

# Fonctions logiques et technologies associées

## M1103 - Architecture des équipements informatiques

`Sebastien.Kramm@univ-rouen.fr`

IUT R&T Rouen, site d'Elbeuf

2018-2019

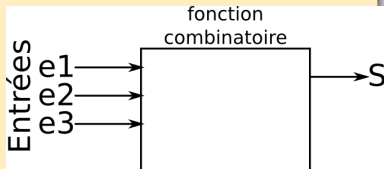
# Sommaire

- 1 Classification
- 2 Introduction à la logique séquentielle
- 3 Aspects technologiques

# Classification des fonctions logiques

## Logique combinatoire

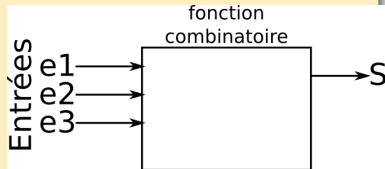
- Les sorties sont une fonction combinatoire des entrées :  $S = f(E)$
- A une configuration des entrées correspond une configuration unique des sorties.



# Classification des fonctions logiques

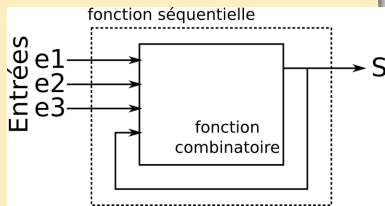
## Logique combinatoire

- Les sorties sont une fonction combinatoire des entrées :  $S = f(E)$
- A une configuration des entrées correspond une configuration unique des sorties.



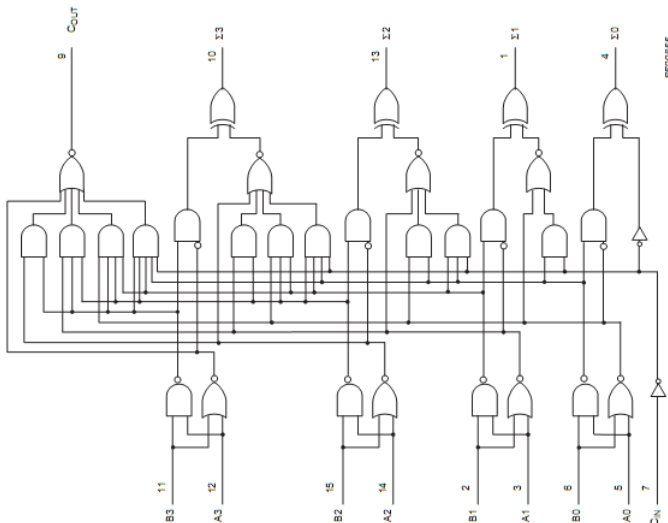
## Logique séquentielle

- Les sorties sont fonctions des entrées ET de l'état interne du système.
- A une configuration des entrées peut correspondre plusieurs configurations des sorties.

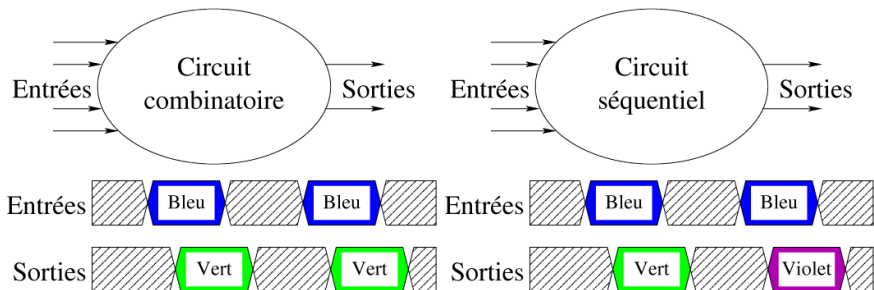


# Exemple de fonction combinatoire

CI 74F283 : additionneur 4 bits (obsolète depuis longtemps...)



# Conséquences

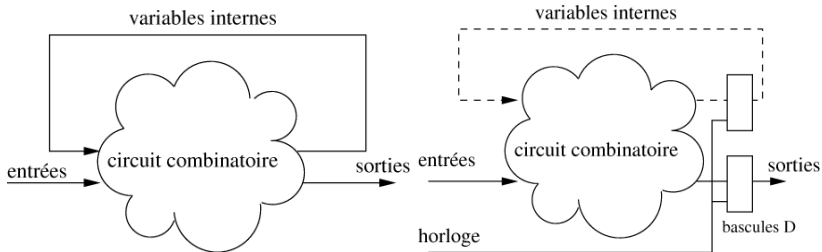


- La logique séquentielle amène la notion d'**état interne**
- La valeur de la sortie dépend des entrées **et** de celui-ci.

# Logique séquentielle : 2 types

On distingue :

- Logique séquentielle **asynchrone** :  
Les sorties changent d'état de façon non synchronisée, selon l'équation propre de chaque sortie.
- Logique séquentielle **synchrone** :  
Les changements d'état des sorties sont synchronisés sur une horloge commune.  
⇒ Plus performant, mais plus complexe.



# Sommaire

- 1 Classification
- 2 Introduction à la logique séquentielle
- 3 Aspects technologiques

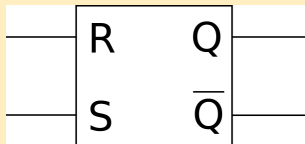


# Fondamentaux de logique séquentielle

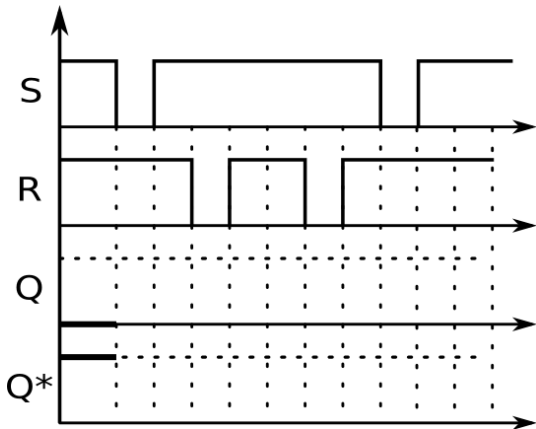
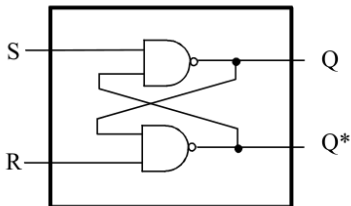
- Toutes les fonctions séquentielles évoluées (registres, compteurs, etc.) utilisent comme brique de base la **bascule**.
- Définition : une bascule est un circuit qui peut maintenir les valeurs de ses sorties **malgré** les changements de valeurs d'entrées :  
⇒ composant comportant un **état mémoire**.
- Différents types
  - Bascule RS (S : *Set*, R : *Reset*) : la plus simple
  - Bascule D : mémoire 1 bit
  - Bascule JK : la plus élaborée

## Fonctionnement de la bascule RS

- Activation de R (*Reset*) : Q passe à
- Activation de S (*Set*) : Q passe à
- au repos :



# Bascule RS élémentaire



- Equations :

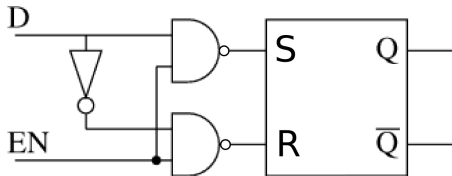
$$Q = \overline{R \cdot Q^*}$$

$$Q^* = \overline{S \cdot Q}$$

- Fonctionnement :

# Bascule D

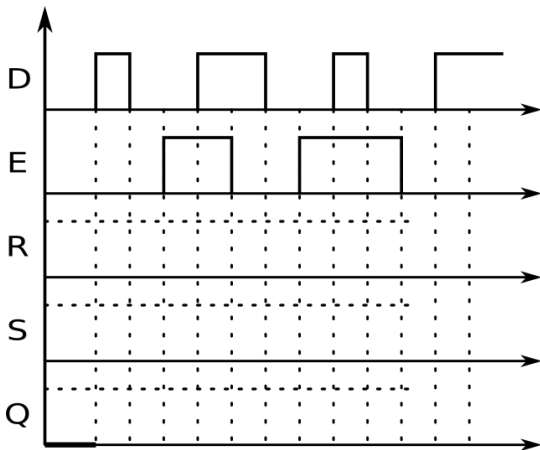
- La bascule D est utilisée pour mémoriser une valeur binaire.
- Elle est construite à partir d'une bascule RS.



- Equations :  
S =  
R =

D	E	R	S	$Q_{n+1}$
0	0			
1	0			
0	1			
1	1			

# Bascule D : chronogramme



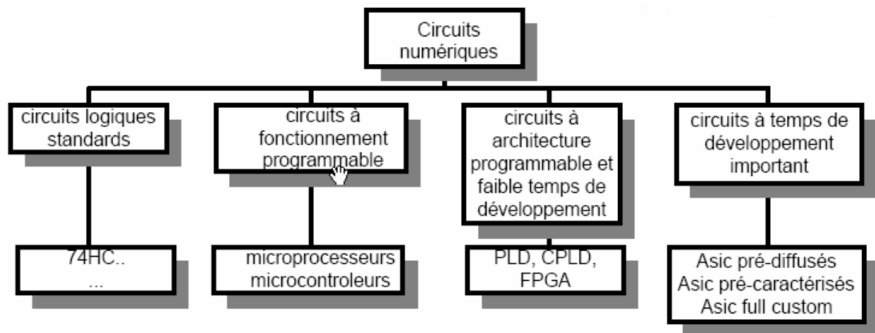
- Inconvénient : transitions sur **niveaux**
- bascules réelles  $\Rightarrow$  transitions sur fronts (mais plus complexes)

# Sommaire

- 1 Classification
- 2 Introduction à la logique séquentielle
- 3 Aspects technologiques**

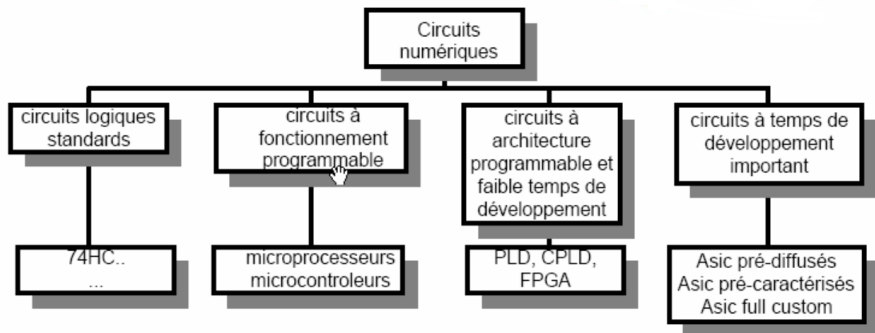
# Implémentation des fonctions complexes

- Lors de la conception d'une fonction numérique, plusieurs choix d'implémentation sont possible.



# Implémentation des fonctions complexes

- Lors de la conception d'une fonction numérique, plusieurs choix d'implémentation sont possible.



- ASIC (*Application Specific Integrated Circuit*) : Circuit intégré spécialisé, construit sur mesure pour une application donnée
  - pertinent que pour des très grandes séries (+ de 100k/an)
  - En perte de vitesse devant les FPGA, qui offrent plus de souplesse et **Time to Market** réduit.

# Implémentation

- Historiquement, ces fonctions étaient implémentées dans des circuits intégrés dédiés



TTL TECHNOLOGY FAMILY : PRODUCT TREE

Device Name	Resources	Device Description	Status*	Approx. 1KU Price (US\$)
SN54163	<a href="#">Datasheet</a> <a href="#">App Notes</a> <a href="#">Samples</a> \Buy	Synchronous 4-Bit Counters	ACTIVE	3.80
SN54180	<a href="#">Datasheet</a> <a href="#">App Notes</a> <a href="#">Samples</a> \Buy	9-Bit Odd/Even Parity Generators/Checkers	ACTIVE	4.85
SN54192	<a href="#">Datasheet</a> <a href="#">App Notes</a> <a href="#">Samples</a> \Buy	Synchronous 4-Bit Up/Down Counters (Dual Clock With Clear)	ACTIVE	4.95
SN54196	<a href="#">Datasheet</a> <a href="#">App Notes</a> <a href="#">Samples</a> \Buy	50/30/100-Mhz Presettable Decade OR Binary Counters/Latches	ACTIVE	8.95
SN54197	<a href="#">Datasheet</a> <a href="#">App Notes</a> <a href="#">Samples</a> \Buy	50/30/100-Mhz Presettable Decade OR Binary Counters/Latches	ACTIVE	4.95

- Aujourd'hui, cette techno est obsolète : ces différentes fonctions sont



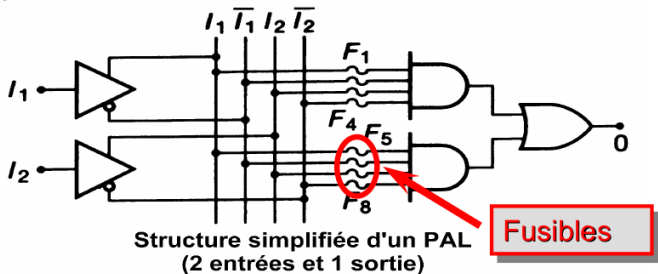
# Familles de PLD

Trois familles principales :

- PAL/GAL (années 1970-1980)
- CPLD (années 1980-1990)
- FPGA (années 1990-2000)

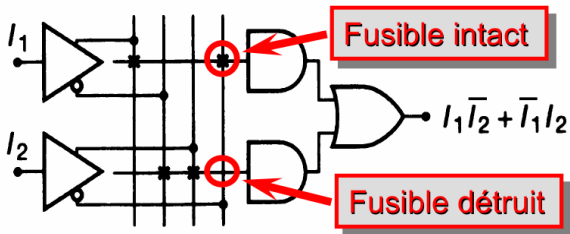
# PAL/GAL

- Basé sur des expressions booléennes sous forme de somme de produits
- Des fusibles permettent de couper les connexions non désirées
- Exemple :



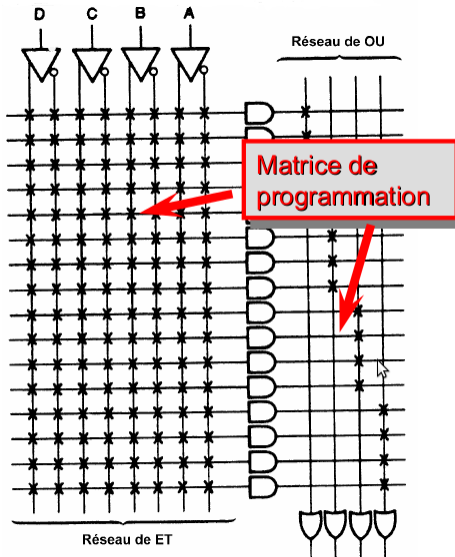
# PAL/GAL

- On représente ces connexions sous forme d'un réseau ligne/colonne
  - colonnes : variables d'entrées (et *feedback* de la sortie)
  - lignes : connexions sur l'opérateur ET
- Exemple : OU Exclusif



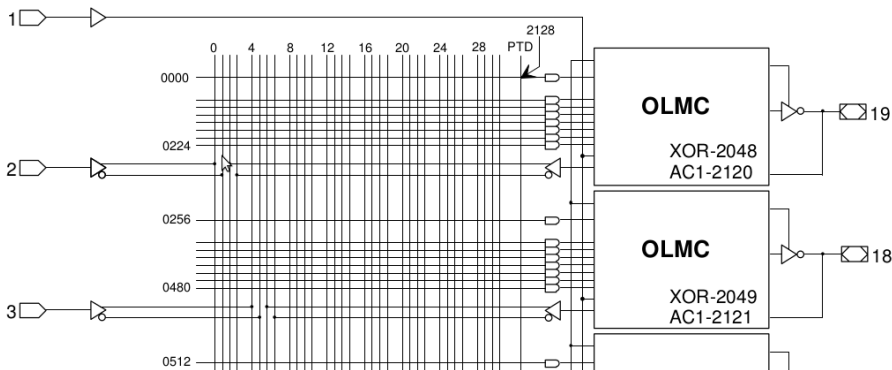
# PAL/GAL : schéma global

- On arrive ainsi au concept de "matrice de programmation"



# PAL/GAL : schéma global

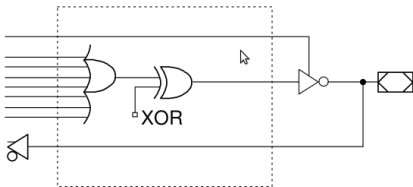
- Les versions modernes de ces composants permettent une personnalisation du comportement via le concept de *Output Logic Macro Cell (OLMC)*



# PAL/GAL : OLMC

- Le bloc de sortie peut fonctionner en 2 modes différents, sélectionnés automatiquement via le fichier de programmation :

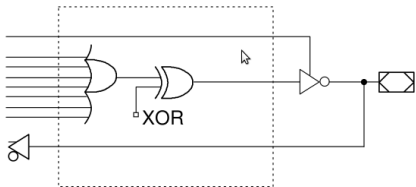
Mode 1 : la sortie implémente une simple "somme de produits", avec en plus une possibilité de sortie "3-états".



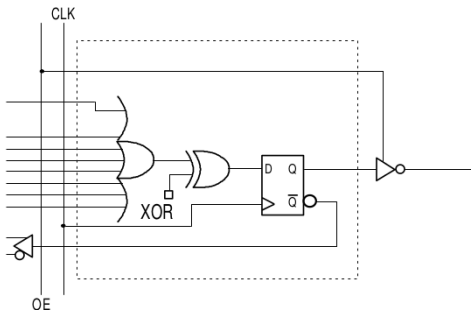
# PAL/GAL : OLMC

- Le bloc de sortie peut fonctionner en 2 modes différents, sélectionnés automatiquement via le fichier de programmation :

Mode 1 : la sortie implémente une simple "somme de produits", avec en plus une possibilité de sortie "3-états".

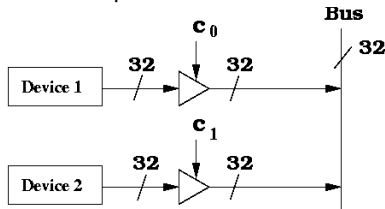


Mode 2 : la sortie est synchronisée sur une horloge commune via une bascule D.



# OLMC : Logique "3-états"

- Le problème : interconnexion de bus

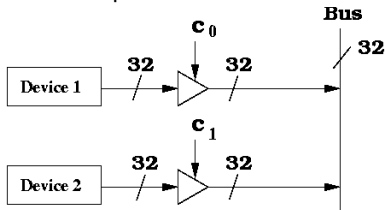


Impossible de connecter des sorties ensemble



# OLMC : Logique "3-états"

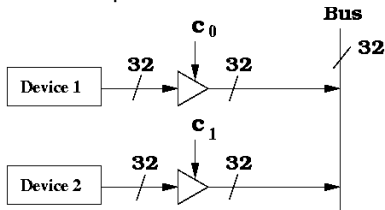
- Le problème : interconnexion de bus



Impossible de connecter des sorties ensemble ("court circuit")

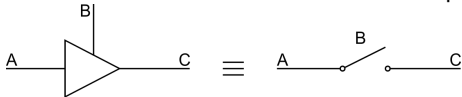
# OLMC : Logique "3-états"

- Le problème : interconnexion de bus



Impossible de connecter des sorties ensemble ("court circuit")

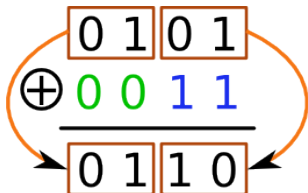
- La solution : circuits doté d'une entrée permettant de déconnecter la sortie



Aussi appelé "High-Z", pour "haute impédance"

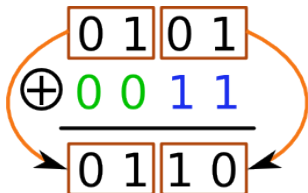
# OLMC : EXOR comme inverseur commandé

- Table de vérité :



# OLMC : EXOR comme inverseur commandé

- Table de vérité :



- Propriété (x est un bit)

$$x \oplus 0 = x$$

$$x \oplus 1 = \overline{x}$$

# OLMC : EXOR comme inverseur commandé

- Table de vérité :

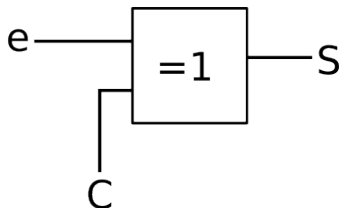
	0	1	0	1
$\oplus$	0	0	1	1
<hr/>				
	0	1	1	0

- Propriété (x est un bit)

$$x \oplus 0 = x$$

$$x \oplus 1 = \overline{x}$$

- Utilisation en inverseur commandé



(commande)

# OLMC : EXOR comme inverseur commandé

- Table de vérité :

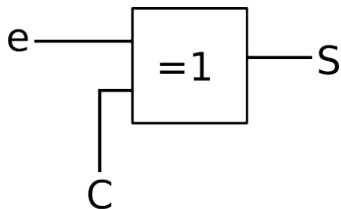
	0	1	0	1
$\oplus$	0	0	1	1
<hr/>				
	0	1	1	0

- Propriété (x est un bit)

$$x \oplus 0 = x$$

$$x \oplus 1 = \bar{x}$$

- Utilisation en inverseur commandé



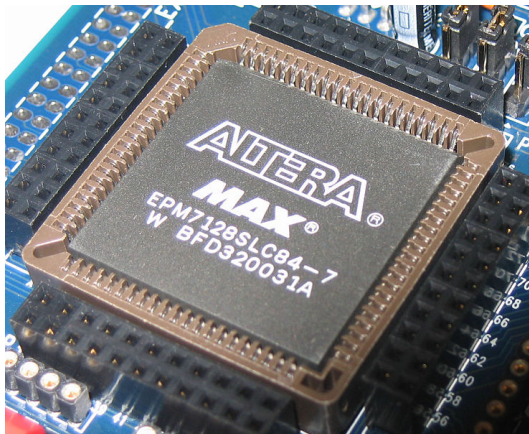
$$C=0 \Rightarrow S=e$$

$$C=1 \Rightarrow S=\bar{e}$$

(commande)

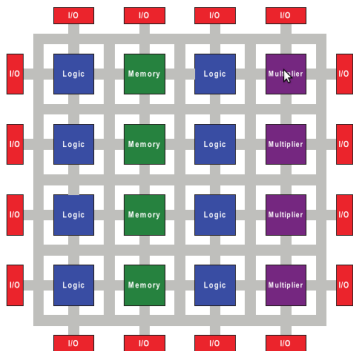
# CPLD

- Basé sur le concept des PLD, avec un plus grand nombre de portes (réseau de connexions + macrocell)
- Nb de portes : de quelques milliers à dizaines de milliers.



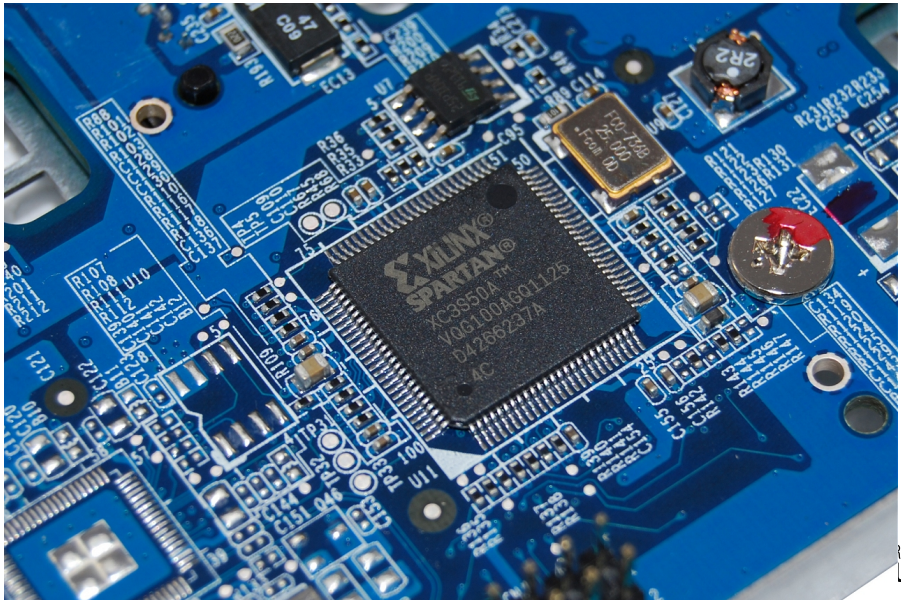
# FPGA

- Inventé par Xilinx en 1985 : XC2064 : 1200 portes, 58 broches
- Aujourd'hui : Plusieurs millions de portes logiques et de bascules, on parle de *Sea of gates*.
- Deux constructeurs : Xilinx & Altera (80% du marché)
- Peuvent intégrer de la mémoire vive (RAM).
- Plus souples au niveau configuration que les CPLD

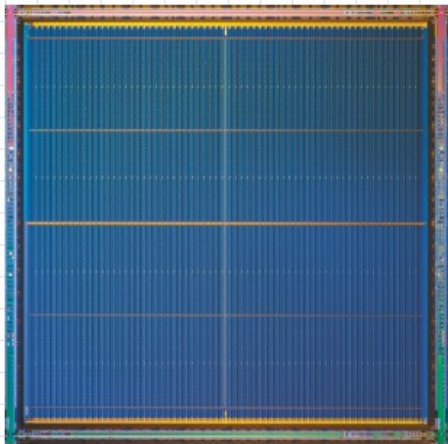




# FPGA



# FPGA vs. Pentium 4



# Evolution : *Soft Processor*

Implémentation d'un CPU dans un FPGA, via un langage de description matérielle (VHDL ou Verilog).

Voir [https://en.wikipedia.org/wiki/Soft\\_microprocessor](https://en.wikipedia.org/wiki/Soft_microprocessor)

