UNIVERSITÀ DEGLI STUDI DI TORINO

start@unito

Architettura del calcolatore

IL PROCESSORE

Prof. Enrico Bini

Il processore

Il processore esegue continuamente istruzioni. Non appena viene acceso inizia ad eseguire la prima istruzione contenuta in una locazione di memoria nota e continua così fino allo spegnimento.

Registri

Per l'esecuzione delle istruzioni utilizza i registri. I registri sono piccole zone simili alla memoria, ma interne al processore, su cui il processore può scrivere e leggere i dati più velocemente rispetto a quanto farebbe sulla memoria. I registri vengono identificati da un nome. Il nome dei registri è diverso da processore a processore in base a caratteristiche interne (per esempio, i registri dei processori Intel si chiamano: EAX, EBX, etc.). Seguendo la metafora della cucina, i registri rappresentano il piano di lavoro sul cui gli ingredienti (i dati) vengono messi appena prima di essere elaborati: il cuoco, prima di iniziare a cucinare prende gli ingredienti di cui ha bisogno e li tiene a disposizione sul piano di lavoro. Allo stesso modo, il processore preleva dalla memoria i dati di cui ha bisogno e li mette nei registri. Non appena tutti i dati saranno sui registri, il processore potrà eseguire le istruzioni più velocemente.

Il vantaggio di svolgere operazioni sui registri invece che sulla memoria è che i primi, essendo piccole celle di memoria fisicamente vicine al processore, sono molto più veloci perché permettono l'accesso ai dati senza dover accedere al bus di comunicazione.

Esistono diversi tipi di registro:

- Registri "general purpose": non hanno utilizzi specifici e possono essere utilizzati per contenere dati generici
- Registri che gestiscono gli accessi in memoria (MBR e MAR)
- Registri di stato: mantengono informazioni relative allo stato del processore (per esempio la temperatura o quanto tempo è trascorso dall'accensione)
- Il Program Counter (PC) e l'Instruction Register (IR) servono a gestire le istruzioni e verranno discussi in maggiore dettaglio più avanti.

Unità del processore

All'interno del processore, oltre ai registri, ci sono unità che svolgono funzioni specializzate.

- la ALU (Arithmetic Logic Unit) permette di svolgere operazioni aritmetico-logiche (somma tra contenuti di registri, AND, OR, NOT...)
- la FPU (Floating Point Unit) permette di svolgere operazioni con numeri in virgola mobile (che sono un modo per rappresentare numeri decimali)

- la Control Unit decodifica le istruzioni e ne supervisiona l'esecuzione.

Proseguendo con la metafora della cucina possiamo dire che la ALU e la FPU corrispondono ai vari strumenti a disposizione nella cucina. La Control Unit, invece, è il cuoco stesso.

Fasi di esecuzione

Vediamo adesso tutti i passi che vengono compiuti per eseguire le istruzioni. Nel processore è presente un registro speciale che si chiama Program Counter (PC) che contiene l'indirizzo di memoria da cui prelevare la prossima istruzione da eseguire. Per l'esecuzione di ogni istruzione, il processore compie le seguenti fasi.

Fase di fetch: Il processore preleva dall'indirizzo di memoria contenuto nel Program Counter, l'istruzione che dovrà essere eseguita e la scrive in un altro registro chiamato Istruction Register (IR).

Fase di decode: Il processore esamina l'istruzione presente nell'IR, la decodifica e, sulla base della sua decodifica, ne calcola la lunghezza (in byte). Una volta calcolata la lunghezza dell'istruzione, il processore incrementa il PC del numero corrispondente alla lunghezza dell'istruzione. In questo modo, durante la prossima fase di fetch il Program Counter conterrà correttamente l'indirizzo di memoria dell'istruzione successiva.

Fase di execute: Il processore preleva gli operandi che saranno necessari per l'esecuzione. Il processore esegue quindi l'istruzione. Una volta eseguita si torna alla prima fase (di fetch) in cui verrà prelevata la prossima istruzione. All'accensione, il valore del PC è pari a zero. Inizia quindi a prelevare la prima istruzione dall'indirizzo zero e così via come descritto in precedenza, fino al suo spegnimento.

Modalità di codifica delle istruzioni

Ci sono modalità diverse per codificare le istruzioni. Le due famiglie di codifica principali sono:

1. Complex Instruction Set Computer (CISC)
2. Reduced Instruction Set Computer (RISC)

La codifica CISC è una codifica ricca: permette di avere a disposizione istruzioni complesse che possono fare più cose. La codifica RISC, invece, è più povera in quanto mette a disposizione poche elementari istruzioni. Per questo motivo, per eseguire uno stesso compito ci sarà bisogno di più istruzioni RISC rispetto al numero di istruzioni CISC necessarie. Inoltre, l'esecuzione di una singola istruzione RISC è più veloce rispetto all'esecuzione di una CISC.

Per capire meglio, si paragonino i due alfabeti cinese e latino. Il primo, parallelamente a CISC, permette di tradurre più informazioni con un unico simbolo (ovvero l'istruzione); al contrario per tradurre nell'alfabeto latino la stessa informazione da un carattere cinese si deve usare più di un carattere. Per

esempio, il singolo carattere cinese “成” viene trascritto con 5 caratteri latini con “cheng”.
Inoltre, una stessa parola si esprime con un numero minore di caratteri cinesi, rispetto al numero di caratteri latini necessari. Per esempio, la parola “computer” composta da 8 caratteri, in cinese si scrive “电脑” quindi con 2 caratteri.
Tipicamente, i processori all'interno di Desktop Computer e Personal Computer sono Intel e usano una codifica CISC. Diversamente, i processori degli smart-phone sono tipicamente processori ARM, che usano una codifica RISC delle istruzioni.

tipo codifica	CISC	RISC
complessità singola istruzione	alta	bassa
numero di istruzioni disponibili	molte	poche
numero di istruzioni per codificare uno stesso programma	poche	molte
velocità di esecuzione di una singola istruzione	lenta	veloce

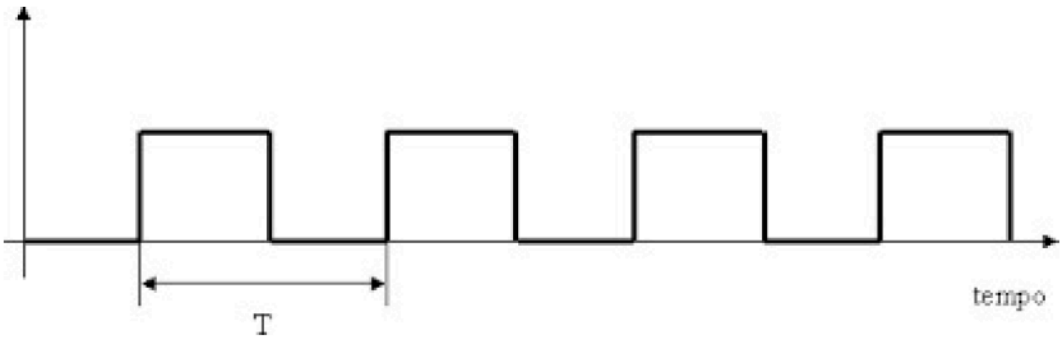
Confronto fra CISC e RISC.

La scelta di codifica delle istruzioni determina l'Instructure Set Architecture (ISA), che è il modo con cui sono codificate le istruzioni del processore. CISC e RISC sono due famiglie di ISA. La scelta dell'ISA è influenzata da motivi commerciali (Monopolio Intel) e dalle tipologie di applicazioni da eseguire (le applicazioni del Desktop sono diverse dalle applicazioni sullo Smartphone).

La codifica delle istruzioni è chiamata anche linguaggio macchina. Un modo più semplice per esprimere un programma è attraverso l'uso dei linguaggi di programmazione di alto livello. Il compilatore svolge un lavoro di traduzione dal linguaggio ad alto livello al linguaggio macchina costituito da byte. In linea teorica, è possibile scrivere un programma direttamente in linguaggio macchina. In pratica, si ricorre ad un linguaggio di alto livello (C, Java, Pascal, Fortran, etc.) che permette di descrivere le operazioni in un modo più naturale e vicino all'uomo. La creazione del codice in linguaggio macchina è delegata al compilatore.

Velocità del processore

La velocità del processore è determinata dalla frequenza del clock. Il processore compie un progresso ogni ciclo di clock (si veda sotto). La velocità di un processore è quindi proporzionale alla frequenza del clock. La frequenza del clock si misura in Hertz (Hz). Per esempio, in un clock con frequenza di 3 GHz, ci sono 3 x 10^9 cicli di clock ogni secondo.



Ciclo di clock

Il numero di cicli di clock necessari per eseguire un'istruzione varia a seconda di diversi fattori:

- Il tipo di istruzione. Ci sono operazioni che sono intrinsecamente più complesse da svolgere di altre (per esempio, la moltiplicazione è più complessa di una addizione o di un'operazione logica AND , OR , NOT , XOR).
- La tipologia di ISA. Come detto in precedenza, una singola operazione richiede mediamente un numero minore di cicli di clock su architettura RISC rispetto a CISC (in architettura RISC il numero di cicli di clock per operazione varia da 1 a 4, mentre nella CISC oscilla da 2 a 20. Queste informazioni vanno interpretate come ordine di grandezza e potrebbero cambiare con l'evoluzione della tecnologia di realizzazione dei processori).
- La disponibilità di operandi (i dati su cui l'istruzione opera). Istruzioni che operano su dati presenti nei registri sono più veloci rispetto alle stesse che operano su operandi in

memoria, perché l'accesso alla memoria è più lento dell'accesso ai registri.

Il numero di cicli per singola istruzione varia da 2 a 20 su macchine CISC e da 1 a 4 su macchine RISC (questi devono essere considerati ordini di grandezza).

Cache memory

Potenzialmente, il processore potrebbe eseguire istruzioni anche ogni nanosecondo, ma la memoria ha un tempo di accesso di circa 50 nanosecondi. Quindi tutte le istruzioni che richiedono un accesso in memoria sono evidentemente rallentate. Si dice che la memoria costituisce un collo di bottiglia.

L'utilizzo della memoria cache (cache tradotto in Italiano è "nascondiglio") permette di ridurre la perdita di prestazioni dovuta alla maggiore lentezza degli accessi in memoria rispetto agli accessi ai registri. Difatti, la cache memory è una memoria più veloce (tempo di accesso di circa 10 nanosecondi rispetto ai 50 della memoria principale), ma più piccola, collocata fra il processore e il bus di comunicazione. Il suo scopo è quello di contenere dati e istruzioni che vengono usati più frequentemente. In questo modo, almeno per l'accesso ai dati/istruzioni di uso frequente, sarà possibile evitare i tempi lunghi della memoria principale.

Come è possibile sapere quali saranno le informazioni di uso più frequente? Per stimare la frequenza di utilizzo dei dati vengono sfruttati due principi: la località temporale e la località spaziale.

Per località temporale si intende il fatto che se una determinata locazione di memoria è stata acceduta, è molto probabile che essa venga acceduta anche nell'immediato futuro. Per località spaziale, invece, si intende che se una certa locazione di memoria è stata acceduta, allora anche locazioni di memoria limitrofe (ovvero con indirizzi vicini) saranno probabilmente accedute prossimamente. Per esempio, nella fase di fetch del codice, le istruzioni vengono prelevate in sequenza.

Seguendo la metafora della cucina, lo sfruttamento della località temporale equivale a tenere a portata di mano in dispensa gli ingredienti (dati) utilizzati più frequentemente, evitando quindi di prelevarli dal supermercato (memoria) tutte le volte.

Nelle architetture di calcolatori moderne, possono esistere livelli diversi di cache, a seconda della loro vicinanza al processore (che poi determina la velocità nell'accesso). Si parla di cache L1, L2, L3, etc. Al crescere dell'indice del livello della cache, le cache memory diventano sempre più grandi, lontane dalla CPU e lente. Quindi, la cache di livello L1 è la più vicina (al processore) e veloce, ma anche la più piccola.

Abbiamo detto che l'obiettivo della cache (a prescindere dal suo livello) è quello di mantenere le informazioni che si potranno rivelare più utili per l'esecuzione del codice. Se il processore trova all'interno della cache i dati cercati avviene un cache hit e l'accesso al dato avviene in modo più rapido. Altrimenti, se i dati richiesti dal processore non si trovano nella cache avviene un cache miss e i dati devono essere prelevati dalla memoria. Evidentemente, l'accesso ai dati è più veloce se avviene un cache hit rispetto al tempo richiesto nel caso di un cache miss.

Gerarchia di memoria

Abbiamo visto che la cache serve a rendere disponibili i dati utilizzati più frequentemente. Questo principio viene applicato estensivamente a vari livelli dal più basso (e più vicino al processore) al più alto (e più lontano). Per descrivere questa organizzazione di tanti tipi diversi di memoria dalle più vicine alle più lontane si usa il termine di gerarchia di memoria.
In Tabella, sono riportati alcuni valori tipici delle memorie coinvolte nel calcolo.

Tipo di memoria	access time (nsec)	cap. (B)	Costo (€/B)
Registri	1	100	(difficile) 0.1
Cache	10	10^7	(diff.) 10^{-5}
Main memory	50	4×10^9	10^{-8}
disco SSD	10^5	10^{11}	5×10^{-10}
disco HDD	10^7	10^{11}	3×10^{-10}
USB key	10^7	10^{10}	5×10^{-10}
cloud storage	(diff.) 10^8	10^{12}	10^{-10}/anno

Caratteristiche di vari tipi di memoria. I valori riportati hanno puramente un carattere esemplificativo. La differenza tra SSD (Solid-State Drive) e HDD (Hard Disk Drive) è che la seconda è costituita da un vero e proprio disco movimentato da un motore, mentre la seconda è costituita unicamente da componenti elettronici.

Si osservi che anche i registri, i dischi e la capacità di memorizzazione offerta dai data center (cloud storage) si possono considerare come forme di memoria. La tabella elenca i vari tipi di memoria dal più vicino al processore (registri) al più lontano (cloud). Si osservi difatti che i tempi di accesso aumentano all'aumentare della distanza dal processore (i valori sono puramente indicativi e variano al variare della tecnologia impiegata). Non sorprendentemente, anche le capacità tipiche aumentano. Difatti avere memorie molto grandi e molto vicine al processore non è tecnologicamente fattibile. Mentre avere memorie piccole (bassa capacità) e con tempi di accesso grandi è semplicemente inutile. Allo stesso tempo, il costo di ogni singolo byte di memoria diminuisce all'aumentare della distanza dal processore. Un'utile metafora, in questo caso, è dall'osservazione del mercato immobiliare. Difatti, il prezzo al metro quadro degli appartamenti non può che diminuire all'aumentare della distanza dal centro della città dove si svolgono molte attività (assimilabile al processore).

Oltre al tempo di accesso, capacità e costo, hanno un'importanza anche

- il tempo di vita di una memoria, che indica per quanto tempo i dati possono rimanere memorizzati,
- l'affidabilità, che indica il grado di errore che si può incontrare nel rileggere dalla memoria un dato scritto e la capacità di mantenere l'informazione immagazzinata anche in presenza di disturbi (per esempio, di origine elettromagnetica)

Architetture multi-processore

Il processore, le memorie e tutti i circuiti integrati (integrated circuits, ICs, in inglese) sono circuiti elettronici realizzati su silicio. Attraverso a un processo simile alla stampa chiamato litografia, il silicio può diventare sia conduttore che isolante. Questo permette di disegnare circuiti integrati sul silicio. Inoltre, il silicio è disponibile in quantità abbondante e a basso prezzo.

Fino al 2004 circa, l'aumento delle prestazioni della CPU è avvenuto aumentando la densità di integrazione: sulla stessa area di silicio si era in grado di disegnare un numero sempre maggiore di transistor e quindi di realizzare circuiti sempre più complessi e processori sempre più potenti. Lo sviluppo dei processori ha seguito una legge empirica detta Legge di Moore (Moore's law) che determina una crescita esponenziale del numero di transistor

in un circuito integrato. Secondo Moore, il numero di transistor presenti in un circuito integrato raddoppia ogni 2 anni. Nel 2004, il nuovo processore della Intel (Intel Tejas) fu cancellato dallo sviluppo perché era fisicamente impossibile raffreddarlo. Difatti, raggiungeva una densità di calore pari al reattore di un missile.

Dal 2004 in poi si è passati da un'architettura a singolo processore a una multi-processore. Oggi l'aumento delle prestazioni non avviene più migliorando la singola CPU, ma aggiungendo un numero sempre crescente di processori. In questo caso si parla di architetture multi-processore o multi-core.