

Politecnico di Torino - III Facoltà di Ingegneria
Esercitazioni del corso “Calcolatori Elettronici 2”
Ing. Paolo Bernardi – email: paolo.bernardi@polito.it

Esercitazione 9 – L’interfaccia di comunicazione seriale 8250

Argomenti trattati:

1. inizializzazione ed uso della periferica di comunicazione seriale 8250.
2. gestione di un sistema basato su comunicazione seriale.

- Per ogni esercizio di quest’esercitazione si consideri la tabella registro/indirizzo riportata di seguito:

	COM1	COM2
RBR, THR, DLRL	03F8H	02F8H
IER, DLRH	03F9H	02F9H
IIR	03FAH	02FAH
LCR	03FBH	02FBH
MCR	03FCH	02FCH
LSR	03FDH	02FDH
MSR	03FEH	02FEH

Esercizio 1: Inizializzare un dispositivo periferico di comunicazione seriale 8250 in modo che soddisfi le seguenti specifiche di funzionamento:

- sia mappato sulla porta COM1
- comunichi con una tasso di trasmissione di 9600 bps
- il pacchetto inviato contenga
 - o 8 bit di dato
 - o 1 bit di stop
- non sia utilizzata nessun tipo di parità
- le richieste di interrupt siano disabilitate

Possibile svolgimento:

```
; valori di inizializzazione
LCR EQU 0000011b ; LINE CONTROL REGISTER (DLAB 0)
LCR_D EQU 1000011b ; LINE CONTROL REGISTER (DLAB 1)
IER EQU 0000000b ; INTERRUPT ENABLE REGISTER
DLR EQU 000CH ; DIVISOR LATCH REGISTER
```

; funzione di inizializzazione dispositivo

```
init_proc PROC
    PUSH AX
    PUSH DX
    MOV DX, LCR_ADD
    MOV AL, 0000011b
    ;OUT DX, AL
    OR AL, 1000000b
    OUT DX, AL
    MOV DX, DLRL_ADD
    MOV AX, 000CH
    OUT DX, AL
    MOV DX, DLRH_ADD
    XCHG AL, AH
    OUT DX, AL
    MOV DX, LCR_ADD
    IN AL, DX
    AND AL, 01111111b
    ; MOV AL, 0000011b
    OUT DX, AL
    MOV DX, IER_ADD
```

```

MOV     AL,    00000000b
OUT     DX,    AL
POP     DX
POP     AX
RET
init_proc ENDP

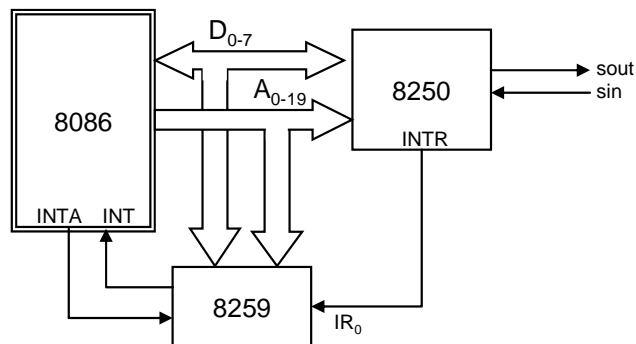
```

Esercizio 2: Inizializzare un dispositivo periferico di comunicazione seriale 8250 in modo che soddisfi le seguenti specifiche di funzionamento:

- sia mappato sulla porta COM1
- comunichi con una tasso di trasmissione di 2400 bps
- il pacchetto inviato contenga
 - o 5 bit di dato
 - o 1,5 bit di stop
 - o 1 bit per il controllo di parità dispari [ha valore 1 se il numero di 1 nella stringa di dato è dispari : 1011 → 1]
- siano abilitate le richieste di interrupt scatenate da errori di trasmissione o dalla ricezione di un dato.

Si scriva inoltre una procedura in grado di inviare il dato contenuto nella variabile VAR definita di tipo *byte*.

Possibile svolgimento:



```

; valori di inizializzazione
LCR_ADD     EQU    00001100b           ; LINE_CONTROL_REGISTER
IER_ADD     EQU    00000101b         ; INTERRUPT_ENABLE_REGISTER
DLR_ADD     EQU    0030H             ; DIVISOR_LATCH_REGISTER

; funzione di inizializzazione dispositivo
init_proc   PROC
PUSH     AX
PUSH     DX
MOV      DX,    LCR_ADD
IN       AL,    DX
OR       AL,    10000000b
OUT      DX,    AL
MOV      DX,    DLR_ADD
MOV      AX,    0030H
OUT      DX,    AL
MOV      DX,    DLRH_ADD
XCHG    AL,    AH
OUT      DX,    AL
MOV      DX,    LCR_ADD
MOV      AL,    00001100b

```

```

        OUT    DX,    AL
        MOV    DX,    IER_ADD
        MOV    AL,    00000101b
        OUT    DX,    AL
        POP    DX
        POP    AX
        RET
init_proc ENDP

; funzione di trasmissione
invia_proc PROC
        PUSH  AX
        PUSH  DX
        MOV   DX,    LSR_ADD
        IN   AL,    DX
        TEST  AL,    00100000b
        JZ   TRerror
        MOV   DX,    THR_ADD
        MOV   AL,    VAR
        OUT   DX,    AL
        JMP   fine
TRerror: ; manage error
fine:    POP    DX
        POP    AX
        RET
Invia_proc ENDP

```

Esercizio 3: Dato il sistema a microprocessore in figura, lo si configuri in modo ogni 50 valori ricevuti, ne calcoli la media e la invii. La comunicazione è seriale ed ha le seguenti caratteristiche:

- tasso di trasmissione 9600 bps
- 8 bit di dato
- 1 bit di stop
- 1 bit per il controllo di parità pari [1010000 → 1]

Possibile svolgimento:

```

; valori di inizializzazione
ICW1_8259 EQU 00010011b
ICW2_8259 EQU 00100000b
ICW4_8259 EQU 00000001b
OCW2_8259 EQU 00010000b
LCR_8250  EQU 00011011b
DLR_8250  EQU 000CH
IER_8250  EQU 00000001b

; main program
        .MODEL small

INT_VEC SEGMENT AT 0 ; istanziazione Interrupt Vector Table
        ORG 80H
IR0 DD ?
INT_VEC ENDS

        .STACK
        .DATA
VETT DB 50 dup (?)
        .CODE
        .STARTUP
        CLI ; disabilita gli interrupt
        CALL INIT_INTER
        CALL INIT_8259

```

```

        CALL INIT_8250
        STI                                ; abilita nuovamente gli interrupt
        MOV  DI, 0
        LEA  SI, VETT
WAIT:   CMP  DI, 50
        JNE  WAIT
        CLI                                ; disabilita gli interrupt
        LEA  SI, VETT
        CALL MEDIA
        CALL SPEDISCI_8250
        LEA  SI, VETT
        MOV  DI, 0
        STI
        JMP  WAIT
        RET
        .EXIT

```

; procedura di inizializzazione della Interrupt Vector Table

```

INIT_INTER PROC
        PUSH DS
        PUSH AX
        PUSH BX
        XOR  AX
        MOV  DS, AX
        MOV  AX, OFFSET inter_event
        LEA  BX, IR0
        MOV  [BX], AX
        MOV  AX, SEG inter_event
        MOV  [BX+2], AX
        POP  BX
        POP  AX
        POP  DS
        RET
INIT_INTER ENDP

```

; procedura di inizializzazione del dispositivo 8259

```

INIT_8259 PROC
        MOV  DX, 40H
        MOV  AL, ICW1
        OUT  DX, AL
        MOV  DX, 41H
        MOV  AL, ICW2_8259
        OUT  DX, AL
        MOV  AL, ICW4_8259
        OUT  DX, AL
        RET
INIT_8259 ENDP

```

; procedura di inizializzazione del dispositivo 8250

```

INIT_8250 PROC
        PUSH AX
        PUSH DX
        MOV  DX, LCR
        IN  AL, DX
        OR  AL, 10000000b
        OUT  DX, AL
        MOV  DX, DLRL
        MOV  AX, 000CH
        OUT  DX, AL
        MOV  DX, DLRH
        XCHG AL, AH
        OUT  DX, AL
        MOV  DX, LCR

```

```

MOV AL, LCR_8250
OUT DX, AL
MOV DX, IER
MOV AL, IER_8250
OUT DX, AL
POP DX
POP AX
RET
INIT_8250 ENDP

```

```

; procedura di elaborazione della media dei 50 numeri ricevuti

```

```

MEDIA PROC
PUSH DI
PUSH SI
MOV DI, 0
XOR AX, AX
SOMMA: ADD AL, [SI]
JNC NUOVO
INC AH
NUOVO: INC SI
INC DI
CMP DI, 50
JNE SOMMA
MOV BL, 50
DIV BL
POP SI
POP DI
RET
MEDIA ENDP

```

```

; procedura di invio dati seriale

```

```

SEND_8250 PROC
PUSH AX
PUSH DX
PUSH BX
MOV BX, AX
MOV DX, LSR
ciclo1: IN AL, DX
TEST AL, 00100000b
JZ ciclo1
MOV DX, THR
MOV AL, BL
OUT DX, AL
ciclo2: IN AL, DX
TEST AL, 00100000b
JZ ciclo2
MOV DX, THR
MOV AL, BH
OUT DX, AL
POP BX
POP DX
POP AX
RET
SEND_8250 ENDP

```

```

; procedura di interrupt

```

```

INTERRUPT_P PROC
PUSH AX
PUSH DX
MOV DX, IIR
IN AL, DX
CMP AL, 00000100b
JNE END

```

```
        MOV     DX,    RBR
        IN      AL,    DX
        MOV     [SI], AL
        INC     SI
        INC     DI
end:     MOV     DX,    OCW2
        MOV     AL,    OCW2_8259
        OUT    DX,    AL
        IRET
INTERRUPT_P ENDP

        END
```