

Architetture hardware di sistemi di calcolo

Lezione 8 di Fondamenti di Informatica

Docenti: Marina Madonia & Giuseppe Scollo

Università di Catania

Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica, I livello, AA 2008-09

Indice

1. Architetture hardware di sistemi di calcolo
2. architettura Von Neumann
3. registri della CPU
4. struttura del linguaggio macchina
5. ciclo macchina
6. comunicazione con altri dispositivi
7. velocità di comunicazione dei dati
8. calcolo in pipeline
9. architetture multiprocessore
10. temi per ulteriori approfondimenti

architettura Von Neumann

elaborazione sequenziale di un programma residente in memoria

CPU: Central Processing Unit, o processore, consta di:

Unità di Controllo: coordinamento dell'esecuzione

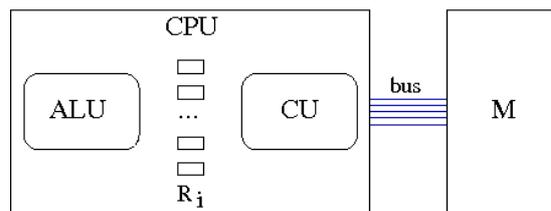
Unità Aritmetico-Logica (ALU): esecuzione di operazioni

registri: celle di memoria ad accesso veloce (~ 1 ns)

memoria cache (nelle architetture odierne)

memoria: ospita il programma in esecuzione e i dati

bus: canale veloce di comunicazione (throughput : ~ 100 MB/s)



architettura Von Neumann

registri della CPU

registri, due categorie:

generici: per gli operandi di operazioni dell'ALU

N.B.: l'ALU non accede al bus

→ anche l'esecuzione di una semplice operazione, su dati in memoria, richiede l'intervento dell'Unità di Controllo

speciali: di pertinenza dell'Unità di Controllo:

PC: Program Counter

contiene l'indirizzo di un'istruzione

IR: Instruction Register

contiene l'istruzione corrente

SP: Stack Pointer

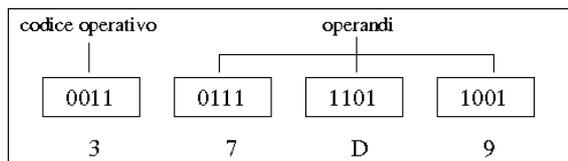
contiene un indirizzo nella pila di sistema

struttura del linguaggio macchina

linguaggio macchina: insieme delle istruzioni accettate dalla CPU, in una data **codifica**

formato binario delle istruzioni:

codice operativo, eventuali
operandi



lunghezza: fissa o variabile
due tipologie di linguaggi macchina (e di architettura della CPU):

RISC: Reduced Instruction-Set Computer

CISC: Complex Instruction-Set Computer

concetto: microprogramma

classificazione del repertorio di istruzioni macchina:

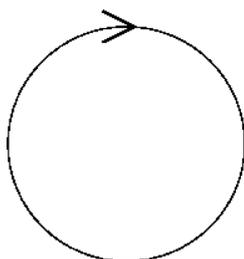
gruppo di trasferimento dei dati (include istruzioni di I/O)

gruppo logico/aritmetico

gruppo di controllo

ciclo macchina

prelievo:
dell'istruzione
dall'indirizzo contenuto
nel PC
suo trasferimento nell'IR
incremento del PC



decodifica:
dell'istruzione nell'IR

esecuzione: dell'istruzione decodificata

→ in quali fasi del ciclo macchina si impiega il bus?

comunicazione con altri dispositivi

comunicazione mediante **controller**

un controller presiede alla comunicazione fra il sistema CPU+memoria e dispositivi periferici di uno stesso tipo (ad es., stampanti, dischi, monitor etc.)

usualmente è una **scheda** alloggiata in una **slot** della scheda madre (ingl. **motherboard**), attraverso cui accede al **bus**

spesso dispone di propria memoria e unità di controllo (progettata ad hoc per il tipo di dispositivo periferico)

memory-mapped I/O : comunicazione CPU-controller mediante istruzioni di trasferimento dati a indirizzi riservati (**porte di I/O**)

STORE: CPU → controller

LOAD: controller → CPU

DMA (direct memory access):

trasferimento di **blocchi** di dati fra memoria e controller senza intervento della CPU

velocità di comunicazione dei dati

bps (bits per second):

velocità di trasferimento dei bit fra due dispositivi

multipli usuali: **Kbps**, **Mbps**, **Gbps**, etc.

capacità o larghezza di banda (bandwidth) di un canale di comunicazione: massima velocità di trasferimento dei bit fra due dispositivi attraverso il canale

tipi di canale di comunicazione:

seriale: trasferimento sequenziale, un bit alla volta (più semplice, più lenta)

parallela: trasferimento simultaneo di più bit su linee fisiche distinte (più complessa, più rapida)

modem (modulatore-demodulatore):

dispositivo di conversione e riconoscimento dei bit nella/dalla rappresentazione in toni di frequenza in uso nei canali telefonici

calcolo in pipeline

architettura **biprocessore**:

si accelera l'esecuzione di un programma sequenziale sfruttando l'**alternanza**, nel ciclo macchina, di:

fasi in cui la CPU **accede al bus**:

prelievo, esecuzione di un'istruzione di trasferimento dati

fasi in cui la CPU **non accede al bus**:

decodifica, esecuzione di un'istruzione logico/aritmetica

con due processori in **pipeline** ("incanalati") si anticipa il prelievo dell'istruzione successiva (**pre-fetch**) da parte di uno dei due, mentre l'altro **decodifica** l'istruzione corrente

ciò è sempre efficace, **tranne** quando l'istruzione corrente è un'**istruzione di controllo**, la cui esecuzione altera la sequenza di esecuzione rispetto a quella di memorizzazione

architetture multiprocessore

la limitata capacità del **bus** ne fa il **collo di bottiglia** dell'architettura Von Neumann

le **architetture multiprocessore** risultano efficaci all'accelerazione dell'esecuzione di algoritmi paralleli, grazie anche alla disponibilità di **più canali di comunicazione** delle istruzioni e/o dei dati

architetture sincrone:

SIMD :Single Instruction, Multiple Data

MIMD :Multiple Instruction, Multiple Data

architetture asincrone:

cluster ("grappolo") di processori, ciascuno con propria **memoria locale**, che cooperano attraverso opportuni **protocolli**, ad es. **MPI: Message Passing Interface**

temi per ulteriori approfondimenti

1. **Architettura di calcolo e dissipazione termica**
La complessità dell'architettura di una CPU spesso, ma non sempre, comporta maggiore consumo energetico e dissipazione termica. Può essere interessante approfondire le cause principali di inefficienza energetica di un'architettura di calcolo.
2. **Microprogrammazione e RISC**
La tecnica della microprogrammazione è stata inizialmente proposta già nel 1957, da Maurice Wilkes, ma non risultò di impiego pratico con le tecnologie dell'epoca. In seguito, invece, già dagli anni '60, l'idea trovò largo impiego, nel progetto di architetture CISC come pure di controller sofisticati. Si può indagare ulteriormente l'argomento per approfondire, ad esempio, perché la microprogrammazione non sia adatta alle architetture RISC, e quali siano le tecniche impiegate con più vantaggio in queste architetture.
3. **Bus di comunicazione con dispositivi periferici**
La crescente varietà di dispositivi periferici per le funzioni più disparate ha fatto emergere l'esigenza di una standardizzazione dei canali di comunicazione fra controller e dispositivi periferici. Una ricognizione sugli standard più diffusi, quali SCSI, IDE, USB, può essere utile.
4. **Supercomputers**
Questo termine designa sistemi di calcolo di altissime prestazioni. Naturalmente, le cifre prestazionali sono relative al periodo storico: le prestazioni di un supercomputer degli anni '60 non distano molto da quelle di un PC odierno, mentre i supercomputer attuali possono constare di decine di migliaia di processori operanti in parallelo. Può essere interessante esplorare lo stato dell'arte nel settore e alcune applicazioni di questi "mostri".