

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
LABORATÓRIO DE CIRCUITOS LÓGICOS**

JOEL FELIPE FERREIRA GOMES

**RELATÓRIO DO TRABALHO DE GERENCIAMENTO DE VAGAS DE UM
ESTACIONAMENTO**

Natal/RN
2016

1. ENTITY estacionamento

Este componente tem as portas:

- SIGNAL *entrou, saiu*: UNSIGNED(0 to 0);
- SIGNAL *numero*: UNSIGNED (5 DOWNT0 0);
- SIGNAL *display1*: OUT STD_LOGIC_VECTOR(0 to 6);
- SIGNAL *display2*: OUT STD_LOGIC_VECTOR(0 to 6)).

E serve para converter o *numero* em uma saída de display de 7 segmentos representados por *display1* e *display2*.

1.1 COMPONENT mux_4x1

Este componente tem as portas:

- SIGNAL *a, b, c, d*: IN UNSIGNED(0 to 0);
- SIGNAL *sel1, sel2*: IN UNSIGNED(0 to 0);
- SIGNAL *s*: OUT UNSIGNED(0 to 0)

E serve para escolher uma das entradas: *a, b, c, d* de acordo com os seletores *sel1, sel2*.

1.2 COMPONENT bin_7seg

Este componente tem as portas:

- NUM_BINARIO : IN UNSIGNED(3 DOWNT0 0);
- A, B, C, D, E, F, G : OUT STD_LOGIC);

E serve para converter o NUM_BINARIO passado para um código de 7 segmentos que está representado abaixo:

- "0000001" = 0
- "1001111" = 1
- "0010010" = 2
- "0000110" = 3
- "1001100" = 4
- "0100100" = 5
- "0100000" = 6
- "0001111" = 7
- "0000000" = 8
- "0000100" = 9
- "1111111" = ERROR

2. SIGNALS

2.1 SIGNAL dezena

O sinal: SIGNAL *dezena*: UNSIGNED (3 DOWNT0 0); representa as dezenas do número a ser mostrado no display de 7 segmentos.

2.2 SIGNAL unidade

O sinal: SIGNAL *unidade*: UNSIGNED (3 DOWNTO 0); representa as unidades do número a ser mostrado no display de 7 segmentos.

2.3 SIGNAL entrou_saiu

O sinal: SIGNAL *entrou_saiu*: UNSIGNED(0 to 0); representa se entrou ou saiu carros do estacionamento.

2.4 SIGNAL sum

O sinal: SIGNAL *sum*: UNSIGNED(0 to 0); representa se entrou um carro e não saiu outro no estacionamento e o limite de 50 carros não foi atingido.

2.5 SIGNAL sub

O sinal: SIGNAL *sub*: UNSIGNED(0 to 0); representa se saiu um carro e não entrou outro no estacionamento e o limite de 0 carros não foi atingido.

2.6 SIGNAL igual_z

O sinal: SIGNAL *igual_z*: UNSIGNED(0 to 0); representa se o limite de 0 carros não foi atingido.

2.7 SIGNAL igual_c

O sinal: SIGNAL *igual_c*: UNSIGNED(0 to 0); representa se o limite de 0 carros não foi atingido.

2.8 SIGNAL num1

O sinal: SIGNAL *num1*: UNSIGNED(5 DOWNTO 0); representa o número de vagas no estacionamento depois de somar com o carro que saiu.

2.9 SIGNAL num2

O sinal: SIGNAL *num2*: UNSIGNED(5 DOWNTO 0); representa o número vagas no estacionamento depois de subtrair com o carro que entrou.

3. BEGIN

3.1 COMANDO 1

Utiliza o componente *mux_4x1* da seguinte maneira: **PORT MAP("0", "1", "1", "0", *entrou*, *saiu*, *entrou_saiu*);** e serve para retornar se entrou ou saiu algum carro.

3.2 COMANDO 2

Atribui a *igual_c* o valor de “1” se *numero* = “110010” ou o valor de “0” se não.

3.3 COMANDO 3

Atribui a *igual_z* o valor de “1” se *numero* = “000000” ou o valor de “0” se não.

3.4 COMANDO 4

Atribui a *sub* o valor de: *entrou AND entrou_saiu AND NOT igual_z*.

3.5 COMANDO 5

Atribui a *sum* o valor de: *saiu AND entrou_saiu AND NOT igual_c*.

3.6 COMANDO 6

Atribui a *num1* o valor de: *numero + sum*. A função + é definida na biblioteca *ieee.numeric_std.all* e retorna a soma dos dois números.

3.7 COMANDO 7

Atribui a *num1* o valor de: *numero - sum*. A função - é definida na biblioteca *ieee.numeric_std.all* e retorna a subtração dos dois números.

3.8 COMANDO 8

Atribui a *dezena* o valor de: *RESIZE((num1 / 10), 4)* se *sum* = “1” ou *RESIZE((num2 / 10), 4)* se *sub* = “1” ou *RESIZE((numero / 10), 4)* se nenhum dos anteriores. A função / é definida na biblioteca *ieee.numeric_std.all* e retorna o valor inteiro da divisão dos dois números. A função *RESIZE* é definida na biblioteca *ieee.numeric_std.all* e retorna, quando truncado, os bits mais à direita do vetor passado no parâmetro.

3.9 COMANDO 9

Atribui a *unidade* o valor de: *RESIZE((num1 MOD 10), 4)* se *sum* = “1” ou *RESIZE((num2 MOD 10), 4)* se *sub* = “1” ou *RESIZE((numero MOD 10), 4)* se nenhum dos anteriores. A função *MOD* é definida na biblioteca *ieee.numeric_std.all* e retorna o resto da divisão dos dois números. A função *RESIZE* é definida na biblioteca *ieee.numeric_std.all* e retorna, quando truncado, os bits mais à direita do vetor passado no parâmetro.

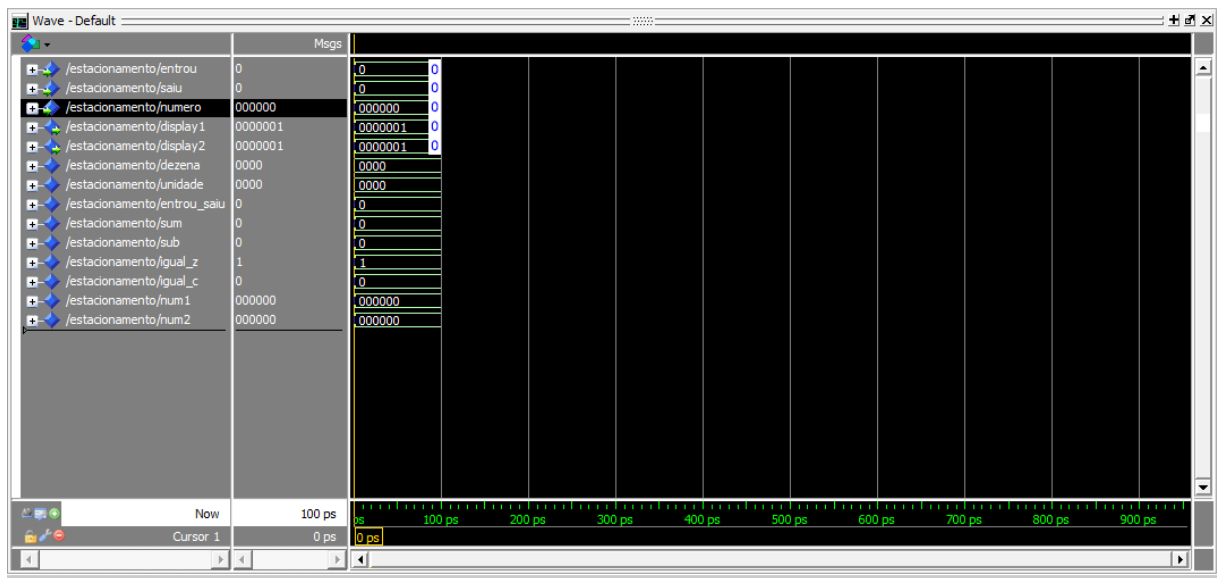
3.10 COMANDO 10

Utiliza o componente *bin_7seg* da seguinte maneira: **PORT MAP(*dezena*, *display1*(0), *display1*(1), *display1*(2), *display1*(3), *display1*(4), *display1*(5), *display1*(6))**; e serve para atribuir a *display1* o valor de *dezena* em código de 7 segmentos.

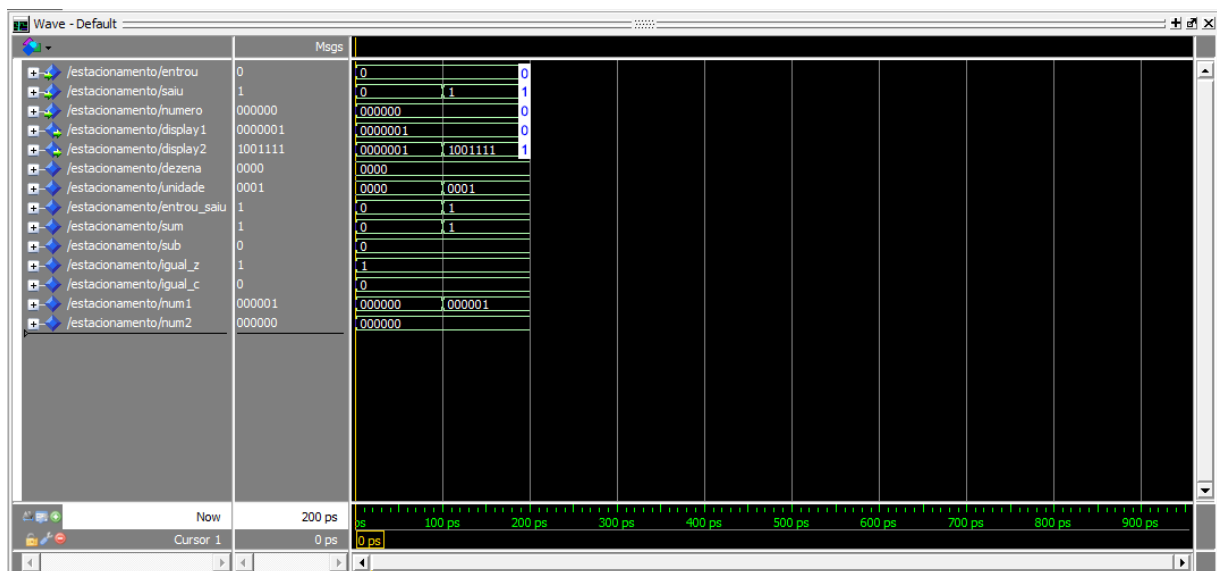
3.11 COMANDO 11

Utiliza o componente *bin_7seg* da seguinte maneira: **PORT MAP(*unidade*, *display2*(0), *display2*(1), *display2*(2), *display2*(3), *display2*(4), *display2*(5), *display2*(6))**; e serve para atribuir a *display2* o valor de *unidade* em código de 7 segmentos.

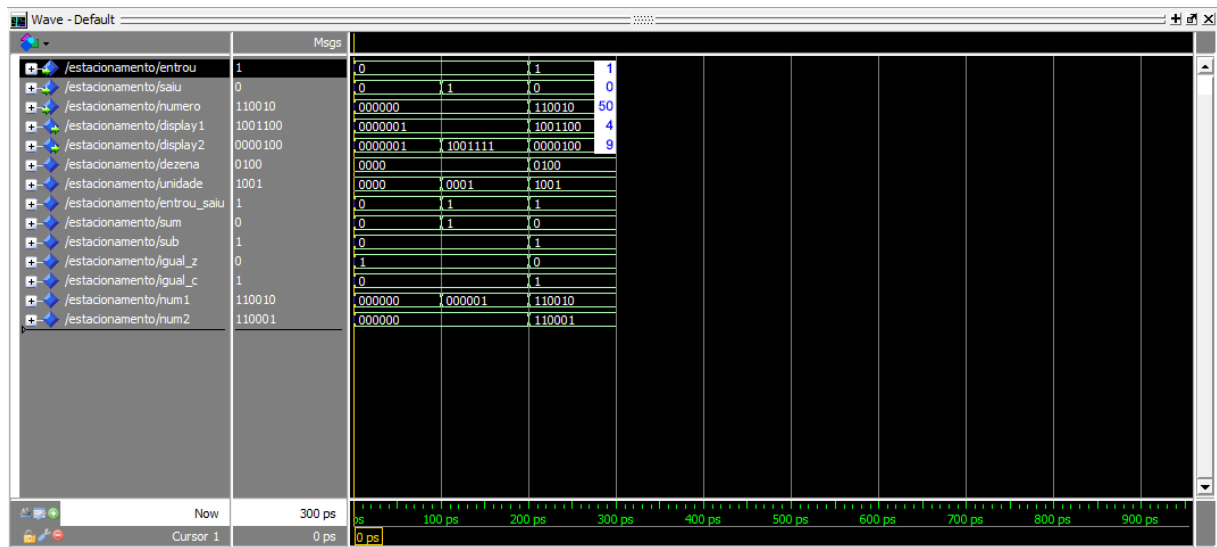
4. SIMULAÇÕES



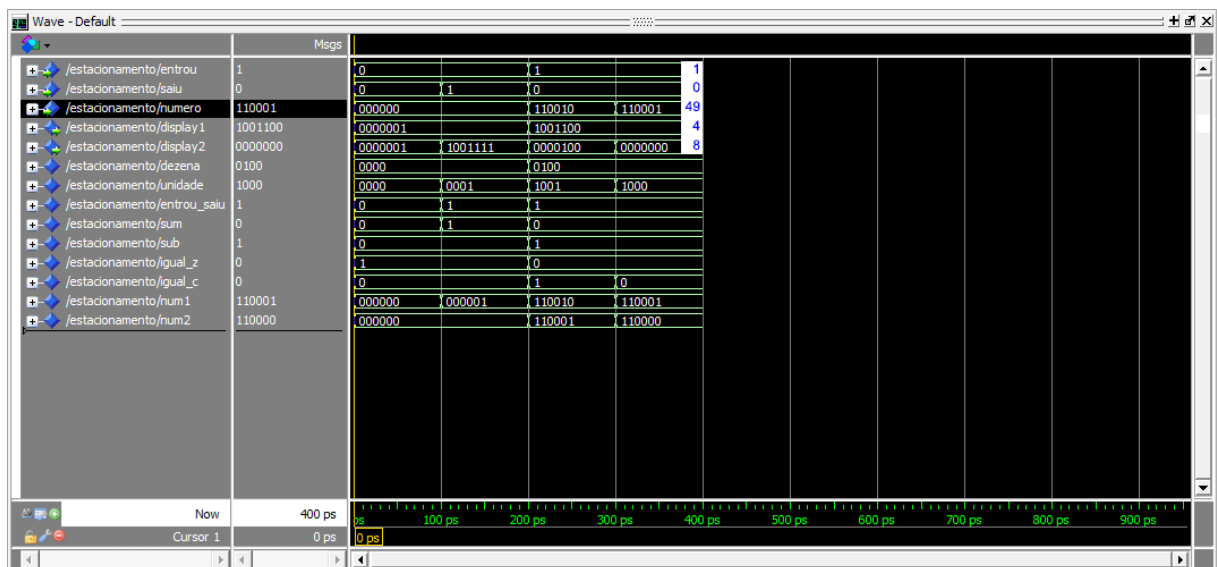
Entradas: *entrou* = 0; *saiu* = 0; *numero* = 0; **Saídas:** *display1* = 0; *display2* = 0;



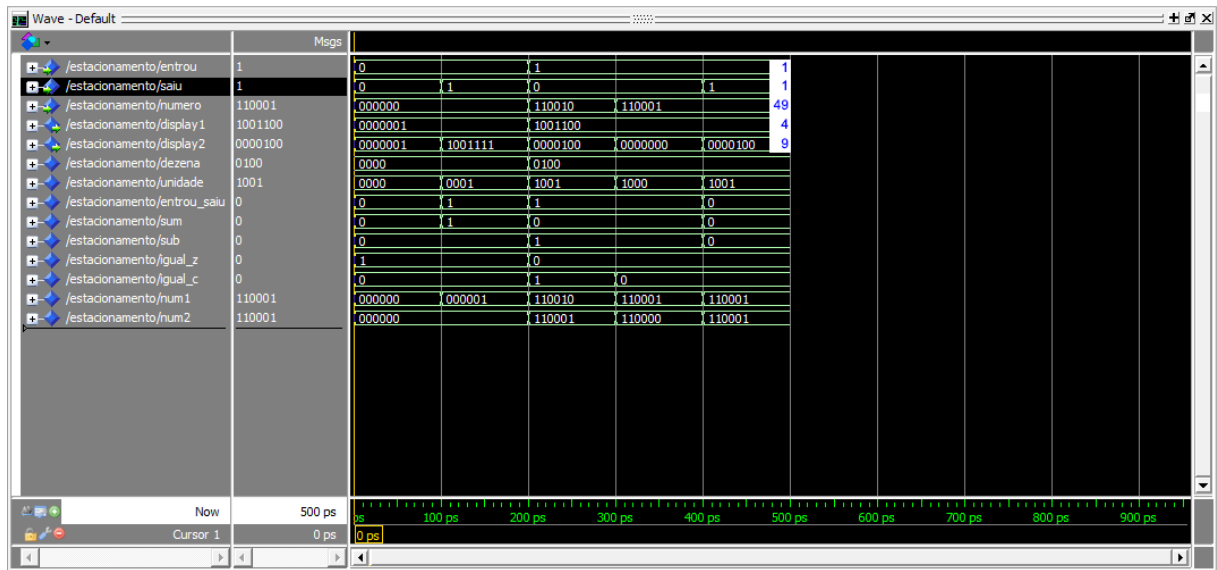
Entradas: *entrou* = 0; *saiu* = 1; *numero* = 0; **Saídas:** *display1* = 0; *display2* = 1;



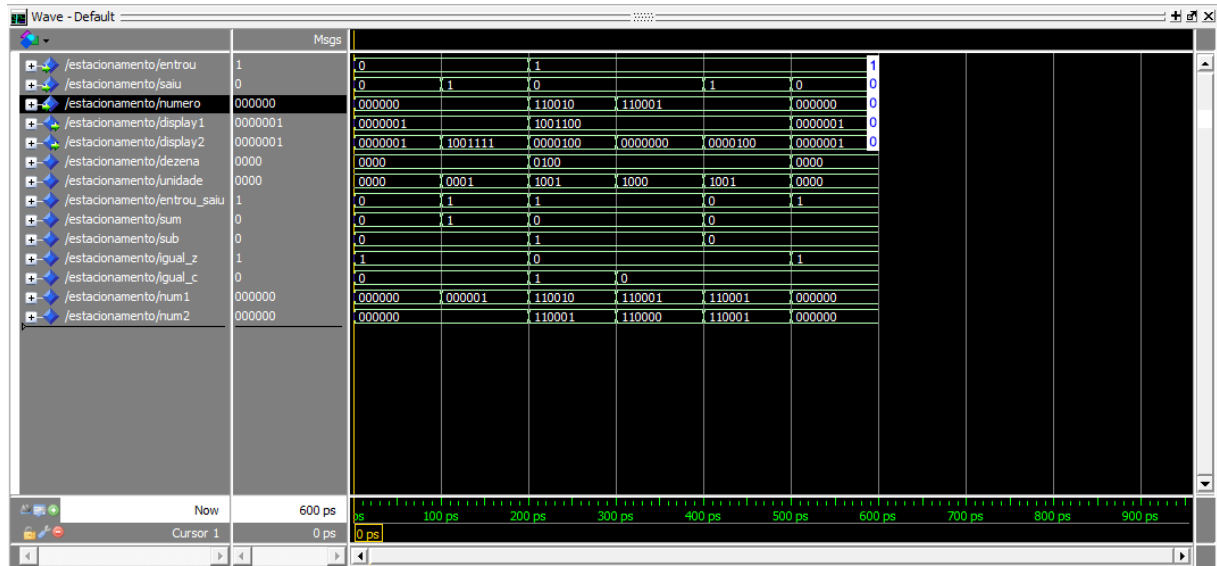
Entradas: *entrou* = 1; *saiu* = 0; *numero* = 50; **Saídas:** *display1* = 4; *display2* = 9;



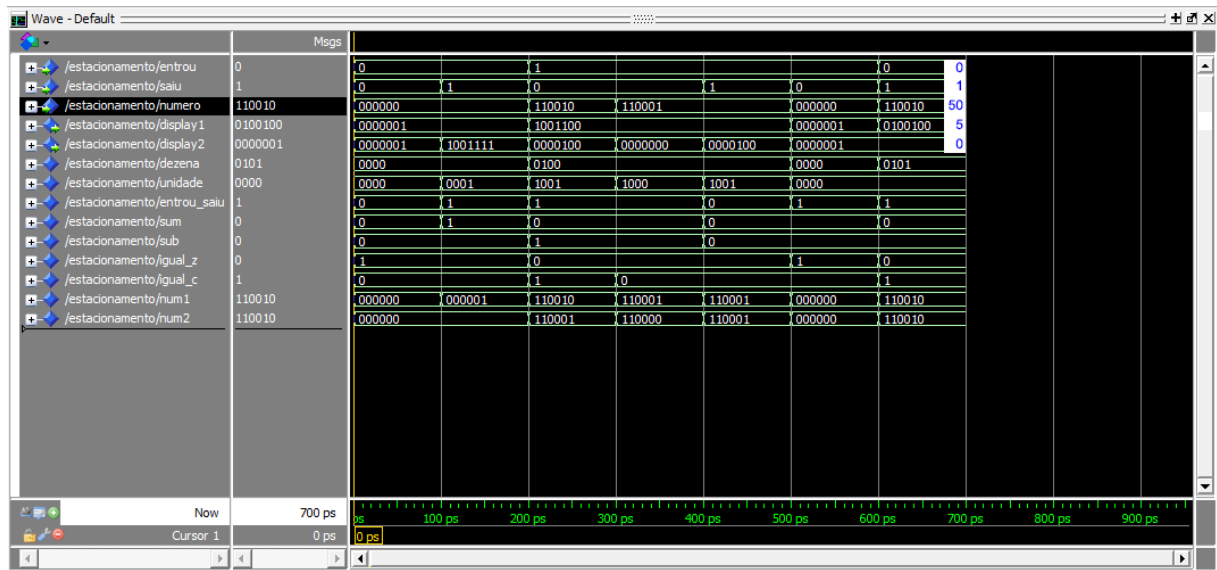
Entradas: *entrou* = 1; *saiu* = 0; *numero* = 49; **Saídas:** *display1* = 4; *display2* = 8;



Entradas: *entrou* = 1; *saiu* = 1; *numero* = 49; **Saídas:** *display1* = 4; *display2* = 9;



Entradas: *entrou* = 1; *saiu* = 0; *numero* = 0; **Saídas:** *display1* = 0; *display2* = 0;



Entradas: *entrou* = 0; *saiu* = 1; *numero* = 50; **Saídas:** *display1* = 5; *display2* = 0;

5. BIBLIOGRAFIA

- http://www.csee.umbc.edu/portal/help/VHDL/packages/numeric_std.vhd
- <http://fpgaparatodos.com.br/exemplos-de-aplicacao/exemplos-didaticos/4-decodificador-para-display-de-7-segmentos-em-vhdl.html>