



Lab. 01 - Verilog

Chaves, LEDs e Multiplexadores

Projetos de Sistemas Integrados
Prof. Roberto Ribeiro Neli

Neste laboratório o aluno irá aprender como realizar conexões simples de entrada e saídas no FPGA e implementar circuitos que utilizam estes dispositivos. Iremos utilizar as chaves **SW₁₇₋₀** como dispositivos de entrada e os LEDs e o Display de 7 segmentos como saídas do circuito a ser projetado.

Parte I

O kit de desenvolvimento DE2-115 possui 18 interruptores, chamados **SW₁₇₋₀** que podem ser utilizadas como entradas de um circuitos e 18 LEDs vermelhos, chamados de **LEDR₁₇₋₀**, que podem ser utilizados como saídas. O exemplo abaixo mostra um modulo simples criado em VERILOG, que conecta as chaves em cada LED vermelho. Neste exemplo o estado lógico das chaves é mostrado nos LEDs. Neste exemplo as declarações são realizadas em forma de vetor, mas poderiam ser realizadas individualmente:

```
assign LEDR[17] = SW[17];  
assign LEDR[16] = SW[16]; → vetor  
...  
assign LEDR[0] = SW[0]; → individual
```

// Módulo simples para conectar as chaves nos LEDs:

```
module part1 (SW, LEDR);  
input [17:0] SW; // chaves  
output [17:0] LEDR; // LEDs vermelhos  
assign LEDR = SW;  
endmodule
```

Figura 1. Código Verilog que utiliza as chaves e LEDs do KIT DE2-115.

Implemente este exemplo no Quartus II e teste o seu funcionamento no KIT DE2-115.



Parte II

A figura 2a mostra um circuito de implementa um multiplexador 2X1, com seleção de entrada no pino s. Se s = 0 o multiplexador passa o valor de x para a saída m e se s = 1 o multiplexador passa o valor de y para a saída m. A figura 2b mostra a tabela verdade e a figura 2c mostra o símbolo do circuito multiplexador.

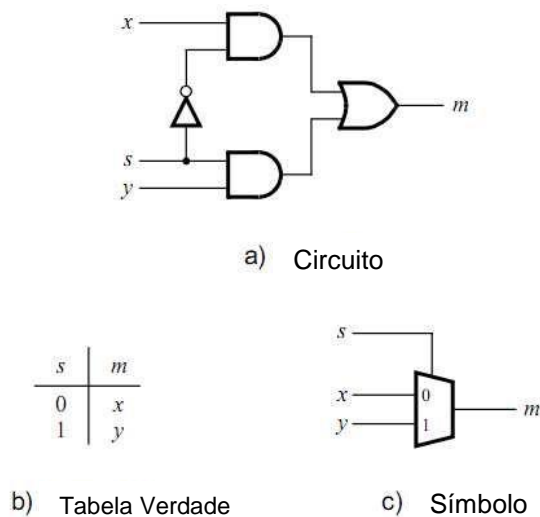


Figura 2 – Multiplexador 2x1.

O multiplexador pode ser descrito simplesmente em uma linha utilizando o Verilog, conforme abaixo:

assign m = (~s & x) | (s & y);

Utilizando o Quartus II, escreva um modulo que execute a função do circuito mostrado na figura 3. O modulo fará a função de 8 multiplexadores 2x1, onde teremos 16 entradas e 8 saídas. Utilize as chaves e os LEDs vermelhos para executar este projeto.

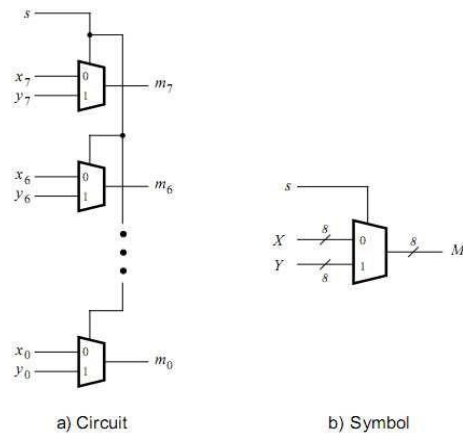


Figure 3. Multiplexador 2x1 de 8 bits.



Parte III

Nesta etapa iremos projetar um multiplexador 5x1, utilizando o que foi aprendido na etapa anterior. Este circuito terá 5 entradas u, v, w, x, e y, que serão transmitidas para a saída m, de acordo com o que for informado nos bits de seleção s0, s1 e s2. O circuito, a tabela verdade e o símbolo deste exemplo são ilustrados na figura 4.

A Figura 5 mostra um circuito de um multiplexador 5x1 com entradas de 3 bits.

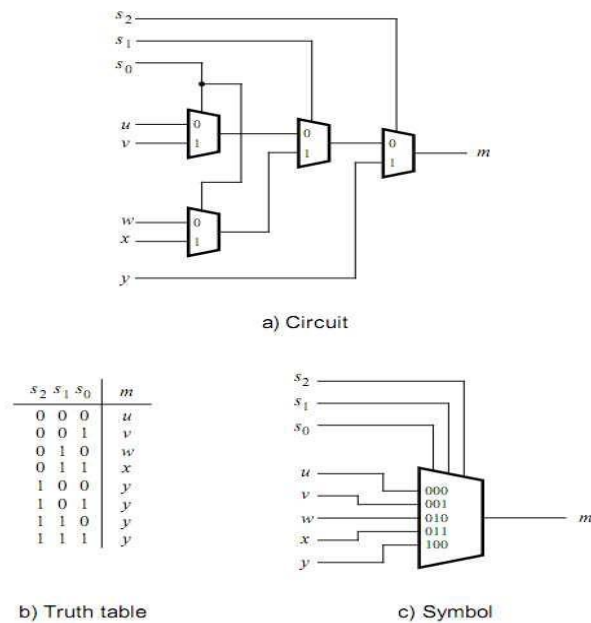


Figure 4. Multiplexador 5x1.

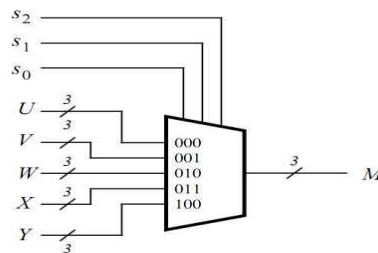


Figura 5. Multiplexador 5x1 de 3 bits.

Desenvolva um algoritmo em Verilog que realize a função de um multiplexador 5x1 com entradas de 3 bits, utilizando:

1. SW₁₇₋₁₅ para os bits de seleção;
2. SW₁₄₋₀ para as entradas de 3 bits de **U** a **Y**;
3. Os LEDR ligados diretamente em cada SW;
4. Os LEDG₂₋₀ para mostrar a saída **M**. Perform the following steps to implement the three-bit wide 5-to-1 multiplexer.



Parte IV

A figura 6 mostram um modulo decodificador BCD – 7Segmentos com 3 entradas e 7 saídas.

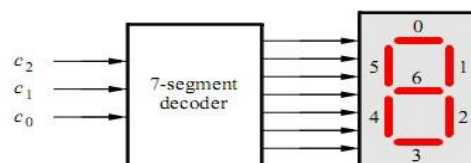


Figura 6. Decodificador BCD-7 Segmentos.

$c_2 c_1 c_0$	Character
000	H
001	E
010	L
011	O
100	
101	
110	
111	

Table 1. Character codes.

Crie um algoritmo em Verilog de um decodificador BCD-7Segmentos que execute a tabela 1, utilizando:

1. SW 2-0 como entradas BCD;
2. HEX0 para escrever a informação decodificada. Os segmentos no display são chamados de HEX00, HEX01, ..., HEX06, correspondendo a numeração da figura 6. Você pode declarar como um vetor: output [0:6] HEX0_D;

Parte V

A figura 7 ilustra um circuito completo onde se utiliza o multiplexador 5x1 e o conversor BCD-7Segmentos criados nos exercícios anteriores.

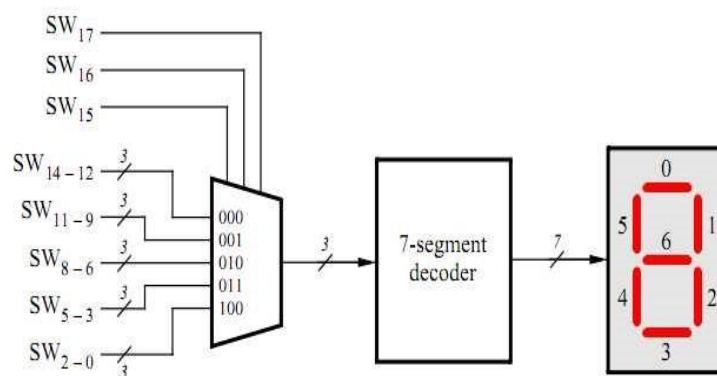


Figura 7. Circuito que mostra 1 de 5 caracteres.

$SW_{17} SW_{16} SW_{15}$	Character pattern				
000	H	E	L	L	O
001	E	L	L	O	H
010	L	L	O	H	E
011	L	O	H	E	L
100	O	H	E	L	L

Tabela 2. Palavra HELLO rotativa escrita nos displays.

Escreva um algoritmo em Verilog que escreva nos displays HEX0, ..., HEX4, as letras de acordo com a tabela 2. Utilize as chaves SW



Parte VI

Modifique o seu projeto da etapa anterior, para que execute o que é mostrado na Tabela 3. Neste algoritmo, você deve utilizar todos os displays de 7 segmentos e fazer com que a palavra HELLO gire, na sequência mostrada na tabela 3, de forma automática. O giro deve acontecer a uma frequência de 1s. O giro deve ser realizado utilizando o comando **Always** do Verilog.

SW_{17} SW_{16} SW_{15}	Character pattern					
000			H	E	L	L O
001			H	E	L	L O
010		H	E	L	L	O
011	H	E	L	L	O	
100	E	L	L	O		H
101	L	L	O			H E
110	L	O			H	E L
111	O			H	E	L L

Tabela 3. Palavra HELLO giratória em 8 displays.



= Chamar o professor para validar o exercício.

Adaptado dos exemplos que acompanham o KIT DE2-115.