

!

GABARITO AD2 - Organização de Computadores 2018.1

- 1) (1,0) Considere uma máquina que utiliza n bits para representar inteiros com sinal em complemento a 2. Em uma operação de soma S de dois valores $N1$ e $N2$, **não há detecção de estouro** (*overflow*). Responda as perguntas abaixo:

- a) Caso ocorra estouro, indique o valor que será considerado pela máquina em função de S e do número de bits n .

Na ocorrência de um estouro o valor passará a ter o sinal inverso ao do valor esperado, exemplo $N1$ e $N2$ serem positivo e resultado negativo S .

- b) Indique uma maneira de se detectar que houve estouro baseando-se nos sinais de $N1$, $N2$ e S

Sinais			Comentários
$N1$	$N2$	S	
0	0	0	Não houve overflow
0	0	1	Houve overflow, soma de 2 números positivos resulta em número negativo
0	1	0	Não houve overflow
0	1	1	Não houve overflow
1	0	0	Não houve overflow
1	0	1	Não houve overflow
1	1	0	Houve overflow, soma de 2 números negativos resulta em número positivo
1	1	1	Não houve overflow

A partir do quadro acima, observamos que na soma de 2 números ($N1, N2$) com sinais diferentes nunca teremos overflow, este só ocorrerá quando houver soma de 2 números ($N1, N2$) de mesmo sinal e o resultado com sinal diferente (S), isto já afirmado no item a

- 2) (2,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 16 bits. Na representação em ponto flutuante, as combinações possíveis de bits representam números normalizados do tipo $\pm(1, b_1b_2b_3b_4b_5b_6b_7b_8b_9b_{10} \times 2^{\text{Expoente}})$, onde o bit mais à esquerda representa o sinal (0 para números positivos e 1 para números negativos), os próximos 5 bits representam o expoente em complemento a 2 e os 10 bits seguintes representam os bits b_1 a b_{10} , como mostrado na figura a seguir:

Sinal	Expoente representado em complemento a 2	$b_1b_2b_3b_4b_5b_6b_7b_8b_9b_{10}$
1	5	10

- a) Considere o seguinte conjunto de bits representado em hexadecimal $(CA90)_{16}$. Indique o valor deste número **em decimal**, considerando-se que o conjunto representa:

$$(CA90)_{16} = (1100\ 1010\ 1001\ 0000)_2$$

- i) um inteiro sem sinal.

$$2^{15} + 2^{14} + 2^{11} + 2^9 + 2^7 + 2^4 = 51.856$$

ii) um inteiro em sinal magnitude.

$$-(2^{14} + 2^{11} + 2^9 + 2^7 + 2^4) = -19.088$$

iii) um inteiro em complemento a 2.

$$-2^{15} + (2^{14} + 2^{11} + 2^9 + 2^7 + 2^4) = -13.680$$

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

i) +58,75

Convertendo para binário = $+111010,11_2 \Rightarrow$
colocando na notação científica $+1,1101011 \times 2^{+5}$
Temos então:
Sinal = 0 (positivo)
Expoente = $+5_{10} \Rightarrow$ (complemento a 2) 00101_2
Mantissa = ,1101011000
Resultado: 0 00101 1101011000 ou 0001011101011000

ii) -43,125

Convertendo para binário = $-101011,001_2 \Rightarrow$
colocando na notação científica $-1,01011001 \times 2^{+5}$
Temos então:
Sinal = 1 (negativo)
Expoente = $+5_{10} \Rightarrow$ (complemento a 2) 00101_2
Mantissa = ,0101100100
Resultado: 1 00101 0101100100 ou 1001010101100100

c) Indique o menor e o maior valor positivo normalizado na representação em ponto flutuante para este computador. Mostre os valores **em decimal**.

Menor valor positivo:
 $0\ 10000\ 0000000000 = 1,0 \times 2^{-16} = +0,00001526$
Maior valor positivo:
 $0\ 01111\ 1111111111 = 1,111111111 \times 2^{+15} = +65504$

d) Caso se deseje utilizar a representação em excesso para representar o expoente, indique o excesso a ser utilizado e o menor e o maior valor positivo normalizado para esta nova representação. Mostre os valores **em decimal**.

Excesso = $2^{(n-1)} - 1$, sendo n igual a quantidade de bits do expoente = $2^{(5-1)} - 1 = 15$
Menor valor positivo:
 $0\ 00001\ 0000000000 = 1,0 \times 2^{-14} = +0,000061$
Maior valor positivo:
 $0\ 11110\ 1111111111 = 1,111111111 \times 2^{+15} = +65504$

3) (1,0) Explique como funciona um DVD. (sugestões de fonte de consulta: livro do Stallings e do Mário Monteiro e o site www.guiadohardware.net/. Na sua resposta indique as suas fontes de consulta).

Textos base encontrados em:
Arquitetura e Organização de Computadores – William Stallings

Organização estruturada de Computadores - Tanenbaum

Os DVDs usam o mesmo design dos CD-ROMs, discos de 120mm de policarbonato. O disco é constituído de uma resina do tipo policarbonato e revestido com uma superfície com alto índice de reflexão, normalmente de alumínio. A informação digital registrada é impressa nessa superfície refletiva (LANDs) como uma série de sulcos microscópicos (PITs) que são iluminados por um diodo a laser e lidos por um fotodetector. Os pits e os lands são escritos em uma única espiral contínua, começando próximo ao buraco central do DVD. Em vez de usar o pit para 0 e o land para 1, usa-se a transição de pit-land ou land-pit para representar o bit 1 e a ausência dessa transição de 0.

Quando comparados com CDs, os DVDs possuem pits menores (0,4 micron), utilizam uma pista espiral mais apertada (0,74 micron) e um laser vermelho mais fino (0,65 micron). A capacidade de um DVD é de 4,7GB utilizando apenas um lado e uma única camada. Um drive de DVD com velocidade de 1x opera a 1,4MB/s. Formatos de DVD: Um único lado com uma única camada (4,7GB), um único lado com duas camadas (8,5GB), dois lados cada um com uma única camada (9,4GB) e, finalmente, dois lados cada um com duas camadas (17GB).

Um aparelho de DVD possui um mecanismo que projeta um raio laser sobre a superfície do disco para ler o padrão de saliências. O aparelho de DVD decodifica o filme MPEG-2, transformando-o em um sinal de vídeo composto padrão. O aparelho também decodifica o fluxo de dados de áudio ("stream") e o envia para um decodificador, onde ele é amplificado e enviado para os alto-falantes.

O trabalho fundamental do aparelho de DVD é focalizar o laser sobre a trilha de saliências. O laser pode focalizar o material refletor semitransparente atrás da camada mais próxima ou, no caso de um disco de camada dupla, através dessa camada e até o material refletor atrás da camada interna. O raio laser passa através da camada de policarbonato, se reflete na camada refletora atrás dela e atinge o dispositivo optoeletrônico que detecta mudanças na luz. As saliências refletem a luz diferentemente das áreas planas do disco e o sensor optoeletrônico detecta essa mudança na refletividade. A eletrônica no drive interpreta as mudanças na refletividade para ler os bits que compõem os bytes.

A parte mais difícil da leitura do DVD é manter o raio laser centralizado sobre a trilha de dados. Essa centralização é o trabalho do sistema de rastreamento. Conforme o DVD é reproduzido, o sistema de rastreamento precisa mover o laser continuamente para fora. À medida que o laser se move para fora em relação ao centro do disco, as saliências passam pelo laser com maior velocidade. Isso acontece por causa da velocidade linear ou tangencial das saliências, que é igual ao raio do disco multiplicado pela velocidade na qual o disco está girando. Assim, conforme o laser se move para fora, o motor do fuso precisa diminuir a velocidade de giro do DVD, para que os ressaltos passem pelo laser em velocidade constante e os dados sejam lidos do disco também a uma taxa constante.

4) (3,0) Explique, comparando:

- a) 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

- b) Sistemas SMP e Sistemas NUMA (Forneça exemplos atuais de sistemas SMP e Sistemas NUMA).

Sistemas SMP (ou UMA) têm como característica o acesso a todas as partes da memória principal com tempo de acesso uniforme. Em sistemas NUMA, todos os processadores possuem também acesso a todas as partes da memória principal podendo diferir o tempo de acesso em relação às posições da memória e processador.

Nos sistemas SMP o aumento no número de processadores tem como consequência problemas de tráfego

no barramento degradando o desempenho. Uma solução para isto é a utilização de clusters, que tem, usualmente, como consequência alterações significativas na aplicação (software). Nos sistemas NUMA podem-se ter vários nós multiprocessadores, cada qual com seu próprio barramento, resultando em pequenas alterações na aplicação (software).

Exemplo de NUMA: Cray T3E, SGI Origin 3000

Exemplo de SMP: Maioria dos servidores da HP, IBM e mais recentemente arquiteturas multicore que acessam a mesma memória principal.

- c) Arquiteturas RISC e Arquiteturas CISC (Forneça exemplos em processadores atuais de características RISC e CISC).

RISC: Reduced Instruction Set Computer – Computador com um conjunto reduzido de instruções

CISC - Complex Instruction Set Computer: Computador com um conjunto complexo de instruções

CISC: Principais características:

Possui microprogramação para aumento da quantidade de instruções incluindo novos modos de endereçamento, de forma a diminuir a complexidade dos compiladores e em consequência permitir linguagens de alto nível com comandos poderosos para facilitar a vida dos programadores. Em contrapartida, muitas instruções significam muitos bits em cada código de operação, instrução com maior comprimento e maior tempo de interpretação.

RISC: Principais características:

Menor quantidade de instruções e tamanho fixo. Não há microprogramação. Permite uma execução otimizada, mesmo considerando que uma menor quantidade de instruções vá conduzir a programas mais longos. Uma maior quantidade de registradores e suas utilizações para passagem de parâmetros e recuperação dos dados, permitindo uma execução mais otimizada de chamada de funções. Menor quantidade de modos de endereçamento com o objetivo de reduzir de ciclos de relógio para execução das instruções. Instruções de formatos simples e únicos tiram maior proveito de execução com pipeline cujos estágios consomem o mesmo tempo.

Os processadores Intel e AMD são exemplos de processadores CISC, já o Power PC e o Sparc da Sun (não tão atuais) são exemplos da arquitetura RISC. Podemos dizer que os processadores atuais são na verdade uma mistura das duas implementando em um chip o que é vantajoso de cada arquitetura.

- d) (1,0) Dados os valores de memória abaixo e uma máquina de 1 endereço com um acumulador:

palavra 20 contém 30

palavra 30 contém 40

palavra 40 contém 50

palavra 50 contém 60

Load imediato 20

Resposta: Nesta instrução o valor a ser colocado no acumulador corresponde ao valor fornecido como operador, portanto:

ACC <- 20 (o valor a ser colocado no acumulador é 20)

Load direto 20

Resposta: Nesta instrução o valor a ser colocado no acumulador corresponde ao valor contido no endereço de memória fornecido como operador, portanto:

ACC <- (20) (o valor a ser colocado no acumulador é 30)

Load indireto 20

Resposta: Nesta instrução o valor a ser colocado no acumulador corresponde ao valor contido no endereço que consta como valor no endereço de memória fornecido como operador, portanto:

ACC <- ((20)) (o valor a ser colocado no acumulador é 40)

- 5) (1,0) Analise os modos de endereçamento direto, indireto e imediato, estabelecendo diferenças de desempenho, vantagens e desvantagens de cada um.

<i>MODO DE ENDEREÇAMENTO</i>	<i>DEFINIÇÃO</i>	<i>VANTAGENS</i>	<i>DESVANTAGENS</i>	<i>DESEMPENHO</i>
<i>Imediato</i>	<i>O campo operando contém o dado</i>	<i>Rapidez na execução da instrução</i>	<i>Limitação do tamanho do dado. Inadequado para o uso com dados de valor variável</i>	<i>Não requer acesso a memória principal. Mais rápido que o modo direto</i>
<i>Direto</i>	<i>O campo operando contém o endereço do dado</i>	<i>Flexibilidade no acesso a variáveis de valor diferente em cada execução do programa</i>	<i>Perda de tempo, se o dado é uma constante</i>	<i>Requer apenas um acesso a memória principal. Mais rápido que o modo indireto</i>
<i>Indireto</i>	<i>O campo operando corresponde ao endereço que contém a posição onde está o conteúdo desejado,</i>	<i>Manuseio de vetores (quando o modo indexado não está disponível). Usar como “ponteiro”</i>	<i>Muitos acessos à MP para execução</i>	<i>Requer 2 acessos a memória principal</i>

- 6) (1,0) Qual é o objetivo do emprego do modo de endereçamento base mais deslocamento? Qual é a diferença de implementação 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