



Trabalho 5: Projeto do Banco de Registradores do RISC-V

Universidade de Brasília
Departamento de Ciência da Computação
Organização e Arquitetura de Computadores
Evelyn Soares Pereira
17/0102785
soares.evelynp@gmail.com

1. DESCRIÇÃO DO TRABALHO

Este trabalho descreve o projeto e implementação de um banco de registradores de 32 bits, seguindo a arquitetura RISC-V, utilizando a linguagem VHDL. O banco de registradores fornece armazenamento temporário para dados e operandos durante a execução de instruções. Aqui foi definida a estrutura do banco de registradores, considerando as especificações da arquitetura RISC-V. Isso inclui a quantidade de registradores, o tamanho de cada registrador e as operações suportadas. Implementado em VHDL o código para o banco de registradores, levando em consideração os requisitos funcionais e não funcionais. Criado um ambiente de simulação utilizando um testbench em VHDL. Verificando o comportamento do banco de registradores em diferentes cenários de leitura e escrita. O projeto foi feito no EDA Playground.

Considerando a descrição do trabalho, o código em vhdl, anexado a este relatório tem a seguinte lógica: É feita a declaração da entidade (componente) chamada XREGS, que possui parâmetro genérico WSIZE (tamanho da palavra) e portas de entrada/saída (clk, wren, rs1, rs2, rd, data, ro1, ro2). Após isso, tem a declaração da arquitetura chamada "behavior" para a entidade XREGS, é definido um tipo de dado que é um array de 32 elementos, cada um sendo um vetor de bits com o tamanho especificado por WSIZE. Depois é declarado um sinal chamado regs do tipo do array. Esse sinal representa o banco de registradores.

É feita a declaração de um processo sensível à borda de subida do sinal de clock (clk) e iniciada uma estrutura condicional que verifica se ocorreu uma borda de subida no sinal de clock. Verifica se a habilitação de escrita (wren) está ativa e verifica se o endereço do registrador (rd) é diferente de zero, se as condições anteriores forem atendidas, escreve o valor presente na entrada data no registrador correspondente ao endereço rd. Atribui à saída ro1 o valor lido do registrador indicado por rs1 e atribui à saída ro2 o valor lido do registrador indicado por rs2.

2. TESTES REALIZADOS

O código do testbench realiza a verificação dos resultados do módulo XREGS usando a diretiva ASSERT. O testbench simula o comportamento de escrever em três registradores (registradores 1, 2 e 3) e lê-los posteriormente. O processo no testbench escreve em alguns registradores e, em seguida, lê esses registradores. O testbench escreve no registrador 0 (constante) e, em seguida, tenta lê-lo. A ASSERT é usada para verificar se o valor do registrador 0 permanece zero, a ASSERT compara o valor lido do registrador 0 com o valor esperado (todos os bits zero). Se a condição for falsa, uma mensagem de erro é relatada.

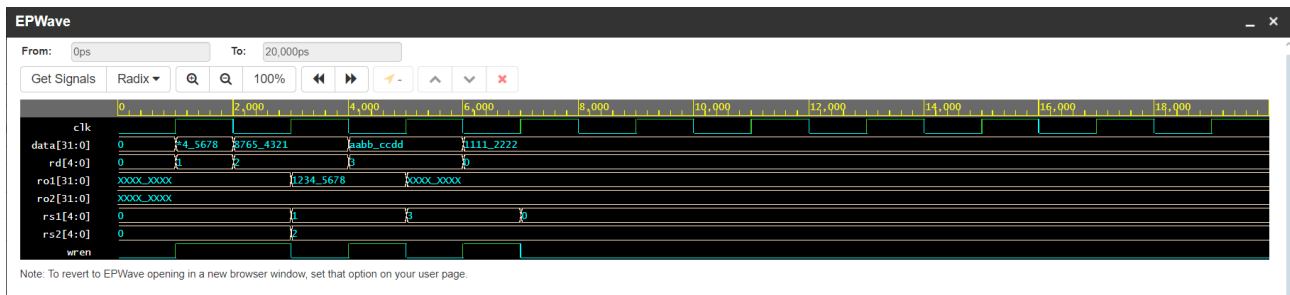


Figure 1. Simulação.

3. SIMULAÇÃO E RESULTADOS

Para controlar o sinal XREGS[0], essencialmente, é verificado se a escrita ($wren = '1'$) está ocorrendo e se o endereço do registrador (rd) é diferente de "00000". Se ambas as condições são atendidas, o valor é escrito no registrador correspondente, caso contrário, a escrita é ignorada. No trecho do testbench, é simulado uma escrita no registrador 0 ($rd \leq "00000"$) com o valor $data \leq x"11112222"$. No entanto, devido à condição $if\ rd \neq "00000"$, esta escrita é efetivamente ignorada, garantindo que XREGS[0] permaneça como uma constante com valor zero.

O testbench 'testbench.vhd' e o código VHDL 'design.vhd' estão anexados com este relatório.