

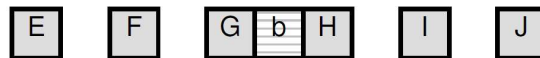
Sistemas Digitais Avançados - 2018/2
Prof. Bruno Zatt

Nome: _____

Projete, implemente e valide uma solução arquitetural para o algoritmo e requisitos descritos abaixo:

Filtro de *upsampling* (aumento de resolução) de ordem 2 para imagens monocromáticas de 8 bits por amostra. Por simplicidade, o *upsampling* será feito apenas na dimensão horizontal utilizando um filtro FIR de seis coeficientes (1,-5,20,20,-5,1), conforme equação abaixo.

$$b = (E - 5 \cdot F + 20 \cdot G + 20 \cdot H - 5 \cdot I + J) / 32$$



A entrada de dados deve ser feita através de um buffer circular implementado utilizando uma memória com 1024 palavras e 8 bits de largura (conforme esboço abaixo). Não preocupe-se com a escrita de dados na memória de entrada. Assuma que a entrada de dados é feita em formato *raster* (varrendo linha a linha da esquerda para a direita).

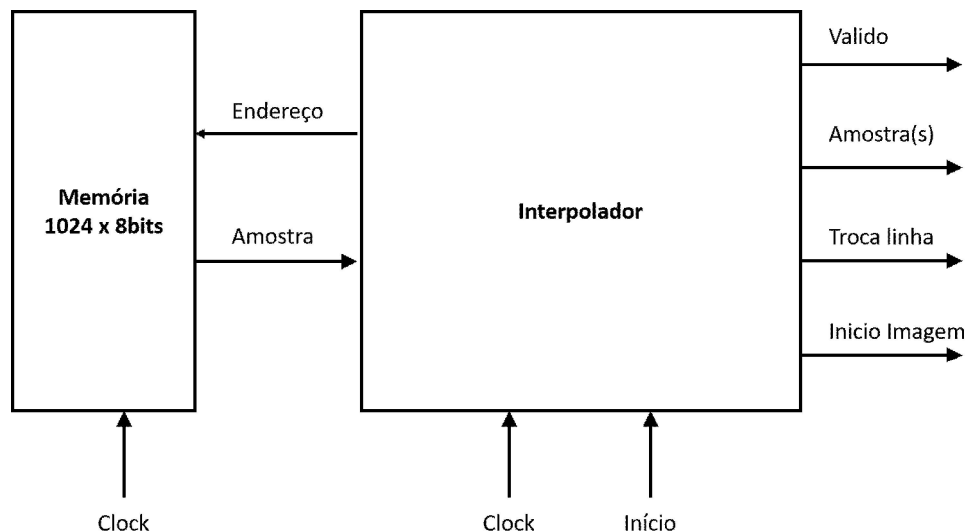
A saída da arquitetura deverá entregar as amostras novas/interpoladas (paralelismo a ser definido pelo grupo), um sinal de validade (para indicar se a saída é válida), uma flag de troca de linha e uma flag de início/troca de imagem.

As amostras de saída (interpoladas) deverão ter 8 bits (sem sinal). Portanto, caso ocorram valores inferiores a zero, a saída deve ser zero. Caso ocorram valores maiores que 255, a saída deve ser 255.

Implementação e validação deverão suportar, ao menos, imagens de 512x512 amostras.

O desempenho alvo é de 100 milhões de amostras filtradas por segundo.

Decisões como organização da arquitetura, paralelismo, barreiras de pipeline, operadores utilizados são liberdades do grupo e fazem parte da avaliação da solução.



Validação:

A validação deverá ser feita por meio de entradas e saídas utilizando arquivos de texto, modelsim, testbench e automação do processo de validação.

Como caso de teste, é fornecida uma imagem (acompanhada de um arquivo texto com suas amostras em valores binários) para validação. Por simplicidade, assuma a memória já inicializada com duas linhas da imagem.

O que entregar:

Desenho detalhado da arquitetura: deixe claro a separação entre parte operativa e parte de controle e o detalhamento dos blocos funcionais; represente também os sinais que fazem a comunicação entre os blocos funcionais.

VHDLs: todos arquivos da descrição do hardware desenvolvido.

Testbench: arquivo VHDL contendo o testbench.

script.do: arquivo que automatize o processo de validação no modelsim (compila, monta o projeto, implementa forma de onda, simula).

wave.do: forma de ondas customizada.

Resultados: área (LUTs e registradores), frequência de operação, vazão (amostras por segundo).

Relatório detalhado: após a prova, o grupo terá até segunda-feira (3/12) às 15h para compilar um relatório detalhado sobre a solução desenvolvida.

Forma de avaliação:

Dentre as soluções consideradas satisfatórias, a avaliação será feita por ranqueamento considerando notas variando entre 7 e 10. Ou seja, as soluções mais completas e eficientes receberão notas maiores. As soluções consideradas insatisfatórias terão atribuídas notas inferiores a 7.

Dica para o desenvolvimento:

O projeto da arquitetura é, certamente, a etapa mais importante. Sendo assim, recomenda-se que o grupo faça esta etapa de forma conjunta e com bastante cuidado. Para as etapas de desenvolvimento, recomenda-se que o desenvolvimento de módulos de HW/testbench/script sejam distribuídos entre os diferentes membros do grupo.

Regras:

É permitida consulta em materiais disponíveis no AVA, busca online para esclarecimento de dúvidas sobre VHDL e soluções desenvolvidas previamente pelos integrantes do grupo.

É expressamente proibida, no entanto, utilização de códigos prontos (independente da origem; exceto o reg32 disponível no AVA) e a troca de códigos entre grupos diferentes. O não cumprimento destas regras representa exclusão da prova para todo o grupo.

A tarefa poderá ser realizada em trios, respeitando a disponibilidade de máquina do laboratório.