-1,(R5)       ;3177
5156 025754 062716 000002      2$:    ADD     #2,(SP)
5157 025760 000002          RTI
5158 025762 104404          3$:    INSTER          ;RETRY
5159 025764 000752          BR      .PAWCH
5160
5161
5162
5163          ;THIS ROUTINE CONVERTS LINE SPEED (LINESP) AND
5164          ;LINE NUMBER (SAVLIN) FOR DZLPR, DZTCR AND DZCSK
5165          ;REGISTER USAGE.
5166
5167 025766 013737 001372 026204  SET:    MOV     SAVLIN,NUMLIN ;SAVE SAVLIN
5168 025774 013700 001372          XTCR0: MOV     SAVLIN,R0      ;COPY THE LINE NUMBER FOR LOOP CONTROL
5169 026000 005037 026206          CLR     NUMTCR          ;SET A DEFAULT OF LINE 0 OR NO LINES
5170 026004 012702 000001          MOV     #1,R2          ;SET A BIT POINTER TO THE FIRST LINE
5171 026010 005300          XTCR1: DEC     R0        ;REDUCE THE INDICATOR.IS IT MINUS YET?
5172 026012 100402          BMI    SET1           ;IF SO, R2 POINTS TO THE RIGHT LINE
5173 026014 006302          ASL    R2             ;IF NOT, MOVE THE POINTER TO THE NEXT LINE
5174 026016 000774          BR     XTCR1          ;GO SEE IF THIS LINE IS THE ONE
5175 026020 012701 026062          SET1:  MOV     #TABLE2,R1
5176 026024 010237 026206          MOV     R2,NUMTCR      ;COPY THE CORRECT BIT POINTER
5177 026030 022137 026200          1$:    CMP     (R1)+,LINESP
5178 026034 001407          BEQ    2$
5179 026036 005721          TST    (R1)+          ;IS IT THE END OF TABLE?
5180 026040 001373          BNE    1$             ;NO
5181 026042 104402 026310          TYPE   ,MINVAL        ;INVALID BAUD RATE,BEGIN AGAIN
5182 026046 012705 024234          MOV     #BAUD,R5      ;JUMP TO BAUD THRU R5
5183 026052 000402          BR     3$
5184 026054 011137 026202          2$:    MOV     (R1),SPEED ;SET UP BAUD RATE
5185 026060 000205          3$:    RTS     R5
5186
5187
5188

```

5189  
5190 026062 000062  
5191 026064 010070  
5192 026066 000113  
5193 026070 010470  
5194 026072 000156  
5195 026074 011070  
5196 026076 000207  
5197 026100 011470  
5198 026102 000226  
5199 026104 012070  
5200 026106 000454  
5201 026110 012430  
5202 026112 001130  
5203 026114 013030  
5204 026116 002260  
5205 026120 013430  
5206 026122 003410  
5207 026124 014030  
5208 026126 003720  
5209 026130 014430  
5210 026132 004540  
5211 026134 015030  
5212 026136 007020  
5213 026140 015430  
5214 026142 011300  
5215 026144 016030  
5216 026146 016040  
5217 026150 016430  
5218 026152 022600  
5219 026154 017070  
5220 026156 177777 000000  
5221  
5222  
5223 026162 000000  
5224 026164 000000  
5225 026166 000000  
5226 026170 000000  
5227 026172 000000  
5228 026174 000000  
5229 026176 000000  
5230 026200 000156  
5231 026202 006307  
5232  
5233 026204 000100  
5234  
5235 026206 000001  
5236 026210 000240  
5237 026212 000000  
5238 026214 000000  
5239 026216 053200 041505 047524  
026240 041600 047117 051124  
026274 050200 051501 020123  
026310 044600 053116 046101  
026336 046200 047111 035105  
026346 041200 052501 020104

THE FOLLOWING IS A TABLE OF LEGAL BAUD RATES (8 BITS/CHAR)  
TABLE2: .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 12430 :ONE STOP BIT  
.WORD 600. :600 BAUD  
.WORD 13030 :ONE STOP BIT  
.WORD 1200. :1200 BAUD  
.WORD 13430 :ONE STOP BIT  
.WORD 1800. :1800 BAUD  
.WORD 14030 :ONE STOP BIT  
.WORD 2000. :2000 BAUD  
.WORD 14430 :ONE STOP BIT  
.WORD 2400. :2400 BAUD  
.WORD 15030 :ONE STOP BIT  
.WORD 3600. :3600 BAUD  
.WORD 15430 :ONE STOP BIT  
.WORD 4800. :4800 BAUD  
.WORD 16030 :ONE STOP BIT  
.WORD 7200. :7200 BAUD  
.WORD 16430 :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  
LINESP: 110. :DEFAULT BAUD RATE  
SPEED: 6307 :DEFAULT 110 BAUD, 8 BITS/CHAR,  
 :FDX, 2 STOP BITS  
NUMLIN: 100 :DEFAULT VALUE, REC. INTERRUPT ENABLED  
NUMTCR: 1 :DEFAULT VALUE, TCR BIT 0  
PRIO: 240 :DEFAULT DEVICE PRIORITY 5  
RECDAT: 0  
TBUF: 0  
MVECTO: .ASCII? <200>/VECTOR ADDRESS- /  
MREGAD: .ASCII? <200>/CONTROL REGISTER ADDRESS- /  
MPASS: .ASCII? <200>/PASS DONE./  
MINVAL: .ASCII? <200>/INVALID BAUD RATE - /  
MLINE: .ASCII? <200>/LINE: /  
MSPEED: .ASCII? <200>/BAUD RATE - /

```

026364 052200 050131 020105 MCHAR: .ASCIZ <200>/TYPE A CHAR. ON DZ11 TERMINAL /
026424 053600 044510 044103 WHICH: .ASCIZ <200>/WHICH TEST ? ECHO OR CABLE (E OR C) /
026472 052200 051105 044515 MTERM: .ASCIZ <200>/TERMINAL ECHO TEST /
026517 200 040503 046102 MCABLE: .ASCIZ <200>/CABLE TEST /
026534 006777 177777 177412 MQUICK: .ASCII <377><15><377><377><12><377><377>
026543 124 042510 050440 .ASCII /THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789/
026640 006777 177777 177412 .ASCII <377><15><377><377><12><377><377><377><0>
026652

```

```

.EVEN
:*****
:UTILITIES
:*****

```

5240  
5241  
5242  
5243  
5244  
5245  
5246  
5247  
5248  
5249  
5250  
5251  
5252  
5253  
5254  
5255  
5256  
5257  
5258  
5259  
5260  
5261  
5262  
5263  
5264  
5265  
5266  
5267  
5268  
5269  
5270  
5271  
5272  
5273  
5274  
5275  
5276  
5277  
5278  
5279  
5280  
5281  
5282  
5283  
5284  
5285

```

026652 006337 027060
026656 006337 027060
026662 006337 027060
026666 006337 027060
026672 006337 027060
026676 013737 027060 027062
026704 162737 000001 027062
026712 042737 000037 027062
026720 013700 002072
026724 062700 000002
026730 010037 002074
026734 062700 000002
026740 010037 002076
026744 062700 000002
026750 010037 002100
026754 013700 001310
026760 010037 002042
026764 005200
026766 010037 002044
026772 005200
026774 010037 002046
027000 010037 002052
027004 005200
027006 010037 002050
027012 010037 002054
027016 005200
027020 010037 002056
027024 005200
027026 010037 002060
027032 005200
027034 010037 002062
027040 010037 002066
027044 005200
027046 010037 002064
027052 010037 002070
027056 000207
027060 000240
027062 000200

```

```

: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 DZRV,RO ;PLACE THE BASE VECTOR ADDRESS IN RO
ADD #2,RO ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
MOV RO,DZRV ;STORE IT HERE
ADD #2,RO ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
MOV RO,DZTV ;STORE IT HERE
ADD #2,RO ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
MOV RO,DZTV ;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

```

			:ERROR ERROR TABLE	
		.ERRTAB:	0	:ERROR 0
5286				
5287	027064	000000	0	
5288	027066	000000	0	
5289	027070	000000	0	
5290				
5291	027072	027304	EM1	:ERROR
5292	027074	030554	DH1	
5293	027076	030752	DT1	
5294				
5295	027100	027357	EM2	:ERROR 2
5296	027102	030577	DH2	
5297	027104	030764	DT2	
5298				
5299	027106	027405	EM3	:ERROR 3
5300	027110	030632	DH3	
5301	027112	031002	DT3	
5302				
5303	027114	027444	EM4	:ERROR 4
5304	027116	030632	DH3	
5305	027120	031002	DT3	
5306				
5307	027122	027473	EM5	:ERROR 5
5308	027124	030644	DH4	
5309	027126	031010	DT4	
5310				
5311	027130	027522	EM6	:ERROR 6
5312	027132	030644	DH4	
5313	027134	031010	DT4	
5314				
5315	027136	027560	EM7	:ERROR 7
5316	027140	030632	DH3	
5317	027142	031002	DT3	
5318				
5319	027144	027621	EM8	:ERROR 10
5320	027146	030632	DH3	
5321	027150	031002	DT3	
5322				
5323	027152	027663	EM9	:ERROR 11
5324	027154	030632	DH3	
5325	027156	031002	DT3	
5326				
5327	027160	027721	EM10	:ERROR 12
5328	027162	030632	DH3	
5329	027164	031002	DT3	
5330				
5331	027166	027760	EM13	:ERROR 13
5332	027170	030632	DH3	
5333	027172	031002	DT3	
5334				
5335	027174	030011	EM14	:ERROR 14
5336	027176	030632	DH3	
5337	027200	031002	DT3	
5338				
5339	027202	030043	EM15	:ERROR 15
5340	027204	000000	0	
5341	027206	000000	0	

CZDZA-F0  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 110  
CZDZA DZ11 DEVICE DIAGNOSTICS.

SEQ 0109

5342				
5343	027210	030105	EM16	
5344	027212	030632	DH3	
5345	027214	031002	DT3	
5346				
5347	027216	030156	EM17	:ERROR 17
5348	027220	030632	DH3	
5349	027222	031002	DT3	
5350				
5351	027224	030214	EM20	
5352	027226	030632	DH3	
5353	027230	031002	DT3	
5354				
5355	027232	030255	EM21	:ERROR 21
5356	027234	030673	DH5	
5357	027236	031026	DT5	
5358				
5359	027240	030305	EM22	:ERROR 22
5360	027242	030644	DH4	
5361	027244	031010	DT4	
5362				
5363	027246	030347	EM23	:ERROR 23
5364	027250	030632	DH3	
5365	027252	031002	DT3	
5366				
5367	027254	030377	EM24	
5368	027256	030632	DH3	
5369	027260	031002	DT3	
5370				
5371	027262	030425	EM25	
5372	027264	030632	DH3	
5373	027266	031002	DT3	
5374				
5375	027270	030455	EM26	
5376	027272	030632	DH3	
5377	027274	031002	DT3	
5378				
5379	027276	030504	EM27	:
5380	027300	030632	DH3	
5381	027302	031002	DT3	

```

5382
5383 027304 047200 020117 046123 EM1: .ASCIZ <200>/NO SLAVE SYNC RESPONSE FROM DZ11 REGISTER/
027357 200 042522 044507 EM2: .ASCIZ <200>/REGISTER R/W FAILURE?
027405 200 051124 047101 EM3: .ASCIZ <200>/TRANSMIT READY (TRDY) NOT SET/
027444 051200 041505 044505 EM4: .ASCIZ <200>/RECEIVER DONE NOT SET/
027473 200 040504 040524 EM5: .ASCIZ <200>/DATA COMPARISON ERROR/
027522 042200 030532 020061 EM6: .ASCIZ <200>/DZ11 *RECEIVER BUFFER* ERROR/
027560 052200 040522 051516 EM7: .ASCIZ <200>/TRANSMITTER FAILED TO INTERRUPT/
027621 200 047125 054105 EM8: .ASCIZ <200>/UNEXPECTED TRANSMITTER INTERRUPT/
027663 200 042522 042503 EM9: .ASCIZ <200>/RECEIVER FAILED TO INTERRUPT/
027721 200 047125 054105 EM10: .ASCIZ <200>/UNEXPECTED RECEIVER INTERRUPT/
027760 051600 046111 020117 EM13: .ASCIZ <200>/SILO ALARM SET TOO SOON/
030011 200 044523 047514 EM14: .ASCIZ <200>/SILO ALARM FAILED TO SET/
030043 200 041501 044524 EM15: .ASCIZ <200>/ACTION DETECTED ON *INVALID LINE./
030105 200 042522 042101 EM16: .ASCIZ <200>/READING DZRBUF DID NOT CLEAR SILO ALARM/
030156 042200 052101 020101 EM17: .ASCIZ <200>/DATA VALID SHOULD NOT BE SET/
030214 051200 041505 044505 EM20: .ASCIZ <200>/RECEIVER DONE SHOULD NOT BE SET/
030255 200 042522 040514 EM21: .ASCIZ <200>/RELATIVE TIMING ERROR./
030305 200 047515 042504 EM22: .ASCIZ <200>/MODEM SIGNAL ERROR ON CABLE TEST/
030347 200 040504 040524 EM23: .ASCIZ <200>/DATA VALID IS NOT SET!/
030377 200 040504 040524 EM24: .ASCIZ <200>/DATA OVERRUN IS SET!/
030425 200 051106 046501 EM25: .ASCIZ <200>/FRAMING ERROR OCCURRED/
030455 200 040520 044522 EM26: .ASCIZ <200>/PARITY ERROR OCCURRED/
030504 043200 046125 020114 EM27: .ASCIZ <200>/FULL BINARY COUNT PATTERN NOT RECEIVED/

030554 052200 040522 020120 DH1: .ASCIZ <200>/TRAP PC DZ11 REG/
030577 200 054105 042520 DH2: .ASCIZ <200>/EXPECTED FOUND REGISTER/
030632 046200 047111 020105 DH3: .ASCIZ <200>/LINE NO./
030644 042600 050130 041505 DH4: .ASCIZ <200>/EXPECTED FOUND LINE/
030673 200 054124 046040 DH5: .ASCIZ <200>/TX LINE PREVIOUS TIME ACTUAL TIME PARAMETER/

```

.EVEN

```

030752 000002 DT1: .DATA TABLES FOR ERROR MESSAGES
030754 006 003 2
030756 001204 $REG1 6,3
030760 006 001 .BYTE 6,1
030762 001202 $REG0

030764 000003 DT2: 3
030766 006 004 .BYTE 6,4
030770 001214 $REG5
030772 006 001 .BYTE 6,1
030774 001212 $REG4
030776 006 001 .BYTE 6,1
031000 001202 $REG0

031002 000001 DT3: 1
031004 003 001 .BYTE 3,1
031006 001372 SAVLIN

031010 000003 DT4: 3
031012 006 004 .BYTE 6,4
031014 001214 $REG5
031016 006 001 .BYTE 6,1
031020 001212 $REG4

```

```

031022      003      001      .BYTE      3,1
031024      001372
031026      000004
031030      003      005      DT5:      4      .BYTE      3,5
031032      001372      SAVLIN
031034      006      011      .BYTE      6,9.
031036      001214      $REG5
031040      006      007      .BYTE      6,7
031042      001220      $TMP1
031044      006      001      .BYTE      6,1
031046      001400      REGIST

```

:TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES

```

031050      002450      DLYTBL: 2450      8      :TIME FOR 50 BAUD
031052      001560      1560      :TIME FOR 75 BAUD
031054      001120      1120      :TIME FOR 110 BAUD
031056      000750      750      1      :TIME FOR 134 BAUD
031060      000660      660      :TIME FOR 150 BAUD
031062      000330      330      :TIME FOR 300 BAUD
031064      000150      150      :TIME FOR 600 BAUD
031066      000060      60      :TIME FOR 1200 BAUD
031070      000040      40      :TIME FOR 1800 BAUD
031072      000030      30      :TIME FOR 2000 BAUD
031074      000020      20      :TIME FOR 2400 BAUD
031076      000010      10      :TIME FOR 3600 BAUD
031100      000001      1      :TIME FOR 4800 BAUD
031102      000001      1      :TIME FOR 7200 BAUD
031104      000001      1      :TIME FOR 9600 BAUD
031106      000001      1      :TIME OF DELAY FOR 19200 BAUD

```

:DELAYS WERE COMPUTED TO ALLOW MAXIMUM TIME AT EACH BAUD RATE  
:FOR ALL TESTS TO FUNCTION CORRECTLY ON A PDP11/45 WITH BIPOLAR  
:MEMORY. THE TIMES WERE ALSO TESTED ON AN 11/40 AND 11/10.

```

031110      002362      CORMAX:  . =2362
002362      000240      NOP
002364      000240      NOP
001512      001512      . =1512
001512      100000      100000
011572      011572      . =11572
011572      105720      TSTB      (R0)+
000001      000001      .END

```





































CZDZA-FO  
CZDZAF.P11

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 130

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0128

.DCLAS	006672	1498	2331#	
.DELAY	006704	1492	2335#	
.DEVIC	006652	1490	2323#	
.ERRTA	027064	2375	5287#	
.INSTE	006134	1476	2*63#	
.INSTR	006030	1474	2142#	
.INST1	006050	2146#	2166	
.MSG	006052	2144*	2147#	
.PARAM	006154	1478	2174#	
.PARMD	025516	1494	5088#	
.PAWCH	025712	1496	5147#	5159
.RES05	006414	1484	2247#	
.SAV05	006354	1482	2233#	
.SCOPE	004772	1100	1939#	2828
.SCOPI	005236	1470	1990#	
.SETFL	010500	1480	2542#	2561
.START	002150	1123	1556#	1569
.TRPSR	006630	1106	2312#	
.TRPTA	002002	1466#	2317	
.TYPE	005262	1472	1997#	
.SASTA-	***** U	2085	2088	
.SX	00*462	1324#	1329	







CZDZA-FO  
CZDZAF.P\*1

MACY11 30A(1052)  
26-NOV-79 11:03

26-NOV-79 11:06 PAGE 134  
CROSS REFERENCE TABLE -- MACRO NAMES

B 11

SEQ 0131

.SR2AZ	1#		
.\$SAVE	1#		
.\$SB2D	1#		
.\$SB2O	1#		
.\$SCOP	1#	779#	1926
.\$SIZE	1#		
.\$SUPR	1#		
.\$TRAP	1#	779#	
.\$TYPB	1#		
.\$TYPD	1#		
.\$TYPE	1#	779#	2001
.\$TYPC	1#		
.\$4OCA	1#		
.1170	1#		

. ABS. 031110 000

ERRORS DETECTED: 0

CZDZAF.BIN,CZDZAF.LST/CRF/SOL/NL:TOC=CZDZAF.SML,CZDZAF.P11  
RUN-TIME: 60 80 6 SECONDS  
RUN-TIME RATIO: 408/148=2.7  
CORE USED: 50K (99 PAGES)