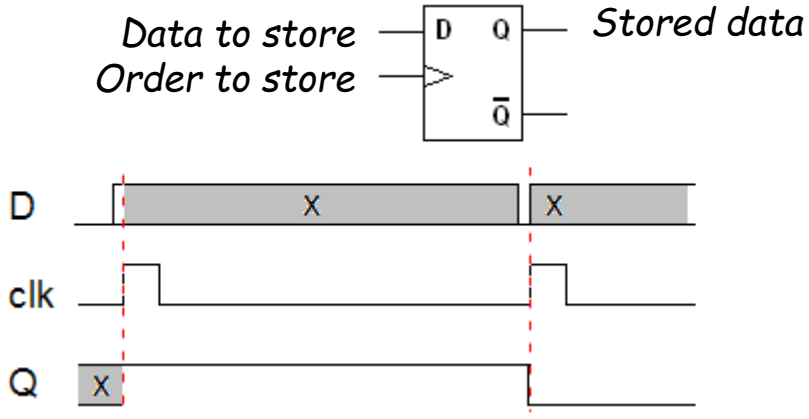
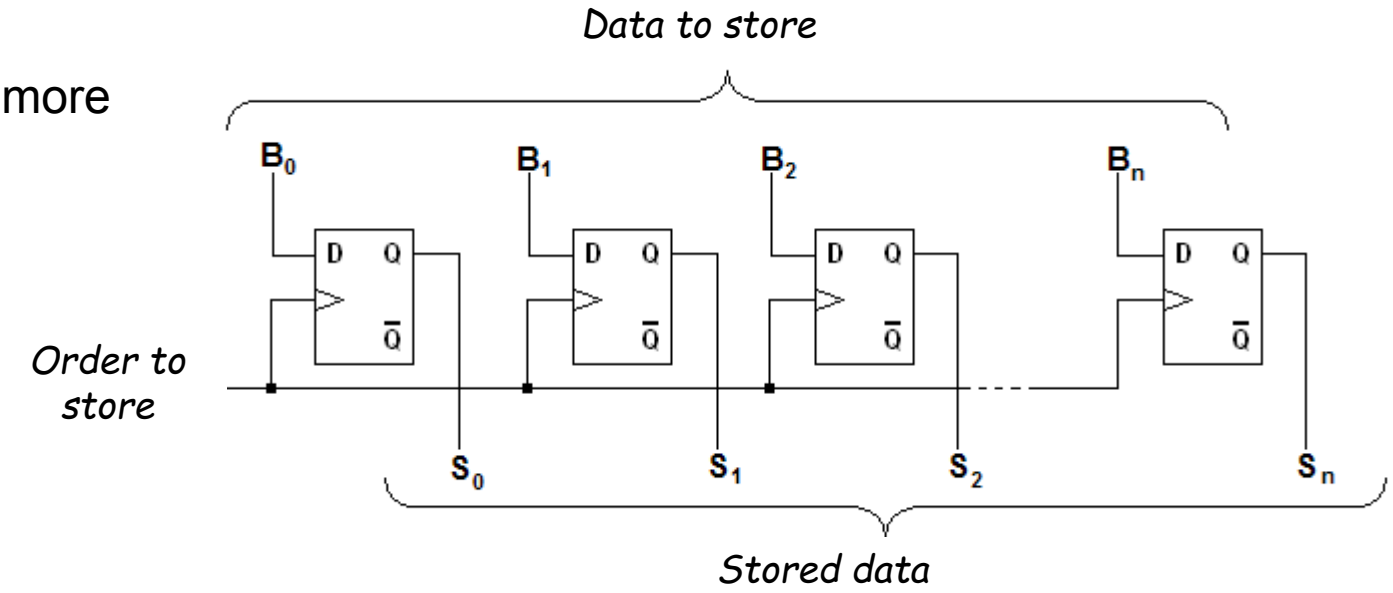
MEMORIES

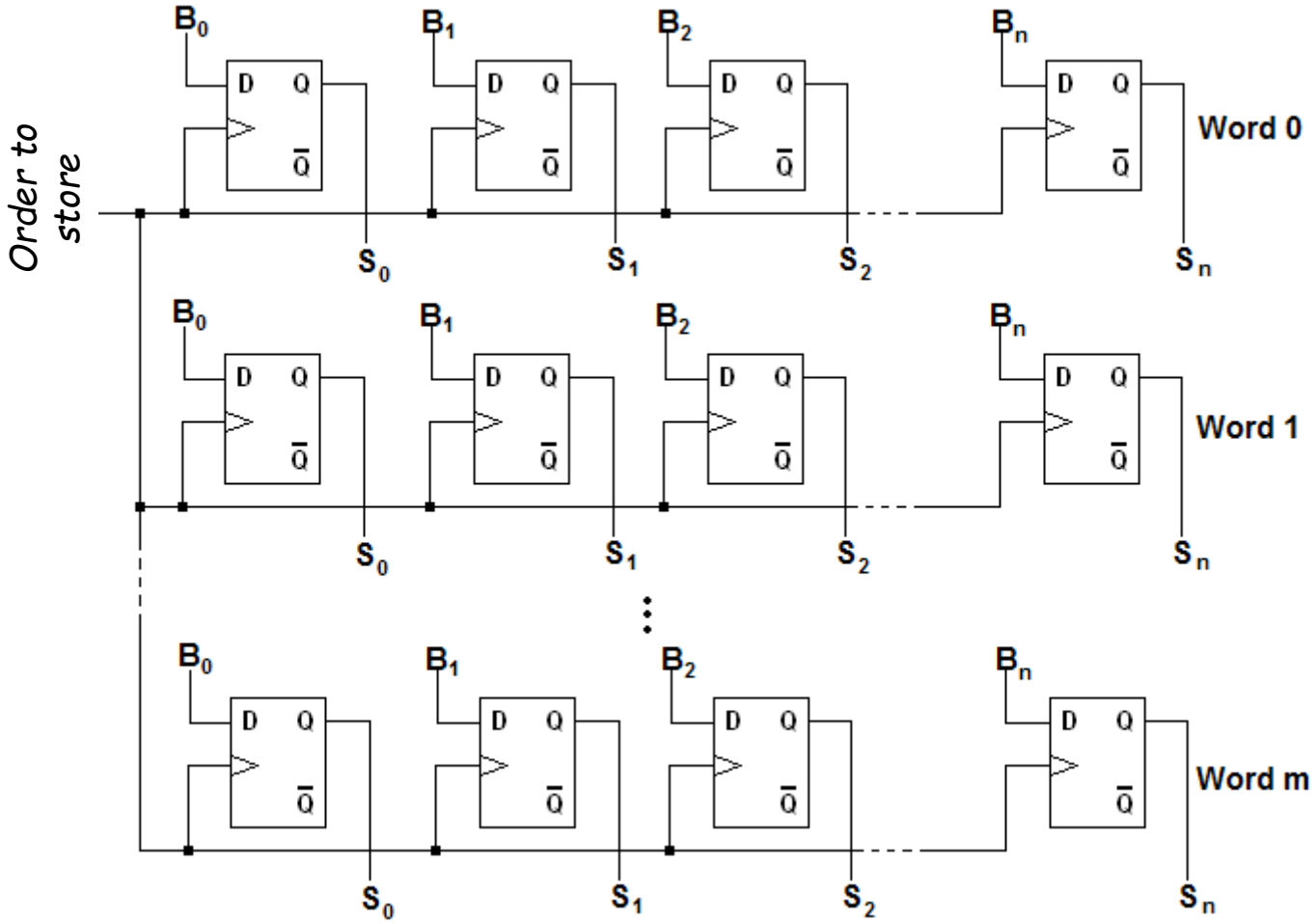


Until this moment ...
MEMORY =FLIP-FLOP

How to store more than one bit?

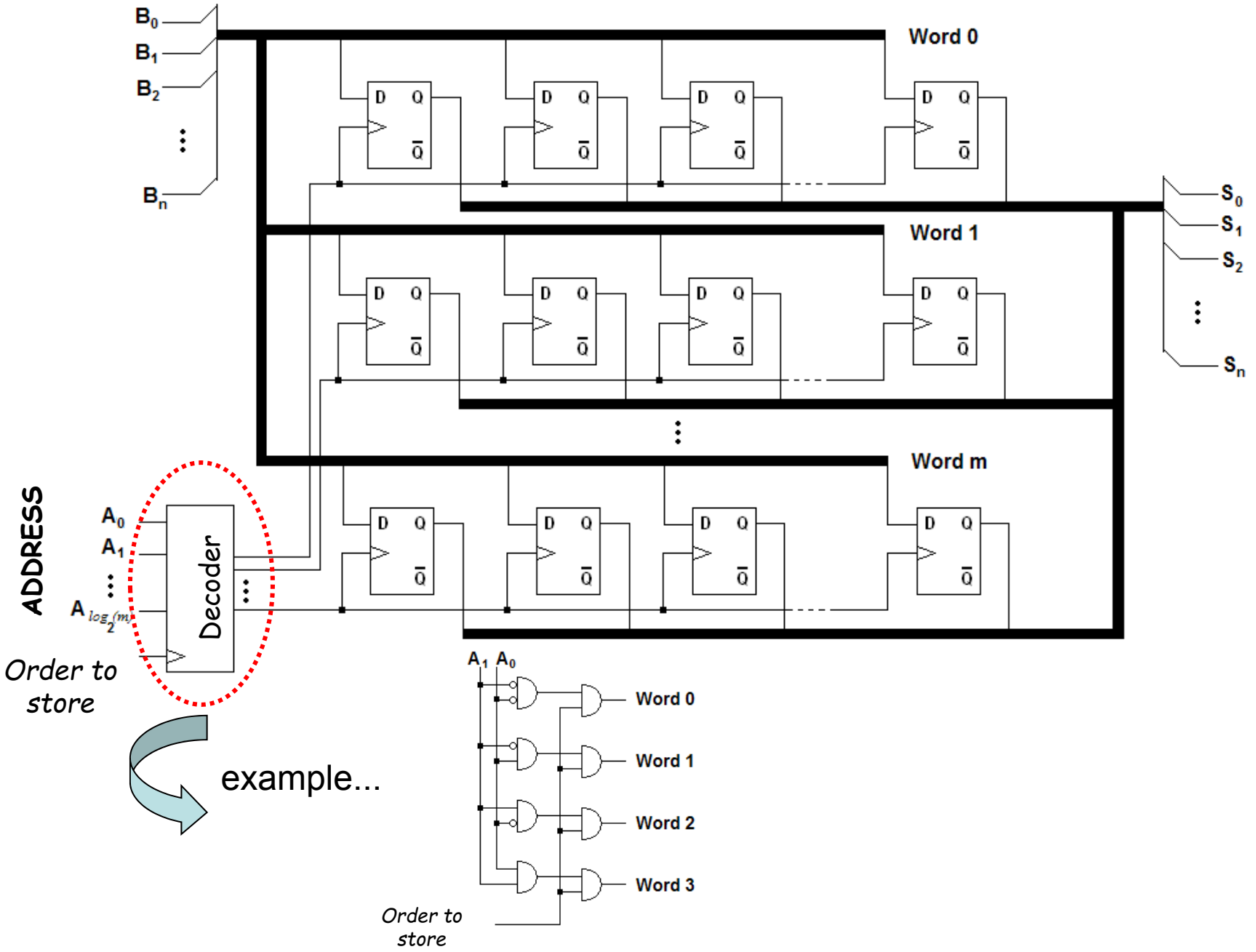


How to store more than one word?

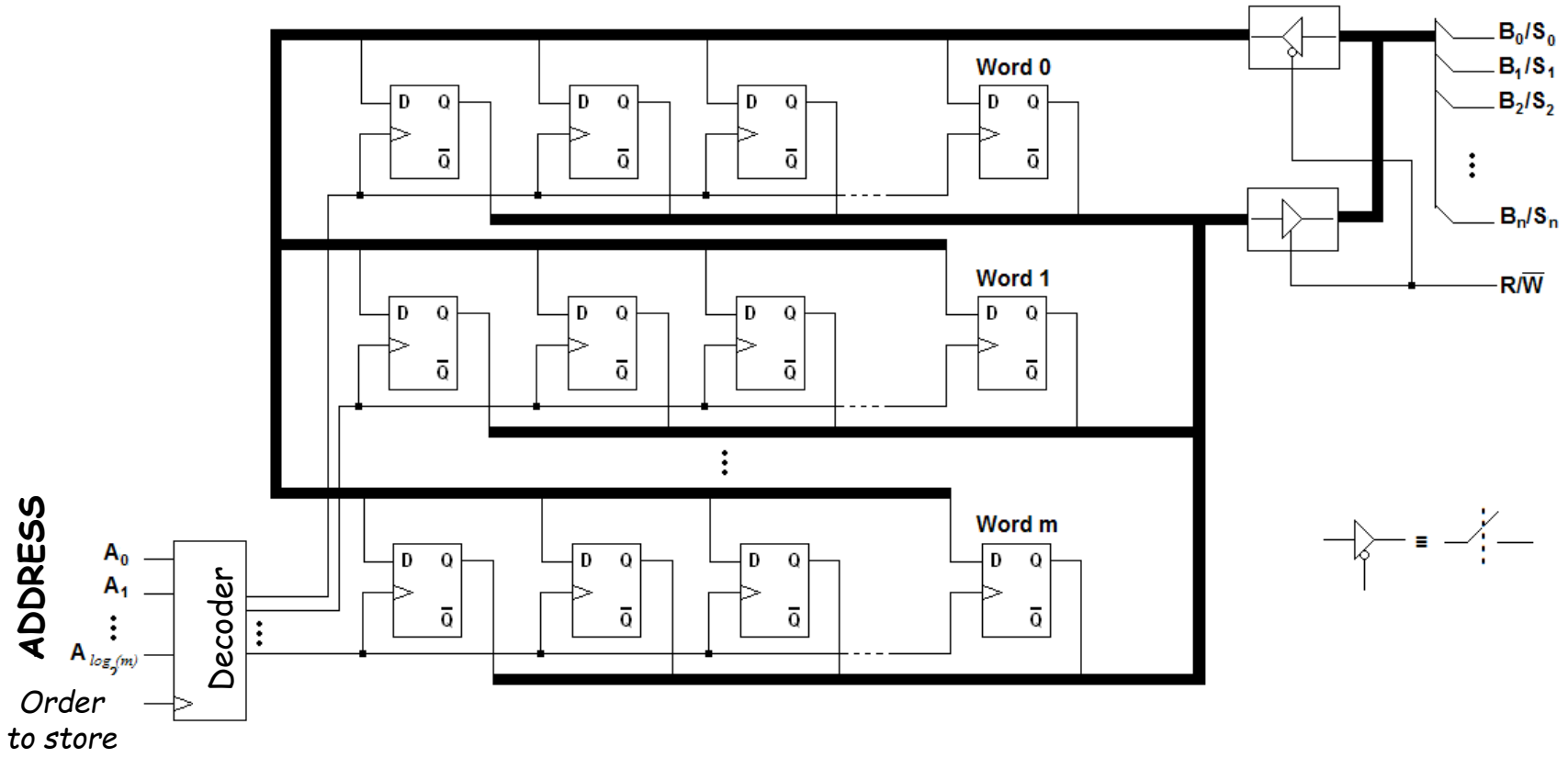


How many **pins** should have an IC capable of storing 1024 bytes (W = 1B)

Concept of **ADDRESS** lines...



To reduce the number of IC pins often the I/O terminals are common…



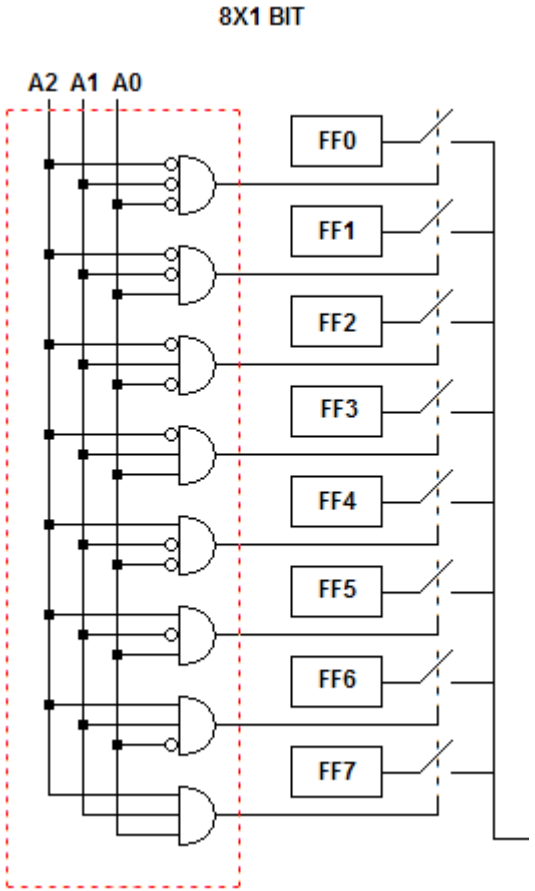
- Is it possible, at any given moment, change the k bit of word w?
- This is a read / write memory (RWM) but is not a RAM!

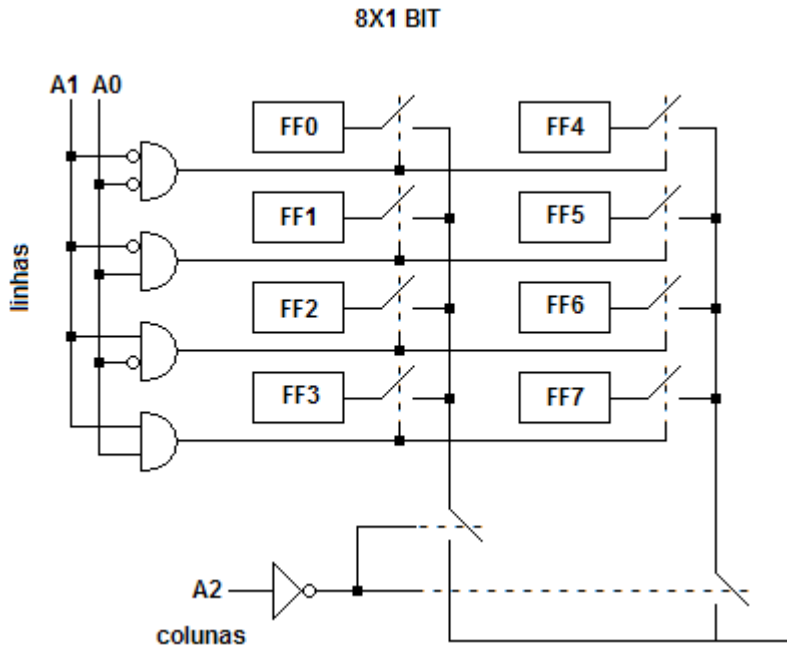
RAM Memory (Random Access Memory)

- Random Access Memory
- Name assigned to RWM with the ability to index cell by cell

INTERNAL ORGANIZATION:

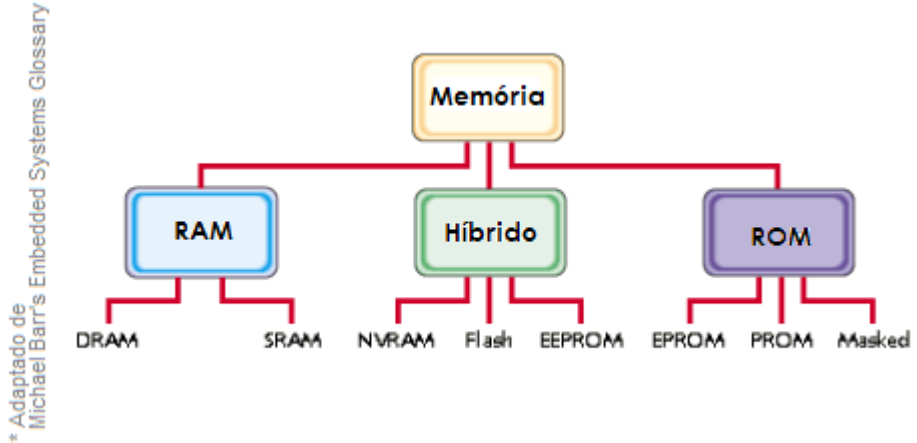
1-D 2-D





The 2-D version requires less “hardware”

- Besides the RAM memories there are many other memory types…



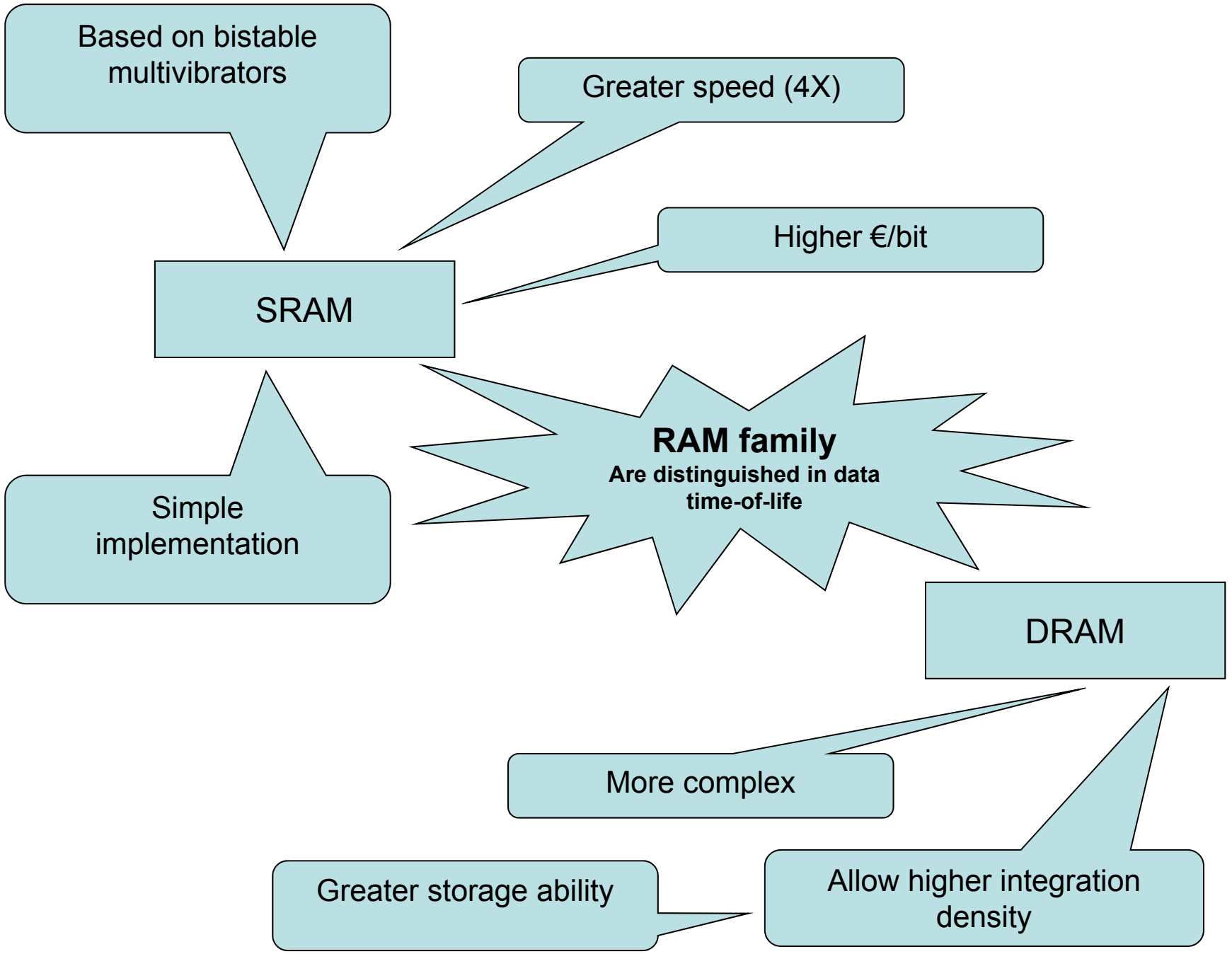
RAM MEMORIES

It can read / write to any memory cell

Requires constant power, in contrary information is lost

RAM
(Random Access Memory)

Used in computer architectures for temporary storage of variables


Based on bistable multivibrators
Greater speed (4X)
Higher €/bit
SRAM
Simple implementation
RAM family
Are distinguished in data time-of-life
DRAM
More complex
Greater storage ability
Allow higher integration density

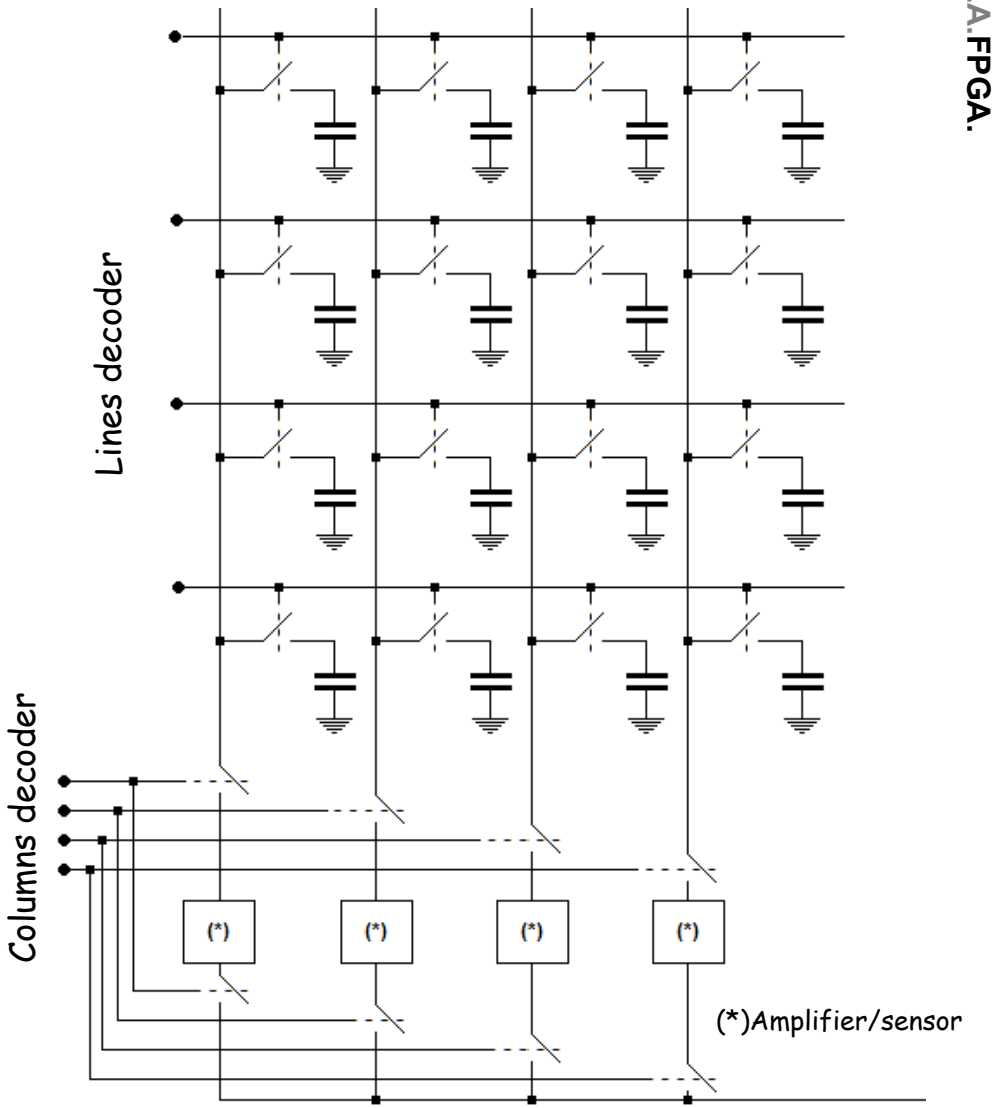
MEMORIES DRAM

-Stores the individual bits in "capacitors"

-The capacitor charging is not maintained indefinitely

-It requires a "refresh" circuit

each cell is visited periodically and its logical value is reset



MEMORIES **ROM**

The cell content is fixed

Unlike RAM does not require constant power supply

It is also a random access memory

ROM
(Read Only Memory)

Available in various technologies: PROM, EPROM, etc.

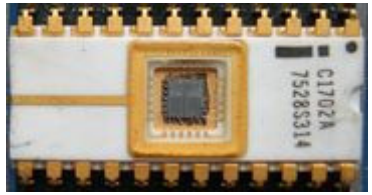
Used to store the BIOS content on PCs

-First type of manufacture programmed ROM (masked ROM's)

PROM (OTP)
Acquired 'virgin' and programmed only once by the user.

"Firmware" development process

EPROM
Which can be reprogrammed:
Write - electrical signal
Delete - U.V.



HYBRID MEMORIES

The advancement of technology has made blurring the boundary between **ROM** and **RAM**

Brings together the best of both worlds...

-They can be read and written repeatedly (such as **RAM**)

-Keeps the same content stored even in the absence of power (such as **ROM**)

Hybrid

FLASH

- High density
- Low cost
- Electrically Programmable
- Contrary to EEPROM can not erase Byte by Byte only sectors.

NVRAM

- SRAM+Batery
- Strategy + costly

EEPROM

- Internally similar to EPROM
- Deletes with electric signal.
- High cost and long writing cycles.

MEMORIES **ROM**: *Hardware Topics*

- Any Boolean function can be written as a "sum of products" or as a "product of sums"

-A ROM memory is a function that performs a mapping ADDRESS / DATA

-A ROM memory can be created as SdP or PdS

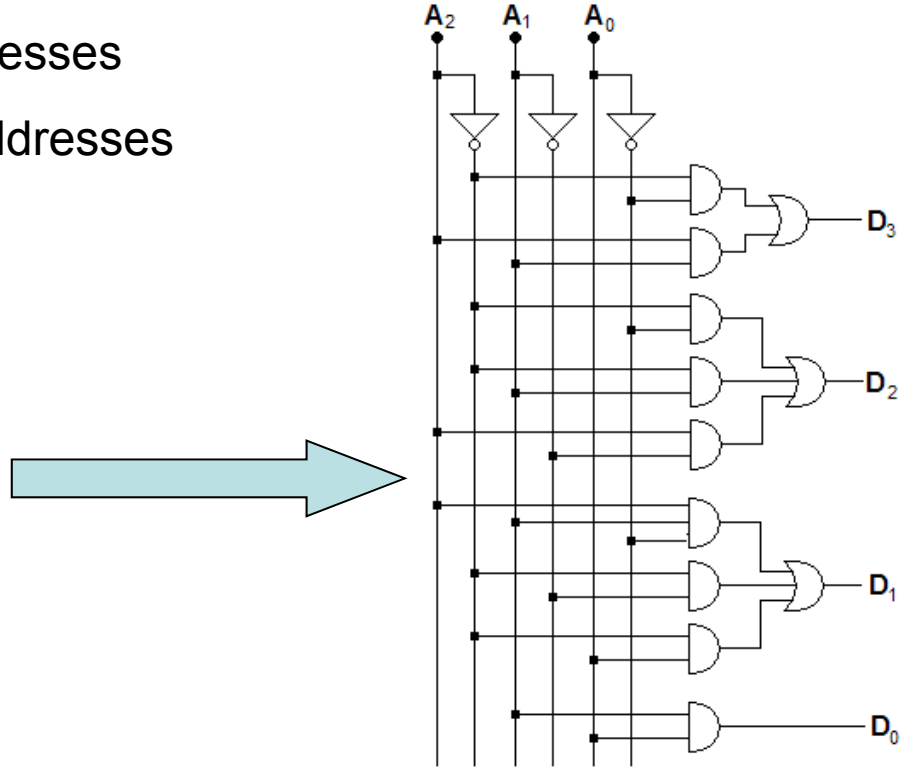
Ex. Create a memory ROM capable of storing 8 words of one nibble

Address != to each word => 8 addresses

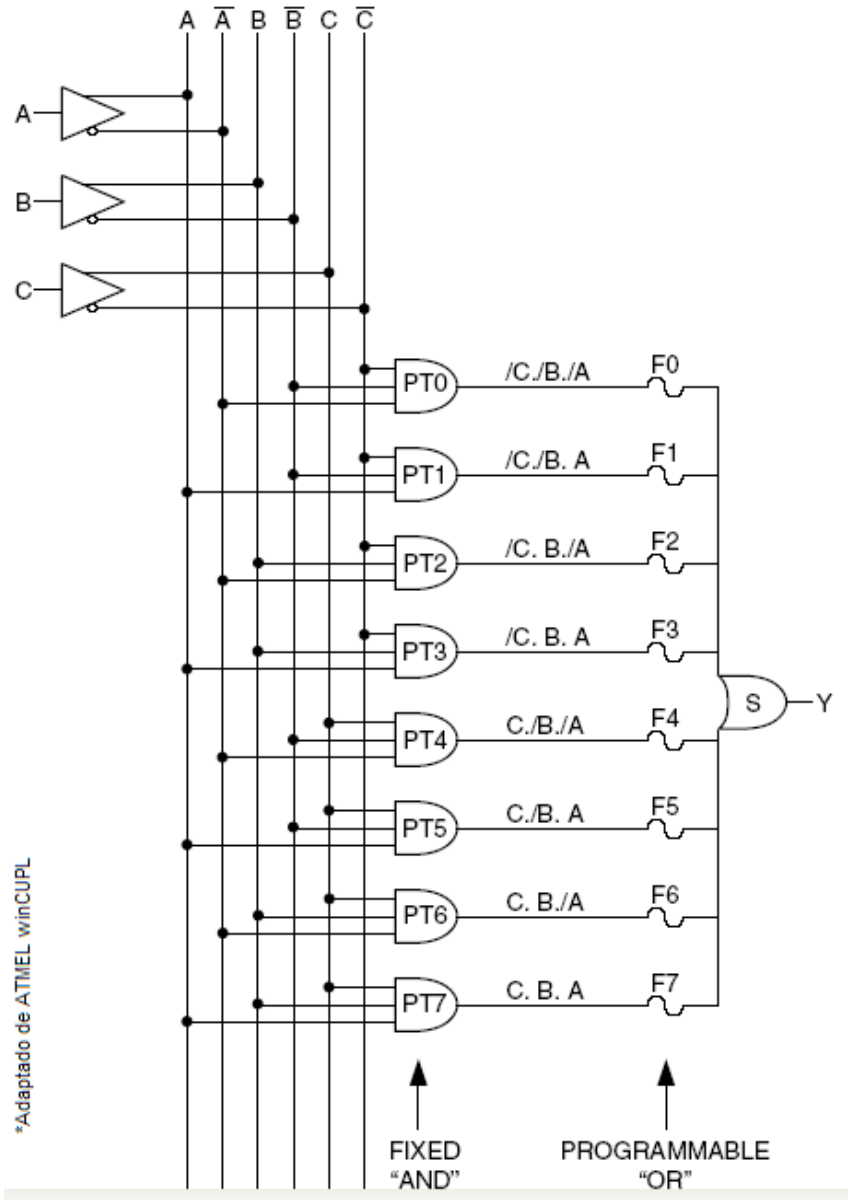
3 bits necessary to distinguish 8 addresses

ROM content

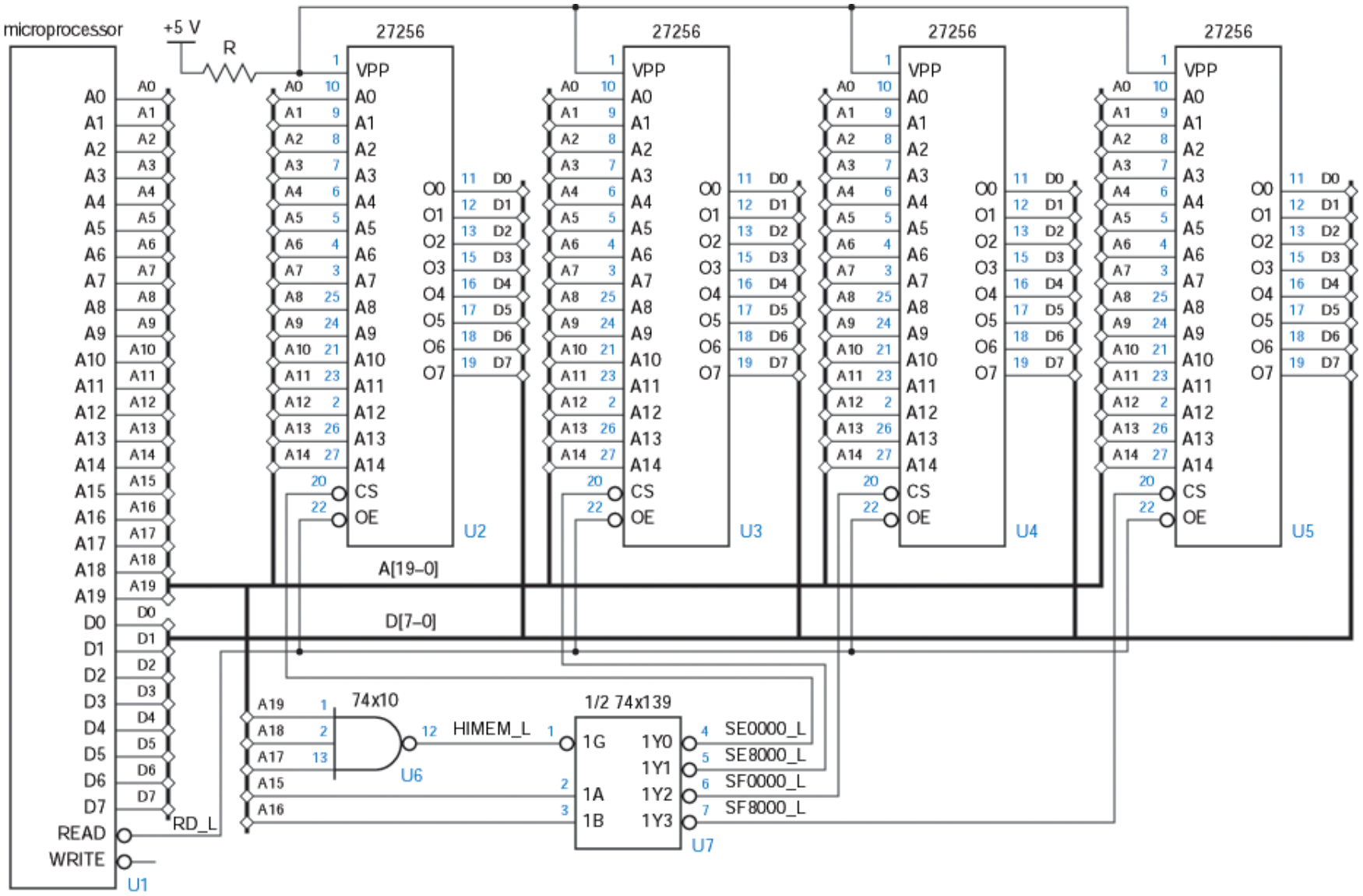
ADDRESS			DATA			
A2	A1	A0	D3	D2	D1	D0
0	0	0	1	1	1	0
0	0	1	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	1	1
1	0	0	1	1	0	0
1	0	1	1	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



From **ROM** to **PROM**...



Embedded system with uP...

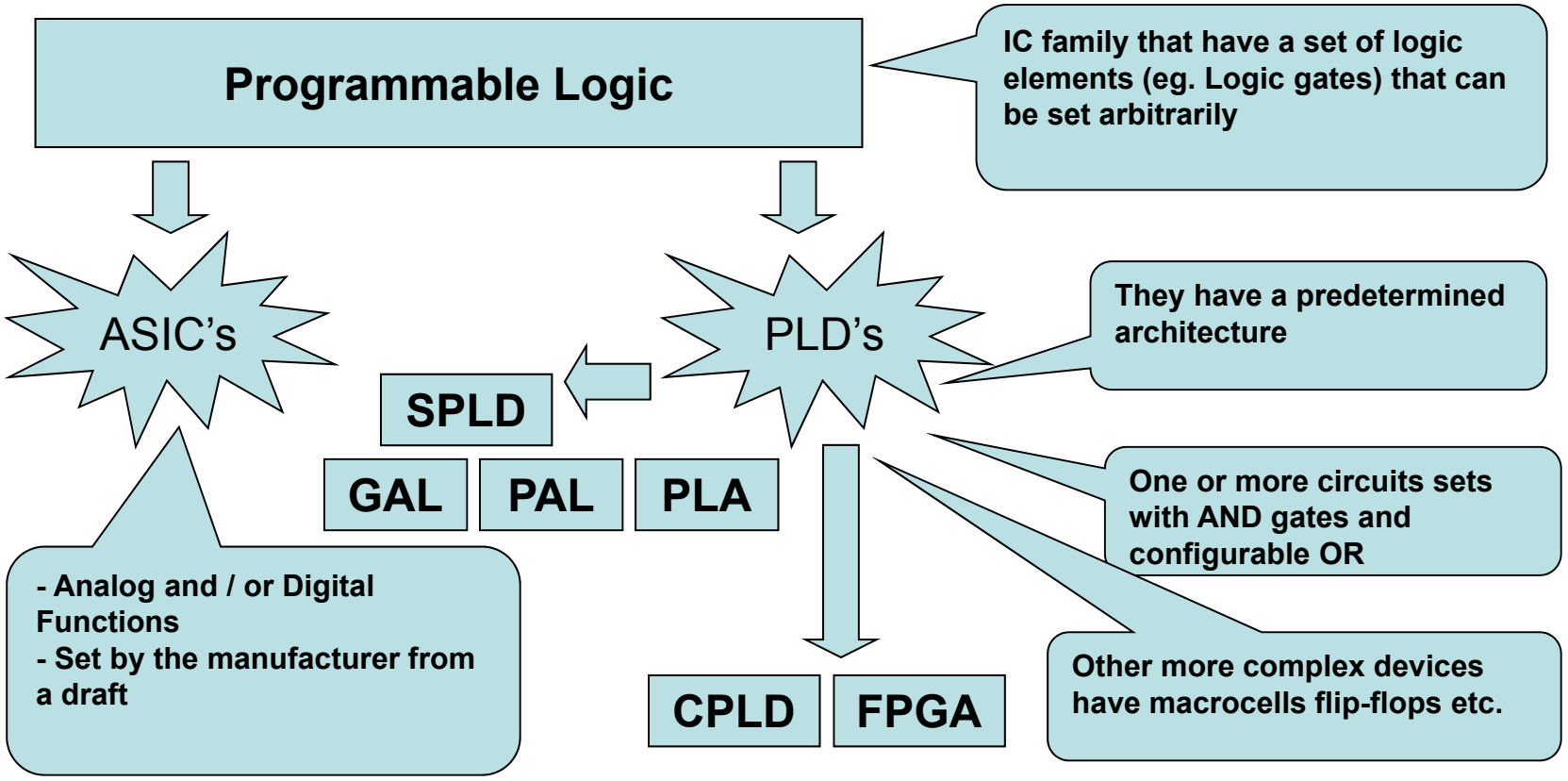


PROGRAMMABLE LOGIC

-Multiplexer as versatile way to implement arbitrary Boolean functions

-ROM's can also be used for this purpose

-Other strategies may involve, in addition to combinatorial logic, sequential logic



Unlike classical digital IC, where its function was rigid, in the PLD's the same IC can perform an infinite number of functions!

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

SPLD

In sharp decline, they are already few manufacturers who still produce SPLDs
-Lattice
-Cypress
-AMD
-PHILIPS
-TI

PAL's

PAL...

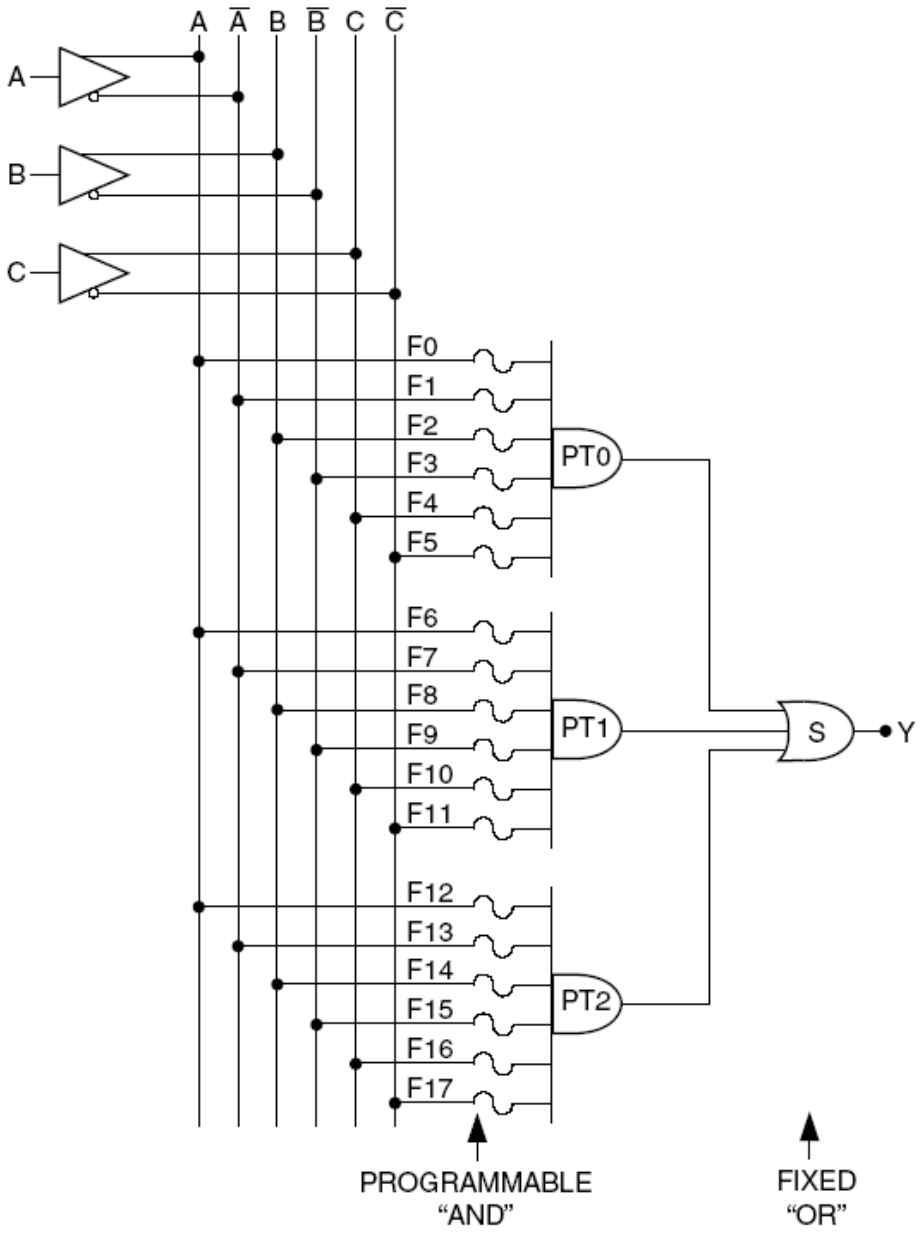
-Programmable Array Logic

-Implementation of the disjunctive form

-Only the AND array is programmable

-Architecture type of PLD most popular

-Only the each minimum term variables can be programmed



PLA's

PLA...

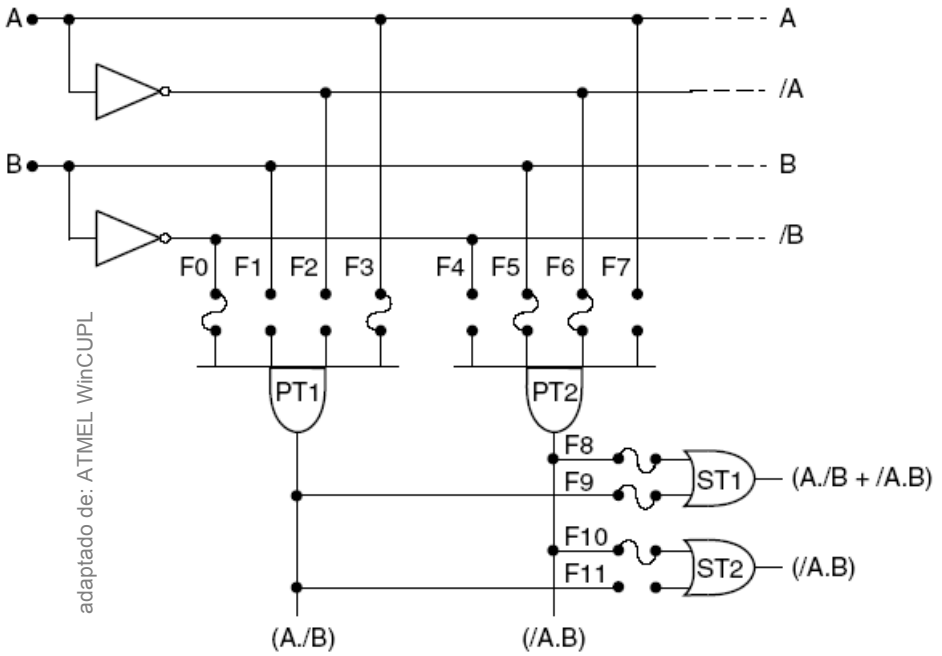
-Programmable Logic Array

-It has both programmable AND and OR terms

-Greater flexibility

-Typically they have feedback between the output of each OR and the input of set AND

-It can implement asynchronous state machines



adaptado de: ATMEL WinCUPL

examples

$F_0 = A\,B\,C$

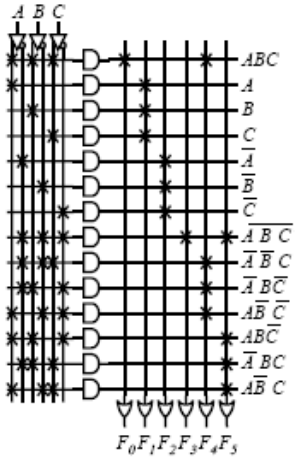
$F_1 = A + B + C$

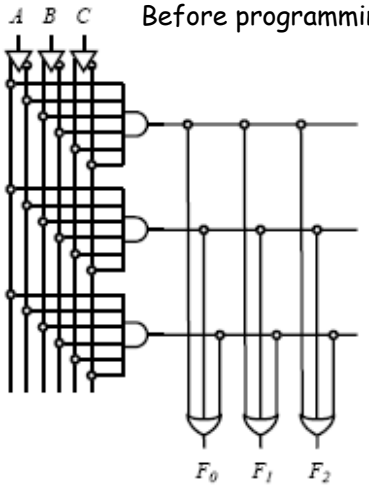
$F_2 = \overline{(A\,B\,C)} = \overline{A} + \overline{B} + \overline{C}$

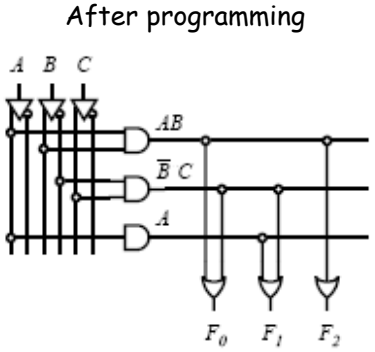
$F_3 = \overline{(A + B + C)} = \overline{A}\ \overline{B}\ \overline{C}$

$F_4 = A \oplus B \oplus C = \overline{A}\ \overline{B}\ C + \overline{A}\ B\,\overline{C} + A\,\overline{B}\ \overline{C} + A\,B\,C$

$F_5 = \overline{(A \oplus B \oplus C)} = A\,B\,\overline{C} + \overline{A}\ B\,C + A\,\overline{B}\ C + \overline{A}\ \overline{B}\ \overline{C}$







adaptado de: Luís Laus, Electrónica Digital

$F_0 = AB + \overline{B}\ C$

$F_1 = A + \overline{B}\ C$

$F_2 = AB$

2ª Generation of PAL's: GAL's

GAL (Generic Array Logic)...

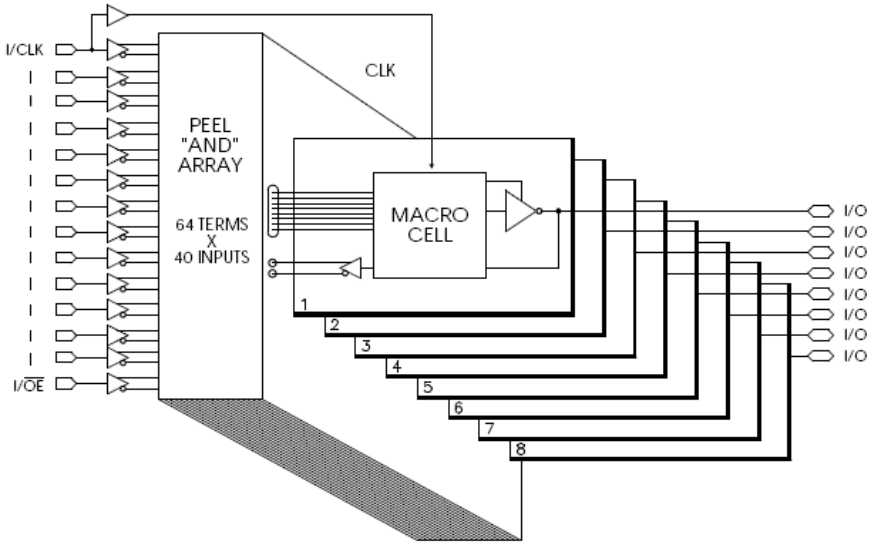
-Developed by LATTICE

-Able to implement both combinatorial and sequential functions (counters, registers, etc.)

-Its speed competes with the 74xx series of TTL

-Can be **reprogrammed** *(~ 100X)*

-Use of "macro-cells" at output



AR –asynchronous RESET
SP – synchronous PRESET

S0 and S1 are "fuses"

REGISTERED MODE

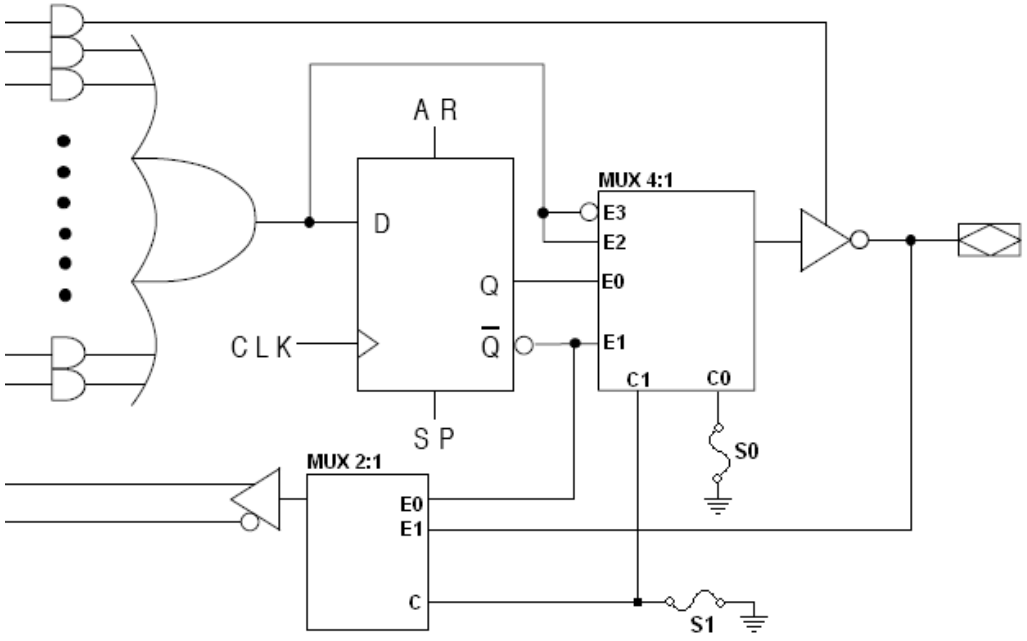
S1=0 S0=X

COMBINATIONAL MODE

S1=1 S0=X

NOTE: *The outputs can also be inputs (how?)*

MACRO-CELL

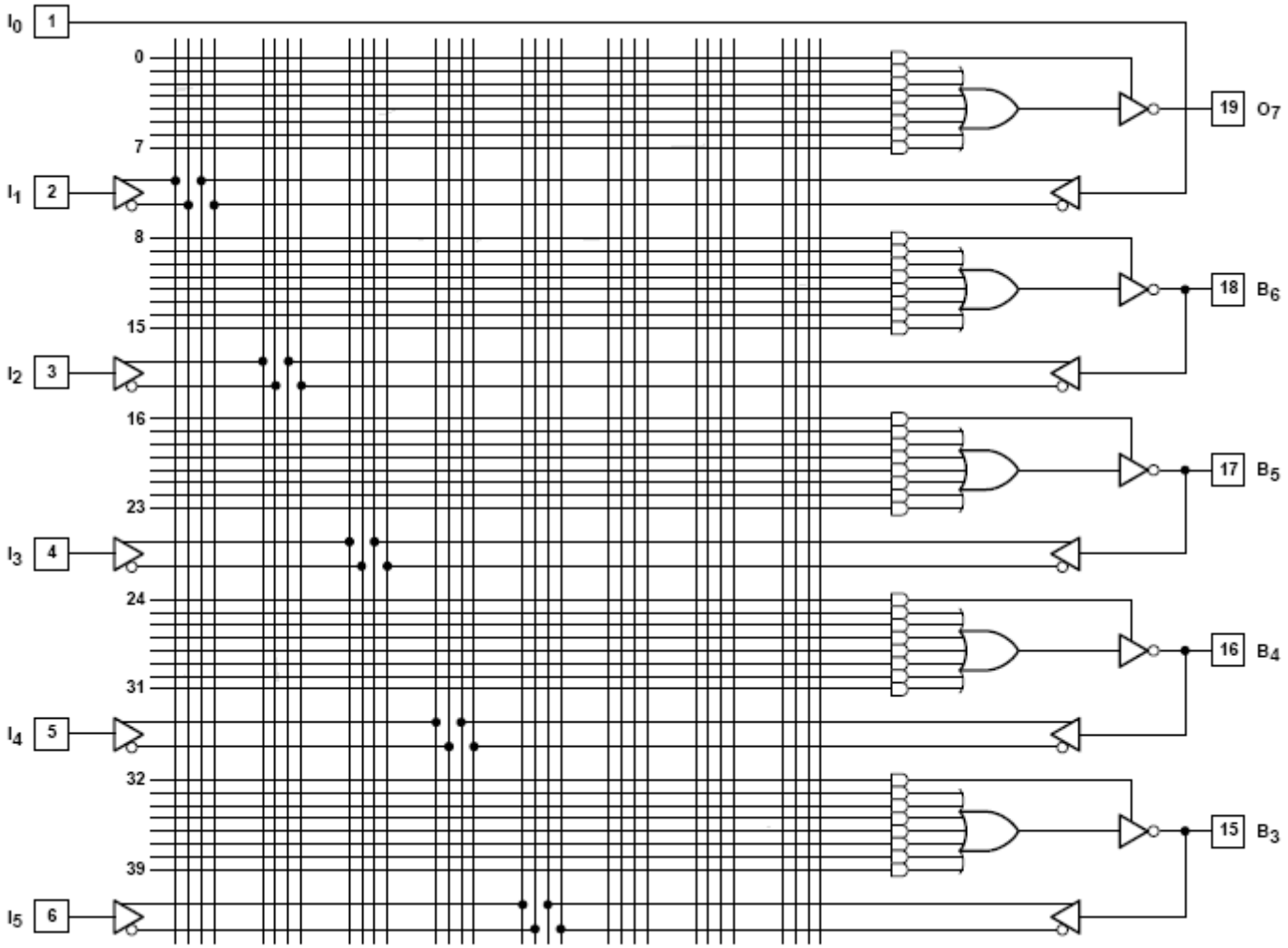


Application Example: PAL 16L8

$x(A,B,C,D) = \Sigma(7,8,9,10,11,12,13,14,15)$
$y(A,B,C,D) = \Sigma(0,2,3,4,5,6,7,8,10,11,15)$
$z(A,B,C,D) = \Sigma(1,2,8,12,13)$
$w(A,B,C,D) = \Sigma(2,12,13)$

after simplification…

$x(A,B,C,D) = A + BCD$
$y(A,B,C,D) = A'B + CD + B'D'$
$z(A,B,C,D) = w + AC'D' + A'B'C'D$
$w(A,B,C,D) = ABC' + A'B'CD'$

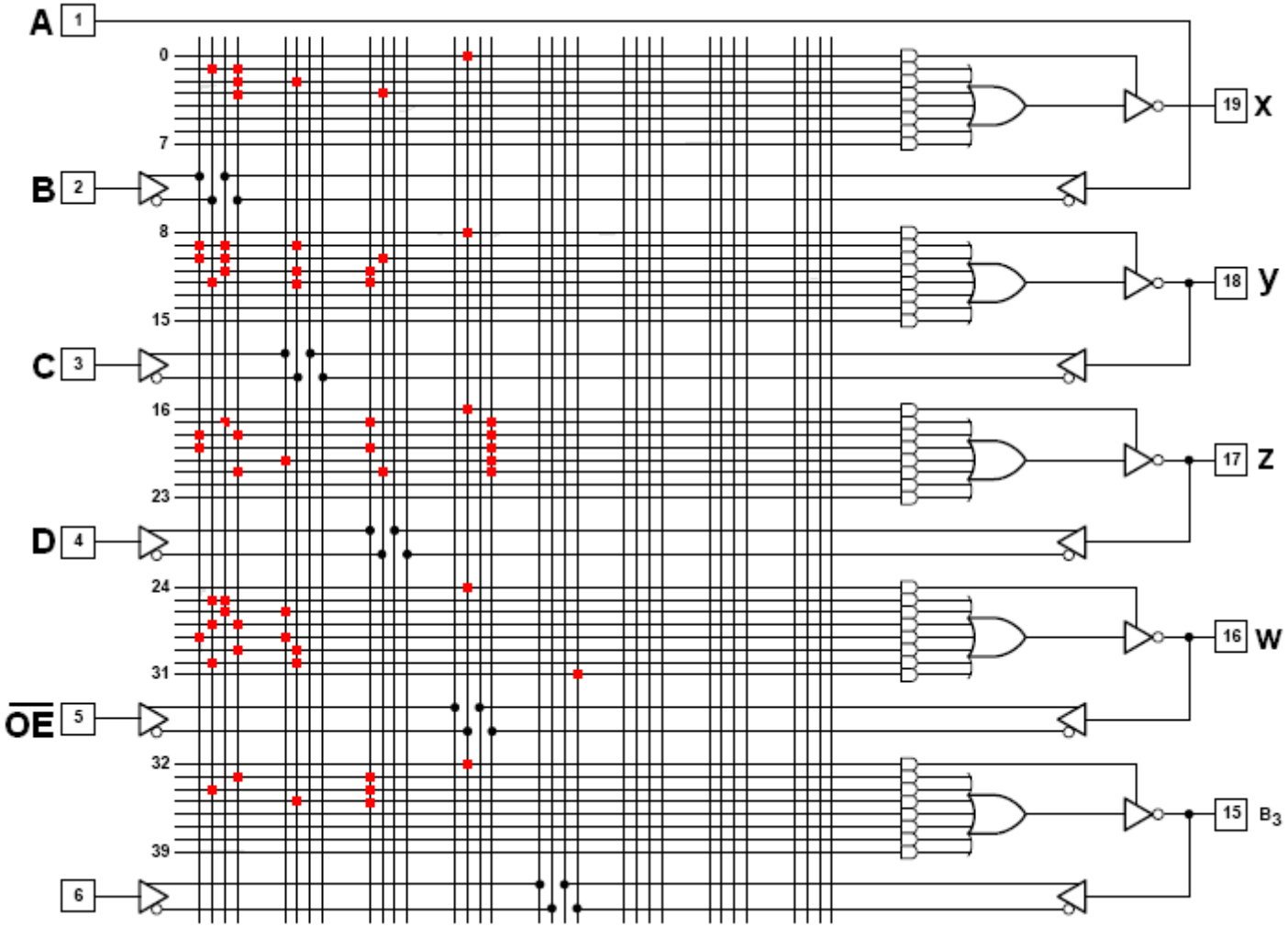


$$x(A,B,C,D) = \overline{\overline{A}\cdot\overline{B} + \overline{A}\cdot\overline{C} + \overline{A}\cdot\overline{D}}$$

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

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

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



How do PLDs store information?

In a PLD the memory is used for storing the pattern which is given to the IC during programming. The methods used for programming may be of the following type

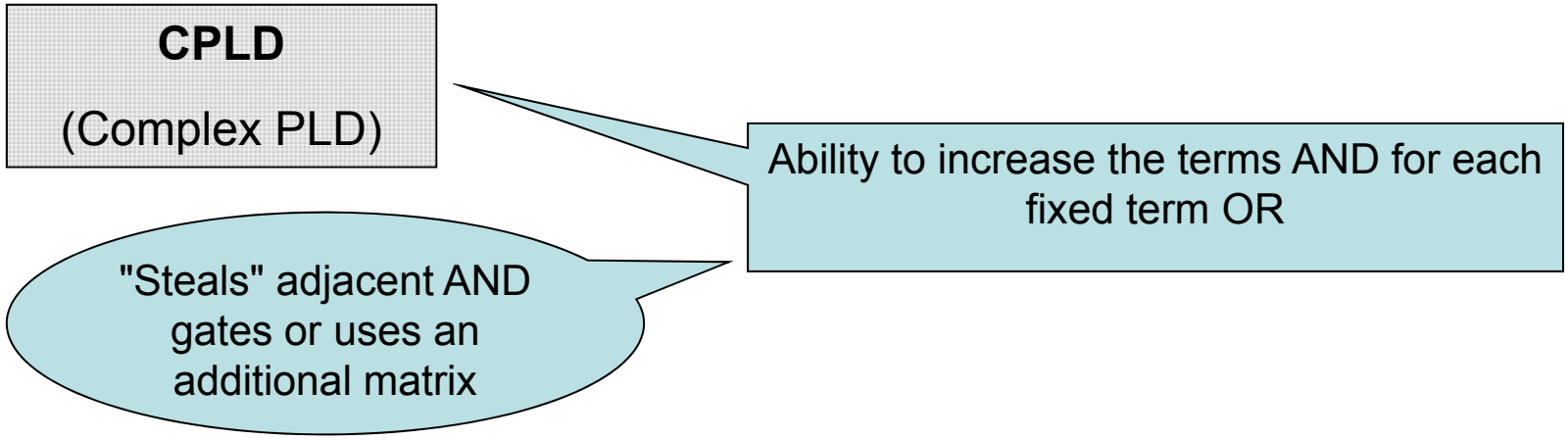
- Silicon Anti-fuses (PROM)
- SRAM
- Cells EPROM, EEPROM
- Flash memory

CPLD's

The PALs and GALs are available only in small sizes, equivalent to a few hundred logic gates.

For additional logic circuits can be used **Complex PLD** or **CPLD**s.

It contains the equivalent of some PALs connected by programmed interconnections.

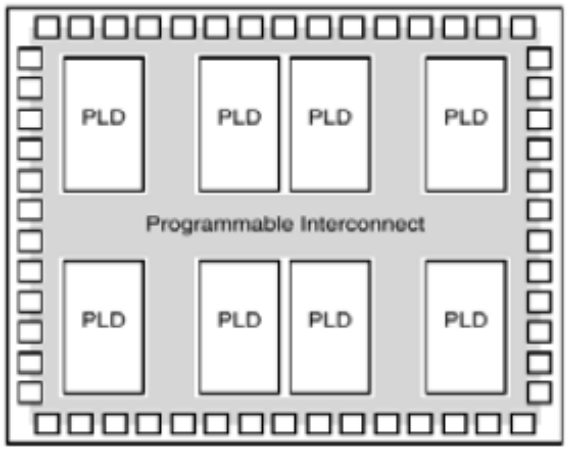


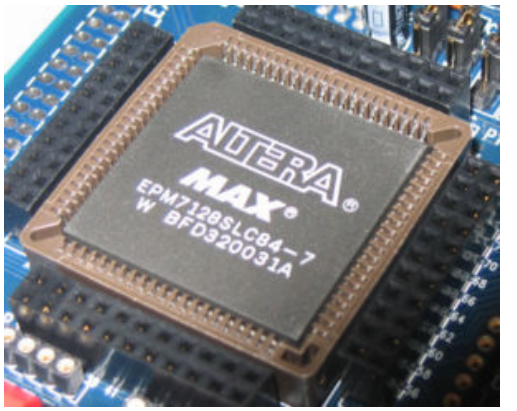
The internal PLDs are designated FB's or **CFB's** (Configurable Functional Bocks)

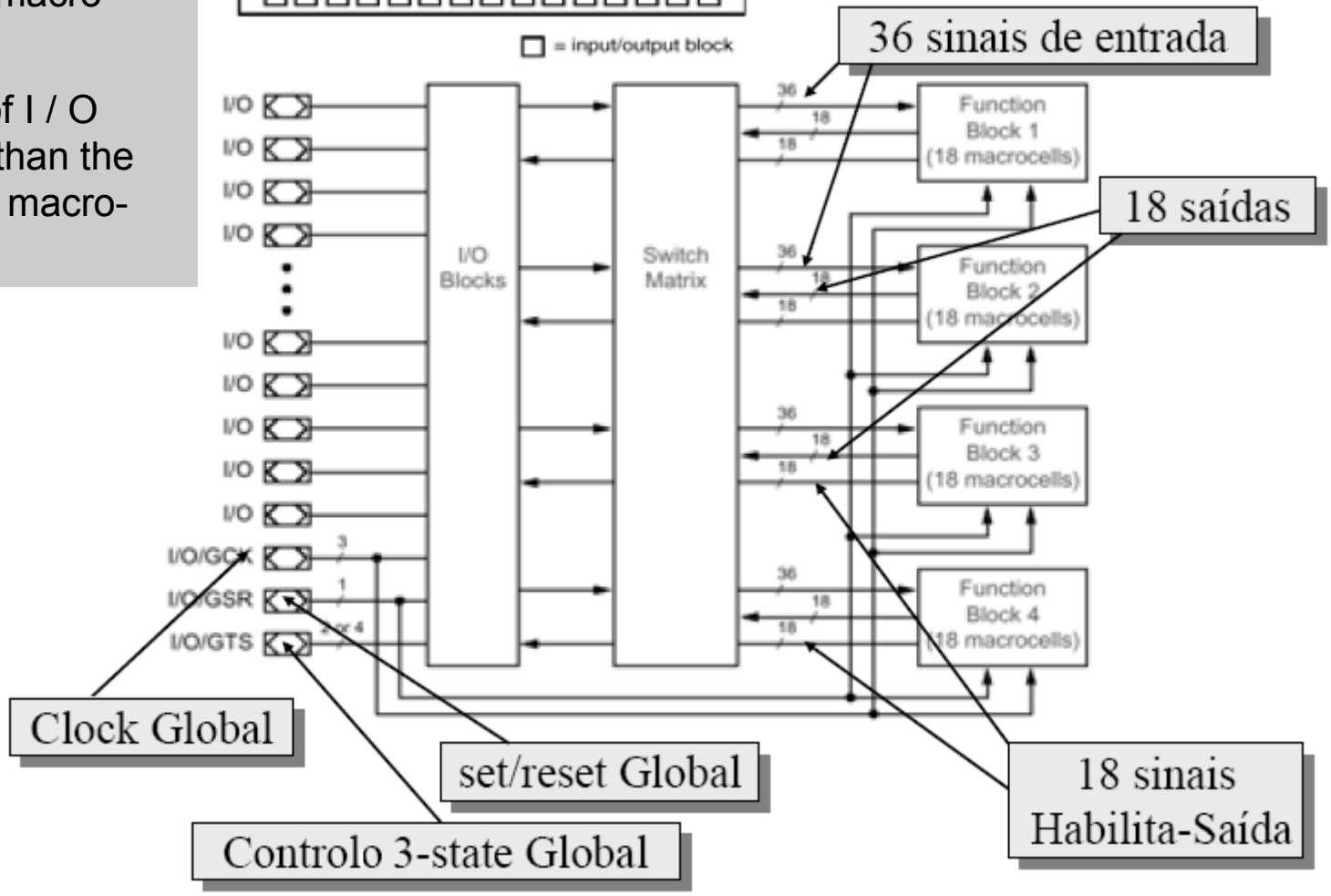
XILINX ‘9500

- Each **CFB** has 36 entries and 18 macro-cells

- The number of I / O pins is smaller than the total number of macro-cells







FPGA's

The device comprises a set of logic cells or logic blocks allocated in the form of a matrix

FPGA
(Field Prog. Gate Arrays)

-High Density of ports
-High performance
-High number of I/O
-Flexible interconnection scheme

It emerged in 1985 with Xilinx
Other companies:
Actel, Altera, Plessey, Plus, AMD, etc.

Internal Structure

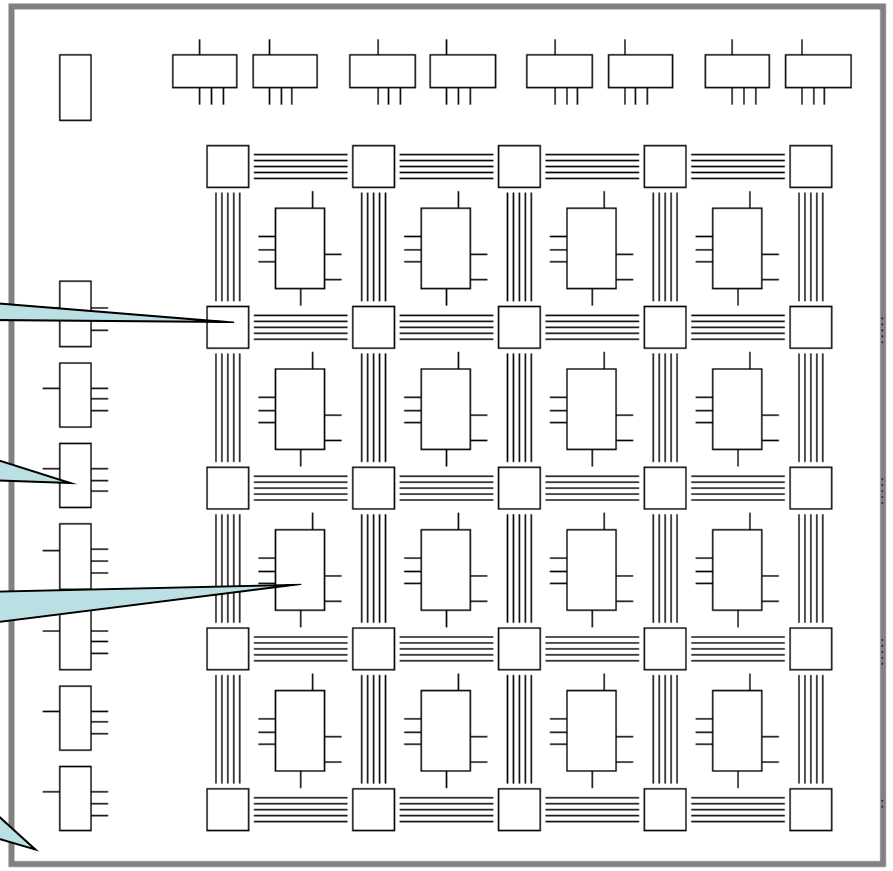
- **CLB**'s (Configurable Logic Block)
- **IOB**'s (In/Out Block)
- **SB**'s (Switch Box)

Switch Box (SB)

Each IOB can be configured as input, output, I/O. (Buffered).

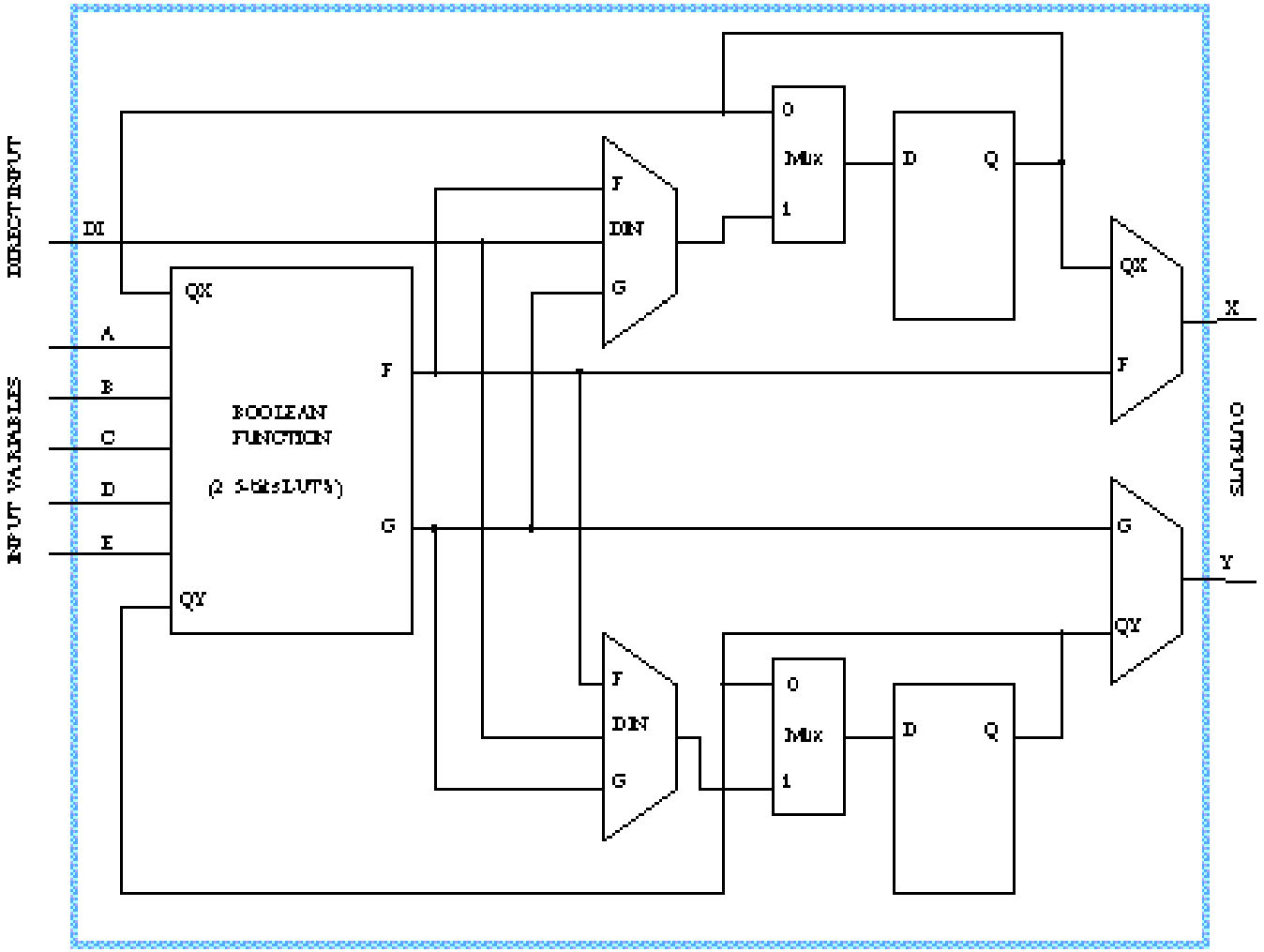
Each CLB has programmable combinatorial logic and memory registers

It has a CLB's matrix [configurable logic blocks] surrounded by a ring of I/O blocks (IOB's)

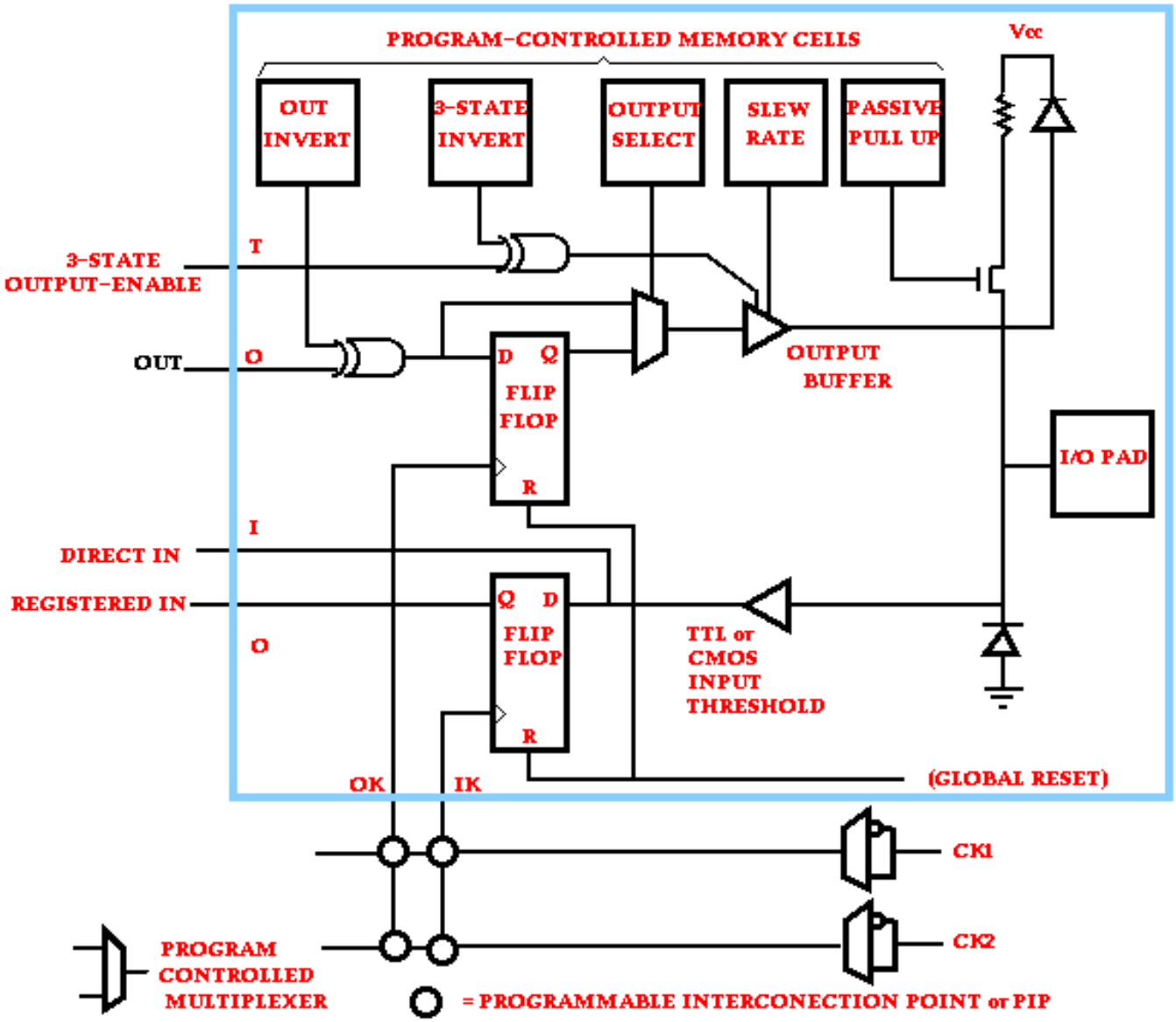


CLB's

Xilinx XC3000

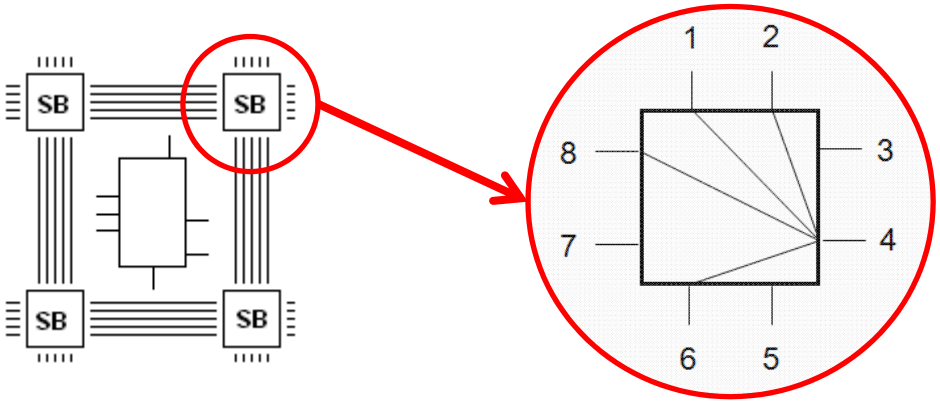


IOB's


PROGRAM-CONTROLLED MEMORY CELLS
Vcc
OUT INVERT
3-STATE INVERT
OUTPUT SELECT
SLEW RATE
PASSIVE PULL UP
3-STATE OUTPUT-ENABLE
T
OUT
O
D Q
FLIP FLOP
R
OUTPUT BUFFER
I/O PAD
I
DIRECT IN
REGISTERED IN
O
Q D
FLIP FLOP
R
TTL or CMOS INPUT THRESHOLD
OK
IK
(GLOBAL RESET)
CK1
CK2
PROGRAM CONTROLLED MULTIPLEXER
= PROGRAMMABLE INTERCONECTION POINT or PIP

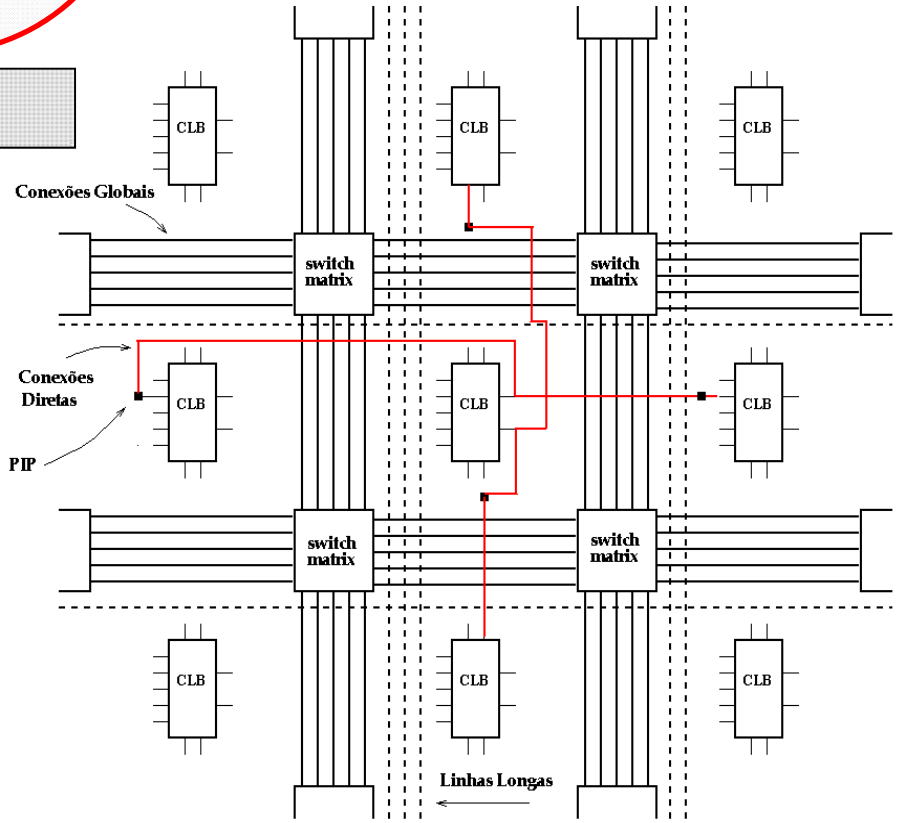
Switch Box

Allows the interconnection between the **CLB**'s through the **routing** channels



Routing

Interconnection between the blocks is made through a network of two metal layers.
The physical connections between the wires are made:
.with pass transistors controlled by bits of memory (PIP),
.with interconnect switches (Switch Matrix).

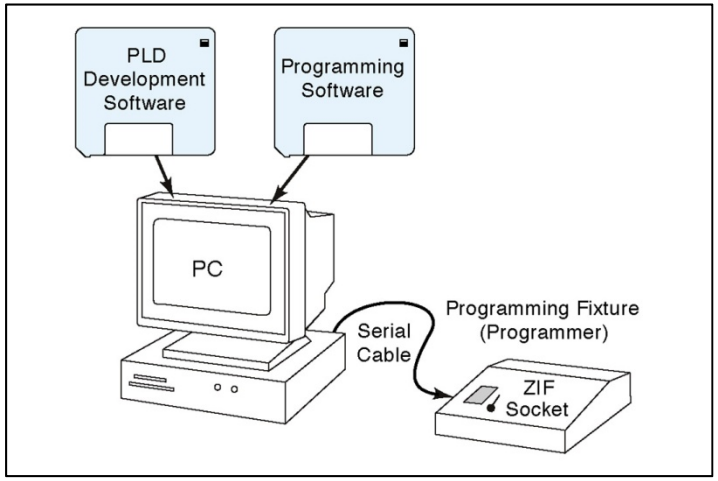


Programming SPLDs, CPLDs and FPGAs

While it is possible to design a simple PLD manually, in case of more complex devices such as the CPLD's and FPGA's is mandatory to use CAD tools for your project.

PAL's:
Accept input in a known file format for file "JEDEC" [**J**oint **E**lectron **D**evice **E**ngineering **C**ouncil].

Several languages for logical compilers:
CUPL, ABEL, HDL e VHDL, etc.



- In CPLDs and FPGAs, due to the used encapsulation type, the programming should be "in circuit".
- Each manufacturer has a proprietary name for this programming system.
- Lattice Semiconductor calls it programming "in-system" (ISP).
- Standardization through the Joint Test Action Group (JTAG).