

Relatório Prática 07 – 06/10/2023
Arthur Souza/João Paulo – PN1

Iniciou a prática criando uma entidade nomeado pseudo_mux no software QUARTUS II 13.0, especificando qual chip está sendo utilizado, no caso o EP2C35F672C6.

A partir disso, adicionou-se o código disponibilizado no sistema e implementado uma arquitetura comportamental para análise e compilação dos dados:

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

entity pseudo_mux is
    port (
        CLOCK      : in      std_logic; -- clock input
        RESET       : in      std_logic; -- reset input
        S           : in      std_logic; -- control input
        A,B,C,D     : in      std_logic; -- data inputs
        Q           : out     std_logic -- data output
    );
end pseudo_mux;

architecture arch of pseudo_mux is
    type estado is (s0, s1, s2, s3);
    signal est_atual, est_futuro : estado;
begin
    process(CLOCK, RESET) is
        begin
            if(RESET= '1') then
                est_atual<=s0;
            elsif(rising_edge(CLOCK)) then
                est_atual<=est_futuro;
            end if;
        end process;
        process(S, est_atual, A, B, C, D) is
            begin
                case est_atual is
                    when s0 =>
                        Q<=A;
                        if(S='1') then
                            est_futuro<=s1;
                        else
                            est_futuro<=s0;
                        end if;
                    when s1 =>
                        Q<=B;
                        if(S='1') then
                            est_futuro<=s2;
                        else
                            est_futuro<=s1;
                        end if;
                end case;
            end process;
        end process;
    end arch;
```

```

when s2 =>
    Q<=C;
    if(S='1') then
        est_futuro<=s3;
    else
        est_futuro<=s2;
    end if;
when s3 =>
    Q<=D;
    if(S='1') then
        est_futuro<=s0;
    else
        est_futuro<=s3;
    end if;
end case;
end process;
end arch;

```

Simulado e atestado ausência de erros, verificou-se a esquematização do código em circuito pela opção RTL viewer e Technology Map Viewer. Segue abaixo as imagens:

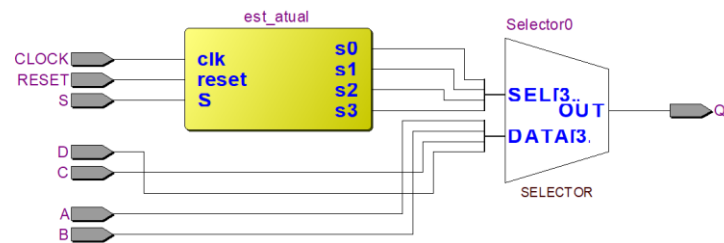


Figura 1: Circuito Pseudo Mux

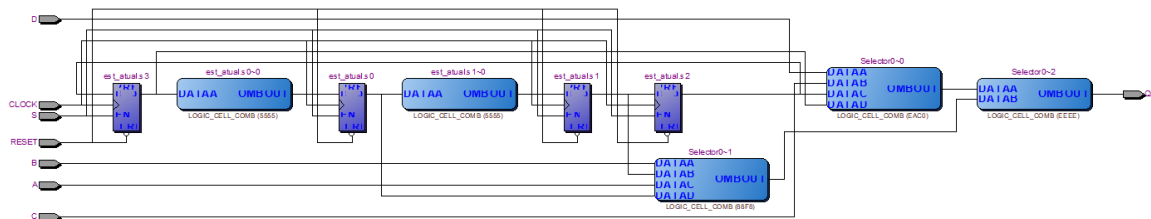


Figura 2: Diagrama do circuito Pseudo Mux

Seguiu-se para a compilação do testbench criado, este chamado de tb_mean_4_clocks, e que irá definir os testes do projeto. Ele está descrito abaixo:

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use ieee.numeric_std.all;

entity tb_pseudo_mux is
end tb_pseudo_mux;

architecture teste of tb_pseudo_mux is

component pseudo_mux is
    port (
        CLOCK      : in      std_logic; -- clock input
        RESET       : in      std_logic; -- reset input
        S           : in      std_logic; -- control input
        A,B,C,D     : in      std_logic; -- data inputs
        Q           : out     std_logic -- data output
    );
end component;

signal fio_clk : std_logic := '0';
signal fio_res : std_logic ;
signal fio_s : std_logic ;
signal fio_a,fio_b,fio_c,fio_d : std_logic;
signal fio_q : std_logic := '0';
begin

    -- Note que o componente é instanciado com apenas 4 bits nas entradas para facilitar
    a simulação:
    instancia_pseudo_mux: pseudo_mux port map(CLOCK=>fio_clk,
    RESET=>fio_res,A=>fio_a,B=>fio_b,C=>fio_c,D=>fio_d,S=>fio_s,Q=>fio_q);

    -- Dados de entrada de 4 bits são expressos em "hexadecimal" usando "x":
    fio_clk <= not fio_clk after 5ns;
    fio_res <= '1', '0' after 5ns;
    fio_s <= '0', '1' after 20ns, '0' after 30ns, '1' after 50ns, '0' after 60ns, '1' after 80ns,
    '0' after 90ns;
    fio_a <= '0', '1' after 10ns, '0' after 20ns;
    fio_b <= '0', '1' after 20ns, '0' after 50ns;
    fio_c <= '0', '1' after 50ns, '0' after 80ns;
    fio_d <= '0', '1' after 80ns, '0' after 90ns, '1' after 100ns, '0' after 110ns;
end teste;
```

Com ele, podemos simular o funcionamento do circuito através do software MULTISIM. Basta apenas indica o arquivo como testbench em simulation e começar a simulação em RTL. Irá abrir um gráfico de sinais com valores determinados pelo testbench e irá auxiliar na verificação da lógica.

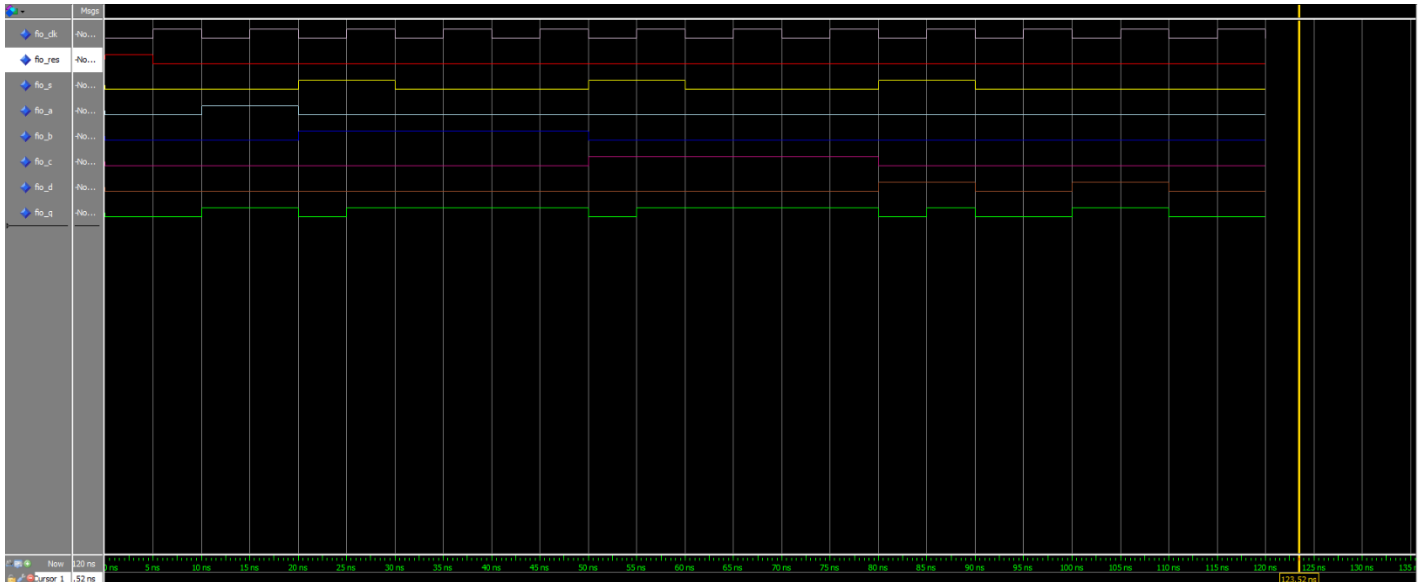


Figura 3: Simulação Pseudo Mux no Multisim

Foi utilizado o arquivo csv do comparador para facilitar a pinagem desse projeto.

$A \leq PIN_N25$

$B \leq PIN_N26$

$C \leq PIN_P25$

$D \leq PIN_AE14$

$CLOCK \leq PIN_AC13$

$RESET \leq PIN_C13$

$S \leq PIN_B13$

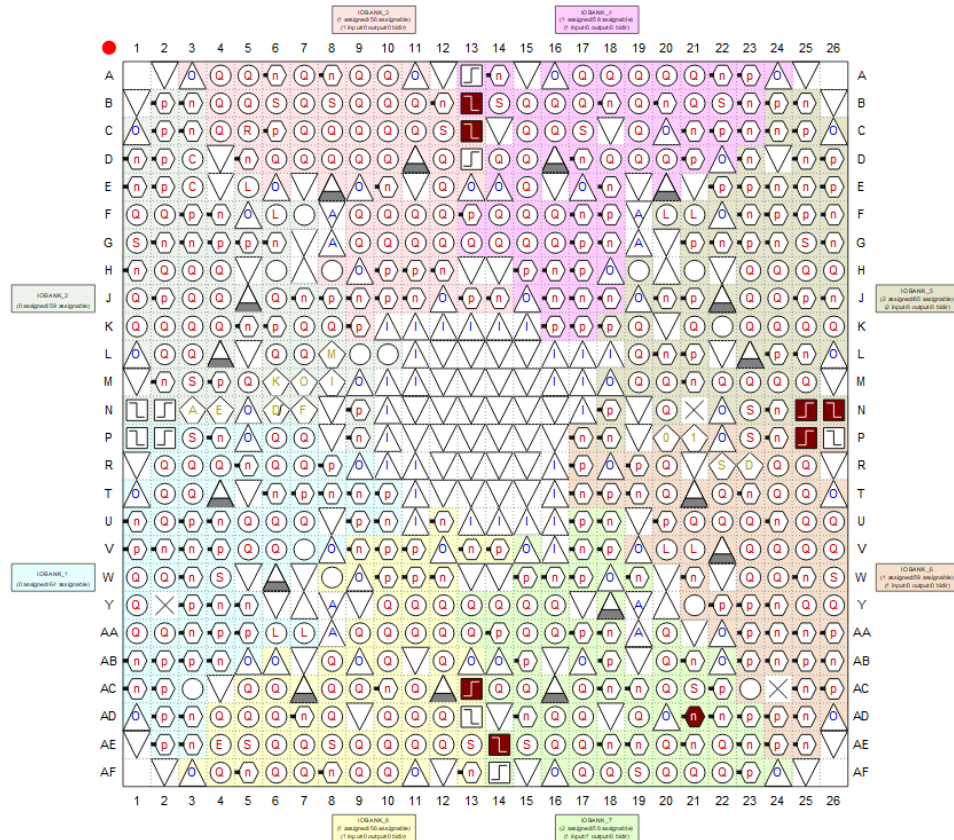


Figura 4: Pin Planner

Simulando novamente para verificação dos pinos, podemos enviar o projeto para o FPGA usando a função PROGRAMMER e o usb blaster.

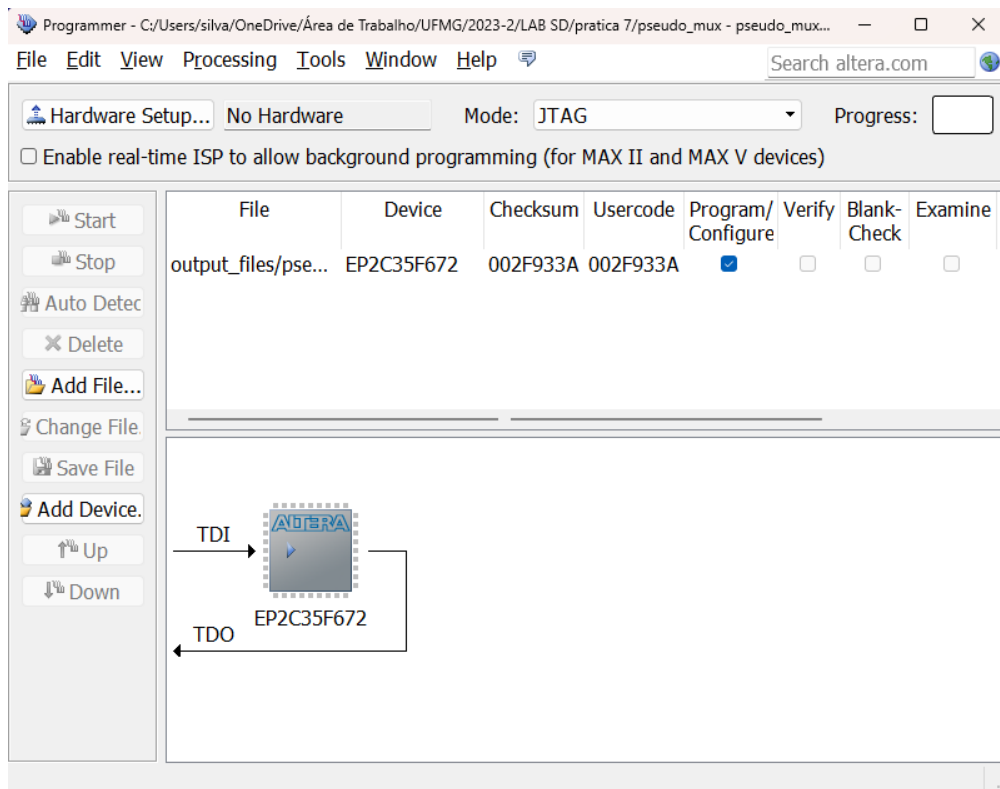


Figura 5: Programmer