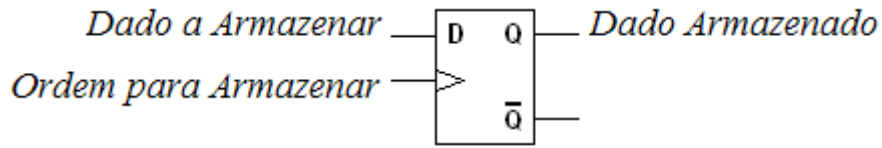
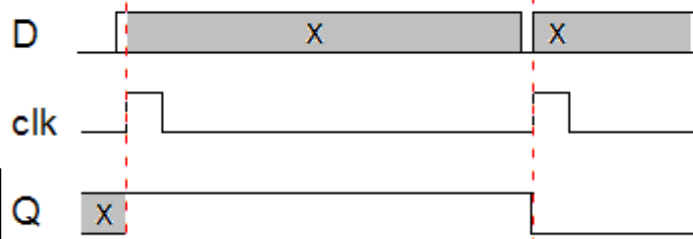


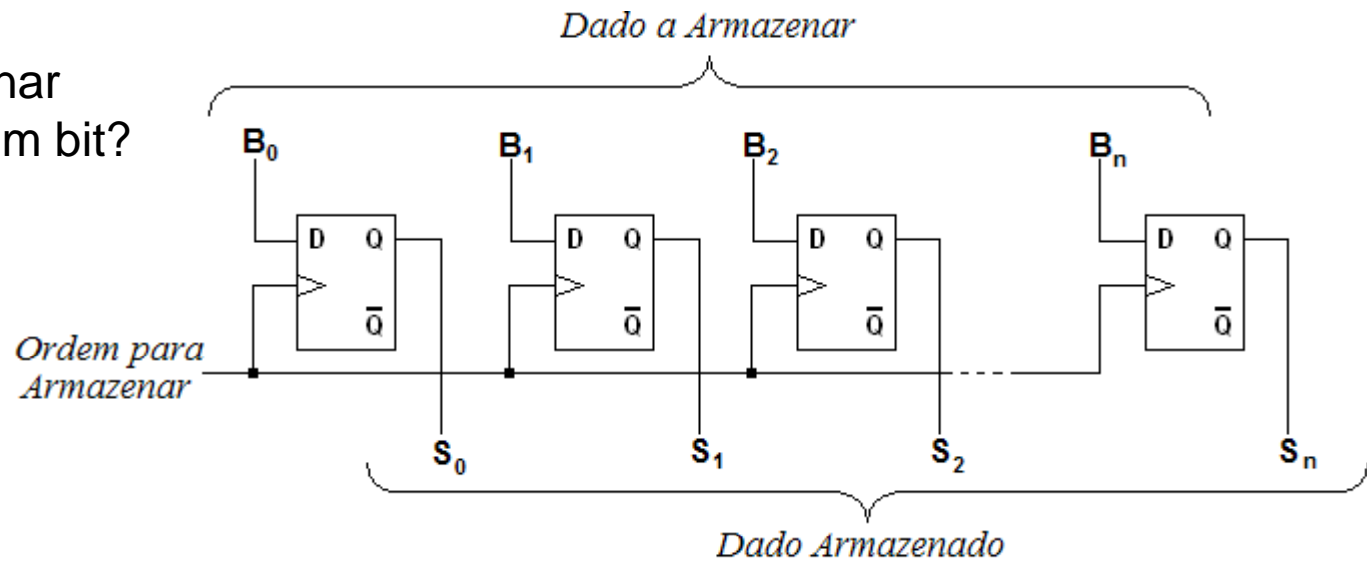
MEMÓRIAS



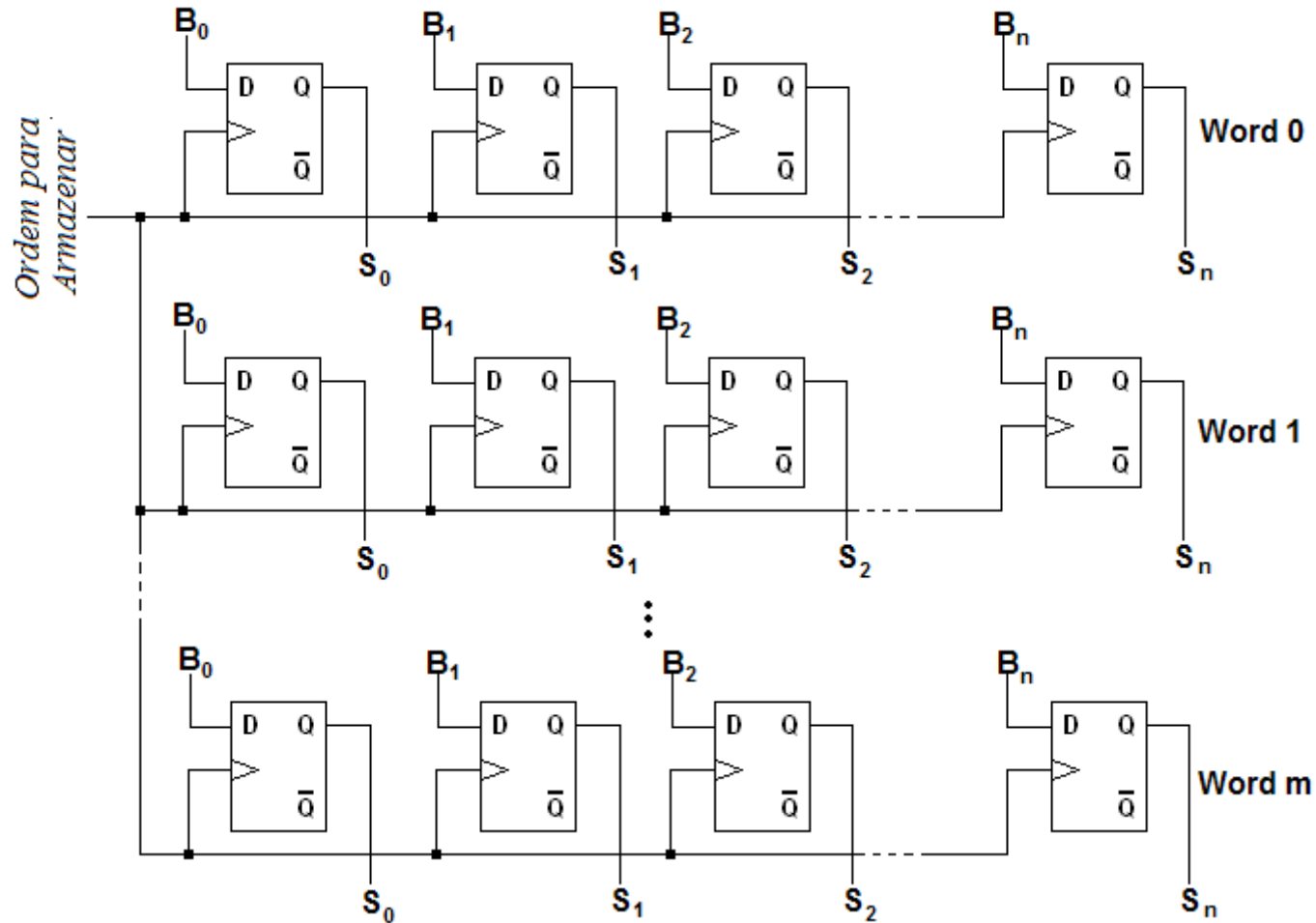
Até ao momento ...
MEMÓRIA = FLIP-FLOP



como armazenar
mais do que um bit?

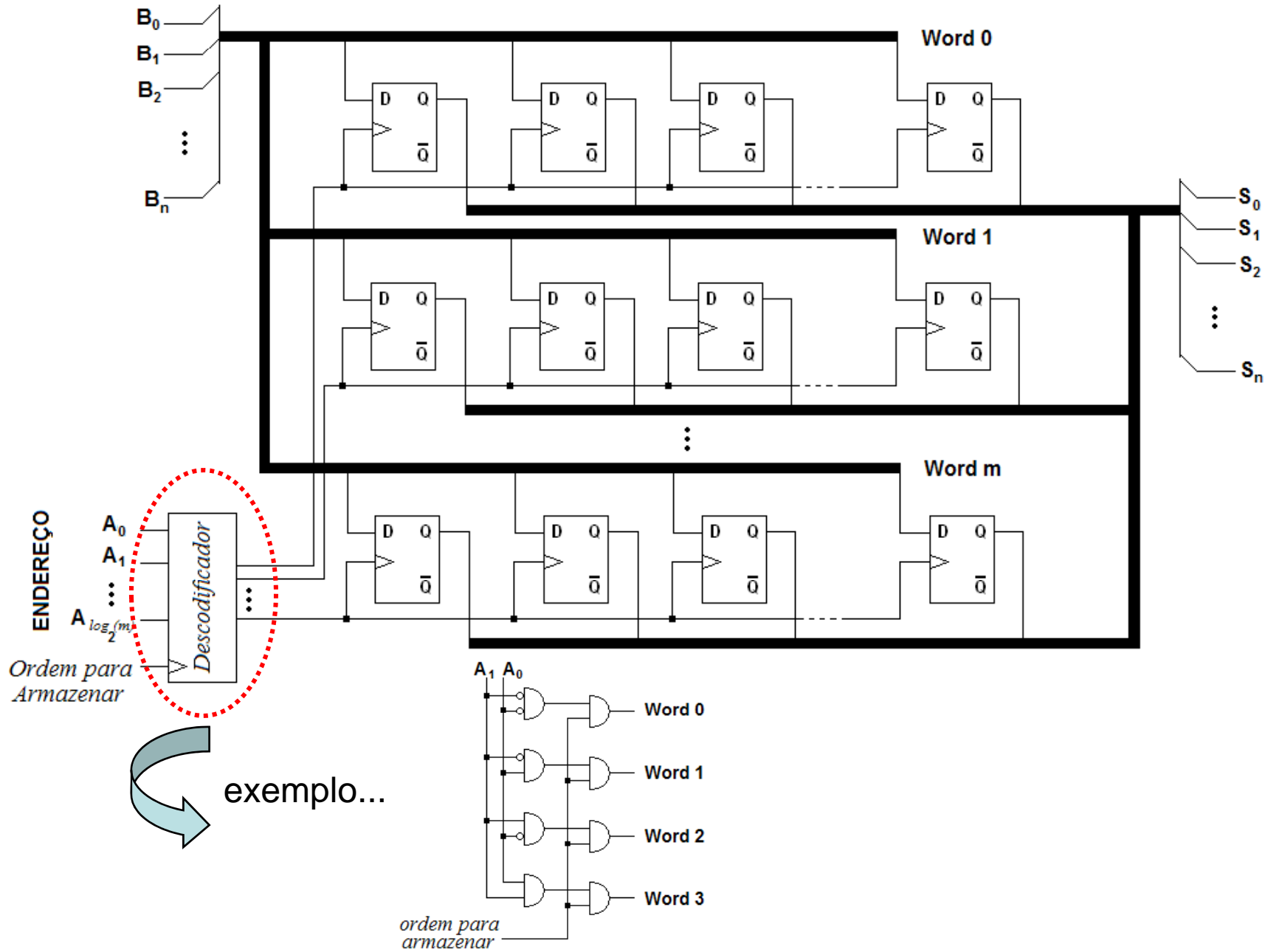


como armazenar mais do que uma palavra?



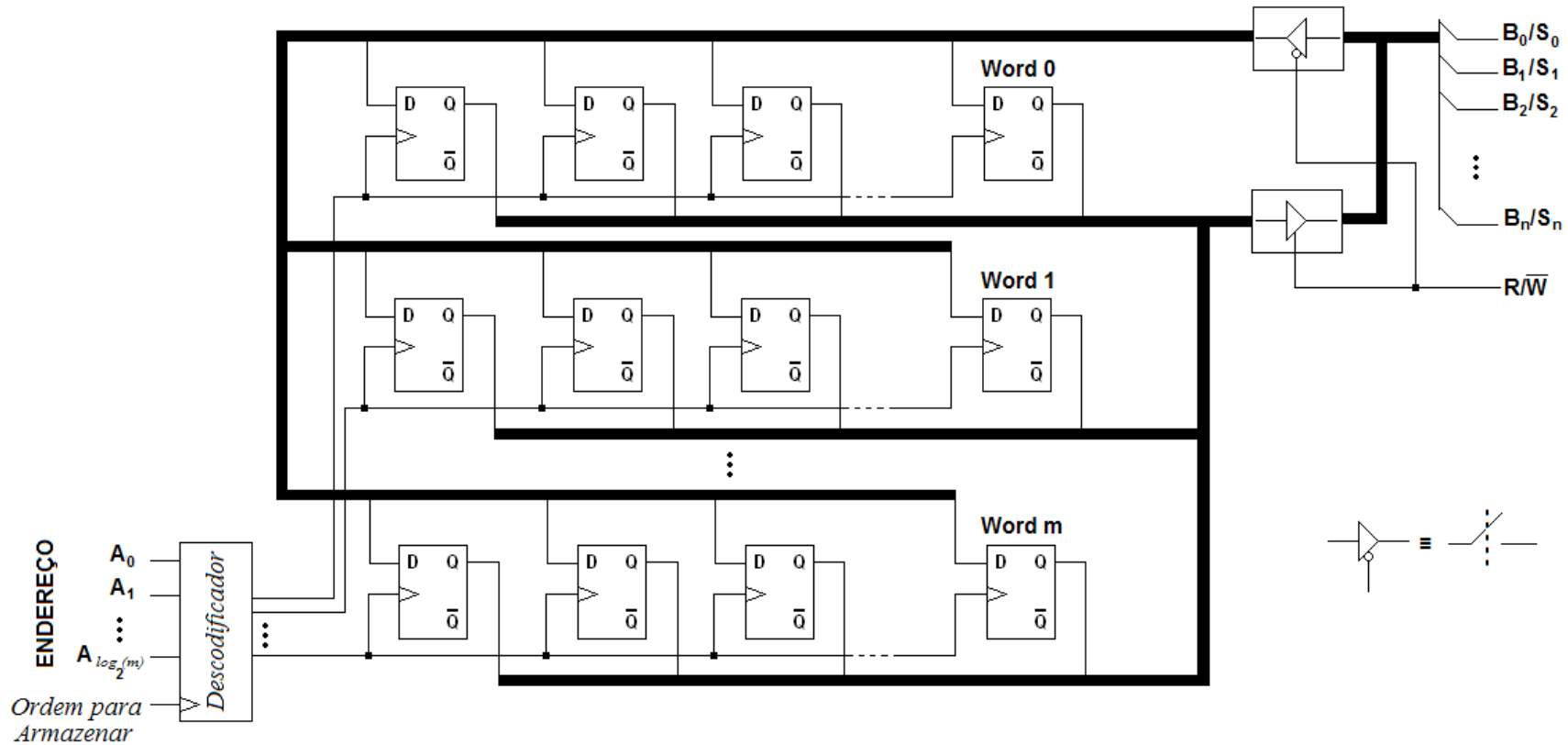
Quantos **pinos** deveria ter um CI capaz de armazenar 1024 Bytes ($W=1B$)

Conceito de linhas de ENDEREÇO...



exemplo...

Para reduzir o número de pinos do CI frequentemente os terminais de I/O são comuns...



- Será possível, num dado instante, alterar o bit k da palavra w?
- Esta é uma memória de leitura/escrita (RWM) mas não é uma RAM!

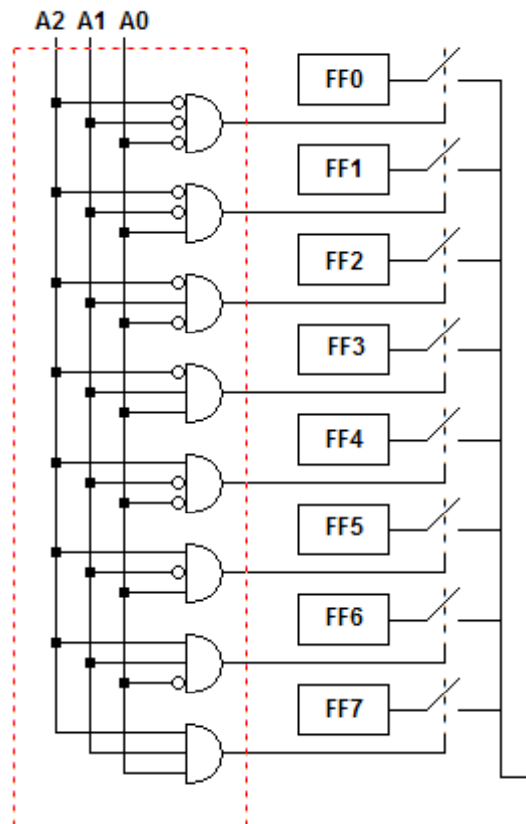
Memória **RAM** (Random Access Memeory)

- Memória de Acesso Aleatório
- Nome atribuído a RWM com a capacidade de indexar célula a célula

ORGANIZAÇÃO INTERNA:

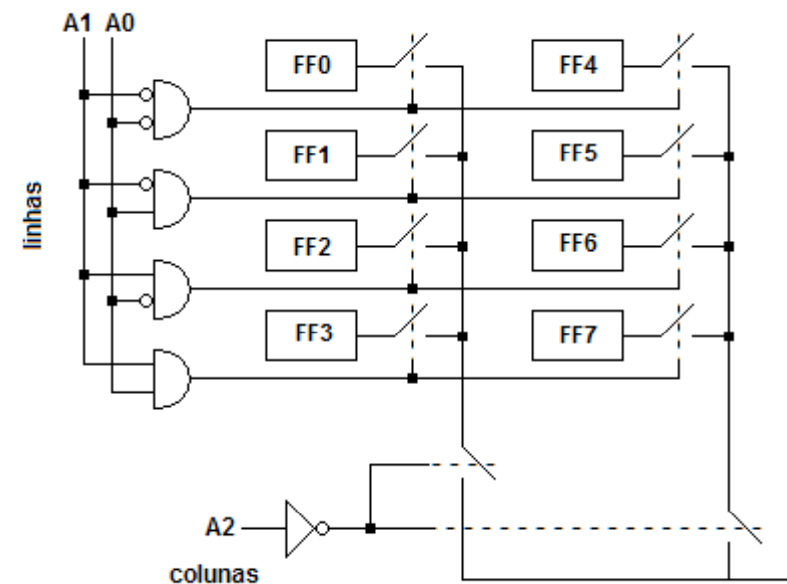
1-D

8X1 BIT



2-D

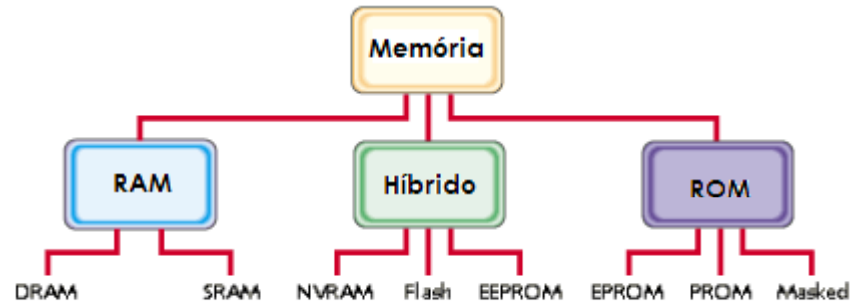
8X1 BIT



A versão 2-D requer menos “hardware”

- Para além das memórias RAM existem muitos outros tipos de memórias...

* Adaptado de
Michael Barr's Embedded Systems Glossary



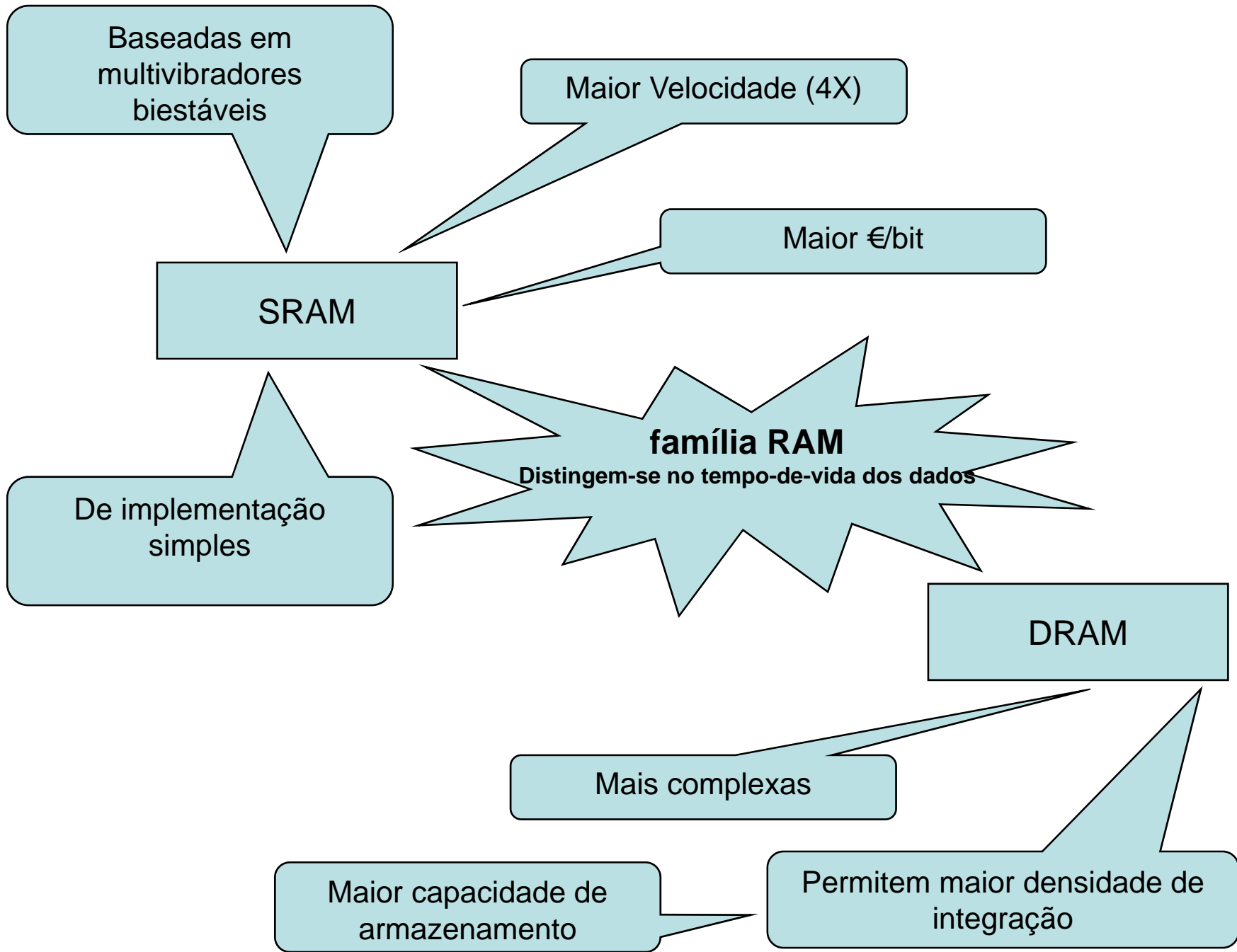
MEMÓRIAS RAM

É possível ler/escrever em qualquer célula de memória

Requer alimentação constante de contrário a informação é perdida

RAM
(Random Access Memory)

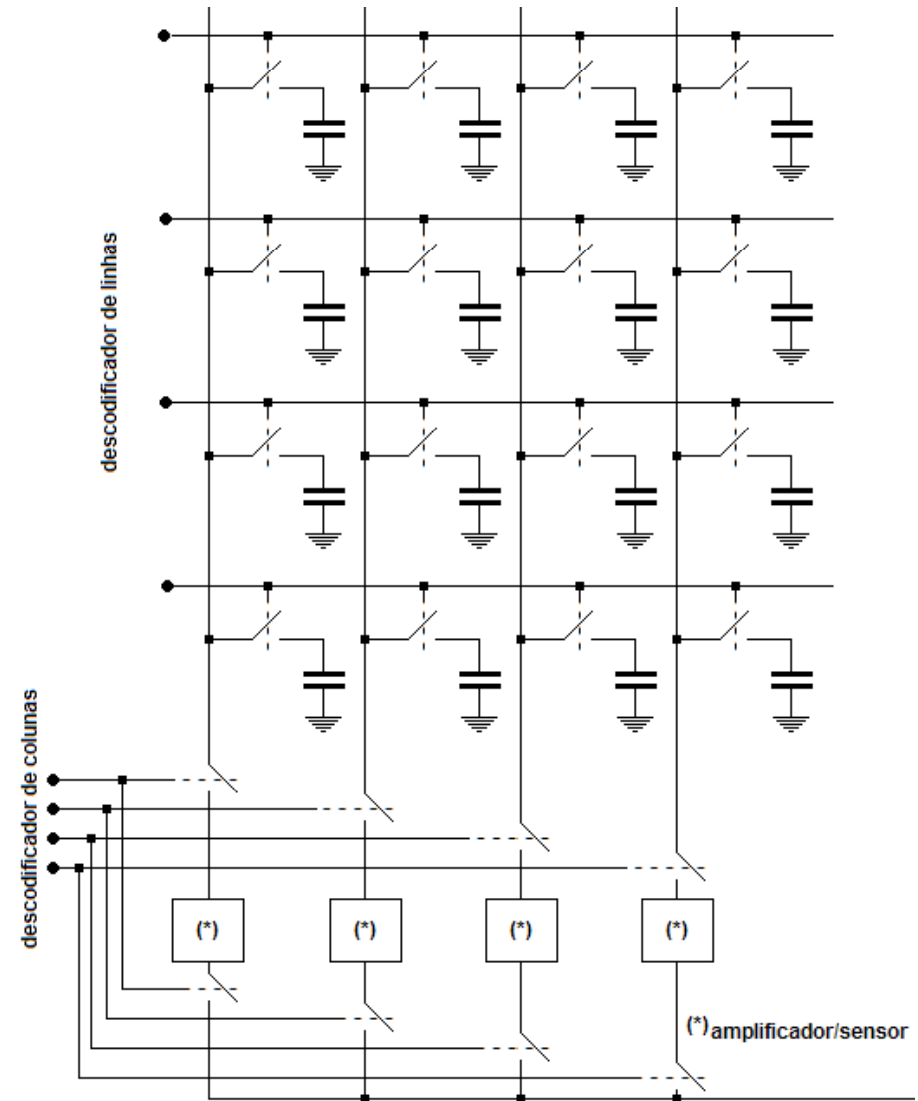
Utilizada em arquitecturas de computadores para o armazenamento temporário de variáveis



MEMÓRIAS DRAM

- Armazena os bits individuais em “condensadores”
- A carga nos condensadores não é mantida indefinidamente
- Requer um circuito de “refresh”

cada célula é visitada periodicamente
e o seu valor lógico é reposto



MEMÓRIAS ROM

O conteúdo das células é fixo

Ao contrário das RAM não requer fornecimento constante de energia

É também uma memória de acesso aleatório

ROM
(Read Only Memory)

Disponível sob diversas tecnologias:
PROM, EPROM, etc.

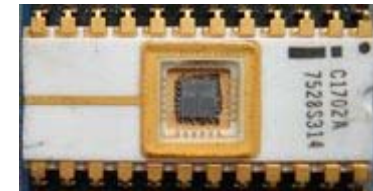
Utilizadas para armazenar o conteúdo da BIOS em PC's

-Primeiro tipo de ROM programada no fabrico (masked ROM's)

PROM (OTP)
Adquirida 'virgem' e programada, apenas uma vez, pelo utilizador.

Processo de desenvolvimento de "firmware"

EPROM
Passível de ser reprogramada:
Escrever – sinal eléctrico
Apagar – U.V.



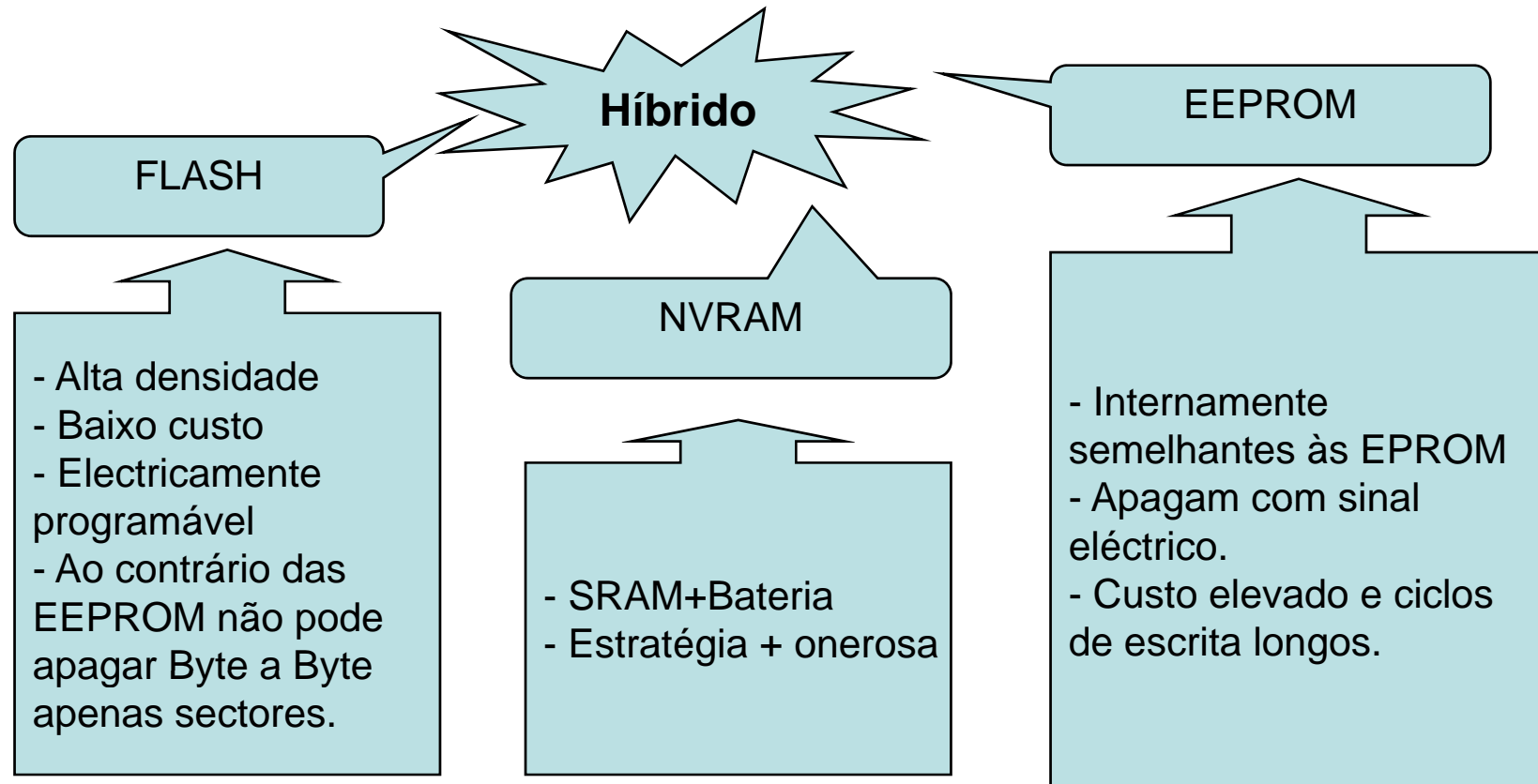
MEMÓRIAS HÍBRIDAS

O avanço da tecnologia fez esbater a fronteira entre **ROM** e **RAM**



Reúnem o melhor dos dois mundos...

- Podem ser lidas e escritas diversas vezes (como as **RAM**)
- Mantém o conteúdo armazenado mesmo na ausência de alimentação (como as **ROM**)



MEMÓRIAS ROM: *Tópicos de Hardware*

- Função booleana qualquer pode ser escrita como uma “soma de produtos” ou como um “produto de somas”
- Uma memória ROM consiste numa função que executa um mapeamento ENDEREÇO/DADO
- Uma memória ROM pode ser criada como SdP ou PdS

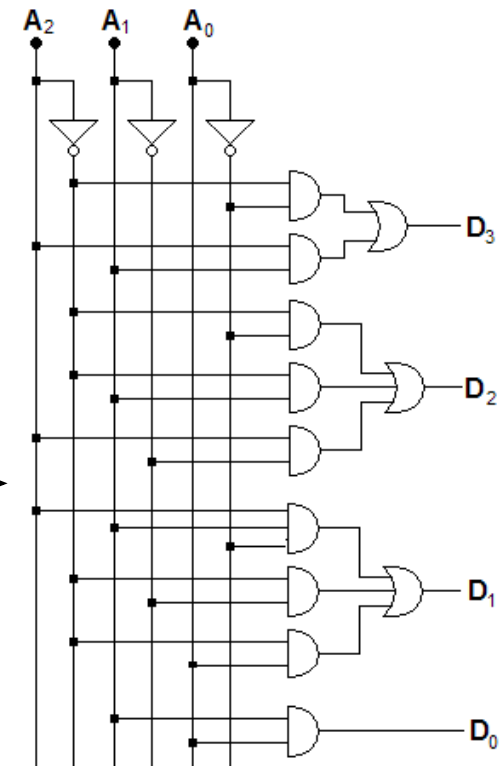
Ex. Criar uma memória **ROM** capaz de armazenar 8 palavras de 1 nibble

Endereço != para cada palavra => 8 endereços

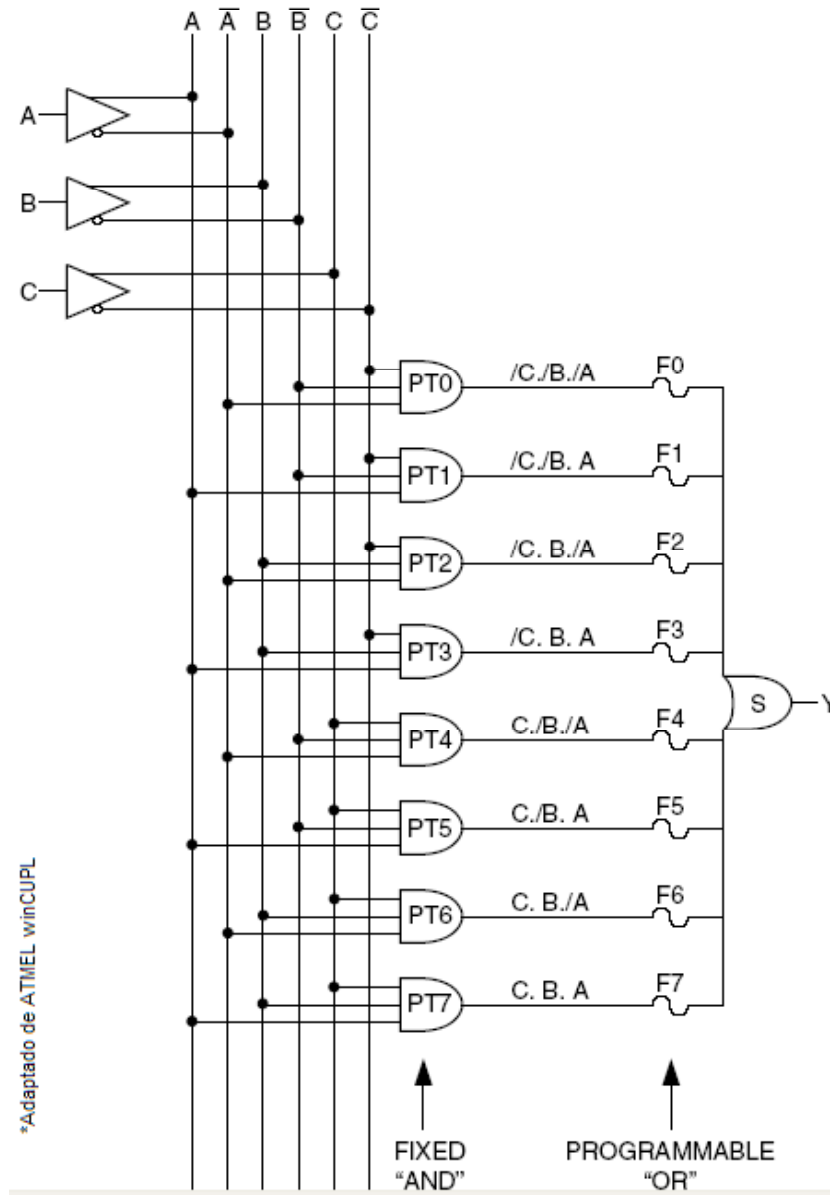
Necessários 3 bits para distinguir 8 endereços

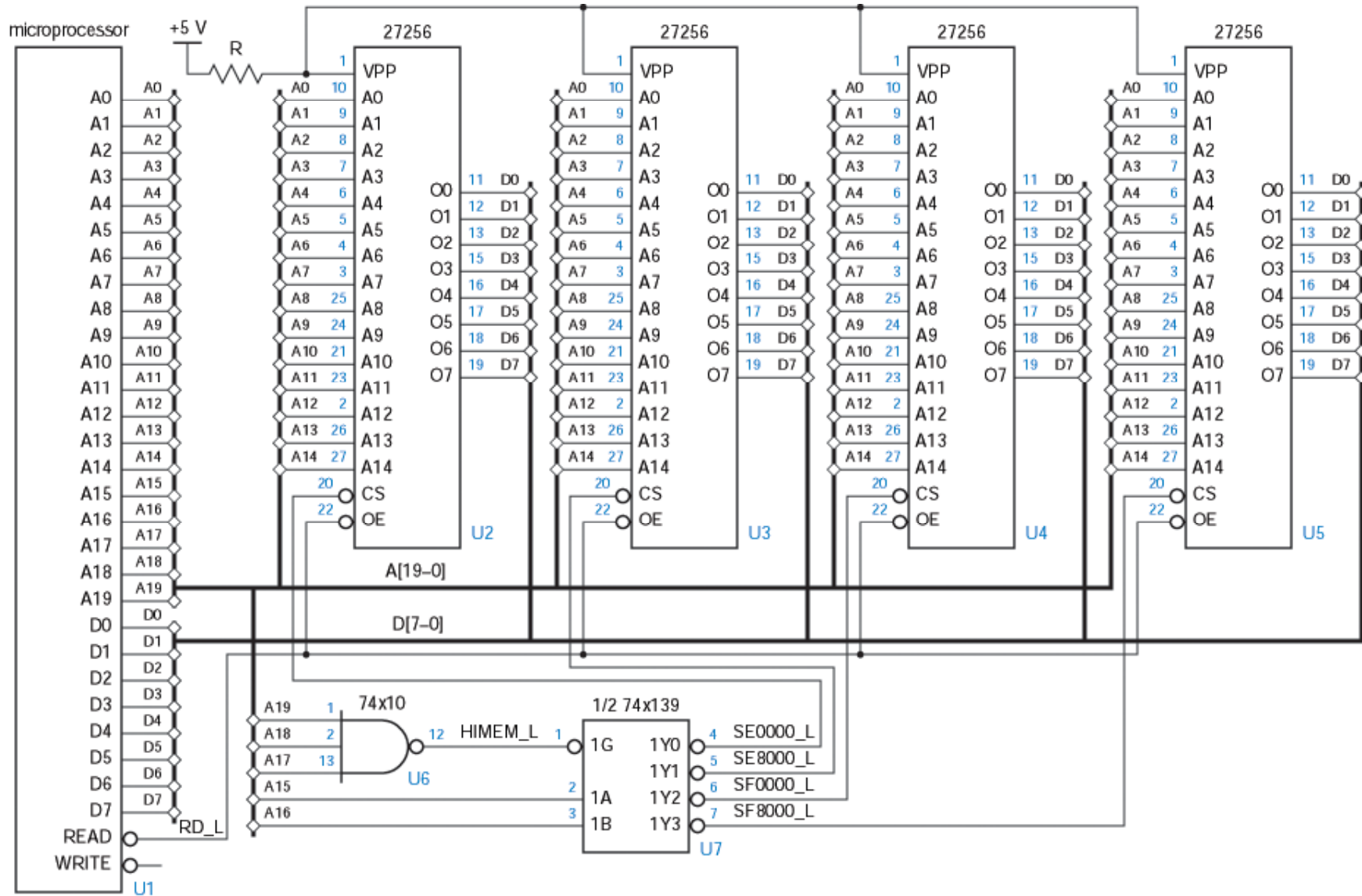
conteúdo da ROM

ENDEREÇO			DADOS			
A2	A1	A0	D3	D2	D1	D0
0	0	0	1	1	1	0
0	0	1	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	1	1
1	0	0	1	1	0	0
1	0	1	1	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



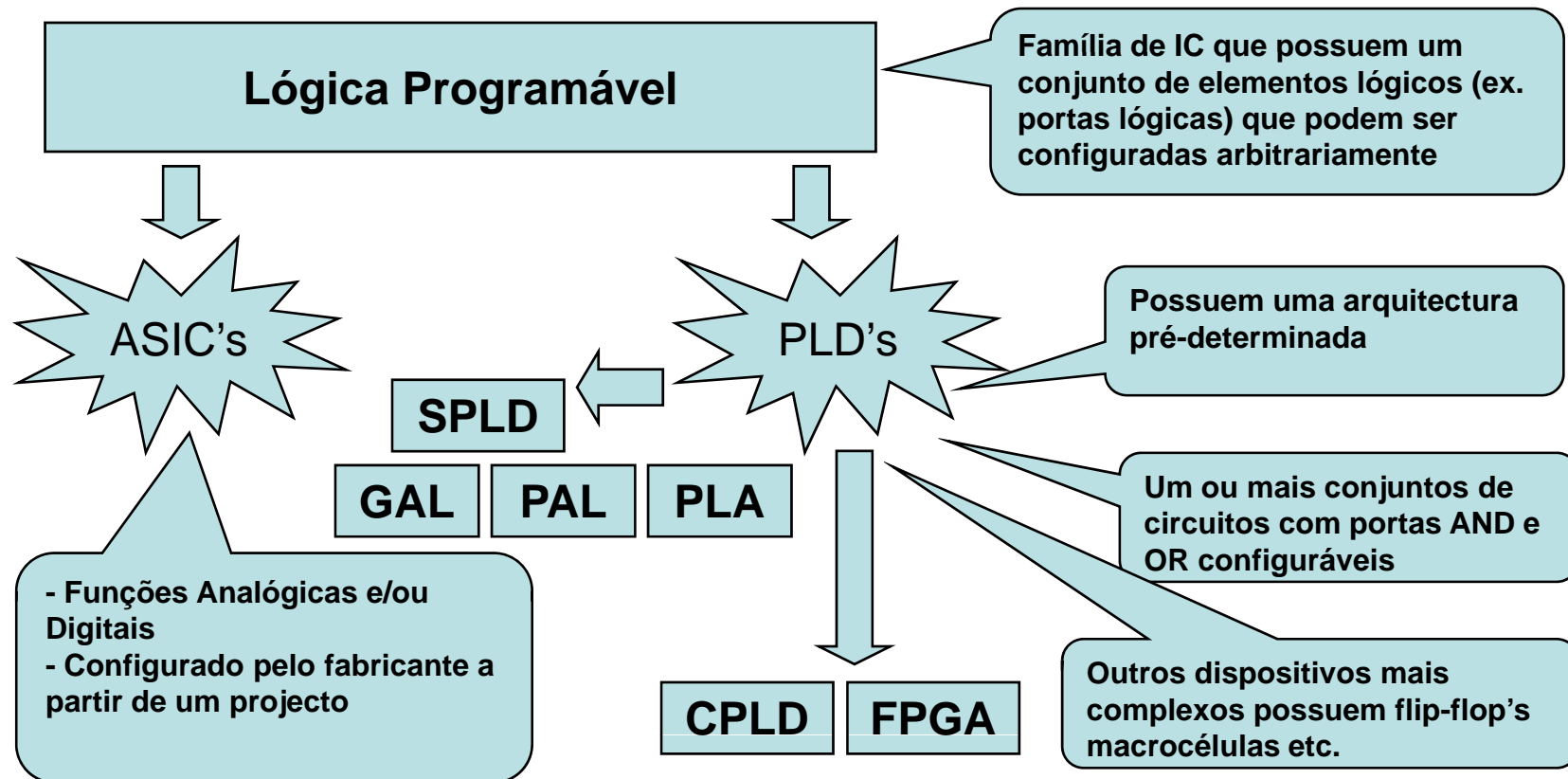
Da ROM para a PROM...





LÓGICA PROGRAMÁVEL

- Multiplexer como forma versátil de implementar funções booleanas arbitrárias
- ROM's também podem ser usadas para esse fim
- Outras estratégias que podem envolver, além de lógica combinatória, lógica sequencial



Ao contrário dos CI digitais clássicos, em que a sua função era rígida, nos PLD's o mesmo CI pode executar um sem número de funções!

	1960s SSI/MSI	1970s LSI	1980s VLSI	1990s Programmable Logic
<i>Components</i>	Logic, Resistor/ Transistor elements	8-bit μ processor, memory, ROM	32-bit μ processor, gate arrays	64-bit μ processor, PALs, FPGAs
<i>Complexity Level (# of gates)</i>	100s	10,000s	1 million	100,000s to millions
<i>Pervasive Components</i>	TTL 7400 Series	Intel 8008, ROM	Intel 8086, Motorola 68000, gate arrays, PALs	Pentium I, II, III, FPGAs, Complex PLDs
<i>Dominant Trend</i>	standard catalog of components	larger, general- purpose components, e.g., microprocessors and random access memories	application- specific integrated circuits	field programmable components

SPLD

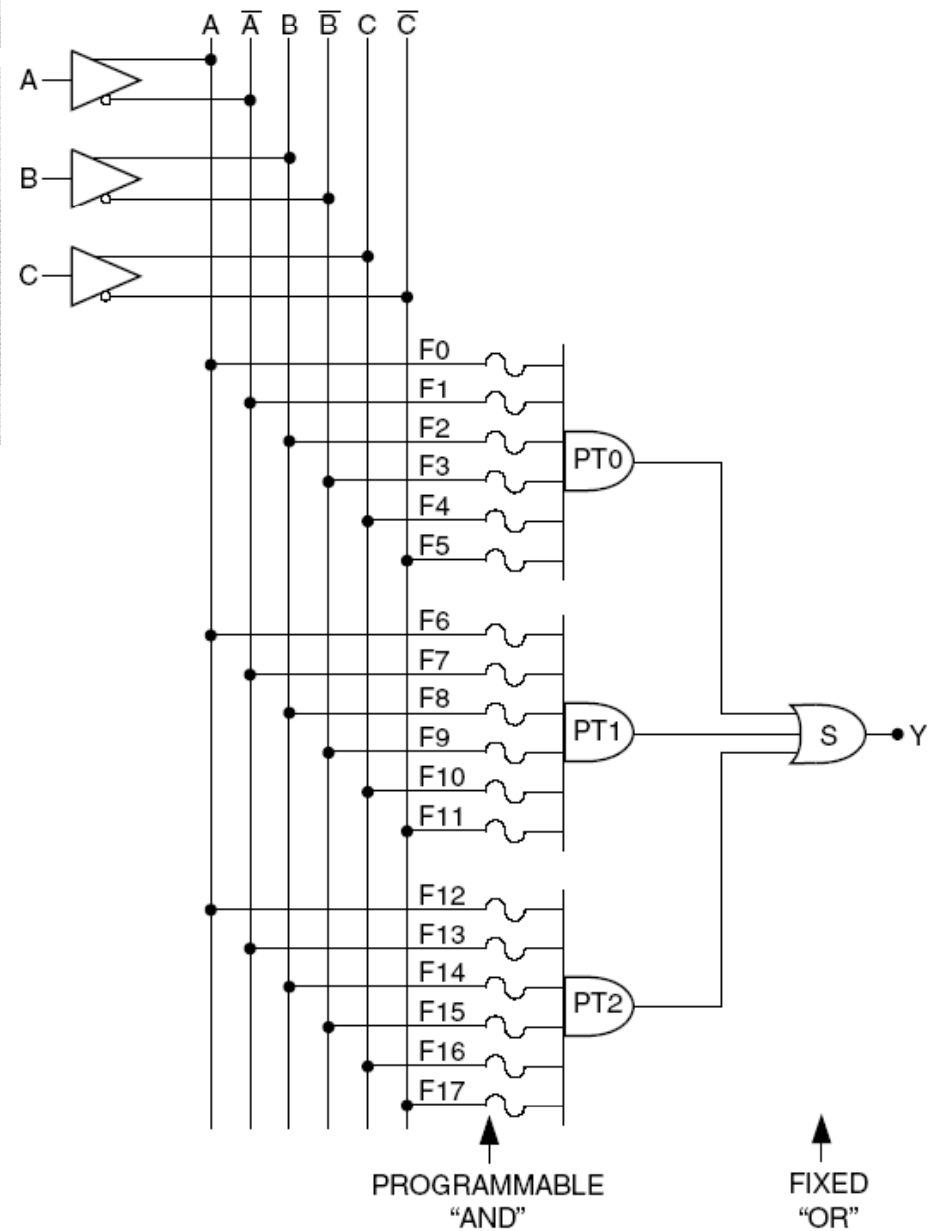
Em forte declínio, são já poucos os fabricantes que ainda produzem SPLDs

- Lattice
- Cypress
- AMD
- PHILIPS
- TI

PAL's

PAL...

- Programmable Array Logic
- Implementação da forma disjuntiva
- Apenas a matriz de AND é programável
- Tipo de Arquitectura PLD mais popular
- Apenas as variáveis de cada termo mínimo podem ser programadas



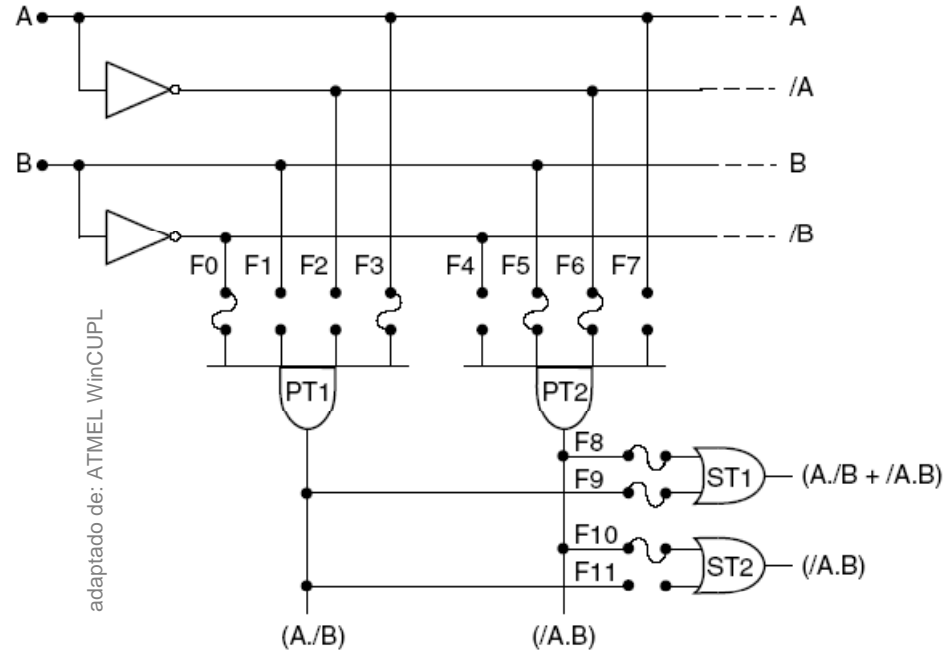
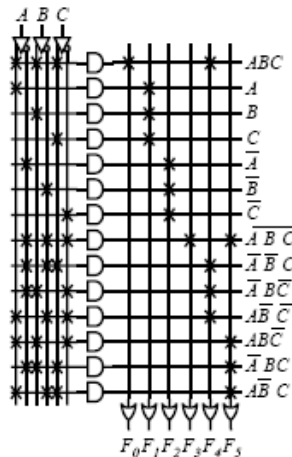
PLA's

PLA...

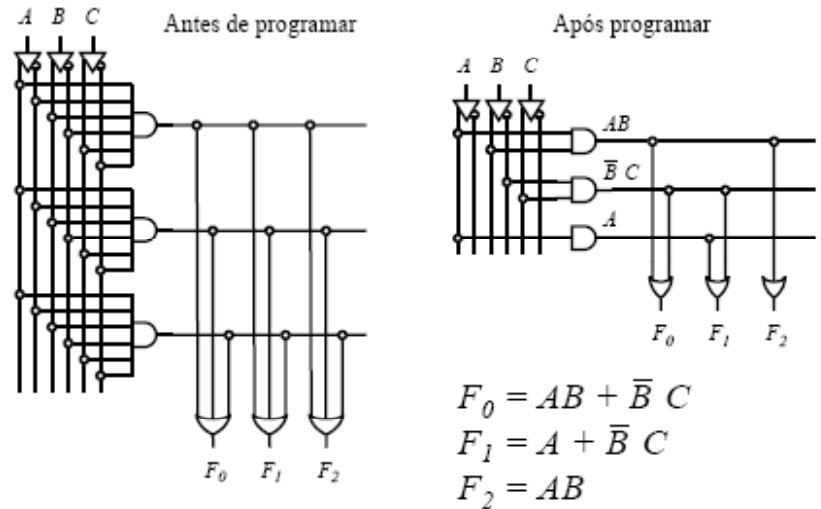
- Programmable Logic Array
- Possui tanto termos programáveis AND como OR
- Maior Flexibilidade
- Normalmente possuem realimentação entre a saída de cada OR e a entrada do conjunto de AND
- Permite implementar máquinas de estado assíncronas

exemplos

- $F_0 = ABC$
- $F_1 = A + B + C$
- $F_2 = \overline{ABC} = \overline{A} + \overline{B} + \overline{C}$
- $F_3 = \overline{(A+B+C)} = \overline{A} \overline{B} \overline{C}$
- $F_4 = A \oplus B \oplus C = \overline{A} \overline{B} C + \overline{A} B \overline{C} + A \overline{B} \overline{C} + ABC$
- $F_5 = \overline{(A \oplus B \oplus C)} = ABC + \overline{A} \overline{B} C + A \overline{B} C + \overline{A} B C$



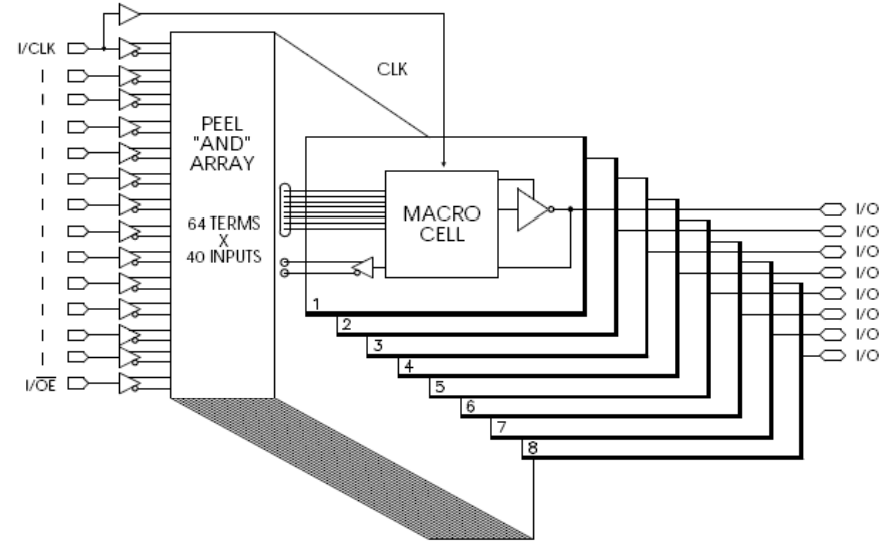
adaptado de: Luís Laus, Electrónica Digital



2ª Geração de PAL's: GAL's

GAL (Generic Array Logic)...

- Desenvolvido pela LATTICE
- Capazes de implementar tanto funções combinatórias como sequenciais (contadores, registadores, etc.)
- A sua velocidade compete com a da série 74XX da TTL
- Podem ser **reprogramadas** (~100X)
- Utilização de "macro-células" à saída



AR –RESET assíncrono
 SP – PRESET síncrono

S0 e S1 são "fuses"

REGISTERED MODE

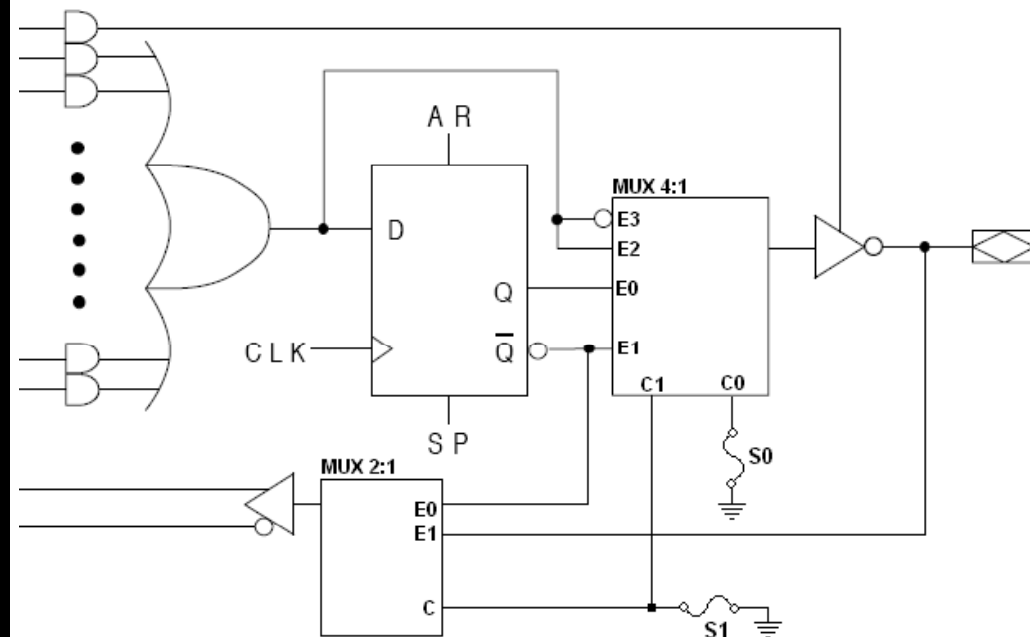
S1=0 S0=X

COMBINATIONAL MODE

S1=1 S0=X

NOTA: As saídas também podem ser entradas (como?)

MACRO-CÉLULA



Exemplo de Aplicação: PAL 16L8

$$x(A,B,C,D) = \Sigma(7,8,9,10,11,12,13,14,15)$$

$$y(A,B,C,D) = \Sigma(0,2,3,4,5,6,7,8,10,11,15)$$

$$z(A,B,C,D) = \Sigma(1,2,8,12,13)$$

$$w(A,B,C,D) = \Sigma(2,12,13)$$

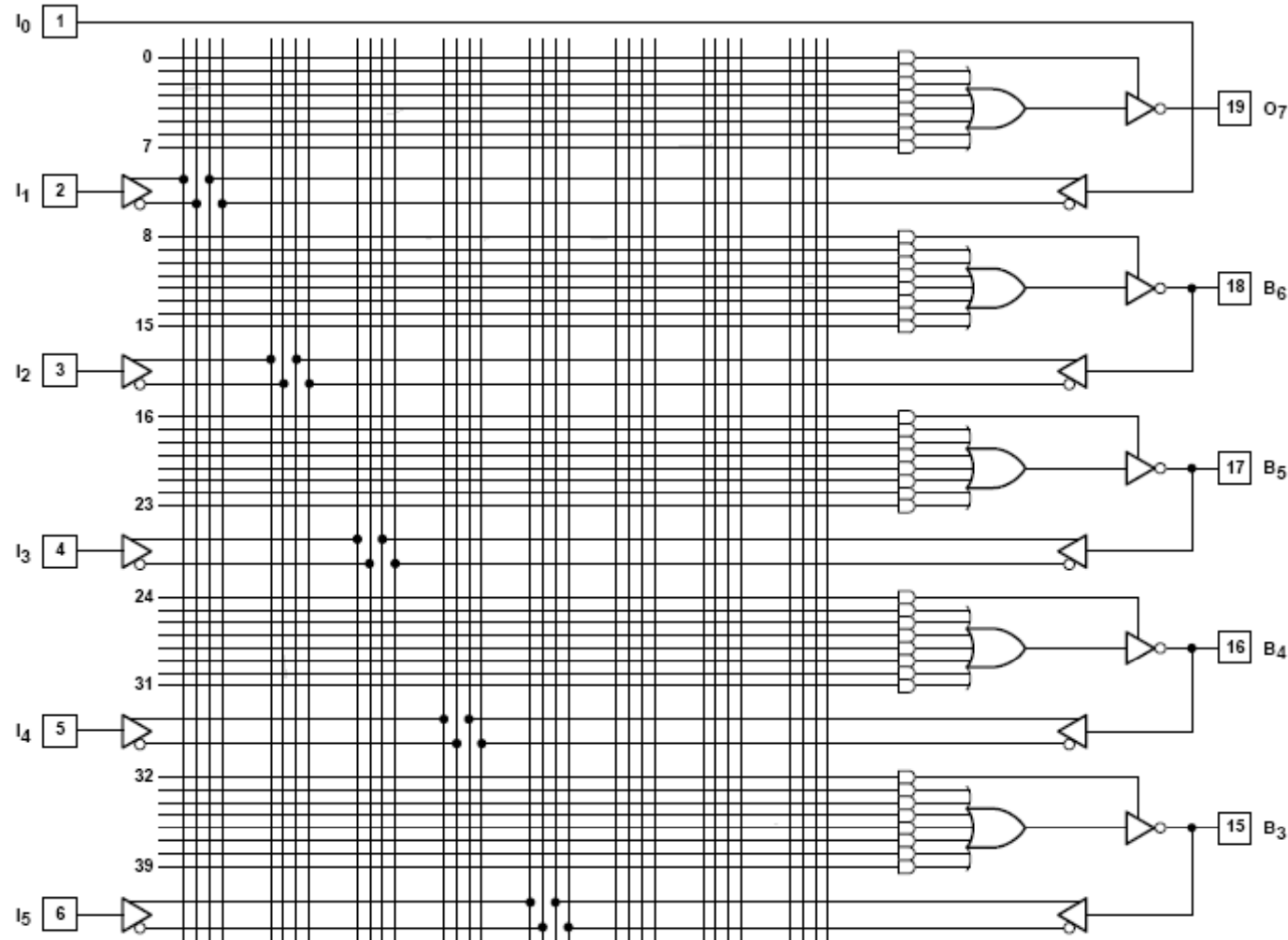
após simplificação...

$$x(A,B,C,D) = A + BCD$$

$$y(A,B,C,D) = A'B + CD + B'D'$$

$$z(A,B,C,D) = w + AC'D' + A'B'C'D$$

$$w(A,B,C,D) = ABC' + A'B'CD'$$

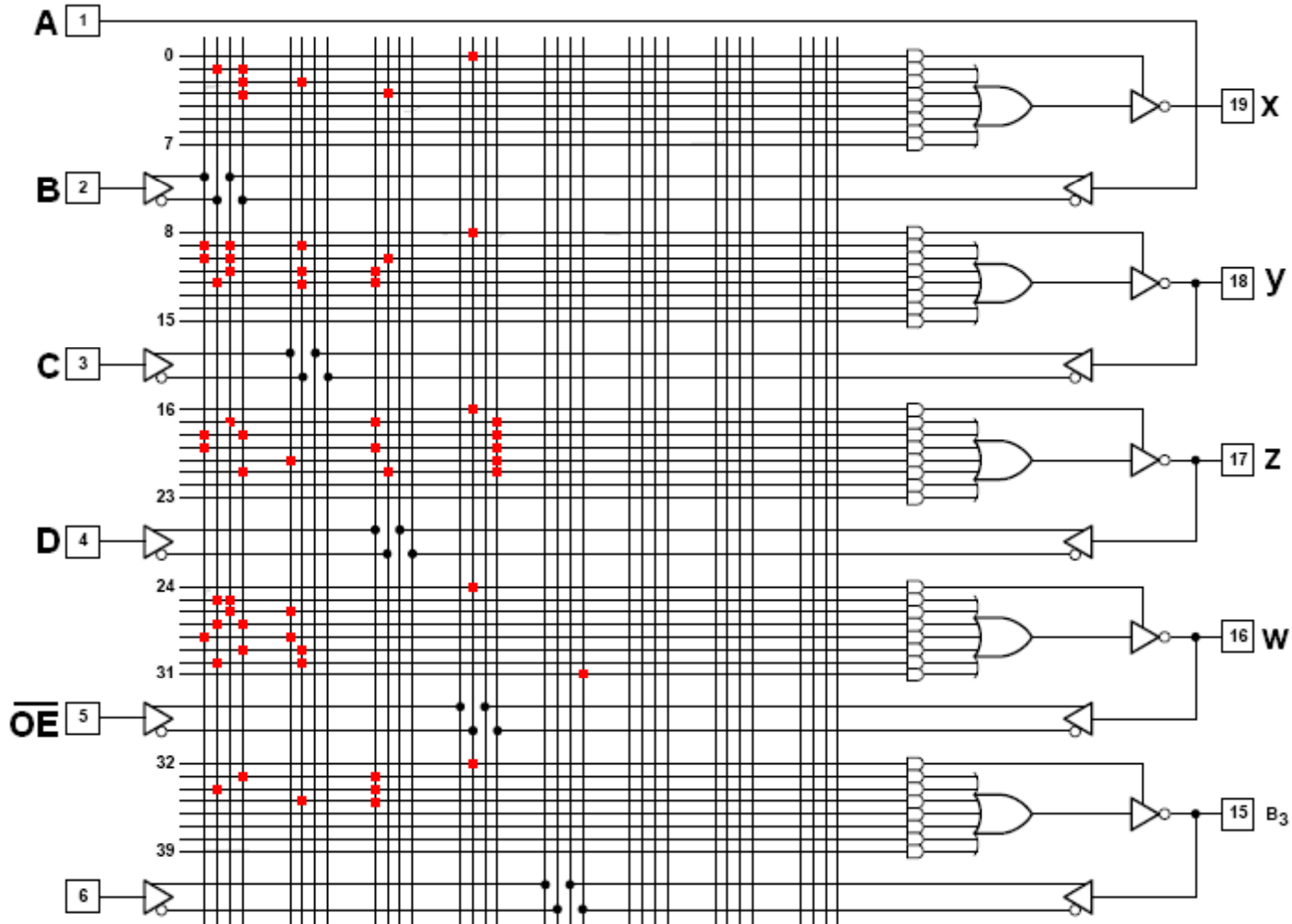


$$x(A,B,C,D) = \overline{\overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C} + \overline{A} \cdot \overline{D}}$$

$$y(A,B,C,D) = \overline{A \cdot \overline{C} \cdot B + A \cdot \overline{D} \cdot B + A \cdot \overline{C} \cdot D + \overline{B} \cdot \overline{C} \cdot D}$$

$$z(A,B,C,D) = \overline{\overline{w} \cdot D \cdot A + \overline{w} \cdot \overline{A} \cdot B + \overline{w} \cdot D \cdot B + \overline{w} \cdot C + \overline{w} \cdot \overline{A} \cdot \overline{D}}$$

$$w(A,B,C,D) = \overline{\overline{B} \cdot A + C \cdot A + \overline{A} \cdot B + C \cdot B + \overline{A} \cdot \overline{C} + \overline{B} \cdot \overline{C} + \overline{A} \cdot D + \overline{B} \cdot D + C \cdot D}$$



Como 'e que as PLD's armazenam a informação?

Num PLD a memória é usada para guardar o padrão que é dado ao IC durante a programação. Os métodos usados para a programação podem ser do seguinte tipo

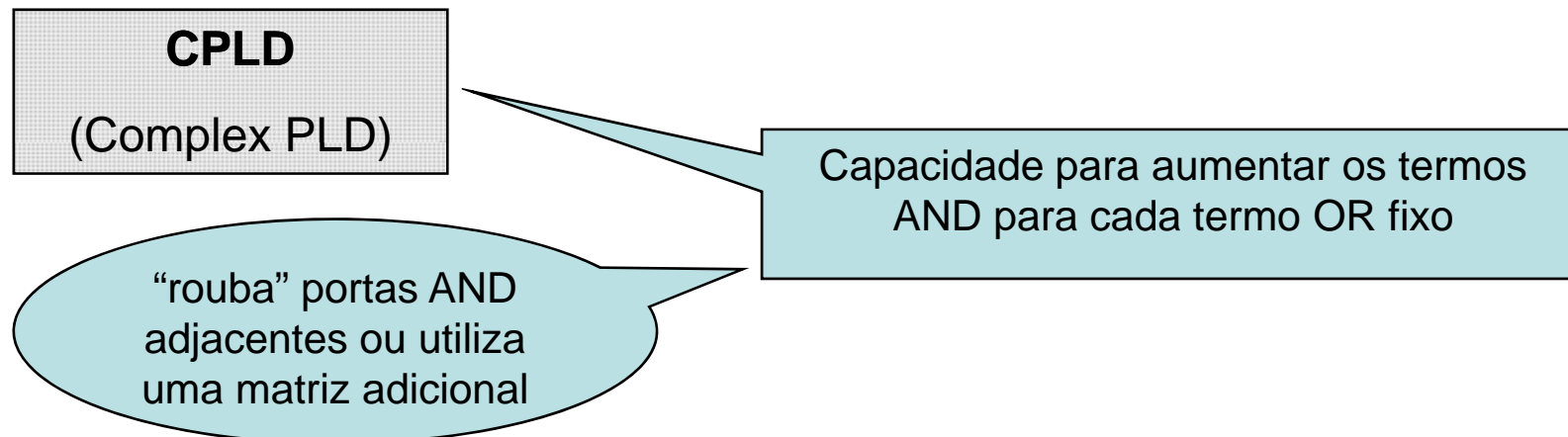
- Anti-fusíveis Silício (PROM)
- SRAM
- Células EPROM, EEPROM
- Memória Flash

CPLD's

As PALs e GALs estão disponíveis somente em tamanhos pequenos, equivalentes a algumas centenas de portas lógicas.

Para circuitos lógicos maiores pode-se usar **Complex PLD** ou **CPLDs**.

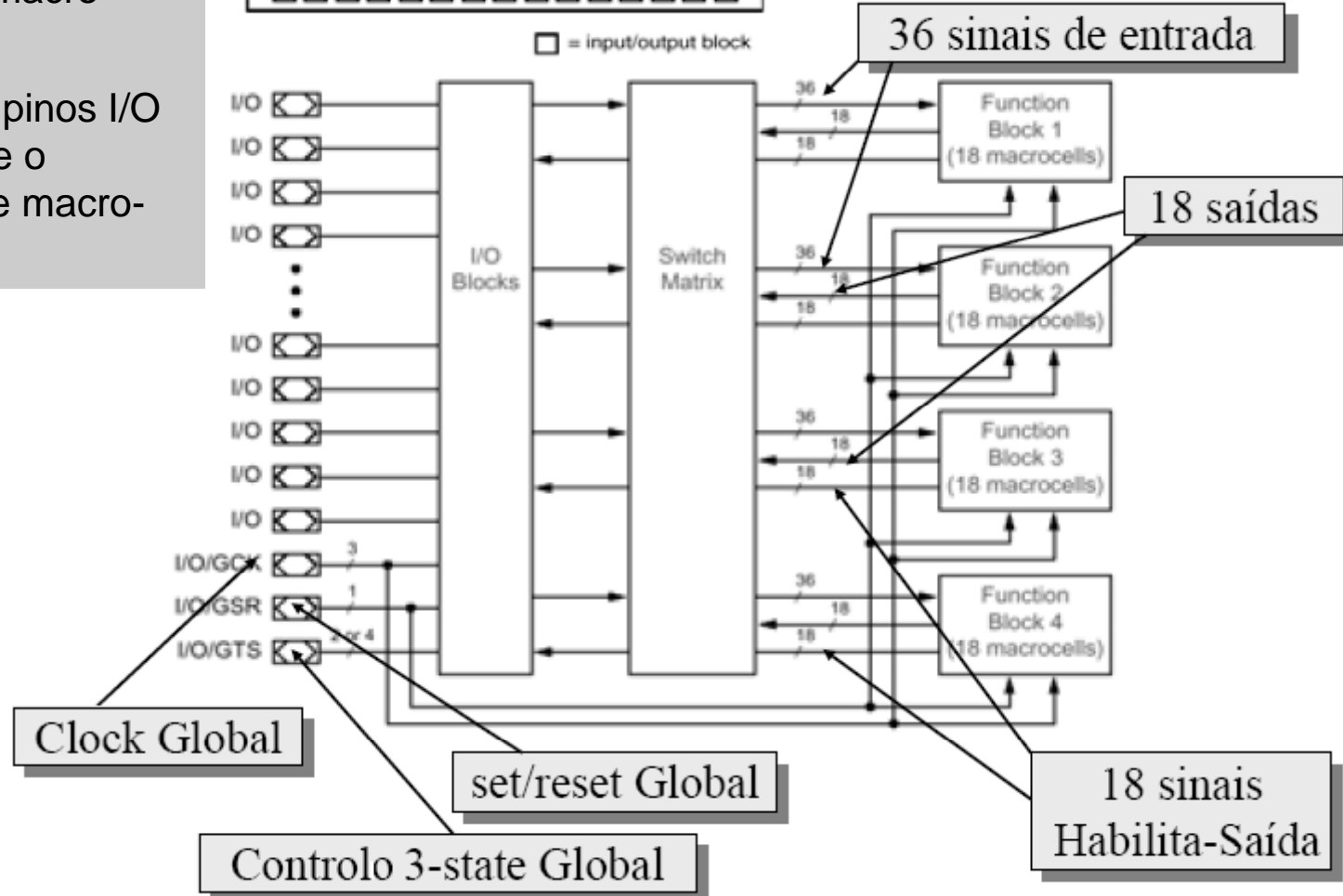
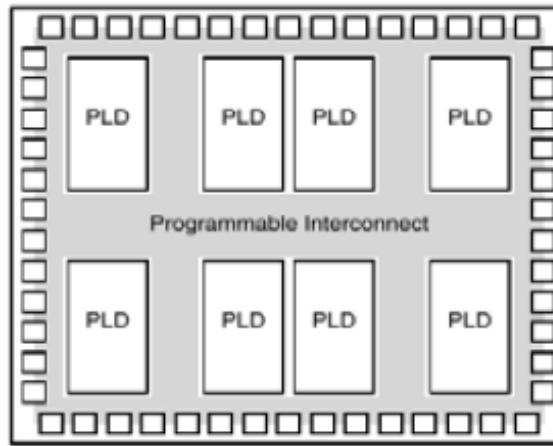
Este contém o equivalente a algumas PALs ligadas por interconexões programadas.



As PLD's internas são designadas por FB's ou **CFB's** (Configurable Functional Blocks)

XILINX '9500

- Cada **CFB** possui 36 entradas e 18 macro-células
- O número de pinos I/O é menor do que o número total de macro-células



FPGA's

Dispositivo constituído por um conjunto de células lógicas ou blocos lógicos alocados em forma de uma matriz

FPGA
(Field Prog. Gate Arrays)

- Elevada densidade de portas
- Elevado desempenho
- Elevado número de I/O
- Esquema de interligação flexível

Surgiram em 1985 com Xilinx
Outras companhias:
Actel, Altera, Plessey, Plus, AMD, etc.

Estrutura Interna

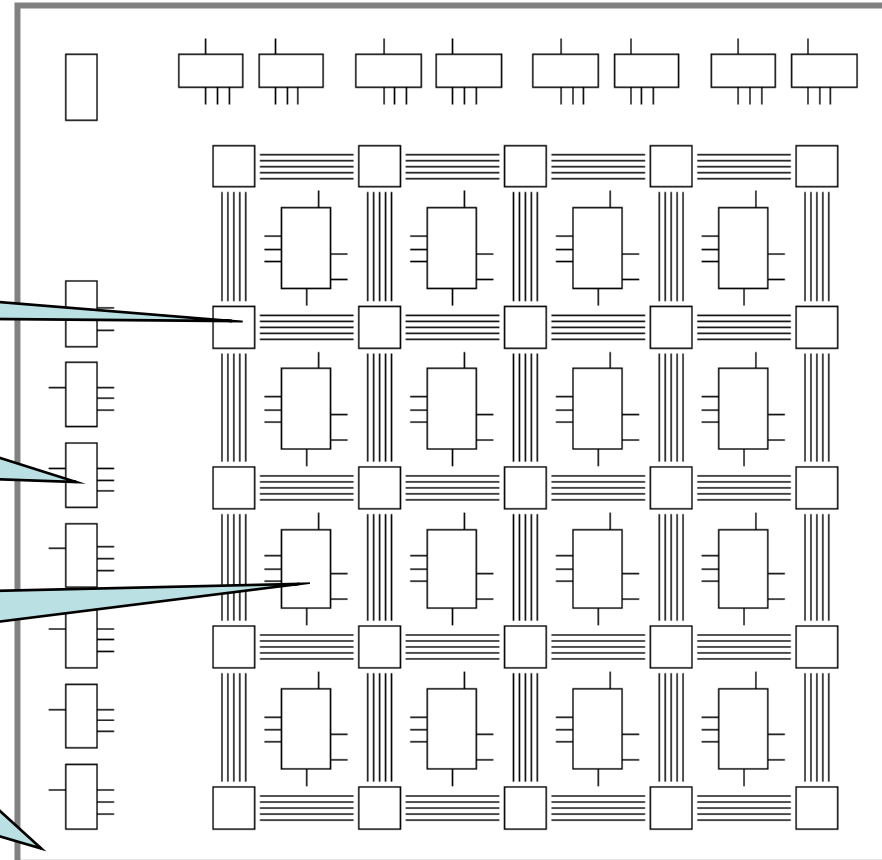
- **CLB's** (Configurable Logic Block)
- **IOB's** (In/Out Block)
- **SB's** (Switch Box)

Switch Box (SB)

Cada IOB pode ser configurado como entrada, saída, I/O.(Bufferizadas).

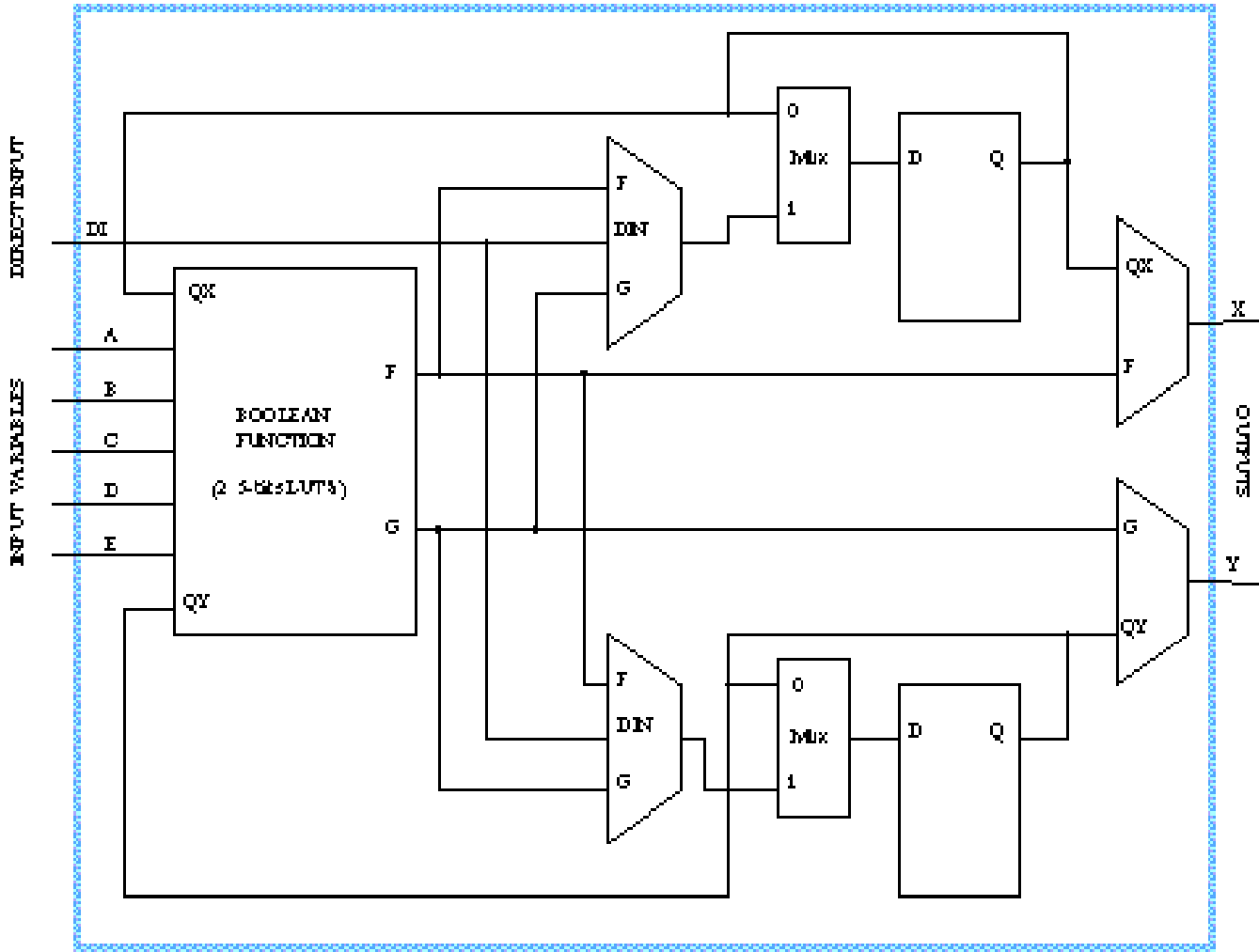
Cada CLB possui lógica combinatória programável e registos de memória.

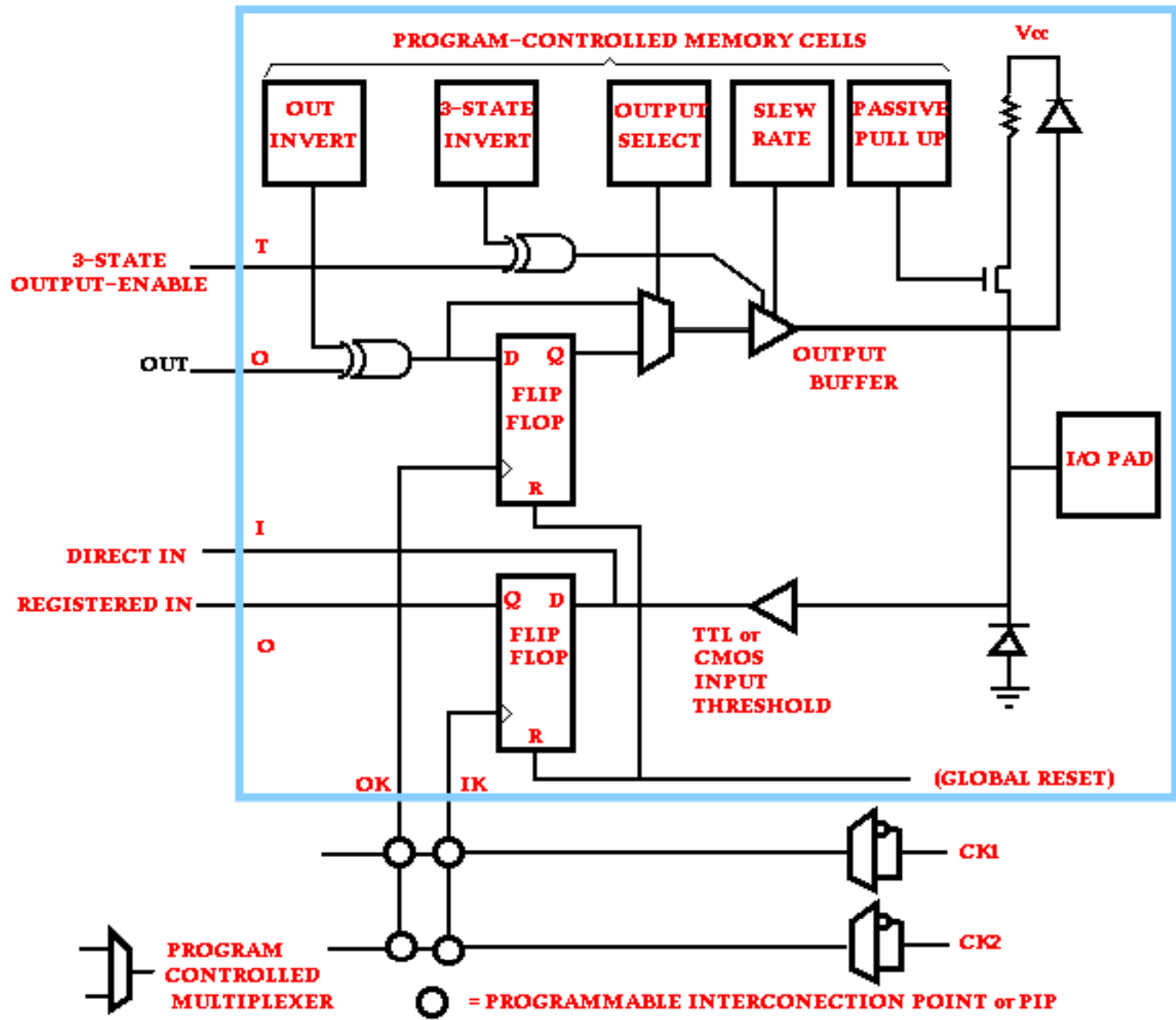
Possui uma matriz de CLB's [configurable logic blocks] circundada por um anel de blocos de I/O (IOB's)



CLB's

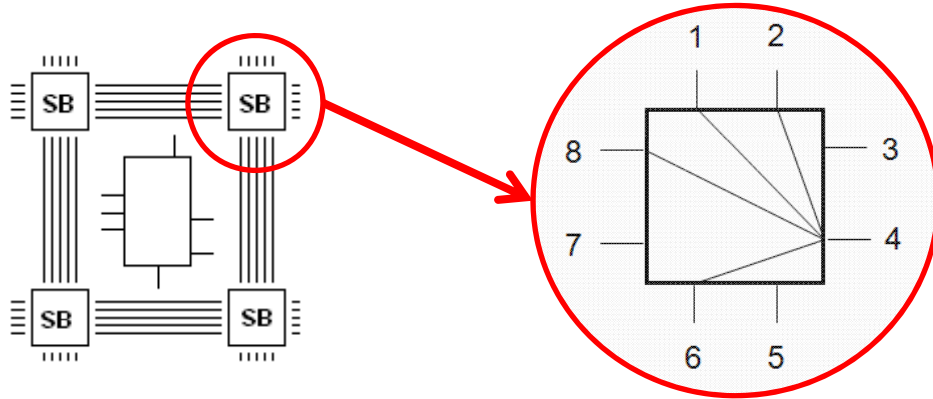
Xilinx XC3000





Switch Box

Permite a interconexão entre os **CLB's** através dos canais de **roteamento**

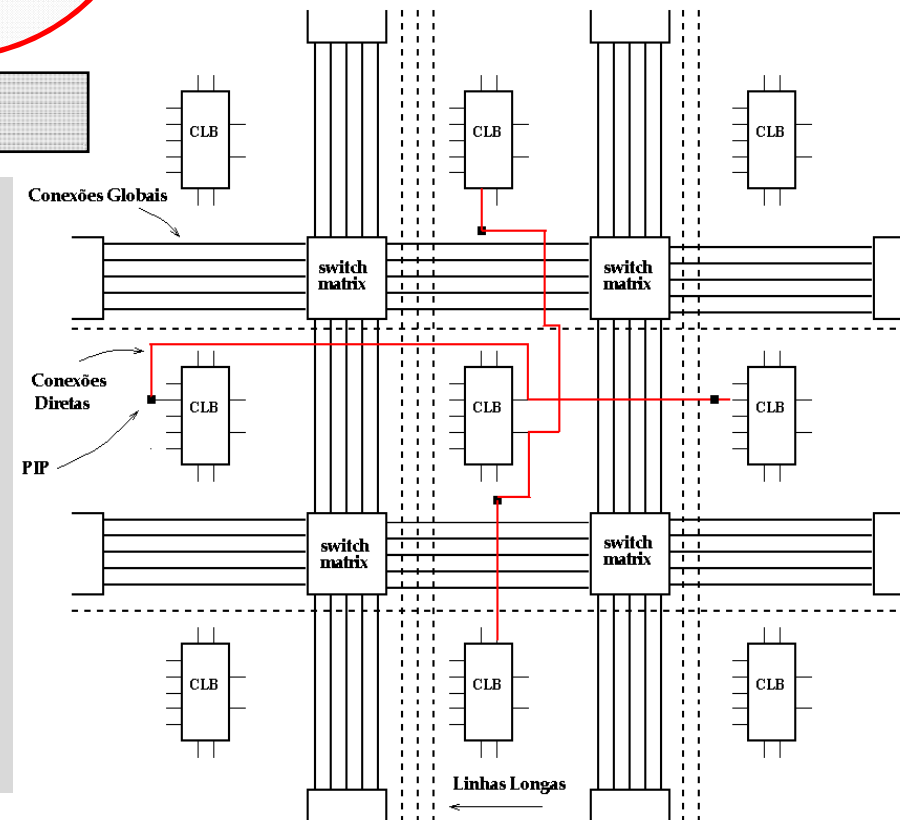


Roteamento

A interligação entre os blocos é feita através de uma rede de duas camadas de metal.

As ligações físicas entre os fios são feitas:

- .Com transistores de passagem controlados por bits de memória (PIP),
- .Com chaves de interconexão (Switch Matrix).



Programação de SPLD's, CPLD's e FPGA's

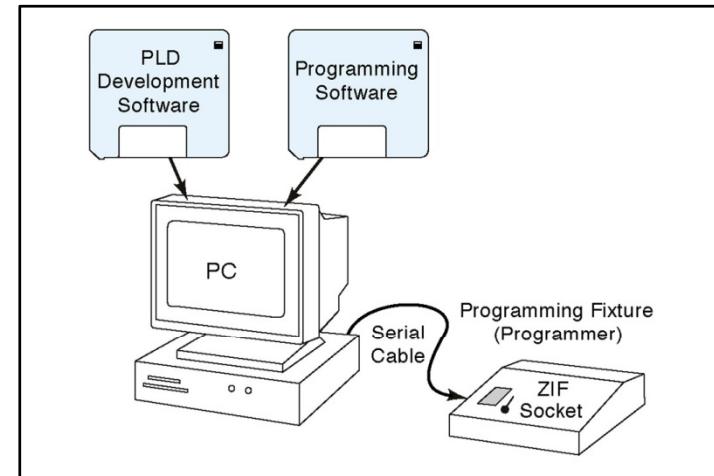
Se bem que seja possível projectar um PLD simples manualmente, no caso de dispositivos mais complexos como é o caso dos CPLD's e FPGA's é obrigatório o uso de ferramentas CAD para o seu projecto.

PAL's:

Aceitam na entrada um formato de ficheiro conhecido por ficheiro "JEDEC" [Joint Electron Device Engineering Council].

Diversas linguagens para compiladores lógicos:

- CUPL, ABEL, HDL e VHDL, etc.



- Nos CPLD's e FPGA's, devido ao tipo de encapsulamento utilizado, a programação deve ser "em circuito".
- Cada fabricante tem um nome proprietário para este sistema de programação.
- A Lattice Semicondutor chama-lhe programação "in-system" (ISP).
- Normalização através da Joint Test Action Group (**JTAG**).