

```

; This is a simple program to test the 8259A with 16 bit systems for
; the S-100 PIC/RTC Board
; John Monahan      (monahan@vitasoft.org)      7/7/2010
;
LF      equ    0ah
CR      equ    0dh
;
SDSTAT      EQU    0H          ;Consol Output Status port
SDDATA      EQU    1H          ;Consol Output Data port
;
NSEOI       equ    20h        ;Non specific end of interrupt command
MasterPICPORT equ    20h      ;Hardware port the 8259A is assigned (two ports
20H & 21H)

MasterICW1  equ    00010111B  ;EDGE triggered, 4 bytes, single Master,ICW4 needed
MasterICW2  equ    8H         ;Base address for 8259A Int Table (8H X 4 = 20H), (Same
as IBM-PC)
MasterICW3  equ    0H         ;No slave
MasterICW4  equ    00000011B  ;No special mode, non buffer, Auto EOI, 8086.

      cseg  0H                ;Everything in this simple program will be at CS: 0H
      org  100H
;
      DB   0DBH,0EDH         ;CPM3/Z80 code. This will input port ED. This switches
in the 8086                  ;The 8086 monitor after initilization, will jump to
location 500H in RAM
      org  0500H
;
init: cld                    ;Set direction up
      cli                    ;Disabel interrupts
      mov  ax,cs              ;Note CS will be 0H
      mov  ds,ax              ;As will DS here
      mov  es,ax
      mov  ss,ax              ;Stack in same segment
      mov  sp,offset stack    ;Must point to a RAM area
      mov  bx,offset SIGNON   ;Send a signon message
      call print
      CALL CI                  ;Wait for stat key
      mov  bx,offset CRLFMSG  ;Send a CRLF
      call print

      mov  al,MasterICW1      ;Initilize the 8259A PIC Controller
      out  MasterPICPORT,al
      mov  al,MasterICW2      ;Ints start at 20H in RAM
      out  MasterPICPORT+1,al
      mov  al,MasterICW4      ;No slaves above, so 8259 does not expect ICW3
      out  MasterPICPORT+1,al

      mov  al,0h              ;NO mask (i.e. all 8 int lines will be accepted)
      out  MasterPICPORT+1,al

      MOV  DI,0                ;Set up Software INT vectors in low memory:
      MOV  AX,Offset TrapInt   ;Location of default interrupt. If chip is programmed
More: MOV  [DI],Word Ptr AX    ;wrong, it MAY end up here.
      INC  DI
      INC  DI
      MOV  [DI],CS
      INC  DI
      INC  DI
      CMP  DI,400H             ;Do from 0 to 3ffH
      JNZ  More

      MOV  DI,3FCH             ;Location of INT FF (8259A is not putting vector
on bus)
      MOV  AX,Offset TrapFFInt

```

```

MOV    [DI],Word Ptr AX

MOV    DI,000CH           ;Location for single byte 8086 CC debug trap
MOV    AX,Offset DebugTrap ;Location of Hardware Int V0 routine
MOV    [DI],Word Ptr AX  ;We will now (one by one) put in the 8
INC    DI                ;interrupt jump locations.
INC    DI
MOV    [DI],CS

                                ;Now setup the 8 jump locations for the hardware
MOV    DI,MasterICW2*4    ;<---- Location of 8259A INT table

MOV    AX,Offset V0int    ;Location of Hardware Int V0 routine
MOV    [DI],Word Ptr AX  ;We will now (one by one) put in the 8
INC    DI                ;interrupt jump locations.
INC    DI
MOV    [DI],CS
INC    DI
INC    DI

MOV    AX,Offset V1int    ;Location of Hardware Int V1 routine
MOV    [DI],Word Ptr AX
INC    DI
INC    DI
MOV    [DI],CS
INC    DI
INC    DI

MOV    AX,Offset V2int    ;Location of Hardware Int V2 routine
MOV    [DI],Word Ptr AX
INC    DI
INC    DI
MOV    [DI],CS
INC    DI
INC    DI

MOV    AX,Offset V3int    ;Location of Hardware Int V3 routine
MOV    [DI],Word Ptr AX
INC    DI
INC    DI
MOV    [DI],CS
INC    DI
INC    DI

MOV    AX,Offset V4int    ;Location of Hardware Int V4 routine
MOV    [DI],Word Ptr AX
INC    DI
INC    DI
MOV    [DI],CS
INC    DI
INC    DI

MOV    AX,Offset V5int    ;Location of Hardware Int V5 routine
MOV    [DI],Word Ptr AX
INC    DI
INC    DI
MOV    [DI],CS
INC    DI
INC    DI

MOV    AX,Offset V6int    ;Location of Hardware Int V6 routine
MOV    [DI],Word Ptr AX
INC    DI
INC    DI
MOV    [DI],CS

```



```

    JZ     CO
    MOV    AL,CL
    CMP    AL,0H          ;SD BOARD CANNOT TAKE A NULL
    JNZ    LX2
    RET
LX2:  OUT   SDDATA,AL
    RET

CSTS:  IN    AL,SDSTAT
    TEST   AL,02H
    JZ     NONE
    XOR    AL,AL
    DEC    AL
    RET                    ;RETURN WITH 0FFH IN [A] IF SOMETHING

NONE:  XOR    AL,AL
    RET

CI:    CALL  CSTS          ;Wait until something is there
    JZ     CI
    IN    AL,SDDATA
    RET

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
;
    ORG    700H

signon  DB    '8086 test of Interrupts on PIC/RTC board 7/6/2010',CR,LF
        DB    'Press any key to start...$'
CRLFMSG DB    CR,LF,'$'

TrapIntMSG DB    'Trap interrupt detected at a non-hardware assigned location.$'
TrapFFIntMSG DB    'Trap interrupt detected at 0FFH in RAM. This is a non-
hardware assigned location.$'
DebugTrapMSG DB    'Trap interrupt detected Software Debug INT at 0CH in RAM.$'

    ORG    900H

Int0MSG  DB    'V0 $'
Int1MSG  DB    'V1 $'
Int2MSG  DB    'V2 $'
Int3MSG  DB    'V3 $'
Int4MSG  DB    'V4 $'
Int5MSG  DB    'V5 $'
Int6MSG  DB    'V6 $'
Int7MSG  DB    'V7 $'

        RW   100          ;50 level Program stack (way more than needed!)

stack    DW   0H          ;Put stack here
;
;-----
;   Interrupt Routine(s) go here
;-----
    org    0B00H

TrapInt:
    Cli                    ;Critical area****
    MOV    BX,Offset TrapIntMSG
    jmp    Int_msg

TrapFFInt:
    Cli                    ;Critical area****
    MOV    BX,Offset TrapFFIntMSG
    jmp    Int_msg

```

```

DebugTrap:
    Cli                                ;Critical area****
    MOV    BX,Offset DebugTrapMSG
    CALL   print                        ;General info dump routine
    IRET

V0int:    Cli                          ;Critical area****
    MOV    BX,Offset Int0MSG ;Will arrive here from int vector at 20H in RAM
    jmp    Int_msg

V1int:    Cli                          ;Will arrive here from int vector at 24H in RAM
    MOV    BX,Offset Int1MSG
    jmp    Int_msg

V2int:    Cli                          ;Will arrive here from int vector at 28H in RAM
    MOV    BX,Offset Int2MSG
    jmp    Int_msg

V3int:    Cli
    MOV    BX,Offset Int3MSG
    jmp    Int_msg

V4int:    Cli
    MOV    BX,Offset Int4MSG
    jmp    Int_msg

V5int:    Cli
    MOV    BX,Offset Int5MSG
    jmp    Int_msg

V6int:    Cli
    MOV    BX,Offset Int6MSG
    jmp    Int_msg

V7int:    Cli
    MOV    BX,Offset Int7MSG
    jmp    Int_msg

Int_msg:
    CALL   print                        ;General info dump routine
;    Mov    Al,00001011B                ;Send OCW3 (Read 8259A Interrupt Service Reg)
;    OUT    MASTER_PIC_PORT,al
;    IN     AL,MasterPICPORT           ;Get and show Bit pattern returned.
;    CALL   ZBITS                      ;Send bit pattern along with a CR/LF

;    Mov    Al,NS_EOI                  ;8259A End of Interrupt command, can now allow another
interrupt
;    OUT    MASTER_PIC_PORT,al
    IRET
;

;END

```