

1 INTRODUZIONE

1.1 Astrazioni

Per isolare i vari livelli di un computer (hardware, kernel, SO), ogni strato fornisce delle interfacce (o set di istruzioni) con la quale interagire al livello sottostante. Le più importanti sono:

- ISA (Instruction Set Architecture) → insieme istruzioni macchina
- ABI (Application Binary Interface) → interfaccia delle applicazioni

1.2 prestazioni

Le prestazioni si misurano con il **tempo di esecuzione di un programma** che è composto da 3 variabili: **numero di istruzioni**, **cicli di clock per istruzione** e **frequenza di clock**

$$\text{tempo di CPU} = \frac{\text{numero istruzioni} \times CPI}{\text{frequenza di clock}}$$

1.3 ISA

- influisce direttamente sul tempo di cpu (un isa è progettata per frequenze di clock più spinte, è esplicitato il numero di cicli per un'istruzione, e il numero di istruzioni stesso per compiere un'operazione) le ISA che affronteremo:

- RISC-V (RISC): cloud computing e sistemi embedded
- Intel (CISC): PC
- ARM (A-RISC): embedded e mobile

2 ARITMETICA DEI CALCOLATORI

2.1 Basi

L'aritmetica nei calcolatori viene fatta su base binaria, quindi bisogna eseguire delle conversioni da una base all'altra. per un passaggio da decimale a binario, prendo il numero in decimale e calcolo ricorsivamente il modulo di 2, da qui tengo la parte il resto. Quando arrivo a 1, il mio numero in binario saranno i resti letti al contrario.

L'addizione in decimale e binario rimane uguale, mentre le moltiplicazioni seguono questo algoritmo: per ogni 0 del moltiplicatore mi sposto di un posto a sinistra, mentre con un 1 copio il moltiplicando. Alla fine addiziono tutto.

2.2 Codifica

2.2.1 Numeri negativi

Esistono 3 metodi per rappresentare i numeri negativi: **modulo e segno**, **complemento a 1**, **complemento a 2**. Con tutti questi metodi, il bit più a sinistra rappresenta il segno

modulo e segno: il più semplice, con il bit più a sinistra rappresenta il segno (1 per -, 0 per +)

complemento a 1: un numero positivo viene rappresentato come valore assoluto, mentre uno negativo lo rappresento con complemento a 1.

I metodi del complemento sono 2:

- cambio tutti i bit da 1 a 0 e da 0 a 1
- sottraggo il numero a un numero della stessa lunghezza con tutti i bit a 1 (11111 - 00101 = 11010)

Somma e sottrazione con complemento a 1: i numeri negativi li rappresento con complemento a 1 e poi sommo i 2 numeri. Il riporto sul bit più significativo lo sommo al risultato.

$$6-3 \rightarrow 6 + (-3) \rightarrow 00110 + 11100 \rightarrow 00010 \text{ con riporto di } 1 \rightarrow 00010 + 1 =$$

$$00011 = 3$$

complemento a 2: per eseguire un complemento a 2 si possono usare 3 metodi:

- metodo 1: dato un numero trovo il primo bit a 1 partendo da destra, poi eseguo il complemento a 1 per tutti i bit successivi
- metodo 2: eseguo il complemento a uno del numero, e poi addiziono 1
- metodo 3: prendo il numero con primo bit di sinistra a 1 e resto a zero e di lunghezza pari al numero che ci interessa, infine sottraggo quest'ultimo

Il complemento a 2 risulta più conveniente per le somme, la codifica dello zero è unica e nell'operazione inversa, infatti per eseguire quest'ultima si osserva il primo bit. Se è pari a 0, allora converto normalmente, se no eseguo il complemento a 2 e sommo 1, poi converto in base 10

Il vantaggio con il complemento a 2 si guadagna un valore negativo. Sfortunatamente però si possono anche generare degli overflow. Bisogna sempre controllare se da somma di positiva risulta in negativo o se da somma di negativi risulta un positivo

2.2.2 Rappresentazione numeri reali

virgola fissa - si dedica una parte della stringa di bit come parte intera e parte frazionaria - genericamente trattiamo il numero come intero e poi moltiplichiamo per $-n$ (dove n rappresenta le cifre decimali) - non ci sono errori di approssimazione, ma risulta difficile gestire numeri particolarmente grandi o piccoli - per la conversione della parte decimale moltiplico per 2 e considero le parti intere

virgola mobile - un numero reale viene suddiviso come mantissa ed esponente