Universidade de Brasília - UnB

19 de agosto de 2024

Trabalho 04 - Geração de Dados Imediatos no RISC-V

Ricardo Mendes D Abadia
202024689@aluno.unb.br
20/2024689
Turma 03

1. Explicação do código

1.1. Plataforma Usada

Para a confecção dos código foi utilizada a plataforma EDA Playground de forma a facilitar tanto a estrutura de design quanto a do teste bench ser realizado. Outra praticidade importante é a de ser possível acessar a plataforma de qualquer dispositivo.

1.2. Estrutura do Gerador de Imediatos

O desenvolvimento do código resultou no seguinte código:

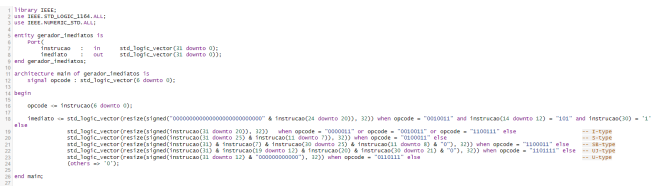

Figura 1. Código

Baseado na figura 1 a entidade possui uma entrada (instrução) e uma saída (imediato), sendo assim, foi realizado uma arquitetura afim de extrair a informação do Opcode, armazenar em um sinal, realizar a analise do sinal, e assim gerar o imediato de acordo com cada Opcode passado na instrução.

1.3. Estrutura do Teste Bench do Gerador

A estrutura do teste bench ficou da seguinte maneira:

Baseado no código desenvolvido, foi realizado a instanciação do design desenvolvido anteriormente, afim de realizar todos os testes passados. Realizando a devida simulação, conseguimos obter o seguinte resultado da figura 4, sendo assim, com as instruções passadas para simularmos, conseguimos obter os imediatos esperados.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity tb_gerador_imediatos is
end tb_gerador_imediatos;

architecture main of tb_gerador_imediatos is
    signal tb_instrucao : std_logic_vector(31 downto 0);
    signal tb_imediato : std_logic_vector(31 downto 0);

    component gerador_imediatos
        Port ( instrucao : in std_logic_vector(31 downto 0);
               imediato : out std_logic_vector(31 downto 0));
    end gerador_imediatos;

begin
    DUT: gerador_imediatos
        Port map (
            instrucao => tb_instrucao,
            imediato => tb_imediato
        );

    Teste: process
    begin
        tb_instrucao <= x"000002B3"; -- R-type
        wait for 10 ns;

        tb_instrucao <= x"01002283"; -- I-type
        wait for 10 ns;

        tb_instrucao <= x"f9c00313"; -- I-type
        wait for 10 ns;
```

Figura 2. Teste Bench Parte 01.

2. Respostas das Perguntas Solicitadas no Roteiro

2.1. Qual a razão do embaralhamento dos bits do imediato no RiscV ?

O embaralhamento dos bits em uma determinada instrução é realizada afim de identificar determinadas instruções, realizando as devidas compactações e simplificações de hardware, por exemplo, uma instrução do tipo-R não geram imediatos, já a instrução do tipo-U gera um imediato de 20 bits, sendo assim, cada instrução, a depender do Opcode, gera um imediato, afim de realizar a determinada instrução.

```
29          tb_instrucao <= x"01002283"; -- I-type
30          wait for 10 ns;
31
32          tb_instrucao <= x"f9c00313"; -- I-type
33          wait for 10 ns;
34
35          tb_instrucao <= x"fff2c293"; -- I-type
36          wait for 10 ns;
37
38          tb_instrucao <= x"16200313"; -- I-type
39          wait for 10 ns;
40
41          tb_instrucao <= x"01800067"; -- I-type
42          wait for 10 ns;
43
44          tb_instrucao <= x"40a3d313"; -- I-type*
45          wait for 10 ns;
46
47          tb_instrucao <= x"00002437"; -- U-type
48          wait for 10 ns;
49
50          tb_instrucao <= x"02542e23"; -- S-type
51          wait for 10 ns;
52
53          tb_instrucao <= x"fe5290e3"; -- SB-type
54          wait for 10 ns;
55
56          tb_instrucao <= x"00c000ef"; -- UJ-type
57          wait for 10 ns;
58
59      end process;
60  end main;
```

Figura 3. Teste Bench Parte 02.

Figura 4. Resultados.

2.2. Por que alguns imediatos não incluem o bit 0 ?

Instruções que realizam saltos não podem incluir o bit zero afim de não gerar um endereço ímpar no imediato, pois o endereço de memoria são valores resultantes pares.

2.3. Os imediatos de operações lógicas estendem o sinal ?

Não estendem o sinal pois realizam a operação sobre todos os bits dos registradores em operação, não sendo necessário realizar a extensão do sinal, sendo assim o valor do imediato é tratado como um valor sem sinal.

2.4. Como é implementada a instrução NOT no RiscV ?

É implementa utilizando a função XORI, sendo assim, é realizado a operação XOR com um valor de imediato igual a -1, esse imediato resulta em todos os bits iguais a 1, e usando a porta XOR teremos a ”negação”te todos os valores do registrador em operação.