# Relatório Prática 06 – 29/09/2023

# Arthur Souza/João Paulo – PN1

Iniciou a prática criando uma entidade nomeado mean\_4\_clocks 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 que segue abaixo:

***library ieee;***

***use ieee.std\_logic\_1164.all;***

***use ieee.numeric\_std.all;***

***entity mean\_4\_clocks is***

***generic (***

***W : integer := 32***

***);***

***port (***

***CLK : in std\_logic;***

***RESET : in std\_logic;***

***INPUT : in std\_logic\_vector(W - 1 downto 0);***

***OUTPUT : out std\_logic\_vector(W - 1 downto 0)***

***);***

***end mean\_4\_clocks;***

***architecture arch of mean\_4\_clocks is***

***begin***

***process(CLK, RESET) is***

***variable var1 : unsigned(W - 1 downto 0);***

***variable var2 : unsigned(W - 1 downto 0);***

***variable var3 : unsigned(W - 1 downto 0);***

***variable var4 : unsigned(W - 1 downto 0);***

***begin***

***if (RESET = '1') then***

***var1 := to\_unsigned(0,W);***

***var2 := to\_unsigned(0,W);***

***var3 := to\_unsigned(0,W);***

***var4 := to\_unsigned(0,W);***

***elsif (rising\_edge(CLK)) then***

***var1 := unsigned("00" & INPUT(W-1 downto 2));***

***var2 := var1;***

***var3 := var2;***

***var4 := var3;***

***end if;***

***OUTPUT <= std\_logic\_vector(var1 + var2 + var3 + var4);***

***end process;***

***end arch;***

Ele apresenta 2 erros fundamentais, o primeiro sendo o uso de variables, que liga todos registradores ao input, e o segundo diz respeito a precisão do calculo e a perda de informações. Ambos foram solucionados com o código abaixo:

***library ieee;***

***use ieee.std\_logic\_1164.all;***

***use ieee.numeric\_std.all;***

***entity mean\_4\_clocks is***

***generic (***

***W : integer := 32***

***);***

***port (***

***CLK : in std\_logic;***

***RESET : in std\_logic;***

***INPUT : in std\_logic\_vector(W - 1 downto 0);***

***OUTPUT : out std\_logic\_vector(W - 1 downto 0)***

***);***

***end mean\_4\_clocks;***

***architecture arch of mean\_4\_clocks is***

***signal var1, var2, var3, var4 : unsigned(W - 1 downto 0);***

***signal soma : unsigned(W+1 downto 0);***

***begin***

***process(CLK, RESET) is***

***begin***

***if (RESET = '1') then***

***var1 <= to\_unsigned(0,W);***

***var2 <= to\_unsigned(0,W);***

***var3 <= to\_unsigned(0,W);***

***var4 <= to\_unsigned(0,W);***

***elsif (rising\_edge(CLK)) then***

***var1 <= unsigned(INPUT(W-1 downto 0));***

***var2 <= var1;***

***var3 <= var2;***

***var4 <= var3;***

***end if;***

***soma<=("00"&var1) + ("00"&var2) + ("00"&var3) + ("00"&var4);***

***OUTPUT <= std\_logic\_vector(soma(W+1 downto 2));***

***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:

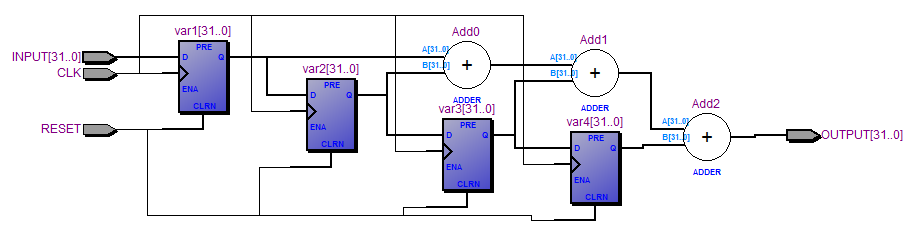


Figura : Circuito Mean\_4\_Clockr

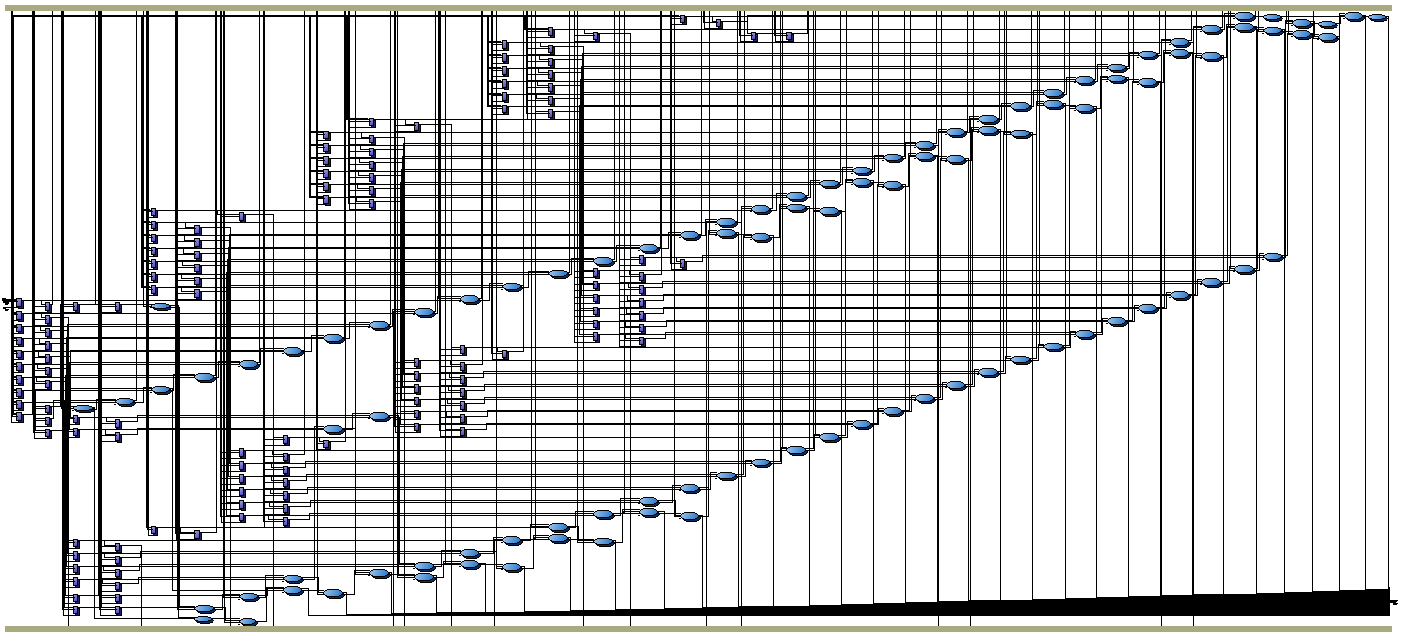


Figura : Diagrama do circuito Mean\_4\_Clock

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\_mean\_4\_clocks is***

***end tb\_mean\_4\_clocks;***

***architecture teste of tb\_mean\_4\_clocks is***

***component mean\_4\_clocks is***

***generic (***

***W : natural := 32***

***);***

***port (***

***CLK : in std\_logic;***

***RESET : in std\_logic;***

***INPUT : in std\_logic\_vector(W - 1 downto 0);***

***OUTPUT : out std\_logic\_vector(W - 1 downto 0)***

***);***

***end component;***

***signal fio\_clk: std\_logic:='0';***

***signal fio\_R: std\_logic;***

***signal fio\_I, fio\_O: std\_logic\_vector(3 downto 0);***

***begin***

***instancia\_mean4clocks: mean\_4\_clocks generic map (W=>4) port map(CLK=>fio\_clk, RESET=>fio\_R, INPUT=>fio\_I, OUTPUT=>fio\_O);***

***-- x nas próximas linhas: os vetores de bits estão expressos em base hexadecimal***

***fio\_clk <=not fio\_clk after 25ns;***

***fio\_R <= '1', '0' after 5ns;***

***fio\_I <= x"1", x"f" after 300ns;***

***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.

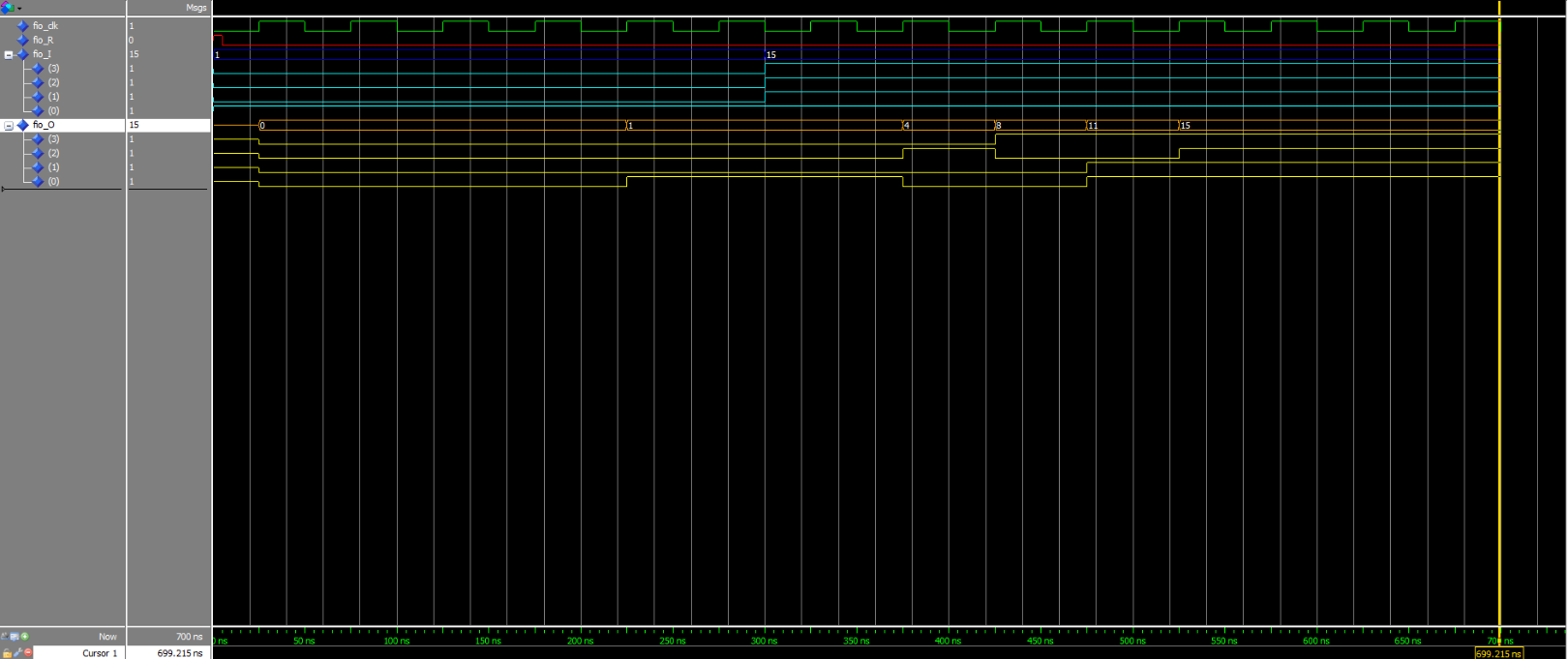


Figura : Simulação Mean\_4\_Clocks no Multisim

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

***INPUT[3] <=*** ***PIN\_N25***

***INPUT[2] <= PIN\_N26***

***INPUT[1] <= PIN\_P25***

***INPUT[0] <= PIN\_AE14***

***CLOCK <= PIN\_AC13***

***RESET <= PIN\_C13***

***OUTPUT[3] <= PIN\_AD21***

***OUTPUT 2] <= PIN\_AC21***

***OUTPUT [1] <= PIN\_AA14***

***OUTPUT [0] <= PIN\_Y13***

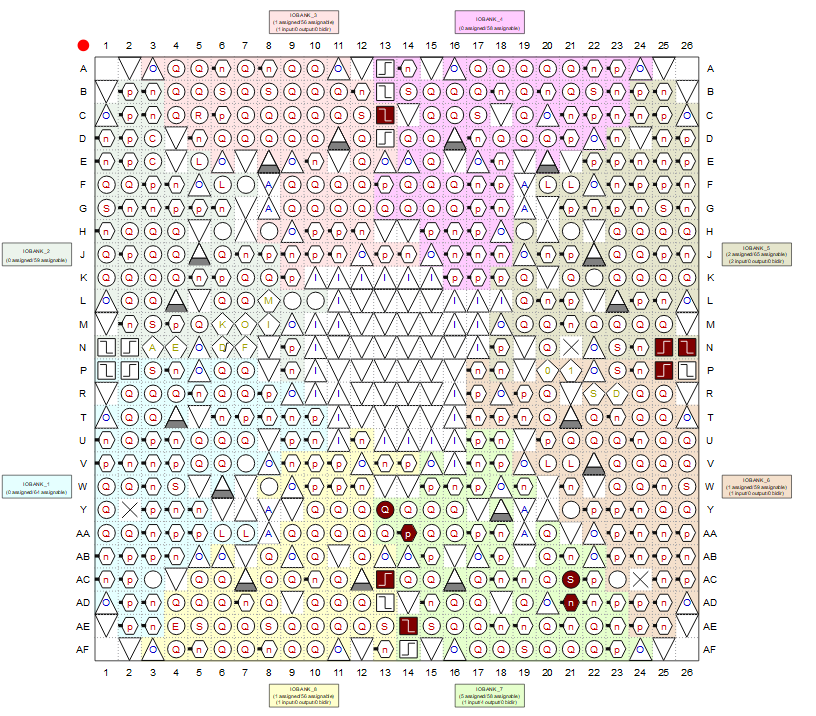


Figura 4: Pin Planner

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

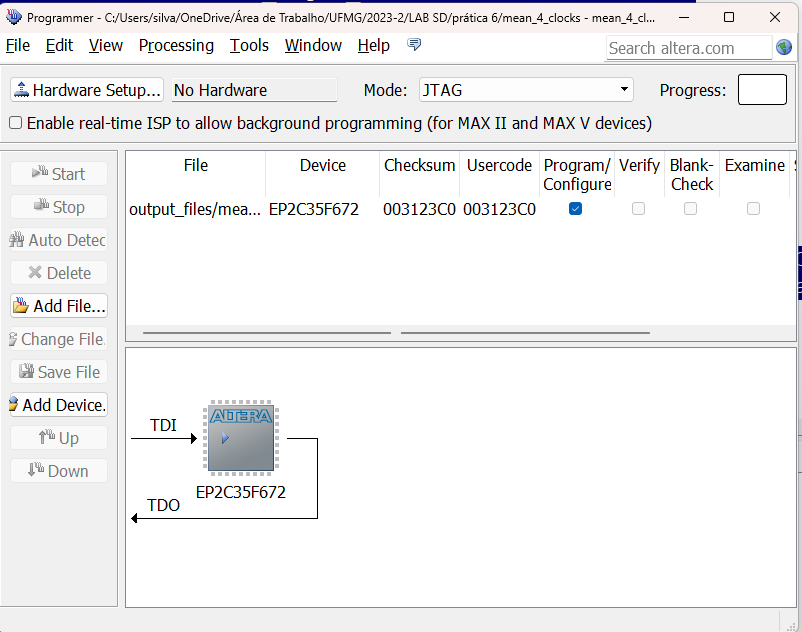


Figura : Programmer