



PROJETO FINAL
PRAZO DE ENTREGA: 10/03/2022

ALUNO(A): Guilherme Lirioberto da Silva Alves e Glisbel de Las Nieves Aponte Lopez

ATENÇÃO: Vale ressaltar as instruções apresentadas aqui já foram apresentadas em aula, logo o foco deste documento é na descrição dos artefatos a serem entregues para o projeto final. Assim, descreva:

1. As soluções com o máximo de detalhes possível inclusive a forma como os testes foram feitos;
2. Todos os artefatos (relatório – modelo disponível no SIGAA no arquivo `modelo_relatorio_processador_AOC.docx`, código fonte de programas, e outros) gerados para este trabalho devem ser adicionados em um repositório no site `github.com`, com o seguinte formato **AOC_Nome1Nome2_UFRR_2022**;
3. O modelo de relatório do projeto está disponível no SIGAA no tópico **Apresentação do Projeto Final - Parte 1 (10/03/2022 - 10/03/2022)**;
4. O referido repositório deve ser adicionado na planilha (incluindo a URL com extensão `.git`) do projeto final no SIGAA no tópico de aula **Programação em VHDL - Arquitetura (18/01/2022 - 18/01/2022)**;
5. O projeto deve apresentar a IDE utilizada e como o compilar/executar o processador do projeto final.

[PROJETO FINAL] Projetar e implementar um processador RISC de **8 bits** (semelhante ao MIPS), segue os requisitos para a elaboração deste projeto:

1. Os componentes do processador deverão ser escritos na linguagem de programação VHDL;
2. A descrição da estrutura das instruções suportadas pelo processador deverá ser apresentada por classe e suas respectivas divisões por bits;
3. A descrição da linguagem (assembly e binário) suportadas pelo processador, como no caso do MIPS;
4. Apresentação do *datapath* (barramento com suas conexões) do processador indicando a quantidade de bits por trilhas e as entradas e saídas para cada componente;
5. Apresentação da unidade de controle e os sinais de controle para cada instrução do processador;
6. As seguintes instruções são obrigatórias para o processador: load, store, soma, subtração, beq, salto incondicional; e
7. Apresentar simulações e testes usando *waveforms* ou *testbench* para cada instrução e pelo menos um programa utilizando todas as instruções suportadas pelo processador.

