

Dispositivos Lógicos Programáveis

PLD22109

Prof. Renan Augusto Starke

Instituto Federal de Santa Catarina – IFSC
Campus Florianópolis
renan.starke@ifsc.edu.br

15 de agosto de 2022



INSTITUTO FEDERAL
SANTA CATARINA

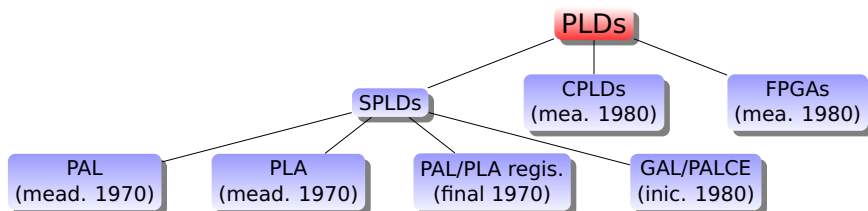
Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
INSTITUTO FEDERAL DE SANTA CATARINA

Programable Logic Devices

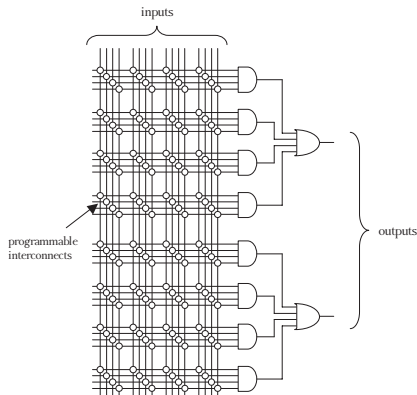
São dispositivos eletrônicos utilizados para implementar funções lógicas. Um PLD é um dispositivo de uso geral cujo hardware pode ser configurado para atender às especificações particulares.

- ▶ *Programmable Logic Devices* – Dispositivos Lógicos Programáveis
- ▶ Um PLD é fabricado sem possuir uma função fixa. Sua função é implementável através de uma Linguagem de Descrição de Hardware (HDL – *Hardware Description Language*)

Programmable Logic Devices



Programmable Array Logic – PAL

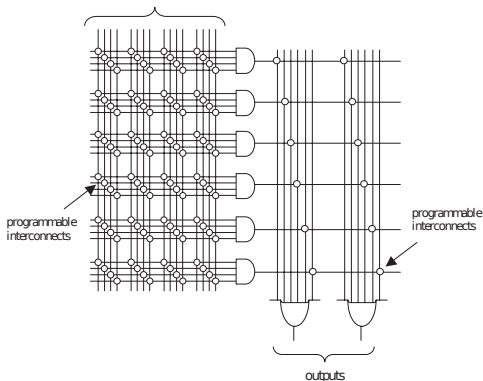


- ▶ Arranjo programável de portas AND seguidos por arranjo fixo de portas OR.
- ▶ Qualquer circuito pode ser representado por uma soma de produtos.
- ▶ **Apenas circuitos combinacionais.**
- ▶ PROM ou EPROM.

$$X = m_1 + m_2 + \cdots m_M$$

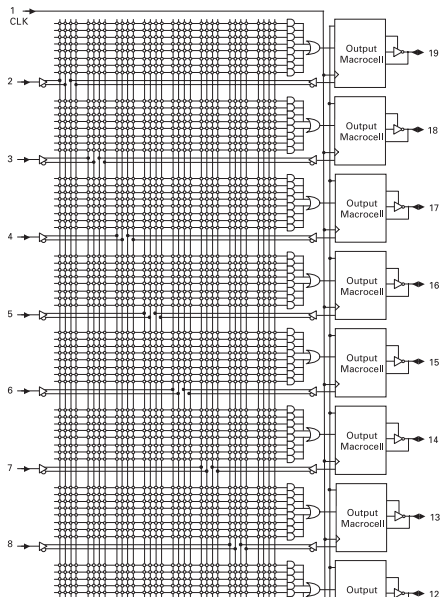
$$X = a_1 \bar{a}_2 + a_2 a_3 \bar{a}_4 \cdots$$

Programmable Logic Array – PLA



- ▶ Arranjo programável de portas AND seguidos por arranjo programável de portas OR.
- ▶ Qualquer circuito pode ser representado por uma soma de produtos.
- ▶ **Apenas circuitos combinacionais.**
- ▶ Maior flexibilidade, com penalização da velocidade do circuito.
- ▶ PROM ou EPROM.

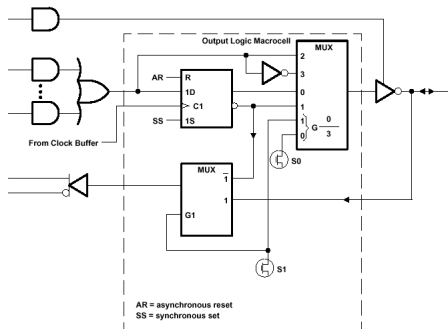
Generic Array Logic – GAL



- ▶ Célula de saída mais sofisticada:
 - flip-flops.
 - multiplexadores.
 - **Programável.**
- ▶ Lógica sequencial e combinacional.
- ▶ EEPROM ou FLash.
- ▶ É ainda encontrado no mercado: 250MHz, 3.3V, CMOS.

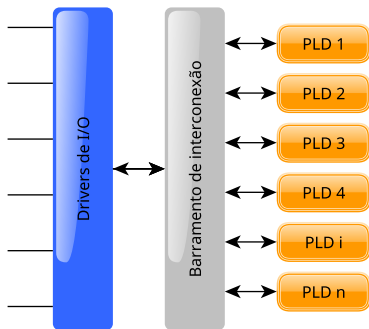
Generic Array Logic – GAL

output logic macrocell diagram



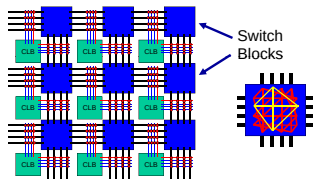
- ▶ Programável como saída registrada ou combinacional

Complex Programmable Logic Devices – CPLD



- ▶ Vários PLDs em um mesmo encapsulamento (GALs).
- ▶ Comunicam-se através de um arranjo de interconexões.
- ▶ *Drivers* de IO modernos.
- ▶ Suporte a JTAG.
- ▶ Inúmeros pinos de IO.
- ▶ Basicamente não volátil: **EEPROM** ou **Flash**.

Field Programmable Gate Arrays – FPGAs



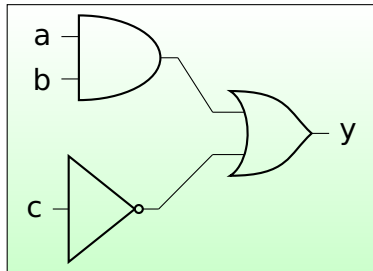
- ▶ Matriz de CLBs (Configurable Logic Block) interconectados por um arranjo de matrizes de interconexão.
- ▶ CLBs não implementam soma de produtos.
 - Funções lógicas são implementas por LUTs.
- ▶ Suporte a JTAG.
- ▶ Inúmeros pinos de IO.
- ▶ Basicamente volátil: **SRAM**.

LUT – LookUp Table

Todas as funções lógicas de FPGAs e CPLDs modernas são implementadas por LUTs. Em termos gerais, é uma tabela verdade que determina saídas em funções de um conjunto de entradas.

- ▶ Gerador universal de funções.
- ▶ Módulo configurável capaz de implementar qualquer tabela verdade de n entradas.
- ▶ Altamente flexível.
- ▶ Método mais utilizado em PLDs modernas.

Funções lógicas

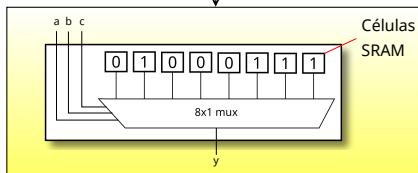


Função requisitada

a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Tabela Verdade

LUT Programada



Estrutura interna

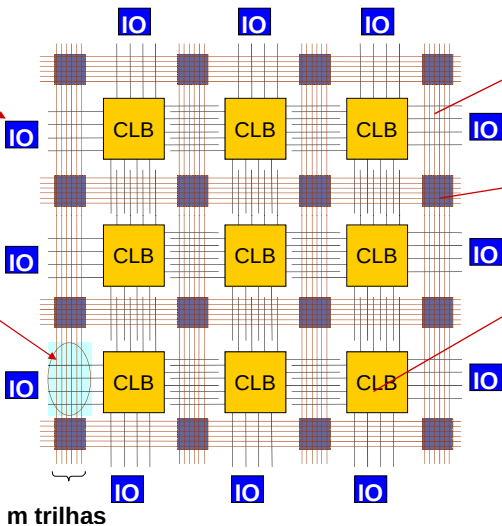
Blocos de E/S configuráveis

Chaves programáveis

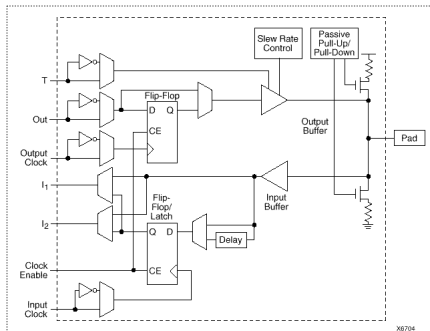
Canal de roteamento:
conexões configuráveis

Caixa de conexão

Funções configuráveis
Configurable Logic Block

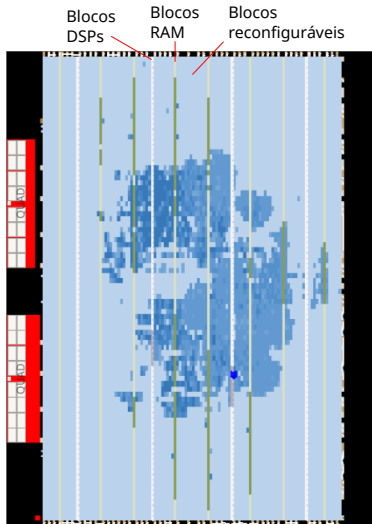


Blocos de entrada/saída



- ▶ Possuem recursos de memória configuráveis (registrador ou *latch*).
- ▶ Pads configuráveis como entrada, saída ou bidirecionais.
- ▶ Entradas podem utilizar registrador o *latch*.
- ▶ Pode-se definir sinais *three-state*.

Blocos adicionais

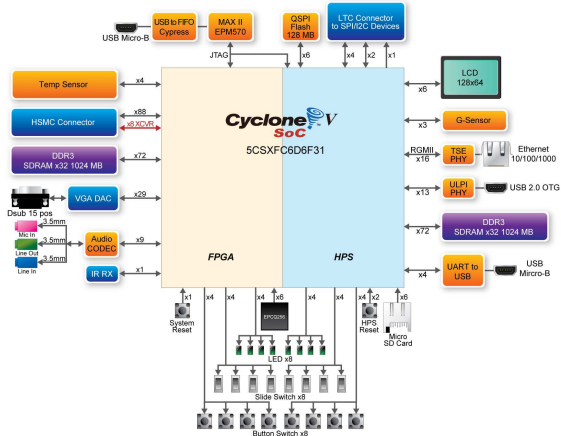


- ▶ Blocos de SRAM “customizáveis”:
 - 9.216 bits.
 - 147.456 bits.
 - 640 bits.
- ▶ Blocos podem ser combinados formando memórias maiores ou menores.
- ▶ Portas simples ou duplas.
- ▶ Operação em até 600 MHz.
- ▶ Distribuídos pelo dispositivos para reduzir latência.
- ▶ Ferramenta capaz de inferir blocos de RAM ao invés de usar LUTs.

- ▶ Blocos de DSPs “customizáveis”:
 - multiplicadores 18×18 .
 - customizáveis para 9×9 , 12×12 ou 36×36 .
- ▶ Blocos podem ser combinados formando multiplicadores maiores.
- ▶ Ferramenta capaz de inferir blocos multiplicadores e unidades MAC.
- ▶ Operação em até 550 MHz.
- ▶ Distribuídos pelo dispositivos para reduzir latência.

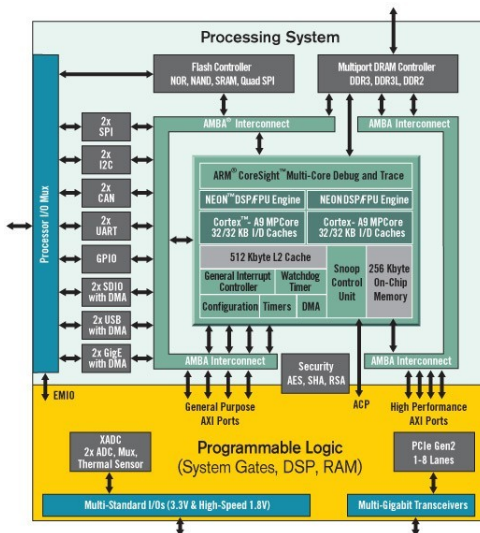
Blocos adicionais: Hard. CPUs

Altera Cyclone V: FPGA + ARM



Blocos adicionais: Hard. CPUs

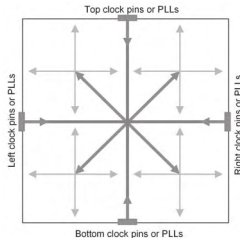
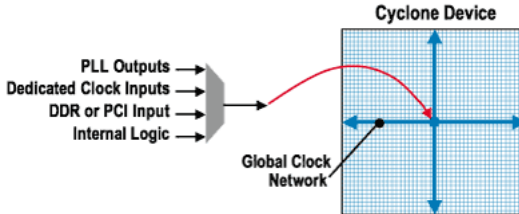
Xilinx Zynq-7000: FPGA + ARM



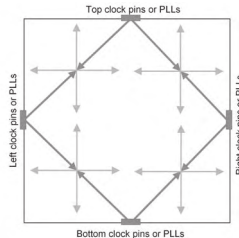
- ▶ Redes de clock organizadas de forma hierárquica:
 - Rede Global (GCLK).
 - Rede Regional .
 - Rede Periférica.

- ▶ PLLs:
 - Ramificação.
 - Geração de novas frequências.

Gerenciamento de *clock*: redes

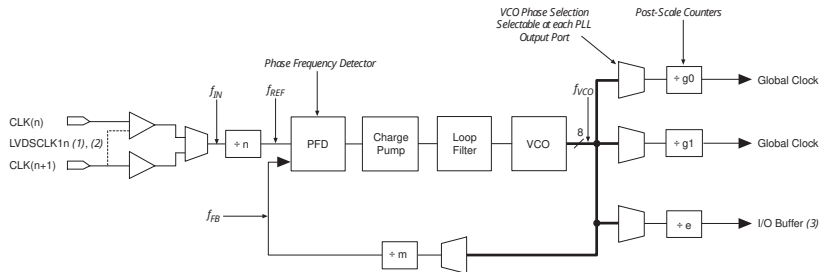


Redes globais



Redes regionais

Gerenciamento de *clock*: PLLs



- ▶ Originalmente, plataforma de prototipação para ASICs.
- ▶ Flexibilidade de software, mas confiabilidade e paralelismo encontradas em hardware.
- ▶ Favorece criação de circuitos digitais customizados com os benefícios de desempenho do hardware e as vantagens competitivas de “time-to-market” existente em projetos de software.
- ▶ Algumas aplicações típicas:
 - Algoritmos de criptografia.
 - IA / redes neurais.
 - Digital Signal Processing (DSP) – operações MAC.
 - Processamento de imagens.
 - Algoritmos de codificação e decodificação (comunicação, compactação, ...)
 - Hard real time : comutação (tempo morto de chaves), sincronização (ADCs), proteção.

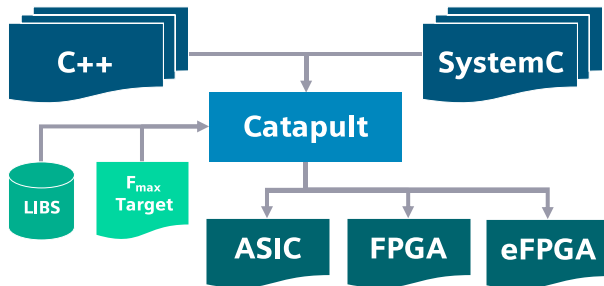
- ▶ Aplicações adequadas para FPGAs
 - Alta velocidade de processamento, throughput e baixa latência.
 - Consumo de energia (em relação a microprocessadores).
 - Paralelização.
 - Operações a nível de bit.
 - Operações aritméticas não ortodoxas (ex. palavras com tamanhos exóticos – 13 bits, ...)
 - Necessidade de reconfiguração.
 - Volume de produção de pequeno para médio.

Vantagens

- ▶ Facilidade para *updates*.
- ▶ Eficiência arquitetural.
- ▶ Eficiência no uso de recursos – apenas os algoritmos necessários são configurados no hardware em um determinado momento.
- ▶ Maior velocidade com menor consumo de energia quando comparado com software.
- ▶ Flexibilidade quando comparado com ASICs.
- ▶ Custo reduzido para produções de tamanho pequeno para médio.
- ▶ Facilidade para inferir paralelismo.

- ▶ Desenvolvimento:
 - Mão de obra qualificada.
 - Licenças.
 - Custo do hardware.
- ▶ Atualmente usa-se HLS (High-level synthesis) para reduzir tempo e custo de desenvolvimento.
 - Intel High Level Synthesis Compiler (Intel/Altera).
 - VivadoHLS (Xilinx).
 - HDL Coder (Mathworks)
 - Stratus HLS (Cadence)
 - Catapult (Mentor/Siemens)
 - gcc2verilog (acadêmico)
 - LegUp (acadêmico)

High-level synthesis



- ▶ *Datasheets e application notes Xilinx e Altera*
- ▶ PEDRONI, Volnei A. **Eletrônica digital moderna e VHDL**. Rio de Janeiro: Elsevier, 2010.
- ▶ PEDRONI, Volnei A. **Circuit Design with VHDL**. MIT Press. 2004.