Corso di Architettura dei Sistemi a Microprocessore

BUS

Luigi Coppolino, Giovanni Mazzeo

Outline

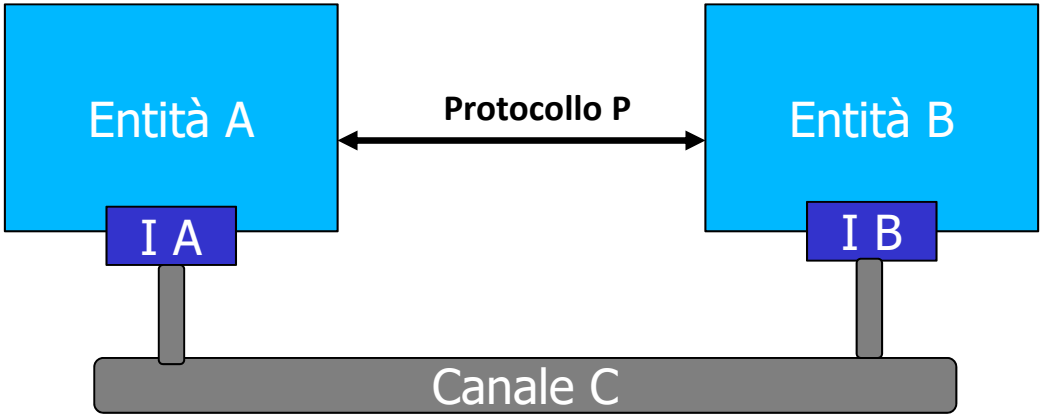
- Generalità su:
  - Sistemi di Interconnessione
  - Bus
- Tipologie di Bus
- Protocolli di Comunicazione
- Bus Sincroni ed Asincroni
- Bus Paralleli
  - AMBA
  - PCI
- Bus Seriali
  - SPI
  - I2C
  - USB

Sistemi di Interconnessi one - 1/2

- Un **sistema digitale** (un calcolatore general purpose, o un sistema dedicato) è composto da **sottosistemi** che scambiano **tre tipi di informazioni:**
  - **dati**, **indirizzi**, informazioni di **stato/controllo**
- Per poter essere effettuata in modo corretto, l'interazione tra gli interlocutori deve comprendere sia lo scambio di informazioni, il **carico utile** (o **payload**), sia lo scambio di **informazioni di servizio** per la sincronizzazione
- Lo scambio di dati può essere di tipo:
  - Uno a uno
  - Uno a molti (*multi-cast*)
  - Uno a tutti (*broadcast*)

Sistemi di Interconnessi one – 2/2

- Due **Entità A e B** scambiano informazioni sulla base di un **Protocollo P**, attraverso un **Canale C**, facendo uso di specifiche **Interfacce I**
- Il canale trasmissivo può essere implementato con **linee distinte** per i dati e lo stato/controllo o con un'**unica linea** sulla quale, nel tempo, transitano tutti i tre tipi di informazione



Parametri Caratteristici di un Sistema di Comunicazione

- L'analisi di un sistema di comunicazione si effettua attraverso i seguenti parametri:
  - **Tempo di Trasmissione T_d** - Tempo di trasferimento di un dato elementare (bit, byte, word)
  - **Tempo di Ciclo T_c** - Tempo di trasferimento di una sequenza di lunghezza fissa di *k* dati elementari
  - **Banda Trasmissiva** – Numero di informazioni che è possibile trasmettere nell'unità di tempo
  - **Latenza** – Intervallo di Tempo tra la trasmissione e la ricezione di un dato elementare
  - **Tasso di Errore** – Percentuale di dati errati rispetto ai dati trasmessi

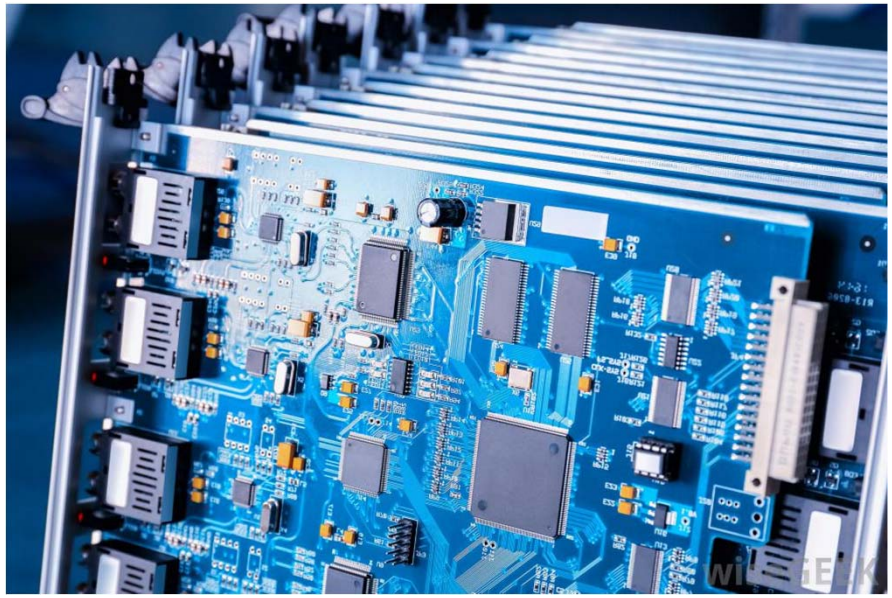
Il Bus – 1/2

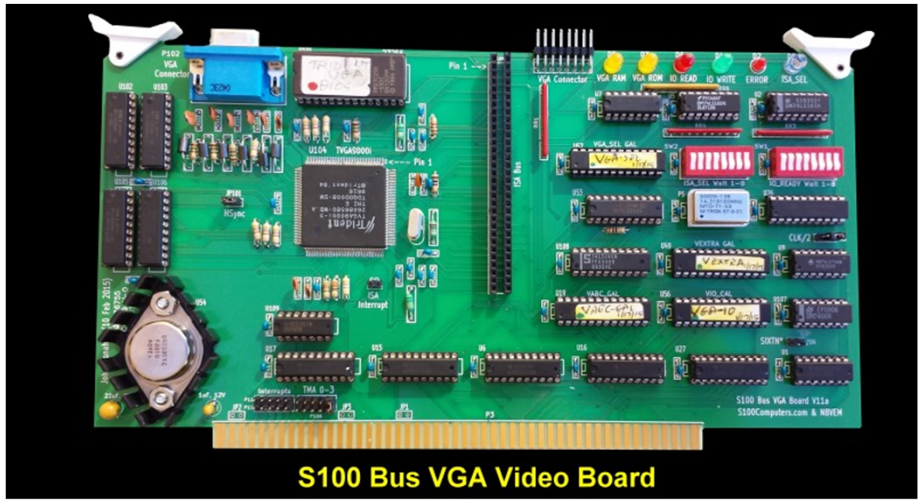
- Il bus è un particolare tipo di rete di interconnessione che condivide un mezzo trasmissivo fra più interlocutori
- I dispositivi collegabili ad un bus possono essere fra loro molto eterogenei sia nelle **velocità operative** sia nel **carico trasmissivo generato**
- I dispositivi connessi possono essere allocati all'interno di un singolo chip, di una scheda o di più schede
- Il bus è una **risorsa condivisa**
- Dunque il suo uso va regolamentato da un apposito **protocollo** al fine di garantire la corretta sincronizzazione dei dispositivi coinvolti nello scambio dei dati

Il Bus – 2/2

- L'uso di un bus per un trasferimento richiede l'assunzione del controllo da parte di un **master**
- Non tutte le unità collegate ad un bus hanno la capacità di operare come master
- Ad es. in un sistema di elaborazione, possono operare da master la CPU, i controllori DMA, i controllori di I/O, ma non i dispositivi di I/O e le memorie che sono entità passive (o **slave)**
- Nel caso in cui più unità connesse ad un bus siano in grado di svolgere il ruolo di master, è necessaria la presenza di un **arbitro** che gestisca le contese
- I circuiti di arbitraggio possono essere o **distributi** o **centralizzati**

Esempi Reali

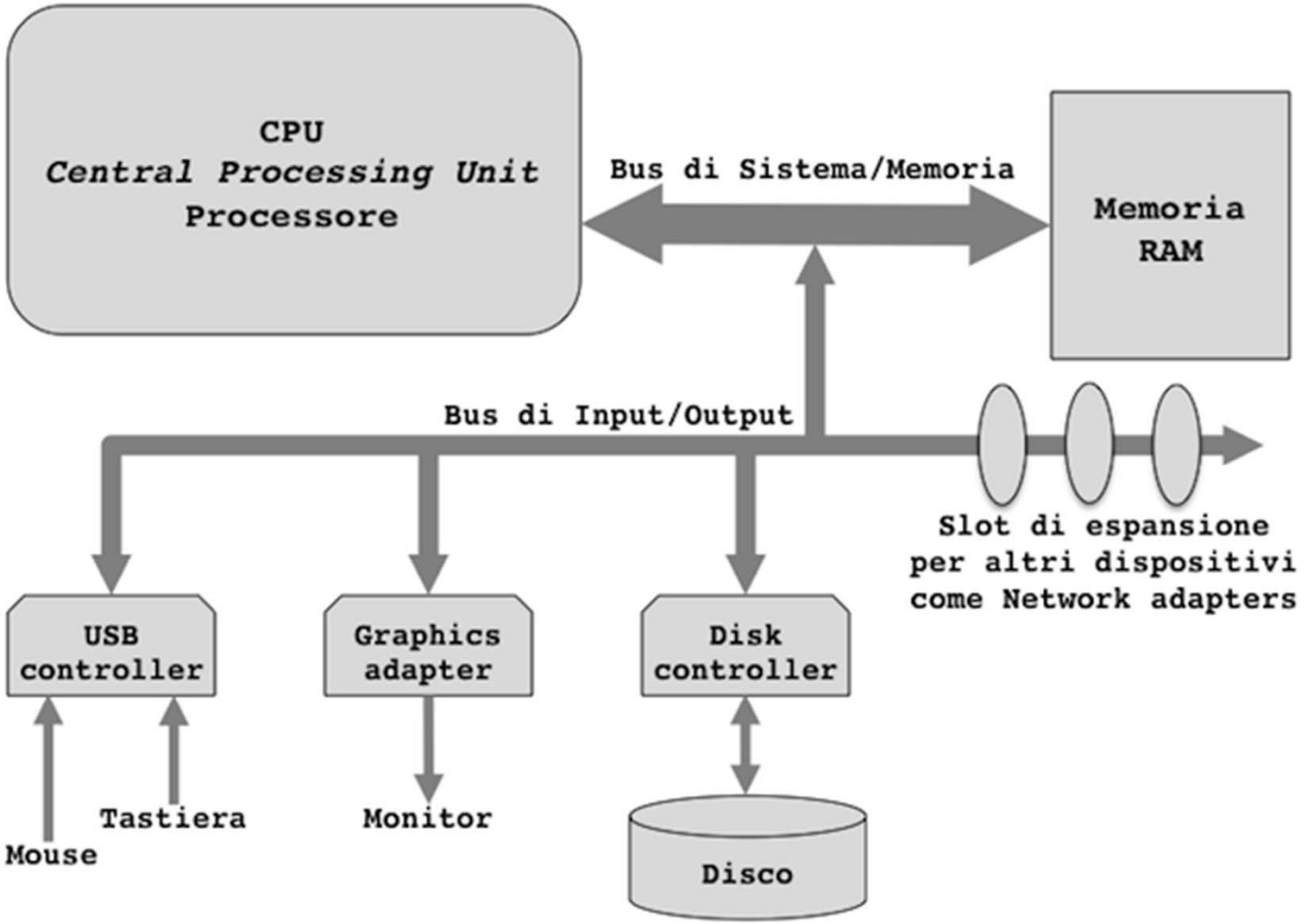




Bus Gerarchici

- Per incrementare l'efficienza dei bus e le relative velocità di trasferimento si è agito:
  - **A livello tecnologico** – impiegando componenti più performanti per aumentare la frequenza dei cicli di trasmissione (nell'ordnie dei MHz) e riducendo la dimensione delle schede che ha consentito di ridurre la lunghezza delle linee
  - **A livello di architettura** – strutturando i bus in gerarchie di bus eterogenei raggruppando insiemi di dispositivi omogenei in termini di velocità. I bus ad elevata velocità sono usati per connettere i processori con le memorie, la cache, eventuali coprocessori. Quelli a più bassa per dispositivi di I/O
- La connessione tra i vari tipi di bus è effettuata da un particolare nodo di comunicazione detto **bridge**

Bus Gerarchici



Protocolli di Comunicazione

- La gestione delle varie forme di comunicazione avviene attraverso protocolli di comunicazione
- Un protocollo è un accordo preventivo fra due o più interlocutori
- I protocolli interessano trasferimenti ai vari livelli di un sistema digitale (ad es. TCP/IP usato in Internet)
- Noi ci soffermiamo su protocolli quasi sempre sviluppati in HW per la trasmissione di dati tra unità di un calcolatore
- Tali protocolli sono caratterizzati da un'elevata efficienza di trasmissione, bassa latenza, basso tasso d'errore, e ridotta distanza

Scelta di un Protocollo

- La scelta di un protocollo è effettuata in fase di progetto sulla base delle caratteristiche del canale trasmissivo e degli interlocutori coinvolti, in particolare delle loro velocità operative
- I principali elementi che influenzano la scelta del protocollo sono:
  - Numero di interlocutori
  - Oggetto dell'interazione (scambio dati, sincronizzazione)
  - Modalità di interazione (sincrona, asincrona)
  - Simmetria nella gestione del protocollo (master-slave o multimaster)
  - Caratteristiche del canale trasmissivo
  - Trattamento o meno degli errori

Organizzazione Temporale di un Protocollo

- La comunicazione fra due interlocutori è organizzata in una successione temporale di **sessioni**
- Una sessione di trasferimento è definita da una sequenza di messaggi utili per:
  - Acquisizione del canale trasmissivo
  - Selezione dell'unità destinataria della trasmissione
  - Scambio di dati
  - Eventuale controllo e/o correzione di errori
  - Rilascio del canale

Protocolli Sincroni e Asincroni

- Nel caso di **protocolli sincroni**, le due interfacce operano alla stessa velocità e i trasferimenti avvengono all'interno di uno o più cicli di clock, fra essi condiviso.
- Il clock è derivato da un clock stabile, detto **base dei tempi**, in grado di garantire la necessaria precisione ai fini della comunicazione
- Ovviamente è necessario che i entrambi gli interlocutori siano in grado di sostenere la velocità del clock utilizzato
- Nel caso dei **protocolli asincroni** lo scambio di informazioni avviene attraverso segnali di controllo/stato che gli interlocutori si scambiano, in aggiunta a quelli dei dati, all'inizio ed alla fine della comunicazione

Confronto tra Protocolli Sincroni ed Asincroni

- Protocolli Sincroni:
  - Maggiore semplicità implementativa dei circuiti
  - Prestazioni superiori in termini di velocità di trasferimento
  - Minore impiego di HW
- Protocolli Asincroni:
  - Più complessi, sopratutto quando il numero di interlocutori cresce
  - Più efficienti per interfacciare dispositivi con velocità operative differenti, non note apriori
  - Maggiore impiego di HW

Handshaking – 1/3

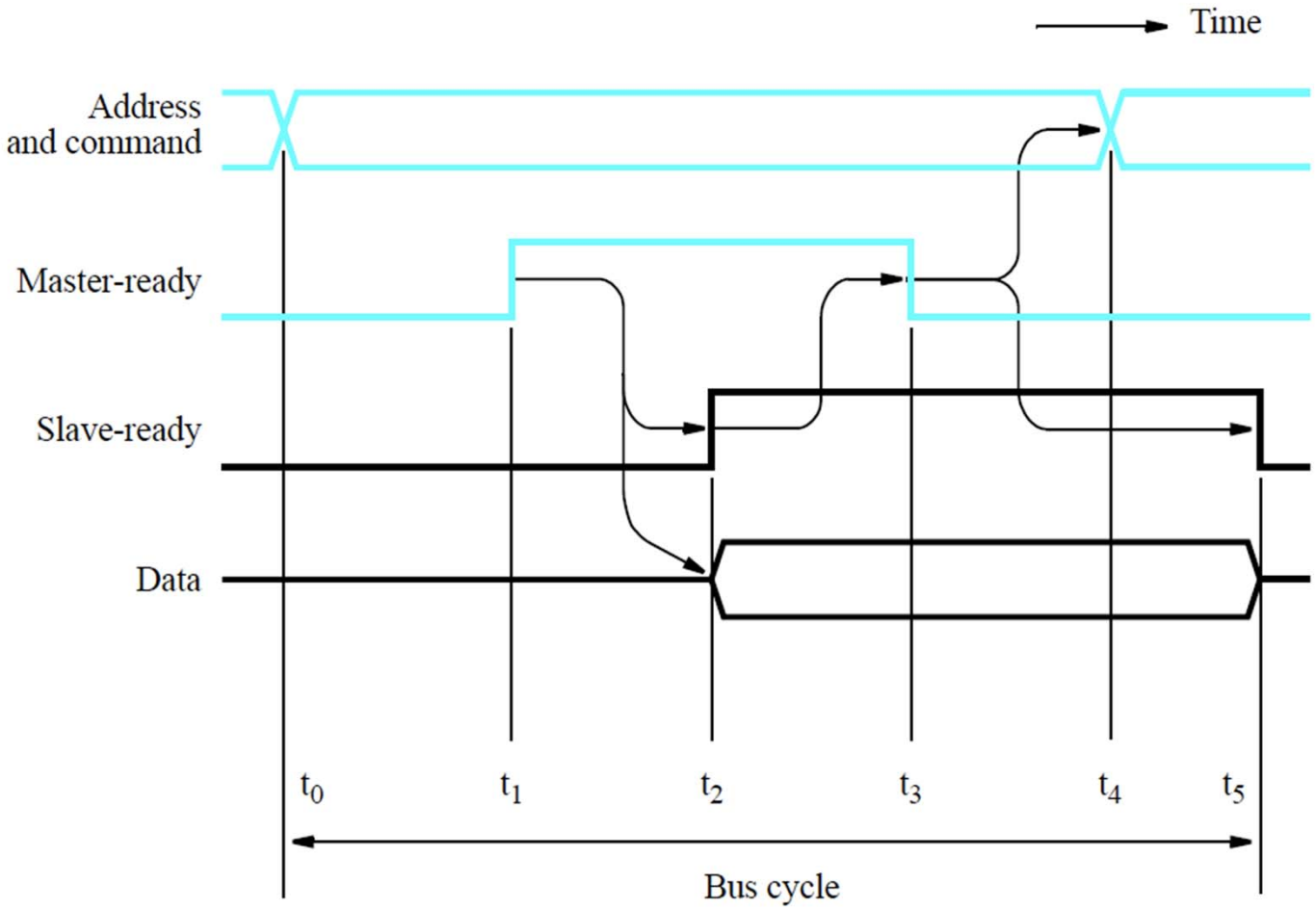
- Un esempio di protocollo asincrono particolarmente semplice è il cosiddetto **handshaking a due variabili.** In un ciclo di lettura:

  - **M=S=0 (riposo)** Il Master in qualsiasi momento, dopo aver verificato che S=0, può decidere di iniziare la comunicazione. Lo Slave controlla continuamente il valore di M in attesa della transizione a 1.
  - **M=1, S=0 (inizio lettura)** Il Master, dopo aver preparato l'indirizzo, invia l'ordine di inizio lettura; attende dunque la risposta dello Slave, ossia il passaggio di S al valore 1. Lo Slave riconosce l'ordine e da inizio all'esecuzione del comando; lo Slave non risponde (ossia mantiene S=0) fin quando non ha terminato l'esecuzione del comando.

Handshaking – 2/3

- **M=1, S=1 (fine lettura)** Il Master legge dal bus il valore dello Slave; mantiene M=1 fin quando non ha "consumato" completamente i dati. Lo Slave ha risposto al Master dopo aver fornito il risultato della lettura sul bus; mantiene questi valori e l'indicazione S=1 fino a quando non si accorge del passaggio di M a 0.
- **M=0, S=1 (conferma lettura)** Il Master ha dato conferma allo Slave dell'avvenuta lettura, ponendo M=0; in questa condizione il Master può procedere con altre attività, ma non può mandare un nuovo comando allo Slave. Lo Slave verificando che M=0 si accorge che la lettura é terminata, e stacca la sua connessione in uscita sul bus; solo dopo aver staccato la propria connessione passa allo stato successivo.
- **M=0, S=0 (bus libero)** Lo Slave segnala di aver rilasciato il bus ponendo S=0, quindi si riporta nello stato iniziale di "attesa di ordini".

Handshaking – 3/3

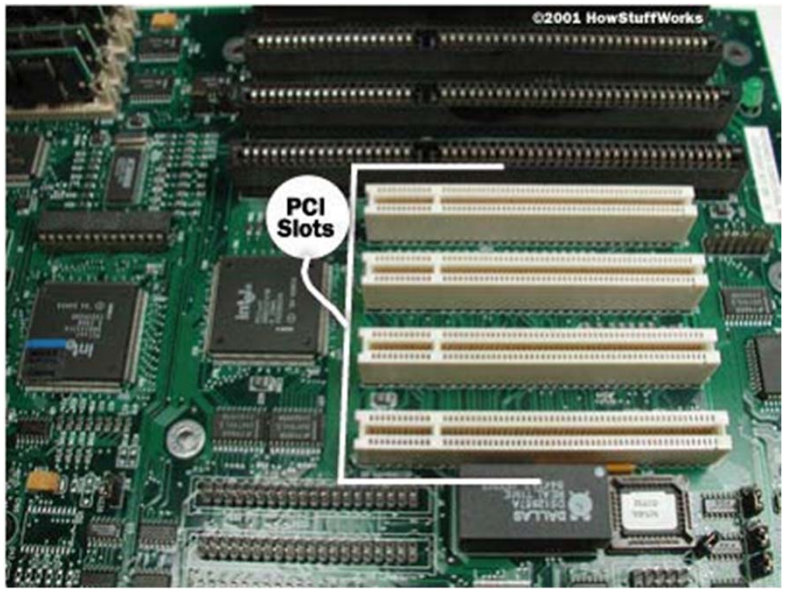


Bus Sincroni - Paralleli

- A seconda del contesto e dei requisiti di velocità, i bus possono essere o **seriali** o **paralleli**
- Per trasmissione parallela si intende la trasmissione di dati in cui tutti i bit sono trasferiti contemporaneamente lungo canali separati del mezzo di comunicazione.
- Un cavo che effettua una trasmissione parallela a *n* bit è formato da almeno *n* linee separate.
- In realtà il cavo sarà dotato quasi sicuramente di una linea aggiuntiva per la massa e anche di altre linee di controllo come quella del clock.

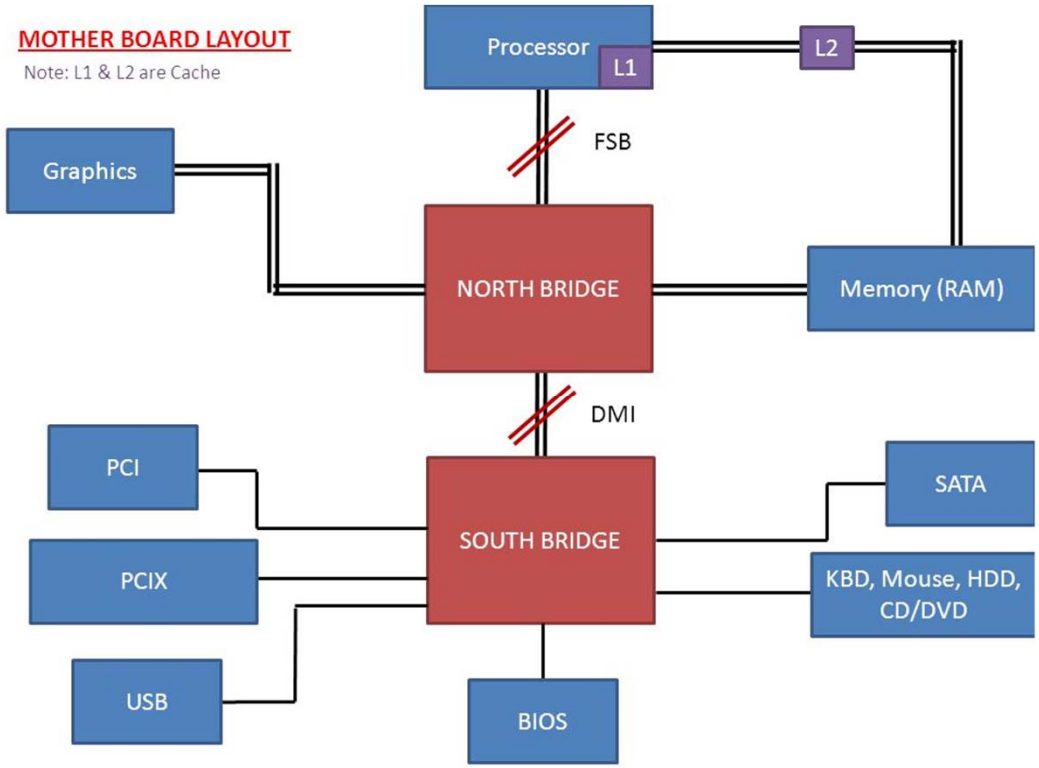
Bus Paralleli: PCI – 1/2

- Il bus sincrono parallelo Peripheral Component Interconnect (PCI) rilasciato nel 1993 per collegare la CPU con le più svariate periferiche interne al computer (o schede) attraverso la scheda madre.
- Molto **utilizzato nei calcolatori general-purpose**
- La velocità di trasmissione dell'interfaccia PCI è rimasta negli anni ancorata a **132 MB/s**, generata da una trasmissione dati con frequenza di clock pari a 33 MHz a 32 bit.



Bus Paralleli: PCI – 2/2

- PCI fa utilizzo di due bridge, detti **north** e **south bridge** per disaccoppiare le unità del calcolatore aventi requisiti prestazionali differenti

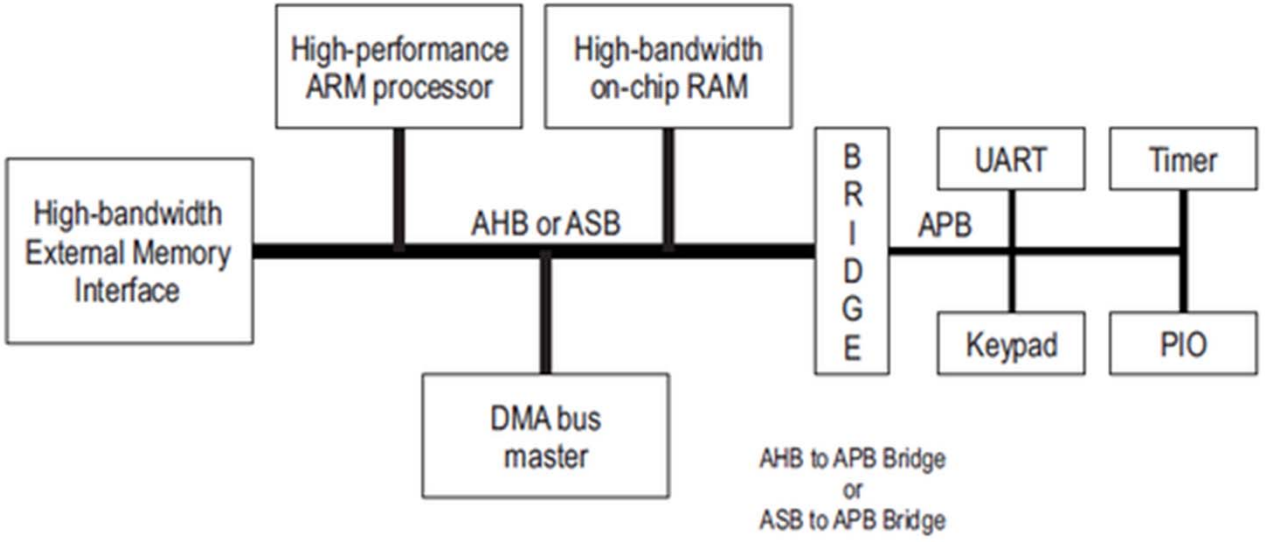


Bus Paralleli: AMBA – 1/2

- Advanced Microcontroller Bus Architecture (AMBA) è uno standard di comunicazione on-chip per microcontrollori con performance elevate
- Sviluppato da ARM nel 1996
- **Molto utilizzato nei sistemi embedded**
- All'interno delle specifiche AMBA sono definiti tre diversi tipi di bus:
  - Advanced High-performance Bus (AHB)
  - Advanced System Bus (ASB)
  - Advanced Peripheral Bus (APB)

Bus Paralleli: AMBA – 2/2

- AMBA si pone i seguenti 4 obiettivi:
  - Facilitare il *right-first-time development*
  - Essere *technology-independent*
  - Incoraggiare il *modular system design*
  - Minimizzare l'occupazione sul silicio



AMBA AHB

- AHB è il bus AMBA ad alte prestazioni
- AHB è caratterizzatio da linee dati ampie (64/128 bits).
- Componenti tipici di AMBA AHB sono:
  - **AHB master**: che inizia eventuali operazioni di lettura o scrittura fornendo uno specifico indirizzo
  - **AHB slave**: che risponde alle operazioni di lettura/scrittura nel suo spazio di indirizzamento
  - **AHB arbiter**: assicura che non ci sia un altro master che voglia iniziare un operazione se c'è già in corso una
  - **AHB decoder:** utilizzato per decodificare gli indirizzi degli slave

AMBA APB

- APB è utilizzato per interconnettere i dispositivi periferici con il resto del sistema aventi basse frequenze operative
- Ha prestazioni più basse rispetto ad AHB
- L'APB Bridge è l'unico master nel bus APB
- A sua volta l'APB bridge è anche uno slave rispetto ad AHB

Bus Sincroni - Seriali

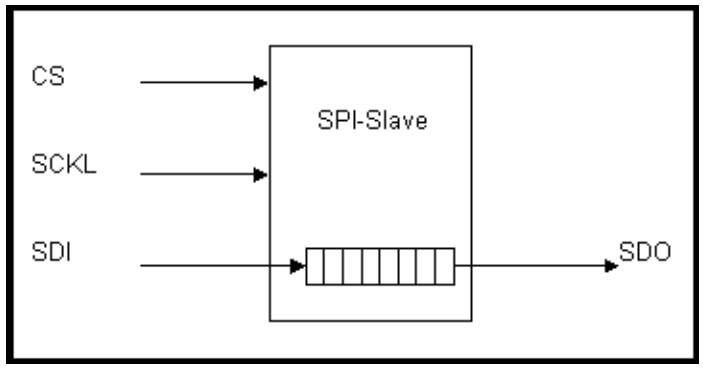
- I bus seriali vogliono favorire, specialmente nei sistemi embedded, la **semplice ed economica interconnessione delle unità digitali**
- Nei bus seriali la trasmissione dati avviene utilizzando segnali trasferiti su **due o quattro linee elettriche**, secondo un protocollo seriale.
- Rispetto al bus parallelo occupa molta **meno area sul silicio**, sia per il numero ridotto di linee che per connettori molto più ridotti
- Di contro, a parità di frequenza, le **prestazioni sono più basse** dei bus paralleli
- Oggi i bus seriali sono utilizzati massivamente per la connessione delle periferiche ad un processore ed, in particolar modo, per connettere la CPU a componenti off-chip.

Bus Seriali: SPI

- Il **Serial Peripheral Interface (SPI)** è un protocollo di comunicazione sviluppato da Motorola
- E' un protocollo **Sincrono e Seriale**
- E' solitamente utilizzato per comunicazioni seriali tra un processore e le periferiche
- Funziona in configurazione **Master/Slave**. Il Master potrebbe essere il microcontrollore, e lo Slave la periferica.
- **Vantaggi**: E' un bus che ha un protocollo di comunicazione molto semplice e dunque un overhead sul processore basso; è tra i più veloci nella sua categoria
- **Svantaggi**: Più crescono le periferiche collegate e più aumentano le linee del bus -> maggiore area di occupazione delle schede

SPI: Operazioni - 1/2

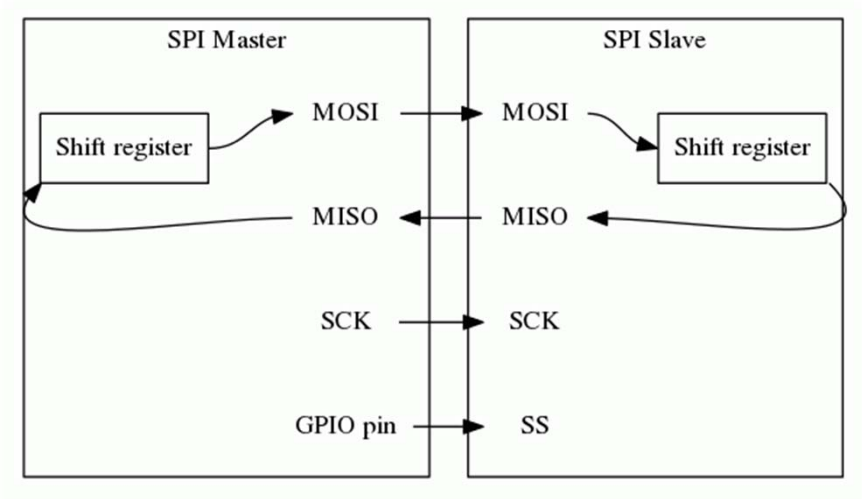
- Il bus SPI fa uso di 4 linee:
  - **Serial Clock (SCLK)**
  - **Chip Select (CS)**
  - **Serial Data In (SDI) o Master input Slave Output (MISO)**
  - **Serial Data Out (SDO) o Master Output Slave Input (MOSI)**

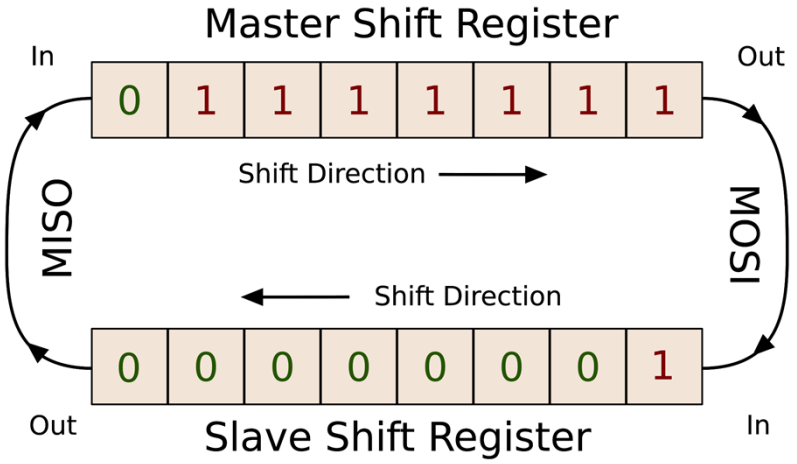


- Il bus permette che solo un Master possa esserci durante una comunicazione
- Il numero di Slaves (o periferiche) dipende dal numero di linee Chip Select (CS) in uscita dal Master
- La frequenza di funzionamento di SPI è solitamente nell'intorno di 1-2 MHz

SPI: Operazioni - 2/2

- Ogni periferica (sia Master che Slave) presentano al loro interno dei **Shift Registers** collegati alle linee MOSI (SDO) e MISO (SDI)
- Ad ogni colpo di clock l'ultimo bit viene inviato dallo shift register del master a quello dello slave. Dopo 8 colpi di clock, il dato sarà trasmesso completamente.

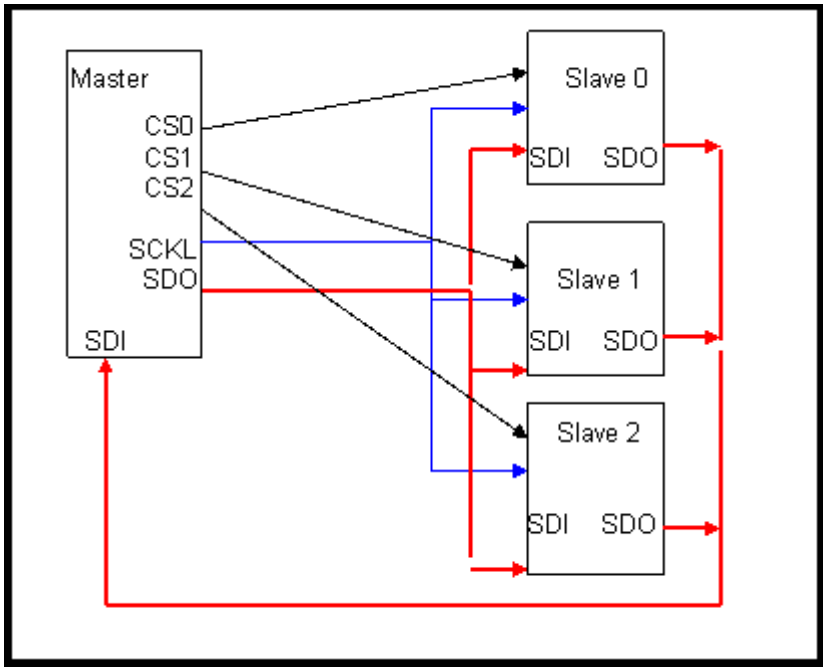




Master-Slave Setup – 1/2

- Un Master per iniziare una comunicazione attiva il clock sulla linea SCKL e seleziona lo slave col quale vuole parlare attraverto l'opportuna linea CS
- Osservare che le linee SDO sono collegate alle SDI. Ciò vale per entrambe le connessioni Master-Slave e Slave-Master

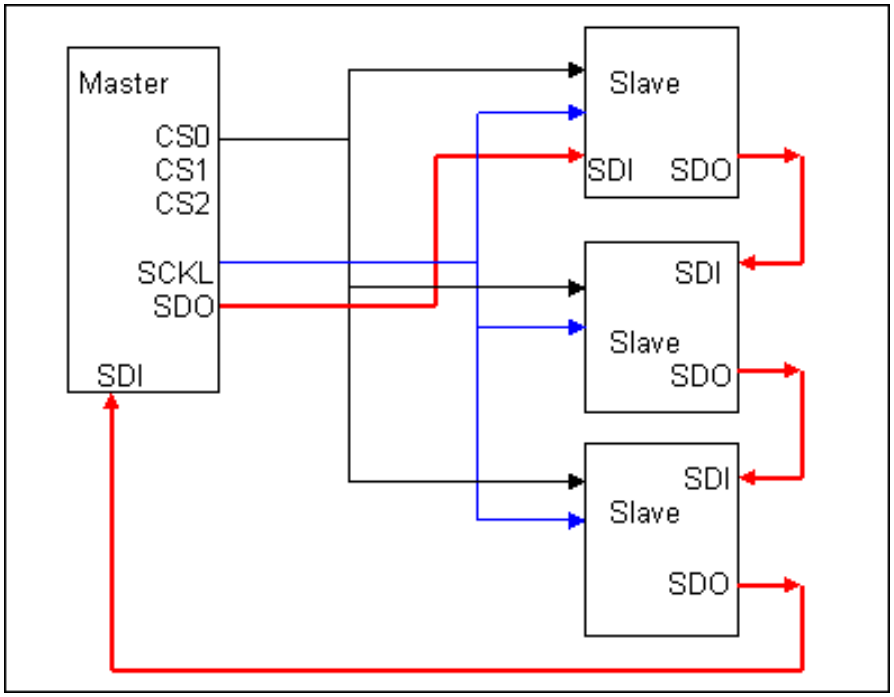
Multiple Independent Slave Configuration



Master-Slave Setup – 2/2

- In questo caso gli Slave sono collegati in cascata tra di loro
- L'uscita di uno andrà in ingresso dell'altro
- In questo caso, gli Slave sono trattati come un solo Slave e connessi alla stessa linea CS

Multiple Slave Cascaded



Bus Seriali: I2C

- Il Bus **Inter Integrated Circuit** (**I2C**) (si legge ”I square si”) è un bus di comunicazione **seriale**, **sincrono, bifilare** basato su pattern Master/Slave
- E’ stato sviluppato da Philips nel 1982 per impiegarlo all’interno di televisori
- E’ spesso utilizzato nei microcontrollori per connettere *low-speed devices* come, EEPROMs, A/D and D/A converters, I/O interfaces ed altre periferiche simili dei sistemi embedded.
- I2C è un bus molto popolare per la sua bassa occupazione di area: ci possono essere uno o più master che controllano un numero limitato di dispositivi di I/O con soli due fili

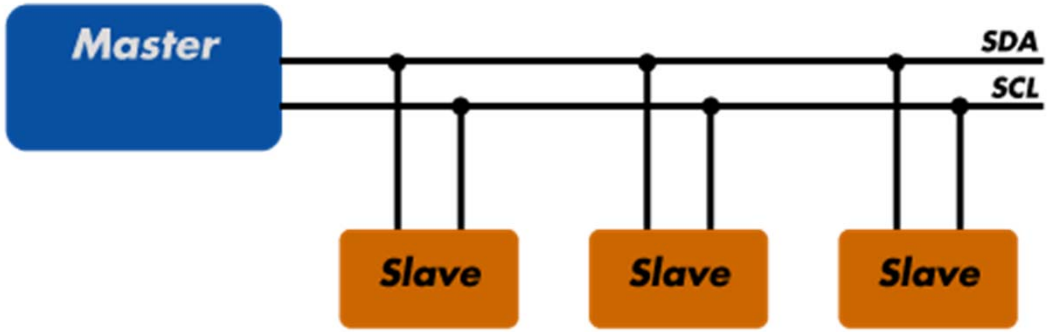
I2C

- I2C, a differenza di SPI, è un protocollo multi-master
- Permette l'utilizzo del protocollo in tre modalità aventi data rate diversi: 100 kbps (standard mode), 400 kbps (fast mode) and 3.4 Mbps (high-speed mode)
- Il bus è composto da 2 sole linee (**SDA** per i dati ed **SCL** per il clock)
- Ogni slave device è caratterizza da un indirizzo di 7 bit che lo identifica in modo univoco

I2C: Operazioni

- Il *Master* invia un segnale di START che informerà tutti gli *Slave* che devono ascoltare sulla linea dati
- Il master dunque invia l ADDRESS dello slave col quale vuole parlare ed un *flag* che indica se vuole fare un'operazione di Read o Write
- Lo Slave con quell'indirizzo risponderà con un ACK
- La comunicazione quindi avrà luogo tra il master e lo slave. Entrambi possono ricevere o trasmettere dati a seconda di come è stata impostata la comunicazione precedenemtenete (se il flag è read o write)
- Ad ogni colpo di clock un bit sarà inviato allo slave
- Quando la comunicazione è completata il master invia uno STOP che indica la terminazione.

I2C


Master
SDA
SCL
Slave
Slave
Slave


START
READ/
WRITE
ACK
ACK
Slave Address
Data
Data
STOP

I2C

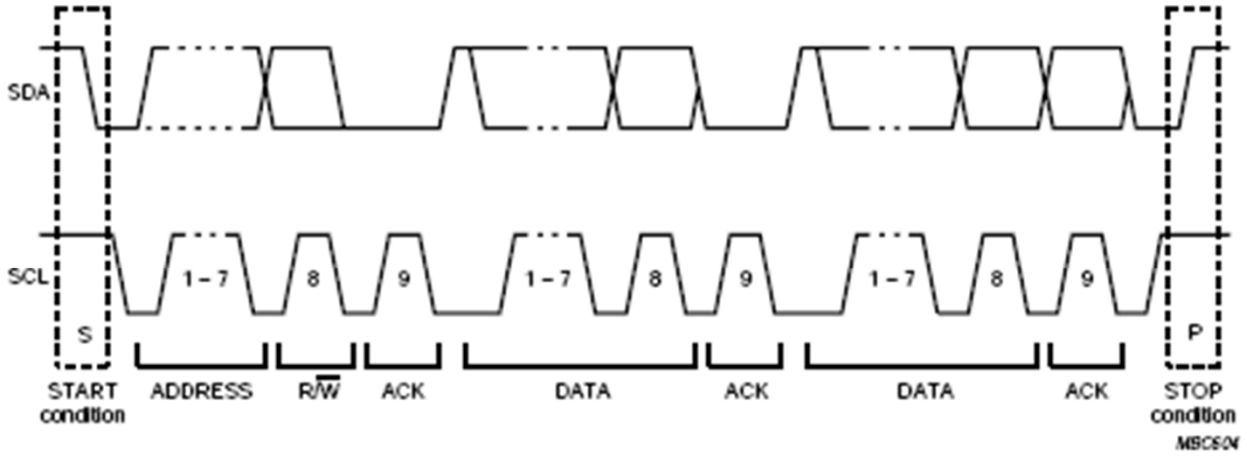
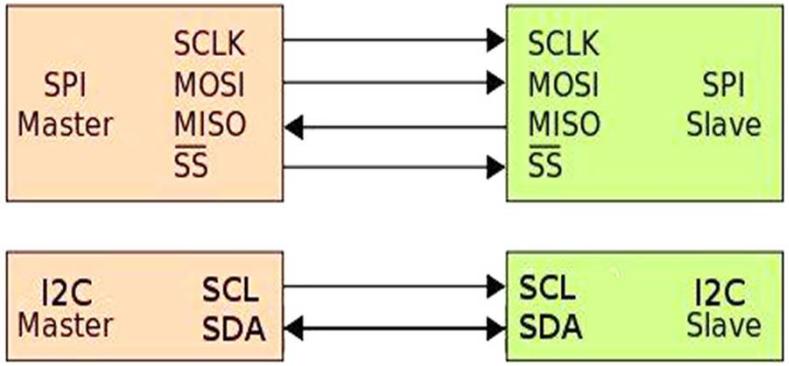


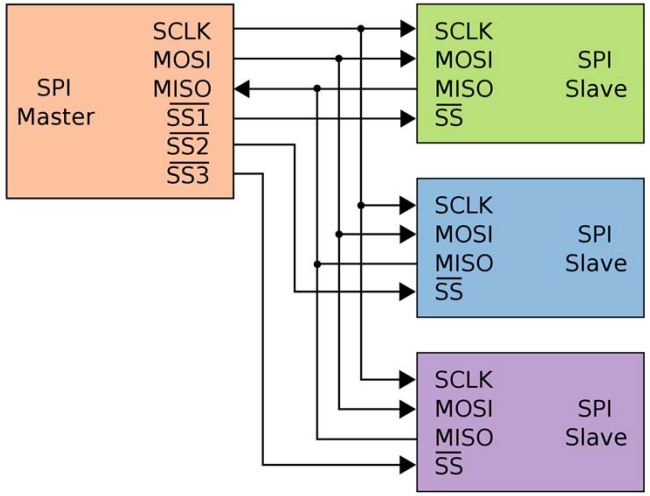
Fig.10 A complete data transfer.

SPI vs I2C

- I due bus possono essere comparati sulla base di:
  - **Topologia**
    - I2C richiede solo 2 linee, mentre SPI almeno 4 che crescono al variare del numero di slave. Ciò ha ovviamente implicazioni sull'occupazione d'area dei chip
    - L'unico svantaggio di I2C è che ha uno spazio di indirizzi limitato (7 bit)
  - **Throughput / Speed:**
    - In caso di trasferimenti ad alta velocità, SPI è migliore di I2C. SPI è full-duplex mentre I2C non lo è. SPI non presenta limiti di velocità. Alcune implementazioni toccano i 10Mbps. I2C è invece limitato.
  - **Ease-of-Implementation:**
    - SPI è molto più semplice rispetto a I2C da implementare

SPI vs I2C


SPI
Master
SCLK
MOSI
MISO
SS
SCLK
MOSI
MISO
SS
SPI
Slave
I2C
Master
SCL
SDA
SCL
SDA
I2C
Slave


SPI
Master
SCLK
MOSI
MISO
SS1
SS2
SS3
SCLK
MOSI
MISO
SS
SPI
Slave
SCLK
MOSI
MISO
SS
SPI
Slave
SCLK
MOSI
MISO
SS
SPI
Slave

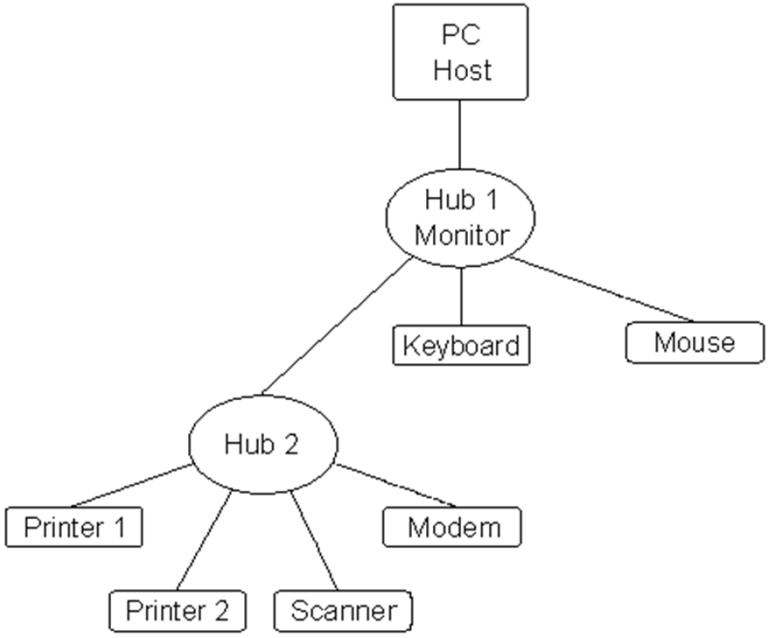
Bus Seriali: USB – 1/5

- L'Universal Serial Bus (USB) è un particolare tipo di bus sviluppato nel 1995 da un insieme d'aziende (Compaq, Digital, IBM, Intel, Microsoft, NEC e Northern Telecom) allo scopo di utilizzare un'unica interfaccia per collegare periferiche accessorie, sostituendo le porte parallele e seriali e altri ingressi e uscite esistenti sui computer.

Nome	Versione	Velocità teorica	Velocità reale	Data di pubblicazione
Low-Speed	USB 1.0	1,5 Mbps (187,5 KB/sec)	1 Mbps (125 KB/sec)	Gennaio 1996
Full-Speed	USB 1.1	12 Mbps (1,5 MB/sec)	7 Mbps (875 KB/sec)	Agosto 1998
Hi-Speed	USB 2.0	480 Mbps (60 MB/sec)	280 Mbps (35 MB/sec)	Aprile 2000
Super-Speed	USB 3.0	4,8 Gbps (600 MB/sec)	3,2 Gbps (400 MB/sec)	Settembre 2008
Super-Speed+	USB 3.1	10 Gbps (1,25 GB/sec)	7,2 Gbps (900 MB/sec)	Gennaio 2013

Bus Seriali: USB – 2/5

- USB permette di connettere al calcolatore fino a 127 dispositivi
- I dispositivi sono collegati al PC in una struttura a "**radice**" (stella) con un massimo di 6 livelli di profondità, in cui il vertice principale è occupato dal calcolatore stesso detto anche host



Bus Seriali: USB – 3/5

- I dispositivi periferici non possono comunicare tra loro, ma rispondono unicamente ai comandi e alle direttive dell'Host.
- Non è possibile servirsi di cavi e commutatori ordinari per collegare simultaneamente lo stesso dispositivo a più computer
- Il protocollo di comunicazione funziona solo se l'Host è unico.
- Confrontato con gli altri bus l'USB risulta molto più flessibile.
- Consente la modalità "*Hot Plug & Play*"
- È permesso connettere e disconnettere periferiche al PC senza riavviarlo.
- Con gli altri tipi di interfacce la connessione di una periferica col PC acceso spesso comporta il blocco del sistema operativo o addirittura in casi estremi il danneggiamento dell'hardware.

Bus Seriali: USB – 4/5

- USB permette di alimentare le periferiche.
- Il connettore dell'USB dispone di quattro terminali:
  - Due per il trasferimento dei dati vero e proprio,
  - Altri due dedicati all'alimentazione (5V), che possono essere sfruttati dalle periferiche per prelevare l'energia di cui necessitano.
  - Questa possibilità comporta l'assenza di alimentatori esterni, riducendo notevolmente il volume delle connessioni esterne al PC.

Bus Seriali: USB – 5/5

- USB prevede tre tipologie di periferiche:
  - **Host** - è il controller del bus e sovraintende a tutte le transizioni indirizzando e interrogando i device.
  - **Device USB** - aggiunge capacità all'host può essere low speed o full speed è caratterizzato da un indirizzo e dai suoi endpoints
  - **Hub USB -** è un replicatore di porte che rigenera i segnali