

Laboratório 5

- CPU MIPS Pipeline –

Objetivos:

- Treinar o aluno com a linguagem de descrição de hardware Verilog;
- Familiarizar o aluno com a plataforma de desenvolvimento FPGA DE2 da Altera e o software QUARTUS II;
- Desenvolver a capacidade de análise e síntese de sistemas digitais usando uma Linguagem de Descrição de Hardware;
- Estudar e implementar uma CPU MIPS com ISA mínima e Pipeline de 5 estágios;

- 1) (0.0) Abra e compile o projeto do processador MIPS PUM v.4.9 com o Processador Pipeline
 - a. Carregue o programa testePIPE.s ;
 - b. Simule no Mars e por forma de onda comparando os resultados;
 - c. Identifique e corrija o que está causando o resultado errôneo.
- 2) (1.0) Analise o processador MIPS Pipeline fornecido. Desenhe o Diagrama de Blocos do Caminho de Dados completo incluindo os registradores de pipeline e especifique a tabela verdade dos sinais de controle por estágio do pipeline;
- 3) (1.0) Analise as unidades de Hazard e Forward, e com base na ISA especifique, através de exemplos, quais riscos de dados e de controle são detectados e tratados.
- 4) (1.0) Use o seu programa teste.s e verifique o correto funcionamento de TODAS as instruções da ISA implementada, teste usando simulação por forma de onda e pela implementação na DE2.
- 5) (1.0) Execute no processador em FPGA o seu programa de desenho das bandeiras desenvolvido no Laboratório 1. Faça vídeos demonstrativos e disponibilize no YouTube com links no relatório. Obs.: mul \$t0,\$t1,\$t2 não está implementada!
- 6) (2.0) Inclua na fp_ula.v uma entrada de iFPStart que indique para a FPULA o início de uma operação de ponto flutuante e uma saída oFPBusy que indique 1 enquanto quando a saída da FPULA não estiver pronta.
Modifique o Caminho de Dados e o Bloco de Controle do MIPS Pipeline para que ocorra *stalls* (bolhas) enquanto a FPULA estiver calculando.
- 7) (2.0) Modifique o processador de forma que a síntese ou não do Co-processador 1 (FPU) seja definida por apenas um parâmetro no início do arquivo TopDE.v ``define FPU`, maximizando a diferença entre os requisitos físicos (número de Elementos Lógico, número de Registradores, memória RAM e máxima frequência de clock) de ambas implementações. Quais foram os requisitos físicos obtidos com a FPU e sem a FPU? Compare com os resultados obtidos para o Unicidade e para o Multiciclo.
- 8) (2.0) Implemente as instruções abaixo em conformidade com a ISA MIPS (livro See MIPS Run e Manual do MIPS):

ceil.w.s \$f1,\$f2	# ceiling single precision to word	\$f1=(int) ceil(\$f2)
floor.w.s \$f1,\$f2	# floor single precision to word	\$f1=(int) floor(\$f2)
round.w.s \$f1,\$f2	# round single precision to word	\$f1=(int) round(\$f2)

 - a. (1.0) Indique as modificações necessárias no caminho de dados
 - b. (1.0) Indique as modificações necessárias no bloco de controle
 - c. (1.0) Crie um programa teste que comprove o correto funcionamento das novas instruções. Faça a simulação em forma de onda e sintetize na DE2.

Obs.: Cuidar que a instrução cvt.w.s do Mars (floor) difere da instrução cvt.w.s implementada no FPGA (round para o par)!

Dicas: Se x não for um inteiro ($x - \text{round}(x) \neq 0$), então $\text{floor} = \text{round}(x - 0.5)$ $\text{ceil} = \text{round}(x + 0.5)$