# Módulo 4
# Hierarquia de Memória: Desempenho e Organização

## Avaliação do Desempenho

$$T_{exec} = CPI * \#I / f$$
$$CPI = CPI_{CPU} + CPI_{MEM}$$
$$CPI_{MEM} = (mr_I + mr_D * \%Mem) * mp$$

**Exercício 1** - Considere um programa com as características apresentadas na tabela 1, executado numa máquina com uma frequência do relógio de 2 GHz. Note que os valores apresentados correspondem ao que normalmente designamos por $CPI_{CPU}$ para cada classe de instruções. Note também que a alínea i+1) refere-se sempre à máquina descrita na alínea i), com as modificações propostas. Por exemplo, na alínea c) deve considerar os tempos de acesso à memoria principal da alínea b).

| Tipo de instrução | Nº Instruções | $CPI_{CPU}$ |
|---|---|---|
| Operações inteiras | $6*10^8$ | 1 |
| Acessos à memória | $12*10^8$ | 1 |
| Operações FP | $2*10^8$ | 3 |

**Tabela 1 - Distribuição das instruções e CPI**

a) Considere que a máquina tem uma *cache* infinita (isto é, não há *cache misses*, todos os dados e código estão sempre na *cache*, logo $mr_I = mr_D = 0$). Qual o CPI médio e o tempo de execução deste programa?

$$CPI = CPI_{CPU} = 0.3*1 + 0.6*1 + 0.1*3 = 1.2$$
$$T_{exec} = 2*10^9 * 1.2 / 2*10^9 = 1.2s$$

b) Suponha agora o mesmo programa a executar numa máquina sem *cache* (logo $mr_I = mr_D = 1$). Os acessos à memória central são realizados em blocos de 4 palavras, sendo necessários 60 ns para iniciar a transferência e 10 ns adicionais por cada palavra transferida. Qual o CPI médio e o tempo de execução?

$$\%Mem = 60\%;\ mp_T = 60 + 10*4 = 100 \text{ ns};\ mp_{cc} = 100*10^{-9}*2*10^9 = 200 \text{ cc}$$
$$CPI_{MEM} = (1 + 1*0.6)*200 = 320;$$
$$T_{exec} = \frac{2E9*(1.2+320)}{2E9} = 321.2 \text{ s}$$

c) Se à máquina da alínea anterior for acrescentado um nível de memória *cache*, exibindo uma *miss rate* de acesso às instruções de 8% e de acesso aos dados de 10%, qual o CPI médio e o tempo de execução do programa? Qual o ganho relativamente à alínea anterior?

$$CPI_{MEM} = (0.08 + 0.1 * 0.6) * 200 = 28;$$
$$T_{exec} = \frac{2E9 * (1.2 + 28)}{2E9} = 29.2 \text{ s}$$

d) Suponha que a capacidade da *cache* é aumentada para o dobro, resultando numa *miss rate* de 4.8% para as instruções e 7% para os dados. Este aumento de capacidade resulta também num aumento do tempo de acesso à *cache (hit time)*, implicando um aumento de 25% do $CPI_{CPU}$. Qual o CPI médio e o tempo de execução do programa?

$$CPI_{MEM} = (0.048 + 0.07 * 0.6) * 200 = 18;$$
$$CPI_{CPU} = 1.2 * 1.25 = 1.5$$
$$T_{exec} = \frac{2E9 * (1.5 + 18)}{2E9} = 19.5 \text{ s}$$

e) Para tirar partido da localidade espacial aumentou-se, na máquina anterior (alínea d) ), o número de palavras por linha da *cache* de 4 para 8, reduzindo a *miss rate* de instruções para 3% e de dados para 5%. Qual o CPI médio e o tempo de execução do programa?

$$mp_T = 60 + 10 * 8 = 140 \text{ ns}; \quad mp_{cc} = 140 * 10^{-9} * 2 * 10^9 = 280 \text{ cc}$$
$$CPI_{MEM} = (0.03 + 0.05 * 0.6) * 280 = 16.8;$$
$$T_{exec} = \frac{2E9 * (1.5 + 16.8)}{2E9} = 18.3 \text{ s}$$

f) Para reduzir a *miss penalty* a memória principal da máquina anterior foi substituída por outra mais rápida, com uma latência de 50ns e 7.5ns por palavra. Qual o CPI médio e o tempo de execução do programa?

$$mp_T = 50 + 7.5 * 8 = 110 \text{ ns}; \quad mp_{cc} = 110 * 10^{-9} * 2 * 10^9 = 220 \text{ cc}$$
$$CPI_{MEM} = (0.03 + 0.05 * 0.6) * 220 = 13.2;$$
$$T_{exec} = \frac{2E9 * (1.5 + 13.2)}{2E9} = 14.7 \text{ s}$$

g) O processador da máquina foi substituído por outro com uma frequência de 3 GHz, mantendo-se constantes todos os outros parâmetros do sistema. Qual o CPI médio e o tempo de execução do programa? Qual o ganho relativo à máquina anterior? Comente esse resultado em termos do ganho obtido relativamente ao aumento da frequência.

$$mp_T = 50 + 7.5 * 8 = 110 \text{ ns} ; \; mp_{cc} = 110 * 10^{-9} * 3 * 10^9 = 330 \text{ cc}$$
$$CPI_{MEM} = (0.03 + 0.05 * 0.6) * 330 = 19.8;$$
$$T_{exec} = \frac{2E9 * (1.5 + 19.8)}{3E9} = 14.2 \text{ s}$$
$$\text{Ganho} = \frac{14.7}{14.2} = 1{,}035$$

**Exercício 2 -** A tabela abaixo apresenta na coluna da esquerda uma sequência de endereços (m=5) de acesso à memória gerados por um determinado programa. As três colunas seguintes representam 3 diferentes modos de mapeamento numa *cache* que usa o algoritmo de substituição LRU. Preencha estas colunas indicando em que *set*/linha (dentro do *set*) mapeia cada endereço e indicando se se trata de um *cold miss*, colisão ou de um *hit* (veja o exemplo na 1ª linha). Considere a *cache* inicialmente fria. Finalmente indique na última linha a *miss rate* observada.

**Sugestão:** Para cada organização mantenha um registo do valor da *tag* para cada linha.

| Addr | (S=4, E=1,B=2,m=5) | (S=1, E=4,B=2,m=5) | (S=2, E=2,B=2,m=5) |
|---|---|---|---|
| 5 (00101) | *Set* 2 / Linha 0 (*cold miss*) | *Set* 0 / Linha 0 (*cold miss*) | *Set* 0 / Linha 0 (*cold miss*) |
| 14 (01110) | Set 3 / L 0 (cold miss) | Set 0 / L 1 (cold miss) | Set 1 / L 0 (cold miss) |
| 10 (01010) | Set 1 / L 0 (cold miss) | Set 0 / L 2 (cold miss) | Set 1 / L 1 (cold miss) |
| 29 (11101) | Set 2 / L 0 (colisão) | Set 0 / L 3 (cold miss) | Set 0 / L 1 (cold miss) |
| 4 (00100) | Set 2 / L 0 (colisão) | Set 0 / L0 (hit) | Set 0 / L 0 (hit) |
| 21 (10101) | Set 2 / L 0 (colisão) | Set0 / L1 (LRU; colisão) | Set0 / L1 (LRU; colisão) |
| 16 (10000) | Set 0 / L0 (cold miss) | Set0 / L2 (LRU; colisão) | Set0 / L0 (LRU; colisão) |
| 5 (00101) | Set 2 / L 0 (colisão) | Set 0 / L 0 (hit) | Set0 / L1 (LRU; colisão) |
| Miss rate: | 8/8 = 100% | 6/8 = 75 % | 7/8 = 87.5 % |