



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Organização de Computadores

GABARITO- AP3 2º semestre de 2007.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1. (3,0) Um computador, que apresenta uma arquitetura similar àquela apresentada ao longo do curso, possui uma capacidade máxima de memória principal de 256K células, cada uma capaz de armazenar uma palavra de 16 bits. Em cada acesso à memória, realiza-se a transferência de uma palavra.

- a) Qual é o maior endereço em decimal desta memória?

O maior endereço deverá ser $256K - 1 = 256 \times 1024 - 1 = 262.143$

- b) Qual é o tamanho do barramento de endereços deste sistema?

O barramento de endereços deverá endereçar 256K células = N

Barramento de endereços = E, sendo $N = 256K \Rightarrow N = 2^{18} \Rightarrow E = 18$ bits

Barramento de endereços = 18 bits

- c) Quantos bits podem ser armazenados no RDM (Registrador de Dados da Memória), no REM (Registrador de Endereços da Memória) e no CI (Contador de Instruções) ?

$\Rightarrow \text{REM} = \text{Barramento de endereços} = 18$ bits

$\Rightarrow \text{RDM} = \text{Barramento de dados}$

Como a quantidade de bits que podem ser transferidos pelo barramento de dados corresponde a uma palavra (16 bits), então, o barramento de dados terá o tamanho de uma palavra = 16 bits.

RDM= Barramento de dados = 16 bits

=> CI terá o tamanho para endereçar toda a memória, CI corresponderá ao tamanho do barramento de endereços, CI = 18bits

d) Qual é o número máximo de bits que pode existir na memória ?

Como o tamanho da memória = 256Kcélulas (N) e o tamanho da célula = 16bits (M).

O total de bits na memória (T) será $M \times N = 256Kcélulas \times 16bits = 4M \text{ bits}$

e) Descreva detalhadamente a execução da instrução **LDA Op.** nesta máquina. A instrução **LDA Op.** carrega o acumulador com o conteúdo da célula de memória cujo endereço é Op.

Passo 1: A CPU coloca no REM o valor do operando ($REM \leftarrow Op$), e é disponibilizado no barramento de endereço

Passo 2: A CPU aciona pelo barramento de controle a leitura de memória

Passo 3: A memória coloca o valor no barramento de dados, e por consequência no RDM da CPU ($RDM \leftarrow MP(Op)$)

Passo 4: O valor do RDM vai ser transferido pelo barramento interno para o acumulador (ACC)

f) Descreva detalhadamente a execução da instrução **STR Op.** nesta máquina. A instrução **STR Op.** armazena o conteúdo do acumulador na célula de memória cujo endereço é Op.

Passo 1: A CPU coloca no REM o valor do operando ($REM \leftarrow Op$), e é disponibilizado no barramento de endereço

Passo 2: A CPU transfere o conteúdo do Acumulador (ACC) para o RDM e este disponibiliza para o barramento de dados

Passo 3: A CPU aciona pelo barramento de controle a escrita de memória

Passo 4: A memória armazena o valor contido no barramento de dados no endereço apontado pelo barramento de endereços. ($MP(Op) \leftarrow RDM$)

2. (2,0) Compare máquinas de 1 e 2 endereços escrevendo programas para calcular:

$$X = C \times (A - B) / D$$

M, X e Y são endereços de memória de 16 bits. A máquina de 1 endereço usa um acumulador, e a outra tem instruções operando sobre endereços de memória. Por exemplo, SUB X,Y subtrai Y de X e coloca o resultado em X. Assumindo códigos de operação de 8 bits e comprimentos de instruções que são múltiplos de 4, quantos bits cada máquina precisa para calcular X?

1. operador:

LOAD M
STORE M
ADD M
SUB M
MUL M
DIV M

2. operadores

MOV X, Y equivale $(X=Y)$
ADD X, Y equivale $(X=X+Y)$
SUB X, Y equivale $(X=X-Y)$
MUL X, Y equivale $(X=X*Y)$
DIV X, Y equivale $(X=X/Y)$

=> Máquina 1 (utiliza 1 operando)

Sabendo que:

LOAD M	equivale	$ACC \leftarrow (M)$
STORE M	equivale	$(M) \leftarrow ACC$
ADD M	equivale	$ACC \leftarrow ACC + (M)$
SUB M	equivale	$ACC \leftarrow ACC - (M)$
MUL M	equivale	$ACC \leftarrow ACC * (M)$
DIV M	equivale	$ACC \leftarrow ACC / (M)$

Solução para $X = C \times (A-B) / D$

LOAD A	=> $ACC \leftarrow (A)$	=> Cód. Oper. + 1 operando = 8bits + 16 bits = 24bits
SUB B	=> $ACC \leftarrow ACC - (B)$	=> Cód. Oper. + 1 operando = 8bits + 16 bits = 24bits
MUL C	=> $ACC \leftarrow ACC * (C)$	=> Cód. Oper. + 1 operando = 8bits + 16 bits = 24bits
DIV D	=> $ACC \leftarrow ACC / (D)$	=> Cód. Oper. + 1 operando = 8bits + 16 bits = 24bits
STORE X	=> $(X) \leftarrow ACC$	=> Cód. Oper. + 1 operando = 8bits + 16 bits = 24bits

Total (máquina 1) = $5 \times 24\text{bits} = 120\text{bits}$

=> Máquina 2 (utiliza 2 operandos)

Solução para $X = C \times (A-B) / D$

MOV X, A	=> $X \leftarrow A$	=> Cód. Oper. + 2 operandos = 8bits + 32 bits = 40bits
SUB X, B	=> $X \leftarrow X - B$	=> Cód. Oper. + 2 operandos = 8bits + 32 bits = 40bits
MUL X, C	=> $X \leftarrow X * C$	=> Cód. Oper. + 2 operandos = 8bits + 32 bits = 40bits
MUL X, D	=> $X \leftarrow X * D$	=> Cód. Oper. + 2 operandos = 8bits + 32 bits = 40bits

Total (máquina 2) = $4 \times 40\text{bits} = 160\text{bits}$

Concluindo, para execução da expressão fornecida, serão necessários mais bits para a máquina 2 (180bits) do que para a máquina 1 (120bits).

3. (1,5) Explique a os principais tipos de endereçamento.

- **Imediato:** Seu campo operando contém o dado, não requer acessos a memória principal sendo mais rápido que o modo direto. Possui como vantagem a rapidez na execução da instrução e como desvantagem a limitação do tamanho do dado, e é inadequado para o uso com dados de valor variável.

LOAD X => Corresponde a ACC <- X (o acumulador receberá o valor X)

- **Direto:** Seu campo operando contém o endereço do dado, requer apenas um acesso a memória principal, sendo mais rápido que o modo indireto. Possui como vantagem a flexibilidade no acesso a variáveis de valor diferente em cada execução do programa e como desvantagem a perda de tempo, se o dado for uma constante.

LOAD X => Corresponde a ACC <- (X) (o acumulador receberá o conteúdo do endereço X)

- **Indireto:** O campo operando corresponde ao endereço que contém a posição onde está o conteúdo desejado, necessita de 2 acessos a memória principal, portanto mais lento que os 2 modos anteriores. Tem como vantagem o manuseio de vetores e utilização como ponteiro, e desvantagem como muitos acessos a memória principal.

LOAD X => Corresponde a ACC <- ((X)) (o acumulador receberá o conteúdo do endereço contido em X)

- **Por registrador:** Semelhante aos modos direto e indireto, mas o endereço de memória é substituído por um endereço de registrador da UCP, possui como vantagem a velocidade de obtenção do dado por estar armazenado em um meio mais rápido e a instrução com menor número de bits

- **Modo Indexado:** O endereço do dado é a soma do valor do campo operando (fixo) e de um valor armazenado em um dos registradores da UCP (registrador índice). Tem o objetivo de permitir a manipulação de endereços para facilitar o acesso a vetores.

- **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.

4. (2,0) Descreva **detalhadamente** os três possíveis métodos de comunicação entre o controlador de Entrada/Saída com a unidade central de processamento e memória principal: por E/S programada, por interrupção e por acesso direto à memória. Indique as vantagens e desvantagens de cada método.

E/S programada: 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 transferência de dados. Para realizar uma transferência de dados, o processador envia um comando para o controlador de E/S e fica monitorando o controlador para identificar o momento em que a transferência pode ser realizada. Após detectar que o controlador 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 controlador de E/S, essa espera representa um desperdício de tempo de processamento. As vantagens deste método são: hardware simples e todos os procedimentos estão sobre controle da UCP. As desvantagens são: utilização do processador para interrogar as interfaces, o que acarreta perda de ciclos de processador que poderiam ser utilizados na execução de outras instruções e utilização do processador para realizar a transferência de dados, o que também acarreta perda de ciclos de processador.

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

E/S por acesso direto à memória: Nesse caso a transferência de dados entre o controlador de E/S e a memória principal é feita diretamente sem envolver o processador. Existe um outro módulo denominado controlador de DMA ou ADM que realiza a transferência direta de dados entre a memória e o controlador de E/S. Quando o processador deseja efetuar a transferência de um bloco de dados com um controlador 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 transferência rápida entre controlador e memória porque existe um controlador dedicado a realizá-la e libera a UCP para executar outras instruções não relacionadas a entrada e saída. A desvantagem é que precisamos de hardware adicional.

5. (1,5) Considere uma máquina que utiliza 32 bits para representar números em ponto fixo e em ponto flutuante.

- a) (0,5) Mostre a representação de -55,0 utilizando-se a representação ponto flutuante precisão dupla IEEE 754 (1 bit de sinal, 8 bits para expoente em excesso de 127, e 23 bits para mantissa)

Sinal negativo, bit igual a 1

$$55,0 = 110111 = 1,10111 \times 2^5$$

Bits da mantissa = 10111000000000000000000

Bits do expoente : 5 representado em excesso de 127 = 10000100

Representação: 11000010010111000000000000000000

- b) Para o conjunto de bits obtido no item anterior, indique o que ele representa na base 10, considerando-se as seguintes representações: **(Não precisa fazer as contas, deixe-as indicadas):**

- i. (0,4) um inteiro sem sinal

$$2^{31} + 2^{30} + 2^{25} + 2^{22} + 2^{20} + 2^{19} + 2^{18} = 3260809216$$

- ii. (0,6) um inteiro utilizando-se a representação em complemento a 2

$$-2^{31} + 2^{30} + 2^{25} + 2^{22} + 2^{20} + 2^{19} + 2^{18} = -1034158080 \text{ ou}$$

$$- \text{Inv}(11000010010111000000000000000000) + 1 =$$

$$-(00111101101001000000000000000000) =$$

$$-(2^{29} + 2^{28} + 2^{27} + 2^{26} + 2^{24} + 2^{23} + 2^{21} + 2^{18}) = -1034158080$$