

Trabalho Prático #2

Professor: Daniel Fernandes Macedo e Omar Paranaíba Vilela Neto

Antes de começar seu trabalho, leia todas as instruções abaixo.

1. Instruções gerais

- O objetivo desse trabalho é te familiarizar com a Linguagem de Descrição de Hardware Verilog. Será disponibilizado no moodle um arquivo .ipynb com uma implementação do RISCv 5 estágios em Verilog. Sua tarefa neste trabalho será **alterar o caminho de dados** fornecidos a **fim de incluir mais operações e módulos (descritos na Seção 3)**. É necessário executar esse arquivo no **Google Colab**, sendo esta a plataforma que será utilizada para avaliar as submissões dos trabalhos.
- O trabalho pode ser feito em grupos compostos por **até 3 alunos**.
- Entregas após o prazo serão aceitas, porém haverá uma penalização. Quanto maior o atraso maior a penalização.
- Você deve entregar um único **arquivo zip**, contendo um arquivo **.ipynb** com a implementação do caminho de dados com as funções pedidas a seguir, em Verilog e um **relatório (descrito a seguir)**. Note que todas as funções devem estar no mesmo caminho de dados, ou seja, o trabalho é incremental, você deve entregar somente um caminho de dados contendo todas as funções solicitadas.
- Deverão ser implementados os arquivos Verilog e os códigos de teste em assembly das instruções. **As suas modificações devem ser feitas em uma cópia do código Verilog fornecido.**
- No mesmo arquivo zip contendo o caminho de dados, você deve **enviar um relatório**, em pdf, explicando suas **decisões de projeto e testes realizados**. Este documento deve conter nome e matrícula de todos os integrantes do grupo.
 - Sugestão de tópicos:
 - Introdução: apresente os objetivos do trabalho, contextualize o tema e indique quais arquivos/funções foram alteradas.
 - Desenvolvimento: descreva a sua implementação separando as subseções para cada um dos problemas solicitados neste trabalho.
 - Testes: descreva os testes implementados por vocês.
 - Conclusão: concluir o relatório fazendo um breve resumo do trabalho, as tomadas de decisão e os desafios durante a implementação
- **Cada grupo deve fazer somente uma submissão.**

2. Orientações para compilar os arquivos Verilog e rodar os testes

No .ipynb fornecido, você encontrará a implementação de um RiscV sem *forwarding* (adiantamento), e com uma unidade de detecção de Hazards (perigos). Para se familiarizar com sua estrutura, siga os passos abaixo:

- Na seção 'Assembly' há um exemplo de um código de testes em hexadecimal, e em seguida um código em Python que pode auxiliar a transformá-lo em hexadecimal. Todos os testes devem ser carregados em hexadecimal para a memória de instruções do processador. Teste diferentes

combinações de instruções. Existe um arquivo instruction.txt de exemplo.

- Execute as células da seção 'Instalação de Extensões'.
- A seção 'Implementação do Processador' contém as descrições em Verilog de todos os estágios do processador. Elas estão divididas por estágio. No final há uma imagem do processador para ilustração. Execute cada uma das células e se familiarize com a implementação.
- Em 'Código de execução' e 'Compilando e Executando' há um exemplo de como escrever as instruções para serem carregadas no processador (que devem estar em **im_data.txt**) e como rodar. Note que após a execução é exibido o estado do pipeline ao longo da execução do programa. Além disso, vocês podem sempre printar os bancos de registradores para conferir o estado final do processador. Esses dados ficam em reg.data e mem.data.
- Em 'Interativo Compilando e Executando' há uma versão interativa do estado do pipeline, onde você pode ver uma animação do estado do processador a cada ciclo de execução.
- Em 'Teste Load e Store' e 'Material Complementar (debug)', há mais exemplos de execução de instruções e também de como gerar uma forma de onda (waveform) para sinais específicos do Design. No relatório é desejado que se coloque waveforms mostrando os sinais de interesse para verificar se a execução das instruções implementadas está correta.
- Utilizem a seção 'Testes' para adicionar seus testes e tudo que seja relevante para explicar o processo de desenvolvimento e validação das novas instruções. **Reportem os seus testes no relatório.**

3. Descrição da implementação:

Para este trabalho prático, deve-se implementar as seguintes instruções:

1. mul Rd, Rs1, Rs2
2. div Rd, Rs1, Rs2
3. addi Rd, Rs1, Rs2
4. beq, Rs1, Rs2, label