

# Arquitetura e Organização de Computadores

## Turma C - 2018/01

### Projeto e Simulação de uma ULA em VHDL

**Objetivo:** projetar, simular e sintetizar uma versão da ULA do MIPS de 32 bits no ambiente Quartus / ModelSim-Altera.

#### Características:

- Duas entradas de dados: A e B
- Uma Saída de dados: Z
- Sinal Zero: detecta valor zero na saída
- Sinal Overflow: ativo quando a operação de soma ou subtração gerar resultado que ultrapasse o limite de representação em 32 bits
- Operações (em Complemento de 2):

Operação	Significado	OpCode
and A, B	Z recebe a operação lógica A and B, bit a bit	0000
or A, B	Z recebe a operação lógica A or B, bit a bit	0001
add A, B	Z recebe a soma das entradas A, B	0010
addu A, B	Z recebe a soma das entradas A, B, sem overflow	0011
sub A, B	Z recebe A - B	0100
subu A, B	Z recebe A - B, sem overflow	0101
slt A, B	Z = 1 se A < B	0110
sltu A, B	Z = 1 se unsigned(A) < unsigned(B)	0111
nor A, B	Z recebe a operação lógica A nor B, bit a bit	1000
xor A, B	Z recebe a operação lógica A xor B, bit a bit	1001
sll A, B	Z recebe B deslocado de A bits à esquerda	1010
srl A, B	Z recebe deslocamento lógico de B por A bits à direita	1011
sra A, B	Z recebe deslocamento arit. de B por A bits à direita	1100
rtr A, B	Z recebe B rotacionado de A bits à direita	1101
rtl A, B	Z recebe B rotacionado de A bits à esquerda	1110

#### Interface:

```
entity ulaMIPS is
    generics (WSIZE : natural := 32);
    port (
        opcode      : in std_logic(4 downto 0)
        A, B        : in std_logic_vector(WSIZE-1 downto 0);
        Z           : out std_logic_vector(WSIZE-1 downto 0)
        zero, ovfl  : out std_logic);
end ulaMIPS;
```

onde:

- *opcode* indica a operação a ser realizada
- *A e B*: operandos, 32 bits.
- *Z*: saída, 32 bits
- *zero*: indicação de zero na saída
- *ovfl*: overflow

**Bibliotecas:** utilizar a biblioteca *numeric\_std* do VHDL para as operações aritméticas sobre os vetores lógicos.

**Simulação e Verificação:** simular o funcionamento da ULA de forma a verificar o funcionamento de cada uma das suas operações. Verificar igualmente a geração dos sinais *zero* e *ovfl*. Utilizar o ModelSim Altera para a simulação, desenvolvendo um *testbench* para acionamento dos sinais.

A verificação deve incluir a execução de ao menos um teste para cada operação da ULA. As operações aritméticas devem ser testadas para resultado zero, negativo, positivo e *overflow*.

**Relatório:** configurar o Quartus para sintetizar a ULA para o FPGA Cyclone II EP270F896C6N. Informar o resultado da síntese para cada caso.

**Entrega:**

- Incluir o código da ULA e do *testbench*
- Mostrar as telas de simulação no ModelSim
- Incluir os dados da síntese que são apresentados pelo Quartus II