



Laboratório 4

- CPU MIPS MULTICICLO –

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;
- Apresentar ao aluno a implementação de uma CPU MIPS Multiciclo;

- 1) (0.0) Abra e compile o projeto do processador MIPS PUM v.4.8 com o Processador Multiciclo
 - a. Carregue o programa testeWAVEFORM.s ;
 - b. Faça e analise a simulação por forma de onda;
- 2) (2.0) Analise o processador Multiciclo desenhando o diagrama de blocos do Caminho de Dados usando a estrutura base vista em aula e a máquina de estados do Bloco Controlador.
- 3) (1.0) Usando seu programa teste.s, verifique o correto funcionamento de TODAS as instruções da ISA implementada, teste usando simulação por forma de onda e pela implementação na DE2 (filme a execução de algumas).
- 4) (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!
- 5) (3.0) 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 Uniciclo.
- 6) (3.0) 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 floor=round(x-0.5) ceil=round(x+0.5)