

# **Trabalho 4 – Organização e Arquitetura de Computadores**

## **Geração de Dados Imediatos no Risc-V**

Aluno: Gustavo Pereira Chaves

Matrícula: 19/0014113

Turma: C

### **1. Objetivos**

Este trabalho tem por objetivo a criação de um gerador de imediatos, em vhd, através do binário de uma instrução, retornando o valor em 32 bits.

### **2. Documentação**

Para a implementação do problema, foram desenvolvidos dois arquivos em vhd, o trabalho4.vhd, que armazena propriamente o módulo desenvolvido que gera os imediatos, e o arquivo testbench.vhd, que testa o programa.

#### **2.1. trabalho4.vhd**

Nesse arquivo foi criado o componente genImm32, que recebe como entrada a instrução e tem como saída o imediato em 32 bits. Para isso, o componente utiliza o opcode para encontrar qual o tipo da instrução (I, S, SB, UJ ou U), e com isso identificar a posição dos bits do imediato, decodificando o valor correspondido (utilizando downto para extrair os bits de um intervalo e '&' para agrupá-los) e redimensionando para o tamanho final do vetor preservando o sinal (utilizando a função resize).

Todo esse trabalho é necessário em virtude de que os bits do imediato nem sempre estão em posições consecutivas e na ordem correta. Isso deve principalmente devido a uma economia de hardware, que ao manter os bits do imediato nas mesmas posições, provoca uma redução no número de multiplexadores utilizados.

Já com relação aos imediatos do tipo SB, UJ e U, pode-se notar que estes não armazenam o bit 0. Esses tipos de instruções executam operações baseadas em endereços, o que suprime a necessidade de salvar o último bit, que sempre será zero em decorrência de que todos os endereços são múltiplos de dois.

Para operações lógicas o sinal é sempre estendido, já que se deve preservar o valor passado no programa em assembly. Ou seja, caso o valor seja negativo, este continuará negativo após a sua extensão para 32 bits.

Por fim, a instrução NOT em assembly é emulada, ou seja, é utilizada somente para abstrair do programador toda a operação realizada. Assim, ao executar a instrução NOT, essa em convertida para um XORI (xor bit a bit) com o imediato -1. Sabendo que uma porta XOR com bit 1 sempre inverte o bit e que -1 converte-se somente para bits 1's, essa instrução provoca a inversão de todos os bits passados no registrador rs1.

#### **2.2 testbench.vhd**

Já no testbench, é declarado o componente genImm32 desenvolvido, além dos sinais auxiliares instruction (ligado na entrada) e imm32\_out (ligado na saída). Dessa forma, é realizado um processo síncrono de alteração dos bits de entrada e realizado um teste com o valor esperado utilizando a função assert. Caso o resultado seja diferente é lançado um erro, caso contrário, o programa passa para a próxima instrução.