



## **Manual de Uso do MIPS com Pipeline na FPGA**

### **Design de Computadores: Projeto 2 - Entrega Final**

Gabriel Hermida e Pedro Civita

**Conceito Desejado:** A+, Entregou o projeto, na organização *pipeline*, com todas instruções (grupos A e B) funcionando sem erros.

São Paulo

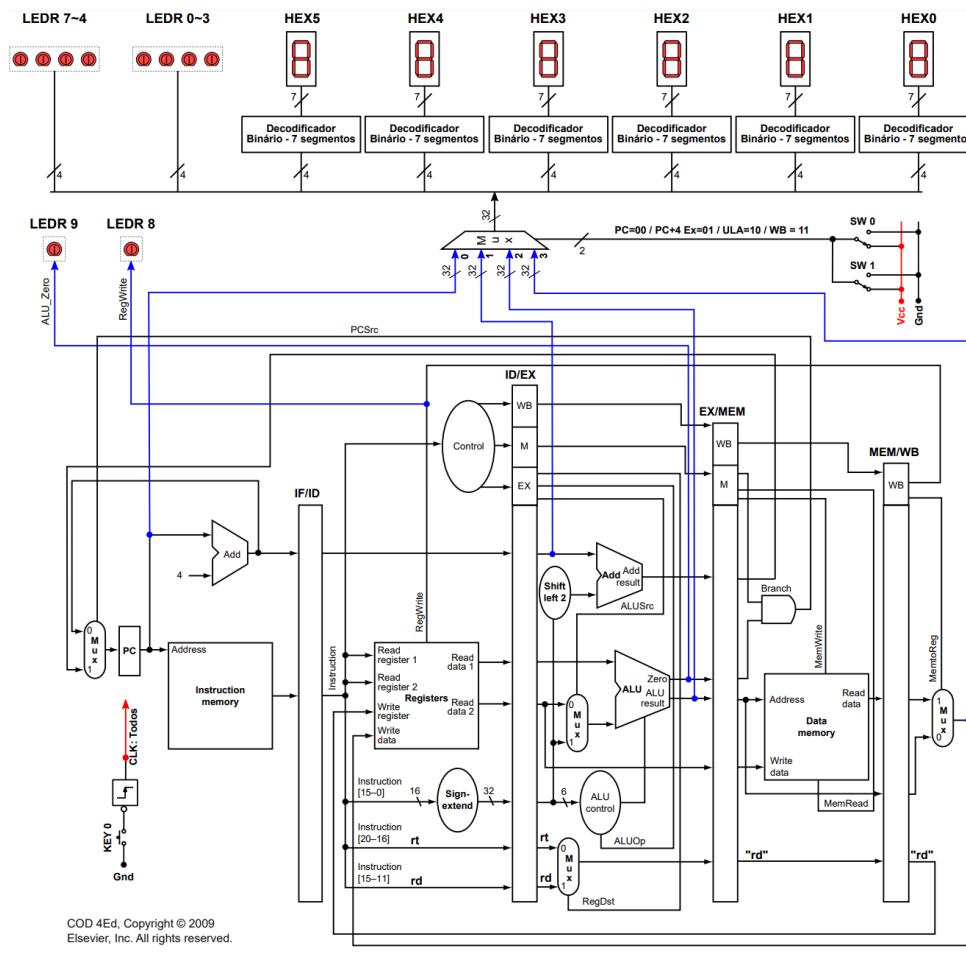
Junho de 2024

## Introdução

Este manual descreve como utilizar o projeto do processador MIPS desenvolvido para o Projeto 2 da disciplina "Design de Computadores" do Insper. O manual aborda as principais funcionalidades, os controles disponíveis na FPGA e como carregar e utilizar o arquivo de memória (ROMcontent.mif).

## Introdução

A imagem a seguir ilustra o fluxo de dados completo do pipeline com a unidade de controle do processador MIPS. Esta imagem ajuda a entender como as instruções são processadas através dos diferentes estágios do pipeline.



Pipeline com a Unidade de Controle.

## Controles na FPGA

KEY0:

- Ao pressionar a KEY0, você avança um ciclo de clock no processador MIPS.

Switch 0 (SW0) e Switch 1 (SW1):

- Os switches SW0 e SW1 controlam qual valor será exibido nos displays de 7 segmentos (HEX0 - HEX5).
  - SW0 e SW1 na posição baixa (para baixo): Os displays mostram o valor do PC no estágio IF.
  - SW0 na posição baixa e SW1 na posição alta: Os displays mostram o valor do PC+4 no estágio ID.
  - SW0 na posição alta e SW1 na posição baixa: Os displays mostram a saída da ULA no estágio EX.
  - SW0 e SW1 na posição alta: Os displays mostram o valor do WB.

## Instruções de Uso

- Configuração Inicial:
  - Certifique-se de que a FPGA está corretamente configurada e que o arquivo de memória (ROMcontent.mif) foi carregado.
- Carregar o Arquivo de Memória:
  - O arquivo ROMcontent.mif deve ser utilizado para inicializar a memória ROM do processador MIPS.
  - Este arquivo contém o conjunto de instruções que serão executadas pelo processador.
- Operação:
  - Utilize a KEY0 para avançar o ciclo de clock e observar o funcionamento do processador.
  - Utilize o SW0 e SW1, em combinação, para verificar a atualização dos displays de 7 segmentos, como mostrado acima. Abaixo, está um breve mapa das seleções:
    - 00: Exibe pcOut (IF)

- 01: Exibe pc4Out (ID)
- 10: Exibe saidaULA (EX)
- 11: Exibe mOutDadoEsc (WB)

## **Arquivo ROMcontent.mif**

-- Copyright (C) 2017 Intel Corporation. All rights reserved.

-- Your use of Intel Corporation's design tools, logic functions

-- and other software and tools, and its AMPP partner logic

-- functions, and any output files from any of the foregoing

-- (including device programming or simulation files), and any

-- associated documentation or information are expressly subject

-- to the terms and conditions of the Intel Program License

-- Subscription Agreement, the Intel Quartus Prime License Agreement,

-- the Intel FPGA IP License Agreement, or other applicable license

-- agreement, including, without limitation, that your use is for

-- the sole purpose of programming logic devices manufactured by

-- Intel and sold by Intel or its authorized distributors. Please

-- refer to the applicable agreement for further details.

WIDTH=32;

DEPTH=64;

ADDRESS\_RADIX=DEC;

DATA \_RADIX=HEX;

## CONTENT BEGIN

```
0 : 3c090000;    --lui $t1, 0x0000;
1 : 3c0baaaa;    --lui $t3, 0xAAAAA;
2 : 3c0f1000;    --lui $t7, 0x1000;
3 : 3529000a;    --ori $t1, $t1, 0x0A;      # $t1 (#9) := 0x0A
4 : 356baaaa;    --ori $t3, $t3, 0xAAAAA;     # $t3 (#11) := 0xAAAAAAAAAA
5 : 35ef0000;    --ori $t7, $t7, 0x0000;
6 : 212a0001;    --addi $t2, $t1, 0x01;      # $t2 (#10) := 0x0B
7 : 01606025;    --or $t4, $t3, $0;        # $t4 (#12) := 0xAAAAAAA
8 : 316dffff;    --andi $t5, $t3, 0xFFFF;    # $t5 (#13) := 0x0000AAAA
9 : 01497022;    --sub $t6, $t2, $t1;      # $t6 (#14) := 0x01
10 : ac090008;   --sw $t1, 8($zero);      # M[8] = 0x0A
11 : 8c080008;   --lw $t0, 8($zero);
12 : 010a7824;   --and $t7, $t0, $t2;      # Hazard Load Use
13 : 290fffff;   --slti $t7, $t0, 0xFFFF;
14 : 012a402a;   --slt $t0, $t1, $t2;
```

--destinoBEQ:

15 : 012e4820; --add \$t1, \$t1, \$t6; # t0 = t2, segunda vez: t0 != t2  
16 : 00000000; --nop;  
17 : 00000000; --nop;  
18 : 112afffc; --beq \$t1, \$t2, destinoBEQ; # Desvia na primeira e nao desvia depois  
19 : 00000000; --nop;  
20 : 00000000; --nop;  
21 : 00000000; --nop;  
22 : 0c000020; --jal subrotina;  
23 : 00000000; --nop;  
24 : 00000000; --nop;  
25 : 00000000; --nop;  
26 : 00000000; --nop;  
27 : 150d0008; --bne \$t0, \$t5, fim  
28 : 00000000; --nop;  
29 : 00000000; --nop;  
30 : 00000000; --nop;  
31 : 00000000; --nop;

--subrotina:

32 : 00000000; --nop;

33 : 03e00008; --jr \$ra;

34 : 00000000; --nop;

35 : 00000000; --nop;

--fim:

36 : 00000000; --nop;

37 : 08000024; --j fim;

38 : 00000000; --nop;

39 : 00000000; --nop;

[40..63] : 00000000; --nop;

END;

## **Conclusão**

Este manual fornece uma visão geral básica de como utilizar o processador MIPS com pipeline desenvolvido para a entrega final do Projeto 2. As principais funcionalidades estão relacionadas ao controle do clock e à exibição de informações nos displays de 7 segmentos.