Multiplexer Exercises

23 de março de 2021 12:53

74153 TRUTH TABLE

1GN	B	A	1Y
1	X	X	0
0	0	0	C0
0	0	1	C1
0	1	0	C2
0	1	1	C3

Exercise 2

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity Mux4_1 is
  port(sel    : in  std_logic_vector(1 downto 0) ;
       inputs : in  std_logic_vector(3 downto 0);
       muxOut : out std_logic);
end Mux4_1;
architecture Mux_Proc_case of Mux4_1 is
begin
  process(sel, inputs)
  begin
    case sel is
          when "00" => muxOut <= inputs(0);
          when "01" => muxOut <= inputs(1);
          when "10" => muxOut <= inputs(2);
          when "11" => muxOut <= inputs(3);
          when others => muxOut <= 'X';
      end case;
  end process;
end Mux_Proc_case;
```

Exercise 3

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity Mux2_1 is
  port(selector    : in  std_logic ;
       input0 : in  std_logic_vector(3 downto 0);
       input1 : in  std_logic_vector(3 downto 0);
       finalOut :out std_logic_vector(3 downto 0););
end Mux2_1;
architecture Behavioral of Mux2_1 is
begin
  finalout <= input1 when(selector= '1') else
              input1;
end Behavioral;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity Mux8_1 is
  port(select1    : in  std_logic;
       select2    : in  std_logic;
       select3    : in  std_logic;
       a, b, c, d, e, f, g, h: in  std_logic_vector(3 downto 0);
       i, j, k, l, m, n : buffer  std_logic_vector(3 downto 0);
       s : out std_logic);
end Mux8_1;
architecture Structural of Mux8_1 is
begin
  MUX0: entity work.Mux2_1
             port map (selector => select1,
                       input0 => a,
                       input1 => b,
                       finalout => i);

MUX1: entity work.Mux2_1
             port map (selector => select1,
                       input0 => c,
                       input1 => d,
                       finalout => j);
MUX2: entity work.Mux2_1
             port map (selector => select1,
                       input0 => e,
                       input1 => f,
                       finalout => k);
MUX3: entity work.Mux2_1
             port map (selector => select1,
                       input0 => g,
                       input1 => h,
                       finalout => l);
MUX4: entity work.Mux2_1
             port map (selector => select2,
                       input0 => i,
                       input1 => j,
                       finalout => m);
MUX5: entity work.Mux2_1
             port map (selector => select2,
                       input0 => k,
                       input1 => l,
                       finalout => n);
MUX6: entity work.Mux2_1
             port map (selector => select3,
                       input0 => m,
                       input1 => n,
                       finalout => s);

      end Structural;
```