



Digital Integrated Circuits

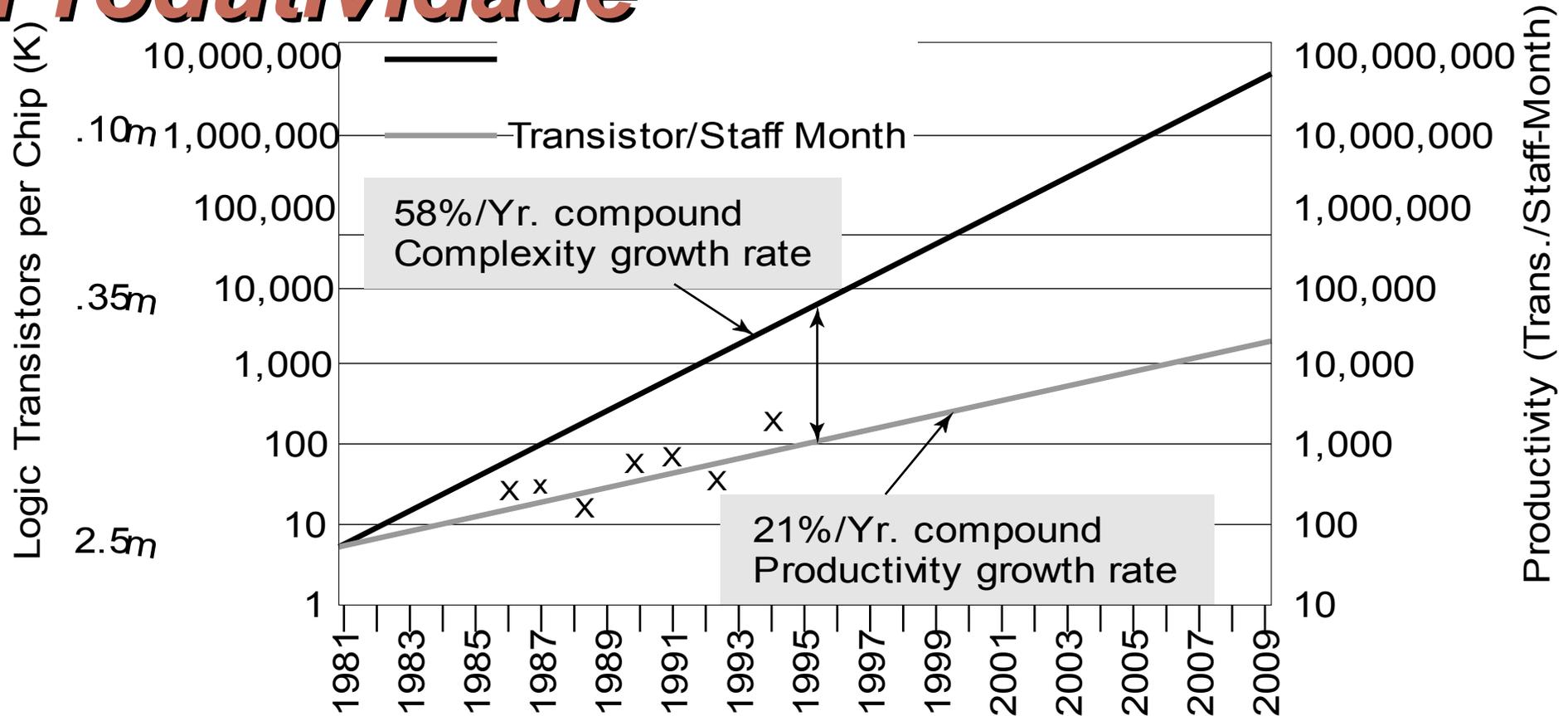
A Design Perspective

Jan M. Rabaey
Anantha Chandrakasan
Borivoje Nikolic

Metodologias De Projeto

December 10, 2002

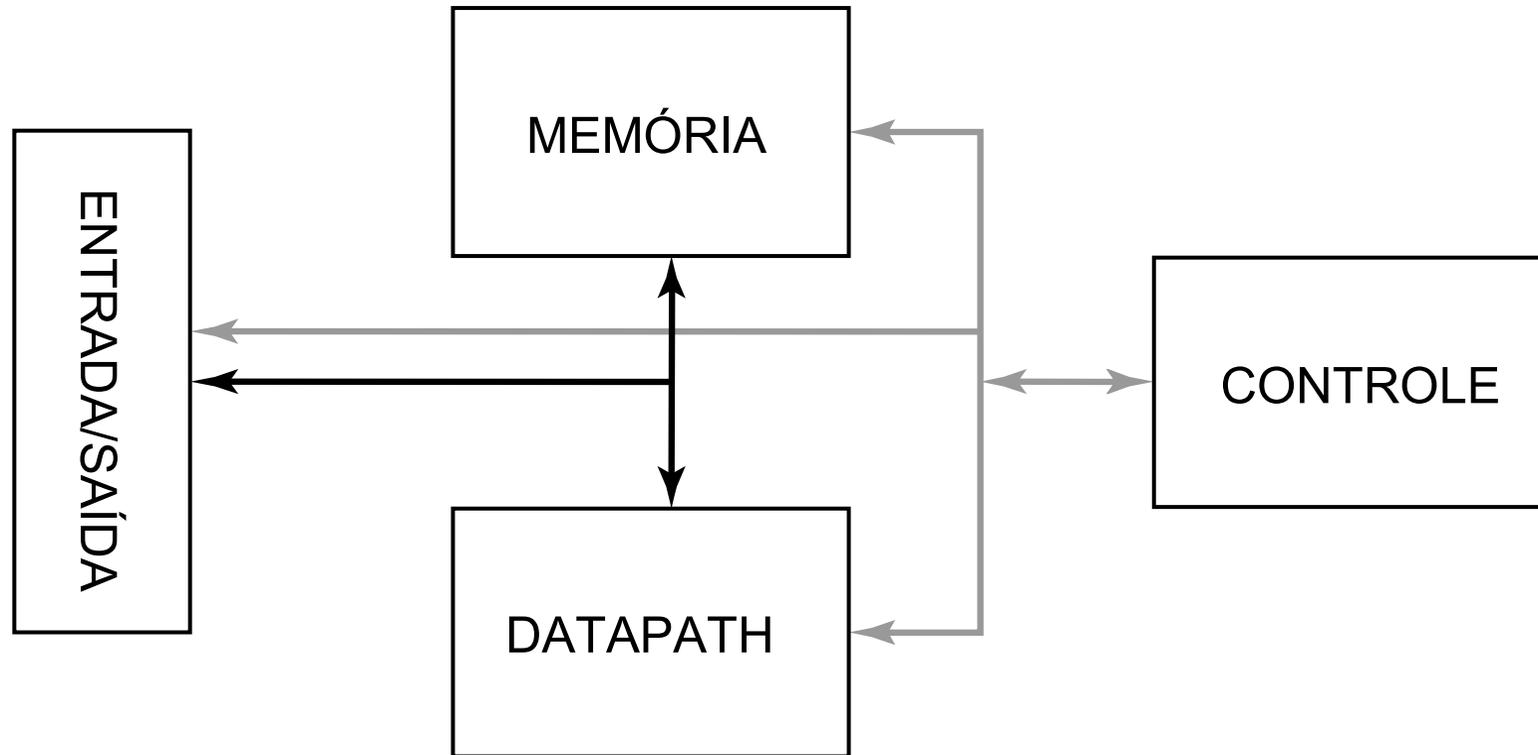
Projeto- Desafio da Produtividade



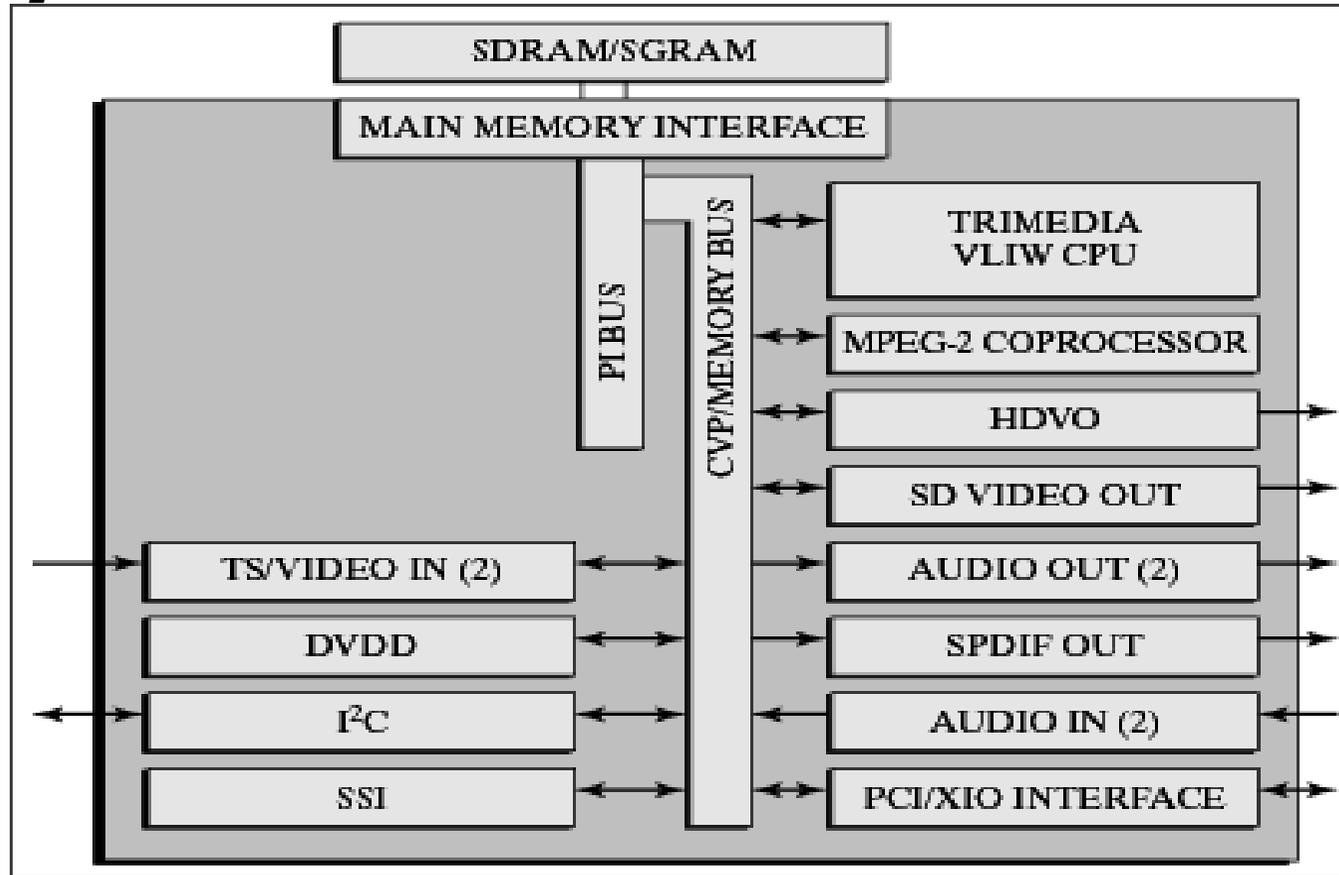
Uma crescente lacuna entre a complexidade e produtividade de projeto

Fonte: sematech97

Um Processador Básico

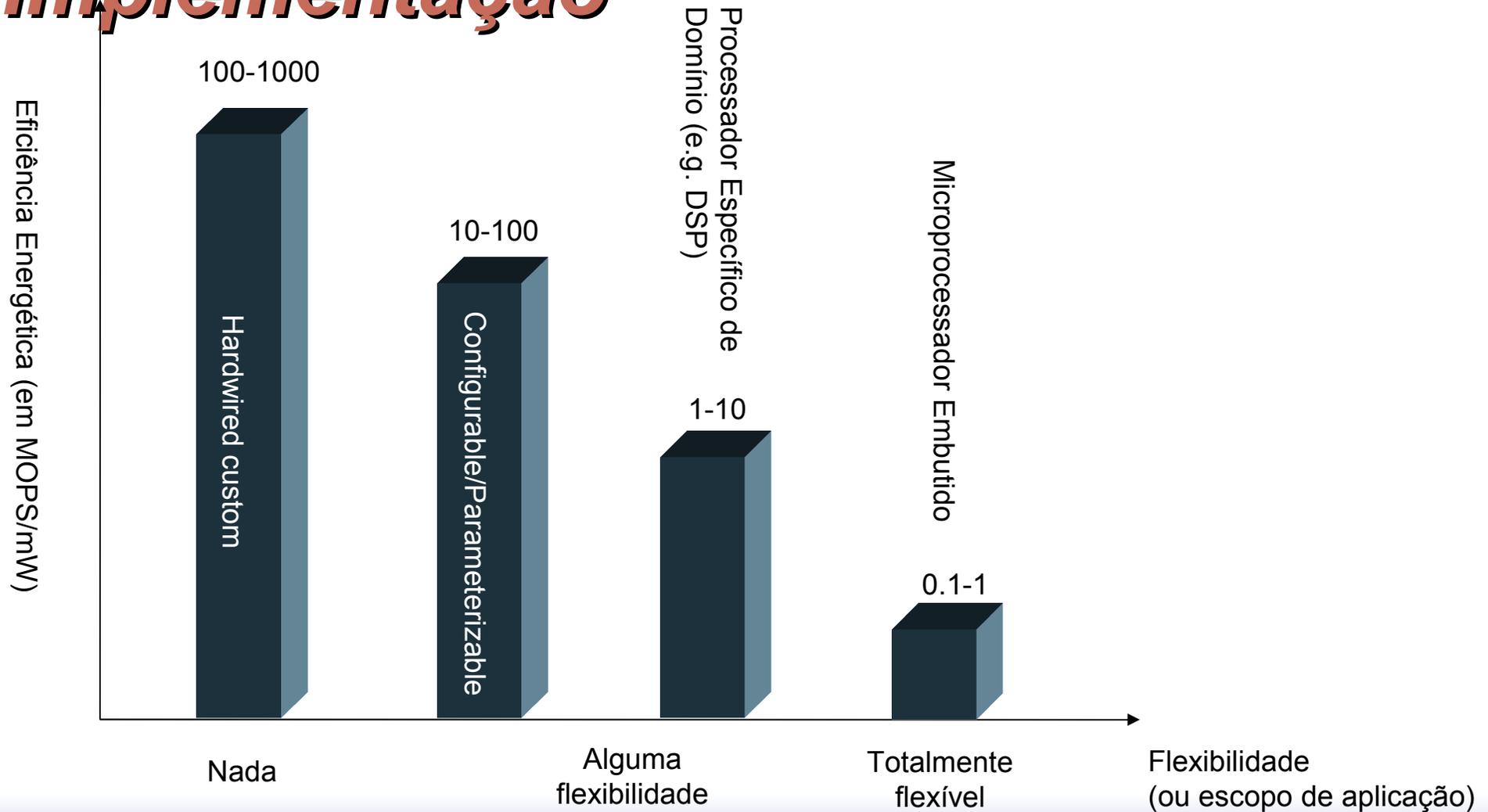


Um Sistema-sobre-Pastilha: Exemplo

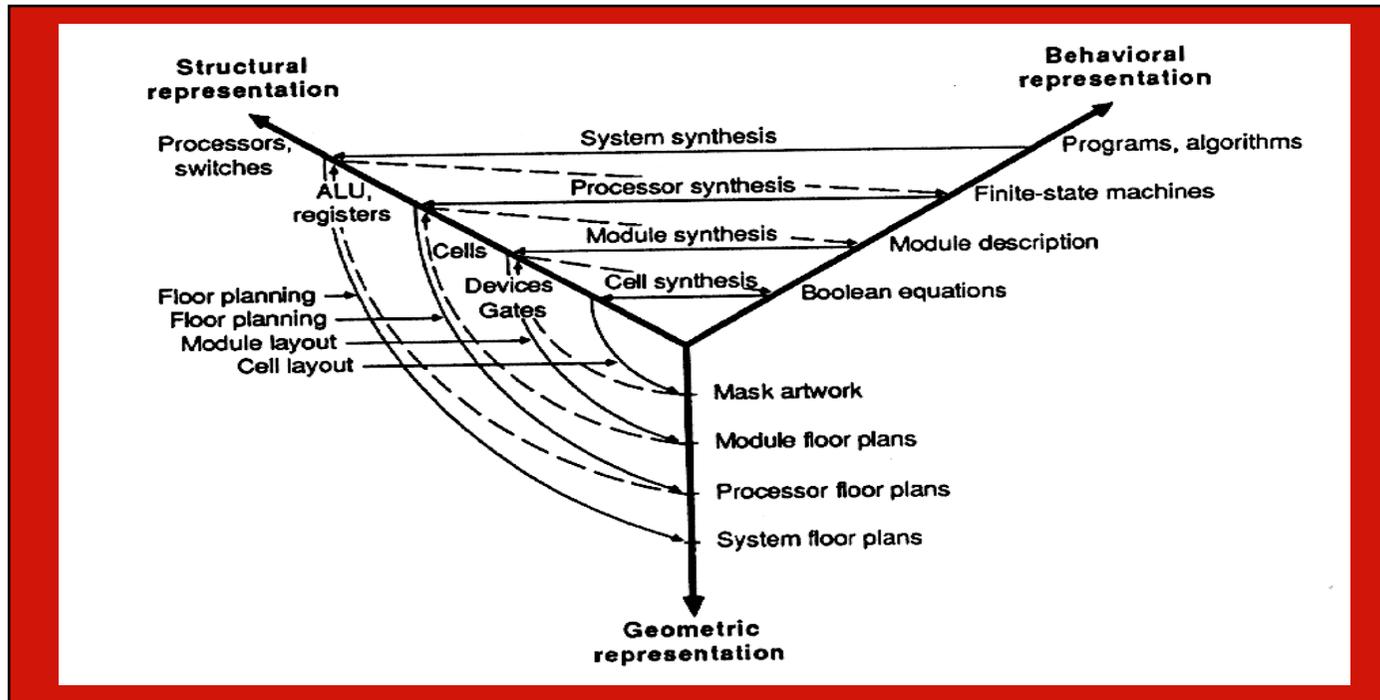


Courtesy: Philips

Impacto das Opções de Implementação

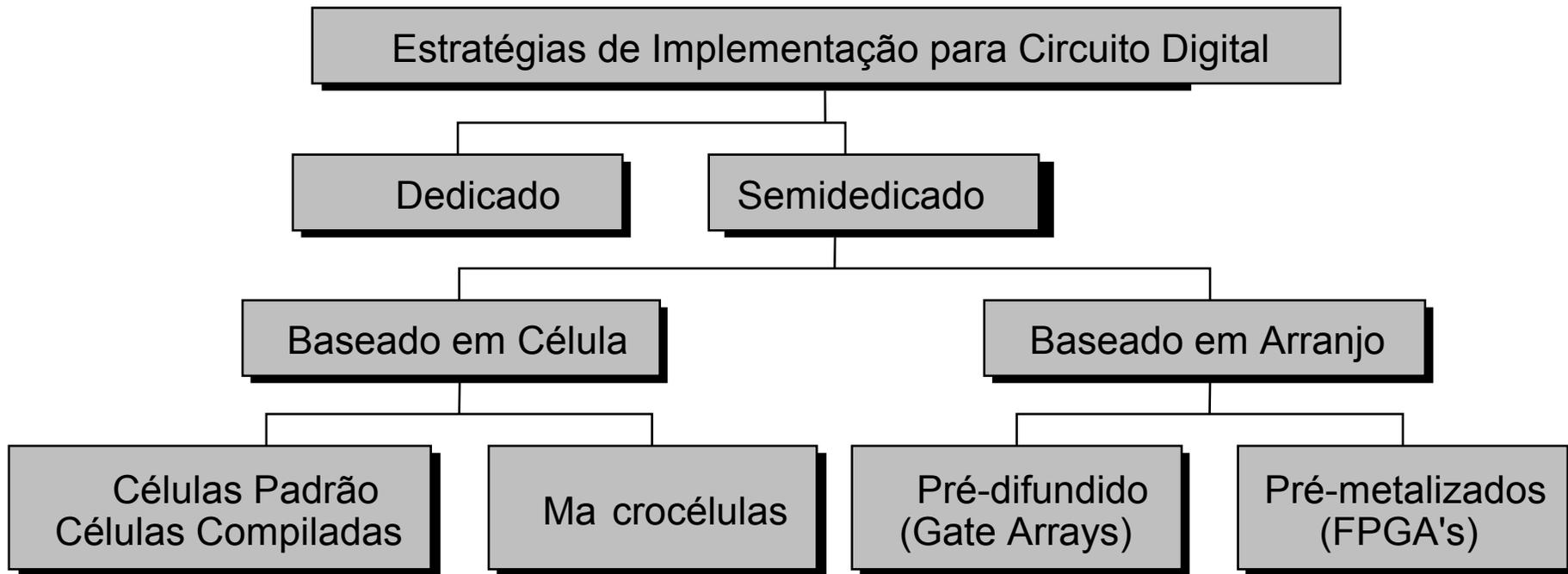


Metodologia de Projeto

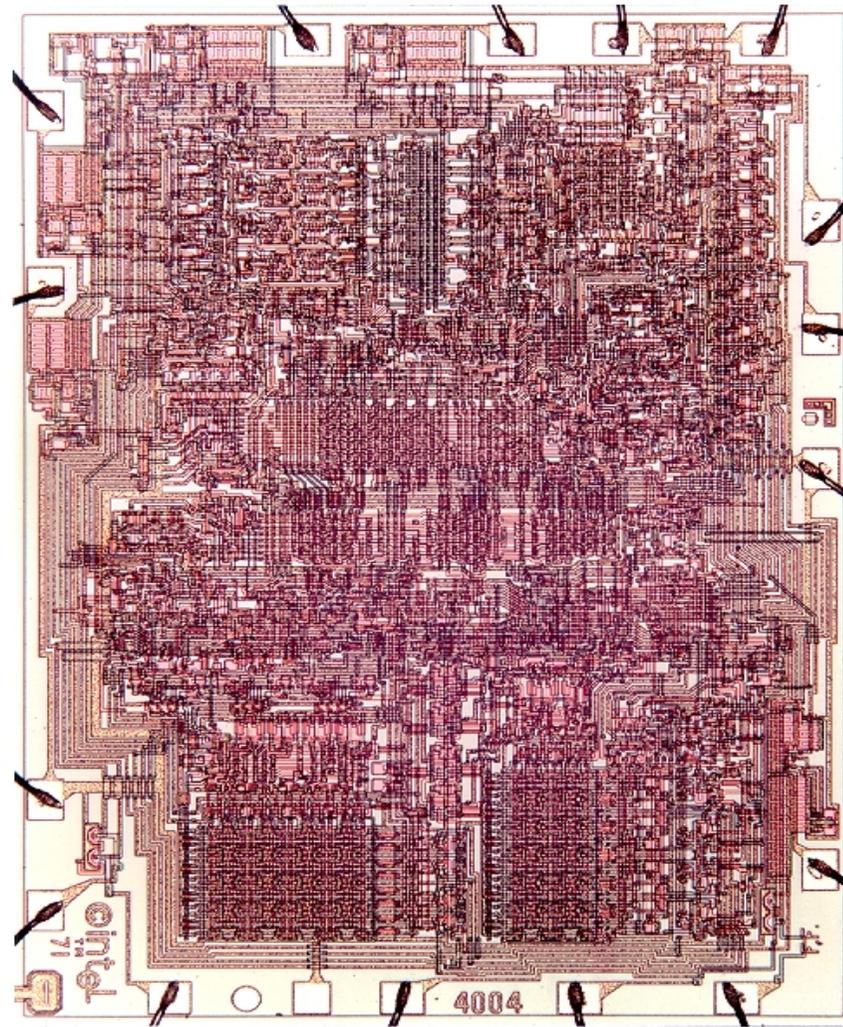


- O processo de projeto flui iterativamente entre três abstrações: comportamento, estrutura e geometria
- Automação crescente para cada um desses passos

Possíveis Implementações

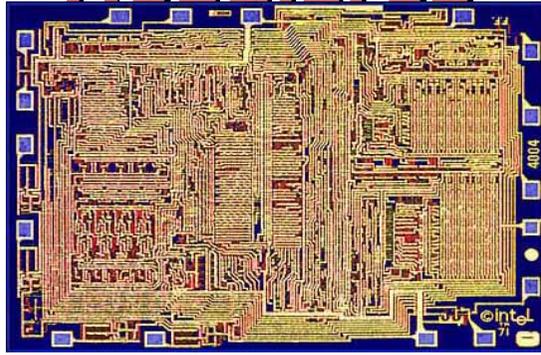


A Abordagem Dedicada

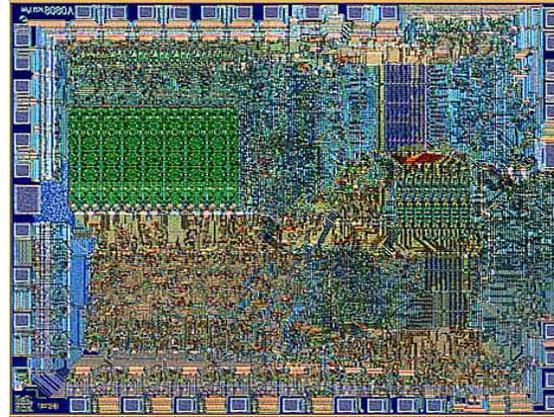


Intel 4004

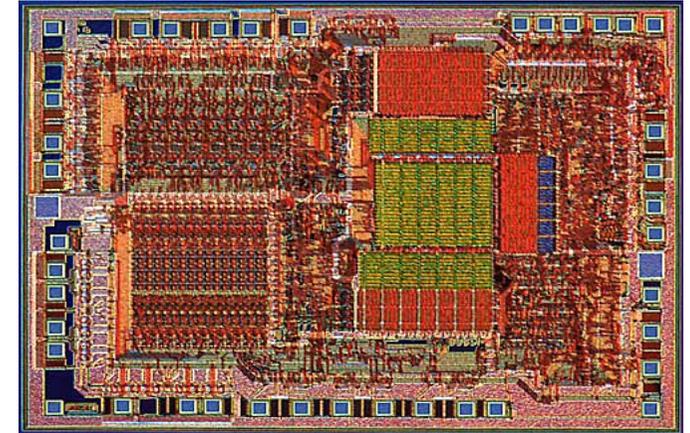
Transição para Automação e Estruturas Regulares



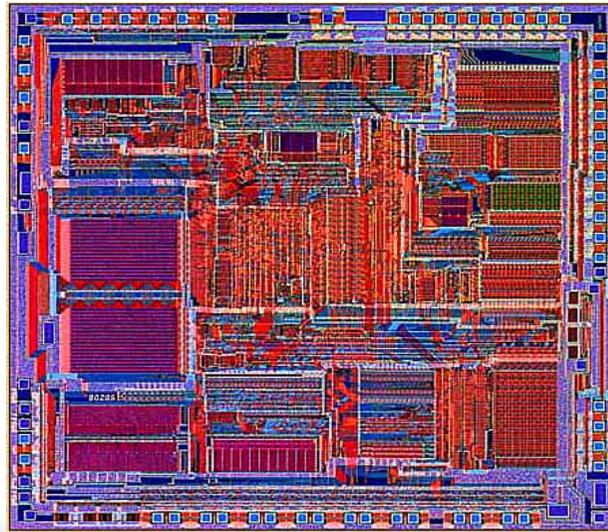
Intel 4004 ('71)



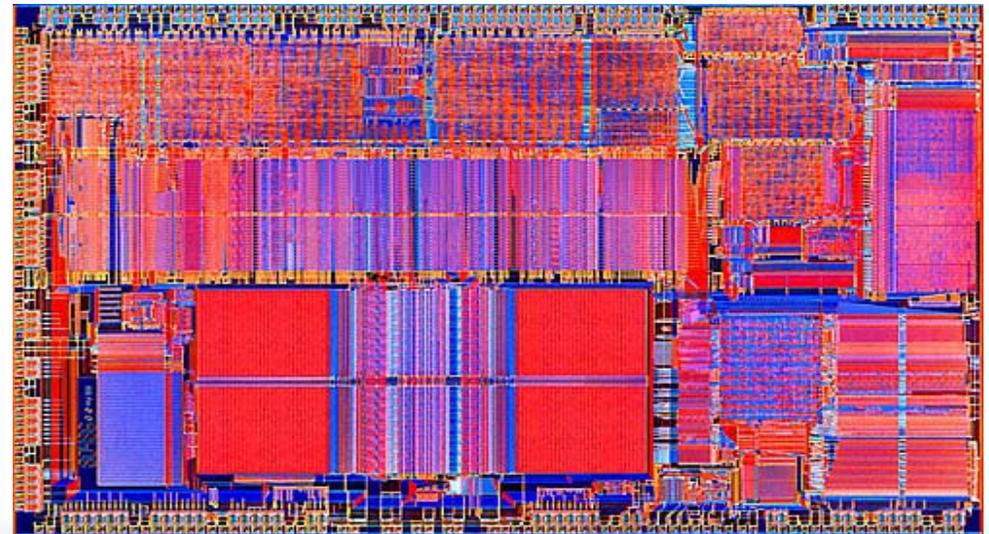
Intel 8080



Intel 8085

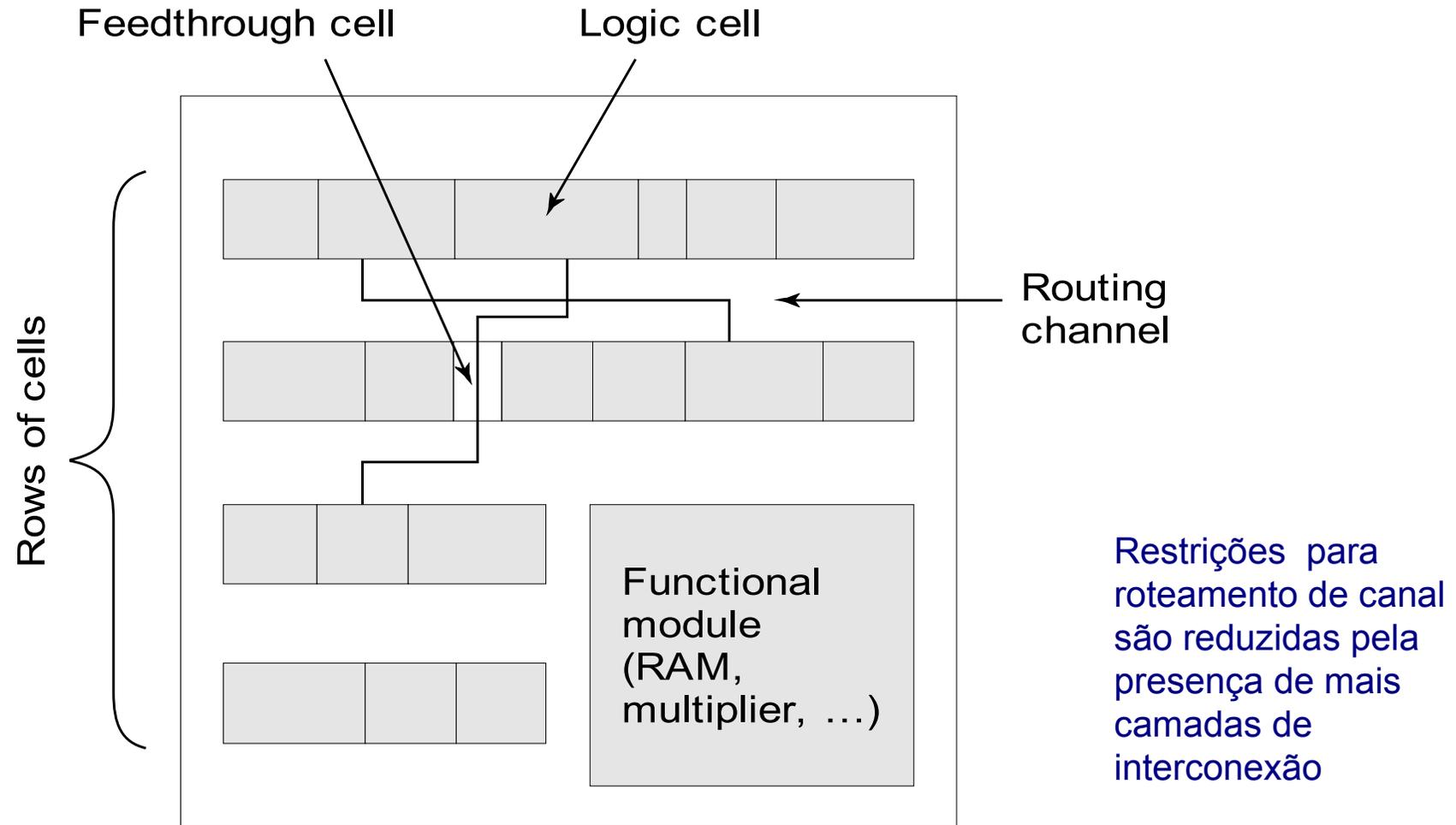


Intel 8286

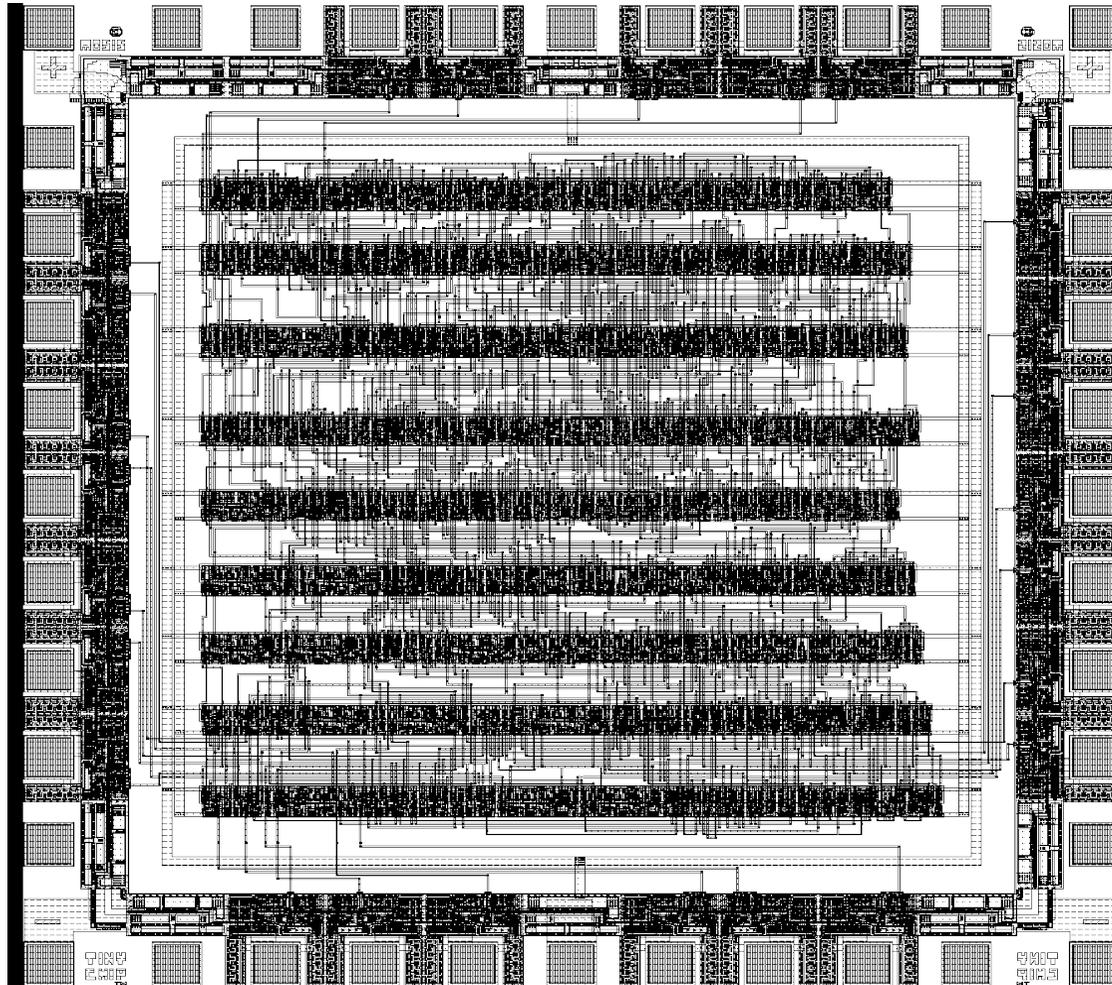


Intel 8486

Projeto Baseado em Células (padrão)

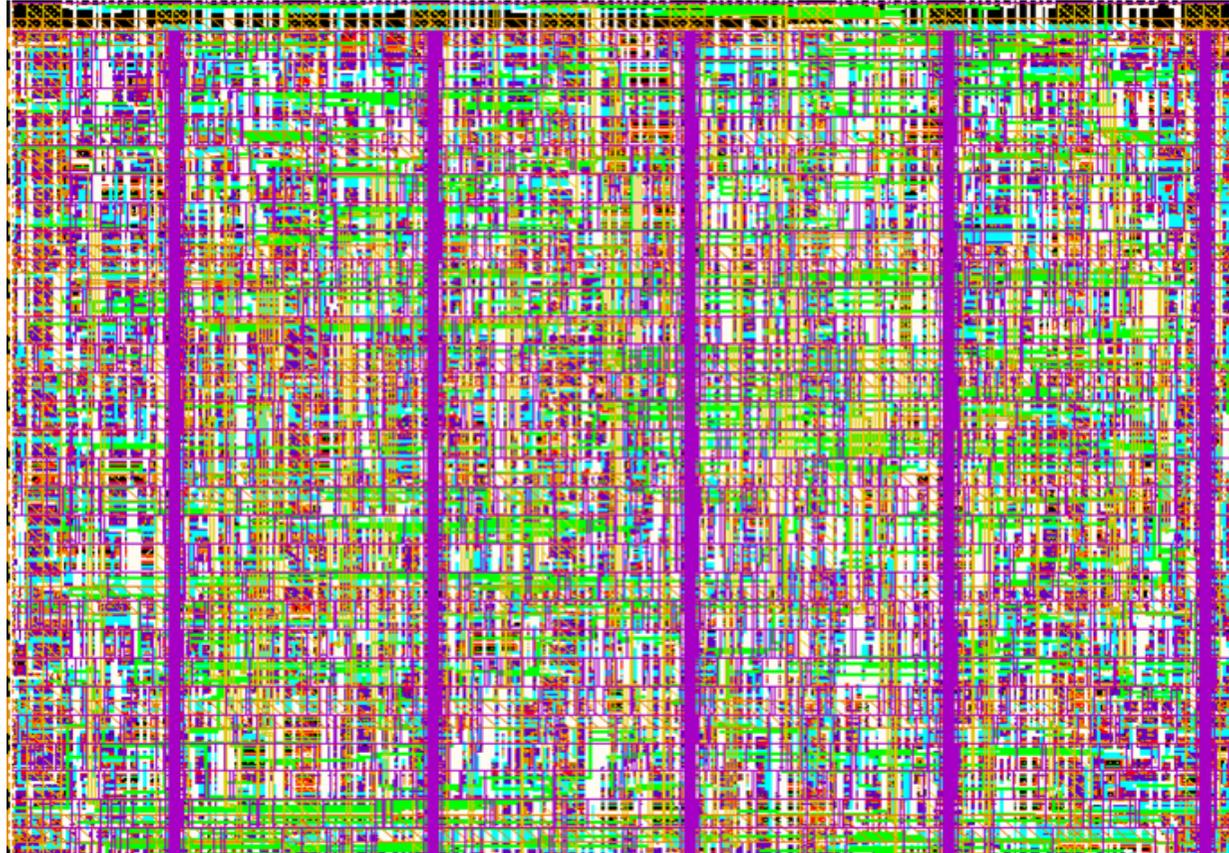


Célula Padrão — Exemplo



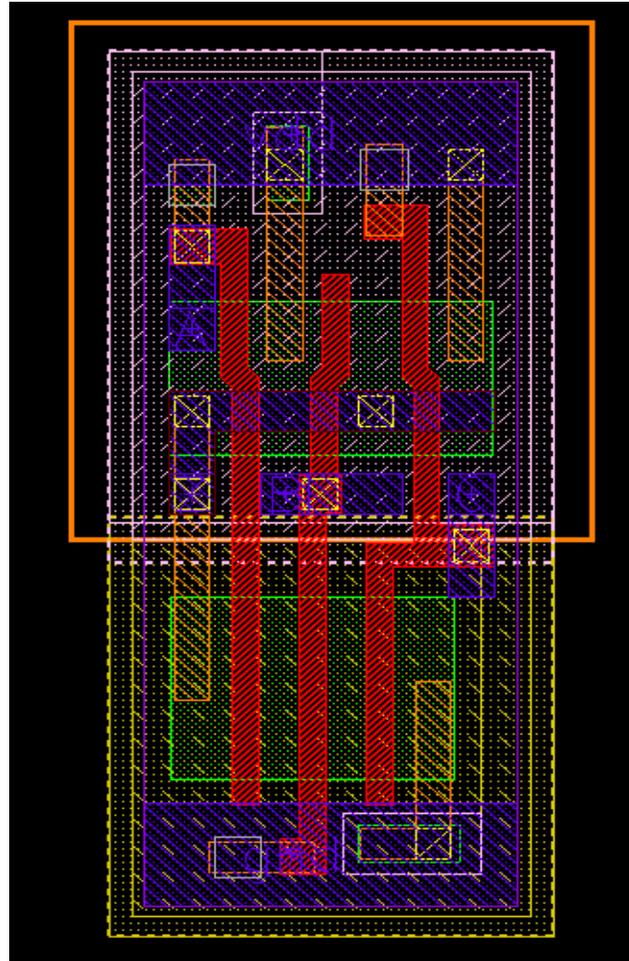
[Brodersen92]

Célula Padrão– A Nova Geração



*Estrutura da célula
escondida entre
camadas de interconexão*

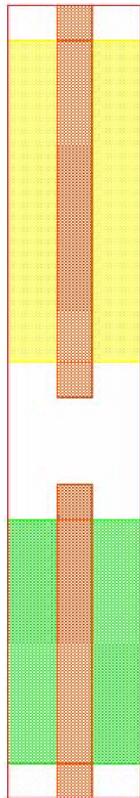
Célula Padrão - Exemplo



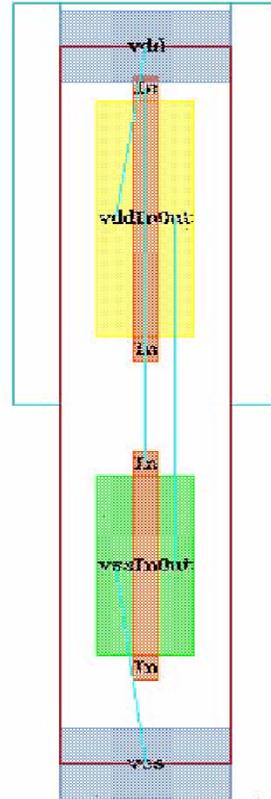
Path	1.2V - 125°C	1.6V - 40°C
$In1-t_{pLH}$	$0.073+7.98C+0.317T$	$0.020+2.73C+0.253T$
$In1-t_{pHL}$	$0.069+8.43C+0.364T$	$0.018+2.14C+0.292T$
$In2-t_{pLH}$	$0.101+7.97C+0.318T$	$0.026+2.38C+0.255T$
$In2-t_{pHL}$	$0.097+8.42C+0.325T$	$0.023+2.14C+0.269T$
$In3-t_{pLH}$	$0.120+8.00C+0.318T$	$0.031+2.37C+0.258T$
$In3-t_{pHL}$	$0.110+8.41C+0.280T$	$0.027+2.15C+0.223T$

Célula NAND de 3-entradas
(da ST Microelectronics):
C = Capacitância de Carga
T = tempo de subida/
descida de entrada

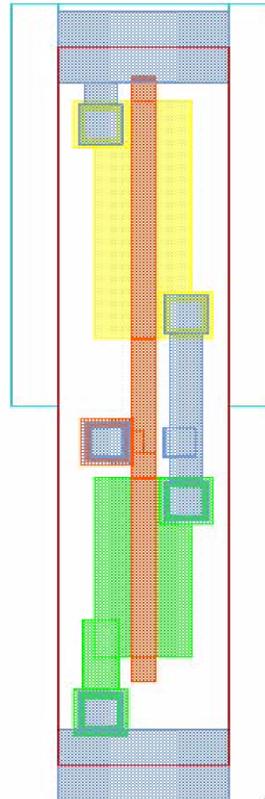
Geração Automática de Célula



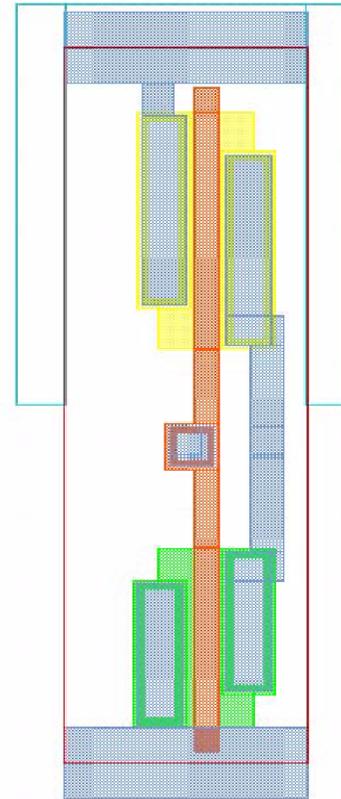
Geometrias iniciais do transistor



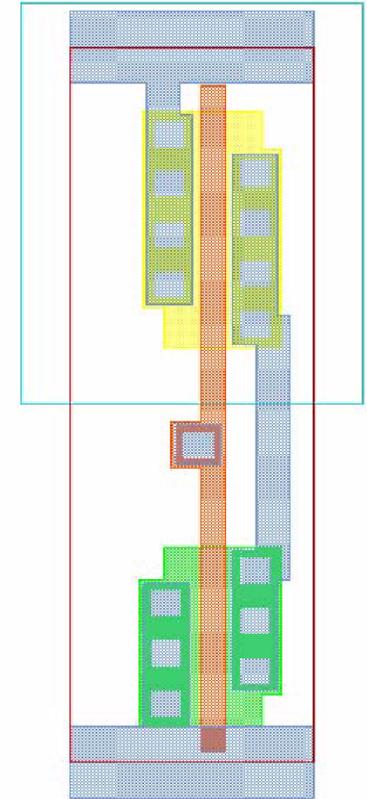
Transistores posicionados



Célula Roteada

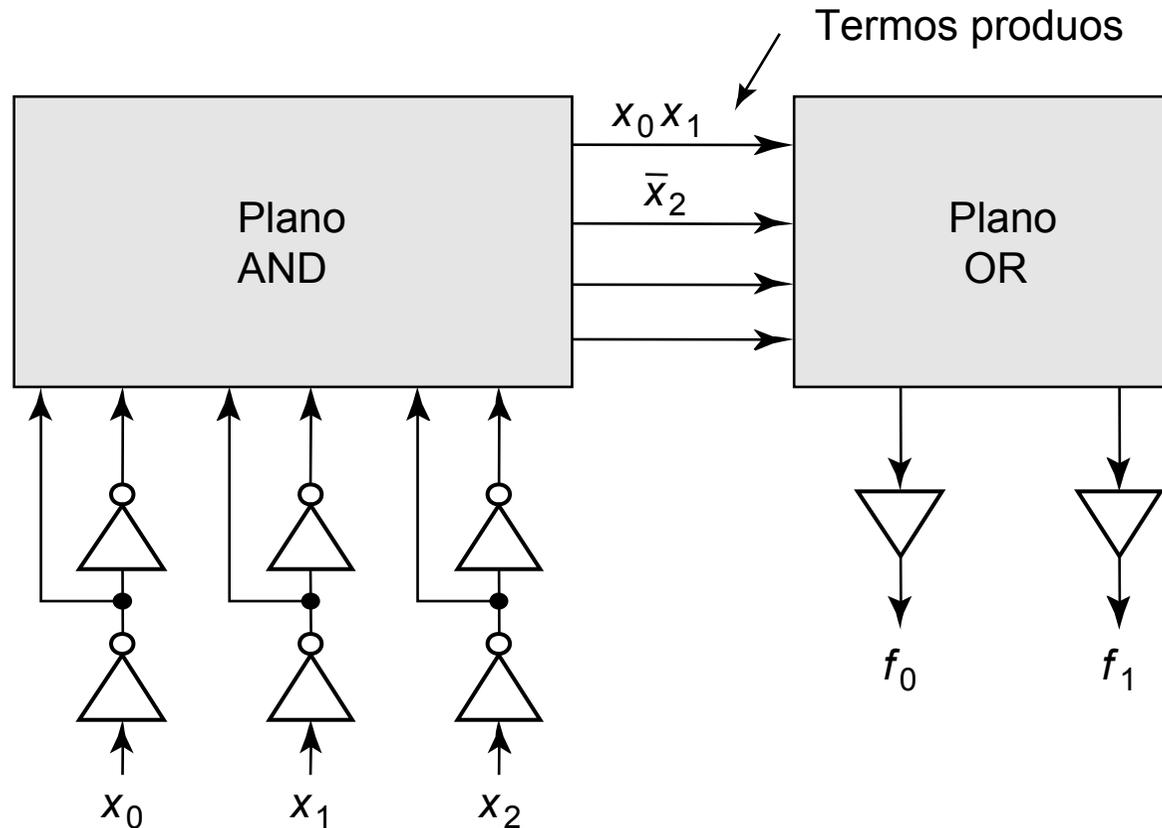


Célula Compactada



Célula Terminada

Uma Perspectiva Histórica: o PLA



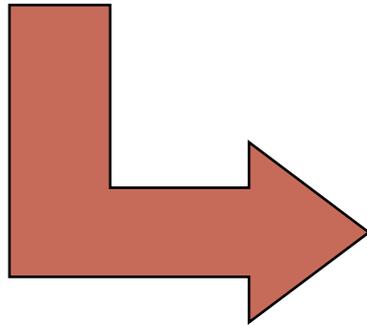
Lógica 2-Níveis

$$f_0 = x_0x_1 + \bar{x}_2$$

$$f_1 = x_0x_1x_2 + \bar{x}_2 + \bar{x}_0x_1$$

Toda função lógica pode ser expressa em formato soma-de-produtos (AND-OR)

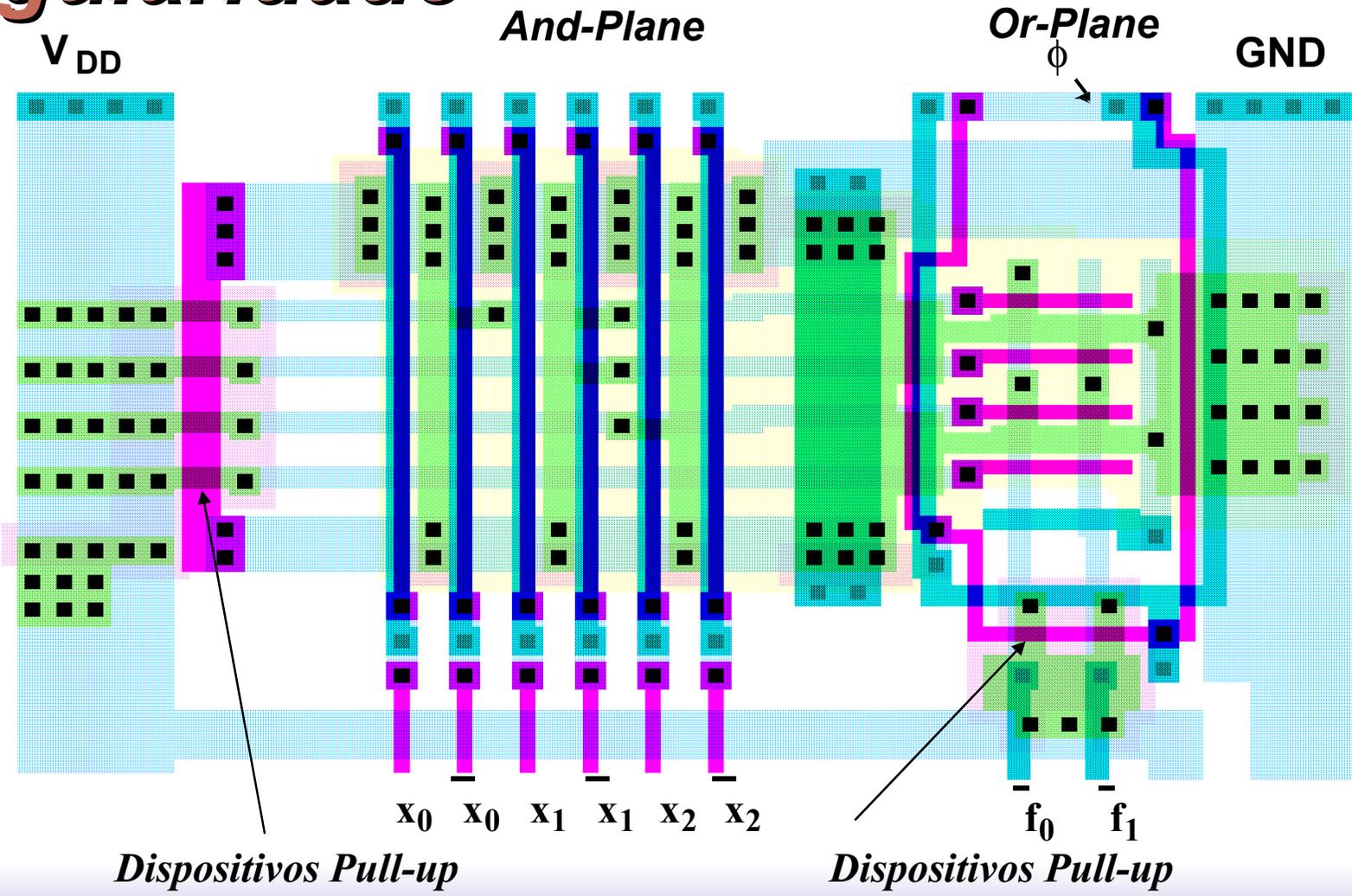
mintermo



Forma invertida (NOR-NOR): mais efetiva

$$\begin{aligned}\bar{f}_0 &= \overline{(x_0 + x_1) + x_2} \\ \bar{f}_1 &= \overline{(x_0 + x_1 + x_2) + x_2 + (x_0 + x_1)}\end{aligned}$$

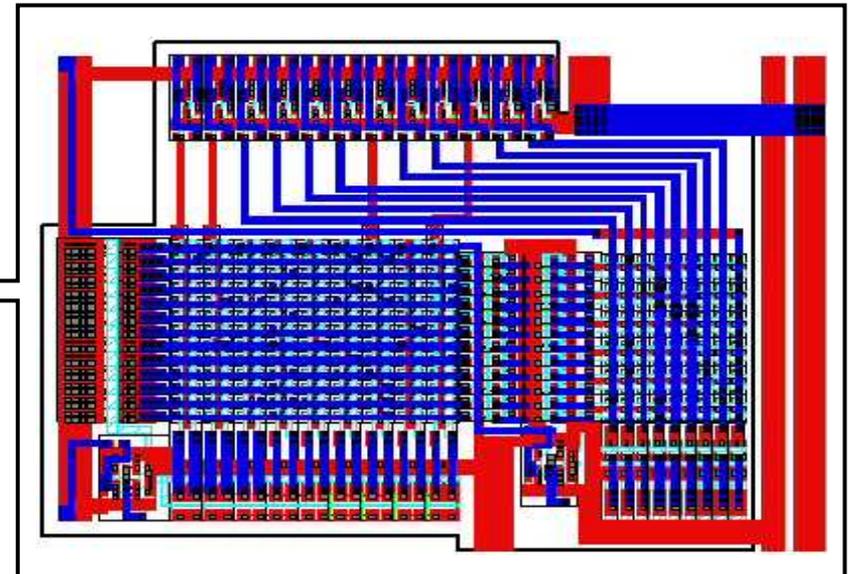
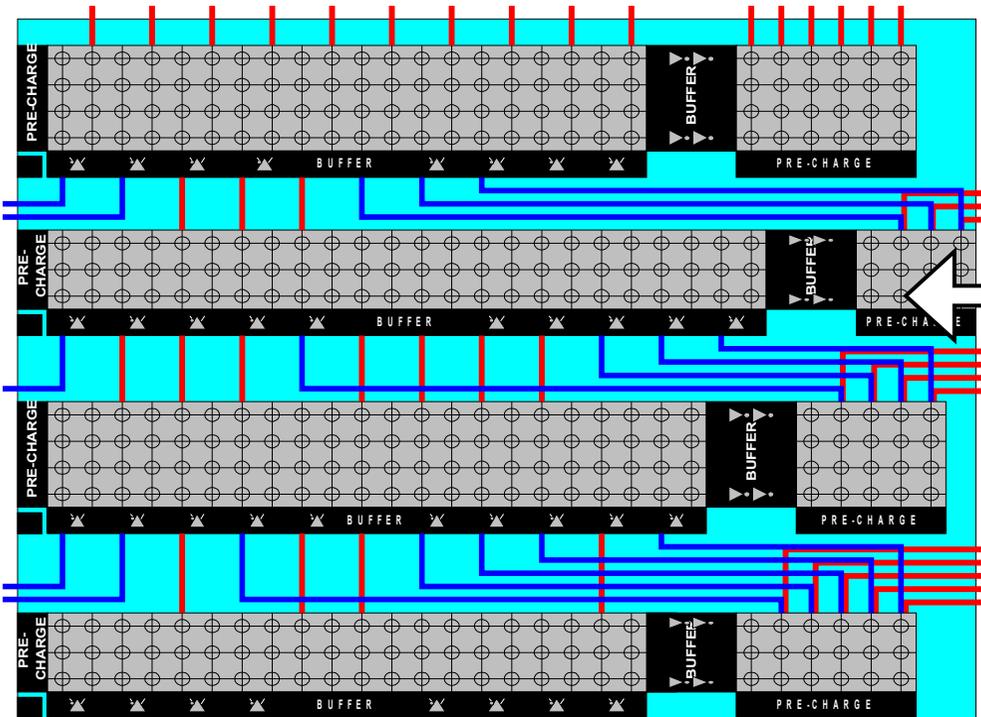
Layout de PLA– Usando a Regularidade



Sopro de (Nova) Vida em PLAs

River PLAs

- Um cascadeamento de PLAs de múltiplas saídas.
- PLAs adjacentes são conectados via roteamento river.



- Posicionamento e roteamento não são necessários.
- Buffers de saída e de entrada (do estágio seguinte) são compartilhadas.

Resultados Experimentais

Área:

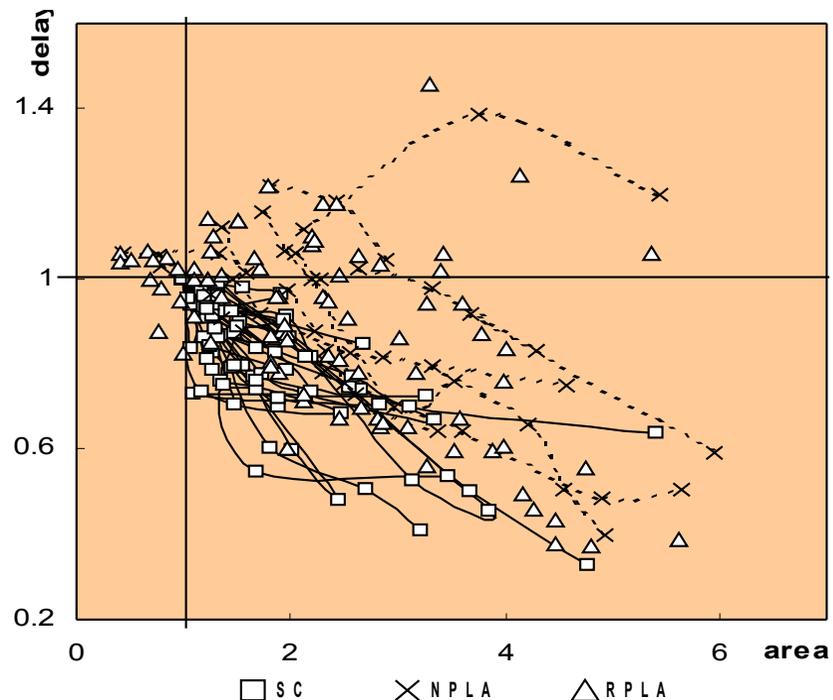
RPLAs (2 layers)	1.23
SCs (3 layers) -	1.00,
NPLAs (4 layers)	1.31

Atraso

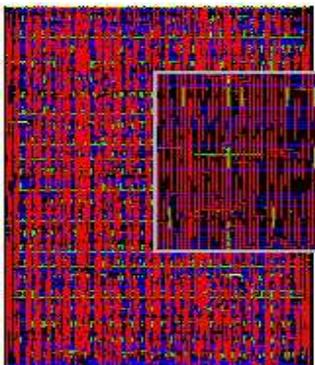
RPLAs	1.04
SCs	1.00
NPLAs	1.09

Tempo de síntese: para RPLA, o tempo de síntese é igual ao tempo de projeto; SCs and NPLAs ainda necessitam de P&R.

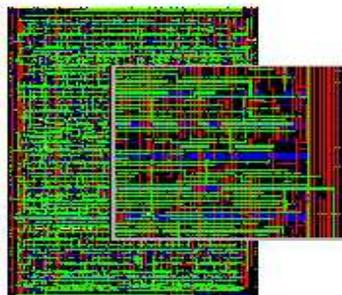
Also: RPLAs são regulares e previsíveis



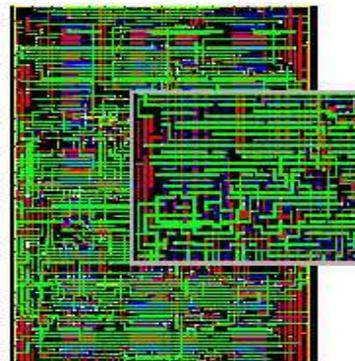
Layout do C2670



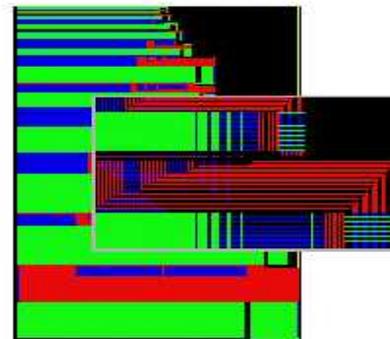
Standard cell,
2 camadas de roteamento de
channel



Standard cell,
3 camadas OTC

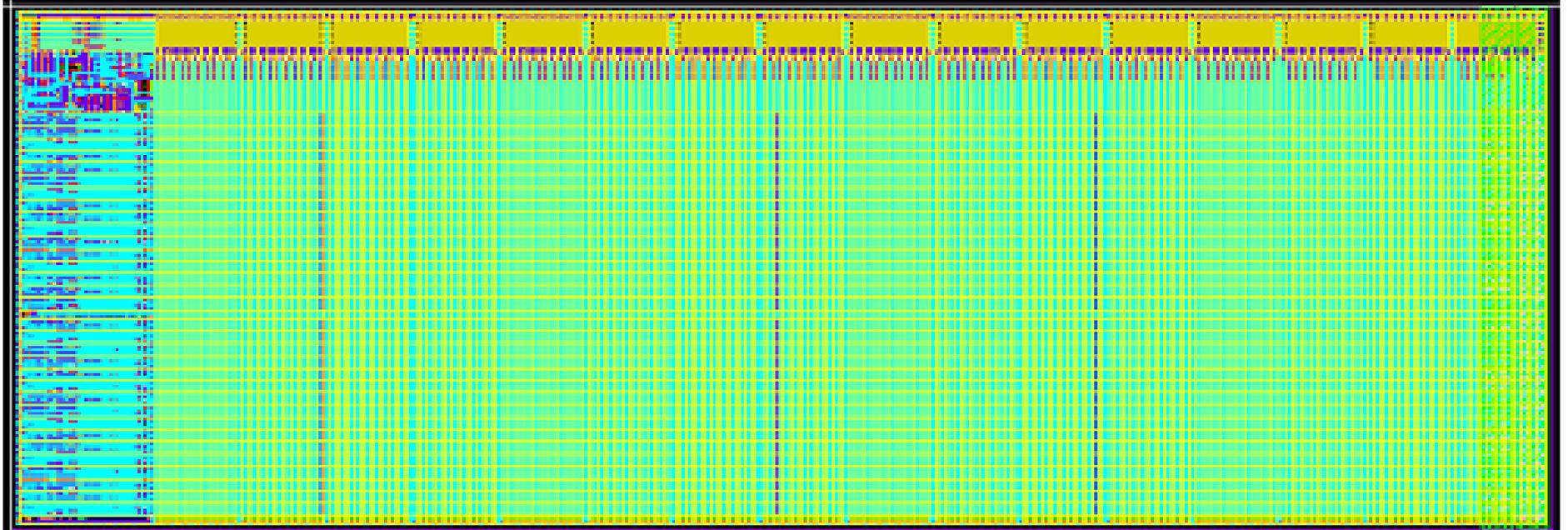


Rede de PLAs,
4 camadas OTC



River PLA,
2 camadas, sem roteamento
adicional

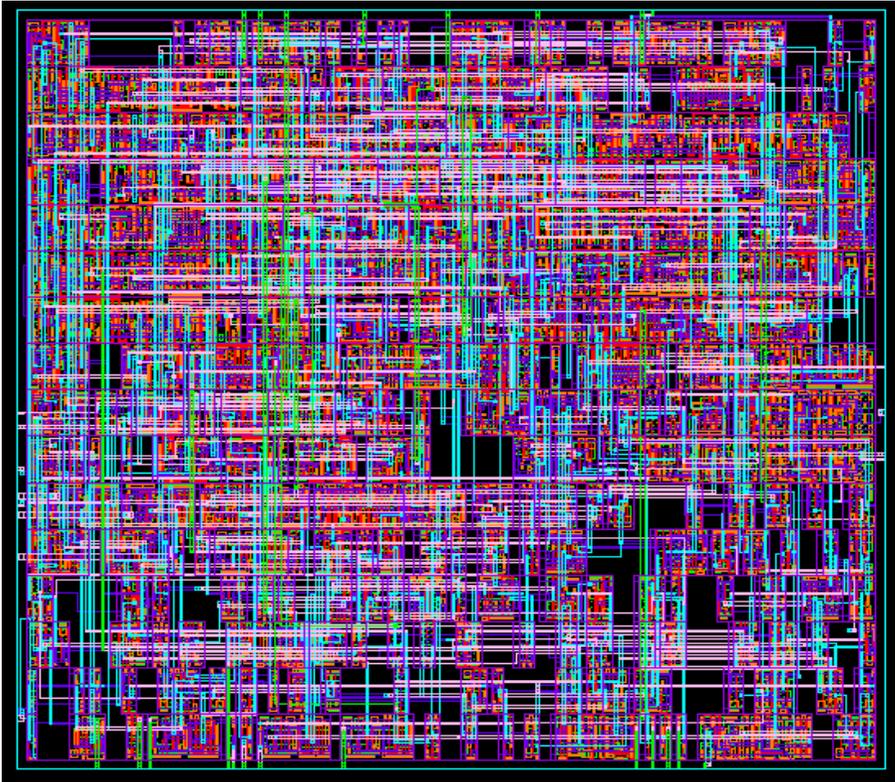
MacroMódulos



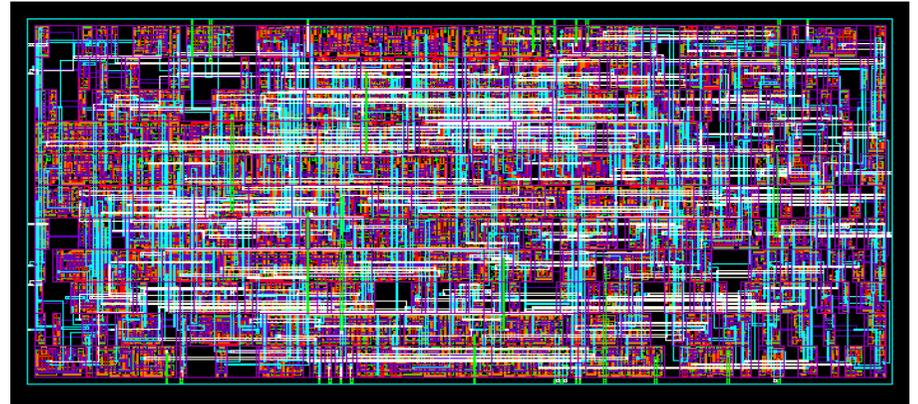
256× 32 (or 8192 bit) SRAM

Obtido com gerador de módulos de hard-macros

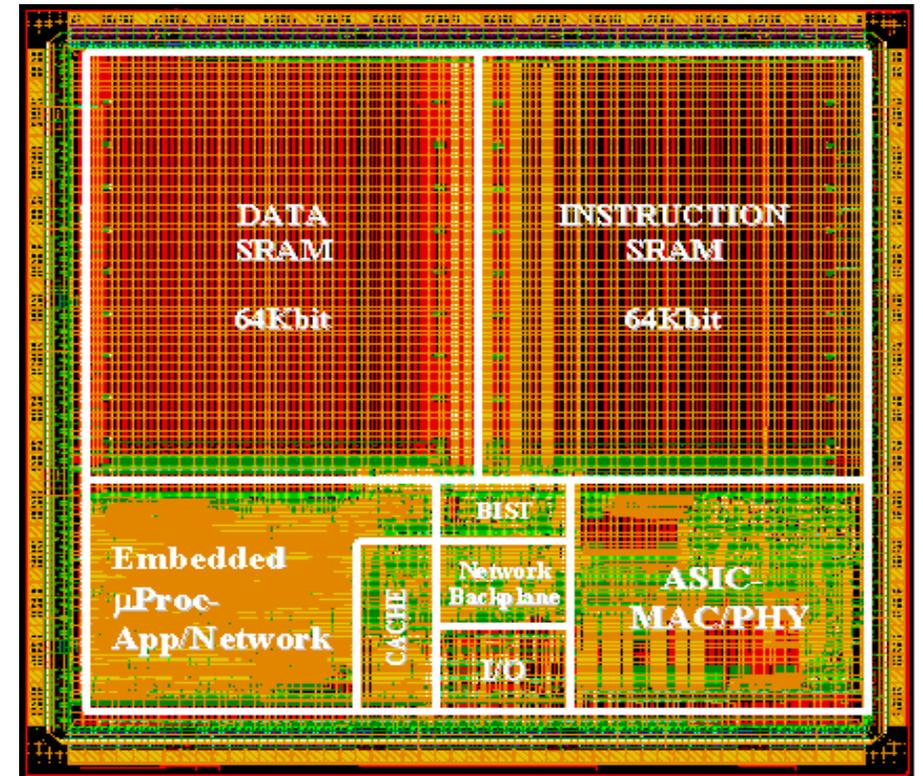
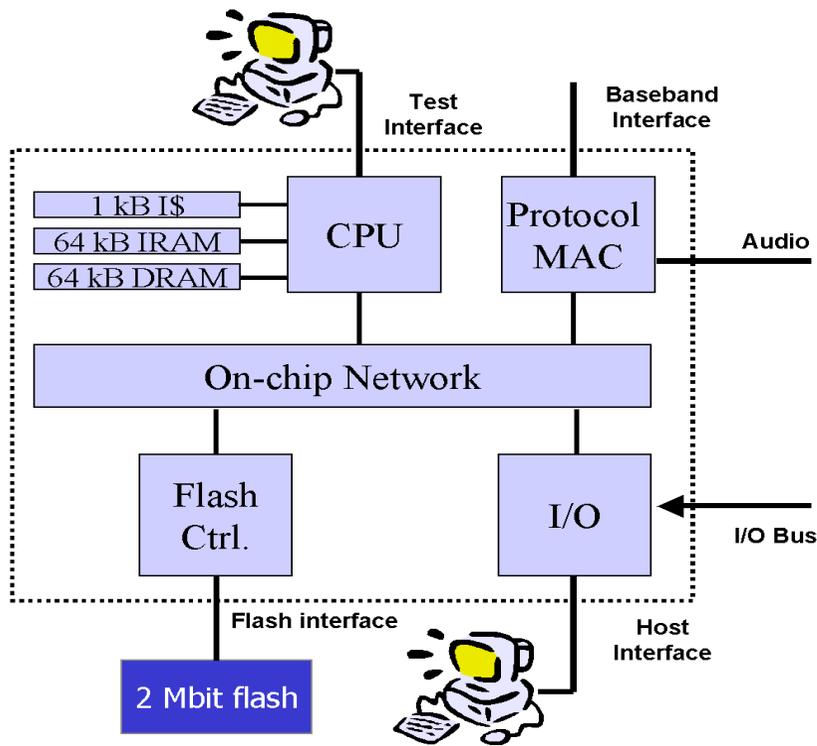
“Soft” MacroMódulos



```
string mat = "booth";  
directive (multtype = mat);  
output signed [16] Z = A * B;
```

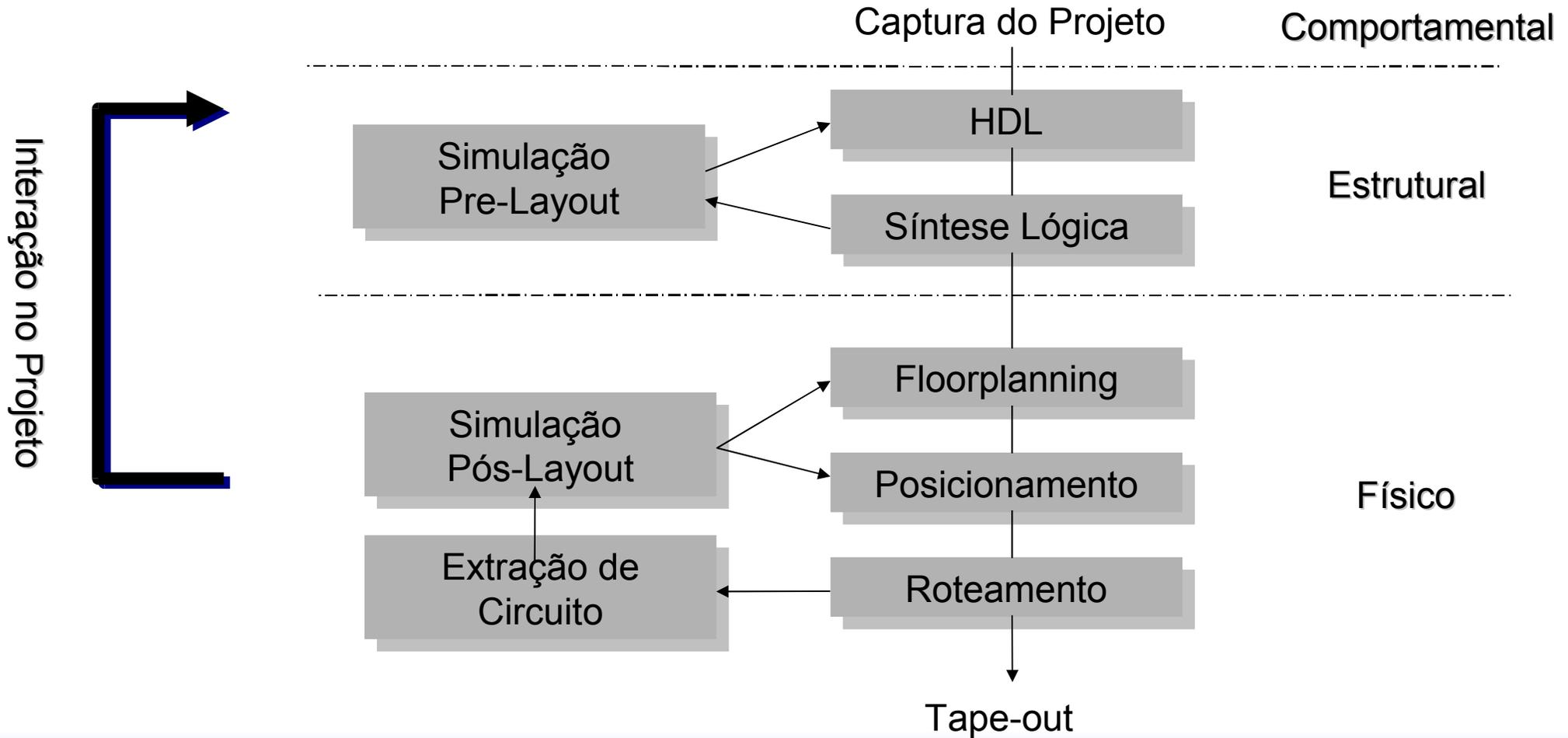


“Propriedade Intelectual (IP)”

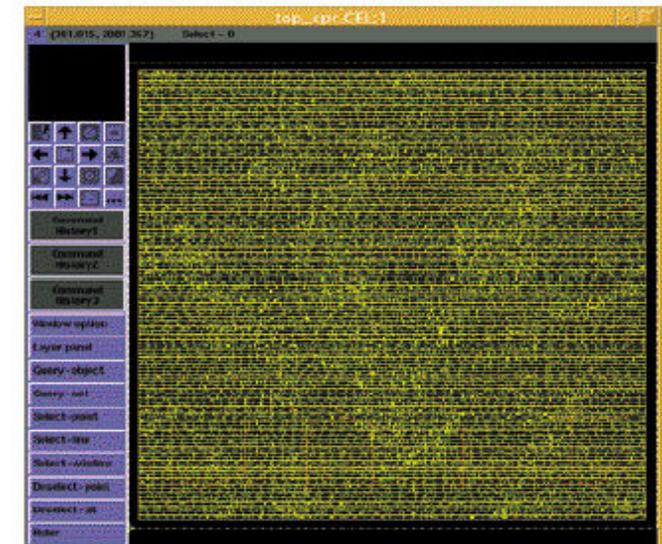
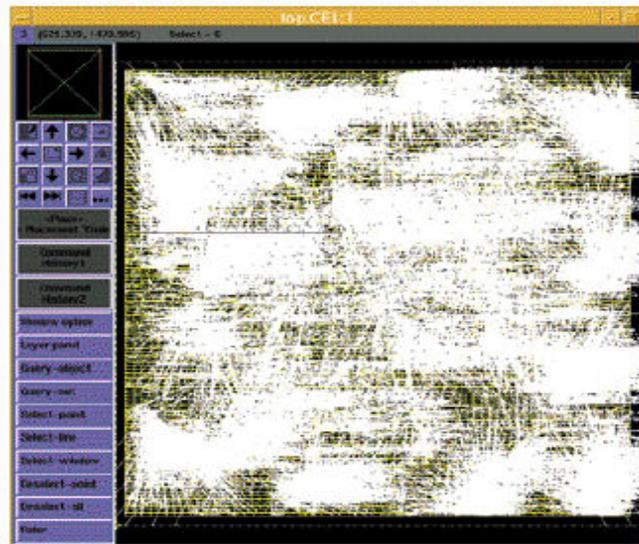


Um Processador de Protocolo Wireless

Fluxo de Projeto Semidedicado

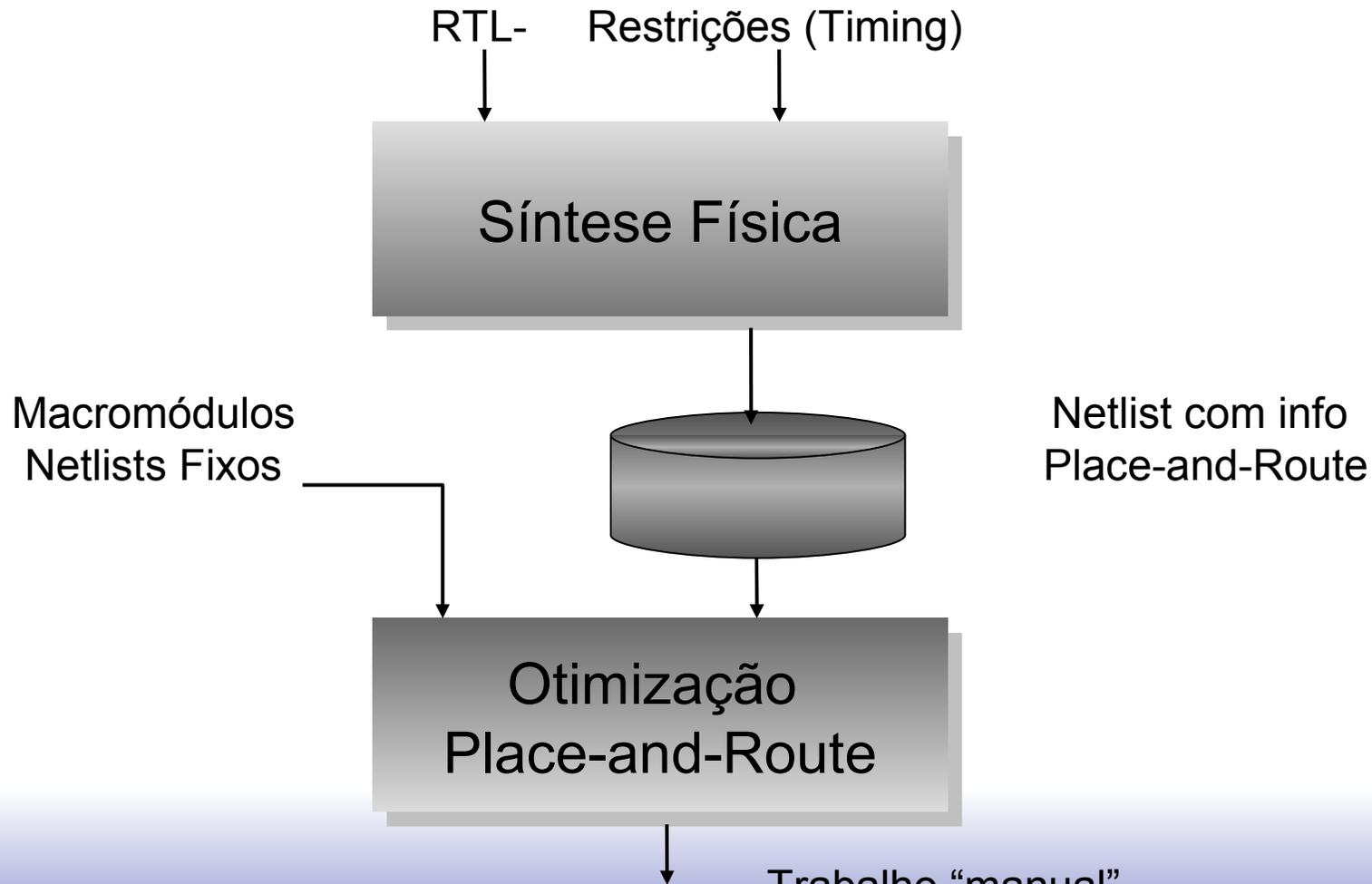


O Problema do “Design Closure”

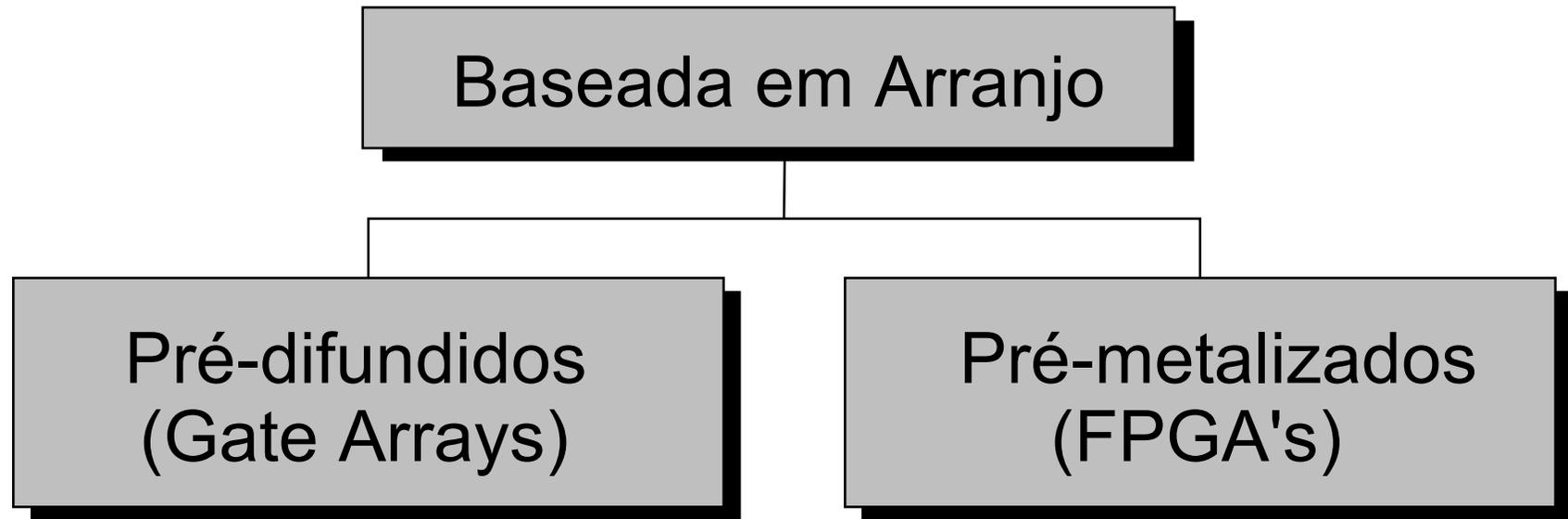


Remoção Iterativa de Violações de Timing (linhas brancas)

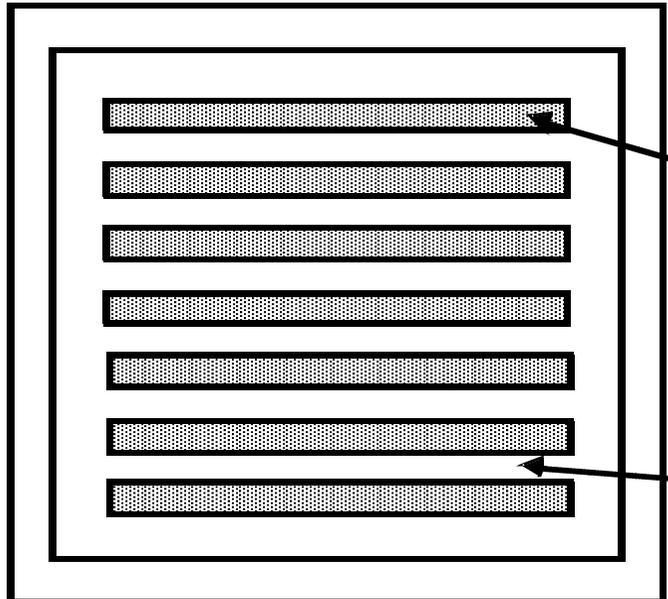
Integrando Síntese com Projeto Físico



Implementação com Associação Potergada

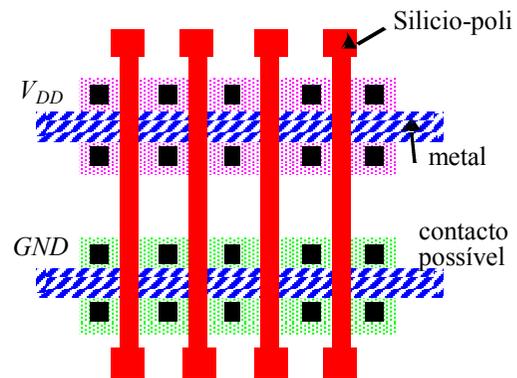


Gate Array — Mar de Portas (Sea-of-gates)

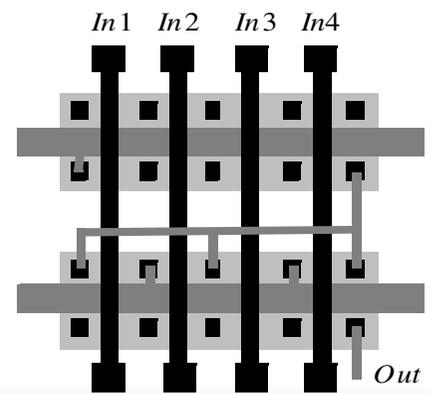


fileiras de células não-personalizadas

canal de roteamento

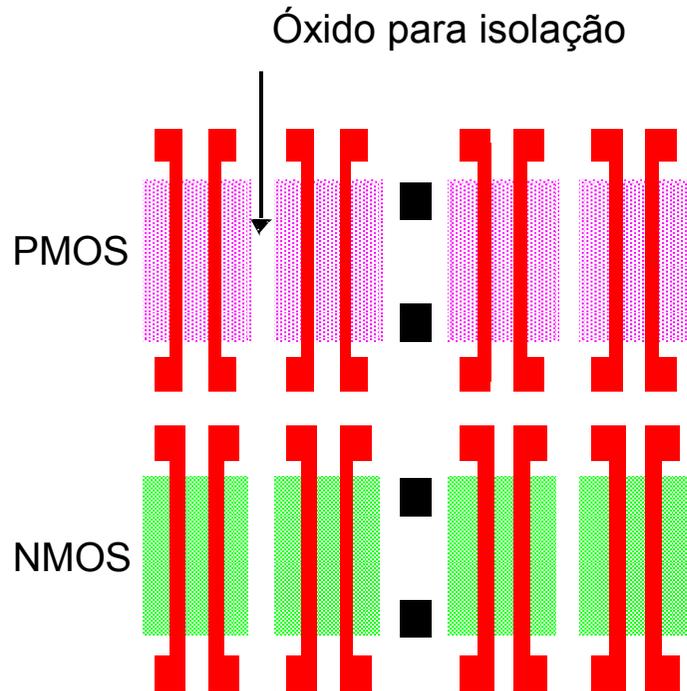


Célula não-personalizada

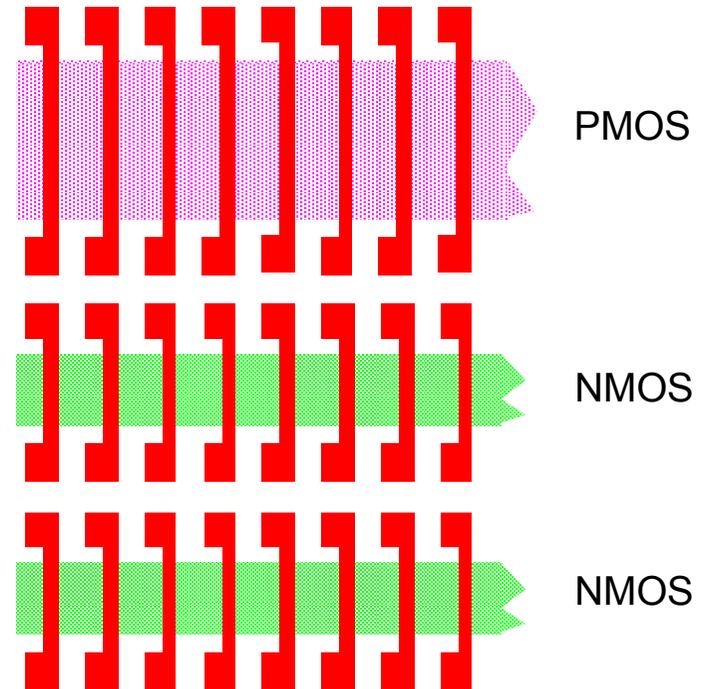


Célula personalizada (4-input NOR)

Células Primitivas de Sea-of-gates

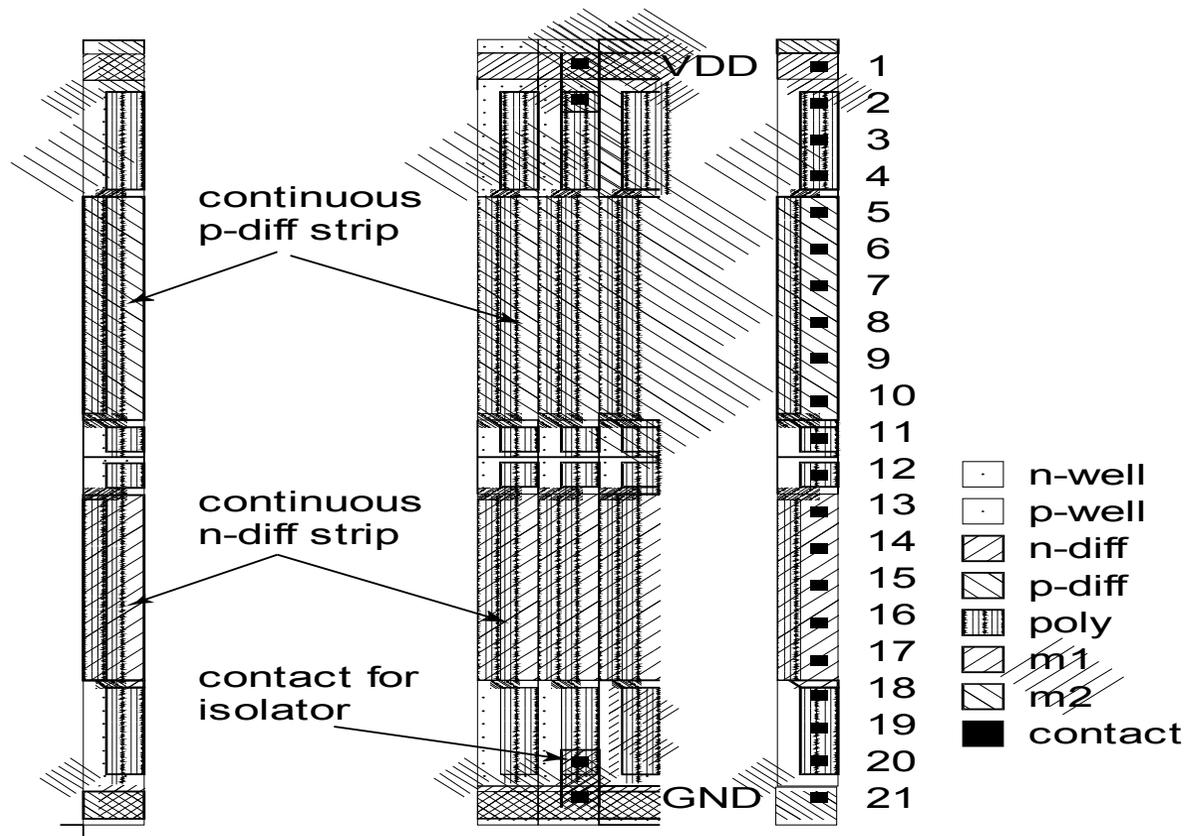


Usando isolamento por óxido

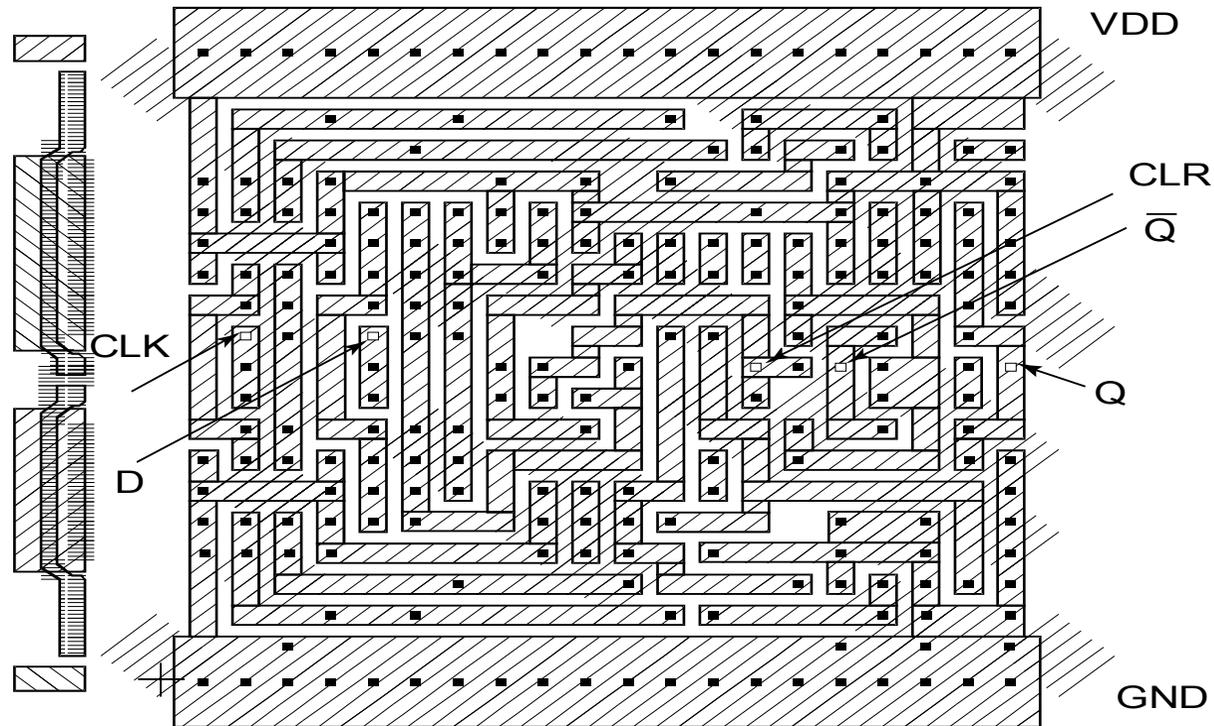


Usando isolamento por porta

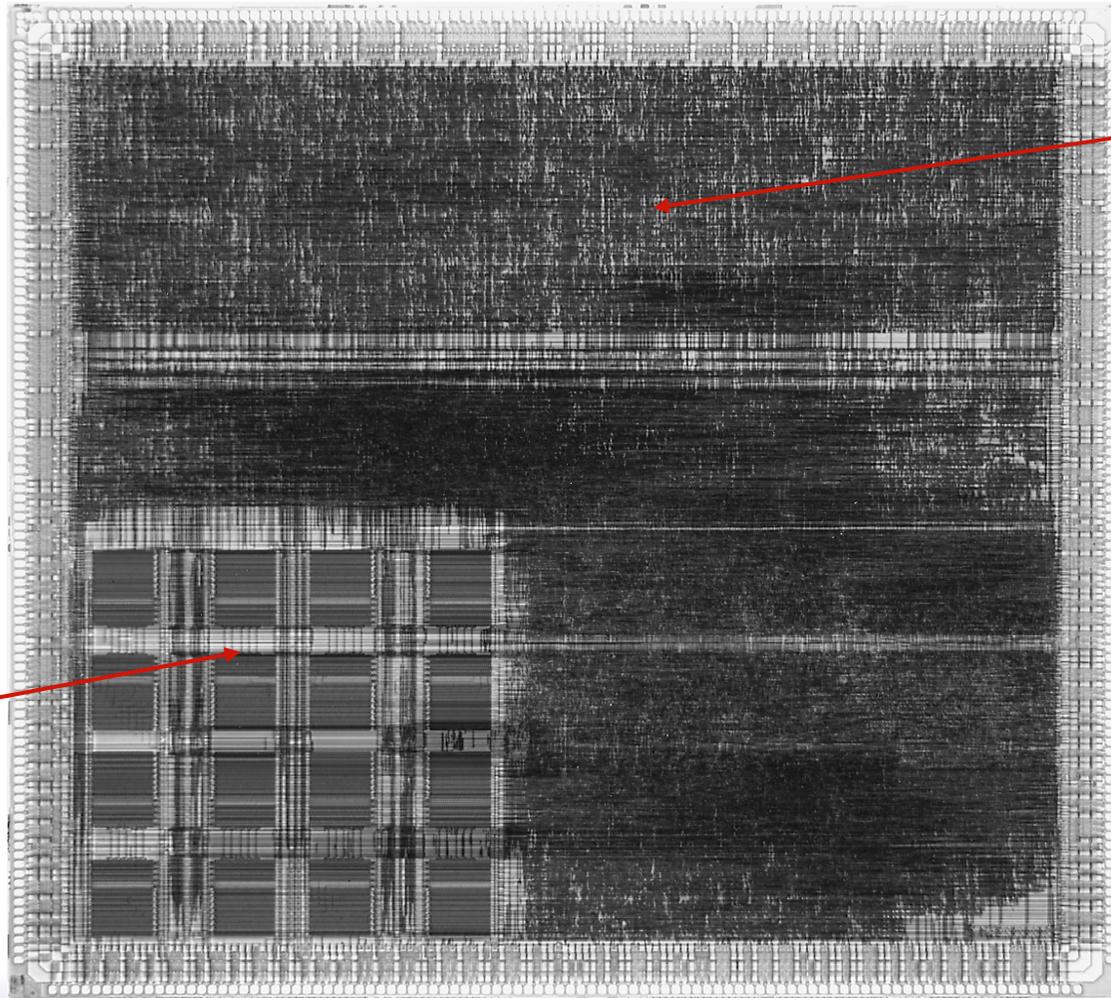
Exemplo: Célula-base de GA Isolado por Porta



Exemplo: Flip-Flop em GA Isolado por Porta



Sea-of-gates



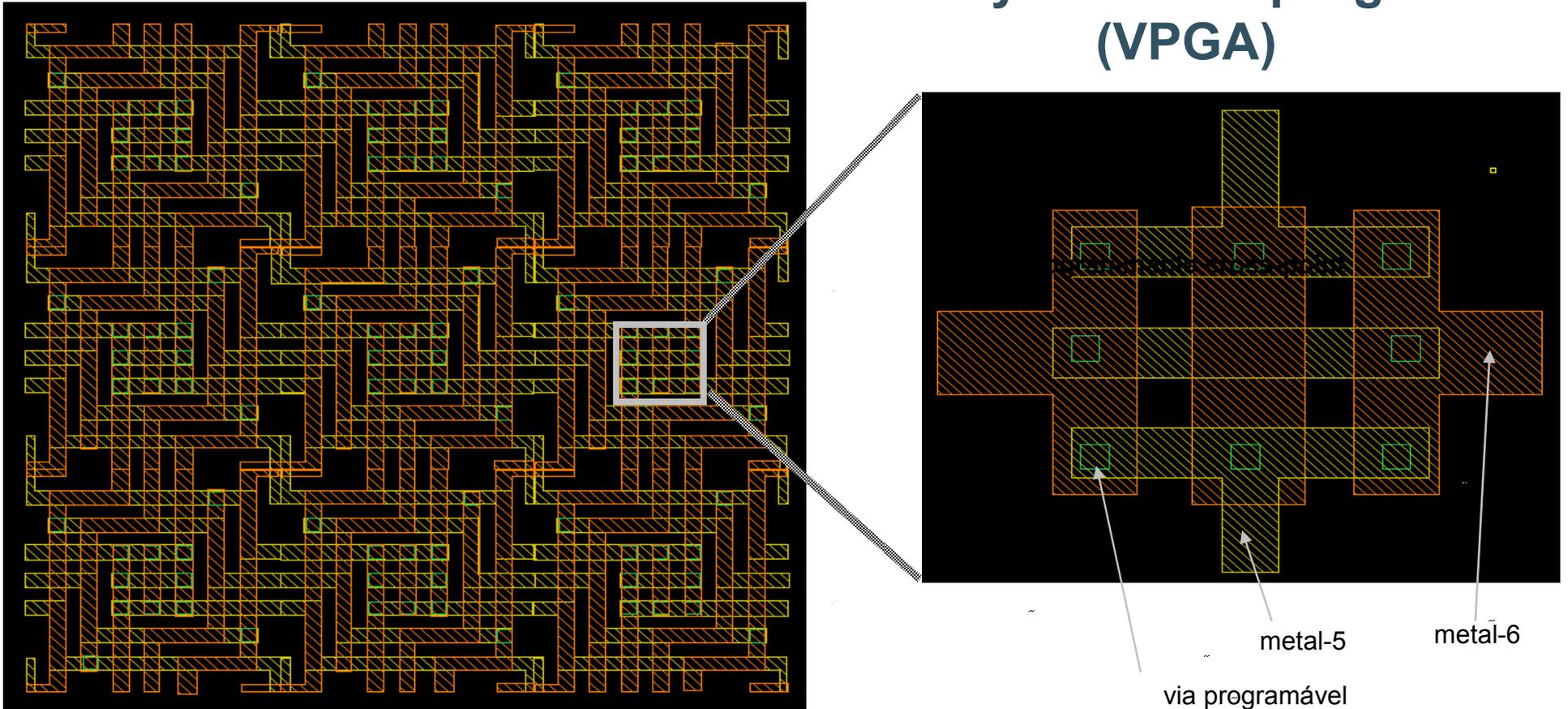
Subsistema
de Memória

Lógica
Aleatória

LSI Logic LEA300K
(0.6 μm CMOS)

O retorno dos gate arrays?

Gate array com vias programáveis
(VPGA)



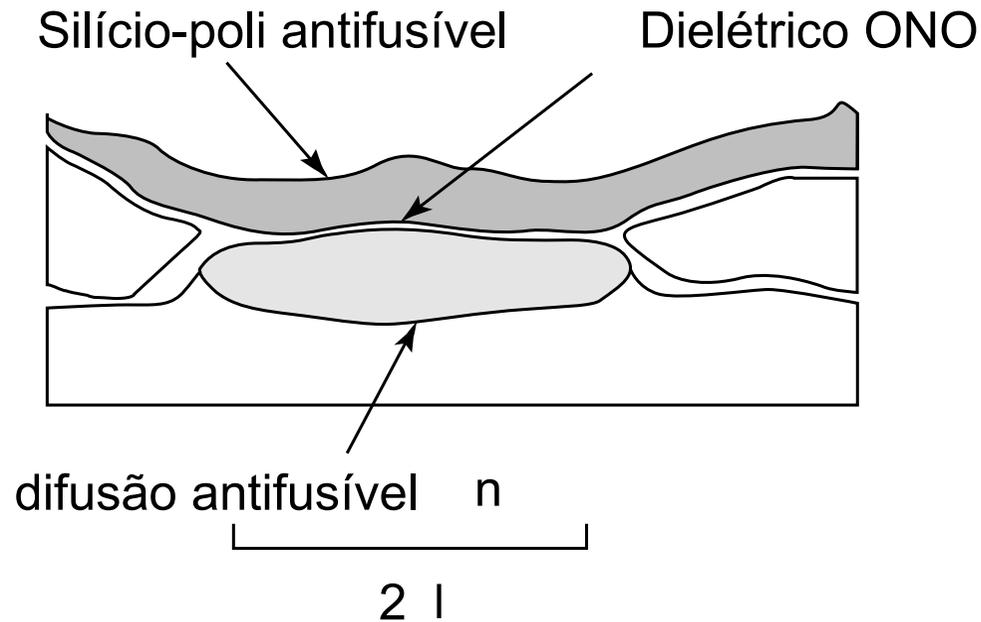
Explora a regularidade das interconexões

Arranjos Pré-Metalizados (Prewired Arrays)

Classificação de arranjos pré-metalizados (ou dispositivos programáveis em campo → field-programmable devices):

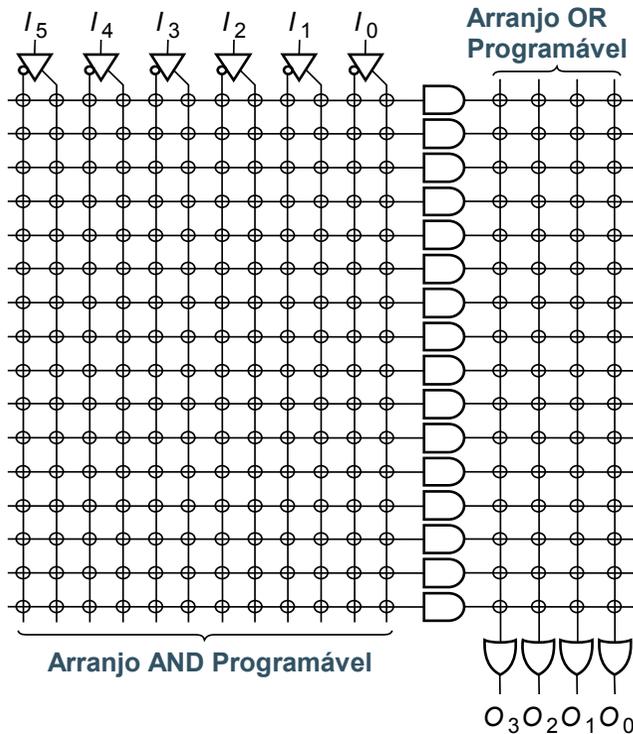
- Baseados em Técnica de Programação
 - Baseados em fusível (programação única)
 - Não voláteis, baseados em EPROM RAM
- Estilo da Lógica Programável
 - Baseado em arranjo
 - Look-up Table (memória RAM)
- Estilo da Interconexão Programável
 - Roteamento por canal
 - Redes malha (*mesh*)

FPGA Baseada em Fusível



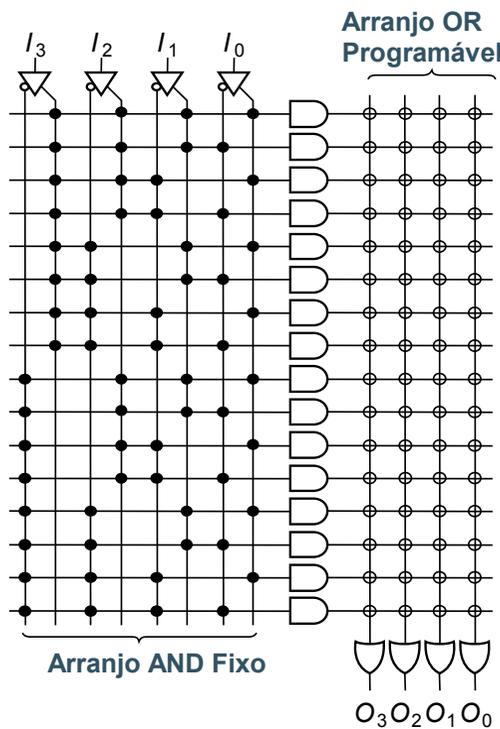
Aberta (default), fechada por aplicação de pulso de corrente

Lógica Programável baseada em Arranjo



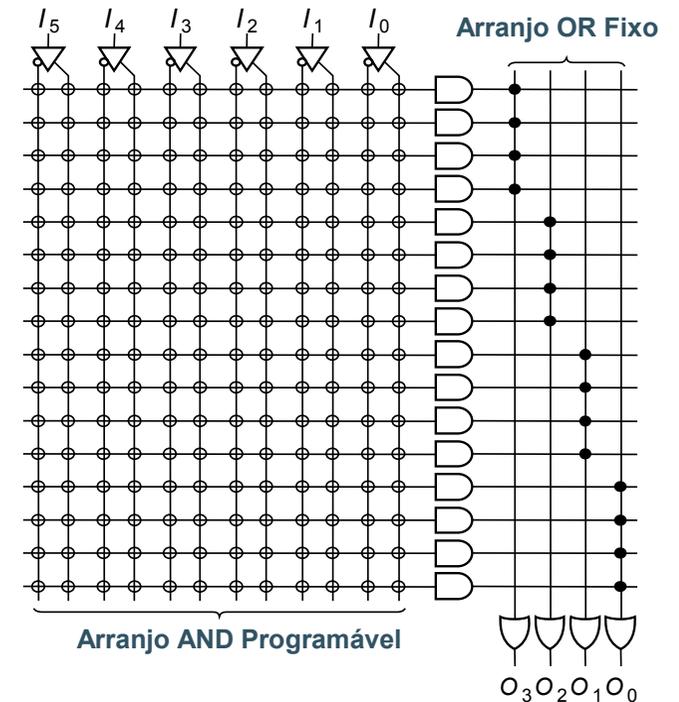
PLA

Arranjo Lógico Programável



PROM

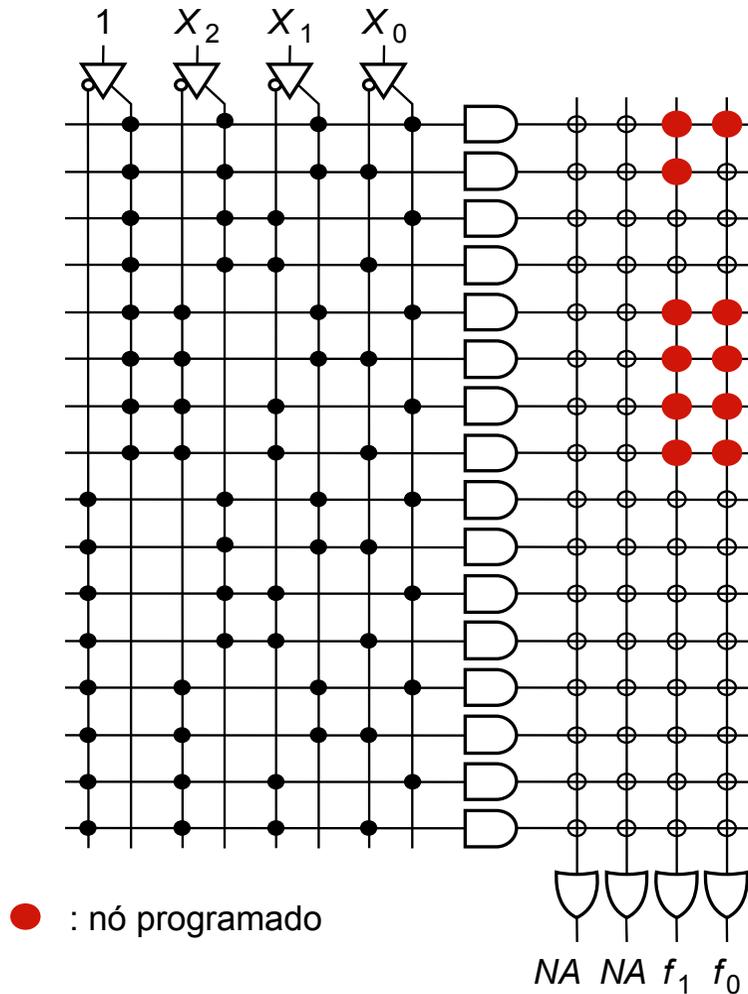
- ⊕ Indica conexão programável
- ◆ Indica conexão fixa



PAL

Lógica de Arranjos Programável

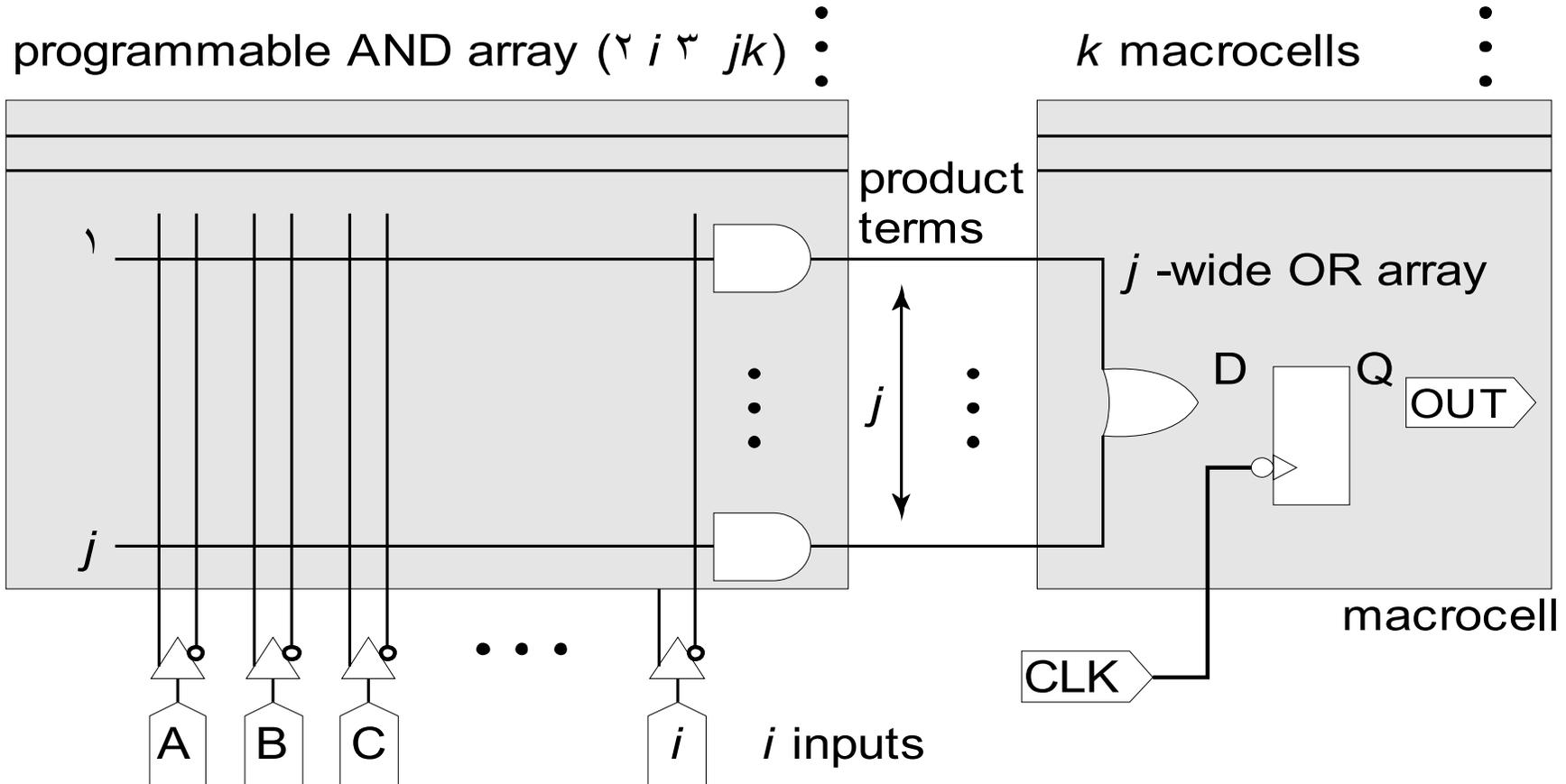
Programando uma PROM



$$f_0 = x_0x_1 + \overline{x_2}$$

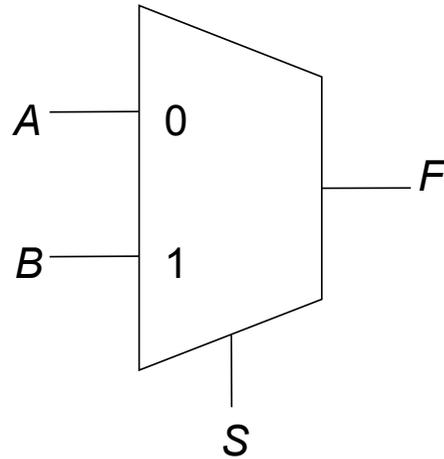
$$f_1 = x_0x_1x_2 + \overline{x_2} + \overline{x_0}x_1$$

PAL Mais Complexa



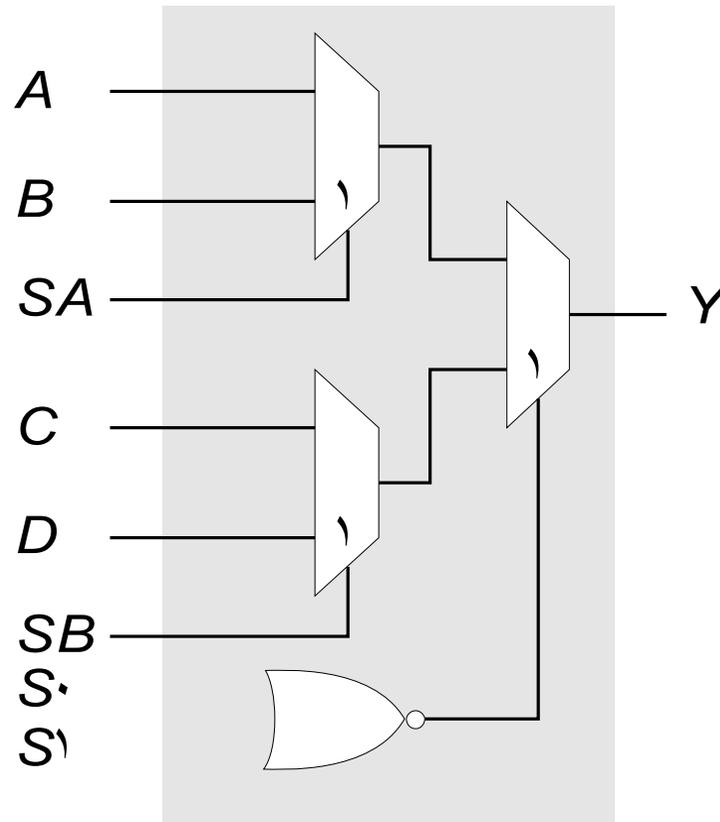
i entradas, j mintermos/macrocélulas, k macrocélulas

Mux de 2 entradas como bloco lógico programável

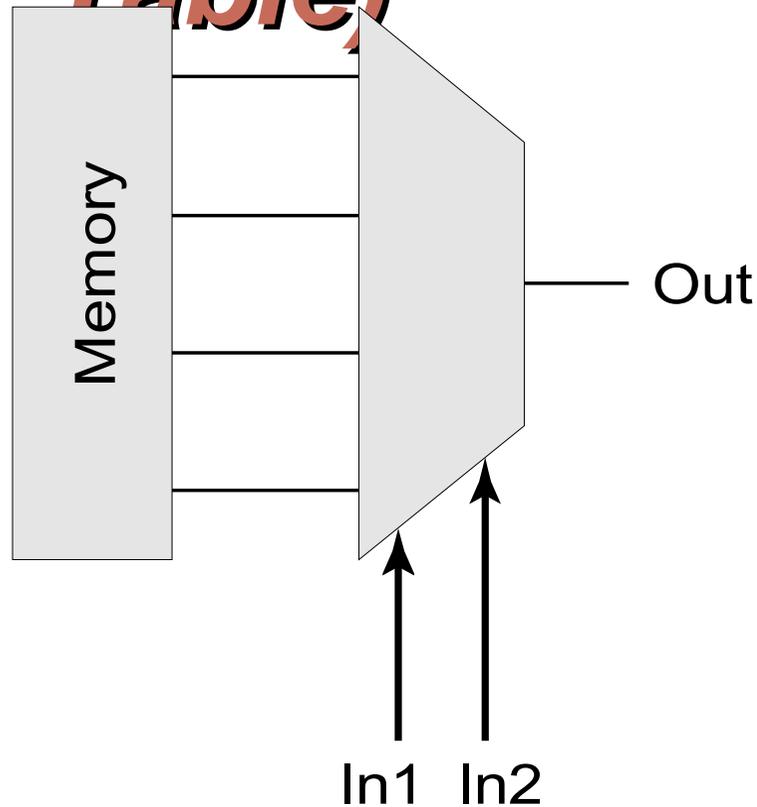


Configuração			F=
A	B	S	
0	0	0	0
0	X	1	X
0	Y	1	Y
0	Y	X	XY
X	0	Y	$X\bar{Y}$
Y	0	X	$\bar{X}Y$
Y	1	X	$X \frac{1}{Y}$
1	0	X	\bar{X}
1	0	Y	\bar{Y}
1	1	1	1

Célula Lógica da FPGA Baseada em Fusível (Actel)

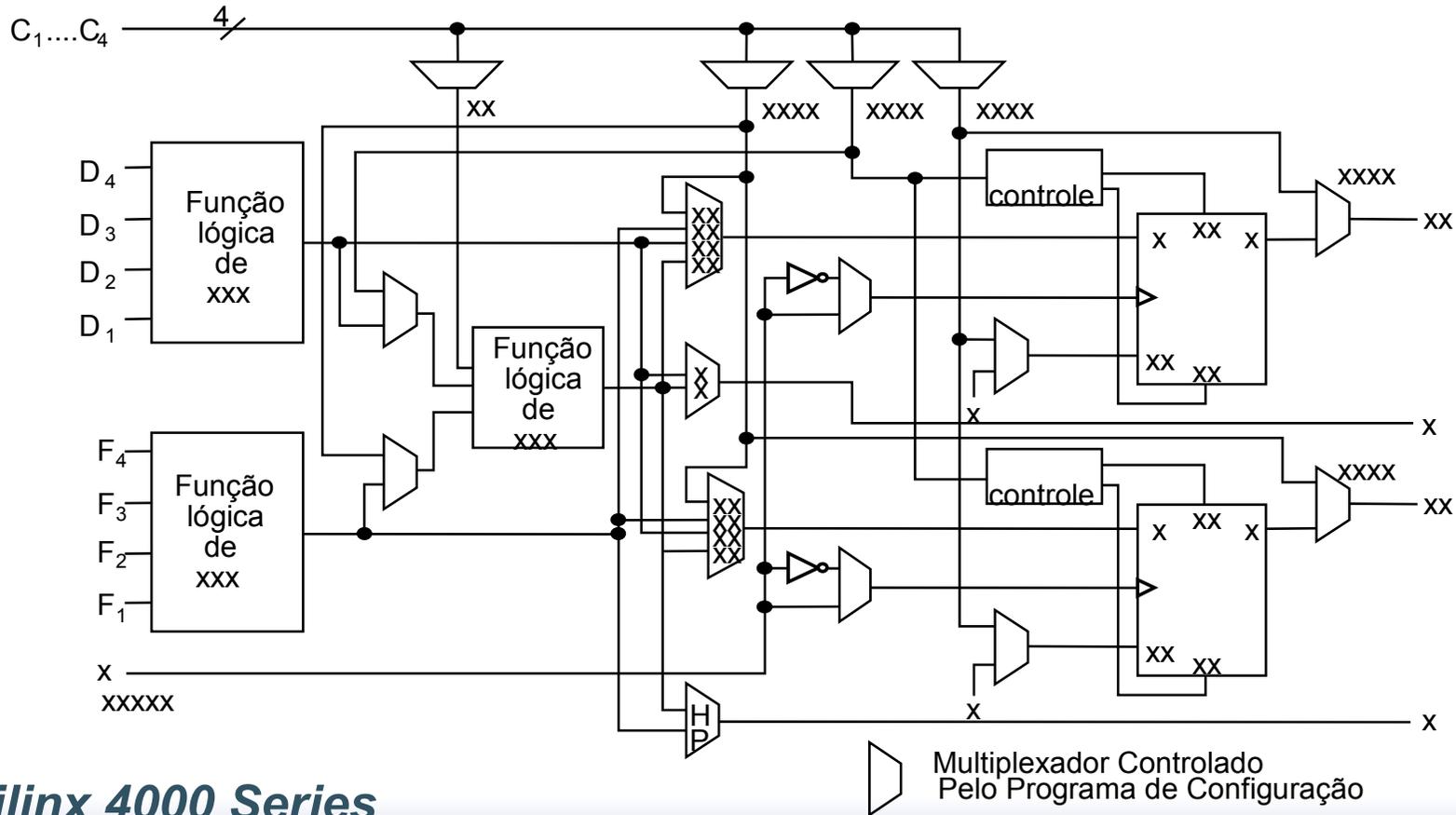


Célula Lógica Baseada em Tabela de Referência (LUT-Look-up Table)



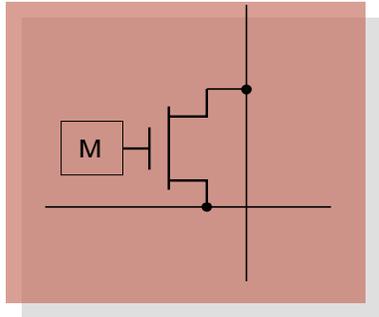
In	Out
00	00
01	1
10	1
11	0

Célula Lógica Baseada em LUT

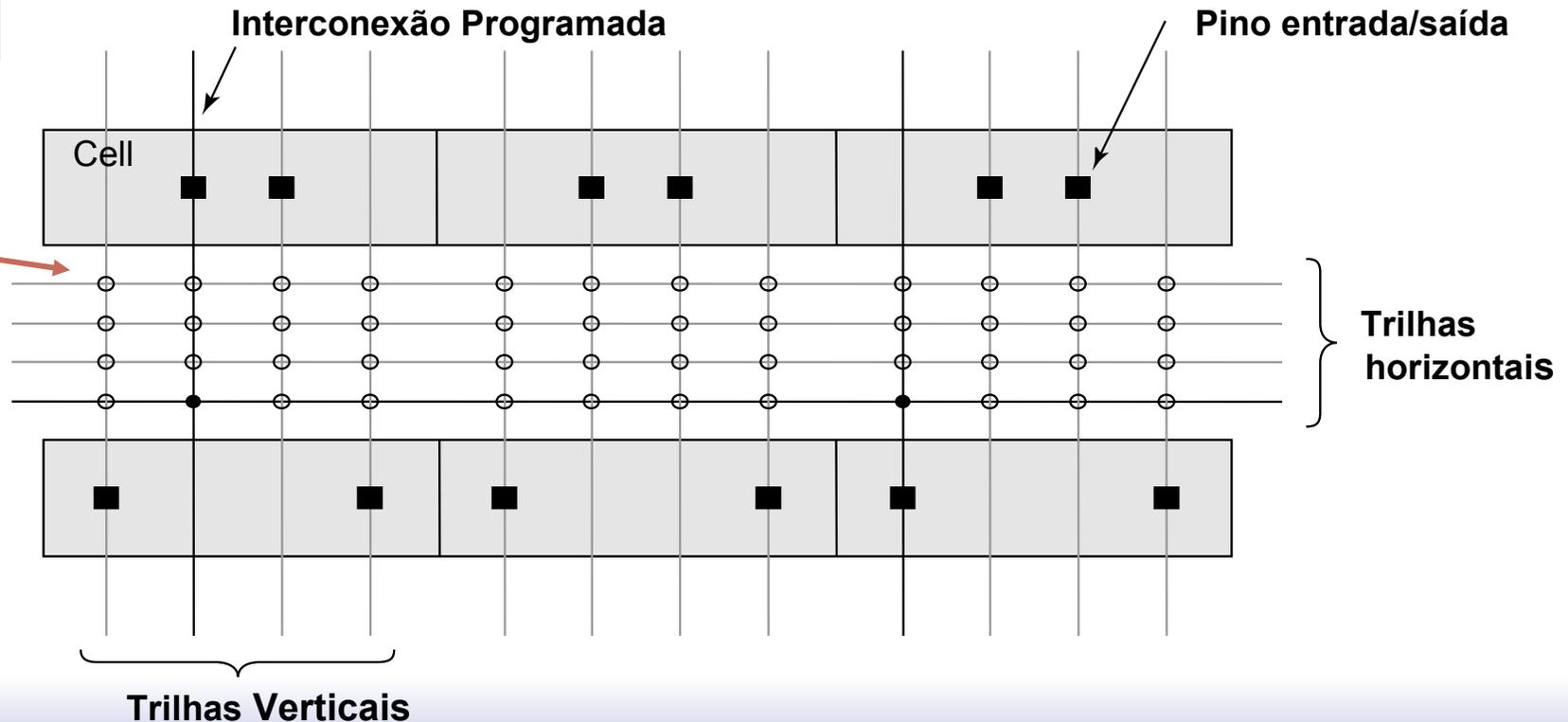


Xilinx 4000 Series

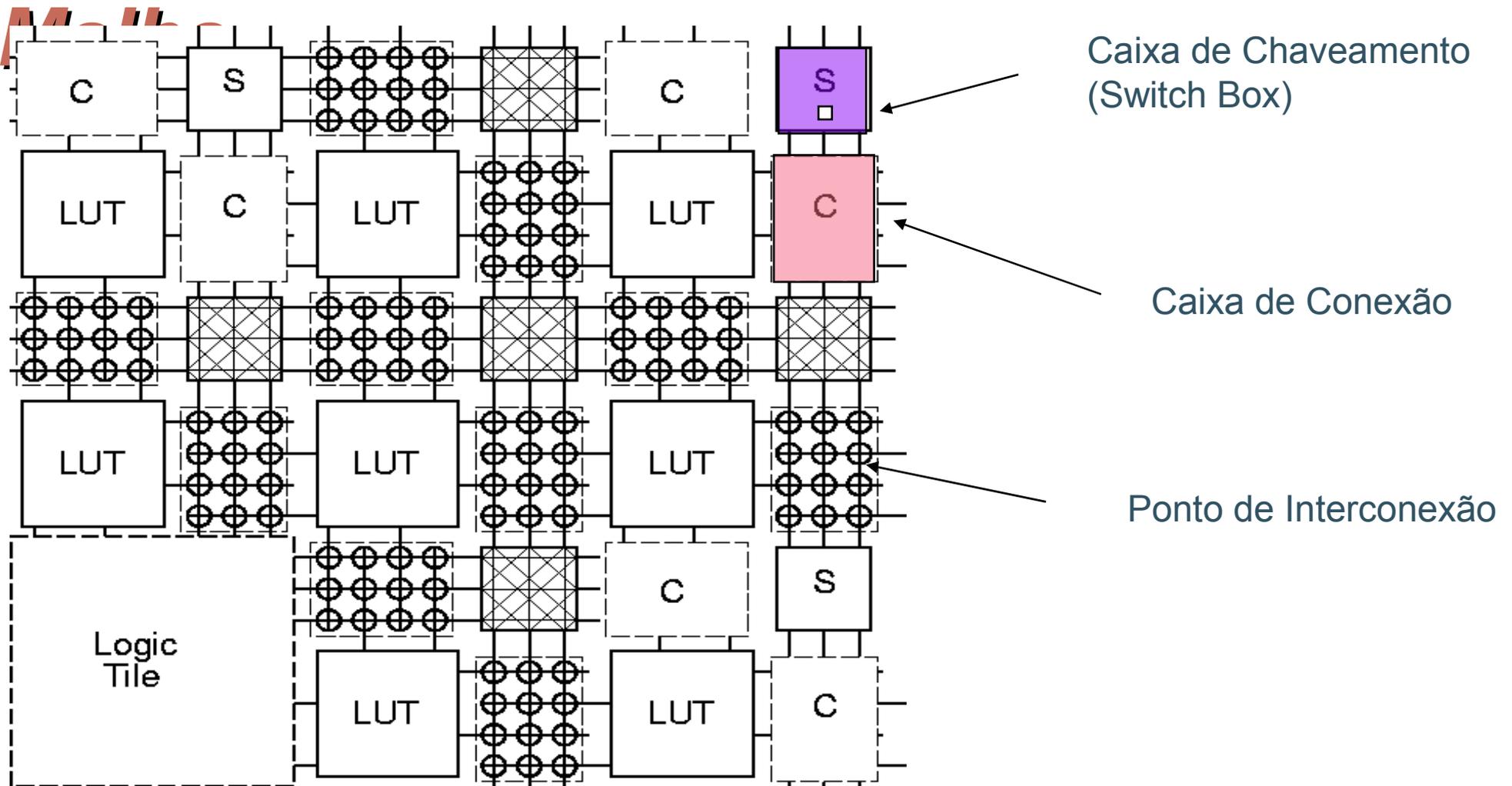
Interconexão Programável Baseada em Arranjo



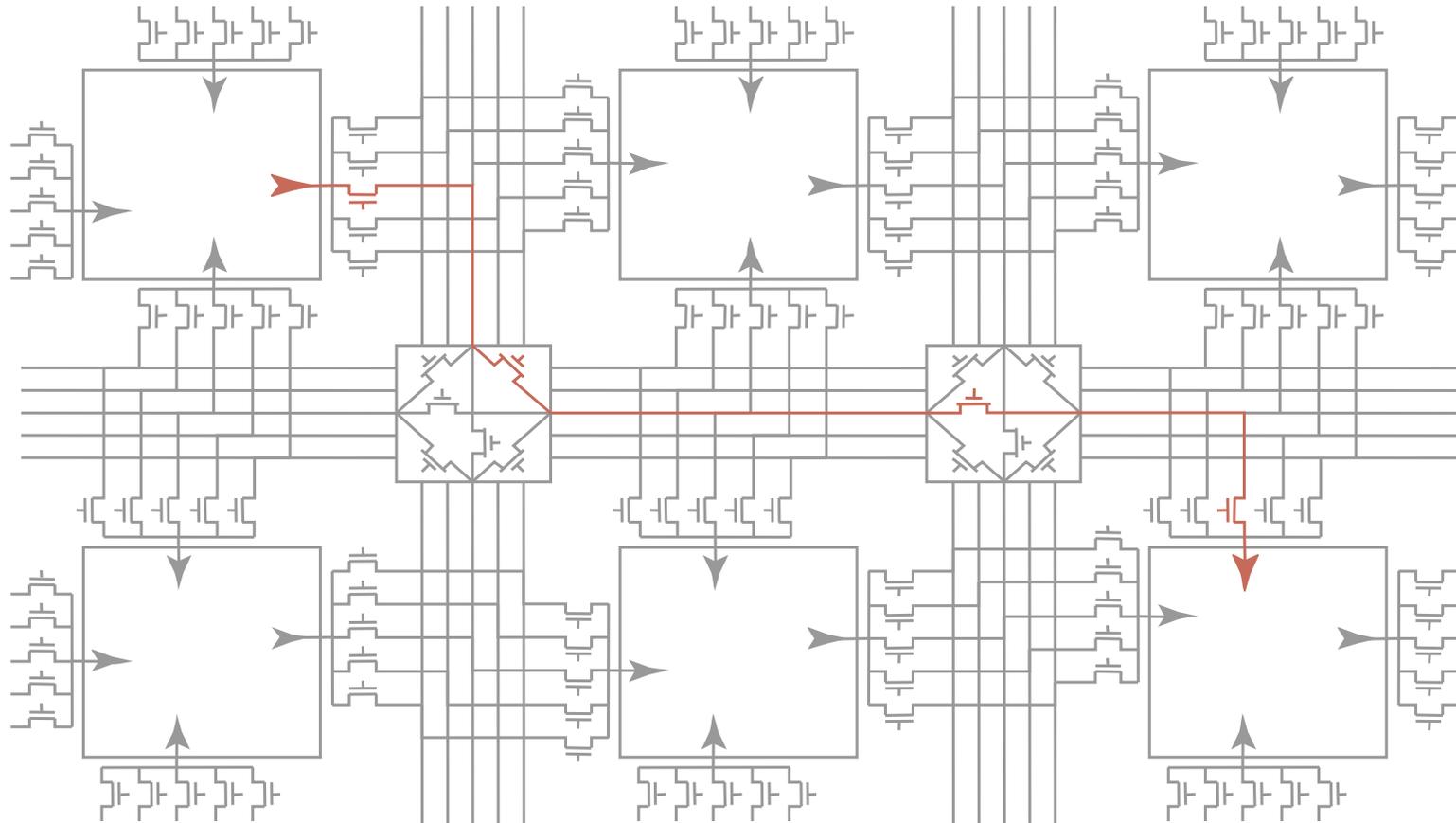
Ponto de
Interconexão



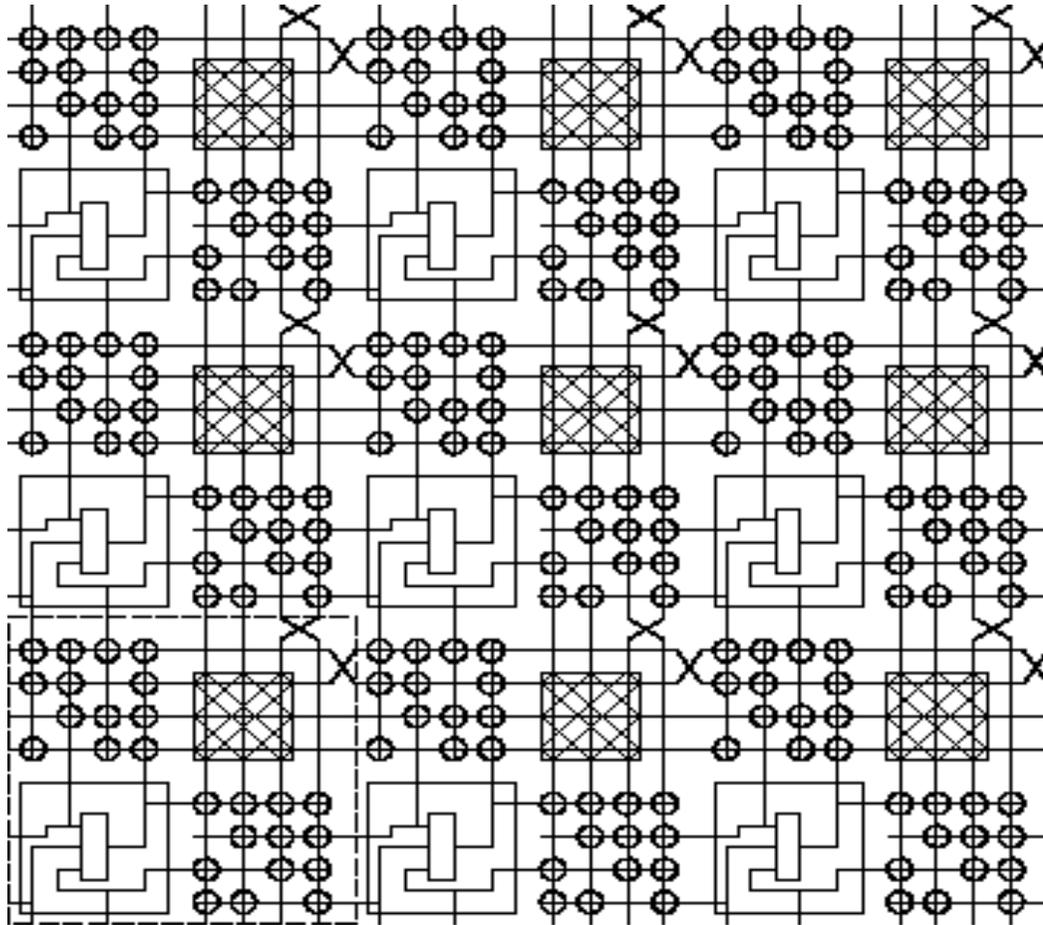
Rede de Interconexão Baseada em



Implementação de Malha por Transistor



Rede Hierárquica em Malha

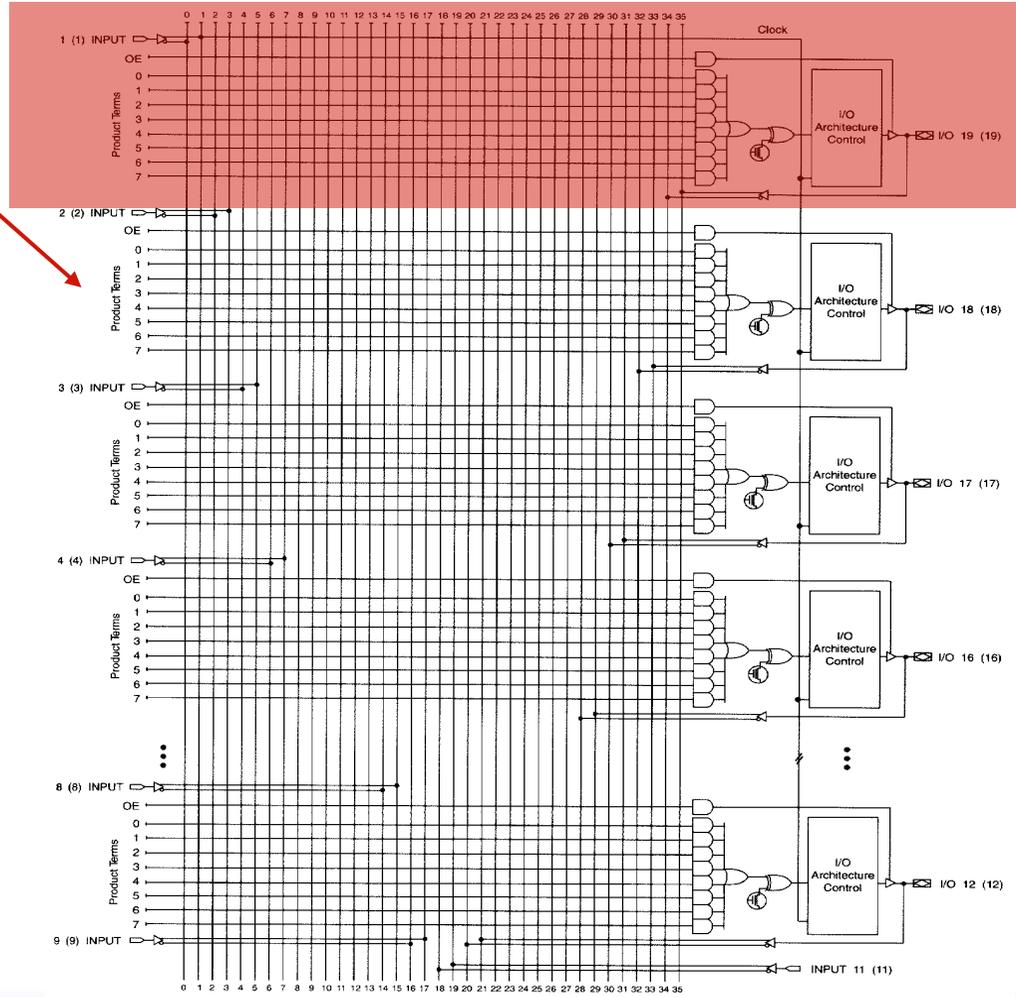


Usa malha sobreposta para suportar conexões + longas

Fanout e resistência reduzidos

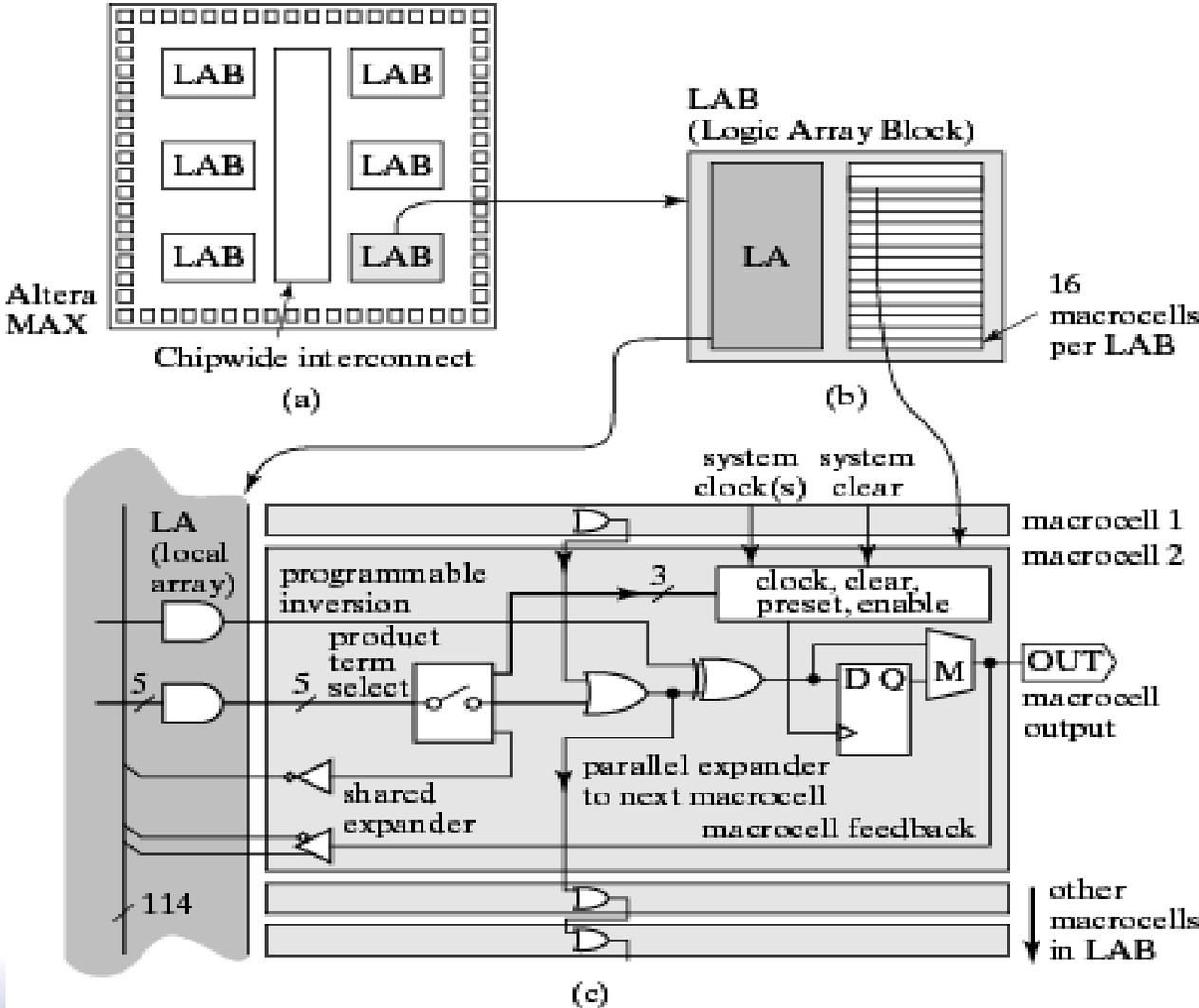
Diagrama de Blocos de EPLD

Entradas primárias

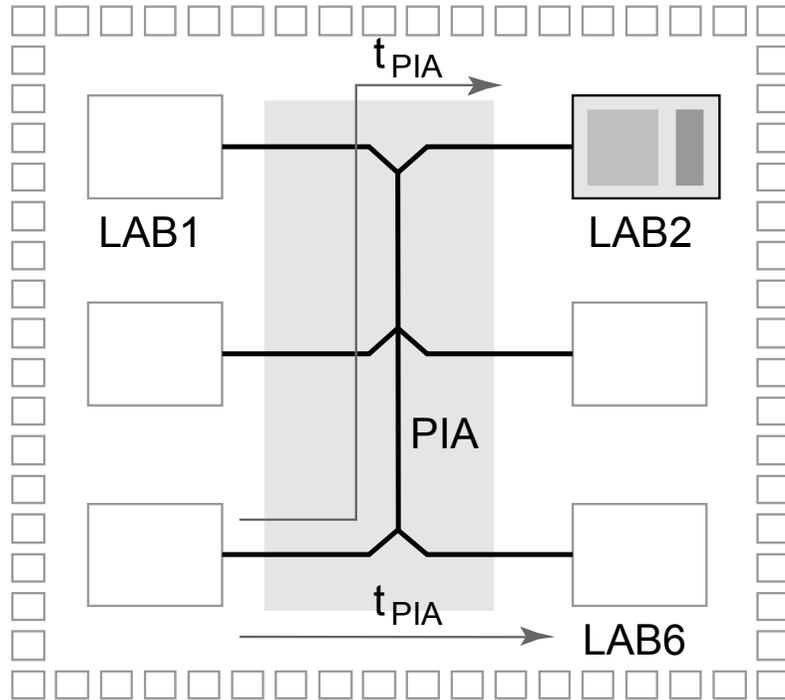


Macro célula

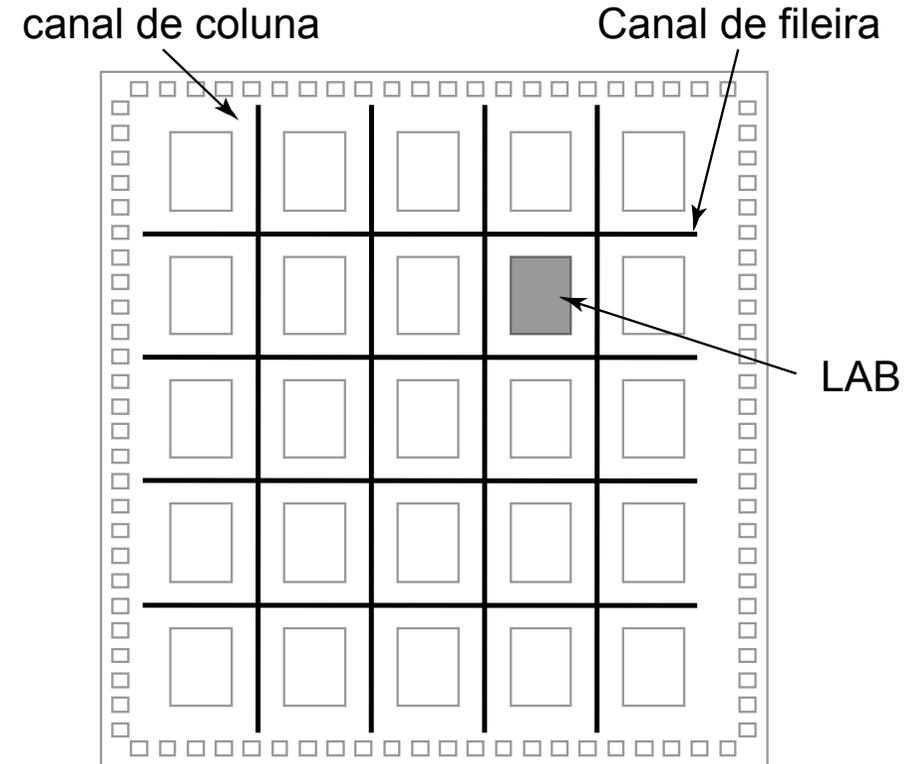
Altera MAX



Arquitetura de Intercção- Altera MAX

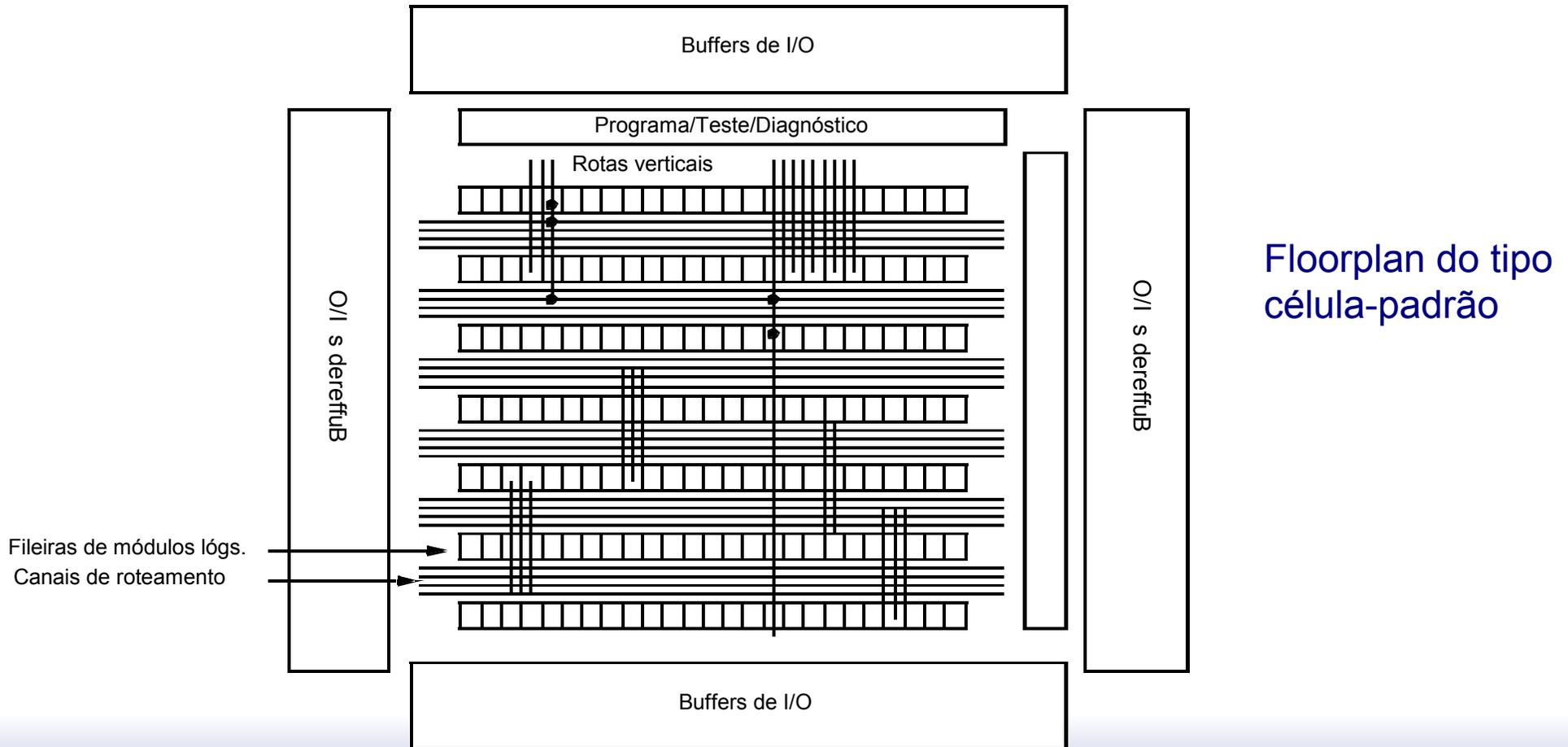


**Baseado em Arranjo
(MAX 3000-7000)**

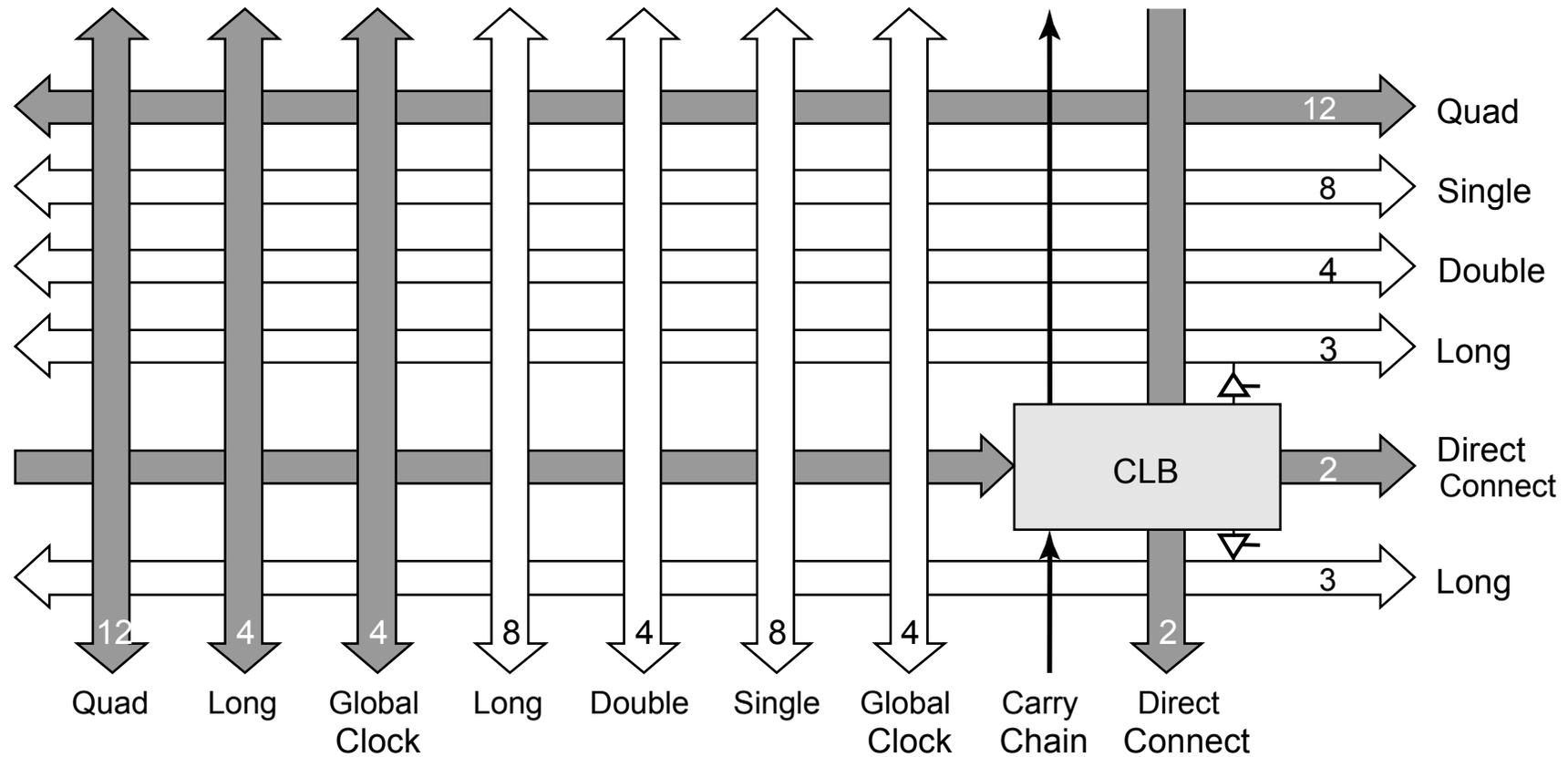


**Baseado em Malha
(MAX 9000)**

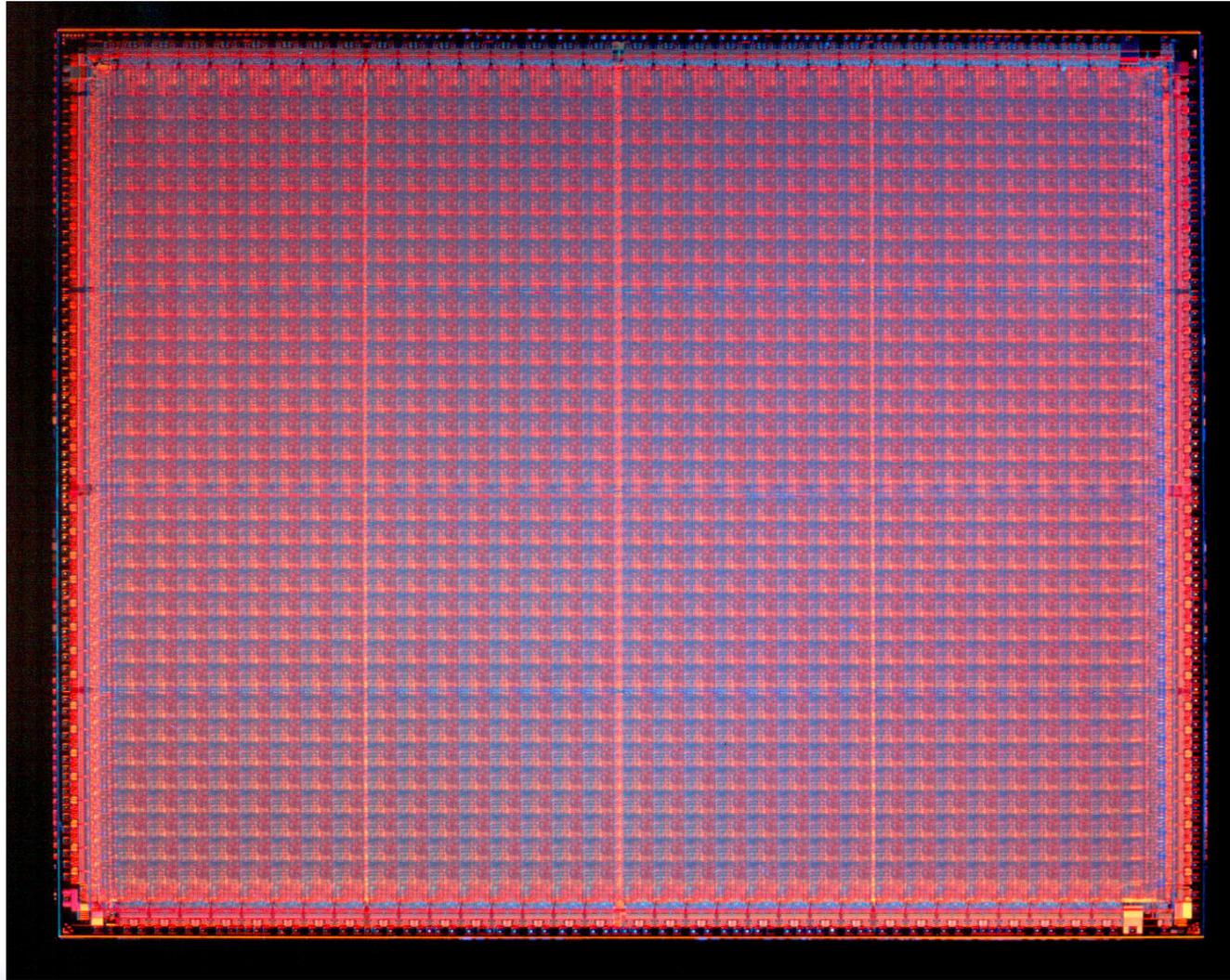
FPGAs- Field-Programmable Gate Arrays Baseados em Fusível



Arquitetura de Intercorção- Xilinx 4000

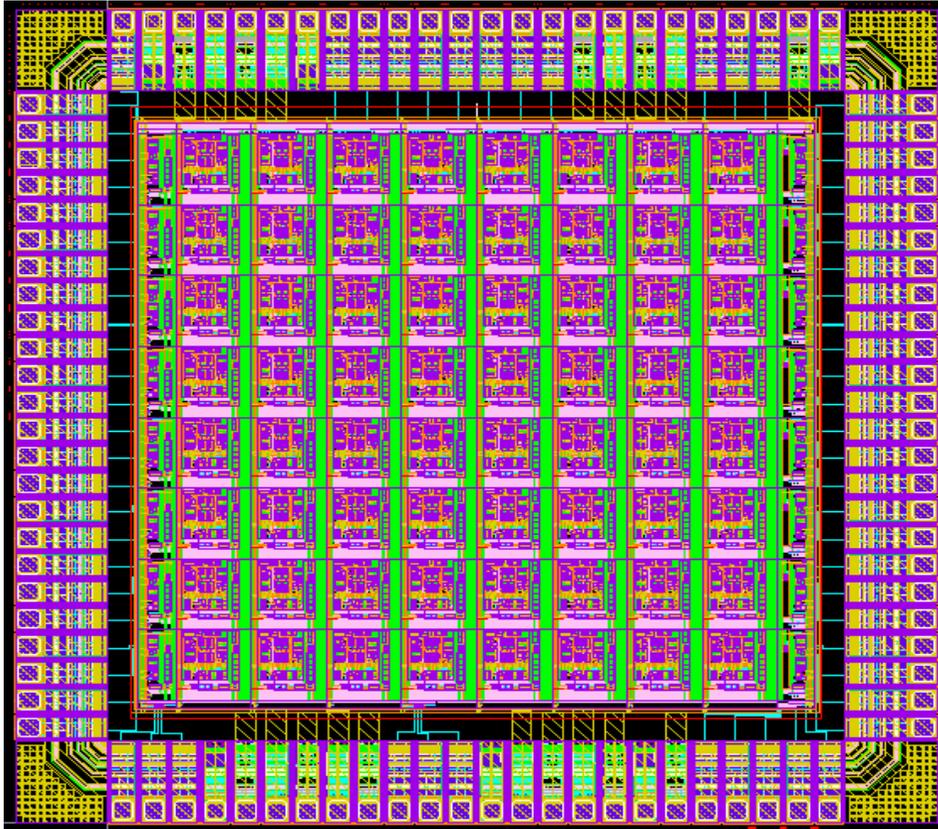


FPGA Baseada em RAM



Xilinx XC4000ex

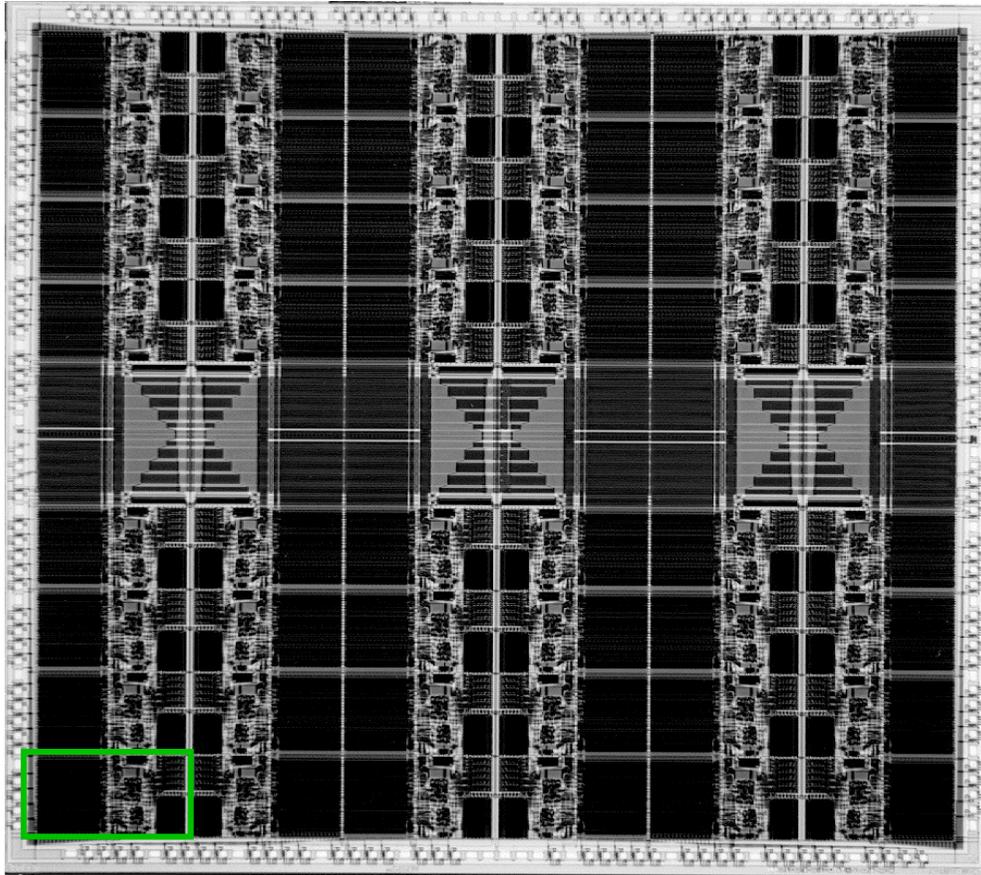
FPGA de Baixa Energia (UC Berkeley)



- ❑ Tamanho do arranjo: 8x8 (2 x 4 LUT)
- ❑ Fonte de Tensão: 1.5V & 0.8V
- ❑ Configuração: Mapeado como RAM
- ❑ Frequência: 125MHz
- ❑ Área: 3mm x 3mm

FPGAs de Alta Granularidade

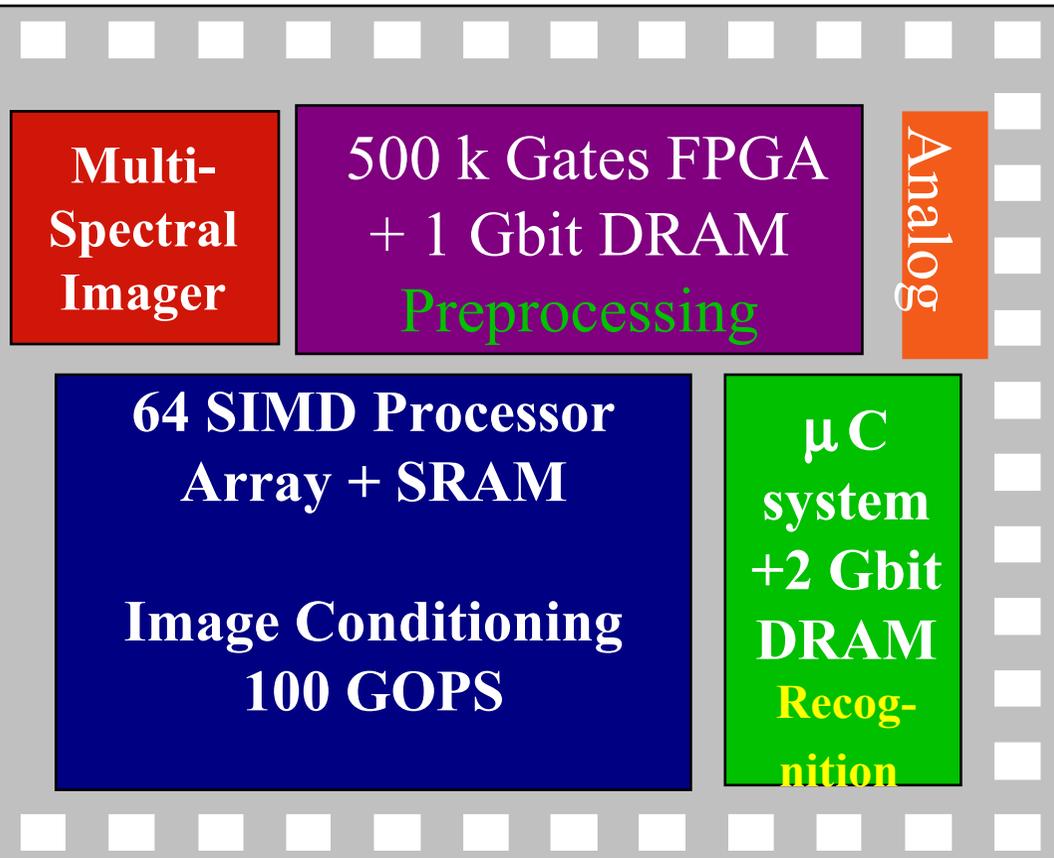
PADDI-2 (UC Berkeley)



- ❑ 1-mm 2-metal CMOS tech
- ❑ 1.2 x 1.2 mm
- ❑ 600k transistores
- ❑ 208-pin PGA
- ❑ $f_{\text{clock}} = 50 \text{ MHz}$
- ❑ $= 3.6 \text{ W @ } 5\text{V}$
- ❑ Módulo Básico: Datapath

Projeto em uma Encruzilhada

System-on-a-Chip



- ❑ Aplicações embarcadas onde custo, desempenho e energia são questões reais!
- ❑ Intensivo- DSP e controle
- ❑ Modo misto
- ❑ Combina módulos programáveis e de aplicação específica
- ❑ SW tem papel crucial

Tratando do Aspecto da Complexity de Projeto

Reuso Arquitetural

Reuso vem em geraçõess

<i>Generation</i>	<i>Reuse element</i>	<i>Status</i>
1^t	Células-padrão	Bem estabelecido
2^d	Núcleos de IP	Em introdução
3^d	Arquitetura	Emergindo
4^h	IC	Primeiras pesquisas

Fonte: Theo Claasen (Philips) – DAC 00

Reuso de Arquitetura

- Plataforma de Sistema em Silício
 - Arquitetura Flexível para hardware e software
 - Componentes específicos (programáveis)
 - Arquitetura de rede
 - Módulos software
 - Regras e guias para projeto de HW e SW
- Tem sido bem sucedidos em PCs
 - Dominância de poucos “players” que especificam e controlam arquitetura
- **Específico de domínio de aplicação** (diferença em restrições)
 - Velocidade (potência de computação)
 - Dissipação
 - Custos
 - Dados em tempo real ou não

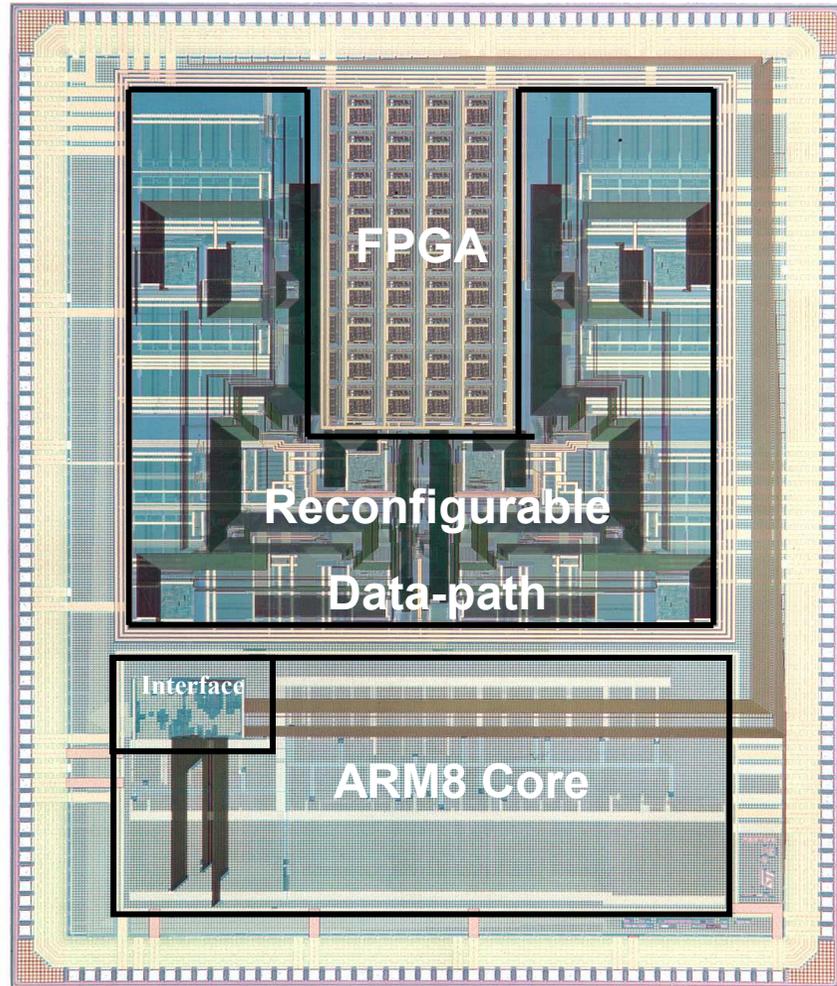
Projeto Baseado em Plataforma

“Sós os consumidores têm liberdade de escolha;

projetistas precisam precisar estar livres da escolha” (Orfali, et al, 1996, p.522)

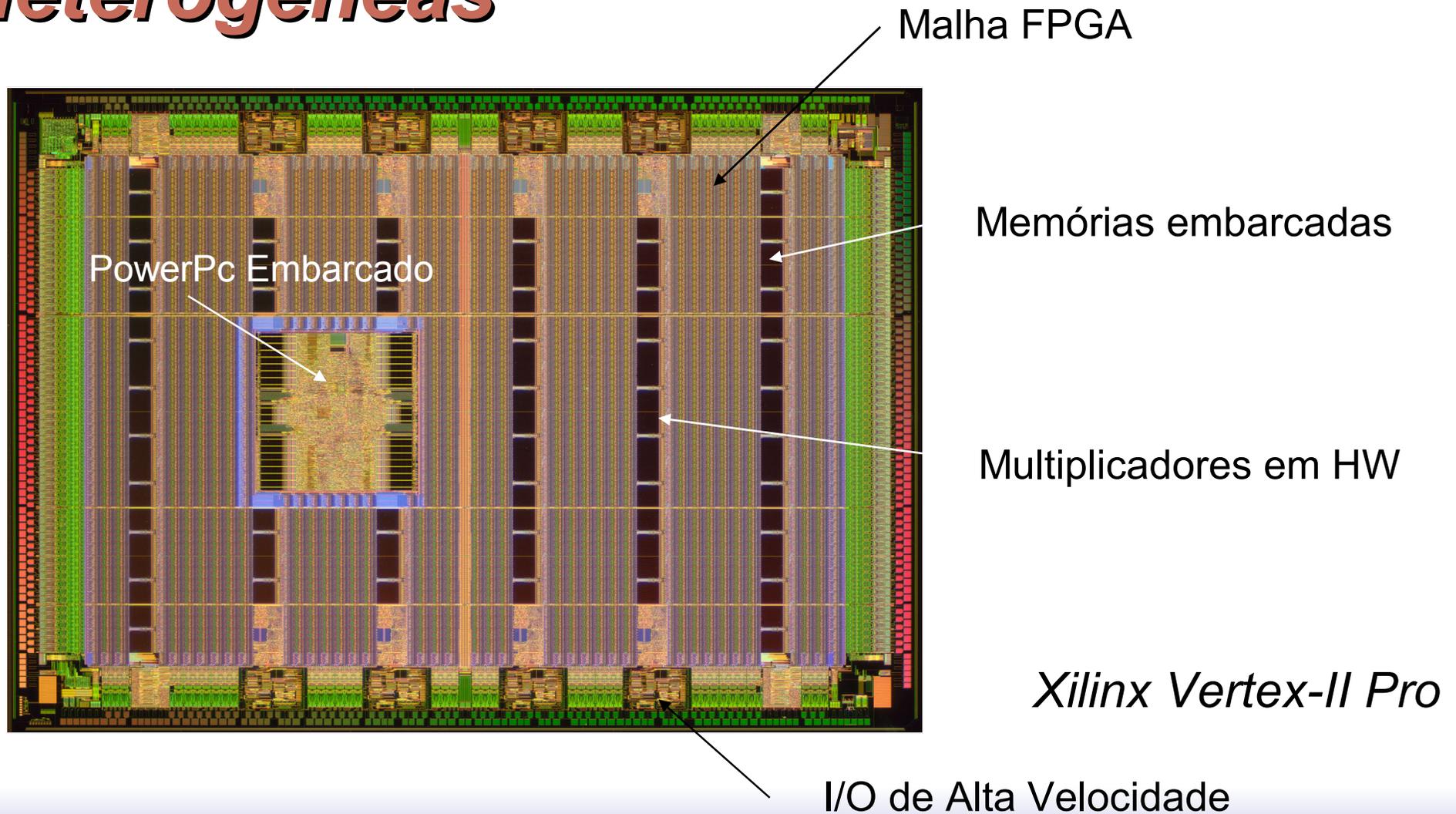
- ❑ Uma plataforma é uma restrição no espaço das possíveis opções de implementação, provendo para o desenvolvedor da aplicação uma abstração bem-definida da tecnologia de base
- ❑ Novas plataformas serão definidas nas fronteiras da arquitetura-micro-arquitetura. Serão baseadas em componentes, e proverão uma gama de opções, das implementações dedicadamente estruturadas aos totalmente programáveis
- ❑ Aspecto chave para tais abordagens é a representação da comunicação no modelo de plataforma

Processador Pleiades de Berkeley



- CMOS 0.25um (6 níveis de metal)
- 5.2mm x 6.7mm
- 1.2 milhões de transistores
- 40 MHz at 1V
- 2 fontes extras: 0.4V, 1.5V
- 1.5~2 mW dissipação de potência

Plataformas Programáveis Heterogêneas



Sumário

- Projeto Digital CMOS está ativo e saudável
- Alguns grandes desafios pela frente causados pela tecnologia Deep Sub-micron
 - Projeto Super GHz
 - Consumo de Potência!!!!
 - Confiabilidade – garantir que funcione

Algumas novas soluções em termos de circuito estão para surgir

- Quem será capaz de projetar nos próximos anos? Algumas grandes mudanças na metodologia de projeto em curso!