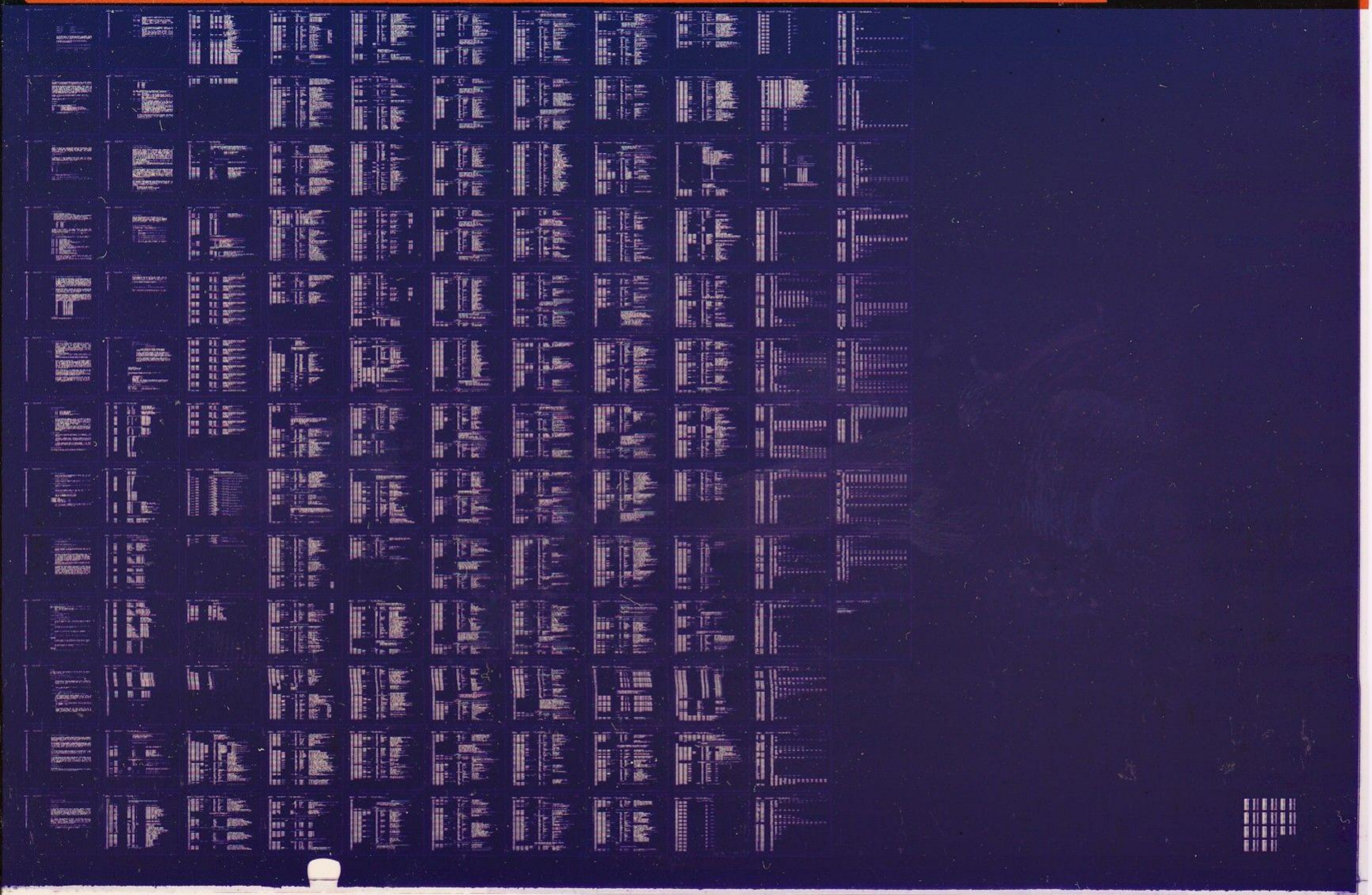
DZ11

DZ11 ASYNC MUX TEST
CZDZAHO

COPYRIGHT (c) 1976-84
AH-8783H-MC
FICHE 01 OF 01

FEB 1985
digital
Made In USA



```
                    .REM *

                                        IDENTIFICATION
                                        - -----------

                          PRODUCT CODE:            AC-8781H-MC

                          PRODUCT NAME:            CZDZAMO DZ11 LN ASYNC MUX TSTS

                          PRODUCT DATE:            JUNE 1984

                          MAINTAINER:              MK-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) 1976,1981,1984 BY DIGITAL EQUIPMENT CORPORATION


                          THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

                          DIGITAL         PDP             UNIBUS          MASSBUS

                          DEC             DECUS           DECTAPE
```

```
                                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 20MA 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 EQUILIVALENT FOR CONSOLE)
                                        DZ11            INTERFACE MODULE (M7819(EIA), M7814(20MA))
                                        M3271           STAGGERED TURNAROUND CONNECTOR FOR EIA MODULE.
                                        M3190           STAGGERED TURNAROUND CONNECTOR FOR 20MA MODULE.
                                        M325            CABLE TURNAROUND AND DIST PNL TESTING FOR EIA MODULE.
                                        M315            THIS MAY BE SUBSITUTED FOR M325.

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

```
                                2.2     STORAGE

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

                                3.      LOADING PROCEEDURE

                                3.1     METHOD

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

                                        ABSOLUTE LOADER STARTING ADDRESS =500

                                        MEMORY  = SIZE

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


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

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

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

```
127
128
129                          4.      STARTING PROCEEDURE
130
131                                  A.  SET SWITCH REGISTER TC 000200
132                                  B.  DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
133                                  C.  SET SWR TO ZERO FOR 'AUTO SIZING'  OR SET SW00=1 FOR  USER PARAMETER
134                                      INPUT FROM CONSOLE TERMINAL.   ON  FIRST  START IF SW07=1 AND SW00=0
135                                      THE PROGRAM WILL DEFAULT TO CONSOLE PARAMETER INPUT (SW00=1).
136                                  D.  DEPRESS 'START KEY' AND RELEASE. THE PROGRAM WILL TYPE MAINDEC  NAME
137                                      AND  PROGRAM  NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM OR
138                                      PARAMETERS WERE CHANGED BY SW00=1) AND ALSO THE FOLLOWING:
139
140                                          'MAP OF DZ11 STATUS'
141                                          1500    160100
142                                          1502    000300
143                                          1504    000005
144                                          1506    000377
145                                          1510    017070
146                                          1512    000000
147
148                                  THE ABOVE IS ONLY AN EXAMPLE!  THIS  WOULD  INDICATE  THE  STATUS  TABLE
149                                  STARTING  AT  ADD.  1500  IN  THE  PROGRAM.   THE  STATUS TABLE MUST BE
150                                  VERIFIED BY THE USER IF AUTO SIZING IS DONE.  FOR INFORMATION OF  STATUS
151                                  TABLE SEE SECTION 8.4 FOR HELP.
152                                  THE PROGRAM WILL TYPE "RUNNING" AND PROCEED TO RUN THE DIAGNOSTIC.
153
154                          4.1     CONTROL SWITCH SETTINGS
155
156                                  NOTE:   IF THERE IS NO REAL  SWR  (177570);   SWR  MAY  BE  MODIFIED  AT
157                                          LOC:176 OR BY HITTING CONTROL "G" <^G> ON CONSOLE TERMINAL.
158
159                                  SW 15   SET:  HALT ON ERROR
160                                  SW 14   SET:  LOOP ON CURRENT TEST
161                                  SW 13   SET:  INHIBIT ERROR PRINT OUT
162                                  SW 12   SET:  INHIBIT **ALL** TYPE OUT/BELL ON ERROR.
163                                  SW 11   SET:  INHIBIT ITERATIONS.  (QUICK PASS)
164                                  SW 10   SET:  ESCAPE TO NEXT TEST
165                                  SW 09   SET:  LOOP WITH CURRENT DATA
166                                  SW 08   SET:  CATCH ERROR AND LOOP ON IT
167                                  SW 07   SET:  NO AUTO SIZE.  IF 1ST START OF  PROGRAM AFTER  LOADING THE
168                                                OPERATOR MUST INPUT ADDRESS AND VECTOR FROM CONSOLE.
169                                  SW 06   SET:  RESELECT DZ11'S DESIRED ACTIVE
170                                  SW 05   SET:  RESERVED
171                                  SW 04   SET:  SELECT DELAY PARAMETER (SEE SEC. 4.1.1)
172                                  SW 03   SET:  EXTRA PARAMETER INPUT  (SEE SEC. 4.1.1)
173                                  SW 02   SET:  LOCK ON SELECTED TEST
174                                **SW 01   SET:  RESTART PROGRAM AT SELECTED TEST
175                                 *SW 00   SET:  GET USERS PARAMETERS FROM CONSOLE
176
177                                  *       FOR  ECHO OR  CABLE TESTS   (PROGRAM STARTED AT LOC. 210)   THIS
178                                          SWITCH  SET TO 1 ALLOWS  THE USER TO  TYPE IN THE VECTOR AND THE
17                                           CSR FOR THE DZ11 UNDER TEST.
:v                                   **      FOR ECHO OR CABLE TEST THIS SWITCH SET TO 1 ALLOWS THE SELECTION
181                                          OF EITHER THE ECHO OR CABLE TEST, BAUD RATE, AND THE LINE NUMBER
182                                          UNDER TEST.
```

```
183
184
185                          4.1.1   SWITCH REGISTER CONTROL OF PARAMETER INPUT FROM CONSOLE
186
187                                  SW 00   GET USERS PARAMETERS FROM CONSOLE.  SETTING THIS SWITCH AT START
188                                          UP  TIME  ALLOWS  THE USER TO INPUT AT THE CONSOLE TERMINAL  THE
189                                          FOLLOWING PARAMETERS:  BASE DEVICE ADDRESS, BASE VECTOR ADDRESS,
190                                          BUS REQUEST LEVEL, DECLARE EIA OR 20MA MODULE, MODE OF OPERATION
191                                          (EXTERNAL,INTERNAL, OR STAGGERED), AND THE NUMBER OF DZ11'S THAT
192                                          ARE RUNNING.   USING  THIS  SWITCH  ALONE DEFAULTS THE FOLLOWING
193                                          PARAMETERS:  ALL 8 LINES ARE SET TO BE TESTED ON EACH DZ11,  THE
194                                          DEFAULT BAUD RATE IS SET AT 9600 BAUD, AND THE CHARACTER LENGTH
195                                          FOR  THE MAJORITY  OF TESTING IS SET AT EIGHT BITS PER CHARACTER
196                                          WITH TWO STOP BITS.
197
198                                  SW 03   EXTRA PARAMETER INPUT   SETTING  THIS  SWITCH  AT  START UP TIME
199                                          PROVIDES  THE  USER WITH THE ABILITY TO SET THE LINES ACTIVE FOR
200                                          TESTING AND TO SET THE DEFAULT  BAUD RATE USED  FOR THE MAJORITY
201                                          OF THE DIAGNOSTIC  TESTS.  THE DELAY PARAMETER  IS AUTOMATICALLY
202                                          ADJUSTED TO THE BAUD RATE GIVEN BY THE USER.
203
204                                  SW 04   SELECT DELAY PARAMETER. THE DELAY PARAMETER THIS SWITCH CONTROLS
205                                          DETERMINES THE LENGTH OF TIME THE PROGRAM STALLS WAITING  FOR  A
206                                          CHARACTER  TO BE COMPLETELY TRANSMITTED OR RECEIVED.  THIS DELAY
207                                          COUNT IS AUTOMATICALLY SET TO PROVIDE ENOUGH DELAY TIME  FOR THE
208                                          DEFAULT BAUD RATE SPECIFIED WHEN RUNNING THE PROGRAM ON AN 11/45
209                                          WITH  BIPOLAR  MEMORY.  WHEN  RUNNING  THIS  PROGRAM ON A FASTER
210                                          PROCESSOR THE DELAY  PARAMETER SHOULD BE ADJUSTED PROPORTIONALLY
211                                          HIGHER THAN THE FOLLOWING DEFAULTED VALUES:
212                                          2450          ;TIME FOR   50 BAUD
213                                          1560          ;TIME FOR   75 BAUD
214                                          1120          ;TIME FOR  110 BAUD
215                                          0750          ;TIME FOR  134 BAUD
216                                          0660          ;TIME FOR  150 BAUD
217                                          0330          ;TIME FOR  300 BAUD
218                                          0150          ;TIME FOR  600 BAUD
219                                          0060          ;TIME FOR 1200 BAUD
220                                          0040          ;TIME FOR 1800 BAUD
221                                          0030          ;TIME FOR 2000 BAUD
222                                          0020          ;TIME FOR 2400 BAUD
223                                          0010          ;TIME FOR 3600 BAUD
224                                          0001          ;TIME FOR 4800 BAUD
225                                          0001          ;TIME FOR 7200 BAUD
226                                          0001          ;TIME FOR 9600 BAUD
227                                          0001          ;TIME FOR 19.2 KBAUD
228
229                                  *** NOTE ***
230                                  19.2K BAUD IS AN UNSUPPORTED BAUD RATE. IT SHOULD NOT NORMALLY BE USED.
231                                  9600 BAUD IS THE SPECIFIED MAXIMUM.
232
```

```
                                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.
                                        E:      NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05)
                                        F:      SET WITH ANY OTHER SWITCH SETTINGS DESIRED.  PRESS CONTINUE.


                                        SW 01   RESTART PROGRAM AT SELECTED TEST        IT IS STRONGLY SUGGESTED
                                                THAT  AT LEAST ONE PASS 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
                                        'SCOP1'  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)
```

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
326
327
328

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 'SCOP1'). 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.

```
                                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,F (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.
```

```
368
369
370                                     "# OF DZ11'S <IN OCTAL> (1:20):  "
371
372                                         TYPE TOTAL NUMBER OF DZ11'S TO BE TESTED IN THE  SYSTEM.   RANGE
373                                         IS 1 THRU 20 IN OCTAL.
374
375                                     ********* IF SW03=1 THEN *********
376                                         IF SW03=1 THE FOLLOWING WILL BE PRINTED.
377
378                                     "LINES ACTIVE BY BIT <IN OCTAL> (001:377):"
379
380                                         EACH BIT REPRESENTS A LINE AND ANY COMBINATION OF LINES  MAY  BE
381                                         SELECTED  (HOWEVER  IN STAGGERED MODE TWO ADJACENT LINES MUST BE
382                                         SELECTED (0-1, 2-3, 4-5, 6-7))..
383
384                                     "DEFAULT BAUD RATE <IN OCTAL> (00:16):  "
385
386                                         THIS GIVES THE USER A CHANCE TO CHANGE  THE  DEFAULT  BAUD  RATE
387                                         USED IN APP. 90 PERCENT OF THE TEST. BAUD RATE CHOICES ARE:
388                                         "00"(  50 BAUD),"01"(  75 BAUD),"02"( 110 BAUD),"03"( 134 BAUD),
389                                         "04"( 150 BAUD),"05"( 300 BAUD),"06"( 600 BAUD),"07"(1200 BAUD),
390                                         "10"(1800 BAUD),"11"(2000 BAUD),"12"(2400 BAUD),"13"(3600 BAUD),
391                                         "14"(4800 BAUD),"15"(7200 BAUD),"16"(9600 BAUD),"17"(19.2 KBAUD)
392                                         LOW DEFAULT BAUD RATES ARE NOT SUGGESTED SINCE THEY LENGTHEN THE
393                                         TIME TO COMPLETE A PROGRAM PASS DRAMATICALLY.
394                                         *** NOTE ***
395                                         SPEED SELECT CODE 17 CAN BE USED TO SELECT 19.2K BAUD, BUT THIS
396                                         SPEED IS NOT SPECIFIED BY DEC, AND SHOULD NOT NORMALLY BE USED.
397                                     ****************************************
398
399                                         IT IS IMPORTANT TO NOTE THAT ALL DZ11'S IN THE  SYSTEM  MUST  BE
400                                         CONTIGIOUS  FOR  BOTH  ADDRESS  AND VECTORS.  ALSO ALL THE EXTRA
401                                         PARAMETERS OTHER THAN CSR AND VECTORS ARE GIVEN TO THE  EXISTING
402                                         DZ11'S IN THE SYSTEM.  IF NOT ALL DZ11'S ARE SAME PRIORITY OR IF
403                                         THE MODE OF OPERATION IS DIFFERENT FOR EACH DZ11;  THIS MUST  BE
404                                         "PATCHED"  INTO THE CORRECT STATUS MAP ENTRY WHICH IS PRINTED AT
405                                         START TIME.  AN ALTERNATIVE IS TO  PUT  SW00=1  AT  START  TIME;
406                                         ANSWER  QUESTIONS ABOUT DZ11 UNDER TEST AND INDICATE ONLY 1 DZ11
407                                         IN THE SYSTEM.  IF THE STATUS MAP IS TO BE "PATCHED" IT MUST  BE
408                                         DONE AFTER THE QUESTIONS ARE ANSWERED OR AFTER THE AUTO SIZE.
```

```
409
410
411                                5.2     HOW TO RUN THE "CABLE/ECHO" TESTS.
412
413                                        NORMAL STARTING FOR THE FIRST TIME WOULD BE:  LOAD ADDRESS  210;   START
414                                        WITH THE SWR EQUAL TO 003.
415                                        NOTE:   SW00=1  ASKS FOR "VECTOR" AND "CSR"
416                                                SW01=1  ASKS FOR "WHICH TEST ECHO OR CABLE", "BAUD RATE", "LINE"
417                                                UNDER TEST.  PROGRAM WILL PRINT OUT:
418
419                                        "VECTOR ADDRESS-"
420
421                                                YOU TYPE VECTOR WITH A <CR>.
422
423                                        "CONTROL REGISTER ADDRESS-"
424
425                                                YOU TYPE IN DZCSR UNDER TEST.
426
427                                        "WHICH TEST ?  ECHO OR CABLE (E OR C)"
428
429                                                LETS DO THE CABLE TEST FIRST.  **THIS TEST IS ONLY TO BE DONE ON
430                                                THE  EIA  VERSION  OF  THE DZ11 NOT THE 20MA VERSION".  TYPE "C"
431                                                <CR>
432
433                                        "BAUD RATE- "
434
435                                                TYPE EITHER 50, 110, 135, 150, 300, 600, 1200 1800, 2000,  2400,
436                                                3600, 4800, 7200, 9600 FOLLOWED BY <CR>
437
438                                        "LINE:  "
439
440                                                YOU TYPE THE LINE WHICH HAS  THE  H325  TEST  CONNECTOR.   (TYPE
441                                                EITHER 0, 1, 2, 3, 4, 5, 6, 7) PROGRAM WILL THEN PRINT:
442
443                                        "CABLE TEST"
444
445                                                AND IF EVERYTHING IS WORKING;  THE FOLLOWING WILL BE PRINTED:
446
447                                        "PASS DONE."
448                                        "PASS DONE."
449                                          ETC.
450                                                TO CHANGE LINES;  HIT ANY PRINTING KEY ON YOUR CONSOLE  TERMINAL
451                                                WHILE THE PROGRAM IS RUNNING AND THE FOLLOWING WILL BE PRINTED:
452
453                                        "LINE:  "
454
455                                                NOW CHANGE THE H325 TEST CONNECTOR TO ANOTHER LINE AND TYPE  THE
456                                                NEW LINE.  PROGRAM WILL THEN PRINT:
457
458                                        "CABLE TEST"
459                                        "PASS DONE."
460                                        "PASS DONE."
461                                                CONTINUE THIS OPERATION UNTIL ALL LINES ARE TESTED.
```

```
452
463
464                                  5.3     ECHO TEST
465
466                                          IF PROGRAM HAS ALREADY BEEN STARTED AT 210 AND THE  VECTOR  AND  ADDRESS
467                                          HAVE  BEEN  TYPED IN;  JUST LOAD ADDRE S 210 AND START WITH SWR EQUAL TO
468                                          002.  PROGRAM WILL PRINT:
469
470                                          "WHICH TEST ?  ECHO OR CABLE (E OR C)"
471
472                                                  NOW TYPE AN "E" TO DO THE ECHO TEST.  PROGRAM WILL PRINT:
473
474                                          "BAUD RATE-"
475
476                                                  TYPE BAUD RATE AT WHICH THE TERMINAL IS SET THAT IS CONNECTED TO
477                                                  THE DZ11 DIST PNL.  BAUD RATE CHOICES ARE: 50, 75, 110, 135, 150
478                                                  300, 600 , 1200,  1800,  2000, 2400, 3600, 4800, 7200, 9600. THE
479                                                  PROGRAM WILL THEN PRINT:
480
481                                          LINE:  "
482
483                                                  TYPE TH  LINE THE TERMINAL IS CONNECTED TO AT THE DIST PNL  THEN
484                                                  THE PROGRAM WILL PRINT:
485
486                                          "TERMINAL ECHO TEST"
487
488                                                  * * AT THIS POINT THE MESSAGE:
489
490                                          "THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789"
491
492                                                  SHOULD BE PRINTED ON THE TERMINAL CONNECTED  TO  THE  DZ11.   IF
493                                                  THIS  MESSAGE IS DESIRED TO BE CONTINIOUSLY OUTPUT;  SET THE SWR
494                                                  TO 377 (SWR=377) WHILE IT IS BEING OUTPUT  OR  WHEN THE LINE NO.
495                                                  IS REQUESTED ABOVE. WHEN THIS MESSAGE IS DONE AND THE SWR IS NOT
496                                                  EQUAL TO 377;  THE CONSOLE WILL PRINT:
497
498                                          "TYPE A CHAR.  ON DZ11 TERMINAL"
499
500                                                  ANY PRINTABLE CHAR HIT ON DZ11 TERMINAL SHOULD BE ECHOED BACK ON
501                                                  THE  TERMINAL.   **IF  YOU HIT CNTRL C <↑C> ON THE DZ11 TERMINAL
502                                                  THE PROGRAM WILL PRINT:
503
504                                          "PASS DONE."
505
506                                                  ON THE CONSOLE TERMINAL  AND  THE  "QUICK  BROWN  FOX"  WILL  BE
507                                                  PRINTED  ON  DZ11  TERMINAL  AGAIN  AND  THE  ECHO  TEST WILL BE
508                                                  RUNNING.  TO CHANGE LINES:  TYPE ANY PRINTABLE CHARACTER  ON THE
509                                                  CONSOLE TERMINAL (NOT THE DZ11 TERMINAL). THE PROGRAM WILL AGAIN
510                                                  TYPE "LINE:  " AND WAIT FOR A RESPONSE.
511
```

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 '$TSTNM' (BYTE 1122) 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 COMPLETLY ISOLATE PROBLEMS.

```
                        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-H  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.
```

```
588
589
590                                   8.4     KEY LOCATIONS
591
592                                           $LPADR (1126)   CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN
593                                                           ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS
594                                                           ASSERTED.
595                                           NEXT   (1360)   CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PEFORMED.
596                                           $TSTNM (1122)   CONTAINS THE NUMBER OF THE TEST NOW BEING PEFORMED.
597                                           RUN    (1406)   THE BIT IN 'RUN' ALWAYS POINTS ONE PAST THE DZ11
598                                                           CURRENTLY    BEING    TESTED.    EXAMPLE:    (RUN)
599                                                           1304/0000000001000000 MEANS THAT DZ11 NO.05 IS THE  DZ11
600                                                           NOW RUNNING.
601                                           STATUS MAP
602                                           (1500)-(2000)
603                                                           THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST
604                                                           UP TO 16 (DECIMAL) DZ11S SEQUENTIALY.  THEY CONTAIN THE
605                                                           CSR,VECTOR AND STATUS CONCERNING THE CONFIGURATION OF
606                                                           EACH DZ11.
607                                           DZACTV (1404)   EACH BIT SET IN THIS LOCATION INDICATES THAT THE
608                                                           ASSOCIATED DZ11 WILL BE TESTED IN TURN.  EXAMPLE:
609                                                           (DZACTV) 1300/0000000000011111 MEANS THAT DZ11 NO.
610                                                           00,01,02,03,04 WILL BE TESTED.  EXAMPLE:  (DZACTV)
611                                                           1300/0000000000010001 MEANS THAT DZ11 NO.  00,04 WILL BE
612                                                           TESTED.
613                                           $BASE  (1310)   CONTAINS THE RECEIVER CSR OF THE CURRENT DZ11 UNDER
614                                                           TEST.
615
```

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

'MAP OF DZ11 STATUS'

1500	160100
1502	000300
1504	000005
1506	000377
1510	017070
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	017070	THIS IS THE PARAMETER LOCATION USED IN MOST OF THE TESTS. IT INDICATES PARAMETERS OF: RX ON, SPEED SELECT 16 (9600 BAUD) EIGHT BITS PER CHAR, AND TWO STOP BITS. THE USER MAY ALTER THE STOP BITS AND THE SPEED, BUT THE REMAINING PARAMETERS SHOULD BE LEFT ALONE.
1512	000000	THIS LOCATION 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. THIS LOCATION WILL CONTAIN EITHER ALL ZEROS INDICATING THAT INTERNAL LOOP WAS SELECTED AS MODE OF OPERATION OR IT WILL CONTAIN 100000 INDICATING THAT "STAGGERED MODE" WAS SELECTED OR IT WILL CONTAIN 000200 INDICATING THAT "EXTERNAL" WAS THE MODE SELECTED.

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

8.5 ••• METHOD OF AUTO SIZING •••

8.5.1 FINDING THE CONTROL STATUS REGISTER.

THE PROGRAM WILL START AT ADDRESS 160300 AND START 'REFERENCING' THE ADDRESS IN THE POINTER. IF A NON-EX MEMORY TRAP OCCURES, THE POINTER (HOLDING 160000) IS UPDATED BY 10 AND THE ABOVE IS REPEATED UNTIL ADDRESS 163700 IS REACHED. IF A 'SLAVE SYNC RESPONSE' WAS ISSUED BY THE DZ11 (OR ANY OTHER DEVICE) (NO NXM TRAP). "MASTER SCAN ENABLE" IS ATTEMPTED TO BE SET AND THE "TCR" BIT FOR LINE 7 IS SET. "TRDY" IS THEN TESTED TO BE SET AND BOTH "TCR07" AND "MASTER SCAN ENABLE" ARE TESTED TO BE STILL SET. IF ALL OF THIS WORKED; THEN A "DEVICE CLEAR" IS ISSUED TESTING THAT THE BIT CAN BE READ BACK AND THAT AFTER SOME TIME IT SELF CLEARS. IF ALL OF THE ABOVE WORKED; THIS DEVICE IS ASSUMED TO BE A DZ11. IF ANY OF THE ABOVE FAILED; UPDATING OF THE POINTER IS DONE AND THE SEQUENCE IS REPEATED.
NOTE: IF THE PROGRAM DOES NOT FIND YOUR DZ11; SOMETHING IS WRONG AND AUTO SIZING SHOULD NOT BE DONE.
AFTER IDENTIFYING A DZ11 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 INTERUPT ENABLE AND MSTSCAN ENABLE) ARE SET INTO THE DZCSR. "TCR07" IS THEN SET. A DELAY IS MADE AND IF NO INTERUPT OCCURES (BECAUSE OF A BAD DZ11) THE PROGRAM ASSUMES VECTOR ADDRESS 300 AND THE PROBLEM SHOULD BE FIXED IN THE DIAGNOSTIC. ONCE THE PROBLEM IS FIXED; THE PROGRAM SHOULD BE RE-SETUP AGAIN TO GET CORRECT VECTOR. IF AN INTERUPT OCCURRED; THE ADDRESS TO WHICH THE DZ11 INTERUPTED 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 16 (9600 BAUD).
4) MODE OF OPERATION IS "INTERNAL MODE".

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

9.0 RUNNING THE DZ11 DIAGNOSTIC UNDER APT

9.1.1 THE APT INTERFACE

CZDZA HAS BEEN REDESIGNED TO BE COMPATIBLE WITH THE APT-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
                        TABLE.  THE USER IS THEN FREE TO MONITOR UNDER APT AS NORMAL.
```

```
                                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.

                                        G0:     MAY 1981 - INCORPORATE XON/XOFF FEATURES (BY ASSEMBLING WITH
                                                LATEST SYSMAC - C5) AND CHANGE ".=" CODE MODIFIERS AT END
                                                OF PROGRAM FROM ABSOLUTE ADDRESS MODE TO RELATIVE. ALSO, THE
                                                DEFAULT BAUD RATE WAS CHANGED TO 9600 BAUD, SINCE 19.2K BAUD IS
                                                NOT SUPPORTED, AND TEST 34 WAS CHANGED TO REMOVE 19.2K TESTING.

                                        H0:     JUNE 1984 - ADDED BAUD RATE TIMING TEST.  ADDED IN ORDER
                                                TO TEST CRYSTAL SPEEDS.  TRANSMITTS CHARACTERS TO ALL 8 LINES
                                                AT ALL BAUD RATES (EXCEPT 19200 BAUD) FOR 1 SECOND AND IF THE
                                                NUMBER OF CHARACTERS TRANSMITTED IS WITHIN A RANGE, THE TEST
                                                WILL PASS.

                                                                        ; -PRGFRT-------------------------------
                                .TITLE  CZDZA-H0
                                ;*COPYRIGHT (C) 1976,1984
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
                                ;*
000001                          $TN=1
                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;LOAD ADDRESS 000200
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE "CZDZA-H0/<200>/CZDZAH0 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 ***
001120                          STACK=  1120
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
```

```
                        
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
                        
                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER
                        
                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7
                        
                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
```

```
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
        100000                  BIT15=  100000
        040000                  BIT14=  40000
        020000                  BIT13=  20000
        010000                  BIT12=  10000
        004000                  BIT11=  4000
        002000                  BIT10=  2000
        001000                  BIT09=  1000
        000400                  BIT08=  400
        000200                  BIT07=  200
        000100                  BIT06=  100
        000040                  BIT05=  40
        000020                  BIT04=  20
        000010                  BIT03=  10
        000004                  BIT02=  4
        000002                  BIT01=  2
        000001                  BIT00=  1
                                .EQUIV  BIT09,BIT9
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

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


                                ;INSTRUCTION DEFINITIONS
                                ;----------------------

        005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
        005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
        010046                  PUSHR0=10046    ;SAVE R0 ON STACK
        012600                  POPR0=12600     ;RESTORE R0 FROM STACK
        024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE
```

```
000626                  POP2SP=22626    ;INCREMENT STACK TWICE

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

000010                  MAINT = BIT3    ;MAINTENANCE MODE ENABLE
000020                  DCLR=BIT4       ;DEVICE CLEAR
000040                  MSENAB=BIT5     ;MASTER SCAN ENABLE
000100                  RIE=BIT6        ;RECEIVER INTERRUPT ENABLE
000200                  RDONE=BIT7      ;RECEIVER DONE
010000                  SILOEN= BIT12   ;SILO ALARM ENABLE
020000                  SILOAL = BIT13  ;SILO ALARM
040000                  TIE=BIT14       ;TRANSMITTER INTERRUPT ENABLE
100000                  TRDY=BIT15      ;TRANSMITTER READY

000021                  $XON=21
000023                  $XOFF=23

                        ;DZCSR WORD DEFINITIONS
                        ;----------------------
000000                  TL0=0           ;TRANSMIT LINE 0
000400                  TL1=BIT8        ;TRANSMIT LINE 1
001000                  TL2=BIT9        ;TRANSMIT LINE 2
001400                  TL3=BIT9!BIT8   ;TRANSMIT LINE 3
002000                  TL4=BIT10       ;TRANSMIT LINE 4
002400                  TL5=BIT10!BIT8  ;TRANSMIT LINE 5
003000                  TL6=BIT10!BIT9  ;TRANSMIT LINE 6
003400                  TL7=BIT10!BIT9!BIT8 ;TRANSMIT LINE 7


                        ;DZRBUF BIT DEFINITIONS
                        ;----------------------

010000                  PARER=BIT12     ;PARITY ERROR
020000                  FRMERR=BIT13    ;FRAME ERROR
040000                  OVRRUN=BIT14    ;OVERRUN ERROR
100000                  DVALID=BIT15    ;DATA VALID

                        ;DZRBUF WORD DEFINITIONS
                        ;----------------------

000000                  RL0=0           ;RECEIVER LINE 0
000400                  RL1=BIT8        ;RECEIVER LINE 1
001000                  RL2=BIT9        ;RECEIVER LINE 2
001400                  RL3=BIT9!BIT8   ;RECEIVER LINE 3
002000                  RL4=BIT10       ;RECEIVER LINE 4
002400                  RL5=BIT10!BIT8  ;RECEIVER LINE 5
003000                  RL6=BIT10!BIT9  ;RECEIVER LINE 6
003400                  RL7=BIT10!BIT9!BIT8 ;RECEIVER LINE 7

                        ;DZLPR WORD DEFINITIONS
                        ;----------------------

000000                  LP0=0           ;LINE PARAMETER 0
000001                  LP1=BIT0        ;LINE PARAMETER 1
```

```
                        LP2=BIT1        ;LINE PARAMETER 2
                        LP3=BIT1!BIT0   ;LINE PARAMETER 3
                        LP4=BIT2        ;LINE PARAMETER 4
                        LP5=BIT2!BIT0   ;LINE PARAMETER 5
                        LP6=BIT2!BIT1   ;LINE PARAMETER 6
                        LP7=BIT2!BIT1!BIT0      ;LINE PARAMETER 7

                        FIVE=0          ;FIVE BITS/CHAR,1 STOP BIT
                        SIX=BIT3        ;SIX BITS/CHAR,1 STOP BIT
                        SEVEN=BIT4      ;SEVEN BITS/CHAR,1 STOP BIT
                        EIGHT=BIT4!BIT3 ;EIGHT BITS/CHAR,1 STOP BIT
                        FIVES=BIT5      ;FIVE BITS/CHAR,2 STOP BITS
                        SIXS=BIT5!BIT3  ;SIX BITS/CHAR,2 STOP BITS
                        SEVENS=BIT5!BIT4        ;SEVEN BITS/CHAR, 2 STOP BITS
                        EIGHTS=BIT5!BIT4!BIT3   ;EIGHT BITS/CHAR, 2 STOP BITS

                        PARITY=BIT6             ;PARITY ENABLED
                        ODDPAR=BIT7             ;ODD PARITY ENABLED
                        ONESTOP=0               ;ONE STOP BIT ENABLED
                        TWOSTOP=BIT5            ;TWO STOP BITS ENABLED
                        EVEPAR=0                ;EVEN PARITY ENABLED
                        RCVON=BIT12             ;ENABLE RECEIVER (RECEIVER ON)

                        S50=0                   ;SPEED 50 BAUD
                        S75=BIT8                ;SPEED 75 BAUD
                        S110=BIT9               ;SPEED 110 BAUD
                        S134=BIT9!BIT8          ;SPEED 134.5 BAUD
                        S150=BIT10              ;SPEED 150 BAUD
                        S300=BIT10!BIT8         ;SPEED 300 BAUD
                        S600=BIT10!BIT9         ;SPEED 600 BAUD
                        S1200=BIT10!BIT9!BIT8   ;SPEED 1200 BAUD
                        S1800=BIT11             ;SPEED 1800 BAUD
                        S2000=BIT11!BIT8        ;SPEED 2000 BAUD
                        S2400=BIT11!BIT9        ;SPEED 2400 BAUD
                        S3600=BIT11!BIT9!BIT8   ;SPEED 3600 BAUD
                        S4800=BIT11!BIT10       ;SPEED 4800 BAUD
                        S7200=BIT11!BIT10!BIT8  ;SPEED 7200 BAUD
                        S9600=BIT11!BIT10!BIT9  ;SPEED 9600 BAUD
                        S19200=BIT11!BIT10!BIT9!BIT8    ;SPEED 19200 BAUD

                                ;DZTCR BIT DEFINITIONS
                                ;---------------------
                        TCR0=BIT0               ;TCR0
                        TCR1=BIT1               ;TCR1
                        TCR2=BIT2               ;TCR2
                        TCR3=BIT3               ;TCR3
                        TCR4=BIT4               ;TCR4
                        TCR5=BIT5               ;TCR5
                        TCR6=BIT6               ;TCR6
                        TCR7=BIT7               ;TCR7
                        DTR0=BIT8               ;DTR0
                        DTR1=BIT9               ;DTR1
                        DTR2=BIT10              ;DTR2
                        DTR3=BIT11              ;DTR3
                        DTR4=BIT12              ;DTR4
                        DTR5=BIT13              ;DTR5
```

```
1050            040000                  DTR6=BIT14              ;DTR6
1051            100000                  DTR7=BIT15              ;DTR7
1052
1053                                    ;DZMSR BIT DEFINITIONS
1054                                    ;---------------------
1055            000001                  RING0=BIT0              ;RING INDICATED ON LINE 0
1056            000002                  RING1=BIT1              ;RING INDICATED ON LINE 1
1057            000004                  RING2=BIT2              ;RING INDICATED ON LINE 2
1058            000010                  RING3=BIT3              ;RING INDICATED ON LINE 3
1059            000020                  RING4=BIT4              ;RING INDICATED ON LINE 4
1060            000040                  RING5=BIT5              ;RING INDICATED ON LINE 5
1061            000100                  RING6=BIT6              ;RING INDICATED ON LINE 6
1062            000200                  RING7=BIT7              ;RING INDICATED ON LINE 7
1063            000400                  CO0=BIT8                ;CARRIER PRESENT ON LINE 0
1064            001000                  CO1=BIT9                ;CARRIER PRESENT ON LINE 1
1065            002000                  CO2=BIT10               ;CARRIER PRESENT ON LINE 2
1066            004000                  CO3=BIT11               ;CARRIER PRESENT ON LINE 3
1067            010000                  CO4=BIT12               ;CARRIER PRESENT ON LINE 4
1068            020000                  CO5=BIT13               ;CARRIER PRESENT ON LINE 5
1069            040000                  CO6=BIT14               ;CARRIER PRESENT ON LINE 6
1070            100000                  CO7=BIT15               ;CARRIER PRESENT ON LINE 7
1071
1072                                    ;DZTDR BIT DEFINITIONS
1073                                    ;---------------------
1074
1075            000400                  BRK0=BIT8               ;BREAK FOR LINE 0
1076            001000                  BRK1=BIT9               ;BREAK FOR LINE 1
1077            002000                  BRK2=BIT10              ;BREAK FOR LINE 2
1078            004000                  BRK3=BIT11              ;BREAK FOR LINE 3
1079            010000                  BRK4=BIT12              ;BREAK FOR LINE 4
1080            020000                  BRK5=BIT13              ;BREAK FOR LINE 5
1081            040000                  BRK6=BIT14              ;BREAK FOR LINE 6
1082            100000                  BRK7=BIT15              ;BREAK FOR LINE 7
1083
1084
1085                                    ;TABLE OF LOOP AROUND FUNCTIONS (H325)
1086                                    ;
1087                                    ;       ----------------
1088                                    ;       I               ↑
1089                                    ;       V               ↑
1090                                    ;       REC             TRANS
1091                                    ;       DATA            DATA
1092                                    ;
1093                                    ;       ----------------
1094                                    ;       I               ↑
1095                                    ;       V               ↑
1096                                    ;       CO              RTS
1097                                    ;
1098                                    ;
1099                                    ;       ----------------
1100                                    ;       I               ↑
1101                                    ;       V               ↑
1102                                    ;       RING            DTR
```

```
1103                                    ;;**********************************************************************
1104                                    ;----------------------------------------------------------------------
1105                                            ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
1106                                            ;THE STANDARD "TRAP CATCHER" IS PLACED
1107                                            ;BETWEEN ADDRESS 0 TO ADDRESS 776.
1108                                            ;IT LOOKS LIKE "PC+2 HALT".
1109                                    ;----------------------------------------------------------------------
1110                                    ;;**********************************************************************
1111
1112             000000                 .=0
1113                                            ;STANDARD INTERRUPT VECTORS
1114                                            ;-------------------------
1115
1116             000010                 .=10
1117    000010   011440                        SET.PS                  ;FAKE "MTPS" INSTRUCTION TRAP
1118    000012   000340                        PR7                     ;MAKE SURE PS IS PRIORITY 7
1119
1120             000020                 .=20
1121    000020   005122                        .SCOPE                  ;SCOPE LOOP HANDLER
1122    000022   000340                        PR7                     ;HANDLE AT PRIORITY 7
1123    000024   010320                        $PWRDN                  ;POWER FAIL HANDLER
1124    000026   000340                        340                     ;SERVICE AT PRIORITY LEVEL 7
1125    000030   007230                        $ERROR                  ;ERROR HANDLER
1126    000032   000340                        340                     ;SERVICE AT PRIORITY LEVEL 7
1127    000034   007122                        .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
1128    000036   000340                        340                     ;SERVICE AT PRIORITY LEVEL 7
1129                                    .SBTTL  ACT11 HOOKS
1130
1131                                    ;;**********************************************************************
1132                                    ;HOOKS REQUIRED BY ACT11
1133             000040                         $SVPC=.                 ;SAVE PC
1134             000046                         .=46
1135    000046   005056                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1136             000052                         .=52
1137    000052   000000                        .WORD   0               ;;2)SET LOC.52 TO ZERO
1138             000040                         .=$SVPC                 ;; RESTORE PC
1139
1140             000174                 .=174
1141    000174   000000                DISPREG:0                       ;SOFTWARE DISPLAY REGISTER FOR SWITCHLESS 11S
1142    000176   000000                SWREG:  0                       ;SOFTARE SWITCH REGISTER FOR SWITCHLESS 11S
1143             000200                 .=200
1144    000200   000137   002150                JMP     .START          ;GO TO START OF PROGRAM
1145             000210                 .=210
1146    000210   000137   025216                JMP     XSTART          ;GOTO CABLE TEST/ECHO TEST
1147
1148
1149             001000                 .=1000
1150    001000   005200   055103  055104 MTITLE: .ASCIZ  <200><12>/CZDZA-HO/<200>/CZDZAHO DZ11 LN ASYNC MUX TSTS /<200>
 (2)
```

```
                                .SBTTL  COMMON TAGS

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

                        001120          .=1120
001120                  $CMTAG:                         ;;START OF COMMON TAGS
001120  000000                  .WORD   0
001122  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001123  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001124  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001126  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001130  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001132  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001134  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001135  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001136  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001140  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001142  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001144  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001146  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001150  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001152  000000                  .WORD   0
001154  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001155  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001156  000000                  .WORD   0
001160  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001162  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001164  177560          $TKS:   177560                  ;;TTY KBD STATUS
001166  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001170  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001172  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001174  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001175  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001176  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001177  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001200  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001202  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001204  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001206  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001210  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001212  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001214  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001216  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001220  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001222  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001224  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001226  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001230  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001231  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001232  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE
```

```
1207                                    ;;*****************************************************************
1208                                    .EVEN
1209  001234                            $MAIL:                          ;;APT MAILBOX
1210  001234  000000                    $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
1211  001236  000000                    $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
1212  001240  000000                    $TESTN: .WORD   ATESTN          ;;TEST NUMBER
1213  001242  000000                    $PASS:  .WORD   APASS           ;;PASS COUNT
1214  001244  000000                    $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
1215  001246  000000                    $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
1216  001250  000000                    $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
1217  001252  000000                    $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
1218  001254                            $ETABLE:                        ;;APT ENVIRONMENT TABLE
1219  001254     000                    $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
1220  001255     000                    $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
1221  001256  000000                    $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
1222  001260  000000                    $USWR:  .WORD   A.SWR           ;;USER SWITCHES
1223  001262  000000                    $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
1224                                    ;*                              BITS 15-12:=CPU TYPE
1225                                    ;*                                      11/04=01,11/05=02,11/20=0   11/40=04,11/45=05
1226                                    ;*                                      11/70=06,PDQ=07,0=10
1227                                    ;*                              BIT 10=REAL TIME CLOCK
1228                                    ;*                              BIT  9=FLOATING POINT PROCESSOR
1229                                    ;*                              BIT  8=MEMORY MANAGEMENT
1230  001264     000                    $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
1231  001265     000                    $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
1232                                    ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
1233                                    ;*                                      900 NSEC CORE=001
1234                                    ;*                                      300 NSEC BIPOLAR=002
1235                                    ;*                                      500 NSEC MOS=003
1236  001266  000000                    $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
1237                                    ;*                                MEM.LAST ADDR.-3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1238  001270     000                    $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
1239  001271     000                    $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
1240  001272  000000                    $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
1241  001274     000                    $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
1242  001275     000                    $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
1243  001276  000000                    $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
1244  001300     000                    $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
1245  001301     000                    $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
1246  001302  000000                    $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
1247  001304  000000                    $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1248  001306  000000                    $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1249  001310  160010                    $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1250  001312  000000                    $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
1251  001314  000000                    $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
1252  001316  000000                    $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
1253  001320  000000                    $DDW0:  .WORD   ADDW0           ;;DEVICE DESCRIPTOR WORD#0
1254  001322  000000                    $DDW1:  .WORD   ADDW1           ;;DEVICE DESCRIPTOR WORD#1
1255  001324  000000                    $DDW2:  .WORD   ADDW2           ;;DEVICE DESCRIPTOR WORD#2
1256  001326  000000                    $DDW3:  .WORD   ADDW3           ;;DEVICE DESCRIPTOR WORD#3
1257  001330  000000                    $DDW4:  .WORD   ADDW4           ;;DEVICE DESCRIPTOR WORD#4
1258  001332  000000                    $DDW5:  .WORD   ADDW5           ;;DEVICE DESCRIPTOR WORD#5
1259  001334  000000                    $DDW6:  .WORD   ADDW6           ;;DEVICE DESCRIPTOR WORD#6
1260  001336  000000                    $DDW7:  .WORD   ADDW7           ;;DEVICE DESCRIPTOR WORD#7
1261  001340  000000                    $DDW8:  .WORD   ADDW8           ;;DEVICE DESCRIPTOR WORD#8
1262  001342  000000                    $DDW9:  .WORD   ADDW9           ;;DEVICE DESCRIPTOR WORD#9
```

```
 1263  001344  000000                    $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
 1264  001346  000000                    $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
 1265  001350  000000                    $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
 1266  001352  000000                    $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
 1267  001354  000000                    $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
 1268  001356  000000                    $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
 1269
 1270
 1271  001360                            $ETEND:
 1272
```

```
1273                                    .SBTTL  ERROR POINTER TABLE
1274
1275                                    ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1276                                    ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1277                                    ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1278                                    ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1279                                    ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1280
1281                                    ;*      EM              ;;POINTS TO THE ERROR MESSAGE
1282                                    ;*      DH              ;;POINTS TO THE DATA HEADER
1283                                    ;*      DT              ;;POINTS TO THE DATA
1284                                    ;*      DF              ;;POINTS TO THE DATA FORMAT
1285
1286
1287  001360                            $ERRTB:
1288
1289                                            ;PROGRAM CONTROL PARAMETERS
1290                                            ;-------------------------
1291
1292  001360  000000                    NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
1293  001362  000000                    LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
1294
1295                                            ;PROGRAM VARIABLES
1296                                            ;----------------
1297
1298  001364  000377                    LINE:   377                     ;DEFAULT ALL EIGHT LINES RUNNING
1299  001366  017070                    PAR:    17070                   ;PARAMETERS: 8 BITS/CHAR,2 STOP BITS.  9600 BAUD,NO PARIT
1300  001370  000000                    MODE:   0                       ;DEFAULT MAINTENANCE MODE
1301  001372  000000                    SAVLIN: 0                       ;LINE NUMBER
1302  001374  000000                    XMTLIN: 0                       ;TRANSMISSION LINE NUMBER
1303  001376  000000                    XMTCNT: 0                       ;COUNT OF WORDS IN A TRANSMISSION PATTERN
1304  001400  000000                    REGIST: 0                       ;DEVICE ADDRESS STORAGE LOCATION
1305  001402  000000                    SAVPC:  0                       ;PROGRAM COUNTER STORAGE
1306  001404  000001                    DZACTV: .BLKW   1               ;*DZ11'S SELECTED ACTIVE.
1307  001406  000001                    RUN:    1                       ;*POINTER ONE PAST RUNNING DEVICE.
1308  001410  000001                    DZNUM:  .BLKB 1                 ;*OCTAL NUMBER OF DZ11'S.
1309  001411     001                    SAVNUM: .BYTE 1                 ;*WORKABLE NUMBER.
1310                                    .EVEN
1311  001412  001500                    ACTIVE: DZ.MAP                  ;TABLE POINTER.
```

```
1312
1313                                    ;PROGRAM CONTROL FLAGS
1314                                    ;-------------- ------
1315
1316  001414     000            EIAFLG: .BYTE   0               ;0=EIA  100000=20MA
1317  001415     000            INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
1318  001416     000            HDRFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG FOR HEADER MAP
1319  001417     000            MNTFLG: .BYTE   0               ;MAINTENANCE BIT SET FLAG
1320  001420     000            DONFLG: .BYTE   0               ;TRANSMISSION COMPLETION FLAG
1321             001422           .EVEN
1322                              ;DATA VARIABLES
1323  001422  000000             TD0:    .WORD   0
1324  001424  000000             TD1:    .WORD   0
1325  001426  000000             TD2:    .WORD   0
1326  001430  000000             TD3:    .WORD   0
1327  001432  000000             TD4:    .WORD   0
1328  001434  000000             TD5:    .WORD   0
1329  001436  000000             TD6:    .WORD   0
1330  001440  000000             TD7:    .WORD   0
1331  001442  000000             TR0:    .WORD   0
1332  001444  000000             TR1:    .WORD   0
1333  001446  000000             TR2:    .WORD   0
1334  001450  000000             TR3:    .WORD   0
1335  001452  000000             TR4:    .WORD   0
1336  001454  000000             TR5:    .WORD   0
1337  001456  000000             TR6:    .WORD   0
1338  001460  000000             TR7:    .WORD   0
1339  001462                     STOP:
1340                              ; -- END 0 MACRO ----------------
1341                              .SBTTL  APT PARAMETER BLOCK
1342
1343                              ;;*****************************************************************
1344                              ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1345                              ;;*****************************************************************
1346             001462                   .$X=.    ;;SAVE CURRENT LOCATION
1347             000024                   .=24     ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1348  000024  000200                     200      ;;FOR APT START UP
1349             000044                   .=44     ;;POINT TO APT INDIRECT ADDRESS PNTR.
1350  000044  001462                     $APTHDR  ;;POINT TO APT HEADER BLOCK
1351             001462                   .=.$X    ;;RESET LOCATION COUNTER
1352                              ;;*****************************************************************
1353                              ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1354                              ;INTERFACE SPEC.
1355
1356  001462                     $APTHD:
1357  001462  000000             $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1358  001464  001234             $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1359  001466  000132             $TSTM:  .WORD   90.     ;;RUN TIM OF LONGEST TEST
1360  001470  000137             $PASTM: .WORD   95.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1361  001472  000137             $UNITM: .WORD   95.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1362  001474  000052                     .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1363                                    ;DZ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
1364                                    ;---------------------------------------
1365
1366             001500           .=1500
1367  001500                     DZ.MAP:
```

```
                                                        ; -$JUNK-------------------------------------
001500 000001           DZCR0:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 0
001502 000001           DZVC0:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 0
001504 000001           DZLV0:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001506 000001           LINE0:  .BLKW   1       ;ALL LINES SELECTED
001510 000001           PAR0:   .BLKW   1       ;PARAMETERS
001512 000001           MANT0:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
                        ; -- END 0 MACRO ------------------
                                                        ; -$JUNK-------------------------------------
001514 000001           DZCR1:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 1
001516 000001           DZVC1:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 1
001520 000001           DZLV1:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001522 000001           LINE1:  .BLKW   1       ;ALL LINES SELECTED
001524 000001           PAR1:   .BLKW   1       ;PARAMETERS
001526 000001           MANT1:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
                        ; -- END 0 MACRO ------------------
                                                        ; -$JUNK-------------------------------------
001530 000001           DZCR2:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 2
001532 000001           DZVC2:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 2
001534 000001           DZLV2:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001536 000001           LINE2:  .BLKW   1       ;ALL LINES SELECTED
001540 000001           PAR2:   .BLKW   1       ;PARAMETERS
001542 000001           MANT2:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
                        ; -- END 0 MACRO ------------------
                                                        ; -$JUNK-------------------------------------
001544 000001           DZCR3:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 3
001546 000001           DZVC3:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 3
001550 000001           DZLV3:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001552 000001           LINE3:  .BLKW   1       ;ALL LINES SELECTED
001554 000001           PAR3:   .BLKW   1       ;PARAMETERS
001556 000001           MANT3:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
                        ; -- END 0 MACRO ------------------
                                                        ; -$JUNK-------------------------------------
001560 000001           DZCR4:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 4
001562 000001           DZVC4:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 4
001564 000001           DZLV4:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001566 000001           LINE4:  .BLKW   1       ;ALL LINES SELECTED
001570 000001           PAR4:   .BLKW   1       ;PARAMETERS
001572 000001           MANT4:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
                        ; -- END 0 MACRO ------------------
                                                        ; -$JUNK-------------------------------------
001574 000001           DZCR5:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 5
001576 000001           DZVC5:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 5
001600 000001           DZLV5:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001602 000001           LINE5:  .BLKW   1       ;ALL LINES SELECTED
001604 000001           PAR5:   .BLKW   1       ;PARAMETERS
001606 000001           MANT5:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
                        ; -- END 0 MACRO ------------------
                                                        ; -$JUNK-------------------------------------
```

```
1424  001610  000001                  DZCR6:   .BLKW   1        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 6
1425  001612  000001                  DZVC6:   .BLKW   1        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 6
1426  001614  000001                  DZLV6:   .BLKW   1        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1427  001616  000001                  LINE6:   .BLKW   1        ;ALL LINES SELECTED
1428  001620  000001                  PAR6:    .BLKW   1        ;PARAMETERS
1429  001622  000001                  MANT6:   .BLKW   1        ;MAINTENANCE MODE FOR THIS DEVICE
1430                                  ; -- END 0 MACRO ----------------
1431                                                            ; -$JUNK-----------------------------------
1432
1433  001624  000001                  DZCR7:   .BLKW   1        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 7
1434  001626  000001                  DZVC7:   .BLKW   1        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 7
1435  001630  000001                  DZLV7:   .BLKW   1        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1436  001632  000001                  LINE7:   .BLKW   1        ;ALL LINES SELECTED
1437  001634  000001                  PAR7:    .BLKW   1        ;PARAMETERS
1438  001636  000001                  MANT7:   .BLKW   1        ;MAINTENANCE MODE FOR THIS DEVICE
1439                                  ; -- END 0 MACRO ----------------
1440                                                            ; -$JUNK-----------------------------------
1441
1442  001640  000001                  DZCR10:  .BLKW   1        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 10
1443  001642  000001                  DZVC10:  .BLKW   1        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 10
1444  001644  000001                  DZLV10:  .BLKW   1        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1445  001646  000001                  LINE10:  .BLKW   1        ;ALL LINES SELECTED
1446  001650  000001                  PAR10:   .BLKW   1        ;PARAMETERS
1447  001652  000001                  MANT10:  .BLKW   1        ;MAINTENANCE MODE FOR THIS DEVICE
1448                                  ; -- END 0 MACRO ----------------
1449                                                            ; -$JUNK-----------------------------------
1450
1451  001654  000001                  DZCR11:  .BLKW   1        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 11
1452  001656  000001                  DZVC11:  .BLKW   1        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 11
1453  001660  000001                  DZLV11:  .BLKW   1        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1454  001662  000001                  LINE11:  .BLKW   1        ;ALL LINES SELECTED
1455  001664  000001                  PAR11:   .BLKW   1        ;PARAMETERS
1456  001666  000001                  MANT11:  .BLKW   1        ;MAINTENANCE MODE FOR THIS DEVICE
1457                                  ; -- END 0 MACRO ----------------
1458                                                            ; -$JUNK-----------------------------------
1459
1460  001670  000001                  DZCR12:  .BLKW   1        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 12
1461  001672  000001                  DZVC12:  .BLKW   1        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 12
1462  001674  000001                  DZLV12:  .BLKW   1        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1463  001676  000001                  LINE12:  .BLKW   1        ;ALL LINES SELECTED
1464  001700  000001                  PAR12:   .BLKW   1        ;PARAMETERS
1465  001702  000001                  MANT12:  .BLKW   1        ;MAINTENANCE MODE FOR THIS DEVICE
1466                                  ; -- END 0 MACRO ----------------
1467                                                            ; -$JUNK-----------------------------------
1468
1469  001704  000001                  DZCR13:  .BLKW   1        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 13
1470  001706  000001                  DZVC13:  .BLKW   1        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 13
1471  001710  000001                  DZLV13:  .BLKW   1        ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1472  001712  000001                  LINE13:  .BLKW   1        ;ALL LINES SELECTED
1473  001714  000001                  PAR13:   .BLKW   1        ;PARAMETERS
1474  001716  000001                  MANT13:  .BLKW   1        ;MAINTENANCE MODE FOR THIS DEVICE
1475                                  ; -- END 0 MACRO ----------------
1476                                                            ; -$JUNK-----------------------------------
1477
1478  001720  000001                  DZCR14:  .BLKW   1        ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 14
1479  001722  000001                  DZVC14:  .BLKW   1        ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 14
```

```
1480  001724  000001                  DZLV14: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1481  001726  000001                  LINE14: .BLKW   1       ;ALL LINES SELECTED
1482  001730  000001                  PAR14:  .BLKW   1       ;PARAMETERS
1483  001732  000001                  MAINT14: .BLKW  1       ;MAINTENANCE MODE FOR THIS DEVICE
1484                                  ; -- END 0 MACRO ----------------
1485                                                  ; -$JUNK-----------------------------------
1486
1487  001734  000001                  DZCR15: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 15
1488  001736  000001                  DZVC15: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 15
1489  001740  000001                  DZLV15: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1490  001742  000001                  LINE15: .BLKW   1       ;ALL LINES SELECTED
1491  001744  000001                  PAR15:  .BLKW   1       ;PARAMETERS
1492  001746  000001                  MAINT15: .BLKW  1       ;MAINTENANCE MODE FOR THIS DEVICE
1493                                  ; -- END 0 MACRO ----------------
1494                                                  ; -$JUNK-----------------------------------
1495
1496  001750  000001                  DZCR16: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 16
1497  001752  000001                  DZVC16: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 16
1498  001754  000001                  DZLV16: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1499  001756  000001                  LINE16: .BLKW   1       ;ALL LINES SELECTED
1500  001760  000001                  PAR16:  .BLKW   1       ;PARAMETERS
1501  001762  000001                  MAINT16: .BLKW  1       ;MAINTENANCE MODE FOR THIS DEVICE
1502                                  ; -- END 0 MACRO ----------------
1503                                                  ; -$JUNK-----------------------------------
1504
1505  001764  000001                  DZCR17: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 17
1506  001766  000001                  DZVC17: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 17
1507  001770  000001                  DZLV17: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
1508  001772  000001                  LINE17: .BLKW   1       ;ALL LINES SELECTED
1509  001774  000001                  PAR17:  .BLKW   1       ;PARAMETERS
1510  001776  000001                  MAINT17: .BLKW  1       ;MAINTENANCE MODE FOR THIS DEVICE
1511                                  ; -- END 0 MACRO ----------------
1512
1513  002000  177777                  DZ.END: 177777
```

```
1514                                    ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
1515                                    ;POINTERS TO SUBROUTINES CAN BE FOUND
1516                                    ;IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS
1517
1518                                ;:*****************************************************************
1519                                ;-----------------------------------------------------------------
1520  002002                        .TRPTAB:
1521          104400                ADVANCE=TRAP+0          ;CALL TO ADVANCE TO NEXT TEST( OR SCOPE THIS ONE)
1522  002002  007216                        .ADVANCE
1523                                ; -- END 0 MACRO ******************
1524          104401                SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
1525  002004  005376                        .SCOP1
1526                                ; -- END 0 MACRO ******************
1527          104402                TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
1528  002006  005422                        .TYPE
1529                                ; -- END 0 MACRO ******************
1530          104403                INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
1531  002010  006252                        .INSTR
1532                                ; -- END 0 MACRO ******************
1533          104404                INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
1534  002012  006426                        .INSTER
1535                                ; -- END 0 MACRO ******************
1536          104405                PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
1537  002014  006446                        .PARAM
1538                                ; -- END 0 MACRO ******************
1539          104406                SETFLG=TRAP+6           ;CALL TO  SET FLAG ROUTINE
1540  002016  011152                        .SETFLG
1541                                ; -- END 0 MACRO ******************
1542          104407                SAV05=TRAP+7            ;CALL TO REGISTER SAVE ROUTINE
1543  002020  006646                        .SAV05
1544                                ; -- END 0 MACRO ******************
1545          104410                RES05=TRAP+10           ;CALL TO REGISTER RESTORE ROUTINE
1546  002022  006706                        .RES05
1547                                ; -- END 0 MACRO ******************
1548          104411                CONVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUTINE
1549  002024  006740                        .CONVRT
1550                                ; -- END 0 MACRO ******************
1551          104412                CNVRT=TRAP+12           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
1552  002026  006744                        .CNVRT
1553                                ; -- END 0 MACRO ******************
1554          104413                DEVICE.CLR=TRAP+13              ;CALL TO ISSUE A DEVICE CLEAR
1555  002030  007144                        .DEVICE.CLR
1556                                ; -- END 0 MACRO ******************
1557          104414                DELAY=TRAP+14           ;CALL TO DELAY FOR FAST CPU'S
1558  002032  007176                        .DELAY
1559                                ; -- END 0 MACRO ******************
1560          104415                PARMD=TRAP+15           ;CONVERT DECIMAL STRING TO OCTAL
1561  002034  026744                        .PARMD
```

```
1562                                    ; -- END 0 MACRO ******************
1563            104416                  PAWCH=TRAP+16           ;SET FLAG    ECHO OR  CABLE
1564  002036   027140                          .PAWCH
1565                                    ; -- END 0 MACRO ******************
1566            104417                  DCLASM=TRAP+17          ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
1567  002040   007164                          .DCLASM
1568                                    ; -- END 0 MACRO ******************
1569
1570                                    ;-----------------------------------------------------------------------
1571                                    ;:***********************************************************************
```

```
                                        ;DZ11 VECTOR AND REGISTER INDIRECT POINTERS
                                        ;WORKING AREA

002042  160040                DZCSR:  160040  ;R/W
002044  160041                MDZCSR: 160041  ;R/W
002046  160042                DZRBUF: 160042  ;READ ONLY
002050  160043                MDZRBUF:        160043  ;READ ONLY
002052  160042                DZLPR:  160042  ;WRITE ONLY
002054  160043                MDZLPR: 160043  ;WRITE ONLY
002056  160044                DZTCR:  160044  ;R/W
002060  160045                MDZTCR: 160045  ;R/W
002062  160046                DZMSR:  160046  ;READ ONLY
002064  160047                MDZMSR: 160047  ;READ ONLY
002066  160046                DZTDR:  160046  ;WRITE ONLY
002070  160047                MDZTDR: 160047  ;WRITE ONLY
                                        ;DEFAULT DZ VECTORS
002072  000300                DZRIV:  300     ;REC INTR VECTOR
002074  000302                DZRIS:  302     ;REC INTR STATUS
002076  000304                DZTIV:  304     ;XMIT INTR VECTOR
002100  000306                DZTIS:  306     ;XMIT INTR STATUS
```

```
1594
1595                                    ;TIME TABLE FOR RELATIVE TIMING TESTS
1596                                    ;-----------------------------------
1597
1598  002102                            TMTBL:
1599  002102  000000                    T50:    0
1600  002104  000000                    T75:    0
1601  002106  000000                    T110:   0
1602  002110  000000                    T134:   0
1603  002112  000000                    T150:   0
1604  002114  000000                    T300:   0
1605  002116  000000                    T600:   0
1606  002120  000000                    T1200:  0
1607  002122  000000                    T1800:  0
1608  002124  000000                    T2000:  0
1609  002126  000000                    T2400:  0
1610  002130  000000                    T3600:  0
1611  002132  000000                    T4800:  0
1612  002134  000000                    T7200:  0
1613  002136  000000                    T9600:  0
1614  002140  000000                    TEIGHT:0
1615  002142  000000                    TSEVEN: 0
1616  002144  000000                    TSIX:   0
1617  002146  000000                    TFIVE:  0
```

```
1618                                        ;PROGRAM INITIALIZATION
1619                                        ;LOCK OUT INTERRUPTS
1620                                        ;SET UP PROCESSOR STACK
1621                                        ;SET UP POWER FAIL VECTOR
1622                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1623                                        ;TYPE TITLE MESSAGE
1624
1625
1626  002150                          .START:
1627  002150  000005                       RESET                   ;CLEAR THE WORLD. START NEW ENVIRONMENT
1628  002152  012706  001120               MOV     #STACK,SP       ;SET UP STACK
1629  002156  106427  000340               MTPS    #PR7            ;LOCK OUT INTERRUPTS
1630  002162  012737  010320  000024       MOV     #$PWRDN,@#24    ;SET UP POWER FAIL VECTOR
1631  002170  012737  005122  000020       MOV     #$SCOPE,@#IOTVEC        ;SET VECTOR FOR SCOPE ROUTINE
1632  002176  012737  000340  000022       MOV     #340,@#IOTVEC+2         ;LEVEL 7
1633  002204  012737  007230  000030       MOV     #$ERROR,@#EMTVEC        ;SET VECTOR FOR ERROR ROUTINE
1634  002212  012737  000340  000032       MOV     #340,@#EMTVEC+2         ;LEVEL 7
1635  002220  012737  007122  000034       MOV     #.TRPSR,@#TRAPVEC       ;SET VECTOR FOR TRAP CALLS
1636  002226  012737  000340  000036       MOV     #340,@#TRAPVEC+2        ;LEVEL 7
1637  002234  012737  010320  000024       MOV     #$PWRDN,@#PWRVEC        ;SET VECTOR FOR POWER FAIL ROUTINE
1638  002242  012737  000340  000026       MOV     #340,@#PWRVEC+2         ;LEVEL 7
1639  002250  113737  001410  001411       MOVB    DZNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
1640  002256  005037  001242               CLR     $PASS           ;CLEAR PASS COUNT
1641  002262  105037  001123               CLRB    $ERFLG          ;CLEAR ERROR FLAG
1642  002266  012737  001500  001412       MOV     #DZ.MAP,ACTIVE  ;GET MAP POINTER.
1643  002274  012737  000001  001406       MOV     #1,RUN          ;POINT POINTER TO FIRST DEVICE.
1644  002302  005037  001132               CLR     $ERTTL          ;CLEAR ERROR COUNT
1645  002306  005037  001136               CLR     $ERRPC          ;CLEAR LAST ERROR POINTER
1646  002312  005037  001122               CLR     $TSTNM          ;SET UP FOR TEST 1
1647  002316  012737  002150  001126       MOV     #.START,@LPADR  ;SET UP FOR POWER FAIL BEFORE
1648                                                                ;TESTING STARTS
1649                                        ;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY
1650  002324  013746  000006               MOV     6,-(SP)         ;SAVE BUS ERROR PS
1651  002330  013746  000004               MOV     4,-(SP)         ;SAVE BUS ERROR PC
1652  002334  012737  002354  000004       MOV     #20$,4          ;SET UP TO TRAP TO THIS ROUTINE
1653  002342  022777  177777  176610       CMP     #-1,@SWR        ;CAN 177570 BE REFERENCED?
1654  002350  001402                       BEQ     22$             ;IF SO AND IT IS -1, TREAT LIKE SWITCHLESS
1655  002352  000407                       BR      21$             ;IF YES,SKIP AROUND THE SETUP
1656  002354  022626                20$:   POP2SP                  ;REMOVE THE TRAP FROM THE STACK
1657  002356  012737  000176  001160  22$:   MOV     #SWREG,SWR      ;IF NO,TRAP COMES HERE.POINT TO SOFTWARE SWR
1658  002364  012737  000174  001162       MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REGISTER
1659  002372  012637  000004          21$:   MOV     (SP)+,4         ;RESTORE THE BUS ERROR VECTOR
1660  002376  012637  000006               MOV     (SP)+,6
1661  002402  105737  001415               TSTB    INIFLG          ;TITLE ALREADY PRINTED?
1662  002406  001010                       BNE     29$             ;BRANCH IF YES
1663  002410  023727  000042  005056       CMP     @#42,#$ENDAD    ;RUNNING UNDER ACT?
1664  002416  001402                       BEQ     31$             ;IF YES DONT PRINT TITLE
1665  002420  104402  001000               TYPE    ,MTITLE         ;PRINT THE DIAGNOSTIC'S TITLE
1666  002424  105337  001415          31$:   DECB    INIFLG          ;SET THE ONCE ONLY FLAG
1667  002430  105737  001255          29$:   TSTB    $ENVM           ;DETERMINE WHETHER APT SIZING SHOULD BE DONE
1668  002434  100006                       BPL     30$             ;IF NOT, GO CHECK FOR AUTO-SIZING
1669  002436  004737  012112               JSR     PC,SETAPT       ;OTHERWISE, GO DO APT SIZING FROM ETABLE
1670                                  ;REPLACE "CLRB HDRFLG" WITH NOP'S
1671  002442  000240                       NOP
1672  002444  000240                       NOP
1673                                  ;       CLRB    HDRFLG          ;MAKE SURE STATUS TABLE IS PRINTED
```

```
1674  002446  000137  004420                    JMP     16$                ;GO PRINT DZ STATUS TABLE
1675  002452  032777  000001  176500  30$:      BIT     @SW00,@SWR         ;RESELECT ?
1676  002460  001011                            BNE     32$                ;IF YES, GO SET UP THE INFORMATION
1677  002462  122737  000377  001415            CMPB    @377,INIFLG        ;ON 1ST START; MUST ANSWER QUESTION
1678  002470  001003                            BNE     .+10               ;IF NOT ANSWERING QUESTIONS
1679  002472  105777  176462                    TSTB    @SWR               ;ARE U AUTO SIZING?
1680  002476  100402                            BMI     32$                ;NO AUTO SIZE! NO SW00=1 ON 1ST START!
1681  002500  000137  003244                    JMP     73$                ;IF NO, SKIP THE INTERROGATION
1682  002504  012700  001500          32$:      MOV     @DZ.MAP,R0         ;POINT TO THE BEGINNING OF THE MAP TABLE
1683  002510  105037  001416                    CLRB    HDRFLG             ;MAKE SURE A MAP GETS PRINTED
1684  002514  005020                  65$:      CLR     (R0)+              ;CLEAR A TABLE LOCATION
1685  002516  020027  002000                    CMP     R0,@DZ.END         ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
1686  002522  001374                            BNE     65$                ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE
1687  002524  105337  001415                    DECB    INIFLG             ;INSURE NO AUTO SIZING IF QUESTIONS ANSWERED!
1688
1689                                  ;THE FOLLOWING ARE PARAMETERS USED TO FILL IN THE MAP
1690                                  ;TABLE AND SET UP THE DIAGNOSTIC.
1691
1692                                          ;GET THE BASE ADDRESS OF THE DZ11'S
1693
1694  002530                          33$:
1695                                                                     ; -$GETPAR----------------------------------
1696  002530  104403                          INSTR                     ;CALL THE STRING INPUT ROUTINE
1697  002532  003464                          66$                       ;POINTER TO MESSAGE TO BE PRINTED
1698  002534  104405                          PARAM                     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1699  002536  160000                          160000                    ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1700  002540  163770                          163770                    ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1701  002542  001500                          DZCR0                     ;POINTER TO MAP LOCATION TO BE FILLED
1702  002544     007                          .BYTE   7                 ;MASK OF INVALID BITS FOR THIS PARAMETER
1703  002545     001                          .BYTE   1                 ;NUMBER OF PARAMETERS TO STORE
1704                                  ; -- END 0 MACRO ------------------
1705  002546  013737  001500  001310          MOV     DZCR0,$BASE       ;COPY BASE ADDRESS TO ETABLE
1706
1707                                          ;GET THE BASE VECTOR ADDRESS
1708
1709  002554                          34$:
1710                                                                     ; -$GETPAR----------------------------------
1711  002554  104403                          INSTR                     ;CALL THE STRING INPUT ROUTINE
1712  002556  003530                          67$                       ;POINTER TO MESSAGE TO BE PRINTED
1713  002560  104405                          PARAM                     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1714  002562  000300                          300                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1715  002564  000776                          776                       ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1716  002566  001502                          DZVC0                     ;POINTER TO MAP LOCATION TO BE FILLED
1717  002570     003                          .BYTE   3                 ;MASK OF INVALID BITS FOR THIS PARAMETER
1718  002571     001                          .BYTE   1                 ;NUMBER OF PARAMETERS TO STORE
1719                                  ; -- END 0 MACRO ------------------
1720  002572  013737  001502  001304          MOV     DZVC0,$VECT1      ;COPY VECTOR TO ETABLE
1721
1722                                          ;GET THE BUS REQUEST LEVEL
1723
1724                                                                     ; -$GETPAR----------------------------------
1725  002600  104403                          INSTR                     ;CALL THE STRING INPUT ROUTINE
1726  002602  003571                          68$                       ;POINTER TO MESSAGE TO BE PRINTED
1727  002604  104405                          PARAM                     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1728  002606  000004                          4                         ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1729  002610  000007                          7                         ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
```

```
1730  002612  001504                          DZLVO                   ;POINTER TO MAP LOCATION TO BE FILLED
1731  002614     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1732  002615     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1733                                  ; -- END 0 MACRO ------------------
1734  002616  113737  001504  001305          MOVB    DZLV0,$VECT1+1  ;GET BUS REQUEST LEVEL INTO ETABLE
1735  002624  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1736  002630  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1737  002634  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1738  002640  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1739  002644  106337  001305                  ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1740
1741                                          ;FIND OUT IF MODULE IS EIA OR 20 MA.
1742
1743  002650  104402  004260                  TYPE    .74$            ;PRINT EIA MESSAGE
1744  002654  005037  001220                  CLR     $TMP1           ;USE $TMP1
1745  002660  105777  176300          80$:    TSTB    @$TKS           ;IS KEYBOARD DONE?
1746  002664  100375                  BPL     80$             ;IF NOT, WAIT FOR IT
1747  002666  017746  176274                  MOV     @$TKB,-(SP)     ;IF YES, PUT CHARACETR ON STACK
1748  002672  042716  177600                  BIC     #177600,(SP)    ;STRIP DOWN CHARACTER           ;;DSH-BHL
1749  002676  122716  000023                  CMPB    #$XOFF,(SP)     ;IS IT A XOFF?                  ;;DSH-BHL
1750  002702  001014                  BNE     83$             ;BR IF NOT                      ;;DSH
1751  002704  105777  176254          101$:   TSTB    @$TKS           ;WAIT FOR A CHARACTER           ;;DSH-BHL
1752  002710  100375                  BPL     101$                                            ;;DSH-BHL
1753  002712  117716  176250                  MOVB    @$TKB,(SP)      ;GET CHARACTER                  ;;DSH-BHL
1754  002716  042716  177600                  BIC     #177600,(SP)    ;STRIP DOWN CHARACTER           ;;DSH-BHL
1755  002722  122716  000021                  CMPB    #$XON,(SP)      ;WAIT FOR A XON?                ;;DSH-BHL
1756  002726  001366                  BNE     101$            ;GET NEXT CHAR IF NOT           ;;DSH-BHL
1757  002730  005726                  TST     (SP)+           ;POP STACK                      ;;DSH
1758  002732  000752                  BR      80$             ;WAIT FOR A CHAR                ;;DSH
1759  002734  122716  000021          83$:    CMPB    #$XON,(SP)      ;IS IT A RANDOM XON             ;;DSH-BHL
1760  002740  001002                  BNE     102$            ;BR IF NO                       ;;DSH-BHL
1761  002742  005726                  TST     (SP)+           ;ELSE, POP STACK                ;;DSH-BHL
1762  002744  000745                  BR      80$             ;GO GET NEXT CHAP               ;;DSH-BHL
1763  002746  122726  000015          102$:   CMPB    #15,(SP)+       ;IS IT <CR> ?
1764  002752  001414                  BEQ     81$             ;IF SO, GET OUT
1765  002754  014677  176212                  MOV     -(SP),@$TPB     ;IF NOT, PRINT CHARACTER
1766  002760  042737  100000  001504          BIC     #BIT15,DZLV0    ;CLEAR EIA FLAG
1767  002766  122726  000102                  CMPB    #102,(SP)+      ;IS IT A B?
1768  002772  001332                  BNE     80$             ;IF NOT, GO BACK FOR INPUT
1769  002774  052737  100000  001504          BIS     #BIT15,DZLV0    ;IF SO, SET FLAG
1770  003002  000726                  BR      80$             ;GET MORE INPUT
1771  003004                          81$:
1772
1773                                          ;GET THE MODE OF OPERATION (E,I,S)
1774
1775                                                          ; -$GETFLG-----------------------------
1776  003004  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
1777  003006  004002                          72$                     ;POINTER TO THE MESSAGE TO BE PRINTED
1778  003010  104406                          SETFLG                  ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
1779  003012  001512                          MANTO                   ;THIS IS THE FLAG BEING SETUP
1780                                  ; -- END 0 MACRO ------------------
1781
1782                                          ;GET THE NUMBER OF DZ11'S RUNNING
1783
1784                                                          ; -$GETPAR-----------------------------
1785  003014  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
```

```
1786  003016  003740                          71$                     ;POINTER TO MESSAGE TO BE PRINTED
1787  003020  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1788  003022  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1789  003024  000020                          16.                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1790  003026  001220                          $TMP1                   ;POINTER TO MAP LOCATION TO BE FILLED
1791  003030     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1792  003031     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1793                                  ; -- END 0 MACRO ------------------
1794
1795  003032  012737  000377  001506          MOV     #377,LINE0      ;SET UP DEFAULT LINES
1796  003040  012737  017070  001510          MOV     #17070,PAR0     ;SET UP DEFAULT LPR PARAMETER
1797                                                                  ;RECEIVER ON; 9600 BAUD; 2STOP BITS; 8 BIT/CHAR
1798  003046  012737  000001  007214          MOV     #1,DLYCNT       ;INITIALIZE DELAY COUNT
1799  003054  032777  000010  176076          BIT     #SW03,@SWR      ;DO YOU WANT PARAMETERS?
1800  003062  001402                          BEQ     40$             ;IF NO, SKIP THE PARAMETER CALL
1801  003064  004737  003274                  JSR     PC,23$          ;GET PARAMETERS
1802  003070  012737  000001  001312  40$:    MOV     #1,$DEVM        ;INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
1803  003076  113737  001220  001410          MOVB    $TMP1,DZNUM     ;COPY THE NUMBER OF DEVICES
1804  003104  113737  001220  001411          MOVB    $TMP1,SAVNUM    ;COPY A BACKUP NUMBER
1805  003112  005337  001220          62$:    DEC     $TMP1           ;$TMP1 CONTAINS THE COUNT OF UNINITIALIZED
1806  003116  001404                          BEQ     61$             ; SELECTED DEVICES
1807  003120  000261                          SEC                     ;SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
1808  003122  006137  001312                  ROL     $DEVM           ;POINT TO THE NEXT DEVICE
1809  003126  000771                          BR      62$             ;GO DO THIS PROCEDURE AGAIN
1810  003130  013737  001312  001222  61$:    MOV     $DEVM,$TMP2     ;# OF TIMES
1811  003136  013737  001312  001404          MOV     $DEVM,DZACTV    ;COPY THE ACTIVE DEVICE PARAMETER
1812  003144  012700  0015C                   MOV     #DZCR0,R0       ;SET A POINTER TO THE SPECIFIED INFORMATION
1813  003150  012701  001514                  MOV     #DZCR1,R1       ;POINT R1 TO THE REST OF THE MAP TABLE
1814  003154  012702  001320                  MOV     #$DDW0,R2       ;POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
1815  003160  000241                          CLC                     ;INITIALIZE THE "C" BIT FOR A ROTATION
1816  003162  006037  001222                  ROR     $TMP2           ;SKIP MAPPING SETUP FOR DEVICE 0- IT'S DONE
1817  003166  006237  001222          64$:    ASR     $TMP2           ;ISOLATE A SELECTION FLAG IN THE "C" BIT
1818  003172  103404                          BCS     41$             ;IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
1819  003174  012711  177777                  MOV     #-1,(R1)        ;TERMINATE THE LIST
1820  003200  000137  004374                  JMP     63$             ;GO TO THE NEXT BLOCK
1821  003204  012011                  41$:    MOV     (R0)+,(R1)      ;ADDRESS
1822  003206  062721  000010                  ADD     #10,(R1)+       ;POINT TO THE NEXT DZ11 ADDRESS VALUE
1823  003212  012011                          MOV     (R0)+,(R1)      ;VECTOR
1824  003214  062721  000010                  ADD     #10,(R1)+       ;POINT TO THE NEXT VECTOR VALUE
1825  003220  012021                          MOV     (R0)+,(R1)+     ;LEVEL
1826  003222  012021                          MOV     (R0)+,(R1)+     ;LINES
1827  003224  016012  177774                  MOV     -4(R0),(R2)     ;GET THE EIA FLAG FROM THE PRIORITY WORD
1828  003230  042712  077777                  BIC     #77777,(R2)     ;ISOLATE THAT FLAG
1829  003234  051022                          BIS     (R0),(R2)+      ;ADD PARAMETERS TO DEVICE DESCRIPTOR WORD
1830  003236  012021                          MOV     (R0)+,(R1)+     ;PARAMETERS
1831  003240  012021                          MOV     (R0)+,(R1)+     ;MAINTENANCE MODE
1832  003242  000751                          BR      64$
1833  003244  032777  000010  175706  73$:    BIT     #SW03,@SWR      ;ASK PARAMETERS ?
1834  003252  001002                          BNE     42$             ;IF NO, GO DO AUTO SIZING
1835  003254  000137  004374                  JMP     63$             ;GO SET UP FOR AUTO SIZING
1836  003260  004737  003274          42$:    JSR     PC,23$          ;GO ASK PARAMETERS
1837  003264  105337  001415                  DECB    INIFLG          ;INSURE NO AUTO SIZE IF QUESTIONS ANSWERED
1838  003270  000137  004420                  JMP     16$             ;GO TO THE NEXT BLOCK
1839
1840                                          ;GET THE ACTIVE LINES PARAMETER
1841
```

```
1842  003274                          23$:
1843                                                    ; -$GETPAR----------
1844  003274  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
1845  003276  003614                          69$                     ;POINTER TO MESSAGE TO BE PRINTED
1846  003300  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1847  003302  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1848  003304  000377                          377                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1849  003306  001506                          LINE0                   ;POINTER TO MAP LOCATION TO BE FILLED
1850  003310     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1851  003311     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1852                                  ; -- END 0 MACRO ----------
1853  003312  105037  001416                  CLRB    HDRFLG          ;MAKE SURE THE CHANGES ARE PRINTED
1854
1855                                  ;THIS SEGMENT CHECKS TO MAKE SURE THE LINE PARAMETER JUST ENTERED
1856                                  ;IS LEGITIMATE IN STAGGERED MODE OPERATION IF THAT MODE WAS SELECTED
1857
1858  003316  005737  001512                  TST     MANTO           ;IS STAGGERED THE MODE OF OPERATION?
1859  003322  100021                          BPL     26$             ;IF NOT, SKIP THIS SEGMENT
1860  003324  013703  001506                  MOV     LINE0,R3        ;GET A SCRATCH COPY OF THE ACTIVE LINES
1861  003330  006003                  24$:    ROR     R3              ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
1862  003332  103410                          BCS     25$             ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
1863  003334  001414                          BEQ     26$             ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
1864  003336  006203                          ASR     R3              ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
1865  003340  103373                          BCC     24$             ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
1866  003342  104402  001230          27$:    TYPE    .$QUES          ;THIS IS AN INCORRECT PARAMETER
1867  003346  104402  011075                  TYPE    .MBADLN         ;LET THE USER KNOW ABOUT IT
1868  003352  000750                          BR      23$             ;GO GET THE CORRECT PARAMETER
1869  003354  001772                  25$:    BEQ     27$             ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
1870  003356  006203                          ASR     R3              ;GET THE NEXT FLAG
1871  003360  103370                          BCC     27$             ;IF IT ISN'T SET, THERE'S AN ERROR
1872  003362  000241                          CLC                     ;INITIALIZE THE "C" BIT FOR TESTING OF THE NEXT PAIR
1873  003364  000761                          BR      24$             ;GO TEST THE NEXT PAIR OF FLAGS
1874
1875                                          ;GET THE LINE PARAMETER REGISTER ARGUMENT
1876
1877  003366                          26$:
1878                                                    ; -$GETPAR----------
1879  003366  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
1880  003370  003670                          70$                     ;POINTER TO MESSAGE TO BE PRINTED
1881  003372  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1882  003374  000000                          0                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1883  003376  000017                          17                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1884  003400  001510                          PAR0                    ;POINTER TO MAP LOCATION TO BE FILLED
1885  003402     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1886  003403     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1887                                  ; -- END 0 MACRO ----------
1888  003404  012702  001506                  MOV     #LINE0,R2       ;POINT TO THE LINE SELECTION PARAMETER
1889  003410  012703  001510                  MOV     #PAR0,R3        ;POINT TO THE CHOSEN PARAMETERS
1890  003414  011304                          MOV     (R3),R4         ;USE BAUD RATE AS AN INDEX IN DELAY TABLE
1891  003416  006304                          ASL     R4              ;ALIGN INDEX ON WORD BOUNDARY
1892  003420  016437  032416  007214          MOV     DLYTBL(R4),DLYCNT ;SET THE DELAY COUNT FOR THIS BAUD RATE
1893  003426  000313                          SWAB    (R3)            ;PLACE IN HIGH BYTE
1894  003430  052713  010070                  BIS     #10070,(R3)     ;PLACE EXTRA PARAMETERS INTO LOC
1895  003434  011262  000014          28$:    MOV     (R2),14(R2)     ;LOAD THE LINES
1896  003440  011363  000014                  MOV     (R3),14(R3)     ;LOAD THE PARAMETERS
1897  003444  062702  000014                  ADD     #14,R2          ;POINT TO THE NEXT SET
```

```
1898  003450  062703  000014                    ADD     #14,R3          ; .. OF BOTH PARAMETERS
1899  003454  020327  001774                    CMP     R3,#PAR17       ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
1900  003460  001365                            BNE     28$             ;IF NOT, GO LOAD SOME MORE PARAMETERS
1901  003462  000207                            RTS     PC              ;RETURN TO CALLING BLOCK
1902  003464  030600  052123  041440  66$:     .ASCIZ  <200>/1ST CSR ADDRESS (160000:163700):  /
 (1)  003530  030600  052123  053040  67$:     .ASCIZ  <200>/1ST VECTOR ADDRESS (300:770):  /
 (1)  003571     200  051102  046040  68$:     .ASCIZ  <200>/BR LEVEL (4:6):  /
 (1)  003614  046200  047111  051505  69$:     .ASCIZ  <200>/LINES ACTIVE BY BIT <IN OCTAL>(001:377):  /
 (1)  003670  042200  043105  052501  70$:     .ASCIZ  <200>/DEFAULT BAUD RATE <IN OCTAL>(00:16):  /
 (1)  003740  021600  047440  020106  71$:     .ASCIZ  <200>/# OF DZ11'S <IN OCTAL> (1:20):  /
 (1)  004002  046600  044501  052116  72$:     .ASCII  <200>/MAINTENANCE MODE/
 (1)  004023     200  055440  054105            .ASCII  <200>/ [EXTERNAL  <M325>-EIA ONLY      (E)]/
 (1)  004071     200  055440  047111            .ASCII  <200>/ [INTERNAL  <DZCSR03=1>          (I)]/
 (1)  004137     200  055440  052123            .ASCII  <200>/ [STAGGERED <M3271:-EIA ONLY     (S)]: /
 (1)  004207     200  055440  052123            .ASCII  <200>/ [STAGGERED <M3190>-20MA ONLY    (S)]: /
 (1)  004260  052200  050131  020105  74$:     .ASCIZ  <200>/TYPE "A" FOR EIA MODULE OR "B" FOR 20 MA (A:B): /
 (1)  004342  042600  052116  051105  75$:     .ASCIZ  <200>/ENTER DELAY PARAMETER: /
 (1)          004374                           .EVEN
 (1)  004374                            63$:
1903  004374  122737  000377  001415            CMPB    #377,INIFLG     ;ONLY DO AUTO SIZE ON 1ST START
1904  004402  001006                            BNE     16$             ;
1905  004404  032777  000200  174546            BIT     #BIT7,@SWR      ;BIT7=1??
1906  004412  001002                            BNE     16$             ;BR IF NO AUTO SIZE
1907  004414  004737  012264                    JSR     PC,AUTO.SIZE    ;GO DO THE AUTO SIZE
1908  004420  105737  001416            16$:    TSTB    HDRFLG          ;HAS THE TABLE BEEN TYPED YET?
1909  004424  001021                            BNE     1$              ;IF SO, DON'T TYPE IT AGAIN
1910  004426  105337  001416                    DECB    HDRFLG          ;INDICATE THAT THE TABLE WILL BE TYPED
1911  004432  104402  011050                    TYPE    .XHEAD          ;TYPE MAP HEADER
1912  004436  012700  001500                    MOV     #DZ.MAP,R0      ;SET POINTER
1913  004442  010037  001220            5$:     MOV     R0,$TMP1        ;POINT TO THE MAP LOCATION
1914  004446  012037  001222                    MOV     (R0)+,$TMP2     ;SET DATA
1915  004452  022737  177777  001222            CMP     #-1,$TMP2       ;END OF LIST?
1916  004460  001403                            BEQ     1$              ;BR IF YES
1917  004462  104411                    17$:    CONVRT                  ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
1918  004464  011140                            XSTATQ                  ;CONVERT THE DATA AT THIS ADDRESS
1919  004466  000765                            BR      5$              ;GO PRINT THE NEXT PARAMETER
1920  004470  005737  000042            1$:     TST     @#42            ;IS PROGRAM RUNNING UNDER MONITOR
1921  004474  001026                            BNE     3$              ;YES
1922  004476  032777  000100  174454            BIT     #SW06,@SWR      ;DESELECT SPECIFIC DEVICES??
1923  004504  001422                            BEQ     3$              ;BR IF NO.
1924  004506  104402  010771                    TYPE    .MNEW           ;TYPE THE MESSAGE.
1925  004512  005000                            CLR     R0              ;ZERO DATA DISPLAY
1926  004514  000000                            HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1927  004516  027737  174436  001312            CMP     @SWR,$DEVM      ;IS THE NUMBER VALID?
1928  004524  101404                            BLOS    2$              ;BR IF NUMBER IS OK.
1929  004526  104402  010643                    TYPE    .MERR3          ;TELL USER OF INVALID NUMBER.
1930  004532  000000                    9$:     HALT                    ;STOP EVERY THING.
1931  004534  000776                            BR      9$              ;RESTART THE PROGRAM AGAIN.
1932  004536  017737  174416  001404    2$:     MOV     @SWR,DZACTV     ;GET NEW DEVICE PATTERN
1933  004544  013700  001404                    MOV     DZACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
1934  004550  000000                            HALT                    ;CONTINUE DYNAMIC SWITCHES.
1935  004552  032777  000020  174400    3$:     BIT     #SW04,@SWR      ;CHECK TO SEE IF DELAY COUNT CHANGES
1936  004560  001407                            BEQ     18$             ;IF NOT, GO CLEAR VECTOR AREA
1937                                            ; -$GETPAR-------------------------------
1938  004562  104403                            INSTR                   ;CALL THE STRING INPUT ROUTINE
1939  004564  004342                            75$                     ;POINTER TO MESSAGE TO BE PRINTED
```

```
1940  004566  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1941  004570  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1942  004572  177777                          177777                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1943  004574  007214                          DLYCNT                  ;POINTER TO MAP LOCATION TO BE FILLED
1944  004576     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1945  004577     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1946                                  ; -- END 0 MACRO ------------------
1947  004600  012700  000300          18$:    MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
1948  004604  012701  000302                  MOV     #302,R1         ;VECTOR AREA. 300-776
1949  004610  010120                  4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
1950  004612  005021                          CLR     (R1)+           ;IN VECTOR AREA.
1951  004614  022021                          CMP     (R0)+,(R1)+     ;POP POINTERS
1952  004616  022700  001000                  CMP     #1000,R0        ;ALL DONE??
1953  004622  001372                          BNE     4$              ;BR IF NO.
1954
1955                                  ;TEST START AND RESTART
1956                                  ;----------------------
1957
1958  004624  012706  001120          .BEGIN: MOV     #STACK,SP       ;SET UP STACK
1959  004630  106427  000340                  MTPS    #PR7            ;LOCK OUT INTERRUPTS
1960  004634  005737  000042                  TST     @#42            ;IS PROGRAM UNDER MONITOR CONTROL
1961  004640  001015                          BNE     2$              ;BR IF YES
1962  004642  032777  000004  174310          BIT     #BIT2,@SWR      ;CHECK FOR LOCK ON TEST
1963  004650  001406                          BEQ     1$              ;BR IF NO LOCK DESIRED.
1964  004652  104402  010667                  TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
1965  004656  012737  000240  005140          MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
1966  004664  000403                          BR      2$              ;CONTINUE ALONG.
1967  004666  013737  005372  005140  1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
1968  004674  012737  011542  001126  2$:     MOV     #CYCLE,@LPADR   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
1969  004702  104402  010560                  TYPE    ,MR             ;TYPE "RUNNING"
1970  004706  000177  174214                  JMP     @@LPADR         ;START TESTING
1971                                  ; -- END 0 MACRO ------------------
1972                                  ; -PRGEND-------------------------------
```

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

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

004712                          $EOP:
                                                        ; -PASEND-----------------------------------
004712  000004                          SCOPE
004714  005037  001136                  CLR     $ERRPC          ;CLEAR LAST ERROR PC
004720  105037  001123                  CLRB    $ERFLG          ;CLEAR ERROR FLAG
004724  104402  010535                  TYPE    .MEPASS         ;TYPE END PASS
004730  104402  010716                  TYPE    .MCSRX          ;TYPE CSR
004734  104412  005072                  CNVRT   .XCSR           ;SHOW IT
004740  104402  010724                  TYPE    .MVECX          ;TYPE VECTOR
004744  104412  005100                  CNVRT   .XVEC           ;SHOW IT
004750  005237  001242                  INC     $PASS           ;RAISE PASS COUNT
004754  104402  010732                  TYPE    .MPASSX         ;TYPE PASSES
004760  104412  005106                  CNVRT   .XPASS          ;SHOW IT
004764  005337  001242                  DEC     $PASS           ;RESTORE PASS COUNT
004770  104402  010743                  TYPE    .MERRX          ;TYPE ERRORS
004774  104412  005114                  CNVRT   .XERR           ;SHOW IT
005000  105337  001411                  DECB    SAVNUM          ;ARE ALL DEVICES TESTED?
005004  001030                          BNE     $DOAGN          ;BR IF NO.
005006  113737  001410  001411          MOVB    DZNUM,SAVNUM    ;RESTORE THE COUNT
005014  005037  001226                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
005020  005237  001242                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
005024  042737  100000  001242          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
005032  005327                          DEC     (PC)+           ;;LOOP?
005034  000001                  $EOPCT: .WORD   1
005036  003013                          BGT     $DOAGN          ;;YES
005040  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
005042  000001                  $ENDCT: .WORD   1
005044  005034                          $EOPCT
005046  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
005052  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
005054  000005                          RESET                   ;;CLEAR THE WORLD
005056  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
005060  000240                          NOP                     ;;SAVE ROOM
005062  000240                          NOP                     ;;FOR
005064  000240                          NOP                     ;;ACT11
005066                          $DOAGN:
005066  000137                          JMP     @(PC)+          ;;RETURN
005070  011542                  $RTNAD: .WORD   CYCLE

005072  000001                  XCSR:   1
005074     006     0C2                  .BYTE   6,2
005076  002042                          DZCSR
005100  000001                  XVEC:   1
005102     003     002                  .BYTE   3,2
```

```
                                        DZRIV
                                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

005122                          @SCOPE:
                                                        ; -SC-----------------------------------
005122  004737  007652          .SCOPE: JSR     PC,SERV.G       ;FIND OUT IF <^G> WAS HIT
005126  005037  001136                  CLR     @ERRPC          ;CLEAR LAST ERROR PC.
005132  022716  013050                  CMP     #TST1+2,(SP)    ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
005136  001417                          BEQ     @XTSTR          ;IF SO, DON'T LOOP ON IT
005140  000412                  TTST:   BR      1@              ;GOTO 1@    (IF LOCK SW02=1; THIS LOC =240)
005142  105777  174016                  TSTB    @@TKS           ;KEYBOARD DONE?
005146  100073                          BPL     @OVER           ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
005150  127727  174012  000021          CMPB    @@TKB,#@XON     ;IS CHAR A RANDOM XON ?         ;;DSH
005156  001467                          BEQ     @OVER           ;BR IF YES                      ;;DSH
005160  017766  174002  177776          MOV     @@TKB,-2(SP)
                                ; -- END 0 MACRO -----------------
005166  032777  040000  173764  1@:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
005174  001060                          BNE     @OVER           ;;YES IF SW14=1
                                ;@@@@@START OF CODE FOR THE XOR TESTER@@@@@
005176  000416                  @XTSTR: BR      6@              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
005200  013746  000004                  MOV     @@ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
005204  012737  005224  000004          MOV     #5@,@@ERRVEC    ;;SET FOR TIMEOUT
005212  005737  177060                  TST     @@177060        ;;TIME OUT ON XOR?
005216  012637  000004                  MOV     (SP)+,@@ERRVEC  ;;RESTORE THE ERROR VECTOR
005222  000436                          BR      @SVLAD          ;;GO TO THE NEXT TEST
005224  022626                  5@:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
005226  012637  000004                  MOV     (SP)+,@@ERRVEC  ;;RESTORE THE ERROR VECTOR
005232  000441                          BR      @OVER           ;;LOOP ON THE PRESENT TEST
005234                          6@:;@@@@@END OF CODE FOR THE XOR TESTER@@@@@
005234  105737  001123          2@:     TSTB    @ERFLG          ;;HAS AN ERROR OCCURRED?
005240  001404                          BEQ     3@              ;;BR IF NO
005242  105037  001123          4@:     CLRB    @ERFLG          ;;ZERO THE ERROR FLAG
005246  005037  001226                  CLR     @TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
005252  032777  004000  173700  3@:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
005260  001011                          BNE     1@              ;;BR IF YES
```

```
2085  005262  005737  001242                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
2086  005266  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
2087  005270  005237  001124                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
2088  005274  023737  001226  001124          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
2089  005302  002015                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
2090  005304  012737  000001  001124  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
2091  005312  013737  005374  001226          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
2092  005320  105237  001122          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
2093  005324  113737  001122  001240          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
2094  005332  011637  001126                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
2095  005336  013777  001122  173616  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
2096  005344  013716  001126                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
2097                                                                  ; -SC1-----------------------------------
2098  005350  105037  001417          3$:     CLRB    MNTFLG          ;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TEST
2099  005354  005737  001370                  TST     MODE            ;HAS THE MODE BEEN CHANGED?
2100  005360  001003                          BNE     4$              ;IF NOT INTERNAL , GO DO A TEST
2101  005362  112737  000010  001417          MOVB    #MAINT,MNTFLG   ;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
2102  005370  000002                  4$:     RTI                     ;GO DO THE TEST
2103  005372  000406                  BRW:    406
2104                                  ; -- END O MACRO ----------------
2105  005374  000005                  $MXCNT: 5                       ;;MAX. NUMBER OF ITERATIONS
2106
2107                                          ;CHECK FOR FREEZE ON CURRENT DATA
2108                                          ;-------------------------------
2109
2110  005376  032777  001000  173554  .SCOP1: BIT     #SW09,@SWR      ;IS SW09=1(SET)?
2111  005404  001405                          BEQ     1$              ;BR IF NOT SET.
2112  005406  005737  001362                  TST     LOCK            ;IS THER A TIGHT LOOP SPECIFIED?
2113  005412  001402                          BEQ     1$              ;IF NO, RETURN
2114  005414  013716  001362                  MOV     LOCK,(SP)       ;IF YES, GOTO THE ADDRESS IN LOCK.
2115  005420  000002                  1$:     RTI                     ;GO BACK.
2116
2117  005422  032777  010000  173530  .TYPE:  BIT     #SW12,@SWR      ;INHIBIT ALL PRINTOUT??
2118  005430  001403                          BEQ     1$              ;IF NOT, GO TYPE
2119  005432  062716  000002                  ADD     #2,(SP)         ;SKIP OVER MESSAGE POINTER
2120  005436  000002                          RTI                     ;RETURN TO WHERE PROCEDURE WAS INVOKED
2121  005440                          1$:
2122                                  .SBTTL  TYPE ROUTINE
2123
2124                                  ;;**************************************************************************
2125                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2126                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2127                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
2128                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
2129                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2130                                  ;*
2131                                  ;*CALL:
2132                                  ;*1) USING A TRAP INSTRUCTION
2133                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2134                                  ;*OR
2135                                  ;*      TYPE
2136                                  ;*      MESADR
2137                                  ;*
2138
2139  005440  105737  001177          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
2140  005444  100002                          BPL     1$              ;;BR IF YES
```

```
 2141  005446  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
 2142  005450  000430                          BR      3$              ;;LEAVE
 2143  005452  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
 2144  005454  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 2145  005460  122737  000001  001254          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 2146  005466  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 2147  005470  132737  000100  001255          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 2148  005476  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 2149  005500  010037  005510                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 2150  005504  004737  006012                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 2151  005510  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
 2152  005512  132737  000040  001255  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 2153  005520  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
 2154  005522  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 2155  005524  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 2156  005526  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 2157  005530  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
 2158  005532  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 2159  005536  000002                          RTI                     ;;RETURN
 2160  005540  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 2161  005544  001430                          BEQ     8$
 2162  005546  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 2163  005552  001006                          BNE     5$
 2164  005554  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
 2165  005556  104402                          TYPE                    ;;TYPE A CR AND LF
 2166  005560  001231                          $CRLF
 2167  005562  105037  006000                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 2168  005566  000755                          BR      2$              ;;GET NEXT CHARACTER
 2169  005570  004737  005652          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 2170  005574  123726  001176          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 2171  005600  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
 2172  005602  013746  001174                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 2173                                          ;;AND THE NULL CHAR.
 2174  005606  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 2175  005612  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 2176  005614  004737  005652                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
 2177  005620  105337  006000                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 2178  005624  000770                          BR      7$              ;;LOOP
 2179
 2180                                  ;HORIZONTAL TAB PROCESSOR
 2181
 2182  005626  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 2183  005632  004737  005652          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 2184  005636  132737  000007  006000          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 2185  005644  001372                          BNE     9$              ;;TAB STOP
 2186  005646  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
 2187  005650  000724                          BR      2$              ;;GET NEXT CHARACTER
 2188  005652                          $TYPEC:
 2189  005652  105777  173306                  TSTB    @#TKS           ;;CHAR IN KYBD BUFFER?                  ;MJD001
 2190  005656  100022                          BPL     10$             ;;BR IF NOT                             ;MJD001
 2191  005660  017746  173302                  MOV     @#KB,-(SP)      ;;GET CHAR                              ;MJD001
 2192  005664  042716  177600                  BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS                 ;MJD001
 2193  005670  122716  000023                  CMPB    #XOFF,(SP)      ;;WAS CHAR XOFF                         ;MJD001
 2194  005674  001012                          BNE     102$            ;;BR IF NOT                             ;MJD001
 2195  005676                          101$:                                                                   ;MJD001
 2196  005676  105777  173262                  TSTB    @#TKS           ;;WAIT FOR CHAR                         ;MJD001
```

```
2197  005702  100375                          BPL     101$                                    ;MJD001
2198  005704  117716  173256                  MOVB    @#TKB,(SP)      ;;GET CHAR              ;MJD001
2199  005710  042716  177600                  BIC     #177600,(SP)    ;;STRIP IT              ;MJD001
2200  005714  122716  000021                  CMPB    #$XON,(SP)      ;;WAS IT XON?           ;MJD001
2201  005720  001366                          BNE     101$            ;;BR IF NOT             ;MJD001
2202  005722                          102$:                                                   ;MJD001
2203  005722  005726                          TST     (SP)+           ;;FIX STACK             ;MJD001
2204  005724                          10$:                                                    ;MJD001
2205  005724  105777  173240                  TSTB    @#TPS           ;;WAIT UNTIL PRINTER IS READY
2206  005730  100375                          BPL     10$                                     ;MJD001
2207  005732  126627  000002  000021          CMPB    2(SP),#$XON     ;;IS CHARACTER A RANDOM XON?  ;RAN001
2208  005740  001420                          BEQ     $TYPEX          ;;BRANCH IF YES         ;RAN001
2209  005742  116677  000002  173222          MOVB    2(SP),@#TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2210  005750  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2211  005756  001003                          BNE     1$              ;;BRANCH IF NO
2212  005760  105037  006000                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2213  005764  000406                          BR      $TYPEX          ;;EXIT
2214  005766  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2215  005774  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
2216  005776  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
2217  006000  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
2218  006002  000207                  $TYPEX: RTS     PC
2219
2220                                  .SBTTL  APT COMMUNICATIONS ROUTINE
2221
2222                                  ;;**************************************************************
2223  006004  112737  000001  006250  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
2224  006012  112737  000001  006246  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
2225  006020  000403                          BR      $ATYC
2226  006022  112737  000001  006250  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
2227  006030                          $ATYC:
2228  006030  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2229  006032  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2230  006034  105737  006246                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
2231  006040  001450                          BEQ     5$              ;;IF NOT:  BR
2232  006042  122737  000001  001254          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
2233  006050  001031                          BNE     3$              ;;IF NOT:  BR
2234  006052  132737  000100  001255          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
2235  006060  001425                          BEQ     3$              ;;IF NOT:  BR
2236  006062  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
2237  006066  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
2238  006074  005737  001234          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
2239  006100  001375                          BNE     1$              ;;IF NOT:  WAIT
2240  006102  010037  001250                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
2241  006106  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
2242  006110  001376                          BNE     2$
2243  006112  163700  001250                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
2244  006116  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
2245  006120  010037  001252                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
2246  006124  012737  000004  001234          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
2247  006132  000413                          BR      5$
2248  006134  017637  000004  006160  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
2249  006142  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
2250  006150  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
2251  006154  004737  005440                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
2252  006160  000000                  4$:     .WORD   0
```

```
2253  006162                          5$:
2254  006162  105737  006250        10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
2255  006166  001416                        BEQ     12$             ;;IF NOT:  BR
2256  006170  005737  001254                TST     $ENV            ;;RUNNING UNDER APT?
2257  006174  001413                        BEQ     12$             ;;IF NOT:  BR
2258  006176  005737  001234        11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
2259  006202  001375                        BNE     11$             ;;IF NOT:  WAIT
2260  006204  017637  000004  001236        MOV     @4(SP),$FATAL   ;;GET ERROR #
2261  006212  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
2262  006220  005237  001234                INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
2263  006224  105037  006250        12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
2264  006230  105037  006247                CLRB    $LFLG           ;;CLEAR LOG FLAG
2265  006234  105037  006246                CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
2266  006240  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
2267  006242  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
2268  006244  000207                        RTS     PC              ;;RETURN
2269  006246     000  $MFLG:   .BYTE   0               ;;MESSG. FLAG
2270  006247     000  $LFLG:   .BYTE   0               ;;LOG FLAG
2271  006250     000  $FFLG:   .BYTE   0               ;;FATAL FLAG
2272          006252          .EVEN
2273          000200  APTSIZE=200
2274          000001  APTENV=001
2275          000100  APTSPOOL=100
2276          000040  APTCSUP=040
2277
2278                          ;STRING INPUT ROUTINE
2279                          ;-------------------------
2280
2281  006252  010346  .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
2282  006254  010446          MOV     R4,-(SP)        ;SAVE R4 ON STACK
2283  006256  017637  000004  006274  MOV     @4(SP),.MSG     ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
2284  006264  062766  000002  000004  ADD     #2,4(SP)        ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
2285  006272  104402  .INST1: TYPE                    ;PRINT THE MESSAGE
2286  006274  000000  .MSG:   0                       ;MESSAGE IS POINTED TO FROM HERE
2287  006276  012704  011272          MOV     #INBUF,R4       ;POINT R4 TO THE INPUT BUFFER
2288  006302  012703  000007          MOV     #7,R3           ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
2289  006306  105777  172652  1$:     TSTB    @#TKS           ;HAS A CHARACTER BEEN RECEIVED?
2290  006312  100375                  BPL     1$              ;IF NO, KEEP WAITING FOR IT
2291  006314  117714  172646          MOVB    @#TKB,(R4)      ;IF YES, SAVE IT IN THE INPUT BUFFER
2292  006320  142714  000200          BICB    #200,(R4)       ;KEEP ONLY THE 7-BIT ASCII INFORMATION
2293  006324  122714  000023          CMPB    #@#XOFF,(R4)    ;IS IT A XOFF?                  ;:DSH-BHL
2294  006330  001014                  BNE     83$             ;BR IF NOT                      ;:DSH
2295  006332  105777  172626  101$:   TSTB    @#TKS           ;WAIT FOR A CHARACTER           ;:DSH-BHL
2296  006336  100375                  BPL     101$                                            ;:DSH-BHL
2297  006340  117714  172622          MOVB    @#TKB,(R4)      ;GET CHARACTER                  ;:DSH-BHL
2298  006344  142714  000200          BICB    #200,(R4)       ;STRIP DOWN CHARACTER           ;:DSH-BHL
2299  006350  122714  000021          CMPB    #@#XON,(R4)     ;WAIT FOR A XON?                ;:DSH-BHL
2300  006354  001366                  BNE     101$            ;GET NEXT CHAR IF NOT           ;:DSH-BHL
2301  006356  105724                  TSTB    (R4)+           ;POP STACK                      ;:DSH
2302  006360  000752                  BR      1$              ;WAIT FOR A CHAR                ;:DSH
2303  006362  122714  000021  83$:     CMPB    #@#XON,(R4)     ;IS IT A RANDOM XON             ;:DSH-BHL
2304  006366  001002                  BNE     102$            ;BR IF NO                       ;:DSH-BHL
2305  006370  105724                  TSTB    (R4)+           ;ELSE, POP STACK                ;:DSH-BHL
2306  006372  000745                  BR      1$              ;GO GET NEXT CHAR               ;:DSH-BHL
2307  006374  122724  000015  102$:   CMPB    #15,(R4)+       ;IS IT <CR> ?
2308  006400  001417                  BEQ     INSTR2          ;IF SO, TERMINATE THE INPUT SEQUENCE
```

```
2309  006402  105777  172562          2$:     TSTB    @#TPS           ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
2310  006406  100375                          BPL     2$              ;IF WE CAN'T, WAIT UNTIL WE CAN
2311  006410  017777  172552  172554          MOV     @#TKB,@#TPB     ;ECHO THE CHARACTER BACK
2312  006416  005303                          DEC     R3              ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
2313  006420  001332                          BNE     1$              ;IF WE DON'T HAVE 7, GO GET SOME MORE
2314  006422  012604                          MOV     (SP)+,R4        ;IF WE HAVE 7, RESTORE R4
2315  006424  012603                          MOV     (SP)+,R3        ;RESTORE R3
2316  006426  010346                  .INSTE: MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
2317  006430  010446                          MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
2318  006432  104402  001230                  TYPE    .$QUES          ;PRINT A QUESTION MARK... WHAT S GOING ON?
2319  006436  000715                          BR      .INST1          ;GO PRINT THE MESSAGE AGAIN
2320  006440  012604                  INSTR2: MOV     (SP)+,R4        ;RESTORE R4
2321  006442  012603                          MOV     (SP)+,R3        ;RESTORE R3
2322  006444  000002                          RTI                     ;RETURN TO THE MAIN PROCEDURE
2323
2324                                          ;CONVERT ASCII STRING TO OCTAL
2325                                          ;----------------------------
2326
2327  006446  010546                  .PARAM: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
2328  006450  010446                          MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
2329  006452  016605  000004                  MOV     4(SP),R5        ;GET THE SETUP INFORMATION POINTER
2330  006456  012537  006636                  MOV     (R5)+,LOLIM     ;SET THE LOW LIMIT FOR THE INPUT
2331  006462  012537  006640                  MOV     (R5)+,HILIM     ;SET THE HIGH LIMIT FOR THE INPUT
2332  006466  012537  006642                  MOV     (R5)+,DEVADR    ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
2333  006472  112537  006644                  MOVB    (R5)+,LOBITS    ;GET THE MASK OF THE INCORRECT BITS
2334  006476  112537  006645                  MOVB    (R5)+,ADRCNT    ;GET THE COUNT OF ITEMS TO BE STORED
2335  006502  010566  000004                  MOV     R5,4(SP)        ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
2336  006506  005005                  PARAM1: CLR     R5              ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
2337  006510  012704  011272                  MOV     #INBUF,R4       ;POINT TO THE INPUT BUFFER
2338  006514  122714  000015                  CMPB    #15,(R4)        ;IS THIS CHARACTER A CARRIAGE RETURN?
2339  006520  001420                          BEQ     PARERR          ;IF SO, PRINT THE MESSAGE AGAIN
2340  006522  121427  000060          1$:     CMPB    (R4),#60        ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
2341  006526  002415                          BLT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
2342  006530  121427  000067                  CMPB    (R4),#67        ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
2343  006534  003012                          BGT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
2344  006536  142714  000060                  BICB    #60,(R4)        ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
2345  006542  152405                          BISB    (R4)+,R5        ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
2346  006544  122714  000015                  CMPB    #15,(R4)        ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
2347  006550  001406                          BEQ     LIMITS          ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
2348  006552  006305                          ASL     R5              ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
2349  006554  006305                          ASL     R5              ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
2350  006556  006305                          ASL     R5              ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
2351                                                                  ;NEXT THREE BITS
2352  006560  000760                          BR      1$              ;GO GET THE NEXT CHARACTER
2353  006562  104404                  PARERR: INSTER                  ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
2354  006564  000750                          BR      PARAM1          ;TRY GETTING THE PARAMETERS AGAIN
2355
2356                                          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2357                                          ;--- -------------------------------
2358
2359  006566  020537  006640          LIMITS: CMP     R5,HILIM        ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
2360  006572  101373                          BHI     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
2361  006574  020537  006636                  CMP     R5,LOLIM        ;IS THE RESULT LOWER THAN ALLOWED?
2362  006600  103770                          BLO     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
2363  006602  133705  006644                  BITB    LOBITS,R5       ;ARE ANY INCORRECT BITS SET IN THE RESULT?
2364  006606  001365                          BNE     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
```

```
                                        ;STORE NUMBER AT SPECIFIED ADDRESS
                                        ;
006610  013704  006642                  MOV     DEVADR,R4       ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
006614  010524                  1$:     MOV     R5,(R4)+        ;STORE THE RESULT
006616  062705  000002                  ADD     #2,R5           ;CALCULATE THE NEXT DATUM
006622  105337  006645                  DECB    ADRCNT          ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
006626  001372                          BNE     1$              ;IF NOT, GO STORE THE NEXT DATUM
006630  012604                          MOV     (SP)+,R4        ;RESTORE R4
006632  012605                          MOV     (SP)+,R5        ;RESTORE R5
006634  000002                          RTI                     ;RETURN TO THE MAIN PROGRAM
                                        ;
006636  000000                  LOLIM:  0                       ;LOWEST ACCEPTABLE VALUE
006640  000000                  HILIM:  0                       ;HIGHEST ACCEPTABLE
006642  000000                  DEVADR: 0                       ; LOCATION WHERE RESULT WILL BE STORED
006644     000                  LOBITS: .BYTE   0               ;INCORRECT BITS MASK
006645     000                  ADRCNT: .BYTE   0               ;COUNT OF ITEMS TO BE STORED
                                        ;
                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5
                                        ;-------------------------------------
                                        ;
006646  016637  000004  001402  .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)
                                        ;
                                        ;SAVE R0-R5
                                        ;
006654  010537  001214          SV05:   MOV     R5,$REG5        ;SAVE R5
006660  010437  001212                  MOV     R4,$REG4        ;SAVE R4
006664  010337  001210                  MOV     R3,$REG3        ;SAVE R3
006670  010237  001206                  MOV     R2,$REG2        ;SAVE R2
006674  010137  001204                  MOV     R1,$REG1        ;SAVE R1
006700  010037  001202                  MOV     R0,$REG0        ;SAVE R0
006704  000002                          RTI                     ;LEAVE.
                                        ;
                                        ;RESTORE R0-R5
                                        ;
006706  013700  001202          .RES05: MOV     $REG0,R0        ;RESTORE R0
006712  013701  001204                  MOV     $REG1,R1        ;RESTORE R1
006716  013702  001206                  MOV     $REG2,R2        ;RESTORE R2
006722  013703  001210                  MOV     $REG3,R3        ;RESTORE R3
006726  013704  001212                  MOV     $REG4,R4        ;RESTORE R4
006732  013705  001214                  MOV     $REG5,R5        ;RESTORE R5
006736  000002                          RTI                     ;LEAVE
                                        ;
                                        ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
                                        ;------------------------------------------------------
                                        ;
006740  104402  001231          .CONVR: TYPE    ,$CRLF          ;PRINT A CARRIAGE RETURN
006744  010046                  .CNVRT: MOV     R0,-(SP)        ;SAVE R0
006746  010146                          MOV     R1,-(SP)        ;SAVE R1
006750  010346                          MOV     R3,-(SP)        ;SAVE R3
006752  010446                          MOV     R4,-(SP)        ;SAVE R4
006754  010546                          MOV     R5,-(SP)        ;SAVE R5
006756  017601  000012                  MOV     @12(SP),R1      ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
006762  062766  000002  000012          ADD     #2,12(SP)       ;POINT TO WHERE MAIN PROGRAM WILL RESUME
006770  012137  007114                  MOV     (R1)+,WRDCNT    ;GET NUMBER OF WORDS TO BE PRINTED
006774  112105                  1$:     MOVB    (R1)+,R5        ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
```

```
2421  006776  112100                  MOVB    (R1)+,R0        ;GET THE NUMBER OF SPACES TO PRINT
2422  007000  013104                  MOV     @(R1)+,R4       ;COPY THE WORD TO BE CONVERTED
2423  007002  110537  007116          MOVB    R5,CHRCNT       ;COPY THE CHARACTER COUNT
2424  007006  010403          3$:     MOV     R4,R3           ;COPY THE ARGUMENT WORD AGAIN
2425  007010  042703  177770          BIC     #^C<7>,R3       ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
2426  007014  062703  000060          ADD     #060,R3         ;MAKE AN ASCII CHARACTER OUT OF THEM
2427  007020  110346                  MOVB    R3,-(SP)        ;SAVE THAT CHARACTER
2428  007022  006004                  ROR     R4              ;MOVE THE NEXT THREE BITS INTO PLACE
2429  007024  006204                  ASR     R4              ;MOVE THEM AGAIN
2430  007026  006204                  ASR     R4              ;AND FINALLY A THIRD TIME
2431  007030  005305                  DEC     R5              ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
2432                                                          ;BUILT?
2433  007032  001365                  BNE     3$              ;IF NO, GO BUILD THE NEXT ONE.
2434  007034  012703  011376          MOV     #MDATA,R3       ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
2435  007040  112623          4$:     MOVB    (SP)+,(R3)+     ;STORE THE CHARACTER, STARTING WITH THE MOST
2436  007042  105337  007116          DECB    CHRCNT          ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
2437  007046  001374                  BNE     4$              ;IF NO, GO TRANSFER ANOTHER
2438  007050  105700                  TSTB    R0              ;ARE ANY SPACES TO BE PRINTED?
2439  007052  001404                  BEQ     6$              ;IF NO, DON'T SET UP ANY
2440  007054  112723  000040  5$:     MOVB    #040,(R3)+      ;ADD A SPACE TO THE OUTPUT BUFFER
2441  007060  105300                  DECB    R0              ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
2442  007062  001374                  BNE     5$              ;IF YES, GO ADD ANOTHER SPACE
2443  007064  105013          6$:     CLRB    (R3)            ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
2444  007066  104402  011376          TYPE    ,MDATA          ;PRINT THE STRING WE JUST BUILT
2445  007072  005337  007114          DEC     WRDCNT          ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
2446  007076  001336                  BNE     1$              ;IF YES, GO CONVERT THEM
2447  007100  012605                  MOV     (SP)+,R5        ;RESTORE R5
2448  007102  012604                  MOV     (SP)+,R4        ;RESTORE R4
2449  007104  012603                  MOV     (SP)+,R3        ;RESTORE R3
2450  007106  012601                  MOV     (SP)+,R1        ;RESTORE R1
2451  007110  012600                  MOV     (SP)+,R0        ;RESTORE R0
2452  007112  000002                  RTI                     ;RETURN TO THE MAIN PROGRAM
2453  007114  000000          WRDCNT: 0
2454  007116     000          CHRCNT: .BYTE                   ;NUMBER OF CHARACTERS TO PRINT
2455  007117     000          SPACNT: .BYTE   0               ;NUMBER OF SPACES TO PRINT
2456
2457  007120  000000          BINWRD: 0
2458
2459
2460                                  ;TRAP DISPATCH SERVICE
2461                                  ;ARGUMENT OF TRAP IS EXTRACTED
2462                                  ;AND USED AS OFFSET TO OBTAIN POINTER
2463                                  ;TO SELECTED SUBROUTINE
2464
2465  007122  010046          .TRPSR: MOV     R0,-(SP)        ;SAVE R0. USE R0 TO FIND TRAP ROUTINE
2466  007124  016600  000002          MOV     2(SP),R0        ;GET TRAP ADDRESS
2467  007130  005740                  TST     -(R0)           ;GET TRAP
2468  007132  111000                  MOVB    (R0),R0         ;GET RIGHT BYTE OF TRAP(TRAP OFFSET)
2469  007134  006300                  ASL     R0              ;POSITION OFFSET FOR TABLE INDEXING
2470  007136  016000  002002          MOV     .TRPTAB(R0),R0  ;PLACE INDEXED ADDRESS OF TABLE IN R0
2471  007142  000200                  RTS     R0              ;TRANSFER TO THAT ADDRESS AND RESTORE OLD R0
2472
2473                                  ;DEVICE CLEAR ROUTINE
2474                                  ;ISSUE A DEVICE CLEAR
2475                                  ;---------------- --
2476  007144                  .DEVICE.CLR:
```

```
2477  007144  052777  000020  172670          BIS     #DCLR,@DZCSR    ;SET DCLR
2478  007152  032777  000020  172662  1$:     BIT     #DCLR,@DZCSR    ;DID IT CLEAR?
2479  007160  001374                          BNE     1$              ;BR IF NO
2480  007162  000002                          RTI                     ;EXIT ROUTINE
2481
2482                                          ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
2483                                          ;-----------------------------------------------------------
2484  007164  104413                  .DCLASM:DEVICE.CLR              ;ISSUE A DEVICE CLEAR
2485  007166  153777  001417  172646          BISB    MNTFLG,@DZCSR   ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
2486  007174  000002                          RTI                     ;RETURN TO CALLING ROUTINE
2487
2488  007176                          .DELAY:
2489  007176  010046                          MOV     R0,-(SP)        ;SAVE R0
2490  007200  013700  007214                  MOV     DLYCNT,R0       ;SET COUNT
2491  007204  005300                  1$:     DEC     R0              ;DELAY
2492  007206  001376                          BNE     1$              ;
2493  007210  012600                          MOV     (SP)+,R0        ;RESTORE R0
2494  007212  000002                          RTI                     ;LEAVE ROUTINE
2495  007214  000001                  DLYCNT: .WORD    1              ;PATCHABLE LOC FOR MORE TIME
2496
2497                                          ;ADVANCE TO NEXT TEST HANDLER
2498                                          ;----------------------------
2499
2500  007216  013716  001360          .ADVANCE:MOV    NEXT,(SP)       ;CRUNCH STACK WITH ADDRESS OF SCOPE CALL
2501  007222  005037  001362                  CLR     LOCK            ;RESET TIGHT LOOP ADDRESS
2502  007226  000002                          RTI                     ;CHECK TO SEE IF OLD TEST GETS REPEATED
2503
2504                                          ;ERROR HANDLER
2505                                          ;-------------
2506
2507  007230  004737  007652          $ERROR: JSR     PC,SERV.G       ;FIND OUT IF <^G> WAS HIT
2508  007234  032777  010000  171716          BIT     #SW12,@SWR      ;BELL ON ERROR?
2509  007242  001406                          BEQ     XBX             ;BR IF NO BELL
2510  007244  105777  171720                  TSTB    @#TPS           ;TTY READY.
2511  007250  100003                          BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
2512  007252  112777  000207  171712          MOVB    #207,@#TPB      ;PUSH A BELL AT THE TTY.
2513  007260  032777  020000  171672  XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
2514  007266  001113                          BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
2515  007270  021637  001136                  CMP     (SP),$ERRPC     ;WAS THIS ERROR FOUND LAST TIME?
2516  007274  001404                          BEQ     1$              ;BR IF YES
2517  007276  011637  001136                  MOV     (SP),$ERRPC     ;RECORD BEING HERE
2518  007302  105037  001123                  CLRB    $ERFLG          ;PREPARE HEADER
2519  007306  104407                  1$:     SAV05                   ;SAVE ALL PROC REGISTERS
2520  007310  011605                          MOV     (SP),R5         ;GET THE PC OF ERROR
2521  007312  162705  000002                  SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
2522  007316  011504                          MOV     (R5),R4         ;GET ERROR INSTRUCTION
2523  007320  110437  001134                  MOVB    R4,$ITEMB       ;COPY TEST NUMBER FOR APT HANDLING
2524  007324  006304                          ASL     R4              ;MULT BY TWO
2525  007326  061504                          ADD     (R5),R4         ;DOUBLE IT
2526  007330  006304                          ASL     R4              ;MULT AGAIN
2527  007332  042704  177001                  BIC     #177001,R4      ;CLEAR JUNK
2528  007336  062704  030312                  ADD     #.ERRTAB,R4     ;GET POINTER
2529  007342  012437  007466                  MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
2530  007346  012437  007500                  MOV     (R4)+,DATAHD    ;GET DATA HEADRER
2531  007352  011437  007512                  MOV     (R4),DATABP     ;GET DATA TABLE
2532  007356  105737  001123                  TSTB    $ERFLG          ;TYPE HEADER
```

```
2533  007362  001403                        BEQ     TYPMSG          ;BR IF YES
2534  007364  005737  007512                TST     DATABP          ;DOES DATA TABLE EXIST?
2535  007370  001044                        BNE     TYPDAT          ;BR IF YES.
2536  007372  104402  001231        TYPMSG: TYPE    .$CRLF          ;TYPE A CARRIAGE RETURN
2537  007376  104402  001231                TYPE    .$CRLF          ;AND TYPE ANOTHER
2538  007402  005737  001362                TST     LOCK
2539  007406  001402                        BEQ     1$
2540  007410  104402  010766                TYPE    .MASTEK
2541  007414  104402  010754        1$:     TYPE    .MTSTN
2542  007420  104412  007644                CNVRT   .XTSTN          ;SHOW IT
2543  007424  104402  011043                TYPE    .MERRPC         ;TYPE PC.
2544  007430  104412  007636                CNVRT   .ERTABO         ;SHOW IT
2545  007434  104402  010716                TYPE    .MCSRX
2546  007440  104412  005072                CNVRT   .XCSR
2547  007444  104402  001231                TYPE    .$CRLF          ;GIVE A CR/LF
2548  007450  112737  177777  001123        MOVB    #-1,$ERFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
2549  007456  005737  007466                TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
2550  007462  001402                        BEQ     WTBS.FM         ;BR IF NO.
2551  007464  104402                        TYPE                    ;TYPE
2552  007466  000000        ERRMSG: 0                               ;    ERROR MESSAGE
2553  007470                WTBS.FM:                                ;
2554  007470  005737  007500                TST     DATAHD          ;DATA HEADER?
2555  007474  001402                        BEQ     TYPDAT          ;BR IF NO
2556  007476  104402                        TYPE                    ;TYPE
2557  007500  000000        DATAHD: 0                               ;    DATA HEADER
2558  007502  005737  007512        TYPDAT: TST     DATABP          ;DATA TABLE?
2559  007506  001402                        BEQ     RESREG          ;BR IF NO.
2560  007510  104411                        CONVRT                  ;SHOW
2561  007512  000000        DATABP: 0                               ;    DATA TABLE
2562  007514  104410        RESREG: RCSO5                           ;RESTORE PROC REGISTERS
2563  007516  122737  000001  001254 HALTS:  CMPB    @APTENV,$ENV    ;IS APT RUNNING?
2564  007524  001007                        BNE     2$              ;SKIP APT CALL IF NOT
2565  007526  113737  001134  007540        MOVB    $ITEMB,7$       ;COPY ERROR NUMBER
2566  007534  004737  006022                JSR     PC,$ATY4        ;CALL APT SERVICE
2567  007540  000000        7$:     .WORD   0                       ;ERROR NUMBER STUCK HERE
2568  007542  000777        8$:     BR      8$                      ;LOCK UP HERE
2569  007544  022737  005056  000042 2$:     CMP     @#ENDAD,#042    ;CHECK TO SEE IF IN ACT-11 MODE
2570  007552  001403                        BEQ     1$              ;IF SO, HANDLE ACCORDINGLY
2571  007554  005777  171400                TST     @SWR            ;HALT ON ERROR?
2572  007560  100004                        BPL     EXITER          ;BR IF NO HALT ON ERROR
2573  007562  016677  000002  171372 1$:.    MOV     2(SP),@DISPLAY  ;SHOW ERROR PC IN DATA DISPLAY
2574  007570  000000                        HALT                    ;HALT
2575  007572  005237  001132        EXITER: INC     $ERTTL          ;UPDATE ERROR COUNT
2576  007576  032777  000400  171354        BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
2577  007604  001007                        BNE     1$              ;BR IF YES
2578  007606  032777  002000  171344        BIT     #SW10,@SWR      ;GOTO NEXT TEST?
2579  007614  001407                        BEQ     2$              ;BR IF NO
2580  007616  013737  001360  001126        MOV     NEXT,$LPADR     ;SET FOR NEXT TEST
2581  007624  012706  001120        1$:     MOV     #STACK,SP       ;RESET SP
2582  007630  000177  171272                JMP     @$LPADR         ;GOTO SPECIFIED TEST
2583  007634  000002        2$:     RTI                             ;RETURN
2584  007636  000001        ERTABO: 1
2585  007640     006     002                .BYTE   6,2
2586  007642  001402                        SAVPC
2587  007644  000001        XTSTN:  1
2588  007646     002     002                .BYTE   2,2
```

```
                                        $TSTNM
2589  007650  001122
2590  007652  022737  177570  001160  SERV.G: CMP     #177570,SWR     ;IS THE SWITCH REGISTER HARDWIRED?
2591  007660  001002                          BNE     99$             ;IF SO, IGNORE ^G
2592  007662  000137  010270                  JMP     6$                                              ;;DSM-BML
2593  007666  017746  171274          99$:    MOV     @$TKB,-(SP)     ;OTHERWISE, GET THE LAST CHARACTER TYPED
2594  007672  042716  177600                  BIC     #177600,(SP)    ;STRIP CHAR                     ;;DSM-BML
2595  007676  122716  000023                  CMPB    #$XOFF,(SP)     ;IS IT A XOFF                   ;;DSM-BML
2596  007702  001012                          BNE     102$            ;BR IF NO                       ;;DSM-BML
2597
2598  007704  105777  171254          101$:   TSTB    @$TKS           ;WAIT FOR A CHAR                ;;DSM-BML
2599  007710  100375                          BPL     101$                                            ;;DSM-BML
2600  007712  117716  171250                  MOVB    @$TKB,(SP)      ;GET THE CHAR                   ;;DSM-BML
2601  007716  042716  177600                  BIC     #177600,(SP)    ;STRIP CHAR                     ;;DSM-BML
2602  007722  122716  000021                  CMPB    #$XON,(SP)      ;IS IT A XON                    ;;DSM-BML
2603  007726  001366                          BNE     101$            ;BR IF NO                       ;;DSM-BML
2604
2605  007730  122716  000021          102$:   CMPB    #$XON,(SP)      ;IS IT RANDOM XON ?             ;;DSM
2606  007734  001002                          BNE     7$              ;BR IF NOT                      ;;DSM
2607  007736  005726                          TST     (SP)+           ;POP STACK                      ;;DSM
2608  007740  000553                          BR      6$              ;IGNORE XON CHAR                ;;DSM
2609
2610  007742  122726  000007          7$:     CMPB    #7,(SP)+        ;IS IT ^G?
2611  007746  001150                          BNE     6$              ;IF NOT, IGNORE INPUT
2612  007750  032777  004000  171206          BIT     #4000,@$TKS     ;RX BUSY?
2613  007756  001335                          BNE     SERV.G          ;BR IF YES
2614  007760  017737  171174  010312          MOV     @SWR,90$        ;SAVE (SWR).
2615  007766  013777  010312  171164  1$:     MOV     90$,@SWR        ;
2616  007774  104402  010272                  TYPE    .89$            ;
2617  010000  104412  010304                  CNVRT   .88$            ;
2618  010004  104402  010314                  TYPE    .91$            ;
2619  010010  105777  171150          9$:     TSTB    @$TKS           ;WAIT FOR DONE.
2620  010014  100375                          BPL     .-4
2621  010016  017746  171144                  MOV     @$TKB,-(SP)     ;
2622  010022  042716  177600                  BIC     #177600,(SP)    ;STRIP CHAR                     ;;DSM-BML
2623  010026  122716  000023                  CMPB    #$XOFF,(SP)     ;IS IT A XOFF                   ;;DSM-BML
2624  010032  001012                          BNE     112$            ;BR IF NO                       ;;DSM-BML
2625
2626  010034  105777  171124          111$:   TSTB    @$TKS           ;WAIT FOR A CHAR                ;;DSM-BML
2627  010040  100375                          BPL     111$                                            ;;DSM-BML
2628  010042  117716  171120                  MOVB    @$TKB,(SP)      ;GET THE CHAR                   ;;DSM-BML
2629  010046  042716  177600                  BIC     #177600,(SP)    ;STRIP CHAR                     ;;DSM-BML
2630  010052  122716  000021                  CMPB    #$XON,(SP)      ;IS IT A XON                    ;;DSM-BML
2631  010056  001366                          BNE     111$            ;BR IF NO                       ;;DSM-BML
2632
2633  010060  122716  000021          112$:   CMPB    #$XON,(SP)      ;IS IT RANDOM XON ?             ;;DSM
2634  010064  001002                          BNE     8$              ;BR IF NOT                      ;;DSM
2635  010066  005726                          TST     (SP)+           ;POP STACK                      ;;DSM
2636  010070  000747                          BR      9$              ;IGNORE XON CHAR                ;;DSM
2637
2638  010072  122726  000015          8$:     CMPB    #15,(SP)+       ;
2639  010076  001472                          BEQ     5$              ;
2640  010100  005077  171054                  CLR     @SWR            ;
2641  010104  105777  171060          2$:     TSTB    @$TPS           ;
2642  010110  100375                          BPL     .-4
2643  010112  016677  177776  171052          MOV     -2(SP),@$TPB    ;
2644  010120  000241                          CLC                     ;
```

```
2645  010122  006177  171032                ROL     @SWR            ;
2646  010126  006177  171026                ROL     @SWR            ;
2647  010132  006177  171022                ROL     @SWR            ;
2648  010136  103713                        BCS     1$              ;ERROR
2649  010140  026627  177776  000060        CMP     -2(SP),#60      ;
2650  010146  002707                        BLT     1$              ;
2651  010150  026627  177776  000067        CMP     -2(SP),#67      ;
2652  010156  003303                        BGT     1$              ;
2653  010160  042766  177770  177776        BIC     #^C<7>,-2(SP)   ;
2654  010166  056677  177776  170764        BIS     -2(SP),@SWR     ;
2655  010174  105777  170764          12$:  TSTB    @#TKS           ;
2656  010200  100375                        BPL     .-4             ;
2657  010202  017746  170760                MOV     @#TKB,-(SP)
2658  010206  042716  177600                BIC     #177600,(SP)    ;STRIP CHAR                     ;;DSH-BHL
2659  010212  122716  000023                CMPB    #^XOFF,(SP)     ;IS IT A XOFF                   ;;DSH-BHL
2660  010216  001012                        BNE     122$            ;BR IF NO                       ;;DSH-BHL
2661
2662  010220  105777  170740          121$: TSTB    @#TKS           ;WAIT FOR A CHAR                ;;DSH-BHL
2663  010224  100375                        BPL     121$                                            ;;DSH-BHL
2664  010226  117716  170734                MOVB    @#TKB,(SP)      ;GET THE CHAR                   ;;DSH-BHL
2665  010232  042716  177600                BIC     #177600,(SP)    ;STRIP CHAR                     ;;DSH-BHL
2666  010236  122716  000021                CMPB    #^XON,(SP)      ;IS IT A XON                    ;;DSH-BHL
2667  010242  001366                        BNE     121$            ;BR IF NO                       ;;DSH-BHL
2668
2669  010244  122716  000021          122$: CMPB    #^XON,(SP)      ;IS IT RANDOM XON ?             ;;DSH
2670  010250  001002                        BNE     10$             ;BR IF NOT                      ;;DSH
2671  010252  005726                        TST     (SP)+           ;POP STACK                      ;;DSH
2672  010254  000747                        BR      12$             ;IGNORE XON CHAR                ;;DSH
2673
2674  010256  122726  000015          10$:  CMPB    #15,(SP)+
2675  010262  001310                        BNE     2$              ;
2676  010264  104402  001231          5$:   TYPE    ,$CRLF          ;
2677  010270  000207                  6$:   RTS     PC
2678
2679  010272  020200  051450  051127  89$:   .ASCIZ  <200>? (SWR)=/?
2680  010300  036451  000057
2681                                  .EVEN
2682  010304  000001                  88$:   1
2683  010306      006     000                .BYTE   6,0
2684  010310  010312                        90$
2685  010312  000000                  90$:   .WORD   0
2686  010314  036457  000057          91$:   .ASCIZ  ?/=/?
2687                                  .EVEN
2688                                  .SBTTL  POWER DOWN AND UP ROUTINES
2689
2690                                  ;;**************************************************************
2691                                  ;POWER DOWN ROUTINE
2692  010320  012737  010464  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
2693  010326  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
2694  010334  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2695  010336  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2696  010340  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2697  010342  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
2698  010344  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
2699  010346  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
2700  010350  017746  170604                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
```

```
2701  010354  010637  010470                  MOV     SP,$SAVR6       ;;SAVE SP
2702  010360  012737  010372  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
2703  010366  000000                          HALT
2704  010370  000776                          BR      .-2             ;;HANG UP
2705
2706                                  ;;*****************************************************************
2707                                  ;POWER UP ROUTINE
2708  010372  012737  010464  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
2709  010400  013706  010470                  MOV     $SAVR6,SP       ;;GET SP
2710  010404  005037  010470                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
2711  010410  005237  010470          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
2712  010414  001375                          BNE     1$              ;;OF  WORD
2713  010416  012677  170536                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
2714  010422  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
2715  010424  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
2716  010426  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
2717  010430  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
2718  010432  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
2719  010434  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
2720  010436  012737  010320  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
2721  010444  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
2722  010452  104402                          TYPE                    ;;REPORT THE POWER FAILURE
2723  010454  010472          $PWRMG: .WORD   MPFAIL                  ;;POWER FAIL MESSAGE POINTER
2724  010456  012716                          MOV     (PC)+,(SP)      ;;RESTART AT RESTART
2725  010460  012106          $PWRAD: .WORD   RESTART                 ;;RESTART ADDRESS
2726  010462  000002                          RTI
2727  010464  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
2728  010466  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
2729  010470  000000          $SAVR6: 0                               ;;PUT THE SP HERE
2730  010472  050200  051127  043040  MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /
 (2)  010535     200  047105  020104  MEPASS: .ASCIZ  <200>/END PASS CZDZA-H  /
 (2)  010560  051200  047125  044516  MR:     .ASCIZ  <200>/RUNNING   /
 (2)  010574  050200  047522  051107  MERR2:  .ASCIZ  <200>/PROGRAM INDICATES NO DEVICES PRESENT./
 (2)  010643     200  047111  052523  MERR3:  .ASCIZ  <200>/INSUFFICIENT DATA!/
 (2)  010667     200  047514  045503  MLOCK:  .ASCIZ  <200>/LOCK ON SELECTED TEST/
 (2)  010716  051503  035122  000040  MCSRX:  .ASCIZ   /CSR: /
 (2)  010724  042526  035103  000040  MVECX:  .ASCIZ  /VEC: /
 (2)  010732  040520  051523  051505  MPASSX: .ASCIZ  /PASSES: /
 (2)  010743     105  051122  051117  MERRX:  .ASCIZ  /ERRORS: /
 (2)  010754  042524  052123  047040  MTSTN:  .ASCIZ  /TEST NO: /
 (2)  010766  020052     000         MASTEK: .ASCIZ  /* /
 (2)  010771     200  042523  020124  MNEW:   .ASCIZ  <200>/SET SWITCH REG TO DZ11'S DESIRED ACTIVE./
 (2)  011043     120  035103  000040  MERRPC: .ASCIZ  /PC: /
 (2)  011050  046600  050101  047440  XHEAD:  .ASCIZ  <200>/MAP OF DZ11 STATUS/<200>
 (2)  011075     200  046111  042514  MBADLN: .ASCIZ  <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
 (2)          011140                  .EVEN
 (2)  011140  000002                  XSTATQ: 2
2731  011142     006     003                  .BYTE   6,3
2732  011144  001220                          $TMP1
2733  011146     006     002                  .BYTE   6,2
2734  011150  001222                          $TMP2
2735                                  .EVEN
2736                                                  ; -$SETFLG-------------------------------
2737                                  ;THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
2738                                  ;-----------------------------------------    ----
2739                                  ;E=EXTERNAL LOOP BACK
```

```
                                        ;I=INTERNAL LOOP BACK
                                        ;S=STAGGERED LOOP BACK
011152  017605  000000          .SETFLG:MOV     @(SP),R5        ;PICK UP ADDRESS OF TAG
011156  042737  000040  011272          BIC     #40,INBUF       ;STRIP LOWER CASE
011164  122737  000105  011272          CMPB    #'E,INBUF       ;IS IT EXTERNAL LOOP BACK ?
011172  001005                          BNE     4$              ;NO
011174  013715  011264                  MOV     1$,(R5)         ;YES STORE INFO
011200  105037  001417                  CLRB    MNTFLG          ;SET MAINT BIT =0
011204  000422                          BR      7$              ;GET OUT
011206  122737  000111  011272  4$:     CMPB    #'I,INBUF       ;IS IT INTERNAL LOOP BACK ?
011214  001006                          BNE     5$              ;NO
011216  013715  011266                  MOV     2$,(R5)         ;YES STORE INFO
011222  112737  000010  001417          MOVB    #MAINT,MNTFLG   ;SET UP THE MAINTENANCE FLAG LOADER
011230  000410                          BR      7$              ;GET OUT
011232  122737  000123  011272  5$:     CMPB    #'S,INBUF       ;IS IT STAGGERED LOOP BACK ?
011240  001007                          BNE     6$              ;WHAT ?
011242  013715  011270                  MOV     3$,(R5)         ;YES STORE INFO
011246  105037  001417                  CLRB    MNTFLG          ;ZERO BITS
011252  062716  000002          7$:     ADD     #2,(SP)         ;POP AROUND
011256  000002                          RTI
011260  104404                  6$:     INSTER                  ;RETRY
011262  000733                          BR      .SETFLG         ;DITTO
011264  000200                  1$:     .WORD   200             ;EXTERNAL = E
011266  000000                  2$:     .WORD   0               ;INTERNAL = I
011270  100000                  3$:     .WORD   100000          ;STAGGERED = S
                                ; -- END 0 MACRO ----------------
                                        ; -$BUFFER-------------------------------

                                        ;BUFFERS FOR INPUT-OUTPUT

011272  000000                  INBUF:  0
        011334                  .=.+40
011334  000000                  TEMP:   0
        011376                  .=.+40
011376  000000                  MDATA:  0
        011440                  .=.+40

011440  011637  011536          SET.PS: MOV     (SP),3$
011444  162737  000002  011536          SUB     #2,3$
011452  017737  000060  011540          MOV     @3$,4$
011460  022737  106427  011540          CMP     #106427,4$
011466  001003                          BNE     1$
011470  011637  011536                  MOV     (SP),3$
011474  000412                          BR      2$
011476  022737  106437  011540  1$:     CMP     #106437,4$
011504  001401                          BEQ     .+4
011506  000000                          HALT            ;RESERVED INSTRUCTION NOT "MTPS"
011510  011637  011536                  MOV     (SP),3$
011514  017737  000016  011536          MOV     @3$,3$
011522  062716  000002          2$:     ADD     #2,(SP)
011526  017766  000004  000002          MOV     @3$,2(SP)
011534  000002                          RTI
011536  000000                  3$:     0
011540  000000                  4$:     0
```

```
                                        ; -- END O MACRO ==================
                                                                        ; -$CYCLE-------------------------------


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

011542  005737  001404          CYCLE:  TST     DZACTV          ;ARE ANY DZ11'S TO BE TESTED?
011546  001004                          BNE     1$              ;BR IF OK.
011550  104402  010574                  TYPE    ,MERR2          ;NO DZ11'S SELECTED!!
011554  000000                          HALT                    ;STOP THE SHOW.
011556  000776                          BR      .-2             ;DISQUALIFY CONT. SW.
011560  013737  005374  001226  1$:     MOV     $MXCNT,$TIMES   ;RESTORE THE NUMBER OF ITERATIONS TO MAKE
011566  033737  001406  001404          BIT     RUN,DZACTV      ;IS THIS ONE "ACTIVE"
011574  001020                          BNE     2$              ;BR IF GOOD ONE FOUND.
011576  000241                          CLC
011600  006137  001406                  ROL     RUN             ;UPDATE POINTER
011604  005537  001406                  ADC     RUN             ;CATCH CARRY FROM RUN
011610  062737  000014  001412          ADD     #14,ACTIVE      ;UPDATE ADDRESS POINTER.
011616  022737  002000  001412          CMP     #DZ.END,ACTIVE  ;HAVE WE PASSED THE END OF THE MAP?
011624  001355                          BNE     1$              ;IF NO, KEEP GOING; NOT ALL TESTED FOR.
011626  012737  001500  001412          MOV     #DZ.MAP,ACTIVE  ;RESET ADDRESS POINTER.
011634  000751                          BR      1$              ;KEEP LOOKING FOR ACTIVE DZ11
011636  000241                  2$:     CLC
011640  006137  001406                  ROL     RUN             ;UPDATE POINTER.
011644  005537  001406                  ADC     RUN             ;CATCH CARRY.
011650  013700  001412                  MOV     ACTIVE,R0       ;GET ADDRESS POINTER.
011654  062737  000014  001412          ADD     #14,ACTIVE      ;UPDATE.
011662  022737  002000  001412          CMP     #DZ.END,ACTIVE
                                                                ;ALL DONE?
011670  001003                          BNE     3$              ;BR IF NO.
011672  012737  001500  001412          MOV     #DZ.MAP,ACTIVE  ;RESTORE POINTER.
011700  012037  001310          3$:     MOV     (R0)+,$BASE     ;LOAD SYSTEM CTRL. REG
011704  012037  002072                  MOV     (R0)+,DZRIV     ;LOAD VECTOR
011710  012037  030306                  MOV     (R0)+,DZPRT     ;LOAD PRIORITY
011714  113737  030307  001414          MOVB    DZPRT+1,EIAFLG  ;EIA OR 20MA.
011722  042737  100000  030306          BIC     #BIT15,DZPRT    ;CLEAR FLAG
011730  012037  001364                  MOV     (R0)+,LINE      ;SET UP LINE DZ LINES ACTIVE
011734  012037  001366                  MOV     (R0)+,PAR       ;SET UP PARAMETERIZATION
011740  012037  001370                  MOV     (R0)+,MODE      ;SET UP MAINTENANCE MODE
011744  004737  030100                  JSR     PC,DZLEV        ;SET UP
011750  005737  000042                  TST     @#42            ;ARE WE UNDER MONITOR CONTROL?
011754  001051                          BNE     4$              ;IF YES, SKIP THIS SETUP
011756  032777  000002  167174          BIT     #SW01,@SWR      ;IF SW01=1, GET STARTING TEST #
011764  001445                          BEQ     4$              ;BR IF NO TEST IS TO BE INPUTTED
011766  104402  001231          7$:     TYPE    ,$CRLF
                                                                ; -$GETPAR--------------------------------
011772  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
011774  010754                          MTSTN                   ;POINTER TO MESSAGE TO BE PRINTED
011776  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
012000  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
012002  001000                          1000                    ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
```

```
2850  012004  001122                        $TSTNM                  ;POINTER TO MAP LOCATION TO BE FILLED
2851  012006     000                        .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
2852  012007     001                        .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
2853                                ; -- END 0 MACRO -----------------
2854  012010  012700  013046               MOV     #TST1,R0
2855  012014  022710  000004        5$:     CMP     #4,(R0)
2856  012020  001020                        BNE     6$
2857  012022  022760  012737  000002        CMP     #12737,2(R0)
2858  012030  001014                        BNE     6$
2859  012032  023760  001122  000004        CMP     $TSTNM,4(R0)    ;IS THIS THE TEST ?
2860  012040  001010                        BNE     6$              ;IF NOT, DON'T PROCESS NUMBER
2861  012042  010037  001126               MOV     R0,$LPADR       ;SAVE PC
2862  012046  062737  000002  001126        ADD     #2,$LPADR       ;POP OVER SCOPE
2863  012054  104402  001231               TYPE    ,$CRLF
2864  012060  000412                        BR      8$
2865  012062  005720                6$:     TST     (R0)+
2866  012064  020027  023740               CMP     R0,#TLAST+10
2867  012070  001351                        BNE     5$
2868  012072  104402  001230               TYPE    ,$QUES
2869  012076  000733                        BR      7$
2870  012100  012737  013046  001126  4$:     MOV     #TST1,$LPADR    ;PREPARE TEST ADDRESS
2871  012106                         8$:
2872  012106  000177  167014        RESTART:JMP     @$LPADR         ;GO START TESTING.***WARNING!****
2873                                                                 ;THIS JUMP IS USED BY POWER UP ROUTINE!!!!
2874
```

```
2875                                        ;-ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.
2876                                        ; IF BIT7 IN THE ENVIRONMENT MODE ($ENVM) BYTE IS SET,
2877                                        ; THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.
2878
2879  012112  012700  001500    SETAPT:  MOV     #DZ.MAP,R0      ;POINT TO THE DEVICE MAP TABLE
2880  012116  013701  001310             MOV     $BASE,R1        ;BUILD DEVICE ADDRESSES IN R1
2881  012122  013702  001304             MOV     $VECT1,R2       ;BUILD DEVICE VECTORS IN R2
2882  012126  042702  177007             BIC     #^C<770>,R2     ;STRIP AWAY OTHER INFORMATION
2883
2884  012132  113703  001305             MOVB    $VECT1+1,R3     ;LOAD THE INTERRUPT PRIORITY FROM R3
2885  012136  106003                      RORB    R3              ;ALIGN THE NUMBER
2886  012140  106003                      RORB    R3              ;ALIGN THE NUMBER
2887  012142  106003                      RORB    R3              ;ALIGN THE NUMBER
2888  012144  106003                      RORB    R3              ;ALIGN THE NUMBER
2889  012146  106003                      RORB    R3              ;ALIGN THE NUMBER
2890  012150  042703  177770             BIC     #^C<7>,R3       ;REMOVE ALL BUT BUS LEVEL NUMBER
2891  012154  012704  001320             MOV     #$DDW0,R4       ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
2892  012160  013705  001312             MOV     $DEVM,R5        ;GET THE MAP OF ACTIVE DEVICES
2893  012164  010537  001404             MOV     R5,DZACTV       ;SAVE THE BIT MAP
2894  012170  006005          1$:        ROR     R5              ;GET A DEVICE SELECTION BIT
2895  012172  103407                      BCS     3$              ;IF IT IS SELECTED, GO SET UP A MAP
2896  012174  001425                      BEQ     5$              ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
2897  012176  005724                      TST     (R4)+           ;POINT TO NEXT DEVICE DESCRIPTOR
2898  012200  062701  000010    2$:      ADD     #10,R1          ;SET UP THE NEXT ADDRESS
2899  012204  062702  000010             ADD     #10,R2          ;SET UP THE NEXT VECTOR GROUP
2900  012210  000767                      BR      1$              ;GO SEE IF MORE DEVICES REMAIN
2901  012212  010120          3$:        MOV     R1,(R0)+        ;LOAD DEVICE ADDRESS
2902  012214  010220                      MOV     R2,(R0)+        ;LOAD THE VECTOR ADDRESS
2903  012216  010320                      MOV     R3,(R0)+        ;LOAD THE INTERRUPT PRIORITY LEVEL
2904  012220  013720  001314             MOV     $COM1,(R0)+     ;GET THE NUMBER OF LINES IN OPERATION
2905  012224  012420                      MOV     (P:)+,(R0)+     ;LOAD DEVICE PARAMETERS
2906  012226  100006                      BPL     4$              ;IF 20MA MODE SELECTED, SET IT UP
2907  012230  052760  100000  177772     BIS     #100000,-6(R0)  ;SET THE 20MA FLAG IN DZLVN
2908  012236  042760  100000  177776     BIC     #100000,-2(R0)  ;CLEAR THE FLAG IN DZPARN
2909  012244  005020          4$:        CLR     (R0)+           ;DEFAULT OPERATION TO INTERNAL MAINTENANCE MODE
2910  012246  000754                      BR      2$              ;GO BUILD THE NEXT ADDRESS
2911  012250  012710  177777    5$:      MOV     #-1,(R0)        ;TERMINATE THE DEVICE MAP
2912  012254  012737  001256  001160     MOV     #$SWREG,SWR     ;SET TO SOFTWARE APT SWITCH REGISTER
2913  012262  000207                      RTS     PC              ;RETURN TO PRINT STATUS TABLE
2914
2915
2916                                        ;*ROUTINE USED TO "AUTO SIZE" THE DZ11
2917                                        ;*CSR AND VECTOR.
2918                                        ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
2919                                        ;*      ADDRESS RANGE (160000:163700)
2920                                        ;*      AND THE VECTOR MAY BE ANY WHERE IN THE
2921                                        ;*      FLOATING VECTOR RANGE (300:770)
2922                                        ;*
2923
2924  012264          AUTO.SIZE:
2925  012264  000005                      RESET                   ;INSURE A BUS INIT.
2926  012266  105337  001415             DECB    INIFLG          ;SHOW THAT I WAS HERE
2927  012272  012702  001500    CSRMAP:  MOV     #DZ.MAP,R2      ;LOAD MAP POINTER.
2928  012276  012703  001320             MOV     #$DDW0,R3       ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
2929  012302  005022          1$:        CLR     (R2)+           ;ZERO ENTIRE MAP
2930  012304  022702  002000             CMP     #DZ.END,R2      ;ALL DONE?
```

```
2931  012310  001374                          BNE     1$              ;BR IF NO
2932  012312  105037  001410                  CLRB    DZNUM           ;SET OCTAL NUMBER OF DZ11'S TO 0
2933  012316  012702  001500                  MOV     #DZ.MAP,R2
2934  012322  012701  160000                  MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
2935  012326  012737  012646  000004          MOV     #6$,@4          ;SET FOR NON-EXISTENT DEVICE TIME OUT
2936  012334  052711  000040          2$:     BIS     #BIT5,(R1)      ;TRY TO SET MASTER SCAN ENABLE
2937  012340  052761  000200  000004          BIS     #BIT7,4(R1)     ;TRY TO TRANSMIT ON LINE 7
2938  012346  005000                          CLR     R0              ;USE R0 AS A COUNTER
2939  012350  005711                  7$:     TST     (R1)            ;HAS TRANSMITTER READY COME UP?
2940  012352  100403                          BMI     8$              ;IF SO, GO GET A FINAL CHECK
2941  012354  005300                          DEC     R0              ;REDUCE COUNT. TIME UP?
2942  012356  001374                          BNE     7$              ;IF NOT, KEEP WAITING
2943  012360  000463                          BR      3$              ;ASSUME IT'S NOT A DZ11
2944  012362  032761  000200  000004  8$:     BIT     #BIT7,4(R1)     ;IS LINE 7 ENABLE STILL SET? IT SHOULD BE
2945  012370  001457                          BEQ     3$              ;IF IT'S NOT, ASSUME IT'S NOT A DZ11
2946  012372  032711  000040                  BIT     #BIT5,(R1)      ;IS MASTER SCAN ENABLE STILL SET?
2947  012376  001454                          BEQ     3$              ;IF NOT, ASSUME IT'S NOT A DZ11
2948  012400  005000                          CLR     R0
2949  012402  052711  000020                  BIS     #20,(R1)        ;SET DEVICE CLEAR
2950  012406  032711  000020                  BIT     #20,(R1)        ;SHOULD STAY SET FOR A WHILE IF DZ
2951  012412  001446                          BEQ     3$              ;BR IF NOT DZ11
2952  012414  032711  000020                  BIT     #20,(R1)        ;WAIT FOR BIT TO CLEAR
2953  012420  001404                          BEQ     .+12            ;BR WHEN CLEARED
2954  012422  104414                          DELAY
2955  012424  005200                          INC     R0
2956  012426  001372                          BNE     .-12
2957  012430  000437                          BR      3$              ;BIT NOT CLEARED! MUST NOT BE DZ11
2958  012432  005011                          CLR     (R1)            ;GET RID OF MASTER SCAN ENABLE
2959  012434  005061  000004                  CLR     4(R1)           ;GET RID OF LINE 7 ENABLE
2960                          ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DZ11 CSR ADDRESS.
2961  012440  010122                          MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
2962  012442  005722                          TST     (R2)+           ;POP OVER VECTOR STORE AREA
2963  012444  012722  000005                  MOV     #5,(R2)+        ;SET THE DEFAULT BUS LEVEL
2964  012450  052761  177400  000004          BIS     #177400,4(R1)   ;TRY TO SET ALL DTR BITS
2965  012456  032761  177400  000004          BIT     #177400,4(R1)   ;IF ANY SET ASSUME EIA BOARD
2966  012464  001003                          BNE     9$              ;IF NONE SET ASSUME BOARD IS
2967  012466  052762  100000  177776          BIS     #BIT15,-2(R2)   ;20 MA, SET 20 MA FLAG
2968  012474  012722  000377          9$:     MOV     #377,(R2)+      ;SET THE DEFAULT LINE SELECTION PARAMETER
2969  012500  012712  017070                  MOV     #17070,(R2)     ;SET THE DEFAULT PARAMETERS
2970  012504  012223                          MOV     (R2)+,(R3)+     ;COPY PARAMETERS INTO ETABLE DESCRIPTOR
2971  012506  005022                          CLR     (R2)+           ;SET THE DEFAULT MODE OF OPERATION
2972  012510  012712  177777                  MOV     #-1,(R2)        ;TERMINATE LIST
2973  012514  105237  001410                  INCB    DZNUM           ;UPDATE DEVICE COUNTER
2974  012520  122737  000020  001410          CMPB    #20,DZNUM       ;ARE MAX. NO. OF DEV FOUND?
2975  012526  001405                          BEQ     100$            ;YES DON'T LOOK FOR ANY MORE.
2976  012530  062701  000010          3$:     ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
2977  012534  022701  163700                  CMP     #163700,R1
2978  012540  001275                          BNE     2$              ;BR IF MORE ADDRESS TO CHECK.
2979  012542                          100$:
2980  012542  105737  001410                  TSTB    DZNUM           ;WERE ANY DZ11'S FOUND AT ALL?
2981  012546  001432                          BEQ     5$              ;ERROR AUTO SIZER FOUND NO DZ11'S IN THIS SYS.
2982  012550  113701  001410                  MOVB    DZNUM,R1
2983  012554  110137  001411                  MOVB    R1,SAVNUM       ;SAVE NUMBER OF DEVICES
2984  012560  012737  000001  001404          MOV     #1,DZACTV
2985  012566  005301                  4$:     DEC     R1
2986  012570  001404                          BEQ     98$
```

```
 2987  012572  000261                          SEC
 2988  012574  006137  001404                  ROL     DZACTV
 2989  012600  000772                          BR      4$
 2990  012602  013737  001500  001310  98$:    MOV     DZCR0,#BASE     ;POINT TO THE ADDRESS OF FIRST DEVICE
 2991  012610  013737  001512  001314          MOV     MANTO,#CDW1     ;INDICATE TO ETABLE WHAT MODE IS BEING USED
 2992  012616  012737  000006  000004  99$:    MOV     #6,#04          ;RESTORE TRAP VECTOR
 2993  012624  013737  001404  001312          MOV     DZACTV,#DEVM    ;SAVE ACTIVE REGISTER
 2994  012632  000410                          BR      VECMAP          ;GO FIND THE VECTOR NOW.
 2995  012634  104402  010574          5$:     TYPE    ,MERR2          ;NOTIFY OPR THAT NO DZ11'S FOUND.
 2996  012640  005000                          CLR     R0              ;MAKE DATA DISPLAY ZERO
 2997  012642  000000                          HALT                    ;STOP THE SHOW
 2998  012644  000776                          BR      .-2             ;DISABLE CONT. SW.
 2999  012646  012716  012530          6$:     MOV     #3$,(SP)        ;ENTERED BY NON-EXISTENT TIME-OUT
 3000  012652  000002                          RTI                     ;RETURN TO MAINSTREAM
 3001
 3002  012654  012737  000340  000022  VECMAP: MOV     #340,#022       ;SET IOT TRAP PRIORITY TO 7
 3003  012662  012737  013000  000020          MOV     #4$,#020        ;SET IOT TRAP VECTOR
 3004  012670  012702  001500                  MOV     #DZ.MAP,R2      ;SET SOFTWARE POINTER
 3005  012674  012700  000300                  MOV     #300,R0         ;FLOATING VECTORS START HERE.
 3006  012700  012701  000302                  MOV     #302,R1         ;PC OF IOT INSTR.
 3007  012704  010120                  1$:     MOV     R1,(R0)+        ;START FILLING VECTOR AREA
 3008  012706  012721  000004                  MOV     #4,(R1)+        ;WITH .+2; IOT
 3009  012712  022021                          CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
 3010  012714  020127  001000                  CMP     R1,#1000        ;HAS THE VECTOR AREA BEEN EXCEEDED?
 3011  012720  101771                          BLOS    1$              ;BR IF MORE TO FILL
 3012  012722  013704  001404                  MOV     DZACTV,R4       ;STORE TEMPORARILY
 3013  012726  000241                  2$:     CLC
 3014  012730  006004                          ROR     R4              ;BRING OUT A BIT
 3015  012732  103036                          BCC     5$              ;BR IF ALL DONE
 3016  012734  106427  000000                  MTPS    #0              ;ZERO CPU PRIO
 3017  012740  012772  040040  000000          MOV     #BIT14+BIT5,@(R2)
 3018  012746  011201                          MOV     (R2),R1         ;GET CSR
 3019  012750  112761  000200  000004          MOVB    #BIT7,4(R1)     ;SET THE TCR BIT!
 3020                                          ;ATTEMPT TO FORCE AN INTERRUPT
 3021  012756  005200                          INC     R0              ;STALL
 3022  012760  001376                          BNE     .-2             ;       FOR TIME TO INTERRUPT
 3023  012762  012762  000300  000002          MOV     #300,2(R2)      ;NO INTERRUPT ASSUME 300 AND FIX DZ11 LATER
 3024  012770  000005                  3$:     RESET                   ;INIT
 3025  012772  062702  000014                  ADD     #14,R2          ;POP SOFTWARE POINTER
 3026  012776  000753                          BR      2$              ;KEEP GOING
 3027  013000  011662  000002          4$:     MOV     (SP),2(R2)      ;GET VECTOR ADDRESS
 3028  013004  162762  000010  000002          SUB     #10,2(R2)       ;POINT BACK TO THE CORRECT VECTOR
 3029  013012  042762  000007  000002          BIC     #7,2(R2)        ;CLEAR JUNK
 3030  013020  022626                          POP2SP                  ;POP IOT JUNK OFF STACK
 3031  013022  012716  012770                  MOV     #3$,(SP)        ;SET FOR RETURN
 3032  013026  000002                          RTI
 3033  013030  013737  001502  001304  5$:     MOV     DZVC0,#VECT1    ;COPY VECTOR OF FIRST DEVICE INTO ETABLE
 3034  013036  012737  005122  000020          MOV     #.SCOPE,IOTVEC  ;RESTORE THE SCOPE TRAP
 3035  013044  000207                          RTS     PC              ;ALL DONE WITH "AUTO SIZING"
 3036
 3037                                  ; -- END O MACRO ------------------
```

```
3038
3039                                                    ; -$UNIBUS-------------------------------
3040                                                    ; -$XZ-----------------------------------
3041                            ;************************ TEST 1 ********************************
3042                            ;*THIS TEST PROVES THE SLAVE SYNC RESPONSE
3043                            ;*DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
3044                            ;*      DZCSR, DZRBUF, DZTCR, DZMSR
3045                                                    ; -$XZ-----------------------------------
3046                        ;:*  TEST 1
3047                        ;;*************************************************************
3048  013046  000004       TST1:   SCOPE
3049  013050  012737  000001  001122                   MOV     #1,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
3050                        ; -- END 0 MACRO ------------------
3051  013056  012737  013236  001360                   MOV     #TST2,NEXT      ;POINT TO THE START OF THE NEXT TEST
3052  013064  012737  013224  000004                   MOV     #5$,4           ;SET TRAP VECTOR
3053  013072  012737  000340  000006                   MOV     #PR7,6          ;SET PRIORITY TO LEVEL 7
3054  013100  012737  013106  001362                   MOV     #1$,LOCK        ;SET RETURN IF SW09-11
3055  013106  013700  002042       1$:         MOV     DZCSR,R0        ;SET ADDRESS TO TEST
3056  013112  011001                           MOV     (R0),R1         ;READ THE ADDRESS
3057  013114  000240                           NOP                     ;WASTE TIME
3058  013116  005010                           CLR     (R0)            ;WRITE THE ADDRESS
3059  013120  000240                           NOP                     ;WASTE TIME
3060  013122  012737  013130  001362                   MOV     #2$,LOCK        ;SET RETURN ADDRESS FOR SW09
3061  013130  013700  002046       2$:         MOV     DZRBUF,R0       ;SET ADDRESS TO TEST
3062  013134  011001                           MOV     (R0),R1         ;READ THE ADDRESS
3063  013136  000240                           NOP                     ;
3064  013140  005010                           CLR     (R0)            ;WRITE THE ADDRESS
3065  013142  000240                           NOP                     ;WASTE TIME
3066  013144  012737  013152  001362                   MOV     #3$,LOCK        ;SET RETURN ADDRESS FOR SW09
3067  013152  013700  002056       3$:         MOV     DZTCR,R0        ;SET ADDRESS TO TEST
3068  013156  011001                           MOV     (R0),R1         ;READ THE ADDRESS
3069  013160  000240                           NOP
3070  013162  005010                           CLR     (R0)            ;WRITE THE ADDRESS
3071  013164  000240                           NOP
3072  013166  012737  013174  001362                   MOV     #4$,LOCK        ;SET RETURN ADDRESS
3073  013174  013700  002062       4$:         MOV     DZMSR,R0        ;SET ADDRESS TO TEST
3074  013200  011001                           MOV     (R0),R1         ;READ FROM ADDRESS
3075  013202  000240                           NOP
3076  013204  005010                           CLR     (R0)            ;WRITE THE ADDRESS
3077  013206  000240                           NOP                     ;
3078  013210  012737  000006  000004                   MOV     #6,4            ;SET TRAP CATCHER BACK TO NORMAL
3079  013216  005037  000006                   CLR     6               ;
3080  013222  104400                           ADVANCE                 ;SCOPE THIS TEST
3081  013224  011601       5$:         MOV     (SP),R1         ;SAVE PC OF TRAP
3082  013226  022626                           CMP     (SP)+,(SP)+     ;POP TRAP OFF STACK
3083  013230  104001                           ERROR   1               ;*NO SLAVE SYNC RESPONSE.
3084  013232  104401                           SCOP1                   ;SW09=1?
3085  013234  000111                           JMP     (R1)            ;RTI
3086                        ; -- END 0 MACRO ------------------
3087                                                    ; -$XZ-----------------------------------
3088                            ;************************ TEST 2 ********************************
3089                            ;*THIS TEST PROVES THAT BIT "DCLR"
3090                            ;*CAN BE SET AND THAT IT WILL CLEAR
3091                            ;*BY ITSELF AFTER A PERIOD OF TIME.
3092                                                    ; -$XZ-----------------------------------
3093                        ;:*  TEST 2
```

```
3094                                    ;;*****************************************************************
3095  013236  000004                    TST2:   SCOPE
3096  013240  012737  000002  001122            MOV     #2,@#TSTNM      ;LOAD THE NUMBER OF THIS TEST
3097                                    ; -- END 0 MACRO ------------------
3098  013246  012737  013322  001360            MOV     #TST3,NEXT      ;POINT TO THE START OF THE NEXT TEST
3099  013254  013700  002042                    MOV     DZCSR,R0        ;SET POINTER
3100  013260  012705  000020                    MOV     #DCLR,R5        ;SET DCLR
3101  013264  010510                            MOV     R5,(R0)         ;WRITE DCLR INTO DZCSR
3102  013266  011004                            MOV     (R0),R4         ;READ BACK DZCSR
3103  013270  020504                            CMP     R5,R4           ;DZCSR OK?
3104  013272  001401                            BEQ     1$              ;IF IT IS SET SKIP THE ERROR CALL
3105  013274  104002                            ERROR   2               ;*DCLR SHOULD BE SET..MOMENTARILY
3106                                        ;NOW LETS WATCH IT DISAPPEAR
3107  013276  005002                    1$:     CLR     R2              ;SET COUNTER TO 0
3108  013300  005005                            CLR     R5              ;SET EXPECTED TO 0
3109  013302  005003                            CLR     R3              ;DUAL LOOP COUNTER
3110  013304  011004                    2$:     MOV     (R0),R4         ;IS DCLR CLEAR?
3111  013306  001405                            BEQ     3$              ;IF YES , GO TO THE NEXT TEST
3112  013310  005203                            INC     R3              ;IF NO,COUNT 1 OF 65535 TICKS
3113                                                                    ;THE WORD CREATED BY THE IMMEDIATE 0 WILL BE
3114                                                                    ;THE COUNTER
3115  013312  001374                            BNE     2$              ;HAS THE TIME EXPIRED? IF NO, GO TEST BIT AGAIN
3116  013314  005302                            DEC     R2              ;HAS THE TOTAL TIME EXPIRED?
3117  013316  001372                            BNE     2$              ;IF NO, CHECK THE BIT AGAIN
3118  013320  104002                            ERROR   2               ;*DCLR FAILED TO CLEAR
3119  013322                            3$:
3120                                                                    ; -@#MMRM-----------------------------------
3121                                                                    ; -@XZ--------------------------------
3122                                        ;********************** TEST 3 ******************************
3123                                        ;*TEST TO VERIFY THAT BIT "MAINT" CAN
3124                                        ;*BE SET. THEN VERIFY THAT BIT "MAINT" CAN
3125                                        ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
3126                                        ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
3127                                        ;*CLEARED BY A "DEVICE CLEAR"
3128                                                                    ; -@XZ----------------------------
3129                                    ;:*  TEST 3
3130                                    ;;*****************************************************************
3131  013322  000004                    TST3:   SCOPE
3132  013324  012737  000003  001122            MOV     #3,@#TSTNM      ;LOAD THE NUMBER OF THIS TEST
3133                                    ; -- END 0 MACRO ------------------
3134  013332  012737  013414  001360            MOV     #TST4,NEXT      ;POINT TO THE START OF THE NEXT TEST
3135  013340  013700  002042                    MOV     DZCSR,R0        ;GET BASE ADDRESS
3136  013344  012705  000010                    MOV     #MAINT,R5       ;SET BIT
3137  013350  010510                            MOV     R5,(R0)         ;SET SET IN DEVICE
3138  013352  011004                            MOV     (R0),R4         ;READ THE BIT FROM DEVICE
3139  013354  020504                            CMP     R5,R4           ;WAS BIT SET?
3140  013356  001401                            BEQ     1$              ;BR IF YES
3141  013360  104002                            ERROR   2               ;*BIT R/W FAILURE
3142  013362  040510                    1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
3143  013364  011004                            MOV     (R0),R4         ;READ DEVICE
3144  013366  001404                            BEQ     2$              ;BR IF BITS WERE CLEARED.
3145  013370  010546                            MOV     R5,-(SP)        ;SAVE THE BIT
3146  013372  005005                            CLR     R5              ;SET EXPECTED RESULTS TO 0
3147  013374  104002                            ERROR   2               ;*BIT FAILED TO CLEAR
3148  013376  012605                            MOV     (SP)+,R5        ;RESTORE THE BIT.
3149  013400  010510                    2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
```

```
3150  013402  104413                        DEVICE.CLR              ;ISSUE DEVICE CLEAR
3151  013404  011004                        MOV     (R0),R4         ;READ THE BIT.
3152  013406  001402                        BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
3153  013410  005005                        CLR     R5              ;SET EXPECTED TO ZERO
3154  013412  104002                        ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
3155  013414                          3$:
3156                                   ; -- END 0 MACRO ******************
3157                                   ; -- END 0 MACRO ******************
3158                                                                  ; -@MRRW-----------------------------------
3159                                                                  ; -@XZ-------------------------------
3160                                           ;************************ TEST 4 ******************************
3161                                           ;*TEST TO VERIFY THAT BIT "RSENAB" CAN
3162                                           ;*BE SET. THEN VERIFY THAT BIT "RSENAB" CAN
3163                                           ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
3164                                           ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
3165                                           ;*CLEARED BY A "DEVICE CLEAR"
3166                                                                  ; -@XZ-------------------------------
3167                                   ;:*  TEST 4
3168                                   ;;*****************************************************************
3169  013414  000004                  TST4:   SCOPE
3170  013416  012737  000004  001122          MOV     #4,@#TSTNM      ;LOAD THE NUMBER OF THIS TEST
3171                                   ; -- END 0 MACRO ******************
3172  013424  012737  013506  001360          MOV     #TST5,@#NEXT    ;POINT TO THE START OF THE NEXT TEST
3173  013432  013700  002042                  MOV     DZCSR,R0        ;GET BASE ADDRESS
3174  013436  012705  000040                  MOV     #RSENAB,R5      ;SET BIT
3175  013442  010510                          MOV     R5,(R0)         ;SET SET IN DEVICE
3176  013444  011004                          MOV     (R0),R4         ;READ THE BIT FROM DEVICE
3177  013446  020504                          CMP     R5,R4           ;WAS BIT SET?
3178  013450  001401                          BEQ     1$              ;BR IF YES
3179  013452  104002                          ERROR   2               ;*BIT R/W FAILURE
3180  013454  040510                  1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
3181  013456  011004                          MOV     (R0),R4         ;READ DEVICE
3182  013460  001404                          BEQ     2$              ;BR IF BITS WERE CLEARED.
3183  013462  010546                          MOV     R5,-(SP)        ;SAVE THE BIT
3184  013464  005005                          CLR     R5              ;SET EXPECTED RESULTS TO 0
3185  013466  104002                          ERROR   2               ;*BIT FAILED TO CLEAR
3186  013470  012605                          MOV     (SP)+,R5        ;RESTORE THE BIT.
3187  013472  010510                  2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
3188  013474  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR
3189  013476  011004                          MOV     (R0),R4         ;READ THE BIT.
3190  013500  001402                          BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
3191  013502  005005                          CLR     R5              ;SET EXPECTED TO ZERO
3192  013504  104002                          ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
3193  013506                          3$:
3194                                   ; -- END 0 MACRO ******************
3195                                   ; -- END 0 MACRO ******************
3196                                                                  ; -@MRRW-----------------------------------
3197                                                                  ; -@XZ-------------------------------
3198                                           ;************************ TEST 5 ******************************
3199                                           ;*TEST TO VERIFY THAT BIT "SILOEN" CAN
3200                                           ;*BE SET. THEN VERIFY THAT BIT "SILOEN" CAN
3201                                           ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
3202                                           ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
3203                                           ;*CLEARED BY A "DEVICE CLEAR"
3204                                                                  ; -@XZ-------------------------------
3205                                   ;:*  TEST 5
```

```
3206                                    ;;*****************************************************************
3207  013506  000004                    TST5:   SCOPE
3208  013510  012737  000005  001122            MOV     #5,@#TSTNM      ;LOAD THE NUMBER OF THIS TEST
3209                                    ; -- END 0 MACRO ******************
3210  013516  012737  013600  001360            MOV     #TST6,NEXT      ;POINT TO THE START OF THE NEXT TEST
3211  013524  013700  002042                    MOV     DZCSR,R0        ;GET BASE ADDRESS
3212  013530  012705  010000                    MOV     #SILOEN,R5      ;SET BIT
3213  013534  010510                            MOV     R5,(R0)         ;SET SET IN DEVICE
3214  013536  011004                            MOV     (R0),R4         ;READ THE BIT FROM DEVICE
3215  013540  020504                            CMP     R5,R4           ;WAS BIT SET?
3216  013542  001401                            BEQ     1$              ;BR IF YES
3217  013544  104002                            ERROR   2               ;*BIT R/W FAILURE
3218  013546  040510                    1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
3219  013550  011004                            MOV     (R0),R4         ;READ DEVICE
3220  013552  001404                            BEQ     2$              ;BR IF BITS WERE CLEARED.
3221  013554  010546                            MOV     R5,-(SP)        ;SAVE THE BIT
3222  013556  005005                            CLR     R5              ;SET EXPECTED RESULTS TO 0
3223  013560  104002                            ERROR   2               ;*BIT FAILED TO CLEAR
3224  013562  012605                            MOV     (SP)+,R5        ;RESTORE THE BIT.
3225  013564  010510                    2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
3226  013566  104413                            DEVICE.CLR              ;ISSUE DEVICE CLEAR
3227  013570  011004                            MOV     (R0),R4         ;READ THE BIT.
3228  013572  001402                            BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
3229  013574  005005                            CLR     R5              ;SET EXPECTED TO ZERO
3230  013576  104002                            ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
3231  013600                            3$:
3232                                    ; -- END 0 MACRO ******************
3233                                    ; -- END 0 MACRO ******************
3234                                                                    ; -@MRRW--------------------------------
3235                                                                    ; -@XZ----------------------------
3236                                            ;************************ TEST 6 ******************************
3237                                            ;*TEST TO VERIFY THAT BIT "RIE" CAN
3238                                            ;*BE SET. THEN VERIFY THAT BIT "RIE" CAN
3239                                            ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
3240                                            ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
3241                                            ;*CLEARED BY A "DEVICE CLEAR"
3242                                                                    ; -@XZ----------------------------
3243                                    ; *  TEST 6
3244                                    ;;*****************************************************************
3245  013600  000004                    TST6:   SCOPE
3246  013602  012737  000006  001122            MOV     #6,@#TSTNM      ;LOAD THE NUMBER OF THIS TEST
3247                                    ; -- END 0 MACRO ******************
3248  013610  012737  013672  001360            MOV     #TST7,NEXT      ;POINT TO THE START OF THE NEXT TEST
3249  013616  013700  002042                    MOV     DZCSR,R0        ;GET BASE ADDRESS
3250  013622  012705  000100                    MOV     #RIE,R5 ;SET BIT
3251  013626  010510                            MOV     R5,(R0)         ;SET SET IN DEVICE
3252  013630  011004                            MOV     (R0),R4         ;READ THE BIT FROM DEVICE
3253  013632  020504                            CMP     R5,R4           ;WAS BIT SET?
3254  013634  001401                            BEQ     1$              ;BR IF YES
3255  013636  104002                            ERROR   2               ;*BIT R/W FAILURE
3256  013640  040510                    1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
3257  013642  011004                            MOV     (R0),R4         ;READ DEVICE
3258  013644  001404                            BEQ     2$              ;BR IF BITS WERE CLEARED.
3259  013646  010546                            MOV     R5,-(SP)        ;SAVE THE BIT
3260  013650  005005                            CLR     R5              ;SET EXPECTED RESULTS TO 0
3261  013652  104002                            ERROR   2               ;*BIT FAILED TO CLEAR
```

```
3262  013654  012605                          MOV     (SP)+,R5        ;RESTORE THE BIT.
3263  013656  010510                  2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
3264  013660  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR
3265  013662  011004                          MOV     (R0),R4         ;READ THE BIT.
3266  013664  001402                          BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
3267  013666  005005                          CLR     R5              ;SET EXPECTED TO ZERO
3268  013670  104002                          ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
3269  013672                          3$:
3270                                  ; -- END 0 MACRO ------------------
3271                                  ; -- END 0 MACRO ------------------
3272                                                                  ; -$MRRW----------------------------------
3273                                                                  ; -$XZ------------------------------
3274                                          ;************************ TEST 7 ********************************
3275                                          ;*TEST TO VERIFY THAT BIT "TIE" CAN
3276                                          ;*BE SET. THEN VERIFY THAT BIT "TIE" CAN
3277                                          ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
3278                                          ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
3279                                          ;*CLEARED BY A "DEVICE CLEAR"
3280                                                                  ; -$XZ------------------------------
3281                                  ;:*  TEST 7
3282                                  ;;************************************************************
3283  013672  000004                  TST7:   SCOPE
3284  013674  012737  000007  001122          MOV     #7,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
3285                                  ; -- END 0 MACRO ------------------
3286  013702  012737  013764  001360          MOV     #TST10,NEXT     ;POINT TO THE START OF THE NEXT TEST
3287  013710  013700  002042                  MOV     DZCSR,R0        ;GET BASE ADDRESS
3288  013714  012705  040000                  MOV     #TIE,R5 ;SET BIT
3289  013720  010510                          MOV     R5,(R0)         ;SET SET IN DEVICE
3290  013722  011004                          MOV     (R0),R4         ;READ THE BIT FROM DEVICE
3291  013724  020504                          CMP     R5,R4           ;WAS BIT SET?
3292  013726  001401                          BEQ     1$              ;BR IF YES
3293  013730  104002                          ERROR   2               ;*BIT R/W FAILURE
3294  013732  040510                  1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
3295  013734  011004                          MOV     (R0),R4         ;READ DEVICE
3296  013736  001404                          BEQ     2$              ;BR IF BITS WERE CLEARED.
3297  013740  010546                          MOV     R5,-(SP)        ;SAVE THE BIT
3298  013742  005005                          CLR     R5              ;SET EXPECTED RESULTS TO 0
3299  013744  104002                          ERROR   2               ;*BIT FAILED TO CLEAR
3300  013746  012605                          MOV     (SP)+,R5        ;RESTORE THE BIT.
3301  013750  010510                  2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
3302  013752  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR
3303  013754  011004                          MOV     (R0),R4         ;READ THE BIT.
3304  013756  001402                          BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
3305  013760  005005                          CLR     R5              ;SET EXPECTED TO ZERO
3306  013762  104002                          ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
3307  013764                          3$:
3308                                  ; -- END 0 MACRO ------------------
3309                                  ; -- END 0 MACRO ------------------
3310                                                                  ; -$TCR-----------------------------------
3311                                                                  ; -$XZ------------------------------
3312                                          ;************************ TEST 10 *******************************
3313                                          ;*THIS TESTS THAT ALL OF THE FOLLOWING
3314                                          ;*BITS CAN BE: SET, CLEARED, CLEARED BY "DEVICE CLEAR "
3315                                          ;*BITS TESTED ARE:
3316                                          ;* TCR0, TCR1, TCR2, TCR3, TCR4, TCR5, TCR6, TCR7
3317                                                                  ; -$XZ------------------------------
```

```
                                        ;:*  TEST 10
                                        ;;*****************************************************************
013764  000004                          TST10:  SCOPE
013766  012737  000010  001122                  MOV     #10,@#TSTNM     ;LOAD THE NUMBER OF THIS TEST
                                        ; -- END 0 MACRO ******************
013774  012737  014122  001360                  MOV     #TST11,NEXT     ;POINT TO THE START OF THE NEXT TEST
014002  013700  002056                          MOV     DZTCR,R0        ;SET DEVICE ADDRESS
014006  012705  000001                          MOV     #TCR0,R5        ;SET EXPECTED RESULTS
014012  012737  014020  001362                  MOV     #1$,LOCK        ;SET FOR SW09
014020  010510                          1$:     MOV     R5,(R0)         ;SET THE BIT
014022  011004                                  MOV     (R0),R4         ;READ THE BIT FROM THE DEVICE
014024  042704  177400                          BIC     #^C<377>,R4     ;CLEAR HIGH BYTE
014030  020504                                  CMP     R5,R4           ;WAS BIT OK?
014032  001401                                  BEQ     2$              ;BR IF YES
014034  104002                                  ERROR   2               ;*BIT FAILED TO SET.
014036  040510                          2$:     BIC     R5,(R0)         ;CLEAR THE BIT
014040  011004                                  MOV     (R0),R4         ;READ THE REGISTER
014042  042704  177400                          BIC     #^C<377>,R4     ;CLEAR HIGH BYTE
014046  005704                                  TST     R4              ;BITS CLEAR?
014050  001404                                  BEQ     3$              ;BR IF YES
014052  010546                                  MOV     R5,-(SP)        ;SAVE GOOD RESULTS
014054  005005                                  CLR     R5              ;SET EXPECTED TO 0
014056  104002                                  ERROR   2               ;*REPORT BIT NOT CLEAR
014060  012605                                  MOV     (SP)+,R5        ;RESTORE R5
014062  010510                          3$:     MOV     R5,(R0)         ;SET THE BIT AGAIN.
014064  104413                                  DEVICE.CLR              ;ISSUE DEVICE CLEAR
014066  011004                                  MOV     (R0),R4         ;READ THE REGISTER
014070  042704  177400                          BIC     #^C<377>,R4     ;CLEAR HIGH BYTE
014074  005704                                  TST     R4              ;BITS CLEAR?
014076  001404                                  BEQ     4$              ;BR IF YES
014100  010546                                  MOV     R5,-(SP)        ;SAVE GOOD RESULTS
014102  005005                                  CLR     R5              ;SET EXPECTED TO 0
014104  104002                                  ERROR   2               ;*REPORT BIT NOT CLEAR
014106  012605                                  MOV     (SP)+,R5        ;RESTORE R5
014110  104401                          4$:     SCOP1                   ;LOCK ON BIT? SET SW09-1
014112  106305                                  ASLB    R5              ;CHANGE TO NEXT BIT
014114  001341                                  BNE     1$              ;CONTINUE TESTING
014116  005037  001362                          CLR     LOCK            ;MAKE SURE TIGHT LOOP IS CLEANED UP
                                        ; -- END 0 MACRO ******************
                                                ; -#TCR-----------------------------------------
                                                ; -#XZ-----------------------------
                                            ;************************* TEST 11 *******************************
                                            ;*THIS TESTS THAT ALL OF THE FOLLOWING
                                            ;*BITS CAN BE: SET, CLEARED, CLEARED BY "RESET INSTR *NOT* DEVICE CLEAR "
                                            ;*BITS TESTED ARE:
                                            ;* DTR0, DTR1, DTR2, DTR3, DTR4, DTR5, DTR6, DTR7
                                            ;*THIS TEST IS NOT DONE IF MODULE IS 20MA VERSION
                                                ; -#XZ-----------------------------
                                        ;:*  TEST 11
                                        ;;*****************************************************************
014122  000004                          TST11:  SCOPE
014124  012737  000011  001122                  MOV     #11,@#TSTNM     ;LOAD THE NUMBER OF THIS TEST
                                        ; -- END 0 MACRO ******************
014132  012737  014304  001360                  MOV     #TST12,NEXT     ;POINT TO THE START OF THE NEXT TEST
014140  013700  002056                          MOV     DZTCR,R0        ;SET DEVICE ADDRESS
014144  012705  000400                          MOV     #DTR0,R5        ;SET EXPECTED RESULTS
```

```
3374  014150  012737  014166  001362          MOV     #1$,LOCK        ;SET FOR SW09
3375  014156  105737  001414                  TSTB    EIAFLG          ;20MA OR EIA
3376  014162  100001                          BPL     1$              ;BR IF EIA
3377  014164  104400                          ADVANCE                 ;EXIT TEST
3378  014166  010510                  1$:     MOV     R5,(R0)         ;SET THE BIT
3379  014170  011004                          MOV     (R0),R4         ;READ THE BIT FROM THE DEVICE
3380  014172  105004                          CLRB    R4              ;CLEAR LOW BYTE
3381  014174  020504                          CMP     R5,R4           ;WAS BIT OK?
3382  014176  001401                          BEQ     2$              ;BR IF YES
3383  014200  104002                          ERROR   2               ;*BIT FAILED TO SET.
3384  014202  040510                  2$:     BIC     R5,(R0)         ;CLEAR THE BIT
3385  014204  011004                          MOV     (R0),R4         ;READ THE REGISTER
3386  014206  105004                          CLRB    R4              ;CLEAR LOW BYTE
3387  014210  005704                          TST     R4              ;BITS CLEAR?
3388  014212  001404                          BEQ     3$              ;BR IF YES
3389  014214  010546                          MOV     R5,-(SP)        ;SAVE GOOD RESULTS
3390  014216  005005                          CLR     R5              ;SET EXPECTED TO 0
3391  014220  104002                          ERROR   2               ;*REPORT BIT NOT CLEAR
3392  014222  012605                          MOV     (SP)+,R5        ;RESTORE R5
3393  014224  010510                  3$:     MOV     R5,(R0)         ;SET THE BIT AGAIN.
3394  014226  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR
3395  014230  011004                          MOV     (R0),R4         ;READ THE REGISTER
3396  014232  105004                          CLRB    R4              ;CLEAR LOW BYTE
3397  014234  030510                          BIT     R5,(R0)         ;WAS BIT CLEARED BY DEVICE.CLR?
3398  014236  001001                          BNE     4$              ;BR IF NO (IT SHOULDN'T BE CLEAR)
3399  014240  104002                          ERROR   2               ;*BIT CLEARED BY DEVICE.CLR
3400  014242  104401                  4$:     SCOP1                   ;LOCK ON BIT? SW09=1
3401  014244  006305                          ASL     R5              ;CHANGE TO NEXT BIT
3402  014246  001347                          BNE     1$              ;IF NOT DONE LOOP
3403  014250  012710  177400                 MOV     #177400,(R0)    ;SET ALL DTR BITS
3404  014254  005005                          CLR     R5              ;CLEAR LOCATION FOR ERROR PRINTOUT
3405  014256  005227  000000          5$:     INC     #0              ;ACT DELAY LOOP FOR
3406  014262  001375                          BNE     5$              ;RESET INSTRUCTION
3407  014264  000005                          RESET                   ;ISSUE A BUS INIT
3408  014266  011004                          MOV     (R0),R4         ;READ REGISTER
3409  014270  105004                          CLRB    R4              ;CLEAR LOW BYTE
3410  014272  005704                          TST     R4              ;DTR BITS CLEAR?
3411  014274  001401                          BEQ     .+4             ;IF YES CONTINUE
3412  014276  104002                          ERROR   2               ;IF NO PRINT ERROR
3413  014300  005037  001362                 CLR     LOCK            ;MAKE SURE TIGHT LOOP IS CLEANED UP
3414                                  ; -- END 0 MACRO ------------------
3415                                                  ; $MRR-------------------------------------
3416                                                  ; -$XZ-----------------------------
3417                                  ;************************ TEST 12 ********************************
3418                                  ;*THIS TEST PERFORMS RESET TESTING &
3419                                  ;*TESTING OF WRITE ONLY OR READ ONLY BIT
3420                                  ;* TEST BITS "RDONE, BIT11, BIT10, BIT9, BIT8, BIT2, BIT1
3421                                  ;*          BIT0, SILOAL" ARE READ ONLY AND THAT TRDY IS
3422                                  ;*          ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.
3423                                  ;*
3424                                                  ; -$XZ-----------------------------
3425                                  ;:*  TEST 12
3426                                  ;;*************************************************************
3427  014304  000004                  TST12:  SCOPE
3428  014306  012737  000012  001122          MOV     #12,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3429                                  ; -- END 0 MACRO ------------------
```

```
3430  014314  012737  014422  001360          MOV     #TST13,NEXT     ;POINT TO THE START OF THE NEXT TEST
3431  014322  013700  002042                  MOV     DZCSR,R0        ;SET ADDRESS TO R0
3432  014326  005005                          CLR     R5              ;SET EXPECTED TO 0
3433  014330  012710  027607                  MOV     #RDONE+BIT11+BIT10+BIT9+BIT8+BIT2+BIT1+BIT0+SILOAL,(R0)
3434                                                                   ;WRITE THE BITS
3435  014334  011004                          MOV     (R0),R4         ;READ BACK THE BITS
3436  014336  001401                          BEQ     1$              ;BR IF NONE ARE SET.
3437  014340  104002                          ERROR   2               ;+BITS WERE SET.
3438  014342  012710  100000          1$:     MOV     #TRDY,(R0)      ;ATTEMPT TO WRITE TRDY
3439  014346  011004                          MOV     (R0),R4         ;READ TRDY
3440  014350  001401                          BEQ     2$              ;BR IF NOT SET
3441  014352  104002                          ERROR   2               ;+
3442  014354  012705  100000          2$:     MOV     #TRDY,R5        ;SET EXPECTED BIT
3443  014360  005077  165466                  CLR     @DZLPR          ;LOAD LINE 0
3444  014364  052777  000001  165464          BIS     #TCR0,@DZTCR    ;SET TCR BIT
3445  014372  052710  000040                  BIS     #MSENAB,(R0)    ;
3446  014376  052705  000040                  BIS     #MSENAB,R5      ;SET SCAN ENABLE
3447  014402  005002                          CLR     R2              ;SET COUNTER TO ZERO
3448  014404  011004                  3$:     MOV     (R0),R4         ;READ THE REGISTER
3449  014406  020504                          CMP     R5,R4           ;BIT SET?
3450  014410  001404                          BEQ     4$              ;BR IF YES
3451  014412  104414                          DELAY                   ;STALL TIME
3452  014414  005202                          INC     R2              ;UPDATE COUNTER
3453  014416  001372                          BNE     3$              ;BR IF COUNTER NOT DONE.
3454  014420  104002                          ERROR   2               ;+TRDY NOT SET!
3455  014422                          4$:
3456                                  ; -- END 0 MACRO ------------------
3457                                                                   ; -$XZ-----------------------------
3458                                          ;+++++++++++++++++++++++++ TEST 13 +++++++++++++++++++++++++++++++
3459                                          ;+THIS TEST PERFORMS RESET TESTING AND
3460                                          ;+TESTING OF READ ONLY AND WRITE ONLY BITS
3461                                          ;+ IN REGISTER DZCSR
3462                                          ;+VERIFY THAT "TIE", "SILOEN", "RIE", "MSENAB", "MAINT"
3463                                          ;+ARE THE ONLY R/W BITS IN THE DZCSR.
3464                                          ;+THEN VERIFY THAT A RESET WILL CLEAR THESE BITS
3465                                          ;+THIS TEST ALSO CHECKS BYTE OPERATIONS ON THE CSR
3466                                                                   ; -$XZ-----------------------------
3467                                  ;:+  TEST 13
3468                                  ;;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3469  014422  000004                  TST13:  SCOPE
3470  014424  012737  000013  001122          MOV     #13.,$TSTNM     ;LOAD THE NUMBER OF THIS TEST
3471                                  ; -- END 0 MACRO ------------------
3472  014432  012737  014552  001360          MOV     #TST14,NEXT     ;POINT TO THE START OF THE NEXT TEST
3473  014440  104413                          DEVICE.CLR
3474  014442  013700  002042                  MOV     DZCSR,R0        ;SET UP FOR ERROR MESSAGE
3475  014446  012710  177757                  MOV     #^C<DCLR>,(R0)  ;TRY TO WRITE
3476  014452  012705  050150                  MOV     #TIE!SILOEN!RIE!MSENAB!MAINT,R5 ;MAKE EXPECTED
3477  014456  011004                          MOV     (R0),R4         ;ACTUAL
3478  014460  020405                          CMP     R4,R5           ;CMP EXPECTED VS ACTUAL
3479  014462  001401                          BEQ     1$              ;YES
3480  014464  104002                          ERROR   2               ;+NO
3481  014466  105010                  1$:     CLRB    (R0)            ;CLEAR LOWER BYTE OF CSR
3482  014470  105005                          CLRB    R5              ;SET EXPECTED
3483  014472  011004                          MOV     (R0),R4         ;READ CSR BITS
3484  014474  020405                          CMP     R4,R5           ;COMPARE ACTUAL TO EXPECTED
3485  014476  001401                          BEQ     3$              ;BRANCH IF SAME
```

```
3486  014500  104002                        ERROR   2                       ;OTHERWISE PRINT ERROR
3487  014502  012710  177757          3$:   MOV     @#C<DCLR>,(R0)          ;RESET CSR BITS
3488  014506  105077  165332                CLRB    @#DZCSR                 ;CLEAR HIGH BYTE OF CSR
3489  014512  012705  000150                MOV     #RIE!MSENAB!MAINT,R5
                                                                             ;SET R5 TO EXPECTED RESULTS
3490
3491  014516  011004                        MOV     (R0),R4                 ;READ CSR
3492  014520  020405                        CMP     R4,R5                   ;ACTUAL = EXPECTED?
3493  014522  001401                        BEQ     4$                      ;BRANCH IF SAME
3494  014524  104002                        ERROR   2                       ;OTHERWISE PRINTOUT ERROR
3495  014526  012710  177757          4$:   MOV     @#C<DCLR>,(R0)          ;RESET CSR BITS
3496  014532  005005                        CLR     R5                      ;SET R5 TO EXPECTED RESULTS
3497  014534  005227  000000          5$:   INC     #0                      ;DELAY TIMER FOR
3498  014540  001375                        BNE     5$                      ;ACT-11 COMPATIBILITY
3499  014542  000005                        RESET                           ;ISSUE BUS INIT
3500  014544  011004                        MOV     (R0),R4                 ;READ CSR REGISTER
3501  014546  001401                        BEQ     2$                      ;BRANCH IF CSR IS CLEAR
3502  014550  104002                        ERROR   2                       ;IF NOT PRINT ERROR
3503  014552                          2$:
3504                                                                       ; -$MRWD-------------------------
3505                                                                       ; -$XZ---------------------------
3506                                        ;********************** TEST 14 ********************************
3507                                        ;*THIS TEST PERFORMS RESET TESTING AND
3508                                        ;*TESTING OF READ ONLY REGISTER DZRBUF
3509                                        ;*AND TESTING OF WRITE ONLY REGISTER DZLPR
3510                                                                       ; -$XZ---------------------------
3511                                  ;:*  TEST 14
3512                                  ;;**************************************************************
3513  014552  000004                  TST14:  SCOPE
3514  014554  012737  000014  001122         MOV     #14,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
3515                                  ; -- END 0 MACRO ------------------
3516  014562  012737  014642  001360         MOV     #TST15,NEXT             ;POINT TO THE START OF THE NEXT TEST
3517  014570  104413                        DEVICE.CLR                      ;CLEAR DZ11
3518  014572  013700  002046                MOV     DZRBUF,R0                       ;SET UP FOR ERROR MESSAGE
3519  014576  011005                        MOV     (R0),R5                 ;SET EXPECTED
3520  014600  012777  177777  165244        MOV     #-1,@DZLPR              ;TRY TO WRITE ALL 1'S
3521  014606  011004                        MOV     (R0),R4                 ;ACTUAL
3522  014610  042705  104000                BIC     #OVALID!BIT11,R5        ;DITTO
3523  014614  020405                        CMP     R4,R5                   ;CMP ACTUAL VS EXPECTED
3524  014616  001401                        BEQ     1$                      ;IF YES,GO CONTINUE PROCESSING
3525  014620  104002                        ERROR   2                       ;*ERROR- BIT PATTERN NOT CORRECT
3526  014622  010403                  1$:   MOV     R4,R3                   ;GET A COPY OF THE ACTUAL BIT PATTERN
3527  014624  005103                        COM     R3                      ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
3528  014626  010377  165220                MOV     R3,@DZLPR               ;TRY TO WRITE
3529  014632  011004                        MOV     (R0),R4                 ;ACTUAL
3530  014634  020405                        CMP     R4,R5                   ;CMP ACTUAL VS EXPECTED
3531  014636  001401                        BEQ     2$                      ;IF YES, GET OUT OF THIS TEST
3532  014640  104002                        ERROR   2                       ;*NO
3533  014642                          2$:
3534                                  ; -- END 0 MACRO ------------------
3535                                                                       ; -$MRWD-------------------------
3536                                                                       ; -$XZ---------------------------
3537                                        ;********************** TEST 15 ********************************
3538                                        ;*THIS TEST PERFORMS RESET TESTING AND
3539                                        ;*TESTING OF READ ONLY REGISTER DZMSR
3540                                        ;*AND TESTING OF WRITE ONLY REGISTER DZTDR
3541                                                                       ; -$XZ---------------------------
```

```
3542                                    ;:*  TEST 15
3543                                    ;;*****************************************************************
3544  014642  000004                    TST15:  SCOPE
3545  014644  012737  000015  001122            MOV     #15,@TSTNM      ;LOAD THE NUMBER OF THIS TEST
3546                                    ; -- END 0 MACRO ------------------
3547  014652  012737  014726  001360            MOV     #TST16,NEXT     ;POINT TO THE START OF THE NEXT TEST
3548  014660  104413                            DEVICE.CLR              ;CLEAR DZ11
3549  014662  013700  002062                    MOV     DZMSR,R0                ;SET UP FOR ERROR MESSAGE
3550  014666  011005                            MOV     (R0),R5         ;SET EXPECTED
3551  014670  012777  177777  165170            MOV     #-1,@DZTDR      ;TRY TO WRITE ALL 1'S
3552  014676  011004                            MOV     (R0),R4         ;ACTUAL
3553  014700  020405                            CMP     R4,R5           ;CMP ACTUAL VS EXPECTED
3554  014702  001401                            BEQ     1$              ;IF YES,GO CONTINUE PROCESSING
3555  014704  104002                            ERROR   2               ;*ERROR- BIT PATTERN NOT CORRECT
3556  014706  010403                    1$:     MOV     R4,R3           ;GET A COPY OF THE ACTUAL BIT PATTERN
3557  014710  005103                            COM     R3              ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
3558  014712  010377  165150                    MOV     R3,@DZTDR       ;TRY TO WRITE
3559  014716  011004                            MOV     (R0),R4         ;ACTUAL
3560  014720  020405                            CMP     R4,R5           ;CMP ACTUAL VS EXPECTED
3561  014722  001401                            BEQ     2$              ;IF YES, GET OUT OF THIS TEST
3562  014724  104002                            ERROR   2               ;*NO
3563  014726                            2$:
3564                                    ; -- END 0 MACRO ------------------
3565
3566
3567                                                    ; -$XZ-----------------------------
3568                                    ;*************************** TEST 16 *********************************
3569                                    ;*VERIFY THAT IF WE ARE IN "STAGGERED" MODE
3570                                    ;*THAT SETTING "DTR" FOR A LINE WILL
3571                                    ;*BRING UP "RING" AND "CARRIER" FOR THE
3572                                    ;*ASSOCIATED LINE IN WHICH WE ARE STAGGERED!
3573                                    ;* LINE0 DTR= LINE1 RING AND CARRIER
3574                                    ;* LINE1 DTR= LINE0 RING AND CARRIER
3575                                    ;* LINE2 DTR= LINE3 RING AND CARRIER
3576                                    ;* LINE3 DTR= LINE 4 RING AND CARRIER
3577                                    ;*              ETC...
3578                                                    ; -$XZ-----------------------------
3579
3580                                    ;:*  TEST 16
3581                                    ;;*****************************************************************
3582  014726  000004                    TST16:  SCOPE
3583  014730  012737  000016  001122            MOV     #16,@TSTNM      ;LOAD THE NUMBER OF THIS TEST
3584                                    ; -- END 0 MACRO ------------------
3585  014736  012737  015122  001360            MOV     #TST17,NEXT     ;POINT TO THE START OF THE NEXT TEST
3586  014744  012737  015016  001362            MOV     #1$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3587  014752  105737  001414                    TSTB    EIAFLG          ;EIA OR 20MA?
3588  014756  100001                            BPL     10$             ;BR IF EIA
3589  014760  104400                            ADVANCE                 ;EXIT TEST
3590  014762  013700  002062            10$:    MOV     DZMSR,R0        ;SET REGISTER
3591  014766  104413                            DEVICE.CLR              ;INIT DZ11
3592  014770  005003                            CLR     R3              ;ZERO LINE NUMBER
3593  014772  012702  000001                    MOV     #1,R2           ;SET POINTER
3594  014776  005737  001370                    TST     MODE            ;ARE WE IN STAGGERED MODE?
3595  015002  100405                            BMI     1$              ;YES WE ARE!
3596  015004  013737  001360  001126            MOV     NEXT,@LPADR     ;LEAVE THIS TEST! NOT STAGGERED
3597  015012  000177  164110                    JMP     @@LPADR         ;EXIT
```

```
3598  015016  130237  001364          1$:     BITB    R2,LINE         ;TEST THIS LINE?
3599  015022  001004                          BNE     3$              ;YES
3600  015024  005203                  2$:     INC     R3              ;LINE #
3601  015026  106302                          ASLB    R2              ;GET NEXT LINE
3602  015030  103372                          BCC     1$              ;KEEP TESTING
3603  015032  104400                          ADVANCE                 ;ADVANCE THIS TEST
3604  015034  010204                  3$:     MOV     R2,R4           ;SAVE BINARY BIT FOR LINE #
3605  015036  032703  000001                  BIT     #BIT0,R3        ;GET STAGGERED COMPANION LINE
3606  015042  001402                          BEQ     4$              ;BR IF LINE EVEN
3607  015044  006204                          ASR     R4              ;ADJUST LINE
3608  015046  000401                          BR      5$              ;
3609  015050  006304                  4$:     ASL     R4              ;ADJUST LINE
3610  015052  005005                  5$:     CLR     R5              ;SET EXPECTED
3611  015054  150405                          BISB    R4,R5           ;
3612  015056  000305                          SWAB    R5              ;
3613  015060  150405                          BISB    R4,R5           ;
3614  015062  150277  164772                  BISB    R2,@#DZTCR      ;SET DTR
3615  015066  104414                          DELAY                   ;CABLE DELAY
3616  015070  011004                          MOV     (R0),R4         ;READ MSR REGISTER
3617  015072  020504                          CMP     R5,R4           ;OK?
3618  015074  001401                          BEQ     6$              ;YES
3619  015076  104002                          ERROR   2               ;*ERROR IN RING OR CARRIER
3620  015100  140277  164754          6$:     BICB    R2,@#DZTCR      ;CLEAR DTR
3621  015104  104414                          DELAY                   ;CABLE DELAY
3622  015106  011004                          MOV     (R0),R4         ;READ MSR
3623  015110  001402                          BEQ     7$              ;BR IF THEY CLEARED
3624  015112  005005                          CLR     R5              ;SET EXPECTED TO 0
3625  015114  104002                          ERROR   2               ;*BITS NOT CLEARED
3626  015116  104401                  7$:     SCOP1                   ;LOCK ON SIGNAL?
3627  015120  000741                          BR      2$              ;CONTINUE TEST
3628
3629                                          ; -$XZ-----------------------------
3630                                          ;********************** TEST 17 ******************************
3631                                          ;*TEST TO VERIFY THAT IF IN "EXTERNAL"
3632                                          ;*MODE; SETTING DTR FOR SELECTED LINES
3633                                          ;*WILL BRING UP "CARRIER" AND "RING"
3634                                          ;*FOR THAT SAME LINE. NOTE: IF YOU HAVE
3635                                          ;*SELECTED MODE AS "EXTERNAL"; THE H325 TEST CONNECTER
3636                                          ;*MUST BE USED ON ALL SPECIFIED LINES.
3637                                          ;*LINES MAY BE SPECIFIED BY SWR03=1
3638                                          ;*AND SWR00=1 AT START TIME OR ALTERING
3639                                          ;*STATUS MAP.
3640                                          ; -$XZ-----------------------------
3641                                  ;:*  TEST 17
3642                                  ;;**************************************************************
3643  015122  000004                  TST17:  SCOPE
3644  015124  012737  000017  001122          MOV     #17,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3645                                  ; -- END 0 MACRO ------------------
3646  015132  012737  015260  001360          MOV     #TST20,NEXT     ;POINT TO THE START OF THE NEXT TEST
3647  015140  012737  015174  001362          MOV     #3$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3648  015146  105737  001370                  TSTB    MODE            ;EXTERNAL?
3649  015152  100401                          BMI     2$              ;BR IF YES
3650  015154  104400                  1$:     ADVANCE                 ;EXIT TEST
3651  015156  105737  001414          2$:     TSTB    EIAFLG          ;YOU BETTER BE IN
3652  015162  100774                          BMI     1$              ;EIA MODE FOR THIS TEST.
3653  015164  013700  002062                  MOV     DZMSR,R0        ;SET REGISTER
```

```
3654  015170  012702  000001            MOV     #1,R2           ;SET LINE POINTER
3655  015174  130237  001364    3$:     BITB    R2,LINE         ;LINE SELECTED?
3656  015200  001003                    BNE     5$              ;BR IF YES
3657  015202  106302            4$:     ASLB    R2              ;NEXT LINE
3658  01520   103373                    BCC     3$              ;CONTINUE TEST
3659  015206  104400                    ADVANCE                 ;ADVANCE THIS TEST
3660  015210  005005            5$:     CLR     R5              ;SET EXPECTED
3661  015212  150205                    BISB    R2,R5           ;
3662  015214  000305                    SWAB    R5              ;
3663  015216  150205                    BISB    R2,R5           ;
3664  015220  150277  164634            BISB    R2,@#DZTCR      ;SET DTR
3665  015224  104414                    DELAY                   ;CABLE DELAY
3666  015226  011004                    MOV     (R0),R4         ;READ MSR
3667  015230  020504                    CMP     R5,R4           ;BITS OK?
3668  015232  001401                    BEQ     6$              ;BR IF YES
3669  015234  104002                    ERROR   2               ;CARRIER OR RING ERROR
3670  015236  140277  164616    6$:     BICB    R2,@#DZTCR      ;CLEAR DTR
3671  015242  104414                    DELAY                   ;CABLE DELAY
3672  015244  011004                    MOV     (R0),R4         ;READ MSR
3673  015246  001402                    BEQ     7$              ;BR IF BITS CLEARED
3674  015250  005005                    CLR     R5              ;CLEAR EXPECTED LOC.
3675  015252  104002                    ERROR   2               ;BITS NOT CLEARED.
3676  015254  104401            7$:     SCOP1                   ;LOCK ON LINE?
3677  015256  000751                    BR      4$              ;CONTINUE TEST
3678
3679                                    ; -$TLINE-------------------------------
3680                                    ; -$XZ---------------------------------
3681                            ;********************** TEST 20 ********************************
3682                            ;* THIS TEST VERIFIES THAT TRDY IS SET WHEN A LINE
3683                            ;* IS READY TO BE LOADED, AND THAT THE LINE SPECI-
3684                            ;* FIED IN BITS 8-10 OF DZCSR CORRESPOND
3685                            ;* TO THE LINE SELECTED IN DZTCR
3686                                    ; -$XZ---------------------------------
3687                            ;:*  TEST 20
3688                            ;;*************************************************************
3689  015260  000004            TST20:  SCOPE
3690  015262  012737  000020  001122    MOV     #20,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3691                            ; -- END 0 MACRO ------------------
3692  015270  012737  015404  001360    MOV     #TST21,NEXT     ;POINT TO THE START OF THE NEXT TEST
3693  015276  104413                    DEVICE.CLR              ;ISSUE A "DEVICE CLEAR" (RESET)
3694  015300  013700  002042            MOV     DZCSR,R0        ;SET POINTER
3695  015304  012705  100040            MOV     #MSENAB!TRDY,R5 ;START THE EXPECTED LINE NUMBER AT 0
3696  015310  005037  001372            CLR     SAVLIN          ;SET UP FOR ERROR PRINTOUTS
3697  015314  012702  000001            MOV     #1,R2           ;USING R2 AS A BIT POINTER, POINT TO LINE 0
3698  015320  130237  001364    1$:     BITB    R2,LINE         ;IS THIS LINE SELECTED?
3699  015324  001420                    BEQ     5$              ;IF NO, SKIP THE STARTUP
3700  015326  050277  164524    2$:     BIS     R2,@#DZTCR      ;SET THE GO BIT FOR THIS LINE
3701  015332  052710  000040            BIS     #MSENAB,(R0)    ;START THE SCANNER
3702  015336  005004                    CLR     R4              ;SET FOR DELAY
3703  015340  032710  100000    3$:     BIT     #TRDY,(R0)      ;TX READY?
3704  015344  001004                    BNE     4$              ;BR IF YES
3705  015346  104414                    DELAY                   ;DELAY
3706  015350  005204                    INC     R4              ;COUNTER
3707  015352  001372                    BNE     3$              ;BR IF <>0!
3708  015354  104003                    ERROR   3               ;*TX NOT READY!
3709  015356  011004            4$:     MOV     (R0),R4         ;GET THE LINE POINTED TO BY THE SCANNER
```

```
3710  015360  020405                          CMP     R4,R5           ;IS THE LINE NUMBER WHAT IT SHOULD BE?
3711  015362  001401                          BEQ     5$              ;IF YES,GO WORK ON THE NEXT LINE
3712  015364  104002                          ERROR   2               ;*LINE NUMBER DID NOT MATCH TCR BIT
3713  015366  062705  000400          5$:     ADD     #400,R5         ;POINT TO THE NEXT EXPECTED LINE
3714  015372  104413                          DEVICE.CLR              ;ISSUE A "DEVICE CLEAR" (RESET)
3715  015374  005237  001372                  INC     SAVLIN          ;ADJUST FOR NEXT LINE
3716  015400  106302                          ASLB    R2              ;POINT TO THE NEXT LINE.ARE ALL LINES TESTED?
3717  015402  103346                          BCC     1$              ;IF NOT, GO DO THE NEXT LINE
3718  015404                          6$:
3719                                  ; -- END 0 MACRO ------------------
3720                                                                  ; -$XZ-----------------------------
3721                                          ;************************ TEST 21 *******************************
3722                                          ;*TEST TO TRANSMIT ONE CHAR AND
3723                                          ;*RECEIVE ONE CHAR ON ONE LINE
3724                                          ;*AT A TIME. THE CHAR IS "252" AND
3725                                          ;*ALL SELECTED LINES WILL BE TURNED ON
3726                                          ;*ONE AT A TIME. THIS IS THE FIRST TIME ANY
3727                                          ;*DATA IS CHECKED IN THE RECEIVER.
3728                                          ;*USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
3729                                          ;*WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.
3730                                                                  ; -$XZ-----------------------------
3731                                  ;:*  TEST 21
3732                                  ;;*************************************************************
3733  015404  000004                  TST21:  SCOPE
3734  015406  012737  000021  001122          MOV     #21,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3735                                  ; -- END 0 MACRO ------------------
3736  015414  012737  015742  001360          MOV     #TST22,NEXT     ;POINT TO THE START OF THE NEXT TEST
3737  015422  012737  015720  001362          MOV     #16$,LOCK       ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3738                                                                  ;$LINEUP---------------------------
3739                                                                  ; -$MRESET-------------------------
3740  015430  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3741                                  ; -- END 0 MACRO ------------------
3742  015432  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
3743  015436  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
3744  015442  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
3745  015446  001402                          BEQ     2$              ;NO
3746  015450  010177  164376                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
3747  015454  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
3748  015456  106302                          ASLB    R2              ;GOT 'EM ALL ?
3749  015460  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
3750  015462  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
3751                                  ; -- END 0 MACRO ------------------
3752  015466  012702  000001                  MOV     #1,R2           ;LINE POINTER
3753  015472  052777  000040  164342          BIS     #MSENAB,@DZCSR  ;START SCANNER
3754  015500  030237  001364          3$:     BIT     R2,LINE         ;VALID LINE ?
3755  015504  001462                          BEQ     14$             ;NO SET UP NEXT LINE
3756  015506  010277  164344                  MOV     R2,@DZTCR       ;SET TCR BIT
3757  015512  032777  000200  164322  4$:     BIT     #RDONE,@DZCSR   ;IS REC DONE = 0 ?
3758  015520  001401                          BEQ     5$              ;IF YES, ALLOW TIME FOR TRDY TO SET
3759  015522  104020                          ERROR   20              ;*REC DONE SHOULD = 0
3760  015524  005005                  5$:     CLR     R5
3761  015526  032777  100000  164306  6$:     BIT     #TRDY,@DZCSR
3762  015534  001004                          BNE     7$
3763  015536  104414                          DELAY
3764  015540  105205                          INCB    R5
3765  015542  001371                          BNE     6$
```

```
3766  015544  104003                          ERROR   3               ;*TROY FAILED TO SET!
3767  015546  112777  000252  164312  7$:     MOVB    #252,@DZTDR     ;LOAD CHARACTER
3768  015554  013705  001372                  MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
3769                                          ; -$STAG----------------------------------
3770  015560  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
3771  015564  001406                          BEQ     10$             ;IF NOT, SKIP STAGGERED SETUP
3772
3773                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3774
3775  015566  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
3776  015570  103402                          BCS     8$              ;IF IT IS SET, GO CLEAR IT
3777  015572  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
3778  015574  000401                          BR      9$              ;SKIP THE CLEARING
3779  015576  000241                  8$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3780  015600  006105                  9$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
3781                                  ; -- END 0 MACRO ----------------
3782  015602  000305                  10$:    SWAB    R5              ;MOVE THE LINE NUMBER TO THE UPPER BYTE
3783  015604  152705  000252                  BISB    #252,R5         ;ADD CHARACTER
3784  015610  052705  100000                  BIS     #DVALID,R5      ;ADD DATA VALID
3785  015614  005003                          CLR     R3
3786  015616  032777  000200  164216  11$:    BIT     #RDONE,@DZCSR
3787  015624  001004                          BNE     12$
3788  015626  104414                          DELAY
3789  015630  005203                          INC     R3
3790  015632  001371                          BNE     11$
3791  015634  104004                          ERROR   4               ;*RDONE FAILED TO SET!
3792  015636  017704  164204          12$:    MOV     @DZRBUF,R4      ;LOAD THE VALUE ACTUALLY RECEIVED
3793  015642  020405                          CMP     R4,R5           ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
3794  015644  001401                          BEQ     13$             ;IF YES, GO DO THE NEXT LINE
3795  015646  104006                          ERROR   6               ;*NO DATA/CONTENTS DID NOT COMPARE
3796  015650  104401                  13$:    SCOP1                   ;CHECK TO SEE IF SWITCH NINE IS SET
3797  015652  040277  164200          14$:    BIC     R2,@DZTCR       ;CLEAR TCR BIT FOR THAT LINE.
3798  015656  005237  001372          15$:    INC     SAVLIN          ;INC EXPECTED LINE
3799  015662  013700  001372                  MOV     SAVLIN,R0       ;SET UP CHARACTER OFFSET
3800  015666  006300                          ASL     R0              ;MAKE THE OFFSET A POWER OF TWO
3801  015670  106302                          ASLB    R2              ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
3802  015672  103302                          BCC     3$              ;IF NO, GO AROUND AGAIN FOR NEXT LINE
3803  015674  005003                          CLR     R3              ;THIS CODE HAS BEEN INSERTED
3804  015676  104414                  17$:    DELAY                   ;TO DETECT A PROBLEM FOUND IN FAULT
3805  015700  105203                          INCB    R3              ;INSERTION. IF AN ERROR OCCURS MORE
3806  015702  001375                          BNE     17$             ;THAN ONE WORD WAS RECIEVED ON
3807  015704  032777  000200  164130          BIT     #RDONE,@DZCSR   ;LINE 7.
3808  015712  001401                          BEQ     18$
3809  015714  104020                          ERROR   20
3810  015716  104400                  18$:    ADVANCE                 ;GO TO NEXT TEST
3811
3812                                  ;TIGHT SCOPE LOOP FOR THIS TEST. LOOP TRANSMITS CHARACTERS ONLY
3813
3814  015720  032777  100000  164114  16$:    BIT     #TRDY,@DZCSR    ;IS TRANSMITTER READY?
3815  015726  001774                          BEQ     16$             ;IF NOT, WAIT FOR IT
3816  015730  112777  000252  164130          MOVB    #252,@DZTDR     ;LOAD THE CHARACTER
3817  015736  104401                          SCOP1                   ;LOOP AGIN IF SW09=1
3818  015740  000744                          BR      14$             ;OTHERWISE, GO PICK UP THE TEST NORMALLY
3819
3820                                          ; -$XZ-----------------------------
3821                                  ;************************ TEST 22 ********************************
```

```
                                        ;* THIS TEST PROVES THAT THE TRANSMITTER TRANSMITS
                                        ;*CHARACTERS (FLAG MODE)AND THE RECEIVER RECEIVES (FLAG MODE)
                                        ;*(ONE LINE AT A TIME   BASED UPON VALID LINES)
                                        ;*THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED
                                                ; -$XZ-----------------------------
                                ;:*  TEST 22
                                ;;*******************************************************
015742  000004                 TST22:  SCOPE
015744  012737  000022  001122          MOV     #22,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
                                ; -- END 0 MACRO ------------------
015752  012737  016270  001360          MOV     #TST23,NEXT     ;POINT TO THE START OF THE NEXT TEST
015760  012737  016074  001362          MOV     #4$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
                                                ;$LINEUP--------------------------------
                                                ; -$MRESET------------------------------
015766  104417                          DCLASH                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
                                ; -- END 0 MACRO ------------------
015770  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
015774  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
016000  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
016004  001402                          BEQ     2$              ;NO
016006  010177  164040                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
016012  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
016014  106302                          ASLB    R2              ;GOT 'EM ALL ?
016016  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
016020  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
                                ; -- END 0 MACRO ------------------
016024  012700  001422                  MOV     #TDO,R0         ;POINT TO THE DATA AREA
016030  005020                          CLR     (R0)+           ;CLEAR A DATA WORD
016032  022700  001462                  CMP     #STOP,R0        ;FINISHED ?
016036  001374                          BNE     .-6             ;NO
016040  005000                          CLR     R0              ;CLEAR OFFSET
016042  013737  002046  001400          MOV     DZRBUF,REGIST   ;SAVE FOR ERROR MSG
016050  012702  000001                  MOV     #1,R2           ;LINE POINTER
016054  052777  000040  163760          BIS     #MSENAB,@DZCSR  ;START SCANNER
016062  030237  001364          3$:     BIT     R2,LINE         ;VALID LINE ?
016066  001465                          BEQ     14$             ;NO SET UP NEXT LINE
016070  010277  163762                  MOV     R2,@DZTCR       ;SET TCR BIT
016074  032777  000200  163740  4$:     BIT     #RDONE,@DZCSR   ;IS REC DONE = 0 ?
016102  001401                          BEQ     5$              ;IF YES, ALLOW TIME FOR TRDY TO SET
016104  104020                          ERROR   20              ;*REC DONE SHOULD = 0
016106  005005                  5$:     CLR     R5
016110  032777  100000  163724  6$:     BIT     #TRDY,@DZCSR
016116  001004                          BNE     7$
016120  104414                          DELAY
016122  105205                          INCB    R5
016124  001371                          BNE     6$
016126  104003                          ERROR   3               ;*TRDY FAILED TO SET!
016130  116077  001422  163730  7$:     MOVB    TDO(R0),@DZTDR  ;LOAD CHARACTER
016136  013705  001372                  MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
                                                ; -$STAG--------------------------------
016142  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
016146  001406                          BEQ     10$             ;IF NOT, SKIP STAGGERED SETUP

                                ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER

016150  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
```

```
016152  103402                            BCS     8$              ;IF IT IS SET, GO CLEAR IT
016154  000261                            SEC                     ;IF IT IS CLEAR SET IT HERE
016156  000401                            BR      9$              ;SKIP THE CLEARING
016160  000241                    8$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
016162  006105                    9$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
                                  ; -- END O MACRO ******************
016164  000305                    10$:    SWAB    R5              ;MOVE THE LINE NUMBER TO THE UPPER BYTE
016166  156005  001422                    BISB    TDC(R0),R5      ;ADD CHARACTER
016172  052705  100000                    BIS     #DVALID,R5      ;ADD DATA VALID
016176  005003                            CLR     R3
016200  032777  000200  163634    11$:    BIT     #RDONE,@DZCSR
016206  001004                            BNE     12$
016210  104414                            DELAY
016212  005203                            INC     R3
016214  001371                            BNE     11$
016216  104004                            ERROR   4               ;*RDONE FAILED TO SET!
016220  017704  163622            12$:    MOV     @DZRBUF,R4      ;LOAD THE VALUE ACTUALLY RECEIVED
016224  020405                            CMP     R4,R5           ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
016226  001401                            BEQ     13$             ;IF YES, GO DO THE NEXT LINE
016230  104006                            ERROR   6               ;*NO DATA/CONTENTS DID NOT COMPARE
016232  104401                    13$:    SCOP1                   ;CHECK TO SEE IF SWITCH NINE IS SET
016234  105260  001422                    INCB    TDC(R0)         ;INCREMENT BINARY PATTERN FOR THIS LINE
016240  001315                            BNE     4$              ;GO 'ROUND AGAIN FOR NEXT CHARACTER
016242  040277  163610            14$:    BIC     R2,@DZTCR       ;CLEAR TCR BIT FOR THAT LINE.
016246  005237  001372            15$:    INC     SAVLIN          ;INC EXPECTED LINE
016252  013700  001372                    MOV     SAVLIN,R0       ;SET UP CHARACTER OFFSET
016256  006300                            ASL     R0              ;MAKE THE OFFSET A POWER OF TWO
016260  106302                            ASLB    R2              ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
016262  103277                            BCC     3$              ;IF NO, GO AROUND AGAIN FOR NEXT LINE
016264  005037  001362                    CLR     LOCK            ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST


                                                                  ; -$XZ-------------------------------
                                          ;************************ TEST 23 *****************************
                                          ;*THIS TEST WILL PROVE THAT EACH RECEIVING LINE CAN
                                          ;*BE DISABLED BY SETTING THE RCVON BIT TO ZERO
                                          ;*FOR EACH LINE IN THE LPR REGISTER.  IT ALSO
                                          ;*VERIFIES THAT MASTER CLEAR WILL ZERO DVALID FOR
                                          ;*CHARACTERS STORED IN THE SILO.
                                                                  ; -$XZ-------------------------------
                                  ;;*  TEST 23
                                  ;;**********************************************************
016270  000004                    TST23:  SCOPE
016272  012737  000023  001122            MOV     #23,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
                                  ; -- END O MACRO ******************
016300  012737  016622  001360            MOV     #TST24,NEXT     ;POINT TO THE START OF THE NEXT TEST
016306  105037  001420                    CLRB    DONFLG          ;INITIALIZE FOR FIRST TEST LOOP
016312  005037  001372                    CLR     SAVLIN          ;ZERO LINE NO. FOR ERROR REPORT
016316  104417                            DCLASM                  ;EXECUTE MASTER CLEAR
016320  013701  001366                    MOV     PAR,R1          ;STORE DEFAULT PARAMETERS
016324  042701  010000                    BIC     #RCVON,R1       ;CLEAR RCVON BIT
016330  012702  000001            1$:     MOV     #1,R2           ;INIT LINE POINTER
016334  010177  163512            2$:     MOV     R1,@DZLPR       ;LOAD LINE PARAMETER REGISTER
016340  005201                            INC     R1              ;SET R1 FOR NEXT LINE
016342  106302                            ASLB    R2              ;SHIFT R2 TO NEXT LINE
016344  103373                            BCC     2$              ;ALL LINES LOADED?
```

```
3934  016346  012701  000252                MOV     #252,R1         ;LOAD TRANSMITTING CHARACTER
3935  016352  013702  001364                MOV     LINE,R2         ;COPY ACTIVE LINE BITS
3936  016356  010277  163474                MOV     R2,@DZTCR       ;LOAD TCR BITS
3937  016362  052777  000040  163452        BIS     #MSENAB,@DZCSR  ;SET SCANNER
3938  016370  005005              3$:       CLR     R5              ;INIT DELAY COUNTER
3939  016372  005777  163444        4$:     TST     @DZCSR          ;TRDY SET?
3940  016376  100404                        BMI     5$              ;IF YES BRANCH
3941  016400  104414                        DELAY                   ;IF NOT THEN WAIT
3942  016402  005205                        INC     R5              ;INCREMENT DELAY COUNTER
3943  016404  001372                        BNE     4$              ;DELAY DONE?
3944  016406  104003                        ERROR   3               ;IF YES TRDY FAILED TO SET
3945  016410  117705  163430        5$:     MOVB    @DZCSR,R5       ;MOVE LINE NO. INTO R5
3946  016414  012703  000001                MOV     #1,R3           ;INIT TCR POINTER
3947  016420  042705  177770                BIC     #^C<7>,R5       ;ISOLATE LINE NO.
3948  016424  001403                        BEQ     21$             ;IF LINE 0 GO TEST TRANSM. FLAG
3949  016426  106303              20$:      ASLB    R3              ;POINT R3 TO NEXT TCR BIT
3950  016430  005305                        DEC     R5              ;DECREMENT R5 UNTIL R3 POINTS
3951  016432  001375                        BNE     20$             ;TO CORRECT TCR BIT
3952  016434  030302              21$:      BIT     R3,R2           ;HAS THIS LINE BEEN SERVICED?
3953  016436  001007                        BNE     6$              ;IF NOT GO SEND CHARACTER
3954  016440  140377  163412                BICB    R3,@DZTCR       ;IF YES CLEAR TCR BIT
3955  016444  001351                        BNE     3$              ;IF MORE LINES SET BRANCH
3956  016446  105737  001420                TSTB    DONFLG          ;IF ALL LOADED IS THIS SECOND PASS
3957  016452  001040                        BNE     12$             ;IF YES BRANCH TO SECOND PART OF TEST
3958  016454  000404                        BR      7$              ;OTHERWISE CONTINUE WITH FIRST PART
3959  016456  110177  163404        6$:     MOVB    R1,@DZTDR       ;TRANSMIT CHARACTER
3960  016462  040302                        BIC     R3,R2           ;CLEAR FLAG FOR THIS LINE
3961  016464  000741                        BR      3$              ;GO WAIT FOR NEXT LINE
3962  016466  005077  163364        7$:     CLR     @DZTCR          ;CLEAR TCR BITS
3963  016472  005005                        CLR     R5              ;CLEAR DELAY COUNTER
3964  016474  104414              8$:       DELAY                   ;WAIT FOR LAST CHARACTER
3965  016476  005205                        INC     R5              ;INCREMENT DELAY COUNTER
3966  016500  001375                        BNE     8$              ;IF NOT FINISHED CONTINUE WAITING
3967  016502  105777  163334                TSTB    @DZCSR          ;RDONE BIT SET?
3968  016506  100003                        BPL     10$             ;IF NO CONTINUE
3969  016510  005037  001372                CLR     SAVLIN          ;IF YES SET LINE NO. TO ZERO
3970  016514  104020                        ERROR   20              ;AND PRINT ERROR
3971  016516  017704  163324        10$:    MOV     @DZRBUF,R4      ;READ SILO
3972  016522  100007                        BPL     11$             ;IF DVALID IS ZERO BRANCH
3973  016524  000304                        SWAB    R4              ;IF SET THEN
3974  016526  042704  177770                BIC     #^C<7>,R4       ;ISOLATE LINE NO. IN R4
3975  016532  010437  001372                MOV     R4,SAVLIN       ;SET SAVLIN FOR ERROR REPORT
3976  016536  104017                        ERROR   17              ;DATA VALID SHOULD NOT BE SET
3977  016540  000766                        BR      10$             ;GO READ SILO AGAIN
3978  016542  105237  001420        11$:    INCB    DONFLG          ;PREPARE FOR SECOND PART OF TEST
3979  016546  013701  001366                MOV     PAR,R1          ;MOVE DEFAULT PARAMETERS TO R1
3980  016552  000666                        BR      1$              ;GO LOAD LPR REGISTER
3981  016554  005005              12$:      CLR     R5              ;INIT DELAY COUNTER
3982  016556  104414              13$:      DELAY                   ;WAIT FOR LAST CHARACTER
3983  016560  005205                        INC     R5              ;TO BE RECEIVED
3984  016562  001375                        BNE     13$             ;DELAY FINISHED?
3985  016564  104413                        DEVICE.CLR              ;IF YES EXECUTE MASTER CLEAR
3986  016566  000240                        NOP
3987  016570  000240                        NOP
3988  016572  105777  163244                TSTB    @DZCSR          ;RDONE SET?
3989  016576  100003                        BPL     14$             ;IF NOT BRANCH
```

```
3990  016600  005037  001372              CLR     SAVLIN          ;IF YES THEN PRINT OUT
3991  016604  104020                      ERROR   20              ;REPORT
3992  016606  017704  163234      14$:    MOV     @DZRBUF,R4      ;READ SILO
3993  016612  100003                      BPL     15$             ;DATA VALID SET?
3994  016614  005037  001372              CLR     SAVLIN          ;IF YES THEN PRINT OUT
3995  016620  104017                      ERROR   17              ;ERROR REPORT
3996  016622                      15$:
3997                                                              ; -$XZ-----------------------------
3998                                      ;************************ TEST 24 *******************************
3999                                      ;*THIS TEST WILL PROVE THAT:
4000                                      ;* 1) THE TRANSMITTER "BREAK BIT" WORKS
4001                                      ;* 2) THE RECEIVER CAN FLAG "FRAMING ERRORS"
4002                                      ;* 3) THE RECEIVER CAN FLAG "PARITY ERRORS"
4003                                      ;*ONLY ONE LINE AT A TIME WILL BE EXERCISED.
4004                                      ;*THIS TEST WILL NOT BE  EXERCISED UNLESS
4005                                      ;*CONNECTED BY AN H325, H3271, OR H3190 CONNECTOR
4006                                                              ; -$XZ-----------------------------
4007                              ;:*  TEST 24
4008                              ;;*************************^^*********************************************
4009  016622  000004              TST24:  SCOPE
4010  016624  012737  000024  001122      MOV     #24,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
4011                              ; -- END 0 MACRO ------------------
4012  016632  012737  017100  001360      MOV     #TST25,NEXT     ;POINT TO THE START OF THE NEXT TEST
4013  016640  012737  016736  001362      MOV     #3$,LOCK        ;SET .OR LOOP
4014  016646  005737  001370              TST     MODE            ;ARE WE RUNNING IN INTERNAL MODE?
4015  016652  001510                      BEQ     12$             ;IF SO, SKIP THIS TEST
4016                                                              ; -$MRESET-------------------------
4017  016654  104417                      DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4018                              ; -- END 0 MACRO ------------------
4019  016656  013701  001366              MOV     PAR,R1          ;PICK UP PARAMETERS
4020  016662  052701  000300              BIS     #ODDPAR!PARITY,R1 ;FORCE ODD PARITY
4021  016666  012700  000001              MOV     #1,R0           ;PICK UP INIT POINTER
4022  016672  030037  001364      1$:     BIT     R0,LINE         ;SHOULD THIS LINE BE SET UP ?
4023  016676  001402                      BEQ     2$              ;IF NOT,DON'T SET IT UP
4024  016700  010177  163146              MOV     R1,@DZLPR       ;OTHERWISE, SET UP LINE PARAMETERS
4025  016704  005201              2$:     INC     R1
4026  016706  106300                      ASLB    R0              ;GOT 'EM ALL ?
4027  016710  103370                      BCC     1$              ;NO
4028  016712  005037  001372              CLR     SAVLIN          ;CLEAR LINE #
4029  016716  012702  000001              MOV     #1,R2           ;LINE POINTER
4030  016722  052777  000040  163112      BIS     #MSENAB,@DZCSR  ;SET MASTER SCAN ENABLE
4031  016730  013737  002046  001400      MOV     DZRBUF,REGIST   ;SAVE FOR ERRR MESSAGE
4032  016736  030237  001364      3$:     BIT     R2,LINE
4033  016742  001446                      BEQ     11$
4034  016744  010277  163106              MOV     R2,@DZTCR       ;SET TCR BIT
4035  016750  110277  163114              MOVB    R2,@#DZTDR      ;SET BREAK BIT
4036  016754  112777  000377  163104  4$: MOVB    #377,@DZTDR     ;LOAD CHARACTER
4037  016762  013705  001372              MOV     SAVLIN,R5       ;MAKE EXPECTED DATA
4038                                                              ; -$STAG---------------------------------
4039  016766  105737  001371              TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
4040  016772  001406                      BEQ     7$              ;IF NOT, SKIP STAGGERED SETUP
4041
4042                                      ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
4043
4044  016774  006205                      ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
4045
```

```
4046  016776  103402                          BCS     5$              ;IF IT IS SET, GO CLEAR IT
4047  017000  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
4048  017002  000401                          BR      6$              ;SKIP THE CLEARING
4049  017004  000241                  5$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
4050  017006  006105                  6$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
4051                                  ; -- END O MACRO ----------------
4052  017010  000305                  7$:     SWAB    R5              ;PUT LINE NUMBER IN UPPER BYTE
4053  017012  052705  130000                  BIS     @DVALID!PARER!FRMERR,R5 ;ADD EXPECTED
4054  017016  005004                          CLR     R4
4055  017020  032777  000200  163014  8$:     BIT     @RDONE,@DZCSR
4056  017026  001004                          BNE     9$
4057  017030  104414                          DELAY
4058  017032  005204                          INC     R4
4059  017034  001371                          BNE     8$
4060  017036  104004                          ERROR   4               ;*RDONE FAILED TO SET!
4061  017040  017704  163002          9$:     MOV     @DZRBUF,R4      ;ACTUAL
4062  017044  020405                          CMP     R4,R5           ;CMP ACTUAL VS EXPECTED. DO THEY MATCH?
4063  017046  001401                          BEQ     10$             ;IF YES, GO CLEAN UP
4064  017050  104006                          ERROR   6               ;*DATA/CONTENTS FAILED TO COMPARE
4065  017052  105077  163012          10$:    CLRB    @#DZTDR         ;CLEAR BREAK BITS
4066  017056  104401                          SCOP1                   ;LOOP?
4067  017060  005237  001372          11$:    INC     SAVLIN          ;INC LINE #
4068  017064  040277  162766                  BIC     R2,@DZTCR       ;CLEAR TCR BIT
4069  017070  106302                          ASLB    R2
4070  017072  103321                          BCC     3$
4071  017074  005037  001362          12$:    CLR     LOCK            ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST
4072                                                                  ; -$LVLTST----------------------------
4073                                                                  ; -$XZ--------------------------------
4074                                  ;********************** TEST 25 ********************************
4075                                  ;* THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
4076                                  ;*WHILE THE PROCESSOR STATUS IS SET EXACTLY
4077                                  ;*TO WHAT THE DZ11 PRIORITY IS SET TO.
4078                                  ;*DEFAULT PRIORITY IS AT  5 (240).
4079                                                                  ; -$XZ--------------------------------
4080                                  ;:*  TEST 25
4081                                  ;;*****************************************************************
4082  017100  000004                  TST25:  SCOPE
4083  017102  012737  000025  001122          MOV     #25,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
4084                                  ; -- END O MACRO ----------------
4085  017110  012737  017410  001360          MOV     #TST26,NEXT     ;POINT TO THE START OF THE NEXT TEST
4086                                                                  ;$LINEUP-----------------------------
4087                                                                  ; -$MRESET---------------------------
4088  017116  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4089                                  ; -- END O MACRO ----------------
4090  017120  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
4091  017124  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
4092  017130  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
4093  017134  001402                          BEQ     2$              ;NO
4094  017136  010177  162710                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
4095  017142  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
4096  017144  106302                          ASLB    R2              ;GOT 'EM ALL ?
4097  017146  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
4098  017150  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
4099                                  ; -- END O MACRO ----------------
4100  017154  106437  030306                  MTPS    @#DZPRT         ;SET CPU STATUS TO DZ11 PRIO.
4101  017160  113777  001364  162670          MOVB    LINE,@DZTCR     ;ENABLE THE VALID LINES
```

```
017166                                  3$:
                                                                        ; -$INTSET-*****************************
017166  012777  017256  162702                  MOV     #6$,@DZTIV      ;SET UP THE TRANSMITTER INTERRUPT VECTOR
017174  012777  017264  162670                  MOV     #7$,@DZRIV      ;SET UP THE RECEIVER INTERRUPT VECTOR
017202  013777  030306  162664                  MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
017210  013777  030306  162662                  MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
017216  052777  040040  162616                  BIS     #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
                                        ; -- END 0 MACRO ******************
017224  005005                                  CLR     R5
017226  032777  100000  162606          4$:     BIT     #TRDY,@DZCSR
017234  001403                                  BEQ     5$
017236  000240                                  NOP
017240  000240                                  NOP
017242  000412                                  BR      8$
017244  104414                          5$:     DELAY
017246  005205                                  INC     R5
017250  001366                                  BNE     4$
017252  104003                                  ERROR   3               ;*TRDY NOT SET!
017254  000405                                  BR      8$
017256  104010                          6$:     ERROR   10              ;*TRANSMITTER SHOULD NOT INTERRUPT
017260  022626                                  CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
017262  000402                                  BR      8$              ;CONTINUE TEST
017264  104012                          7$:     ERROR   12              ;*RECEIVER SHOULD NOT INTERRUPT
017266  022626                                  CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
017270  042777  040000  162544          8$:     BIC     #TIE,@DZCSR     ;RESET TRANSMITTER INTERRUPT ENABLE
                                                                        ; -$INTSET-*****************************
017276  012777  017374  162572                  MOV     #11$,@DZTIV     ;SET UP THE TRANSMITTER INTERRUPT VECTOR
017304  012777  017402  162560                  MOV     #12$,@DZRIV     ;SET UP THE RECEIVER INTERRUPT VECTOR
017312  013777  030306  162554                  MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
017320  013777  030306  162552                  MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
017326  052777  000140  162506                  BIS     #RIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
                                        ; -- END 0 MACRO ******************
017334  113777  001422  162524                  MOVB    TDO,@DZTDR      ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
017342  005005                                  CLR     R5
017344  032777  000200  162470          9$:     BIT     #RDONE,@DZCSR
017352  001403                                  BEQ     10$
017354  000240                                  NOP
017356  000240                                  NOP
017360  000412                                  BR      13$
017362  104414                          10$:    DELAY
017364  005205                                  INC     R5
017366  001366                                  BNE     9$
017370  104004                                  ERROR   4               ;*NO RX DONE! (NOT SET)
017372  000405                                  BR      13$             ;CONTINUE TEST
017374  104010                          11$:    ERROR   10              ;*TRANSMITTER SHOULD NOT INTERRUPT
017376  022626                                  CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
017400  000402                                  BR      13$             ;CONT TEST
017402  104012                          12$:    ERROR   12              ;*RECEIVER SHOULD NOT INTERRUPT
017404  022626                                  CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
017406                                  13$:
017406  104413                                  DEVICE.CLR              ;ISSUE DEVICE CLEAR (RESET)
                                        ; -- END 0 MACRO ******************
                                        ; -- END 0 MACRO ******************
                                                                        ; -$LVLTST-*************************
                                                                        ; -$XZ-****************************
                                                ;*********************** TEST 26 ****************************
```

```
                                        ;* THIS TEST VERIFIES THAT THE DEVICE DOES INTERRUPT
                                        ;*WHILE THE PROCESSOR STATUS IS SET TO EXACTLY
                                        ;*ONE LEVEL LOWER THAN THE DZ11. DZ11 PRIORITY
                                        ;*DEFAULT TO LEVEL 5 MINUS ONE LEVEL IS LEVEL 4.
                                                        ; -$XZ-----------------------------
                                ;:*  TEST 26
                                ;;*****************************************************************
017410  000004                TST26:  SCOPE
017412  012737  000026  001122        MOV     #26.,@#TSTNM    ;LOAD THE NUMBER OF THIS TEST
                                ; -- END 0 MACRO ------------------
017420  012737  017736  001360        MOV     #TST27,NEXT     ;POINT TO THE START OF THE NEXT TEST
                                                        ;$LINEUP---------------------------------
                                                        ; -$MRESET-------------------------------
017426  104417                        DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
                                ; -- END 0 MACRO ------------------
017430  013701  001366                MOV     PAR,R1          ;PICK UP PARAMETERS
017434  012702  000001                MOV     #1,R2           ;PICK UP INIT POINTER
017440  030237  001364        1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
017444  001402                        BEQ     2$              ;NO
017446  010177  162400                MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
017452  005201                2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
017454  106302                        ASLB    R2              ;GOT 'EM ALL ?
017456  103370                        BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
017460  005037  001372                CLR     SAVLIN          ;CLEAR LINE # INDICATOR
                                ; -- END 0 MACRO ------------------
017464  106437  030310                MTPS    @#LESS1         ;MAKE CPU ONE LEVEL LOWER THAN DZ11
017470  113777  001364  162360        MOVB    LINE,@DZTCR     ;ENABLE THE VALID LINES
017476                          3$:
                                                        ; -$INTSET-------------------------------
017476  012777  017570  162372        MOV     #6$,@DZTIV      ;SET UP THE TRANSMITTER INTERRUPT VECTOR
017504  012777  017606  162360        MOV     #7$,@DZRIV      ;SET UP THE RECEIVER INTERRUPT VECTOR
017512  013777  030306  162354        MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
017520  013777  030306  162352        MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
017526  052777  040040  162306        BIS     #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
                                ; -- END 0 MACRO ------------------
017534  005005                        CLR     R5
017536  032777  100000  162276  4$:   BIT     #TRDY,@DZCSR
017544  001404                        BEQ     5$
017546  000240                        NOP
017550  000240                        NOP
017552  104007                        ERROR   7               ;*TRANSMITTER FAILED TO INTERRUPT
017554  000416                        BR      8$
017556  104414                5$:     DELAY
017560  005205                        INC     R5
017562  001365                        BNE     4$
017564  104003                        ERROR   3               ;*TRDY NOT SET!
017566  000411                        BR      8$
017570  022626                6$:     POP2SP                  ;REMOVE THE INTERRUPT FROM THE STACK
017572  042777  040000  162242        BIC     #TIE,@DZCSR     ;DON'T LET ANY MORE INTERRUPTS OCCUR
017600  106437  030310                MTPS    @#LESS1         ;MAKE CPU ONE LEVEL LOWER THAN DZ11
017604  000402                        BR      8$              ;RETURN TO THE NORMAL FLOW
017606  104012                7$:     ERROR   12              ;*RECEIVER SHOULD NOT INTERRUPT
017610  022626                        CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
017612  042777  040000  162222  8$:   BIC     #TIE,@DZCSR     ;RESET TRANSMITTER INTERRUPT ENABLE
                                                        ; -$INTSET-------------------------------
017620  012777  01772(  162250        MOV     #11$,@DZTIV     ;SET UP THE TRANSMITTER INTERRUPT VECTOR
```

```
4214  017626  012777  017726  162236          MOV     #12$,@DZRIV     ;SET UP THE RECEIVER INTERRUPT VECTOR
4215  017634  013777  030306  162232          MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
4216  017642  013777  030306  162230          MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
4217  017650  052777  000140  162164          BIS     #RIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
4218                                  ; -- END 0 MACRO ------------------
4219  017656  113777  001422  162202          MOVB    TDO,@DZTDR      ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
4220  017664  005005                          CLR     R5
4221  017666  032777  000200  162146  9$:     BIT     #RDONE,@DZCSR
4222  017674  001404                          BEQ     10$
4223  017676  000240                          NOP
4224  017700  000240                          NOP
4225  017702  104011                          ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
4226  017704  000413                          BR      13$
4227  017706  104414                  10$:    DELAY
4228  017710  005205                          INC     R5
4229  017712  001365                          BNE     9$
4230  017714  104004                          ERROR   4               ;*NO RX DONE! (NOT SET)
4231  017716  000406                          BR      13$             ;CONTINUE TEST
4232  017720  104010                  11$:    ERROR   10              ;*TRANSMITTER SHOULD NOT INTERRUPT
4233  017722  022626                          CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
4234  017724  000403                          BR      13$             ;CONT TEST
4235  017726  022626                  12$:    POP2SP                  ;REMOVE THE INTERRUPT FROM THE STACK
4236  017730  005077  162106                  CLR     @DZCSR          ;DON'T ALLOW ANY MORE INTERRUPTS
4237  017734                          13$:
4238  017734  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR (RESET)
4239                                  ; -- END 0 MACRO ------------------
4240                                  ; -- END 0 MACRO ------------------
4241
4242                                                  ; -$XZ-----------------------------
4243                                  ;************************ TEST 27 *******************************
4244                                  ;*THIS TEST VERIFIES THAT THE RECEIVER WILL
4245                                  ;*INTERRUPT BEFORE THE TRANSMITTER EVEN
4246                                  ;*THOUGH THE TRANSMITTER WAS ENABLED
4247                                  ;*FIRST.  SET PS TO LEVEL 7;
4248                                  ;*GET RDONE AND TRDY TO SET;
4249                                  ;*SET TX IE AND RX IE;
4250                                  ;*CLEAR PS AND EXPECT RX TO INTERRUPT FIRST
4251                                                  ; -$XZ-----------------------------
4252                                  ;:*  TEST 27
4253                                  ;;*************************************************************
4254  017736  000004          000027  TST27:  SCOPE
4255  017740  012737  000027  001122          MOV     #27,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
4256                                  ; -- END 0 MACRO ------------------
4257  017746  012737  020370  001360          MOV     #TST30,NEXT     ;POINT TO THE START OF THE NEXT TEST
4258                                                  ;$LINEUP---------------------------------
4259                                                  ; -$MRESET-------------------------------
4260  017754  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4261                                  ; -- END 0 MACRO ------------------
4262  017756  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
4263  017762  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
4264  017766  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
4265  017772  001402                          BEQ     2$              ;NO
4266  017774  010177  162052                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
4267  020000  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
4268  020002  106302                          ASLB    R2              ;GOT 'EM ALL ?
4269  020004  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
```

```
4270  020006  005037  001372                    CLR     SAVLIN          ;CLEAR LINE # INDICATOR
4271                                    ; -- END 0 MACRO ------------------
4272  020012  012777  020242  162052            MOV     #8$,@DZRIV      ;SETUP INTERRUPT STUFF
4273  020020  013777  030306  162046            MOV     DZPRT,@DZRIS    ;
4274  020026  012777  030332  162042            MOV     #12$,@DZTIV     ;
4275  020034  013777  030306  162036            MOV     DZPRT,@DZTIS    ;
4276  020042  052777  000040  161772            BIS     #MSENAB,@DZCSR
4277  020050  012702  000001                    MOV     #1,R2           ;LINE POINTER
4278  020054  030237  001364          3$:       BIT     R2,LINE         ;VALID LINE ?
4279  020060  001004                            BNE     4$
4280  020062  005237  001372                    INC     SAVLIN
4281  020066  106302                            ASLB    R2
4282  020070  000771                            BR      3$
4283  020072  106427  000340          4$:       MTPS    #PR7
4284  020076  000240                            NOP
4285  020100  000240                            NOP
4286  020102  110277  161750                    MOVB    R2,@DZTCR       ;SET TCR BIT
4287  020106  005777  161734                    TST     @DZRBUF         ;VALID DATA?
4288  020112  100001                            BPL     .+4             ;IT BETTER NOT BE SET
4289  020114  104017                            ERROR   17              ;DATA VALID SHOULD NOT BE SET
4290  020116  105777  161720          5$:       TSTB    @DZCSR          ;RECEIVER DONE ?
4291  020122  100001                            BPL     .+4
4292  020124  104020                            ERROR   20              ;RECEIVER DONE BIT SHOULD NOT BE SET
4293  020126  005005                            CLR     R5
4294  020130  005004                            CLR     R4
4295  020132  005777  161704          99$:      TST     @DZCSR          ;WAIT FOR TRDY
4296  020136  100404                            BMI     100$            ;BR IF READY
4297  020140  104414                            DELAY                   ;STALL TIME
4298  020142  005204                            INC     R4              ;
4299  020144  001372                            BNE     99$
4300  020146  104003                            ERROR   3               ;TRDY FAILED TO SET
4301  020150  105077  161712          100$:     CLRB    @DZTDR
4302  020154  005004                            CLR     R4
4303  020156  032777  000200  161656  6$:       BIT     #RDONE,@DZCSR
4304  020164  001004                            BNE     7$
4305  020166  104414                            DELAY
4306  020170  005204                            INC     R4
4307  020172  001371                            BNE     6$
4308  020174  104004                            ERROR   4               ;*RDONE FAILED TO SET!
4309  020176  005777  161640          7$:       TST     @DZCSR          ;TRANS DONE BIT = 1 ?
4310  020202  100401                            BMI     .+4             ;YES
4311  020204  104003                            ERROR   3               ;*NO   TRANS DONE FAILED TO SET
4312                                            ;NOW THAT BOTH TRANSMITTER AND RECEIVER DONE BIT =1
4313                                            ;SET INTERRUPT ENABLES AND WATCH THE FUR FLY
4314  020206  052777  040000  161626            BIS     #TIE,@DZCSR
4315  020214  052777  000100  161620            BIS     #RIE,@DZCSR
4316  020222  106427  000000                    MTPS    #0
4317  020226  000240                            NOP
4318  020230  000240                            NOP
4319  020232  104007                            ERROR   7               ;*TRANSMITTER FAILED TO INTERRUPT
4320  020234  104011                            ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
4321                                            ;CHECK BR LEVEL
4322  020236  000137  020336                    JMP     13$     ;GET OUT
4323
4324                                            ;RECEIVER INTERRUPT ROUTINE
4325  020242  017704  161600          8$:       MOV     @DZRBUF,R4              ;ACTUAL
```

```
4326  020246  010403                          MOV     R4,R3
4327  020250  000303                          SWAB    R3
4328  020252  042703  177770                  BIC     #^C<7>,R3       ;STRIP JUNK
4329                                          ; -$STAG----------
4330  020256  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
4331  020262  001406                          BEQ     11$             ;IF NOT, SKIP STAGGERED SETUP
4332
4333                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
4334
4335  020264  006203                          ASR     R3              ;GET THE LAST BIT INTO THE CARRY BIT
4336  020266  103402                          BCS     9$              ;IF IT IS SET, GO CLEAR IT
4337  020270  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
4338  020272  000401                          BR      10$             ;SKIP THE CLEARING
4339  020274  000241                  9$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
4340  020276  006103                  10$:    ROL     R3              ;GET THE NEW BIT BACK INTO R3
4341                                  ; -- END O MACRO ----------
4342  020300  020337  001372          11$:    CMP     R3,SAVLIN       ;IS THIS A VALID LINE
4343  020304  001401                          BEQ     .+4             ;YES
4344  020306  104015                          ERROR   15              ;*INVALID LINE
4345  020310  042704  177400                  BIC     #^C<377>,R4     ;STRIP JUNK
4346  020314  120504                          CMPB    R5,R4           ;DATA COMPARE ?
4347  020316  001401                          BEQ     .+4             ;YES
4348  020320  104005                          ERROR   5               ;*DATA DOES NOT COMPARE
4349  020322  040277  161530                  BIC     R2,@DZTCR       ;CLEAR TCR BIT
4350  020326  022626                          POP2SP                  ;REMOVE HE INTERRUPT VECTOR FROM THE STACK
4351  020330  000402                          BR      13$             ;GO GET OUT OF INTERRUPT MODE
4352                                          ;TRANSMITTER INTERRUPT SVC ROUTINE
4353  020332  104011                  12$:    ERROR   11              ;THE RECEIVER INTERRUPT FAILED
4354                                                                  ;TO OVERRIDE THE TRANSMITTER
4355  020334  022626                          POP2SP                  ;REMOVE THE INTERRUPT VECTOR FROM THE STACK
4356  020336  042777  040100  161476  13$:    BIC     #TIE!RIE,@DZCSR ;CLEAR INTERRUPT ENABLES
4357  020344  013777  002074  161520          MOV     DZRIS,@DZRIV    ;RESTORE TRAPCATCHER
4358  020352  005077  161516                  CLR     @DZRIS
4359  020356  013777  002100  161512          MOV     DZTIS,@DZTIV
4360  020364  005077  161510                  CLR     @DZTIS
4361                                          ; -$XZ----------
4362                                          ;********** TEST 30 **********
4363                                          ;*TEST TO VERIFY THAT 'RDONE DOES NOT SET
4364                                          ;*IF THE SCANNER IS DISABLED.
4365                                          ;*TURN ON SCANNER, WAIT FOR TRDY,
4366                                          ;*TURN OFF SCANNER, TRANSMIT A CHARACTER
4367                                          ;*'RDONE SHOULD NOT SET.
4368                                          ; -$XZ----------
4369                                  ;:*  TEST 30
4370                                  ;;**********
4371  020370  000004                  TST30:  SCOPE
4372  020372  012737  000030  001122          MOV     #30,@TSTNM      ;LOAD THE NUMBER OF THIS TEST
4373                                  ; -- END O MACRO ----------
4374  020400  012737  020556  001360          MOV     #TST31,NEXT     ;POINT TO THE START OF THE NEXT TEST
4375                                          ;$LINEUP----------
4376                                          ; -$MRESET----------
4377  020406  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4378                                  ; -- END O MACRO ----------
4379  020410  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
4380  020414  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
4381  020420  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
```

```
4382  020424  001402                          BEQ     2$                  ;NO
4383  020426  010177  161420                  MOV     R1,@DZLPR           ;SET UP LINE PARAMETERS
4384  020432  005201                  2$:     INC     R1                  ;POSITION POINTER TO THE NEXT LINE
4385  020434  106302                          ASLB    R2                  ;GOT 'EM ALL ?
4386  020436  103370                          BCC     1$                  ;IF NO, GO SET UP THE NEXT LINE
4387  020440  005037  001372                  CLR     SAVLIN              ;CLEAR LINE # INDICATOR
4388                                  ; -- END O MACRO ----------------
4389  020444  052777  000040  161370          BIS     #MSENAB,@DZCSR      ;TURN ON SCANNER
4390  020452  012702  000001                  MOV     #1,     R2          ;INIT LINE COUNTER
4391  020456  030237  001364          3$:     BIT     R2,     LINE        ;FIND A VALID LINE
4392  020462  001004                          BNE     4$                  ;IF WE FOUND ONE GO TO TEST
4393  020464  005237  001372                  INC     SAVLIN              ;IF NOT
4394  020470  106302                          ASLB    R2                  ;KEEP LOOKING
4395  020472  000771                          BR      3$
4396  020474  110277  161356          4$:     MOVB    R2,     @DZTCR      ;SET TCR BIT
4397  020500  005005                          CLR     R5
4398  020502  005777  161334          5$:     TST     @DZCSR              ;IS TRDY SET
4399  020506  100404                          BMI     6$                  ;CON'T TESTING IF IT IS
4400  020510  104414                          DELAY                       ;IF IT NOT WAIT A WHILE
4401  020512  005205                          INC     R5
4402  020514  001372                          BNE     5$
4403  020516  104003                          ERROR   3                   ;WE WAITED LONG ENOUGH-ERROR
4404  020520  042777  000040  161314  6$:     BIC     #MSENAB, @DZCSR     ;TURN OFF SCANNER
4405  020526  105077  161334                  CLRB    @DZTDR              ;TRANSMIT A CHARACTER
4406  020532  005005                          CLR     R5                  ;CLEAR COUNTER
4407  020534  104414                  7$:     DELAY                       ;WAIT SUFFICIENT TIME FOR
4408  020536  005205                          INC     R5                  ;RDONE TO SET
4409  020540  001375                          BNE     7$
4410  020542  032777  000200  161272          BIT     #RDONE, @DZCSR      ;RDONE SET
4411  020550  001401                          BEQ     8$                  ;IT SHOULDN'T BE-CONTINUE
4412  020552  104020                          ERROR   20                  ;IF IT IS THERE'S AN ERROR
4413  020554  104400                  8$:     ADVANCE
4414                                                                      ; -$XZ-----------------------------
4415                                          ;************************ TEST 31 ******************************
4416                                          ;*THIS TEST VERIFIES OVERRUN AND SILO ALARM
4417                                          ;*ONE LINE AT A TIME  - BASED UPON VALID LINES
4418                                          ;*AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
4419                                          ;*TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN
4420                                          ;*EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
4421                                          ;*SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
4422                                          ;*CHAR PULLED OUT OUT THE SILO.
4423                                          ;*USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
4424                                          ;*ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
4425                                          ;*USED TO SCOPE SILO ALARM PULSES, ETC.
4426                                                                      ; -$XZ-----------------------------
4427                                  ;:*  TEST 31
4428                                  ;;*************************************************************
4429  020556  000004                  TST31:  SCOPE
4430  020560  012737  000031  001122          MOV     #31,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
4431                                  ; -- END O MACRO ------------------
4432  020566  012737  021304  001360          MOV     #TST32,NEXT         ;POINT TO THE START OF THE NEXT TEST
4433  020574  012737  021210  001362          MOV     #10$,LOCK           ;SET FOR LOOP
4434                                                                      ;$LINEUP---------------------------
4435                                                                      ; -$MRESET-------------------------
4436  020602  104417                          DCLASM                      ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4437                                  ; -- END O MACRO ------------------
```

```
4438  020604  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
4439  020610  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
4440  020614  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
4441  020620  001402                          BEQ     2$              ;NO
4442  020622  010177  161224                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
4443  020626  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
4444  020630  106302                          ASLB    R2              ;GOT 'EM ALL ?
4445  020632  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
4446  020634  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
4447                                  ; -- END 0 MACRO ----------------
4448  020640  012700  001422                  MOV     #TDO,R0         ;POINT TO THE DATA AREA
4449  020644  005020                          CLR     (R0)+           ;CLEAR A DATA WORD
4450  020646  022700  001462                  CMP     #STOP,R0        ;FINISHED ?
4451  020652  001374                          BNE     .-6             ;NO
4452  020654  005000                          CLR     R0              ;CLEAR OFFSET
4453  020656  012702  000001                  MOV     #1,R2           ;LINE POINTER
4454  020662  052777  010040  161152          BIS     #MSENAB!SILOEN,@DZCSR   ;START SCANNER & SET SILO ENABLE
4455  020670  030237  001364          3$:     BIT     R2,LINE         ;VALID LINE?
4456  020674  001002                          BNE     .+6             ;YES
4457  020676  000137  021172                  JMP     22$             ;TRY NEXT LINE
4458  020702  013700  001372                  MOV     SAVLIN,R0       ;MAKE OFFSET
4459  020706  006300                          ASL     R0              ;MAKE POWER OF TWO
4460  020710  010277  161142                  MOV     R2,@DZTCR       ;SET TCR BIT
4461  020714  105777  161122          4$:     TSTB    @DZCSR          ;REC DONE = 1 ?
4462  020720  100001                          BPL     .+4
4463  020722  104020                          ERROR   20              ;REC DONE SHOULD NOT = 1
4464  020724  005003                          CLR     R3              ;SET CHARACTER COUNT
4465  020726  005004                  5$:     CLR     R4
4466  020730  032777  100000  161104  6$:     BIT     #TRDY,@DZCSR
4467  020736  001004                          BNE     7$
4468  020740  104414                          DELAY
4469  020742  105204                          INCB    R4
4470  020744  001371                          BNE     6$
4471  020746  104003                          ERROR   3               ;#TRDY FAILED TO SET
4472  020750  116077  001422  161110  7$:     MOVB    TDO(R0),@DZTDR  ;LOAD A CHARACTER
4473  020756  005260  001422                  INC     TDO(R0)         ;SET UP NEXT CHARACTER
4474  020762  020327  000017                  CMP     R3,#15.         ;16 CHARACTERS ?
4475  020766  103006                          BHIS    8$
4476  020770  032777  020000  161044          BIT     #SILOAL,@DZCSR  ;SILO ALARM = 0 ?
4477  020776  001401                          BEQ     .+4             ;YES
4478  021000  104013                          ERROR   13              ;#SILO ALARM SHOULD NOT = 1
4479                                                                  ;UNTIL 16. DATA CHARACTERS
4480  021002  000411                          BR      10$
4481  021004  005004                  8$:     CLR     R4
4482  021006  032777  020000  161026  9$:     BIT     #SILOAL,@DZCSR
4483  021014  001004                          BNE     10$
4484  021016  104414                          DELAY
4485  021020  005204                          INC     R4
4486  021022  001371                          BNE     9$
4487  021024  104014                          ERROR   14              ;#SILO ALARM FAILED TO SET!
4488                                                                  ;SILO ALARM SHOULD =1 AFTER 16.
4489                                                                  ;DATA CHARACTERS
4490  021026  005203                  10$:    INC     R3              ;INC CHAR COUNT
4491  021030  022703  000102                  CMP     #66.,R3         ;FINISHED SENDING CHARACTERS ?
4492  021034  001334                          BNE     5$              ;NO
4493  021036  005004                          CLR     R4
```

```
4494  021040  104414                          DELAY
4495  021042  105204                          INCB    R4
4496  021044  001375                          BNE     .-4
4497                                          ;NOW LETS READ THE SILO
4498  021046  013705  001372                  MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
4499                                          ; -$STAG-----------------------------------
4500  021052  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
4501  021056  001406                          BEQ     13$             ;IF NOT, SKIP STAGGERED SETUP
4502
4503                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
4504
4505  021060  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
4506  021062  103402                          BCS     11$             ;IF IT IS SET, GO CLEAR IT
4507  021064  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
4508  021066  000401                          BR      12$             ;SKIP THE CLEARING
4509  021070  000241                  11$:    CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
4510  021072  006105                  12$:    ROL     R5              ;GET THE NEW BIT BACK INTO R5
4511                                  ; -- END 0 MACRO ------------------
4512  021074  000305                  13$:    SWAB    R5              ;PUT IN UPPER BYTE
4513  021076  052705  100000                  BIS     #DVALID,R5      ;ADD DATA VALID
4514  021102  017704  160740          14$:    MOV     @DZRBUF,R4      ;ACTUAL
4515  021106  020405                          CMP     R4,R5           ;ACTUAL VS. EXPECTED
4516  021110  001401                          BEQ     15$             ;YES
4517  021112  104006                          ERROR   6               ;+DATA/CONTENTS DID NOT COMPARE
4518  021114  032777  020000  160720  15$:    BIT     #SILOAL,@DZCSR  ;SILO ALARM= 0 ?
4519  021122  001401                          BEQ     16$             ;YES
4520  021124  104016                          ERROR   16              ;READING DZRBUF DID NOT CLEAR SILO ALARM
4521  021126  005205                  16$:    INC     R5              ;UP CHARACTER
4522  021130  120527  000077                  CMPB    R5,#63.         ;LAST SILO CHAR ?....64TH CHAR
4523  021134  101762                          BLOS    14$
4524  021136  005205                          INC     R5              ;ADD 1 MORE FOR THE CLOBBERED CHAR
4525  021140  052705  040000                  BIS     #OVRRUN,R5      ;ADD OVERRUN TO EXPECTED
4526  021144  120527  000101                  CMPB    R5,#65.         ;LAST CHARACTER ?
4527  021150  001754                          BEQ     14$
4528  021152  017704  160670                  MOV     @DZRBUF,R4      ;FOR GOOD MEASURE
4529  021156  005704                          TST     R4              ;DATA VALID SHOULD = 0
4530  021160  100001                          BPL     17$             ;YES
4531  021162  104017                          ERROR   17              ;DATA VALID SHOULD = 0
4532  021164  040277  160666          17$:    BIC     R2,@DZTCR       ;CLR TCR BIT
4533  021170  104401                          SCOP1                   ;LOOP?
4534  021172  005237  001372          22$:    INC     SAVLIN          ;INC EXPECTED LINE
4535  021176  106302                          ASLB    R2              ;NEXT LINE
4536  021200  103402                          BCS     .+6             ;NO
4537  021202  000137  020670                  JMP     3$              ;YES
4538  021206  104400                          ADVANCE                 ;GO TO NEXT TEST
4539
4540                                          ;TIGHT SCOPE LOOP FOR THIS TEST. SENDS 20. CHARACTERS
4541                                          ;ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
4542                                          ;USED TO SCOPE SILO ALARM PULSES, ETC.
4543
4544  021210  052777  010040  160624  18$:    BIS     #MSENAB!SILOEN,@DZCSR   ;SETUP DEVICE
4545  021216  012777  021274  160652          MOV     #20$,@DZTIV     ;SETUP TRANSMITTER VECTOR
4546  021224  012~37  000024  001216          MOV     #20.,$TMPO      ;TEMPORARY COUNT OF CHARACTER BURST
4547  021232  050277  160620                  BIS     R2,@DZTCR       ;ENABLE LINE
4548  021236  052777  040000  160576          BIS     #TIE,@DZCSR     ;ENABLE INTERRUPTS
4549  021244  106427  000000                  MTPS    #0              ;LOWER PRIORITY
```

```
 4550  021250  000001                        19$:    WAIT                    ;ALLOW INTERRUPTS
 4551  021252  005337  001216                        DEC     @TMPO           ;REDUCE COUNT. ALL CHARACTERS SENT?
 4552  021256  001374                              BNE     19$             ;IF NO, WAIT FOR MORE
 4553  021260  042777  050040  160554              BIC     #SILOEN!MSENAB!TIE,@DZCSR ;RESET SILO COUNTER, CLEAR STROBE
 4554  021266  104401                              SCOP1                   ;LOOP AGAIN?
 4555  021270  000137  021164                      JMP     17$             ;IF NOT, RETURN TO WHERE YOU LEFT OFF
 4556  021274  112777  000252  160564  20$:       MOVB    #252,@DZTDR     ;SEND A CHARACTER
 4557  021302  000002                              RTI                     ;ALLOW MORE CHARACTERS TO COME
 4558                                                                      ; -$XZ-----------------------------
 4559                                              ;********************** TEST 32 ******************************
 4560                                              ;*THIS TEST THAT "SILO ENABLE" WILL INHIBIT
 4561                                              ;*RECEIVER INTERRUPTS AND THAT ON THE
 4562                                              ;*16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
 4563                                              ;*INTERRUPT WITH "RIE" SET.
 4564                                              ;*THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
 4565                                                                      ; -$XZ-----------------------------
 4566                                              ;:*  TEST 32
 4567                                              ;;**************************************************************
 4568  021304  000004                        TST32:  SCOPE
 4569  021306  012737  000032  001122                MOV     #32,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
 4570                                              ; -- END 0 MACRO ------------------
 4571  021314  012737  021666  001360                MOV     #TST33,NEXT     ;POINT TO THE START OF THE NEXT TEST
 4572  021322  012737  021410  001362                MOV     #3$,LOCK        ;SET FOR LOOP
 4573                                                                      ;$LINEUP--------------------------------
 4574                                                                      ; -$MRESET------------------------------
 4575  021330  104417                              DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
 4576                                              ; -- END 0 MACRO ------------------
 4577  021332  013701  001366                      MOV     PAR,R1          ;PICK UP PARAMETERS
 4578  021336  012702  000001                      MOV     #1,R2           ;PICK UP INIT POINTER
 4579  021342  030237  001364              1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
 4580  021346  001402                              BEQ     2$              ;NO
 4581  021350  010177  160476                      MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
 4582  021354  005201                        2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
 4583  021356  106302                              ASLB    R2              ;GOT 'EM ALL ?
 4584  021360  103370                              BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
 4585  021362  005037  001372                      CLR     SAVLIN          ;CLEAR LINE # INDICATOR
 4586                                              ; -- END 0 MACRO ------------------
 4587  021366  012700  001422                      MOV     #TDO,R0         ;POINT TO THE DATA AREA
 4588  021372  005020                              CLR     (R0)+           ;CLEAR A DATA WORD
 4589  021374  022700  001462                      CMP     #STOP,R0        ;FINISHED ?
 4590  021400  001374                              BNE     .-6             ;NO
 4591  021402  005000                              CLR     R0              ;CLEAR OFFSET
 4592  021404  012702  000001                      MOV     #1,R2           ;LINE POINTER
 4593  021410  012777  021630  160454  3$:        MOV     #11$,@DZRIV     ;SET FOR UNEXPECTED INTER.
 4594  021416  012777  000340  160450              MOV     #PR7,@DZRIS     ;SET PRIO.
 4595  021424  052777  010140  160410              BIS     #MSENAB!SILOEN!RIE,@DZCSR
 4596                                                                      ;START SCANNER & SET SILO ENABLE
 4597  021432  030237  001364                      BIT     R2,LINE         ;VALID LINE?
 4598  021436  001002                              BNE     .+6             ;YES
 4599  021440  000137  021646                      JMP     22$             ;TRY NEXT LINE
 4600  021444  005777  160376                      TST     @DZRBUF         ;EMPTY THE SILO
 4601  021450  100775                              BMI     .-4             ;BR IF DATA VALID IS SET!
 4602  021452  106427  000000                      MTPS    #0              ;SET PROCESSOR PRIORITY TO 0
 4603  021456  013700  001372                      MOV     SAVLIN,R0       ;MAKE OFFSET
 4604  021462  006300                              ASL     R0              ;MAKE POWER OF TWO
 4605  021464  010277  160366                      MOV     R2,@DZTCR       ;SET TCR BIT
```

```
 4606  021470  005004                    5$:     CLR     R4
 4607  021472  032777  100000  160342    6$:     BIT     #TRDY,@DZCSR
 4608  021500  001004                            BNE     7$
 4609  021502  104414                            DELAY
 4610  021504  005204                            INC     R4
 4611  021506  001371                            BNE     6$
 4612  021510  104003                            ERROR   3               ;*TRDY FAILED TO SET
 4613  021512  116077  001422  160346    7$:     MOVB    TDO(R0),@DZTDR  ;LOAD A CHARACTER
 4614  021520  005260  001422                    INC     TDO(R0)         ;SET UP NEXT CHARACTER
 4615  021524  022760  000017  001422            CMP     #15.,TDO(R0)    ;15 CHARS YET?
 4616  021532  001406                            BEQ     8$
 4617  021534  032777  020000  160300            BIT     #SILOAL,@DZCSR  ;SILO ALARM = 0 ?
 4618  021542  001401                            BEQ     .+4             ;YES
 4619  021544  104013                            ERROR   13              ;*SILO ALARM SHOULD NOT = 1
 4620                                                                    ;UNTIL 16. DATA CHARACTERS
 4621  021546  000750                            BR      5$
 4622  021550  012777  021636  160314    8$:     MOV     #12$,@DZRIV     ;SET NEW VECTOR
 4623  021556  032777  100000  160256            BIT     #TRDY,@DZCSR    ;READY FOR 16TH CHAR
 4624  021564  001774                            BEQ     .-6
 4625  021566  016077  001422  160272            MOV     TDO(R0),@DZTDR  ;LOAD THE 16TH CHAR.
 4626  021574  005004                            CLR     R4
 4627  021576  032777  020000  160236    9$:     BIT     #SILOAL,@DZCSR
 4628  021604  001005                            BNE     10$
 4629  021606  104414                            DELAY
 4630  021610  005204                            INC     R4
 4631  021612  001371                            BNE     9$
 4632  021614  104014                            ERROR   14              ;*SILO ALARM FAILED TO SET!
 4633  021616  000410                            BR      17$             ;SILO ALARM SHOULD =1 AFTER 16.
 4634                                                                    ;DATA CHARACTERS
 4635  021620  000240                   10$:     NOP                     ;STALL
 4636  021622  000240                            NOP
 4637  021624  104000                            ERROR                   ;SILO ALARM NOT INTERRUPTING.
 4638  021626  000404                            BR      17$             ;CONTINUE TEST.
 4639  021630  022626                   11$:     CMP     (SP)+,(SP)+     ;FAKE RTI
 4640  021632  104012                            ERROR   12              ;RX SHOULD NOT INTERRUPT
 4641  021634  000401                            BR      17$             ;CONTINUE
 4642  021636  022626                   12$:     CMP     (SP)+,(SP)+     ;GOOD INTERRUPT TO HERE.
 4643  021640  040277  160212           17$:     BIC     R2,@DZTCR       ;CLR TCR BIT
 4644  021644  104401                            SCOP1                   ;LOOP?
 4645  021646  005237  001372           22$:     INC     SAVLIN          ;INC EXPECTED LINE
 4646  021652  106302                            ASLB    R2              ;NEXT LINE
 4647  021654  103402                            BCS     .+6             ;NO
 4648  021656  000137  021410                    JMP     3$              ;YES
 4649  021662  005037  001362                    CLR     LOCK            ;CLEAR TIGHT LOOP FOR NEXT TEST
```

```
4650                                                    ; -$XZ-----------------------------
4651                                            ;********************** TEST 33 ******************************
4652                                            ;*THIS TEST RUNS ALL LINES FULL BORE
4653                                            ;*BASED UPON QUALIFIED LINES
4654                                            ;*..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
4655                                            ;*TRANSMITTER
4656                                                    ; -$XZ-----------------------------
4657                                    ;:*  TEST 33
4658                                    ;;**************************************************************
4659  021666  000004                    TST33:  SCOPE
4660  021670  012737  000033  001122            MOV     #33,@#TSTNM     ;LOAD THE NUMBER OF THIS TEST
4661                                    ; -- END 0 MACRO ------------------
4662  021676  012737  022474  001360            MOV     #TST34,NEXT     ;POINT TO THE START OF THE NEXT TEST
4663                                                    ; -$MRESET-----------------------------
4664  021704  104417                            DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4665                                    ; -- END 0 MACRO ------------------
4666  021706  013737  001364  022472            MOV     LINE,RXTCR      ;SET IMAGE OF TCR BITS
4667  021714  013701  001366            RSTART: MOV     PAR,R1          ;PICK UP PARAMETER
4668  021720  012700  000001                    MOV     #1,R0           ;;PICK UP INIT POINTER
4669  021724  030037  001364            INIT:   BIT     R0,LINE         ;SHOULD THIS LINE BE SET UP
4670  021730  001402                            BEQ     1@              ;NO
4671  021732  010177  160114                    MOV     R1,@#DZLPR      ;SET UP LINE PARAM REGISTER
4672  021736  005201                    1@:     INC     R1
4673  021740  106300                            ASLB    R0              ;GOT 'EM ALL ?
4674  021742  103370                            BCC     INIT            ;NO
4675  021744  012700  001422                    MOV     #TDO,R0         ;CLEAR TRANS DATA POINTER & REC POINTERS
4676  021750  005020                    INIT1:  CLR     (R0)+
4677  021752  022700  001462                    CMP     #STOP,R0                ;FINISHED ?
4678  021756  001374                            BNE     INIT1           ;NO, CONTINUE CLEARING
4679  021760  012777  022214  160104            MOV     #RXSVC,@#DZRIV  ;SET UP REC INTR VECTOR
4680  021766  012777  000340  160100            MOV     #PR7,@#DZRIS    ;STATUS
4681  021774  012777  022116  160074            MOV     #TXSVC,@#DZTIV  ;SET UP TRANS INTR VECTOR
4682  022002  012777  000340  160070            MOV     #PR7,@#DZTIS    ;STATUS
4683  022010  052777  000100  160024            BIS     #RIE,@#DZCSR            ;SET REC INTR ENABLE
4684  022016  052777  040000  160016            BIS     #TIE,@#DZCSR            ;SET TRANS INTR ENABLE
4685  022024  052777  000040  160010            BIS     #MSENAB,@#DZCSR ;SET MASTER SCAN ENABLE
4686  022032  113777  001364  160016            MOVB    LINE,@#DZTCR    ;SET TCR BITS...UP UP AND AWAY !
4687  022040  106437  030310                    MTPS    @#LESS1                 ;ALLOW INTERRUPTS
4688
4689
4690  022044  005037  022114            SNAP:   CLR     66@
4691  022050  013727  007?14            67@:    MOV     DLYCNT,(PC)+    ;SET FOR DELAY
4692  022054  000000                    68@:    0
4693  022056  005337  022054                    DEC     68@
4694  022062  001375                            BNE     .-4
4695  022064  105737  022472                    TSTB    RXTCR           ;WAIT FOR ALL RECIEVERS TO FINISH
4696  022070  001002                            BNE     3@              ;
4697  022072  000137  022372                    JMP     OUT
4698  022076  005237  022114            3@:     INC     66@
4699  022102  001362                            BNE     67@
4700  022104  104007                            ERROR   7               ;*TRANSMITTER FAILED TO INTERRUPT
4701  022106  104011                            ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
4702  022110  000137  022444                    JMP     FINI
4703  022114  000000                    66@:    0
4704
4705                                    ;TRANS INTR SVC ROUTINE
```

```
4706  022116  005777  157720          TXSVC:  TST     @DZCSR          ;TRANS INTR ?
4707  022122  100401                          BMI     .+4
4708  022124  104003                          ERROR   3               ;*TRANSMITTER FAILED
4709  022126  117703  157712                  MOVB    @#DZCSR,R3      ;SAVE IT
4710                                  ;NOW TEST FOR LINE # ETC
4711  022132  042703  177770                  BIC     #^C<7>,R3       ;STRIP JUNK
4712  022136  010304                          MOV     R3,R4           ;SAVE
4713  022140  010337  001372                  MOV     R3,SAVLIN       ;ADJUST LOCATION FOR ERROR PRINTOUT
4714  022144  012702  000001                  MOV     #1,R2           ;SET UP POSITION POINTER
4715  022150  105303                  3$:     DECB    R3              ;IS IT THIS LINE ?
4716  022152  100402                          BMI     4$              ;YES
4717  022154  006302                          ASL     R2              ;UP THE LINE #
4718  022156  000774                          BR      3$              ;GO 'ROUND AGAIN
4719  022160  030237  001364          4$:     BIT     R2,LINE         ;VALID LINE?
4720  022164  001001                          BNE     .+4             ;YES
4721  022166  104010                          ERROR   10              ;NO,INVALID LINE!!!!
4722  022170  006304                          ASL     R4              ;MAKE POWER OF 2
4723  022172  116477  001422  157666          MOVB    TDO(R4),@DZTDR  ;LOAD CHARACTER
4724  022200  105264  001422                  INCB    TDO(R4)         ;SET UP NEXT CHARACTER
4725  022204  001002                          BNE     5$              ;LAST CHARACTER ?
4726  022206  040277  157644                  BIC     R2,@DZTCR       ;YES ,CLEAR TCR BIT
4727  022212  000002                  5$:     RTI
4728
4729
4730                                  ;REC INTR SVC ROUTINE
4731  022214  105777  157622          RXSVC:  TSTB    @DZCSR          ;REC DONE ?
4732  022220  100401                          BMI     .+4             ;YES
4733  022222  104004                          ERROR   4               ;FALSE INTERRUPT
4734  022224  017704  157616                  MOV     @DZRBUF,R4      ;SAVE IT
4735  022230  010403                          MOV     R4,R3
4736  022232  000303                          SWAB    R3
4737  022234  042703  177770                  BIC     #^C<7>,R3       ;STRIP JUNK
4738  022240  010337  001372                  MOV     R3,SAVLIN       ;SAVE LINE NUMBER
4739  022244  032777  020000  157570          BIT     #SILOAL,@DZCSR  ;SILO ALARM?
4740  022252  001401                          BEQ     .+4             ;NO
4741  022254  104000                          ERROR                   ;SILO ALARM SHOULD NOT =1
4742  022256  005704                          TST     R4              ;DATA VALID SET?
4743  022260  100401                          BMI     .+4             ;YES
4744  022262  104023                          ERROR   23              ;YOU LOSE  ...DATA VALID WAS'NT SET
4745  022264  032704  070000                  BIT     #OVRRUN!FRMERR!PARER,R4
4746  022270  001401                          BEQ     .+4
4747  022272  104000                          ERROR                   ;RECEIVER ERROR FLAG/S WERE SET
4748  022274  012702  000001                  MOV     #1,R2           ;SET UP POSITION POINTER
4749  022300  105303                  5$:     DECB    R3
4750  022302  100402                          BMI     6$
4751  022304  006302                          ASL     R2              ;RE POSITION POINTER
4752  022306  000774                          BR      5$              ;GO 'ROUND AGAIN
4753  022310  030237  001364          6$:     BIT     R2,LINE         ;LINE VALID ?
4754  022314  001001                          BNE     .+4             ;YES
4755  022316  104011                          ERROR   11              ;INVALID LINE #
4756  022320  013703  001372                  MOV     SAVLIN,R3       ;GET THE LINE NUMBER AGAIN
4757  022324  006303                          ASL     R3              ;USE R3 AS A POINTER IN THE DATA TABLE
4758  022326  126304  001442                  CMPB    TRO(R3),R4      ;DOES THE DATA CHARACTER COMPARE ?
4759  022332  001405                          BEQ     2$              ;YES
4760  022334  016305  001442                  MOV     TRO(R3),R5      ;SAVE EXPECTED
4761  022340  042704  177400                  BIC     #^C<377>,R4     ;CLEAR JUNK
```

```
                                                        ;R2 = LINE # BY BIT POSITION
                                                        ;R4 = ACTUAL DATA
                                                        ;R5 = EXPECTED DATA
022344  104005                          ERROR   5       ;*NO. DATA DOES NOT COMPARE
022346  005263  001442          2$:     INC     TRO(R3) ;SET UP FOR NEXT CHARACTER
022352  105763  001442                  TSTB    TRO(R3) ;ALL CHARS DONE?
022356  001002                          BNE     .+6
022360  040237  022472                  BIC     R2,RXTCR        ;ZERO LINE DONE INDICATOR.
022364  012716  022044                  MOV     #SNAP,(SP)      ;RESET THE BACKGROUND TIMING LOOP
022370  000002                          RTI

                                        ;FINISH UP ROUTINE
022372  106427  000340          OUT:    MTPS    #PR7                    ;STOP ALL INTERRUPTS
022376  104413                          DEVICE.CLR              ;CLEAR ALL INTERRUPTS AWAY
022400  005003                          CLR     R3
022402  005037  001372                  CLR     SAVLIN
022406  012702  000001                  MOV     #1,R2
022412  030237  001364          1$:     BIT     R2,LINE         ;VALID LINE ?
022416  001405                          BEQ     2$              ;NO
022420  022763  000400  001442          CMP     #400,TRO(R3)    ;RECEIVED A BINARY COUNT PATTERN ?
022426  001401                          BEQ     .+4             ;YES
022430  104027                          ERROR   27              ;THE LINE FAILED TO RECEIVE A FULL
                                                                ;BINARY COUNT PATTERN
022432  005237  001372          2$:     INC     SAVLIN          ;SET UP FOR NEXT LINE
022436  005723                          TST     (R3)+           ;ADD 2
022440  106302                          ASLB    R2              ;SET UP NEXT LINE POINTER
022442  103363                          BCC     1$              ;FINISHED ?
022444                          FINI:
022444  013777  002074  157420          MOV     DZRIS,@DZRIV    ;RESTORE TRAPCATCHER
022452  005077  157416                  CLR     @DZRIS
022456  013777  002100  157412          MOV     DZTIS,@DZTIV
022464  005077  157410                  CLR     @DZTIS
022470  104400                          ADVANCE                 ;GO TO THE NEXT TEST
022472  000000                  RXTCR:  0                       ;RX IMAGE OF TCR BITS

                                                        ; -$XZ-----------------------------
                                        ;************************ TEST 34 *******************************
                                        ;*DZ11 RELATIVE TIMING TEST.
                                        ;*EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
                                        ;*AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
                                        ;*WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
                                        ;*DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
                                        ;*THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
                                        ;* AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
                                        ;*PARAMETERS ARE:
                                        ;*  EIGHT BITS/PER/CHAR - TWO STOP BITS AT
                                        ;*       50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
                                        ;*       2400, 3600, 4800, 7200, 9600 BAUD.
                                        ;*  THEN, 9600 BAUD - TWO STOP BITS AT
                                        ;*       SEVEN, SIX, FIVE BITS/PER/CHAR.
                                        ;*AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
                                        ;*THE NEXT SELECTED LINE IS THE TESTED.
                                                        ; -$XZ-----------------------------
                                ;:*  TEST 34
```

```
4818                                    ;;*****************************************************************
4819  022474  000004                    TST34:  SCOPE
4820  022476  012737  000034  001122            MOV     #34,@TSTNM      ;LOAD THE NUMBER OF THIS TEST
4821                                    ; -- END 0 MACRO ------------------
4822  022504  012737  000002  001226            MOV     #2,@TIMES
4823  022512  012737  023204  001360            MOV     #TST35,NEXT     ;POINT TO THE START OF THE NEXT TEST
4824  022520  012737  022644  001362            MOV     #3$,LOCK        ;SET FOR LOOP
4825  022526  005037  025214                    CLR     OFFSET          ;RESET THIS VARIABLE
4826  022532  005037  001372                    CLR     SAVLIN          ;RESET LINE NUMBER INDICATOR
4827  022536  005037  001374                    CLR     XMTLIN          ;USE THIS WORD TO TELL WHAT LINE TRANSMITTED
4828  022542  012737  000001  001216            MOV     #1,@TMP0        ;USE @TMP0 AS A BIT POINTER
4829  022550  012737  010070  023202            MOV     #RCVON!SSO!EIGHT!TWOSTOP,7@ ;BUILD TEMPORARY PARAMETERS
4830  022556  033737  001216  001364  1$:       BIT     @TMP0,LINE      ;IS THIS LINE ACTIVE?
4831  022564  001027                            BNE     3$              ;IF SO, GO GET STARTED
4832  022566  012737  010070  023202  2$:       MOV     #RCVON!SSO!EIGHT!TWOSTOP,7@ ;LOAD PARAMETERS TEMPORARILY
4833  022574  012700  001422                    MOV     #TDO,R0         ;POINT TO THE DATA AREA
4834  022600  005020                            CLR     (R0)+           ;CLEAR A DATA WORD
4835  022602  022700  001462                    CMP     #STOP,R0        ;FINISHED ?
4836  022606  001374                            BNE     .-6             ;NO
4837  022610  005237  001374                    INC     XMTLIN          ;POINT TO THE NEXT LINE TO TRANSMIT
4838  022614  042737  000007  023202            BIC     #7,7@           ;MAKE SURE TEMPORARY PARAMETERS POINT TO 0
4839  022622  053737  001374  023202            BIS     XMTLIN,7@       ;ADD DESIRED LINE NUMBER
4840  022630  005037  025214                    CLR     OFFSET
4841  022634  106337  001216                    ASLB    @TMP0           ;POINT TO THE NEXT LINE
4842  022640  103346                            BCC     1$              ;PROCESS THE NEXT LINE
4843  022642  104400                            ADVANCE                 ;TEST TO SEE IF THIS TEST GETS REPEATED
4844  022644                          3$:
4845                                            ; -@MRESET-------------------------
4846  022644  104417                            DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4847                                    ; -- END 0 MACRO ------------------
4848  022646  042737  010000  023202            BIC     #RCVON,7@       ;ZERO PARAMTERS FOR TX LINE
4849  022654  013777  023202  157170            MOV     7@,@DZLPR       ;LOAD PARAMTERS FOR TX
4850  022662  005737  001370                    TST     MODE            ;STAGGERED?
4851  022666  100011                            BPL     100$            ;BR IF NO
4852  022670  000241                            CLC                     ;SET UP LINE
4853  022672  006037  023202                    ROR     7@              ;
4854  022676  103002                            BCC     98$             ;BR IF LINE WAS EVEN
4855  022700  000241                            CLC                     ;PREPARE TO MKE LINE EVEN
4856  022702  000401                            BR      99$             ;CONTINUE
4857  022704  000261                  98$:      SEC                     ;PREPARE TO MAKE LINE ODD
4858  022706  006137  023202          99$:      ROL     7@              ;SET ALTERED LINE
4859  022712  052737  010000  023202  100$:     BIS     #RCVON,7@       ;SET RX ON
4860  022720  013777  023202  157124            MOV     7@,@DZLPR       ;LOAD RX PARAMETERS
4861  022726  013737  023202  001372            MOV     7@,SAVLIN       ;ADJUST LOCATION FOR ERROR PRINTOUT
4862  022734  042737  177770  001372            BIC     #^C<7>,SAVLIN   ;STRIP JUNK
4863  022742  042737  000007  023202            BIC     #7,7@           ;CLEAR OLD LINE #
4864  022750  053737  001374  023202            BIS     XMTLIN,7@       ;SET LINE UP AGAIN
4865  022756  013737  023202  001400            MOV     7@,REGIST       ;SAVE PARAMETERS FOR PRINTOUT
4866  022764  012700  001422                    MOV     #TDO,R0         ;POINT TO THE DATA AREA
4867  022770  005020                            CLR     (R0)+           ;CLEAR A DATA WORD
4868  022772  022700  001462                    CMP     #STOP,R0        ;FINISHED ?
4869  022776  001374                            BNE     .-6             ;NO
4870  023000  005002                            CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
4871  023002  005003                            CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
4872  023004  005037  001220                    CLR     @TMP1           ;INITIALIZE THE TIMER
4873  023010  005037  001224                    CLR     @TMP3           ;INITIALIZE THESE BITS ALSO
```

```
 4874  023014  012737  000020  001376          MOV     #20,XMTCNT      ;SET HOW MANY CHARACTERS TO TRANSMIT
 4875  023022  012777  02463C  157046          MOV     #XMTSRV,@DZTIV
 4876  023030  012777  025002  157034          MOV     #RXISR1,@DZRIV
 4877  023036  013777  030306  157030          MOV     CZPRT,@DZRIS
 4878  023044  013777  030306  157026          MOV     DZPRT,@DZTIS
 4879  023052  113777  001216  156776          MOVB    $TMP0,@DZTCR    ;START THE VALID LINE
 4880  023060  052777  040140  156754          BIS     #TIE!RIE!MSENAB,@DZCSR
 4881  023066  106427  000000                  MTPS    #0              ;LOWER THE PRIORITY TO ALLOW INTERRUPTS
 4882  023072  032777  000100  156742  4$:     BIT     #RIE,@DZCSR     ;IS ROUTINE DONE?
 4883  023100  001407                          BEQ     5$              ;WHEN ALL IS DONE RX IE IS CLEARED IN ISR.
 4884  023102  005237  001220                  INC     $TMP1           ;COUNT TIME
 4885  023106  001371                          BNE     4$              ;CONTINUE TEST
 4886  023110  105237  001224                  INCB    $TMP3           ;DOUBLE COUNT
 4887  023114  001366                          BNE     4$              ;CONTINUE TEST
 4888  023116  104011                          ERROR   11              ;INTERRUPTS NOT FINISHED
 4889  023120  004737  007652          5$:     JSR     PC,SERV.G       ;<^G>?
 4890  023124  104401                          SCOP1                   ;LOOP?
 4891  023126  062737  000002  025214          ADD     #2,OFFSET
 4892  023134  013700  023202                  MOV     7$,R0
 4893  023140  042700  170377                  BIC     #^C<17*400>,R0
 4894  023144  022700  007000                  CMP     #<16*400>,R0
 4895  023150  001010                          BNE     6$
 4896  023152  032737  000030  023202          BIT     #BIT4+BIT3,7$
 4897  023160  001602                          BEQ     2$
 4898  023162  162737  000010  023202          SUB     #BIT3,7$
 4899  023170  000625                          BR      3$
 4900  023172  062737  000400  023202  6$:     ADD     #400,7$
 4901  023200  000621                          BR      3$
 4902  023202  000000                  7$:     0
 4903                                          ; -$PARTST----------------------------------
 4904                                          ; -$XZ--------------------------------------
 4905                                  ;************************ TEST 35 ********************************
 4906                                  ;* THIS TEST VERIFIES THAT EVEN PARITY WORKS
 4907                                  ;* FOR ALL ODD LINES SELECTED AND THAT ODD PARITY WORKS FOR ALL
 4908                                  ;* EVEN LINES SELECTED.
 4909                                  ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
 4910                                  ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
 4911                                  ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
 4912                                  ;*YOU ARE IN "STAGGERED" MODE.
 4913                                  ;*40(8) CHARS ARE USED FOR THIS TEST.
 4914                                  ;*ALL SELECTED LINES WILL BE ENABLED
 4915                                  ;*AT THE SAME TIME!
 4916                                          ; -$XZ--------------------------------------
 4917                                  ;:*  TEST 35
 4918                                  ;;*****************************************************************
 4919  023204  000004                  TST35:  SCOPE
 4920  023206  012737  000035  001122          MOV     #35,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
 4921                                  ; -- END 0 MACRO ------------------
 4922  023214  012737  023456  001360          MOV     #TST36,NEXT     ;POINT TO THE START OF THE NEXT TEST
 4923  023222  005737  001370                  TST     MODE            ;IS THIS STAGGERED MODE?
 4924  023226  100112                          BPL     6$              ;IF NOT, DON'T DO THIS TEST
 4925                                          ; -$MRESET----------------------------------
 4926  023230  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
 4927                                  ; -- END 0 MACRO ------------------
 4928  023232  013701  001366                  MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
 4929  023236  042701  000200                  BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
```

```
023242 052701 000100                 BIS     #PARITY,R1        ;MAKE SURE PARITY IS TURNED ON
023246 012702 000001                 MOV     #1,R2             ;USE R2 AS A LINE POINTER
023252 030237 001364         1$:     BIT     R2,LINE           ;IS THIS A VALID LINE?
023256 001411                        BEQ     3$                ;IF NOT, SKIP TO THE NEXT LINE
023260 032701 000001                 BIT     #BIT0,R1          ;IS THIS LINE AN ODD LINE?
023264 001002                        BNE     2$                ;IF IT'S ODD, USE EVEN PARITY
023266 052701 000200                 BIS     #ODDPAR,R1        ;IF IT'S EVEN, USE ODD PARITY
023272 010177 156554         2$:     MOV     R1,@DZLPR         ;LOAD THE LINE PARAMETER REGISTER
023276 042701 000200                 BIC     #ODDPAR,R1        ;SET UP THE NEXT PARITY TO EVEN
023302 005201                3$:     INC     R1                ;POINT TO THE NEXT LINE
023304 106302                        ASLB    R2                ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
023306 103361                        BCC     1$                ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
023310 005037 001372                 CLR     SAVLIN            ;CLEAR THE LINE NUMBER INDICATOR
023314 005002                        CLR     R2                ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
023316 005037 024632                 CLR     COUNT0            ; ;;2-10-84 ECB REV I
023322 005037 024634                 CLR     COUNT1            ; ;;2-10-84 ECB REV I
023326 005003                        CLR     R3                ;USE R3 TO COUNT TOT. NUMBER OF RECEPTIONS
023330 012737 000040 001376          MOV     #40,XMTCNT        ;TRANSMIT A BINARY COUNT PATTERN(00-40)
023336 012700 001422                 MOV     #TD0,R0           ;POINT TO THE DATA AREA
023342 005020                        CLR     (R0)+             ;CLEAR A DATA WORD
023344 022700 001462                 CMP     #STOP,R0          ;FINISHED ?
023350 001374                        BNE     .-6               ;NO
023352 005000                        CLR     R0                ;CLEAR OFFSET
                                                               ; -$INTSET-----------------------------
023354 012777 024636 156514          MOV     #XMTSRV,@DZTIV    ;SET UP THE TRANSMITTER INTERRUPT VECTOR
023362 012777 024460 156502          MOV     #PARESE,@DZRIV    ;SET UP THE RECEIVER INTERRUPT VECTOR
023370 013777 030306 156476          MOV     DZPRT,@DZRIS      ;SET THE INTERRUPT VECTOR STATUS
023376 013777 030306 156474          MOV     DZPRT,@DZTIS      ;SET TRANSMITTER INTERRUPT PRIORITY
023404 052777 040140 156430          BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
                             ; -- END 0 MACRO ===================
023412 113777 001364 156436          MOVB    LINE,@DZTCR       ;ENABLE ALL SELECTED LINES
023420 106427 000000                 MTPS    #0                ;ALLOW INTERRUPTS
023424 032777 000100 156410  5$:     BIT     #RIE,@DZCSR       ;WHEN RX DONE; RIE WILL =0
023432 001410                        BEQ     6$                ;BR IF ALL DONE
023434 104414                        DELAY                     ; ;; 2-10-84 ECB REV I
023436 005237 024632                 INC     COUNT0
023442 102770                        BVS     5$
023444 105237 024634                 INCB    COUNT1
023450 100365                        BPL     5$
023452 104011                        ERROR   11                ;*RX FAILED TO FINISH (INTERRUPT)
023454 104400                6$:     ADVANCE                   ;ADVANCE LOOP
                             ; -- END 0 MACRO ===================
                                                               ; -$PARTST-----------------------------------
                                                               ; -$XZ-----------------------------
                                     ;************************ TEST 36 ******************************
                                     ;*THIS TEST VERIFIES THAT ODD PARITY WORKS FOR ALL ODD LINES
                                     ;* SELECTED AND THAT EVEN PARITY WORKS FOR ALL EVEN LINES SELECTED
                                     ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
                                     ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
                                     ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
                                     ;*YOU ARE IN "STAGGERED" MODE.
                                     ;*40(8) CHARS ARE USED FOR THIS TEST.
                                     ;*ALL SELECTED LINES WILL BE ENABLED
                                     ;*AT THE SAME TIME!
                                                               ; -$XZ-----------------------------
                             ;:*  TEST 36
```

```
                                        ;;*******************************************************************
023456  000004                          TST36:  SCOPE
023460  012737  000036  001122                  MOV     #36,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
                                        ; -- END 0 MACRO ******************
023466  012737  023730  001360                  MOV     #TST37,NEXT     ;POINT TO THE START OF THE NEXT TEST
023474  005737  001370                          TST     MODE            ;IS THIS STAGGERED MODE?
023500  100112                                  BPL     6$              ;IF NOT, DON'T DO THIS TEST
                                                                        ; -$MRESET-**************************
023502  104417                                  DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
                                        ; -- END 0 MACRO ******************
023504  013701  001366                          MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
023510  042701  000200                          BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
023514  052701  000100                          BIS     #PARITY,R1      ;MAKE SURE PARITY IS TURNED ON
023520  012702  000001                          MOV     #1,R2           ;USE R2 AS A LINE POINTER
023524  030237  001364                  1$:     BIT     R2,LINE         ;IS THIS A VALID LINE?
023530  001411                                  BEQ     3$              ;IF NOT, SKIP TO THE NEXT LINE
023532  032701  000001                          BIT     #BIT0,R1        ;IS THIS LINE AN ODD LINE?
023536  001402                                  BEQ     2$              ;IF IT'S EVEN, USE EVEN PARITY
023540  052701  000200                          BIS     #ODDPAR,R1      ;IF IT'S ODD, USE ODD PARITY
023544  010177  156302                  2$:     MOV     R1,@DZLPR       ;LOAD THE LINE PARAMETER REGISTER
023550  042701  000200                          BIC     #ODDPAR,R1      ;SET UP THE NEXT PARITY TO EVEN
023554  005201                          3$:     INC     R1              ;POINT TO THE NEXT LINE
023556  106302                                  ASLB    R2              ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
023560  103361                                  BCC     1$              ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
023562  005037  001372                          CLR     SAVLIN          ;CLEAR THE LINE NUMBER INDICATOR
023566  005002                                  CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
023570  005037  024632                          CLR     COUNT0          ; ;;2-10-84 ECB REV I
023574  005037  024634                          CLR     COUNT1          ; ;;2-10-84 ECB REV I
023600  005003                                  CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
023602  012737  000040  001376                  MOV     #40,XMTCNT      ;TRANSMIT A BINARY COUNT PATTERN(00-40)
023610  012700  001422                          MOV     #TD0,R0         ;POINT TO THE DATA AREA
023614  005020                                  CLR     (R0)+           ;CLEAR A DATA WORD
023616  022700  001462                          CMP     #STOP,R0        ;FINISHED ?
023622  001374                                  BNE     .-6             ;NO
023624  005000                                  CLR     R0              ;CLEAR OFFSET
                                                                        ; -$INTSET-**************************
023626  012777  024636  156242                  MOV     #XMTSRV,@DZTIV  ;SET UP THE TRANSMITTER INTERRUPT VECTOR
023634  012777  024460  156230                  MOV     #PARESE,@DZRIV  ;SET UP THE RECEIVER INTERRUPT VECTOR
023642  013777  030306  156224                  MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
023650  013777  030306  156222                  MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
023656  052777  040140  156156                  BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
                                        ; -- END 0 MACRO ******************
023664  113777  001364  156164                  MOVB    LINE,@DZTCR     ;ENABLE ALL SELECTED LINES
023672  106427  000000                          MTPS    #0              ;ALLOW INTERRUPTS
023676  032777  000100  156136          5$:     BIT     #RIE,@DZCSR     ;WHEN RX DONE; RIE WILL =0
023704  001410                                  BEQ     6$              ;BR IF ALL DONE
023706  104414                                  DELAY                   ; ;; 2-10-84 ECB REV I
023710  005237  024632                          INC     COUNT0
023714  102770                                  BVS     5$
023716  105237  024634                          INCB    COUNT1
023722  100365                                  BPL     5$
023724  104011                                  ERROR   11              ;*RX FAILED TO FINISH (INTERRUPT)
023726  104400                          6$:     ADVANCE                 ;ADVANCE LOOP
                                        ; -- END 0 MACRO ******************
```

```
                                                                   ; -$XZ-------------------------------
                                             ;************************ TEST 37 ******************************
                                             ;*
                                             ;* ECB  21-FEB-84
                                             ;* BAUD RATE TIMING TEST.  THIS TEST ADDED IN ORDER TO TEST CRYSTAL
                                             ;* SPEEDS.  IT LOOKS FOR EITHER A KW11L OR KW11P.  IF EITHER IS AVAIL.
                                             ;* THEN THE TEST IS RAN.  IF NEITHER AVAILABLE, THEN TEST IS NOT RAN.
                                             ;*
                                             ;* KR   18-JUN-84
                                             ;* ADDED NUMBERS IN TTABLE AND NOW TEST ALL LINES AT ALL BAUD RATES
                                             ;* INSTEAD OF JUST LINE 0 AT ALL BAUD RATES.
                                             ;*
                                                                   ; -$XZ-------------------------------
                                      ;;*  TEST 37
                                      ;;**********************************************************
023730  000004                        TST37:  SCOPE
023732  012737  000037  001122                MOV     #37,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
                                      ; -- END 0 MACRO ----------------
023740  012737  004712  001360                MOV     #$EOP,NEXT       ;POINT TO THE END-OF-PASS HANDLER

023746  106427  000000                        MTPS    #0               ;DROP PRIORITY TO 0
023752  013700  000004                        MOV     @#4,R0           ;SAVE CONTENTS OF LOCATION 4.
023756  012737  024022  000004                MOV     #10$,@#4         ;SET TO TRAP IF NOT KW11L
023764  012701  177546                        MOV     #177546,R1       ;GET KW11L ADDRESS.
023770  012737  024076  000100                MOV     #CLKSRV,@#100    ;SET FOR INTERRUPT.
023776  012737  000340  000102                MOV     #340,@#102       ;PRORITY 7 ON INTERRUPT.
024004  005037  024106                        CLR     BCOUNT           ;OUR OWN CLOCK TICK COUNTER
024010  005037  024104                        CLR     CCOUNT           ;CHAR SENT COUNTER
024014  012711  000100                        MOV     #100,(R1)        ;THIS WILL CAUSE A BOMB TO OCCUR IF
                                                                       ;THE KW11L DOES NOT EXIST
                                                                       ;BUT IF ITS THERE, THEN WE'RE STARTED.
024020  000472                                BR      TCCONT           ;START TEST

024022  022626                        10$:    CMP     (SP)+,(SP)+      ;READJUST STACK
024024  012737  024066  000004                MOV     #20$,@#4         ;SET TO TRAP IF NO KW11P
024032  012737  024076  000104                MOV     #CLKSRV,@#104    ;SET FOR INTERRUPT.
024040  012737  000340  000106                MOV     #340,@#106       ;PROITY 7 ON INTERRUPT.
024046  012701  172540                        MOV     #172540,R1       ;KW11P ADDRESS.
024052  012761  177777  000002                MOV     #-1,2(R1)        ;SET THE # OF COUNTS FOR INTERRUPT (1)
                                                                       ;NOTE, IF KW11P DOES NOT EXIST, THEN
                                                                       ;WE WILL BOMB FROM THIS PLACE.
024060  012711  000133                        MOV     #133,(R1)        ;SET INTERRUPT ENABLE,UP COUNT,REPEATED
                                                                       ;INTERRUPT,LF CLOCK, AND RUN.
024064  000450                                BR      TCCONT           ;START TEST.

024066  022626                        20$:    CMP     (SP)+,(SP)+      ;READJUST STACK, NO CLOCKS AT ALL!
024070  010037  000004                        MOV     R0,@#4           ;RESTORE LOCATION 4
024074  104400                                ADVANCE                  ;GET OUT.

024076  005237  024104                CLKSRV: INC     CCOUNT           ;CLOCK INTERRUPTS TO HERE, UPDATE COUNT.
024102  000002                                RTI                      ;THEN EXIT.

024104  000000                        CCOUNT: .WORD   0                ;INCREMENTED BY CLKSRV. #CLOCK TICKS.
024106  000000                        BCOUNT: .WORD   0                ;USED BY TCCONT AS TIMER.
024110  000000                        DCOUNT: .WORD   0                ;USED BY TCCONT AS TIMER.
```

```
                                ;*
                                ; THESE NUMBER WHERE WORKING GREAT ON A PDP-11/24 RUNNING AT 60 HERTZ
                                ;
                                ;TTABLE:.WORD   5., 7.          ;COUNT OF CHARS OUT FOR 50 BAUD
                                ;       .WORD   7., 9.          ;COUNT OF CHARS OUT FOR 75 BAUD
                                ;       .WORD   11., 13.        ;COUNT OF CHARS OUT FOR 110 BAUD
                                ;       .WORD   13., 15.        ;COUNT OF CHARS OUT FOR 134.5 BAUD
                                ;       .WORD   15., 17.        ;COUNT OF CHARS OUT FOR 150 BAUD
                                ;
                                ;       .WORD   29., 32.        ;COUNT OF CHARS OUT FOR 300 BAUD
                                ;       .WORD   59., 62.        ;COUNT OF CHARS OUT FOR 600 BAUD
                                ;       .WORD   119., 122.      ;COUNT OF CHARS OUT FOR 1200 BAUD
                                ;       .WORD   179., 182.      ;COUNT OF CHARS OUT FOR 1800 BAUD
                                ;       .WORD   199., 202.      ;COUNT OF CHARS OUT FOR 2000 BAUD
                                ;
                                ;       .WORD   239., 242.      ;COUNT OF CHARS OUT FOR 2400 BAUD
                                ;       .WORD   359., 362.      ;COUNT OF CHARS OUT FOR 3600 BAUD
                                ;       .WORD   479., 482.      ;COUNT OF CHARS OUT FOR 4800 BAUD
                                ;       .WORD   719., 722.      ;COUNT OF CHARS OUT FOR 7200 BAUD
                                ;       .WORD   959., 962.      ;COUNT OF CHARS OUT FOR 9600 BAUD
                                ;-
                                ;*
                                ; THESE NUMBER WHERE MODIFIED SO THEY WOULD WORK ON A
                                ; 50 HERTZ MACHINE (TESTED ON PDP-11/70) AS WELL AS 60 HERTZ.
                                ; AT 9600 BAUD THERE IS ABOUT 17% ERROR.
                                ; THE LOWER THE BAUD RATE THE LESS THE ERROR.
                                ;-
                                ;-
024112  000005  000007       TTABLE: .WORD   5., 7.          ;COUNT OF CHARS OUT FOR 50 BAUD
024116  000007  000014               .WORD   7., 12.         ;COUNT OF CHARS OUT FOR 75 BAUD
024122  000013  000021               .WORD   11., 17.        ;COUNT OF CHARS OUT FOR 110 BAUD
024126  000015  000024               .WORD   13., 20.        ;COUNT OF CHARS OUT FOR 134.5 BAUD
024132  000017  000025               .WORD   15., 21.        ;COUNT OF CHARS OUT FOR 150 BAUD

024136  000035  000050               .WORD   29 , 40.        ;COUNT OF CHARS OUT FOR 300 BAUD
024142  000073  000114               .WORD   59., 76.        ;COUNT OF CHARS OUT FOR 600 BAUD
024146  000167  000224               .WORD   119., 148.      ;COUNT OF CHARS OUT FOR 1200 BAUD
024152  000263  000334               .WORD   179., 220.      ;COUNT OF CHARS OUT FOR 1800 BAUD
024156  000307  000364               .WORD   199., 244.      ;COUNT OF CHARS OUT FOR 2000 BAUD

024162  000357  000444               .WORD   239., 292.      ;COUNT OF CHARS OUT FOR 2400 BAUD
024166  000547  000664               .WORD   359., 436.      ;COUNT OF CHARS OUT FOR 3600 BAUD
024172  000737  001104               .WORD   479., 580.      ;COUNT OF CHARS OUT FOR 4800 BAUD
024176  001317  001544               .WORD   719., 868.      ;COUNT OF CHARS OUT FOR 7200 BAUD
024202  001677  002204               .WORD   959., 1156.     ;COUNT OF CHARS OUT FOR 9600 BAUD
        024206                  ENDTTB  .                       ;END OF TTABLE

                                ;
                                ; CHECK OUT THE CLOCK
                                ;

024206  010037  000004       TCCONT: MOV     R0,@#4          ;RESTORE LOCATION 4.
024212  005037  024104               CLR     CCOUNT          ;START WITH LINE CLOCK AT ZERO COUNT
024216  012737  177750  024106       MOV     #-30,BCOUNT     ;30-18MS LOOPS TO WAIT
```

```
 5152  024224  005737  024104          10$:    TST     CCOUNT          ;WAIT FOR COUNT TO GO TO 1 TO AVOID
 5153  024230  001010                          BNE     20$             ;USING THAT DANGEROUS FIRST COUNT!
 5154  024232  005237  024110                  INC     DCOUNT          ;BUT AS A SAFE GUARD,
 5155  024236  001372                          BNE     10$             ;DON'T HANG HERE
 5156  024240  005237  024106                  INC     BCOUNT
 5157  024244  001367                          BNE     10$
 5158  024246  005011                          CLR     (R1)            ;CLEAR CLOCK
 5159  024250  104400                          ADVANCE                 ;CLOCK NOT WORKING RIGHT- ABORT TEST
 5160                                                                  ;NOT THE FAULT OF THE DZ!
 5161
 5162                                  ;
 5163                                  ; START XMITTING CHARACTERS TO ALL 8 LINES AT ALL BAUD RATES.
 5164                                  ; IF THE NUMBER OF CHARACTERS TRANSMITTED IS BETWEEN THE
 5165                                  ; TWO NUMBERS IN TTABLE THEN THE LINE IS OK, ELSE ERROR
 5166                                  ;
 5167
 5168  024252  012777  000050  155562  20$:    MOV     #50,@DZCSR      ;SET MSE AND MAINT.
 5169  024260  012700  000030          30$:    MOV     #30,R0          ;SET BAUD RATE 50, 8 BIT CHAR LEN
 5170  024264  012703  000001                  MOV     #1,R3           ;HOLDS CURRENT LINE, START AT LINE 0
 5171
 5172  024270  110377  155562                  MOVB    R3,@DZTCR       ;ENABLE THE NEXT LINE TO XMIT
 5173  024274  012702  024112                  MOV     #TTABLE,R2      ;GET TABLE OF EXPECTED RESULTS
 5174
 5175  024300  010077  155546          40$:    MOV     R0,@DZLPR       ;UPDATE THE BAUD RATE AND NEXT LINE
 5176  024304  005037  024106                  CLR     BCOUNT          ;REUSE TIMER AS XMIT CHAR COUNTER
 5177  024310  005037  024104                  CLR     CCOUNT          ;CLEAR CLOCK INTERRUPT COUNTER
 5178
 5179  024314  022737  000074  024104  50$:    CMP     #60.,CCOUNT     ;DO THIS LOOP FOR 60 CLOCK TICKS
 5180  024322  001413                          BEQ     60$             ;EXIT WHEN THAT OCCURS.
 5181  024324  005777  155512                  TST     @DZCSR          ;READY TO XMIT A CHAR?
 5182  024330  100371                          BPL     50$             ;NO, THEN WAIT
 5183  024332  112777  000101  155526          MOVB    #101,@DZTDR     ;XMIT A CHAR
 5184  024340  005237  024106                  INC     BCOUNT          ;UPDATE XMIT COUNT, SHOULD NOT OVERFLOW.
 5185  024344  102363                          BVC     50$             ;IF NO OVERFLOW, CONTINUE
 5186  024346  005011                          CLR     (R1)            ;IF OVERFLOW, CLOCK COULD BE BAD, CLEAR IT
 5187  024350  104400                          ADVANCE                 ;ABORT THIS TEST IF CLOCK 10-2.'
 5188
 5189  024352  023712  024106          60$:    CMP     BCOUNT,(R2)     ;IF COUNT < LOW
 5190  024356  103422                          BLO     70$             ;       THEN ERROR
 5191  024360  023762  024106  000002          CMP     BCOUNT,2(R2)    ;IF COUNT > HI
 5192  024366  101016                          BHI     70$             ;       THEN ERROR
 5193  024370  062700  000400                  ADD     #400,R0         ;NEXT BAUD RATE
 5194  024374  062702  000004                  ADD     #4,R2           ;GET NEXT PAIR OF NUMBERS
 5195  024400  022702  024206                  CMP     #ENDTTB,R2      ;HIT END OF TTABLE?
 5196  024404  001335                          BNE     40$             ;NO, CONTINUE TESTING THIS LINE
 5197  024406  042700  007400                  BIC     #7400,R0        ;BRING THE BAUD RATE BACK TO 50
 5198  024412  005200                          INC     R0              ;AND GET NEXT LINE NUMBER
 5199  024414  106303                          ASLB    R3              ;MOVE ONTO THE NEXT LINE
 5200  024416  103324                          BCC     30$             ;IF NOT DONE WITH ALL LINES THEN CONTINUE
 5201  024420  005011                          CLR     (R1)            ;ELSE STOP CLOCK
 5202  024422  104400                          ADVANCE                 ;AND DO THE NEXT TEST
 5203
 5204                                  ;
 5205                                  ; ERROR - COUNTER NOT IN THE RANGE OF THE NUMBERS IN TTABLE
 5206                                  ;
 5207
```

```
5208  024424  005011                    70$:    CLR     (R1)              ;STOP CLOCK.
5209  024426  011237  024104                    MOV     (R2),CCOUNT       ;MOVE LOW COUNT FOR ERROR MESSAGE
5210  024432  016237  000002  024110            MOV     2(R2),DCOUNT      ;MOVE HIGH COUNT FOR ERROR MESSAGE
5211  024440  042700  177770                    BIC     #^C<7>,R0         ;STRIP EVERYTHING BUT LINE NUMBER
5212  024444  010037  001372                    MOV     R0,SAVLIN         ;MOVE LINE NUMBER FOR ERROR MESSAGE
5213  024450  104030                            ERROR   30                ;COUNT TOO HIGH OR LOW
5214  024452  004737  007652                    JSR     PC,SERV.G         ;<^G?>
5215  024456  104400                            ADVANCE                   ;DO NEXT TEST
5216
5217                                            ;
5218                                            ;RECEIVER SERVICE ROUTINE(PARITY TEST ONLY)
5219                                            ;
5220
5221  024460  017704  155362           PARESE: MOV     @DZRBUF,R4        ;GET THE CHARACTER
5222  024464  010401                            MOV     R4,R1             ;COPY THE RECEIVED INFORMATION
5223  024466  000301                            SWAB    R1                ;GET THE LINE NUMBER IN THE LOWER BYTE
5224  024470  042701  177770                    BIC     #^C<7>,R1         ;ISOLATE THE LINE NUMBER
5225  024474  010137  001372                    MOV     R1,SAVLIN         ;FILL LOC. FOR ERROR PRINTOUT
5226  024500  005704                            TST     R4                ;WAS DATA VALID?
5227  024502  100401                            BMI     10$               ;BRANCH IF YES
5228  024504  104023                            ERROR   23                ;ERROR - DATA VALID NOT SET!
5229  024506  006301                    10$:    ASL     R1                ;ALIGN IT ON A WORD BOUNDARY
5230  024510  032704  010000                    BIT     #PARER,R4         ;PARITY ERROR SHOULD BE SET. IS IT?
5231  024514  001013                            BNE     11$               ;IF SO, GO CHECK CHARACTER
5232  024516  013737  002046  001400            MOV     DZRBUF,REGIST     ;SET UP FOR THE ERROR MESSAGE
5233  024524  010405                            MOV     R4,R5
5234  024526  042705  000377                    BIC     #377,R5
5235  024532  156105  001442                    BISB    TRO(R1),R5        ;GET THE CORRECT CHARACTER
5236  024536  052705  110000                    BIS     #DVALID!PARER,R5  ;BUILD WHAT WAS EXPECTED
5237  024542  104006                            ERROR   6                 ;<ERROR- DID NOT GET CORRECT INFORMATION
5238  024544  126104  001442           11$:    CMPB    TRO(R1),R4        ;CHECK THE CHARACTER. IS IT CORRECT?
5239  024550  001407                            BEQ     12$               ;IF SO, GO SET UP NEXT CHARACTER
5240  024552  116105  001442                    MOVB    TRO(R1),R5        ;LOAD THE CHARACTER FOR ERROR REPORTING
5241  024556  042705  177400                    BIC     #^C<377>,R5       ;CLEAR SIGN EXTEND
5242  024562  042704  177400                    BIC     #^C<377>,R4       ;REMOVE THE JUNK FROM R4. THE ACTUAL CHARACTER
5243  024566  104005                            ERROR   5                 ;DATA ERROR
5244  024570  005261  001442           12$:    INC     TRO(R1)           ;SET UP THE NEXT CHARACTER
5245  024574  005203                            INC     R3                ;ADD TO THE TOTAL RECEIVED COUNT
5246  024576  005037  024632                    CLR     COUNT0            ;RESET COUNTERS TO NEXT
5247  024602  005037  024634                    CLR     COUNT1            ;RECIEVER INTERRUPT
5248  024606  032777  040000  155226           BIT     #TIE,@DZCSR       ;ARE TRANSMISSIONS DONE?
5249  024614  001005                            BNE     13$               ;IF NO, GO RECEIVE SOME MORE
5250  024616  020203                            CMP     R2,R3             ;ARE ALL CHARACTERS RECEIVED?
5251  024620  001003                            BNE     13$               ;IF NO, GO RECEIVE SOME MORE
5252  024622  042777  000100  155212           BIC     #RIE,@DZCSR       ;DISABLE RECEIVER INTERRUPTS
5253  024630  000002                    13$:    RTI                       ;GO BACK TO RECEIVER WAIT LOOP
5254  024632  000000                    COUNT0: 0
5255  024634  000000                    COUNT1: 0
5256
5257
5258                                            ;TRANSMITTER INTERRUPT SERVICE
5259                                            ;-----------------------------
5260
5261  024636  117701  155202           XMTSRV: MOVB    @DZCSR,R1         ;GET THE LINE NUMBER. IS THE TRANSMITTER
5262  024642  100411                            BMI     1$                ;REALLY READY? IF SO, GO LOAD THE CHARACTER
5263  024644  013700  001372                    MOV     SAVLIN,R0         ;ADJUST LOCATION SAVLIN
```

```
5264  024650  042701  177770            BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
5265  024654  010137  001372            MOV     R1,$SVLIN       ;FOR ERROR PRINTOUT
5266  024660  104003                    ERROR   3               ;*TRANSMITTER NOT READY- FALSE INTERRUPT
5267  024662  010037  001372            MOV     R0,SAVLIN       ;RESET SAVLIN TO PREVIOUS VALUE
5268  024666  042701  177770    1$.     BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
5269  024672  006301                    ASL     R1              ;MAKE SURE IT REFERENCES A WORD BOUNDARY
5270  024674  116177  001422  155164    MOVB    TDO(R1),@DZTDR  ;LOAD THE CURRENT CHARACTER FOR THIS LINE
5271  024702  005261  001422            INC     TDO(R1)         ;SET UP NEXT CHARACTER FOR THIS LINE
5272                                    ;*****************************************************************
5273                                                            ;(REV. F0: DELETED  INC
5274                                    ;*****************************************************************
5275  024706  ^2376.  001376  001422    CMP     XMTCNT,TDO(R1)  ;HAVE WE DONE ALL PATTERNS ON THIS LINE?
5276  024714  00.^:5                    BNE     4$              ;IF NOT, KEEP ON TRANSMITTING
5277  024716  012700  000001            MOV     #1,R0           ;SET UP A DESELECTION POINTER
5278  024722  006201                    ASR     R1              ;GET THE LINE NUMBER AGAIN
5279  024724  005301            2$:     DEC     R1              ;REDUCE THE COUNT. WAS THIS THE LINE?
5280  024726  100402                    BMI     3$              ;IF SO, GO DISABLE THE ENABLE BIT FOR IT
5281  024730  006300                    ASL     R0              ;MOVE THE POINTER TO THE NEXT LINE
5282  024732  000774                    BR      2$              ;GO CHECK THE NEXT LINE
5283  024734  140077  155116    3$:     BICB    R0,@DZTCR       ;DISABLE THE LINE POINTED TO BY R0
5284  024740  001003                    BNE     4$              ;IF MORE LINES ARE ACTIVE , GO CONTINUE TRANSMIT
5285  024742  042777  040000  155072    BIC     #TIE,@DZCSR     ;IF NOT, DISABLE TRANSMITTER INTERRUPTS
5286                                    ;*****************************************************************
5287  024750  005202            4$:     INC     R2              ;UP THE NUMBER OF TRANSMISSIONS         (REV. F0)
5288                                    ;*****************************************************************
5289  024752  000002                    RTI                     ;RETURN TO THE TIMING LOOP
5290
5291                                    ; RELATIVE TIME BUILDING ROUTINE
5292                                    ; -----------------------------
5293
5294  024754  012737  000004  001222  BUILD:  MOV     #4,$TMP2        ;ROTATE 4 BITS BACK INTO $TMP1
5295  024762  006037  001224            1$:  ROR     $TMP3           ;GET THE BITS FROM $TMP3, THE HIGH BYTE
5296  024766  006037  001220                 ROR     $TMP1           ;OF THE RELATIVE TIME COUNTER. PUT THEM BACK
5297  024772  005337  001222                 DEC     $TMP2           ;INTO $TMP1 USING THE CARRY BIT WITH
5298                                                                 ;ROTATE INSTRUCTIONS
5299  024776  001371                       BNE     1$              ;REDUCE COUNT. ALL BITS BACK? IF NOT, GET MORE
5300  025000  000207                       RTS     PC              ;RETURN TO CALLING TEST
5301
```

```
5302                                        ;RECEIVER SERVICE ROUTINE
5303
5304  025002  105777  155034        RXISR1: TSTB    @DZCSR          ;IS THE RECEIVER REALLY READY?
5305  025006  100401                        BMI     1$              ;IF SO, GO SERVICE IT
5306  025010  104004                        ERROR   4               ;<ERROR- RECEIVER DONE FLAG ISN'T SET
5307  025012  017704  155030        1$:     MOV     @DZRBUF,R4      ;SAVE THE RECEIVER INFORMATION
5308  025016  100401                        BMI     2$              ;IF IT WAS VALID, GO PROCESS IT
5309  025020  104023                        ERROR   23              ;ERROR- DATA VALID WASN'T SET
5310  025022  032704  070000        2$:     BIT     #OVRRUN!FRMERR!PARER,R4 ;ARE ANY ERROR FLAGS SET?
5311  025026  001403                        BEQ     3$              ;IF NOT, GO CONTINUE PROCESSING
5312  025030  013700  002046                MOV     DZRBUF,R0       ;SET UP FOR ERROR REPORTING
5313  025034  104002                        ERROR   2               ;ERROR- RECEIVER ERROR FLAG SET
5314  025036  010401        3$:     MOV     R4,R1           ;COPY THE RECEIVER INFORMATION
5315  025040  000301                        SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
5316  025042  042701  177770                BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
5317  025046  006301                        ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
5318  025050  120461  001442                CMPB    R4,TRO(R1)      ;IS THE CHARACTER WHAT IT SHOULD BE?
5319  025054  001413                        BEQ     4$              ;IF SO,GO CONTINUE PROCESSING
5320  025056  116105  001442                MOVB    TRO(R1),R5      ;GET WHAT WAS EXPECTED FOR ERROR REPORTING
5321  025062  042705  177400                BIC     #^C<377>,R5     ;ELIMINATE PROPAGATED SIGN
5322  025066  042704  177400                BIC     #^C<377>,R4     ;ISOLATE THE ACTUAL CHARACTER
5323  025072  010137  001372                MOV     R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER ERROR
5324  025076  006237  001372                ASR     SAVLIN          ;ALIGN IT CORRECTLY FOR REPORTING
5325  025102  104005                        ERROR   5               ;<DATA ERROR
5326  025104  005261  001442        4$:     INC     TRO(R1)         ;SET UP THE NEXT EXPECTED CHARACTER
5327  025110  005203                        INC     R3              ;INCREMENT THE COUNT OF RECEIVED CHARACTERS
5328  025112  032761  000020  001442        BIT     #20,TRO(R1)     ;HAVE ALL CHARACTERS BEEN RECEIVED?
5329  025120  001402                        BEQ     5$              ;IF NOT, GO RECEIVE SOME MORE
5330  025122  020203                        CMP     R2,R3           ;HAVE WE RECEIVED ALL CHARACTERS?
5331  025124  001401                        BEQ     6$              ;IF SO,GO DETERMINE THE TIMING
5332  025126  000002        5$:     RTI                             ;GO CONTINUE TIMING AND ALLOW INTERRUPTS
5333  025130  004737  024754        6$:     JSR     PC,BUILD        ;GET THE RELATIVE TIME (SIGNIFICANT BITS)
5334
5335  025134  013700  025214                MOV     OFFSET,R0       ;GET POINTER
5336  025140  013760  001220  002102        MOV     $TMP1,TMTBL(R0) ; SAVE THIS TEST'S TIME
5337  025146  005737  025214                TST     OFFSET          ;FIRST TEST?
5338  025152  001414                        BEQ     7$              ;IF NOT, GO CHECK THE TIME
5339  025154  005740                        TST     -(R0)           ;POINT TO THE PREVIOUS  TIME TAKEN
5340  025156  026037  002102  001220        CMP     TMTBL(R0),$TMP1 ;IS THIS TIME WHAT IT SHOULD BE?
5341  025164  101007                        BHI     7$              ;IF SO, GO TO THE NEXT TEST
5342  025166  016005  002102                MOV     TMTBL(R0),R5    ;PLACE WHAT WAS EXPECTED IN R5
5343  025172  010137  001372                MOV     R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER
5344  025176  006237  001372                ASR     SAVLIN          ;MAKE SURE IT'S THE LINE NUMBER
5345  025202  104021                        ERROR   21              ;TIMING ERROR
5346  025204  042777  000140  154630  7$:   BIC     #RIE!MSENAB,@DZCSR ;DISABLE THE DEVICE
5347  025212  000002                        RTI                     ;RETURN TO THE PROGRAM
5348  025214  000000        OFFSET: 0
```

```
                        ;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 STARTE


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

025216  012706  001120         XSTART: MOV     #STACK,SP       ;SET UP PROCESSOR STACK
025222  106427  000340                 MTPS    #PR7            ;LOCK OUT INTERRUPTS
025226  012737  025216  001126         MOV     #XSTART,#LPADR  ;SET UP IN CASE OF POWER FAIL
025234  005037  027412                 CLR     STFLG           ;CLEAR TEST START FLAG
025240  005037  001242                 CLR     #PASS           ;CLEAR PASS COUNT
025244  005037  001132                 CLR     #ERTTL          ;CLEAR ERROR COUNT
025250  105037  001123                 CLRB    #ERFLG          ;CLEAR ERROR FLAG
025254  005037  027416                 CLR     LAST            ;CLEAR LAST ERROR PC
```

```
5405  025260  032777  000001  153672  VEC1:   BIT     @SW00,@SWR      ;IF SW00=1, GET NEW VECTOR
5406  025266  001465                          BEQ     OTHER           ;AND CSR
5407  025270  012701  000300          VEC2:   MOV     #300,R1
5408  025274  012702  000302                  MOV     #302,R2
5409  025300  010221                  1$:     MOV     R2,(R1)+        ;RESTORE TRAPCATCHER
5410  025302  005022                          CLR     (R2)+           ;IN FLOATING VECTOR AREA
5411  025304  022122                          CMP     (R1)+,(R2)+     ;UPDATE THE POINTERS
5412  025306  020127  001000                  CMP     R1,#1000
5413  025312  001372                          BNE     1$
5414  025314  104403                          INSTR                   ;INPUT ADDRESS OF DEVICE VECTOR
5415  025316  027444                          MVECTOR                 ;MESSAGE "VECTOR ADDRESS-"
5416  025320  104405                          PARAM                   ;CONVERT STRING TO OCTAL
5417  025322  000300                          300                     ;LOW LIMIT
5418  025324  000770                          770                     ;HIGH  LIMIT
5419  025326  002072                          DZRIV                   ;LOCATIONS TO BE FILLED
5420  025330     003                  .BYTE   3                       ;LSB MASK
5421  025331     004                  .BYTE   4                       ;NUMBER OF LOCATIONS
5422  025332  104403                          INSTR                   ;INPUT ADDRESS OF DEVICE CSR
5423  025334  027466                          MREGAD                  ;MESSAGE "CONTROL REGISTER ADDRESS-"
5424  025336  104405                          PARAM                   ;CONVERT STRING TO OCTAL
5425  025340  160000                          160000                  ;LOW LIMIT
5426  025342  163700                          163700                  ;HIGH LIMIT
5427  025344  002042                          DZCSR                   ;LOCATIONS TO BE FILLED
5428  025346     007                  .BYTE   7                       ;LSB MASK
5429  025347     001                  .BYTE   1                       ;NUMBER OF LOCATIONS
5430  025350  013737  002042  002046          MOV     DZCSR,DZRBUF    ;BEGIN BUILDING DEVICE ADDRESSES
5431  025356  062737  000002  002046          ADD     #2,DZRBUF       ;FORM THE READ BUFFER ADDRESS
5432  025364  013737  002046  002052          MOV     DZRBUF,DZLPR    ;REMEMBER THAT THIS IS ALSO LINE PARAMETER REG.
5433  025372  013737  002046  002056          MOV     DZRBUF,DZTCR    ;BEGIN BUILDING TRANSMITTER CONTROL REGISTER
5434  025400  062737  000002  002056          ADD     #2,DZTCR        ;FORM THE TRANSMITTER CONTROL REGISTER POINTER
5435  025406  013737  002056  002060          MOV     DZTCR,MDZTCR
5436  025414  005237  002060                  INC     MDZTCR
5437  025420  013737  002056  002066          MOV     DZTCR,DZTDR     ;BEGIN FORMING TRANSMITTER DATA REGISTER
5438  025426  062737  000002  002066          ADD     #2,DZTDR        ;FORM THE TRANSMITTER DATA REGISTER
5439  025434  013737  002066  002062          MOV     DZTDR,DZMSR
5440  025442  032777  000002  153510  OTHER:  BIT     @SW01,@SWR      ;RESELECT OF TEST?
5441  025450  001427                          BEQ     XBEGIN          ;IF NOT, SKIP ASKING WHICH ONE
5442  025452  104403                          INSTR                   ;INPUT WHICH TEST YOU ARE RUNNING
5443  025454  027652                          MWHICH                  ;ECHO OR CABLE
5444  025456  104416                          PWHCH                   ;SET FLAG
5445  025460  027410                          WCHFLG                  ;THIS FLAG
5446  025462  104403            BAUD:         INSTR                   ;INPUT BAUD RATE
5447  025464  027574                          MSPEED                  ;MESSAGE "BAUD RATE-"
5448  025466  104415                          PARMD                   ;CONVERT DECIMAL STRING TO OCTAL
5449  025470  000062                          50.                     ;LOW LIMIT
5450  025472  022600                          9600.                   ;HIGH LIMIT
5451  025474  027426                          LINESP                  ;LOCATION TO BE FILLED
5452  025476     000                  .BYTE   0                       ;LSB MASK
5453  025477     001                  .BYTE   1                       ;NUMBER OF LOCATIONS
5454  025500  104413            LINEX:        DEVICE.CLR              ;CLEAR DEVICE
5455  025502  005037  027412                  CLR     STFLG           ;CLEAR PROGRAM START FLAG
5456  025506  104403                          INSTR                   ;INPUT LINE NUMBER
5457  025510  027564                          MLINE                   ;MESSAGE "LINE NUMBER-"
5458  025512  104405                          PARAM                   ;CONVERT  STRING TO OCTAL
5459  025514  000000                          0                       ;LOW LIMIT
5460  025516  000007                          7                       ;HIGH LIMIT
```

```
 5461  025520  001372                        SAVLIN                  ;LOCATION TO BE FILLED
 5462  025522     000              .BYTE      0                       ;LSB MASK
 5463  025523     001                         .BYTE   1               ;NUMBER OF LOCATIONS
 5464  025524  004537  027214                 JSR     R5,SET
 5465
 5466  025530  106427  000340      XBEGIN:    MTPS    #PR7            ;LOCK OUT INTERRUPTS
 5467  025534  012706  001120                 MOV     #STACK,SP       ;SET UP PROCESSOR STACK
 5468  025540  005037  027414                 CLR     LOCKUP          ;CLEAR TIMEOUT
 5469  025544  005737  027410                 TST     WCHFLG          ;ECHO OR CABLE TEST ?
 5470  025550  001413                         BEQ     2$              ;ECHO
 5471  025552  012737  026266  001126        MOV     #TEST2,@LPADR   ;CABLE TEST
 5472  025560  005737  027412                 TST     STFLG           ;ARE YOU LOOPING ?
 5473  025564  001017                         BNE     1$              ;YES
 5474  025566  005137  027412                 COM     STFLG           ;NO
 5475  025572  104402  027745                 TYPE    ,MCABLE         ;TYPE CABLE TEST
 5476  025576  000412                         BR      1$
 5477  025600  012737  025630  001126  2$:    MOV     #TEST1,@LPADR   ;SET UP ECHO TEST
 5478  025606  005737  027412                 TST     STFLG           ;ARE YOU LOOPING ?
 5479  025612  001004                         BNE     1$              ;YES
 5480  025614  005137  027412                 COM     STFLG           ;NO
 5481  025620  104402  027720                 TYPE    ,MTERM          ;TYPE ECHO TEST
 5482  025624  000177  153276      1$:        JMP     @@LPADR         ;START TESTING
 5483                                         ;THIS TEST WILL ACCEPT 1 CHARACTER AT A TIME
 5484                                         ;(IN INTERRUPT MODE) AND TRANSMIT THAT SAME CHARACTER,
 5485                                         ;ONE LINE AT A TIME, ANY LINE 0 THRU 7 (OCTAL)
 5486
 5487  025630  104413              TEST1:     DEVICE.CLR              ;CLEAR DZ11
 5488  025632  012737  000001  001122         MOV     #1,@TSTNM
 5489  025640  013777  027434  154210         MOV     NUMTCR,@DZTCR   ;SET TCR BIT
 5490  025646  013737  027432  001366         MOV     NUMLIN,PAR      ;SET PARAMETERS
 5491  025654  053737  027430  001366         BIS     SPEED,PAR       ;SET BAUD RATE
 5492  025662  013777  001366  154162         MOV     PAR,@DZLPR      ;LOAD PARAM.
 5493  025670  012777  000040  154144         MOV     #MSENAB,@DZCSR  ;SET SCANN ENABLE
 5494  025676  005004                         CLR     R4
 5495  025700  012705  027762                 MOV     #MQUICK,R5      ;SET MESSAGE BUFFER
 5496  025704  005777  154132          3$:    TST     @DZCSR          ;TRDY?
 5497  025710  100404                         BMI     2$              ;BR IF YES
 5498  025712  104414                         DELAY                   ;WAIT
 5499  025714  005304                         DEC     R4              ;
 5500  025716  001372                         BNE     3$              ;
 5501  025720  104003                         ERROR   3               ;NO TRDY SET! WHY?
 5502  025722  005004                  2$:    CLR     R4              ;RESET COUNTER TO 0
 5503  025724  112577  154136                 MOVB    (R5)+,@DZTDR    ;LOAD CHAR
 5504  025730  001365                         BNE     3$              ;
 5505  025732  004737  007652                 JSR     PC,SERV.G       ;<^G>?
 5506  025736  122777  000377  153214         CMPB    #377,@SWR       ;RE-DO QUICK BROWN?
 5507  025744  001731                         BEQ     TEST1           ;BR IF REPEAT PATTERN
 5508  025746  104413                         DEVICE.CLR
 5509  025750  106427  000340                 MTPS    #PR7            ;LOCK OUT INTERRUPTS
 5510  025754  012737  026724  001360         MOV     #XEOP,NEXT
 5511  025762  104413                         DEVICE.CLR
 5512  025764  013737  027432  001366         MOV     NUMLIN,PAR      ;SELECT LINE # & SET INTERRUPT ENABLE
 5513  025772  053737  027430  001366         BIS     SPEED,PAR       ;SET LINE SPEED AND
 5514                                         ;CHARACTER LENGTH (TRANS. & REC.)
 5515  026000  052737  010000  001366         BIS     #RCVON,PAR      ;MAKE SURE RECEIVER IS TURNED ON
 5516  026006  013777  001366  154036         MOV     PAR,@DZLPR      ;LOAD THE LINE PARAMETER REGISTER
```

```
 5517  026014  012777  026070  154050          MOV     #INTSVC,@DZRIV  ;SET UP INTERRUPT SERVICE
 5518  026022  013777  027436  154044          MOV     PRIO,@DZRIS     ;AND LEVEL
 5519  026030  106437  030310                  MTPS    #LESS1          ;ALLOW INTERRUPTS
 5520  026034  012777  000140  154000          MOV     #RIE!MSENAB,@DZCSR ;SET RECEIVER INTERRUPT ENABLE
 5521  026042  104402  027612                  TYPE    ,MCHAR          ;TYPE "ANY CHARACTER"
 5522  026046  105777  153112          1$:     TSTB    @#TKS           ;IF SOMBODY HITS A KEY- GET NEW LINE #
 5523  026052  100375                          BPL     1$              ;LOOP HERE
 5524  026054  005777  153106                  TST     @#TKB           ;CLEAR CHAR
 5525  026060  004737  007652                  JSR     PC,SERV.G       ;MAKE SURE IT WASN'T <^G>
 5526  026064  000137  025500                  JMP     LINEX           ;
 5527
 5528
 5529                                          ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
 5530  026070  105777  153746          INTSVC: TSTB    @DZCSR          ;TEST REC. FLAG
 5531  026074  100401                          BMI     .+4
 5532  026076  104004                          ERROR   4               ;ERROR - INTERRUPT NOT CAUSED BY FLAG
 5533  026100  017737  153742  027440          MOV     @DZRBUF,RECDAT
 5534  026106  100401                          BMI     .+4
 5535  026110  104023                          ERROR   23              ;NON- VALID CHARACTER
 5536  026112  032737  020000  027440          BIT     #BIT13,RECDAT   ;CHECK FOR FRAMING ERROR
 5537  026120  001401                          BEQ     .+4             ;BR IF NO ERROR
 5538  026122  104025                          ERROR   25              ;EITHER SOMBODY HIT THE
 5539                                          ;"BREAK KEY" OR YOU HAVE AN ERROR!
 5540  026124  113737  027440  027442          MOVB    RECDAT,TBUF     ;MOVE CHARACTER TO OUTPUT AREA
 5541  026132  113737  027440  011272          MOVB    RECDAT,INBUF    ;MOVE CHARACTER TO CHECK FOR ^C
 5542  026140  042737  177600  011272          BIC     #^C<177>,INBUF  ;STRIP JUNK PLUS PARITY
 5543  026146  042737  174377  027440          BIC     #174377,RECDAT  ;SAVE ONLY LINE  NUMBER
 5544  026154  000337  027440                  SWAB    RECDAT
 5545  026160  023737  001372  027440          CMP     SAVLIN,RECDAT   ;DOES THE LINE # COMPARE?
 5546  026166  001401                          BEQ     .+4
 5547  026170  104015                          ERROR   15              ;#WRONG LINE NUMBER
 5548  026172  012777  000040  153642          MOV     #MSENAB,@DZCSR  ;START THE TRANSMITTERS SCANNER
 5549  026200  123727  011272  000003          CMPB    INBUF,#3                ;IS IT A ^C ?
 5550  026206  001004                          BNE     1$              ;NO
 5551  026210  104413                          DEVICE.CLR
 5552  026212  012716  026724                  MOV     #XEOP,(SP)      ;CRUNCH STACK
 5553  026216  000002                          RTI
 5554  026220  005003                  1$:     CLR     R3              ;INITIALIZE DELAY
 5555  026222  013777  027434  153626          MOV     NUMTCR,@DZTCR   ;ENABLE THE LINE
 5556  026230  005777  153606          10$:    TST     @DZCSR          ;TRANSMITTER READY?
 5557  026234  100403                          BMI     2$              ;IF YES BRANCH
 5558  026236  005203                          INC     R3              ;INCREMENT DELAY
 5559  026240  001373                          BNE     10$             ;DELAY DONE?
 5560  026242  104003                          ERROR   3               ;TRANSMIT READY NOT SET!
 5561  026244  113777  027442  153614  2$:     MOVB    TBUF,@DZTDR     ;TRANSMIT THE CHARACTER
 5562  026252  012777  000140  153562          MOV     #RIE!MSENAB,@DZCSR      ;RESTART THE RECEIVER
 5563  026260  005077  153572                  CLR     @DZTCR          ;CLEAR TCR BIT
 5564  026264  000002                          RTI
 5565
 5566
 5567                                  ;THIS TEST TRANSMITS A BINARY COUNT PATTERN
 5568                                  ;VIA INTERRUPT MODE TO THE RECEIVER
 5569                                  ;...THE LINE UNDER TEST MUST BE TERMINATED WITH THE TEST CONNECTOR
 5570  026266  106427  000340          TEST2:  MTPS    #PR7            ;DISABLE INTERRUPTS
 5571  026272  012737  000002  001122          MOV     #2,@#TSTNM
 5572  026300  012737  026724  001360          MOV     #XEOP,NEXT
```

```
                                        DEVICE.CLR              ;
                                        ;*TEST TO VERIFY THAT SETTING DTR FOR A GIVEN LINE
                                        ;*WILL BRING UP "CO" AND "RING" FOR THE SAME LINE
                                        ;*THE DIST PNL MUST HAVE JUMPER FROM DTR TO RQST TO SEND
                                        ;*IN ORDER FOR THIS TEST TO WORK!
026310  012737  026316  001362         MOV     #1$,LOCK        ;LOOP
026316  113777  027434  153534  1$:    MOVB    NUMTCR,@#DZTCR  ;SET DTR
026324  005005                         CLR     R5              ;
026326  153705  027434                 BISB    NUMTCR,R5       ;BUILD EXPECTED
026332  000305                         SWAB    R5              ;PUT IN HIGH BYTE
026334  153705  027434                 BISB    NUMTCR,R5       ;
026340  104414                         DELAY                   ;WAIT FOR CABLE DELAY
026342  017704  153514                 MOV     @#DZMSR,R4      ;READY MODEM BITS
026346  020504                         CMP     R5,R4           ;ARE THEY OK?
026350  001401                         BEQ     2$              ;BR IF YES
026352  104022                         ERROR   22              ;IS THE TEST CONNECTOR ON?
                                                                ;HAS RIGHT LINE BEEN SELECTED?
                                                                ;IF SO- YOU HAVE A PROBLEM!
                                                                ;MODEM BITS NOT RIGHT
026354  104401                 2$:     SCOP1                   ;LOOP
026356  104413                 3$:     DEVICE.CLR              ;INIT DZ11
026360  013737  027430  001366         MOV     SPEED,PAR       ;SET LINE SPEED
026366  053737  027432  001366         BIS     NUMLIN,PAR      ;SELECT LINE # & REC.  INTERRUPT ENABLE
026374  052737  010000  001366         BIS     #RCVON,PAR      ;ENABLE THE RECEIVER FOR THIS LINE
026402  052777  040140  153432         BIS     #TIE!RIE!MSENAB,@#DZCSR ;SET TRANSMITTER INTERRUPT ENABLE
026410  012777  026524  153454         MOV     #INTREC,@#DZRIV ;SET UP INTR SERVICE
026416  013777  027436  153450         MOV     PRIO,@#DZRIS    ;SET UP LEVEL
026424  012777  026704  153444         MOV     #INTRAN,@#DZTIV ;SET UP INTR SERVICE
026432  013777  027436  153440         MOV     PRIO,@#DZTIS    ;SET UP LEVEL
026440  005001                         CLR     R1              ;RX DATA POINTER- SET TO 0
026442  005002                         CLR     R2              ;TX DATA POINTER- SET TO 0
026444  013777  001366  153400         MOV     PAR,@#DZLPR     ;SET THE PARAMETERS AND TURN ON RECEIVER
026452  106437  030310                 MTPS    #PLESS1         ;ALLOW INTERRUPTS
026456  013777  027434  153372         MOV     NUMTCR,@#DZTCR  ;SET UP TCR BIT
                                        ;YOU RETURN HERE AFTER EVERY RECEIVER INTERRUPT
026464  105777  152474         SPIN:   TSTB    @#TKS           ;IF SOMEBODY HITS A KEY- GET A NEW LINE #
026470  100006                         BPL     1$              ;BR IF NO KEY HIT
026472  005777  152470                 TST     @#TKB           ;CLEAR CHAR
026476  004737  007652                 JSR     PC,SERV.G       ;MAKE SURE IT WASN'T <^G>
026502  000137  025500                 JMP     LINEX           ;SW02=1
026506  005237  027414         1$:     INC     LOCKUP          ;INC TIMEOUT FLAG
026512  001364                         BNE     SPIN            ;IF NOT 0  RETURN SPINNING
026514  104011                         ERROR   11              ;*RECEIVER FAILED TO INTERRUPT  CHECK CABLE/TERMINATOR
026516  104413                 QUITS:  DEVICE.CLR
026520  000137  026724                 JMP     XEOP            ;CALL FOR END OF PASS
026524  005037  027414         INTREC: CLR     LOCKUP          ;CLEAR TIMEOUT FLAG
026530  105777  153306                 TSTB    @#DZCSR         ;TEST REC DONE
026534  100401                         BMI     .+4             ;YES
026536  104004                         ERROR   4               ;*FALSE INTERRUPT
026540  017737  153302  027440         MOV     @#DZRBUF,RECDAT ;SAVE WORD
026546  100401                         BMI     .+4
026550  104023                         ERROR   23              ;*NON VALID CHARACTER
026552  032737  040000  027440         BIT     #BIT14,RECDAT   ;DATA OVERRUN ?
026560  001401                         BEQ     .+4             ;NO
026562  104024                         ERROR   24              ;*YES
```

```
5629  026564  032737  020000  027440          BIT     @BIT13,RECDAT   ;FRAMING ERROR ?
5630  026572  001401                          BEQ     .+4             ;NO
5631  026574  104025                          ERROR   25              ;*YES
5632  026576  032737  010000  027440          BIT     @BIT12,RECDAT   ;PARITY ERROR ?
5633  026604  001401                          BEQ     .+4             ;NO
5634  026606  104026                          ERROR   26              ;*YES
5635  026610  110105                          MOVB    R1,R5           ;SET EXPECTED
5636  026612  042705  177400                  BIC     @#C<377>,R5     ;CLEAR HIGH BYTE
5637  026616  113704  027440                  MOVB    RECDAT,R4       ;GET FOUND
5638  026622  042704  177400                  BIC     @#C<377>,R4     ;CLEAR HIGH BYTE
5639  026626  020504                          CMP     R5,R4   ;OK?
5640  026630  001401                          BEQ     .+4
5641  026632  104005                          ERROR   5               ;DATA ERROR
5642  026634  042737  174377  027440          BIC     @174377,RECDAT  ;SAVE ONLY LINE NUMBER
5643  026642  000337  027440                  SWAB    RECDAT
5644  026646  023737  001372  027440          CMP     SAVLIN,RECDAT   ;DOES THE LINE # COMPARE ?
5645  026654  001401                          BEQ     .+4             ;YES
5646  026656  104015                          ERROR   15              ;*WRONG LINE #
5647  026660  120127  000377                  CMPB    R1,@377         ;LAST CHARACTER ?
5648  026664  001003                          BNE     1$              ;NO
5649  026666  012716  026516                  MOV     @QUITS,(SP)     ;CRUNCH STACK
5650  026672  000403                          BR      2$
5651  026674  105201                  1$:     INCB    R1              ;UPDATE EXPECTED DATA
5652  026676  012716  026464                  MOV     @SPIN,(SP)      ;CRUNCH STACK
5653  026702  000002                  2$:     RTI
5654
5655  026704  005777  153132          INTRAN: TST     @DZCSR  ;TEST TRANSMIT FLAG
5656  026710  100401                          BMI     .+4
5657  026712  104003                          ERROR   3               ;*FALSE INTERRUPT
5658  026714  110277  153146                  MOVB    R2,@DZTDR       ;TRANSMIT A CHARACTER
5659  026720  105202                          INCB    R2              ;UPDATE TX DATA
5660  026722  000002                          RTI     ;RETURN
```

```
5661
5662                                        ;END OF PASS
5663                                        ;RESTART TEST
5664
5665  026724  104402              XEOP:   TYPE                    ;TYPE NAME OF TEST
5666  026726  027522                      MPASS
5667  026730  005037  027416              CLR     LAST            ;CLEAR LAST ERROR PC
5668  026734  105037  001123              CLRB    $ERFLG          ;CLEAR ERROR FLAG
5669  026740  000137  025530      RSTRT:  JMP     XBEGIN
5670
5671                                        ;CONVERT DECIMAL ASCII STRING TO OCTAL
5672  026744  011605              .PARMD: MOV     (SP),R5
5673  026746  012537  027130              MOV     (R5)+,6$
5674  026752  012537  027132              MOV     (R5)+,7$
5675  026756  012537  027134              MOV     (P5)+,8$
5676  026762  112537  027136              MOVB    (R5)+,9$
5677  026766  112537  027137              MOVB    (R5)+,10$
5678  026772  010516                      MOV     R5,(SP)
5679  026774  005005              2$:     CLR     P5
5680  026776  012704  011272              MOV     #INBUF,R4
5681  027002  122714  000015              CMPB    #15,(R4)
5682  027006  001424                      BEQ     3$
5683  027010  121427  000060      1$:     CMPB    (R4),#'0
5684  027014  002421                      BLT     3$
5685  027016  121427  000071              CMPB    (R4),#'9
5686  027022  003016                      BGT     3$
5687  027024  142714  000060              BICB    #'0,(R4)
5688  027030  005002                      CLR     R2
5689  027032  152402                      BISB    (R4)+,R2
5690  027034  060205                      ADD     R2,R5
5691  027036  122714  000015              CMPB    #15,(R4)
5692  027042  001410                      BEQ     4$
5693  027044  006305                      ASL     R5      ;X2
5694  027046  010502                      MOV     R5,R2   ;SAVE X2
5695  027050  006305                      ASL     R5      ;X4
5696  027052  006305                      ASL     R5      ;X8
5697  027054  060205                      ADD     R2,R5   ;TIMES 10
5698  027056  000754                      BR      1$
5699  027060  104404              3$:     INSTER
5700  027062  000744                      BR      2$
5701
5702                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
5703
5704  027064  020537  027132      4$:     CMP     R5,7$
5705  027070  101373                      BHI     3$
5706  027072  020537  027130              CMP     R5,6$
5707  027076  103770                      BLO     3$
5708  027100  133705  027136              BITB    9$,R5
5709  027104  001365                      BNE     3$
5710
5711                                        ;STORE NUMBER AT SPECIFIED ADDRESS
5712
5713  027106  013704  027134              MOV     8$,R4
5714  027112  010524              5$:     MOV     R5,(R4)+
5715  027114  062705  000002              ADD     #2,R5
5716  027120  105337  027137              DECB    10$
```

```
5717  027124  001372                          BNE     5$
5718  027126  000002                          RTI
5719  027130  000000                  6$:     0
5720  027132  000000                  7$:     0
5721  027134  000000                  8$:     0
5722  027136     000                  9$:     .BYTE 0
5723  027137     000                  10$:    .BYTE 0
5724
5725
5726                                          ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
5727                                          ;BUFFER TO THE CHARACTERS "E" AND "C".
5728                                          ;IF THE CHARACTER IS "E" CLEAR THE FLAG
5729                                          ;IF THE CHARACTER IS "C" SET THE FLAG
5730
5731  027140  017605  000000          .PAWCH:MOV      @(SP),R5
5732  027144  142737  000040  011272          BICB    #40,INBUF       ;SET FOR LOWER CASE INPUT
5733  027152  122737  000105  011272          CMPB    #'E,INBUF       ;IS IT "E" ?
5734  027160  001002                          BNE     1$
5735  027162  105015                          CLRB    (R5)            ;000
5736  027164  000406                          BR      2$
5737  027166  122737  000103  011272  1$:     CMPB    #'C,INBUF       ;IS IT "C" ?
5738  027174  001005                          BNE     3$
5739  027176  112715  177777                  MOVB    #-1,(R5)        ;3177
5740  027202  062716  000002          2$:     ADD     #2,(SP)
5741  027206  000002                          RTI
5742  027210  104404                  3$:     INSTER                  ;RETRY
5743  027212  000752                          BR      .PAWCH
5744
5745
5746
5747                                          ;THIS ROUTINE CONVERTS LINE SPEED (LINESP) AND
5748                                          ;LINE NUMBER (SAVLIN) FOR DZLPR, DZTCR AND DZCSR
5749                                          ;REGISTER USAGE.
5750
5751  027214  013737  001372  027432  SET:    MOV     SAVLIN,NUMLIN   ;SAVE SAVLIN
5752  027222  013700  001372          XTCR0:  MOV     SAVLIN,R0       ;COPY THE LINE NUMBER FOR LOOP CONTROL
5753  027226  005037  027434                  CLR     NUMTCR          ;SET A DEFAULT OF LINE 0 OR NO LINES
5754  027232  012702  000001                  MOV     #1,R2           ;SET A BIT POINTER TO THE FIRST LINE
5755  027236  005300                  XTCR1:  DEC     R0              ;REDUCE THE INDICATOR.IS IT MINUS YET?
5756  027240  100402                          BMI     SET1            ;IF SO, R2 POINTS TO THE RIGHT LINE
5757  027242  006302                          ASL     R2              ;IF NOT, MOVE THE POINTER TO THE NEXT LINE
5758  027244  000774                          BR      XTCR1           ;GO SEE IF THIS LINE IS THE ONE
5759  027246  012701  027310          SET1:   MOV     #TABLE2,R1
5760  027252  010237  027434                  MOV     R2,NUMTCR       ;COPY THE CORRECT BIT POINTER
5761  027256  022137  027426          1$:     CMP     (R1)+,LINESP
5762  027262  001407                          BEQ     2$
5763  027264  005721                          TST     (R1)+           ;IS IT THE END OF TABLE?
5764  027266  001373                          BNE     1$              ;NO
5765  027270  104402  027536                  TYPE    ,MINVAL         ;INVALID BAUD RATE,BEGIN AGAIN
5766  027274  012705  025462                  MOV     #BAUD,R5        ;JUMP TO BAUD THRU R5
5767  027300  000402                          BR      3$
5768  027302  011137  027430          2$:     MOV     (R1),SPEED      ;SET UP BAUD RATE
5769  027306  000205                  3$:     RTS     R5
5770
5771
5772
```

```
5773                                        ;THE FOLLOWING IS A TABLE OF LEGAL BAUD RATES (8 BITS/CHAR)
5774  027310  000062            TABLE2: .WORD   50.             ;50 BAUD
5775  027312  010070                    .WORD   10070           ;
5776  027314  000113                    .WORD   75.             ;75 BAUD
5777  027316  010470                    .WORD   10470           ;
5778  027320  000156                    .WORD   110.            ;110 BAUD
5779  027322  011070                    .WORD   11070           ;TWO STOP BITS
5780  027324  000207                    .WORD   135.            ;134.5 BAUD
5781  027326  011470                    .WORD   11470           ;TWO STOP BITS
5782  027330  000226                    .WORD   150.            ;150 BAUD
5783  027332  012070                    .WORD   12070           ;TWO STOP BITS
5784  027334  000454                    .WORD   300.            ;300 BAUD
5785  027336  012430                    .WORD   12430           ;ONE STOP BIT
5786  027340  001130                    .WORD   600.            ;600 BAUD
5787  027342  013030                    .WORD   13030           ;ONE STOP BIT
5788  027344  002260                    .WORD   1200.           ;1200 BAUD
5789  027346  013430                    .WORD   13430           ;ONE STOP BIT
5790  027350  003410                    .WORD   1800.           ;1800 BAUD
5791  027352  014030                    .WORD   14030           ;ONE STOP BIT
5792  027354  003720                    .WORD   2000.           ;2000 BAUD
5793  027356  014430                    .WORD   14430           ;ONE STOP BIT
5794  027360  004540                    .WORD   2400.           ;2400 BAUD
5795  027362  015030                    .WORD   15030           ;ONE STOP BIT
5796  027364  007020                    .WORD   3600.           ;3600 BAUD
5797  027366  015430                    .WORD   15430           ;ONE STOP BIT
5798  027370  011300                    .WORD   4800.           ;4800 BAUD
5799  027372  016030                    .WORD   16030           ;ONE STOP BIT
5800  027374  016040                    .WORD   7200.           ;7200 BAUD
5801  027376  016430                    .WORD   16430           ;ONE STOP BIT
5802  027400  022600                    .WORD   9600.           ;9600 BAUD
5803  027402  017070                    .WORD   17070           ;
5804  027404  177777  000000            .WORD   -1,0            ;TABLE TERMINATOR
5805
5806
5807  027410  000000            WCHFLG:0                        ;ECHO OR CABLE FLAG
5808  027412  000000            STFLG: 0                        ;PROGRAM START FLAG
5809  027414  000000            LOCKUP: 0                       ;TIMEOUT FLAG
5810  027416  000000            LAST:   0                       ;LAST ERROR PC
5811  027420  000000            TDATA:  0
5812  027422  000000            RDATA:  0
5813  027424  000000            BYTCNT: 0
5814  027426  000156            LINESP: 110.                    ;DEFAULT BAUD RATE
5815  027430  006307            SPEED:  6307                    ;DEFAULT 110 BAUD, 8 BITS/CHAR,
5816                                                            ;FDX, 2 STOP BITS
5817  027432  000100            NUMLIN: 100                     ;DEFAULT VALUE, REC. INTERRUPT ENABLED
5818
5819  027434  000001            NUMTCR: 1                       ;DEFAULT VALUE,TCR BIT 0
5820  027436  000240            PRIO:   240                     ;DEFAULT DEVICE PRIORITY 5
5821  027440  000000            RECDAT: 0
5822  027442  000000            TBUF:   0
5823  027444  053200  041505  047524  MVECTO: .ASCIZ  <200>/VECTOR ADDRESS- /
      027466  041600  047117  051124  MREGAD: .ASCIZ  <200>/CONTROL REGISTER ADDRESS- /
      027522  050200  051501  020123  MPASS:  .ASCIZ  <200>/PASS DONE./
      027536  044600  053116  046101  MINVAL: .ASCIZ  <200>/INVALID BAUD RATE - /
      027564  046200  047111  035105  MLINE:  .ASCIZ  <200>/LINE: /
      027574  041200  052501  020104  MSPEED: .ASCIZ  <200>/BAUD RATE - /
```

```
        027612  052200  050131  020105  MCHAR:  .ASCIZ  <200>/TYPE A CHAR. ON DZ11 TERMINAL /
        027652  053600  044510  044103  MWHICH: .ASCIZ  <200>/WHICH TEST ? ECHO OR CABLE (E OR C) /
        027720  052200  051105  044515  MTERM:  .ASCIZ  <200>/TERMINAL ECHO TEST /
        027745     200  040503  046102  MCABLE: .ASCIZ  <200>/CABLE TEST /
        027762  006777  177777  177412  MQUICK: .ASCII  <377><15><377><377><12><377><377>
        027771     124  042510  050440          .ASCII  /THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789/
        030066  006777  177777  177412          .ASCII  <377><15><377><377><12><377><377><377><0>
                030100                  .EVEN
                                        ;::*****************************************************************
  5824                                          ;UTILITIES
  5825                                  ;::*****************************************************************
  5826
  5827                                  ;THIS UTILITY CALCULATES PRIORITY LEVEL,SETS UP CSR'S,SETS UP VECTORS.
  5828  030100  006337  030306          DZLEV:  ASL     DZPRT           ;BUILD PRIORITY IN THIS LOCATION
  5829  030104  006337  030306                  ASL     DZPRT           ;USING ARITHMETIC SHIFTS, ROTATE
  5830  030110  006337  030306                  ASL     DZPRT           ;       THE PRIORITY LEVEL PAST
  5831  030114  006337  030306                  ASL     DZPRT           ;       THE BIT POSITIONS CORRE-
  5832  030120  006337  030306                  ASL     DZPRT           ;       SPONDING TO THE CONDITION CODES
  5833  030124  013737  030306  030310          MOV     DZPRT,LESS1     ;MOVE THIS TO LESS1
  5834  030132  162737  000001  030310          SUB     #1,LESS1        ;CREATE THE NEXT LOWEST PRIORITY
  5835  030140  042737  000037  030310          BIC     #37,LESS1       ;INSURE THAT THE  TNZVC BITS ARE CLEAR
  5836  030146  013700  002072                  MOV     DZRIV,R0        ;PLACE THE BASE VECTOR ADDRESS IN R0
  5837  030152  062700  000002                  ADD     #2,R0           ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
  5838  030156  010037  002074                  MOV     R0,DZRIS        ;STORE IT HERE
  5839  030162  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
  5840  030166  010037  002076                  MOV     R0,DZTIV        ;STORE IT HERE
  5841  030172  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
  5842  030176  010037  002100                  MOV     R0,DZTIS        ;STORE IT HERE
  5843
  5844                                  ;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN DZ11. $BASE IS THE BASE ADDRESS
  5845                                  ;OF THE DEVICE
  5846  030202  013700  001310                  MOV     $BASE,R0        ;COPY THE ADDRESS BEING LOADED
  5847  030206  010037  002042                  MOV     R0,DZCSR        ;XXX0
  5848  030212  005200                          INC     R0
  5849  030214  010037  002044                  MOV     R0,MDZCSR       ;XXX1
  5850  030220  005200                          INC     R0
  5851  030222  010037  002046                  MOV     R0,DZRBUF       ;XXX2
  5852  030226  010037  002052                  MOV     R0,DZLPR        ;XXX2
  5853  030232  005200                          INC     R0
  5854  030234  010037  002050                  MOV     R0,MDZRBUF      ;XXX3
  5855  030240  010037  002054                  MOV     R0,MDZLPR       ;XXX3
  5856  030244  005200                          INC     R0
  5857  030246  010037  002056                  MOV     R0,DZTCR        ;XXX4
  5858  030252  005200                          INC     R0
  5859  030254  010037  002060                  MOV     R0,MDZTCR       ;XXX5
  5860  030260  005200                          INC     R0
  5861  030262  010037  002062                  MOV     R0,DZMSR        ;XXX6
  5862  030266  010037  002066                  MOV     R0,DZTDR        ;XXX6
  5863  030272  005200                          INC     R0
  5864  030274  010037  002064                  MOV     R0,MDZMSR       ;XXX7
  5865  030300  010037  002070                  MOV     R0,MDZTDR       ;XXX7
  5866  030304  000207                          RTS     PC
  5867  030306  000240          DZPRT:  PR5
  5868  030310  000200          LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS
  5869
```

```
5870                                            ;ERROR ERROR TABLE
5871  030312  000000                .ERRTAB:      0       ;ERROR 0
5872  030314  000000                              0
5873  030316  000000                              0
5874
5875  030320  030540                              EM1     ;ERROR
5876  030322  032045                              DM1
5877  030324  032276                              DT1
5878
5879  030326  030613                              EM2     ;ERROR 2
5880  030330  032070                              DM2
5881  030332  032310                              DT2
5882
5883  030334  030641                              EM3     ;ERROR 3
5884  030336  032123                              DM3
5885  030340  032326                              DT3
5886
5887  030342  030700                              EM4     ;ERROR 4
5888  030344  032123                              DM3
5889  030346  032326                              DT3
5890
5891  030350  030727                              EM5     ;ERROR 5
5892  030352  032135                              DM4
5893  030354  032334                              DT4
5894
5895  030356  030756                              EM6     ;ERROR 6
5896  030360  032135                              DM4
5897  030362  032334                              DT4
5898
5899  030364  031014                              EM7     ;ERROR 7
5900  030366  032123                              DM3
5901  030370  032326                              DT3
5902
5903  030372  031055                              EM8     ;ERROR 10
5904  030374  032123                              DM3
5905  030376  032326                              DT3
5906
5907  030400  031117                              EM9     ;ERROR 11
5908  030402  032123                              DM3
5909  030404  032326                              DT3
5910
5911  030406  031155                              EM10    ;ERROR 12
5912  030410  032123                              DM3
5913  030412  032326                              DT3
5914
5915  030414  031214                              EM13    ;ERROR 13
5916  030416  032123                              DM3
5917  030420  032326                              DT3
5918
5919  030422  031245                              EM14    ;ERROR 14
5920  030424  032123                              DM3
5921  030426  032326                              DT3
5922
5923  030430  031277                              EM15    ;ERROR 15
5924  030432  000000                              0
5925  030434  000000                              0
```

```
5926
5927  030436  031341                        EM16
5928  030440  032123                        DH3
5929  030442  032326                        DT3
5930
5931  030444  031412                        EM17    ;ERROR 17
5932  030446  032123                        DH3
5933  030450  032326                        DT3
5934
5935  030452  031450                        EM20
5936  030454  032123                        DH3
5937  030456  032326                        DT3
5938
5939  030460  031511                        EM21    ;ERROR 21
5940  030462  032164                        DH5
5941  030464  032352                        DT5
5942
5943  030466  031541                        EM22    ;ERROR 22
5944  030470  032135                        DH4
5945  030472  032334                        DT4
5946
5947  030474  031603                        EM23    ;ERROR 23
5948  030476  032123                        DH3
5949  030500  032326                        DT3
5950
5951  030502  031633                        EM24
5952  030504  032123                        DH3
5953  030506  032326                        DT3
5954
5955  030510  031661                        EM25
5956  030512  032123                        DH3
5957  030514  032326                        DT3
5958
5959  030516  031711                        EM26
5960  030520  032123                        DH3
5961  030522  032326                        DT3
5962
5963  030524  031740                        EM27
5964  030526  032123                        DH3
5965  030530  032326                        DT3
5966
5967  030532  032010                        EM30
5968  030534  032243                        DH6
5969  030536  032374                        DT6
5970
```

```
5971                                        ;ERROR MESSAGES
5972  030540  047200  020117  046123  EM1:    .ASCIZ  <200>/NO SLAVE SYNC RESPONSE FROM DZ11 REGISTER/
      030613     200  042522  044507  EM2:    .ASCIZ  <200>/REGISTER R/W FAILURE?
      030641     200  051124  047101  EM3:    .ASCIZ  <200>/TRANSMIT READY (TRDY) NOT SET/
      030700  051200  041505  044505  EM4:    .ASCIZ  <200>/RECEIVER DONE NOT SET/
      030727     200  040504  040524  EM5:    .ASCIZ  <200>/DATA COMPARISON ERROR/
      030756  042200  030532  020061  EM6:    .ASCIZ  <200>/DZ11 =RECEIVER BUFFER= ERROR/
      031014  052200  040522  051516  EM7:    .ASCIZ  <200>/TRANSMITTER FAILED TO INTERRUPT/
      031055     200  047125  054105  EM8:    .ASCIZ  <200>/UNEXPECTED TRANSMITTER INTERRUPT/
      031117     200  042522  042503  EM9:    .ASCIZ  <200>/RECEIVER FAILED TO INTERRUPT/
      031155     200  047125  054105  EM10:   .ASCIZ  <200>/UNEXPECTED RECEIVER INTERRUPT/
      031214  051600  046111  020117  EM13:   .ASCIZ  <200>/SILO ALARM SET TOO SOON/
      031245     200  044523  047514  EM14:   .ASCIZ  <200>/SILO ALARM FAILED TO SET/
      03127?     200  041501  044524  EM15:   .ASCIZ  <200>/ACTION DETECTED ON INVALID LINE./
      031341     200  042522  042101  EM16:   .ASCIZ  <200>/READING DZRBUF DID NOT CLEAR SILO ALARM/
      031412  042200  052101  020101  EM17:   .ASCIZ  <200>/DATA VALID SHOULD NOT BE SET/
      031450  051200  041505  044505  EM20:   .ASCIZ  <200>/RECEIVER DONE SHOULD NOT BE SET/
      031511     200  042522  040514  EM21:   .ASCIZ  <200>/RELATIVE TIMING ERROR./
      031541     200  047515  042504  EM22:   .ASCIZ  <200>/MODEM SIGNAL ERROR ON CABLE TEST/
      031603     200  040504  040524  EM23:   .ASCIZ  <200>/DATA VALID IS NOT SET!/
      031633     200  040504  040524  EM24:   .ASCIZ  <200>/DATA OVERRUN IS SET!/
      031661     200  051106  046501  EM25:   .ASCIZ  <200>/FRAMING ERROR OCCURRED/
      031711     200  040520  044522  EM26:   .ASCIZ  <200>/PARITY ERROR OCCURRED/
      031740  043200  046125  020114  EM27:   .ASCIZ  <200>/FULL BINARY COUNT PATTERN NOT RECEIVED/
      032010  041200  052501  020104  EM30:   .ASCIZ  <200>/BAUD RATE TIMING TEST ERROR/
      032045     200  051124  050101  DM1:    .ASCIZ  <200>/TRAP PC  DZ11 REG/
      032070  042600  050130  041505  DM2:    .ASCIZ  <200>/EXPECTED  FOUND  REGISTER/
      032123     200  044514  042516  DM3:    .ASCIZ  <200>/LINE NO./
      032135     200  054105  042520  DM4:    .ASCIZ  <200>/EXPECTED  FOUND  LINE/
      032164  052200  020130  044514  DM5:    .ASCIZ  <200>/TX LINE PREVIOUS TIME  ACTUAL TIME  PARAMETER/
      032243     200  044510  044107  DM6:    .ASCIZ  <200>/HIGH   LOW    COUNT  LINE/
                                      .EVEN
                                              ;DATA TABLES FOR ERROR MESSAGES
      032276  000002                  DT1:    2
      032300     006      003                 .BYTE   6,3
      032302  001204                          $REG1
      032304     006      001                 .BYTE   6,1
      032306  001202                          $REG0

      032310  000003                  DT2:    3
      032312     006      004                 .BYTE   6,4
      032314  001214                          $REG5
      032316     006      001                 .BYTE   6,1
      032320  001212                          $REG4
      032322     006      001                 .BYTE   6,1
      032324  001202                          $REG0

      032326  000001                  DT3:    1
      032330     003      001                 .BYTE   3,1
      032332  001372                          SAVLIN

      032334  000003                  DT4:    3
      032336     006      004                 .BYTE   6,4
      032340  001214                          $REG5
      032342     006      001                 .BYTE   6,1
      032344  001212                          $REG4
```

```
032346  003  001          .BYTE  3,1
032350  001372            SAVLIN

032352  000004       DT5: 4
032354  003  005          .BYTE  3,5
032356  001372            SAVLIN
032360  006  011          .BYTE  6,9.
032362  001214            $REGS
032364  006  007          .BYTE  6,7
032366  001220            $TMP1
032370  006  001          .BYTE  6,1
032372  001400            REGIST
032374  000004       DT6: 4                    ;FOR BAUD RATE ERROR MESSAGE
032376  006  001          .BYTE  6,1
032400  024110            DCOUNT                ;HIGH COUNTER
032402  006  001          .BYTE  6,1
032404  024104            CCOUNT                ;LOW COUNTER
032406  006  001          .BYTE  6,1
032410  024106            BCOUNT                ;ACTUAL XMIT COUNT
032412  001  001          .BYTE  1,1
032414  001372            SAVLIN                ;THE LINE NUMBER ERROR OCCURED ON

                          ;TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES
                          ;---------------------------------------------

032416  002450       DLYTBL: 2450              ;TIME FOR   50 BAUD
032420  001560               1560              ;TIME FOR   75 BAUD
032422  001120               1120              ;TIME FOR  110 BAUD
032424  000750                750              ;TIME FOR  134 BAUD
032426  000660                660              ;TIME FOR  150 BAUD
032430  000330                330              ;TIME FOR  300 BAUD
032432  000150                150              ;TIME FOR  600 BAUD
032434  000060                 60              ;TIME FOR 1200 BAUD
032436  000040                 40              ;TIME FOR 1800 BAUD
032440  000030                 30              ;TIME FOR 2000 BAUD
032442  000020                 20              ;TIME FOR 2400 BAUD
032444  000010                 10              ;TIME FOR 3600 BAUD
032446  000001                  1              ;TIME FOR 4800 BUAD
032450  000001                  1              ;TIME FOR 7200 BAUD
032452  000001                  1              ;TIME FOR 9600 BAUD
032454  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.

032456                 CORMAX:
        001512                .=MANTO
001512  100000                100000
        000001          .END
```

```
ABASE = 160010            796#  1208  1249
ACDW1 = 000000           1208  1251
ACDW2 = 000000           1208  1252
ACPUOP= 000000           1208  1223
ACTIVE  001412           1311#  1642*  2816*  2817  2819*  2824  2825*  2826  2829*
ADDW0 = 000000           1208  1253
ADDW1 = 000000           1208  1254
ADDW10= 000000           1208  1263
ADDW11= 000000           1208  1264
ADDW12= 000000           1208  1265
ADDW13= 000000           1208  1266
ADDW14= 000000           1208  1267
ADDW15= 000000           1208  1268
ADDW2 = 000000           1208  1255
ADDW3 = 000000           1208  1256
ADDW4 = 000000           1208  1257
ADDW5 = 000000           1208  1258
ADDW6 = 000000           1208  1259
ADDW7 = 000000           1208  1260
ADDW8 = 000000           1208  1261
ADDW9 = 000000           1208  1262
ADEVCT= 000000           1208  1214
ADEVM = 000000           1208  1250
ADRCNT  006645           2334*  2371*  2381#
ADVANC= 104400           1521#  3080  3377  3589  3603  3650  3659  3810  4413  4538  4795  4843  4970
                         5038  5087  5159  5187  5202  5215
AENV  = 000000           1208  1219
AENVM = 000000           1208  1220
AFATAL= 000000           1208  1211
AMADR1= 000000           1208  1236
AMADR2= 000000           1208  1240
AMADR3= 000000           1208  1243
AMADR4= 000000           1208  1246
AMAMS1= 000000           1208  1230
AMAMS2= 000000           1208  1238
AMAMS3= 000000           1208  1241
AMAMS4= 000000           1208  1244
AMSGAD= 000000           1208  1216
AMSGLG= 000000           1208  1217
AMSGTY= 000000           1208  1210
AMTYP1= 000000           1208  1231
AMTYP2= 000000           1208  1239
AMTYP3= 000000           1208  1242
AMTYP4= 000000           1208  1245
APASS = 000000           1208  1213
APRIOR= 000000           1208
APTCSU= 000040           2152  2276#
APTENV= 000001           2145  2232  2274#  2563
APTSIZ= 000200           2273#
APTSPO= 000100           2147  2234  2275#
ASWREG= 000000           1208  1221
ATESTN= 000000           1208  1212
AUNIT = 000000           1208  1215
AUSWR = 000000           1208  1222
AUTO.S  012264           1907  2924#
AVECT = 000300            796#
```

```
AVECT1= 000000          1208    1247
AVECT2= 000000          1208    1248
BAUD    025462          5446#   5766
BCOUNT  024106          5066*   5093#   5150*   5156*   5176*   5184*   5189    5191    5972
BITMRO  007120          2457#
BIT0    = 000001        913#    993     995     997     999     1036    1055    3433    3605    4934    5002
BIT00   = 000001        903#    913
BIT01   = 000002        902#    912
BIT02   = 000004        901#    911
BIT03   = 000010        900#    910
BIT04   = 000020        899#    909
BIT05   = 000040        898#    908
BIT06   = 000100        897#    907
BIT07   = 000200        896#    906
BIT08   = 000400        895#    905
BIT09   = 001000        894#    904
BIT1    = 000002        912#    994     995     998     999     1037    1056    3433
BIT10   = 002000        893#    963     964     965     966     984     985     986     987     1021    1022    1023    1024
                        1029    1030    1031    1032    1046    1065    1077    3433
BIT11   = 004000        892#    1025    1026    1027    1028    1029    1030    1031    1032    1047    1066    1078    2083
                        3433    3522
BIT12   = 010000        891#    949     972     1015    1048    1067    1079    5632
BIT13   = 020000        890#    950     973     1049    1068    1080    5536    5629
BIT14   = 040000        889#    951     974     1050    1069    1081    2065    3017    5626
BIT15   = 100000        888#    952     975     1051    1070    1082    1766    1769    2834    2967
BIT2    = 000004        911#    996     997     998     999     1038    1057    1962    3433
BIT3    = 000010        910#    944     1002    1004    1006    1008    1039    1058    4896    4898
BIT4    = 000020        909#    945     1003    1004    1007    1008    1040    1059    4896
BIT5    = 000040        908#    946     1005    1006    1007    1008    1013    1041    1060    2936    2946    3017
BIT6    = 000100        907#    947     1010    1042    1061
BIT7    = 000200        906#    948     1011    1043    1062    1905    2937    2944    3019
BIT8    = 000400        905#    960     962     964     966     981     983     985     987     1018    1020    1022    1024
                        1026    1028    1030    1032    1044    1063    1075    3433
BIT9    = 001000        904#    961     962     965     966     982     983     986     987     1019    1020    1023    1024
                        1027    1028    1031    1032    1045    1064    1076    3433
BPTVEC= 000014          920#
BRK0    = 000400        1075#
BRK1    = 001000        1076#
BRK2    = 002000        1077#
BRK3    = 004000        1078#
BRK4    = 010000        1079#
BRK5    = 020000        1080#
BRK6    = 040000        1081#
BRK7    = 100000        1082#
BRW     005372          1967    2103#
BUILD   024754          5294#   5333
BYTCNT  027424          5813#
CCOUNT  024104          5067*   5089*   5092#   5149*   5152    5177*   5179    5209*   5972
CHRCNT  007116          2423*   2436*   2454#
CLKSRV  024076          5064    5075    5089#
CNVRT   = 104412        1551#   1992    1994    1997    2000    2542    2544    2546    2617
CONVRT= 104411          1548#   1917    2560
CORMAX  032456          5972#
COUNT0  024632          4944*   4965*   5012*   5033*   5246*   5254#
COUNT1  024634          4945*   4967*   5013*   5035*   5247*   5255#
COO     = 000400        1063#
```

```
C01    = 001000          1064#
C02    = 002000          1065#
C03    = 004000          1066#
C04    = 010000          1067#
C05    = 020000          1068#
C06    = 040000          1069#
C07    = 100000          1070#
CR     = 000015           828#  2210    2220
CRLF   = 000200           829#  2162    2220
CSRMAP  012272          2927#
CYCLE   011542          1968    2022    2805#
DATABP  007512          2531*   2534    2558    2561#
DATAHD  007500          2530*   2554    2557#
DCLASM= 104417          1566#   3740    3836    3926    4018    4088    4171    4260    4377    4436    4575    4664    4846
                        4926    4994
DCLR   = 000020           945#  2477    2478    3100    3475    3487    3495
DCOUNT  024110          5094#   5154*   5210*   5972
DDISP  = 177570           835#  1179
DELAY  = 104414         1557#   2954    3451    3615    3621    3665    3671    3705    3763    3788    3804    3865    3890
                        3941    3964    3982    4057    4116    4141    4200    4227    4297    4305    4400    4407    4468
                        4484    4494    4609    4629    4964    5032    5498    5584
DEVADR  006642          2332*   2368    2379#
DEVICE= 104413          1554#   2484    3150    3188    3226    3264    3302    3343    3394    3473    3517    3548    3591
                        3693    3714    3985    4152    4238    4776    5454    5487    5508    5511    5551    5573    5593
                        5617
DH1     032045          5876    5972#
DH2     032070          5880    5972#
DH3     032123          5884    5888    5900    5904    5908    5912    5916    5920    5928    5932    5936    5948    5952
                        5956    5960    5964    5972#
DH4     032135          5892    5896    5944    5972#
DH5     032164          5940    5972#
DH6     032243          5968    5972#
DISPLA  001162          1179#   1658*   2095*   2573*
DISPRE  000174          1141#   1658
DLYCNT  007214          1798*   1892*   1943    2490    2495#   4691
DLYTBL  032416          1892    5972#
DONFLG  001420          1320#   3924*   3956    3978*
DSWR   = 177570           834#  1178
DTR0   = 000400          1044#  3373
DTR1   = 001000          1045#
DTR2   = 002000          1046#
DTR3   = 004000          1047#
DTR4   = 010000          1048#
DTR5   = 020000          1049#
DTR6   = 040000          1050#
DTR7   = 100000          1051#
DT1     032276          5877    5972#
DT2     032310          5881    5972#
DT3     032326          5885    5889    5901    5905    5909    5913    5917    5921    5929    5933    5937    5949    5953
                        5957    5961    5965    5972#
DT4     032334          5893    5897    5945    5972#
DT5     032352          5941    5972#
DT6     032374          5969    5972#
DVALID= 100000           975#  3522    3784    3886    4053    4513    5236
DZACTV  001404          1306#   1811*   1932*   1933    2805    2811    2893*   2984*   2988*   2993    3012
DZCR0   001500          1370#   1701    1705    1812    2990
```

```
DZCR1   001514          13790#  1813
DZCR10  001640          14420#
DZCR11  001654          14510#
DZCR12  001670          14600#
DZCR13  001704          14690#
DZCR14  001720          14780#
DZCR15  001734          14870#
DZCR16  001750          14960#
DZCR17  001764          15050#
DZCR2   001530          13880#
DZCR3   001544          13970#
DZCR4   001560          14060#
DZCR5   001574          14150#
DZCR6   001610          14240#
DZCR7   001624          14330#
DZCSR   002042          15750#  2026    2477#   2478    2485#   3055    3099    3135    3173    3211    3249    3287    3431
                        3474    3694    3753#   3757    3761    3786    3807    3814    3855#   3859    3863    3888    3937#
                        3939    3967    3988    4031#   4055    4108#   4111    4126#   4132#   4136    4191#   4194    4206#
                        4211#   4217#   4221    4236#   4276#   4290    4295    4303    4309    4314#   4315#   4356#   4389#
                        4398    4404#   4410    4454#   4461    4466    4476    4482    4518    4544#   4548#   4553#   4595#
                        4607    4617    4623    4627    4683#   4684#   4685#   4706    4731    4739    4880#   4882    4958#
                        4962    5026#   5030    5168#   5181    5248    5252#   5285#   5304    5346#   5427    5430    5493#
                        5496    5520#   5530    5548#   5556    5562#   5597#   5620    5655    5847#
DZLEV   030100          2838    5828#
DZLPR   002052          15790#  3443#   3520#   3528#   3746#   3842#   3930#   4025#   4094#   4177#   4266#   4383#   4442#
                        4581#   4671#   4849#   4860#   4937#   5005#   5175#   5432#   5492#   5516#   5604#   5852#
DZLV0   001504          1372#   1730    1734    1766#   1769#
DZLV1   001520          13810#
DZLV10  001644          14440#
DZLV11  001660          14530#
DZLV12  001674          14620#
DZLV13  001710          14710#
DZLV14  001724          14800#
DZLV15  001740          14890#
DZLV16  001754          14980#
DZLV17  001770          15070#
DZLV2   001534          13900#
DZLV3   001550          13990#
DZLV4   001564          14080#
DZLV5   001600          14170#
DZLV6   001614          14260#
DZLV7   001630          14350#
DZMSR   002062          15830#  3073    3549    3590    3653    5439#   5585    5861#
DZNUM   001410          13080#  1639    1803#   2003    2932#   2973#   2974    2980    2982
DZPRT   030306          2832#   2833    2834#   4100#   4106    4107    4130    4131    4189    4190    4215    4216    4273
                        4275    4877    4878    4956    4957    5024    5025    5828#   5829#   5830#   5831#   5832#   5833
                        5867#
DZRBUF  002046          15770#  3061    3518    3792    3853    3894    3971    3992    4032    4061    4287    4325    4514
                        4528    4600    4734    5221    5232    5307    5312    5430#   5431#   5432    5433    5533    5623
                        5851#
DZRIS   002074          15890#  4106#   4130#   4189#   4215#   4273#   4357    4358#   4594#   4680#   4791    4792#   4877#
                        4956#   5024#   5518#   5599#   5838#
DZRIV   002072          15880#  2029    2831#   4105#   4129#   4188#   4214#   4272#   4357#   4593#   4622#   4679#   4791#
                        4876#   4955#   5023#   5419    5517#   5598#   5836
DZTCR   002056          15810#  3067    3324    3372    3444#   3700#   3756#   3797#   3858#   3901#   3936#   3954#   3962#
                        4035#   4068#   4101#   4184#   4286#   4349#   4396#   4460#   4532#   4547#   4605#   4643#   4686#
```

```
                      4726* 4879* 4960* 5028* 5172* 5283* 5433* 5434* 5435  5437  5489* 5555* 5563*
                      5606* 5857*
DZTDR   002066        1585# 3551* 3558* 3767* 3816* 3869* 3959* 4037* 4134* 4219* 4301* 4405* 4472*
                      4556* 4613* 4625* 4723* 5183* 5270* 5437* 5438* 5439  5503* 5561* 5658* 5862*
DZTIS   002100        1591# 4107* 4131* 4190* 4216* 4275* 4359  4360* 4682* 4793  4794* 4878* 4957*
                      5025* 5601* 5842*
DZTIV   002076        1590# 4104* 4128* 4187* 4213* 4274* 4359* 4545* 4681* 4793* 4875* 4954* 5022*
                      5600* 5840*
DZV   - ****** U      2777
DZVC0   001502        1371# 1716  1720  3033
DZVC1   001516        1380#
DZVC10  001642        1443#
DZVC11  001656        1452#
DZVC12  001672        1461#
DZVC13  001706        1470#
DZVC14  001722        1479#
DZVC15  001736        1488#
DZVC16  001752        1497#
DZVC17  001766        1506#
DZVC2   001532        1389#
DZVC3   001546        1398#
DZVC4   001562        1407#
DZVC5   001576        1416#
DZVC6   001612        1425#
DZVC7   001626        1434#
DZ.END  002000        1513# 1685  2817  2826  2930
DZ.MAP  001500        1311  1367# 1642  1682  1912  2819  2829  2879  2927  2933  3004
EIAFLG  001414        1316# 2833* 3375  3587  3651
EIGHT   000030        1004# 4829  4832
EIGHTS= 000070        1008#
EMTVEC= 000030         923# 1633* 1634*
EM1     030540        5875  5972#
EM10    031155        5911  5972#
EM13    031214        5915  5972#
EM14    031245        5919  5972#
EM15    031277        5923  5972#
EM16    031341        5927  5972#
EM17    031412        5931  5972#
EM2     030613        5879  5972#
EM20    031450        5935  5972#
EM21    031511        5939  5972#
EM22    031541        5943  5972#
EM23    031603        5947  5972#
EM24    031633        5951  5972#
EM25    031661        5955  5972#
EM26    031711        5959  5972#
EM27    031740        5963  5972#
EM3     030641        5883  5972#
EM30    032010        5967  5972#
EM4     030700        5887  5972#
EM5     030727        5891  5972#
EM6     030756        5895  5972#
EM7     031014        5899  5972#
EM8     031055        5903  5972#
EM9     031117        5907  5972#
ENDTTB= 024206        5142# 5195
```

```
ERRMSG  007466          2529* 2549  2552#
ERRVEC- 000004           916# 2070  2071* 2073* 2076*
ERTABO  007636          2544  2584#
EVEPAR- 000000          1014#
EXITER  007572          2572  2575#
FINI    022444          4702  4790#
FIVE  - 000000          1001#
FIVES - 000040          1005#
FRMERR- 020000           973# 4053  4745  5310
HALTS   007516          2514  2563#
HDRFLG  001416          1318# 1683* 1853* 1908  1910*
HDZCSR  002044          1576# 3488* 3945  4709  5261  5849*
HDZLPR  002054          1580# 5855*
HDZMSR  002064          1584# 5864*
HDZRBU  002050          1578# 5854*
HDZTCR  002060          1582# 3614* 3620* 3664* 3670* 5435* 5436* 5579* 5859*
HDZTDR  002070          1586# 4036* 4065* 5865*
HILIM   006640          2331* 2359  2378#
HT    - 000011           826# 2160  2220
INBUF   011272          2287  2337  2743* 2744  2749  2754  2770# 5541* 5542* 5549  5680  5732* 5733
                        5737
INIFLG  00_415          1317# 1661  1666* 1677  1687* 1837* 1903  2926*
INIT    021724          4669# 4674
INIT1   021750          4676# 4678
INSTER- 104404          1533# 2353  2760  5699  5742
INSTR - 104403          1530# 1696  1711  1725  1776  1785  1844  1879  1938  2845  5414  5422  5442
                        5446  5456
INSTR2  006440          2308  2320#
INTRAN  026704          5600  5655#
INTREC  026524          5598  5619#
INTSVC  026070          5517  5530#
IOTVEC- 000020           921# 1631* 1632* 3034*
LAST    027416          5404* 5667* 5810#
LESS1   030310          4183* 4207* 4687* 5519* 5605* 5833* 5834* 5835* 5868#
LF    - 000012           827# 2214  2220
LIMITS  006566          2347  2359#
LINE    001364          1298# 2835* 3598  3655  3698  3744  3754  3840  3856  3935  4023  4033  4092
                        4101  4175  4184  4264  4278  4381  4391  4440  4455  4579  4597  4666  4669
                        4686  4719  4753  4780  4830  4932  4960  5000  5028
LINESP  027426          5451  5761  5814#
LINEX   025500          5454# 5526  5613
LINE0   001506          1373# 1795* 1849  1860  1888
LINE1   001522          1382#
LINE10  001646          1445#
LINE11  001662          1454#
LINE12  001676          1463#
LINE13  001712          1472#
LINE14  001726          1481#
LINE15  001742          1490#
LINE16  001756          1499#
LINE17  001772          1508#
LINE2   001536          1391#
LINE3   001552          1400#
LINE4   001566          1409#
LINE5   001602          1418#
LINE6   001616          1427#
```

```
LINE7   001632          1436#
LOBITS  006644          2333*  2363   2380#
LOCK    001362          1293#  2112   2114   2501*  2538   3054*  3060*  3066*  3072*  3326*  3355*  3374*  3413*
                        3586*  3647*  3737*  3833*  3907*  4014*  4071*  4433*  4572*  4649*  4824*  5578*
LOCKUP  027414          5468*  5614*  5619*  5809#
LOLIM   006636          2330*  2361   2377#
LP0   = 000000           992#
LP1   = 000001           993#
LP2   = 000002           994#
LP3   = 000003           995#
LP4   = 000004           996#
LP5   = 000005           997#
LP6   = 000006           998#
LP7   = 000007           999#
MAINT = 000010           944#  2101   2752   3136   3476   3489
MANT0   001512          1375#  1779   1858   2991   5972
MANT1   001526          1384#
MANT10  001652          1447#
MANT11  001666          1456#
MANT12  001702          1465#
MANT13  001716          1474#
MANT14  001732          1483#
MANT15  001746          1492#
MANT16  001762          1501#
MANT17  001776          1510#
MANT2   001542          1393#
MANT3   001556          1402#
MANT4   001572          1411#
MANT5   001606          1420#
MANT6   001622          1429#
MANT7   001636          1438#
MASTEK  010766          2540   2730#
MBADLN  011075          1867   2730#
MCABLE  027745          5475   5823#
MCHAR   027612          5521   5823#
MCSRX   010716          1991   2545   2730#
MDATA   011376          2434   2444   2774#
MEPASS  010535          1990   2730#
MERRPC  011043          2543   2730#
MERRX   010743          1999   2730#
MERR2   010574          2730#  2807   2995
MERR3   010643          1929   2730#
MINVAL  027536          5765   5823#
MLINE   027564          5457   5823#
MLOCK   010667          1964   2730#
MNEW    010771          1924   2730#
MNTFLG  001417          1319#  2098*  2101*  2485   2747*  2752*  2757*
MODE    001370          1300#  2099   2837*  3594   3648   3770   3872   4015   4040   4330   4500   4850   4923
                        4991
MPASS   027522          5666   5823#
MPASSX  010732          1996   2730#
MPFAIL  010472          2723   2730#
MQUICK  027762          5495   5823#
MR      010560          1969   2730#
MREGAD  027466          5423   5823#
MSENAB= 000040           946#  3174   3445   3446   3476   3489   3695   3701   3753   3855   3937   4031   4108
```

```
                    4132    4191    4217    4276    4389    4404    4454    4544    4553    4595    4685    4880    4958
                    5026    5346    5493    5520    5548    5562    5597
MSPEED  027574      5447    5823#
MTERM   027720      5481    5823#
MTITLE  001000     11500#   1665
MTSTN   010754      2541   27304    2846
MVECTO  027444      5415    5823#
MVECX   010724      1993   27304
MWHICH  027652      5443    5823#
NEXT    001360     1292#    2500    2580    3051*   3098*   3134*   3172*   3210*   3248*   3286*   3323*   3371*   3430*
                   3472*   3516*   3547*   3585*   3596    3646*   3692*   3736*   3832*   3923*   4013*   4085*   4168*
                   4257*   4374*   4432*   4571*   4662*   4823*   4922*   4990*   5058*   5510*   5572*
NOLIST= ****** U    798
NUMLIN  027432      5490    5512    5595    5751*   5817#
NUMTCR  027434      5489    5555    5579    5581    5583    5606    5753*   5760*   5819#
ODDPAR= 000200     1011#    4021    4929    4936    4938    4997    5004    5006
OFFSET  025214      4825*   4840*   4891*   5335    5337    5348#
ONESTO= 000000     1012#
OTHER   025442      5406    5440#
OUT     022372      4697    4775#
OVRRUN= 040000      9744    4525    4745    5310
PAR     001366     1299#    2836*   3742    3838    3927    3979    4020    4090    4173    4262    4379    4438    4577
                    4667    4928    4996    5490*   5491*   5492    5512*   5513*   5515*   5516    5594*   5595*   5596*
                    5604
PARAM = 104405     1536#    1698    1713    1727    1787    1846    1881    1940    2847    5416    5424    5458
PARAM1  006506     2336#    2354
PARER = 010000      972#    4053    4745    5230    5236    5310
PARERR  006562      2339    2341    2343   2353#    2360    2362    2364
PARESE  024460      4955    5023   5221#
PARITY= 000100     1010#    4021    4930    4998
PARMD = 104415     1560#    5448
PAR0    001510     1374#   1796*    1884    1889
PAR1    001524     1383#
PAR10   00165C     1446#
PAR11   001664     1455#
PAR12   001700     1464#
PAR13   001714     1473#
PAR14   001730     1482#
PAR15   001744     1491#
PAR16   001760     1500#
PAR17   001774     1509#    1899
PAR2    001540     1392#
PAR3    001554     1401#
PAR4    001570     1410#
PAR5    001604     1419#
PAR6    001620     1428#
PAR7    001634     1437#
PAWCH = 104416     1563#    5444
PIRQ  = 177772      833#
PIRQVE= 000240      927#
POPR0 = 012600      936#
POP1SP= 005726      934#
POP2SP= 022626      938#    1656    3030    4205    4235    4350    4355
PRIO    027436     5518     5599    5601    5820#
PR0   = 000000      850#
PR1   = 000040      851#
```

M10


```
PR2     = 000100            8520
PR3     = 000140            8530
PR4     = 000200            8540  5868
PR5     = 000240            8550  5867
PR6     = 000300            8560
PR7     = 000340            8570  1118  1122  1629*  1959*  3053  4283*  4594  4680  4682  4775*  5398*  5466*
                            5509* 5570*
PS      = 177776            8300   831
PSW     = 177776            8310
PUSHR0= 010046              9350
PUSHR1S= 005746             9330
PUSHR2S= 024646             9370
PWRVEC= 000024              9220  1637*  1638*  2692*  2693*  2702*  2708*  2720*  2721*
QUITS   026516              5617* 5649
RCVCN   = 010000            1015* 3928  4829  4832  4848  4859  5515  5596
RDATA   027422              5812*
RDONE   = 000200             948* 3433  3757  3786  3807  3859  3888  4055  4136  4221  4303  4410
RECDAT  027440              5533* 5536  5540  5541  5543*  5544*  5545  5623*  5626  5629  5632  5637  5642*
                            5643* 5644  5821*
REGIST  001400              1304* 3853*  4032*  4865*  5232*  5972
RESREG  007514              2559  2562*
RESTAR  012106              2725  2872*
RESVEC= 000010               917*
RESR5   = 104410            1545* 2562
RIE     = 000100             947* 3250  3476  3489  4132  4217  4315  4356  4595  4683  4880  4882  4958
                            4962  5026  5030  5252  5346  5520  5562  5597
RING0   = 000001            1055*
RING1   = 000002            1056*
RING2   = 000004            1057*
RING3   = 000010            1058*
RING4   = 000020            1059*
RING5   = 000040            1060*
RING6   = 000100            1061*
RING7   = 000200            1062*
RL0     = 000000             980*
RL1     = 000400             981*
RL2     = 001000             982*
RL3     = 001400             983*
RL4     = 002000             984*
RL5     = 002400             985*
RL6     = 003000             986*
RL7     = 003400             987*
RSTART  021714              4667*
RSTRT   026740              5669*
RUN     001406              1307*  1643*  2811  2814*  2815*  2822*  2823*
RXISR1  025002              4876  5304*
RXSVC   022214              4679  4731*
RXTCR   022472              4666*  4695  4769*  4796*
SAVLIN  001372              1301*  3696*  3715*  3750*  3768  3798*  3799  3846*  3870  3902*  3903  3925*  3969*
                            3975*  3990*  3994*  4029*  4038  4067*  4098*  4181*  4270*  4280*  4342  4387*  4393*
                            4446*  4458  4498  4534*  4585*  4603  4645*  4713*  4738*  4756  4778*  4786*  4826*
                            4861*  4862*  4942*  5010*  5212*  5225*  5263  5265*  5267*  5323*  5324*  5343*  5344*
                            5461  5545  5644  5751  5752  5972
SAVNUM  001411              1309*  1639*  1804*  2001*  2003*  2983*
SAVPC   001402              1305*  2386*  2586
SAV05   = 104407            1542*  2519
```

```
SCOP1 = 104401          15240  3084  3352  3400  3626  3676  3796  3817  3898  4066  4533  4554  4644
                        4890  5592
SERV.G  007652          2054  2507  25904  2613  4889  5214  5505  5525  5612
SET     027214          5464  57514
SETAPT  012112          1669  28794
SETFLG= 104406          15394  1778
SET.PS  011440          1117  27774
SET1    027246          5756  57594
SEVEN = 000020          10034
SEVENS= 000060          10074
SILOAL= 020000           9504  3433  4476  4482  4518  4617  4627  4739
SILOEN= 010000           9494  3212  3476  4454  4544  4553  4595
SIX   = 000010          10024
SIXS  = 000050          10064
SNAP    022044          46904  4770
SPACNT  007117          24554
SPEED   027430          5491  5513  5594  5768*  58154
SPIN    026464          56094  5615  5652
STACK = 001120           8214  1628  1958  2581  5397  5467
STFLG   027412          5400*  5455*  5472  5474*  5478  5480*  58084
STKLMT= 177774           8324
STOP    001462          13394  3850  4450  4589  4677  4835  4868  4950  5018
SVOS    006654          23904
SWR     001160          11784  1653  1657*  1675  1679  1799  1833  1905  1922  1927  1932  1935  1962
                        2065  2083  2110  2117  2508  2513  2571  2576  2578  2590  2614  2615*  2640*
                        2645*  2646*  2647*  2654*  2700  2713*  2841  2912*  5405  5440  5506
SWREG   000176          11424  1657
SW0   = 000001           8854
SW00  = 000001           8754  885  1675  5405
SW01  = 000002           8744  884  2841  5440
SW02  = 000004           8734  883
SW03  = 000010           8724  882  1799  1833
SW04  = 000020           8714  881  1935
SW05  = 000040           8704  880
SW06  = 000100           8694  879  1922
SW07  = 000200           8684  878
SW08  = 000400           8674  877  2576
SW09  = 001000           8664  876  2110
SW1   = 000002           8844
SW10  = 002000           8654  2578
SW11  = 004000           8644
SW12  = 010000           8634  2117  2508
SW13  = 020000           8624  2513
SW14  = 040000           8614
SW15  = 100000           8604
SW2   = 000004           8834
SW3   = 000010           8824
SW4   = 000020           8814
SW5   = 000040           8804
SW6   = 000100           8794
SW7   = 000200           8784
SW8   = 000400           8774
SW9   = 001000           8764
S110  = 001000          10134
S1200 = 003400          10244
S134  = 001400          10204
```

```
S150  = 002000          10210#
S1800 = 004000          10250#
S19200= 007400          10320#
S2000 = 004400          10260#
S2400 = 005000          10270#
S300  = 002400          10220#
S3600 = 005400          10280#
S4800 = 006000          10290#
S50   = 000000          10170#  4829    4832
S600  = 003000          10230#
S7200 = 006400          10300#
S75   = 000400          10180#
S9600 = 007000          10310#
TABLE2  027310          5759    5774#
TBITVE= 000014           918#
TBUF    027442          5540#   5561    5822#
TCCONT  024206          5071    5083    5148#
TCR0  = 000001          10360#  3325    3444
TCR1  = 000002          10370#
TCR2  = 000004          10380#
TCR3  = 000010          10390#
TCR4  = 000020          10400#
TCR5  = 000040          10410#
TCR6  = 000100          10420#
TCR7  = 000200          10430#
TDATA   027420          5811#
TD0     001422          13230#  3848    3869    3885    3899#   4134    4219    4448    4472    4473#   4587    4613    4614#
                        4615    4625    4675    4723    4724#   4833    4866    4948    5016    5270    5271#   5275
TD1     001424          13240#
TD2     001426          13250#
TD3     001430          13260#
TD4     001432          13270#
TD5     001434          13280#
TD6     001436          13290#
TD7     001440          13300#
TEIGHT  002140          16140#
TEMP    011334          27720#
TEST1   025630          5477    5487#   5507
TEST2   026266          5471    5570#
TFIVE   002146          16170#
TIE   = 040000           951#   3288    3476    4108    4126    4191    4206    4211    4314    4356    4548    4553    4684
                        4880    4958    5026    5248    5285    5597
TKVEC = 000060           925#
TLAST = 023730          2866    5972#
TL0   = 000000           959#
TL1   = 000400           960#
TL2   = 001000           961#
TL3   = 001400           962#
TL4   = 002000           963#
TL5   = 002400           964#
TL6   = 003000           965#
TL7   = 003400           966#
TMTBL   002102          15984#  5336#   5340    5342
TPVEC = 000064           926#
TRAPVE= 000034           924#   1635#   1636#
TRDY  = 100000           952#   3438    3442    3695    3703    3761    3814    3863    4111    4194    4466    4607    4623
```

```
TRTVEC• 000014            9190
TR0     001442           13310#  4758    4760    4766•   4767    4782    5235    5238    5240    5244•   5318    5320    5326•
                         5328
TR1     001444           13320#
TR2     001446           13330#
TR3     001450           13340#
TR4     001452           13350#
TR5     001454           13360#
TR6     001456           13370#
TR7     001460           13380#
TSEVEN  002142           16150#
TSIX    002144           16160#
TST1    013046            2056#  2854    2870    30480#
TST10   013764            3286#  33200#
TST11   014122            3323#  33680#
TST12   014304            3371#  34270#
TST13   014422            3430#  34690#
TST14   014552            3472#  35130#
TST15   014642            3516#  35440#
TST16   014726            3547#  35820#
TST17   015122            3585#  36430#
TST2    013236            3052#  30950#
TST20   015260            3646#  36890#
TST21   015404            3692#  37330#
TST22   015742            3736#  38290#
TST23   016270            3832#  39200#
TST24   016622            3923#  40100#
TST25   017100            4013#  40820#
TST26   017410            4085#  41650#
TST27   017736            4168#  42540#
TST3    013322            3098#  31310#
TST30   020370            4257#  43710#
TST31   020556            4374#  44290#
TST32   021304            4432#  45680#
TST33   021666            4571#  46590#
TST34   022474            4662#  48190#
TST35   023204            4823#  49190#
TST36   023456            4922#  49870#
TST37   023730            4990#  50550#  5972
TST4    013414            3134#  31690#
TST40 - ****** U          5058#
TST5    013506            3172#  32070#
TST6    013600            3210#  32450#
TST7    013672            3248#  32830#
TTABLE  024112           51250#  5173
TTST    005140            1965•  1967•  20580#
TWOSTO• 000040           10130#  4829    4832
TXSVC   022116            4681#  47060#
TYPDAT  007502            2535#  2555#   25580#
TYPE  - 104402           15270#  1665    1743    1866    1867    1911    1924    1929    1964    1969    1990    1991    1993
                          1996    1999    2165    2285    2318    2411    2444    2536    2537    2540    2541    2543    2545
                          2547    2551    2556    2616    2618    2676    2722    2807    2843    2863    2868    2995    5475
                          5481    5521    5665    5765
TYPMSG  007372            2533#  25360#
T110    002106           16010#
T1200   002120           16060#
```

```
T134     002110          16020
T150     002112          16030
T1800    002122          16070
T2000    002124          16080
T2400    002126          16090
T300     002114          16040
T3600    002130          16100
T4800    002132          16110
T50      002102          15990
T600     002116          16050
T7200    002134          16120
T75      002104          16000
T9600    002136          16130
VECMAP   012654          2994    30020
VEC1     025260          54050
VEC2     025270          54070
WCHFLG   027410          5445    5469    58070
WRDCNT   007114          2419*   2445*   24530
WTBS.F   007470          2550    25530
XBEGIN   025530          5441    54660   5669
XBX      007260          2509    2511    25130
XCSR     005072          1992    20240   2546
XEOP     026724          5510    5552    5572    5618    56650
XERR     005114          2000    20330
XHEAD    011050          1911    27300
XMTCNT   001376          13030   4874*   4947*   5015*   5275
XMTLIN   001374          13020   4827*   4837*   4839    4864
XMTSRV   024636          4875    4954    5022    52610
XPASS    005106          1997    20300
XSTART   025216          1146    53970   5399
XSTATQ   011140          1918    27300
XTCR0    027222          57520
XTCR1    027236          57550   5758
XTSTN    007644          2542    25870
XVEC     005100          1994    20270
XX     = 160210          13680   13770   13860   13950   14040   14130   14220   14310   14400   14490   14580   14670   14760
                         14850   14940   15030   15120
YY     = 000500          13680   13770   13860   13950   14040   14130   14220   14310   14400   14490   14580   14670   14760
                         14850   14940   15030   15120
ZZ     = 000020          13680   13770   13860   13950   14040   14130   14220   14310   14400   14490   14580   14670   14760
                         14850   14940   15030   15120
$APTHD   001462          1350    13560
$ASTAT=  ****** U        2254    2269
$ATYC    006030          2225    22270
$ATY1    006004          22230
$ATY3    006012          2150    22240
$ATY4    006022          22260   2566
$AUTOB   001154          11750
$BASE    001310          12490   1705*   2830*   2880    2990*   5846
$BDADR   001142          11700
$BDDAT   001146          11720
$CDW1    001314          12510   2904    2991*
$CDW2    001316          12520
$CHARC   006000          2167*   2177*   2184    2212*   22170
$CMTAG   001120          11580
$CM1   = 000006          11900   11910   11920   11930   11940   11950   11960
```

```
$CM2  - 000014  11900  11910  11920  11930  11940  11950  11960
$CM3  - 000006  11880  1190
$CM4  - 000004  11960  11970  11980  11990  12000
$CPUOP  001262  12230
$CRAP - 177777   7960  30410  30460  30880  30930  31220  31290  31600  31670  31980  32050  32360  32430
                32740  32810  33120  33180  33590  33660  34170  34250  34580  34670  35060  35110  35370
                35420  35680  35790  36300  36410  36810  36870  37210  37310  38210  38270  39110  39180
                39990  40080  40740  40800  41570  41630  42430  42520  43620  43690  44150  44270  45590
                45660  46510  46570  48000  48170  49050  49170  49740  49850  50410  50530
$CRLF  001231  12020  2166  2220  2411  2536  2537  2547  2676  2843  2863
$DDM0  001320  12530  1814  2891  2928
$DDM1  001322  12540
$DDM10  001344  12630
$DDM11  001346  12640
$DDM12  001350  12650
$DDM13  001352  12660
$DDM14  001354  12670
$DDM15  001356  12680
$DDM2  001324  12550
$DDM3  001326  12560
$DDM4  001330  12570
$DDM5  001332  12580
$DDM6  001334  12590
$DDM7  001336  12600
$DDM8  001340  12610
$DDM9  001342  12620
$DEVCT  001244  12140
$DEVM  001312  12500  1802*  1808*  1810  1811  1927  2892  2993*
$DOAGN  005066  2002  2009  2014  20200
$E  - 000041   7960  3051  30520  3098  30990  3134  31350  3172  31730  3210  32110  3248  32490
                3286  32870  3323  33240  3371  33720  3430  34310  3472  34730  3516  35170  3547
                3548*  3585  35870  3646  36480  3692  36930  3736  37380  3832  38340  3923  39240
                4013  40140  4085  40860  4168  41690  4257  42580  4374  43750  4432  44330  4571
                4572*  4662  46630  4823  48240  4922  49230  4990  49910  5058  50590
$ENDAD  005056  1135  1663  20160  2069
$ENDCT  005042  20110
$ENV  001254  12190  2145  2232  2256  2563
$ENVM  001255  12200  1667  2147  2152  2234
$EOP  004712  19850  5058
$EOPCT  005034  20080  2012
$ERFLG  001223  11610  1641*  1989*  2045  2079  2081*  2106  2518*  2532  2548*  5403*  5668*
$ERMAX  001135  11670  2106
$ERROR  007230  1125  1633  25070
$ERRPC  001136  11680  1645*  1988*  2055*  2515  2517*
$ERRTB  001360  12870
$ERTTL  001132  11650  1644*  2035  2575*  5402*
$ETABL  001254  12180
$ETEND  001360  12710  1362
$FATAL  001236  12110  2260*
$FFLG  006250  2223*  2226*  2254  2263*  22710
$FILLC  001176  11860  2170  2220
$FILLS  001175  11850  2220
$GDADR  001140  11690
$GDDAT  001144  11710
$GET42  005046  20130
$HD  - 000001   808  809
```

```
$HIBTS  001462          1357#
$ICNT   001124          1162#  2087*  2088    2090*  2105
$ILLUP  010464          2692   2708   2727#
$INTAG  001155          1176#
$ITEMB  001134          1166#  2523*  2565
$LF     001232          1203#  2220
$LFLG   006247          2264*  2270#
$LPADR  001126          1163#  1647*  1968*  1970   2094*  2096   2105   2580*  2582   2861*  2862*  2870*  2872
                        3596*  3597   5399*  5471*  5477*  5482
$LPERR  001130          1164#
$MADR1  001266          1236#
$MADR2  001272          1240#
$MADR3  001276          1243#
$MADR4  001302          1246#
$MAIL   001234          1209#  1358   1362   2093   2145
$MAMS1  001264          1230#
$MAMS2  001270          1238#
$MAMS3  001274          1241#
$MAMS4  001300          1244#
$MBADR  001464          1358#
$MFLG   006246          2224*  2230   2265*  2269#
$MSGAD  001250          1216#  2240*  2243
$MSGLG  001252          1217#  2245*
$MSGTY  001234          1210#  2238   2246*  2258   2262*
$MTYP1  001265          1231#
$MTYP2  001271          1239#
$MTYP3  001275          1242#
$MTYP4  001301          1245#
$MXCNT  005374          2091   2105#  2810
$N    = 000037           796#  3041   3046   3052#  3088   3093   3099#  3122   3129   3135#  3160   3167   3173#
                        3198   3205   3211#  3236   3243   3249#  3274   3281   3287#  3312   3318   3324#  3359
                        3366   3372#  3417   3425   3431#  3458   3467   3473#  3506   3511   3517#  3537   3542
                        3548#  3568   3580   3587#  3630   3641   3648#  3681   3687   3693#  3721   3731   3738#
                        3821   3827   3834#  3911   3918   3924#  3999   4008   4014#  4074   4080   4086#  4157
                        4163   4169#  4243   4252   4258#  4362   4369   4375#  4415   4427   4433#  4559   4566
                        4572#  4651   4657   4663#  4800   4817   4824#  4905   4917   4923#  4974   4985   4991#
                        5041   5053   5059#  5972#
$NULL   001174          1184#  2172   2220
$NWTST= 000000          3047#  3094#  3130#  3168#  3206#  3244#  3282#  3319#  3367#  3426#  3468#  3512#  3543#
                        3581#  3642#  3688#  3732#  3828#  3919#  4009#  4081#  4164#  4253#  4370#  4428#  4567#
                        4658#  4818#  4918#  4986#  5054#
$OVER   005336          2060   2062   2066   2077   2089   2095#
$PASS   001242          1213#  1640*  1995*  1998*  2005*  2006*  2023   2032   2085   2106   5401*
$PASTM  001470          1360#
$PWRAD  010460          2725#
$PWRDN  010320          1123   1630   1637   2692#  2720
$PWRMG  010454          2723#
$PWRUP  010372          2702   2708#
$QUES   001230          1201#  1866   2220   2318   2868
$REGAD  001200          1188#
$REG0   001202          1190#  2395*  2400   5972
$REG1   001204          1191#  2394*  2401   5972
$REG2   001206          1192#  2393*  2402
$REG3   001210          1193#  2392*  2403
$REG4   001212          1194#  2391*  2404   5972
$REG5   001214          1195#  2390*  2405   5972
```

CROSS REFERENCE TABLE - USER SYMBOLS

```
$RTNAD  005070          2022#
$SAVR6  010470          2701*  2709    2710*  2711*  2729#
$SCOPE  005122          1631   2052#
$SETUP= 000000          2004   2053
$SVLAD  005320          2074   2092#
$SVPC = 000040          1133#  1138
$SWR  = 164000           796#   808    1200   1201   1982   2004   2015   2021   2023   2046   2047   2048   2049
                        2065   2077    2079   2080   2081   2082   2083   2095   2105   2726   3049   3096   3132
                        3170   3208    3246   3284   3321   3369   3428   3470   3514   3545   3583   3644   3690
                        3734   3830    3921   4011   4083   4166   4255   4372   4430   4569   4660   4820   4920
                        4988   5056
$SWREG  001256          1221#  2912
$SWRMK= 000000          2049
$TESTN  001240          1212#  2093*
$TIMES  001226          1200#  2004*  2082*  2088   2091*  2105   2810*  4822*
$TKB    001166          1181#  1747    1753   2061   2063   2191   2198   2220   2291   2297   2311   2593   2600
                        2621   2628    2657   2664   5524   5611
$TKS    001164          1180#  1745    1751   2059   2189   2196   2220   2289   2295   2598   2612   2619   2626
                        2655   2662    5522   5609
$TMP0   001216          1196#  4546*  4551*  4828*  4830   4841*  4879
$TMP1   001220          1197#  1744*  1790   1803   1804   1805*  1913*  2732   4872*  4884*  5296*  5336   5340
                        5972
$TMP2   001222          1198#  1810*  1816*  1817*  1914*  1915   2734   5294*  5297*
$TMP3   001224          1199#  4873*  4886*  5295*
$TN   = 000040           808#  3047   3049#  3094   3096#  3130   3132#  3168   3170#  3206   3208#  3244   3246#
                        3282   3284#  3319   3321#  3367   3369#  3426   3428#  3468   3470#  3512   3514#  3543
                        3545#  3581   3583#  3642   3644#  3688   3690#  3732   3734#  3828   3830#  3919   3921#
                        4009   4011#  4081   4083#  4164   4166#  4253   4255#  4370   4372#  4428   4430#  4567
                        4569#  4658   4660#  4818   4820#  4918   4920#  4986   4988#  5054   5056#
$TPB    001172          1183#  1765*  2209*  2220   2311*  2512*  2643*
$TPFLG  001177          1187#  2139   2220
$TPS    001170          1182#  2205   2220   2309   2510   2641
$TSTM   001466          1359#
$TSTNM  001122          1160#  1646*  2045   2092*  2093   2095   2106   2589   2850   2859   3049*  3096*  3132*
                        3170*  3208*  3246*  3284*  3321*  3369*  3428*  3470*  3514*  3545*  3583*  3644*  3690*
                        3734*  3830*  3921*  4011*  4083*  4166*  4255*  4372*  4430*  4569*  4660*  4820*  4920*
                        4988*  5056*  5488*  5571*
$TYPE   005440          2139#  2251
$TYPEC  005652          2169   2176    2183   2188#
$TYPEX  006002          2208   2213    2215   2218#
$UNIT   001246          1215#
$UNITM  001472          1361#
$USWR   001260          1222#
$VECT1  001304          1247#  1720*  1734*  1735*  1736*  1737*  1738*  1739*  2881   2884   3033*
$VECT2  001306          1248#
$XOFF = 000023           955#  1749   2193   2220   2293   2595   2623   2659
$XON  = 000021           954#  1755   1759   2061   2200   2207   2220   2299   2303   2602   2605   2630   2633
                        2666   2669
$XTSTR  005176          2057   2068#
$Y    = 000020          1514#  1521   1523#  1524   1526#  1527   1529#  1530   1532#  1533   1535#  1536   1538#
                        1539   1541#  1542   1544#  1545   1547#  1548   1550#  1551   1553#  1554   1556#  1557
                        1559#  1560   1562#  1563   1565#  1566   1568#
$$GET4= 000000          2015#
$40CAT= ****** U        2065
.     = 0015.4          1112#  1113   1116#  1120#  1133   1134#  1136#  1138#  1140#  1143#  1145#  1149#  1157#
                        1204   1306#  1308#  1321#  1346   1347#  1349#  1351#  1366#  1370#  1371#  1372#  1373#
```

```
                    13740   13750   13790   13800   13810   13820   13830   13840   13880   13890   13900   13910   13920
                    13930   13970   13980   13990   14000   14010   14020   14060   14070   14080   14090   14100   14110
                    14150   14160   14170   14180   14190   14200   14240   14250   14260   14270   14280   14290   14330
                    14340   14350   14360   14370   14380   14420   14430   14440   14450   14460   14470   14510   14520
                    14530   14540   14550   14560   14600   14610   14620   14630   14640   14650   14690   14700   14710
                    14720   14730   14740   14780   14790   14800   14810   14820   14830   14870   14880   14890   14900
                    14910   14920   14960   14970   14980   14990   15000   15010   15050   15060   15070   15080   15090
                    15100   1678    19020   2023    2105    2106    2220    22720   2620    2642    2656    2704    2728
                    27300   27710   27730   27750   2785    2809    2953    2956    2998    3022    3411    3851    4288
                    4291    4310    4343    4347    4451    4456    4462    4477    4496    4536    4590    4598    4601
                    4618    4624    4647    4694    4707    4720    4732    4740    4743    4746    4754    4768    4783
                    4836    4869    4951    5019    5142    5531    5534    5537    5546    5621    5624    5627    5630
                    5633    5640    5645    5656    58230   59720
.ADVAN  007216       1522    25000
.BEGIN  004624       19580
.CNVRT  006744       1552    24120
.CONVR  006740       1549    24110
.DCLAS  007164       1567    24840
.DELAY  007176       1558    24880
.DEVIC  007144       1555    24760
.ERRTA  030312       2528    58710
.INSTE  006426       1534    23160
.INSTR  006252       1531    22810
.INST1  006272       22850   2319
.MSG    006274       2283*   22860
.PARAM  006446       1537    23270
.PARMD  026744       1561    56720
.PAMCH  027140       1564    57310   5743
.RES05  006706       1546    24000
.SAV05  006646       1543    23860
.SCOPE  005122       1121    20540   3034
.SCOP1  005376       1525    21100
.SETFL  011152       1540    27420   2761
.START  002150       1144    16260   1647
.TRPSR  007122       1127    1635    24650
.TRPTA  002002       15200   2470
.TYPE   005422       1528    21170
.$ASTA= ****** U     2224    2227
.$X   = 001462       13460   1351
```

```
COMMEN   9280#
ENDCOM   9280#
ERROR     822#  3083  3105  3118  3141  3147  3154  3179  3185  3192  3217  3223  3230  3255  3261
          3268  3293  3299  3306  3332  3340  3350  3383  3391  3399  3412  3437  3441  3454  3480
          3486  3494  3502  3525  3532  3555  3562  3619  3625  3669  3675  3708  3712  3759  3766
          3791  3795  3809  3861  3868  3893  3897  3944  3970  3976  3991  3995  4060  4064  4119
          4121  4124  4144  4146  4149  4198  4203  4209  4225  4230  4232  4289  4292  4300  4308
          4311  4319  4320  4344  4348  4353  4403  4412  4463  4471  4478  4487  4517  4520  4531
          4612  4619  4632  4637  4640  4700  4701  4708  4721  4733  4741  4744  4747  4755  4765
          4784  4888  4969  5037  5213  5228  5237  5243  5266  5306  5309  5313  5325  5345  5501
          5532  5535  5538  5547  5560  5588  5616  5622  5625  5628  5631  5634  5641  5646  5657
ESCAPE   9280#
GETPRI   9280#
GETSWR   9280#
MULT     9280#
NEWTST   9280#  3047  3094  3130  3168  3206  3244  3282  3319  3367  3426  3468  3512  3543  3581
         3642  3688  3732  3828  3919  4009  4081  4164  4253  4370  4428  4567  4658  4818  4918
         4986  5054
PASEND   7960#  1986
POP      9280#  2266  2267  2713  2714
PRGEND   7960#  1972
PRGFRT   7960#   798
PUSH     9280#  2227  2229  2250  2694  2700
REPORT   7960#  9280#
SC       7960#  2053
SCOPE    8230#  1987  3048  3095  3131  3169  3207  3245  3283  3320  3368  3427  3469  3513  3544
         3582  3643  3689  3733  3829  3920  4010  4082  4165  4254  4371  4429  4568  4659  4819
         4919  4987  5055
SC1      7960#  2097
SETPRI   9280#
SETUP    9280#
SKIP     9280#
SLASH    9280#
SPACE    9280#
STARS    9280#  1131  1153  1204  1207  1343  1345  1352  1980  2042  2124  2222  2690  2706  3047
         3094  3130  3168  3206  3244  3282  3319  3367  3426  3468  3512  3543  3581  3642  3688
         3732  3828  3919  4009  4081  4164  4253  4370  4428  4567  4658  4818  4918  4986  5054
SWRSU    9280#
TYPBIN   9280#
TYPDEC   9280#
TYPNAM   9280#
TYPNUM   9280#
TYPOCS   9280#
TYPOCT   9280#
TYPTXT   9280#
$BUFFE   7960#  2766
$CYCLE   7960#  2795
$EOP     7960#  1973
$GETFL   7960#  1775
$GETPA   7960#  1694  1709  1724  1784  1842  1877  1937  2844
$HEADE   7960#   809
$INTSE   7960#  4102  4127  4185  4212  4953  5021
$JUNK    7960#  1368  1377  1386  1395  1404  1413  1422  1431  1440  1449  1458  1467  1476  1485
         1494  1503
$LINEU   7960#  3738  3834  4086  4169  4258  4375  4434  4573
$LVLTS   7960#  4072  4155
```

```
$MRESE   7960  3739  3835  4017  4087  4151  4170  4237  4259  4376  4435  4574  4663  4844  4925
         4993
$MRR     7960  3415
$MRRW    7960  3120  3158  3196  3234  3272
$MRWD    7960  3504  3535
$MSG     7960  2730
$PARTS   7960  4903  4972
$SCOPE   7960  2036
$SETFL   7960  2736
$STAG    7960  3769  3871  4039  4329  4499
$STAGF   7960
$TCR     7960  3310  3357
$TLINE   7960  3679
$TRPDE   7960  1521  1524  1527  1530  1533  1536  1539  1542  1545  1548  1551  1554  1557  1560
         1563  1566
$TSTN    7960  3046  3093  3129  3167  3205  3243  3281  3318  3366  3425  3467  3511  3542  3580
         3641  3687  3731  3827  3918  4008  4080  4163  4252  4369  4427  4566  4657  4817  4917
         4985  5053
$UNIBU   7960  3039
$VARIA   7960  1148
$XZ      7960  3040  3045  3087  3092  3121  3128  3159  3166  3197  3204  3235  3242  3273  3280
         3311  3317  3358  3365  3416  3424  3457  3466  3505  3510  3536  3541  3567  3578  3629
         3640  3680  3686  3720  3730  3820  3826  3910  3917  3998  4007  4073  4079  4156  4162
         4242  4251  4361  4368  4414  4426  4558  4565  4650  4656  4799  4816  4904  4916  4973
         4984  5040  5052
$$CMRE   11510  1190  1191  1192  1193  1194  1195
$$CMTM   11510  1196  1197  1198  1199
$$ESCA    9280
$$NEWT    9280  3047  3094  3130  3168  3206  3244  3282  3319  3367  3426  3468  3512  3543  3581
          3642  3688  3732  3828  3919  4009  4081  4164  4253  4370  4428  4567  4658  4818  4918
          4986  5054
$$SKIP    9280
.EQUAT    7960   818
.HEADE    7960   799
.SETUP    7960
.$ACT1    7960  1129
.$APTB    7960  12050
.$APTH    7960  1341
.$APTY    7960  2220
.$CATC    7960
.$CMTA    7960  1151
.$EOP     7960  1978
.$ERRO    7960
.$POWE    7960  2688
.$SCOP    7960  2040
.$TRAP    7960
.$TYPE    7960  2121
```

```
. ABS.  032456     000

 ERRORS DETECTED:  0

 CZDZAH,CZDZAH/CRF/SOL/NL:TOC=CZDZAH.P11
 RUN-TIME: 24 19 2 SECONDS
 RUN-TIME RATIO: 132/46=2.8
 CORE USED:  38K  (75 PAGES)
```