

Hardware Reconfigurável

Prof. Dr. Oscar Eduardo Anaconda Mosquera

oscar.mosquera@ufmt.br

4 de Agosto de 2025

Conteúdo

1 Objetivos

2 Introdução

3 Famílias da Altera

4 Fluxo de projeto

5 FPGA

6 Aplicação

7 DE0-nano

8 Bibliografia

Objetivos

- **Fluxo de Projeto em FPGA:**
 - ▶ Compreender o fluxo de projeto típico para desenvolver aplicações em FPGA.
 - ▶ Explorar as etapas desde a especificação do projeto até a implementação e teste em hardware.
- **FPGA: CLBs, LUTs, PIPs, IOBs:**
 - ▶ Entender os componentes principais de um FPGA, incluindo Blocos de Lógica Configurável (CLBs), Tabelas de Busca (LUTs), Pontos de Interconexão Programável (PIPps) e Blocos de Entrada/Saída (IOBs).
- **Módulos dos FPGAs Antigos e Modernos:**
 - ▶ Comparar e contrastar as arquiteturas de FPGAs antigos e modernos.
 - ▶ Identificar as melhorias e avanços nas capacidades de processamento, eficiência energética e recursos de integração.
- **SoC-FPGA:**
 - ▶ Explorar os conceitos de SoC-FPGA (System-on-Chip FPGA) e suas vantagens.
 - ▶ Compreender como a integração de FPGA com processadores de aplicativos em um único chip permite soluções altamente integradas e flexíveis.

Conteúdo

1 Objetivos

2 Introdução

3 Famílias da Altera

4 Fluxo de projeto

5 FPGA

6 Aplicação

7 DE0-nano

8 Bibliografia

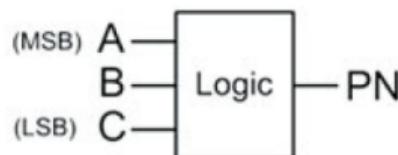
Classical digital design flow

Specifications

Projete um detetor de números primos para valores entre 0_{10} a 7_{10} . O circuito deve ser capaz de indicar um número primo com um atraso inferior a 200 ns.

Functional design

A funcionalidade e o comportamento do sistema são definidos em termos de entradas, saídas e operações necessárias para realizar uma tarefa específica.

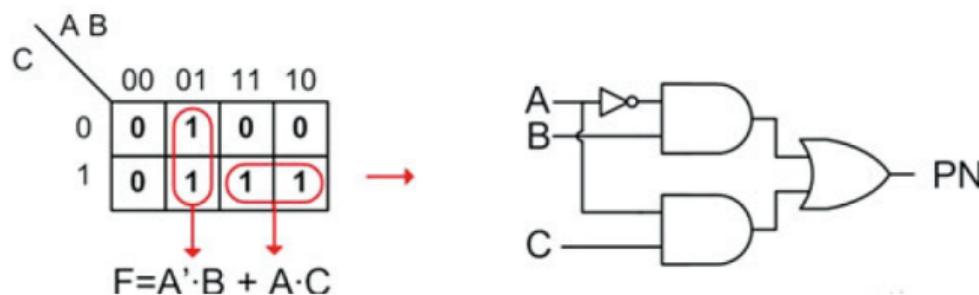


A	B	C	PN
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Classical digital design flow

Synthesis

Processo de transformar uma descrição de alto nível do comportamento funcional de um sistema em uma representação física ou lógica que pode ser implementada em hardware.

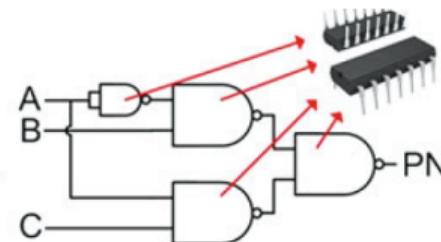


Classical digital design flow

Technology mapping

Nesta fase, a **netlist** gerada durante a síntese lógica é mapeada para os recursos específicos disponíveis no hardware de destino.

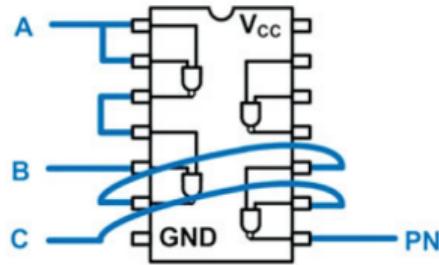
- It is decided that a 74HC logic family will be the most cost-effective technology for this design. To minimize the number of parts, the logic will be implemented with only NAND-gates.



Classical digital design flow

Place and route

Processo de transformar uma descrição de alto nível do comportamento funcional de um sistema em uma representação física ou lógica que pode ser implementada em hardware.

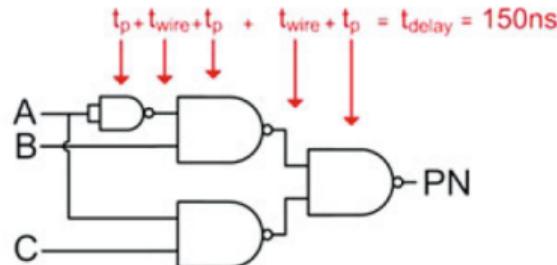


Classical digital design flow

Verification

Se concentra em garantir que o projeto atenda aos requisitos de funcionalidade, desempenho e confiabilidade estabelecidos durante a fase de especificação.

- Based on the layout, the wire delays are found. The delays of the gates are taken from the data sheet.

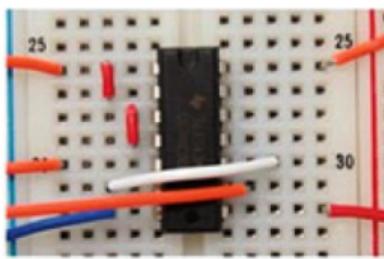


Classical digital design flow

Fabrication

O projeto em que o circuito integrado é efetivamente construído a partir do layout físico desenvolvido durante as etapas anteriores do projeto, como colocação e roteamento.

- The verified circuit is implemented in hardware.



Conteúdo

- 1 Objetivos
- 2 Introdução
- 3 Famílias da Altera
- 4 Fluxo de projeto
- 5 FPGA
- 6 Aplicação
- 7 DE0-nano
- 8 Bibliografia

Portafólio completo de soluções da Altera

ALTERA.



Nios® II

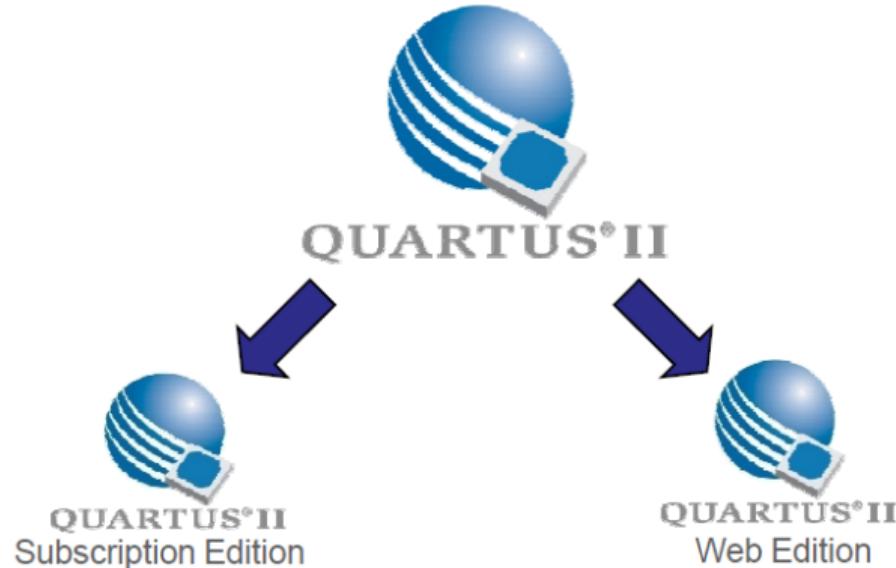
Embedded

Intellectual

Design

Development

Software Quartus II

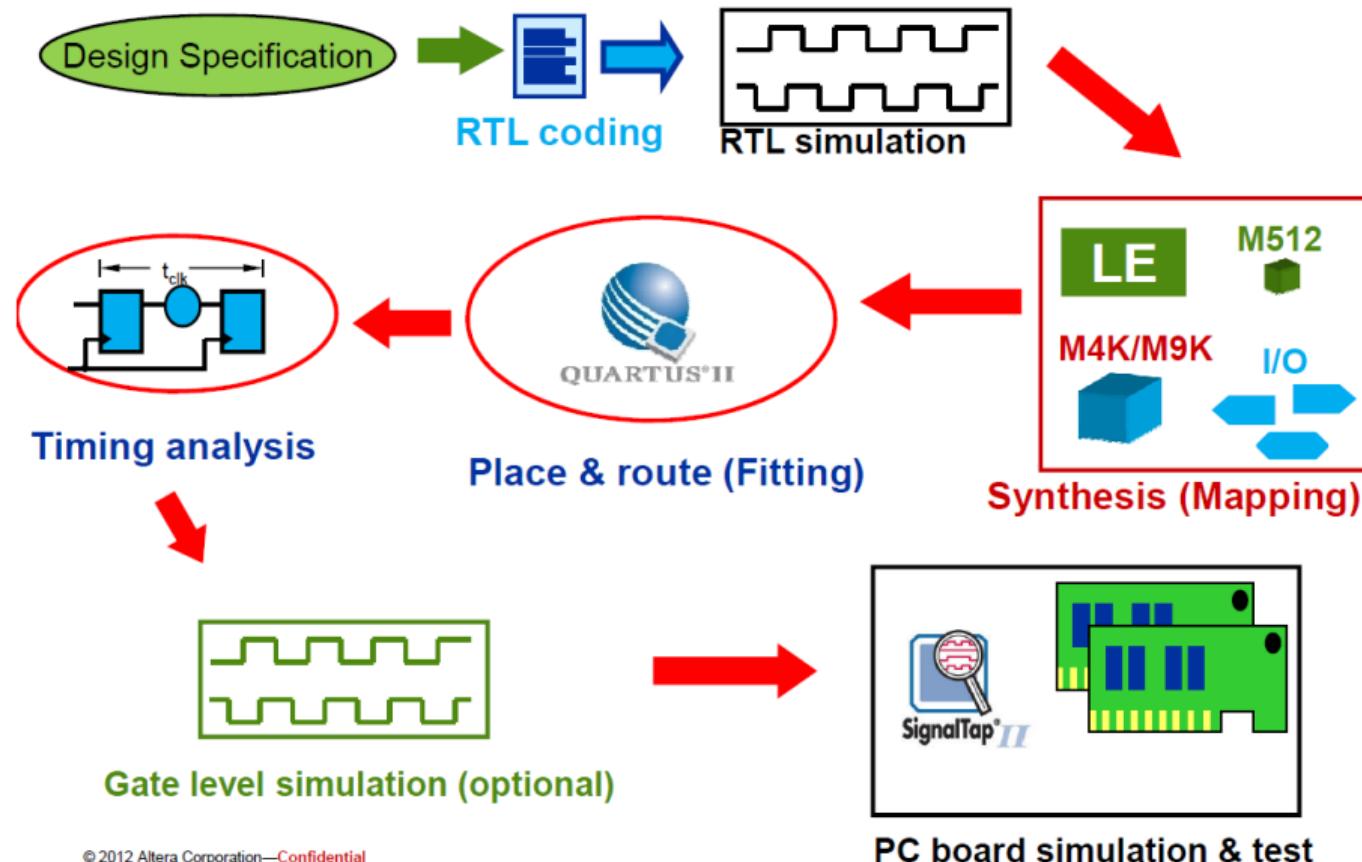


Devices Supported	All	Selected Devices
Features	100%	95%
Distribution	Internet & DVD	Internet & DVD
Price	Paid	Free

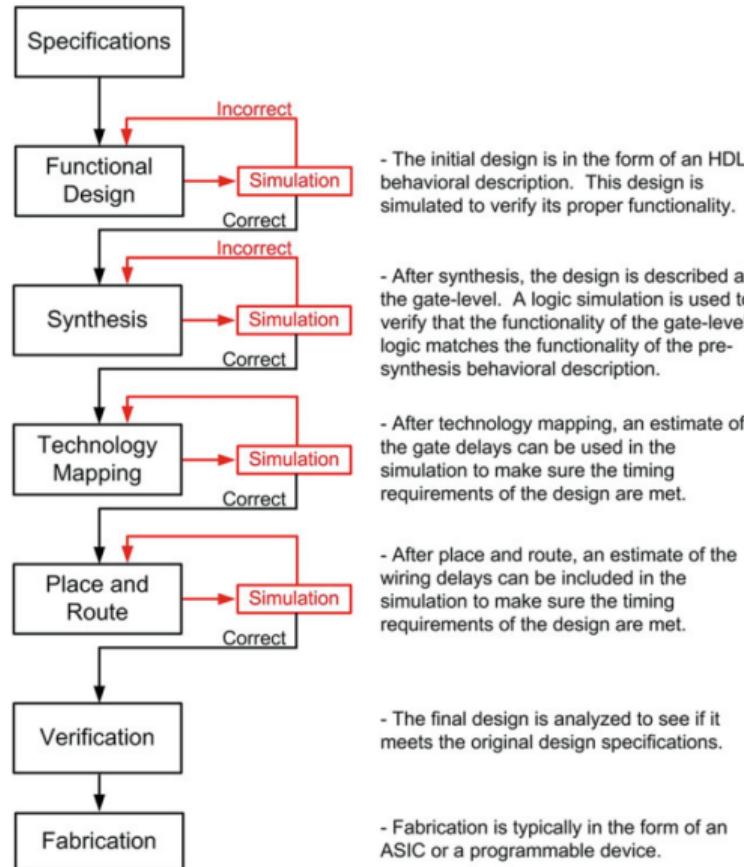
Conteúdo

- 1 Objetivos
- 2 Introdução
- 3 Famílias da Altera
- 4 Fluxo de projeto
- 5 FPGA
- 6 Aplicação
- 7 DE0-nano
- 8 Bibliografia

Typical PLD Design Flow



Modern Digital Design Flow



Conteúdo

1 Objetivos

2 Introdução

3 Famílias da Altera

4 Fluxo de projeto

5 FPGA

6 Aplicação

7 DE0-nano

8 Bibliografia

Field Programmable Gate Array (FPGA)

Um FPGA consiste em uma matriz de blocos de lógica programáveis (ou elementos lógicos) e uma rede de interconexão programável que pode ser usada para conectar qualquer elemento lógico a qualquer outro elemento lógico. Cada bloco lógico contém circuitos para implementar circuitos lógicos combinacionais arbitrários, além de um D-Flip-Flop e um multiplexador para direcionamento de sinal.

Em contraste com ASICs (Application-Specific Integrated Circuits), os FPGAs oferecem flexibilidade, permitindo que os usuários configurem a funcionalidade do hardware conforme necessário.

Field Programmable Gate Array (FPGA)

Arquitetura

- Composta por uma matriz de blocos de lógica configurável (CLBs) e uma rede de interconexão programável.
- CLBs contêm LUTs (Look-Up Tables), flip-flops, multiplexadores e outros elementos essenciais para implementar lógica combinacional e sequencial.
- A rede de roteamento permite conectar os CLBs para criar circuitos personalizados.

Field Programmable Gate Array (FPGA)

Programação

- A configuração do FPGA é armazenada em uma memória SRAM volátil.
- A programação em campo (field-programmability) permite atualizações e modificações de hardware mesmo após a fabricação.

Aplicações

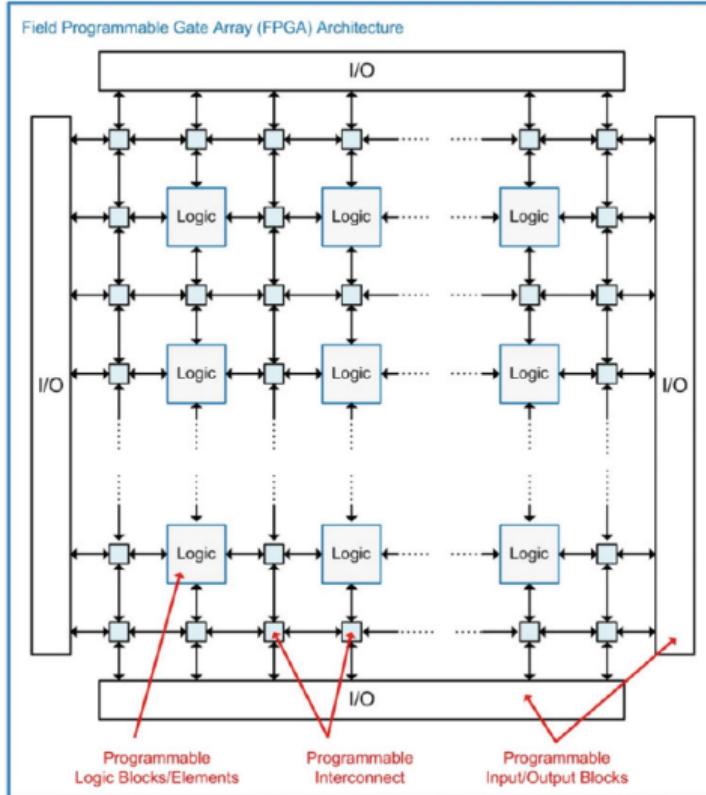
- Usado em uma variedade de aplicações, incluindo processamento de sinais digitais, comunicação, computação de alto desempenho, controle industrial, e muito mais.
- Amplamente adotado em prototipagem rápida, design de sistemas embarcados e prototipagem de ASICs.

Field Programmable Gate Array (FPGA)

Vantagens

- Flexibilidade e reconfigurabilidade.
- Custos mais baixos em comparação com ASICs.
- Consumo de energia potencialmente maior do que soluções ASIC.

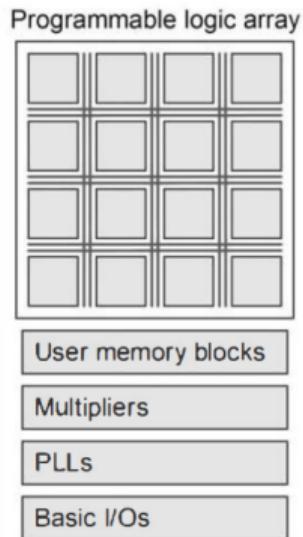
Field Programmable Gate Array (FPGA)



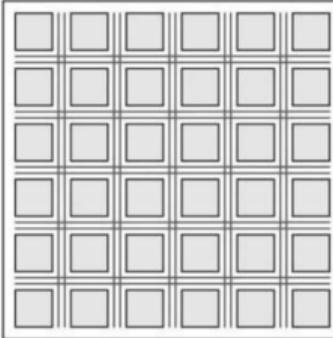
SoC-FPGA

É uma integração de um FPGA com uma ou mais unidades de processamento, como um processador de aplicação, processador de sinal digital (DSP) ou processador, em um único chip. Isso combina a flexibilidade de programação do FPGA com o poder de processamento de um processador, permitindo a implementação de sistemas complexos e customizados em um único dispositivo.

FPGAs hoje



Programmable logic array (up to 1.7M macrocells,
1.7M LUTs, and 3.4M flip-flops)



DDR memory interface (DDR2 to DDR4-2666 Mbps)

High Bandwidth Memory, HBM (highest BWavailable)

Transceivers (>100, NRZ and PAM4, up to 58 Gbps)

Hard blocks for PCIe, Ethernet, and other interfaces

Hard ARM processor cores (dual, quad)

High-speed LVDS I/Os (>500, 1.6 Gbps)

RF ADCs (12 and 14 bits, 2 to 10 Gsps)

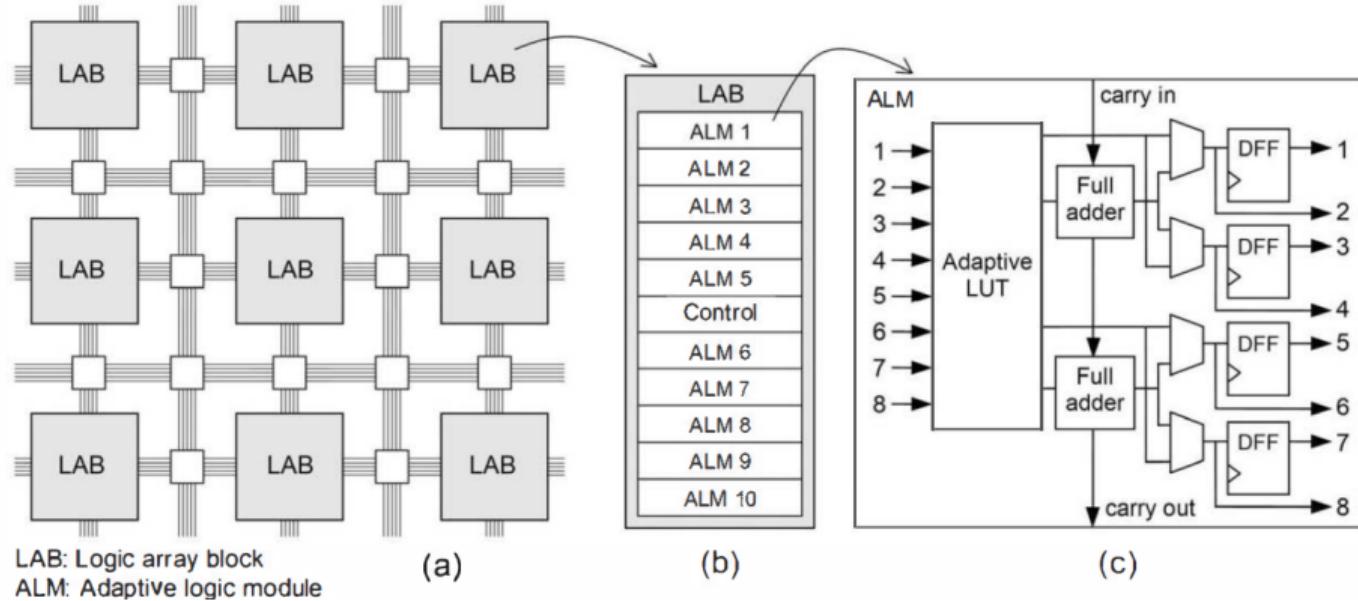
Advanced clock management tiles

Large number of user pins (>1000)

(b)

FPGAs hoje

Intel architecture for the logic part of Stratix 10 and other FPGAs.



Softcore e Hardcore nos FPGAs

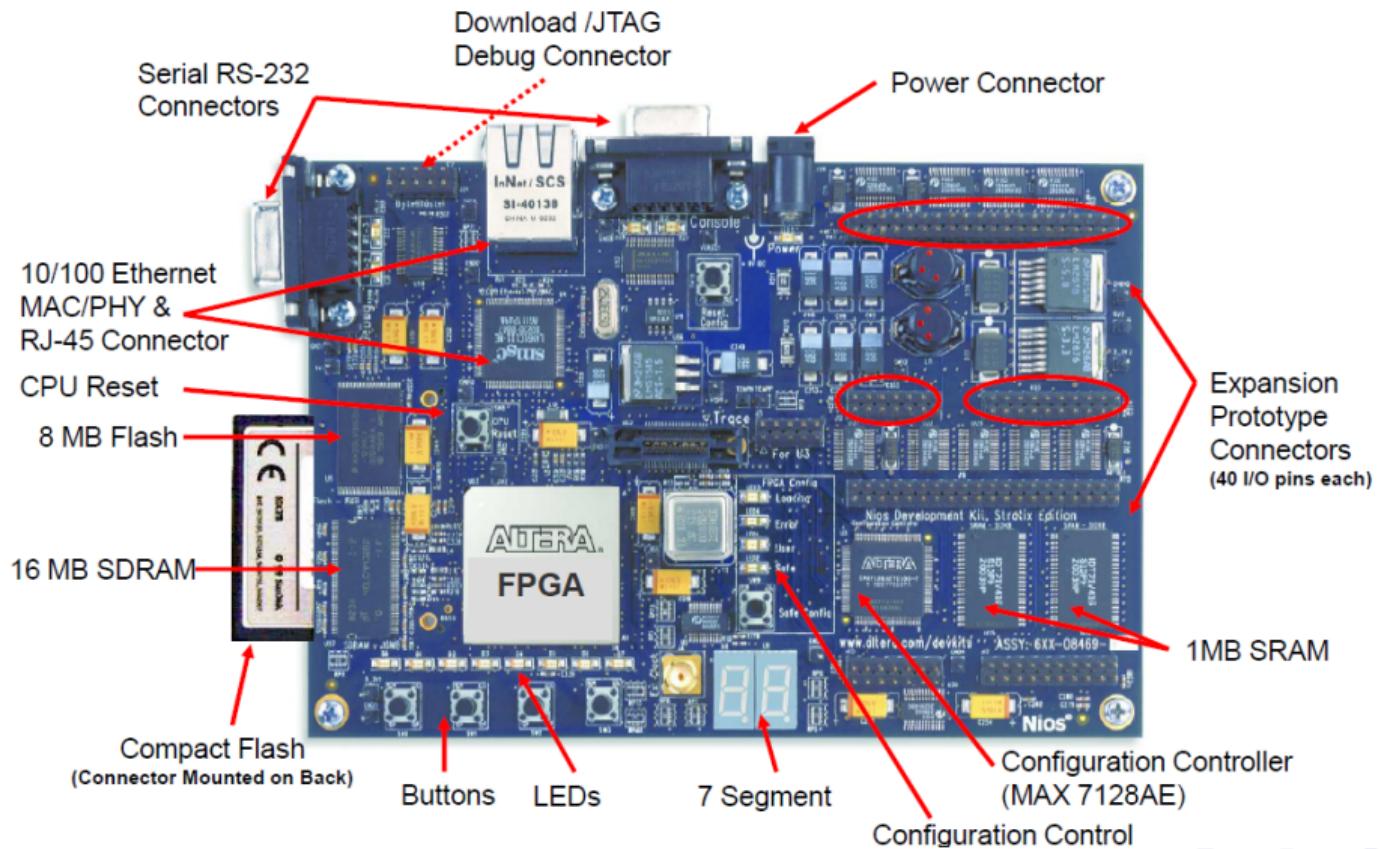
Softcore

Um processador softcore é um processador que é implementado em uma FPGA por meio de configuração lógica programável. Esses processadores são altamente flexíveis e podem ser facilmente adaptados às necessidades de uma aplicação específica. Exemplos incluem processadores como o NIOS II da Intel (anteriormente Altera) e o MicroBlaze da Xilinx.

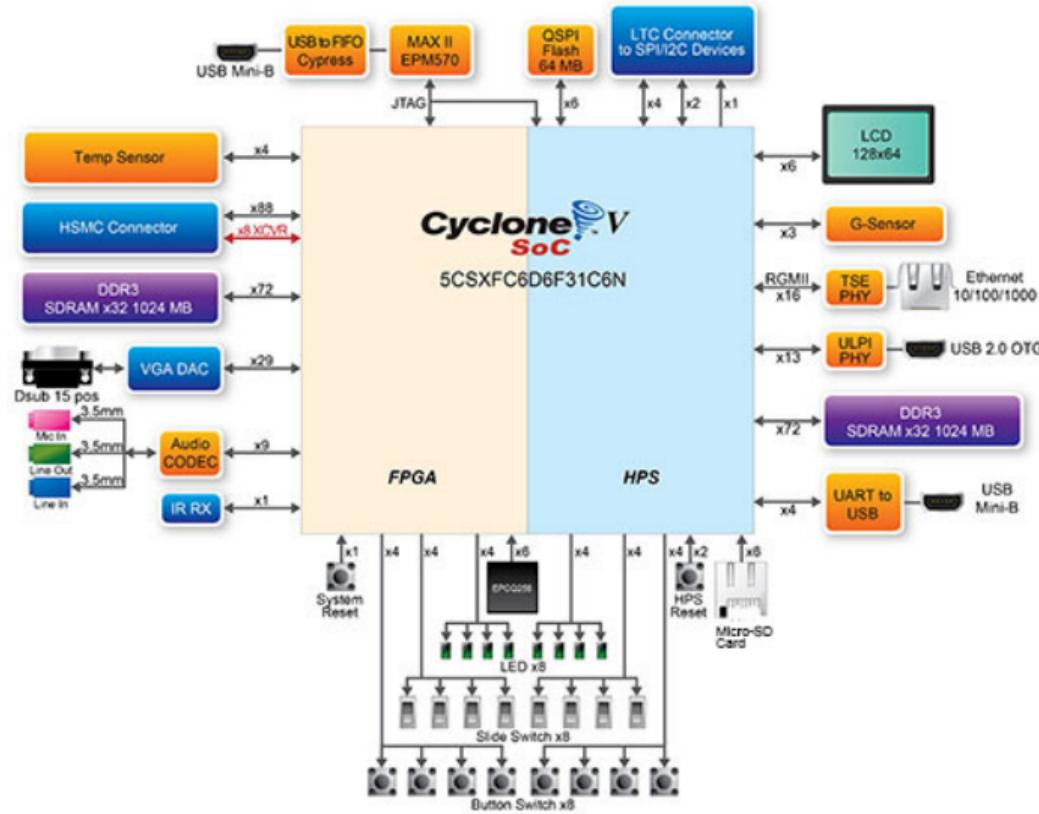
Hardcore

Um processador hardcore é um processador que é integrado diretamente na estrutura física do FPGA durante o processo de fabricação. Esses processadores são otimizados para desempenho e eficiência em termos de área e consumo de energia. Geralmente, eles oferecem um desempenho mais previsível e consistente em comparação com os processadores softcore. Exemplos incluem os núcleos ARM Cortex-A9 ou Cortex-A53 encontrados em alguns SoC-FPGAs.

Exemplo de um kit de desenvolvimento



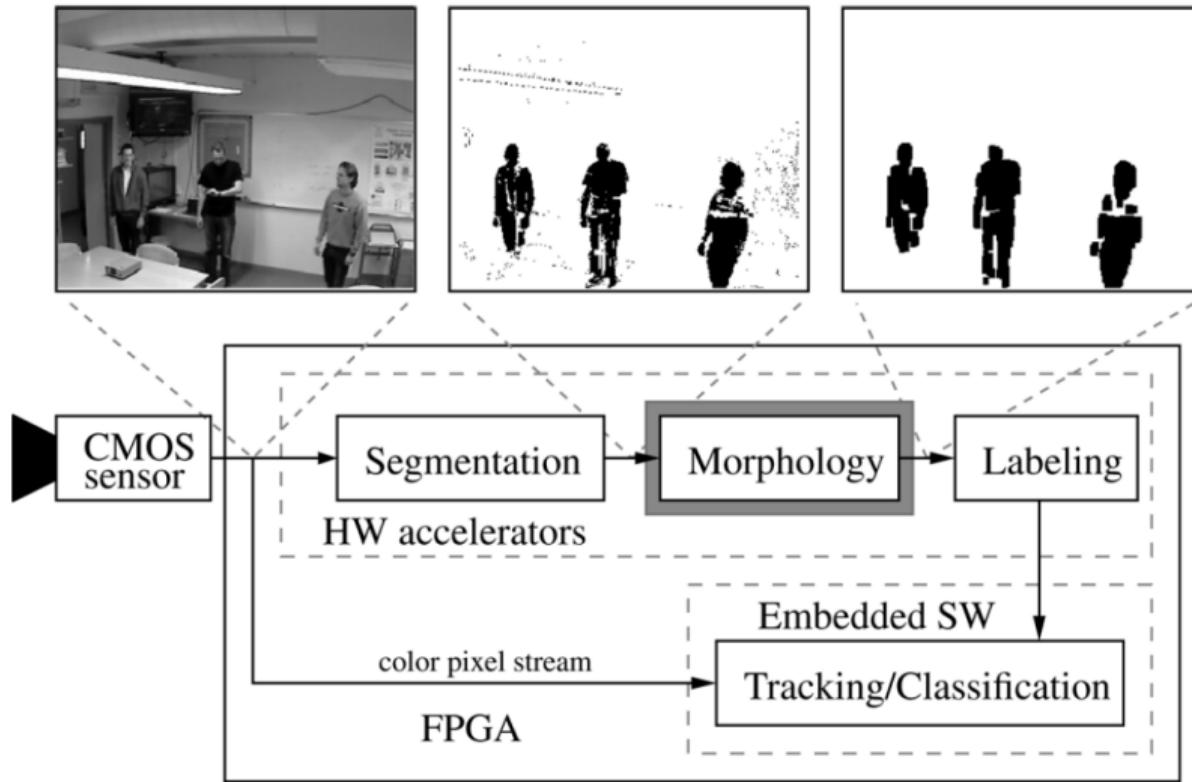
SoCkit



Conteúdo

- 1 Objetivos
- 2 Introdução
- 3 Famílias da Altera
- 4 Fluxo de projeto
- 5 FPGA
- 6 Aplicação
- 7 DE0-nano
- 8 Bibliografia

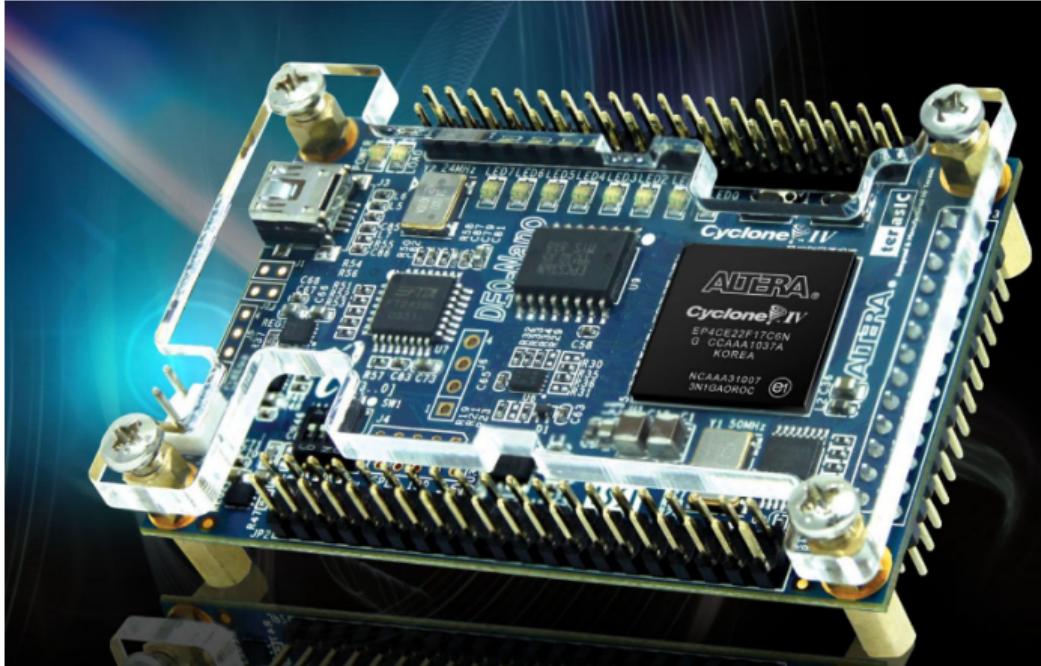
Aplicação



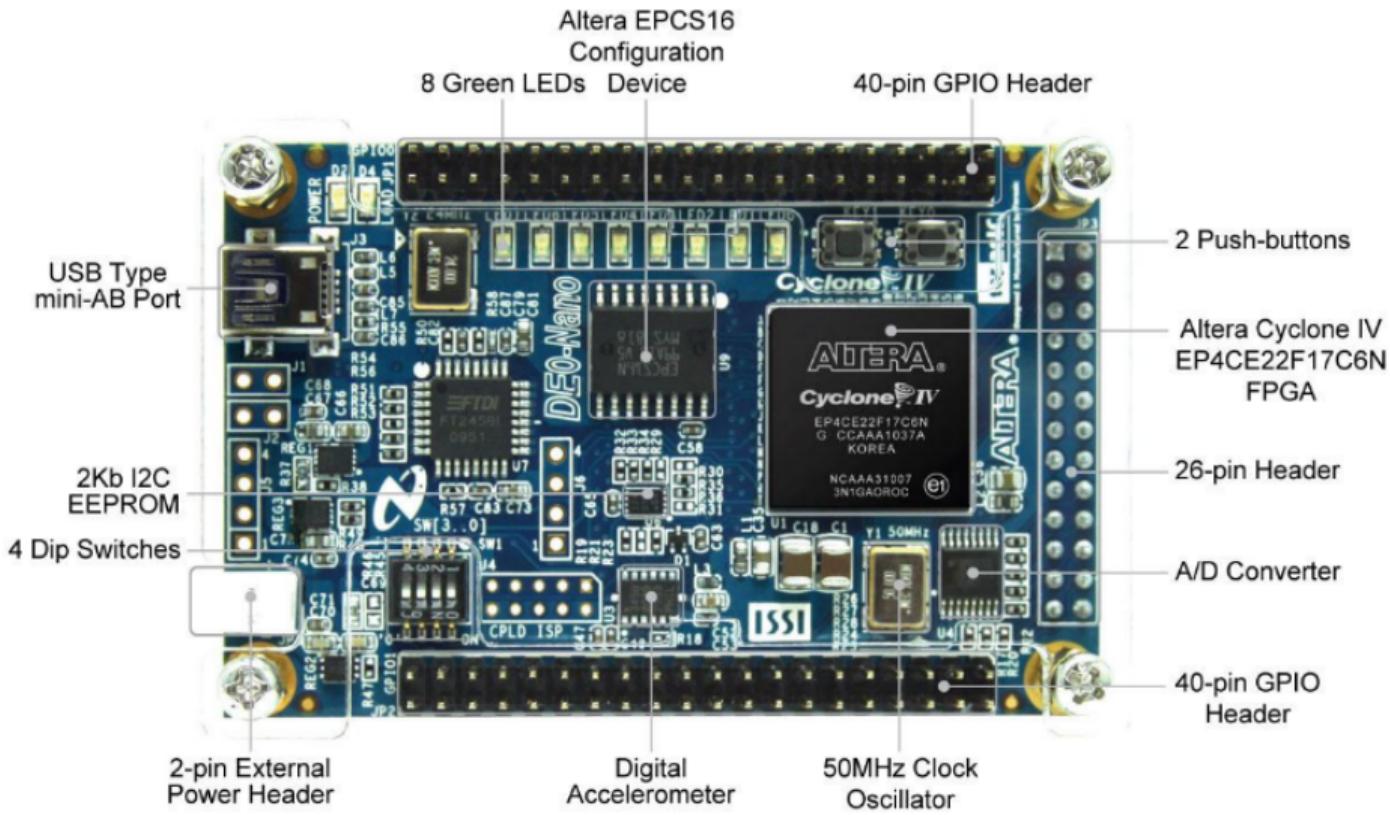
Conteúdo

- 1 Objetivos
- 2 Introdução
- 3 Famílias da Altera
- 4 Fluxo de projeto
- 5 FPGA
- 6 Aplicação
- 7 DE0-nano
- 8 Bibliografia

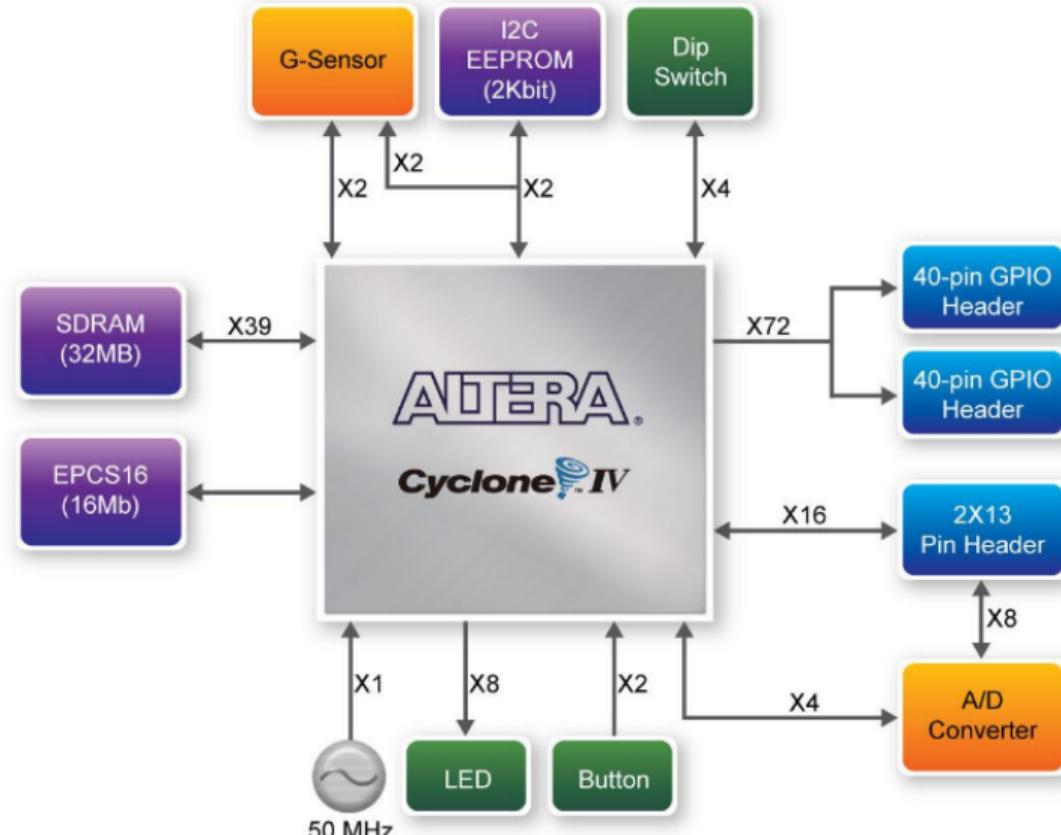
FPGA DE0-nano



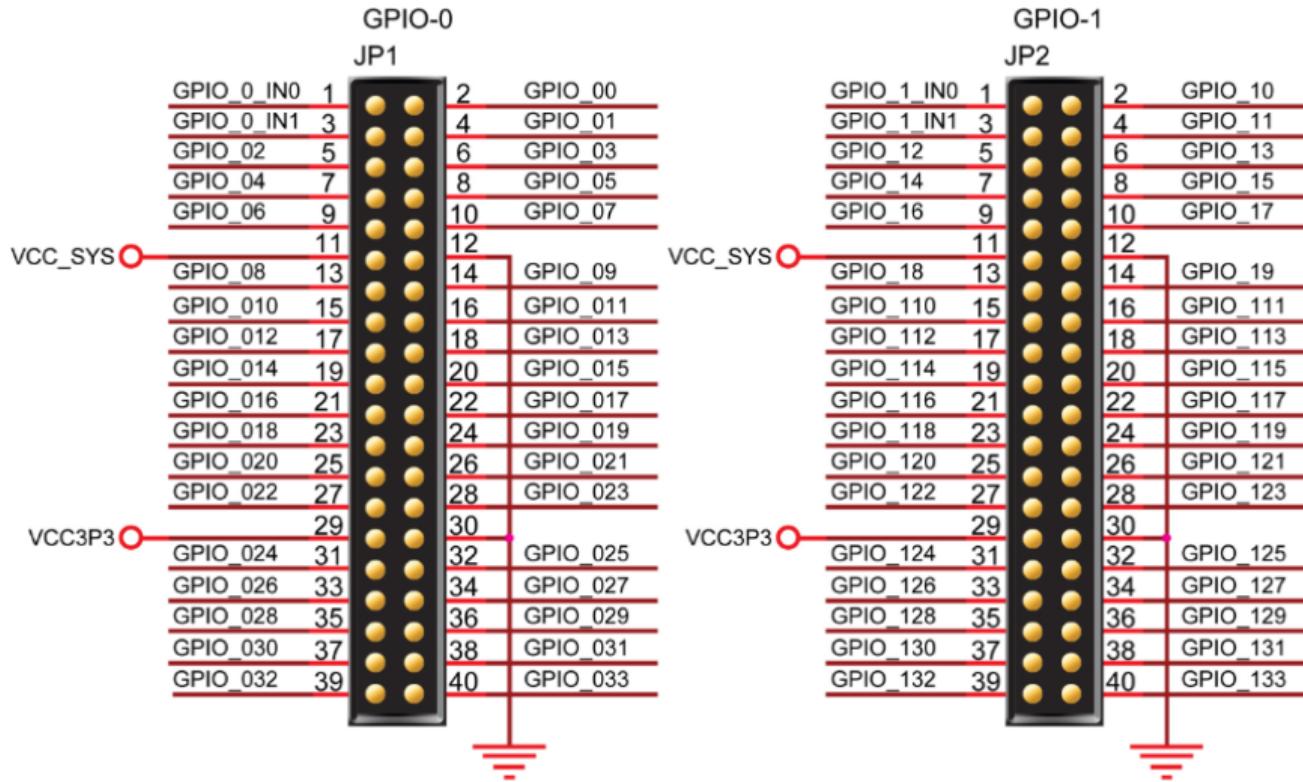
FPGA DE0-nano



FPGA DE0-nano



FPGA DE0-nano saídas/entradas digitais



FPGA DE0-nano alimentação



Conteúdo

1 Objetivos

2 Introdução

3 Famílias da Altera

4 Fluxo de projeto

5 FPGA

6 Aplicação

7 DE0-nano

8 Bibliografia

Bibliografia

- Pedroni, V.A. Circuit Design with VHDL. MIT Press, 2004
- LaMeres, B. J. (2023). Introduction to Logic Circuits & Logic Design with VHDL. Suíça: Springer International Publishing.