

## Projeto Final

O projeto final da disciplina consiste em uma parte a ser desenvolvida em Assembly e uma parte a ser desenvolvida em VHDL. O projeto deve ser executado **preferencialmente** em trios, mas serão aceitas duplas. **Não serão** aceitas entregas individuais. Os grupos deverão escolher **UMA** entre duas opções diferentes de projetos (os detalhes dos projetos estão descritos mais à frente neste documento):

- **Opção 1:** Assembly no SPIM ou MARS + Processador simples em VHDL.
- **Opção 2:** Processador MIPS em VHDL + Fibonacci.

Os grupos que escolherem a **opção 1** deverão entregar, juntamente com os códigos desenvolvidos, um relatório em PDF conforme o formato especificado no detalhamento de cada parte do projeto abaixo. Os grupos que escolherem a **opção 2** estarão **dispensados** de entregar o relatório em PDF e **deverão apresentar** para o professor, até a aula da entrega final dos projetos, o código em VHDL do processador MIPS e a simulação mostrando a computação dos números de Fibonacci.

A opção escolhida deve ser enviada por e-mail para o professor **até dia 23 de outubro de 2023**, juntamente com os nomes dos integrantes do grupo, e **não poderá** ser alterada posteriormente. O prazo final de entrega dos projetos é dia **4 de dezembro de 2023**.

## Opção 1: Assembly no SPIM ou MARS + Processador simples em VHDL

### Parte I – Assembly

Deve ser desenvolvido um programa em linguagem assembly do MIPS para registrar as notas das atividades práticas de Arquitetura de Computadores. O programa desenvolvido deve ser capaz de:

- 1) Cadastrar novos alunos: receber do usuário o nome de um aluno e armazenar. O sistema deve poder guardar os nomes de pelo menos 8 alunos. Você pode limitar o tamanho dos nomes dos alunos em 8 caracteres.
- 2) Cadastrar notas: armazenar uma nota para cada um dos 6 laboratórios e 1 projeto para cada um dos alunos armazenados.
- 3) Alterar nota: modificar ou excluir notas de laboratórios ou projeto para um dos alunos armazenados.
- 4) Exibir notas: mostrar notas e médias finais para todos os alunos, ordenados em ordem alfabética, juntamente com a média aritmética das notas da turma inteira.

Utilize os valores da nota de cada laboratório e projeto conforme o atual oferecimento da disciplina. Você pode tomar decisões de projeto sobre como implementar cada funcionalidade, e essas deverão ser explicitadas e explicadas no relatório – ver formato de relatório ao final.

### Parte II – VHDL

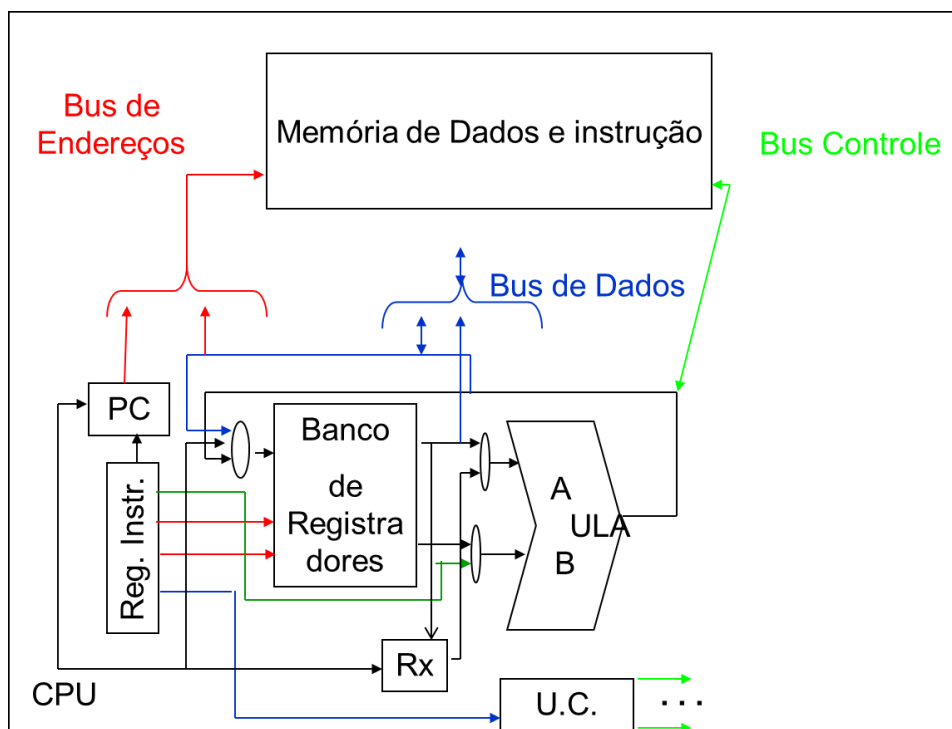
Descrever em VHDL uma CPU capaz de executar as seguintes instruções:

Instrução	Significado	Descrição
MOV Ri,Rj	$R_i \leftarrow R_j$	Move
MOV Ri,Imed	$R_i \leftarrow \text{Imed}$	Move Immediate
ADD Ri,Rj,Rk	$R_i \leftarrow R_j + R_k$	Add
ADDI Ri,Rj,Imed	$R_i \leftarrow R_j + \text{Imed}$	Add Immediate
SUB Ri,Rj,Rk	$R_i \leftarrow R_j - R_k$	Subtract

SUBI	$R_i, R_j, \text{Imed}$	$R_i \leftarrow R_j - \text{Imed}$	Subtract Immediate
AND	$R_i, R_j, R_k$	$R_i \leftarrow R_j \& R_k$	And
ANDI	$R_i, R_j, \text{Imed}$	$R_i \leftarrow R_j \& \text{Imed}$	And Immediate
OR	$R_i, R_j, R_k$	$R_i \leftarrow R_j   R_k$	Or
ORI	$R_i, R_j, \text{Imed}$	$R_i \leftarrow R_j   \text{Imed}$	Or Immediate

Onde  $R_i, R_j, R_k$  são registradores de  $n$  bits (pertencentes a um banco de registradores com pelo menos 4 registradores de 8 bits cada) e  $\text{Imed}$  é um valor imediato.

Arquitetura da CPU sugerida:



Utilizem como referência os livros:

1 – BROWN, Stephen e VRANESIC, Svonko – Fundamentals of Digital Logic with VHDL Design.

2 - PATTERSON, David A. e HENNESSY, John L. – Computer Organization and Design – The Hardware and Software Interface

## **Relatório**

Além dos códigos Assembly MIPS da parte I e VHDL da parte II, a entrega do projeto deverá estar acompanhada de um relatório, em formato PDF, contendo:

1. Introdução;
2. Especificação do projeto Assembly:
  - a. Detalhes de projeto e escolhas feitas;
  - b. Detalhes de implementação.
3. Especificação do projeto VHDL:
  - a. Banco de registradores (quantidade, tamanho da palavra, endereçamento);
  - b. Formato das instruções;
  - c. Unidade de controle (diagramas e tabelas de estados, sinais de entrada e saída e seus significados);
  - d. Diagramas do caminho de dados completo.
4. Resultados – descrições dos testes realizados, saídas do programa Assembly, formas de onda da simulação VHDL, etc;

### **Opção 2: Processador MIPS em VHDL + Fibonacci**

Utilizando as especificações do Capítulo 4 do livro texto da disciplina (Patterson & Hennessy, Organização e Projeto de Computadores – A Interface Hardware/software), desenvolva em VHDL uma implementação do processador MIPS sem Pipeline. Para teste, desenvolva em Assembly MIPS e insira na memória de instruções um pequeno programa que preencha a memória de dados com, pelo menos, os 16 primeiros números de Fibonacci.

O código em VHDL pode ser desenvolvido e testado utilizando o Quartus + Modelsim ou qualquer outra ferramenta escolhida pelo grupo e que permita a demonstração e avaliação pelo professor até o prazo final de entrega do projeto. Os alunos deverão estar preparados para responderem perguntas sobre o código do projeto no dia da apresentação.