

## PIPELINE

La pipeline è un meccanismo di parallelismo a livello di istruzioni. In pratica ogni istruzione è divisa in  $n$  fasi, e il processore ha  $n$  componenti separate, ognuna delle quali lavora su una fase di un'istruzione.

Ogni componente può lavorare in parallelo: le istruzioni sono, per così dire, in fila.

Esempio: sulla prima istruzione viene eseguita la prima fase. Al ciclo di clock successivo, sulla prima istruzione viene eseguita la seconda fase, mentre sulla seconda istruzione viene eseguita la prima fase. Al ciclo successivo sulla prima istruzione viene eseguita la terza fase, sulla seconda la seconda fase e sulla terza la prima fase, e così via.

La latenza è il tempo di esecuzione di una istruzione, ed è data da  $n \cdot T$ , dove  $n$  è il numero delle fasi e  $T$  è la durata di un ciclo di clock (calcolata in nanosecondi).

L'ampiezza di banda misura quante istruzioni vengono effettuate in un secondo (viene calcolato in MIPS, Milioni di Istruzioni Per Secondo), ed è data da  $1000/T$ :

infatti considerando che ad ogni ciclo viene terminata un'istruzione, dividendo  $1 \text{ sec} = 10^9 \text{ nsec}$  per la durata di un ciclo ( $T$ ) ottengo quante istruzioni vengono eseguite al secondo. Dividendo per un milione ( $10^6$ ) ottengo i MIPS. Perciò  $(10^9/10^6)/T = 1000/T$ .

Esempio:  $T=20 \text{ nsec}$ . L'ampiezza di banda è  $1000/20=50 \text{ MIPS}$ .

Le architetture superscalari sono architetture in cui vengono avviate più istruzioni alla volta e in cui ci sono più pipeline. È possibile anche parallelizzare lo stadio più lento della pipeline.

Parallelismo a livello di processori, ovvero si aumenta il numero di CPU nel calcolatore. Tipi di approcci:

-Data Parallelism. SIMD (Single Instruction Multiple Data):

- Processore Matriciale (Array) = molti processori eseguono le stesse istruzioni su insieme di dati diversi
- Processore Vettoriale = le unità aritmetiche lavorano su registri vettoriali

-Task Parallelism. MIMD (Multiple Instructions Multiple Data):

- Multiprocessore = più CPU lavorano indipendentemente e hanno una memoria privata e una condivisa tra tutti.
- Multicomputer = più PC o Workstation
- Multicore = ogni processore è dotato di più core, con cache privata e/o condivisa.

### Esercizio

**Si consideri un microprocessore con architettura RISC, 10 stadi di pipeline e un clock di 2.5 Ghz e si supponga di lavorare in condizioni ideali. Rispondere alle domande che seguono.**

- **Indicare il tempo di esecuzione (in nanosecondi) di una singola istruzione macchina.**

$$T = 1000/2,5 \text{ Ghz} = 1000 / 2500 \text{ Mhz} = 0,4 \text{ nsec.}$$

Una singola istruzione macchina impiega  $n \cdot T = 10 \cdot 0,4 = \underline{4} \text{ nsec}$

- **Indicare ogni quanto tempo (in nanosecondi) si completa a regime una istruzione macchina.**

A regime viene terminata un'istruzione ogni  $T$ . Quindi 0,4 nsec.

- **Indicare l'ampiezza di banda del microprocessore (espressa in MIPS).**

$$\text{Ampiezza di banda} = 1000/T = 1000/0,4 = \underline{2500} \text{ MIPS}$$

- **Se la frequenza di clock scende a 2 Ghz il tempo di esecuzione di una istruzione diminuisce.**

Il tempo di esecuzione aumenta con il diminuire della frequenza. Infatti  $T = 1000/2000 = 0,5 \text{ nsec}$  e un'istruzione viene eseguita in  $n \cdot T = \underline{5} \text{ nsec}$ .

- **Indicare il tempo di esecuzione (in nanosecondi) di un programma macchina composto da 5 istruzioni.**

La prima istruzione viene eseguita in 4 nsec, le successive 4 in 0,4 nsec. Quindi  $4 + 0,4 \cdot 4 = \underline{5,6} \text{ nsec}$ .

- **Indicare di quanti nanosecondi si riduce il tempo di esecuzione di un'istruzione se nella pipeline si eliminano cinque stadi.**

Con  $n = 5$ ,  $n \cdot T = 2 \text{ nsec}$ . Prima  $T$  era 4 nsec, quindi il tempo di esecuzione di un'istruzione si è ridotto di 2 nsec.

- **Indicare il valore della banda (espressa in MIPS) se nella pipeline vengono aggiunti cinque stadi.**

La banda non cambia con il variare degli stadi: resta sempre 2500 MIPS.

## DISTANZA DI HAMMING

La distanza di hamming tra 2 codifiche è il numero di bit diversi. Es. 0101 e 1011 →  $h=3$

La distanza di hamming di un codice è la distanza più piccola tra due codifiche di quel codice.

Es.

0000000000 La distanza di hamming più piccola tra questa e le successive 3 codifiche è 5.

0000011111 La distanza di hamming più piccola tra questa e le successive 2 codifiche è 8.

1111100000 La distanza di hamming tra questa e la successiva codifica è 2.

1111111000 Segue che la distanza di hamming del codice è 2.

Con una distanza di hamming  $h$  posso rilevare fino a  $h-1$  errori e posso correggere fino a  $(h-1)/2$  errori.

Es. con  $h=5$  posso rilevare 4 errori e correggerne 2. Con 8 posso rilevare 7 errori e correggerne 3 (si arrotonda per difetto).

Per rilevare un errore singolo basta 1 solo bit di parità. Più errori si vogliono correggere più bit di parità saranno necessari. La percentuale dei bit di parità necessari in una parola diminuisce con l'aumentare della dimensione della parola.

## CACHE HIT RATIO

Se una parola viene letta  $k$  volte,  $k-1$  volte sarà presente in cache.  $H(\text{it}) = \text{Cache Hit Ratio}$ .

$$H = (k-1)/k$$

$A(\text{ccesso}) =$  tempo medio di accesso a memoria.  $C =$  tempo di accesso alla cache.  $M =$  tempo di accesso alla memoria.

$$A = C + (1-H)*M$$

### Esercizio 1

Si consideri un programma che utilizza le variabili  $X$  e  $Y$ . Complessivamente,  $X$  viene usata in 5 istruzioni,  $Y$  in 8. Il programma viene eseguito su un microprocessore a 32 bit che dispone di una cache con tempo di accesso di 2 nanosecondi e una memoria con tempo di accesso di 40 nanosecondi. Si assuma che i trasferimenti tra memoria a cache avvengano per blocchi di 128B. Rispondere alle domande che seguono.

- Indicare la percentuale di successo nell'accesso alla cache (cache hit ratio) per la variabile  $X$

Si accede 5 volte a  $X$ . Perciò  $H = (5-1)/5 = 0,8 = \underline{80\%}$

- Indicare la percentuale di successo nell'accesso alla cache per la variabile  $Y$

Si accede 8 volte a  $Y$ . Perciò  $H = (8-1)/8 = 0,875 = \underline{87,5\%}$

- Indicare il tempo medio di accesso alla variabile  $X$ , espresso in nanosecondi.

Il tempo medio di accesso è  $A = 2 + (1-0,8)*40 = \underline{10}$  nsec

- Indicare il tempo necessario al terzo accesso alla variabile  $X$ , espresso in nanosecondi.

Al terzo accesso la variabile  $X$  è sicuramente nella cache, quindi ci impiega 2 nsec.

- Nell'ipotesi che  $X$  e  $Y$  siano allocate in locazioni consecutive della memoria, indicare la percentuale complessiva di successo del programma nell'accesso alla cache (cache hit ratio).

Quando accedo ad  $X$  la prima volta, porto in cache l'intero blocco che contiene sia  $X$  che  $Y$  (essendo in locazioni consecutive). Perciò basta sommare il numero di accessi di entrambe le variabili.

$$H = [(5+8)-1]/(5+8) = 12/13 = 0,92 = \underline{92\%}$$

- Indicare il tempo necessario al primo accesso a una variabile del programma, espresso in nanosecondi.

Si cerca prima nella cache (2 nsec). Poiché è il primo accesso non si troverà la variabile, perciò si accede alla memoria (40 nsec). Tempo di accesso =  $2 + 40 = \underline{42}$  nsec.

## Esercizio 2

Si consideri un semplice programma che calcola la somma degli elementi di un array A composto da 10 elementi di 1 byte memorizzati in locazioni contigue della memoria principale.

Il programma viene eseguito su un microprocessore a 32 bit che dispone di una cache con tempo di accesso di 2 nsec e una memoria con tempo di accesso di 15 nsec.

Si assuma che i trasferimenti tra memoria e cache avvengano per blocchi di 16B e che la somma di due elementi sia eseguita dal microprocessore in 1 nsec.

- Indicare il cache hit ratio (percentuale di successo di ricerca in cache) e il tempo medio di accesso al primo elemento dell'array;

Accedo una sola volta al primo elemento dell'array. Quindi  $H = (1-1)/1 = 0\%$  e il tempo medio è quindi  $A = 2 + (1-0) * 15 = 17$  nsec

- Indicare il cache hit ratio complessivo (la percentuale globale di successo nell'accesso alla cache) e il tempo medio di accesso alla memoria del programma;

Accedo una volta ad ogni elemento dell'array, quindi 10 volte.  $H = 9/10 = 90\%$ . Quindi il tempo medio è  $A = 2 + (0,1) * 15 = 3,5$  nsec.

- Indicare il tempo complessivo necessario all'esecuzione del programma.

Il primo elemento dell'array lo cerco prima in cache e poi, non trovandolo, in memoria; quindi impiega  $2+15=17$  nsec. Quando porto in cache il primo elemento, trasferisco l'intero blocco di 16B che lo contiene, compresi tutti i 10 byte dell'array (ogni elemento è un byte come detto sopra). Quindi quando accedo ai 9 restanti elementi dell'array, che ormai sono in cache, impiega 2 nsec per ogni accesso, perciò  $2*9=18$  nsec.

Si effettuano 9 operazioni di somma che impiegano ognuna 1 nsec, quindi in totale 9 nsec.

Il tempo per l'esecuzione del programma  $17 \text{ nsec} + 18 \text{ nsec} + 9 \text{ nsec} = 44 \text{ nsec}$ .

## RAID

I RAID (Redundant Array of Inexpensive Disks) sono composti da una scatola piena di dischi SCSI e un controllore SCSI RAID che appaiono al sistema operativo come un unico disco fisso. Le strip di dati consecutive vengono memorizzate in dischi diversi per consentire un accesso parallelo alla memoria secondaria, e vengono eventualmente copiate in altri dischi per recuperare le informazioni in caso di guasti. Esistono 6 livelli di RAID:

Livello 0, si effettua lo striping (divisione delle strip consecutive in diversi dischi) ma non la ridondanza.

Livello 1, si effettua lo striping e per ogni disco c'è una sua copia.

Livello 2, invece delle strip, si distribuiscono in ogni disco i bit delle parole di dati.

Livello 3, come il 2, ma in più c'è un disco (unità di parità) che contiene i bit di parità delle parole di dati.

Livello 4, si effettua lo striping, e c'è un'unità di parità che contiene le strip di parità dei dati

Livello 5, come il 4, solo che le strip di parità non sono su un unico disco, ma vengono distribuite tra le altre.

Per calcolare il numero di strip in un disco basta dividere la memoria di un disco per la dimensione di una strip. Es. in un disco da 128GB con strip da 512KB, ci sono  $128*1024/512 = 262144$  strip.

## Esercizio RAID

Sapendo che:

Al livello 0, tutto lo spazio utilizzabile viene distribuito tra i dischi.

Al livello 1, tutto lo spazio utilizzabile viene raddoppiato e distribuita tra i dischi.

Al livello 2, si sceglie un numero  $n$  di dischi per i dati, e il numero  $r$  di dischi di copia deve rispettare la formula  $(n+r+1) < 2^r$ . Es. prendiamo 4 dischi,  $(4+r+1) < 2^r$ . Con  $r=3$  la formula è rispettata.

Al livello 3, si fa come il due. Come per il 2 se si conosce in quale disco c'è stato errore è possibile correggerlo.

Al livello 4, tutto lo spazio utilizzabile viene distribuito tra tutti i dischi meno uno.  
Al livello 5, come il 4.

**Si consideri un'unità disco RAID di 1TB (spazio utilizzabile di memoria fisica) e con blocchi (strip) di 256 KB.**

- **In un RAID di livello 0 con 2 dischi, quanto è grande ogni disco?**

A livello 0 si divide semplicemente la memoria totale per il numero di dischi  $1TB/2 = 1024GB/2 = \underline{512GB}$  a disco.

- **In un RAID di livello 1, quanto deve essere grande la memoria totale?**

A livello 1 si ha una copia di ogni disco, perciò basta moltiplicare la memoria fisica per 2.  $1TB*2 = \underline{2TB}$

- **In un RAID di livello 4 con 5 dischi, quanto è grande ogni disco?**

A livello 4 la memoria fisica si divide tra tutti i dischi meno uno (quello per la parità).  $1TB / (5-1) = \underline{256GB}$

- **In un RAID di livello 2, se si rompe un disco con i bit di controllo è possibile recuperare i dati.**

Vero

- **In un RAID di livello 5 con 9 dischi, quanto è grande ogni disco?**

Come per il livello 4.  $1024GB / (9-1) = \underline{128GB}$

- **Se ho dischi da 512GB, di quanti dischi ho bisogno per realizzare un RAID di livello 5?**

Divido la memoria fisica per la dimensione di un singolo disco, per ottenere il numero di dischi meno uno (quello di parità).  $1024GB / 512GB = 2$  dischi. Aggiungiamo il disco di parità  $2+1=\underline{3}$ .

- **Il RAID di livello 0 se si rompe un disco perdo i dati.**

Vero