SUPER ROTEIROS

FPGA DE10-LITE

Universidade Federal do Amazonas
Faculdade de Tecnologia
Departamento de Eletrônica e Computação
Coordenação Geral de Laboratórios de Eletrônica

FPGA DE10-LITE

Sumário

Tutorial de uso do FPGA DE10-LITE

Histórico

Versão	Data	Descrição	Responsável
1.0	01/09/2022	Criação do tutorial de uso	Hitalo Perseu
2.0	15/09/2023	Mudança de Template	Hitalo Perseu
2.1	12/06/2024	Atualização do Conteúdo	Luiz Neto

Aprovação

Autor	Revisão
Técnico	Coordenador

Tutorial de uso do FPGA DE10-LITE

Introdução

Uma plataforma FPGA (Field-Programmable Gate Array) é um dispositivo de hardware altamente configurável que permite a criação de circuitos digitais personalizados. Existem vários tipo de FPGA, como por exemplo: FPGA DE0-NANO, DE10-LITE, DE2-115 e entre outros. No caso deste roteiro, iremos abordar o modelo DE10-LITE.

DE2-115

DE10-LITE

DE0-NANO

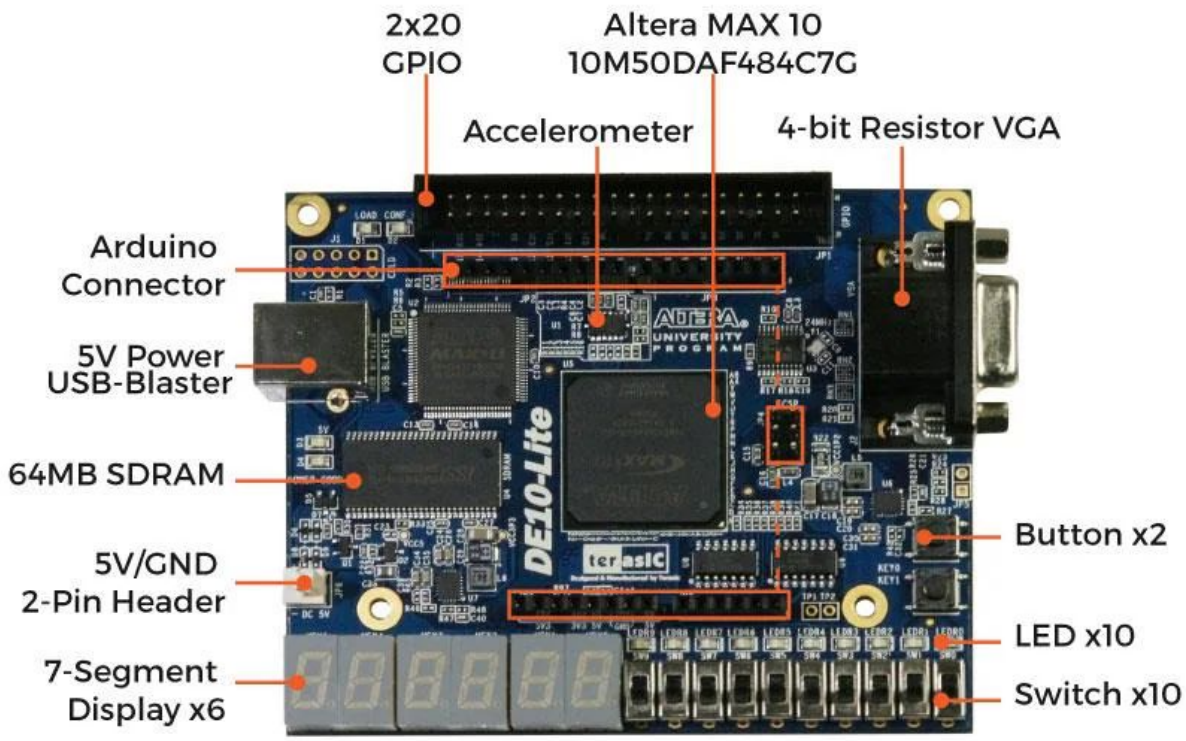
Tutorial de uso do FPGA DE10-LITE

Conhecendo o hardware

A plataforma

O modelo da plataforma FPGA que será utilizado de referência para o roteiro será o DE10-LITE. Este roteiro tem a finalidade de orientar no entendimento, instalação, configuração e testes.

Será necessário, para embarcamos os projetos, o DataSheet da plataforma FPGA DE10-LITE, para ter acesso clique aqui.



Tutorial de uso do FPGA DE10-LITE

Conhecendo o hardware

O cabo

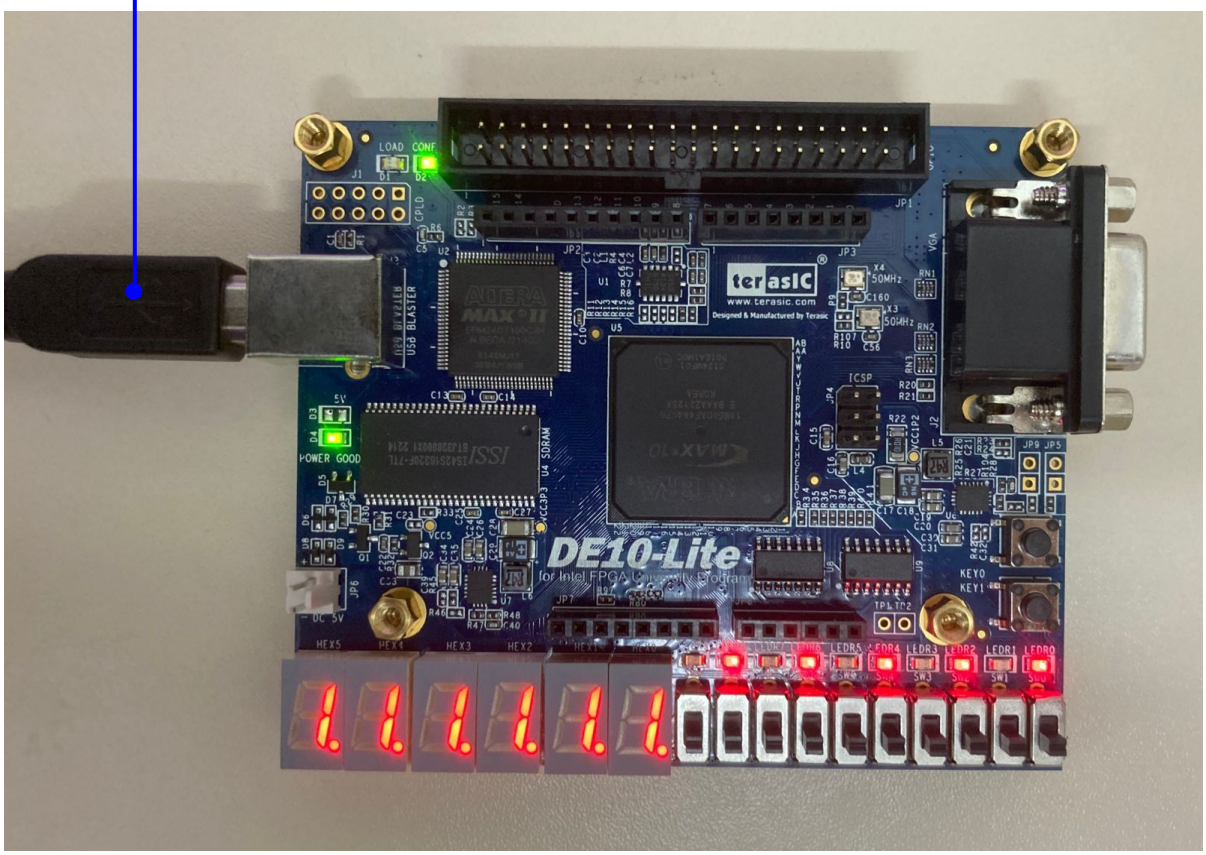
A plataforma FPGA DE10-LITE vem acompanhado de um cabo **USB-AB**, para transferir dados e alimentar a plataforma.

Tutorial de uso do FPGA DE10-LITE

Conhecendo o hardware

Conexão da plataforma

Para utilizarmos a plataforma, basta conectar a entrada USB-A no computador e a entrada **USB-B** na plataforma.

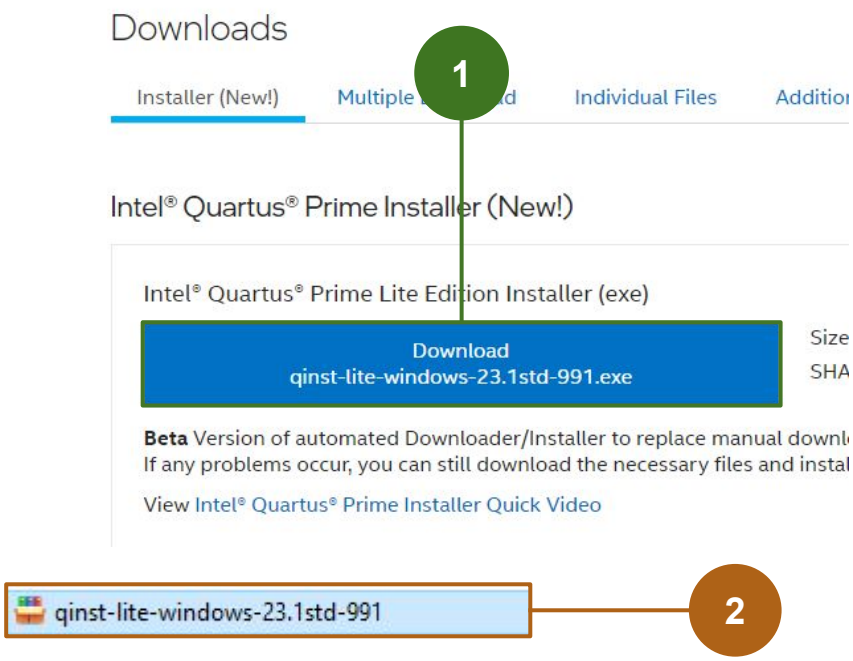


Tutorial de uso do FPGA DE2-115

Instalação do Software Quartus Prime Version 23.1std Lite Edition

Download do Software

Para utilizarmos a plataforma FPGA precisamos do Software Quartus Prime Standard Edition. Para acessar o site do fornecedor, clique aqui. Logo depois, clique no **botão "Download"**. Efetuado o download, **o arquivo baixado é do tipo <.exe>**, ou seja, executável, dê um duplo clique no arquivo.

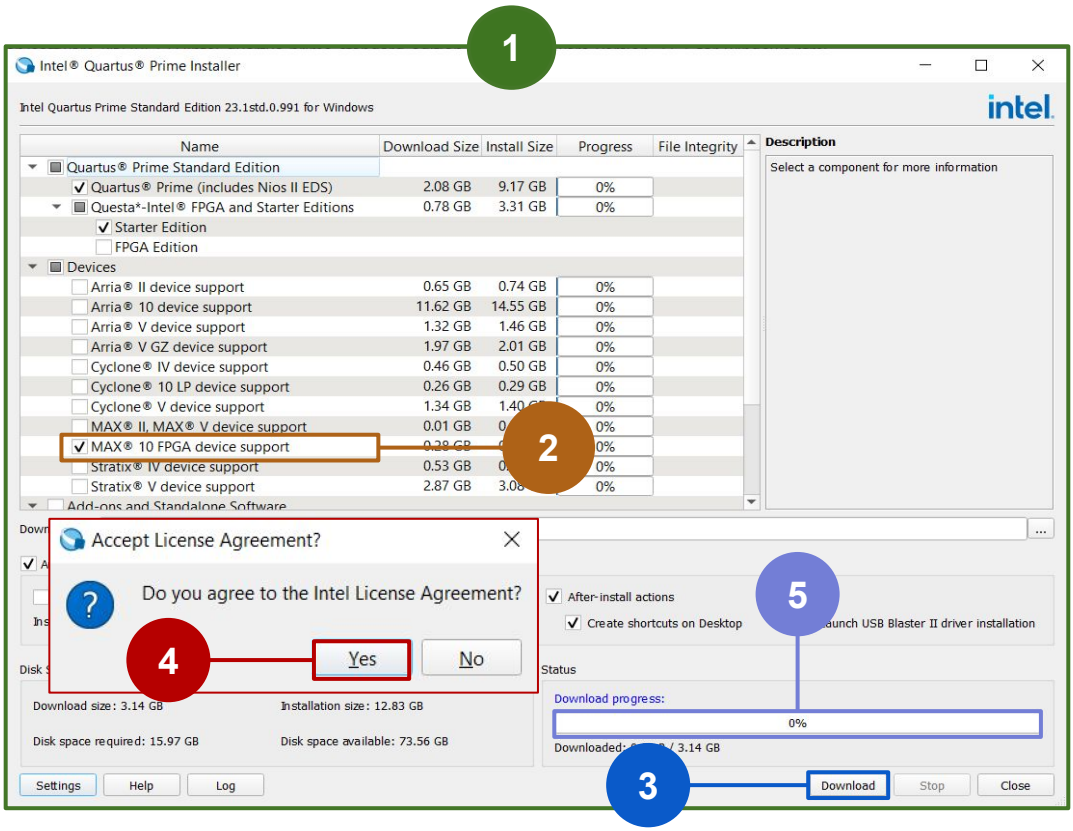


Tutorial de uso do FPGA DE2-115

Instalação do Software Quartus Prime Version 23.1std Lite Edition

Configuração do Software

Em seguida abrirá a **aba de instalação**, nesta parte selecione **"MAX 10 FPGA device Support"**, depois clique em **"download"**. Depois aceite os termos da licença clicando em **"Yes"**. E então espere a **barra de download** ser completada.

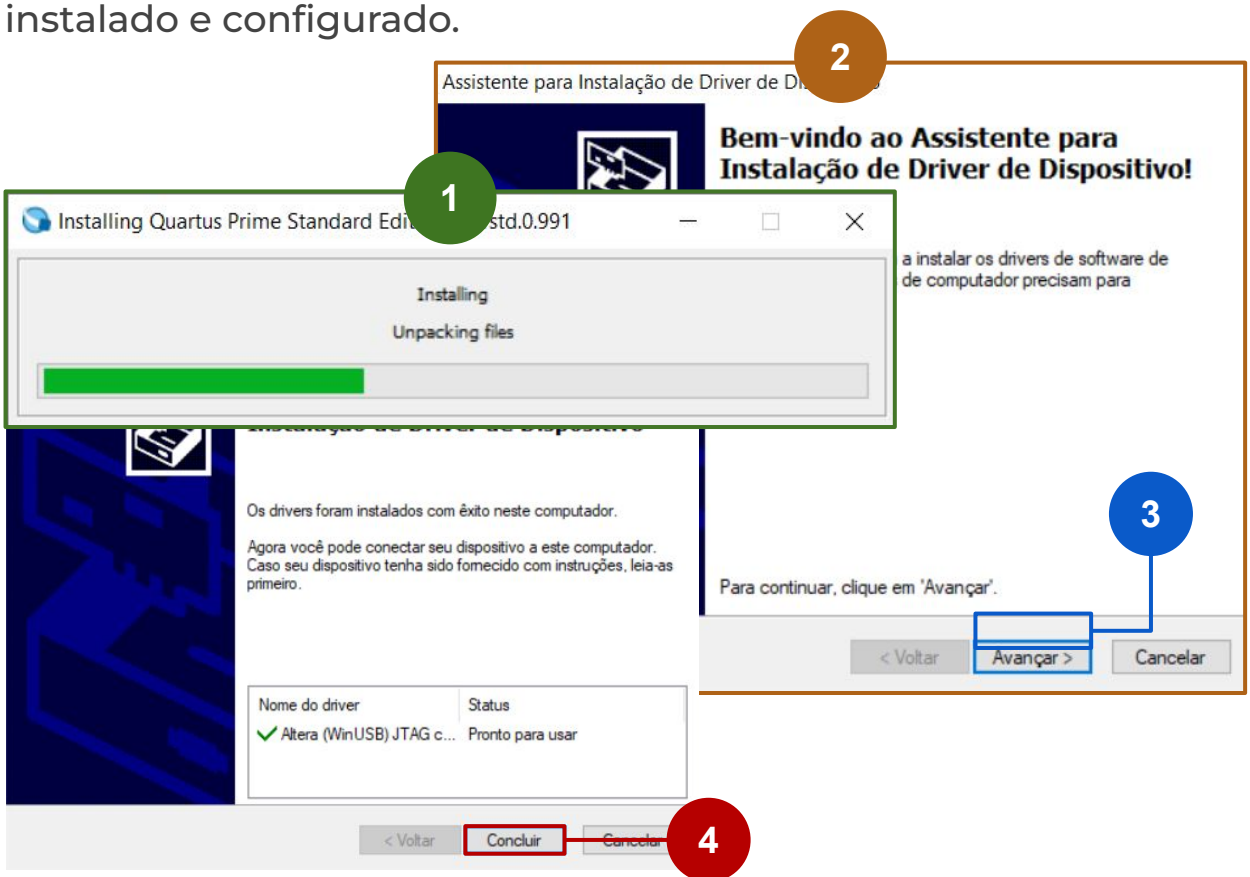


Tutorial de uso do FPGA DE2-115

Instalação do Software Quartus Prime Version 23.1std Lite Edition

Configuração do Software

Posteriormente abrirá a **janela instalação**, espere a barra de instalação ser completada. Depois de completa, abrirá a aba **"assistente de instalação de drive"**, clique em **"Avançar"** e posteriormente em **"Concluído"**. Com isso o software estará instalado e configurado.

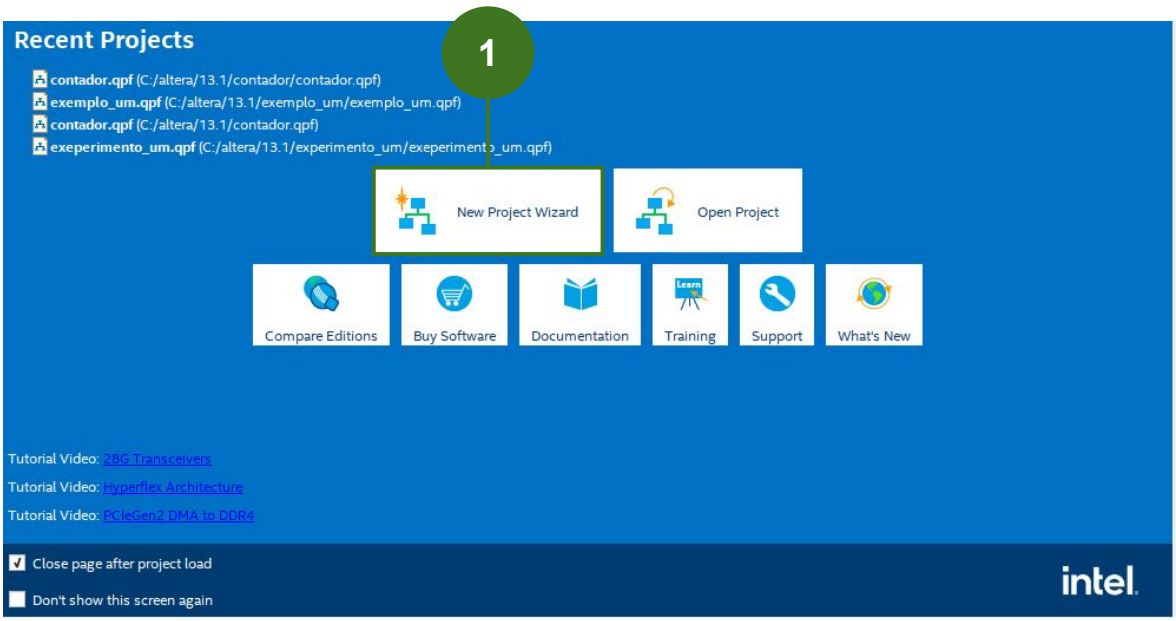


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software
Quartus Prime Version 23.1std Lite Edition

Criação de Projeto

Ao abrir o Software Quartus Prime, clique em **“Create New Project”**.

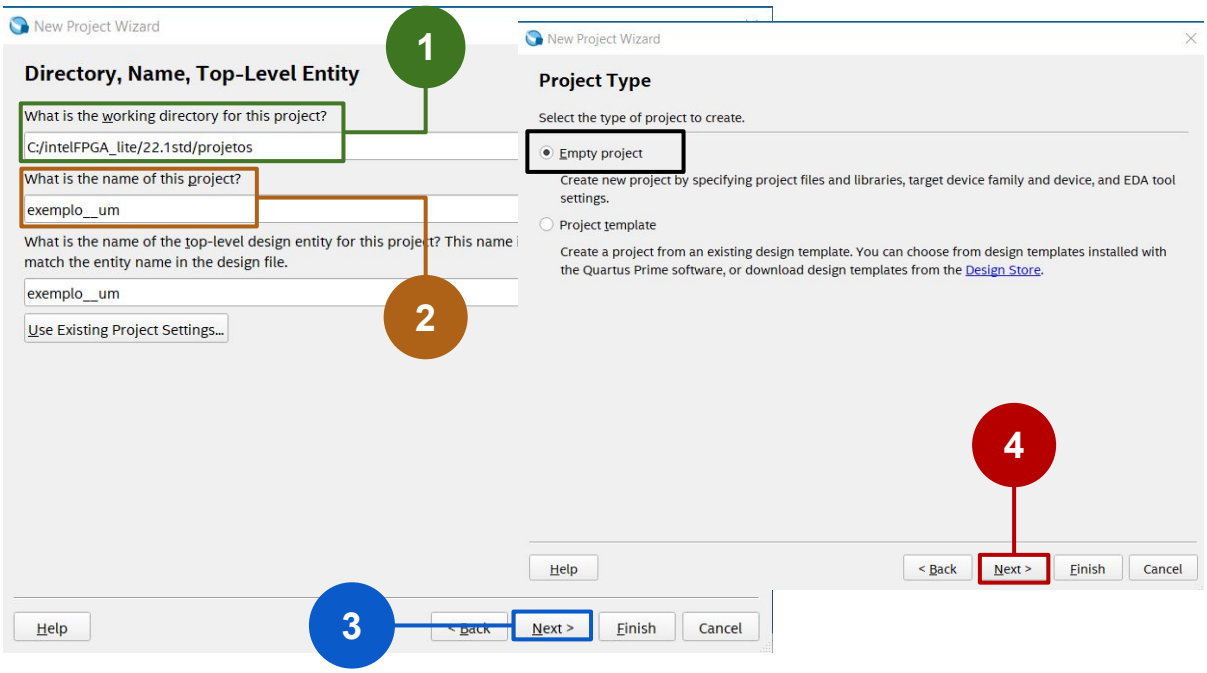


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 23.1std Lite Edition

Criação de Projeto

Na janela seguinte, defina o **diretório** e o **nome do projeto**, que será automaticamente o mesmo nome da entidade do código a ser criado posteriormente, depois clique em **"Next"**. Na próxima aba, deixe marcado a opção **"Empty Project"** e clique em **"Next"** novamente.

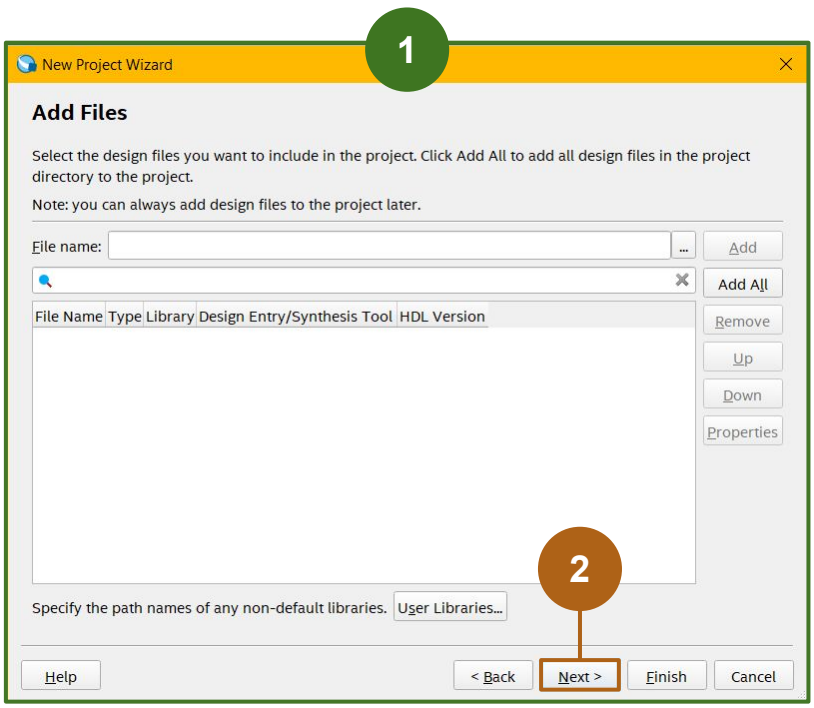


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software
Quartus Prime Version 23.1std Lite Edition

Criação de Projeto

Nessa aba de **“add filles”**, caso não tenha nenhum arquivo para ser adicionado, apenas clicar **“Next”** novamente.

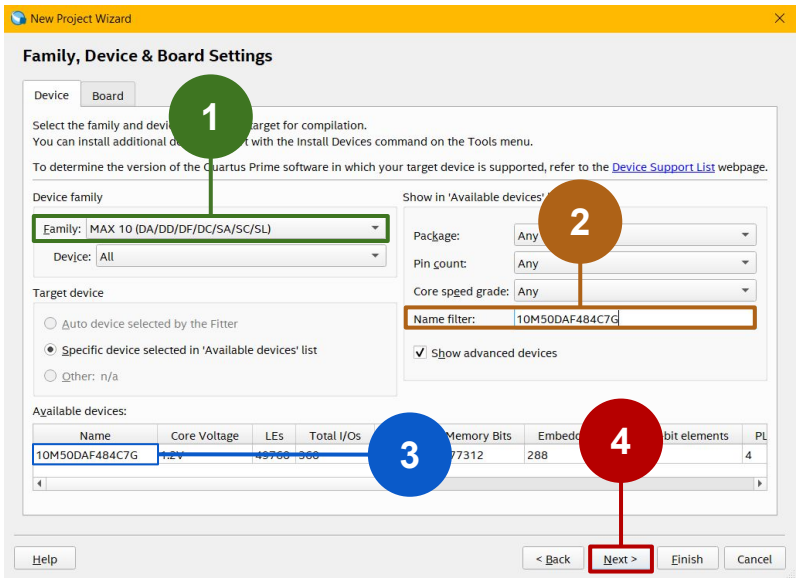


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software
Quartus Prime Version 23.1std Lite Edition

Criação de Projeto

Nesta janela, insira as configurações da plataforma FPGA, informando a família **MAX 10** e o modelo **10M50DAF484C7G**, selecione o modelo em **Available Devices**. Por fim, clique em **"Next"**.



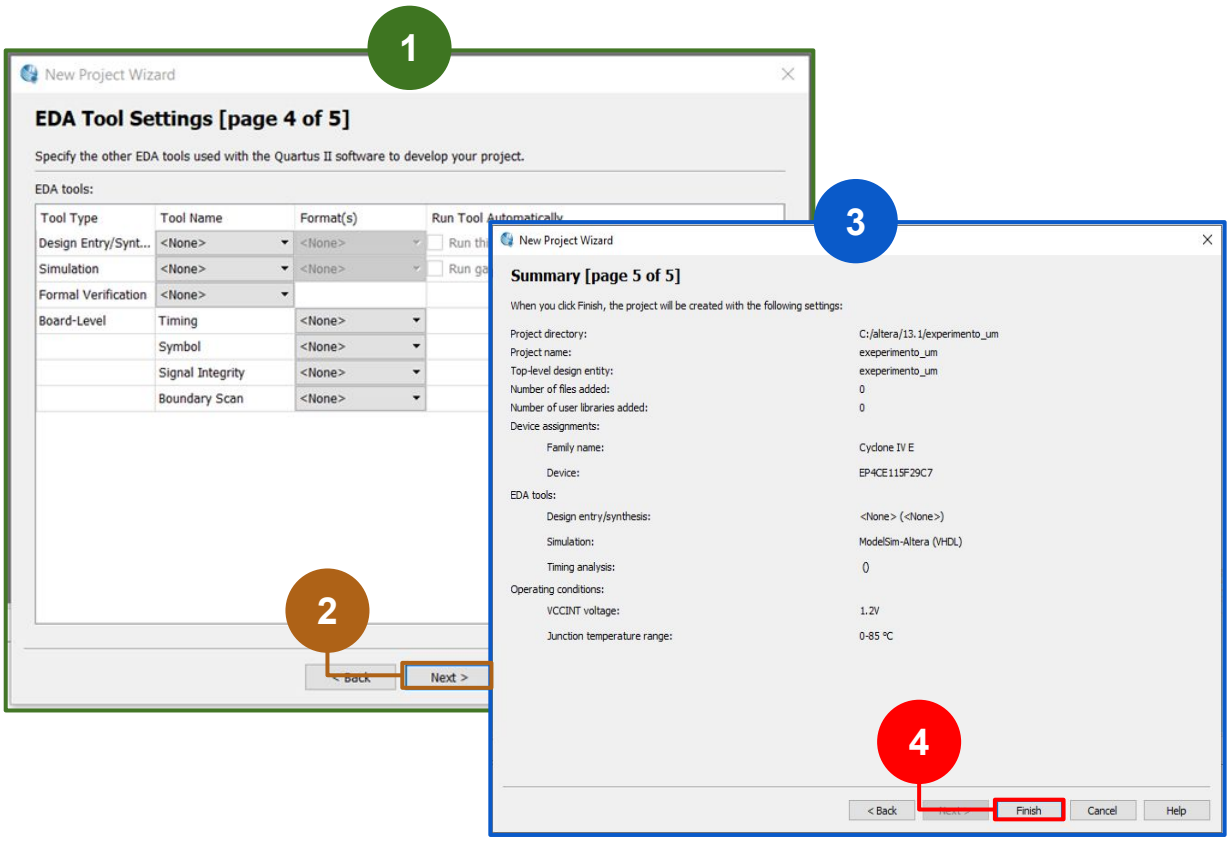
As informações de Família e o modelo da plataforma, estão disponíveis no processador da Placa FPGA.
Exemplo - Família: MAX 10 e o Modelo: 10M50DAF484C7G.

Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 23.1std Lite Edition

Criação de Projeto

Nesta aba de **inserção dos recursos do projetos**, apenas clicar **"Next"**. Na última aba, visualizaremos o **resumo das informações** incluídas para criação do projeto, após leitura, clicar em **"Finish"**. Com isso, o projeto foi criado.

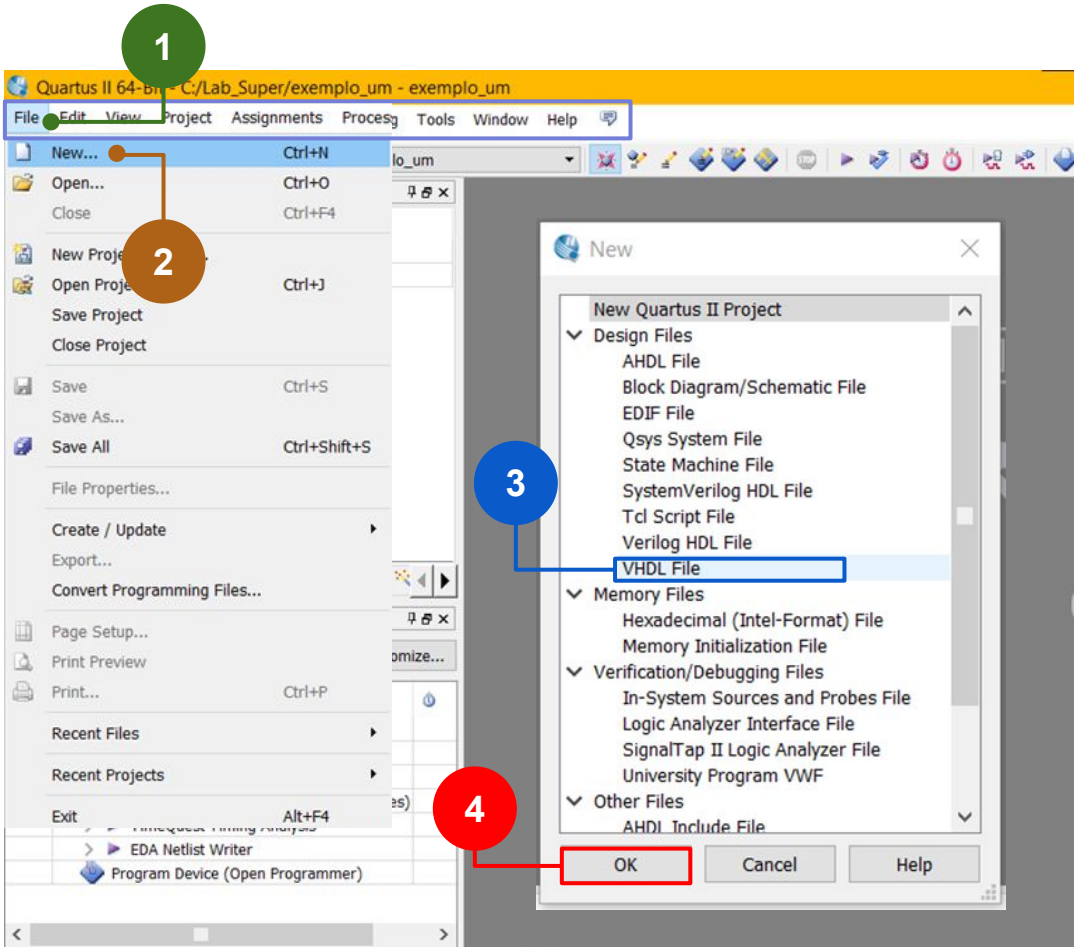


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Criação do Programa VHDL

Após a criação do projeto, iremos criar um arquivo <.vhd>. Na **barra de ferramentas**, clicar em **"File"**, posteriormente clicar em **"New"**, escolher **"VHDL File"** e clicar **"OK"**.

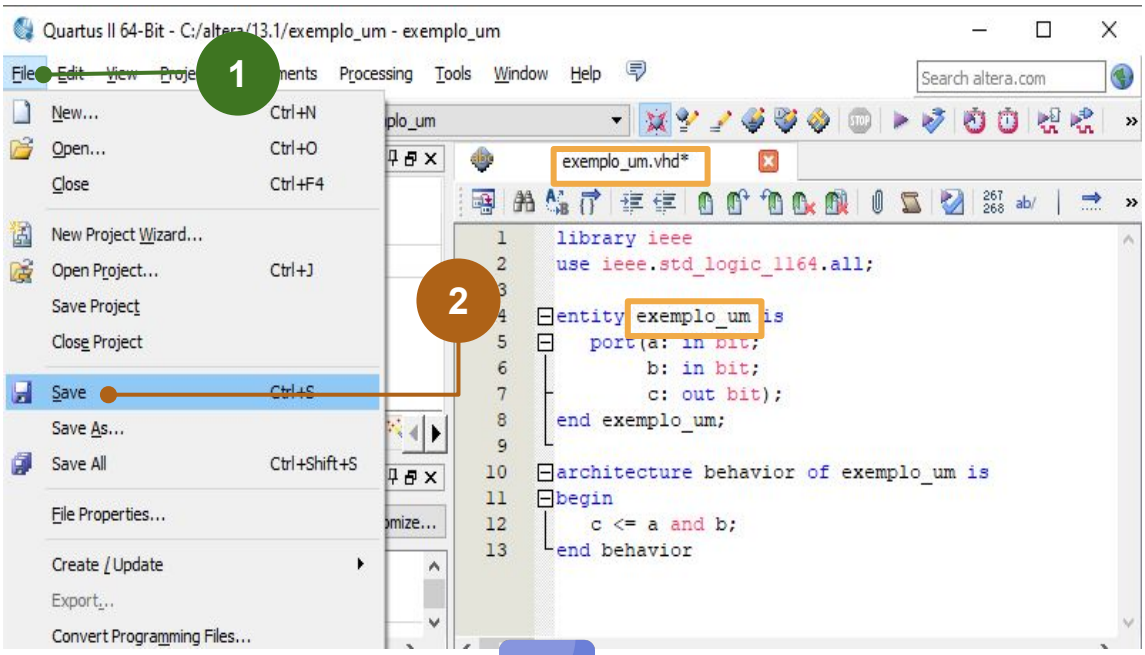


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Criação do Programa VHDL

Criado o arquivo VHDL, podemos adicionar o algoritmo em VHDL. Para salvar o algoritmo clique em **"File"**, posteriormente em **"Save"** ou podemos utilizar o atalho "Ctrl+S".



A linguagem VHDL não é case sensitive, ou seja, letras maiúsculas e minúsculas não altera a sintaxe do código.

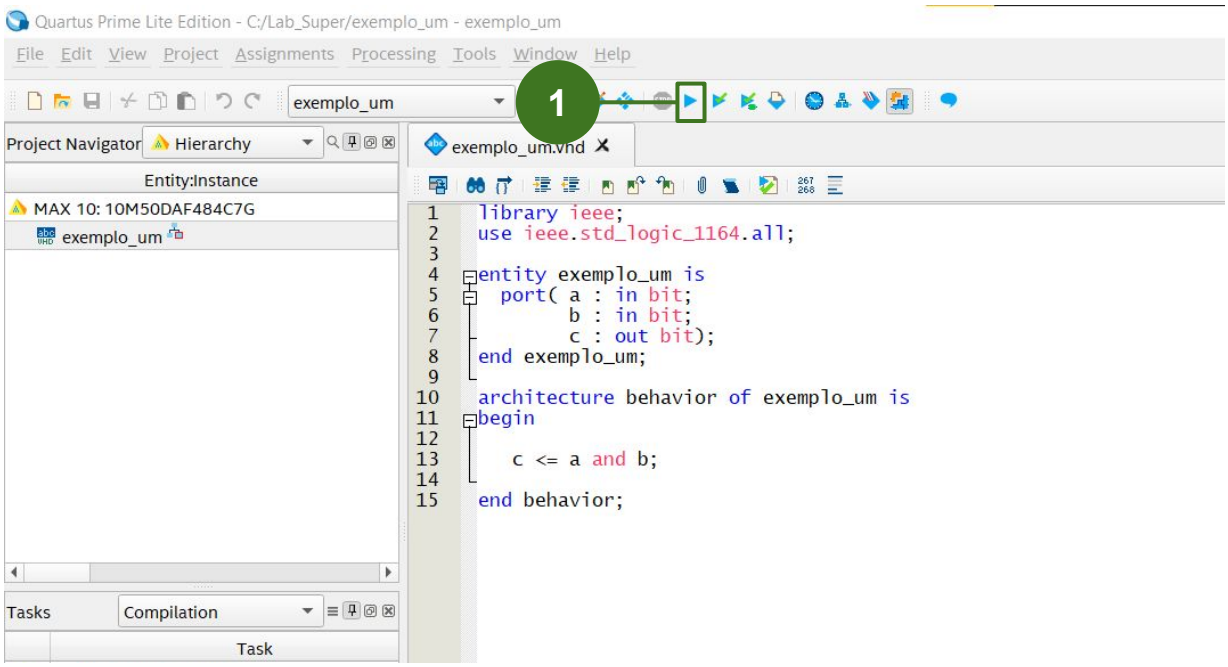
O nome do arquivo <.vhd> deverá ser o mesmo nome `entity` do código criado.

Tutorial de uso do FPGA DE10-LITE

Utilizando o Software
Quartus Prime Version 22.1std Lite Edition

Compilação do arquivo de código .vhd

Finalizado e salvo, clique em **“Start Compilation”** para iniciar a compilação do código, com intuito de verificar se está correto em relação à sintaxe da linguagem.

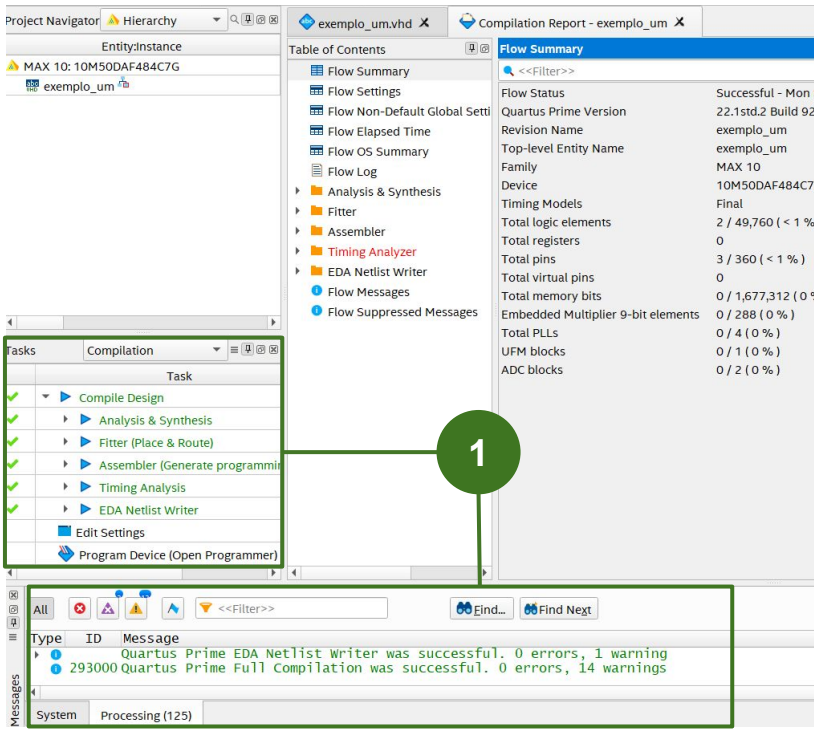


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Compilação do arquivo de código .vhd

Caso seja identificado que não contém erros, na janela **“task”** e na **janelas de avisos**, as letras ficarão verdes.

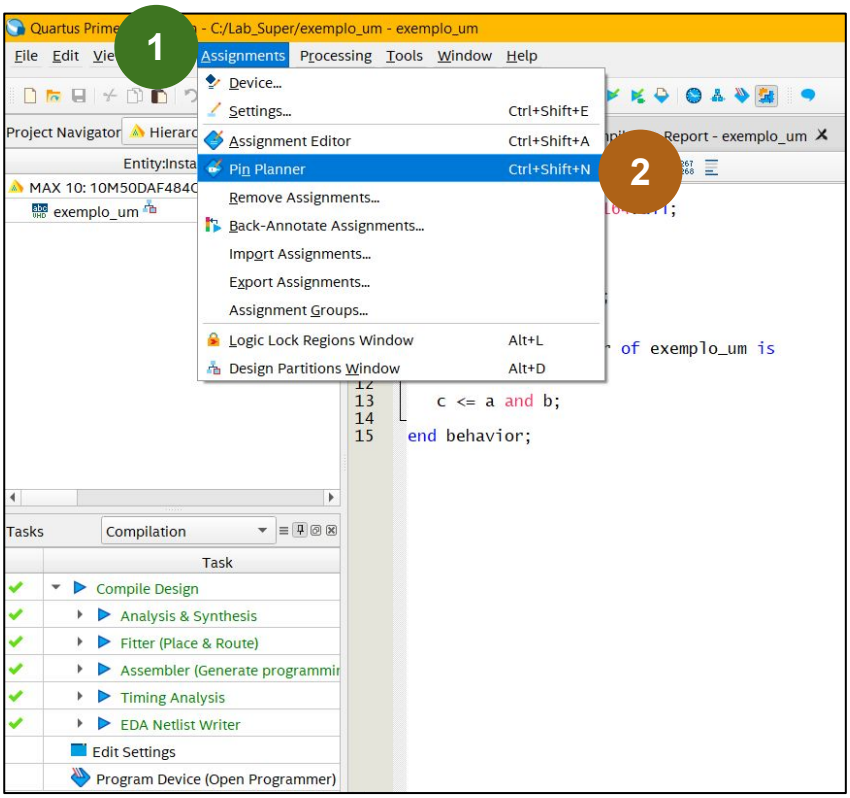


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software
Quartus Prime Version 22.1std Lite Edition

Mapeamento dos Pinos da FPGA

Nesta parte iremos mapear os pinos que foram estabelecidos no código. Na barra de ferramentas clique em **“Assignments”** e depois em **“Pin Planner”**.

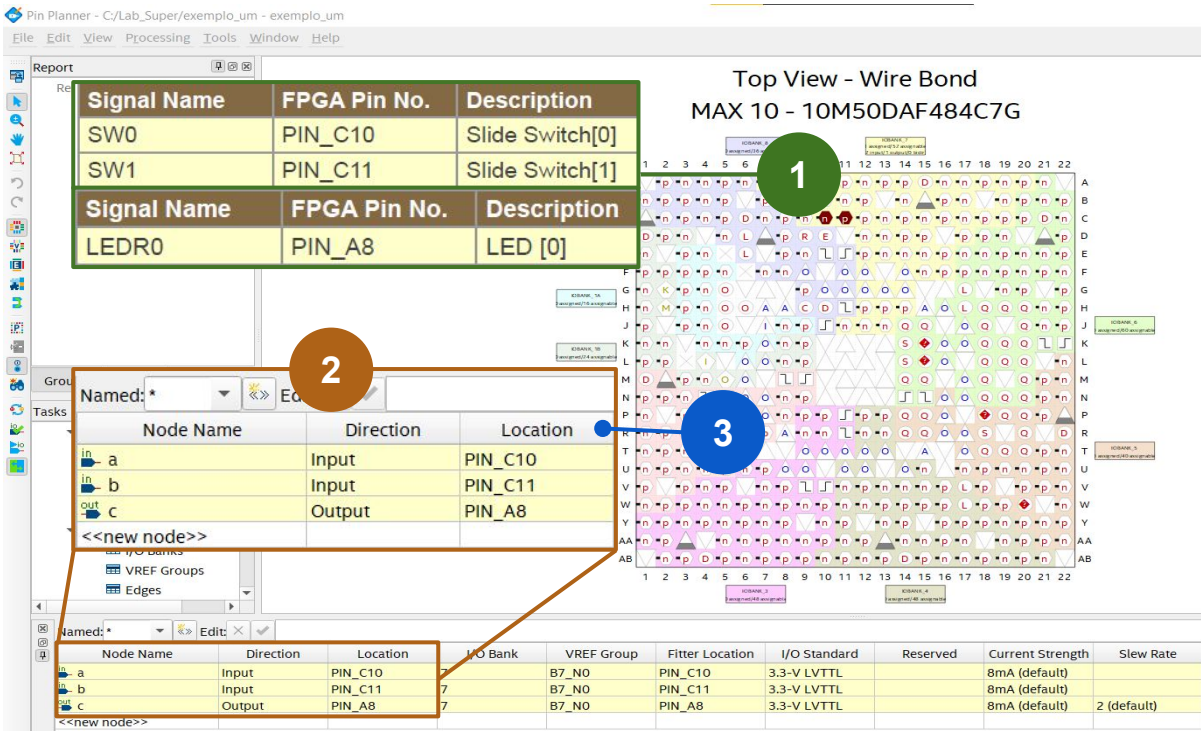


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Mapeamento dos Pinos da FPGA

Para mapear os pinos iremos precisar do DataSheet da FPGA, procure e selecione os pinos específicos para seu projeto nas **tabelas de pinagem** do manual. Depois, na parte inferior da aba "Pin Planner" têm as **entradas e saídas do seu projeto**, insira os pinos, escolhidos na tabela, na coluna **"Location"** e depois feche a aba.

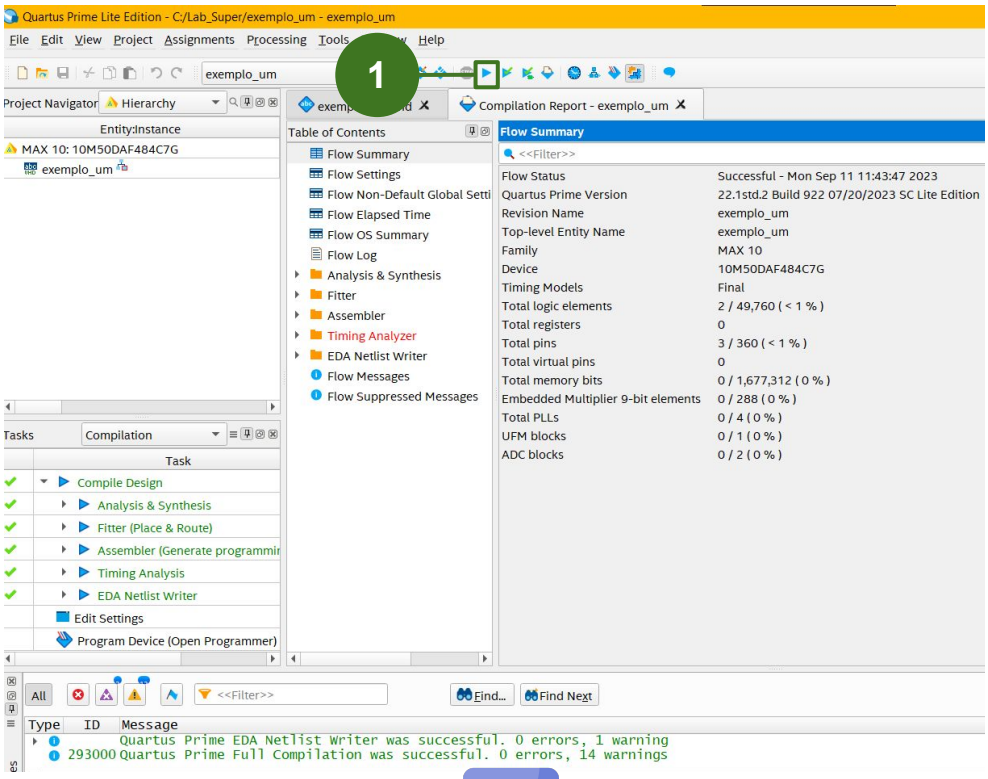


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Embarque do arquivo de código .vhd

Com o Pin Planner devidamente preenchido, deverá śer feito novamente a compilação do código, clicar em **“Start Compilation”**.



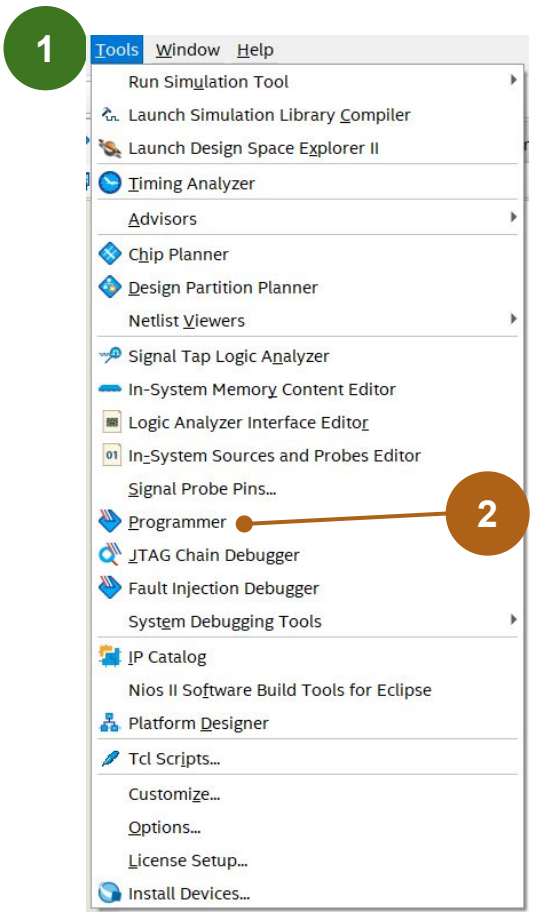
Qualquer alteração a ser realizado na escrita do código ou na configuração do Pin Planner, deverá ser feito a compilação.

Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Embarque do arquivo de código .vhd

Na barra de ferramentas, clicar em **“Tools”** e posteriormente clicar em **“Programmer”**.

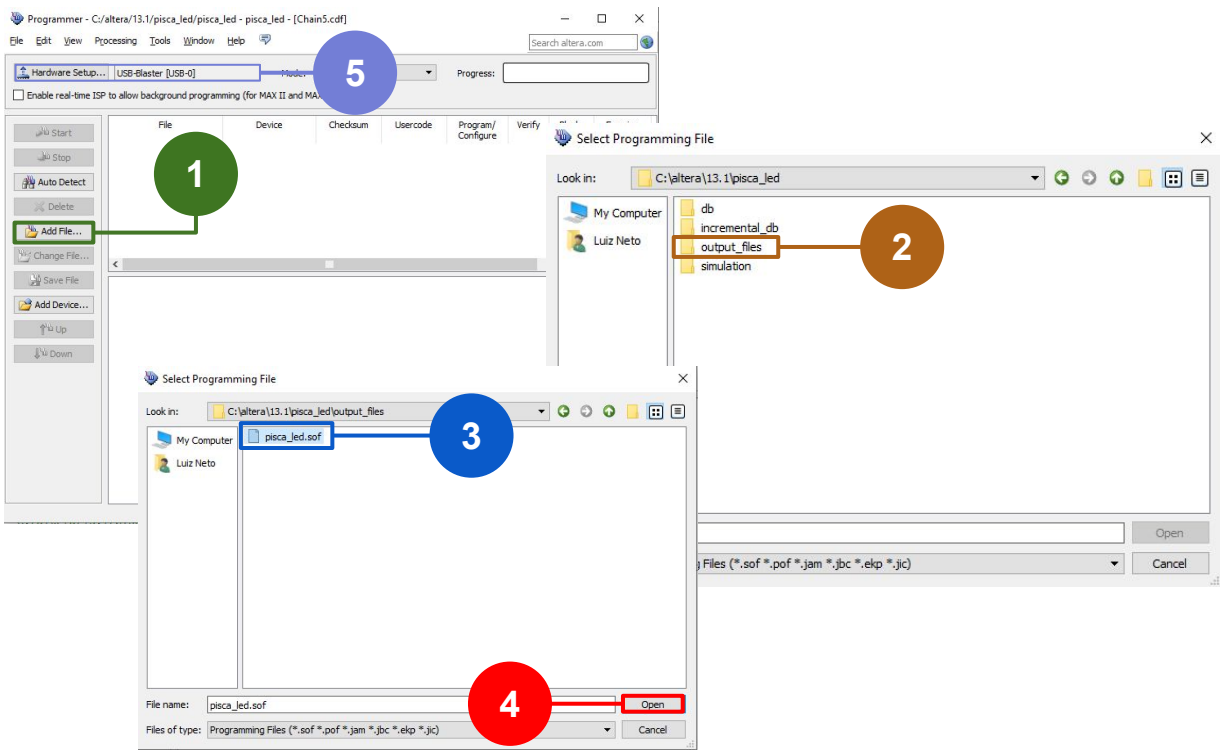


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Embarque do arquivo de código .vhd

Com a janela(Programmer) aberta, clique em **“Add File”**. Selecione a pasta **“output_files”**, posteriormente **selecione o arquivo .sof** e clique em **“open”**. Além disso, certifique-se que o software está reconhecendo o **Hardware Setup USB-Blaster**.

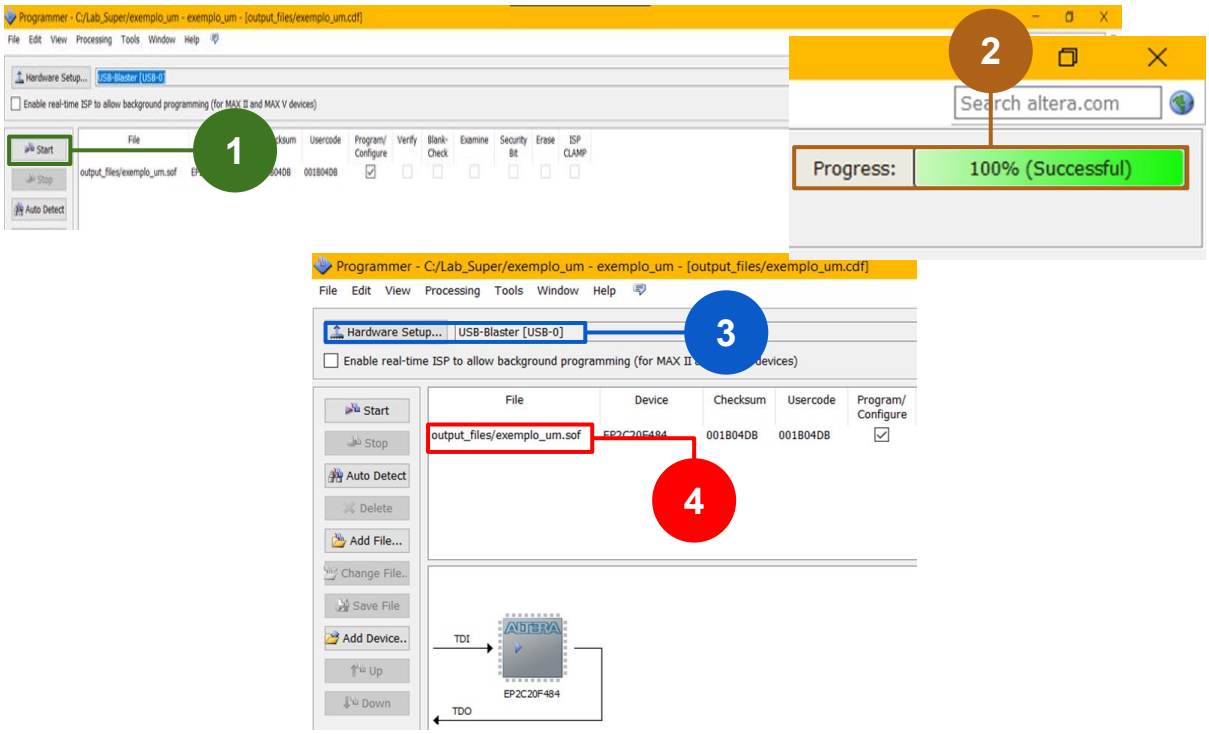


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Embarque do arquivo de código .vhd

Então clique em **"start"** e espere a **mensagem "100%(Successful) na barra "Progress"**. Em alguns casos o Software reconhece **Hardware Setup** como USB-Blaster e adiciona o **arquivo .sof** automaticamente. Carregado a barra de progresso o seu código estará embarcado.



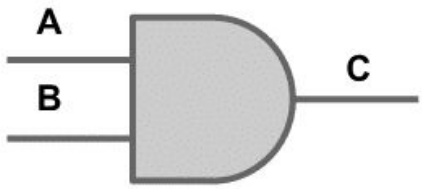
Tutorial de uso do FPGA DE10-LITE

Utilizando o Software
Quartus Prime Version 22.1std Lite Edition

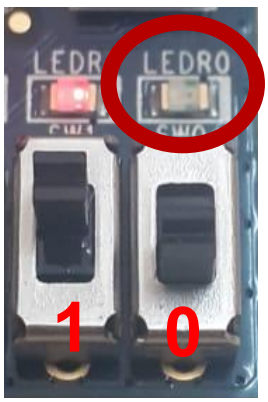
Embarque do arquivo de código .vhd

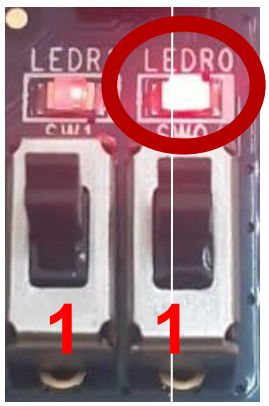
O código embarcado demonstra o funcionamento da porta and. **As chaves SW0 e SW1(PIN_AB28 e PIN_AC28)** irão representar os bits de entrada(A e B) e **o Led LEDR0(PIN_G19)** irá representar a saída C. O código deve seguir a seguinte tabela verdade:

PORTA E (AND) C=A•B



A	B	C
0	0	0
0	1	0
1	0	0
1	1	1



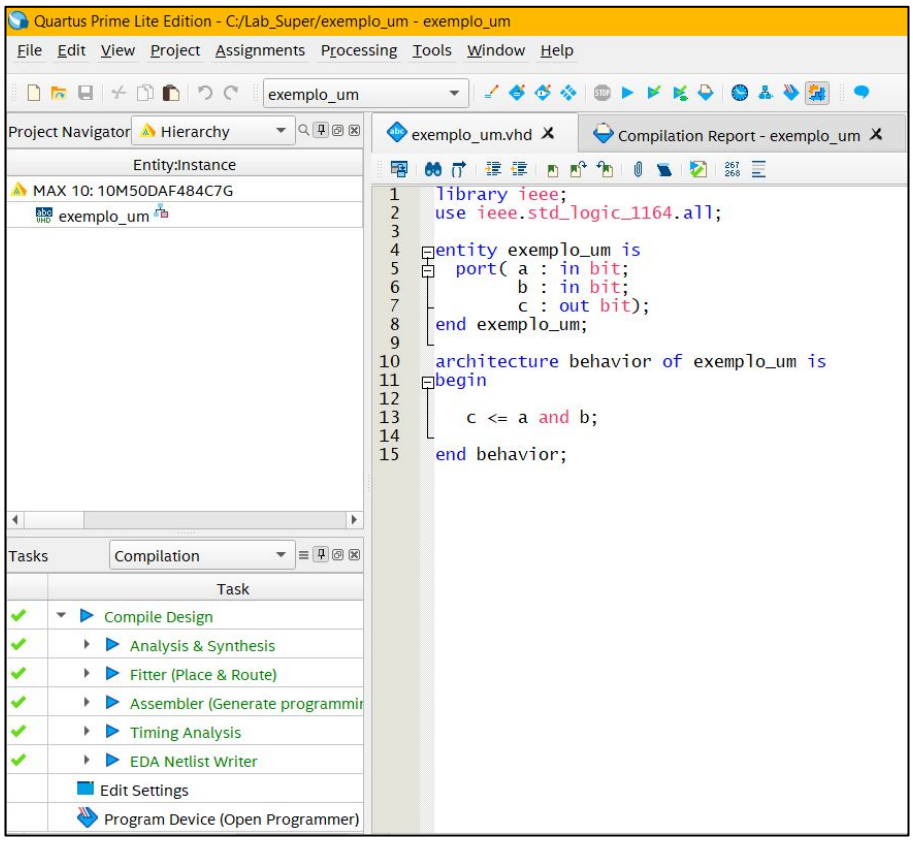


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software
Quartus Prime Version 22.1std Lite Edition

Simulação do código VHDL

Após a criação do arquivo .vhd, compilado com sucesso, iremos realizar a simulação do código escrito, com intuito de verificar de maneira prévia, se o mesmo funcionará conforme desejado.
Observação: Utilizar o código da imagem abaixo como exemplo.

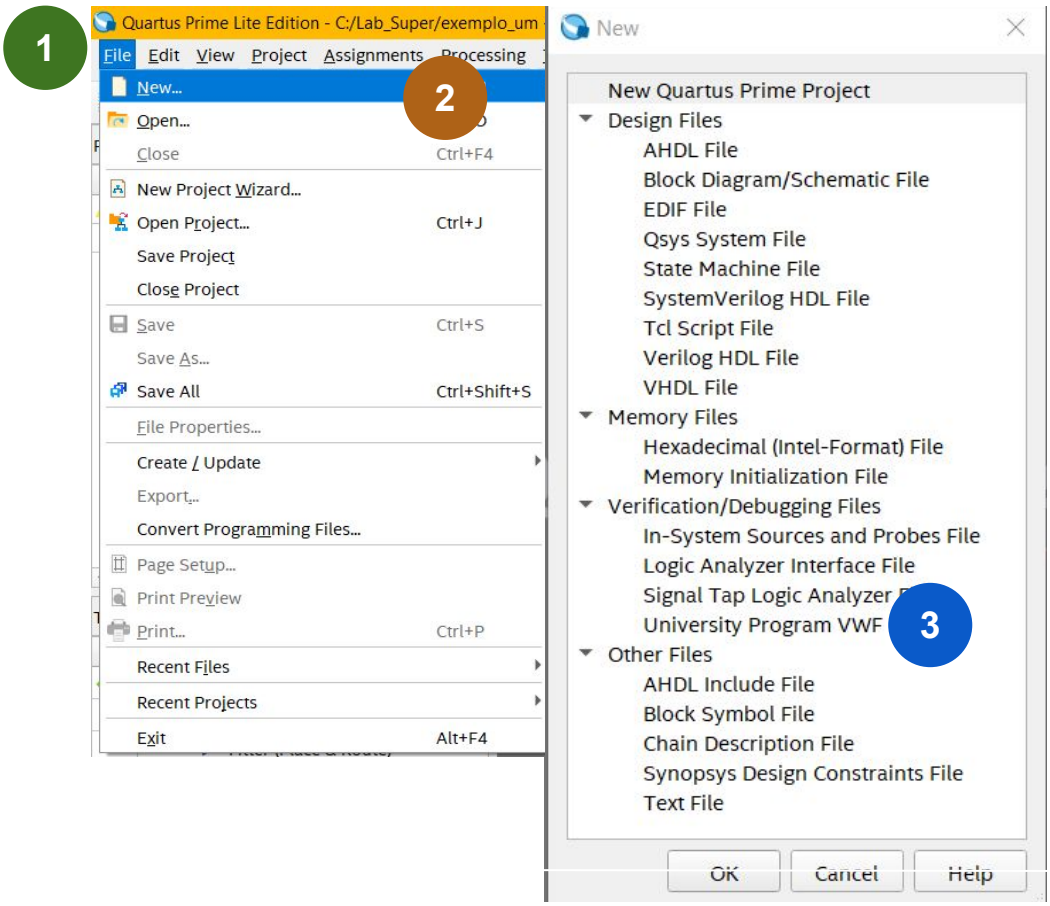


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Simulação do código VHDL

Na barra de ferramentas, clicar em **“File”** e posteriormente em **"New"**. Quando abrir a aba “New”, clicar 2 vezes em **“University Program VWF”**.

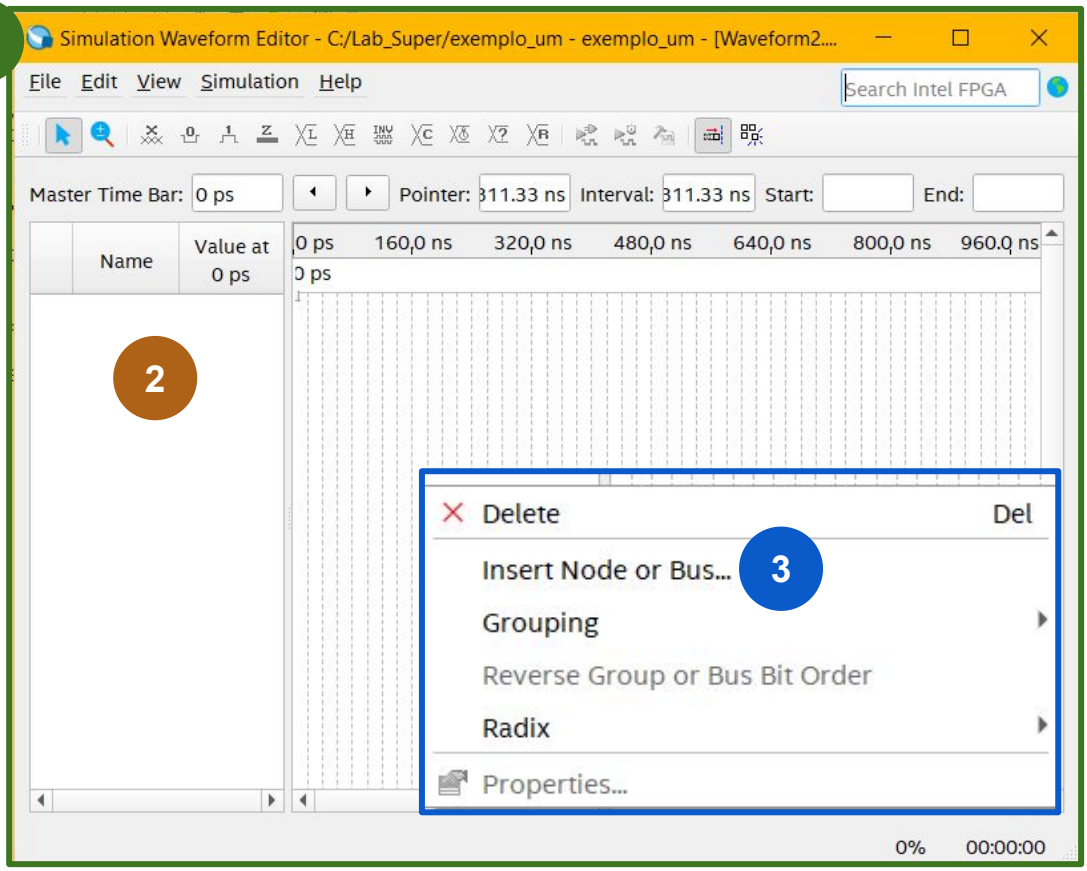


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Simulação do código VHDL

Quando abrir a aba **“Simulator Waveform Editor”**, clicar com o **botão direito na região em branco** mostrada abaixo, posteriormente, clicar em **"Insert Node ou Bus”**.

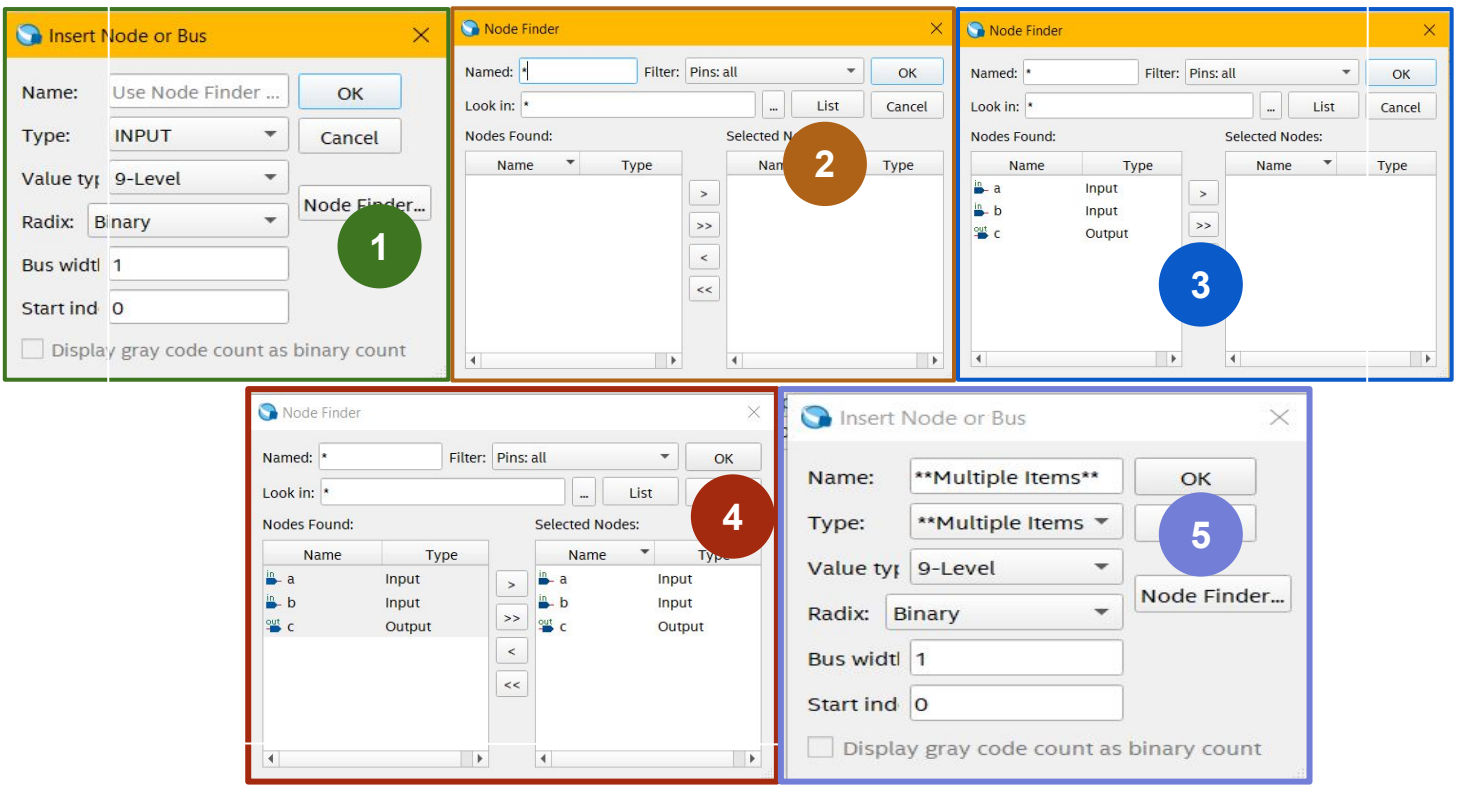


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Simulação do código VHDL

Quando abrir a aba Insert Node or Bus, clicar em **"Node Finder..."**. Na aba Node Finder, iremos adicionar todas as entradas e saídas descritas no código, clicar em **"List"** e quando todas as informações forem adicionadas, clicar em **">>"** e por fim **"OK"** e **"OK"**.

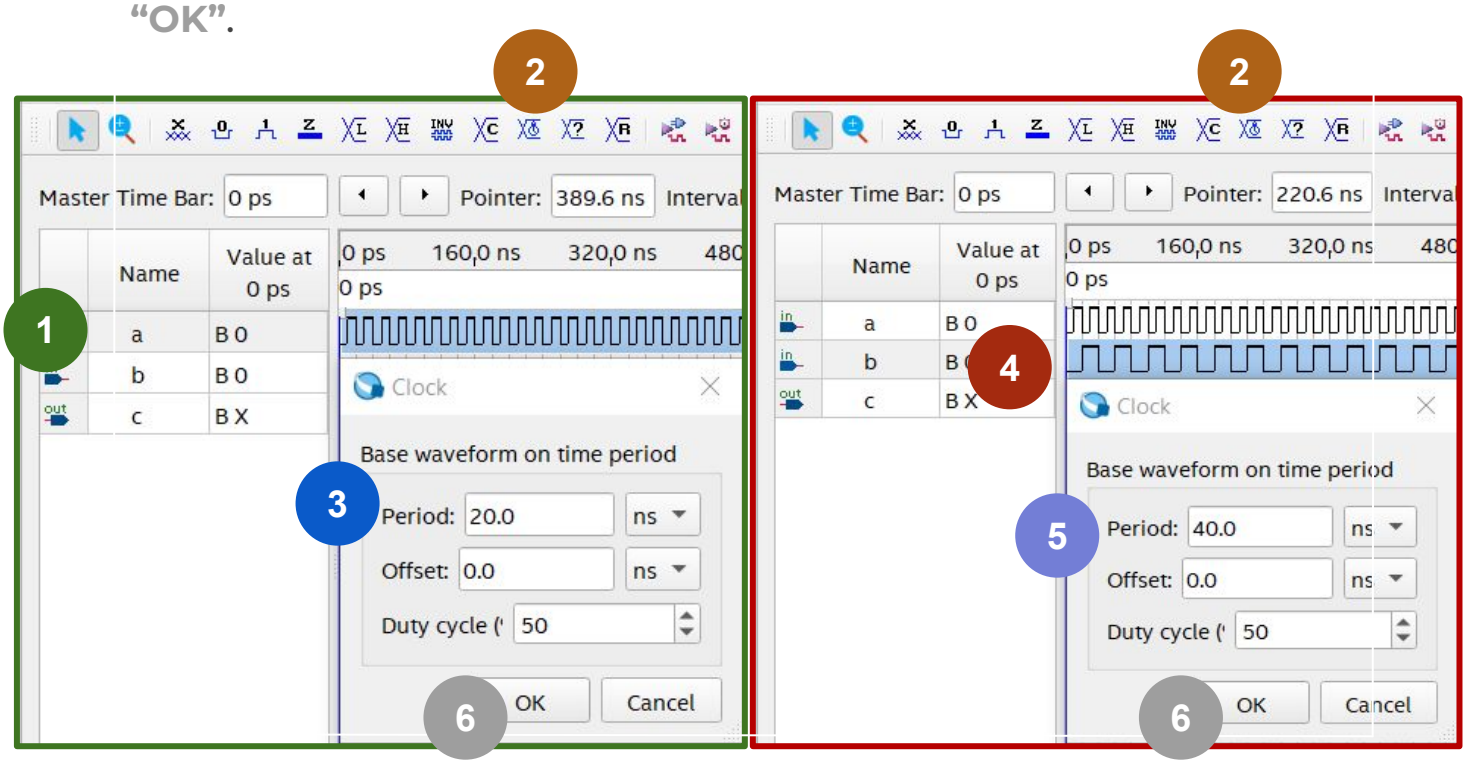


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Simulação do código VHDL

Realizar as configurações de valores de simulação das entradas, nesse caso abaixo, **a** e **b**. Selecionar cada entrada individualmente e clicar em **"Overwrite Clock"**, no qual, adicionaremos simulações de pulsos de forma de onda quadrada com período de **a para 20ns** e **b para 40ns** e clicar "OK".

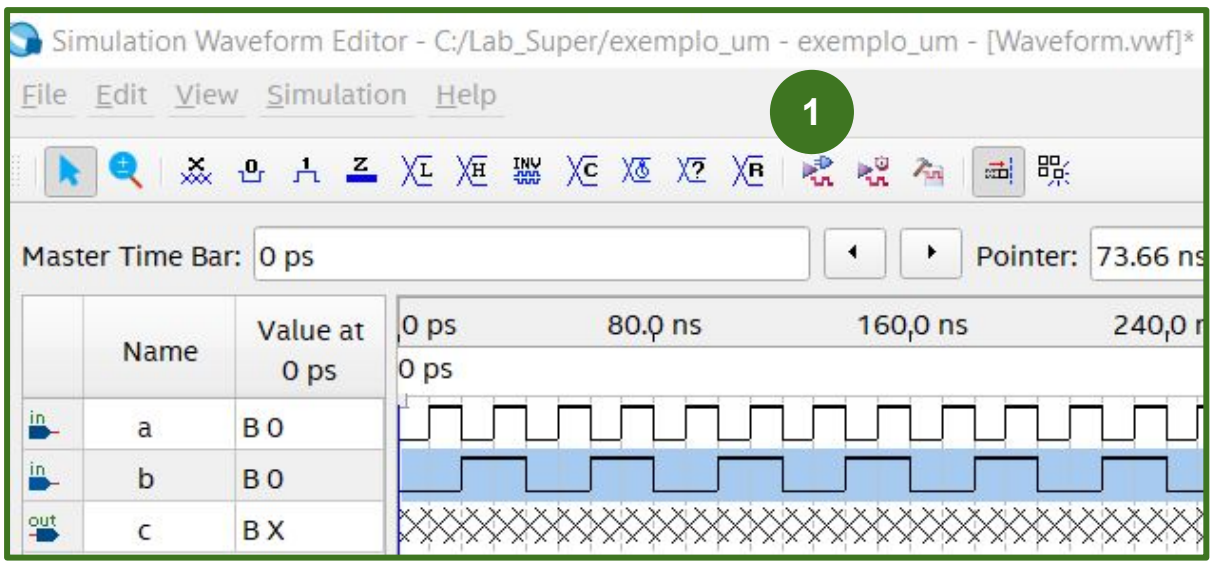


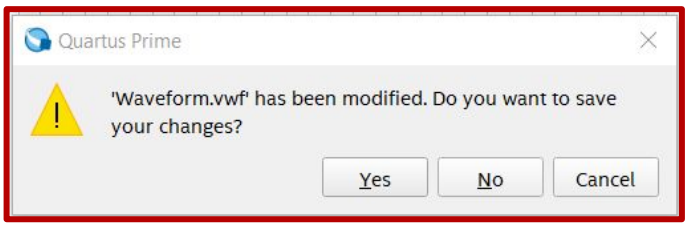
Tutorial de uso do FPGA DE10-LITE

Utilizando o Software Quartus Prime Version 22.1std Lite Edition

Simulação do código VHDL

Terminado as configurações, clicar em **"Run Functional Simulation"**, **salvar o arquivo de simulação** e aguardar o término da aba de compilação, que o resultado da simulação abrirá automaticamente.



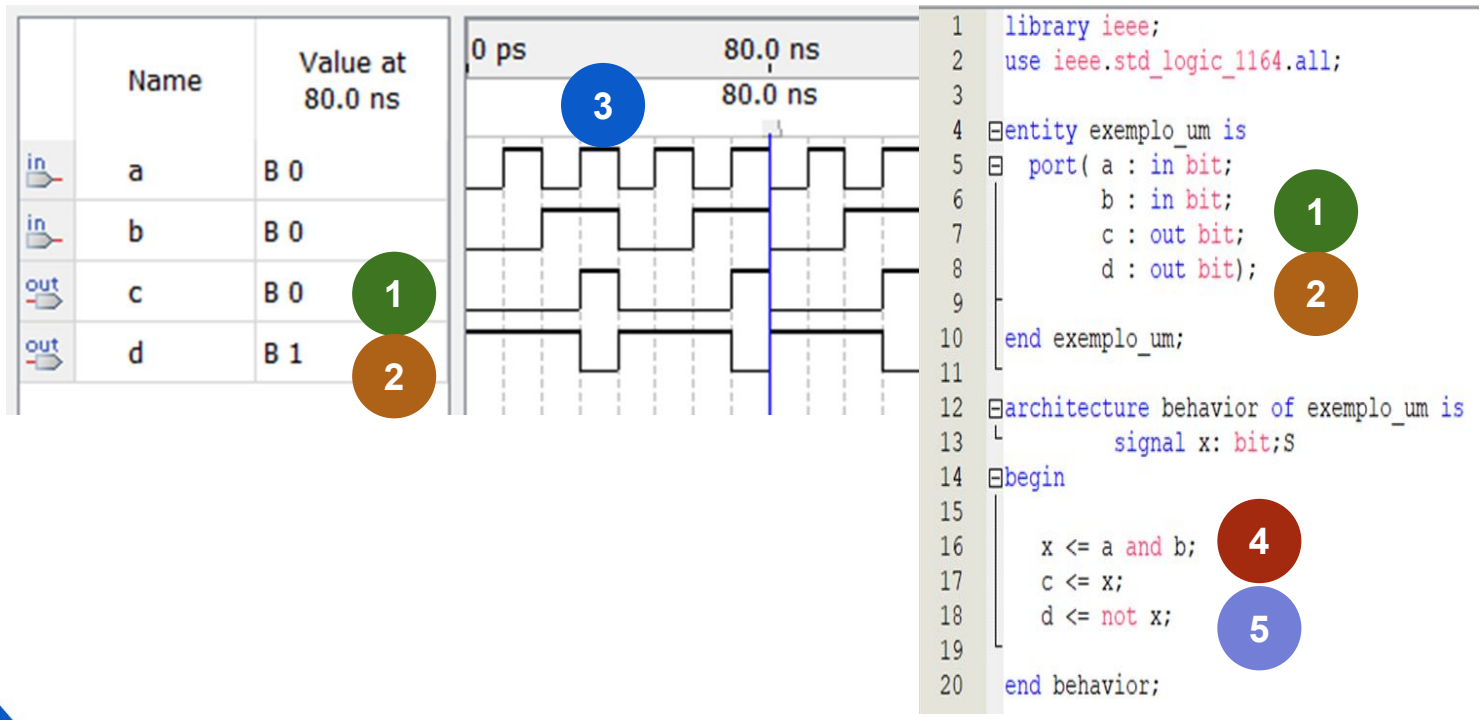


Tutorial de uso do FPGA DE10-LITE

Utilizando o Software
Quartus Prime Version 22.1std Lite Edition

Simulação do código VHDL

Com o arquivo aberto automaticamente no formato no formato .vwf, apenas analisar o resultado. Conforme o código, só temos 2 saídas (**c** e **d**), no qual **c**, ficará em nível **lógico alto**, quando **a** e **b** estiverem ao mesmo tempo em nível **lógico alto**, conforme funcionamento de **portas AND**. E a saída **d**, será apenas o resultado invertido, utilizando o código de **porta NOT**. Confirmando o perfeito funcionamento do código.

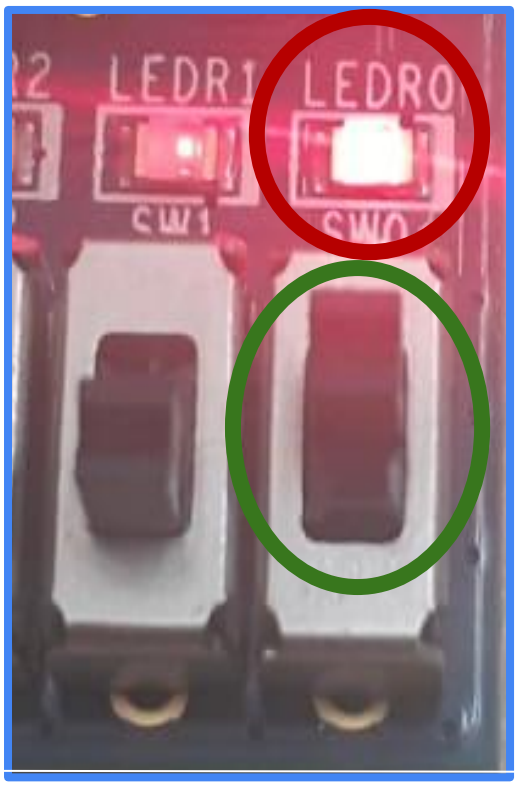


Tutorial de uso do FPGA DE10-LITE

Experimento 1

Pisca LED ativado por chave

O experimento não precisará de nenhum componente além da placa FPGA e do computador. O experimento consiste em acionar via **chave (switch)** o piscar de **led.**



Tutorial de uso do FPGA DE10-LITE

Experimento 1

Pisca LED ativado por chave

O intuito do roteiro não é ensinar VHDL ou VERILOG mas há a necessidade de explicar parte do código.

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity pisca_led is
5     port(
6        sys_clk_50mhz : in std_logic;
7        sys_rst       : in std_logic;
8        led           : out std_logic
9       );
10  end entity pisca_led;
11
12  architecture behavior of pisca_led is
13    signal led_delay : NATURAL range 0 to 50000000;
14    signal led_reg   : std_logic;
15  begin
16    led <= led_reg;
17    process(sys_clk_50mhz, sys_rst)
18     begin
19      if sys_rst = '0' then
20         led_delay <= 0;
21         led_reg <= '0';
22      elsif rising_edge(sys_clk_50mhz) then
23         led_delay <= led_delay + 1;
24         if led_delay = 50000000 then
25           led_delay <= 0;
26           led_reg <= not led_reg;
27         end if;
28      end if;
29    end process;
30  end architecture behavior;
```

Conjunto de sub-programas que descrevem, elementos e componentes já programados para serem reutilizados .

Definição das portas de entrada e saída do código.

Implementação do projeto, descrevendo as relações entre as portas sobre o comportamento do código.

Tutorial de uso do FPGA DE10-LITE

Experimento 1

Pisca LED ativado por chave

Este é o código completo em VHDL. Com tudo pronto pode salvar, **compilar**, realizar o **PIN PLANNER**, **compilar novamente** e embarcar na placa FPGA.

Node Name	Direction	Location
out led	Output	PIN_A8
in sys_clk_50mhz	Input	PIN_P11
in sys_rst	Input	PIN_C10
<<new node>>		

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity pisca_led is
5     port(
6       sys_clk_50mhz : in std_logic;
7       sys_rst       : in std_logic;
8       led           : out std_logic
9         );
10  end entity pisca_led;
11
12  architecture behavior of pisca_led is
13    signal led_delay : NATURAL range 0 to 50000000;
14    signal led_reg   : std_logic;
15    begin
16    led <= led_reg;
17    process(sys_clk_50mhz, sys_rst)
18     begin
19      if sys_rst = '0' then
20         led_delay <= 0;
21         led_reg <= '0';
22      elsif rising_edge(sys_clk_50mhz) then
23         led_delay <= led_delay + 1;
24         if led_delay = 50000000 then
25           led_delay <= 0;
26           led_reg <= not led_reg;
27         end if;
28      end if;
29    end process;
30  end architecture behavior;
```

Tutorial de uso do FPGA DE10-LITE

Experimento 1

Pisca LED ativado por chave

A chave (sys_rst) quando acionado em **nivel lógico alto (1) ligará o led (led) que piscará** conforme implernentado no código utilizando uma frequência de clock interna (**sys_clk_50mhz**).

```
4  entity pisca_led is
5    port(
6      sys_clk_50mhz : in std_logic;
7      sys_rst       : in std_logic;
8      led           : out std_logic
9        );
10 end entity pisca_led;
```

1

0

1

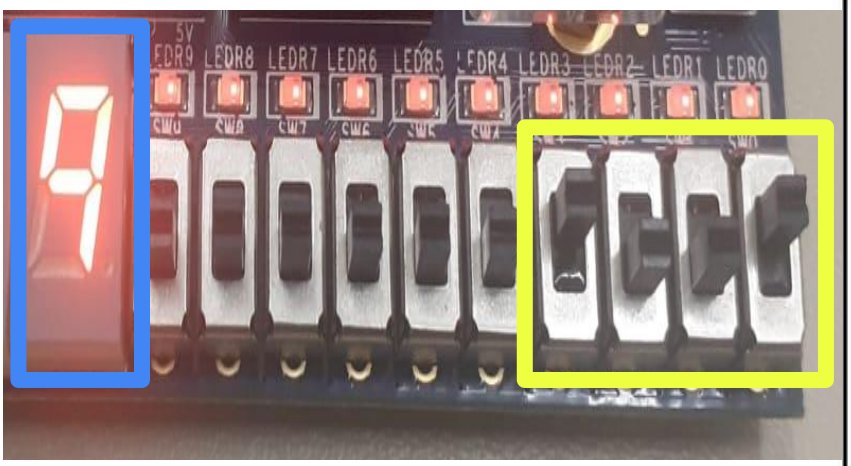
0

Tutorial de uso do FPGA DE10-LITE

Experimento 2

Decodificador BCD para 7 Segmentos

O experimento não precisará de nenhum componente além da placa FPGA e do computador. O experimento consiste implementar **valores BCD via chaves** para decodificar e exibir no **display de 7 segmentos.**



entradas BCD				segmentos de saída							DISPLAY
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9
1	1	1	1	0	0	0	0	0	0	0	

Tutorial de uso do FPGA DE10-LITE

Experimento 2

Decodificador BCD para 7 Segmentos

Este é o código completo em VHDL. Com tudo pronto pode salvar, **compilar**, realizar o **PIN PLANNER**, **compilar novamente** e embarcar na placa FPGA.

DEC_BCD_7SEG.vhd

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use IEEE.numeric_std.all;
4
5  entity DEC_BCD_7SEG is
6  port(
7      BCD      :in  UNSIGNED(3 downto 0);
8      SEGMENT7 :out UNSIGNED(6 downto 0)
9      );
10 end Entity;
11
12 architecture behavior of DEC_BCD_7SEG is
13     signal SEGMENT:UNSIGNED(6 downto 0);
14 begin
15     SEGMENT <= "0111111" when BCD = "0000" else  -- 0
16                "0000110" when BCD = "0001" else -- 1
17                "1011011" when BCD = "0010" else -- 2
18                "1001111" when BCD = "0011" else -- 3
19                "1100110" when BCD = "0100" else -- 4
20                "1101101" when BCD = "0101" else -- 5
21                "1111101" when BCD = "0110" else -- 6
22                "0000111" when BCD = "0111" else -- 7
23                "1111111" when BCD = "1000" else -- 8
24                "1100111" when BCD = "1001" else -- 9
25                "1110111" when BCD = "1010" else -- A
26                "1111100" when BCD = "1011" else -- B
27                "0111001" when BCD = "1100" else -- C
28                "1011110" when BCD = "1101" else -- D
29                "1111001" when BCD = "1110" else -- E
30                "1110001" when BCD = "1111";     -- F
31     SEGMENT7 <= not SEGMENT;
32 end behavior;
```

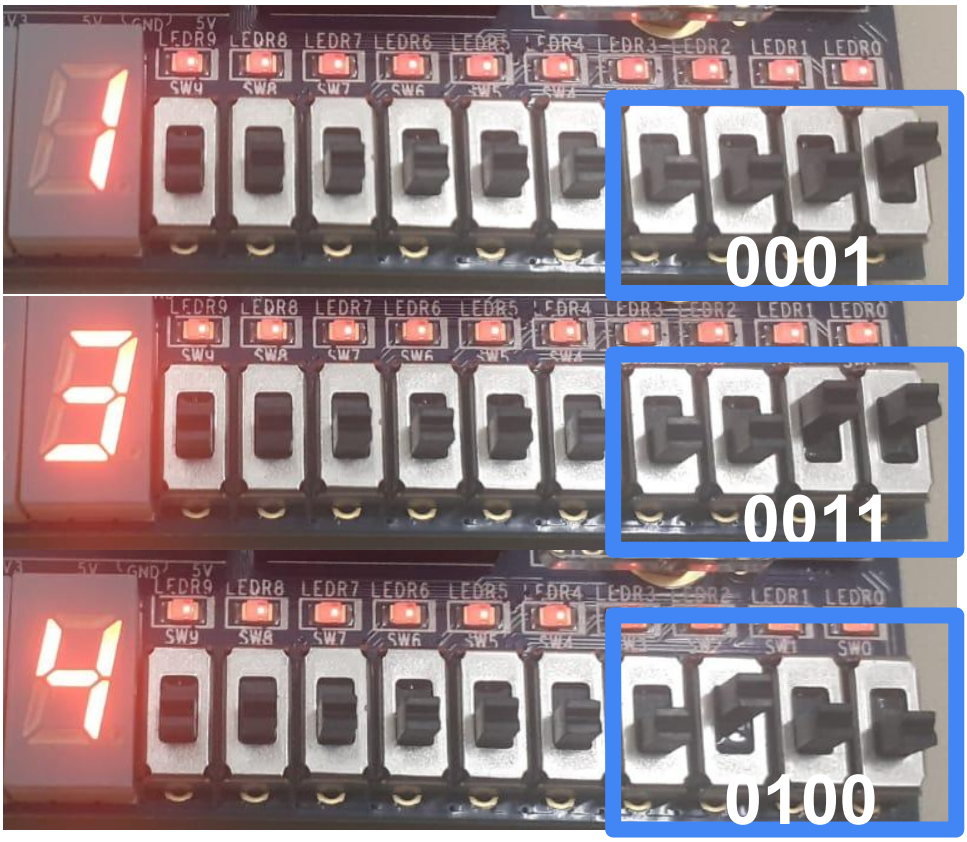
Node Name	Direction	Location
BCD[3]	Input	PIN_C12
BCD[2]	Input	PIN_D12
BCD[1]	Input	PIN_C11
BCD[0]	Input	PIN_C10
SEGMENT7[6]	Output	PIN_C17
SEGMENT7[5]	Output	PIN_D17
SEGMENT7[4]	Output	PIN_E16
SEGMENT7[3]	Output	PIN_C16
SEGMENT7[2]	Output	PIN_C15
SEGMENT7[1]	Output	PIN_E15
SEGMENT7[0]	Output	PIN_C14
<<new node>>		

Tutorial de uso do FPGA DE10-LITE

Experimento 2

Decodificador BCD para 7 Segmentos

Cada **chave** está realizado a um dígito binário BCD (BCD 3 DOWNTO 0), com isso, ao acionarmos cada um de maneira diferente, será exibido um valor diferente, conforme código, **no display de 7 segmentos** (SEGMENT).

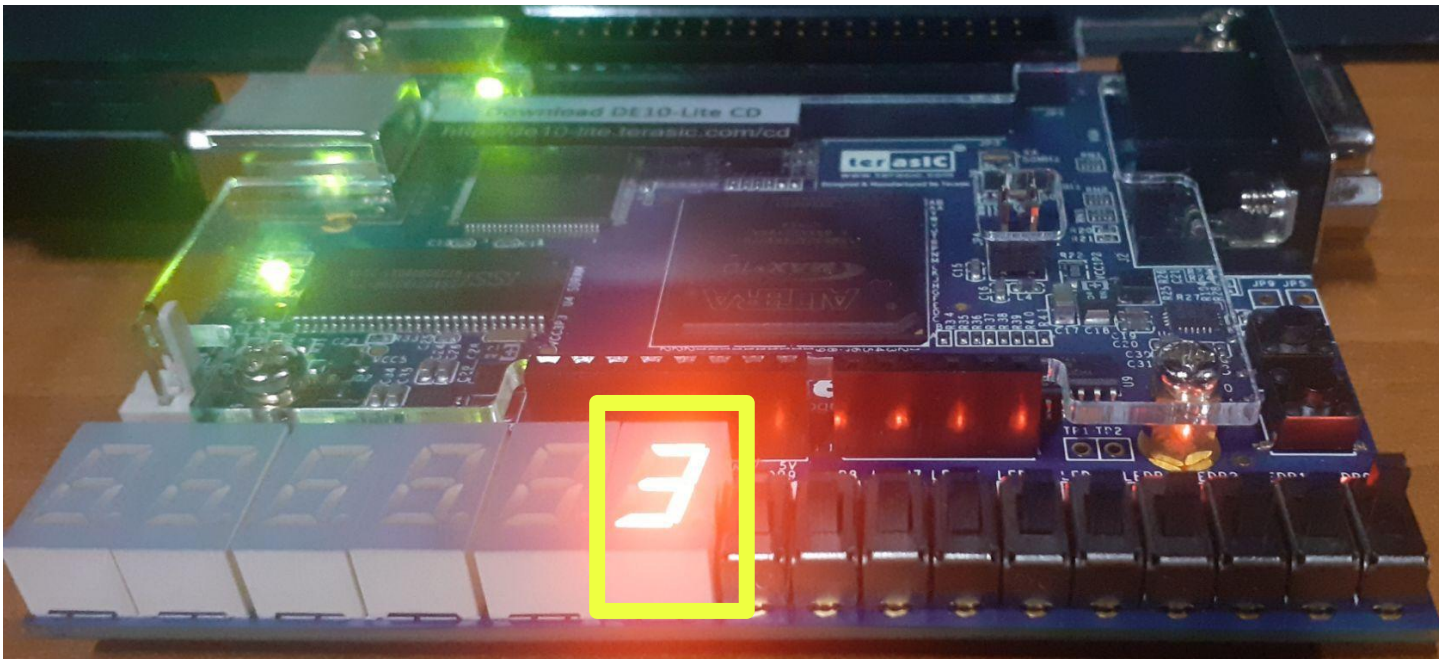


Tutorial de uso do FPGA DE10-LITE

Experimento 3

Contador de 4 dígitos

O experimento não precisará de nenhum componente além da placa FPGA e do computador. O experimento consiste incrementar via pulso de clock, acionado por botão, um contador de 4 dígitos para ser exibido no **display de 7 segmentos.**



Tutorial de uso do FPGA DE10-LITE

Experimento 3

Contador de 4 dígitos

Este é o código completo em VHDL. Com tudo pronto pode salvar, **compilar**, realizar o **PIN PLANNER**, **compilar novamente** e embarcar na placa FPGA.

contador.vhd*

```
1   Library IEEE;
2   use IEEE.std_logic_1164.all;
3   use IEEE.numeric_std.all;
4
5   Entity contador is
6      PORT(RST  : in std_logic;
7           CLK  : in std_logic;
8           Q    : out unsigned(6 downto 0);
9           EN   : in std_logic;
10          CLR  : in std_logic;
11          LD   : in std_logic;
12          LOAD : in unsigned (3 downto 0));
13  end entity;
14  Architecture behavior of contador is
15  Signal CONT: unsigned (3 downto 0);
16  Signal Q1   : unsigned(6 downto 0);
17  Begin
18  Process (CLK, RST)
19     Begin
20     If RST = '1' then
21        CONT <= "0000";
22     Elsif CLK' event and CLK = '1' then
23        If CLR = '1'then
24        CONT <= "0000";
25        Else
26        If EN = '1' then
27           If LD = '1' then
28           CONT <= LOAD;
29           Else
30           CONT <= CONT+1;
31           End IF;
32        End If;
33        End If;
34        End If;
35  End process;
```

Node Name	Direction	Location
CLK	Input	PIN_R24
CLR	Input	PIN_AC28
EN	Input	PIN_AC27
LD	Input	PIN_AD27
LOAD[3]	Input	PIN_AC26
LOAD[2]	Input	PIN_AD26
LOAD[1]	Input	PIN_AB26
LOAD[0]	Input	PIN_AC25
Q[6]	Output	PIN_G18
Q[5]	Output	PIN_F22
Q[4]	Output	PIN_E17
Q[3]	Output	PIN_L26
Q[2]	Output	PIN_L25
Q[1]	Output	PIN_J22
Q[0]	Output	PIN_H22
RST	Input	PIN_AB27

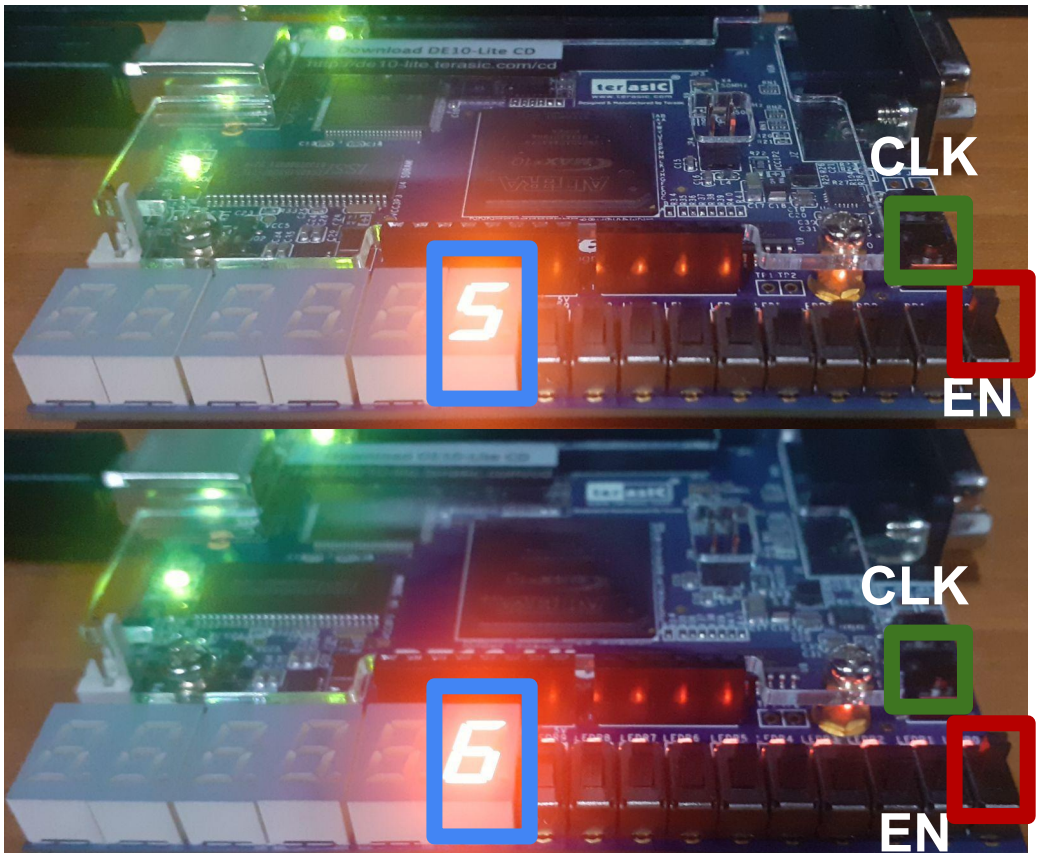
```
End process;
Q1 <= "0111111" when CONT = "0000" else  -- 0
                "0000110" when CONT = "0001" else -- 1
                "1011011" when CONT = "0010" else -- 2
                "1001111" when CONT = "0011" else -- 3
                "1100110" when CONT = "0100" else -- 4
                "1101101" when CONT = "0101" else -- 5
                "1111101" when CONT = "0110" else -- 6
                "0000111" when CONT = "0111" else -- 7
                "1111111" when CONT = "1000" else -- 8
                "1100111" when CONT = "1001" else -- 9
                "1110111" when CONT = "1010" else -- A
                "1111100" when CONT = "1011" else -- B
                "0111001" when CONT = "1100" else -- C
                "1011110" when CONT = "1101" else -- D
                "1111001" when CONT = "1110" else -- E
                "1110001" when CONT = "1111";     -- F
   Q <= not Q1;


End architecture;
```

Tutorial de uso do FPGA DE10-LITE

Experimento 3

Contador de 4 dígitos

Conforme código, a **chave enable (EN) quando acionado**, permite que inicie a contagem após a ativação do **pulso de clock (CLK) via botão**, a cada acionamento do botão, haverá incremento da contagem a ser exibido no **display de 7 segmentos** de 0 à 9 e A. B, C, D, E, F.



Tutorial de uso do FPGA DE10-LITE

Agradecimento

Obrigado por ler o roteiro de uso do FPGA, espero que tenha sido esclarecedor e faça bons experimentos.