

# Criando Blocos Lógicos com Dispositivos de Lógica Programável



**THE ALL-IN-ONE PLD.**

## **Objetivos:**

- Adquirir conhecimentos em dispositivos de lógica programável;
- Familiarizar-se com o uso do software Quartus II;
- Aprender a criar blocos lógicos através de elementos básicos;
- Simular e testar através do kit da intel o bloco lógico criado.

### Dispositivo de lógica programável:

Um dispositivo lógico programável, DLP ou inglês *Programmable logic device* – *PLD*, é um componente eletrônico utilizado para construção de circuitos digitais.

Ao contrário de uma porta lógica, que tem uma função fixa, um PLD tem uma função indefinida na sua fabricação. Antes de se utilizar um PLD num circuito, este deve ser programado.

Hoje nesses tipos de dispositivos se destacam-se:

Os CPLDs que são os *Complex PLDs*, que permitem a construção de lógicas mais complexas do que simples PLDs.

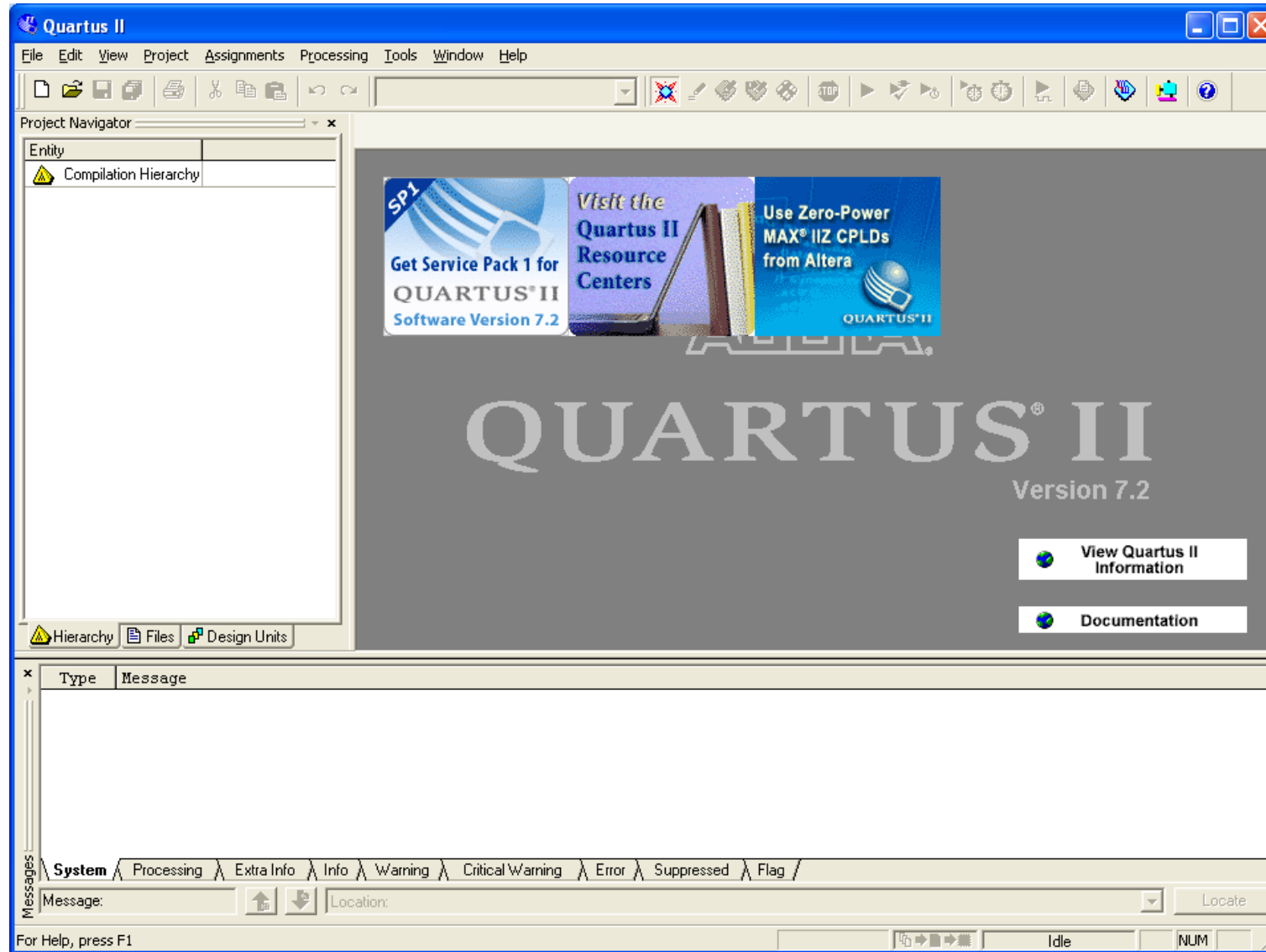
E os *FPGAs (field programmable gate array)*.

Basicamente a diferença entre eles é apenas construtiva.

Possuem recursos similares em ambos os sistemas.

FPGA's e CPLD's oferecem boas escolhas para tarefas particulares. Algumas vezes a decisão é mais econômica do que técnica, ou pode depender da experiência ou preferência pessoal do projetista.

Para o estudos dos circuitos lógicos usaremos o programa Quartus II da ALTERA.

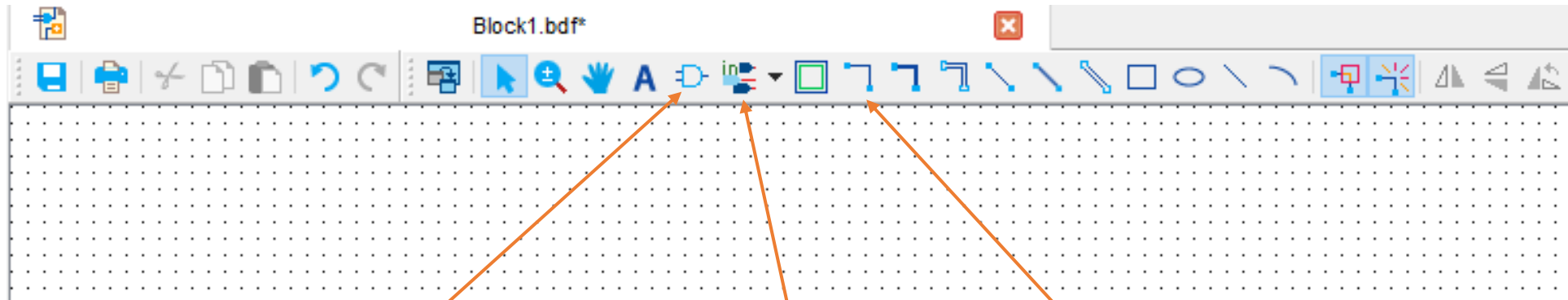


## Etapas para criação do projeto

- File -> New Project Wizard -> Next.
- Selecionar diretório e indicar o nome do projeto.
- Next -> (selecionar empty project) Next -> Next.
- Selecionar família Cyclone IV GX, dispositivo EP4CGX150DF31C7.
- Next -> Next -> Finish.

# Etapas para criação do projeto

- File -> New -> Block Diagram / Schematic File.



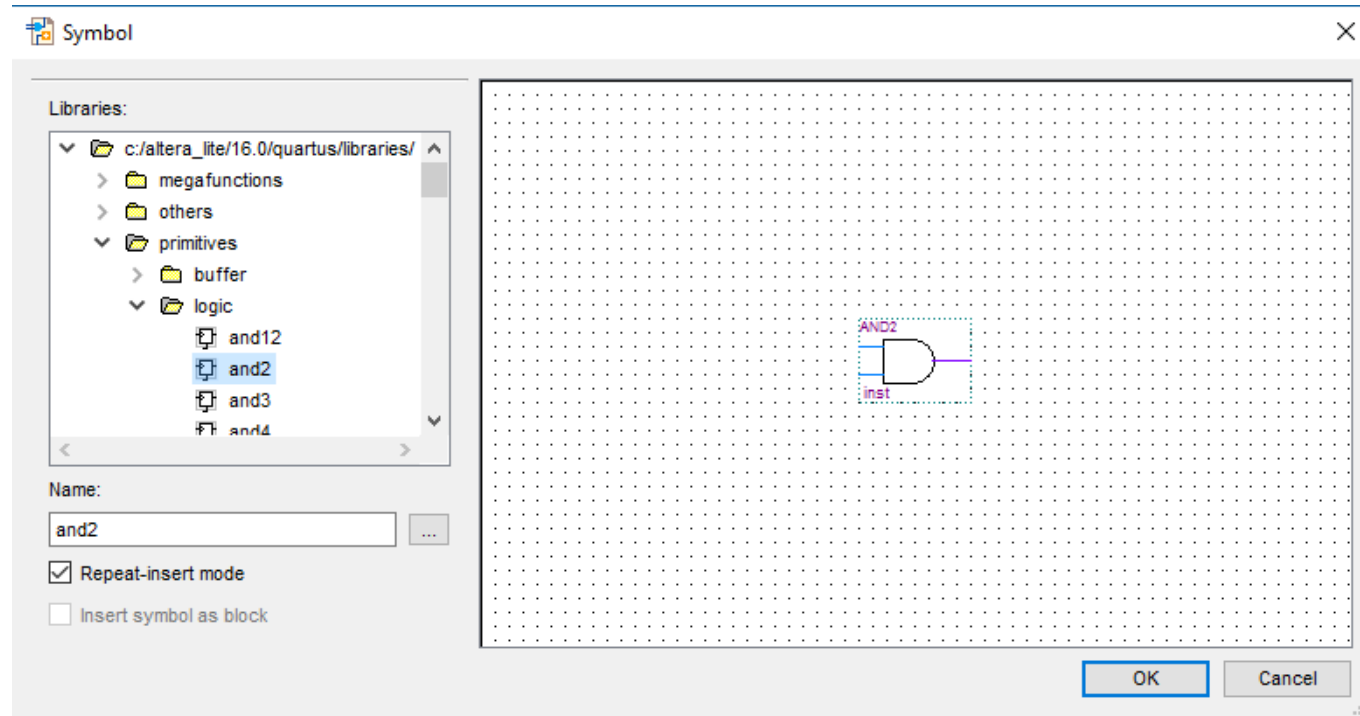
Portas lógicas, elementos de memória, pinos, IP core, etc.

Pinos de entrada e saída

Ferramentas de conexão

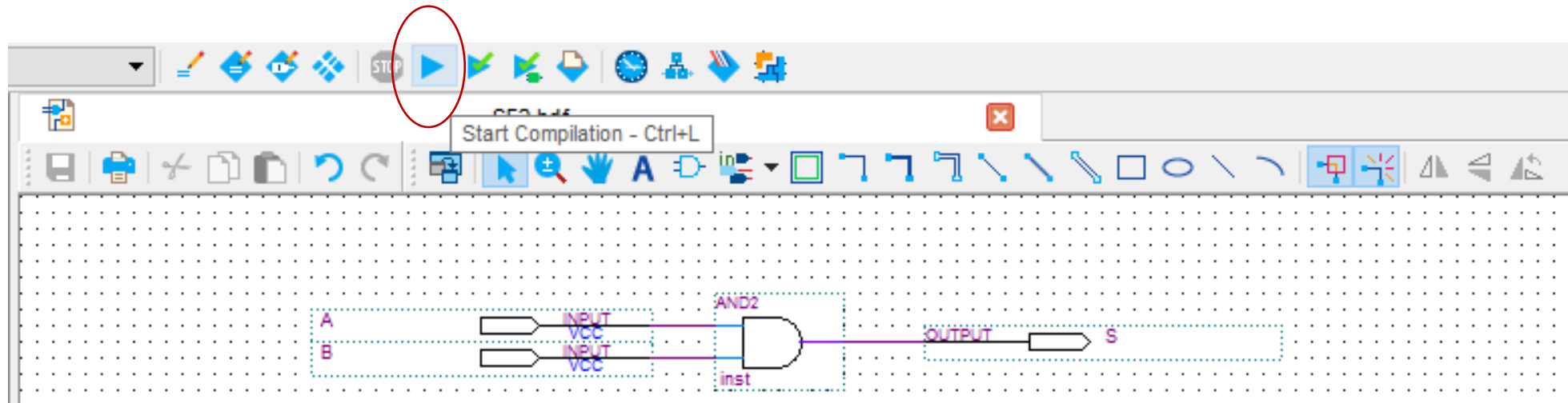
# Etapas para criação do projeto

- No diretório primitives->logic são listadas as portas lógicas.
- Em primitives->pin são definidos os pinos de I/O.



## Criar um bloco de 1 célula de memória.

- Salvar o design e iniciar o processo de compilação.

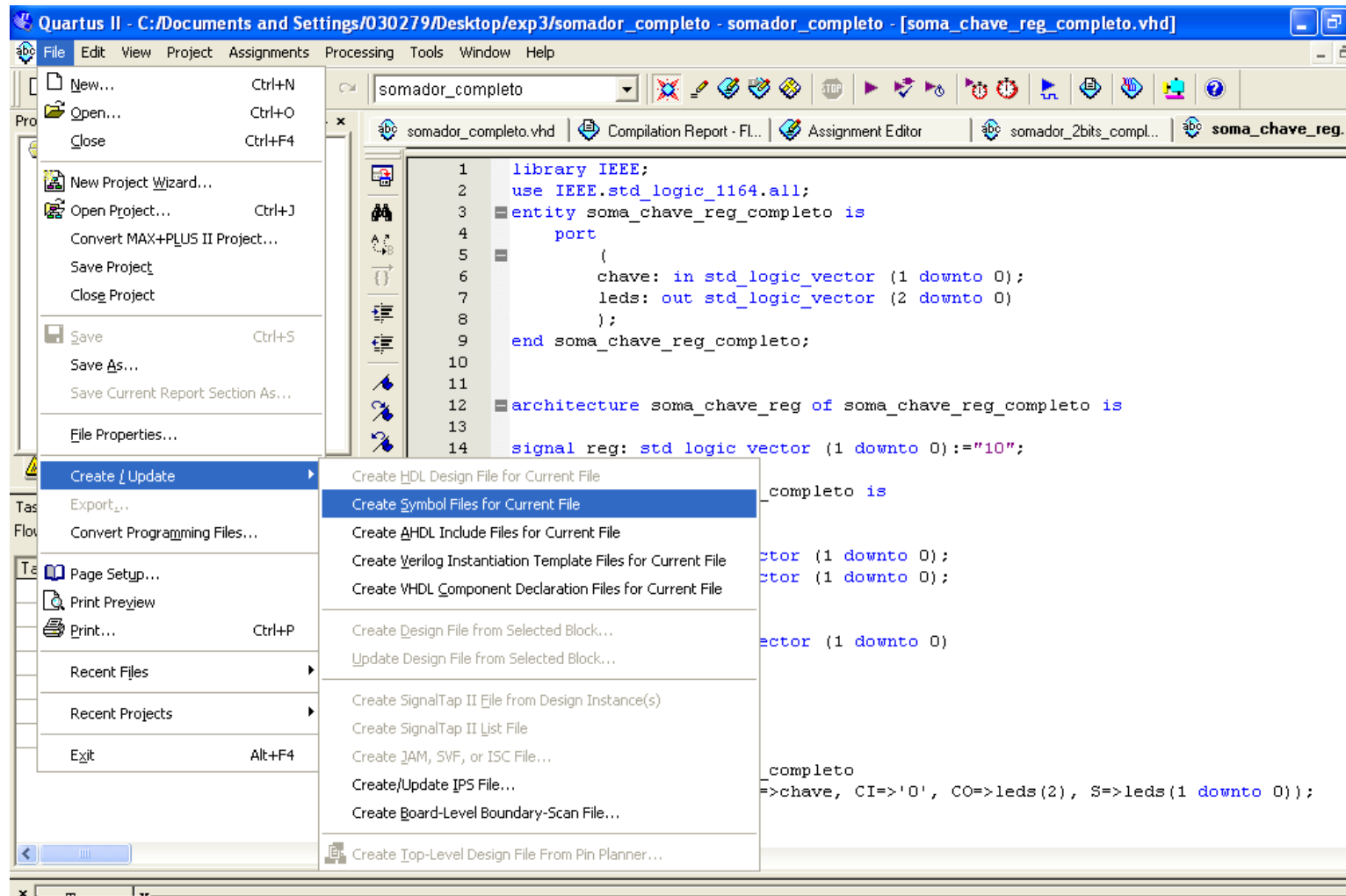




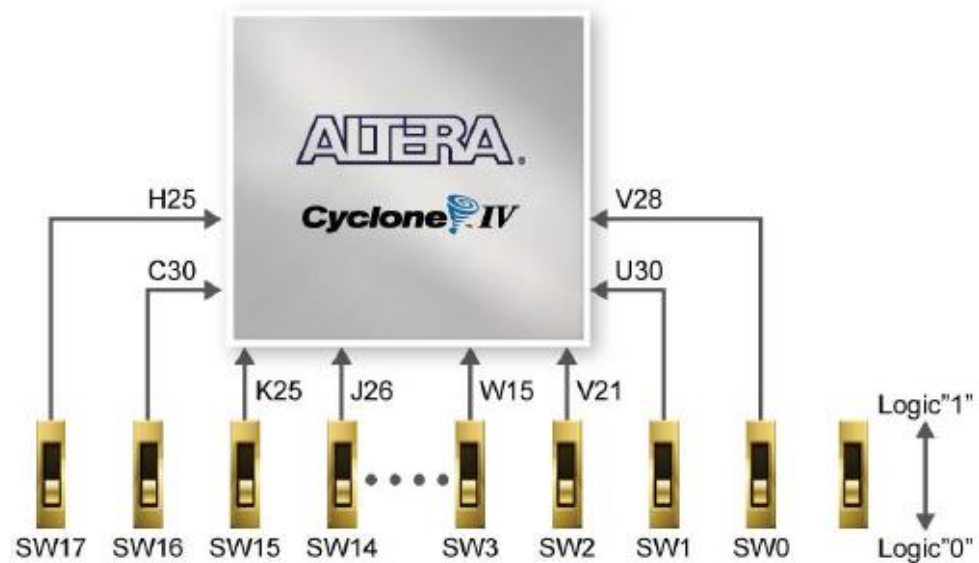
# Relatório de compilação

Flow Summary	
Flow Status	Successful - Thu Aug 03 14:24:32 2017
Quartus Prime Version	16.0.0 Build 211 04/27/2016 SJ Lite Edition
Revision Name	SE2
Top-level Entity Name	SE2
Family	Cyclone IV GX
Device	EP4CGX150DF31C7
Timing Models	Final
Total logic elements	1 / 149,760 ( < 1 % )
Total combinational functions	1 / 149,760 ( < 1 % )
Dedicated logic registers	0 / 149,760 ( 0 % )
Total registers	0
Total pins	3 / 508 ( < 1 % )
Total virtual pins	0
Total memory bits	0 / 6,635,520 ( 0 % )
Embedded Multiplier 9-bit elements	0 / 720 ( 0 % )
Total GXB Receiver Channel PCS	0 / 8 ( 0 % )
Total GXB Receiver Channel PMA	0 / 8 ( 0 % )
Total GXB Transmitter Channel PCS	0 / 8 ( 0 % )
Total GXB Transmitter Channel PMA	0 / 8 ( 0 % )
Total PLLs	0 / 8 ( 0 % )

# Criando blocos do projeto



## Determinando pinos de entrada e saída



## Lista de pinos conectados aos switches




<i>Signal Name</i>	<i>FPGA Pin No.</i>	<i>Description</i>	<i>I/O Standard</i>
SW[0]	PIN_V28	Slide Switch[0]	2.5V
SW[1]	PIN_U30	Slide Switch[1]	2.5V
SW[2]	PIN_V21	Slide Switch[2]	2.5V
SW[3]	PIN_C2	Slide Switch[3]	2.5V
SW[4]	PIN_AB30	Slide Switch[4]	2.5V
SW[5]	PIN_U21	Slide Switch[5]	2.5V
SW[6]	PIN_T28	Slide Switch[6]	2.5V
SW[7]	PIN_R30	Slide Switch[7]	2.5V
SW[8]	PIN_P30	Slide Switch[8]	2.5V
SW[9]	PIN_R29	Slide Switch[9]	2.5V
SW[10]	PIN_R26	Slide Switch[10]	2.5V
SW[11]	PIN_N26	Slide Switch[11]	2.5V
SW[12]	PIN_M26	Slide Switch[12]	2.5V
SW[13]	PIN_N25	Slide Switch[13]	2.5V
SW[14]	PIN_J26	Slide Switch[14]	2.5V
SW[15]	PIN_K25	Slide Switch[15]	2.5V
SW[16]	PIN_C30	Slide Switch[16]	2.5V
SW[17]	PIN_H25	Slide Switch[17]	2.5V




## Lista de pinos conectados aos LEDs

<i>Signal Name</i>	<i>FPGA Pin No.</i>	<i>Description</i>	<i>I/O Standard</i>
LEDR[0]	PIN_T23	LED Red[0]	2.5V
LEDR[1]	PIN_T24	LED Red[1]	2.5V
LEDR[2]	PIN_V27	LED Red[2]	2.5V
LEDR[3]	PIN_W25	LED Red[3]	2.5V
LEDR[4]	PIN_T21	LED Red[4]	2.5V
LEDR[5]	PIN_T26	LED Red[5]	2.5V
LEDR[6]	PIN_R25	LED Red[6]	2.5V
LEDR[7]	PIN_T27	LED Red[7]	2.5V
LEDR[8]	PIN_P25	LED Red[8]	2.5V
LEDR[9]	PIN_R24	LED Red[9]	2.5V
LEDR[10]	PIN_P21	LED Red[10]	2.5V
LEDR[11]	PIN_N24	LED Red[11]	2.5V
LEDR[12]	PIN_N21	LED Red[12]	2.5V
LEDR[13]	PIN_M25	LED Red[13]	2.5V
LEDR[14]	PIN_K24	LED Red[14]	2.5V
LEDR[15]	PIN_L25	LED Red[15]	2.5V
LEDR[16]	PIN_M21	LED Red[16]	2.5V
LEDR[17]	PIN_M22	LED Red[17]	2.5V
LEDG[0]	PIN_AA25	LED Green[0]	2.5V
LEDG[1]	PIN_AB25	LED Green[1]	2.5V
LEDG[2]	PIN_F27	LED Green[2]	2.5V
LEDG[3]	PIN_F26	LED Green[3]	2.5V
LEDG[4]	PIN_W26	LED Green[4]	2.5V
LEDG[5]	PIN_Y22	LED Green[5]	2.5V
LEDG[6]	PIN_Y25	LED Green[6]	2.5V
LEDG[7]	PIN_AA22	LED Green[7]	2.5V
LEDG[8]	PIN_J25	LED Green[8]	2.5V

## Pin Planner

- Menu Assignments, Pin Planner.
- Selecionar os pinos V28 e U30 para as entradas A e B, respectivamente.
- Conectar a saída no pino T23.

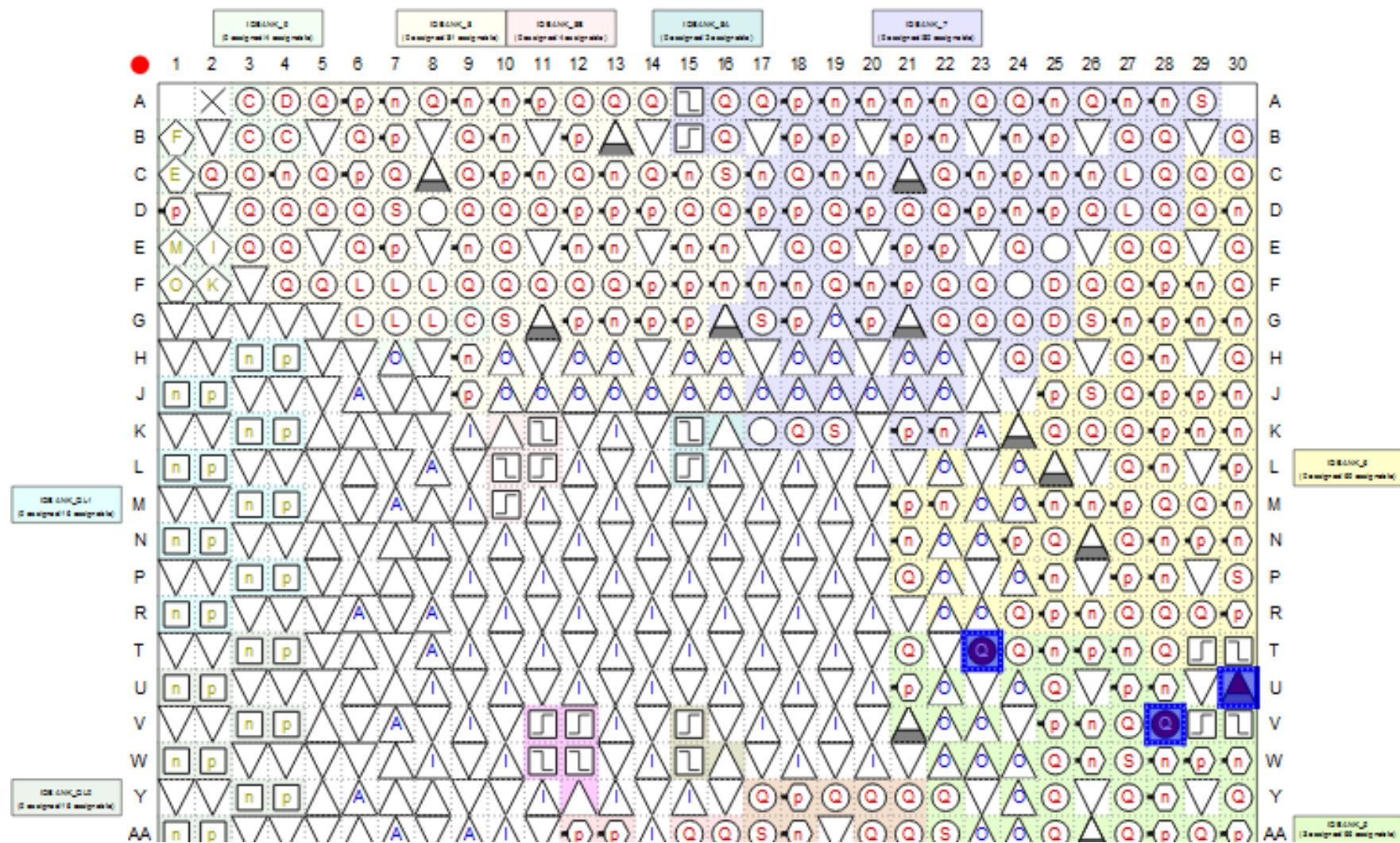
Named: \*  Edit:  

	Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate
	A	Input	PIN_V28	5	B5_N0	PIN_AH12	2.5 V (default)		16mA (default)	
	B	Input	PIN_U30	5	B5_N0	PIN_AG11	2.5 V (default)		16mA (default)	
	S	Output	PIN_T23	5	B5_N0	PIN_AH11	2.5 V (default)		16mA (default)	2 (default)

- Compilar o projeto novamente.

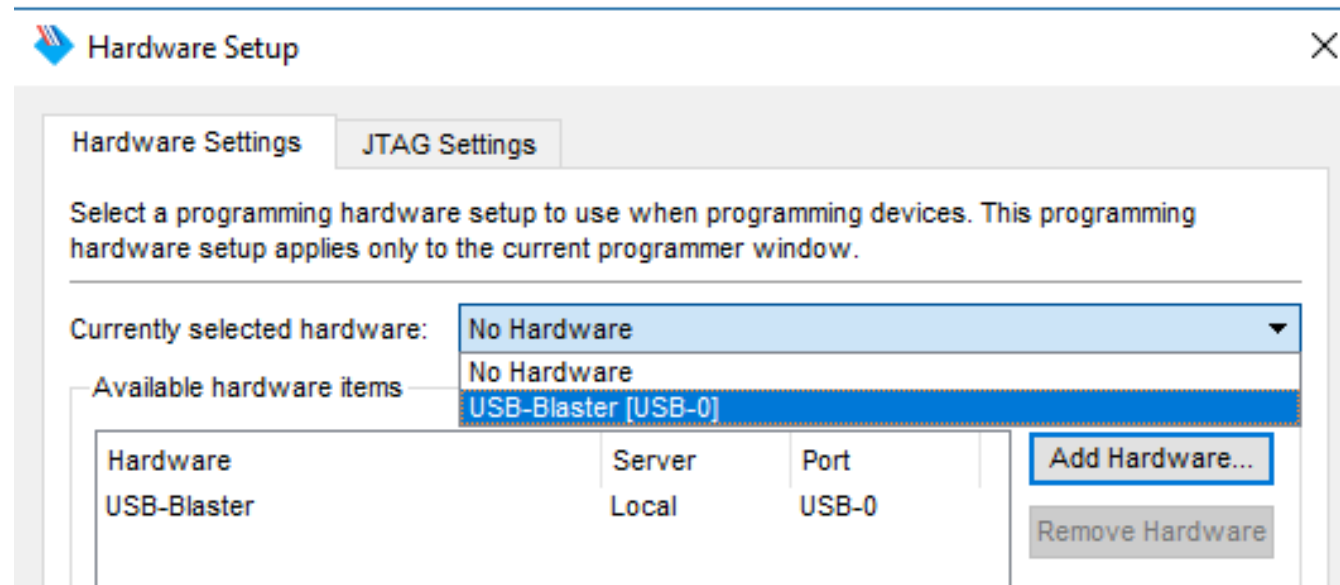


# Pin Planner



# Programmer

- Menu Tools, Programmer.
- Em Hardware Setup, selecione USB-Blaster e volte para tela anterior.





## Programmer

- No menu Add File, acesse o diretório 'output\_files' que está presente na pasta do projeto.
- Indique o arquivo com a extensão '.sof'.
- Selecione a opção 'Program/Configure'.
- Pressione start.