

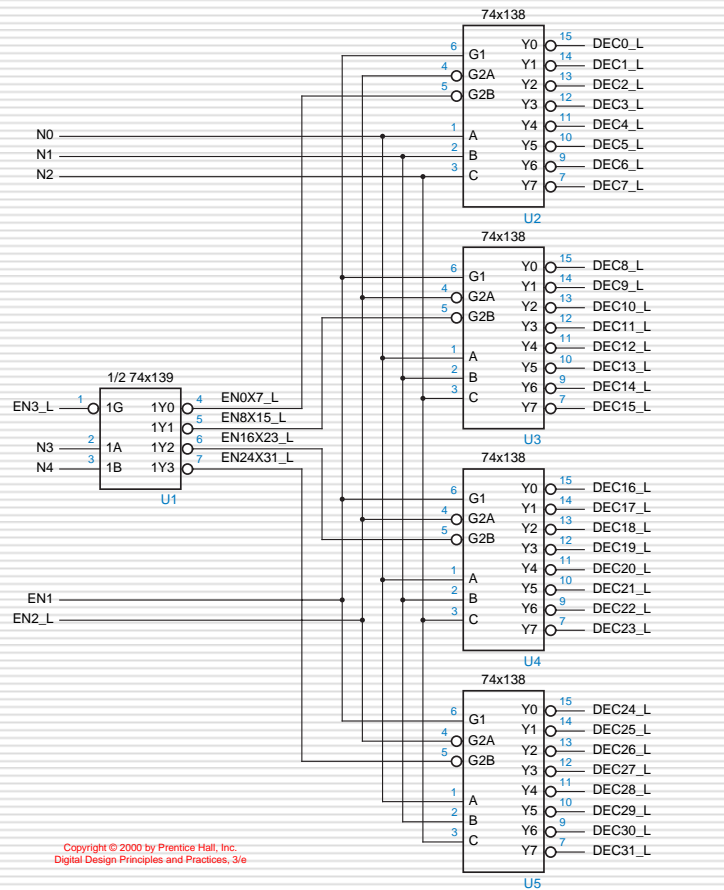
# INF1500 :

## Logique des systèmes numériques

---

### □ Cours 7: Implémentations

# Composantes



Copyright © 2000 by Prentice Hall, Inc.  
Digital Design Principles and Practices, 3/e

## SN54HC139, SN74HC139 DUAL 2-LINE TO 4-LINE DECODERS/DEMULPLEXERS

SCLS108B - DECEMBER 1982 - REVISED MAY 1997

- Designed Specifically for High-Speed Memory Decoders and Data Transmission Systems
- Incorporate Two Enable Inputs to Simplify Cascading and/or Data Reception
- Package Options Include Plastic Small-Outline (D), Thin Shrink Small-Outline (PW), and Ceramic Flat (W) Packages, Ceramic Chip Carriers (FK), and Standard Plastic (N) and Ceramic (J) 300-mil DIPs

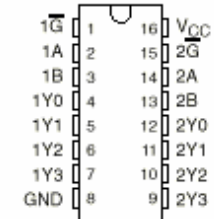
### description

The 'HC139 are designed for high-performance memory-decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems, these decoders can minimize the effects of system decoding. When employed with high-speed memories utilizing a fast enable circuit, the delay time of these decoders and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoders is negligible.

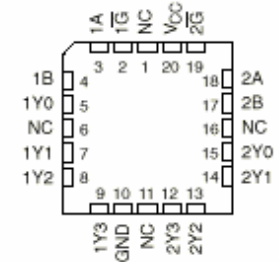
The 'HC139 comprise two individual 2-line to 4-line decoders in a single package. The active-low enable ( $\bar{G}$ ) input can be used as a data line in demultiplexing applications. These decoders/demultiplexers feature fully buffered inputs, each of which represents only one normalized load to its driving circuit.

The SN54HC139 is characterized for operation over the full military temperature range of  $-55^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ . The SN74HC139 is characterized for operation from  $-40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ .

SN54HC139... J OR W PACKAGE  
SN74HC139... D, N, OR PW PACKAGE  
(TOP VIEW)



SN54HC139... FK PACKAGE  
(TOP VIEW)



NC - No internal connection

FUNCTION TABLE

$\bar{G}$	INPUTS		OUTPUTS			
	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

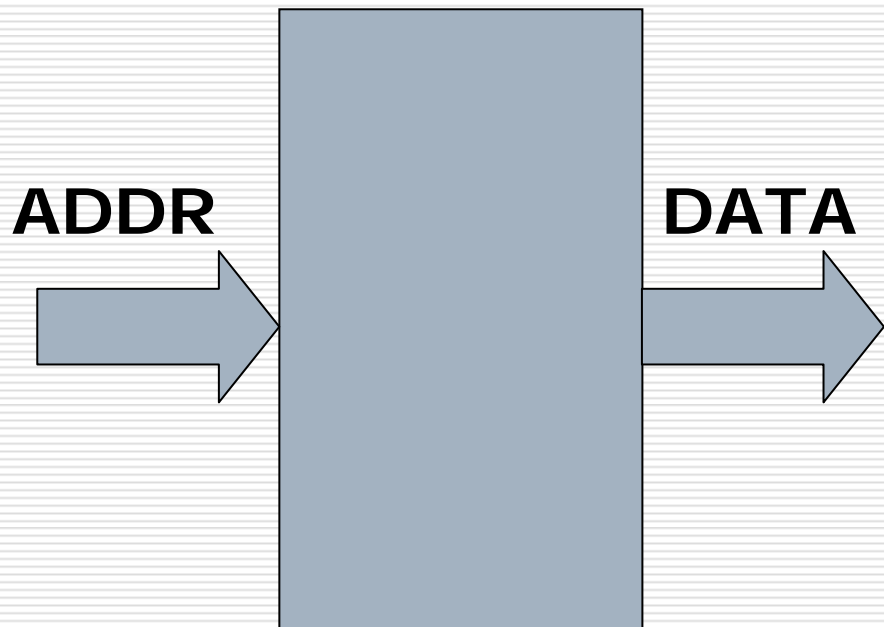
## ROM (Read Only Memory)

---

- ❑ Une mémoire ROM consiste en :
- ❑ Un décodeur avec  $n$  signaux d'entrée;
- ❑ Un réseau OU programmé (ou programmable une fois) avec  $2n$  mots de  $m$  bits chacun ;
- ❑  $m$  signaux de sortie.
- ❑ Le réseau programmé comporte  $2n$  lignes de  $m$  colonnes chacune. À l'intersection de chaque ligne avec une colonne on trouve un élément électronique à mémoire.
- ❑ Une fois programmée, la ROM de  $2n$  mots de  $m$  bits peut générer  $m$  fonctions de  $n$  variables simultanément.

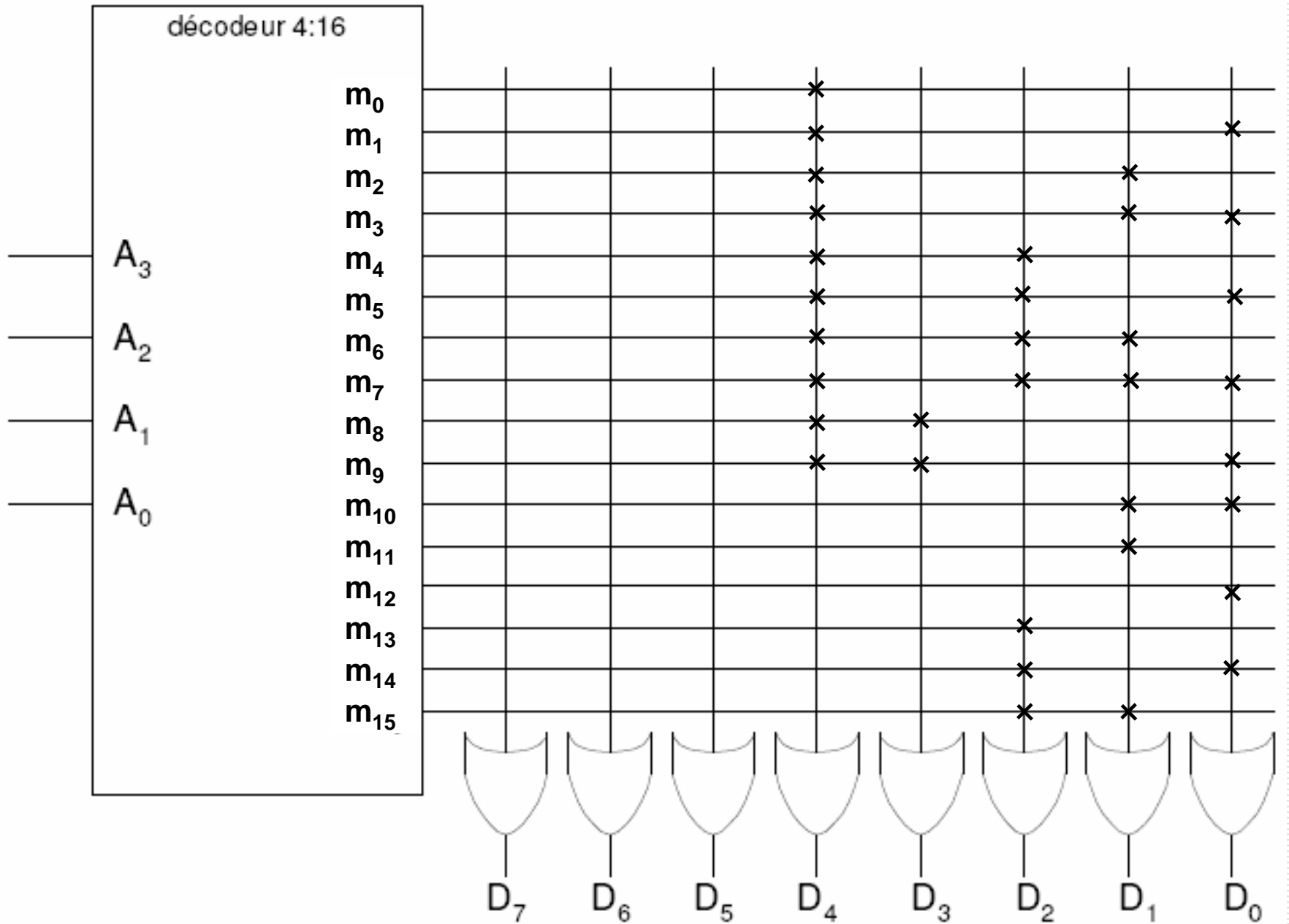
# ROM (Read Only Memory)

---



#	A	B	C	D	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

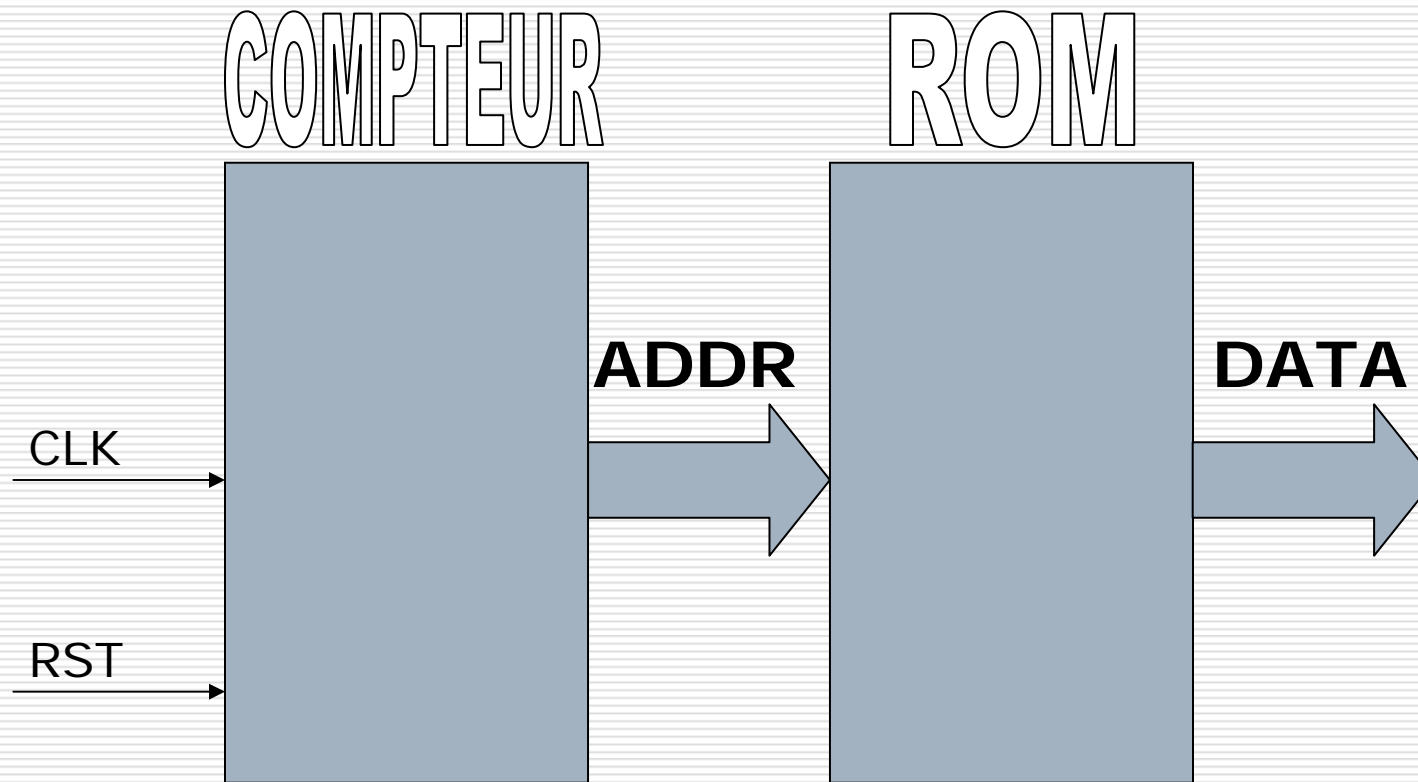
# ROM 16×8 bits



A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Hexa-décimal	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	<b>0</b>	0	1	1	0	0	0	0
0	0	0	1	<b>1</b>	0	1	1	0	0	0	1
0	0	1	0	<b>2</b>	0	1	1	0	0	1	0
0	0	1	1	<b>3</b>	0	1	1	0	0	1	1
0	1	0	0	<b>4</b>	0	1	1	0	1	0	0
0	1	0	1	<b>5</b>	0	1	1	0	1	0	1
0	1	1	0	<b>6</b>	0	1	1	0	1	1	0
0	1	1	1	<b>7</b>	0	1	1	0	1	1	1
1	0	0	0	<b>8</b>	0	1	1	1	0	0	0
1	0	0	1	<b>9</b>	0	1	1	1	0	0	1
1	0	1	0	<b>A</b>	1	0	0	0	0	0	1
1	0	1	1	<b>B</b>	1	0	0	0	0	1	0
1	1	0	0	<b>C</b>	1	0	0	0	0	1	1
1	1	0	1	<b>D</b>	1	0	0	0	1	0	0
1	1	1	0	<b>E</b>	1	0	0	0	1	0	1
1	1	1	1	<b>F</b>	1	0	0	0	1	1	0

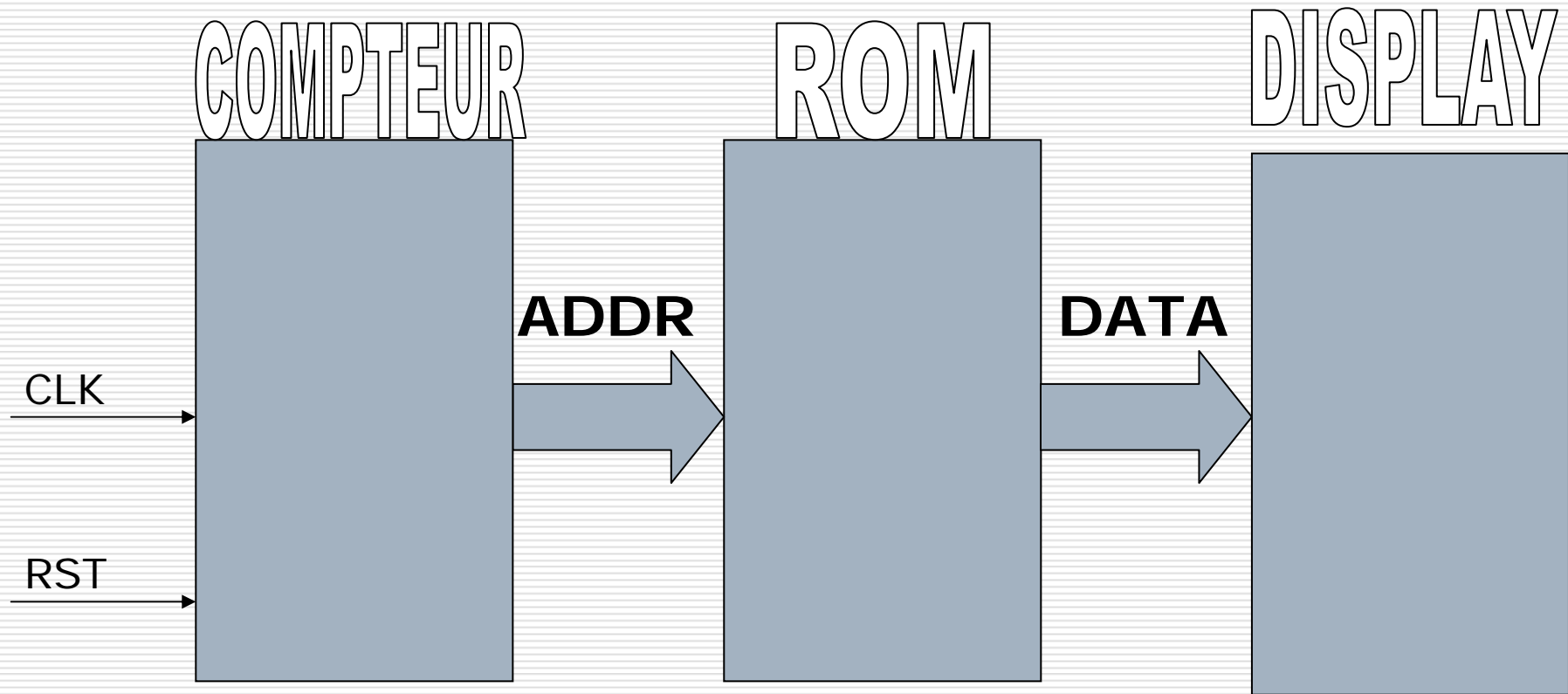
# ROM (Read Only Memory) – Exemple: Changer la séquence normale d'un compteur

---



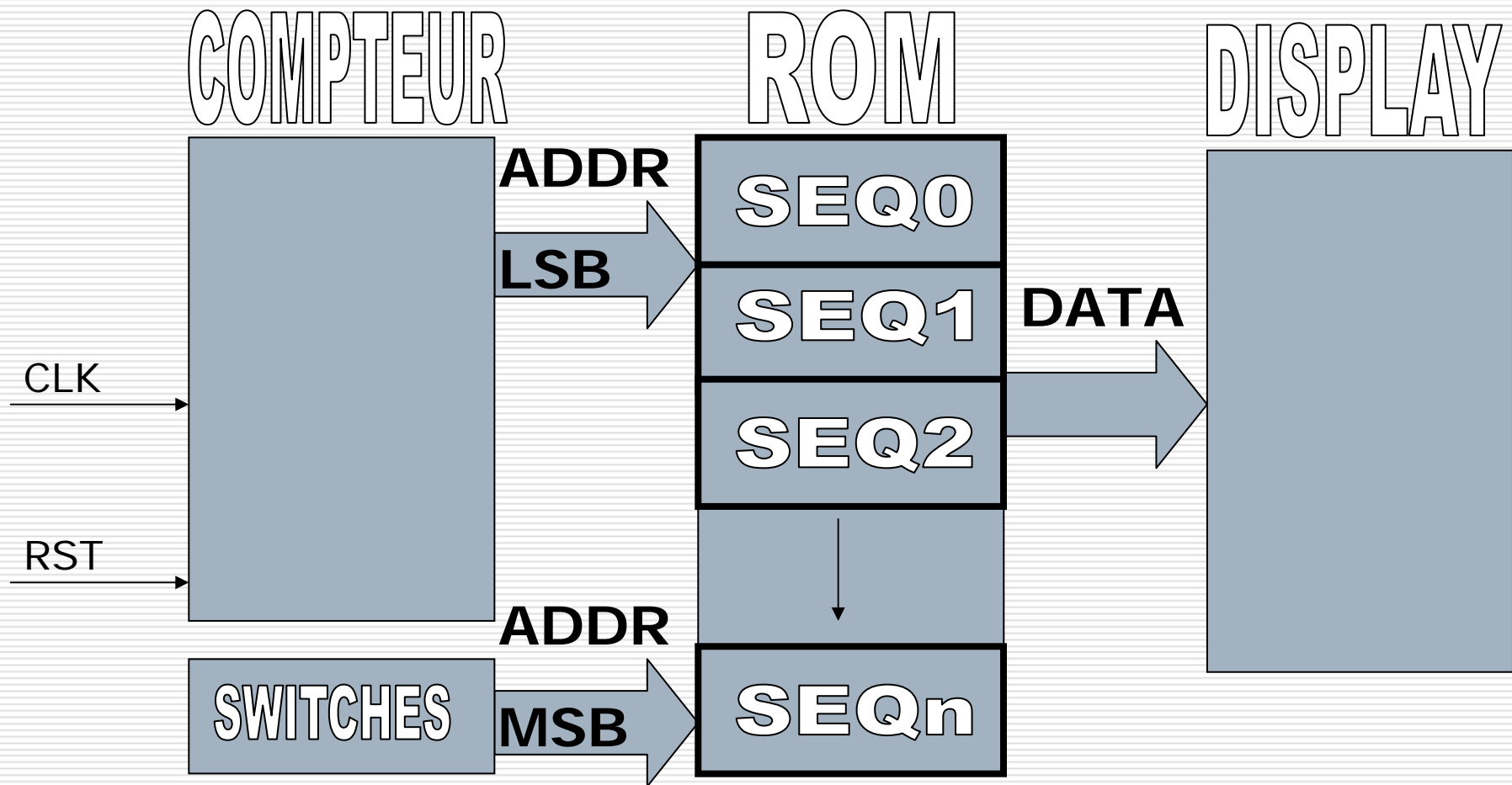
# ROM (Read Only Memory) – Exemple: Changer la séquence normale d'un compteur avec DISPLAY

---



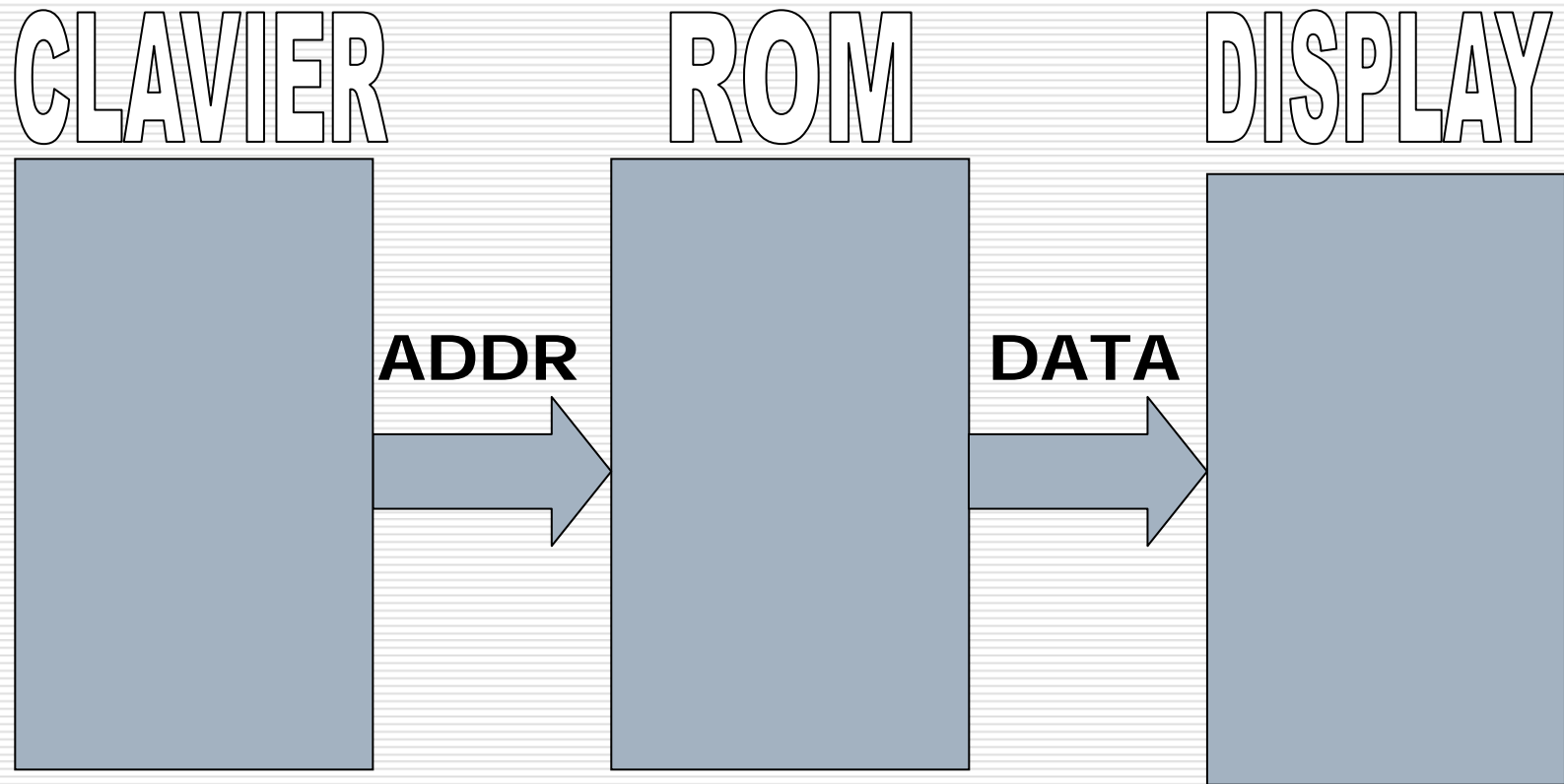


# ROM (Read Only Memory) – Exemple: Séquences d'effets spéciaux



# ROM (Read Only Memory) – Exemple: CLAVIER avec DISPLAY

---



## ROM (Read Only Memory) - Suite

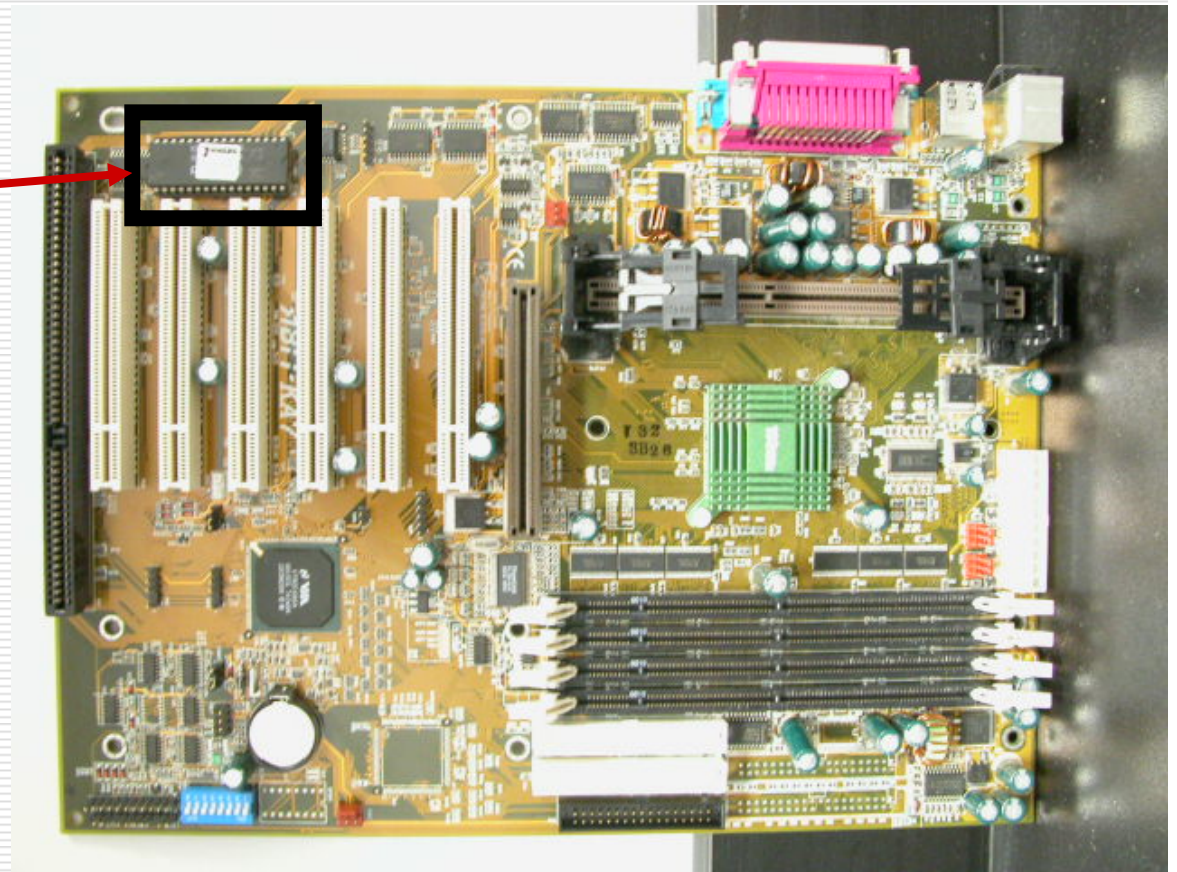
---

- ❑ Il existe plusieurs sortes de mémoire ROM :
- ❑ PROM : *Programmable Read Only Memory*, programmable une seule fois;
- ❑ EPROM : *Erasable Programmable Read Only Memory*, programmable à plusieurs reprises, et effaçable à l'aide de rayons ultraviolets (facile à reconnaître avec la petite fenêtre); et
- ❑ EEPROM : *Electrically Erasable Programmable Read Only Memory*, programmable à plusieurs reprises, et effaçable à l'aide d'impulsions électriques.
- ❑ + *NVRAM, FLASH RAM, etc...*

## Autres types de mémoire que l'on retrouve dans un ordinateur (volatile vs. non-volatile)

---

- ROM,
- PROM,
- EPROM,
- EEPROM,
- FLASH
- ...

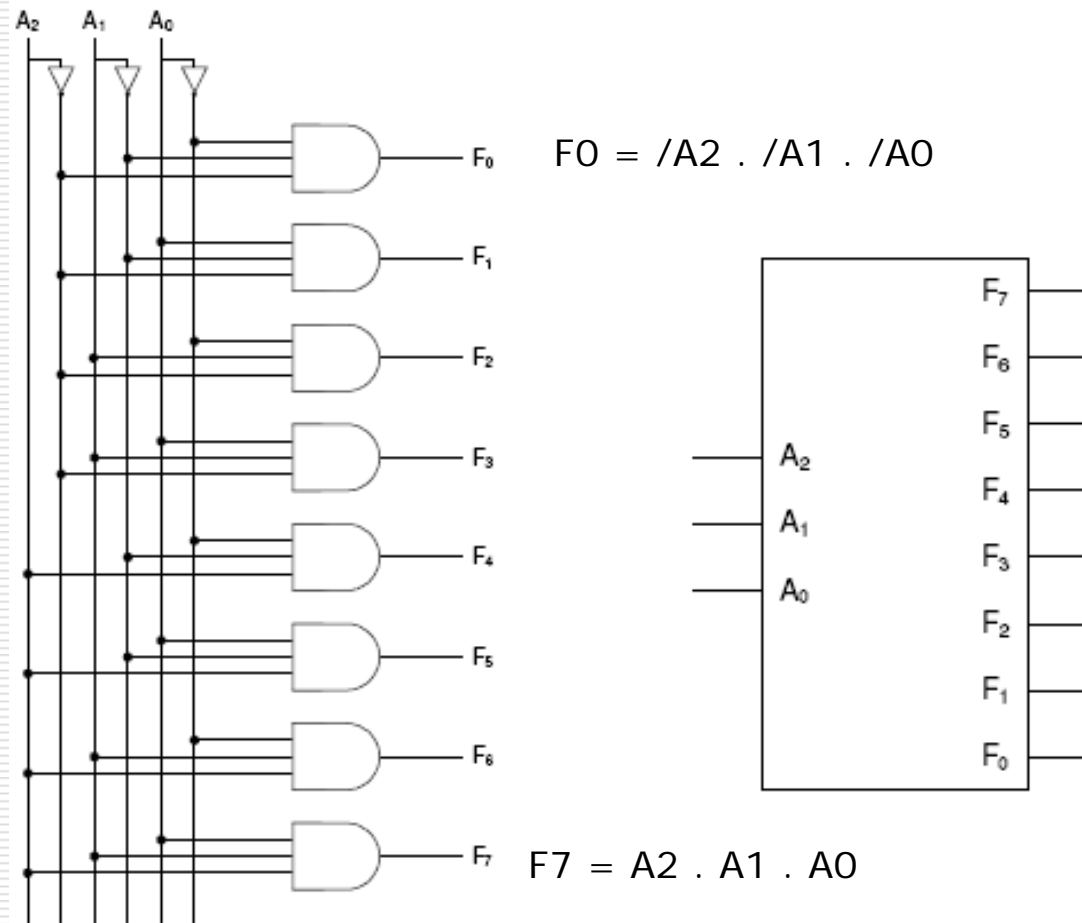


## Choix – Electronique programmable

---

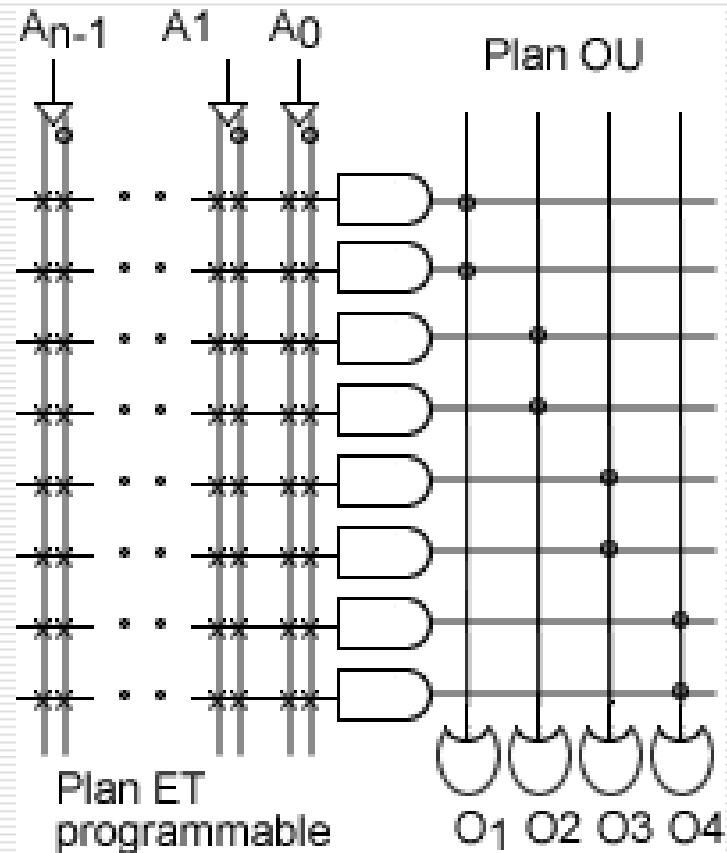
- (PLA) PAL
- GAL
- CPLD
- FPGA

# Circuits combinatoires (PAL)

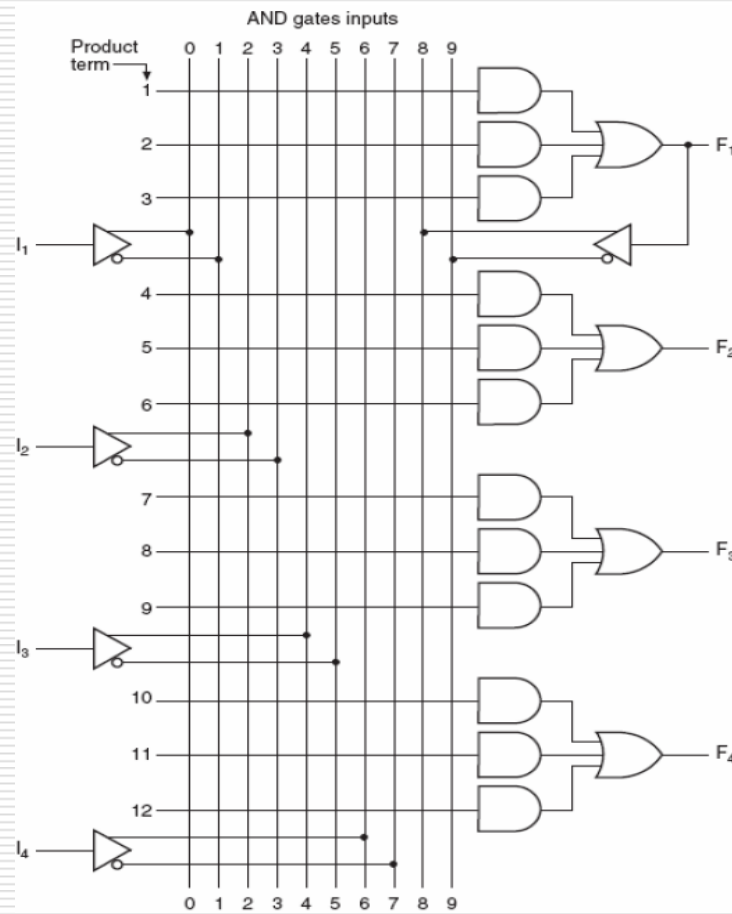


# PAL (*Programmable Array Logic*)

- Logique restreinte (par rapport aux CPLDs et FPGAs).
- C'est le plus populaire des simple PLDs.
- Programmable une seule fois par survoltage.
- Plan ET programmable.
- Plan OU fixe.

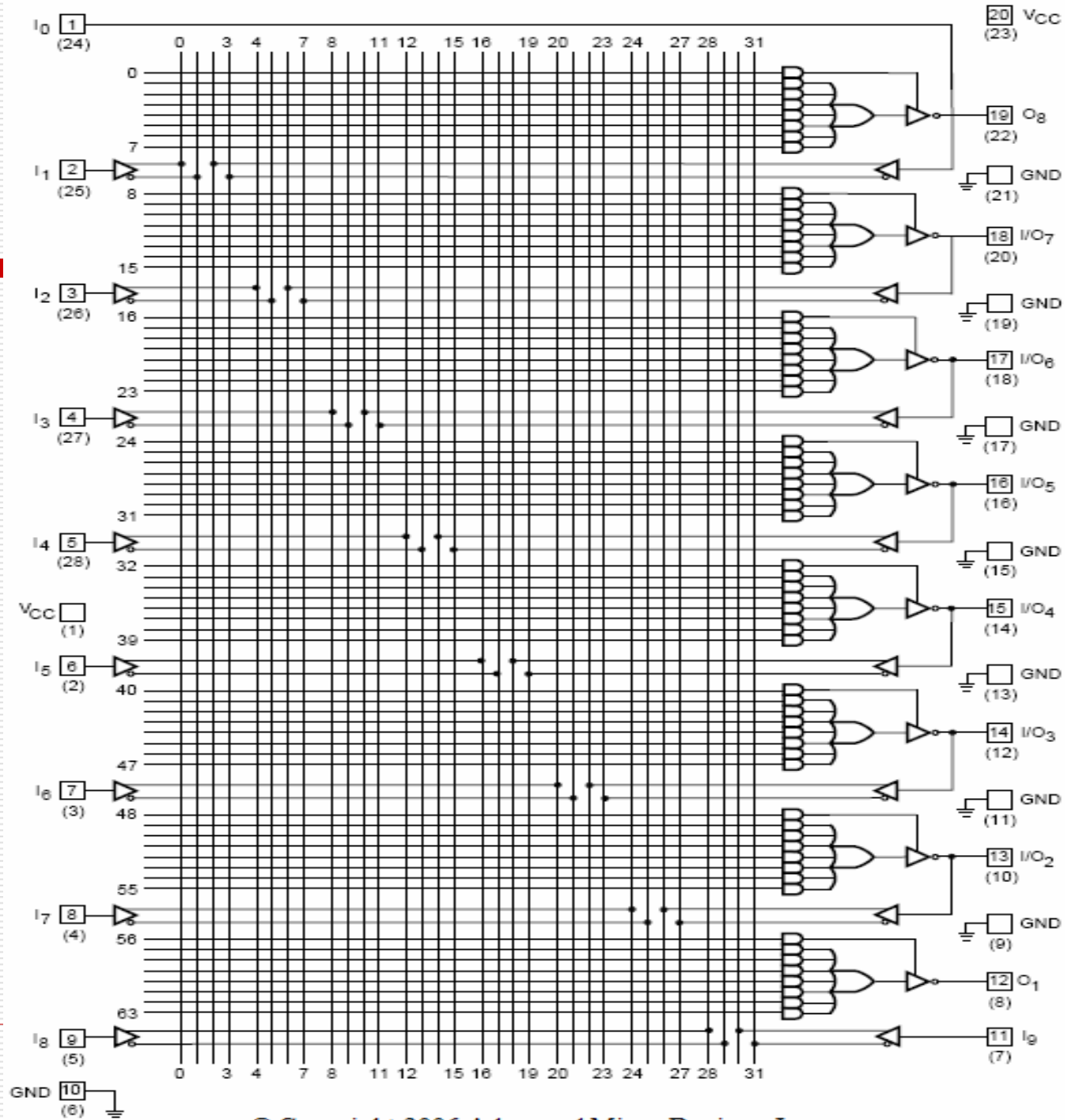


# PAL 4L4





# PAL 16L8



## PLA (*Programmable Logic Array*)

---

- Un PLA (*Programmable Logic Array*) est similaire à une ROM, mais il ne réalise pas tous les produits de termes comme une ROM. Un PLA à  $n$  entrées et  $m$  sorties peut réaliser  $m$  fonctions de  $n$  variables, en autant que chacune requiert un nombre limité de produits des variables en entrée. (En pratique, c'est presque toujours le cas).
- Un PLA est composé de deux réseaux programmables, ET et OU.

**PLA**  
**4 entrées,**  
**6 termes**  
**et 3 sorties**

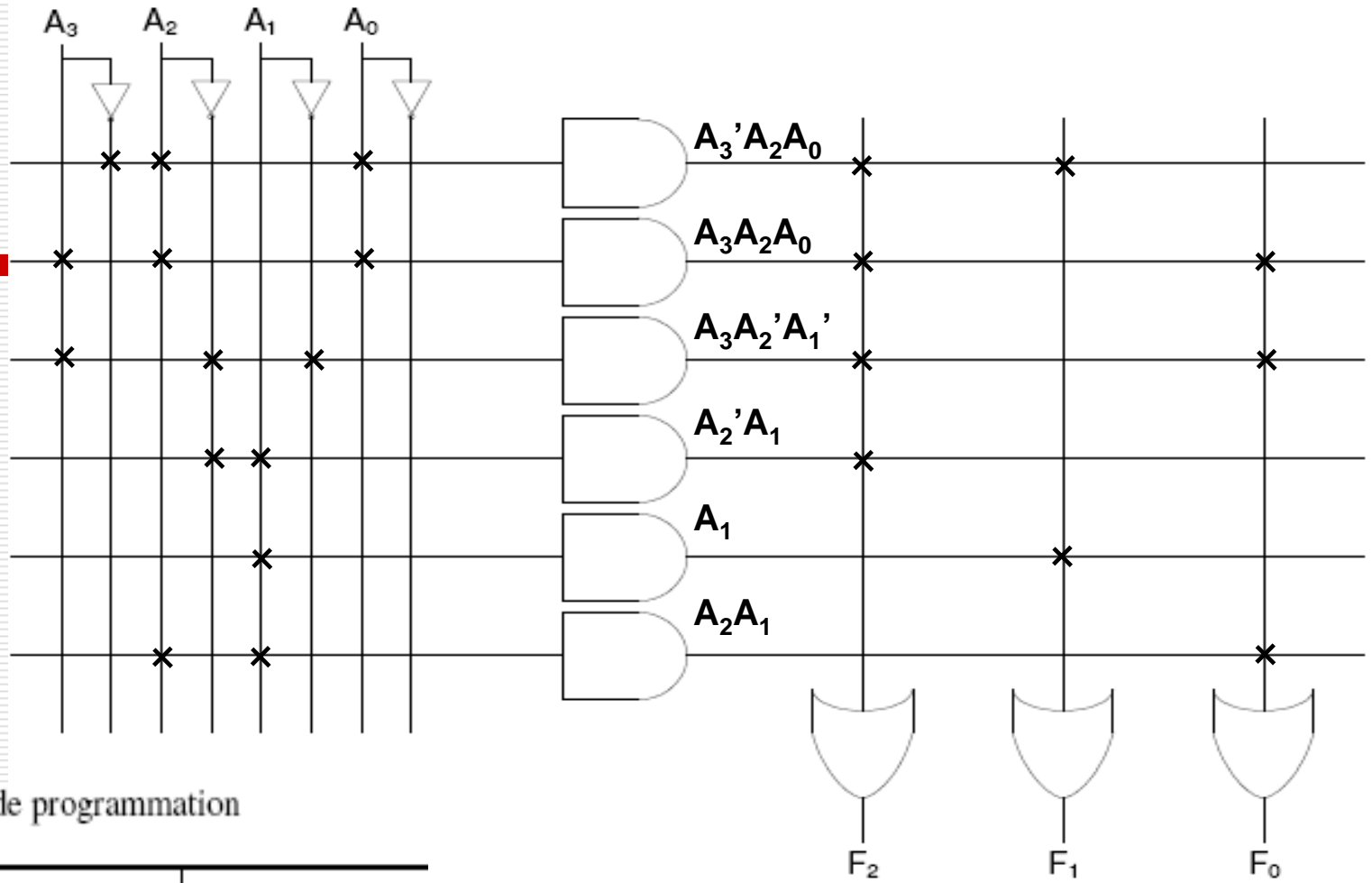


tableau de programmation

terme	$A_3$	$A_2$	$A_1$	$A_0$	$F_2$	$F_1$	$F_0$
0	0	1	-	1	1	1	0
1	1	1	-	1	1	0	1
2	1	0	0	-	1	0	1
3	-	0	1	-	1	0	0
4	-	-	1	-	0	1	0
5	-	1	1	-	0	0	1

## Résumé: ROM, PAL, PLA

---

Les ROM, PAL et PLA sont composés de deux réseaux : un réseau ET qui génère des mintermes, et un réseau OU qui permet de combiner plusieurs mintermes. La possibilité de programmer chacun de ces réseaux détermine si on a affaire à une ROM, un PAL ou un PLA. Le tableau suivant résume la situation.

type de dispositif	réseau ET	réseau OU
ROM	fixe (tous les mintermes sont générés par un décodeur)	programmable
PLA	programmable (un nombre limité de mintermes peuvent être générés)	programmable
PAL	programmable (un nombre limité de mintermes peuvent être générés)	fixe (un nombre limité de mintermes peuvent être combinés)

## **GAL (*Generic Array Logic*)**

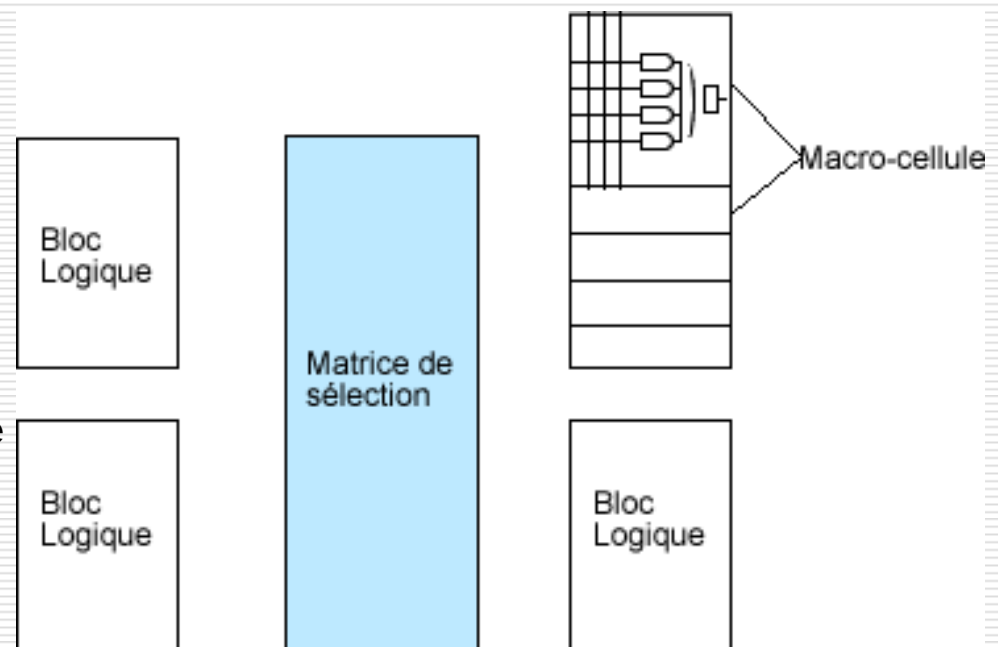
---

- C'est un PAL reprogrammable
- Il est constitué de plusieurs macro-cellules que l'utilisateur peut configurer en un bloc combinatoire simple, complexe ou bien en sortie avec registres.

# CPLD (*Complex Programmable Logic Devices*)

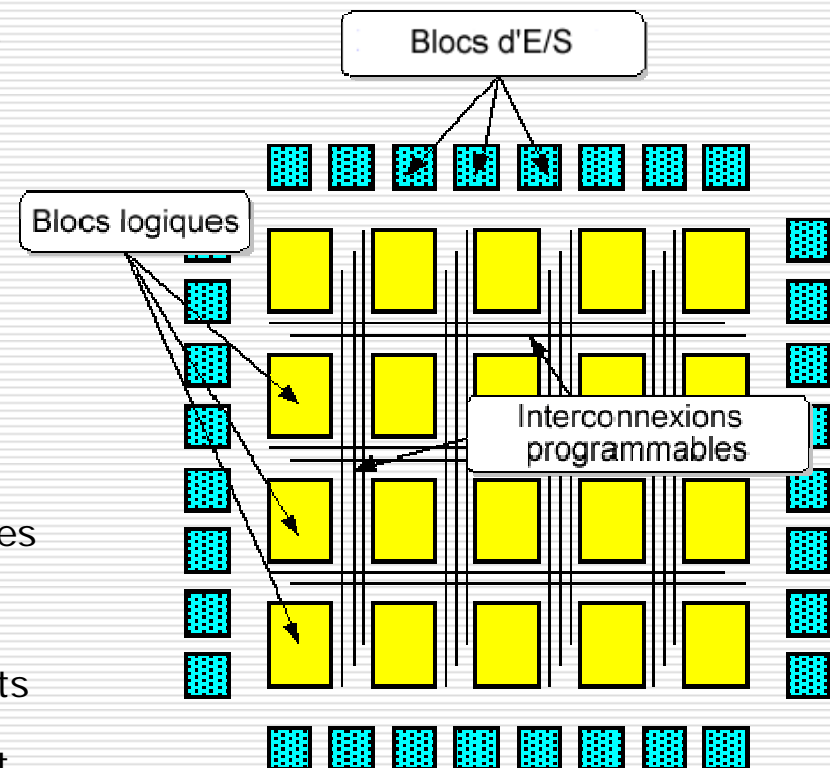
---

- D'un point de vue conceptuel, un CPLD consiste en plusieurs macro-cellules de PAL interconnectées par l'entremise d'une matrice de sélection.
- Contient de quelques dizaines à quelques centaines de macro-cellules.
- En général, les groupes de 8 à 16 macro-cellules se regroupés dans un bloc de fonction.
- Selon le distributeur et la famille du CPLD, les blocs de fonction sont également interconnectés.



# FPGA (*Field Programmable Gate Array*)

- Offre la plus grande capacité logique.
- Un FPGA est une matrice de blocs logiques, entourés de blocs d'E/S, reliés par des connexions programmables.
- Un FPGA peut contenir entre 64 et plusieurs dizaines de milliers de blocs logiques et un nombre de bascules encore plus grand.
- Il existe deux types de FPGA:
  - À gros grains (*'coarse-grained'*): contient des blocs logiques relativement gros, contenant deux tables de conversion ou plus ainsi que deux bascules ou plus.
  - À grains fins (*'fine-grained'*): contient d'avantage de blocs logiques simples. Ces blocs contiennent soit une fonction d'entrée double ou un multiplexeur 4-1 et des bascules.
- Les FPGAs les plus denses sont construits à l'aide de la technologie des SRAMs
- Le procédé d'anti-fusibles est également utilisé parfois.

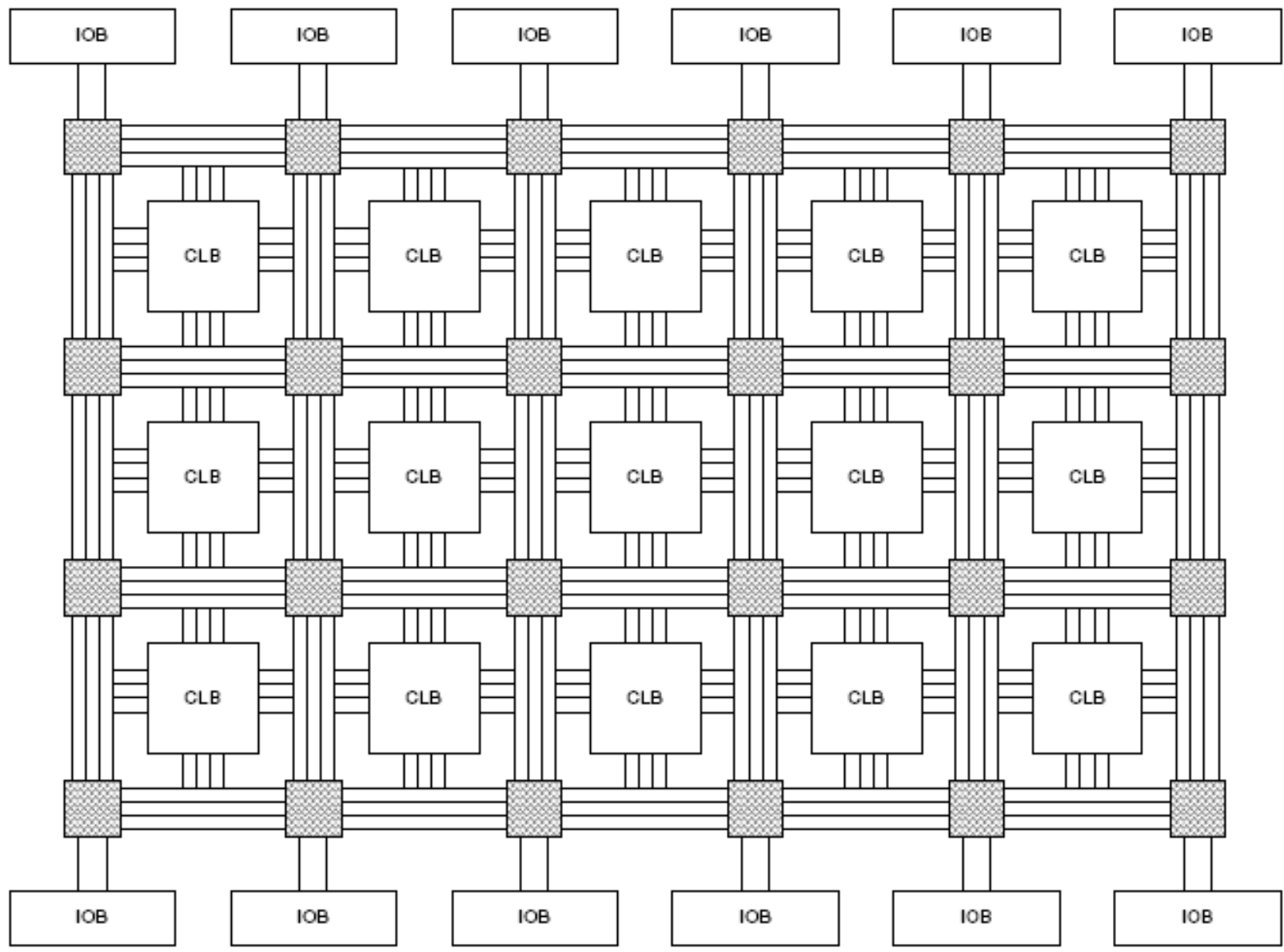


## FPGA (*Field Programmable Gate Array*)

---

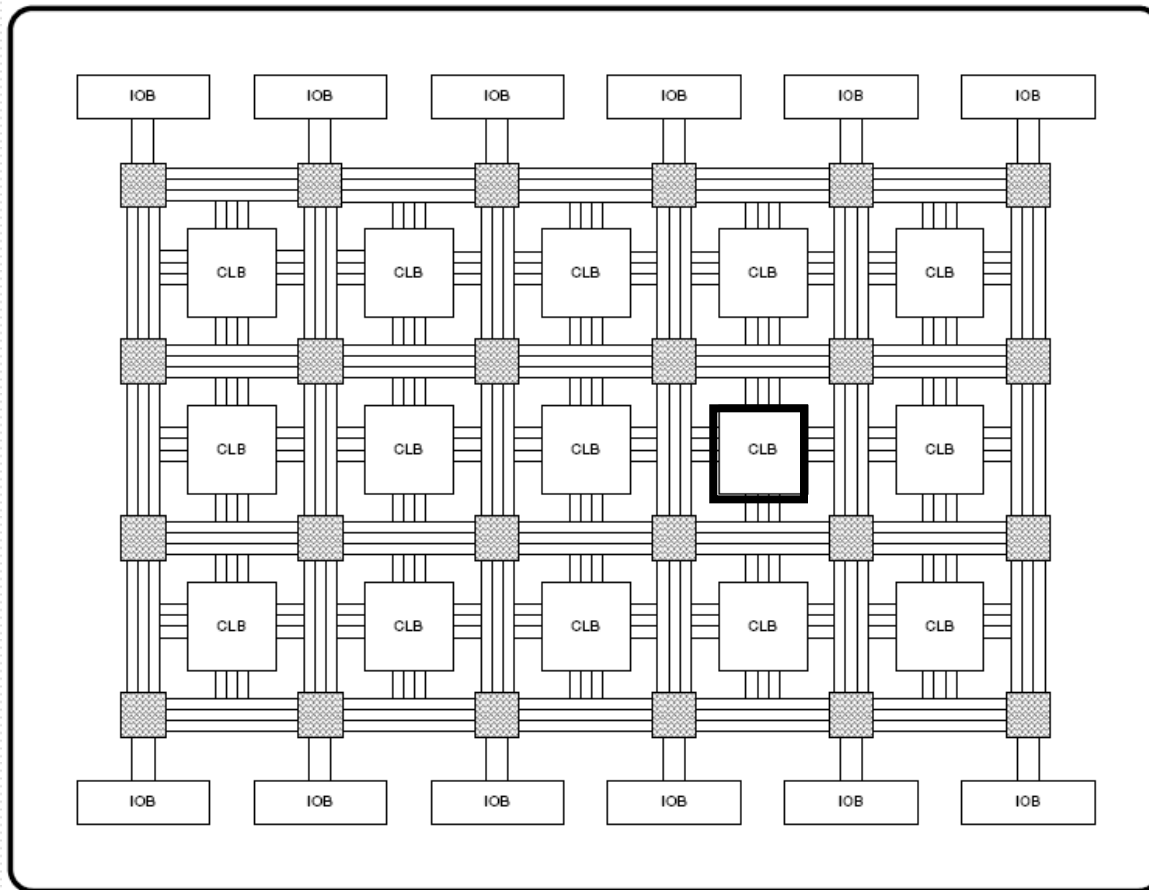
- Un FPGA est composé à la base de :
  - un réseau de blocs de logique programmable (*Configurable Logic Block - CLB*), chaque bloc pouvant réaliser des fonctions complexes de plusieurs variables, et comportant des éléments à mémoire;
  - un réseau d'interconnexions programmables entre les blocs; et,
  - des blocs spéciaux d'entrée et de sortie avec le monde extérieur (*Input/Output Block – IOB*).



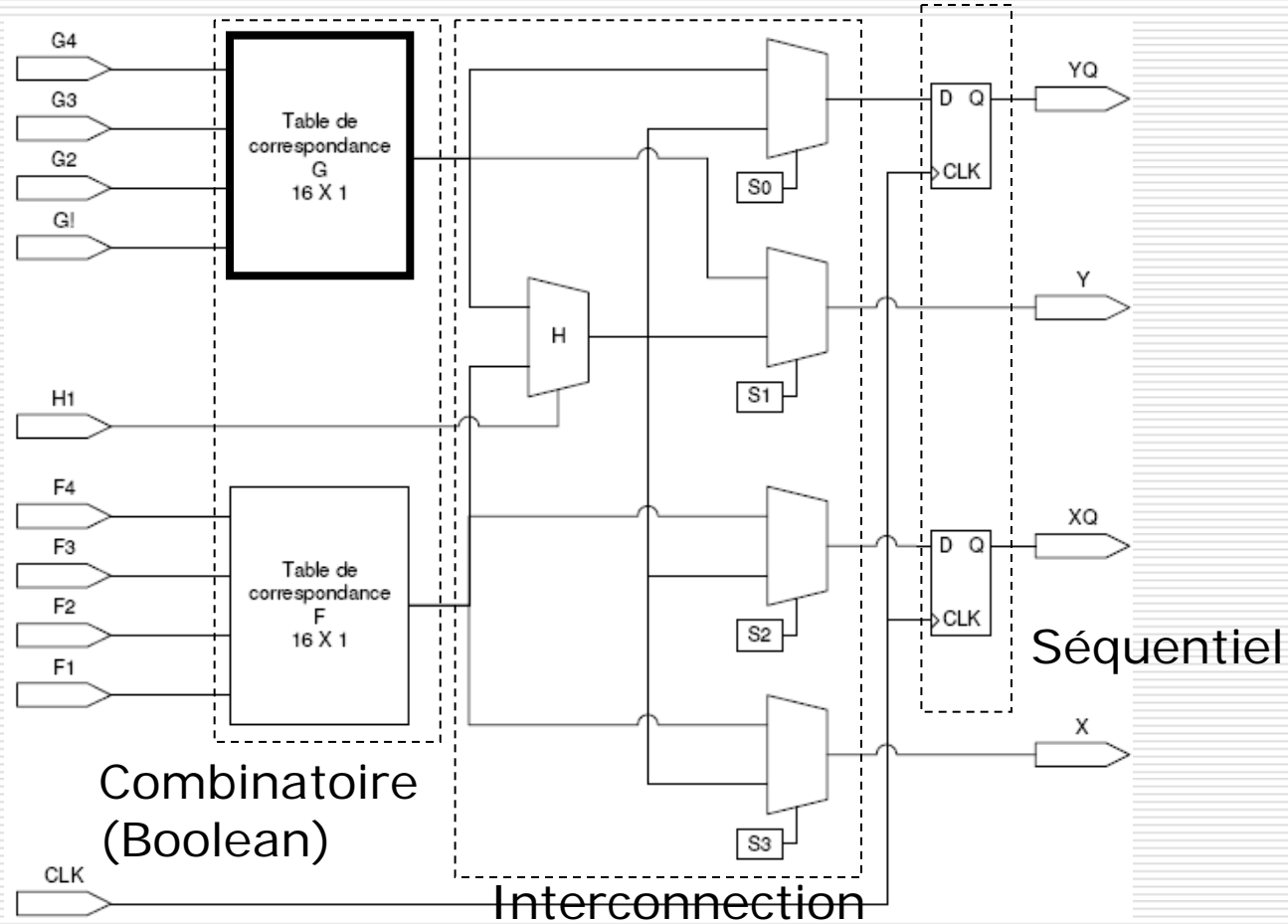


# CLB

---

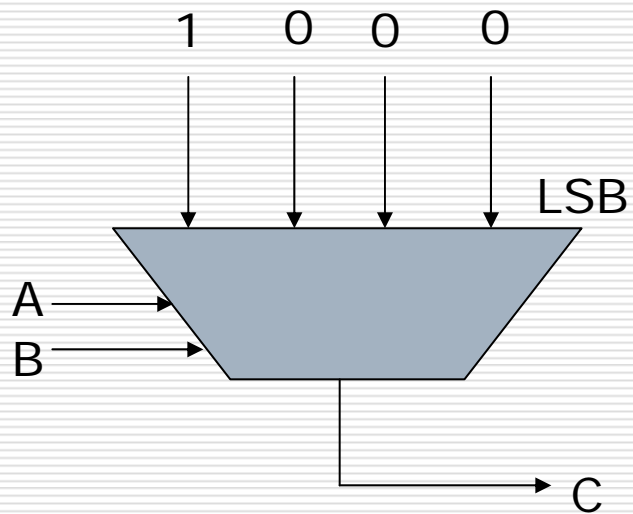


# CLB

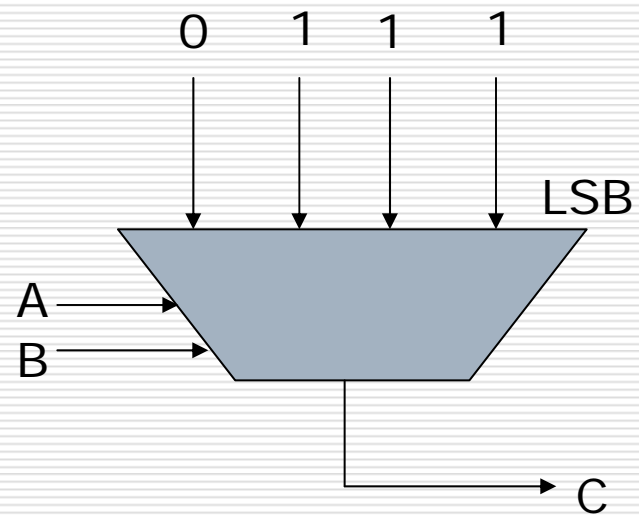


## Table de correspondance - MUX

---



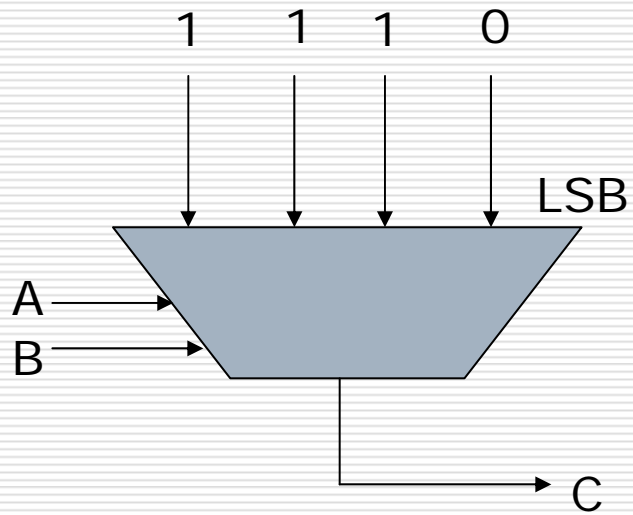
AND



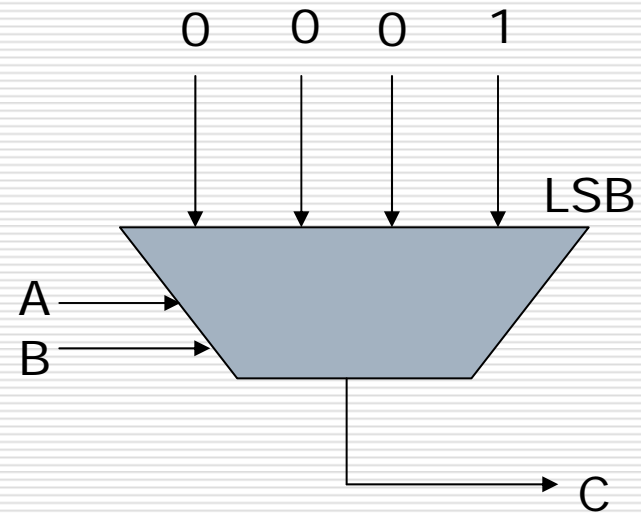
NAND

## Table de correspondance - MUX

---

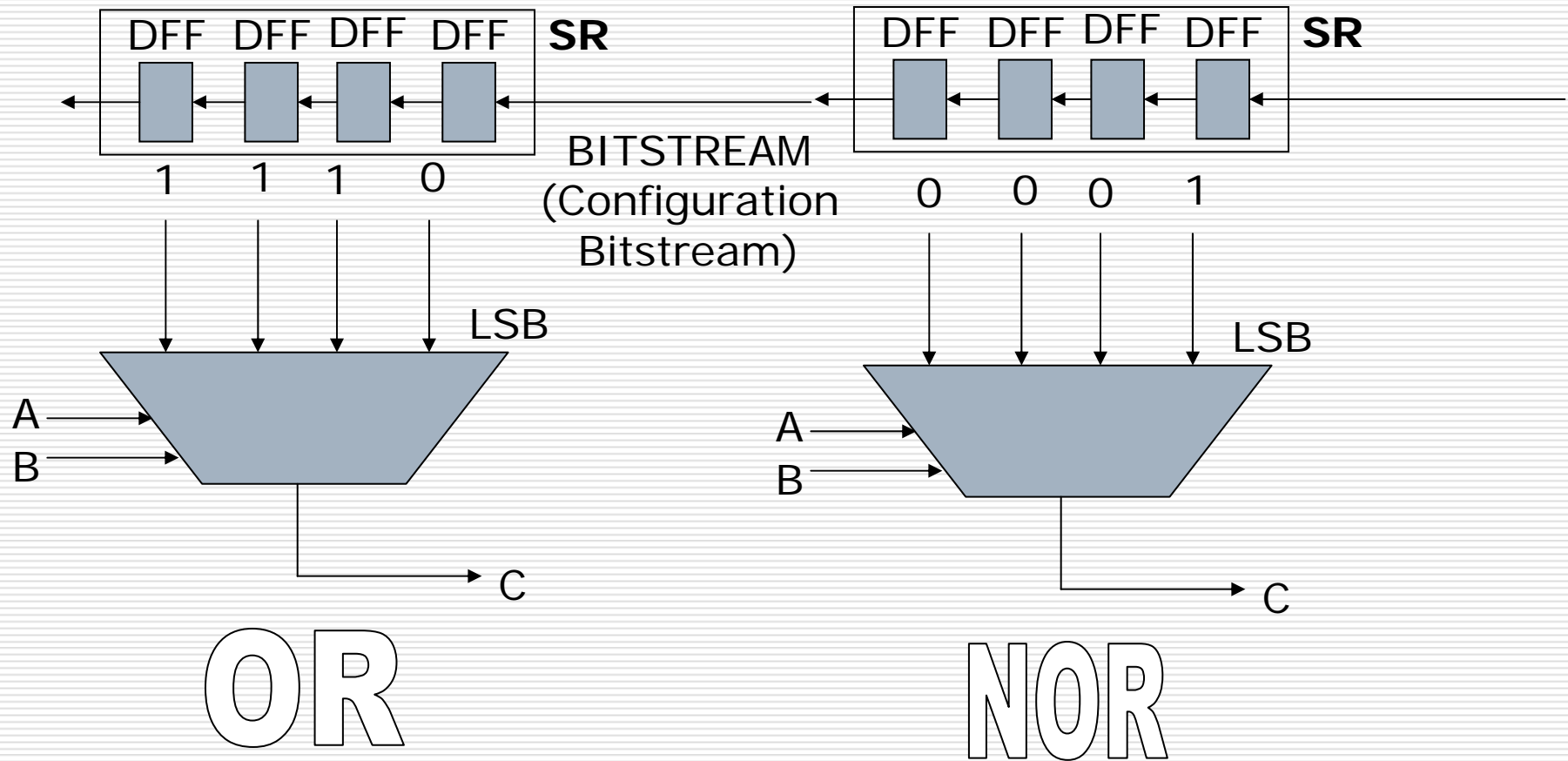


OR

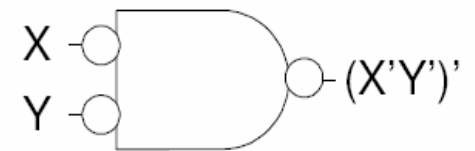
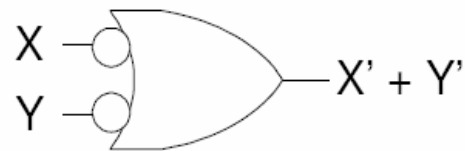
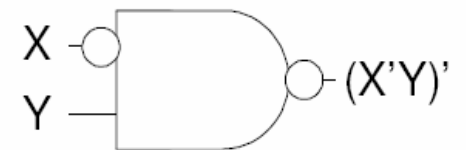
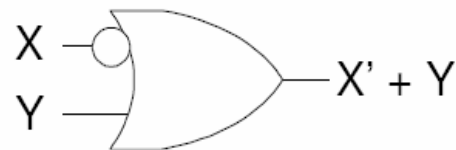
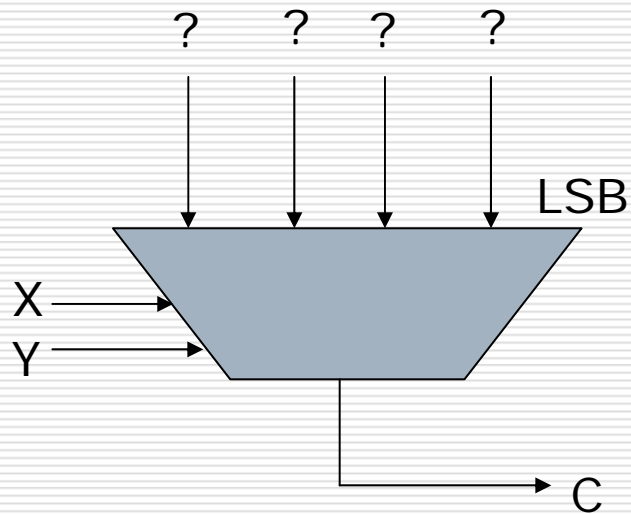


NOR

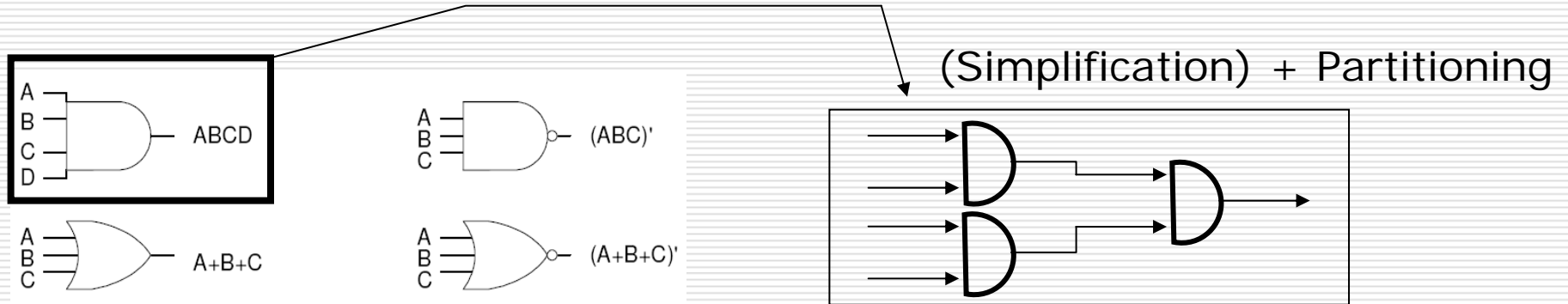
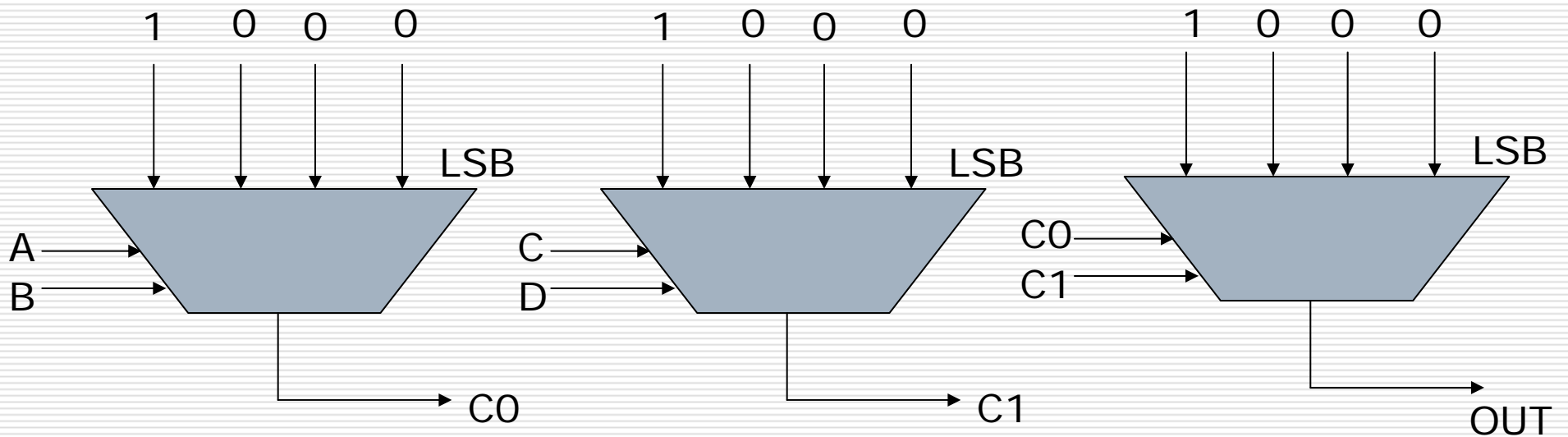
# Table de correspondance – LUT + MUX



# Table de correspondance – MUX - Exercices



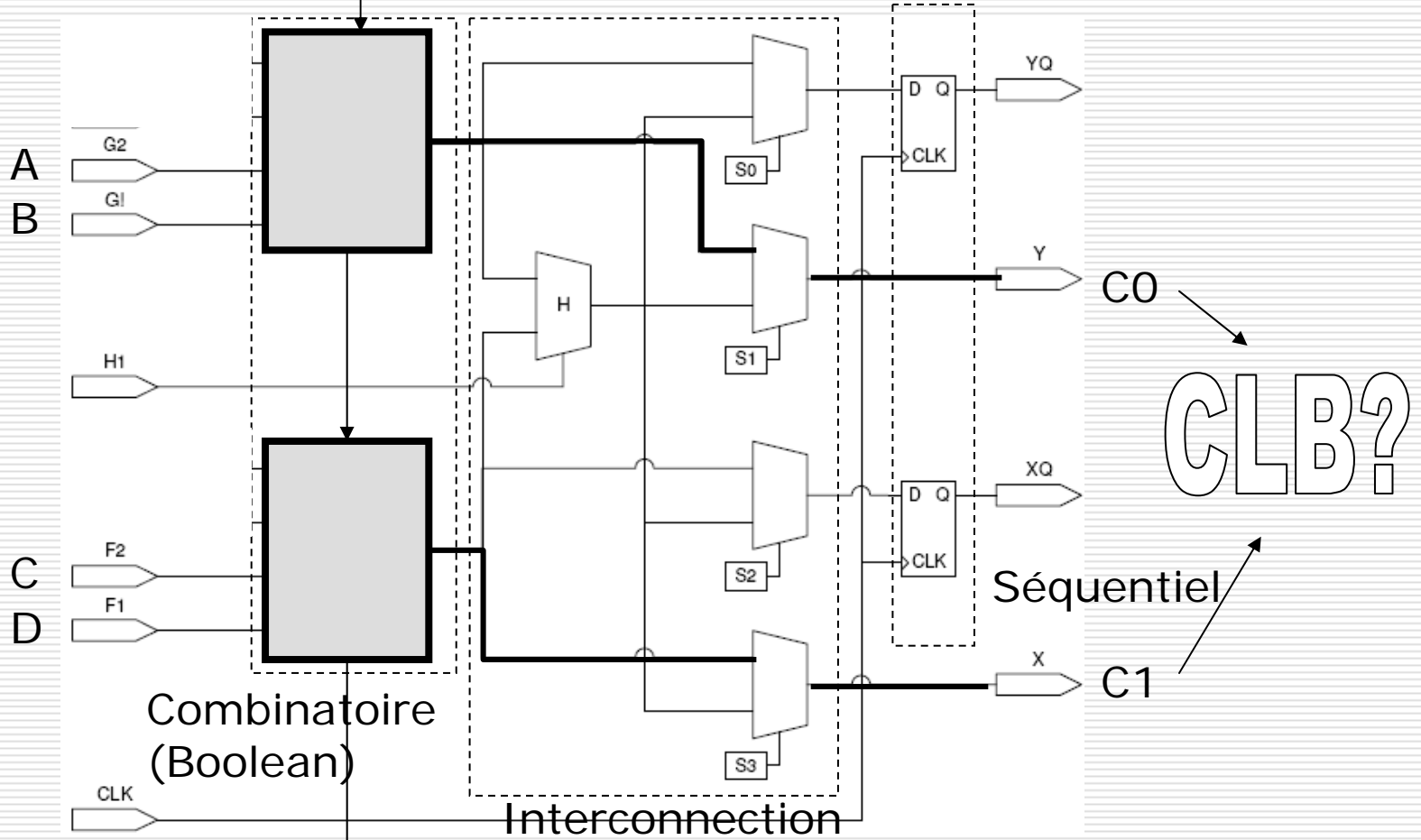
# Table de correspondance – « Partition » - Exercices



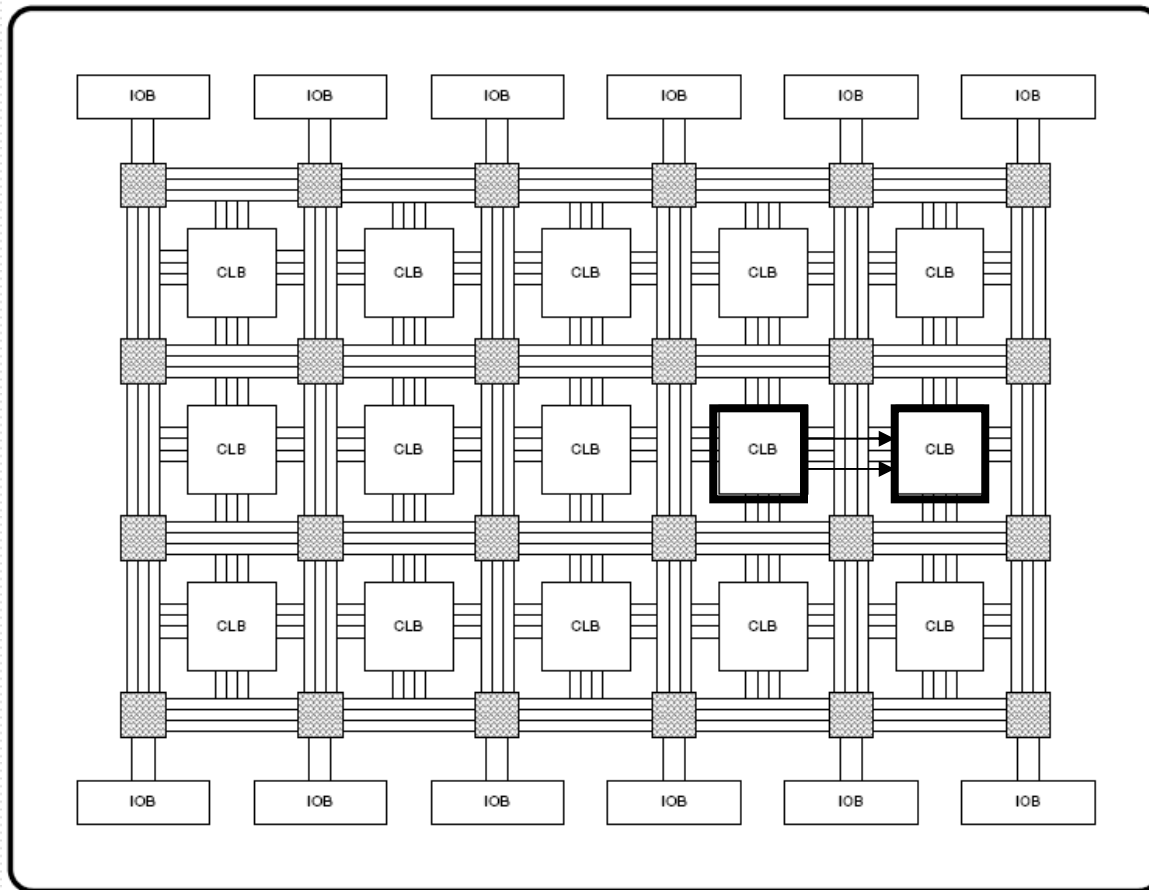


Bitstream → Ex: S0 = X; S1 = 0; S2 = X; S3 = 0  
H1 = X

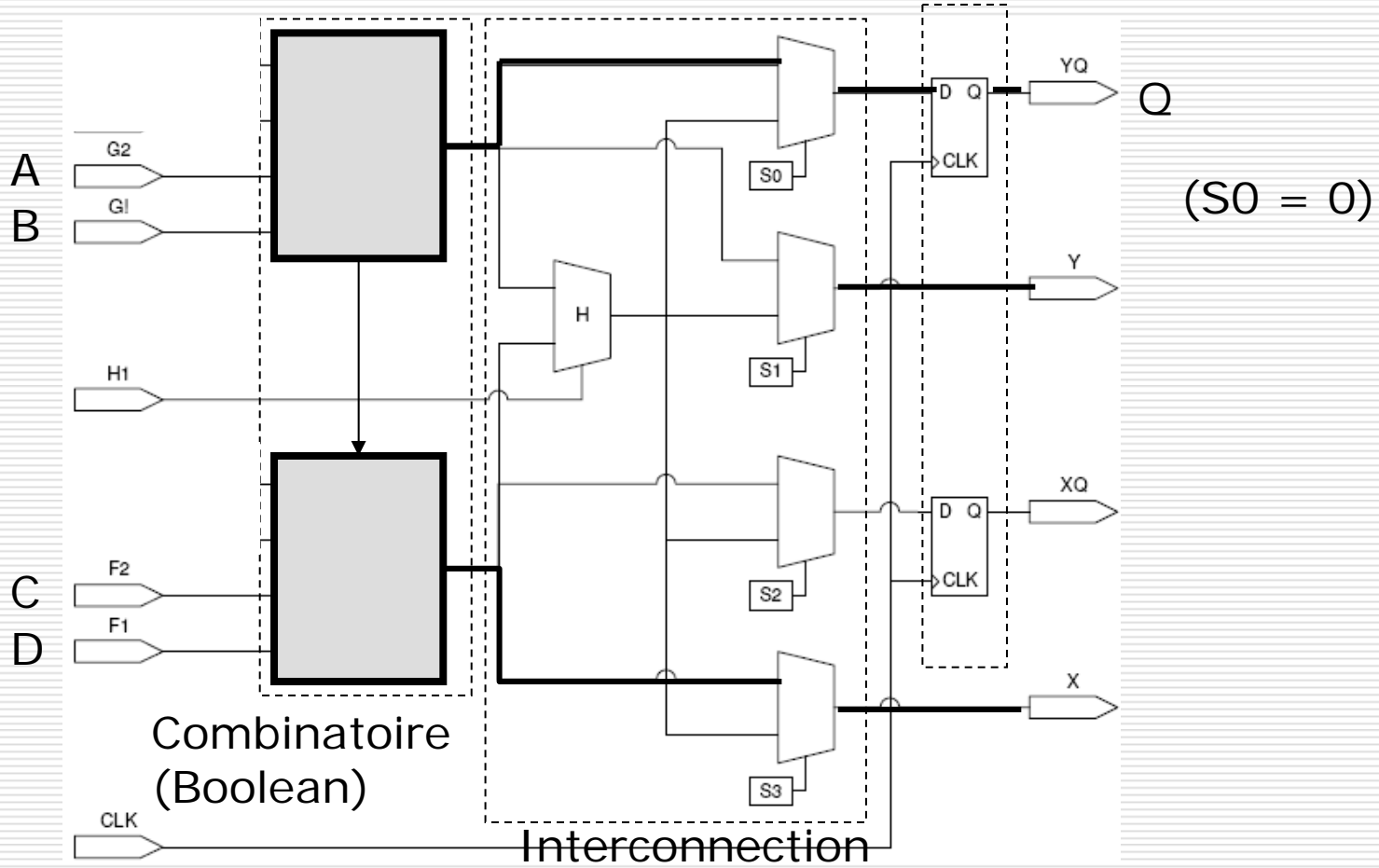
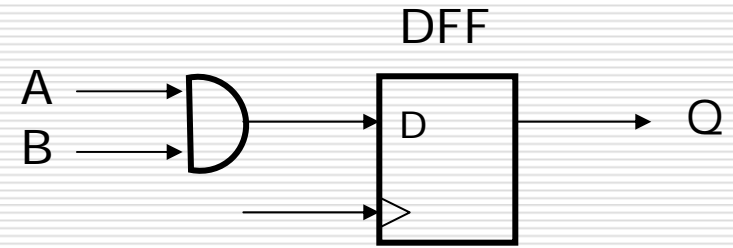
# Placement + Routage



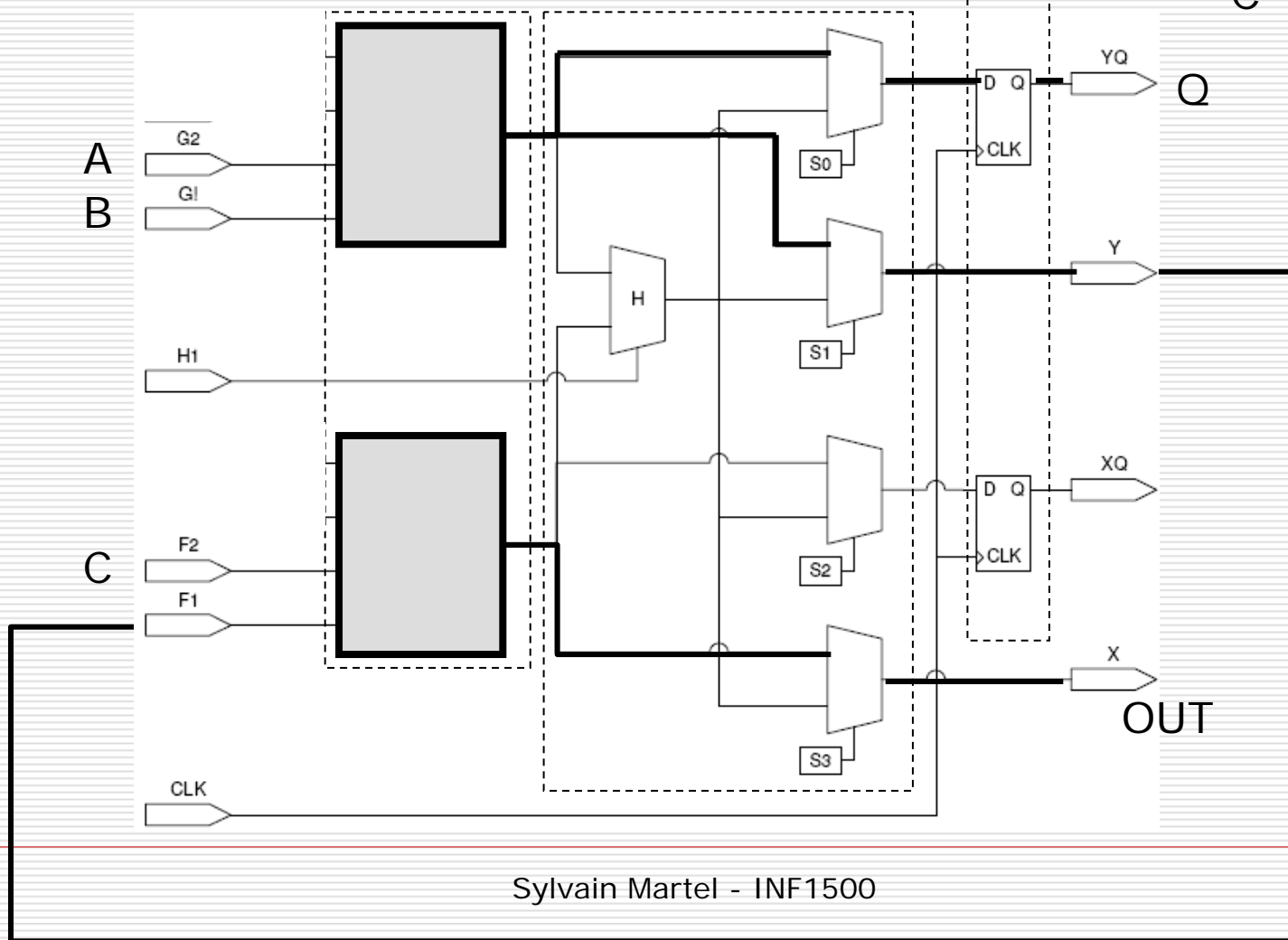
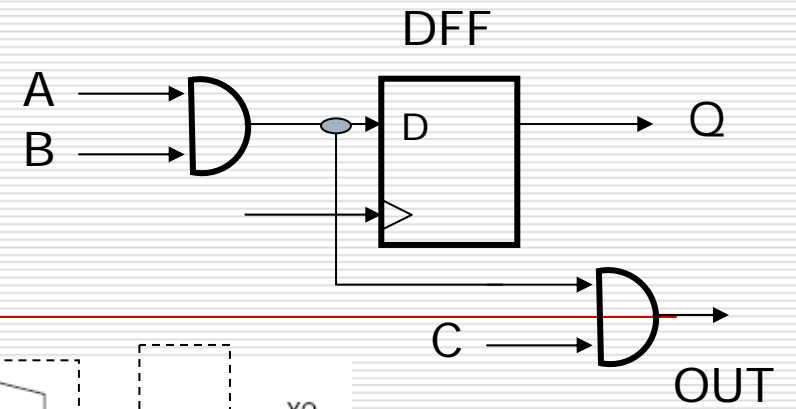
# Placement: CLB? + Routage: bits de configuration dans le bitstream pour interconnexions entre CLBs



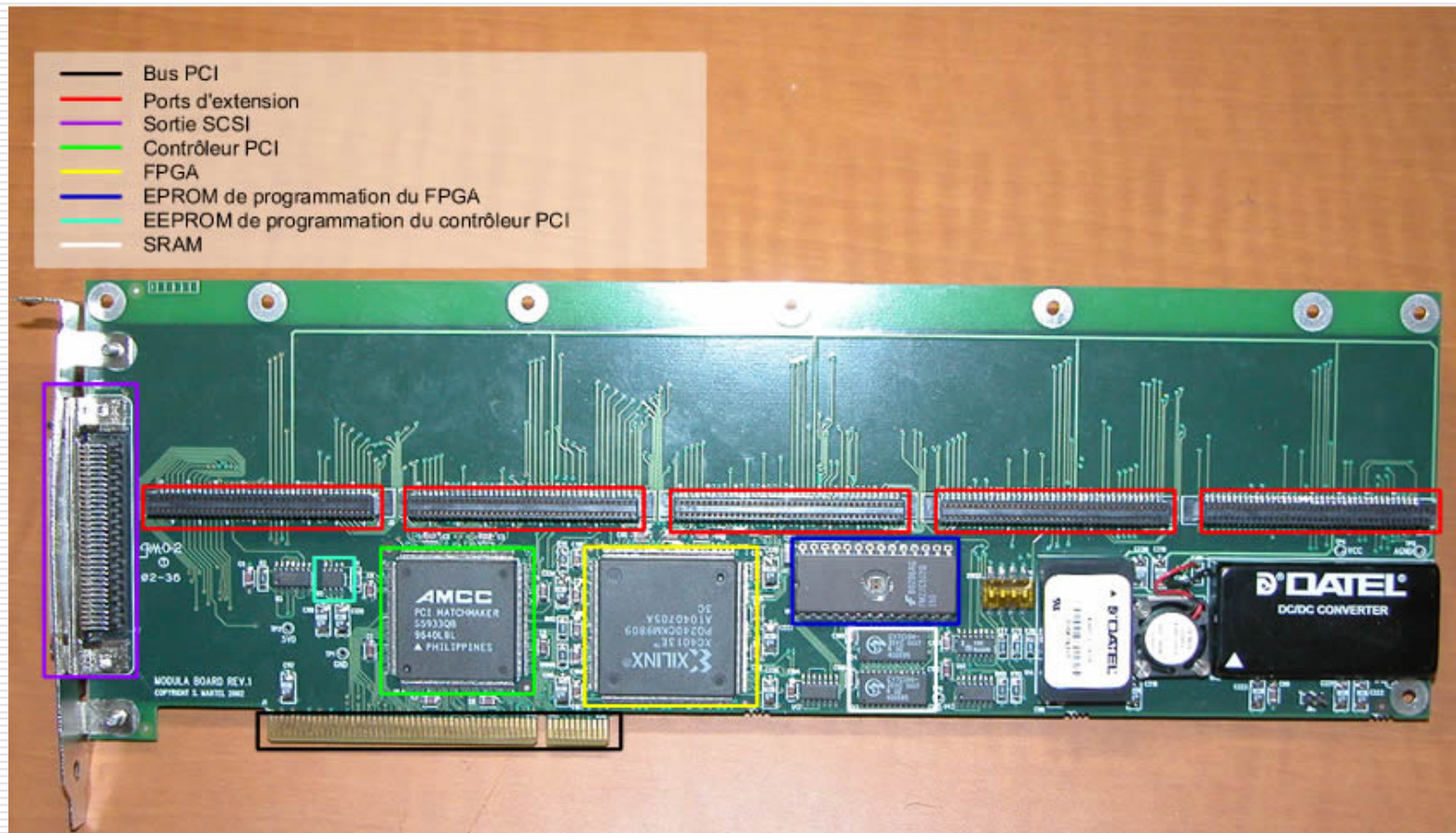
# Avec FF

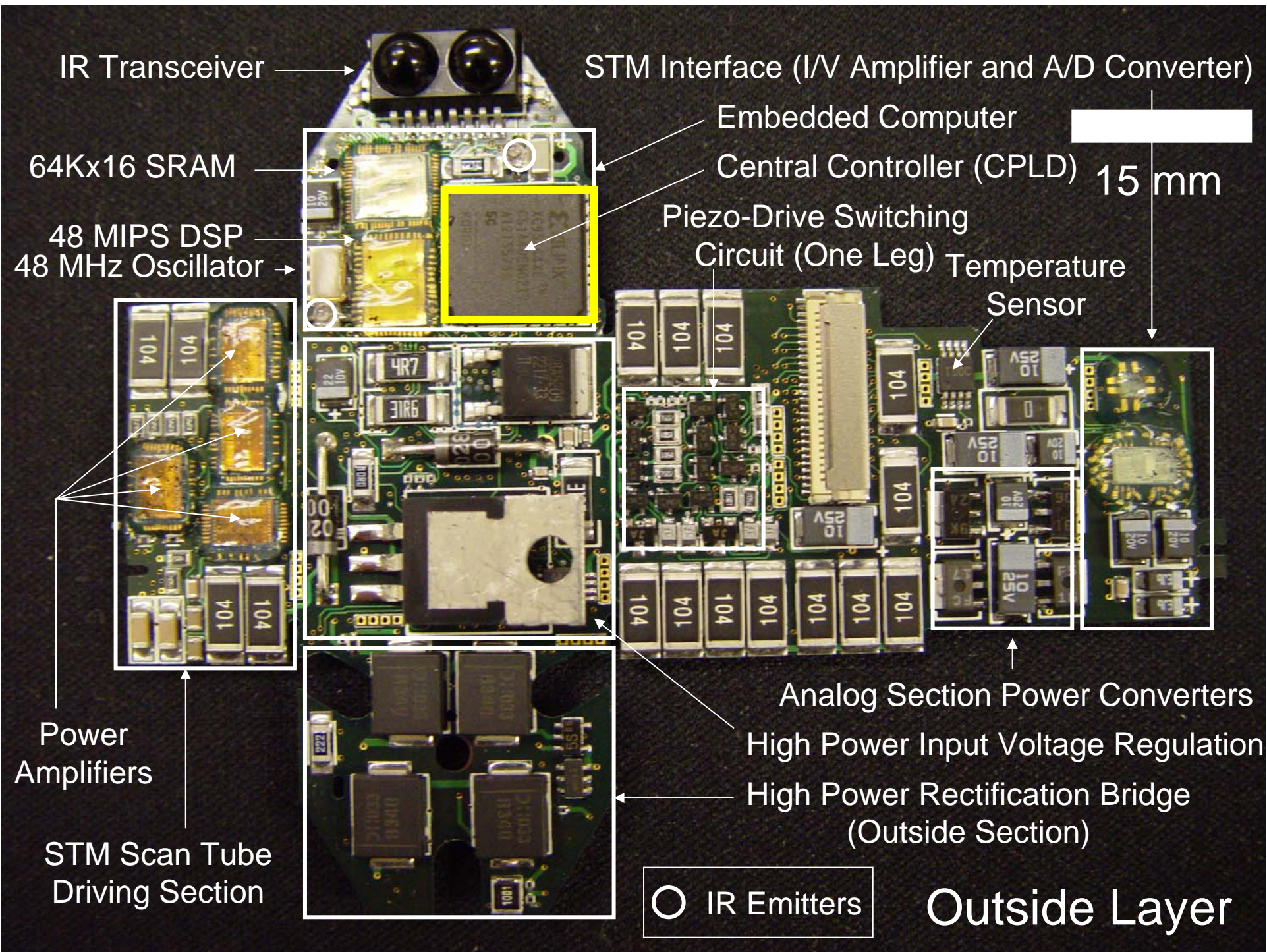


# Un autre exemple



# Carte MODULA





IR Transceiver

STM Interface (I/V Amplifier and A/D Converter)

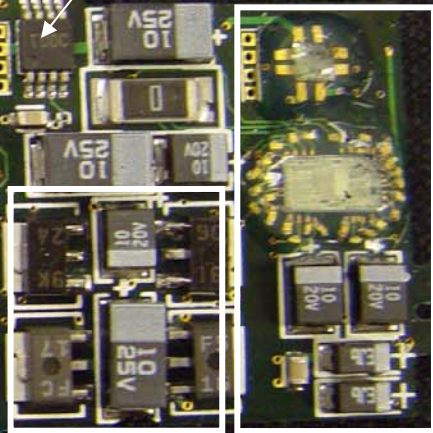
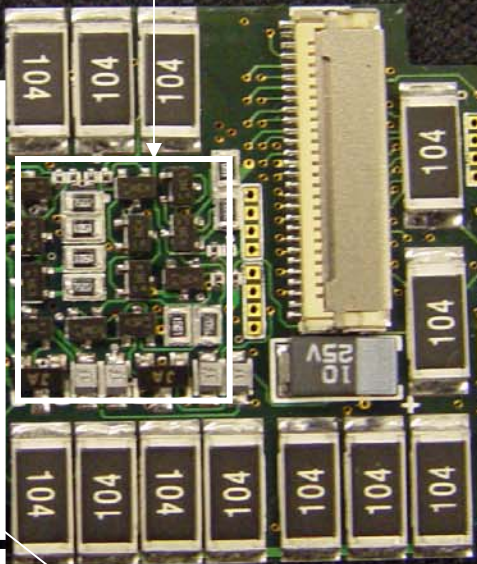
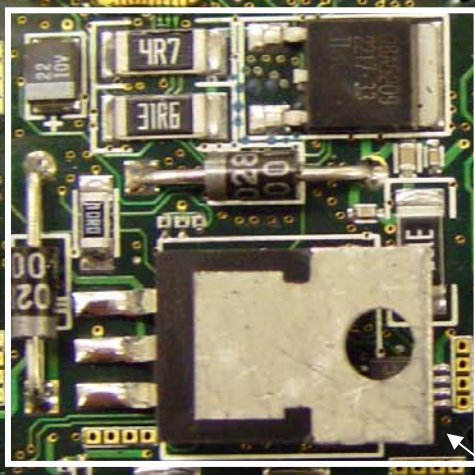
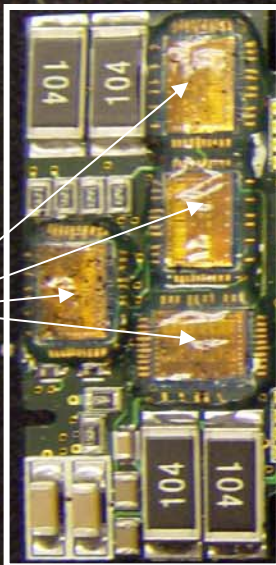
64Kx16 SRAM

Embedded Computer

Central Controller (CPLD) 15 mm

48 MIPS DSP  
48 MHz Oscillator

Piezo-Drive Switching  
Circuit (One Leg) Temperature  
Sensor



Power Amplifiers

Analog Section Power Converters

High Power Input Voltage Regulation

STM Scan Tube Driving Section

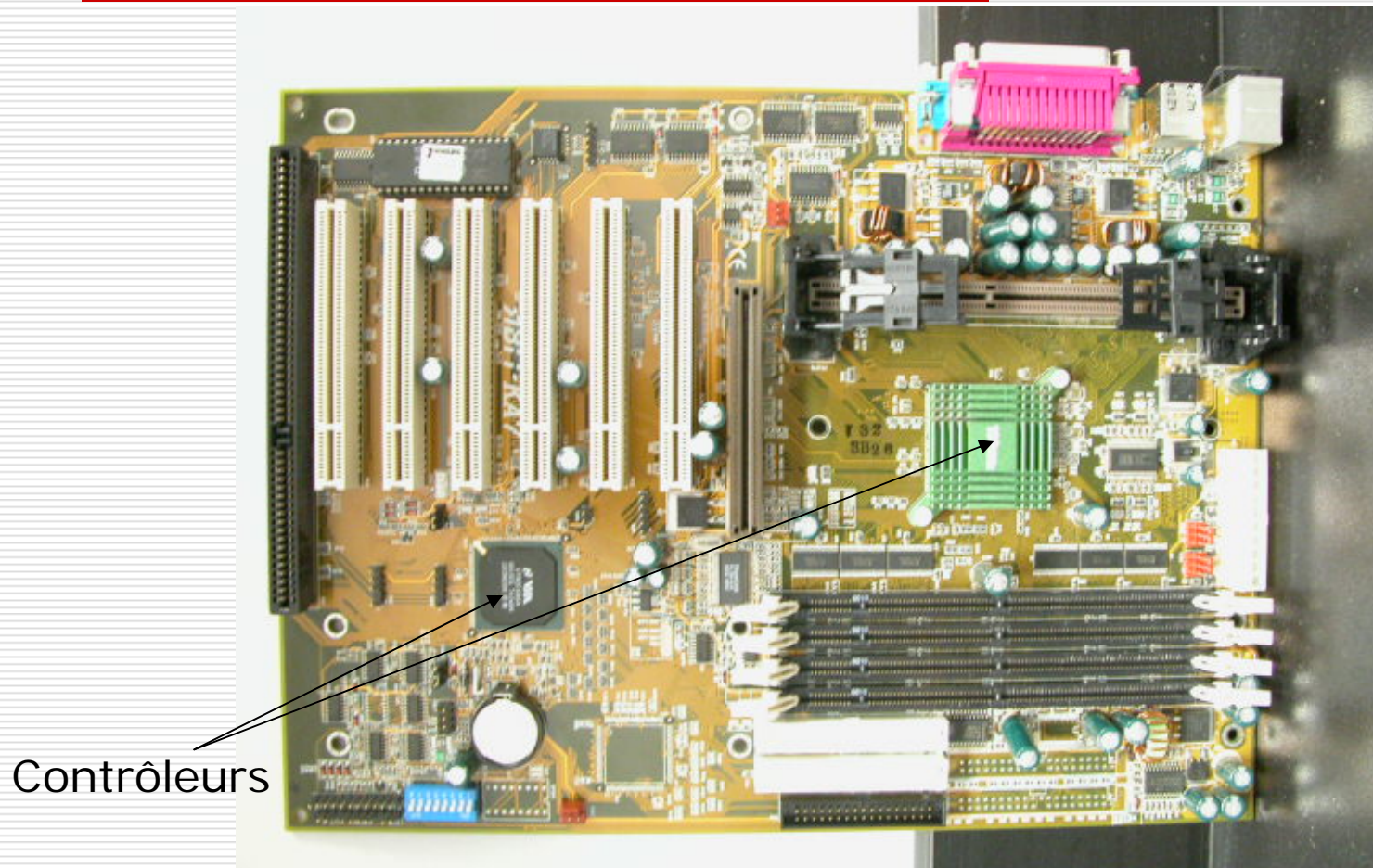
High Power Rectification Bridge (Outside Section)

○ IR Emitters

Outside Layer

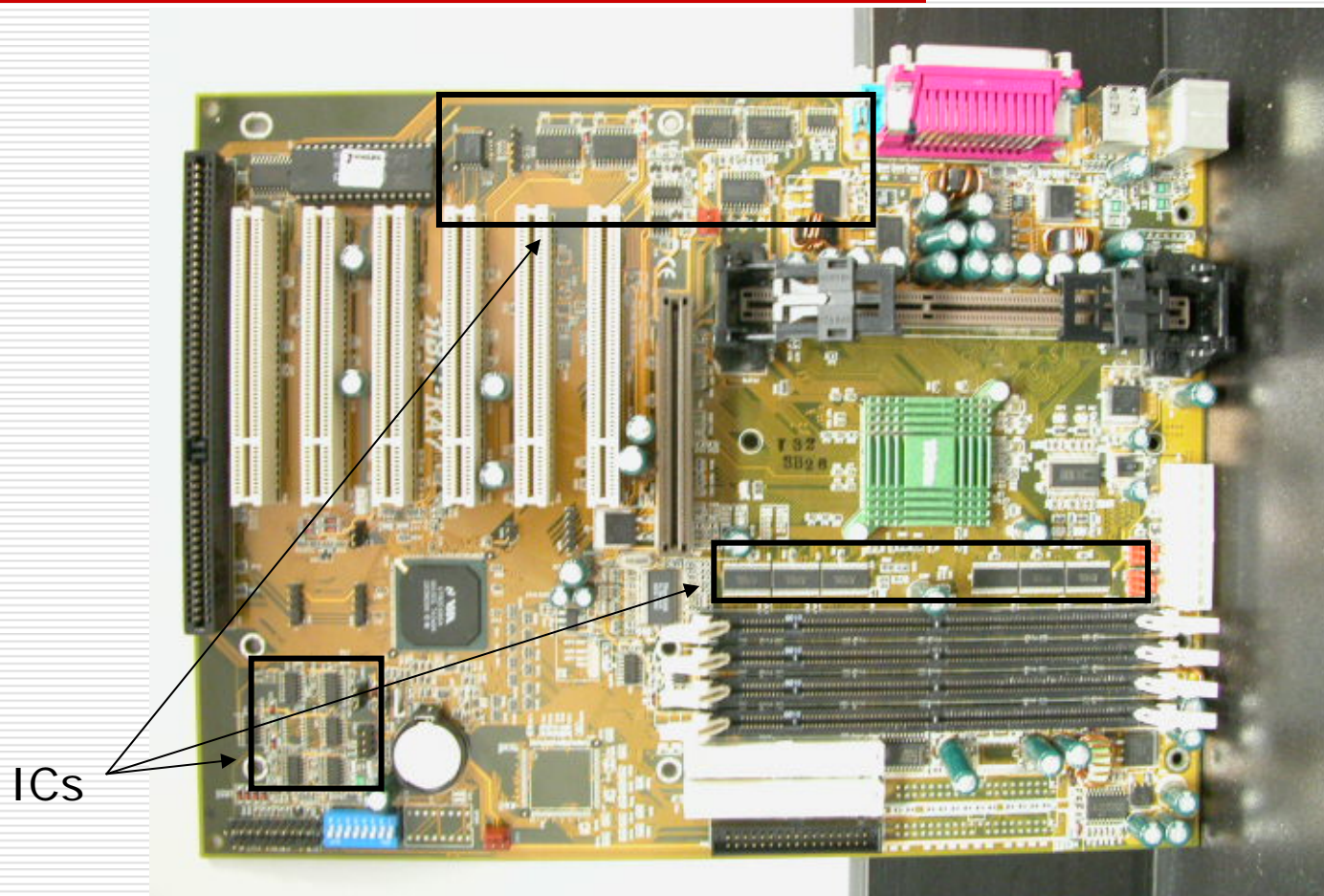
# ASIC (Application Specific Integrated Circuit) and VLSI (Very Large Scale Integration)

---



# TTL, CMOS, etc...

---





## Exercices – Bitstreams

---

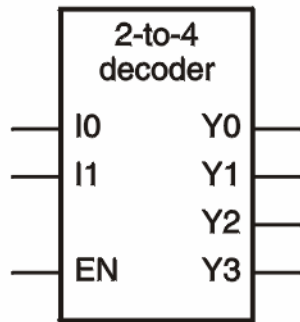
#	<i>A</i>	<i>B</i>	<i>C</i>	<i>F3</i>
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

#	<i>A</i>	<i>B</i>	<i>F4</i>
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	1

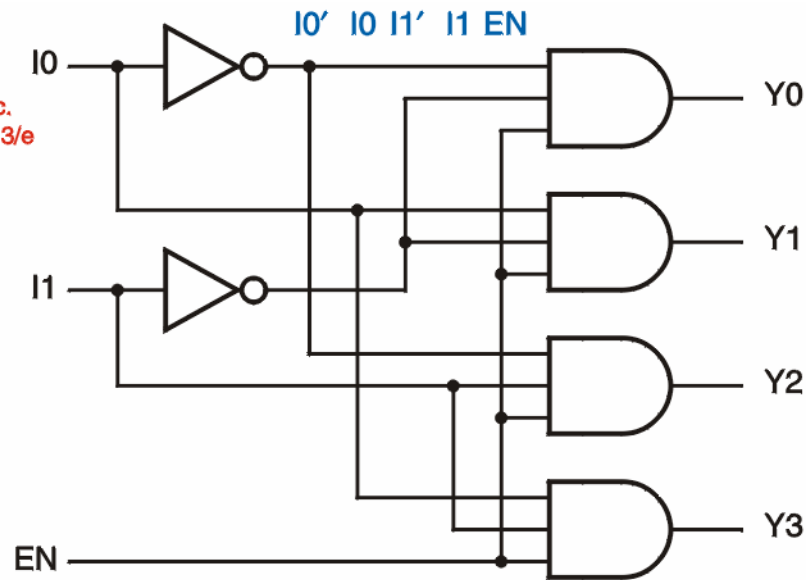
#	<i>A</i>	<i>B</i>	<i>F5</i>
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	1

# Exercices - Décodeur 2-à-4 - Bitstream

Copyright © 2000 by Prentice Hall, Inc.  
Digital Design Principles and Practices, 3/e



(a)



(b)

**Table 5-4**  
Truth table for a 2-to-4  
binary decoder.

Inputs			Outputs			
EN	I1	I0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

