

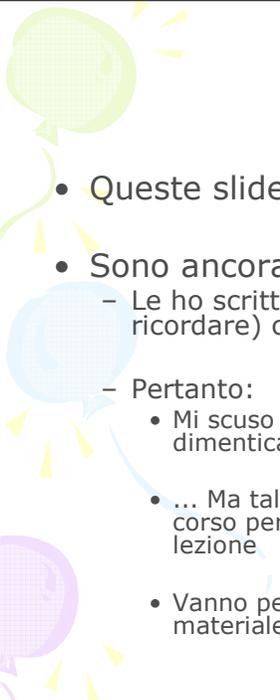


Informatica per Ingegneria Gestionale

versione a.a. 2006-2007

Pierfrancesco Foglia

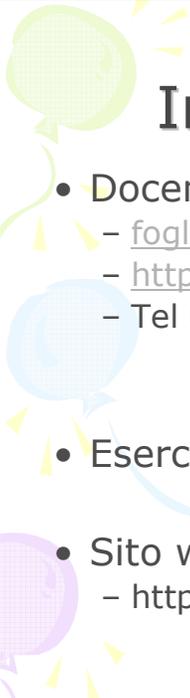
P. Foglia 1



Osservazione

- Queste slide **non** sono state proiettate in aula
- Sono ancora "in progress" e non definitive
 - Le ho scritte di getto, cercando di riassumere (e ricordare) ciò che è stato detto a lezione
 - Pertanto:
 - Mi scuso sin da ora per eventuali imprecisioni e dimenticanze
 - ... Ma tali slide possono essere usate dagli studenti del corso per avere una traccia degli argomenti trattati a lezione
 - Vanno però integrate con il libro di consultazione e il materiale reperibile sul sito del corso

P. Foglia 2



Informazioni Generali

- Docente: Ing. Pierfrancesco Foglia
 - foglia@iet.unipi.it
 - <http://garga.iet.unipi.it>
 - Tel 050-2217530
- Esercitatore: Ing. Tommaso Cucinotta
- Sito web del corso:
 - <http://garga.iet.unipi.it/didattica/>

P. Foglia 3



Materiale Didattico

- Luca Cabibbo: Fondamenti di Informatica, Oggetti e Java – McGraw-Hill
- Materiale disponibile sul sito
- Di consultazione:
 - Sciuto et al.: Introduzione ai Sistemi Informatici, MCGraw-Hill
 - Un qualunque testo specifico su Java (per approfondimenti) quale Horstmann o Deitel

P. Foglia 4



Argomenti del Corso

- L'informatica, gli algoritmi ed il calcolatore
 - Cos'è l'informatica ed a cosa serve
- L'architettura dei calcolatori
 - Acquisire le nozioni di base sull'hardware, sul software e sul sistema operativo
- La rappresentazione e la codifica delle informazioni
 - Comprendere come l'informazione viene rappresentata all'interno del calcolatore
- La programmazione in Java
 - Imparare i fondamenti dei linguaggi di programmazione
 - Imparare i fondamenti del linguaggio Java
 - Imparare le tecniche di base della programmazione e della programmazione OO
 - Imparare a risolvere problemi tramite algoritmi
 - Imparare a implementare gli algoritmi tramite un linguaggio di programmazione
- Gli strumenti di produttività individuale

P. Foglia

5



l'informatica, gli algoritmi ed il calcolatore

P. Foglia

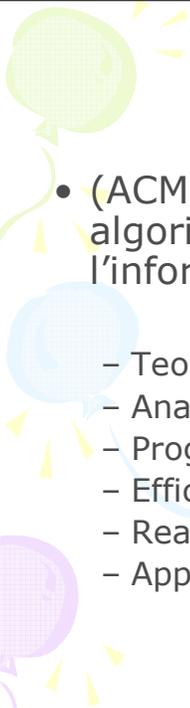
P. Foglia

6



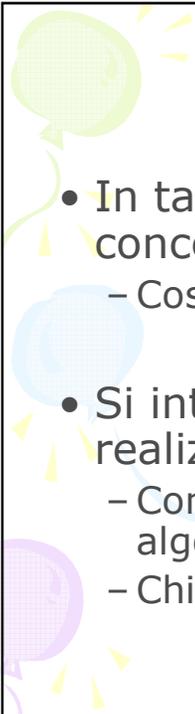
Definizione/I

- L'informatica è la **scienza** della **rappresentazione** e dell'**elaborazione** dell'**informazione**
 - **Scienza**: approccio rigoroso e sistematico
 - **Informazione**: è parte di qualsiasi attività umana
 - **Rappresentazione**: il problema di astrarre i concetti importanti per modellare la realtà di interesse
 - Nel nostro caso studieremo i metodi di rappresentazione appropriati all'elaborazione da parte di una macchina digitale
 - **Elaborazione**: uso e trasformazione dell'informazione in maniera funzionale al raggiungimento di obiettivi
- Al centro dell'informatica vi sono i concetti di
 - informazione
 - i dati del problema e come vengono rappresentati
 - elaborazione
 - la trasformazione per il raggiungimento di un obiettivo
- L'importanza delle informazioni nella nostra vita evidenzia l'importanza dell'informatica



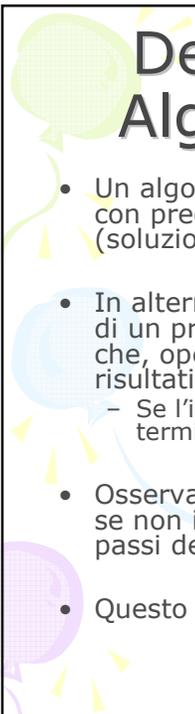
Definizione/II

- (ACM): E' lo studio sistematico degli algoritmi che descrivono e trasformano l'informazione
 - Teoria
 - Analisi
 - Progetto
 - Efficienza
 - Realizzazione
 - Applicazione



osservazioni

- In tale definizione si introduce il concetto di algoritmo
 - Cos'è un algoritmo?
- Si introduce il concetto di realizzazione/applicazione:
 - Come si realizza/implementa un algoritmo?
 - Chi lo applica/implementa?



Definizione informale di Algoritmo e di Esecutore

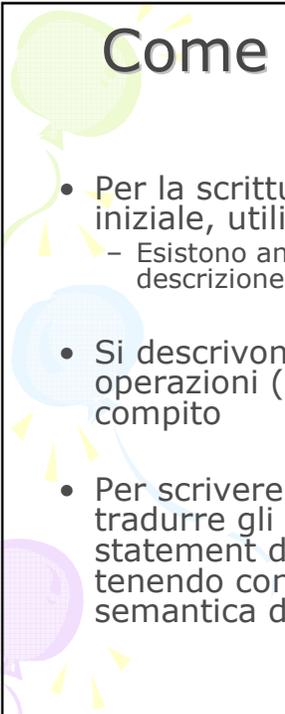
- Un algoritmo è una sequenza finita di passi/azioni, definiti con precisione, che portano alla realizzazione di un compito (soluzione di un problema)
- In alternativa: un algoritmo è la descrizione della soluzione di un problema in termini di un insieme di regole (passi) che, operando sui dati iniziali, consentono di ottenere dei risultati che costituiscono la soluzione del problema
 - Se l'insieme dei dati di ingresso è finito, l'algoritmo deve terminare dopo un numero finito di passi
- Osservazione: descrivere un algoritmo non ha molto senso, se non in relazione a qualcuno/qualcosa che, eseguendo i passi dell'algoritmo, risolverà il problema
- Questo qualcuno/qualcosa prende il nome di **ESECUTORE**

Proprietà dell'esecutore

- Un esecutore è definito in base a tre elementi:
 - L'insieme delle operazioni che è capace di compiere
 - Univocamente definito
 - Deterministico (porta allo stesso risultato)
 - L'insieme delle istruzioni che comprende/interpreta
 - Definite in modo formale (sintassi)
 - Costituiscono un linguaggio
 - Quali operazioni associare ad ogni istruzione
 - L'insieme delle regole che ad ogni costrutto sintatticamente corretto associano le relative operazioni da compiere deve essere univocamente definito
 - Caratterizzazione semantica

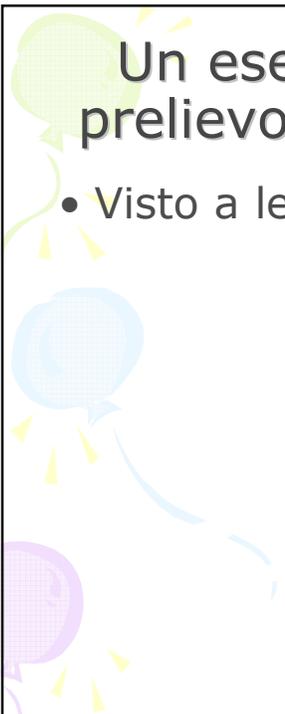
Il calcolatore, i linguaggi di programmazione ed i programmi

- **Il calcolatore** è un particolare tipo di esecutore
- **I linguaggi di programmazione** sono i linguaggi che il calcolatore è in grado "di comprendere"
 - Ossia è in grado di tradurre in operazioni
 - In realtà è il linguaggio macchina o assembly l'unico linguaggio che il calcolatore è in grado di comprendere
- Se un algoritmo è descritto in modo da poter essere eseguito da un calcolatore, ossia tramite un linguaggio di programmazione, l'algoritmo prende il nome di **programma**



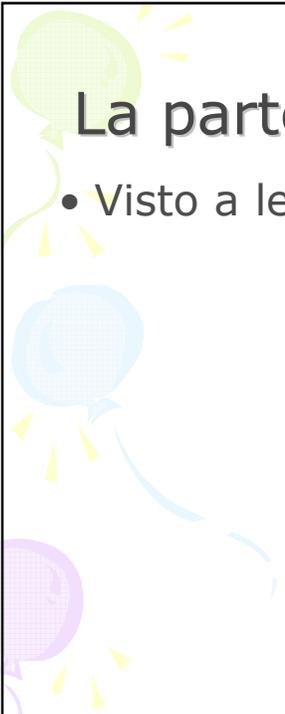
Come scrivere algoritmi e programmi

- Per la scrittura di algoritmi si può, in una fase iniziale, utilizzare il linguaggio naturale
 - Esistono anche dei linguaggi appositi per la loro descrizione
- Si descrivono “semplicemente” le sequenza di operazioni (o step) da eseguire per realizzare un compito
- Per scrivere il relativo programma, occorrerà tradurre gli “step” dal linguaggio naturale in statement del linguaggio di programmazione, tenendo conto sia della sintassi che della semantica del linguaggio



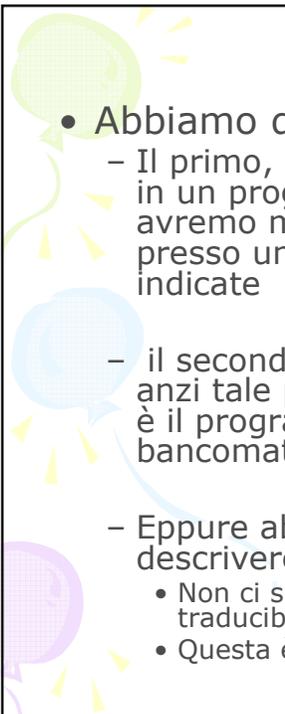
Un esempio di algoritmo: il prelievo dei soldi al Bancomat

- Visto a lezione



La parte duale: il bancomat

- Visto a lezione



osservazioni

- Abbiamo descritto due algoritmi ma
 - Il primo, realisticamente, non si tradurrà mai in un programma, almeno fintanto che non avremo macchine in grado di andare da sole presso un bancomat ed eseguire le operazioni indicate
 - il secondo si può tradurre in un programma, anzi tale programma è stato già scritto, perché è il programma che fa funzionare lo sportello bancomat
 - Eppure abbiamo fatto lo stesso sforzo nel descrivere i due algoritmi
 - Non ci siamo occupati del fatto che fossero o meno traducibili in un linguaggio di programmazione
 - Questa è la fase "iniziale" di scrittura di un algoritmo



Un esempio di architettura hw per implementare l'algoritmo "duale" del bancomat

- Vista a lezione



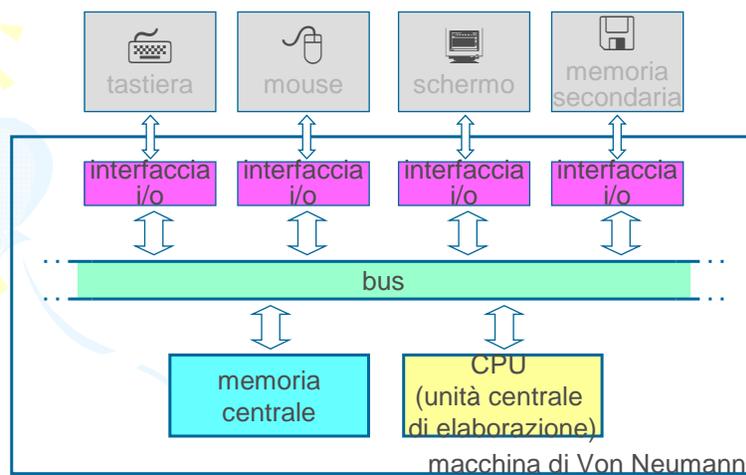
L'architettura del calcolatore

P. Foglia

Il calcolatore

- Adotta l'architettura di Von Neumann

L'architettura di Von Neumann

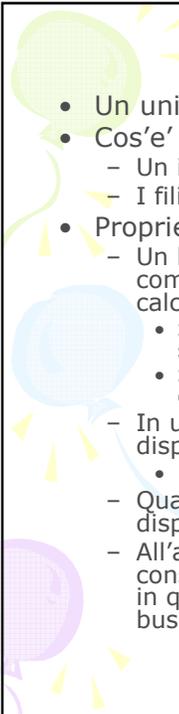


Elementi dell'architettura

- La CPU
 - E' l'esecutore
 - esegue gli step dell'algoritmo/programma coordinando la comunicazione fra i vari componenti
- La memoria principale
 - Contiene le istruzioni ed i dati che costituiscono un programma
 - È volatile
- Il bus
 - E' un insieme di fili che permette lo scambio di dati fra i vari elementi del sistema
- l'insieme dei dispositivi di I/O
 - Sono indispensabili, in quanto consentono alla CPU di dialogare con il mondo esterno,
 - Consentono di acquisire dati dal mondo esterno
 - Rendono "visibili" i risultati dell'elaborazione
- Nell'insieme è un **sistema**: insieme di componenti interagenti fra di loro

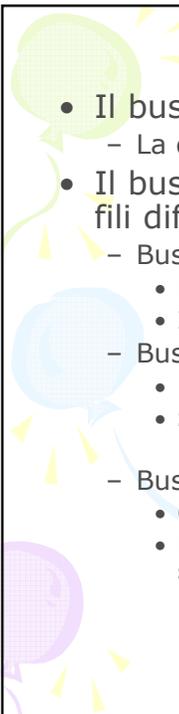
Caratteristiche

- Venne sviluppata negli anni 50 ma è ancora attuale
- È l'architettura del calcolatore sequenziale
- Dati e istruzioni sono memorizzati in una memoria unica che permette sia la lettura che la scrittura
- I contenuti della memoria sono indirizzati in base alla loro posizione, indipendentemente dal tipo di dato o istruzione contenuto
- Le istruzioni vengono eseguite in modo sequenziale



Caratteristiche: il bus

- Un unico bus connette tutti i dispositivi
- Cos'è un bus:
 - Un insieme di fili
 - I fili hanno differenti significati
- Proprietà e limiti del bus
 - Un bus è il modo più semplice di collegare (e di far comunicare) l'insieme dei dispositivi che costituiscono il calcolatore
 - Se si utilizzassero collegamenti punto-punto, quanti collegamenti sarebbero necessari?
 - Se si aggiunge una periferica (espansione) in un sistema a collegamenti punto-punto, come si gestisce l'espansione?
 - In un sistema a bus si possono aggiungere semplicemente più dispositivi
 - Basta connetterli
 - Quando il bus è impegnato in una comunicazione fra 2 dispositivi, nessun altro dispositivo può comunicare (limite)
 - All'aumentare del numero dei dispositivi collegati, in conseguenza della proprietà precedente il bus non è efficiente, in quanto per la maggior parte del tempo i dispositivi trovano il bus occupato



I fili del bus

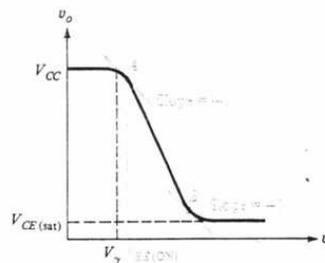
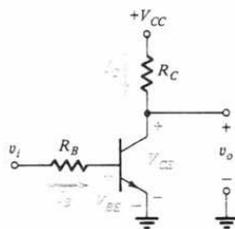
- Il bus può essere mono o bidirezionale
 - La direzione è relativa al verso in cui viaggiano i dati
- Il bus è costituito da 3 bus differenti (3 insiemi di fili differenti)
 - Bus degli indirizzi
 - Monodirezionale
 - Individua la risorsa su cui si vuole operare
 - Bus dei comandi
 - monodirezionale
 - Specifica il tipo di operazione che si vuole effettuare
 - Tipicamente lettura/scrittura + altre informazioni
 - Bus dei dati
 - Contiene i dati che vengono scambiati
 - È tipicamente bidirezionale, in base al tipo di operazione specificato dal bus dei comandi

il calcolatore è digitale

- Il calcolatore è un insieme di circuiti elettronici
- Tali circuiti ricevono in ingresso e producono in uscita 2 soli livelli di tensione (codificati come 0 e 1)
 - Definizione di circuito digitale
- Tutti i segnali che "viaggiano" nel sistema calcolatore, ed i dati immagazzinati in memoria sono digitali, ossia sono sequenze di 0 e 1
- Anche il linguaggio che comprende il calcolatore, ossia l'assembler, è una sequenza di 0 e 1

Perché è digitale/I?

- Per motivi tecnologici:
 - E' stato possibile costruire dei circuiti elettronici digitali
 - Ossia possono trovarsi in uno di due stati possibili
 - Esempio: invertitore a transistor BJT



Perché è digitale/II?

- Mettendo insieme numerosi circuiti di tale tipo è stato possibile costruire microprocessori e memorie, ossia la macchina di Von Neumann
- Tali circuiti sono diventati sempre più veloci, affidabili, economici e piccoli
- Allora, è stato possibile assemblarli in quantità sempre maggiore, ottenendo
 - capacità di memorizzazione crescenti
 - capacità di elaborazione crescente
 - Costi decrescenti
 - Spazi contenuti
- Ossia microprocessori, memorie, dispositivi di I/O sempre più potenti e capaci, in grado di risolvere, nell'insieme, problemi sempre più complessi e, grazie al loro costo, di essere sempre più diffusi

Esempi CPU/freq/transistor

CPU	Anno	Freq. (Mhz)	Transistor
8086	1978	4.77-12	29.000
80486	1989	33-50	1.200.000
Pentium	1993	60-200	3.100.000
PentiumII	1997	233-400	7.500.000
PentiumIV	2000	1.600+	42.000.000

I linguaggi di basso e alto livello

- il linguaggio che il calcolatore è in grado di comprendere è fatto da 0 e 1
- Eppure si programma in Java:
 - Perché? Perché è più facile da comprendere per gli essere umani, ossia è più facile descrivere gli algoritmi in Java
 - Come fa il calcolatore ad eseguire un programma Java?
 - Il calcolatore non può eseguire programmi scritti in Java, ma solo programmi scritti in linguaggio assembler
 - Essendo sia la sintassi che la semantica di ogni linguaggio di programmazione definita in modo formale, è possibile tradurre, in modo automatico, i programmi scritti in Java in programmi scritti in Assembler
- I linguaggi di programmazione si dividono in linguaggi di basso e alto livello
 - Basso livello: sono più vicini alla macchina (l'assembler, ad esempio)
 - Alto livello: sono più vicini al linguaggio dell'utente (Java, ad esempio)

P. Foglia

29

Struttura del calcolatore e le CPU

P. Foglia

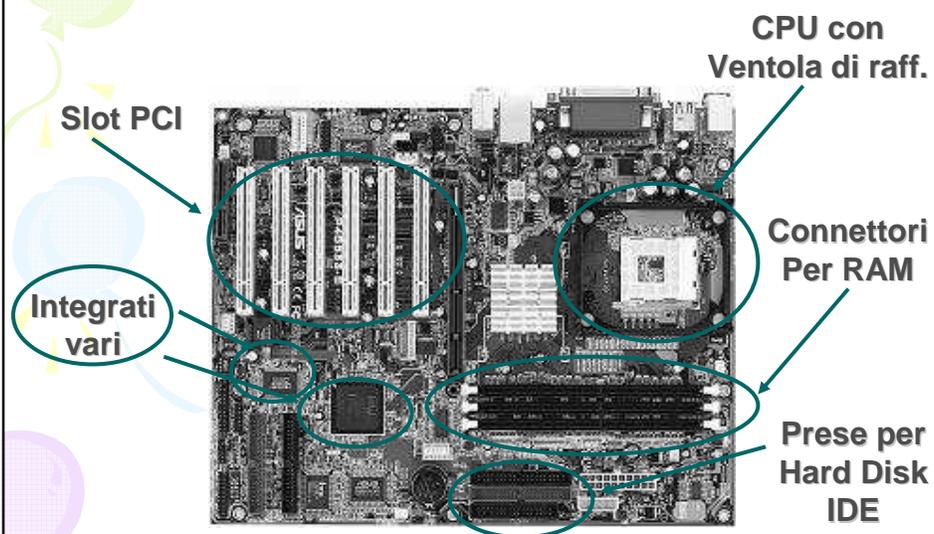
P. Foglia

30

Struttura fisica di un Calcolatore

- Vediamo come sono disposti gli elementi principali di un calcolatore all'interno di un PC
 - Senza essere esaustivi nelle tecnologie
 - E facendo riferimento all'architettura di Von Neumann
- Nel case è contenuta la scheda madre
- La scheda madre contiene:
 - La CPU
 - Connettori per inserire i moduli di memoria
 - Vari chip di supporto
 - Prese in cui si possono inserire i connettori delle schede di I/O che agiscono da controllori dei dispositivi di I/O
 - Vari bus

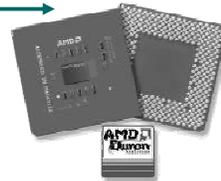
La mainboard



esempio

CPU e pin

Ventola di raffreddamento

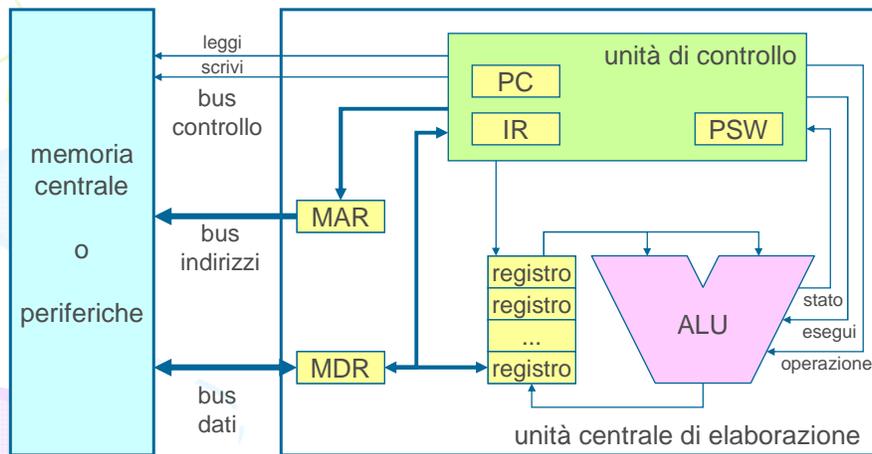


Socket

Le cpu

- sono caratterizzate da:
 - Frequenza di clock
 - È il segnale che fa "progredire" l'esecuzione del processore
 - Valori tipici attuali per processori General Purpose sono sull'ordine del Ghz
 - Set di Istruzioni, ossia Linguaggio Macchina
 - Architettura Interna, ossia
 - Numero e tipo di ALU, numero dei registri, ampiezze dei bus dati e indirizzi
 - Esempi di CPU: Pentium, ARM7, SPARC, etc.

Architettura semplificata di una CPU



P. Foglia

35

Funzionamento dell'unità centrale di elaborazione

- L'unità centrale di elaborazione esegue un programma, istruzione per istruzione, svolgendo ripetutamente le seguenti tre operazioni
 - fetch
 - legge dalla memoria la prossima istruzione da eseguire
 - L'indirizzo di tale locazione è contenuta nel PC (program Counter), registro (ossia memoria) contenuta all'interno del processore
 - decode
 - determina il tipo di istruzione che deve essere eseguito
 - execute
 - richiede lo svolgimento di tutte le azioni necessarie per l'esecuzione dell'istruzione

P. Foglia

36

Il ciclo fetch/decode/execute

1. Viene letto il PC
2. Si preleva dalla memoria l'istruzione contenuta nel PC
3. Si incrementa PC di 1
4. Si decodifica l'istruzione, per comprendere l'insieme di operazioni che la CPU deve eseguire in corrispondenza della istruzione letta
5. Si esegue l'istruzione
6. Si salta al punto 1
 - Ossia si legge l'istruzione successiva

commenti

- Il processore esegue le istruzioni in sequenza, una dopo l'altra
 - Alcune istruzioni particolari, quelle di salto, alterano tale esecuzione sequenziale
- All'accensione, il PC viene inizializzato ad un valore prestabilito
- In tale area viene memorizzato il programma che deve essere eseguito all'avvio del computer
 - Tipicamente il BIOS, che poi avvia il sistema operativo

La memoria principale

P. Foglia

P. Foglia

39

La memoria principale

- È utilizzata per memorizzare i dati ed i programmi
- E' organizzata in sequenze di bit, di nome celle, variabile al variare del tipo di CPU, accessibili tramite un indirizzo
- Poiché il processore accede alla memoria principale ogni qualvolta deve leggere un dato, è fondamentale che sia veloce, altrimenti rende inutile la velocità del processore
- Dimensioni tipiche delle RAM sui PC attuali vanno dai 128M byte a 1G byte

P. Foglia

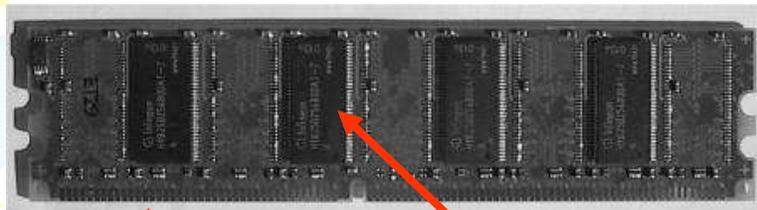
40

proprietà

- La memoria principale è detta anche RAM, ed è accessibile sia in scrittura che in lettura
- E' di natura elettronica, per cui è veloce ma generalmente costosa e volatile (rispetto alle memorie magnetiche ed ottiche)
- È volatile, ossia perde il contenuto quando non è alimentata
 - Esistono anche le ROM, memorie non volatili accessibili in sola lettura
 - Esistono le FLASH, che sono non volatili come le ROM, scrivibili, ma un numero limitato di volte
- E' caratterizzata dal Tempo di accesso, ossia il tempo che intercorre fra la richiesta di un dato e la sua ricezione da parte della CPU
- La sua dimensione può essere variata in un calcolatore, per aumentarne le prestazioni
 - Per questo sulla main-board sono presenti più slot
- Può essere Statica o dinamica
 - sono due differenti tecniche di realizzazione della RAM. La seconda è più lenta ma più economica.
 - La memoria centrale dei PC è realizzata esclusivamente con memorie dinamiche, per motivi di costo

41

Un esempio di RAM



Connettori per il bus

Chip di Memoria DDR 32 MB

P. Foglia

42

Ram statiche e dinamiche: funzionamento

- Ram dinamiche:
 - Lo stato della cella è rappresentato dalla carica di un condensatore
 - Poiché un condensatore si scarica, la sua carica deve essere rinfrescata, ossia riportata a quella iniziale
 - Sono lente ma piccole
- Statiche:
 - Sono degli interruttori, che vengono messi a 0 (massa) o 1 (Vdd) sulla base del contenuto della cella
 - Quando vengono lette, si legge il valore della tensione dell'interruttore
 - Gli interruttori si realizzano in modo da commutare molto velocemente ma
 - Sono costose
 - Sono più grandi delle RAM dinamiche (4-6 transistor contro 1 Condensatore)

La memoria cache

- Le memorie cache sono delle RAM statiche, di piccole dimensioni per motivi di costo, poste vicino al processore (integrate nel chip del processore sui processori attuali)
- Hanno tempo di accesso confrontabile a quello del processore o di poco superiori
- Il processore ricerca un dato/istruzione in cache. Solo se non è presente in cache, accede alla RAM, trasportando in RAM non solo il dato richiesto ma anche quelli adiacenti
- Tale meccanismo funziona ed è estremamente efficiente (più del 90% degli accessi avviene in cache), in quanto il processore non fa accessi "disordinati" ma ordinati: nell'esecuzione di un programma accede ad una istruzione dopo l'altra (la macchina di Von Neumann è sequenziale) ed elabora sempre gli stessi dati
 - Ciò costituisce il principio di località

Il principio di località

- È conseguenza della struttura ciclica dei programmi e della tecnica sequenziale di lettura delle istruzioni da parte della CPU
 - È del fatto che tali proprietà hanno anche gli accessi ai dati su cui il programma effettua l'elaborazione
- Principio di località temporale
 - Se all'istante t si accede ad un dato, è molto probabile che in un successivo intervallo Δt tale dato verrà riacceduto
 - In conseguenza di ciò, quando si legge un dato, lo si mette nella cache
- Principio di località spaziale
 - Se all'istante t si accede ad un dato in posizione d , è molto probabile che all'istante successivo si accederà al dato in posizione $d+1$
 - In conseguenza di ciò, quando si legge un dato, si mette in cache non solo il dato, ma anche n dati successivi

Hit/miss

- Quando il processore effettua un accesso in cache, può verificarsi:
 - Una hit, quando il processore trova il dato in cache
 - In tal caso il processore funziona molto velocemente, perché non accede alla RAM
 - Una miss, quando il processore non trova il dato in cache
 - In tal caso, il processore deve accedere alla memoria per prelevare il dato
- Nei sistemi attuali, la percentuale di hit su di un accesso è del 90-98%

Esempio di vantaggio ottenibile con una cache

- In un sistema con 64Kb di cache con tempo di accesso di 1ns, e 1Gb di RAM con tempo di accesso di 100ns, si osserva un hit rate del 98%. Qual è il tempo medio di accesso alla memoria?
 - $T_{mem} = (n_{hit} * t_{hit} + n_{miss} * t_{miss}) / (n_{hit} + n_{miss})$
 - $T_{mem} = (98 * 1 + 2 * 100) / 100ns = 2,98 \text{ ns}$
- Ossia il sistema, nel complesso, si comporta come un sistema avente un'unica memoria da 1GB con velocità di 2,98 ns
 - Quale sarebbe il costo di un sistema avente 1G di tale RAM
 - Per il suo basso tempo di accesso, deve essere necessariamente statica

Le gerarchie di memoria

- Poiché, per tutte le memorie in un calcolatore, la velocità è inversamente proporzionale al prezzo, la memoria (principale e di massa) viene organizzata in una gerarchia
 - registri (max velocità, max costo, min capacità), nella CPU
 - Livello successivo: cache, integrata o accanto alla CPU
 - Anch'essa può essere a più livelli
 - MEMORIA PRINCIPALE
 - DISCO magnetico
 - Disco ottico
 - Nastro (min velocità, min costo, max capacità)
- In tal modo si riesce ad avere un sistema che ha una capacità di memorizzazione pari alla capacità della memoria più lenta (che può essere elevata per i suoi bassi costi), ma prestazioni non di molto lontane da quelle di un sistema con solo la memoria veloce (in virtù del principio di località) e ad un costo contenuto

Caratteristiche dei diversi livelli

	Capacità	Velocità	Eu/MByte
Registri	1KB	1ns	NA
Cache	64K-2MB	1-10ns	300
RAM	64-2048MB	10-100ns	2
HD	8-100 GB	10ms	0.005
Nastri/ CD	GB per Unità	100ms	0.005

P. Foglia

49

Organizzazione della memoria

- La quantità minima di informazione memorizzabile è il bit
 - Bit: binary digit, Può assumere due valori
- Trasferire fra CPU e memoria 1 singolo bit non è conveniente, perché
 - La memoria è più lenta del processore
 - la quantità di informazione di 1 bit è troppo piccola, per cui la CPU dovrebbe effettuare un numero elevato di trasferimenti
- Pertanto i bit di memoria vengono organizzati in **celle o locazioni**, ed in un trasferimento CPU-memoria viene trasferita una cella, ossia più bit
 - tutte le celle sono formate **dallo stesso numero di bit**;
 - una cella composta da k bit, è in grado di contenere una qualunque tra 2^k combinazioni diverse di bit.
 - Non si trasferisce mai una quantità inferiore alla cella
- **Dimensioni della cella**
 - In quasi tutti i calcolatori è di **8 bit** (un **byte**).
 - Altre dimensioni: word (16 bit), double or quad word (32, 64 bit)
 - I byte vengono raggruppati in **parole** (che oggi sono di **32/64 bit**), su cui la CPU esegue le operazioni.

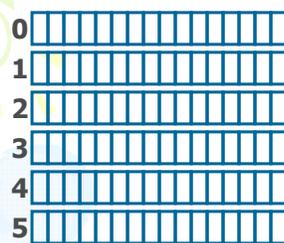
P. Foglia

50

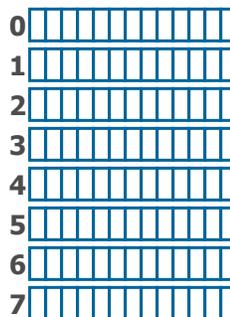
Indirizzi di memoria

- Ogni cella ha un **indirizzo**:
 - serve come accesso all'informazione;
 - in una memoria con **N celle** gli indirizzi vanno da **0** a **N-1**.
- Anche gli indirizzi della memoria sono rappresentati come numeri binari:
 - un indirizzo di **M** bit consente di indirizzare **2^M** celle;
 - per 6 o 8 celle bastano 3 bit, per 12 celle ne servono 4;
 - il **numero di bit nell'indirizzo** determina il **numero massimo di celle indirizzabili** nella memoria ed è indipendente dal numero di bit per cella (una memoria con 2^{12} celle richiede sempre 12 bit di indirizzo, quale che sia la dimensione di una cella).
- Una memoria può essere organizzata in diversi modi:
 - con 96 bit possiamo avere 6 celle di 16 bit ($6 \cdot 16 = 96$), o 8 celle di 12 bit ($8 \cdot 12 = 96$) o 12 celle di 8 bit ($12 \cdot 8 = 96$).

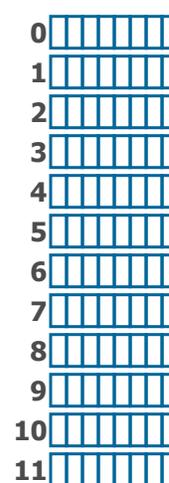
Esempi di possibili organizzazioni



6 parole da 16 bit



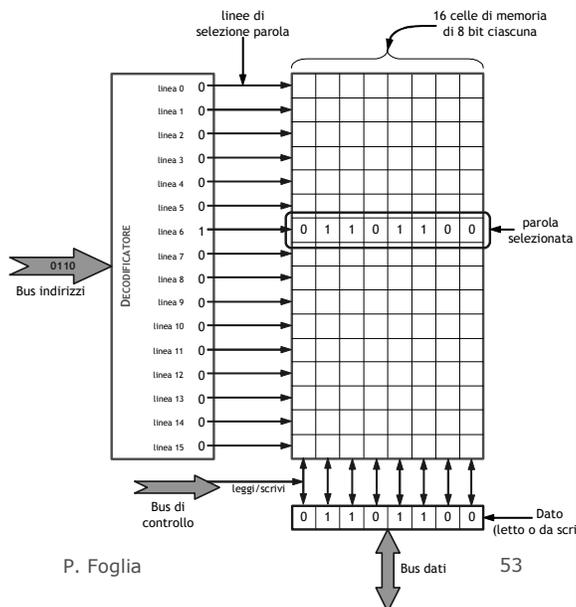
8 parole da 12 bit



12 parole da 8 bit

Memoria: schema logico del circuito

- Un decoder, ricevuti in ingresso gli indirizzi, seleziona la cella opportuna
 - Mettendo a 1 l'uscita a cui è collegata la cella
- Ogni bit della cella di memoria viaggia su di un filo del bus dati
- Tramite il bus di controllo viene specificato il verso del trasferimento:
 - Se lettura dalla memoria alla CPU
 - Se scrittura dalla CPU alla memoria
- Nota: tutta questa logica è inclusa nel package della memoria, da cui escono solo i fili del bus (come PIN, piedini)



Il decodificatore

- E' un circuito digitale che ha n ingressi e 2^n uscite
 - Le uscite sono pari al numero di stati che gli n ingressi possono assumere
- In corrispondenza di ciascun ingresso, mette ad uno una sola delle 2^n uscite, selezionandola
- La rappresentazione binaria della configurazione in ingresso che attiva l'uscita rappresenta "l'indirizzo" della uscita e dunque della cella attivata dall'uscita
- Il comportamento del decodificatore può essere specificato completamente tramite una tabella di verità

Decoder 2X4

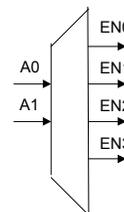


Tabella di verità di un decoder 2X4

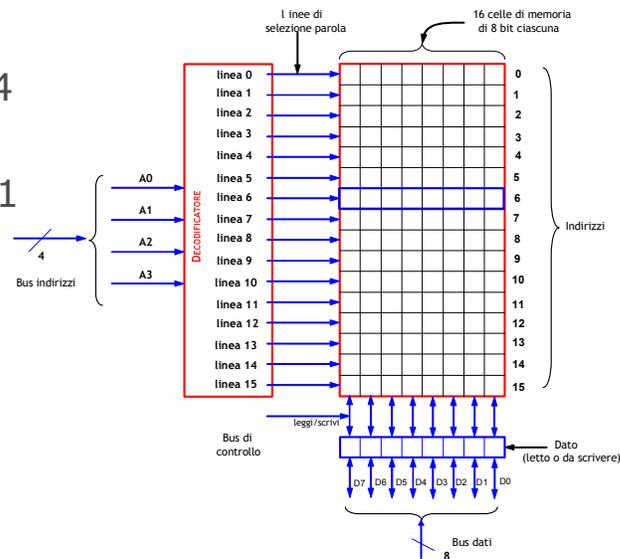
A1	A0	EN0	EN1	EN2	EN3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Dimensioni dei vari bus

- Dim. Bus degli indirizzi: $\log_2 \text{mem-size}$
 - In realtà è l'intero superiore del log precedente
 - Dimensioni "frazionarie" non hanno senso
 - I fili di tale bus si indicano con A_0-A_{n-1} se n è il numero di fili
- Dim bus dei comandi: è indipendente dalla dimensione della ram e dalla cella
 - Nei ns esempi è sempre 1
- Dim. Bus dei dati
 - È pari alla dimensione in bit della cella
 - I fili di tale bus si indicano con d_0-d_{m-1} se m è il numero di fili
 - d_0 prende il nome di bit meno significativo, d_{m-1} bit più significativo

esempio dimensionamento 16X8 bit

- Bus indirizzi: 4
- Bus controllo: 1
- Bus dati: 8



Esempio di funzionamento

- Operazione: lettura della locazione 6

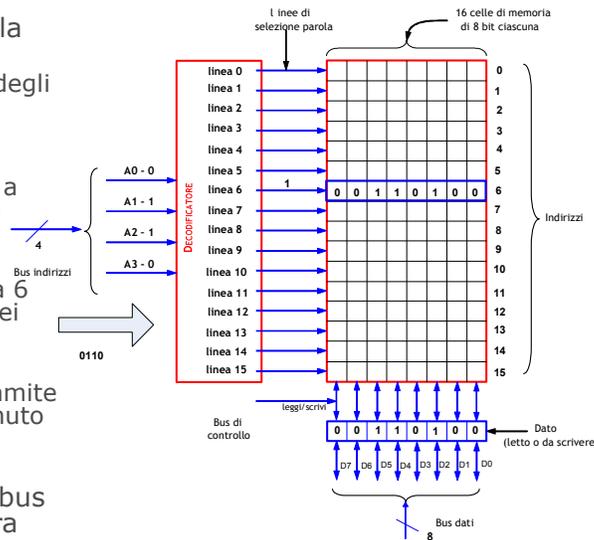
- La cpu prepara il bus degli indirizzi, settandolo a 0110 come in figura

- Il decoder, in risposta a tale ingresso, attiva la linea 6

- Il contenuto della cella 6 viene messo sul bus dei dati dalla memoria

- Il processore legge tramite il bus dei dati il contenuto della cella di memoria

- Dalla preparazione del bus degli indirizzi alla lettura del dato passa del tempo: **e' la velocità della memoria**



Accessi per operandi multipli della cella

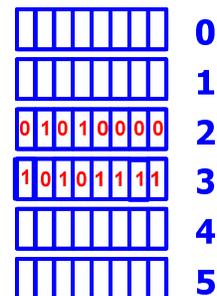
- Le istruzioni del processore in genere accettano degli operandi che hanno come dimensione un multiplo della dimensione della cella di memoria
- L'esecuzione della istruzione comporta allora l'esecuzione di più cicli di lettura/scrittura verso la memoria
- Occorre definire dove e come viene memorizzato un operando "multiplo"
- Ciò viene specificato tramite due algoritmi di lettura e scrittura, che semplificano i casi reali

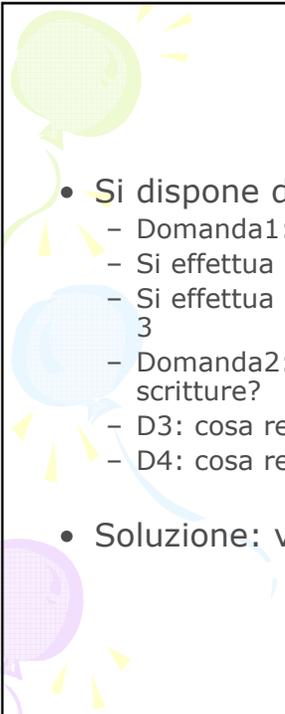
Algoritmo di lettura

- Istruzione da eseguire:
 - Leggi LocA Op
 - Ossia leggi dalla locazione A l'operando Op
- Sia #Op la dimensione dell'operando in bit, B la dimensione della cella di memoria in bit
- Viene calcolato il rapporto n fra la dimensione dell'operando e la dimensione della cella di memoria (espresse in bit)
- Vengono effettuati n accessi consecutivi in memoria, a partire dall'indirizzo A
 - Nel primo vengono trasferiti i B bit della memoria alla locazione A, che costituiscono i B bit meno significativi dell'operando
 - Nel secondo vengono trasferiti i B bit della memoria alla locazione A+1, che costituiscono la parte intermedia dell'operando
 - Etc
 - Ossia la lettura avviene da dx a sx
 - Il primo letto è la parte più a dx dell'operando
 - Il secondo letto è la parte immediatamente a sx dell'operando

esempio

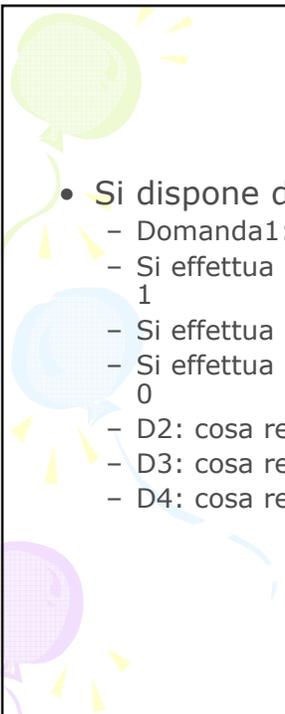
- Leggere 1 word (16 bit) dalla locazione 2, in una memoria organizzata in byte
 - Il contenuto è quello in figura
- Il numero di letture da effettuare è 2
 - La prima legge la locazione 2, per cui la parte bassa dell'operando sarà 0101.0000
 - La seconda legge la locazione 3, per la parte alta dell'operando sarà 1010.1111
- L'operando trasferito vale
1010.1111.0101.0000
- Nel caso di una lettura di un operando di 3 byte dalla locazione 2, supponendo che il contenuto della locazione 4 sia 1111.1111, l'operando letto sarà
1111.1111.1010.1111.0101.0000
- Nota: i puntini vengono letti per semplificare la lettura (e per racchiudere cifre esadecimali)





Esercizio 1

- Si dispone di una RAM 6X8bit, inizializzata a zero
 - Domanda1: Determinare la dimensione dei vari bus
 - Si effettua la scrittura di 00100010 all'indirizzo 1
 - Si effettua la scrittura di 0001000000110011 all'indirizzo 3
 - Domanda2: qual è il contenuto della RAM dopo le due scritture?
 - D3: cosa restituisce la lettura di una word all'indirizzo 0?
 - D4: cosa restituisce la lettura di una word all'indirizzo 4?
- Soluzione: vista a lezione



Esercizio 2

- Si dispone di una RAM 8X8bit, inizializzata a uno
 - Domanda1: Determinare la dimensione dei vari bus
 - Si effettua la scrittura di 0001000000110011 all'indirizzo 1
 - Si effettua la scrittura di 00000000 all'indirizzo 2
 - Si effettua la scrittura di 0000000000000000 all'indirizzo 0
 - D2: cosa restituisce la lettura di un byte all'indirizzo 0?
 - D3: cosa restituisce la lettura di una word all'indirizzo 0?
 - D4: cosa restituisce la lettura di una word all'indirizzo 4?



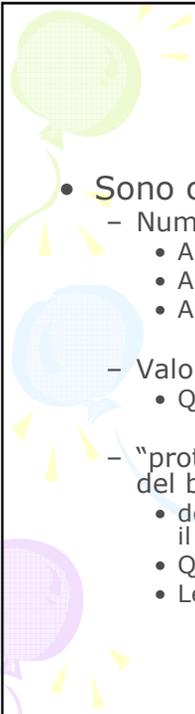
Esercizio 3

- Si dispone di una RAM 12X4bit, inizializzata a zero
 - Domanda1: Determinare la dimensione dei vari bus
 - D2: cosa restituisce la lettura di 4 bit all'indirizzo 11?
 - Si scrive 1111 all'indirizzo 1
 - Si scrive 1100 all'indirizzo 8
 - Si scrive 11110000 all'indirizzo 0
 - D3: cosa restituisce la lettura di 4 bit all'indirizzo 1?
 - D4: cosa restituisce la lettura di 8 bit all'indirizzo 0?
 - D5: cosa restituisce la lettura di 16 bit all'indirizzo 6?
 - D5: cosa restituisce la lettura di 8 bit all'indirizzo 10?



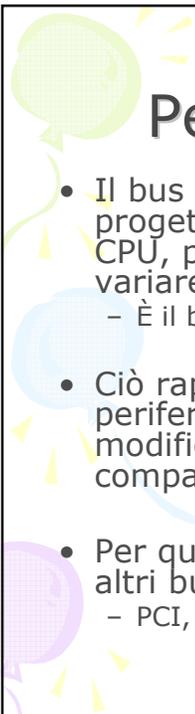
Complementi sui bus

P. Foglia



I bus

- Sono caratterizzati da:
 - Numero e tipo di fili
 - Ampiezza del bus dati
 - Ampiezza del bus indirizzi
 - Ampiezza e tipo del bus di controllo
 - Valori delle tensioni
 - Quale tensione rappresenta lo 0, quale l'uno
 - "protocollo di comunicazione" fra le entità agli estremi del bus
 - dopo quanto tempo dalla preparazione degli indirizzi arriva il dato?
 - Quando devo dare il segnale di read e di write?
 - Le due entità agli estremi del bus devono intendersi



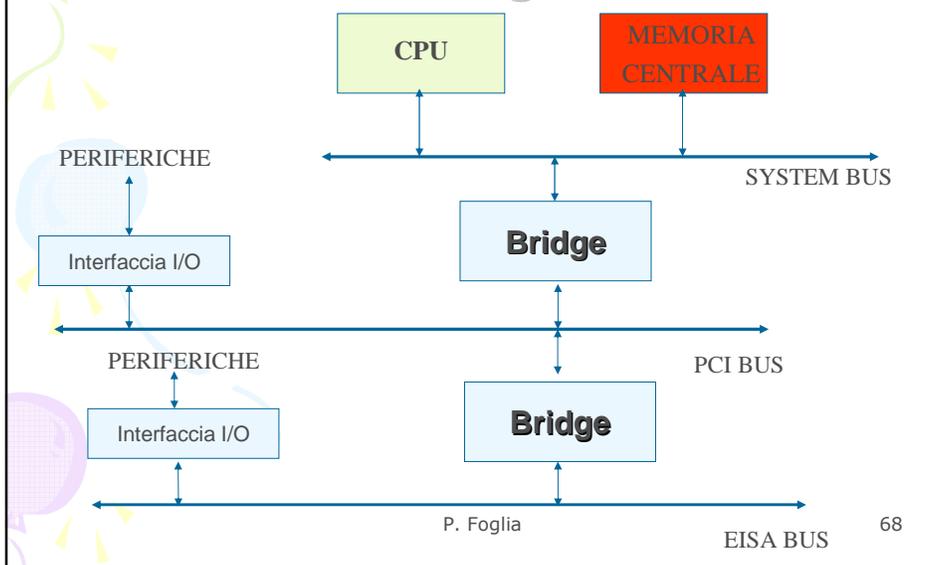
Perché più bus sui PC?

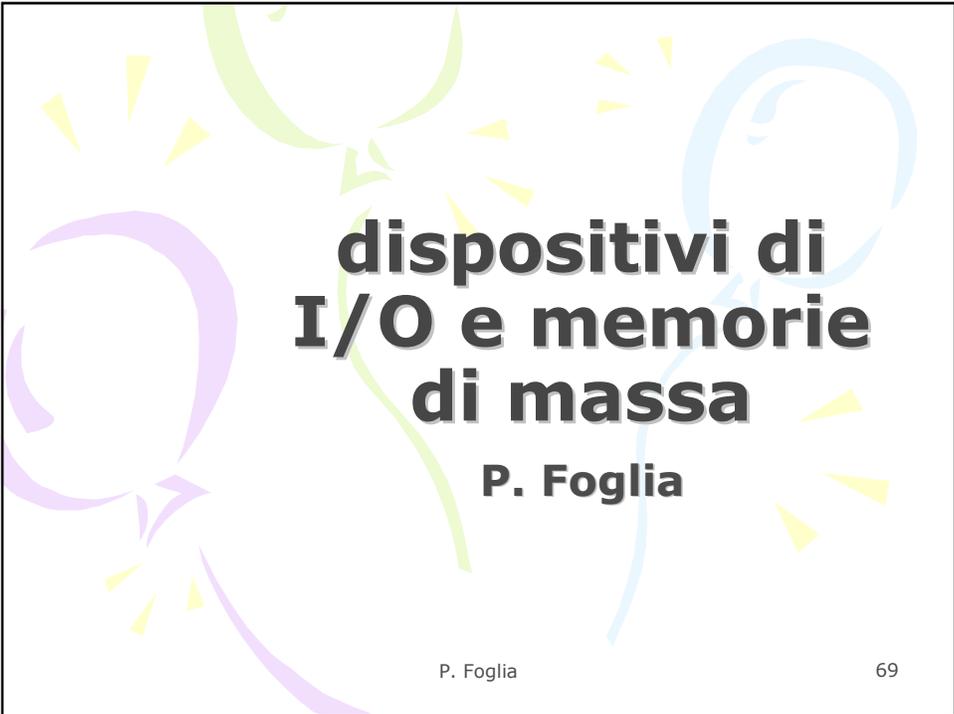
- Il bus principale (system bus in alcuni sistemi) è progettato per ottimizzare le prestazioni della CPU, per cui ha caratteristiche che variano al variare del tipo di CPU
 - È il bus dell'architettura di Von Neumann, per intendersi
- Ciò rappresenta una difficoltà per i costruttori di periferiche (controller), che dovrebbero modificare i propri dispositivi ogni volta che compare una nuova famiglia di processori
- Per questo, accanto al bus di sistema, esistono altri bus standard
 - PCI, EISA, AGP sono esempi di tali bus

I Bridge

- Poiché però i vari bus hanno caratteristiche diverse, occorrono dei componenti che consentano ai vari bus di parlare
- Tali componenti prendono il nome di Bridge
 - Sono alcuni degli integrati presenti sulla main-board

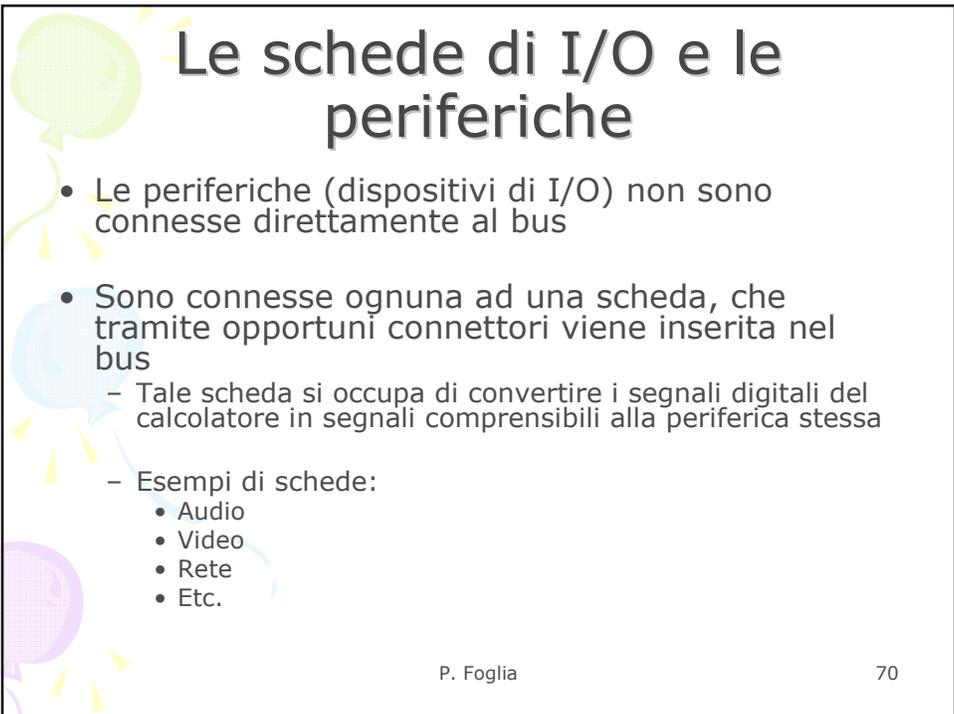
L'architettura con bus e bridge





dispositivi di I/O e memorie di massa

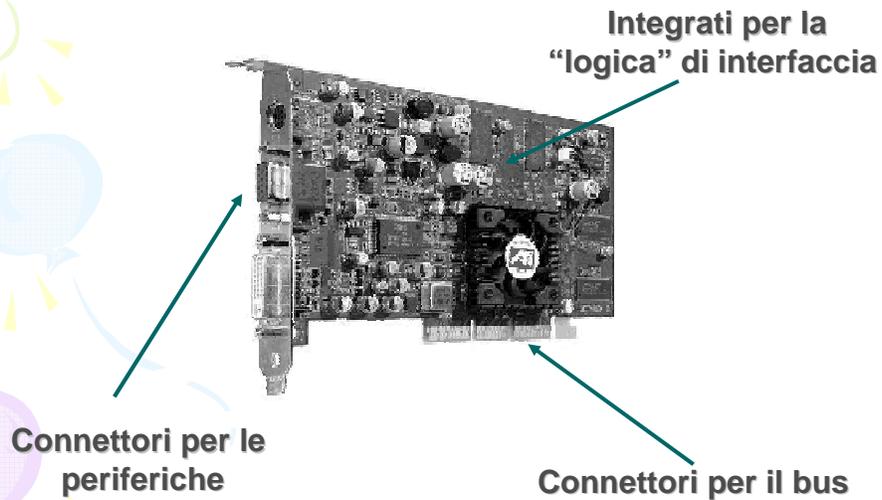
P. Foglia



Le schede di I/O e le periferiche

- Le periferiche (dispositivi di I/O) non sono connesse direttamente al bus
- Sono connesse ognuna ad una scheda, che tramite opportuni connettori viene inserita nel bus
 - Tale scheda si occupa di convertire i segnali digitali del calcolatore in segnali comprensibili alla periferica stessa
 - Esempi di schede:
 - Audio
 - Video
 - Rete
 - Etc.

Un esempio di scheda



P. Foglia

71

Periferiche tipiche di un PC

- Sono sempre presenti in un PC
 - Le memorie di massa
 - La tastiera
 - Il mouse
 - Il video
 - Con relativa scheda di interfaccia
 - La scheda per la connessione della stampante
- Quasi sempre:
 - Scheda audio
 - Scheda di rete
 - modem

P. Foglia

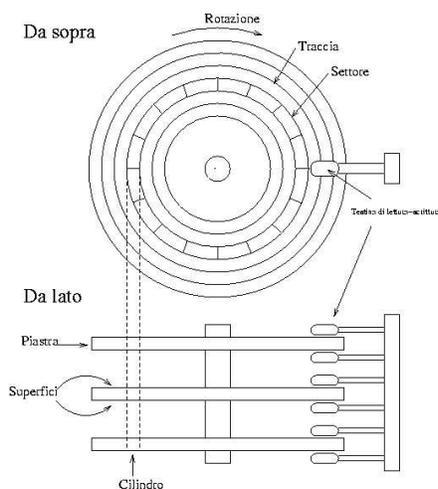
72

Le memorie di massa

- La cpu utilizza la memoria principale per leggere dati e programmi
- È veloce ma è volatile, ossia se ne perde il contenuto quando non è alimentata
- Nel calcolatore è sempre presente un supporto non volatile di memorizzazione (memoria di massa)
 - In esso sono contenuti i programmi ed i dati
 - All'atto dell'avvio di una applicazione, il programma ed i dati vengono trasferiti dal supporto di massa alla memoria principale
 - Quindi la CPU inizia ad eseguire le istruzioni
 - E' il sistema operativo (un particolare programma) che si occupa del trasferimento dalla memoria di massa alla memoria centrale
- Le memorie di massa sono più lente ma più economiche della memoria principale
- Sono di vario tipo
 - L'hard disk, che è la principale, sempre presente in un calcolatore
 - Ma esistono anche altri tipi di supporto:
 - I dischi ottici
 - I nastri magnetici
 - I floppy disk

Il disco rigido (hard disk)

- Sono organizzati in tracce e settori
- Il blocco è la unità di informazione trasferita fra disco e cpu
- Sono di natura elettromeccanica e perciò lenti ma economici





Le prestazioni del disco rigido

- Tempo di accesso (ms)
 - Seek time
 - La testina deve arrivare alla traccia
 - Dipende dalla meccanica (5-15 ms, 1 per tracce adiacenti)
 - Latenza
 - Il disco deve ruotare fino a portare il dato nella posizione giusta
 - Dipende dalla velocità di rotazione (5400-10800 RPM = 2.7 - 5.4 ms)
- Transfer Rate (MBps)
 - Dipende dalla densità di registrazione e dalla velocità di rotazione
 - Un settore di 512 byte richiede da 25 a 100 micros



Come aumentare le prestazioni del disco

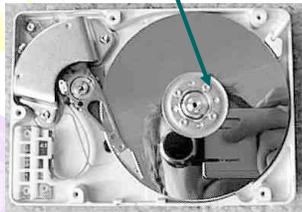
- Le prestazioni di un disco rigido possono limitare significativamente le prestazioni di un sistema di elaborazione
- Per aumentare le prestazioni dei dischi, si ricorre alle seguenti tecniche:
 - aumentare i piatti presenti in un singolo hard-disk (vedi figura precedente)
 - Le testine sono solidali, e vengono letti i settori di più dischi in un'unica operazione
 - Si parla allora anche di cilindri (l'insieme delle tracce lette contemporaneamente)
 - Memorizzare l'informazione su più hard-disk, ossia memorizzare i dati di un file sulle tracce di più dischi, che vengono letti in parallelo
 - Gli hard disk RAID in configurazione Striped funzionano in tal modo

esempio

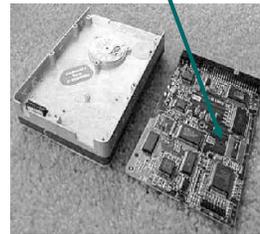
Hard disk (esterno)



Interno - disco



Interno - Elettronica



Dischi Ottici

- La lettura avviene utilizzando la riflessione (o mancata riflessione di un raggio laser)
- La densità di registrazione è più alta dei dischi magnetici
- Utilizzati per registrare audio/tv, in seguito come memoria dei computer
- Diversi tipi/caratteristiche:
 - CD-ROM (a sola lettura, per dati, diversi dagli audio)
 - CD-R (scrivibile)
 - CD-RW (riscrivibile)
 - DVD

CD-ROM e DVD

- Velocità base (1X)
 - 75 settori al sec.
 - 152.6 KByte/sec
- Capacità: 700 MB
- Tempo di accesso: 100 msec
- Il DVD ha maggiore densità
 - 4.7 GByte
 - 1x ha una velocità di trasferimento di 1.4 Mbyte/sec



Nastri magnetici e DAT

- Sono dei supporti di memorizzazione di lettura/scrittura
- Capacità di diversi GigaB
- Accesso sequenziale
- Molto lenti
- Utilizzati solo per il back-up