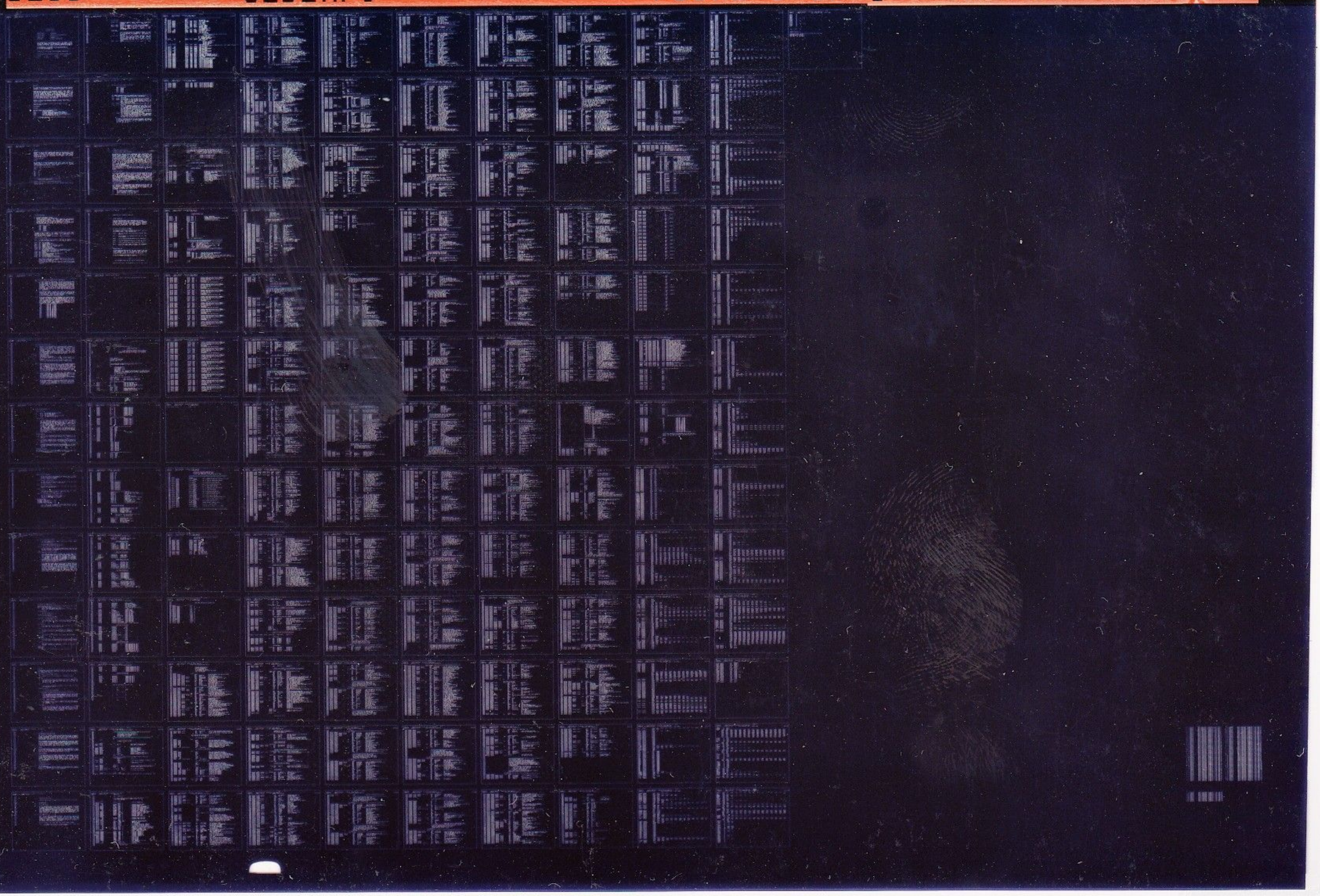
DZ11 DZ11 LN ASYNC MUX TST AH-8783F-MC
CZDZAF0 FICHE 1 OF 1



```
.REM %
```

IDENTIFICATION

PRODUCT CODE: AC-8781F-MC

PRODUCT NAME: CZDZAF0 DZ11 LN ASYNC MUX TSTS

PRODUCT DATE: NOVEMBER, 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

```
                        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))
                        H3271           STAGGERED TURNAROUND CONNECTOR FOR EIA MODULE.
                        H3190           STAGGERED TURNAROUND CONNECTOR FOR 20MA MODULE.
                        H325            CABLE TURNAROUND AND DIST PNL TESTING FOR EIA MODULE.
                        H315            THIS MAY BE SUBSITUTED FOR H325.

                        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)
```

```
                        4.      STARTING PROCEEDURE

                                A.  SET SWITCH REGISTER TO 000200
                                B.  DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
                                C.  SET SWR TO ZERO FOR 'AUTO SIZING'  OR SET SW00=1 FOR  USER PARAMETER
                                    INPUT FROM CONSOLE TERMINAL.   ON  FIRST  START IF SW07=1 AND SW00=0
                                    THE PROGRAM WILL DEFAULT TO CONSOLE PARAMETER INPUT (SW00=1).
                                D.  DEPRESS 'START KEY' AND RELEASE, THE PROGRAM WILL TYPE MAINDEC  NAME
                                    AND  PROGRAM  NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM OR
                                    PARAMETERS WERE CHANGED BY SW00=1) AND ALSO THE FOLLOWING:

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

                                THE ABOVE IS ONLY AN EXAMPLE!  THIS  WOULD  INDICATE  THE  STATUS  TABLE
                                STARTING  AT  ADD.   1 00  IN  THE  PROGRAM.   THE  STATUS TABLE MUST BE
                                VERIFIED BY THE USER IF AUTO SIZING IS DONE.  FOR INFORMATION OF  STATUS
                                TABLE SEE SECTION 8.4 FOR HELP.
                                THE PROGRAM WILL TYPE 'RUNNING'' AND PROCEED TO RUN THE DIAGNOSTIC.

                        4.1     CONTROL SWITCH SETTINGS

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

                                SW 15   SET:  HALT ON ERROR
                                SW 14   SET:  LOOP ON CURRENT TEST
                                SW 13   SET:  INHIBIT ERROR PRINT OUT
                                SW 12   SET:  INHIBIT **ALL** TYPE OUT/BELL ON ERROR.
                                SW 11   SET:  INHIBIT ITERATIONS.  (QUICK PASS)
                                SW 10   SET:  ESCAPE TO NEXT TEST
                                SW 09   SET:  LOOP WITH CURRENT DATA
                                SW 08   SET:  CATCH ERROR AND LOOP ON IT
                                SW 07   SET:  NO AUTO SIZE.  IF 1ST START OF  PROGRAM AFTER  LOADING THE
                                              OPERATOR MUST INPUT ADDRESS AND VECTOR FROM CONSOLE.
                                SW 06   SET:  RESELECT DZ11'S DESIRED ACTIVE
                                SW 05   SET:  RESERVED
                                SW 04   SET:  SELECT DELAY PARAMETER (SEE SEC. 4.1.1)
                                SW 03   SET:  EXTRA PARAMETER INPUT  (SEE SEC. 4.1.1)
                                SW 02   SET:  LOCK ON SELECTED TEST
                               **SW 01  SET:  RESTART PROGRAM AT SELECTED TEST
                                *SW 00  SET:  GET USERS PARAMETERS FROM CONSOLE

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

```
183
184
185                          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 19.2 KBAUD, 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
```

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)

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.

```
326
327
328                                5.1     NORMAL START OF DIAGNOSTIC
329
330                                                ON THE FIRST START OF THE DIAGNOSTIC AT ADDRESS  200;   IF  AUTO
331                                                SIZING  IS NOT USED OR WHENEVER SW00=1;  THE FOLLOWING QUESTIONS
332                                                ARE ASKED AND MUST BE ANSWERED.
333
334                                        ''1ST CSR ADDRESS (160000:163700):  ''
335
336                                                YOU MUST TYPE IN THE FIRST DZ11  CSR  IN  THE  SYSTEM  YOU  WISH
337                                                TESTING TO BEGIN AT.  RANGE:  160000:163700
338
339                                        ''1ST VECTOR ADDRESS (300:770):  ''
340
341                                                YOU MUST TYPE IN THE VECTOR OF THE  FIRST  DZ11  IN  THE  SYSTEM
342                                                UNDER TEST.  RANGE 300:770
343
344                                        'BR LEVEL (4:6):  ''
345
346                                                TYPE  IN  THE  PRIORITY  LEVEL  OF  THE  DZ11  THAT  THE   ABOVE
347                                                INFORMATION HAS BEEN GIVEN ABOUT.  RANGE 4 OR 5 OR 6.
348
349                                        ''TYPE ''A'' FOR EIA MODULE OR 'B'' FOR 20MA (A:B):  ''
350
351                                                TYPE ''A'' IF RUNNING A DZ11-A,B,E (EIA).
352                                                TYPE 'B'' IF RUNNING A DZ11-C,D,F (20MA).
353                                                TYPING A <CR> DEFAULTS TO EIA MODULES.
354
355                                        'MAINTENANCE MODE
356                                         [EXTERNAL   <H325>-EIA ONLY      (E)]
357                                         [INTERNAL   <DZCSR03=1>          (I)]
358                                         [STAGGERED <H3271>-EIA ONLY      (S)]
359                                         [STAGGERED <H3190>-20MA ONLY     (S)]  :
360
361                                                TYPE 'E'' OR ''I'' OR ''S'' DEPENDING ON WHICH MODE YOU WISH  TO  RUN
362                                                IN.     IF  RUNNING  'EXTERNAL';   ALL  SELECTED  LINES  MUST  BE
363                                                TERMINATED BY AN H325 TEST CONNECTOR.
364
```

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

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

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

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

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

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

                                THIS GIVES THE USER A CHANCE TO CHANGE  THE  DEFAULT  BAUD  RATE
                                USED IN APP. 90 PERCENT OF THE TEST. BAUD RATE CHOICES ARE:
                                '00''(  50 BAUD),'01''(  75 BAUD),'02''( 110 BAUD),'03''( 134 BAUD),
                                '04''( 150 BAUD),'05''( 300 BAUD),'06''( 600 BAUD),'07''(1200 BAUD),
                                '10''(1800 BAUD),'11''(2000 BAUD),'12''(2400 BAUD),'13''(3600 BAUD),
                                '14''(4800 BAUD),'15''(7200 BAUD),'16''(9600 BAUD),'17''(19.2 KBAUD)
                                LOW DEFAULT BAUD RATES ARE NOT SUGGESTED SINCE THEY LENGTHEN THE
                                TIME TO COMPLETE A PROGRAM PASS DRAMATICALLY.
                            ************************************

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

```
403
404
405                          5.2     HOW TO RUN THE ''CABLE/ECHO'' TESTS.
406
407                                  NORMAL STARTING FOR THE FIRST TIME WOULD BE:  LOAD ADDRESS  210;   START
408                                  WITH THE SWR EQUAL TO 003.
409                                  NOTE:   SW00=1  ASKS FOR ''VECTOR'' AND ''CSR''
410                                          SW01=1  ASKS FOR 'WHICH TEST ECHO OR CABLE'', 'BAUD RATE'', 'LINE''
411                                          UNDER TEST.  PROGRAM WILL PRINT OUT:
412
413                                  ''VECTOR ADDRESS-''
414
415                                          YOU TYPE VECTOR WITH A <CR>.
416
417                                  ''CONTROL REGISTER ADDRESS-''
418
419                                          YOU TYPE IN DZCSR UNDER TEST.
420
421                                  ''WHICH TEST ?  ECHO OR CABLE (E OR C)''
422
423                                          LETS DO THE CABLE TEST FIRST.  **THIS TEST IS ONLY TO BE DONE ON
424                                          THE  EIA  VERSION  OF  THE DZ11 NOT THE 20MA VERSION''.  TYPE ''C''
425                                          <CR>
426
427                                  ''BAUD RATE- ''
428
429                                          TYPE EITHER 50, 110, 135, 150, 300, 600, 1200 1800, 2000,  2400,
430                                          3600, 4800, 7200, 9600 FOLLOWED BY <CR>
431
432                                  ''LINE:  ''
433
434                                          YOU TYPE THE LINE WHICH HAS  THE  H325  TEST  CONNECTOR.   (TYPE
435                                          EITHER 0, 1, 2, 3, 4, 5, 6, 7) PROGRAM WILL THEN PRINT:
436
437                                  ''CABLE TEST''
438
439                                          AND IF EVERYTHING IS WORKING;  THE FOLLOWING WILL BE PRINTED:
440
441                                  ''PASS DONE.''
442                                  ''PASS DONE.''
443                                    ETC.
444                                          TO CHANGE LINES;  HIT ANY PRINTING KEY ON YOUR CONSOLE  TERMINAL
445                                          WHILE THE PROGRAM IS RUNNING AND THE FOLLOWING WILL BE PRINTED:
446
447                                  ''LINE:  ''
448
449                                          NOW CHANGE THE H325 TEST CONNECTOR TO ANOTHER LINE AND TYPE  THE
450                                          NEW LINE.  PROGRAM WILL THEN PRINT:
451
452                                  ''CABLE TEST''
453                                  ''PASS DONE.''
454                                  ''PASS DONE.''
455                                          CONTINUE THIS OPERATION UNTIL ALL LINES ARE TESTED.
```

5.3 ECHO TEST

IF PROGRAM HAS ALREADY BEEN STARTED AT 210 AND THE VECTOR AND ADDRESS HAVE BEEN TYPED IN; JUST LOAD ADDRESS 210 AND START WITH SWR EQUAL TO 002. PROGRAM WILL PRINT:

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

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

''BAUD RATE-''

TYPE BAUD RATE AT WHICH THE TERMINAL IS SET THAT IS CONNECTED TO THE DZ11 DIST PNL. BAUD RATE CHOICES ARE: 50, 75, 110, 135, 150 300, 600 , 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600. THE PROGRAM WILL THEN PRINT:

LINE: ''

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

''TERMINAL ECHO TEST''

*** AT THIS POINT THE MESSAGE:

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

SHOULD BE PRINTED ON THE TERMINAL CONNECTED TO THE DZ11. IF THIS MESSAGE IS DESIRED TO BE CONTINIOUSLY OUTPUT; SET THE SWR TO 377 (SWR=377) WHILE IT IS BEING OUTPUT OR WHEN THE LINE NO. IS REQUESTED ABOVE. WHEN THIS MESSAGE IS DONE AND THE SWR IS NOT EQUAL TO 377; THE CONSOLE WILL PRINT:

''TYPE A CHAR. ON DZ11 TERMINAL''

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

''PASS DONE.''

ON THE CONSOLE TERMINAL AND THE ''QUICK BROWN FOX'' WILL BE PRINTED ON DZ11 TERMINAL AGAIN AND THE ECHO TEST WILL BE RUNNING. TO CHANGE LINES: TYPE ANY PRINTABLE CHARACTER ON THE CONSOLE TERMINAL (NOT THE DZ11 TERMINAL). THE PROGRAM WILL AGAIN TYPE ''LINE: '' AND WAIT FOR A RESPONSE.

```
506
507
508                         5.4     PROGRAM AND/OR OPERATOR ACTION
509
510                                 THE  VARIETY  OF  PROGRAM  CONTROL SWITCHES PROVIDED IN THIS  DIAGNOSTIC
511                                 PACKAGE  IS DESIGNED TO  PROVIDE THE USER  WITH A WIDE RANGE OF TROUBLE-
512                                 SHOOTING TECHNIQUES.  BEFORE THE USER ATTEMPTS TO RUN THIS DIAGNOSTIC HE
513                                 SHOULD BECOME  FAMILIAR WITH THE USE OF THESE CONTROL SWITCHES AND THEIR
514                                 RESTRICTIONS.  (SEE SEC. 4.1, 4.1.1, 4.1.2, 4.1.3)
515
516                                 WHEN THE PROGRAM DETECTS AN ERROR THE TEST NUMBER AND PC WILL BE   TYPED
517                                 OUT AND POSSIBLY  AN ERROR MESSAGE  (DEPENDING ON THE PARTICULAR ERROR).
518                                 IF IT IS NECESSARY TO KNOW MORE INFORMATION  CONCERNING THE ERROR REPORT
519                                 THEN LOOK IN THE  PROGRAM LISTING FOR THAT TEST NUMBER AND THEN NOTE THE
520                                 PC OF THE  ERROR  REPORT.  THE  REASON FOR THE  ERROR REPORT WILL BECOME
521                                 CLEARER WHEN READING THE COMMENTS IN THE PROGRAM LISTING.
522
523                         6.      ERRORS
524
525                                 AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC  TYPED
526                                 OUT  AT  THE  TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 12=0).  IN MOST
527                                 CASES ADDITIONAL INFORMATION WILL BE SUPPLIED TO THE THE  ERROR  MESSAGE
528                                 WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.
529
530                         6.2     ERROR RECOVERY
531
532                                 IF FOR SOME REASON THE DZ11 SHOULD 'HANG THE BUS' (GAIN CONTROL  OF  BUS
533                                 SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP
534                                 IS NECESSARY FOR OPERATOR TO REGAIN CONTROL  OF  CPU.   IF  THIS  SHOULD
535                                 HAPPEN;  LOOK  IN  LOCATION 'TSTNO' (ADDRESS 1216)FOR THE NUMBER OF THE
536                                 TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC  ERROR.   IN  THIS
537                                 WAY  THE OPERATOR WILL HAVE AN IDEA AS TO WHAT THE DZ11 WAS DOING AT THE
538                                 TIME OF THE ERROR.
539
540                         7.      RESTRICTIONS
541
542                         7.1     STARTING RESTRICTIONS
543
544                                 SEE SECTION 4.1.2
545                                 STATUS TABLE SHOULD BE VERIFIED REGARDLESS OF HOW PROGRAM  WAS  STARTED.
546                                 ALSO  IT  IS  IMPORTANT  TO  USE THIS LISTING ALONG WITH THE INFORMATION
547                                 PRINTED ON THE TTY TO COMPLETLY ISOLATE PROBLEMS.
548
```

549
550
551 7.2 OPERATING RESTRICTIONS
552
553 PARAMETER MUST BE INPUT FROM USER OR APT IF ''AUTO SIZING'' IS NOT USED.
554
555
556 8. MISCELLANEOUS
557
558 8.1 EXECUTION TIME
559
560 ALL DZ11 DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS' MESSAGE (PROVIDING
561 NO ERRORS AND SW12=0) WITHIN 2 MIN. THIS IS ASSUMING SW11=1 (INHIBIT
562 ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION. THE ACTUAL
563 EXECUTION TIME DEPENDS GREATLY ON THE PDP11 CPU CONFIGURATION. AN 11/40
564 WITH CORE MEMORY WILL TAKE AROUND 100 SECONDS TO EXECUTE A PASS WITH NO
565 ITERATIONS AND ABOUT 400 SECONDS TO EXECUTE A FULLY ITERATED PASS. ANY
566 OTHER PDP11 CPU TYPE WILL EXECUTE A PASS IN TIME PROPORTIONAL TO THE
567 EXECUTION SPEED OF THE CPU 'S MEMORY IN RELATION TO THAT OF AN 11/40.
568
569 8.2 PASS COMPLETE
570
571 NOTE: *EVERY* TIME THE PROGRAM IS STARTED; THE TESTS WILL RUN AS IF
572 SW11 (DELETE ITERATIONS) WAS UP (=1). THIS IS TO 'VERIFY NO *HARD*
573 ERRORS' AS SOON AS POSSIBLE. THEREFORE THE FIRST PASS -EACH TIME
574 PROGRAM IS STARTED- WILL BE A 'QUICK PASS' UNTIL ALL DZ11'S IN SYSTEM
575 ARE TESTED. WHEN THE DIAGNOSTIC HAS COMPLETED A PASS THE FOLLOWING IS
576 AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.
577
578 END PASS CZDZA-F CSR: 160010 VEC: 300 PASSES: 000001 ERRORS:
579
580 NOTE: THE NUMBERS FOR CSR AND VEC ARE NOT NECESSARILY THE VALUES FOR
581 THE DEVICE. THEY ARE ONLY FOR THIS EXAMPLE.

```
582
583
584                          8.4     KEY LOCATIONS
585
586                                  $LPADR (1126)   CONTAINS THE ADDRESS  WHERE  PROGRAM  WILL  RETURN  WHEN
587                                                  ITERATION  COUNT  IS  REACHED  OR  IF  LOOP  ON  TEST  IS
588                                                  ASSERTED.
589                                  NEXT   (1360)   CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PEFORMED.
590                                  $TSTNM (1122)   CONTAINS THE NUMBER OF THE TEST NOW BEING PEFORMED.
591                                  RUN    (1406)   THE BIT  IN  'RUN'  ALWAYS  POINTS  ONE  PAST  THE  DZ11
592                                                  CURRENTLY     BEING     TESTED.     EXAMPLE:     (RUN)
593                                                  1304/0000000001000000 MEANS THAT DZ11 NO.05 IS THE  DZ11
594                                                  NOW RUNNING.
595                                  STATUS MAP
596                                  (1500)-(2000)
597                                                  THESE LOCATIONS CONTAIN THE INFORMATION NEEDED  TO  TEST
598                                                  UP  TO 16 (DECIMAL) DZ11S SEQUENTIALY.  THEY CONTAIN THE
599                                                  CSR,VECTOR AND STATUS CONCERNING  THE  CONFIGURATION  OF
600                                                  EACH DZ11.
601                                  DZACTV (1404)   EACH  BIT  SET  IN  THIS  LOCATION  INDICATES  THAT  THE
602                                                  ASSOCIATED  DZ11  WILL  BE  TESTED  IN  TURN.   EXAMPLE:
603                                                  (DZACTV)  1300/0000000000011111  MEANS  THAT  DZ11   NO.
604                                                  00,01,02,03,04   WILL  BE  TESTED.   EXAMPLE:   (DZACTV)
605                                                  1300/0000000000010001 MEANS THAT DZ11 NO.  00,04 WILL BE
606                                                  TESTED.
607                                  $BASE  (1310)   CONTAINS THE RECEIVER CSR  OF  THE  CURRENT  DZ11  UNDER
608                                                  TEST.
609
```

610
611
612 8.4A MORE ON THAT 'STATUS TABLE' (1500-2000)
613
614 'MAP OF DZ11 STATUS'
615 1500 160100
616 1502 000300
617 1504 000005
618 1506 000377
619 1510 017470
620 1512 000000
621
622
623 THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 16 DZ11'S IN
624 THE SYSTEM(THESE WILL FOLLOW UNDER THIS TABLE). EXPLANATION:
625 1500 160100 THIS IS THE SYSTEM CONTROL REGISTER FOR THE 1ST DZ11 IN
626 THE SYSTEM.
627 1502 000300 THIS IS VECTOR 'A' FOR THE FIRST DZ11 IN THE SYSTEM.
628 1504 000005 THIS REPRESENTS THE BUS INTERRUPT PRIORITY LEVEL OF THE
629 DZ11. BIT15 OF THIS LOCATION INDICATES EITHER EIA OR
630 20MA. IF BIT15=0 MODULE SHOULD BE AN M7819, IF BIT15=1
631 MODULE SHOULD BE AN M7814.
632 1506 000377 THIS IS THE BINARY REPRESENTATION OF WHAT LINES ARE TO
633 BE TESTED.
634 1510 017470 THIS IS THE PARAMETER LOCATION USED IN MOST OF THE
635 TESTS. IT INDICATES PARAMETERS OF: RX ON, SPEED SELECT
636 17 (19.2K BAUD) EIGHT BITS PER CHAR, AND TWO STOP BITS.
637 THE USER MAY ALTER THE STOP BITS AND THE SPEED, BUT THE
638 REMAINING PARAMETERS SHOULD BE LEFT ALONE.
639 THIS LOCATION IS USED TO LOAD THE DZ11 LINE PARAMETER
640 REGISTER FOR EACH LINE. THE MEANING OF THE BITS SET IN
641 THIS LOCATION IS THE SAME AS THE FUNCTION OF THE RELATED
642 BITS IN THE DEVICE LINE PARAMETER REGISTER.
643 1512 000000 THIS LOCATION WILL CONTAIN EITHER ALL ZEROS INDICATING
644 THAT INTERNAL LOOP WAS SELECTED AS MODE OF OPERATION OR
645 IT WILL CONTAIN 10000 INDICATING THAT "STAGGERED MODE"
646 WAS SELECTED OR IT WILL CONTAIN 000200 INDICATING THAT
647 "EXTERNAL" WAS THE MODE SELECTED.
648
649 THE ABOVE IS REPEATED FOR EACH DZ11 IN THE SYSTEM. THE TABLE IS
650 FILLED BY AUTO SIZING OR BY THE MANUAL PARAMETER INPUT PROGRAM
651 AS DESCRIBED PREVIOUSLY. ALSO IF DESIRED BY USER; THE
652 LOCATIONS MAY BE ALTERED BY HAND (TOGGLED IN) TO SUIT THE
653 SPECIFIC CONFIGURATION.
654

655
656
657 8.5 *** METHOD OF AUTO SIZING ***
658
659 8.5.1 FINDING THE CONTROL STATUS REGISTER.
660
661 THE PROGRAM WILL START AT ADDRESS 160000 AND START 'REFERENCING' THE
662 ADDRESS IN THE POINTER. IF A NON-EX MEMORY TRAP OCCURES, THE POINTER
663 (HOLDING 160000) IS UPDATED BY 10 AND THE ABOVE IS REPEATED UNTIL
664 ADDRESS 163700 IS REACHED. IF A 'SLAVE SYNC RESPONSE' WAS ISSUED BY THE
665 DZ11 (OR ANY OTHER DEVICE) (NO NXM TRAP), 'MASTER SCAN ENABLE'' IS
666 ATTEMPTED TO BE SET AND THE 'TCR'' BIT FOR LINE 7 IS SET. 'TRDY'' IS THEN
667 TESTED TO BE SET AND BOTH 'TCR07'' AND 'MASTER SCAN ENABLE'' ARE TESTED TO
668 BE STILL SET. IF ALL OF THIS WORKED; THEN A 'DEVICE CLEAR'' IS ISSUED
669 TESTING THAT THE BIT CAN BE READ BACK AND THAT AFTER SOME TIME IT SELF
670 CLEARS. IF ALL OF THE ABOVE WORKED; THIS DEVICE IS ASSUMED TO BE A
671 DZ11. IF ANY OF THE ABOVE FAILED; UPDATING OF THE POINTER IS DONE AND
672 THE SEQUENCE IS REPEATED.
673 NOTE: IF THE PROGRAM DOES NOT FIND YOUR DZ11; SOMETHING IS WRONG AND
674 AUTO SIZING SHOULD NOT BE DONE.
675 AFTER IDENTIFYING A DZ11 THE PROGRAM THEN ATTEMPTS TO SET ALL DTR BITS
676 IN DEVICE REGISTER 4. IF ANY DTR BITS DID SET THE MODULE IS ASSUMED TO
677 BE AN EIA MODULE (M7819) OTHERWISE THE STATUS MAP ENTRY IS SET FOR 20MA
678 (M7814).
679
680 8.5.2 FINDING THE VECTOR
681
682 THE VECTOR AREA (ADDRESS 300-776) IS FILLED WITH THE INSTRUCTION IOT AND
683 '.+2' (NEXT ADDRESS). BIT14 AND BIT5 (TX INTERUPT ENABLE AND MSTSCAN
684 ENABLE) ARE SET INTO THE DZCSR. 'TCR07'' IS THEN SET. A DELAY IS MADE
685 AND IF NO INTERUPT OCCURES (BECAUSE OF A BAD DZ11) THE PROGRAM ASSUMES
686 VECTOR ADDRESS 300 AND THE PROBLEM SHOULD BE FIXED IN THE DIAGNOSTIC.
687 ONCE THE PROBLEM IS FIXED; THE PROGRAM SHOULD BE RE-SETUP AGAIN TO GET
688 CORRECT VECTOR. IF AN INTERUPT OCCURRED; THE ADDRESS TO WHICH THE DZ11
689 INTERUPTED TO IS PICKED UP AND REPORTED AS THE VECTOR. NOTE: IF THE
690 VECTOR REPORTED IS NOT THE VECTOR SET UP BY YOU; THERE IS A PROBLEM AND
691 AUTO SIZING SHOULD NOT BE DONE.
692
693 8.5.3 PARAMETER ASSUMPTIONS.
694
695 SINCE TOO MUCH HARDWARE WOULD NEED TO BE TURNED ON TO SIZE THE REST OF
696 THE PARAMETERS; THE PROGRAM MUST ASSUME THE REMAINING VARIATIONS. THE
697 RESULT IF NOT TO YOUR SPECIFIC CONFIGURATION MAY BE ALTERED BY HAND
698 (TOGGLE IN) IF DESIRED). IN THIS WAY 95 PERCENT OF THE PARAMETER SETUP
699 WAS DONE BY THE PROGRAM, AND 5 PERCENT BY YOU.
700 THEREFORE:
701 1) BUS PRIORITY IS SET TO LEVEL5.
702 2) ALL EIGHT LINES ARE ASSUMED TO BE TESTED.
703 3) DEFAULT BAUD RATE IS SET TO 17 (19.2 K).
704 4) MODE OF OPERATION IS ''INTERNAL MODE''.
705
706 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

```
   763                                  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.


                                %


                                .TITLE  CZDZA-F0
                                ;*COPYRIGHT (C) 1976,1979
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
000001                          $TN=1
                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;LOAD ADDRESS 000200
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE ''CZDZA-F0/<200>/CZDZAF0 DZ11 LN ASYNC MUX TSTS ''
                                        ;PROGRAM WILL TYPE 'RUNNING'' TO INDICATE THAT TESTING HAS STARTED
                                        ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                                        ;AND THEN RESUME TESTING

                                .SBTTL  BASIC DEFINITIONS

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

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

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

                        ;*''SW'TCH REGISTER'' SWITCH DEFINITIONS
                        SW15=   100000
                        SW14=   40000
                        SW13=   20000
                        SW12=   10000
                        SW11=   4000
                        SW10=   2000
                        SW09=   1000
                        SW08=   400
                        SW07=   200
                        SW06=   100
                        SW05=   40
                        SW04=   20
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        BIT15=  100000
                        BIT14=  40000
                        BIT13=  20000
                        BIT12=  10000
                        BIT11=  4000
                        BIT10=  2000
```

```
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
022626                  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

                                ;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
                000002          LP2=BIT1        ;LINE PARAMETER 2
                000003          LP3=BIT1!BIT0   ;LINE PARAMETER 3
                000004          LP4=BIT2        ;LINE PARAMETER 4
                000005          LP5=BIT2!BIT0   ;LINE PARAMETER 5
                000006          LP6=BIT2!BIT1   ;LINE PARAMETER 6
                000007          LP7=BIT2!BIT1!BIT0      ;LINE PARAMETER 7

                000000          FIVE=0          ;FIVE BITS/CHAR,1 STOP BIT
                000010          SIX=BIT3        ;SIX BITS/CHAR,1 STOP BIT
                000020          SEVEN=BIT4      ;SEVEN BITS/CHAR,1 STOP BIT
                000030          EIGHT=BIT4!BIT3 ;EIGHT BITS/CHAR,1 STOP BIT
                000040          FIVES=BIT5      ;FIVE BITS/CHAR,2 STOP BITS
                000050          SIXS=BIT5!BIT3  ;SIX BITS/CHAR,2 STOP BITS
                000060          SEVENS=BIT5!BIT4        ;SEVEN BITS/CHAR, 2 STOP BITS
                000070          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
                        DTR6=BIT14              ;DTR6
                        DTR7=BIT15              ;DTR7

                                ;DZMSR BIT DEFINITIONS
                                ;---------------------

                        RING0=BIT0              ;RING INDICATED ON LINE 0
                        RING1=BIT1              ;RING INDICATED ON LINE 1
                        RING2=BIT2              ;RING INDICATED ON LINE 2
                        RING3=BIT3              ;RING INDICATED ON LINE 3
                        RING4=BIT4              ;RING INDICATED ON LINE 4
                        RING5=BIT5              ;RING INDICATED ON LINE 5
                        RING6=BIT6              ;RING INDICATED ON LINE 6
                        RING7=BIT7              ;RING INDICATED ON LINE 7
                        CO0=BIT8                ;CARRIER PRESENT ON LINE 0
                        CO1=BIT9                ;CARRIER PRESENT ON LINE 1
```

```
                        CO2=BIT10               ;CARRIER PRESENT ON LINE 2
                        CO3=BIT11               ;CARRIER PRESENT ON LINE 3
                        CO4=BIT12               ;CARRIER PRESENT ON LINE 4
                        CO5=BIT13               ;CARRIER PRESENT ON LINE 5
                        CO6=BIT14               ;CARRIER PRESENT ON LINE 6
                        CO7=BIT15               ;CARRIER PRESENT ON LINE 7

                                ;DZTDR BIT DEFINITIONS
                                ;---------------------

                        BRK0=BIT8               ;BREAK FOR LINE 0
                        BRK1=BIT9               ;BREAK FOR LINE 1
                        BRK2=BIT10              ;BREAK FOR LINE 2
                        BRK3=BIT11              ;BREAK FOR LINE 3
                        BRK4=BIT12              ;BREAK FOR LINE 4
                        BRK5=BIT13              ;BREAK FOR LINE 5
                        BRK6=BIT14              ;BREAK FOR LINE 6
                        BRK7=BIT15              ;BREAK FOR LINE 7


                        ;TABLE OF LOOP AROUND FUNCTIONS (H325)
                        ;
                        ;       ----------------
                        ;       I               ^
                        ;       V               ^
                        ;       REC             TRANS
                        ;       DATA            DATA
                        ;
                        ;       ----------------
                        ;       I               ^
                        ;       V               ^
                        ;       CO              RTS
                        ;
                        ;       ----------------
                        ;       I               ^
                        ;       V               ^
                        ;       RING            DTR
```

```
                        ;:*****************************************************************************
                        ;-----------------------------------------------------------------------------
                                ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
                                ;THE STANDARD ''TRAP CATCHER'' IS PLACED
                                ;BETWEEN ADDRESS 0 TO ADDRESS 776.
                                ;IT LOOKS LIKE ''PC+2 HALT''.
                        ;-----------------------------------------------------------------------------
                        ;:*****************************************************************************

                000000  .=0
                                ;STANDARD INTERRUPT VECTORS
                                ;-------------------------

                000010  .=10
000010  010766                  SET.PS                  ;FAKE ''MTPS'' INSTRUCTION TRAP
000012  000340                  PR7                     ;MAKE SURE PS IS PRIORITY 7

                000020  .=20
000020  004772                  .SCOPE                  ;SCOPE LOOP HANDLER
000022  000340                  PR7                     ;HANDLE AT PRIORITY 7
000024  007646                  $PWRDN                  ;POWER FAIL HANDLER
000026  000340                  340                     ;SERVICE AT PRIORITY LEVEL 7
000030  006736                  $ERROR                  ;ERROR HANDLER
000032  000340                  340                     ;SERVICE AT PRIORITY LEVEL 7
000034  006630                  .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                  340                     ;SERVICE AT PRIORITY LEVEL 7
                        .SBTTL  ACT11 HOOKS

                        ;:*****************************************************************************
                        ;HOOKS REQUIRED BY ACT11
                000040          $SVPC=.                 ;SAVE PC
                000046          .=46
000046  004726                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052          .=52
000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
                000040          .=$SVPC                 ;; RESTORE PC

                000174  .=174
000174  000000  DISPREG:0                               ;SOFTWARE DISPLAY REGISTER FOR SWITCHLESS 11S
000176  000000  SWREG:  0                               ;SOFTARE SWITCH REGISTER FOR SWITCHLESS 11S
                000200  .=200
000200  000137  002150          JMP     .START          ;GO TO START OF PROGRAM
                000210  .=210
000210  000137  023770          JMP     XSTART          ;GOTO CABLE TEST/ECHO TEST


                001000  .=1000
001000  005200  055103  055104  MTITLE: .ASCIZ  <200><12>/CZDZA-F0/<200>/CZDZAF0 DZ11 LN ASYNC MUX TSTS /<200>
  (2)
```

```
1130                                    .SBTTL  COMMON TAGS
1131
1132                                    ;;*****************************************************************
1133                                    ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
1134                                    ;*USED IN THE PROGRAM.
1135
1136    001120                                  .=1120
1137    001120                          $CMTAG:                         ;;START OF COMMON TAGS
1138    001120  000000                          .WORD   0
1139    001122  000                     $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
1140    001123  000                     $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
1141    001124  000000                  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
1142    001126  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
1143    001130  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
1144    001132  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
1145    001134  000                     $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
1146    001135  001                     $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
1147    001136  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
1148    001140  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
1149    001142  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
1150    001144  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
1151    001146  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
1152    001150  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
1153    001152  000000                          .WORD   0
1154    001154  000                     $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
1155    001155  000                     $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
1156    001156  000000                          .WORD   0
1157    001160  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
1158    001162  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
1159    001164  177560                  $TKS:   177560                  ;;TTY KBD STATUS
1160    001166  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
1161    001170  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
1162    001172  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
1163    001174  000                     $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
1164    001175  002                     $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
1165    001176  012                     $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
1166    001177  000                     $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
1167    001200  000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
1168                                                                    ;;WHICH  ($REG0) WAS OBTAINED
1169    001202  000000                  $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
1170    001204  000000                  $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
1171    001206  000000                  $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
1172    001210  000000                  $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
1173    001212  000000                  $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
1174    001214  000000                  $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
1175    001216  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
1176    001220  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
1177    001222  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
1178    001224  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
1179    001226  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
1180    001230  077                     $QUES:  .ASCII  /?/             ;;QUESTION MARK
1181    001231  015                     $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
1182    001232  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
1183                                    ;;*****************************************************************
1184                                    .SBTTL  APT MAILBOX-ETABLE
1185
```

```
1186                                    ;;*****************************************************************
1187                                    .EVEN
1188  001234                            $MAIL:                          ;;APT MAILBOX
1189  001234  000000                    $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
1190  001236  000000                    $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
1191  001240  000000                    $TESTN: .WORD   ATESTN          ;;TEST NUMBER
1192  001242  000000                    $PASS:  .WORD   APASS           ;;PASS COUNT
1193  001244  000000                    $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
1194  001246  000000                    $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
1195  001250  000000                    $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
1196  001252  000000                    $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
1197  001254                            $ETABLE:                        ;;APT ENVIRONMENT TABLE
1198  001254     000                    $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
1199  001255     000                    $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
1200  001256  000000                    $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
1201  001260  000000                    $USWR:  .WORD   AUSWR           ;;USER SWITCHES
1202  001262  000000                    $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
1203                                    ;*                              BITS 15-11=CPU TYPE
1204                                    ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1205                                    ;*                                      11/70=06,PDQ=07,Q=10
1206                                    ;*                              BIT 10=REAL TIME CLOCK
1207                                    ;*                              BIT  9=FLOATING POINT PROCESSOR
1208                                    ;*                              BIT  8=MEMORY MANAGEMENT
1209  001264     000                    $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
1210  001265     000                    $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
1211                                    ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
1212                                    ;*                                      900 NSEC CORE=001
1213                                    ;*                                      300 NSEC BIPOLAR=002
1214                                    ;*                                      500 NSEC MOS=003
1215  001266  000000                    $MADR1: .WORD   AMADR1   ;;HIGH ADDRESS,BLK#1
1216                                    ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
1217  001270     000                    $MAMS2: .BYTE   AMAMS2   ;;HIGH ADDRESS,M.S. BYTE
1218  001271     000                    $MTYP2: .BYTE   AMTYP2   ;;MEM.TYPE,BLK#2
1219  001272  000000                    $MADR2: .WORD   AMADR2   ;;MEM.LAST ADDRESS,BLK#2
1220  001274     000                    $MAMS3: .BYTE   AMAMS3   ;;HIGH ADDRESS,M.S.BYTE
1221  001275     000                    $MTYP3: .BYTE   AMTYP3   ;;MEM.TYPE,BLK#3
1222  001276  000000                    $MADR3: .WORD   AMADR3   ;;MEM.LAST ADDRESS,BLK#3
1223  001300     000                    $MAMS4: .BYTE   AMAMS4   ;;HIGH ADDRESS,M.S.BYTE
1224  001301     000                    $MTYP4: .BYTE   AMTYP4   ;;MEM.TYPE,BLK#4
1225  001302  000000                    $MADR4: .WORD   AMADR4   ;;MEM.LAST ADDRESS,BLK#4
1226  001304  000000                    $VECT1: .WORD   AVECT1   ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1227  001306  000000                    $VECT2: .WORD   AVECT2   ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1228  001310  160010                    $BASE:  .WORD   ABASE    ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1229  001312  000000                    $DEVM:  .WORD   ADEVM    ;;DEVICE MAP
1230  001314  000000                    $CDW1:  .WORD   ACDW1    ;;CONTROLLER DESCRIPTION WORD#1
1231  001316  000000                    $CDW2:  .WORD   ACDW2    ;;CONTROLLER DESCRIPTION WORD#2
1232  001320  000000                    $DDW0:  .WORD   ADDW0    ;;DEVICE DESCRIPTOR WORD#0
1233  001322  000000                    $DDW1:  .WORD   ADDW1    ;;DEVICE DESCRIPTOR WORD#1
1234  001324  000000                    $DDW2:  .WORD   ADDW2    ;;DEVICE DESCRIPTOR WORD#2
1235  001326  000000                    $DDW3:  .WORD   ADDW3    ;;DEVICE DESCRIPTOR WORD#3
1236  001330  000000                    $DDW4:  .WORD   ADDW4    ;;DEVICE DESCRIPTOR WORD#4
1237  001332  000000                    $DDW5:  .WORD   ADDW5    ;;DEVICE DESCRIPTOR WORD#5
1238  001334  000000                    $DDW6:  .WORD   ADDW6    ;;DEVICE DESCRIPTOR WORD#6
1239  001336  000000                    $DDW7:  .WORD   ADDW7    ;;DEVICE DESCRIPTOR WORD#7
1240  001340  000000                    $DDW8:  .WORD   ADDW8    ;;DEVICE DESCRIPTOR WORD#8
1241  001342  000000                    $DDW9:  .WORD   ADDW9    ;;DEVICE DESCRIPTOR WORD#9
```

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

```
                        .SBTTL  ERROR POINTER TABLE

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

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

001360                  $ERRTB:

                                ;PROGRAM CONTROL PARAMETERS
                                ;--------------------------

001360  000000          NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001362  000000          LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA

                                ;PROGRAM VARIABLES
                                ;-----------------

001364  000377          LINE:   377                     ;DEFAULT ALL EIGHT LINES RUNNING
001366  017470          PAR:    17470                   ;PARAMETERS: 8 BITS/CHAR,2 STOP BITS,19200 BAUD,NO PARIT
001370  000000          MODE:   0                       ;DEFAULT MAINTENANCE MODE
001372  000000          SAVLIN: 0                       ;LINE NUMBER
001374  000000          XMTLIN: 0                       ;TRANSMISSION LINE NUMBER
001376  000000          XMTCNT: 0                       ;COUNT OF WORDS IN A TRANSMISSION PATTERN
001400  000000          REGIST: 0                       ;DEVICE ADDRESS STORAGE LOCATION
001402  000000          SAVPC:  0                       ;PROGRAM COUNTER STORAGE
001404  000001          DZACTV: .BLKW   1               ;*DZ11'S SELECTED ACTIVE.
001406  000001          RUN:    1                       ;*POINTER ONE PAST RUNNING DEVICE.
001410  000001          DZNUM:  .BLKB 1                 ;*OCTAL NUMBER OF DZ11'S.
001411     001          SAVNUM: .BYTE 1                 ;*WORKABLE NUMBER.
                        .EVEN
001412  001500          ACTIVE: DZ.MAP                  ;TABLE POINTER.
```

```
1291
1292                                    ;PROGRAM CONTROL FLAGS
1293                                    ;---------------------
1294
1295  001414      000           EIAFLG: .BYTE   0               ;0=EIA  100000=20MA
1296  001415      000           INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
1297  001416      000           HDRFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG FOR HEADER MAP
1298  001417      000           MNTFLG: .BYTE   0               ;MAINTENANCE BIT SET FLAG
1299  001420      000           DONFLG: .BYTE   0               ;TRANSMISSION COMPLETION FLAG
1300              001422          .EVEN
1301                            ;DATA VARIABLES
1302  001422  000000            TD0:    .WORD   0
1303  001424  000000            TD1:    .WORD   0
1304  001426  000000            TD2:    .WORD   0
1305  001430  000000            TD3:    .WORD   0
1306  001432  000000            TD4:    .WORD   0
1307  001434  000000            TD5:    .WORD   0
1308  001436  000000            TD6:    .WORD   0
1309  001440  000000            TD7:    .WORD   0
1310  001442  000000            TR0:    .WORD   0
1311  001444  000000            TR1:    .WORD   0
1312  001446  000000            TR2:    .WORD   0
1313  001450  000000            TR3:    .WORD   0
1314  001452  000000            TR4:    .WORD   0
1315  001454  000000            TR5:    .WORD   0
1316  001456  000000            TR6:    .WORD   0
1317  001460  000000            TR7:    .WORD   0
1318  001462                    STOP:
1319                            .SBTTL  APT PARAMETER BLOCK
1320
1321                            ;;**********************************************************************
1322                            ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1323                            ;;**********************************************************************
1324              001462                .$X=.   ;;SAVE CURRENT LOCATION
1325              000024                .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1326  000024  000200                    200     ;;FOR APT START UP
1327              000044                .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1328  000044  001462                    $APTHDR ;;POINT TO APT HEADER BLOCK
1329              001462                .=.$X   ;;RESET LOCATION COUNTER
1330                            ;;**********************************************************************
1331                            ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1332                            ;INTERFACE SPEC.
1333
1334  001462                    $APTHD:
1335  001462  000000            $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1336  001464  001234            $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1337  001466  000132            $TSTM:  .WORD   90.     ;;RUN TIM OF LONGEST TEST
1338  001470  000137            $PASTM: .WORD   95.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1339  001472  000137            $UNITM: .WORD   95.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1340  001474  000052                    .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1341                                    ;DZ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
1342                                    ;-----------------------------------------
1343
1344              001500        .=1500
1345  001500                    DZ.MAP:
1346
```

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

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

```
  1459  002000  177777                    DZ.END: 177777
```

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

                                ;:**************************************************************************
                                ;--------------------------------------------------------------------------
002002                          .TRPTAB:
        104400                  ADVANCE=TRAP+0          ;CALL TO ADVANCE TO NEXT TEST( OR SCOPE THIS ONE)
002002  006724                          .ADVANCE
        104401                  SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
002004  005236                          .SCOP1
        104402                  TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
002006  005262                          .TYPE
        104403                  INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
002010  006030                          .INSTR
        104404                  INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
002012  006134                          .INSTER
        104405                  PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
002014  006154                          .PARAM
        104406                  SETFLG=TRAP+6           ;CALL TO  SET FLAG ROUTINE
002016  010500                          .SETFLG
        104407                  SAV05=TRAP+7            ;CALL TO REGISTER SAVE ROUTINE
002020  006354                          .SAV05
        104410                  RES05=TRAP+10           ;CALL TO REGISTER RESTORE ROUTINE
002022  006414                          .RES05
        104411                  CONVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUTINE
002024  006446                          .CONVRT
        104412                  CNVRT=TRAP+12           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
002026  006452                          .CNVRT
        104413                  DEVICE.CLR=TRAP+13              ;CALL TO ISSUE A DEVICE CLEAR
002030  006652                          .DEVICE.CLR
        104414                  DELAY=TRAP+14           ;CALL TO DELAY FOR FAST CPU'S
002032  006704                          .DELAY
        104415                  PARMD=TRAP+15           ;CONVERT DECIMAL STRING TO OCTAL
002034  025516                          .PARMD
        104416                  PAWCH=TRAP+16           ;SET FLAG    ECHO OR  CABLE
002036  025712                          .PAWCH
        104417                  DCLASM=TRAP+17          ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
002040  006672                          .DCLASM

                                ;--------------------------------------------------------------------------
                                ;:**************************************************************************
```

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

002042  160040                DZCSR:  160040  ;R/W
002044  160041                HDZCSR: 160041  ;R/W
002046  160042                DZRBUF: 160042  ;READ ONLY
002050  160043                HDZRBUF:        160043  ;READ ONLY
002052  160042                DZLPR:  160042  ;WRITE ONLY
002054  160043                HDZLPR: 160043  ;WRITE ONLY
002056  160044                DZTCR:  160044  ;R/W
002060  160045                HDZTCR: 160045  ;R/W
002062  160046                DZMSR:  160046  ;READ ONLY
002064  160047                HDZMSR: 160047  ;READ ONLY
002066  160046                DZTDR:  160046  ;WRITE ONLY
002070  160047                HDZTDR: 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
```

```
1524
1525                                    ;TIME TABLE FOR RELATIVE TIMING TESTS
1526                                    ;-----------------------------------
1527
1528  002102                            TMTBL:
1529  002102  000000                    T50:    0
1530  002104  000000                    T75:    0
1531  002106  000000                    T110:   0
1532  002110  000000                    T134:   0
1533  002112  000000                    T150:   0
1534  002114  000000                    T300:   0
1535  002116  000000                    T600:   0
1536  002120  000000                    T1200:  0
1537  002122  000000                    T1800:  0
1538  002124  000000                    T2000:  0
1539  002126  000000                    T2400:  0
1540  002130  000000                    T3600:  0
1541  002132  000000                    T4800:  0
1542  002134  000000                    T7200:  0
1543  002136  000000                    T9600:  0
1544  002140  000000                    TEIGHT:0
1545  002142  000000                    TSEVEN: 0
1546  002144  000000                    TSIX:   0
1547  002146  000000                    TFIVE:  0
```

```
1548
1549                                        ;PROGRAM INITIALIZATION
1550                                        ;LOCK OUT INTERRUPTS
1551                                        ;SET UP PROCESSOR STACK
1552                                        ;SET UP POWER FAIL VECTOR
1553                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1554                                        ;TYPE TITLE MESSAGE
1555
1556  002150                          .START:
1557  002150  000005                       RESET                   ;CLEAR THE WORLD. START NEW ENVIRONMENT
1558  002152  012706  001120               MOV     #STACK,SP       ;SET UP STACK
1559  002156  106427  000340               MTPS    #PR7            ;LOCK OUT INTERRUPTS
1560  002162  012737  007646  000024       MOV     #$PWRDN,@#24    ;SET UP POWER FAIL VECTOR
1561  002170  113737  001410  001411       MOVB    DZNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
1562  002176  005037  001242               CLR     $PASS           ;CLEAR PASS COUNT
1563  002202  105037  001123               CLRB    $ERFLG          ;CLEAR ERROR FLAG
1564  002206  012737  001500  001412       MOV     #DZ.MAP,ACTIVE  ;GET MAP POINTER.
1565  002214  012737  000001  001406       MOV     #1,RUN          ;POINT POINTER TO FIRST DEVICE.
1566  002222  005037  001132               CLR     $ERTTL          ;CLEAR ERROR COUNT
1567  002226  005037  001136               CLR     $ERRPC          ;CLEAR LAST ERROR POINTER
1568  002232  005037  001122               CLR     $TSTNM          ;SET UP FOR TEST 1
1569  002236  012737  002150  001126       MOV     #.START,$LPADR  ;SET UP FOR POWER FAIL BEFORE
1570                                                                ;TESTING STARTS
1571                                        ;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY
1572  002244  013746  000006               MOV     6,-(SP)         ;SAVE BUS ERROR PS
1573  002250  013746  000004               MOV     4,-(SP)         ;SAVE BUS ERROR PC
1574  002254  012737  002274  000004       MOV     #20$,4          ;SET UP TO TRAP TO THIS ROUTINE
1575  002262  022777  177777  176670       CMP     #-1,@SWR        ;CAN 177570 BE REFERENCED?
1576  002270  001402                       BEQ     22$             ;IF SO AND IT IS -1, TREAT LIKE SWITCHLESS
1577  002272  000407                       BR      21$             ;IF YES,SKIP AROUND THE SETUP
1578  002274  022626                20$:   POP2SP                  ;REMOVE THE TRAP FROM THE STACK
1579  002276  012737  000176  001160  22$:  MOV     #SWREG,SWR      ;IF NO,TRAP COMES HERE.POINT TO SOFTWARE SWR
1580  002304  012737  000174  001162       MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REGISTER
1581  002312  012637  000004          21$: MOV     (SP)+,4         ;RESTORE THE BUS ERROR VECTOR
1582  002316  012637  000006               MOV     (SP)+,6
1583  002322  105737  001415               TSTB    INIFLG          ;TITLE ALREADY PRINTED?
1584  002326  001010                       BNE     29$             ;BRANCH IF YES
1585  002330  023727  000042  004726       CMP     @#42,#$ENDAD    ;RUNNING UNDER ACT?
1586  002336  001402                       BEQ     31$             ;IF YES DONT PRINT TITLE
1587  002340  104402  001000               TYPE    ,MTITLE         ;PRINT THE DIAGNOSTIC'S TITLE
1588  002344  105337  001415          31$: DECB    INIFLG          ;SET THE ONCE ONLY FLAG
1589  002350  105737  001255          29$: TSTB    $ENVM           ;DETERMINE WHETHER APT SIZING SHOULD BE DONE
1590  002354  100006                       BPL     30$             ;IF NOT, GO CHECK FOR AUTO-SIZING
1591  002356  004737  011440               JSR     PC,SETAPT       ;OTHERWISE, GO DO APT SIZING FROM ETABLE
1592  002362  105037  001416               CLRB    HDRFLG          ;MAKE SURE STATUS TABLE IS PRINTED
1593  002366  000137  004270               JMP     16$             ;GO PRINT DZ STATUS TABLE
1594  002372  032777  000001  176560  30$: BIT     #SW00,@SWR      ;RESELECT ?
1595  002400  001011                       BNE     32$             ;IF YES, GO SET UP THE INFORMATION
1596  002402  122737  000377  001415       CMPB    #377,INIFLG     ;ON 1ST START; MUST ANSWER QUESTION
1597  002410  001003                       BNE     .+10            ;IF NOT ANSWERING QUESTIONS
1598  002412  105777  176542               TSTB    @SWR            ;ARE U AUTO SIZING?
1599  002416  100402                       BMI     32$             ;NO AUTO SIZE! NO SW00=1 ON 1ST START!
1600  002420  000137  003114               JMP     73$             ;IF NO, SKIP THE INTERROGATION
1601  002424  012700  001500          32$: MOV     #DZ.MAP,R0      ;POINT TO THE BEGINNING OF THE MAP TABLE
1602  002430  105037  001416               CLRB    HDRFLG          ;MAKE SURE A MAP GETS PRINTED
1603  002434  005020                65$:   CLR     (R0)+           ;CLEAR A TABLE LOCATION
```

```
1604  002436  020027  002000          CMP     R0,#DZ.END      ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
1605  002442  001374                  BNE     65$             ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE
1606  002444  105337  001415          DECB    INIFLG          ;INSURE NO AUTO SIZING IF QUESTIONS ANSWERED!
1607
1608                                ;THE FOLLOWING ARE PARAMETERS USED TO FILL IN THE MAP
1609                                ;TABLE AND SET UP THE DIAGNOSTIC.
1610
1611                                        ;GET THE BASE ADDRESS OF THE DZ11'S
1612
1613  002450                        33$:
1614  002450  104403                  INSTR                   ;CALL THE STRING INPUT ROUTINE
1615  002452  003334                  66$                     ;POINTER TO MESSAGE TO BE PRINTED
1616  002454  104405                  PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1617  002456  160000                  160000                  ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1618  002460  163770                  163770                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1619  002462  001500                  DZCR0                   ;POINTER TO MAP LOCATION TO BE FILLED
1620  002464     007                  .BYTE   7               ;MASK OF INVALID BITS FOR THIS PARAMETER
1621  002465     001                  .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1622  002466  013737  001500  001310  MOV     DZCR0,$BASE     ;COPY BASE ADDRESS TO ETABLE
1623
1624                                        ;GET THE BASE VECTOR ADDRESS
1625
1626  002474                        34$:
1627  002474  104403                  INSTR                   ;CALL THE STRING INPUT ROUTINE
1628  002476  003400                  67$                     ;POINTER TO MESSAGE TO BE PRINTED
1629  002500  104405                  PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1630  002502  000300                  300                     ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1631  002504  000776                  776                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1632  002506  001502                  DZVC0                   ;POINTER TO MAP LOCATION TO BE FILLED
1633  002510     003                  .BYTE   3               ;MASK OF INVALID BITS FOR THIS PARAMETER
1634  002511     001                  .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1635  002512  013737  001502  001304  MOV     DZVC0,$VECT1    ;COPY VECTOR TO ETABLE
1636
1637                                        ;GET THE BUS REQUEST LEVEL
1638
1639  002520  104403                  INSTR                   ;CALL THE STRING INPUT ROUTINE
1640  002522  003441                  68$                     ;POINTER TO MESSAGE TO BE PRINTED
1641  002524  104405                  PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1642  002526  000004                  4                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1643  002530  000007                  7                       ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1644  002532  001504                  DZLV0                   ;POINTER TO MAP LOCATION TO BE FILLED
1645  002534     000                  .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1646  002535     001                  .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1647  002536  113737  001504  001305  MOVB    DZLV0,$VECT1+1  ;GET BUS REQUEST LEVEL INTO ETABLE
1648  002544  106337  001305          ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1649  002550  106337  001305          ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1650  002554  106337  001305          ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1651  002560  106337  001305          ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1652  002564  106337  001305          ASLB    $VECT1+1        ;ALIGN THE BITS PROPERLY
1653
1654                                ;FIND OUT IF MODULE IS EIA OR 20 MA.
1655
1656  002570  104402  004130          TYPE    ,74$            ;PRINT EIA MESSAGE
1657  002574  005037  001220          CLR     $TMP1           ;USE $TMP1
1658  002600  105777  176360  80$:    TSTB    @$TKS           ;IS KEYBOARD DONE?
1659  002604  100375                  BPL     80$             ;IF NOT, WAIT FOR IT
```

```
1660  002606  017746  176354                MOV     @$TKB,-(SP)     ;IF YES, PUT CHARACETR ON STACK
1661  002612  042716  000240                BIC     #240,(SP)       ;STRIP DOWN CHARACTER
1662  002616  122726  000015                CMPB    #15,(SP)+       ;IS IT ?
1663  002622  001414                       BEQ     81$             ;IF SO, GET OUT
1664  002624  014677  176342                MOV     -(SP),@$TPB     ;IF NOT, PRINT CHARACTER
1665  002630  042737  100000  001504        BIC     #BIT15,DZLV0    ;CLEAR EIA FLAG
1666  002636  122726  000102                CMPB    #102,(SP)+      ;IS IT A B?
1667  002642  001356                       BNE     80$             ;IF NOT, GO BACK FOR INPUT
1668  002644  052737  100000  001504        BIS     #BIT15,DZLV0    ;IF SO, SET FLAG
1669  002652  000752                       BR      80$             ;GET MORE INPUT
1670  002654                          81$:
1671
1672                                        ;GET THE MODE OF OPERATION (E,I,S)
1673
1674  002654  104403                       INSTR                   ;CALL THE STRING INPUT ROUTINE
1675  002656  003652                       72$                     ;POINTER TO THE MESSAGE TO BE PRINTED
1676  002660  104406                       SETFLG                  ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
1677  002662  001512                       MANT0                   ;THIS IS THE FLAG BEING SETUP
1678
1679                                        ;GET THE NUMBER OF DZ11'S RUNNING
1680
1681  002664  104403                       INSTR                   ;CALL THE STRING INPUT ROUTINE
1682  002666  003610                       71$                     ;POINTER TO MESSAGE TO BE PRINTED
1683  002670  104405                       PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1684  002672  000001                       1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1685  002674  000020                       16.                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1686  002676  001220                       $TMP1                   ;POINTER TO MAP LOCATION TO BE FILLED
1687  002700     000                       .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1688  002701     001                       .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1689
1690  002702  012737  000377  001506        MOV     #377,LINE0      ;SET UP DEFAULT LINES
1691  002710  012737  017470  001510        MOV     #17470,PAR0     ;SET UP DEFAULT LPR PARAMETER
1692                                        ;RECEIVER ON; 19.2 KBAUD; 2STOP BITS; 8 BIT/CHAR
1693  002716  012737  000001  006722        MOV     #1,DLYCNT       ;INITIALIZE DELAY COUNT
1694  002724  032777  000010  176226        BIT     #SW03,@SWR      ;DO YOU WANT PARAMETERS?
1695  002732  001402                       BEQ     40$             ;IF NO, SKIP THE PARAMETER CALL
1696  002734  004737  003144                JSR     PC,23$          ;GET PARAMETERS
1697  002740  012737  000001  001312  40$:  MOV     #1,$DEVM        ;INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
1698  002746  113737  001220  001410        MOVB    $TMP1,DZNUM     ;COPY THE NUMBER OF DEVICES
1699  002754  113737  001220  001411        MOVB    $TMP1,SAVNUM    ;COPY A BACKUP NUMBER
1700  002762  005337  001220          62$:  DEC     $TMP1           ;$TMP1 CONTAINS THE COUNT OF UNINITIALIZED
1701  002766  001404                       BEQ     61$             ; SELECTED DEVICES
1702  002770  000261                       SEC                     ;SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
1703  002772  006137  001312                ROL     $DEVM           ;POINT TO THE NEXT DEVICE
1704  002776  000771                       BR      62$             ;GO DO THIS PROCEDURE AGAIN
1705  003000  013737  001312  001222  61$:  MOV     $DEVM,$TMP2     ;# OF TIMES
1706  003006  013737  001312  001404        MOV     $DEVM,DZACTV    ;COPY THE ACTIVE DEVICE PARAMETER
1707  003014  012700  001500                MOV     #DZCR0,R0       ;SET A POINTER TO THE SPECIFIED INFORMATION
1708  003020  012701  001514                MOV     #DZCR1,R1       ;POINT R1 TO THE REST OF THE MAP TABLE
1709  003024  012702  001320                MOV     #$DDW0,R2       ;POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
1710  003030  000241                       CLC                     ;INITIALIZE THE "C" BIT FOR A ROTATION
1711  003032  006037  001222                ROR     $TMP2           ;SKIP MAPPING SETUP FOR DEVICE 0- IT'S DONE
1712  003036  006237  001222          64$:  ASR     $TMP2           ;ISOLATE A SELECTION FLAG IN THE "C" BIT
1713  003042  103404                       BCS     41$             ;IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
1714  003044  012711  177777                MOV     #-1,(R1)        ;TERMINATE THE LIST
1715  003050  000137  004244                JMP     63$             ;GO TO THE NEXT BLOCK
```

```
1716  003054  012011                    41$:    MOV     (R0)+,(R1)      ;ADDRESS
1717  003056  062721  000010                    ADD     #10,(R1)+       ;POINT TO THE NEXT DZ11 ADDRESS VALUE
1718  003062  012011                            MOV     (R0)+,(R1)      ;VECTOR
1719  003064  062721  000010                    ADD     #10,(R1)+       ;POINT TO THE NEXT VECTOR VALUE
1720  003070  012021                            MOV     (R0)+,(R1)+     ;LEVEL
1721  003072  012021                            MOV     (R0)+,(R1)+     ;LINES
1722  003074  016012  177774                    MOV     -4(R0),(R2)     ;GET THE EIA FLAG FROM THE PRIORITY WORD
1723  003100  042712  077777                    BIC     #77777,(R2)     ;ISOLATE THAT FLAG
1724  003104  051022                            BIS     (R0),(R2)+      ;ADD PARAMETERS TO DEVICE DESCRIPTOR WORD
1725  003106  012021                            MOV     (R0)+,(R1)+     ;PARAMETERS
1726  003110  012021                            MOV     (R0)+,(R1)+     ;MAINTENANCE MODE
1727  003112  000751                            BR      64$
1728  003114  032777  000010  176036    73$:    BIT     #SW03,@SWR      ;ASK PARAMETERS ?
1729  003122  001002                            BNE     42$             ;IF NO, GO DO AUTO SIZING
1730  003124  000137  004244                    JMP     63$             ;GO SET UP FOR AUTO SIZING
1731  003130  004737  003144            42$:    JSR     PC,23$          ;GO ASK PARAMETERS
1732  003134  105337  001415                    DECB    INIFLG          ;INSURE NO AUTO SIZE IF QUESTIONS ANSWERED
1733  003140  000137  004270                    JMP     16$             ;GO TO THE NEXT BLOCK
1734
1735                                            ;GET THE ACTIVE LINES PARAMETER
1736
1737  003144                            23$:
1738  003144  104403                            INSTR                   ;CALL THE STRING INPUT ROUTINE
1739  003146  003464                            69$                     ;POINTER TO MESSAGE TO BE PRINTED
1740  003150  104405                            PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1741  003152  000001                            1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1742  003154  000377                            377                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1743  003156  001506                            LINE0                   ;POINTER TO MAP LOCATION TO BE FILLED
1744  003160     000                            .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1745  003161     001                            .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1746  003162  105037  001416                    CLRB    HDRFLG          ;MAKE SURE THE CHANGES ARE PRINTED
1747
1748                                    ;THIS SEGMENT CHECKS TO MAKE SURE THE LINE PARAMETER JUST ENTERED
1749                                    ;IS LEGITIMATE IN STAGGERED MODE OPERATION IF THAT MODE WAS SELECTED
1750
1751  003166  005737  001512                    TST     MANTO           ;IS STAGGERED THE MODE OF OPERATION?
1752  003172  100021                            BPL     26$             ;IF NOT, SKIP THIS SEGMENT
1753  003174  013703  001506                    MOV     LINE0,R3        ;GET A SCRATCH COPY OF THE ACTIVE LINES
1754  003200  006003                    24$:    ROR     R3              ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
1755  003202  103410                            BCS     25$             ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
1756  003204  001414                            BEQ     26$             ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
1757  003206  006203                            ASR     R3              ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
1758  003210  103373                            BCC     24$             ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
1759  003212  104402  001230            27$:    TYPE    ,$QUES          ;THIS IS AN INCORRECT PARAMETER
1760  003216  104402  010423                    TYPE    ,MBADLN         ;LET THE USER KNOW ABOUT IT
1761  003222  000750                            BR      23$             ;GO GET THE CORRECT PARAMETER
1762  003224  001772                    25$:    BEQ     27$             ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
1763  003226  006203                            ASR     R3              ;GET THE NEXT FLAG
1764  003230  103370                            BCC     27$             ;IF IT ISN'T SET, THERE'S AN ERROR
1765  003232  000241                            CLC                     ;INITIALIZE THE "C" BIT FOR TESTING OF THE NEXT PAIR
1766  003234  000761                            BR      24$             ;GO TEST THE NEXT PAIR OF FLAGS
1767
1768                                            ;GET THE LINE PARAMETER REGISTER ARGUMENT
1769
1770  003236                            26$:
1771  003236  104403                            INSTR                   ;CALL THE STRING INPUT ROUTINE
```

```
                                        70$                     ;POINTER TO MESSAGE TO BE PRINTED
                                        PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                                        0                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                        17                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                        PAR0                    ;POINTER TO MAP LOCATION TO BE FILLED
                                        .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
                                        .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
                                        MOV     #LINE0,R2       ;POINT TO THE LINE SELECTION PARAMETER
                                        MOV     #PAR0,R3        ;POINT TO THE CHOSEN PARAMETERS
                                        MOV     (R3),R4         ;USE BAUD RATE AS AN INDEX IN DELAY TABLE
                                        ASL     R4              ;ALIGN INDEX ON WORD BOUNDARY
                                        MOV     DLYTBL(R4),DLYCNT ;SET THE DELAY COUNT FOR THIS BAUD RATE
                                        SWAB    (R3)            ;PLACE IN HIGH BYTE
                                        BIS     #10070,(R3)     ;PLACE EXTRA PARAMETERS INTO LOC
                                28$:    MOV     (R2),14(R2)     ;LOAD THE LINES
                                        MOV     (R3),14(R3)     ;LOAD THE PARAMETERS
                                        ADD     #14,R2          ;POINT TO THE NEXT SET
                                        ADD     #14,R3          ;  .. OF BOTH PARAMETERS
                                        CMP     R3,#PAR17       ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
                                        BNE     28$             ;IF NOT, GO LOAD SOME MORE PARAMETERS
                                        RTS     PC              ;RETURN TO CALLING BLOCK
                                66$:    .ASCIZ  <200>/1ST CSR ADDRESS (160000:163700):  /
                                67$:    .ASCIZ  <200>/1ST VECTOR ADDRESS (300:770):  /
                                68$:    .ASCIZ  <200>/BR LEVEL (4:6):  /
                                69$:    .ASCIZ  <200>/LINES ACTIVE BY BIT <IN OCTAL>(001:377):  /
                                70$:    .ASCIZ  <200>/DEFAULT BAUD RATE <IN OCTAL>(00:17):  /
                                71$:    .ASCIZ  <200>/# OF DZ11'S <IN OCTAL> (1:20):  /
                                72$:    .ASCII  <200>/MAINTENANCE MODE/
                                        .ASCII  <200>/ [EXTERNAL  <H325>-EIA ONLY      (E)]/
                                        .ASCII  <200>/ [INTERNAL  <DZCSR03=1>          (I)]/
                                        .ASCII  <200>/ [STAGGERED <H3271>-EIA ONLY     (S)]: /
                                        .ASCIZ  <200>/ [STAGGERED <H3190>-20MA ONLY    (S)]: /
                                74$:    .ASCIZ  <200>/TYPE "A" FOR EIA MODULE OR "B" FOR 20 MA (A:B): /
                                75$:    .ASCIZ  <200>/ENTER DELAY PARAMETER: /
                                        .EVEN
                                63$:
                                        CMPB    #377,INIFLG     ;ONLY DO AUTO SIZE ON 1ST START
                                        BNE     16$             ;
                                        BIT     #BIT7,@SWR      ;BIT7=1??
                                        BNE     16$             ;BR IF NO AUTO SIZE
                                        JSR     PC,AUTO.SIZE    ;GO DO THE AUTO SIZE
                                16$:    TSTB    HDRFLG          ;HAS THE TABLE BEEN TYPED YET?
                                        BNE     1$              ;IF SO, DON'T TYPE IT AGAIN
                                        DECB    HDRFLG          ;INDICATE THAT THE TABLE WILL BE TYPED
                                        TYPE    ,XHEAD          ;TYPE MAP HEADER
                                        MOV     #DZ.MAP,R0      ;SET POINTER
                                5$:     MOV     R0,$TMP1        ;POINT TO THE MAP LOCATION
                                        MOV     (R0)+,$TMP2     ;SET DATA
                                        CMP     #-1,$TMP2       ;END OF LIST?
                                        BEQ     1$              ;BR IF YES
                                17$:    CONVRT                  ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
                                        XSTATQ                  ;CONVERT THE DATA AT THIS ADDRESS
                                        BR      5$              ;GO PRINT THE NEXT PARAMETER
                                1$:     TST     @#42            ;IS PROGRAM RUNNING UNDER MONITOR
                                        BNE     3$              ;YES
                                        BIT     #SW06,@SWR      ;DESELECT SPECIFIC DEVICES??
```

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

```
                                        ;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

004562                          $EOP:
004562  000004                          SCOPE
004564  005037  001136                  CLR     $ERRPC          ;CLEAR LAST ERROR PC
004570  105037  001123                  CLRB    $ERFLG          ;CLEAR ERROR FLAG
004574  104402  010063                  TYPE    ,MEPASS         ;TYPE END PASS
004600  104402  010244                  TYPE    ,MCSRX          ;TYPE CSR
004604  104412  004742                  CNVRT   ,XCSR           ;SHOW IT
004610  104402  010252                  TYPE    ,MVECX          ;TYPE VECTOR
004614  104412  004750                  CNVRT   ,XVEC           ;SHOW IT
004620  005237  001242                  INC     $PASS           ;RAISE PASS COUNT
004624  104402  010260                  TYPE    ,MPASSX         ;TYPE PASSES
004630  104412  004756                  CNVRT   ,XPASS          ;SHOW IT
004634  005337  001242                  DEC     $PASS           ;RESTORE PASS COUNT
004640  104402  010271                  TYPE    ,MERRX          ;TYPE ERRORS
004644  104412  004764                  CNVRT   ,XERR           ;SHOW IT
004650  105337  001411                  DECB    SAVNUM          ;ARE ALL DEVICES TESTED?
004654  001030                          BNE     $DOAGN          ;BR IF NO.
004656  113737  001410  001411          MOVB    DZNUM,SAVNUM    ;RESTORE THE COUNT
004664  005037  001226                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
004670  005237  001242                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
004674  042737  100000  001242          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
004702  005327                          DEC     (PC)+           ;;LOOP?
004704  000001          $EOPCT: .WORD   1
004706  003C13                          BGT     $DOAGN          ;;YES
004710  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
004712  000001          $ENDCT: .WORD   1
004714  004704                          $EOPCT
004716  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
004722  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
004724  000005                          RESET                   ;;CLEAR THE WORLD
004726  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
004730  000240                          NOP                     ;;SAVE ROOM
004732  000240                          NOP                     ;;FOR
004734  000240                          NOP                     ;;ACT11
004736                  $DOAGN:
004736  000137                          JMP     @(PC)+          ;;RETURN
004740  011070          $RTNAD: .WORD   CYCLE

004742  000001          XCSR:   1
004744     006  002                   .BYTE   6,2
004746  002042                          DZCSR
004750  000001          XVEC:   1
004752     003  002                   .BYTE   3,2
004754  002072                          DZRIV
```

```
1916  004756  000001                  XPASS:  1
1917  004760     006     002                  .BYTE   6,2
1918  004762  001242                          $PASS
1919  004764  000001                  XERR:   1
1920  004766     006     002                  .BYTE   6,2
1921  004770  001132                          $ERTTL
1922
1923                                          ;SCOPE LOOP AND ITERATION HANDLER
1924                                          ;-------------------------------
1925
1926                                  .SBTTL  SCOPE HANDLER ROUTINE
1927
1928                                  ;;**********************************************************************
1929                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
1930                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
1931                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
1932                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
1933                                  ;*SW14=1         LOOP ON TEST
1934                                  ;*SW11=1         INHIBIT ITERATIONS
1935                                  ;*CALL
1936                                  ;*      SCOPE            ;;SCOPE=IOT
1937
1938  004772                          $SCOPE:
1939  004772  004737  007360          .SCOPE: JSR     PC,SERV.G       ;FIND OUT IF <^G> WAS HIT
1940  004776  005037  001136                  CLR     $ERRPC          ;CLEAR LAST ERROR PC.
1941  005002  022716  012376                  CMP     #TST1+2,(SP)    ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
1942  005006  001413                          BEQ     $XTSTR          ;IF SO, DON'T LOOP ON IT
1943  005010  000406                  TTST:   BR      1$              ;GOTO 1$    (IF LOCK SW02=1; THIS LOC =240)
1944  005012  105777  174146                  TSTB    @$TKS           ;KEYBOARD DONE?
1945  005016  100067                          BPL     $OVER           ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
1946  005020  017766  174142  177776          MOV     @$TKB,-2(SP)    ;CLEAR DONE BIT
1947  005026  032777  040000  174124  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
1948  005034  001060                          BNE     $OVER           ;;YES IF SW14=1
1949                                  ;#####START OF CODE FOR THE XOR TESTER#####
1950  005036  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
1951                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
1952  005040  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
1953  005044  012737  005064  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
1954  005052  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
1955  005056  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
1956  005062  000436                          BR      $SVLAD          ;;GO TO THE NEXT TEST
1957  005064  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
1958  005066  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
1959  005072  000441                          BR      $OVER           ;;LOOP ON THE PRESENT TEST
1960  005074                          6$:;#####END OF CODE FOR THE XOR TESTER#####
1961  005074  105737  001123          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
1962  005100  001404                          BEQ     3$              ;;BR IF NO
1963  005102  105037  001123          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
1964  005106  005037  001226                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
1965  005112  032777  004000  174040  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
1966  005120  001011                          BNE     1$              ;;BR IF YES
1967  005122  005737  001242                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
1968  005126  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
1969  005130  005237  001124                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
1970  005134  023737  001226  001124          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
1971  005142  002015                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
```

```
 1972  005144  012737  000001  001124  1$:      MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
 1973  005152  013737  005234  001226           MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
 1974  005160  105237  001122           $SVLAD: INCB    $TSTNM            ;;COUNT TEST NUMBERS
 1975  005164  113737  001122  001240           MOVB    $TSTNM,$TESTN     ;;SET TEST NUMBER IN APT MAILBOX
 1976  005172  011637  001126                    MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
 1977  005176  013777  001122  173756  $OVER:   MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
 1978  005204  013716  001126                    MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
 1979  005210  105037  001417           3$:     CLRB    MNTFLG            ;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TEST
 1980  005214  005737  001370                    TST     MODE              ;HAS THE MODE BEEN CHANGED?
 1981  005220  001003                            BNE     4$                ;IF NOT INTERNAL , GO DO A TEST
 1982  005222  112737  000010  001417           MOVB    #MAINT,MNTFLG     ;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
 1983  005230  000002                    4$:     RTI                       ;GO DO THE TEST
 1984  005232  000406                    BRW:    406
 1985  005234  000005                    $MXCNT: 5                         ;;MAX. NUMBER OF ITERATIONS
 1986
 1987                                            ;CHECK FOR FREEZE ON CURRENT DATA
 1988                                            ;-------------------------------
 1989
 1990  005236  032777  001000  173714  .SCOP1: BIT     #SW09,@SWR        ;IS SW09=1(SET)?
 1991  005244  001405                            BEQ     1$                ;BR IF NOT SET.
 1992  005246  005737  001362                    TST     LOCK              ;IS THER A TIGHT LOOP SPECIFIED?
 1993  005252  001402                            BEQ     1$                ;IF NO, RETURN
 1994  005254  013716  001362                    MOV     LOCK,(SP)         ;IF YES, GOTO THE ADDRESS IN LOCK.
 1995  005260  000002                    1$:     RTI                       ;GO BACK.
 1996
 1997  005262  032777  010000  173670  .TYPE:  BIT     #SW12,@SWR        ;INHIBIT ALL PRINTOUT??
 1998  005270  001403                            BEQ     1$                ;IF NOT, GO TYPE
 1999  005272  062716  000002                    ADD     #2,(SP)           ;SKIP OVER MESSAGE POINTER
 2000  005276  000002                            RTI                       ;RETURN TO WHERE PROCEDURE WAS INVOKED
 2001  005300                            1$:
 2002                                    .SBTTL  TYPE ROUTINE
 2003
 2004                                    ;;**********************************************************************
 2005                                    ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 2006                                    ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 2007                                    ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 2008                                    ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 2009                                    ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 2010                                    ;*
 2011                                    ;*CALL:
 2012                                    ;*1) USING A TRAP INSTRUCTION
 2013                                    ;*      TYPE    ,MESADR           ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 2014                                    ;*OR
 2015                                    ;*      TYPE
 2016                                    ;*      MESADR
 2017                                    ;*
 2018
 2019  005300  105737  001177           $TYPE:  TSTB    $TPFLG            ;;IS THERE A TERMINAL?
 2020  005304  100002                            BPL     1$                ;;BR IF YES
 2021  005306  000000                            HALT                      ;;HALT HERE IF NO TERMINAL
 2022  005310  000430                            BR      3$                ;;LEAVE
 2023  005312  010046                    1$:     MOV     R0,-(SP)          ;;SAVE R0
 2024  005314  017600  000002                    MOV     @2(SP),R0         ;;GET ADDRESS OF ASCIZ STRING
 2025  005320  122737  000001  001254           CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
 2026  005326  001011                            BNE     62$               ;;NO,GO CHECK FOR APT CONSOLE
 2027  005330  132737  000100  001255           BITB    #APTSPOOL,$ENVM   ;;SPOOL MESSAGE TO APT
```

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

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

```
                                        ;-------------------------
                                .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
                                        MOV     R4,-(SP)        ;SAVE R4 ON STACK
006030  010346                         MOV     @4(SP),.MSG     ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
006032  010446
006034  017637  000004  006052         MOV     #2,4(SP)        ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
006042  062766  000002  000004
006050  104402                 .INST1: TYPE                    ;PRINT THE MESSAGE
006052  000000                 .MSG:   0                       ;MESSAGE IS POINTED TO FROM HERE
006054  012704  010620                 MOV     #INBUF,R4       ;POINT R4 TO THE INPUT BUFFER
006060  012703  000007                 MOV     #7,R3           ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
006064  105777  173074         1$:     TSTB    @$TKS           ;HAS A CHARACTER BEEN RECEIVED?
006070  100375                         BPL     1$              ;IF NO, KEEP WAITING FOR IT
006072  117714  173070                 MOVB    @$TKB,(R4)      ;IF YES, SAVE IT IN THE INPUT BUFFER
006076  142714  000200                 BICB    #200,(R4)       ;KEEP ONLY THE 7-BIT ASCII INFORMATION
006102  122427  000015                 CMPB    (R4)+,#15       ;IS THIS CHARACTER A LINE FEED?
006106  001417                         BEQ     INSTR2          ;IF SO, TERMINATE THE INPUT SEQUENCE
006110  105777  173054         2$:     TSTB    @$TPS           ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
006114  100375                         BPL     2$              ;IF WE CAN'T, WAIT UNTIL WE CAN
006116  017777  173044  173046         MOV     @$TKB,@$TPB     ;ECHO THE CHARACTER BACK
006124  005303                         DEC     R3              ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
006126  001356                         BNE     1$              ;IF WE DON'T HAVE 7, GO GET SOME MORE
006130  012604                         MOV     (SP)+,R4        ;IF WE HAVE 7, RESTORE R4
006132  012603                         MOV     (SP)+,R3        ;RESTORE R3
006134  010346                 .INSTE: MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
006136  010446                         MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
006140  104402  001230                 TYPE    ,$QUES          ;PRINT A QUESTION MARK... WHAT'S GOING ON?
006144  000741                         BR      .INST1          ;GO PRINT THE MESSAGE AGAIN
006146  012604                 INSTR2: MOV     (SP)+,R4        ;RESTORE R4
006150  012603                         MOV     (SP)+,R3        ;RESTORE R3
006152  000002                         RTI                     ;RETURN TO THE MAIN PROCEDURE

                                        ;CONVERT ASCII STRING TO OCTAL
                                        ;-----------------------------

006154  010546                 .PARAM: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
006156  010446                         MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
006160  016605  000004                 MOV     4(SP),R5        ;GET THE SETUP INFORMATION POINTER
006164  012537  006344                 MOV     (R5)+,LOLIM     ;SET THE LOW LIMIT FOR THE INPUT
006170  012537  006346                 MOV     (R5)+,HILIM     ;SET THE HIGH LIMIT FOR THE INPUT
006174  012537  006350                 MOV     (R5)+,DEVADR    ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
006200  112537  006352                 MOVB    (R5)+,LOBITS    ;GET THE MASK OF THE INCORRECT BITS
006204  112537  006353                 MOVB    (R5)+,ADRCNT    ;GET THE COUNT OF ITEMS TO BE STORED
006210  010566  000004                 MOV     R5,4(SP)        ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
006214  005005                 PARAM1: CLR     R5              ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
006216  012704  010620                 MOV     #INBUF,R4       ;POINT TO THE INPUT BUFFER
006222  122714  000015                 CMPB    #15,(R4)        ;IS THIS CHARACTER A CARRIAGE RETURN?
006226  001420                         BEQ     PARERR          ;IF SO, PRINT THE MESSAGE AGAIN
006230  121427  000060         1$:     CMPB    (R4),#60        ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
006234  002415                         BLT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
006236  121427  000067                 CMPB    (R4),#67        ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
006242  003012                         BGT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
006244  142714  000060                 BICB    #60,(R4)        ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
006250  152405                         BISB    (R4)+,R5        ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
006252  122714  000015                 CMPB    #15,(R4)        ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
006256  001406                         BEQ     LIMITS          ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
006260  006305                         ASL     R5              ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
```

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

```
                                        ;RESTORE R5
                                        ;LEAVE

                                        ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
                                        ;-------------------------------------------------------

.CONVR: TYPE    ,$CRLF                  ;PRINT A CARRIAGE RETURN
.CNVRT: MOV     R0,-(SP)                ;SAVE R0
        MOV     R1,-(SP)                ;SAVE R1
        MOV     R3,-(SP)                ;SAVE R3
        MOV     R4,-(SP)                ;SAVE R4
        MOV     R5,-(SP)                ;SAVE R5
        MOV     @12(SP),R1              ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
        ADD     #2,12(SP)               ;POINT TO WHERE MAIN PROGRAM WILL RESUME
        MOV     (R1)+,WRDCNT            ;GET NUMBER OF WORDS TO BE PRINTED
1$:     MOVB    (R1)+,R5                ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
        MOVB    (R1)+,R0                ;GET THE NUMBER OF SPACES TO PRINT
        MOV     @(R1)+,R4               ;COPY THE WORD TO BE CONVERTED
        MOVB    R5,CHRCNT               ;COPY THE CHARACTER COUNT
3$:     MOV     R4,R3                   ;COPY THE ARGUMENT WORD AGAIN
        BIC     #^C<7>,R3               ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
        ADD     #060,R3                 ;MAKE AN ASCII CHARACTER OUT OF THEM
        MOVB    R3,-(SP)                ;SAVE THAT CHARACTER
        ROR     R4                      ;MOVE THE NEXT THREE BITS INTO PLACE
        ASR     R4                      ;MOVE THEM AGAIN
        ASR     R4                      ;AND FINALLY A THIRD TIME
        DEC     R5                      ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
                                        ;BUILT?
        BNE     3$                      ;IF NO, GO BUILD THE NEXT ONE.
        MOV     #MDATA,R3               ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
4$:     MOVB    (SP)+,(R3)+             ;STORE THE CHARACTER, STARTING WITH THE MOST
        DECB    CHRCNT                  ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
        BNE     4$                      ;IF NO, GO TRANSFER ANOTHER
        TSTB    R0                      ;ARE ANY SPACES TO BE PRINTED?
        BEQ     6$                      ;IF NO, DON'T SET UP ANY
5$:     MOVB    #040,(R3)+              ;ADD A SPACE TO THE OUTPUT BUFFER
        DECB    R0                      ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
        BNE     5$                      ;IF YES, GO ADD ANOTHER SPACE
6$:     CLRB    (R3)                    ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
        TYPE    ,MDATA                  ;PRINT THE STRING WE JUST BUILT
        DEC     WRDCNT                  ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
        BNE     1$                      ;IF YES, GO CONVERT THEM
        MOV     (SP)+,R5                ;RESTORE R5
        MOV     (SP)+,R4                ;RESTORE R4
        MOV     (SP)+,R3                ;RESTORE R3
        MOV     (SP)+,R1                ;RESTORE R1
        MOV     (SP)+,R0                ;RESTORE R0
        RTI                             ;RETURN TO THE MAIN PROGRAM
WRDCNT: 0
CHRCNT: .BYTE                           ;NUMBER OF CHARACTERS TO PRINT
SPACNT: .BYTE   0                       ;NUMBER OF SPACES TO PRINT

BINWRD: 0


        ;TRAP DISPATCH SERVICE
```

```
                                        ;ARGUMENT OF TRAP IS EXTRACTED
                                        ;AND USED AS OFFSET TO OBTAIN POINTER
                                        ;TO SELECTED SUBROUTINE

006630  010046                .TRPSR: MOV     R0,-(SP)        ;SAVE R0. USE R0 TO FIND TRAP ROUTINE
006632  016600  000002                MOV     2(SP),R0        ;GET TRAP ADDRESS
006636  005740                        TST     -(R0)           ;GET TRAP
006640  111000                        MOVB    (R0),R0         ;GET RIGHT BYTE OF TRAP(TRAP OFFSET)
006642  006300                        ASL     R0              ;POSITION OFFSET FOR TABLE INDEXING
006644  016000  002002                MOV     .TRPTAB(R0),R0  ;PLACE INDEXED ADDRESS OF TABLE IN R0
006650  000200                        RTS     R0              ;TRANSFER TO THAT ADDRESS AND RESTORE OLD R0

                                        ;DEVICE CLEAR ROUTINE
                                        ;ISSUE A DEVICE CLEAR
                                        ;--------------------
006652                          .DEVICE.CLR:
006652  052777  000020  173162          BIS     #DCLR,@DZCSR    ;SET DCLR
006660  032777  000020  173154  1$:     BIT     #DCLR,@DZCSR    ;DID IT CLEAR?
006666  001374                        BNE     1$              ;BR IF NO
006670  000002                        RTI                     ;EXIT ROUTINE

                                        ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
                                        ;-----------------------------------------------------------
006672  104413                .DCLASM:DEVICE.CLR              ;ISSUE A DEVICE CLEAR
006674  153777  001417  173140          BISB    MNTFLG,@DZCSR   ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
006702  000002                        RTI                     ;RETURN TO CALLING ROUTINE

006704                          .DELAY:
006704  010046                        MOV     R0,-(SP)        ;SAVE R0
006706  013700  006722                MOV     DLYCNT,R0       ;SET COUNT
006712  005300                1$:     DEC     R0              ;DELAY
006714  001376                        BNE     1$              ;
006716  012600                        MOV     (SP)+,R0        ;RESTORE R0
006720  000002                        RTI                     ;LEAVE ROUTINE
006722  000001                DLYCNT: .WORD     1             ;PATCHABLE LOC FOR MORE TIME

                                        ;ADVANCE TO NEXT TEST HANDLER
                                        ;----------------------------
006724  013716  001360        .ADVANCE:MOV    NEXT,(SP)       ;CRUNCH STACK WITH ADDRESS OF SCOPE CALL
006730  005037  001362                CLR     LOCK            ;RESET TIGHT LOOP ADDRESS
006734  000002                        RTI                     ;CHECK TO SEE IF OLD TEST GETS REPEATED

                                        ;ERROR HANDLER
                                        ;-------------
006736  004737  007360        $ERROR: JSR     PC,SERV.G       ;FIND OUT IF <^G> WAS HIT
006742  032777  010000  172210          BIT     #SW12,@SWR      ;BELL ON ERROR?
006750  001406                        BEQ     XBX             ;BR IF NO BELL
006752  105777  172212                TSTB    @$TPS           ;TTY READY.
006756  100003                        BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
006760  112777  000207  172204        MOVB    #207,@$TPB      ;PUSH A BELL AT THE TTY.
006766  032777  020000  172164  XBX:  BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
006774  001113                        BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
006776  021637  001136                CMP     (SP),$ERRPC     ;WAS THIS ERROR FOUND LAST TIME?
007002  001404                        BEQ     1$              ;BR IF YES
```

```
2364  007004  011637  001136                MOV     (SP),$ERRPC     ;RECORD BEING HERE
2365  007010  105037  001123                CLRB    $ERFLG          ;PREPARE HEADER
2366  007014  104407                1$:     SAV05                   ;SAVE ALL PROC REGISTERS
2367  007016  011605                        MOV     (SP),R5         ;GET THE PC OF ERROR
2368  007020  162705  000002                SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
2369  007024  011504                        MOV     (R5),R4         ;GET ERROR INSTRUCTION
2370  007026  110437  001134                MOVB    R4,$ITEMB       ;COPY TEST NUMBER FOR APT HANDLING
2371  007032  006304                        ASL     R4              ;MULT BY TWO
2372  007034  061504                        ADD     (R5),R4         ;DOUBLE IT
2373  007036  006304                        ASL     R4              ;MULT AGAIN
2374  007040  042704  177001                BIC     #177001,R4      ;CLEAR JUNK
2375  007044  062704  027064                ADD     #.ERRTAB,R4     ;GET POINTER
2376  007050  012437  007174                MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
2377  007054  012437  007206                MOV     (R4)+,DATAHD    ;GET DATA HEADRER
2378  007060  011437  007220                MOV     (R4),DATABP     ;GET DATA TABLE
2379  007064  105737  001123                TSTB    $ERFLG          ;TYPE HEADER
2380  007070  001403                        BEQ     TYPMSG          ;BR IF YES
2381  007072  005737  007220                TST     DATABP          ;DOES DATA TABLE EXIST?
2382  007076  001044                        BNE     TYPDAT          ;BR IF YES.
2383  007100  104402  001231        TYPMSG: TYPE    ,$CRLF          ;TYPE A CARRIAGE RETURN
2384  007104  104402  001231                TYPE    ,$CRLF          ;AND TYPE ANOTHER
2385  007110  005737  001362                TST     LOCK
2386  007114  001402                        BEQ     1$
2387  007116  104402  010314                TYPE    ,MASTEK
2388  007122  104402  010302        1$:     TYPE    ,MTSTN
2389  007126  104412  007352                CNVRT   ,XTSTN          ;SHOW IT
2390  007132  104402  010371                TYPE    ,MERRPC         ;TYPE PC.
2391  007136  104412  007344                CNVRT   ,ERTABO         ;SHOW IT
2392  007142  104402  010244                TYPE    ,MCSRX
2393  007146  104412  004742                CNVRT   ,XCSR
2394  007152  104402  001231                TYPE    ,$CRLF          ;GIVE A CR/LF
2395  007156  112737  177777  001123        MOVB    #-1,$ERFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
2396  007164  005737  007174                TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
2397  007170  001402                        BEQ     WTBS.FM         ;BR IF NO.
2398  007172  104402                        TYPE                    ;TYPE
2399  007174  000000        ERRMSG: 0                               ;    ERROR MESSAGE
2400  007176                WTBS.FM:                                ;
2401  007176  005737  007206                TST     DATAHD          ;DATA HEADER?
2402  007202  001402                        BEQ     TYPDAT          ;BR IF NO
2403  007204  104402                        TYPE                    ;TYPE
2404  007206  000000        DATAHD: 0                               ;    DATA HEADER
2405  007210  005737  007220        TYPDAT: TST     DATABP          ;DATA TABLE?
2406  007214  001402                        BEQ     RESREG          ;BR IF NO.
2407  007216  104411                        CONVRT                  ;SHOW
2408  007220  000000        DATABP: 0                               ;    DATA TABLE
2409  007222  104410        RESREG: RES05                           ;RESTORE PROC REGISTERS
2410  007224  122737  000001  001254  HALTS: CMPB    #APTENV,$ENV    ;IS APT RUNNING?
2411  007232  001007                        BNE     2$              ;SKIP APT CALL IF NOT
2412  007234  113737  001134  007246        MOVB    $ITEMB,7$       ;COPY ERROR NUMBER
2413  007242  004737  005600                JSR     PC,$ATY4        ;CALL APT SERVICE
2414  007246  000000        7$:     .WORD   0                       ;ERROR NUMBER STUCK HERE
2415  007250  000777        8$:     BR      8$                      ;LOCK UP HERE
2416  007252  022737  004726  000042  2$:     CMP     #$ENDAD,@#42    ;CHECK TO SEE IF IN ACT-11 MODE
2417  007260  001403                        BEQ     1$              ;IF SO, HANDLE ACCORDINGLY
2418  007262  005777  171672                TST     @SWR            ;HALT ON ERROR?
2419  007266  100004                        BPL     EXITER          ;BR IF NO HALT ON ERROR
```

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

```
2476  007610  001332                        BNE     2$                      ;
2477  007612  104402  001231          5$:   TYPE    ,$CRLF                  ;
2478  007616  000207                  6$:   RTS     PC
2479
2480  007620  020200  051450  051127  89$:  .ASCIZ  <200>? (SWR)=/?
2481  007626  036451  000057
2482                                  .EVEN
2483  007632  000001                  88$:  1
2484  007634     006     000                .BYTE   6,0
2485  007636  007640                        90$
2486  007640  000000                  90$:  .WORD   0
2487  007642  036457  000057          91$:  .ASCIZ  ?/=/?
2488                                  .EVEN
2489                                  .SBTTL  POWER DOWN AND UP ROUTINES
2490
2491                                  ;;**********************************************************************
2492                                  ;POWER DOWN ROUTINE
2493  007646  012737  010012  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
2494  007654  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
2495  007662  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2496  007664  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2497  007666  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2498  007670  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
2499  007672  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
2500  007674  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
2501  007676  017746  171256                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
2502  007702  010637  010016                MOV     SP,$SAVR6       ;;SAVE SP
2503  007706  012737  007720  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
2504  007714  000000                        HALT
2505  007716  000776                        BR      .-2             ;;HANG UP
2506
2507                                  ;;**********************************************************************
2508                                  ;POWER UP ROUTINE
2509  007720  012737  010012  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
2510  007726  013706  010016                MOV     $SAVR6,SP       ;;GET SP
2511  007732  005037  010016                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
2512  007736  005237  010016          1$:   INC     $SAVR6          ;;WAIT FOR THE INC
2513  007742  001375                        BNE     1$              ;;OF  WORD
2514  007744  012677  171210                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
2515  007750  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
2516  007752  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
2517  007754  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
2518  007756  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
2519  007760  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
2520  007762  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
2521  007764  012737  007646  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
2522  007772  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
2523  010000  104402                        TYPE                    ;;REPORT THE POWER FAILURE
2524  010002  010020          $PWRMG: .WORD  MPFAIL                  ;;POWER FAIL MESSAGE POINTER
2525  010004  012716                        MOV     (PC)+,(SP)      ;;RESTART AT RESTART
2526  010006  011434          $PWRAD: .WORD  RESTART                 ;;RESTART ADDRESS
2527  010010  000002                        RTI
2528  010012  000000          $ILLUP: HALT                           ;;THE POWER UP SEQUENCE WAS STARTED
2529  010014  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
2530  010016  000000          $SAVR6: 0                              ;;PUT THE SP HERE
2531  010020  050200  051127  043040  MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /
```

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

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

```
                                        ;
                                        ;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.
                                        ;
011070  005737  001404          CYCLE:  TST     DZACTV          ;ARE ANY DZ11'S TO BE TESTED?
011074  001004                          BNE     1$              ;BR IF OK.
011076  104402  010122                  TYPE    ,MERR2          ;NO DZ11'S SELECTED!!
011102  000000                          HALT                    ;STOP THE SHOW.
011104  000776                          BR      .-2             ;DISQUALIFY CONT. SW.
011106  013737  005234  001226  1$:     MOV     $MXCNT,$TIMES   ;RESTORE THE NUMBER OF ITERATIONS TO MAKE
011114  033737  001406  001404          BIT     RUN,DZACTV      ;IS THIS ONE "ACTIVE"
011122  001020                          BNE     2$              ;BR IF GOOD ONE FOUND.
011124  000241                          CLC
011126  006137  001406                  ROL     RUN             ;UPDATE POINTER
011132  005537  001406                  ADC     RUN             ;CATCH CARRY FROM RUN
011136  062737  000014  001412          ADD     #14,ACTIVE      ;UPDATE ADDRESS POINTER.
011144  022737  002000  001412          CMP     #DZ.END,ACTIVE  ;HAVE WE PASSED THE END OF THE MAP?
011152  001355                          BNE     1$              ;IF NO, KEEP GOING; NOT ALL TESTED FOR.
011154  012737  001500  001412          MOV     #DZ.MAP,ACTIVE  ;RESET ADDRESS POINTER.
011162  000751                          BR      1$              ;KEEP LOOKING FOR ACTIVE DZ11
011164  000241                  2$:     CLC
011166  006137  001406                  ROL     RUN             ;UPDATE POINTER.
011172  005537  001406                  ADC     RUN             ;CATCH CARRY.
011176  013700  001412                  MOV     ACTIVE,R0       ;GET ADDRESS POINTER.
011202  062737  000014  001412          ADD     #14,ACTIVE      ;UPDATE.
011210  022737  002000  001412          CMP     #DZ.END,ACTIVE
                                                                ;ALL DONE?
011216  001003                          BNE     3$              ;BR IF NO.
011220  012737  001500  001412          MOV     #DZ.MAP,ACTIVE  ;RESTORE POINTER.
011226  012037  001310          3$:     MOV     (R0)+,$BASE     ;LOAD SYSTEM CTRL. REG
011232  012037  002072                  MOV     (R0)+,DZRIV     ;LOAD VECTOR
011236  012037  027060                  MOV     (R0)+,DZPRT     ;LOAD PRIORITY
011242  113737  027061  001414          MOVB    DZPRT+1,EIAFLG  ;EIA OR 20MA
011250  042737  100000  027060          BIC     #BIT15,DZPRT    ;CLEAR FLAG
011256  012037  001364                  MOV     (R0)+,LINE      ;SET UP LINE DZ LINES ACTIVE
011262  012037  001366                  MOV     (R0)+,PAR       ;SET UP PARAMETERIZATION
011266  012037  001370                  MOV     (R0)+,MODE      ;SET UP MAINTENANCE MODE
011272  004737  026652                  JSR     PC,DZLEV        ;SET UP
011276  005737  000042                  TST     @#42            ;ARE WE UNDER MONITOR CONTROL?
011302  001051                          BNE     4$              ;IF YES, SKIP THIS SETUP
011304  032777  000002  167646          BIT     #SW01,@SWR      ;IF SW01=1, GET STARTING TEST #
011312  001445                          BEQ     4$              ;BR IF NO TEST IS TO BE INPUTTED
011314  104402  001231          7$:     TYPE    ,$CRLF
011320  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
011322  010302                          MTSTN                   ;POINTER TO MESSAGE TO BE PRINTED
011324  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
011326  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
011330  001000                          1000                    ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
011332  001122                          $TSTNM                  ;POINTER TO MAP LOCATION TO BE FILLED
011334     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
011335     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
```

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

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

011440  012700  001500          SETAPT: MOV     #DZ.MAP,R0      ;POINT TO THE DEVICE MAP TABLE
011444  013701  001310                  MOV     $BASE,R1        ;BUILD DEVICE ADDRESSES IN R1
011450  013702  001304                  MOV     $VECT1,R2       ;BUILD DEVICE VECTORS IN R2
011454  042702  177007                  BIC     #^C<770>,R2     ;STRIP AWAY OTHER INFORMATION

011460  113703  001305                  MOVB    $VECT1+1,R3     ;LOAD THE INTERRUPT PRIORITY FROM R3
011464  106003                          RORB    R3              ;ALIGN THE NUMBER
011466  106003                          RORB    R3              ;ALIGN THE NUMBER
011470  106003                          RORB    R3              ;ALIGN THE NUMBER
011472  106003                          RORB    R3              ;ALIGN THE NUMBER
011474  106003                          RORB    R3              ;ALIGN THE NUMBER
011476  042703  177770                  BIC     #^C<7>,R3       ;REMOVE ALL BUT BUS LEVEL NUMBER
011502  012704  001320                  MOV     #$DDW0,R4       ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
011506  013705  001312                  MOV     $DEVM,R5        ;GET THE MAP OF ACTIVE DEVICES
011512  010537  001404                  MOV     R5,DZACTV       ;SAVE THE BIT MAP
011516  006005                  1$:     ROR     R5              ;GET A DEVICE SELECTION BIT
011520  103407                          BCS     3$              ;IF IT IS SELECTED, GO SET UP A MAP
011522  001425                          BEQ     5$              ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
011524  005724                          TST     (R4)+           ;POINT TO NEXT DEVICE DESCRIPTOR
011526  062701  000010          2$:     ADD     #10,R1          ;SET UP THE NEXT ADDRESS
011532  062702  000010                  ADD     #10,R2          ;SET UP THE NEXT VECTOR GROUP
011536  000767                          BR      1$              ;GO SEE IF MORE DEVICES REMAIN
011540  010120                  3$:     MOV     R1,(R0)+        ;LOAD DEVICE ADDRESS
011542  010220                          MOV     R2,(R0)+        ;LOAD THE VECTOR ADDRESS
011544  010320                          MOV     R3,(R0)+        ;LOAD THE INTERRUPT PRIORITY LEVEL
011546  013720  001314                  MOV     $CDW1,(R0)+     ;GET THE NUMBER OF LINES IN OPERATION
011552  012420                          MOV     (R4)+,(R0)+     ;LOAD DEVICE PARAMETERS
011554  100006                          BPL     4$              ;IF 20MA MODE SELECTED, SET IT UP
011556  052760  100000  177772          BIS     #100000,-6(R0)  ;SET THE 20MA FLAG IN DZLVN
011564  042760  100000  177776          BIC     #100000,-2(R0)  ;CLEAR THE FLAG IN DZPARN
011572  005020                          CLR     (R0)+           ;DEFAULT OPERATION TO INTERNAL MAINTENANCE MODE
011574  000754                          BR      2$              ;GO BUILD THE NEXT ADDRESS
011576  012710  177777          5$:     MOV     #-1,(R0)        ;TERMINATE THE DEVICE MAP
011602  012737  001256  001160          MOV     #$SWREG,SWR     ;SET TO SOFTWARE APT SWITCH REGISTER
011610  000207                          RTS     PC              ;RETURN TO PRINT STATUS TABLE


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

011612                          AUTO.SIZE:
011612  000005                          .RESET                  ;INSURE A BUS INIT.
011614  105337  001415                  DECB    INIFLG          ;SHOW THAT I WAS HERE
011620  012702  001500          CSRMAP: MOV     #DZ.MAP,R2      ;LOAD MAP POINTER.
011624  012703  001320                  MOV     #$DDW0,R3       ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
011630  005022                  1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
011632  022702  002000                  CMP     #DZ.END,R2      ;ALL DONE?
```

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

```
2781  012120  000261                          SEC
2782  012122  006137  001404                  ROL     DZACTV
2783  012126  000772                          BR      4$
2784  012130  013737  001500  001310  98$:    MOV     DZCR0,$BASE     ;POINT TO THE ADDRESS OF FIRST DEVICE
2785  012136  013737  001512  001314          MOV     MANT0,$CDW1     ;INDICATE TO ETABLE WHAT MODE IS BEING USED
2786  012144  012737  000006  000004  99$:    MOV     #6,@#4          ;RESTORE TRAP VECTOR
2787  012152  013737  001404  001312          MOV     DZACTV,$DEVM    ;SAVE ACTIVE REGISTER
2788  012160  000410                          BR      VECMAP          ;GO FIND THE VECTOR NOW.
2789  012162  104402  010122          5$:     TYPE    ,MERR2          ;NOTIFY OPR THAT NO DZ11'S FOUND.
2790  012166  005000                          CLR     R0              ;MAKE DATA DISPLAY ZERO
2791  012170  000000                          HALT                    ;STOP THE SHOW
2792  012172  000776                          BR      .-2             ;DISABLE CONT. SW.
2793  012174  012716  012056          6$:     MOV     #3$,(SP)        ;ENTERED BY NON-EXISTENT TIME-OUT
2794  012200  000002                          RTI                     ;RETURN TO MAINSTREAM
2795
2796  012202  012737  000340  000022  VECMAP: MOV     #340,@#22       ;SET IOT TRAP PRIORITY TO 7
2797  012210  012737  012326  000020          MOV     #4$,@#20        ;SET IOT TRAP VECTOR
2798  012216  012702  001500                  MOV     #DZ.MAP,R2      ;SET SOFTWARE POINTER
2799  012222  012700  000300                  MOV     #300,R0         ;FLOATING VECTORS START HERE.
2800  012226  012701  000302                  MOV     #302,R1         ;PC OF IOT INSTR.
2801  012232  010120          1$:     MOV     R1,(R0)+        ;START FILLING VECTOR AREA
2802  012234  012721  000004                  MOV     #4,(R1)+        ;WITH .+2; IOT
2803  012240  022021                          CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
2804  012242  020127  001000                  CMP     R1,#1000        ;HAS THE VECTOR AREA BEEN EXCEEDED?
2805  012246  101771                          BLOS    1$              ;BR IF MORE TO FILL
2806  012250  013704  001404                  MOV     DZACTV,R4       ;STORE TEMPORARILY
2807  012254  000241          2$:     CLC
2808  012256  006004                          ROR     R4              ;BRING OUT A BIT
2809  012260  103036                          BCC     5$              ;BR IF ALL DONE
2810  012262  106427  000000                  MTPS    #0              ;ZERO CPU PRIO
2811  012266  012772  040040  000000          MOV     #BIT14+BIT5,@(R2)
2812  012274  011201                          MOV     (R2),R1         ;GET CSR
2813  012276  112761  000200  000004          MOVB    #BIT7,4(R1)     ;SET THE TCR BIT!
2814                                           ;ATTEMPT TO FORCE AN INTERRUPT
2815  012304  005200                          INC     R0              ;STALL
2816  012306  001376                          BNE     .-2             ;       FOR TIME TO INTERRUPT
2817  012310  012762  000300  000002          MOV     #300,2(R2)      ;NO INTERRUPT ASSUME 300 AND FIX DZ11 LATER
2818  012316  000005          3$:     RESET                   ;INIT
2819  012320  062702  000014                  ADD     #14,R2          ;POP SOFTWARE POINTER
2820  012324  000753                          BR      2$              ;KEEP GOING
2821  012326  011662  000002          4$:     MOV     (SP),2(R2)      ;GET VECTOR ADDRESS
2822  012332  162762  000010  000002          SUB     #10,2(R2)       ;POINT BACK TO THE CORRECT VECTOR
2823  012340  042762  000007  000002          BIC     #7,2(R2)        ;CLEAR JUNK
2824  012346  022626                          POP2SP                  ;POP IOT JUNK OFF STACK
2825  012350  012716  012316                  MOV     #3$,(SP)        ;SET FOR RETURN
2826  012354  000002                          RTI
2827  012356  013737  001502  001304  5$:     MOV     DZVC0,$VECT1    ;COPY VECTOR OF FIRST DEVICE INTO ETABLE
2828  012364  012737  004772  000020          MOV     #.SCOPE,IOTVEC  ;RESTORE THE SCOPE TRAP
2829  012372  000207                          RTS     PC              ;ALL DONE WITH ''AUTO SIZING''
2830
```

```
                                        ;************************ TEST 1 *********************************
                                        ;*THIS TEST PROVES THE SLAVE SYNC RESPONSE
                                        ;*DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
                                        ;*      DZCSR, DZRBUF, DZTCR, DZMSR
                                ;:*  TEST 1
                                ;;*****************************************************************
012374  000004                TST1:   SCOPE
012376  012737  000001  001122        MOV     #1,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012404  012737  012564  001360        MOV     #TST2,NEXT      ;POINT TO THE START OF THE NEXT TEST
012412  012737  012552  000004        MOV     #5$,4           ;SET TRAP VECTOR
012420  012737  000340  000006        MOV     #PR7,6          ;SET PRIORITY TO LEVEL 7
012426  012737  012434  001362        MOV     #1$,LOCK        ;SET RETURN IF SW09=11
012434  013700  002042            1$: MOV     DZCSR,R0        ;SET ADDRESS TO TEST
012440  011001                        MOV     (R0),R1         ;READ THE ADDRESS
012442  000240                        NOP                     ;WASTE TIME
012444  005010                        CLR     (R0)            ;WRITE THE ADDRESS
012446  000240                        NOP                     ;WASTE TIME
012450  012737  012456  001362        MOV     #2$,LOCK        ;SET RETURN ADDRESS FOR SW09
012456  013700  002046            2$: MOV     DZRBUF,R0       ;SET ADDRESS TO TEST
012462  011001                        MOV     (R0),R1         ;READ THE ADDRESS
012464  000240                        NOP                     ;
012466  005010                        CLR     (R0)            ;WRITE THE ADDRESS
012470  000240                        NOP                     ;WASTE TIME
012472  012737  012500  001362        MOV     #3$,LOCK        ;SET RETURN ADDRESS FOR SW09
012500  013700  002056            3$: MOV     DZTCR,R0        ;SET ADDRESS TO TEST
012504  011001                        MOV     (R0),R1         ;READ THE ADDRESS
012506  000240                        NOP
012510  005010                        CLR     (R0)            ;WRITE THE ADDRESS
012512  000240                        NOP
012514  012737  012522  001362        MOV     #4$,LOCK        ;SET RETURN ADDRESS
012522  013700  002062            4$: MOV     DZMSR,R0        ;SET ADDRESS TO TEST
012526  011001                        MOV     (R0),R1         ;READ FROM ADDRESS
012530  000240                        NOP
012532  005010                        CLR     (R0)            ;WRITE THE ADDRESS
012534  000240                        NOP                     ;
012536  012737  000006  000004        MOV     #6,4            ;SET TRAP CATCHER BACK TO NORMAL
012544  005037  000006                CLR     6               ;
012550  104400                        ADVANCE                 ;SCOPE THIS TEST
012552  011601                    5$: MOV     (SP),R1         ;SAVE PC OF TRAP
012554  022626                        CMP     (SP)+,(SP)+     ;POP TRAP OFF STACK
012556  104001                        ERROR   1               ;*NO SLAVE SYNC RESPONSE.
012560  104401                        SCOP1                   ;SW09=1?
012562  000111                        JMP     (R1)            ;RTI
                                        ;************************ TEST 2 *********************************
                                        ;*THIS TEST PROVES THAT BIT 'DCLR''
                                        ;*CAN BE SET AND THAT IT WILL CLEAR
                                        ;*BY ITSELF AFTER A PERIOD OF TIME.
                                ;:*  TEST 2
                                ;;*****************************************************************
012564  000004                TST2:   SCOPE
012566  012737  000002  001122        MOV     #2,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012574  012737  012650  001360        MOV     #TST3,NEXT      ;POINT TO THE START OF THE NEXT TEST
012602  013700  002042            1$: MOV     DZCSR,R0        ;SET POINTER
012606  012705  000020                MOV     #DCLR,R5        ;SET DCLR
012612  010510                        MOV     R5,(R0)         ;WRITE DCLR INTO DZCSR
```

```
2887  012614  011004                        MOV     (R0),R4         ;READ BACK DZCSR
2888  012616  020504                        CMP     R5,R4           ;DZCSR OK?
2889  012620  001401                        BEQ     1$              ;IF IT IS SET SKIP THE ERROR CALL
2890  012622  104002                        ERROR   2               ;*DCLR SHOULD BE SET..MOMENTARILY
2891                                    ;NOW LETS WATCH IT DISAPPEAR
2892  012624  005002                1$:     CLR     R2              ;SET COUNTER TO 0
2893  012626  005005                        CLR     R5              ;SET EXPECTED TO 0
2894  012630  005003                        CLR     R3              ;DUAL LOOP COUNTER
2895  012632  011004                2$:     MOV     (R0),R4         ;IS DCLR CLEAR?
2896  012634  001405                        BEQ     3$              ;IF YES , GO TO THE NEXT TEST
2897  012636  005203                        INC     R3              ;IF NO,COUNT 1 OF 65535 TICKS
2898                                                                ;THE WORD CREATED BY THE IMMEDIATE 0 WILL BE
2899                                                                ;THE COUNTER
2900  012640  001374                        BNE     2$              ;HAS THE TIME EXPIRED? IF NO, GO TEST BIT AGAIN
2901  012642  005302                        DEC     R2              ;HAS THE TOTAL TIME EXPIRED?
2902  012644  001372                        BNE     2$              ;IF NO, CHECK THE BIT AGAIN
2903  012646  104002                        ERROR   2               ;*DCLR FAILED TO CLEAR
2904  012650                        3$:
2905                                        ;************************ TEST 3 *********************************
2906                                        ;*TEST TO VERIFY THAT BIT 'MAINT'' CAN
2907                                        ;*BE SET. THEN VERIFY THAT BIT 'MAINT'' CAN
2908                                        ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
2909                                        ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
2910                                        ;*CLEARED BY A 'DEVICE CLEAR''
2911                                    ;:*  TEST 3
2912                                    ;;*****************************************************************
2913  012650  000004                TST3:   SCOPE
2914  012652  012737  000003  001122        MOV     #3,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
2915  012660  012737  012742  001360        MOV     #TST4,NEXT      ;POINT TO THE START OF THE NEXT TEST
2916  012666  013700  002042                MOV     DZCSR,R0        ;GET BASE ADDRESS
2917  012672  012705  000010                MOV     #MAINT,R5       ;SET BIT
2918  012676  010510                        MOV     R5,(R0)         ;SET SET IN DEVICE
2919  012700  011004                        MOV     (R0),R4         ;READ THE BIT FROM DEVICE
2920  012702  020504                        CMP     R5,R4           ;WAS BIT SET?
2921  012704  001401                        BEQ     1$              ;BR IF YES
2922  012706  104002                        ERROR   2               ;*BIT R/W FAILURE
2923  012710  040510                1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
2924  012712  011004                        MOV     (R0),R4         ;READ DEVICE
2925  012714  001404                        BEQ     2$              ;BR IF BITS WERE CLEARED.
2926  012716  010546                        MOV     R5,-(SP)        ;SAVE THE BIT
2927  012720  005005                        CLR     R5              ;SET EXPECTED RESULTS TO 0
2928  012722  104002                        ERROR   2               ;*BIT FAILED TO CLEAR
2929  012724  012605                        MOV     (SP)+,R5        ;RESTORE THE BIT.
2930  012726  010510                2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
2931  012730  104413                        DEVICE.CLR              ;ISSUE DEVICE CLEAR
2932  012732  011004                        MOV     (R0),R4         ;READ THE BIT.
2933  012734  001402                        BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
2934  012736  005005                        CLR     R5              ;SET EXPECTED TO ZERO
2935  012740  104002                        ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
2936  012742                        3$:
2937                                        ;************************ TEST 4 *********************************
2938                                        ;*TEST TO VERIFY THAT BIT 'MSENAB'' CAN
2939                                        ;*BE SET. THEN VERIFY THAT BIT 'MSENAB'' CAN
2940                                        ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
2941                                        ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
2942                                        ;*CLEARED BY A 'DEVICE CLEAR''
```

```
                                        ;::*  TEST 4
                                        ;;*************************************************************************
                                        TST4:   SCOPE
                                                MOV     #4,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
                                                MOV     #TST5,NEXT      ;POINT TO THE START OF THE NEXT TEST
                                                MOV     DZCSR,R0        ;GET BASE ADDRESS
                                                MOV     #MSENAB,R5      ;SET BIT
                                                MOV     R5,(R0)         ;SET SET IN DEVICE
                                                MOV     (R0),R4         ;READ THE BIT FROM DEVICE
                                                CMP     R5,R4           ;WAS BIT SET?
                                                BEQ     1$              ;BR IF YES
                                                ERROR   2               ;*BIT R/W FAILURE
                                        1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
                                                MOV     (R0),R4         ;READ DEVICE
                                                BEQ     2$              ;BR IF BITS WERE CLEARED.
                                                MOV     R5,-(SP)        ;SAVE THE BIT
                                                CLR     R5              ;SET EXPECTED RESULTS TO 0
                                              . ERROR   2               ;*BIT FAILED TO CLEAR
                                                MOV     (SP)+,R5        ;RESTORE THE BIT.
                                        2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
                                                DEVICE.CLR              ;ISSUE DEVICE CLEAR
                                                MOV     (R0),R4         ;READ THE BIT.
                                                BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
                                                CLR     R5              ;SET EXPECTED TO ZERO
                                                ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
                                        3$:
                                                ;************************* TEST 5 ****************************
                                                ;*TEST TO VERIFY THAT BIT 'SILOEN' CAN
                                                ;*BE SET. THEN VERIFY THAT BIT 'SILOEN' CAN
                                                ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                                ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                                ;*CLEARED BY A 'DEVICE CLEAR'
                                        ;::*  TEST 5
                                        ;;*************************************************************************
                                        TST5:   SCOPE
                                                MOV     #5,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
                                                MOV     #TST6,NEXT      ;POINT TO THE START OF THE NEXT TEST
                                                MOV     DZCSR,R0        ;GET BASE ADDRESS
                                                MOV     #SILOEN,R5      ;SET BIT
                                                MOV     R5,(R0)         ;SET SET IN DEVICE
                                                MOV     (R0),R4         ;READ THE BIT FROM DEVICE
                                                CMP     R5,R4           ;WAS BIT SET?
                                                BEQ     1$              ;BR IF YES
                                                ERROR   2               ;*BIT R/W FAILURE
                                        1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
                                                MOV     (R0),R4         ;READ DEVICE
                                                BEQ     2$              ;BR IF BITS WERE CLEARED.
                                                MOV     R5,-(SP)        ;SAVE THE BIT
                                                CLR     R5              ;SET EXPECTED RESULTS TO 0
                                                ERROR   2               ;*BIT FAILED TO CLEAR
                                                MOV     (SP)+,R5        ;RESTORE THE BIT.
                                        2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
                                                DEVICE.CLR              ;ISSUE DEVICE CLEAR
                                                MOV     (R0),R4         ;READ THE BIT.
                                                BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
                                                CLR     R5              ;SET EXPECTED TO ZERO
```

```
                                        ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
                                3$:
                                        ;************************ TEST 6 *********************************
                                        ;*TEST TO VERIFY THAT BIT 'RIE' CAN
                                        ;*BE SET. THEN VERIFY THAT BIT 'RIE' CAN
                                        ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                        ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                        ;*CLEARED BY A 'DEVICE CLEAR'
                                ;:*  TEST 6
                                ;;*************************************************************************
013126  000004                  TST6:   SCOPE
013130  012737  000006  001122          MOV     #6,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013136  012737  013220  001360          MOV     #TST7,NEXT      ;POINT TO THE START OF THE NEXT TEST
013144  013700  002042                  MOV     DZCSR,R0        ;GET BASE ADDRESS
013150  012705  000100                  MOV     #RIE,R5 ;SET BIT
013154  010510                          MOV     R5,(R0)         ;SET SET IN DEVICE
013156  011004                          MOV     (R0),R4         ;READ THE BIT FROM DEVICE
013160  020504                          CMP     R5,R4           ;WAS BIT SET?
013162  001401                          BEQ     1$              ;BR IF YES
013164  104002                          ERROR   2               ;*BIT R/W FAILURE
013166  040510                  1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
013170  011004                          MOV     (R0),R4         ;READ DEVICE
013172  001404                          BEQ     2$              ;BR IF BITS WERE CLEARED.
013174  010546                          MOV     R5,-(SP)        ;SAVE THE BIT
013176  005005                          CLR     R5              ;SET EXPECTED RESULTS TO 0
013200  104002                          ERROR   2               ;*BIT FAILED TO CLEAR
013202  012605                          MOV     (SP)+,R5        ;RESTORE THE BIT.
013204  010510                  2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
013206  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR
013210  011004                          MOV     (R0),R4         ;READ THE BIT.
013212  001402                          BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
013214  005005                          CLR     R5              ;SET EXPECTED TO ZERO
013216  104002                          ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
013220                          3$:
                                        ;************************ TEST 7 *********************************
                                        ;*TEST TO VERIFY THAT BIT 'TIE' CAN
                                        ;*BE SET. THEN VERIFY THAT BIT 'TIE' CAN
                                        ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                        ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                        ;*CLEARED BY A 'DEVICE CLEAR'
                                ;:*  TEST 7
                                ;;*************************************************************************
013220  000004                  TST7:   SCOPE
013222  012737  000007  001122          MOV     #7,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013230  012737  013312  001360          MOV     #TST10,NEXT     ;POINT TO THE START OF THE NEXT TEST
013236  013700  002042                  MOV     DZCSR,R0        ;GET BASE ADDRESS
013242  012705  040000                  MOV     #TIE,R5 ;SET BIT
013246  010510                          MOV     R5,(R0)         ;SET SET IN DEVICE
013250  011004                          MOV     (R0),R4         ;READ THE BIT FROM DEVICE
013252  020504                          CMP     R5,R4           ;WAS BIT SET?
013254  001401                          BEQ     1$              ;BR IF YES
013256  104002                          ERROR   2               ;*BIT R/W FAILURE
013260  040510                  1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
013262  011004                          MOV     (R0),R4         ;READ DEVICE
013264  001404                          BEQ     2$              ;BR IF BITS WERE CLEARED.
013266  010546                          MOV     R5,-(SP)        ;SAVE THE BIT
```

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

```
                                        ;* DTR0, DTR1, DTR2, DTR3, DTR4, DTR5, DTR6, DTR7
                                        ;*THIS TEST IS NOT DONE IF MODULE IS 20MA VERSION
                                ;:*  TEST 11
                                ;:*************************************************************************
013450  000004                TST11:  SCOPE
013452  012737  000011  001122        MOV     #11,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
013460  012737  013632  001360        MOV     #TST12,NEXT     ;POINT TO THE START OF THE NEXT TEST
013466  013700  002056              MOV     DZTCR,R0        ;SET DEVICE ADDRESS
013472  012705  000400              MOV     #DTR0,R5        ;SET EXPECTED RESULTS
013476  012737  013514  001362      MOV     #1$,LOCK        ;SET FOR SW09
013504  105737  001414              TSTB    EIAFLG          ;20MA OR EIA
013510  100001                      BPL     1$              ;BR IF EIA
013512  104400                      ADVANCE                 ;EXIT TEST
013514  010510                1$:   MOV     R5,(R0)         ;SET THE BIT
013516  011004                      MOV     (R0),R4         ;READ THE BIT FROM THE DEVICE
013520  105004                      CLRB    R4              ;CLEAR LOW BYTE
013522  020504                      CMP     R5,R4           ;WAS BIT OK?
013524  001401                      BEQ     2$              ;BR IF YES
013526  104002                      ERROR   2               ;*BIT FAILED TO SET.
013530  040510                2$:   BIC     R5,(R0)         ;CLEAR THE BIT
013532  011004                      MOV     (R0),R4         ;READ THE REGISTER
013534  105004                      CLRB    R4              ;CLEAR LOW BYTE
013536  005704                      TST     R4              ;BITS CLEAR?
013540  001404                      BEQ     3$              ;BR IF YES
013542  010546                      MOV     R5,-(SP)        ;SAVE GOOD RESULTS
013544  005005                      CLR     R5              ;SET EXPECTED TO 0
013546  104002                      ERROR   2               ;*REPORT BIT NOT CLEAR
013550  012605                      MOV     (SP)+,R5        ;RESTORE R5
013552  010510                3$:   MOV     R5,(R0)         ;SET THE BIT AGAIN.
013554  104413                      DEVICE.CLR              ;ISSUE DEVICE CLEAR
013556  011004                      MOV     (R0),R4         ;READ THE REGISTER
013560  105004                      CLRB    R4              ;CLEAR LOW BYTE
013562  030510                      BIT     R5,(R0)         ;WAS BIT CLEARED BY DEVICE.CLR?
013564  001001                      BNE     4$              ;BR IF NO (IT SHOULDN'T BE CLEAR)
013566  104002                      ERROR   2               ;*BIT CLEARED BY DEVICE.CLR
013570  104401                4$:   SCOP1                   ;LOCK ON BIT? SW09=1
013572  006305                      ASL     R5              ;CHANGE TO NEXT BIT
013574  001347                      BNE     1$              ;IF NOT DONE LOOP
013576  012710  177400              MOV     #177400,(R0)    ;SET ALL DTR BITS
013602  005005                      CLR     R5              ;CLEAR LOCATION FOR ERROR PRINTOUT
013604  005227  000000        5$:   INC     #0              ;ACT DELAY LOOP FOR
013610  001375                      BNE     5$              ;RESET INSTRUCTION
013612  000005                      RESET                   ;ISSUE A BUS INIT
013614  011004                      MOV     (R0),R4         ;READ REGISTER
013616  105004                      CLRB    R4              ;CLEAR LOW BYTE
013620  005704                      TST     R4              ;DTR BITS CLEAR?
013622  001401                      BEQ     .+4             ;IF YES CONTINUE
013624  104002                      ERROR   2               ;IF NO PRINT ERROR
013626  005037  001362              CLR     LOCK            ;MAKE SURE TIGHT LOOP IS CLEANED UP
                                ;************************* TEST 12 *********************************
                                ;*THIS TEST PERFORMS RESET TESTING &
                                ;*TESTING OF WRITE ONLY OR READ ONLY BIT
                                ;* TEST BITS 'RDONE, BIT11, BIT10, BIT9, BIT8, BIT2, BIT1
                                ;*           BIT0, SILOAL' ARE READ ONLY AND THAT TRDY IS
                                ;*           ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.
                                ;*
```

```
                                        ;::*  TEST 12
                                        ;;*************************************************************************
013632  000004                        TST12:  SCOPE
013634  012737  000012  001122                MOV     #12,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
013642  012737  013750  001360                MOV     #TST13,NEXT     ;POINT TO THE START OF THE NEXT TEST
013650  013700  002042                        MOV     DZCSR,R0        ;SET ADDRESS TO R0
013654  005005                                CLR     R5              ;SET EXPECTED TO 0
013656  012710  027607                        MOV     #RDONE+BIT11+BIT10+BIT9+BIT8+BIT2+BIT1+BIT0+SILOAL,(R0)
                                                                        ;WRITE THE BITS
013662  011004                                MOV     (R0),R4         ;READ BACK THE BITS
013664  001401                                BEQ     1$              ;BR IF NONE ARE SET.
013666  104002                                ERROR   2               ;*BITS WERE SET.
013670  012710  100000                1$:     MOV     #TRDY,(R0)      ;ATTEMPT TO WRITE TRDY
013674  011004                                MOV     (R0),R4         ;READ TRDY
013676  001401                                BEQ     2$              ;BR IF NOT SET
013700  104002                                ERROR   2               ;*
013702  012705  100000                2$:     MOV     #TRDY,R5        ;SET EXPECTED BIT
013706  005077  166140                        CLR     @DZLPR          ;LOAD LINE 0
013712  052777  000001  166136                BIS     #TCR0,@DZTCR    ;SET TCR BIT
013720  052710  000040                        BIS     #MSENAB,(R0)    ;
013724  052705  000040                        BIS     #MSENAB,R5      ;SET SCAN ENABLE
013730  005002                                CLR     R2              ;SET COUNTER TO ZERO
013732  011004                        3$:     MOV     (R0),R4         ;READ THE REGISTER
013734  020504                                CMP     R5,R4           ;BIT SET?
013736  001404                                BEQ     4$              ;BR IF YES
013740  104414                                DELAY                   ;STALL TIME
013742  005202                                INC     R2              ;UPDATE COUNTER
013744  001372                                BNE     3$              ;BR IF COUNTER NOT DONE.
013746  104002                                ERROR   2               ;*TRDY NOT SET!
013750                                  4$:
                                        ;************************** TEST 13 ********************************
                                        ;*THIS TEST PERFORMS RESET TESTING AND
                                        ;*TESTING OF READ ONLY AND WRITE ONLY BITS
                                        ;* IN REGISTER DZCSR
                                        ;*VERIFY THAT 'TIE', 'SILOEN', 'RIE', 'MSENAB', 'MAINT'
                                        ;*ARE THE ONLY R/W BITS IN THE DZCSR.
                                        ;*THEN VERIFY THAT A RESET WILL CLEAR THESE BITS
                                        ;*THIS TEST ALSO CHECKS BYTE OPERATIONS ON THE CSR
                                        ;::*  TEST 13
                                        ;;*************************************************************************
013750  000004                        TST13:  SCOPE
013752  012737  000013  001122                MOV     #13,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
013760  012737  014100  001360                MOV     #TST14,NEXT     ;POINT TO THE START OF THE NEXT TEST
013766  104413                                DEVICE.CLR
013770  013700  002042                        MOV     DZCSR,R0        ;SET UP FOR ERROR MESSAGE
013774  012710  177757                        MOV     #^C<DCLR>,(R0)  ;TRY TO WRITE
014000  012705  050150                        MOV     #TIE!SILOEN!RIE!MSENAB!MAINT,R5 ;MAKE EXPECTED
014004  011004                                MOV     (R0),R4         ;ACTUAL
014006  020405                                CMP     R4,R5           ;CMP EXPECTED VS ACTUAL
014010  001401                                BEQ     1$              ;YES
014012  104002                                ERROR   2               ;*NO
014014  105010                          1$:   CLRB    (R0)            ;CLEAR LOWER BYTE OF CSR
014016  105005                                CLRB    R5              ;SET EXPECTED
014020  011004                                MOV     (R0),R4         ;READ CSR BITS
014022  020405                                CMP     R4,R5           ;COMPARE ACTUAL TO EXPECTED
014024  001401                                BEQ     3$              ;BRANCH IF SAME
```

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

```
3279  014216  012777  177777  165642          MOV     #-1,@DZTDR      ;TRY TO WRITE ALL 1'S
3280  014224  011004                          MOV     (R0),R4         ;ACTUAL
3281  014226  020405                          CMP     R4,R5           ;CMP ACTUAL VS EXPECTED
3282  014230  001401                          BEQ     1$              ;IF YES,GO CONTINUE PROCESSING
3283  014232  104002                          ERROR   2               ;*ERROR- BIT PATTERN NOT CORRECT
3284  014234  010403                  1$:     MOV     R4,R3           ;GET A COPY OF THE ACTUAL BIT PATTERN
3285  014236  005103                          COM     R3              ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
3286  014240  010377  165622                  MOV     R3,@DZTDR       ;TRY TO WRITE
3287  014244  011004                          MOV     (R0),R4         ;ACTUAL
3288  014246  020405                          CMP     R4,R5           ;CMP ACTUAL VS EXPECTED
3289  014250  001401                          BEQ     2$              ;IF YES, GET OUT OF THIS TEST
3290  014252  104002                          ERROR   2               ;*NO
3291  014254                          2$:
3292
3293
3294                                          ;************************* TEST 16 *********************************
3295                                          ;*VERIFY THAT IF WE ARE IN "STAGGERED" MODE
3296                                          ;*THAT SETTING "DTR" FOR A LINE WILL
3297                                          ;*BRING UP "RING" AND "CARRIER" FOR THE
3298                                          ;*ASSOCIATED LINE IN WHICH WE ARE STAGGERED!
3299                                          ;* LINE0 DTR= LINE1 RING AND CARRIER
3300                                          ;* LINE1 DTR= LINE0 RING AND CARRIER
3301                                          ;* LINE2 DTR= LINE3 RING AND CARRIER
3302                                          ;* LINE3 DTR= LINE 4 RING AND CARRIER
3303                                          ;*              ETC...
3304
3305                                  ;:*  TEST 16
3306                                  ;;*****************************************************************
3307  014254  000004                  TST16:  SCOPE
3308  014256  012737  000016  001122          MOV     #16,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3309  014264  012737  014450  001360          MOV     #TST17,NEXT     ;POINT TO THE START OF THE NEXT TEST
3310  014272  012737  014344  001362          MOV     #1$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3311  014300  105737  001414                  TSTB    EIAFLG          ;EIA OR 20MA?
3312  014304  100001                          BPL     10$             ;BR IF EIA
3313  014306  104400                          ADVANCE                 ;EXIT TEST
3314  014310  013700  002062          10$:    MOV     DZMSR,R0        ;SET REGISTER
3315  014314  104413                          DEVICE.CLR              ;INIT DZ11
3316  014316  005003                          CLR     R3              ;ZERO LINE NUMBER
3317  014320  012702  000001                  MOV     #1,R2           ;SET POINTER
3318  014324  005737  001370                  TST     MODE            ;ARE WE IN STAGGERED MODE?
3319  014330  100405                          BMI     1$              ;YES WE ARE!
3320  014332  013737  001360  001126          MOV     NEXT,$LPADR     ;LEAVE THIS TEST! NOT STAGGERED
3321  014340  000177  164562                  JMP     @$LPADR         ;EXIT
3322  014344  130237  001364          1$:     BITB    R2,LINE         ;TEST THIS LINE?
3323  014350  001004                          BNE     3$              ;YES
3324  014352  005203                  2$:     INC     R3              ;LINE #
3325  014354  106302                          ASLB    R2              ;GET NEXT LINE
3326  014356  103372                          BCC     1$              ;KEEP TESTING
3327  014360  104400                          ADVANCE                 ;ADVANCE THIS TEST
3328  014362  010204                  3$:     MOV     R2,R4           ;SAVE BINARY BIT FOR LINE #
3329  014364  032703  000001                  BIT     #BIT0,R3        ;GET STAGGERED COMPANION LINE
3330  014370  001402                          BEQ     4$              ;BR IF LINE EVEN
3331  014372  006204                          ASR     R4              ;ADJUST LINE
3332  014374  000401                          BR      5$              ;
3333  014376  006304                  4$:     ASL     R4              ;ADJUST LINE
3334  014400  005005                  5$:     CLR     R5              ;SET EXPECTED
```

```
3335  014402  150405                          BISB    R4,R5           ;
3336  014404  000305                          SWAB    R5              ;
3337  014406  150405                          BISB    R4,R5           ;
3338  014410  150277  165444                  BISB    R2,@HDZTCR      ;SET DTR
3339  014414  104414                          DELAY                   ;CABLE DELAY
3340  014416  011004                          MOV     (R0),R4         ;READ MSR REGISTER
3341  014420  020504                          CMP     R5,R4           ;OK?
3342  014422  001401                          BEQ     6$              ;YES
3343  014424  104002                          ERROR   2               ;*ERROR IN RING OR CARRIER
3344  014426  140277  165426          6$:     BICB    R2,@HDZTCR      ;CLEAR DTR
3345  014432  104414                          DELAY                   ;CABLE DELAY
3346  014434  011004                          MOV     (R0),R4         ;READ MSR
3347  014436  001402                          BEQ     7$              ;BR IF THEY CLEARED
3348  014440  005005                          CLR     R5              ;SET EXPECTED TO 0
3349  014442  104002                          ERROR   2               ;*BITS NOT CLEARED
3350  014444  104401                  7$:     SCOP1                   ;LOCK ON SIGNAL?
3351  014446  000741                          BR      2$              ;CONTINUE TEST
3352
3353                                          ;************************* TEST 17 *********************************
3354                                          ;*TEST TO VERIFY THAT IF IN 'EXTERNAL'
3355                                          ;*MODE; SETTING DTR FOR SELECTED LINES
3356                                          ;*WILL BRING UP 'CARRIER' AND 'RING'
3357                                          ;*FOR THAT SAME LINE. NOTE: IF YOU HAVE
3358                                          ;*SELECTED MODE AS 'EXTERNAL'; THE H325 TEST CONNECTER
3359                                          ;*MUST BE USED ON ALL SPECIFIED LINES.
3360                                          ;*LINES MAY BE SPECIFIED BY SWR03=1
3361                                          ;*AND SWR00=1 AT START TIME OR ALTERING
3362                                          ;*STATUS MAP.
3363                                  ;:*  TEST 17
3364                                  ;;*****************************************************************
3365  014450  000004                  TST17:  SCOPE
3366  014452  012737  000017  001122          MOV     #17,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3367  014460  012737  014606  001360          MOV     #TST20,NEXT     ;POINT TO THE START OF THE NEXT TEST
3368  014466  012737  014522  001362          MOV     #3$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3369  014474  105737  001370                  TSTB    MODE            ;EXTERNAL?
3370  014500  100401                          BMI     2$              ;BR IF YES
3371  014502  104400                  1$:     ADVANCE                 ;EXIT TEST
3372  014504  105737  001414          2$:     TSTB    EIAFLG          ;YOU BETTER BE IN
3373  014510  100774                          BMI     1$              ;EIA MODE FOR THIS TEST.
3374  014512  013700  002062                  MOV     DZMSR,R0        ;SET REGISTER
3375  014516  012702  000001                  MOV     #1,R2           ;SET LINE POINTER
3376  014522  130237  001364          3$:     BITB    R2,LINE         ;LINE SELECTED?
3377  014526  001003                          BNE     5$              ;BR IF YES
3378  014530  106302                  4$:     ASLB    R2              ;NEXT LINE
3379  014532  103373                          BCC     3$              ;CONTINUE TEST
3380  014534  104400                          ADVANCE                 ;ADVANCE THIS TEST
3381  014536  005005                  5$:     CLR     R5              ;SET EXPECTED
3382  014540  150205                          BISB    R2,R5           ;
3383  014542  000305                          SWAB    R5              ;
3384  014544  150205                          BISB    R2,R5           ;
3385  014546  150277  165306                  BISB    R2,@HDZTCR      ;SET DTR
3386  014552  104414                          DELAY                   ;CABLE DELAY
3387  014554  011004                          MOV     (R0),R4         ;READ MSR
3388  014556  020504                          CMP     R5,R4           ;BITS OK?
3389  014560  001401                          BEQ     6$              ;BR IF YES
3390  014562  104002                          ERROR   2               ;CARRIER OR RING ERROR
```

```
3391  014564  140277  165270          6$:     BICB    R2,@HDZTCR      ;CLEAR DTR
3392  014570  104414                          DELAY                   ;CABLE DELAY
3393  014572  011004                          MOV     (R0),R4         ;READ MSR
3394  014574  001402                          BEQ     7$              ;BR IF BITS CLEARED
3395  014576  005005                          CLR     R5              ;CLEAR EXPECTED LOC.
3396  014600  104002                          ERROR   2               ;BITS NOT CLEARED.
3397  014602  104401                  7$:     SCOP1                   ;LOCK ON LINE?
3398  014604  000751                          BR      4$              ;CONTINUE TEST
3399
3400                                          ;************************ TEST 20 *********************************
3401                                          ;* THIS TEST VERIFIES THAT TRDY IS SET WHEN A LINE
3402                                          ;* IS READY TO BE LOADED, AND THAT THE LINE SPECI-
3403                                          ;* FIED IN BITS 8-10 OF DZCSR CORRESPOND
3404                                          ;* TO THE LINE SELECTED IN DZTCR
3405                                  ;:*  TEST 20
3406                                  ;;*************************************************************************
3407  014606  000004                  TST20:  SCOPE
3408  014610  012737  000020  001122          MOV     #20,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3409  014616  012737  014732  001360          MOV     #TST21,NEXT     ;POINT TO THE START OF THE NEXT TEST
3410  014624  104413                          DEVICE.CLR              ;ISSUE A ''DEVICE CLEAR'' (RESET)
3411  014626  013700  002042                  MOV     DZCSR,R0        ;SET POINTER
3412  014632  012705  100040                  MOV     #MSENAB!TRDY,R5 ;START THE EXPECTED LINE NUMBER AT 0
3413  014636  005037  001372                  CLR     SAVLIN          ;SET UP FOR ERROR PRINTOUTS
3414  014642  012702  000001                  MOV      #1,R2          ;USING R2 AS A BIT POINTER, POINT TO LINE 0
3415  014646  130237  001364          1$:     BITB    R2,LINE         ;IS THIS LINE SELECTED?
3416  014652  001420                          BEQ     5$              ;IF NO, SKIP THE STARTUP
3417  014654  050277  165176          2$:     BIS     R2,@DZTCR       ;SET THE GO BIT FOR THIS LINE
3418  014660  052710  000040                  BIS     #MSENAB,(R0)    ;START THE SCANNER
3419  014664  005004                          CLR     R4              ;SET FOR DELAY
3420  014666  032710  100000          3$:     BIT     #TRDY,(R0)      ;TX READY?
3421  014672  001004                          BNE     4$              ;BR IF YES
3422  014674  104414                          DELAY                   ;DELAY
3423  014676  005204                          INC     R4              ;COUNTER
3424  014700  001372                          BNE     3$              ;BR IF <>0!
3425  014702  104003                          ERROR   3               ;*TX NOT READY!
3426  014704  011004                  4$:     MOV     (R0),R4         ;GET THE LINE POINTED TO BY THE SCANNER
3427  014706  020405                          CMP     R4,R5           ;IS THE LINE NUMBER WHAT IT SHOULD BE?
3428  014710  001401                          BEQ     5$              ;IF YES,GO WORK ON THE NEXT LINE
3429  014712  104002                          ERROR   2               ;*LINE NUMBER DID NOT MATCH TCR BIT
3430  014714  062705  000400          5$:     ADD     #400,R5         ;POINT TO THE NEXT EXPECTED LINE
3431  014720  104413                          DEVICE.CLR              ;ISSUE A ''DEVICE CLEAR'' (RESET)
3432  014722  005237  001372                  INC     SAVLIN          ;ADJUST FOR NEXT LINE
3433  014726  106302                          ASLB    R2              ;POINT TO THE NEXT LINE.ARE ALL LINES TESTED?
3434  014730  103346                          BCC     1$              ;IF NOT, GO DO THE NEXT LINE
3435  014732                          6$:
3436                                          ;************************ TEST 21 *********************************
3437                                          ;*TEST TO TRANSMIT ONE CHAR AND
3438                                          ;*RECEIVE ONE CHAR ON ONE LINE
3439                                          ;*AT A TIME. THE CHAR IS '252'' AND
3440                                          ;*ALL SELECTED LINES WILL BE TURNED ON
3441                                          ;*ONE AT A TIME. THIS IS THE FIRST TIME ANY
3442                                          ;*DATA IS CHECKED IN THE RECEIVER.
3443                                          ;*USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
3444                                          ;*WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.
3445                                  ;:*  TEST 21
3446                                  ;;*************************************************************************
```

```
3447  014732  000004                        TST21:  SCOPE
3448  014734  012737  000021  001122                MOV     #21,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3449  014742  012737  015270  001360                MOV     #TST22,NEXT     ;POINT TO THE START OF THE NEXT TEST
3450  014750  012737  015246  001362                MOV     #16$,LOCK       ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
3451  014756  104417                                DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3452  014760  013701  001366                        MOV     PAR,R1          ;PICK UP PARAMETERS
3453  014764  012702  000001                        MOV     #1,R2           ;PICK UP INIT POINTER
3454  014770  030237  001364            1$:         BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
3455  014774  001402                                BEQ     2$              ;NO
3456  014776  010177  165050                        MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
3457  015002  005201                    2$:         INC     R1              ;POSITION POINTER TO THE NEXT LINE
3458  015004  106302                                ASLB    R2              ;GOT 'EM ALL ?
3459  015006  103370                                BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
3460  015010  005037  001372                        CLR     SAVLIN          ;CLEAR LINE # INDICATOR
3461  015014  012702  000001                        MOV     #1,R2           ;LINE POINTER
3462  015020  052777  000040  165014                BIS     #MSENAB,@DZCSR  ;START SCANNER
3463  015026  030237  001364            3$:         BIT     R2,LINE         ;VALID LINE ?
3464  015032  001462                                BEQ     14$             ;NO SET UP NEXT LINE
3465  015034  010277  165016                        MOV     R2,@DZTCR       ;SET TCR BIT
3466  015040  032777  000200  164774    4$:         BIT     #RDONE,@DZCSR   ;IS REC DONE = 0 ?
3467  015046  001401                                BEQ     5$              ;IF YES, ALLOW TIME FOR TRDY TO SET
3468  015050  104020                                ERROR   20              ;*REC DONE SHOULD = 0
3469  015052  005005                    5$:         CLR     R5
3470  015054  032777  100000  164760    6$:         BIT     #TRDY,@DZCSR
3471  015062  001004                                BNE     7$
3472  015064  104414                                DELAY
3473  015066  105205                                INCB    R5
3474  015070  001371                                BNE     6$
3475  015072  104003                                ERROR   3               ;*TRDY FAILED TO SET!
3476  015074  112777  000252  164764    7$:         MOVB    #252,@DZTDR     ;LOAD CHARACTER
3477  015102  013705  001372                        MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
3478  015106  105737  001371                        TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
3479  015112  001406                                BEQ     10$             ;IF NOT, SKIP STAGGERED SETUP
3480
3481                                                ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3482
3483  015114  006205                                ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
3484  015116  103402                                BCS     8$              ;IF IT IS SET, GO CLEAR IT
3485  015120  000261                                SEC                     ;IF IT IS CLEAR SET IT HERE
3486  015122  000401                                BR      9$              ;SKIP THE CLEARING
3487  015124  000241                    8$:         CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3488  015126  006105                    9$:         ROL     R5              ;GET THE NEW BIT BACK INTO R5
3489  015130  000305                    10$:        SWAB    R5              ;MOVE THE LINE NUMBER TO THE UPPER BYTE
3490  015132  152705  000252                        BISB    #252,R5         ;ADD CHARACTER
3491  015136  052705  100000                        BIS     #DVALID,R5      ;ADD DATA VALID
3492  015142  005003                                CLR     R3
3493  015144  032777  000200  164670    11$:        BIT     #RDONE,@DZCSR
3494  015152  001004                                BNE     12$
3495  015154  104414                                DELAY
3496  015156  005203                                INC     R3
3497  015160  001371                                BNE     11$
3498  015162  104004                                ERROR   4               ;*RDONE FAILED TO SET!
3499  015164  017704  164656            12$:        MOV     @DZRBUF,R4      ;LOAD THE VALUE ACTUALLY RECEIVED
3500  015170  020405                                CMP     R4,R5           ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
3501  015172  001401                                BEQ     13$             ;IF YES, GO DO THE NEXT LINE
3502  015174  104006                                ERROR   6               ;*NO DATA/CONTENTS DID NOT COMPARE
```

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

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

```
3615                                    ;;*****************************************************************
3616  015616  000004                    TST23:  SCOPE
3617  015620  012737  000023  001122            MOV     #23,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3618  015626  012737  016150  001360            MOV     #TST24,NEXT     ;POINT TO THE START OF THE NEXT TEST
3619  015634  105037  001420                    CLRB    DONFLG          ;INITIALIZE FOR FIRST TEST LOOP
3620  015640  005037  001372                    CLR     SAVLIN          ;ZERO LINE NO. FOR ERROR REPORT
3621  015644  104417                            DCLASM                  ;EXECUTE MASTER CLEAR
3622  015646  013701  001366                    MOV     PAR,R1          ;STORE DEFAULT PARAMETERS
3623  015652  042701  010000                    BIC     #RCVON,R1       ;CLEAR RCVON BIT
3624  015656  012702  000001            1$:     MOV     #1,R2           ;INIT LINE POINTER
3625  015662  010177  164164            2$:     MOV     R1,@DZLPR       ;LOAD LINE PARAMETER REGISTER
3626  015666  005201                            INC     R1              ;SET R1 FOR NEXT LINE
3627  015670  106302                            ASLB    R2              ;SHIFT R2 TO NEXT LINE
3628  015672  103373                            BCC     2$              ;ALL LINES LOADED?
3629  015674  012701  000252                    MOV     #252,R1         ;LOAD TRANSMITTING CHARACTER
3630  015700  013702  001364                    MOV     LINE,R2         ;COPY ACTIVE LINE BITS
3631  015704  010277  164146                    MOV     R2,@DZTCR       ;LOAD TCR BITS
3632  015710  052777  000040  164124            BIS     #MSENAB,@DZCSR  ;SET SCANNER
3633  015716  005005                    3$:     CLR     R5              ;INIT DELAY COUNTER
3634  015720  005777  164116            4$:     TST     @DZCSR          ;TRDY SET?
3635  015724  100404                            BMI     5$              ;IF YES BRANCH
3636  015726  104414                            DELAY                   ;IF NOT THEN WAIT
3637  015730  005205                            INC     R5              ;INCREMENT DELAY COUNTER
3638  015732  001372                            BNE     4$              ;DELAY DONE?
3639  015734  104003                            ERROR   3               ;IF YES TRDY FAILED TO SET
3640  015736  117705  164102            5$:     MOVB    @HDZCSR,R5      ;MOVE LINE NO. INTO R5
3641  015742  012703  000001                    MOV     #1,R3           ;INIT TCR POINTER
3642  015746  042705  177770                    BIC     #^C<7>,R5       ;ISOLATE LINE NO.
3643  015752  001403                            BEQ     21$             ;IF LINE 0 GO TEST TRANSM. FLAG
3644  015754  106303                    20$:    ASLB    R3              ;POINT R3 TO NEXT TCR BIT
3645  015756  005305                            DEC     R5              ;DECREMENT R5 UNTIL R3 POINTS
3646  015760  001375                            BNE     20$             ;TO CORRECT TCR BIT
3647  015762  030302                    21$:    BIT     R3,R2           ;HAS THIS LINE BEEN SERVICED?
3648  015764  001007                            BNE     6$              ;IF NOT GO SEND CHARACTER
3649  015766  140377  164064                    BICB    R3,@DZTCR       ;IF YES CLEAR TCR BIT
3650  015772  001351                            BNE     3$              ;IF MORE LINES SET BRANCH
3651  015774  105737  001420                    TSTB    DONFLG          ;IF ALL LOADED IS THIS SECOND PASS
3652  016000  001040                            BNE     12$             ;IF YES BRANCH TO SECOND PART OF TEST
3653  016002  000404                            BR      7$              ;OTHERWISE CONTINUE WITH FIRST PART
3654  016004  110177  164056            6$:     MOVB    R1,@DZTDR       ;TRANSMIT CHARACTER
3655  016010  040302                            BIC     R3,R2           ;CLEAR FLAG FOR THIS LINE
3656  016012  000741                            BR      3$              ;GO WAIT FOR NEXT LINE
3657  016014  005077  164036            7$:     CLR     @DZTCR          ;CLEAR TCR BITS
3658  016020  005005                            CLR     R5              ;CLEAR DELAY COUNTER
3659  016022  104414                    8$:     DELAY                   ;WAIT FOR LAST CHARACTER
3660  016024  005205                            INC     R5              ;INCREMENT DELAY COUNTER
3661  016026  001375                            BNE     8$              ;IF NOT FINISHED CONTINUE WAITING
3662  016030  105777  164006                    TSTB    @DZCSR          ;RDONE BIT SET?
3663  016034  100003                            BPL     10$             ;IF NO CONTINUE
3664  016036  005037  001372                    CLR     SAVLIN          ;IF YES SET LINE NO. TO ZERO
3665  016042  104020                            ERROR   20              ;AND PRINT ERROR
3666  016044  017704  163776            10$:    MOV     @DZRBUF,R4      ;READ SILO
3667  016050  100007                            BPL     11$             ;IF DVALID IS ZERO BRANCH
3668  016052  000304                            SWAB    R4              ;IF SET THEN
3669  016054  042704  177770                    BIC     #^C<7>,R4       ;ISOLATE LINE NO. IN R4
3670  016060  010437  001372                    MOV     R4,SAVLIN       ;SET SAVLIN FOR ERROR REPORT
```

```
                                        ERROR   17              ;DATA VALID SHOULD NOT BE SET
                                        BR      10$             ;GO READ SILO AGAIN
                                11$:    INCB    DONFLG          ;PREPARE FOR SECOND PART OF TEST
                                        MOV     PAR,R1          ;MOVE DEFAULT PARAMETERS TO R1
                                        BR      1$              ;GO LOAD LPR REGISTER
                                12$:    CLR     R5              ;INIT DELAY COUNTER
                                13$:    DELAY                   ;WAIT FOR LAST CHARACTER
                                        INC     R5              ;TO BE RECEIVED
                                        BNE     13$             ;DELAY FINISHED?
                                        DEVICE.CLR              ;IF YES EXECUTE MASTER CLEAR
                                        NOP
                                        NOP
                                        TSTB    @DZCSR          ;RDONE SET?
                                        BPL     14$             ;IF NOT BRANCH
                                        CLR     SAVLIN          ;IF YES THEN PRINT OUT
                                        ERROR   20              ;REPORT
                                14$:    MOV     @DZRBUF,R4      ;READ SILO
                                        BPL     15$             ;DATA VALID SET?
                                        CLR     SAVLIN          ;IF YES THEN PRINT OUT
                                        ERROR   17              ;ERROR REPORT
                                15$:

                                        ;************************ TEST 24 *********************************
                                        ;*THIS TEST WILL PROVE THAT:
                                        ;* 1) THE TRANSMITTER 'BREAK BIT'' WORKS
                                        ;* 2) THE RECEIVER CAN FLAG 'FRAMING ERRORS''
                                        ;* 3) THE RECEIVER CAN FLAG 'PARITY ERRORS''
                                        ;*ONLY ONE LINE AT A TIME WILL BE EXERCISED.
                                        ;*THIS TEST WILL NOT BE  EXERCISED UNLESS
                                        ;*CONNECTED BY AN H325, H3271, OR H3190 CONNECTOR
                                ;:*  TEST 24
                                ;:*************************************************************************
                                TST24:  SCOPE
                                        MOV     #24,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
                                        MOV     #TST25,NEXT     ;POINT TO THE START OF THE NEXT TEST
                                        MOV     #3$,LOCK        ;SET FOR LOOP
                                        TST     MODE            ;ARE WE RUNNING IN INTERNAL MODE?
                                        BEQ     12$             ;IF SO, SKIP THIS TEST
                                        DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
                                        MOV     PAR,R1          ;PICK UP PARAMETERS
                                        BIS     #ODDPAR!PARITY,R1 ;FORCE ODD PARITY
                                        MOV     #1,R0           ;PICK UP INIT POINTER
                                1$:     BIT     R0,LINE         ;SHOULD THIS LINE BE SET UP ?
                                        BEQ     2$              ;IF NOT,DON'T SET IT UP
                                        MOV     R1,@DZLPR       ;OTHERWISE, SET UP LINE PARAMETERS
                                2$:     INC     R1
                                        ASLB    R0              ;GOT 'EM ALL ?
                                        BCC     1$              ;NO
                                        CLR     SAVLIN          ;CLEAR LINE #
                                        MOV     #1,R2           ;LINE POINTER
                                        BIS     #MSENAB,@DZCSR  ;SET MASTER SCAN ENABLE
                                        MOV     DZRBUF,REGIST   ;SAVE FOR ERRR MESSAGE
                                3$:     BIT     R2,LINE
                                        BEQ     11$
                                        MOV     R2,@DZTCR       ;SET TCR BIT
                                        MOVB    R2,@HDZTDR      ;SET BREAK BIT
```

```
3727  016302  112777  000377  163556  4$:     MOVB    #377,@DZTDR     ;LOAD CHARACTER
3728  016310  013705  001372                  MOV     SAVLIN,R5       ;MAKE EXPECTED DATA
3729  016314  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
3730  016320  001406                          BEQ     7$              ;IF NOT, SKIP STAGGERED SETUP
3731
3732                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3733
3734  016322  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
3735  016324  103402                          BCS     5$              ;IF IT IS SET, GO CLEAR IT
3736  016326  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
3737  016330  000401                          BR      6$              ;SKIP THE CLEARING
3738  016332  000241                  5$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3739  016334  006105                  6$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
3740  016336  000305                  7$:     SWAB    R5              ;PUT LINE NUMBER IN UPPER BYTE
3741  016340  052705  130000                  BIS     #DVALID!PARER!FRMERR,R5 ;ADD EXPECTED
3742  016344  005004                          CLR     R4
3743  016346  032777  000200  163466  8$:     BIT     #RDONE,@DZCSR
3744  016354  001004                          BNE     9$
3745  016356  104414                          DELAY
3746  016360  005204                          INC     R4
3747  016362  001371                          BNE     8$
3748  016364  104004                          ERROR   4               ;*RDONE FAILED TO SET!
3749  016366  017704  163454          9$:     MOV     @DZRBUF,R4      ;ACTUAL
3750  016372  020405                          CMP     R4,R5           ;CMP ACTUAL VS EXPECTED. DO THEY MATCH?
3751  016374  001401                          BEQ     10$             ;IF YES, GO CLEAN UP
3752  016376  104006                          ERROR   6               ;*DATA/CONTENTS FAILED TO COMPARE
3753  016400  105077  163464          10$:    CLRB    @HDZTDR         ;CLEAR BREAK BITS
3754  016404  104401                          SCOP1                   ;LOOP?
3755  016406  005237  001372          11$:    INC     SAVLIN          ;INC LINE #
3756  016412  040277  163440                  BIC     R2,@DZTCR       ;CLEAR TCR BIT
3757  016416  106302                          ASLB    R2
3758  016420  103321                          BCC     3$
3759  016422  005037  001362          12$:    CLR     LOCK            ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST
3760                                          ;************************ TEST 25 *********************************
3761                                          ;* THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
3762                                          ;*WHILE THE PROCESSOR STATUS IS SET EXACTLY
3763                                          ;*TO WHAT THE DZ11 PRIORITY IS SET TO.
3764                                          ;*DEFAULT PRIORITY IS AT  5 (240).
3765                                  ;:*  TEST 25
3766                                  ;;*****************************************************************
3767  016426  000004                  TST25:  SCOPE
3768  016430  012737  000025  001122          MOV     #25,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
3769  016436  012737  016736  001360          MOV     #TST26,NEXT     ;POINT TO THE START OF THE NEXT TEST
3770  016444  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
3771  016446  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
3772  016452  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
3773  016456  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
3774  016462  001402                          BEQ     2$              ;NO
3775  016464  010177  163362                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
3776  016470  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
3777  016472  106302                          ASLB    R2              ;GOT 'EM ALL ?
3778  016474  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
3779  016476  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
3780  016502  106437  027060                  MTPS    @#DZPRT         ;SET CPU STATUS TO DZ11 PRIO,
3781  016506  113777  001364  163342          MOVB    LINE,@DZTCR     ;ENABLE THE VALID LINES
3782  016514                          3$:
```

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

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

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

```
3951  017470  005204                          INC     R4              ;
3952  017472  001372                          BNE     99$
3953  017474  104003                          ERROR   3               ;TRDY FAILED TO SET
3954  017476  105077  162364          100$:   CLRB    @DZTDR
3955  017502  005004                          CLR     R4
3956  017504  032777  000200  162330  6$:     BIT     #RDONE,@DZCSR
3957  017512  001004                          BNE     7$
3958  017514  104414                          DELAY
3959  017516  005204                          INC     R4
3960  017520  001371                          BNE     6$
3961  017522  104004                          ERROR   4               ;*RDONE FAILED TO SET!
3962  017524  005777  162312          7$:     TST     @DZCSR          ;TRANS DONE BIT = 1 ?
3963  017530  100401                          BMI     .+4             ;YES
3964  017532  104003                          ERROR   3               ;*NO   TRANS DONE FAILED TO SET
3965                                          ;NOW THAT BOTH TRANSMITTER AND RECEIVER DONE BIT =1
3966                                          ;SET INTERRUPT ENABLES AND WATCH THE FUR FLY
3967  017534  052777  040000  162300        BIS     #TIE,@DZCSR
3968  017542  052777  000100  162272        BIS     #RIE,@DZCSR
3969  017550  106427  000000                MTPS    #0
3970  017554  000240                        NOP
3971  017556  000240                        NOP
3972  017560  104007                        ERROR   7               ;*TRANSMITTER FAILED TO INTERRUPT
3973  017562  104011                        ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
3974                                                                ;CHECK BR LEVEL
3975  017564  000137  017664                JMP     13$     ;GET OUT
3976
3977                                        ;RECEIVER INTERRUPT ROUTINE
3978  017570  017704  162252          8$:   MOV     @DZRBUF,R4              ;ACTUAL
3979  017574  010403                        MOV     R4,R3
3980  017576  000303                        SWAB    R3
3981  017600  042703  177770              BIC     #^C<7>,R3       ;STRIP JUNK
3982  017604  105737  001371              TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
3983  017610  001406                        BEQ     11$             ;IF NOT, SKIP STAGGERED SETUP
3984
3985                                        ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
3986
3987  017612  006203                        ASR     R3              ;GET THE LAST BIT INTO THE CARRY BIT
3988  017614  103402                        BCS     9$              ;IF IT IS SET, GO CLEAR IT
3989  017616  000261                        SEC                     ;IF IT IS CLEAR SET IT HERE
3990  017620  000401                        BR      10$             ;SKIP THE CLEARING
3991  017622  000241            9$:         CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
3992  017624  006103            10$:        ROL     R3              ;GET THE NEW BIT BACK INTO R3
3993  017626  020337  001372    11$:        CMP     R3,SAVLIN       ;IS THIS A VALID LINE
3994  017632  001401                        BEQ     .+4             ;YES
3995  017634  104015                        ERROR   15              ;*INVALID LINE
3996  017636  042704  177400              BIC     #^C<377>,R4     ;STRIP JUNK
3997  017642  120504                        CMPB    R5,R4           ;DATA COMPARE ?
3998  017644  001401                        BEQ     .+4             ;YES
3999  017646  104005                        ERROR   5               ;*DATA DOES NOT COMPARE
4000  017650  040277  162202              BIC     R2,@DZTCR       ;CLEAR TCR BIT
4001  017654  022626                        POP2SP                  ;REMOVE HE INTERRUPT VECTOR FROM THE STACK
4002  017656  000402                        BR      13$             ;GO GET OUT OF INTERRUPT MODE
4003                                        ;TRANSMITTER INTERRUPT SVC ROUTINE
4004  017660  104011            12$:        ERROR   11              ;THE RECEIVER INTERRUPT FAILED
4005                                                                ;TO OVERRIDE THE TRANSMITTER
4006  017662  022626                        POP2SP                  ;REMOVE THE INTERRUPT VECTOR FROM THE STACK
```

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

```
                                        ;*EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
                                        ;*SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
                                        ;*CHAR PULLED OUT OUT THE SILO.
                                        ;*USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
                                        ;*ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
                                        ;*USED TO SCOPE SILO ALARM PULSES, ETC.
                                ;:*  TEST 31
                                ;;*********************************************************************
020104  000004                  TST31:  SCOPE
020106  012737  000031  001122          MOV     #31,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
020114  012737  020632  001360          MOV     #TST32,NEXT     ;POINT TO THE START OF THE NEXT TEST
020122  012737  020536  001362          MOV     #18$,LOCK       ;SET FOR LOOP
020130  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
020132  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
020136  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
020142  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
020146  001402                          BEQ     2$              ;NO
020150  010177  161676                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
020154  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
020156  106302                          ASLB    R2              ;GOT 'EM ALL ?
020160  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
020162  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
020166  012700  001422                  MOV     #TDO,R0         ;POINT TO THE DATA AREA
020172  005020                          CLR     (R0)+           ;CLEAR A DATA WORD
020174  022700  001462                  CMP     #STOP,R0        ;FINISHED ?
020200  001374                          BNE     .-6             ;NO
020202  005000                          CLR     R0              ;CLEAR OFFSET
020204  012702  000001                  MOV     #1,R2           ;LINE POINTER
020210  052777  010040  161624          BIS     #MSENAB!SILOEN,@DZCSR   ;START SCANNER & SET SILO ENABLE
020216  030237  001364          3$:     BIT     R2,LINE         ;VALID LINE?
020222  001002                          BNE     .+6             ;YES
020224  000137  020520                  JMP     22$             ;TRY NEXT LINE
020230  013700  001372                  MOV     SAVLIN,R0       ;MAKE OFFSET
020234  006300                          ASL     R0              ;MAKE POWER OF TWO
020236  010277  161614                  MOV     R2,@DZTCR       ;SET TCR BIT
020242  105777  161574          4$:     TSTB    @DZCSR          ;REC DONE = 1 ?
020246  100001                          BPL     .+4
020250  104020                          ERROR   20              ;REC DONE SHOULD NOT = 1
020252  005003                          CLR     R3              ;SET CHARACTER COUNT
020254  005004                  5$:     CLR     R4
020256  032777  100000  161556  6$:     BIT     #TRDY,@DZCSR
020264  001004                          BNE     7$
020266  104414                          DELAY
020270  105204                          INCB    R4
020272  001371                          BNE     6$
020274  104003                          ERROR   3               ;*TRDY FAILED TO SET
020276  116077  001422  161562  7$:     MOVB    TDO(R0),@DZTDR  ;LOAD A CHARACTER
020304  005260  001422                  INC     TDO(R0)         ;SET UP NEXT CHARACTER
020310  020327  000017                  CMP     R3,#15.         ;16 CHARACTERS ?
020314  103006                          BHIS    8$
020316  032777  020000  161516          BIT     #SILOAL,@DZCSR  ;SILO ALARM = 0 ?
020324  001401                          BEQ     .+4             ;YES
020326  104013                          ERROR   13              ;*SILO ALARM SHOULD NOT = 1
                                                                ;UNTIL 16. DATA CHARACTERS
020330  000411                          BR      10$
020332  005004                  8$:     CLR     R4
```

```
 4119  020334  032777  020000  161500  9$:     BIT     #SILOAL,@DZCSR
 4120  020342  001004                          BNE     10$
 4121  020344  104414                          DELAY
 4122  020346  005204                          INC     R4
 4123  020350  001371                          BNE     9$
 4124  020352  104014                          ERROR   14              ;*SILO ALARM FAILED TO SET!
 4125                                                                  ;SILO ALARM SHOULD =1 AFTER 16.
 4126                                                                  ;DATA CHARACTERS
 4127  020354  005203                  10$:    INC     R3              ;INC CHAR COUNT
 4128  020356  022703  000102                  CMP     #66.,R3         ;FINISHED SENDING CHARACTERS ?
 4129  020362  001334                          BNE     5$              ;NO
 4130  020364  005004                          CLR     R4
 4131  020366  104414                          DELAY
 4132  020370  105204                          INCB    R4
 4133  020372  001375                          BNE     .-4
 4134                                          ;NOW LETS READ THE SILO
 4135  020374  013705  001372                  MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
 4136  020400  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
 4137  020404  001406                          BEQ     13$             ;IF NOT, SKIP STAGGERED SETUP
 4138
 4139                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER
 4140
 4141  020406  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
 4142  020410  103402                          BCS     11$             ;IF IT IS SET, GO CLEAR IT
 4143  020412  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
 4144  020414  000401                          BR      12$             ;SKIP THE CLEARING
 4145  020416  000241                  11$:    CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
 4146  020420  006105                  12$:    ROL     R5              ;GET THE NEW BIT BACK INTO R5
 4147  020422  000305                  13$:    SWAB    R5              ;PUT IN UPPER BYTE
 4148  020424  052705  100000                  BIS     #DVALID,R5      ;ADD DATA VALID
 4149  020430  017704  161412          14$:    MOV     @DZRBUF,R4      ;ACTUAL
 4150  020434  020405                          CMP     R4,R5           ;ACTUAL VS. EXPECTED
 4151  020436  001401                          BEQ     15$             ;YES
 4152  020440  104006                          ERROR   6               ;*DATA/CONTENTS DID NOT COMPARE
 4153  020442  032777  020000  161372  15$:    BIT     #SILOAL,@DZCSR  ;SILO ALARM= 0 ?
 4154  020450  001401                          BEQ     16$             ;YES
 4155  020452  104016                          ERROR   16              ;READING DZRBUF DID NOT CLEAR SILO ALARM
 4156  020454  005205                  16$:    INC     R5              ;UP CHARACTER
 4157  020456  120527  000077                  CMPB    R5,#63.         ;LAST SILO CHAR ?....64TH CHAR
 4158  020462  101762                          BLOS    14$
 4159  020464  005205                          INC     R5              ;ADD 1 MORE FOR THE CLOBBERED CHAR
 4160  020466  052705  040000                  BIS     #OVRRUN,R5      ;ADD OVERRUN TO EXPECTED
 4161  020472  120527  000101                  CMPB    R5,#65.         ;LAST CHARACTER ?
 4162  020476  001754                          BEQ     14$
 4163  020500  017704  161342                  MOV     @DZRBUF,R4      ;FOR GOOD MEASURE
 4164  020504  005704                          TST     R4              ;DATA VALID SHOULD = 0
 4165  020506  100001                          BPL     17$             ;YES
 4166  020510  104017                          ERROR   17              ;DATA VALID SHOULD = 0
 4167  020512  040277  161340          17$:    BIC     R2,@DZTCR       ;CLR TCR BIT
 4168  020516  104401                          SCOP1                   ;LOOP?
 4169  020520  005237  001372          22$:    INC     SAVLIN          ;INC EXPECTED LINE
 4170  020524  106302                          ASLB    R2              ;NEXT LINE
 4171  020526  103402                          BCS     .+6             ;NO
 4172  020530  000137  020216                  JMP     3$              ;YES
 4173  020534  104400                          ADVANCE                 ;GO TO NEXT TEST
 4174
```

```
                                        ;TIGHT SCOPE LOOP FOR THIS TEST. SENDS 20. CHARACTERS
                                        ;ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
                                        ;USED TO SCOPE SILO ALARM PULSES, ETC.

020536  052777  010040  161276  18$:    BIS     #MSENAB!SILOEN,@DZCSR   ;SETUP DEVICE
020544  012777  020622  161324          MOV     #20$,@DZTIV             ;SETUP TRANSMITTER VECTOR
020552  012737  000024  001216          MOV     #20.,$TMP0              ;TEMPORARY COUNT OF CHARACTER BURST
020560  050277  161272                  BIS     R2,@DZTCR               ;ENABLE LINE
020564  052777  040000  161250          BIS     #TIE,@DZCSR             ;ENABLE INTERRUPTS
020572  106427  000000                  MTPS    #0                      ;LOWER PRIORITY
020576  000001                  19$:    WAIT                            ;ALLOW INTERRUPTS
020600  005337  001216                  DEC     $TMP0                   ;REDUCE COUNT. ALL CHARACTERS SENT?
020604  001374                          BNE     19$                     ;IF NO, WAIT FOR MORE
020606  042777  050040  161226          BIC     #SILOEN!MSENAB!TIE,@DZCSR ;RESET SILO COUNTER, CLEAR STROBE
020614  104401                          SCOP1                           ;LOOP AGAIN?
020616  000137  020512                  JMP     17$                     ;IF NOT, RETURN TO WHERE YOU LEFT OFF
020622  112777  000252  161236  20$:    MOVB    #252,@DZTDR             ;SEND A CHARACTER
020630  000002                          RTI                             ;ALLOW MORE CHARACTERS TO COME
                                        ;************************* TEST 32 *******************************
                                        ;*THIS TEST THAT "SILO ENABLE" WILL INHIBIT
                                        ;*RECEIVER INTERRUPTS AND THAT ON THE
                                        ;*16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
                                        ;*INTERRUPT WITH "RIE" SET.
                                        ;*THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
                                ;:*  TEST 32
                                ;;*************************************************************
020632  000004                  TST32:  SCOPE
020634  012737  000032  001122          MOV     #32,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
020642  012737  021214  001360          MOV     #TST33,NEXT             ;POINT TO THE START OF THE NEXT TEST
020650  012737  020736  001362          MOV     #3$,LOCK                ;SET FOR LOOP
020656  104417                          DCLASM                          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
020660  013701  001366                  MOV     PAR,R1                  ;PICK UP PARAMETERS
020664  012702  000001                  MOV     #1,R2                   ;PICK UP INIT POINTER
020670  030237  001364          1$:     BIT     R2,LINE                 ;SHOULD THIS LINE BE SET UP ?
020674  001402                          BEQ     2$                      ;NO
020676  010177  161150                  MOV     R1,@DZLPR               ;SET UP LINE PARAMETERS
020702  005201                  2$:     INC     R1                      ;POSITION POINTER TO THE NEXT LINE
020704  106302                          ASLB    R2                      ;GOT 'EM ALL ?
020706  103370                          BCC     1$                      ;IF NO, GO SET UP THE NEXT LINE
020710  005037  001372                  CLR     SAVLIN                  ;CLEAR LINE # INDICATOR
020714  012700  001422                  MOV     #TD0,R0                 ;POINT TO THE DATA AREA
020720  005020                          CLR     (R0)+                   ;CLEAR A DATA WORD
020722  022700  001462                  CMP     #STOP,R0                ;FINISHED ?
020726  001374                          BNE     .-6                     ;NO
020730  005000                          CLR     R0                      ;CLEAR OFFSET
020732  012702  000001                  MOV     #1,R2                   ;LINE POINTER
020736  012777  021156  161126  3$:     MOV     #11$,@DZRIV             ;SET FOR UNEXPECTED INTER.
020744  012777  000340  161122          MOV     #PR7,@DZRIS             ;SET PRIO.
020752  052777  010140  161062          BIS     #MSENAB!SILOEN!RIE,@DZCSR
                                                                        ;START SCANNER & SET SILO ENABLE
020760  030237  001364                  BIT     R2,LINE                 ;VALID LINE?
020764  001002                          BNE     .+6                     ;YES
020766  000137  021174                  JMP     22$                     ;TRY NEXT LINE
020772  005777  161050                  TST     @DZRBUF                 ;EMPTY THE SILO
020776  100775                          BMI     .-4                     ;BR IF DATA VALID IS SET!
021000  106427  000000                  MTPS    #0                      ;SET PROCESSOR PRIORITY TO 0
```

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

```
                                        ;************************* TEST 33 *******************************
                                        ;*THIS TEST RUNS ALL LINES FULL BORE
                                        ;*BASED UPON QUALIFIED LINES
                                        ;*..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
                                        ;*TRANSMITTER
                                ;:*  TEST 33
                                ;;*************************************************************************
021214  000004                  TST33:  SCOPE
021216  012737  000033  001122          MOV     #33,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
021224  012737  022022  001360          MOV     #TST34,NEXT     ;POINT TO THE START OF THE NEXT TEST
021232  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
021234  013737  001364  022020          MOV     LINE,RXTCR      ;SET IMAGE OF TCR BITS
021242  013701  001366          RSTART: MOV     PAR,R1          ;PICK UP PARAMETER
021246  012700  000001                  MOV     #1,R0           ;;PICK UP INIT POINTER
021252  030037  001364          INIT:   BIT     R0,LINE         ;SHOULD THIS LINE BE SET UP
021256  001402                          BEQ     1$              ;NO
021260  010177  160566                  MOV     R1,@DZLPR       ;SET UP LINE PARAM REGISTER
021264  005201                  1$:     INC     R1
021266  106300                          ASLB    R0              ;GOT 'EM ALL ?
021270  103370                          BCC     INIT            ;NO
021272  012700  001422                  MOV     #TD0,R0         ;CLEAR TRANS DATA POINTER & REC POINTERS
021276  005020                  INIT1:  CLR     (R0)+
021300  022700  001462                  CMP     #STOP,R0                ;FINISHED ?
021304  001374                          BNE     INIT1           ;NO, CONTINUE CLEARING
021306  012777  021542  160556          MOV     #RXSVC,@DZRIV   ;SET UP REC INTR VECTOR
021314  012777  000340  160552          MOV     #PR7,@DZRIS     ;STATUS
021322  012777  021444  160546          MOV     #TXSVC,@DZTIV   ;SET UP TRANS INTR VECTOR
021330  012777  000340  160542          MOV     #PR7,@DZTIS     ;STATUS
021336  052777  000100  160476          BIS     #RIE,@DZCSR             ;SET REC INTR ENABLE
021344  052777  040000  160470          BIS     #TIE,@DZCSR             ;SET TRANS INTR ENABLE
021352  052777  000040  160462          BIS     #MSENAB,@DZCSR  ;SET MASTER SCAN ENABLE
021360  113777  001364  160470          MOVB    LINE,@DZTCR     ;SET TCR BITS...UP UP AND AWAY !
021366  106437  027062                  MTPS    @#LESS1                 ;ALLOW INTERRUPTS
                                
                                
021372  005037  021442          SNAP:   CLR     66$
021376  013727  006722          67$:    MOV     DLYCNT,(PC)+    ;SET FOR DELAY
021402  000000                  68$:    0
021404  005337  021402                  DEC     68$
021410  001375                          BNE     .-4
021412  105737  022020                  TSTB    RXTCR           ;WAIT FOR ALL RECIEVERS TO FINISH
021416  001002                          BNE     3$              ;
021420  000137  021720                  JMP     OUT
021424  005237  021442          3$:     INC     66$
021430  001362                          BNE     67$
021432  104007                          ERROR   7               ;*TRANSMITTER FAILED TO INTERRUPT
021434  104011                          ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
021436  000137  021772                  JMP     FINI
021442  000000                  66$:    0
                                
                                ;TRANS INTR SVC ROUTINE
021444  005777  160372          TXSVC:  TST     @DZCSR          ;TRANS INTR ?
021450  100401                          BMI     .+4
021452  104003                          ERROR   3               ;*TRANSMITTER FAILED
021454  117703  160364                  MOVB    @HDZCSR,R3      ;SAVE IT
                                ;NOW TEST FOR LINE # ETC
```

```
 4334  021460  042703  177770                BIC     #^C<7>,R3       ;STRIP JUNK
 4335  021464  010304                        MOV     R3,R4           ;SAVE
 4336  021466  010337  001372                MOV     R3,SAVLIN       ;ADJUST LOCATION FOR ERROR PRINTOUT
 4337  021472  012702  000001                MOV     #1,R2           ;SET UP POSITION POINTER
 4338  021476  105303              3$:       DECB    R3              ;IS IT THIS LINE ?
 4339  021500  100402                        BMI     4$              ;YES
 4340  021502  006302                        ASL     R2              ;UP THE LINE #
 4341  021504  000774                        BR      3$              ;GO 'ROUND AGAIN
 4342  021506  030237  001364        4$:     BIT     R2,LINE         ;VALID LINE?
 4343  021512  001001                        BNE     .+4             ;YES
 4344  021514  104010                        ERROR   10              ;NO,INVALID LINE!!!!
 4345  021516  006304                        ASL     R4              ;MAKE POWER OF 2
 4346  021520  116477  001422  160340        MOVB    TDO(R4),@DZTDR  ;LOAD CHARACTER
 4347  021526  105264  001422                INCB    TDO(R4)         ;SET UP NEXT CHARACTER
 4348  021532  001002                        BNE     5$              ;LAST CHARACTER ?
 4349  021534  040277  160316                BIC     R2,@DZTCR       ;YES ,CLEAR TCR BIT
 4350  021540  000002              5$:       RTI
 4351
 4352
 4353                                  ;REC INTR SVC ROUTINE
 4354  021542  105777  160274        RXSVC:  TSTB    @DZCSR          ;REC DONE ?
 4355  021546  100401                        BMI     .+4             ;YES
 4356  021550  104004                        ERROR   4               ;FALSE INTERRUPT
 4357  021552  017704  160270                MOV     @DZRBUF,R4      ;SAVE IT
 4358  021556  010403                        MOV     R4,R3
 4359  021560  000303                        SWAB    R3
 4360  021562  042703  177770                BIC     #^C<7>,R3       ;STRIP JUNK
 4361  021566  010337  001372                MOV     R3,SAVLIN       ;SAVE LINE NUMBER
 4362  021572  032777  020000  160242        BIT     #SILOAL,@DZCSR  ;SILO ALARM?
 4363  021600  001401                        BEQ     .+4             ;NO
 4364  021602  104000                        ERROR                   ;SILO ALARM SHOULD NOT =1
 4365  021604  005704                        TST     R4              ;DATA VALID SET?
 4366  021606  100401                        BMI     .+4             ;YES
 4367  021610  104023                        ERROR   23              ;YOU LOSE  ...DATA VALID WAS'NT SET
 4368  021612  032704  070000                BIT     #OVRRUN!FRMERR!PARER,R4
 4369  021616  001401                        BEQ     .+4
 4370  021620  104000                        ERROR                   ;RECEIVER ERROR FLAG/S WERE SET
 4371  021622  012702  000001                MOV     #1,R2           ;SET UP POSITION POINTER
 4372  021626  105303              5$:       DECB    R3
 4373  021630  100402                        BMI     6$
 4374  021632  006302                        ASL     R2              ;RE POSITION POINTER
 4375  021634  000774                        BR      5$              ;GO 'ROUND AGAIN
 4376  021636  030237  001364        6$:     BIT     R2,LINE         ;LINE VALID ?
 4377  021642  001001                        BNE     .+4             ;YES
 4378  021644  104011                        ERROR   11              ;INVALID LINE #
 4379  021646  013703  001372                MOV     SAVLIN,R3       ;GET THE LINE NUMBER AGAIN
 4380  021652  006303                        ASL     R3              ;USE R3 AS A POINTER IN THE DATA TABLE
 4381  021654  126304  001442                CMPB    TRO(R3),R4      ;DOES THE DATA CHARACTER COMPARE ?
 4382  021660  001405                        BEQ     2$              ;YES
 4383  021662  016305  001442                MOV     TRO(R3),R5      ;SAVE EXPECTED
 4384  021666  042704  177400                BIC     #^C<377>,R4     ;CLEAR JUNK
 4385                                                                ;R2 = LINE # BY BIT POSITION
 4386                                                                ;R4 = ACTUAL DATA
 4387                                                                ;R5 = EXPECTED DATA
 4388  021672  104005                        ERROR   5               ;*NO, DATA DOES NOT COMPARE
 4389  021674  005263  001442        2$:     INC     TRO(R3)         ;SET UP FOR NEXT CHARACTER
```

```
4390  021700  105763  001442                    TSTB    TRO(R3) ;ALL CHARS DONE?
4391  021704  001002                            BNE     .+6
4392  021706  040237  022020                    BIC     R2,RXTCR        ;ZERO LINE DONE INDICATOR.
4393  021712  012716  021372                    MOV     #SNAP,(SP)      ;RESET THE BACKGROUND TIMING LOOP
4394  021716  000002                            RTI
4395
4396
4397                                            ;FINISH UP ROUTINE
4398  021720  106427  000340            OUT:    MTPS    #PR7                    ;STOP ALL INTERRUPTS
4399  021724  104413                            DEVICE.CLR              ;CLEAR ALL INTERRUPTS AWAY
4400  021726  005003                            CLR     R3
4401  021730  005037  001372                    CLR     SAVLIN
4402  021734  012702  000001                    MOV     #1,R2
4403  021740  030237  001364            1$:     BIT     R2,LINE         ;VALID LINE ?
4404  021744  001405                            BEQ     2$              ;NO
4405  021746  022763  000400  001442            CMP     #400,TRO(R3)    ;RECEIVED A BINARY COUNT PATTERN ?
4406  021754  001401                            BEQ     .+4             ;YES
4407  021756  104027                            ERROR   27              ;THE LINE FAILED TO RECEIVE A FULL
4408                                                                    ;BINARY COUNT PATTERN
4409  021760  005237  001372            2$:     INC     SAVLIN          ;SET UP FOR NEXT LINE
4410  021764  005723                            TST     (R3)+           ;ADD 2
4411  021766  106302                            ASLB    R2              ;SET UP NEXT LINE POINTER
4412  021770  103363                            BCC     1$              ;FINISHED ?
4413  021772                            FINI:
4414  021772  013777  002074  160072            MOV     DZRIS,@DZRIV    ;RESTORE TRAPCATCHER
4415  022000  005077  160070                    CLR     @DZRIS
4416  022004  013777  002100  160064            MOV     DZTIS,@DZTIV
4417  022012  005077  160062                    CLR     @DZTIS
4418  022016  104400                            ADVANCE                 ;GO TO THE NEXT TEST
4419  022020  000000                    RXTCR:  0                       ;RX IMAGE OF TCR BITS
4420
4421
4422                                            ;************************* TEST 34 *******************************
4423                                            ;*DZ11 RELATIVE TIMING TEST.
4424                                            ;*EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
4425                                            ;*AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
4426                                            ;*WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
4427                                            ;*DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
4428                                            ;*THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
4429                                            ;* AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
4430                                            ;*PARAMETERS ARE:
4431                                            ;*  EIGHT BITS/PER/CHAR - TWO STOP BITS AT
4432                                            ;*       50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
4433                                            ;*       2400, 3600, 4800, 7200, 9600 BAUD.
4434                                            ;*  19.2 K BAUD - TWO STOP BITS AT
4435                                            ;*       SEVEN, SIX, FIVE BITS/PER/CHAR.
4436                                            ;*AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
4437                                            ;*THE NEXT SELECTED LINE IS THE TESTED.
4438                                    ;::*  TEST 34
4439                                    ;;*****************************************************************
4440  022022  000004                    TST34:  SCOPE
4441  022024  012737  000034  001122            MOV     #34,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
4442  022032  012737  000002  001226            MOV     #2,$TIMES
4443  022040  012737  022532  001360            MOV     #TST35,NEXT     ;POINT TO THE START OF THE NEXT TEST
4444  022046  012737  022172  001362            MOV     #3$,LOCK        ;SET FOR LOOP
4445  022054  005037  023766                    CLR     OFFSET          ;RESET THIS VARIABLE
```

```
4446  022060  005037  001372                    CLR     SAVLIN          ;RESET LINE NUMBER INDICATOR
4447  022064  005037  001374                    CLR     XMTLIN          ;USE THIS WORD TO TELL WHAT LINE TRANSMITTED
4448  022070  012737  000001  001216            MOV     #1,$TMP0        ;USE $TMP0 AS A BIT POINTER
4449  022076  012737  010070  022530            MOV     #RCVON!S50!EIGHT!TWOSTOP,7$ ;BUILD TEMPORARY PARAMETERS
4450  022104  033737  001216  001364  1$:       BIT     $TMP0,LINE      ;IS THIS LINE ACTIVE?
4451  022112  001027                            BNE     3$              ;IF SO, GO GET STARTED
4452  022114  012737  010070  022530  2$:       MOV     #RCVON!S50!EIGHT!TWOSTOP,7$ ;LOAD PARAMETERS TEMPORARILY
4453  022122  012700  001422                    MOV     #TD0,R0         ;POINT TO THE DATA AREA
4454  022126  005020                            CLR     (R0)+           ;CLEAR A DATA WORD
4455  022130  022700  001462                    CMP     #STOP,R0        ;FINISHED ?
4456  022134  001374                            BNE     .-6             ;NO
4457  022136  005237  001374                    INC     XMTLIN          ;POINT TO THE NEXT LINE TO TRANSMIT
4458  022142  042737  000007  022530            BIC     #7,7$           ;MAKE SURE TEMPORARY PARAMETERS POINT TO 0
4459  022150  053737  001374  022530            BIS     XMTLIN,7$       ;ADD DESIRED LINE NUMBER
4460  022156  005037  023766                    CLR     OFFSET
4461  022162  106337  001216                    ASLB    $TMP0           ;POINT TO THE NEXT LINE
4462  022166  103346                            BCC     1$              ;PROCESS THE NEXT LINE
4463  022170  104400                            ADVANCE                 ;TEST TO SEE IF THIS TEST GETS REPEATED
4464  022172                          3$:
4465  022172  104417                            DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4466  022174  042737  010000  022530            BIC     #RCVON,7$       ;ZERO PARAMTERS FOR TX LINE
4467  022202  013777  022530  157642            MOV     7$,@DZLPR       ;LOAD PARAMTERS FOR TX
4468  022210  005737  001370                    TST     MODE            ;STAGGERED?
4469  022214  100011                            BPL     100$            ;BR IF NO
4470  022216  000241                            CLC                     ;SET UP LINE
4471  022220  006037  022530                    ROR     7$              ;
4472  022224  103002                            BCC     98$             ;BR IF LINE WAS EVEN
4473  022226  000241                            CLC                     ;PREPARE TO MKE LINE EVEN
4474  022230  000401                            BR      99$             ;CONTINUE
4475  022232  000261                  98$:      SEC                     ;PREPARE TO MAKE LINE ODD
4476  022234  006137  022530          99$:      ROL     7$              ;SET ALTERED LINE
4477  022240  052737  010000  022530  100$:     BIS     #RCVON,7$       ;SET RX ON
4478  022246  013777  022530  157576            MOV     7$,@DZLPR       ;LOAD RX PARAMETERS
4479  022254  013737  022530  001372            MOV     7$,SAVLIN       ;ADJUST LOCATION FOR ERROR PRINTOUT
4480  022262  042737  177770  001372            BIC     #^C<7>,SAVLIN   ;STRIP JUNK
4481  022270  042737  000007  022530            BIC     #7,7$           ;CLEAR OLD LINE #
4482  022276  053737  001374  022530            BIS     XMTLIN,7$       ;SET LINE UP AGAIN
4483  022304  013737  022530  001400            MOV     7$,REGIST       ;SAVE PARAMETERS FOR PRINTOUT
4484  022312  012700  001422                    MOV     #TD0,R0         ;POINT TO THE DATA AREA
4485  022316  005020                            CLR     (R0)+           ;CLEAR A DATA WORD
4486  022320  022700  001462                    CMP     #STOP,R0        ;FINISHED ?
4487  022324  001374                            BNE     .-6             ;NO
4488  022326  005002                            CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
4489  022330  005003                            CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
4490  022332  005037  001220                    CLR     $TMP1           ;INITIALIZE THE TIMER
4491  022336  005037  001224                    CLR     $TMP3           ;INITIALIZE THESE BITS ALSO
4492  022342  012737  000020  001376            MOV     #20,XMTCNT      ;SET HOW MANY CHARACTERS TO TRANSMIT
4493  022350  012777  023410  157520            MOV     #XMTSRV,@DZTIV
4494  022356  012777  023554  157506            MOV     #RXISR1,@DZRIV
4495  022364  013777  027060  157502            MOV     DZPRT,@DZRIS
4496  022372  013777  027060  157500            MOV     DZPRT,@DZTIS
4497  022400  113777  001216  157450            MOVB    $TMP0,@DZTCR    ;START THE VALID LINE
4498  022406  052777  040140  157426            BIS     #TIE!RIE!MSENAB,@DZCSR
4499  022414  106427  000000                    MTPS    #0              ;LOWER THE PRIORITY TO ALLOW INTERRUPTS
4500  022420  032777  000100  157414  4$:       BIT     #RIE,@DZCSR     ;IS ROUTINE DONE?
4501  022426  001407                            BEQ     5$              ;WHEN ALL IS DONE RX IE IS CLEARED IN ISR.
```

```
4502  022430  005237  001220                    INC     $TMP1           ;COUNT TIME
4503  022434  001371                            BNE     4$              ;CONTINUE TEST
4504  022436  105237  001224                    INCB    $TMP3           ;DOUBLE COUNT
4505  022442  001366                            BNE     4$              ;CONTINUE TEST
4506  022444  104011                            ERROR   11              ;INTERRUPTS NOT FINISHED
4507  022446  004737  007360          5$:       JSR     PC,SERV.G       ;<^G>?
4508  022452  104401                            SCOP1                   ;LOOP?
4509  022454  062737  000002  023766            ADD     #2,OFFSET
4510  022462  013700  022530                    MOV     7$,R0
4511  022466  042700  170377                    BIC     #^C<17*400>,R0
4512  022472  022700  007400                    CMP     #<17*400>,R0
4513  022476  001010                            BNE     6$
4514  022500  032737  000030  022530            BIT     #BIT4+BIT3,7$
4515  022506  001602                            BEQ     2$
4516  022510  162737  000010  022530            SUB     #BIT3,7$
4517  022516  000625                            BR      3$
4518  022520  062737  000400  022530  6$:       ADD     #400,7$
4519  022526  000621                            BR      3$
4520  022530  000000                    7$:     0
4521                                            ;************************** TEST 35 ********************************
4522                                            ;* THIS TEST VERIFIES THAT EVEN PARITY WORKS
4523                                            ;* FOR ALL ODD LINES SELECTED AND THAT ODD PARITY WORKS FOR ALL
4524                                            ;* EVEN LINES SELECTED.
4525                                            ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
4526                                            ;*THAT 'PE' (PARITY ERROR) CAN BE FLAGGED BY
4527                                            ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
4528                                            ;*YOU ARE IN 'STAGGERED' MODE.
4529                                            ;*40(8) CHARS ARE USED FOR THIS TEST.
4530                                            ;*ALL SELECTED LINES WILL BE ENABLED
4531                                            ;*AT THE SAME TIME!
4532                                    ;:*  TEST 35
4533                                    ;;*************************************************************************
4534  022532  000004                    TST35:  SCOPE
4535  022534  012737  000035  001122            MOV     #35,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
4536  022542  012737  022772  001360            MOV     #TST36,NEXT     ;POINT TO THE START OF THE NEXT TEST
4537  022550  005737  001370                    TST     MODE            ;IS THIS STAGGERED MODE?
4538  022554  100105                            BPL     6$              ;IF NOT, DON'T DO THIS TEST
4539  022556  104417                            DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
4540  022560  013701  001366                    MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
4541  022564  042701  000200                    BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
4542  022570  052701  000100                    BIS     #PARITY,R1      ;MAKE SURE PARITY IS TURNED ON
4543  022574  012702  000001                    MOV     #1,R2           ;USE R2 AS A LINE POINTER
4544  022600  032737  001364          1$:       BIT     R2,LINE         ;IS THIS A VALID LINE?
4545  022604  001411                            BEQ     3$              ;IF NOT, SKIP TO THE NEXT LINE
4546  022606  032701  000001                    BIT     #BIT0,R1        ;IS THIS LINE AN ODD LINE?
4547  022612  001002                            BNE     2$              ;IF IT'S ODD, USE EVEN PARITY
4548  022614  052701  000200                    BIS     #ODDPAR,R1      ;IF IT'S EVEN, USE ODD PARITY
4549  022620  010177  157226          2$:       MOV     R1,@DZLPR       ;LOAD THE LINE PARAMETER REGISTER
4550  022624  042701  000200                    BIC     #ODDPAR,R1      ;SET UP THE NEXT PARITY TO EVEN
4551  022630  005201                    3$:     INC     R1              ;POINT TO THE NEXT LINE
4552  022632  106302                            ASLB    R2              ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
4553  022634  103361                            BCC     1$              ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
4554  022636  005037  001372                    CLR     SAVLIN          ;CLEAR THE LINE NUMBER INDICATOR
4555  022642  005002                            CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
4556  022644  005003                            CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
4557  022646  012737  000040  001376            MOV     #40,XMTCNT      ;TRANSMIT A BINARY COUNT PATTERN(00-40)
```

```
 4558  022654  012700  001422                        MOV     #TDO,R0         ;POINT TO THE DATA AREA
 4559  022660  005020                                CLR     (R0)+           ;CLEAR A DATA WORD
 4560  022662  022700  001462                        CMP     #STOP,R0        ;FINISHED ?
 4561  022666  001374                                BNE     .-6             ;NO
 4562  022670  005000                                CLR     R0              ;CLEAR OFFSET
 4563  022672  012777  023410  157176                MOV     #XMTSRV,@DZTIV  ;SET UP THE TRANSMITTER INTERRUPT VECTOR
 4564  022700  012777  023232  157164                MOV     #PARESE,@DZRIV  ;SET UP THE RECEIVER INTERRUPT VECTOR
 4565  022706  013777  027060  157160                MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
 4566  022714  013777  027060  157156                MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
 4567  022722  052777  040140  157112                BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
 4568  022730  113777  001364  157120                MOVB    LINE,@DZTCR     ;ENABLE ALL SELECTED LINES
 4569  022736  106427  000000                        MTPS    #0              ;ALLOW INTERRUPTS
 4570  022742  032777  000100  157072  5$:           BIT     #RIE,@DZCSR     ;WHEN RX DONE; RIE WILL =0
 4571  022750  001407                                BEQ     6$              ;BR IF ALL DONE
 4572  022752  005237  023404                        INC     COUNT0
 4573  022756  102771                                BVS     5$
 4574  022760  105237  023406                        INCB    COUNT1
 4575  022764  100366                                BPL     5$
 4576  022766  104011                                ERROR   11              ;*RX FAILED TO FINISH (INTERRUPT)
 4577  022770  104400                        6$:     ADVANCE                 ;ADVANCE LOOP
 4578                                                ;************************* TEST 36 *********************************
 4579                                                ;*THIS TEST VERIFIES THAT ODD PARITY WORKS FOR ALL ODD LINES
 4580                                                ;* SELECTED AND THAT EVEN PARITY WORKS FOR ALL EVEN LINES SELECTED
 4581                                                ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
 4582                                                ;*THAT 'PE' (PARITY ERROR) CAN BE FLAGGED BY
 4583                                                ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
 4584                                                ;*YOU ARE IN 'STAGGERED' MODE.
 4585                                                ;*40(8) CHARS ARE USED FOR THIS TEST.
 4586                                                ;*ALL SELECTED LINES WILL BE ENABLED
 4587                                                ;*AT THE SAME TIME!
 4588                                        ;:*  TEST 36
 4589                                        ;;*******************************************************************
 4590  022772  000004                        TST36:  SCOPE
 4591  022774  012737  000036  001122                MOV     #36,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
 4592  023002  012737  004562  001360                MOV     #$EOP,NEXT      ;POINT TO THE END-OF-PASS HANDLER
 4593  023010  005737  001370                        TST     MODE            ;IS THIS STAGGERED MODE?
 4594  023014  100105                                BPL     6$              ;IF NOT, DON'T DO THIS TEST
 4595  023016  104417                                DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
 4596  023020  013701  001366                        MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
 4597  023024  042701  000200                        BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
 4598  023030  052701  000100                        BIS     #PARITY,R1      ;MAKE SURE PARITY IS TURNED ON
 4599  023034  012702  000001                        MOV     #1,R2           ;USE R2 AS A LINE POINTER
 4600  023040  030237  001364                1$:     BIT     R2,LINE         ;IS THIS A VALID LINE?
 4601  023044  001411                                BEQ     3$              ;IF NOT, SKIP TO THE NEXT LINE
 4602  023046  032701  000001                        BIT     #BIT0,R1        ;IS THIS LINE AN ODD LINE?
 4603  023052  001402                                BEQ     2$              ;IF IT'S EVEN, USE EVEN PARITY
 4604  023054  052701  000200                        BIS     #ODDPAR,R1      ;IF IT'S ODD, USE ODD PARITY
 4605  023060  010177  156766                2$:     MOV     R1,@DZLPR       ;LOAD THE LINE PARAMETER REGISTER
 4606  023064  042701  000200                        BIC     #ODDPAR,R1      ;SET UP THE NEXT PARITY TO EVEN
 4607  023070  005201                        3$:     INC     R1              ;POINT TO THE NEXT LINE
 4608  023072  106302                                ASLB    R2              ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
 4609  023074  103361                                BCC     1$              ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
 4610  023076  005037  001372                        CLR     SAVLIN          ;CLEAR THE LINE NUMBER INDICATOR
 4611  023102  005002                                CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
 4612  023104  005003                                CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
 4613  023106  012737  000040  001376                MOV     #40,XMTCNT      ;TRANSMIT A BINARY COUNT PATTERN(00-40)
```

```
4614  023114  012700  001422                  MOV     #TD0,R0          ;POINT TO THE DATA AREA
4615  023120  005020                          CLR     (R0)+            ;CLEAR A DATA WORD
4616  023122  022700  001462                  CMP     #STOP,R0         ;FINISHED ?
4617  023126  001374                          BNE     .-6              ;NO
4618  023130  005000                          CLR     R0               ;CLEAR OFFSET
4619  023132  012777  023410  156736          MOV     #XMTSRV,@DZTIV   ;SET UP THE TRANSMITTER INTERRUPT VECTOR
4620  023140  012777  023232  156724          MOV     #PARESE,@DZRIV   ;SET UP THE RECEIVER INTERRUPT VECTOR
4621  023146  013777  027060  156720          MOV     DZPRT,@DZRIS     ;SET THE INTERRUPT VECTOR STATUS
4622  023154  013777  027060  156716          MOV     DZPRT,@DZTIS     ;SET TRANSMITTER INTERRUPT PRIORITY
4623  023162  052777  040140  156652          BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
4624  023170  113777  001364  156660          MOVB    LINE,@DZTCR      ;ENABLE ALL SELECTED LINES
4625  023176  106427  000000                  MTPS    #0               ;ALLOW INTERRUPTS
4626  023202  032777  000100  156632  5$:     BIT     #RIE,@DZCSR      ;WHEN RX DONE; RIE WILL =0
4627  023210  001407                          BEQ     6$               ;BR IF ALL DONE
4628  023212  005237  023404                  INC     COUNT0
4629  023216  102771                          BVS     5$
4630  023220  105237  023406                  INCB    COUNT1
4631  023224  100366                          BPL     5$
4632  023226  104011                          ERROR   11               ;*RX FAILED TO FINISH (INTERRUPT)
4633  023230  104400                  6$:     ADVANCE                  ;ADVANCE LOOP
```

```
4634
4635                                    ;RECEIVER SERVICE ROUTINE(PARITY TEST ONLY)
4636
4637  023232  017704  156610      PARESE: MOV     @DZRBUF,R4      ;GET THE CHARACTER
4638  023236  010401                      MOV     R4,R1           ;COPY THE RECEIVED INFORMATION
4639  023240  000301                      SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
4640  023242  042701  177770              BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
4641  023246  010137  001372              MOV     R1,SAVLIN       ;FILL LOC. FOR ERROR PRINTOUT
4642  023252  005704                      TST     R4              ;WAS DATA VALID?
4643  023254  100401                      BMI     10$             ;BRANCH IF YES
4644  023256  104023                      ERROR   23              ;ERROR - DATA VALID NOT SET!
4645  023260  006301              10$:    ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
4646  023262  032704  010000              BIT     #PARER,R4       ;PARITY ERROR SHOULD BE SET. IS IT?
4647  023266  001013                      BNE     11$             ;IF SO, GO CHECK CHARACTER
4648  023270  013737  002046  001400      MOV     DZRBUF,REGIST   ;SET UP FOR THE ERROR MESSAGE
4649  023276  010405                      MOV     R4,R5
4650  023300  042705  000377              BIC     #377,R5
4651  023304  156105  001442              BISB    TR0(R1),R5      ;GET THE CORRECT CHARACTER
4652  023310  052705  110000              BIS     #DVALID!PARER,R5 ;BUILD WHAT WAS EXPECTED
4653  023314  104006                      ERROR   6               ;*ERROR- DID NOT GET CORRECT INFORMATION
4654  023316  126104  001442      11$:    CMPB    TR0(R1),R4      ;CHECK THE CHARACTER. IS IT CORRECT?
4655  023322  001407                      BEQ     12$             ;IF SO, GO SET UP NEXT CHARACTER
4656  023324  116105  001442              MOVB    TR0(R1),R5      ;LOAD THE CHARACTER FOR ERROR REPORTING
4657  023330  042705  177400              BIC     #^C<377>,R5     ;CLEAR SIGN EXTEND
4658  023334  042704  177400              BIC     #^C<377>,R4     ;REMOVE THE JUNK FROM R4, THE ACTUAL CHARACTER
4659  023340  104005                      ERROR   5               ;DATA ERROR
4660  023342  005261  001442      12$:    INC     TR0(R1)         ;SET UP THE NEXT CHARACTER
4661  023346  005203                      INC     R3              ;ADD TO THE TOTAL RECEIVED COUNT
4662  023350  005037  023404              CLR     COUNT0          ;RESET COUNTERS TO NEXT
4663  023354  005037  023406              CLR     COUNT1          ;RECIEVER INTERRUPT
4664  023360  032777  040000  156454      BIT     #TIE,@DZCSR     ;ARE TRANSMISSIONS DONE?
4665  023366  001005                      BNE     13$             ;IF NO, GO RECEIVE SOME MORE
4666  023370  020203                      CMP     R2,R3           ;ARE ALL CHARACTERS RECEIVED?
4667  023372  001003                      BNE     13$             ;IF NO, GO RECEIVE SOME MORE
4668  023374  042777  000100  156440      BIC     #RIE,@DZCSR     ;DISABLE RECEIVER INTERRUPTS
4669  023402  000002              13$:    RTI                     ;GO BACK TO RECEIVER WAIT LOOP
4670  023404  000000              COUNT0: 0
4671  023406  000000              COUNT1: 0
4672
4673
4674                                    ;TRANSMITTER INTERRUPT SERVICE
4675                                    ;-----------------------------
4676
4677  023410  117701  156430      XMTSRV: MOVB    @HDZCSR,R1      ;GET THE LINE NUMBER. IS THE TRANSMITTER
4678  023414  100411                      BMI     1$              ;REALLY READY? IF SO, GO LOAD THE CHARACTER
4679  023416  013700  001372              MOV     SAVLIN,R0       ;ADJUST LOCATION SAVLIN
4680  023422  042701  177770              BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
4681  023426  010137  001372              MOV     R1,SAVLIN       ;FOR ERROR PRINTOUT
4682  023432  104003                      ERROR   3               ;*TRANSMITTER NOT READY- FALSE INTERRUPT
4683  023434  010037  001372              MOV     R0,SAVLIN       ;RESET SAVLIN TO PREVIOUS VALUE
4684  023440  042701  177770      1$:     BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
4685  023444  006301                      ASL     R1              ;MAKE SURE IT REFERENCES A WORD BOUNDARY
4686  023446  116177  001422  156412      MOVB    TD0(R1),@DZTDR  ;LOAD THE CURRENT CHARACTER FOR THIS LINE
4687  023454  005261  001422              INC     TD0(R1)         ;SET UP NEXT CHARACTER FOR THIS LINE
4688                                  ;*****************************************************************
4689                                                                  ;(REV. F0: DELETED  INC   *
```

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

```
                                        ;RECEIVER SERVICE ROUTINE

023554  105777  156262          RXISR1: TSTB    @DZCSR          ;IS THE RECEIVER REALLY READY?
023560  100401                          BMI     1$              ;IF SO, GO SERVICE IT
023562  104004                          ERROR   4               ;*ERROR- RECEIVER DONE FLAG ISN'T SET
023564  017704  156256          1$:     MOV     @DZRBUF,R4      ;SAVE THE RECEIVER INFORMATION
023570  100401                          BMI     2$              ;IF IT WAS VALID, GO PROCESS IT
023572  104023                          ERROR   23              ;ERROR- DATA VALID WASN'T SET
023574  032704  070000          2$:     BIT     #OVRRUN!FRMERR!PARER,R4 ;ARE ANY ERROR FLAGS SET?
023600  001403                          BEQ     3$              ;IF NOT, GO CONTINUE PROCESSING
023602  013700  002046                  MOV     DZRBUF,R0       ;SET UP FOR ERROR REPORTING
023606  104002                          ERROR   2               ;ERROR- RECEIVER ERROR FLAG SET
023610  010401          3$:     MOV     R4,R1           ;COPY THE RECEIVER INFORMATION
023612  000301                  SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
023614  042701  177770          BIC     #^C<7>,R1       ;ISOLATE THE LINE NUMBER
023620  006301                  ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
023622  120461  001442          CMPB    R4,TR0(R1)      ;IS THE CHARACTER WHAT IT SHOULD BE?
023626  001413                  BEQ     4$              ;IF SO,GO CONTINUE PROCESSING
023630  116105  001442          MOVB    TR0(R1),R5      ;GET WHAT WAS EXPECTED FOR ERROR REPORTING
023634  042705  177400          BIC     #^C<377>,R5     ;ELIMINATE PROPAGATED SIGN
023640  042704  177400          BIC     #^C<377>,R4     ;ISOLATE THE ACTUAL CHARACTER
023644  010137  001372          MOV     R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER ERROR
023650  006237  001372          ASR     SAVLIN          ;ALIGN IT CORRECTLY FOR REPORTING
023654  104005                  ERROR   5               ;*DATA ERROR
023656  005261  001442  4$:     INC     TR0(R1)         ;SET UP THE NEXT EXPECTED CHARACTER
023662  005203                  INC     R3              ;INCREMENT THE COUNT OF RECEIVED CHARACTERS
023664  032761  000020  001442  BIT     #20,TR0(R1)     ;HAVE ALL CHARACTERS BEEN RECEIVED?
023672  001402                  BEQ     5$              ;IF NOT, GO RECEIVE SOME MORE
023674  020203                  CMP     R2,R3           ;HAVE WE RECEIVED ALL CHARACTERS?
023676  001401                  BEQ     6$              ;IF SO,GO DETERMINE THE TIMING
023700  000002          5$:     RTI                     ;GO CONTINUE TIMING AND ALLOW INTERRUPTS
023702  004737  023526  6$:     JSR     PC,BUILD        ;GET THE RELATIVE TIME (SIGNIFICANT BITS)

023706  013700  023766          MOV     OFFSET,R0       ;GET POINTER
023712  013760  001220  002102  MOV     $TMP1,TMTBL(R0) ; SAVE THIS TEST'S TIME
023720  005737  023766          TST     OFFSET          ;FIRST TEST?
023724  001414                  BEQ     7$              ;IF NOT, GO CHECK THE TIME
023726  005740                  TST     -(R0)           ;POINT TO THE PREVIOUS  TIME TAKEN
023730  026037  002102  001220  CMP     TMTBL(R0),$TMP1 ;IS THIS TIME WHAT IT SHOULD BE?
023736  101007                  BHI     7$              ;IF SO, GO TO THE NEXT TEST
023740  016005  002102          MOV     TMTBL(R0),R5    ;PLACE WHAT WAS EXPECTED IN R5
023744  010137  001372          MOV     R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER
023750  006237  001372          ASR     SAVLIN          ;MAKE SURE IT'S THE LINE NUMBER
023754  104021                  ERROR   21              ;TIMING ERROR
023756  042777  000140  156056  7$:     BIC     #RIE!MSENAB,@DZCSR ;DISABLE THE DEVICE
023764  000002                  RTI                     ;RETURN TO THE PROGRAM
023766  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

023770  012706  001120  XSTART: MOV     #STACK,SP       ;SET UP PROCESSOR STACK
023774  106427  000340          MTPS    #PR7            ;LOCK OUT INTERRUPTS
024000  012737  023770  001126  MOV     #XSTART,$LPADR  ;SET UP IN CASE OF POWER FAIL
024006  005037  026164          CLR     STFLG           ;CLEAR TEST START FLAG
024012  005037  001242          CLR     $PASS           ;CLEAR PASS COUNT
024016  005037  001132          CLR     $ERTTL          ;CLEAR ERROR COUNT
024022  105037  001123          CLRB    $ERFLG          ;CLEAR ERROR FLAG
024026  005037  026170          CLR     LAST            ;CLEAR LAST ERROR PC
```

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

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

```
4933  024566  012777  024642  155276          MOV     #INTSVC,@DZRIV  ;SET UP INTERRUPT SERVICE
4934  024574  013777  026210  155272          MOV     PRIO,@DZRIS     ;AND LEVEL
4935  024602  106437  027062                  MTPS    @#LESS1         ;ALLOW INTERRUPTS
4936  024606  012777  000140  155226          MOV     #RIE!MSENAB,@DZCSR ;SET RECEIVER INTERRUPT ENABLE
4937  024614  104402  026364                  TYPE    ,MCHAR          ;TYPE ''ANY CHARACTER''
4938  024620  105777  154340          1$:     TSTB    @$TKS           ;IF SOMBODY HITS A KEY- GET NEW LINE #
4939  024624  100375                          BPL     1$              ;LOOP HERE
4940  024626  005777  154334                  TST     @$TKB           ;CLEAR CHAR
4941  024632  004737  007360                  JSR     PC,SERV.G       ;MAKE SURE IT WASN'T <^G>
4942  024636  000137  024252                  JMP     LINEX           ;
4943
4944
4945                                  ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
4946  024642  105777  155174          INTSVC: TSTB    @DZCSR          ;TEST REC. FLAG
4947  024646  100401                          BMI     .+4
4948  024650  104004                          ERROR   4               ;ERROR - INTERRUPT NOT CAUSED BY FLAG
4949  024652  017737  155170  026212          MOV     @DZRBUF,RECDAT
4950  024660  100401                          BMI     .+4
4951  024662  104023                          ERROR   23              ;NON- VALID CHARACTER
4952  024664  032737  020000  026212          BIT     #BIT13,RECDAT   ;CHECK FOR FRAMING ERROR
4953  024672  001401                          BEQ     .+4             ;BR IF NO ERROR
4954  024674  104025                          ERROR   25              ;EITHER SOMBODY HIT THE
4955                                          ;''BREAK KEY'' OR YOU HAVE AN ERROR!
4956  024676  113737  026212  026214          MOVB    RECDAT,TBUF     ;MOVE CHARACTER TO OUTPUT AREA
4957  024704  113737  026212  010620          MOVB    RECDAT,INBUF    ;MOVE CHARACTER TO CHECK FOR ^C
4958  024712  042737  177600  010620          BIC     #^C<177>,INBUF  ;STRIP JUNK PLUS PARITY
4959  024720  042737  174377  026212          BIC     #174377,RECDAT  ;SAVE ONLY LINE  NUMBER
4960  024726  000337  026212                  SWAB    RECDAT
4961  024732  023737  001372  026212          CMP     SAVLIN,RECDAT   ;DOES THE LINE # COMPARE?
4962  024740  001401                          BEQ     .+4
4963  024742  104015                          ERROR   15              ;*WRONG LINE NUMBER
4964  024744  012777  000040  155070          MOV     #MSENAB,@DZCSR  ;START THE TRANSMITTERS SCANNER
4965  024752  123727  010620  000003          CMPB    INBUF,#3                ;IS IT A ^C ?
4966  024760  001004                          BNE     1$              ;NO
4967  024762  104413                          DEVICE.CLR
4968  024764  012716  025476                  MOV     #XEOP,(SP)      ;CRUNCH STACK
4969  024770  000002                          RTI
4970  024772  005003                  1$:     CLR     R3              ;INITIALIZE DELAY
4971  024774  013777  026206  155054          MOV     NUMTCR,@DZTCR   ;ENABLE THE LINE
4972  025002  005777  155034          10$:    TST     @DZCSR          ;TRANSMITTER READY?
4973  025006  100403                          BMI     2$              ;IF YES BRANCH
4974  025010  005203                          INC     R3              ;INCREMENT DELAY
4975  025012  001373                          BNE     10$             ;DELAY DONE?
4976  025014  104003                          ERROR   3               ;TRANSMIT READY NOT SET!
4977  025016  113777  026214  155042  2$:     MOVB    TBUF,@DZTDR     ;TRANSMIT THE CHARACTER
4978  025024  012777  000140  155010          MOV     #RIE!MSENAB,@DZCSR      ;RESTART THE RECEIVER
4979  025032  005077  155020                  CLR     @DZTCR          ;CLEAR TCR BIT
4980  025036  000002                          RTI
4981
4982
4983                                  ;THIS TEST TRANSMITS A BINARY COUNT PATTERN
4984                                  ;VIA INTERRUPT MODE TO THE RECEIVER
4985                                  ;...THE LINE UNDER TEST MUST BE TERMINATED WITH THE TEST CONNECTOR
4986  025040  106427  000340          TEST2:  MTPS    #PR7            ;DISABLE INTERRUPTS
4987  025044  012737  000002  001122          MOV     #2,$TSTNM
4988  025052  012737  025476  001360          MOV     #XEOP,NEXT
```

```
 4989  025060  104413                                DEVICE.CLR              ;
 4990                                                ;*TEST TO VERIFY THAT SETTING DTR FOR A GIVEN LINE
 4991                                                ;*WILL BRING UP ''CO'' AND 'RING'' FOR THE SAME LINE
 4992                                                ;*THE DIST PNL MUST HAVE JUMPER FROM DTR TO RQST TO SEND
 4993                                                ;*IN ORDER FOR THIS TEST TO WORK!
 4994  025062  012737  025070  001362                MOV     #1$,LOCK        ;LOOP
 4995  025070  113777  026206  154762  1$:     MOVB    NUMTCR,@HDZTCR  ;SET DTR
 4996  025076  005005                        CLR     R5              ;
 4997  025100  153705  026206                BISB    NUMTCR,R5       ;BUILD EXPECTED
 4998  025104  000305                        SWAB    R5              ;PUT IN HIGH BYTE
 4999  025106  153705  026206                BISB    NUMTCR,R5       ;
 5000  025112  104414                        DELAY                   ;WAIT FOR CABLE DELAY
 5001  025114  017704  154742                MOV     @DZMSR,R4       ;READY MODEM BITS
 5002  025120  020504                        CMP     R5,R4           ;ARE THEY OK?
 5003  025122  001401                        BEQ     2$              ;BR IF YES
 5004  025124  104022                        ERROR   22              ;IS THE TEST CONNECTOR ON?
 5005                                                                ;HAS RIGHT LINE BEEN SELECTED?
 5006                                                                ;IF SO- YOU HAVE A PROBLEM!
 5007                                                                ;MODEM BITS NOT RIGHT
 5008  025126  104401                  2$:     SCOP1                   ;LOOP
 5009  025130  104413                  3$:     DEVICE.CLR              ;INIT DZ11
 5010  025132  013737  026202  001366          MOV     SPEED,PAR       ;SET LINE SPEED
 5011  025140  053737  026204  001366          BIS     NUMLIN,PAR      ;SELECT LINE # & REC.  INTERRUPT ENABLE
 5012  025146  052737  010000  001366          BIS     #RCVON,PAR      ;ENABLE THE RECEIVER FOR THIS LINE
 5013  025154  052777  040140  154660          BIS     #TIE!RIE!MSENAB,@DZCSR ;SET TRANSMITTER INTERRUPT ENABLE
 5014  025162  012777  025276  154702          MOV     #INTREC,@DZRIV  ;SET UP INTR SERVICE
 5015  025170  013777  026210  154676          MOV     PRIO,@DZRIS     ;SET UP LEVEL
 5016  025176  012777  025456  154672          MOV     #INTRAN,@DZTIV  ;SET UP INTR SERVICE
 5017  025204  013777  026210  154666          MOV     PRIO,@DZTIS     ;SET UP LEVEL
 5018  025212  005001                          CLR     R1              ;RX DATA POINTER- SET TO 0
 5019  025214  005002                          CLR     R2              ;TX DATA POINTER- SET TO 0
 5020  025216  013777  001366  154626          MOV     PAR,@DZLPR      ;SET THE PARAMETERS AND TURN ON RECEIVER
 5021  025224  106437  027062                  MTPS    @#LESS1         ;ALLOW INTERRUPTS
 5022  025230  013777  026206  154620          MOV     NUMTCR,@DZTCR   ;SET UP TCR BIT
 5023
 5024                                          ;YOU RETURN HERE AFTER EVERY RECEIVER INTERRUPT
 5025  025236  105777  153722          SPIN:   TSTB    @$TKS           ;IF SOMEBODY HITS A KEY- GET A NEW LINE #
 5026  025242  100006                          BPL     1$              ;BR IF NO KEY HIT
 5027  025244  005777  153716                  TST     @$TKB           ;CLEAR CHAR
 5028  025250  004737  007360                  JSR     PC,SERV.G       ;MAKE SURE IT WASN'T <^G>
 5029  025254  000137  024252                  JMP     LINEX           ;SW02=1
 5030  025260  005237  026166          1$:     INC     LOCKUP          ;INC TIMEOUT FLAG
 5031  025264  001364                          BNE     SPIN            ;IF NOT 0  RETURN SPINNING
 5032  025266  104011                          ERROR   11              ;*RECEIVER FAILED TO INTERRUPT  CHECK CABLE/TERMINATOR
 5033  025270  104413                  QUITS:  DEVICE.CLR
 5034  025272  000137  025476                  JMP     XEOP            ;CALL FOR END OF PASS
 5035  025276  005037  026166          INTREC: CLR     LOCKUP          ;CLEAR TIMEOUT FLAG
 5036  025302  105777  154534                  TSTB    @DZCSR          ;TEST REC DONE
 5037  025306  100401                          BMI     .+4             ;YES
 5038  025310  104004                          ERROR   4               ;*FALSE INTERRUPT
 5039  025312  017737  154530  026212          MOV     @DZRBUF,RECDAT  ;SAVE WORD
 5040  025320  100401                          BMI     .+4
 5041  025322  104023                          ERROR   23              ;*NON VALID CHARACTER
 5042  025324  032737  040000  026212          BIT     #BIT14,RECDAT   ;DATA OVERRUN ?
 5043  025332  001401                          BEQ     .+4             ;NO
 5044  025334  104024                          ERROR   24              ;*YES
```

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

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

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

```
5189                                    ;THE FOLLOWING IS A TABLE OF LEGAL BAUD RATES (8 BITS/CHAR)
5190  026062  000062              TABLE2: .WORD   50.             ;50 BAUD
5191  026064  010070                      .WORD   10070           ;
5192  026066  000113                      .WORD   75.             ;75 BAUD
5193  026070  010470                      .WORD   10470           ;
5194  026072  000156                      .WORD   110.            ;110 BAUD
5195  026074  011070                      .WORD   11070           ;TWO STOP BITS
5196  026076  000207                      .WORD   135.            ;134.5 BAUD
5197  026100  011470                      .WORD   11470           ;TWO STOP BITS
5198  026102  000226                      .WORD   150.            ;150 BAUD
5199  026104  012070                      .WORD   12070           ;TWO STOP BITS
5200  026106  000454                      .WORD   300.            ;300 BAUD
5201  026110  012430                      .WORD   12430           ;ONE STOP BIT
5202  026112  001130                      .WORD   600.            ;600 BAUD
5203  026114  013030                      .WORD   13030           ;ONE STOP BIT
5204  026116  002260                      .WORD   1200.           ;1200 BAUD
5205  026120  013430                      .WORD   13430           ;ONE STOP BIT
5206  026122  003410                      .WORD   1800.           ;1800 BAUD
5207  026124  014030                      .WORD   14030           ;ONE STOP BIT
5208  026126  003720                      .WORD   2000.           ;2000 BAUD
5209  026130  014430                      .WORD   14430           ;ONE STOP BIT
5210  026132  004540                      .WORD   2400.           ;2400 BAUD
5211  026134  015030                      .WORD   15030           ;ONE STOP BIT
5212  026136  007020                      .WORD   3600.           ;3600 BAUD
5213  026140  015430                      .WORD   15430           ;ONE STOP BIT
5214  026142  011300                      .WORD   4800.           ;4800 BAUD
5215  026144  016030                      .WORD   16030           ;ONE STOP BIT
5216  026146  016040                      .WORD   7200.           ;7200 BAUD
5217  026150  016430                      .WORD   16430           ;ONE STOP BIT
5218  026152  022600                      .WORD   9600.           ;9600 BAUD
5219  026154  017070                      .WORD   17070           ;
5220  026156  177777  000000              .WORD   -1,0            ;TABLE TERMINATOR
5221
5222
5223  026162  000000              WCHFLG:0                        ;ECHO OR CABLE FLAG
5224  026164  000000              STFLG: 0                        ;PROGRAM START FLAG
5225  026166  000000              LOCKUP: 0                       ;TIMEOUT FLAG
5226  026170  000000              LAST:   0                       ;LAST ERROR PC
5227  026172  000000              TDATA:  0
5228  026174  000000              RDATA:  0
5229  026176  000000              BYTCNT: 0
5230  026200  000156              LINESP: 110.                    ;DEFAULT BAUD RATE
5231  026202  006307              SPEED:  6307                    ;DEFAULT 110 BAUD, 8 BITS/CHAR,
5232                                                              ;FDX, 2 STOP BITS
5233  026204  000100              NUMLIN: 100                     ;DEFAULT VALUE, REC. INTERRUPT ENABLED
5234
5235  026206  000001              NUMTCR: 1                       ;DEFAULT VALUE,TCR BIT 0
5236  026210  000240              PRIO:   240                     ;DEFAULT DEVICE PRIORITY 5
5237  026212  000000              RECDAT: 0
5238  026214  000000              TBUF:   0
5239  026216  053200  041505  047524  MVECTO: .ASCIZ  <200>/VECTOR ADDRESS- /
      026240  041600  047117  051124  MREGAD: .ASCIZ  <200>/CONTROL REGISTER ADDRESS- /
      026274  050200  051501  020123  MPASS:  .ASCIZ  <200>/PASS DONE./
      026310  044600  053116  046101  MINVAL: .ASCIZ  <200>/INVALID BAUD RATE - /
      026336  046200  047111  035105  MLINE:  .ASCIZ  <200>/LINE: /
      026346  041200  052501  020104  MSPEED: .ASCIZ  <200>/BAUD RATE - /
```

```
        026364  052200  050131  020105  MCHAR:  .ASCIZ  <200>/TYPE A CHAR. ON DZ11 TERMINAL /
        026424  053600  044510  044103  MWHICH: .ASCIZ  <200>/WHICH TEST ? ECHO OR CABLE (E OR C) /
        026472  052200  051105  044515  MTERM:  .ASCIZ  <200>/TERMINAL ECHO TEST /
        026517     200  040503  046102  MCABLE: .ASCIZ  <200>/CABLE TEST /
        026534  006777  177777  177412  MQUICK: .ASCII  <377><15><377><377><12><377><377>
        026543     124  042510  050440          .ASCII  /THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789/
        026640  006777  177777  177412          .ASCII  <377><15><377><377><12><377><377><377><0>
                026652                  .EVEN
                                        ;;*****************************************************************
 5240                                   ;UTILITIES
 5241                                   ;;*****************************************************************
 5242
 5243                                   ;THIS UTILITY CALCULATES PRIORITY LEVEL,SETS UP CSR'S,SETS UP VECTORS.
 5244  026652  006337  027060          DZLEV:  ASL     DZPRT           ;BUILD PRIORITY IN THIS LOCATION
 5245  026656  006337  027060                  ASL     DZPRT           ;USING ARITHMETIC SHIFTS, ROTATE
 5246  026662  006337  027060                  ASL     DZPRT           ;       THE PRIORITY LEVEL PAST
 5247  026666  006337  027060                  ASL     DZPRT           ;       THE BIT POSITIONS CORRE-
 5248  026672  006337  027060                  ASL     DZPRT           ;       SPONDING TO THE CONDITION CODES
 5249  026676  013737  027060  027062          MOV     DZPRT,LESS1     ;MOVE THIS TO LESS1
 5250  026704  162737  000001  027062          SUB     #1,LESS1        ;CREATE THE NEXT LOWEST PRIORITY
 5251  026712  042737  000037  027062          BIC     #37,LESS1       ;INSURE THAT THE  TNZVC BITS ARE CLEAR
 5252  026720  013700  002072                  MOV     DZRIV,R0        ;PLACE THE BASE VECTOR ADDRESS IN R0
 5253  026724  062700  000002                  ADD     #2,R0           ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
 5254  026730  010037  002074                  MOV     R0,DZRIS        ;STORE IT HERE
 5255  026734  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
 5256  026740  010037  002076                  MOV     R0,DZTIV        ;STORE IT HERE
 5257  026744  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
 5258  026750  010037  002100                  MOV     R0,DZTIS        ;STORE IT HERE
 5259
 5260                                   ;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN DZ11. $BASE IS THE BASE ADDRESS
 5261                                   ;OF THE DEVICE
 5262  026754  013700  001310                  MOV     $BASE,R0        ;COPY THE ADDRESS BEING LOADED
 5263  026760  010037  002042                  MOV     R0,DZCSR        ;XXX0
 5264  026764  005200                          INC     R0
 5265  026766  010037  002044                  MOV     R0,HDZCSR       ;XXX1
 5266  026772  005200                          INC     R0
 5267  026774  010037  002046                  MOV     R0,DZRBUF       ;XXX2
 5268  027000  010037  002052                  MOV     R0,DZLPR        ;XXX2
 5269  027004  005200                          INC     R0
 5270  027006  010037  002050                  MOV     R0,HDZRBUF      ;XXX3
 5271  027012  010037  002054                  MOV     R0,HDZLPR       ;XXX3
 5272  027016  005200                          INC     R0
 5273  027020  010037  002056                  MOV     R0,DZTCR        ;XXX4
 5274  027024  005200                          INC     R0
 5275  027026  010037  002060                  MOV     R0,HDZTCR       ;XXX5
 5276  027032  005200                          INC     R0
 5277  027034  010037  002062                  MOV     R0,DZMSR        ;XXX6
 5278  027040  010037  002066                  MOV     R0,DZTDR        ;XXX6
 5279  027044  005200                          INC     R0
 5280  027046  010037  002064                  MOV     R0,HDZMSR       ;XXX7
 5281  027052  010037  002070                  MOV     R0,HDZTDR       ;XXX7
 5282  027056  000207                          RTS     PC
 5283  027060  000240          DZPRT:  PR5
 5284  027062  000200          LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS
 5285
```

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

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

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

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

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

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

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

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

```
031022    003     001                 .BYTE   3,1
031024  001372                        SAVLIN

031026  000004                DT5:    4
031030    003     005                 .BYTE   3,5
031032  001372                        SAVLIN
031034    006     011                 .BYTE   6,9.
031036  001214                        $REG5
031040    006     007                 .BYTE   6,7
031042  001220                        $TMP1
031044    006     001                 .BYTE   6,1
031046  001400                        REGIST
                                      ;TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES
                                      ;-----------------------------------------------

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

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

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

```
ABASE = 160010            779#  1187  1228
ACDW1 = 000000           1187  1230
ACDW2 = 000000           1187  1231
ACPUOP= 000000           1187  1202
ACTIVE  001412           1290#  1564*  2612*  2613  2615*  2620  2621*  2622  2625*
ADDW0 = 000000           1187  1232
ADDW1 = 000000           1187  1233
ADDW10= 000000           1187  1242
ADDW11= 000000           1187  1243
ADDW12= 000000           1187  1244
ADDW13= 000000           1187  1245
ADDW14= 000000           1187  1246
ADDW15= 000000           1187  1247
ADDW2 = 000000           1187  1234
ADDW3 = 000000           1187  1235
ADDW4 = 000000           1187  1236
ADDW5 = 000000           1187  1237
ADDW6 = 000000           1187  1238
ADDW7 = 000000           1187  1239
ADDW8 = 000000           1187  1240
ADDW9 = 000000           1187  1241
ADEVCT= 000000           1187  1193
ADEVM = 000000           1187  1229
ADRCNT  006353           2181*  2218*  2228#
ADVANC= 104400           1467#  2869  3123  3313  3327  3371  3380  3517  4057  4173  4418  4463  4577
                         4633
AENV  = 000000           1187  1198
AENVM = 000000           1187  1199
AFATAL= 000000           1187  1190
AMADR1= 000000           1187  1215
AMADR2= 000000           1187  1219
AMADR3= 000000           1187  1222
AMADR4= 000000           1187  1225
AMAMS1= 000000           1187  1209
AMAMS2= 000000           1187  1217
AMAMS3= 000000           1187  1220
AMAMS4= 000000           1187  1223
AMSGAD= 000000           1187  1195
AMSGLG= 000000           1187  1196
AMSGTY= 000000           1187  1189
AMTYP1= 000000           1187  1210
AMTYP2= 000000           1187  1218
AMTYP3= 000000           1187  1221
AMTYP4= 000000           1187  1224
APASS = 000000           1187  1192
APRIOR= 000000           1187
APTCSU= 000040           2032  2137#
APTENV= 000001           2025  2093  2135#  2410
APTSIZ= 000200           2134#
APTSPO= 000100           2027  2095  2136#
ASWREG= 000000           1187  1200
ATESTN= 000000           1187  1191
AUNIT = 000000           1187  1194
AUSWR = 000000           1187  1201
AUTO.S  011612           1798  2718#
AVECT = 000300            779#
```

```
AVECT1= 000000          1187    1226
AVECT2= 000000          1187    1227
BAUD    024234          4862#   5182
BINWRD  006626          2304#
BIT0   = 000001          895#    972     974     976     978    1015    1034    3174    3329    4546    4602
BIT00  = 000001          885#    895
BIT01  = 000002          884#    894
BIT02  = 000004          883#    893
BIT03  = 000010          882#    892
BIT04  = 000020          881#    891
BIT05  = 000040          880#    890
BIT06  = 000100          879#    889
BIT07  = 000200          878#    888
BIT08  = 000400          877#    887
BIT09  = 001000          876#    886
BIT1   = 000002          894#    973     974     977     978    1016    1035    3174
BIT10  = 002000          875#    942     943     944     945     963     964     965     966    1000    1001    1002    1003
                        1008    1009    1010    1011    1025    1044    1056    3174
BIT11  = 004000          874#   1004    1005    1006    1007    1008    1009    1010    1011    1026    1045    1057    1965
                        3174    3255
BIT12  = 010000          873#    931     951     994    1027    1046    1058    5048
BIT13  = 020000          872#    932     952    1028    1047    1059    4952    5045
BIT14  = 040000          871#    933     953    1029    1048    1060    1947    2811    5042
BIT15  = 100000          870#    934     954    1030    1049    1061    1665    1668    2630    2761
BIT2   = 000004          893#    975     976     977     978    1017    1036    1851    3174
BIT3   = 000010          892#    926     981     983     985     987    1018    1037    4514    4516
BIT4   = 000020          891#    927     982     983     986     987    1019    1038    4514
BIT5   = 000040          890#    928     984     985     986     987     992    1020    1039    2730    2740    2811
BIT6   = 000100          889#    929     989    1021    1040
BIT7   = 000200          888#    930     990    1022    1041    1796    2440    2453    2474    2731    2738    2813
BIT8   = 000400          887#    939     941     943     945     960     962     964     966     997     999    1001    1003
                        1005    1007    1009    1011    1023    1042    1054    3174
BIT9   = 001000          886#    940     941     944     945     961     962     965     966     998     999    1002    1003
                        1006    1007    1010    1011    1024    1043    1055    3174
BPTVEC= 000014           902#
BRK0   = 000400         1054#
BRK1   = 001000         1055#
BRK2   = 002000         1056#
BRK3   = 004000         1057#
BRK4   = 010000         1058#
BRK5   = 020000         1059#
BRK6   = 040000         1060#
BRK7   = 100000         1061#
BRW     005232          1856    1984#
BUILD   023526          4710#   4749
BYTCNT  026176          5229#
CHRCNT  006624          2270*   2283*   2301#
CNVRT  = 104412         1487#   1878    1880    1883    1886    2389    2391    2393    2448
CONVRT= 104411          1485#   1808    2407
CORMAX  031110          5383#
COUNT0  023404          4572*   4628*   4662*   4670#
COUNT1  023406          4574*   4630*   4663*   4671#
CO0    = 000400         1042#
CO1    = 001000         1043#
CO2    = 002000         1044#
CO3    = 004000         1045#
```

```
CO4    = 010000          1046#
CO5    = 020000          1047#
CO6    = 040000          1048#
CO7    = 100000          1049#
CR     = 000015           810#  2071  2081
CRLF   = 000200           811#  2042  2081
CSRMAP  011620           2721#
CYCLE   011070           1857  1908  2601#
DATABP  007220           2378* 2381  2405  2408#
DATAHD  007206           2377* 2401  2404#
DCLASM= 104417           1497# 3451  3538  3621  3709  3770  3839  3915  4023  4075  4205  4288  4465
                         4539  4595
DCLR   = 000020            927# 2324  2325  2885  3212  3224  3232
DDISP  = 177570            817# 1158
DELAY  = 104414          1491# 2748  3192  3339  3345  3386  3392  3422  3472  3495  3511  3565  3588
                         3636  3659  3677  3745  3794  3817  3864  3889  3950  3958  4044  4051  4105
                         4121  4131  4237  4257  4914  5000
DEVADR  006350           2179* 2215  2226#
DEVICE= 104413           1489# 2331  2931  2963  2995  3027  3059  3094  3140  3210  3250  3276  3315
                         3410  3431  3680  3828  3900  4399  4870  4903  4924  4927  4967  4989  5009
                         5033
DH1     030554           5292  5383#
DH2     030577           5296  5383#
DH3     030632           5300  5304  5316  5320  5324  5328  5332  5336  5344  5348  5352  5364  5368
                         5372  5376  5380  5383#
DH4     030644           5308  5312  5360  5383#
DH5     030673           5356  5383#
DISPLA  001162           1158# 1580* 1977* 2420*
DISPRE  000174           1120# 1580
DLYCNT  006722           1693* 1783* 1833  2337  2342#  4314
DLYTBL  031050           1783  5383#
DONFLG  001420           1299# 3619* 3651  3673*
DSWR   = 177570            816# 1157
DTR0   = 000400           1023# 3119
DTR1   = 001000           1024#
DTR2   = 002000           1025#
DTR3   = 004000           1026#
DTR4   = 010000           1027#
DTR5   = 020000           1028#
DTR6   = 040000           1029#
DTR7   = 100000           1030#
DT1     030752           5293  5383#
DT2     030764           5297  5383#
DT3     031002           5301  5305  5317  5321  5325  5329  5333  5337  5345  5349  5353  5365  5369
                         5373  5377  5381  5383#
DT4     031010           5309  5313  5361  5383#
DT5     031026           5357  5383#
DVALID= 100000            954# 3255  3491  3584  3741  4148  4652
DZACTV  001404           1285# 1706* 1823* 1824  2601  2607  2687* 2778* 2782* 2787  2806
DZCR0   001500           1347# 1619  1622  1707  2784
DZCR1   001514           1354# 1708
DZCR10  001640           1403#
DZCR11  001654           1410#
DZCR12  001670           1417#
DZCR13  001704           1424#
DZCR14  001720           1431#
```

```
DZCR15  001734  1438#
DZCR16  001750  1445#
DZCR17  001764  1452#
DZCR2   001530  1361#
DZCR3   001544  1368#
DZCR4   001560  1375#
DZCR5   001574  1382#
DZCR6   001610  1389#
DZCR7   001624  1396#
DZCSR   002042  1505#  1912  2324*  2325  2332*  2844  2884  2916  2948  2980  3012  3044  3172
                3211  3411  3462*  3466  3470  3493  3514  3521  3555*  3559  3563  3586  3632*
                3634  3662  3683  3721*  3743  3787*  3789  3804*  3809*  3812  3856*  3858  3870*
                3875*  3880*  3883  3898*  3929*  3943  3948  3956  3962  3967*  3968*  4007*  4033*
                4042  4048*  4054  4091*  4098  4103  4113  4119  4153  4179*  4183*  4188*  4223*
                4235  4245  4251  4255  4306*  4307*  4308*  4329  4354  4362  4498*  4500  4567*
                4570  4623*  4626  4664  4668*  4701*  4720  4762*  4843  4846  4909*  4912  4936*
                4946  4964*  4972  4978*  5013*  5036  5071  5263*
DZLEV   026652  2634  5244#
DZLPR   002052  1509#  3184*  3253*  3261*  3456*  3543*  3625*  3715*  3775*  3844*  3920*  4028*  4080*
                4210*  4294*  4467*  4478*  4549*  4605*  4848*  4908*  4932*  5020*  5268*
DZLV0   001504  1349#  1644  1647  1665*  1668*
DZLV1   001520  1356#
DZLV10  001644  1405#
DZLV11  001660  1412#
DZLV12  001674  1419#
DZLV13  001710  1426#
DZLV14  001724  1433#
DZLV15  001740  1440#
DZLV16  001754  1447#
DZLV17  001770  1454#
DZLV2   001534  1363#
DZLV3   001550  1370#
DZLV4   001564  1377#
DZLV5   001600  1384#
DZLV6   001614  1391#
DZLV7   001630  1398#
DZMSR   002062  1513#  2862  3277  3314  3374  4855*  5001  5277*
DZNUM   001410  1287#  1561  1698*  1889  2726*  2767*  2768  2774  2776
DZPRT   027060  2628*  2629  2630*  3780*  3785  3786  3807  3808  3854  3855  3878  3879  3926
                3928  4495  4496  4565  4566  4621  4622  5244*  5245*  5246*  5247*  5248*  5249
                5283#
DZRBUF  002046  1507#  2850  3251  3499  3553  3592  3666  3687  3722  3749  3940  3978  4149
                4163  4228  4357  4637  4648  4723  4728  4846*  4847*  4848  4849  4949  5039
                5267*
DZRIS   002074  1519#  3785*  3807*  3854*  3878*  3926*  4008  4009*  4222*  4303*  4414  4415*  4495*
                4565*  4621*  4934*  5015*  5254*
DZRIV   002072  1518#  1915  2627*  3784*  3806*  3853*  3877*  3925*  4008*  4221*  4250*  4302*  4414*
                4494*  4564*  4620*  4835  4933*  5014*  5252
DZTCR   002056  1511#  2856  3075  3118  3185*  3417*  3465*  3504*  3558*  3599*  3631*  3649*  3657*
                3725*  3756*  3781*  3850*  3939*  4000*  4040*  4097*  4167*  4182*  4233*  4271*  4309*
                4349*  4497*  4568*  4624*  4699*  4849*  4850*  4851  4853  4905*  4971*  4979*  5022*
                5273*
DZTDR   002066  1515#  3279*  3286*  3476*  3523*  3569*  3654*  3727*  3810*  3881*  3954*  4049*  4109*
                4191*  4241*  4253*  4346*  4686*  4853*  4854*  4855  4919*  4977*  5074*  5278*
DZTIS   002100  1521#  3786*  3808*  3855*  3879*  3928*  4010  4011*  4305*  4416  4417*  4496*  4566*
                4622*  5017*  5258*
```

```
DZTIV   002076          1520#  3783*  3805*  3852*  3876*  3927*  4010*  4180*  4304*  4416*  4493*  4563*  4619*
                        5016*  5256*
DZV   = ****** U        2575
DZVC0   001502          1348#  1632   1635   2827
DZVC1   001516          1355#
DZVC10  001642          1404#
DZVC11  001656          1411#
DZVC12  001672          1418#
DZVC13  001706          1425#
DZVC14  001722          1432#
DZVC15  001736          1439#
DZVC16  001752          1446#
DZVC17  001766          1453#
DZVC2   001532          1362#
DZVC3   001546          1369#
DZVC4   001562          1376#
DZVC5   001576          1383#
DZVC6   001612          1390#
DZVC7   001626          1397#
DZ.END  002000          1459#  1604   2613   2622   2724
DZ.MAP  001500          1290   1345#  1564   1601   1803   2615   2625   2673   2721   2727   2798
EIAFLG  001414          1295#  2629*  3121   3311   3372
EIGHT = 000030           983#  4449   4452
EIGHTS= 000070           987#
EMTVEC= 000030           905#
EM1     027304          5291   5383#
EM10    027721          5327   5383#
EM13    027760          5331   5383#
EM14    030011          5335   5383#
EM15    030043          5339   5383#
EM16    030105          5343   5383#
EM17    030156          5347   5383#
EM2     027357          5295   5383#
EM20    030214          5351   5383#
EM21    030255          5355   5383#
EM22    030305          5359   5383#
EM23    030347          5363   5383#
EM24    030377          5367   5383#
EM25    030425          5371   5383#
EM26    030455          5375   5383#
EM27    030504          5379   5383#
EM3     027405          5299   5383#
EM4     027444          5303   5383#
EM5     027473          5307   5383#
EM6     027522          5311   5383#
EM7     027560          5315   5383#
EM8     027621          5319   5383#
EM9     027663          5323   5383#
ERRMSG  007174          2376*  2396   2399#
ERRVEC= 000004           898#  1952   1953*  1955*  1958*
ERTAB0  007344          2391   2431#
EVEPAR= 000000           993#
EXITER  007300          2419   2422#
FINI    021772          4325   4413#
FIVE  = 000000           980#
FIVES = 000040           984#
```

```
FRMERR= 020000  952#  3741  4368  4726
HALTS   007224  2361  2410#
HDRFLG  001416  1297#  1592*  1602*  1746*  1799  1801*
HDZCSR  002044  1506#  3225*  3640  4332  4677  5265*
HDZLPR  002054  1510#  5271*
HDZMSR  002064  1514#  5280*
HDZRBU  002050  1508#  5270*
HDZTCR  002060  1512#  3338*  3344*  3385*  3391*  4851*  4852*  4995*  5275*
HDZTDR  002070  1516#  3726*  3753*  5281*
HILIM   006346  2178*  2206  2225#
HT    = 000011   808#  2040  2081
INBUF   010620  2148  2184  2543*  2544  2549  2554  2568#  4957*  4958*  4965  5096  5148*  5149
                5153
INIFLG  001415  1296#  1583  1588*  1596  1606*  1732*  1794  2720*
INIT    021252  4292#  4297
INIT1   021276  4299#  4301
INSTER= 104404  1475#  2200  2560  5115  5158
INSTR = 104403  1473#  1614  1627  1639  1674  1681  1738  1771  1828  2640  4830  4838  4858
                4862  4872
INSTR2  006146  2155  2167#
INTRAN  025456  5016  5071#
INTREC  025276  5014  5035#
INTSVC  024642  4933  4946#
IOTVEC= 000020   903#  2828*
LAST    026170  4820*  5083*  5226#
LESS1   027062  3849*  3871*  4310*  4935*  5021*  5249*  5250*  5251*  5284#
LF    = 000012   809#  2075  2081
LIMITS  006274  2194  2206#
LINE    001364  1277#  2631*  3322  3376  3415  3454  3463  3541  3556  3630  3713  3723  3773
                3781  3842  3850  3918  3931  4026  4035  4078  4092  4208  4225  4289  4292
                4309  4342  4376  4403  4450  4544  4568  4600  4624
LINESP  026200  4867  5177  5230#
LINEX   024252  4870#  4942  5029
LINE0   001506  1350#  1690*  1743  1753  1779
LINE1   001522  1357#
LINE10  001646  1406#
LINE11  001662  1413#
LINE12  001676  1420#
LINE13  001712  1427#
LINE14  001726  1434#
LINE15  001742  1441#
LINE16  001756  1448#
LINE17  001772  1455#
LINE2   001536  1364#
LINE3   001552  1371#
LINE4   001566  1378#
LINE5   001602  1385#
LINE6   001616  1392#
LINE7   001632  1399#
LOBITS  006352  2180*  2210  2227#
LOCK    001362  1272#  1992  1994  2348*  2385  2843*  2849*  2855*  2861*  3077*  3106*  3120*  3159*
                3310*  3368*  3450*  3537*  3605*  3706*  3759*  4074*  4204*  4277*  4444*  4994*
LOCKUP  026166  4884*  5030*  5035*  5225#
LOLIM   006344  2177*  2208  2224#
LP0   = 000000   971#
LP1   = 000001   972#
```

```
LP2    = 000002              973#
LP3    = 000003              974#
LP4    = 000004              975#
LP5    = 000005              976#
LP6    = 000006              977#
LP7    = 000007              978#
MAINT  = 000010              926#  1982  2552  2917  3213  3226
MANT0    001512             1352#  1677  1751  2785
MANT1    001526             1359#
MANT10   001652             1408#
MANT11   001666             1415#
MANT12   001702             1422#
MANT13   001716             1429#
MANT14   001732             1436#
MANT15   001746             1443#
MANT16   001762             1450#
MANT17   001776             1457#
MANT2    001542             1366#
MANT3    001556             1373#
MANT4    001572             1380#
MANT5    001606             1387#
MANT6    001622             1394#
MANT7    001636             1401#
MASTEK   010314             2387  2531#
MBADLN   010423             1760  2531#
MCABLE   026517             4891  5239#
MCHAR    026364             4937  5239#
MCSRX    010244             1877  2392  2531#
MDATA    010724             2281  2291  2572#
MEPASS   010063             1876  2531#
MERRPC   010371             2390  2531#
MERRX    010271             1885  2531#
MERR2    010122             2531# 2603  2789
MERR3    010171             1820  2531#
MINVAL   026310             5181  5239#
MLINE    026336             4873  5239#
M  OCK   010215             1853  2531#
M  EW    010317             1815  2531#
M  TFLG  001417             1298# 1979* 1982* 2332  2547* 2552* 2557*
MODE     001370             1279# 1980  2633* 3318  3369  3478  3571  3707  3729  3982  4136  4468  4537
                            4593
MPASS    026274             5082  5239#
MPASSX   010260             1882  2531#
MPFAIL   010020             2524  2531#
MQUICK   026534             4911  5239#
MR       010106             1858  2531#
MREGAD   026240             4839  5239#
MSENAB=  000040              928# 2949  3186  3187  3213  3226  3412  3418  3462  3555  3632  3721  3787
                            3809  3856  3880  3929  4033  4048  4091  4179  4188  4223  4308  4498  4567
                            4623  4762  4909  4936  4964  4978  5013
MSPEED   026346             4863  5239#
MTERM    026472             4897  5239#
MTITLE   001000             1129# 1587
MTSTN    010302             2388  2531#  2641
MVECTO   026216             4831  5239#
MVECX    010252             1879  2531#
```

```
MWHICH  026424          4859    5239#
NEXT    001360          1271#   2347    2427    2840*   2883*   2915*   2947*   2979*   3011*   3043*   3074*   3117*   3171*
                        3209*   3249*   3275*   3309*   3320    3367*   3409*   3449*   3536*   3618*   3705*   3769*   3838*
                        3914*   4022*   4073*   4203*   4287*   4443*   4536*   4592*   4926*   4988*
NOLIST= ****** U         781
NUMLIN  026204          4906    4928    5011    5167*   5233#
NUMTCR  026206          4905    4971    4995    4997    4999    5022    5169*   5176*   5235#
ODDPAR= 000200           990#   3711    4541    4548    4550    4597    4604    4606
OFFSET  023766          4445*   4460*   4509*   4751    4753    4764#
ONESTO= 000000           991#
OTHER   024214          4822    4856#
OUT     021720          4320    4398#
OVRRUN= 040000           953#   4160    4368    4726
PAR     001366          1278#   2632*   3452    3539    3622    3674    3710    3771    3840    3916    4024    4076    4206
                        4290    4540    4596    4906*   4907*   4908    4928*   4929*   4931*   4932    5010*   5011*   5012*
                        5020
PARAM = 104405          1477#   1616    1629    1641    1683    1740    1773    1830    2642    4832    4840    4874
PARAM1  006214          2183#   2201
PARER = 010000           951#   3741    4368    4646    4652    4726
PARERR  006270          2186    2188    2190    2200#   2207    2209    2211
PARESE  023232          4564    4620    4637#
PARITY= 000100           989#   3711    4542    4598
PARMD = 104415          1493#   4864
PAR0    001510          1351#   1691*   1776    1780
PAR1    001524          1358#
PAR10   001650          1407#
PAR11   001664          1414#
PAR12   001700          1421#
PAR13   001714          1428#
PAR14   001730          1435#
PAR15   001744          1442#
PAR16   001760          1449#
PAR17   001774          1456#   1790
PAR2    001540          1365#
PAR3    001554          1372#
PAR4    001570          1379#
PAR5    001604          1386#
PAR6    001620          1393#
PAR7    001634          1400#
PAWCH = 104416          1495#   4860
PIRQ  = 177772           815#
PIRQVE= 000240           909#
POPR0 = 012600           918#
POP1SP= 005726           916#
POP2SP= 022626           920#   1578    2824    3869    3897    4001    4006
PRIO    026210          4934    5015    5017    5236#
PR0   = 000000           832#
PR1   = 000040           833#
PR2   = 000100           834#
PR3   = 000140           835#
PR4   = 000200           836#   5284
PR5   = 000240           837#   5283
PR6   = 000300           838#
PR7   = 000340           839#   1097    1101    1559*   1848*   2842    3936*   4222    4303    4305    4398*   4814*   4882*
                        4925*   4986*
PS    = 177776           812#    813
```

```
PSW    = 177776            813#
PUSHR0= 010046             917#
PUSH1S= 005746             915#
PUSH2S= 024646             919#
PWRVEC= 000024             904#  2493*  2494*  2503*  2509*  2521*  2522*
QUITS   025270            5033#  5065
RCVON  = 010000             994#  3623   4449   4452   4466   4477   4931   5012
RDATA   026174            5228#
RDONE  = 000200             930#  3174   3466   3493   3514   3559   3586   3743   3812   3883   3956   4054
RECDAT  026212             4949*  4952   4956   4957   4959*  4960*  4961   5039*  5042   5045   5048   5053   5058*
                           5059*  5060   5237#
REGIST  001400             1283#  3553*  3722*  4483*  4648*  5383
RESREG  007222             2406   2409#
RESTAR  011434             2526   2666#
RESVEC= 000010              899#
RES05  = 104410            1483#  2409
RIE    = 000100             929#  3013   3213   3226   3809   3880   3968   4007   4223   4306   4498   4500   4567
                           4570   4623   4626   4668   4762   4936   4978   5013
RING0  = 000001            1034#
RING1  = 000002            1035#
RING2  = 000004            1036#
RING3  = 000010            1037#
RING4  = 000020            1038#
RING5  = 000040            1039#
RING6  = 000100            1040#
RING7  = 000200            1041#
RL0    = 000000             959#
RL1    = 000400             960#
RL2    = 001000             961#
RL3    = 001400             962#
RL4    = 002000             963#
RL5    = 002400             964#
RL6    = 003000             965#
RL7    = 003400             966#
RSTART  021242            4290#
RSTRT   025512            5085#
RUN     001406            1286#  1565*  2607   2610*  2611*  2618*  2619*
RXISR1  023554             4494   4720#
RXSVC   021542             4302   4354#
RXTCR   022020             4289*  4318   4392*  4419#
SAVLIN  001372             1280#  3413*  3432*  3460*  3477   3505*  3506   3547*  3570   3600*  3601   3620*  3664*
                           3670*  3685*  3689*  3719*  3728   3755*  3779*  3848*  3924*  3933*  3993   4032*  4037*
                           4084*  4095   4135   4169*  4214*  4231   4273*  4336*  4361*  4379   4401*  4409*  4446*
                           4479*  4480*  4554*  4610*  4641*  4679   4681*  4683*  4739*  4740*  4759*  4760*  4877
                           4961   5060   5167   5168   5383
SAVNUM  001411             1288#  1561*  1699*  1887*  1889*  2777*
SAVPC   001402             1284#  2233*  2433
SAV05  = 104407            1481#  2366
SCOP1  = 104401            1469#  2873   3103   3146   3350   3397   3503   3524   3596   3754   4168   4189   4272
                           4508   5008
SERV.G  007360             1939   2354   2437#  2444   4507   4921   4941   5028
SET     025766             4880   5167#
SETAPT  011440             1591   2673#
SETFLG= 104406             1479#  1676
SET.PS  010766             1096   2575#
SET1    026020             5172   5175#
```

```
SEVEN = 000020          982#
SEVENS= 000060          986#
SILOAL= 020000          932#  3174  4113  4119  4153  4245  4255  4362
SILOEN= 010000          931#  2981  3213  4091  4179  4188  4223
SIX   = 000010          981#
SIXS  = 000050          985#
SNAP    021372         4313#  4393
SPACNT  006625         2302#
SPEED   026202         4907   4929  5010  5184* 5231#
SPIN    025236         5025#  5031  5068
STACK = 001120          803#  1558  1847  2428  4813  4883
STFLG   026164         4816*  4871* 4888  4890* 4894  4896* 5224#
STKLMT= 177774          814#
STOP    001462         1318#  3550  4087  4217  4300  4455  4486  4560  4616
SV05    006362         2237#
SWR     001160         1157#  1575  1579* 1594  1598  1694  1728  1796  1813  1818  1823  1826  1851
                       1947   1965  1990  1997  2355  2360  2418  2423  2425  2437  2445  2446* 2456*
                       2461*  2462* 2463* 2470* 2501  2514* 2637  2706* 4821  4856  4922
SWREG   000176         1121#  1579
SW0   = 000001          867#
SW00  = 000001          857#   867   1594  4821
SW01  = 000002          856#   866   2637  4856
SW02  = 000004          855#   865
SW03  = 000010          854#   864   1694  1728
SW04  = 000020          853#   863   1826
SW05  = 000040          852#   862
SW06  = 000100          851#   861   1813
SW07  = 000200          850#   860
SW08  = 000400          849#   859   2423
SW09  = 001000          848#   858   1990
SW1   = 000002          866#
SW10  = 002000          847#   2425
SW11  = 004000          846#
SW12  = 010000          845#   1997  2355
SW13  = 020000          844#   2360
SW14  = 040000          843#
SW15  = 100000          842#
SW2   = 000004          865#
SW3   = 000010          864#
SW4   = 000020          863#
SW5   = 000040          862#
SW6   = 000100          861#
SW7   = 000200          860#
SW8   = 000400          859#
SW9   = 001000          858#
S110  = 001000          998#
S1200 = 003400         1003#
S134  = 001400          999#
S150  = 002000         1000#
S1800 = 004000         1004#
S19200= 007400         1011#
S2000 = 004400         1005#
S2400 = 005000         1006#
S300  = 002400         1001#
S3600 = 005400         1007#
S4800 = 006000         1008#
```

```
S50    = 000000          996#  4449  4452
S600   = 003000         1002#
S7200  = 006400         1009#
S75    = 000400          997#
S9600  = 007000         1010#
TABLE2   026062         5175  5190#
TBITVE= 000014           900#
TBUF     026214         4956* 4977  5238#
TCR0   = 000001         1015# 3076  3185
TCR1   = 000002         1016#
TCR2   = 000004         1017#
TCR3   = 000010         1018#
TCR4   = 000020         1019#
TCR5   = 000040         1020#
TCR6   = 000100         1021#
TCR7   = 000200         1022#
TDATA    026172         5227#
TD0      001422         1302# 3548  3569  3583  3597* 3810  3881  4085  4109  4110* 4215  4241  4242*
                        4243  4253  4298  4346  4347* 4453  4484  4558  4614  4686  4687* 4691
TD1      001424         1303#
TD2      001426         1304#
TD3      001430         1305#
TD4      001432         1306#
TD5      001434         1307#
TD6      001436         1308#
TD7      001440         1309#
TEIGHT   002140         1544#
TEMP     010662         2570#
TEST1    024402         4893  4903# 4923
TEST2    025040         4887  4986#
TFIVE    002146         1547#
TIE    = 040000          933# 3045  3213  3787  3804  3856  3870  3875  3967  4007  4183  4188  4307
                        4498  4567  4623  4664  4701  5013
TKVEC  = 000060           907#
TLAST  = 022772         2660  5383#
TL0    = 000000          938#
TL1    = 000400          939#
TL2    = 001000          940#
TL3    = 001400          941#
TL4    = 002000          942#
TL5    = 002400          943#
TL6    = 003000          944#
TL7    = 003400          945#
TMTBL    002102         1528# 4752* 4756  4758
TPVEC  = 000064           908#
TRAPVE= 000034            906#
TRDY   = 100000           934# 3179  3183  3412  3420  3470  3521  3563  3789  3858  4103  4235  4251
TRTVEC= 000014            901#
TR0      001442         1310# 4381  4383  4389* 4390  4405  4651  4654  4656  4660* 4734  4736  4742*
                        4744
TR1      001444         1311#
TR2      001446         1312#
TR3      001450         1313#
TR4      001452         1314#
TR5      001454         1315#
TR6      001456         1316#
```

```
TR7     001460            1317#
TSEVEN  002142            1545#
TSIX    002144            1546#
TST1    012374            1941    2648    2664    2838#
TST10   013312            3043    3072#
TST11   013450            3074    3115#
TST12   013632            3117    3169#
TST13   013750            3171    3207#
TST14   014100            3209    3247#
TST15   014170            3249    3273#
TST16   014254            3275    3307#
TST17   014450            3309    3365#
TST2    012564            2840    2881#
TST20   014606            3367    3407#
TST21   014732            3409    3447#
TST22   015270            3449    3534#
TST23   015616            3536    3616#
TST24   016150            3618    3703#
TST25   016426            3705    3767#
TST26   016736            3769    3836#
TST27   017264            3838    3912#
TST3    012650            2883    2913#
TST30   017716            3914    4020#
TST31   020104            4022    4071#
TST32   020632            4073    4201#
TST33   021214            4203    4285#
TST34   022022            4287    4440#
TST35   022532            4443    4534#
TST36   022772            4536    4590#   5383
TST37 = ****** U          4592
TST4    012742            2915    2945#
TST5    013034            2947    2977#
TST6    013126            2979    3009#
TST7    013220            3011    3041#
TTST    005010            1854*   1856*   1943#
TWOSTO= 000040             992#   4449    4452
TXSVC   021444            4304    4329#
TYPDAT  007210            2382    2402    2405#
TYPE  = 104402            1471#   1587    1656    1759    1760    1802    1815    1820    1853    1858    1876    1877    1879
                          1882    1885    2045    2146    2165    2258    2291    2383    2384    2387    2388    2390    2392
                          2394    2398    2403    2447    2449    2477    2523    2603    2639    2657    2662    2789    4891
                          4897    4937    5081    5181
TYPMSG  007100            2380    2383#
T110    002106            1531#
T1200   002120            1536#
T134    002110            1532#
T150    002112            1533#
T1800   002122            1537#
T2000   002124            1538#
T2400   002126            1539#
T300    002114            1534#
T3600   002130            1540#
T4800   002132            1541#
T50     002102            1529#
T600    002116            1535#
T7200   002134            1542#
```

```
T75     002104          1530#
T9600   002136          1543#
VECMAP  012202          2788    2796#
VEC1    024032          4821#
VEC2    024042          4823#
WCHFLG  026162          4861    4885    5223#
WRDCNT  006622          2266*   2292*   2300#
WTBS.F  007176          2397*   2400#
XBEGIN  024302          4857    4882#   5085
XBX     006766          2356    2358    2360#
XCSR    004742          1878    1910#   2393
XEOP    025476          4926    4968    4988    5034    5081#
XERR    004764          1886    1919#
XHEAD   010376          1802    2531#
XMTCNT  001376          1282#   4492*   4557*   4613*   4691
XMTLIN  001374          1281#   4447*   4457*   4459    4482
XMTSRV  023410          4493    4563    4619    4677#
XPASS   004756          1883    1916#
XSTART  023770          1125    4813#   4815
XSTATQ  010466          1809    2531#
XTCR0   025774          5168#
XTCR1   026010          5171#   5174
XTSTN   007352          2389    2434#
XVEC    004750          1880    1913#
XX    = 160210          1346#   1353#   1360#   1367#   1374#   1381#   1388#   1395#   1402#   1409#   1416#   1423#   1430#
                        1437#   1444#   1451#   1458#
YY    = 000500          1346#   1353#   1360#   1367#   1374#   1381#   1388#   1395#   1402#   1409#   1416#   1423#   1430#
                        1437#   1444#   1451#   1458#
ZZ    = 000020          1346#   1353#   1360#   1367#   1374#   1381#   1388#   1395#   1402#   1409#   1416#   1423#   1430#
                        1437#   1444#   1451#   1458#
$APTHD  001462          1328    1334#
$ASTAT= ****** U        2115    2130
$ATYC   005606          2086    2088#
$ATY1   005562          2084#
$ATY3   005570          2030    2085#
$ATY4   001300          2087#   2413
$AUTOB  001154          1154#
$BASE   001310          1228#   1622*   2626*   2674    2784*   5262
$BDADR  001142          1149#
$BDDAT  001146          1151#
$CDW1   001314          1230#   2698    2785*
$CDW2   001316          1231#
$CHARC  005556          2047*   2057*   2064    2073*   2078#
$CMTAG  001120          1137#
$CM1  = 000006          1169#   1170#   1171#   1172#   1173#   1174#   1175#
$CM2  = 000014          1169#   1170#   1171#   1172#   1173#   1174#   1175#
$CM3  = 000006          1167#   1169
$CM4  = 000004          1175#   1176#   1177#   1178#   1179#
$CPUOP  001262          1202#
$CRAP = 177777           779#   2832#   2836#   2875#   2879#   2905#   2911#   2937#   2943#   2969#   2975#   3001#   3007#
                        3033#   3039#   3065#   3070#   3107#   3113#   3160#   3167#   3197#   3205#   3241#   3245#   3267#
                        3271#   3294#   3304#   3353#   3363#   3400#   3405#   3436#   3445#   3527#   3532#   3608#   3614#
                        3693#   3701#   3760#   3765#   3829#   3834#   3902#   3910#   4012#   4018#   4058#   4069#   4193#
                        4199#   4278#   4283#   4422#   4438#   4521#   4532#   4578#   4588#
$CRLF   001231          1181#   2046    2081    2258    2383    2384    2394    2477    2639    2657
$DDW0   001320          1232#   1709    2685    2722
```

```
$DDW1   001322          1233#
$DDW10  001344          1242#
$DDW11  001346          1243#
$DDW12  001350          1244#
$DDW13  001352          1245#
$DDW14  001354          1246#
$DDW15  001356          1247#
$DDW2   001324          1234#
$DDW3   001326          1235#
$DDW4   001330          1236#
$DDW5   001332          1237#
$DDW6   001334          1238#
$DDW7   001336          1239#
$DDW8   001340          1240#
$DDW9   001342          1241#
$DEVCT  001244          1193#
$DEVM   001312          1229#  1697*  1703*  1705   1706   1818   2686   2787*
$DOAGN  004736          1888   1895   1900   1906#
$E      = 000040         779#  2840   2841#  2883   2884#  2915   2916#  2947   2948#  2979   2980#  3011   3012#
                        3043   3044#  3074   3075#  3117   3118#  3171   3172#  3209   3210#  3249   3250#  3275
                        3276#  3309   3311#  3367   3369#  3409   3410#  3449   3451#  3536   3538#  3618   3619#
                        3705   3706#  3769   3770#  3838   3839#  3914   3915#  4022   4023#  4073   4074#  4203
                        4204#  4287   4288#  4443   4444#  4536   4537#  4592   4593#
$ENDAD  004726          1114   1585   1902#  2416
$ENDCT  004712          1897#
$ENV    001254          1198#  2025   2093   2117   2410
$ENVM   001255          1199#  1589   2027   2032   2095
$EOP    004562          1872#  4592
$EOPCT  004704          1894#  1898
$ERFLG  001123          1140#  1563*  1875*  1931   1961   1963*  1986   2365*  2379   2395*  4819*  5084*
$ERMAX  001135          1146#  1986
$ERROR  006736          1104   2354#
$ERRPC  001136          1147#  1567*  1874*  1940*  2362   2364*
$ERRTB  001360          1266#
$ERTTL  001132          1144#  1566*  1921   2422*  4818*
$ETABL  001254          1197#
$ETEND  001360          1250#  1340
$FATAL  001236          1190#  2121*
$FFLG   006026          2084*  2087*  2115   2124*  2132#
$FILLC  001176          1165#  2050   2081
$FILLS  001175          1164#  2081
$GDADR  001140          1148#
$GDDAT  001144          1150#
$GET42  004716          1899#
$HD     = 000001         790    791
$HIBTS  001462          1335#
$ICNT   001124          1141#  1969*  1970   1972*  1985
$ILLUP  010012          2493   2509   2528#
$INTAG  001155          1155#
$ITEMB  001134          1145#  2370*  2412
$LF     001232          1182#  2081
$LFLG   006025          2125*  2131#
$LPADR  001126          1142#  1569*  1857*  1859   1976*  1978   1985   2427*  2429   2655*  2656*  2664*  2666
                        3320*  3321   4815*  4887*  4893*  4898
$LPERR  001130          1143#
$MADR1  001266          1215#
```

```
$MADR2  001272          1219#
$MADR3  001276          1222#
$MADR4  001302          1225#
$MAIL   001234          1188#  1336    1340    1975    2025
$MAMS1  001264          1209#
$MAMS2  001270          1217#
$MAMS3  001274          1220#
$MAMS4  001300          1223#
$MBADR  001464          1336#
$MFLG   006024          2085*  2091    2126*   2130#
$MSGAD  001250          1195#  2101*   2104
$MSGLG  001252          1196#  2106*
$MSGTY  001234          1189#  2099    2107*   2119    2123*
$MTYP1  001265          1210#
$MTYP2  001271          1218#
$MTYP3  001275          1221#
$MTYP4  001301          1224#
$MXCNT  005234          1973   1985#   2606
$N    = 000036           779#  2832    2836    2841#   2875    2879    2884#   2905    2911    2916#   2937    2943    2948#
                        2969   2975    2980#   3001    3007    3012#   3033    3039    3044#   3065    3070    3075#   3107
                        3113   3118#   3160    3167    3172#   3197    3205    3210#   3241    3245    3250#   3267    3271
                        3276#  3294    3305    3311#   3353    3363    3369#   3400    3405    3410#   3436    3445    3451#
                        3527   3532    3538#   3608    3614    3619#   3693    3701    3706#   3760    3765    3770#   3829
                        3834   3839#   3902    3910    3915#   4012    4018    4023#   4058    4069    4074#   4193    4199
                        4204#  4278    4283    4288#   4422    4438    4444#   4521    4532    4537#   4578    4588    4593#
                        5383#
$NULL   001174          1163#  2052    2081
$NWTST= 000000          2837#  2880#   2912#   2944#   2976#   3008#   3040#   3071#   3114#   3168#   3206#   3246#   3272#
                        3306#  3364#   3406#   3446#   3533#   3615#   3702#   3766#   3835#   3911#   4019#   4070#   4200#
                        4284#  4439#   4533#   4589#
$OVER   005176          1945   1948    1959    1971    1977#
$PASS   001242          1192#  1562*   1881*   1884*   1891*   1892*   1909    1918    1967    1986    4817*
$PASTM  001470          1338#
$PWRAD  010006          2526#
$PWRDN  007646          1102   1560    2493#   2521
$PWRMG  010002          2524#
$PWRUP  007720          2503   2509#
$QUES   001230          1180#  1759    2081    2165    2662
$REGAD  001200          1167#
$REG0   001202          1169#  2242*   2247    5383
$REG1   001204          1170#  2241*   2248    5383
$REG2   001206          1171#  2240*   2249
$REG3   001210          1172#  2239*   2250
$REG4   001212          1173#  2238*   2251    5383
$REG5   001214          1174#  2237*   2252    5383
$RTNAD  004740          1908#
$SAVR6  010016          2502*  2510    2511*   2512*   2530#
$SCOPE  004772          1938#
$SETUP= 000000          1890   1939
$SVLAD  005160          1956   1974#
$SVPC   = 000040        1112#  1117
$SWR    = 164000         779#   790    1179    1180    1869    1890    1901    1907    1909    1932    1933    1934    1935
                        1947   1959    1961    1962    1963    1964    1965    1977    1985    2527    2839    2882    2914
                        2946   2978    3010    3042    3073    3116    3170    3208    3248    3274    3308    3366    3408
                        3448   3535    3617    3704    3768    3837    3913    4021    4072    4202    4286    4441    4535
                        4591
```

```
$SWREG  001256          1200# 2706
$SWRMK= 000000          1935
$TESTN  001240          1191# 1975*
$TIMES  001226          1179# 1890* 1964* 1970 1973* 1985 2606* 4442*
$TKB    001166          1160# 1660 1946 2152 2158 2439 2452 2473 4940 5027
$TKS    001164          1159# 1658 1944 2150 2443 2450 2471 4938 5025
$TMP0   001216          1175# 4181* 4186* 4448* 4450 4461* 4497
$TMP1   001220          1176# 1657* 1686 1698 1699 1700* 1804* 2533 4490* 4502* 4712* 4752 4756
                        5383
$TMP2   001222          1177# 1705* 1711* 1712* 1805* 1806 2535 4710* 4713*
$TMP3   001224          1178# 4491* 4504* 4711*
$TN   = 000037           790# 2837 2839# 2880 2882# 2912 2914# 2944 2946# 2976 2978# 3008 3010#
                        3040 3042# 3071 3073# 3114 3116# 3168 3170# 3206 3208# 3246 3248# 3272
                        3274# 3306 3308# 3364 3366# 3406 3408# 3446 3448# 3533 3535# 3615 3617#
                        3702 3704# 3766 3768# 3835 3837# 3911 3913# 4019 4021# 4070 4072# 4200
                        4202# 4284 4286# 4439 4441# 4533 4535# 4589 4591#
$TPB    001172          1162# 1664* 2070* 2081 2158* 2359* 2459*
$TPFLG  001177          1166# 2019 2081
$TPS    001170          1161# 2068 2081 2156 2357 2457
$TSTM   001466          1337#
$TSTNM  001122          1139# 1568* 1931 1974* 1975 1977 1986 2436 2645 2653 2839* 2882* 2914*
                        2946* 2978* 3010* 3042* 3073* 3116* 3170* 3208* 3248* 3274* 3308* 3366* 3408*
                        3448* 3535* 3617* 3704* 3768* 3837* 3913* 4021* 4072* 4202* 4286* 4441* 4535*
                        4591* 4904* 4987*
$TYPE   005300          2019# 2112
$TYPEC  005512          2049 2056 2063 2068# 2069
$TYPEX  005560          2074 2076 2079#
$UNIT   001246          1194#
$UNITM  001472          1339#
$USWR   001260          1201#
$VECT1  001304          1226# 1635* 1647* 1648* 1649* 1650* 1651* 1652* 2675 2678 2827*
$VECT2  001306          1227#
$XTSTR  005036          1942 1950#
$Y    = 000020          1460# 1467 1469# 1471# 1473# 1475# 1477# 1479# 1481# 1483# 1485# 1487# 1489#
                        1491# 1493# 1495# 1497# 1499#
$$GET4= 000000          1901#
$40CAT= ****** U        1947
.     = 011574          1091# 1092 1095# 1099# 1112 1113# 1115# 1117# 1119# 1122# 1124# 1128# 1136#
                        1183 1285# 1287# 1300# 1324 1325# 1327# 1329# 1344# 1347# 1348# 1349# 1350#
                        1351# 1352# 1354# 1355# 1356# 1357# 1358# 1359# 1361# 1362# 1363# 1364# 1365#
                        1366# 1368# 1369# 1370# 1371# 1372# 1373# 1375# 1376# 1377# 1378# 1379# 1380#
                        1382# 1383# 1384# 1385# 1386# 1387# 1389# 1390# 1391# 1392# 1393# 1394# 1396#
                        1397# 1398# 1399# 1400# 1401# 1403# 1404# 1405# 1406# 1407# 1408# 1410# 1411#
                        1412# 1413# 1414# 1415# 1417# 1418# 1419# 1420# 1421# 1422# 1424# 1425# 1426#
                        1427# 1428# 1429# 1431# 1432# 1433# 1434# 1435# 1436# 1438# 1439# 1440# 1441#
                        1442# 1443# 1445# 1446# 1447# 1448# 1449# 1450# 1452# 1453# 1454# 1455# 1456#
                        1457# 1597 1793# 1909 1985 1986 2081 2133# 2451 2458 2472 2505 2529
                        2531# 2569# 2571# 2573# 2583 2605 2747 2750 2792 2816 3157 3551 3941
                        3944 3963 3994 3998 4088 4093 4099 4114 4133 4171 4218 4226 4229
                        4246 4252 4275 4317 4330 4343 4355 4363 4366 4369 4377 4391 4406
                        4456 4487 4561 4617 4947 4950 4953 4962 5037 5040 5043 5046 5049
                        5056 5061 5072 5239# 5383#
.ADVAN  006724          1468 2347#
.BEGIN  004474          1847#
.CNVRT  006452          1488 2259#
.CONVR  006446          1486 2258#
```

```
.DCLAS  006672            1498    2331#
.DELAY  006704            1492    2335#
.DEVIC  006652            1490    2323#
.ERRTA  027064            2375    5287#
.INSTE  006134            1476    2163#
.INSTR  006030            1474    2142#
.INST1  006050            2146#   2166
.MSG    006052            2144*   2147#
.PARAM  006154            1478    2174#
.PARMD  025516            1494    5088#
.PAWCH  025712            1496    5147#   5159
.RES05  006414            1484    2247#
.SAV05  006354            1482    2233#
.SCOPE  004772            1100    1939#   2828
.SCOP1  005236            1470    1990#
.SETFL  010500            1480    2542#   2561
.START  002150            1123    1556#   1569
.TRPSR  006630            1106    2312#
.TRPTA  002002            1466#   2317
.TYPE   005262            1472    1997#
.$ASTA= ****** U          2085    2088
.$X   = 001462            1324#   1329
```

```
COMMEN      1#    910#
ENDCOM      1#    910#
ERROR     804#   2872   2890   2903   2922   2928   2935   2954   2960   2967   2986   2992   2999   3018   3024
         3031   3050   3056   3063   3083   3091   3101   3129   3137   3145   3158   3178   3182   3195   3217
         3223   3231   3239   3258   3265   3283   3290   3343   3349   3390   3396   3425   3429   3468   3475
         3498   3502   3516   3561   3568   3591   3595   3639   3665   3671   3686   3690   3748   3752   3797
         3799   3802   3820   3822   3825   3862   3867   3873   3887   3892   3894   3942   3945   3953   3961
         3964   3972   3973   3995   3999   4004   4047   4056   4100   4108   4115   4124   4152   4155   4166
         4240   4247   4260   4265   4268   4323   4324   4331   4344   4356   4364   4367   4370   4378   4388
         4407   4506   4576   4632   4644   4653   4659   4682   4722   4725   4729   4741   4761   4917   4948
         4951   4954   4963   4976   5004   5032   5038   5041   5044   5047   5050   5057   5062   5073
ESCAPE      1#    910#
GETPRI      1#    910#
GETSWR      1#    910#
MULT        1#    910#
NEWTST      1#    910#   2837   2880   2912   2944   2976   3008   3040   3071   3114   3168   3206   3246   3272
         3306   3364   3406   3446   3533   3615   3702   3766   3835   3911   4019   4070   4200   4284   4439
         4533   4589
PASEND    779#   1873
POP         1#    910#   2127   2128   2514   2515
PRGEND    779#   1860
PRGFRT    779#    781
PUSH        1#    910#   2088   2090   2111   2495   2501
REPORT      1#    779#    910#
SC        779#   1939
SCOPE     805#   1873   2838   2881   2913   2945   2977   3009   3041   3072   3115   3169   3207   3247   3273
         3307   3365   3407   3447   3534   3616   3703   3767   3836   3912   4020   4071   4201   4285   4440
         4534   4590
SC1       779#   1979
SETPRI      1#    910#
SETUP       1#    910#
SKIP        1#    910#
SLASH       1#    910#
SPACE     910#
STARS       1#    910#   1110   1132   1183   1186   1321   1323   1330   1867   1928   2004   2083   2491   2507
         2837   2880   2912   2944   2976   3008   3040   3071   3114   3168   3206   3246   3272   3306   3364
         3406   3446   3533   3615   3702   3766   3835   3911   4019   4070   4200   4284   4439   4533   4589
SWRSU       1#    910#
TYPBIN      1#    910#
TYPDEC      1#    910#
TYPNAM      1#    910#
TYPNUM      1#    910#
TYPOCS      1#    910#
TYPOCT      1#    910#
TYPTXT      1#    910#
$BUFFE    779#   2565
$CYCLE    779#   2592
$EOP      779#   1860
$GETFL    779#   1674
$GETPA    779#   1613   1626   1639   1681   1737   1770   1828   2640
$HEADE    779#    791
$INTSE    779#   3782   3805   3851   3876   4563   4619
$JUNK     779#   1346   1353   1360   1367   1374   1381   1388   1395   1402   1409   1416   1423   1430   1437
         1444   1451
$LINEU    779#   3451   3538   3770   3839   3915   4023   4075   4205
$LVLTS    779#   3760   3829
```

```
$MRESE   779# 3451  3538  3709  3770  3827  3839  3899  3915  4023  4075  4205  4288  4464  4539
         4595
$MRR     779# 3160
$MRRW    779# 2905  2937  2969  3001  3033
$MRWD    779# 3241  3267
$MSG     779# 2531
$PARTS   779# 4521  4578
$SCOPE   779# 1922
$SETFL   779# 2537
$STAG    779# 3478  3571  3729  3982  4136
$STAGF   779#
$TCR     779# 3065  3107
$TLINE   779# 3400
$TRPDE   779# 1467  1469  1471  1473  1475  1477  1479  1481  1483  1485  1487  1489  1491  1493
         1495  1497
$TSTN    779# 2836  2879  2911  2943  2975  3007  3039  3070  3113  3167  3205  3245  3271  3305
         3363  3405  3445  3532  3614  3701  3765  3834  3910  4018  4069  4199  4283  4438  4532
         4588
$UNIBU   779# 2832
$VARIA   779# 1127
$XZ      779# 2832  2836  2875  2879  2905  2911  2937  2943  2969  2975  3001  3007  3033  3039
         3065  3070  3107  3113  3160  3167  3197  3205  3241  3245  3267  3271  3294  3304  3353
         3363  3400  3405  3436  3445  3527  3532  3608  3614  3693  3701  3760  3765  3829  3834
         3902  3910  4012  4018  4058  4069  4193  4199  4278  4283  4422  4438  4521  4532  4578
         4588
$$CMRE   1130# 1169  1170  1171  1172  1173  1174
$$CMTM   1130# 1175  1176  1177  1178
$$ESCA     1#  910#
$$NEWT     1#  910#  2837  2880  2912  2944  2976  3008  3040  3071  3114  3168  3206  3246  3272
         3306  3364  3406  3446  3533  3615  3702  3766  3835  3911  4019  4070  4200  4284  4439
         4533  4589
$$SKIP     1#  910#
.EQUAT     1#  779#   800
.HEADE     1#  779#   781
.KT11      1#
.SETUP     1#  779#
.SWRHI     1#
.$ACT1     1#  779#  1108
.$APTB     1#  779#  1184#
.$APTH     1#  779#  1319
.$APTY     1#  779#  2081
.$ASTA     1#
.$CATC     1#  779#
.$CMTA     1#  779#  1130
.$DB2D     1#
.$DB2O     1#
.$DIV      1#
.$EOP      1#  779#  1865
.$ERRO     1#  779#
.$ERRT     1#
.$MULT     1#
.$POWE     1#  779#  2489
.$RAND     1#
.$RDDE     1#
.$RDOC     1#
.$READ     1#
```

```
CZDZA-F0        MACY11 30A(1052)  26-NOV-79  11:06  PAGE 134
CZDZAF.P11    26-NOV-79 11:03          CROSS REFERENCE TABLE -- MACRO NAMES                    SEQ 0131

.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#     779#   1926
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#     779#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#     779#   2001
.$TYPO      1#
.$40CA      1#
.1170       1#


. ABS.  031110     000


 ERRORS DETECTED:  0

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