

Universidade de Brasília – UNB

Curso: Engenharia de Redes de Comunicação

Disciplina: Laboratório de Sistemas Digitais

Turma: 08



Relatório da Disciplina Laboratório de Sistemas Digitais

Tema: Experimento 02 – Introdução à
linguagem VHDL

Aluno: Pedro Henrique Dias Avelar

Matrícula: 241037112

Professor: Eduardo Paiva

Introdução

O presente experimento tem os seguintes objetivos:

- Implementar circuitos combinacionais simples utilizando a linguagem VHDL.
- Desenvolver módulos básicos – somador completo e multiplexador – que podem ser usados futuramente para implementar circuitos mais complexos.
- Simulação no ModelSim.

Experimento 01 – Somador Completo

O somador completo é descrito pelas funções lógicas abaixo:

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + AC_{in} + BC_{in}$$

Podemos representar as funções acima no Logisim pelo código abaixo:

```
-- Experimento 02 - Questão 01

-- Aluno: Pedro Henrique Dias Avelar 241037112
-- Turma 08
-- Data: 02/11/2024

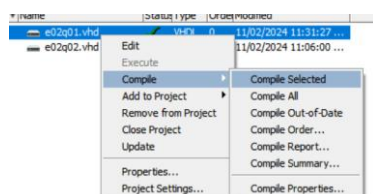
-- Funções lógicas do somador completo:
-- S = A xor B xor Cin
-- Cout = AB or ACin or BCin

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY CIRCUITO_E02Q01 IS
    PORT (A,B,CIN: IN STD_LOGIC;          --ENTRADAS
          S,COUT: OUT STD_LOGIC);         --SAÍDAS
END CIRCUITO_E02Q01;

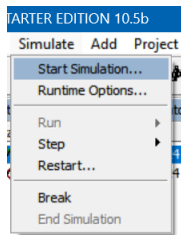
ARCHITECTURE ARC_CIRCUITO_E02Q01 OF CIRCUITO_E02Q01 IS
BEGIN
    S <= A XOR B XOR CIN;                  --S = A xor B xor Cin
    COUT <= (A AND B) OR (A AND CIN) OR (B AND CIN); --Cout = AB or ACin or BCin
END ARC_CIRCUITO_E02Q01;
```

Para a simulação do circuito, foram realizados os seguintes procedimentos:

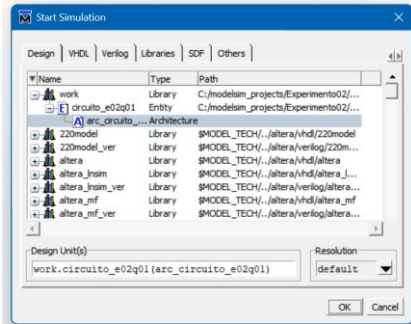


Inicialmente a compilação do código VHDL;

Não precisa explicar como usar o ModelSim

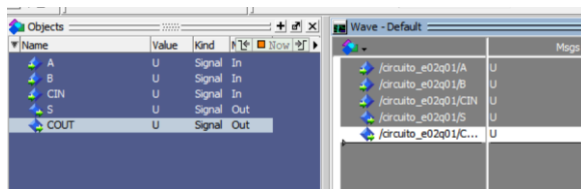


Em seguida selecionei a opção “Start Simulation...” na aba “Simulate”



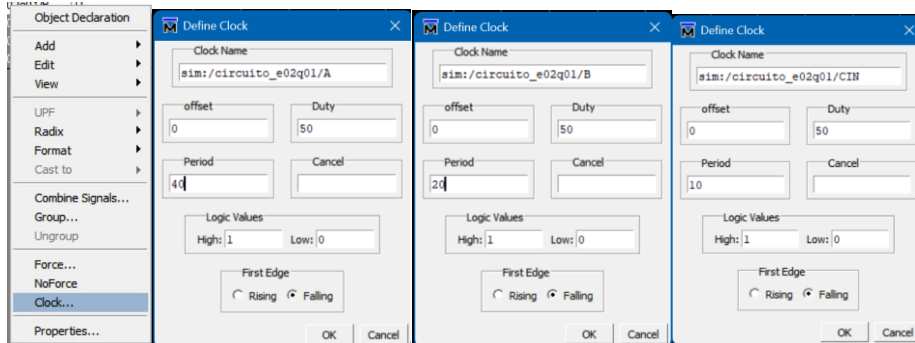
essas imagens que não são separadas do texto ficam estranhas

Para a simulação, selecionei a arquitetura ARC_CIRCUITO_E02Q01 descrita no código VHDL acima

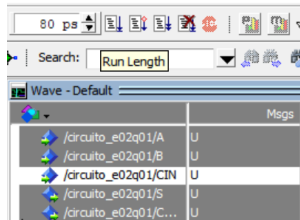


Repassei as portas lógicas de entrada e saída

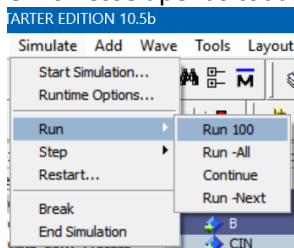
da aba “Objects” para a aba “Wave”



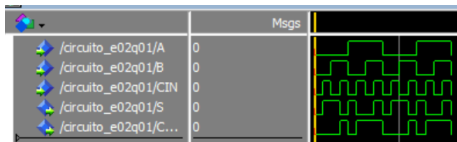
O sinal das portas lógicas A, B e C_{in} foi configurado “Clock” com a opção “First Edge – Falling” e um período de 40 para a porta A, 20 para a porta B e 10 para porta C. Isto foi feito para que a forma de onda da simulação se aproximasse da tabela verdade.



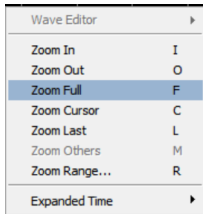
A “Run Length” foi ajustada para 80, para que a simulação então envolvesse apenas todas as combinações de entrada para as portas A,B e C_{in}



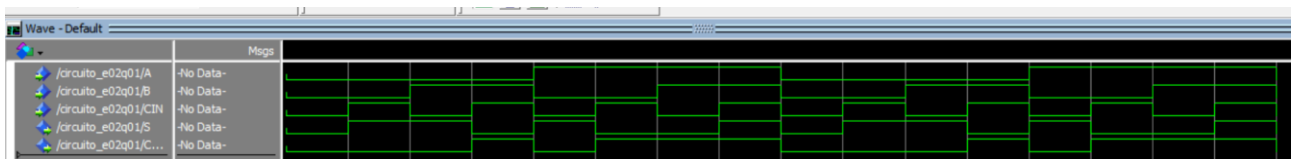
Executei então a simulação com a opção “Simulate->Run->Run 100”



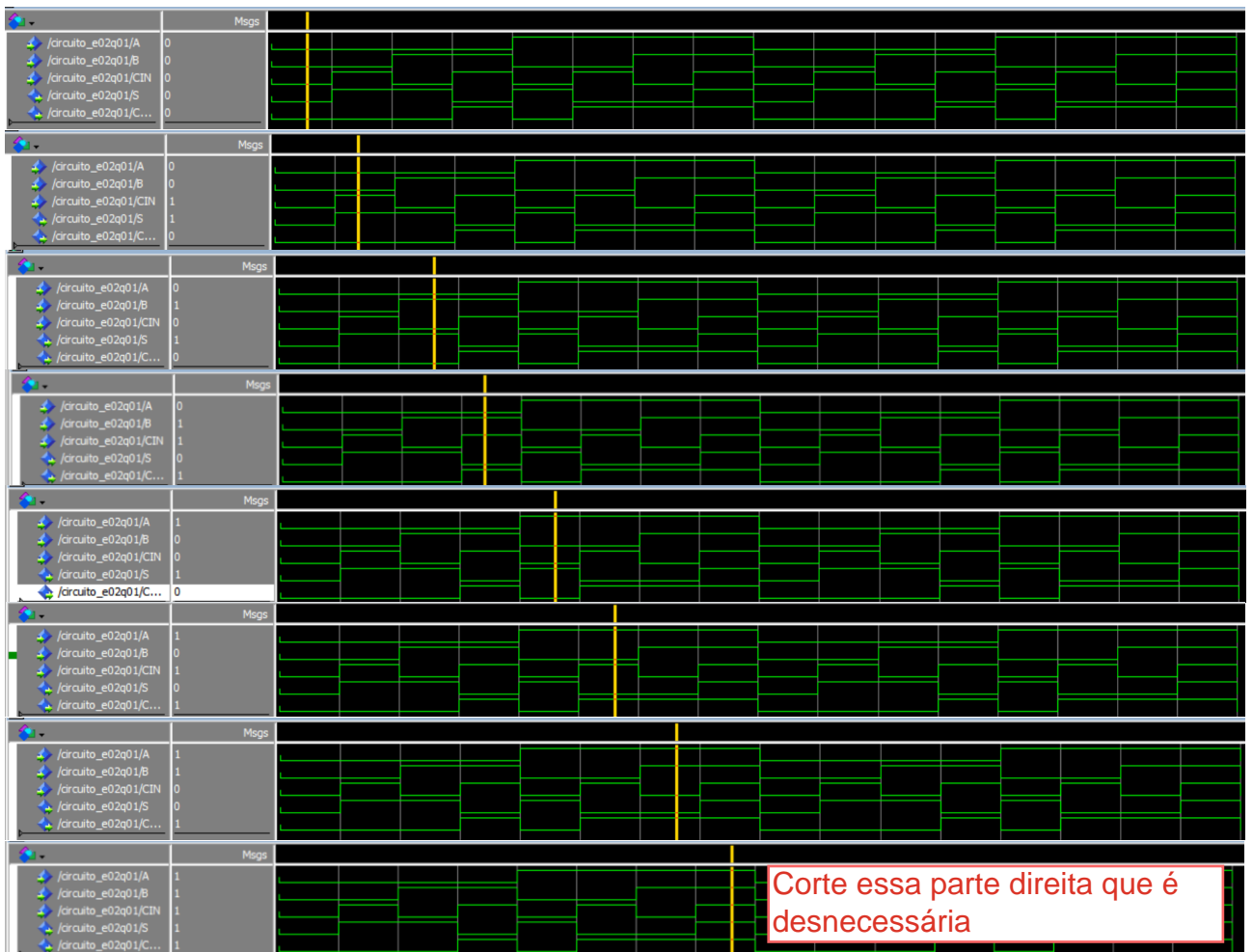
Este foi o resultado da simulação;



Porém o resultado pode ser melhor visualizado com a opção “Zoom Full”



O sinal das entradas e saídas pode ser obtido clicando no gráfico de onda – Na coluna “Msgs” temos o sinal da respectiva porta para o período (linha amarela) selecionado:



Corte essa parte direita que é desnecessária

Coloque títulos e legendas nas imagens e tabelas

Faça a tabela-verdade a partir da função lógica e faça a análise detalhada mostrando que os resultados da simulação reproduzem a tabel-verdade

Montando a tabela-verdade, temos então que:

A	B	C _{in}	S	C _{out}
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

Experimento 02 – Multiplexador de 4 para 1

O multiplexador de 4 para 1 é descrito pela função lógica abaixo:

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

O multiplexador foi representado no Logisim pelo código abaixo:

```
-- Experimento 02 - Questão 02

-- Aluno: Pedro Henrique Dias Avelar 241037112
-- Turma 08
-- Data: 02/11/2024

-- Função lógica do multiplexador 4 para 1:
-- Y = D0!S1!S0 + D1!S1S0 + D2S1!S0 + D3S1S0

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

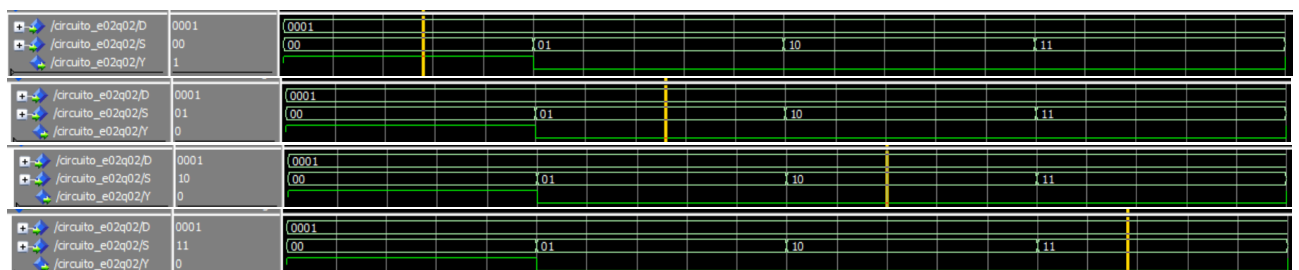
ENTITY CIRCUITO_E02Q02 IS
    PORT (D: IN STD_LOGIC_VECTOR (3 DOWNTO 0);
          S: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
          Y: OUT STD_LOGIC);
END CIRCUITO_E02Q02;

ARCHITECTURE ARC_CIRCUITO_E02Q02 OF CIRCUITO_E02Q02 IS
BEGIN
    Y <= (D(0) AND NOT(S(1)) AND NOT(S(0))) OR
          (D(1) AND NOT(S(1)) AND S(0)) OR
          (D(2) AND S(1) AND NOT(S(0))) OR
          (D(3) AND S(1) AND S(0));
END ARC_CIRCUITO_E02Q02;
```

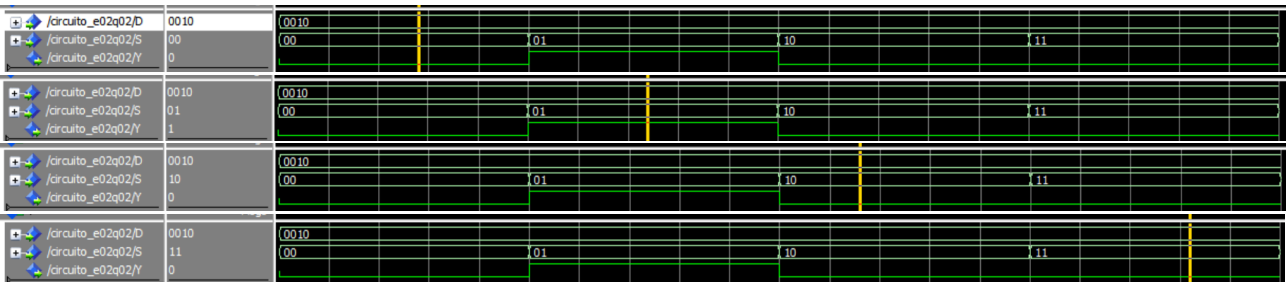
Explique os códigos

E a simulação foi então realizada para D = 0001, 0010, 0100 e 1000 e a seleção alternando entre 00, 01, 10 e 11:

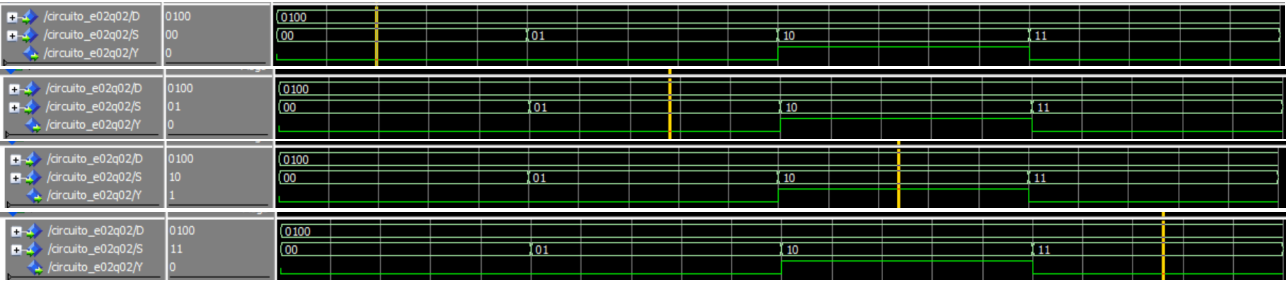
D=0001



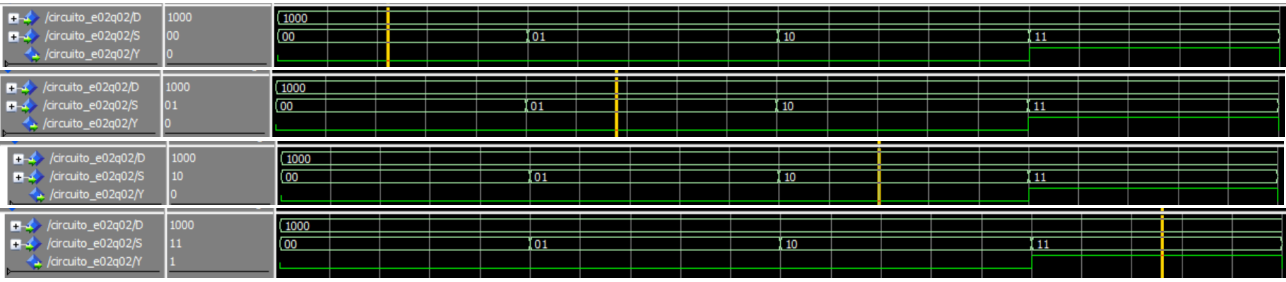
D=0010



D=0100



D=1000



Montando a tabela-verdade, temos então que:

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

É preciso explicar o que foi feito. Você praticamente só colocou prints dos códigos e das simulações

Nota final: 3,5 de 10