

AD2 - Organização de Computadores 2015.2

Data de entrega:17/11/2015

1. (2,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 20 bits. Na representação em ponto flutuante, as combinações possíveis de bits representam números normalizados do tipo $\pm(1, b_{-1} b_{-2} b_{-3} b_{-4} b_{-5} b_{-6} b_{-7} b_{-8} b_{-9} b_{-10} b_{-11} b_{-12} b_{-13} \times 2^{\text{Expoente}})$, onde o bit mais à esquerda representa o sinal (0 para números positivos e 1 para números negativos), os próximos 6 bits representam o expoente em complemento a 2 e os 13 bits seguintes representam os bits $b_{-1} b_{-13}$, como mostrado na figura a seguir:

a) (0,4) Considere o seguinte conjunto de bits representado em hexadecimal CF8A0. Indique o valor deste número em decimal, considerando-se que o conjunto representa:

$$(CF8A0)_{16} = (1100\ 11111000\ 10100000)$$

(a.1) um inteiro sem sinal

$$2^{19} + 2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^7 + 2^5 = 850.080$$

(a.2) um inteiro em sinal magnitude

$$-(2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^7 + 2^5) = -325.792$$

(a.3) um inteiro em complemento a 2

$$-2^{19} + (2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^7 + 2^5) = -198.496$$

(a.4) um real em ponto flutuante conforme descrição do enunciado.

1100 11111000 10100000

Sinal = 1 (negativo)

Expoente = $100111_2 = -25$

Mantissa = , 1100010100000

$$-1,1100010100000 \times 2^{-25} = -(2^{-25} + 2^{-26} + 2^{-27} + 2^{-31} + 2^{-33}) = -5,27 \times 10^{-8}$$

b) (0,2) Qual será a representação em ponto flutuante dos seguintes valores decimais neste computador:

b.1) +297,55

Convertendo para binário = 100101001,100011001100110011...

$$100101001,100011001100110011... = 1,00101001100011001100110011... \times 2^{+8}$$

Temos então:

Sinal = 0 (positivo)

Expoente = +8 = (representando em complemento a 2) = 001000

Mantissa = , 0010100110001

Resultado: 0 001000 0010100110001

b.2) -8,8

Convertendo para binário = 1000,1100110011001100110011...

$$1000,1100110011001100110011... = 1,0001100110011001100110011... \times 2^{+3}$$

Temos então:

Sinal = 1 (negativo)

Expoente = +3 = (representando em complemento a 2) = 000011

Mantissa = , 0001100110011

Resultado: 1 000011 0001100110011

c) (0,6) Indique o menor e o maior valor positivo normalizado na representação em ponto flutuante para este computador. Mostre os valores em decimal.

$$\begin{aligned} \text{Maior positivo: } 0 \ 011111 \ 111111111111 &= \\ 1, 111111111111 \times 2^{+31} &= +4.294.705.152 \\ \text{Menor positivo: } 0 \ 100000 \ 000000000000 &= \\ 1, 000000000000 \times 2^{-32} &= +2,33 \times 10^{-10} \end{aligned}$$

d) (0,8) 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.

$$\text{Excesso} = 2^n - 1 \Rightarrow \text{com } n=6 \Rightarrow \text{Excesso} = 31$$

$$\begin{aligned} \text{Maior positivo: } 0 \ 111110 \ 111111111111 &= \\ 1, 111111111111 \times 2^{+31} &= +4.294.705.152 \\ \text{Menor positivo: } 0 \ 000001 \ 000000000000 &= \\ 1, 000000000000 \times 2^{-30} &= +9,31 \times 10^{-10} \end{aligned}$$

2. (1,0) Considere um sistema onde o número de ciclos de relógio para uma operação por programa é igual a 200, o processador utiliza um relógio de 2GHZ para executar as instruções e nenhuma transferência de dados pode ser perdida. Determine o overhead, em termos de fração de tempo de CPU consumida, que ocorre quando se utiliza a interface por programa para os seguintes dispositivos:

a) Um mouse que deve ser interrogado pelo sistema 24 vezes por segundo para garantir que nenhum movimento dele seja perdido.

O mouse é consultado 24 vezes por segundo. Cada operação de consulta consome 200 ciclos. São necessários, então, $24 \times 200 = 4800$ ciclos/segundo para as operações de consulta ao mouse. O relógio é de 2GHz, que corresponde a 2×10^9 ciclos. O overhead será obtido pela seguinte razão:
 $4800 \text{ ciclos/seg} / (2 \times 10^9 \text{ ciclos/seg}) = 0,0000024 \text{ ou } 0,00024\%$

b) Um CD-ROM que transfere dados para o processador em unidades de 16 bits e possui uma taxa de transferência de dados de 200KB/segundo.

Taxa de transferência = 200KB/s = 200.000 B/s. Em cada operação é transferida uma unidade de 16 bits ou 2 bytes. A cada segundo ocorrem $200.000 \text{ B} / 2 \text{ B} = 100.000$ operações, como cada operação leva 200 ciclos, então, o total de ciclos será $100.000 \times 200 = 20 \times 10^6$ ciclos/s. O relógio é de 2GHz, que corresponde a 2.000×10^6 ciclos/s. O overhead será obtido pela seguinte razão:
 $20 \times 10^6 \text{ ciclos/s} / 2.000 \times 10^6 \text{ ciclos/s} = 0,01 \text{ ou } 1\%$

OBS: as taxas de transferência utilizam a base 10, e não a base 2. 1KB/s corresponde a 1.000B/s e 1MB/s a 1.000.000B/s.

c) Um disco rígido que transfere dados para o processador em unidades de 32 bits e possui uma taxa de transferência de 5MB/segundo.

Taxa de transferência = 5MB/s = 5.000.000 B/s. Em cada operação é transferida uma unidade de 32 bits ou 4 bytes.

A cada segundo ocorrem $5.000.000B / 4B = 1.250.000$ operações, como cada operação leva 200 ciclos, então, o total de ciclos será $1.250.000 \times 200 = 250 \times 10^6$ ciclos/s.
O relógio é de 2GHz, que corresponde a 2.000×10^6 ciclos/s.
O overhead será obtido pela seguinte razão:
 $250 \times 10^6 \text{ ciclos/seg} / 2.000 \times 10^6 \text{ ciclos/s} = 0,125$ ou 12,5%

3. (1,0) Explique como funcionam os três mecanismos utilizados para transferir dados entre um dispositivo de E/S e a memória de um sistema de computação: por programa (polling), por interrupção e por acesso direto à memória.

E/S por programa: O processador tem controle direto sobre a operação de E/S, incluindo a detecção do estado do dispositivo, o envio de comandos de leitura ou escrita e a transferência de dados. Para realizar uma transferência de dados, o processador envia um comando para o módulo de E/S e fica monitorando o módulo para identificar o momento em que a transferência pode ser realizada. Após detectar que o módulo está pronto, a transferência de dados é realizada através do envio de comandos de leitura ou escrita pelo processador. Se o processador for mais rápido que o módulo de E/S ele terá que aguardar, essa espera representa um desperdício de tempo de processamento. As vantagens deste método são: hardware simples e todos os procedimentos estão sob controle da UCP. As desvantagens são: utilização do processador para interrogar as interfaces, ocasionando a perda de ciclos de processamento que poderiam ser utilizados na execução de outras instruções, utilização do processador para realizar a transferência de dados, o que também ocasiona a perda de ciclos do processamento.

E/S por interrupção: Neste caso, o processador envia um comando para o módulo de E/S e continua a executar outras instruções, sendo interrompido pelo módulo quando este estiver pronto para realizar a transferência de dados. A transferência é executada pelo processador obtendo os dados da memória principal em uma operação de saída, ou armazenando dados na memória principal em uma operação de entrada. A vantagem deste método é que não ocorre perda de ciclos de processador para interrogar a interface, já que neste caso, não se precisa mais interrogar a interface, ela avisa quando pronta. As desvantagens são: a necessidade de um hardware adicional (controlador de interrupções, por exemplo), o gerenciamento de múltiplas interrupções e perda de ciclos de relógio para salvar e recuperar o contexto dos programas que são interrompidos.

E/S por DMA: Nesse caso, a transferência de dados entre o módulo de E/S e a memória principal é feita diretamente sem envolver o processador. Existe um outro módulo denominado controlador de DMA que realiza a transferência direta de dados entre a memória e o módulo de E/S. Quando o processador deseja efetuar a transferência de um bloco de dados com um módulo de E/S, ele envia um comando para o controlador de DMA indicando o tipo de operação a ser realizada (leitura ou escrita de dados), endereço do módulo de E/S envolvido, endereço de memória para início da operação de leitura ou escrita de dados e número de palavras a serem lidas ou escritas. Depois de enviar estas informações ao controlador de DMA, o processador pode continuar executando outras instruções. O controlador de DMA executa a transferência de todo o bloco de dados e ao final envia um sinal de interrupção ao processador, indicando que a transferência foi realizada. As vantagens deste método são: permite a transferência rápida entre interface e memória porque existe um controlador dedicado a realizá-la liberando a UCP para executar outras instruções não relacionadas a entrada e saída. A desvantagem é que precisamos de hardware adicional.

4. (1,0) Explique como funcionam os monitores de vídeo de cristal líquido (LCD) (sugestões de fonte de consulta: livro do Mário Monteiro e a página: <http://www.infowester.com/monlcd.php>)

Fontes: <http://www.hardware.com.br/livros/hardware-manual/como-funciona-lcd.html>
<http://www.infowester.com/monlcd.php>
Livro texto do Mário Monteiro

Tal como o seu nome indica, o "segredo" do LCD está em um material chamado cristal líquido. Esse tipo de cristal se situa em um estado intermediário entre o líquido e o sólido e é sensível a campos elétricos.

Quando uma voltagem é aplicada, as moléculas do cristal modificam sua forma, alternando a polarização da luz incidente. Em sua constituição mais simples, as moléculas de cristal líquido são distribuídas entre duas lâminas transparentes polarizadas chamadas substratos. Este processo é orientado de maneira diferente nas duas lâminas, de forma que estas formem eixos polarizadores perpendiculares, como se formassem um ângulo de 90°. A grosso modo, é como se uma lâmina recebesse polarização horizontal, e a outra, polarização vertical, formando um esquema do tipo "linhas e colunas".

Em uma linguagem mais clara, os cristais líquidos são substâncias que tem sua estrutura molecular alterada quando recebem corrente elétrica (Figura 1). Em seu estado normal, estas substâncias são transparentes, mas ao receberem uma carga elétrica tornam-se opacas, impedindo a passagem da luz.

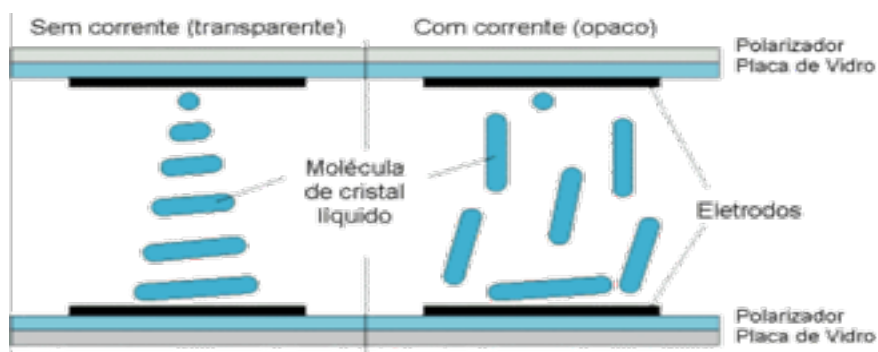


Figura 1: Estrutura molecular do LCD.

Fonte: <http://www.hardware.com.br/livros/hardware-manual/como-funciona-lcd.html>

Em telas monocromáticas (comuns em relógios e calculadoras, por exemplo), as moléculas assumem dois estados: transparente (a luz passa) e opaco (a luz não passa). Nos visores mais sofisticados, como os usados em notebooks, temos também estados intermediários, que formam as tonalidades de cinza ou as cores através do uso de filtros sobre a luz branca. Estes tons intermediários são obtidos usando-se tensões diferentes.

As telas LCD se dividem, basicamente, em duas categorias: matriz ativa (Active Matrix LCD) e matriz passiva (Passive Matrix LCD). O primeiro tipo tem como principal diferença a aplicação de transistores para cada pixel (em poucas palavras, pixel é um ponto que representa a menor parte da imagem em uma tela), enquanto que, na matriz passiva, os transistores são aplicados tomando como base o já mencionado esquema de linhas e colunas. Com isso, nas telas de matriz ativa, cada pixel desta pode receber uma tensão diferente, permitindo, entre outros, a utilização de resoluções altas. Por outro lado, sua fabricação é tão complexa que não é raro encontrar monitores novos que possuem pixels que não funcionam - os chamados "dead pixels". As telas de matriz passiva, por sua vez, têm constituição mais simples, com o cristal líquido posicionado entre dois substratos. Circuitos integrados são encarregados de controlar as cargas que ativam os pixels, permitindo que as imagens sejam formadas na tela.

Para formar a tela de um monitor, uma fina camada de cristal líquido é colocada entre duas camadas de vidro. Estas finas placas possuem pequenos sulcos, isolados entre si, cada um com um eletrodo ligado a um transistor. Cada um destes sulcos representa um dos pontos da imagem. Este sanduíche por sua vez é colocado entre duas camadas de um elemento polarizador. Atrás desta tela é instalada uma fonte de luz, geralmente composta de lâmpadas fluorescentes (usadas por gerarem pouco calor) ou então LEDs, responsáveis pela iluminação da tela.

5. (2,0) Crie um conjunto de instruções de dois operandos, definidas em Linguagem Assembly, necessárias para a realização de operações aritméticas e elabore programas para o cálculo das seguintes equações:

a) $X = A + (B * (C - A) + (D - E / B) * D)$

b) $Y = (A + B * (C - D * (E / (B - F)) + B) * E)$

I_ CONJUNTO DE INSTRUÇÕES PARA 2 OPERANDOS:

```

ADD X,Y    =>  (X) <- (X) + (Y)
SUB X,Y    =>  (X) <- (X) - (Y)
MUL X,Y    =>  (X) <- (X) * (Y)
DIV X,Y    =>  (X) <- (X) / (Y)
MOV X,Y    =>  (X) <- (Y)

```

a) $X = A + (B * (C - A) + (D - E / B) * D)$

```

MOV X, C    =>  (X) <- (C)
SUB X, A    =>  (X) <- (X) - (A)
MUL X, B    =>  (X) <- (X) * (B)
MOV T1, E   =>  (T1) <- (E)
DIV T1, B   =>  (T1) <- (T1) / (B)
MOV T2, D   =>  (T2) <- (D)
SUB T2, T1  =>  (T2) <- (T2) - (T1)
MUL T2, D   =>  (T2) <- (T2) * (D)
ADD X, T2   =>  (X) <- (X) + (T2)
ADD X, A    =>  (X) <- (X) + (A)

```

b) $Y = (A + B * (C - D * (E / (B - F)) + B) * E)$

```

MOV T1, B   =>  (T1) <- (B)
SUB T1, F   =>  (T1) <- (T1) - (F)
MOV T2, E   =>  (T2) <- (E)
DIV T2, T1  =>  (T2) <- (T2) / (T1)
MUL T2, D   =>  (T2) <- (T2) * (D)
MOV T1, C   =>  (T1) <- (C)
SUB T1, T2  =>  (T1) <- (T1) - (T2)
ADD T1, B   =>  (T1) <- (T1) + (B)
MUL T1, B   =>  (T1) <- (T1) * (B)
MUL T1, E   =>  (T1) <- (T1) * (E)
ADD T1, A   =>  (T1) <- (T1) + (A)
MOV Y, T1   =>  (Y) <- (T1)

```

6. (3,0) Descreva:

a) Os modos de endereçamentos, explicitando suas aplicações, vantagens e desvantagens.

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

Por registrador: característica semelhante aos modos direto e indireto, exceto que a célula (ou palavra) de memória referenciada na instrução é substituída por um dos registradores da UCP. O endereço mencionado na instrução passa a ser o de um dos registradores. Vantagens: Menor quantidade de bits para endereçar os registradores, por consequência, redução do tamanho da instrução. E o dado pode ser armazenado em um meio mais rápido (registrador). Desvantagens: O número reduzido de registradores existentes na UCP causa uma dificuldade em se definir quais dados serão armazenados nos registradores e quais permanecerão na UCP.

No modo indexado: neste modo o endereço do dado é a soma do valor do campo operando (que é fixo para todos os elementos de um dado vetor) e de um valor armazenado em um dos registradores da UCP

(normalmente denominado registrador índice).

Vantagem: Rapidez de execução das instruções de acesso aos dados, visto que a alteração do endereço dos elementos é realizada na própria UCP

No modo de endereçamento base mais deslocamento o endereço é obtido da soma do campo de deslocamento com o conteúdo do registrador base. Este modo de endereçamento tem como principal objetivo permitir a modificação de endereço de programas ou módulos destes, bastando para isso alterar o registrador base.

Vantagem: Reduz o tamanho das instruções e facilita o processo de relocação de programas.

b) Os modos compilação e interpretação, indicando em que circunstâncias um modo é mais vantajoso do que o outro.

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.

c) Computadores matriciais

Os computadores matriciais correspondem a sistemas compostos por processadores matriciais cuja organização é formada de ULAs paralelas. Uma unidade de controle distribui uma sequência de instruções em broadcast para todas as ULAs. As ULAs executam a mesma sequência de instruções, mas sobre conjunto de dados diferentes.

d) Sistemas SMP

Sistemas SMP (ou UMA) têm como característica o acesso a todas as partes da memória principal com tempo de acesso uniforme. Ocorre em um sistema computacional com vários processadores com memória compartilhada sob controle de um único sistema operacional. A memória principal da máquina é compartilhada por todos os processadores através de um único barramento que os interliga.

Nos sistemas SMP, o aumento no número de processadores tem como consequência problemas de tráfego no barramento comum degradando o desempenho. Uma solução para isto é a utilização de clusters, que tem, usualmente, como consequência alterações significativas na aplicação (software).

e) Sistemas NUMA

Em sistemas NUMA, todos os processadores possuem acesso a todas as partes da memória principal podendo diferir o tempo de acesso em relação às posições da memória e processador. Um processador pode acessar sua própria memória mais rapidamente do que uma memória não local (memória local de outro computador ou compartilhada entre os processadores). Nos sistemas NUMA podemos ter vários nós multiprocessadores, cada qual com seu próprio barramento, resultando em pequenas alterações na aplicação (software).