

LABORATÓRIO 1 – PORTAS LÓGICAS

EFRAIN MARCELO PULGAR PANTALEON¹
FERNANDO LUCAS SOUSA SILVA²
MATHEUS GOMES DINIZ ANDRADE³
TEÓFILO VITOR DE CARVALHO CLEMENTE⁴

Universidade Federal do Rio Grande do Norte, Departamento de Engenharia de Computação e Automação

1. INTRODUÇÃO

O presente relatório tem por objetivo analisar o funcionamento das portas lógicas e consequentemente explorar as possibilidades de funcionamento que elas podem trazer quando usadas, sozinhas ou combinadas. Assim, apresentaremos os resultados obtidos nas simulações em etapas, de modo a mostrar os códigos usados para a emulação das portas lógicas e respectivos resultados adquiridos via software.

2. METODOLOGIA

Para a realização deste laboratório foi necessário o conhecimento teórico a respeito do funcionamento das portas lógicas e quais suas finalidades práticas. Desse modo, para a experimentação prática foi utilizado o software Quartus II, ele possibilita a construção de diversos circuitos digitais a partir do uso de portas lógicas, flip-flops e outros componentes que podem ser simulados com seu uso, com isso foi possível a construção dos códigos em VHDL e posterior aplicação como será mostrado nas seções a seguir.

3. RESULTADOS

Nesta seção serão apresentados os resultados obtidos a cada simulação realizada, como também os meios utilizados para as obter no software.

3.1. PORTA OR

A porta OR como já vista na parte teórica do conteúdo funciona a partir do recebimento de sinais digitais, 1 ou 0, mediante o recebimento desses sinais ela irá ter como resposta 1 caso pelo menos uma das entradas seja de nível alto (1) e será baixa (0) somente quando todas as entradas na porta forem baixas.

Então já no simulador foi criado um projeto com o nome da porta a ser simulada, então com o uso da linguagem VHDL vista de forma teórica foi possível a construção do código que veremos na imagem a seguir.

```
1  entity or_gate is
2  port(
3      a, b : in bit;
4      z    : out bit
5  );
6  end or_gate;
7
8  architecture main of or_gate is
9  begin
10     z <= a or b;
11
12 end architecture main;
```

Figura 1 – Código do protótipo da porta OR.

A partir da validação do código foi feita a compilação para a respectiva simulação onde submetemos valores lógicos e com isso obtemos os resultados computados pelo software e verificamos o funcionamento da porta projetada como a seguir.

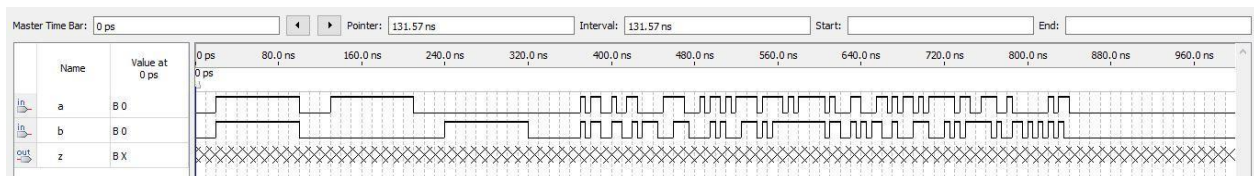


Figura 2 – Entradas lógicas usadas para a simulação.

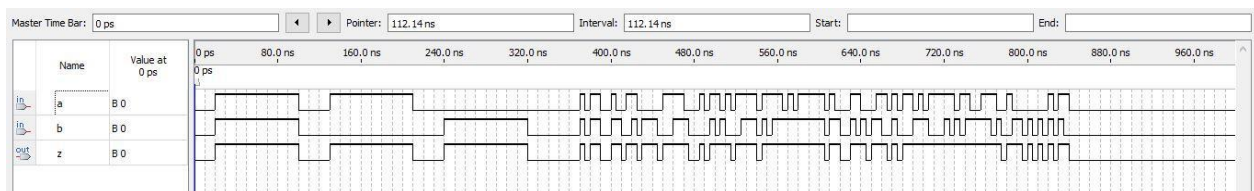


Figura 3 – Resultado da simulação para a porta OR.

3.2. PORTA AND

A porta AND semelhante a anterior funciona a partir do recebimento de uns sinais digitais, 1 ou 0, a partir de tal ela irá diferir em como responderá os sinais, a saída dela será de nível alto somente quando todas as entradas também forem altas, nas demais possibilidades ela será de nível baixo.

No simulador foi criado outro projeto com o nome da porta, então foi executada a construção do código que simula a porta e foi obtido o mostrado a seguir.

```
1  entity and_gate is
2  port(
3      a, b : in bit;
4      z    : out bit
5  );
6  end and_gate;
7
8  architecture main of and_gate is
9
10 begin
11
12     z <= a and b;
13
14 end architecture main;
```

Figura 4 – Código do protótipo da porta AND.

Posterior a isso foi submetido a simulação onde foram dados valores lógicos nas entradas da porta e com isso temos a resposta do software que foi correspondente ao funcionamento da porta AND atentando o funcionamento do código simulado como veremos a seguir.

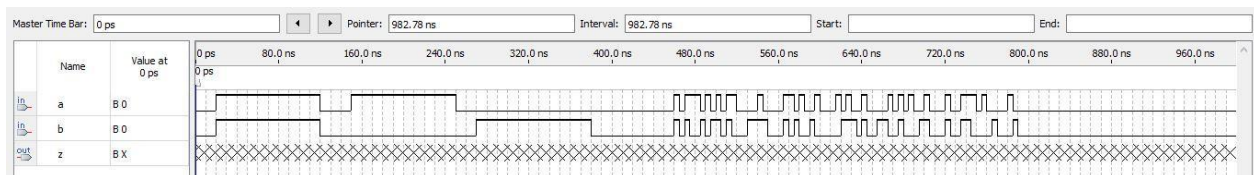


Figura 5 – Entradas lógicas usadas para a simulação.

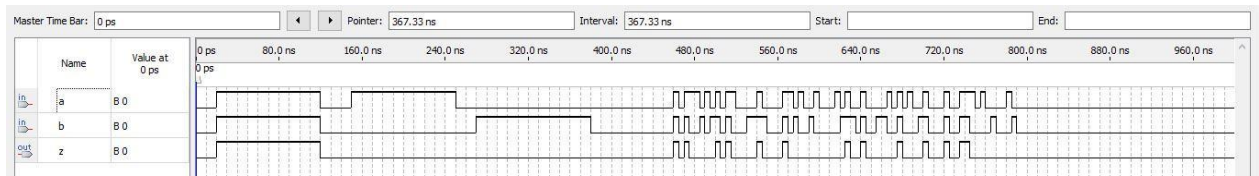


Figura 6 – Resultado da simulação para a porta AND.

3.3. TESTE DAS PROPRIEDADES E POSTULADOS

Para este tópico foi pedido o teste e com isso a validação de todas as propriedades das portas lógicas vistas na parte teórica. Para a realização desta parte foram criadas diversas saídas lógicas para validação de todos os testes pedidos, de modo que cada propriedade a ser simulada é independente como comentado no código e assim teremos uma simulação completa do pedido no roteiro.

```
1 entity all_properties is
2   port(
3     a, b, c           : in bit;
4     id1, id2, id3     : out bit;
5     ac1, ac2, mc1, mc2 : out bit;
6     aa1, aa2, ma1, ma2 : out bit;
7     md1, md2, ad1, ad2 : out bit;
8     lm1_1, lm1_2, lm2_1, lm2_2 : out bit
9   );
10 end all_properties;
11
12 architecture main of all_properties is
13
14 begin
15   -- Identidade
16   id1 <= a or '0';
17   id2 <= a and '1';
18   id3 <= not(not(a));
19
20   -- Adicao comutativa
21   ac1 <= a or b;
22   ac2 <= b or a;
23   -- Multiplicacao comutativa
24   mc1 <= a and b;
25   mc2 <= b and a;
26
27   -- Adicao associativa
28   aa1 <= a or (b or c);
29   aa2 <= (a or b) or c;
30   -- Multiplicacao associativa
31   ma1 <= a and (b and c);
32   ma2 <= (a and b) and c;
33
34   -- Multiplicacao distributiva
35   md1 <= a and (b or c);
36   md2 <= (a and b) or (a and c);
37   -- Adicao distributiva
38   ad1 <= a or (b and c);
39   ad2 <= (a or b) and (a or c);
40
41   -- 1 Lei de Morgan
42   lm1_1 <= not(a and b);
43   lm1_2 <= not(a) or not(b);
44   -- 2 Lei de Morgan
45   lm2_1 <= not(a or b);
46   lm2_2 <= not(a) and not(b);
47
48
49 end architecture main;
```

Figura 7 – Código para simulação das propriedades.

Com a idealização do código foi realizada a compilação para verificar se não havia algum erro já que este era maior que os códigos anteriores e mais complexo. Após isso foi dado início a simulação onde temos as entradas a, b e c a serem usadas e respectivamente as saídas que vão assegurar a validade das propriedades testadas como veremos a seguir.

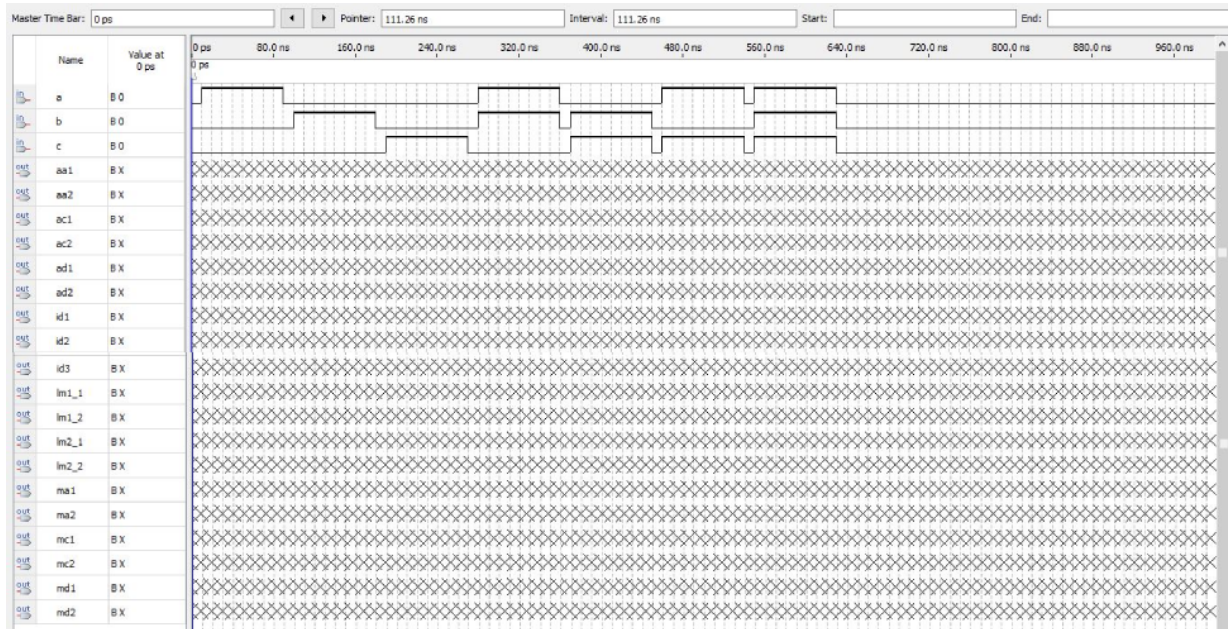


Figura 8 – Entradas lógicas usadas para a simulação.

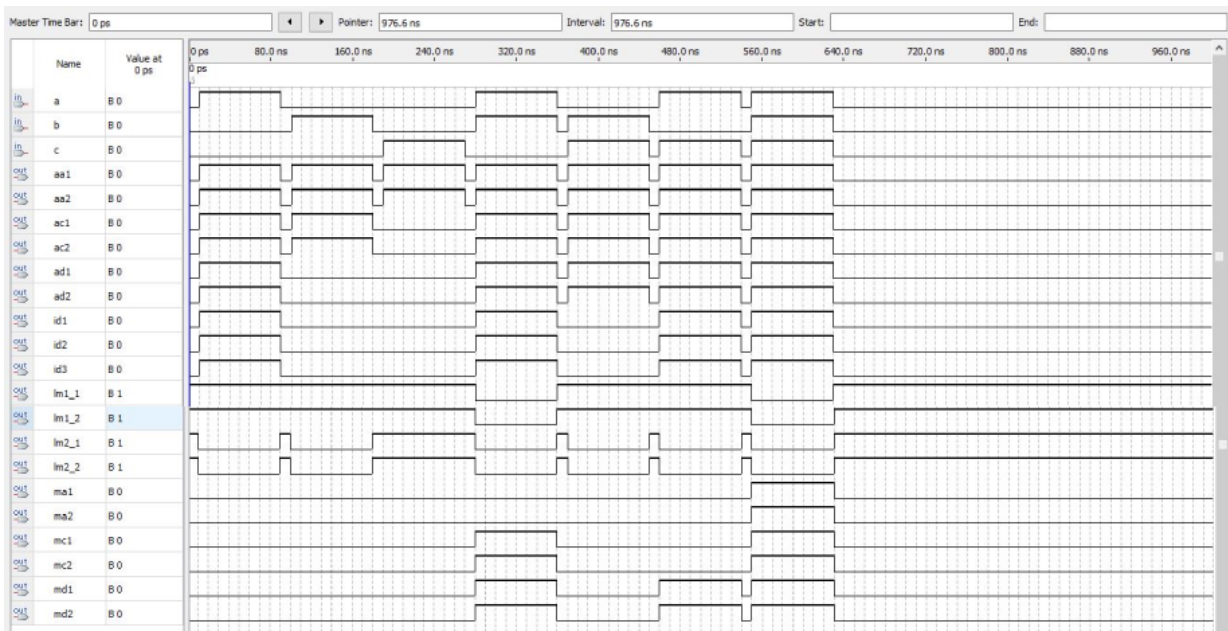


Figura 9 – Resultado da simulação para as propriedades.

4. CONCLUSÕES

A realização deste laboratório permitiu o melhor entendimento a respeito dos assuntos vistos de forma teórica. Portanto, foi possível comprovar o funcionamento das portas lógicas e suas propriedades com a aplicação prática no simulador utilizando códigos em VHDL.

5. REFERÊNCIAS

[1] QUARTUS II. Software de simulação.

[2]Vahid, Frank. Digital Design with RTL Design, VHDL, and Verilog Solution Manual

. 2º Edição.2010.