



**FUNDAÇÃO EDSON QUEIROZ**  
**UNIVERSIDADE DE FORTALEZA**  
ENSINANDO E APRENDENDO

## **LABORATÓRIO DE SISTEMAS DIGITAIS AVANÇADOS**

Prof. Imbiriba

### **Exercício 01 – Chaves, LEDs, e Multiplexadores**

O objetivo destes exercícios é aprender a conectar dispositivos simples de entrada e saída a um chip FPGA e implementar um circuito que usa esses dispositivos. Serão usadas as chaves nas placas DE0 como entradas para o circuito. Também serão usados diodos emissores de luz (LEDs) e displays de 7 segmentos como dispositivos de saída.

#### **PARTE 1**

As placas DE10-Lite, DE0-CV e DE1-SoC fornecem dez interruptores e leds, chamadas SW9-0 e LEDG9-0. Da mesma forma, o DE2-115 fornece dezesseis interruptores e leds. Essas chaves podem ser usadas para fornecer entradas e os leds podem ser usadas como dispositivos de saída. A Figura 1 mostra uma entidade VHDL simples que usa dez chaves e mostra seus estados nos LEDs. As várias chaves e leds são convenientes para apresentar os vetores de entradas e saídas mostrado no código VHDL abaixo. Foi usada uma única declaração de atribuição para todas as saídas de LEDR, que é equivalente às atribuições individuais: LEDR (9) <= SW (9); LEDR (8) <= SW (8); ... LEDR (0) <= SW (0).

As placas da série DE possuem conexões cabeadas entre o chip FPGA, os leds, e as chaves. Para usar leds e chaves, é necessário incluir no seu projeto Quartus as atribuições de pinos corretas, fornecidas no manual do usuário da sua placa. Por exemplo, o manual DE0 especifica que SW0 está conectado ao pino J6 da FPGA e LEDG0 está conectado ao pino J1. A atribuição de pinos no software Quartus é feita mediante a ferramenta PIN PLANNER.

Se usado o arquivo de atribuição de pinos fornecido para a placa DE, é importante perceber que essas atribuições definidas no arquivo são úteis apenas se os nomes dos pinos que aparecem no arquivo forem exatamente iguais aos nomes de portas usados na sua entidade VHDL. Por exemplo, se o arquivo de atribuição de pinos usa os nomes SW (0), ..., SW (9) e LEDR (0), ..., LEDR (9), esses são os nomes que devem ser usados para entrada e portas de saída no código VHDL, como feito na Figura 1.

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  ENTITY part1 IS
5  PORT ( SW : IN STD_LOGIC_VECTOR(9 DOWNTO 0);
6        LEDR : OUT STD_LOGIC_VECTOR(9 DOWNTO 0));
7  END part1;
8
9  ARCHITECTURE hardware OF part1 IS
10 BEGIN LEDR <= SW;
11 END hardware;
12

```

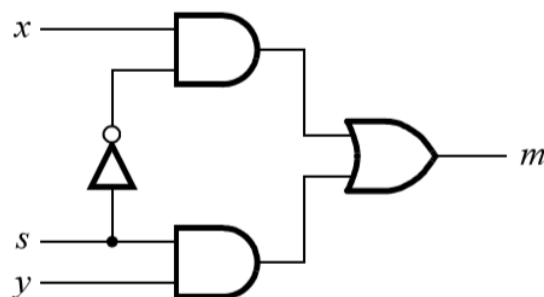
Figura 1: Código VHDL utilizando as 10 chaves e leds da DE0.

Execute as seguintes etapas para implementar um circuito correspondente ao código da Figura 1 na placa da série DE0.

- Crie um novo projeto Quartus para o seu circuito. Selecione o chip alvo que corresponde à sua placa da série DE0.
- Crie uma entidade VHDL para o código na Figura 1 e inclua-a no seu projeto.
- Inclua em seu projeto as atribuições de pinos da DE0, conforme discutido acima. Compile o projeto.
- Faça o download do circuito compilado no chip FPGA usando a ferramenta Quartus Programmer (o procedimento para usar a ferramenta Programmer é descrito no tutorial Introdução ao Quartus). Teste a funcionalidade do circuito alternando as chaves e observando os LEDs.

## PARTE 2

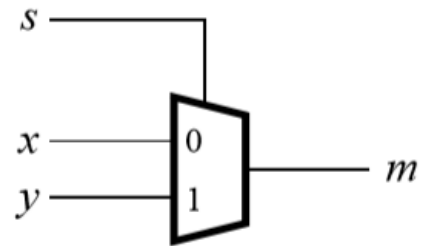
A Figura 2a mostra um circuito de soma de produtos que implementa um multiplexador 2 para 1 com uma entrada selecionada  $s$ . Se  $s = 0$ , a saída  $m$  do multiplexador é igual à entrada  $x$ , e se  $s = 1$ , a saída é igual a  $y$ . A parte b da figura fornece uma tabela verdade para este multiplexador e a parte c mostra seu símbolo de circuito.



- Circuito

$s$	$m$
0	$x$
1	$y$

- Tabela verdade



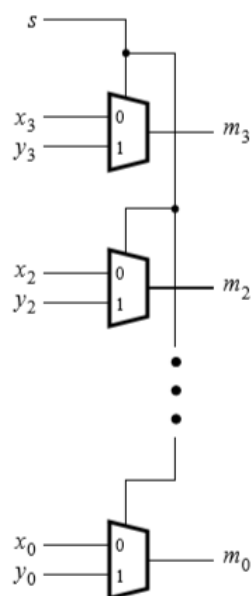
c) Símbolo

Figura 2: Multiplex de 2 para 1.

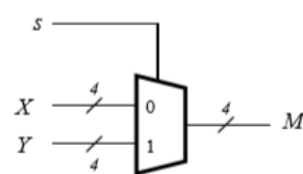
Um multiplexador pode ser descrito pela seguinte sentença VHDL:

$m \leq (\text{NOT } (s) \text{ AND } x) \text{ OR } (s \text{ AND } y);$

Vamos criar uma entidade VHDL constituída de 4 circuitos multiplexadores como mencionado acima e de acordo com o mostrado na Figura 3a. Este circuito possui duas entradas de quatro bits, X e Y, e produz a saída de quatro bits M. Se  $s = 0$ , então  $M = X$ , enquanto que se  $s = 1$ , então  $M = Y$ . Nós nos referimos a este circuito como um multiplexador de 2 para 1 de quatro bits. Possui o símbolo de circuito mostrado na Figura 3b, no qual X, Y e M são representados como fios de quatro bits.



a) Circuito



b) Símbolo

Figura 3: Multiplexador de 2 para 1 de 4 bits.

Execute as etapas listadas abaixo.

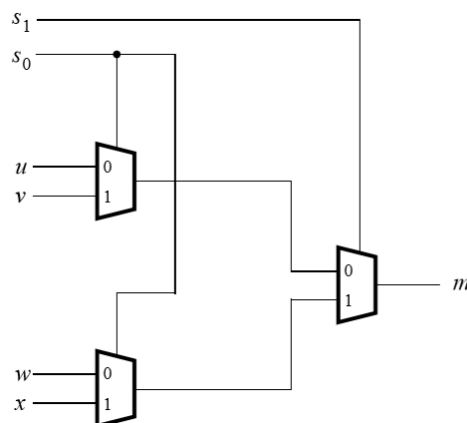
1. Crie um novo projeto Quartus para o seu circuito.

2. Inclua seu arquivo VHDL para o multiplexador 2 para 1 de quatro bits no seu projeto. Use o comutador SW9 como a entrada  $s$ , alterna o SW3–0 como a entrada  $X$  e o SW7–4 como a entrada  $Y$ . Exiba o valor da entrada no LEDG9, conecte a saída  $M$  ao LEDG3–0 e conecte os LEDG não utilizadas ao valor constante 0.
3. Inclua no seu projeto as atribuições de pinos necessárias para a placa DE0. Conforme discutido na Parte 1. Essas atribuições garantem que os códigos do seu VHDL serão as entradas no código do FPGA conectado às chaves SW e LEDG.
4. Compile o projeto e faça o download do circuito resultante no chip FPGA. Teste a funcionalidade do multiplexador 2 para 1 de quatro bits, alternando os comutadores e observando os LEDs.

### PARTE 3

Na Figura 2, mostramos um multiplexador 2 para 1 que seleciona entre as duas entradas  $x$  e  $y$ . Para esta parte, considere um circuito no qual a saída  $m$  deve ser selecionada entre quatro entradas  $u$ ,  $v$ ,  $w$  e  $x$ . O circuito da Figura 4 mostra como podemos construir um multiplexador 4 para 1 usando três multiplexadores 2 para 1. O circuito usa uma entrada de seleção de 2 bits  $s_1$ ,  $s_0$  e implementa a tabela verdade mostrada na Figura 4b. Um símbolo de circuito para este multiplexador é dado na figura 4c.

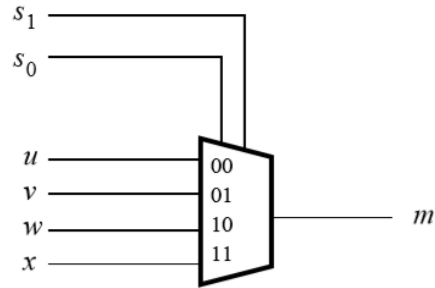
Lembre-se da Figura 3 de que um multiplexador 2 para 1 com largura de quatro bits pode ser construído usando quatro instâncias de um multiplexador 2 para 1. A Figura 5 aplica esse conceito para definir um multiplexador de 4 para 1 de dois bits. Ele contém duas instâncias do circuito na Figura 4a.



a) Circuito

$s_1$	$s_0$	$m$
0	0	$u$
0	1	$v$
1	0	$w$
1	1	$x$

b) Tabela verdade



c) Símbolo

Figura 4: Multiplexador de 4 para 1.

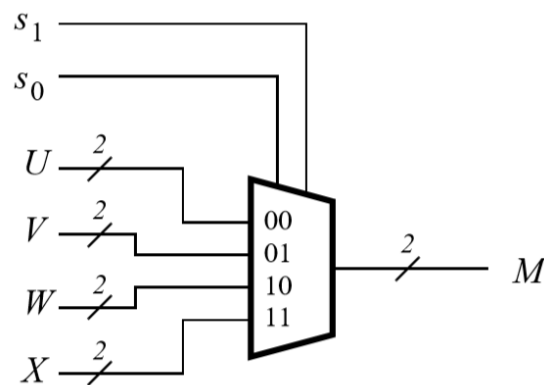


Figura 5: Multiplexador de 4 para 1 de 2 bits

Execute as seguintes etapas para implementar o multiplexador de 4 para 1 de dois bits. 1. Crie um novo projeto Quartus para o seu circuito.

2. Crie uma entidade VHDL para o multiplexador de 4 para 1 de largura de dois bits. Conecte suas entradas de seleção às chaves SW9–8 e use as chaves SW7–0 para fornecer as quatro entradas de 2 bits U a X. Conecte a saída M aos leds LEDG1–0.

3. Inclua no seu projeto as atribuições de pinos necessárias para a placa DE0. Compile o projeto.

4. Faça o download do circuito compilado no chip FPGA. Teste a funcionalidade do multiplexador de 4 para 1 de dois bits de largura, alternando as chaves e observando

os LEDs. Certifique-se de que cada uma das entradas U a X está associada corretamente a saída M.

#### PARTE 4

O objetivo desta parte é exibir um caractere em um display de 7 segmentos. O caractere específico exibido depende de uma entrada de dois bits. A Figura 6 mostra uma entidade decodificadora de 7 segmentos que possui uma entrada de dois bits  $c_1$ ,  $c_0$ . Esse decodificador produz sete saídas que são usadas para exibir um caractere em um display de 7 segmentos. A Tabela 2 lista os caracteres que devem ser exibidos no display da DE0 dependendo da variação de  $c_1$ ,  $c_0$ . Observe que, em alguns casos, o caractere 'em branco' é selecionado pelo código 11.

Os sete segmentos do display são identificados pelos índices de 0 a 6 mostrados na figura. Cada segmento é iluminado levando-o ao valor lógico 0. Você deve escrever uma entidade VHDL que implemente funções lógicas para ativar cada um dos sete segmentos. Use apenas instruções de atribuição VHDL simples no seu código para especificar cada função lógica usando uma expressão booleana.

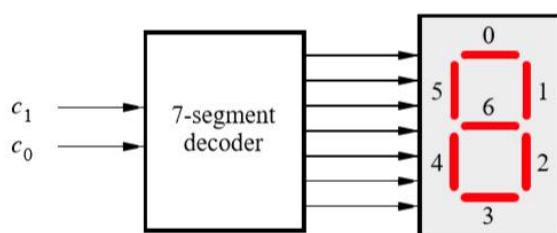


Figura 6: Um decodificador de 7 segmentos

$c_1 c_0$	DE10-Lite	DE0-CV	DE1-SoC	DE2-115
00	d	d	d	d
01	E	E	E	E
10	1	0	1	2
11	0			

Tabela 2: Código de caracteres para as placas da série DE.

Execute as seguintes etapas:

1. Crie um novo projeto Quartus para o seu circuito.
2. Crie uma entidade VHDL para o decodificador de 7 segmentos. Conecte as entradas  $c_1$ ,  $c_0$  às chaves SW1-0 e conecte as saídas do decodificador ao display HEX0 na placa DE0. Os segmentos nesta exibição são chamados HEX00, HEX01, ..., HEX06, correspondendo à Figura 6. Você deve declarar a porta de 7 bits:

## HEX0: OUT STD\_LOGIC\_VECTOR (0 A 6);

No seu projeto Quartus faça a atribuição de pinos na FPGA usando a ferramenta PIN PLANNER.

3. Depois de fazer as atribuições de pinos necessárias, compile o projeto.
4. Faça o download do circuito compilado no chip FPGA. Teste a funcionalidade do circuito alternando os interruptores SW1–0 e observando a tela de 7 segmentos.

### PARTE 5

Considere o circuito mostrado na Figura 7. Ele usa um multiplexador de 4 para 1 de dois bits de largura para permitir a seleção de quatro caracteres que são exibidos em um display de 7 segmentos. Usando o decodificador de 7 segmentos da Parte IV, este circuito pode exibir os caracteres d, E, 0, 1, 2 ou 'em branco', dependendo da placa da série DE. Os códigos dos caracteres são definidos de acordo com a Tabela 2, usando as chaves SW7–0, e um caractere específico é selecionado no display mediante a seleção das chaves SW9–8.

Um esboço do código VHDL que representa esse circuito é fornecido na Figura 8. Observe que usamos os circuitos das Partes III e IV como subcircuitos neste código. Você deve estender o código na Figura 8 para que ele use quatro displays de 7 segmentos em vez de apenas um. Você precisará usar quatro instâncias de cada um dos subcircuitos. O objetivo do circuito é exibir qualquer palavra nos três displays de 7 segmentos, composta pelos caracteres da Tabela 2, e poder girar essa palavra de forma circular entre os displays quando os comutadores SW9–8 são alternados.

Como exemplo, se a palavra exibida for dE10, seu circuito deverá produzir os padrões de saída ilustrados na Tabela 3.

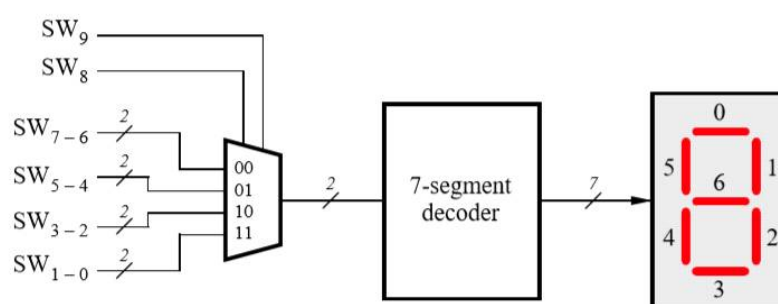


Figura 7: Circuito de seleção de display de 7 segmentos para exibição de 4 caracteres.

$SW_{9-8}$	Characters			
00	d	E	1	0
01	E	1	0	d
10	1	0	d	E
11	0	d	E	1

Tabela 3: Rotação dos caracteres “dE10” através dos 4 displays de 7 segmentos

### Atividade Extra

Execute os seguintes passos.

1. Crie um novo projeto Quartus para o seu circuito.
2. Inclua a entidade VHDL no projeto Quartus. Conecte as chaves SW9–8 às entradas selecionadas de cada uma das três instâncias dos multiplexadores de 3 para 1 de dois bits. Conecte também o SW5–0 a cada instância dos multiplexadores, conforme necessário para produzir os padrões de caracteres mostrados na Tabela 2. Conecte os comutadores SW ao LEDG dos leds e conecte as saídas dos três multiplexadores aos displays de 7 segmentos HEX2, HEX1 e HEX0.
3. Faça as atribuições de pinos necessárias para a placa da série DE0 para todos as chaves, LEDs e displays de 7 segmentos. Compile o projeto.
4. Faça o download do circuito compilado na FPGA. Teste a funcionalidade do circuito configurando os códigos de caracteres adequados nos comutadores SW5-0 e depois alternando SW9–8 para observar a rotação dos caracteres.