

Trabalho de Programação 1 - Processador RAMSES

1. Descrição Geral

O objetivo deste trabalho é escrever um programa para o processador Ramses (usando necessariamente o montador Daedalus para desenvolver o código fonte) que processe um vetor contendo valores de 16 bits (2 bytes por elemento) e forneça um resultado também em 16 bits. As operações a realizar sobre os elementos do vetor exercitam o deslocamento e rotação de dados e a soma de valores em 16 bits usando o conjunto de instruções do Ramses.

2. Dados de entrada e saída

O programa receberá como dados de entrada o endereço inicial do vetor e o vetor propriamente dito e deverá escrever o resultado do processamento numa variável de 16 bits na memória. Os valores dos elementos do vetor serão inteiros positivos diferentes de 0 e um elemento com valor zero (nos dois bytes) indicará o fim do vetor.

Para os dados de entrada e saída devem ser utilizadas, **obrigatoriamente**, as seguintes posições de memória:

Endereço	197	198	199
Dado	Endereço do vetor	Bits 15 a 8 do resultado	Bits 7 a 0 do resultado

O vetor estará na área de memória acima dos dados de entrada e saída (endereços 200 a 255) e é garantido que existirá um elemento marcador de fim (dois bytes com valor 0) no vetor.

3. Processamento

O processamento a ser realizado pelo programa consistirá em percorrer todos os elementos do vetor, até encontrar o elemento marcador de fim do vetor. Para cada elemento diferente de zero, deverão ser executadas as seguintes operações, nesta ordem:

1. Girar o primeiro byte do elemento para a direita em 4 bits
2. Girar o segundo byte do elemento para a direita em 4 bits
3. Permutar os valores do primeiro e do segundo byte
4. Somar o resultado obtido (com 16 bits) à variável que armazena o resultado

Após processados todos os elementos do vetor, o programa deve aplicar novamente as operações descritas nas etapas 1 a 3, acima, sobre a variável que armazena o resultado e terminar a execução.

Note que a operação descrita nas etapas 1 e 2 inverte os dígitos hexadecimais armazenados em um byte. Por exemplo, se o byte contiver o valor 10100101_2 ($A5_{16}$) ficará com 01011010_2 ($5A_{16}$). E a operação descrita no item 3, aplicada após as duas anteriores, vai resultar na inversão dos 4 dígitos hexadecimais do valor de 16 bits armazenado no elemento processado. Por exemplo, se o elemento contiver AB_{16} no primeiro byte e CD_{16} no segundo byte, após o processamento das etapas 1, 2 e 3 ele conterá os valores DC_{16} e BA_{16} , no primeiro e segundo bytes, respectivamente.

Exemplo: supondo que os dados de entrada para o programa sejam os seguintes:

Endereço	197	198	199	...	208	209	210	211	212	213	214	215	216	217
Dado (decimal)	208	?	?	...	1	2	5	26	43	60	77	94	0	0
Dado (hexadecimal)	D2	?	?	...	01	02	05	1A	2B	3C	4D	5E	00	00

O resultado do processamento das etapas 1 a 3 para os valores contidos no vetor será o seguinte:

Endereço do valor de entrada	208	209	210	211	212	213	214	215	216	217
Resultado (decimal)	32	16	161	80	195	178	229	212	0	0
Resultado (hexadecimal)	20	10	A1	50	C3	B2	E5	D4	00	00

Note que, dependendo do algoritmo usado para solucionar o problema, o programa não precisa armazenar os resultados de volta no vetor de entrada. Os endereços foram mostrados na tabela acima apenas para correlacionar os dados de entrada com o resultado correspondente. Na correção dos trabalhos será verificado apenas o valor da variável resultado (endereços 198 e 199), que será: $6EA6_{16}$ (a soma dos elementos processados será $6AE6_{16}$ e depois de aplicadas as operações 1 a 3 sobre o resultado ele conterá $6EA6_{16}$).

4. Correção dos Trabalhos

Os arquivos fonte do RAMSES entregues serão montados usando o montador DAEDALUS. A seguir serão aplicados 20 casos de teste, de forma sequencial e contínua (sem recarga ou inicialização da memória). A nota final do trabalho será proporcional ao número de casos de teste em que o programa produzir a resposta correta (cada caso de teste vale 5 pontos de um total de 100).

5. Bônus

Os programas que fornecerem resultados corretos para todos os 20 casos de teste poderão concorrer a um Bônus de Desempenho, no valor de 10 pontos (10% da nota). Esse bônus será concedido para as soluções que utilizarem o menor número de **acessos** à memória (aqueles mais rápidos). O desempenho será medido somando os números de **acessos** usados para solucionar todos os 20 casos de teste.

6. Entregáveis: o que deve ser entregue?

Somente os arquivos fonte (arquivos .RAD) escritos na linguagem simbólica do RAMSES com a solução do problema apresentado deverão ser entregues via Moodle da disciplina. **Esses arquivos serão montados com o DAEDALUS.** O programa fonte deverá conter comentários descritivos da implementação. Por exemplo, nos comentários podem ser usados comandos em alto nível da linguagem "C". **Para nomear os arquivos, utilize todo o seu nome, usando maiúsculas e minúsculas, sem acentos.**

O trabalho deverá ser entregue até a data especificada no link de entrega no sistema Moodle. **Não serão aceitos trabalhos após o prazo estabelecido.**

7. Observações

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplinar Discente e a tomada das medidas cabíveis para essa situação (**tanto o trabalho original quanto os copiados receberão nota zero**).

O professor da disciplina reserva-se o direito, caso necessário, de solicitar uma demonstração do programa, onde o aluno será arguido sobre o trabalho como um todo. Nesse caso, a nota final do trabalho levará em consideração o resultado da demonstração.

8. Casos de Teste

Para testar sua solução, use os casos de teste abaixo, da seguinte forma:

1. Carregue seu código (arquivo .MEM gerado pelo Daedalus a partir do seu .RAD) no simulador Ramses
2. Coloque os valores indicados nas palavras 197 a 217 (as palavras em branco não precisam ser preenchidas)
3. Coloque o valor 0 no PC
4. Execute o programa
5. Verifique se os resultados nas palavras 198 e 199 são os mesmos indicados no final da linha de cada caso de teste
6. Para testar os demais casos, volte à etapa 2 **SEM RECARREGAR O CÓDIGO DE SUA SOLUÇÃO NO SIMULADOR**

		Dados de entrada																				Resultado		
Caso	End. →	197	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	...	198	199	
1	Dados	208									1	2	5	26	43	60	77	94	0	0	...	110	166	
2		200	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	0	0	...	0	88	
3		216																	0	0		0	0	
4		200	255	255	0	0																255	255	
5		210											15	240	0	0						15	240	
6		205						0	255	0	255	0	255	0	255	0	255	0	255	0	0	0	0	175
7		200	0	1	0	2	0	3	0	4	0	5	0	6	0	7	0	8	0	0		0	4	
8		200	1	0	2	0	3	0	4	0	5	0	6	0	7	0	8	0	0	0		4	32	
9		200	0	16	0	32	0	48	0	64	0	80	0	96	0	112	0	128	0	0		0	66	
10		205						16	32	48	64	80	96	0	0							144	192	