

.MAIN. - Parallel port test for the DSC 3/4
PORTTEST

```

09 0001          version =          1
10 0000          revision=         0
11              .pabs
12              .phex
13 0100          .loc    100h
14
15              ;      Program:  PORTTEST
16
17              ;      This program outputs a byte pattern (starting
18              ; with 00 and ending with FFh) from the first para-
19              ; ller port, then reads it in from the first para-
20              ; ller port, and compares the two byte patterns.
21              ; If they are not identical, an error message is
22              ; printed to the console. The procedure is then
23              ; reversed, and the same comparison test is made.
24              ;      If an error is found, a message is output
25              ; to the console, and also to an error buffer (that
26              ; starts at 500h). Error buffer writing is achieved
27              ; by a error flag check at CONOUT.
28              ;
29              ;      Console commands are as follows:
30              ;
31              ;      S  -  Print out a summary of all errors
32              ;          that have occurred thus far.
33              ;
34              ;      Q  -  Optionally quit the program
35              ;
36              ;      ?  -  Print a command summary
37              ;
38              ; NOTE: A special connector is needed for this test.
39              ;
40              ;
41              ;          D. Stein  11/10/80
42 0100          C3 0109          JMP      START
43
44              ; Port addresses and definitions
45
46 0001          port1   =          01   ; 1st parallel port addr
47 0002          port2   =          02   ; 2nd parallel port addr
48 0008          PIOAD   =          08   ; Parallel I/O port addr
49 0103          00      bitpat: .byte  00   ; bit pattern byte
50 0104          00      testpat: .byte  00   ;
51 0105          portname:
52 0105          00      .byte  00   ; Name of current port
53              ; under test.
54              ; 01 = port 1
55              ; 02 = port 2
56
57              ; Error Data addresses and definitions
58
59 0500          errbuf  =          500h ; Buffer for holding
60              ; test errors.
61 0106          erbufaddr:
62 0106          0000      .word  00h   ; Current addr in
63              ; error buffer.

```

```

09 0108      00          errfls# .byte  00h  ; Flag for writing to
10                                     ; the error buffer.
11                                     ; FF = set = write to buf
12
13 000D      cr          =          0Dh  ; ascii carriage return
14 000A      lf          =          0Ah  ; ascii line feed
15
16 0109      START:
17          ; Initialize
18 0109      31 0427     lxi          sp,stack
19
20 010C      DB01       in          port1
21 010E      DB02       in          port2      ; Kill any present port
22                                     ; contents.
23 0110      3E00       mvi          A,0
24 0112      32 0103     sta          bitpat   ; Initialize bit pattern
25                                     ; byte to 00000000
26
27 0115      21 0500     lxi          H,errbuf
28 0118      22 0104     shld         erbufaddr ; initialize the err buffer
29
30          ..buffill:
31 011B      3600       mvi          M,00      ; fill the error buffer
32 011D      23         inx          H          ; with nulls.
33 011E      7C         mov          A,H
34 011F      FE00       cpi          0000h    ; filled up to 0000h yet?
35 0121      20F8       jrnz         ..buffill
36
37 0123      3E00       mvi          A,00
38 0125      32 0103     sta          errfls   ; error buffer write flag
39                                     ; 0 = reset = off

41          ;-----
42          ; BEGINNING OF MAIN TEST LOOP
43          ;-----
44 0128      STRTTEST:
45 0128      21 0230     lxi          H,testmsg   ; Print the header
46 012B      CD 02FC     call         prtmsg
47 012E      21 02F7     lxi          H,crlf     ; and space down.
48 0131      CD 02FC     call         prtmsg
49 0134      PITEST:
50 0134      21 02FA     lxi          H,crctn   ; reset cursor to line strt
51 0137      CD 02FC     call         prtmsg
52 013A      3A 0103     lda          bitpat
53 013D      CD 0344     call         prtbit    ; print the pattern to con
54 0140      21 0243     lxi          H,space1
55          0143      CD 02FC     call         prtmsg
56 0146      3E01       mvi          A,01
57 0148      32 0105     sta          portname  ; portname=port1
58 014B      DB08       ..1:    in          P10AD    ; Wait for port 1 ready
59 014D      CB4F       bit          5,A
60 014F      20FA       jrnz         ..1
61 0151      3A 0103     lda          bitpat   ; Get the bit pattern byte
62 0154      4F         mov          C,A
63 0155      D301       out          port1    ; Output the bit pattern.
    
```

```

09 0157    DB08      ..2:   in      FIOAD
10 0159    CB5F      bit      3,A
11 015B    28FA      Jrz     ..2
12 015D    DB02      in      port2      ; read in the bit pattern
13 015F    32 0104   sta     testpat    ; and save it.
14 0162    CD 01E2   call    TEST       ; Compare bitpat & testpat
15
16 0165                                P2TEST:
17 0165    3E02      mvi     A,02
18 0167    32 0105   sta     portname   ; portname=port2
19 016A    DB08      ..3:   in      FIOAD
    016C    CB77      bit      4,A
21 016E    20FA      Jrnz    ..3
22 0170    3A 0103   lda     bitpat     ; Get the bit pattern byte.
23 0173    4F         mov     C,A
24 0174    D302      out     port2     ; Output the bit pattern.
25 0176    DB08      ..4:   in      FIOAD
26 0178    CB67      bit      4,A
27 017A    28FA      Jrz     ..4
28 017C    DB01      in      port1     ; Read in the bit pattern
29 017E    32 0104   sta     testpat    ; and save it.
30 0181    CD 01E2   call    TEST       ; Compare bitpat & testbit
31
32
33 0184    21 0103   lxi     H,bitpat
34 0187    34         inr     M          ; Increment to next pattern
35
36 0188                                CHECKCON:
37 0188    CD 03A7   call    CONSTAT    ; Any console input?
38 018B    D2 0134   Jnc     PITEST     ; No, continue testing.
39 018E    CD 03B5   call    CONIN      ; Yes, read the chr.
40 0191    FE53      cpi     'S'        ; User wants error list?
41 0193    2806      Jrz     ..5        ; Yes.
42 0195    FE51      cpi     'Q'        ; Does user want to quit?
43 0197    2814      Jrz     ..6        ; Yes.
44 0199    1832      Jmp     ..7        ; No. Print a help summary.
45
46 019B    21 02F7   ..5:   lxi     H,crLf    ; Space down a line.
47 019E    CD 02FC   call    prtmsg
48 01A1    CD 0307   call    prterr     ; Print the errs to console
49 01A4    21 02F7   lxi     H,crLf    ; Space down a line.
50 01A7    CD 02FC   call    prtmsg
51 01AA    C3 0128   jmp     STRTTEST   ; resume testing
52
53 01AD    21 02F7   ..6:   lxi     H,crLf    ; Space down a line.
54 01B0    CD 02FC   call    prtmsg
55 01B3    21 025A   lxi     H,QUITmsg ; Print quit verify msg
56 01B6    CD 02FC   call    prtmsg
57 01B9    CD 03B5   call    CONIN      ; and set a console chr.
58 01BC    F5         push    PSW
    01BD    CD 03CF   call    CONOUT     ; Echo the character.
60 01C0    F1         pop     PSW
61 01C1    FE59      cpi     'Y'        ; Do we quit?
62 01C3    CA 0000   Jz     WBOOT-1    ; Yes.
63 01C6    FE4E      cpi     'N'
    
```

.MAIN. - Parallel port test for the DSC 3/4

PORTTEST

```

09 01C8      CA 0128      Jz      STRTTEST      ; No. Resume testing
10 01CB      18E0      Jmpr    ..6          ; Bad entry. Ask again.
11
12 01CD      21 02F7      ..7:    lxi      H,crLf      ; Space down a line.
13 01D0      CD 02FC      call    prtmsg
14 01D3      21 0283      lxi      H,HELPMSS   ; Print the help summary.
15 01D6      CD 02FC      call    prtmsg
16 01D9      21 02F7      lxi      H,crLf      ; Space down a line
17 01DC      CD 02FC      call    prtmsg
18 01DF      C3 0128      jmp     STRTTEST      ; and resume testing.
19
20
21           ;-----
22           ; END OF MAIN TEST LOOP
23           ;-----
24
25
26           ;-----
27           ; Subroutine:  TEST
28           ; Regs in:    none
29           ; Regs out:   none
30           ; Destroyed:  B
31           ; Compare the sent byte (bitpat)
32           ; with the received byte(testpat).
33           ; If an error is found, turn on the error-buffer-write
34           ; flag, so the message is saved in the error buffer.
35 01E2      TEST:
36 01E2      3A 0104      lda     testpat ; B = received byte
37 01E5      47          mov     B,A
38 01E6      3A 0103      lda     bitpat  ; A = sent byte
39 01E9      90          sub     B       ; A = A - B
40 01EA      FE00      cpi     0       ; Are bytes identical?
41 01EC      C8          rz          ; Yes. Return
42 01ED      3EFF      mvi     A,OFFh
43 01EF      32 0108      sta     errfls ; Set the error flag
44 01F2      21 0222      lxi     H,portmsg
45 01F5      CD 02FC      call    prtmsg ; Print which port got the err
46 01F8      3A 0105      lda     portname
47 01FB      CD 0328      call    prtbyt
48 01FE      21 0243      lxi     H,space1
49 0201      CD 02FC      call    prtmsg
50 0204      3A 0103      lda     bitpat ; Print the sent bit pattern
51 0207      CD 0344      call    prtbit
52 020A      21 0245      lxi     H,errmsg
53 020D      CD 02FC      call    prtmsg
54 0210      3A 0104      lda     testpat
55 0213      CD 0344      call    prtbit ; Print the recvd bit pattern.
56 0216      3E00      mvi     A,00
57 0218      32 0108      sta     errfls ; Reset the error flag
58 021B      21 02F7      lxi     H,crLf ; and space down
59 021E      CD 02FC      call    prtmsg
60 0221      C9          ret
61
62           ;-----
63           ; Messages

```

```

09          ;
10 0222      0D0A4552524F portmsg: .asciz  [cr][lf]'ERROR port '
11 0230      0D0A54657374 testmsg: .asciz  [cr][lf]'Testing pattern '
12 0243      2000          space1: .asciz  ' '
13 0245      2073656E742C errmsg: .asciz  ' sent, but received '
14 025A      0D0A20446F20 QUITmsg: .ascii  [cr][lf]' Do you wish to '
15 026C      657869742050          .asciz  'exit PORTTEST? (Y/N) - '
16 0283      0D0A20504F52 HELPmsg: .ascii  [cr][lf]' PORTTEST COMMAND SUMMARY'
17 029E      0D0A202D2D2D          .ascii  [cr][lf]' -----'
18 02B9      0D0A20202020          .ascii  [cr][lf]'      S - Print the '
19 02D0      6572726F7220          .ascii  'error summary'
20 02DD      0D0A20202020          .ascii  [cr][lf]'      Q - Optional quit'
21 02F7      0D0A00          crlf: .asciz  [cr][lf]
22 02FA      0D00          crctn: .asciz  [cr]
23
24          ;-----
25          ; Utility Subroutines
26          ;
27
28          ;-----
29          ; Print a message
30 02FC      prtmsg:
31 02FC      7E          mov     A,M     ; set next char
32 02FD      B7          ora     A
33 02FE      C8          rz          ; return if zero
34 02FF      E5          push    H
35 0300      CD 03CF    call   CONOUT
36 0303      E1          pop     H
37 0304      23          inc     H     ; increment to next char
38 0305      18F5      jmp     prtmsg ; and print it
39
40          ;-----
41          ; Print the error buffer to the console,
42          ; pausing if a console character is input--
43          ; and resuming when another console character
44          ; is received.
45 0307      prterr:
46 0307      21 0500    lxi     H,errbuf
47 030A      E5          ..check: push   H
48 030B      CD 03A7    call   CONSTAT ; Is a con chr ready?
49 030E      300B      jrnc   ..0     ; No. Resume printing.
50 0310      CD 03B5    call   CONIN   ; Read the chr.
51 0313      CD 03A7    ..wait: call   CONSTAT ; Is another chr ready?
52 0316      30FB      jrnc   ..wait  ; No.
53 0318      CD 03B5    call   CONIN   ; Yes. Read chr and continue.
54 031B      E1          ..0:   pop     H
55 031C      7E          mov     A,M     ; Get the next error buffer chr
56 031D      FE00      cpi     00
57 031F      C8          rz
58 0320      E5          push    H
59 0321      CD 03CF    call   CONOUT
60 0324      E1          pop     H
61 0325      23          inc     H
62 0326      18E2      jmp     ..check ; Loop back for next chr
63

```

09
10
11

```
13 ;-----  
13 ; Print the A register as a hex number  
14 0328 prtbyt:  
15 0328 F5          push    PSW      ; save A  
16 0329 07          rlc         ; set upper nibble  
17 032A 07          rlc  
18 032B 07          rlc  
19 032C 07          rlc  
20 032D CD 0335     call    prt nib  ; print it  
21 0330 F1          pop     PSW      ; restore A  
22 0331 CD 0335     call    prt nib  ; print lower nibble  
23 0334 C9          ret
```

```
24 ;-----  
25 ; Print a nibble  
26 0335 prt nib:  
27 0335 E6 0F          ani     0Fh     ; mask to get nibble  
28 0337 FE 0A          cpi     10      ; convert to ASCII  
29 0339 FA 033E       jm     ..1  
30 033C C6 07          adi     7  
31 033E C6 30          ..1:  adi    '0'  
32 0340 CD 03CF       call   CONOUT  
33 0343 C9          ret
```

```
34 ;-----  
35 ; Subroutine: prt bit  
36 ; Res: in: A = byte to be printed 'bit by bit'  
37 ;  
38 ; Print out all 8 bits of a byte  
39 prt bit:  
40 0344 F5          push    PSW  
41 0344 E6 80          ani     128  
42 0345 0F          rrc  
43 0347 0F          rrc  
44 0348 0F          rrc  
45 0349 0F          rrc  
46 034A 0F          rrc  
47 034B 0F          rrc  
48 034C 0F          rrc  
49 034D 0F          rrc  
50 034E C6 30          adi     '0'  
51 0350 CD 03CF       call   CONOUT  
52 0353 F1          pop     PSW  
53 0354 F5          push    PSW  
54 0355 E6 40          ani     64  
55 0357 0F          rrc  
56 0358 0F          rrc  
57 0359 0F          rrc  
58 035A 0F          rrc  
59 035B 0F          rrc  
60 035C 0F          rrc  
61 035D C6 30          adi     '0'  
62 035F CD 03CF       call   CONOUT  
63 0362 F1          pop     PSW
```

```
09 0363      F5          push    PSW
10 0364      E620       ani     32
11 0366      0F        rrc
12 0367      0F        rrc
13 0368      0F        rrc
14 0369      0F        rrc
15 036A      0F        rrc
16 036B      C630       adi     '0'
17 036D      CD 03CF    call   CONOUT
18 0370      F1        pop     PSW
19 0371      F5          push    PSW
20 0372      E610       ani     16
21 0374      0F        rrc
22 0375      0F        rrc
23 0376      0F        rrc
24 0377      0F        rrc
25 0378      C630       adi     '0'
26 037A      CD 03CF    call   CONOUT
27 037D      F1        pop     PSW
28 037E      F5          push    PSW
29 037F      E608       ani     08
30 0381      0F        rrc
31 0382      0F        rrc
32 0383      0F        rrc
33 0384      C630       adi     '0'
34 0386      CD 03CF    call   CONOUT
35 0389      F1        pop     PSW
36 038A      F5          push    PSW
37 038B      E604       ani     04
38 038D      0F        rrc
39 038E      0F        rrc
40 038F      C630       adi     '0'
41 0391      CD 03CF    call   CONOUT
42 0394      F1        pop     PSW
43 0395      F5          push    PSW
44 0396      E602       ani     02
45 0398      0F        rrc
46 0399      C630       adi     '0'
47 039B      CD 03CF    call   CONOUT
48 039E      F1        pop     PSW
49 039F      E601       ani     01
50 03A1      C630       adi     '0'
51 03A3      CD 03CF    call   CONOUT
52 03A6      C9          ret
```

```
53
54          ;-----
55          ; Call bios directly using WBOOT in low memory
56 0001      WBOOT    =      1
57
58 03A7      CONSTAT:
59 03A7      21 03B3    lxi     H,...1    ; return to ...
60 03AA      E5        push    H
61 03AB      2A 0001    lhl    WBOOT
62 03AE      11 0003    lxi     D,03h
63 03B1      19        dad     D
```

```

09 03B2    E9                pchl
10
11 03B3    0F                ..1:  rrc                ; rotate into carry
    03B4    C9                ret                ; and return
13
14 03B5                                CONIN:
15 03B5    21 03C1           lxi    H,..2      ; return to ..2
16 03B6    E5                push   H
17 03B9    2A 0001           hlhd   WBOOT
18 03BC    11 0006           lxi    D,06h
19 03BF    19                dad    D
20 03C0    E9                pchl
21
22 03C1    FE03           ..2:  cpi    3      ; 3 is Control-C
23 03C3    CA 0000           jz     0          ; Wboot if Control-C
24 03C6    FE61           cpi    'a'       ; make result UPPER CASE
25 03C8    F8                rm
26 03C9    FE7B           cpi    'z'+1
27 03CB    F0                rp
28 03CC    D620           sui    'a'-'A'
29 03CE    C9                ret
30
31 03CF                                CONOUT:
32 03CF    F5                push   PSW        ; Save the chr to be output.
33 03D0    3A 0108           lda    errflg
34 03D3    FEFF           cpi    0FFh      ; Is the error flag set?
35 03D5    200A           jrnz   ..3       ; No. Just output the chr.
36 03D7    2A 0106           hlhd   erbufaddr ; Yes. Put chr in the
37 03DA    F1                pop    PSW
38 03DB    F5                push   PSW
39 03DC    77                mov    M,A       ; error buffer.
40 03DD    23                inc    H         ; Increment and store
41 03DE    22 0106           shld   erbufaddr ; the error buffer address.
42
43 03E1    F1                ..3:  pop    PSW        ; Restore the chr.
44 03E2    4F                mov    C,A       ; Output the chr to the
45 03E3    2A 0001           hlhd   WBOOT     ; console.
46 03E6    11 0009           lxi    D,09h
47 03E9    19                dad    D
48 03EA    E9                pchl
49
50
51
52
53 03EB                                .blkb    60
54 0427    00                stack: .byte    00      ; room for stack
55
56                                .end

```


BITPAT 0103	CHECKC 0188	CONIN 03B5	CONOUT 03CF
CONSTA 03A7	CR 000D	CRLF 02F7	CRRTN 02FA
ERBUFA 0106	ERRBUF 0500	ERRFLG 0108	ERRMSG 0245
HELPMS 0283	LF 000A	P1TEST 0134	P2TEST 0165
PIOAD 0008	PORT1 0001	PORT2 0002	PORTMS 0222
PORTNA 0105	PRTBIT 0344	PRTBYT 0328	PRTERR 0307
PRTMSG 02FC	PRTNIB 0335	QUITMS 025A	REVISI 0000
SPACE1 0243	STACK 0427	START 0109	STRTTE 0128
TEST 01E2	TESTMS 0230	TESTPA 0104	VERSID 0001
WBOOT 0001	.BLNK. 0000:03 X	.DATA. 0000* X	.PRG. 0000' X

BITPAT	0103	-	001,49*	002,24	002,52	002,61	003,14	003,22	003,30
			003,33	004,31	004,38	004,50			
CHECKC	0188	-	003,36*						
CONIN	03B5	-	003,39	003,57	005,50	005,53	008,14*		
CONOUT	03CF	-	001,27	003,59	005,35	005,59	006,32	006,51	006,62
			007,17	007,26	007,34	007,41	007,47	007,51	008,31*
CONSTA	03A7	-	003,37	005,48	005,51	007,58*			
CR	000D	-	002,13*	005,10	005,11	005,14	005,16	005,17	005,18
			005,20	005,21	005,22				
CRLF	02F7	-	002,47	003,46	003,49	003,53	004,12	004,16	004,58
			005,21*						
CRRTN	02FA	-	002,50	005,22*					
ERBUFA	0106	-	001,61*	002,28	008,36	008,41			
ERRBUF	0500	-	001,59*	002,27	005,46				
ERRFLG	0108	-	002,09*	002,38	004,43	004,57	008,33		
ERRMSG	0245	-	004,52	005,13*					
HELPMS	0283	-	004,14	005,16*					
LF	000A	-	002,14*	005,10	005,11	005,14	005,16	005,17	005,18
			005,20	005,21					
P1TEST	0134	-	002,49*	003,38					
P2TEST	0165	-	003,16*						
PIQAD	0008	-	001,48*	002,58	003,09	003,19	003,25		
PORT1	0001	-	001,46*	002,20	002,57	002,63	003,28		
PORT2	0002	-	001,47*	002,21	003,12	003,18	003,24		
PORTMS	0222	-	004,44	005,10*					
PORTNA	0105	-	001,51*	002,57	002,57*	003,18	003,18*	004,46	
PRTBIT	0344	-	002,53	004,51	004,55	006,36	006,40*		
PRTBYT	0328	-	004,47	006,14*					
PRTERR	0307	-	003,48	005,45*					
PRTMSG	02FC	-	002,46	002,48	002,51	002,55	003,47	003,50	003,54
			003,56	004,13	004,15	004,17	004,45	004,49	004,53
			004,59	005,30*	005,38				
PRTNIB	0335	-	006,20	006,22	006,26*				
QUITMS	025A	-	003,55	005,14*					
REVISI	0000	-	001,10*						
SPACE1	0243	-	002,54	004,48	005,12*				
STACK	0427	-	002,18	008,54*	008,54				
START	0109	-	001,42	002,16*					
STRTTE	0128	-	002,44*	003,51	004,09	004,13			
TEST	01E2	-	001,23	001,60	002,42	003,14	003,30	004,22	004,27
			004,35*						
TESTMS	0230	-	002,45	005,11*					
TESTPA	0104	-	001,50*	003,13	003,14	003,29	004,32	004,36	004,54
VERSID	0001	-	001,09*						
WBOOT	0001	-	003,62	007,55	007,56*	007,61	008,17	008,23	008,45
.BLNK.	0000	-	000,00						
.DATA.	0000	-	000,00						
.PROG.	0000	-	000,00						