

Trabalho de Programação – Processador RAMSES

1. Descrição Geral

Nesse trabalho você deverá desenvolver um programa para o processador RAMSES que seja capaz de reorganizar os bits de um vetor de entrada, e escrevê-los nos bits de um vetor de saída.

O vetor de entrada pode ter um número qualquer de bytes, desde 1 (um) até 8 (oito). O vetor de saída sempre terá 8 bytes.

A reorganização dos bits deve ser feita da seguinte forma:

- Os bits de posição “0” dos bytes do vetor de entrada devem ser copiados para o primeiro byte do vetor de saída (posição [0] do vetor de saída);
- Os bits de posição “1” dos bytes do vetor de entrada devem ser copiados para o segundo byte do vetor de saída (posição [1] do vetor de saída);
- Isso deve ser repetido com os bits das posições 2, 3, 4, 5, 6 e 7, do vetor de entrada;
- Ou seja, os bits de posição “n-1” dos bytes do vetor de entrada devem ser copiados para o n-ésimo byte do vetor de saída (posição [n-1] do vetor de saída);

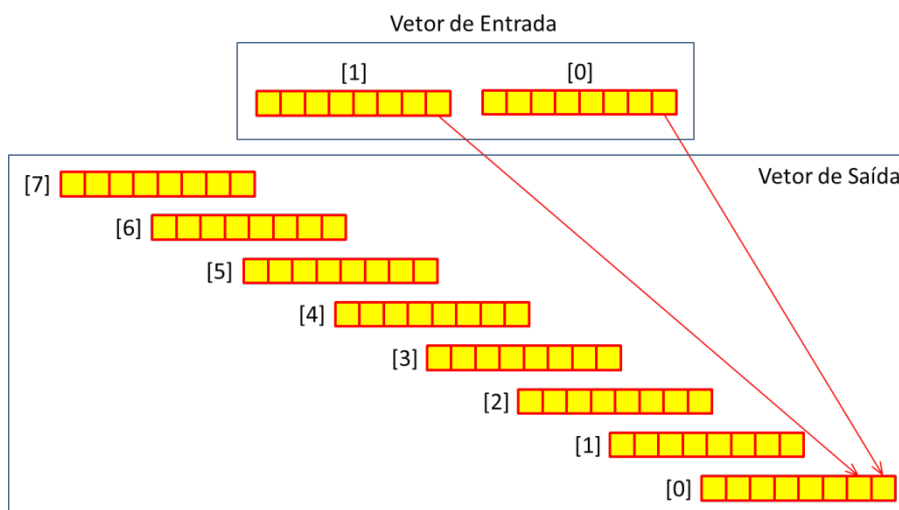
Além disso, para realizar cada uma das cópias de bits, a posição do bit de destino será definida pela posição do byte no vetor de entrada.

Por exemplo, vamos analisar o que ocorre com os bits de posição “0” dos bytes do vetor de entrada. Conforme explicado acima, esses bits devem ser colocados, todos, no primeiro byte do vetor de saída. Mas, como esses bits devem ser organizados no primeiro byte do vetor de saída? A resposta é que:

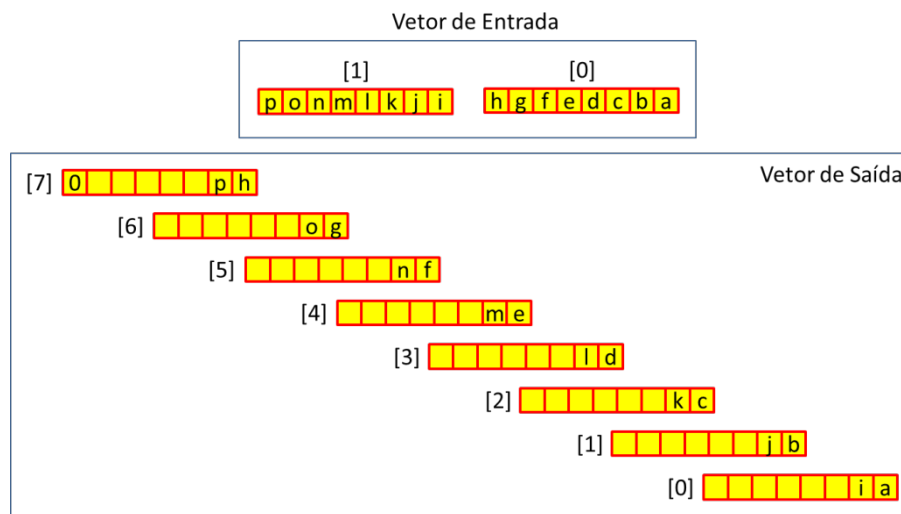
- O bit “0” obtido do byte [0] do vetor de entrada deve ser colocado no bit “0” do byte [0] do vetor de saída;
- O bit “0” obtido do byte [1] do vetor de entrada deve ser colocado no bit “1” do byte [0] do vetor de saída;
- E assim por diante, para os bits “0” dos bytes 2, 3, 4, 5, 6 e 7 do vetor de entrada, que devem ser colocados no bits 2, 3, 4, 5, 6 e 7, respectivamente, do byte [0] do vetor de saída.

Como o tamanho do vetor de entrada pode ter qualquer valor entre 1 e 8, pode haver bits não utilizados no vetor de saída. Esses bits não usados devem ser preenchidos com “0” (zero).

Na figura abaixo é apresentado a transferência de bits para o caso em que o vetor de entrada tem 2 bytes. Na figura, as setas indicam como deve ser feita a transferência dos bits “0” dos bytes do vetor de entrada. Verifica-se que o byte [0] do vetor de saída serão preenchidos com os bits “0” dos bytes do vetor de entrada.



Na próxima figura é apresentado o resultado final após a transferência de todos os bits do vetor de entrada (ainda com 2 bytes no vetor de entrada) para o vetor de saída. Os bits no vetor de entrada estão representados por letras minúsculas. As letras correspondentes estão representadas no vetor de saída. Verifica-se que apenas 2 bits de cada byte do vetor de saída receberão valores. Os outros 6 bits devem ser mantidos com “0”s (não representados por clareza da figura).



2. Disposição dos dados na memória

Informações de entrada

O endereço onde inicia o vetor de entrada (posição [0] do vetor) está na posição $C0_{16}$ (192_{10}). O tamanho do vetor de entrada, em bytes, está na posição $C1_{16}$ (193_{10}).

Os vetores de entrada e de saída estarão completamente contidos no espaço de memória que inicia em $C3_{16}$ (195_{10}) e termina em FF_{16} (255_{10}).

Informações de saída

O endereço onde inicia o vetor de saída (posição [0] do vetor) está na posição $C2_{16}$ (194_{10}).

3. Restrições e garantias

- Seu programa e todas as suas variáveis de trabalho devem estar restritos a região de memória 00_{16} até BF_{16} ;
- Seu programa deve inicializar, usando instruções, todas as variáveis necessárias para a sua correta execução. Lembramos que a diretiva “DB” é um mecanismo para a alocação de espaço na memória, não sendo um mecanismo de inicialização;
- Você não deve alterar a região reservada para os vetores, exceto na escrita do resultado do processamento no vetor de saída;
- É garantido que os vetores de entrada e saída estejam completamente contidos na região reservada para eles;
- É garantido que o tamanho do vetor de entrada será um valor entre “1” (um) e “8” (oito);
- O tamanho do vetor de saída é, sempre, 8 bytes;

4. Exemplo

Suponha que na memória estejam os seguintes dados:

Endereço	192	193	194	195	196
Dados (decimal)	195	2	224	85	170
Dados (binário)				0101.0101	1010.1010

No endereço 192 está informado o início do vetor de entrada (que é 195). No endereço 193 está informado o tamanho do vetor de entrada (que é 2 bytes). Finalmente, no endereço 194 está informado o endereço de início do vetor de saída (que é 224).

Os dados do vetor de entrada, que está nos endereços 195 e 196 estão representados em decimal e em binário. Abaixo estão representados os bits dos bytes que formam o vetor de entrada.

Endereço	195 (posição [0])									196 (posição [1])							
Bit	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
Valor	0	1	0	1	0	1	0	1		1	0	1	0	1	0	1	0

Após o processamento de seu programa, o vetor de saída deverá apresentar o seguinte resultado:

Endereço	224	225	226	227	228	229	230	231
Dados (decimal)	1	2	1	2	1	2	1	2
Dados (binário)	0001	0010	0001	0010	0001	0010	0001	0010

Notar que, na tabela, estão representados apenas os 4 bits menos significativos, na representação dos dados binários, uma vez que os 4 bits mais significativos são, sempre, "0000".

5. Correção dos Trabalhos

Os arquivos fonte do RAMSES entregues serão montados usando o montador DAEDALUS.

Para a correção, o programa será carregado uma única vez. Após essa carga, será aplicado o caso de teste. Ao final da execução, será verificada a correção do resultado.

Depois de verificar o resultado, a área de memória entre 192 e 255 será alterada, o PC será colocado em "0" (zero), e seu programa será executado novamente. **Observe que a área entre 00₁₆ e BF₁₆, onde estão seu programa e dados, permanecerá com as mesmas informações do final da última execução. Ou seja, seu programa não será recarregado.**

Esse procedimento será repetido até que seu programa tenha sido submetido a 20 (vinte) casos de teste diferentes.

A nota final do trabalho será proporcional ao número de casos de teste em que o programa produzir a resposta correta.

6. O que deve ser entregue?

Deverá ser entregue somente o arquivo fonte (arquivos .RAD) escrito na linguagem simbólica do RAMSES, com a solução do problema apresentado, no Moodle da disciplina.

O programa fonte deverá conter comentários descritivos da implementação. Por exemplo, nos comentários podem ser usados comandos da linguagem "C".

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. Inicialmente, **ambos os trabalhos: original e cópias, 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.