

Le comunicazioni seriali

Lo scambio di informazioni tra due dispositivi o periferici deve essere effettuata attraverso comunicazioni digitali. Si può avere la necessità far comunicare diversi dispositivi separati da brevi o lunghe distanze. L'esigenza di trasmettere i dati ha dettato la divisione di due categorie di comunicazione a seconda della tipologia di canale di comunicazione:

- comunicazione *seriale*: le informazioni sono comunicate una di seguito all'altra.
- comunicazione *parallela*: Questa categoria prevede di poter inviare / ricevere più segnali contemporaneamente, a scapito di costi, architetture e protocolli più complessi.

La differenza tra i due tipi di comunicazioni sta principalmente nel numero di segnali che connettono il trasmettitore al ricevitore. 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 la transazione è composta da un solo bit, allora baud e bps sono equivalenti.

Modi operativi

I dispositivi possono essere messi in comunicazione in modalità diverse a seconda delle caratteristiche intrinseche del canale:

- *Simplex*: la comunicazione è unidirezionale;
- *Half Duplex*: la comunicazione è bidirezionale, ma avviene su un'unica linea; Il canale quindi è monodirezionale ma il verso può essere invertito.
- *Full Duplex*: la comunicazione avviene attraverso due linee, una per ciascuna direzione.

Un esempio di trasmissione:

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

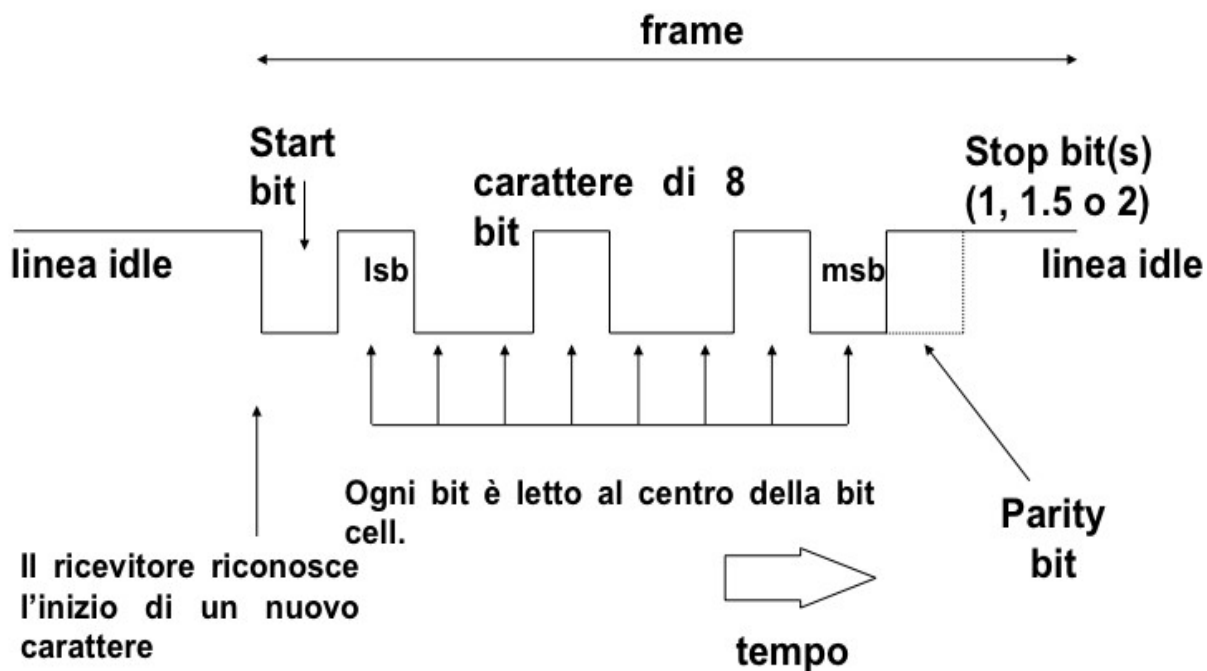


Figura 1

La frequenza di trasmissione dati (fattore di scalamento K) è concordata a priori tra ricevitore e trasmettitore e all'atto della programmazione, si comunica al dispositivo ricevente 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 (esempio in figura 1). 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. Il frame di trasmissione è composto:

- start-bit
- payload di 8 bit
- stop-bit

in base al valore dello start-bit (idle o controllo linea) il ricevitore si sincronizza rispetto alla trasmissione ponendosi in ascolto e acquisendo i bit trasmessi. I dati trasmessi sono privi di errori grazie ad un meccanismo di controllo implementato tramite bit di parità. L'8250 identifica anche errori riguardo la presenza di idle troppo lunghi o la mancanza di stop-bit.

*Le comunicazioni seriali: **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.

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.

Le comunicazioni seriali e gli 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.

Il dispositivo 16550 UART è l'attuale circuito integrato che permette di implementare una interfaccia per la comunicazione seriale. Questo dispositivo è considerato lo standard sulle architetture IBM o sui Personal computer ed è spesso integrato sulla motherboard e collegato a dispositivi quali modem o stampanti. Come si vede nella figura 2 il dispositivo 16550 è il successore del dispositivo trattato nel corso.

INS8250 -> INS8250B

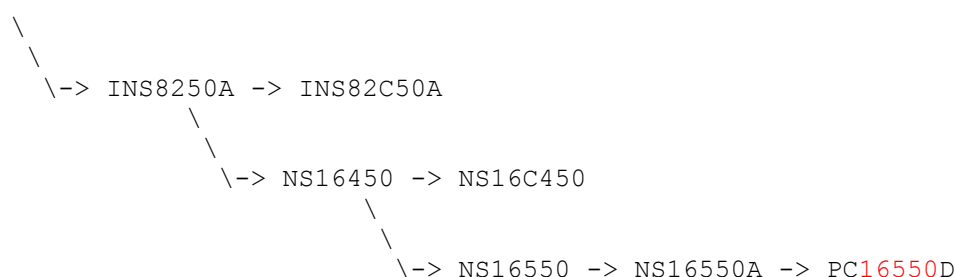


Figura 2:

Modelli 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.

Il software scritto per diverse versioni è compatibile!

Le comunicazioni seriali: 8250

- L'8250 è l'interfaccia standard di comunicazione 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.

Descrizione operativa: Piedinatura e registri

- RD: Lettura
- WR: Scrittura
- S: Selezione
- A0-A2: Bit di indirizzo
- D0-D7: Bus dati
- INTR: Richiesta Interrupt
- SIN: Ingresso Seriale
- SOUT: Uscita seriale
- OUT1, OUT2: Uscita utente
- RTS: Richiesta di invio
- CTS: Cancellazione di invio
- DCD: Data Carrier Detect
- DTR: Data Terminal Ready
- DSR: Data Set Ready
- RI: Ring Indicator

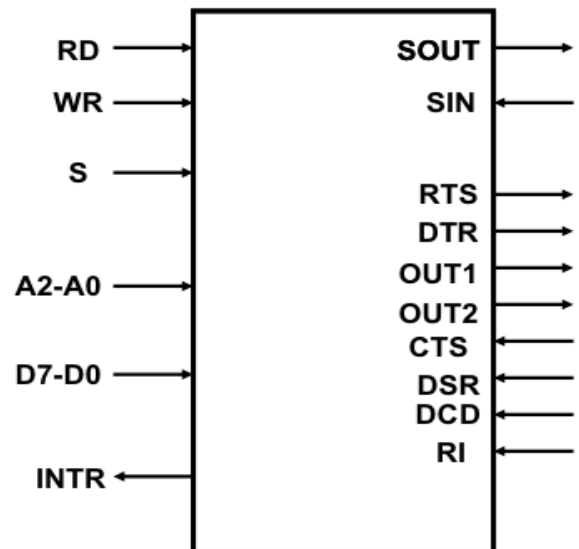


Figura 3:

Registri interni :

L'8250 contiene al suo interno 10 registri interni (figura 4): alcuni contengono dati ricevuti dall'esterno, altri controllano la trasmissione e gli interrupt. Poiché tre bit non sono sufficienti a indirizzare 10 registri si utilizza il DLAB.

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

	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

Figura 4:

dlab: Il bit denominato “DLAB”(DivisorLatchAccess 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 DivisorLatchRegister.

ReceiverBuffer Register: Contiene il dato ricevuto dall'interfaccia seriale tramite il segnale SIN pervenuto dal processore. Il bit meno significativo del registro contiene il bit ricevuto per primo.

TransmitterHolding Register: Contiene il dato da trasmettere tramite il piedino SOUT. Il bit meno significativo del registro contiene il bit che sarà trasmesso per primo.

Line Status Register(struttura in figura 5): È accessibile esclusivamente in lettura. Contiene le informazioni relative allo stato dell'interfaccia.

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

Figura 5:

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 livello basso quando il processore scrive in THR.

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 azione correttive.

Line Control Register (struttura in figura 6): Definisce il formato del frame(sia in trasmissione che in ricezione)



Figura 6:

- 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)

- 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

- 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.

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 attraverso l'espressione: $\text{Bit-rate} = \text{frequenza di pilotaggio} / (16 * \text{costante di tempo})$. In base al valore del DLR si sceglie il fattore di divisione per la comunicazione.

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)	costante di tempo (hex)
110		417
300		180
600		00C0
1200		60
2400		30
4800		18
9600		000C
19200		6

InterruptEnableRegister(struttura in figura 7): Il registro IER permette di abilitare le richieste di interruzioni selettivamente rispetto ad alcune condizioni di errore.

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

Figura 7:

- RDR (Received Data Ready) = 1: abilitazione ad effettuare una richiesta di interruzione quando un byte è pronto in RBR
- TBE (TransmitterBuffer 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.

InterruptIdentificationRegister(struttura in figura 8): 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: *Figura 8:*
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 (priorità3: min)

ID1 = 0 ID0 = 1: THR vuoto (priorità2)

ID1 = 1 ID0 = 0: RBR pieno (priorità1)

ID1 = 1 ID0 = 1: errore in ricezione o break (priorità0: max)

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

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

Indirizzi dei registri:

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

Registro	COM1	COM2
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

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 disabilite

Procedura in MASM:

; DEFINIZIONI DI COSTANTI

33 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; 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