DR11C

DEVICE REGISTER TEST

MD-11-DZDRC-F

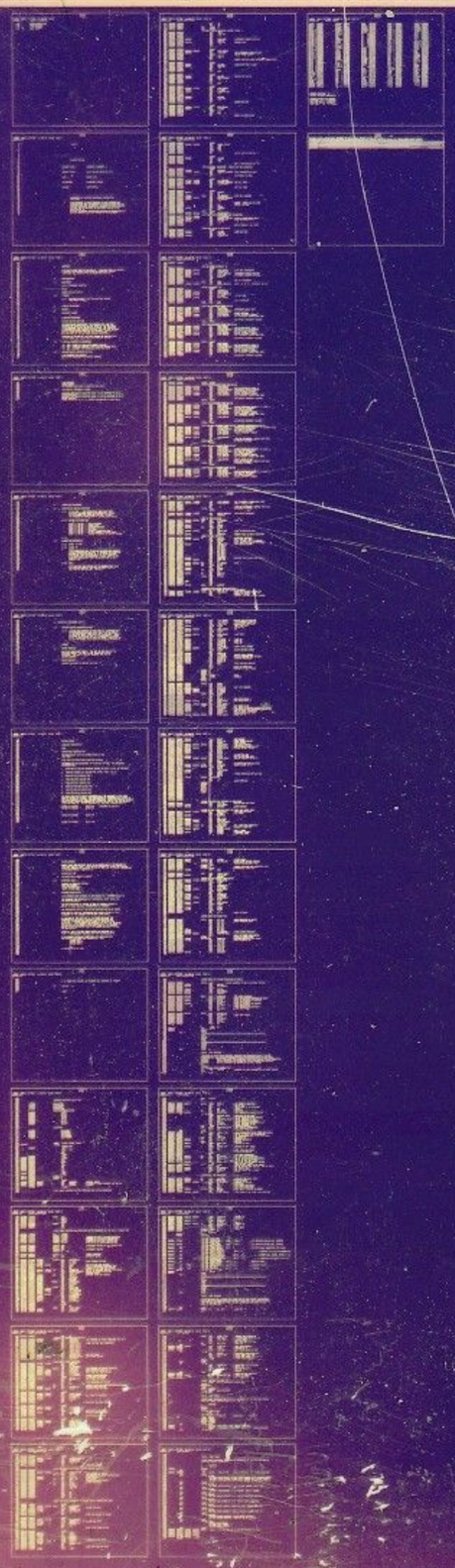
EP-DZDRC-F-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A



.MAIN. MACY11 27(732) 16-JAN-76 08:56
DZDRCF.P11 TABLE OF CONTENTS

.REPT 0

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDRC-F-D
PRODUCT NAME:	DR11C DEVICE REGISTER TEST
DATE :	MARCH 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	JOHN HITTELL

1. ABSTRACT

THIS IS A LOGIC TEST OF THE DR11C. FOR THIS TEST TO OPERATE
A SPECIAL MAINTENANCE CABLE MUST BE CONNECTED (BC08R).
THIS TEST WILL CHECK UP TO 32 SEQUENTIAL DR11C'S.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 STANDARD COMPUTER

DR11C

BC08R FOR EACH DR11C

2.2 STORAGE

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

3. LOADING PROCEDURE

3.1 METHOD

ABSOLUTE LOADER

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTING

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

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

4.2 STARTING ADDRESS OR ADDRESSES

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

4.3 PROGRAM AND/OR OPERATOR ACTION

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

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

                                4.3.1   FOR SPECIAL ENTRANCE - SA204

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

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

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

5.1.2 SWITCH SETTINGS ARE

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

5.1.3

5.2. SUBROUTINE ABSTRACTS

5.2.1 BEGIN SA 200

5.2.2 SCOPE

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

5.2.3 HALT

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

(5. OPERATING PROCEDURE CONT'D)

5.3 PROGRAM AND/OR OPERATOR ACTION

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

6. ERRORS

6.1 ERROR PRINTOUT

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

6.2 ERROR RECOVERY

DEPRESS CONTINUE TO RESTART SECTION

7. RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7.2 OPERATIONAL RESTRICTION

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

NOTE THAT THE DR11C HAS FLOATING VECTORS:

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

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

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

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

```
                        8.      MISCELLANEOUS

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

                        8.1     EXECUTION TIME

                        FOR EACH DR11C ABOUT 1 MINUTE

                        8.2     UNTESTED LOGIC

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

                        9.      PROGRAM DESCRIPTION

                        THIS PROGRAM WHEN STARTED AT 200 CHECKS THE  STANDARD DR11-C'S

                        THE PROGRAM THEN PERFORMES AN INCREMENTAL LOGIC CHECK FOR
                        THE SELECTED DR11C.

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

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

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

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

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

                        IF NO ERRORS OCCUR AND THREE DR11-C'S ARE AVAILABLE
                        AND SWITCH 9 IS DOWN THE PROGRAM WILL TYPE.
                                160000 770 *
                                157770 1000 *
                                157760 1010 *
                                /
                                ETC.
                        IF SWITCH 9 IS UP THEN
                                *
                                *
                                *
```

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

                                10.     LISTING

                                        .ENDR
```

```
                                ;GENERAL REGISTER LOGIC TEST

            177776              PSW=177776
            104000              HLT=104000
            167770              CSR=167770
            001200              STKPTR=1200
                                ;REGISTER DEFINITIONS
            000000                      R0=%0
            000001                      R1=%1
            000002                      R2=%2
            000003                      R3=%3
            000004                      R4=%4
            000005                      R5=%5
            000006                      SP=%6
            000007                      PC=%7

                                ;SWITCHES

            001000                      SW9=1000
            002000                      SW10=2000
            004000                      SW11=4000
            020000                      SW13=20000
            040000                      SW14=40000


                                .MCALL  .STYPE,.STRAP,.EQUAT
000000'                         .EQUAT
                                        .ENABLE ABS
            000000                      .=0
                                        .REPT   200
                                        .+2
                                        HALT
                                        .ENDR
            000020                      .=20
000020  004364                          .SCOPE
000022  000340                          340
000024  004624                          PFAIL
000026  000340                          340
000030  003654                          .HLT
000032  000340                          340
000034  005012                          TYP
000036  000340                          340
            000046                      .=46
000046  003636                          LOGIC
            000174                      .=174
000174  000000              ;ISPRG:     0
000176  000000              SWREG:      0
            000200                      .=200
000200  000137  001246                  JMP     @#START1        ;INITIAL START
000204  000137  001334                  JMP     @#SPEC          ;TO SELECT UNIQUE ADDRESS AND VECTOR
000210  000137  001444                  JMP     @#START         ;RESTART
            001200                      .=1200

                                ;THIS TABLE CONTAINS INITIAL REGISTER AND VECTOR ADDRESSES
```

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

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

                                SR:     177570
                                DISPLA: 177570
                                DRCSR:  167770                  ;ADDRESS OF DR11-C STATUS REGISTER
                                DROBUF: 167772                  ;ADDRESS OF DR OUTPUT BUFFER REG.
                                DRIBUF: 167774                  ;ADDRESS OF DR INPUT BUFFER REG.
                                DRBHIO: 167773                  ;HIGH BYTE OF OUTPUT BUFFER REG.

                                DRVECA: 300                     ;INTERRUPT VECTOR OF UNIT UNDER TEST
                                DRLVL:  302
                                DRVECB: 304                     ;INTERRUPT VECTOR
                                XORFLG: 0

                                COUNT:  0                       ;COUNT LOCATION
                                PL:     240                     ;PRIORITY LEVEL

001246  012706  001200         START1: MOV     #STKPTR,%6
001252  016746  176530                 MOV     6,-(SP)         ;SAVE CURRENT VECTOR
001256  016746  176522                 MOV     4,-(SP)
001262  012767  001276  176514         MOV     #1$,4           ;SET UP TIME OUT VECTOR
001270  005777  177722                 TST     @SR             ;TRY TO REFERENCE HARDWARE SWR
001274  000407                         BR      2$              ;BRANCH IF NO TIME OUT OCCURS
001276  012767  000176  177712 1$:     MOV     #SWREG,SR       ;POINT TO SOFTWARE SWITCH 1 REGISTER
001304  012767  000174  177706         MOV     #DISPRG,DISPLA  ;POINT TO DISPLAY REGISTER
001312  022626                         CMP     (SP)+,(SP)+     ;RESTORE STACK
001314  012667  176464         2$:     MOV     (SP)+,4         ;RESTORE TIME OUT VECTOR
001320  012667  176462                 MOV     (SP)+,6
001324  004767  000020                 JSR     PC,FIRST
001330  000137  001444                 JMP     @#START
001334  012706  001200         SPEC:   MOV     #STKPTR,%6
001340  004767  000004                 JSR     PC,FIRST
001344  000167  003342                 JMP     SPECO
001350  013746  000004         FIRST:  MOV     @#4,-(%6)
001354  012737  001430  000004       MOV     #XORA,@#4
001362  012737  000031  177060       MOV     #31,@#177060
001370  012637  000004               MOV     (%6)+,@#4
001374  012737  177777  001240       MOV     #-1,@#XORFLG
001402  012701  160000               MOV     #160000,R1
001406  004737  004742               JSR     PC,@#SPEC1
001412  012701  000770               MOV     #770,R1
001416  004737  004772               JSR     PC,@#SPEC2
001422  104400                       TYPE
001424  005270                       MESG1
001426  000207                       RTS     PC
001430  022626          XORA:   CMP     (%6)+,(%6)+
001432  012637  000004               MOV     (%6)+,@#4
001436  005037  001240               CLR     @#XORFLG
```

```
001442  000207                          RTS     PC
                                ;INITIALIZE ADDRESS AND VECTORS
001444  012700  001200          START:  MOV     #RCSR,R0        ;GET ADDRESS OF FIRST POSSIBLE DR11-C'S
001450  012701  001222                  MOV     #DRCSR,R1
001454  012021                          MOV     (R0)+,(R1)+     ;LOAD INITIAL TEST ADDRESSES
001456  012021                          MOV     (R0)+,(P1)+
001460  012021                          MOV     (R0)+,(R1)+
001462  012021                          MOV     (R0)+,(R1)+
001464  012021                          MOV     (R0)+,(R1)+
001466  012021                          MOV     (R0)+,(R1)+
001470  012021                          MOV     (R0)+,(R1)+
001472  012706  001200          RSTART: MOV     #STKPTR,%6      ;SET UP STACK
001476  012767  001524  002766          MOV     #BEGIN,RETURN   ;SET SCOPE RETURN
001504  005037  004470                  CLR     @#SCOPEF

                                ;DOES RESET CLEAR REGISTER?
001510  032777  001000  177500          BIT     #SW9,@SR
001516  001002                          BNE     BEGIN
001520  004737  004510                  JSR     PC,@#MOREID
001524  016705  177472          BEGIN:  MOV     DRCSR,R5        ;GET ADDRESS OF STATUS REGISTER
001530  012777  000240  176240          MOV     #240,@PSW       ;SET PRIORITY LEVEL 6
001536  012737  001574  000004          MOV     #1$,@#4         ;SET TIME OUT TRAP VECTOR
001544  012767  000010  002714          MOV     #10,ICOUNT
001552  012777  177777  177444          MOV     #-1,@DROBUF     ;PRESET OUTPUT BUFFER
001560  000005                          RESET                   ;CLEAR DATA REGISTER
001562  017700  177436                  MOV     @DROBUF,%0      ;GET RESULT OF RESET
001566  001403                          BEQ     2$
001570  104000                          HLT                     ;DATA REGISTER NOT CLEAR
001572  000401                          BR      2$
001574  104000                  1$:     HLT                     ;ERROR! TIMED OUT WHEN REFERENCING DROBUF.
001576  012706  001200          2$:     MOV     #STKPTR,SP      ;RESET STACK POINTER
001602  012737  000006  000004          MOV     #6,@#4          ;RESTORE TIME OUT TRAP

001610  000004                          SCOPE
001612  012767  004000  002646          MOV     #4000,ICOUNT
001620  012777  177777  177376          MOV     #-1,@DROBUF     ;ALL ONES TO REGISTER
001626  017700  177372                  MOV     @DROBUF,%0
001632  022700  177777                  CMP     #-1,%0
001636  001401                          BEQ     .+4
001640  104000                          HLT                     ;REG WILL NOT HOLD ONES

001642  000004                          SCOPE
001644  012767  000010  002614          MOV     #10,ICOUNT
001652  012777  177777  177344          MOV     #-1,@DROBUF
001660  000005                          RESET                   ;SET DATA TO ALL ONES
001662  005777  177340                  TST     @DRIBUF         ;SHOULD CLEAR REGISTER
001666  001401                          BEQ     .+4
001670  104000                          HLT                     ;REG FAILED TO CLEAR

001672  000004                          SCOPE
001674  012767  004000  002564          MOV     #4000,ICOUNT
001702  012777  052525  177314          MOV     #52525,@DROBUF
001710  017700  177310                  MOV     @DROBUF,%0
001714  022700  052525                  CMP     #52525,%0
001720  001401                          BEQ     .+4
001722  104000                          HLT                     ;DATA NOT=52525
```

```
                                                SCOPE
001724  000004
001726  012777  125252  177270                  MOV     #125252,@DROBUF
001734  017700  177264                          MOV     @DROBUF,%0
001740  022700  125252                          CMP     #125252,%0
001744  001401                                  BEQ     .+4
001746  104000                                  HLT                             ;DATA NOT=125252

                                        ;TEST RELIABILITY OF DR11-C OUTPUT BUFFER REGISTER
001750  000004                                  SCOPE
001752  012737  000040  004466                  MOV     #40,@#ICOUNT
001760  010502                          BUFTST: MOV     R5,R2                   ;GET ADDRESS OF DRCSR
001762  005722                                  TST     (R2)+                   ;R2=ADDRESS OF OUTPUT BUFFER REG.
001764  012703  000401                          MOV     #401,R3                 ;LOAD CONSTANT
001770  012704  000400                  1$:     MOV     #256.,R4                ;SET COUNT
001774  005000                                  CLR     R0                      ;PRESET EXPECTED RESULT
001776  005012                                  CLR     (R2)                    ;CLEAR REGISTER
002000  060300                          2$:     ADD     R3,R0
002002  060312                                  ADD     R3,(R2)
002004  021200                                  CMP     (R2),R0
002006  001401                                  BEQ     .+4
002010  104000                                  HLT
002012  005304                                  DEC     R4
002014  001371                                  BNE     2$
002016  006303                                  ASL     R3
002020  001363                                  BNE     1$

                                        ;TEST THAT BYTE REFERENCE TO DROBUF AFFECT PROPER BYTE ONLY

002022  000004                                  SCOPE
002024  012777  177777  177172          TAG:    MOV     #-1,@DROBUF
002032  105077  177166                          CLRB    @DROBUF                 ;CLEAR LOW BYTE
002036  017700  177162                          MOV     @DROBUF,%0
002042  022700  177400                          CMP     #177400,%0
002046  001401                                  BEQ     .+4
002050  104000                                  HLT                             ;BYTE LOW FAILED TO CLEAR

002052  000004                                  SCOPE
002054  012777  177777  177142                  MOV     #-1,@DROBUF
002062  105077  177142                          CLRB    @DRBHIO                 ;CLEAR HIGH BYTE
002066  017700  177132                          MOV     @DROBUF,%0
002072  022700  000377                          CMP     #377,%0
002076  001401                                  BEQ     .+4
002100  104000                                  HLT                             ;HIGH BYTE CLEAR FAILED

002102  000004                                  SCOPE
002104  005037  002154                          CLR     @#2$
002110  012704  002154                          MOV     #2$,R4
002114  005077  177104                          CLR     @DROBUF
002120  105077  177104                          CLRB    @DRBHIO
002124  105277  177100                  1$:     INCB    @DRBHIO                 ;INCREMENT HIGH BYTE
002130  105264  000001                          INCB    1(R4)
002134  027714  177064                          CMP     @DROBUF,(R4)
002140  001401                                  BEQ     .+4
002142  104000                                  HLT                             ;HIGH BYTE HAS BAD DATA
002144  105764  000001                          TSTB    1(R4)
```

```
                                                BEQ     3$
                                                BR      1$
                                        2$:     .WORD   0
                                        3$:     SCOPE
                                        ;CONTROL STATUS REGISTER (DRCSR) TESTS.
002160  005015                                  CLR     (R5)
002162  011500                                  MOV     (R5),R0
002164  001401                                  BEQ     .+4
002166  104000                                  HLT
002170  012715  000140                          MOV     #140,@R5        ;INTERRUPT ENABLE FOR A+B
002174  011500                                  MOV     @R5,%0
002176  022700  000140                          CMP     #140,%0         ;ENABLE BITS
002202  001401                                  BEQ     .+4
002204  104000                                  HLT

002206  000004                                  SCOPE
002210  012767  000010  002250                  MOV     #10,ICOUNT
002216  012715  000140                          MOV     #140,@R5        ;SET INTERRUPT ENABLE FLOPS
002222  000005                                  RESET                   ;CLEAR THOSE FLOPS
002224  011500                                  MOV     @R5,%0
002226  001401                                  BEQ     .+4
002230  104000                                  HLT                     ;RESET DID NOT CLEAR INTERRUPT ENABLE BITS

002232  000004                                  SCOPE
002234  052715  000001                          BIS     #1,@R5          ;SHOULD SET REQ A ALSO
002240  021527  000201                          CMP     @R5,#201
002244  001401                                  BEQ     .+4
002246  104000                                  HLT
002250  005015                                  CLR     @R5

002252  000004                                  SCOPE
002254  052715  000002                          BIS     #2,@R5          ;SHOULD SET REQ B
002260  021527  100002                          CMP     @R5,#100002
002264  001401                                  BEQ     .+4
002266  104000                                  HLT
002270  005015                                  CLR     @R5

002272  000004                                  SCOPE
002274  052737  000340  177776                  BIS     #340,@#PSW
002302  052715  177777                          BIS     #-1,@R5
002306  022715  100343                          CMP     #100343,(R5)
002312  001401                                  BEQ     .+4
002314  104000                                  HLT
002316  042715  000003                          BIC     #3,@R5
002322  022715  000140                          CMP     #140,@R5
002326  001401                                  BEQ     .+4
002330  104000                                  HLT             ;WRONG BITS SET

002332  000004                                  SCOPE
002334  012737  000340  177776                  MOV     #340,@#PSW
002342  052715  000003                          BIS     #3,@R5
002346  000005                                  RESET
002350  005715                                  TST     @R5
002352  001401                                  BEQ     .+4
002354  104000                                  HLT             ;RESET DID NOT CLEAR
```

```
002356  000004                          SCOPE
002360  012767  004000  002100          MOV     #4000,ICOUNT
002366  005015                          CLR     @R5
002370  005215                          INC     @R5
002372  105715                          TSTB    @R5
002374  100401                          BMI     .+4
002376  104000                          HLT                             ;BIT 0 DID NOT SET BIT 7

002400  000004                          SCOPE
002402  012715  000002                  MOV     #2,@R5
002406  005715                          TST     @R5
002410  100401                          BMI     .+4
002412  104000                          HLT                             ;BIT 1 DID NOT SET BIT 15

                                ;TEST EXTERNAL TRANSFERS - CABLE MUST BE CONNECTED
002414  000004                          SCOPE
002416  005077  176602                  CLR     @DROBUF
002422  017777  176600  176574          MOV     @DRIBUF,@DROBUF ;TEST TRANSFER OF ZERO
002430  001401                          BEQ     .+4
002432  104000                          HLT                     ;NOT EQUAL TO ZERO

002434  000004                          SCOPE
002436  012777  177777  176560          MOV     #-1,@DROBUF
002444  017777  176556  176552          MOV     @DRIBUF,@DROBUF ;MOV ALL ONES
002452  022777  177777  176544          CMP     #-1,@DROBUF
002460  001401                          BEQ     .+4
002462  104000                          HLT                     ;NOT ALL ONES

002464  000004                          SCOPE
002466  005067  001774                  CLR     ICOUNT
002472  005000                          CLR     %0
002474  010077  176524          TST6:   MOV     %0,@DROBUF      ;TEST ALL NUMBERS
002500  017777  176522  176516          MOV     @DRIBUF,@DROBUF
002506  020077  176512                  CMP     %0,@DROBUF
002512  001401                          BEQ     .+4
002514  104000                          HLT                     ;ERROR - CHECK %0 FOR GOOD
002516  005200                          INC     %0              ;DROBUF FOR BAD
002520  001403                          BEQ     TST9
002522  005077  176476                  CLR     @DROBUF
002526  000762                          BR      TST6
002530  000004                  TST9:   SCOPE
002532  012737  000005  004466          MOV     #5,@#ICOUNT
                                ;TEST DATA FROM BLACK BOX (NOT CONNECTED)
002540  012777  177777  176456          MOV     #-1,@DROBUF
002546  017777  176452  176452          MOV     @DROBUF,@DRIBUF ;STATIC LINES EQUAL ONES
002554  017700  176446                  MOV     @DRIBUF,%0      ;DATA REGISTER TO %0
002560  022700  177777                  CMP     #-1,%0
002564  001401                          BEQ     .+4
002566  104000                          HLT                     ;REG 0 SHOULD = ALL ONES

                                ;READY BIT IS IN A ONE STATE
002570  000004                          SCOPE
002572  012715  000003                  MOV     #3,@R5          ;CSR0 AND CSR1
002576  011500                          MOV     (R5),R0
002600  022700  100203                  CMP     #100203,R0
002604  001401                          BEQ     .+4
```

```
                                                HLT
                                ;CAN WE RAISE INTERUPT "A"
                                                SCOPE
002610  000004
002612  052737  000340  177776                  BIS     #340,@#PSW      ;LOCK OUT INTERRUPTS
002620  012706  001200                          MOV     #STKPTR,%6
002624  012777  002646  176400                  MOV     #TST4,@DRVECA   ;INTERRUPT RETURN POINTER
002632  012715  000101                          MOV     #101,@R5        ;INTERRUPT ENABLE AND CSR0
002636  005037  177776                          CLR     @#PSW
002642  000240                                  NOP
002644  104000                                  HLT                     ;NO "A" INTERRUPT
002646  005015                  TST4:   CLR     @R5
002650  016777  176360  176354                  MOV     DRLVL,@DRVECA   ;MOVE .+2 TO "A" INTERRUPT VECTOR

                                ;RAISE INTERRUPT "B"
002656  000004                                 SCOPE
002660  012706  001200                          MOV     #STKPTR,%6
002664  052737  000340  177776                  BIS     #340,@#PSW
002672  012777  002716  176336                  MOV     #TST5,@DRVECB
002700  012715  000042                          MOV     #42,@R5         ;IE AND CSR1
002704  042737  000377  177776                  BIC     #377,@#PSW
002712  000240                                  NOP
002714  104000                                  HLT                     ;NO B INTERRUPT
002716  005015                  TST5:   CLR     @R5

                                ;TEST FOR INTERRUPT FROM DEVICE
002720  016777  176320  176306                  MOV     PL,@DRLVL
002726  042737  000340  177776                  BIC     #340,@#PSW      ;PROCESSOR LEVEL ZERO
002734  012777  002766  176270                  MOV     #TINT1,@DRVECA
002742  012706  001200                          MOV     #STKPTR,%6      ;STACK POINTER
002746  042777  000100  176246                  BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
002754  052777  000101  176240                  BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
002762  000240                                  NOP
002764  104000                                  HLT                     ;NO DEVICE INTERRUPT OCCURED
002766  000004                  TINT1:  SCOPE

                                ;TEST FOR INTERRUPT FROM THE DEVICE
002770  042737  000340  177776                  BIC     #340,@#PSW
002776  052737  000040  177776                  BIS     #040,@#PSW      ;SET TO PRIORITY LEVEL 1
003004  012777  003036  176220                  MOV     #TINT2,@DRVECA  ;INTERRUPT VECTOR ADDRESS
003012  012706  001200                          MOV     #STKPTR,%6      ;SET UP STACK POINTER
003016  042777  000100  176176                  BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
003024  052777  000101  176170                  BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
003032  000240                                  NOP
003034  104000                                  HLT                     ;NO DEVICE INTERRUPT OCCURED

003036  000004                  TINT2:  SCOPE
003040  042737  000340  177776                  BIC     #340,@#PSW
003046  052737  000100  177776                  BIS     #100,@#PSW      ;SET TO PRIORITY LEVEL 2
003054  012777  003106  176150                  MOV     #TINT3,@DRVECA  ;INTERRUPT VECTOR ADDRESS
003062  012706  001200                          MOV     #STKPTR,%6      ;SET UP STACK POINTER
003066  042777  000100  176126                  BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
003074  052777  000101  176120                  BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
003102  000240                                  NOP
003104  104000                                  HLT                     ;NO DEVICE INTERRUPT OCCURED
```

```
                                        TINT3:  SCOPE
                                        ;TEST FOR INTERRUPT FROM THE DEVICE
003110  042737  000340  177776                BIC     #340,@#PSW
003116  052737  000140  177776                BIS     #140,@#PSW      ;SET TO PRIORITY LEVEL 3
003124  012777  003156  176100                MOV     #TINT4,@DRVECA  ;INTERRUPT VECTOR ADDRESS
003132  012706  001200                        MOV     #STKPTR,%6      ;SET UP STACK POINTER
003136  042777  000100  176056                BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
003144  052777  000101  176050                BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
003152  000240                                NOP
003154  104000                                HLT                     ;NO DEVICE INTERRUPT OCCURED
003156  000004                        TINT4:  SCOPE

                                        ;TEST FOR INTERRUPT FROM DEVICE
003160  042737  000340  177776                BIC     #340,@#PSW
003166  052737  000200  177776                BIS     #200,@#PSW      ;RAISE PROCESSOR PRIORITY TO LEVEL 4
003174  012777  003236  176030                MOV     #TINT5,@DRVECA  ;IN CASE OF INTERRUPT
003202  012706  001200                        MOV     #STKPTR,%6      ;SET STACK POINTER
003206  042777  000100  176006                BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
003214  052777  000101  176000                BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE AND CSR0
003222  000240                                NOP                     ;LET INTERRUPT OCCUR
003224  042777  000100  175770                BIC     #100,@DRCSR
003232  000240                                NOP
003234  104000                                HLT                     ;NO DEVICE INTERRUPT OCCURED
003236  000004                        TINT5:  SCOPE

                                        ;TEST FOR NO INTERRUPT FROM DEVICE (HIGHEST PROCESSOR PRIORITY)
003240  052737  000340  177776                BIS     #340,@#PSW      ;RAISE PROCESSOR PRIORITY TO HIGHEST LEVEL
003246  012777  003306  175756                MOV     #TINT6,@DRVECA  ;IN CASE OF INTERRUPT
003254  012706  001200                        MOV     #STKPTR,%6      ;SET STACK POINTER
003260  042777  000100  175734                BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
003266  052777  000101  175726                BIS     #101,@DRCSR
003274  000240                                NOP
003276  042777  000100  175716                BIC     #100,@DRCSR
003304  000401                                BR      .+4             ;WITH NO INTERRUPT, BRANCH OVER HALT
003306  104000                        TINT6:  HLT                     ;INTERRUPT OCCURED
003310  000004                                SCOPE

                                        ;TEST FOR NO INTERRUPT FROM DEVICE
003312  042737  000340  177776                BIC     #340,@#PSW
003320  052737  000240  177776                BIS     #240,@#PSW      ;RAISE PROCESSOR PRIORITY TO LEVEL 5
003326  012777  003366  175676                MOV     #TINT7,@DRVECA  ;IN CASE OF INTERRUPT
003334  012706  001200                        MOV     #STKPTR,%6      ;SET STACK POINTER
003340  042777  000100  175654                BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
003346  052777  000101  175646                BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE AND CSR0
003354  000240                                NOP
003356  042777  000100  175636                BIC     #100,@DRCSR     ;DON'T LEAVE IT SET
003364  000401                                BR      .+4             ;WITH NO INTERRUPT, BRANCH OVER HALT
003366  104000                        TINT7:  HLT                     ;INTERRUPT OCCURED
003370  000004                                SCOPE

                                        ;TEST FOR NO INTERRUPT FROM DEVICE
003372  042737  000340  177776                BIC     #340,@#PSW
003400  052737  000300  177776                BIS     #300,@#PSW      ;RAISE PROCESSOR PRIORITY TO LEVEL 6
003406  012777  003446  175616                MOV     #TINT8,@DRVECA  ;IN CASE OF INTERRUPT
003414  012706  001200                        MOV     #STKPTR,%6      ;SET STACK POINTER
003420  042777  000100  175574                BIC     #100,@DRCSR     ;CLEAR INTERRUPT ENABLE
```

```
  740  003426  052777  000101  175566          BIS     #101,@DRCSR     ;SET INTERRUPT ENABLE-AND CSR0
  741  003434  042777  000100  175560          BIC     #100,@DRCSR     ;DON'T LEAVE IT SET
  742  003442  000240                          NOP
  743  003444  000401                          BR      .+4             ;WITH NO INTERRUPT, BRANCH OVER HALT
  744  003446  104000                  TINTB:  HLT                     ;INTERRUPT OCCURED
  745  003450  000004                          SCOPE
  746
  747  003452  016777  175556  175552          MOV     DRLVL,@DRVECA   ;FOR FALSE INTERRUPT
  748  003460  005077  175546                  CLR     @DRVECA
  749
  750                           ;END OF TEST ROUTINE
  751  003464  012777  000052  000354  END:    MOV     #'*,@TDBR       ;TYPE '*'
  752  003472  105777  000352          1$:     TSTB    @TCSR
  753  003476  100375                          BPL     1$
  754  003500  005077  000342                  CLR     @TDBR
  755  003504  105777  000340          2$:     TSTB    @TCSR
  756  003510  100375                          BPL     2$
  757  003512  032777  002000  175476          BIT     #SW10,@SR       ;LOOP ON SELECTED DR?
  758  003520  001402                          BEQ     4$
  759  003522  000137  001472                  JMP     @#RSTART        ;REPEAT TEST ON DR11C SELECTED
  760                           ;STEP TO NEXT DR11-C
  761  003526  012700  000010          4$:     MOV     #10,R0          ;STEPPING CONSTANT
  762  003532  012737  003602  000004          MOV     #5$,@#4         ;SET TIME OUT TRAP
  763  003540  160005                          SUB     R0,R5           ;STEP TO NEXT DR11-C ADDRESS
  764  003542  005715                          TST     (R5)            ;WILL TIME OUT IF NOT AVAILABLE
  765  003544  012705  001222                  MOV     #DRCSR,R5       ;SET TABLE POINTER
  766  003550  160025                          SUB     R0,(R5)+
  767  003552  160025                          SUB     R0,(R5)+
  768  003554  160025                          SUB     R0,(R5)+
  769  003556  160025                          SUB     R0,(R5)+
  770  003560  060025                          ADD     R0,(R5)+
  771  003562  060025                          ADD     R0,(R5)+
  772  003564  060025                          ADD     R0,(R5)+
  773  003566  000137  001472                  JMP     @#RSTART        ;RESTART TEST USING NEXT DR11-C
  774  003572  032777  001000  175416          BIT     #SW9,@SR
  775  003600  001013                          BNE     8$
  776  003602  012777  000057  000236  5$:     MOV     #'/,@TDBR       ;TYPE '/'
  777  003610  105777  000234          6$:     TSTB    @TCSR
  778  003614  100375                          BPL     6$
  779  003616  005077  000224                  CLR     @TDBR
  780  003622  105777  000222          7$:     TSTB    @TCSR
  781  003626  100375                          BPL     7$
  782  003630  013700  000042          8$:     MOV     @#42,R0
  783  003634  001405                          BEQ     END1
  784  003636  004710                  LOGIC:  JSR     PC,(R0)
  785  003640  000240                          NOP
  786  003642  000240                          NOP
  787  003644  000240                          NOP
  788  003646  000240                          NOP
  789  003650  000137  001444          END1:   JMP     @#START
  790
  791                           ;ENTERED WITH SYSTEM TRAP CALL(HLT)
  792                           ;PRINT OUT THE ERROR PC AND STATUS REGISTER
  793  003654  037727  175336  020000  .HLT:   BIT     @SR,#SW13       ;TEST FOR INHIBIT PRINT OUT
  794  003662  001401                          BEQ     .+4             ;BRANCH TO PRINT
  795  003664  000002                          RTI                     ;INHIBIT, RETURN TO MAIN STREAM
```

```
  796  003666  012667  000160                MOV     (6)+,SAVPC      ;PC OF FAILING ROUTINE
  797  003672  012667  000156                MOV     (6)+,SAVCC      ;CC OF ERROR CONDITION
  798  003676  024646                        CMP     -(6),-(6)       ;REPOSITION THE STACK
  799  003700  105777  000144                TSTB    @TCSR           ;WAIT FOR FLAG
  800  003704  100375                        BPL     .-4             ;IF NOT UP.
  801  003706  012777  000215  000132        MOV     #215,@TDBR      ;CR
  802  003714  105777  000130                TSTB    @TCSR
  803  003720  100375                        BPL     .-4
  804  003722  012777  000212  000116        MOV     #212,@TDBR      ;LINE FEED
  805  003730  105777  000114                TSTB    @TCSR
  806  003734  100375                        BPL     .-4
  807  003736  010267  000076                MOV     %2,SAVR2        ;SAVE R2
  808  003742  010367  000074                MOV     %3,SAVR3        ;SAVE R3
  809  003746  010467  000072                MOV     %4,SAVR4        ;SAVE R4
  810  003752  016702  000074                MOV     SAVPC,%2
  811  003756  004767  000074                JSR     %7,PRTAB        ;PRINT OCTAL NUMBER
  812  003762  012777  000240  000056        MOV     #240,@TDBR
  813  003770  105777  000054                TSTB    @TCSR           ;SPACE BETWEEN WORDS
  814  003774  100375                        BPL     .-4
  815  003776  016702  000052                MOV     SAVCC,%2
  816  004002  004767  000050                JSR     %7,PRTAB        ;PRINT OCTAL NUMBER
  817  004006  004767  000476                JSR     %7,MOREID       ;DEVICE ADDRESS AND VECTORS
  818  004012  016702  000022                MOV     SAVR2,%2        ;RESTORE REGISTERS
  819  004016  016703  000020                MOV     SAVR3,%3
  820  004022  016704  000016                MOV     SAVR4,%4
  821  004026  005777  175164                TST     @SR             ;TEST FOR HALT SWITCH
  822  004032  100001                        BPL     .+4
  823  004034  000000                        HALT                    ;HALT ON ERROR SET
  824  004036  000002                        RTI                     ;RETURN TO MAIN STREAM
  825  004040  000000              SAVR2:    0
  826  004042  000000              SAVR3:    0
  827  004044  000000              SAVR4:    0
  828  004046  177566              TDBR:     177566                  ;DATA
  829  004050  177564              TCSR:     177564                  ;STATUS
  830  004052  000000              SAVPC:    0
  831  004054  000000              SAVCC:    0
  832
  833  004056  005067  000252      PRTAB:    CLR     BINCT
  834  004062  005067  000244                CLR     WGTCT
  835  004066  012704  004340                MOV     #LIST,%4        ;GET LIST ADDRESS
  836  004072  012767  000005  000236        MOV     #5,ASCNT
  837  004100  012767  000007  000220        MOV     #7,SEVEN
  838  004106  012767  000001  000214        MOV     #1,DECML
  839  004114  105777  177730      WAIT1:    TSTB    @TCSR
  840  004120  100375                        BPL     WAIT1
  841  004122  005702                        TST     %2
  842  004124  100404                        BMI     MINUS           ;NEG SIGN PRINT 1
  843  004126  012777  000260  177712        MOV     #260,@TDBR      ;POS SIGN PRINT 0
  844  004134  000403                        BR      STAR
  845  004136  012777  000261  177702  MINUS:MOV     #261,@TDBR
  846  004144  016703  000156      STAR:     MOV     SEVEN,%3        ;PUT MASK IN R3
  847  004150  010267  000150                MOV     %2,TOODLE       ;GET READY TO DOODLE NUMBER IN TOODLE
  848  004154  005167  000144                COM     TOODLE          ;COMPENSATES FOR COMPLEMENT DURING BIC
  849  004160  046703  000140                BIC     TOODLE,%3       ;AND IN OCTAL CHARACTER
  850  004164  001410                        BEQ     WRTOC           ;ZERO, WRITE 0 IN LIST
  851  004166  066767  000136  000136  MKNUM:ADD     DECML,WGTCT     ;COUNT UP TO
```

```
  852  004174  005267  000134                    INC     BINCT           ;AND RECORD
  853  004200  026703  000126                    CMP     WGTCT,%3        ;SAME BINARY WEIGHT
  854  004204  001370                            BNE     MKNUM           ;KEEP COUNTN
  855  004206  062767  000260  000120  WRTOC:    ADD     #260,BINCT      ;ADD ASCII PREFIX
  856  004214  016724  000114                    MOV     BINCT,(4)+      ;WRITE ASCII CHAR IN LIST
  857  004220  066767  000102  000102            ADD     SEVEN,DECML     ;EXPAND BINARY WEIGHT
  858  004226  005067  000100                    CLR     WGTCT
  859  004232  005067  000076                    CLR     BINCT
  860  004236  005367  000074                    DEC     ASCNT
  861  004242  001410                            BEQ     XLIST           ;5 CHAR IN LIST
  862  004244  012703  000003                    MOV     #3,%3           ;SET X3 FOR ADD LOOP
  863  004250  066767  000052  000050  MOADD:    ADD     SEVEN,SEVEN     ;MAKING SEVENTY BY SEVEN
  864  004256  005303                            DEC     %3
  865  004260  001373                            BNE     MOADD
  866  004262  000730                            BR      STAR            ;NX SEVEN SET GET NX OCTAL
  867  004264  012767  000005  000044  XLIST:    MOV     #5,ASCNT        ;SEND 5 CHAR TO TTY
  868  004272  105777  177552          WAIT2:    TSTB    @TCSR
  869  004276  100375                            BPL     WAIT2
  870  004300  014477  177542                    MOV     -(4),@TDBR
  871  004304  005367  000026                    DEC     ASCNT
  872  004310  001401                            BEQ     HOFHM           ;FINISH PRINTING GET NXT NUM
  873  004312  000767                            BR      WAIT2
  874  004314  105777  177530          HOFHM:    TSTB @TCSR
  875  004320  100375                            BPL     .-4
  876  004322  000207                            RTS     %7              ;HEAD FOR HOME
  877  004324  000000                  TOODLE:   0
  878  004326  000000                  SEVEN:    0
  879  004330  000000                  DECML:    0
  880  004332  000000                  WGTCT:    0
  881  004334  000000                  BINCT:    0
  882  004336  000000                  ASCNT:    0
  883  004340  000000                  LIST:     0
  884  004342  000000                            0
  885  004344  000000                            0
  886  004346  000000                            0
  887  004350  000000                            0
  888                                    ;SCOPE LOOP ROUTINE ENTERED BY USER TRAP
  889  004352  022606                  SCOPEB:   CMP     (6)+,%6         ;REPOSITION THE STACK
  890  004354  012637  177776                    MOV     (6)+,@#PSW
  891  004360  000177  000106                    JMP     @RETURN         ;SCOPE RETURN
  892
  893                                    ;SCOPE OR/AND ITERATION LOOP FOR EACH TEST 4000 TIMES
  894  004364  032777  040000  174624  .SCOPE:   BIT     #SW14,@SR       ;TEST SR FOR SCOPE
  895  004372  001367                            BNE     SCOPEB          ;YES SCOPE
  896  004374  005737  001240                    TST     @#XORFLG
  897  004400  100012                            BPL     1$
  898  004402  013746  000004                    MOV     @#4,-(%6)
  899  004406  012737  004500  000004            MOV     #XOR,@#4
  900  004414  012737  000031  177060            MOV     #31,@#177060
  901  004422  012637  000004                    MOV     (%6)+,@#4
  902  004426  032777  004000  174562  1$:       BIT     #SW11,@SR       ;NO - TEST FOR ITERATION
  903  004434  001011                            BNE     SCOPEA          ;INHIBIT ITERATION
  904  004436  026767  000026  000022            CMP     SCOPEF,ICOUNT
  905  004444  001403                            BEQ     SCOPEG          ;EXIT - DONE
  906  004446  005267  000016                    INC     SCOPEF          ;INCREMENT COUNT
  907  004452  000737                            BR      SCOPEB          ;LOOP SOME MORE
```

```
  908  004454  005067  000010          SCOPEG: CLR     SCOPEF          ;CLEAR COUNT
  909  004460  011667  000006          SCOPEA: MOV     @X6,RETURN      ;SAVE SCOPE RETURN POINTER
  910  004464  000002                          RTI                     ;RETURN INLINE-NEXT TEST
  911  004466  004000                  ICOUNT: 4000
  912  004470  000000                  SCOPEF: 0                       ;COUNT LOCATION FOR ITERATION LOOP
  913  004472  001524                  RETURN: BEGIN                   ;ADDRESS OF LAST TEST
  914  004474  000167  173500                  JMP     200
  915
  916  004500  022626                  XOR:    CMP     (X6)+,(X6)+
  917  004502  012637  000004                  MOV     (X6)+,@#4
  918  004506  000721                          BR      SCOPEB
  919                                   ;PRINT DEVICE ADDRESS AND VECTOR
  920  004510  012777  000240  177330  MOREID: MOV     #240,@TDBR
  921  004516  105777  177326                  TSTB    @TCSR
  922  004522  100375                          BPL     .-4
  923  004524  013702  001222                  MOV     @#DRCSR,X2
  924  004530  004767  177322                  JSR     X7,PRTAB
  925  004534  012777  000240  177304          MOV     #240,@TDBR
  926  004542  105777  177302                  TSTB    @TCSR
  927  004546  100375                          BPL     .-4
  928  004550  016702  174456                  MOV     DRVECA,X2
  929  004554  004767  177276                  JSR     X7,PRTAB
  930  004560  012777  000215  177260          MOV     #215,@TDBR
  931  004566  105777  177256                  TSTB    @TCSR
  932  004572  100375                          BPL     .-4
  933  004574  012777  000212  177244          MOV     #212,@TDBR
  934  004602  105777  177242                  TSTB    @TCSR
  935  004606  100375                          BPL     .-4
  936  004610  005077  177232                  CLR     @TDBR
  937  004614  105777  177230                  TSTB    @TCSR
  938  004620  100375                          BPL     .-4
  939  004622  000207                          RTS     X7              ;BACK TO PRINT
  940
  941                                   ;ENTER HERE FOR POWER FAIL
  942
  943  004624  010046                  PFAIL:  MOV     X0,-(6)         ;SAVE REGISTER OR STACK
  944  004626  010146                          MOV     X1,-(6)         ;WHEN POWERING DOWN
  945  004630  010246                          MOV     X2,-(6)
  946  004632  010346                          MOV     X3,-(6)
  947  004634  010446                          MOV     X4,-(6)
  948  004636  010546                          MOV     X5,-(6)
  949  004640  016746  173160                  MOV     24,-(6)
  950  004644  010637  004660                  MOV     X6,@#SAVR6      ;STORE STACK POSITION
  951  004650  012737  004662  000024          MOV     #RESTAR,@#24
  952  004656  000000                          HALT                    ;HALT ON POWER DOWN NORMAL
  953  004660  000000                  SAVR6:  0                       ;STACK IS SAVED HERE
  954  004662  016706  177772          RESTAR: MOV     SAVR6,X6        ;RESTORE REGISTER OFF STACK
  955  004666  012667  173132                  MOV     (6)+,24         ;WHEN POWERING UP
  956  004672  012605                          MOV     (6)+,X5
  957  004674  012604                          MOV     (6)+,X4
  958  004676  012603                          MOV     (6)+,X3
  959  004700  012602                          MOV     (6)+,X2
  960  004702  012601                          MOV     (6)+,X1
  961  004704  012600                          MOV     (6)+,X0
  962  004706  000137  001472                  JMP     @#RSTART
  963
```

```
                                ;ENTER HERE FOR UNIQUE SELECTION OF DR11C
                                
004712  000000                  SPEC0:  HALT            ;PLACE ADDRESS OF DR11-C CONTROL STATUS
004714  017701  174276                  MOV     @SR,R1
004720  004737  004742                  JSR     PC,@#SPEC1
004724  000000                          HALT
004726  017701  174264                  MOV     @SR,R1
004732  004737  004772                  JSR     PC,@#SPEC2
004736  000137  001444                  JMP     @#START
                                
004742  012700  001200          SPEC1:  MOV     #RCSR,R0        ;SET TABLE ADDRESS
004746  010120                          MOV     R1,(R0)+        ;LOAD INTO TABLE STARTING AT RCSR
004750  062701  000002                  ADD     #2,R1           ;STEP TO ADDRESS OF DROUTBUF
004754  010120                          MOV     R1,(R0)+        ;LOAD INTO TABLE
004756  062701  000002                  ADD     #2,R1           ;STEP TO ADDRESS OF DRINBUF
004762  010120                          MOV     R1,(R0)+        ;LOAD INTO TABLE
004764  005301                          DEC     R1              ;FORM ADDRESS OF DROUTBUF+1
004766  010120                          MOV     R1,(R0)+        ;LOAD INTO TABLE
004770  000207                          RTS     PC
                                
004772  012700  001210          SPEC2:  MOV     #RCSR1,R0
004776  010120                          MOV     R1,(R0)+        ;LOAD INTO TABLE
005000  005721                          TST     (R1)+
005002  010120                          MOV     R1,(R0)+
005004  005721                          TST     (R1)+
005006  010120                          MOV     R1,(R0)+
005010  000207                          RTS     PC
                                
                                        .LIST   ME
005012                          TYP:    .STYPE
005012                          STARS
        001                             .IF B
                                ;#######################################################################
                                        .IFF
                                        .NLIST
                                        .REPT
                                        .LIST
                                ;#######################################################################
                                        .NLIST
                                        .ENDR
                                        .LIST
        000                             .ENDC
                                
                                        .SBTTL  TYPE ROUTINE
                                
                                ;#ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;#THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;#NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;#NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;#NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;#
                                ;#CALL:
                                ;#1) USING A TRAP INSTRUCTION
                                ;#      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;#OR
                                ;#      TYPE
```

```
 1020                                   ;*      MESADR
 1021                                   ;*
 1022
 1023  005012  105767  000217  $TYPE:   TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 1024  005016  100002                   BPL     1$              ;;BR IF YES
 1025  005020  000000                   HALT                    ;;HALT HERE IF NO TERMINAL
 1026  005022  000407                   BR      3$              ;;LEAVE
 1027  005024  010046          1$:      MOV     R0,-(SP)        ;;SAVE R0
 1028  005026  017600  000002           MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 1029              001                  .IF DF $MAIL
 1030                                   CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 1031                                   BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 1032                                   BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 1033                                   BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 1034                                   MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 1035                                   JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 1036                          61$:     .WORD   0               ;;MESSAGE ADDRESS
 1037                          62$:     BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 1038                                   BNE     60$             ;;YES,SKIP TYPE OUT
 1039              000                  .ENDC
 1040  005032  112046          2$:      MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 1041  005034  001005                   BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 1042  005036  005726                   TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 1043  005040  012600          60$:     MOV     (SP)+,R0        ;;RESTORE R0
 1044  005042  062716  000002  3$:      ADD     #2,(SP)         ;;ADJUST RETURN PC
 1045  005046  000002                   RTI                     ;;RETURN
 1046  005050  122716  000011  4$:      CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 1047  005054  001426                   BEQ     8$
 1048  005056  122716  000200           CMPB    #ICRLF,(SP)     ;;BRANCH IF NOT <CRLF>
 1049  005062  001004                   BNE     5$
 1050  005064  005726                   TST     (SP)+           ;;POP  <CR><LF> EQUIV
 1051  005066  104400                   TYPE                    ;;TYPE A CR AND LF
 1052  005070  005237                   $CRLF
 1053  005072  000757                   BR      2$              ;;GET NEXT CHARACTER
 1054  005074  004767  000056  5$:      JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 1055  005100  126726  000130  6$:      CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 1056  005104  001352                   BNE     2$              ;;IF NO GO GET NEXT CHAR.
 1057  005106  016746  000120           MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
 1058                                                           ;;AND THE NULL CHAR.
 1059  005112  105366  000001  7$:      DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 1060  005116  002770                   BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 1061  005120  004767  000032           JSR     PC,$TYPEC       ;;GO TYPE A NULL
 1062  005124  105367  000072           DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 1063  005130  000770                   BR      7$              ;;LOOP
 1064
 1065                                   ;HORIZONTAL TAB PROCESSOR
 1066
 1067  005132  112716  000040  8$:      MOVB    #40,(SP)        ;;REPLACE TAB WITH SPACE
 1068  005136  004767  000014  9$:      JSR     PC,$TYPEC       ;;TYPE A SPACE
 1069  005142  132767  000007  000052   BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 1070  005150  001372                   BNE     9$              ;;TAB STOP
 1071  005152  005726                   TST     (SP)+           ;;POP SPACE OFF STACK
 1072  005154  000726                   BR      2$              ;;GET NEXT CHARACTER
 1073  005156  105777  000044  $TYPEC:  TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 1074  005162  100375                   BPL     $TYPEC
 1075  005164  116677  000002  000036   MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
```

```
1076  005172  122766  000015  000002          CMPB    #15,2(SP)       ;;BRANCH IF
1077  005200  001003                          BNE     1$              ;;NOT <CR>
1078  005202  105067  000014                  CLRB    $CHARCNT        ;;
1079  005206  000406                          BR      $TYPEX          ;;EXIT
1080  005210  122766  000012  000002  1$:     CMPB    #12,2(SP)       ;;BRANCH IF
1081  005216  002002                          BGE     $TYPEX          ;;<LF>
1082  005220  105227                          INCB    (PC)+           ;;INC SPACE
1083  005222  000000                  $CHARCNT:.WORD  0               ;;COUNT
1084  005224  000207                  $TYPEX: RTS     PC
1085                                   ;       EQUATES
1086          000011                   TAT=11
1087          000200                   TCRLF=200
1088
1089                                   .IIF NDF $TPS,$TPS:
1090  005226  177564                   .IIF EQ .-$TPS,$TPS:    .WORD   177564          ;;TTY PRINTER STATUS REG. ADDRESS
1091                                   .IIF NDF $TPB,$TPB:
1092  005230  177566                   .IIF EQ .-$TPB,$TPB:    .WORD   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
1093                                   .IIF NDF $NULL,$NULL:
1094  005232     000                   .IIF EQ .-$NULL,$NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
1095                                   .IIF NDF $FILLS,$FILLS:
1096  005233     002                   .IIF EQ .-$FILLS,$FILLS:        .BYTE   2               ;;CONTAINS # OF FILLER CHARACTER
1097                                   .IIF NDF $FILLC,$FILLC:
1098  005234     012                   .IIF EQ .-$FILLC,$FILLC:        .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LI
1099                                   .IIF NDF $TPFLG,$TPFLG:
1100  005235     000                   .IIF EQ .-$TPFLG,$TPFLG:        .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT
1101                                   .IIF NDF $QUES,$QUES:
1102  005236     077                   .IIF EQ .-$QUES,$QUES:  .ASCII  "?"             ;;QUESTION MARK
1103                                   .IIF NDF $CRLF,$CRLF:
1104  005237     015     000           .IIF EQ .-$CRLF,$CRLF:  .ASCIZ  <15>            ;;CARRAIGE RETURN
1105                                   .IIF NDF $LF,$LF:
1106  005241     012     000           .IIF EQ .-$LF,$LF:      .ASCIZ  <12>            ;;LINEFEED
1107          005244                   .IIF NE i&.     .EVEN
1108  005244                                   .$TRAP
1109  005244                          STARS
1110          001                      .IF B
1111                                   ;#######################################################################
1112                                   .IFF
1113                                   .NLIST
1114                                   .REPT
1115                                   .LIST
1116                                   ;#######################################################################
1117                                   .NLIST
1118                                   .ENDR
1119                                   .LIST
1120          000                      .ENDC
1121
1122                                   .SBTTL  TRAP DECODER
1123
1124                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
1125                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
1126                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
1127                                   ;*GO TO THAT ROUTINE.
1128
1129          001                      .IF B
1130  005244  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
1131                                   .IFF
```

```
1132                                    STRAP:  MOV     2(SP),-(SP)     ;;ASSUME THE STATUS OF
1133                                            BIC     #20,(SP)        ;;  THE CALLER--DONOT ALLOW
1134                                            MOV     #1$,-(SP)       ;;  T-BIT TRAPS
1135                                            RTI                     ;;SET THE NEW STATUS
1136                                    1$:     MOV     R0,-(SP)        ;;SAVE R0
1137                 000                .ENDC
1138  005246  016600  000002                    MOV     2(SP),R0        ;;GET TRAP ADDRESS
1139  005252  005740                            TST     -(R0)           ;;BACKUP BY 2
1140  005254  111000                            MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
1141                 001                .IF NB
1142                                            BPL     $TRAP1          ;;NON-USER TRAP,BELOW 200
1143                                            BIC     #^C177,R0       ;;STRIP AWAY THE JUNK
1144                                            JMP     (PC)            ;;USER TRAP,ABOVE 177, GO TO
1145                                            .WORD                   ;;  USER TRAP HANDLER-
1146                                    $TRAP1:
1147                 000                .ENDC
1148                 001                .IF NB
1149                                            CMP     #$TERM,R0       ;;CHECK FOR OUT OF BOUNDS
1150                                            BGT     .+6             ;;BR IF OK
1151                                            HALT                    ;;OUT OF BOUNDS
1152                                            BR      .-2             ;;HANGUP
1153                 000                .ENDC
1154  005256  006300                            ASL     R0              ;;POSITION FOR INDEXING
1155  005260  016000  005266                    MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
1156  005264  000200                            RTS     R0              ;;GO TO ROUTINE
1157
1158                                    .MACRO  SETTRAP A,B,MSG
1159                                            $$SET   A,B,\<TRAP+$TRP>,\$TRP,<MSG>
1160                                    .NLIST
1161                                    $TRP=$TRP+1
1162                                    .LIST
1163                                    .ENDM   SETTRAP
1164                                    .MACRO  $$SET   A,B,C,D,COMNT
1165                                    .IF EQ $TRP
1166
1167                                    .SBTTL  TRAP TABLE
1168
1169                                    ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1170                                    ;*BY THE "TRAP" INSTRUCTION.
1171
1172                                    ;       ROUTINE
1173                                    ;       -------
1174                                    $TRPAD:
1175                                    .ENDC
1176                                    .IIF NDF GNS,.NLIST
1177                                            A=      C
1178                                    .IIF NDF GNS,.LIST
1179                                            B               ;;CALL=A        TRAP+D(C)       COMNT
1180                                    .ENDM   $$SET
1181                                    .MACRO  TRMTRP
1182                                    $TERM=.-$TRPAD
1183                                    .ENDM   TRMTRP
1184                 001                .IF DF $TYPE
1185  005266                                    SETTRAP TYPE,$TYPE,^/TTY TYPEOUT ROUTINE/
1186  005266                                    $$SET   TYPE,$TYPE,\<TRAP+$TRP>,\$TRP,<TTY TYPEOUT ROUTINE>
1187                 002                .IF EQ $TRP
```

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

```
ASCNT   004336
BEGIN   001524
BINCT   004334
BIT0    =  000001
BIT00   =  000001
BIT01   =  000002
BIT02   =  000004
BIT03   =  000010
BIT04   =  000020
BIT05   =  000040
BIT06   =  000100
BIT07   =  000200
BIT08   =  000400
BIT09   =  001000
BIT1    =  000002
BIT10   =  002000
BIT11   =  004000
BIT12   =  010000
BIT13   =  020000
BIT14   =  040000
BIT15   =  100000
BIT2    =  000004
BIT3    =  000010
BIT4    =  000020
BIT5    =  000040
BIT6    =  000100
BIT7    =  000200
BIT8    =  000400
BIT9    =  001000
BPTVEC=  000014
BUFTST  001760
COUNT   001242
CSR     =  167770
DDISP   =  177570
DECML   004330
DISPLA  001220
DISPRG  000174
DRBHIO  001230
DRIBUF  001226
DRLVL   0012..4
DROBUF  001224
DRVECA  001232
DRVECB  001236
DSWR    =  177570
EMTVEC=  000030
END     003464
END1    003650
ERRVEC=  000004
FIRST   001350
HDFHM   004314
HLT     =  104000
ICOUNT  004466
IOTVEC=  000020
LIST    004340
LOGIC   003636
MESS1   005270
MINUS   004136
MKNUM   004166
MORDD   004250
MOREID  004510
PC      =%000007
PFAIL   004624
PIRQ    =  177772
PIRQVE=  000240
PL      001244
PRTAB   004056
PR0     =  000000
PR1     =  000040
PR2     =  000100
PR3     =  000140
PR4     =  000200
PR5     =  000240
PR6     =  000300
PR7     =  000340
PS      =  177776
PSW     =  177776
RCSR    001200
RCSR1   001210
RESTAR  004662
RESVEC=  000010
RETURN  004472
RSTART  001472
R0      =%000000
R1      =%000001
R2      =%000002
R3      =%000003
R4      =%000004
R5      =%000005
R6      =%000006
R7      =%000007
SAVCC   004054
SAVPC   004052
SAVR2   004040
SAVR3   004042
SAVR4   004044
SAVR6   004660
SCOPEA  004460
SCOPEB  004352
SCOPEF  004470
SCOPEG  004454
SEVEN   004326
SP      =%000006
SPEC    001334
SPEC0   004712
SPEC1   004742
SPEC2   004772
SR      001216
STACK   =  001100
STAR    004144
START   001444
START1  001246
STKLMT=  177774
STKPTR=  001200
SWREG   000176
SW00    =  000001
SW01    =  000002
SW02    =  000004
SW03    =  000010
SW04    =  000020
SW05    =  000040
SW06    =  000100
SW07    =  000200
SW08    =  000400
SW09    =  001000
SW1     =  000002
SW10    =  002000
SW11    =  004000
SW12    =  010000
SW13    =  020000
SW14    =  040000
SW15    =  100000
SW2     =  000004
SW3     =  000010
SW4     =  000020
SW5     =  000040
SW6     =  000100
SW7     =  000200
SW8     =  000400
SW9     =  001000
TAG     002024
TBITVE=  000014
TCRLF   =  000200
TCSR    004050
TDBR    004046
THT     =  000011
TINT1   002766
TINT2   003036
TINT3   003106
TINT4   003156
TINT5   003236
TINT6   003306
TINT7   003366
TKVEC   =  000060
TOODLE  004324
TPVEC   =  000064
TRAPVE=  000034
TRTVEC=  000014
TST4    002646
TST5    002716
TST6    002474
TST9    002530
TYP     005012
TYPE    =  104400
WAIT1   004114
WAIT2   004272
WGTCT   004332
WRTOC   004206
XLIST   004264
XOR     004500
XORA    001430
XORFLG  001240
$CHARC  005222
$CRLF   005237
$FILLC  005234
$FILLS  005233
$LF     005241
$NULL   005232
$QUES   005236
$TPB    005230
$TPFLG  005235
$TPS    005226
$TRAP   005244
$TRP    =  000001
$TRPAD  005266
$TYPE   005012
$TYPEC  005156
$TYPEX  005224
.HLT    003654
.SCOPE  004364
.       =  005325
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

#DZDRCF,DZDRCF/SOL=DZDRCF
RUN-TIME: 12.7 .4 SECONDS
RUN-TIME RATIO: 75/20=3.7
CORE USED: 11K (21 PAGES)

Spooler runtime 4 Seconds, 22 KCS, 97 disk reads, 3 disk writes, 27 pages