



Laboratório 3

- CPU MIPS UNICICLO –

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;

PARTE A: Apresentação do ambiente de desenvolvimento e interface com o processador

- 1) (0.0) Abra e compile o projeto do processador MIPS PUM v.5.0 e carregue no kit DE-70
 - a. Descreva e execute o programa default carregado;
 - b. Execute o programa focafofa.s disponível no diretório Docs;
- 2) (0.0) Teste o processador e suas interfaces com os arquivos de testes fornecidos (testeSIMPLES, testeVGA, testeTECLADO, testeLCD, testeRS232, testeSRAM, testeSINTETIZADOR, testeReadSD, testeIRDA);
- 3) (0.0) Dado o programa testeSYSCALLv4.s, inclua o Exception Handler SYSTEMv54.s no Mars.
 - a. Execute no Mars usando a interface normal (\$s7=0)
 - b. Execute no Mars usando o BitMap Display Tool; (\$s7=100)
 - c. Execute no processador MIPS da DE2-70; (\$s7=0)

Obs.: Alguns syscalls não estão implementados!

- 4) (1.0) Faça um diagrama de fluxo genérico da rotina de tratamento de exceção presente no SYSTEMv54.s
- 5) (2.0) 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.

PARTE B: Processador Uniciclo e FPU

- 6) (0.0) Abra e compile o projeto do processador MIPS PUM v.5.1
 - a. Carregue os programas testeWAVEFORM2.s e focafofa.s;
 - b. Faça e analise a simulação por forma de onda usando o TopDE2r_versão3.vwf ;
- 7) (2.0) Analise o processador MIPS PUMv.5.1 UNICICLO desenhando o diagrama de blocos do Caminho de Dados usando a estrutura base vista em aula e a tabela verdade por instrução do Bloco Controlador.
- 8) (2.0) Crie um programa teste.s que verifique todas as instruções da ISA implementada. Encontre qual a máxima frequência na qual o programa é executado corretamente indicando a instrução mais demorada. Comprove o teste usando simulação por forma de onda e pela implementação na DE2 (filme a execução de algumas).
- 9) (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.