

Laboratório 5

- CPU MIPS Pipeline –

GRUPO 6

Dayanne Fernandes da Cunha, 13/0107191

Lucas Mafra Chagas, 12/0126443

Marcelo Giordano Martins Costa de Oliveira, 12/0037301

Lucas Junior Ribas, 16/0052289

Caio Nunes de Alencar Osório, 16/0115132

Diego Vaz Fernandes, 16/0117925

¹Dep. Ciência da Computação – Universidade de Brasília (UnB)
CiC 116394 - OAC - Turma A

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 Pipeline*.

Ferramentas

Todos os códigos escritos neste laboratório podem ser encontrados no repositório <https://github.com/Dayof/OAC172> do *GitHub*.

- FPGA DE2 da Altera
- QUARTUS-II
- Verilog HDL

Exercício 2. Análise do processador Pipeline

Diagrama de Blocos do Caminho de Dados

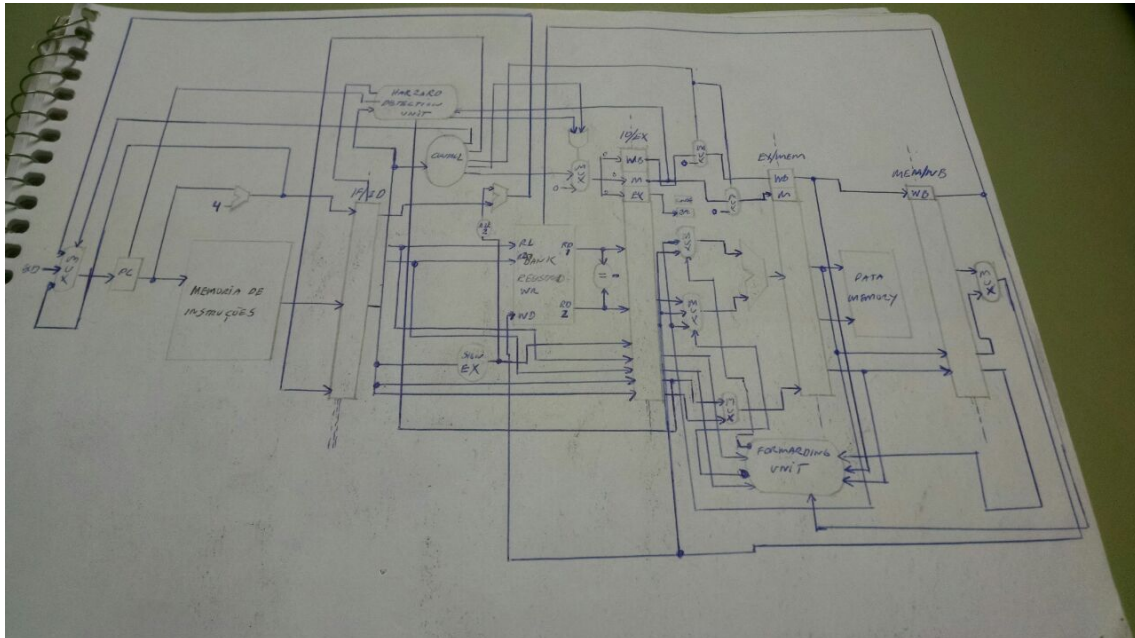


Figure 1. Diagrama de Blocos do Caminho de Dados.

Tabela Verdade dos Sinais de Controle

Exercício 3. Análise unidades de Hazard e Forward

A unidade de Forwarding detecta e trata os hazards que podem acontecer com instruções nas etapas EX e WB, mais especificamente em operações que tentam ler registradores que estão sendo modificados por instruções em etapas posteriores do processador.

add \$t0, \$t1, \$t2 add \$t3, \$t0, \$t0

é um exemplo e

add \$t0, \$t1, \$t2 sub \$s0, \$s1, \$s2 add \$s0, \$t0, \$t0

também.

A unidade de Hazard implementada trata os casos:

- Desvio condicional (beq, bne), que dão flush na etapa anterior para que aconteça o desvio.

bne \$t0, \$t1, LABEL add \$t0, \$t1, \$t2

- jr, quando uma instrução em EX tiver o registrador destino igual ao Rs da instrução em ID ou a instrução em Mem for escrever em \$ra.

addi \$ra, \$ra, 0x0AC jr \$ra

- lw, caso padrão onde alguma instrução precisa do valor que ainda não foi modificado na memória.

lw \$t0, 0 (\$t1) sub \$t2, \$t1, \$t1

Esses casos foram observados na ISA implementada.

Exercício 4. Teste do funcionamento das instruções da ISA

A demonstração e explicação do código estão presentes nos seguintes vídeos:

- Formas de Onda
- Implementação na DE2

Exercício 5. Software de lançamento de bola de canhão na *FPGA*

Segue a simulação da bola de canhão:

Bola de canhão

Exercício 6. Implementação do Cartão SD

As limitações observadas ao executar os cenários no cartão sd e impressas no monitor através de um cabo vga foram:

-
-

Segue o vídeo dos cenários:

Cenários no cartão SD

Exercício 7. Novas instruções usando a *ISA MIPS*

Parêmtros

Caminho de dados

Bloco de controle

Teste das novas instruções

References