



AD2 - Organização de Computadores 2017.1

Data de entrega: 23/05/2017

- 1) (1,0) Descreva os modos endereçamento existentes, suas vantagens, desvantagens e aplicações.

Abaixo seguem os modos de endereçamento abordados em aula:

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: 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, 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 da instrução. E o dado pode ser armazenado em um meio mais rápido (registrador).

Desvantagens: Devido ao 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: consiste em que 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.

Desvantagem: a complexidade da instrução.

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.

Desvantagem: a complexidade da instrução, como no modo indexado.

Além dos apresentados acima, podemos citar outros modos de endereçamento (fonte livro do Willian Stallings), em sua maior parte variações dos anteriores.

Modo de operando registrador: o valor a ser utilizado na instrução é localizado em um registrador. Para instruções de caráter geral, tais como transferências de dados e instruções aritméticas ou lógicas, o operando pode ser um dos registradores de propósito geral de 32bits, 16 bits ou 8 bits.

Ex. $LA = R$ (sendo R = registrador)

Modo de endereçamento por deslocamento: O endereço relativo do operando é especificado como parte da instrução, como um deslocamento de 8, 16 ou 32 bits. O modo de endereçamento por deslocamento é usado em poucas máquinas porque implica em instruções com tamanho grande.

Ex. $LA = (SR) + A$ (sendo SR = registrador de segmento, A = conteúdo de um campo de endereço na instrução)

Modo base com deslocamento: A instrução inclui um deslocamento a ser adicionado a um registrador-base, que pode ser qualquer registrador de uso geral. Possui utilização em compiladores para apontar para início da área de variáveis locais, indexamento de vetores, endereçamento de campos de registro, sendo o tamanho do campo o deslocamento e registrador-base o início do campo (ou registro).

Ex. $LA = (SR) + (B) + A$ (sendo SR = registrador de segmento, B = registrador-base, A = conteúdo de um campo de endereço na instrução)

Modo índice com fator de escala e deslocamento: a instrução inclui um deslocamento que é somado a um registrador índice. O registrador índice pode ser qualquer um registradores de uso geral, exceto o ESP (utilizado para endereçamento de pilha). Para calcular o endereço efetivo, o conteúdo do registrador índice é multiplicado por um fator de escala igual a 1,2,4 ou 8, e então é adicionado ao deslocamento, sendo muito útil para vetores.

Ex. $LA = (SR) + (I) \times S + A$ (sendo SR = registrador de segmento, I = registrador índice, S = fator de escala, A = conteúdo de um campo de endereço na instrução)

Modo base mais índice e deslocamento: o endereço efetivo é obtido somando os conteúdos do registrador-base, do registrador índice e do deslocamento, tem seu uso direcionado para endereçamento de elementos em um vetor local e um registro de ativação localizado em pilha.

Ex. $LA = (SR) + (B) + (I) + A$ (sendo SR = registrador de segmento, I = registrador índice, B = registrador-base, A = conteúdo de um campo de endereço na instrução)

Modo base mais índice com fator de escala e deslocamento: o conteúdo do registrador índice é multiplicado por um fator de escala e somado com o conteúdo do registrador-base e o deslocamento. Tem seu uso direcionado a vetores que estejam armazenados em um registro de ativação; sendo o tamanho dos elementos do vetor iguais a 2, 4 ou 8 bytes. Permite também indexação para matrizes de 2 dimensões, com elementos do mesmo tamanho do vetor citado.

Ex. $LA = (SR) + (I) \times S + (B) + A$ (sendo SR = registrador de segmento, I = registrador índice, S = fator de escala, B = registrador-base, A = conteúdo de um campo de endereço na instrução).

Modo relativo: Onde um deslocamento é adicionado ao valor do contador de programa, que aponta para a próxima instrução a ser executada, sendo o deslocamento tratado como um valor com sinal permitindo aumentar ou diminuir o valor do endereço no contador de programa.

Ex. $LA = (PC) + A$ (sendo PC = contador de instruções, A = conteúdo de um campo de endereço na instrução).

- 2) (1,0) Que modo de endereçamento você utilizaria em um campo de operando de 8 bits para endereçar 256 endereços não contíguos? Explique.

Uma solução seria usar endereçamento por registrador base mais deslocamento. Por exemplo, teríamos 2 bits para especificar um registrador e 6 bits para especificar um deslocamento. Poderíamos, assim, usar 4 registradores, cada um com até 64 deslocamentos possíveis, fornecendo 256 endereços diferentes.

- 3) (1,0) Explique os modos de tradução de interpretação e de compilação, especificando vantagens e desvantagens de cada um.

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 código 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.

- 4) (1,0) Explique a classificação das arquiteturas segundo Flynn.

Consiste em uma das formas mais comuns de classificação de processamento paralelo. São estas as categorias de sistemas de computação:

SISD - Single instruction stream, single data stream. Um único processador executa uma única sequência de instruções sobre dados armazenados em uma única memória. Exemplo: Processadores de computadores pessoais.

SIMD - Single instruction stream, multiple data stream. Vários elementos de processamento. Cada um tem uma memória de dados. Cada instrução é executada sobre um conjunto de dados diferente. Exemplo: Processadores matriciais.

MISD - Multiple instruction stream, single data stream. A sequência de dados é transmitida para um conjunto de processadores, cada um dos quais executa uma sequência de instruções diferente. Não existem processadores comerciais que utilizam este modelo.

MIMD - Multiple instruction stream, multiple data stream. Conjunto de processadores que executa simultaneamente sequências diferentes de instruções sobre conjuntos de dados diferentes. Exemplo: SMPs, clusters, sistemas NUMA.

- 5) (1,0) Classifique as arquiteturas abaixo como SISD, MIMD, SIMD ou MISD, justificando.

a) Cluster de processadores

É um MIMD, pois um cluster de processadores é composto de processadores independentes, esse conjunto executa simultaneamente sequências diferentes de instruções sobre conjunto de dados diferentes.

b) Máquina de arquitetura matricial

É um SIMD, pois uma arquitetura matricial possui vários elementos de processamento que executam a mesma sequência de instruções simultaneamente. Cada elemento de processamento tem uma memória local. Assim, cada elemento de processamento executa a mesma instrução sobre um conjunto de dados diferente.

c) Computador pessoal com um único processador

É um SISD, pois um computador com um único processador executa uma única sequência de instrução sobre uma sequência de dados armazenados em uma única memória principal.

d) Computador com múltiplos processadores.

É um MIMD, um computador com múltiplos processadores executam simultaneamente sequências diferentes de instruções sobre conjunto de dados diferentes.

6) (3,0) Considere um computador, cuja representação para ponto fixo e para ponto flutuante utilize 20 bits. Na representação em ponto flutuante, as combinações possíveis e 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} a b_{-13} , como mostrado na figura a seguir:

Sinal	Expoente representado em complemento a 2	$b_{-1}b_{-2}b_{-3}b_{-4}b_{-5}b_{-6}b_{-7}b_{-8}b_{-9}b_{-10}b_{-11}b_{-12}b_{-13}$
1	6	13

a) Considere o seguinte conjunto de bits representado em hexadecimal 9B500. Indique o valor deste número em decimal, considerando-se que o conjunto representa: $(A9600)_{16}$

DEVIDO A DIVERGÊNCIA DOS DOIS VALORES EM HEXADECIMAL CONSTANTES DO TÍTULO DA QUESTÃO, AO CORRIGIREM A AD2, SERÁ CONSIDERADA A RESPOSTA QUE ATENDA A QUALQUER UM DOS DOIS VALORES CITADOS.

PARA O VALOR $9B500_{16} = 1001\ 1011\ 0101\ 0000\ 0000_2$

i) um inteiro sem sinal.

$$2^{19} + 2^{16} + 2^{15} + 2^{13} + 2^{12} + 2^{10} + 2^8 = +636.160$$

ii) um inteiro em sinal magnitude.

$$-(2^{16} + 2^{15} + 2^{13} + 2^{12} + 2^{10} + 2^8) = -111.872$$

iii) um inteiro em complemento a 2.

$$-2^{19} + (2^{16} + 2^{15} + 2^{13} + 2^{12} + 2^{10} + 2^8) = -412.416$$

PARA O VALOR $A9600_{16} = 1010\ 1001\ 0110\ 0000\ 0000_2$

i) um inteiro sem sinal.

$$2^{19} + 2^{17} + 2^{15} + 2^{12} + 2^{10} + 2^9 = +693.760$$

ii) um inteiro em sinal magnitude.

$$-(2^{17} + 2^{15} + 2^{12} + 2^{10} + 2^9) = -169.472$$

iii) um inteiro em complemento a 2.

$$-2^{19} + (2^{17} + 2^{15} + 2^{12} + 2^{10} + 2^9) = -354.816$$

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

i) +147,375

Convertendo para binário = $10010011,011 = 1,0010011011 \times 2^7$

Temos então:

Sinal = 0 (positivo)

Expoente = +7 = (representando em complemento a 2) = 000111

Mantissa = 0010011011

Resultado: 0 000111 0010011011000

ii) -31,5

Convertendo para binário = 11111,1 = 1,11111 x 2⁺⁴

Temos então:

Sinal = 1 (negativo)

Expoente = +4 = (representando em complemento a 2) = 000100

Mantissa = , 11111

Resultado: 1 000100 1111100000000

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

Maior positivo: 0 011111 111111111111 =

$1,111111111111 \times 2^{+31} = +4.294.443.008$

Menor positivo: 0 100000 000000000000 =

$1,000000000000 \times 2^{-32} = +2,328 \times 10^{-10}$

e) 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 = $2^{e-1} - 1$, sendo e = 6, excesso=31

Menor positivo 0 000001 000000000000 =

$1,000000000000 \times 2^{-30} = +9,313 \times 10^{-10}$

Maior positivo 0 111110 111111111111 =

$1,111111111111 \times 2^{+31} = +4.294.443.008$

7) (2,0) 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