

Comunicazione Seriale

Prof. Antonino Mazzeo

UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II

Corso di Laurea Specialistica in
Ingegneria Informatica





UART/USART

- Intel 8251A
- Standard Universal Synchronous/Asynchronous Receiver/Trasmitter (USART)
- Progettata per le comunicazioni con le famiglie di microprocessori INTEL
- Device Periferico programmabile da CPU



UART/USART Trasmissione

- USART accetta dati dalla CPU in formato PARALLELO e
- li converte in uno stream di dati seriali



UART/USART Ricezione

- USART riceve un flusso di dati seriali e li
- converte in formato parallelo per inviarli alla CPU



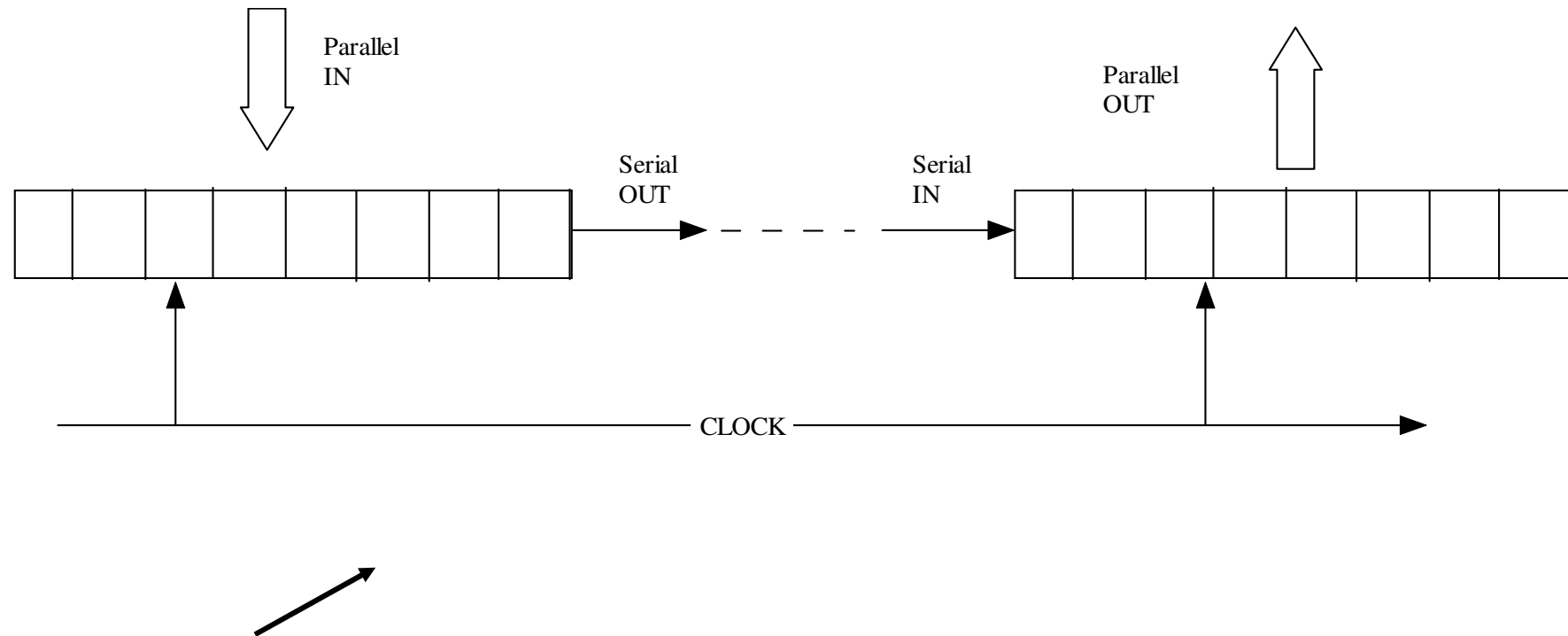
UART/USART

- USART segnala alla CPU se può accettare un nuovo carattere da trasmettere, oppure
- Se ha ricevuto un nuovo carattere che può comunicare alla CPU
- La CPU può leggere lo stato completo della USART in ogni istante
 - Data Transmission Error
 - Segnali di Controllo



Modalità di trasmissione

- Simplex: esiste una linea monodirezionale (1 trasmette e 1 riceve)
- Half Duplex: esiste una linea bidirezionale (1 trasmette e 1 riceve ma possono scambiarsi i ruoli)
- Full Duplex: esiste una linea bidirezionale entrambi possono trasmettere e ricevere contemporaneamente.

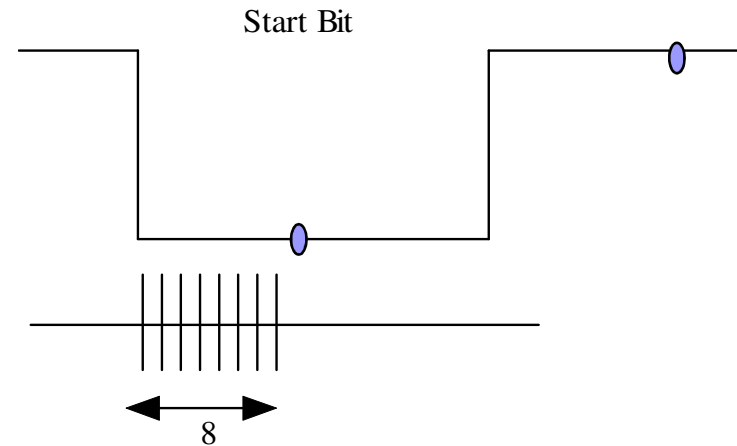


In realtà non si trasmette il clock ma il RX è dotato di oscillatori locali che generano Un clock di campionamento adeguato oppure sono in grado di sincronizzarsi con il Trasmettitore.

L'informazione elementare che viene trasmessa è "un carattere" codificato su massimo 8 bit (start, msg, controllo, coda)

Comunicazione Asincrona

- Il clock del RX viene settato ad una velocità molto superiore del TX (16x o 64x) in modo da non perdere la transizione 1->0 dello start bit,
- Poi il clock del RX rifasa il campionamento con un shift di 8 impulsi posizionandosi al centro del bit e campiona ogni 16 impulsi





Comunicazione Asincrona (2)

- Sono necessarie delle informazioni aggiuntive per controllare che non si sia persa la sincronizzazione:
 - 1 bit di parità per ogni carattere inviato,
 - Controllo di parità per l'intero messaggio,
 - Controlli di parità longitudinale, aggiungendo byte CHECK SUM



Comunicazione Sincrona

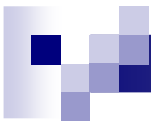
- Per aumentare le prestazioni e diminuire gli sprechi occorre:
 - Eliminare start e stop bits,
 - Eliminare bit di parità,
 - Rilievo errori più sicuro
- Nella comunicazione sincrona mandiamo solo messaggi “significativi” come flusso continuo di bit,
- la sincronizzazione viene fatta solo all’inizio oppure dopo un numero **PREDETERMINATO** di caratteri inviando una sequenza di caratteri speciali (BYTE SYNC)
- Per la correzione degli errori: si applica un codice ridondante sull’intero frame (stesso numero di bit indipendentemente dalla lunghezza del msg).



Tipologie di errori

- **OVERRUN**: il dato è stato copiato dallo shift register nel registro in prima che quest'ultimo fosse stato svuotato dalla CPU (dato perso!);
- **FRAMING**: si è persa la sincronizzazione;
- **PARITY**: è fallito il controllo sul bit di parità;

NOTA: tipicamente la comunicazione seriale avviene su canali molto rumorosi (linea telefonica).... Problema delle tensioni adeguate



Architettura di una UART/USART

■ 4 sezioni:

- ☐ Scambio dati e comandi con il PC,
- ☐ Trasmitter: contiene uno shift register ed un buffer temporaneo
- ☐ Receiver: contiene uno shift register ed un buffer temporaneo
- ☐ Rate generator: genera la frequenza di trasmissione



Modello di Programmazione

- Prima di iniziare le operazioni di trasmissione o ricezione l' 8251A deve essere programmato mediante un insieme di parole di controllo generate dalla CPU
- Queste istruzioni sono strutturate in
 - Una parola di MODE e
 - Una parola di COMMAND
- Entrambe scritte nel registro di comando secondo una sequenza opportuna



Modello di Programmazione

- Dopo un'operazione di RESET
- Il device considera come MODE instruction il primo comando ricevuto
- E i successivi comandi sono trattati come parole di COMMAND
- Oppure (nella modalità sincrona) come codice binario dei caratteri di sincronismo



Segnali di comunicazione

RS-232

- TD ed RD per trasmettere e ricevere dati,
- RTS(Request to Send): indica intenzione a trasmettere;
- CTS(Clear to Send): indica lo stato di pronto a ricevere;
- DSR (Data set Ready): Tx ed RX sono collegati in trasmissione
- DTR (Data Terminal Ready): pronto a comunicare



MODE Register

MODE | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |

*

* | | | | | | | | Trasmissione Asincrona
*

* | | | | | Non utilizzato
*

* | | | | 8 bit per dato
*

* | | | bit di parità
*

* | | tipo di parità dispari
*

* | 2 bit di stop
*

* #bit di sync in trasmissione asincrona
*



Attivare la trasmissione (1/2)

1. Definire la modalità di scambio (MODE word):
 - Asincrono (fattore molt del clock, lunghezza del char, tipo controllo parità, num di top bit);
 - Sincrono (num caratteri SYNC, codice del SYNC)
2. Nella COMMAND word: abilitare la Tx, chiedere un RESET



Attivare la trasmissione (2/2)

- Attivare TxRDY in attesa che il carattere da trasmettere sia nel buffer (inviato dalla CPU),
 - Il char viene trasferito nel serializzatore (se libero)
 - Si abilita il segnale TxC che scandisce la serializzazione
 - Ricomincia da capo con il prossimo carattere
- Se la CPU ritarda nel caricare il buffer vengono inviati dei SYNC oppure degli “IDLE”



Abilitare la Ricezione (1/2)

- Dopo aver inizializzato la periferica con una MODE word, abilitare nella COMMAND word il bit RxEN
 1. La periferica cerca lo start bit per sincronizzarsi e viene abilitato lo shift register a ricevere n bit
 2. Esegue il test sul frame arrivato
 3. Trasferisce il frame nel buffer ed alza il flag RxRDY (INT alla CPU), (si abbasserà quando il dato sarà letto dalla CPU) se RxRDY era già alto viene generato un OVERRUN ERROR.



Abilitare la Ricezione

- HUNT MODE : modalità di esecuzione in cui si riceve il carattere di sincronizzazione e lo si controlla per verificare sia uguale a quello previsto
- Una volta che il pattern di bit di sincronizzazione è stato riconosciuto
- Il clock di campionamento del bit
 - Basato sulla variazione di fronte 1-0 presente nel messaggio stesso
- Viene fatto ripartire e non più arrestato fino al ricevimento del successivo codice di sincronizzazione
 - Posto in posizione prefissata (es: ogni 256 caratteri)



CONTROL Register

CNTRL | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

*

* | | | | | | | | ___ Abilita trasmettitore (TxEN)
*

* | | | | | | | | ___ Attiva DTR (Data Terminal Ready):
pronto a comunicare
*

* | | | | | | | | ___ Attiva ricevitore (RxEN)
*

* | | | | | | | | ___ Non utilizzato (forza TxD a 0)
*

* | | | | | | | | ___ Azzera bits di errore in STATUS
*

* | | | | | | | | ___ Attiva RTS (Request to Send): indica
intenzione a trasmettere;

* | | | | | | | | ___ Resetta la periferica
*

* | | | | | | | | ___ Ricerca sincronizzazione

STATUS Register

STATUS | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- * | | | | | | | | | **b0 (TxRDY)** diviene alto quando DATA OUT viene copiato in TSHIFT REG., torna basso quando il processore copia un nuovo carattere in DATA OUT
- * | | | | | | | | | **b1 (RxRDY)** diviene alto quando RSHIFT REG. viene copiato in DATA IN, torna basso in seguito a lettura da DATA IN
- * | | | | | | | | | **Underrun**
- * | | | | | | | | | **Errore di parità**
- * | | | | | | | | | **Errore di overrun**
- * | | | | | | | | | **Errore di framing**
- * | | | | | | | | | **Rilevati bit di sincronismo**
- * | | | | | | | | | **DSR attivo** (Data Terminal Ready): pronto a comunicare

RS-232

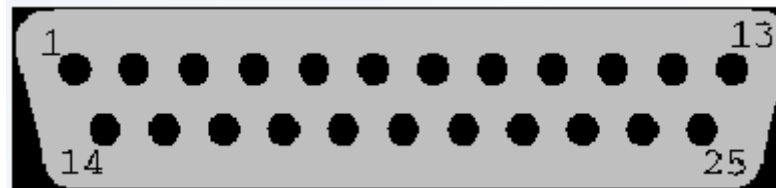
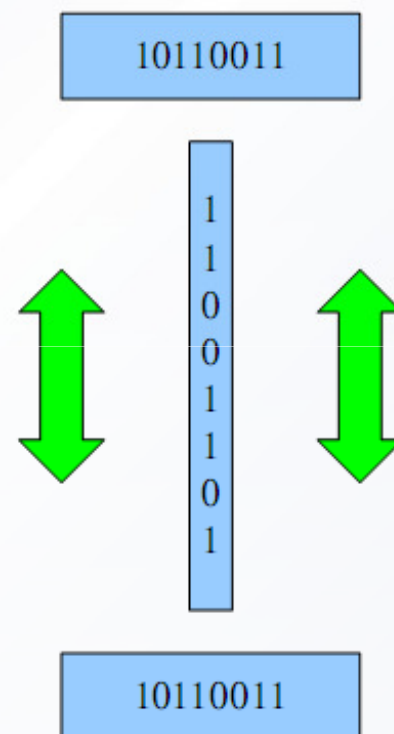
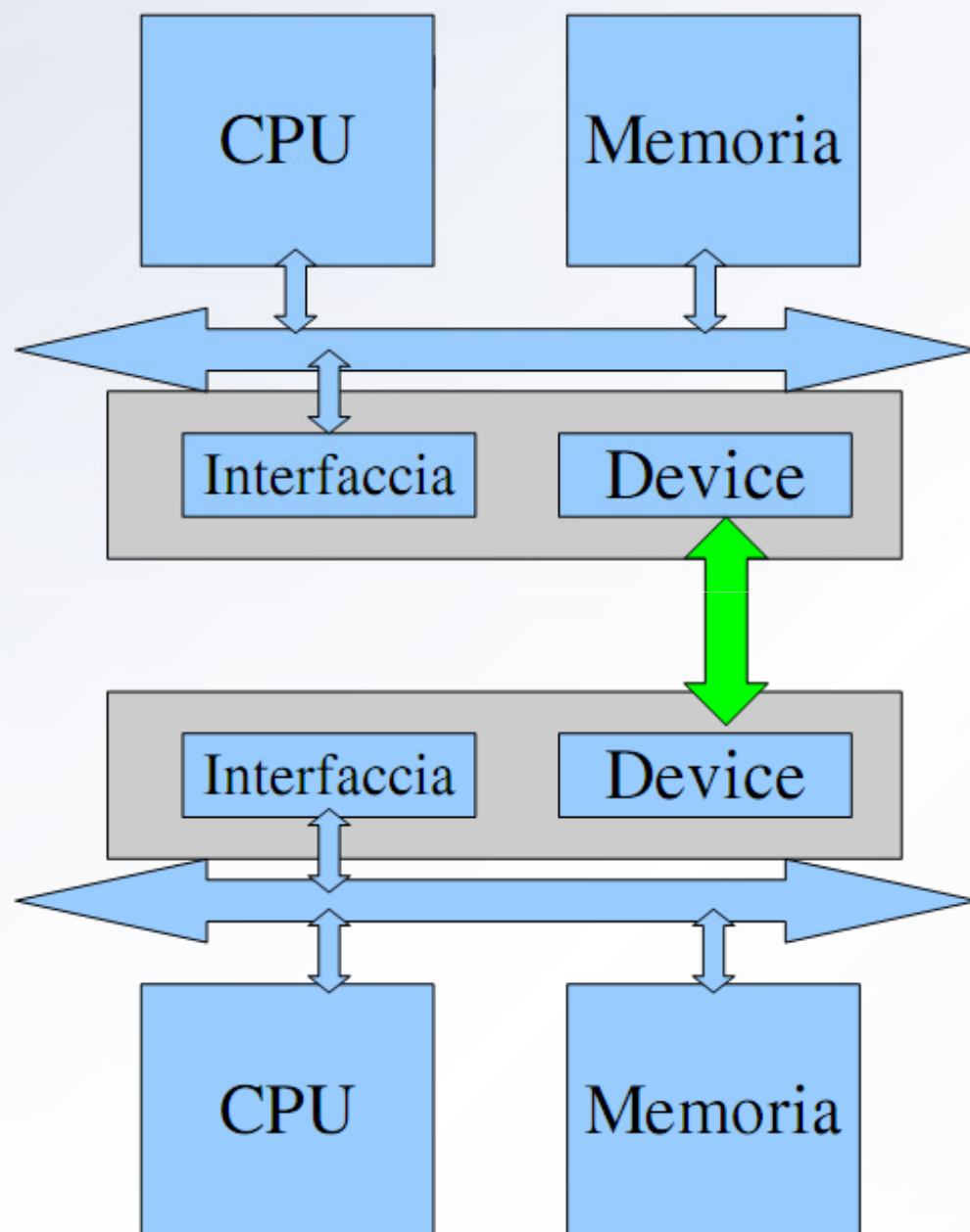


Table 12 - RS-232 Signals

PIN	DESCRIPTION	PIN	DESCRIPTION
1	Earth Ground	14	Secondary TXD
2	TXD - Transmitted Data	15	Transmit Clock
3	RXD - Received Data	16	Secondary RXD
4	RTS - Request To Send	17	Receiver Clock
5	CTS - Clear To Send	18	Unassigned
6	DSR - Data Set Ready	19	Secondary RTS
7	GND - Logic Ground	20	DTR - Data Terminal Ready
8	DCD - Data Carrier Detect	21	Signal Quality Detect
9	Reserved	22	Ring Detect
10	Reserved	23	Data Rate Select
11	Unassigned	24	Transmit Clock
12	Secondary DCD	25	Unassigned
13	Secondary CTS		



Simulazione ASIM: CONFIGURAZIONE





Simulazione ASIM: ISTRUZIONI e ACCORGIMENTI

- Inserire l'intero progetto in una cartella senza spazi
- caricare i lis per ogni processore
- Settare il valore di PC a 8200 (modify register)
- Caricare le ROM (load memory) per caricare il vettore delle interruzioni