

03. Componenti e Connessioni dei Calcolatori

Architettura di Von Neumann

L'architettura di Von Neumann si fonda su tre principi:

- Dati e istruzioni risiedono nella stessa memoria (accessibile in lettura e scrittura)
- La memoria è indirizzabile per locazione
- L'esecuzione delle istruzioni avviene in modo sequenziale

Un programma può essere realizzato in forma "cablata" (hardware fisso, non modificabile, che esegue solo operazioni predeterminate) oppure in forma software. Nel secondo caso, un hardware generico riceve codici di istruzione che un interprete traduce in segnali di controllo, permettendo di cambiare programma senza modificare i circuiti. La CPU si compone quindi di un interprete delle istruzioni e di un modulo per operazioni aritmetico-logiche.

Componenti Principali

I componenti fondamentali sono: CPU, Memoria principale, Sistemi di I/O e le connessioni tra essi.

CPU — Contiene i registri essenziali:

- **PC** (Program Counter): indirizzo della prossima istruzione
- **IR** (Instruction Register): istruzione corrente
- **MAR** (Memory Address Register): indirizzo di memoria
- **MBR** (Memory Buffer Register): buffer dati memoria
- **I/O AR**: registro indirizzi I/O
- **I/O BR**: registro buffer I/O
- **Unità di esecuzione**: elabora i dati secondo le istruzioni decodificate

Memoria principale — Immagazzina istruzioni e dati organizzati in celle indirizzabili. Supporta lettura, scrittura, esecuzione sequenziale e salti.

Moduli I/O — Gestiscono la comunicazione tra CPU e periferiche attraverso porte, buffer e segnali di interruzione.

Ciclo di Esecuzione

Il ciclo base della CPU comprende due fasi:

- **Fetch**: prelievo dell'istruzione dalla memoria all'indirizzo indicato da PC, poi incremento di PC
- **Execute**: esecuzione dell'operazione codificata nell'istruzione (ora in IR)

Il ciclo completo prevede più sottofasi:

1. Calcolo dell'indirizzo dell'istruzione
2. Fetch dell'istruzione
3. Decodifica dell'operazione
4. Calcolo dell'indirizzo degli operandi
5. Fetch degli operandi
6. Esecuzione dell'operazione sui dati
7. Calcolo dell'indirizzo di destinazione
8. Store del risultato

Per operazioni su stringhe o vettori, le fasi di fetch e store degli operandi possono ripetersi.

Tipi di operazioni:

- **Processore-memoria:** trasferimento dati tra CPU e memoria
- **Processore-I/O:** trasferimento dati tra CPU e periferiche
- **Elaborazione dati:** operazioni logiche o aritmetiche
- **Controllo:** alterazione della sequenza di esecuzione (salti)

Interruzioni

Le interruzioni permettono ad altri moduli di interrompere la normale sequenza di esecuzione della CPU.

Tipi principali:

- **Program:** generate da condizioni come overflow o divisione per zero
- **Timer:** generate da un timer interno alla CPU
- **I/O:** segnalano il completamento di operazioni di I/O
- **Hardware failure:** segnalano guasti (es. mancanza di alimentazione)

Scopo: migliorare l'efficienza evitando che la CPU rimanga inattiva in attesa del completamento di operazioni di I/O (i dispositivi esterni sono più lenti della CPU).

Gestione di un'interruzione:

1. Sospensione del programma corrente
2. Salvataggio del contesto (incluso l'indirizzo della prossima istruzione)
3. Impostazione di PC all'indirizzo del gestore dell'interruzione (interrupt handler)
4. Esecuzione della routine di gestione
5. Ripristino del contesto e ripresa del programma interrotto

Il ciclo di esecuzione con interruzioni aggiunge una fase di controllo dopo l'execute: se le interruzioni sono abilitate, la CPU verifica la presenza di interruzioni pendenti e le gestisce prima di proseguire.

Interruzioni multiple — due approcci:

- **Disabilitazione:** la CPU ignora nuove interruzioni durante la gestione della prima, processandole poi in sequenza
- **Sistema a priorità:** interruzioni a bassa priorità possono essere interrotte da interruzioni a priorità maggiore, creando un annidamento gestito con ritorno a cascata

Connessioni e Bus

Connessioni per tipo di componente:

- **Memoria:** riceve/spedisce dati, riceve indirizzi e segnali di controllo (lettura/scrittura)
- **Moduli I/O:** funzionano similmente alla memoria dal punto di vista della CPU, ma gestiscono anche dati esterni e segnali di interruzione
- **CPU:** legge istruzioni e dati, scrive risultati, emette segnali di controllo, riceve interruzioni

Bus — Principale mezzo di interconnessione:

- Collega due o più dispositivi
- Mezzo di trasmissione condiviso (un segnale trasmesso è disponibile a tutti)
- Solo un dispositivo alla volta può trasmettere
- Costituito da più linee parallele che trasmettono simultaneamente bit multipli

Bus di sistema — tre gruppi di linee:

- **Bus dati:** trasporta dati e istruzioni; l'ampiezza determina l'efficienza del sistema
- **Bus indirizzi:** indica sorgente o destinazione dei dati; l'ampiezza determina la massima memoria indirizzabile
- **Bus di controllo:** gestisce accesso e uso delle altre linee (segnali: memory write/read, richiesta bus, bus grant, interrupt request, clock)

Uso del bus:

- Per inviare dati: ottenere l'accesso al bus, poi trasferire i dati
- Per ricevere dati: ottenere l'accesso, inviare una richiesta, attendere la risposta

Bus singoli vs multipli: con un solo bus si rischia ritardo e congestione; architetture a bus multipli (local bus, system bus, high-speed bus, expansion bus) separano il traffico per velocità e tipo di dispositivo.

Temporizzazione sincrona: eventi determinati da un clock condiviso; una linea di clock trasmette una sequenza alternata di 0 e 1, e tutti gli eventi iniziano all'inizio di un ciclo di clock.

QuickPath Interconnect (QPI)

QPI è un'architettura di interconnessione punto a punto che sostituisce il bus tradizionale con connessioni dirette multiple tra componenti. Utilizza trasferimento dati a pacchetto con intestazioni di controllo e codici di errore.

Architettura a quattro livelli:

- **Livello fisico:** gestisce cavi, circuiti e logica di trasmissione/ricezione; unità di trasferimento di 20 bit (Phit); utilizza trasmissione differenziale a bassa tensione (LVDS) e distribuzione multilinea su canali paralleli
- **Livello link:** garantisce trasmissione affidabile e controllo del flusso; opera su unità di 80 bit (Flit = 72 bit payload + 8 bit CRC); controllo flusso a crediti per evitare sovraccarichi; controllo errori con ritrasmissione
- **Livello routing:** determina il percorso dei pacchetti attraverso tabelle di instradamento definite dal firmware
- **Livello protocollo:** definisce le regole di scambio pacchetti con formato flessibile e supporto per la coerenza della cache tra core e memoria principale