# Relatório Prática 02 – 25/08/2023

# Arthur Souza/João Paulo – PN1

Iniciou a prática criando uma entidade nomeada Door\_opener no software QUARTUS II 13.0, especificando qual chip está sendo utilizado, no caso o EP2C35F672C6.

A partir disso, criou o primeiro arquivo em VHDL de mesmo nome que a entidade e que teria o código disponibilizado no moodle e que segue logo abaixo:

***library ieee;***

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

***entity Door\_opener is***

***port (***

***pino\_c,pino\_h,pino\_p : in std\_logic;***

***pino\_f : out std\_logic***

***);***

***end Door\_opener;***

***architecture dataflow of Door\_opener is***

***begin***

***pino\_f <= not(pino\_c) and (pino\_h or pino\_p);***

***end dataflow;***

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

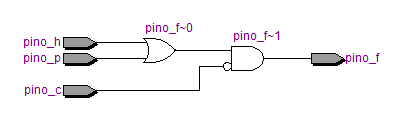


Figura 1: Circuito Door\_opener

Seguiu-se para a criação do testbench, este chamado de tb\_Door\_opener e que irá definir os testes do projeto. Ele está descrito abaixo:

***library IEEE;***

***use IEEE.STD\_LOGIC\_1164.all;***

***entity tb\_Door\_opener is***

***end tb\_Door\_opener;***

***architecture teste of tb\_Door\_opener is***

***component Door\_opener is***

***port (***

***pino\_c,pino\_h,pino\_p : in std\_logic;***

***pino\_f : out std\_logic***

***);***

***end component;***

***signal fio\_c,fio\_h,fio\_p,fio\_f : std\_logic;***

***begin***

***instancia\_door\_opener : Door\_opener port map (pino\_c=>fio\_c, pino\_h=>fio\_h, pino\_p=>fio\_p, pino\_f=>fio\_f);***

***fio\_c<='0','1' after 200 ns, '0' after 400 ns, '1' after 600 ns;***

***fio\_h<='0','1' after 100 ns, '0' after 200 ns, '1' after 300 ns, '0' after 400 ns, '1' after 500 ns;***

***fio\_p<='0','1' after 50 ns, '0' after 100 ns, '1' after 150 ns, '0' after 200 ns, '1' after 250 ns, '0' after 300 ns, '1' after 350 ns, '0' after 400 ns;***

***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.

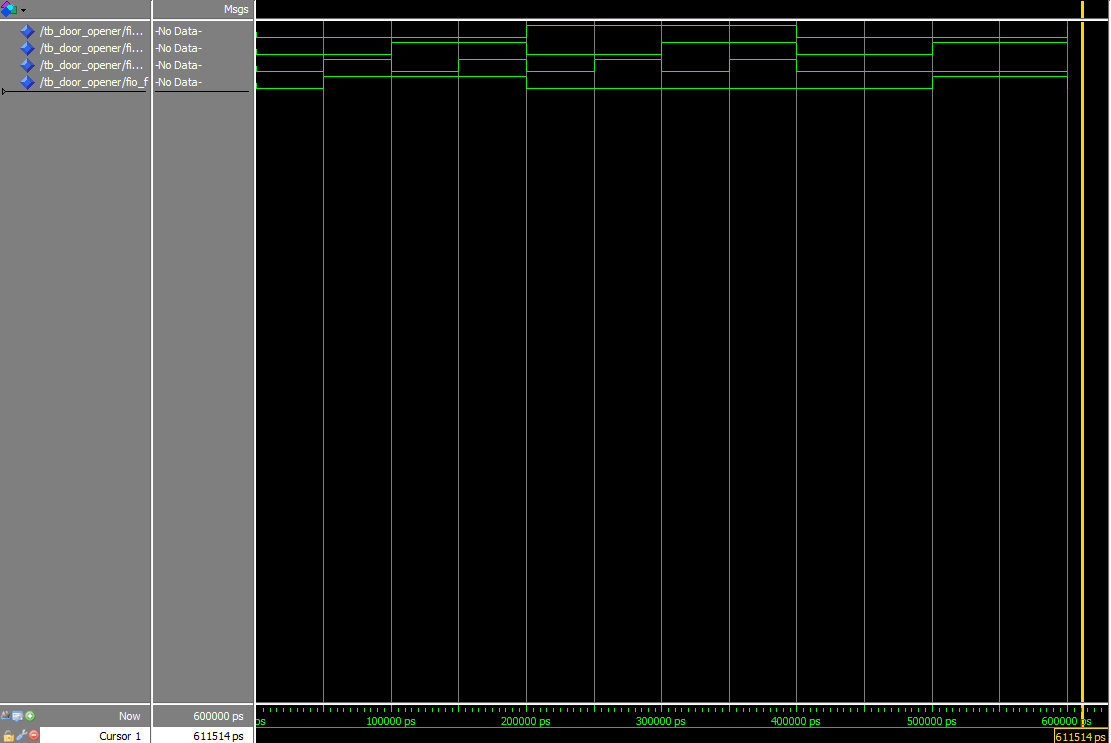


Figura 2: Simulação Door\_opener no Multisim

Caso esteja tudo correto, podemos enviá-lo para o FPGA primeiro definindo a pinagem no PIN PLANNER onde, com ajuda de uma tabela, iremos associar as entradas e saídas do programa com os pinos da placa. No caso, os pinos foram configurados do seguinte modo:

***Pino\_c = PIN\_N25***

***Pino\_f = PIN\_AE23***

***Pino\_h = PIN\_N26***

***Pino\_p = PIN\_P25***

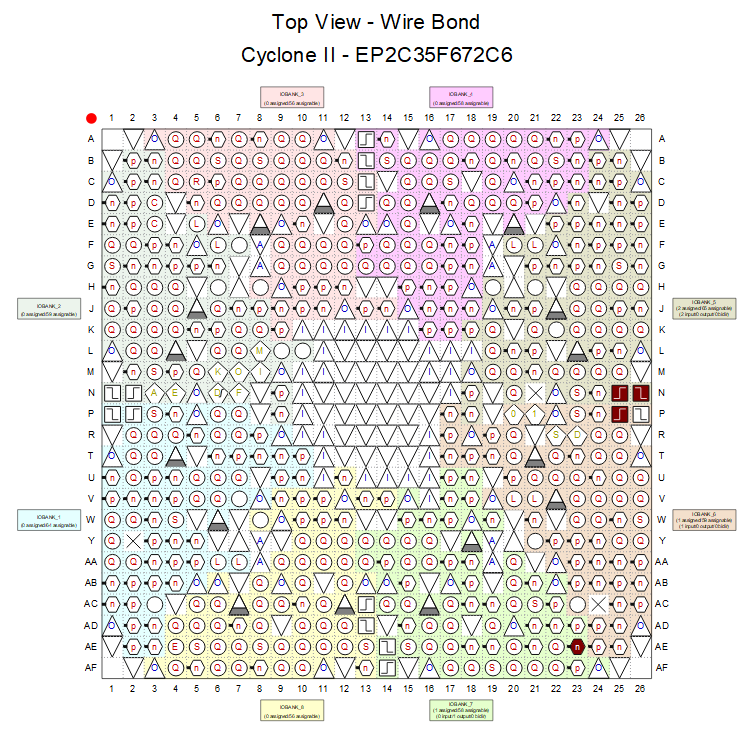


Figura 3: Pin Planner

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

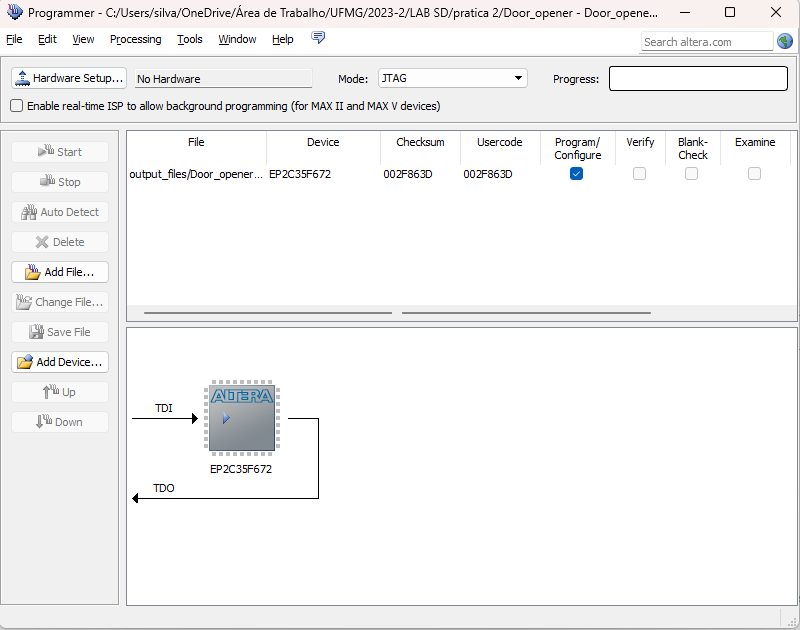


Figura 4: Programmer