Vitor Amadeu Souza

FPGA

Aplicações em VHDL
Volume IV

© 2023 by Cerne Tecnologia e Treinamento Ltda.

© 2023 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Maio de 2023

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: *Cerne Tecnologia e Treinamento*

E-mail da Empresa: *cerne@cerne-tec.com.br*

Home Page: *www.cerne-tec.com.br.com.br*

Atendimento ao Consumidor: *sac@cerne-tec.com.br*

Contato com o Autor: *vitor@cerne-tec.com.br*



FEITO NO BRASIL

“Melhor é repelir dez verdades do que admitir uma única falsidade.”

Allan Kardec

Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia têm uma linha completa de aprendizado para os microcontroladores da família PIC, 8051, Holtek, dsPIC, ARM e etc. Veja os detalhes de cada um nas figuras abaixo:



Kit Cerne CPLD

- CPLD ALTERA
- Botões
- Leds
- Gravação ICSP
- Display 7 Segmentos
- E muito mais!

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais e etc.

Visite a nossa página na Internet, no endereço **www.cerne-tec.com.br** e conheça melhor nossos serviços e produtos.

Cerne
Conhecimento para o Desenvolvimento

www.cerne-tec.com.br

Sumário

Capítulo I – Metodologia de desenvolvimento.....6

Capítulo II – Exemplos no VHDL.....7

1. Codificador Decimal → BCD.....7
2. Codificador Decimal → Stibitz ou excesso 3.....9
3. Codificador Decimal → Gray.....12
4. Codificador BCD → Gray.....15
5. Decodificador 74LS139.....18
6. Decodificador 74LS138.....23
7. Decodificador CD4028.....28
8. Decodificador SN7447.....32

Capítulo I
Metodologia de desenvolvimento

A proposta desta obra é continuar apresentando exemplos em VHDL para FPGA. Exemplos com codificador e decodificador são temas abordados neste volume.

Para isso, recomenda-se que o leitor leia as obras dos volumes passados como *FPGA Aplicações em VHDL (2014), FPGA Aplicações em VHDL Volume II (2023) e FPGA Aplicações em VHDL Volume III (2023)* do mesmo autor e editora de modo a continuar seus estudos com eficiência.

Capítulo II
Exemplos no VHDL

1. Codificador Decimal → BCD

Um codificador é aquele que traduz um código qualquer para um resultado binário, como o exemplo proposto que converte decimal para BCD.

ENTRADAS (Decimal)									SAÍDAS (BCD)			
9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**A**	**B**	**C**	**D**
0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	1

Observe pela tabela que dependendo da entrada ativa de 1 a 9 um código BCD específico é obtido como resultado. Caso mais de uma entrada decimal esteja ativa, a saída BCD ficará em 0000.

Acompanhe a seguir o código VHDL que implementa tal solução.

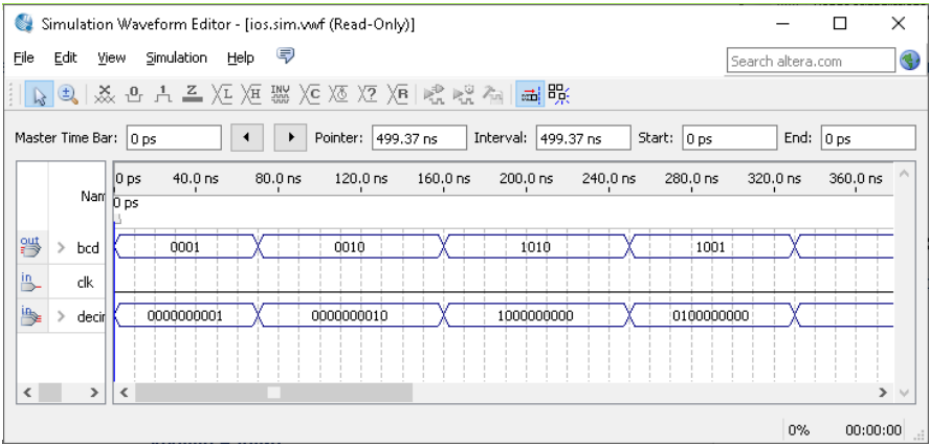
```
library ieee;
use ieee.std_logic_1164.all;

entity ios is
   port (clk     : in std_logic;
         decimal : in std_logic_vector(9 downto 0);
         bcd     : out std_logic_vector(3 downto 0));
end ios;

architecture codificador of ios is
begin
   process(clk)
   begin
          case decimal is
                 when "0000000001" => --1
                        bcd<="0001";
                 when "0000000010" => --2
                        bcd<="0010";
                 when "0000000100" => --3
                        bcd<="0011";
                 when "0000001000" => --4
                        bcd<="0100";
                 when "0000010000" => --5
                        bcd<="0101";
                 when "0000100000" => --6
                        bcd<="0110";
                 when "0001000000" => --7
                        bcd<="0111";
                 when "0010000000" => --8
                        bcd<="1000";
                 when "0100000000" => --9
                        bcd<="1001";
                 when others =>
                        bcd<="0000";
          end case;
```

```
    end process;
end codificador;
```

Observe que dependendo do valor presente no barramento de entrada de 10 vias um valor BCD é dado na saída. Caso o valor presente no barramento seja diferente dos casos acima a opção `when others` será tratada e o barramento BCD fica em 0000. Acompanhe o resultado de uma simulação feita no Quartus II.



2. Codificador Decimal → Stibitz ou excesso 3

Este codificador adiciona o valor 3 ao resultado BCD, funcionando conforme a tabela abaixo:

ENTRADAS (Decimal)									SAÍDAS (Stibitz)			
9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**A**	**B**	**C**	**D**
0	0	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	1	1	0
0	0	0	0	0	1	0	0	0	0	1	1	1
0	0	0	0	1	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	0	0	0	1	0	1	0
0	1	0	0	0	0	0	0	0	1	0	1	1
1	0	0	0	0	0	0	0	0	1	1	0	0

Observe pela tabela que dependendo da entrada ativa de 1 a 9 um código BCD no formato Stibitz ou excesso 3 específico é obtido como resultado. Caso mais de uma entrada decimal esteja ativa, a saída BCD ficará em 0000.

Acompanhe a seguir o código VHDL que implementa tal solução.

```
library ieee;
use ieee.std_logic_1164.all;

entity ios is
   port (clk     : in std_logic;
         decimal : in std_logic_vector(9 downto 0);
         bcd     : out std_logic_vector(3 downto 0));
end ios;
```