

---

UNIVERSIDADE FEDERAL FLUMINENSE – UFF  
ESCOLA DE ENGENHARIA – TCE  
CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES – TGT  
PROGRAMA DE EDUCAÇÃO TUTORIAL – PET  
GRUPO PET-TELE

## Tutorial PET-Tele

# Configurando o *kit* FPGA DE10-Lite

(Versão: A2020M08D26)

Autores: Gabriel Bueno  
João Pedro Mateus  
Lucca Sabbatini  
Vinícius Corrêa Figueira  
Cleyton da Cunha Gomes

Tutor: Alexandre Santos de la Vega

Niterói – RJ  
Agosto / 2020

---

# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Motivação . . . . .	2
1.2	Objetivo . . . . .	2
<b>2</b>	<b>Sobre os FPGAs</b>	<b>2</b>
2.1	O que é um FPGA? . . . . .	2
2.2	O <i>kit</i> DE10-Lite . . . . .	3
2.3	Requisitos para utilização . . . . .	3
<b>3</b>	<b>Configurando o <i>kit</i> pelo sistema operacional Windows</b>	<b>5</b>
3.1	Instalando o Quartus II . . . . .	5
3.2	Criando projetos . . . . .	8
3.3	Passando projetos para o FPGA . . . . .	10
<b>4</b>	<b>Configurando o <i>kit</i> pelo sistema operacional Linux</b>	<b>12</b>
4.1	Instalando o Quartus II . . . . .	12
4.2	Criando projetos . . . . .	15
4.3	Passando projetos para o FPGA . . . . .	17
<b>5</b>	<b>Exemplo de implementação: MUX 2x1</b>	<b>19</b>
	<b>Referências</b>	<b>22</b>

# 1 Introdução

O Programa de Educação Tutorial (PET) [Pro], do Ministério da Educação (MEC), exige que os grupos PET desenvolvam atividades que contemplem, de forma indissociável, itens de Pesquisa, de Ensino e de Extensão. Além disso, os grupos devem estimular uma evolução positiva dos seus integrantes, dos demais alunos do seu curso de graduação, do próprio curso e da sua instituição.

Nesse sentido, o PET-Tele [PET] procura desenvolver atividades e/ou atender a demandas que cumpram tais exigências.

A seguir, são apresentadas as motivações e o objetivo para o trabalho em questão.

## 1.1 Motivação

Recentemente, o PET-Tele adquiriu um *kit* FPGA DE10-Lite de desenvolvimento. Como nenhum dos integrantes possuía experiências anteriores com o *kit*, o tutor do grupo propôs a elaboração de um documento que serviria não só como guia para qualquer pessoa que queira interagir com o FPGA, mas também como uma oportunidade para os próprios integrantes se familiarizarem com o ele.

## 1.2 Objetivo

Este documento tem como objetivos:

- Servir como oportunidade de estudo sobre como manipular o *kit* para os integrantes do grupo, para que seja possível montar experimentos no futuro;
- Preparar o leitor com o básico para fazer o *kit* funcionar.
- Sedimentar e propagar conhecimento sobre o assunto;

No fim da leitura, é esperado que o leitor seja capaz de implementar qualquer tipo de circuito ou algoritmo que desejar apenas seguindo a bibliografia e os passos apresentados aqui.

# 2 Sobre os FPGAs

## 2.1 O que é um FPGA?

FPGA é um acrônimo para *Field Programmable Gate Array*, ou Arranjo de Portas Programáveis em Campo. Recebe esse nome por consistir em um conjunto, configurável pelo usuário após a fabricação, de circuitos integrados. É composto por blocos lógicos, entradas e saídas. Os blocos lógicos podem ser configurados para representar diversos elementos de circuitos digitais, desde circuitos complexos até uma única porta lógica. Seu propósito é permitir a flexibilização em projetos digitais para que o usuário possa configurá-la da maneira que desejar, proporcionando diversas aplicações para o mesmo. É amplamente utilizado no âmbito acadêmico para estudo e pesquisa, como também por projetistas no desenvolvimento de *chips* para o mercado. Pode ser útil em diversas áreas, seja na medicina ou até no auxílio de pilotagem aérea ou automobilística, por exemplo.

## 2.2 O *kit* DE10-Lite

O *kit* DE10-Lite é uma placa Altera MAX 10 baseada em FPGA, que possibilita a utilização da capacidade máxima do FPGA MAX 10. A placa conta com cerca de cinquenta mil elementos lógicos (LEs) e conversor analógico-digital (ADC) integrado.

O *kit* permite a prototipagem de aplicações industriais, automotivas, de consumo, entre outras. Além disso, os recursos do dispositivo e o seu *design* permitem também a implementação ampla de circuitos mais simples, possibilitando facilmente o desenvolvimento de projetos por usuários iniciantes. O desenvolvimento dos projetos no *kit* DE10-Lite, por sua vez, é realizado exclusivamente no *software* Quartus Prime Lite Edition.

Alguns dos componentes da placa são:

- 1.638 Kb de memória, em blocos de memória M9K.
- Conversor analógico-digital duplo.
- 144 multiplicadores de 18x18 *bits*.
- 4 PLLs para geração de *clocks*.
- Um conector GPIO de 2x20 pinos (nível de tensão: 3.3V).
- Conector compatível com Arduino Uno R3, incluindo 6 canais analógicos (ADC).
- Acelerômetro integrado.
- 10 LEDs.
- 10 chaves seletoras.
- 2 botões de pressão.
- 6 dígitos numéricos (7 segmentos).
- Saída VGA.
- Alimentação através do cabo USB.

**Obs.:** a ficha completa de especificações técnicas pode ser encontrada através do seguinte [URL](https://www.macnicadhw.com.br/produtos/kits/altera-de10-lite):

<https://www.macnicadhw.com.br/produtos/kits/altera-de10-lite>

## 2.3 Requisitos para utilização

Para interagir com o *kit* DE10-Lite é necessário antes o usuário instalar em sua máquina o arquivo “DE10-Lite CD-ROM” e o programa Quartus II. O primeiro consiste em um material de suporte elaborado pela própria fornecedora da placa (Terasic), em que nele pode ser acessado manuais contendo informações técnicas sobre a placa e suas componentes, além de possuir o “*System Builder*” utilizado na criação de projetos e que será explorado posteriormente, esse é apenas útil no sistema operacional Windows. O Quartus II consiste de um programa de *design* de dispositivos lógico programáveis (*Programmable Logic Devices - PLD*) para elaboração de circuitos lógicos em blocos, ou por *scripts* de Linguagens de Descrição de *Hardware* (*Hardware Description Languages - HDL*), tendo suporte para VHDL, AHDL, Verilog e SystemVerilog, podendo, portanto, passar projetos para circuitos integrados, no caso o FPGA.

Esse manual não tem como intuito explicar com muita profundidade todas os componentes disponibilizados pelo Quartus II e pelo *kit* DE10-Lite, cabendo ao leitor tomar alguns estudos próprios da ferramenta utilizada. Com relação ao estudo, a Intel disponibiliza um livro introdutório gratuito para a familiarização com FPGAs através do seguinte [URL](#):

`https://plan.seek.intel.com/PSG_WW_NC_LPCD_FR_2018_FPGAforDummiesbook`

Por outro lado, será mostrado aqui o procedimento de instalação do programa, de modo que esse tenha suporte para o *kit* tratado.

Vale ressaltar que, para realizar o *download* do Quartus II, o usuário precisará criar uma conta na Intel. Para isso, basta selecionar a opção “*Sign up here*” no seguinte [URL](#):

`https://www.intel.com/content/www/us/en/my-intel/fpga-sign-in.html?redirect=/content/www/us/en/secure/my-intel/dashboard.html`

Para acessar a página de cadastro, clique em “*Sign up here*”, como mostra a Figura 1. Depois, preencha os campos com suas informações, como mostra a Figura 2.

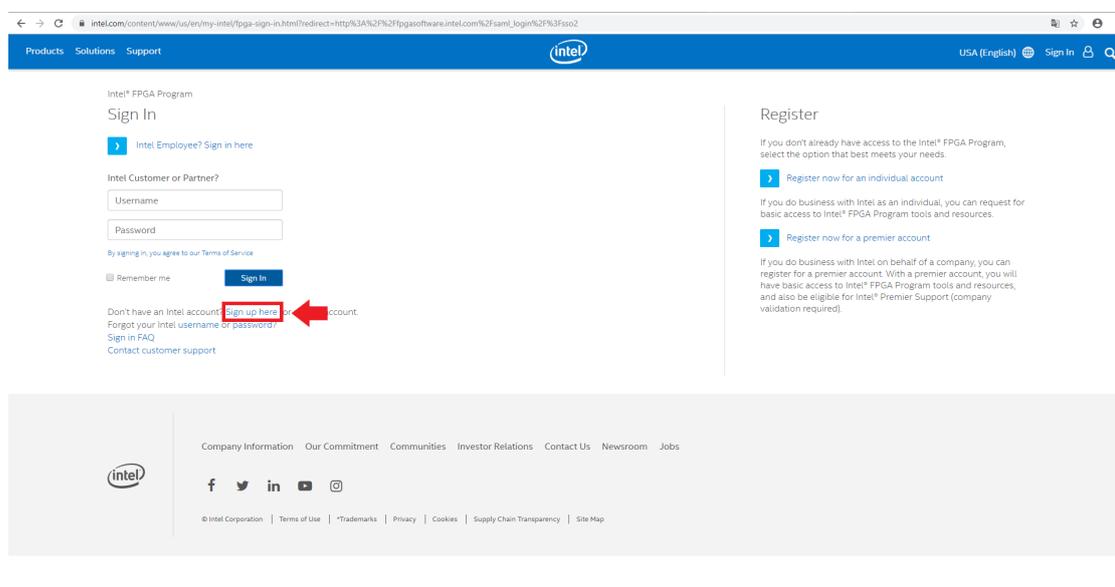


Figura 1: Página de registro da conta Intel.

Register for Basic Intel® Account  
Sign up for a Basic Intel® Account to be able to login to My Intel and start your journey with Intel. You will still need to register for specific programs in order to gain additional privileges.

Personal Information

First Name Last Name  
Business Email Address Username  
Password Confirm Password  
Country/Region  
Job Function  
Country/Region Code Phone Extension (optional)  
Next Step

Terms and Conditions

Figura 2: Página de cadastro dos dados pessoais.

### 3 Configurando o *kit* pelo sistema operacional Windows

A seguir, encontra-se um passo a passo de como instalar o Quartus II, onde encontrar e como utilizar os arquivos do “DE10-Lite CD-ROM” no sistema operacional Windows.

Caso alguma dúvida referente à instalação do Quartus II permaneça após a leitura do tutorial, pode-se encontrar um guia de instalação rápida em inglês mais esmiuçado no seguinte [URL](#):

```
https://fpgasoftware.intel.com/static/quick_start_guide/quick_start_guide_19.1_en.pdf
```

#### 3.1 Instalando o Quartus II

Como requisito da programação na placa, deve-se instalar até no mínimo a versão 16.0 do Quartus II, do tipo *lite*. Os demais tipos disponíveis (“*Standard*” e “*Pro*”) requerem licença para uso. O programa se encontra disponível no seguinte [URL](#):

```
https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html
```

Para fazer o *download* da versão desejada do *software*, acesse o *URL* acima e selecione as seguintes opções:

1. “*Select by Device*”.
2. Em “*Devices*”, escolha a opção “MAX Series”.
3. Selecione a opção “MAX 10”.
4. Por fim, escolha a versão que se deseja instalar das disponíveis na aba “*Lite*” (lembrando que deverá ser no mínimo a versão 16.0).

A Figura 3 a seguir demonstra o passo a passo das opções que devem ser selecionadas.

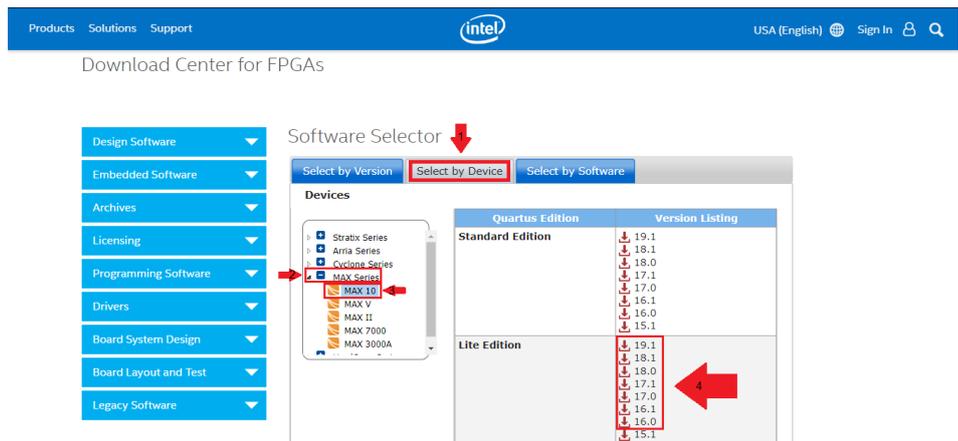


Figura 3: Passo a passo da seleção de opções para a escolha da versão do Quartus II.

Ao ser redirecionado para a próxima página, selecione a versão compatível com Windows, como mostra a Figura 4.

Confira se é a versão *lite* do *software* que está selecionada, logo acima do seletor de versão. Para baixar o programa, selecione a aba “*Individual Files*” na mesma página.

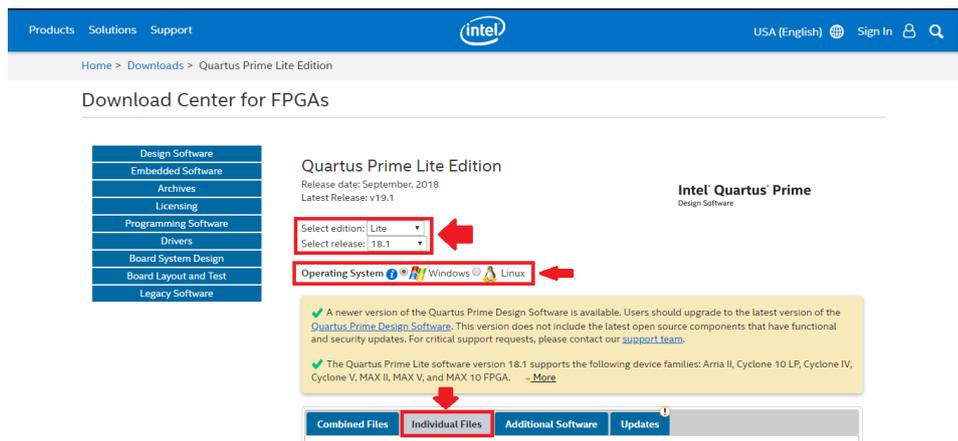


Figura 4: Destaque dos seletores e opções de *download* da página.

**Obs.:** pode ser que a versão selecionada, por ser ainda muito recente, tenha compatibilidade somente com o sistema Linux (uma mensagem como a da Figura 5 a seguir será exibida logo abaixo do seletor de sistema operacional).

Caso isso aconteça, selecione versões anteriores no seletor “*Select release*”, também evidenciado na Figura 4, até que uma versão esteja disponível para Windows.

## Quartus Prime Lite Edition

Release date: September, 2019  
Latest Release: v19.1

Intel® Quartus® Prime  
Design Software

Select edition: Lite  
Select release: 19.1

Operating System  Windows  Linux

\*\* The Quartus Prime Lite Edition version 19.1 for Windows is not yet available. The Windows version will be released in the near future. Check back here for updates.

Figura 5: A versão 19.1, por exemplo, só está disponível para Linux na data de confecção deste manual.

Junto com a instalação do programa é necessário fazer o *download* de um arquivo de suporte para a família “MAX 10” correspondente ao *kit* usado. Ambos procedimentos são apresentados na Figura 6 e Figura 7.

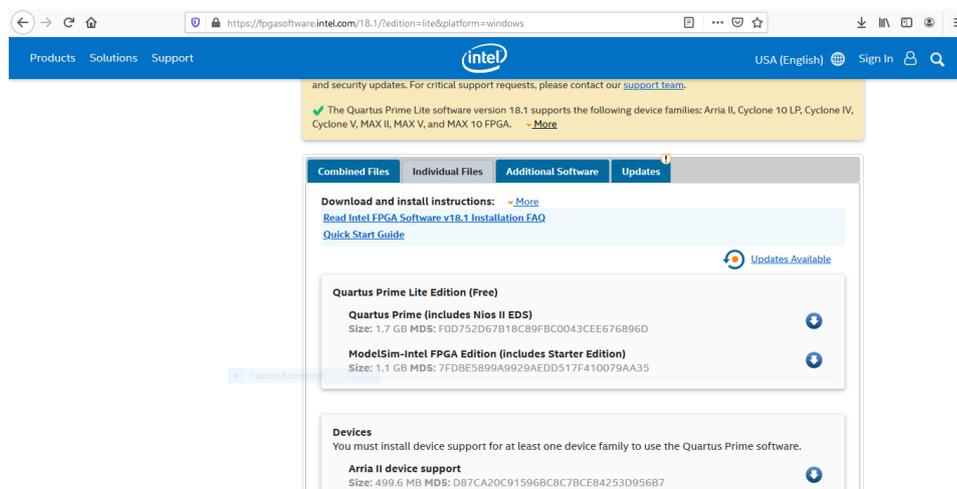


Figura 6: Página para a instalação do Quartus II, a primeira opção consiste no Quartus Prime Lite Edition o programa de interesse do tutorial. A segunda é o ModelSim, de simulação de FPGAs, sendo opcional.

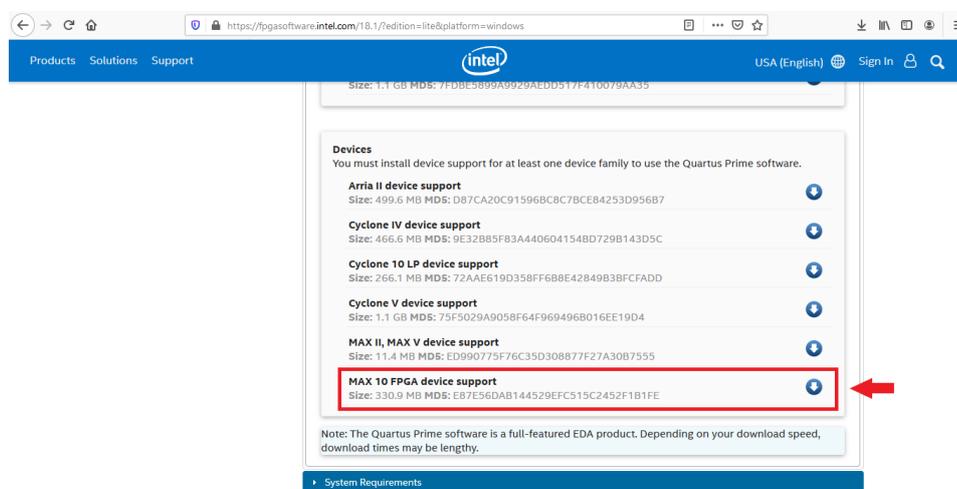


Figura 7: Arquivos de suporte para diferentes tipos de FPGA, escolha o suporte para a família “MAX 10” destacado em vermelho.

Após o *download*, deve-se iniciar o instalador e seguir as instruções. Ao longo do procedimento será perguntado se é necessário algum suporte para alguma família de placas, onde deve-se apenas indicar o arquivo correspondente ao “MAX 10 FPGA Device Support”. Também é possível adicionar o suporte para famílias de FPGA após a instalação do Quartus II, bastando procurar nos programas por “*Device Installer*” e executar.

### 3.2 Criando projetos

Uma vez concluída a instalação do Quartus II, será criado um projeto para interagir com a placa.

Para isso, deve-se antes realizar o *download* do arquivo explicitado anteriormente, o “DE10 System CD”, que se encontra disponível no seguinte *URL*:

<https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1021&PartNo=4>

A página aberta é exibida na Figura 8. Deve-se ir, então, para a seção indicada por “CD-ROM” e escolher a opção “DE10-Lite CD-ROM”.

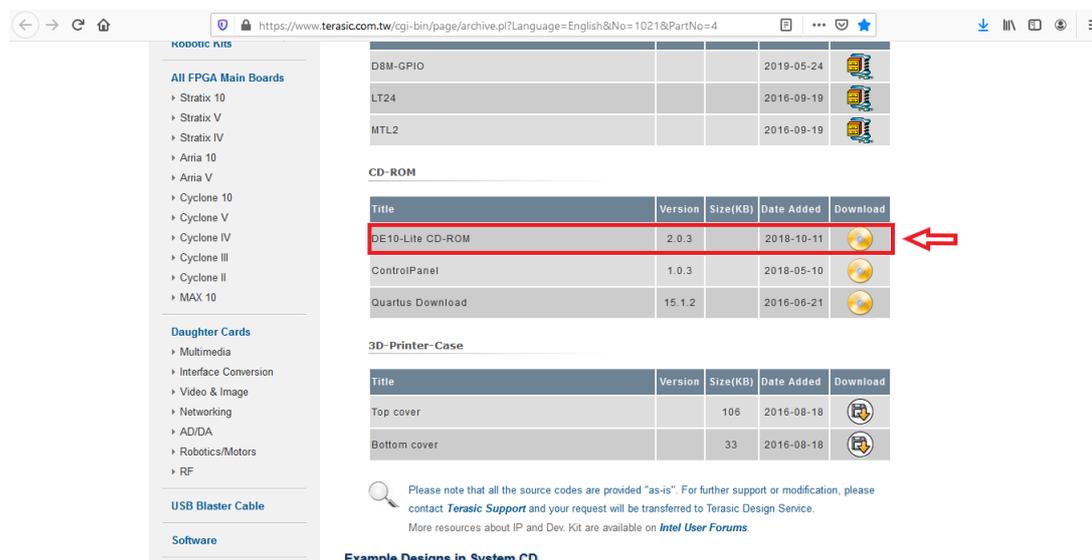


Figura 8: Selecione e faça o *download* do arquivo destacado em vermelho.

Será aberta uma outra página com algumas opções de *download*. Selecione a opção cujo nome é “DE10-Lite\_v.2.0.3\_SystemCD.zip”. Esse é um arquivo muito importante para o tutorial, pois o mesmo possui uma ferramenta que gera automaticamente o projeto a ser utilizado especificamente para o *kit*.

Após realizar o *download*, deve-se descompactar o arquivo, extrair o conteúdo e ir para o seguinte endereço:

“ ~ \DE10-Lite\_v.2.0.3\_SystemCD\Tools\SystemBuilder”

Execute o arquivo “DE10\_Lite\_SystemBuilder”. Será aberta a janela mostrada na Figura 9.

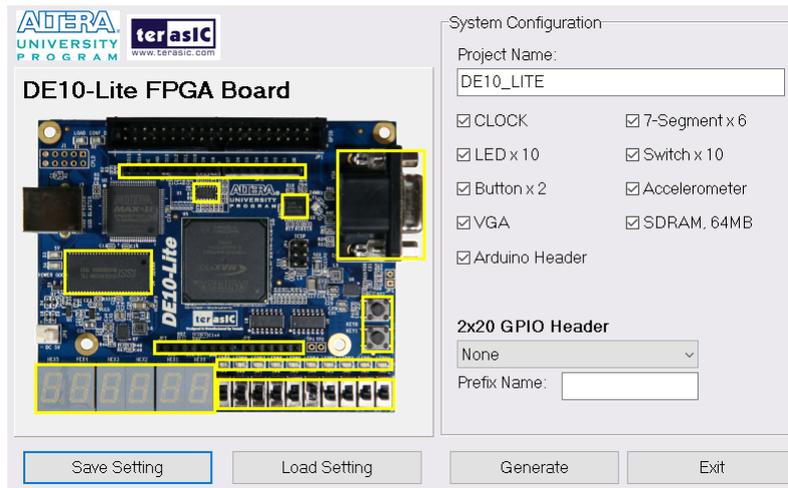


Figura 9: Especificação da placa. Selecione as componentes que serão utilizadas no projeto.

A janela aberta exibe todos os componentes do *kit*, cabendo ao usuário escolher quais serão utilizados. Ao selecionar a opção “*Generate*”, um projeto correspondente será criado contendo todas as declarações para o uso dos componentes especificados. Feito isso, o arquivo principal será do tipo “Quartus II Project”, bastando apenas selecioná-lo e abri-lo no programa. Após esperar alguns instantes para a execução do Quartus II, será exibida a janela ilustrada na Figura 10. Para acessar o *script* criado na linguagem Verilog, basta verificar no canto superior esquerdo da janela uma parte com o nome “*Project Navigator*”. Logo abaixo, tem-se o arquivo criado com o nome dado pelo usuário, nesse caso “DE10-LITE”, destacado em vermelho na Figura 10.

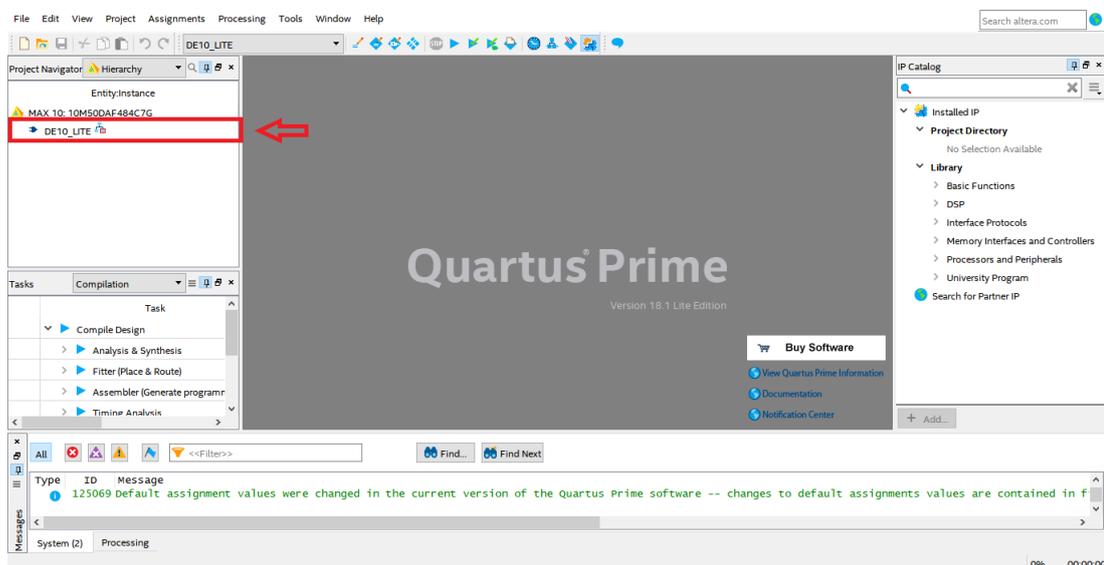


Figura 10: Selecione o projeto criado na parte compreendida por “*Project Navigator*”, destacado em vermelho, nesse caso com o nome “DE10-LITE”.

Feito isso, o *script* na linguagem de descrição de *hardware* Verilog será exibido na tela, pronto para edição. Nele, pode ser verificado que todos os componentes especificados anteriormente para uso são declarados no bloco “*Module*”. Vale destacar que na criação do projeto tratado

neste exemplo foi apenas selecionada a opção para “LEDx10”.

Nesse momento do tutorial, será feita uma implementação simples no *script*, tendo apenas em vista mostrar como compilar no Quartus II e testar o código criado na placa.

No *script* pode ser observado que há comentários que mostram como ele é separado. Assim, será adicionada uma linha de código logo abaixo do comentário “// REG/WIRE declarations”, sendo:

```
assign LEDR[0] = 1;
```

Resta apenas compilar o código, bastando selecionar a opção “*Compile Design*”, logo abaixo de “*Project Navigator*”, sendo tal procedimento ilustrado pela Figura 11.

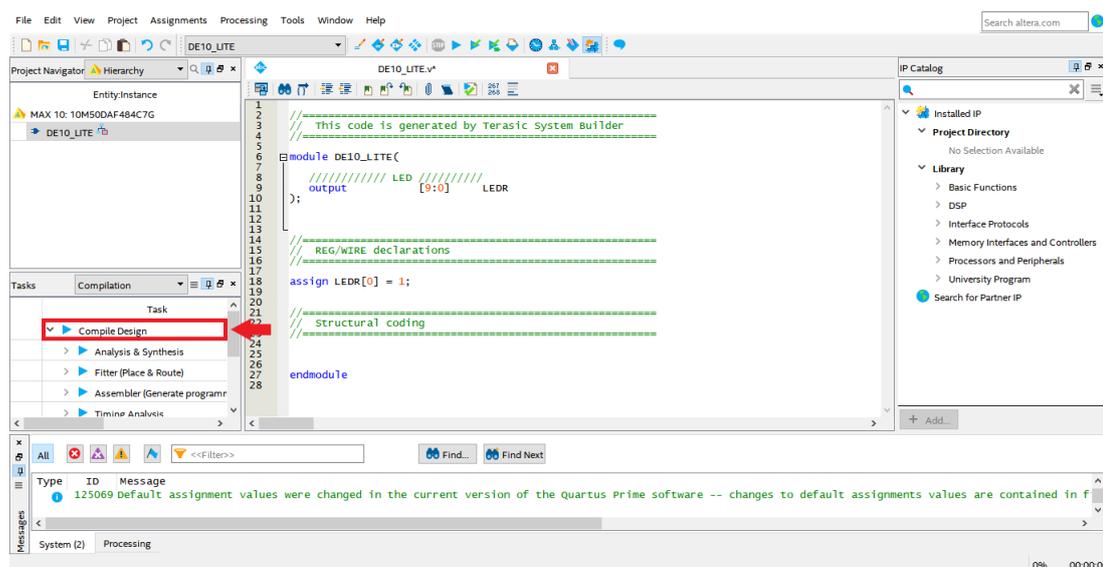


Figura 11: Código criado em Verilog, sendo destacado em vermelho a opção a ser selecionada para compilar.

A compilação demorará um tempo até ser concluída. Observa-se, na aba de compilação, que uma série de procedimentos são realizados. Logo após, será exibida na tela alguns dados interessantes do uso dos recursos da placa.

### 3.3 Passando projetos para o FPGA

Para que seja possível fazer o *upload* de projetos para o FPGA, deve-se antes instalar o *driver* para que a placa seja reconhecida pelo sistema. Para isso, tendo a mesma ligada a uma porta USB, vá para “Gerenciador de Dispositivos” e verifique em “Outros Dispositivos” por “USB-Blaster”, como ilustrado pela Figura 12.

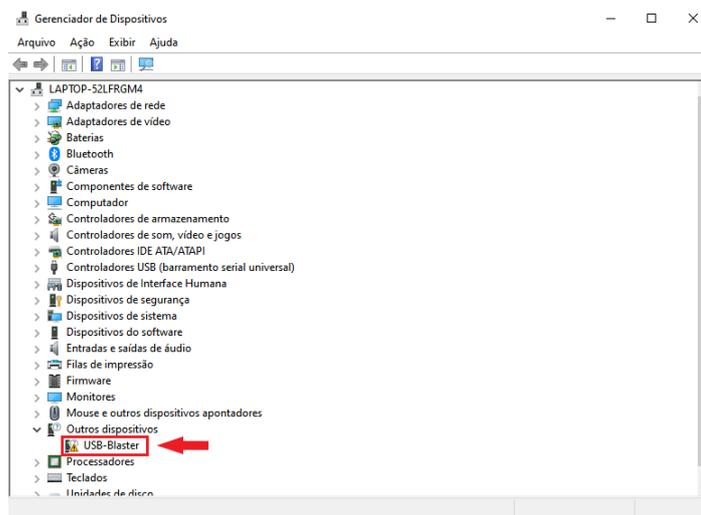


Figura 12: Selecione com o botão direito do *mouse* no dispositivo “USB-Blaster” e escolha a opção de “Propriedades”.

Selecione o dispositivo com o botão direito do *mouse*, escolha “Propriedades” e, logo após, a opção de “Atualizar *Driver...*”. Escolha a opção “Procurar *software de driver* no computador”, como mostra a Figura 13.

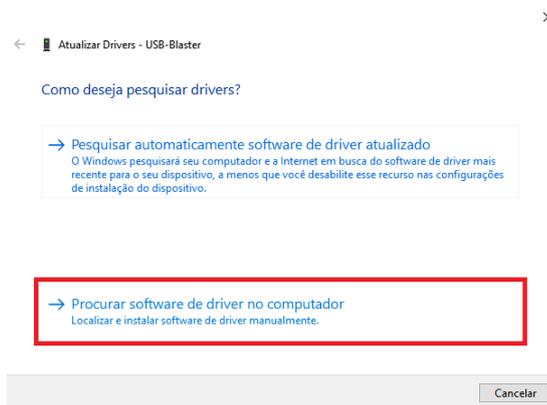


Figura 13: Escolha para “Procurar *software de driver* no computador” como em destaque.

Em seguida, escolha o caminho de diretório indicado pela Figura 14. Vale destacar que a versão tratada é a 18.1, sendo necessário indicar o caminho de acordo com a versão utilizada.

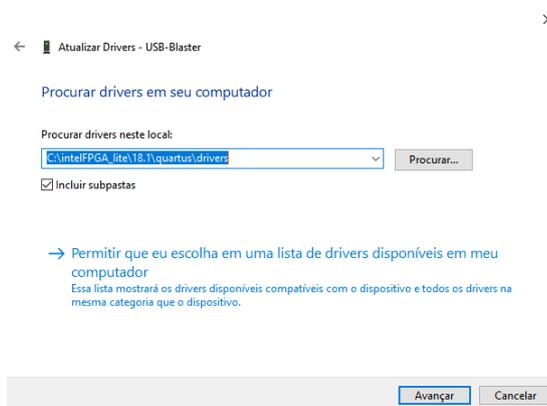


Figura 14: Procure pelo caminho indicado, correspondendo a versão utilizada do Quartus II.

Feito isso, o FPGA será reconhecida pelo sistema como Altera Blaster, estando pronta para ser configurada. Resta apenas, então, passar o *script* para a placa, sendo necessário selecionar a opção “*Program Device*” no final da aba “*Tasks*”. A seguir, será exibida a janela ilustrada na Figura 15. Inicialmente, selecione a opção “*Hardware Setup*” e o dispositivo com o nome USB-Blaster. Feito isso, feche a aba e volte para a aba de programação, onde o dispositivo terá sido reconhecido.

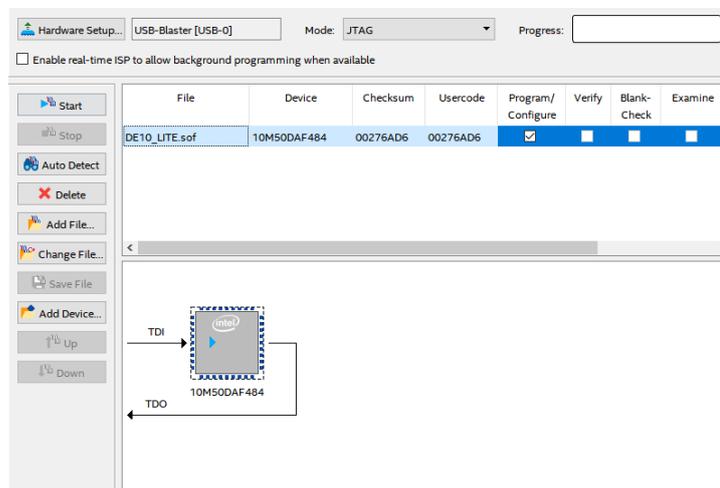


Figura 15: Assim que o dispositivo tenha sido identificado, selecione “*Start*” para que o FPGA seja configurado.

Selecionando a opção “*Start*”, o projeto finalmente será passado para a placa e o LED na posição 0 acenderá.

## 4 Configurando o *kit* pelo sistema operacional Linux

A seguir, encontra-se um passo a passo de como instalar o Quartus II no sistema operacional Linux.

No sistema operacional Linux, deve-se unicamente instalar o programa Quartus II, uma vez que o “DE10-Lite CD-ROM” possui suporte para criação de projetos apenas no sistema operacional Windows.

Caso alguma dúvida referente à instalação do Quartus II permaneça após a leitura do tutorial, pode-se encontrar um guia de instalação rápida em inglês mais esmiuçado no seguinte [URL](https://fpgasoftware.intel.com/static/quick_start_guide/quick_start_guide_19.1_en.pdf):

[https://fpgasoftware.intel.com/static/quick\\_start\\_guide/quick\\_start\\_guide\\_19.1\\_en.pdf](https://fpgasoftware.intel.com/static/quick_start_guide/quick_start_guide_19.1_en.pdf)

### 4.1 Instalando o Quartus II

Vale lembrar que a versão mínima do programa com suporte para a família “MAX 10” do *kit* é a 16.0.

Para fazer o *download*, primeiramente acesse o centro de *downloads* da Intel através do seguinte [URL](#):

<https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html>

Depois, escolha a versão requisitada atentando-se para selecionar uma versão do tipo *lite*. Feito isso, escolha a opção de sistema operacional Linux. Observe que haverá uma única opção de *download*, na qual estarão contidas diversos suportes para famílias de FPGA. O procedimento é apresentado na Figura 16.

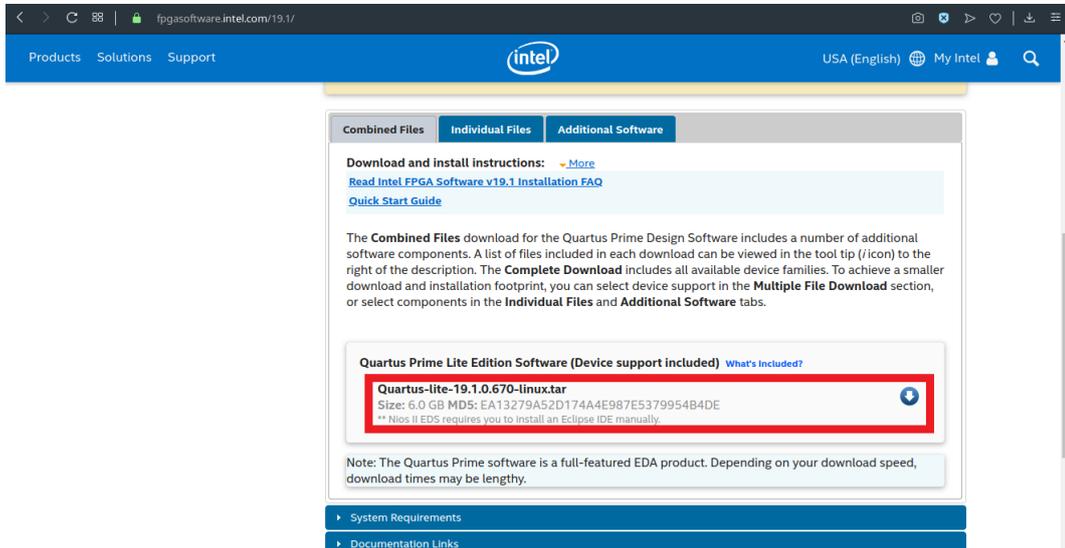


Figura 16: Faça o *download* do arquivo destacado em vermelho. Nesse caso, foi escolhida a versão 19.1 .

O arquivo compactado possui formato “.tar”. Tendo em vista o tamanho do arquivo, certifique-se da boa conexão de sua máquina, de forma a evitar interrupções ou problemas durante o *download*. Uma vez terminado o procedimento, abra o terminal e vá para a pasta em que o arquivo se encontra. Em seguida, use o comando

```
“$ tar -xvf Quartus-lite-*-linux.tar ”
```

para extrair o conteúdo do arquivo “.tar”, em que a parte preenchida com \* representa a versão que está sendo instalada. Tal procedimento é apresentado na Figura 17.

```
teste@teste-300E5M-300E5L:~$ cd Downloads/
teste@teste-300E5M-300E5L:~/Downloads$ tar -xvf Quartus-lite-19.1.0.670-linux.tar
components/ModelSimSetup-19.1.0.670-linux.run
components/QuartusHelpSetup-19.1.0.670-linux.run
components/QuartusLiteSetup-19.1.0.670-linux.run
components/arria_lite-19.1.0.670.qdz
components/cyclone-19.1.0.670.qdz
components/cyclone10lp-19.1.0.670.qdz
components/cyclonev-19.1.0.670.qdz
components/max-19.1.0.670.qdz
components/max10-19.1.0.670.qdz
setup.sh
readme.txt
teste@teste-300E5M-300E5L:~/Downloads$
```

Figura 17: Movendo-se para o diretório que contém o arquivo e descompactando-o

Após esse procedimento, notar-se-á que três arquivos serão obtidos, sendo uma pasta com o nome “*components*”, que possui os componentes necessários para a instalação e também os suportes para as diferentes famílias de FPGA, um arquivo de extensão “.txt” com o nome “*readme*” contendo informações técnicas e requisitos do programa, e o arquivo “*setup.sh*”, um *script* para *shell*, sendo esse o arquivo de interesse. Siga no terminal, na pasta atual, e digite o seguinte comando:

“`$chmod +x setup.sh`”.

Isso dará a permissão para o *script* ser executado pelo usuário. E assim deve ser feito, através do seguinte comando:

“`$ ./setup.sh`”.

Tais procedimentos são apresentados pela Figura 18

```
teste@teste-300E5M-300E5L:~/Downloads$ ls
components  Quartus-lite-19.1.0.670-linux.tar  readme.txt  setup.sh
teste@teste-300E5M-300E5L:~/Downloads$ chmod +x setup.sh
teste@teste-300E5M-300E5L:~/Downloads$ ./setup.sh
```

Figura 18: Dando as permissões para executar o instalador e executando-o.

Após essa etapa, o instalador do Quartus II será exibido. Em seguida, os passos apresentados, como escolher o diretório de instalação e os arquivos de suporte desejados (no caso o “MAX 10”), devem ser executados.

## 4.2 Criando projetos

Por não poder contar com o “DE10-Lite CD-ROM” para executar o criador de projetos no Linux, deve-se, portanto, criar o projeto manualmente pelo Quartus II, tendo em vista todas as especificidades da placa.

Ao abrir o Quartus II pela primeira vez na máquina, será exibida uma janela perguntando sobre o interesse do usuário em adquirir uma licença ou usar uma já adquirida. Nesse caso, selecione a opção “Run Quartus Prime Lite”.

Será exibida a janela de abertura do programa. Ao terminar de carregar, selecione a opção “New Project Wizard”. Após a abertura de uma aba de introdução, escolha o diretório do novo projeto e informe o seu nome. Escolhendo a opção “Next”, será aberta uma janela contendo duas formas de projeto. A primeira sendo “Empty project” (projeto vazio) e a segunda sendo “Project template”. Escolha a segunda. Dessa forma, deverá ser exibida uma aba como a Figura 19.

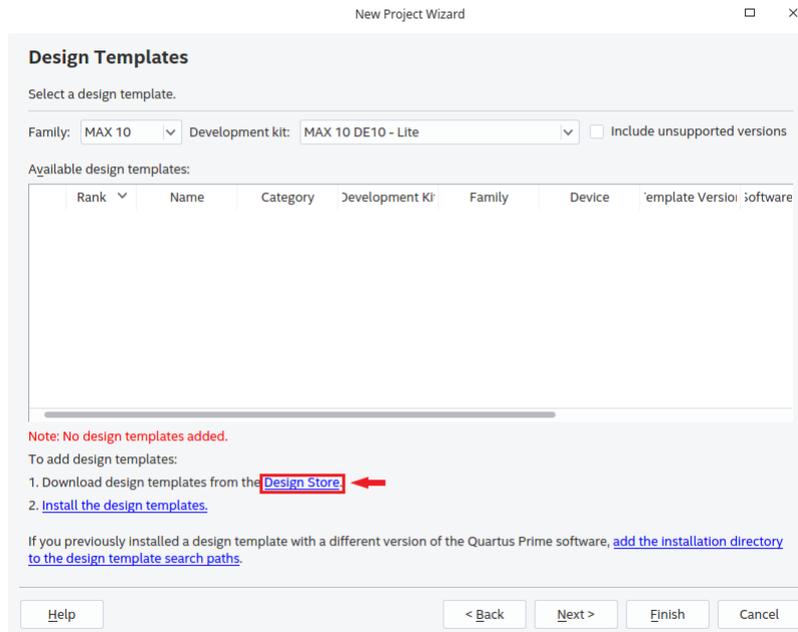


Figura 19: Escolha a opção “*Design Store*”.

Siga diretamente para “*Design Store*”, como em destaque na Figura 19. Será aberta uma *webpage*, sendo necessário informar, como família, a opção “MAX 10” e, como *kit* de desenvolvimento, a opção “MAX 10 DE10 - Lite”. Assim, deverá ser exibida uma lista de *templates* com tais especificações. Procure pela opção “*Baseline Pinout - MAX10 DE10 Lite*” na versão 16.1 e realize o *download*. Tal procedimento pode ser visualizado na Figura 20

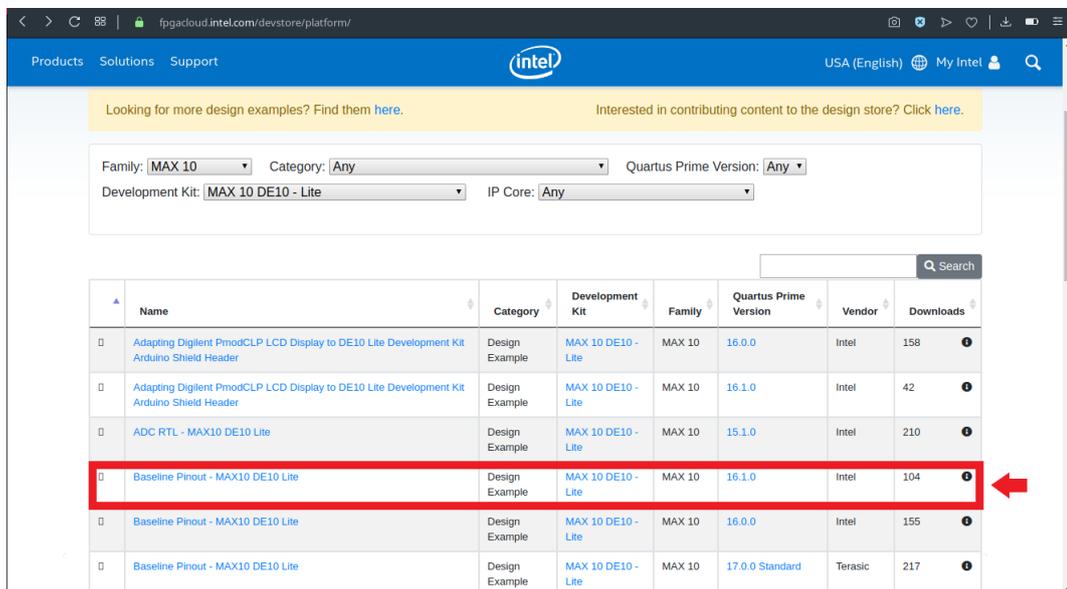


Figura 20: Selecione a opção em “*Baseline Pinout - MAX10 DE10 Lite*” na versão 16.1 .

Instale o *template* na mesma janela do Quartus II na opção “*Install the Design Templates*”. Feito isso, basta selecionar a opção instalada para iniciar o projeto, como mostra a Figura 21.

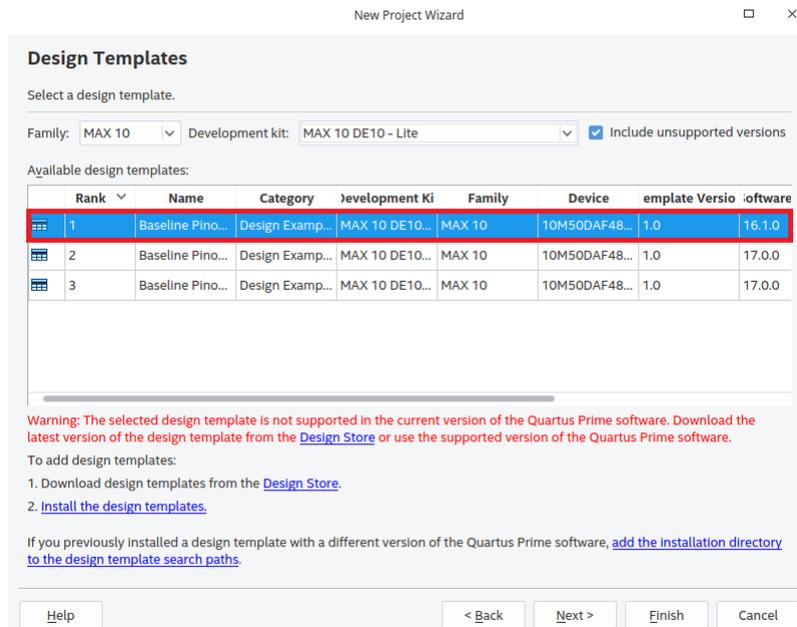


Figura 21: Escolha o *template* obtido, sendo este em destaque.

O projeto finalmente será exibido com todas as especificações da placa. Tal procedimento feito para criar projetos é o “padrão” para as placas de desenvolvimento. Porém, por ter disponível a ferramenta “System Builder” do “DE10-Lite CD-ROM” para Windows, torna-se uma alternativa muito mais prática para criação de projetos em tal sistema operacional.

Nessa seção não foi abordado o processo de compilação do código já que o procedimento pode ser encontrado na seção 2.2.

### 4.3 Passando projetos para o FPGA

Nessa seção será abordado como passar projetos para a placa. Ela é reconhecida e possui seus *drivers* instalados na máquina, mas não possui permissões e nem informações de sua localidade na porta USB da máquina em uso.

Primeiro, deve-se abrir o terminal Linux e dar o seguinte comando, certificando-se que o *kit* se encontra ligado na porta USB:

“\$ lsusb”.

Serão exibidos todos os dispositivos em portas USB da máquina, como mostrado na Figura 22.

```
teste@teste-300E5M-300E5L:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 2232:1080 Silicon Motion
Bus 001 Device 004: ID 0cf3:e500 Atheros Communications, Inc.
Bus 001 Device 003: ID 3938:1031
Bus 001 Device 006: ID 09fb:6001 Altera Blaster
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
teste@teste-300E5M-300E5L:~$
```

Figura 22: Lista de dispositivos USB no sistema. Observe que foi identificado o *kit* como Altera Blaster.

Pode-se observar que o FPGA foi identificado pelo sistema, com o nome Altera Blaster. Assim, pode-se retirar a placa da porta USB e executar o seguinte comando para interromper as atividades relacionadas a placa:

```
“$ sudo killall -9 jtagd”.
```

Feito isso, deve-se adicionar um arquivo texto com as instruções do sistema para a placa. Execute o seguinte comando:

```
“$ sudo touch /etc/udev/rules.d/usbblaster.rules”
```

Agora basta executar os seguintes cinco comandos no terminal para adicionar as instruções ao arquivo criado:

```
$ echo "SUBSYSTEM==\"usb\", ATTR{idVendor}==\"09fb\",
ATTR{idProduct}==\"6001\", MODE=\"0666\"" | sudo tee
/etc/udev/rules.d/usbblaster.rules -a
```

```
$ echo "SUBSYSTEM==\"usb\", ATTR{idVendor}==\"09fb\",
ATTR{idProduct}==\"6002\", MODE=\"0666\"" | sudo tee
/etc/udev/rules.d/usbblaster.rules -a
```

```
$ echo "SUBSYSTEM==\"usb\", ATTR{idVendor}==\"09fb\",
ATTR{idProduct}==\"6003\", MODE=\"0666\"" | sudo tee
/etc/udev/rules.d/usbblaster.rules -a
```

```
$ echo "SUBSYSTEM==\"usb\", ATTR{idVendor}==\"09fb\",
ATTR{idProduct}==\"6010\", MODE=\"0666\"" | sudo tee
/etc/udev/rules.d/usbblaster.rules -a
```

```
$ echo "SUBSYSTEM=="usb\", ATTR{idVendor}=="09fb\",
ATTR{idProduct}=="6810\", MODE=="0666\" | sudo tee
/etc/udev/rules.d/usbblaster.rules -a
```

Certifique-se de que tais informações tenham sido adicionadas, restando apenas executar o seguinte comando para que as alterações sejam reconhecidas:

“sudo udevadm control –reload”

Inicie novamente o projeto no Quartus II e passe o projeto para a placa. Dessa vez, ela deverá ser reconhecida.

## 5 Exemplo de implementação: MUX 2x1

Um MUX 2x1 é um Multiplexador com duas entradas, um sinal de controle e uma saída. A função desse circuito combinacional é copiar para saída uma das entradas de acordo com o padrão aplicado no sinal de controle.

Considerando que tal Multiplexador é representado pelas entradas SW[1] e SW[0], pelo sinal de controle SW[2] e pela saída LEDR[0], seu funcionamento pode ser definido por

$$LEDR[0] = \begin{cases} SW[0], & SW[2] = 0, \\ SW[1], & SW[2] = 1 \end{cases}$$

cuja tabela verdade é representada por:

SW[2]	SW[1]	SW[0]	LEDR[0]
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

O circuito lógico implementado dentro do MUX 2x1 pode ser visto na Figura 23.

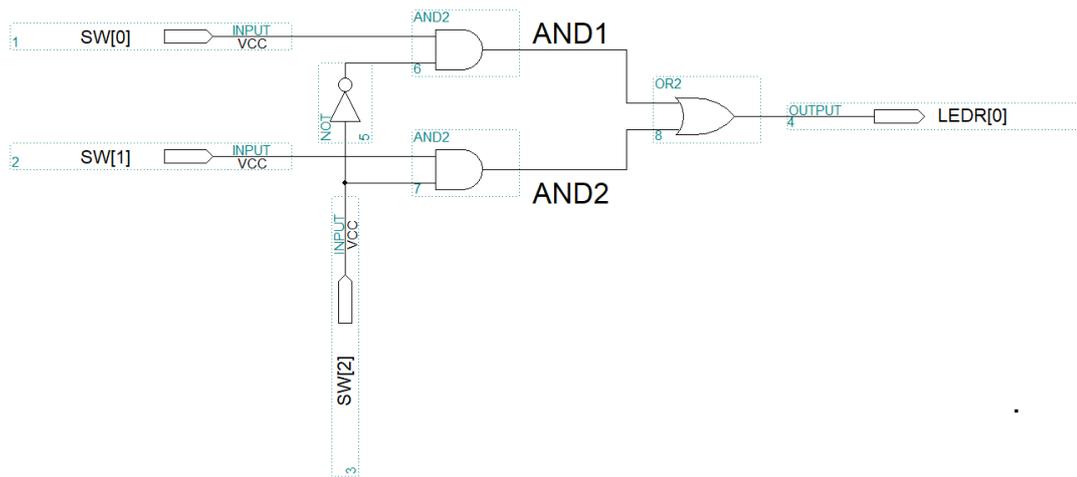


Figura 23: Circuito lógico implementado dentro do MUX 2x1.

Para transmitir essa ideia de implementação para linguagem Verilog, é necessário primeiramente declarar quantos *switches* serão utilizados e os LEDs presentes na placa. Dessa forma, definimos, no módulo do projeto, essas especificações:

```
module DE1_LITE(
    input [2:0] SW,
    output [9:0] LEDR
);
```

Após a declaração das entradas e da saída, é necessário que as portas lógicas “AND1” e “AND2”, ilustradas na figura 23, sejam ligadas na porta lógica “OR”. Em Verilog, isso é realizado através do comando “*wire*”:

```
wire AND1, AND2;
```

Por fim, precisa-se implementar o comportamento das entradas e da saída através do comando “*assign*”. Esse comando é utilizado para atribuições de funções lógicas no escopo dos circuitos combinacionais:

```
assign AND1 = ~SW[2] & SW[0];
assign AND2 = SW[2] & SW[1];
assign LEDR[0] = AND1 | AND2;
endmodule
```

O código implementado no Quartus II pode ser visto na figura 24.

```

1
2 //=====
3 // This code is generated by Terasic System Builder
4 //=====
5
6 module DE10_LITE(
7
8     //////////////// SW ////////////////
9     input [2:0] SW,
10
11     //////////////// LED ////////////////
12     output [9:0] LEDR
13
14 );
15
16 //=====
17 // REG/WIRE declarations
18 //=====
19
20 wire AND1, AND2;
21
22 assign AND1 = ~SW[2] & SW[0];
23 assign AND2 = SW[2] & SW[1];
24 assign LEDR[0] = AND1 | AND2;
25
26
27 endmodule
28
29
30
31

```

Figura 24: Implementação do MUX 2x1 em Verilog.

## Referências

- [Ato] AtomMiner. Fighting altera usb-blaster on ubuntu. “<https://blog.atomminer.com/fighting-altera-usb-blaster-on-ubuntu/>”. Acesso em: 08/08/2020.
- [Chu08] Pong P. Chu. *FPGA Prototyping By VERILOG Examples*. Wiley, 2008.
- [Inta] Intel. The nios ii embedded “hello world” lab: For the de10-lite development kit. “[https://www.intel.com/content/dam/altera-www/global/en\\_US/uploads/5/55/Hello\\_World\\_Lab\\_Manual\\_DE10\\_Lite.pdf](https://www.intel.com/content/dam/altera-www/global/en_US/uploads/5/55/Hello_World_Lab_Manual_DE10_Lite.pdf)”. Acesso em: 08/08/2020.
- [Intb] Intel. What is an fpga? fpga documentation. “<https://www.intel.com/content/www/us/en/products/programmable/fpga/new-to-fpgas/resource-center/overview.html>”. Acesso em: 08/08/2020.
- [PET] Grupo PET-Tele. <http://www.telecom.uff.br/pet>. Acesso em: 08/08/2020.
- [Pro] Programa de Educação Tutorial - PET. [http://portal.mec.gov.br/index.php?option=com\\_content&view=article&id=12223&ativo=481&Itemid=480](http://portal.mec.gov.br/index.php?option=com_content&view=article&id=12223&ativo=481&Itemid=480). Acesso em: 08/08/2020.
- [Stu] FPGA 4 Students. Verilog projects. “<https://www.fpga4student.com/p/verilog-project.html>”. Acesso em: 08/08/2020.
- [Ter] Terasic. De10-lite user manual. “[https://www.intel.com/content/dam/altera-www/global/en\\_US/portal/dsn/42/doc-us-dsnbk-42-2912030810549-de10-lite-user-manual.pdf](https://www.intel.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-2912030810549-de10-lite-user-manual.pdf)”. Acesso em: 08/08/2020.