

Esercizi Assembly 10

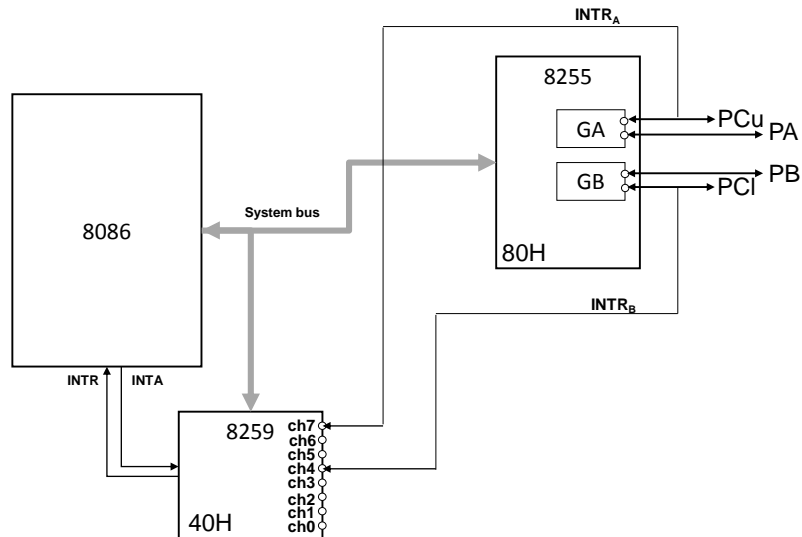
M. Sonza Reorda – M. Grosso

Politecnico di Torino
Dipartimento di Automatica e Informatica

Emulazione del dispositivo 8259

- Per consentire l'avvio dell'emulatore PIC 8259 è necessario anteporre al codice la seguente direttiva per l'emulatore:
`#START=8259.exe#`
 - Questo avvierà anche gli emulatori dell'8255 e dell'8253
- Il dispositivo Intel 8259 è accessibile a partire dall'indirizzo I/O 0x40
- Si ricordi di:
 - Chiudere tutte le finestre al termine dell'emulazione
 - impostare *delay* ≥ 100 ms in modalità *Run*.

Sistema emulato



Esercizio 1

- Si configuri l'Intel 8255 in modo 1 per i gruppi A e B, con il gruppo A in modalità di *input* e il gruppo B in *output*
- Si scrivano le procedure di servizio dell'interrupt che realizzino le seguenti specifiche:
 - Alla ricezione di un segnale di *interrupt* dal gruppo A, la porta A deve essere letta
 - Se il byte acquisito corrisponde al codice ASCII di una cifra decimale $n > 0$, si scriva sulla porta B il valore di tale cifra
 - Si scrivano poi, sempre sulla porta B, i valori $n-1, n-2, \dots, 1$
 - Si assuma che non arrivi sulla porta A un nuovo carattere sino a che non si è completata la scrittura sulla porta B di tutti i valori derivanti dal precedente carattere.

Implementazione

- Configurazione Interrupt Vector Table
- Configurazione PIC 8259
- Configurazione Intel 8255
- Definizione procedure di servizio dell'interrupt e realizzazione di un ciclo infinito
 - Verificare le funzionalità di salvataggio e ripristino dei registri da parte della ISR aggiungendo qualche istruzione di elaborazione nel programma principale (es: INC AX).

Codice

```

PORTA EQU 80h
PORTB EQU PORTA+1
PORTC EQU PORTA+2
CONTROL EQU PORTA+3

PIC EQU 40h

#start=8259.exe#

.model small
.data
count db 0
.stack
.code

; procedura di configurazione della porta parallela
INIT_8255 PROC
    PUSH AX
    PUSH DX
    MOV DX, CONTROL
    ; inizializzazione 8255
    MOV AL, 10110100b ; Gruppo A: modo 1, input
                        ; Gruppo B: modo 1, output
    OUT DX, AL
    ; abilitazione interrupt PA input (INTE A)
    MOV AL, 00001001b
    OUT DX, AL
    ; abilitazione interrupt PB output (INTE B)
    MOV AL, 00000101b
    OUT DX, AL
    POP DX
    POP AX
    RET
INIT_8255 ENDP

```

Codice

```
; procedura di inizializzazione della interrupt vector table
INIT_IVT PROC
    PUSH AX
    PUSH BX
    PUSH CX
    PUSH DS
    XOR AX, AX
    MOV DS, AX
    MOV BX, 00100111b    ; channel 7 (39)
    MOV CL, 2
    SHL BX, CL
    MOV AX, offset ISR_PA_IN
    MOV DS:[BX], AX
    MOV AX, seg ISR_PA_IN
    MOV DS:[BX+2], AX
    MOV BX, 00100100b    ; channel 4 (36)
    MOV CL, 2
    SHL BX, CL
    MOV AX, offset ISR_PB_OUT
    MOV DS:[BX], AX
    MOV AX, seg ISR_PB_OUT
    MOV DS:[BX+2], AX
    POP DS
    POP CX
    POP BX
    POP AX
    RET
INIT_IVT ENDP
```

Codice

```
; procedura di configurazione dell'interrupt controller
INIT_8259 PROC
    PUSH DX
    PUSH AX
    MOV DX, PIC
    MOV AL, 00011011b    ; ICW1
    ; LTIM = 1 (fronti), SNGL = 1, IC4 = 1
    OUT DX, AL
    MOV DX, PIC+1
    MOV AL, 00100000b    ; ICW2
    ; 5 bit alti del vettore interrupt = 00100
    OUT DX, AL
    MOV AL, 00000011b    ; ICW4
    ; SFNM = 0, BUF = 0, M/S = 0 (no buffer mode)
    ; AEIOI = 1, uPM = 1 (8086)
    OUT DX, AL
    MOV DX, PIC+1
    MOV AL, 01101111b    ; OCW1
    ; abilitazione canali 4 e 7
    OUT DX, AL
    POP DX
    POP AX
    RET
INIT_8259 ENDP
```

Codice

```

ISR_PA_IN  PROC
            PUSH AX
            PUSH DX

            MOV DX, PORTA
            IN AL, DX
            CMP AL, '1'
            JB ritorno
            CMP AL, '9'
            JA ritorno
            SUB AL, '0'

            MOV DX, PORTB
            OUT DX, AL
            DEC AL
            MOV count, AL

ritorno:    POP DX
            POP AX
            IRET
ISR_PA_IN  ENDP

ISR_PB_OUT PROC
            PUSH AX
            PUSH DX

            CMP count, 0
            JE fine
            MOV AL, count
            MOV DX, PORTB
            OUT DX, AL
            DEC count

            fine: POP DX
                  POP AX
                  IRET
ISR_PB_OUT ENDP

```

Codice

```

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

XOR AX, AX
block:
INC AX      ; per verificare funzionalita' di salvataggio e ripristino registri
JMP block  ; ciclo infinito

.exit

end

```

Esercizio 2

- Si scriva una procedura di servizio dell'interrupt che, dato il modulo Intel 8255 configurato con il gruppo A in modo 1 (input), sia in grado di acquisire una sequenza di caratteri ASCII nel vettore `parola`. Si memorizzino i caratteri corrispondenti a lettere maiuscole e minuscole e si scartino gli altri. La variabile `count` tenga il conto dei caratteri acquisiti
- Si scrivano anche le parole di configurazione del PIC 8259.

Codice

```

PORTA EQU 80h
PORTB EQU PORTA+1
PORTC EQU PORTA+2
CONTROL EQU PORTA+3

PIC EQU 40h

DIM EQU 100

#start=8259.exe#

.model small
.data
parola db DIM DUP (?)
count db 0
.stack
.code

; procedura di configurazione della porta parallela
INIT_8255 PROC
    PUSH AX
    PUSH DX
    MOV DX, CONTROL
    ; init 8255
    MOV AL, 10110000b ; Gruppo A: modo 1, input
    OUT DX, AL
    ; abilitazione interrupt PA input (INTE A)
    MOV AL, 00001001b
    OUT DX, AL
    POP DX
    POP AX
    RET
INIT_8255 ENDP

```

Codice

```
; procedura di inizializzazione della interrupt vector table
INIT_IVT PROC
    PUSH AX
    PUSH BX
    PUSH CX
    PUSH DS
    XOR AX, AX
    MOV DS, AX
    MOV BX, 39      ; channel 7
    MOV CL, 2
    SHL BX, CL
    MOV AX, offset ISR_PA_IN
    MOV DS:[BX], AX
    MOV AX, seg ISR_PA_IN
    MOV DS:[BX+2], AX
    POP DS
    POP CX
    POP BX
    POP AX
    RET
INIT_IVT ENDP
```

Codice

```
; procedura di configurazione dell'interrupt controller
INIT_8259 PROC
    PUSH DX
    PUSH AX
    MOV DX, PIC
    MOV AL, 00011011b ; ICW1
    ; LTIM = 1 (fronti), SNGL = 1, IC4 = 1
    OUT DX, AL
    MOV DX, PIC+1
    MOV AL, 00100000b ; ICW2
    ; 5 bit alti del vettore interrupt = 00100
    OUT DX, AL
    MOV AL, 00000011b ; ICW4
    ; SFNM = 0, BUF = 0, M/S = 0 (no buffer mode)
    ; AE0I = 1, uPM = 1 (8086)
    OUT DX, AL
    MOV DX, PIC+1
    MOV AL, 01111111b ; OCW1
    ; abilitazione canale 7
    OUT DX, AL
    POP DX
    POP AX
    RET
INIT_8259 ENDP
```

Codice

```

ISR_PA_IN  PROC                                ; programma principale
            PUSH AX                            .startup
            PUSH BX                            CLI
            PUSH DX                            call INIT_IVT
                                                call INIT_8259
                                                call INIT_8255
                                                STI

            MOV DX, PORTA
            IN AL, DX
            CMP AL, 'A'
            JB ritorno                         MOV count, 0
            CMP AL, 'Z'                       block:  jmp block ; ciclo infinito
            JB lettura
            CMP AL, 'a'                       .exit
            JB ritorno
            CMP AL, 'z'                       end
            JA ritorno

lettura:    XOR BH, BH
            MOV BL, count
            MOV parola[BX], AL
            INC count

ritorno:    POP DX
            POP BX
            POP AX
            IRET
ISR_PA_IN  ENDP

```

Esercizio 3

- Si scriva una procedura di servizio dell'interrupt che, dato il modulo Intel 8255 configurato con il gruppo A in modo 1 (output), sia in grado di scrivere un intero memorizzato in una variabile *doubleword* pre-inizializzata, a partire dal byte più significativo
- Si scrivano anche le parole di configurazione del PIC 8259
- Informazioni aggiuntive:
 - Si supponga che l'interrupt dalla porta A sia connesso al canale 6 del PIC 8259
 - Per la stampa del primo carattere è possibile ricorrere all'uso di una chiamata software INT.

Codice

```

PORTA EQU 80h
PORTB EQU PORTA+1
PORTC EQU PORTA+2
CONTROL EQU PORTA+3

PIC EQU 40h

DIM EQU 100

        #start=8259.exe#

        .model small
        .data
numero dd 0FAB3791Ch
        .stack
        .code

; procedura di configurazione della porta parallela
INIT_8255 PROC
        PUSH AX
        PUSH DX
        MOV DX, CONTROL
        ; init 8255
        MOV AL, 10100000b ; Gruppo A: modo 1, output
        OUT DX, AL
        ; abilitazione interrupt PA output (INTE A)
        MOV AL, 00001101b
        OUT DX, AL
        POP DX
        POP AX
        RET
INIT_8255 ENDP

```

Codice

```

; procedura di inizializzazione della interrupt vector table
INIT_IVT PROC
        PUSH AX
        PUSH BX
        PUSH CX
        PUSH DS
        XOR AX, AX
        MOV DS, AX
        MOV BX, 38 ; channel 6
        MOV CL, 2
        SHL BX, CL
        MOV AX, offset ISR_PA_OUT
        MOV DS:[BX], AX
        MOV AX, seg ISR_PA_OUT
        MOV DS:[BX+2], AX
        POP DS
        POP CX
        POP BX
        POP AX
        RET
INIT_IVT ENDP

```

Codice

```

; procedura di configurazione dell'interrupt controller
INIT_8259 PROC
    PUSH DX
    PUSH AX
    MOV DX, PIC
    MOV AL, 00011011b ; ICW1
    ; LTIM = 1 (fronti), SNGL = 1, IC4 = 1
    OUT DX, AL
    MOV DX, PIC+1
    MOV AL, 00100000b ; ICW2
    ; 5 bit alti del vettore interrupt = 00100
    OUT DX, AL
    MOV AL, 00000011b ; ICW4
    ; SFNM = 0, BUF = 0, M/S = 0 (no buffer mode)
    ; AE0I = 1, uPM = 1 (8086)
    OUT DX, AL
    MOV DX, PIC+1
    MOV AL, 10111111b ; OCW1
    ; abilitazione canale 6
    OUT DX, AL
    POP DX
    POP AX
    RET
INIT_8259 ENDP

```

Codice

```

ISR_PA_OUT PROC                                     ; programma principale
    PUSH AX                                         .startup
    PUSH DX                                         CLI
    PUSH SI                                         call INIT_IVT
                                                    call INIT_8259
                                                    call INIT_8255
                                                    STI

    CMP BX, 0
    JE ritorno
    DEC BX
    MOV DX, PORTA
    MOV AL, byte ptr numero[BX]
    OUT DX, AL
    block:
        MOV BX, 4 ; BX come contatore dei byte
        INT 38 ; stampa del primo byte
        jmp block ; ciclo infinito

ritorno:
    POP SI
    POP DX
    POP AX
    IRET
ISR_PA_OUT ENDP

```