

# Cisc VS Risc

# Filosofia di Progetto pre-RISC (1)

- Hardware considerato meno complesso dei compilatori
- Memoria RAM scarsa e costosa:
  - maggiore priorità nel massimizzare l'efficienza della codifica delle istruzioni piuttosto che nella velocità di decodifica
- CPU equipaggiate con pochi registri:
  - registri hanno costo maggiore risp. a RAM
  - + registri = > costo codifica istruzioni

# Filosofia di Progetto pre-RISC (2)

- Istruzioni assembly complesse in grado di compiere autonomamente molte operazioni:
  - istruzioni di dimensione variabile
  - accessi multipli a RAM all'interno di istruzioni aritmetiche
  - modalità di indirizzamento variegata e complesse

***Complex Instruction Set Computers  
(CISC)***

# Filosofia RISC (1)

- Contesto storico:
  - fine anni 70'
- Viene evidenziato che:
  - modalità di indirizzamento complesse sono usate di rado dai compilatori
  - istruzioni usate di rado tendono ad essere poco ottimizzate:
    - progettisti ottimizzano operazioni comuni
    - + istruzioni semplici a volte sono + veloci!
- Crescita divario prestazioni CPU/RAM:
  - importanza incremento capacità dei registri (e caches):
    - > registri = > spazio sul die

# Filosofia RISC (2)

- 98% costanti è contenuta in 13 bits:
  - usare multipli di 8 bits (tipo PD32) appositamente dedicati è inefficiente
  - set di istruzioni dovrebbero permettere di inglobare costanti all'interno della stessa istruzione:
    - le istruzioni devono essere più compatte per lasciar spazio agli operandi
- Tempo di clock è limitato dall'istruzione più lenta:
  - ottimizzare tali istruzioni (ad es. riducendo il numero di modalità di indirizzamento) produce speed-up generale

# Filosofia RISC (2)

- 98% costanti è contenuta in 13 bits:
  - usare multipli di 8 bits (tipo PD32) appositamente dedicati è inefficiente

## ***Reduced Instruction Set Computers (RISC)***

agli operandi

- Tempo di clock è limitato dall'istruzione più lenta:
  - ottimizzare tali istruzioni (ad es. riducendo il numero di modalità di indirizzamento) produce speed-up generale

# RISC vs CISC

- In processori RISC le istruzioni:
  - o lavorano sui registri
  - o interagiscono con la memoria
  - hanno una struttura più omogenea
- sono perciò anche detti *load-store* CPUs
- PRO:
  - > spazio nelle istruzioni per codificare dati:
    - minore uso di registri o RAM
  - > spazio sul chip per registri
  - > maggiore propensione al pipe-lining
- Contro:
  - + istruzioni per eseguire lo stesso task:
    - + accessi a RAM