

# Microprocessadores e Microcontroladores (27146)



FPGAs e ferramentas CAD

---

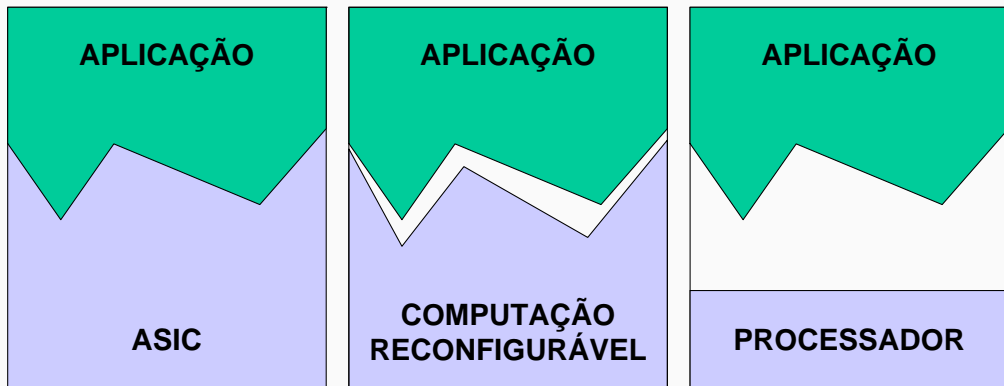
Prof. Ricardo Menotti (menotti@ufscar.br)

Atualizado em: 26 de abril de 2021

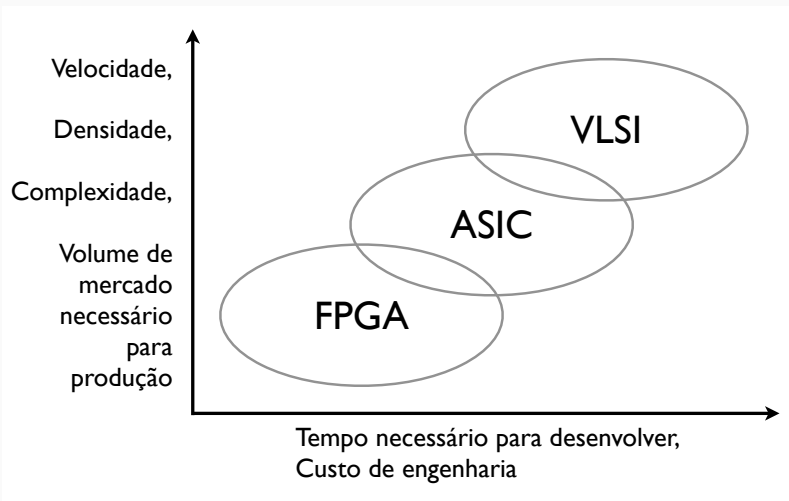
**Departamento de Computação**

Centro de Ciências Exatas e de Tecnologia

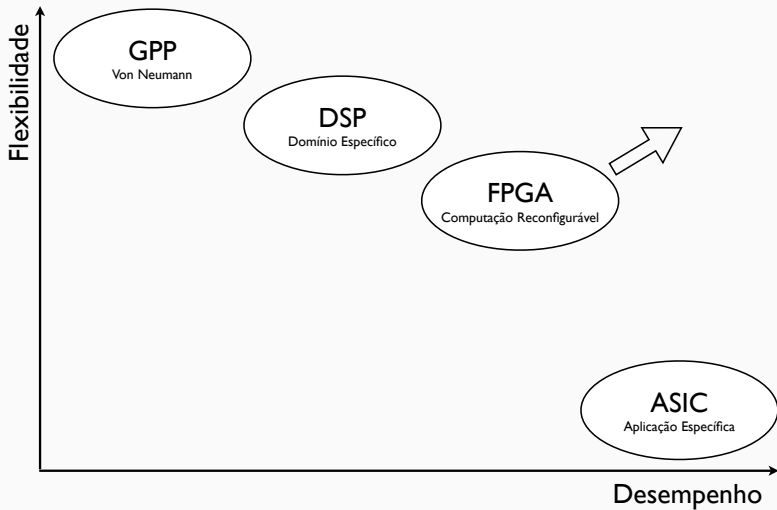
Universidade Federal de São Carlos



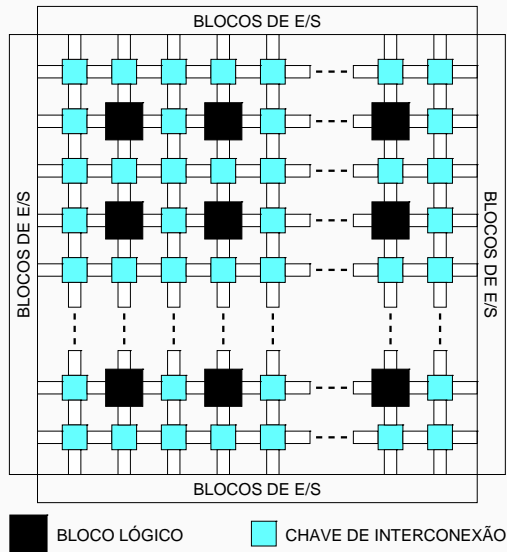
## Mercado de lógica digital [Hamblen and Furman(2001)]



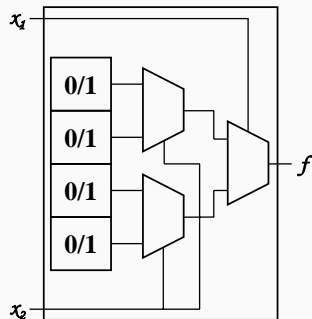
# Relação entre flexibilidade e desempenho [Bobda(2007)]



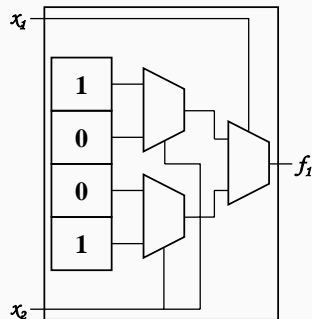
# Estrutura básica de um FPGA



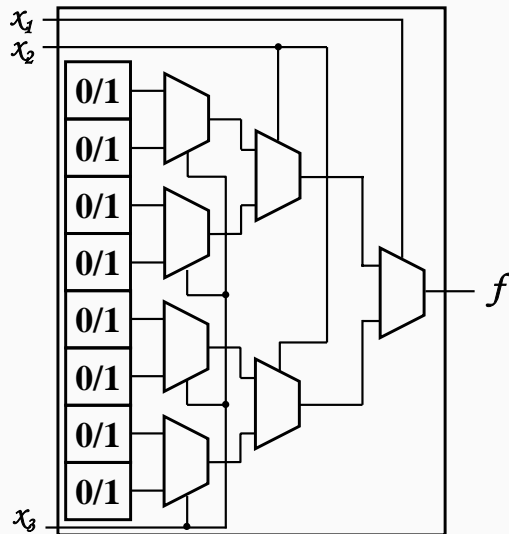
# Uso de LUTs para implementação de funções lógicas



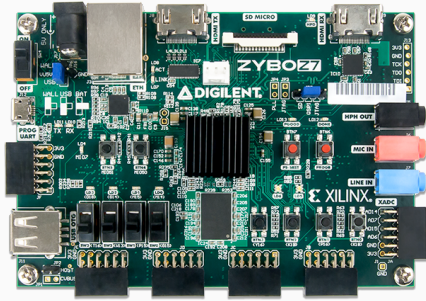
$x_1$	$x_2$	$f_1$
0	0	1
0	1	0
1	0	0
1	1	1



## Circuito de uma LUT de três entradas



# Field-Programmable Gate Array (FPGA)



- Programmable Logic Devices (PLDs);
- Field-Programmable Gate Arrays (FPGA);





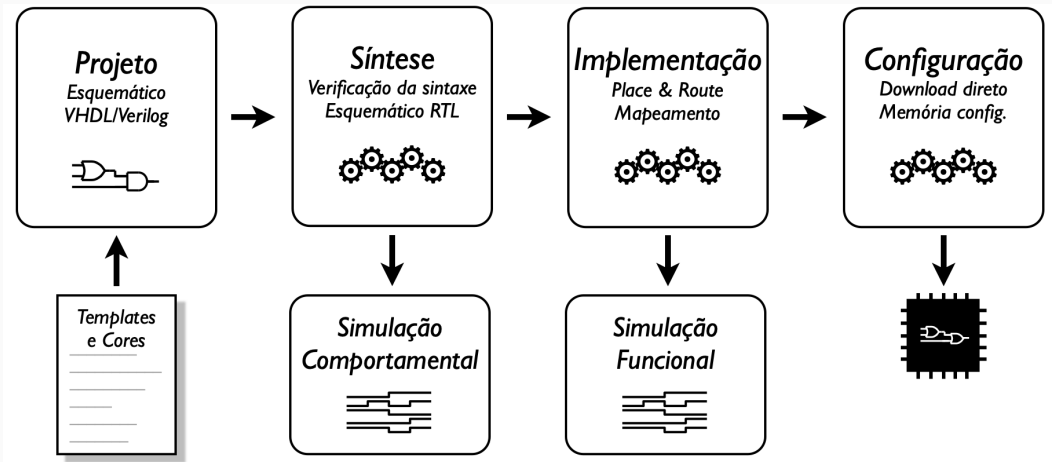
# Application-Specific Integrated Circuit (ASIC)

- FPGAs permitem implementar praticamente qualquer circuito, mas têm algumas desvantagens:
  - custam mais caro;
  - ocupam mais espaço; e
  - possuem desempenho limitado;
- Para atingir melhores resultados é possível criar um chip do zero:
  - *[semi-]custom design*;
- *Application-Specific Integrated Circuit (ASIC)*

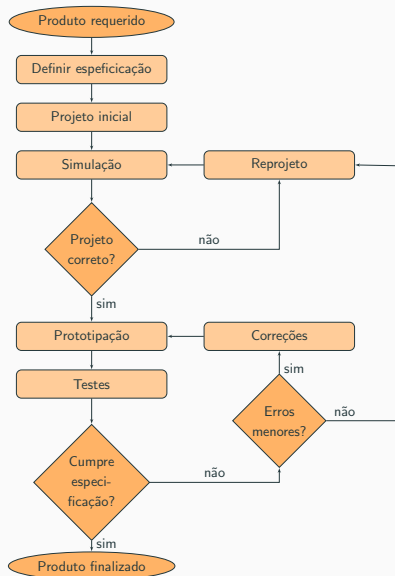
## Application-Specific Integrated Circuit (ASIC)

- Podem ser otimizados para um determinada tarefa, apresentando melhor desempenho;
- Seu custo é alto, mas se fabricado em grandes quantidades oferece a melhor relação custo-benefício;
- Uma desvantagem é que a fabricação de um chip personalizado muitas vezes leva uma quantidade considerável de tempo, na ordem dos meses.

# Fluxo de desenvolvimento para FPGAs [Menotti(2010)]



# Fluxo de desenvolvimento para FPGAs



- Diagramas esquemáticos podem ser usadas para projetar circuitos relativamente pequenos;
- Para projetar grandes circuitos, como os dos computadores atuais, são necessárias ferramentas CAD<sup>1</sup> que automatizam boa parte do processo;
- Geralmente, elas são comercializadas em pacotes, que possuem uma ferramenta para cada etapa do projeto.

---

<sup>1</sup>Do inglês: *Computer-Aided Design*

<sup>2</sup>Do inglês: *Hardware Description Language*

## Projeto inicial (design entry)

- Feito manualmente, pois requer experiência e intuição;
- Esquemáticos;
  - Projeta-se o circuitos desenhando suas portas lógicas e suas conexões;
  - Impraticável para circuitos grandes;
- Linguagem de Descrição de Hardware (HDL);
  - Similar a uma linguagem de programação, mas descreve o hardware ao invés de um programa;
  - Linguagens proprietárias, abertas e padronizadas (IEEE);
  - Oferecem portabilidade e reúso de código;
  - VHDL e Verilog são as linguagens mais difundidas;
  - Apesar de serem diferentes, oferecem recursos similares;
- Pode ser feito de forma hierárquica.

- Influenciada pelas linguagens Ada e Pascal;
- Criada pelo DoD (US) em 1983 para documentar CIs, posteriormente usada para simulação e síntese;
- Padronizada pelo IEEE 1076-(1987, 1993, 2000, 2002 e 2008).

---

<sup>3</sup>Do inglês: *Very High Speed Integrated Circuit*

<sup>3</sup>Do Inglês: *VHSIC<sup>4</sup> Hardware Description Language*

- Influenciada pelas linguagens C e Fortran;
- Criada em 1983 pela Cadence para simulação, posteriormente usada para síntese e verificação;
- Padronizada pelo IEEE 1364-(1995, 2001 e 2005);
- Incorporada ao SystemVerilog IEEE 1800-(2005, 2009 e 2017).

---

<sup>5</sup>Da combinação de *verification* e *logic*



- Geração de um circuito lógico a partir de sua especificação;
- Expressões obtidas inicialmente não são ótimas, pois refletem a especificação;
- Parte importante desse processo é obter circuitos equivalentes melhores;
- Nem sempre o objetivo é reduzir o custo, depende da tecnologia de implementação;

## Simulação funcional comportamental

- A partir das expressões de um circuito, é possível realizar a sua simulação;
- Assume que o circuito será implementado com portas perfeitas, ou seja, ignora o atraso de propagação do sinal elétrico;
- O simulador exige que o usuário especifique as entradas a serem testadas e fornece as saídas correspondentes;
- Sua saída normalmente é exibida em diagramas de formas de onda;
- Também é possível informar as saídas esperadas para que a ferramenta as verifique automaticamente.

- Determina exatamente como implementar o circuito em um determinado chip;
- Mapeiam um circuito especificado na forma de expressões lógicas para uma realização que faz uso dos recursos disponíveis no chip de destino;
- Os recursos não são necessariamente portas simples;
- Determina as conexões que devem ser feitas entre esses recursos para implementar o circuito desejado;




## Simulação temporal (Timing/Gate-level simulation)

- Circuitos eletrônicos não podem executar sua função com atraso zero, apresentam um **atraso de propagação**;
- Quando as entradas de um circuito mudam, leva certo tempo para que a sua saída seja atualizada corretamente;
- Além disso, há o atraso dos sinais que se propagam pelos fios usados para conectar os elementos;
- O simulador avalia os atrasos esperados para que se possa determinar se o circuito atinge os requisitos de desempenho;
- Se a meta não for atingida, o projetista pode tentar repetir o projeto físico informando novas restrições à ferramenta;
- Se isso não for suficiente, o projetista precisa tentar otimizações na síntese ou mesmo modificar o projeto inicial;

- Tendo verificado que o circuito projetado atende a todos os requisitos da especificação, o circuito é implementado em um chip real;
- Se um chip fabricado sob encomenda for criado para esse projeto, essa etapa será chamada de fabricação;
- Se um dispositivo de hardware programável for usado, então este passo é chamado de configuração ou programação.

## Para saber mais...

- <https://www.fpga4fun.com/>
- <https://www.sparkfun.com/fpga>
- <https://www.hackster.io/projects/tags/fpga>
- [https://www.reddit.com/r/FPGA/comments/cgwozx/fun\\_projects\\_to\\_learn\\_fpga\\_for\\_a\\_software\\_engineer/](https://www.reddit.com/r/FPGA/comments/cgwozx/fun_projects_to_learn_fpga_for_a_software_engineer/)

-  Christophe Bobda.  
**Introduction to Reconfigurable Computing: Architectures, Algorithms and Applications.**  
Springer, 2007.
-  J. O. Hamblen and M. D. Furman.  
**Rapid Prototyping of Digital Systems.**  
Kluwer, 2001.
-  Ricardo Menotti.  
**LALP: uma linguagem para exploração do paralelismo de loops em computação reconfigurável.**  
PhD thesis, Universidade de São Paulo, 2010.

# Microprocessadores e Microcontroladores (27146)



FPGAs e ferramentas CAD

---

Prof. Ricardo Menotti (menotti@ufscar.br)

Atualizado em: 26 de abril de 2021

**Departamento de Computação**

Centro de Ciências Exatas e de Tecnologia

Universidade Federal de São Carlos