



Departamento de Ciência da Computação

Prof. Bruno de Abreu Silva

GCC260 – Laboratório de Circuitos Digitais

Dispositivos lógicos programáveis e o software Quartus Prime

1. Objetivos

- Conhecer os dispositivos lógicos programáveis ou reconfiguráveis;
- Implementar um circuito combinacional utilizando o software Quartus Prime.

2. Embasamento teórico

Dispositivos lógicos programáveis (PLDs – *Programmable Logic Devices*) são circuitos integrados que podem ser reorganizados internamente de acordo com alguma descrição de hardware. Sendo assim, é possível atribuir funções lógicas específicas a um CI de propósito geral. Embora seja usado o termo “programável”, não se trata de gravar um programa para ser executado e sim configurar (ou reconfigurar) uma descrição de hardware. Por isso, o termo *dispositivo reconfigurável* é mais adequado para esses tipos de CIs e será usado como sinônimo de PLD ao longo deste material.

As principais vantagens em usar dispositivos reconfiguráveis são:

- Diminuição do espaço ocupado pelo circuito na placa;
- Redução do consumo de energia;
- Dificulta a cópia não autorizada do circuito ou mesmo do *layout* da placa, uma vez que toda a lógica está implementada em um único CI;
- Flexibilidade na alteração de projetos por meio da reconfiguração.

Os dispositivos reconfiguráveis podem ser divididos em três categorias:

- SPLD (*Simple Programmable Logic Device*): Dispositivo mais simples composto por arranjos de portas lógicas AND e OR. Sendo assim, permite a implementação de funções lógicas simples;
- CPLD (*Complex Programmable Logic Device*): Permite a implementação de funções lógicas mais complexas, uma vez que é basicamente um conjunto de diversos SPLDs conectados entre si;
- FPGA (*Field Programmable Gate Array*): Possui diversos elementos de processamento simples. Por isso, permite uma densidade (quantidade de funções dentro de um *chip*) e velocidade muito superiores às encontradas em CPLDs.

2.1. SPLD

A Figura 1 apresenta a estrutura típica de um SPLD. Ele possui um conjunto de entradas e seus valores invertidos que podem ter suas conexões configuradas com o arranjo de portas AND e OR. Outros componentes podem ser usados como flip-flops e multiplexadores.

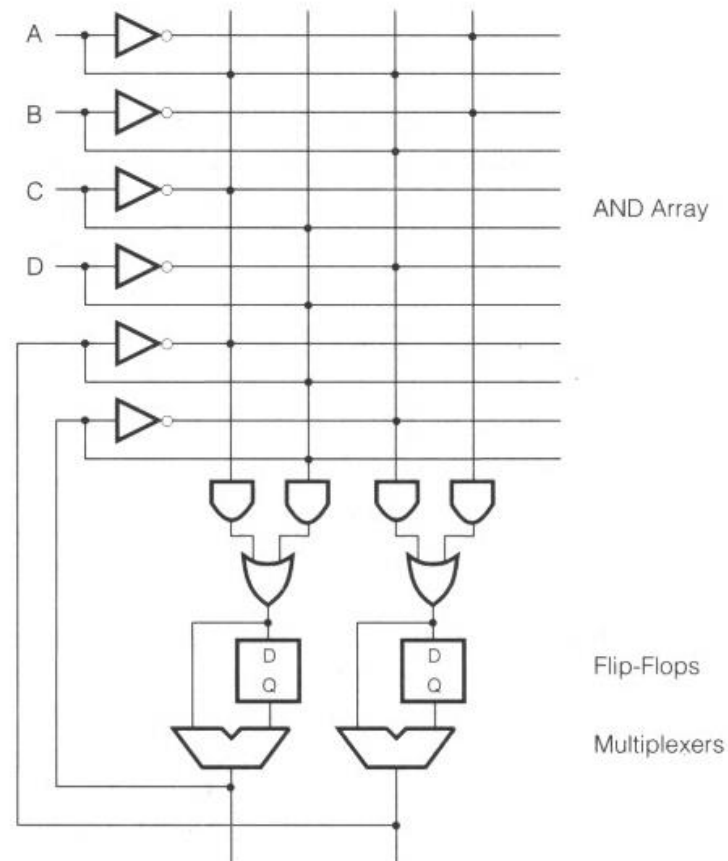


Figura 1: Estrutura típica de um SPLD.

2.2. CPLD

A Figura 2 apresenta a estrutura típica de um CPLD.

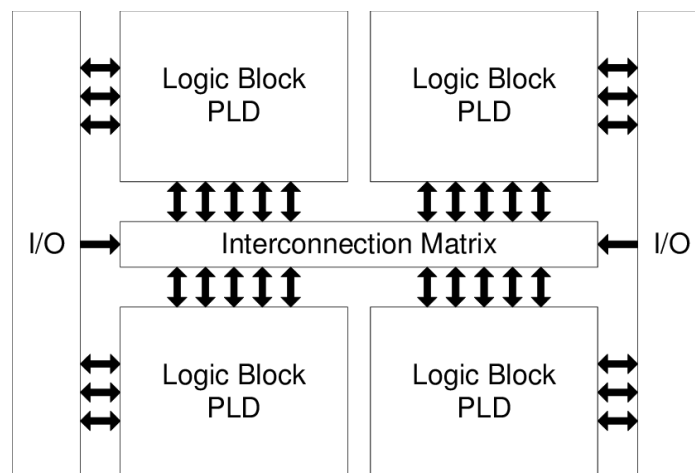


Figura 2: Estrutura típica de um CPLD.

Conforme dito anteriormente, o CPLD é um conjunto de SPLD (na figura chamados de *Logic Block PLD*) contendo também uma matriz de interconexões programáveis (*Interconnection Matrix*) e blocos de entrada e saída (*I/O*).

O que determina a capacidade de um CPLD é a quantidade de blocos lógicos que ele possui. Além disso, ele é um dispositivo não-volátil, mantendo assim a configuração mesmo se a alimentação for retirada do circuito.

2.3. FPGA

Na Figura 3, é apresentada a estrutura interna de um FPGA.

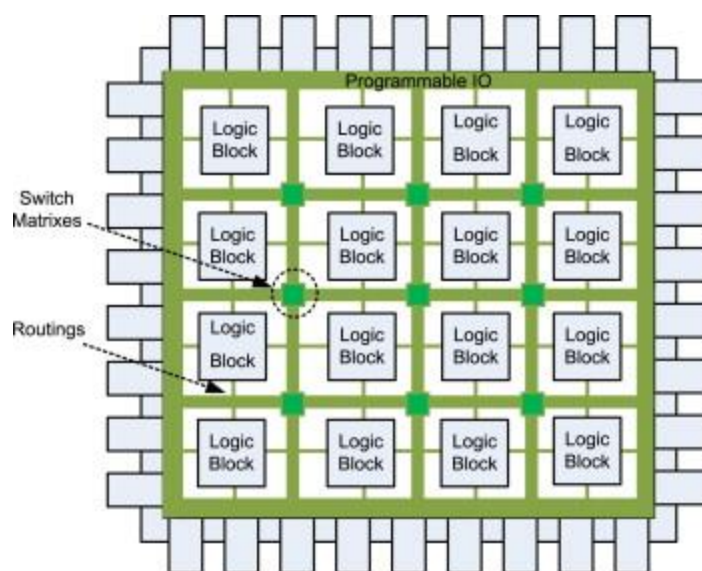


Figura 3: Estrutura interna de um FPGA.

A estrutura de um FPGA é composta por blocos lógicos, uma matriz de interconexões programáveis e blocos de entrada e saída programável. Embora seja muito parecida com a estrutura de um CPLD, o FPGA utiliza blocos lógicos simples em grande quantidade, enquanto o CPLD utiliza poucos blocos complexos.

Os FPGAs atuais possuem blocos lógicos com funções mais específicas, como circuitos de memória RAM, multiplicadores-acumuladores e até processadores completos.

Normalmente, os FPGAs são dispositivos voláteis, ou seja, perdem sua configuração quando são desligados. Por isso, muitas vezes, existe uma memória não-volátil, como memória ROM, associada ao FPGA, funcionando como uma memória de programação.

2.4. CPLD versus FPGA

Os dois dispositivos são utilizados atualmente. Entretanto, cada um deles possui características diferentes e a escolha de qual é o mais adequado depende da aplicação. O CPLD é normalmente usado para implementação de lógica propriamente dita e é um CI mais barato. O FPGA é mais caro e é mais usado em aplicações de processamento digital de sinais, pois permite implementar circuitos matemáticos de altíssima velocidade.

2.5. Fluxo de projeto utilizando ferramentas EDA

Para desenvolver projetos para CPLDs e FPGAs é necessário o auxílio de softwares específicos para projeto automático de circuitos, ou ferramentas EDA (*Electronic Design Automation*). Cada fabricante costuma fornecer a ferramenta EDA necessária para projeto em seus dispositivos. Uma das principais fabricantes de FPGA, a Altera/Intel, fornece o software *Quartus Prime*. O fluxo de projeto é composto pelas seguintes etapas:

- *Entrada do projeto*: nessa etapa é feita a descrição do projeto. Essa descrição pode ser feita de diferentes formas, como diagrama esquemático, diagrama de estados ou linguagem de descrição de hardware (como VHDL e Verilog);
- *Síntese*: o processo de síntese traduz a descrição de hardware em uma estrutura chamada de *netlist* definindo as ligações dos elementos internos do componente. Essa etapa não depende do dispositivo utilizado;
- *Simulação funcional (functional simulation)*: após a descrição do projeto, é possível realizar simulação para verificar o funcionamento do circuito de acordo com a lógica implementada. A simulação funcional não depende do dispositivo utilizado e apenas verifica o funcionamento lógico do circuito, sem considerar o *timing* do circuito;
- *Implementação*: Nessa etapa, a *netlist* criada na síntese é mapeada dentro da estrutura do dispositivo escolhido para o projeto. Esse processo é chamado de *fitting* e *place and route*. Como resultado dessa etapa, é gerado um arquivo binário, chamado de *bitstream*, que será usado para configurar o dispositivo;
- *Simulação temporal (timing simulation)*: as características de temporização, relacionadas aos tempos de propagação, subida e descida dos elementos internos são inerentes ao componente escolhido. Para verificar se essas características estão consistentes e não causarão falhas no projeto, é feita uma simulação de temporização, usando as características do componente escolhido para o projeto;
- *Download do bitstream*: O *bitstream* é configurado no dispositivo usando um gravador, que geralmente usa a interface JTAG (*Join Test Action Group*).

3. O software Quartus Prime

Esta seção apresenta o software Quartus Prime e um breve tutorial baseado na versão 18.1. Os estudantes deverão seguir as instruções do tutorial para concluírem esta atividade prática.

O software Quartus Prime é disponibilizado pela Intel para realizar o projeto e configuração de circuitos em seus dispositivos reconfiguráveis. A versão utilizada nessa disciplina é o **Quartus Prime Lite Edition, versão 18.1**. É um software gratuito e seu instalador pode ser baixado neste link: <https://www.intel.com.br/content/www/br/pt/products/details/fpga/development-tools/quartus-prime/resource.html>.

No laboratório pode existir mais de uma versão do Quartus instalada.

Para encontrar a versão correta, digite “*Quartus Prime*” na barra de pesquisa ao lado do botão Iniciar do Windows, conforme destacado no retângulo verde na Figura 4 e procure o resultado correto, conforme indicado pelo retângulo vermelho na Figura 4. Clique no aplicativo “Quartus (Quartus Prime 18.1)” e, após alguns segundos, o software estará aberto.

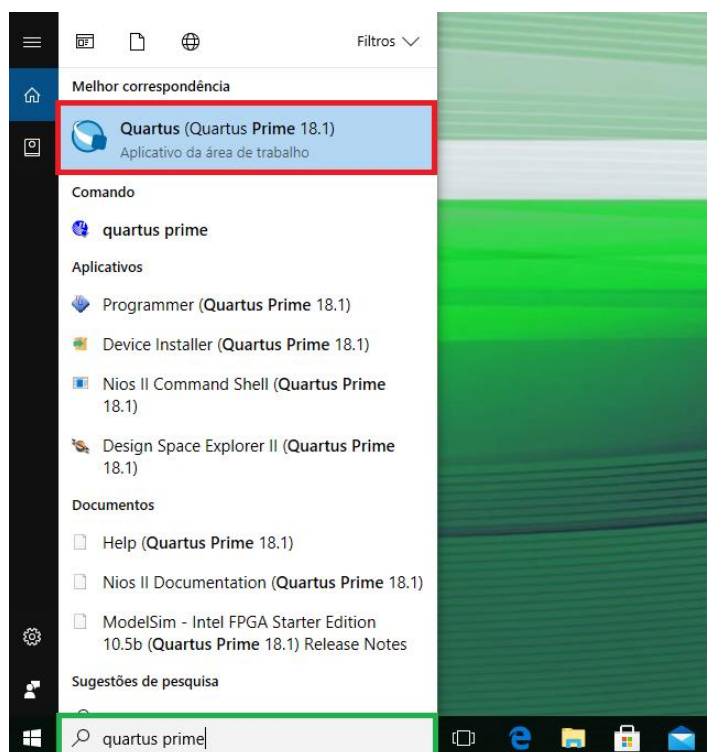


Figura 4: Iniciando o Quartus (Quartus Prime 18.1).

DICA: Mantenha apenas uma janela do Quartus Prime aberta de cada vez.

Se for a primeira vez que você está abrindo o Quartus Prime em sua conta no laboratório, aparecerá uma janela, conforme a Figura 5, onde você deverá marcar a opção “*Run the Quartus Prime software*” e clique em OK. **Após, isso,**

será necessário abrir o software Quartus Prime novamente. Porém, nas próximas vezes em que for aberto, essa janela não será mais exibida e o software abrirá normalmente.

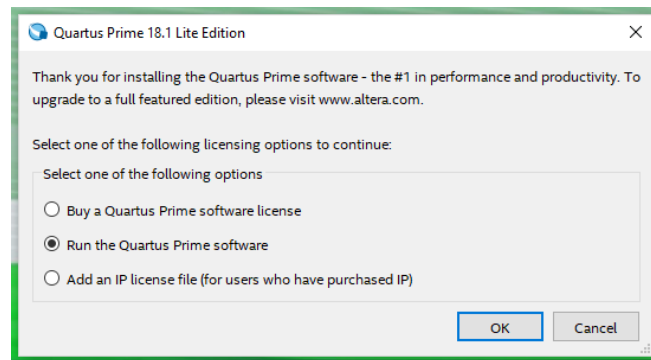


Figura 5: Primeira inicialização do Quartus Prime.

Nas próximas vezes em que for aberto, Quartus Prime apresentará a tela inicial, conforme ilustrado na Figura 6. Ao centro, pode-se ver, em azul, a janela “Home” com alguns atalhos para projetos recentes, documentação, treinamento, suporte e botões para criar um novo projeto ou abrir algum projeto existente.

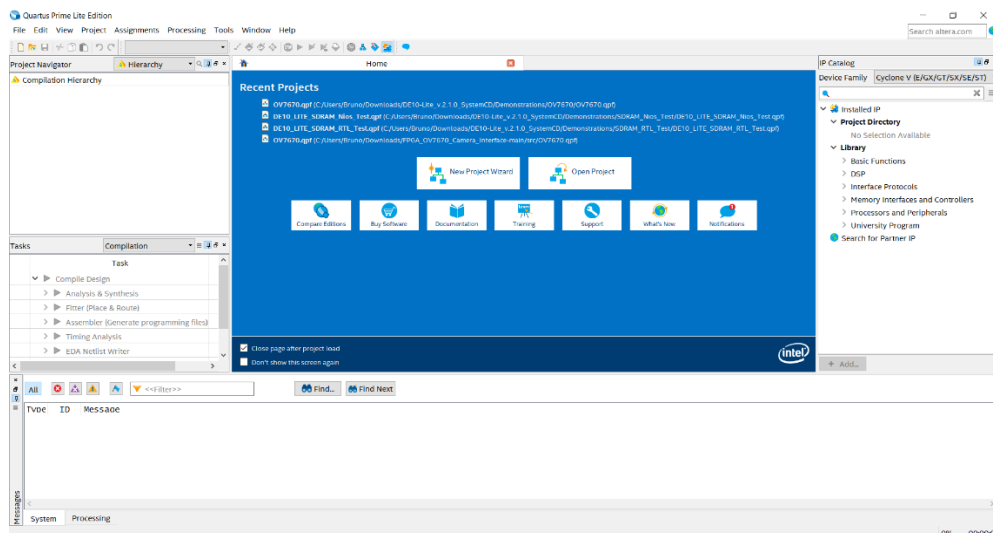


Figura 6: Quartus Prime após inicialização.

Outros elementos estão presentes na IDE (*Integrated Development Environment*) do Quartus Prime. No topo da janela principal, se encontra o Menu com todos os submenus *File*, *Edit*, *View*, *Project*, *Assignments*, *Processing*, *Tools*, *Window* e *Help*. Logo abaixo do Menu, se encontra a barra de ferramentas. Abaixo da barra de ferramentas e à esquerda, se localiza o *Project Navigator*, onde são exibidas as informações sobre o projeto que está aberto. Abaixo do *Project Navigator*, se encontram as tarefas do fluxo de compilação (*Tasks*). No canto inferior, há a barra de mensagens, onde os erros e avisos, como os de compilação, são exibidos. No canto direito, se encontra o IP Catalog, onde é possível encontrar a biblioteca com diversos componentes que podem ser usados nos projetos. E, por fim, na janela maior azul ao centro, é onde serão

exibidos os arquivos de descrição do hardware do projeto: diagramas esquemáticos, de estados ou código-fonte em linguagem de descrição de hardware.

3.1. Criando o primeiro projeto

Para descrever um hardware no Quartus Prime, é necessário primeiramente criar um projeto. Esse projeto consiste, além dos arquivos de descrição e simulação criados pelo usuário, em alguns arquivos criados automaticamente pelo software que armazenarão informações importantes como o dispositivo utilizado, pinos de entrada e saída, restrições de velocidade do circuito, entre outras coisas.

Antes de criar o projeto, crie uma pasta nova no *Desktop* ou em seu pendrive, chamada *TutorialQuartusPrime* (**sem espaços**). Ao criar o projeto, informaremos que o mesmo será armazenado nessa pasta.

Após criar a pasta, acesse o *Menu File->New Project Wizard...* Será aberta a janela mostrada na Figura 7.

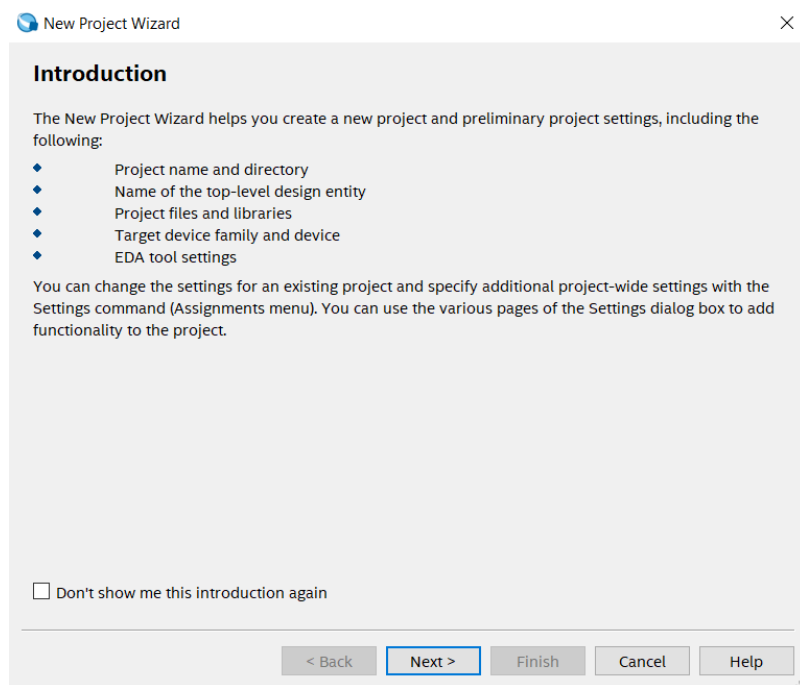


Figura 7: New Project Wizard: Introduction.

DICA: Sempre crie o projeto em uma pasta em que há permissão do sistema para criar, editar e ver arquivos para não ter erros de compilação. No laboratório, ao criar uma pasta no Desktop ou no pendrive como sugerido, as permissões são garantidas.

DICA: Nunca utilize espaços ou caracteres especiais ou acentuação nos nomes de projetos ou nas pastas do diretório onde o projeto será salvo. O Quartus Prime não gosta desse tipo de coisa. Procure mantê-lo de bom humor.

A janela introdutória apresenta o que será definido na criação do projeto, como o nome e diretório, nome do arquivo principal do projeto (*top-level design entity*), arquivos e bibliotecas usadas, dispositivo e configurações da ferramenta EDA. Após ler as informações dessa janela, clique em *Next*.

Na janela seguinte, *New Project Wizard: Directory, Name, Top-Level Entity*, mostrada na Figura 8, serão informados o diretório em que o projeto será criado (**no nosso caso na pasta *TutorialQuartusPrime*, no *Desktop* ou *pendrive***), o nome do projeto (nomeie como *tutorial*) e o nome do arquivo principal (*top-level design entity*) que é preenchido automaticamente com o mesmo nome do projeto. **Não se esqueça de colocar o diretório correto!**

Após preencher as informações conforme a Figura 8, clique em *Next*.

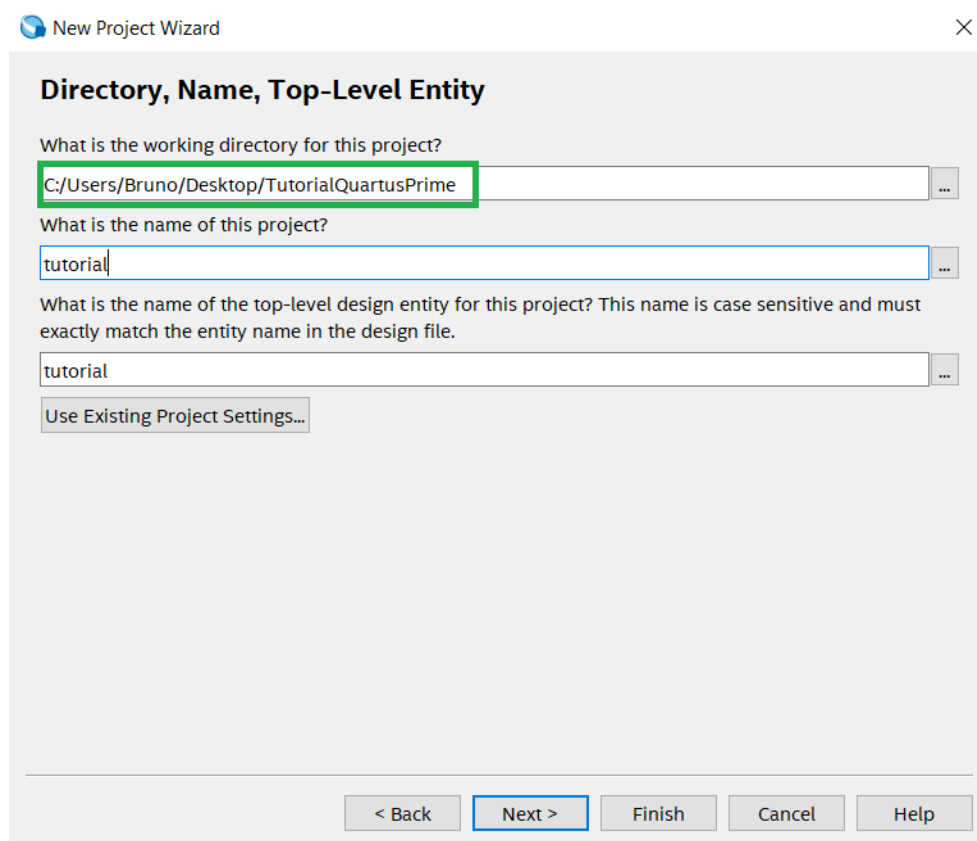


Figura 8: *New Project Wizard*, página 1.

DICA: Sempre altere o diretório onde o Quartus Prime irá criar o seu projeto, pois o diretório padrão é o diretório de instalação do Quartus Prime.

A tela seguinte, Figura 9, permite criar um projeto vazio ou utilizar um template de projeto. Neste tutorial, deixaremos marcada a opção *Empty project*, que é o projeto vazio e clique em *Next*.

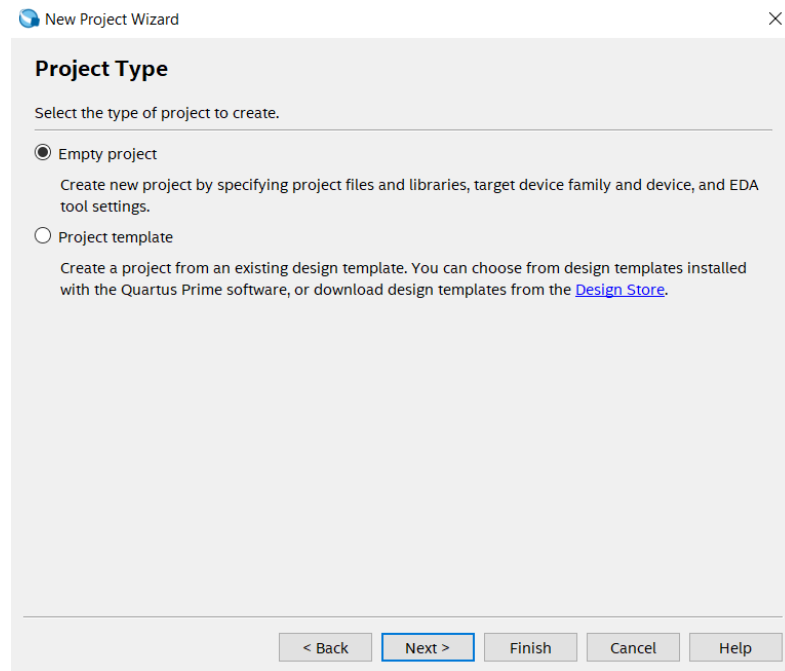


Figura 9: Project Type.

A próxima página, Figura 10, é utilizada quando já se tem arquivos de descrição de hardware desenvolvidos previamente e que serão reaproveitados no novo projeto. Nesse caso, são adicionados. Porém, como estamos criando um projeto do zero, não há nenhum arquivo a ser adicionado. Portanto, clique em *Next*.

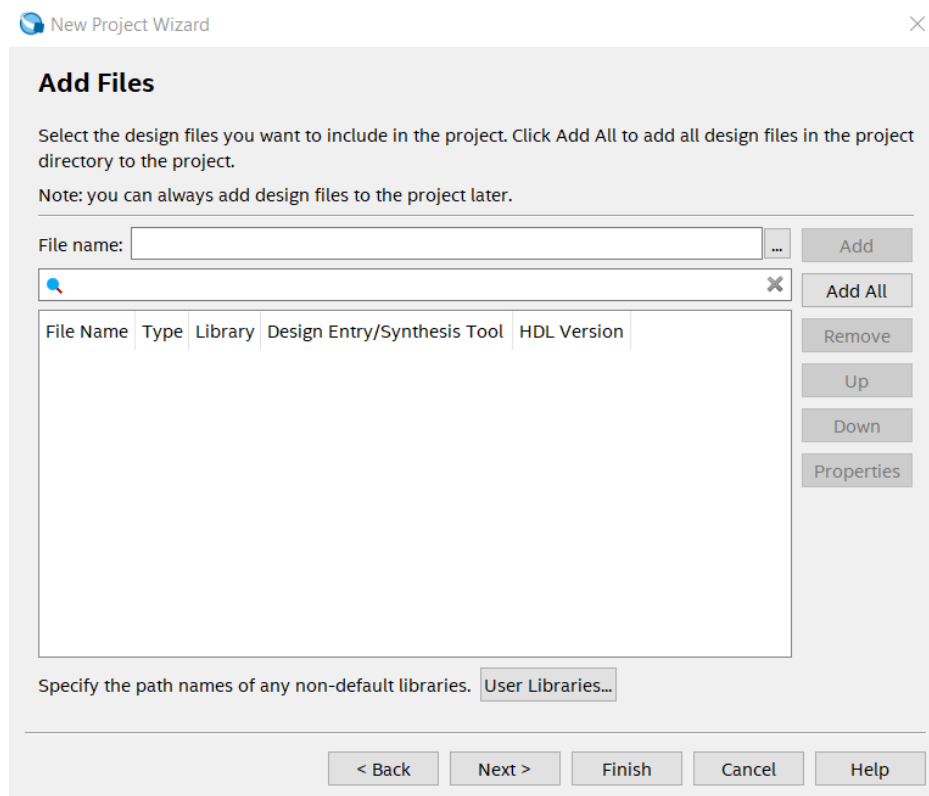


Figura 10: Add files.

Caso houvesse arquivos de descrição de hardware já criados, bastaria clicar em "...", selecionar os arquivos e clicar em *Add*. Dessa forma, o projeto já seria criado com arquivos. Entretanto, como já foi dito, no caso deste tutorial, criaremos um projeto vazio inicialmente, para depois criarmos os arquivos. Então, não há nada a ser feito nessa janela.

A próxima janela é uma das mais importantes. Nela é especificado qual o dispositivo que será usado pelo projeto. No laboratório, usaremos um kit didático, chamado DE10-Lite, com a FPGA MAX10. Para selecioná-lo, primeiro clique em Board, conforme indicado pelo retângulo verde na Figura 11. Em seguida, altere Family para MAX 10 e, depois, selecione a linha MAX 10 DE10-Lite, como na Figura 12. **Por fim, na parte inferior da janela, deixe marcada a opção *Create top-level design file*.** Clique em *Next*.

Family, Device & Board Settings

Device: **Board**

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: Cyclone V (E/GX/GT/SX/SE/ST)

Device: All

Target device

☐ Auto device selected by the Filter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Core speed grade: Any

Name filter:

☒ Show advanced devices

Available devices:

Name	Core Voltage	ALMs	Total I/Os	GPIOs	GXB Channel PMA	GXB Channel PCS	PCIe
5CGXFC7C6U19A7	1.1V	56480	268	240	6	6	1
5CGXFC7C6U19C6	1.1V	56480	268	240	6	6	1
5CGXFC7C6U19C7	1.1V	56480	268	240	6	6	1
5CGXFC7C6U19I7	1.1V	56480	268	240	6	6	1
5CGXFC7C7F23C8	1.1V	56480	268	240	6	6	1
5CGXFC7C7U19C8	1.1V	56480	268	240	6	6	1
5CGXFC7C7U19C9	1.1V	56480	268	240	6	6	1

< Back Next > Finish Cancel Help

Figura 11: Family, Device & Board Settings.

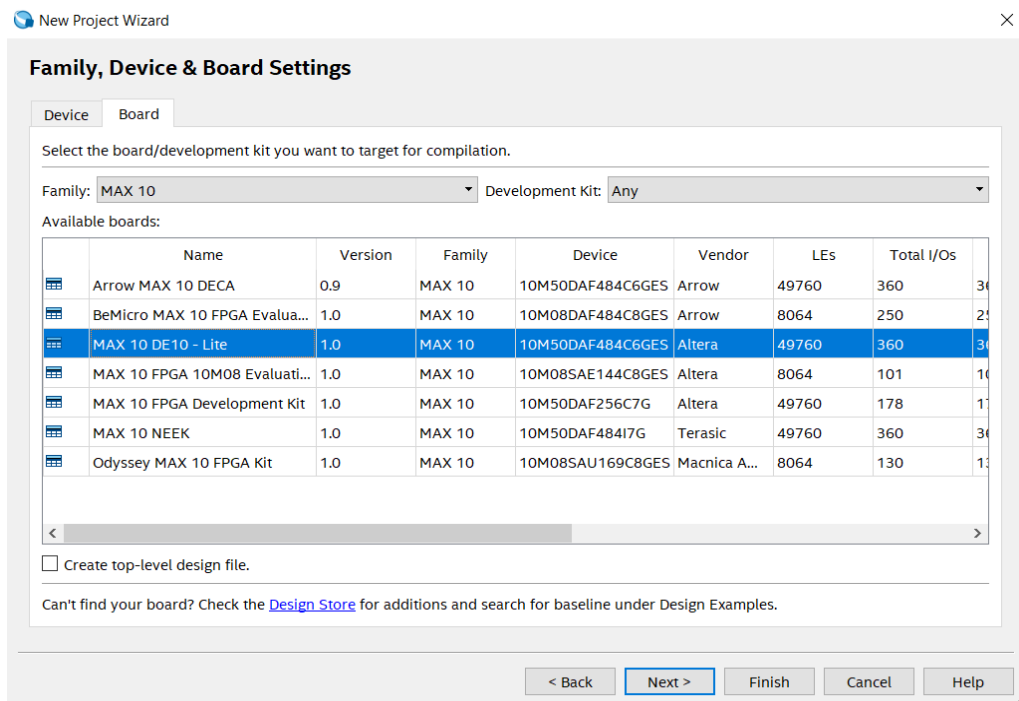


Figura 12: Selecionando MAX 10 DE10-Lite.

As próximas janelas são para escolher a ferramenta de simulação e para ver o resumo das configurações do projeto criado. Como não iremos alterar mais nada, simplesmente clique em Finish para o projeto ser criado.

Após o projeto ser criado, o *Project Navigator* já exibirá as informações do projeto, como seu nome e o dispositivo associado a ele.

Na pasta *TutorialQuartusPrime*, foram criados dois arquivos e duas pastas:

- **tutorial.qpf:** *Quartus Project File*. É o arquivo que armazena informações gerais sobre o projeto, como versão, horário de criação;
- **tutorial.qsf:** *Quartus Settings File*. Armazena informações sobre os pinos de entrada e saída, restrições de velocidade do projeto, entre outras informações que serão adicionadas posteriormente;
- **db:** pasta que armazena base de dados sobre o projeto (pasta atualizada automaticamente pelo Quartus Prime);
- **devkits:** Arquivos com as informações sobre o kit MAX 10 DE10-Lite.

DICA: Na próxima vez em que for abrir um projeto já criado pelo Quartus Prime, abra pelo arquivo .qpf.

3.2. Desenvolvendo um projeto usando diagrama de blocos (esquemático)

Embora o Quartus Prime permita diferentes formas de descrever o projeto, inicialmente utilizaremos o diagrama de bloco. Para isso, devemos criar um arquivo de diagrama de bloco acessando o menu *File->New....* Na janela que abrir, selecione a opção *Block Diagram/Schematic File*, conforme a Figura 13, e clique em *OK*. Clique em *File->Save As...* e **salve o arquivo com o mesmo nome do projeto, ou seja, *tutorial* e na mesma pasta do projeto**. O arquivo de diagrama de blocos possui a extensão *.bdf* (*Block Diagram File*).

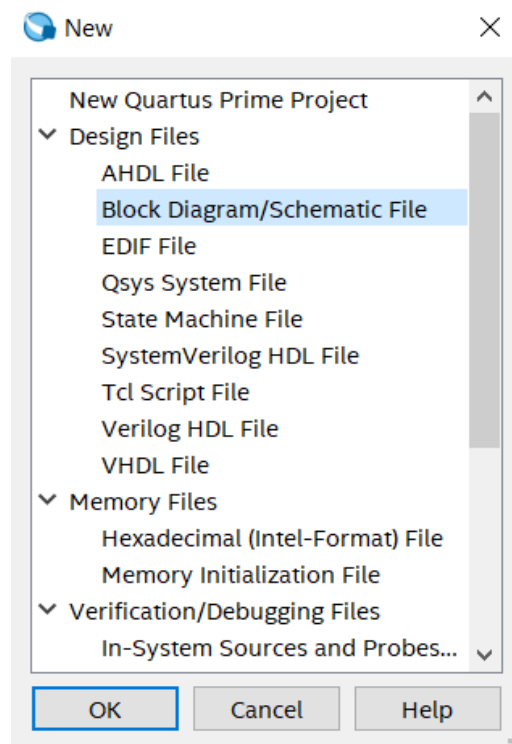


Figura 13: Criando um arquivo esquemático.

DICA: Sempre salve o arquivo diagrama de bloco principal com o mesmo nome do projeto e sempre salve todos os arquivos na pasta do projeto. Verifique o diretório destino sempre que for salvar.

A janela principal do Quartus Prime exibirá agora o arquivo de diagrama de blocos vazio recém-criado, conforme a Figura 14.

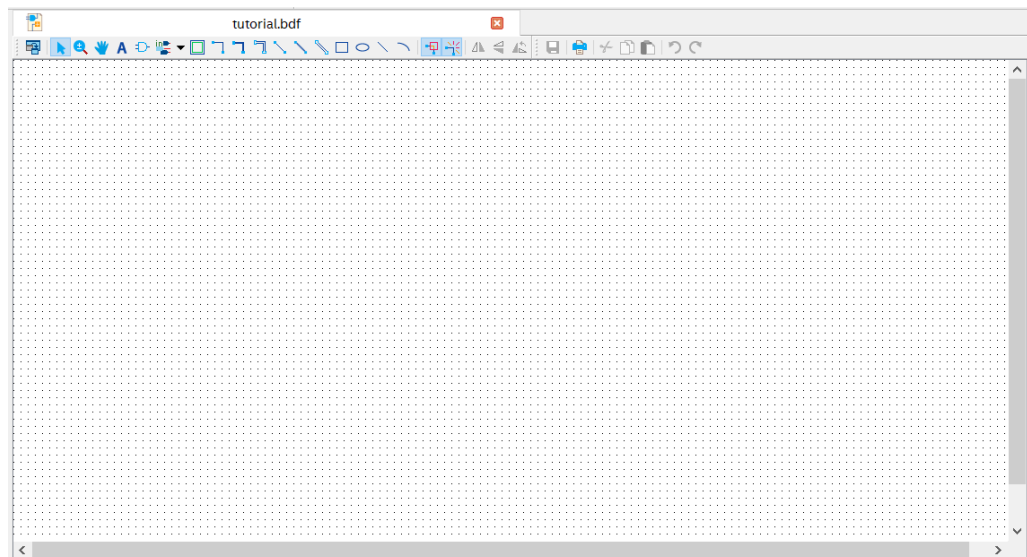


Figura 14: Arquivo de diagrama de bloco.

Na região pontilhada é que serão inseridos os componentes do circuito. No canto superior da janela principal, existe uma barra de ferramentas com as principais funções descritas a seguir:

- **Detach Window:** Juntar/Separar a janela do diagrama de blocos da janela principal;
- **Selection Tool:** permite selecionar os elementos inseridos no arquivo;
- **Text Tool:** permite inserir texto no arquivo, com fins explicativos ou de documentação;
- **Symbol Tool:** Permite inserir componentes no arquivo (detalhado a seguir);
- **Orthogonal Node Tool:** realiza conexão simples (1 bit) entre componentes;
- **Orthogonal Bus Tool:** realiza conexões de conjuntos de sinais associados (mais de 1 bit).

Por meio da *Symbol Tool* é possível inserir componentes no arquivo. Clique no ícone dela (ou dê um duplo-clique na região pontilhada do arquivo). A Figura 15 apresenta a janela que será aberta.

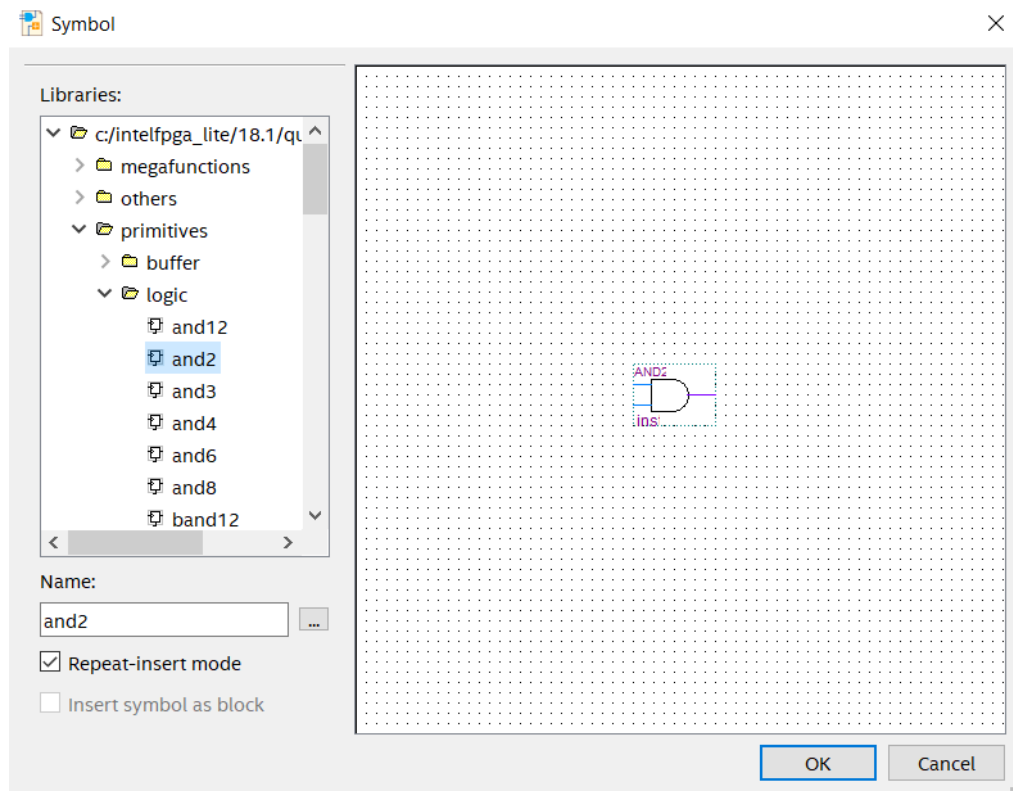


Figura 15: Symbol Tool.

Nessa janela, em *Libraries*, é possível explorar as diversas coleções de componentes do Quartus Prime. A seguir, são listadas as mais relevantes:

- **Megafunctions:** tratam-se de funções complexas já prontas. Não serão tratadas nessa prática, pois demandam configuração;
- **Others:**
 - **Maxplus2:** essa coleção possui modelos de componentes equivalentes aos da família TTL. Dessa forma é possível realizar qualquer combinação lógica feita com componentes discretos dentro do dispositivo reconfigurável;
- **Primitives:** componentes básicos
 - **Buffers:** diversos tipos de buffers, incluindo os de saída *tri-state*;
 - **Logic:** elementos lógicos básicos (AND, OR, NOR, XOR, etc) com uma, duas ou várias entradas;
 - **Others:** elementos diversos;
 - **Pin:** diversos tipos de pinos suportados pelo componente: entrada (input), saída (output) ou bidirecional;
 - **Storage:** elementos de armazenamento de dados, como flip-flops.

Após selecionar o componente desejado, basta clicar em OK que será possível inseri-lo no diagrama de blocos. Nesta prática, usaremos os componentes **and2**, **and3**, **or2**, **input** e **output**.

Na prática deste tutorial, vocês irão implementar a seguinte função lógica $S = AB + BCD$, conforme a Figura 16. Porém, observe que na Figura 16, os nomes dos pinos de entrada e saída estão diferentes, pois já foram colocados os nomes dos componentes de entrada e saída do kit que serão usados, conforme a seguir:

- A: receberá o nome de SW[0], por usar a *switch* (chave) número 0 do kit;
- B: receberá o nome de SW[1], por usar a *switch* (chave) número 1 do kit;
- C: receberá o nome de SW[2], por usar a *switch* (chave) número 2 do kit;
- D: receberá o nome de SW[3], por usar a *switch* (chave) número 3 do kit;
- S: receberá o nome de LEDR[0], por usar o LED vermelho número 0.

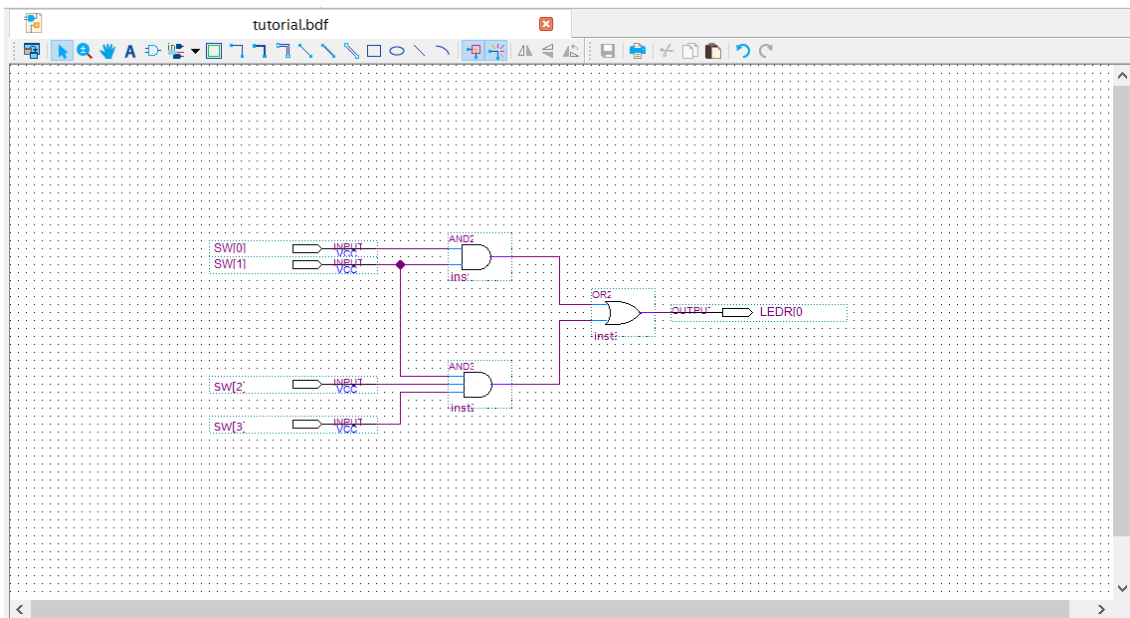


Figura 16: $S = AB + BCD$.

Na **Symbol Tool**, procure os componentes **input** para inserir as entradas, **output** para inserir as saídas, **and2** para AND de duas entradas, **and3** para AND com três entradas e **or2** para OR com duas entradas. Para realizar as conexões utilize o **Orthogonal Node Tool** e para mudar os nomes das entradas e saídas, clique duas vezes sobre os pinos de entrada e saída e digite o nome desejado.

Salve o arquivo, clicando em *File->Save*.

Com o arquivo de diagrama de blocos aberto, vá em *Project->Set as Top-Level Entity* para definir este arquivo como o arquivo principal do projeto.

Compile o projeto, clicando em *Processing->Start Compilation*.

Durante o processo de compilação, a janela *Tasks* apresenta o progresso da compilação em cada etapa, no canto inferior direito da janela do Quartus Prime é exibido o tempo total da compilação e a porcentagem geral. Na janela de mensagens (parte inferior da tela) são exibidas diversas mensagens informativas (em verde), *warnings* (em azul) e erros (em vermelho).

A compilação do hardware é um processo mais demorado que a compilação do software, pois é uma tarefa que envolve mais passos e passos mais complexos. Na compilação do hardware, além de fazer a verificação de erros na descrição, também são feitas otimizações de expressões booleanas e dos circuitos. Após isso, é feita a distribuição dos componentes do circuito no chip FPGA assim como a ligação entre eles. Ao fim da compilação, são gerados arquivos que serão usados para programar o FPGA do kit DE10-Lite.

Caso seja encontrado algum erro na compilação, verifique e corrija os erros em seu projeto e recompile.

DICA: Se for exibida alguma mensagem de erro na barra de mensagens, procure a primeira mensagem de erro, leia e tente entender o que ocorreu. Se clicar duas vezes na mensagem de erro, será destacado o ponto onde ocorreu o erro no arquivo de descrição de hardware. Resolva os erros sequencialmente, a partir do primeiro, pois erros subsequentes podem ser desdobramento de erros anteriores.

Após a compilação ser concluída, o Quartus Prime exibe uma janela chamada *Compilation Report* (ver Figura 17). Essa janela apresenta um relatório de compilação com diversas informações, como, por exemplo, a quantidade de elementos lógicos que seu projeto usa do dispositivo escolhido (observe em *Total logic elements* que nosso projeto usa menos de 1% da capacidade do FPGA).

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Apr 25 10:12:29 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	tutorial
Top-level Entity Name	tutorial
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	2 / 49,760 (< 1 %)
Total registers	0
Total pins	5 / 360 (1 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

Figura 17: Compilation Report.

DICA: O Quartus Prime exibe diversos *warnings* durante a compilação. Em geral não há razão para se preocupar com eles. De um modo geral, nessa disciplina, podem ser ignorados.

3.3 Definindo a pinagem do projeto

Ao criar o projeto de acordo com o que foi explicado neste tutorial, a pinagem do projeto será feita automaticamente. A pinagem pode ser conferida em *Assignments->Pin Planner*. Se, por qualquer motivo, a pinagem não estiver preenchida, siga as instruções a seguir para defini-la mais facilmente nos projetos em laboratório. Se já estiver preenchida, não há nada mais a se fazer e esta seção pode ser ignorada.

Uma vez criado o projeto e o circuito, precisamos definir a qual pino físico do FPGA está ligado cada pino do esquemático antes de realizar o *download* do *bitstream* no FPGA. Isso pode ser feito de forma manual, adicionando os pinos um a um, de acordo com o *datasheet* do kit DE10-Lite ou podemos reutilizar os pinos de algum projeto já criado. Por praticidade, iremos optar pela segunda forma. Para isso, vá em *Assignments->Import Assignments...* Será aberta a janela apresentada na Figura 18.

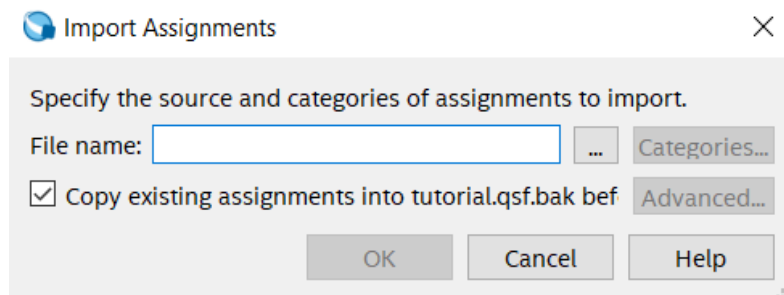


Figura 18: Import Assignments.

Clique nos três pontinhos ao lado direito do campo *File name* e selecione o arquivo no diretório onde estão salvos alguns arquivos importantes para o kit DE10-Lite no laboratório:

C:/intelFPGA_lite/DE10-Lite_v.2.1.0_SystemCD/Demonstrations/Golden_Top/DE10_LITE_Golden_Top.qsf.

Após isso, clique em OK e, pronto, já definimos todos os pinos para a DE10-Lite. Recompile o projeto em *Processing->Start Compilation*.

Verifique no arquivo de diagrama de blocos se o valor dos pinos apareceu ao lado dos nomes dos pinos, conforme a Figura 19. Se sim, é porque está tudo certo e o projeto está pronto para ser testado na placa.

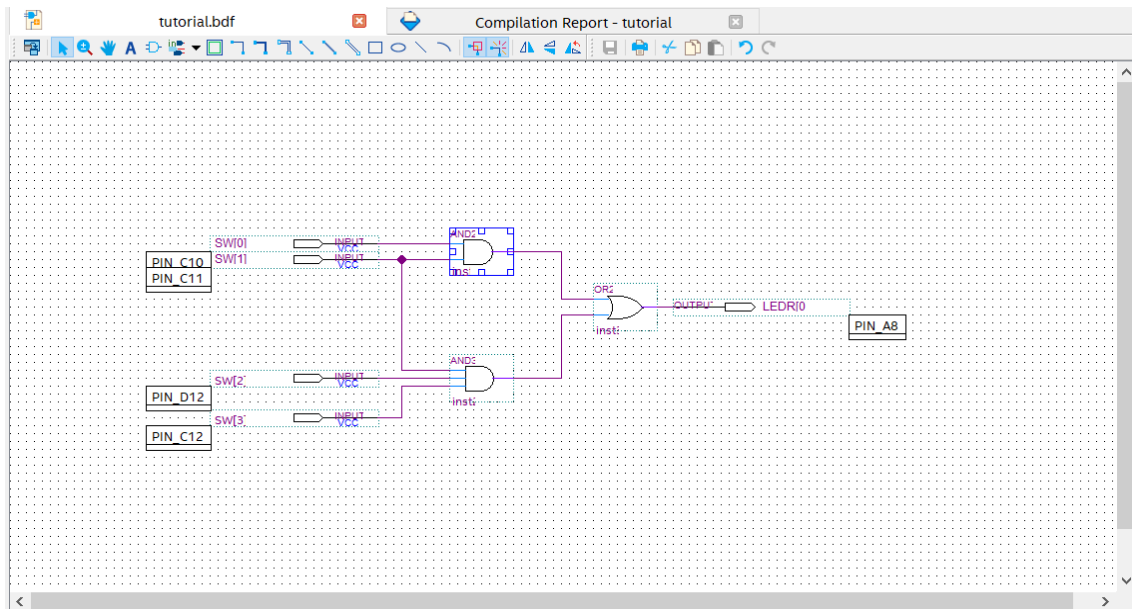


Figura 19: Pinos no diagrama de blocos.

A lista de todos os pinos disponíveis no kit DE10-Lite pode ser vista em *Assignments->Pin Planner*, conforme a Figura 20.

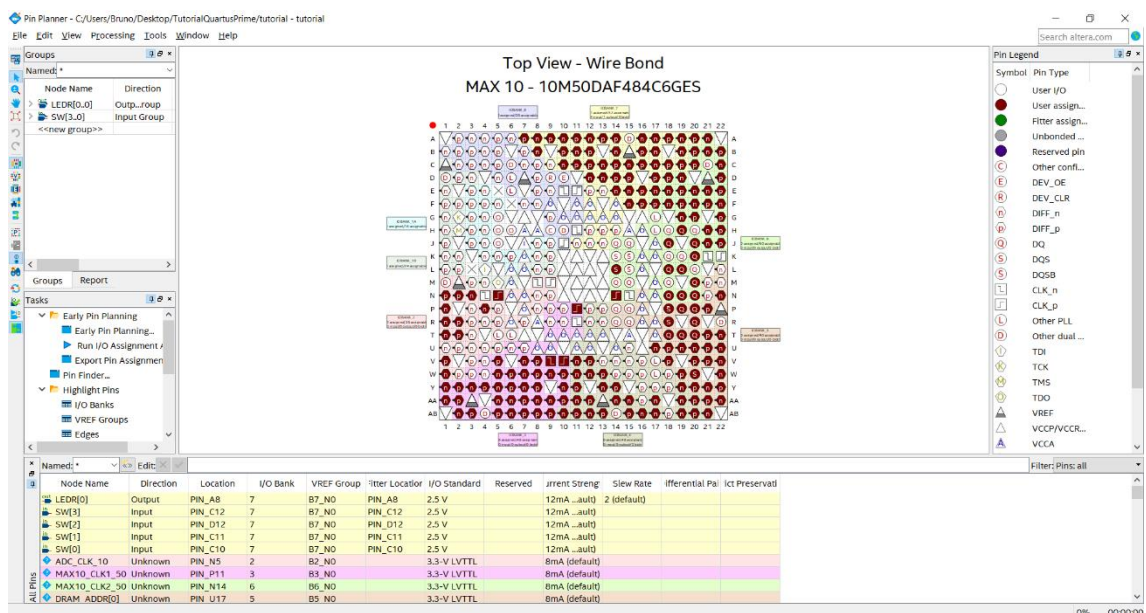


Figura 20: Pin Planner realizado.

Feche a janela do *Pin Planner*, pois não é necessário alterar nada nela, uma vez que já fizemos a importação dos pinos anteriormente.

3.4. Download do *bitstream* no FPGA

Após conectar o kit ao computador por meio do cabo USB, voltemos ao software Quartus Prime. Para realizar o *download* do *bitstream*, vá ao Menu *Tools->Programmer*. A janela exibida na Figura 26 será apresentada.

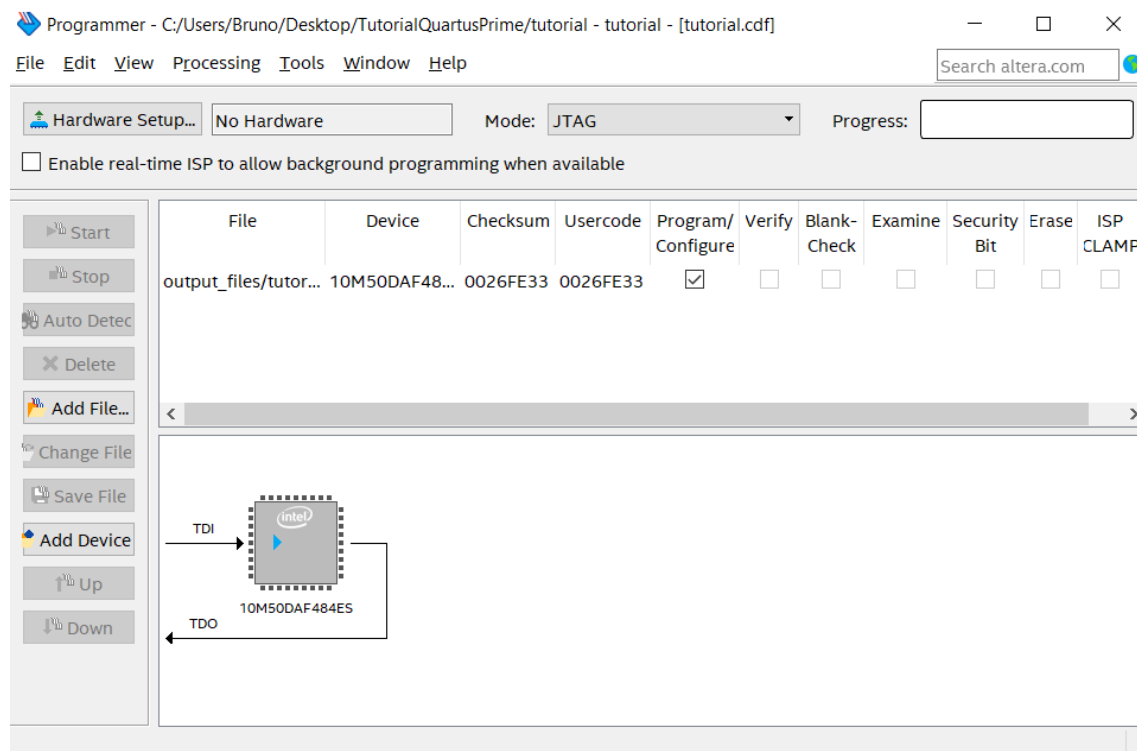


Figura 21: Programmer.

Clique em *Hardware Setup...* e, em *Currently selected hardware*, escolha *USB-Blaster*, clique em *Close* e verifique se apareceu *USB-Blaster* ao lado do botão *Hardware Setup...* Após isso, clique em *Start* e a configuração do kit com o projeto se iniciará. Quando a barra em *Progress* atingir os 100%, o processo estará completo.

Pronto! Agora, o FPGA já está executando o circuito do projeto.

3.5. Verificando o funcionamento do circuito

Para verificar o funcionamento do circuito gravado em FPGA, devemos utilizar as entradas e verificar as saídas definidas no *Pin Planner*. Foram usadas as *switches* de 0 a 3 (verifique numeração impressas nas *switches* do kit DE10-Lite) e o LED vermelho logo acima da SW[0].

Ao usar pinos de entrada e saída, devemos consultar o manual do kit para entender seu funcionamento. De acordo com o manual, uma *switch* é *setada* para nível lógico 1 quando está na posição para cima e *setada* para nível lógico 0 quando está na posição para baixo.

0 quando está na posição para baixo (próximo da borda da placa). O LED, por sua vez, estará iluminado quando o pino de saída respectivo do FPGA estiver em nível lógico 1 e apagado quando estiver em nível lógico 0.

Dito isso, verifique se o funcionamento do circuito em FPGA está de acordo com o esperado. Novamente, para verificar o circuito, deve-se ter em mente que o LED representa a saída S e as *switches* de 0 a 3 representam as entradas A, B, C e D. Todas as combinações de valores para A, B, C e D devem ser testados e a saída S verificada para cada combinação. A tabela-verdade obtida pela execução do circuito deve ser a mesma tabela-verdade gerada pela função booleana **$S = AB + BCD$** .

4. Finalização da atividade

Após realizar todas as tarefas propostas nesta prática-tutorial, apresente o projeto ao Professor, **compacte a pasta contendo todos os arquivos do projeto e envie para a sala virtual da disciplina no Campus Virtual**. Guarde este PDF das práticas e o projeto para consultas futuras.

DICA: Sempre consulte o manual do kit DE10-Lite para entender o funcionamento dos seus componentes.

DICA-MASTER: Sempre que houver dúvidas sobre o uso básico do software Quartus Prime, consulte este PDF!