



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.5.3 com o Processador Multiciclo
 - a. Carregue o programa testeWAVEFORM2.s ;
 - b. Faça a análise do resultado da simulação por forma de onda gerada pelo DE2.vwf;
- 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).
- 4) (1.0) Execute no processador em FPGA o seu programa de simulação de lançamento de bola de canhão desenvolvido no Laboratório 1 (Dica: defina os parâmetros no seu programa, sem usar syscall 6). Grave vídeos demonstrativos e disponibilize no YouTube com links no relatório.
- 5) (3.0) Verifique o correto funcionamento do Syscall 49 (leitura do cartão SD). Defina no PC e grave em um cartão SD os 12 cenários do jogo Street Fighter. Faça um programa que leia sequencialmente as telas e as apresente no monitor VGA. Faça comentários sobre as limitações e máxima taxa de quadros atingida. Filme o experimento com links no relatório.
- 6) (3.0) Implemente as instruções abaixo em conformidade com a ISA MIPS (livro See MIPS Run e Manual do MIPS):
 - mul \$t1,\$t2,\$t3 Multiplication without overflow : Set HI to high-order 32 bits, LO and \$t1 to low-order 32 bits of the product of \$t2 and \$t3
 - jalr \$t1 Jump and link register : Set \$ra to Program Counter (return address) then jump to statement whose address is in \$t1
 - jalr \$t1,\$t2 Jump and link register : Set \$t1 to Program Counter (return address) then jump to statement whose address is in \$t2
 - 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.