

Atividade de avaliação 3 - Simulador Ahmes e Montador Daedalus

Um dos processos realizados para aumentar a resolução de uma imagem digital (ou seja, o tamanho da imagem em número de pixels) é o filtro de interpolação, que gera os pixels faltantes da imagem a partir dos valores dos pixels existentes. Um processo possível baseado nos mais modernos filtros de interpolação existentes é mostrado na equação abaixo, onde P é um pixel faltante, a ser gerado pelo filtro de interpolação, a partir dos valores dos pixels existentes a , b , c e d . A operação $\gg 4$ é um deslocamento à direita em 4 bits, ou seja, quatro deslocamentos à direita de 1 bit.

$$P = (2a + 6b + 6c + 2d) \gg 4$$

Como os coeficientes $\{2, 6, 6, 2\}$ são constantes, a multiplicação deles pelos valores dos pixels pode ser realizada de forma mais simplificada através de somas e deslocamentos à esquerda. A equação pode ser modificada para esta abaixo, onde P é um pixel faltante, a ser gerado pelo filtro de interpolação, a , b , c e d são os pixels existentes, as operações $\ll 1$ e $\ll 2$ são deslocamentos à esquerda de 1 e 2 bits, respectivamente, e a operação $\gg 4$ é um deslocamento à direita de 4 bits.

$$P = (a \ll 1 + b \ll 2 + b \ll 1 + c \ll 2 + c \ll 1 + d \ll 1) \gg 4$$

O programa a ser desenvolvido receberá os valores a , b , c e d , cada um deles com 8 bits em representação inteiro positivo (valores entre 0 e 255) e produz o resultado do pixel P , também em inteiro positivo (valores entre 0 e 255). Devem ser **obrigatoriamente** utilizadas as seguintes posições de memória (mostradas em decimal):

- Posição 200 – entrada a (inteiro positivo entre 0 e 255 em decimal)
- Posição 201 – entrada b (inteiro positivo entre 0 e 255 em decimal)
- Posição 202 – entrada c (inteiro positivo entre 0 e 255 em decimal)
- Posição 203 – entrada d (inteiro positivo entre 0 e 255 em decimal)
- Posição 204 – saída P (inteiro positivo entre 0 e 255 em decimal)

Os trabalhos serão corrigidos de forma automática, com **10** casos de teste 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 para os operandos e para o resultado devem ser exatamente os especificados acima;
- usar para variáveis adicionais, constantes, ou para código extra os endereços de memória 205 em diante;
- no cálculo, os valores de entrada a , b , c e d (endereços 200, 201, 202, 203) não devem ser modificados. Ou seja, se necessário, deve-se copiar os valores de entrada para variáveis de trabalho (no endereço 205 ou superior) e codificar o algoritmo usando estas variáveis de trabalho;
- variáveis alteradas durante o programa devem ser inicializadas pelo próprio programa. Sempre que necessário, utilizar posições de memória não alteradas (constantes) para realizar a inicialização.

O trabalho deverá ser entregue através do sistema Moodle, na área de “Entrega da Atividade 3”, na forma de um arquivo compactado (formato Zip) contendo:

- um arquivo de memória do Ahmes (`.mem`), com o código de máquina do programa.

- um arquivo texto (*.ahd gerado no Daedalus), com comentários contendo uma breve descrição do método utilizado. Não se esqueça de incluir seu nome completo e seu número de cartão nas primeiras linhas deste arquivo (como um comentário, o arquivo não deve gerar erros de montagem).
- Para dar nomes aos arquivos, utilize o seu nome completo, sem espaços e sem acentos, seguido do seu número de cartão, sem zeros à esquerda. Por exemplo: João da Silva, cartão 00123456 utilizará JoaodaSilva123456.mem, JoaodaSilva123456.txt e JoaodaSilva123456.zip.
A entrega de arquivos cujos nomes não obedecem a esta regra implicará em um desconto de 5% na nota do trabalho.

IMPORTANTE: Este é um trabalho **individual**. Trabalhos copiados terão a sua nota dividida entre os envolvidos.

Data de Entrega: até 11/05/2022 às 23h59, via Moodle. Não haverá prorrogação deste prazo.

Alguns casos de teste:

Teste	a (end. 200)	b (end. 201)	c (end. 202)	d (end. 203)	P (end. 204)
1	1	1	1	1	1
2	255	255	255	255	255
3	100	200	201	202	188
4	10	11	12	13	11
5	1	255	127	64	151