

# CIRCUITOS ARITMÉTICOS

Júlia Pessoa Souza

Teoria de Eletrônica Digital-Turma C  
Faculdade Gama - Universidade de Brasília  
15/0133294  
Laboratório 3 Parte I

João Pedro Vergara Coneglian

Teoria de Eletrônica Digital-Turma C  
Faculdade Gama - Universidade de Brasília  
14/0145974  
Laboratório 3 Parte I

## RESUMO

O experimento foi criar um circuito com uma chave seletora, uma entrada de 3 bits e uma saída de 3 bits. Quando a chave é 0, a saída é igual a entrada, se a chave for 1, os bits da entrada são todos trocados e a saída é o complemento da entrada.

## 1. INTRODUÇÃO

Na Unidade Lógica e Aritmética de um hardware acontecem as operações lógicas. Essas operações são feitas por circuitos que recebem dados binários e podem somar, subtrair, multiplicar e dividir, de acordo com instruções da Unidade de Controle. Nesta experiência revisamos os circuitos somadores em VHDL e implementamos em uma placa. Viu-se também a chave seletora, a qual faz uma opção para qual parte do circuito ir de acordo com o comando dado.

## 2. EXPERIMENTO

### 2.1. Vivado

Foi utilizado o programa Vivado para escrever o algoritmo em VHDL. Criou-se um novo programa com a opção RTL project selecionada. foi escolhida a board Basys3 para fazer a implementação posteriormente. Foi adicionado o arquivo Basys3\_Master.xdc, então, criou-se o arquivo fonte em VHDL. Foram adicionadas as entradas A e SEL e a saída Z. Criou-se um vetor de 3 posições para A e outro para Z na entidade do algoritmo com o comando (2 DOWNT0 0). Na arquitetura do programa foi escrito o comando:  $Z \leq A \text{ when } (SEL = '0') \text{ else not } A \text{ when } (SEL = '1') \text{ else } "000"$ . O código foi simulado na parte de simulação do software, selecionando force clock nas entradas, utilizamos a opção binário e, adicionamos 50ns em A e 100ns em SEL.

### 2.2. Implementação

Abriu-se o arquivo Basys3\_Master.xdc e editou-se para que A[0] ficasse na porta V17, A[1] na V16, A[2] na W16 e SEL na W17. A saída Z[0] ficou no pino U16, Z[1] no E19 e Z[2] no V19. O arquivo foi compilado para ver o esquemático da síntese. Editou-se as propriedades para possibilitar a implementação. Foi colocado como verdadeiro a opção *Enable Bitstream Compression*, editou-se para 33MHz a Configuration Rate e foi selecionado o modo Master SPIx4. A implementação foi feita, gerou-se o bitstream, a placa foi conectada ao computador, visualizou-se o esquema do programa e estava implementado.

### 2.3. Figuras e Tabelas

```
C:/Users/AlunosFGA207/Downloads/lab_sd2_1-20170405T130713Z-001/project_1/project_1
29 -- Uncomment the following library declaration if instantiat
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity lab3 is
35     Port ( A : in STD_LOGIC_VECTOR (2 DOWNT0 0);
36           SEL : in STD_LOGIC;
37           Z : out STD_LOGIC_VECTOR (2 DOWNT0 0));
38 end lab3;
39
40 architecture Behavioral of lab3 is
41
42 begin
43
44     Z <= A when (SEL = '0') else
45     not A when (SEL = '1') else
46     "000";
47
48
49
50 end Behavioral;
51
```

Figura 1: algoritmo VHDL

```

9      #create_clock -add -name sys_clk_pin -period 10.00 -waveform
10
11 ## Switches
12 set_property PACKAGE_PIN V17 [get_ports {A[0]}]
13 set_property IOSTANDARD LVCMOS33 [get_ports {A[0]}]
14 set_property PACKAGE_PIN V16 [get_ports {A[1]}]
15 set_property IOSTANDARD LVCMOS33 [get_ports {A[1]}]
16 set_property PACKAGE_PIN W16 [get_ports {A[2]}]
17 set_property IOSTANDARD LVCMOS33 [get_ports {A[2]}]
18 set_property PACKAGE_PIN W17 [get_ports {SEL}]
19 set_property IOSTANDARD LVCMOS33 [get_ports {SEL}]
20 #set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
21 #set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
22 #set_property PACKAGE_PIN V15 [get_ports {sw[5]}]

```

Figura 2: Pinos de entrada.

```

47 set_property PACKAGE_PIN U16 [get_ports {Z[0]}]
48 set_property IOSTANDARD LVCMOS33 [get_ports {Z[0]}]
49 set_property PACKAGE_PIN E19 [get_ports {Z[1]}]
50 set_property IOSTANDARD LVCMOS33 [get_ports {Z[1]}]
51 set_property PACKAGE_PIN U19 [get_ports {Z[2]}]
52 set_property IOSTANDARD LVCMOS33 [get_ports {Z[2]}]
53 #set_property PACKAGE_PIN V19 [get_ports {led[3]}]

```

Figura 3: Pinos de saída.

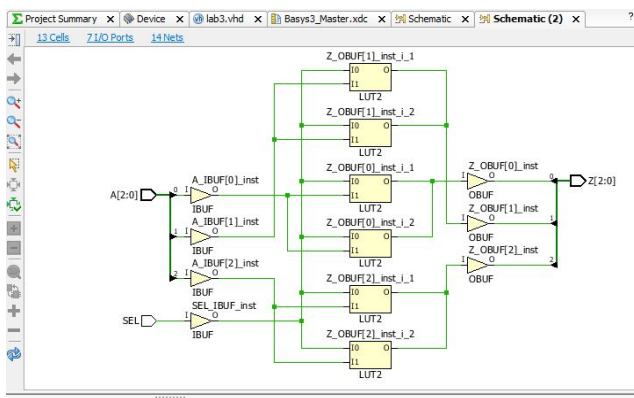


Figura 4: Síntese Design.

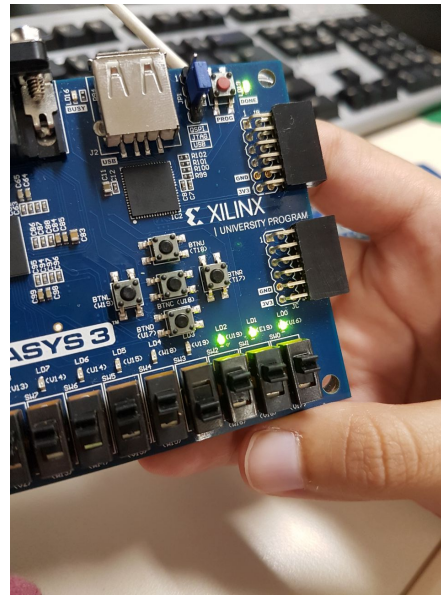


Figura 5: SEL=0 e A=111, Z=111.

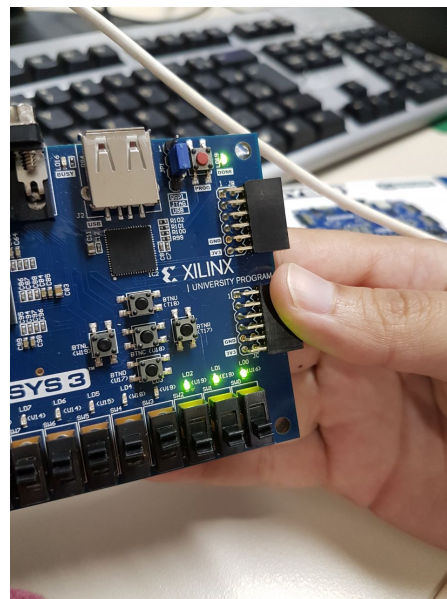


Figura 6: SEL=1 e A=000, Z=111.

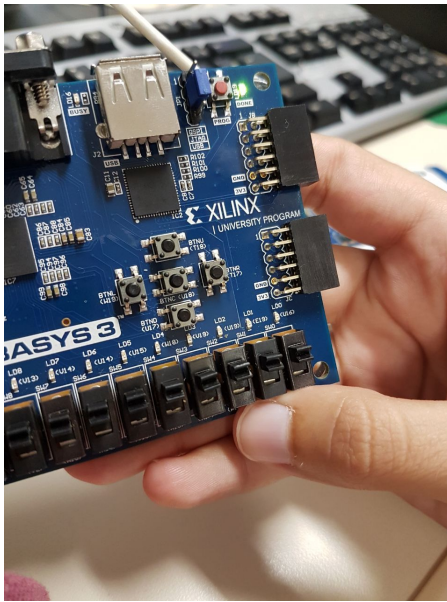


Figura 7: SEL=1 e A=111, Z=000.

### 3. RESULTADOS

Na Figura 1 está o algoritmo feito em VHDL. Nele foi informado que quando a chave seletora é 0, os bits de A devem continuar iguais. Mas quando a chave seletora é 1, os bits de A devem ser trocados, e, quando não é informada a chave seletora, os bits devem ser 000. Nas Figuras 2 e 3 estão os pinos utilizados para as entradas e a saída, já explicados anteriormente no item 2.2. Na Figura 4 é mostrada a síntese do circuito, onde pode ser constatado que, para o Z[0] são utilizadas as entradas A[0] e SEL, para Z[1] são utilizadas A[1] e SEL, para Z[2], A[2] e SEL. Nas Figuras 5, 6 e 7, pode-se verificar que o circuito foi implementado corretamente conforme a teoria. Quando

a chave SEL é 0, Z tem que apresentar os mesmos valores de A, quando é 1, Z tem que apresentar valores opostos de A. Na Figura 5, SEL está abaixada, então é 0, porém todos os bits de A estão levantados, portanto A é 111, então Z está com todos os bits acesos, o que significa que é 111. Na Figura 6 SEL é 1 então os bits de A tem que ser trocados, A é 000 e, conforme previsto, Z é 111. Na Figura 7 SEL também é 1 e A é 111, fazendo assim Z ser 000.

### 4. DISCUSSÃO E CONCLUSÕES

O experimento obteve sucesso ao mostrar os resultados esperados quando implantado na placa. Na teoria a entrada SEL deveria definir se a saída seria igual a entrada A ou o oposto da entrada A, e isto aconteceu corretamente de acordo com as imagens e os resultados. O circuito resultante da síntese também foi montado corretamente como pode ser visto na Figura 4, com cada bit de saída dependendo do bit de mesma posição de entrada e da chave SEL. Portanto, pode-se concluir que o código foi escrito corretamente e a experiência foi feita da maneira certa.

### 5. REFERÊNCIAS

- [1] I. Idoeta e I. Valeije, “Elementos de Eletrônica Digital”, Érica Ltda, São Paulo, 2008.
- [2] 2R. Tocci, N.Widmer e G. Moss, “Sistemas Digitais: Princípios e aplicações” Pearson.
- [3] G. Bezerra, “Apostila Prática de Eletrônica Digital I”, 2017
- [4] G. Bezerra, “Tutorial de Simulação e Implementação no Vivado”, 2017.