

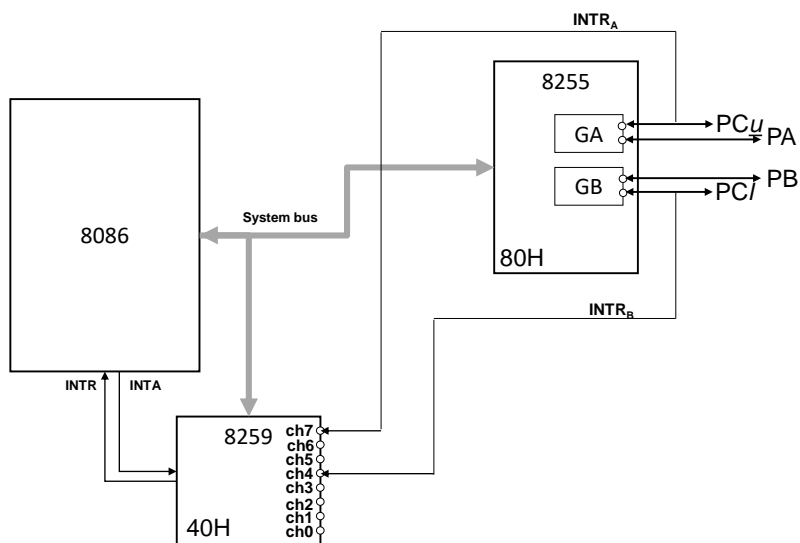
# Esercitazione in aula

## 05-06-2017

M. Rebaudengo – R. Ferrero

Politecnico di Torino  
Dipartimento di Automatica e Informatica

## Sistema emulato



# 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*  $\geq 100$  ms in modalità *Run*.

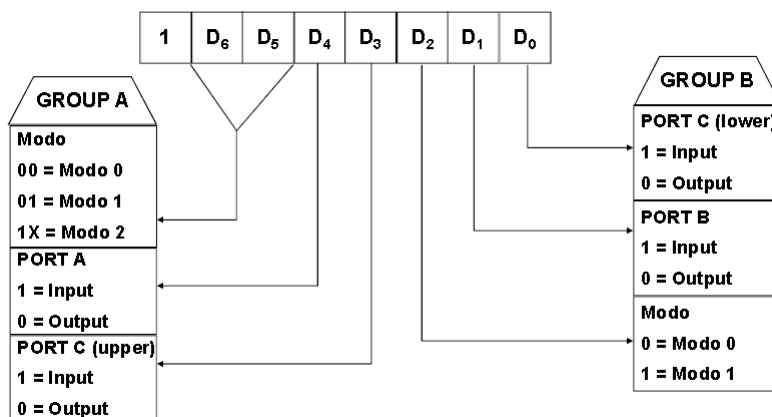
## Intel 8255 in modo 1

- 2 gruppi
- Ogni gruppo è composto da una porta di dato di 8 bit e da una porta di controllo di 4 bit
- I bit di dato possono essere di Input o di Output
- Input e Output sono latched.

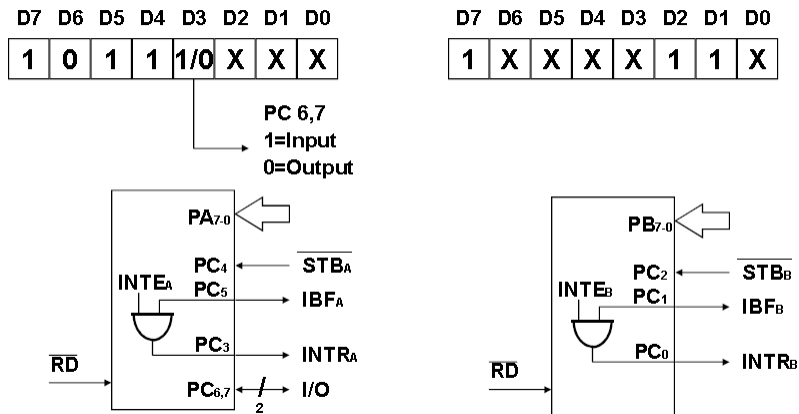
## Intel 8255 in modo 1: segnali di controllo in input

- STB (Strobe Input): un valore basso carica il dato nell'input latch
- IBF (Input Buffer Full): un valore alto indica che il dato è stato caricato nell'input latch (funziona da *acknowledge*)
- INTR (Interrupt Request): un valore alto può essere usato come richiesta di interrupt per la CPU
- INTEA (Interrupt Enable per il gruppo A): controllato dal bit set/reset di PC4
- INTEB (Interrupt Enable per il gruppo B): controllato dal bit set/reset di PC2.

## Intel 8255 - configurazione



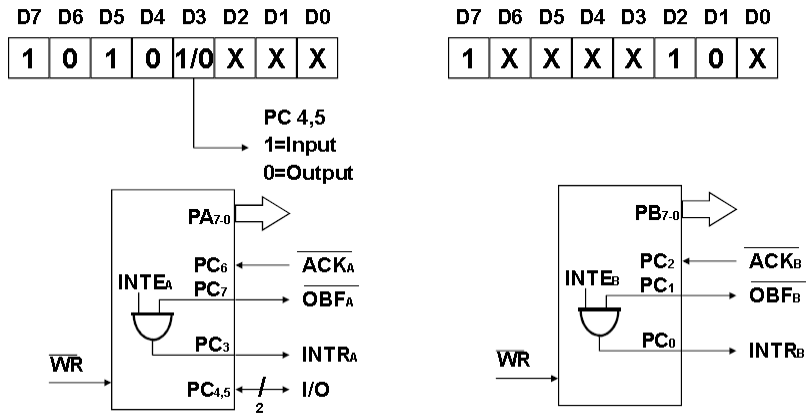
# Intel 8255 – Modo 1 Input



## Intel 8255 in modo 1: segnali di controllo in output

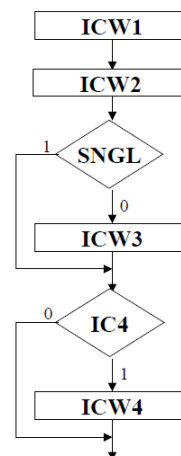
- OBF (Output Buffer Full): un valore basso indica che la CPU ha scritto il dato sulla porta
- ACK (Acknowledge Input): un valore basso informa l'8255 che il dato è stato ricevuto dalla periferica
- INTR (Interrupt Request): un valore alto può essere usato come richiesta di interrupt per la CPU
- INTEA (Interrupt Enable per il gruppo A): controllato dal bit set/reset di PC6
- INTEB (Interrupt Enable per il gruppo B): controllato dal bit set/reset di PC2.

# Intel 8255 – Modo 1 Output



## Intel 8259

- La configurazione del dispositivo richiede l'uso di:
  - *Initialization Command Words (ICW)*
  - *Operation Command Words (OCW)*
- Di seguito vedremo l'esempio di una semplice configurazione...



# 8259: esempio di configurazione

## ICW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	X	X	X	1	LTIM	X	SINGL	IC4
				0		1	1	

- LTIM (Level Triggered Mode)
- SINGL: modo Single o Cascade
- IC4: presenza della parola ICW4
- D4 = 1
- X: bit usati solo per CPU della famiglia 8080/85.

## ICW2

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	I7	I6	I5	I4	I3	X	X	X
	0	0	1	0	0			

Con la parola ICW2 la CPU determina i tipi di interrupt corrispondenti agli 8 segnali di richiesta di interruzione:

- I3-I7: 5 bit alti dell'indirizzo del vettore degli interrupt.
- X: bit usati solo per CPU della famiglia 8080/85.

## ICW4

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	SFNM	BUF	M/S	AEOI	μPM
				0	0	x	1	1

- SFNM: Special fully nested mode.
- BUF & M/S: Buffer Mode.
- AEOI: Automatic end of interrupt.
- μPM: Microprocessor mode.

## OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0
	0			0				

OCW1 permette di caricare il registro IMR.

- Ad ogni bit della parola OCW1 corrisponde un bit nel registro IMR.
- Settando il bit  $M_i$  ad 1 si setta il bit IMR<sub>i</sub>, e dunque si maschera il canale di interrupt IR<sub>i</sub>.

## 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, 00010011b ; ICW1
    ; LTIM = 0 (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, 00010011b ; ICW1
          ; LTIM = 0 (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
            CMP count, DIM                    call INIT_8259
            JE ritorno                        call INIT_8255
            MOV DX, PORTA                     STI
            IN AL, DX
            CMP AL, 'A'                       MOV count, 0
            JB ritorno                        block: jmp block ; ciclo infinito
            CMP AL, 'Z'
            JBE lettura                       .exit
            CMP AL, 'a'
            JB ritorno                        end
            CMP AL, 'z'
            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
```