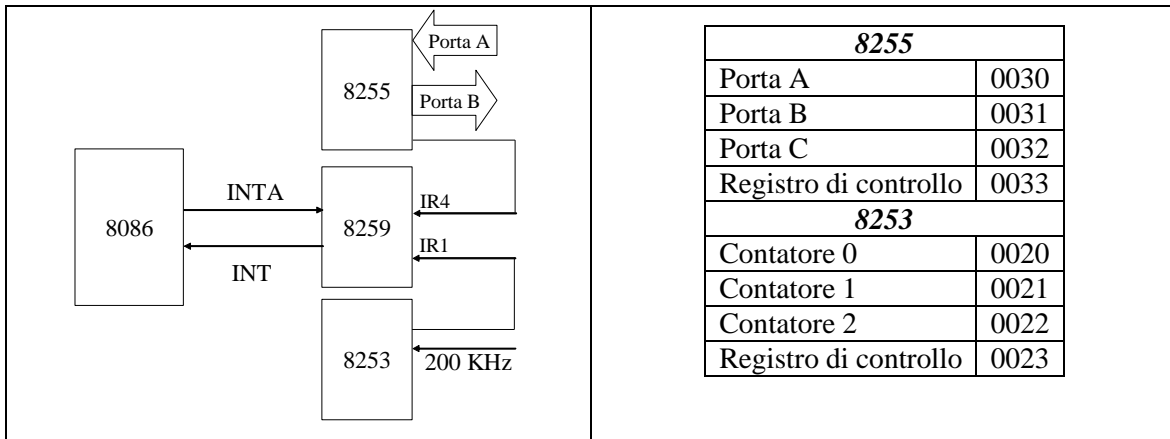


Laurea in Ingegneria Informatica
Sistemi a Microprocessori
Esame del 6 novembre 2009

Programmazione: è possibile consultare libri o appunti. Durata: 60 minuti.

Si faccia riferimento al *sistema di elaborazione* mostrato nella figura seguente:



La specifica di funzionamento del sistema è la seguente.

1. Il sistema legge un numero con segno (espressi in complemento a 2) dalla porta A dell'8255 non appena questo è disponibile (la segnalazione avviene sull'IR4 dell'8259)
2. Scrive il dato letto in una matrice 5x5, riempiendola per righe. Una volta completata la scrittura della matrice (cella 5,5) il successivo dato viene scritto nella prima posizione (cella 0,0) come mostrato nell'esempio
3. Ogni 10 ms, a seguito di una richiesta di interruzione su IR1 dell'8259 da parte dell'8253, calcola la differenza tra il valore massimo e quello minimo memorizzati sulla diagonale della matrice, come mostrato nell'esempio, e lo invia sulla porta B dell'8255.

Salvataggio 3° valore letto					Salvataggio 26° valore letto					Diagonale → differenza = 17				
2	4	6	2	-3	2	4	6	2	-3	2	4	6	2	-3
5	8	1	0	5	5	8	1	0	5	5	8	1	0	5
6	8	-9	1	2	6	8	-9	1	2	6	8	-9	1	2
-1	6	5	4	3	-1	6	5	4	3	-1	6	5	4	3
5	2	-4	6	8	5	2	-4	6	8	5	2	-4	6	8

Si richiede la scrittura di:

- ☐ procedura di inizializzazione dell'8253 e dell'8255
- ☐ programma principale
- ☐ procedure di servizio dell'interrupt.

```

.model small
.data
matr    db    25 dup (0)
.stack
.code

```

```

; procedura di inizializzazione della interrupt vector table

```

```

INIT_IVT    PROC
            PUSH    AX
            PUSH    BX
            PUSH    DX
            PUSH    DS
            XOR     AX,    AX
            MOV     DS,    AX

            ; channel 7
            MOV     BX,    39
            SHL     BX,    2
            MOV     AX,    offset ISR_PA_IN
            MOV     DS:[BX],    AX
            MOV     AX,    seg ISR_PA_IN
            MOV     DS:[BX+2],    AX

            ; channel 3
            MOV     BX,    35
            SHL     BX,    2
            MOV     AX,    offset ISR_COUNT0
            MOV     DS:[BX],    AX
            MOV     AX,    seg ISR_COUNT0
            MOV     DS:[BX+2],    AX

            POP     DS
            POP     DX
            POP     BX
            POP     AX
            RET
INIT_IVT    ENDP

```

```

; ISR for reading the value received on PA

```

```

ISR_PA_IN    PROC
            STI
            push    ax
            in      al, 030h
            mov     matr[si],al
            inc     si
            cmp     si, 25
            jne     valid
            mov     si, 0
valid:       pop     ax
            out     031h, al
            MOV     AL, 00100000b
            OUT     DX, AL
            CLI
            IRET
ISR_PA_IN    ENDP

```

; ISR for calculating MAX-MIN and send to Port B

```
ISR_COUNT0 PROC
    push    ax
    push    di
    push    cx
    mov     cx, 4
    mov     di, 0
    mov     al, matr[di]
    mov     ah, matr[di]
next:      add     di, 6
    cmp     al, matr[di]
    jl      no_max
    mov     al, matr[di]
    jmp     no_min
no_max:    cmp     ah, matr[di]
    jg      no_min
    mov     ah, matr[di]
no_min:    loop    next
    sub     al, ah
    out     031h, al
    MOV     AL, 00100000b
    OUT     DX, AL
    pop     cx
    pop     di
    pop     ax
    IRET
ISR_COUNT0 ENDP
```

; init 8255

```
INIT_8255 PROC
    ; init 8255
    mov al, 10110000b ; GA(m=1,pa=in,pcu=hndsk/intrpt) ;
GB(m=0,pb=out,pcl=out)
    out 033h, al
    ; set PC4 to enable interrupt on PA in
    mov al, 00001001b
    out 033h, al
    RET
INIT_8255 ENDP
```

;init 8253

```
INIT_8253 PROC
    ;init 8253
    ;counter0 init
    mov al, 00110100b
    out 023h, al
    mov ax, 2000-1
    xchg al, ah
    out 020h, al
    xchg al, ah
    out 020h, al
    RET
INIT_8253 ENDP
```

```

; init 8259
INIT_8259    PROC
                PUSH DX
                    PUSH AX
                        MOV  DX, 40H
                        MOV  AL, 00010011b
OUT            DX, AL
MOV            DX, 41H
MOV            AL, 00100000b
OUT            DX, AL
MOV            AL, 00000001b
OUT            DX, AL
MOV            AL, 11101101b
MOV DX, 41H
OUT            DX, AL
PUSH DX
PUSH AX
RET
INIT_8259    ENDP

```

```

;programma principale
    .startup
    CLI
    call INIT_IVT
call INIT_8259
    call INIT_8255
    call INIT_8253
    STI

    mov  si, 0
block: ;hlt
        jmp  block

    .exit

end ; set entry point and stop the assembler.

```