



## Projeto: Implementação do MIPS pipeline

### Orientações gerais:

- Atividade deve ser realizada em grupo composto por até 3 alunos.
- Entrega através da ferramenta de Atividades do Ae até **01/dez/2019 às 23:55 h** (a existência de prazo maior na ferramenta por tolerância ou ressubmissão não indica extensão de prazo).
- O que entregar:
  - relatório respondendo as atividades propostas. Insira figuras, diagramas, capturas de tela, no relatório, que deve ser entregue em formato pdf. Seja preciso e justifique as suas respostas. Imagens ilegíveis (muito pequenas ou borradas, por exemplo) não serão consideradas na correção.
  - Arquivo com a implementação VHDL do MIPS pipeline, que será executada no ghdl (um único arquivo contendo FD e UC, e o que mais for necessário).
  - Tanto o relatório como o código deve conter o número USP e nome dos membros do grupo na mesma ordem e grafia.

Objetivo: entender e implementar o MIPS (conjunto de instruções reduzido) pipeline em VHDL.

### Material de apoio:

- *Projeto Digital e Arquitetura de Computadores*. David Money Harris & Sarah L. Harris. O assunto de interesse é o capítulo 7 - Microarquitetura, mais especificamente a seção 7,6 que trata da representação do MIPS em HDL.
- Implementação em VHDL do MIPS de ciclo único (arquivo mipssingle.vhd corrigido). O arquivo inclui testbench, que já está no padrão que será usado na correção automática.
- Para o ghdl, utilize os seguintes comandos para o arquivo mipssingle.vhd:
  - `$ ghdl -a --std=08 --ieee=synopsys mipssingle.vhd`
  - `$ ghdl -e --std=08 --ieee=synopsys testbench`
  - `$ ghdl -r --std=08 --ieee=synopsys testbench`
- Dois programas de referência para testar o funcionamento correto das instruções são: `mipstest.asm` (código fonte no livro e arquivo de memória correspondente no Ae) e `test2.asm` abaixo.

```
# test2.asm
# 23 March 2006 S. Harris sharris@hmc.edu
# Test MIPS instructions.

#Assembly Code
main:      ori  $t0, $0, 0x8000
           addi $t1, $0, -32768
           ori  $t2, $t0, 0x8001
           beq  $t0, $t1, there
           slt  $t3, $t1, $t0
           bne  $t3, $0, here
           j    there
here:      sub  $t2, $t2, $t0
           ori  $t0, $t0, 0xFF
there:     add  $t3, $t3, $t2
           sub  $t0, $t2, $t0
           sw   $t0, 82($t3)
```
- *Convert the program to machine language and put it in a file named memfile2.dat. You may choose to use the MPLAB assembler to check your work. Modify imem to load this file. Modify the testbench to check for the appropriate address and data value indicating that the simulation succeeded. Run the program and check your results. Debug if necessary. When you are done, print out the waveforms as before and indicate the*



*address and data value written by the sw instruction.*

### Conjunto de instruções do MIPS

O conjunto de instruções suportado pelo processador MIPS a ser implementado deve incluir as seguintes instruções:

- referência a memória: lw, sw
- Lógico-aritmética: add, addi, sub, and, or, ori, slt
- transferência de controle: beq, bne, j

### Atividades

1. Explique como é implementada a versão pipeline do MIPS.  
Indique claramente se modificou (e como) a estrutura do MIPS de ciclo único (arquivo de referência mipssingle.vhd), ou se fará uma implementação independente (neste caso, explique a sua estrutura).
2. Implemente em VHDL o MIPS pipeline com 5 estágios, conforme o livro-texto, que deve suportar o conjunto de instruções indicado. Note que é necessário implementar encaminhamento, detecção de conflitos e inserção de bolhas (stall, ou nop) para permitir a execução correta das instruções.  
O relatório deve conter diagramas e explicações associadas a implementação.
3. Simule a execução do MIPS pipeline. O arquivo memfile.dat já possui o equivalente ao programa mipstest.asm.  
Se tudo correr bem na simulação, a testbench imprimirá “(report failure): NO ERRORS: Simulation succeeded.”  
Execute o programa test2.asm e outros programas de teste que julgar adequados.  
Apresente e discuta os resultados no seu relatório. Se julgar pertinente, pode incluir figuras com formas de onda (desde que legíveis).
4. Discuta as limitações da sua implementação, indicando como estas poderiam ser resolvidas.