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:

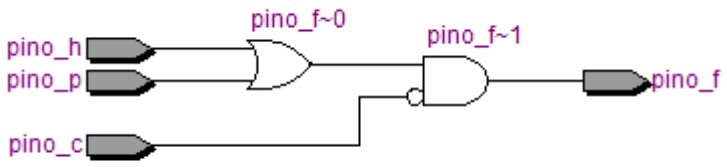

Figura 1: Circuito Door_opener

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.

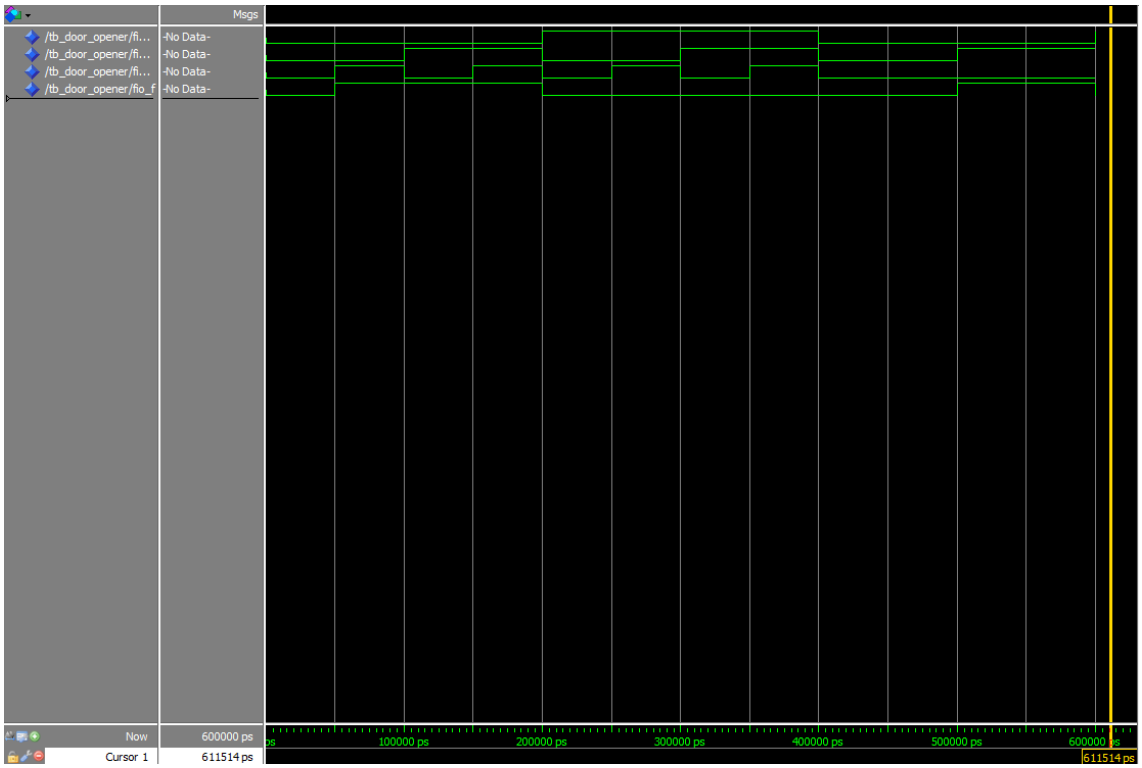

Figura 2: Simulação Door_opener no Multisim

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

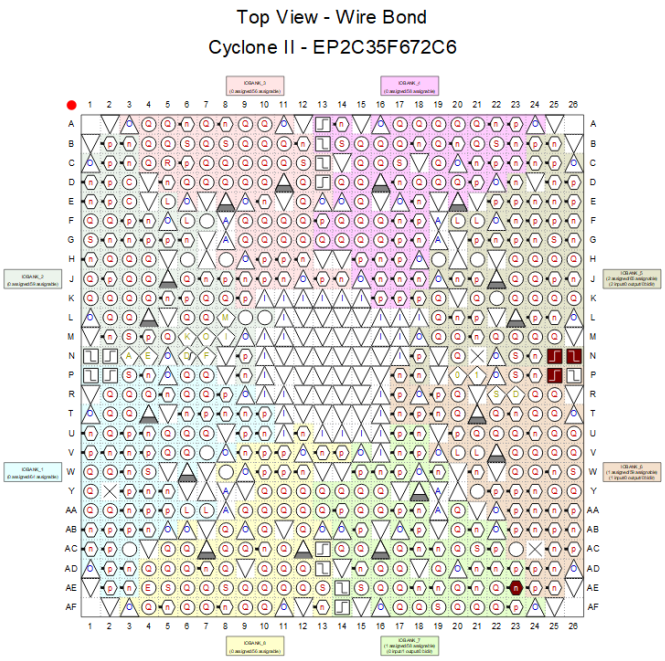

Figura 3: Pin Planner

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

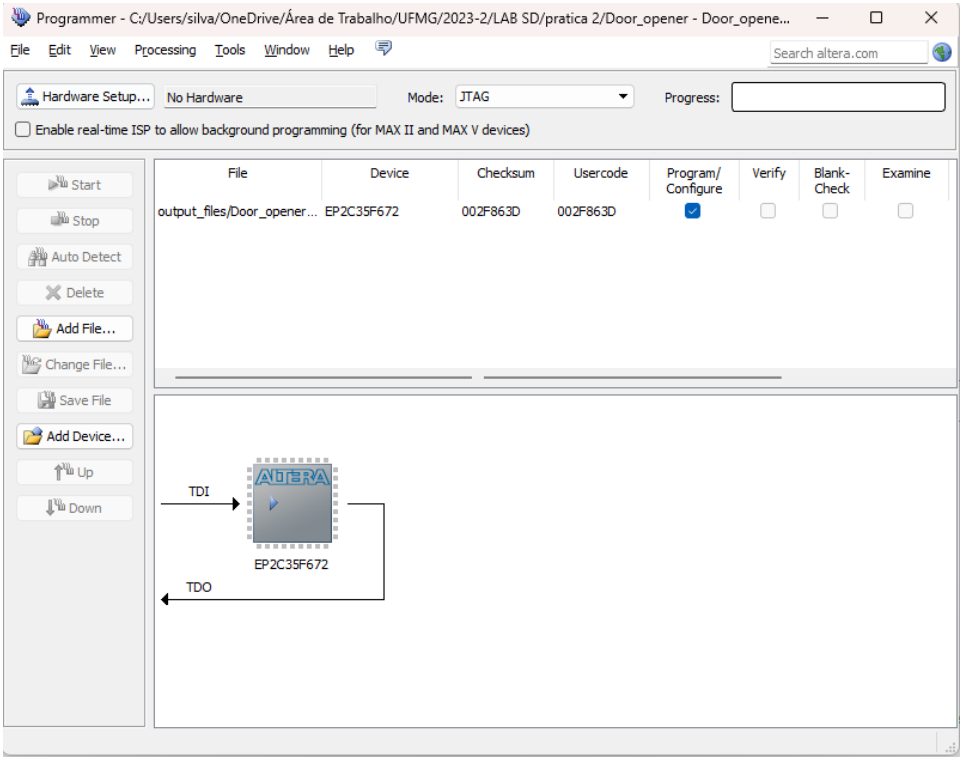

Figura 4: Programmer