Universidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Departamento de Engenharia Elétrica - DEE

Nome: Alysson Machado de Oliveira Barbosa
Email: alysson.barbosa@ee.ufcg.edu.br

Disciplina: Laboratório de Circuitos Lógicos
Professora: Fernanda Cecília Correia Lima Loureiro

Experimento 04 - Introdução ao Quartus

Objetivo 1

Apresentar a forma de descrever a conexão de portas de entradas e portas de saída de um módulo.

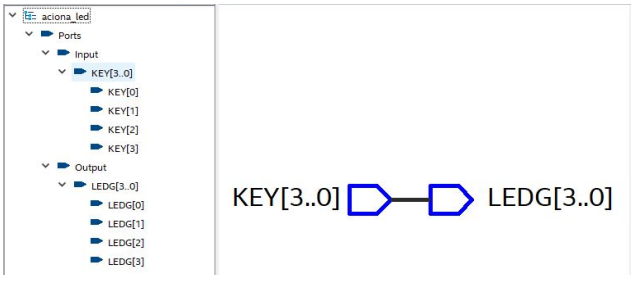
- Definindo botões de 0 a 3 conectados aos Led's verdes de 0 a 3. Os botões são referenciados no Quartus II pela palavra KEY; portanto a primeira linha será: assign LEDG[0] = KEY[0]. As outras 3 linhas serão semelhantes, mudando apenas o índice do LEDG e do KEY.

Figura 01 - Implementação do código em verilog do experimento 1-a

```
1   module aciona_led
2   (
3       input [3:0]KEY,
4       output [3:0]LEDG
5   );
6
7       assign LEDG[0] = KEY[0];
8       assign LEDG[1] = KEY[1];
9       assign LEDG[2] = KEY[2];
10      assign LEDG[3] = KEY[3];
11
12      // assign LEDG[3:0] = KEY[3:0];
13
14  endmodule
```

Fonte: Imagem Autoral

Figura 02 - Diagrama lógico obtido pelo Quartus relativo ao experimento 1-a



Fonte: Imagem Autoral

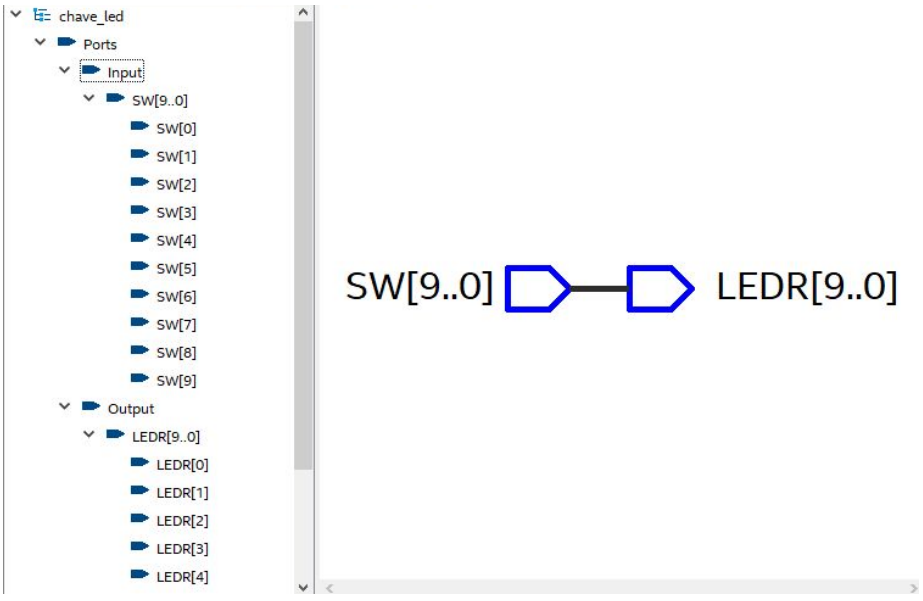
- Definindo chaves de 0 a 10, conectadas aos Led's vermelhos (Red) de 0 a 10. As chaves são referenciadas com SW, portanto a primeira linha será: assign LEDR[0] = SW[0]. As outras 10 linhas irão mudar apenas os índices do LEDR e do SW.

Figura 03 - Implementação do código em verilog do experimento 1-b

```
1   module chave_led
2   (
3       input [9:0]SW,
4       output [9:0]LEDR
5   );
6
7       assign LEDR[0] = SW[0];
8       assign LEDR[1] = SW[1];
9       assign LEDR[2] = SW[2];
10      assign LEDR[3] = SW[3];
11      assign LEDR[4] = SW[4];
12      assign LEDR[5] = SW[5];
13      assign LEDR[6] = SW[6];
14      assign LEDR[7] = SW[7];
15      assign LEDR[8] = SW[8];
16      assign LEDR[9] = SW[9];
17
18      // assign LEDR[9:0] = SW[9:0];
19
20  endmodule
```

Fonte: Imagem Autoral

Figura 04 - Diagrama lógico obtido pelo Quartus relativo ao experimento 1-b



Fonte: Imagem Autoral

Objetivo 2

Sabendo que os operadores lógicos para a porta AND é o & , OR é o | , e para inverter o resultado da operação é utilizado ~, foi elaborado os módulos para as portas AND, NAND, OR e NOR por tabela-verdade. Além disso, será verificado os diagramas RTL gerados pelo Quartus II para vários níveis de hierarquia.

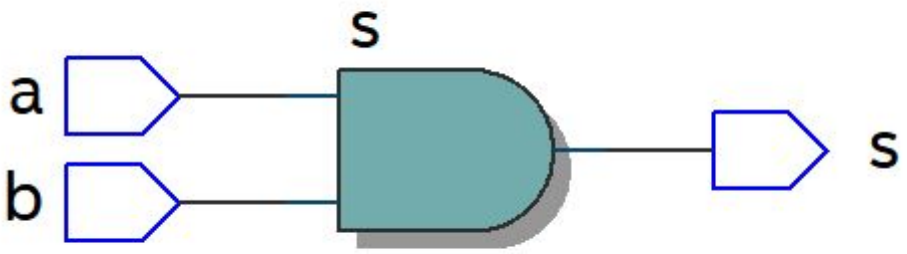
- Porta AND

Figura 05 - Implementação do código em verilog do experimento 2-a

```
1   module porta_and
2   (
3       input a, b,
4       output s
5   );
6
7       assign s = a & b;
8
9   endmodule
```

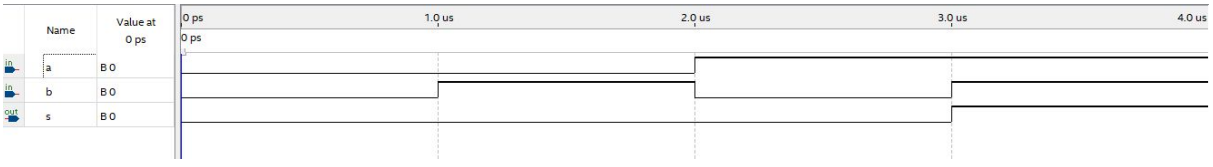
Fonte: Imagem Autoral

Figura 06 - Diagrama lógico obtido pelo Quartus relativo ao experimento 2-a



Fonte: Imagem Autoral

Figura 07 - Diagrama de tempos obtido pelo Quartus relativo ao experimento 2-a



Fonte: Imagem Autoral

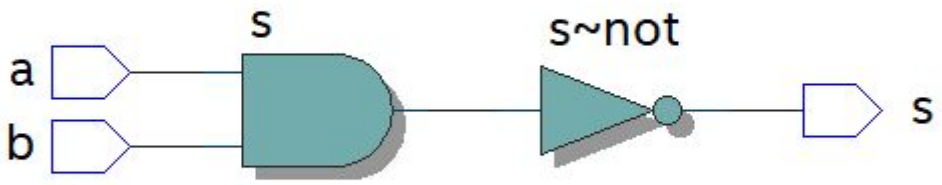
- Porta NAND

Figura 08 - Implementação do código em verilog do experimento 2-b

```
1   module porta_nand
2   (
3       input a, b,
4       output s
5   );
6
7       assign s = ~(a & b);
8
9   endmodule
10
11
```

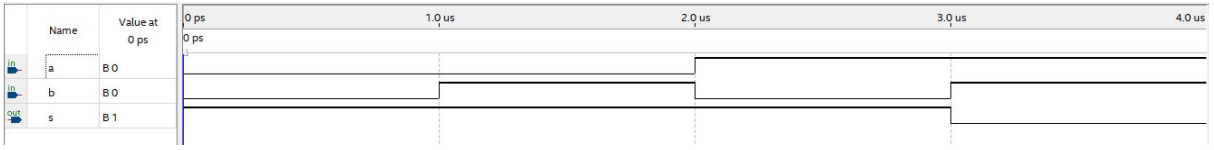
Fonte: Imagem Autoral

Figura 09 - Diagrama lógico obtido pelo Quartus relativo ao experimento 2-b



Fonte: Imagem Autoral

Figura 10 - Diagrama de tempos obtido pelo Quartus relativo ao experimento 2-b



Fonte: Imagem Autoral

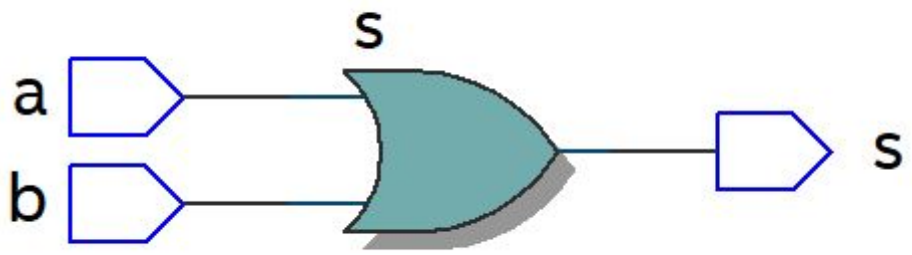
- Porta OR

Figura 11 - Implementação do código em verilog do experimento 2-c

```
1   module porta_or
2   (
3       input a, b,
4       output s
5   );
6
7       assign s = a | b;
8
9   endmodule
```

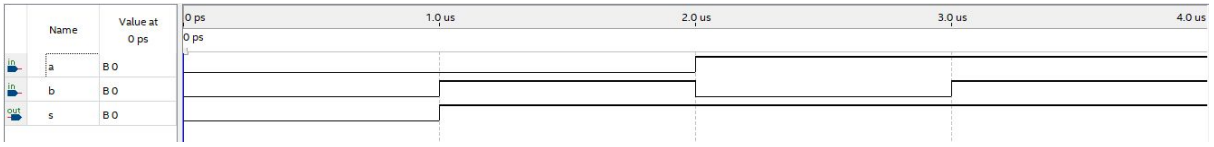
Fonte: Imagem Autoral

Figura 12 - Diagrama lógico obtido pelo Quartus relativo ao experimento 2-c



Fonte: Imagem Autoral

Figura 13 - Diagrama de tempos obtido pelo Quartus relativo ao experimento 2-c



Fonte: Imagem Autoral

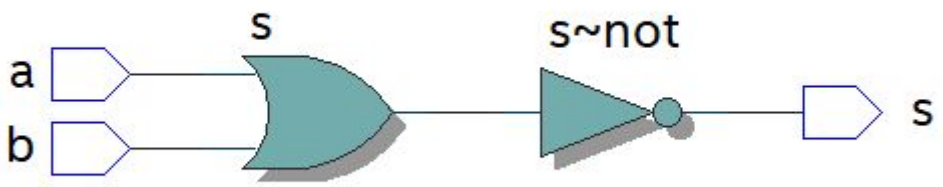
- Porta NOR

Figura 14 - Implementação do código em verilog do experimento 2-d

```
1   module porta_nor
2   (
3       input a, b,
4       output s
5   );
6
7       assign s = ~(a | b);
8
9   endmodule
```

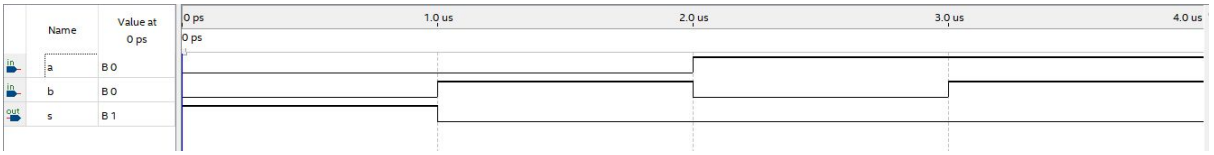
Fonte: Imagem Autoral

Figura 15 - Diagrama lógico obtido pelo Quartus relativo ao experimento 2-d



Fonte: Imagem Autoral

Figura 16 - Diagrama de tempos obtido pelo Quartus relativo ao experimento 2-d



Fonte: Imagem Autoral

Objetivo 3

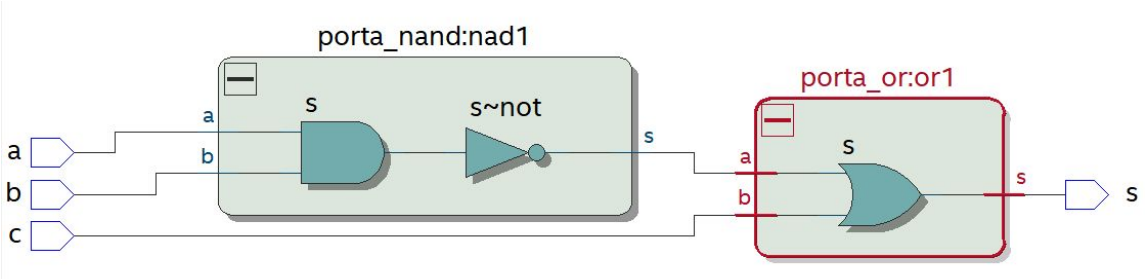
Utilizando o conceito de hierarquia, foi realizado a implementação em Verilog de um circuito lógico com 2 variáveis de entrada que terá saída igual 1 sempre que a quantidade de indivíduos no ambiente seja menor que 2, essa saída será comparada com uma outra entrada e terão saídas iguais a 1 sempre os valores de entrada não forem iguais a 0.

Figura 17 - Implementação do código em verilog do experimento 3

```
1   module verifica_individuos
2   (
3       input a, b, c,
4       output s
5   );
6
7       wire ab;
8
9       porta_nand nad1(a, b, ab);
10
11      wire ec;
12
13      porta_or or1(ab, c, ec);
14
15      assign s = ec;
16
17  endmodule
```

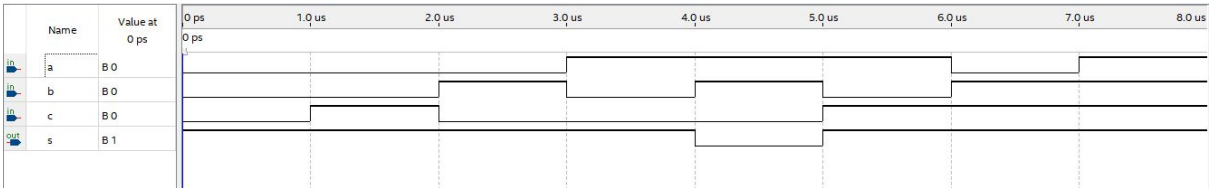
Fonte: Imagem Autoral

Figura 18 - Diagrama lógico obtido pelo Quartus relativo ao experimento 3



Fonte: Imagem Autoral

Figura 19 - Diagrama de tempos obtido pelo Quartus relativo ao experimento 3



Fonte: Imagem Autoral