

INATEL - Instituto Nacional de Telecomunicações

E207 – Eletrônica Digital I

3º Período / 1º Semestre de 2021

Professores: Bruno de Oliveira Monteiro

bruno@inatel.br

Monitores: Felipe Pereira Silveira
Carlos Daniel Borges Vilela Marques
Gualter Machado Mesquita
Isabela Rezende Barbosa da Silva
Maíra Alves Chagas
Pedro Henrique Praxedes dos Reis
Thalita Fortes Domingos

felipecpereira@gea.inatel.br
carlos.marques@gea.inatel.br
machadomgualter@gmail.com
isabela.r@gec.inatel.br
mairaalves@gec.inatel.br
pedro.reis@gea.inatel.br
thalita.fortes@gec.inatel.br

RELATÓRIO 7

SOMADOR E SUBTRATOR EM VHDL

EXERCÍCIOS

Questão 1. Implemente o circuito meio somador e confirme a operação através da tabela da verdade. Preencha a tabela da verdade, retire a expressão lógica das saídas S e Ts e, com base nas expressões, desenhe o esquema elétrico de cada saída.

Para simular os valores obtidos, monte o esquema no software ISE.

A	B	S	Ts
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

```

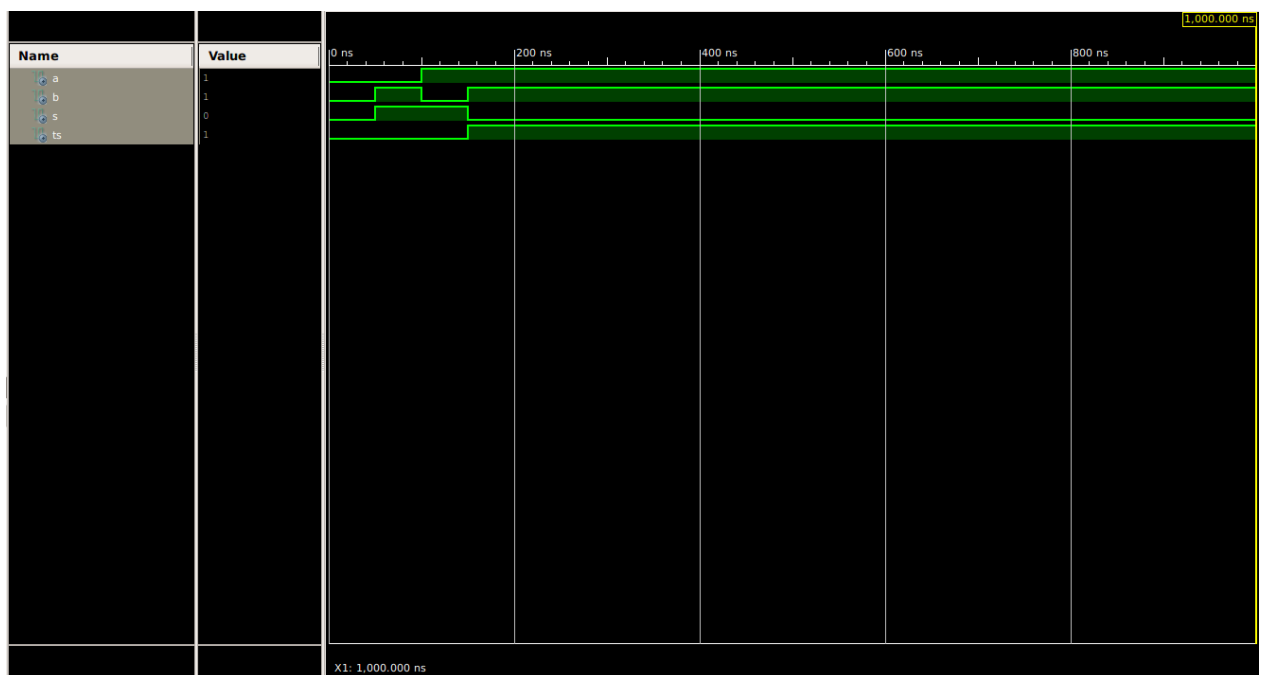
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity meioSomador_module is
6      Port ( A : in  STD_LOGIC;
7            B : in  STD_LOGIC;
8            S : out  STD_LOGIC;
9            Ts : out  STD_LOGIC);
10 end meioSomador_module;
11
12 architecture Behavioral of meioSomador_module is
13
14 begin
15
16     S <= A XOR B;
17     Ts <= A AND B;
18
19 end Behavioral;

```

```

45  -- Stimulus process
46  stim_proc: process
47  begin
48      A <= '0';
49      B <= '0';
50      wait for 50 ns;
51      A <= '0';
52      B <= '1';
53      wait for 50 ns;
54      A <= '1';
55      B <= '0';
56      wait for 50 ns;
57      A <= '1';
58      B <= '1';
59
60      wait;
61  end process;

```



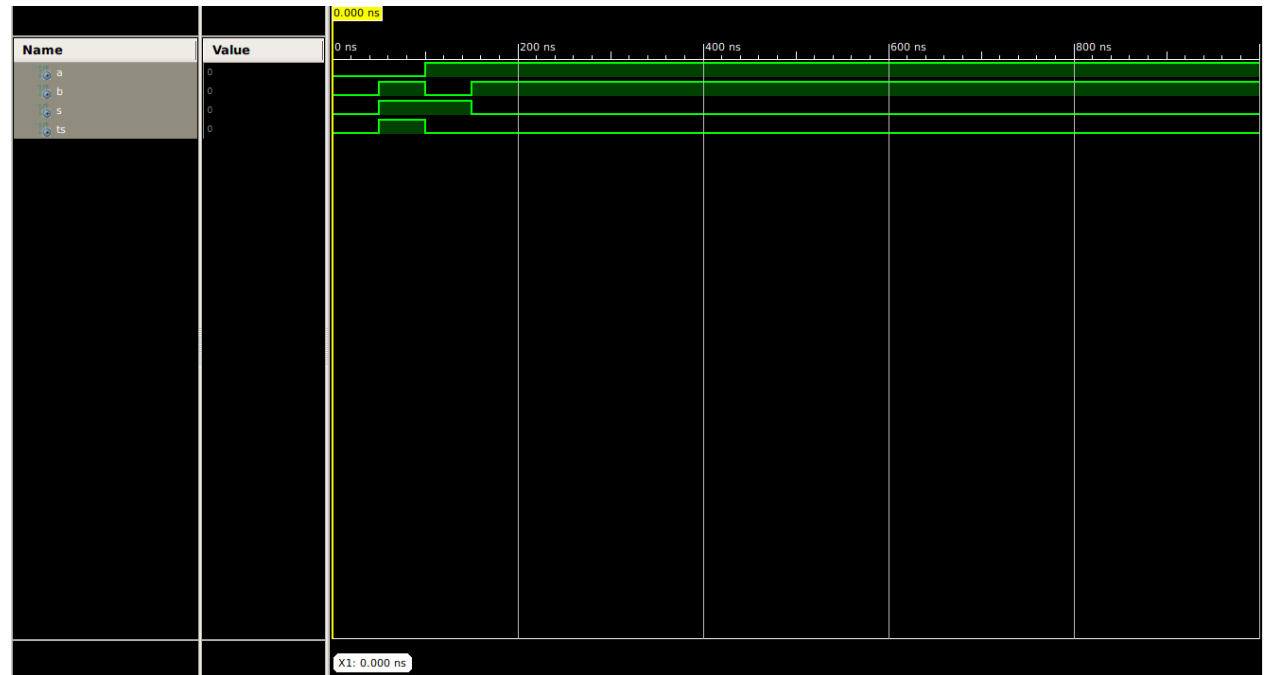
Questão 2. Agora, implemente um circuito subtrator e confirme a operação através da tabela da verdade. Preencha a tabela da verdade, retire a expressão lógica das saídas S e Ts e com base nas expressões desenhe o esquema elétrico de cada saída.

Para simular os valores obtidos, monte o esquema no software ISE

A	B	S	Ts
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

```
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4
5 entity meioSubtrator_module is
6     Port ( A : in  STD_LOGIC;
7           B : in  STD_LOGIC;
8           S : out STD_LOGIC;
9           Ts : out STD_LOGIC);
10 end meioSubtrator_module;
11
12 architecture Behavioral of meioSubtrator_module is
13 begin
14     S <= A XOR B;
15     Ts <= (NOT A) AND B;
16 end Behavioral;
```

```
44 -- Stimulus process
45 stim_proc: process
46 begin
47     A <= '0';
48     B <= '0';
49     wait for 50 ns;
50     A <= '0';
51     B <= '1';
52     wait for 50 ns;
53     A <= '1';
54     B <= '0';
55     wait for 50 ns;
56     A <= '1';
57     B <= '1';
58
59     wait;
60 end process;
```



Questão 3. Projeto do hardware em VHDL:



Projete um somador completo de duas entradas, cada entrada deve possuir uma posição binária e incrementa um pino de transporte de entrada (carry in).

Para as saídas serão necessários duas posições, uma que represente a posição inicial e outra que represente o transporte de saída (carry out).

Utilize exclusivamente as seguintes variáveis e monte a tabela da verdade nessa sequência:

- Somador de duas entradas: variáveis A e B.
- Transporte de entrada (carry in): variável C.
- Saídas: posição inicial e transporte de saída (carry out), respectivamente: variáveis S e T.

LEMBRETE: todo projeto deve conter a tabela da verdade, simplificação por **Mapa de Karnaugh** e o desenho do circuito elétrico. Depois de feito isso, simular o projeto no Proteus ISIS 7 e construir o hardware em VHDL no software ISE.

Entradas			Saídas	
A	B	C	S	T
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

Para S

S	B'C'	BC'	BC	B'C
A'	0	1	0	1
A	1	0	1	0

$$S = (A \oplus B) \oplus C$$

Para T

T	B'C'	BC'	BC	B'C
A'	0	0	1	0
A	0	1	1	1

$$T = AC + BC + AB$$

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4
5  entity somadorCompleto is
6      Port ( A : in  STD_LOGIC;
7            B : in  STD_LOGIC;
8            C : in  STD_LOGIC;
9            S : out  STD_LOGIC;
10           T : out  STD_LOGIC);
11 end somadorCompleto;
12
13 architecture Behavioral of somadorCompleto is
14
15 begin
16
17 S <= (A XOR B) XOR C;
18 T <= (A AND C) OR (B AND C) OR (A AND B);
19
20 end Behavioral;

```

```

48  -- Stimulus process
49  stim_proc: process
50  begin
51      A <= '0';
52      B <= '0';
53      C <= '0';
54      wait for 50 ns;
55      A <= '0';
56      B <= '0';
57      C <= '1';
58      wait for 50 ns;
59      A <= '0';
60      B <= '1';
61      C <= '0';
62      wait for 50 ns;
63      A <= '0';
64      B <= '1';
65      C <= '1';
66      wait for 50 ns;
67      A <= '1';
68      B <= '0';
69      C <= '0';
70      wait for 50 ns;
71      A <= '1';
72      B <= '0';
73      C <= '1';
74      wait for 50 ns;
75      A <= '1';
76      B <= '1';
77      C <= '0';
78      wait for 50 ns;
79      A <= '1';
80      B <= '1';
81      C <= '1';
82
83      wait;
84  end process;

```

