

Introduzione (cap.1)

Architettura e organizzazione

- Architettura: caratteristiche visibili al programmatore
 - Istruzioni
 - Spazio (numero bit) usato per rappresentare i dati
 - Tecniche di indirizzamento della memoria
- Organizzazione: unità operative e loro connessioni
 - Interfacce tra calcolatore e periferiche
 - Tecnologia per le memorie

Esempio

- Istruzione per la moltiplicazione:
 - Decidere se è disponibile, è una decisione architeturale
 - Come implementarla (circuiti per la moltiplicazione o somme ripetute) è una decisione di organizzazione (costo, velocità, ...)
- Modelli diversi della stessa marca: stessa architettura, organizzazione diversa
- Esempio: architettura dell'IBM 370 (dal 1970)
 - Fino ad oggi per calcolatori mainframe
 - Varie organizzazioni con costo e prestazioni diverse

Struttura e funzione

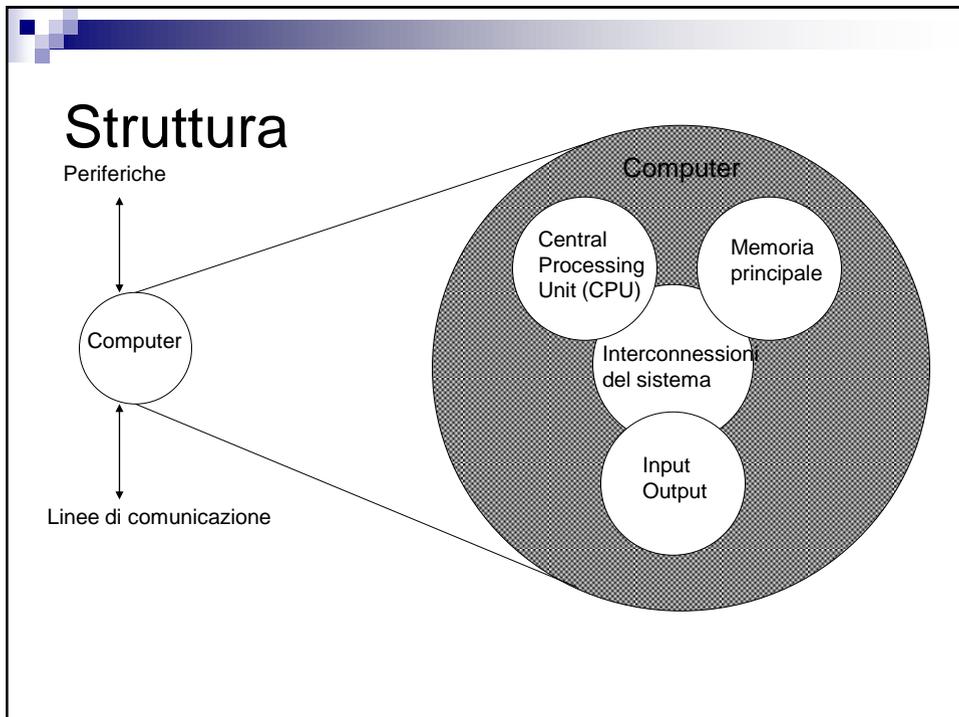
- Calcolatore:
 - Insieme di componenti connesse tra loro
- Visione gerarchica
 - Insieme di sottosistemi correlati
 - Ogni sistema ad un livello si basa sulla descrizione astratta del livello successivo
- Ad ogni livello
 - Struttura: come sono correlati i componenti
 - Funzione: cosa fa ciascun componente
- Descrizione top-down:
 - da componenti principali a sottocomponenti, fino a una descrizione completa dei dettagli

Funzioni basilari di un calcolatore (livello più alto della gerarchia)

- Elaborazione dati
- Memorizzazione dati
- Trasmissione dati
 - Input/output o verso un dispositivo remoto
- Controllo
 - Delle tre funzioni sopra

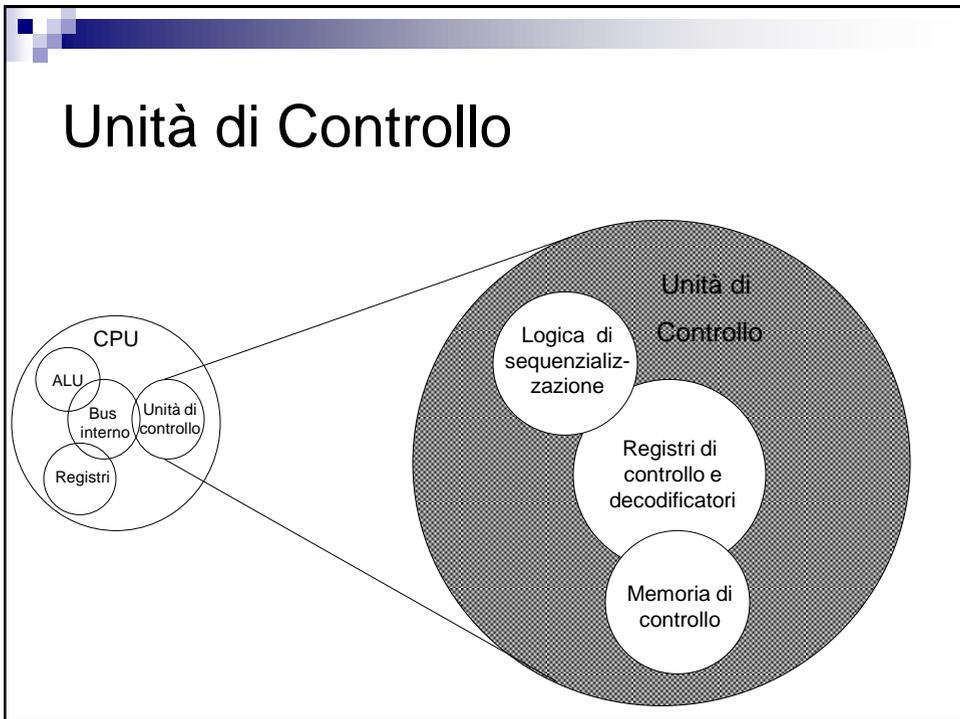
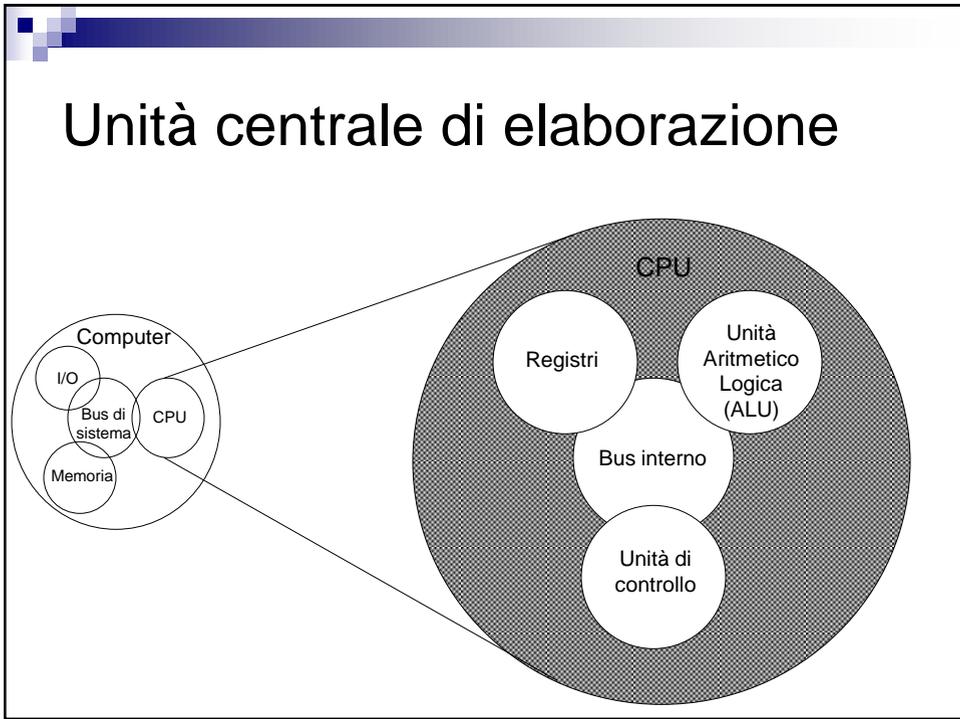
Struttura (livello più alto della gerarchia)

- Quattro componenti principali:
 - Unità centrale di elaborazione (CPU)
 - Esegue le funzioni di elaborazione dati
 - Memoria centrale
 - Per immagazzinare i dati
 - I/O (input/output)
 - Per trasferire i dati tra calcolatore ed esterno
 - Interconnessioni
 - Per far comunicare CPU, memoria centrale, e I/O



Central Processing Unit (Unità Centrale di Elaborazione)

- Unità di controllo
 - Controlla la sequenza di operazioni
- Unità aritmetico-logica (ALU)
 - Elaborazione dati
- Registri
 - Memoria interna della CPU
- Interconnessioni
 - Comunicazione tra unità di controllo, ALU e registri



Perché studiare l'architettura dei calcolatori?

- Capire i compromessi costo-prestazioni
 - Esempio: scegliere il calcolatore migliore a parità di costo
 - spesa maggiore ma memoria più grande o frequenza di clock più alta e quindi maggiore velocità
- Supporto ai linguaggi di programmazione
 - Diverso a seconda delle architetture

L'evoluzione dei calcolatori (cap.2)

Cosa ha influito sull'evoluzione dei calcolatori

- Processori sempre più veloci
- Componenti sempre più piccoli → più vicini → elaborazione più veloce
 - Ma la velocità è derivata anche da nuove tecniche (pipeline, parallelismo, ecc.) che tengono occupato il processore il più possibile
- Memoria sempre più grande
- Capacità e velocità di I/O sempre maggiore
- Tecniche per bilanciare velocità diverse di processore e memoria
 - Memoria cache, ecc.

A cosa servono i calcolatori?

- Ad eseguire compiti ripetitivi o complessi al posto dell'uomo
- Prime idee in campi anche diversi, dove c'era un lavoro ripetitivo che si voleva automatizzare

1801 – telaio programmato

- 1801, Joseph Marie Jacquard
- Telaio con schede perforate
- Dove c'è il buco, la spoletta passa
- Programmare il lavoro del telaio



1822: Charles Babbage

- Analytic Engine:
 - Schede perforate per programmare e anche per immagazzinare dati
 - Motore a vapore
- Ada Byron
 - Programmava per l'AE
 - Subroutine, loop

1890: Hollerith desk

- Schede perforate per fare i calcoli per il censimento USA
- 3 anni invece di 7 previsti
- Bucare schede basandosi sui buchi di altre schede



IBM

- Hollerith creò l'IBM (International Business Machines), 1911
- Schede perforate per molti utilizzi
- Calcolatori per inventari e conti economici
 - Addizioni e sottrazioni
 - No moltiplicazioni (varie addizioni)
 - No numeri negativi

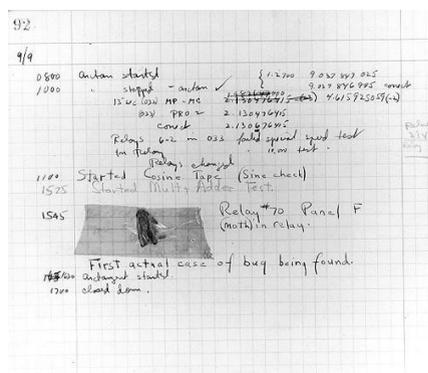
Calcoli complessi per i militari

- Calcoli balistici
- 1944: IBM e Harvard costruirono Mark I
 - Programmabile, digitale
 - 5 tonnellate, 500 miglia di cavo, motore elettrico
 - Usato per 15 anni



Primo bug!

- Grace Hooper: programmatrice per Mark I
- Trovò un insetto (bug) dentro Mark I e inventò la parola "debugging"
- 1953: inventò il primo linguaggio di programmazione ad alto livello Flow-Matic, che poi diventò COBOL
- Anche il primo compilatore

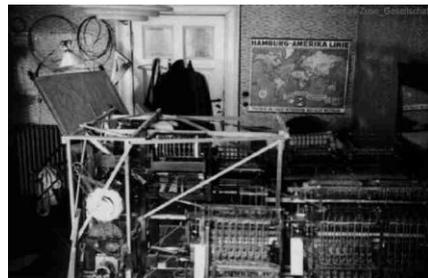


Capacità di Mark I

- Numeri da 23 cifre
 - Somma, sottrazione in 3/10 di sec
 - Divisione in 10 sec
 - Nel 2000: somma in 1 bilionesimo di secondo!
- Solo 72 numeri memorizzati
- Oggi: decine di GB in una RAM e vari TB in un disco rigido
 - Calcolatori meccanici non potevano avere queste velocità

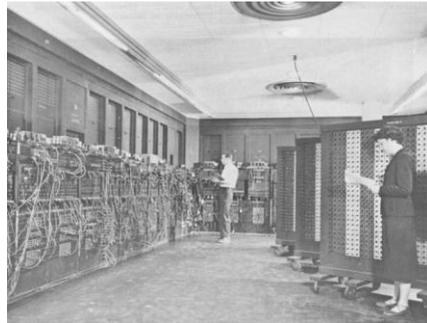
1941: Zuse machine

- General purpose
- Programmabile
- Digital computer
- **Numeri binari**
- **Unità aritmetica per fare i calcoli**
- **Memoria per immagazzinare numeri**
- **Sistema di controllo delle operazioni**
- **Dispositivi di input-output**



1943: ENIAC

- ENIAC (Electrical Numerical Integrator And Computer)
- Prof. Mauchly e Prof. Eckert, Univ. Pennsylvania

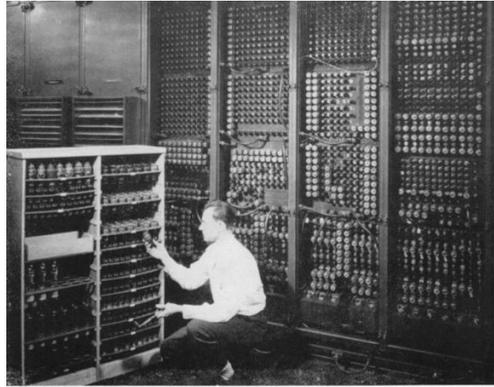


Metà ENIAC

Cosa faceva ENIAC

- Soluzione di equazioni balistiche in campo militare, altri calcoli per militari
- Tecnologia dei tubi a vuoto (valvole delle vecchie televisioni)
- 30 tonnellate, 450 mq, 140 Kilowatt, 18000 tubi a vuoto, 5000 addizioni al secondo
- Macchina decimale: numeri di 10 cifre, ogni cifra in un anello di 10 tubi a vuoto (solo un tubo alla volta in stato ON)

Tubi a vuoto dell'ENIAC



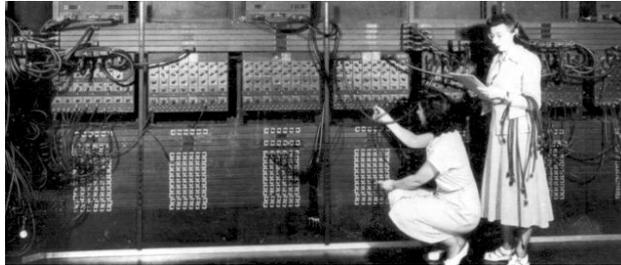
Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

Capacità ENIAC

- 20 numeri alla volta
- Molto più veloce di Mark I
 - Moltiplicazione: 6 sec su Mark I, 2.8/100 di sec su ENIAC
- Primo test ENIAC: 20 sec per un problema che richiedeva 40 ore su un calcolatore meccanico
- Primo lavoro: decidere se la bomba ad idrogeno era fattibile (6 settimane)

Programmare l'ENIAC

- Programmazione manuale: connettere e disconnettere cavi, impostare interruttori
- Giorni per riprogrammare



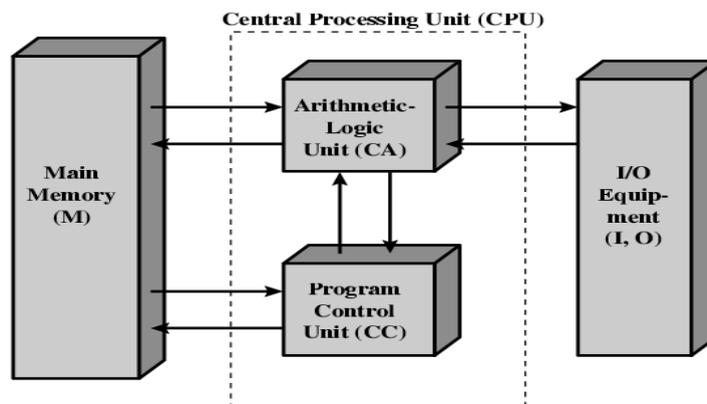
Macchina di Von Neumann

- **Programma memorizzabile come i dati**
- Istruzioni in memoria: decidere il programma specificando una porzione di memoria
- Idea di John von Neumann (consulente ENIAC)
 - Insieme a Mauchly e Eckert
- Proposta di nuovo calcolatore (1945): EDVAC (Electronic Discrete Variable Computer)
- Nuovo elaboratore completo nel 1952 (IAS, presso Institute for Advanced Studies, Princeton)

Struttura della macchina di von Neumann

- Memoria, contiene dati e istruzioni
- Molte operazioni di aritmetica, quindi dispositivi specializzati per eseguirle → unità aritmetico-logica (dati binari)
- Organo centrale per il controllo della sequenza delle operazioni, generico → unità di controllo
- Organi di ingresso e uscita

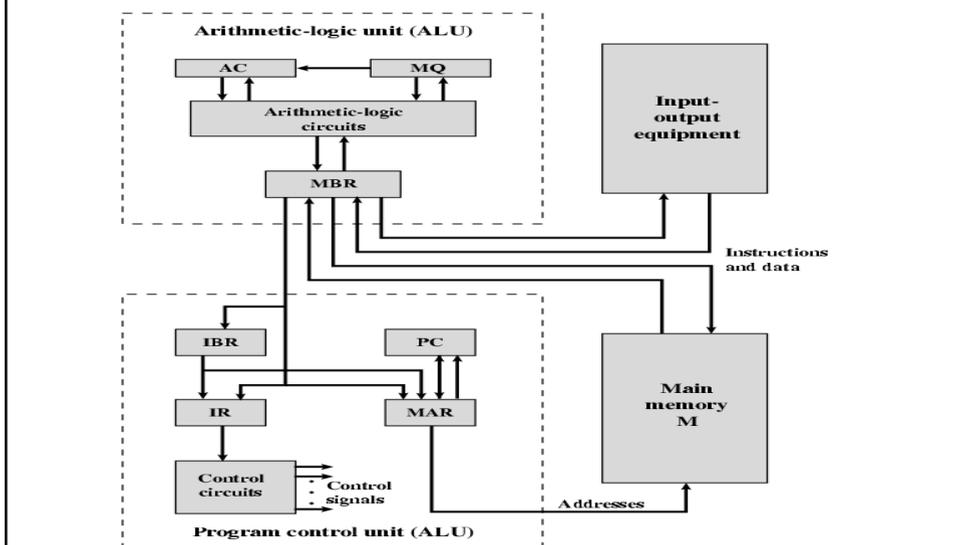
Struttura della macchina di von Neumann



IAS

- Memoria:
 - 1000 locazioni (parole), numerate da 0 a 999 (indirizzo)
 - Ogni parola: 40 cifre binarie (0 o 1, bit)
- Dati e istruzioni in memoria:
 - numeri in forma binaria: bit di segno + 39 bit per il numero
 - istruzioni con codice binario:
 - due in ogni parola
 - 8 bit per codice istruzione, 12 bit per indirizzo parola di memoria
- Unità di controllo: preleva le istruzioni dalla memoria e le esegue una alla volta

Struttura IAS



Registri IAS

- MBR (memory buffer register)
 - Contiene una parola da immagazzinare in memoria, o da leggere dalla memoria
- MAR (memory address register)
 - Contiene un indirizzo di una parola di memoria (dove scrivere il contenuto di MBR o da trasferire in MBR)
- IR (instruction register)
 - Contiene 8 bit per il codice operativo dell'istruzione in corso
- IBR (instruction buffer register)
 - Contiene temporaneamente l'istruzione destra di una parola
- PC (program counter)
 - Indirizzo della prossima coppia di istruzioni da prendere dalla memoria
- AC (accumulator) e MQ (multiplier quotient)
 - Temporaneamente, operandi e risultati parziali delle operazioni della ALU

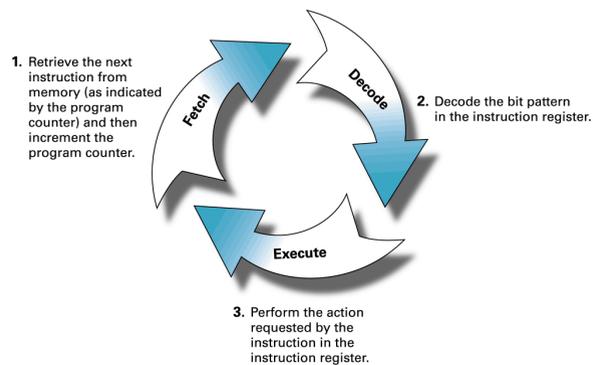
Ciclo della CPU

- La CPU esegue un programma memorizzato in memoria prendendo ad una ad una le istruzioni
- Ordine: quello in cui sono memorizzate

Ciclo della CPU

- **Prelievo dell'istruzione (fetch):**
 - Istruzione letta da IBR o dalla memoria tramite MBR, IBR, IR e MAR
 - Carica il codice dell'istruzione successiva nell'IR e indirizzo in MAR
- **Esecuzione dell'istruzione:**
 - Attiva i circuiti necessari per l'operazione da eseguire

Ciclo della CPU



Istruzioni IAS

- 21 in totale
 - Trasferimento dati (dalla M ai registri ALU, o viceversa, o tra due registri ALU)
 - Salto incondizionato
 - Salto condizionato
 - Aritmetiche
 - Modifica di indirizzo
 - Inserisce indirizzi in istruzioni da mettere in M
- Primi 8 bit: uno dei 21 codici
- Successivi 12 bit: quale delle 1000 celle di M è coinvolta nell'istruzione

Esempi di istruzioni IAS

- LOAD MQ, M(X)
 - Trasferisce il contenuto della cella di memoria di indirizzo X in MQ
- STOR M(X)
 - Trasferisce il contenuto dell'accumulatore nella locazione X della memoria
- JUMP M(X,0:19)
 - Carica l'istruzione dalla metà sinistra di M(X)
- ADD M(X)
 - Somma M(X) ad AC e mette il risultato in AC

Calcolatori commerciali

- Anni '50, due aziende principali: Sperry e IBM
- Sperry: UNIVAC I, UNIVAC II, ...
- 1953: IBM 701 per applicazioni scientifiche
- 1955: IBM 702 per applicazioni business
- Distinzione poi persa

L'avvento dei transistor

- Transistor: componenti discreti
 - Come condensatori, resistori, ...
 - Prodotti separatamente e poi fissati su schede di masonite, poi installate sui calcolatori
 - Rispetto a tubi a vuoto: più piccolo, meno costoso, meno calore
 - Inventato a Bell Labs nel 1947
 - Seconda generazione di calcolatori
- Per inserire un transistor:
 - Costoso e lungo se molti transistor

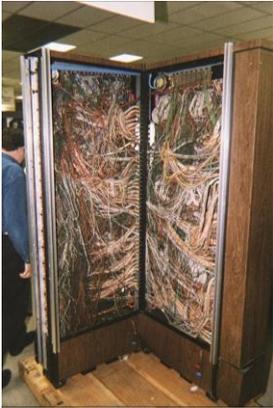


1959: IBM Stretch



Console

Cavi di un calcolatore dell'epoca



Seconda generazione

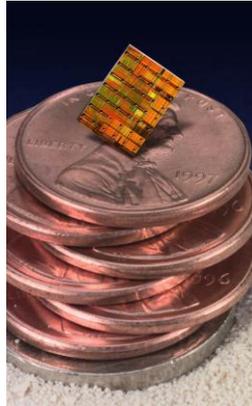
- Transistor
- Unità aritmetiche/logiche più complesse
- Linguaggi di programmazione ad alto livello
- Software di sistema
- Mini-computer (come DEC PDP-1, 1957)

IBM Stretch: transistor

- 150.000 transistor invece che tubi a vuoto
- Più piccoli ma sempre elementi separati
- Nel 1980: tutti in un unico circuito integrato
- Pentium 4: 42,000,000 transistor su un pezzo di silicio

Terza generazione: circuiti integrati

- 1958: invenzione del circuito integrato
 - Unico pezzo di silicio per molti componenti e le loro connessioni
 - Col tempo, sempre più componenti in un circuito integrato



Microelettronica

- Porta logica
 - Dispositivo che esegue una semplice funzione logica
 - Esempio: se A e B sono veri allora C è vero (porta AND)
- Cella di memoria: dispositivo in grado di memorizzare un bit (due stati possibili)
- Calcolatore: numero grandissimo di porte logiche e celle di M

Funzioni

- Memorizzazione dati
 - celle di memoria
- Elaborazione dati
 - porte logiche
- Trasferimento dati
 - tra memoria e memoria, direttamente o attraverso porte logiche
- Controllo
 - segnali di controllo per attivare le porte logiche o leggere/scrivere una cella di memoria

Fine anni 50

- Non solo pezzi singoli per università e laboratori di ricerca
- Mauchly e Eckert produssero UNIVAC (Universal Automatic Computer), primo computer commerciale
- Primo ad avere il nastro magnetico
- Poi UNIVAC fallì, e invece IBM prese il predominio sul mercato



Mainframe computer

- 1970: mainframe computers (come IBM 360, IBM 7094)



IBM 7094

Time sharing

- Come interagire con un mainframe?
- Time-sharing (condivisione di tempo): tanti utenti, un pò di tempo per ciascuno
- Macchina da scrivere a motore per inserire comandi, carta per risultati (10 caratteri al sec)



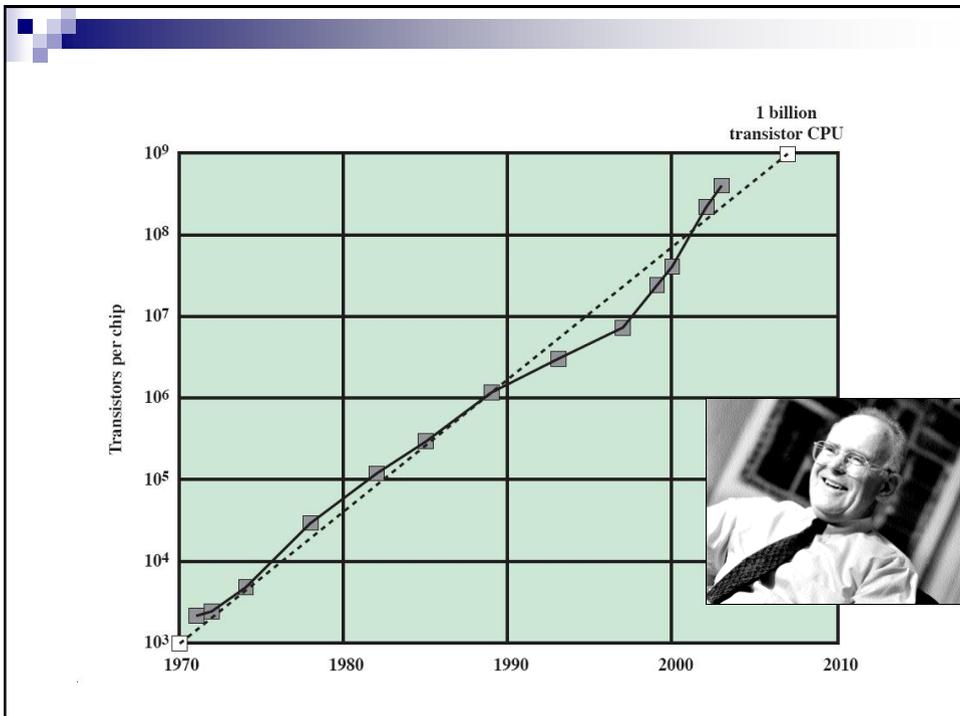
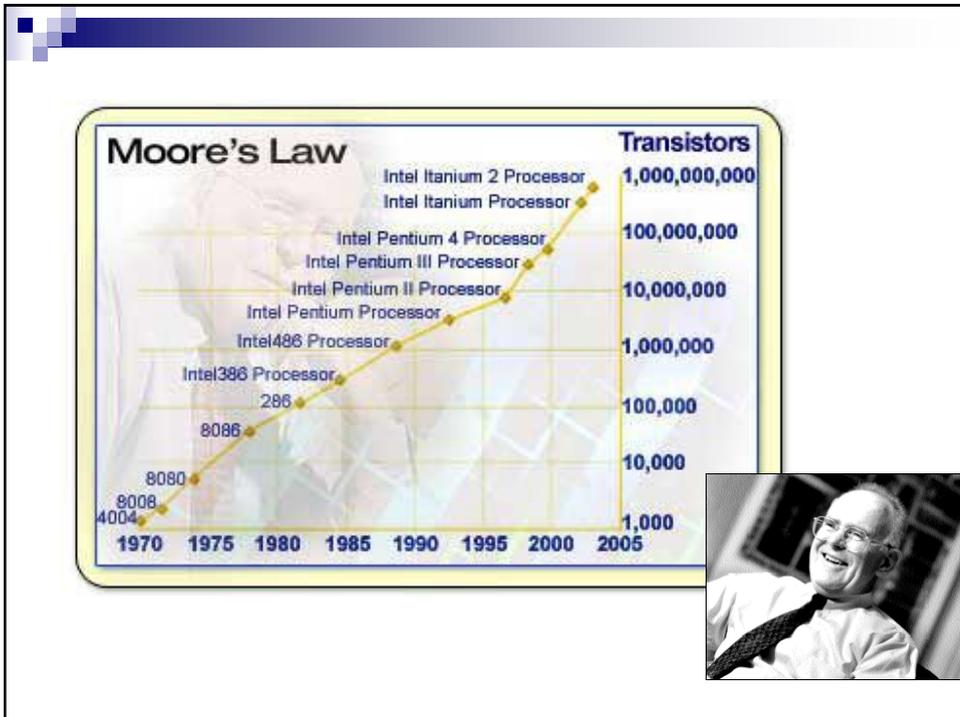
Elaborazione batch

- Secondo modo di interagire con un mainframe
- Tutto il tempo per un utente
- Programma preparato prima su schede perforate



Legge di Moore

- Moore (uno dei fondatori di Intel), 1965
 - Prestazioni e numero di transistor in un chip raddoppia ogni anno
- Dal 1970, più lento: raddoppia ogni 18 mesi
- Conseguenze:
 - Costo del chip invariato → minor costo totale
 - Circuiti più vicini → maggiore velocità
 - Calcolatori più piccoli
 - Minori requisiti di raffreddamento e alimentazione



Generazioni di calcolatori

- Tubi a vuoto - 1946-1957
- Transistor - 1958-1964
- Integrazione su piccola scala – dal 1965
 - Fino a 100 componenti su un chip
- Integrazione su media scala – fino al 1971
 - 100-3,000 dispositivi su un chip
- Integrazione su larga scala- 1971-1977
 - 3,000 - 100,000 dispositivi su un chip
- Integrazione su grandissima scala – dal 1978 fino ad oggi
 - 100,000 - 100,000,000 dispositivi su un chip
- Integrazione su ultra larga scala
 - Più di 100,000,000 dispositivi su un chip

Computer personali

- Microprocessori: computer su un circuito integrato (Intel 1971)
- Primo microprocessore: Intel 4004
 - General purpose
 - 2300 transistor
 - 108.000 cicli al secondo (108 kHz)
 - Invece di 42 mil. transistor e 2GHz del Pentium 4
 - Costo: \$360 invece dei mil. di dollari per IBM360
- Intel 8080 usato nel computer MITS Altair, primo personal computer generico (1974)
- Pentium 4 compatibile con Intel 8080
- 16-bit microprocessors a fine anni '70
- 32-bit microprocessore Intel (80386) nel1985



Evoluzione architettura Intel x46

- Intel 8080 (1974): primo microprocessore general-purpose, 8 bit (dati verso la memoria)
- 8086 (1978): 16 bit, cache
- 80286 (1982): memoria di 16 MByte
- 80386: 32-bit, più programmi eseguiti sulla stessa macchina (multitasking)
- 80486 (1985): cache, pipeline, processore per operazioni matematiche complesse
- Pentium (1993), Pentium Pro (1995): più istruzioni in parallelo
- Pentium II (1997): video, audio, grafica
- Pentium III (1999): grafica 3D
- Pentium 4 (2000): multimedia
- Core: due processori su un solo chip
- Core 2 (2006): 64 bit
- Core 2 quad (2008): quattro processori in un chip
- Architettura Intel x86: dal 1978 ad oggi una nuova istruzione al mese → quasi 500 istruzioni oggi
- 1978, 8086: clock 5 MHz, 29.000 transistor
- 2008, Core 2: 3GHz (600 volte più veloce), 820 milioni di transistor (28000 volte in più), quasi lo stesso costo e poco più grande



Intel® Core™
i7-980X
6 core