

## MCTA024 – Sistemas Digitais

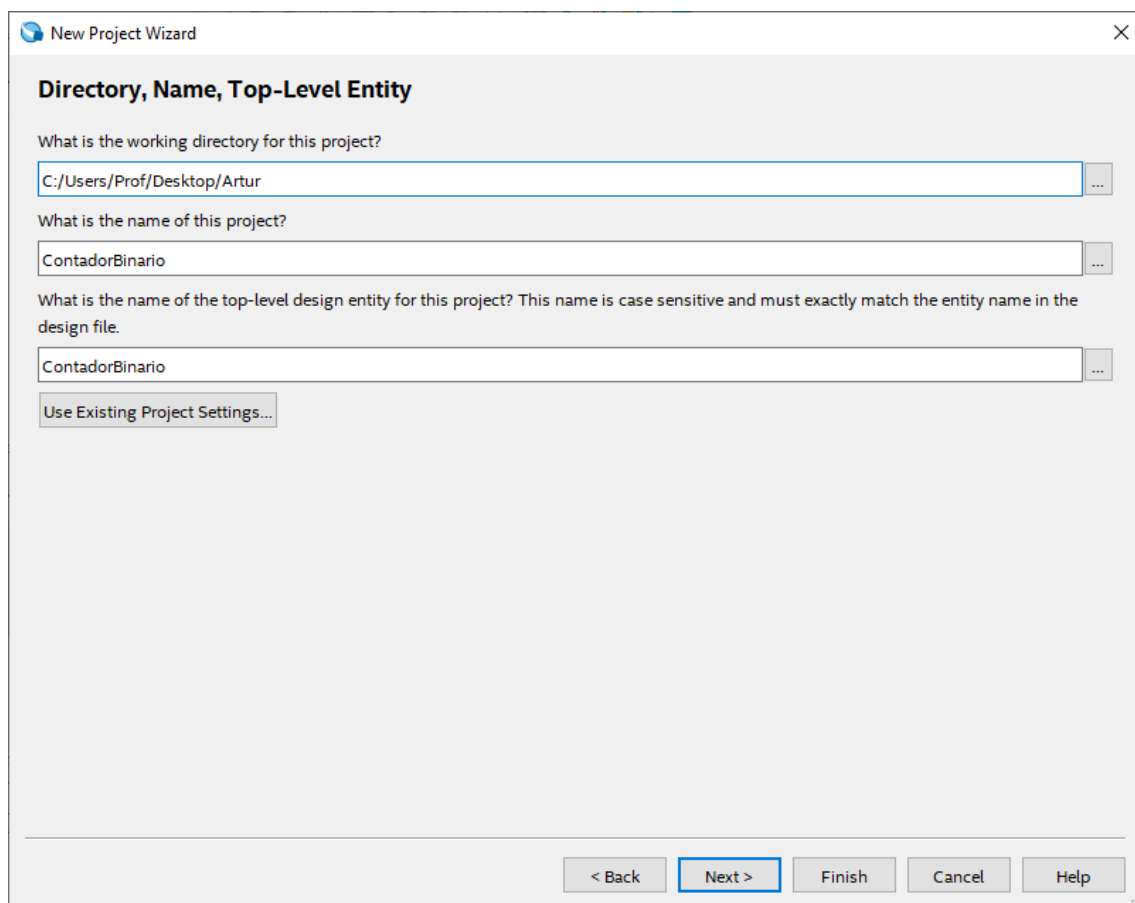
josé artur quilici-gonzalez – CMCC - UFABC

### Lab 3 – Placa DE10-Lite da Altera

Nesta experiência, vamos aprender a criar uma **configuração** para o projeto de um **Contador Binário**, baixar a configuração para a placa **DE10-Lite da terasic/Intel/Altera** e depois fazer o **teste** do circuito na placa.

### Contador Binário

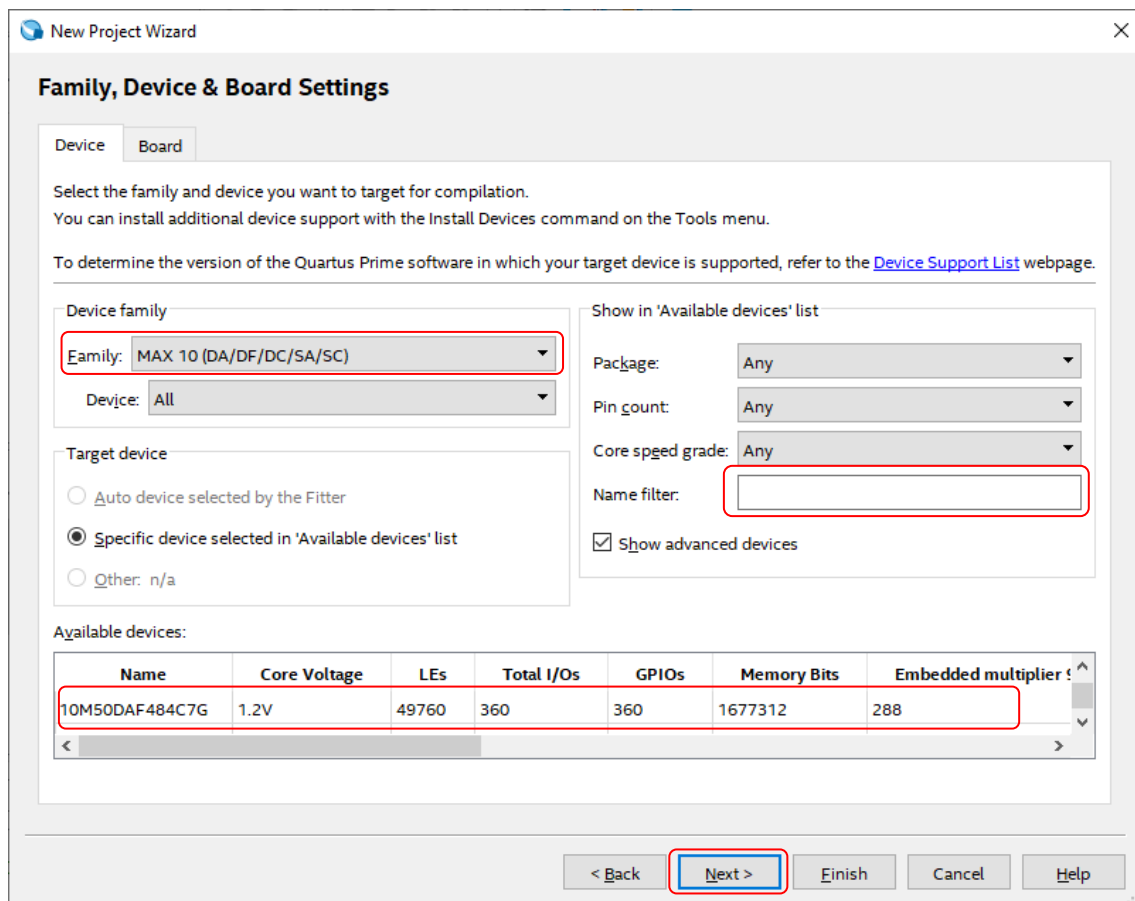
1. Primeiramente crie uma pasta com seu nome, e dentro dela crie um novo projeto no **Quartus Prime Lite Edition** cuja **“entity”** receba o nome de **“ContadorBinario”** (sem aspas e sem acento em Binário), conforme ilustra a **Fig. 1**.



**Fig. 1 – Criação de Projeto no Quartus Prime Lite Edition**

2. Na próxima janela que se abre (**“Project Type”**), deixe a opção default **“Empty projet”**. Na janela **“Add Files”**, não preencha nada, simplesmente dê um clique em **“Next”**.

3. Como agora vamos testar o circuito em uma placa de *hardware* (e não apenas uma simulação virtual), é muito importante especificar exatamente o código da **FPGA** da placa. Para a placa **DE10-Lite** que temos em nossos labs, a **FPGA** é da família “**MAX 10 (DA/DF/DC/AS/SC)**”, e o dispositivo (“**Available devices**”) é o “**10M50DAF484C7G**”, conforme ilustra a **Fig. 2**. (Ao encontrar o dispositivo “10M50DAF484C7G”, dê um clique sobre esta seleção). Alternativamente, você pode digitar o código **10M50DAF484C7G** no campo “**Name filter:**” para que o respectivo dispositivo apareça no campo “**Available devices**”. Nesse caso, você também precisa dar um clique sobre o dispositivo para que a seleção se efetive.



**Fig. 2 – Escolha da FPGA “MAX10(DA/DF/DC/SA/SC”, com o código “10M50DAF484C7G”.**

4. A seguir tome a opção “**Next**” desta página e da “**EDA Tool Settings**”. Quando aparecer a janela “**Summary**” clique em “**Finish**” para que o novo projeto seja criado.

5. Vá em “**File**” e tome a opção “**New...**”, escolha a opção “**VHDL File**” e dê um “**OK**”. Para não perdermos tempo com digitações, o arquivo “**ContadorBinario.vhd**” foi anexado à pasta **Lab3** desta experiência. Basta copiá-lo para o editor de VHDL do **Quartus Prime Lite** e salvar com o nome “**ContadorBinario**” e extensão do tipo “**vhd**”.

6. Agora vamos fornecer ao **Quartus Prime Lite** o arquivo que atribui nomes aos elementos físicos da placa **DE10-Lite**, como as chaves, os **LEDs** etc. Clique em “**Assignments**”, depois em “**Import Assignments**” e quando aparecer o quadro “**Import Assignments**”, na opção “**File name:**” carregue o arquivo “**DE10\_Lite.qsf**”, também anexado na pasta **Lab3**, conforme ilustra a **Fig. 3**.

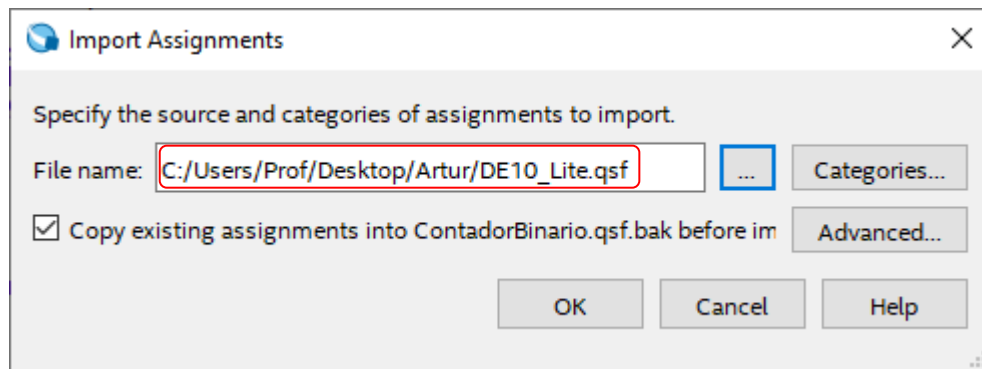


Fig. 3 – O arquivo “DE10\_Lite.qsf” atribui nomes aos elementos físicos da placa DE10-Lite

7. Depois destas etapas, podemos fazer a compilação do projeto. Dê um clique no triângulo azul (“**Start Compilation**”) que aparece na parte superior do **Quartus Prime Lite** (veja Fig. 4).

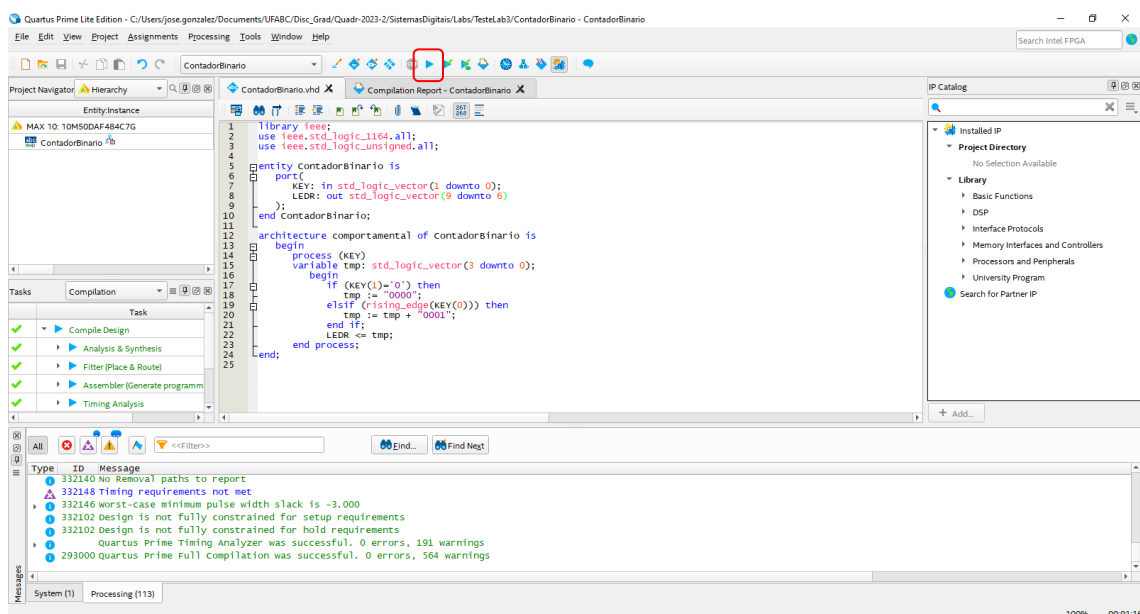


Fig. 4 – Para compilar o projeto, clicar no triângulo azul (“**Start Compilation**”).

Se não houver nenhum erro no projeto, a mensagem “**Quartus Prime Full Compilation was succesfull. 0 errors, 564 warnings**” deve aparecer na seção “**Messages**” (região inferior).

8. Faça a conexão entre a placa **DE10-Lite** e o PC através do cabo USB. Se os LEDs se acenderem e os “**displays seven segments**” iniciarem uma contagem crescente, a placa está funcionando bem.

9 Vamos agora “programar” ou “configurar” a FPGA, isto é, criar um programa para injetar 1s e 0s na FPGA através de um caminho serial chamado “**JTAG**”. Na parte esquerda do **Quartus Prime Lite Edition**, na seção “**Task**”, dê dois cliques na opção “**Program Device (Open Programmer)**”, como ilustra a Fig. 5.

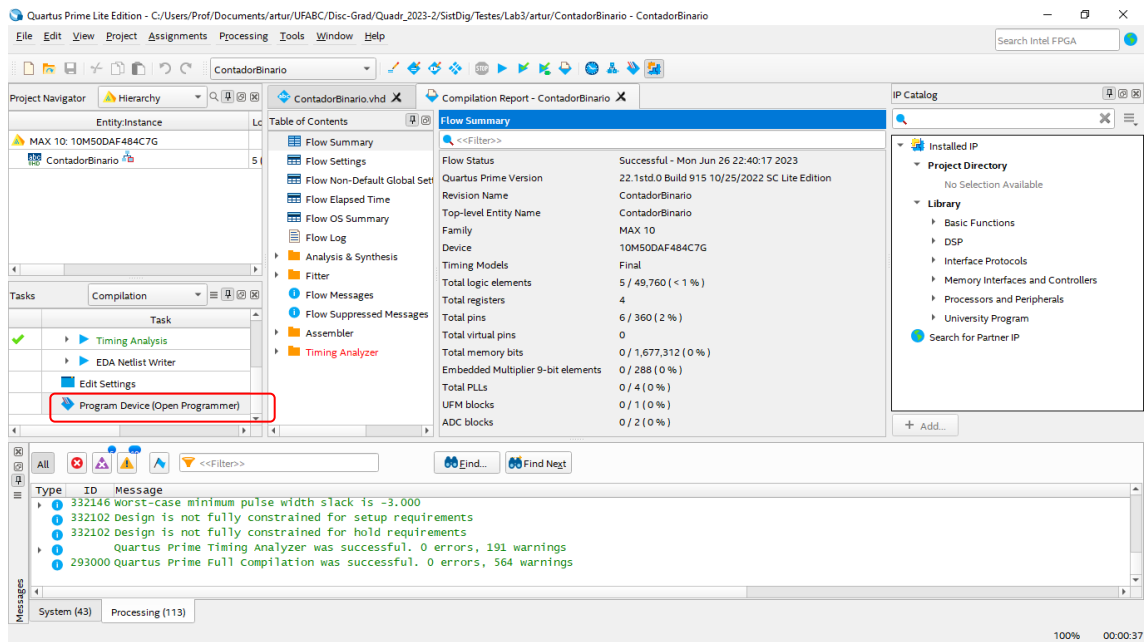


Fig. 5 – Depois da compilação, é possível criar a configuração para a FPGA.

A tela do “**Programmer**” deve aparecer (Fig. 6). Certifique-se que as opções “**USB-Blaster**” e “**Mode: JTAG**” estejam ajustadas.

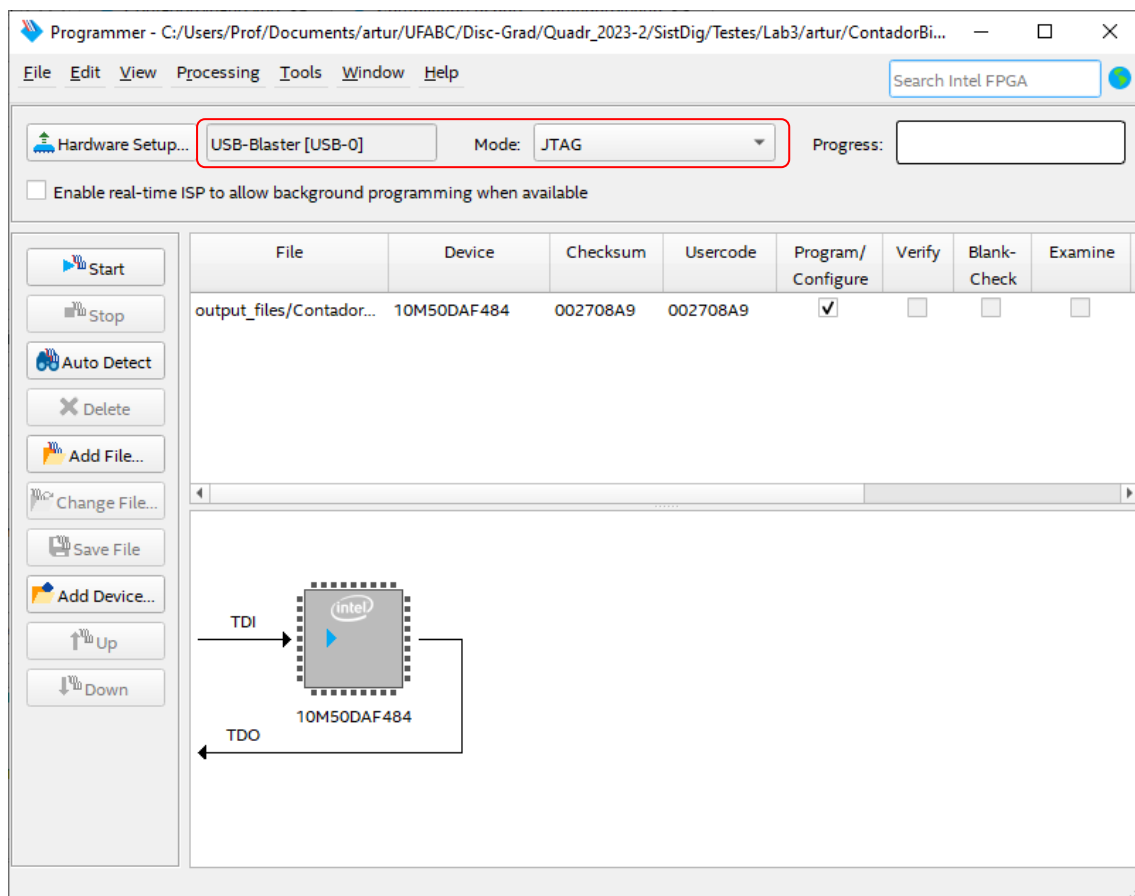
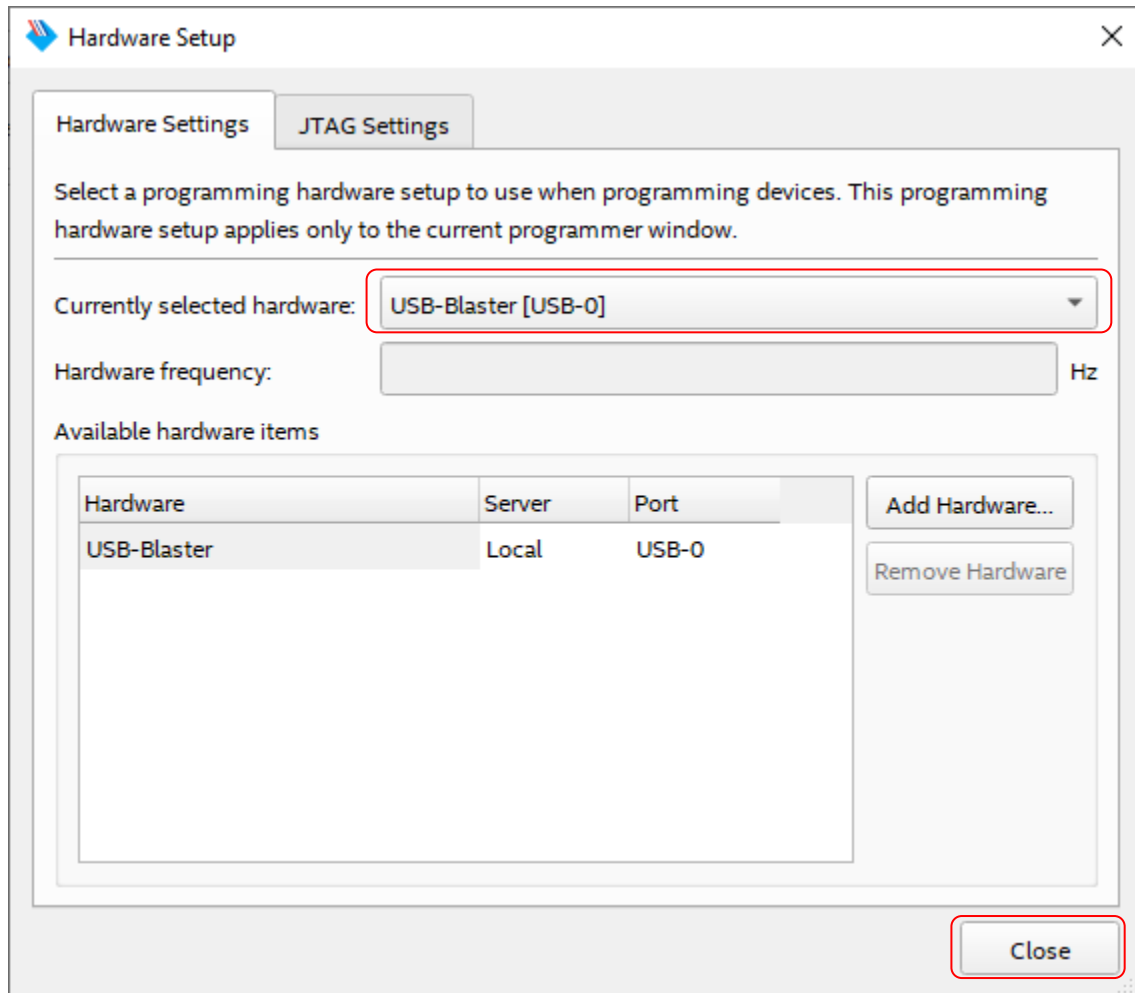


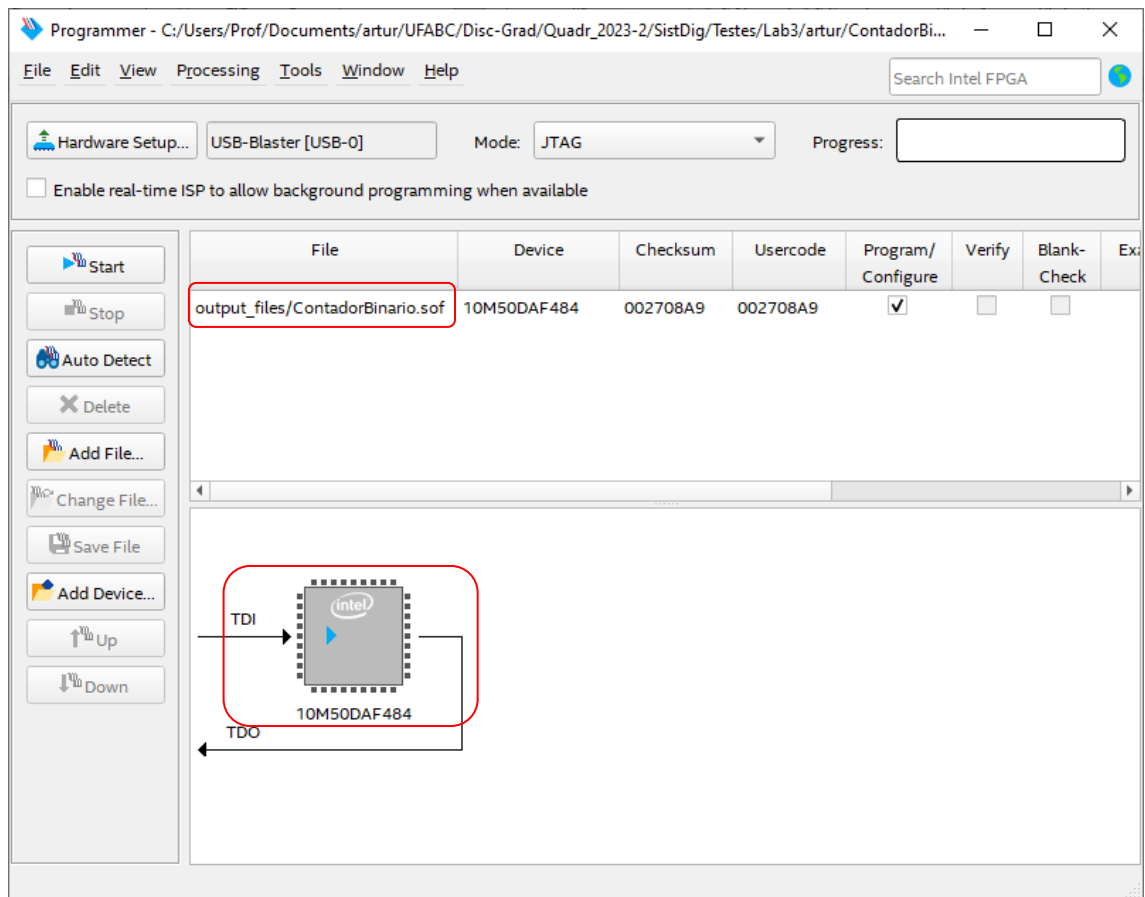
Fig. 6 – A tela do programador (“**Programmer**”) da FPGA.

10. Caso não apareça a opção ***“USB-Blaster [USB-0]”*** (na primeira vez que a placa **DE10-Lite** for ligada), clique em ***“Hardware Setup ...”*** e, na janela de ***“Hardware Setup”*** que se abre (**Fig. 7**), ajuste a opção ***“USB-Blaster [USB-0]”*** no campo ***“Currently selected hardware”***. Depois, clique em ***“Close”***.



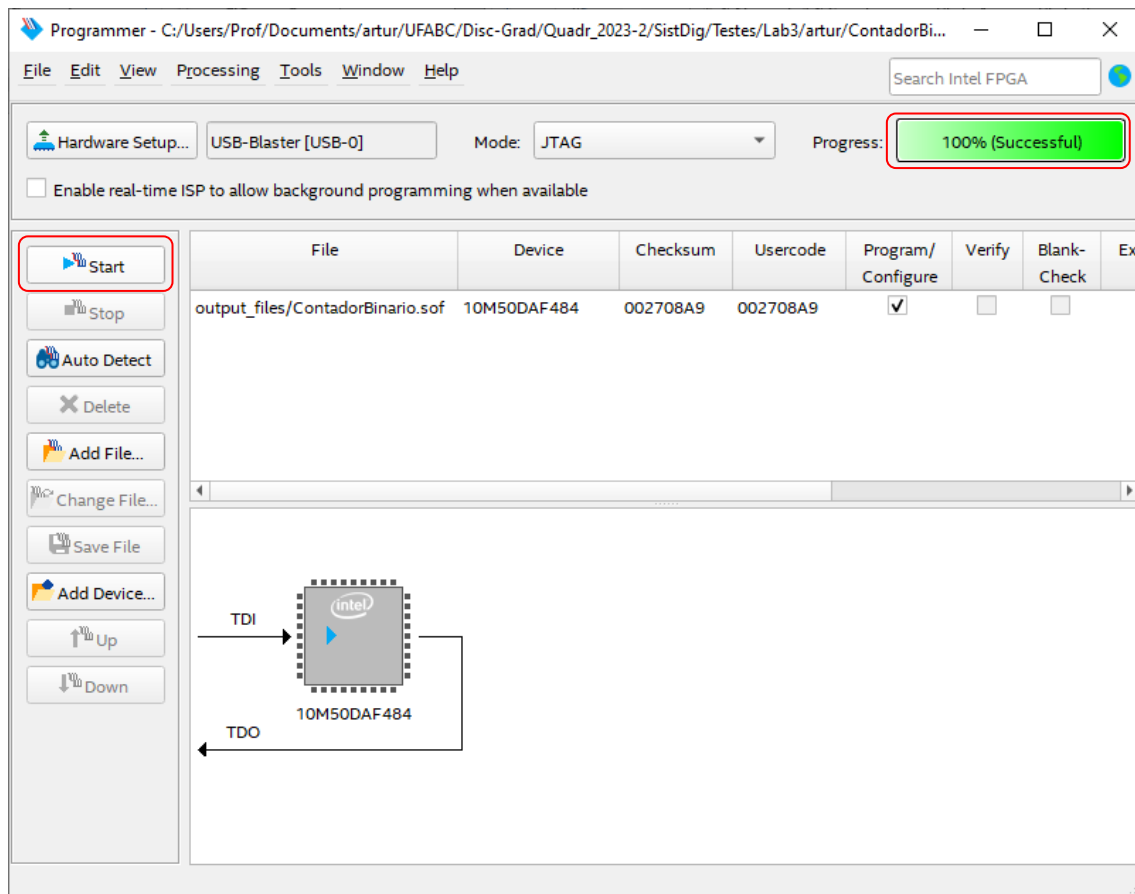
**Fig. 7 – Ajuste do hardware *“USB-Blaster [USB-0]”***

Depois desse ajuste, devem aparecer o arquivo ***“ContadorBinario.sof”*** e o dispositivo da **Intel**, conforme ilustra a **Fig. 8**.



**Fig. 8 – Dispositivo “10M50DAF484” da Intel que foi detectado através do cabo USB.**

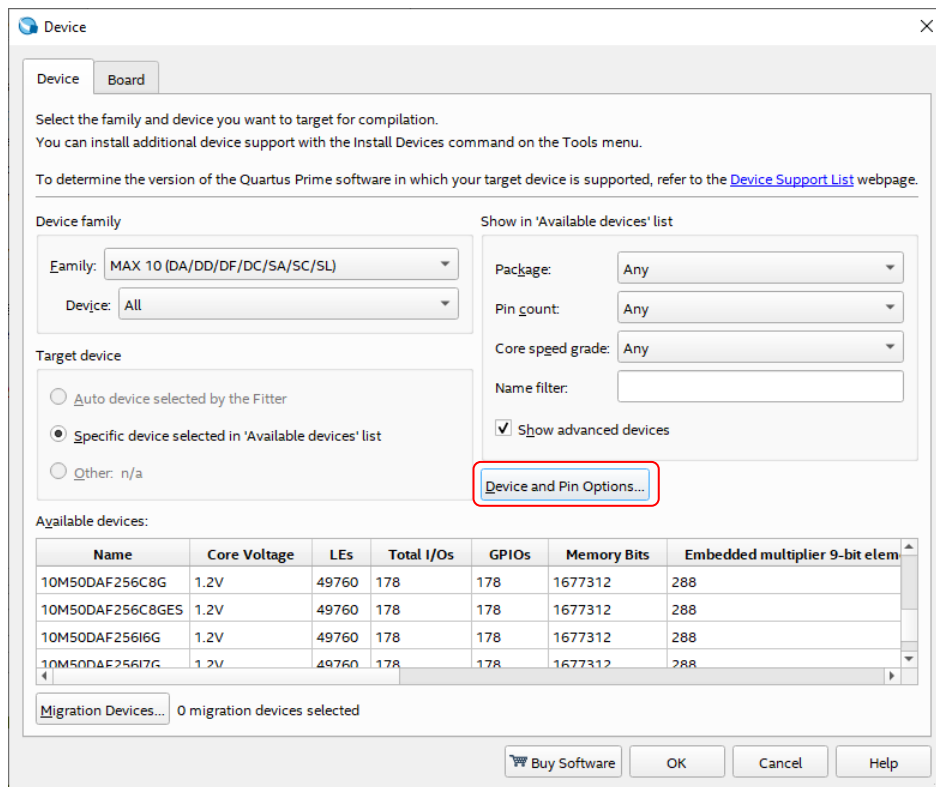
11. Com a placa “*DE10-Lite*” ligada, dê um clique em “*Start*”. Depois de algum tempo deve aparecer o aviso “*Progress: 100% (Successful)*” (Fig. 9).



**Fig. 9 – O aviso “Progress: 100% (Successful)” indica que a configuração está terminada.**

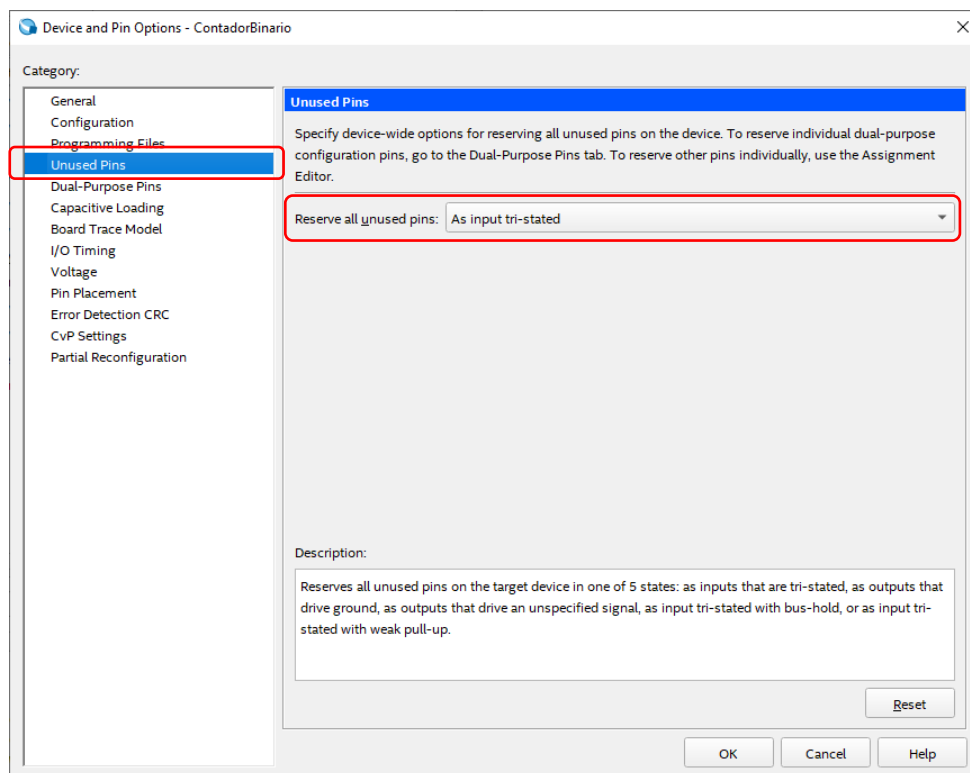
12. Feche a janela da **Fig. 9** (não há necessidade de salvar). Agora o circuito já pode ser testado na placa **DE10-Lite** através das chaves, dos botões, dos LEDs, dos displays sete segmentos etc. Ao apertar sucessivamente a chave “**KEY0**”, os LEDs vermelhos devem começar a acender em ordem binária crescente. A chave “**KEY1**” funciona como “**Reset**” do contador.

13. Os LEDs não utilizados, por default, ficam levemente acesos. Se você preferir que os LEDs não utilizados fiquem completamente apagados, na parte superior do painel do **Quartus Prime Lite Edition**, vá em “**Assignments**” e depois “**Device**”. Na janela que se abre, clique na aba “**Device and Pin Options...**”, situada na parte central e à direita da janela, como ilustra a **Fig. 10**.



**Fig. 10 – Opções para o ajuste do valor default dos pinos não utilizados.**

14. Na janela seguinte que se abre, escolha as opções ***“Unused Pins”*** e ***“As input tri-stated”***, conforme ilustra a **Fig.11** e salve a opção escolhida clicando em ***“OK”***.



**Fig. 11 – Ajuste dos pinos não utilizados como “As input tri-stated”.**



## Seven Segment Display

Uma forma simples de acionar o “Seven Segment Display” (Fig. 12) da placa **DE10-Lite** é através de um **decodificador**, como mostrado na Fig. 13. (Este código VHDL foi adaptado do livro *FPGA Prototyping by VHDL Examples*, de Pong P. Chu, originalmente concebido para ser implementado numa placa da Xilinx).

O decodificador para o “Seven Segment Display” recebe 4 bits em suas entradas e os decodifica em 7 bits, que acionam o **display de 7 segmentos** (Fig. 12). Cada um dos segmentos recebe uma numeração entre 0 e 6, e são ativos em “0”, ou seja, ele se apaga quando recebe “1” no respectivo segmento.

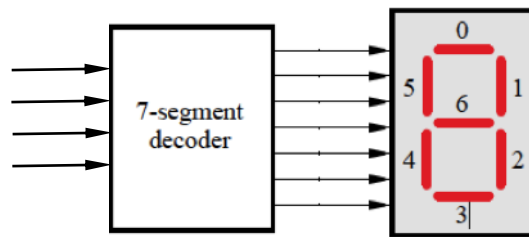


Fig. 12 – Decodificador e “Seven Segment Display”.

Na placa **DE10-Lite** há 6 displays, identificados como HEX0x, HEX1x, HEX2x, HEX3x, HEX4x e HEX5x, sendo cada um deles declarado como um **“HEX0: out std\_logic\_vector(0 to 6)”**.

No código VHDL mostrado na Fig. 13, o vetor foi declarado em ordem inversa (ou seja, **6 downto 0**), desta forma introduzindo propositalmente “erros” para que você possa corrigi-los e entender como é acionado um display de 7 segmentos. Os primeiros seis algoritmos estão corretamente decodificados, mas o restante precisa ser corrigido.

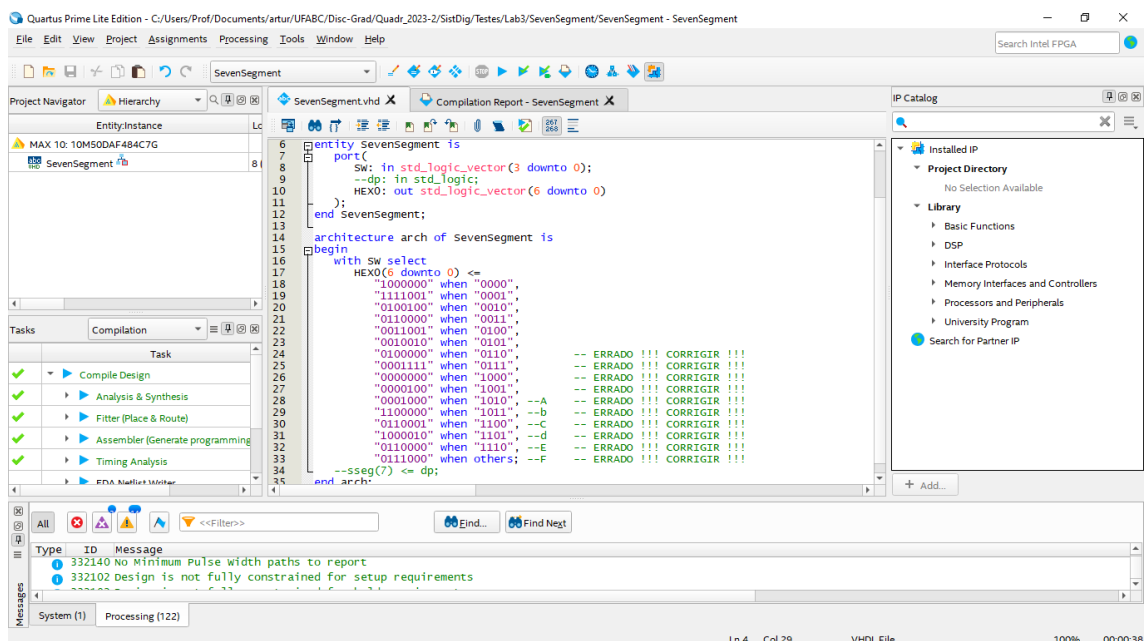


Fig. 13 – Código VHDL simplificado para o “Seven Segment Display” da placa DE10-Lite.

15. Corrija o código da Fig. 13 para que apareça no display **HEX0** os 16 caracteres hexadecimais, ou seja, **0, 1, 2, ..., 9, A, b, C, d, E e F**. (Para não gastar tempo com digitação, o código "**SevenSegment.vhd**" foi anexado a esta pasta **Lab3**).

### Contador Hexadecimal

Fazendo com que a saída do **Contador Binário** seja a entrada do "**Seven Segment Display**", é possível implementar um **Contador Hexadecimal**. Para isso, é preciso alterar a saída do **Contador Binário** (i.e., desconectá-la dos **LEDRs**), assim como alterar a entrada do "**Seven Segment Display**" (i.e., desconectá-la das chaves **SW**) e conectar ambos através de um "**signal**" de 4 bits denominado "Ponte".

16. Crie um novo projeto que implemente o **Contador Hexadecimal** descrito acima, com uma **Entity** denominada "**ContadorHexadecimal**" de tal forma que pressionando-se **KEY0**, o display **HEX0** mostre todos os 16 caracteres hexadecimais em ordem crescente.

### Bibliografia

Chu, P. P. **FPGA Prototyping para VHDL Examples**. Hoboken, NJ, John Wiley & Sons, Inc., 2008.

Intel/Altera. **Laboratory Exercise 1**. In

[https://ftp.intel.com/Public/Pub/fpgaup/pub/Teaching\\_Materials/current/Laboratory\\_Exercises/Digital\\_Logic/VHDL/lab1.pdf](https://ftp.intel.com/Public/Pub/fpgaup/pub/Teaching_Materials/current/Laboratory_Exercises/Digital_Logic/VHDL/lab1.pdf) . Acessado em 26.06.23.