Early von Neumann-architecture computers [edit]

The *First Draft* described a design that was used by many universities and corporations to construct their computers.[16] Among these various computers, only ILLIAC and ORDVAC had compatible instruction sets.

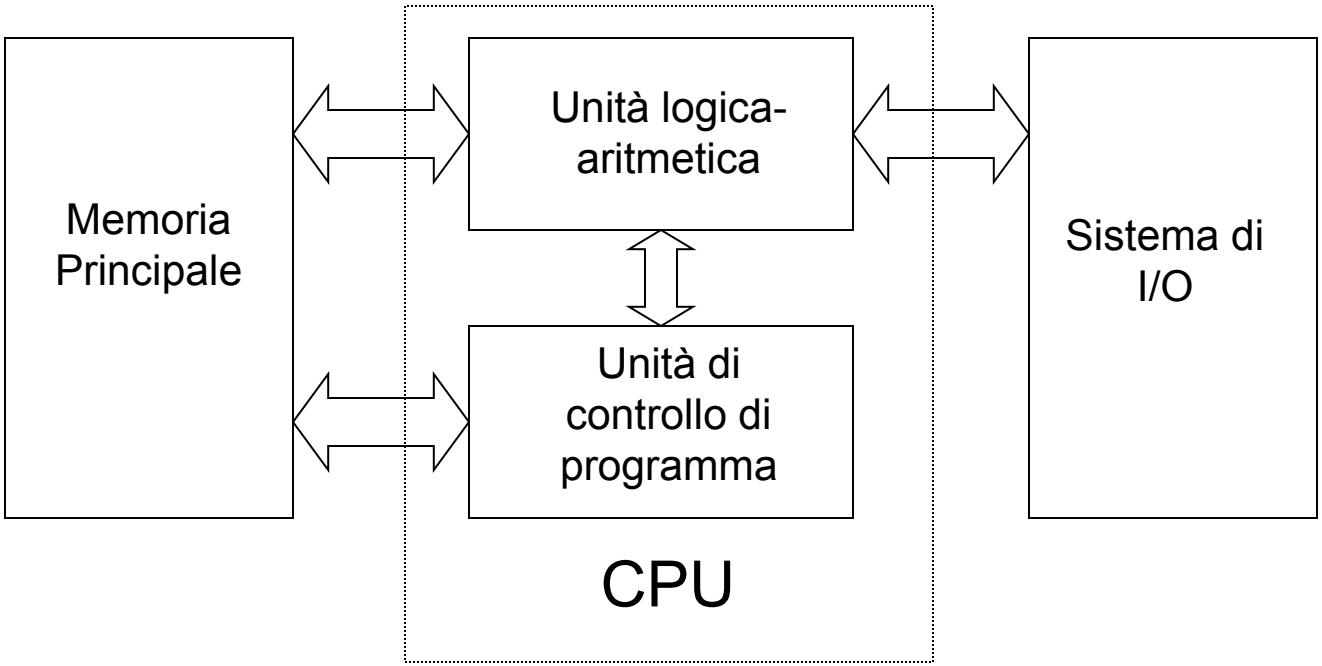
- ARC2 (Birkbeck, University of London) officially came online on May 12, 1948.[17]
- Manchester Small-Scale Experimental Machine (SSEM), nicknamed "Baby" (University of Manchester, England) made its first successful run of a stored-program on June 21, 1948.
- EDSAC (University of Cambridge, England) was the first practical stored-program electronic computer (May 1949)
- Manchester Mark 1 (University of Manchester, England) Developed from the SSEM (June 1949)
- CSIRAC (Council for Scientific and Industrial Research) Australia (November 1949)
- EDVAC (Ballistic Research Laboratory, Computing Laboratory at Aberdeen Proving Ground 1951)
- ORDVAC (U-Illinois) at Aberdeen Proving Ground, Maryland (completed November 1951)[18]
- IAS machine at Princeton University (January 1952)
- MANIAC I at Los Alamos Scientific Laboratory (March 1952)
- ILLIAC at the University of Illinois, (September 1952)
- BESM-1 in Moscow (1952)
- AVIDAC at Argonne National Laboratory (1953)
- ORACLE at Oak Ridge National Laboratory (June 1953)
- BESK in Stockholm (1953)
- JOHNNIAC at RAND Corporation (January 1954)
- DASK in Denmark (1955)
- WEIZAC at the Weizmann Institute of Science in Rehovot, Israel (1955)
- PERM in Munich (1956?)
- SILLIAC in Sydney (1956)

Early stored-program computers [edit]

Macchina di von Neumann/Turing

- Concetto di programma memorizzato
- Memoria principale per dati e istruzioni
- ALU opera su dati in formato binario
- Unità di controllo che "interpreta" le istruzioni in memoria e le esegue
- I sistemi di input e output sono governati dall'unità di controllo
- Princeton Institute for Advanced Studies
  - » IAS
- Completata nel 1952

La Macchina di von Neumann



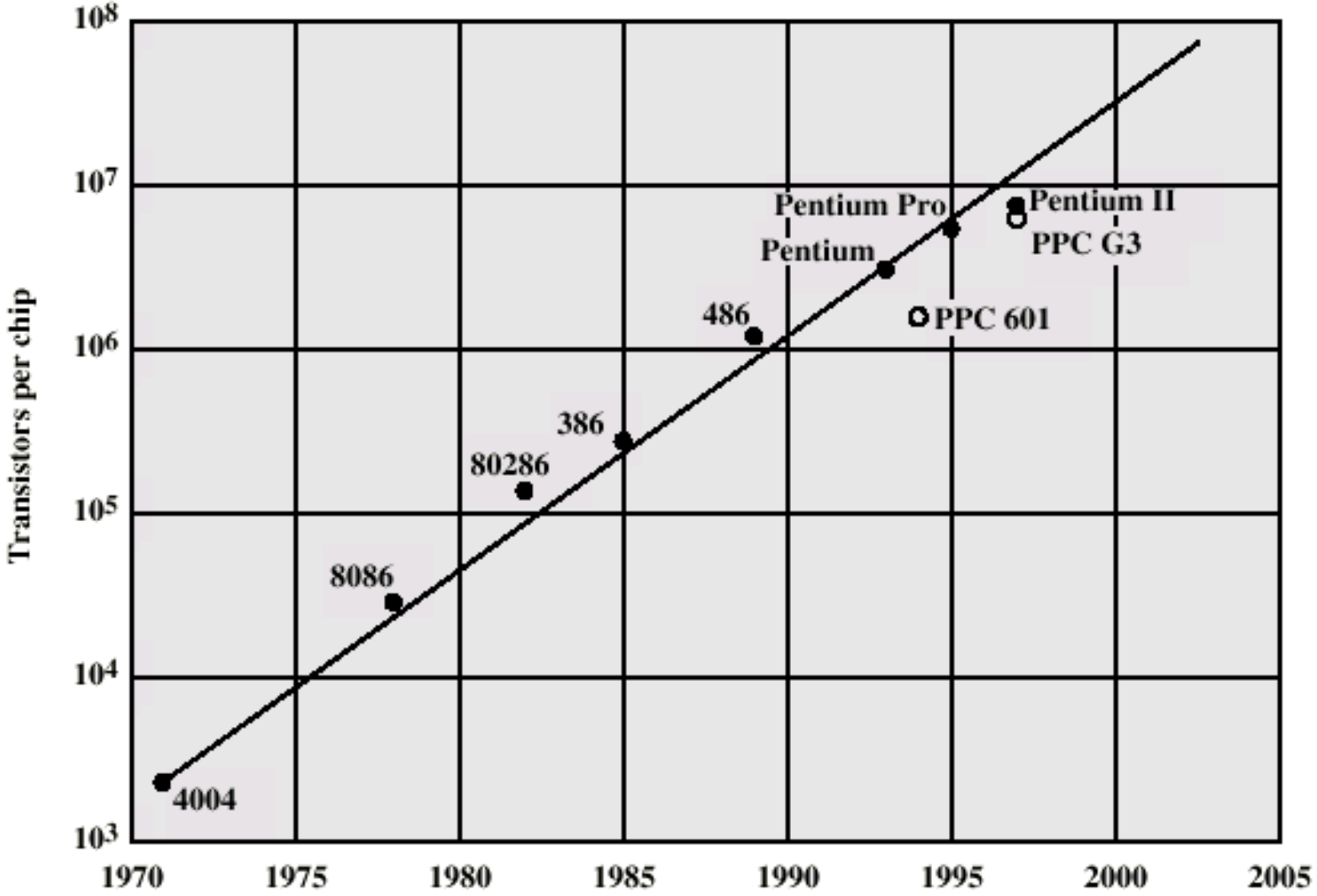
Generazioni di Computer

- Tubi a vuoto - 1946-1957
- Transistor - 1958-1964
- Small scale integration SSI - 1965
  - Fino a 100 dispositivi per chip
- Medium scale integration MSI - 1971
  - 100-3,000 dispositivi per chip
- Large scale integration LSI - 1971-1977
  - 3,000 - 100,000 dispositivi per chip
- Very large scale integration VLSI - 1978 in poi
  - 100,000 - 100,000,000 dispositivi per chip
- Ultra large scale integration
  - Oltre 100,000,000 dispositivi per chip

Legge di Moore (~1965)

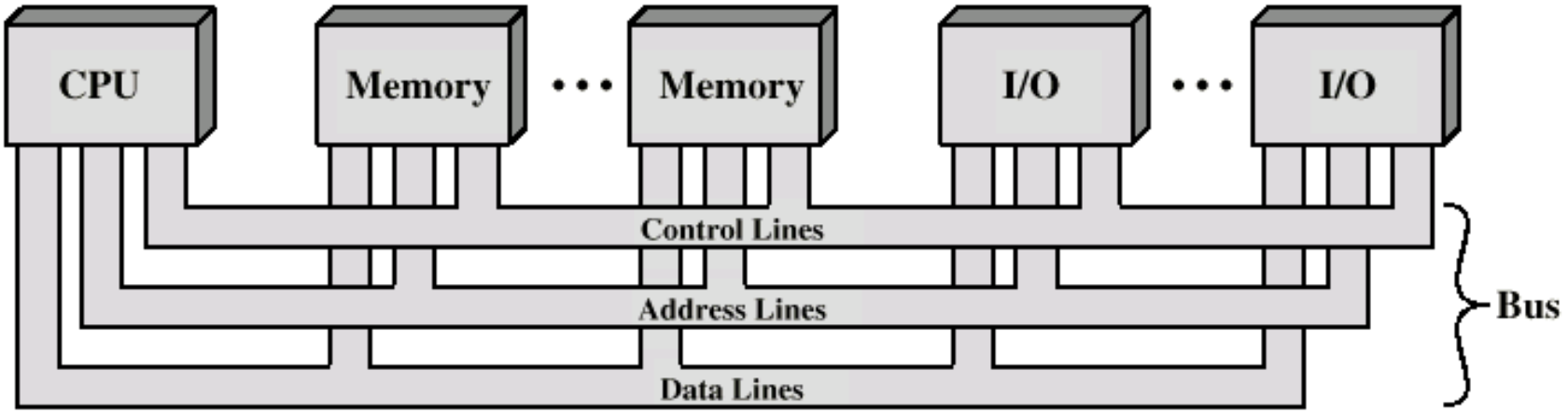
- Gordon Moore – co-fondatore della Intel
- Legge di Moore: Numero dei transistor per chip raddoppierà ogni anno
- Negli anni '70 lo sviluppo si è rallentato
  - » Numero di transistors per chip raddoppierà ogni 18 mesi
- Costo dei chip rimasto invariato
- Alta densità di componenti implica minori distanze
  - » Maggiori prestazioni
- Altri vantaggi:
  - » Minore potenza/raffreddamento
  - » Minor numero di interconnessioni, maggiore affidabilità

Crescita del numero di transistor della CPU

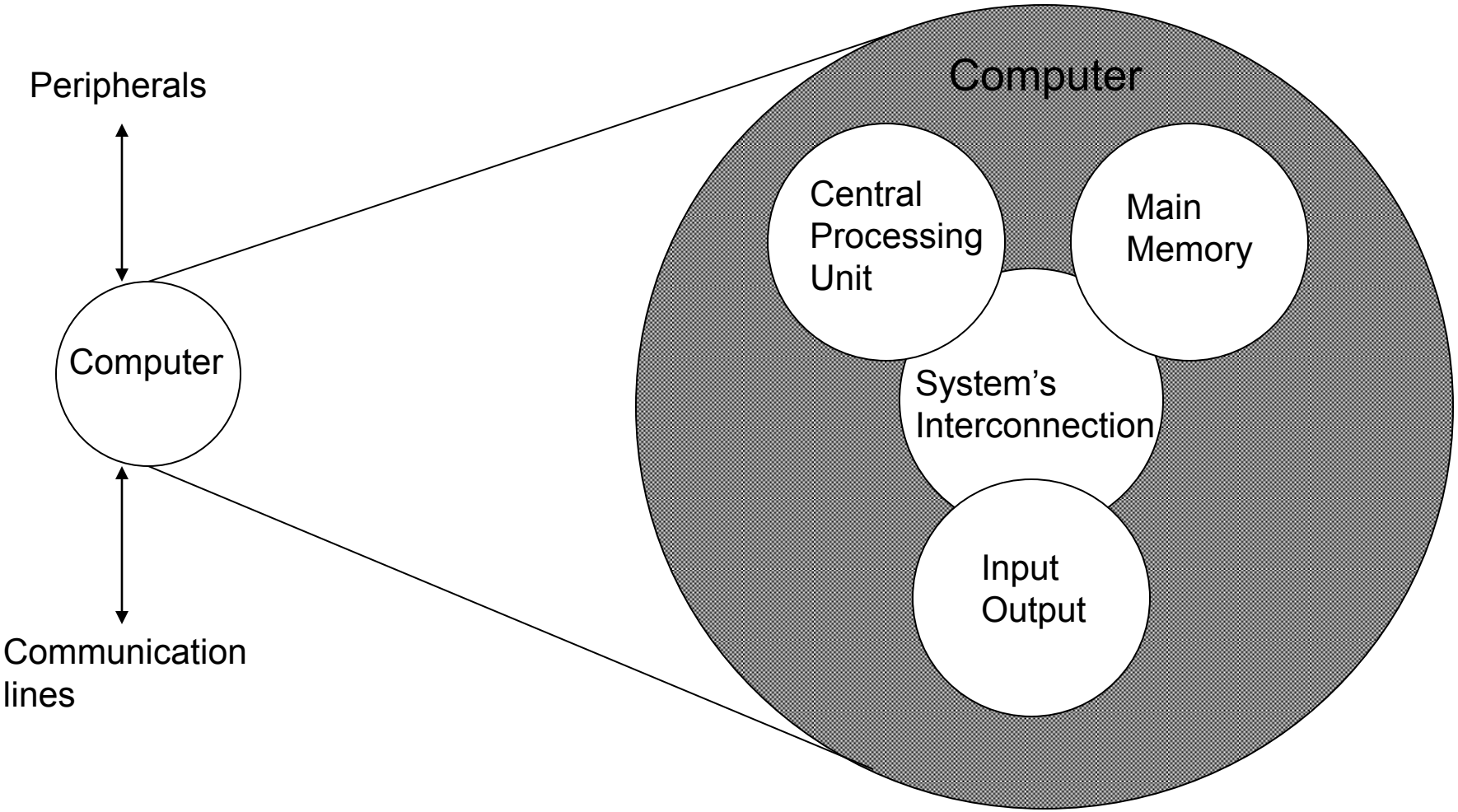


Calcolatore Convenzionale

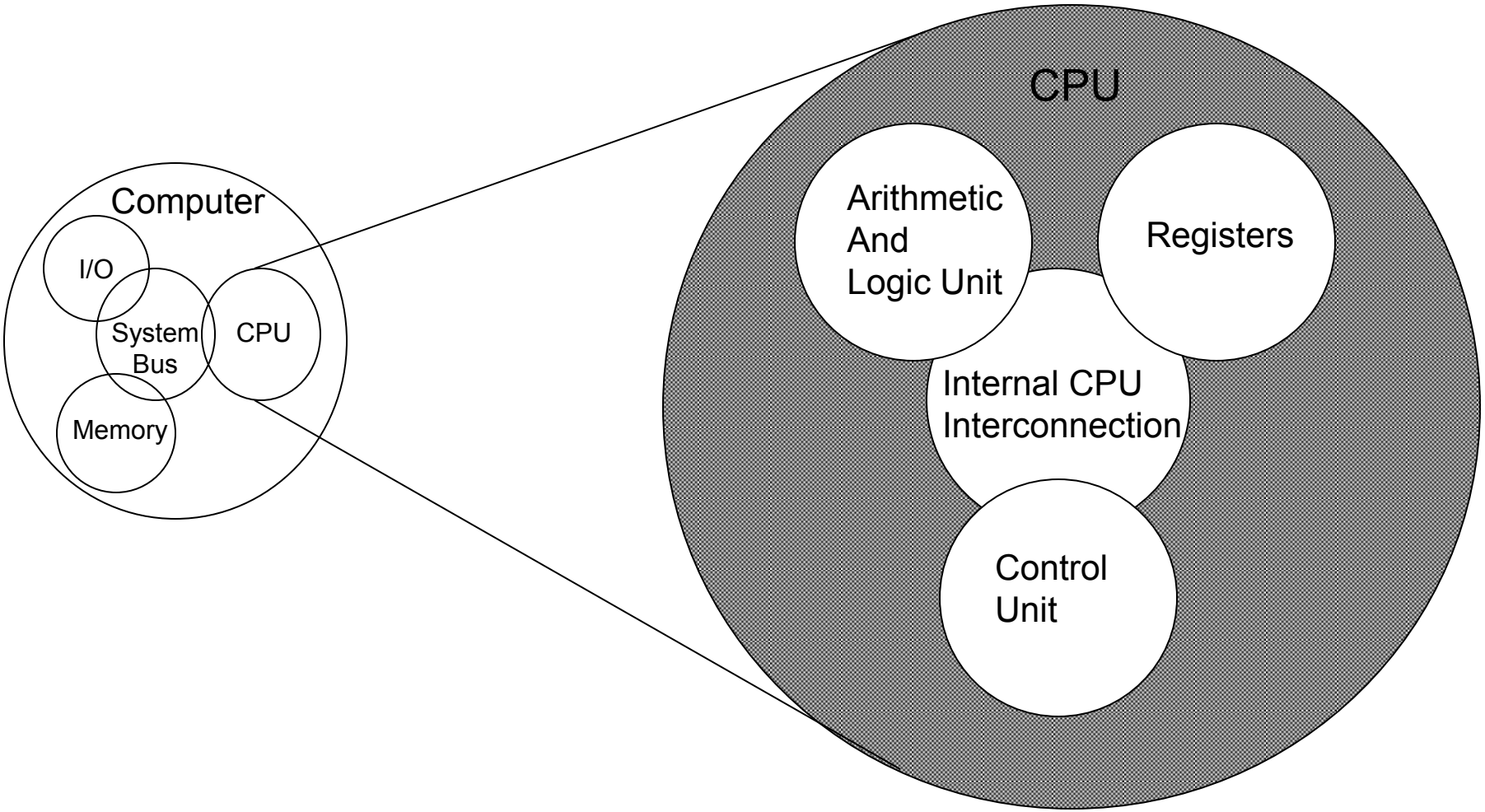
- Comunicazione avviene tramite una struttura condivisa detta bus



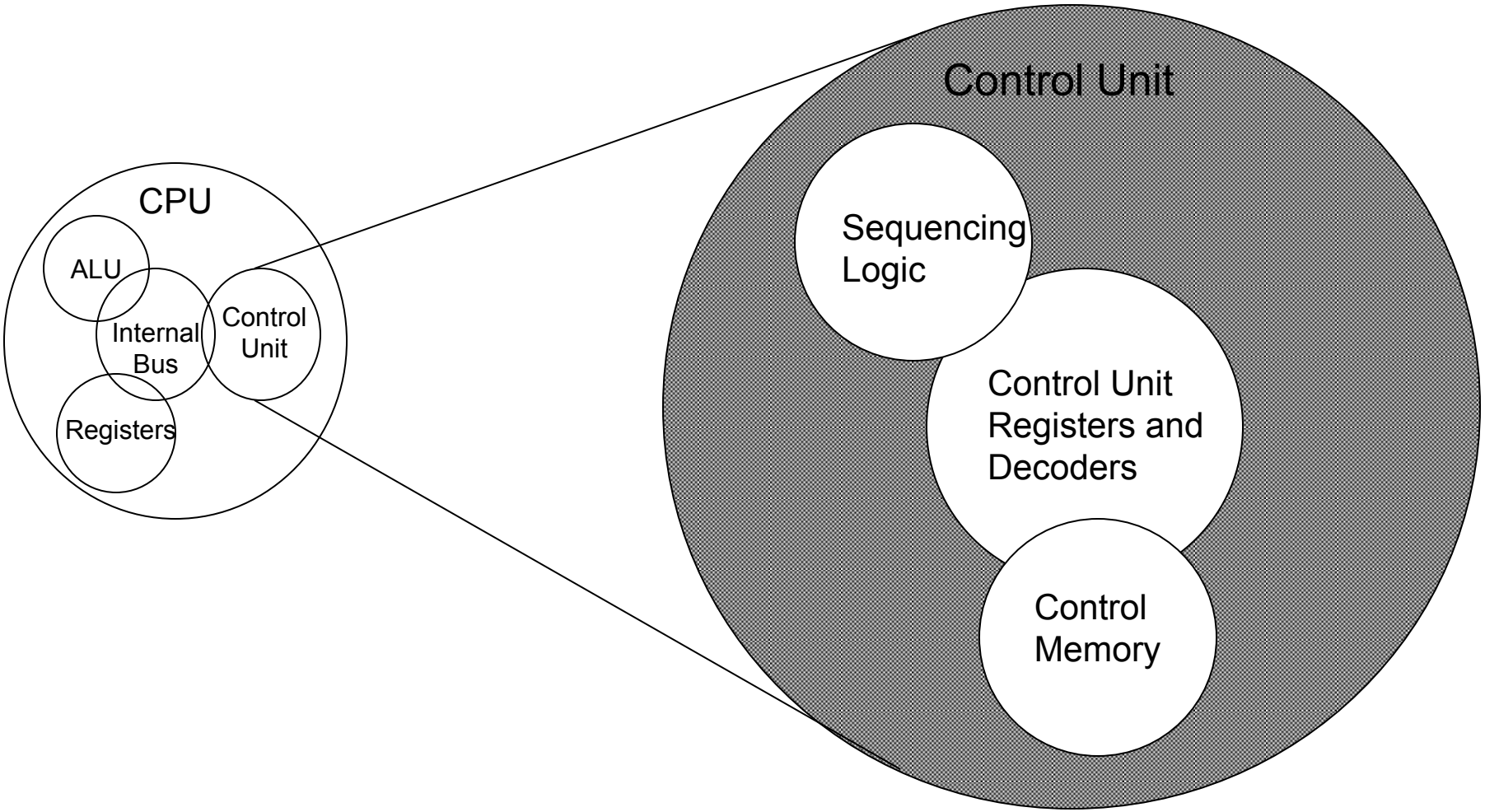
Struttura di un calcolatore



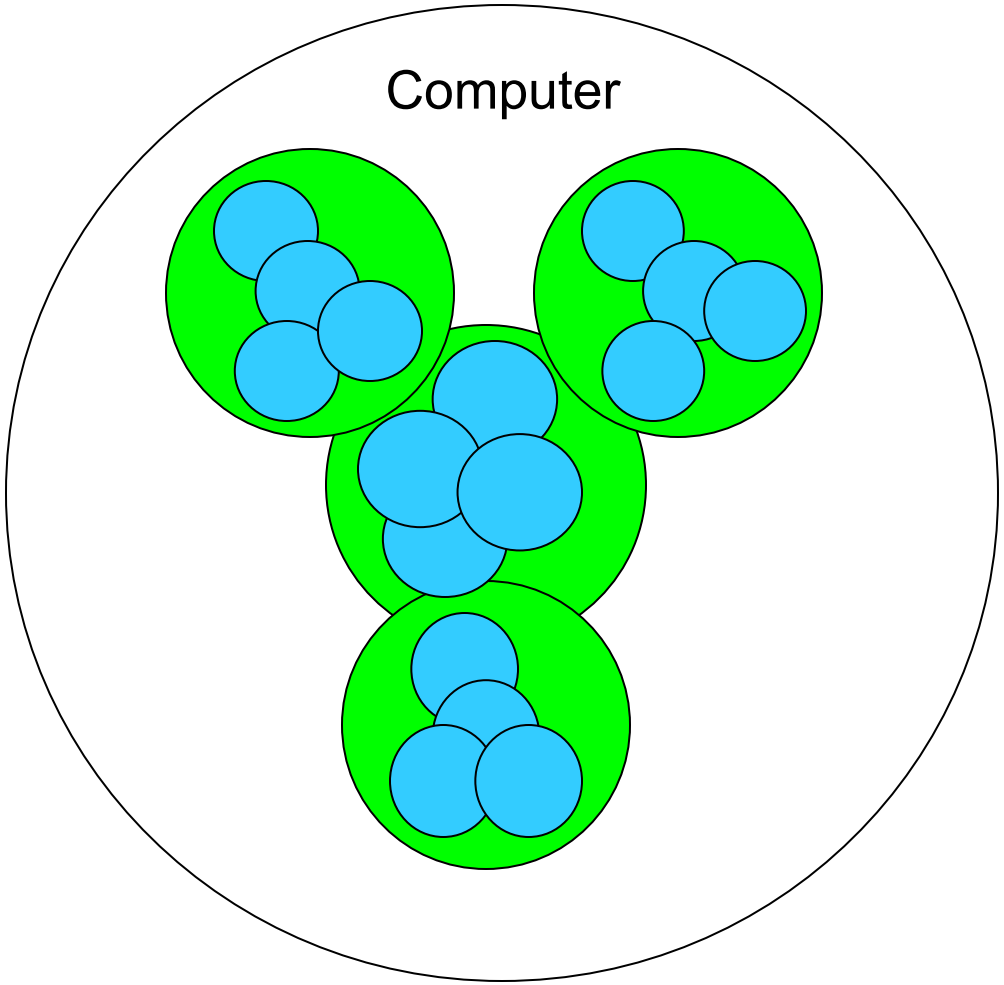
Struttura della CPU



Struttura dell'unità di controllo

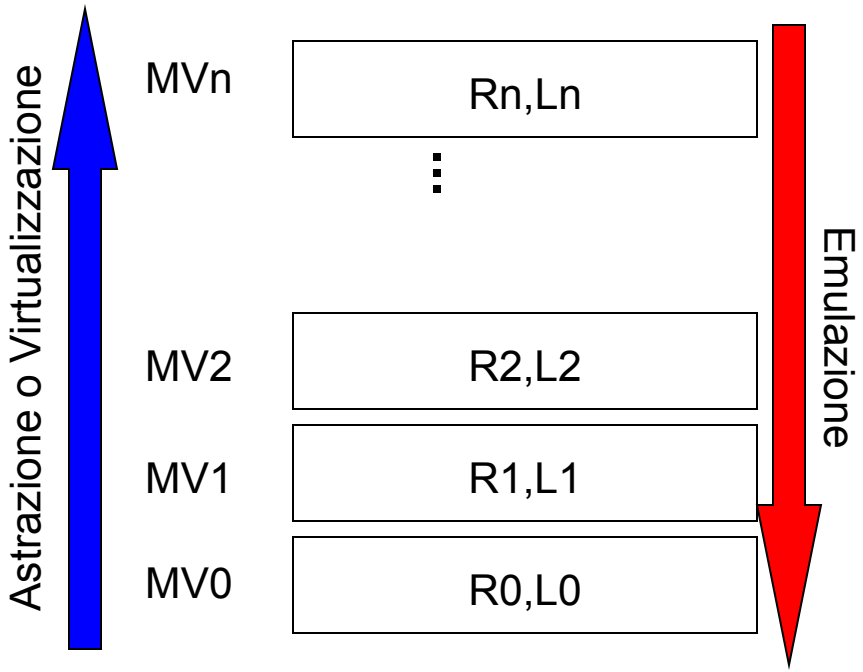


Strutturazione a livelli di un Sistema di Elaborazione



Macchina Virtuale

- Funzionalità ripartite secondo un numero di livelli, o macchine virtuali {MV0,…,MVn}
- Ogni livello MVi è caratterizzato da:
  - Struttura: Un insieme di risorse Ri, visibili a chi programma al livello i
  - Funzionalità: Un linguaggio Li per il controllo e gestione degli elementi di Ri
- MV0 è il livello dei componenti fisici (elettronici)
- MVn: es. Java Virtual Machine



Macchina Virtuale (2)

- MV0=Macchina Hardware
  - R0={componenti elettronici,…}, L0=?
- MV1=Macchina Firmware
  - R1={registri, ALU,…}, L1=ling. di microprogrammazione
- MV2=Macchina Assembler
  - R2={locazioni di memoria,…}, L2=linguaggio "macchina"
- MV3=Sistema Operativo
  - …
- …

Relazione tra Macchine Virtuali

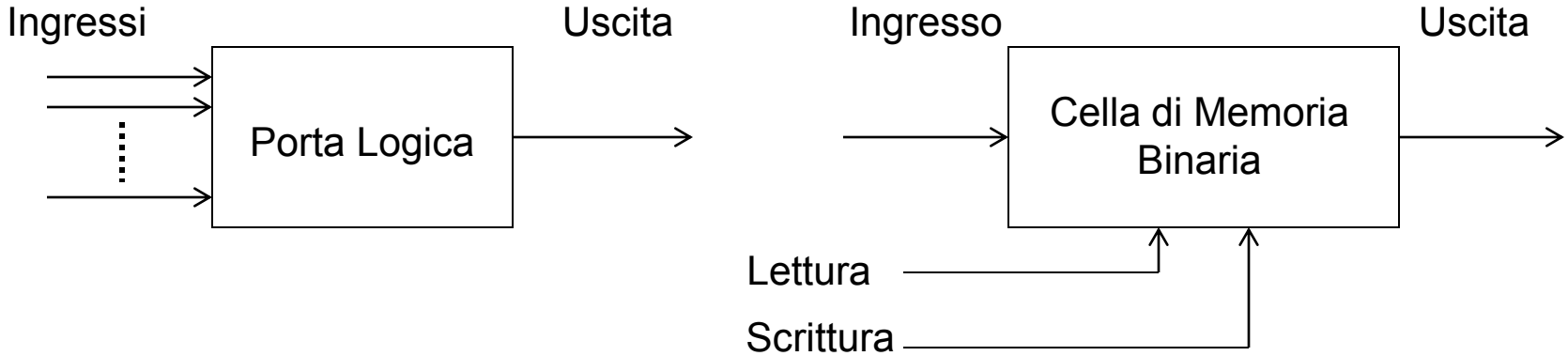
- Concetto di Supporto a Tempo di Esecuzione
  - » Per ogni livello MVi (i>0) l'insieme dei livelli

    {MVj | j<i}

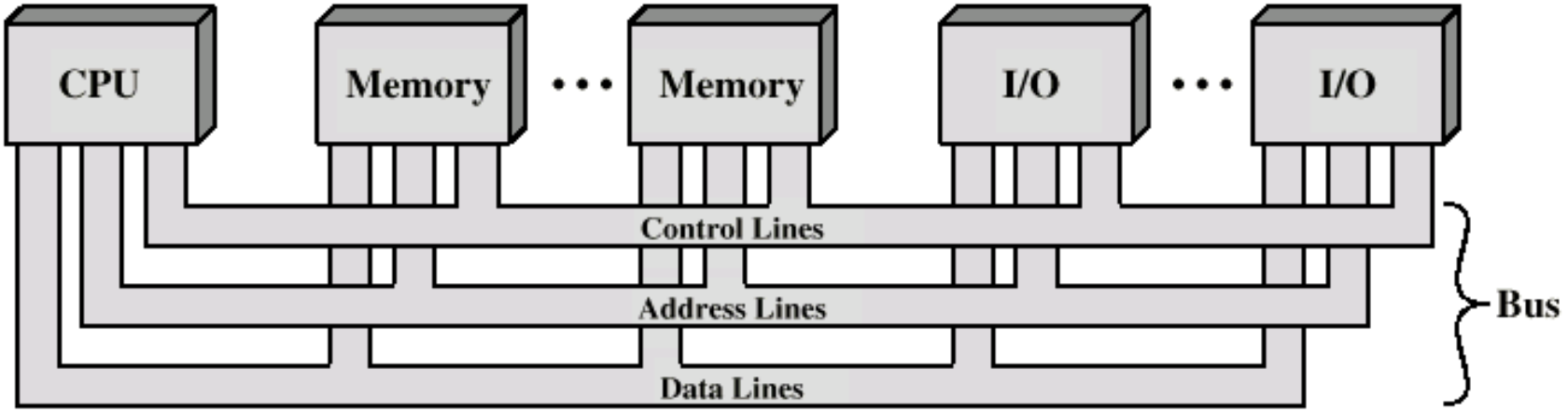
    permette di implementare complessivamente il Supporto aTempo di Esecuzione di Li, cioè una collezione di algoritmi e strutture dati che provvedono all'interpretazione dei meccanismi di Li

Macchina Hardware: RO={Porte Logiche, Celle di Memoria,…}

- Funzioni da realizzare:
  - Memorizzazione: Celle di Memoria
  - Elaborazione: Porte Logiche
    - Ingressi-Uscite di tipo binario
      - 0/1 – assenza/presenza di tensione elettrica
  - Trasferimento: Collegamenti fisici tra componenti
  - Controllo: Segnali di Controllo



Calcolatore Convenzionale: Funzione Principale

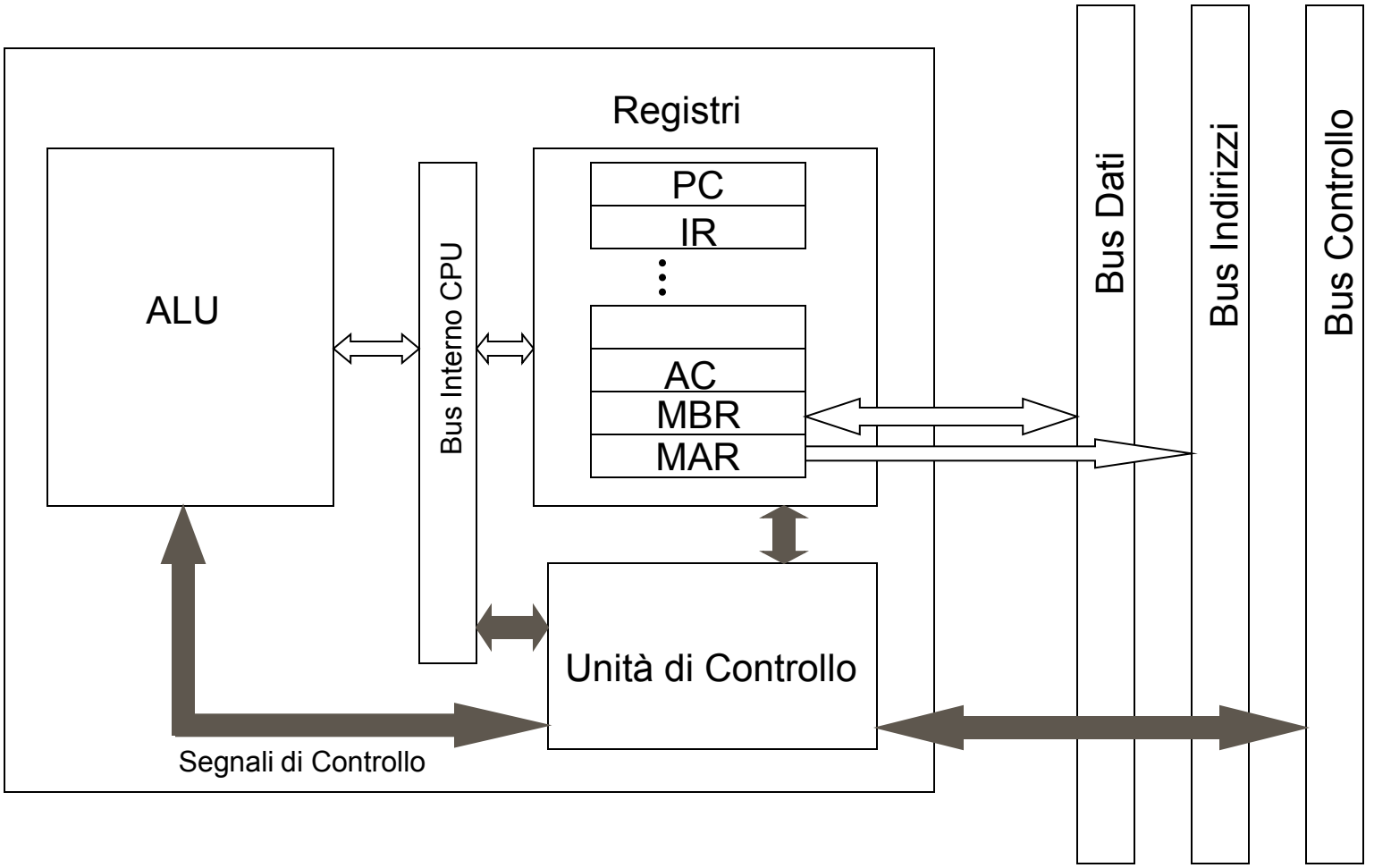


- Esecuzione di Programmi
  - » Programma: una sequenza di istruzioni "macchina"
  - » Istruzione: operazione logica o aritmetica su dati o trasferimento dati

Calcolatore Convenzionale: Componenti

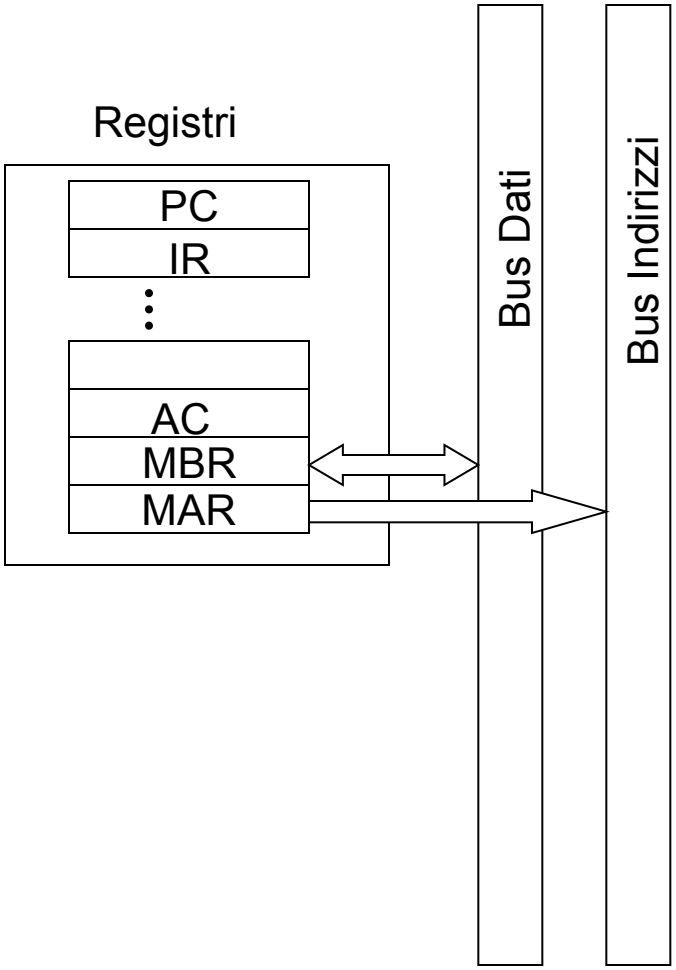
- CPU: Esegue le istruzioni
  - » Unità di Controllo
    - controlla le operazioni della CPU generando i segnali di controllo
  - » ALU
    - Esegue operazioni logico aritmetiche sui dati
  - » Registri
    - Memoria interna CPU
- Memoria: Memorizzazione temporanea di dati e istruzioni
- I/O: Trasferimento di dati e istruzioni da e verso il calcolatore

CPU


Registri
ALU
Bus Interno CPU
PC
IR
AC
MBR
MAR
Unità di Controllo
Segnali di Controllo
Bus Dati
Bus Indirizzi
Bus Controllo

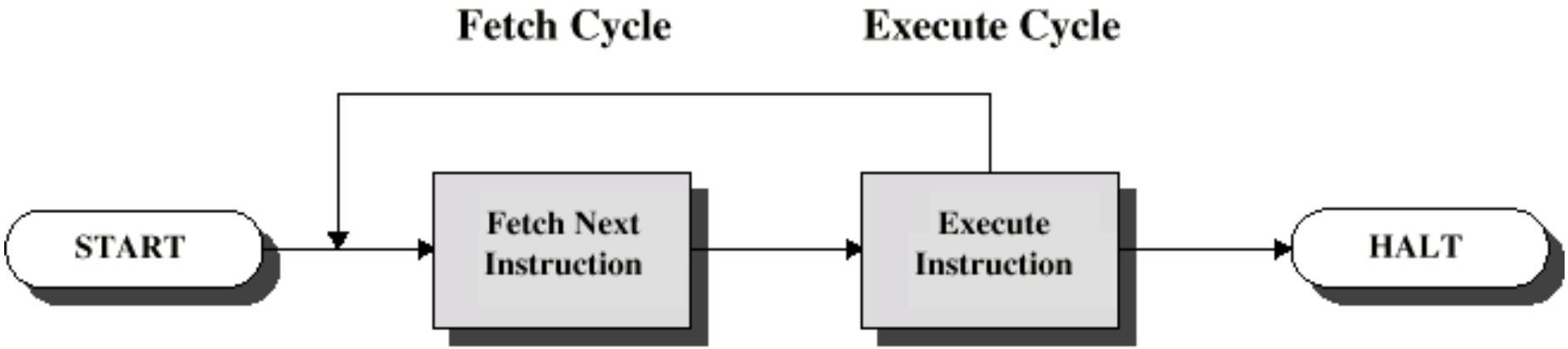
Registri della CPU

- Program Counter
  - Indirizzo Prossima Istruzione
- Instruction Register
  - Codice Istruzione da eseguire
- Memory Address Register
  - Indirizzo di Memoria da R/W
- Memory Buffer Register
  - Dato da R/W in Memoria
- Accumulatore
  - Operandi o Risultati delle operazioni svolte dall'ALU
- …



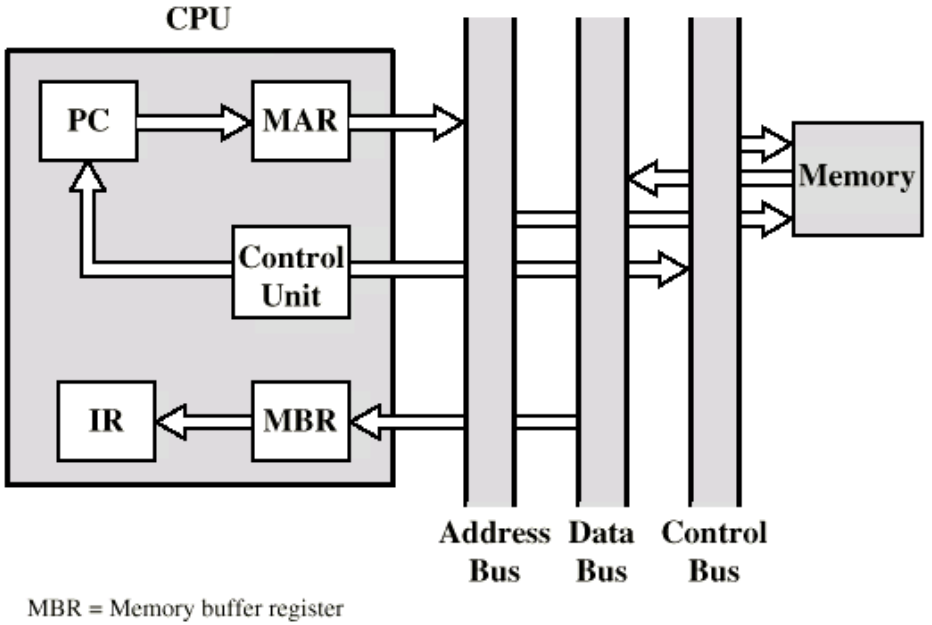
Ciclo di Istruzione

- Due passi:
  - Prelievo
  - Esecuzione



Fase ("Ciclo") di Prelievo

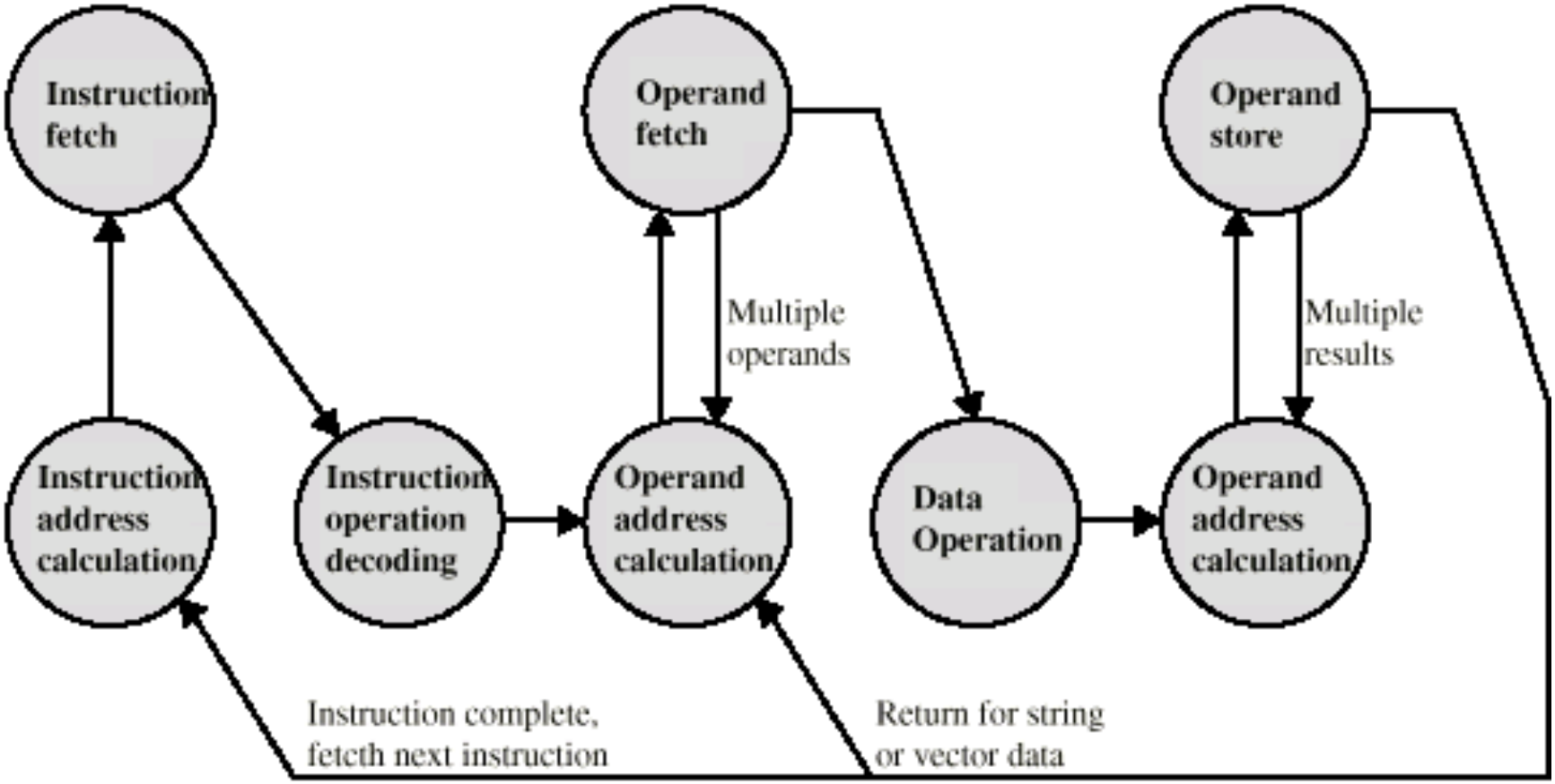
- Processore preleva l'istruzione dalla locazione di memoria puntata da PC
- Incrementa PC
  - » A meno di "salti"
- L'istruzione viene caricata in IR
- L'istruzione viene decodificata e le operazioni eseguite



Fase di Esecuzione: Tipi di Istruzioni

- Processore-Memoria
  - » Trasferimento dati CPU-memoria (Load, Store)
- Processore-I/O
  - » Trasferimento dati CPU-I/O
- Elaborazione Dati
  - » Operazione logico-aritmetica sui dati
- Controllo
  - » Alterazione della sequenza delle istruzioni
  - » e.g. un salto
- Una Combinazione di queste

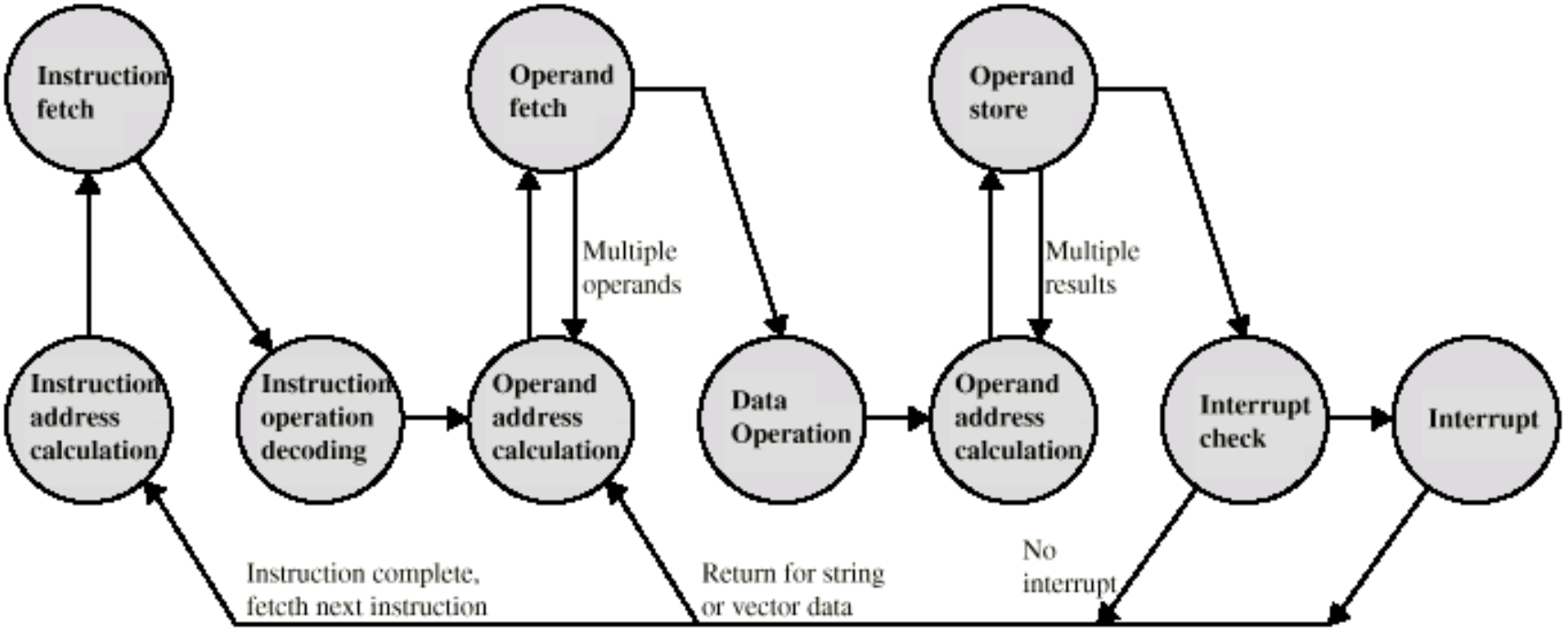
Ciclo di Istruzione: Diagramma di Stato



Interruzioni

- Meccanismo che permette ai moduli (di I/O – memoria) di interrompere l'elaborazione del processore per segnalare un evento (asincrono)
  - I/O
    - I/O controller
      - pressione tasto
  - Programma
    - overflow, divisione per zero
  - Timer
    - Generato dal timer interno del processore
  - Guasto Hardware
    - Errore di parità della memoria

Ciclo di Istruzione (con interruzioni): Diagramma di Stato



Ciclo di Interruzione

- Al termine del ciclo di esecuzione, il processore controlla se ci sono state richieste di interruzione e…
  - Indicate da un segnale di interrupt

1. Senza Interrupt: preleva la prossima istruzione
2. Se c'è un Interrupt in attesa:
   - Sospende l'esecuzione del programma corrente
   - Salva il contesto (stato di tutti i registri)
   - Esegue la routine di gestione dell'interrupt
     - PC=indirizzo gestore interrupt
   - Ripristina il contesto di un programma interrotto e ne riprende l'esecuzione