

Projeto com Circuitos Reconfiguráveis

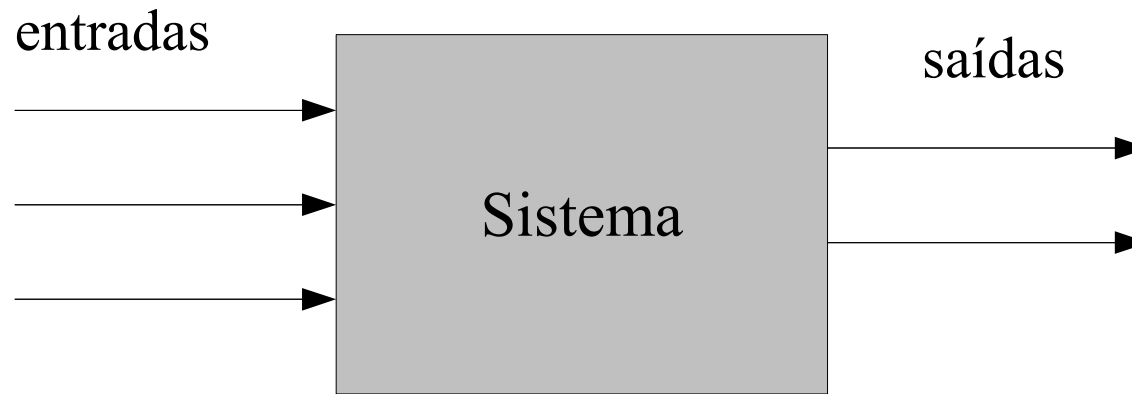
Aula 2 – Projeto com Dispositivos Lógicos Programáveis

Prof. Daniel M. Muñoz Arboleda

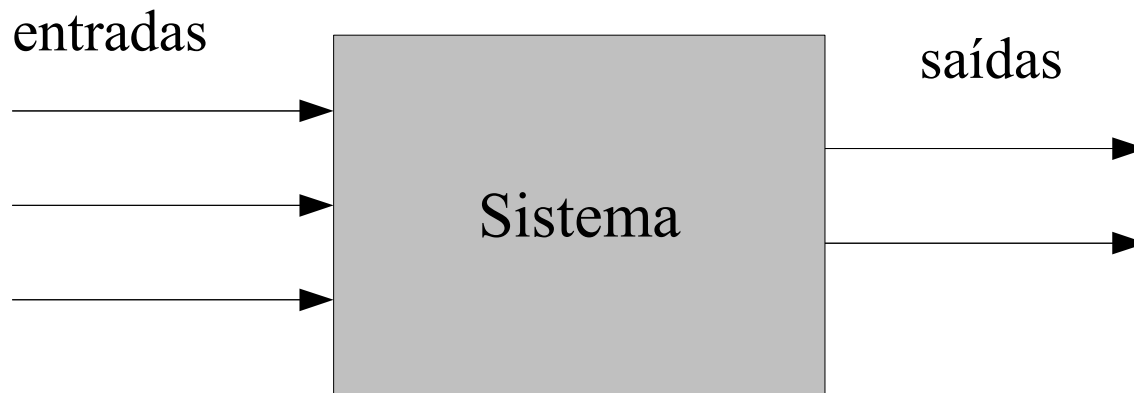
FGA - UnB

Definição de Sistema

- O que é sistema?
- O que é um sistema digital?
- O que é um sistema digital binário?

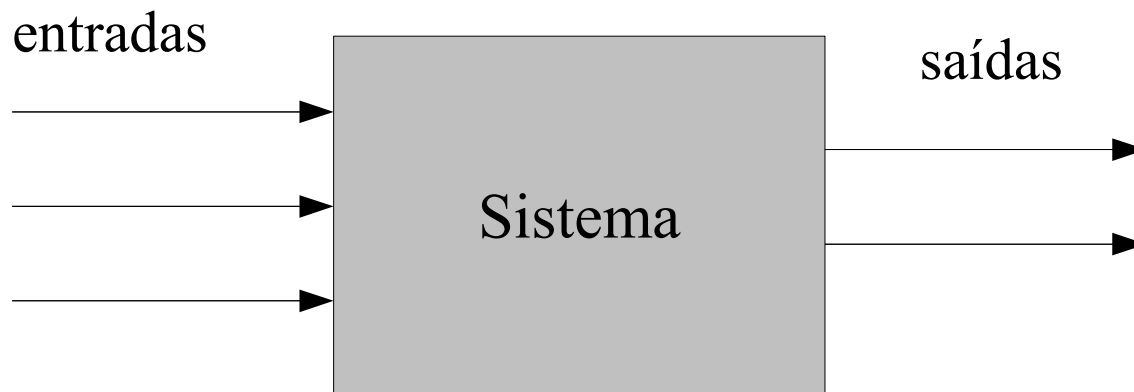


Definição de Sistema Digital



Definição. Um sistema é uma porção (parte) do universo que se individualiza do restante por meio de uma “*fronteira*”. O sistema interage com o universo aceitando entradas e produzindo saídas para o mesmo. Pode ser construído pela interligação de elementos. Dita interligação pode se dar por fluxo de informação (matéria e/ou energia).

Definição de Sistema Digital Binário



Definição. Um **Sistema Digital** é uma porção (parte) do universo concreta ou abstrata que se individualiza do restante por meio de uma “*fronteira*”. O Sistema Digital está dotado de um conjunto finito de entradas e um conjunto finito de saídas e é capaz de processar informação de forma numérica. Para tanto, cada entrada e cada saída pode assumir ao longo do tempo valores de algum conjunto finito de números, denominado *domínio*.

Definição. Se todas as entradas e saídas assumirem valores somente do domínio $B = \{0, 1\}$, o sistema é denominado **Sistema digital binário**.

Processo de Projeto Sistemas Digitais

O *processo de projeto* digital pode ser definido informalmente como a transformação de uma *descrição inicial* do sistema, frequentemente denominada *especificação*, em uma descrição final, também chamada de *projeto final* ou *projeto detalhado*.

Caso de sistemas complexos.

Quando se trata de projetar sistemas complexos, a transformação dificilmente pode ser feita de maneira direta. Nestes casos, a atividade de projeto produz um *continuum* de descrições, obtidas pela sucessiva agregação de informações à descrição inicial, culminando com a descrição final.

Modelos.

Existem, fundamentalmente, três tipos de modelos para representar o processo de projeto.

- Modelo de Suzim
- Modelo de Gajski – Kuhn ou Diagrama Y
- Modelo Medland

Processo de Projeto Sistemas Digitais

Dada a *complexidade* dos sistemas digitais, impõe-se a decomposição de seu processo de projeto.

Um primeiro critério a empregar para decompor o processo de projeto é fazer com que cada passo manipule descrições com um determinado grau de detalhamento de informações.

Definição. Nível de abstração: é um conjunto de descrições de projeto com o mesmo grau de detalhamento.

Exemplo ?

Processo de Projeto Sistemas Digitais

Dada a *complexidade* dos sistemas digitais, impõe-se a decomposição de seu processo de projeto.

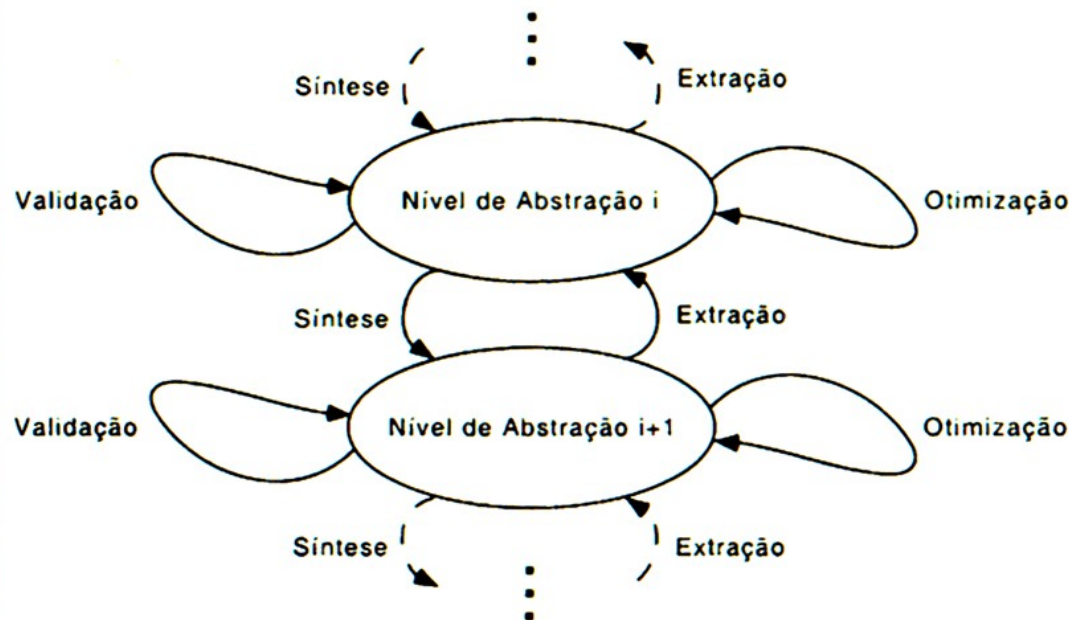
Um primeiro critério a empregar para decompor o processo de projeto é fazer com que cada passo manipule descrições com um determinado grau de detalhamento de informações.

Definição. Nível de abstração: é um conjunto de descrições de projeto com o mesmo grau de detalhamento.

Exemplo: para um mesmo sistema digital, o diagrama de portas lógicas, contém significativamente menos informação que uma descrição elétrica do tipo SPICE, estando portanto cada um em um nível de abstração diferente.

Exemplo: descrição de um microprocessador feita pelo programador e o layout do mesmo entregue para o fabricante.

O Modelo de Suzim.

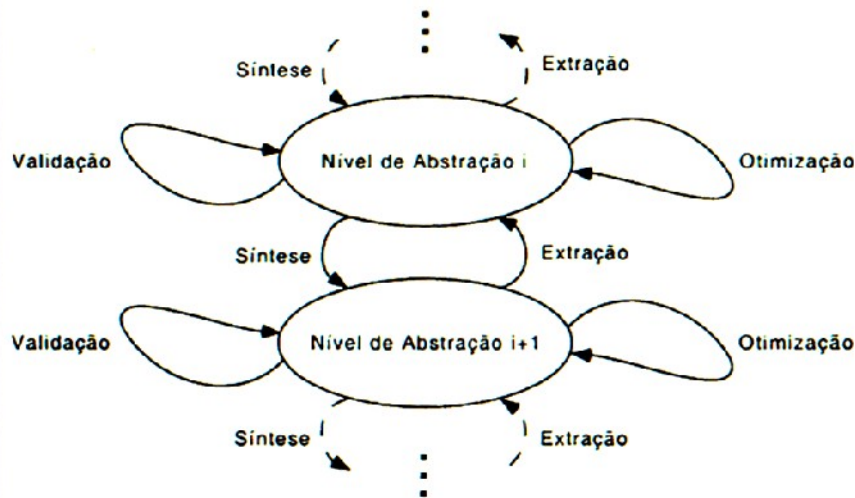


No *modelo de Suzim* é feita uma decomposição do processo de projeto segundo o critério de níveis de abstração de projeto e na classificação das operações de projeto segundo o tipo de transformação que estas realizam.

Estabelece-se uma hierarquia linear de conjuntos de descrições.

As elipses representam descrições e arcos correspondem a transformações ou operações de projeto sobre as descrições.

O Modelo de Suzim.



Operações do *modelo de Suzim*

Operações de síntese. Acrescentam detalhes a uma descrição em um nível, gerando uma descrição de um nível menos abstrato. Ex: edição de esquemáticos.

Operações de extração. Transformam a descrição lógica original, produzindo uma descrição mais abstrata. (operação oposta da síntese). Ex: *back annotation*.

Operações de validação. Existem para garantir que a funcionalidade de uma descrição corresponde à esperada. Exemplo: simuladores lógicos.

Operações de otimização. Modificam a descrição de entrada, visando melhorar as figuras de mérito do sistema em algum nível. Exemplo: minimização de lógica combinacional.

Níveis de abstração.

Descrições de um Sistema Digital

As descrições de um sistema digital podem ser classificadas em quatro ou cinco níveis de abstração.

Nível sistêmico. O sistema é descrito como um conjunto de algoritmos ou módulos.

Nível arquitetural. Descrição dos modelos através de ULAs , registradores, decodificadores, muxes, grafos de fluxo de dados e de controle, etc.

Nível lógico. Sistema digital é descrito a partir de noções elementares da teoria de circuitos digitais. Portas lógicas, flip-flops, equações Booleanas, etc.

Nível elétrico. Modelos basados na teoria de circuitos elétricos/eletrônicos, e/ou da física de semicondutores.

Nível de abstração: “Conjunto coerente de primitivas que possibilitam a descrição de um objeto” (Suzim).

Diagrama Y ou Diagrama de Gajski-Kuhn

Domínio de descrição. Conjunto de descrições que compartilham um mesmo tipo de informação. O Modelo Y é identificado por três domínios digitais.

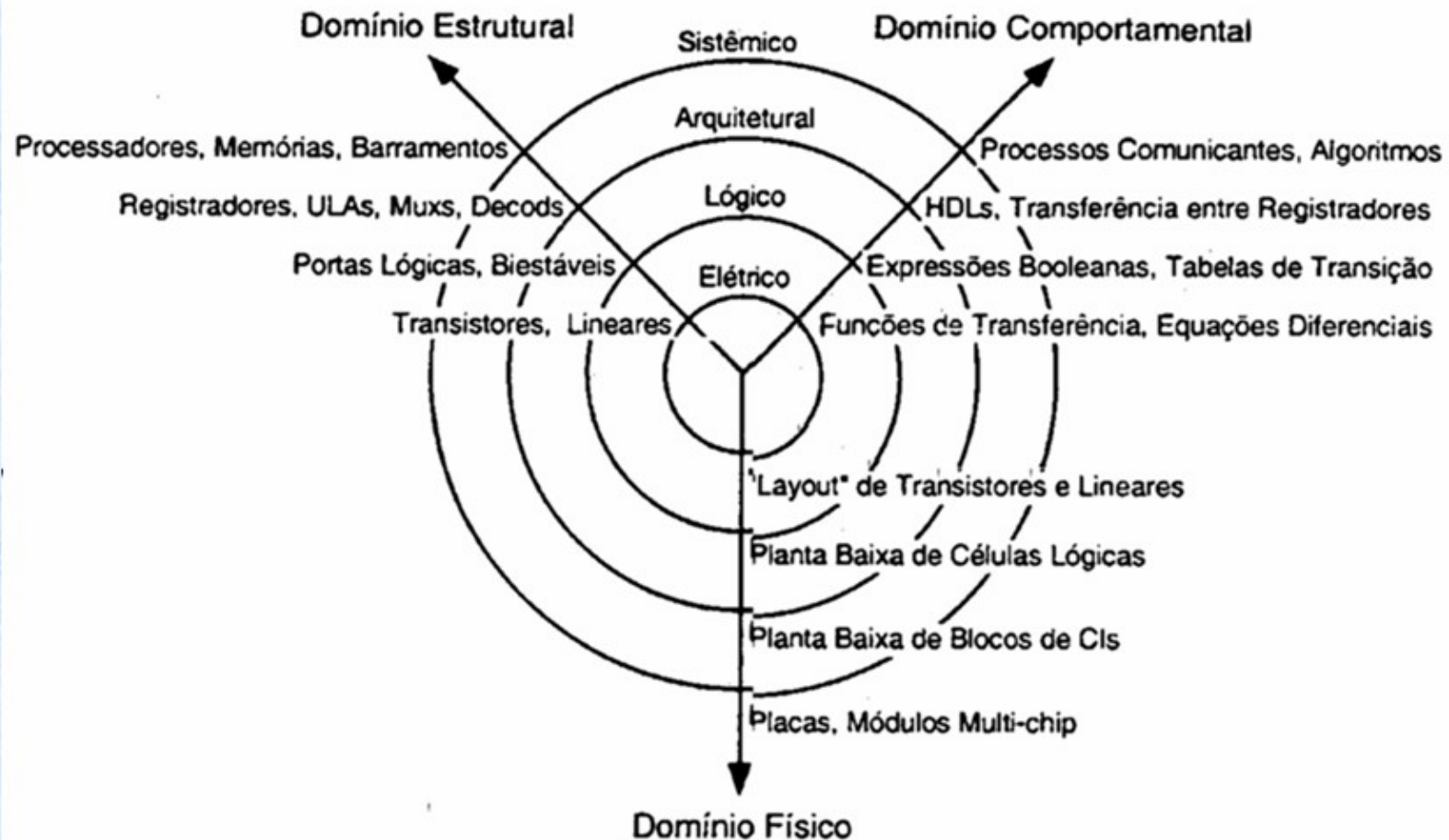


Diagrama Y ou Diagrama de Gajski-Kuhn

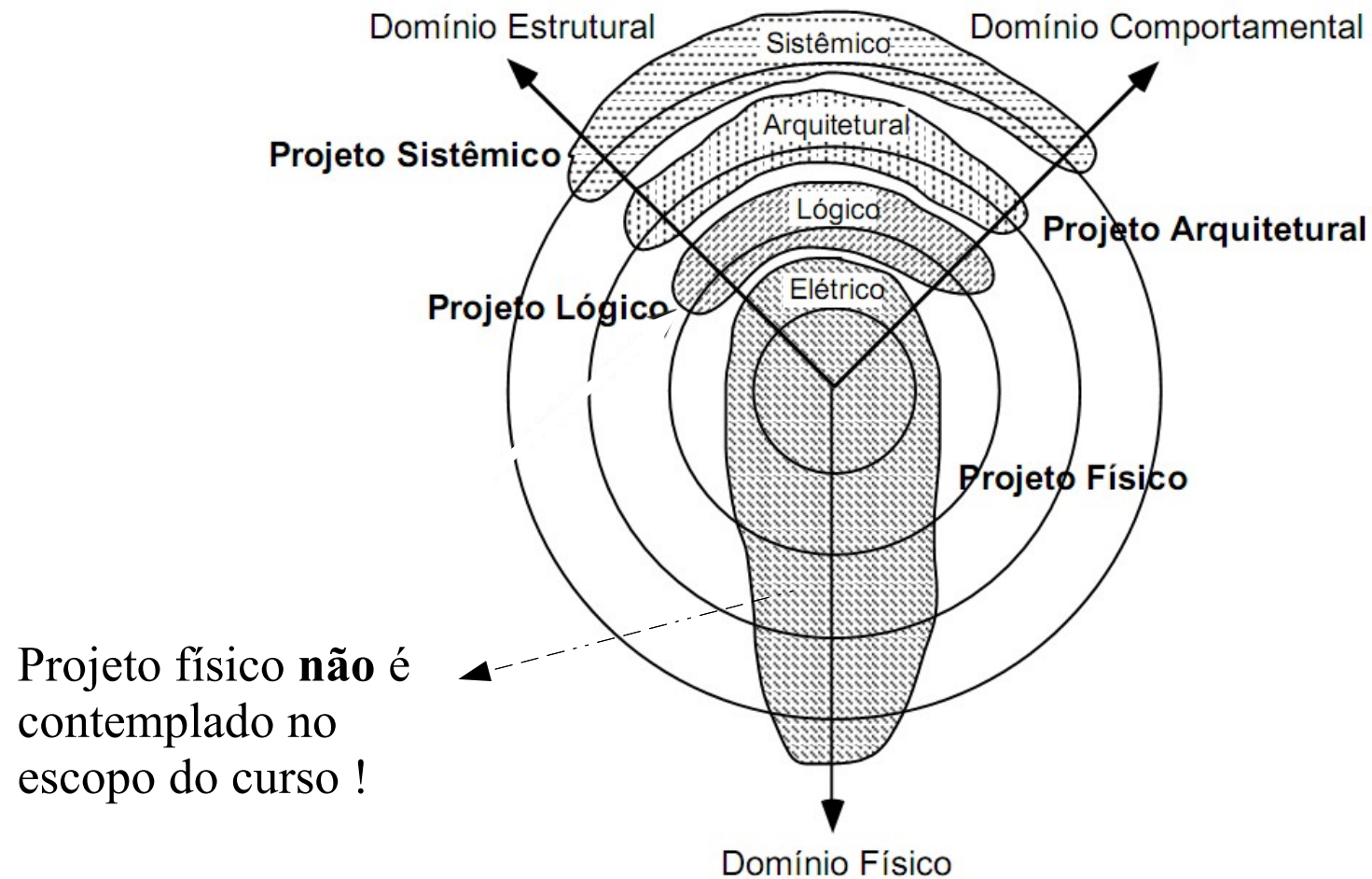
É um modelo obtido pela combinação de níveis de abstração e domínios de descrição. Os círculos concêntricos correspondem a níveis de abstração, enquanto os segmentos de reta radiais correspondem a domínios de descrição.

Cada intersecção de um círculo com um segmento radial representa um tipo de descrição distinta do sistema digital.

Tipos de *domínios de descrição* para sistemas digitais.

- *Físico ou geométrico*. Descrição sobre a geometria dos componentes/módulos e a disposição espacial destes no sistema a ser fabricado.
- *Estrutural*. Descrições sobre como interconectar blocos de base de comportamento sem se ocupar com a descrição física dos mesmos.
- *Comportamental*. Envolve descrições com informação sobre o comportamento do sistema, sem se preocupar como tal comportamento pode ser obtido, seja do ponto de vista físico, seja do ponto de vista estrutural.

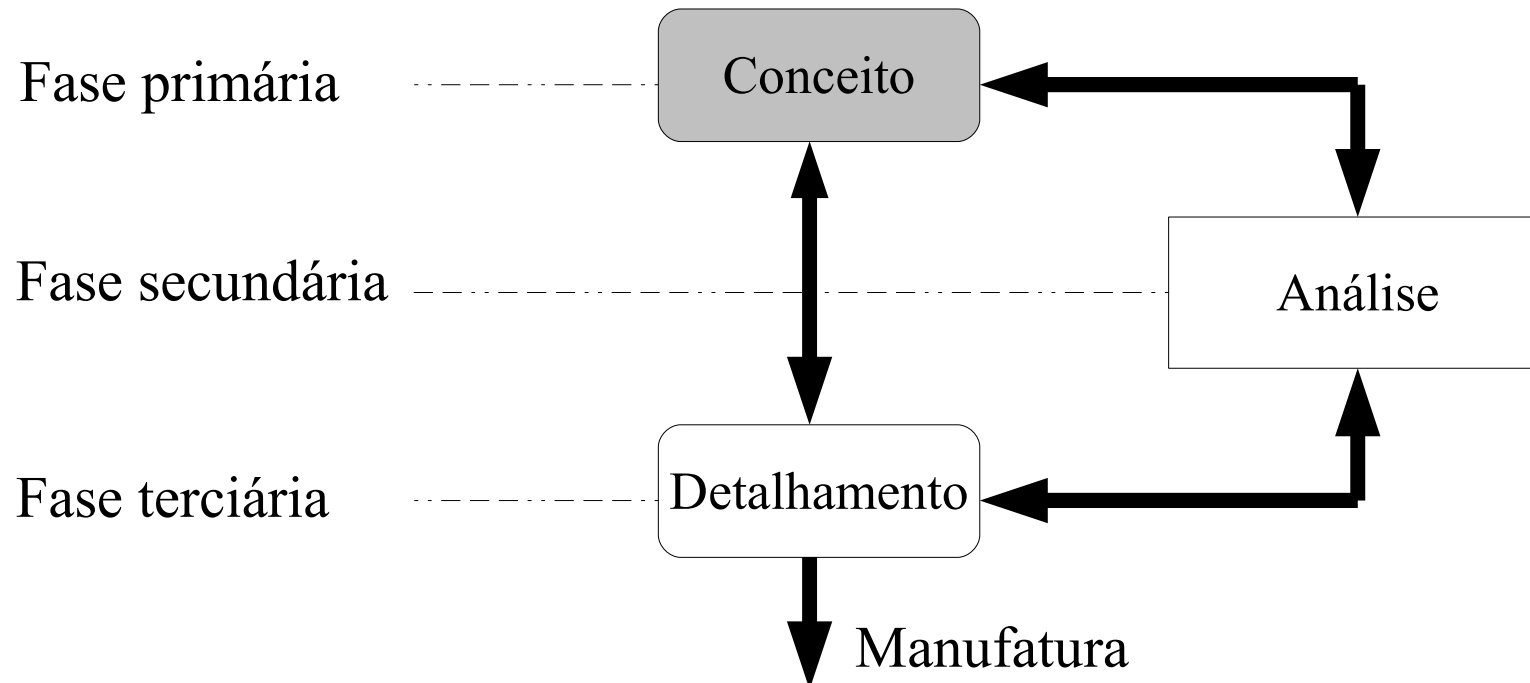
Escopo do Curso PCR



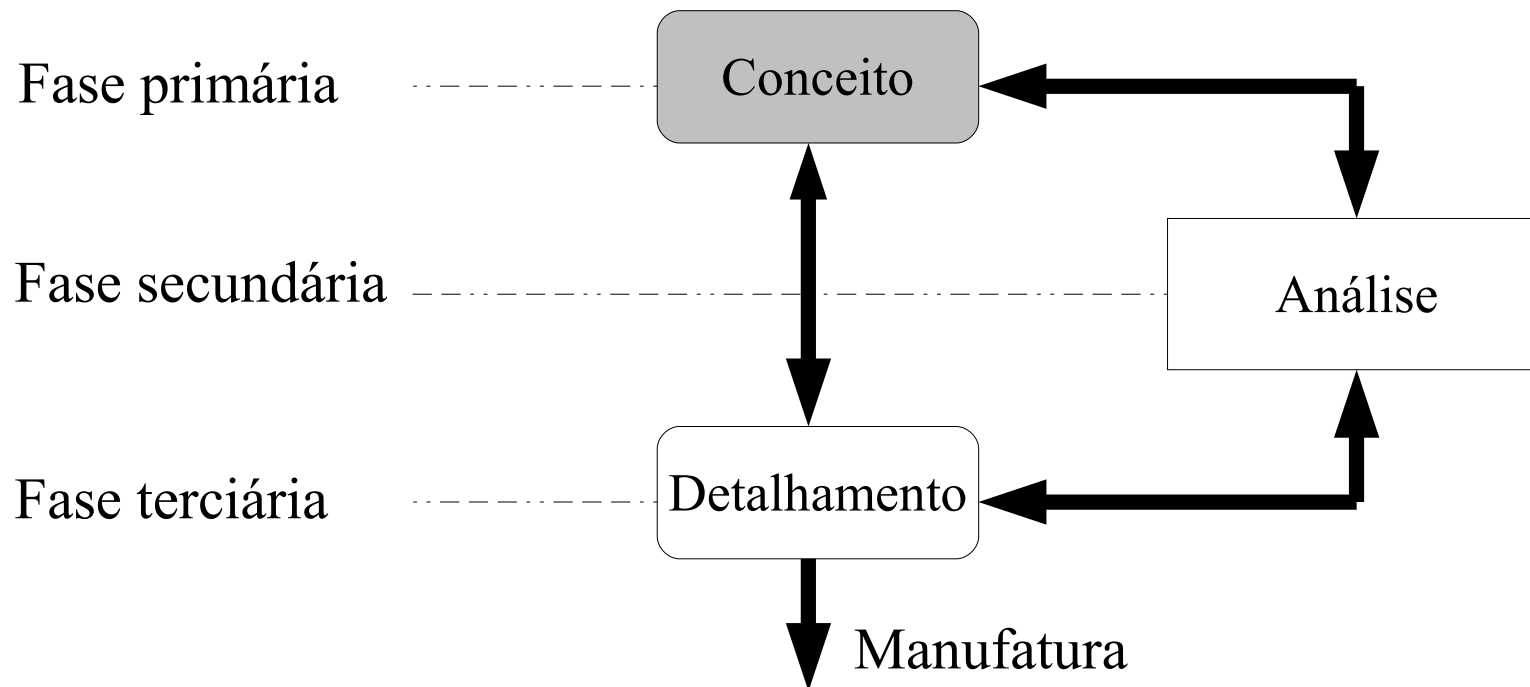
Modelo de Medland

Enquanto que o modelo de Suzim e de Gajski-Kuhn foram elaborados no contexto de projeto de sistemas eletrônicos digitais, o modelo de Medland originou-se primordialmente do projeto mecânico auxiliado por computador.

Este modelo, de propósito mais geral, está baseado em três subprocessos: ***conceito, análise e detalhamento***. Os processos evoluem em paralelo, as setas indicam o fluxo de informações.



Modelo de Medland



Subprocessos:

Conceito envolve atividades destinadas a formular e limitar o escopo do problema.

Detalhamento inclui atividades de teste de viabilidade de implementação dos resultados da fase conceitual.

Funções objetivo para Sistemas Digitais

Um bom projetista faz um projeto ótimo.

- A partir de uma descrição abstrata elabora uma descrição detalhada
- Sistema final com a funcionalidade desejada
- Sistema final com **custo mínimo e desempenho máximo**

Critérios de otimalidade:

- *Espaço*: o sistema digital deve ser o menor possível.
- *Desempenho*: o sistema digital deve ser o mais veloz possível.
- *Energia*: o sistema digital deve consumir o mínimo de energia por unidade de tempo (potência máxima).
- *Robustez*: o sistema digital não deve falhar.
- *Testabilidade*: o sistema digital deve ser fácil de testar.
- *Custo*: o sistema digital deve ser o mais barato possível. Este critério muitas vezes está sujeito a fatores externos.

Critérios conflitantes —————> Análise de *tradeoff*

Fases de Projeto de Sistemas Digitais

- *Projeto sistêmico*: descrições com altíssimo nível de abstração. Técnicas de projeto integrado de *hardware* e *software* (HW/SW) e cossimulação. Úteis para desenho de SoCs.
- *Projeto comportamental*: é o projeto algorítmico de alto nível. Envolve técnicas tais como alocação de recursos de *hardware* para execução das funcionalidades desejadas, escalonamento de tarefas, simulação comportamental baseada em alguma linguagem de descrição de *hardware* (HDL).
- *Projeto lógico*: descrições mais detalhadas do que as anteriores, porém ainda é uma descrição abstrata. Inclui técnicas como a tradução de especificações funcionais de unidades de controle em portas lógicas e a minimização do número destas e do número de elementos biestáveis necessários.
- *Projeto físico*: compreende a geração do **conjunto completo** de informações necessárias à construção do sistema digital. Inclui técnicas que determinam a planta baixa do sistema digital, a geometria entre as interconexões físicas entre os diferentes módulos. Inclui técnicas que traduzem descrições lógicas em descrições geométricas (máscara de um CI).

Fases de Projeto de Sistemas Digitais

Projeto sistêmico

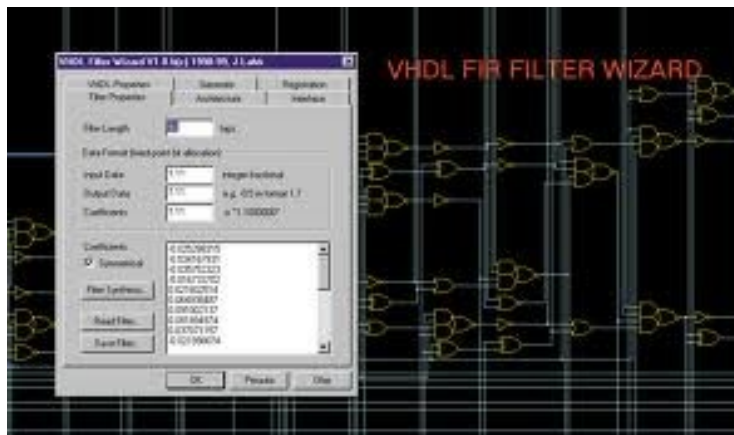


Projeto comportamental

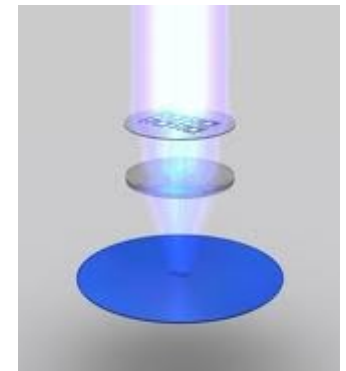
```

1 LIBRARY ieee;
2 USE ieee.std_logic_1164.all;
3 USE ieee.std_logic_arith.all;
4 USE ieee.std_logic_unsigned.all;
5
6 ENTITY adder IS
7     PORT(
8         A      IN      std_logic_vector ( 7 DOWNTO 0 ),
9         B      IN      std_logic_vector ( 7 DOWNTO 0 ),
10        clk     IN      std_logic,
11        rst_n   IN      std_logic,
12        carry   OUT     std_logic,
13        sum     OUT     std_logic_vector ( 7 DOWNTO 0 )
14    );
15 END adder ;
16
17 ARCHITECTURE rtl OF adder IS
18     signal sum_int : std_logic_vector (8 DOWNTO 0);
19     signal all      : std_logic_vector (8 DOWNTO 0);
20     signal B0       : std_logic_vector (8 DOWNTO 0);
21 BEGIN
22
23     A0 <= '0' & A;
24     B0 <= '0' & B;
25     sum_int <= A0 + B0;
26
27     adder_process : process (clk, rst_n)
28     begin
29         if rst_n = '0' then
30             carry <= '0';
31             sum <= "00000000";
32             while all = sum_int and clk = '1' loop
33                 carry <= sum_int(8);
34                 sum <= sum_int(7 DOWNTO 0);
35             end if;
36         end process adder;
37 END ARCHITECTURE rtl;
    
```

Projeto lógico



Projeto físico



Ferramentas de Síntese Lógica

Register Transfer Level (RTL): é a descrição de um circuito síncrono digital. O RTL descreve o comportamento do circuito em termos de fluxo de sinais ou transferência de dados entre os registradores e a os blocos que implementam operações lógicas.

Síntese lógica:

Transformação de um código RTL para o nível de portas lógicas. Recebe como entrada um componente RTL e entrega como saída um componente de *Netlist*

Síntese de alto nível:

Ferramentas que transformam um circuito descrito em alto nível (ANSI C/C++, SystemC) para nível RTL.

Ferramentas de Síntese Lógica.

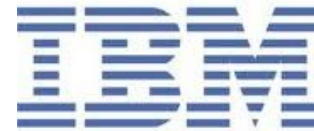
- Ferramentas para ASIC

The Synopsys logo, consisting of the word 'SYNOPSYS' in a bold, blue, sans-serif font with a registered trademark symbol.

Design Compiler

The Cadence logo, featuring the word 'cadence' in a lowercase, black, sans-serif font with a red horizontal bar above the 'a' and a trademark symbol.

Encounter RTL Compiler

The IBM logo, consisting of the letters 'IBM' in a blue, striped, sans-serif font.

BooleDozer

The Magma logo, featuring the word 'MAGMA' in a bold, black, sans-serif font with a red stylized 'M' icon to the left.

TalusDesign

- Ferramentas para FPGAs

The Xilinx logo, consisting of a red stylized 'X' icon followed by the word 'XILINX' in a bold, black, sans-serif font with a registered trademark symbol.

XST

The Altera logo, featuring the word 'ALTERA' in a blue, outlined, sans-serif font.

Quartus II

The Lattice Semiconductor Corporation logo, featuring a yellow grid icon to the left of the word 'Lattice' in a bold, blue, sans-serif font, with 'Semiconductor Corporation' in a smaller font below it.

ispLEVER

The Mentor Graphics logo, consisting of the words 'Mentor Graphics' in a bold, red, sans-serif font with a registered trademark symbol.

LeonardoSpectrum

Ferramentas EDA (*electronic design automation*)

- Ferramentas para ASIC

SYNOPSYS®

Design Compiler
Astro - place and route
Hercules - physical
verification
Proteus OPC
HSPICE
HSIM
Cosmos – Scope

cādence™

Encounter RTL Compiler
Encounter - RTL Compiler Physical
Encounter - Conformal Low Power
Encounter – Nanoroute
Encounter - Digital IC design
Allegro - PC/MCM design
SPECCTRA Autorouter
Orcad
Virtuoso - IC Artist
Virtuoso - IC Layout
Design IP

Mentor
Graphics®

Catapult – high level
synthesis
Calibre - physical
verification
ModelSim
QuestaSim – Digital and
mixed signal
Olympus-SoC - place and
route
Nucleus EDGE
Board Station - PCB

Ferramentas EDA (*electronic design automation*) para FPGAs



ISE Development tool
EDK Embedded Development tool
System Generator
Core Generator
ChipScope,
Vivado

Famílias:
Spartan, Virtex
Kintex, Artix
Zynq

Características:
Reconfiguração dinâmica
DSP, SoC, High Density



QuartusII
SOPC Builder
Qsys
DSP Builder

Famílias:
Cyclone
Arria
Stratix

Características:
HardCopy
DSP, SoC, High Density



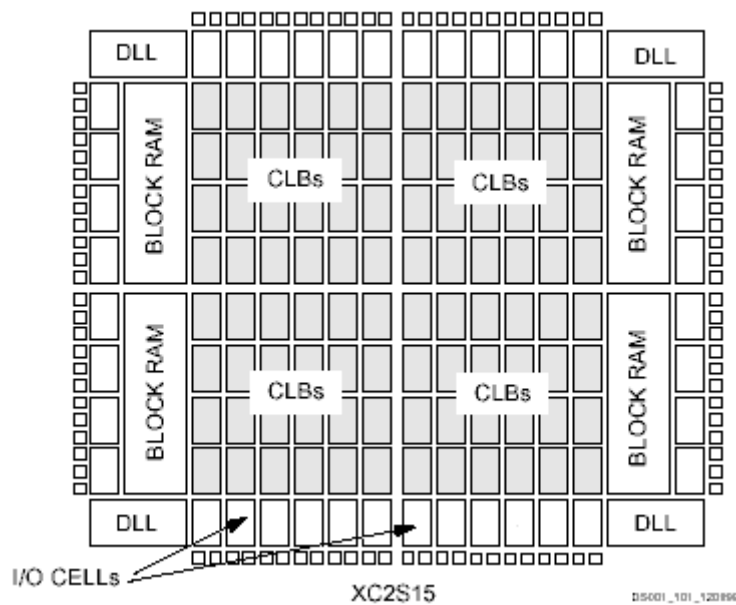
Libero SoC
Libero IDE
Smart Time
Smart Power

Famílias:
ProASIC3
IGLOO
SmartFusion

Características:
UltraLowPower, SoC
MixedSignal, Antifuse

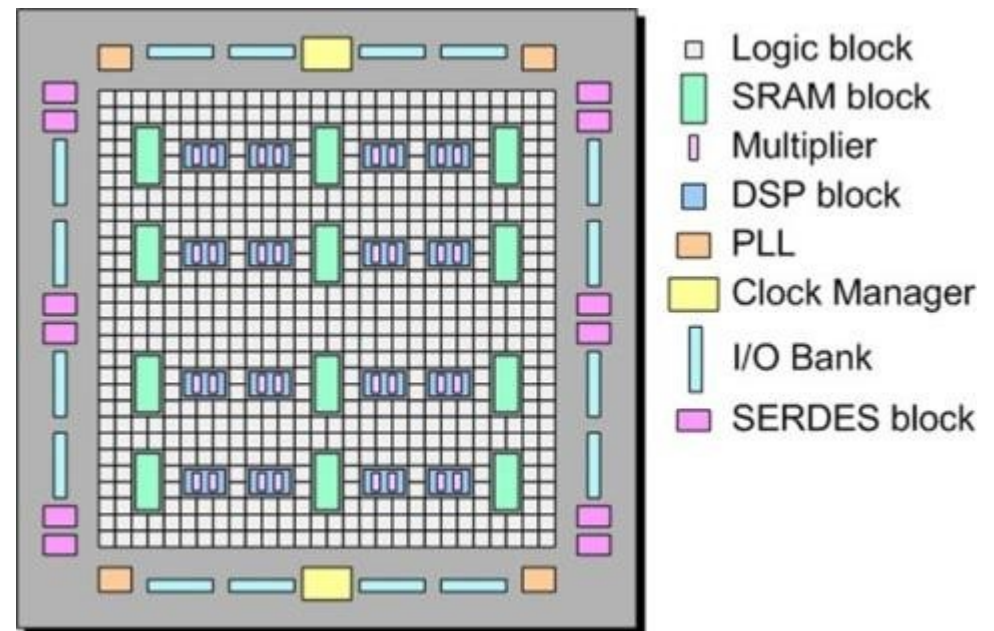
Estrutura interna de um FPGA

Xilinx XC2S15 (1985)



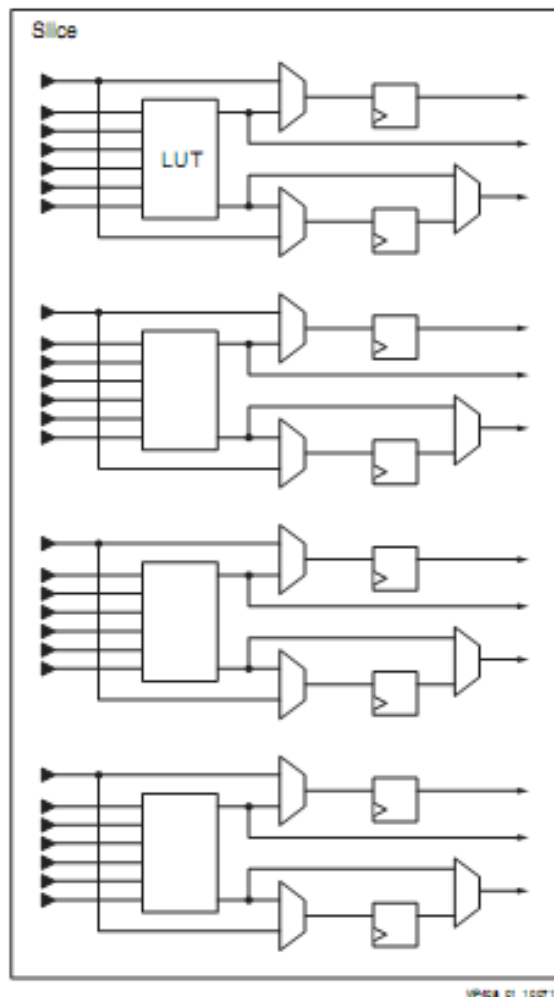
LUTs de 4 entradas

FPGAs em 2012



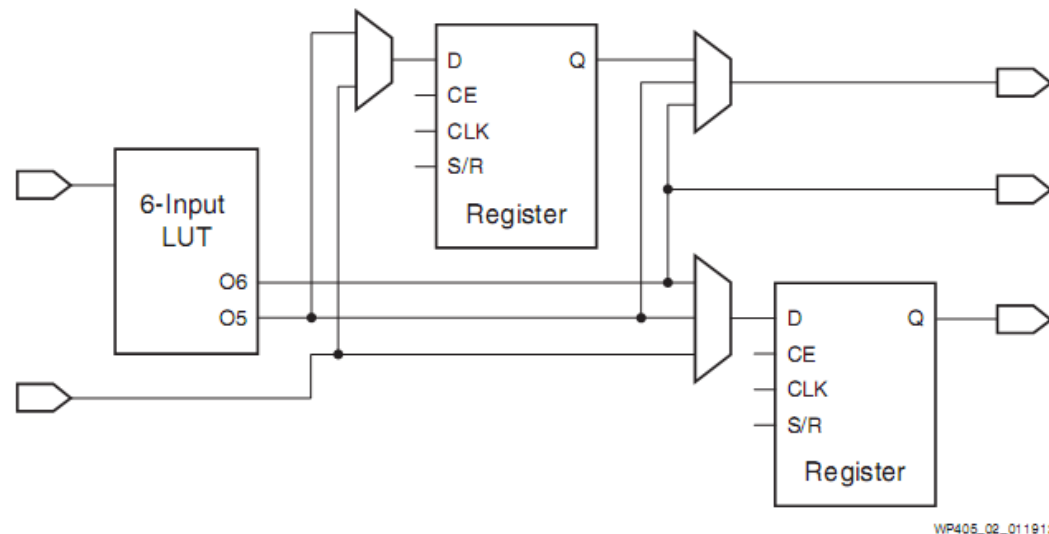
LUTs com 5, 6, 7 e 8 entradas
Centos de somadores, multiplicadores
Colunas de DPSs e RAMs
Clock manager (DCM: DLLs, clk deskewing,...
PCI Express, Processadores embarcados

Estrutura interna de um FPGA



Familia 7 - Xilinx:

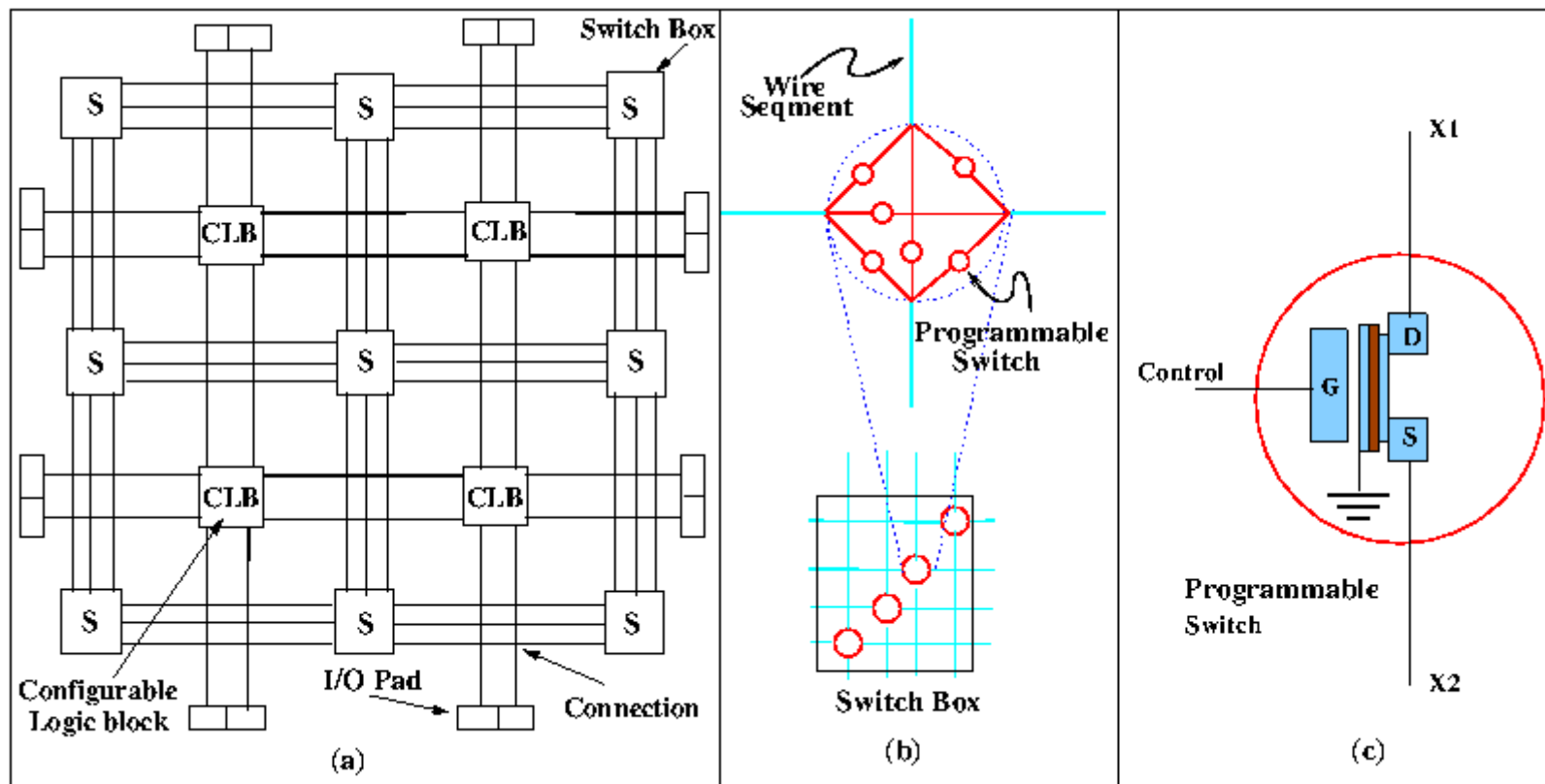
Cada bloco lógico configurável (CLB) tem 2 Slices.
Cada Slice tem 4 LUTs de 6 entradas e 8 registradores.



WP405_02_011912

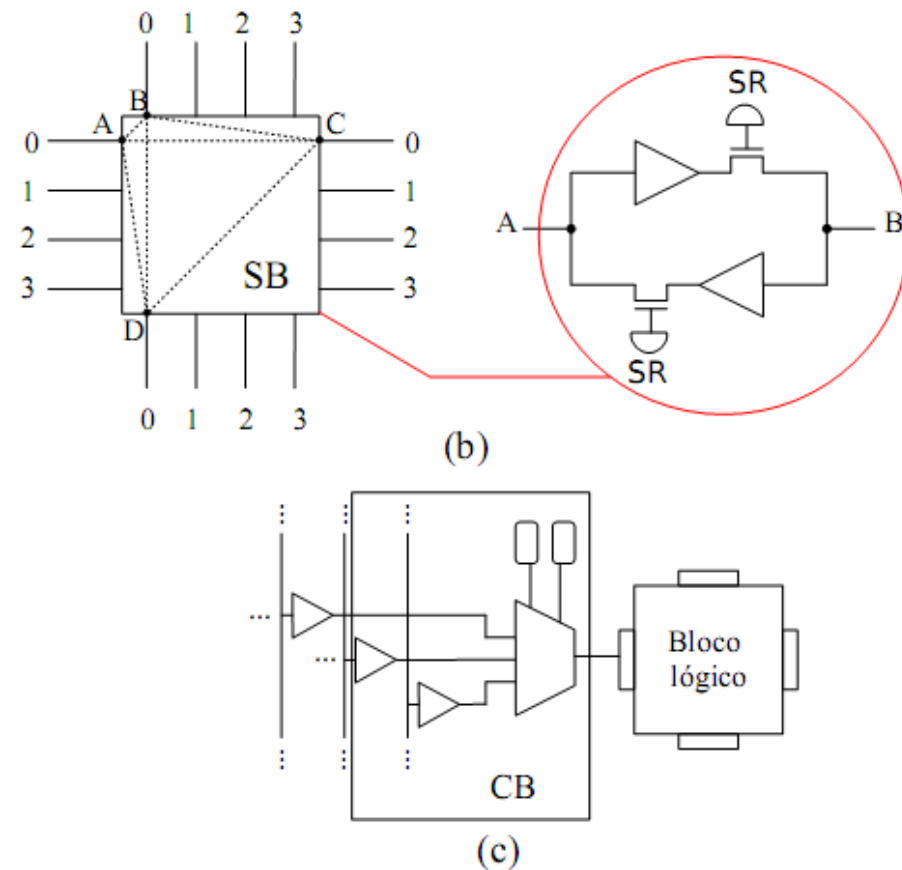
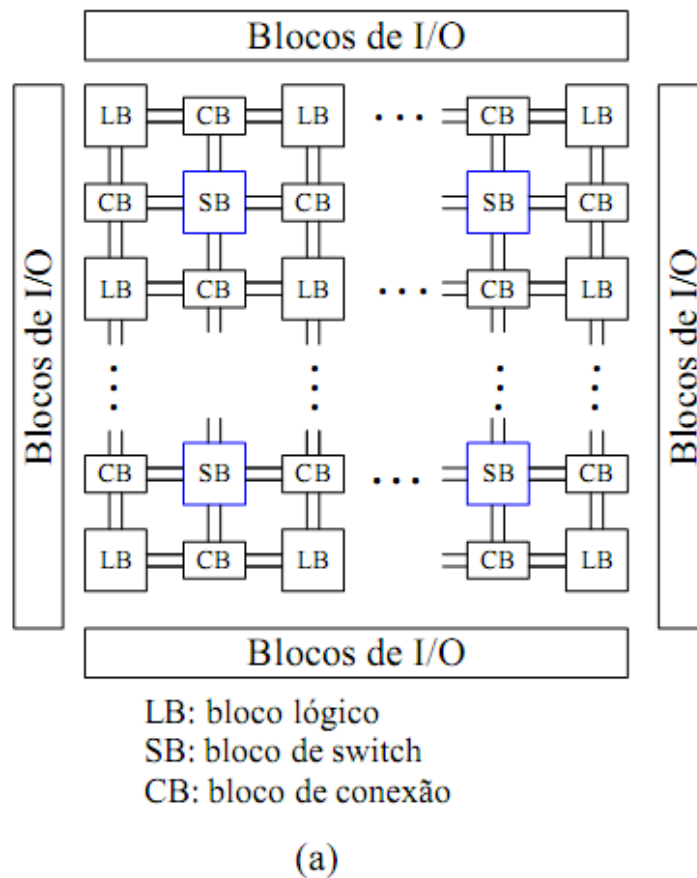
Estrutura interna de um FPGA

Blocos de conexão e Blocos de roteamento



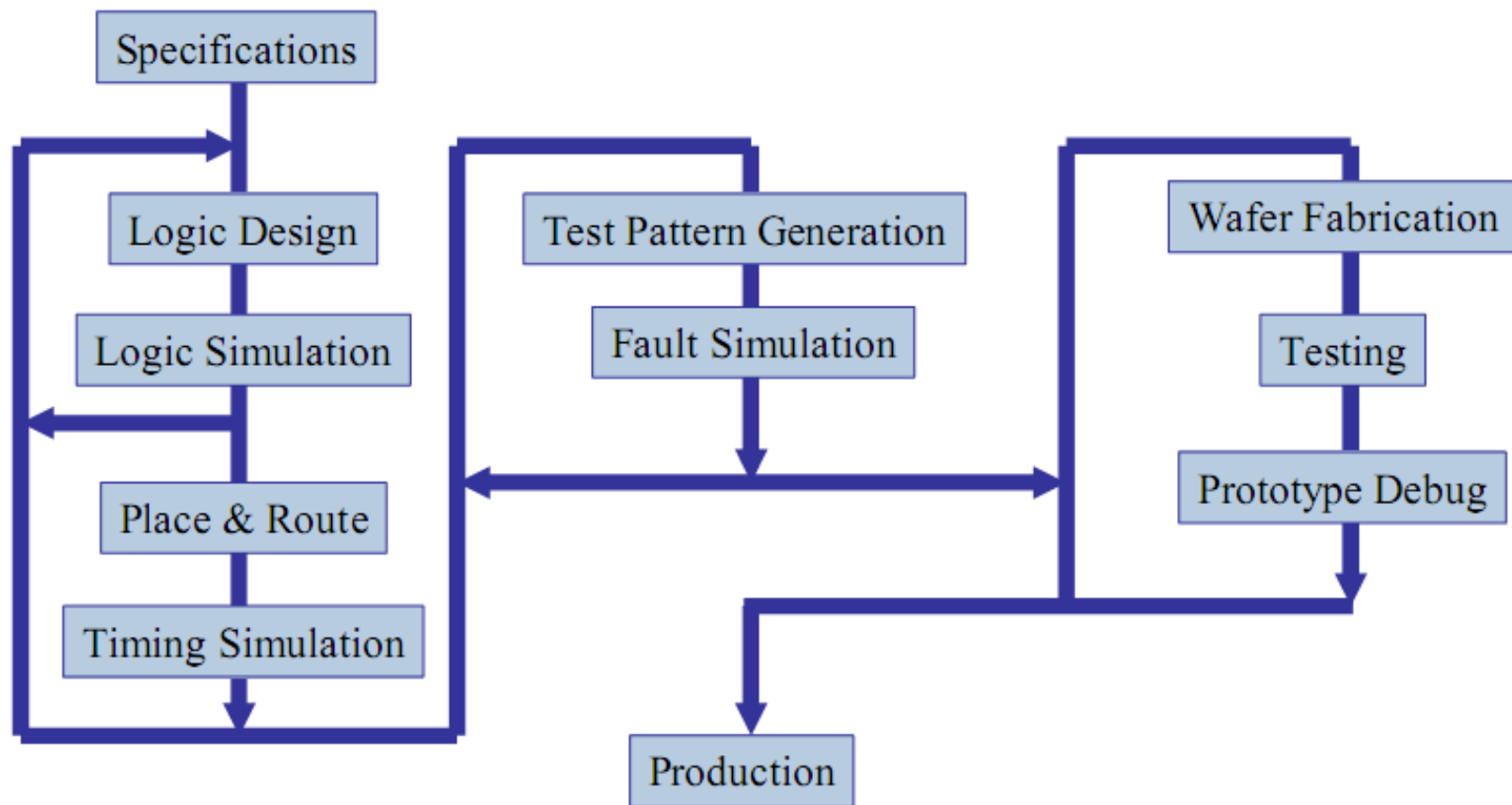
Estrutura interna de um FPGA

Blocos de conexão e Blocos de roteamento

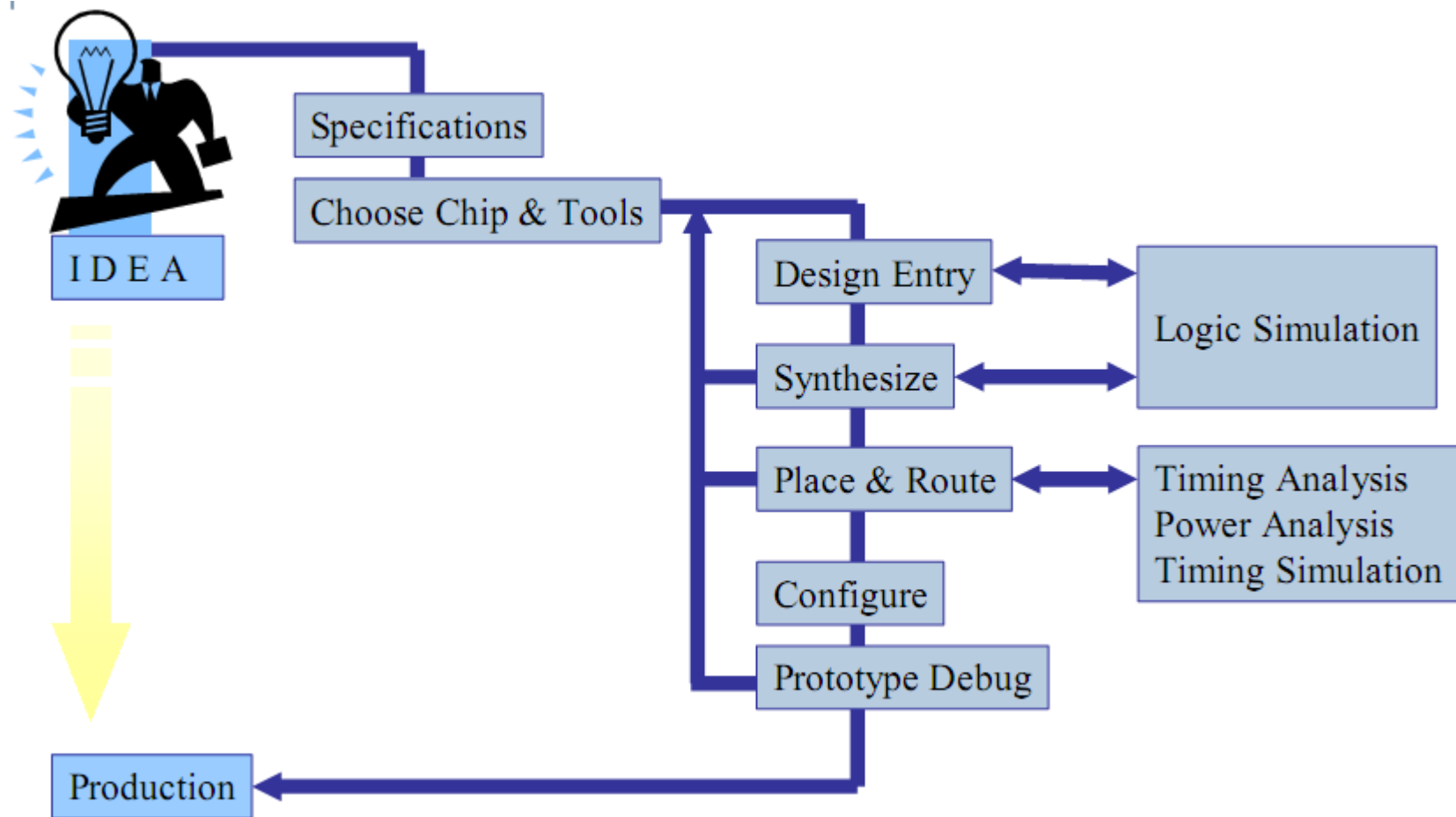


(a) Estrutura geral da FPGA (b) bloco de switch (c) bloco de conexão

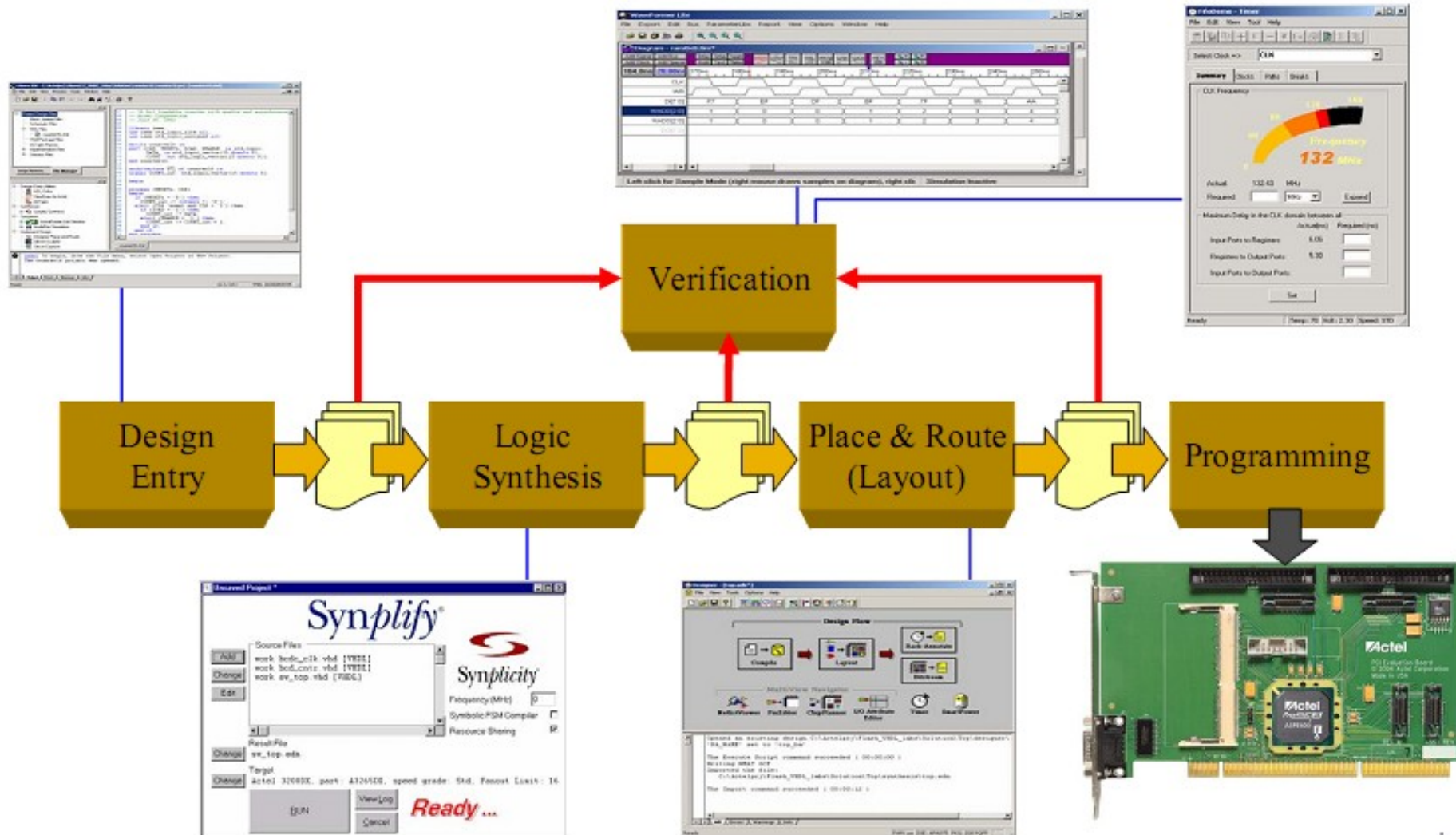
Fluxo de Projeto para ASICs



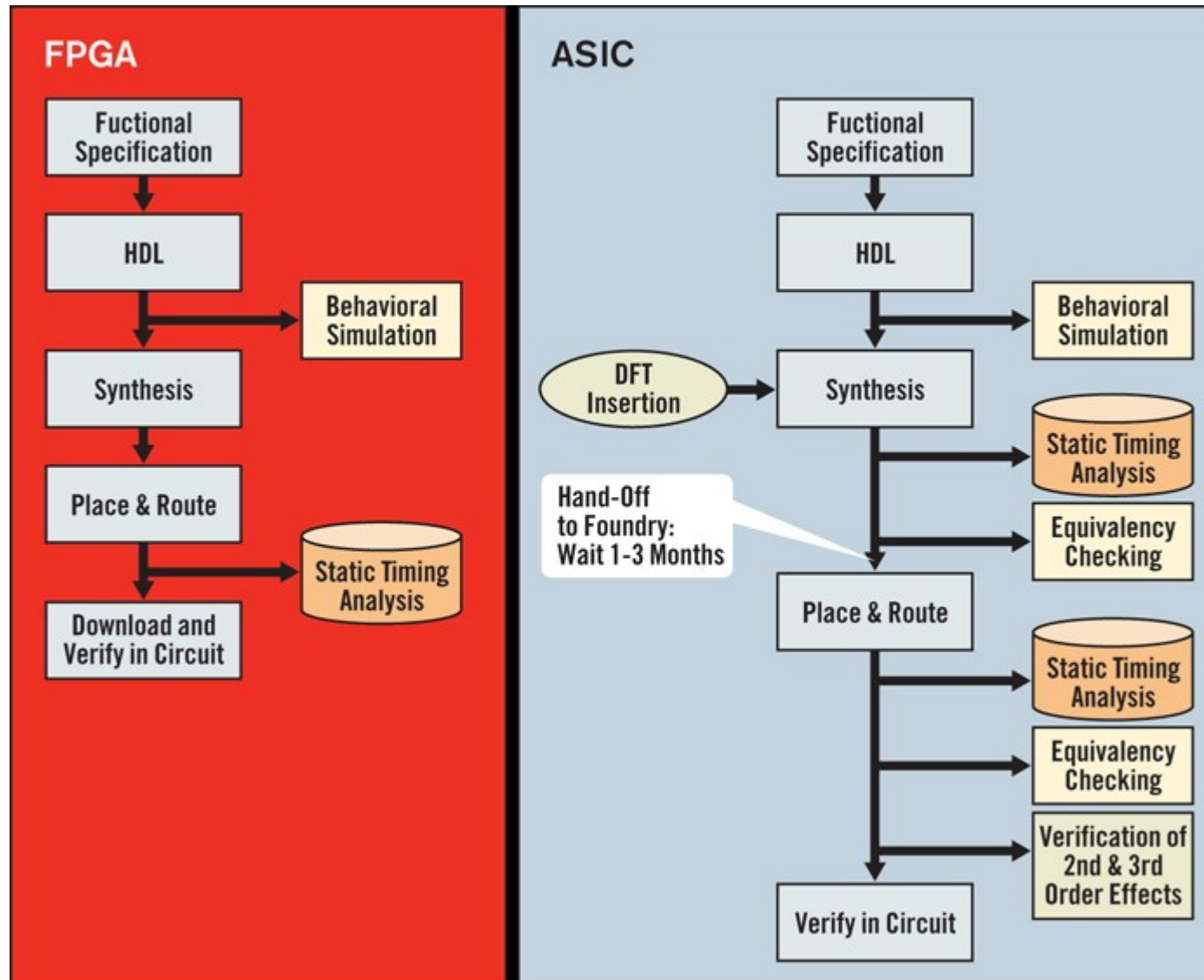
Fluxo de Projeto para FPGAs



Fluxo de Projeto para FPGAs



Comparação do fluxo de Projeto FPGAs vs ASIC



Questionário

1. O que é processo de projeto de sistemas digitais?
2. Quais são os níveis de abstração no projeto de sistemas digitais?
3. Quais são os domínios de descrição?
4. Como funciona o Diagrama Y?
5. Quais são as variáveis de desenho de circuitos digitais?
6. Quais são as fases de projeto de sistemas digitais?
7. O que é descrição RTL?
8. O que é síntese lógica?
9. O que é síntese de alto nível?
10. Como funciona um FPGA?
10. Quais as principais diferenças entre o fluxo de projeto para ASICs e o fluxo de projeto em FPGAs?

Níveis de desenho

