



SUPER
ROTEIROS

FPGA DE2-115

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





FPGA DE2-115

Sumário

Histórico	3
Introdução	4
Conhecendo o Hardware	5
A placa	5
O cabo	6
Conexão da placa	7
Utilizando o Software Quartus II 13.0 SP1	5
Criação do projeto	8
Criação do programa VHDL	12
Compilação do arquivo de código .vhd	14
Embarque do arquivo de código .vhd	16
Simulação do código VHDL	22
Experimentos	29
Experimento 1	29
Experimento 2	33
Experimento 3	36





Tutorial de uso do FPGA DE2-115

Histórico

Versão	Data	Descrição	Responsável
1.0	01/12/2022	Criação do tutorial de uso	Hitalo Perseu
2.0	30/07/2023	Mudança de template	Hitalo Perseu
2.1	15/08/2023	Inclusão de Experimentos	Hitalo Perseu
2.2	16/05/2024	Ajuste visual e conceitual	Luiz Neto

Aprovação

Autor

Revisão

Técnico

Coordenador





Tutorial de uso do FPGA DE2-115

Introdução

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

DE2-115



DE10-LITE



DE0-NANO





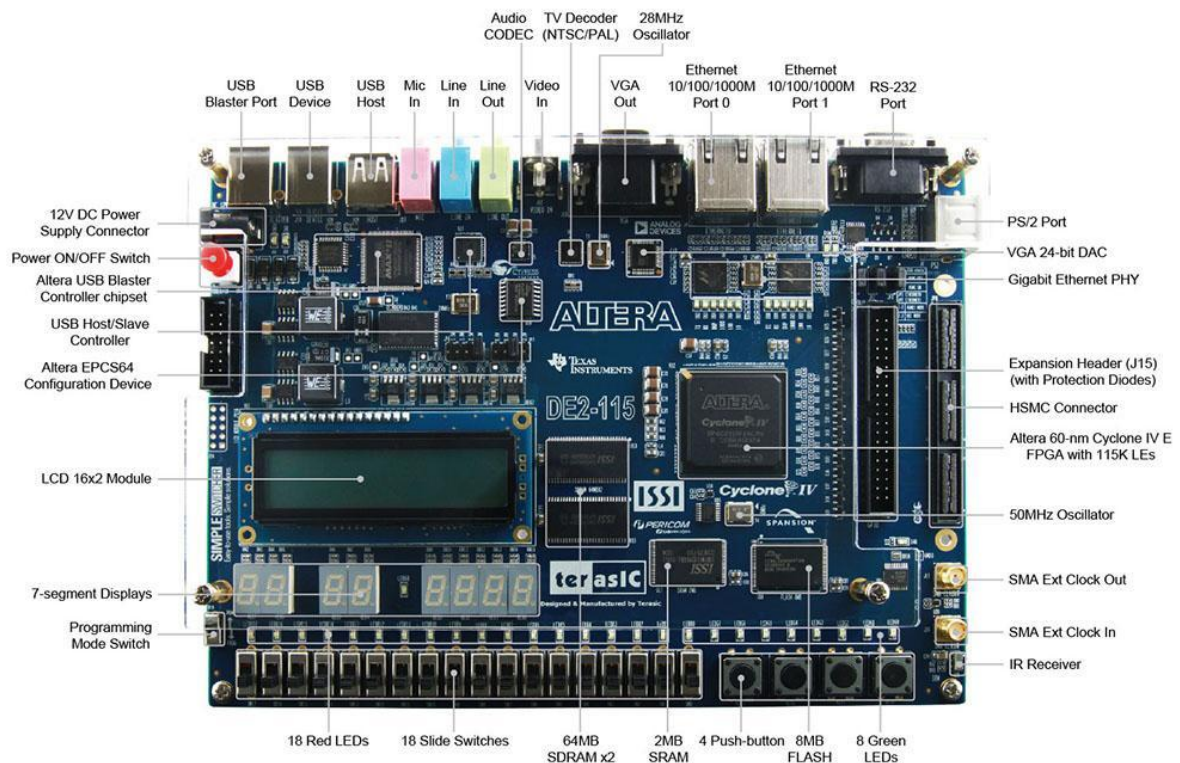
Tutorial de uso do FPGA DE2-115

Conhecendo o hardware

A plataforma

O modelo da placa FPGA que será utilizado de referência para o roteiro será o DE2-115. Este roteiro tem a finalidade de orientar no entendimento, instalação, configuração e testes.

Será necessário, para embarcamos os projetos, o DataSheet da plataforma FPGA DE2-115, para ter acesso [clique aqui](#).





Tutorial de uso do FPGA DE2-115

Conhecendo o hardware

Os cabos

A plataforma FPGA vem acompanhado de um cabo **USB-AB**, para transferir dados. E uma **fonte de alimentação**, para ligar a plataforma.



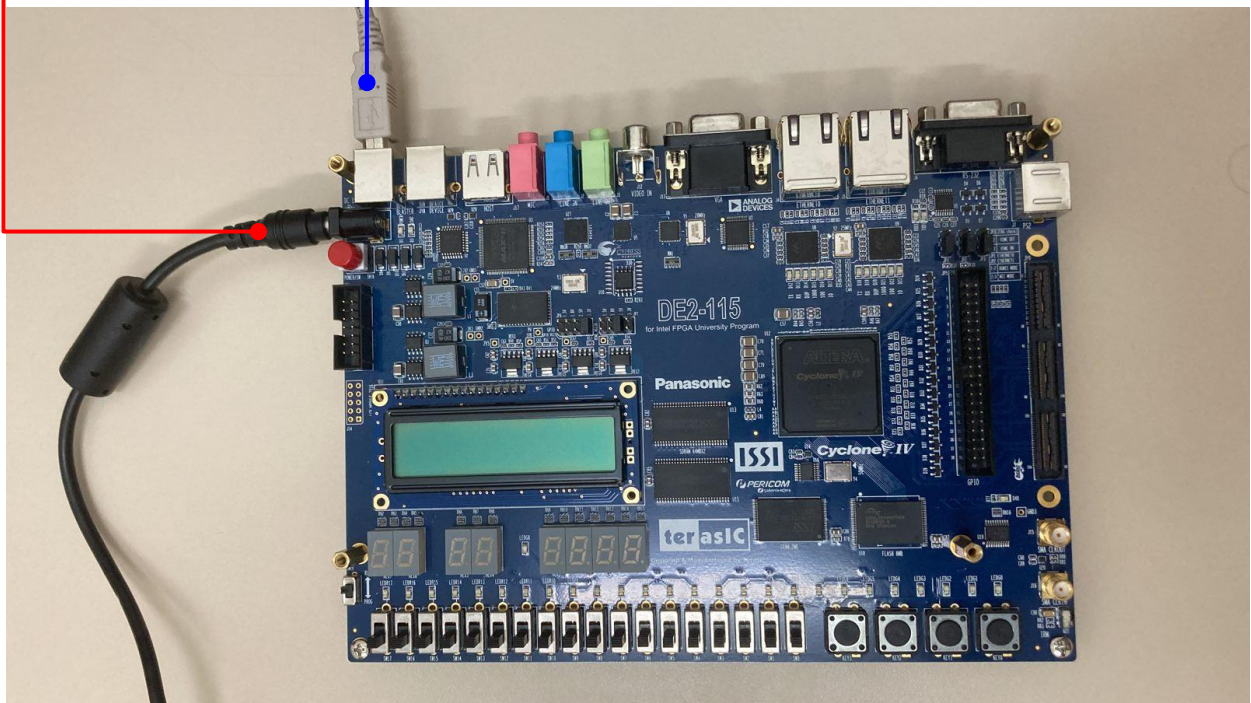


Tutorial de uso do FPGA DE2-115

Conhecendo o hardware

Conexão da plataforma

Para usar o **cabo de alimentação** conecte na FPGA e depois na tomada. Para usar o cabo **USB-AB** conecte na FPGA e depois no computador.



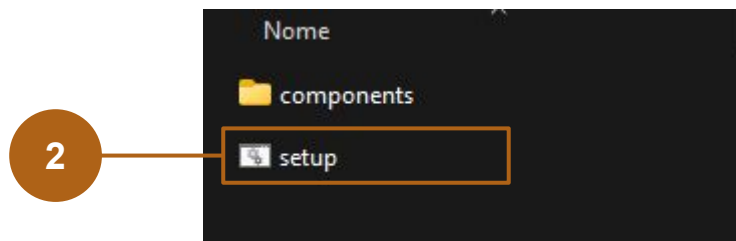
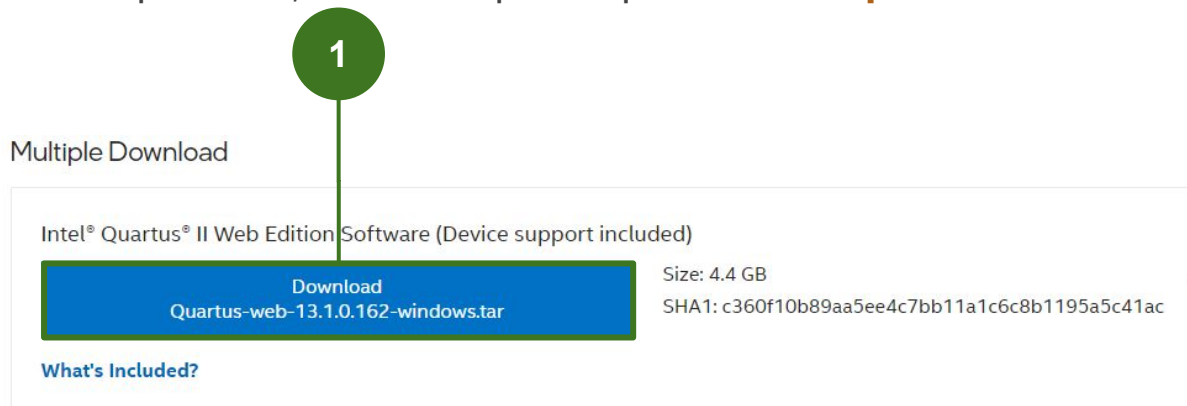


Tutorial de uso do FPGA DE2-115

Instalação do Software Quartus II 13.1.0.162

Download do Software

Para utilizarmos a plataforma FPGA precisamos do Software Quartus II. Para acessar o site do fornecedor, [clique aqui](#). Logo depois, clique no **botão “Download”**. Depois de baixado, descompacte o arquivo **.tar**. Posteriormente, com arquivo descompactado, dê um duplo clique em **“setup”**.



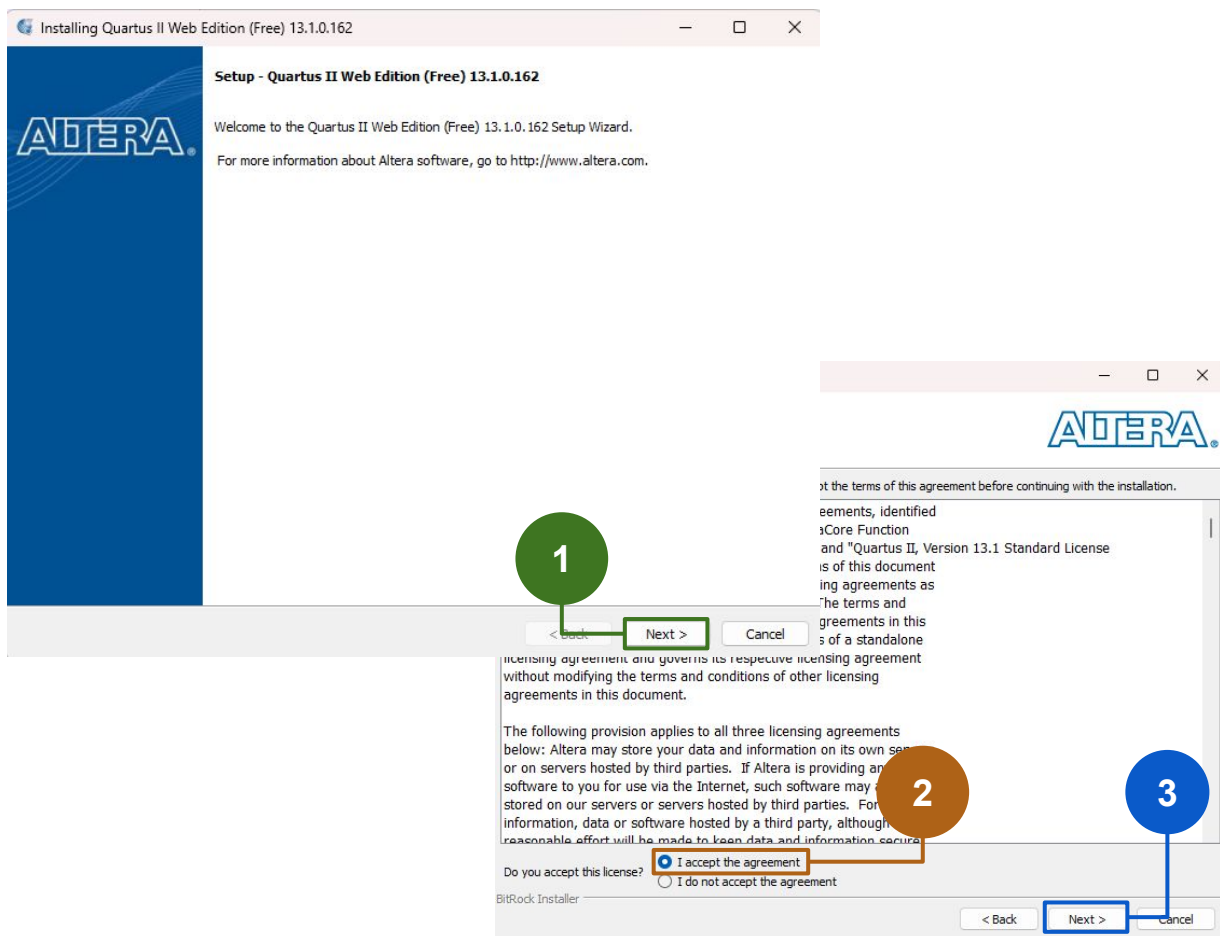


Tutorial de uso do FPGA DE2-115

Instalação do Software Quartus II 13.1.0.162

Configuração do Software

Abrirá a aba de instalação do Quartus, clique em **“Next”**. Logo em seguida, clique em **“I accept the agreement”** e clique em **“Next”** novamente.



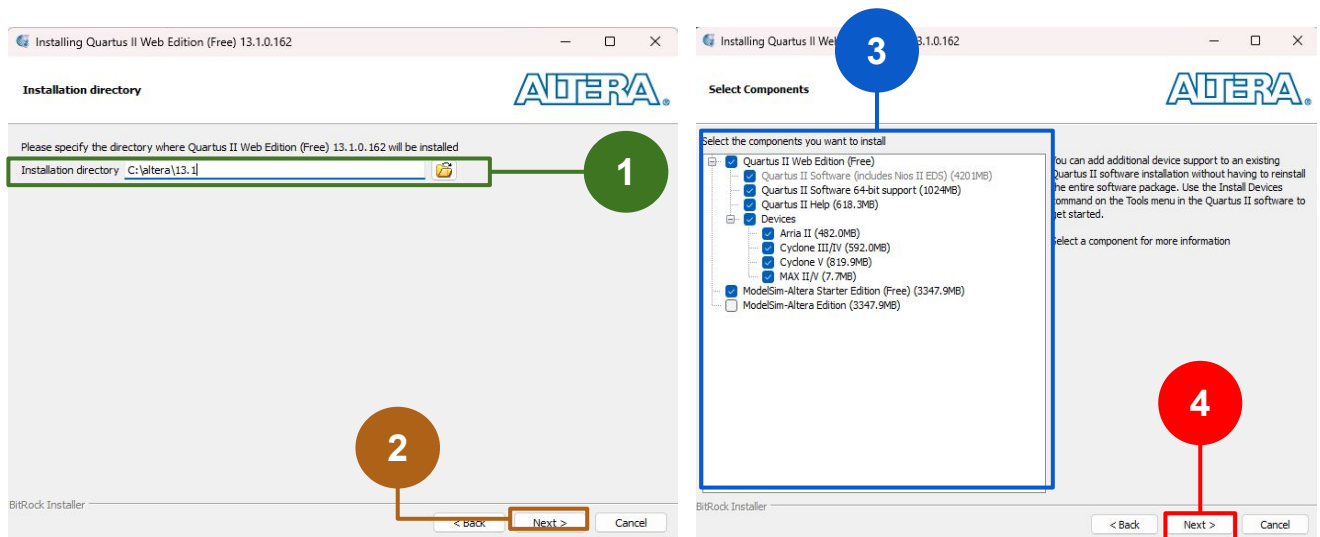


Tutorial de uso do FPGA DE2-115

Instalação do Software Quartus II 13.1.0.162

Configuração do Software

Agora **selecione o Diretório**, local em que deseja instalar o Quartus II, clique em **“Next”**. Na aba seguinte certifique que as **opções estejam marcada** como a seguir e clique **“Next”**.



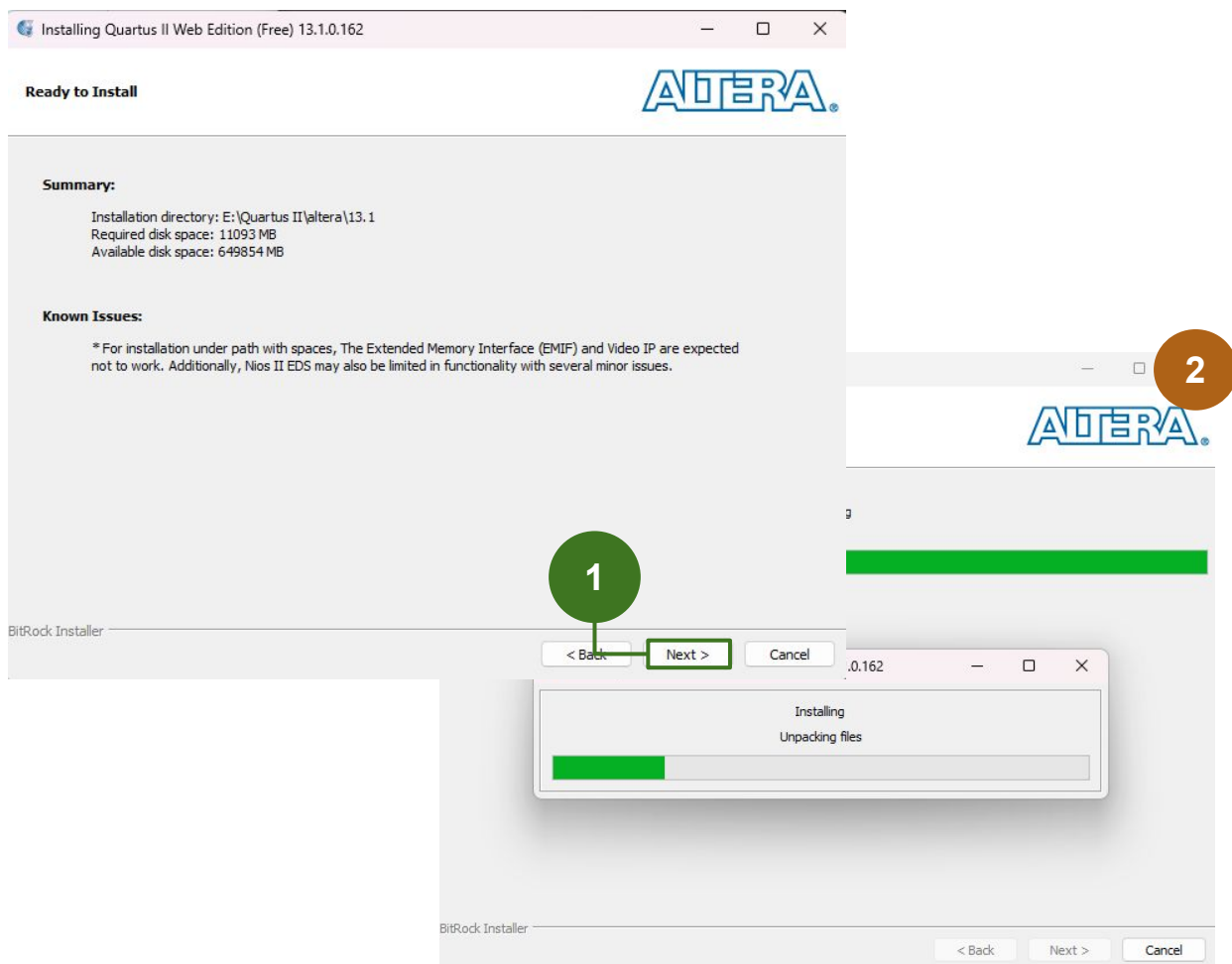


Tutorial de uso do FPGA DE2-115

Instalação do Software Quartus II 13.1.0.162

Configuração do Software

Nesta aba apenas clique em **“Next”** e espere terminar a **instalação**.



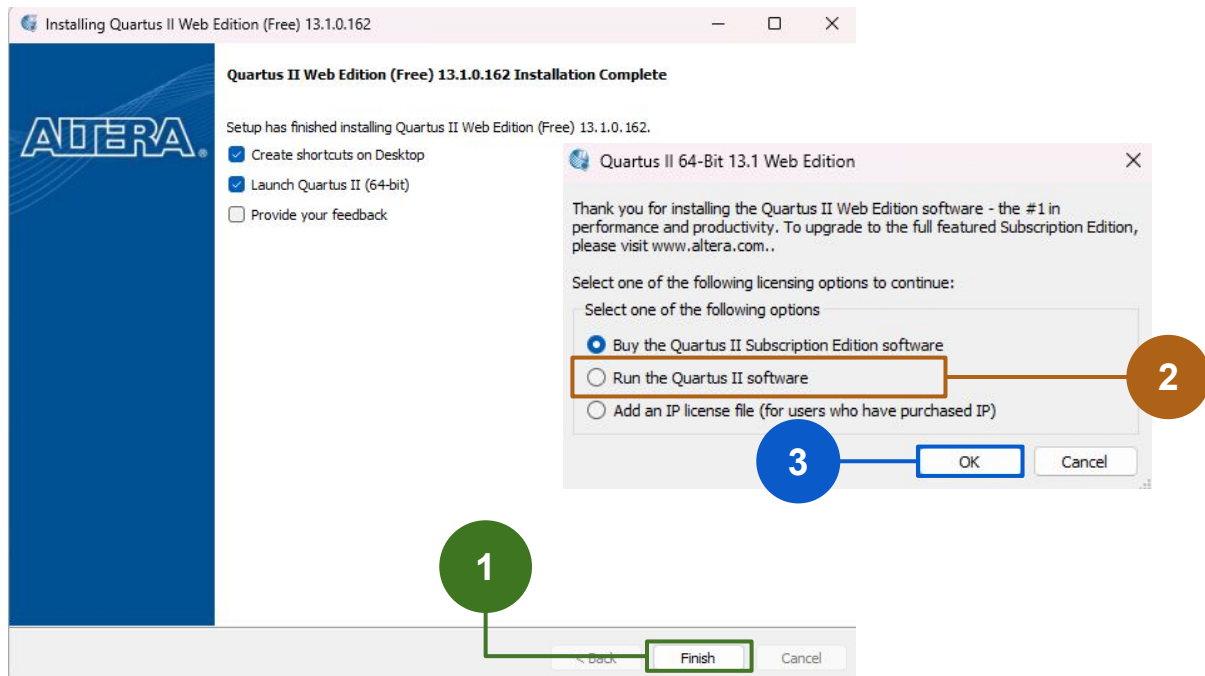


Tutorial de uso do FPGA DE2-115

Instalação do Software Quartus II 13.1.0.162

Configuração do Software

Ao término da instalação abrirá uma nova aba, clique em **“Finish”**. Por fim clique em **“Run the Quartus II software”** e em **“OK”**. Pronto o Quartus II foi instalado com sucesso.



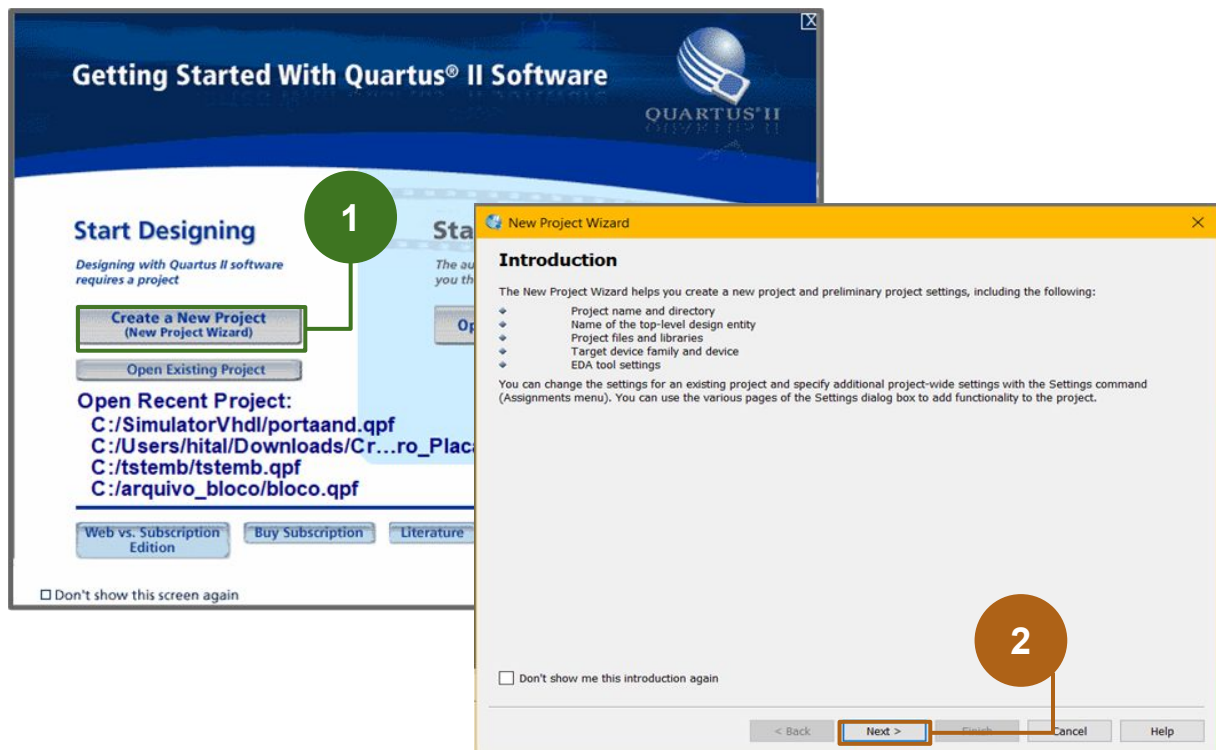


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Criação de Projeto

Ao abrir o Software Quartus, clique em **“Create New Project”**.
Na janela seguinte (Introduction) clique em **“Next”**.



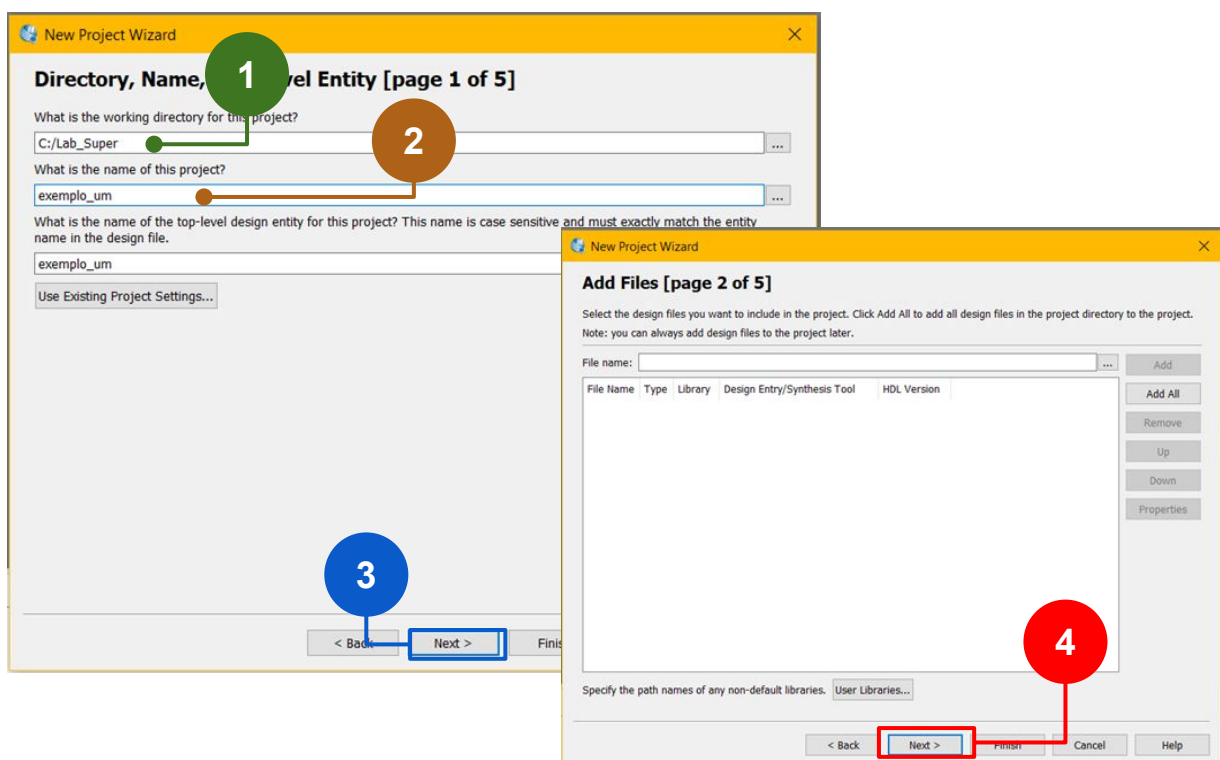


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

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, clique em **“Next”** novamente.



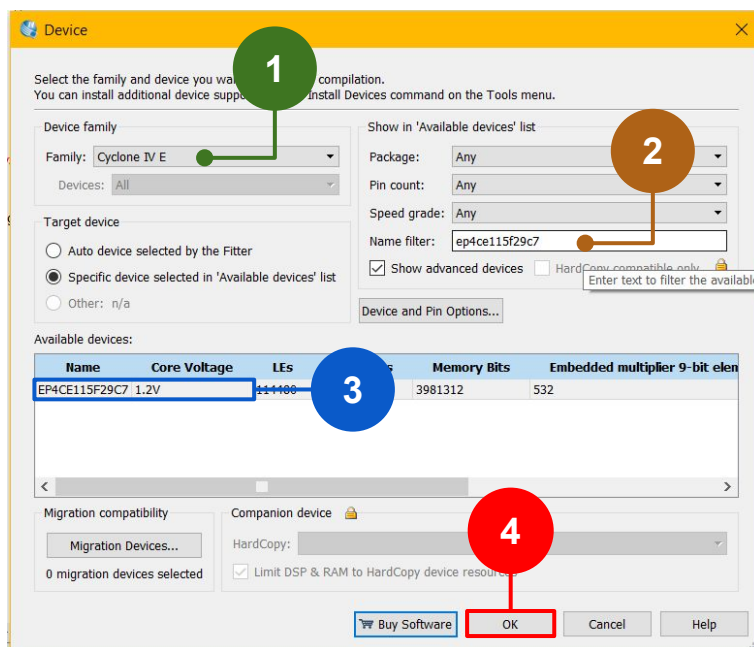


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Criação de Projeto

Nesta janela, insira as configurações da plataforma FPGA, informando a família **Cyclone IV E** e o modelo **EP4CE115F29C7**, selecione o modelo em **Available Devices**. Por fim, clique em **“Ok”**.



As informações de família e o modelo, estão disponíveis no processador da plataforma FPGA.

Exemplo - Família: Cyclone IV E e o modelo: EP4CE115F29C7.





Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Criação de Projeto

Posteriormente, nesta aba de **inserção dos recursos do projetos**, clique em **“Next”**. Na última aba, visualizaremos o **resumo das informações** incluídas no projeto, clique em **“Finish”**.

1

2

3

4

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synt...	<None>	<None>	<input type="checkbox"/> Run the
Simulation	<None>	<None>	<input type="checkbox"/> Run ga
Formal Verification	<None>	<None>	
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

- Project directory: C:/altera/13.1/experimento_um
- Project name: experimento_um
- Top-level design entity: experimento_um
- Number of files added: 0
- Number of user libraries added: 0
- Device assignments:
 - Family name: Cyclone IV E
 - Device: EP4CE115F29C7
- EDA tools:
 - Design entry/synthesis: <None> (<None>)
 - Simulation: ModelSim-Altera (VHDL)
 - Timing analysis: 0
- Operating conditions:
 - VCCINT voltage: 1.2V
 - Junction temperature range: 0-85 °C



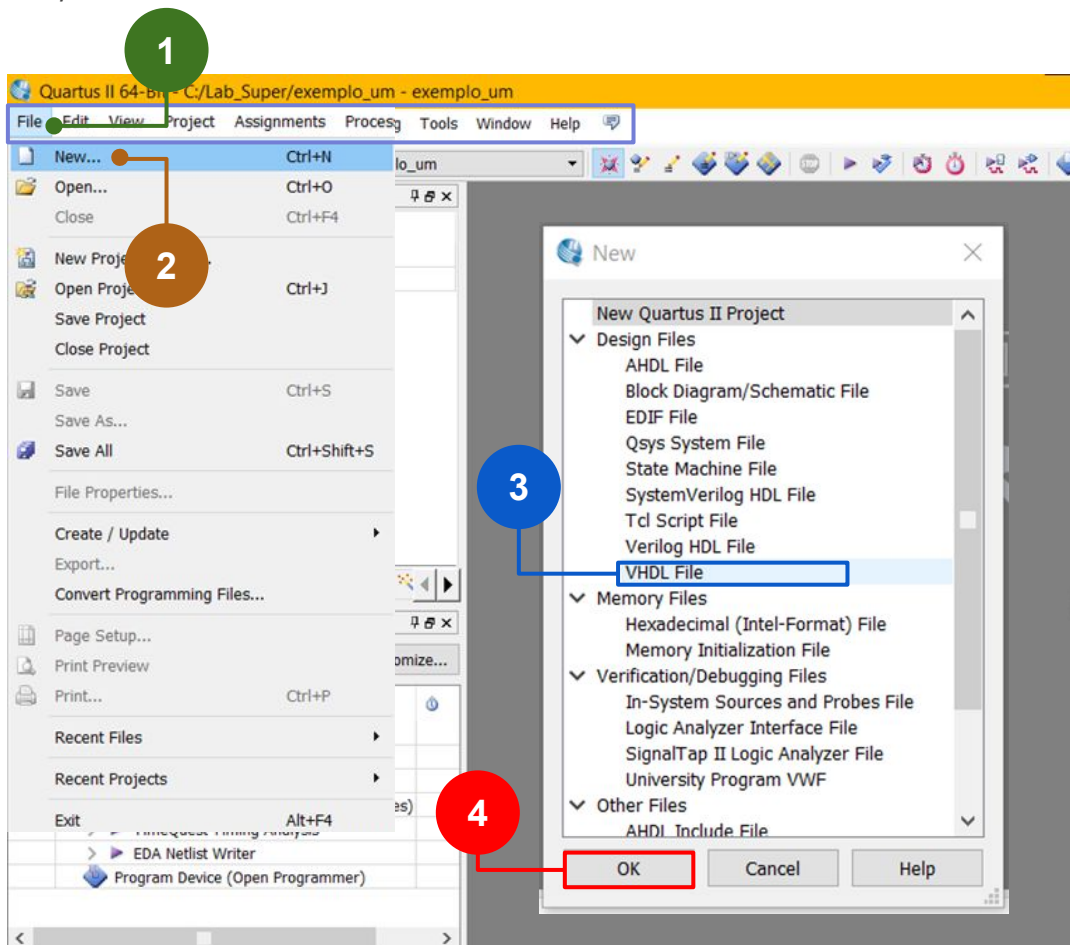


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

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”**.



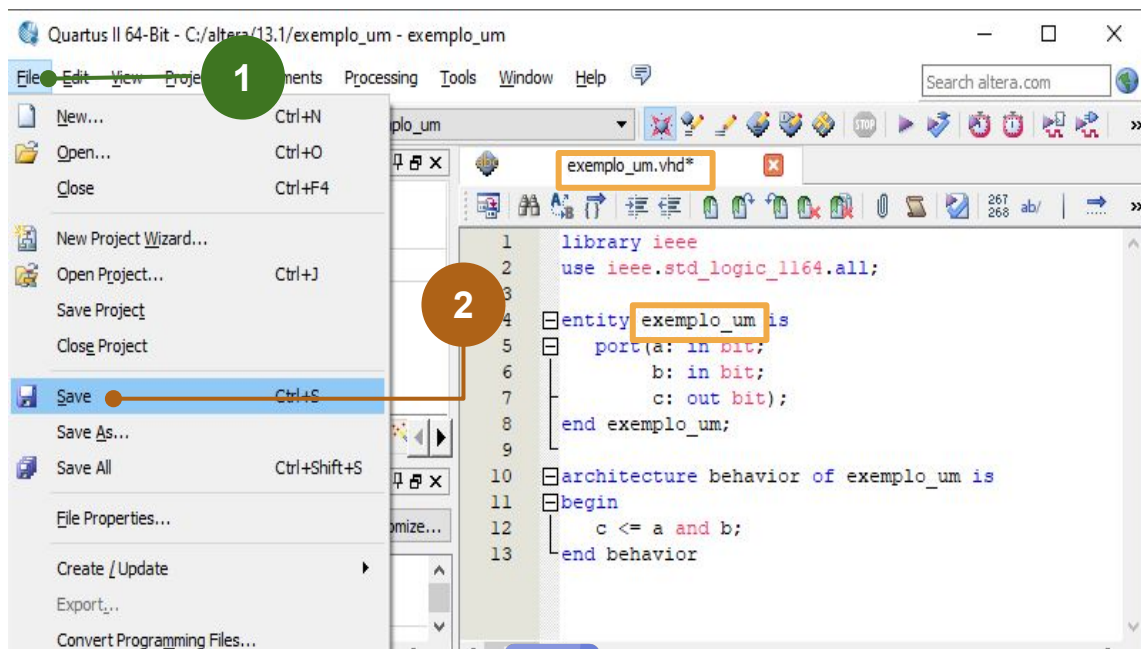


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

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 nada no código.

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



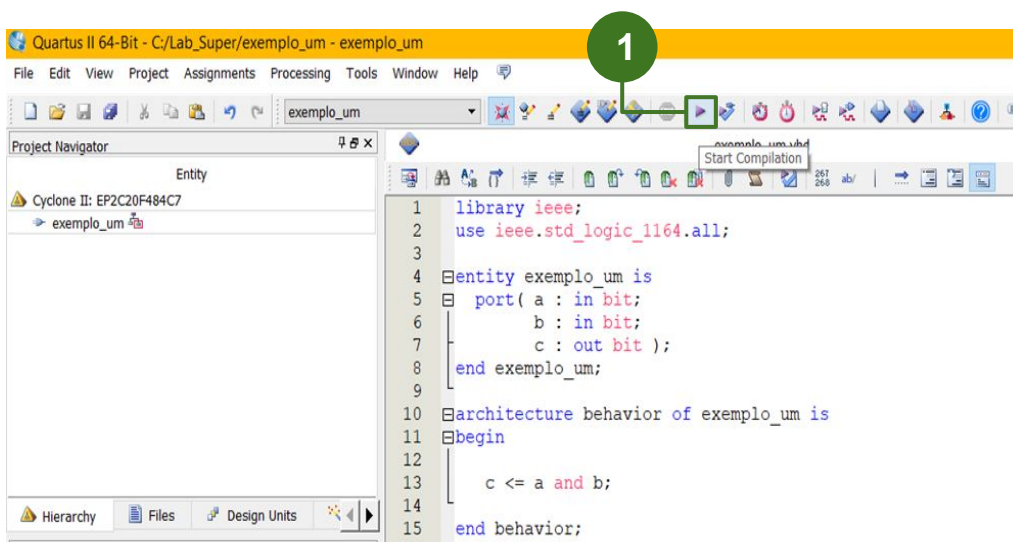


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

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 DE2-115

Utilizando o Software Quartus II 13.1.0.162

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.

The screenshot displays the Quartus II software interface during the compilation of a .vhd file. The 'Flow Summary' window shows the following details:

Flow	Status
Flow Status	Successful - Mon May 20 16:51:10 2024
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name	exemplo_um
Top-level Entity Name	exemplo_um
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	1 / 114,480 (< 1 %)
Total combinational functions	1 / 114,480 (< 1 %)
Dedicated logic registers	0 / 114,480 (0 %)
Total registers	0
Total pins	4 / 529 (< 1 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

The 'Tasks' window shows the following tasks completed:

- Analysis & Synthesis
- Fitter (Place & Route)
- Assembler (Generate programming file)
- TimeQuest Timing Analysis
- EDA Netlist Writer
- Program Device (Open Programmer)

The 'Messages' window shows the following messages:

```
Running Quartus II 64-Bit EDA Netlist Writer
Command: quartus_eda --read_settings_files=off --write_settings_files=off exemplo_um -c exemplo_um
204019 Generated file exemplo_um.vo in folder "C:/altera/13.1/simulation/qsim/" for EDA simulation tool
Quartus II 64-Bit EDA Netlist Writer was successful. 0 errors, 0 warnings
293000 Quartus II Full Compilation was successful. 0 errors, 11 warnings
```



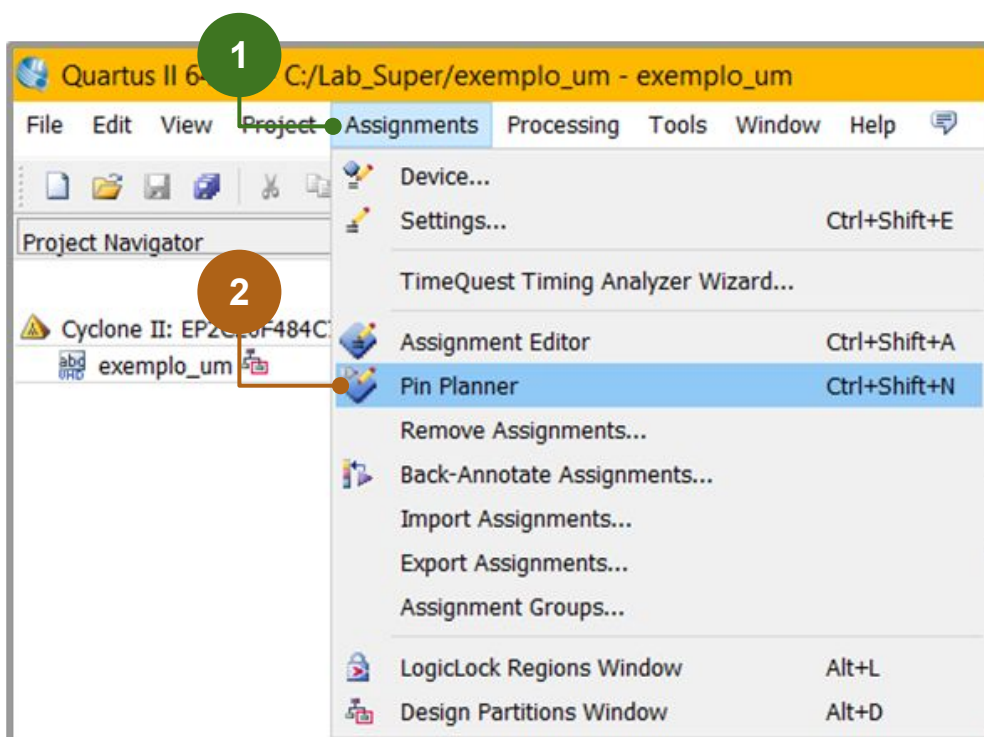


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

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 DE2-115

Utilizando o Software Quartus II 13.1.0.162

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.

Top View - Wire Bond
Cyclone IV E - EP4CE115

Signal Name	FPGA Pin No.
SW[0]	PIN_AB28
SW[1]	PIN_AC28
LEDR[0]	PIN_G19

Node Name	Direction	Location
in a	Input	PIN_AB28
in b	Input	PIN_AC28
out c	Output	PIN_G19
<<new node>>		

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair
in a	Input	PIN_AB28	5	B5_N1	PIN_AB28	3.3-V L...efault		8mA (default)		
in b	Input	PIN_AC28	5	B5_N2	PIN_AC28	3.3-V L...efault		8mA (default)		
out c	Output	PIN_G19	7	B7_N2	PIN_G19	3.3-V L...efault		8mA (default)	2 (default)	
<<new node>>										



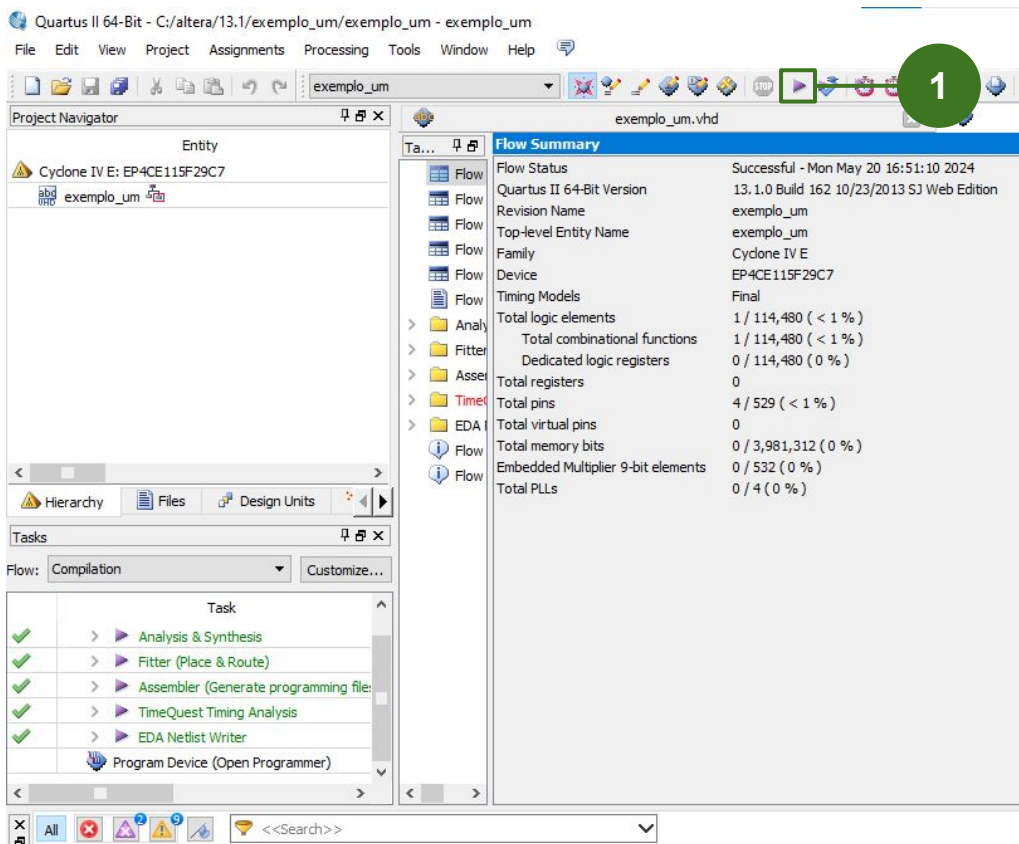


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Mapeamento dos Pinos da FPGA

Após isso, deverá ser 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.



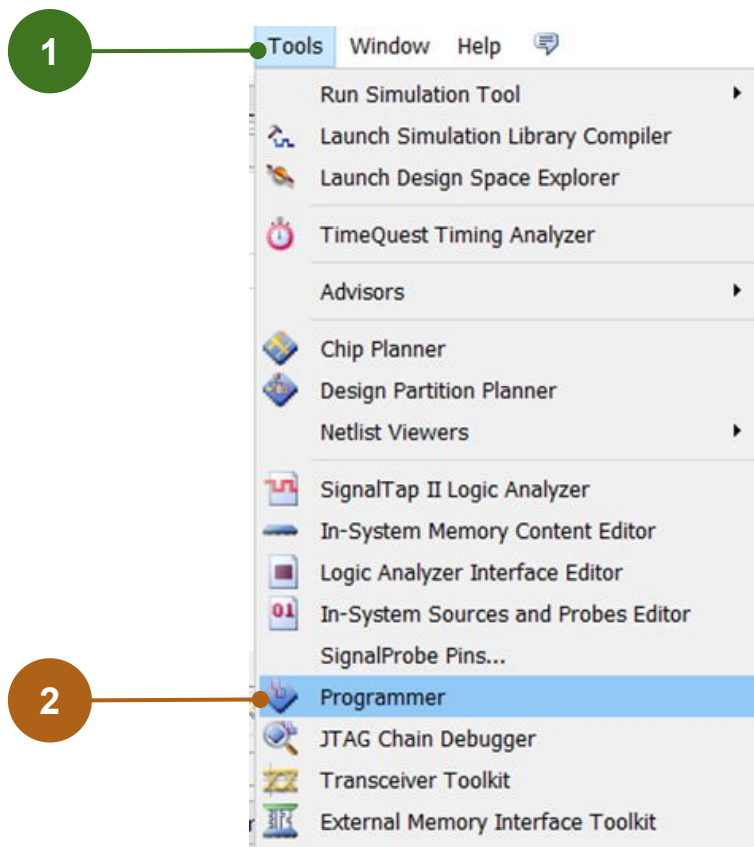


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Embarque do arquivo de código .vhd

Depois devemos ir na barra de ferramentas, clicar em **“Tools”** e posteriormente clicar em **“Programmer”**.



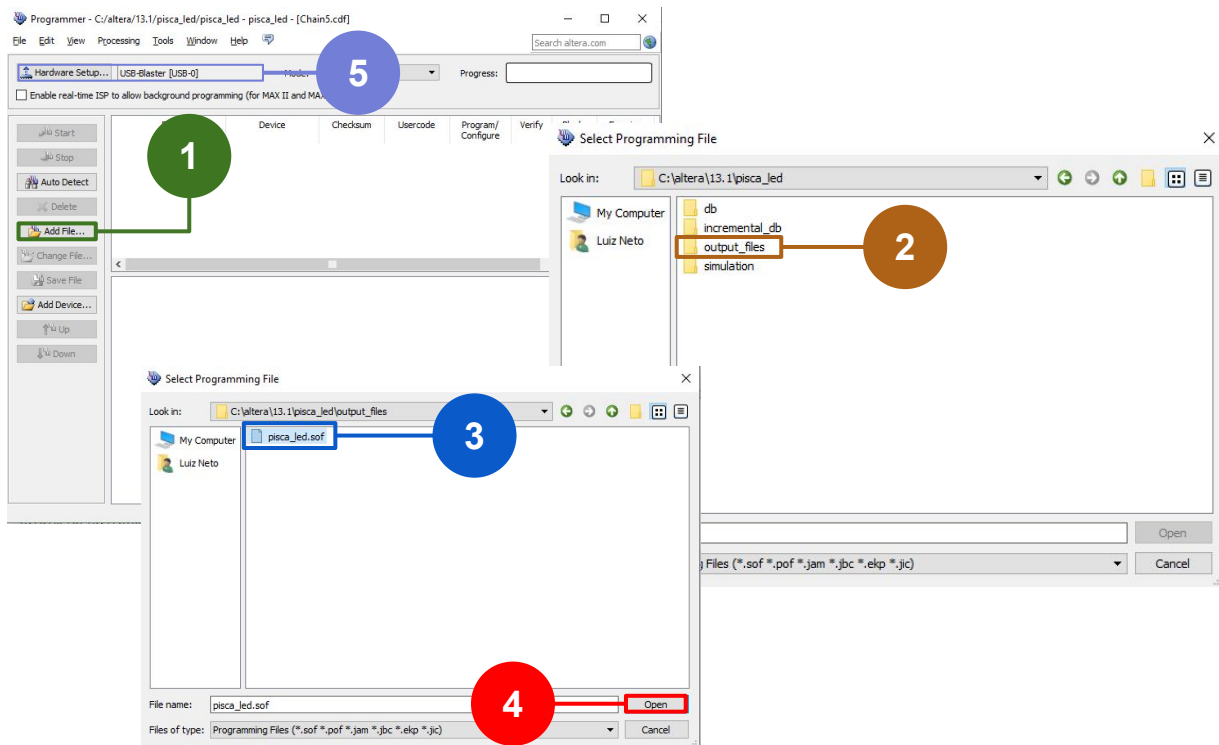


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

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 DE2-115

Utilizando o Software Quartus II 13.1.0.162

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.

The image displays two screenshots of the Quartus II Programmer software interface. The first screenshot shows the 'Start' button highlighted with a green circle (1) and the progress bar at 100% (Successful) with a green circle (2). The second screenshot shows the 'Hardware Setup' window with the 'USB-Blaster [USB-0]' selected (3) and the 'output_files/exemplo_um.sof' file highlighted in the file list with a red circle (4).

File	Device	Checksum	Usercode	Program/Configure
output_files/exemplo_um.sof	EP2C20E484	001B04DB	001B04DB	<input checked="" type="checkbox"/>





Tutorial de uso do FPGA DE2-115

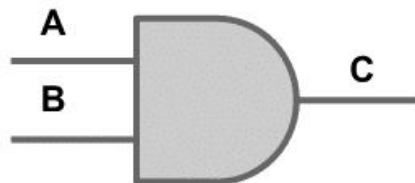
Utilizando o Software Quartus II 13.1.0.162

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 LEDRO(PIN_G19)** irá representar a saída C. O código deve seguir a seguinte tabela verdade:

PORTA E (AND)

$C=A \cdot B$



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



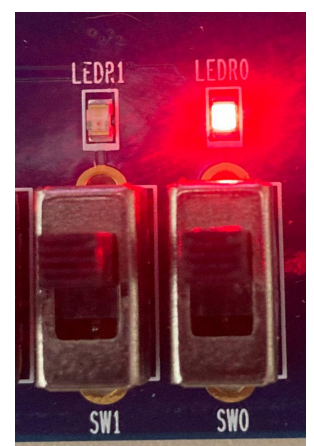
0 0



0 1



1 0



1 1





Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Simulação gráfica do código VH

Nesta parte faremos a simulação gráfica com o intuito de verificar o funcionamento desejado. Iremos utilizar o seguinte código:

```
library                                ieee;
use                                    ieee.std_logic_1164.all;

entity                                exemplo_um                                is
    port (
        a:                                in                                bit;
        b:                                in                                bit;
        c:                                out                               bit;
        d:                                out                               bit);
end                                    exemplo_um;

architecture behavior of exemplo_um is
    signal x:                                bit;
begin
    x <= a and b;
    c <= x;
    d <= not x;
end behavior;
```



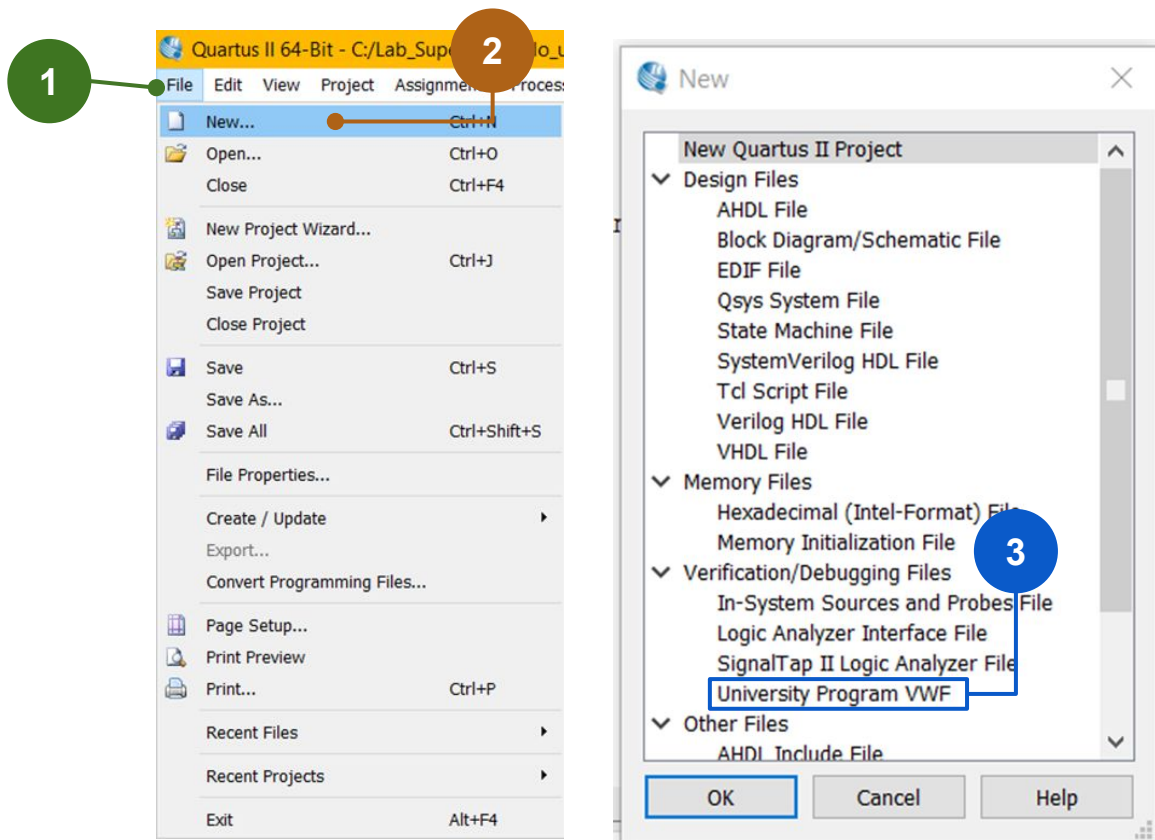


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Simulação gráfica do código VHDL

Para fazermos a simulação gráfica do código iremos na barra de ferramentas, clicar em **"File"** e posteriormente em **"New"**. Quando abrir a aba "New", clicar 2 vezes em **"University Program VWF"**.



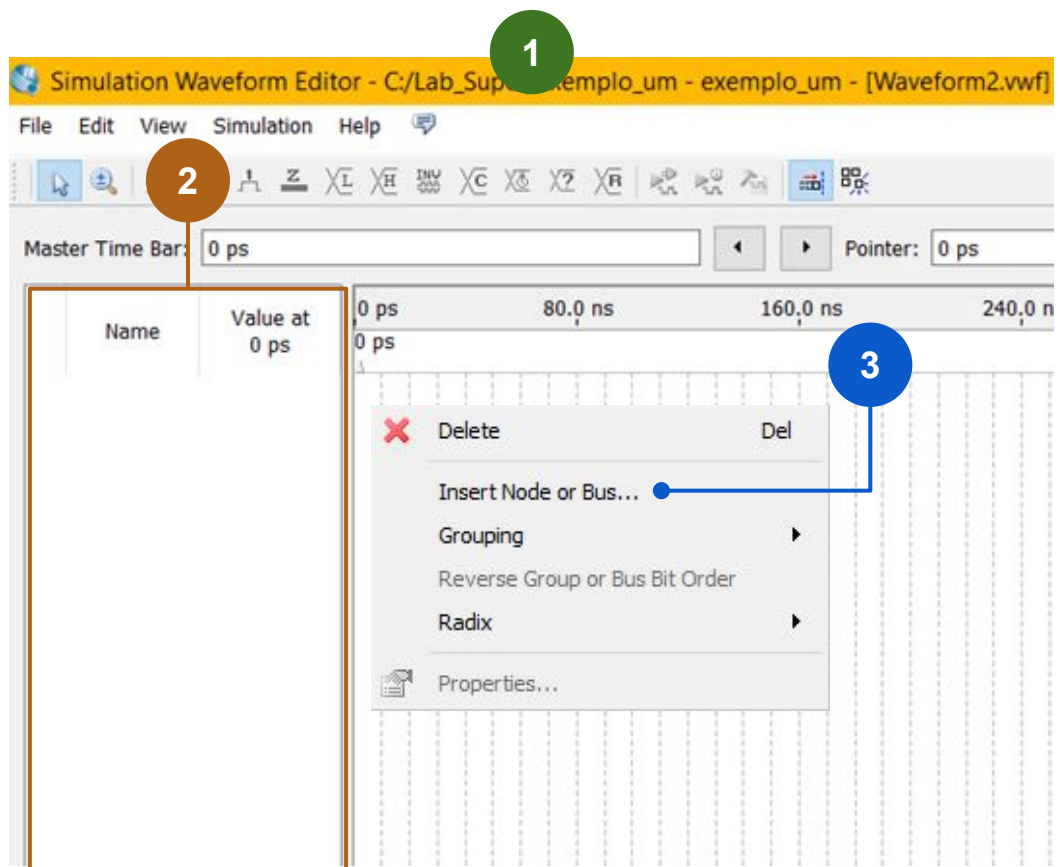


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Simulação gráfica do código VHDL

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



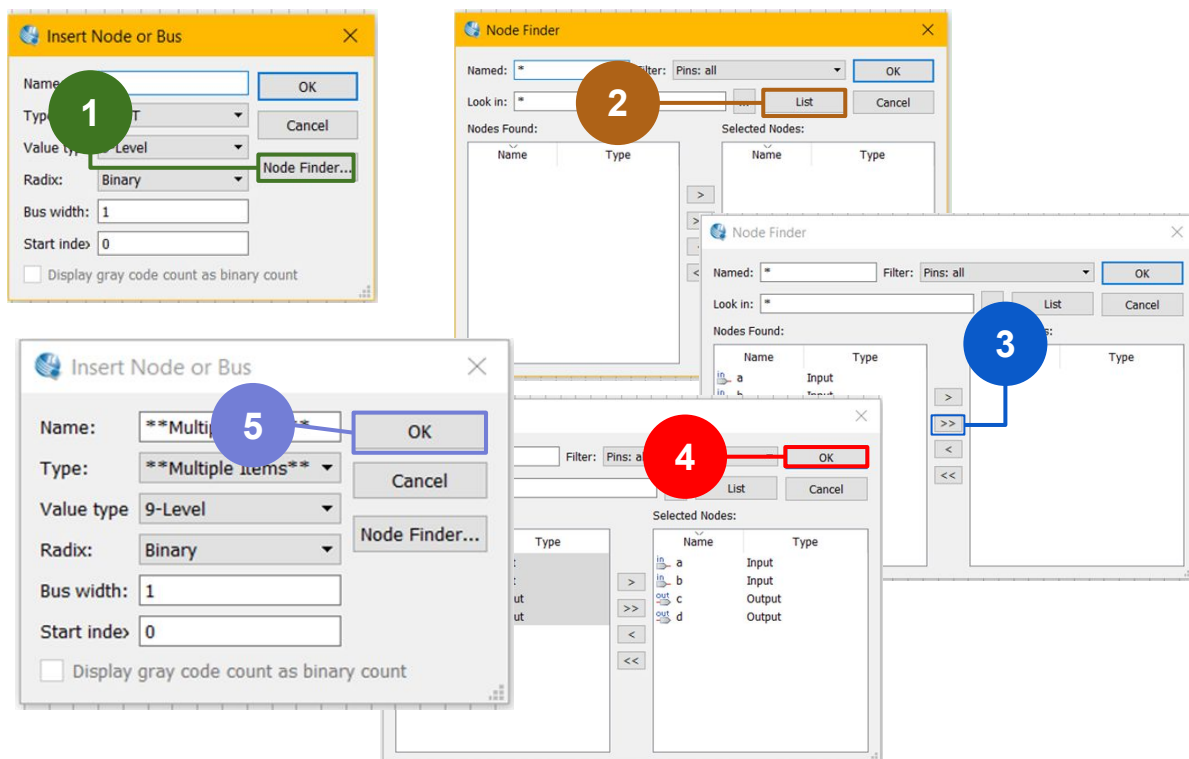


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Simulação do código VHDL

Na aba Insert Node or Bus clique em **“Node Finder”**. Depois disso iremos adicionar todas as entradas e saídas, então clique em **“List”** e quando todas as informações forem adicionadas, clicar em **“>>”** e por fim **“OK”** e **“OK”**.





Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Simulação do código VHDL

No exemplo em questão temos as entradas “a” e “b”. Selecione cada entrada individualmente e clique em **“Overwrite Clock”** para definir as configurações das formas de onda. Neste exemplo usaremos pulsos de forma de onda quadrada com período de **20ns para “a”** e **40ns para “b”** e clique **“OK”**.

The screenshot displays the Quartus II simulation environment. At the top, the menu bar (File, Edit, View, Simulation, Help) and a toolbar with various simulation icons are visible. Below the toolbar, two waveform windows are shown side-by-side. The left window displays a square wave for signal 'a' with a period of 20.0 ns. The right window displays a square wave for signal 'b' with a period of 40.0 ns. Both windows have a 'Clock' dialog box open, which is used to configure the waveform parameters. The dialog box includes fields for 'Period', 'Offset', and 'Duty cycle (%)'. The 'OK' button in the dialog box is highlighted with a purple circle (6). Numbered callouts (1-6) indicate the sequence of actions: 1. Selecting signal 'a' in the list; 2. Clicking the 'Overwrite Clock' button in the toolbar; 3. Clicking the 'Clock' button in the dialog; 4. Selecting signal 'b' in the list; 5. Clicking the 'Clock' button in the dialog; 6. Clicking the 'OK' button in the dialog.



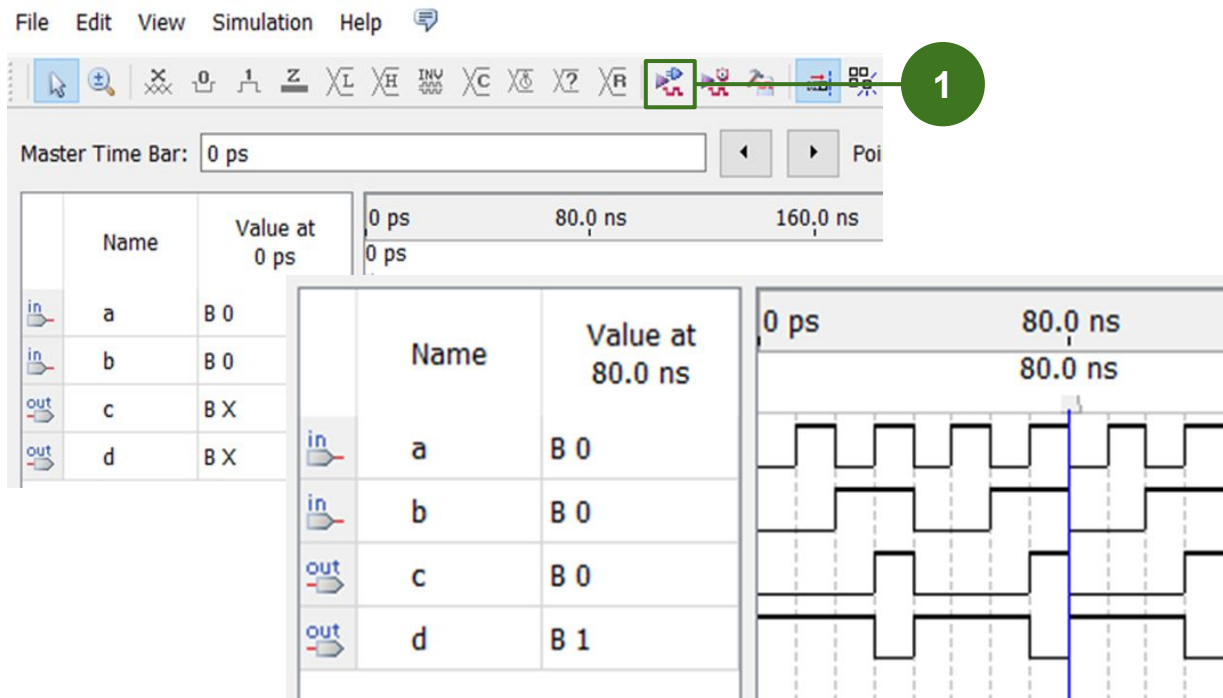


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Simulação do código VHDL

Terminado as configurações, clicar em **“Run Functional Simulation”** e aguarde o término da compilação e o resultado da simulação abrirá automaticamente. E assim conseguimos visualizar a simulação gráfica de uma porta AND.



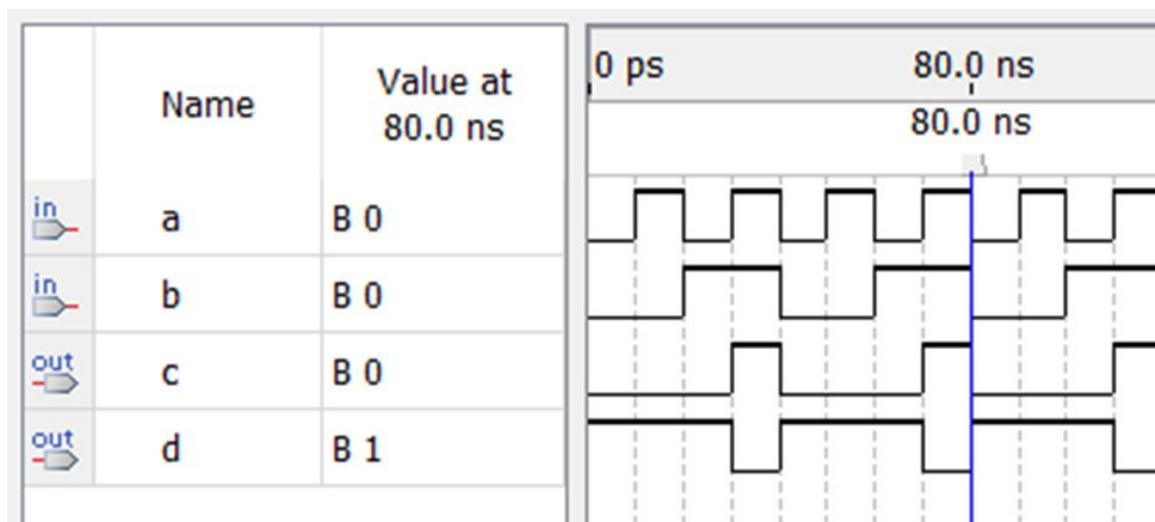


Tutorial de uso do FPGA DE2-115

Utilizando o Software Quartus II 13.1.0.162

Simulação do código VHDL

E então conseguimos verificar a simulação gráfica de uma porta AND. Através da simulação agora podemos fazer as análises de funcionamento.



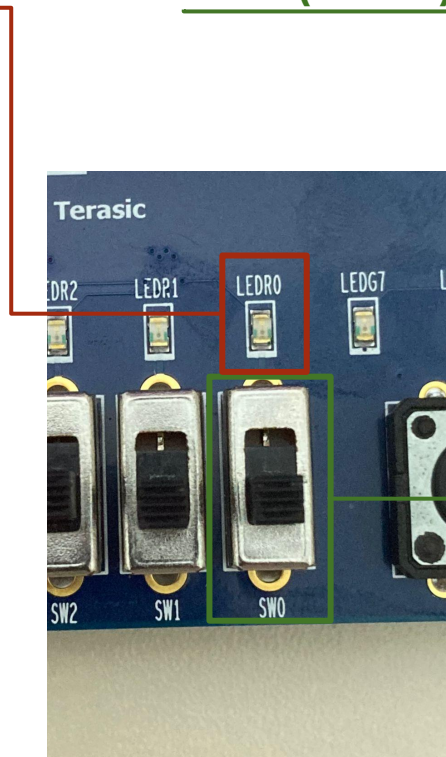


Tutorial de uso do FPGA DE2-115

Experimento 1

Pisca LED ativado por chave

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





Tutorial de uso do FPGA DE2-115

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. Esse é o código que iremos utilizar neste projeto:

```
library
use ieee.std_logic_1164.all;
```

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

```
entity pisca_led is
    port(sys_clk_50mhz: in std_logic;
          sys_rst: in std_logic;
          led: out std_logic);
end entity pisca_led;
```

Definição das portas de entrada e saída do código.

```
architecture behavior of pisca_led is
    signal led_delay: NATURAL range 0 to 50000000;
    signal led_reg: std_logic;
begin
    led <= led_reg;
    process(sys_clk_50mhz, sys_rst)
    begin
        if sys_rst = '0' then
            led_delay <= 0;
            led_reg <= '0';
        elsif rising_edge(sys_clk_50mhz) then
            led_delay <= led_delay + 1;
            if led_delay = 50000000 then
                led_delay <= 0;
                led_reg <= not led_reg;
            end if;
        end if;
    end process;
end architecture behavior;
```

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

Node Name	Direction	Location
out led	Output	PIN_G19
in sys_clk_50mhz	Input	PIN_Y2
in sys_rst	Input	PIN_AB28





Tutorial de uso do FPGA DE2-115

Experimento 1

Pisca LED ativado por chave

Este é o código completo em VHDL e os pinos utilizados da FPGA:

```
library ieee;
use ieee.std_logic_1164.all;
entity pisca_led is
    port(sys_clk_50mhz: in std_logic;
          sys_rst: in std_logic;
          led: out std_logic);
end entity pisca_led;
architecture behavior of pisca_led is
    signal led_delay: NATURAL range 0 to 50000000;
    signal led_reg: std_logic;
begin
    led <= led_reg;
    process(sys_clk_50mhz, sys_rst)
    begin
        if sys_rst = '0' then
            led_delay <= 0;
            led_reg <= '0';
        elsif rising_edge(sys_clk_50mhz) then
            led_delay <= led_delay + 1;
            if led_delay = 50000000 then
                led_delay <= 0;
                led_reg <= not led_reg;
            end if;
        end if;
    end process;
end architecture behavior;
```

	Node Name	Direction	Location
out	led	Output	PIN_G19
in	sys_clk_50mhz	Input	PIN_Y2
in	sys_rst	Input	PIN_AB28





Tutorial de uso do FPGA DE2-115

Experimento 1

Pisca LED ativado por chave

A chave (SW0) em **nível lógico alto (1)** fará **piscar o led(LED0)** com um **clock interno de frequência em 50 MHz**.

```
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
```



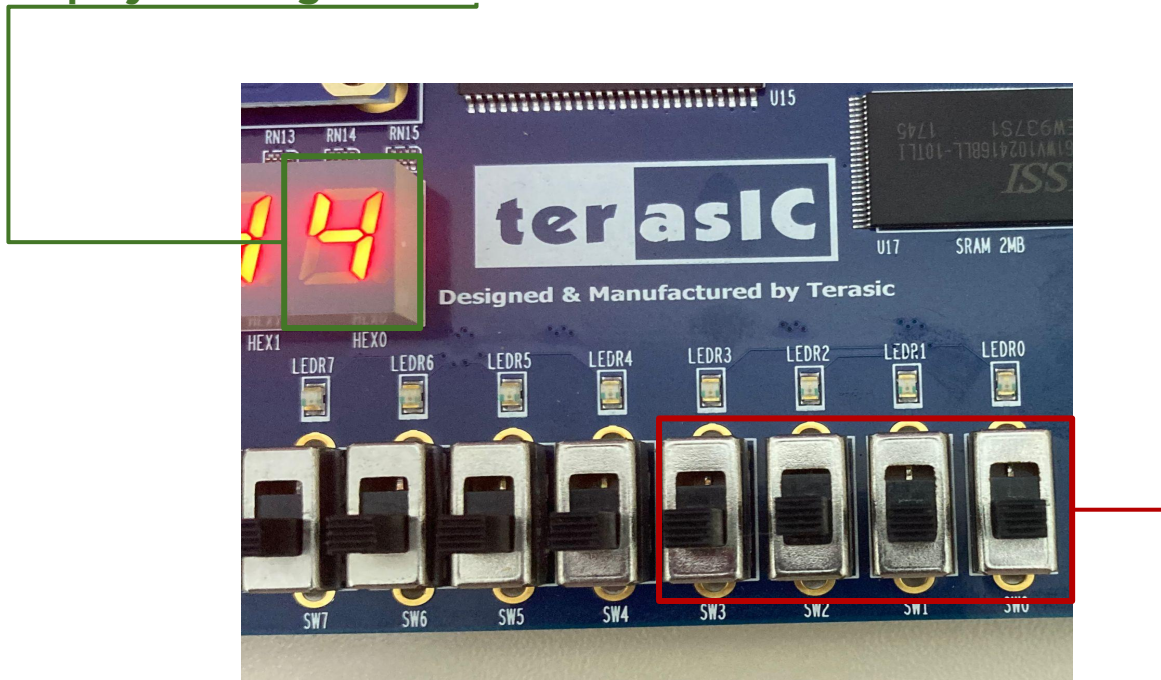


Tutorial de uso do FPGA DE2-115

Experimento 2

Decodificador BCD para 7 Segmentos

O experimento não precisará de nenhum componente além da plataforma FPGA e do computador. O experimento consiste implementar valores de 1 a 9 em BCD via de **chaves (Switch)** no **display de 7 segmentos**.





Tutorial de uso do FPGA DE2-115

Experimento 2

Decodificador BCD para 7 Segmentos

Este é o código completo em VHDL e os pinos utilizados da FPGA:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity DEC_BCD_7SEG is
    port(BCD          :in UNSIGNED(3 downto 0);
          SEGMENT7   :out UNSIGNED(6 downto 0)
    );
end entity;
architecture behavior of DEC_BCD_7SEG is
    signal SEGMENT:UNSIGNED(6 downto 0);
begin
    SEGMENT <= "0111111" when BCD = "0000" else -- 0
              "0000110" when BCD = "0001" else -- 1
              "1011011" when BCD = "0010" else -- 2
              "1001111" when BCD = "0011" else -- 3
              "1100110" when BCD = "0100" else -- 4
              "1101101" when BCD = "0101" else -- 5
              "1111101" when BCD = "0110" else -- 6
              "0000111" when BCD = "0111" else -- 7
              "1111111" when BCD = "1000" else -- 8
              "1100111" when BCD = "1001" else -- 9
              "1110111" when BCD = "1010" else -- A
              "1111100" when BCD = "1011" else -- B
              "0111001" when BCD = "1100" else -- C
              "1011110" when BCD = "1101" else -- D
              "1111001" when BCD = "1110" else -- E
              "1110001" when BCD = "1111"; -- F
    SEGMENT7 <= not SEGMENT;
end behavior;
```

Node Name	Direction	Location
BCD[3]	Input	PIN_AD27
BCD[2]	Input	PIN_AC27
BCD[1]	Input	PIN_AC28
BCD[0]	Input	PIN_AB28
SEGMENT7[6]	Output	PIN_H22
SEGMENT7[5]	Output	PIN_J22
SEGMENT7[4]	Output	PIN_L25
SEGMENT7[3]	Output	PIN_L26
SEGMENT7[2]	Output	PIN_E17
SEGMENT7[1]	Output	PIN_F22
SEGMENT7[0]	Output	PIN_G18



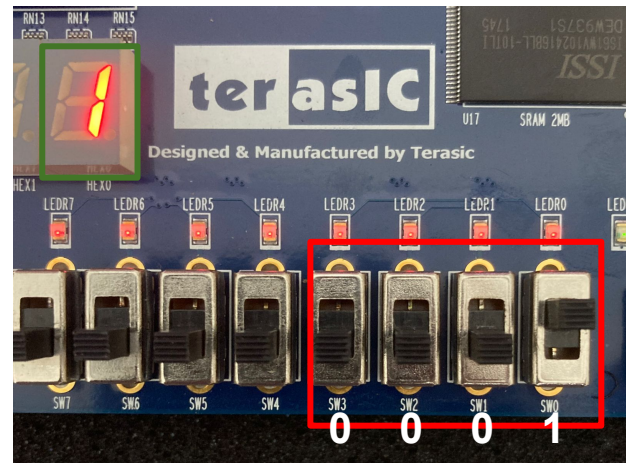
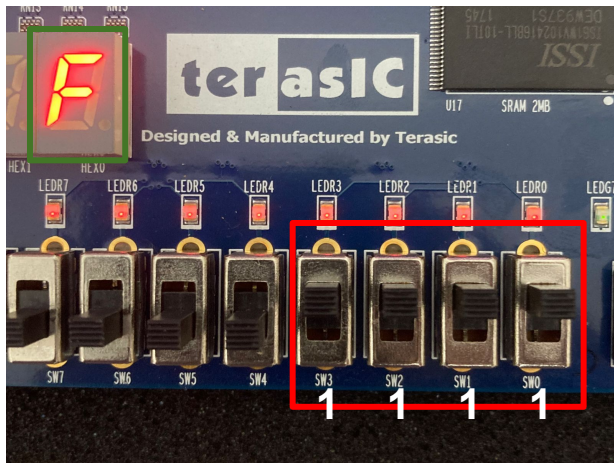


Tutorial de uso do FPGA DE2-115

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 DE2-115

Experimento 3

Contador de 4 dígitos

O experimento não precisará de nenhum componente além da plataforma FPGA e do computador. O experimento consiste em, incrementar via pulso de clock acionado por **botão(KEY3)**, um contador de 4 dígitos no **display de 7 segmentos(HEXO)**.





Tutorial de uso do FPGA DE2-115

Experimento 3

Contador de 4 dígitos

Este é o código completo em VHDL:

```
contador.vhd X
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
in CLK	Input	PIN_A7
in CLR	Input	PIN_C11
in EN	Input	PIN_C10
in LD	Input	PIN_D12
in LOAD[3]	Input	PIN_A14
in LOAD[2]	Input	PIN_A13
in LOAD[1]	Input	PIN_B12
in LOAD[0]	Input	PIN_A12
out Q[6]	Output	PIN_C17
out Q[5]	Output	PIN_D17
out Q[4]	Output	PIN_E16
out Q[3]	Output	PIN_C16
out Q[2]	Output	PIN_C15
out Q[1]	Output	PIN_E15
out Q[0]	Output	PIN_C14
in RST	Input	PIN_C12

```
36  --Q <= CONT;
37  Q1 <= "0111111" when CONT = "0000" else -- 0
38  "0000110" when CONT = "0001" else -- 1
39  "1011011" when CONT = "0010" else -- 2
40  "1001111" when CONT = "0011" else -- 3
41  "1100110" when CONT = "0100" else -- 4
42  "1101101" when CONT = "0101" else -- 5
43  "1111101" when CONT = "0110" else -- 6
44  "0000111" when CONT = "0111" else -- 7
45  "1111111" when CONT = "1000" else -- 8
46  "1100111" when CONT = "1001" else -- 9
47  "1110111" when CONT = "1010" else -- A
48  "1111100" when CONT = "1011" else -- B
49  "0111001" when CONT = "1100" else -- C
50  "1011110" when CONT = "1101" else -- D
51  "1111001" when CONT = "1110" else -- E
52  "1110001" when CONT = "1111"; -- F
53  Q <= not Q1;
54
55
56  End architecture;
```



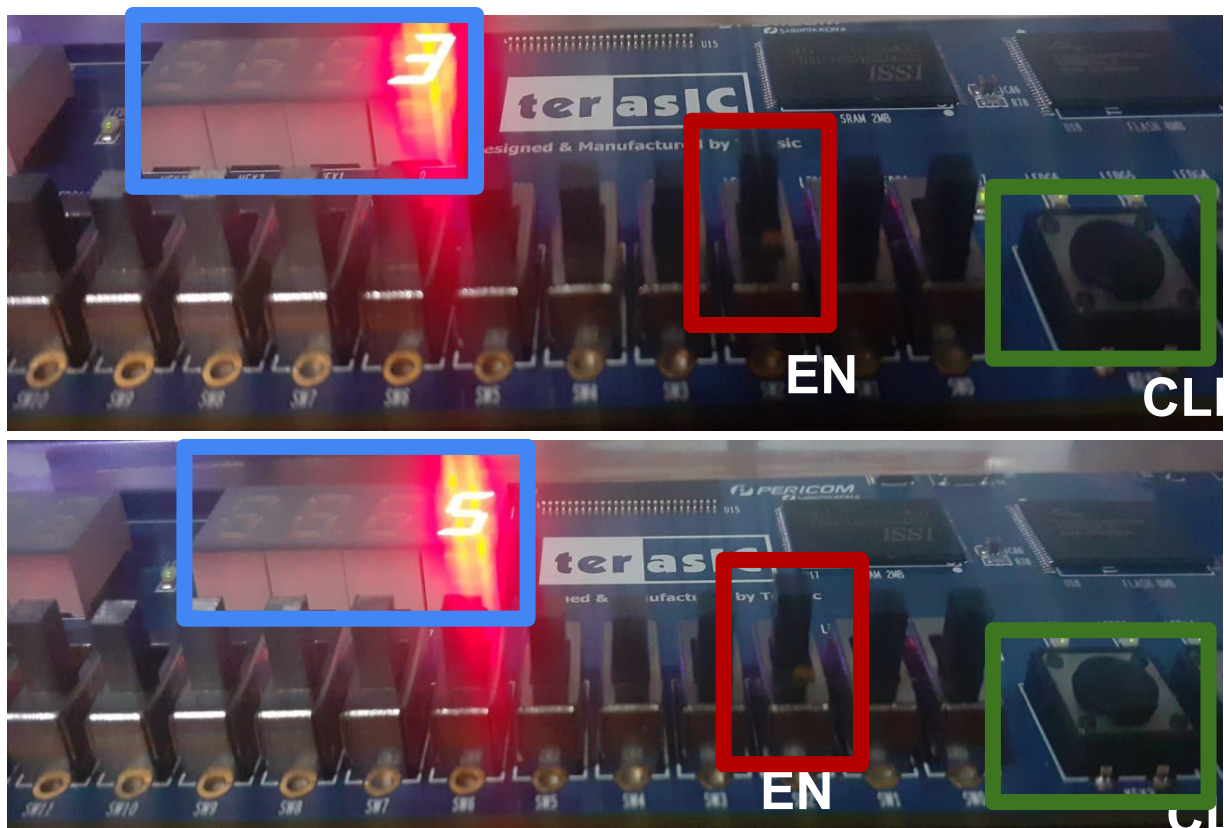


Tutorial de uso do FPGA DE2-115

Experimento 3

Contador de 4 dígitos

Conforme código, a **chave enable (EN)**, 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**.





Tutorial de uso do FPGA DE2-115

Agradecimento

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

