

Politecnico di Torino - III Facoltà di Ingegneria
Esercitazioni del corso “Sistemi a microprocessore”
Ing. Paolo Bernardi – email: paolo.bernardi@polito.it

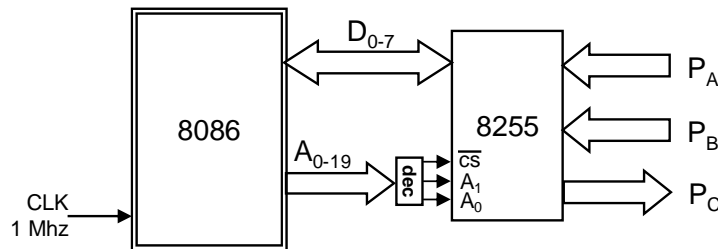
Esercitazione 6 – Le comunicazioni parallele: la periferica 8255

Argomenti trattati:

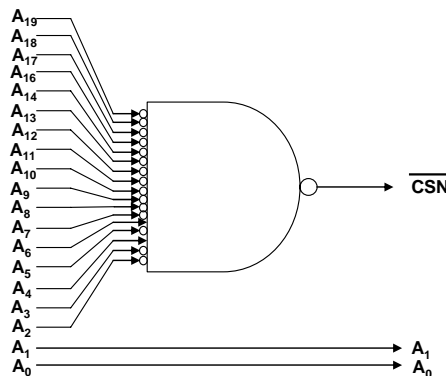
1. Lettura e scrittura in modo 0.
2. Lettura e scrittura in modo 1.
 - a. *Polling.*
 - b. *Funzioni di interrupt.*

Esercizio 1: Un sistema a microprocessore basato su 8086 deve comunicare tramite un'interfaccia di comunicazione parallela 8255: acquisito un valore dalla porta P_A lo deve ripetere sulla porta P_C dopo 1 millisecondo. L'8255 è mappato all'indirizzo 50H ed il microprocessore è alimentato da un segnale di clock di frequenza pari a 1 Mhz. Lo schema strutturale del sistema da programmare è quello riportato di seguito.

Si progetti la logica di decodifica per l'indirizzamento del dispositivo periferico 8255 e si programmi il sistema in modo che soddisfi le specifiche richieste.



Possibile svolgimento:



```

; valori di inizializzazione
PORTA    EQU    50H
PORTB    EQU    51H
PORTC    EQU    52H
CONTROL  EQU    53H
CW       EQU    10010010b

```

```

; Gruppo A e B programmati in modo 0
; PA e PB sono in ingresso, PC (upper e
; lower) in output

```

```

; gestione del ritardo - modo software
; consiste nel impostare un ciclo di lunghezza calcolata con la seguente formula

```

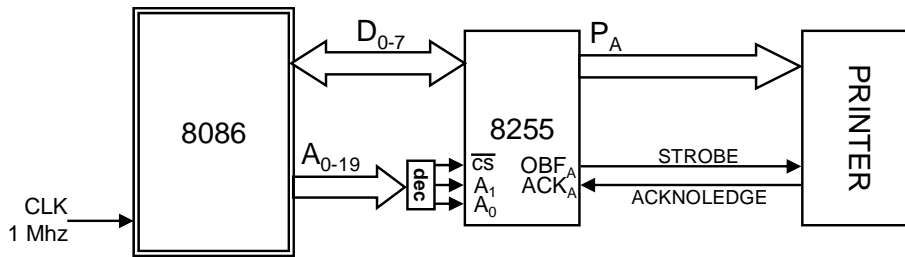
$(CK_{init_CX} (1) + CK_{se_loop} (2) * (N_{loop} - 1) + CK_{se_non_loop} (3) + CK_{init_DX} (4)) * T_{ck} = \text{ritardo}$
 ; nell'esercizio: $(4 + 17 * (N_{loop} - 1) + 5 + 4) * 10^{-6} = 0,001 \rightarrow N_{loop} = 60$

```

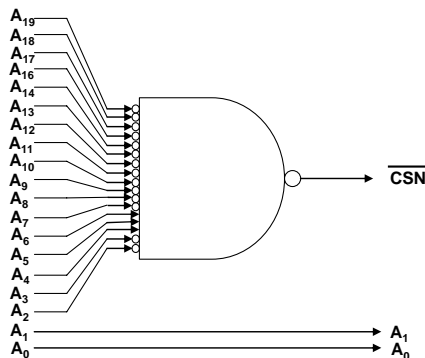
; programma
MOV    DX,    CONTROL
MOV    AL,    CW
OUT    DX,    AL                ; programmazione del modo funzionale
MOV    DX,    PORTA
IN     AL,    DX
MOV    CX,    60                ; (1) 4 cicli di clock
ritardo: LOOP ritardo          ; (2) 17 (se esegue salto)
MOV    DX,    PORTC            ; (3) 5 (se procede) cicli di clock
OUT    DX,    AL                ; (4) 4 cicli di clock
    
```

Esercizio 2: Un sistema a microprocessore basato su 8086 deve controllare una stampante tramite un'interfaccia parallela. La stampante riceve dalla porta P_A di un dispositivo periferico 8255 il carattere da stampare e richiede un segnale di STROBE che indichi che il valore trasmesso è valido. La stampante infine segnala al sistema a microprocessore che il dato è stato stampato correttamente tramite un segnale di ACKNOWLEDGE; ogni nuova trasmissione deve essere preceduta dalla conferma della corretta stampa. Il dispositivo 8255 è mappato all'indirizzo 70H.

Si progetti la logica di decodifica per l'indirizzamento del dispositivo periferico 8255 e si programmi il sistema in modo che soddisfi le specifiche richieste.



Possibile svolgimento:



```

; valori di inizializzazione
PORTA    EQU    70H
PORTB    EQU    71H
PORTC    EQU    72H
CONTROL  EQU    73H
CW        EQU    10100000b    ; gruppo A in modo 1
SPC6     EQU    00001101b    ; necessario settare il bit INTE_A
    
```

```

; comunicazione tra 8255 e stampante

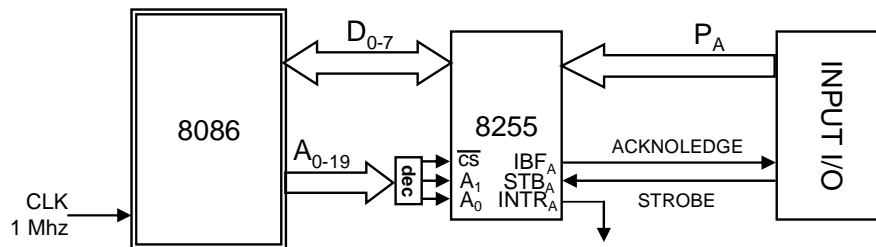
; programma
MOV  DX,  CONTROL
MOV  AL,  CW
OUT  DX,  AL
MOV  AL,  SPC6
OUT  DX,  AL
MOV  AL,  "A"
OUT  PORTA, AL
check: MOV  DX,  PORTC
      IN  AL,  DX
      AND 01000000, AL          ; controlla se il dato
                                ; precedente è stato stampato

      JZ  CHECK
      INC AL
      OUT PORTA, AL
      JMP check

```

Esercizio 3: Un sistema a microprocessore basato su 8086 deve ricevere ed elaborare valori provenienti da un dispositivo esterno. Tramite un'interfaccia parallela 8255 all'indirizzo 50H deve ricevere dal dispositivo un valore ed un segnale di STROBE che indichi che un dato valido è stato inviato. Ogni 200 valori ricevuti, il processore 8086 deve eseguirne la media ed inviarla, sempre tramite il dispositivo 8255.

Si programmi il sistema in modo che soddisfi le specifiche richieste.



Possibile svolgimento:

```

; valori di inizializzazione
PORTA EQU 50H
PORTB EQU 51H
PORTC EQU 52H
CONTROL EQU 53H
CW EQU 10110000b
SPC4 EQU 00001001b

; gruppo A in modo 1
; necessario settare il bit INTE_A
; per abilitare il meccanismo di
; comunicazione tra 8255 e stampante

; inizializzazione dispositivo
MOV  AL,  CW
MOV  DX,  53H
OUT  DX,  AL
MOV  AL,  SPC4
OUT  DX,  AL

; gestione dell'evento di scrittura
; vengono mostrate due possibili soluzioni
; a) viene impiegato il meccanismo di polling (controllo da parte

```

```

;      della CPU 8086)
;      b) viene impiegato il meccanismo di interrupt (controllo da
;      parte di una periferica di gestione dell'interrupt opportunamente
;      inizializzata)

;      a)      POLLING (controllo da parte della CPU 8086)

                MOV     BX,    0
                MOV     DI,    0
                MOV     CX,    200
                MOV     DX,    PORTC
non_pronto:    IN      AL,    DX
                TEST    AL,    00001000b      ; controlla il pin INTRA
                JZ     non_pronto
                INC     DI
                MOV     DX,    PORTA
                IN      AL,    DX
                CBW
                ADD     BX,    AX
                CMP     DI,    CX
                JNE    non_pronto
                XCHG   BX,    AX
                DIV    CL                    ; AL contiene il valore media
                MOV     DX,    PORTB
                OUT    DX,    AL

;      b)      INTERRUPT (controllo da parte dell'8259 preventivamente
;      inizializzato - sistema in attesa fino a nuovo interrupt)

; programma principale
                MOV     DI,    0
                MOV     CX,    200
non_pronto:    HALT                          ; pone il sistema in uno stato di
                                                ; attesa da cui si esce esclusivamente
                                                ; con un evento di interrupt

                CMP     DI,    CX
                JNE    non_pronto
                XCHG   BX,    AX
                DIV    CL
                MOV     DX,    PORTB
                OUT    DX,    AL

; procedura di INTERRUPT
leggi          PROC
                MOV     DX,    PORTA
                IN      AL,    DX
                CBW
                ADD     BX,    AX
                INC     DI
                IRET
leggi          ENDP

```

Esercizio 4: Un sistema di trasmissione deve soddisfare le seguenti specifiche:

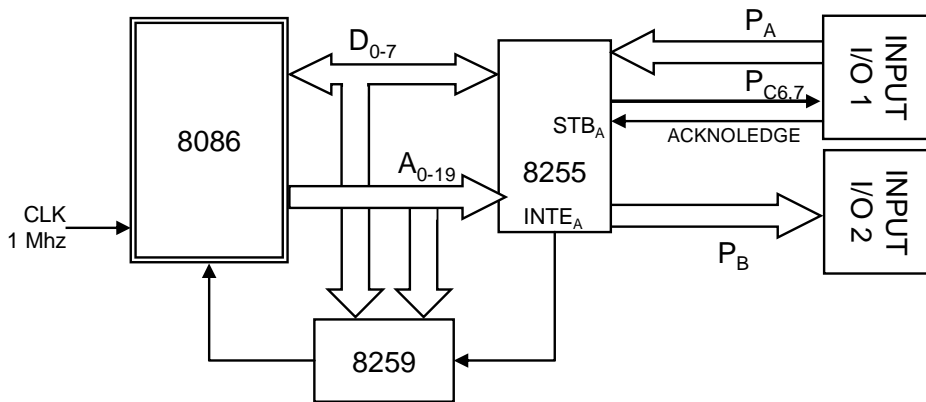
- ricevere ogni decimo di secondo un valore su 8 bit
- controllare la parità della parola ricevuta; l'ultimo bit di ogni parola ricevuta è infatti 0 se il numero di 1 nei restanti 7 bit è pari, 1 se dispari

- se il dato è stato trasmesso correttamente, salva il dato in memoria attende il dato successivo
- se rileva un errore,
 - chiede una ritrasmissione al dispositivo trasmittente inviando un codice di errore su 2 bit (3H)
 - attende che il dispositivo trasmittente invii un segnale di acknowledge e ricampiona il valore.
- il sistema, ottenuti 10 valori ne calcola la media e la spedisce ad un secondo dispositivo.

Si progetti il suddetto sistema a microprocessore, utilizzando un numero adeguato di periferiche tra le seguenti:

- 8086
- 8255 (80H)
- 8259 (da non programmare).

Possibile svolgimento:



```

CW          EQU    10110000b          ; gruppo A in modo 1
                                ; gruppo B in modo 0
SPC4       EQU    00001001b          ; x setta il bit INTEA per
                                ; abilitare il meccanismo di
RPC4       EQU    00001000b          ; comunicazione tra 8255 e stampante
                                ; resetta il bit INTEA

```

```

.MODEL small
.DATA
RES        DB    ?
.CODE
.STARTUP
CALL      init_8255
MOV       SI,    0
ciclo:    MOV     DX,    80H
          MOV     CX,    5881          ; (4+(17*(n-1))+5+4)*10^-6=0,1 → n=5881
ritar:    LOOP   ritar
          IN      AL,    DX
pa_tst:   MOV     AH,    0
          SHR     AL,    1
          JNC    pari
          MOV     AH,    1
pari:     MOV     CX,    8
check:    ROL     AL,    1
          JNC    zero
          INC    AH

```

```

zero:      LOOP   check
           AND    zero, 0001B
           CMP    zero, AH
           JZ     ok
           MOV    DX, 83H           ; ritrasmissione
           MOV    AL, SPC4
           OUT    DX, AL
           MOV    DX, 82H
           MOV    AL, 1100000
           OUT    DX, AL
           HALT   ; attesa ricezione nuovo interrupt
ok:        JMP    pa_tst
           ADD    RES, AL
           INC    SI
           CMP    SI, 10
           JNE   ciclo
           MOV    AL, RES
           MOV    CH, 10
           CBW
           DIV   CH
           MOV    DX, 81H
           OUT    DX, AL
           RET
           .EXIT

init_8255  PROC
           MOV    DX, 83H
           MOV    AL, CW
           OUT    DX, AL
           RET
init_8255  ENDP

interrupt  PROC
           MOV    DX, 80H
           IN     AL, DX
           MOV    DX, 83H
           MOV    AL, RPC4
           OUT    DX, AL
           IRET
interrupt  ENDP

END

```