

## Prática 5 - Linguagem VHDL

Prof. Cesar da Costa

### Circuito Decodificador BCD para Display de Sete Segmentos

Um dos métodos mais simples para apresentação de dígitos alfanuméricos, em circuitos digitais, é um display que usa uma configuração de sete segmentos, conforme indica a Figura 5.37 (a). O display pode ser ligado em dois tipos de configurações, sendo (i) anodo comum e (ii) catodo comum. Essas configurações são ilustradas na Figura 5.37 (b) e (c).

Um decodificador BCD para display de sete segmentos é um circuito digital, usado para receber uma entrada no código BCD de quatro bits (D0, D1, D2 e D3) e gerar sete saídas (a, b, c, d, e, f, g), que acionam os segmentos apropriados do display, para indicar um dígito decimal. O oitavo bit (h) de saída do decodificador BCD representa o ponto decimal, que neste exercício será ignorado e mantido em nível 1 (alto), configuração anodo comum.

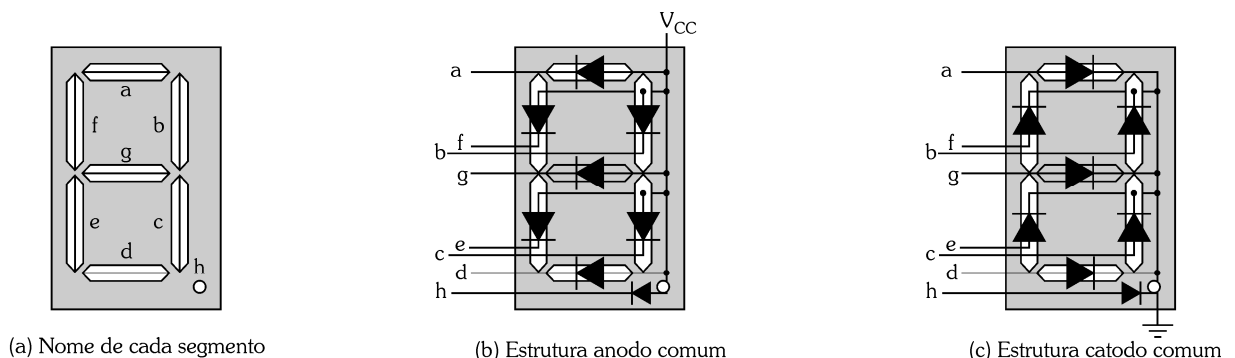
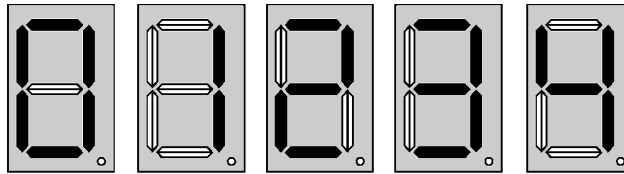
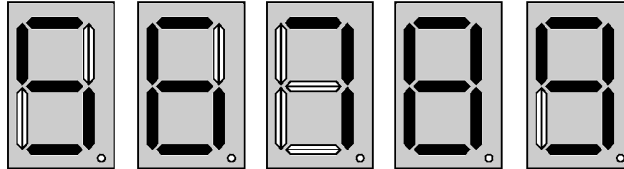


Figura 5.37 - Display de sete segmentos.

O nível lógico do código de saída dos sinais que acionam os sete segmentos do display depende da configuração anodo comum ou catodo comum, utilizada na ligação do display. A Figura 5.38 ilustra os níveis lógicos de saída, do decodificador BCD, para valores decimais de 0 a 9, nas configurações anodo comum e catodo comum do display. Por exemplo, o número 1 em decimal, para ser exibido num display com configuração anodo comum, necessita que o decodificador BCD gere a seguinte sequência de bits em sua saída: 11111001 (segmentos h, g, f, e, d, c, b, a), o que significa que um nível lógico baixo (zero) deve ser aplicado aos segmentos b e c do display para que eles acendam. Os demais segmentos devem permanecer em nível lógico alto (um), ou seja, apagados.



Anodo comum: 11000000    11111001    10100100    10110000    10011001  
 Catodo comum: 00111111    00000110    01011011    01001111    01100110



Anodo comum: 10010010    10000010    11111000    10000000    10010000  
 Catodo comum: 01101101    01111101    00000111    01111111    01101111

Figura 5.38 - Níveis lógicos de saída do decodificador BCD.

A Tabela 5.9 apresenta a tabela verdade do decodificador BCD para display de sete segmentos, para valores decimais de 0 a 9, com display ligado na configuração anodo comum.

Tabela 5.9 - Tabela verdade do decodificador BCD para display de sete segmentos.

Entrada D3 D2 D1 D0	Saída (segmentos) h g f e d c b a	Dígito decimal
0000	11000000	0
0001	11111001	1
0010	10100100	2
0011	10110000	3
0100	10011001	4
0101	10010010	5
0110	10000010	6
0111	11111000	7
1000	10000000	8
1001	10010000	9

Observe na Tabela 5.9 que cada segmento de saída será energizado quando sua saída correspondente for nível baixo (zero), pois a tabela foi criada para ligação de display na configuração anodo comum.

### Exercício resolvido

Criar e simular em VHDL um circuito decodificador BCD para display de sete segmentos, conforme a Tabela 5.9. Verificar o circuito projetado no interior do FPGA (RTL viewer).

### Solução

1. Crie um projeto chamado DECOD7SEG. Não se esqueça de especificar o dispositivo FPGA.

2. Crie o arquivo *DECOD7SEG.vhd* em VHDL.

declare as bibliotecas;  
declare a entidade;  
declare a arquitetura.

A Figura 5.39 apresenta o programa escrito em código VHDL no Editor de Texto. Salve o programa.

```
1 LIBRARY ieee;
2 USE ieee.std_logic_1164.all;
3 Use ieee.std_logic_arith.all;
4 Use ieee.std_logic_unsigned.all;
5
6 ENTITY Decod7seg IS
7
8     PORT(
9         Data_In : IN STD_LOGIC_VECTOR(3 downto 0);
10        a,b,c,d,e,f,g,h : OUT STD_LOGIC);
11
12 END ENTITY Decod7seg;
13
14 ARCHITECTURE comportamental OF Decod7seg IS
15 BEGIN
16
17 PROCESS (Data_In)
18 BEGIN
19
20 CASE Data_In IS
21
22
23 when "0000" => a<='0';b<='0';c<='0';d<='0';e<='0';f<='0';g<='1';h<='1';
24 when "0001" => a<='1';b<='0';c<='0';d<='1';e<='1';f<='1';g<='1';h<='1';
25 when "0010" => a<='0';b<='0';c<='1';d<='0';e<='0';f<='1';g<='0';h<='1';
26 when "0011" => a<='0';b<='0';c<='0';d<='0';e<='1';f<='1';g<='0';h<='1';
27 when "0100" => a<='1';b<='0';c<='0';d<='1';e<='1';f<='0';g<='0';h<='1';
28 when "0101" => a<='0';b<='1';c<='0';d<='0';e<='1';f<='0';g<='0';h<='1';
29 when "0110" => a<='0';b<='1';c<='0';d<='0';e<='0';f<='0';g<='0';h<='1';
30 when "0111" => a<='0';b<='0';c<='0';d<='1';e<='1';f<='1';g<='1';h<='1';
31 when "1000" => a<='0';b<='0';c<='0';d<='0';e<='0';f<='0';g<='0';h<='1';
32 when "1001" => a<='0';b<='0';c<='0';d<='0';e<='1';f<='0';g<='0';h<='1';
33 When OTHERS => a<='1';b<='1';c<='1';d<='1';e<='1';f<='1';g<='1';h<='1';
34 END CASE;
35 END PROCESS;
36
37 END ARCHITECTURE comportamental;
```

Figura 5.39 - Programa escrito em código VHDL.

O elemento fundamental para a descrição comportamental de uma arquitetura é a instrução PROCESS. A arquitetura do decodificador BCD contém um único processo. A entrada Data\_In é do tipo IN STD\_LOGIC\_VECTOR (3 downto 0), que aplica quatro bits de entrada na entidade.

A instrução CASE - WHEN, quando é executada, para cada valor específico da entrada Data\_In gera um conjunto de bits na saída, correspondente aos segmentos que formarão o número decimal correspondente no display.

A linha 33 define que para qualquer caso que não esteja previsto na entrada de dados (Data\_In), na Tabela 5.9, as saídas serão mantidas em nível 1 (alto), desligadas.

3. Compile e salve o projeto para verificar a existência de erros.
4. Se o compilador acusar algum erro, verifique as mensagens de erros na parte inferior da janela, corrija os erros e compile novamente.
5. Faça a simulação do funcionamento do circuito.
6. Escolha a simulação funcional.
7. Escolha o arquivo Vector waveform.
8. Insira todas as entradas e saídas no Editor de Formas de Ondas.  
Selecione como entrada Data\_In (barramento contendo as quatro entradas D0, D1, D2 e D3) e não selecione as entradas individuais.  
Selecione as saídas a, b, c, d, e, f, g e h individualmente.
9. Altere as propriedades das entradas para binário.
10. Selecione o tipo de sinal de estímulo para cada entrada.
11. Selecione o sinal Count Value (gerador de onda quadrada) para a entrada Data\_In, que deve iniciar com 0000.
12. Teste o arquivo de sinais de estímulos criado.
13. Salve o arquivo **DECOD7SEG.vwf**.
14. Inicie a simulação:

O arquivo de sinais de estímulo é compilado e simulado. Se a compilação ocorrer sem nenhum erro, o resultado da simulação, saídas a, b, c, d, e, f, g, h, em função das

entradas Data\_IN (D3, D2, D1 e D0), é igual à Tabela 5.9 e o resultado da simulação gráfica do DECOD7SEG é apresentado na Figura 5.40.

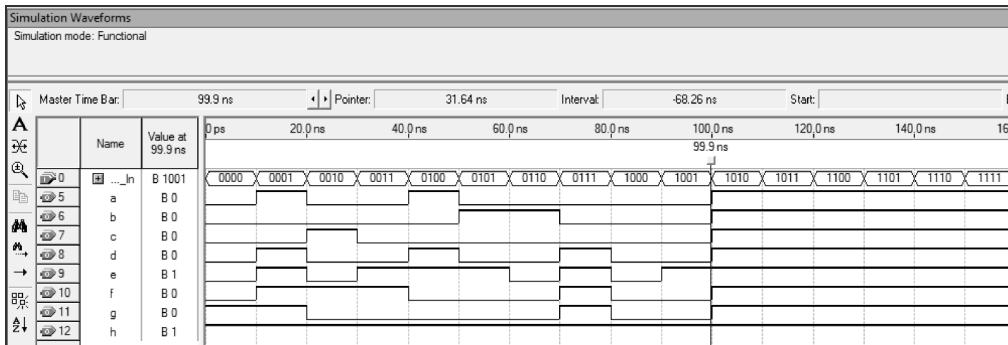


Figura 5.40 - Resultado da simulação funcional do circuito.

Observe que na simulação funcional, no intervalo de 0 a 10 ns, Data\_In (D0=0, D1=0, D2=0, D3=0), as saídas a=0, b=0, c=0, d=0, e=0, f=0, g=1, h=1, correspondente ao decimal 0 (zero). No intervalo de 10 a 20 ns, Data\_In (D0=1, D1=0, D2=0, D3=0), as saídas a=1, b=0, c=0, d=1, e=1, f=1, g=1, h=1, correspondente ao decimal 1 (um). No intervalo de 20 a 30 ns, Data\_In (D0=0, D1=1, D2=0, D3=0), as saídas a=0, b=0, c=1, d=0, e=0, f=1, g=0, h=1, correspondente ao decimal 2 (dois). No intervalo de 30 a 40 ns, Data\_In (D0=1, D1=1, D2=0, D3=0), as saídas a=0, b=0, c=0, d=0, e=1, f=1, g=0, h=1, correspondente ao decimal 3 (três). No intervalo de 90 a 100 ns, Data\_In (D0=1, D1=0, D2=0, D3=1), as saídas a=0, b=0, c=1, d=0, e=1, f=0, g=0, h=1, correspondente ao decimal 9 (nove).

Para os valores de Data\_In (1010 a 1111) com saídas não previstas na Tabela 5.9, os sete segmentos são mantidos em nível alto (um), ou seja, correspondente ao display apagado.

15. A Figura 5.41 mostra o circuito lógico do decodificador BCD para sete segmentos em termos de portas lógicas (RTL) e suas interconexões (netlist), implementado no interior do FPGA.

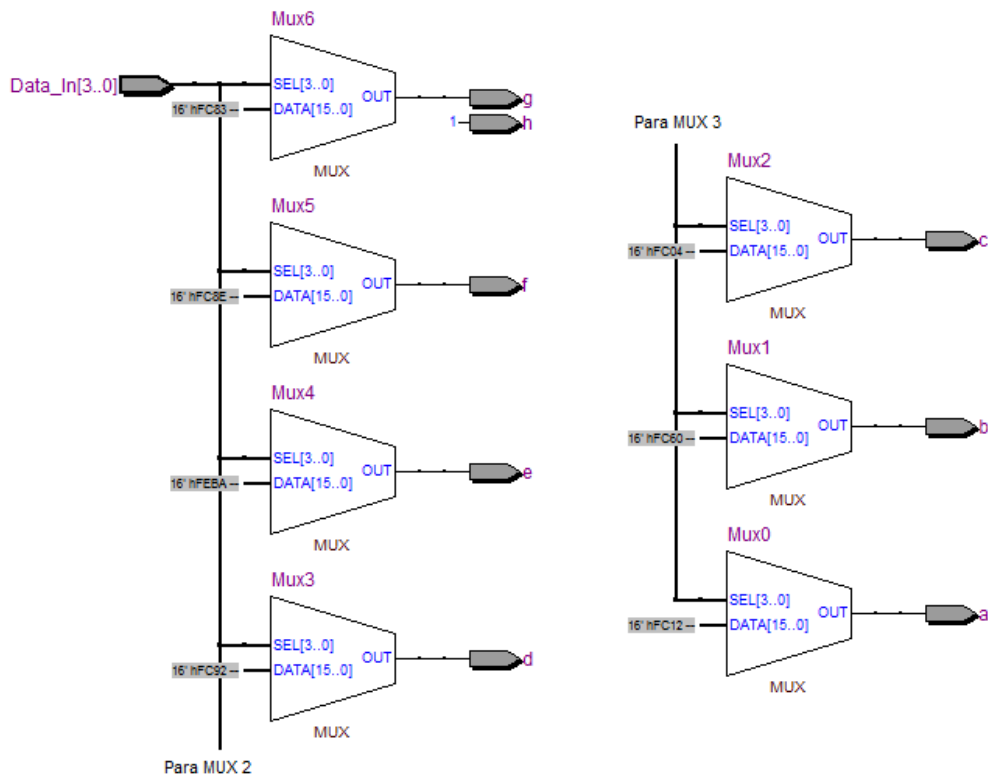


Figura 5.41 - RTL viewer do projeto DECOD7SEG.

16. Crie um bloco lógico para símbolo gráfico do projeto. A Figura 5.42 mostra o bloco lógico criado, arquivo *DECOD7SEG.bsf*, que agora faz parte da biblioteca do software Quartus II.

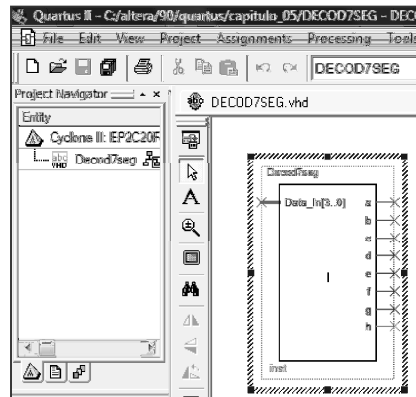


Figura 5.42 - Bloco lógico correspondente ao projeto DECOD7SEG.

17. Para configurar e testar o projeto *DECOD7SEG.Vhd*, no kit DE2, proceda conforme os itens 5.2.2.7 e 5.2.2.9 e a Tabela 5.10, a seguir.

Tabela 5.10 - Designação dos pinos do FPGA.

Nome do sinal	Pinos do FPGA	Nome do sinal	Pinos do FPGA
D0	PIN_N25	a	PIN_AF10
D1	PIN_N26	b	PIN_AB12
D2	PIN_P25	c	PIN_AC12
D3	PIN_AE14	d	PIN_AD11
		e	PIN_AE11
		f	PIN_V14
		g	PIN_V13
		h	VREFGROUP_B1_N1