

Trabalho Prático 1 - Simulador NEANDER

Escrever um programa para o simulador Neander que multiplique dois valores inteiros positivos de 8 bits e armazene o produto obtido (16 bits) em duas palavras consecutivas da memória. Os valores dados e calculados devem estar obrigatoriamente nas seguintes posições:

- Palavra 128 - multiplicando
- Palavra 129 - multiplicador
- Palavra 130 - oito bits menos significativos do produto
- Palavra 131 - oito bits mais significativos do produto

Os valores do multiplicando e do multiplicador não devem ser alterados pelo programa e, antes de iniciar a multiplicação, o programa deve garantir que as palavras 130 e 131 contêm o valor 0. Tendo em vista que o processador Neander é capaz apenas de fazer adições, o produto deve ser calculado por adições sucessivas. Como o Neander opera apenas com valores de 8 bits, a determinação dos bits mais significativos do produto deve ser feita adicionando 1 à palavra 131 cada vez que ocorrer um "vai um" na soma dos bits menos significativos.

Dicas:

1. O processador Neander não tem um código de condição para indicar a ocorrência de "vai um" na adição. Portanto, esta situação deve ser determinada pela análise do bit mais significativo dos valores somados e do bit mais significativo do resultado da soma. O exercício 5 do Capítulo 4 do livro texto é um exemplo de determinação de estouro na soma de inteiros em complemento de 2. Embora já tenhamos visto que "estouro" e "vai um" são conceitos distintos e independentes, a solução daquele exercício pode ser usada como referência para determinar a ocorrência de "vai um" na soma de inteiros positivos.
2. Como o valor máximo de cada número inteiro positivo no Neander é 255, serão necessárias no máximo 255 adições. Se em todas as adições ocorresse um "vai um", o valor máximo da palavra 131 seria 255. Portanto, como estamos armazenando o produto em 16 bits, nunca ocorre "estouro" nesta multiplicação.
3. Como a multiplicação é comutativa, o produto obtido somando o multiplicando tantas vezes quantas for o valor do multiplicador é o mesmo obtido fazendo tantas adições do multiplicador quantas for o valor do multiplicando. Se você se preocupa com velocidade na execução do programa, este fato pode ser usado para reduzir o número de instruções necessárias para calcular um determinado produto. O número de instruções e acessos à memória durante a execução de um programa é medido pelo simulador Neander. Use estas medidas para avaliar o desempenho de sua solução (*este item é apenas uma sugestão - não é obrigatório*).

Os trabalhos serão corrigidos de forma automática, com 20 pares de valores diferentes. Portanto, devem ser observadas rigorosamente as seguintes especificações:

- o código do programa deve iniciar no endereço 0 da memória
- a primeira instrução executável deve estar no endereço 0
- os endereços dos fatores e do produto devem ser exatamente os especificados acima
- usar para variáveis adicionais os endereços de memória de 132 em diante

O programa deverá ser entregue em disquete (ou outro meio de armazenamento removível), juntamente com uma documentação (em meio digital) contendo uma breve descrição do algoritmo utilizado e uma listagem do programa fonte usando mnemônicos simbólicos (LDA, STA, etc ...) para as instruções e indicando (em decimal) os endereços de memória nos quais as instruções ficam armazenadas. Para os nomes dos arquivos, utilize a letra inicial do seu primeiro nome, seguida do seu número de identificação. Assim, por exemplo, o aluno João José da Silva, cartão 123456, deve denominar o seu arquivo de J123456.MEM e J123456.TXT.

Data de Entrega: 03/05/2006 (Turma A) - no horário de aula

Exemplos de casos de teste (valores decimais; **x** indica que a palavra pode conter um valor qualquer)

Endereço	Dados		Produto	
	128	129	131	130
Caso 1	0	0	0	0
Caso 2	0	x	0	0
Caso 3	x	0	0	0
Caso 4	1	255	0	255
Caso 5	2	128	1	0
Caso 6	128	8	4	0
Caso 7	100	100	39	16

Caso 8	255	255	254	1
--------	-----	-----	-----	---