Importar as configurações dos pinos salvas em um arquivo `.csv`

- O resultado do procedimento de importação apresentado aqui está no arquivo `Output_Pin_Planner.pdf`.

1 - Arquivo com as configurações dos pinos do kit FPGA:

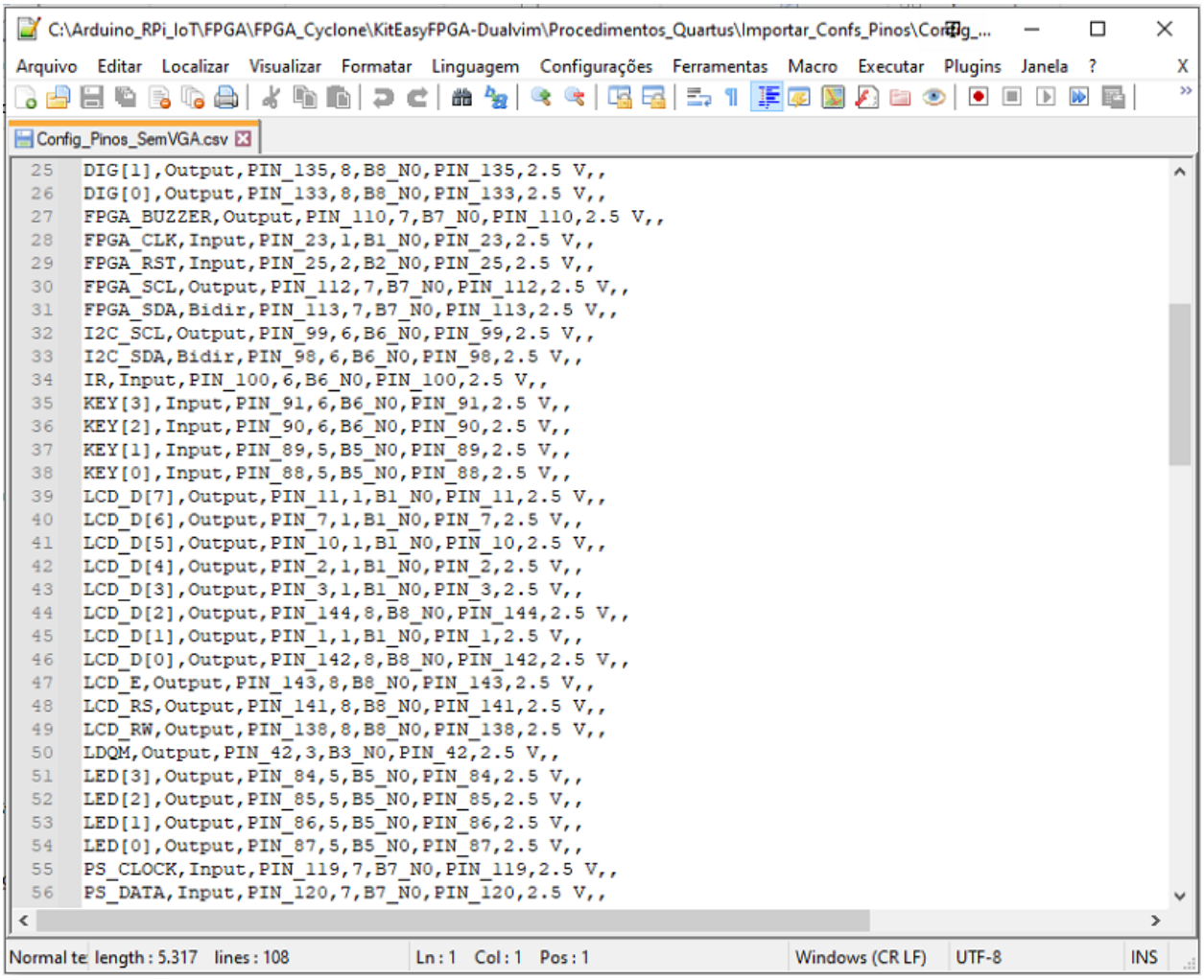


Figure 1: Arquivo com as configurações dos pinos

- O arquivo `Config_Pinos_SemVGA.csv` contém as configurações dos pinos do kit RZ-EasyFPGA A2.2.
- Esse arquivo será importado ao projeto do Quartus.

2 - Importar as configurações do arquivo `.csv` no projeto

3 - Visualizando o resultado da importação no arquivo `Teste.qsf`

4 - Programa de teste:

```
module Teste(
```

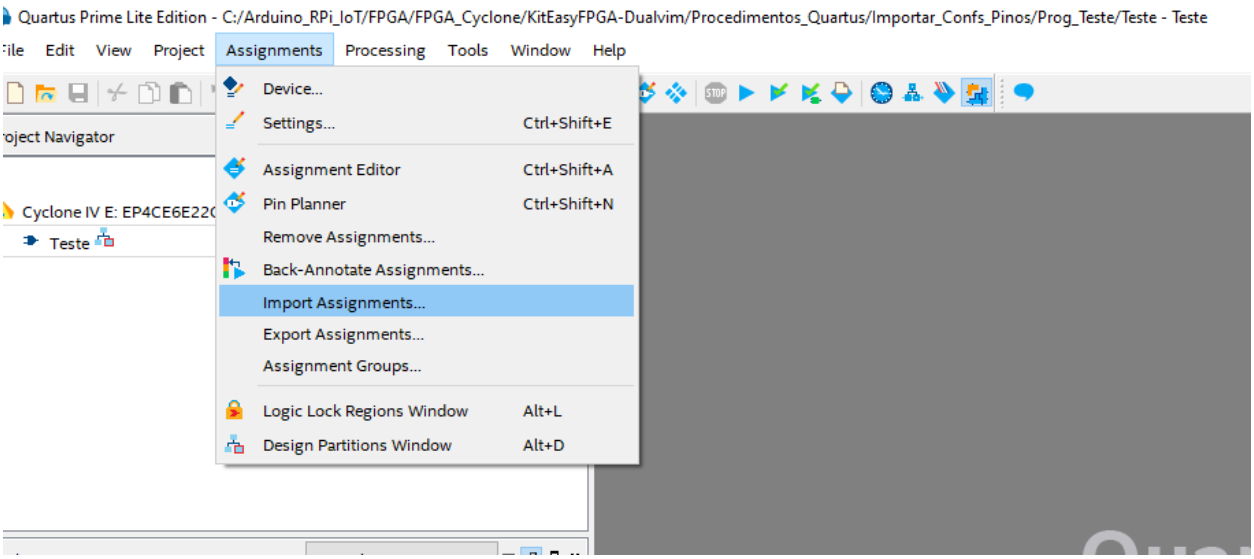


Figure 2: `Assignments/Import Assignments...`

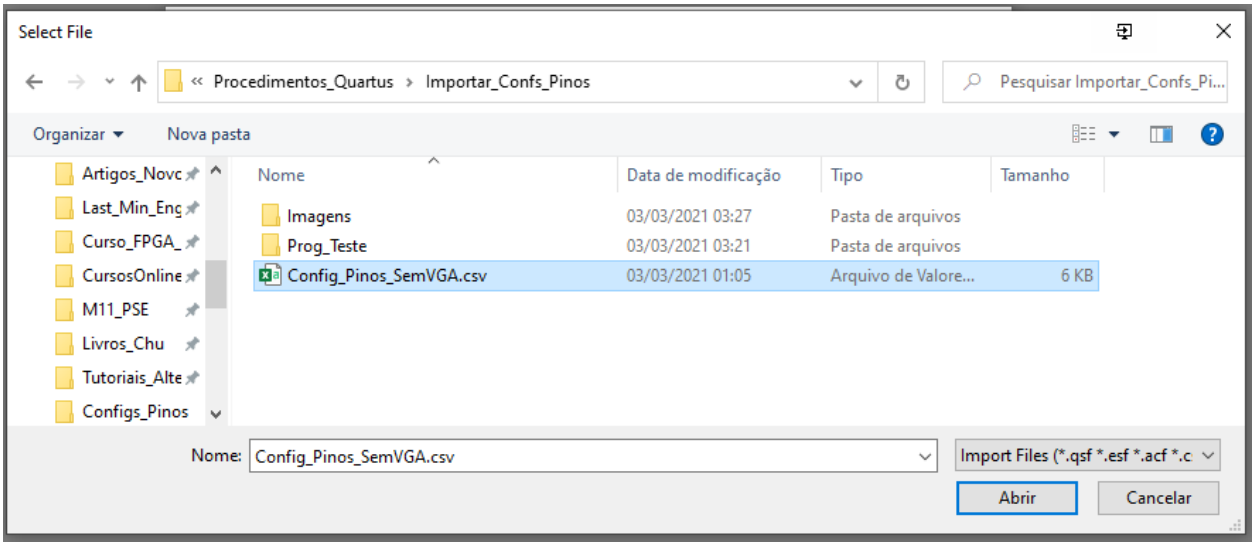


Figure 3: Selecionando o arquivo com as configurações

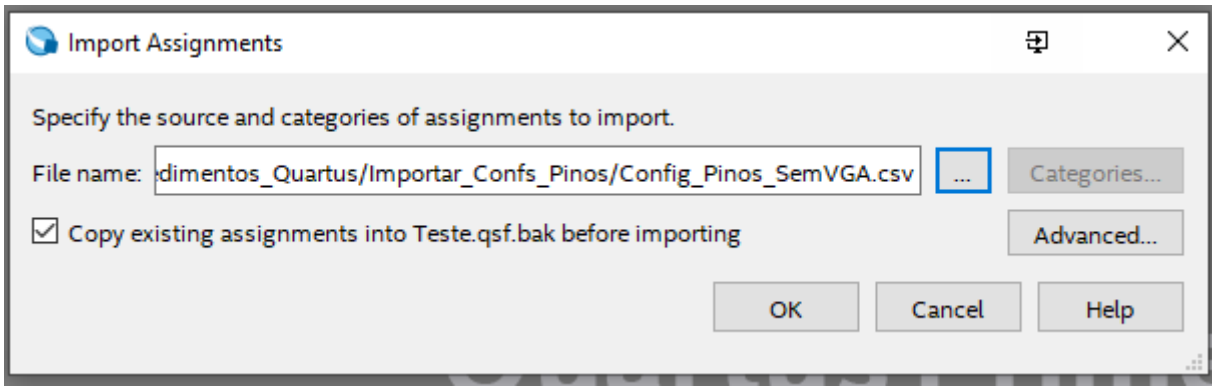


Figure 4: Confirmar a importação

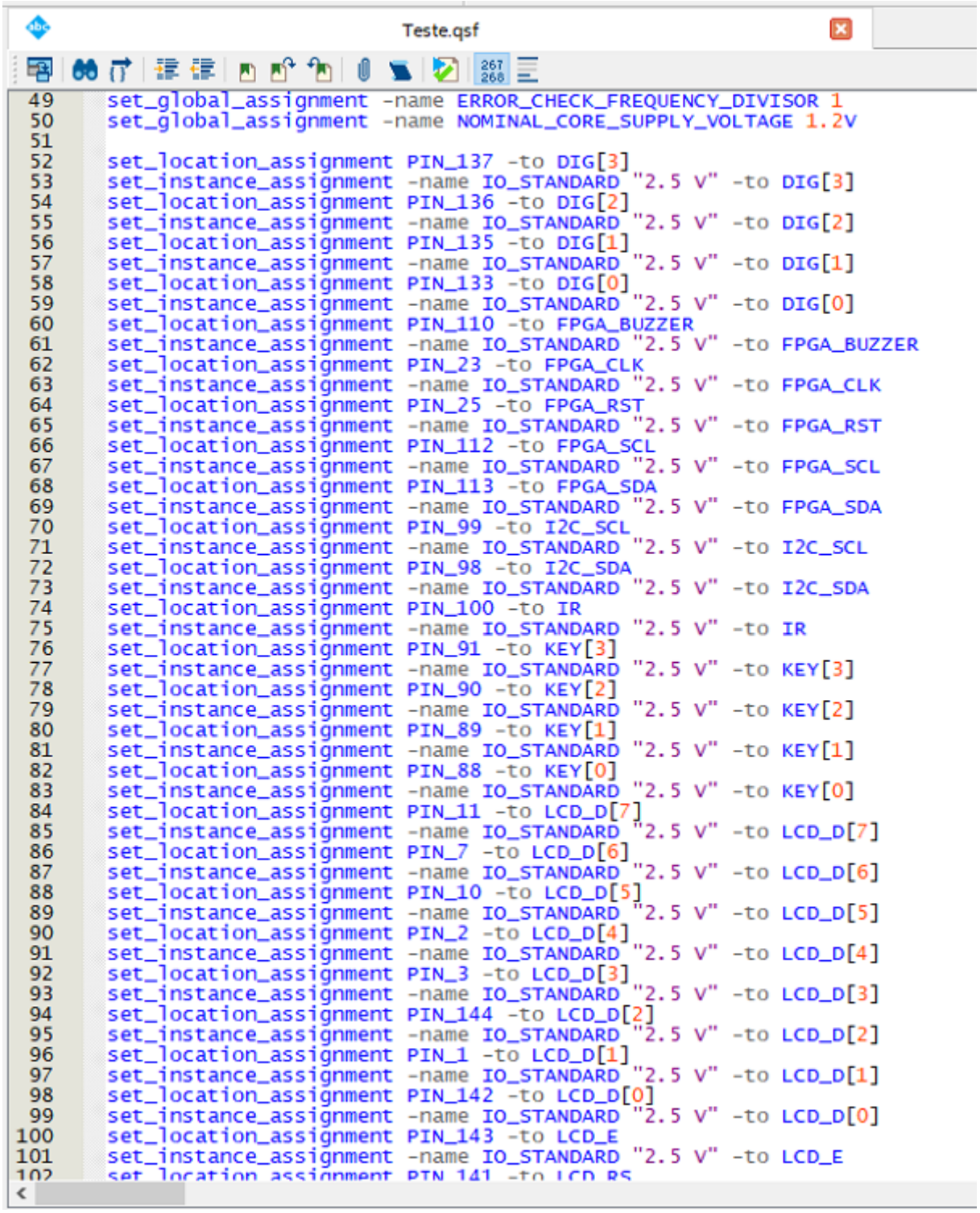


Figure 5: Resultado da importação no arquivo `Teste.qsf`

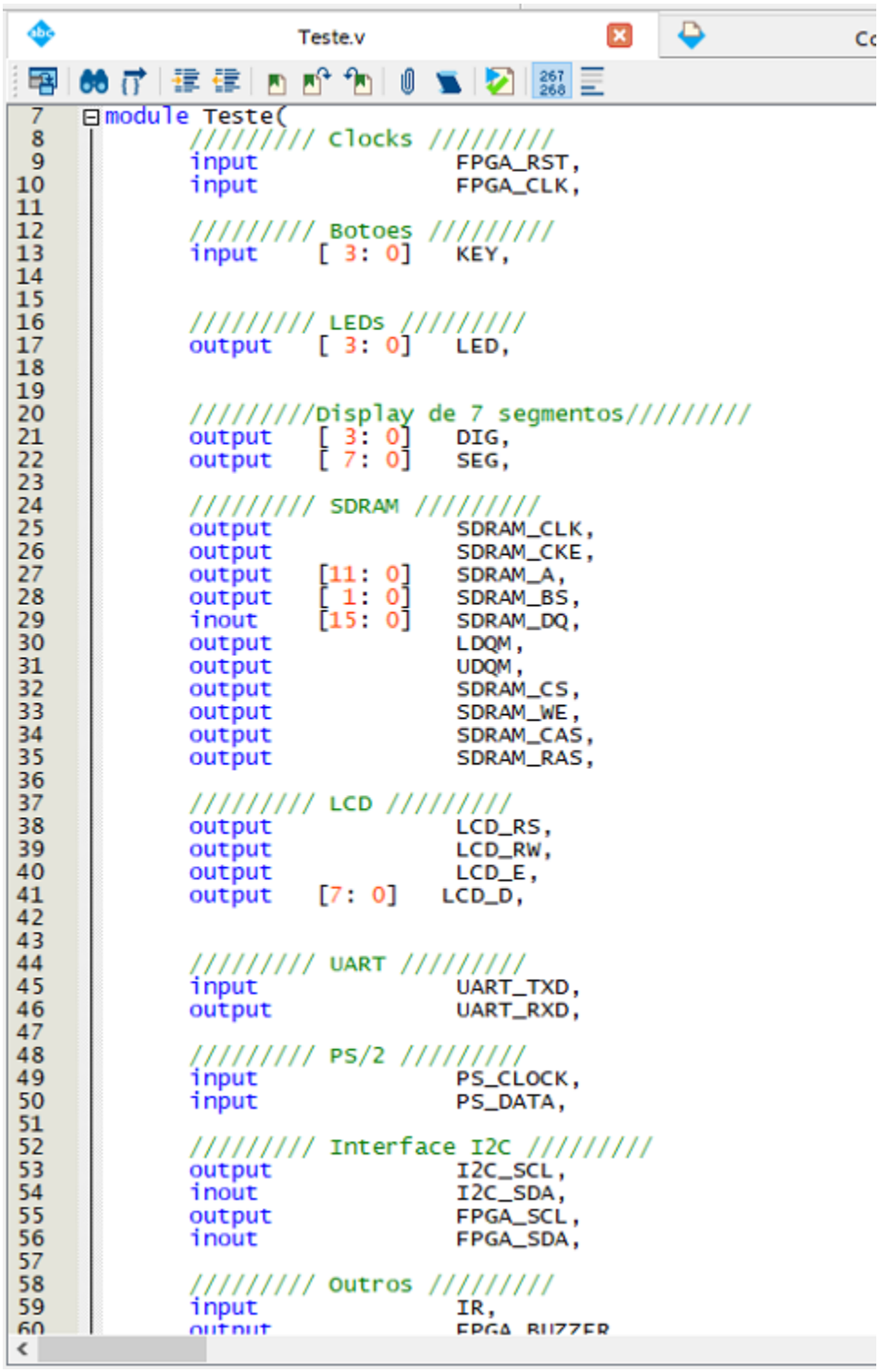


Figure 6: Script em Verilog `Teste.v`

```
    ///////// Clocks /////////
      input              FPGA_RST,
      input              FPGA_CLK,
      ///////// Botoes /////////
      input    [3: 0]   KEY,
      ///////// LEDs /////////
      output   [3: 0]   LED,
      /////////Display de 7 segmentos/////////
      output   [3: 0]   DIG,
      output   [7: 0]   SEG,
      ///////// SDRAM /////////
      output             SDRAM_CLK,
      output             SDRAM_CKE,
      output   [11: 0]   SDRAM_A,
      output   [ 1: 0]   SDRAM_BS,
      inout    [15: 0]   SDRAM_DQ,
      output             LDQM,
      output             UDQM,
      output             SDRAM_CS,
      output             SDRAM_WE,
      output             SDRAM_CAS,
      output             SDRAM_RAS,
    ///////// LCD /////////
    output             LCD_RS,
    output             LCD_RW,
    output             LCD_E,
    output   [7: 0]   LCD_D,
    ///////// UART /////////
    input              UART_TXD,
    output             UART_RXD,
    ///////// PS/2 /////////
    input              PS_CLOCK,
    input              PS_DATA,
    ///////// Interface I2C /////////
      output             I2C_SCL,
      inout              I2C_SDA,
    output             FPGA_SCL,
      inout              FPGA_SDA,
    ///////// Outros /////////
    input              IR,
    output             FPGA_BUZZER

);

    //Acender os LEDs de acordo com o botao pressionado
    assign LED[0] = ~KEY[0];
    assign LED[1] = ~KEY[1];
    assign LED[2] = ~KEY[2];
    assign LED[3] = ~KEY[3];
endmodule
```

configurações importadas no *Pin Planner*

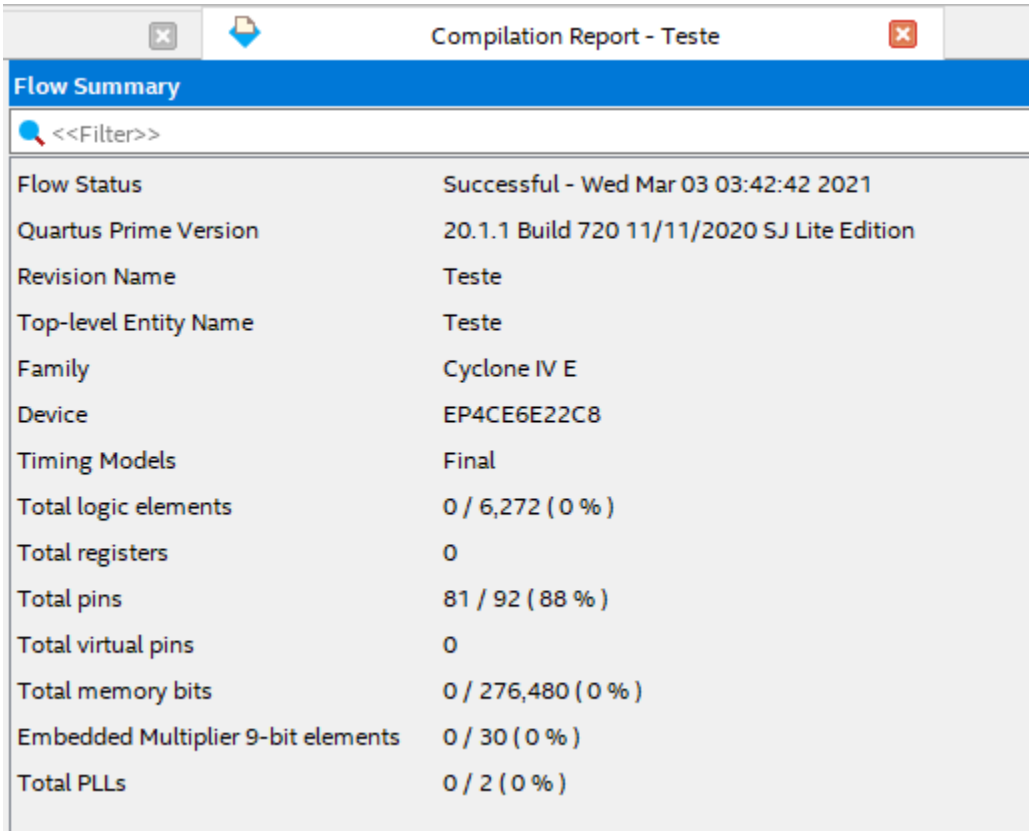


Figure 7: *Compilation Rport*

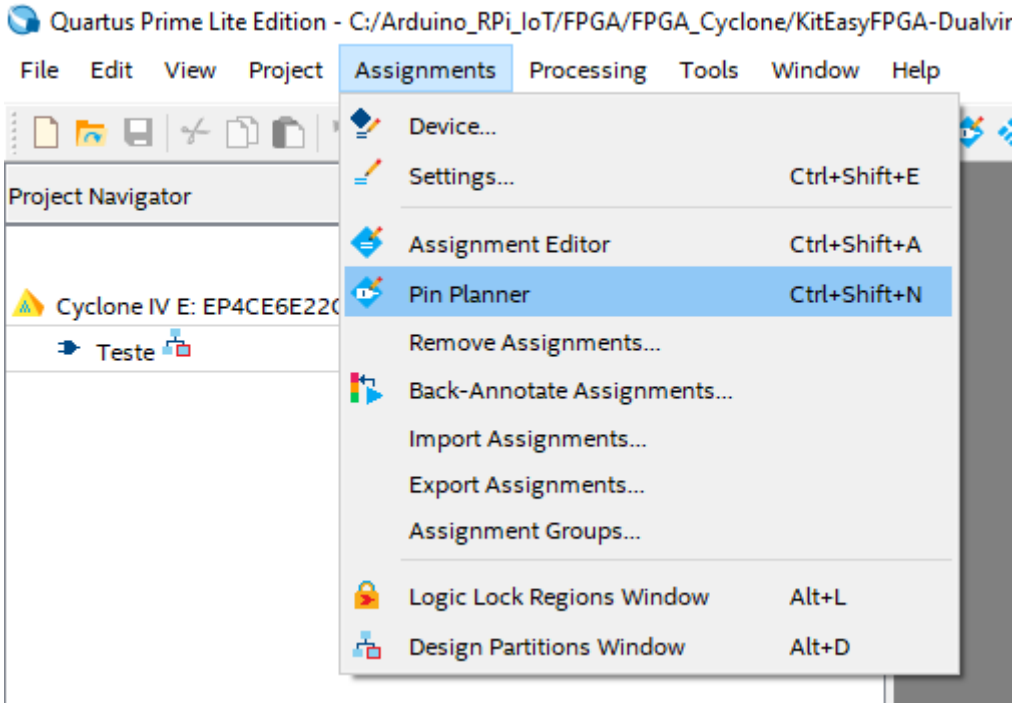
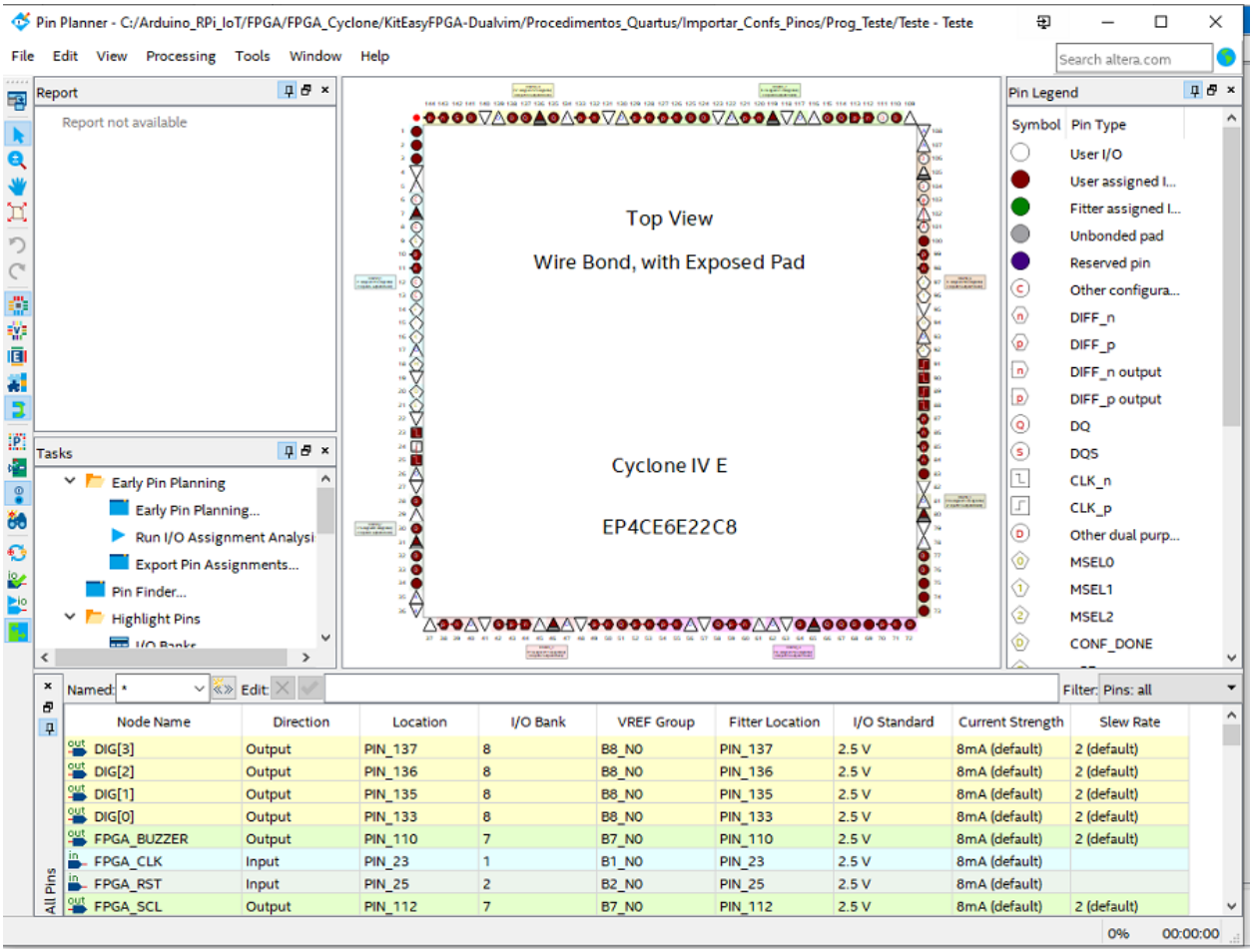


Figure 8: `Assignments/Pin Planner`



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Current Strength	Slew Rate
DIG[3]	Output	PIN_137	8	B8_N0	PIN_137	2.5 V	8mA (default)	2 (default)
DIG[2]	Output	PIN_136	8	B8_N0	PIN_136	2.5 V	8mA (default)	2 (default)
DIG[1]	Output	PIN_135	8	B8_N0	PIN_135	2.5 V	8mA (default)	2 (default)
DIG[0]	Output	PIN_133	8	B8_N0	PIN_133	2.5 V	8mA (default)	2 (default)
FPGA_BUZZER	Output	PIN_110	7	B7_N0	PIN_110	2.5 V	8mA (default)	2 (default)
FPGA_CLK	Input	PIN_23	1	B1_N0	PIN_23	2.5 V	8mA (default)	
FPGA_RST	Input	PIN_25	2	B2_N0	PIN_25	2.5 V	8mA (default)	
FPGA_SCL	Output	PIN_112	7	B7_N0	PIN_112	2.5 V	8mA (default)	2 (default)

Figure 9: Tela do *Pin Planner*