

# Introdução à computação reconfigurável

## Introdução aos kits de desenvolvimento e software de design

---

AULA 1

IOUL IIA SKL I AROVA

# Computação Reconfigurável

---

Uma arquitetura de computador que facilita a computação mais rápida e complexa usando circuitos integrados reprogramáveis para processar dados que resolvem problemas e direcionam funções específicas.

Oferece maior velocidade e eficiência do hardware do computador para executar tarefas seriais e paralelas, combinadas com a capacidade flexível do software para permitir alterações na finalidade de um circuito no campo para se ajustar às mudanças nas circunstâncias.

Principais vantagens:

- Mais funcionalidade em designs de hardware mais simples e menores
- Economia de custos em produtos de baixo volume e aqueles cuja vida útil é prolongada pela atualização
- Tempo de lançamento no mercado mais curto/mais rápido

# Ideia original

O primeiro trabalho sobre arquitetura de computador reconfigurável foi feito na Universidade da Califórnia em Los Angeles – em 1960, Gerald Estrin propôs um processador padrão estendido com uma série de hardware “variável”.

O principal objetivo era permitir cálculos que estão além das capacidades dos sistemas atuais, fornecendo um inventário de subestruturas de alta velocidade e regras para interconectando-os de tal forma que todo o sistema possa ser temporariamente distorcido em um computador para fins especiais orientado a problemas.

Naquela época, a tecnologia digital não estava preparada para uma mudança tão revolucionária.

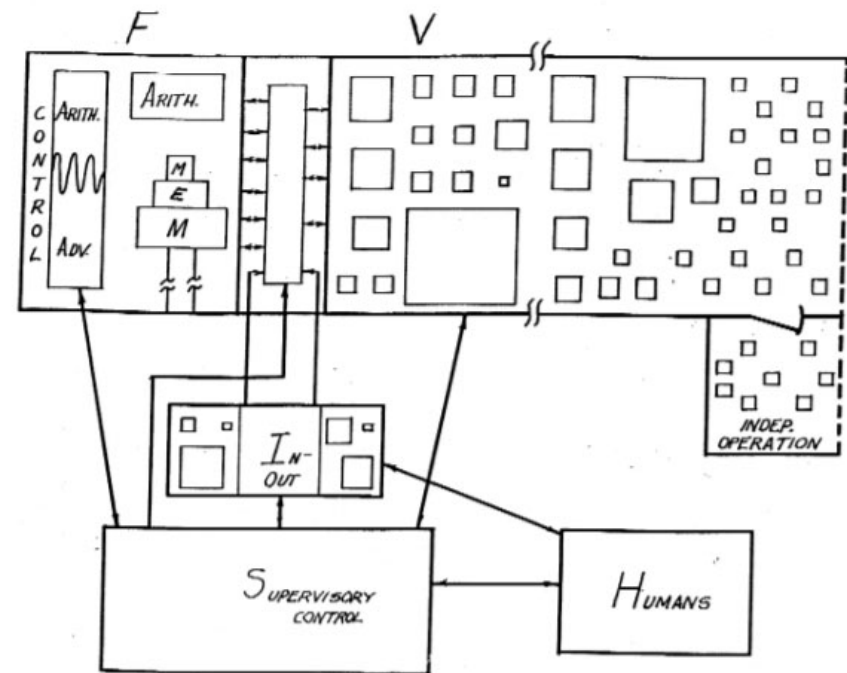


Figure 1. This diagram shows the relations between the fixed machine, the variable structure part, the input-output, the supervisory control, and the humans. (Courtesy of the 1960 Western Joint Computer Conference.)

# Outros modelos de computação

---

Tradicionalmente, a computação era classificada em Computação de Propósito Geral, executada por um **Processador de uso geral (GPP)** e computação específica de aplicação realizada por um **Circuito Integrado Específico de Aplicação (ASIC)**.

A computação reconfigurável atua como uma compensação entre as duas características extremas do GPP e do ASIC, combinando as vantagens de ambos.

Quando comparado ao GPP:

- fornece capacidade de fazer alterações substanciais no próprio caminho de dados, além do fluxo de controle
- pode ter melhor desempenho em relação a uma implementação de software em GPP, mas pagando isso em termos de tempo para implementar

Quando comparado ao ASIC:

- possibilidade de adaptar o hardware <durante o tempo de execução> "carregando" um novo circuito na malha reconfigurável
- pode ser usado para projetar um sistema sem exigir o mesmo tempo e complexidade de design em comparação com uma solução totalmente personalizada, mas sendo derrotado em termos de desempenho

# Visão geral

---

## Vantagens / Recursos

- Co-design flexível de hardware/software
- Customização (exploração de paralelismo, controle especializado e caminhos de dados)
- Prototipagem rápida
- Ciclos de projeto mais rápidos, menor risco, mais barato para mercados de pequeno e médio porte em comparação com ASICs
- Atualizações dinâmicas de sistema (atualizações de campo para hardware e software)

## Aplicativos de destino

- Aceleradores/coprocessadores/processadores de hardware específicos da aplicação
- Telecomunicações, redes, processamento de imagem/áudio, criptografia, espaço, automotivo, médico, etc.

# Visão geral

---

## Limitações

- Sobrecarga devido ao tempo de reconfiguração e/ou atrasos adicionais no circuito de hardware

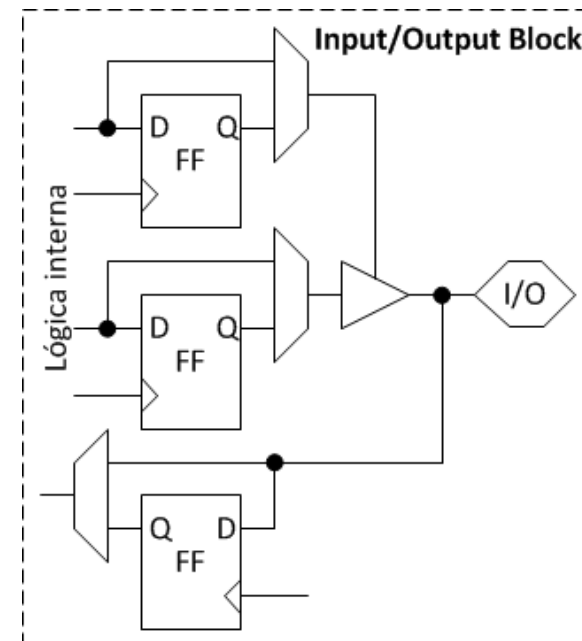
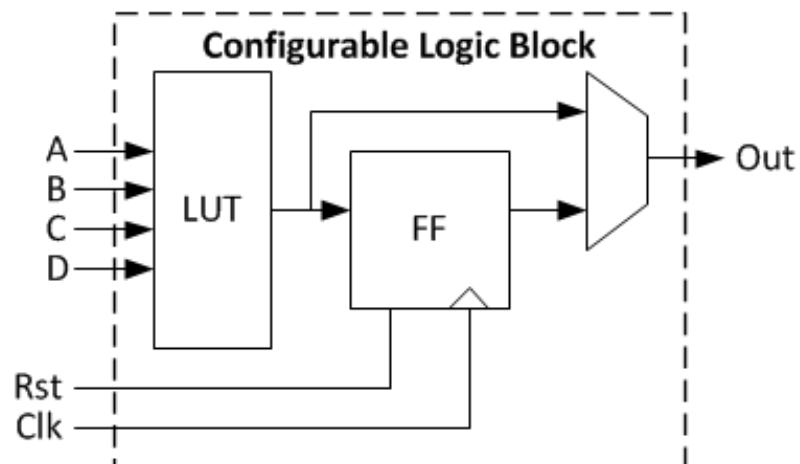
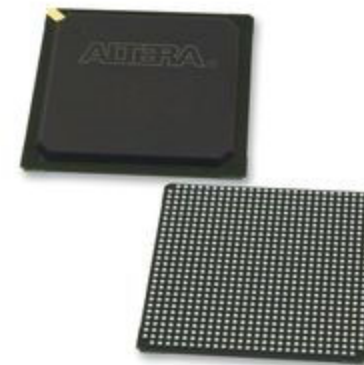
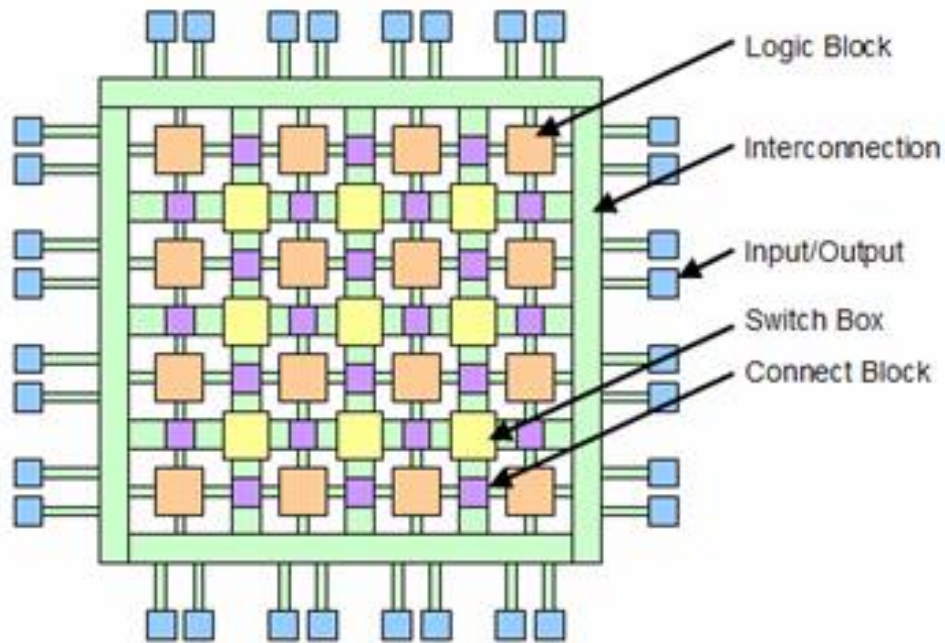
## Tecnologias de apoio

- FPGAs/SoC programáveis de alta capacidade
- Estruturas de personalização do processador
  - Núcleos de processador fixos + coprocessadores de extensão/processadores flexíveis

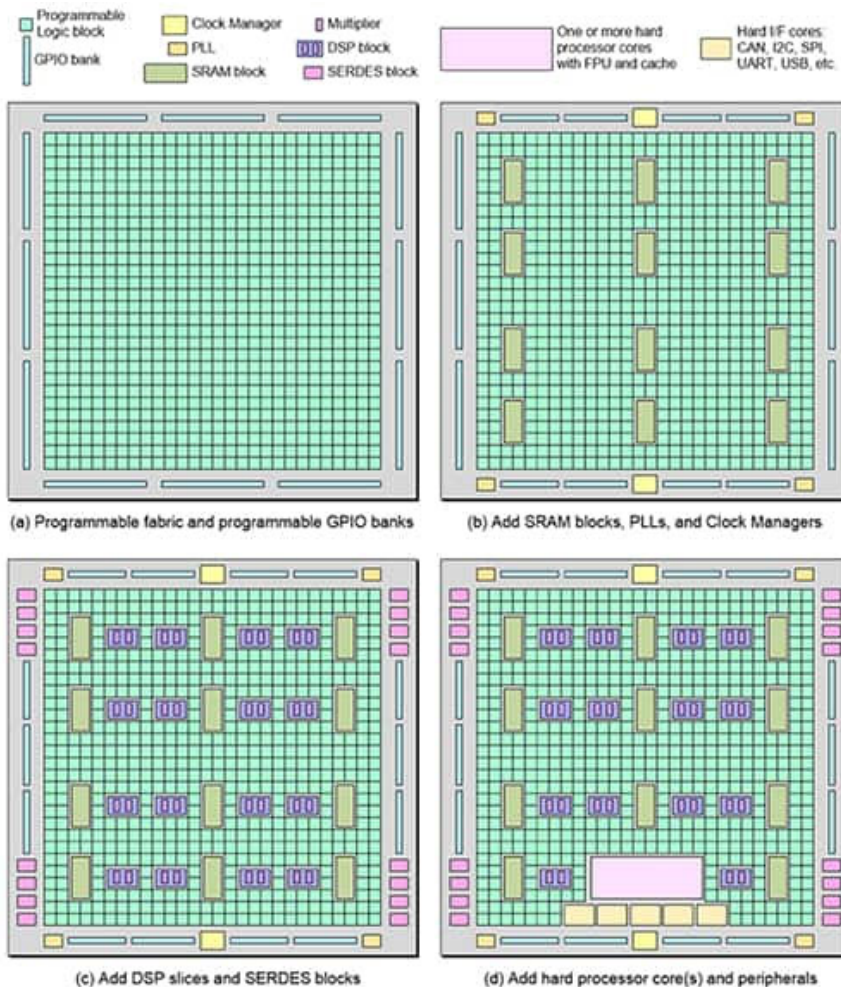
## Abordagens de reconfiguração

- Estático (na fabricação, implantação ou inicialização)
- Dinâmico
  - Total (em tempo de execução entre as fases de execução)
  - Parcial (em tempo de execução substituindo alguns componentes, enquanto outros permanecem operacionais)

# FPGA – Matriz de portas programáveis em campo



# FPGA moderno



Os FPGAs mais simples contêm apenas malha programável e IO de uso geral configurável (GPIO) (a); diferentes arquiteturas aumentam essa estrutura fundamental com blocos de memória, PLLs e gerenciadores de clock (b); Blocos DSP e interfaces SERDES (c); e núcleos de processador rígido e periféricos (d). (Fonte da imagem: Max Maxfield)

LUTs de 6 entradas configurável como memória distribuída

Blocos de memória incorporados- bloco de RAM com lógica FIFO integrada para buffer de dados no chip

Conectividade serial de alta velocidade com transceptores multi-gigabit

Configurável pelo usuário interfaces analógicas, incorporando conversores analógico-digitais

Fatias DSP com multiplicadores dedicados para filtragem de alto desempenho

Blocos de gerenciamento de relógio fornecendo funcionalidade de síntese de frequência de clock, enquadramento e filtragem de jitter

Processadores suaves e rígidos

- =>FPGAs com núcleos de processador rígidos são chamados desistemas programáveis em chip (PSoC)



# Dispositivos programáveis Xilinx

O desempenho e os recursos das ofertas de dispositivos programáveis da Xilinx variam de modestos a extremamente altos:

- FPGAs tradicionais
- PSoCs (fabric programável FPGA com um único processador de núcleo rígido)
- MPSoCs (tecido programável FPGA com vários processadores de núcleo rígido)
- RFSoc (MPSoCs com capacidade de RF)
- ACAPs (plataformas de aceleração de computação adaptativa)

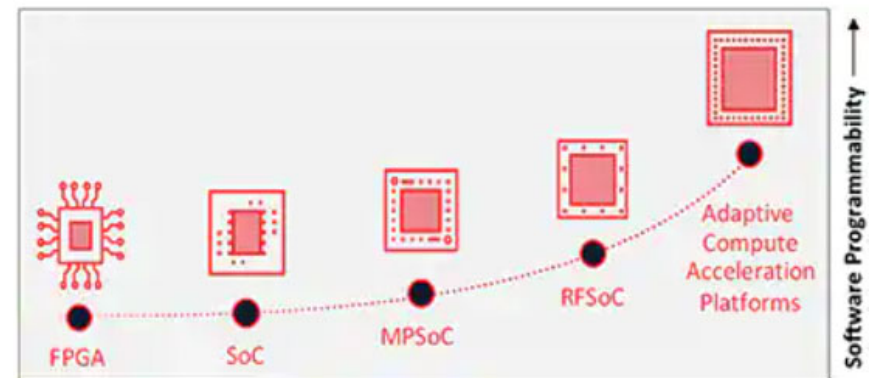


Figure 2: Over time, the Xilinx architectural portfolio has evolved from simple FPGAs containing only programmable fabric, to SoC devices in which the programmable fabric is augmented with a hard core processor, to MPSoCs with multiple processors, to RFSoc with RF capabilities, to the latest generation of ACAPs, which are targeted toward applications like AI. (Image source: Max Maxfield)

# FPGA Xilinx

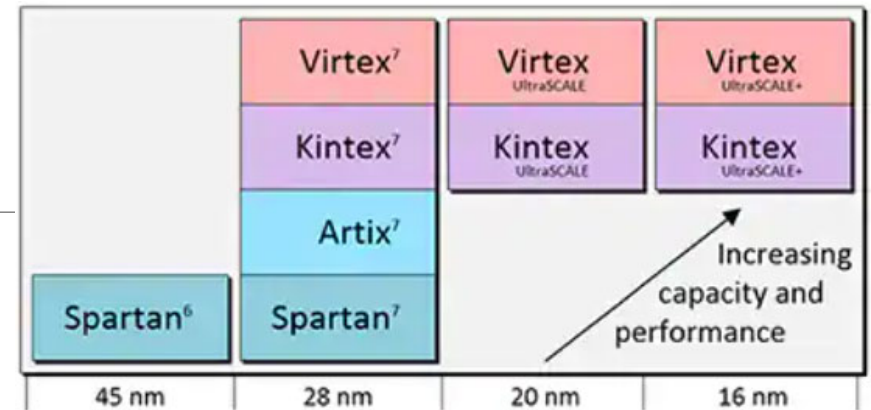


Figure 3: Xilinx FPGA offerings provide a comprehensive multi-node portfolio to address requirements across a wide set of applications. (Image source: Max Maxfield)

Table 4: Artix-7 FPGA Feature Summary by Device

| Device          | Logic Cells | Configurable Logic Blocks (CLBs) |                          | DSP48E1 Slices <sup>(2)</sup> | Block RAM Blocks <sup>(3)</sup> |       |          | CMTs <sup>(4)</sup> | PCIe <sup>(5)</sup> | GTPs | XADC Blocks | Total I/O Banks <sup>(6)</sup> | Max User I/O <sup>(7)</sup> |
|-----------------|-------------|----------------------------------|--------------------------|-------------------------------|---------------------------------|-------|----------|---------------------|---------------------|------|-------------|--------------------------------|-----------------------------|
|                 |             | Slices <sup>(1)</sup>            | Max Distributed RAM (Kb) |                               | 18 Kb                           | 36 Kb | Max (Kb) |                     |                     |      |             |                                |                             |
| XC7A12T         | 12,800      | 2,000                            | 171                      | 40                            | 40                              | 20    | 720      | 3                   | 1                   | 2    | 1           | 3                              | 150                         |
| XC7A15T         | 16,640      | 2,600                            | 200                      | 45                            | 50                              | 25    | 900      | 5                   | 1                   | 4    | 1           | 5                              | 250                         |
| XC7A25T         | 23,360      | 3,650                            | 313                      | 80                            | 90                              | 45    | 1,620    | 3                   | 1                   | 4    | 1           | 3                              | 150                         |
| XC7A35T         | 33,280      | 5,200                            | 400                      | 90                            | 100                             | 50    | 1,800    | 5                   | 1                   | 4    | 1           | 5                              | 250                         |
| XC7A50T         | 52,160      | 8,150                            | 600                      | 120                           | 150                             | 75    | 2,700    | 5                   | 1                   | 4    | 1           | 5                              | 250                         |
| XC7A75T         | 75,520      | 11,800                           | 892                      | 180                           | 210                             | 105   | 3,780    | 6                   | 1                   | 8    | 1           | 6                              | 300                         |
| <b>XC7A100T</b> | 101,440     | 15,850                           | 1,188                    | 240                           | 270                             | 135   | 4,860    | 6                   | 1                   | 8    | 1           | 6                              | 300                         |
| XC7A200T        | 215,360     | 33,650                           | 2,888                    | 740                           | 730                             | 365   | 13,140   | 10                  | 1                   | 16   | 1           | 10                             | 500                         |

## Notes:

- Each 7 series FPGA slice contains four LUTs and eight flip-flops; only some slices can use their LUTs as distributed RAM or SRLs.
- Each DSP slice contains a pre-adder, a 25 x 18 multiplier, an adder, and an accumulator.
- Block RAMs are fundamentally 36 Kb in size; each block can also be used as two independent 18 Kb blocks.
- Each CMT contains one MMCM and one PLL.
- Artix-7 FPGA Interface Blocks for PCI Express support up to x4 Gen 2.
- Does not include configuration Bank 0.
- This number does not include GTP transceivers.

# Artix-7 CLB(Bloco Lógico Configurável)

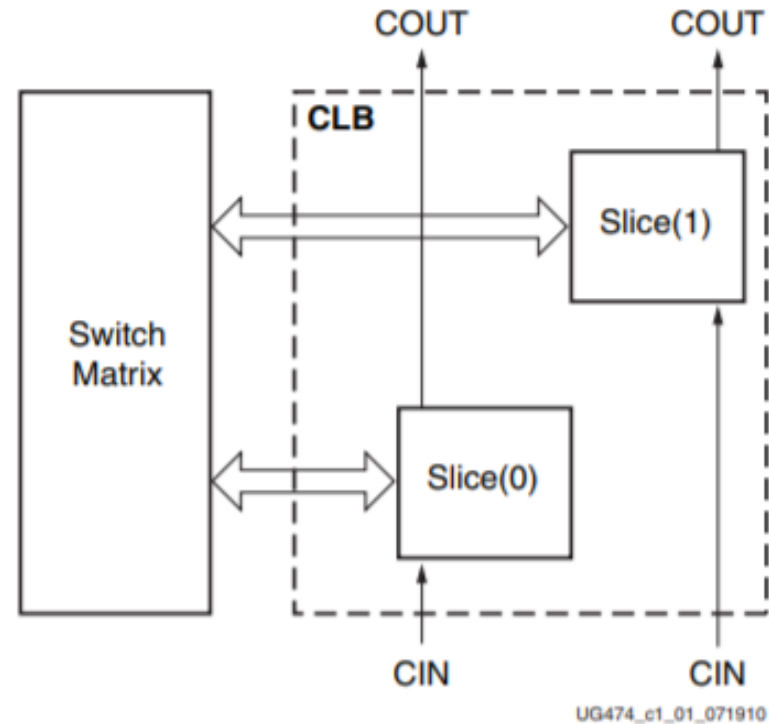
CLBs são os principais recursos lógicos para implementação de circuitos sequenciais e combinatórios.

Cada elemento CLB está conectado a uma matriz de switch para acesso à matriz de roteamento geral.

Um elemento CLB contém um par de fatias.

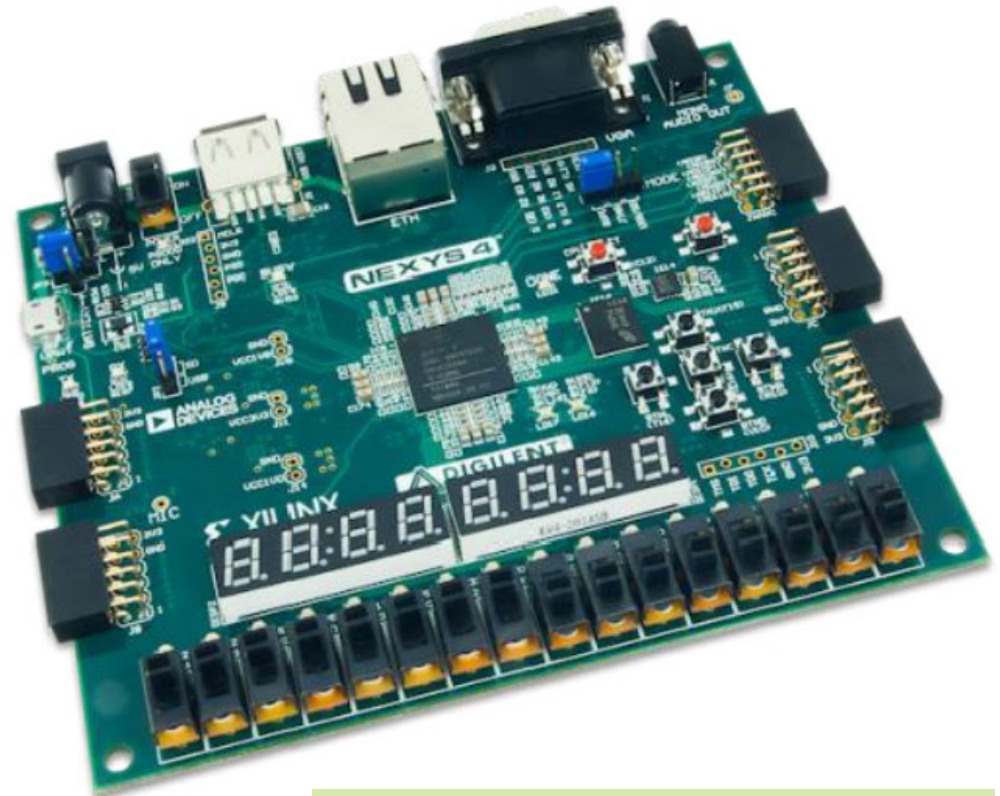
Cada fatia contém:

- Quatro geradores de funções lógicas (ou tabelas de consulta)
- Oito elementos de armazenamento
- Multiplexadores de ampla função
- Leve lógica



# Placa de Desenvolvimento Nexys-4

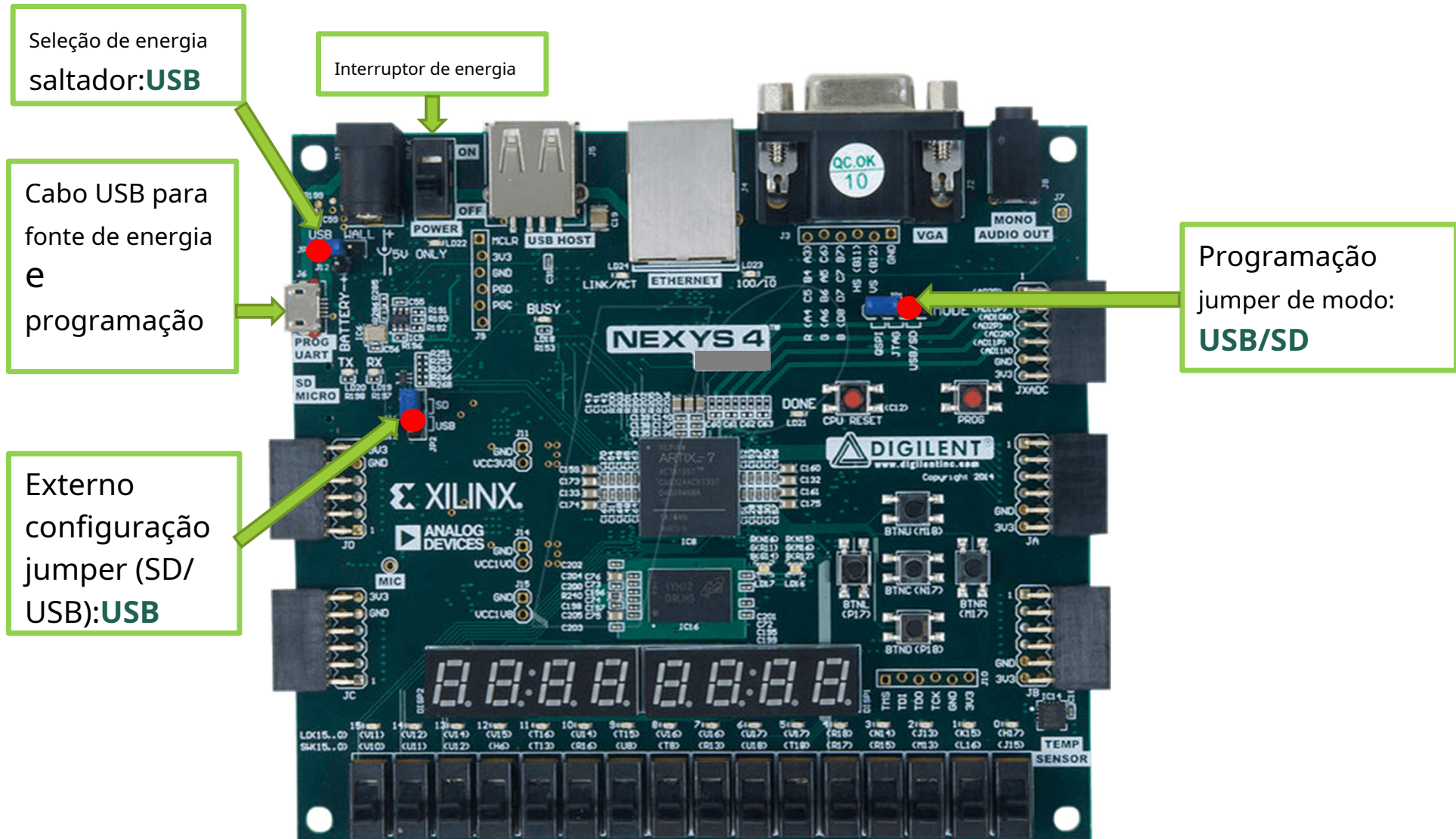
- 16 interruptores de usuário
- 16 LEDs de usuário
- Dois displays de 4 dígitos e 7 segmentos
- Ponte USB-UART
- Dois LEDs tricolores
- Conector de cartão micro SD
- Saída VGA de 12 bits
- Saída de áudio PWM
- Microfone PDM
- Acelerômetro de 3 eixos
- Sensor de temperatura
- Ethernet 10/100 física
- 16 MB de RAM celular
- Flash serial
- Quatro portas Pmod
- Pmod para sinais XADC
- Porta Digilent USB-JTAG para programação e comunicação FPGA
- Host USB HID para mouses, teclados e cartões de memória



FPGA: xc7a100Tcsg324-1



# Placa de Desenvolvimento Nexys-4



# Ferramentas de desenvolvimento

## Xilinx Vitis

- Para portabilidade de projetos entre diferentes computadores, é recomendado para instalar **versão 2022.2**
- As instruções de instalação estão disponíveis no site do curso
- Requer cerca de **65GB** de espaço em disco
- Licença: Licença de avaliação de 30 dias do Vivado Design Suite -> flutuante  
licença disponível no campus da UA ou com VPN ativa

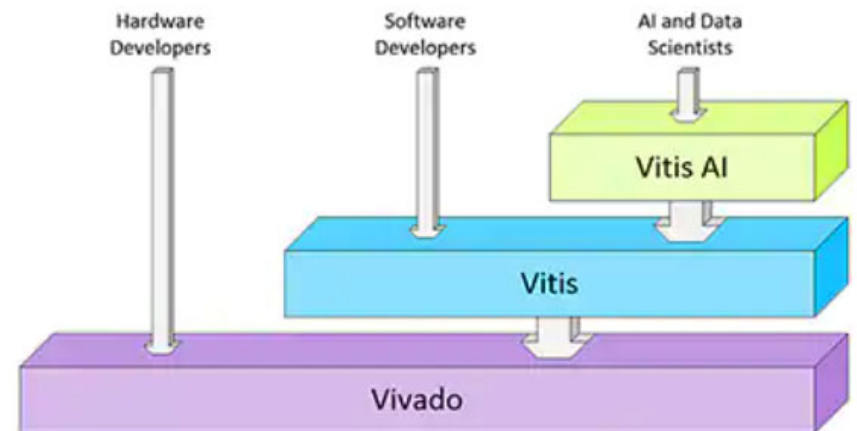
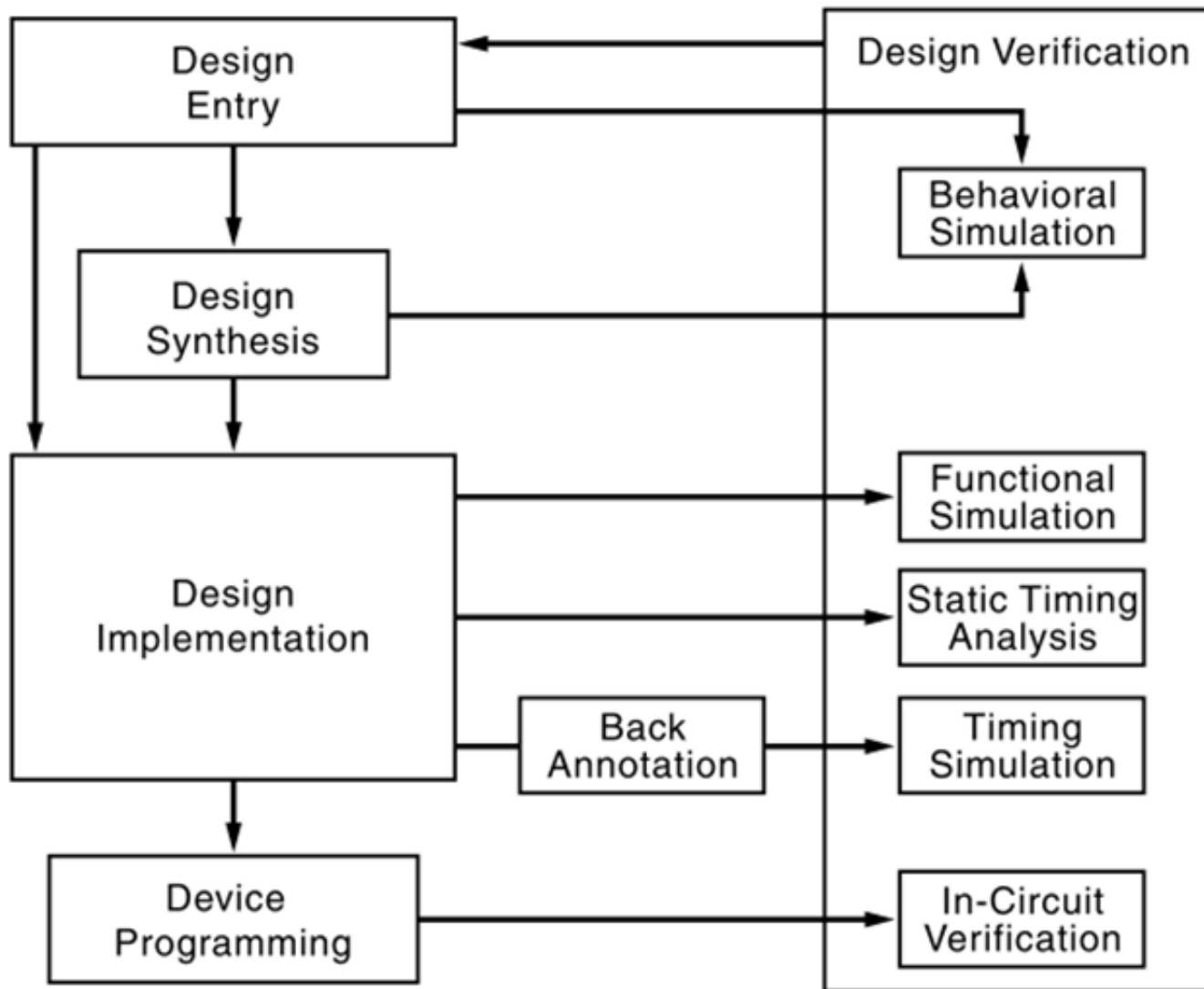


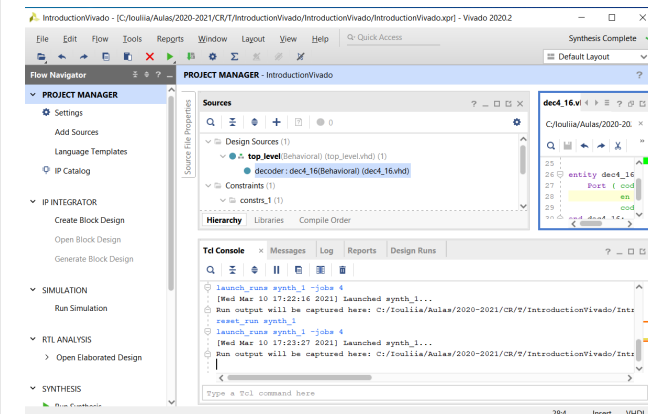
Figure 6: A high-level view of the Xilinx Vivado and Vitis design tool stack reflects how users can work with the tools at the most appropriate levels of abstraction. Hardware designers work with Vivado, software developers work with Vitis, and AI and data scientists work with Vitis AI. (Image source: Max Maxfield)

# Fluxo de projeto de FPGA

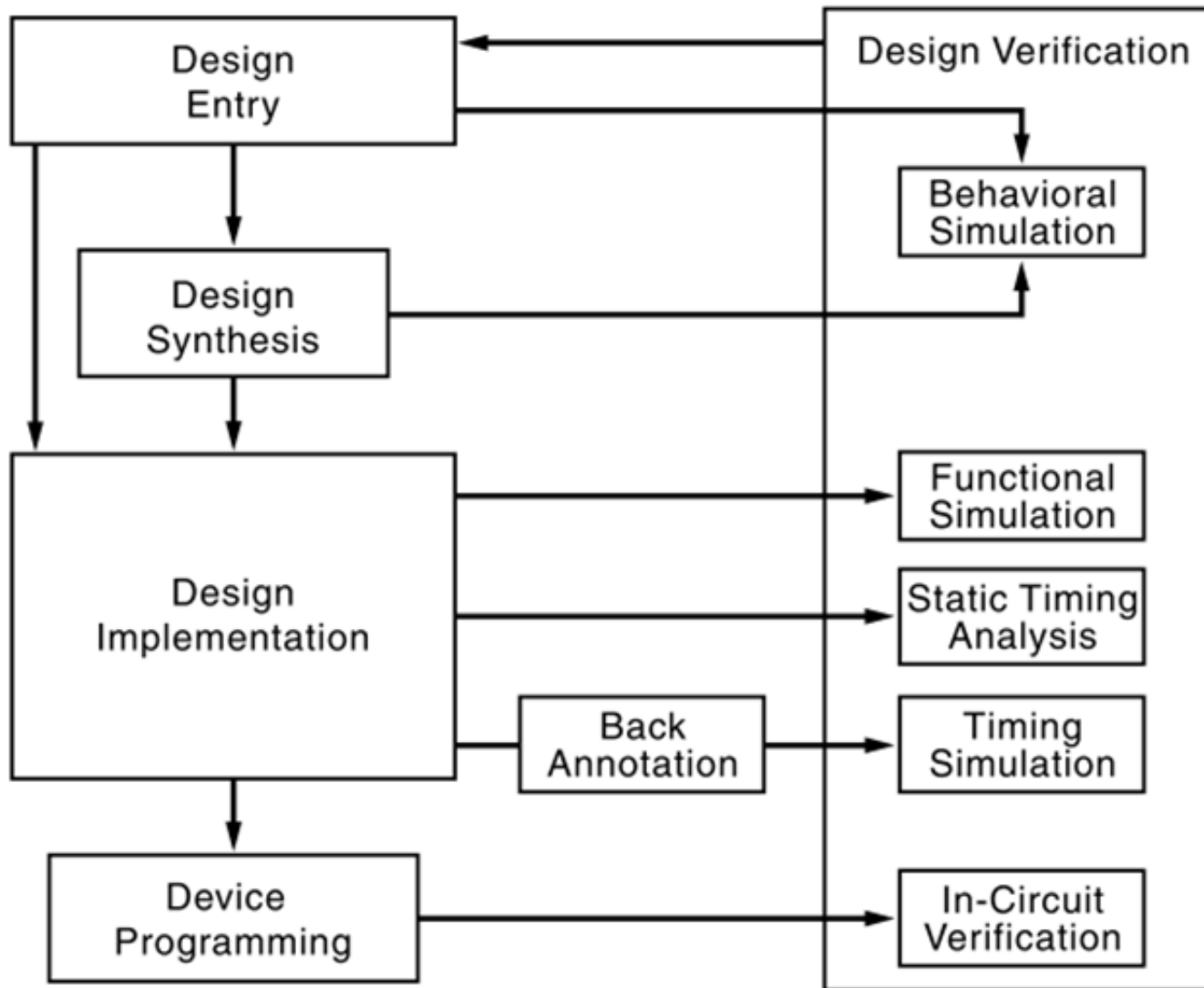


Entrada de projeto baseado em:

- Descrição do equipamento
- Línguas
- Diagramas de estado
- Captura esquemática
- C/C++ e outras linguagens de alto nível



# Síntese Lógica



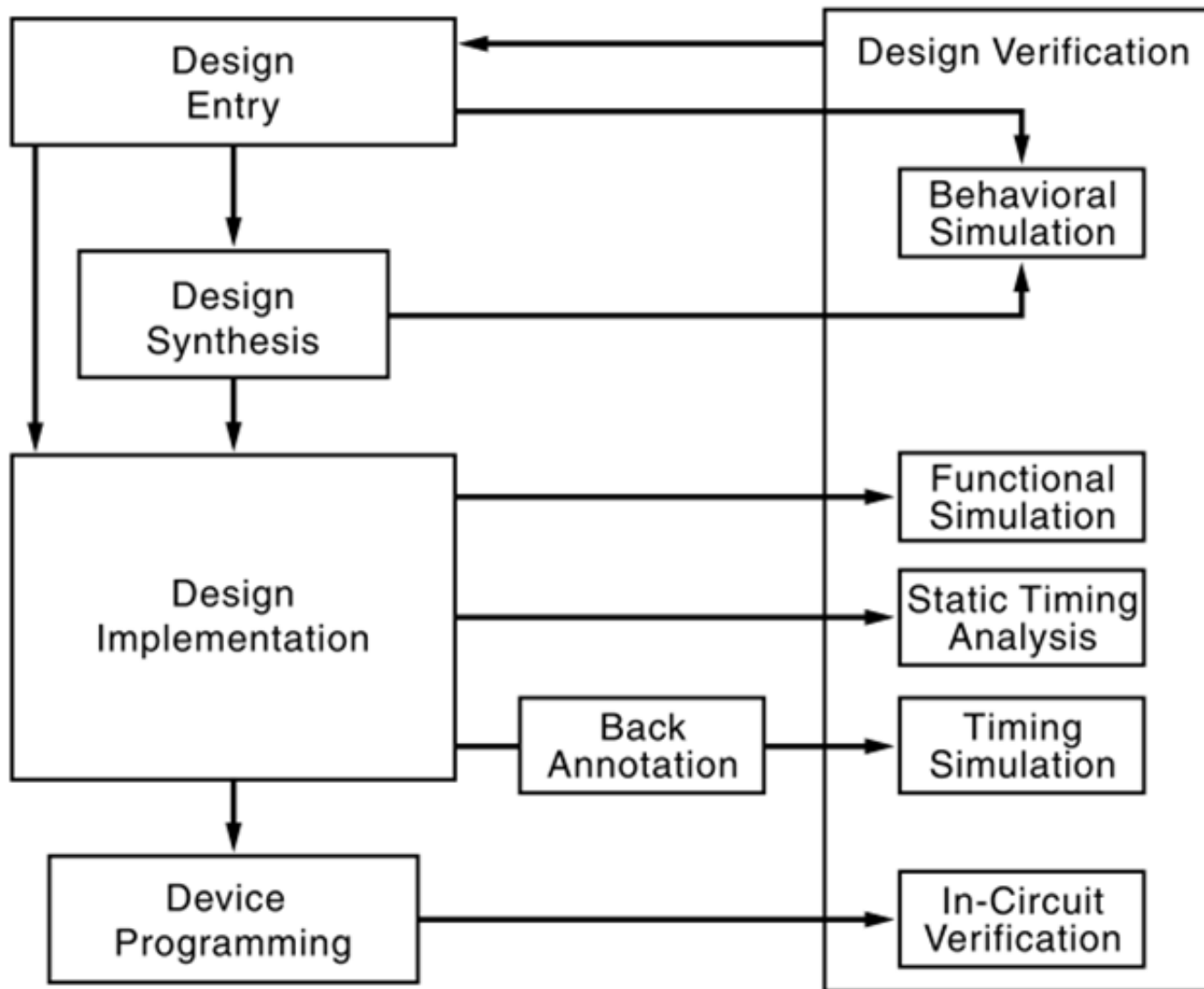
Resulta numa netlist (ou seja, componentes de hardware e seus interconexões) que implementar o modelado comportamento e estrutura

## Resultados

- Lista de rede
- Desempenho do circuito e uso de recursos estimativas



# Implementação (Mapa, Local e Rota)



Mapeia a netlist em primitivas FPGA

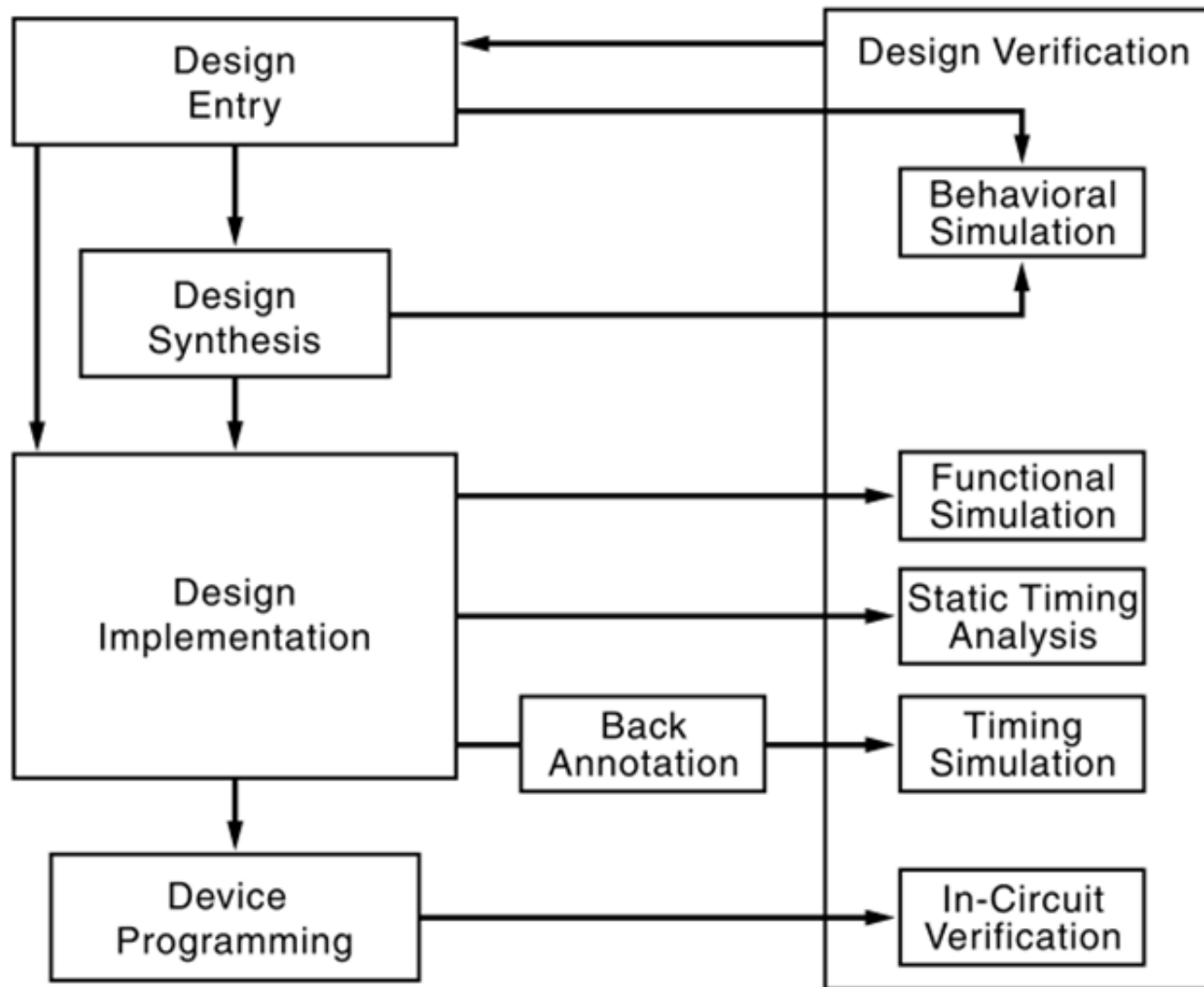
Coloca as primitivas em locais específicos do FPGA

Roteia o interconexões entre primitivos

Resultados

- Arquivo de configuração FPGA
- Relatórios com uso de recursos FPGA, atrasos de circuito, energia consumo, ...

# Programação de dispositivo (FPGA)



## Transfere o

arquivo de configuração para o FPGA

- Feito com uma ferramenta de software e um cabo de programação
- A maioria dos FPGAs são baseados em SRAM (volátil configuração)
- Alguns dispositivos são baseados em memória FLASH não volátil

# Sistema em chip (SoC)

---

A **SoC** é um único circuito integrado que integra pelo menos componentes como um processador e periféricos de E/S para executar tarefas gerais de interface do usuário e tarefas domésticas e aceleradores de hardware especiais para lidar com operações de computação intensiva.

A **SoC** integra um processador, módulos de memória, periféricos de E/S e aceleradores de hardware personalizados em um único chip.

Construiremos vários SoCs contendo um processador soft-core Xilinx MicroBlaze, subsistema de E/S mapeado em memória que pode incorporar núcleos de E/S personalizados, ....

# Co-design de hardware-software

---

A tecnologia FPGA nos permite personalizar o processador, selecionar apenas os periféricos de E/S necessários, criar uma interface de E/S personalizada e desenvolver aceleradores de hardware especializados para tarefas de computação intensiva.

Tanto o hardware quanto o software podem ser personalizados para atender a necessidades específicas.

A metodologia de explorar os compromissos entre hardware e software e desenvolvê-los e integrá-los simultaneamente é referida como **co-design de hardware-software**.

# Fluxo de Desenvolvimento do SoC

---

- Divida as tarefas em rotinas de software e aceleradores de hardware
- Projete núcleos personalizados do usuário, se necessário
- **Desenvolva o hardware**
- Desenvolva o software
- Implemente o hardware e software e execute testes.

# Considerações finais

---

Ao final desta palestra você deverá ser capaz de:

- Descrever as principais características da computação reconfigurável
- Descreva uma arquitetura FPGA típica
- Saiba a diferença entre FPGA e PSoC
- Identifique as principais etapas do fluxo de design do FPGA
- Definir co-design de software-hardware

Pendência:

- Instale o Xilinx Vitis 2022.2