

## RELATÓRIO – TRABALHO 1 – SISTEMAS DIGITAIS 2023.2

**ALUNOS: ARTHUR MONTEIRO, CAROLINA FALCÃO, MARIA VICTORIA**

. Descrição: O trabalho consiste em uma ULA (unidade lógico-aritmética) de 4 *bits* de entrada, codificada em VHDL na placa FPGA SPARTAN XCS700AN, a qual conta com 8 operações (soma, subtração, incremento de 1 unidade, inversão de *bits*, deslocamento para a direita, deslocamento para a esquerda, comparação e complemento a 2), implementadas de forma modular (cada operação conta com 4 módulos, que realizam o procedimento *bit a bit*) e selecionadas através de um módulo multiplexador de 8 entradas, contido na própria ULA.

As duas entradas (A e B) têm seus 3 bits de magnitude gerados por um contador síncrono de módulo 64 (000000 a 111111 – 3 *bits* mais significativos são de B, e os 3 menos, de A), com um *clock* de 2 segundos, e os seus *bits* de sinal possuem um valor fixo.

Ao inicializar a placa, com o programa já carregado, seleciona-se a operação utilizando três chaves seletoras (SW0, SW1 e SW2). Os valores das entradas A e B, do resultado da operação selecionada e dos sinais de *flag – carry-out* (LD0), resultado nulo (LD1), resultado negativo (LD2) e *overflow* (LD3) - são demonstrados, nesta ordem, por *leds*, conforme mudança de *clock* (LD0, LD1 e LD2, em grau crescente de significância, representam a magnitude; LD4 indica o bit de *sinal*; por último, são demonstrados os sinais de *flag*, com seus respectivos *leds*). Dessa forma, tem-se quatro estados de exibição: valor de A (LD6 e LD7 apagados), valor de B (LD6 aceso), valor do resultado (LD7 aceso) e valor dos sinais de *flag* (LD6 e LD7 acesos). O processo é feito de forma cíclica.

. Funcionalidade: As funções de cada operação:

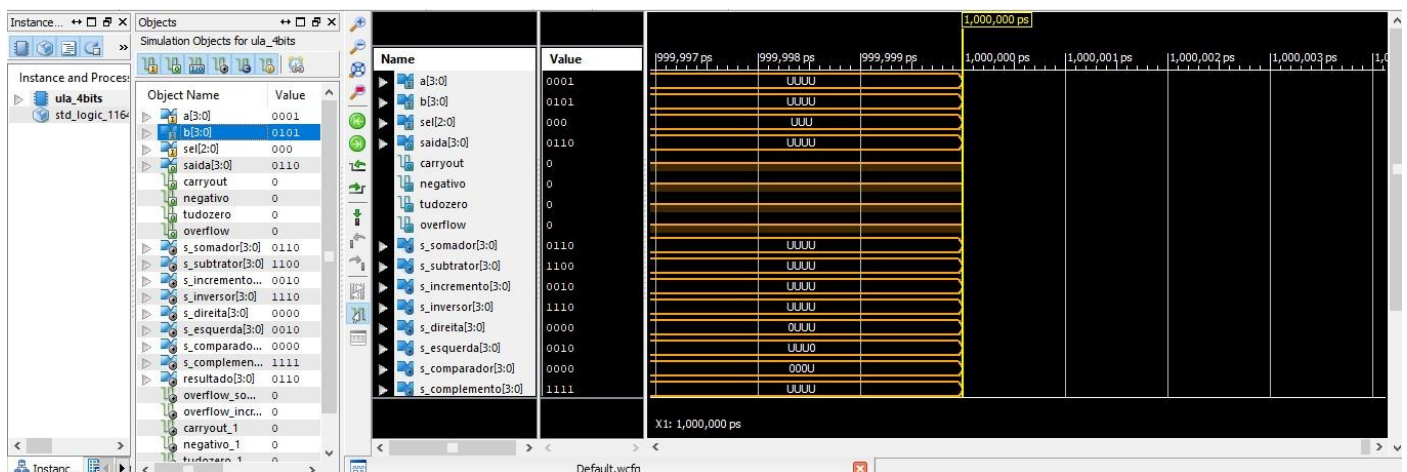
- a) Soma: calcula  $A + B$
- b) Subtração: calcula  $A - B$
- c) Incremento de 1 unidade: soma 1 unidade a A

- d) Inversão de *bits*: inverte todos os *bits* de A
- e) Deslocamento para a direita: desloca os *bits* de A para a direita, igualando a 0 o MSB
- f) Deslocamento para a esquerda: desloca os *bits* de A para a esquerda, igualando a 0 o LSB
- g) Comparação: retorna 1 caso  $A > B$
- h) Complemento a 2: calcula o complemento a 2 de A, para  $A > 0$

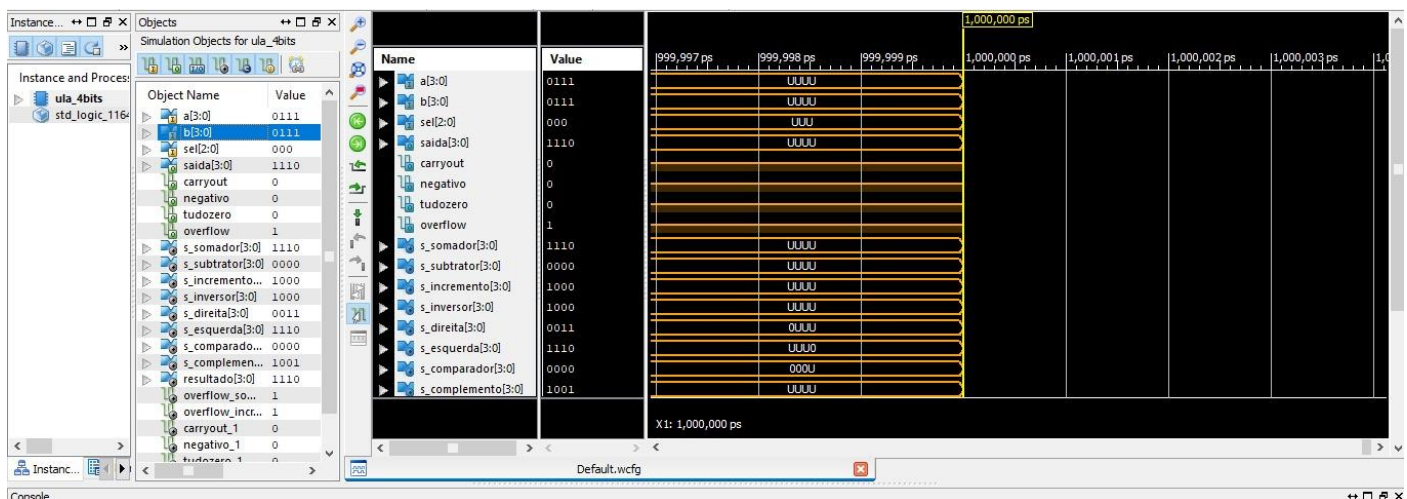
. Resultados: A seguir, alguns exemplos das operações implementadas:

1) Soma ( $SW2 = 0 \mid SW1 = 0 \mid SW0 = 0$ )

a)  $A = 0001 (+1) \mid B = 0101 (+5)$

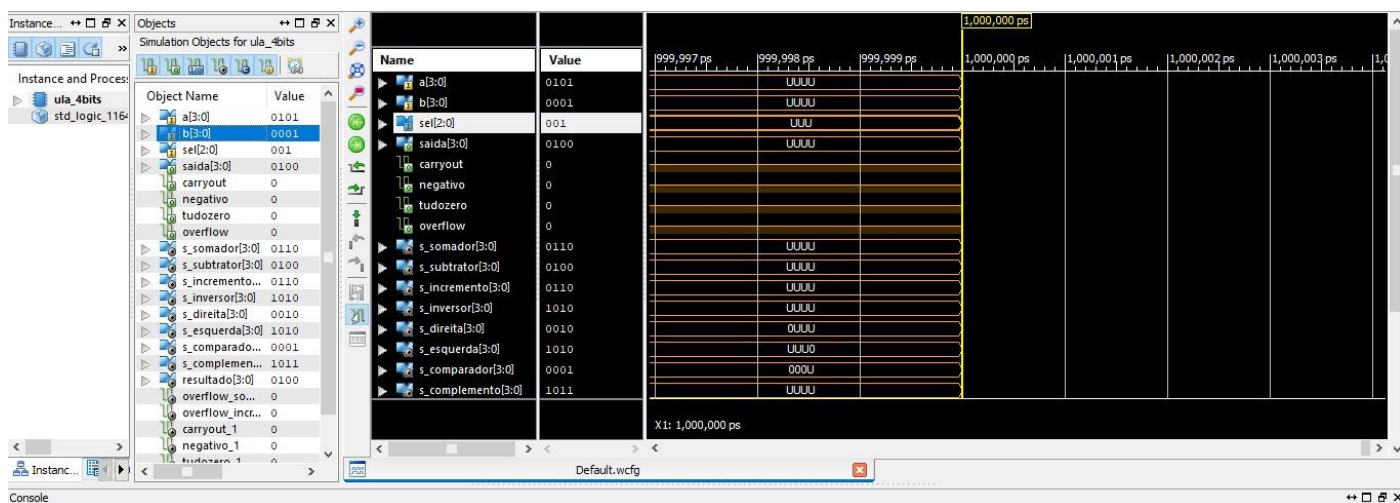


b)  $A = 0111 (+7) \mid B = 0111 (+7) \mid \text{Flag de overflow}$



## 2) Subtração (SW2 = 0 | SW1 = 0 | SW0 = 1)

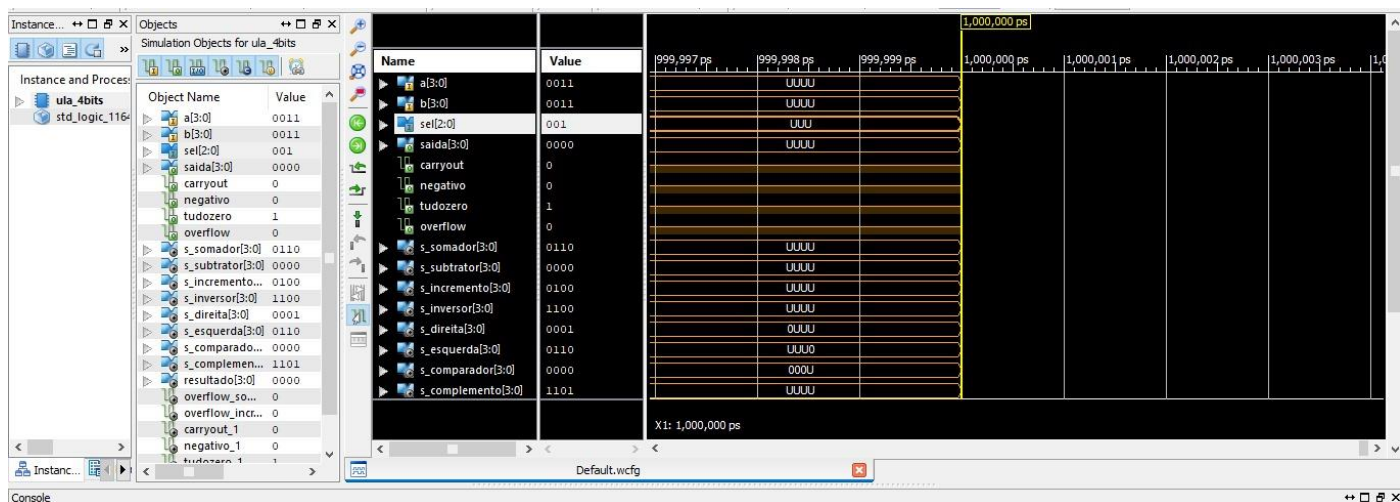
a) A = 0101 (+5) | B = 0001 (+1)



b) A = 0101 (+1) | B = 0001 (+5) | Flag de resultado negativo

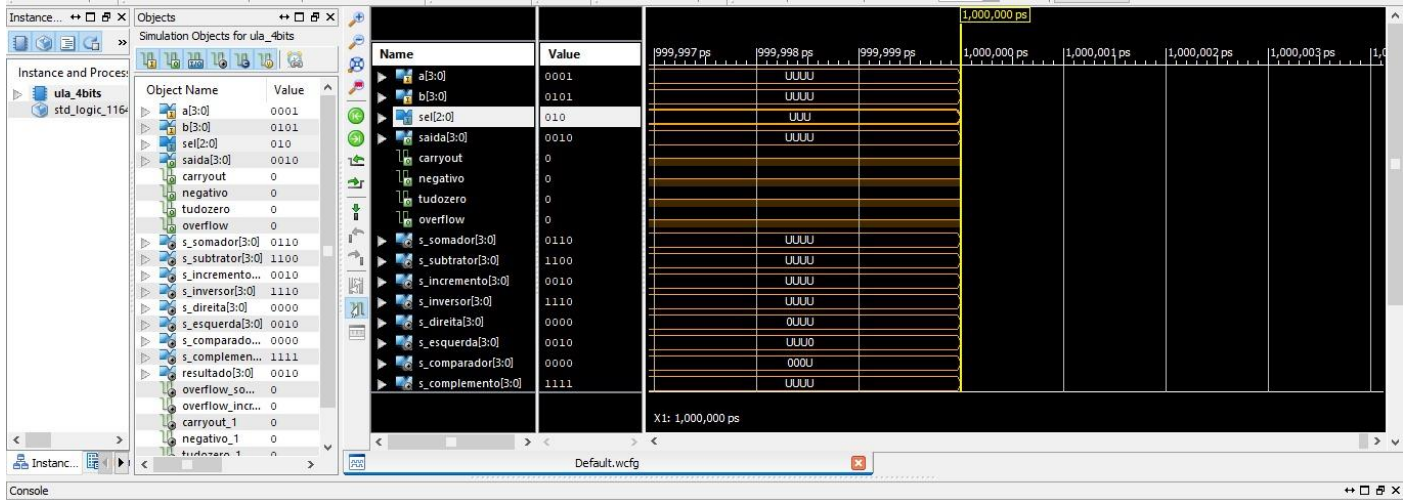


c) A = 0011 (+3) | B = 0011 (+3) | Flag de resultado nulo

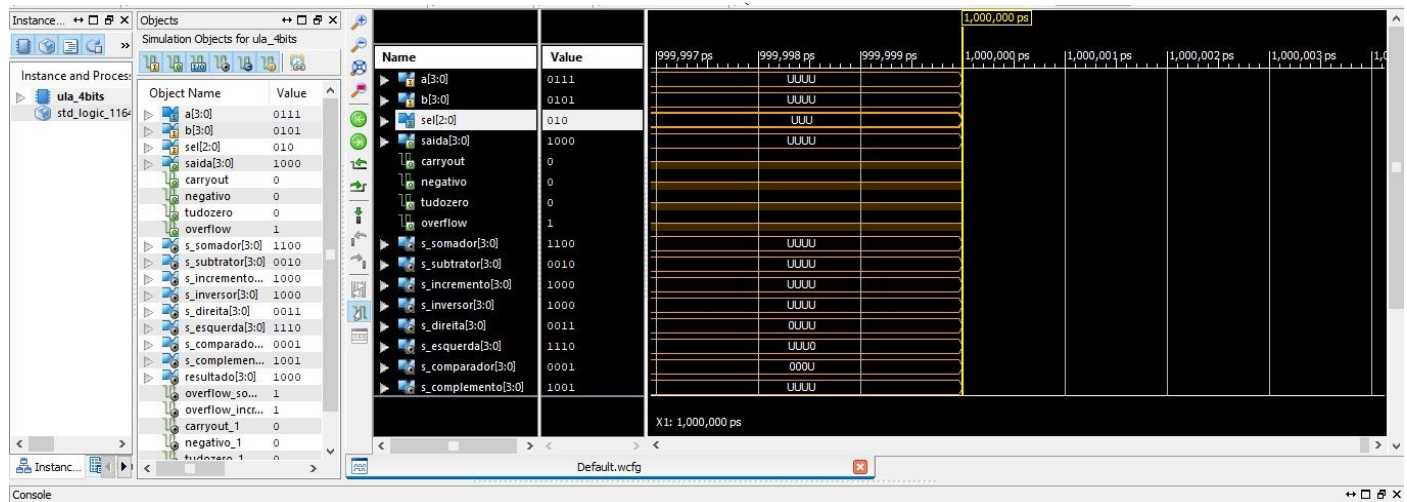


### 3) Incremento de 1 unidade (SW2 = 0 | SW1 = 1 | SW0 = 0)

a) A = 0001 (+1)



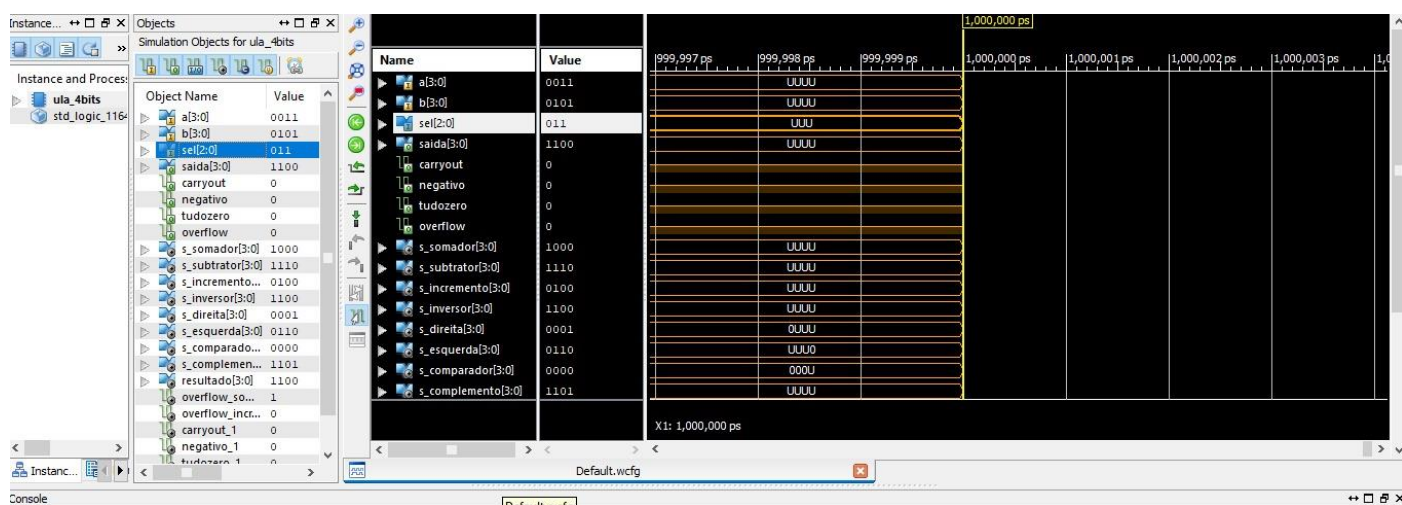
b) A = 0111 (+7) | *Flag de overflow*





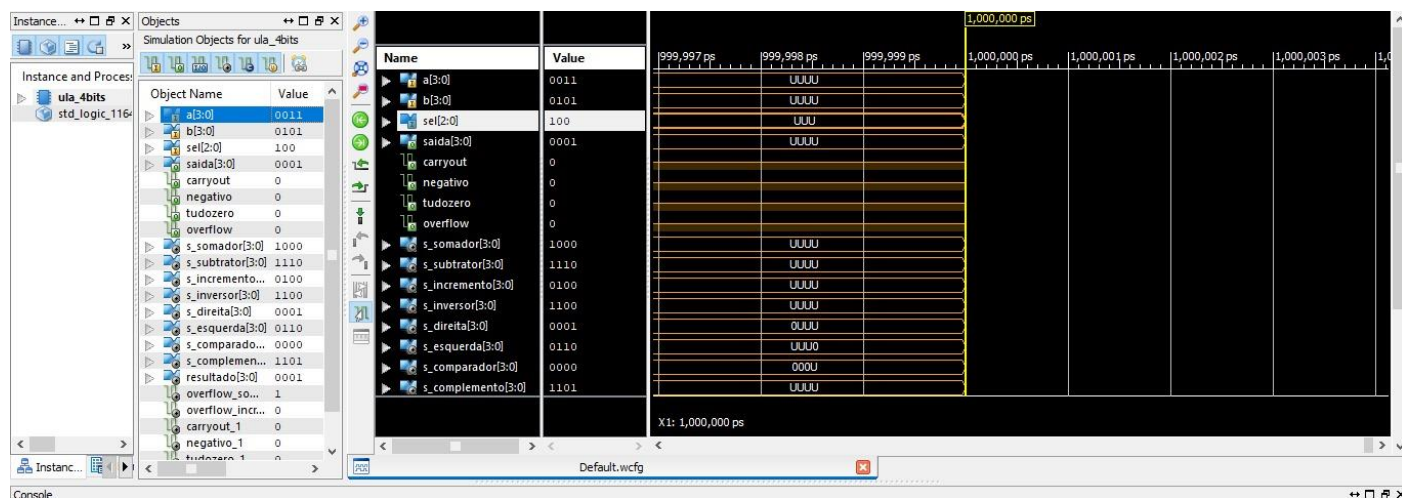
#### 4) Inversão de *bits* (SW2 = 0 | SW1 = 1 | SW0 = 1)

a) A = 0011

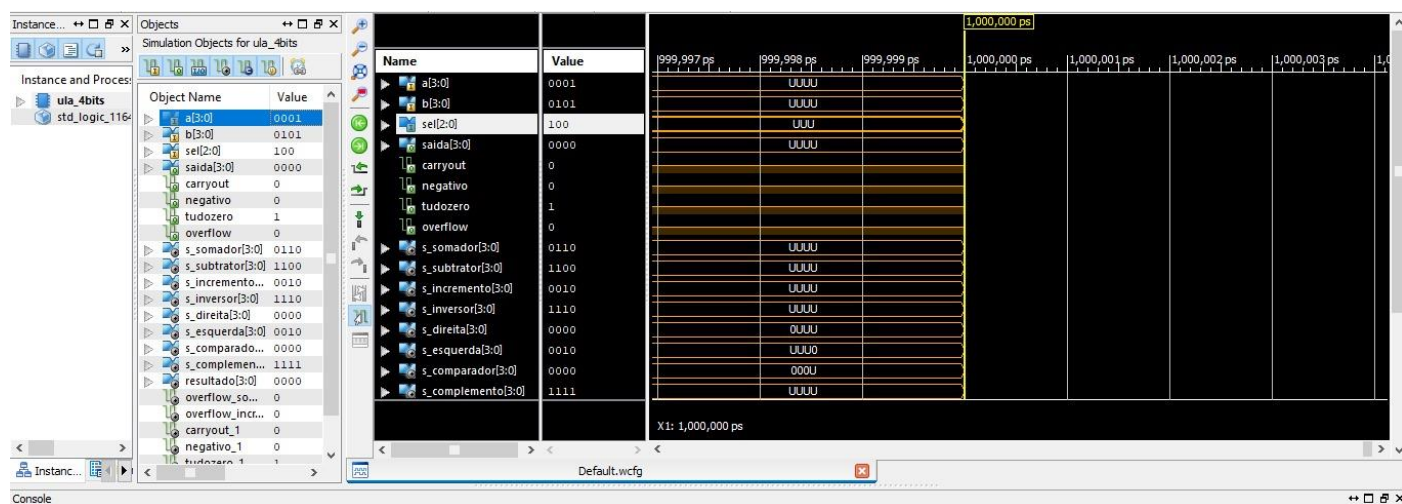


#### 5) Deslocamento para a direita (SW2 = 1 | SW1 = 0 | SW0 = 0)

a) A = 0011

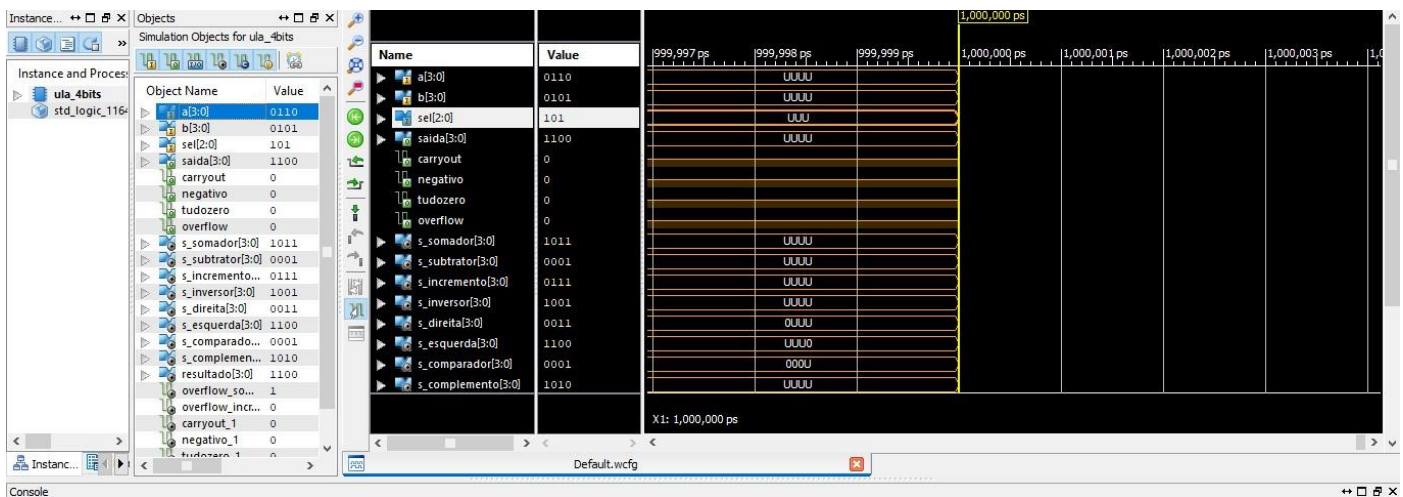


b) A = 0001 | *Flag* de resultado nulo



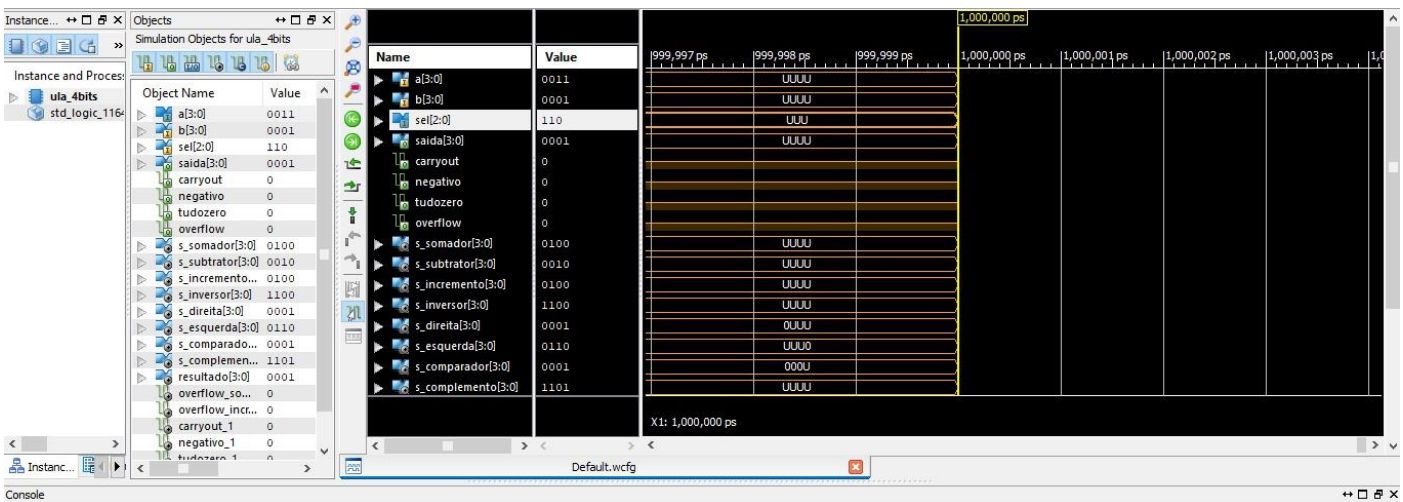
## 6) Deslocamento para a esquerda (SW2 = 1 | SW1 = 0 | SW0 = 1)

a) A = 0110

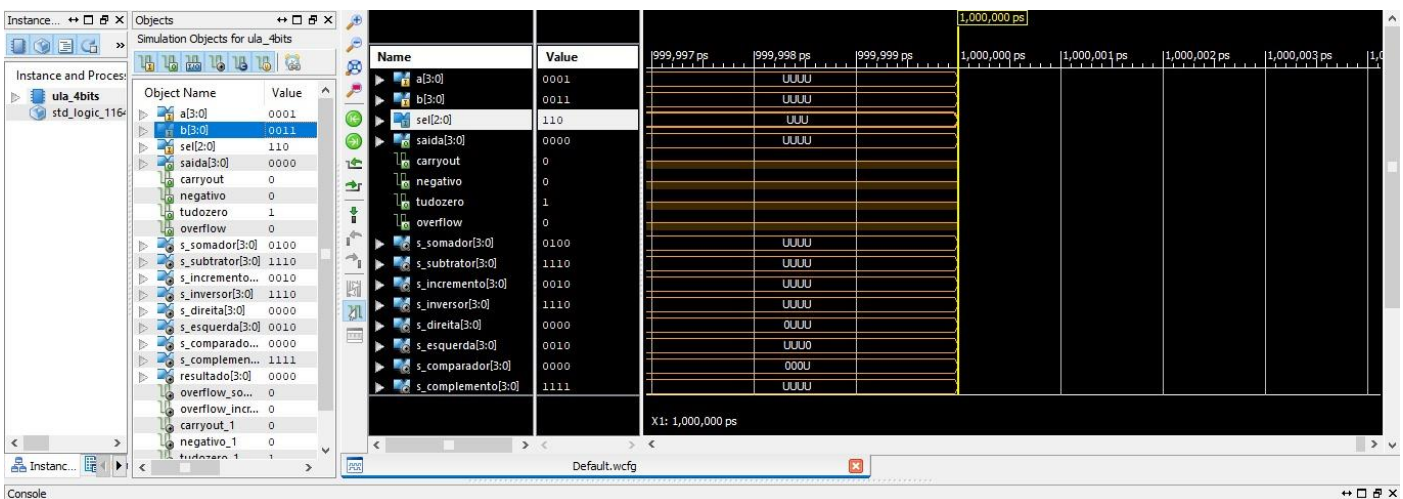


## 7) Comparação (SW2 = 1 | SW1 = 1 | SW0 = 0)

a) A = 0011 (+3) | B = 0001 (+1)

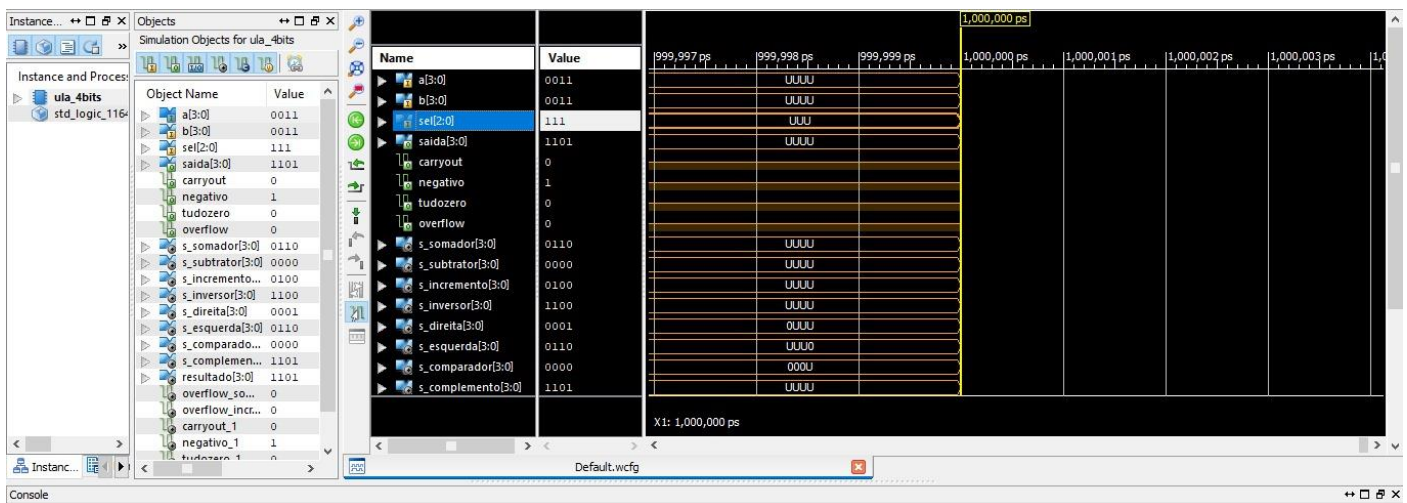


b) A = 0001 (+1) | B = 0011 (+3)



8) Complemento a 2 (SW2 = 1 | SW1 = 1 | SW0 = 1)

a) A = 0011 (+4)



Link do vídeo que demonstra o funcionamento da placa: <https://youtu.be/srT57ZQhvk8?>

Link do projeto no GitHub: <https://github.com/carolinafernandezds/Trabalho-1-SD>