Vitor Amadeu Souza

FPGA

Aplicações em VHDL
Volume único

© 2023 by Cerne Tecnologia e Treinamento Ltda.

© 2023 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Maio de 2023

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: *Cerne Tecnologia e Treinamento*

E-mail da Empresa: *cerne@cerne-tec.com.br*

Home Page: *www.cerne-tec.com.br.com.br*

Atendimento ao Consumidor: *sac@cerne-tec.com.br*

Contato com o Autor: *vitor@cerne-tec.com.br*



FEITO NO BRASIL

“Melhor é repelir dez verdades do que admitir uma única falsidade.”

Allan Kardec

Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia têm uma linha completa de aprendizado para os microcontroladores da família PIC, 8051, Holtek, dsPIC, ARM e etc. Veja os detalhes de cada um nas figuras abaixo:



Kit Cerne CPLD

- CPLD ALTERA
- Botões
- Leds
- Gravação ICSP
- Display 7 Segmentos
- E muito mais!

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais e etc.

Visite a nossa página na Internet, no endereço **www.cerne-tec.com.br** e conheça melhor nossos serviços e produtos.

Cerne

Conhecimento para o Desenvolvimento

www.cerne-tec.com.br

Sumário

Capítulo I – Introdução ao FPGA.....10

1. A importância do FPGA.....10
2. Conhecendo o FPGA.....11
3. Programando o FPGA.....13
4. Características da família Cyclone IV.....13

Capítulo II – Ferramenta de desenvolvimento.....19

1. Hardware.....19
2. Software.....20
3. Criando um projeto.....21
4. Simulando o projeto.....30
5. Gravando o FPGA.....51

Capítulo III – Exemplos no Schematic.....55

1. Introdução.....55
2. Botões e Leds.....55
3. Botão Invertido.....56
4. Lógica OR.....57
5. Lógica AND.....59
6. Projeto de Alarme I.....60
7. Projeto de Alarme II.....61
8. Display de 7 segmentos.....63
9. Comparador.....64
10. Flip-Flop Tipo D.....65
11. MUX I.....66
12. MUX II.....68
13. DEMUX I.....69
14. Decoder I.....70
15. Decoder II.....72
16. Half-Adder.....73
17. Full-Adder.....74
18. Multiplicação.....77
19. Contador.....78

20. Contador com Código Gray....80
21. Somador full adder 74283....81
22. ULA 74382....83

Capítulo IV – VHDL....87

1. Introdução....87
2. Estrutura básica do VHDL....87
3. Comentários....89
4. Identificadores....89
5. Representação numérica em diversas bases....90
6. Números inteiros e reais....90
7. Representação de strings e caracteres....91
8. Tipos de dados....92
9. Constantes....93
10. Variáveis....94
11. Definindo faixas de valores....94
12. Conversão de tipos....94
13. Operador de atribuição....95
14. Operadores aritméticos....95
15. Operadores relacionais....96
16. Operadores lógicos....96
17. Operadores de deslocamento....97
18. Comando de decisão IF....97
19. Comando de decisão IF-ELSE....98
20. O comando de decisão CASE....99
21. O comando de loop FOR....99
22. O comando de loop WHILE....100

Capítulo V – Exemplos no VHDL....102

1. Introdução....102
2. Ligando um LED....103
3. Botão e Led....104
4. Botões e Leds....105
5. Lógica OR com botões....108
6. Lógica AND com botões....109
7. Lógica XOR com botões....111
8. Pisca-pisca....112
9. Contador binário....114

10. Contador de década....115
11. Display de 7 segmentos....117
12. Comparador de 4 bits....118
13. Lógica AND com Nibble....120
14. Lógica OR com Nibble....121
15. Lógica XOR com Nibble....122
16. Lógica NAND com Nibble....124
17. Lógica NOR com Nibble....125
18. Lógica XNOR com Nibble....126
19. Lógica AND com Byte....127
20. Lógica OR com Byte....129
21. Lógica XOR com Byte....131
22. Lógica NAND com Byte....132
23. Lógica NOR com Byte....134
24. Lógica XNOR com Byte....135
25. Somador com Nibble....137
26. Representação complemento 2....142
27. Subtrator com Nibble....144
28. Unidade Lógica Aritmética (ULA)....148
29. Multiplexador....168
30. MUX/DEMUX digital e analógico....170
31. Multiplexador 74HC157N....170
32. Multiplexador 74HC151N....170
33. Projeto de Multiplexador com 4 entradas....175
34. Projeto de Multiplexador com 8 entradas....178
35. Multiplexador 74HC4067....181
36. MUX/DEMUX CD4051....184
37. MUX/DEMUX CD4052....186
38. MUX/DEMUX CD4053....189
39. Projeto de Demultiplexador com 4 saídas....191
40. Codificador Decimal → BCD....195
41. Codificador Decimal → Stibitz ou excesso 3....198
42. Codificador Decimal → Gray....201
43. Codificador BCD → Gray....203
44. Decodificador 74LS139....207
45. Decodificador 74LS138....211
46. Decodificador CD4028....216
47. Decodificador SN7447....220
48. Flip-flop assíncrono RS com porta NAND....224

49. Flip-flop assíncrono RS com porta NOR....228
50. Flip-flop assíncrono RS com PRESET e RESET....232
51. Flip-flop síncrono RS....237
52. Flip-flop JK....245
53. FF tipo D rising edge....253
54. FF tipo D falling edge....258
55. FF tipo T rising edge....260
56. FF tipo T falling edge....263
57. Latch com EN=1....265
58. Latch com EN=0....269
59. Registrador de deslocamento....272
60. Contador....278
61. Temporizador....287
62. Biestável ou Multivibrador biestável....290
63. Monoestável ou Multivibrador monoestável....291
64. Astável ou Multivibrador astável....293
65. Máquina de estados....294
66. Memória RAM 62256....303
67. Operação de leitura (read cycle)....306
68. Operação de escrita (write cycle)....311

Capítulo VI – Exercícios com respostas....317

Referências....358

Capítulo I
Metodologia de desenvolvimento

A proposta desta obra é reunir todos os volumes da publicação *FPGA Aplicações em VHDL (2014)* do volume I ao IX em um único volume, de modo a apresentar em uma única obra diversos temas fundamentais na programação em VHDL. Assuntos como portas lógicas, flip-flops, contadores, acesso à memória, codificador, decodificador, mux e demux, registrador de deslocamento, máquina de estados, ULA dentre outros são explorados neste livro. Ao término do livro há vários exercícios com respostas para facilitar a aprendizagem nesta ferramenta.

Capítulo I
Introdução ao FPGA

1. A importância do FPGA

Um ASIC (Application Specific Integrated Circuit) é um componente eletrônico com uma funcionalidade específica, como exemplo, um CI projetado especialmente para ser um MODEM 3G ou um relógio digital. A construção destes chips em sua fase de projeto é muito onerosa, pois deve ser feita uma máscara e fabricado poucos chips para se testar a viabilidade do projeto.

Uma das formas de diminuir tal custo é utilizar o FPGA (Field Programmable Gate Array), pois este é um dispositivo eletrônico reconfigurável, ou seja, o projetista tem a facilidade de reconfigurar o hardware e testar o funcionamento de determinada aplicação inúmeras vezes até chegar ao modelo ideal que leve a fabricação do ASIC, diminuindo desta forma drasticamente o custo de desenvolvimento de novos circuitos integrados. Após esta fase, fabricar um lote de ASIC fica com um custo bem menor, fazendo o acesso a novas tecnologias mais rápida se comparado ao processo anterior para os consumidores.

No geral, os equipamentos disponíveis hoje no mercado com uma aplicação específica, como DVDs, TVs, Microcontroladores, Microprocessadores, Smartphone e etc. passam primeiramente por uma

prototipagem no FPGA antes de atingirem o modelo que leve a fabricação em série, a ponto de se tornar um ASIC.

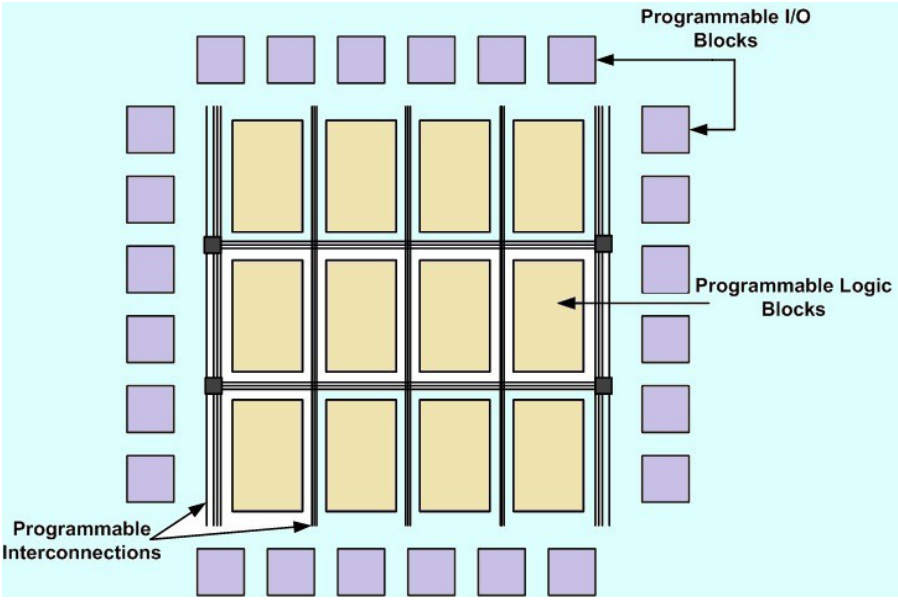
2. Conhecendo o FPGA

Como dito, o FPGA é um hardware reconfigurável, ou seja, o mesmo vem equipado com blocos lógicos internos que se ajustam de acordo com a necessidade específica de determinado projeto. Há inúmeros fabricantes de FPGA, como os citados abaixo.

Fabricante	Site
Xilinx	www.xilinx.com
ALTERA	www.altera.com
Actel	www.actel.com
ATMEL	www.atmel.com
Achronix	www.achronix.com

O primeiro FPGA foi lançado no mercado em 1985 pela Xilinx. Nesta literatura, será utilizado um FPGA modelo EP4CE22E22C6 da família Cyclone IV da ALTERA. Tal família, está disponível em duas sub-famílias, sendo estas as famílias Cyclone IV E e Cyclone IV GX. O modelo adotado, é integranda da família E, que apresenta como características principais o fato de ser de baixa potência, baixo custo e possuir alta funcionalidade.

Internamente, um FPGA é composto de três blocos, conforme verificado na figura a seguir.



Fonte: http://vedikapatki.wordpress.com/2012/05/01/fpga-basics-and-architecture/

O *Programmable I/O Block ou* IOB (Input/Output Block) fica responsável pelo acesso aos pinos de I/O (Input/Output) do FPGA.

O *Programmable Logic Block ou* CLB (Configuration Logical Block) é construído através de portas lógicas como flip-flops, em que desta forma é possível configurar de que forma o CLB irá funcionar.

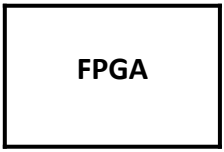
O *Programmable Interconnections* (Switch Matrix) faz a conexão entre a CLB e a IOB, conectando os pinos de I/O as funções lógicas definidas na CLB.

3. Programando o FPGA

No mercado, encontram-se normalmente os FPGAs do tipo SRAM, ou seja, a configuração feita no CLB, IOB e Switch Matrix são perdidos no momento em que o dispositivo é desligado, sendo necessário refazer tal configuração. Isso é alcançado conectando-se uma memória EEPROM ao FPGA, de modo a este se reconfigurar no momento de seu start-up, conforme sugere a figura a seguir.



No entanto, já se encontra no mercado FPGAs com EEPROM interna, o que elimina a necessidade da EEPROM externa, diminuindo assim a quantidade de circuitos na placa eletrônica. Neste caso, a configuração ficaria apenas com o FPGA, como sugere a próxima figura.



4. Características da família Cyclone IV

As principais características da família Cyclone IV estão listadas na próxima tabela.

Característica
Disponível em versões com 6k a 150k de elementos lógicos
Até 6,3 Mb de memória
Até 360 máquinas de multiplicação de 18x18 bits

Fonte: Cyclone IV Device Handbook

Os modelos integrantes da família Cyclone IV E estão apresentados na próxima tabela. Observe que o modelo EP4CE22E22C6 faz parte da coluna EP4CE22, onde observa-se que este possui 22k de elementos lógicos, 594 kb de memória embutida, 66 máquinas de multiplicação e até 153 pinos de I/O (Input/Output).

Resources	EP4CE6	EP4CE10	EP4CE15	EP4CE22	EP4CE30	EP4CE40	EP4CE55	EP4CE75	EP4CE115
Logic elements (LEs)	6,272	10,320	15,408	22,320	28,848	39,600	55,856	75,408	114,480
Embedded memory (Kbits)	270	414	504	594	594	1,134	2,340	2,745	3,888
Embedded 18 × 18 multipliers	15	23	56	66	66	116	154	200	266
General-purpose PLLs	2	2	4	4	4	4	4	4	4
Global Clock Networks	10	10	20	20	20	20	20	20	20
User I/O Banks	8	8	8	8	8	8	8	8	8
Maximum user I/O (1)	179	179	343	153	532	532	374	426	528

Fonte: Cyclone IV Device Handbook

A família Cyclone IV E está disponível para aplicações em diversas áreas, podendo ser identificado através da nomenclatura do FPGA o tipo de aplicação, conforme ilustra a próxima tabela.

Identificação	Aplicação
C	Comercial
I	Industrial
A	Automotivo

A principal diferença está ligada a temperatura de junção, em que a comercial opera na faixa de 0 ° a 85° C, a industrial de -40 ° a 100° C e a automotiva de -40 ° a 125° C. A velocidade é definida através do último caractere, em que -6 indica a velocidade mais rápida e -8 a mais lenta.

A próxima tabela mostra a velocidade máxima obtida na saída do PLL de acordo com o índice de velocidade do FPGA. Note que quanto menor o índice, maior a velocidade de clock obtida.

f_{OUT} (to global clock)	PLL output frequency (–6 speed grade)	—	—	472.5	MHz
	PLL output frequency (–7 speed grade)	—	—	450	MHz
	PLL output frequency (–8 speed grade)	—	—	402.5	MHz
	PLL output frequency (–8L speed grade)	—	—	362	MHz
	PLL output frequency (–9L speed grade)	—	—	265	MHz

Fonte: Cyclone IV Device Datasheet pg 24.

A alimentação do núcleo (core) varia de acordo com o modelo. Os modelos em que há a presença do caracter “L” a alimentação do núcleo é de 1,0 V enquanto que nos outros modelos é de 1,2 V. A próxima tabela mostra a faixa de tensão suportada pelo FPGA para alimentar o seu core (V_{CCINT}).

V_{CCINT} (3)	Supply voltage for internal logic, 1.2-V operation	—	1.15	1.2	1.25	V
	Supply voltage for internal logic, 1.0-V operation	—	0.97	1.0	1.03	V

Fonte: Cyclone IV Device Datasheet pg 4.

A alimentação do bloco de I/O (IOB) é diferente do núcleo, já que esta faz a interface com o mundo externo. Há diversas faixas disponíveis, como 3,3V, 3,0V, 2,5V, 1,8V, 1,5V e 1,2V. A próxima tabela mostra a faixa de tensão aceitável de acordo com o modelo utilizado.

V_{CCIO} (3), (4)	Supply voltage for output buffers, 3.3-V operation	—	3.135	3.3	3.465	V
	Supply voltage for output buffers, 3.0-V operation	—	2.85	3	3.15	V
	Supply voltage for output buffers, 2.5-V operation	—	2.375	2.5	2.625	V
	Supply voltage for output buffers, 1.8-V operation	—	1.71	1.8	1.89	V
	Supply voltage for output buffers, 1.5-V operation	—	1.425	1.5	1.575	V
	Supply voltage for output buffers, 1.2-V operation	—	1.14	1.2	1.26	V

Fonte: Cyclone IV Device Datasheet pg 4.

O bloco de PLL (Phase Locked Loop) permite elevar a velocidade de clock e este FPGA possui este recurso interno. A parte analógica é alimentada através de 2,5V (V_{CCA}) e parte digital(V_{CCD_PLL}) será a mesma da tensão de núcleo. A próxima tabela mostra a faixa de tensão de acordo com o modelo utilizado.

V_{CCA} (3)	Supply (analog) voltage for PLL regulator	—	2.375	2.5	2.625	V
V_{CCD_PLL} (3)	Supply (digital) voltage for PLL, 1.2-V operation	—	1.15	1.2	1.25	V
	Supply (digital) voltage for PLL, 1.0-V operation	—	0.97	1.0	1.03	V

Fonte: Cyclone IV Device Datasheet pg 4.

Um pino configurado como entrada (V_I), pode suportar tensões na faixa de 0 a 3,6V e enquanto saída impor tensões de 0 a V_{CCIO}, que é a alimentação do bloco de I/O. Acompanhe a tabela a seguir.

V_I	Input voltage	—	–0.5	—	3.6	V
V_O	Output voltage	—	0	—	V_{CCIO}	V

Fonte: Cyclone IV Device Datasheet pg 4.

Os pinos configurados como entrada podem ter resistores de pull-up (ligado a V_{CCIO}) ou pull-down (ligado a GND) habilitados. O valor tipico deste resistor varia de acordo com a tensão de V_{CCIO}, conforme tabela abaixo.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$R_{_PU}$	Value of the I/O pin pull-up resistor before and during configuration, as well as user mode if you enable the programmable pull-up resistor option	V_{CCIO} = 3.3 V ± 5% (2), (3)	7	25	41	kΩ
		V_{CCIO} = 3.0 V ± 5% (2), (3)	7	28	47	kΩ
		V_{CCIO} = 2.5 V ± 5% (2), (3)	8	35	61	kΩ
		V_{CCIO} = 1.8 V ± 5% (2), (3)	10	57	108	kΩ
		V_{CCIO} = 1.5 V ± 5% (2), (3)	13	82	163	kΩ
		V_{CCIO} = 1.2 V ± 5% (2), (3)	19	143	351	kΩ
$R_{_PD}$	Value of the I/O pin pull-down resistor before and during configuration	V_{CCIO} = 3.3 V ± 5% (4)	6	19	30	kΩ
		V_{CCIO} = 3.0 V ± 5% (4)	6	22	36	kΩ
		V_{CCIO} = 2.5 V ± 5% (4)	6	25	43	kΩ
		V_{CCIO} = 1.8 V ± 5% (4)	7	35	71	kΩ
		V_{CCIO} = 1.5 V ± 5% (4)	8	50	112	kΩ

Fonte: Cyclone IV Device Datasheet pg 11.

Esta família de FPGA fornece blocos de multiplicação internos, que são fundamentais em aplicações que necessitem de algum tipo de processamento de sinal como FFT (Fast Fourier Transformer) ou DFT (Discrete Fourier Transformer). Ao todo, há 1 bloco de multiplicação de 9x9 bits e 1 bloco de 18x18 bits, conforme apresentado na tabela a seguir.

Mode	Resources Used	Performance					Unit
	Number of Multipliers	C6	C7, I7, A7	C8	C8L, I8L	C9L	
9 × 9-bit multiplier	1	340	300	260	240	175	MHz
18 × 18-bit multiplier	1	287	250	200	185	135	MHz

Fonte: Cyclone IV Device Datasheet pg 26.

A reconfiguração do FPGA é feita através de uma interface JTAG, onde através de um programador que pode estar conectado a porta USB ou Paralela permite transferir a nova configuração ao FPGA através das linhas TCLK, TMS, TDI e TDO.

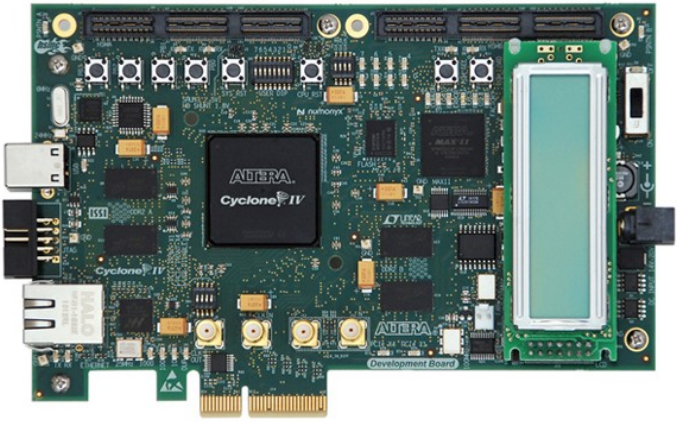
Capítulo II
Ferramenta de desenvolvimento

1. Hardware

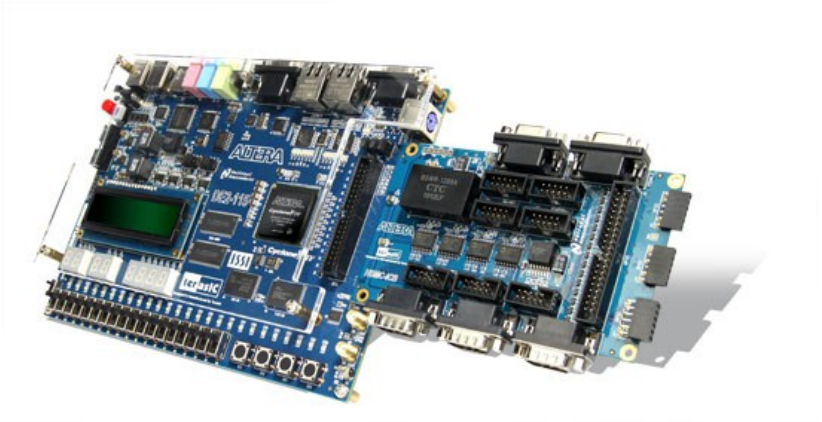
No mercado eletrônico, encontram-se disponíveis diversos kits que possibilitam testar de maneira prática um projeto feito para o FPGA ALTERA, conforme mostra os links a seguir.



Fonte: http://www.altera.com/products/devkits/altera/kit-cyclone-iv-starter.html



Fonte: http://www.altera.com/products/devkits/altera/kit-cyclone-iv-gx.html



Fonte: http://www.altera.com/products/devkits/partners/kit-terasic-ink-industrial-networking-kit.html

Nesta literatura, todos os exemplos serão simulados de forma a apresentar ao leitor o funcionamento do experimento sem depender do kit. Todavia, ao transferir o projeto de hardware a um kit de desenvolvimento é esperado que o resultado seja o mesmo do que foi simulado através do programa.

2. Software

Para o desenvolvimento a nível de software para o FPGA, será utilizado o Quartus II, fornecido pela ALTERA, o mesmo fabricante do Cyclone IV. O Quartus II será utilizado para realizar a programação, seja através de VHDL ou Schematic e o QSIM para simulações. O Quartus II pode ser baixado através do link a seguir.

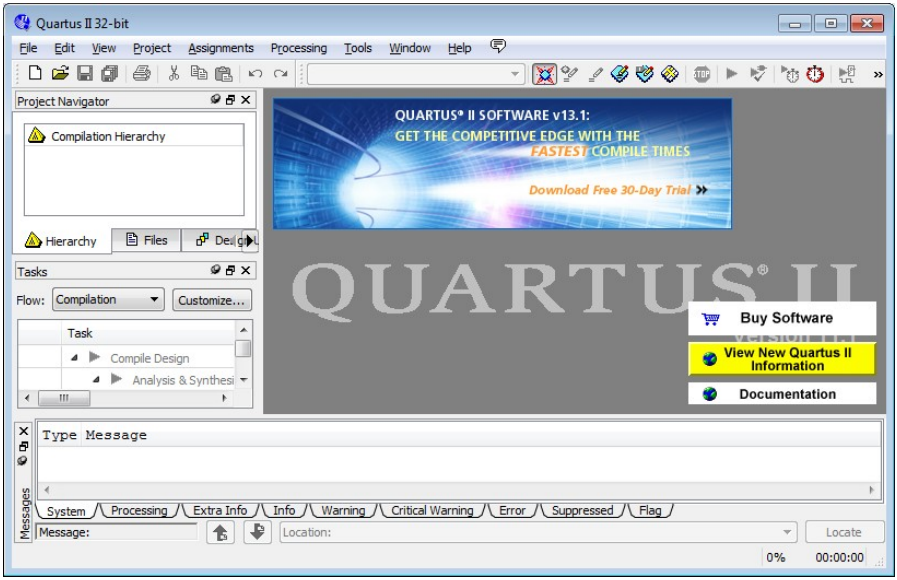
https://www.altera.com/download/sw/dnl-sw-index.jsp

Obs: O QSIM é instalado no mesmo pacote do Quartus II

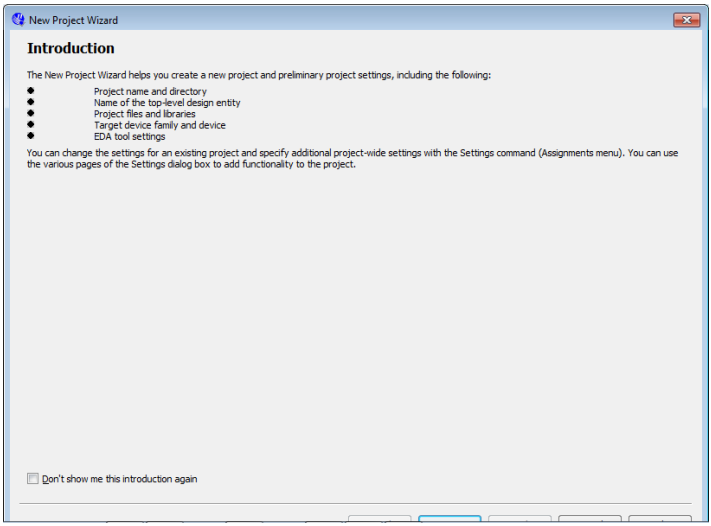
A versão utilizada foi a Quartus II Web Edition, que é uma versão sem licença disponibilizada pelo fabricante. Nesta literatura, a versão 11.1 foi a adotada, mas nada impede que versões mais recentes sejam utilizadas.

3. Criando um projeto

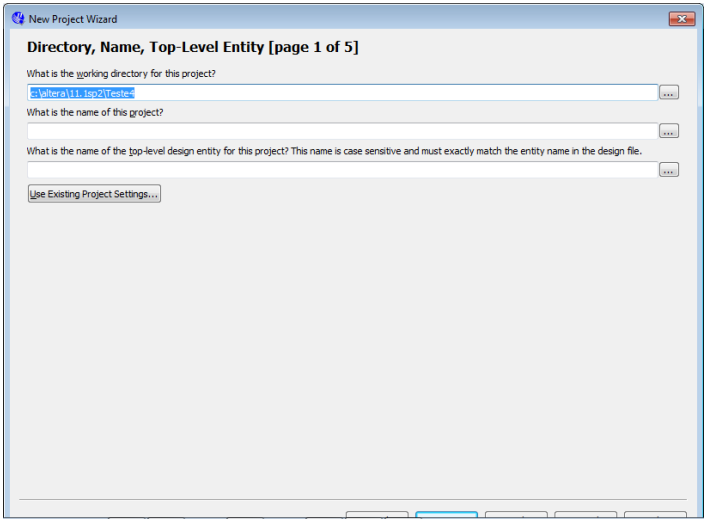
Após o download e instalação dos softwares do tópico passado, inicialize o Quartus II, a seguinte tela será apresentada.



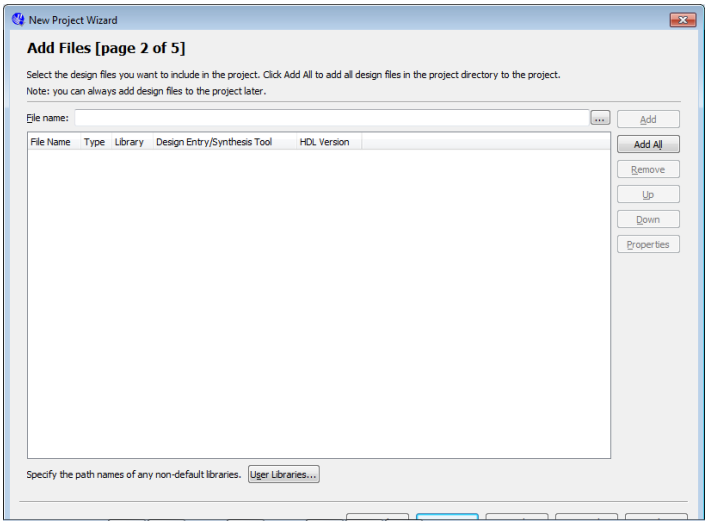
Para criar um novo projeto, vá ao menu File -> New Project Wizard. Neste instante, a seguinte tela será apresentada.



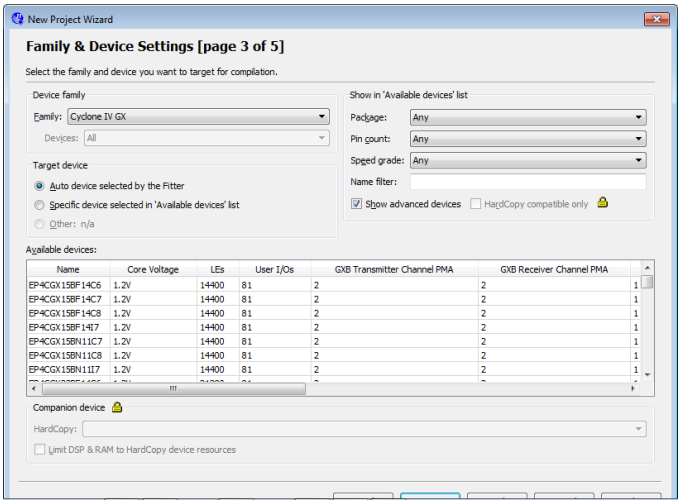
Esta é a tela inicial, que relata todos os passos para criação do projeto. Clique no botão Next, a tela a seguir será carregada.



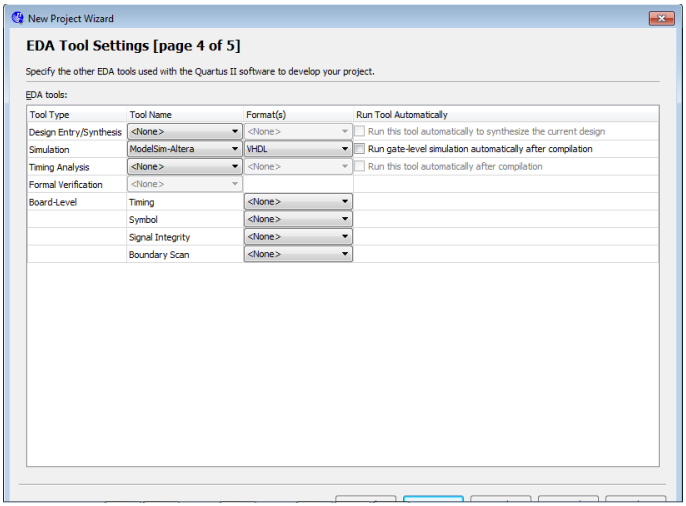
Nesta janela, é solicitado ao usuário o caminho em que o projeto será salvo através da primeira caixa de texto. A segunda caixa permite fornecer um nome ao projeto, como sugestão *exemplo1*. Passada esta etapa, clique no botão Next. A próxima tela será mostrada.



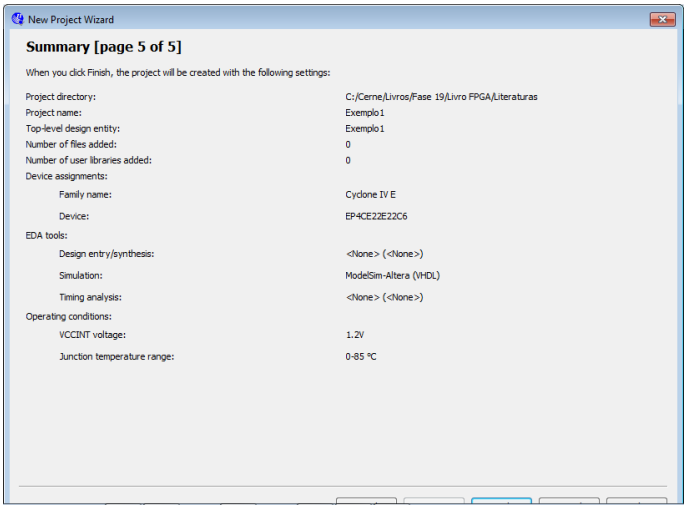
Esta janela permite adicionar ao projeto arquivos no qual o desenvolvedor já disponha. Como não é este o caso neste momento, clique no botão Next, no qual a próxima janela permitirá selecionar o modelo de FPGA que será utilizado no decorrer do projeto.



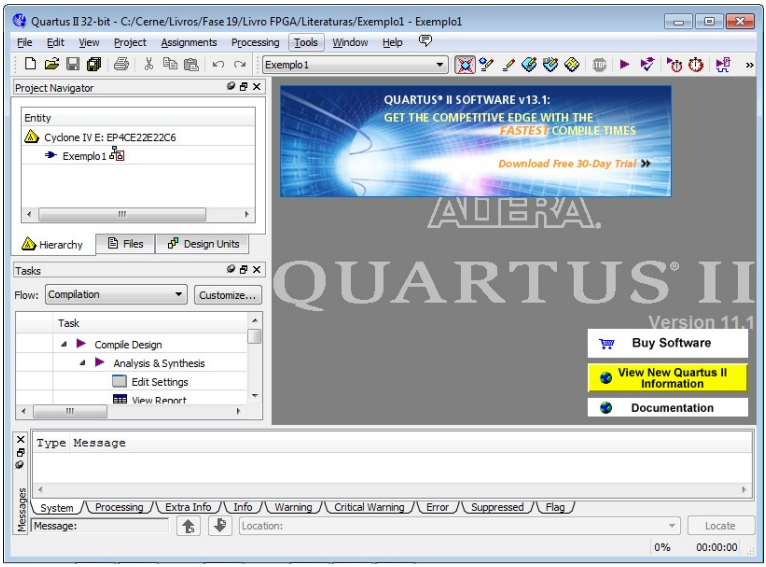
Na opção *Family*, escolha a opção Cyclone IV E. Em *Available devices* selecione o modelo EP4CE22E22C6. Em seguida, pressione o botão Next, em que a próxima janela será apresentada, permitindo escolher a ferramenta utilizada no desenvolvimento do projeto.



Por enquanto, mantenha a opção default e clique no botão Next. A tela a seguir mostrará um resumo do projeto recém criado, onde o modelo, tensão do núcleo, temperatura, caminho do projeto, nome do projeto dentre outras informações são apresentadas.

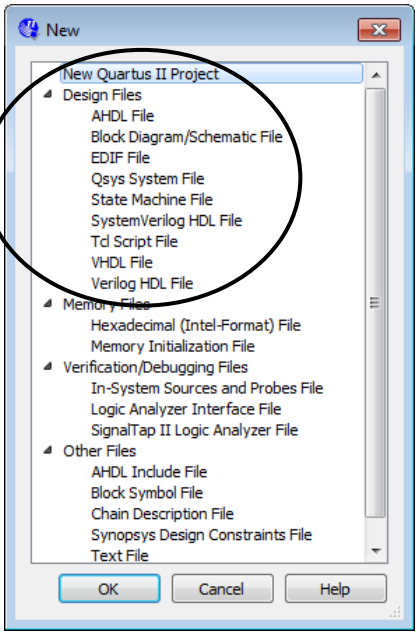


Basta clicar no botão Finish para a criação do projeto seja finalizada e encerrada. Neste instante, a tela apresentada ficará conforme figura abaixo.

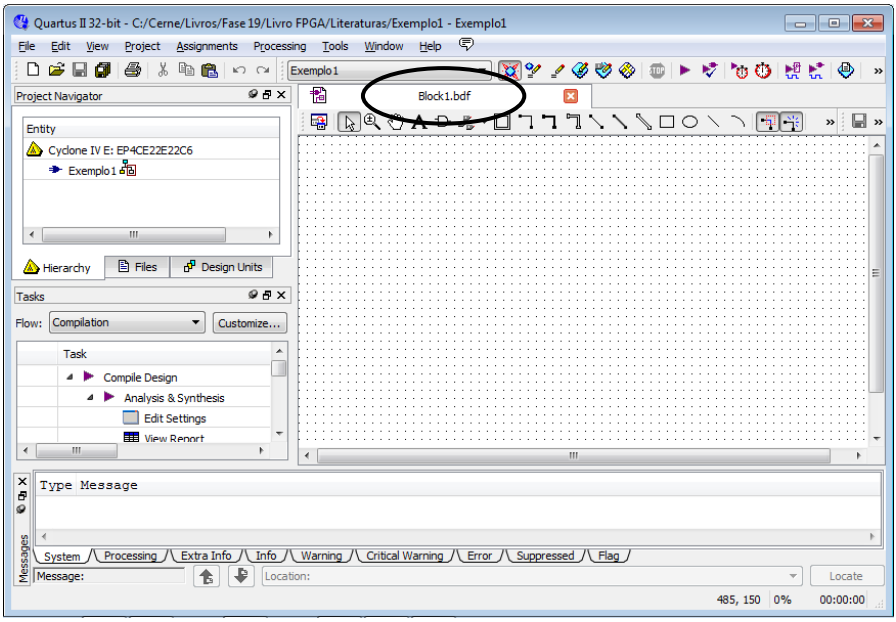


O desenvolvimento de projetos no FPGA pode ser feito de diversas formas, porém nesta literatura será utilizada o modo gráfico através de esquemas elétricos.

Para criar um novo arquivo, vá ao menu *File->New*. A seguinte tela será carregada, onde um arquivo esquemático (Block Diagram/Schematic File) ou VHDL poderão ser criados, conforme marcado na próxima figura.

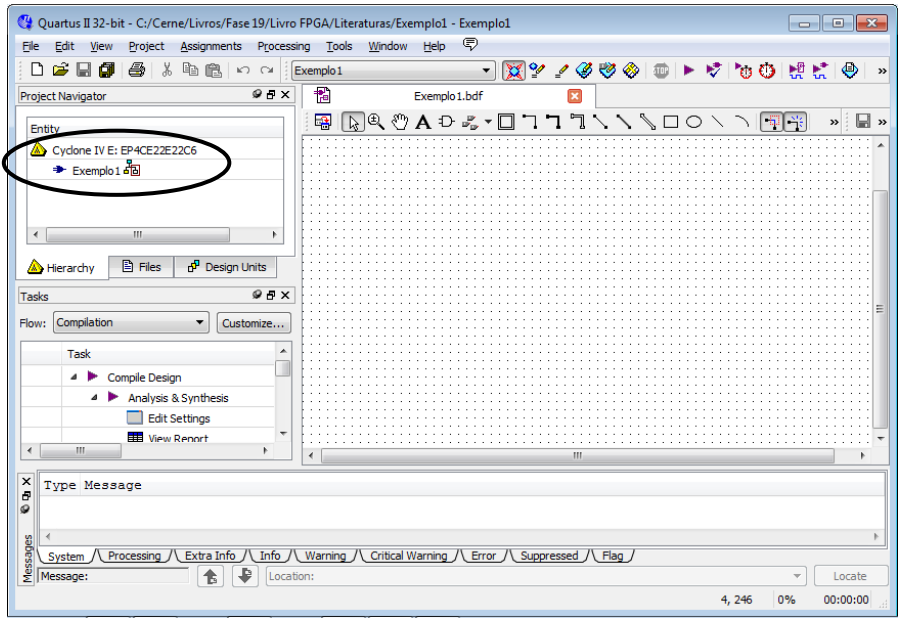


Escolha a opção *Block Diagram/Schematic File*, já que esta será a utilizada no decorrer do próximo capítulo. Em seguida, pressione o botão *Ok*. A tela do Quartus II ficará conforme figura a seguir.

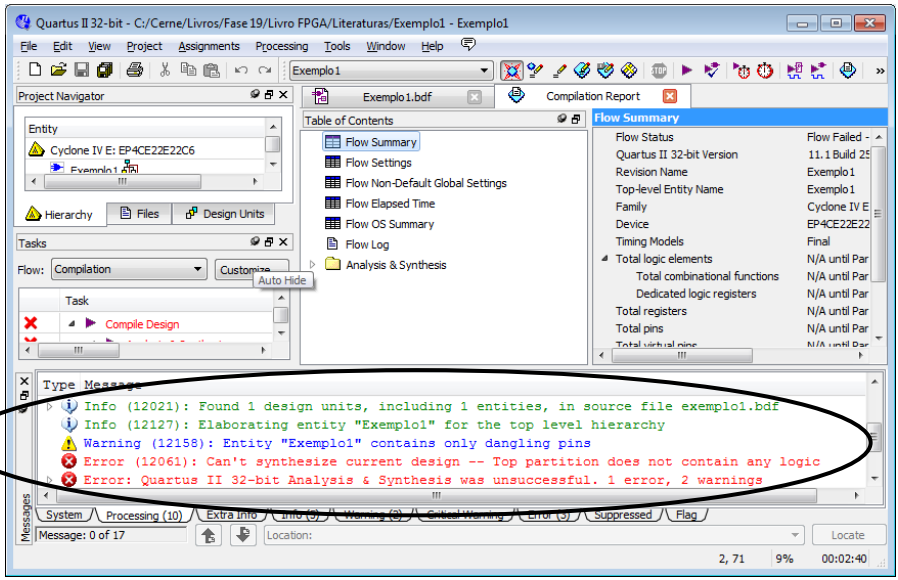


Note que foi criado um arquivo de diagramação, onde poderá ser descrita a lógica de funcionamento do FPGA através de esquemas elétricos. A extensão do arquivo é do tipo .bdf e para salvar e associar este arquivo ao projeto, vá ao menu *File -> Save As*.

Observe na próxima figura que o arquivo foi salvo e já está associado ao projeto.



Após o desenho do esquema elétrico, o próximo passo consiste em compilar, ou seja, montar o roteamento interno do FPGA através dos blocos IOB, CLB e Switch Matrix. Isso será feito indo ao menu *Processing -> Start Compilation*. A próxima janela mostra o resultado da compilação. Observe que há erros, já que nenhuma lógica foi implementada na tela do esquemático.

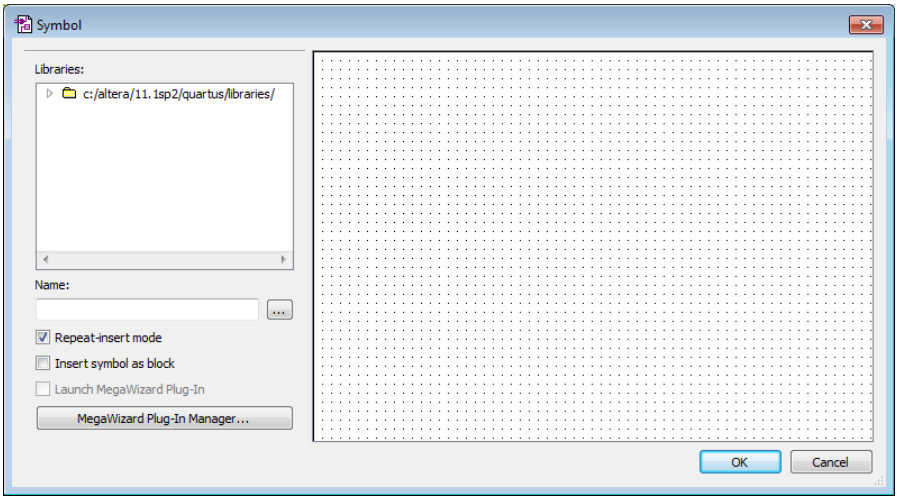


4. Simulando o projeto

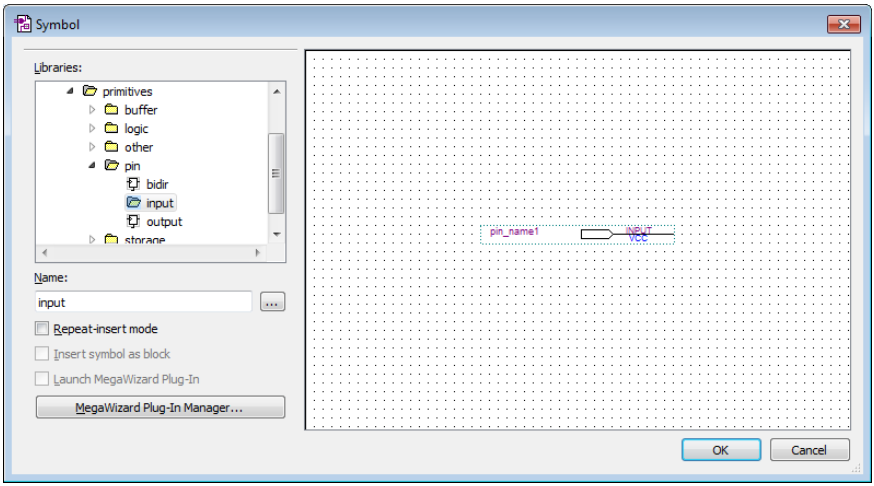
Partindo do ponto anterior, será desenvolvido no Quartus II um primeiro projeto prático no Schematic. Este primeiro projeto elucidativo, irá mostrar o estado de um botão através de um led, para isso, será utilizado apenas uma entrada e uma saída. Clique duas vezes na área pontilhada do *Graphic Editor* ou clique na figura abaixo, disponível na Toolbar do editor gráfico.



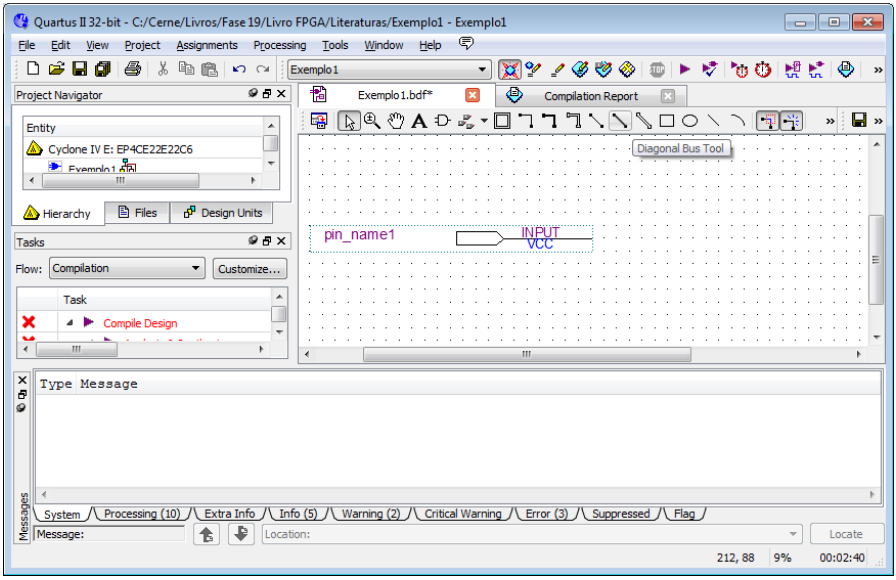
Neste momento, a seguinte tela será carregada.



Na caixa *Name* digite *input* ou clique duas vezes no caminho apresentado em Libraries (c:/altera/...) e selecione o componente desejado na lista, através da opção primitives -> pin -> input. A tela ficará da forma mostrada abaixo.

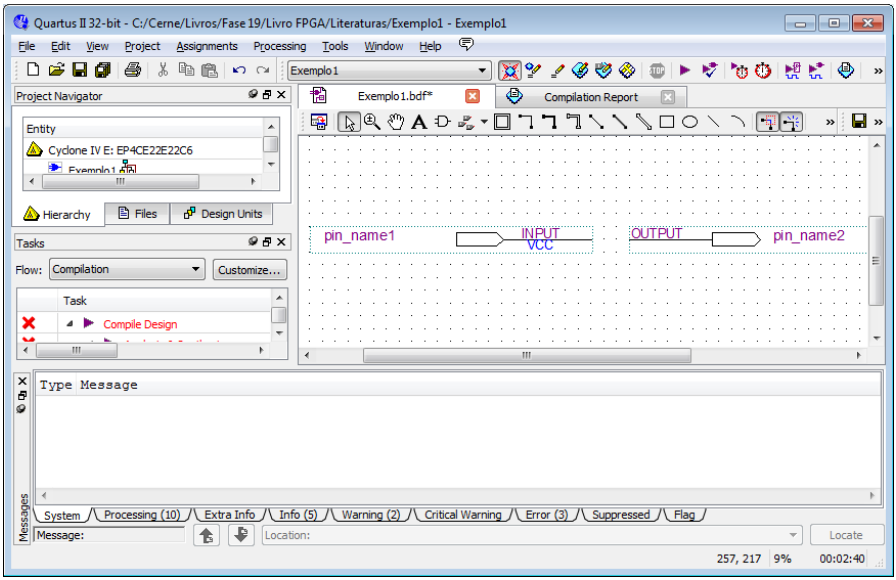


Clique no botão Ok e posicione a entidade *input* na tela do editor gráfico. O resultado ficará conforme a próxima figura.

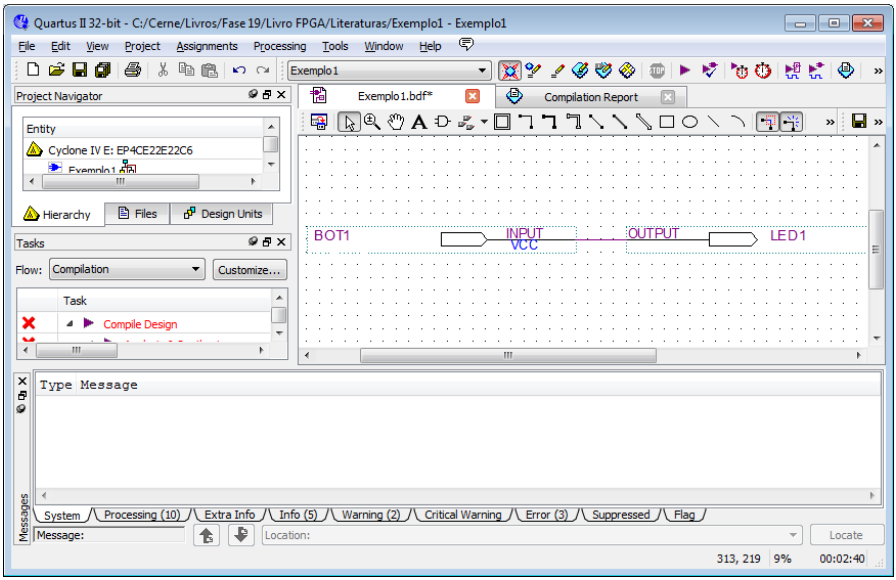


Há diversas bibliotecas disponíveis, sendo algumas delas a primitive (primitiva) onde se encontram vários componentes primitivos como portas lógicas (logic), mega função (megafunctions) com funções de grande complexidade (multiplicação, divisão, etc) dentre outros. Nesta experiência, será utilizado as entradas e saídas da biblioteca primitive.

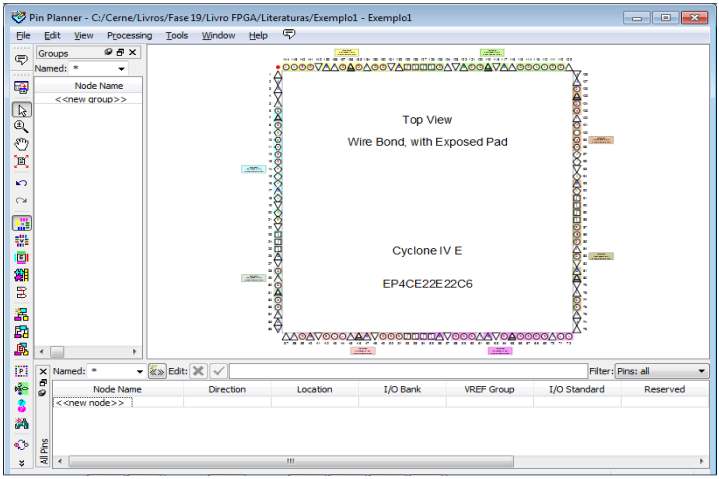
Adicione também a saída, digitando no campo Name o símbolo *output* ou procurando a mesma na biblioteca primitiva. O resultado ficará conforme figura a seguir.



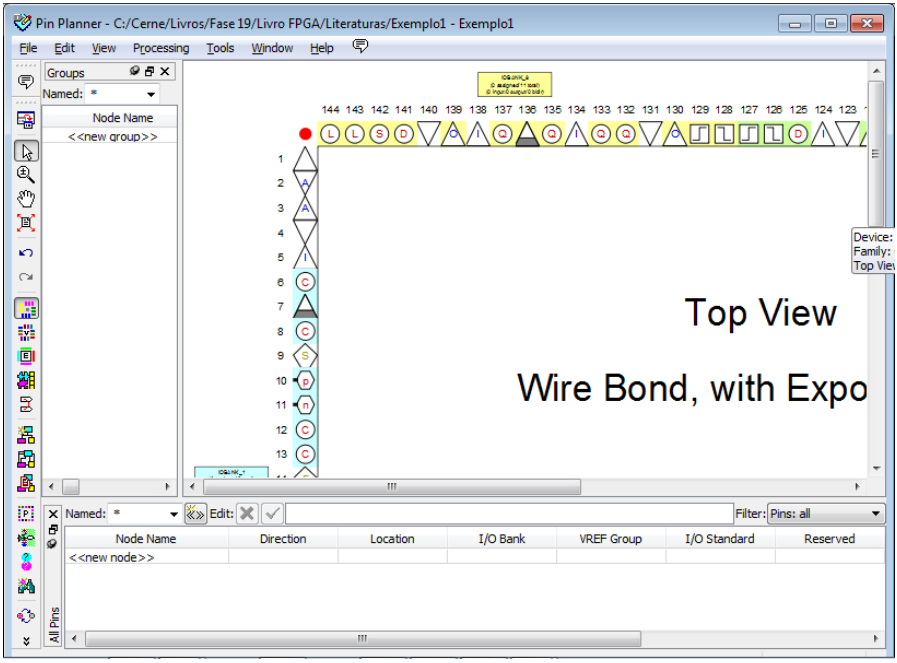
Neste ponto, será necessário interligar a entrada na saída e nomear estes acessos. A entrada será chamada de BOT1 e a saída de LED1. Para interligar as entidades, basta ir com o mouse nestes terminais, pressionar e arrastar, de modo que seja feita esta ligação. Para renomear os terminais basta clicar duas vezes sobre o mesmo e digitar o novo nome. O aspecto final deve ser o seguinte.



Nesta etapa, é possível configurar em qual pino o botão (BOT1) e led (LED1) estão conectados. Para isso, vá ao menu *Assignments -> Pin Planner*. A seguinte tela será apresentada.



Todos os pinos disponíveis no FPGA são mostrados, em que desta forma pode ser feita a configuração em qual deles o botão e led ficarão conectados. Pressionando a tecla CTRL+SPACE, é possível obter um zoom e observar com mais detalhes os pinos, conforme mostra a figura a seguir.



Os pinos marcados através de um círculo são de I/O. Ao todo, este FPGA possui 8 bancos de I/O (I/O BANKs), conforme ilustrado na figura a seguir.