



Faculdade de Computação

Arquitetura e Organização de Computadores 1

2º Homework – 5 pontos

Prof. Cláudio C. Rodrigues

Organização Hierárquica de Memória:

- P1)** Um dos objetivos que orientam o projeto de um sistema de memória é alcançar capacidade de armazenagem, com desempenho aceitável a um custo razoável. Como este objetivo pode ser alcançado? Faça um esboço desta estrutura?
- P2)** o acesso à memória é um enorme gargalo em todos os computadores modernos porque o tempo de ciclo da CPU é muito superior ao tempo de acesso à memória. A Figura 1 mostra a performance da memória e da CPU no período 1980-2010. Dessa forma, uma das maneiras de melhorar o desempenho de computadores atuais é evitar acessar à memória principal. O tempo de acesso à memória principal é alto e degrada o desempenho do sistema computacional. A estratégia de usar memória cache tenta suprir essa lacuna existente no desempenho relativo CPU/Memória e assim o uso de memória cache tornou-se um dos principais temas a ser analisado no projeto de um sistema computacional.

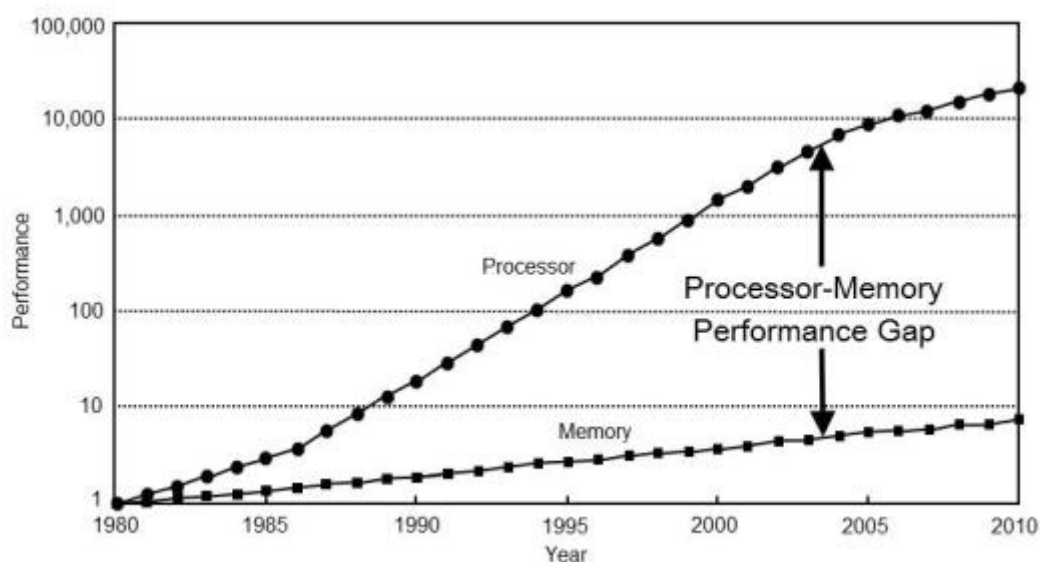


Figura 1-CPU and Memory performance gap.

Considerando tais informações e seus conhecimentos a respeito do projeto de memória cache, responda as questões a seguir:

- a) apresente uma discussão mostrando os *trade-offs* envolvidos nos seguintes aspectos relacionados ao projeto de memória cache: tamanho da cache, tamanho da linha e associatividade. Sugestão: faça uma tabela mostrando como esses aspectos impactam no tempo de acesso, na penalidade por falta e na taxa de acerto. Escreva um parágrafo explicando a tabela.
- b) As CPU's modernas apresentam pelo menos dois níveis de cache. Considerando uma CPU com dois níveis de cache, com cache L1 (nível 1) e a cache L2 (nível 2). O tamanho da cache L2 é maior que o tamanho da cache L1. Apresente duas justificativa para a cache L2 ser maior que a cache L1.
- P3)** Qual é o significado dos termos **byte-endereçável** ou **palavra-endereçável**?
- P4)** O que significam **princípio da localidade da referência** e **princípio da inclusão** em organização hierárquica de memória?
- P5)** Uma medida dos benefícios de diferentes organizações de cache é a taxa de falta (*miss rate*). Onde taxa de falta é a fração de acessos a cache que resultam em faltas. Denomine e descreva quais são as três categorias associadas as **causas** de falta de cache (*miss cache*):

- P6)** Um sistema de **cache** tem uma taxa de acerto de 75%, um tempo de acesso de 10ns quando o dado for encontrado no **cache** e um tempo de acesso de 100ns se o **cache** não contiver o dado. Qual é o tempo de acesso efetivo?
- P7)** Por que é tão difícil construir um *cache full associative*?
- P8)** Para esta tarefa, você deve assumir que a memória principal consiste em 128 palavras (*words*) dividida em blocos de 4 palavras cada (32 blocos). A memória *cache* consiste em 8 linhas *cache* (*slots*).
- Desenhe um diagrama de blocos, similar aquele mostrado em aula, que ilustra esta configuração de memória.
 - Assumindo que uma função de mapeamento direto é utilizada
 - Para cada slot no cache, liste os blocos que seriam mapeados para o slot.
 - Quantos bits são necessários para um endereço? Liste os bits e indique qual parte do endereço que seriam utilizados para (palavra, etiqueta, etc.).
 - Qual é o endereço para as seguintes palavras: 4, 13, 39 e 89.
 - Assumindo que uma função de *mapeamento associativo* é utilizada
 - Para cada slot no cache, liste os blocos que seriam mapeados para este slot (atenção)
 - Quantos bits são necessários para um endereço? Liste os bits e indique qual parte do endereço que seriam utilizados para (palavra, etiqueta, etc.).
 - Qual é o endereço para as seguintes palavras: 5, 11, 29 e 99.
 - Mostre o estado do cache após a CPU ter requerido as seguintes palavras, se o algoritmo de substituição de blocos *Menos Recentemente Utilizado* (LRU) for empregado: 3, 7, 15, 16, 17, 28, 30, 56, 57, 58, 78, 79, 20, 21, 23, 98, 99, 24, 25 e 26.
- P9)** Escreva a fórmula do tempo médio de acesso a memória (**Average Memory Access Time - AMAT**) quando há dois níveis de cache L1 e L2.
- P10)** Por que um cache de segundo ou terceiro nível é **útil**? Que tipo de benefícios proporciona ao desempenho do sistema?
- Explique como isso afeta o cálculo do tempo médio de acesso à memória (**AMAT**).
 - Quais outras técnicas de aprimoramento de cache podem fornecer uma solução para o mesmo problema?
- P11)** Vamos considerar um computador com um cache L1 e uma hierarquia de memória cache L2. Suponha que em 1000 referências de memória haja 80 erros em L1 e 40 em L2.
- Quais são as consequentes taxas de falta de cache?
 - Suponha que o tempo de acerto L1 = 1 ciclo de clock; Tempo de acerto L2 = 20 ciclos de clock; Miss Penalty L2 = 200 ciclos de clock; Acessos a memória por instrução = 150%. Qual é o tempo médio de acesso à memória (AMAT)?
- P12)** O tempo médio (esperado) para acessar a memória (AMAT - *Average Memory Access Time*), pode ser calculado pela seguinte fórmula:
- $$\text{AMAT} = \text{hit time} + (\text{miss rate} \times \text{miss penalty})$$
- Lembre-se que a taxa de penalidade (*miss penalty*) é o tempo adicional que é consumido para acessar a memória em um evento de falta de cache (*cache miss*). Por essa razão, uma falta de cache é calculada como (*hit time + miss penalty time*).
- Considere um sistema de memórias cache com as seguintes propriedades. Responda, qual é o tempo de acesso médio a esse sistema de memória (AMAT)?
- L1\$ hits in 1 cycle (local miss rate 45%)
 - L2\$ hits in 10 cycles (local miss rate 30%)
 - L3\$ hits in 50 cycles (global miss rate 10%)
 - Memória principal hits in 100 cycles (always hits)

P13) Um **cache** está sendo projetado para um computador com 2^{32} bytes de memória. O **cache** terá 2K slots e usará um bloco de 32 bytes. Calcule, tanto para um **cache associativo** quanto para um **cache com mapeamento direto**, quantos bytes o **cache** precisará ter capacidade de armazenar.

P14) Projete e desenhe um diagrama detalhado de uma memória cache com capacidade de 1 Mbytes, mapeamento direto (associatividade unária), 8 palavras por bloco e escrita preguiçosa. O processador emite endereços de 32 bits.

a) Indique como um endereço é interpretado pelo controlador da cache.

b) Repita o exercício, considerando associatividade octa-ária (**8-way set-associative**). Indique o mecanismo de substituição de blocos num mesmo conjunto.

P15) Apresente quais técnicas poderíamos empregar no projeto de organizações hierárquicas de memória para reduzir a penalidade de uma falta (**miss penalty**)?

P16) Considere o seguinte código

```
for (i=0; i<20; i++)  
    for ( j=0; j< 10; j++)  
        a[i] = a[i]*j;
```

a) Dê um exemplo de localidade espacial no código.

b) Dê um exemplo de localidade temporal no código.

P17) Considere o programa de multiplicação de matrizes abaixo. As matrizes contêm 1024x1024 elementos, cada elemento um *double* (8 bytes). O programa é executado num único processador, num sistema de memória virtual com páginas de 4 Kbytes. Existe uma cache primária com 64 Kbytes e uma cache secundária com 1 Mbytes. Os blocos de cache primária tem 32 bytes de largura, e os blocos da cache secundária 64 bytes. Descreva o comportamento da hierarquia de caches durante a execução deste programa.

```
for (i = 0; i < 1024; i++) {  
    for (j = 0; j < 1024; j++) {  
        sum = 0.0;  
        for (k = 0; k < 1024; k++)  
            sum += a[i][k] * b[k][j];  
        c[i][j] = sum;  
    }  
}
```

P18) Considere duas organizações de memória cache. A **primeira** é estruturada como um **cache de 32 KB, 2-way set associative, blocos de 32-bytes**. A **segunda** organização é estruturada como um **cache de 32 KB , mapeamento direto, blocos de 32-bytes**. O tamanho do **endereço** de ambas é de **32 bits**. Considere que um multiplexador 2/1 tenha uma latência de 0.6ns, enquanto um comparador de kbits tem uma latência de k/10ns. A latência de um **cache hit** (acerto) na organização associativa por conjuntos é definida por **H1**, enquanto a latência do **cache hit** da organização com mapeamento direto é definida por **H2**. Determine o valor de **H1**.

P19) Considere o fragmento de código a seguir, armazenado na memória a partir do endereço 0x1000100C. Todas as instruções na máquina alvo possuem 4 bytes de tamanho:

```
loop:  
    lw  $r2 , 0( $r0 )  
    addi $r3 , $r2 , 20  
    sw  $r3 , 0( $r1 )  
    addi $r0 , $r0 , 4  
    addi $r1 , $r1 , 4  
    bnez $r2 , loop
```

Este fragmento de código executará em uma máquina com a seguinte organização de memória: **Cache L1** dividido (splited), ambos com **32 KB** e política de mapeamento **2-ways set-associative**. **Cache L2** unificado, com **1 MB** e política de mapeamento **8-ways set-associative**. Nos dois casos o tamanho do **bloco** é de **32 bytes**. Assumir que o **tempo de acerto** (*cache hit*) no **cache L1** consome **4 ciclos**, o tempo de acerto no **cache L2** consome **14 ciclos** e a **penalidade** para transferir um bloco da memória principal (DRAM) para o **cache L2** é de **80 ciclos**. Todos os caches implantam a política de escrita **write-back**.

Inicialmente, os registradores armazenam os seguintes valores:

\$r0: 0x00010000.

\$r1: 0x00080000.

Começando na posição 0x00010000 todos os valores na memória são diferentes de zero até a posição 0x000100FC. Na posição de memória 0x000100FC está armazenado o valor zero.

- a) Determine qual deve ser o Tempo Médio de Acesso a Memória (AMAT) assumindo que um determinado programa (diferente do mostrado acima) realiza em média 2 acessos de dados por instrução e apresenta a seguinte taxa de falha (**miss rate**).

L1 instructions: 10% - L1 data: 5% - L2: 2%

- b) Determine o número de faltas de cache (**miss cache**), nos módulos *data L1 cache, instruction L1 cache* e *L2 cache*, produzidos durante a execução do fragmento de código apresentado acima.

P20) Códigos de Detecção e Correção de Erros:

- a) Aplique a **codificação Hamming** para armazenar a seguinte palavra de 8 bits "**10011010**". Utilize a tabela abaixo para apresentar a codificação resultante.

	12	11	10	9	8	7	6	5	4	3	2	1
Posição	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data Bits												
Check Bits												
Código												

- b) A **codificação Hamming** foi utilizada para armazenar o seguinte código de 12 bits "**010101100011**". Verifique se o código está correto, considerando que ele foi criado usando a codificação de paridade par de *Hamming*. Se o código estiver incorreto, apresente o código correto e qual é a palavra de dados original.