



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA - CT

CIRCUITOS DIGITAIS

PORTAS LÓGICAS

DCA0212.1 - Laboratório 1

Igor Michael Araujo de Macedo

Isaac de Lyra Júnior

Paulo Costa Braga

Pedro Henrique de Freitas Silva

Natal, 12 de outubro de 2021

1. INTRODUÇÃO

O presente relatório tem por objetivo desenvolver os conceitos teóricos de circuitos digitais de forma prática. Neste trabalho serão abordados os conceitos de portas lógicas, circuitos lógicos, funções booleanas, dentre outros. Para isso, utilizaremos o *software* ModelSim, onde serão praticados os conceitos com a linguagem de descrição de *hardware* VHDL.

2. DESENVOLVIMENTO

Para colocar em prática os conceitos iniciais estudados da álgebra booleana aplicando no VHDL, primeiramente serão feitos dois exercícios para simular uma porta AND e uma OR, e, em seguida, serão testados postulados booleanos, propriedades e leis De Morgan.

Os postulados de identidade para as operações de Adição (OR/+), Multiplicação (AND/·) e Negação (NOT/') são os seguintes:

- Identidade

1. $A + 0 = A$

2. $A \cdot 1 = A$

3. $(A')' = A$

As propriedades das operações booleanas são:

- Comutatividade:

1. Adição: $A + B = B + A$

2. Multiplicação: $A \cdot B = B \cdot A$

- Associatividade:

1. Adição: $A + (B + C) = (A + B) + C$

2. Multiplicação: $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

- Distributividade:

1. $A \cdot (B + C) = A \cdot B + A \cdot C$

2. $A + (B \cdot C) = (A + B) \cdot (A + C)$

E as leis de Morgan são:

- 1ª lei de Morgan:

- $(A \cdot B)' = A' + B'$

- 2ª lei de Morgan:

- $(A + B)' = A' \cdot B'$

3. RESULTADOS

Os resultados dos experimentos supracitados na seção 2 foram obtidos utilizando o *software* ModelSim.

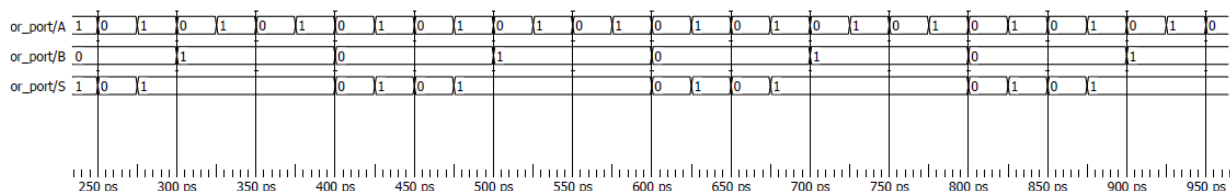
No decorrer deste experimento, desenvolvemos alguns circuitos lógicos para simular as portas OR, AND e também para verificar as propriedades booleanas, postulados e as leis de De Morgan.

A começar pelo desenvolvimento da porta lógica OR, utilizamos uma entidade que recebe duas entradas lógicas e apenas uma saída. Feito isso, desenvolvemos a arquitetura que esta entidade terá, ou seja, implementamos o resultado que esta saída devolverá de acordo com as entradas inseridas. Dessa forma, a nossa saída (S) é tida como uma equação lógicas definida por:

$$S = A \text{ or } B$$

Para validar o funcionamento deste circuito lógico, utilizamos a função de formato de onda e forçamos algumas entradas como podemos observar no Anexo I. O resultado do circuito lógico pode ser visto na Figura 1.

Figura 1: Simulação do funcionamento da porta lógica OR.



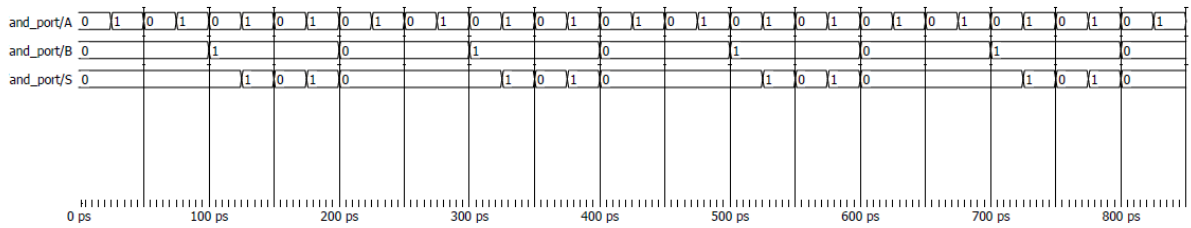
Fonte: Elaboração própria

De forma análoga ao desenvolvimento da porta lógica OR, utilizamos uma entidade que recebe duas entradas lógicas e apenas uma saída. Feito isso, desenvolvemos a arquitetura que esta entidade terá, ou seja, implementamos o resultado que esta saída devolverá de acordo com as entradas inseridas. Dessa forma, a nossa saída (S) é tida como uma equação lógicas definida por:

$$S = A \text{ and } B$$

Para validar o funcionamento deste circuito lógico, utilizamos a função de formato de onda e forçamos algumas entradas como podemos observar no Anexo I. O resultado do circuito lógico pode ser visto na Figura 2.

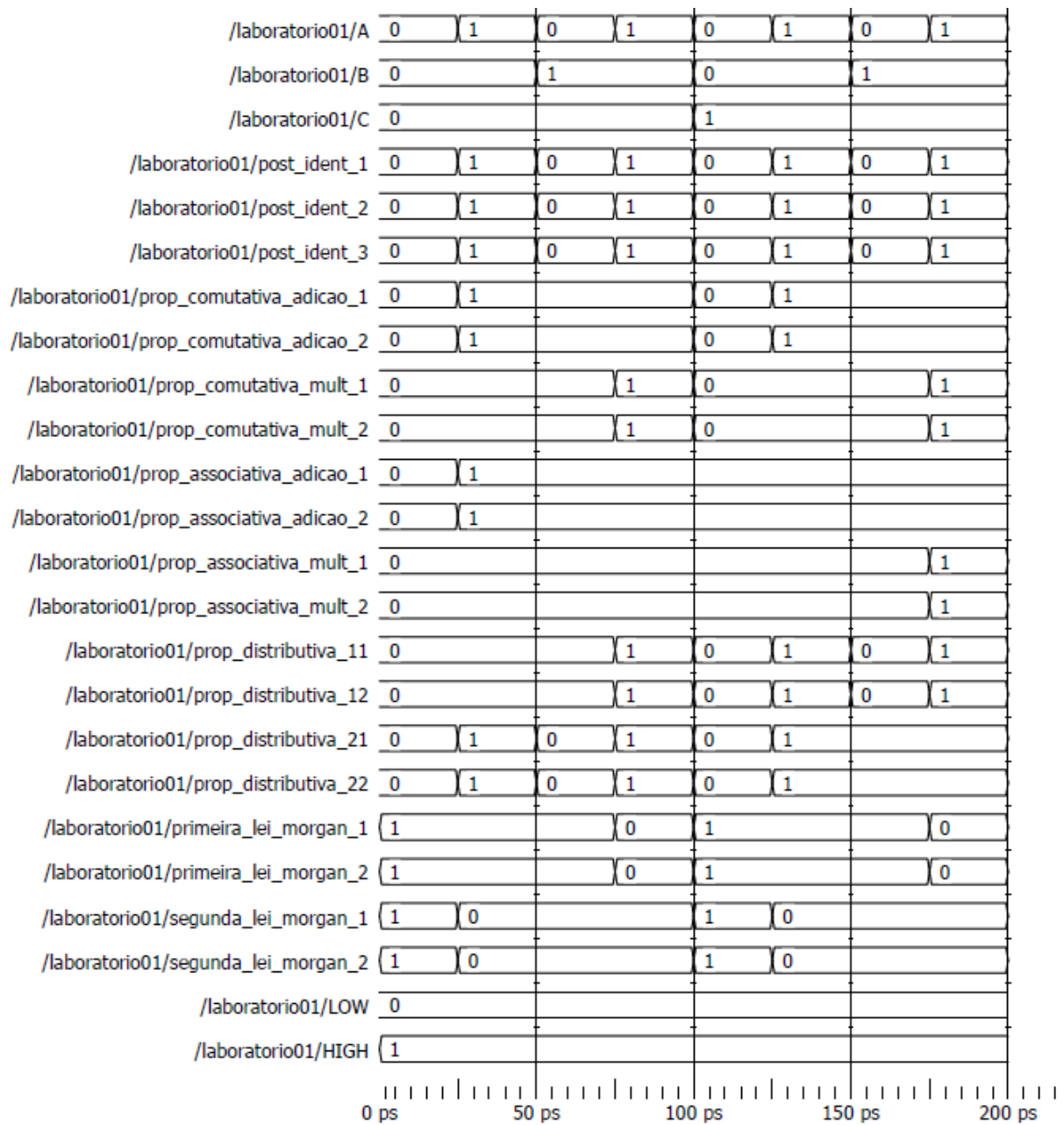
Figura 2: Simulação do funcionamento da porta lógica AND.



Fonte: Elaboração própria

Para realizar os testes de todas as propriedades, postulados e a Lei de Morgan foi seguida a dica da atividade, onde foi criada uma entidade nova com três entradas e várias saídas, sendo cada saída definida como uma das equações vistas na seção 2. O resultado pode ser visto na Figura 3, onde foi possível comprovar todas as propriedades, postulados e a Lei de Morgan de forma experimental. As saídas de nomes iguais mas terminadas em 1 e 2 dizem respeito aos lados esquerdos e direitos das equações, a fim de provar a validade delas. Os códigos utilizados para a simulação podem ser vistos no Anexo III.

Figura 3: Simulação das propriedades, postulados e a Lei de Morgan.



Fonte: Elaboração própria

4. CONCLUSÃO

O presente trabalho tem por fim introduzir sobre o mundo da eletrônica digital, onde primeiramente foram apresentadas as portas lógicas *AND* e *OR*. Em seguida, ainda sobre os circuitos digitais, foram expostos os postulados, propriedades e a lei De Morgan. Para comprovar todas as propriedades foi utilizado o *software* Modelsim para escrever o código em VHDL e fazer simulações.

ANEXOS

ANEXO I - PORTA LÓGICA OR

Código em VHDL:

```
entity OR_PORT is
    port (
        A, B: in bit;
        S: out bit
    );
end OR_PORT;

architecture logic of OR_PORT is
    begin
        S <= A or B;
    end logic;
```

Código para a simulação:

```
vsim OR_PORT
add wave *
force A 0 0, 1 25 -repeat 50
force B 0 0, 1 100 -repeat 200
run 1000
```

ANEXO II - PORTA LÓGICA AND

Código em VHDL:

```
entity AND_PORT is
    port (
        A, B: in bit;
        S: out bit
    );
end AND_PORT;

architecture logic of AND_PORT is
begin
    S <= A and B;
end logic;
```

Código para a simulação:

```
vsim AND_PORT
add wave *
force A 0 0, 1 25 -repeat 50
force B 0 0, 1 100 -repeat 200
run 1000
```

ANEXO III - POSTULADOS, PROPRIEDADES E LEIS DE MORGAN

Código em VHDL:

```
entity laboratorio01 is
  port (
    A, B, C: in bit;
    post_ident_1, post_ident_2, post_ident_3: out bit;
    prop_comutativa_adicao_1, prop_comutativa_adicao_2: out bit;
    prop_comutativa_mult_1, prop_comutativa_mult_2: out bit;
    prop_associativa_adicao_1, prop_associativa_adicao_2: out bit;
    prop_associativa_mult_1, prop_associativa_mult_2: out bit;
    prop_distributiva_11, prop_distributiva_12: out bit;
    prop_distributiva_21, prop_distributiva_22: out bit;
    primeira_lei_morgan_1: out bit;
    primeira_lei_morgan_2: out bit;
    segunda_lei_morgan_1: out bit;
    segunda_lei_morgan_2: out bit
  );
end laboratorio01;

architecture main of laboratorio01 is
  signal LOW, HIGH: bit;

begin
  -- Auxiliares
  LOW <= '0';
  HIGH <= '1';

  -- Postulados - identidade
  post_ident_1 <= A or LOW;
  post_ident_2 <= A and HIGH;
  post_ident_3 <= not (not A);

  -- Propriedade comutativa
  prop_comutativa_adicao_1 <= A or B;
  prop_comutativa_adicao_2 <= B or A;

  prop_comutativa_mult_1 <= A and B;
  prop_comutativa_mult_2 <= B and A;
```

```

-- Propriedade associativa
prop_associativa_adicao_1 <= A or (B or C);
prop_associativa_adicao_2 <= (A or B) or C;

prop_associativa_mult_1 <= A and (B and C);
prop_associativa_mult_2 <= (A and B) and C;

-- Propriedade distributiva
prop_distributiva_11 <= A and (B or C);
prop_distributiva_12 <= (A and B) or (A and C);

prop_distributiva_21 <= A or (B and C);
prop_distributiva_22 <= (A or B) and (A or C);

-- Primeira lei de Morgan
primeira_lei_morgan_1 <= not (A and B);
primeira_lei_morgan_2 <= not (A) or not (B);

-- Segunda lei de Morgan
segunda_lei_morgan_1 <= not (A or B);
segunda_lei_morgan_2 <= (not A) and (not B);

end main;

```

Código para simulação:

```

vsim laboratorio01
add wave *
force A 0 0, 1 25 -repeat 50
force B 0 0, 1 50 -repeat 100
force C 0 0, 1 100
run 200

```