



Universidade Federal do Rio de Janeiro
Escola Politécnica – Departamento de Eletrônica e Computação
EEL480 – Sistemas Digitais
– Professora Mariana Maciel

Unidade Lógico-Aritmética

Nome: Daniel Rodrigues Ferreira
DRE: 118024378

Nome: Lucas Christensem Lima
DRE: 119050665

Nome: kauã gomes de paiva da costa
DRE: 122070496

Data: 31/10/2023

- Introdução

No primeiro trabalho de Sistemas Digitais foi designado aos alunos o desenvolvimento de uma ULA que recebe dois números de 4 bits capaz de realizar oito operações, tanto aritméticas quanto lógicas, e criando uma interface com a mesma. Esse trabalho foi feito utilizando uma placa Spartan-3AN e a linguagem de programação VHDL, através do programa ISE da Xilinx.

As oito operações serão:

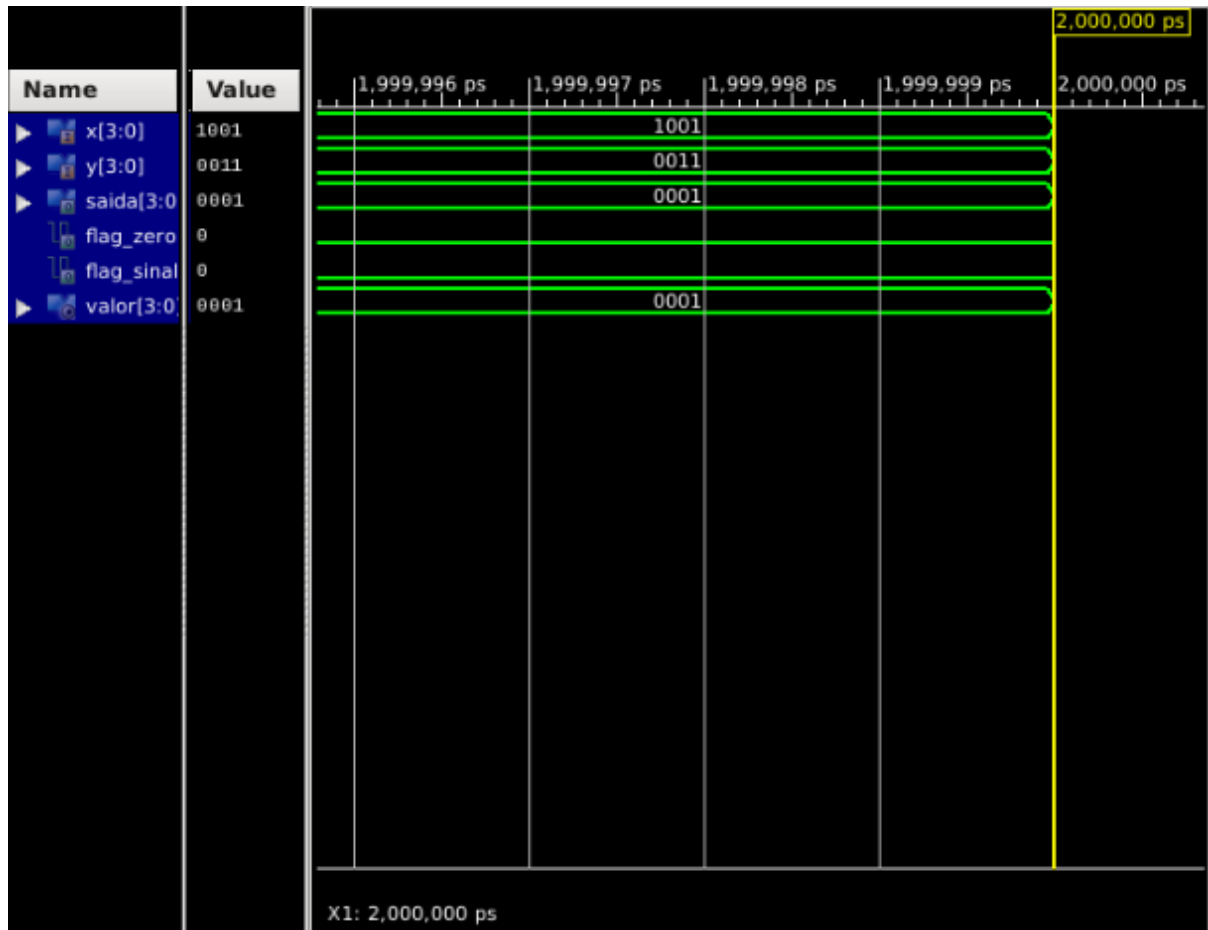
- And
- Or
- Inversor
- Xor
- Soma
- Incremento
- Subtração
- Multiplicação
- Desenvolvimento
- Portas Lógicas:

A primeira operação realizada pelos alunos foram as funções lógicas. Já são operações nativas em VHDL. Portanto foram criadas duas entradas A e B de 4 bits e estas são comparadas bit-a-bit de acordo com a função lógica utilizada.

- Função AND:

A função AND recebe duas entradas X e Y e sua saída que é 1 se ambas as entradas forem 1 e 0 caso contrário.

Figura 1:

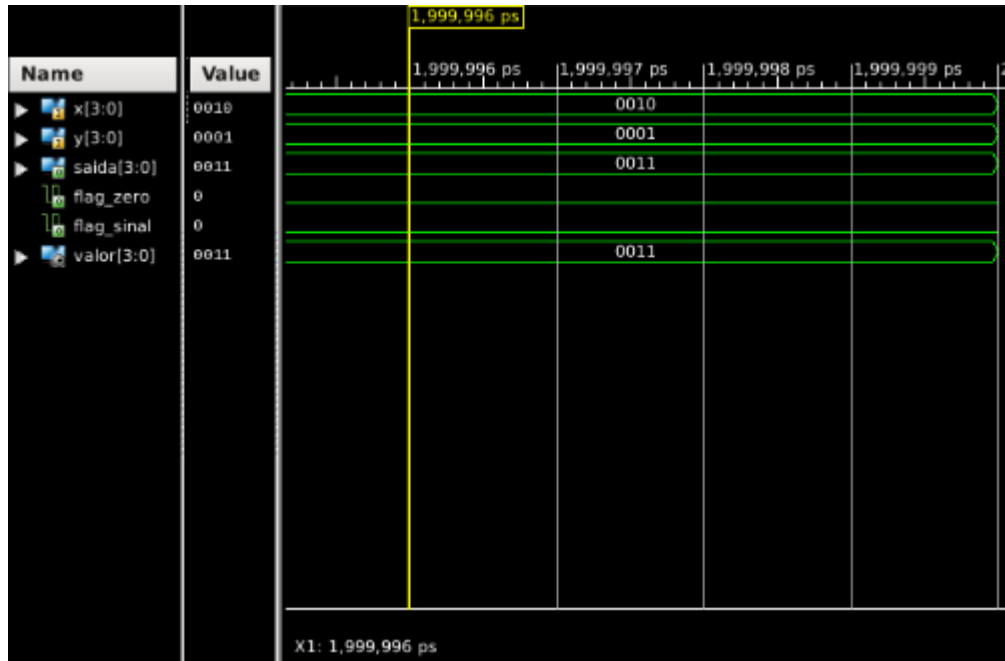


Simulação da função AND.

- Função OR:

A função OR recebe duas entradas X e Y e sua Saída é 1 se qualquer uma das entradas for 1 e 0 caso contrário.

Figura 2:

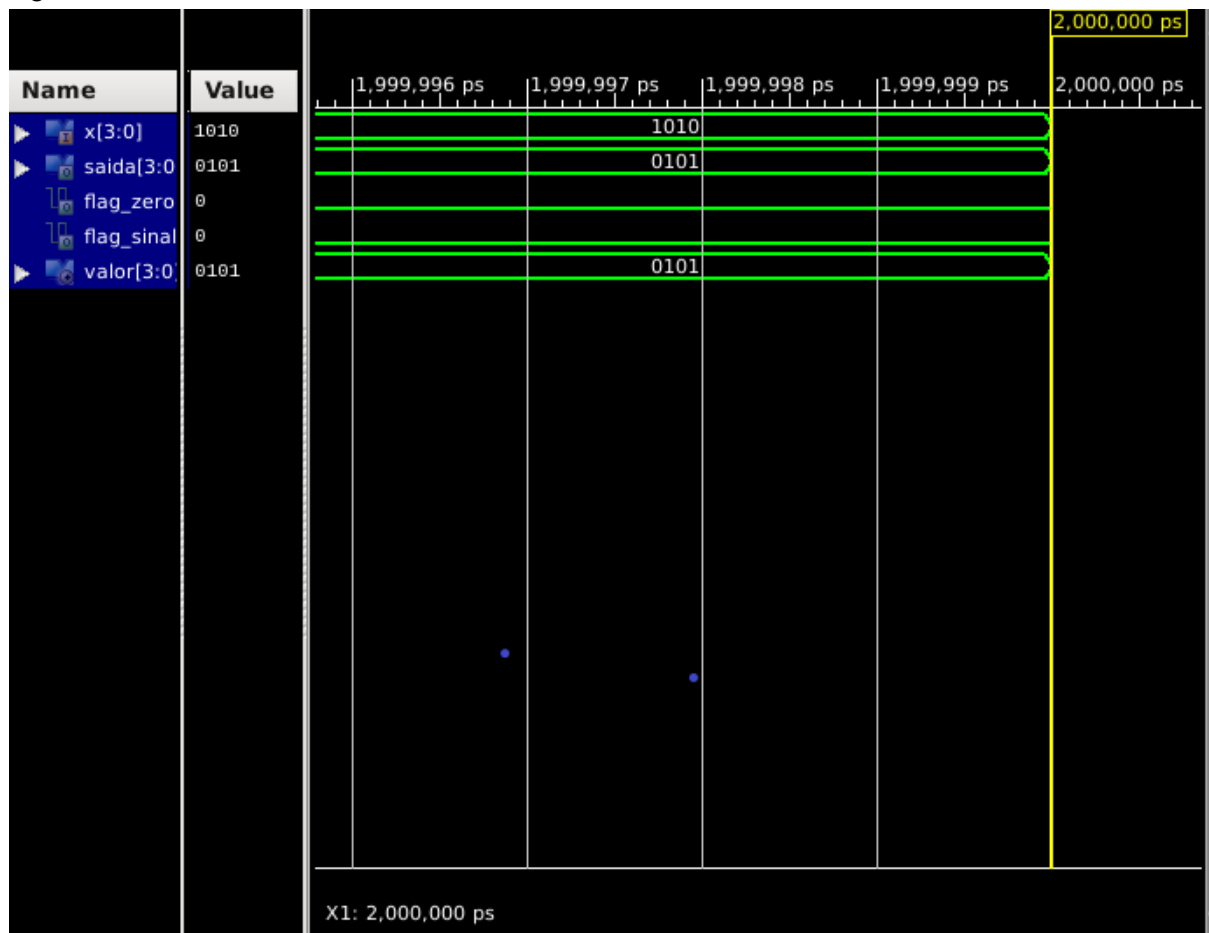


Simulação da função OR.

- Função Inversor:

A função Inversor, também chamada de Not, recebe uma entrada X e sua saída é o inverso da entrada. Se receber 0 a saída é 1, se receber 1 sua saída é 0.

Figura 3:



Simulação da função Inversor.

- Função XOR:

A função XOR, ou “ou exclusivo”, recebe duas entradas A e B e tem saída 1 se somente uma das entradas for 1 e 0 caso contrário.

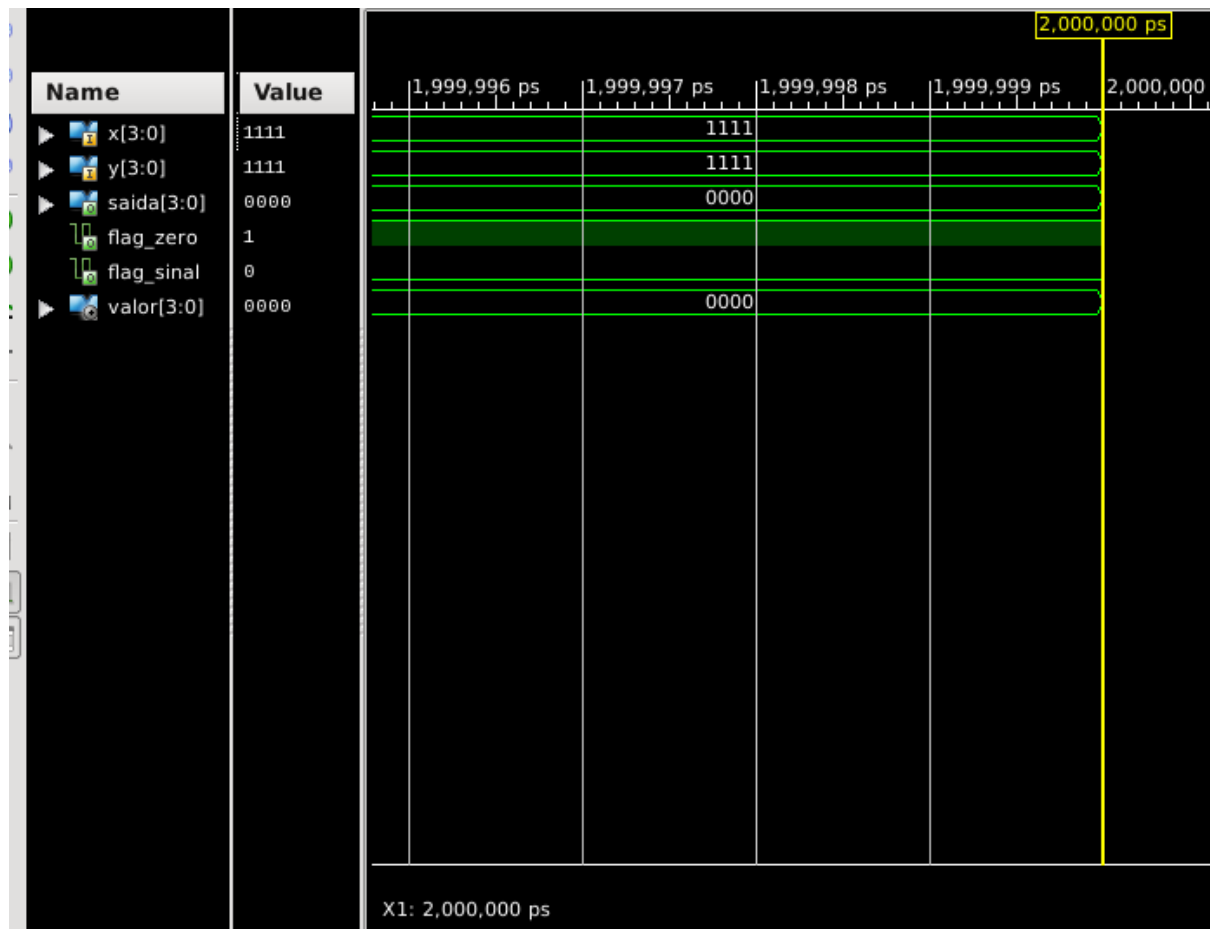


Figura 4:

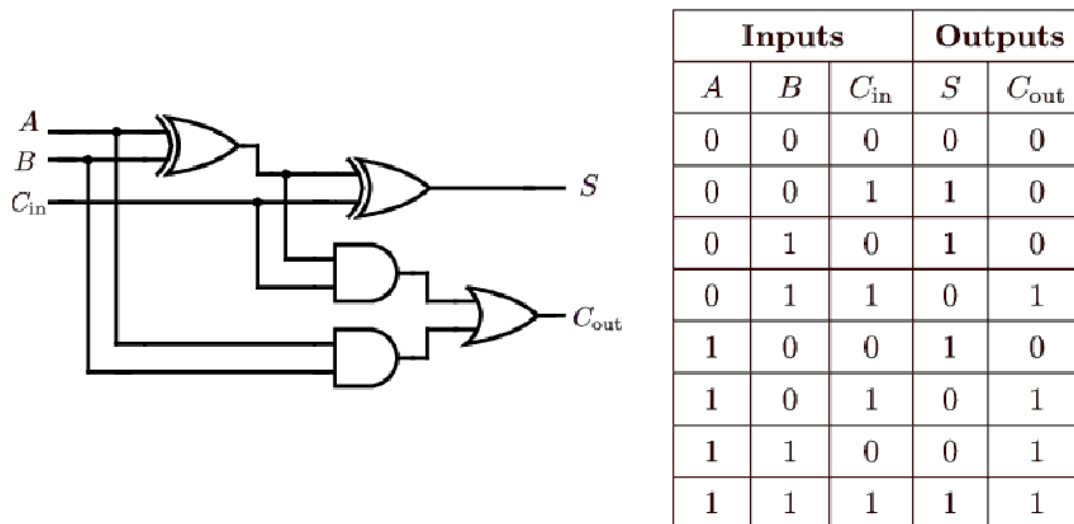
Simulação da função XOR.

OBS.: Vale ressaltar que nesse exemplo foi possível verificar que a flag de sinal está funcionando corretamente.

- Somador:

A primeira função obrigatória implementada foi a do somador. Esta função recebe dois números de 4 bits e faz a soma binária entre eles. Foi criado um somador simples e depois ele foi usado como uma entidade para a soma de 4 bits. Esse somador simples foi criado de acordo com a tabela verdade e design abaixo:

Figura 5:

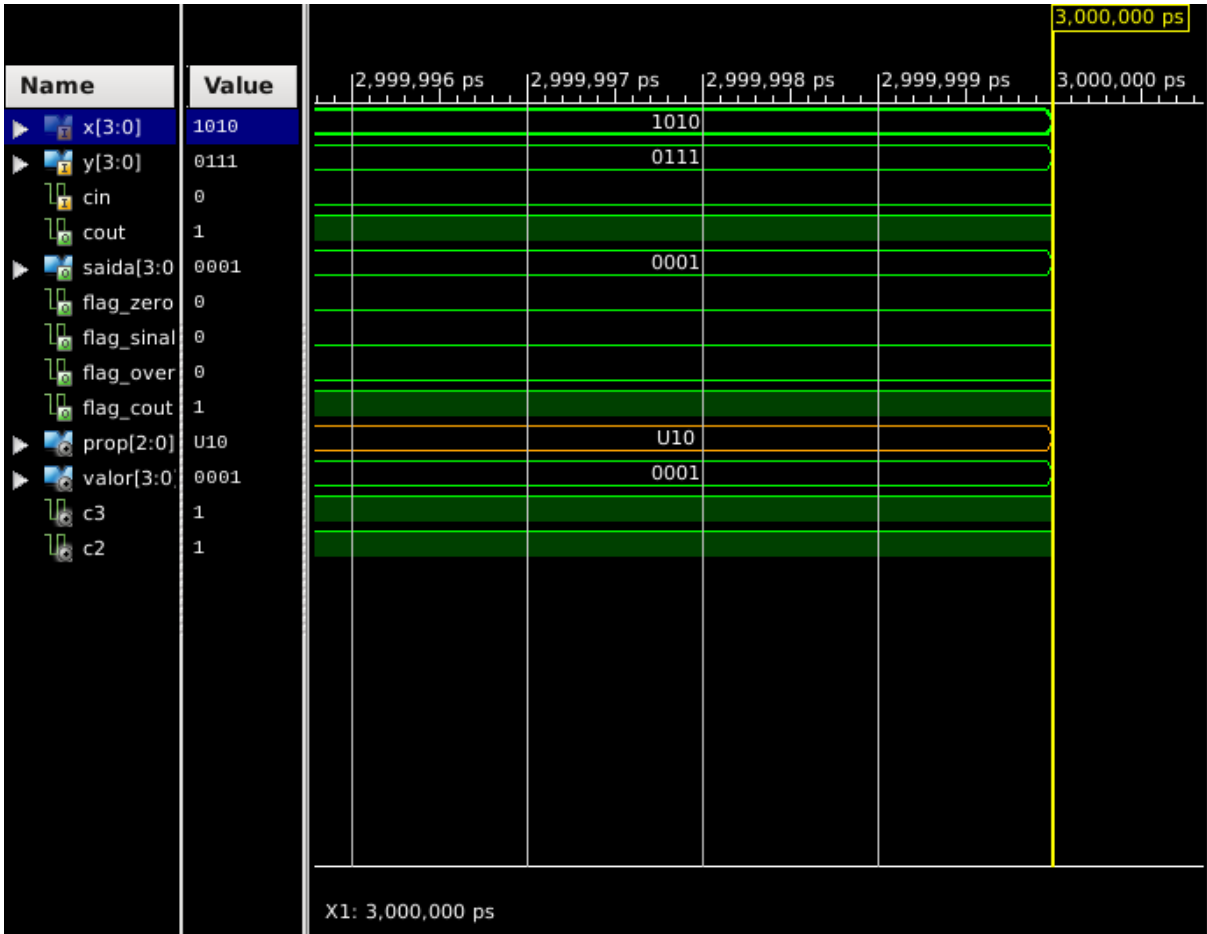


Desenho circuito de um somador e sua tabela verdade.

Para implementar esse desenho foram criados cinco sinais x1, x2, x3, x4 e y1 e usando as operações lógicas foi implementado o full-adder de um bit de acordo com o código no apêndice.

Com este código, foi implementado um “component” no desenvolvimento do somador de 4 bits, e criado três sinais de C_{in} para serem os carries do próximo bit. Utilizando a função “port map” com a função de somador escrita anteriormente para cada bit do somador, ficou pronto o somador de 4 bits.

Figura 6:

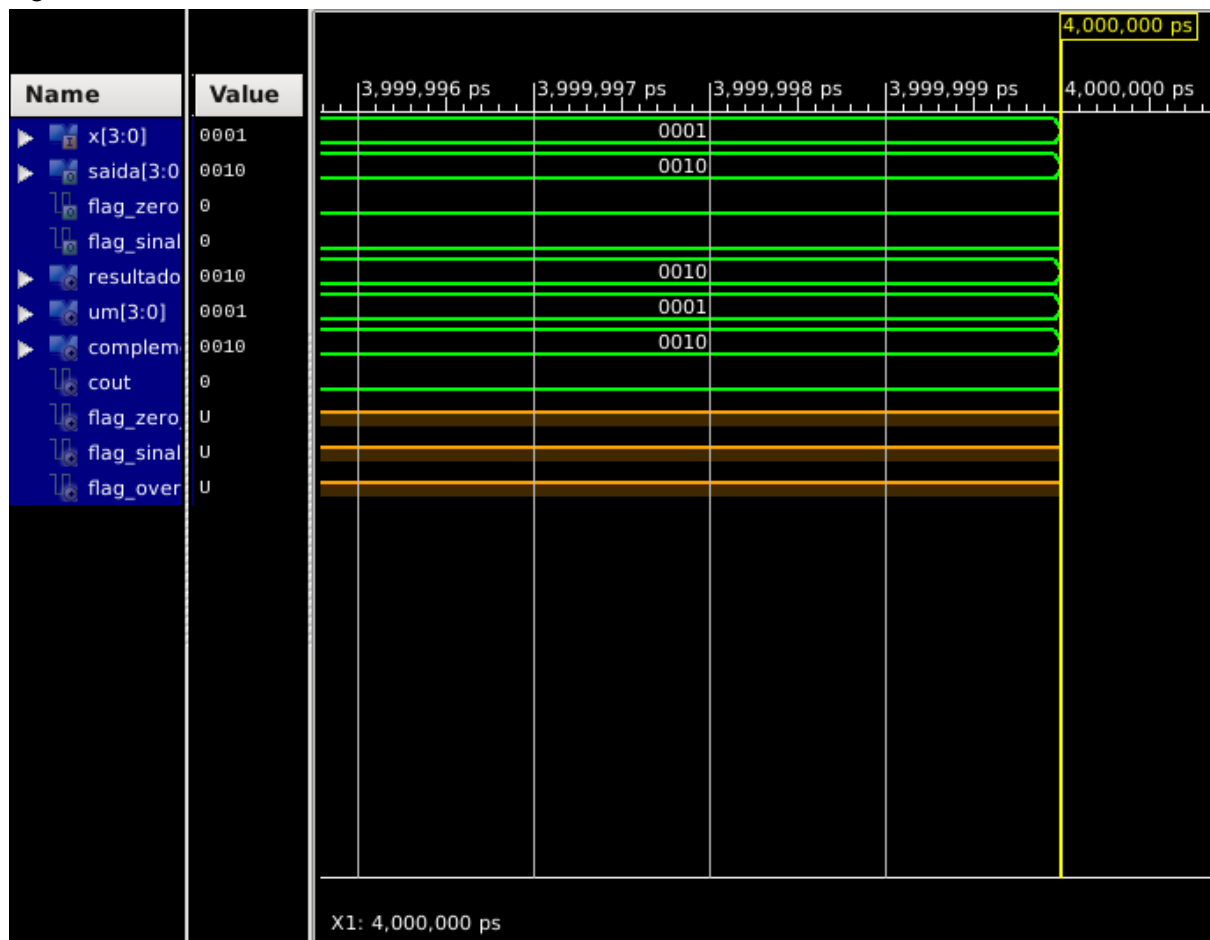


Simulação do somador.

- Incremento de 1:

A segunda operação implementada foi o incremento de 1. Esta operação consiste em utilizar o somador de 4 bits e adicionar uma unidade a este número. Portanto é feito um “port map” e somamos “0001” ao vetor X.

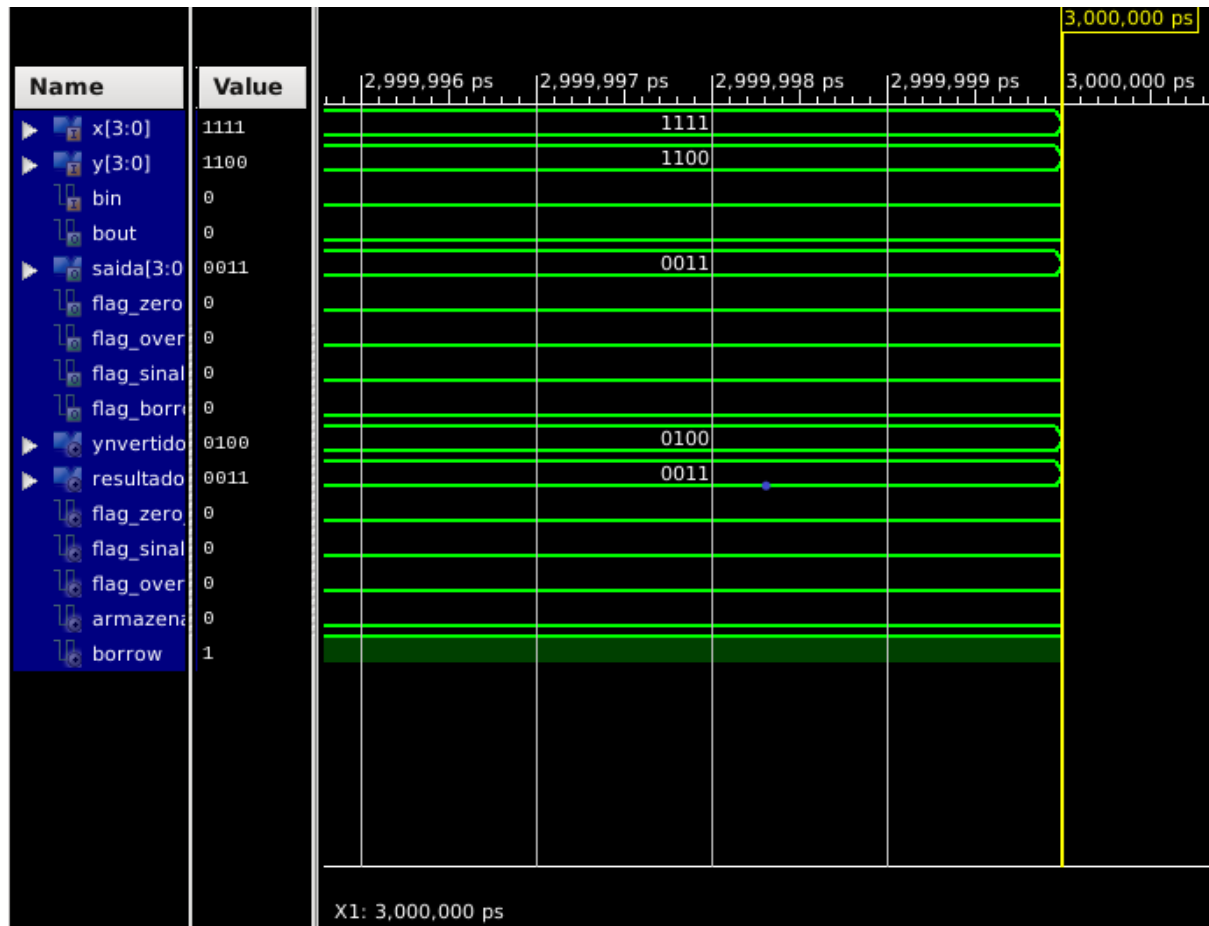
Figura 7:



Simulação da operação de soma e do incremento de 1.

- Subtrator:
O subtrator foi a segunda operação obrigatória a ser implementada e também faz uso do somador. Ele é definido como um “component” assim como no somador de 4 bits, porém sua segunda entrada é invertida usando a função NOT.

Figura 8:

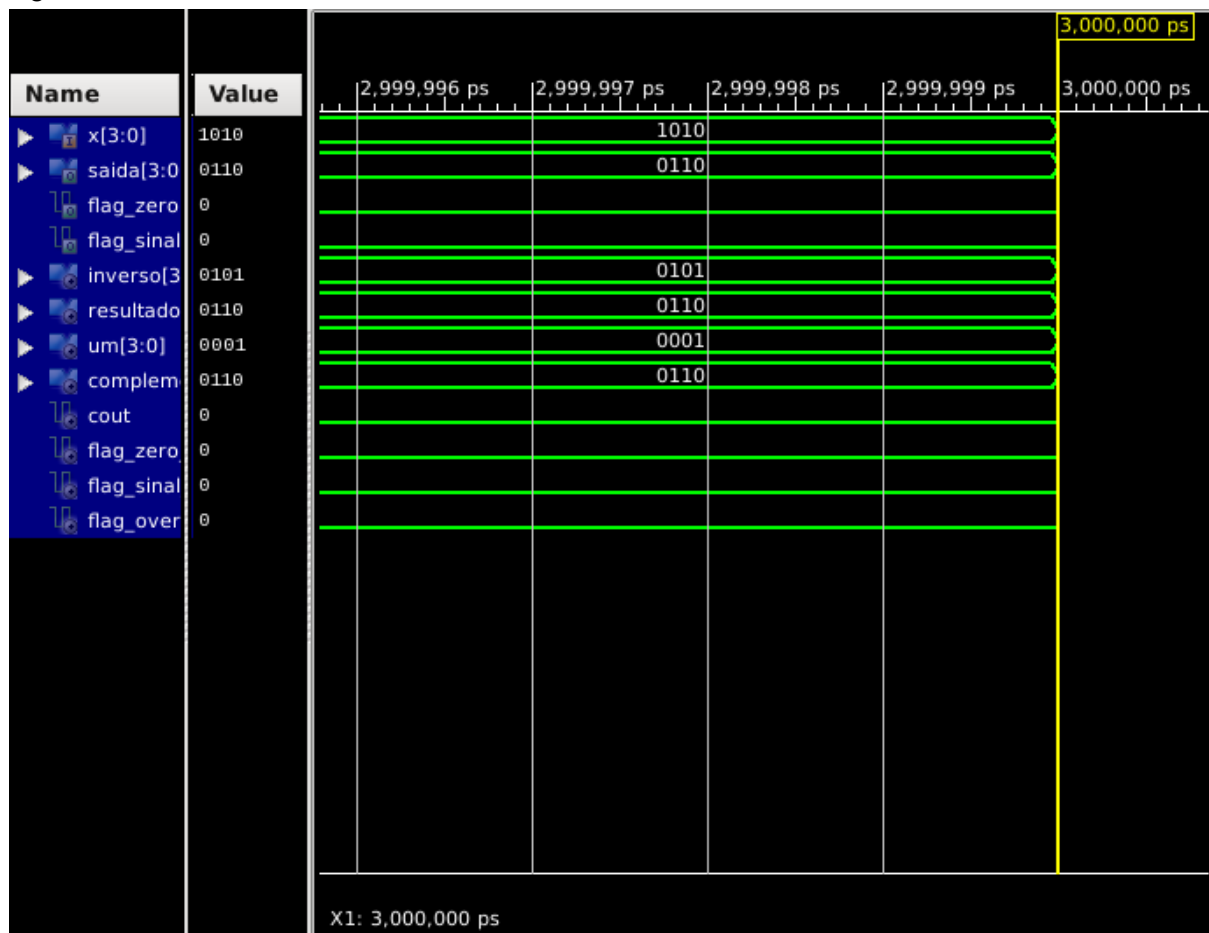


Simulação da operação de subtração.

- Complementa:

A terceira operação implementada foi o incremento de 1, esta escolhida pelos alunos. Esta operação consiste em inverter o número de 4 bits utilizando a função NOT e em seguida utilizar o somador para adicionar 1.

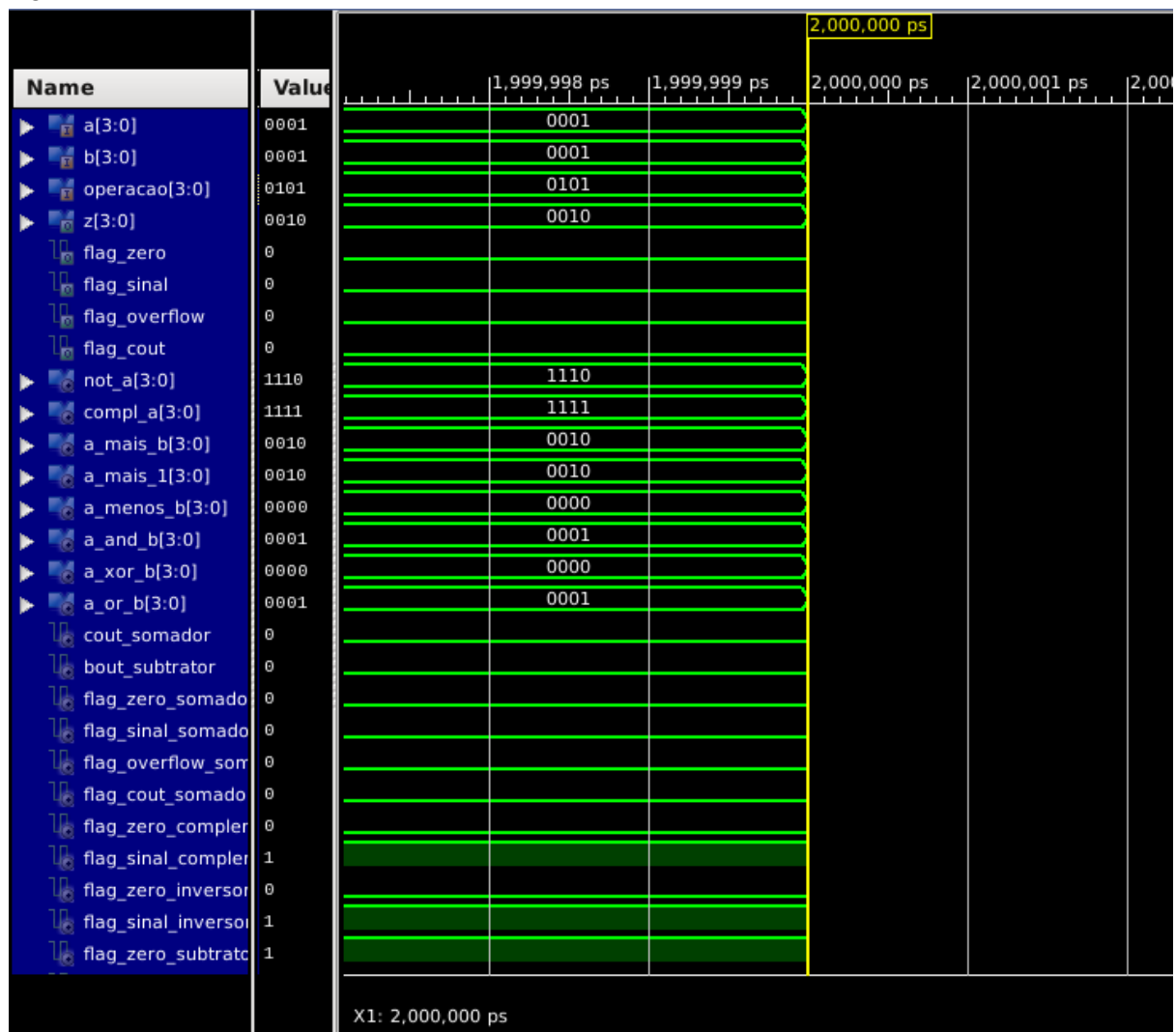
Figura 9:



Simulação da operação de complemento a 2.

- ULA:
Por fim a ULA foi simulada, com todas as suas funções.

Figura 9:



Simulação da ULA.

- FPGA:
Na placa os leds acenderam da seguinte forma, os 4 leds mais a esquerda servem somente para indicar em qual step estamos, ou seja, irão variar de 1 até 4. Já os leds da direita irão mostrar os valores de cada step, sendo:
step 1 = Vetor A
step 2 = Vetor B
step 3 = Vetor resultado (de acordo com a operação) entre A e B
step 4 = Flags
- Conclusão:
Neste trabalho foi possível que os alunos começassem a ter domínio de algumas funções mais básicas da linguagem VHDL e implementassem os circuitos FPGA estudados durante as aulas teóricas da matéria. Usando o software ISE da Xilinx conseguiu-se implementar

uma ULA mesmo com algumas dificuldades se apresentando no sentido de aplicação na placa.

Link do github : <https://github.com/lucaschristensemlima/ULA-VHDL-SD-2023>