

# Interfacce Seriali

slide del prof. Steven Butner Università della  
California, Santa Barbara.

Traduzione e Rielaborazione a cura del  
prof. Cleto Azzani (2006)  
IPSIA Moretto Brescia



# Interfacce Seriali

## ❖ Interfacce seriali Semplici

### – RS-232

- Utilizzate nelle comunicazioni punto-punto
  - comunicazioni seriali più semplici
  - non necessitano di decodifica di indirizzi

- Max velocità ~ 1 M bits al sec

### – I<sup>2</sup>C (Inter – IC) Bus

- Bus Seriale a 2 fili
  - filosofia di funzionamento diversa rispetto alla comunicazione punto - punto
- Bus popolare nei “sistemi embedded”

# Interfacce Seriali

## ❖ Interfacce Seriali Complesse

### – USB

- Universal Serial Bus
- Connessione fra PC ed unità periferiche
- Protocolli complessi (sia dal punto di vista hardware che software)

### – Firewire IEEE 1394

- Scambio dati ad elevata velocità fra dispositivi elettronici;
- Sviluppata dalla Apple e standardizzata poi con la denominazione IEEE 1394
- Protocolli complessi (sia dal punto di vista hardware che software)

# Interfacce Seriali

## ❖ Interfacce Seriali Complesse

- Ethernet
  - Reti Locali LAN
    - Connessioni tra computer per la condivisione di dispositivi periferici (risorse comuni condivise)
  - Sviluppata da Xerox PARC, IEEE 802.3
  - Protocolli complessi

# Interfacce Seriali

- Interfacce Seriali Wireless
  - Bluetooth
    - Comunicazioni “senza fili” fra dispositivi elettronici;
    - Protocolli emergenti basati sul modello “ethernet”
    - Hardware RF piuttosto complesso
  - WiFi (IEEE 802.11)
    - Reti LAN Wireless (senza fili)
    - Connessioni tra computer
    - Protocolli basati sul modello “ethernet” modificato
    - Hardware RF molto complesso

# Comunicazioni Sincrone e Asincrone

- La trasmissione dati fra due dispositivi seriali può avvenire in modo SINCRONO nel caso in cui sia presente oltre alla linea DATI la linea di CLOCK e ASINCRONO nel caso in cui la linea di CLOCK sia assente.
- La comunicazione sincrona consente di trasferire dati più velocemente rispetto alla comunicazione asincrona ma l'implementazione di una comunicazione sincrona è più complessa.
- Le porte seriali disponibili su un PC IBM-compatibile sono dispositivi asincroni perciò esse supportano unicamente le comunicazioni seriali asincrone.

# Comunicazioni Asincrone

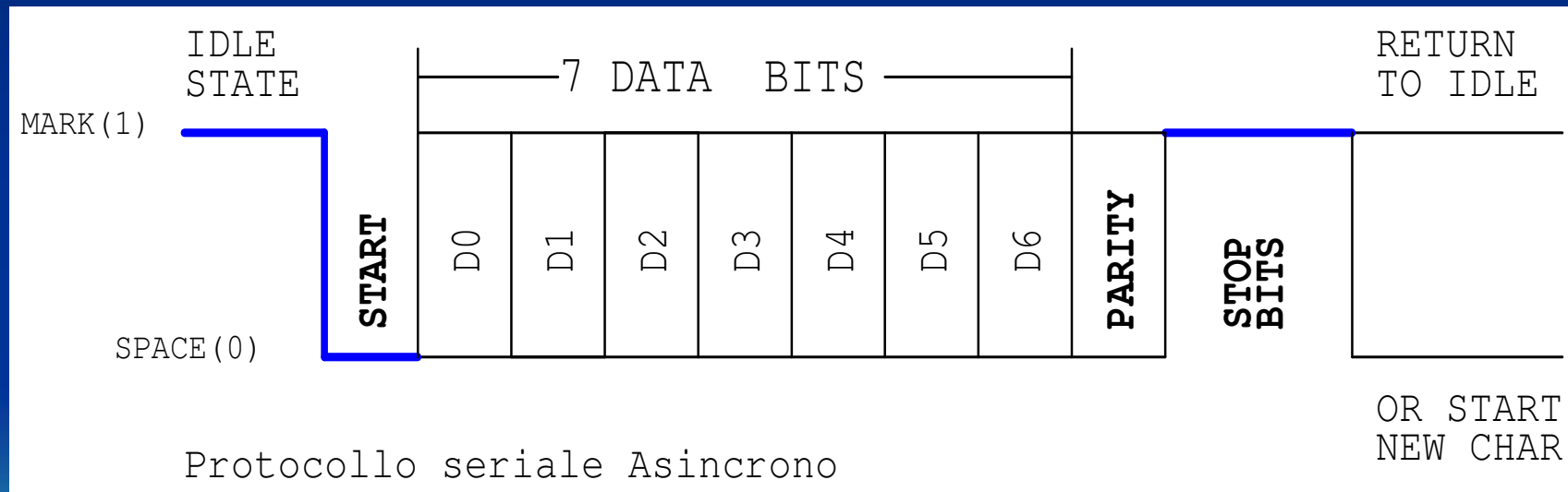
- Asincrono significa “assenza di clock” (segnale di sincronizzazione)
- L’inizio e la fine di ciascun byte trasmesso deve essere identificato da appositi bit :
  - Bit di Start
    - indica che il “dato” sta per iniziare
  - Bit di Stop
    - indica che il “dato” è terminato
  - Poiché si rende necessaria l’aggiunta di questi due bit addizionali, le comunicazioni asincrone risultano più lente di quelle sincrone.

# Comunicazioni Asincrone

- Una linea asincrona in stato “idle” (riposo) si trova a livello logico “1” (Mark).
  - I dispositivi ricevitori sono in grado di distinguere fra la situazione “idle” (nessun dato in linea) e lo stato di linea disconnessa;
  - Quando un carattere sta per essere trasmesso, viene inviato un bit di start corrispondente a un livello logico “0” (Space).
    - Perciò, quando la linea commuta da livello alto a livello basso, il ricevitore riceve una segnalazione che lo avverte che un dato sta per essere trasmesso.

# Comunicazioni Seriali

Bit di START, bit che costituiscono i DATI, bit di parità, bit di STOP



# Comunicazioni Seriali

- Bit di START,
  - Il bit di START è rappresentato da un livello “basso” (0) (SPACE); successivamente al bit di START, il trasmettitore invia i bit che compongono il dato
- Bit che costituiscono i DATI
  - Essi possono essere 5, 6, 7 oppure 8 bit
  - Sia il ricevitore che il trasmettitore devono essere predisposti per inviare e ricevere lo stesso numero di bit; strutturati allo stesso modo ed, inoltre alla stessa velocità (“baud rate”)
    - La maggior parte dei dispositivi seriali trasmette dati costituiti da 7 bit (caratteri) oppure 8 bit (byte).

# Comunicazioni Seriali

- Bit di STOP
  - Dopo che il dato è stato trasmesso viene inviato il bit di STOP
  - Il bit di STOP è rappresentato da un livello “alto” (1) (MARK);
    - La durata del bit di STOP può variare fra 1, 1.5, o 2 volte la durata di un bit di dati.

# Comunicazioni Seriali

- Bit di Parità
  - Il bit di parità consente di effettuare un “error check” di tipo elementare
    - Rivela la presenza di “dati corrotti” che possono essere presenti nella trasmissione seriale
    - Posso utilizzare parità Even, Odd, escludere il controllo di parità (no parity) o forzare il bit a (1) Mark o a (0) Space
      - Quando viene utilizzata parità E (even/pari) oppure O (odd/dispari) vengono contati i livelli (1) presenti nei dati da trasmettere; il bit di parità collocato in coda ai dati potrà valere (0) oppure (1); il suo valore verrà scelto in modo tale da rendere pari oppure dispari il numero complessivo di livelli (1) trasmessi (dato + bit di parità) a seconda del tipo di parità prescelta. Es. se trasmetto il dato 100001 con parità E il bit di parità vale 0 con parità O il bit di parità vale 1

# Comunicazioni Seriali

- Il controllo di parità è un test di tipo “molto rudimentale”
  - Esso è in grado di segnalare la presenza di un errore in un singolo bit del messaggio ricevuto ma non è in grado di indicare qual è il bit in errore
    - Il controllo di parità è un esempio di test su singolo bit
  - Se dovessero essere in errore un numero pari di bit nel messaggio ricevuto, il test di parità non ci fornisce alcuna segnalazione di errore.

# Comunicazioni Seriali

START	sempre 1	SPACE	0
DATO	5 - 6 - 7 - 8 bit	-	
PARITA'	N - E - O - M - S		
STOP	1 - 1.5 - 2	MARK	1

- Protocollo 8E1 significa
  - 8 bit dato – parità Even – 1 bit di stop
- Protocollo 7O15 significa
  - 7 bit dato – parità Odd – 1.5 bit di stop
- Il più corto 5N1 :
  - dato 5 bit – Nessuna parità – 1 bit Stop (totale 7 bit)
- Il più lungo 8E2/8O2 :
  - dato 8 – Parità – 2 bit Stop (totale 12 bit)
- *N.B. Nel conteggio dei bit va compreso il bit di Start sempre presente*

# Comunicazioni Seriali Bidirezionali

- Full-duplex
  - Contemporaneità fra trasmissione e ricezione dati (circuiti telefonici)
    - Comunicazioni bidirezionali sono possibili solo su un sistema a tre conduttori separati: uno per spedire i dati uno per ricevere i dati e uno per funzionare da massa comune fra i due circuiti oppure, in campo radio, due canali distinti.
- Half-duplex
  - Trasmissione e ricezione dati non simultanea (circuiti rice-trasmittitori walkie-talkie)
    - Si utilizzano solo due conduttori: la linea dati e la linea di massa oppure, in campo radio, un unico canale.

# RS-232C

- RS-232C : Standard Raccomandato
  - C è l'ultima revisione dello standard.
  - Le porte seriali presenti nella maggior parte dei PC utilizzano un “subset” dello standard RS-232C
    - Lo standard RS-232C prescrive l'utilizzo di un connettore 25-pin maschio (a vaschetta) con 22 pin utilizzati
  - La maggior parte di questi pin non sono necessari per le normali comunicazioni del PC
    - La maggior parte dei PC di ultima generazione sono equipaggiati con connettori a vaschetta maschio dotati solo di 9 pin.

# Dispositivi DTE e DCE

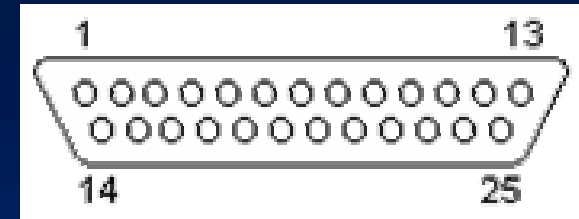
- DTE : Data Terminal Equipment
- DCE : Data Communications Equipment
  - Storicamente la distinzione fra DTE e DCE si riferisce alla connessione che si stabilisce fra un computer (DTE) e un modem (DCE)
  - In ambito RS-232C, questa termini vengono utilizzati per indicare il “pin-out” dei connettori presenti su un dispositivo e la direzione dei segnali sui pin del connettore.
  - Normalmente con il termine DTE si indica un computer e con DCE si indica qualsiasi altro dispositivo.

# Connettori RS-232C

- I dispositivi DTE utilizzano un connettore 25-pin maschio a vaschetta DB25M
- I dispositivi DCE utilizzano un connettore 25-pin femmina a vaschetta DB25F
  - La comunicazione fra un dispositivo DTE e un DCE avviene utilizzando un cavo diretto pin to pin.
  - Per connettere due dispositivi dello stesso tipo (due DTE oppure due DCE) deve essere utilizzato un cavo denominato “null modem”
    - I cavi “null modem” incrociano fra di loro le linee di trasmissione con quelle di ricezione come verrà meglio specificato nelle slide seguenti.

# Connettore RS-232C

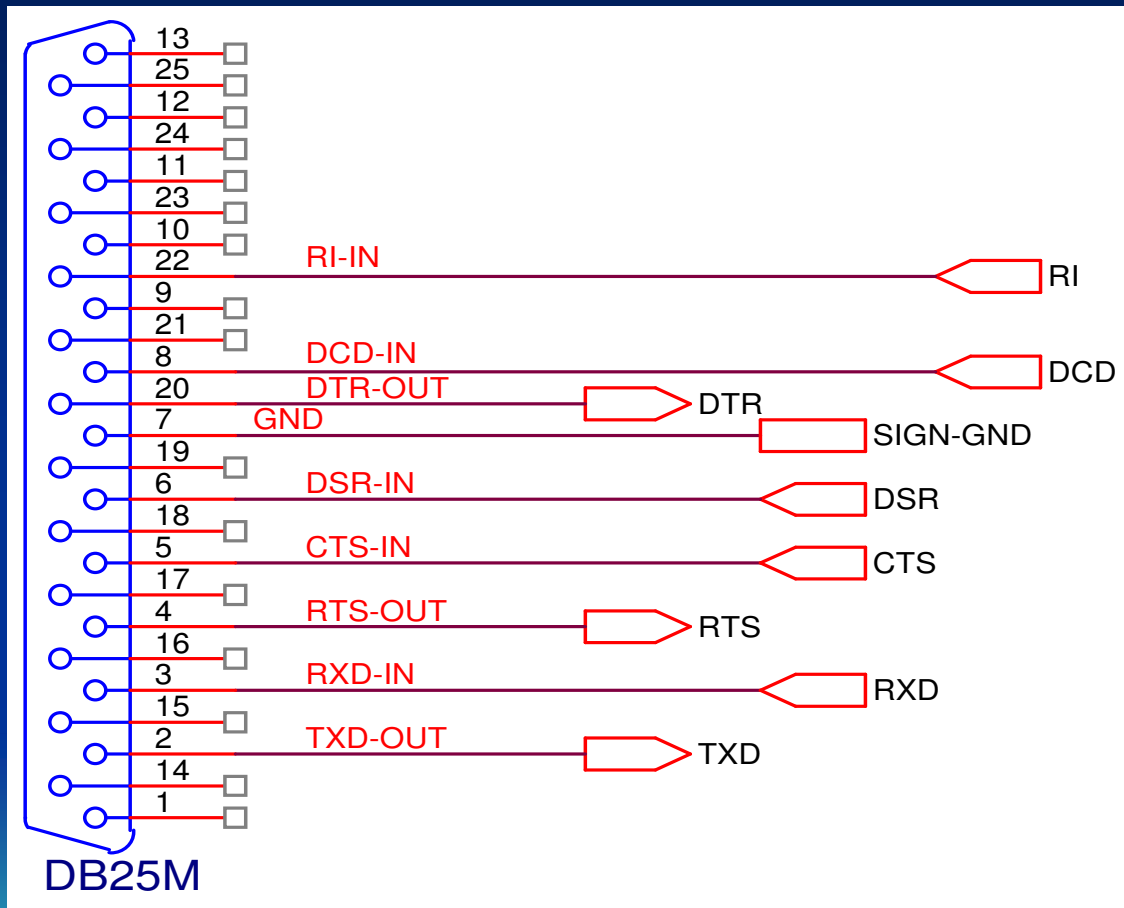
MASCHIO DB25M DTE (PC)



PIN	DIRECTION OF SIGNAL	DTE
1	Protective Ground	
2	TD - Transmitted Data	O
3	RD - Received Data	I
4	RTS - Request To Send	O
5	CTS - Clear To Send	I
6	DSR - Data Set Ready Incoming Handshaking	I
7	GND - Signal ground	
8	CD - Carrier Detect (from modem)	I
20	DTR - Data Terminal Ready Outgoing Handshaking	O
22	RI - Ring Indicator (from modem)	I

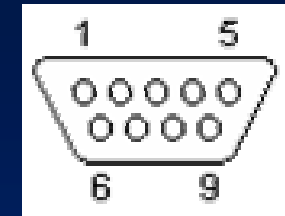
# Connettore RS-232C

MASCHIO DB25M DTE (PC)



# Connettore RS-232C

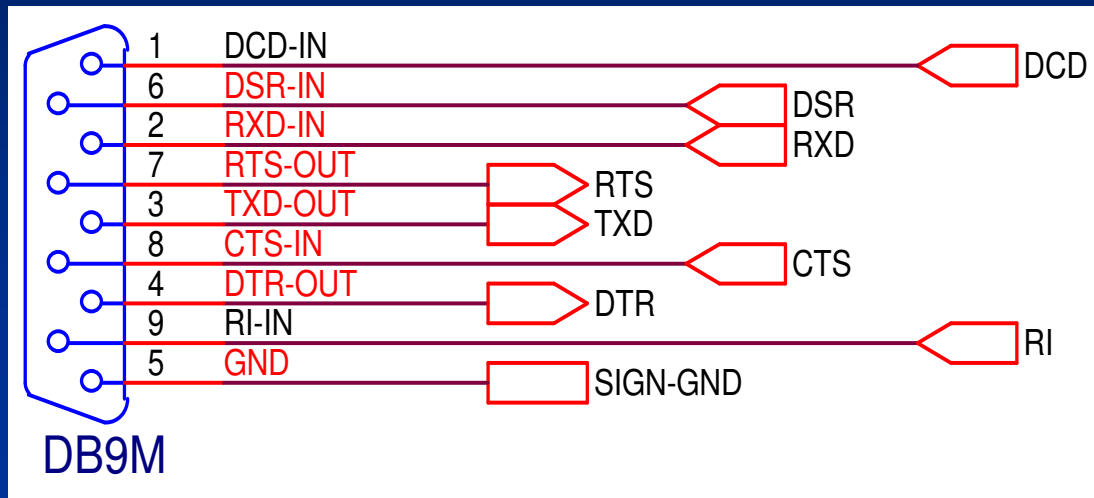
MASCHIO DB9M DTE (PC)



PIN	DIRECTION OF SIGNAL	DTE
1	CD - Carrier Detect (from modem)	I
2	RD - Received Data	I
3	TD - Transmitted Data	O
4	DTR - Data Terminal Ready Outgoing Handshaking	O
5	GND - Signal ground	
6	DSR - Data Set Ready Incoming Handshaking	I
7	RTS - Request To Send	O
8	CTS - Clear To Send	I
9	RI - Ring Indicator (from modem)	I

# Connettore RS-232C

MASCHIO DB9M DTE (PC)



# Controllo di Flusso Hardware

Trasmissione Dati dal DCE verso il DTE (PC)

- RTS sta per “Request To Send”
  - Questa linea viene utilizzata assieme alla linea CTS quando risulta abilitato il “controllo di flusso hardware” su entrambi i dispositivi DTE e DCE
    - Il dispositivo DTE pone la linea RTS in condizione “Mark” (1) per avvertire il dispositivo remoto DCE che il DTE è pronto a ricevere dati dal DCE.

# Controllo di Flusso Hardware

Trasmissione Dati dal DCE verso il DTE (PC)

- RTS
  - Se il DTE non è in grado di ricevere dati (normalmente perché il buffer di ricezione è pieno) esso pone la linea RTS in condizione “Space” per richiedere al DCE di arrestare l’invio di dati
    - Quando il DTE è pronto a ricevere ulteriori dati (dopo che il buffer di ricezione è stato svuotato), la linea RTS verrà ricollocata in stato “Mark”.

# Controllo di Flusso Hardware

Trasmissione Dati dal DTE (PC) verso il DCE

- **RTS** sta per “Request To Send” (Richiesta di spedizione Dati)
  - Questa linea viene utilizzata assieme alla linea CTS quando risulta abilitato il “controllo di flusso hardware” su entrambi i dispositivi DTE e DCE
    - Il dispositivo DTE pone la linea RTS in condizione “Mark” (1) per avvertire il dispositivo remoto DCE che il DTE è pronto a spedire dati verso il DCE.

# Controllo di Flusso Hardware

Trasmissione Dati dal DTE (PC) verso il DCE

- RTS
  - Se il DTE non è in grado di ricevere dati (normalmente perché il buffer di ricezione è pieno) esso pone la linea RTS in condizione “Space” per richiedere al DCE di arrestare l’invio di dati
    - Quando il DTE è pronto a ricevere ulteriori dati (dopo che il buffer di ricezione è stato svuotato), la linea RTS verrà ricollocata in stato “Mark”.

# Controllo di Flusso Hardware

- ❖ La funzione complementare della linea RTS è svolta dalla linea CTS che sta per “Clear To Send”
  - Il DCE pone la linea RTS in stato “Mark” per segnalare al DTE che esso è pronto a ricevere dati
  - Se il DCE non è in grado di ricevere dati la linea RTS viene posta in stato “Space”.
  - Le due linee RTS/CTS consentono quindi di realizzare il controllo di flusso “hardware”.

# Controllo di Flusso Software

- Xon/Xoff o controllo di flusso "software"
  - Il controllo di flusso software utilizza due appositi caratteri di controllo ASCII che vengono trasmessi da un dispositivo ad un altro per bloccare l'invio dei dati (Xoff) o per riavviare l'invio dei dati (Xon).
    - Nel controllo di flusso software le linee RTS e CTS non vengono usate.
    - Xoff = \$13 DC3      Xon = \$11 DC1

# Codice ASCII

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M	]	m	
E	SO	RS	.	>	N	^	n	
F	SI	US	/	?	O	_	o	DEL

# Segnali di “Handshake”

- DTR sta per Data Terminal Ready
  - DTR è una uscita del dispositivo DTE
  - La funzione del segnale DTR è molto simile a quella del segnale RTS
- DSR sta per Data Set Ready
  - DSR è un ingresso del dispositivo DTE
  - La funzione del segnale DSR è molto simile a quella del segnale CTS

# Segnali di “Handshake”

- ❖ Alcuni dispositivi seriali utilizzano DTR e DSR semplicemente per confermare che un dispositivo è connesso ed è acceso
  - Le linee DTR e DSR sono state in origine previste per fornire un metodo alternativo di “Hardware Handshaking”.
    - E' inutile utilizzare contemporaneamente sia le linee RTS/CTS che le linee DTR/DSR per un controllo di flusso di tipo “hardware”. Per questa ragione DTR e DSR vengono raramente utilizzate in questo campo.

# Rivelazione di Portante

- CD sta per “Carrier Detect”
  - Carrier Detect viene usato dal modem per segnalare al DTE che è attiva una connessione con un altro modem oppure che è presente sulla linea telefonica il segnale di centrale

# Indicatore di Chiamata

- L'ultima linea rimasta è la linea RI o "Ring Indicator"
  - Un modem commuta lo stato della linea RI quando una chiamata giunge sulla linea telefonica connessa al modem
    - Le linee CD e RI sono disponibili solo quando un DTE è connesso ad un modem.
    - Poiché la maggior parte dei modem trasmettono ad un PC informazioni di stato quando è presente il segnale di centrale oppure quando viene stabilita una connessione con un modem remoto, oppure quando giunge una chiamata sulla linea telefonica, queste due linee sono raramente utilizzate.

# Baud e Bit al Secondo (BPS)

- Baud è una unità di misura che ha tratto il suo nome da Jean Maurice Emile Baudot, impiegato presso il Servizio Telegrafico Francese
  - Egli si occupò di sviluppare il primo codice a lunghezza costante a 5 bit per la codifica delle lettere dell'alfabeto nei primi anni del 1900.
  - Baud misura la “velocità di modulazione” oppure il numero di simboli al secondo che vengono emessi dal modem.

# Baud e Bit al Secondo (BPS)

- Il BAUD non coincide sempre con i BPS (bit al secondo)
  - Se si connettono assieme due dispositivi seriali utilizzando un cavo diretto BPS e BAUD coincidono.
  - Se i due dispositivi viaggiano a 19200 BPS, la linea cambia stato 19200 volte al secondo; ma quando si utilizzano i modem ciò non è più vero.

# Baud e Bit al Secondo (BPS)

- Poiché i modem trasferiscono segnali su una linea telefonica, il “baud rate” è attualmente limitato ad un massimo di 2400 baud.
  - Questa è una restrizione fisica delle linee messe a disposizione dalle Compagnie Telefoniche.
  - Le accresciute performance di velocità raggiunte dai modem a 9600 o dai modem di ultima generazione a 57600 baud è stata raggiunta utilizzando sofisticate tecniche di modulazione di fase e di compressione dei dati.

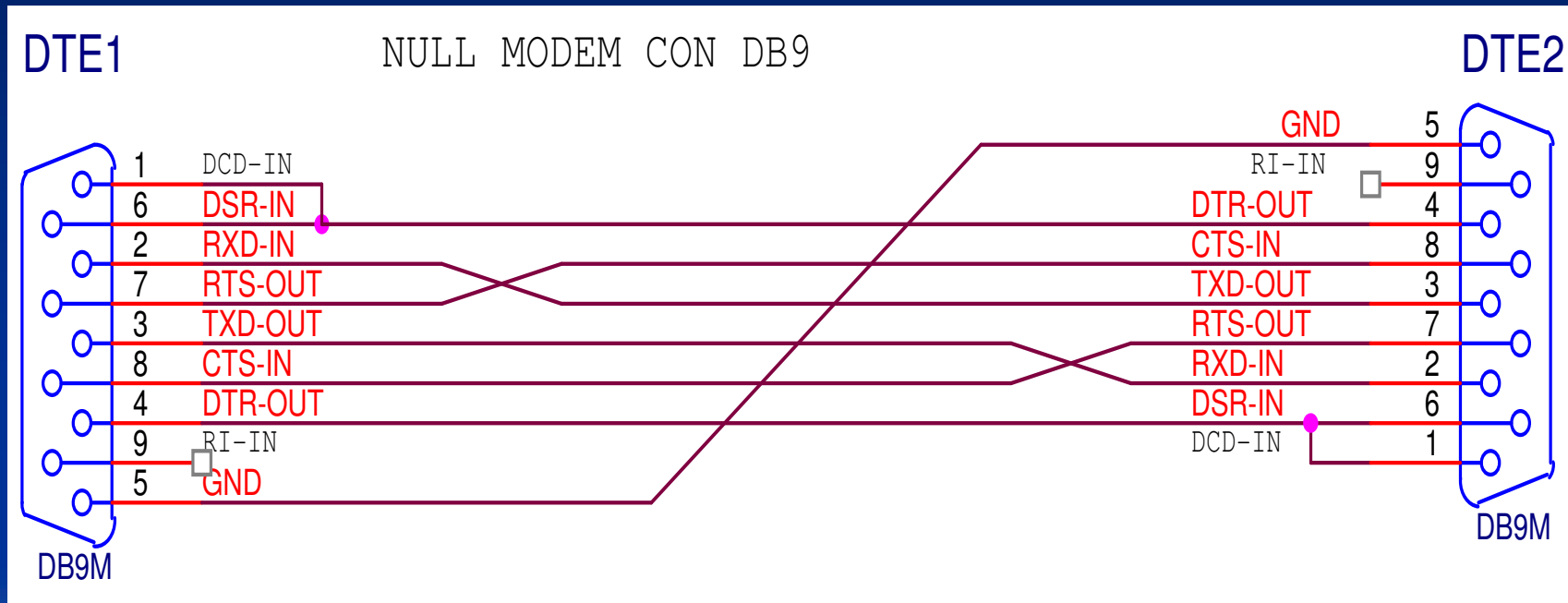
# Lunghezza dei Cavi

- Lo standard RS-232C impone una lunghezza massima del cavo pari a 15 metri (50 feet)
  - E' possibile ignorare questo limite imposto dallo standard; un cavo può anche essere lungo fino a 3 Km. (10,000 feet) con baud rate massimo di 19200 se viene utilizzato un cavo schermato di ottime qualità.
  - Le condizioni ambientali esterne hanno un effetto pesante sulla lunghezza di cavi non schermati.
    - In ambienti elettricamente disturbati, anche cavi piuttosto corti sono in grado di raccogliere disturbi di una certa entità.

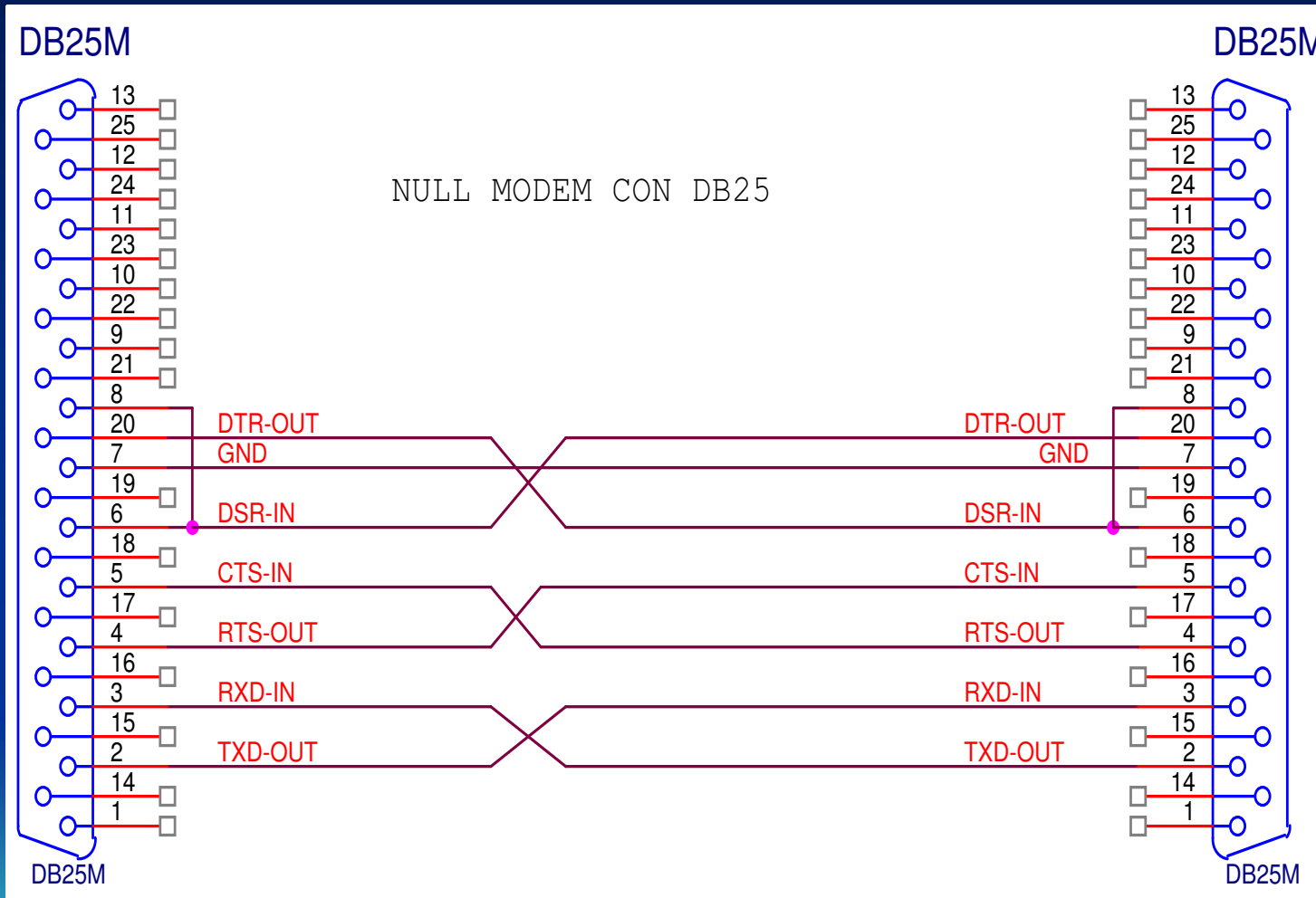
# Cavi e Adattatori Null Modem

- Due dispositivi dello stesso tipo (DTE o DCE) connessi con un cavo diretto RS232C
  - Non funzionano
    - La linea di Trasmissione sul dispositivo1 viene connessa alla medesima linea sul dispositivo2;
    - La linea di Ricezione sul dispositivo1 viene connessa alla medesima linea sul dispositivo2;
  - A meno che non venga utilizzato un cavo “Null Modem” o un adattatore Null Modem;
    - Il cavo null-modem incrocia le linee di trasmissione con quelle di ricezione in modo tale che la Trasmissione su un lato viene connessa alla Ricezione sull’altro lato e vice-versa.
    - Anche le linee di controllo DTR & DSR, come pure RTS & CTS vengono fra di loro incrociate in un cavo “Null Modem”.

# Null Modem Cable DB9M



# Null Modem Cable DB25M



# Comunicazioni Sincrone e Asincrone

- Ci sono due tipi di comunicazioni seriali: le comunicazioni sincrone ed asincrone.
  - In caso di comunicazioni sincrone, i due dispositivi inizialmente si sincronizzano tra loro, e successivamente continuano ad inviare caratteri per mantenersi in sincronismo.
  - Anche quando il dato non viene realmente spedito, un costante flusso di bit fra trasmettitore e ricevitore permette a ciascun dispositivo di conoscere lo stato dell'altro dispositivo in qualunque istante.
    - Perciò, il carattere trasmesso può essere o un dato attuale oppure un carattere di "idle".

# Comunicazioni Sincrone e Asincrone

- Le comunicazioni sincrone consentono di trasferire dati più velocemente rispetto alle comunicazioni asincrone; nelle comunicazioni sincrone infatti sono assenti sia il bit di Start che quello di Stop.
- Le porte seriali disponibili su un PC IBM-compatibile sono dispositivi asincroni perciò esse supportano unicamente le comunicazioni seriali asincrone.

# Comunicazioni Sincrone e Asincrone

- Asincrono significa “nessuna sincronizzazione” e perciò non si richiede l’invio o la ricezione di caratteri “idle”.
- Tuttavia, l’inizio e la fine di ciascun byte trasmesso deve essere identificato da un bit di Start e da un bit di Stop.
  - Il bit di start indica che il “dato” sta per iniziare e il bit di stop indica quando che il dato è terminato.
    - Poiché si rende necessaria l’aggiunta di questi due bit addizionali, le comunicazioni asincrone risultano leggermente più lente delle sincrone tuttavia in caso di comunicazioni asincrone, il processore non deve occuparsi di inviare caratteri addizionali “idle”.

# Comunicazioni Sincrone e Asincrone

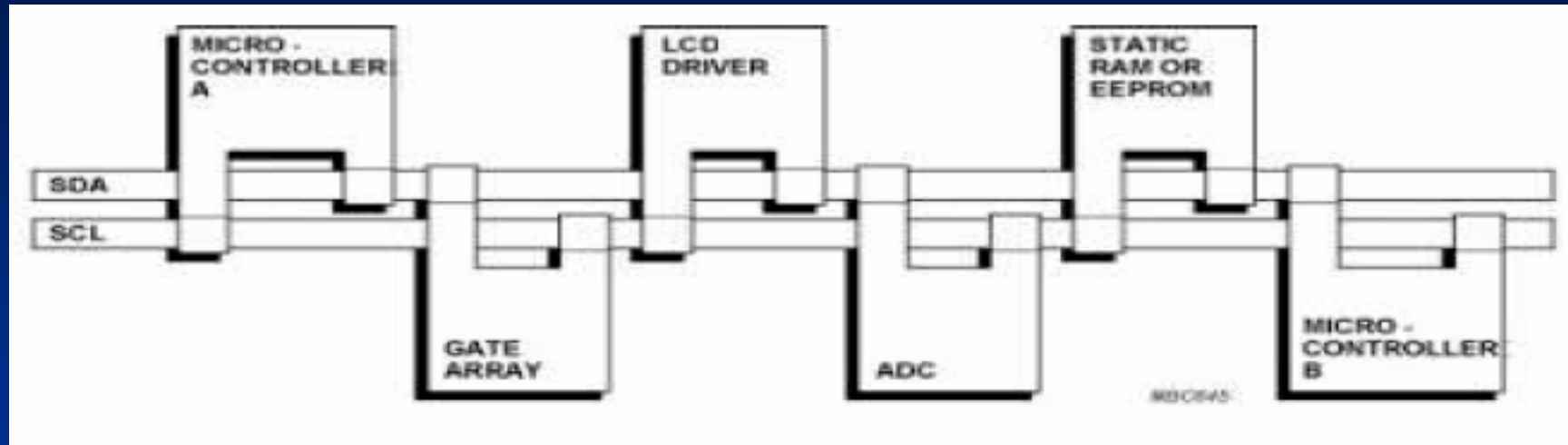
- Una linea asincrona in stato “idle” si trova a livello logico “1” (stato Mark).
  - I dispositivi sono in grado di distinguere fra la situazione “idle” (nessun dato inviato in linea) e lo stato di linea disconnessa;
  - Quando un carattere sta per essere trasmesso, viene inviato un bit di start corrispondente a un livello logico “0” (stato space).
    - Perciò, quando la linea commuta da livello alto a livello basso, il ricevitore riceve una segnalazione che lo avverte che un dato sta per essere trasmesso.

# Il Bus I<sup>2</sup>C

## – I<sup>2</sup>C (Inter – IC) Bus

- Bus Seriale, 2-fili
  - Contrapposti alla comunicazione punto-punto
- Sviluppato dalla “Philips Semiconductor” agli inizi del 1980
- Massime velocità
  - 100 Kbps in modalità standard
  - 400 Kbps in modalità “fast”
  - 3.4 Mbps in modalità “high-speed”
- Indirizzamento controllato via software
  - Nessun hardware di decodifica indirizzi

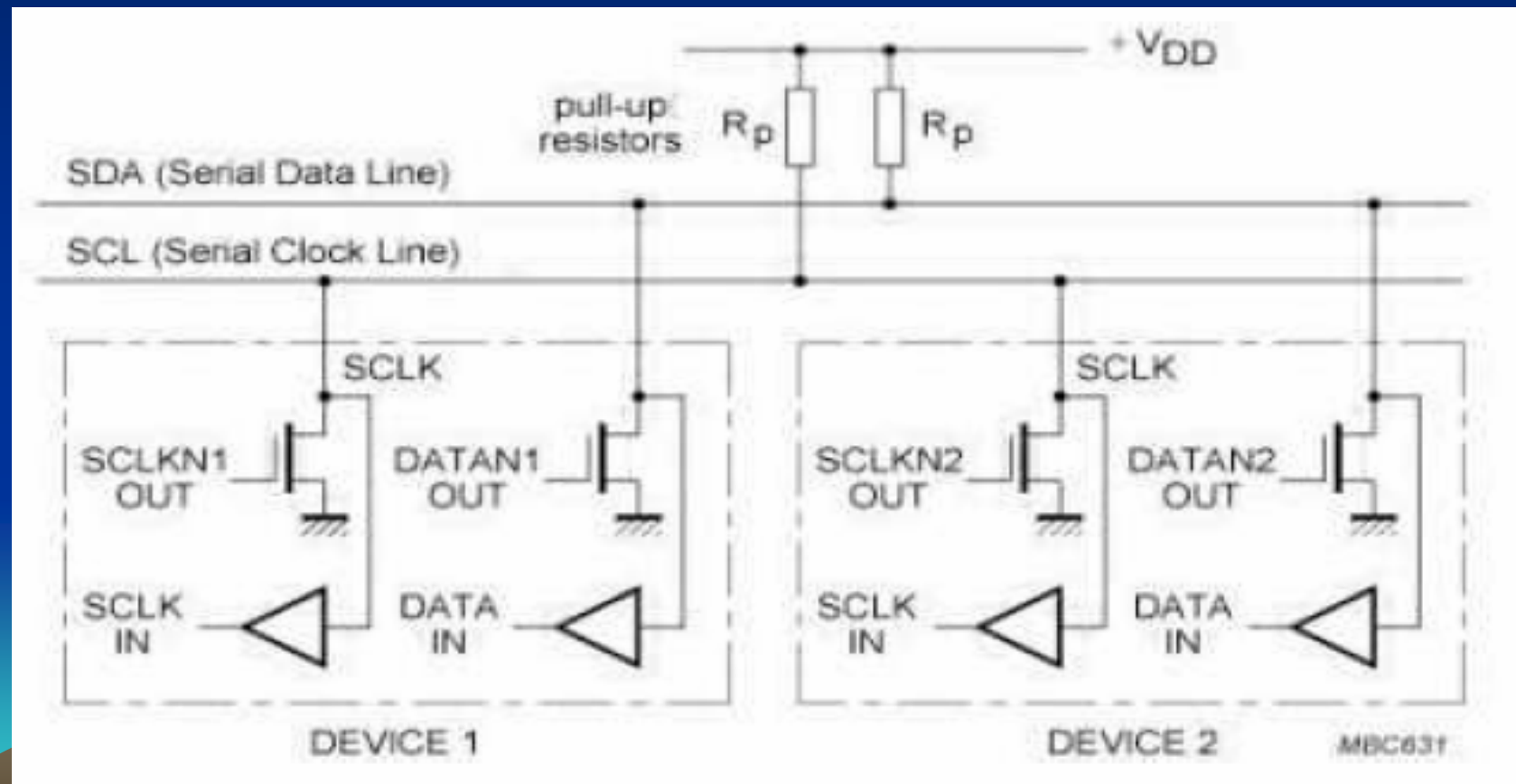
# Il Bus I<sup>2</sup>C



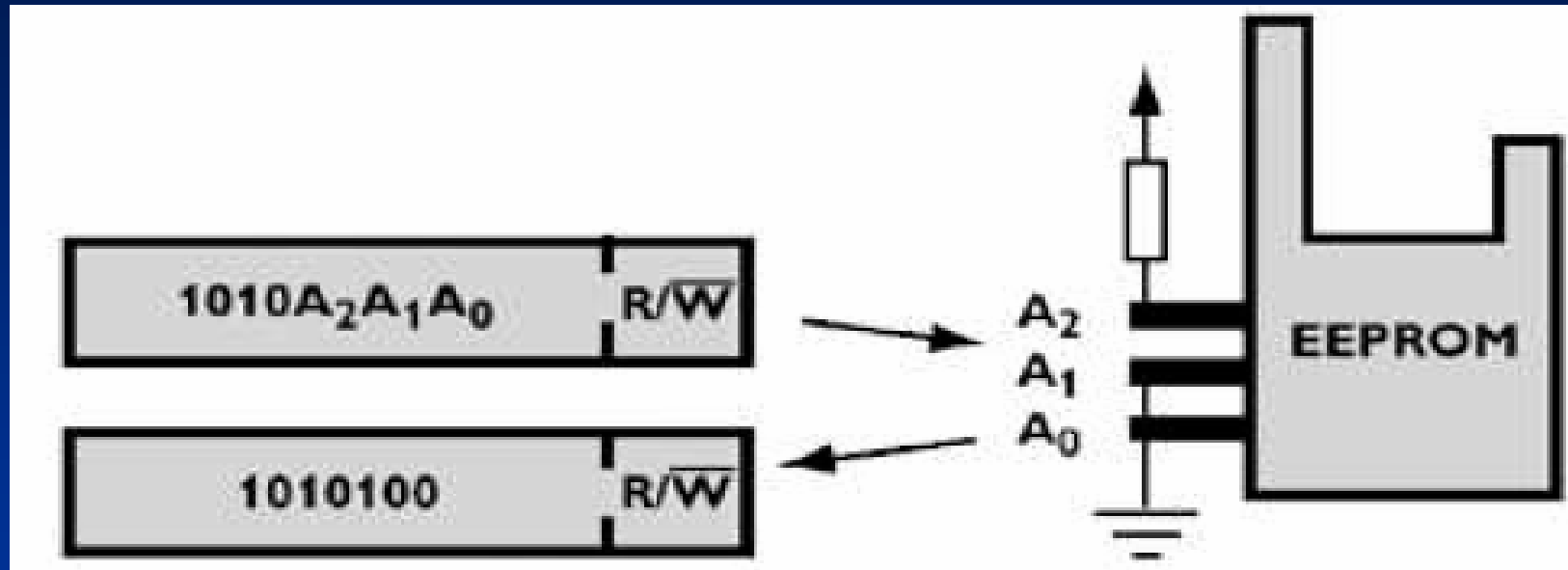
- Bus Seriale a 2 fili
  - SDA – Serial Data
  - SCL – Serial Clock
- Possibilità di avere più dispositivi “master” e più “slave”
  - Necessita almeno un master (MCU o DSP)

# Il Bus I<sup>2</sup>C

- Implementazione Elettrica
  - Connessione Open Drain sia sulla linea dati che sulla linea del clock



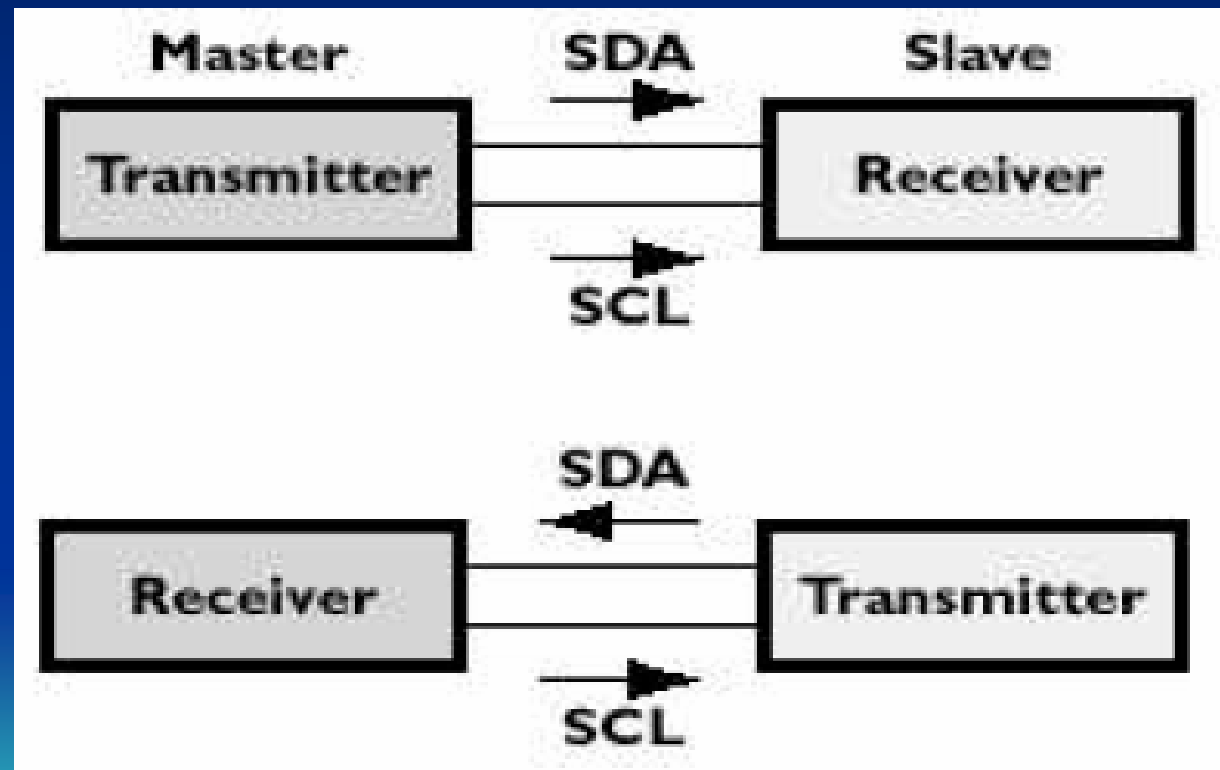
# Il Bus I<sup>2</sup>C



- Ogni dispositivo ha indirizzo univoco I<sup>2</sup>C a 7 bit
  - 4 bit più significativi risultano fissi ed assegnati a specifiche categorie di dispositivi:
    - 1010 assegnata alle EEPROM seriali
    - 3 bit meno significativi sono programmabili attraverso i pin di programmazione dell'hardware

# Il Bus I<sup>2</sup>C

## Terminologia I<sup>2</sup>C Bus



# Il Bus I<sup>2</sup>C

## ▪ Terminologia I<sup>2</sup>C Bus

- Master

Componente che inizializza un trasferimento (Comando Start), genera il segnale di clock (SCL) e conclude il trasferimento (Comando Stop)

- Slave

Dispositivo indirizzato dal “master”. Un dispositivo “slave” può essere sia un ricevitore che un trasmettitore.

# Il Bus I<sup>2</sup>C

## ▪ Terminologia I<sup>2</sup>C Bus

- Trasmittitore
  - Dispositivo che invia dati sul bus
    - Può essere un trasmettitore-master oppure un trasmettitore-slave
- Ricevitore
  - Dispositivo che riceve dati dal bus
    - Può essere un ricevitore-master oppure un ricevitore-slave

# Il Bus I<sup>2</sup>C

## – Terminologia I<sup>2</sup>C Bus

- Multi-master

- Possibilità per più master di coesistere sul medesimo bus nello stesso tempo senza collisioni o perdita di dati

- Arbitraggio

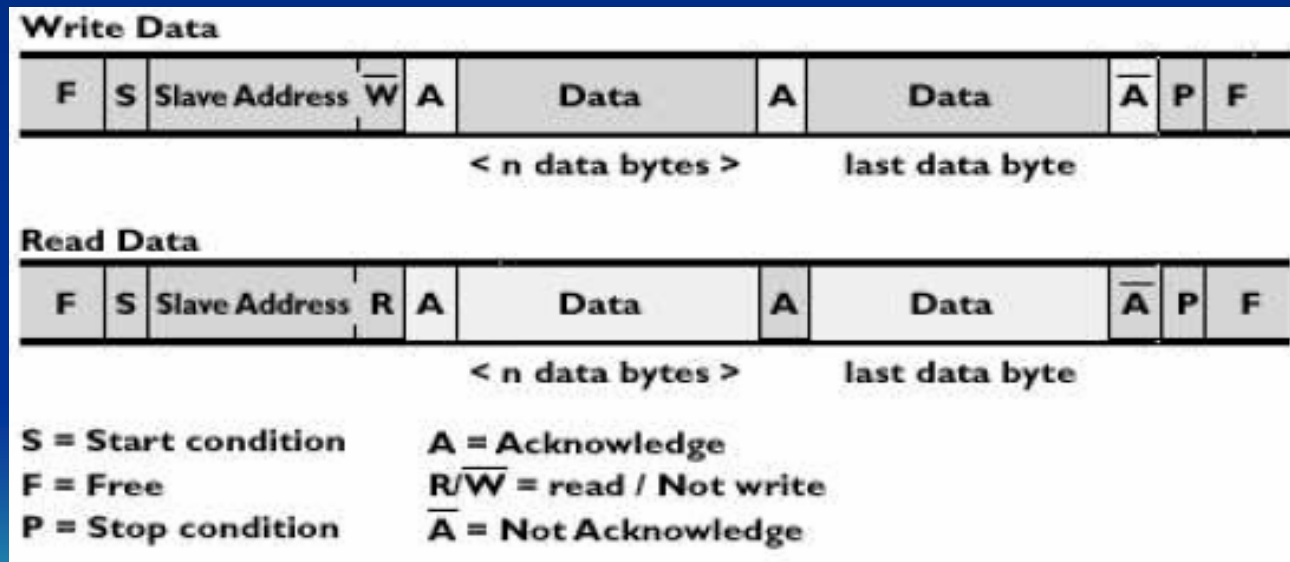
- Procedura predisposta che autorizza solo un master alla volta ad assumere il controllo del bus.

- Sincronizzazione

- Procedura predisposta che sincronizza i segnali di clock forniti da due o più dispositivi master.

# Il Bus I<sup>2</sup>C

- Indirizzo I<sup>2</sup>C del dispositivo “target” viene inviato nel primo byte
  - Il bit LSB del byte iniziale, indica la direzione di trasferimento dei dati (R/W)
- Ciascuna sequenza di trasmissione inizia con la condizione Start e termina con la condizione Stop o ReStart.



# Il Bus I<sup>2</sup>C

- Terminologia I<sup>2</sup>C Bus Transfer
  - F (FREE)
    - Il bus è libero o si trova in condizione “idle”; La linea dati SDA e la linea del clock SCL si trovano entrambe a livello alto.
  - S (START) or R (RESTART)
    - Il trasferimento dati inizia con una condizione start. Il livello della linea dati SDA cambia da livello alto a livello basso, mentre la linea di clock SCL rimane a livello alto. Quando si verifica questo evento, il bus diviene occupato (busy).
  - C (CHANGE)
    - Mentre la linea di clock SCL si trova a livello basso, il bit che deve essere trasferito può essere applicato alla linea dati SDA da un trasmettitore. Durante questo tempo, SDA può cambiare il suo stato fino a che la linea SCL rimane a livello basso.

# Il Bus I<sup>2</sup>C

- Terminologia I<sup>2</sup>C Bus Transfer
  - D (DATA)
    - Un livello alto o basso sulla linea dati SDA è valido durante il livello alto della segnale di clock SCL. Questo livello deve essere mantenuto stabile durante l'intero tempo in cui il segnale di clock si mantiene alto per evitare errate interpretazioni che lo possono fare confondere con condizioni di Start o di Stop.
  - P (STOP)
    - Il trasferimento dati si conclude in condizione Stop. Questo capita quando il livello sulla linea dati SDA passa da livello basso a livello alto mentre la linea di clock SCL rimane a livello alto. Quando il trasferimento dati è terminato, il bus risulta nuovamente libero.

# USB

- Universal Serial Bus



# USB

- L'interfaccia USB è stata sviluppata per fornire un semplice sistema di connessione fra un PC e un'ampia gamma di dispositivi periferici diversi
  - Fino ad un massimo di 127 dispositivi USB possono essere connessi ad un computer
- USB è contemporaneamente un protocollo seriale e un "physical link"
  - Una coppia di fili trasmette dati utilizzando uno standard differenziale
  - Una seconda coppia di fili fornisce energia alle periferiche ("downstream peripherals")

# USB

- Velocità di flusso dei dati
  - USB 1.1 : 1.5 Mbps - 12 Mbps
  - USB 2.0 : 480 Mbps
    - Molto alta se confrontata con la I<sup>2</sup>C, 100 Kbps – 3.4 Mbps
- Le periferiche possono essere connesse e rimosse senza spegnere l'intero sistema
  - Si parla di “hot swapping”
  - L'energia (per l'interfaccia) è fornita dalla connessione USB

# USB

- Le unità periferiche USB sono di tipo “slave”; esse rispondono a comandi provenienti dal dispositivo “host”.
  - Quando una periferica è connessa ad una rete USB, l’unità host comunica con il dispositivo USB per conoscere la sua identità e per sapere quale driver è necessario al suo funzionamento
    - Il Processo viene chiamato “enumeration”
  - Dotate di caratteristiche “Plug and play”

# USB

- Le periferiche USB comunicano solamente con il dispositivo “host”
  - Le periferiche non possono comunicare direttamente fra di loro
- Le comunicazioni USB hanno luogo tra il dispositivo “host” e gli “endpoints” localizzati all’interno delle periferiche

# USB

- Tutti i trasferimenti USB fanno uso di “virtual pipes”
  - Quando si attiva una comunicazione con la periferica, ciascun “endpoint” restituisce un descrittore
    - Un descrittore è una struttura-dati che comunica al dispositivo host la configurazione e le aspettative del “endpoint”
    - Il descrittore comprende il tipo di trasferimento, la massima dimensione dei pacchetti e in alcuni casi l’intervallo per il trasferimento dati e la larghezza di banda richiesta

# USB

- “Device driver” dal lato host
  - USB è uno standard piuttosto complesso che richiede un consistente supporto software, sia dal lato firmware, sia dal lato “host computer”
  - Un driver USB è certamente una “bestia difficile”
    - La maggior parte dei dispositivi, tuttavia, utilizza drivers forniti con il sistema operativo Windows
  - I produttori di circuiti integrati mettono a disposizione parti specifiche (controller di macchine fotografiche, dispositivi audio, ecc.) per il lato periferico delle interfacce.

# Firewire

- Firewire è uno standard sviluppato nel 1986 da Apple Computer
  - Standard IEEE 1394 nel 1987
  - iLink, Linux1394 (e altri)
- E' stato sviluppato per realizzare interfacce realtime sincrone ad elevata larghezza di banda
  - IEEE 1394a : 400 Mbs
  - IEEE 1394b : 800 Mbs

# Firewire

- Confronto con USB
  - Entrambe utilizzano bus seriali su coppie intrecciate “twisted” per la trasmissione dati.
  - Firewire consente velocità di trasmissione più elevate (800 Mbps in confronto ai 480 Mbps della USB)
  - Firewire può supportare fino ad un massimo di 63 dispositivi (che sono 127 nel caso della USB)
  - Entrambe consentono lo “hot swapping”
  - Entrambe hanno modalità di funzionamento “plug and play”
  - Firewire è una interfaccia più costosa

# Ethernet

- Protocollo Reti Locali LAN
- IEEE 802.3
- Tre velocità di trasferimento dati su fibra ottica o su coppie di cavi intrecciate “twisted-pair cables”
  - 10 Mbps – 10Base-T Ethernet
  - 100 Mbps – Fast Ethernet
  - 1000 Mbps – Gigabit Ethernet
  - 10000 Mbps – 10-Gigabit Ethernet

# Ethernet

- Cenni Storici
  - Sviluppata come rete sperimentale in cavo coassiale nel 1970 dalla Xerox
    - Originariamente la Velocità era pari a 3 Mbps
  - Alla Xerox si sono poi unite la Intel, la Digital Equipment Corp.; si è giunti, nel 1980 così allo sviluppo delle specifiche per la Versione Ethernet 1.0 con una velocità di 10 Mbps.
  - IEEE 802.3 fu pubblicata come primo standard ufficiale nel 1985

# Ethernet

- Ethernet mantiene i concetti di DTE e DCE introdotti dalla RS-232C
- Ethernet Media Access Control (MAC) sottostrato
  - Incapsulazione dei dati, compreso l'assemblaggio dei frame prima della trasmissione e la scomposizione dei "frame" e la rivelazione degli errori durante e dopo la ricezione
  - Controllo di accesso al mezzo (MAC), compresa l'inizializzazione della trasmissione dei "frame" ed il recupero di guasti in trasmissione

# Ethernet

- Strato Fisico della Ethernet
  - 10Base-T = 10 Mbps, trasmissione in banda base su due coppie intrecciate (“twisted-pair”).
  - 100Base-T2 = 100 Mbps, trasmissione in banda base su due coppie intrecciate (“twisted-pair”).
  - 100Base-T4 = 100 Mbps, trasmissione in banda base su quattro coppie intrecciate (“twisted-pair”).
  - 1000Base-LX = 100 Mbps, trasmissione in banda base su cavo in fibra ottica con segnali con lunghezza d’onda elevata.

# Bluetooth

- Bluetooth è un protocollo seriale “wireless”, che prevede l’uso di segnali a radio frequenza (RF) studiato per la connessione di dispositivi elettronici
  - Introdotto dalla Ericsson (Svezia)
- Perché questo nome?
  - Harald “Bluetooth” era re in Danimarca alla fine del 10° secolo

# Bluetooth

- Noto come “Bluetooth” per la sua predilezione nei confronti dei mirtilli
  - Unì la Danimarca e la Norvegia sotto un’unica legge
  - Fu ammazzato con suo figlio Sven Forkbeard in una battaglia svoltasi nel 986.
  - Il suo nome è stato scelto per l’importanza notevole che hanno le regioni baltiche nella moderna industria delle comunicazioni (Nokia, Ericsson, etc.)

# Bluetooth

- I Dispositivi Bluetooth funzionano nella banda di frequenze ISM (industriale, scientifica e medica)
  - Frequenza centrale 2.45 GHz
- Potenza di uscita limitata ad 1 mW per evitare interferenze con altri dispositivi funzionanti nella medesima banda di frequenza
  - Trasmissione è limitata a 10 metri, senza avere le limitazioni in linea retta dei raggi infrarossi.
    - Consente trasmissioni attraverso muri di separazione

# Bluetooth

- I Dispositivi Bluetooth evitano interferenze usando una tecnica chiamata “diffusione spettrale con salto in frequenza”
  - I Dispositivi cambiano la frequenza di funzionamento ad una velocità pari a 1.6 KHz
  - Le Frequenze sono selezionate in modo random fra 79 valori disponibili tra 2.402 e 2.48 GHz

# Bluetooth

- Quando un dispositivo Bluetooth entra nel campo di frequenze di un altro, il protocollo stabilisce le regole per condividere dati oppure per stabilire se una unità ha necessità di controllare un'altra unità
  - Se così fosse, si viene a costituire una rete PAN (personal-area network) o piconet
    - Una volta che la rete “piconet” si attiva, gli apparati di questa rete sincronizzano la frequenza dei salti per mantenere attiva la comunicazione e per evitare interferenze con altri dispositivi e piconets.
    - Tutto ciò accade automaticamente.

# Bluetooth

- Specifiche
  - Capacità di un canale Piconet: 1 Mbps
  - Negli U.S.A. e in Europa, sono previsti 79 canali a RF (1 MHz ciascuno) con frequenza compresa fra 2.402 e 2.480 GHz
  - In Giappone, sono previsti 23 canali a RF (1 MHz) fra 2.472 e 2.497 GHz
  - In tutti i casi i salti in frequenza si verificano ad un ritmo pari a 1.6 KHz
    - 625  $\mu$ sec ogni time slot

# Bluetooth

- Specifiche
  - Una rete piconet ha un master e fino ad un massimo di sette dispositivi “slave”
    - Il master trasmette nei time slot pari, gli “slaves” trasmettono nei time slot dispari

# WiFi

- Protocollo di rete “Wireless”
  - IEEE 802.11
  - Tre versioni 802.11 (a,b e g)
- IEEE 802.11a
  - Prima versione
  - E’ il più lento e il meno costoso
  - Funziona a 2.4 GHz con una velocità per i dati fino ad un massimo di 11 Mbps

# WiFi

- IEEE 802.11b
  - Seconda versione
  - Funziona a 5 GHz con una velocità massima per i dati pari a 54 Mbps
- IEEE 802.11g
  - Migliore delle due versioni precedenti
  - Funziona a 2.4 GHz (più economici dei 5 MHz) ma mantiene la velocità massima per i dati pari a 54 Mbps (802.11b)

# WiFi

- La implementazione della WiFi è piuttosto complicata sia dal punto di vista RF che dal punto di vista della teoria della comunicazione
- Utilizza la medesima tecnica impiegata dalla interfaccia Bluetooth : “spread-spectrum frequency hopping”
- Impiega tecniche avanzate DSP e tecniche avanzate di compressione dati.

# Bibliografia



- University of California, Santa Barbara
- Department of Electrical and Computer Engineering
- ECE 153-B
- Sensor and Peripheral Interface Design
- Instructor: Prof. Steven Butner