

# Trabalho 4 – Organização e Arquitetura de Computadores

## Geração de Dados Imediatos no RISC-V

Aluno(a): Júlia Yuri Garcia Baba

Matrícula: 19/0057921

Turma C

### 1. Objetivos

O trabalho tem por objetivo desenvolver um código em VHDL, que gere dados imediatos utilizados nas instruções do RISC-V.

### 2. Documentação do código

Primeiramente, foi criado um sinal para identificar a instrução, por meio de um processo foi usado o sinal (opcodes) para receber o comando lido, podendo ser dos tipos R, I, S, SB, U e UJ. Posteriormente, em um segundo processo, o imediato foi desembaralhado e redimensionado para 32 bits.

Para os testes, foi feito um testbench, no qual foi feito um processo de estímulo que atribui valores ao sinal de entrada (inst) e outro processo que monitora a saída do componente declarado, escrevendo na tela o resultado a cada alteração.

### 3. Perguntas

- **Qual a razão do embaralhamento dos bits do imediato no RiscV ?**

A principal razão para o embaralhamento dos bits é a redução do custo, já que essa técnica reduz o número de multiplexadores a serem utilizados, já que assim, o máximo possível de bits estará na mesma posição em cada instrução, simplificando a implementação do hardware.

- **Por que alguns imediatos não incluem o bit 0 ?**

Em RISC-V todas as instruções devem ser alinhadas em 4 bytes, mas através de extensões que permitem tamanho de instruções de 16, 48 ou 64 bits, essas podem ser alinhadas em 2 bytes. Por conta disso, por exemplo, que o bit menos significativo no endereço de destino JARL deve ser sempre zero, já que assim é reutilizado o formato, em vez de criar um novo, onde o imediato seria multiplicado por dois. Sendo assim, alguns imediatos não incluem o bit zero pois essa foi uma forma encontrada para simplificar o hardware, já que permitem que os ponteiros sejam usados para armazenar informações auxiliares. Além de que, há uma pequena chance de um possível erro, já que, se o ponteiro da função contém um endereço inválido, esse erro provavelmente causará uma nova exceção.

- **Os imediatos de operações lógicas estendem o sinal ?**

Os imediatos possuem sempre sinal estendido, e geralmente são compactados do bit mais à esquerda da instrução, sendo alocados para diminuir a complexidade de hardware, com exceção dos imediatos de 5-bits usados nas instruções CSR.

- **Como é implementada a instrução NOT no RiscV ?**

A instrução NOT, que na verdade é uma pseudo-instrução, é executada através da operação lógica XORI, esta executa um XOR bit a bit, no registrador rs1 e estende o sinal de 12 bits do imediato, armazenando o resultado em rd. Dessa forma, para a instrução NOT é feito: XORI rd, rs1, -1, ou seja, uma inversão lógica bit a bit do registrador rs1.