

INF01 118

# Técnicas Digitais para Computação

## Lógica Programável

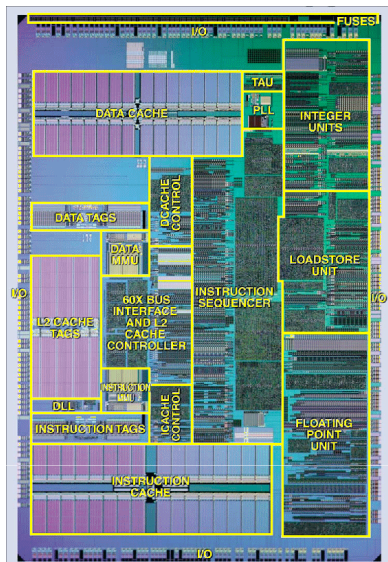


**Aula 17**

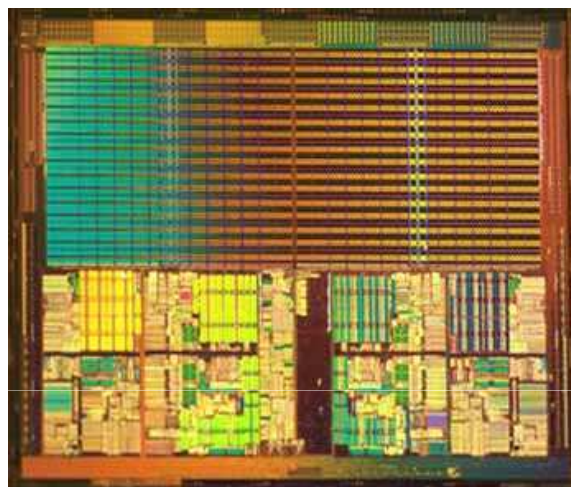
- Programabilidade é noção associada normalmente ao processador de instruções.
- Esta aula introduz conceitos relacionados à programabilidade de hardware, i.e. a maneira de configurar blocos lógicos utilizando a programação de conexões.

## Chips programáveis: Processador(es) em um Chip

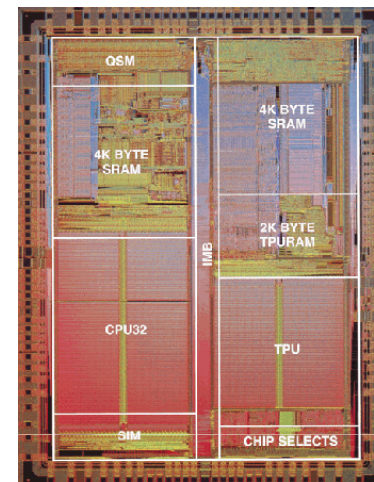
IBM Power PC 750



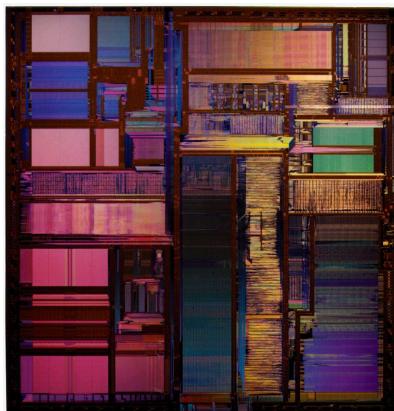
AMD Opteron Dual Core



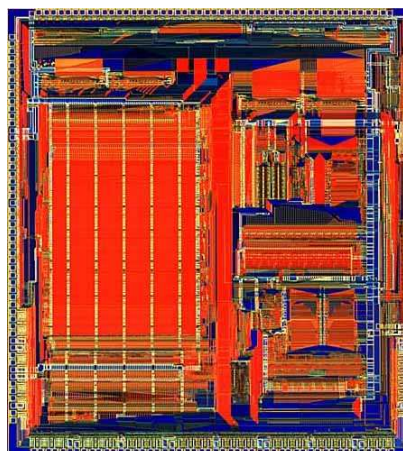
Motorola M68355 Microcontroller



Intel Pentium Processor

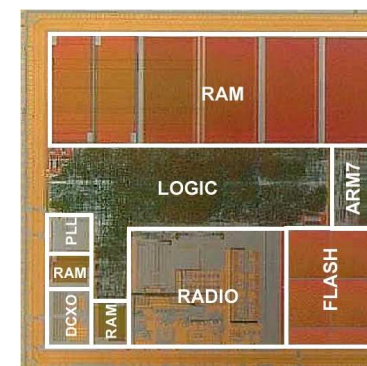


TI DSP 8847



### System-on-Chip

(Bluetooth SoC: Eynde et al., Alcatel, 2001)



Conceitos básicos de Lógica Programável.  
“Programmable Logic Devices” - PLD

## Introdução

Especificação de uma função como  
SDP (Soma de Produtos)

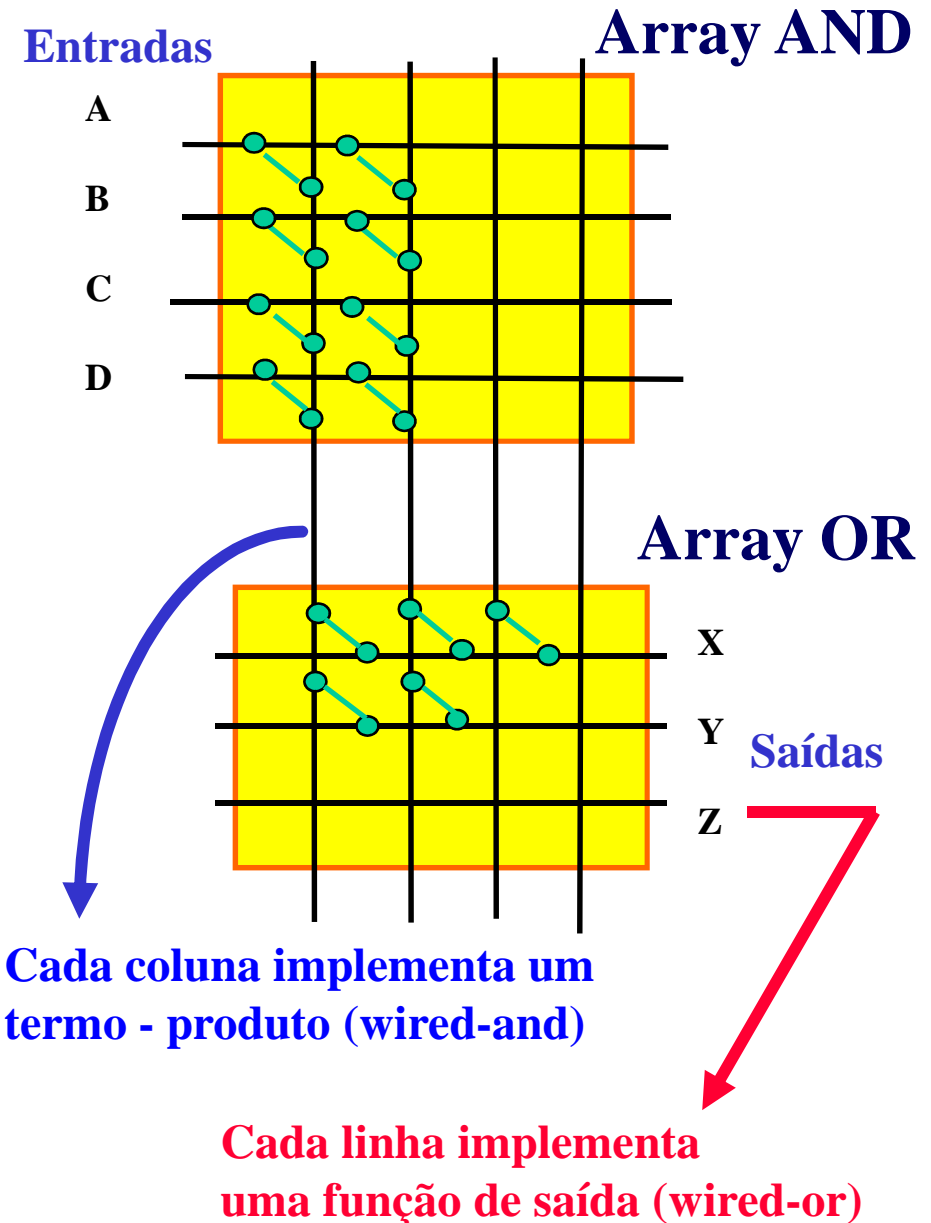


Implementação trivial por um  
circuito AND-OR

PLD's



- dispositivos pré-difundidos
- arrays AND-OR
- “fusíveis” inicialmente intactos  
( fusíveis são na verdade implementados  
por transistores/diodos)
- programação corresponde à queima  
dos “fusíveis”



## **Tipos de PLD's** (Dispositivo Lógico Programável)

- **PROM**
- **PLA**
- **PAL**
- **FPGA**

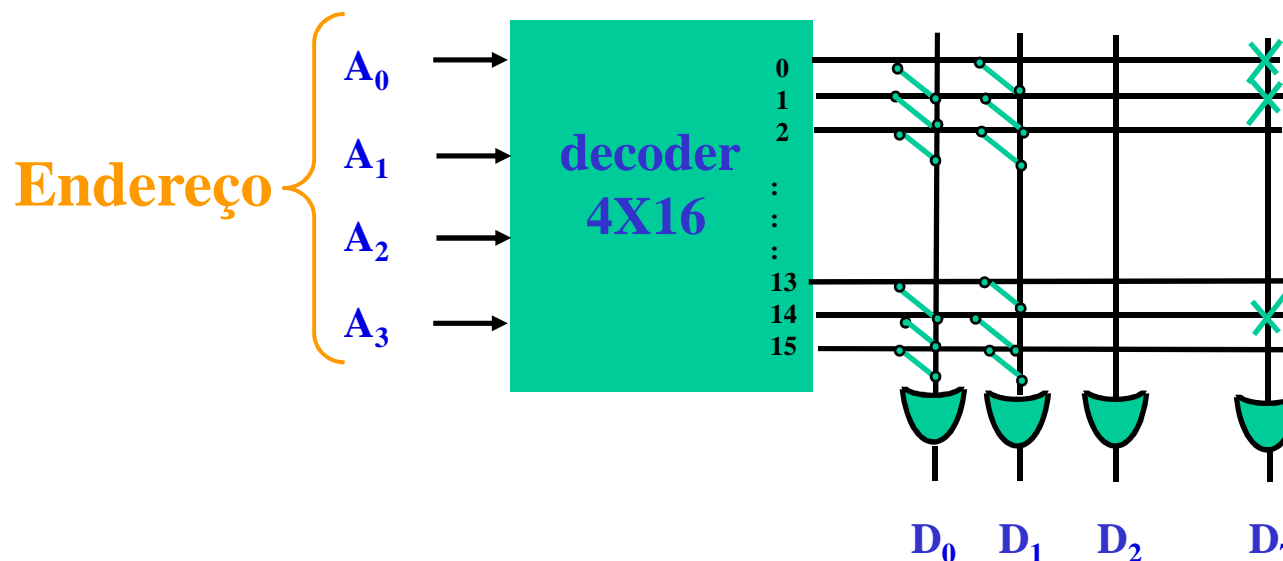
### **Diferem:**

- Na organização dos arrays AND e OR
- Na programabilidade dos arrays ( colocação dos fusíveis ou transistores para realizar conexões)
- Programáveis pelo fabricante (ROM, p. ex.) ou pelo usuário (EEPROM, p.ex.)

## Memória PROM usada como PLD

PROM - Programmable Read-Only-Memory

Representação simbólica



Note:  $D_i$  é uma função combinacional das Entradas de Endereço.

Qualquer função de 4 entradas é implementada em  $D_i$ .

**A (E-E-P-)ROM é um bloco combinacional universal.**

- Notar que se trata de uma **memória**, onde cada célula tem originalmente todos os bits iguais a 1.

- Programação corresponde à queima dos “fusíveis”

- Encarando a PROM como um PLD

- Bits de endereço  $A_i \rightarrow$  variáveis de entrada da função

- Decodificador: efetua a decodificação de todos os mintermos  
cada palavra corresponde então a um mintermo

- Bits de dados de saída  $\rightarrow$  saída de múltiplas funções

- Por exemplo:  $F = \sum m(0,1,14) = m_0 + m_1 + m_{14}$

Implementada por  $D_7$  na figura

Cada mintermo endereça uma linha da memória  $\rightarrow$

Ex. : 16 mintermos

4 variáveis de entrada

4 bits de endereço

- Comparando com a forma geral de um PLD

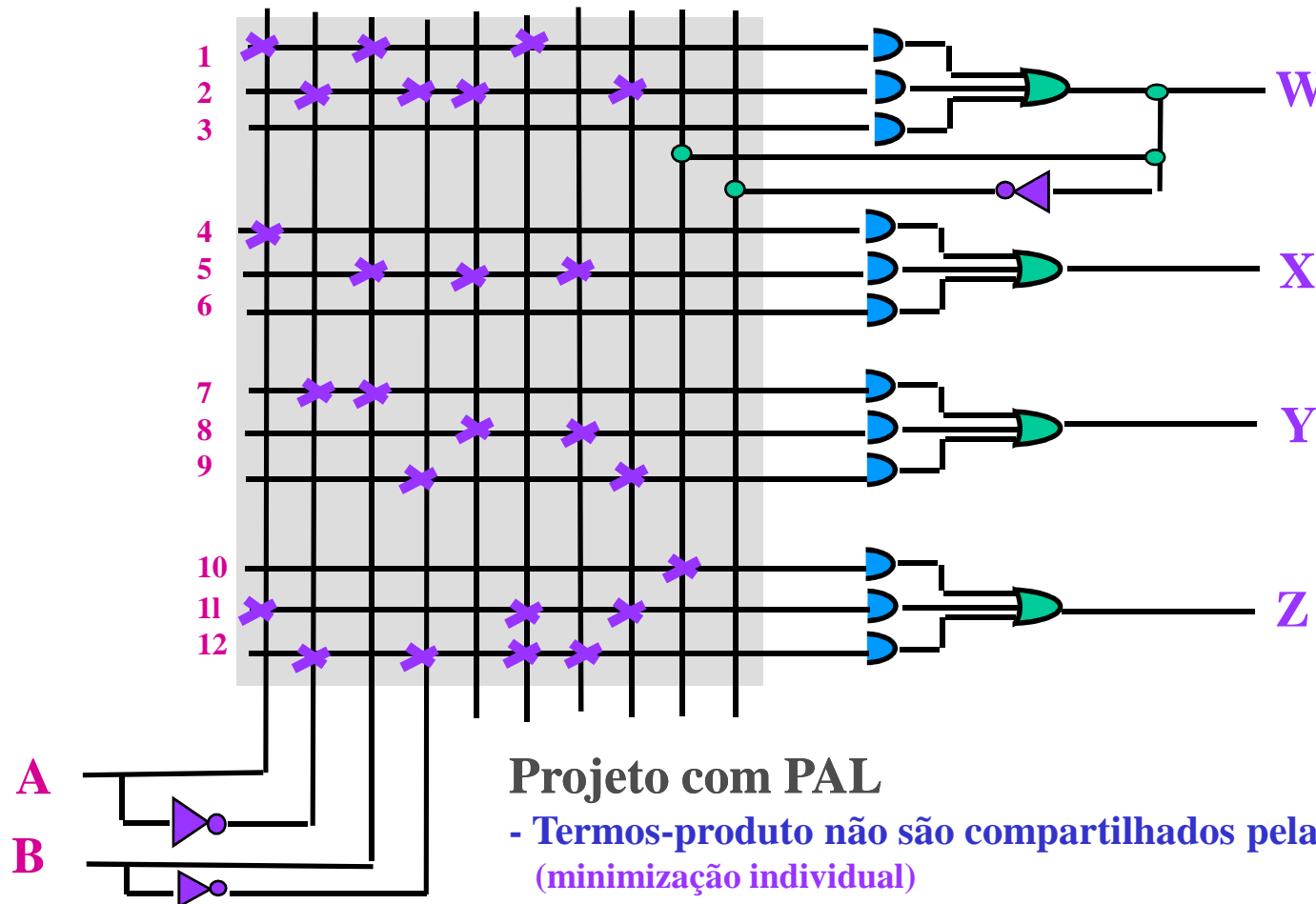
- PROM - decodifica todos os mintermos
- ARRAY AND é fixo (Decodificador) - saída do Decod. são mintermos
- ARRAY OR é programável - (OU LÓGICO) dos mintermos
- Número de funções = número de bits das palavras
- Número de variáveis de entrada = número de bits de endereço



## PAL - Programmable Array Logic

- Características:
- Array AND é programável
  - Array OR é fixo ( cada OR é limitado a  $n$  termos-produto )
  - Feedbacks das saídas para o array **and** permitem funções com mais de  $n$  termos

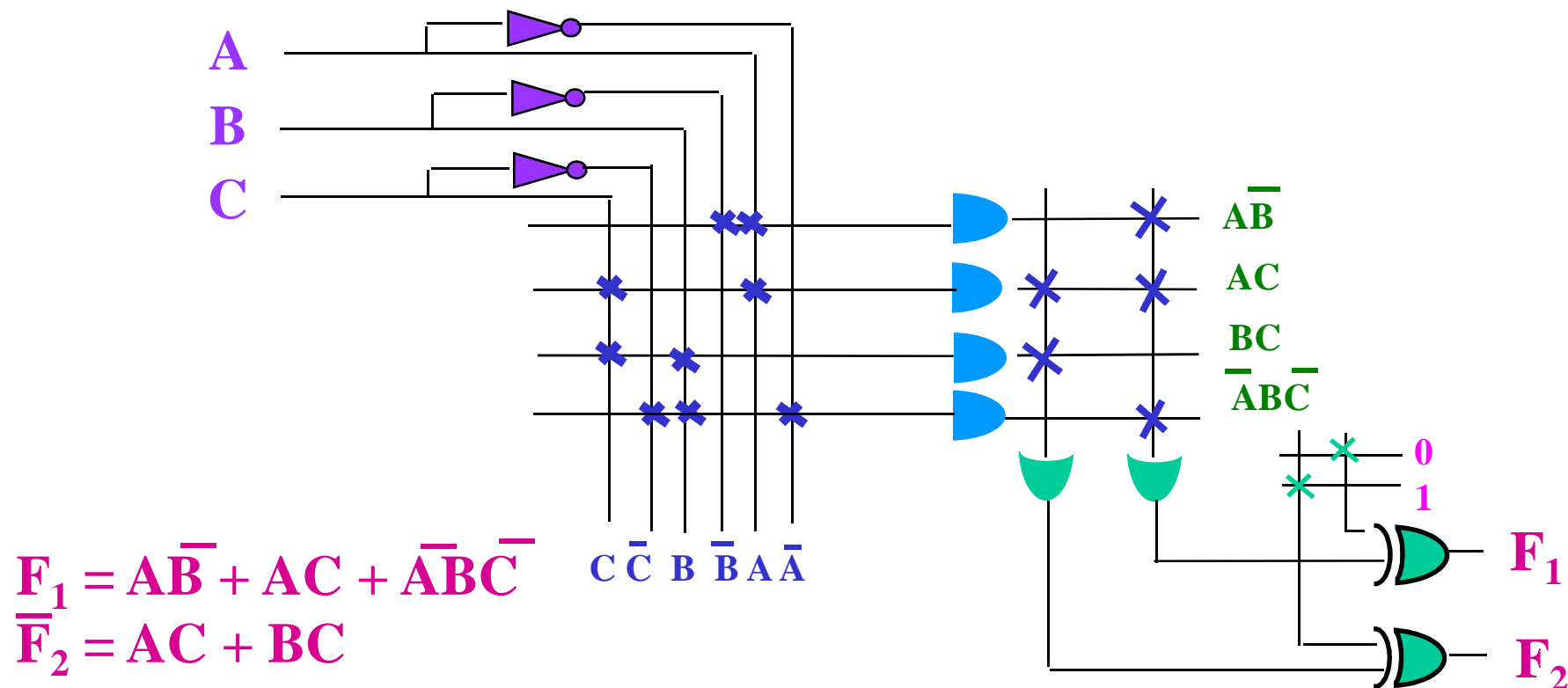
Termos-produto  $A \bar{A} B \bar{B} C \bar{C} D \bar{D} W \bar{W}$



## PLA - Programmable Logic Array

### Características

- Arrays AND e OR são programáveis
- Para  $n$  variáveis de entrada, o array AND tem  $2n$  linhas (variável e variável complementada)
- São decodificados apenas os termos-produto necessários



## Projeto com PLA

- Deve-se minimizar todas as funções de saída simultaneamente, procurando-se o menor número de termos-produto comuns a todas as funções

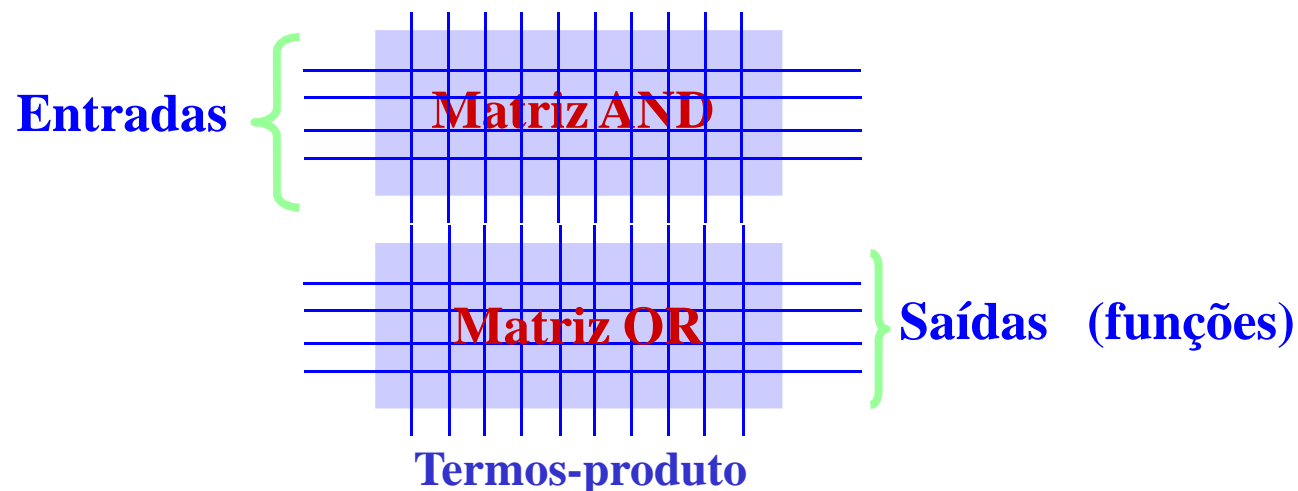
- Deve-se tentar solução ótima tanto com  $F$  como com  $\bar{F}$ , procurando:

- menor número de termos
- termos comuns a outras funções

No exemplo:  $AC$  é comum a  $F1$  e  $\bar{F}2$

- **Tamanho do PLA**

- Nro. entradas = número de variáveis da função
- Nro. linhas do array **AND** = nro. máximo de termos-produto no conjunto das funções
- Nro. saídas = número de funções simultâneas



- **Tipos de PLA's:**

- Programável por “máscaras”, na fábrica de chips (silicon foundry).

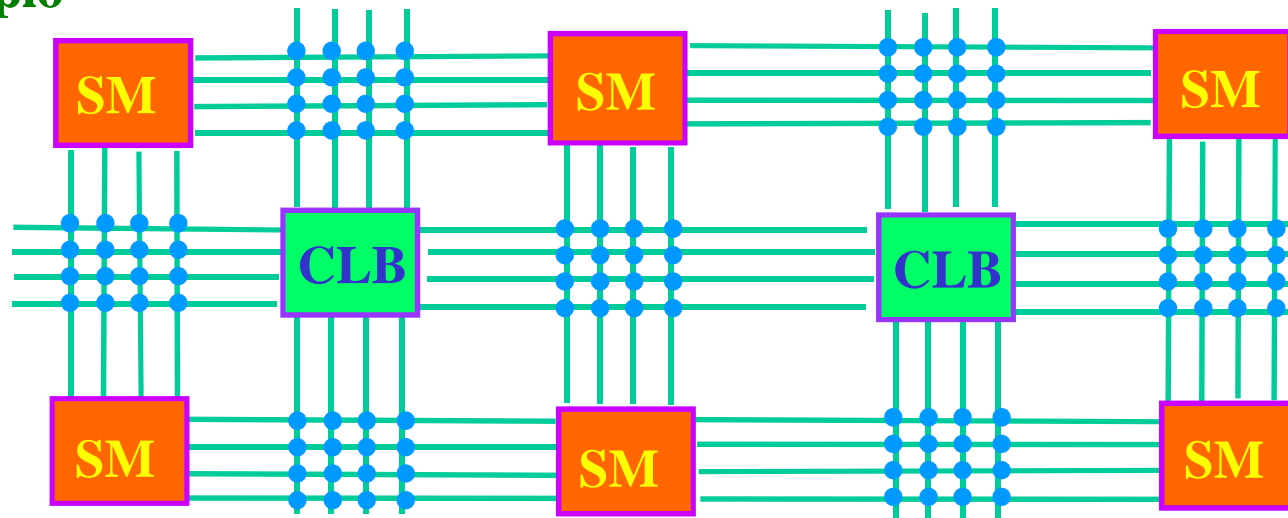
•

## FPGA - Field Programmable Gate Array

### Características

- Reprogramável **n** vezes
- Array de blocos lógicos (função definida por seleção)
- Blocos lógicos e conexões são programáveis

### Exemplo

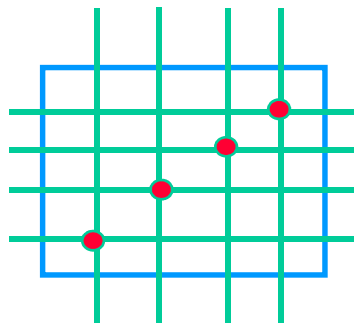


**PLB** { Blocos Lógicos Programáveis (Programmable Logic Blocks)  
**CLB** { Configurable Logic Blocks

CLB

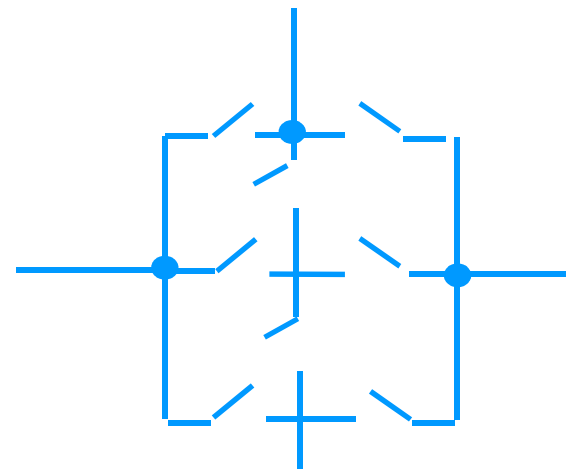


SM

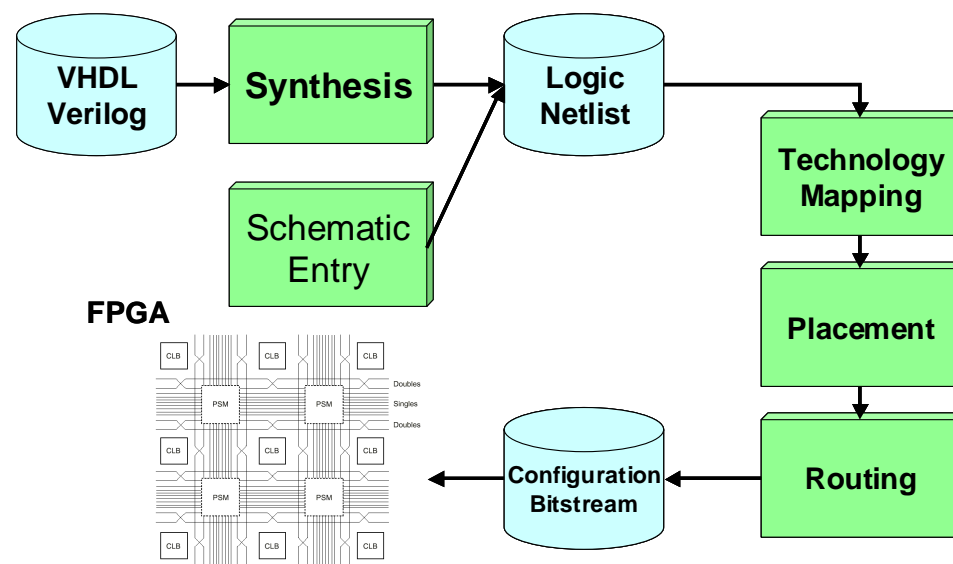
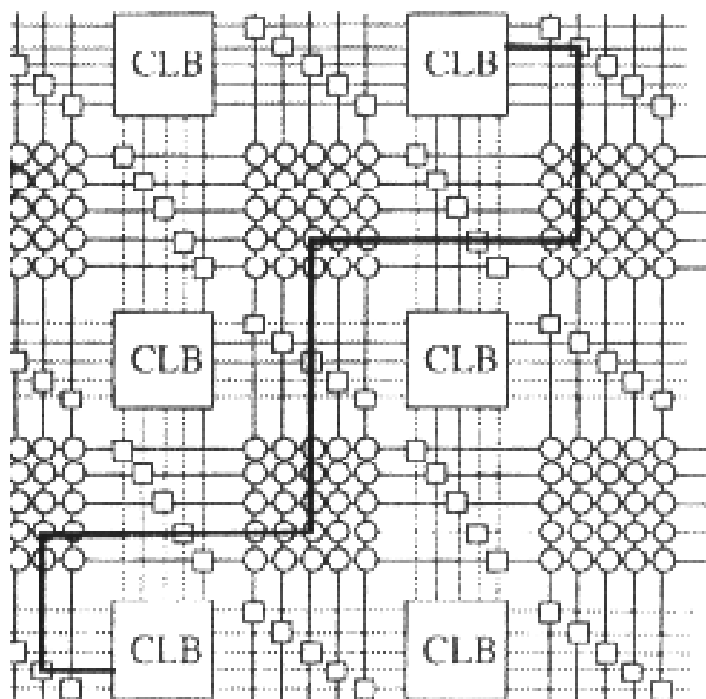


SM - Switching Matrix

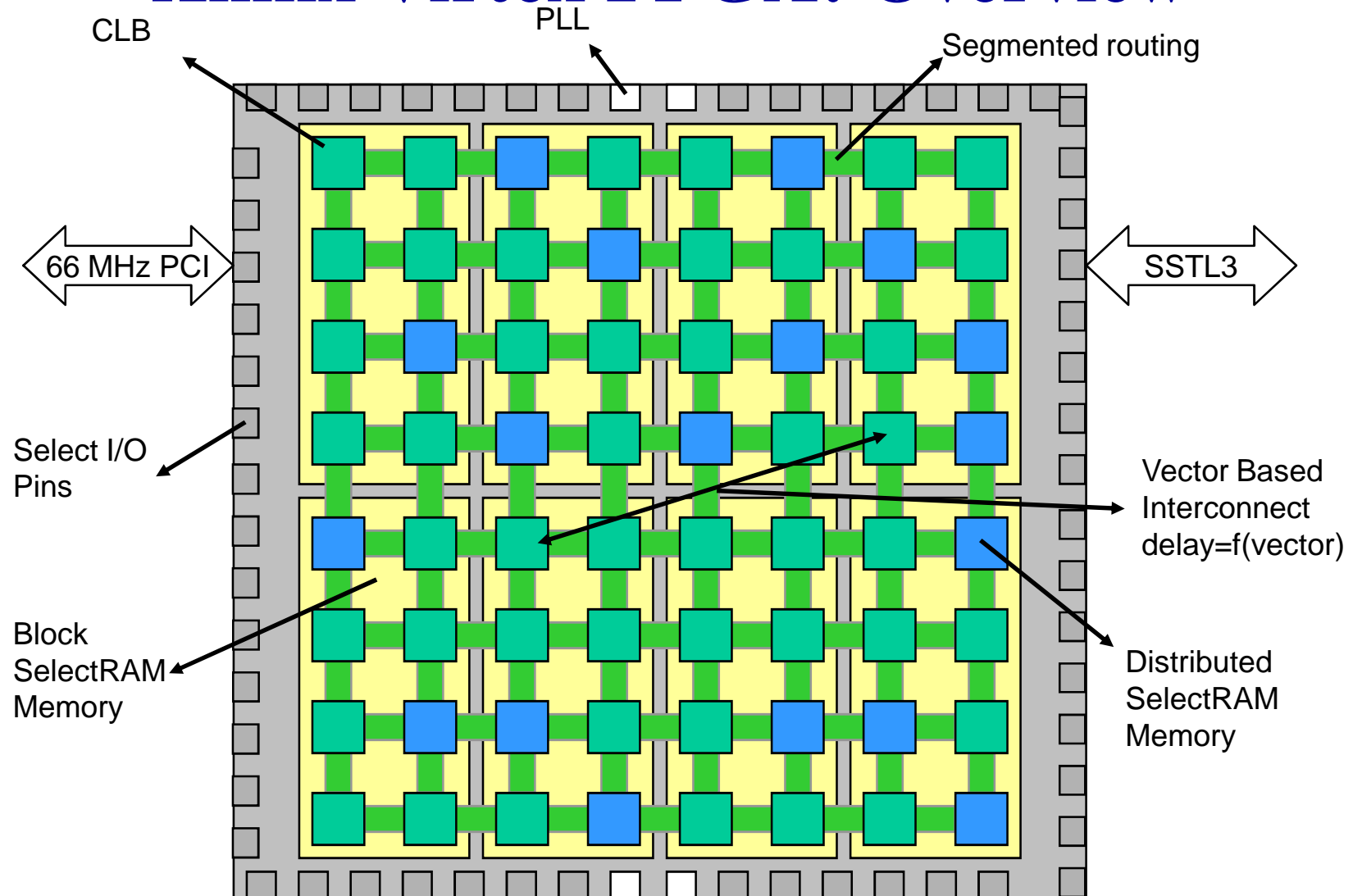
Ponto de Interconexão.  
Programada por  
transistores MOS



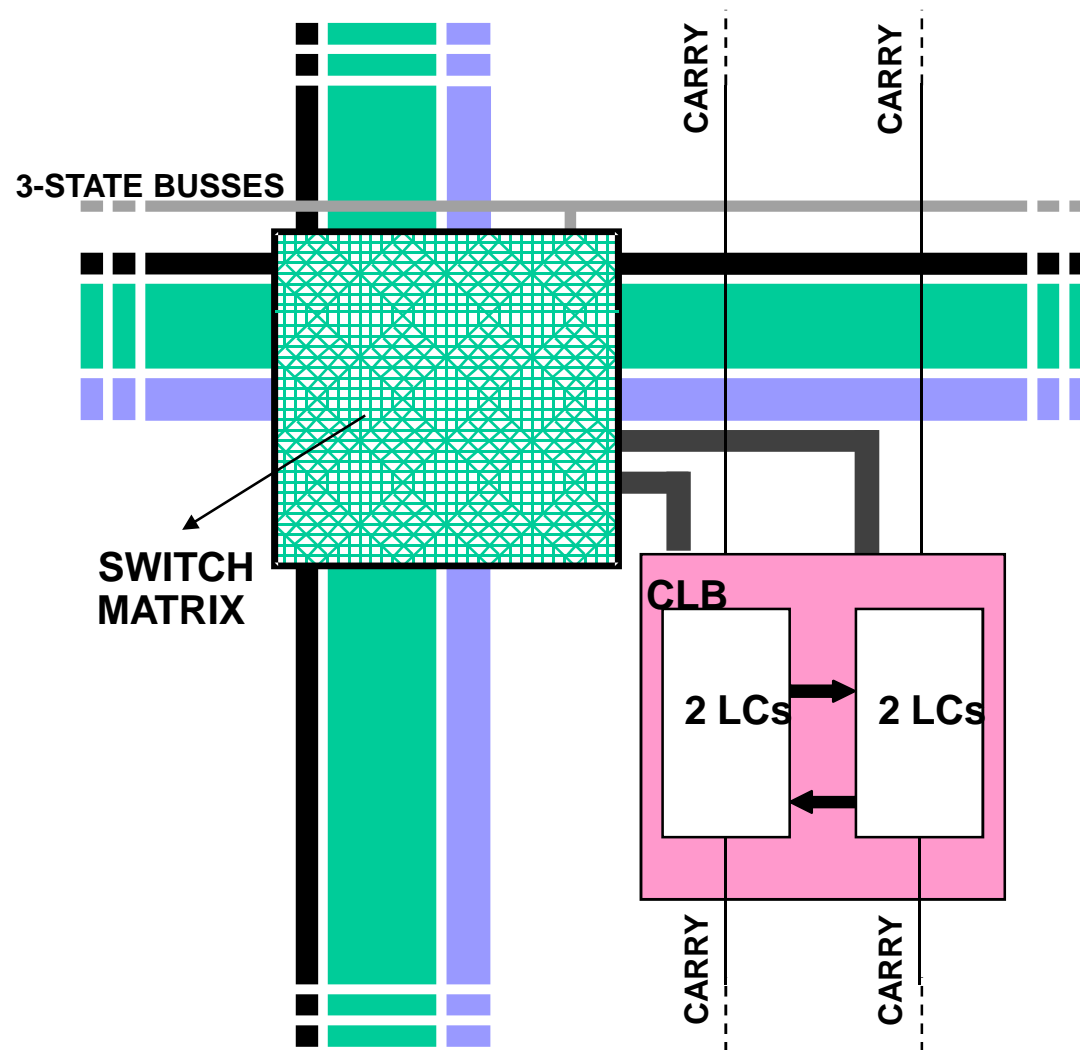
## FPGA



# Xilinx Virtex FPGA: Overview



- Conexão (fio) local dentro das CLBs
- Interconexão entre CLBs (global)
- 1 CLB tem 4 células lógicas programáveis.
- Interconexões rápidas e longas
  - 8ns entre blocos com 250.00 portas
- Previsão da frequência de acordo com o comprimento do fio.



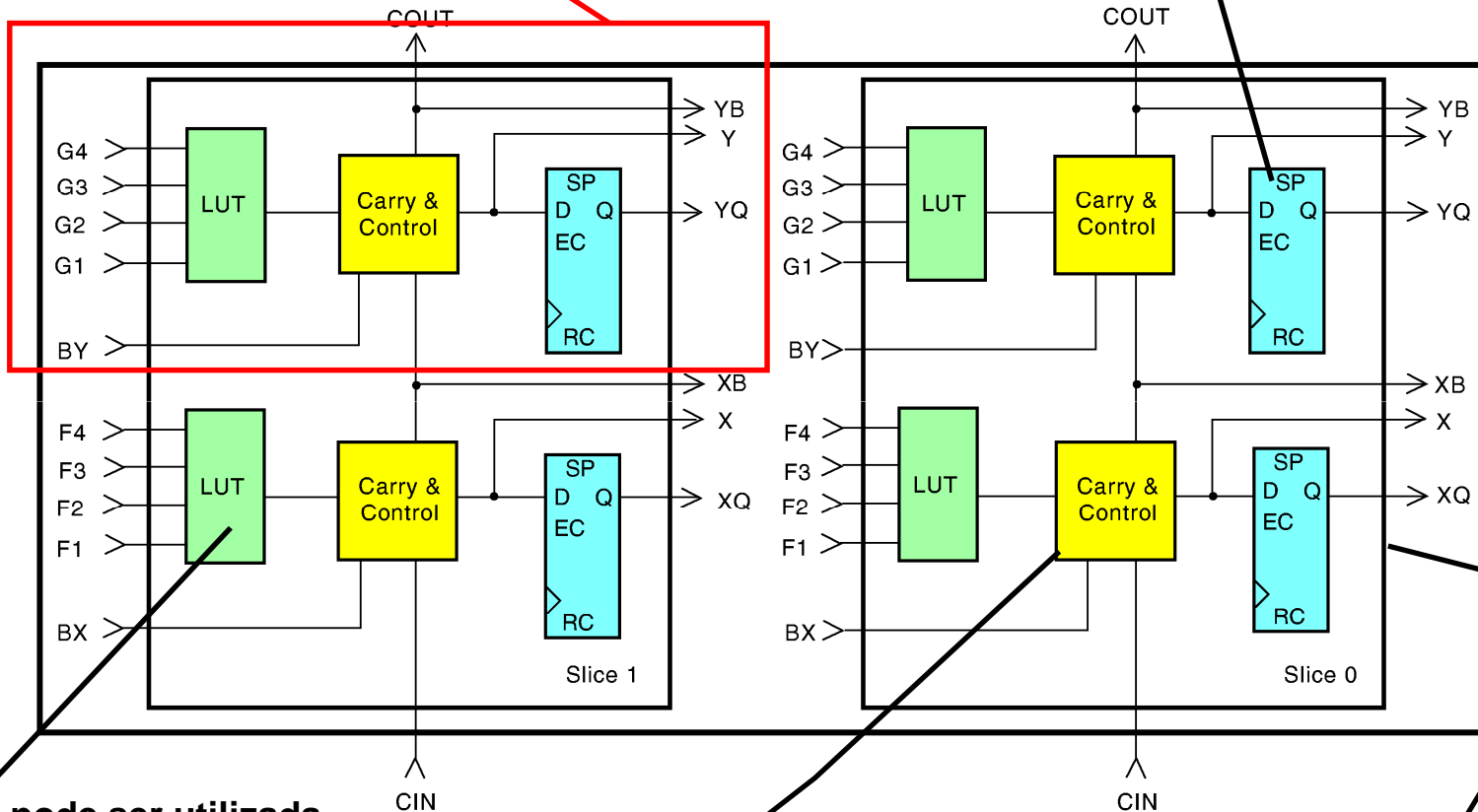
## Xilinx Virtex FPGA: Roteamento Segmentado



## FPGA

Logic Cell  
LUT com 4 entradas  
Carry-Logic e Memória

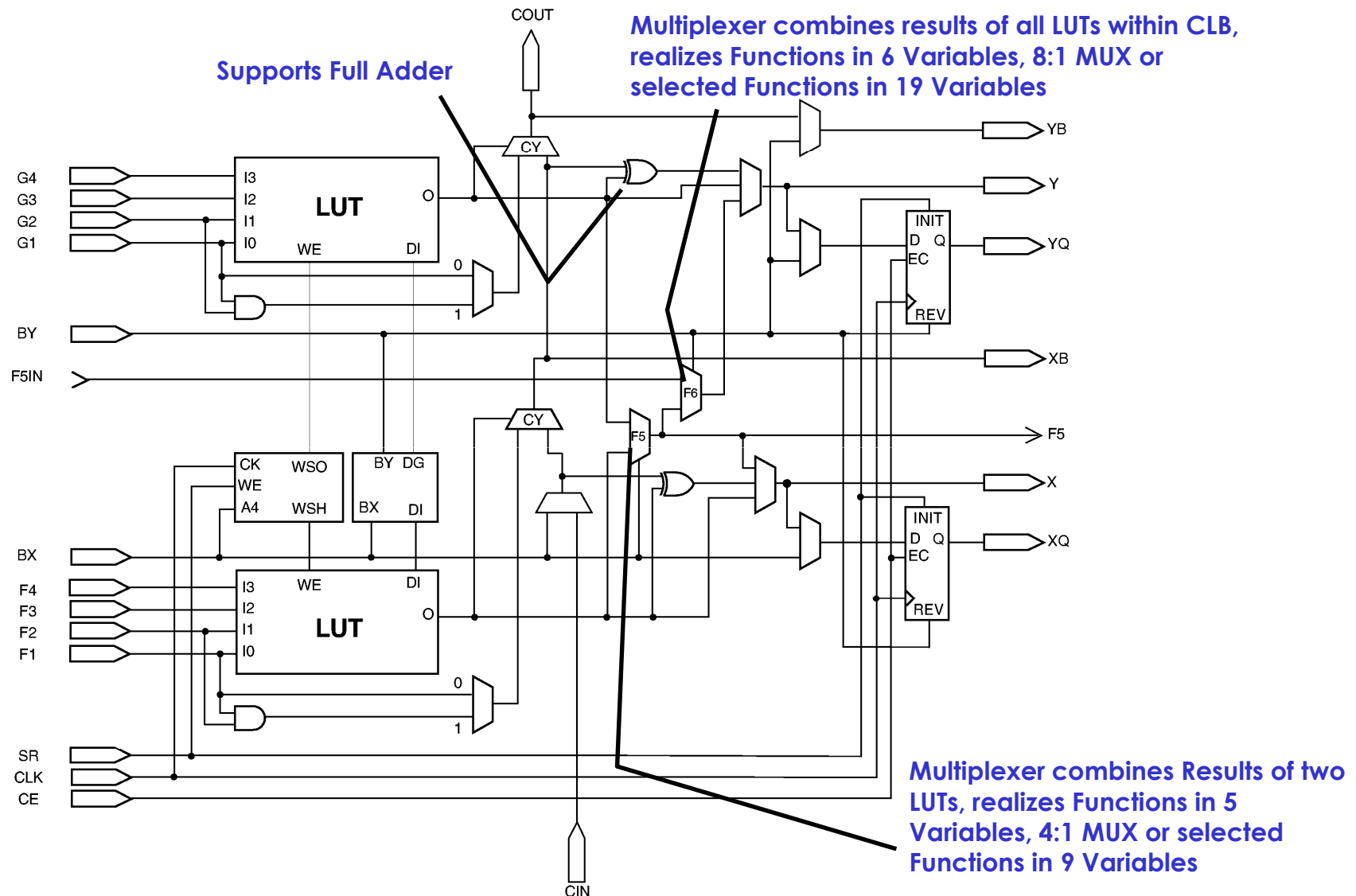
Memory as Edge-triggered D-FlipFlops  
or Level-sensitive Latches



Uma LUT pode ser utilizada  
como uma RAM de 16 bits.  
2 bytes / LUT.

CLB divided in 4 Logic Cells = 2 Slices per CLB  
Contains additional Logic, to realize Function  
Generators with 5 or 6 Inputs

## Xilinx Virtex FPGA: CLB Structure

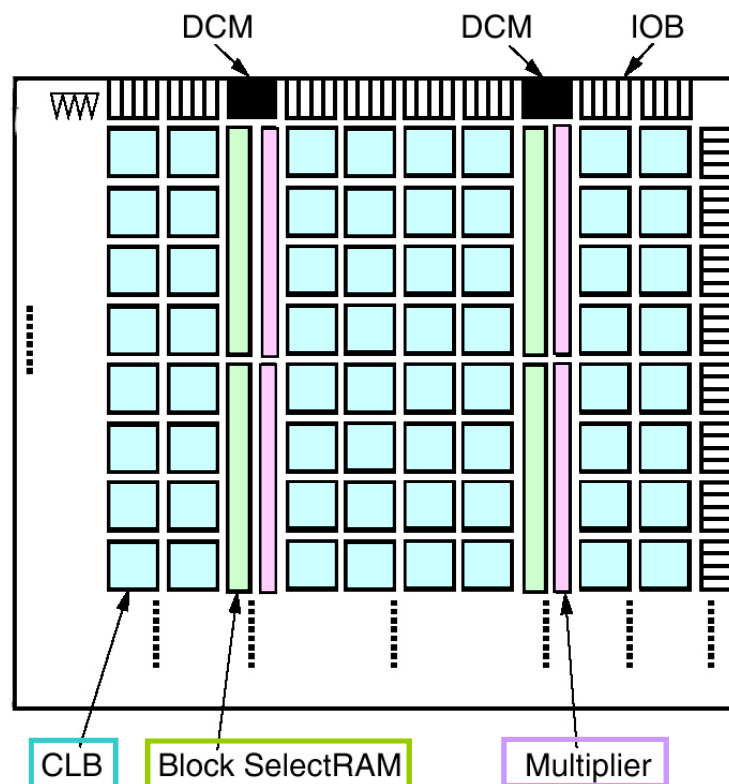


## Xilinx Virtex FPGA: CLB Slice Structure

# Xilinx Virtex-II Pro FPGA: Architecture

- Até 3 Mbits de Dual-Port™ RAM em blocos de 18-Kbit.  
**SelectRAM Resources**
- Até 12 DCM (Digital Clock Manager) Módulos
- Multiplicadores 18-bit x 18-bit
- Fast look-ahead Carry Logic Chains
- 0.15µm / 0.12µm CMOS 8-Camadas de metalização para interconexão, distribuição de Clock e VDD/GND.

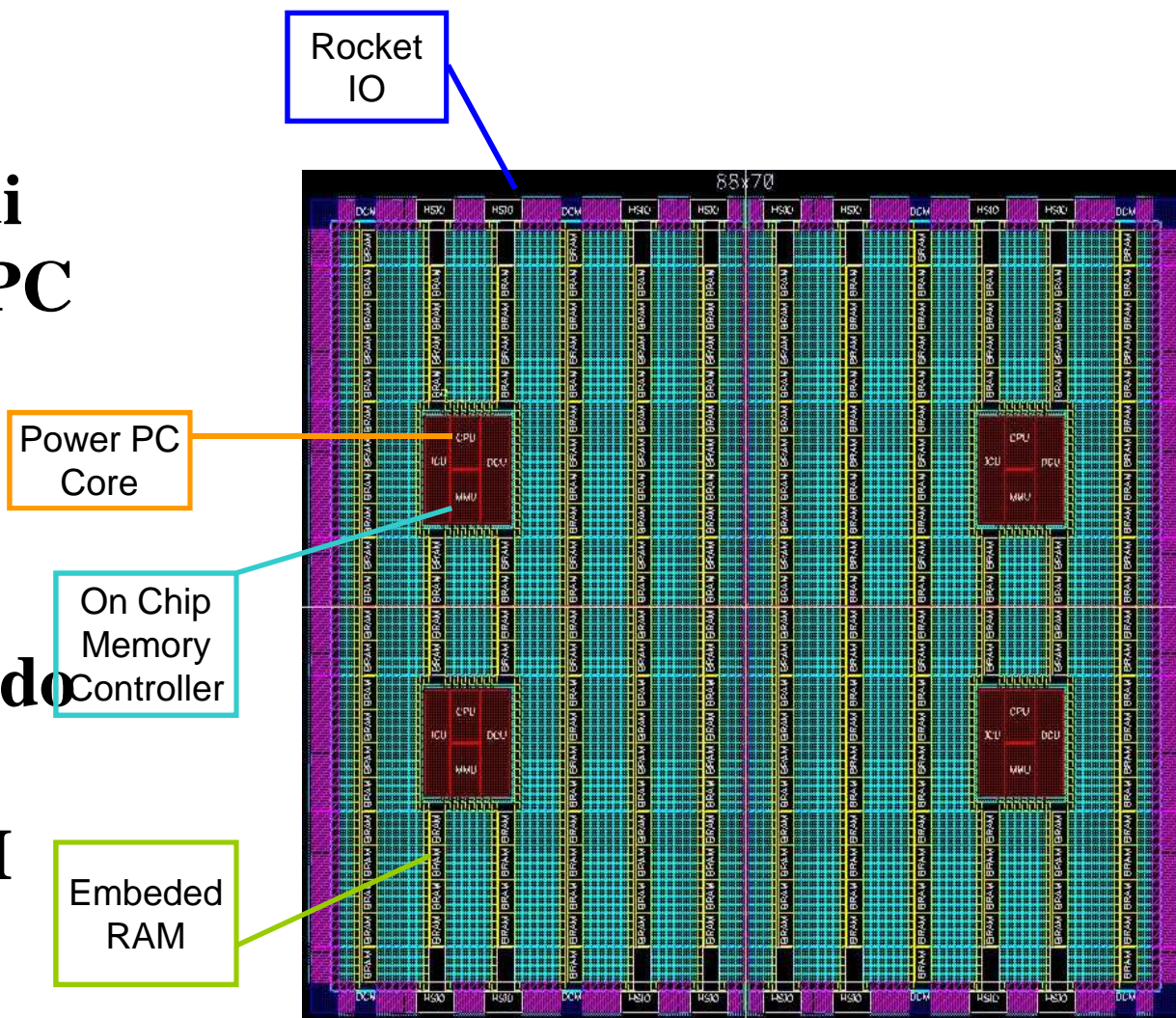
	Fastest DSP	Virtex-II
8x8 MAC	4.8 Billion MAC/s	0.5 Tera MAC/s
256-FIR	9.3 MSPS	180 MSPS
1024-FFT	10 µs	1 µs



# Architecture

- Além da matriz de FPGA, inclui até 4 PowerPC® 405 RISC CPU cores

- Chip layout do FPGA tipo Virtex-II



Source: Ivo Bolsens, Xilinx

## Discussão crítica do projeto de blocos combinacionais

- Alternativas de projeto de circuito digitais

Componentes discretos (de prateleira)

Circuito integrado dedicado

- Estilos de projeto de blocos combinacionais

Lógica programável

Lógica hardwired

- Decodificadores/multiplexadores
- Lógica em 2 níveis
- Lógica aleatória multi-nível

- Fatores a considerar na escolha

Eficiência da solução: área, timing, consumo

Custo de produção

Custo de projeto

Volume de produção

Número de entradas/saídas (encapsulamento)

**Somente em função do problema a ser resolvido é que podemos definir a melhor solução.**