

# Laboratório de Sistemas Digitais

## Aula Teórico-Prática 1

Ano Letivo 2016/17

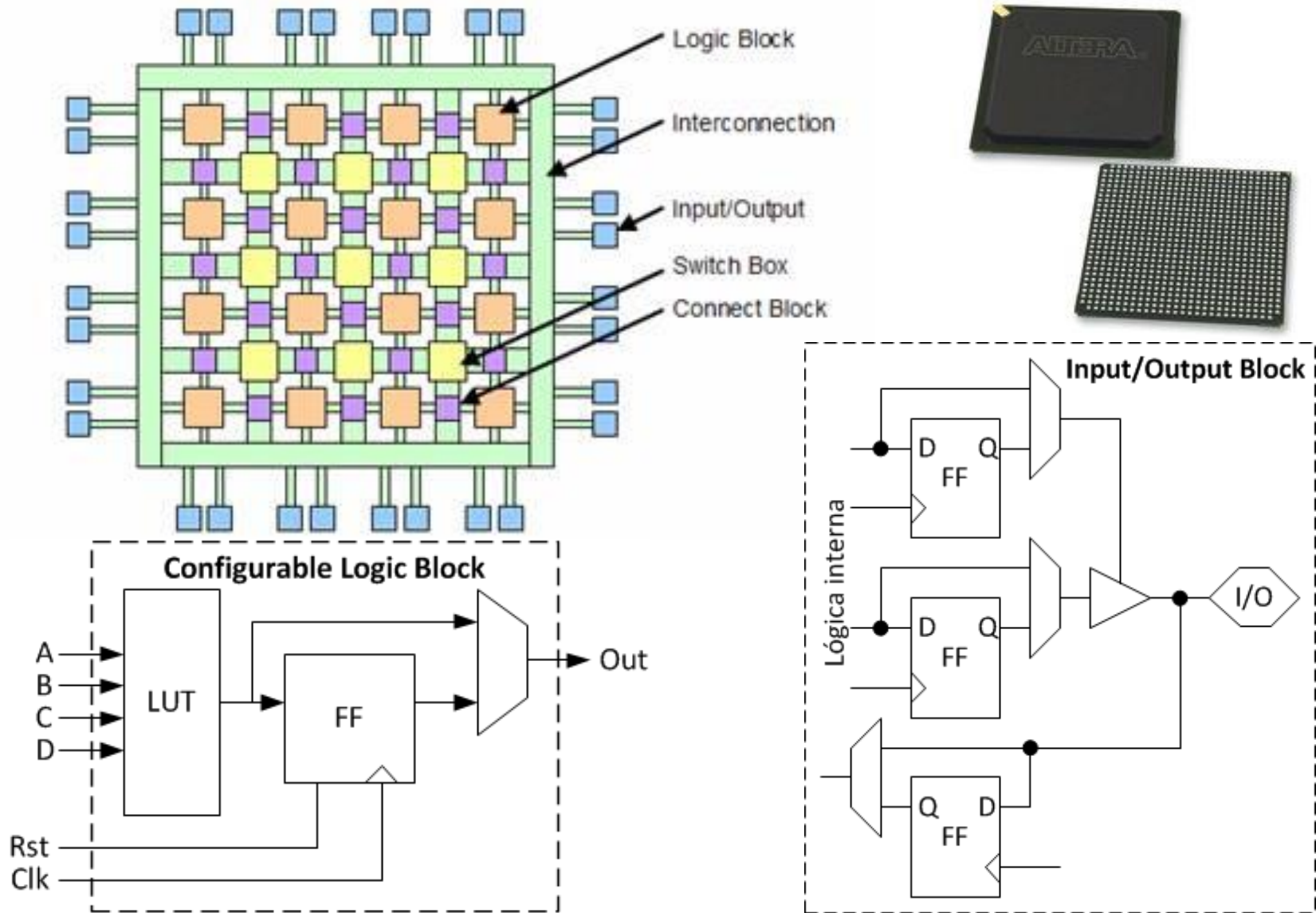
Introdução às FPGAs, ferramentas de projeto e kit de desenvolvimento

Arnaldo Oliveira, Guilherme Campos, Iouliia Skliarova, Tomás Oliveira e Silva

# Conteúdo

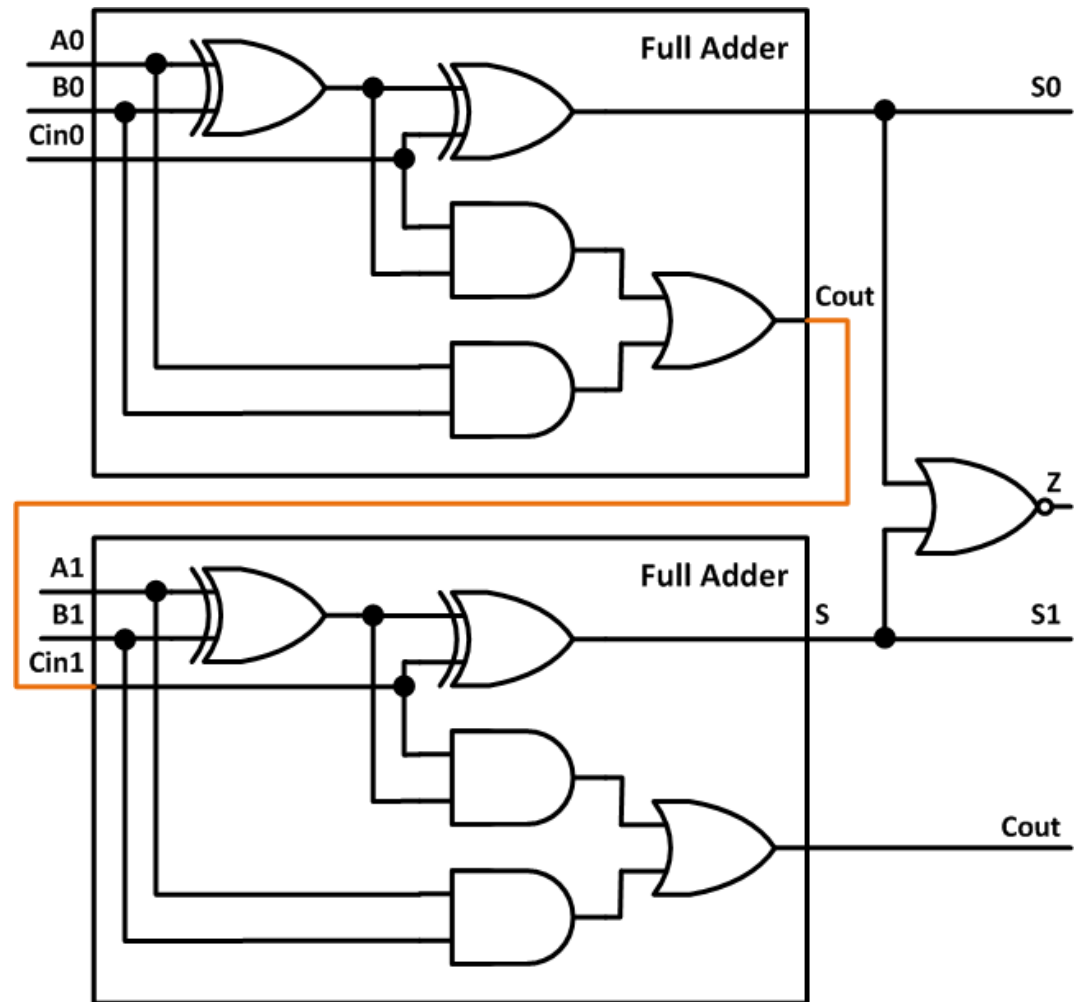
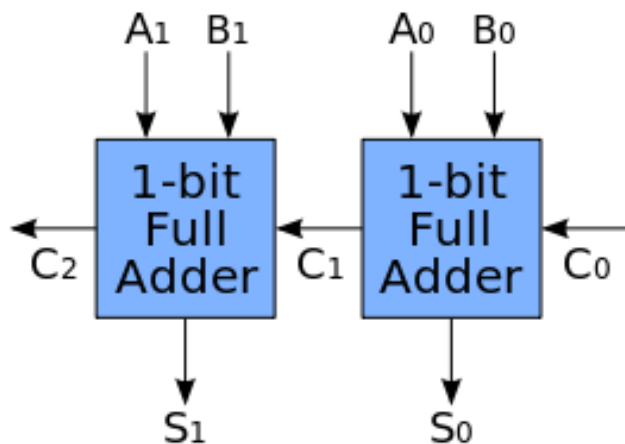
- Breve introdução às FPGAs
  - Arquitetura básica
  - Fluxo e ferramentas de projeto
  - Placas de desenvolvimento
    - O kit Terasic DE2-115 (usado nas aulas práticas)

# FPGA – Field Programmable Gate Array

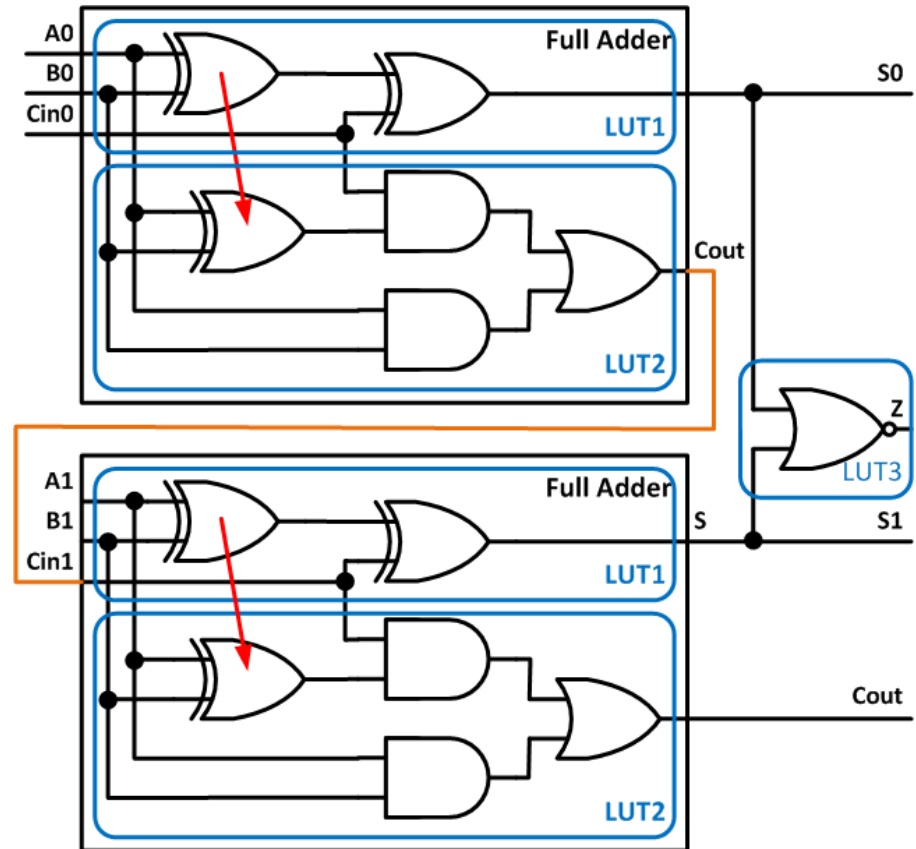
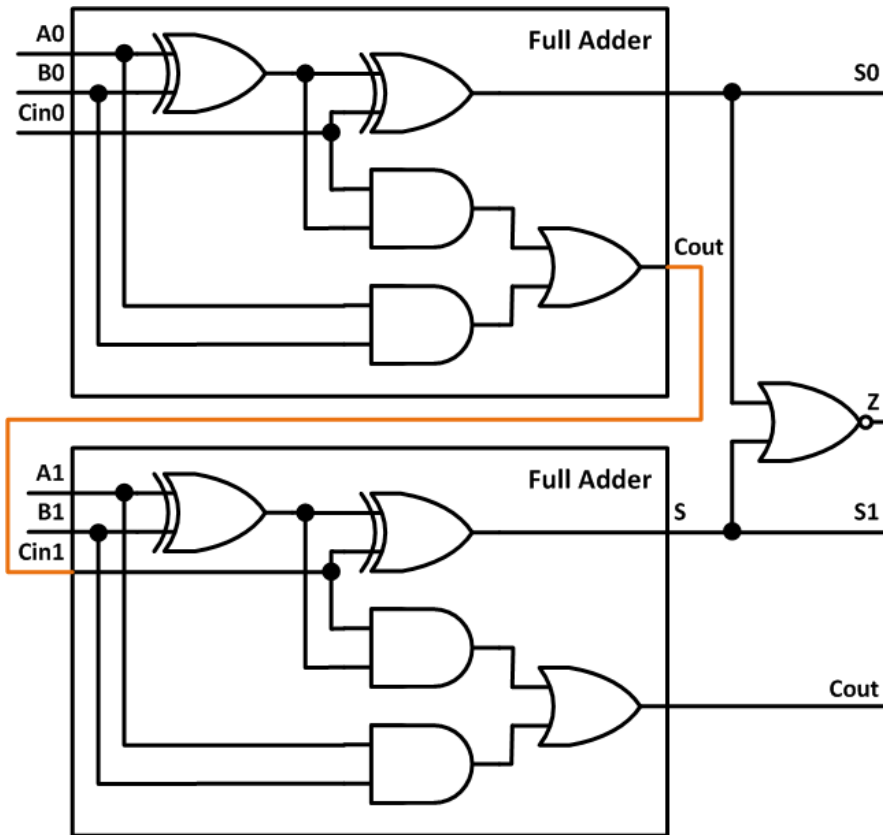


# Circuitos Combinatórios com LUTs

- Exemplo de um somador de dois bits com indicação de resultado (S1 S0) igual a “Zero”



# Adaptação do Circuito para Implementação com LUTs



Por vezes é necessário “replicar” lógica (absorvida pela LUT). Porquê?

# Circuitos Combinatórios com LUTs

**LUT1 (3 variáveis)**

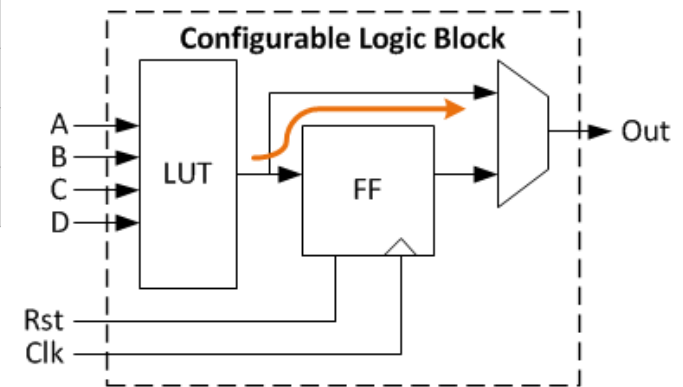
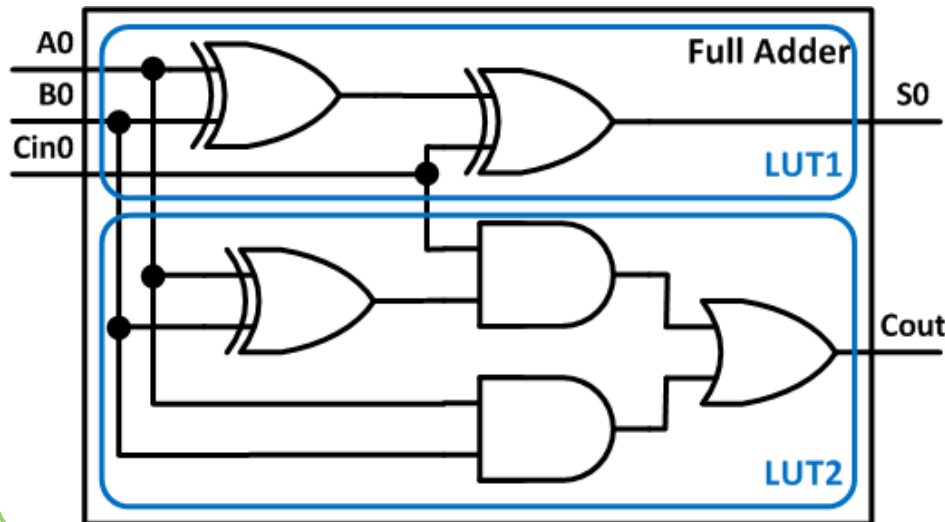
A	B	Cin	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

**LUT2 (3 variáveis)**

A	B	Cin	Cout
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

**LUT3 (2 entradas)**

S1	S0	Z
0	0	1
0	1	0
1	0	0
1	1	0

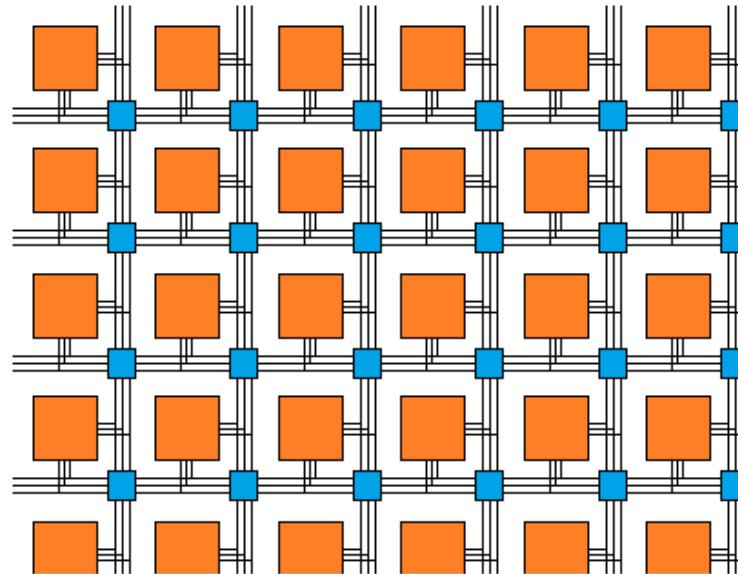
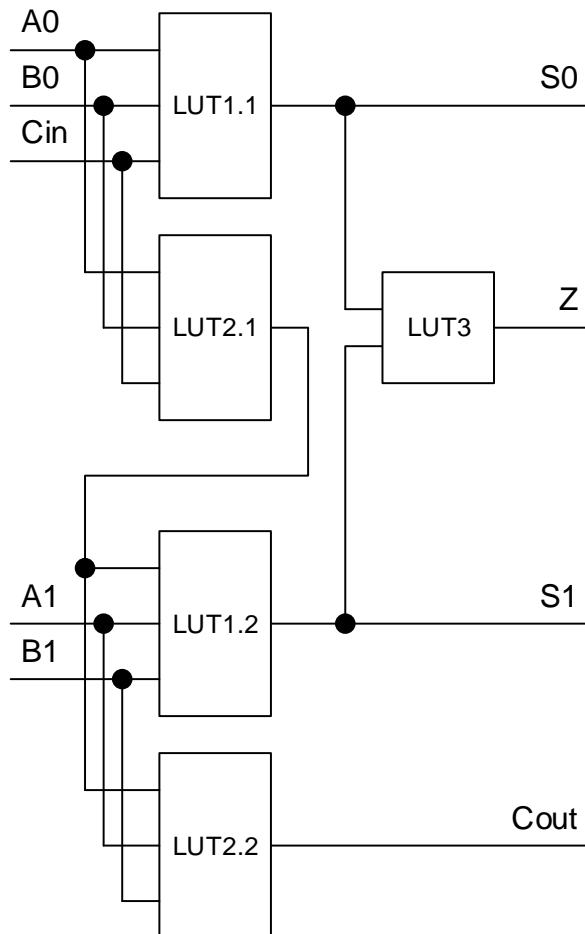


Mapeamento em LUTs realizado pela ferramentas de implementação

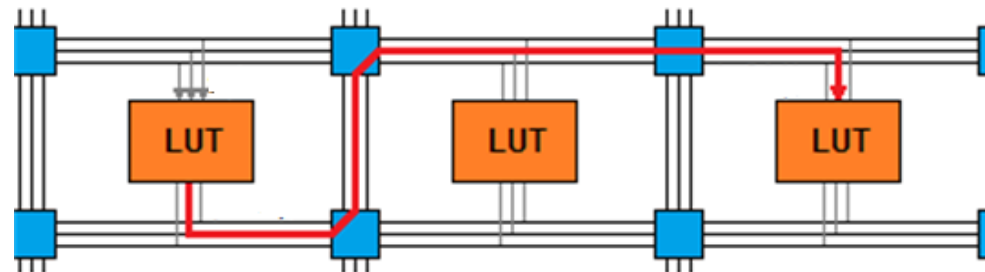
Como implementar uma LUT de 2 ou 3 entradas a partir de uma de 4 entradas?

# Posicionamento e Interligação de LUTs (e Logic Blocks) nas FPGAs

Implementação do  
somador com LUTs



"Clean slate" FPGA: programmable **gates** and **routers**

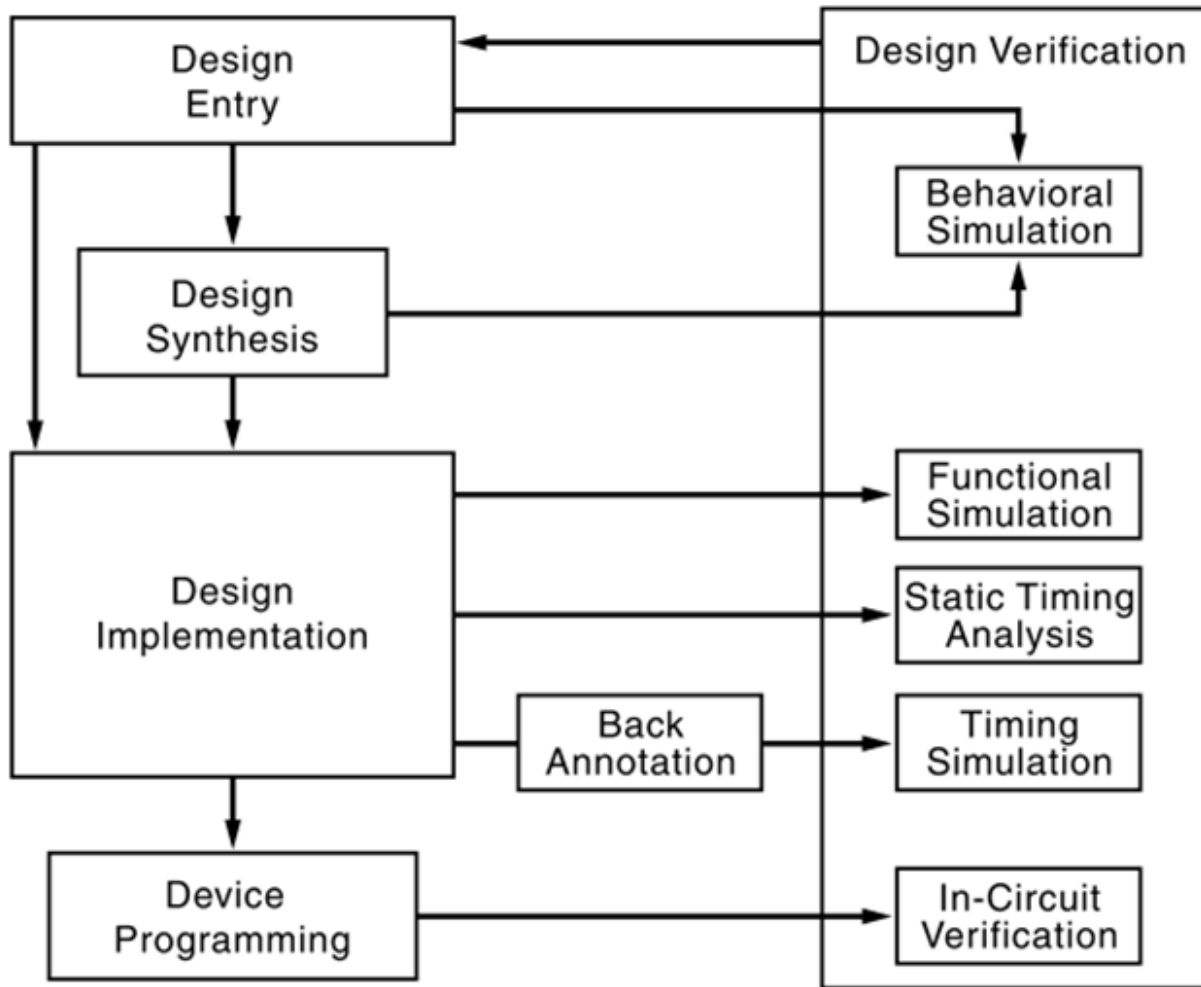


Routing from the LUT to another through **switch boxes**

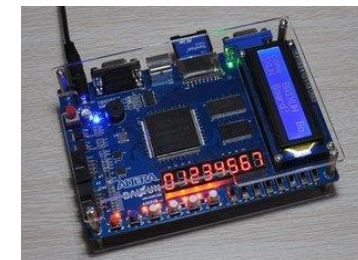
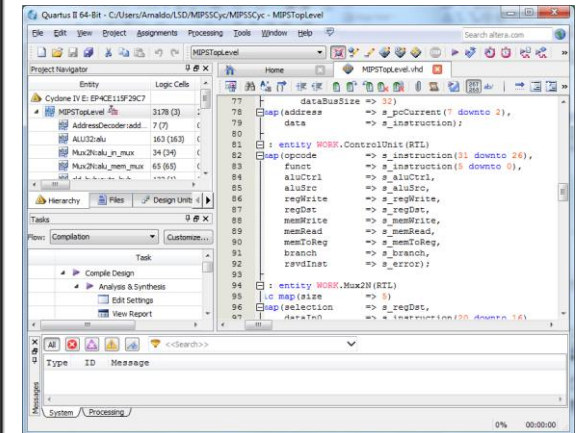
Realizado pelas  
ferramentas de  
compilação/  
projeto

Fonte:  
<http://www.yosefk.com/blog/category/hardware>

# FPGA Design Flow

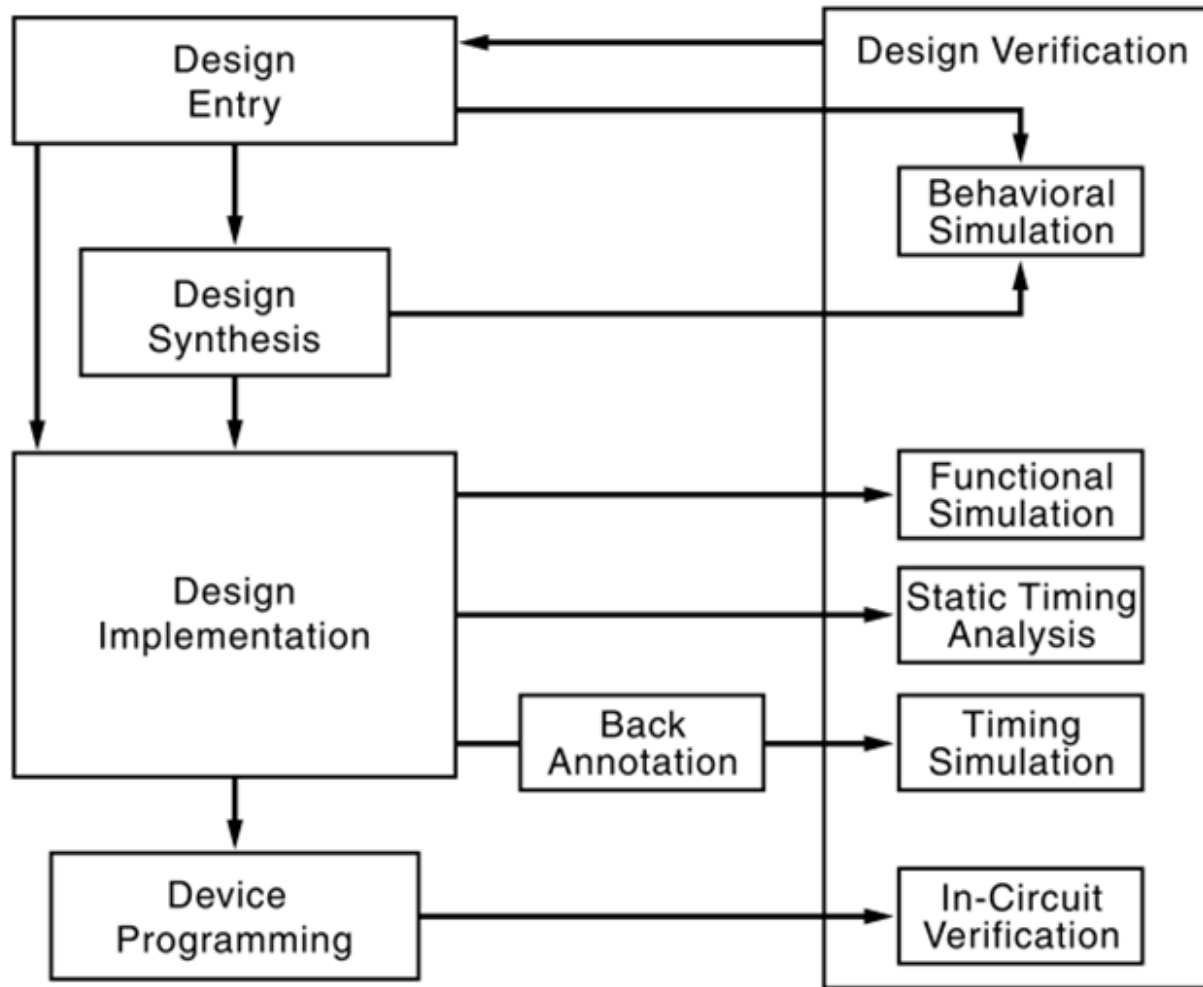


- *Design entry* baseado em:
  - Linguagens de descrição de hardware
  - Diagramas esquemáticos
  - Diagramas de estado



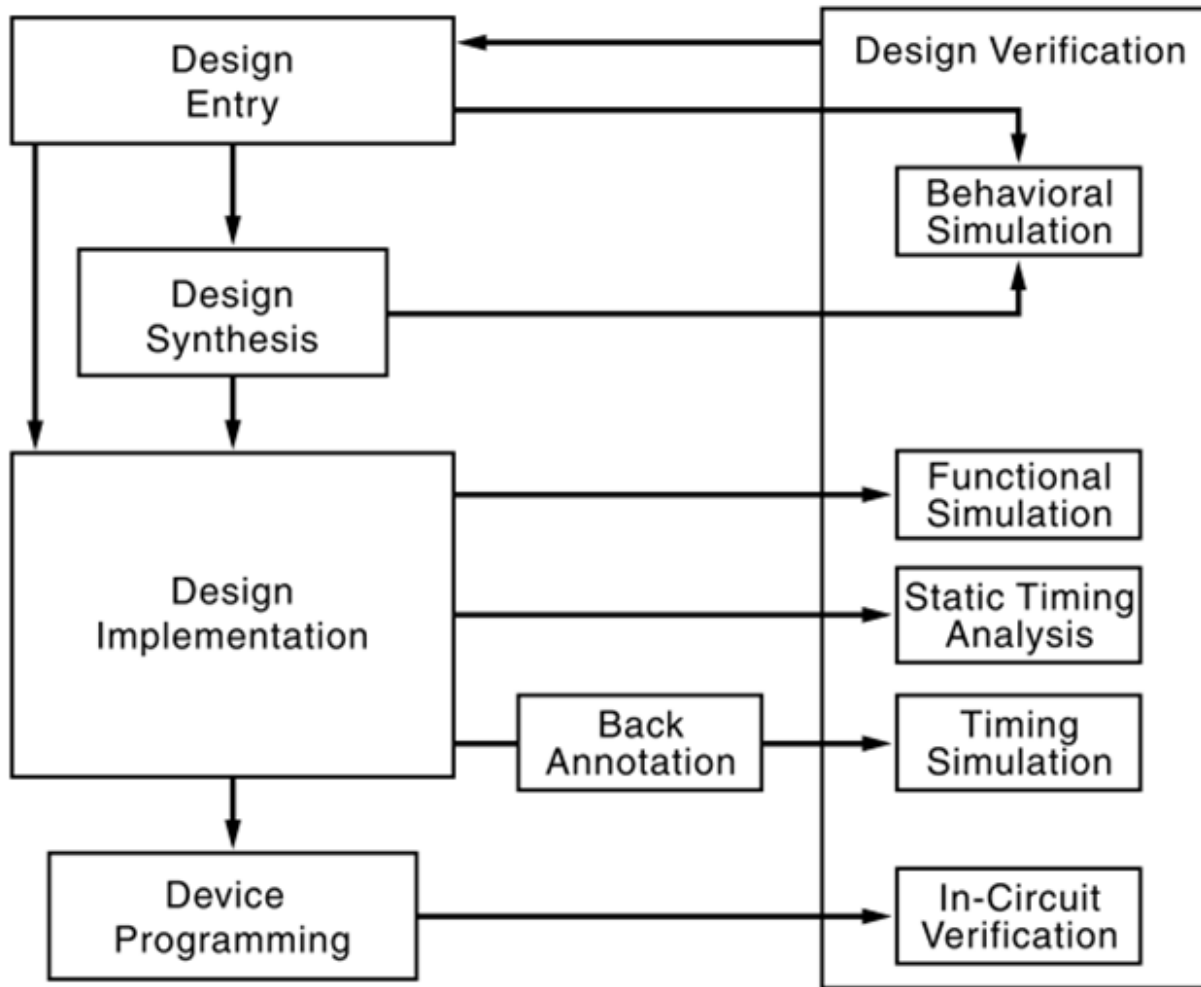


# Síntese Lógica (Synthesis)



- Resulta numa netlist (i.e. nos componentes de hardware e suas interconexões) que implementam o comportamento e a estrutura modeladas
- Resultado
  - Netlist
  - Estimativas de desempenho do circuito e recursos lógicos necessários

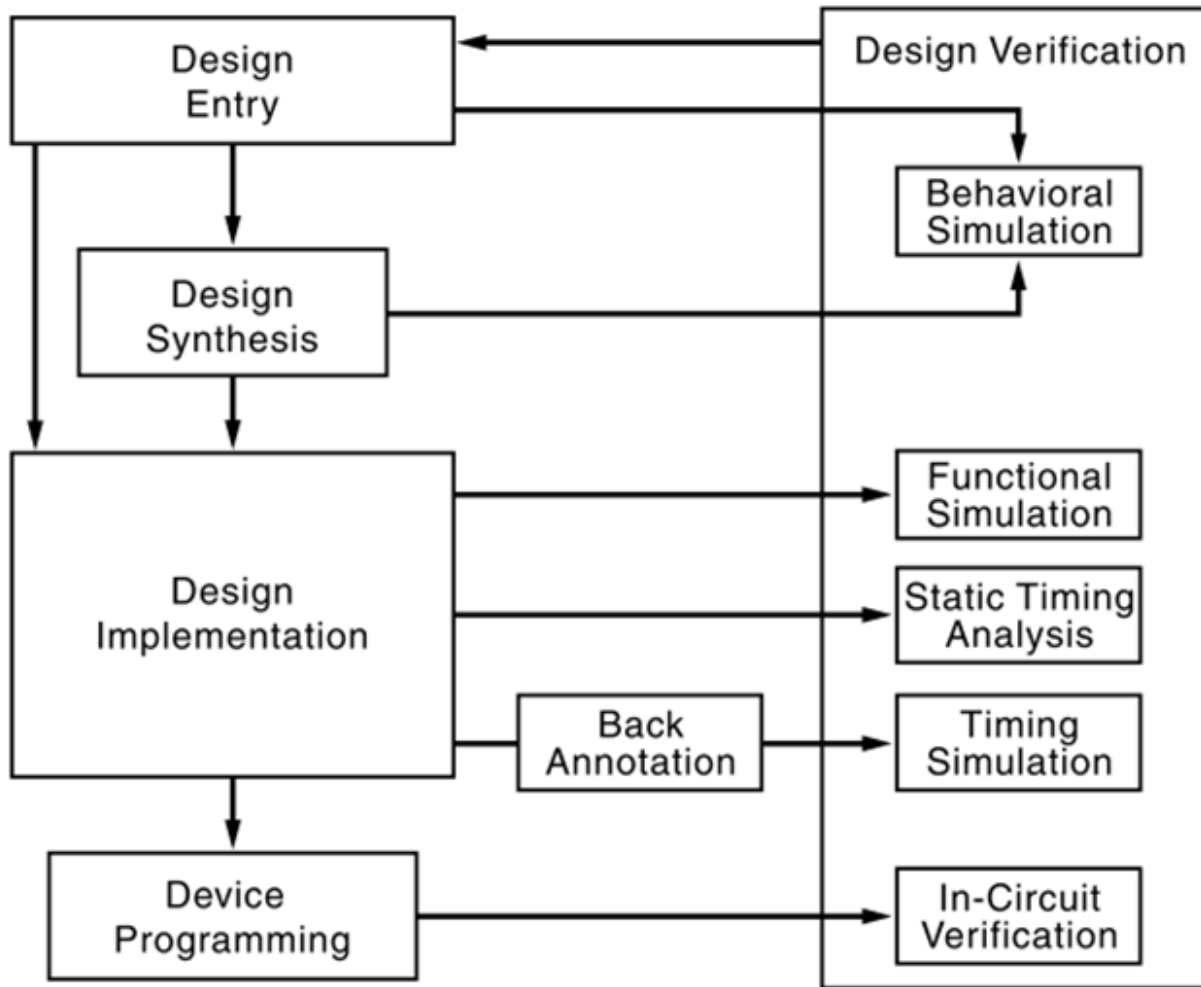
# Implementação (Fit / Place and Route)



- Mapeia a netlist em primitivas da FPGA
- Posiciona as primitivas em localizações específicas da FPGA
- Realiza (encaminha) as interconexões entre as primitivas
- Resultados
  - Ficheiro de configuração da FPGA
  - Relatório sobre os recursos utilizados da FPGA, tempos de atraso e outras métricas (consumo energético, ...)

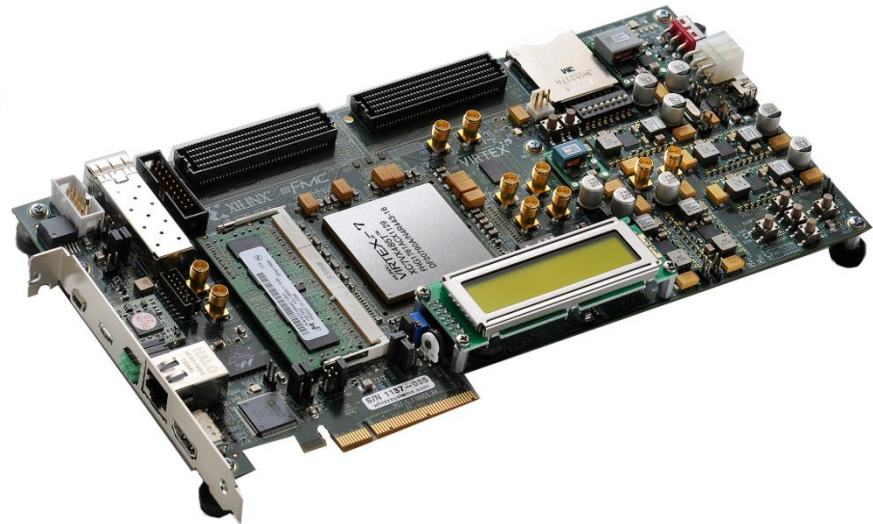
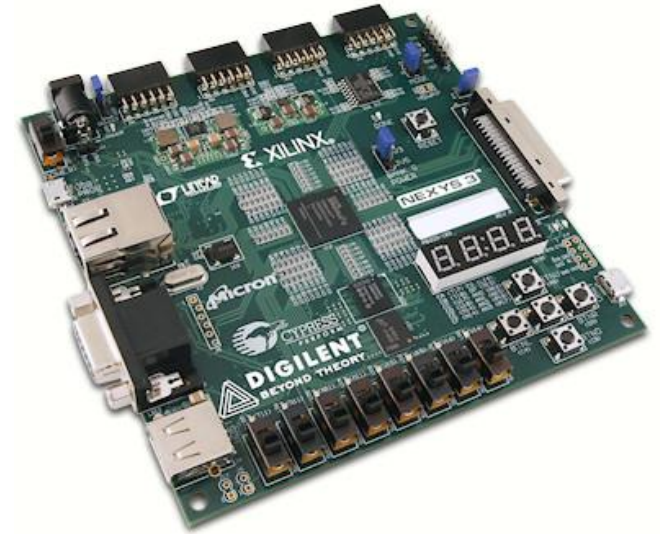


# Programação do Dispositivo (FPGA)



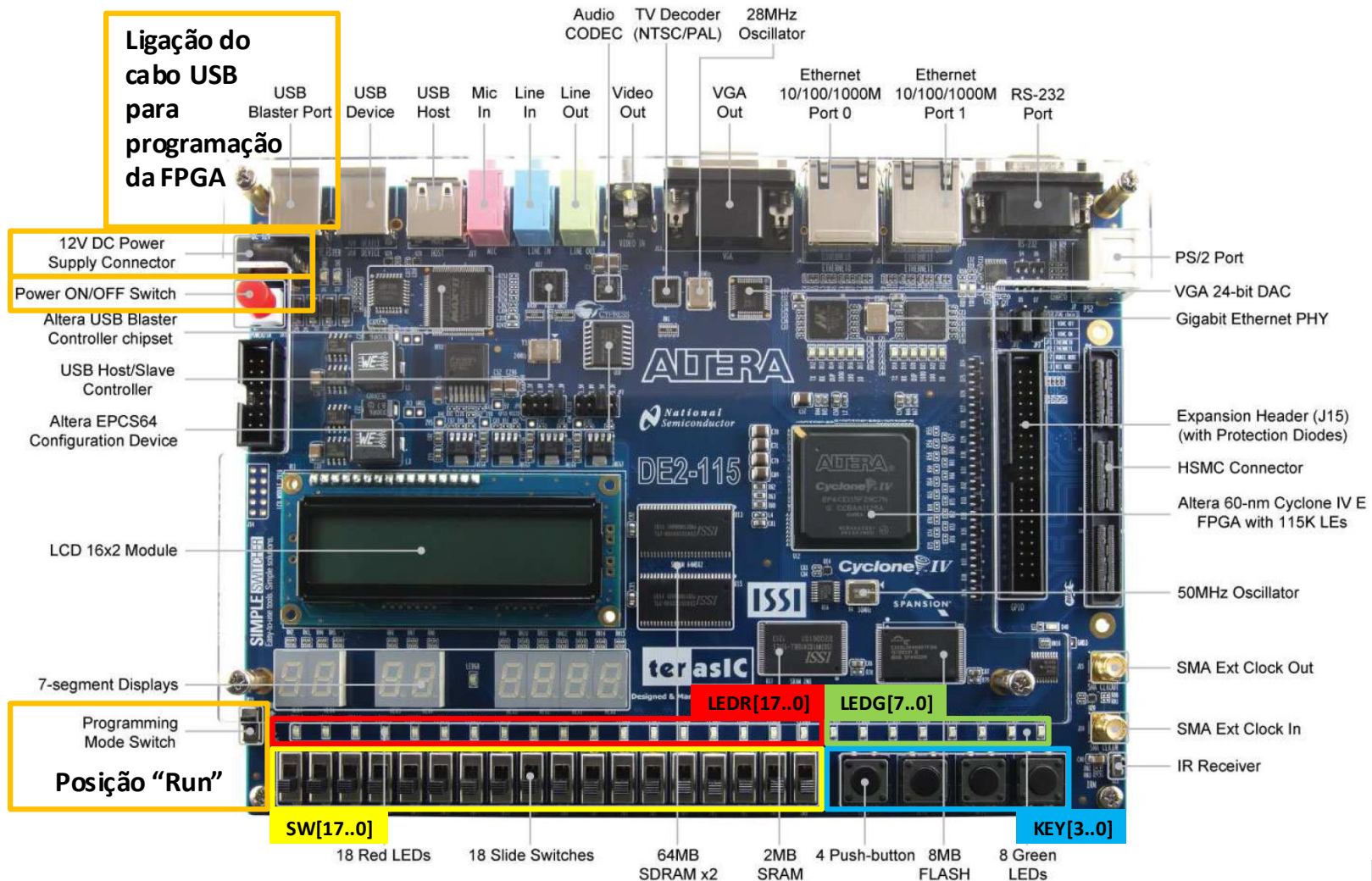
- Transfere o ficheiro de configuração para a FPGA
  - Realizada através de software e de um cabo de programação adequado
  - FPGA normalmente baseada em SRAM (configuração volátil)
  - Existem também soluções não voláteis baseadas em memórias FLASH

# Placas de Desenvolvimento com FPGAs

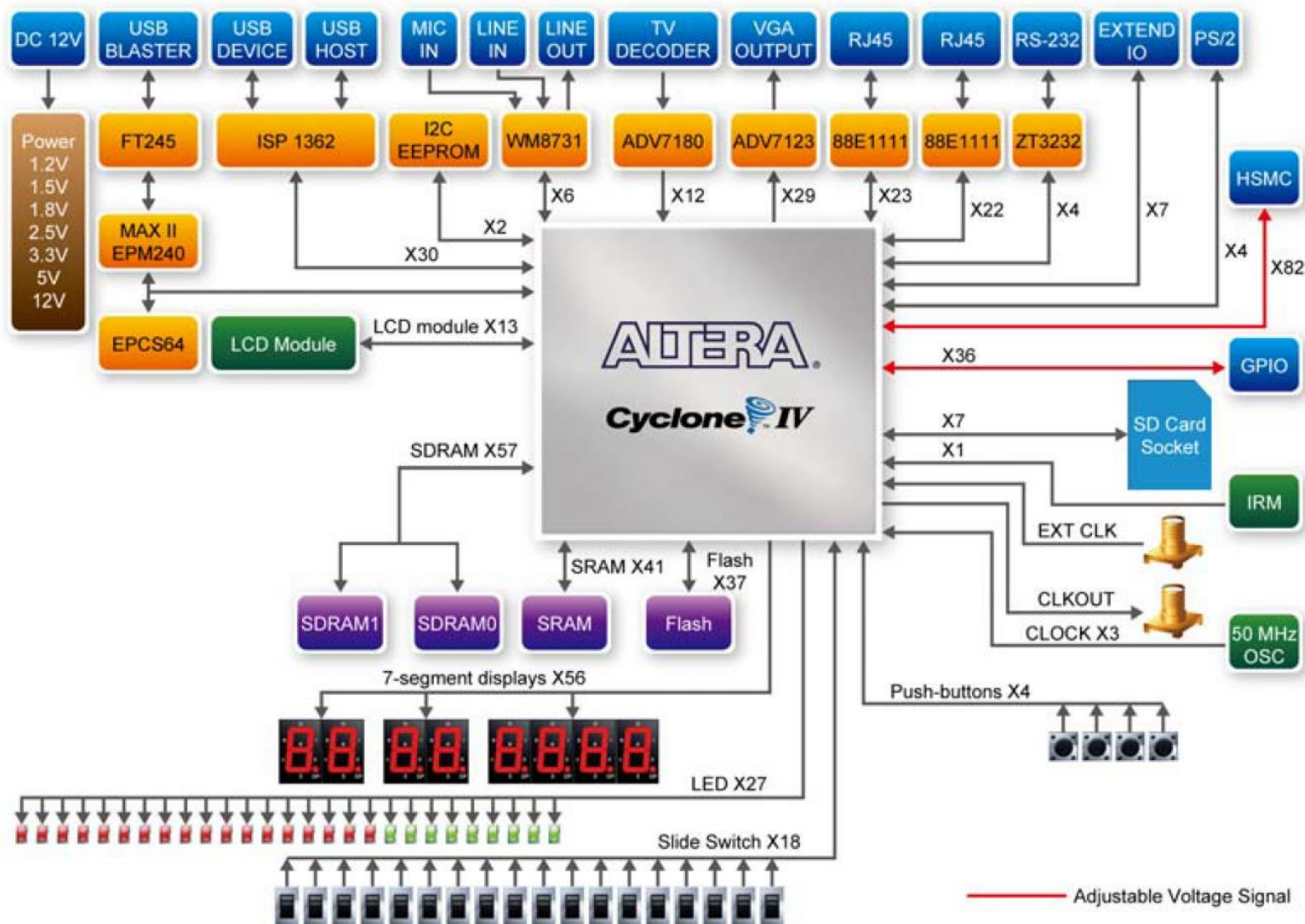




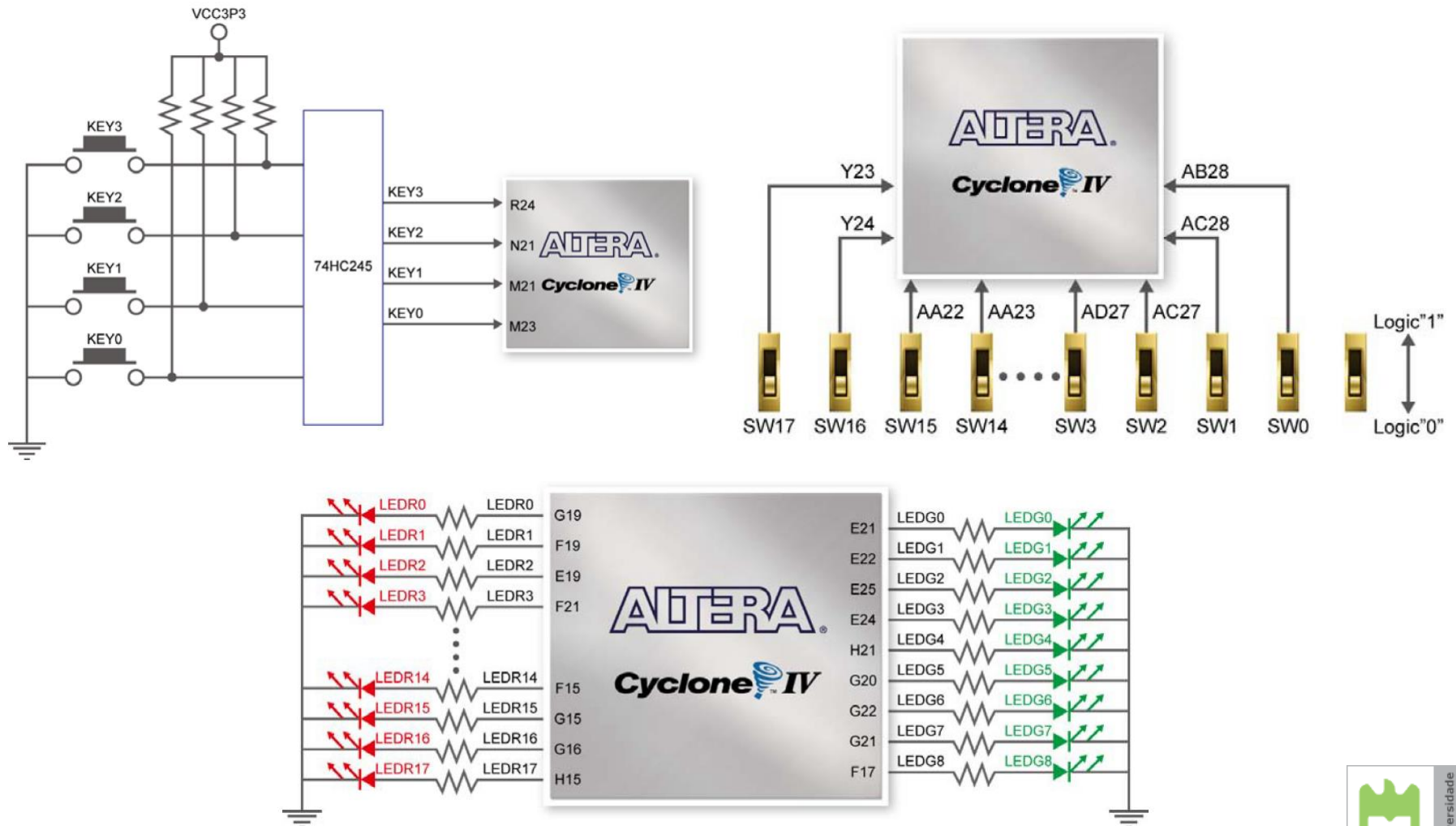
# O Kit Terasic DE2-115



# Diagrama de Blocos do Kit



# Alguns Dispositivos do Kit (botões, interruptores e LEDs)



# Comentários Finais

- No final da primeira semana de aulas de LSDig, deverá ser capaz de:
  - Saber o que é uma FPGA, conhecer em traços gerais a sua arquitetura interna típica e descrever os passos principais do fluxo de projeto
  - Compreender e utilizar as construções mais elementares da linguagem VHDL
  - Criar projetos simples baseados em VHDL e diagramas esquemáticos, sintetizá-los, implementá-los e testá-los em FPGA
- Mais informação sobre as aulas práticas, kit com FPGA e ferramentas de projeto no site da UC
  - [elearning.ua.pt](http://elearning.ua.pt)