

Arquitetura de Computadores – Dispositivos Lógico Programáveis (*Hardware Reconfigurável*)

Arranjo de Portas Programável em Campo - FPGA

Rodolfo Labiapari Mansur Guimarães

rodolfolabiapari@decom.ufop.br

Lattes: <http://goo.gl/MZv4Dc>

Departamento de Computação – Universidade Federal de Ouro Preto
Ouro Preto - MG – Brasil

Última Atualização: 7 de maio de 2018

Sobre esta Aula

Sobre esta Aula

“Não existe pergunta idiota, existe idiota que não pergunta”.

Sobre Mim

■ **Formação:** Instituto Federa de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) - Campus Formiga.

■ **Pesquisas Aplicadas:**

- Projeto e Desenvolvimento de um Carro Robô controlado por *Smartphone*, Utilizando a Plataforma Amarino (Google Android e Arduino);
- Projeto e Desenvolvimento de um *Hardware Reconfigurável* de Criptografia para a Transmissão Segura de Dados.

■ **Trabalho de Conclusão de Curso:**

- Desenvolvimento de um *Device Driver* para Comunicação com um *Hardware Reconfigurável*.

■ **Lattes:** <http://goo.gl/MZv4Dc>.



UFOP

Sumário

1 Matemática Discreta

- Lógica Proposicional e Eletrônica Digital

2 Dispositivos Lógico Programáveis

3 FPGA

4 UFOP - Laboratório iMobilis

5 Bibliografia

Lógica e Eletrônica - Tabelas Verdades

- **Tabela verdade de uma função boleana:** apresenta **os valores da função** $y = f(A, B)$ para **todas as combinações possíveis** dos valores das variáveis.

Figura: Exemplo de tabela verdade com 2 entradas

- Tabela é **uma** de várias formas de representação de funções lógicas.

A	B	y
0	0	1
0	1	0
1	0	1
1	1	1

Fonte: Disponível em: <http://eletro.g12.br/arquivos/materiais/electronica4.pdf>.

Lógica e Eletrônica - Tabelas Verdades

Figura: Exemplo de tabela verdade com 3 entradas

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Lógica e Eletrônica - Tabelas Verdades

Figura: Exemplo de tabela verdade com 3 entradas

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Diagram illustrating the mapping of rows to minterms:

- Row 1 (A=0, B=0, C=0, F=1) maps to $\bar{A}\bar{B}\bar{C}$
- Row 4 (A=0, B=1, C=1, F=1) maps to $\bar{A}BC$
- Row 5 (A=1, B=0, C=0, F=1) maps to $A\bar{B}\bar{C}$
- Row 8 (A=1, B=1, C=1, F=1) maps to ABC

Fonte: Disponível em: <http://eletro.g12.br/arquivos/materiais/electronica4.pdf>.

Lógica e Eletrônica - Portas Lógicas

- **Portas lógicas:** circuitos eletrônicos básicos que possuem uma ou mais entradas e uma única saída.
- Nas entradas e na saída, podemos associar valores booleanos.
- Em eletrônica digital atribuímos valores de tensão
 - Associamos ao 5 V o estado “1” e ao 0 V, o estado “0”.

Lógica e Eletrônica - Portas Lógicas - Porta NOT

Figura: Representação da Porta Lógica NOT

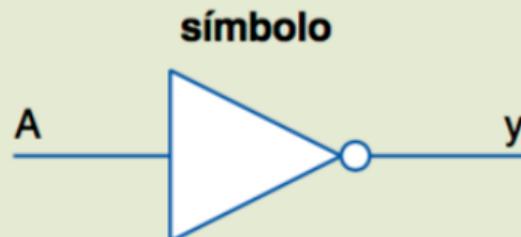


tabela verdade

A	y
0	1
1	0

expressão booleana

$$y = \bar{A}$$

A → entrada e y → saída

PORCA INVERSORA tem somente 1 entrada



Fonte: Disponível em: <http://eletro.g12.br/arquivos/materiais/electronica4.pdf>.

Lógica e Eletrônica - Portas Lógicas - Porta AND

Figura: Representação da Porta Lógica AND

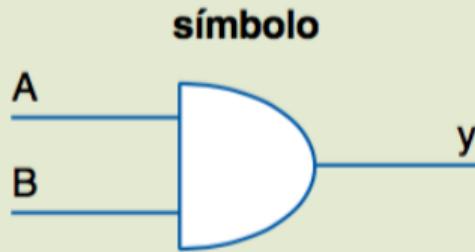


tabela verdade

A	B	y
0	0	0
0	1	0
1	0	0
1	1	1

expressão booleana

$$y = AB \text{ ou } y = A \cdot B$$

A saída é “1” somente se todas as entradas forem “1”

Lógica e Eletrônica - Portas Lógicas - Porta NAND

Figura: Representação da Porta Lógica NAND

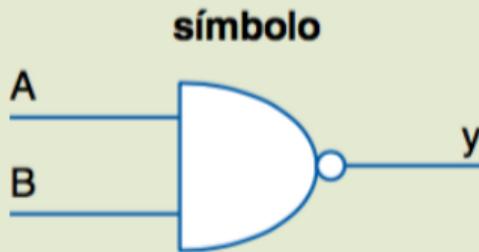


tabela verdade

A	B	y
0	0	1
0	1	1
1	0	1
1	1	0

expressão booleana

$$y = \overline{AB} \text{ ou } y = \overline{A} \cdot \overline{B}$$

A saída é “0” somente se todas as entradas forem “1”

Lógica e Eletrônica - Portas Lógicas - Porta OR

Figura: Representação da Porta Lógica OR

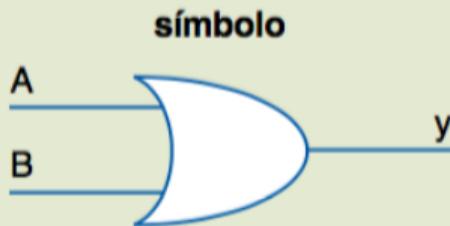


tabela verdade

A	B	y
0	0	0
0	1	1
1	0	1
1	1	1

expressão booleana

$$y = A + B \text{ (lê-se A OU B)}$$

A e B → entradas

y → saída

A saída é “0” somente se todas as entradas forem zero

Lógica e Eletrônica - Portas Lógicas - Porta NOR

Figura: Representação da Porta Lógica NOR

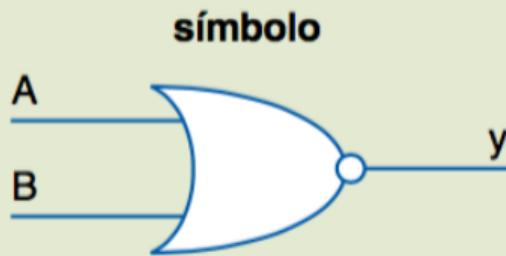


tabela verdade

A	B	y
0	0	1
0	1	0
1	0	0
1	1	0

expressão booleana

$$y = \overline{A + B}$$

A saída é “1” somente se todas as entradas forem zero



UFOP

Fonte: Disponível em: <http://eletro.g12.br/arquivos/materiais/electronica4.pdf>.

Lógica e Eletrônica - Portas Lógicas - Porta XOR

Figura: Representação da Porta Lógica XOR

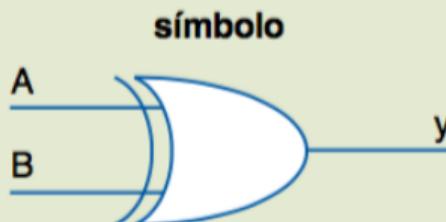


tabela verdade

A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

expressão booleana

$$y = A \oplus B$$

Saída 1 se as entradas forem diferentes

Lógica e Eletrônica - Portas Lógicas - Porta XNOR

Figura: Representação da Porta Lógica XNOR

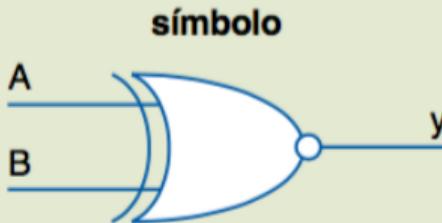


tabela verdade

A	B	y
0	0	1
0	1	0
1	0	0
1	1	1

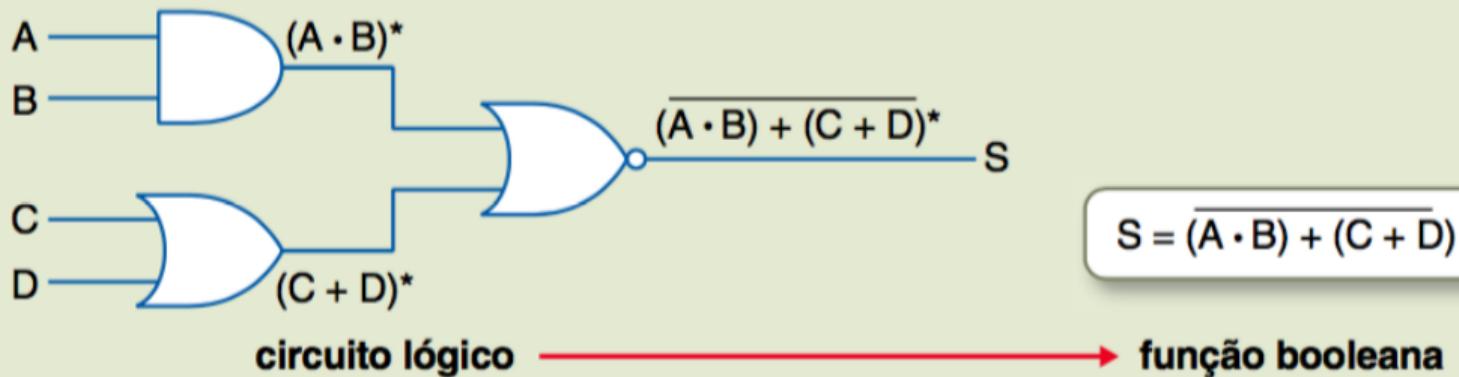
expressão booleana

$$y = A \odot B$$

Saída 1 se as entradas forem iguais

Lógica e Eletrônica - Circuito Digital

Figura: Representação de um circuito simples



Fonte: Disponível em: <http://eletro.g12.br/arquivos/materiais/electronica4.pdf>.

Lógica e Eletrônica - Introdução ao Assunto

Questão: Porque falar sobre Portas Lógicas se o assunto da aula é *Hardware Reconfigurável*?

- Circuitos comprado hoje, como um processador, não permitem que sejam alterados.
- E se



UFOP

Sumário

1 Matemática Discreta

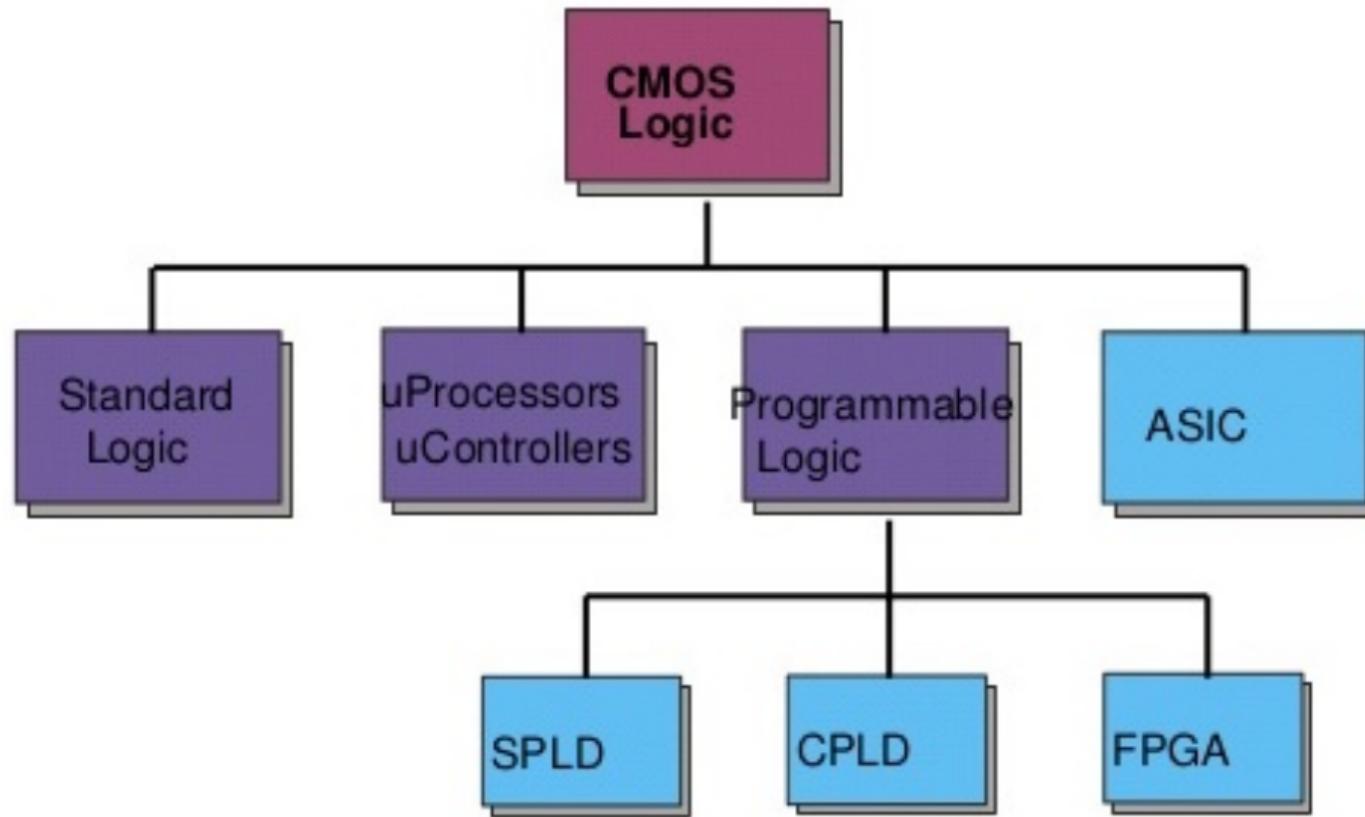
2 **Dispositivos Lógico Programáveis**

3 FPGA

4 UFOP - Laboratório iMobilis

5 Bibliografia

Dispositivos Lógico Programáveis



Dispositivos Lógico Programáveis

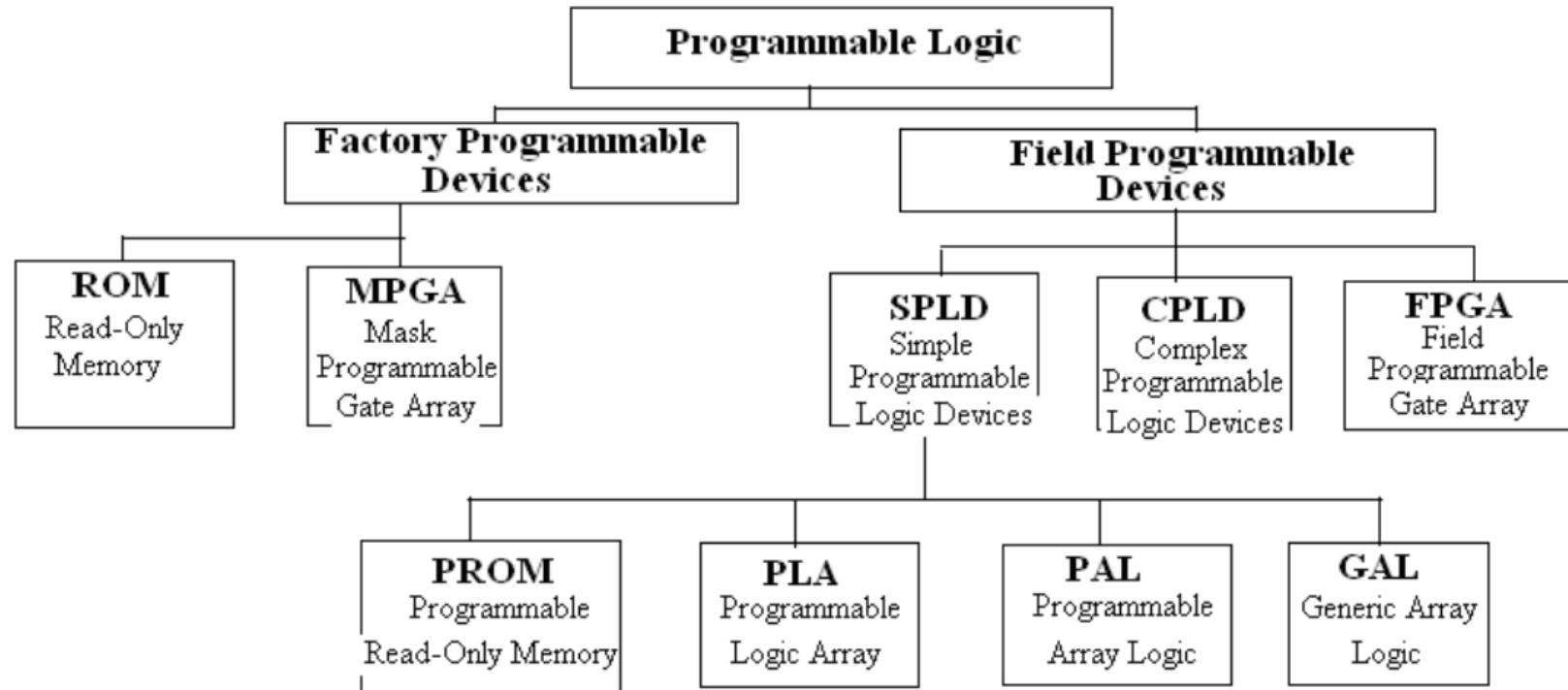


Figura: Programmable Logic Device.

Dispositivos Lógico Programáveis

- Tocci [7] cita que os dispositivos lógicos programáveis (PLD¹) são as “*maravilhas de flexibilidade de projeto*”.
- Ao contrário de uma porta lógica, que tem uma função fixa, um PLD tem uma função indefinida após a sua fabricação.
- Para utilizá-lo em um circuito, deve-se **programá-lo previamente**.
- A lógica programável proporciona ao projetista:
 - A possibilidade de se **adequar aos vários níveis de projetos** sendo estes:
 - Para desenvolvimento de protótipos ou mesmo circuitos finais.
 - **Fácil alteração** do projeto a **qualquer momento**;
 - Programação por meio de Álgebra Boole, Mapa de Karnaugh, e Linguagens de Descrição de *Hardware*.



Dispositivos Lógico Programáveis

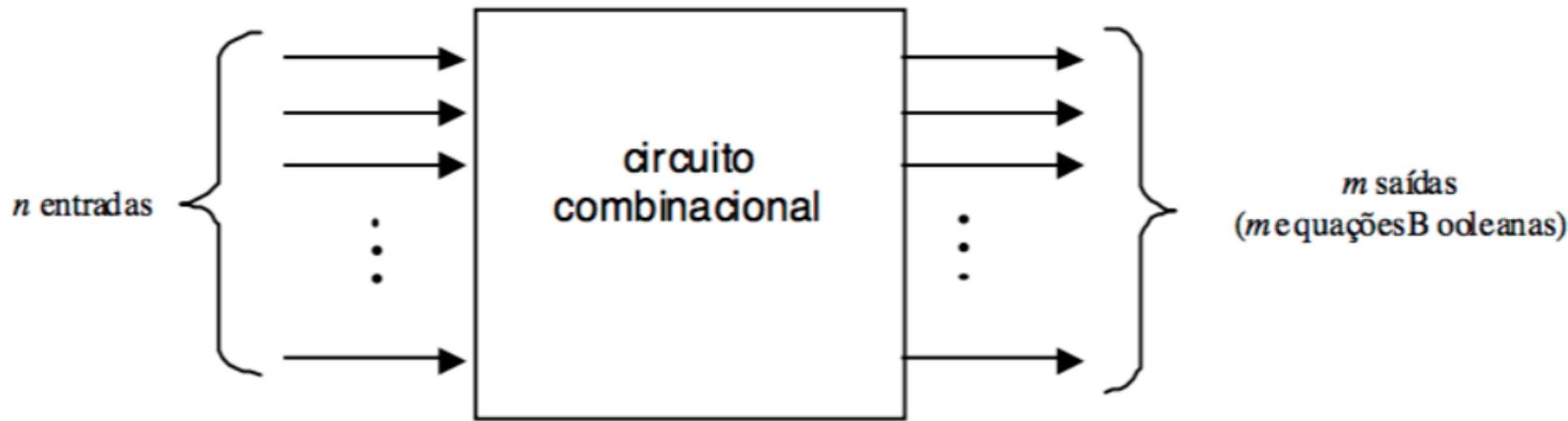


Figura: Exemplo de um circuito combinacional.



Dispositivos Lógico Programáveis - FPGA

- Em meados dos anos 80, a empresa **Xilinx** e Gerald Estrin anuncia uma nova arquitetura.
 - Isso foi um **choque** na época pois cria-se então um **novo paradigma de projetos de circuitos integrados**;
 - Entre vários tipos de PLD existe um tipo que é baseado na tecnologia *gate array* e é chamado de **FPGA**².
- O FPGA usa uma rede de portas lógicas cuja programação é feita pelo **projetista e não pelo fabricante**.
 - Sendo reprogramados, o usuário pode utilizá-lo num projeto e logo em seguida reprogramá-lo para que execute outro projeto.
- Deixando mais claro, o termo “**campo**” na nomenclatura é apenas um termo da engenharia utilizado para indicar o **mundo de fora da fábrica**.



Dispositivos Lógico Programáveis - FPGA

- Excelentes para **desenvolver um protótipo de circuito integrado** a fim da realização de testes **antes da fabricação em massa** [6] e ensino.
- Seu custo em relação ao ASIC³ é menor;
- Seu tempo total gasto desde a especificação do projeto até sua sintetização também é menor
 - ASICs são produzidos geralmente em **larga escala**
 - O custo de fabricação de 1 circuito é caro;
 - O custo de fabricação em massa também é caro.
 - Qualquer erro no ASIC, deve-se reiniciar todo o processo de desenvolvimento
 - Sendo que no FPGA isso levaria talvez algumas horas/minutos e custo zero.
 - Circuitos feitos manualmente...



Dispositivos Lógico Programáveis - FPGA - Projetando um Circuito

1 Descrição em Alto Nível;

2 Simulação Lógica;

Dispositivos Lógico Programáveis - FPGA - Projetando um Circuito

- 1 Descrição em Alto Nível;**
- 2 Simulação Lógica;**
- 3 Síntese e Mapeamento;**
- 4 Distribuição dos Blocos Lógicos e Roteamento dos Sinais;**
- 5 Análise Temporal e Simulação;**

Dispositivos Lógico Programáveis - FPGA - Projetando um Circuito

- 1 Descrição em Alto Nível;**
- 2 Simulação Lógica;**
- 3 Síntese e Mapeamento;**
- 4 Distribuição dos Blocos Lógicos e Roteamento dos Sinais;**
- 5 Análise Temporal e Simulação;**
- 6 Configuração de Dispositivos de Lógica Programável;**

Dispositivos Lógico Programáveis - FPGA - Projetando um Circuito

- 1 Descrição em Alto Nível;**
- 2 Simulação Lógica;**
- 3 Síntese e Mapeamento;**
- 4 Distribuição dos Blocos Lógicos e Roteamento dos Sinais;**
- 5 Análise Temporal e Simulação;**
- 6 Configuração de Dispositivos de Lógica Programável;**
- 7 Teste de Avaliação.**



UFOP

Sumário

1 Matemática Discreta

2 Dispositivos Lógico Programáveis

3 **FPGA**

- Hardware
- Software
- Programação

4 UFOP - Laboratório iMobilis

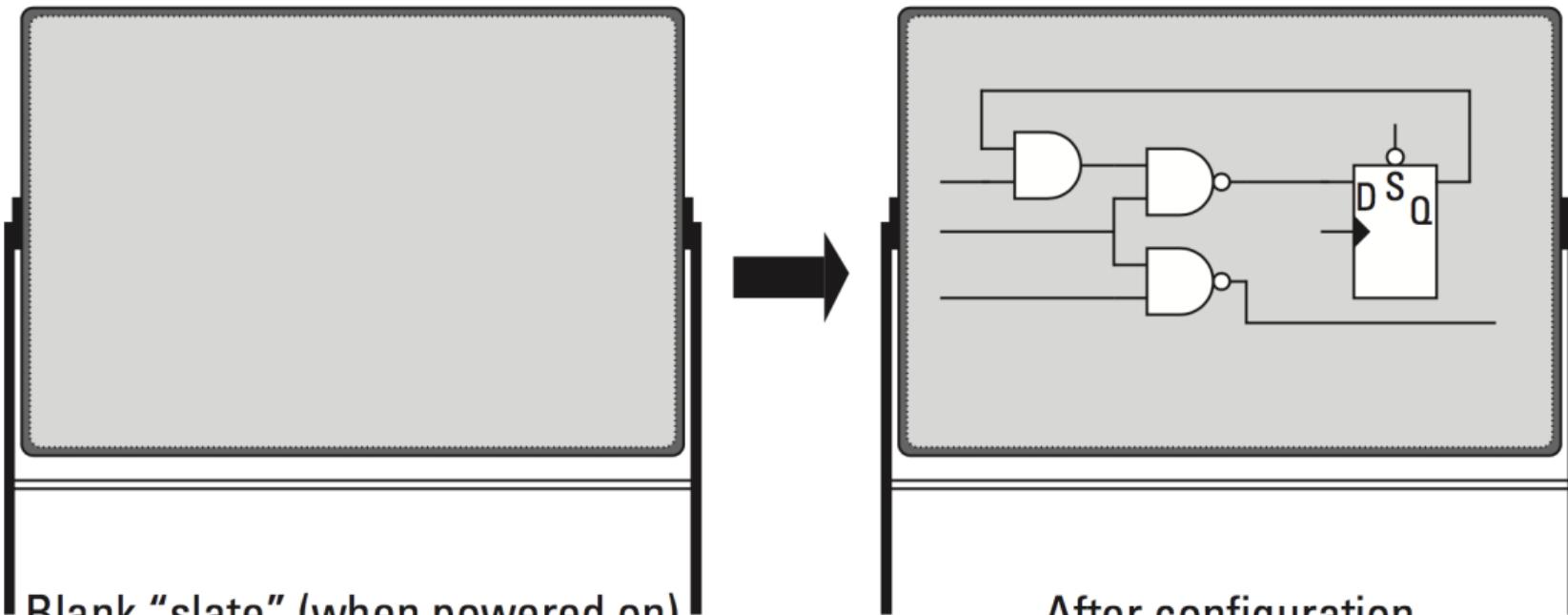
5 Bibliografia

FPGA - Hardware - Visão Geral

- Nos FPGAs, sua configuração é totalmente **volátil**
 - O projeto deve ser **recarregada** sempre que é **ligada**;
 - Para isso, a configuração é normalmente guardada numa memória não volátil como a EEPROM.
- Internamente:
 - Consiste num **arranjo de blocos lógicos e canais de roteamento**.
 - Significa que é capaz de alterar seus caminhos de dados/fluxos **habilitando/desabilitando módulos** [2].

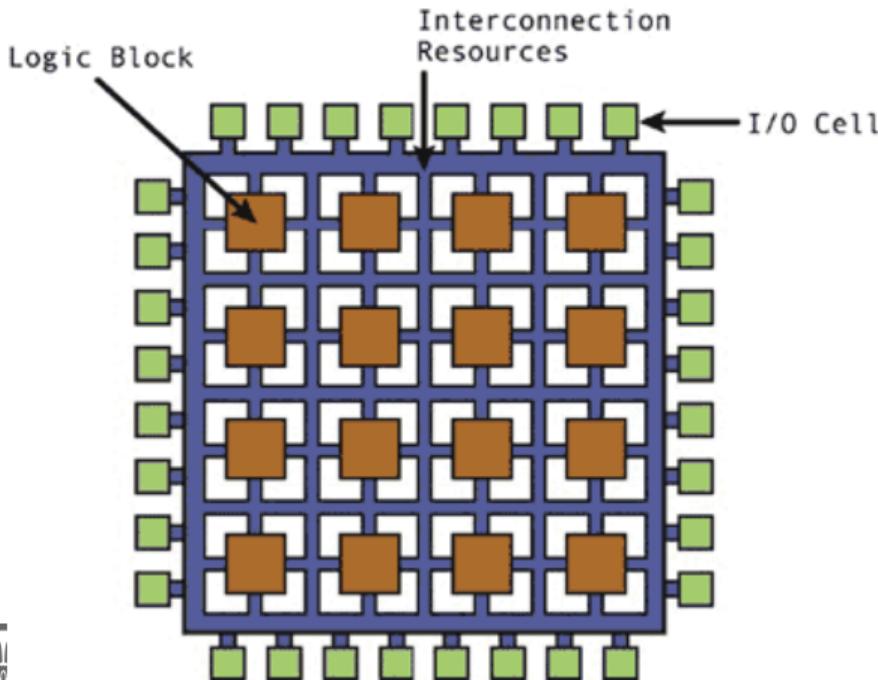
FPGA - Hardware - Visão Geral

Figura: “FPGA é um *quadro negro* que pode ser *desenhado* qualquer circuito digital”.



UFOP

FPGA - Hardware - Roteamento de Blocos

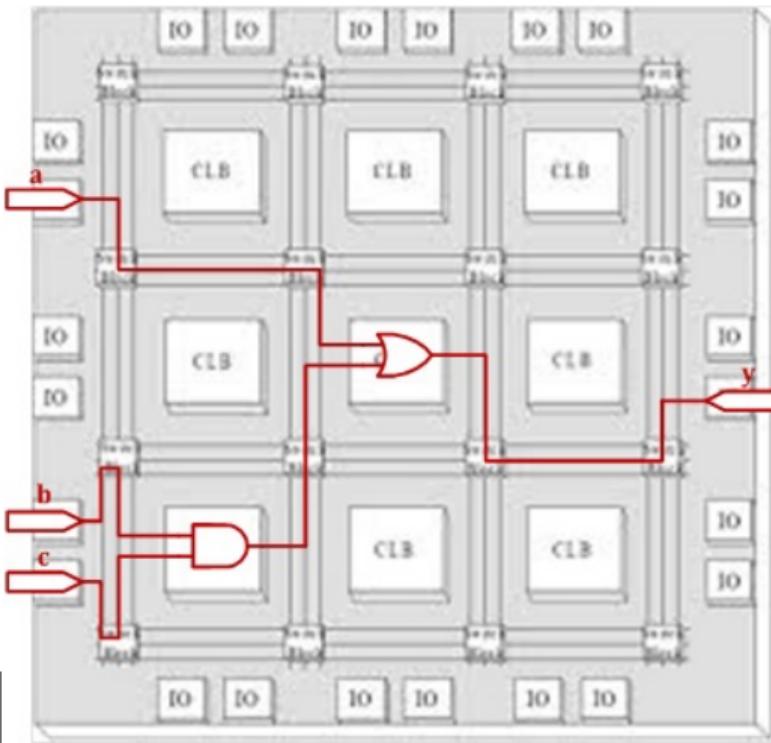


Três módulos:

- Blocos de Entrada e Saída;
- Blocos Lógicos Configuráveis;
- Chaves de Interconexão.

Figura: Exemplo de roteamento interno.

FPGA - Hardware - Roteamento de Blocos

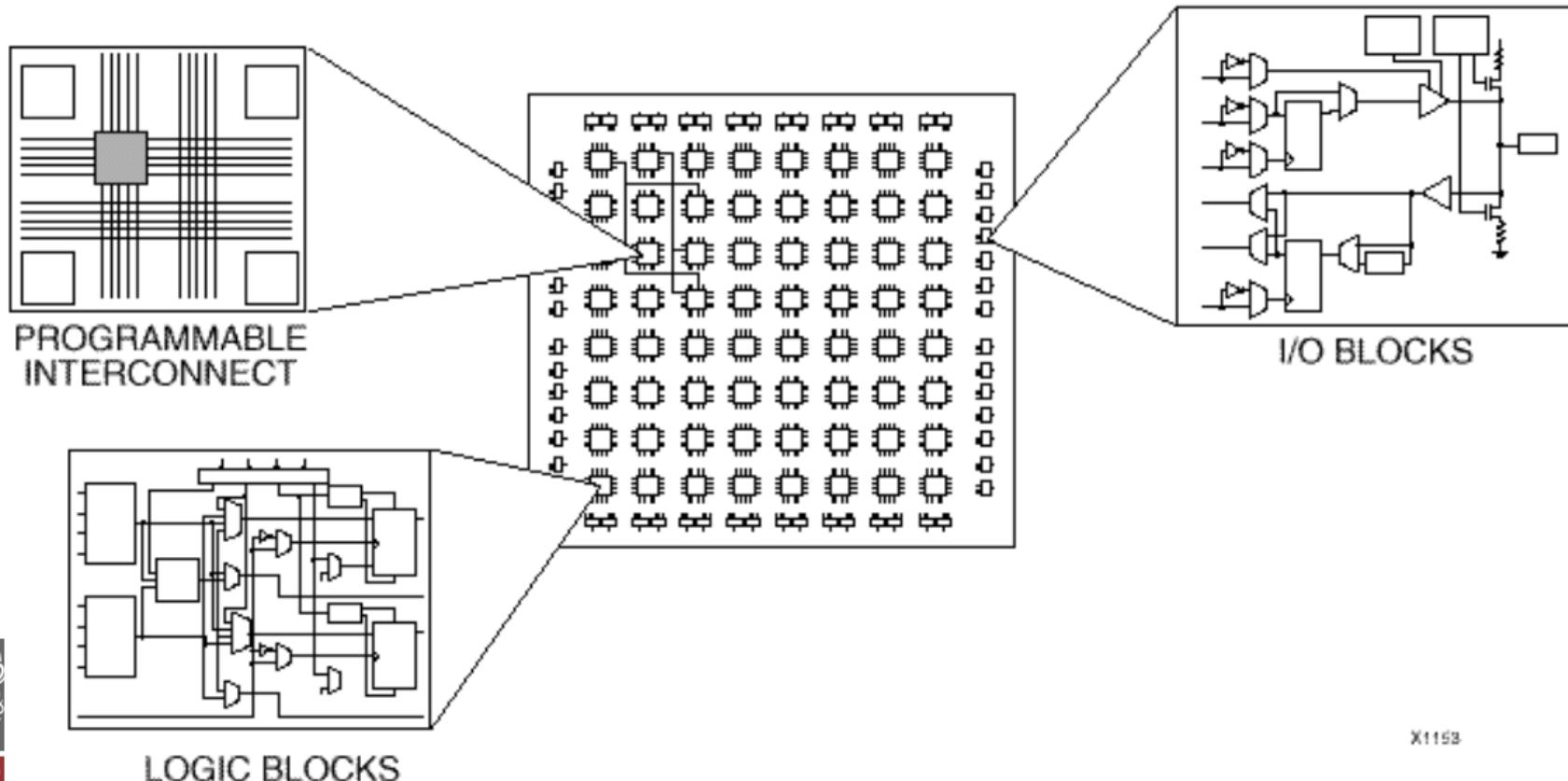


Três módulos:

- Blocos de Entrada e Saída;
- Blocos Lógicos Configuráveis;
- Chaves de Interconexão.

Figura: Outro exemplo de roteamento interno no

FPGA - Hardware - Roteamento de Blocos



FPGA - Hardware - Roteamento de Blocos

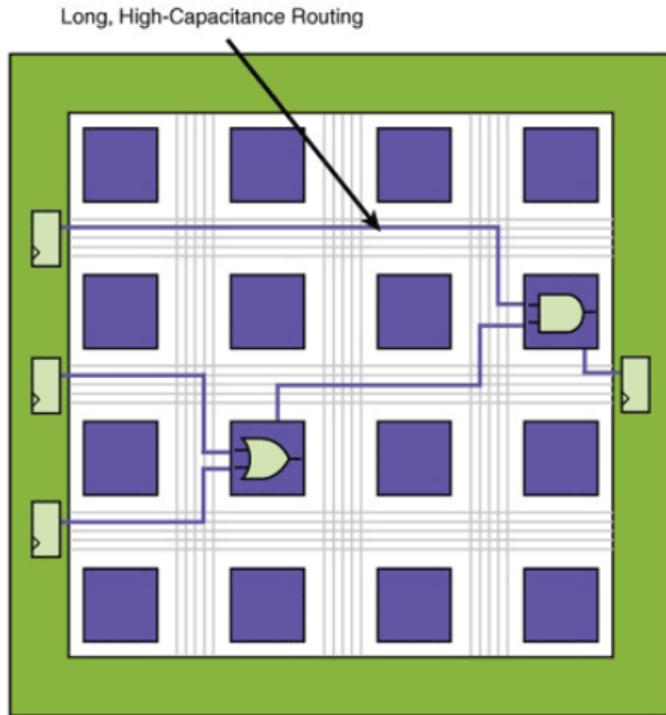
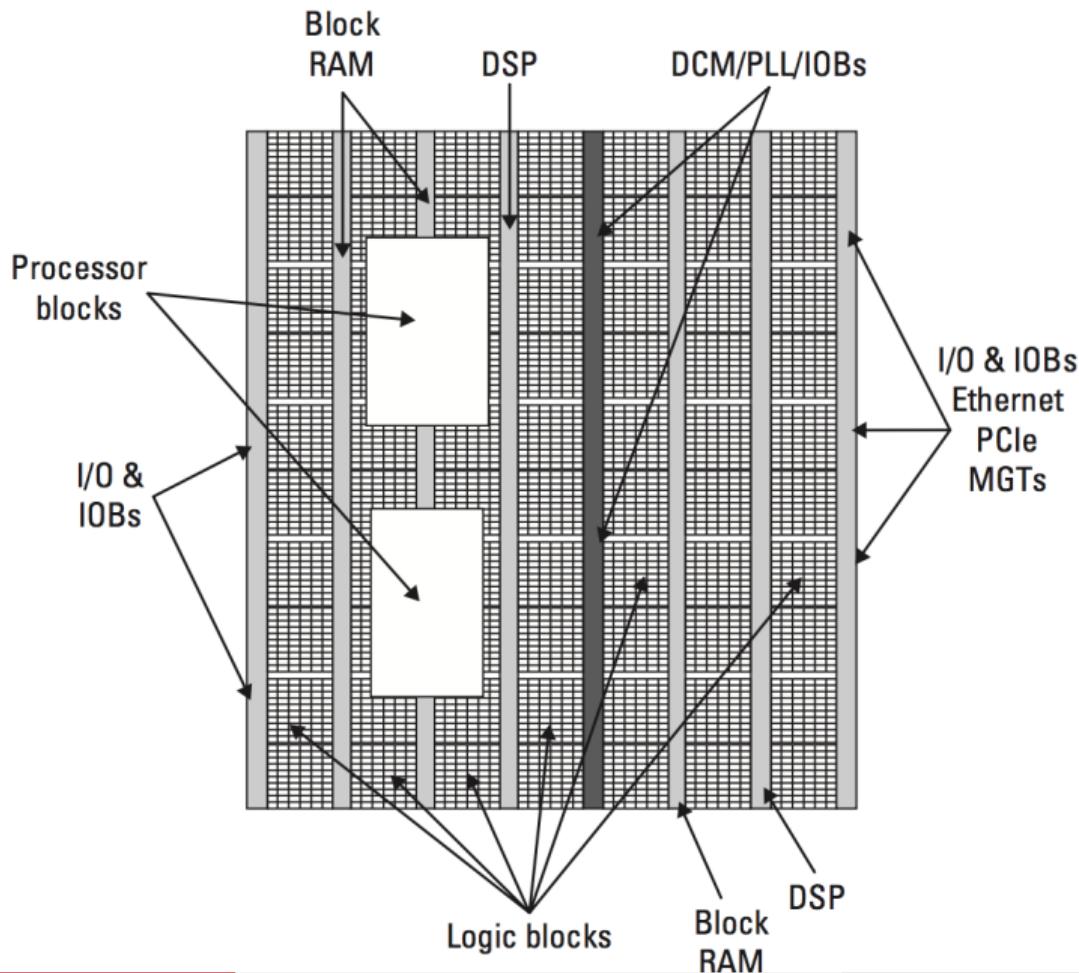
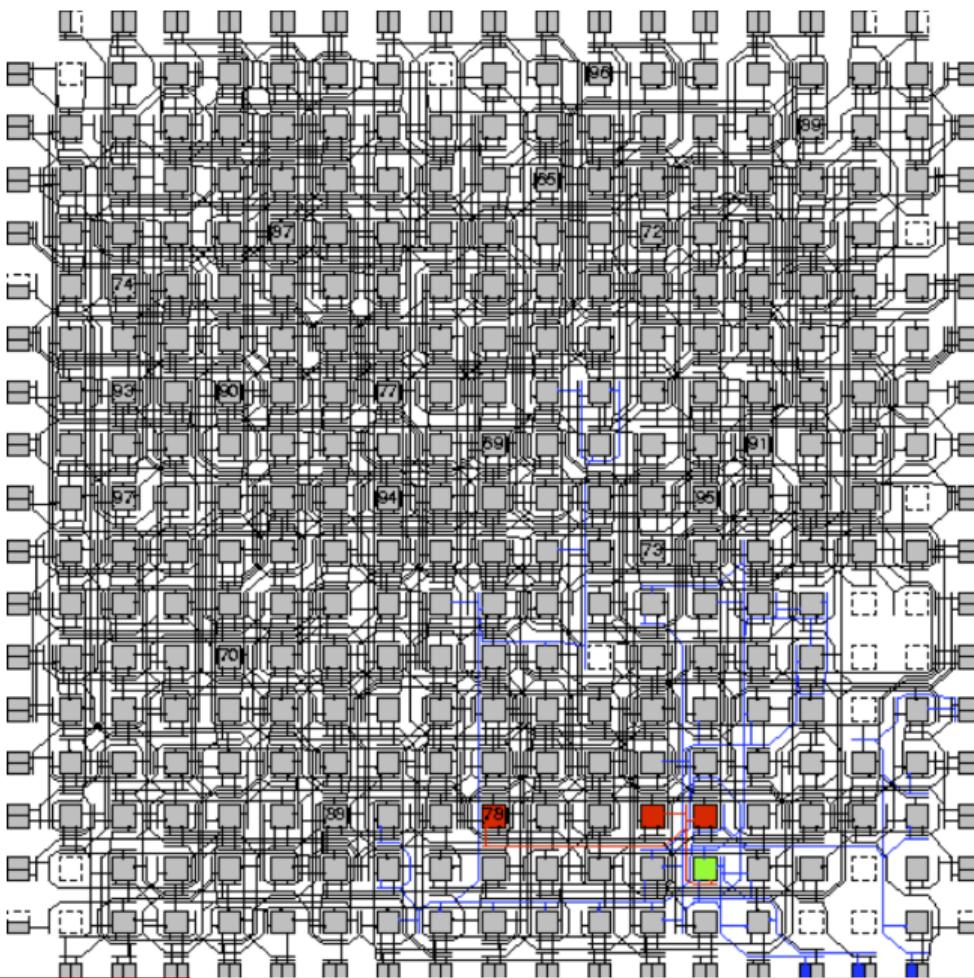


Figura: Exemplo de roteamento interno.

- O FPGA **não possui funções lógicas** como AND, OR...
 - Seu arranjo de células reconfiguráveis permite a **implementação** dessas portas/tabelas.
- Distribuições lógicas ruins e encaminhamentos mal feitos geram atrasos de processamento
 - Estudar qual a melhor maneira projetar cada circuito;
 - A implementação afeta diretamente seu design e por fim, no desempenho.





FPGA - Hardware - Placa

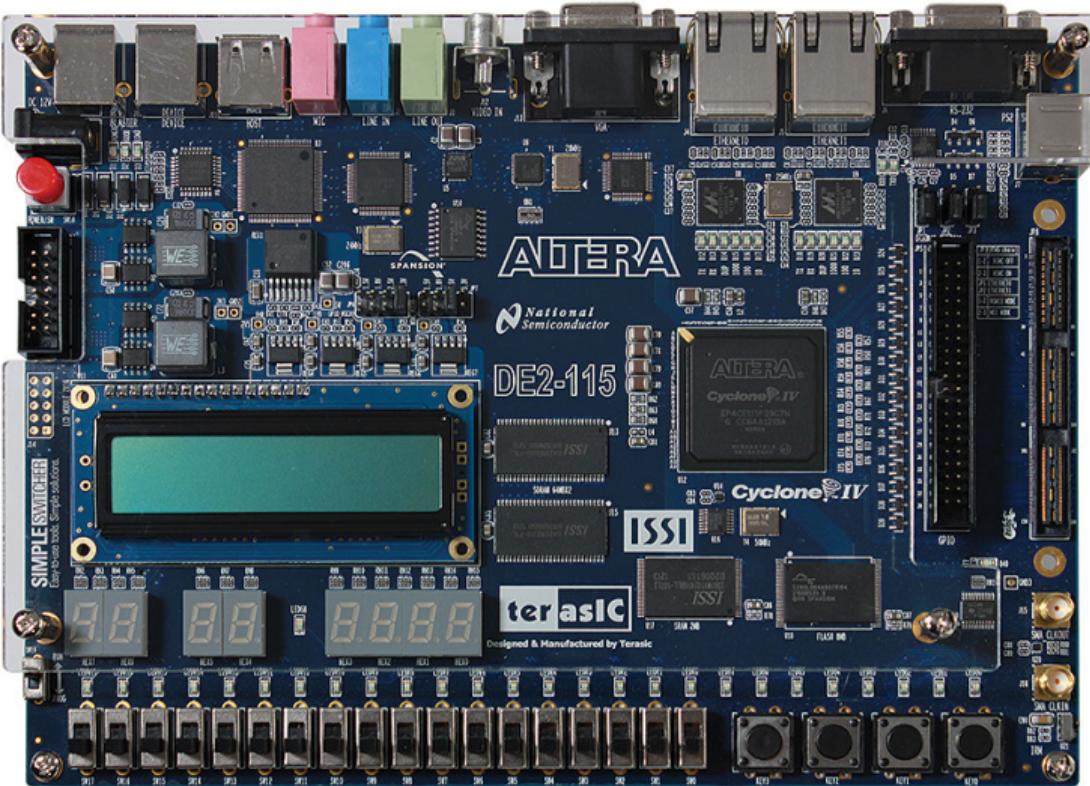
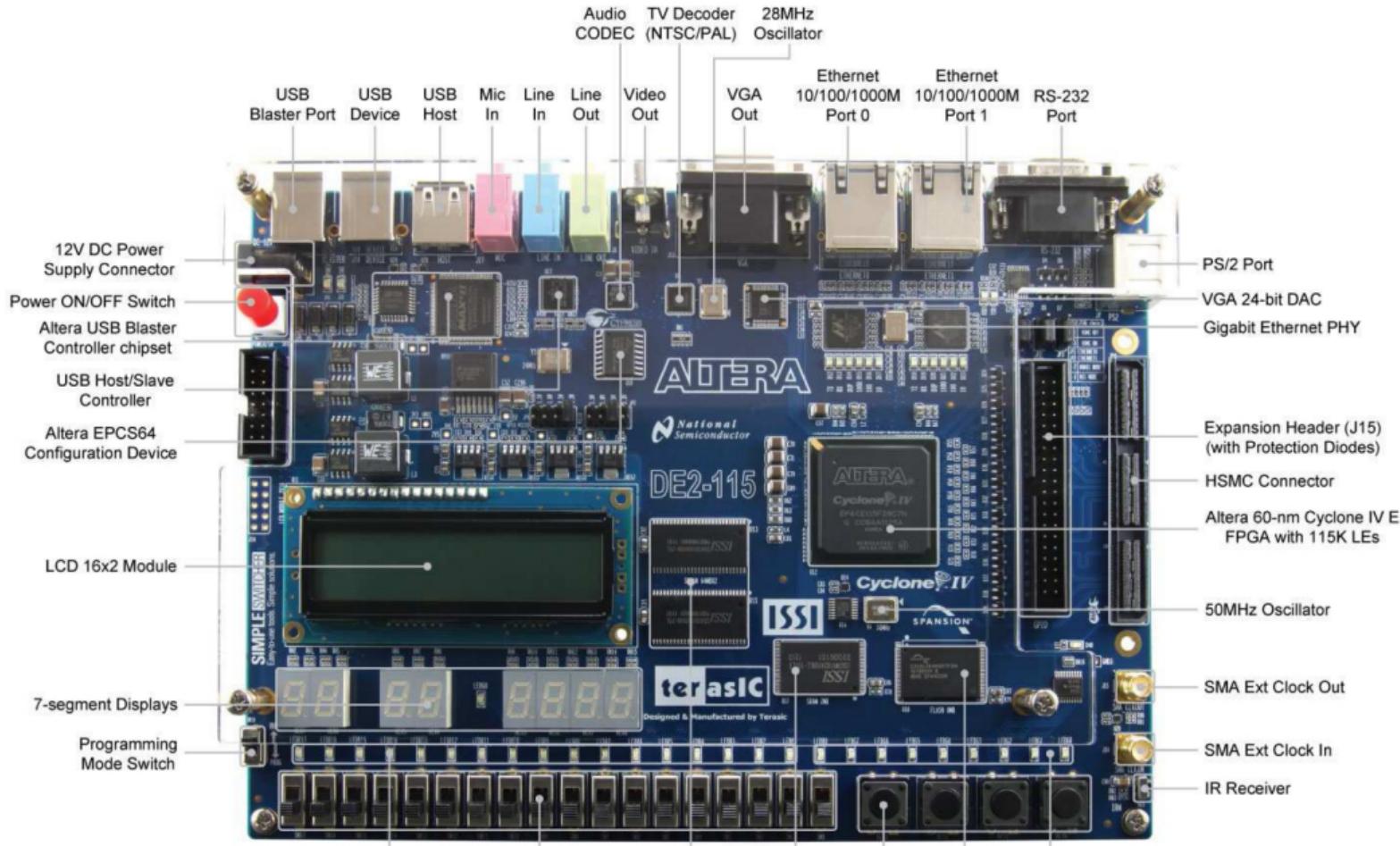


Figura: FPGA de Desenvolvimento Altera DE2-115.





FPGA - Software - IDE de Desenvolvimento

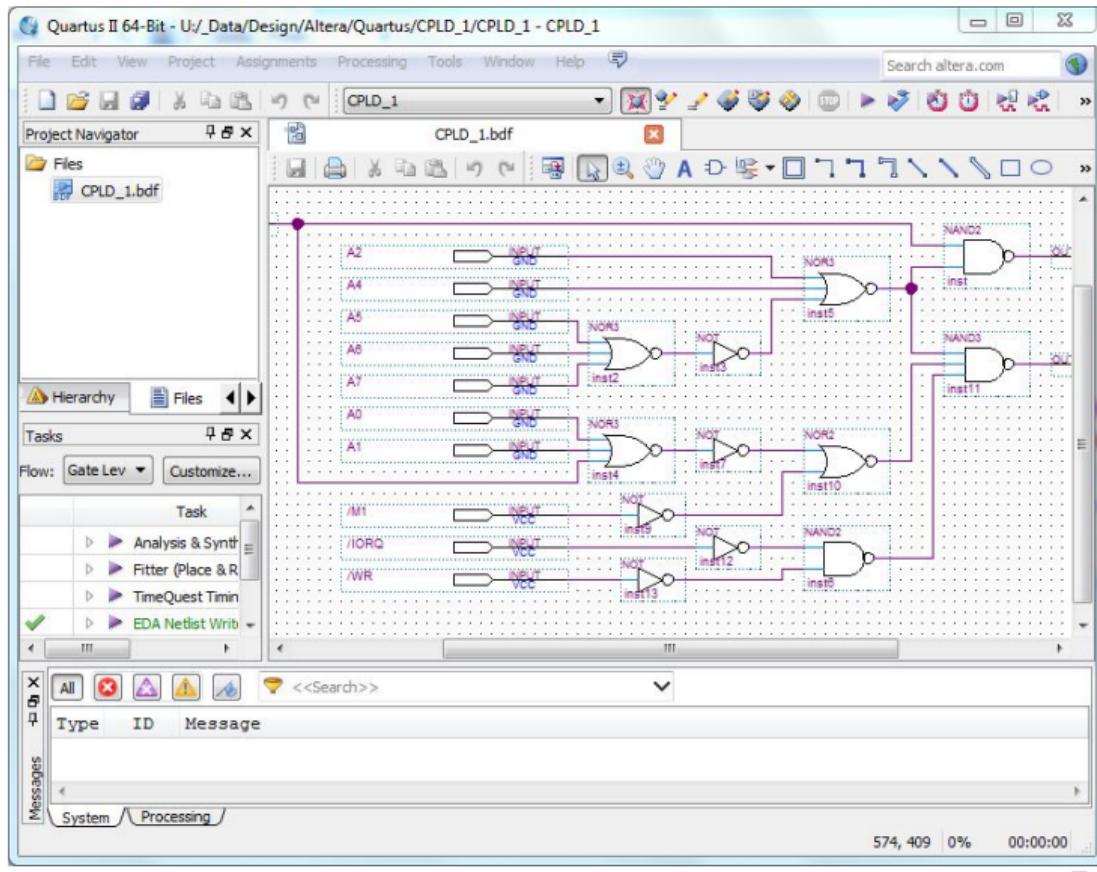
The screenshot shows the Quartus II 64-Bit software interface. The main window title is "Quartus II 64-Bit - C:/altera/projetos/ldd/InterfaceDES/InterfaceDES - InterfaceDES". The menu bar includes File, Edit, View, Project, Assignments, Processing, Tools, Window, Help, and a search bar for "Search altera.com".

The left sidebar contains the "Project Navigator" and "Status" panes. The "Project Navigator" pane shows the project structure: "Entity" (Cyclone IV E: EP4CE115F29C7) and "interfaceDES". The "Status" pane displays the progress of various compilation steps: Full Compilation (100%), Analysis & Synthesis (100%), Fitter (100%), Assembler (100%), TimeQuest Timing Analysis (100%), and EDA Netlist Writer (100%).

The central area is titled "Compilation Report - InterfaceDES" and contains a "Flow Summary" table:

Flow Summary	
Flow Status	Successful - Wed Nov 11 15:47:10 2013.0.0 Build 156 04/24/2013 SJ Web
Quartus II 64-Bit Version	InterfaceDES
Revision Name	interfaceDES
Top-level Entity Name	Cyclone IV E
Family	EP4CE115F29C7
Device	Final
Timing Models	5,239 / 114,480 (5 %)
Total logic elements	2,853 / 114,480 (2 %)
Total combinational functions	4,283 / 114,480 (4 %)
Total registers	4283
Total pins	78 / 529 (15 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

FPGA - Software - IDE de Desenvolvimento



FPGA - Software - IDE de Desenvolvimento

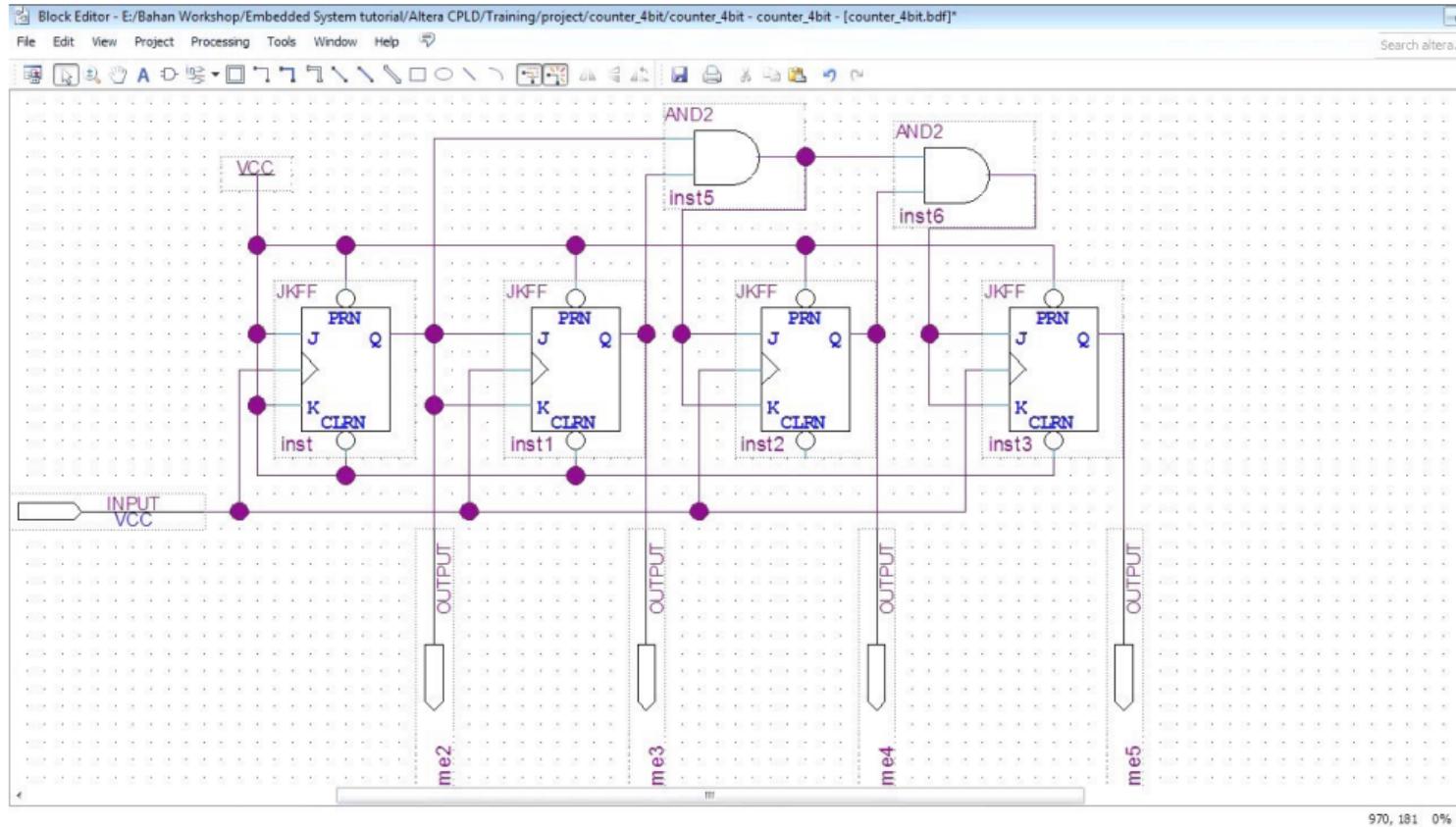


Figura: Projeto Contador 4-bit usando portas lógicas
Arquitetura de Computadores – Hardware Reconfigurável

Última Atualização: 7 de maio de 2018

39 / 78



FPGA - Software - IDE de Desenvolvimento

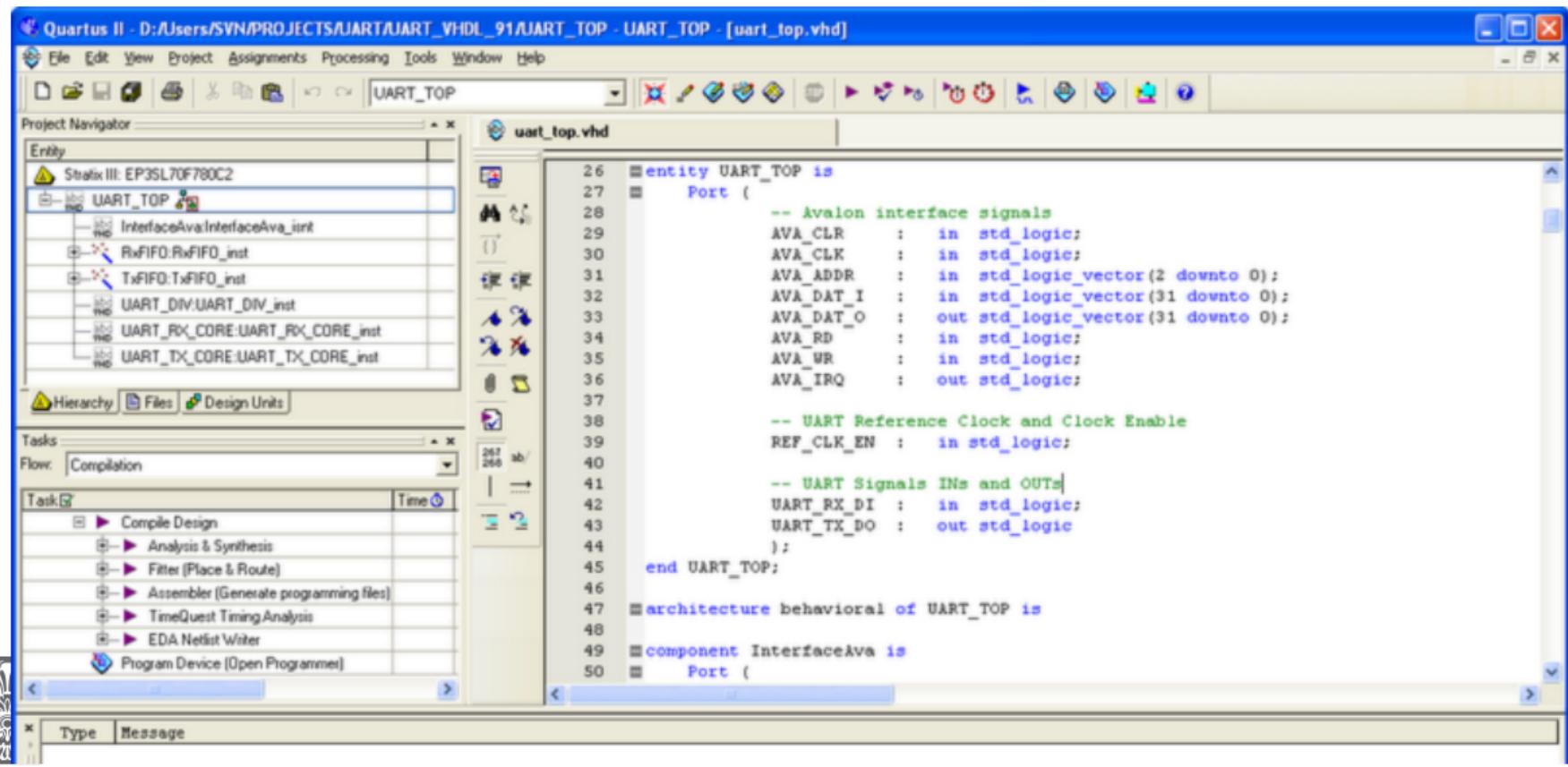


Figura: Projeto de uma interface UART usando VHDL

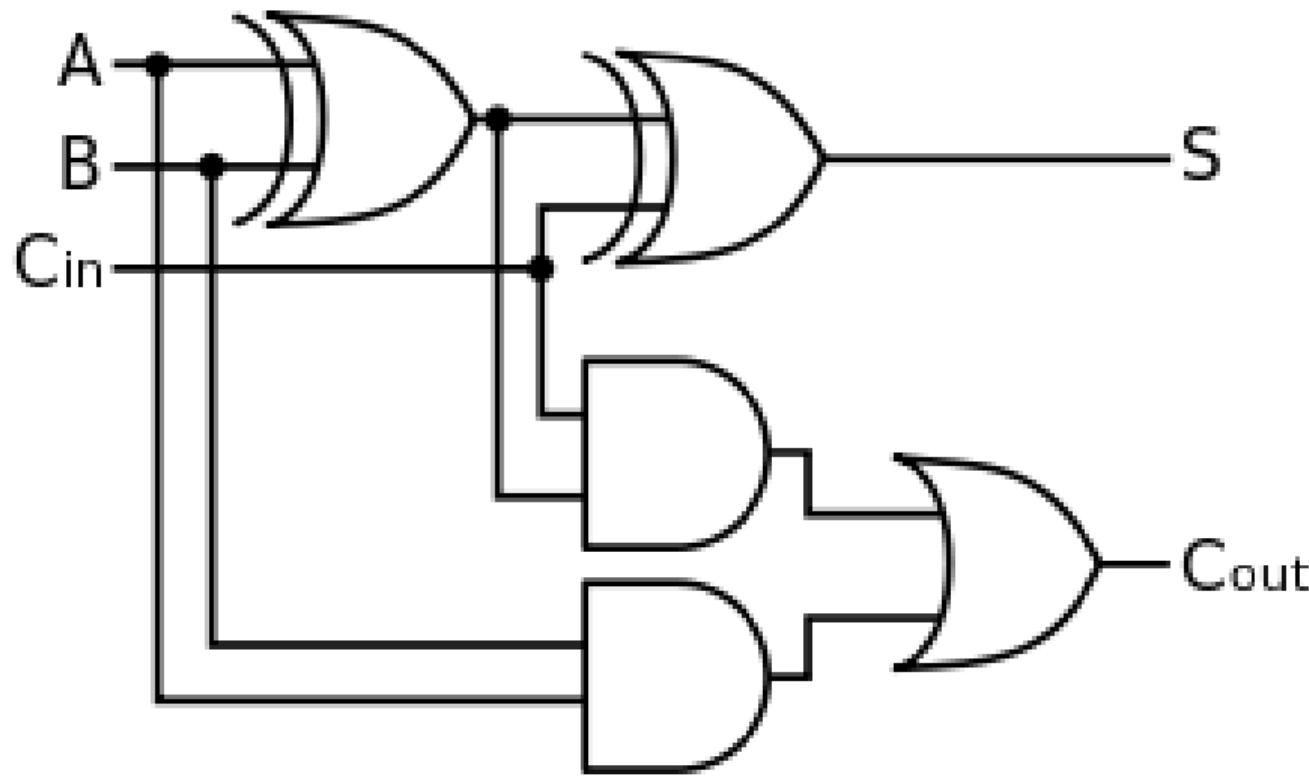
Arquitetura de Computadores – Hardware Reconfigurável

FPGA - Software - IDE de Desenvolvimento

Row	Inputs			Outputs		Comment
	x	y	c_{in}	c_{out}	s	
0	0	0	0	0	0	$0 + 0 + 0 = 00_2$
1	0	0	1	0	1	$0 + 0 + 1 = 01_2$
2	0	1	0	0	1	$0 + 1 + 0 = 01_2$
3	0	1	1	1	0	$0 + 1 + 1 = 10_2$
4	1	0	0	0	1	$1 + 0 + 0 = 01_2$
5	1	0	1	1	0	$1 + 0 + 1 = 10_2$
6	1	1	0	1	0	$1 + 1 + 0 = 10_2$
7	1	1	1	1	1	$1 + 1 + 1 = 11_2$



FPGA - Software - IDE de Desenvolvimento

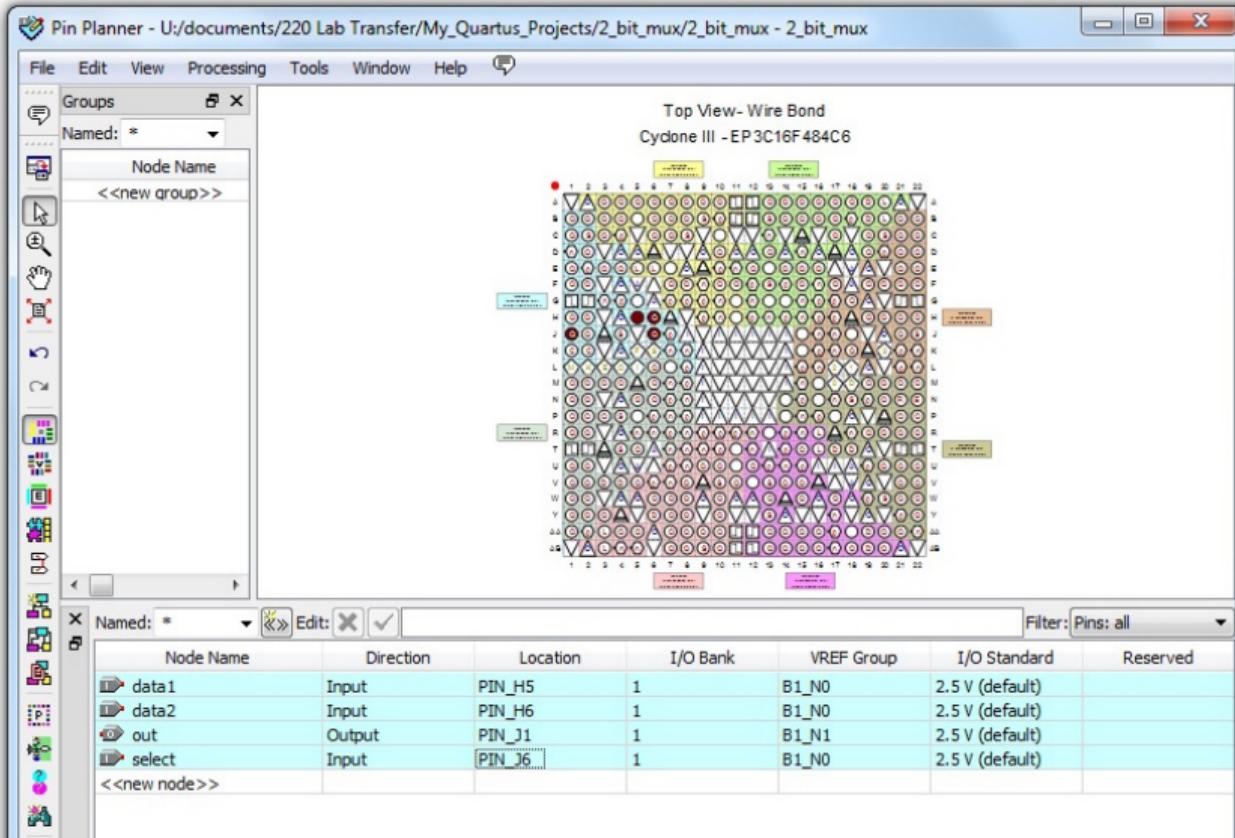


```
library ieee;
use ieee.std_logic_1164.all;

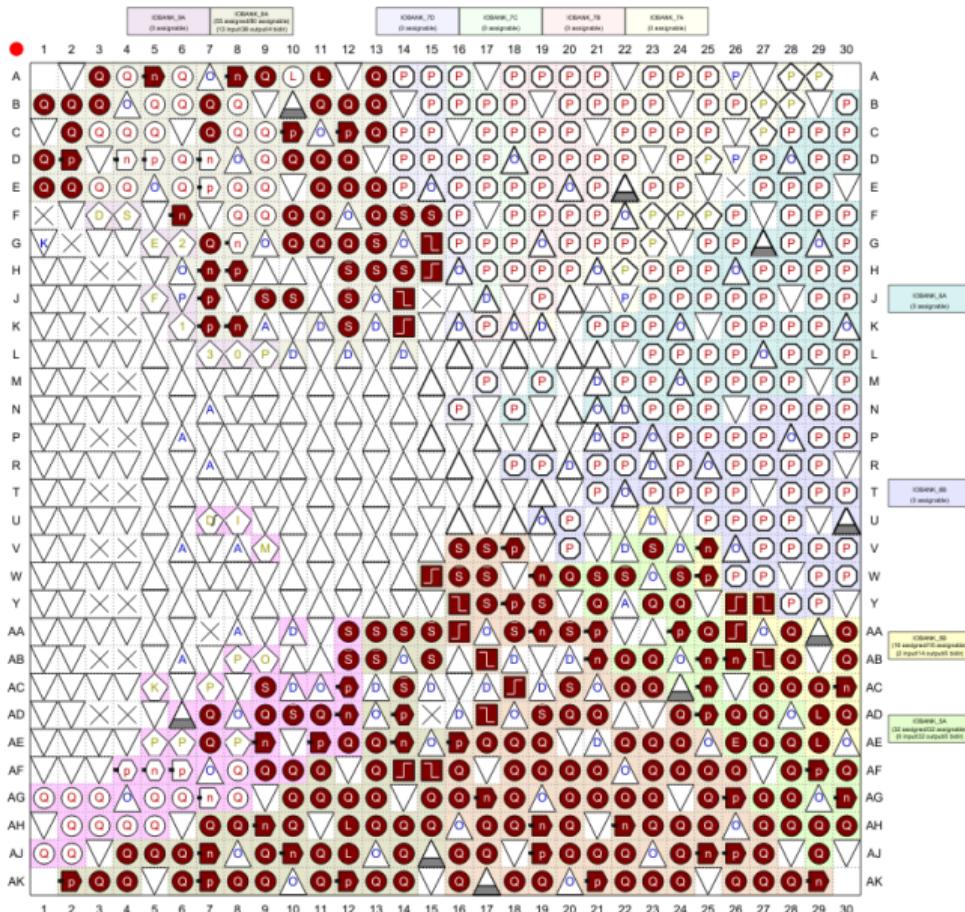
entity fadder is port (
    a, b : in std_logic;
    cin  : in std_logic;
    sum   : out std_logic;
    cout  : out std_logic);
end fadder;

architecture beh of fadder is
begin
    sum  <= a xor b xor cin;
    cout <= (a and b) or (b and cin) or (a and cin);
end beh;
```

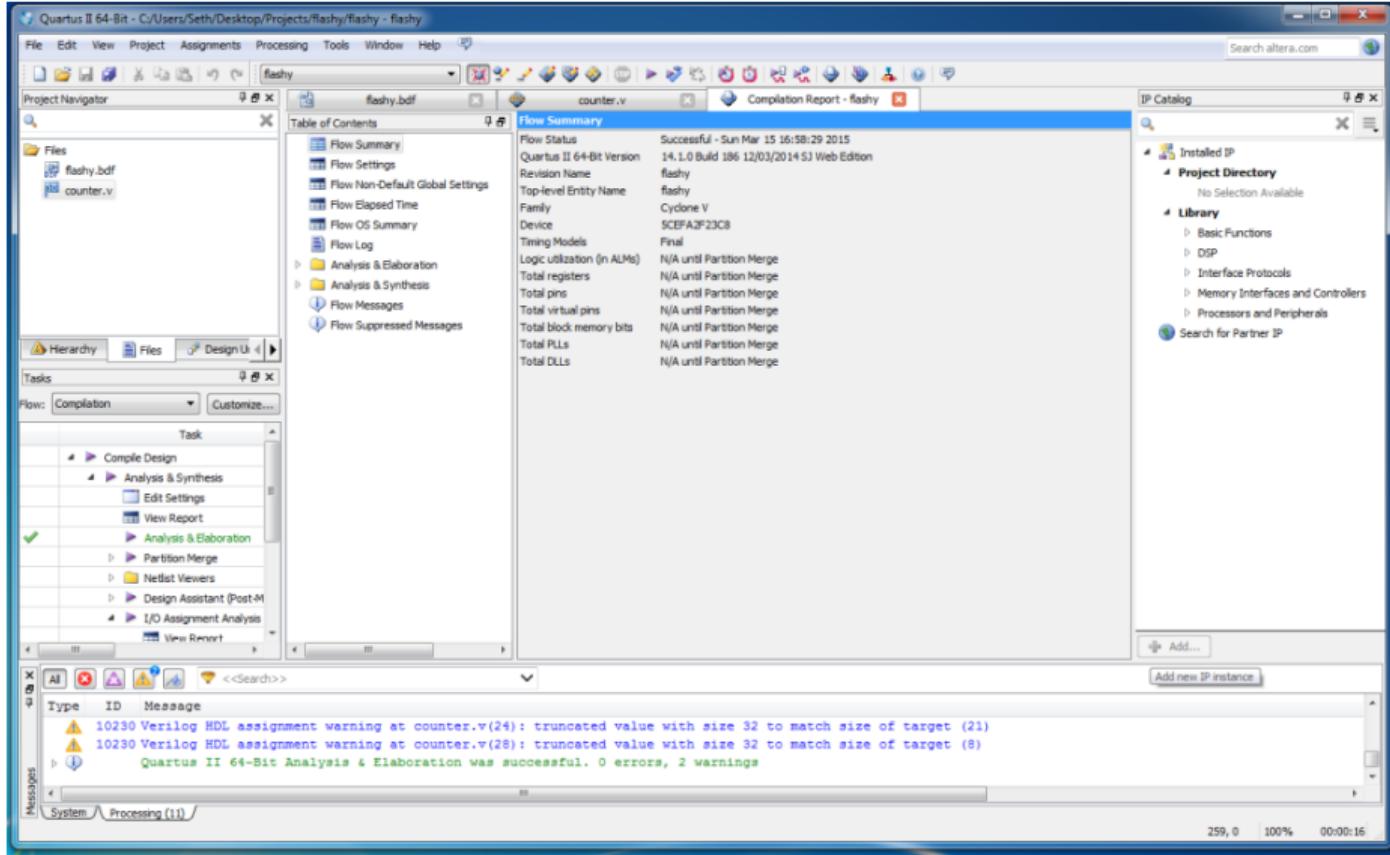
FPGA - Software - IDE de Desenvolvimento



Cyclone V - 5CSEMA5F31C6



FPGA - Software - IDE de Desenvolvimento



FPGA - Software - IDE de Desenvolvimento

Tasks

Flow: Compilation ▾ Customize...

	Task	⌚ Time
✓	▶ Compile Design	00:00:50
✓	▶ Analysis & Synthesis	00:00:24
✓	▶ Fitter (Place & Route)	00:00:14
✓	▶ Assembler (Generate programming files)	00:00:08
✓	▶ TimeQuest Timing Analysis	00:00:04
	▶ EDA Netlist Writer	
	▶ Program Device (Open Programmer)	

FPGA - Software - IDE de Desenvolvimento

Compilation Report - Flow Summary

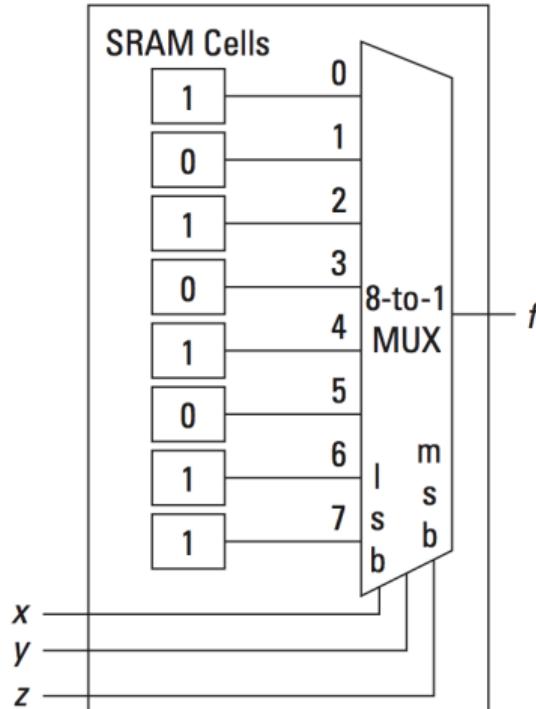
Flow Summary

Flow Status	Successful - Fri Aug 25 11:51:28 2006
Quartus II Version	6.1 Internal Build 142 08/21/2006 SJ Full Version
Revision Name	filtrf
Top-level Entity Name	filtrf
Family	Cyclone
Device	EP1C6F256C6
Timing Models	Final
Met timing requirements	Yes
Total logic elements	119 / 5,980 (2 %)
Total pins	22 / 185 (12 %)
Total virtual pins	0
Total memory bits	0 / 92,160 (0 %)
Total PLLs	0 / 2 (0 %)

FPGA - Software - IDE de Desenvolvimento

x	y	z	$xy + z'$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

(a)

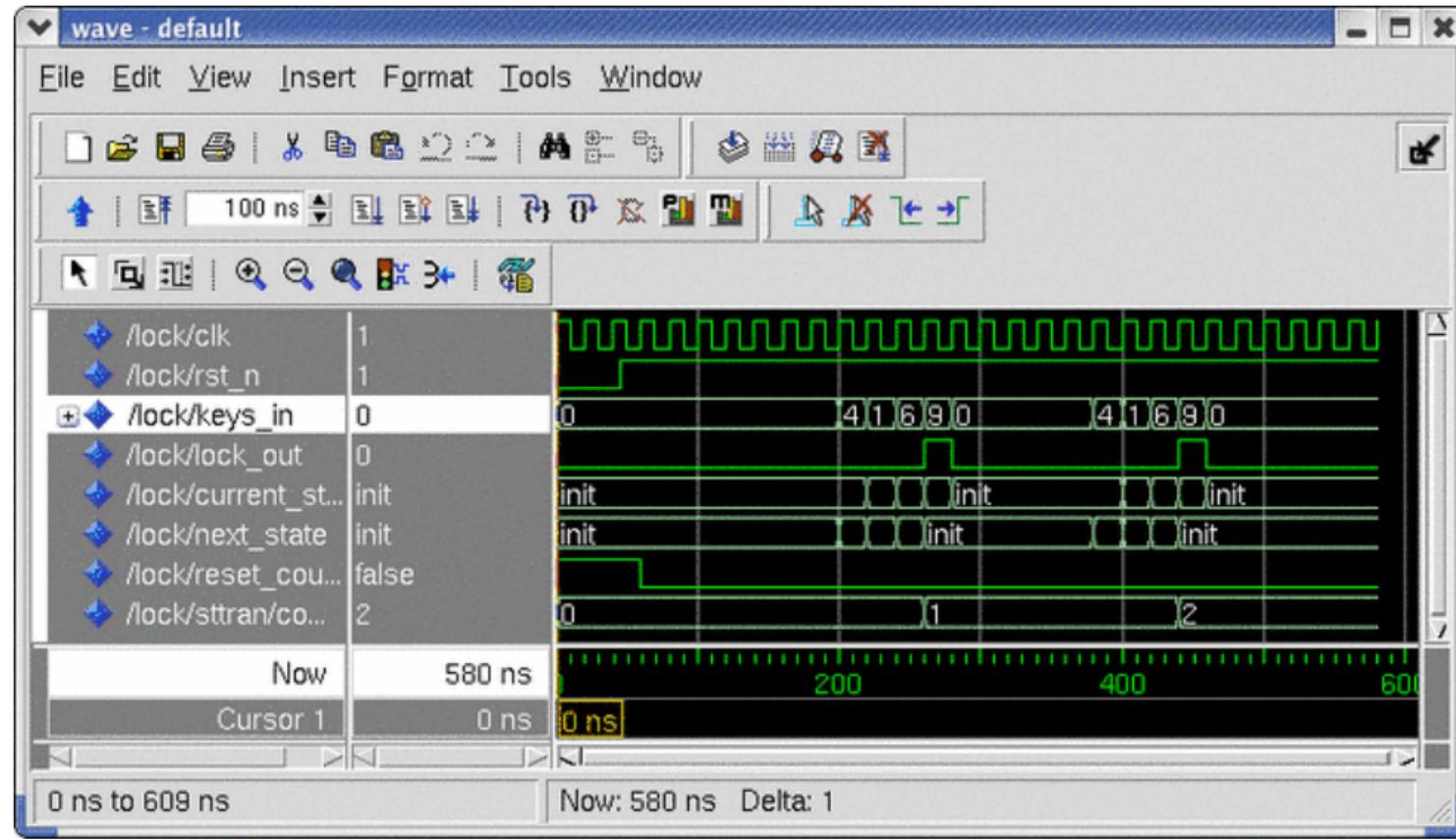


(b)



Figura: Gerador de Função. a) tabela verdade e b) Look-Up Table (LUT) com Flip-flops para

FPGA - Software - IDE de Simulação



FPGA - Programação

- A implementação na placa FPGA é feita por:
 - Linguagens de descrição chamadas de **Linguagens de Descrição de Hardware (HDL)**
 - São algumas linguagens: VHDL (*foto*), Verilog, ABEL, etc..
 - Ou por meio do desenvolvimento de **diagramas esquemáticos com ferramentas gráficas assistidas pelo computador** (*foto*).
 - Hoje, também é possível escrever por meio de **linguagens de alto nível** para como Handel-C, C-like ou mesmo o SystemC.
- Alguns conceitos de fundamentais sobre linguagens de descrição de *hardware*:
 - Estas linguagens permitem por exemplo *loops* infinitos ou portas lógicas com inúmeras entradas;
 - Essas são possível serem descritas em *software* mas impossível de implementar em *hardware*.

FPGA - Programação

The screenshot shows the Quartus II interface with the following details:

- Title Bar:** Quartus II - D:/Users/SVN/PROJECTS/UART/UART_VHDL_91/UART_TOP - UART_TOP - [uart_top.vhd]
- Menu Bar:** File, Edit, View, Project, Assignments, Processing, Tools, Window, Help
- Toolbar:** Various icons for file operations like Open, Save, Import, Export, etc.
- Project Navigator:** Shows the project structure:
 - Entity: Stratix III: EP3SL70F780C2
 - UART_TOP (selected)
 - Sub-components: InterfaceAva, Rx FIFO, Tx FIFO, UART DIV, UART RX CORE, and UART TX CORE.
- Hierarchy, Files, Design Units tabs:** Hierarchy is selected.
- Tasks:** Flow: Compilation, Task list:
 - Compile Design
 - Analysis & Synthesis
 - Filter (Place & Route)
 - Assembler (Generate programming files)
 - TimeQuest Timing Analysis
 - EDA Netlist Writer
 - Program Device (Open Programmer)
- VHDL Editor:** Displays the VHDL code for the UART_TOP entity.

```
entity UART_TOP is
  Port (
    -- Avalon interface signals
    AVA_CLR : in std_logic;
    AVA_CLK : in std_logic;
    AVA_ADDR : in std_logic_vector(2 downto 0);
    AVA_DAT_I : in std_logic_vector(31 downto 0);
    AVA_DAT_O : out std_logic_vector(31 downto 0);
    AVA_RD : in std_logic;
    AVA_WR : in std_logic;
    AVA_IRQ : out std_logic;

    -- UART Reference Clock and Clock Enable
    REF_CLK_EN : in std_logic;

    -- UART Signals INs and OUTs
    UART_RX_DI : in std_logic;
    UART_TX_DO : out std_logic
  );
end UART_TOP;

architecture behavioral of UART_TOP is
component InterfaceAva is
  Port (...
```
- Status Bar:** Type, Message

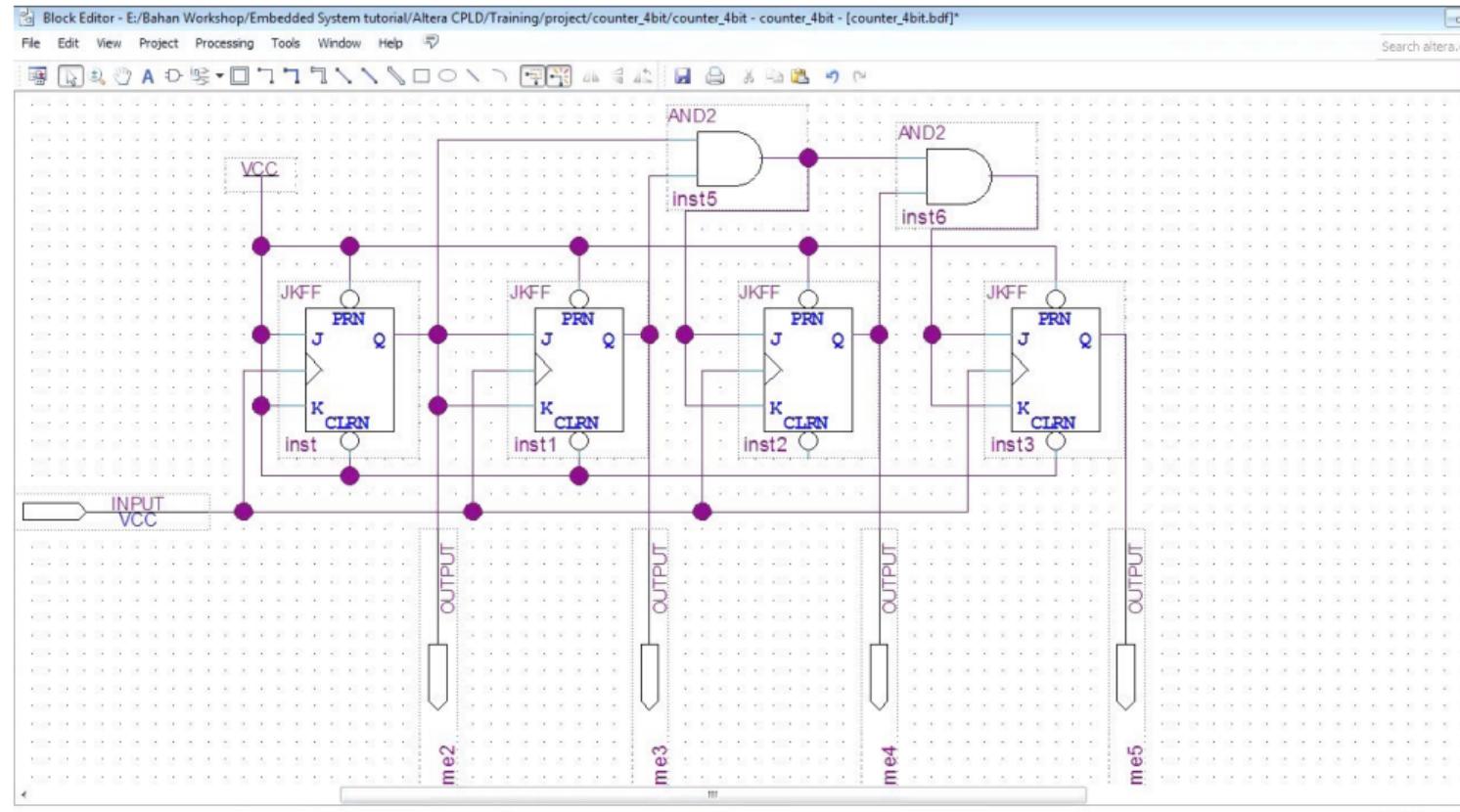
FPGA - Programação

- A implementação na placa FPGA é feita por:
 - Linguagens de descrição chamadas de **Linguagens de Descrição de Hardware (HDL)**
 - São algumas linguagens: VHDL, Verilog, ABEL, etc..
 - Ou por meio do desenvolvimento de **diagramas esquemáticos com ferramentas gráficas assistidas pelo computador**.
 - Hoje, também é possível escrever por meio de **linguagens de alto nível** para como Handel-C ou mesmo o SystemC.
- Alguns conceitos de fundamentais sobre linguagens de descrição de *hardware*:
 - Estas linguagens permitem por exemplo *loops* infinitos ou portas lógicas com inúmeras entradas;
 - Essas são possível serem descritas em *software* mas impossível de implementar em *hardware*.



UFOP

FPGA - Programação



FPGA - Programação

- A implementação na placa FPGA é feita por:
 - Linguagens de descrição chamadas de **Linguagens de Descrição de Hardware (HDL)**
 - São algumas linguagens: VHDL, Verilog, ABEL, etc..
 - Ou por meio do desenvolvimento de **diagramas esquemáticos com ferramentas gráficas assistidas pelo computador.**
 - Hoje, também é possível escrever por meio de **linguagens de alto nível** para como Handel-C ou mesmo o SystemC.
- Alguns conceitos de fundamentais sobre linguagens de descrição de *hardware*
 - Estas linguagens permitem por exemplo *loops* infinitos ou portas lógicas com inúmeras entradas;
 - Essas são possível serem descritas em *software* mas impossível de implementar em *hardware*.

FPGA - Programação - VHDL

- Linguagem para descrever o **comportamento e estrutura** de um sistema digital.
- A abreviatura VHDL significa VHSIC Hardware Description Language
 - Sendo que VHSIC significa Very High Speed Integrated Circuit.
- Ou seja, seu nome em português é **Linguagem de Descrição de Hardware de Circuitos Integrados com Altíssima Velocidade**.
- Ela tem propriedades suficientes para ser **independente tecnologicamente**
 - Ou seja, a mesma linguagem VHDL utilizada para descrever as tecnologias de hoje, poderá descrever as futuras tecnologias [4].

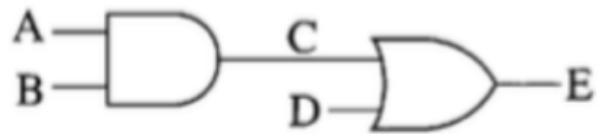
FPGA - Programação - VHDL

```
1 C <= A and B;  
2 E <= C or D;
```

FPGA - Programação - VHDL

```
1 C <= A and B;  
2 E <= C or D;
```

Figura: Altera Cyclone IV acoplado à placa da Terasic Altera DE2-115



FPGA - Bibliografia

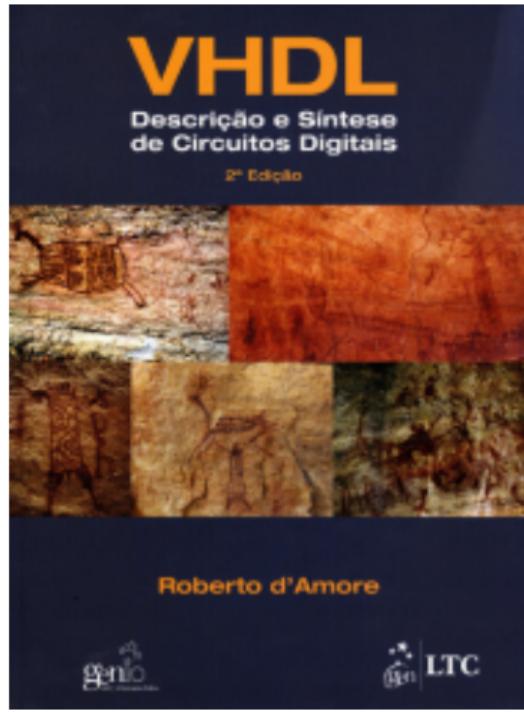


Figura: VHDL: Descrição e síntese de circuitos digitais [1].

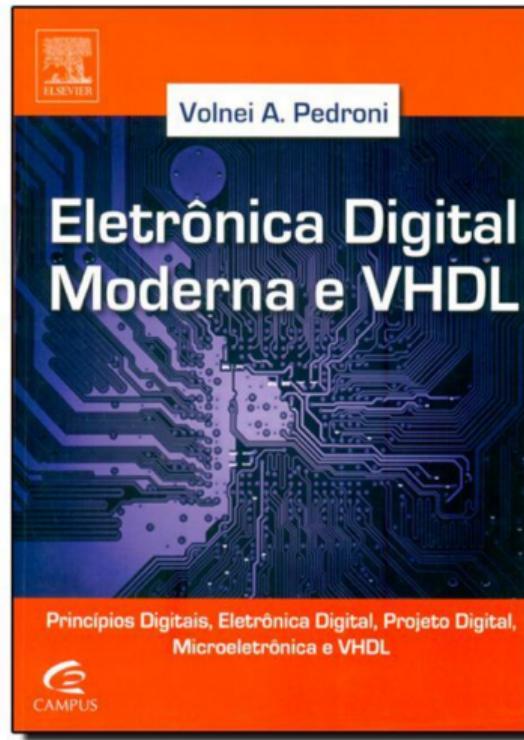


Figura: Eletrônica Digital Moderna e VHDL. [3].

FPGA - Bibliografia

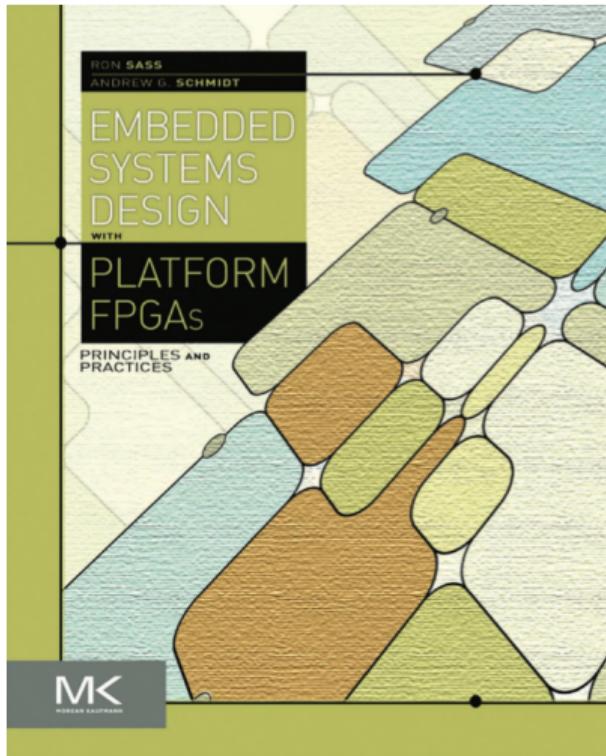


Figura: *Embedded Systems Design with Platform FPGAs, Principles and Practices* [5] .

FPGA - Vantagens e Desvantagens

■ Vantagens:

- Não é necessário uma ASIC pra realizar testes:
 - Com o uso do FPGA, é possível fazer os testes necessários prevendo erros;
- É possível adicionar mais recursos em sua placa para que o projeto tente simular o mais próximo possível do produto final
 - Câmeras, telas *touch* ...
- Amplamente usado para didática para a construção de circuitos integrados;
- Aplicação em diversas áreas inclusive áreas de grande importância como projetos no qual necessita de sistema críticos
 - A Nasa...
 - A Intel... (*foto*)

■ Desvantagens:

- Seu custo é bastante elevado para a utilização em projetos pequenos;
- Ele não possui capacidade de usar recurso analógicos;
- Seu desempenho entra em desvantagem em comparação com placas ASIC.





Official At Last: Intel Completes \$16.7 Billion Buy of Altera

by Barb Darrow

@gigabarb

DECEMBER 28, 2015, 1:33 PM EDT



Photograph by Justin Sullivan—Getty Images

The acquisition is a game changer for the microprocessor king.



FPGA - Vantagens e Desvantagens

■ Vantagens:

- Não é necessário uma ASIC pra realizar testes:
 - Com o uso do FPGA, é possível fazer os testes necessários prevendo erros;
- É possível adicionar mais recursos em sua placa para que o projeto tente simular o mais próximo possível do produto final
 - Câmeras, telas *touch* ...
- Amplamente usado para didática para a construção de circuitos integrados;
- Aplicação em diversas áreas inclusive áreas de grande importância como projetos no qual necessita de sistema críticos
 - A Nasa...
 - A Intel...

■ Desvantagens:

- Seu custo é bastante elevado para a utilização em projetos pequenos;
- Ele não possui capacidade de usar recurso analógicos;
- Seu desempenho entra em desvantagem em comparação com placas ASIC.



Sumário

1 Matemática Discreta

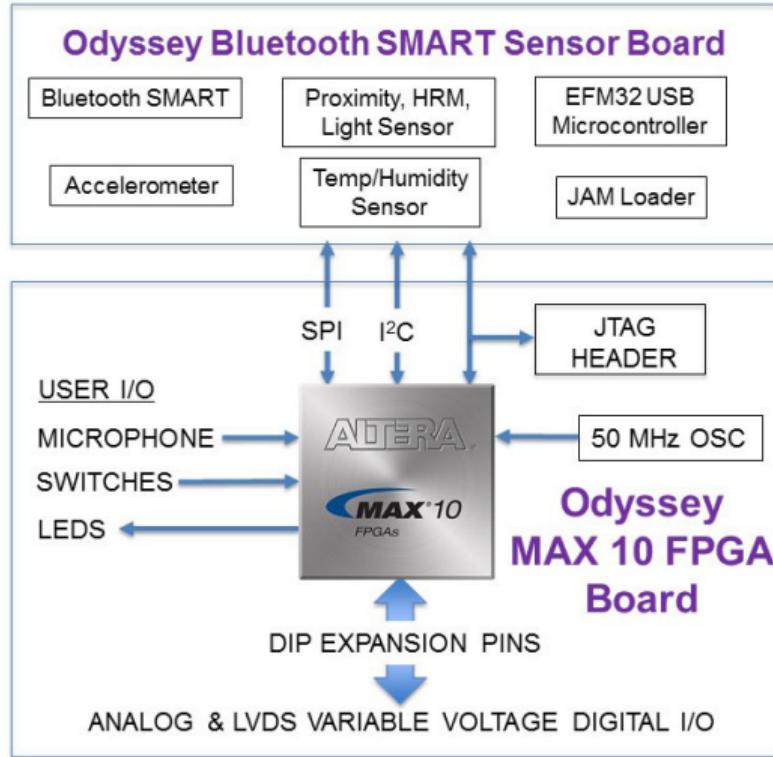
2 Dispositivos Lógico Programáveis

3 FPGA

4 UFOP - Laboratório iMobilis

5 Bibliografia

Kit Odyssey



■ Características:

- Altera FPGA 10M08;
- 8 mil elementos lógicos
- 169 pinos;
- 55 nm.

Kit Odyssey

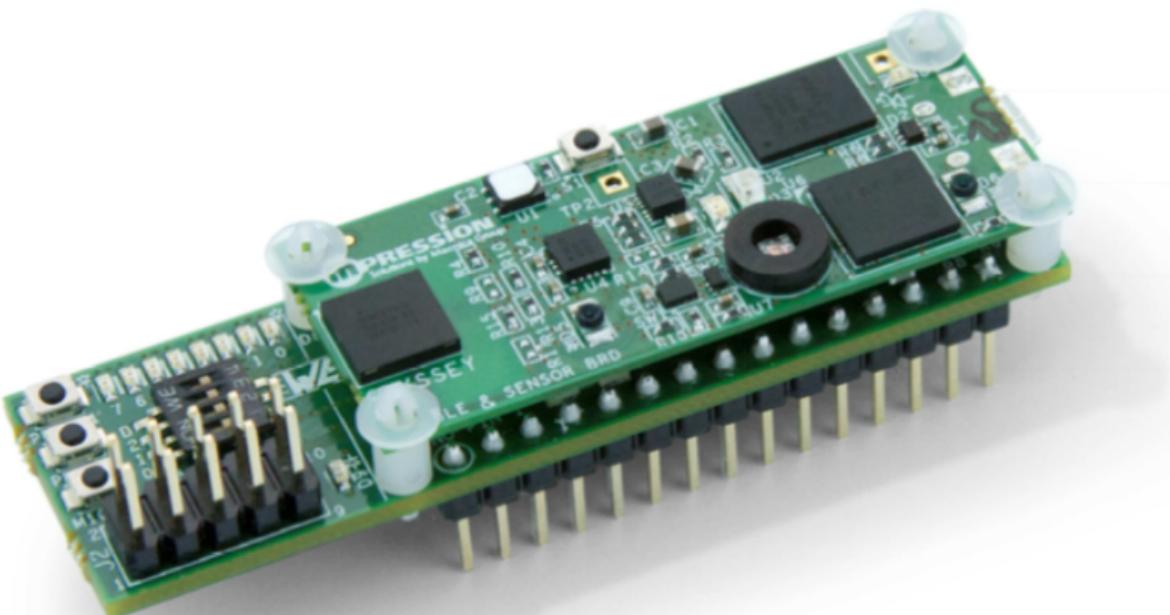


Figura: Odyssey IV.

Kit Mercurio IV

- Modelo EP4CE30F23 responsável possui cerca de 30 mil elementos lógicos, com 484 pinos de entrada e saída e utiliza *clock* de 50Mhz.

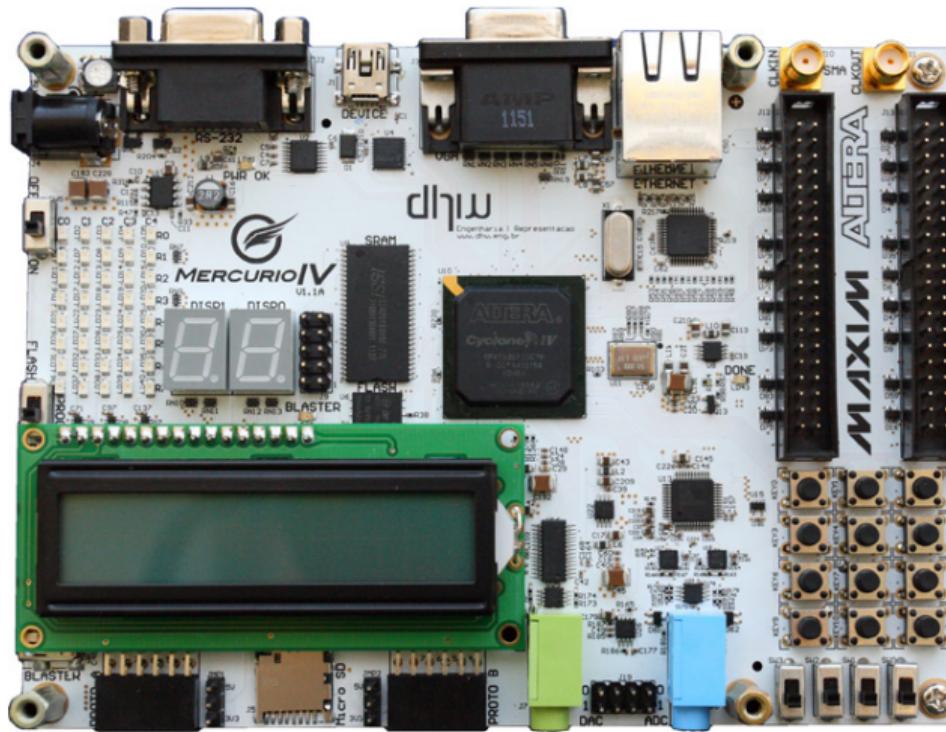
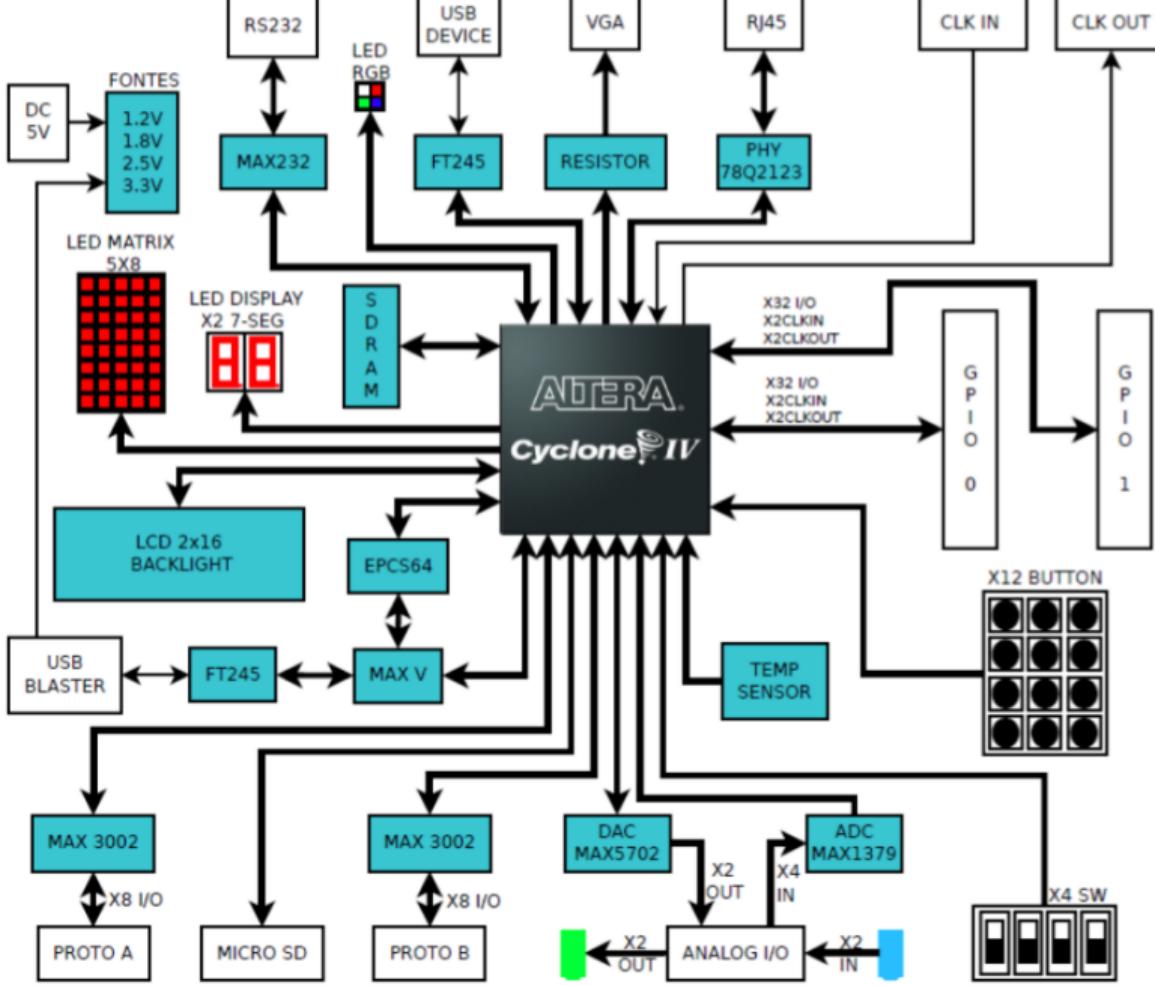


Figura: Mercurio IV.



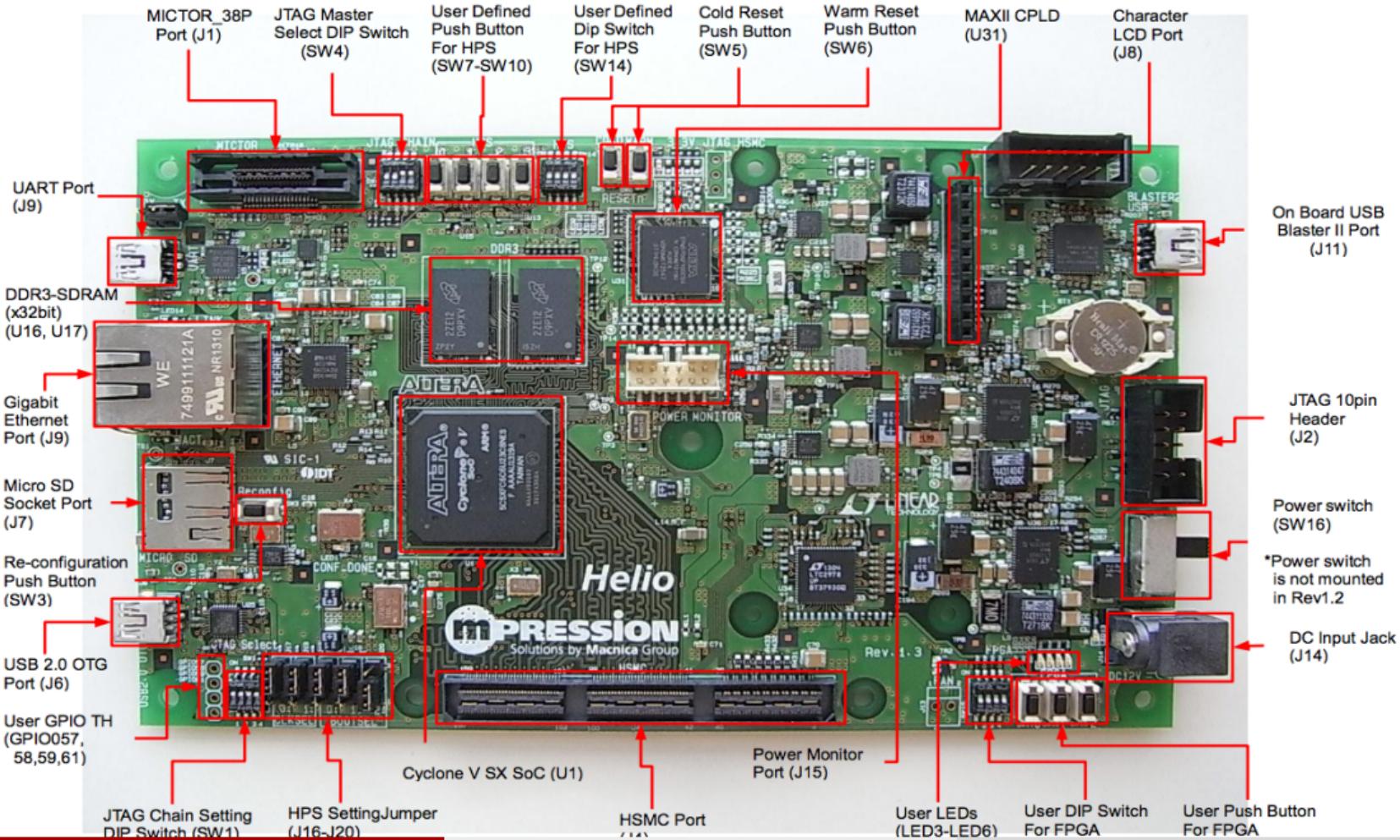
Kit Helio Cyclone V

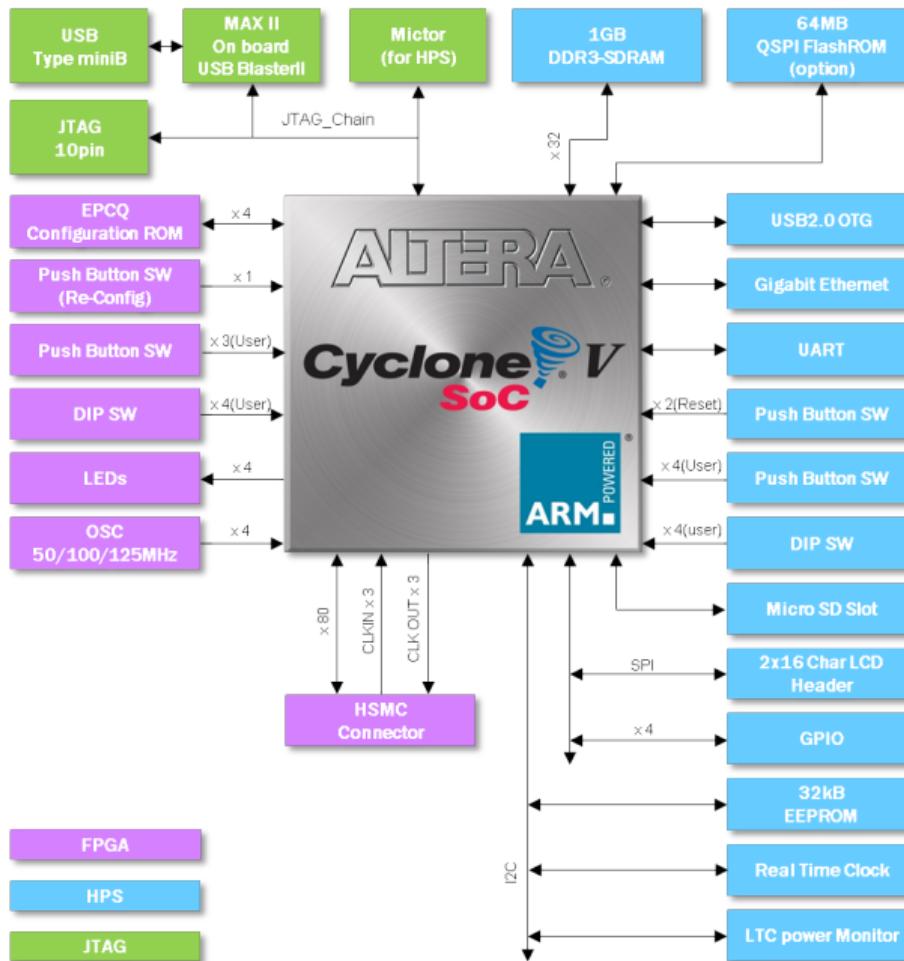
- Fabricado em tecnologia de silício de 28nm.
- Este FPGA integra num único dispositivo um sistema em *hardware* baseado no processador ARM Cortex A9 dual-core e mais um grande conjunto de periféricos.
- Características:
 - Altera Cyclone V SoC – 5CSXFC6C6U23C8NES / 5CSXFC5C6U23C7N
 - HSMC expansion connector
 - DDR3-SDRAM
 - Micro SD Card
 - Interfaces: 1Gb Ethernet, USB OTG, UART...
 - Yocto Linux BSP

Kit Helio Cyclone V



Figura: Componentes do Helio Cyclone V.





Digilent Arty

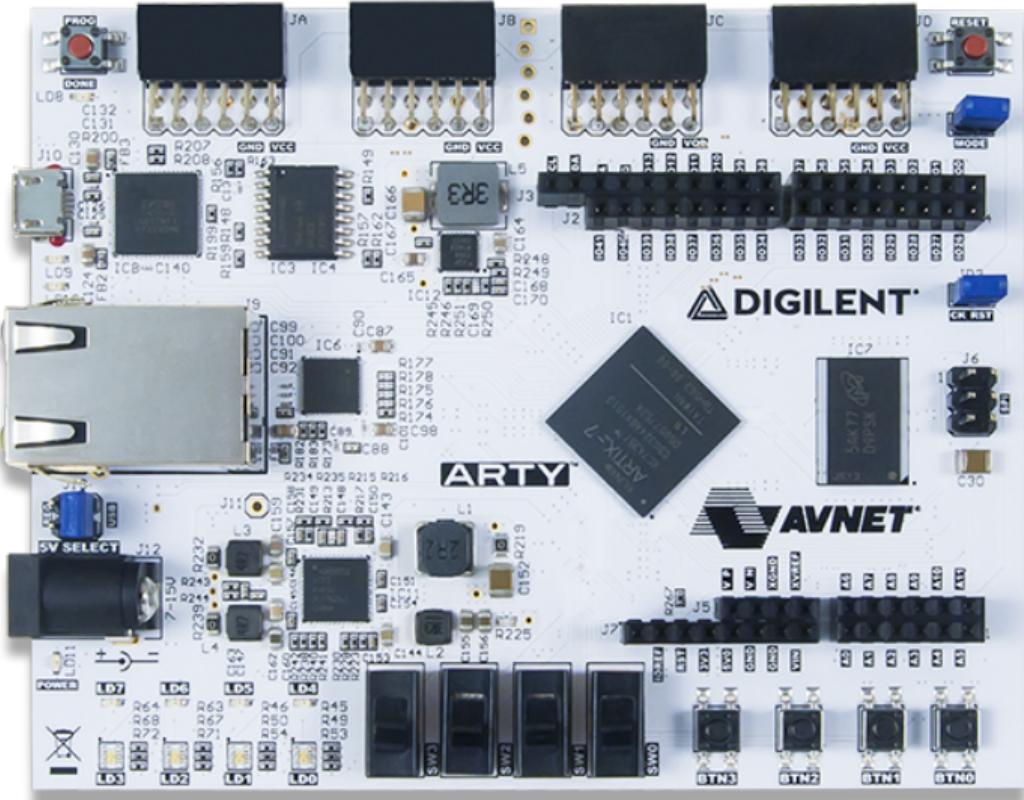
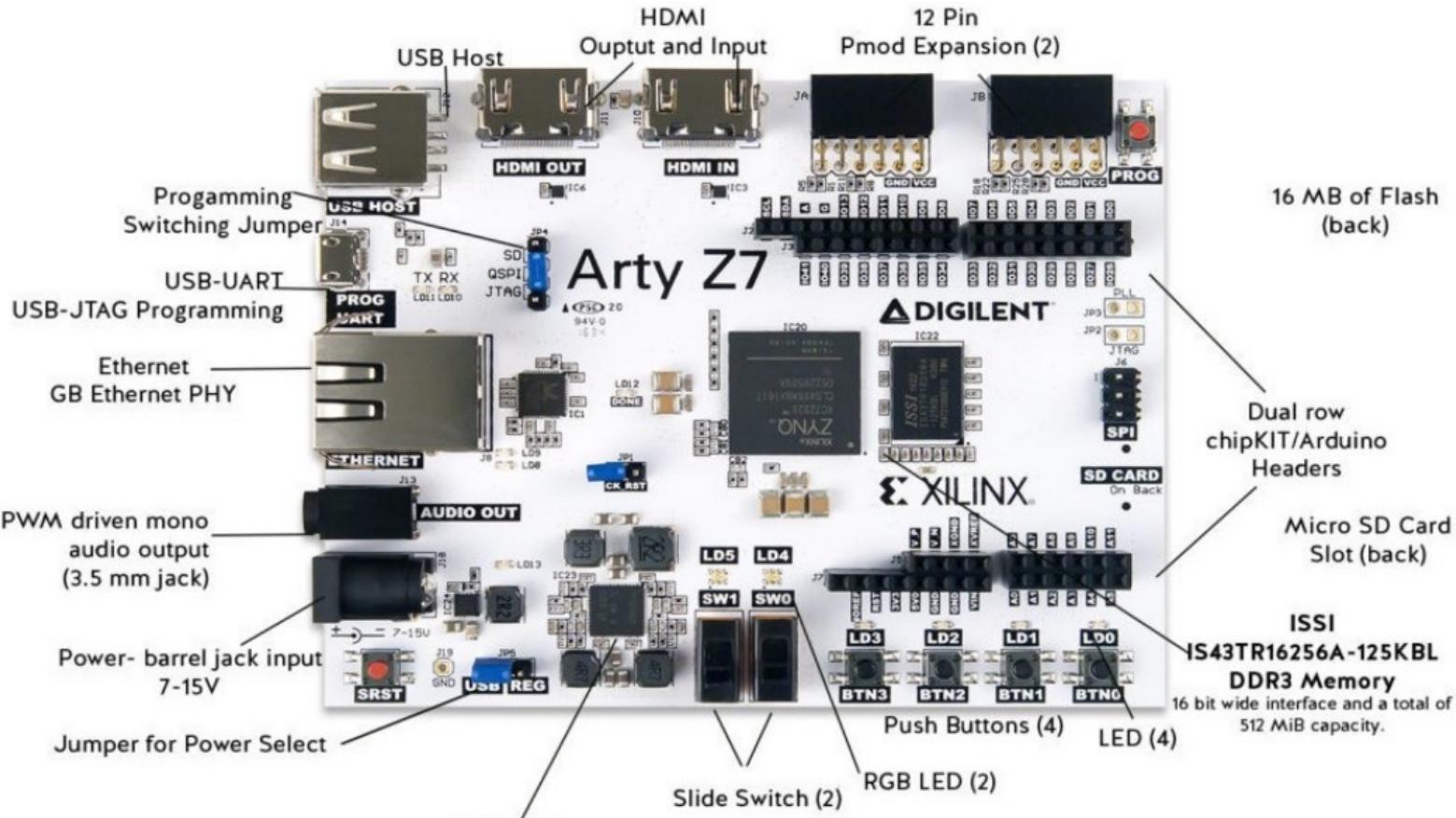


Figura: Componentes do Helio Cyclone V
Arquitetura de Computadores – Hardware Reconfigurável



Features	
<ul style="list-style-type: none"> • Programmable over JTAG and Quad-SPI Flash • On-chip analog-to-digital converter 	
Key Specifications	
FPGA Part #	XC7A35TICSG324-1L
Logic Slices	5,200
Block RAM	1,800 Kbits
DSP Slices	90
DDR3	256 MB @ 667 MHz
Internal clock	450 MHz+
Quad-SPI Flash	16 MB
Ethernet	10/100 Mbps
Connectivity and Onboard I/O	
Pmod Connectors	4
Shield Connector	Arduino / chipKIT compatible
Switches	4
Buttons	4
Reset LED	1
User LED	4
User RGB LED	4
Electrical	
Power	USB 7-15v (2.5mm coaxial) supply
Logic Level	3.3v
Physical	
Width	3.4 in
Length	4.3 in

Papilio Platform

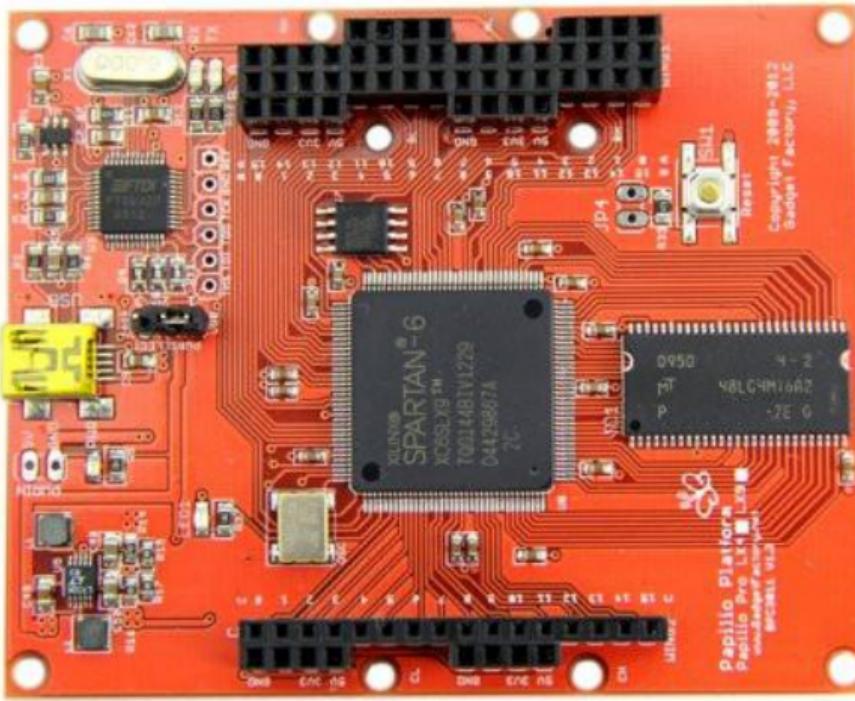


Figura: Esquemático do Papilio Platform.

Arquitetura de Computadores – Dispositivos Lógico Programáveis (*Hardware Reconfigurável*)

Arranjo de Portas Programável em Campo - FPGA

Rodolfo Labiapari Mansur Guimarães

rodolfolabiapari@decom.ufop.br

Lattes: <http://goo.gl/MZv4Dc>

Departamento de Computação – Universidade Federal de Ouro Preto
Ouro Preto - MG – Brasil

Última Atualização: 7 de maio de 2018



Sumário

1 Matemática Discreta

2 Dispositivos Lógico Programáveis

3 FPGA

4 UFOP - Laboratório iMobilis

5 Bibliografia



R. D' AMORE.

VHDL Descrição e Síntese de Circuitos Digitais.

Editora LTC, 2012.



V. Moreira.

Plataforma reconfigurável para ensino e pesquisa em laboratório de sistemas digitais a distância.

Anais do Simpósio . . . , (Sbie):542–551, 2008.



V. PEDRONI.

Eletrônica digital moderna e VHDL.

Campus, 1 edition, 2010.



C. Roth.

Digital systems design using VHDL.

CL Engineering, 2 edition, 1998.



R. Sass and A. Schmidt.

Embedded Systems Design with Platform FPGAs Principles and Practices.

Morgan Kaufmann, 1 edition, 2010.



UFOP



R. Tocci, N. Widmer, and G. Moss.

Sistemas digitais: princípios e aplicações.

Pearson, 11 edition, 2003.



Arquitetura de Computadores – Dispositivos Lógico Programáveis (*Hardware Reconfigurável*)

Arranjo de Portas Programável em Campo - FPGA

Rodolfo Labiapari Mansur Guimarães

rodolfolabiapari@decom.ufop.br

Lattes: <http://goo.gl/MZv4Dc>

Departamento de Computação – Universidade Federal de Ouro Preto
Ouro Preto - MG – Brasil

Última Atualização: 7 de maio de 2018