



5. Comunicazione Seriale

Your Name

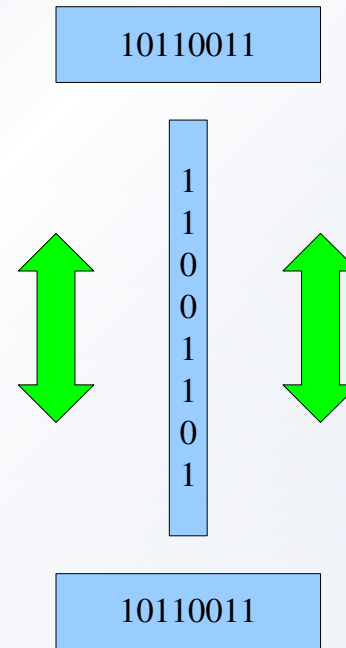
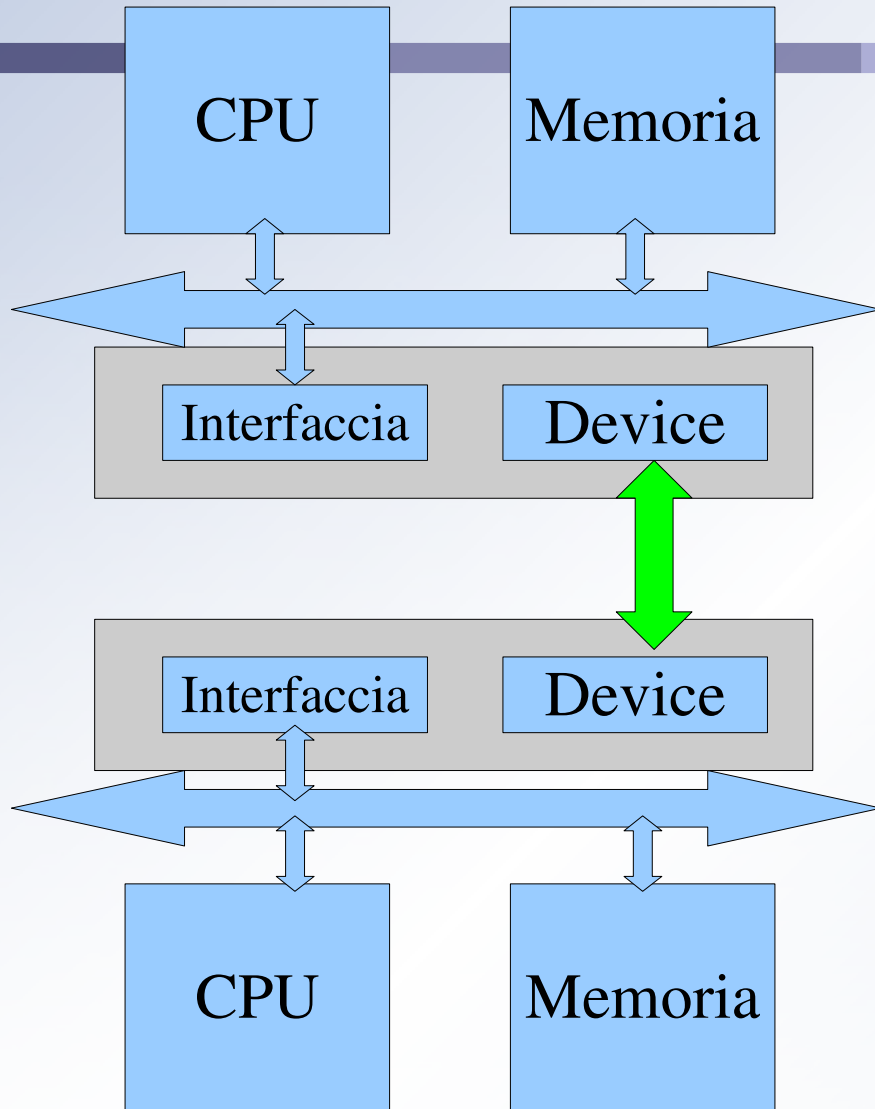
Your Title

Your Organization (Line #1)

Your Organization (Line #2)

2005-12-31

Comunicazione Seriale

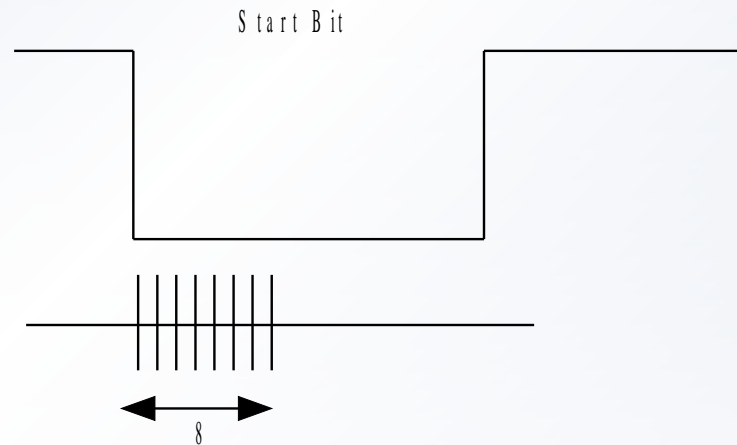


Comunicazione Seriale

- Problemi
 - Bisogna capire QUANDO campionare i bit
 - Teoricamente i due devices dovrebbero condividere lo stesso CLK
 - TX e RX sono dotati di oscillatori locali
 - Bisogna capire quando inizia e termina la comunicazione

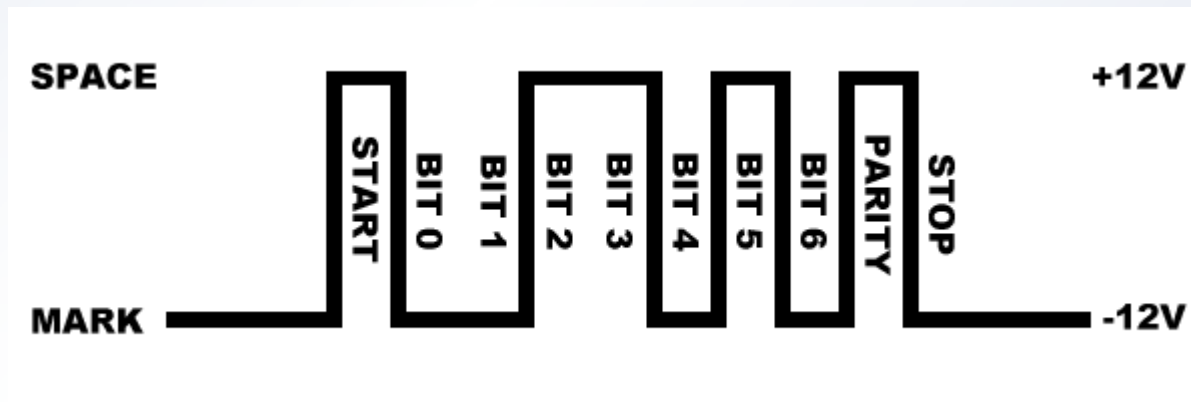
Comunicazione Asincrona

- 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

- Un bit di START comincia la trasmissione
- Seguono i bit del messaggio
- Eventualmente bit di parità
- Bit di STOP



Comunicazione Asincrona

- Full Duplex:
 - TX e RX simultanei
- Half Duplex
 - TX e RX interleaved

Flow Control

- Necessità di regolare il flusso dei dati
 - SW FC: Usa caratteri speciali per cominciare o fermare il flusso dei dati
 - XON, XOFF etc.
 - HW FC: Usa le linee RTS e CTS di RS 232.

Comunicazione Seriale

- Accordo su un clock comune
- I messaggi appaiono come stream di bit regolari
- Necessario un protocollo per iniziare e terminare lo stream
 - Serial Data Link Control
 - High-speed Data Link Control
- Non si usano extra bit
 - 25% improvements

RS-232

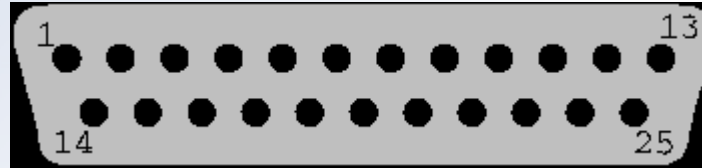


Table 12 - RS-232 Signals

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		

RS-574 (IBM PC-AT)

Figure 4 - RS-574 Connector

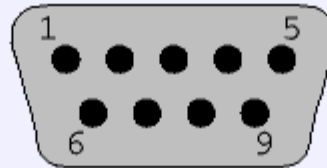


Table 14 - RS-574 (IBM PC/AT) Signals

PIN	DESCRIPTION	PIN	DESCRIPTION
1	DCD - Data Carrier Detect	6	Data Set Ready
2	RXD - Received Data	7	RTS - Request To Send
3	TXD - Transmitted Data	8	CTS - Clear To Send
4	DTR - Data Terminal Ready	9	Ring Detect
5	GND - Logic Ground		

Errori di trasmissione

- **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

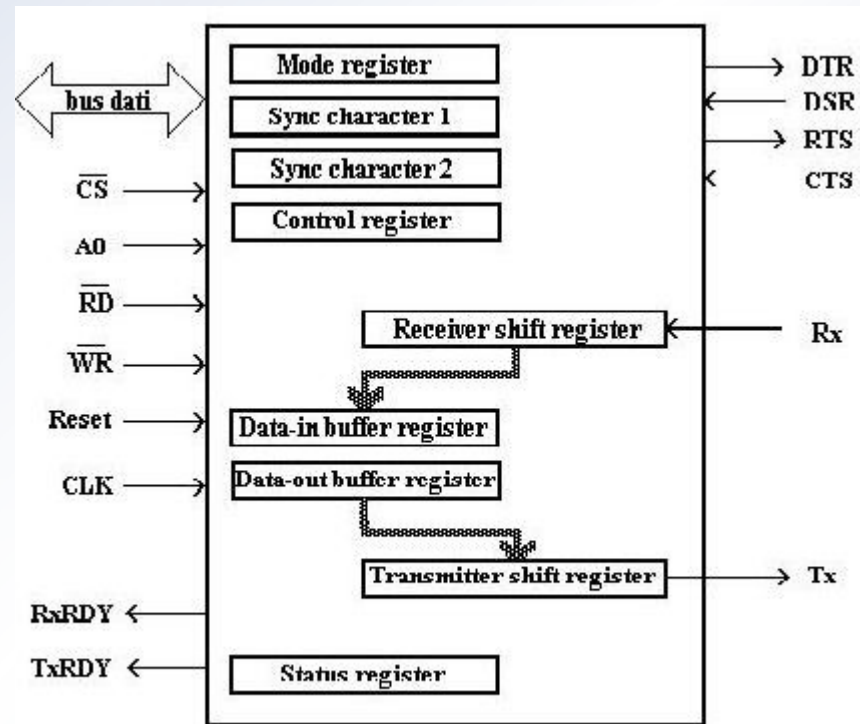
Segnali per l'Hand Shaking (HW)

- TxD ed RxD 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

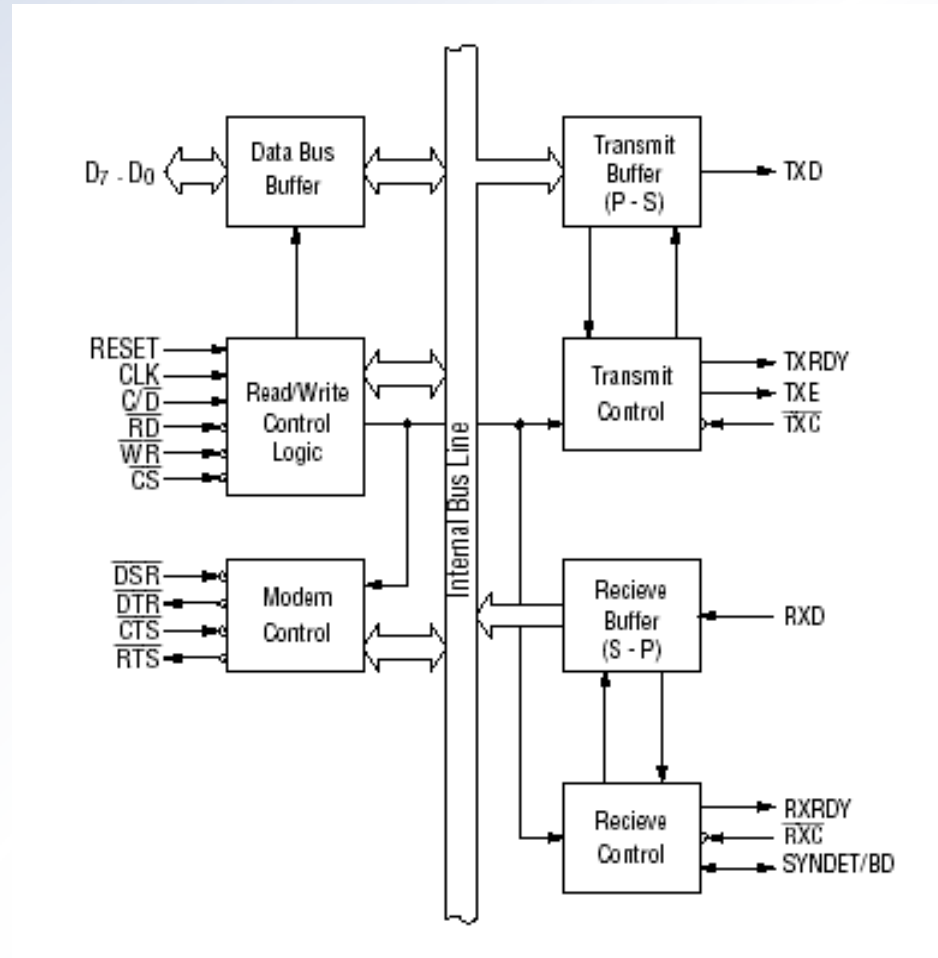
Architettura USART i8251

- 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

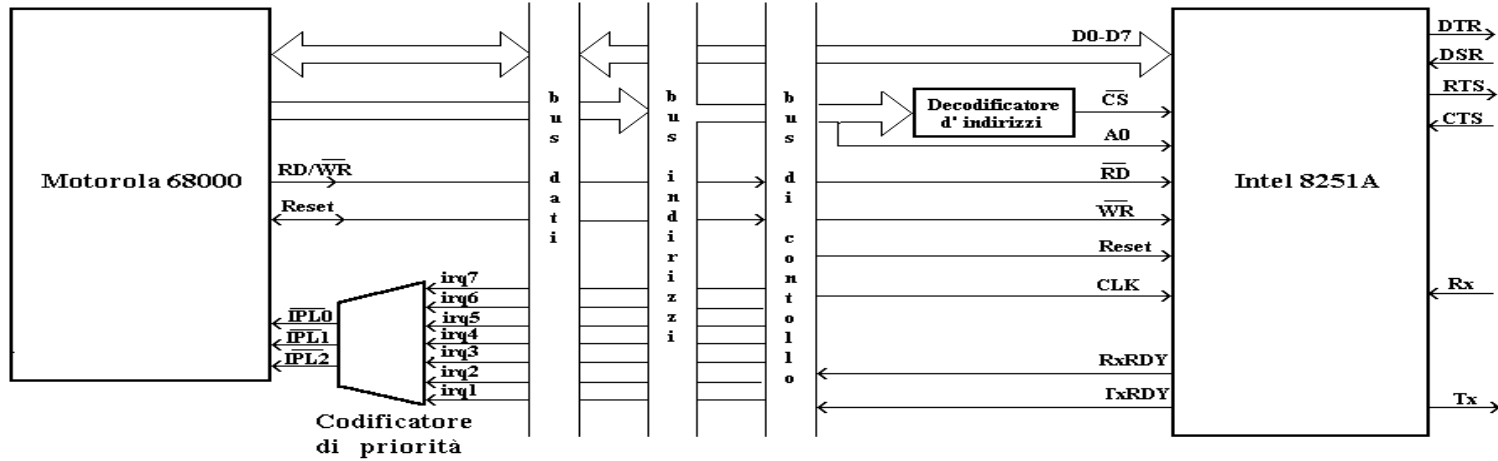
Architettura USART i8251



Architettura USART i8251



Architettura USART i8251



Modello di Programmazione

- Attivazione della TX
 - 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)
 - Nella COMMAND word: abilitare la Tx, chiedere un RESET,.....
 - Attivare TxRDY in attesa che il carattere da trasmettere sia nel buffer (INT alla CPU),
 - Il char viene trasferito nel serializzatore (se libero) e abilitando il segnale TxC si avvia la serializzazione
 - Ricomincia da capo con il prossimo carattere

Se la CPU ritarda nel caricare il buffer vengono inviati dei SYNC oppure degli "IDLE"

Modello di Programmazione

- Attivazione Ricezione
 - Dopo aver inizializzato la periferica con una MODE word, abilitare nella COMMAND word il bit RxEN
 - La periferica cerca lo start bit per sincronizzarsi e viene abilitato lo shift register a ricevere n bit
 - Esegue il test sul frame arrivato
 - 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

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
*

Control Register

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

*

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

* | | | | | | | | ___ Attiva DTR
*

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

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

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

* | | | | | | | | ___ Attiva RTS
*

* | | | | | | | | ___ 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**