Projeto com Circuitos Reconfiguráveis

Projeto lógico combinacional Multiplexadores, decodificadores e comparadores

Prof. Daniel M. Muñoz Arboleda

FGA – UnB

Descrição VHDL de multiplexadores (exemplo 1)

```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.all;
3
4   entity mux1 is port(
5       a, b, sel: in std_logic;
6       s: out std_logic);
7   end mux1;
8
9   architecture comportamental of mux1 is
10  begin
11      with sel select
12          s <= a when '0',
13               b when others;
14
15  end comportamental;
16
```

Descrição VHDL de multiplexadores (exemplo 2)

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  entity mux2 is port(
5      a, b: in bit_vector (7 downto 0);
6      sel: in std_logic;
7      s: out bit_vector (7 downto 0));
8  end mux2;
9
10 architecture comportamental of mux2 is
11 BEGIN
12     with sel select
13         s <= a when '0',
14              b when '1';
15
16 END comportamental;
17
```

Descrição VHDL de multiplexadores (exemplo 3)

```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.all;
3
4   entity mux3 is port(
5       a, b, c, d: in bit_vector (7 downto 0);
6       sel: in bit_vector (1 downto 0);
7       s: out bit_vector (7 downto 0));
8   end mux3;
9
10  architecture comportamental of mux3 is
11  BEGIN
12      with sel select
13          s <= a when "00",
14               b when "01",
15               c when "10",
16               d when others;
17
18  END comportamental;
19
```

Descrição VHDL de multiplexadores (exemplo 4)

```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.all;
3
4   entity mux4 is port(
5       a, b, c, d: in bit_vector (7 downto 0);
6       sel: in bit_vector (1 downto 0);
7       s: out bit_vector (7 downto 0));
8   end mux4;
9
10  architecture comportamental of mux4 is
11  BEGIN
12          s <= a when sel = "00" else
13               b when sel = "01" else
14               c when sel = "10" else
15               d;
16
17  END comportamental;
18
```

Descrição VHDL de multiplexadores (exemplo 5)

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  entity mux5 is port(
5      a, b, c, d: in bit_vector (7 downto 0);
6      sel: in bit_vector (1 downto 0);
7      s: out bit_vector (7 downto 0);
8      s2 : out bit_vector(7 downto 0));
9  end mux5;
10
11 architecture comportamental of mux5 is
12
13 BEGIN
14     process (a,b,c,d,sel)
15     begin
16       if sel = "00"  then
17             s <= a;
18       elsif sel = "01" then
19             s <= b;
20       elsif sel = "10" then
21             s <= c;
22       else
23             s <= d;
24       end if;
25     end process;
26
```

```
27 process (sel)
28   s2<= c-d;
29 end process;
30 END comportamental;
31
```

Descrição VHDL de um decodificador binário a 7 segmentos

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  USE work.data_types.all;
4
5  entity mux_7_seg is port(
6      a, b   : in std_logic_vector (MAX downto 0);
7      sel    : in std_logic;
8      SEG_A, SEG_B, SEG_C, SEG_D, SEG_E, SEG_F, SEG_G, DP: out std_logic);
9  end mux_7_seg;
10
11 architecture comportamental of mux_7_seg is
12 -------signal definitions-----------------------------
13 signal mux_output: std_logic_vector (MAX downto 0);
14 signal RES : std_logic_vector(Segment_Number downto 0);
15 ------------------------------------------------------
16 BEGIN
17     process (a,b,sel)
18     begin
19       if sel = '0'  then
20             mux_output <= a;
21       elsif sel = '1' then
22             mux_output <= b;
23       end if;
24     end process;
25
```

Descrição VHDL de um decodificador binário a 7 segmentos

```
26  process(mux_output)
27  begin
28      case mux_output is
29          when "0000" => RES <= "11000000";
30          when "0001" => RES <= "11111001";
31          when "0010" => RES <= "10100100";
32          when "0011" => RES <= "10110000";
33          when "0100" => RES <= "10011001";
34          when "0101" => RES <= "10010010";
35          when "0110" => RES <= "10000010";
36          when "0111" => RES <= "11111000";
37          when "1000" => RES <= "10000000";
38          when "1001" => RES <= "10010000";
39          when "1010" => RES <= "10001000";
40          when "1011" => RES <= "10000011";
41          when "1100" => RES <= "11000110";
42          when "1101" => RES <= "10100001";
43          when "1110" => RES <= "10000110";
44          when "1111" => RES <= "10001110";
45          when others => RES <= "11111111";
46      end case;
47  end process;
48
49  SEG_A <= RES(0);
50  SEG_B <= RES(1);
51  SEG_C <= RES(2);
52  SEG_D <= RES(3);
53  SEG_E <= RES(4);
54  SEG_F <= RES(5);
55  SEG_G <= RES(6);
56  DP <= RES(7);
57  END comportamental;
```

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  Package data_types is
5      constant Max : integer := 3;
6      constant Segment_Number: integer := 7;
7  end data_types;
8  ------------------------------------------------
```

Decodificador binário a 7 segmentos por componentes

```
3   library IEEE;
4   use ieee.std_logic_1164.all;
5   use ieee.std_logic_unsigned.all;
6   USE work.data_types.all;
7
8   entity COMPONENTES is
9   port    (
10      data_1, data_2  : in std_logic_vector (MAX downto 0);
11      control   : in std_logic;
12      D1A, D1B, D1C, D1D, D1E, D1F, D1G, DP1 : out std_logic);
13
14  end COMPONENTES;
15
16  architecture ESTRUTURA of COMPONENTES is
17  ---------------signals declaration--------------
18  signal mux_output : std_logic_vector(3 downto 0);
19
20  ------------Components declaration---------------
21
22  component Mux5
23  port (
24      a, b  : in std_logic_vector (MAX downto 0);
25      sel   : in std_logic;
26      s: out std_logic_vector (MAX downto 0)
27  );
28  end component;
29
30  component BIN_7SEG
31  port    (
32      BIN : in std_logic_vector(3 downto 0);
33      SEG_A, SEG_B, SEG_C, SEG_D, SEG_E, SEG_F, SEG_G, DP : out std_logic
34      );
35  end component;
```

Decodificador binário a 7 segmentos por componentes

```
37  --------------Maping the instances---------------
38  begin
39  Mux : Mux5 port map (a => data_1,
40                       b => data_2,
41                       sel => control,
42                       s => mux_output);
43
44  conv1 : BIN_7SEG port map (BIN => mux_output,
45                             SEG_A => D1A,
46                             SEG_B => D1B,
47                             SEG_C => D1C,
48                             SEG_D => D1D,
49                             SEG_E => D1E,
50                             SEG_F => D1F,
51                             DP    =>DP1);
52  end ESTRUTURA;
53
```

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  Package data_types is
5     constant Max : integer := 3;
6     constant Segment_Number: integer := 7;
7  end data_types;
8  ------------------------------------------------
```

Descrição VHDL de um comparador

```
25  entity comparador is
26  port (
27      A   : in std_logic_vector(3 downto 0);
28      B   : in std_logic_vector(3 downto 0);
29      AlB : out std_logic;
30      AeB : out std_logic;
31      AhB : out std_logic
32  );
33  end comparador;
34
35  architecture behavioral of comparador is
36  begin -- architecture body
37
38      process(A,B)
39      begin
40          if A=B then
41              AlB <= '0';
42              AeB <= '1';
43              AhB <= '0';
44          elsif A<B then
45              AlB <= '1';
46              AeB <= '0';
47              AhB <= '0';
48          else
49              AlB <= '0';
50              AeB <= '0';
51              AhB <= '1';
52          end if;
53      end process;
54
55  end behavioral;
```