

**Projeto e Simulação de uma ULA RISC-V em VHDL**  
Dérick Daniel Silva de Andrade - 23/1003522  
CIC0099 - Organização e Arquitetura de Computadores

## **Introdução**

O presente trabalho tem como objetivo a construção e simulação de uma ULA em VHDL para o processador RISC-V de 32 bits no EdaPlayground. O circuito possui três entradas, uma para o opcode da instrução e duas para os dados; e três saídas: o resultado da operação realizada, uma flag para o resultado zero e outra flag para o resultado dos testes de condição. O funcionamento da ULA se dá através do comando case-when para verificar o opcode e comandos if then else para atribuir o valor das flags de zero e de condição.

## **Comparações com e sem sinal**

As comparações sem sinal são feitas utilizando o conversor unsigned() que faz com que o número seja considerado como um inteiro positivo. Já a comparação com sinal é feita com o conversor signed e faz com que o número esteja representado em complemento de 2 com o bit mais significativo como representante do sinal.

## **Detecção de overflow**

A detecção de overflow na soma pode ser feita a partir da comparação do bit mais significativo de ambos números, se ambos forem iguais e o resultado da soma for diferente deles, então ocorreu overflow.

Na subtração, o overflow ocorre quando subtrai-se números de sinais opostos e pode ser detectado comparando o bit mais significativo de ambos, se forem diferentes verifica-se o sinal do subtraendo, que deve ser o oposto do sinal do resultado.

Segue a tabela e as forma de onda do testbench:

A	Op	B	Z
3	ADD	5	8
3	ADD	-3	0
3	ADD	-5	-2
5	SUB	3	2
5	SUB	5	0
5	SUB	-9	14
5	SUB	10	-5
0xFFFFFFFF	AND	0x0F0F0F0F	0x0F0F0F0F
0x0000FFFF	OR	0xFFFF0000	0xFFFFFFFF
0xAAAAAAAA	XOR	0x55555555	0xFFFFFFFF
0x00000001	SLL	0x00000004	0x00000010
5	SRL	1	2
0x80000005	SRA	0x00000001	0xC0000002
3	SLT	4	1
4	SLT	3	0

A	Op	B	Z
4	SLT	4	0
0xFFFFFFFFE	SLTU	0xFFFFFFFFF	1
0xFFFFFFFFF	SLTU	0xFFFFFFFFE	0
0xFFFFFFFFF	SLTU	0xFFFFFFFFF	0
4	SGE	3	1
4	SGE	-4	0
3	SGE	4	0
4	SGE	4	1
0xFFFFFFFFF	SGEU	0xFFFFFFFFE	1
0xFFFFFFFFE	SGEU	0xFFFFFFFFF	0
0xFFFFFFFFF	SGEU	0xFFFFFFFFF	1
0xFFFFFFFFF	SEQ	0xFFFFFFFFF	1
0xFFFFFFFFF	SEQ	0xFFFFFFFFE	0
0xFFFFFFFFF	SNE	0xFFFFFFFFE	1

