

# Experimento 2

João Pedro Fernandes Santos

222025342

TURMA – 09

## Introdução

Nesse experimento iremos descrever em VHDL e simular no ModelSim os dois circuitos combinacionais implementados por meio do Logisim no experimento 1, sendo eles um somador completo e um multiplexador 4x1, ambos baseados em FPGA.

## QUESTÃO 1 Teoria

O somador completo é composto por: três bits de entrada (A, B e Cin) e dois bits de saída (S e Cout) e descrito pelas seguintes funções lógicas:

$$S = A \oplus B \oplus \text{Cin}$$

$$\text{Cout} = A \cdot B + A \cdot \text{Cin} + B \cdot \text{Cin}$$

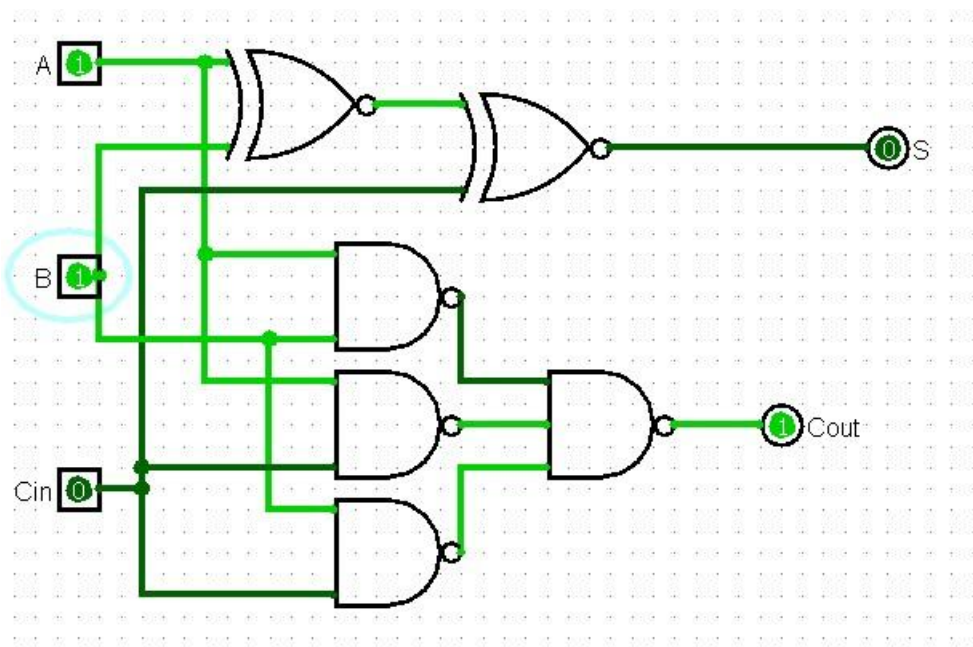


Figura 1. Exemplo de somador completo implementado no Logisim.

## Código

Utilizando a linguagem de descrição de hardware VHDL pelo software Modelsim, implementamos e testamos o circuito do somador completo em código para testar nossa implementação gerando entradas “A”, “B” e “Cin”, de 1 bit cada, com saídas “S” e “Cout”, também com 1 bit cada.

Ln#	
1	LIBRARY IEEE;
2	USE IEEE.STD_LOGIC_1164.ALL;
3	
4	ENTITY MEUCIRCUITO IS
5	PORT (A, B, Cin: IN STD_LOGIC;
6	S, Cout: OUT STD_LOGIC);
7	END MEUCIRCUITO;
8	
9	ARCHITECTURE MEUCIRCUITO_ARCH OF MEUCIRCUITO IS
10	BEGIN
11	S <= (A XOR B) XOR Cin;
12	Cout <= (A AND B) OR (A AND Cin) OR (B AND Cin);
13	END MEUCIRCUITO_ARCH;

Figura 2. Código somador em VHDL.

## Compilação

Os códigos gerados foram compilados para garantir seu funcionamento, como mostrado abaixo, nenhum dos códigos apresenta erros de sintaxe.

```
# Compile of EXP_2_1.vhd was successful.  
# Compile of EXP_2_2.vhd was successful.
```

Figura 4. Compilação mostrando 0 erros, 0 avisos.

## Simulação

A simulação foi implementada aplicando a função clock às 3 entradas “A”, “B” e “Cin” com diferentes períodos (50ps, 100ps e 150ps, respectivamente), para ilustrar como os diferentes valores de entrada, geram diferentes valores de saída, de acordo com o código implementado na figura 2.

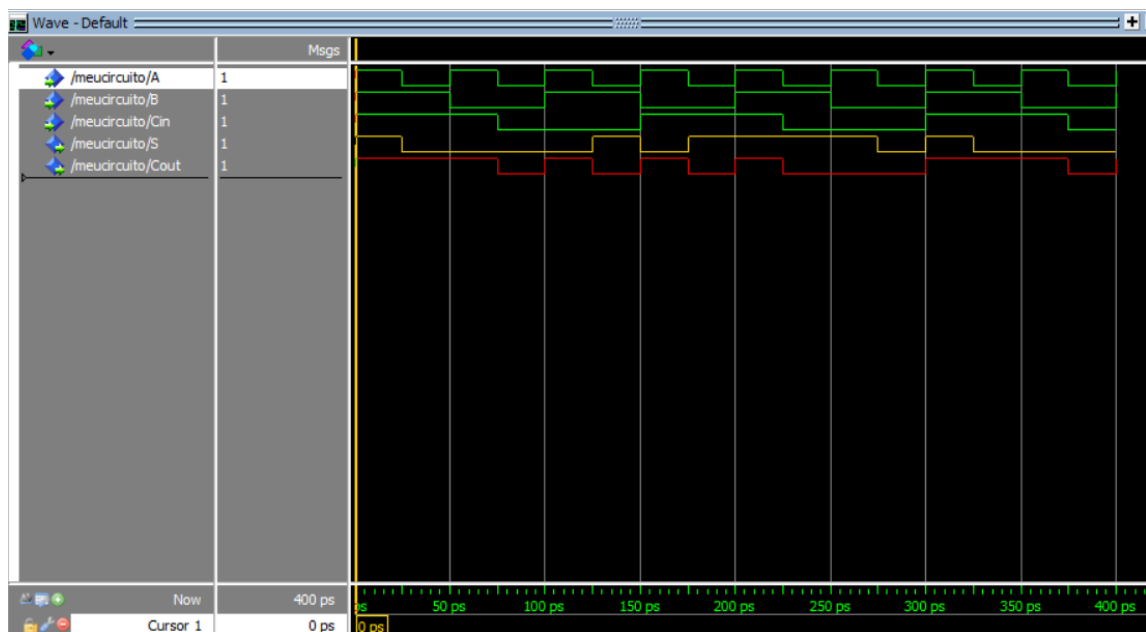


Figura 5. Simulação de variação dos valores de entrada em verde, e resposta dos valores de saída, em amarelo e vermelho.

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figura 6. Tabela verdade do circuito de somador completo.

## Análise

Observa-se que aplicando todos os valores de entrada, em períodos diferentes, às variáveis “A”, “B” e “Cin”, representados pelas linhas verdes como sinais periódicos de 50 ps, 100 ps e 150 ps, respectivamente, tem-se como resposta, as saídas “S” e “Cout”, representadas pelas linhas amarela e vermelha, respectivamente, ambas com 25ps de período. Comparando a simulação do ModelSim com a tabela verdade do somador no Logisim, percebe-se que, os resultados conferem, provando que a implementação do circuito pelo ModelSim foi exitosa.

## Conclusão

Neste experimento conseguimos com êxito implementar e descrever o comportamento de um somador completo bem como entender suas características. Por meio das simulações no Modelsim, foi possível obter os dados esperados e compará-los com a tabela verdade do circuito elaborado no logisim. Não foram encontrados erros ou divergências na realização do experimento.

## QUESTÃO 2 Teoria

O multiplexador 4x1 é composto por: dois vetores de entrada (S com 2 bits e D com 4 bits) e um bit de saída (Y) e descrito pela seguinte função lógica:

$$Y = D_0\bar{S}_1\bar{S}_0 + D_1\bar{S}_1S_0 + D_2S_1\bar{S}_0 + D_3S_1S_0$$

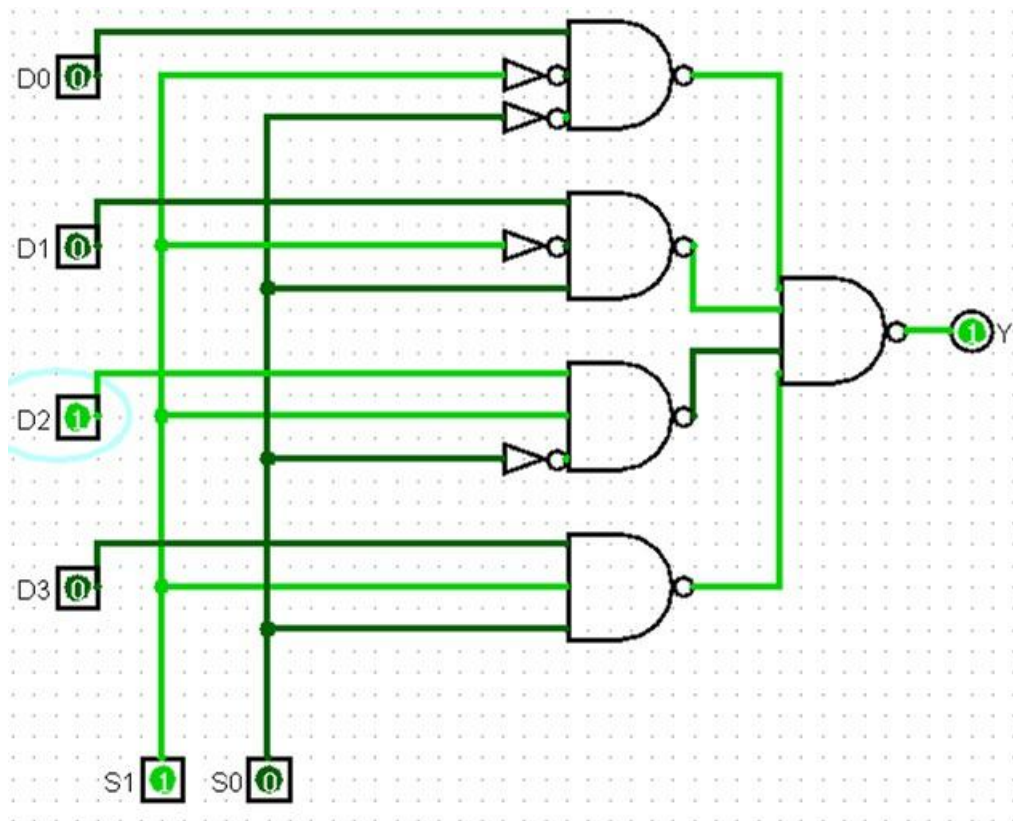


Figura 7. Exemplo de multiplexador 4x1 implementado no Logisim.

## Código

Utilizando a linguagem de descrição de hardware VHDL pelo software Modelsim, nos moldes da primeira questão, implementamos o multiplexador 4x1, com 2 vetores de entrada, S com 2 bits e D com 4 bits, e 1 vetor de saída Y com 1 bit apenas.

Ln#	
1	LIBRARY IEEE;
2	USE IEEE.STD_LOGIC_1164.ALL;
3	
4	ENTITY MEUCIRCUITO2 IS
5	PORT (S: IN STD_LOGIC_VECTOR(0 TO 1);
6	D: IN STD_LOGIC_VECTOR(0 TO 3);
7	Y: OUT STD_LOGIC);
8	END MEUCIRCUITO2;
9	
10	ARCHITECTURE MEUCIRCUITO2_ARCH2 OF MEUCIRCUITO2 IS
11	BEGIN
12	Y <= (D(0) AND (NOT(S(1))) AND (NOT(S(0)))) OR (D(1) AND (NOT(S(1)))
13	AND S(0)) OR (D(2) AND S(1) AND (NOT(S(0)))) OR (D(3) AND S(1) AND S(0));
14	END MEUCIRCUITO2_ARCH2;

Figura 8. Código multiplexador em VHDL.

## Compilação

Os códigos gerados foram compilados para garantir seu funcionamento, como mostrado abaixo, nenhum dos códigos apresenta erros de sintaxe.

```
# Compile of EXP_2_2.vhd was successful.
```

Figura 9. Compilação mostrando 0 erros, 0 avisos.

## Simulação

A simulação foi implementada aplicando a função clock às cada um dos bits das entradas “S” e “D” com diferentes períodos variando de 50 em 50 ps para ilustrar como os diferentes valores de entrada, geram diferentes valores de saída, de acordo com o código implementado na figura 8.

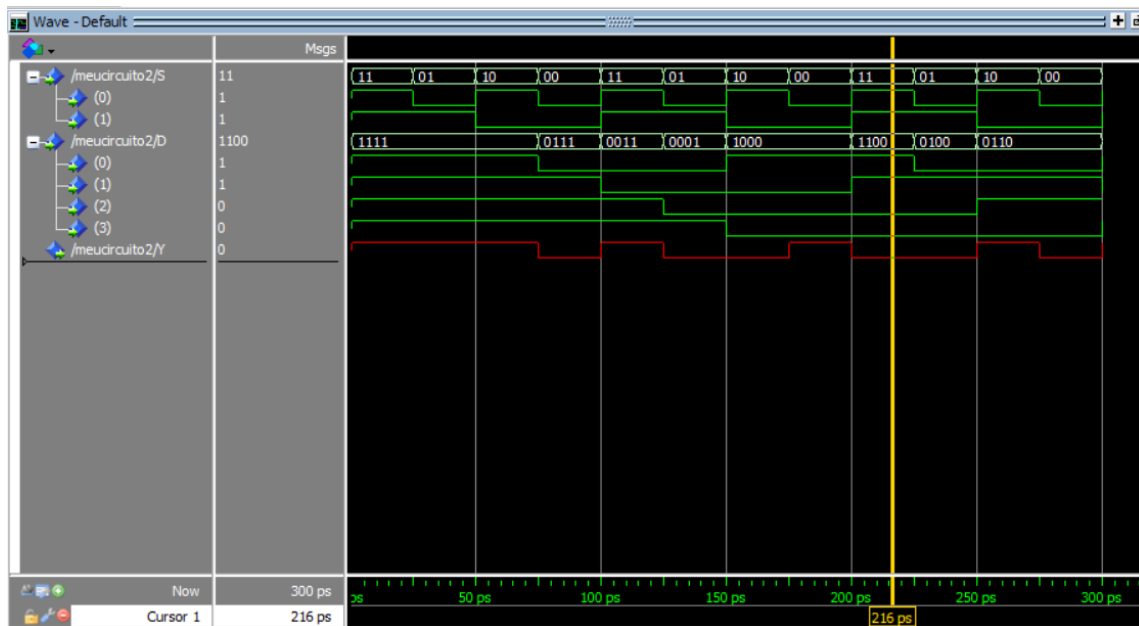


Figura 10. Simulação de variação dos valores de entrada em verde, e resposta dos valores de saída, em vermelho.

D0	D1	D2	D3	S1	S0	Y
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	1	1	0
0	0	0	1	0	0	0
0	0	0	1	0	1	0
0	0	0	1	1	0	0
0	0	0	1	1	1	1
0	0	1	0	0	0	0
0	0	1	0	0	1	0
0	0	1	0	1	0	1
0	0	1	0	1	1	0
0	0	1	1	0	0	0
0	0	1	1	0	1	0
0	0	1	1	1	0	1
0	0	1	1	1	1	1
0	1	0	0	0	0	0
0	1	0	0	0	1	1
0	1	0	0	1	0	0
0	1	0	0	1	1	0
0	1	0	1	0	0	0
0	1	0	1	0	1	1
0	1	0	1	1	0	0
0	1	0	1	1	1	1
0	1	1	0	0	0	0
0	1	1	0	0	1	1
0	1	1	0	1	0	1
0	1	1	0	1	1	0
0	1	1	1	0	0	0
0	1	1	1	0	1	1
0	1	1	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	1
1	0	0	0	0	1	0
1	0	0	0	1	0	0
1	0	0	0	1	1	0
1	0	0	1	0	0	1
1	0	0	1	0	1	0
1	0	0	1	1	0	0
1	0	0	1	1	1	1
1	0	1	0	0	0	1
1	0	1	0	0	1	0
1	0	1	0	1	0	1
1	0	1	0	1	1	0
1	0	1	1	0	0	1
1	0	1	1	0	1	0
1	0	1	1	1	0	1
1	0	1	1	1	1	1
1	1	0	0	0	0	1
1	1	0	0	0	1	1
1	1	0	0	1	0	0
1	1	0	0	1	1	0
1	1	0	1	0	0	1
1	1	0	1	0	1	1
1	1	0	1	1	0	0
1	1	0	1	1	1	1
1	1	1	0	0	0	1
1	1	1	0	0	1	1
1	1	1	0	1	0	1
1	1	1	0	1	1	0
1	1	1	1	0	0	1
1	1	1	1	0	1	1
1	1	1	1	1	0	1
1	1	1	1	1	1	1

Figura 11. Tabela verdade do multiplexador.

### Análise

Observa-se que aplicando todos os valores de entrada, em períodos diferentes, em cada um dos bits das variáveis “S” e “D”, representados pelas linhas verdes como sinais periódicos de período variando em 50ps, tem-se como resposta, as saídas “Y”, representada pela linha vermelha, 25ps de período. Comparando a simulação do ModelSim com a tabela verdade do somador no Logisim, percebe-se que, os resultados conferem, provando que a implementação do circuito pelo ModelSim foi exitosa.

### Conclusão

Neste experimento conseguimos com êxito implementar e descrever o comportamento de um multiplexador 4x1 bem como entender suas características. Por meio das simulações no Modelsim, foi possível obter os dados esperados e compará-los com a tabela verdade do circuito elaborada no logisim. Não foram encontrados erros ou divergências na realização do experimento.