



# Microeletrônica I

LAB I

# Projeto I

- Utilizando a ferramenta ISE, faça o tutorial apresentado na aula desta semana (em anexo) que implementa o circuito composto por um multiplexador e um flip-flop, mostrado na figura abaixo:

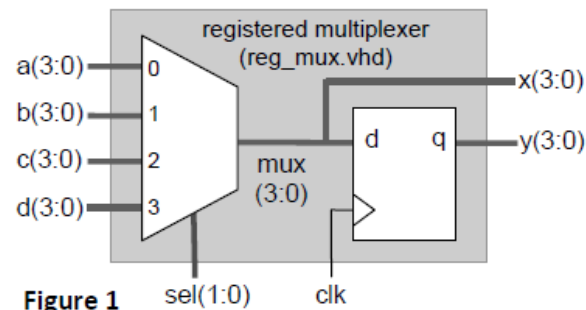


Figure 1

# Projeto I

- No passo 2 do tutorial, modifique as configurações sobre a FPGA da seguinte maneira:
  - Family: Spartan3E
  - Device: XC3S500E ou XC3S1200E
  - Package: FG320
- Para o projeto I, os passos 6 (*simulating de circuit with ModelSim*), 7 (*making pin assignments*) e 8 (*physical implementation*) do tutorial devem ser desconsiderados.

# Projeto I

```
1  -----
2  LIBRARY ieee;
3  USE ieee.std_logic_1164.all;
4  -----
5  ENTITY reg_mux IS
6      PORT (a, b, c, d: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
7            sel: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
8            clk: IN STD_LOGIC;
9            x, y: OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
10 END ENTITY;
11 -----
12 ARCHITECTURE reg_mux OF reg_mux IS
13     SIGNAL mux: STD_LOGIC_VECTOR(3 DOWNTO 0);
14 BEGIN
15     mux <= a WHEN sel="00" ELSE
16           b WHEN sel="01" ELSE
17           c WHEN sel="10" ELSE
18           d;
19     x <= mux;
20     PROCESS (clk)
21     BEGIN
22         IF (clk'EVENT AND clk='1') THEN
23             y <= mux;
24         END IF;
25     END PROCESS;
26 END ARCHITECTURE;
27 -----
```

reg\_mux.vhd

# Projeto I

```
1  -----
2  LIBRARY ieee;
3  USE ieee.std_logic_1164.all;
4  -----
5  ENTITY reg_mux_tb IS
6  END ENTITY;
7  -----
8  ARCHITECTURE testbench OF reg_mux_tb IS
9  ----DUT declaration:-----
10     COMPONENT reg_mux IS
11         PORT (a, b, c, d: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
12              sel: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
13              clk: IN STD_LOGIC;
14              x, y: OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
15     END COMPONENT;
16     ----Signal declarations:----
17     SIGNAL a_tb: STD_LOGIC_VECTOR(3 DOWNTO 0) := "0010";
18     SIGNAL b_tb: STD_LOGIC_VECTOR(3 DOWNTO 0) := "0100";
19     SIGNAL c_tb: STD_LOGIC_VECTOR(3 DOWNTO 0) := "0110";
20     SIGNAL d_tb: STD_LOGIC_VECTOR(3 DOWNTO 0) := "1000";
21     SIGNAL sel_tb: STD_LOGIC_VECTOR(1 DOWNTO 0) := "00";
22     SIGNAL clk_tb: STD_LOGIC := '0';
23     SIGNAL x_tb: STD_LOGIC_VECTOR(3 DOWNTO 0);
24     SIGNAL y_tb: STD_LOGIC_VECTOR(3 DOWNTO 0);
25 BEGIN
26     ----DUT instantiation:-----
27     dut: reg_mux PORT MAP (
28         a => a_tb,
29         b => b_tb,
30         c => c_tb,
31         d => d_tb,
32         clk => clk_tb,
33         sel => sel_tb,
34         x => x_tb,
35         y => y_tb);
36     ----Stimuli generation:-----
37     clk_tb <= NOT clk_tb AFTER 40ns;
38     a_tb <= "0011" AFTER 80ns, "0000" AFTER 640ns;
39     b_tb <= "0101" AFTER 240ns;
40     c_tb <= "0111" AFTER 400ns;
41     d_tb <= "1001" AFTER 560ns;
42     sel_tb <= "01" AFTER 160ns,
43              "10" AFTER 320ns,
44              "11" AFTER 480ns,
45              "00" AFTER 640ns;
46 END ARCHITECTURE;
47 -----
```

reg\_mux\_tb.vhd

# Projeto 2

- Modifique o projeto 1 alterando a dimensão dos sinais de entrada ( $a$ ,  $b$ ,  $c$ ,  $d$ ), de saída ( $x$ ,  $y$ ) e do sinal *mux* (que são vetores de 4 bits) para um único bit (escalar).
- Para o projeto 2, o passo 6 (*simulating de circuit with ModelSim*) do tutorial deve ser desconsiderado.
- Para implementação no kit Nexys 2, realize o mapeamento dos pinos. No passo 7 do tutorial, desconsidere os item (b) e (e).
  - Entradas:
    - $a = G18$
    - $b = H18$
    - $c = K18$
    - $d = K17$
    - $sel(1) = L14$
    - $sel(0) = L13$
    - $clk = B8$
  - Saídas:
    - $x = J14$
    - $y = J15$

# Apresentação dos Resultados

- Implementar os projetos na ferramenta ISE e apresentar o funcionamento de cada projeto no kit Nexys2 (arquivo .BIT) durante a aula de laboratório (projeto I apenas simulação).
- Elaborar um relatório técnico incluindo a descrição e o desenvolvimento de cada projeto, assim como os esquemáticos RTL obtidos após a síntese. As simulações para diferentes combinações das entradas também devem ser apresentadas (todos os resultados mostrados devem ser comentados e justificados). Lembre-se de incluir no relatório os códigos do design e do testbench e resumo dos recursos lógicos utilizados.
- O trabalho deverá ser entregue pela UFPR Virtual (faça um arquivo compactado incluindo os projetos no ISE e o relatório).