Tarefa: Decodificador BCD para display de 7 Segmentos

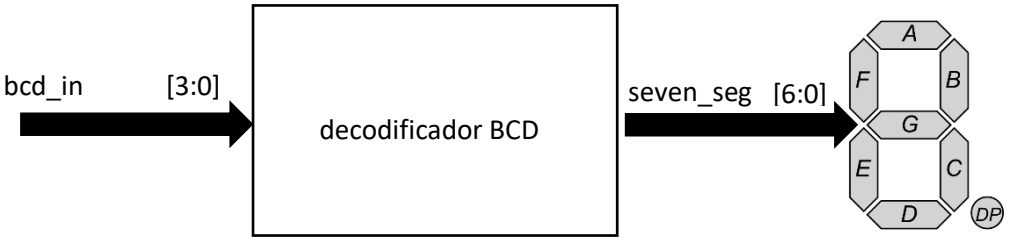
Nome: Mauricio Konrath
Matrícula: 20203635

Uma das formas mais simples para exibir os dígitos alfanuméricos, é o display de sete segmentos. Um decodificador BCD para display de sete segmentos é um circuito digital, usado para receber uma entrada de quatro bits, e gerar sete saídas, que acionam os segmentos corretos do display.

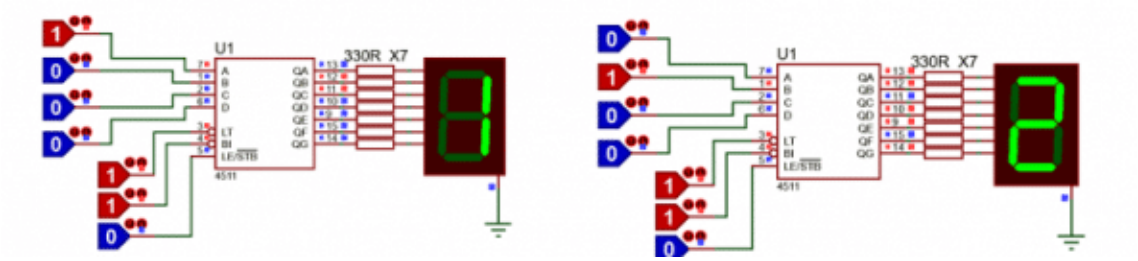
O projeto foi elaborado de forma que cada número de entrada de 0-15 possuísse a maneira correta que cada um ligaram o display de sete segmentos caso fossem implementados.

O Display é um componente bastante usado no mundo da eletroeletrônica, pois torna muito simples a exibição de valores numérico. É empregado na maioria das vezes a partir de circuitos digitais como, circuitos integrados, microcontroladores e outros processos que trabalham em sistema binário, sendo representado por dois níveis lógicos, 0 (nível lógico alto) e 1 (nível lógico baixo).

Abaixo foi criada a arquitetura top level, uma breve demonstração de como funciona, uma entrada declarada como bcd_in de 4 bits e uma saída de 7 bits as quais ligaram cada segmento do display.



0001 em código BCD é equivalente ao número decimal 1 0010 em código BCD é equivalente ao número decimal 2 e assim sucessivamente entre 0 e 9. Também é possivel exibir os número de 10 a 15, porém esses serão exibidos através de letras, onde 10 é igual a A, 11 é igual a b, 12 é igual a C, 13 é d, 14 é E e 15 é F.



Logo abaixo está o código do Decodificador BCD para display de sete segmentos implementado em VHDL assim como o seu testbench.

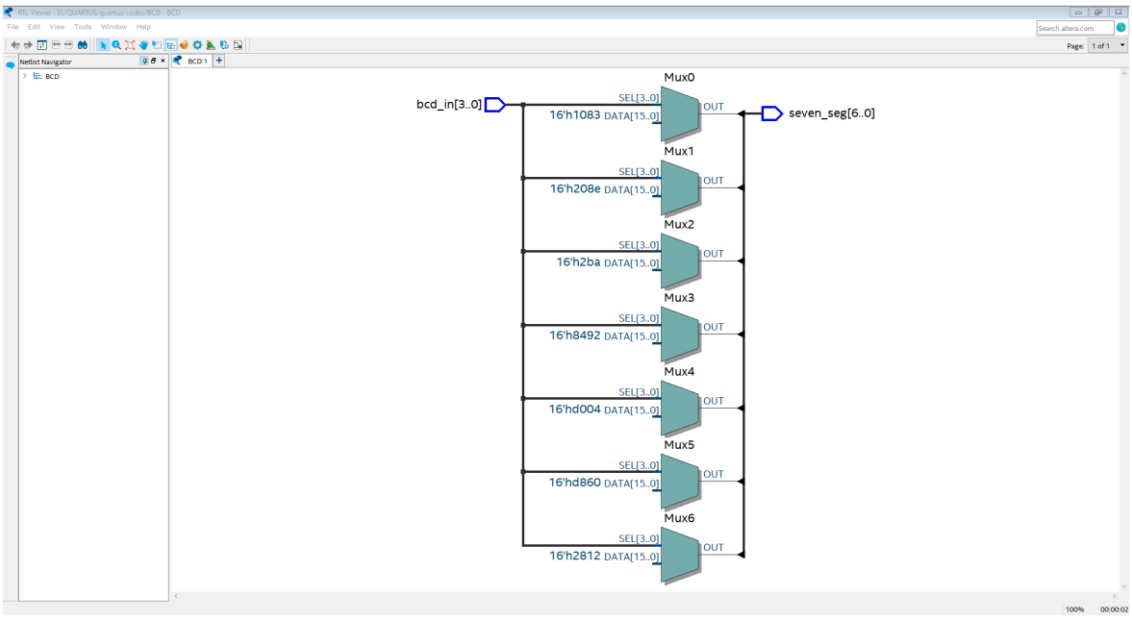
```
library IEEE;
use IEEE.STD_Logic_1164.all;

entity BCD is
    port(
    bcd_in: in std_Logic_Vector(3 downto 0);
    seven_seg: out std_Logic_Vector(6 downto 0)
    );
end BCD;

architecture comportamento of BCD is
begin

    with bcd_in select
        seven_seg <= "1000000" when "0000",
                "1111001" when "0001",
                "0100100" when "0010",
                "0110000" when "0011",
                "0011001" when "0100",
                "0010010" when "0101",
                "0000010" when "0110",
                "1111000" when "0111",
                "0000000" when "1000",
                "0010000" when "1001",
                "0001000" when "1010",
                "0000011" when "1011",
                "1000110" when "1100",
                "0100001" when "1101",
                "0000110" when "1110",
                "0001110" when others;

end comportamento;
```

Decodificador

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

entity BCD_tb is
end BCD_tb;

architecture tb of BCD_tb is
    signal bcd_in: std_Logic_Vector(3 downto 0);
    signal seven_seg: std_Logic_Vector(6 downto 0);

    component BCD
        port(
            bcd_in: in std_Logic_Vector(3 downto 0);
            seven_seg: out std_Logic_Vector(6 downto 0)
            );
    end component;

begin

        MD: BCD port map (bcd_in, seven_seg);
     process
      constant period: time := 20 ns;
      begin
       bcd_in <= "0000"; --0
       wait for period;
       bcd_in <= "0001"; --1
       wait for period;
       bcd_in <= "0010"; --2
       wait for period;
       bcd_in <= "0011"; --3
       wait for period;
       bcd_in <= "0100"; --4
       wait for period;
       bcd_in <= "0101"; --5
       wait for period;
       bcd_in <= "0110"; --6
       wait for period;
       bcd_in <= "0111"; --7
       wait for period;
       bcd_in <= "1000"; --8
       wait for period;
       bcd_in <= "1001"; --9
       wait for period;
       bcd_in <= "1010"; --10
       wait for period;
       bcd_in <= "1011"; --11
       wait for period;
       bcd_in <= "1100"; --12
       wait for period;
       bcd_in <= "1101"; --13
       wait for period;
       bcd_in <= "1110"; --14
       wait for period;
       bcd_in <= "1111"; --15
       wait;
     end process;
end tb;
```

Testbench do Decodificador

Netlist é uma descrição da conectividade de um circuito eletrônico. Consiste em uma lista dos componentes eletrônicos em um circuito. No caso, o circuito projetado foi criado através de diversos multiplexadores (MUXs), e dessa forma, para cada número de entrada a saída exibirá uma resposta. Os multiplexadores são usados para fazer uma seleção na entrada e que deve ir para a saída, selecionando apenas um valor.



Foi utilizado para o projeto a versão do Quartus Prime 20.1, no dispositivo Cyclone IV GX EP4CGX15BF14C6, com um total de 7 Logic Elements e 11 Total Pins. Podemos ver o resultado da simulação com o atraso na imagem abaixo.