

Le comunicazioni seriali

M. Rebaudengo - M. Sonza Reorda

Politecnico di Torino
Dip. di Automatica e Informatica

1

M. Rebaudengo, M. Sonza Reorda

Trasmissione di Dati

La trasmissione di dati implica lo scambio di informazioni digitali tra due sistemi o dispositivi.

La distanza tra i due sistemi comunicanti può variare da poche decine di centimetri fino a diverse centinaia di chilometri.

2

M. Rebaudengo, M. Sonza Reorda

Modi di comunicazione

Esistono due diversi modi di trasferire i dati:

- comunicazione *parallela*
- comunicazione *seriale*.

La differenza tra i due tipi di comunicazioni sta principalmente nel numero di segnali che connettono il trasmettitore al ricevitore.

3

M. Rebaudengo, M. Sonza Reorda

Comunicazione parallela

Quando la distanza tra i dispositivi che comunicano è breve, il costo della linea di trasmissione è minimo.

In questi casi le comunicazioni avvengono in maniera parallela.

La comunicazione parallela garantisce tempi di trasmissione più veloci.

4

M. Rebaudengo, M. Sonza Reorda

Comunicazione seriale

Quando la distanza tra i dispositivi è superiore al metro, il costo della linea di trasmissione diventa un parametro molto importante.

In particolare quando si fa uso di linee di comunicazione affittate (quali le linee telefoniche), i cavi in fibra ottica o le linee via satellite, l'unica modalità di comunicazione economicamente accettabile è quella seriale.

La comunicazione seriale è più lenta, ma più economica della comunicazione parallela.

5

M. Rebaudengo, M. Sonza Reorda

Modi operativi

Quando due dispositivi sono in comunicazione tra di loro esistono tre possibili modi in cui la comunicazione può svolgersi:

- *Simplex*: i dati sono trasmessi in un'unica direzione;
- *Half Duplex*: la comunicazione è bidirezionale, ma avviene su un'unica linea;
- *Full Duplex*: la comunicazione avviene attraverso due linee, una per ciascuna direzione.

6

M. Rebaudengo, M. Sonza Reorda

Bps e Baud

Le unità di misura usate per la velocità di trasmissione su una linea seriale sono:

- *bps*: numero di bit trasmessi per secondo;
- *baud*: numero di transizioni trasmesse per secondo.

Solo se il segnale trasmesso può assumere due soli possibili stati, allora baud e bps sono equivalenti.

Esempio

Un segnale a 300 baud che contiene 4 bit di informazione per ogni bit cell ha una velocità di trasmissione pari a 1200 bps.

7

M. Rebaudengo, M. Sonza Reorda

Sincronismi in ricezione

In una trasmissione seriale, il dispositivo ricevente, per poter decodificare ed interpretare correttamente i dati ricevuti, deve conoscere:

- la durata di ciascuna cella di bit (*clock synchronism*);
- l'inizio e la fine di ogni gruppo di bit, o carattere (*character synchronism*);
- l'inizio e la fine di ogni blocco di informazione (*frame synchronism*).

8

M. Rebaudengo, M. Sonza Reorda

Circuiti di controllo della trasmissione

I dati sono trasmessi serialmente in caratteri di n bit.

All'interno di ciascun dispositivo i dati sono memorizzati in forma parallela.

I dispositivi di controllo della trasmissione devono svolgere le seguenti funzioni:

- **conversione parallelo-seriale di ciascun elemento per prepararlo alla trasmissione**
- **conversione seriale-parallela di ciascun elemento ricevuto**
- **sincronizzazione in fase ricevente**
- **controllo di eventuali errori di comunicazione.**

9

M. Rebaudengo, M. Sonza Reorda

Tipi di Comunicazione Seriale

La comunicazione seriale può essere di due tipi:

- **sincrona**
- **asincrona.**

10

M. Rebaudengo, M. Sonza Reorda

Trasmissione Asincrona

In questo caso gli intervalli di tempo tra i caratteri trasmessi hanno lunghezza variabile.

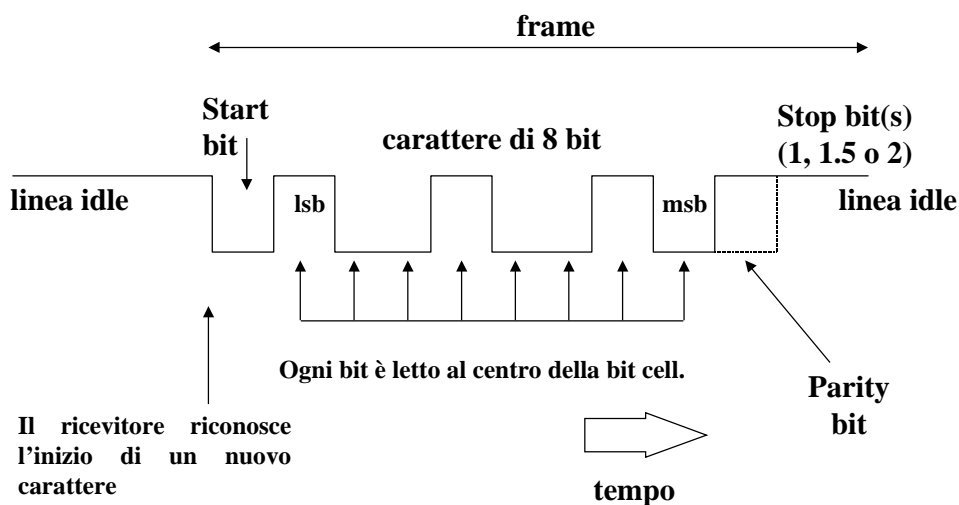
Ne consegue che ogni carattere è trasmesso indipendentemente dagli altri ed il ricevitore si deve risincronizzare all'inizio di ciascun carattere ricevuto.

Tra un carattere ed il successivo il segnale sulla linea di trasmissione è *idle* (off).

11

M. Rebaudengo, M. Sonza Reorda

Trasm issione Asincrona



12

M. Rebaudengo, M. Sonza Reorda

Funzionamento

All'atto della programmazione, si comunica al dispositivo ricevente il *fattore di scalamento* K tra il suo clock ed il clock di ricezione.

Dall'istante della prima transizione $1 \rightarrow 0$ il ricevitore lascia passare $K/2$ colpi di clock, e poi campiona il segnale di dato.

Se questo è basso, il bit viene interpretato come bit di start, altrimenti la transizione viene considerata fasulla.

A partire dal bit di start, il ricevitore campiona il segnale di dato ogni K colpi di clock.

Una volta letti i bit di dato, il ricevitore campiona i bit di stop, in modo da verificare la corretta terminazione del carattere.

La polarità dei bit di start e di stop è differente.

13

M. Rebaudengo, M. Sonza Reorda

Bit per carattere

La dimensione di ciascun carattere (in bit) fa parte dei parametri di connessione.

Solitamente si utilizzano caratteri di 5, 6, 7 o 8 bit.

È possibile prevedere un bit aggiuntivo per ciascun carattere, che contenga un'informazione di *parità*, che può venire utilizzata per rilevare eventuali errori di trasmissione.

14

M. Rebaudengo, M. Sonza Reorda

Tasso di Trasmissione

Per trasmettere ciascun carattere sono necessari 10 (1 bit di start, 8 bit di dato e 1 bit di stop) o 11 bit (1 + 8 + 2 bit).

Supponendo di trasmettere ad una frequenza di 1200 bit al secondo, il tasso di trasmissione è quindi pari a 1200/11 byte al secondo.

15

M. Rebaudengo, M. Sonza Reorda

Trasmissione Sincrona

In questo caso gli intervalli di tempo tra i caratteri trasmessi hanno lunghezza multipla della durata della trasmissione di un carattere.

I clock del trasmettitore e del ricevitore devono quindi essere in sincronismo per lunghi periodi di tempo.

È necessaria una opportuna informazione di sincronizzazione che si affianchi all'informazione dei dati e che consenta al ricevitore di campionare lo stato della linea in modo da prelevare correttamente i bit di dato (*clock synchronism*).

16

M. Rebaudengo, M. Sonza Reorda

Clock Synchronism

Comunemente si utilizzano 2 tecniche di sincronizzazione:

- il trasmettitore genera ed invia al ricevitore un unico segnale contenente sia le informazioni di temporizzazione che i dati; il ricevitore utilizza questo segnale sia per rifasare il proprio clock tramite un circuito PLL (*Phase Lock Loop*), sia per estrarre i dati
- il trasmettitore ed il ricevitore si scambiano i dati su un canale ed il segnale di temporizzazione su un altro canale.

17

M. Rebaudengo, M. Sonza Reorda

Codifica Manchester

Un esempio di segnale che contiene sia informazione di temporizzazione che di dato è quello ottenuto dal trasmettitore attraverso la codifica Manchester.

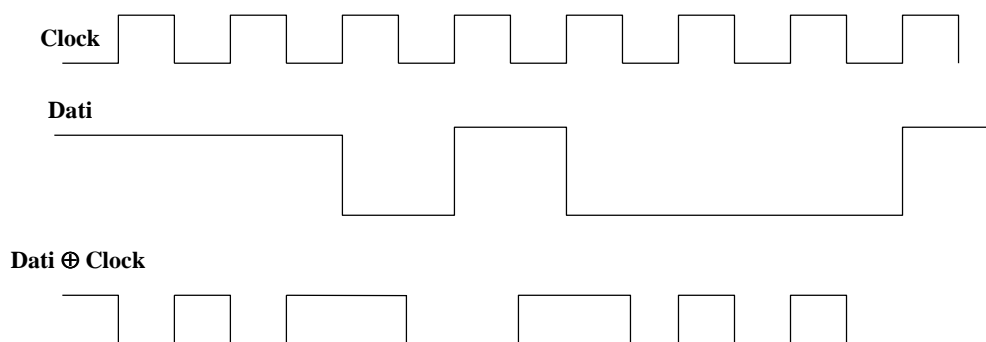
Tale codifica si ottiene mediante l'operazione di EXOR tra il segnale del clock ed il segnale che rappresenta i dati.

Il ricevitore può estrarre i dati mediante un'analogica operazione di EXOR tra il segnale ricevuto ed il segnale di clock riprodotto tramite circuito PLL.

18

M. Rebaudengo, M. Sonza Reorda

Codifica Manchester



19

M. Rebaudengo, M. Sonza Reorda

Caratteri di Sincronismo

Durante i periodi di inattività, il trasmettitore invia comunque dei caratteri, denominati *caratteri di sincronismo*.

Tali periodi hanno quindi una durata multipla di quella necessaria per la trasmissione di un carattere.

Uno o più caratteri di sincronismo vengono anche inviati all'inizio della trasmissione, per permettere la sincronizzazione del ricevente.

20

M. Rebaudengo, M. Sonza Reorda

EIA RS-232

È lo standard più usato per l'interfacciamento seriale tra dispositivi.

È stato definito dalla *Electrical Industry Association* (EIA) e concerne le caratteristiche elettriche e la definizione delle linee con cui avvengono le comunicazioni.

In Europa lo stesso standard prende il nome di *Raccomandazione V.24* del CCITT (*Comité Consultatif International Télégraphique et Téléphonique*).

Si fa di solito riferimento ai connettori a 25 pin.

21

M. Rebaudengo, M. Sonza Reorda

DCE e DTE

Lo standard RS-232 definisce due tipi di dispositivi:

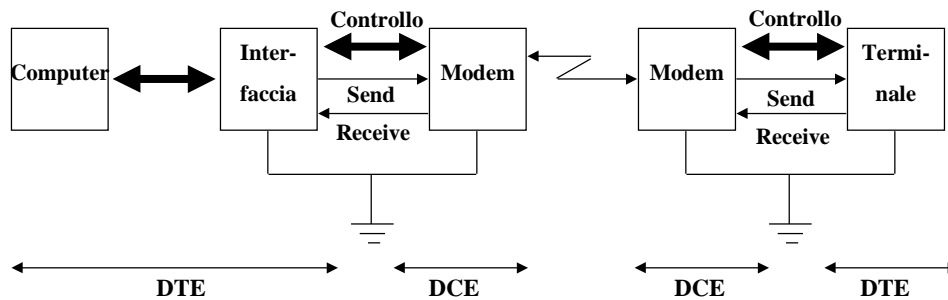
- dispositivi di comunicazione: *Data Carrier Equipment* (o *DCE*)
- dispositivi terminali: *Data Terminal Equipment* (o *DTE*).

Un DCE è tipicamente costituito da un modem, mentre un DTE è tipicamente un terminale o un computer o un video.

22

M. Rebaudengo, M. Sonza Reorda

Comunicazioni Seriali su lunghe distanze



23

M. Rebaudengo, M. Sonza Reorda

Segnali Base

In una comunicazione full duplex secondo lo standard RS-232 si utilizzano i seguenti segnali :

- **RTS** (Request To Send): comunica che il trasmettitore è pronto ad inviare dati;
- **CTS** (Clear To Send): il ricevitore è pronto a ricevere dati;
- **TXD** (Transmit Data): segnale di dato in trasmissione;
- **DCD** (Data Carrier Detect): attivato dal DCE quando riconosce un segnale *portante* dall'altro lato della comunicazione
- **DTR** (Data Terminal Ready): attivato all'accensione del DTE, indica che è pronto ad operare
- **RXD** (Receive Data - Input): segnale di dato in ricezione;
- **DSR** (Data Set Ready): il DCE informa il DTE che è acceso, ha completato tutte le operazioni per una connessione e può comunicare con il DTE
- **RI** (Ring Indicator): informa il DTE che una richiesta di connessione è giunta al DCE
- **GND** (Ground): segnale di massa.

24

M. Rebaudengo, M. Sonza Reorda

Segnali di handshake

DTR e DSR servono a stabilire la connessione tra DCE e DTE: vengono attivati all'accensione rispettivamente dal DTE e dal DCE quando questi sono pronti ad utilizzare il collegamento.

RTS e CTS servono ad eseguire il trasferimento dei dati: quando il DTE vuole trasmettere, attiva RTS e aspetta che CTS divenga attivo; terminata la trasmissione il DTE disattiva RTS ed aspetta che CTS venga disattivato.

25

M. Rebaudengo, M. Sonza Reorda

Comunicazione DTE/DCE

Inizializzazione

- il DTE attiva il segnale *Request To Send* verso il DCE
- il DCE risponde con il segnale *Data Set Ready* (dopo aver completato tutte le operazioni di preparazione).

26

M. Rebaudengo, M. Sonza Reorda

Comunicazione DTE/DCE

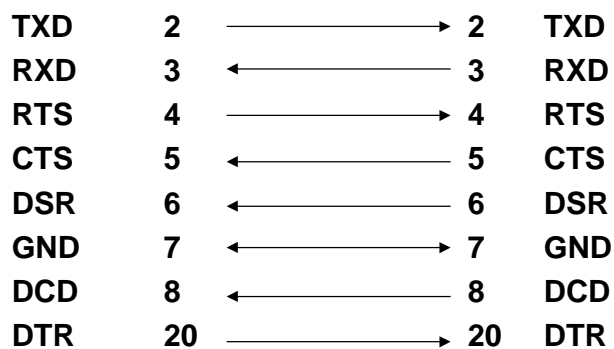
Trasmissione

- il DTE attiva il segnale *Data Terminal Ready* verso il DCE
- il DCE risponde attivando il segnale *Clear To Send*
- la trasmissione verso il modem può iniziare sulla linea *Transmitted Data*.

27

M. Rebaudengo, M. Sonza Reorda

Esem pio di Connessione



DTE

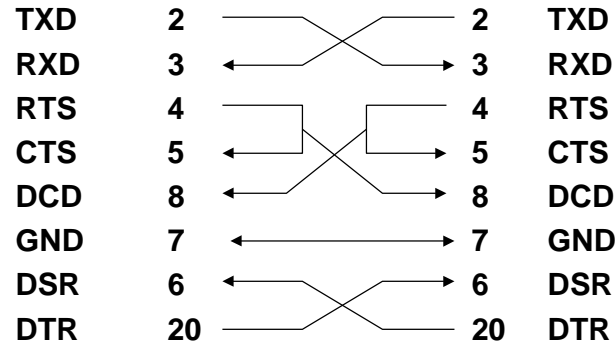
DCE

28

M. Rebaudengo, M. Sonza Reorda

Connessione *null-modem* tra 2 DTE

Qualora si debbano connettere 2 DTE, il cavo di connessione deve effettuare scambi di collegamenti ed è detto *null-modem*.



29

DTE**DTE**

M. Rebaudengo, M. Sonza Reorda

UART

I circuiti di interfaccia per gestire la comunicazione asincrona sono noti come *Universal Asynchronous Receiver and Transmitter (UART)*.

L'attributo di universale sta a significare che il dispositivo è programmabile e dunque l'utente può specificare le caratteristiche operative richieste inviando una opportuna parola di controllo.

30

M. Rebaudengo, M. Sonza Reorda

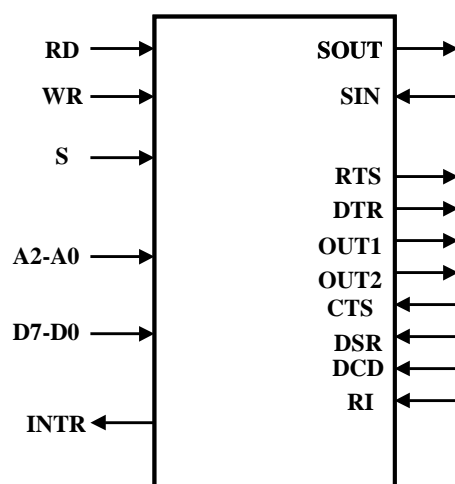
8250

- L'8250 è l'interfaccia seriale di tipo UART della National Semiconductor
- Permette di generare internamente il bit-rate nell'intervallo di valori previsti dallo standard RS-232
- Comunicazione Asincrona
- Adotta un fattore di scalamento pari a 16
- Riconoscimento di errori di trasmissione.

31

M. Rebaudengo, M. Sonza Reorda

Schema dell'8250



32

M. Rebaudengo, M. Sonza Reorda

Pin

- **RD: Read**
- **WR: Write**
- **S: Select**
- **A0-A2: Address Bits**
- **D0-D7: Data Bus**
- **INTR: Interrupt Request**
- **SIN: Serial Input**
- **SOUT: Serial Output**
- **OUT1, OUT2: User Outputs**
- **RTS: Request To Send**
- **CTS: Clear To Send**
- **DCD: Data Carrier Detect**
- **DTR: Data Terminal Ready**
- **DSR: Data Set Ready**
- **RI: Ring Indicator**

33

M. Rebaudengo, M. Sonza Reorda

Registri interni

- **Receiver Buffer Register (RBR)**
- **Transmitter Holding Register (THR)**
- **Line Status Register (LSR)**
- **Line Control Register (LCR)**
- **Divisor Latch Registers (DLRs)**
- **Modem Status Register (MSR)**
- **Modem Control Register (MCR)**
- **Interrupt Enable Register (IER)**
- **Interrupt Identification Register (IIR)**

34

M. Rebaudengo, M. Sonza Reorda

Indirizzi dei registri

	A2	A1	A0	DLAB	Accesso
RBR	0	0	0	0	lettura
THR	0	0	0	0	scrittura
DLR-LSB	0	0	0	1	lettura e scrittura
DLR-MSB	0	0	1	1	lettura e scrittura
IER	0	0	1	0	lettura e scrittura
IIR	0	1	0	-	lettura
LCR	0	1	1	-	lettura e scrittura
MCR	1	0	0	-	lettura e scrittura
LSR	1	0	1	-	lettura
MSR	1	1	0	-	lettura

35

M. Rebaudengo, M. Sonza Reorda

DLAB

Il bit denominato “DLAB” (Divisor Latch Access Bit) è un bit di indirizzo aggiuntivo implementato nel bit 7 del registro LCR.

Esso è normalmente tenuto a 0, tranne per la programmazione del *Divisor Latch Register*.

36

M. Rebaudengo, M. Sonza Reorda

Receiver Buffer Register

Contiene il dato ricevuto dall'interfaccia seriale tramite il segnale SIN.

Il bit meno significativo del registro contiene il bit ricevuto per primo.

37

M. Rebaudengo, M. Sonza Reorda

Transmitter Holding Register

Contiene il dato da trasmettere tramite il piedino SOUT.

Il bit meno significativo del registro contiene il bit che sarà trasmesso per primo.

38

M. Rebaudengo, M. Sonza Reorda

Line Status Register

È accessibile esclusivamente in lettura. Contiene le informazioni relative allo stato dell'interfaccia.

X	X	EO	B	FE	PE	OE	FI
---	---	----	---	----	----	----	----

- **FI (Full Input) = 1:** un nuovo carattere è stato ricevuto dall'interfaccia ed è disponibile in RBR. Posto a 0 quando il processore legge RBR.
- **OE (Overrun Error) = 1:** errore di Overrun.
- **PE (Parity Error) = 1:** errore di Parità.
- **FE (Frame Error) = 1:** errore di Frame.
- **B (Break) = 1:** ricezione di un break, stato in cui la linea di dato è tenuta a 0 per un tempo superiore al tempo necessario per trasmettere una parola (incluso il tempo di trasmissione del bit di start, stop e parità).
- **EO (Empty Output) = 1:** il dato contenuto nel registro THR è stato trasmesso. Posto a 0 quando il processore scrive in THR.

39

M. Rebaudengo, M. Sonza Reorda

Flag di Errore

I flag di errore nel registro LSR sono:

- *parity error*: errore di trasmissione rilevato attraverso il bit di parità;
- *framing error*: mancata ricezione di un bit di stop;
- *overrun error*: il ricevitore ha ricevuto un nuovo dato senza che la CPU abbia letto il dato contenuto nel buffer ricevente.

I flag di errore sono posti a 0 dopo che il processore legge il registro LSR.

La segnalazione di tali errori non blocca il funzionamento dell'UART, ma è compito della CPU operare le necessarie azioni correttive.

40

M. Rebaudengo, M. Sonza Reorda

Line Control Register

Definisce il formato del frame (sia in trasmissione che in ricezione).

DLAB	BE	SP	PS	P	STOP	L2	L1
------	----	----	----	---	------	----	----

- **L2, L1 (Number of Data Bits)** definiscono il numero di bit di dato (bit per carattere)
 - L2= 0 L1 =0: 5 bit L2= 0 L1 =1: 6 bit
 - L2= 1 L1 =0: 7 bit L2= 1 L1 =1: 8 bit
- **STOP (Number of Stop Bits)** Numero di bit di stop.
 - LE=0: 1 bit di stop
 - LE=1: 1.5 bit di stop (se il carattere è di 5 bit)
2 bit di stop (se il carattere è di 6, 7 o 8 bit)

41

M. Rebaudengo, M. Sonza Reorda

Line Control Register (II)

- **P (Parity)= 1:** presenza di un bit di parità
 - Se **PE = 1** i bit **SP (Sticky Parity)** e **PS (Parity Select)** assumono il seguente significato:
 - * **SP=0 PS=0:** bit di parità a 1 se nel dato c'è un numero dispari di bit a 1
 - * **SP=0 PS=1:** bit di parità a 1 se nel dato c'è un numero pari di bit a 1
 - * **SP=1 PS=0:** bit di parità = 1
 - * **SP=1 PS=1:** bit di parità = 0

42

M. Rebaudengo, M. Sonza Reorda

Line Control Register (III)

- **BE (Break Enable) = 1:** trasmissione di un segnale di *break*: il piedino SOUT si porta a 0 e rimane in questo stato fino a che BE non viene posto a 0.
- **DLAB (Divisor Latch Access Bit).** DLAB = 1 accesso ai registri DLR.

43

M. Rebaudengo, M. Sonza Reorda

Registri DLR

I due registri DLR (*Divisor Latch Register*) permettono di determinare il bit-rate (unico per la trasmissione e per la ricezione).

L'interfaccia ricava il bit-rate dividendo la sua frequenza di pilotaggio per 16 e per il numero contenuto nei due registri DLR (detto *costante di tempo*).

Il valore della costante di tempo è quindi calcolato come:

Bit-rate = frequenza di pilotaggio / (16 * costante di tempo)

44

M. Rebaudengo, M. Sonza Reorda

Costanti di tempo

In un Personal Computer la frequenza di pilotaggio vale 1.8432 Mhz.

La tabella seguente specifica le costanti di tempo relative ai valori tipici di bit-rate

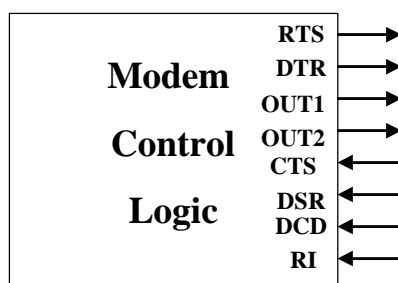
bit-rate	costante di tempo (hex)
110	0417
300	0180
600	00C0
1200	0060
2400	0030
4800	0018
9600	000C
19200	0006

45

M. Rebaudengo, M. Sonza Reorda

Interfaccia verso il Modem

L'8250 ha un insieme di segnali di controllo utilizzabili per interfacciarsi con un Modem.



46

M. Rebaudengo, M. Sonza Reorda

Modem Control Register

Attraverso il registro MCR il programmatore può forzare lo stato dei 4 piedini di uscita dell'interfaccia.

0	0	0	0	OUT2	OUT1	RTS	DTR
---	---	---	---	------	------	-----	-----

47

M. Rebaudengo, M. Sonza Reorda

Modem Status Register

Il registro MSR permette di avere informazioni sullo stato e sulle variazioni dei piedini di input DCD, RI, DSR e CTS.

DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS
-----	----	-----	-----	------	-----	------	------

I bit 0-3 se ad 1 indicano che lo stato del rispettivo segnale è cambiato nell'intervallo di tempo che va dalla precedente lettura all'attuale lettura

I bit 4-7 riflettono lo stato del rispettivo segnale: se 1 indica che il piedino ha assunto il valore 0.

48

M. Rebaudengo, M. Sonza Reorda

Interrupt Enable Register

Il registro IER permette di abilitare le richieste di interruzioni.

0	0	0	0	SINP	ERBK	TBE	RDR
---	---	---	---	------	------	-----	-----

- RDR (*Received Data Ready*) = 1: abilitazione ad effettuare una richiesta di interruzione quando un byte è pronto in RBR
- TBE (*Transmitter Buffer Empty*) = 1: abilitazione ad effettuare una richiesta di interruzione quando THR è vuoto
- ERBK (*Error & Break*) = 1: abilitazione ad effettuare una richiesta di interruzione quando viene rilevato un errore o un segnale di break
- SINP (*Serial Input*) = 1: abilitazione ad effettuare una richiesta di interruzione quando uno dei segnali di input del protocollo RS-232 (CTS, DSR, DCD, RI) cambia di stato
- I bit 4-7 sono sempre a 0.

49

M. Rebaudengo, M. Sonza Reorda

Interrupt Identification Register

Il registro IIR è accessibile esclusivamente in lettura e permette di identificare lo stato delle richieste di interruzioni

0	0	0	0	0	ID1	ID0	PND
---	---	---	---	---	-----	-----	-----

- PND (*Pending Bit*) = 1: nessun interrupt è pendente; PND = 0 interrupt pendente
- ID1, ID0 (*Identify Bits*): bit di identificazione delle richieste di interruzione
 - ID1 = 0 ID0 = 0: cambio di un segnale RS-232 (priorità 3: min)
 - ID1 = 0 ID0 = 1: THR vuoto (priorità 2)
 - ID1 = 1 ID0 = 0: RBR pieno (priorità 1)
 - ID1 = 1 ID0 = 1: errore in ricezione o break (priorità 0: max)

Le richieste di interruzione a priorità più basse sono bloccate se è pendente una richiesta a priorità più alta.

50

- I bit 3-7 devono essere posti a 0.

M. Rebaudengo, M. Sonza Reorda

Tipi di UART

8250: baud rate massimo: 9600

16450: baud rate massimo: 115200

16550: buffer FIFO da 16 byte per memorizzare dati ricevuti o da trasmettere

16650: buffer FIFO da 32 byte

16750: buffer FIFO da 64 byte.

Le diverse versioni sono compatibili software.

51

M. Rebaudengo, M. Sonza Reorda

Interfacce seriali nei PC

Nei PC viene utilizzata un'interfaccia per le comunicazioni seriali asincrone compatibile software con l'8250.

L'interfaccia prevede la presenza di amplificatori che servono ad adattare i livelli elettrici propri dell'interfaccia (standard TTL) con i livelli elettrici richiesti dallo standard RS-232.

52

M. Rebaudengo, M. Sonza Reorda

Indirizzi dei registri

L'interfaccia seriale può essere posizionata in due diversi gruppi di indirizzi: COM1 e COM2.

<i>Registro</i>	<i>COM1</i>	<i>COM2</i>
RBR,THR,DLR-LSB	03F8h	02F8h
IER,DLR-MSB	03F9h	02F9h
IIR	03FAh	02FAh
LCR	03FBh	02FBh
MCR	03FCh	02FCh
LSR	03FDh	02FDh
MSR	03FEh	02FEh

53

M. Rebaudengo, M. Sonza Reorda

Esempio

Si riporta un frammento di codice che esegue una procedura di inizializzazione dell'interfaccia 8250 con riferimento alla configurazione COM1.

Si supponga che i parametri della configurazione seriale siano i seguenti:

- bit-rate: 9600 bit/s
- bit per carattere: 8
- parità: disabilitata
- bit di stop: 1
- richieste di interruzione: tutte disabilitate

54

M. Rebaudengo, M. Sonza Reorda

```

rbr      equ    03f8h
dlr_lsb  equ    03f8h
dlr_lsb  equ    03f9h
lcr      equ    03fbh
ier      equ    03f9h
ini_com  proc near          push  ax
                    push  dx
                    mov   dx, lcr
                    in   al, dx      ; lettura del contenuto di LCR
                    or   al, 80h    ; set DLAB
                    out  dx, al     ; scrittura in LCR
                                        ; bit-rate: 9600

                    mov   ax, 000ch
                    mov   dx, dlr_lsb
                    out  dx, al
                    mov   al, ah
                    inc  dx
                    out  dx, al

```

55

M. Rebaudengo, M. Sonza Reorda

```

; 1 bit di stop, 8 bit/carattere, parita' disabilitata
; DLAB = 0
mov   al, 03h
mov   dx, lcr
out  dx, al
; disabilitazione delle richieste di interruzione
mov   al, 00h
mov   dx, ier
out  dx, al
pop   dx
pop   ax
ret
ini_com  endp

```

56

M. Rebaudengo, M. Sonza Reorda