

Organização de Computadores 2017.2

GABARITO AD2

- 1) (2,0) Faça uma busca na lista dos 500 sistemas de computadores com melhor desempenho do mundo em <http://www.top500.org> e descreva o primeiro colocado (pesquise neste mesmo site e na internet).

Em primeiro lugar na lista do Top 500 está o supercomputador Chinês Sunway TaihuLight (“A luz da divindade Taihu”, numa mistura de chinês e inglês). A ordem da lista do top500 é definida pelo desempenho em petaflop/s, cada petaflop/s representa um quadrilhão de operações de ponto flutuante por segundo. O TaihuLight tem um rendimento de 93 petaflop/s podendo chegar a velocidade máxima de 125 petaflop/s, sua velocidade supera em 3 vezes o segundo colocado, também Chinês, o Tianhe-2. A lista é atualizada 2 vezes por ano o TaihuLight está no topo nas últimas 3 atualizações.

O TaihuLight foi desenvolvido pelo Centro de Nacional de Pesquisas em Tecnologia e Engenharia Computacional em Paralelo e está instalado no Centro Nacional do Supercomputação da China, em Wuxi, província de Jiangsu. Tem uma característica muito peculiar: é 100% fabricado na China, inclusive seus processadores (SW26010 - processador RISC de 64bits baseados na arquitetura Sunway) são desenhados e fabricados ali mesmo. São cerca de 40.960 processadores, cada um com 256 cores, o que totaliza mais ou menos 10 milhões de unidades de processamento (cores). Cada um deles funciona a uma velocidade bastante convencional: 1,45 GHz. Sua memória de trabalho é de cerca de 1.310.720 GB e o sistema operacional para gerenciar esse supercomputador é o Sunway RaiseOS 2.0.5.

Este supercomputador exige cerca de 15 MW (megawatts) de potência em pleno rendimento, o que resulta numa eficiência de 6 gigaflop/s/watt. Isso o coloca no terceiro lugar entre os mais eficientes do mundo, embora o resultado se deva em parte a ele não ter tanta memória quanto seria de esperar num equipamento desse porte. Comparativamente, um laptop usa 50W, e um desktop fica entre 100 e 200W – ou seja, o TaihuLight consome tanta energia quanto um milhão de computadores caseiros.

- 2) (2,0) Responda as questões abaixo:

- a) Analise os modos de endereçamento direto, indireto e imediato, estabelecendo diferenças de desempenho, vantagens e desvantagens de cada um.

Imediato: O campo operando contém o dado, desta forma o dado é transferido da memória juntamente com a instrução.

Desempenho: Não requer acesso a memória principal. Mais rápido que o modo direto

Vantagem: Rapidez na execução da instrução, pois não requer acesso à memória principal, apenas na busca da própria instrução.

Desvantagem. Limitação do tamanho do campo operando das instruções reduzindo o valor máximo do dado a ser manipulado. Trabalho excessivo para alteração de valores quando o programa é executado repetidamente e o conteúdo das variáveis serem diferentes em cada execução.

Direto: O campo operando da instrução contém o endereço onde se localiza o dado.

Desempenho: Requer apenas um acesso a memória principal. Mais rápido que o modo indireto

Vantagem. Flexibilidade no acesso a variáveis de valor diferente em cada execução do programa

Desvantagem. Limitação de memória a ser usada conforme o tamanho do operando.

Indireto: O campo de operando contém o endereço de uma célula, sendo o valor contido nesta célula o endereço do dado desejado.

Desempenho: Requer 2 acessos a memória principal

Vantagem: Usar como “ponteiro”. Elimina o problema do modo direto de limitação do valor do endereço do dado. Manuseio de vetores (quando o modo indexado não está disponível).

Desvantagem: Muitos acesso à MP para execução, requer pelo menos 2 acessos à memória principal.

- b) Qual é o objetivo do emprego do modo de endereçamento base mais deslocamento? Qual é a diferença de implementação e utilização entre esse modo e o modo indexado?

O base mais deslocamento tem como seu principal objetivo permitir a modificação de endereço de programas ou módulos destes (que é a relocação de programa), bastando para isso uma única alteração no registrador base.

O base mais deslocamento tem como característica o endereço ser obtido da soma do deslocamento com o registrador base, diferindo do modo indexado onde o do registrador base é fixo e variar no deslocamento, ao contrário deste onde o deslocamento é fixo e com a alteração do registrador base permite-se a mudança do endereço.

Exemplos de instruções modo indexado:

LDX Ri, Op ==> ACC <--- ((Op) + (Ri))

ADX Ri, Op ==> ACC <--- ACC + ((Op) + (Ri))

Exemplo: instrução base mais deslocamento:

LDB Rb, Op ==> (ACC) <--- ((Op) + (Rb))

ADB Rb, Op ==> ACC <--- ACC + ((Op) + (Rb))

Sendo,

Op = Operando

Ri = Registrador de índice

Rb = Registrador base

- c) Compilação e Interpretação (Dê exemplos de linguagens que se utilizem de compiladores e de linguagens que se utilizem de interpretadores).

A compilação consiste na análise de um programa escrito em linguagem de alto nível (programa fonte) e sua tradução em um programa em linguagem de máquina (programa objeto).

Na interpretação cada comando do código fonte é lido pelo interpretador, convertido em código executável e imediatamente executado antes do próximo comando.

A interpretação tem como vantagem sobre a compilação a capacidade de identificação e indicação de um erro no programa-fonte (incluindo erro da lógica do algoritmo) durante o processo de conversão do fonte para o executável.

A interpretação tem como desvantagem o consumo de memória devido ao fato de o interpretador permanecer na memória durante todo o processo de execução do programa. Na compilação o compilador somente é mantido na memória no processo de compilação e não utilizado durante a execução. Outra desvantagem da interpretação está na necessidade de tradução de partes que sejam executadas diversas vezes, como os loops que são traduzidos em cada passagem. No processo de compilação isto só ocorre uma única vez. Da mesma forma pode ocorrer para o programa inteiro, em caso de diversas execuções, ou seja, a cada execução uma nova interpretação.

Exemplos de linguagem interpretadas: ASP, BASIC, Java, PHP, Python, Lisp entre outras.

Exemplos de linguagem compilada: C, Pascal, Delphi, Visual Basic, entre outras.

- d) Sistemas MIMD e Sistemas SIMD (Forneça exemplos atuais desses sistemas).

SIMD – Single instruction stream, multiple data stream, onde vários elementos de processamento executam a mesma sequência de instruções ao mesmo tempo sobre um conjunto de dados diferente. Exemplo: Processadores matriciais. Os aceleradores de hardware baseados em placas gráficas GPU também pertencem a essa classificação. Uma GPU é composta de vários multiprocessadores e cada multiprocessador consiste de um grupo de processadores escalares.

MIMD – Multiple instruction stream, multiple data stream, classificação esta onde um conjunto de processadores executam simultaneamente sequências diferentes de instruções sobre conjunto de dados diferentes. Exemplo: SMPs, clusters, sistemas NUMA. Os multicores, processadores com 2 ou mais núcleos que atuam como processadores distintos e executam instruções independentes, também são classificados como MIMD.

- 3) (2,0) Faça uma pesquisa e descreva a técnica de Simultaneous Multi-Threading (SMT), também chamada de Hyper-threading. Sugestão:
https://en.wikipedia.org/wiki/Simultaneous_multithreading.

O Multithreading simultâneo (SMT) ou Hyper-threading é uma técnica que melhora a eficiência geral das CPUs superscalares com multithreading de hardware . O SMT permite vários segmentos de execução independentes para melhor aproveitar os recursos fornecidos pelas arquiteturas modernas do processador [2]. O SMT combina a capacidade de despacho de múltiplas instruções das arquiteturas superescalares, com a habilidade de esconder latência das arquiteturas multithreading.

Em um processador super-escalar convencional, apenas as instruções de uma thread são executadas por vez. Embora esse tipo de processador ofereça paralelismo em nível de instruções (já que o pipeline processa trechos de várias instruções simultaneamente), muitos ciclos são perdidos em stalls (principalmente devido a latência no acesso à memória quando da ocorrência de um cache miss) e o processador não é utilizado em sua totalidade (devido ao baixo paralelismo de algumas instruções)[3]

A Figura 1 (Superscalar) representa a utilização temporal do processador; cada linha representa um ciclo e os blocos representam a utilização das unidades funcionais. Blocos vazios representam desperdício de uso do processador; stalls são desperdícios verticais e unidades funcionais não utilizadas em um ciclo são desperdícios horizontais.

A Figura 1 (Multithreading) representa o comportamento dos processadores multithreads que buscam diminuir o desperdício vertical, trocando a thread em execução quando da ocorrência de um stall; essa troca de contexto também é chamada de troca leve (pois requer poucos ciclos), para diferenciar da troca realizada pelo sistema operacional (S.O.), que pode custar até centenas de ciclos.

Já os processadores SMT, Figura 1(Simultaneous multithreading), vão um passo além na diminuição do desperdício: além de não precisar trocar o contexto das threads (desperdício vertical), como o processador tem as instruções de mais de uma thread disponível simultaneamente, ele pode escolher melhor quais instruções enviar ao pipeline, melhorando o paralelismo das instruções (o que diminui o desperdício horizontal) e mitigando a latência causada pelos caches misses (diminuição do desperdício vertical).

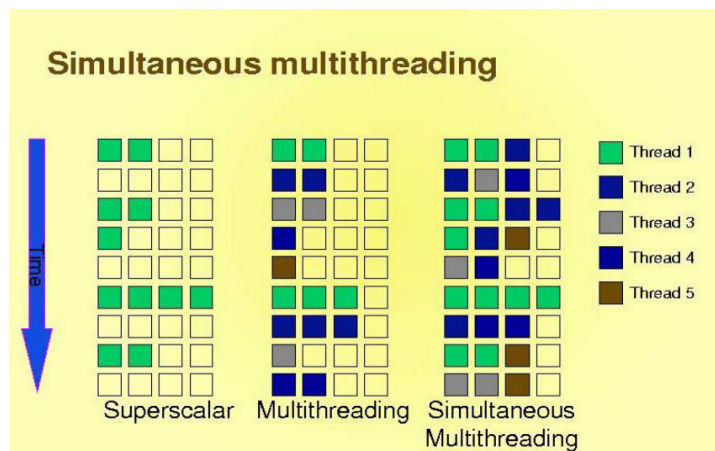


Figura 1: Multithreads simultâneas (Fonte: [1])

Desvantagens do SMT: Dependendo do design e da arquitetura do processador, o multithreading simultâneo pode diminuir o desempenho se qualquer um dos recursos compartilhados forem “engargalados” em função do o desempenho. Existe também uma

preocupação de segurança com certas implementações de multithreading simultâneas. O Hyperthreading da Intel em processadores baseados no NetBurst possui uma vulnerabilidade através da qual é possível que um aplicativo roube uma chave criptográfica de outro aplicativo executado no mesmo processador, monitorando o uso do cache.

Fontes:

- [1] - Eggers, S. J., Emer J. S., Levy, H. M., Lo, J. L., Stamm, R. L., Tullsen, D. M. Simultaneous Multithreading: A Platform for Next-Generation Processors. IEEE Micro, (Set/Out 1997), 12-19
- [2] – Simultaneous Multi-Threading , disponível em <https://en.wikipedia.org/wiki/Simultaneous_multithreading> acesso em 24/10/2017.
- [3] - Leme, F. "Processadores SMT e paralelismo em nível de threads." . Unicamp.

4) (3,5) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 32 bits. Na representação para ponto flutuante, utiliza-se o padrão IEEE 754 para precisão simples.

- a) Caso o computador tenha armazenado o conteúdo $(F2A86800)_{16}$, qual o valor deste número em decimal, se considerarmos que este padrão de bits representa um inteiro utilizando-se representação sinal magnitude, um inteiro em complemento a 2 e um real em ponto flutuante ? (1,0)

$$(F2A86800)_{16} = (1111\ 0010\ 1010\ 1000\ 0110\ 1000\ 0000\ 0000)_2$$

Sinal magnitude:

$$- (+ 2^{30} + 2^{29} + 2^{28} + 2^{25} + 2^{23} + 2^{21} + 2^{19} + 2^{14} + 2^{13} + 2^{11}) = -1.923.639.296$$

Complemento a 2

$$- 2^{31} + (+ 2^{30} + 2^{29} + 2^{28} + 2^{25} + 2^{23} + 2^{21} + 2^{19} + 2^{14} + 2^{13} + 2^{11}) = -223.844.352$$

Real em ponto flutuante

Sinal = 1 (negativo)

Expoente = 11100101 = 229 - 127 = +102₁₀

Mantissa = , 0101000011010000000000

Formatando na notação científica +1,010100001101 x 2⁺¹⁰²

Resultado: - (+ 2¹⁰² + 2¹⁰⁰ + 2⁹⁸ + 2⁹³ + 2⁹² + 2⁹⁰) = - 6,67 x 10⁺³⁰

- b) Qual será a representação em ponto flutuante dos seguintes valores decimais neste computador:

i) +9.112,25 (0,5)

Convertendo para binário = +10001110011000,01 =>

colocando na notação científica +1,000111001100001 x 2⁺¹³

Temos então:

Sinal = 0 (positivo)

Expoente = +13₁₀ = 13 + 127 = 140 = 10001100₂

Mantissa = , 000111001100001

Resultado: 0 10001100 000111001100001000000000

ou 01000110 00001110 01100001 00000000

ii) -91,625 (0,5)

Convertendo para binário = -1011011,101 =>
colocando na notação científica $-1,011011101 \times 2^{+6}$

Temos então:

Sinal = 1(negativo)

Expoente = $+6_{10} = 6 + 127 = 133 = 10000101_2$

Mantissa = , 011011101

Resultado: 1 10000101 011011101000000000000000
ou 11000010 10110111 01000000 00000000

c) Mostre a representação dos números dos itens da questão acima, caso se utilizasse a representação complemento a 2 para representar o expoente. (0,7)

i) +9.112,25 (0,5)

Convertendo para binário = +10001110011000,01 =>
colocando na notação científica $+1,000111001100001 \times 2^{+13}$

Temos então:

Sinal = 0 (positivo)

Expoente = $+13_{10} = +13$ (por complemento a 2) = 00001101₂

Mantissa = , 000111001100001

Resultado: 0 00001101 000111001100001000000000
ou 00000110 10001110 01100001 00000000

ii) -91,625 (0,5)

Convertendo para binário = -1011011,101 =>
colocando na notação científica $-1,011011101 \times 2^{+6}$

Temos então:

Sinal = 1(negativo)

Expoente = $+6_{10} = +6$ (por complemento a 2) = 00000110₂

Mantissa = , 011011101

Resultado: 1 00000110 011011101000000000000000
ou 10000011 00110111 01000000 00000000

d) Supondo que se utilize a representação complemento a 2 para o expoente ao invés da representação em excesso, indique quais o menor e o maior valor positivos normalizados na representação em ponto flutuante para este computador (Considere, neste caso, que todas as representações são utilizadas para números normalizados, não existem os casos especiais). (0,8)

OBS: não existem casos especiais

Maior positivo : 0 01111111 11111111111111111111

Normalizado: $+1,11111111111111111111 \times 2^{+127} = +3,4 \times 10^{+38}$

Menor positivo: 0 10000000 000000000000000000000000

Normalizado: $+1,00000000000000000000 \times 2^{-128} = +2,94 \times 10^{-39}$

5) (1,5) Considere um sistema com relógio de frequência igual a 1600 MHz. Este sistema pode realizar operações de entrada e saída por programação (polling) ou por interrupção. O número de ciclos de relógio para uma operação por programa é igual a 200 (inclui a chamada à rotina de “polling”, acesso ao dispositivo e retorno da chamada). O número de ciclos de relógio para atender uma interrupção é igual a 500. Define-se overhead como o percentual de ciclos de relógio que são utilizados para operação de entrada e saída em relação ao número total de ciclos de relógio disponíveis em 1 segundo. Baseado nesta definição, responda as seguintes questões:

a) Determine o overhead que ocorre quando se utiliza a interface por programa para os seguintes dispositivos:

i) (0,2) Um mouse que deve ser interrogado pelo sistema 60 vezes por segundo para garantir que nenhum movimento dele seja perdido.

O mouse é consultado 60 vezes por segundo. Cada operação de consulta consome 200 ciclos. São necessários, então, $60 \times 200 = 12.000$ ciclos/segundo para as operações de consulta ao mouse.

O relógio é de 1600MHz, que corresponde a $1,6 \times 10^9$ ciclos.

O overhead será obtido pela seguinte razão:

$$(12.000 \text{ ciclos/seg}) / (1,6 \times 10^9 \text{ ciclos/seg}) = 0,0000075 \text{ ou } 0,00075\%$$

ii) (0,3) Um disco rígido que gera dados a uma taxa de 12 MBytes/segundo para sua controladora que possui um buffer de 32 bytes, ou seja, os dados são transferidos para o sistema em unidades de 24 bytes.

OBS: As unidade de transferência e o tamanho do buffer deveriam ter o mesmo tamanho, como houve um erro na digitação dos valores, serão consideradas as respostas que utilizem a unidade de armazenamento de 24bytes ou de 32bytes

1ª. Solução (unidade de transferência = 32 Bytes)

Taxa de transferência = 12MB/s = 12.000.000 B/s.

Em cada operação é transferida uma unidade de 32 bytes.

A cada segundo ocorrem $12.000.000B / 32B = 375.000$ operações, como cada operação leva 200 ciclos, então, o total de ciclos será $375.000 \times 200 = 75 \times 10^6$ ciclos/s.

O relógio é de 1,6GHz, que corresponde a 1600×10^6 ciclos/s.

O overhead será obtido pela seguinte razão:

$$75 \times 10^6 \text{ ciclos/s} / 1600 \times 10^6 \text{ ciclos/s} = 0,046875 \text{ ou } 4,6875\%$$

2ª. Solução (unidade de transferência = 24 Bytes)

Taxa de transferência = 12MB/s = 12.000.000 B/s.

Em cada operação é transferida uma unidade de 24 Bytes.

A cada segundo ocorrem $12.000.000B / 24B = 500.000$ operações, como cada operação leva 200 ciclos, então, o total de ciclos será $500.000 \times 200 = 100 \times 10^6$ ciclos/s.

O relógio é de 1,6GHz, que corresponde a 1600×10^6 ciclos/s.

O overhead será obtido pela seguinte razão:

$$100 \times 10^6 \text{ ciclos/s} / 1600 \times 10^6 \text{ ciclos/s} = 0,0625 \text{ ou } 6,25\%$$

b) Determine o overhead que ocorre quando se utiliza a interface por interrupção para o disco rígido nas seguintes situações:

i) (0,3) O disco está ativo (gera dados) 100 % do tempo.

1ª. Solução (unidade de transferência = 32 Bytes)

Taxa de transferência = 12MB/s = 12.000.000 B/s. .

Em cada operação é transferida uma unidade de 32Bytes.

A cada segundo ocorrem $12.000.000B / 32B = 375.000$ operações,

como cada operação leva 500 ciclos, então, o total de ciclos

será $375.000 \times 500 = 187.500.000$ ciclos/s.

O overhead será obtido pela seguinte razão:

$187.500.000 \text{ ciclos/s} / 1.600.000.000 \text{ ciclos/s}$

= 11,72% de overhead estando ativo 100% do tempo

2ª. Solução (unidade de transferência = 24 Bytes)

Taxa de transferência = 12MB/s = 12.000.000 B/s. .

Em cada operação é transferida uma unidade de 24Bytes.

A cada segundo ocorrem $12.000.000B / 24B = 500.000$ operações,

como cada operação leva 500 ciclos, então, o total de ciclos

será $500.000 \times 500 = 250.000.000$ ciclos/s.

O overhead será obtido pela seguinte razão:

$250.000.000 \text{ ciclos/s} / 1.600.000.000 \text{ ciclos/s}$

= 15,63% de overhead estando ativo 100% do tempo

ii) (0,2) O disco está ativo (gera dados) em média 8 % do tempo.

1ª. Solução (unidade de transferência = 32 Bytes)

Taxa de transferência = 12MB/s = 12.000.000 B/s. .

Em cada operação é transferida uma unidade de 32Bytes.

A cada segundo ocorrem $12.000.000B / 32B = 375.000$ operações,

como cada operação leva 500 ciclos, então, o total de ciclos

será $375.000 \times 500 = 187.500.000$ ciclos/s.

O overhead será obtido pela seguinte razão:

$187.500.000 \text{ ciclos/s} / 1.600.000.000 \text{ ciclos/s}$

= 11,72% de overhead caso estivesse ativo 100% do tempo

Como está ativo apenas 8% do tempo, o overhead será: $11,72\% \times 8\% = 0,938\%$.

2ª. Solução (unidade de transferência = 24 Bytes)

Taxa de transferência = 12MB/s = 12.000.000 B/s.

Em cada operação é transferida uma unidade de 24Bytes.

A cada segundo ocorrem $12.000.000B / 24B = 500.000$ operações,

como cada operação leva 500 ciclos, então, o total de ciclos

será $500.000 \times 500 = 250.000.000$ ciclos/s.

O overhead será obtido pela seguinte razão:

$250.000.000 \text{ ciclos/s} / 1.600.000.000 \text{ ciclos/s}$

= 15,63% de overhead caso estivesse ativo 100% do tempo

Como está ativo apenas 8% do tempo, o overhead será: $15,63\% \times 8\% = 1,25\%$.

iii) (0,5) Para se tentar diminuir o overhead, aumentou-se o tamanho do buffer da controladora para 64 bytes, ou seja somente quando o buffer estiver com 64

bytes é que deverá ser gerada uma transferência. Mostre se esta medida irá diminuir o overhead quando o disco está 100% ativo

Taxa de transferência = 12MB/s = 12.000.000 B/s

Em cada operação é transferida uma unidade de 64Bytes.

A cada segundo ocorrem $12.000.000B / 64B = 187.500$ operações,

como cada operação leva 500 ciclos, então, o total de ciclos

será $187.500 \times 500 = 93.750.000$ ciclos/s.

O overhead será obtido pela seguinte razão:

$93.750.000 \text{ ciclos/s.} / 1.600.000.000 \text{ ciclos/s}$

= 5,86% de overhead caso estivesse ativo 100% do tempo

Observa-se, então, com o aumento do buffer para 64Bytes, o overhead diminui.