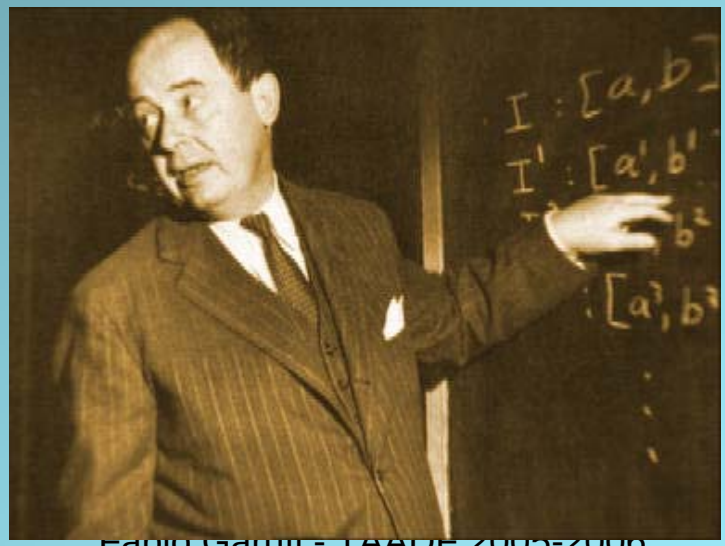
Tecniche Automatiche di Acquisizione Dati

Richiami di Architettura degli elaboratori

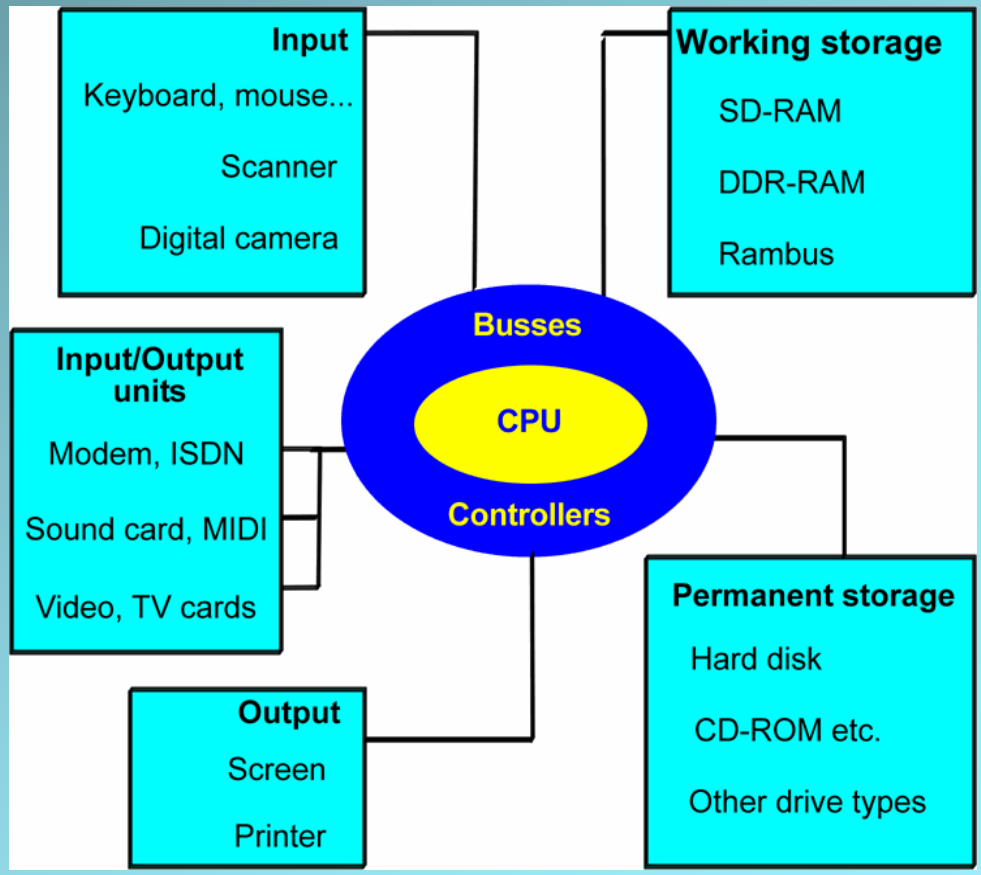
Modello di Von Neumann

- L’architettura del moderno microcomputer ha radici che risalgono agli anni ‘40.
- Il Matematico ungherese John Von Neumann (1903-1957) ha sviluppato un modello di computer che ancor oggi è utilizzato per la descrizione degli elaboratori



Il modello

- Von Neumann divise l'hardware del computer nelle seguenti parti:
  - CPU
  - Input
  - Output
  - Working Storage (archiviazione temporanea)
  - Permanent Storage
- Applicando il modello di Von Neumann al personal computer odierno otterremmo uno schema come in figura:

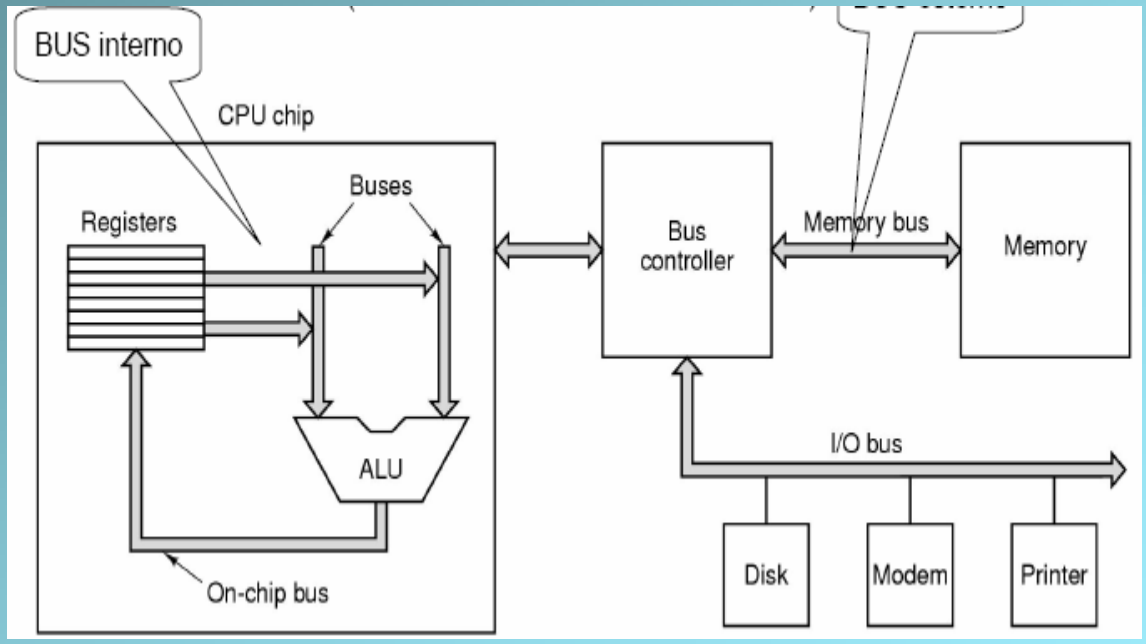


Cosa è un BUS?

- È un mezzo di trasmissione pubblico dei dati
- Fisicamente è composto da più connettori che collegano fra loro più circuiti e permettono loro lo scambio di dati.
- I dati possono essere trasmessi in modo seriale (bastano pochi conduttori) o parallelo (serve un conduttore per ciascun bit della parola)
- Nei bus paralleli, spesso, i conduttori che trasmettono i dati (linee dati) sono separati da quelli che trasmettono gli indirizzi (linee indirizzi).
- Sempre i segnali di controllo sono trasmessi su connettori dedicati.
- Il **protocollo** di trasmissione è definito da un circuito detto **Bus Controller**

I componenti di un elaboratore: I BUS

- I bus all'interno del computer sono linee di trasmissione che collegano le varie unità funzionali del computer
- Si dividono in
  - Bus Interni (per es. Interni alla CPU)
  - Bus Esterni (solitamente standardizzati)



I componenti di un elaboratore: la CPU

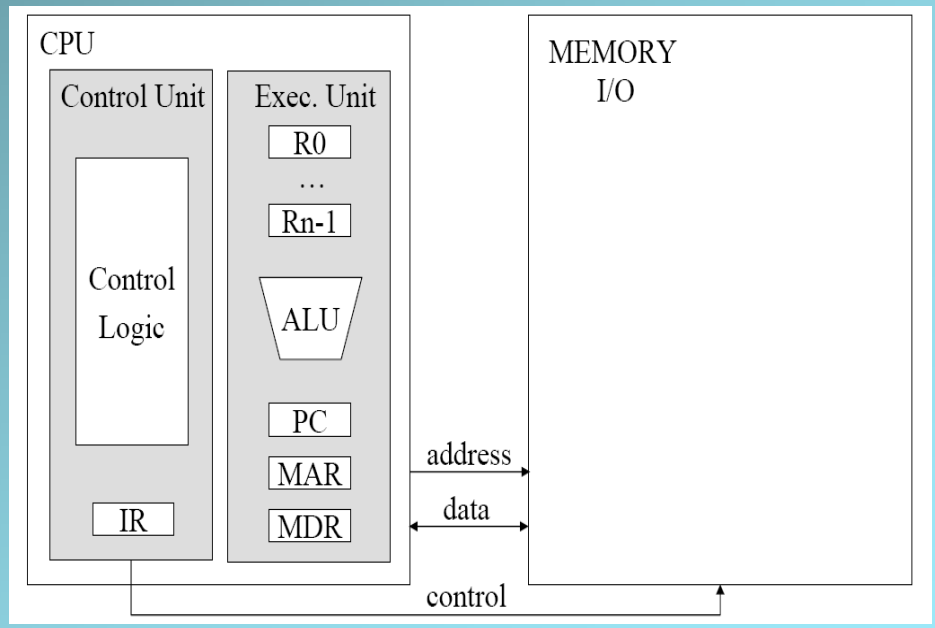
La CPU (Central Processing Unit), è il cuore dell'elaboratore.

Dal punto di vista funzionale la CPU:

- Preleva le istruzioni dalla memoria del calcolatore
- Esegue le istruzioni prelevate (somme, moltiplicazioni, confronti, modifica registri)
- Legge e scrive I dati dalla memoria centrale del computer e li elabora
- Invia o riceve dati dalle periferiche.

La CPU: struttura interna

- Il processore è un complesso insieme di sotto-unità funzionali combinatorie e sequenziali.
  - **L'unità di controllo**: legge le istruzioni dalla memoria,ne determina il tipo e genera I segnali di controllo
  - **L'unità aritmetico-logica (ALU)**: esegue le operazioni necessarie al completamento delle istruzioni
  - **I Registri**: elementi di memoria interni alla CPU usati per immagazzinare temporaneamente i dati da elaborare, i risultati e le informazioni di controllo.

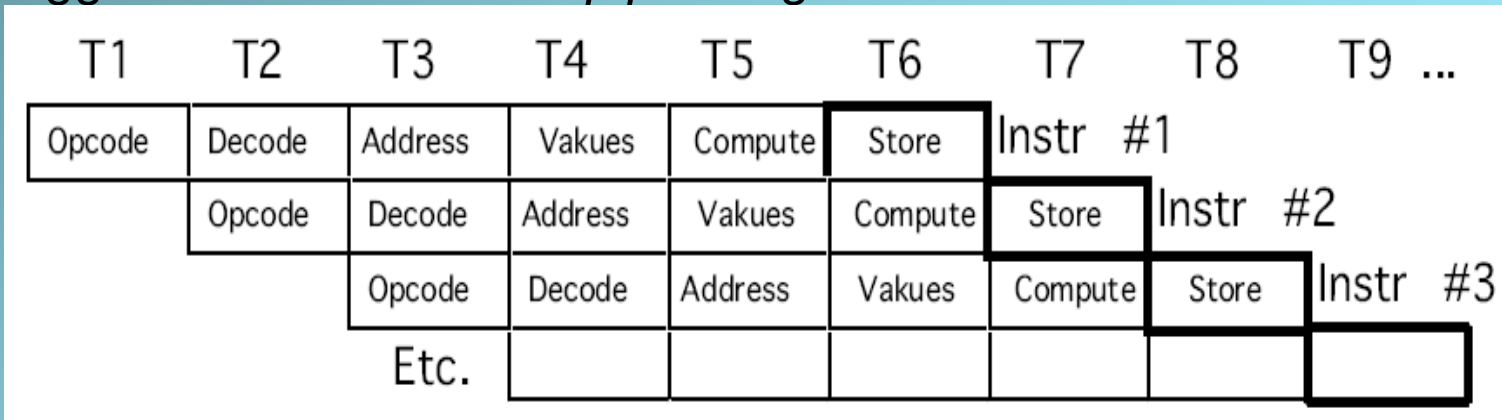


Esecuzione delle istruzioni

1	fase di fetch	Prendi l'istruzione seguente dalla memoria (l'indirizzo in memoria e' indicato dal registro PC) e mettila nel registro delle istruzioni (IR)
2		Cambia il program counter per indicare l'istruzione seguente (PC = PC + 1)
3	fase di decodifica	Determina il tipo dell'istruzione appena letta
4		Se l'istruzione usa una parola in memoria, determina dove si trova
5	fase di esecuzione	Metti la parola, se necessario in un registro della CPU
6		Esegui l'istruzione
7		Torna al punto 1 per eseguire l'istruzione successiva

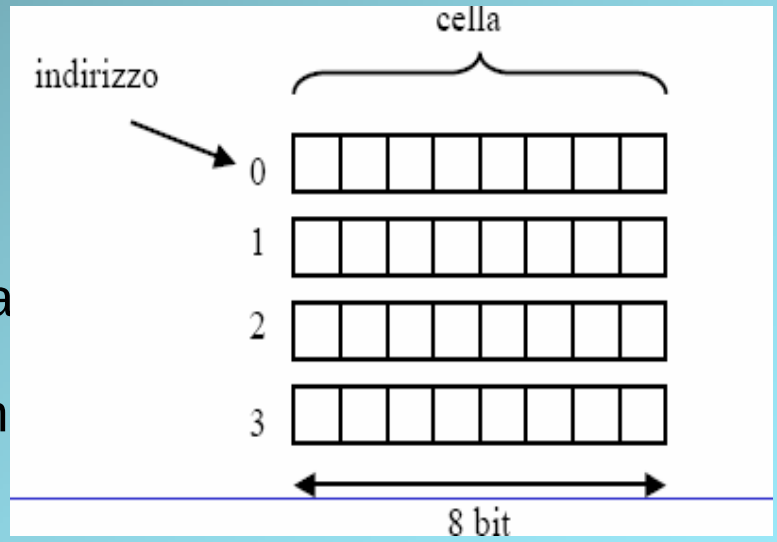
Clock e pipelining

- La sequenza di passi del processo di esecuzione è sincronizzata da
- un segnale esterno periodico (clock). Frequenze tipiche del clock
- Per un moderno calcolatore variano tra alcune centinaia di MHz e
- 1-2 GHz
- In generale *per una data architettura* la velocità aumenta con
- l'aumentare della frequenza di clock.
- Se la fase IF è l'unica che richiede accesso alla RAM, e le fasi
- di decodifica ed esecuzione sono molto più rapide, questo
- suggerisce la tecnica del *pipelining*



La memoria centrale

- L’unità di base della memoria è il BIT che rappresenta un numero binario [0-1]
- La memoria è organizzata in “celle” (o locazioni), ciascuna di k bit.
- In una memoria contenente n celle, ogni cella è univocamente identificata da un indirizzo numerico cha va da 0 a n-1.
- La capacità della memoria si misura in byte. 8 bit sono un byte (4 un Nibble)
- La minima unità indirizzabile di memoria è detta parola
- Nei moderni calcolatori le memorie sono organizzate in parole da 4 o 8 byte.



Rappresentazione dei dati in memoria (endianness)

- I dati in memoria sono sequenze di bit, ma qual'è il byte più significativo?
- Ci sono due possibilità:
  - il MSB della parola è quello che ha l'indirizzo maggiore (Little Endian – finisce con il più piccolo)
  - È quello con l'indirizzo minore (Big Endian)

  Esempio: 4A3B2C1D è immagazzinato nelle locazioni da 100 a 103:

  Big Endian

100	101	102	103
4A	3B	2C	1D

  Little Endian

100	101	102	103
1D	2C	3B	4A

- entrambe sono utilizzate dai costruttori di CPU
  - Intel© X86 sono Little Endian
  - Motorola © 68000 e PPC970 (G5) sono Big Endian
  - L'Internet protocol definisce un Network Byte Order come big endian

Codifica dei numeri interi

- Nell' aritmetica binaria vi sono 4 tecniche di organizzazione dei numeri interi con segno.
  - Modulo e segno;
  - Complemento alla base diminuita;
  - Complemento alla base;
  - Eccesso M (o bias);

- **Modulo e segno**: un bit per il segno e gli altri per il modulo
- **Complemento alla base**: $C(N_b) = b^n - N$ ove n è il numero di cifre, b la base (per es. 2) e N il numero da rappresentare.
- **Metodo del bias**: Si pone lo 0 a metà della scala. Per es. se ho 8 bit, la scala va da 0x0 a 0xFF; 0 rappresenterà -127 e 0xFF +127.

Memoria Cache

- Sono memorie molto veloci (e costose) che si trovano sullo stesso chip della CPU
- Quando una parola di memoria viene utilizzata dalla CPU, questa e le sue vicine hanno un'alta probabilità di venir riutilizzate a breve tempo e vengono memorizzate nella cache.
- La CPU che deve utilizzare una parola di memoria, prima controlla se non è già in cache:
  - se lo è, preleva la parola e la utilizza
  - Se non è in cache, trasferisce un blocco di parole contenente quella cercata dalla memoria centrale alla cache e poi la utilizza.
- Grande riduzione dei tempi medi di accesso alla memoria

Classificazione delle memorie

- Memorie Volatili: perdono l'informazione quando si spegne l'alimentazione
- Memorie non volatili: mantengono l'informazione anche in assenza di alimentazione
- Memorie a semiconduttore
  - ROM (Read Only Memory)
  - PROM (Programmable ROM)
  - EPROM (Erasable Prom)
  - EEPROM (Electrically Erasable PROM)
  - FLASH (read-most non-volatile memory)
  - RAM (Random Access Memory, Read/Write)
    - SRAM (Static RAM)
    - DRAM (Dynamic RAM)

Memoria virtuale

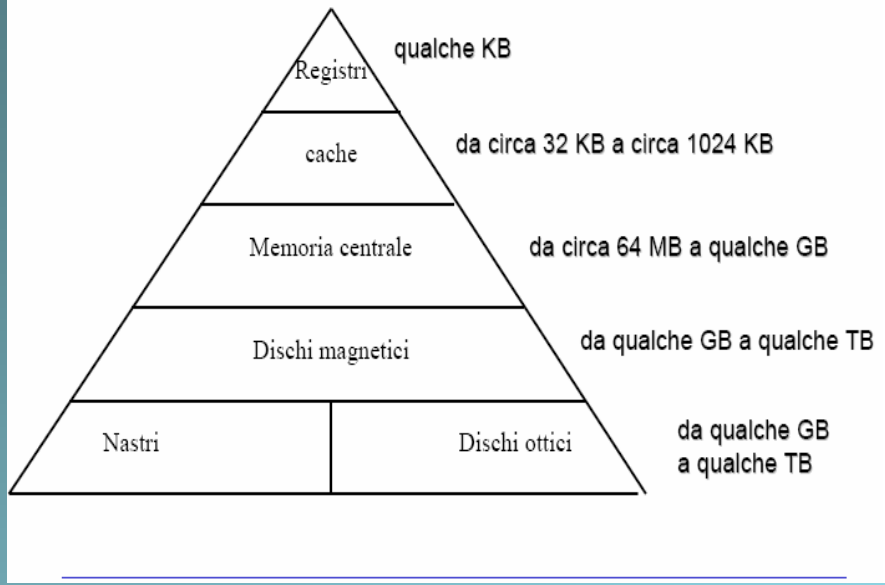
- L’esigenza è quella di eseguire programmi più grandi della memoria fisica disponibile, o gestire più processi.
- La soluzione è uno spazio di indirizzamento *logico* più grande di quello fisico, organizzato in *pagine*.
- Le pagine di memoria logica vengono mappate sulla memoria fisica (RAM) quando utilizzate e riposte in memoria di massa quando non utilizzate (swapping).
- La funzione che trasforma gli indirizzi logici in indirizzi fisici è assolta dalla Memory Management Unit (MMU)
- La MMU può essere a monte (cache fisica) o in parallelo alla cache (cache virtuale).
- La traduzione tra indirizzi logici e indirizzi fisici è rappresentata in una tabella delle pagine (page-map table – PMT)

I registri della CPU

- I registri sono elementi di memoria molto particolari che non risiedono nella memoria centrale.
- In genere sono implementati direttamente nella CPU
- I processori della famiglia x86 hanno 4 registri principali, tutti a 16 bit:
  - L'accumulatore
  - Il Base Address register
  - Il Count Register
  - Il Data Register

  Ed un registro invisibile ai programmatori che è il program counter e i registri *flag* che contengono il risultato dei confronti.
- L'accesso ai registri è estremamente veloce (meno di un clock tick)

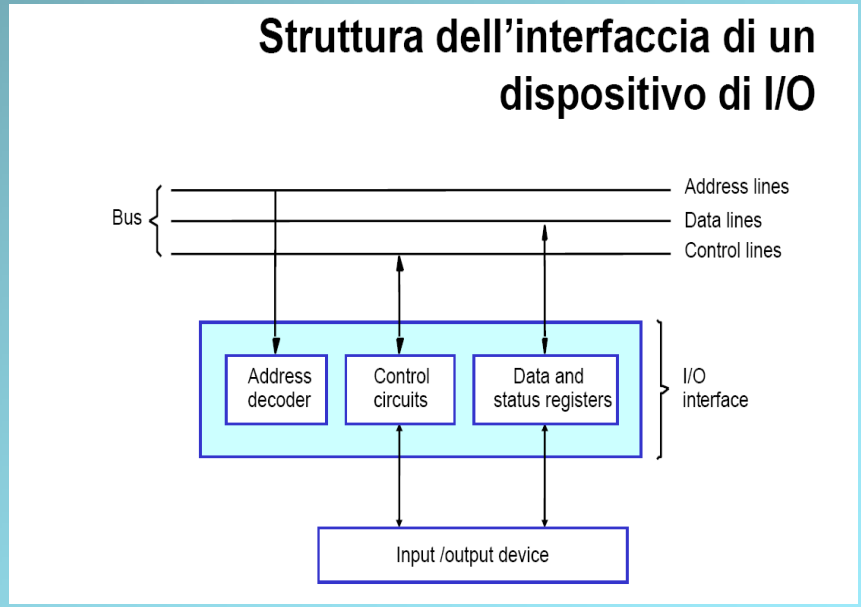
Gerarchia di memorie e tempi di accesso



	Capacità	Velocità (TA)
registri	~1 KB	~1 ns
cache	64 ÷ 1024 KB	~10 ns
RAM	64 ÷ 2048 MB	~100 ns
HD	8 ÷ 100 GB	~10 ms
nastri/CD	~GB per unità	~100 ms

Input/Output (I/O)

- I dispositivi connessi ad un bus, compresa la CPU devono avere
  - Una interfaccia di accesso al bus
  - Un protocollo per usarlo correttamente
- L'indirizzamento dei diversi dispositivi può essere integrato (memory mapped I/O) o separato per ogni dispositivo



Indirizzi di I/O

- Oltre alle line di indirizzo di memoria, alcuni processori (per es. X86) mettono a disposizione un bus di indirizzi per i dispositivi di I/O, in genere più limitato.
- Questo crea due spazi di indirizzamento separati per la memoria e per l'I/O. Le linee di controllo differenziano fra I diversi tipi di indirizzamento
- Per l'accesso alle linee di I/O I processori X86 usano istruzioni particolari.

Controllo dell'I/O

L'I/O può essere :

- controllato da programma (polling)
  - Inefficiente
  - Tiene il bus impegnato continuamente per il controllo
  - Non gestisce la concorrenza
- Controllato da interrupt
- Ad accesso diretto (DMA – Direct Memory Access)

I segnali di Interrupt

- Sono segnali, spesso propagati su linee dedicate dei bus, che istruiscono la CPU ad interrompere l'azione che sta eseguendo ed iniziare un'azione differente (Interrupt Service Routine – ISR), alla fine della quale riprende ad eseguire l'operazione precedente, dal punto in cui si era interrotta.
- L'azione associata a ciascun segnale possibile di interrupt è definita in una serie di registri che contengono l'indirizzo in memoria dell'inizio della sequenza di operazioni da eseguire (interrupt table).
- Ciascuna linea di interrupt si chiama **livello.**
- Alcuni livelli di interrupt della CPU sono dedicati e non assegnabili
- A seconda dei processori e/o di come li si configura, l'azione eseguita a seguito di un interrupt può essere a sua volta interrotta oppure no. Nel primo caso si parla di interrupt *annidiati* (nested interrupt).
- Ai livelli di interrupt può essere assegnata una priorità (memorizzata in un apposito registro) che stabilisce l'accodamento o la possibilità di annidiare l'operazione.

Interrupt – Collegamento dei dispositivi

I dispositivi possono essere collegati:

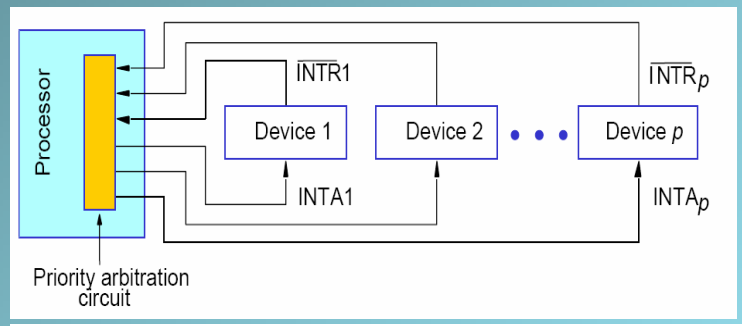
- con linee individuali
  - Facile gestione delle priorità
  - Esplosione del numero delle linee

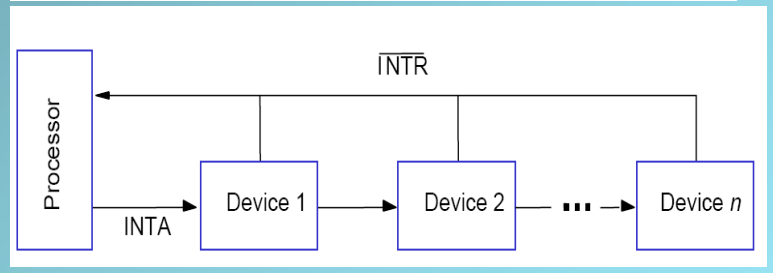
•Con una sola linea in daisy chain

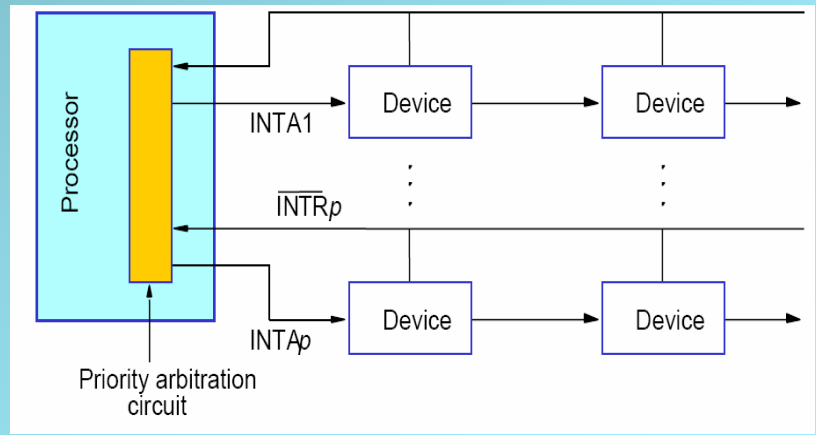
–Difficile gestione delle priorità

–Arbitraggio rigido

•Con linee condivise ma separate per priorità







Direct Memory Access (DMA)

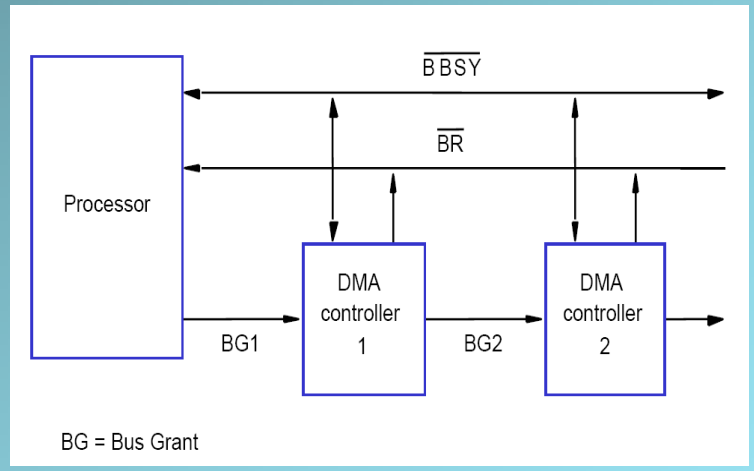
- Trasferimento dei dati tra i dispositivi senza l'intervento del processor
- Necessita di un'unità preposta – il DMA controller - che, prima di accedere al bus per il trasferimento:
  - Controlla lo stato del bus
  - Riceve l'indirizzo di partenza e il numero di parole da trasferire.
  - Imposta la direzione del trasferimento
- Le periferiche di DMA hanno priorità nell'accesso in memoria rispetto al processore
- Possono agire in:
  - Burst mode: negoziano col processore il tempo necessario a trasferire grandi quantità di dati
  - Cycle stealing: "rubano" cicli di accesso al processore per trasferire pochi dati.

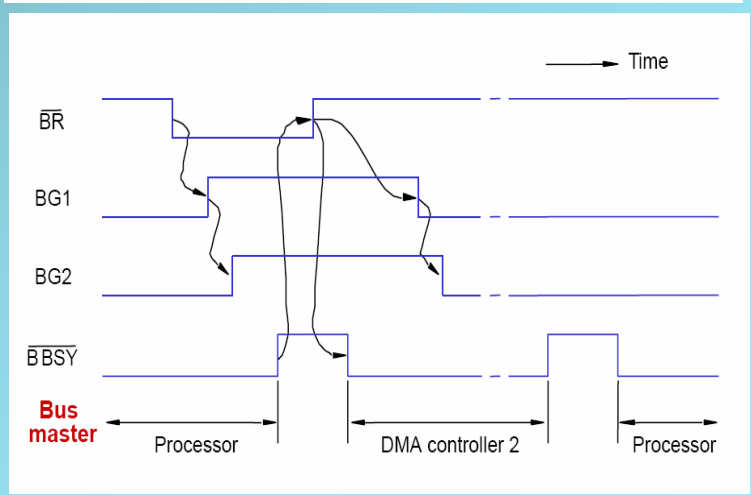
Arbitraggio del bus

- È necessario per risolvere le contese tra quale dispositivo abbia il controllo del bus (bus mastership)
- Centralizzato: semplice ma richiede l'uso di un dispositivo ad hoc o il processore (poco efficiente) – se l'arbitro non è disponibile si blocca tutto (deadlock)
- Distribuito: più robusto ma pi complicato

Arbitraggio centralizzato

- Il dispositivo 2 richiede il bus – bus Request BR.
- Il processore attiva le linee di Bus Grant
- Il dispositivo 2 diventa master del bus e attiva la linea di Bus Busy (BBSY)





Arbitraggio distribuito (4 fili – 16 indirizzi)

- Tutti scrivono il proprio indirizzo
- A partire dal bit più significativo, se riscontrano una differenza, mettono a 0 il bit corrispondente e tutti quelli a priorità inferiore
- Il valore sui fili cambia per un numero di cicli al massimo pari al numero di fili, poi si stabilizza all'indirizzo del vincente che riconosce il proprio indirizzo e acquisisce la mastership.