

Soluções da Lista 3

Arquitetura e Organização de Computadores

Junho de 2024

Tema: memória cache

Para as questões 1-6, considere uma cache, até então sem dados validos, com 16 conjuntos, 2 vias por conjunto, 8 palavras por via, palavras e endereços de 16 bits, write-back como estratégia de escrita e *Menos Recentemente Usada* como política de substituição.

1. O campo de índice tem quantos bits nesta cache? $\log_2 16 = 4$
2. O campo de tag tem quantos bits nesta cache? $16 - 4 - 3 - 1 = 8$
3. Qual o tamanho, em bits, da cache? $16 \text{ bit/palavra} * 8 \text{ palavra/via} * 2 \text{ via/conjunto} * 16 \text{ conjunto} + 8 \text{ bit/tag} * 1 \text{ tag/via} * 2 \text{ via/conjunto} * 16 \text{ conjunto} + 1 \text{ bit_validade/via} * 2 \text{ via/conjunto} * 16 \text{ conjunto} + 1 \text{ bit_escrita/via} * 2 \text{ via/conjunto} * 16 \text{ conjunto} + 1 \text{ bit_uso/conjunto} * 16 \text{ conjunto} = 4432 \text{ bits}$

Para as questões 4 a 6, considere que o processador acessou, nesta ordem, os endereços 0x00F0(L), 0x00F1(L), 0x0010(E), 0x1018(L), 0x2018(E), 0x3018(E), 0xFFFF0(L), 0xFF00(L). L → Leitura, E → Escrita.

4. Quantas requisições foram feitas à memória principal? 0x00F0(L), 0x0010(L), 0x1018(L), 0x0010(E), 0x2018(L), 0x3018(L), 0xFFFF0(L) e 0xFF00(L): 8 requisições.
5. Quais endereços são escritos na memória principal? 0x0010, porque, após sua requisição, são acessados mais blocos com o mesmo índice, 0001, do que o conjunto suporta armazenar. 0x1010 não é escrito na memória principal porque seu conteúdo não foi alterado na cache.
6. Qual o valor armazenado na tag do endereço 0xBCDE? 10111100
7. Para uma cache de tamanho fixo, qual a consequência do aumento do tamanho de suas vias? Mais palavras são armazenadas na cache a cada requisição à memória principal, o que beneficia a localidade espacial dos programas. Em contrapartida, ou a quantidade de conjuntos, ou a quantidade de vias será reduzida, o que prejudica programas com maior localidade temporal, já que há maior possibilidade de um endereço recentemente acessado ser substituído.
8. Para uma cache de tamanho fixo, qual a consequência do aumento em sua quantidade de vias? Mais endereços com um mesmo índice poderão ser mantidos na cache concomitantemente. Isto beneficia a localidade temporal, já que a probabilidade de um endereço recentemente acessado ser substituído na cache é reduzida. Por outro lado, se este aumento sacrificar o tamanho das vias, a localidade espacial é prejudicada, já que menos palavras adjacentes são transferidas em cada acesso à memória principal.
9. Uma cache que sacrifique quantidade de vias por um maior tamanho de via beneficia qual tipo de localidade? Espacial.
10. Uma cache que sacrifique tamanho de via por uma maior quantidade de vias beneficia qual tipo de localidade? Temporal.

11. Como aumentar a localidade espacial do programa a seguir?

```
int main() {
    int i, j;
    int m[8][8], a[8][8], b[8][8];

    for(j = 0; j < 8; j++)
        for(i = 0; i < 8; i++)
            m[i][j] = a[i][j] + b[i][j];

    return 0;
}
```

```
int main() {
    int i, j;
    int m[8][8], a[8][8], b[8][8];

    for(i = 0; i < 8; i++)
        for(j = 0; j < 8; j++)
            m[i][j] = a[i][j] + b[i][j];

    return 0;
}
```

12. Como aumentar a localidade temporal do programa a seguir?

```
int main() {
    int i = 0, j = 0;
    int a = 0, b = 0;

    while(i < 10) {
        j++;
        a += i;
        b = j + 2;
        i++;
        j = b % 2;
    }
}
```

```

int main() {
    int i = 0, j = 0;
    int a = 0, b = 0;

    while(i < 10) {
        a += i;
        i++;

        j++;
        b = j + 2;
        j = b % 2;
    }
}

```

13. Quanto à estratégia Write-Back:

- (a) em que esta consiste? somente transmitir uma via da cache à memória principal quando o bloco desta via precisar ser substituído na cache.
- (b) qual parte do sistema faz seu controle? a própria cache, em hardware.
- (c) como esta interfere na complexidade do hardware implementado? a aumenta consideravelmente
- (d) como esta interfere na miss-penalty? a miss-penalty é aumentada porque, quando um endereço será substituído na cache, seu conteúdo precisa ser escrito na memória principal
- (e) como esta interfere na velocidade das escritas feitas pelo processador? aumenta consideravelmente, já que as escritas são feitas somente na cache

14. Quanto à estratégia Write-Through:

- (a) em que esta consiste? em escrever na cache e na memória principal a cada escrita
- (b) qual parte do sistema faz seu controle? o processador
- (c) como esta interfere na complexidade do hardware implementado? a mantém inferior à da write-back
- (d) como esta interfere na miss-penalty? não a altera
- (e) como esta interfere na velocidade das escritas feitas pelo processador? as mantém lentas, já que toda escrita acessa a memória principal

Tema: memória virtual

Para as questões a seguir, considere um sistema com 4 GB de memória física, 64 GB de memória virtual e páginas de 8 kB.

- 15. Quantos bits são destinados ao offset de página? $\log_2 8 * 1024 = 13$ bits
- 16. Quantos bits são destinados ao número de página físico? $32 - 13 = 19$ bits
- 17. Quantos bits são destinados ao número de página virtual? $36 - 13 = 23$ bits
- 18. Quantas páginas tem a tabela de páginas? $2^{23} = 8388608$ páginas
- 19. Em bits, qual o tamanho da tabela de páginas? $8388608 * (19 + 1 \text{ bit_de_validade} + 1 \text{ bit_de_escrita} + 1 \text{ bit_de_uso}) = 184549376$ bits

20. Quais componentes do sistema gerenciam a memória virtual? Toda a operação da memória virtual é gerenciada pelo sistema operacional.
21. Em um computador executando quatro processos, quantas tabelas de página estão instanciadas? Quantos registradores de tabela de página estão carregados? São instanciadas quatro tabelas de página. Entretanto, há somente um registrador para armazenar o endereço da página. O conteúdo deste é alterado conforme o processo ativo.
22. Quando o estado da memória é refletido na área de swap no disco? Quando seu conteúdo precisa ser substituído e houve uma escrita no endereço correspondente da memória física.
23. Descreva a sequência de acesso a um endereço registrado no TLB.
- (a) o processador soma o número de página virtual ao valor contido no registrador de tabela de página;
 - (b) o processador transmite o resultado ao TLB;
 - (c) o TLB compara este número com a tag de cada uma de suas entradas;
 - (d) o TLB indica ao processador que encontrou correspondência e retorna o número de página física;
 - (e) o processador concatena este número ao offset do endereço e o acessa na memória principal;
24. Descreva a sequência de acesso a um endereço não registrado no TLB, mas mapeado na memória principal.
- (a) o processador soma o número de página virtual ao valor contido no registrador de tabela de página;
 - (b) o processador transmite o resultado ao TLB;
 - (c) o TLB compara este número com cada uma de suas entradas;
 - (d) o TLB indica ao processador que o endereço não está registrado nele;
 - (e) o processador acessa a memória no endereço resultante, referente à respectiva entrada na tabela de páginas do processo;
 - (f) nesta entrada, lê que o endereço está na memória principal e extrai o respectivo número de página física;
 - (g) o processador concatena este número ao offset do endereço e acessa a memória principal no endereço resultante;
25. Descreva a sequência de acesso a um endereço não registrado no TLB, nem mapeado na memória principal.
- (a) o processador soma o número de página virtual ao valor contido no registrador de tabela de página;
 - (b) o processador transmite o resultado ao TLB;
 - (c) o TLB compara este número com cada uma de suas entradas;
 - (d) o TLB indica ao processador que o endereço não está registrado nele;
 - (e) o processador acessa a memória no endereço resultante, referente à respectiva entrada na tabela de páginas do processo;
 - (f) nesta entrada, lê que o endereço não está na memória principal;
 - (g) ocorrendo um *page fault*, o sistema operacional interrompe o processo;
 - (h) o SO, então, busca uma página cujo bit de uso seja 0;
 - (i) caso o bit de escrita desta seja 1, o SO transfere seu conteúdo para o disco;
 - (j) o SO transfere o conteúdo da página no disco para a página recém-liberada na memória;
 - (k) o processo é retomado, busca, novamente, o dado e o acessa na memória principal;

Tema: entrada e saída endereçada por memória

26. Qual instrução envia dados a uma I/O mapeada em memória? $s(w, h \text{ ou } b)$
27. Qual instrução lê dados de uma I/O mapeada em memória? $l(w, h \text{ ou } b)$
28. Há alguma informação que indica ao processador com I/O mapeado em memória se este está acessando a memória ou um periférico? Se sim, qual? Não há.
29. Como o processador pode acessar I/O por instruções destinadas a acesso à memória? Há suporte em hardware para tal: o decodificador de endereços recebe o mesmo endereço que a memória e os sinais de controle necessários para identificar a instrução executada. Com base nestes e caso se aplique, seleciona o periférico que deve receber o dado ou de qual periférico a saída deve ser transmitida ao processador.
30. Suponha um periférico com as seguintes características:

- registradores:

nome	endereço	modo	descrição
ctrl	0xFFFFF000	escrita	0: reconhece status 1: inicia leitura
status	0xFFFFF004	leitura	0: aguardando 1: realizando leitura 2: leitura concluída
leitura	0xFFFFF008	leitura	contém o resultado da leitura

- sua leitura retorna valores que devem ser interpretados em ponto fixo;
- é inicializado automaticamente, estando pronto para executar quando o programa começa;
- ao ser inicializado, está no estado **aguardando** e o valor em **leitura** não é válido;
- quando uma leitura é inicializada, o valor em **leitura** só é válido após sua conclusão;
- quando uma leitura é concluída, a próxima só pode ser iniciada após **ctrl** receber 0, indicando que o usuário reconhece a sinalização de conclusão;

escreva um código MIPS que computa a média de cinco leituras deste periférico.

https://github.com/Eduardo-bat/AOC_2024/blob/main/mmio.asm

Tema: barramentos

31. Compare um barramento multiplexado entre endereço e dado com um par de barramentos dedicados. Barramentos multiplexados consomem menos material e área, a custo de redução na velocidade de transmissão. Barramentos dedicados são mais rápidos, mas, também, mais custosos.
32. É possível multiplexar, também, os sinais de controle? Por quê? É possível, sim. Sequências específicas podem ser adotadas como controle, reduzindo linhas, mas aumentando a latência das transações. O protocolo I_2C , por exemplo, usa esta estratégia. Entretanto, outras especificações, como a da linha 808x da Intel, optam por manter as linhas de controle separadas, em prol de reduzir a latência de cada transmissão.
33. Como é dada a organização temporal dos sinais em um barramento síncrono? Como os protocolos verificam a validade das transmissões neste tipo de barramento? Em um barramento síncrono, a temporização dos sinais é feita em relação aos ciclos de clock. Durante uma transmissão, o significado do valor em cada linha é determinado pelo ciclo de clock no qual a

transmissão está. Os protocolos definidos sobre este tipo de barramento podem distinguir a validade das transmissões por meio dos valores capturados em certas linhas em determinados ciclos de clock. Por exemplo, um transmissor pode verificar se a linha **status** tem valor 0x00 no último ciclo de uma transmissão para determinar se esta foi bem-sucedida.

34. Como é dada a organização temporal dos sinais em um barramento síncrono? Como os protocolos verificam a validade das transmissões neste tipo de barramento? Em um barramento assíncrono, a temporização dos sinais é feita em relação aos próprios sinais. Durante a transmissão, o significado do valor em cada linha é determinado por qual foi o último sinal transmitido. Os protocolos definidos sobre este tipo de barramento podem distinguir a validade das transmissões por meio do intervalo entre os sinais ou pelo valor dos sinais. Por exemplo, um transmissor pode aguardar até 1 s pelo sinal de **acknowledge** e verificar se este contém o valor 0xAA.
35. Qual o papel do bus-master em um barramento compartilhado? É o bus-master quem inicia transações.
36. Qual o papel do bus-arbiter em um barramento compartilhado? É o bus-arbiter quem determina qual bus-master pode iniciar uma transação.
37. Em um barramento no qual um mestre deve ter a solicitação atendida imediatamente, qual deve ser a estratégia de alocação implementada? Preemptiva com prioridade.