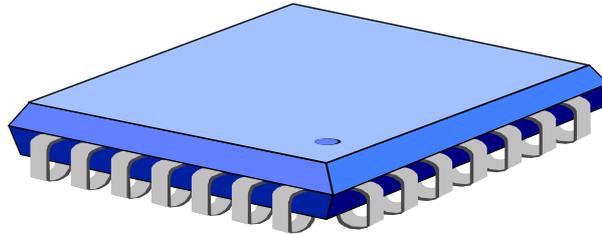


# Les circuits logiques programmables FPGAs

---



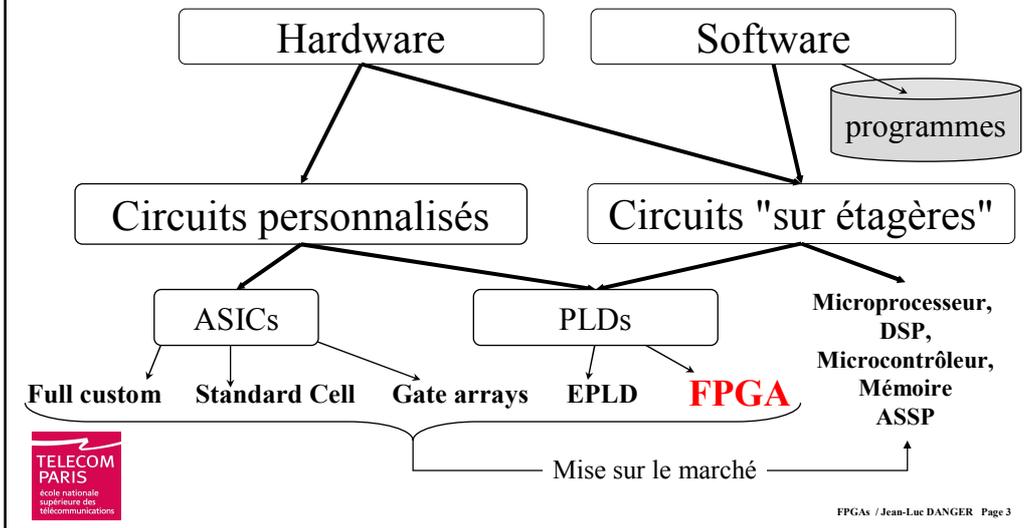
Jean-Luc Danger

## PLAN

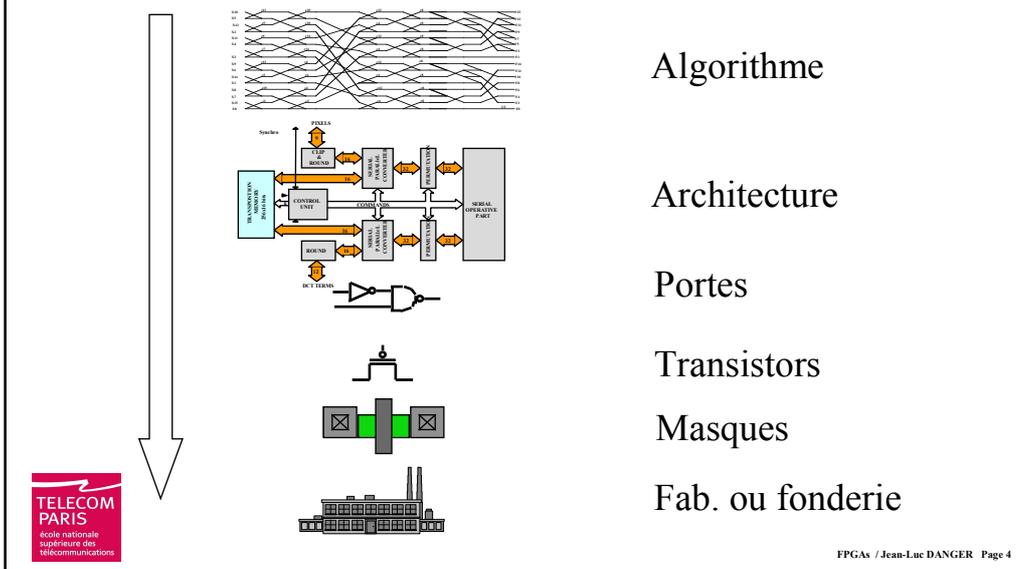
---

- ➔  **Place des FPGAs dans les Filières technologiques**
-  **Architectures de la logique**
-  **Architectures des blocs embarqués**
-  **Architectures d'interconnexion**
-  **Architectures de configuration**
-  **Consommation et robustesse**
-  **Méthodes de développement**
-  **Analyse des architectures des constructeurs**

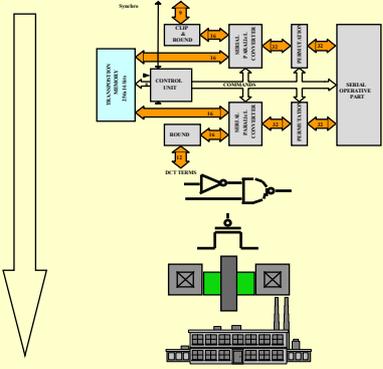
## Filières technologiques



## Conception d'une application Approche descendante ou "Top-Down"

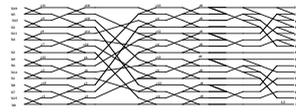


## Circuits "sur étagères"



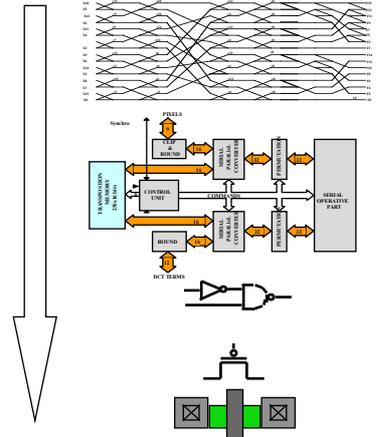
- Microprocesseurs
- Microcontrôleurs
- DSP
- Mémoires
- ASSP

*Intel, TI, Freescale, STM, ...*



Algorithme : **software**

## Circuits "Full Custom" ou "cousu main"



L'algorithme est **sur la puce** :

- Performances optimales
- Coût optimal
- Système complet "SoC"
- Développement long et coûteux
- Réservé aux très forts volumes (amortissement du développement)

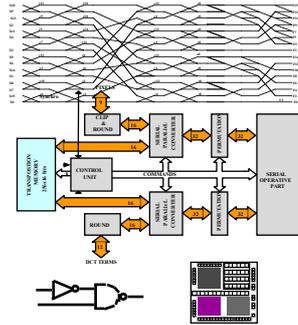
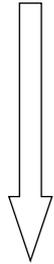
*Intel, TI, STM, Atmel, Freescale, NXP, ...*



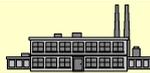
## Circuits ASIC "standard cell" ou précaractérisés



Bibliothèque "Standard cell"  
STM, Atmel, TSMC, UMC



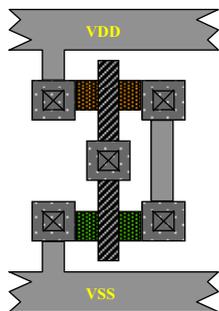
- Accessible aux entreprises "Fables"
- Bon Compromis performances/coût à fort volume



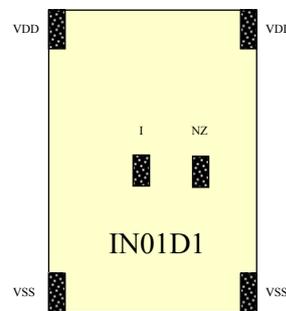
STM, Atmel, TSMC, UMC, LSI Logic, ...

## Standard cell

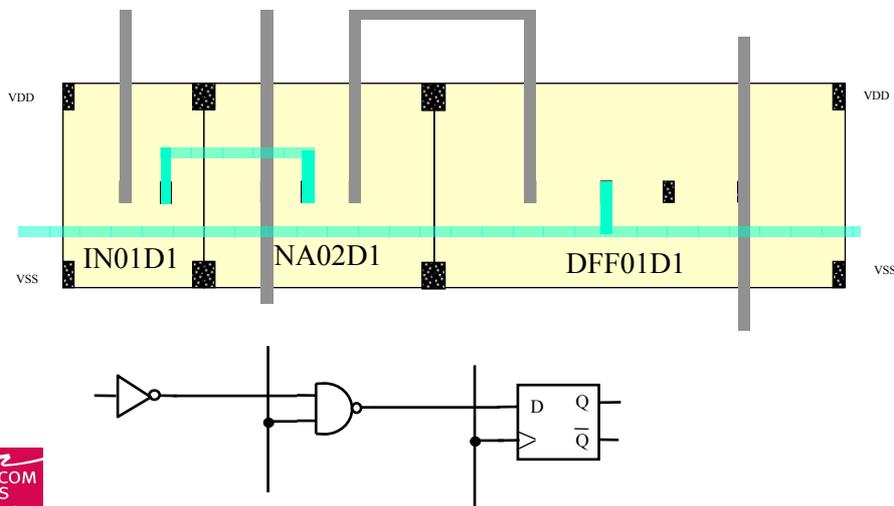
25 objets



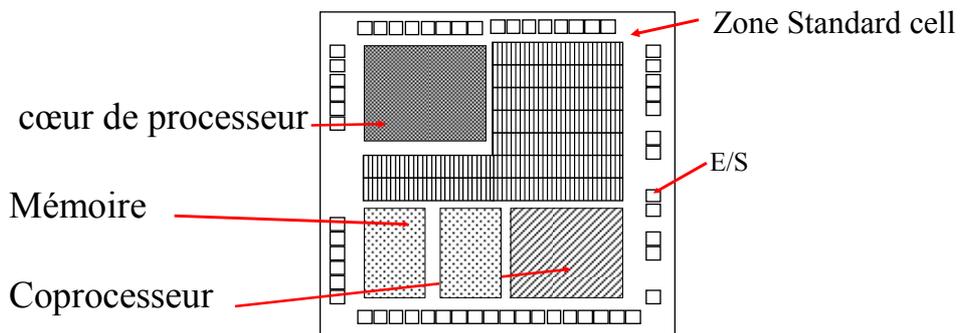
1 objet



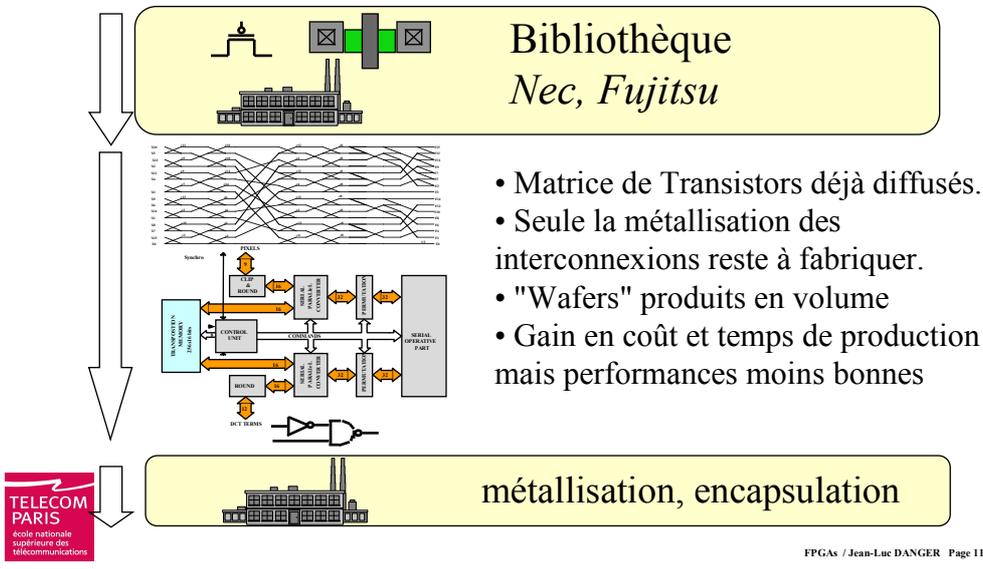
## Interconnexion des cellules standard



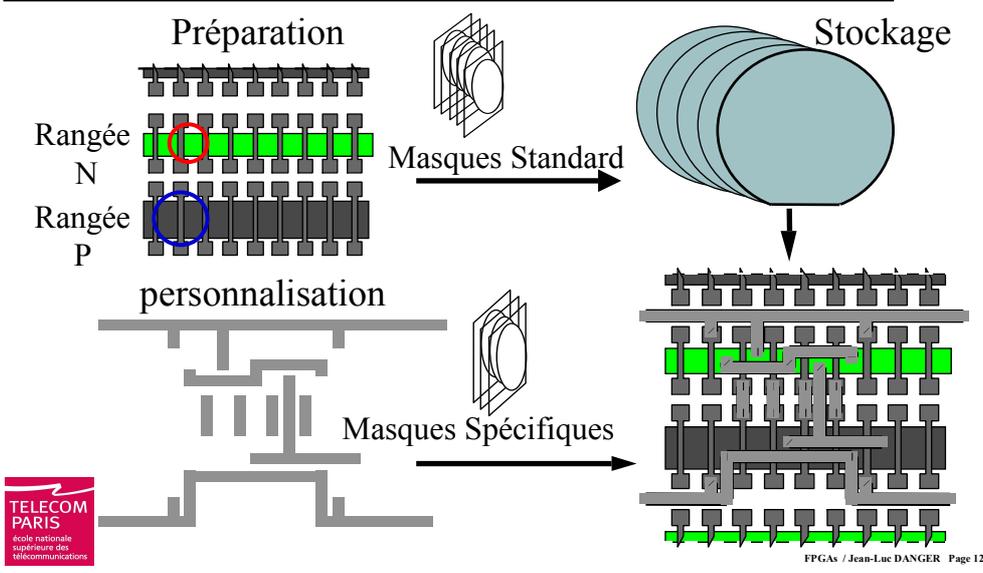
## Floorplan



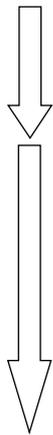
## Circuits ASIC "Gate Arrays" ou prédiffusés

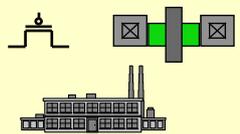


## Fabrication de Gate arrays

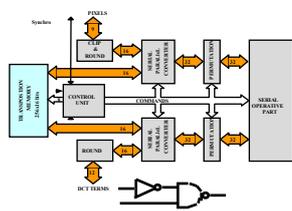
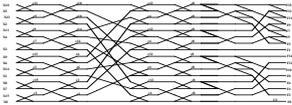


# Circuits programmables : PLD "Programmable Logic Devices"





**FPGA, EPLD**  
*Altera, Xilinx, Actel, Lattice*



- Circuit avec interconnexion et cellules programmables.
- Mémoire interne de programmation
- Circuit "sur étagères" testé.
- Coût de développement faible
- Coût à la pièce élevé à fort volume
- Performances à priori + faibles
- Utilise les dernières technologies

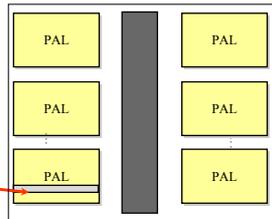
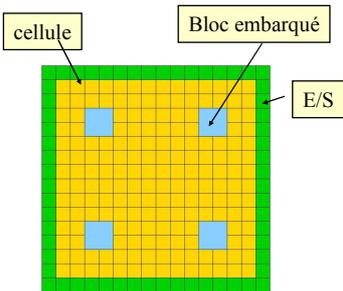


## PLD : FPGA et EPLD

- FPGA :**

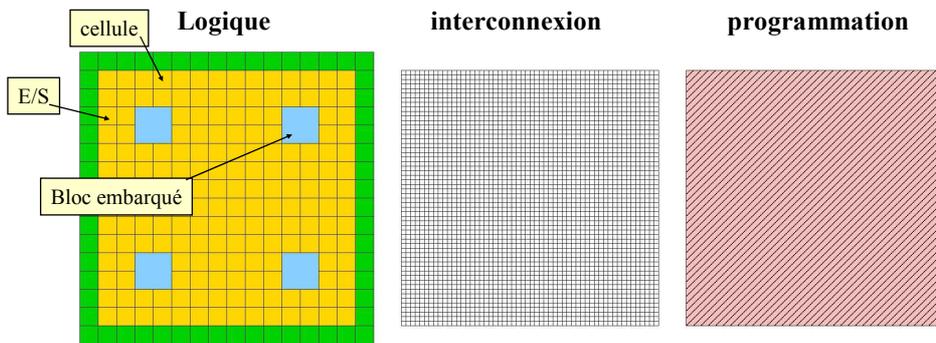
  - Circuit de type "Gate Array"
  - Matrice de "cellules"
  - Facilement extensible
    - Quelques millions de portes
  - Performants
    - >500MHz (Virtex5 et StratixIII)

- EPLD :**
- Circuit à base de "macrocellules"
  - Dédié uniquement au contrôle



# Architecture générique d'un FPGA

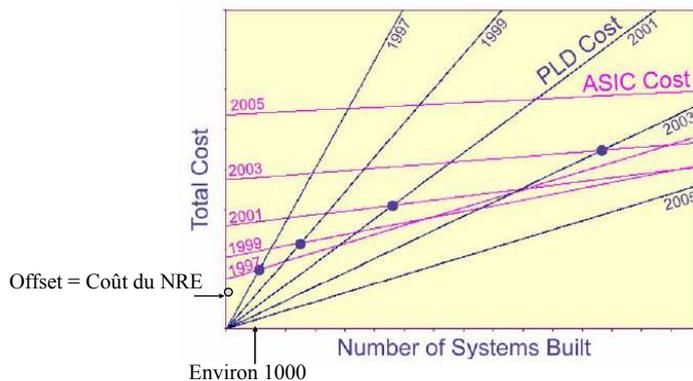
3 plans virtuels superposés :



Surface environ de 20 à 50 fois plus grosse qu'un ASIC à fonction identique dans une même technologie



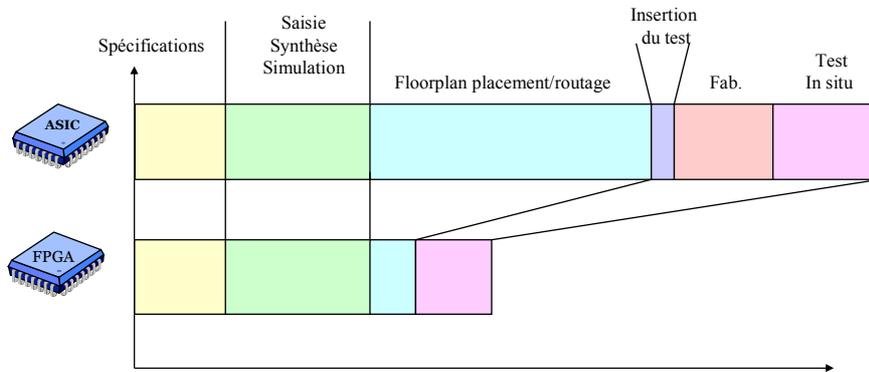
# Comparaison du coût ASIC/FPGA



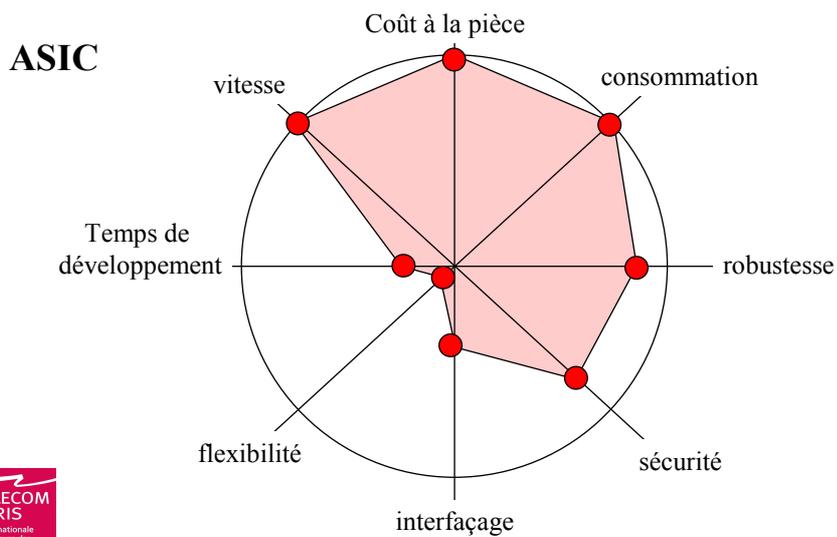
Nick Tredennick, Brian Shimamoto. *The Rise of Reconfigurable Systems*. In proceeding of Engineering of Reconfigurable Systems and Algorithms, ERSA'2003. June 23-26, Las Vegas, Nevada, USA.



## Comparaison du temps de développement ASIC/FPGA

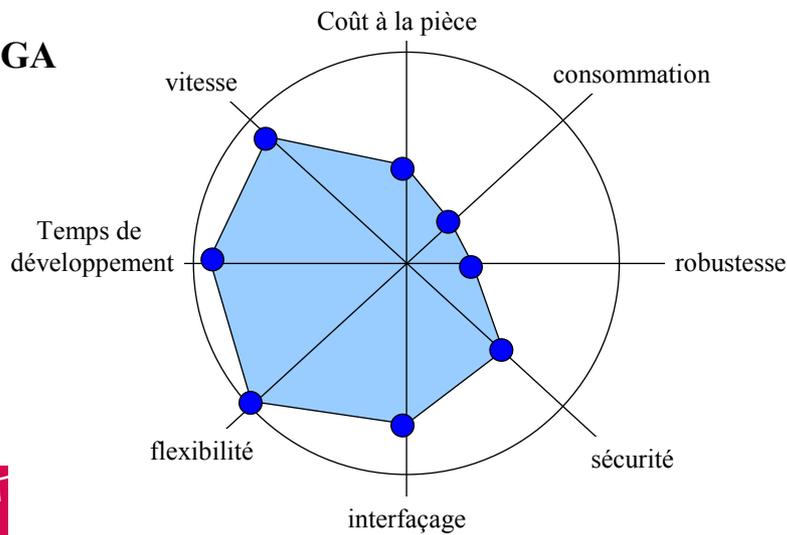


## Comparaison multi-critères ASIC/FPGA



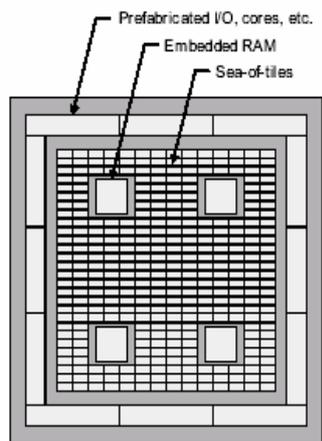
## Comparaison multi-critères ASIC/FPGA

FPGA



## Solution mixte 1 : "Structured ASIC"

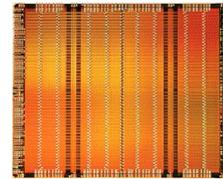
- ASiCs précaractérisés
- Matrice d'éléments structurés
- Adaptés à recevoir une netlist FPGA
- Gain si volume important (coût de NRE si volume faible)



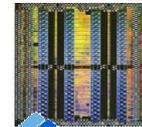
## Structured ASIC : exemple ALTERA

### Hardcopy II

- ❑ Pas de mémoire de programmation
- ❑ Métallisation à la place
- ❑ La mémoire RAM embarquée peut être corrigée par un code ECC
- ❑ Migration en moins de 18 semaines
- ❑ Gain de 70% en surface et coût



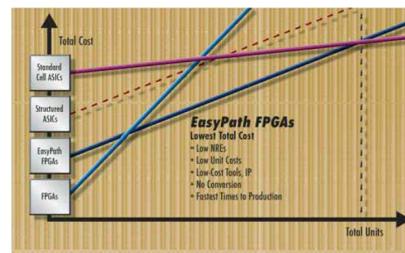
Remove Configuration



HardCopy II

## Solution mixte 2 : XILINX "Easy Path"

- ❑ Les FPGAs ne sont pas modifiés
- ❑ Le test ne prend en compte que la fonctionnalité du client
  - ❑ Forte réduction du coût



Total Cost Driver	Structured ASICs	Xilinx EasyPath
Time-to-Cost Reduction	20 to 24 weeks	8 to 12 weeks
NRE Costs	\$300K to \$400K	\$100K
Cost of Requalification	\$100K to \$500K	\$0
Engineering Costs	\$250K to \$300K	\$0
Cost of Design Tools	\$100K to \$200K	\$0
Unit Costs	Lowest	Low
Cost of Respin	High	Low

Estimation du coût selon XILINX

## Solution mixte 3 : FPGA embarqué

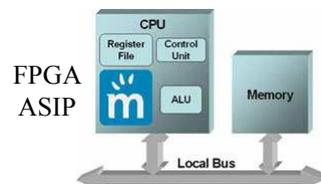
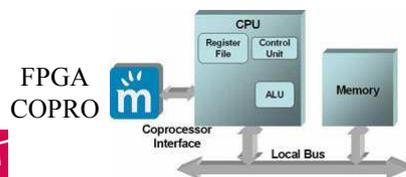
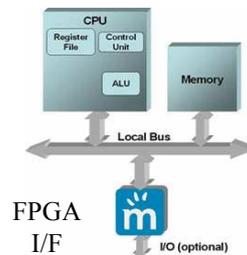
### Cœurs FPGA dans les ASICs

#### exemples :

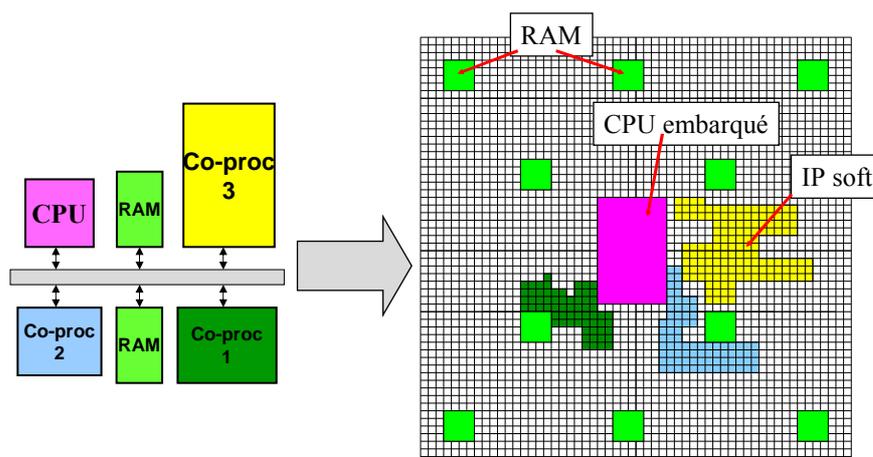
- Partenariat Xilinx/IBM,STM
- Actel Varicore

### Marché naissant

#### Start-up : Menta, M2000

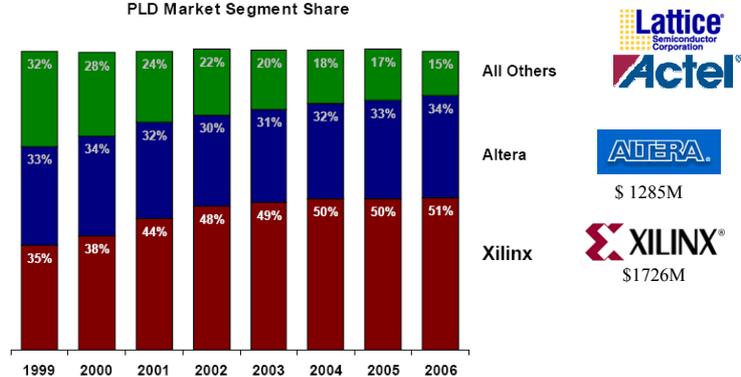


## Solution mixte 4 : Hard IPs dans les FPGAs



## Marché des FPGAs

PLD Market Segment Share



\$ 1285M

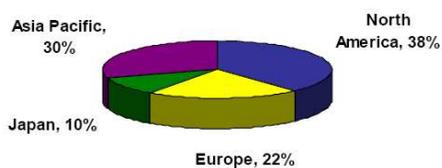


\$1726M

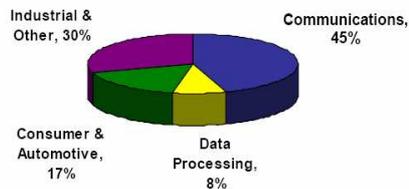


## Géographie et types des marchés FPGAs

Revenue by Geography



Revenue by End Market



**Marché FPGA = marché ASIC dédié**

- Prototypage
- Production Petits volumes
- Produit innovant avec "Time to Market" rapide
- Contrainte forte de flexibilité ou maintenabilité



# PLAN

- ☞ Place des FPGAs dans les Filières technologiques
- ➔ ☞ Architectures de la logique
- ☞ Architectures des blocs embarqués
- ☞ Architectures d'interconnexion
- ☞ Architectures de configuration
- ☞ Consommation et robustesse
- ☞ Méthodes de développement
- ☞ Analyse des architectures des constructeurs

## Comment générer une fonction logique quelconque?

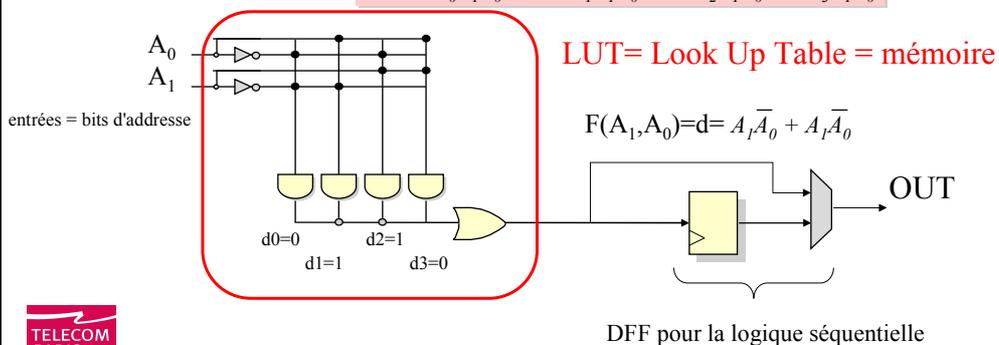
### La LUT

Fonction à 2 entrées :

$$f(A_1, A_0) = f(0,0)\bar{A}_1\bar{A}_0 + f(0,1)A_1\bar{A}_0 + f(1,0)\bar{A}_1A_0 + f(1,1)A_1A_0$$

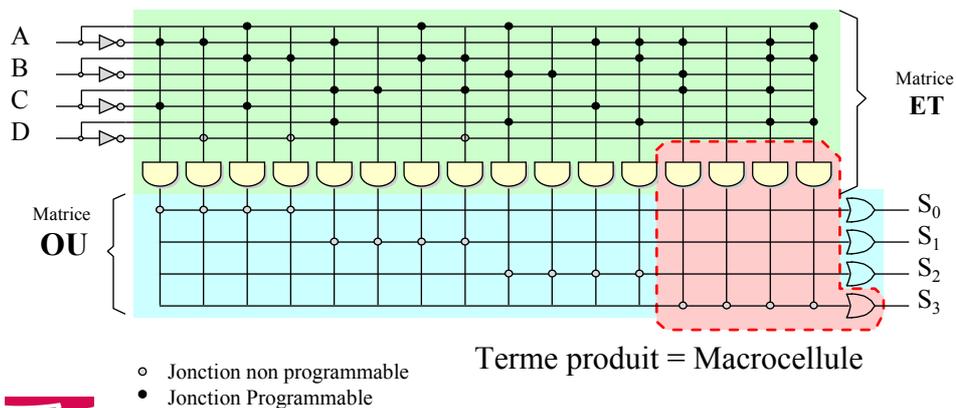
Mémoire 4 bits :

$$d = d_0\bar{A}_1\bar{A}_0 + d_1A_1\bar{A}_0 + d_2\bar{A}_1A_0 + d_3A_1A_0$$



## Comment générer une fonction logique quelconque

### La macrocellule



## Comparaison LUT/macrocellule

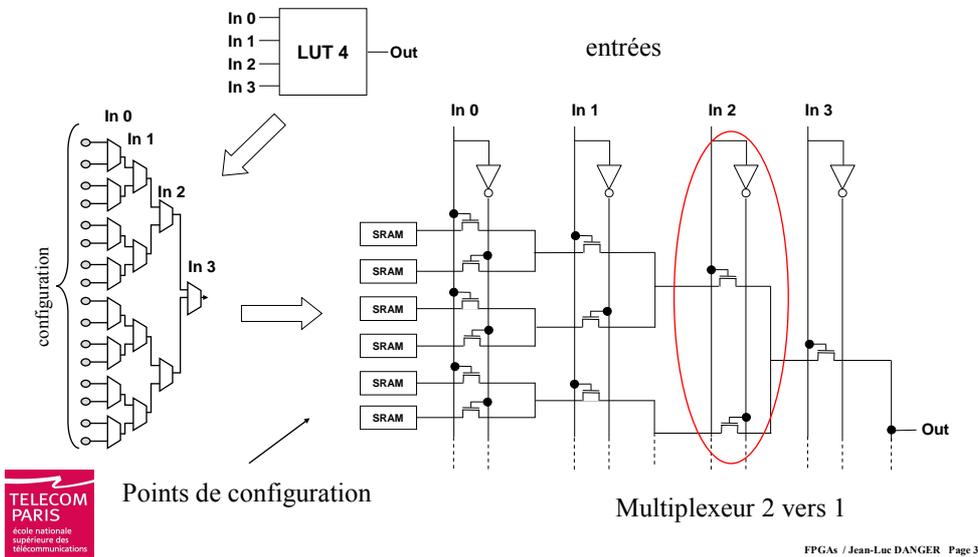
### Macrocellule :

- Grand nombre de variables d'entrée possibles (qqz dizaines) par terme produit
- Nombre de termes limités (<10)
- Adapté à la logique de contrôle

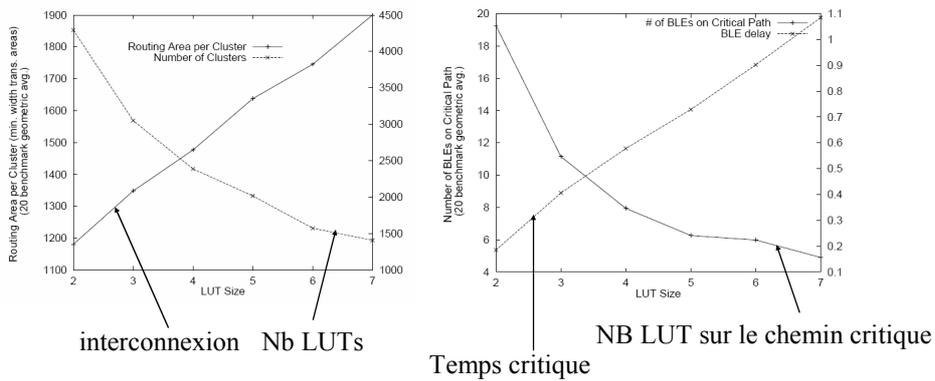
### LUT

- N'importe quelle fonction logique de N entrées ( $2^N$ ) mais N faible
- Calcul générique avec une logique spécifique pour l'arithmétique

# Architecture d'une LUT



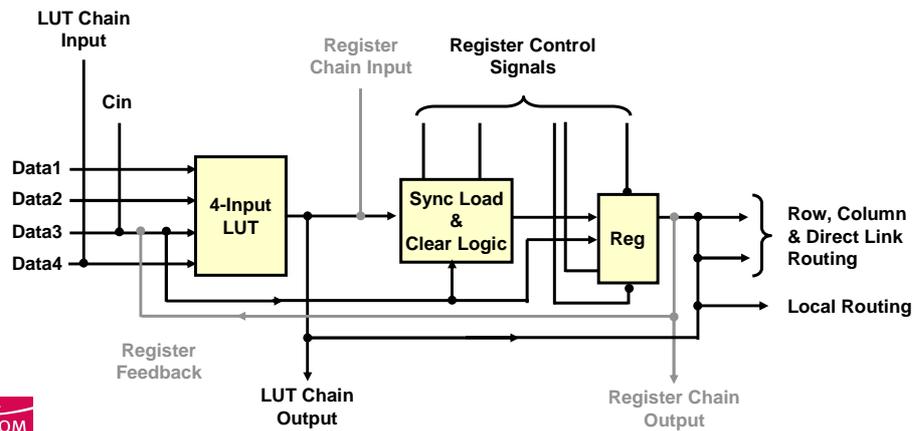
# Combien d'entrées/LUT ?



Elias Ahmed, Jonathan Rose: The effect of LUT and cluster size on deep-submicron FPGA performance and density. IEEE Trans. VLSI Syst. 12(3): 288-298 (2004)

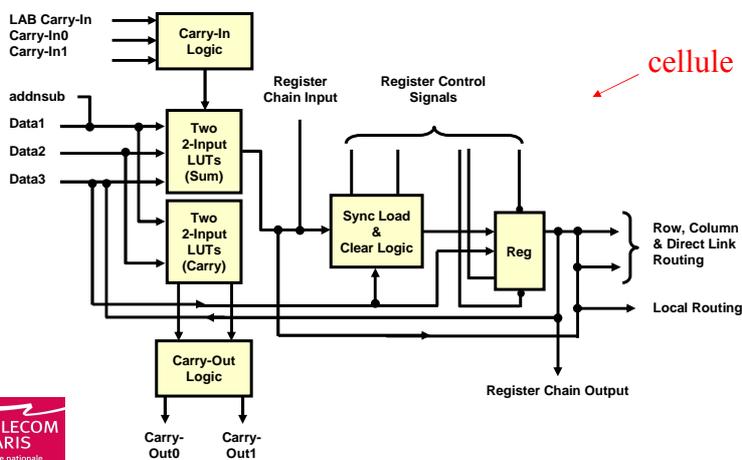
# Cellule à LUT : ALTERA STRATIX

## mode normal

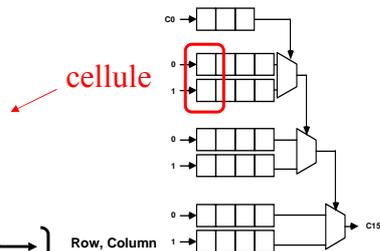


# Cellule à LUT : ALTERA STRATIX

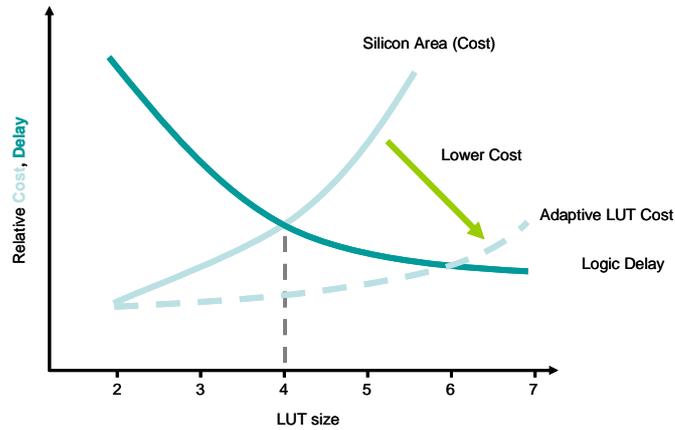
## mode arithmétique



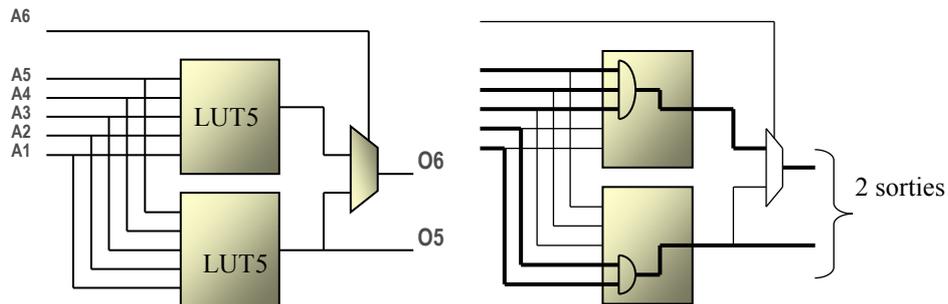
Algorithme : CARRY SELECT



## Evolution du nombre d'entrées des LUTs

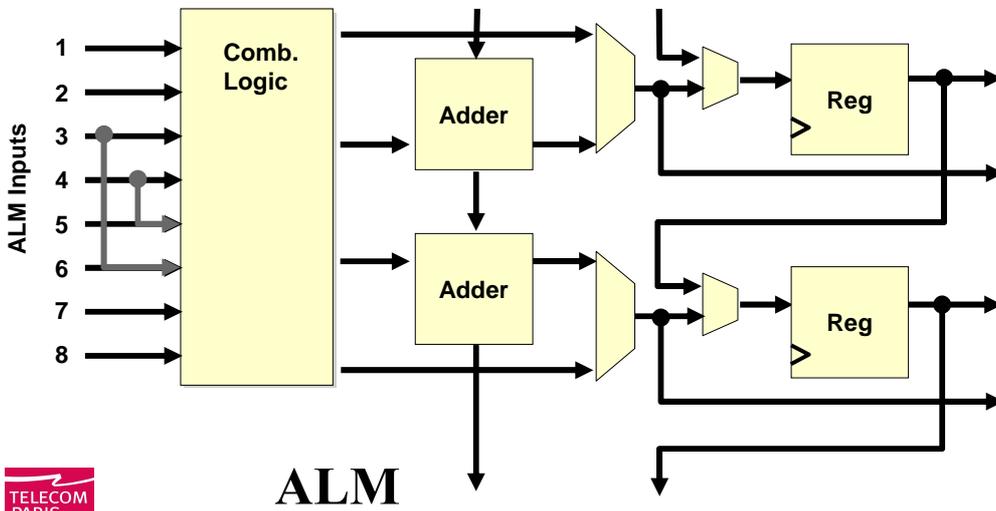


## XILINX Virtex5 : LUT6

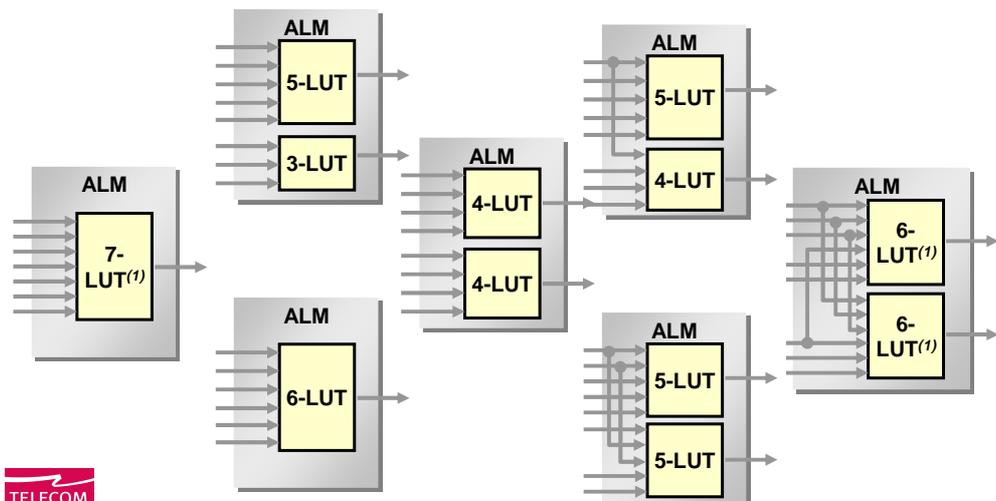


Configuration en 1 LUT6 ou 2 LUT5

## ALTERA Adaptive Logic Module (STRATIXII)

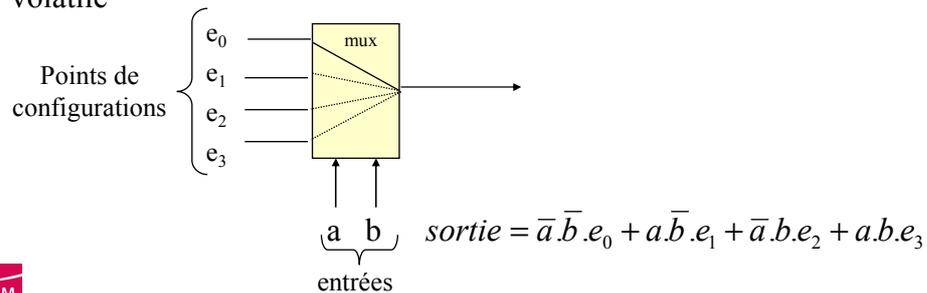


## StratixII : Configurations ALM



## Cellule FPGA : MUX

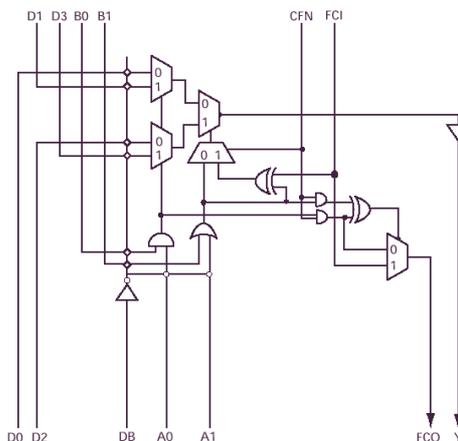
- Equivalut à une LUT câblée
- Un combinaison de MUX permet d'avoir un grain de calcul plus fin (donc une surface réduite).
- Utilisé en général avec une technologie de programmation non volatile



## Cellule à MUX(1)

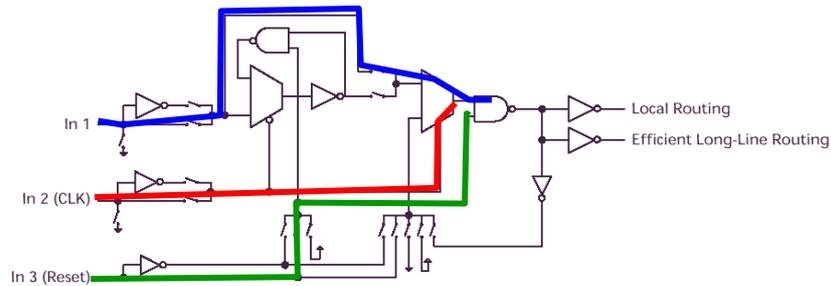
### cellule C famille AXcelerator de ACTEL

- Nombreuses combinaisons
- Logique pour l'arithmétique
- Cellule R (DFF) indépendante
- Technologie Antifusible



## Cellule à MUX : ProASIC en mode combinatoire

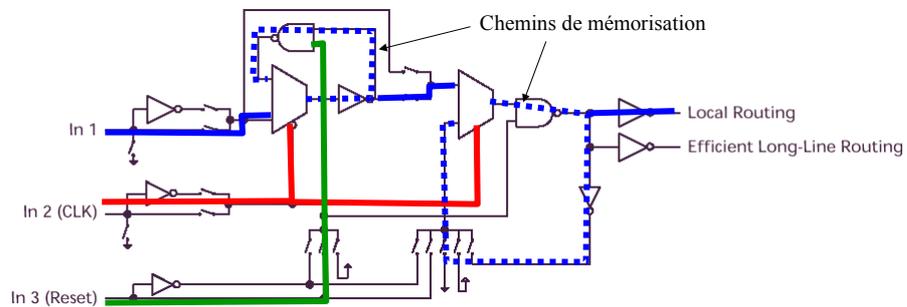
- Exemple : ProASIC<sup>PLUS</sup> de ACTEL (technologie FLASH)



Exemple 1 : porte NAND à 3 entrées

## Cellule à MUX ProASIC : mode séquentiel

- Exemple : ProASIC<sup>PLUS</sup> de ACTEL (technologie FLASH)



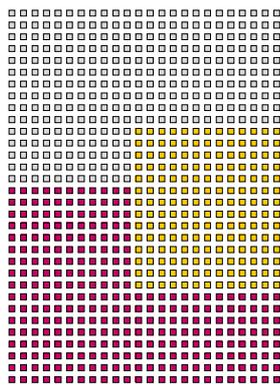
Exemple 2 : bascule D avec reset asynchrone

# PLAN

- ☞ Place des FPGAs dans les Filières technologiques
- ☞ Architectures de la logique
- ➔ ☞ Architectures des blocs embarqués
- ☞ Architectures d'interconnexion
- ☞ Architectures de configuration
- ☞ Consommation et robustesse
- ☞ Méthodes de développement
- ☞ Analyse des architectures des constructeurs

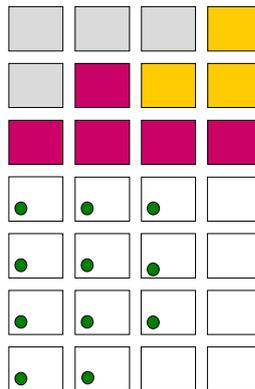


## Taille des cellules Nécessité des blocs gros grains embarqués



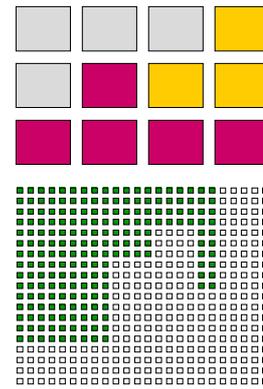
grain Fin

Tout est utilisé



gros grain

Sous utilisation



Mixte

Utilisation optimale



## Nécessité des mémoires

### 📄 Besoins dans toutes les applications de traitement

- ❑ **RAM multi-accès**
  - Exemple : FFT
- ❑ **ROM**
  - Exemple : Coefficients de filtres
- ❑ **LUT**
  - Cellule de calcul gros grain
  - Pratique pour la synthèse de grosse machines à états
- ❑ **FIFO**
  - Interface multicaudence

### 📄 Mais besoin de différentes tailles

## Type de mémoire

### 📄 Distribuée

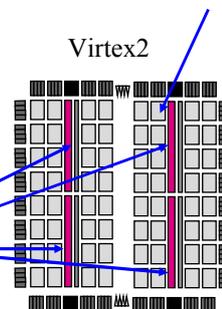
Utilisation des LUTs des cellules en mode mémoire

Exemple : VirtexII    1 Slice = 2 LUT 4 entrées    ➡    1 mémoire 32 bits

### 📄 Embarquée

La mémoire est un bloc spécifique

Embedded Array Block  
18Kbits



# Type d'accès

## Exemple : Xilinx Virtex5

	Each 18K	
<b>True dual-port</b>	16Kx1, 8Kx2, 4Kx4, 2Kx9, 1Kx18	2 independent read and write ports
<b>Simple dual-port</b>	16Kx1, 8Kx2, 4Kx4, 2Kx9, 1Kx18, 512x36	1 read & 1 write port Read AND write in 1 cycle
<b>Single-port</b>	16Kx1, 8Kx2, 4Kx4, 2Kx9, 1Kx18, 512x36	1 read & 1 write port Read OR write in 1 cycle Backwards compatible to V4



Pour FIFO

# Evolution des tailles mémoires STRATIX

Memory Functions	Stratix II Devices	Stratix III Devices*
<ul style="list-style-type: none"> <li>Processor code storage</li> <li>Packet buffers</li> <li>Video frame buffers</li> </ul>	MRAM 576K Bits	M144K 144K Bits
<ul style="list-style-type: none"> <li>General purpose memory</li> </ul>	M4K 4.5K Bits	M9K 9K Bits
<ul style="list-style-type: none"> <li>Shift registers</li> <li>Small FIFO buffers</li> <li>Filter delay lines</li> </ul>	M512 576 Bits	MLAB 640 Bits

## Nécessité des blocs de calcul arithmétique

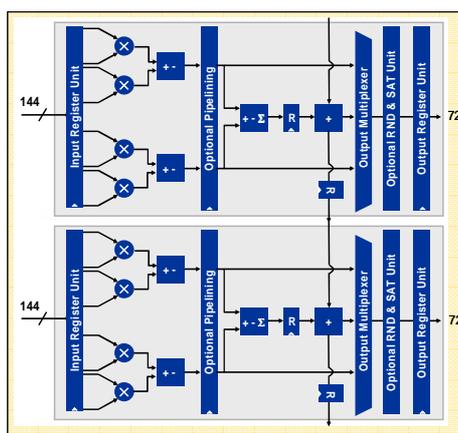
### Besoins forts pour toutes les applications de traitement du signal

- ❑ *Opérateur MACs (FIR, IIR, FFT,...)*
- ❑ *Opérations non linéaires*
  - Troncature, arrondi, saturation
- ❑ *Précision des calculs entre 4 et 32 bits*
  - Rarement besoin des flottants

### Les LUT ne peuvent réaliser que des additions rapides

- ❑ *Multiplieur  $N.N \Rightarrow$  environ  $N^2$  LUTs 4 entrées*

## Exemple : ALTERA StratixIII DSP block



### Basic Multiplier Modes

- ❑  $8 \times (9 \times 9)$
- ❑  $6 \times (12 \times 12)$
- ❑  $4 \times (18 \times 18)$
- ❑  $2 \times (36 \times 36)$
- ❑  $2 \times$  complex ( $18 \times 18$ )

### Sum Modes

- ❑  $4 \times$  Sum of Two ( $18 \times 18$ )
- ❑  $2 \times$  Sum of Four ( $18 \times 18$ )

### Accumulation

- ❑  $2 \times$  Acc

### Cascade Modes

- ❑ *Input Cascade*
- ❑ *Output Cascade*

### Rounding

- ❑ *Unbiased and Biased*

### Saturation

- ❑ *Asymm and Symmetrical*

### Barrel Shifter

- ❑ *Arithmetic, Logical and Rotation*

## Nécessité de blocks de gestion d'horloge

### Problèmes

- ❑ *Skew d'horloge*
  - Du à la forte charge et la longueur des lignes
- ❑ *Fréquence interne différente de la fréquence externe*
  - Nécessité d'un asservissement fréquence
- ❑ *Besoin de plusieurs horloges corrélées*

### Solutions 2 niveaux :

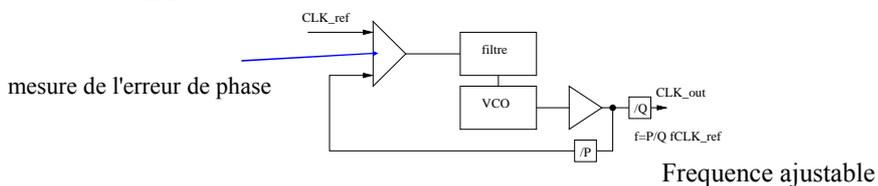
- ❑ *Interconnexion*
    - lignes dédiées pour les horloges
  - ❑ *Bloc dédié à la génération :*
    - Asservissement sur l'extérieur
    - Compensation du skew d'horloge
    - Multiplication ou division
- Synthèse d'horloge  $f_{out} = P/Q f_{in}$*



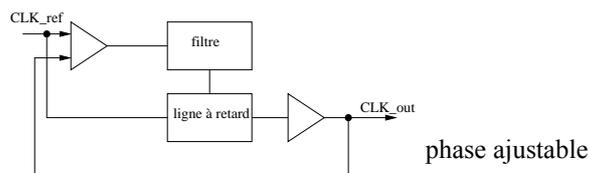
## Principe de génération d'horloge

### Système asservi en fréquence sur une oscillateur externe

#### ❑ PLL

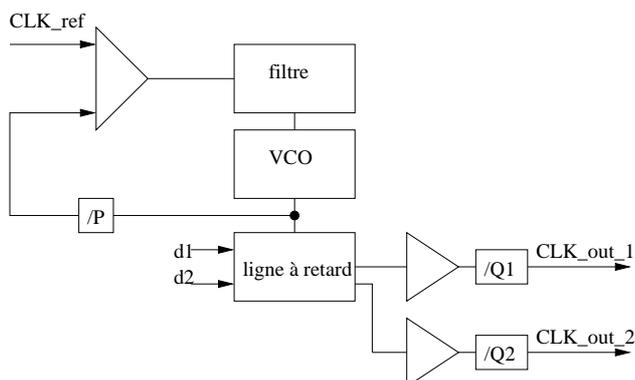


#### ❑ DLL



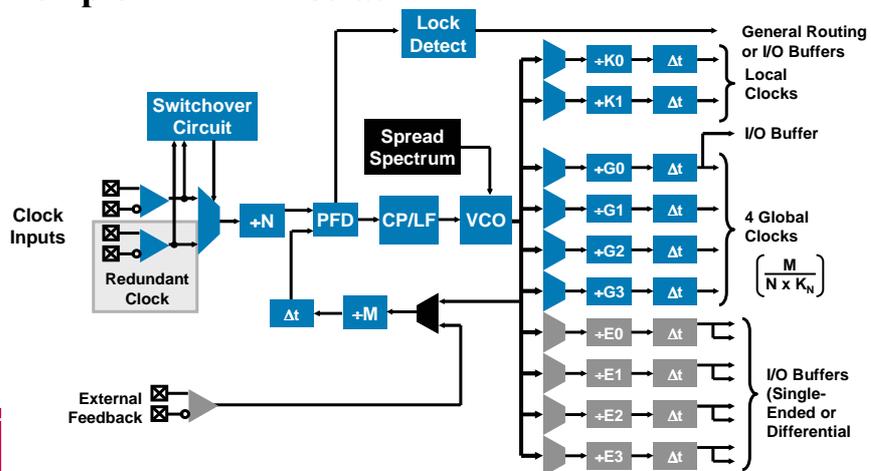
# Bloc amélioré de génération d'horloge

## Ajustement en fréquence et phase sur plusieurs horloges



# PLL

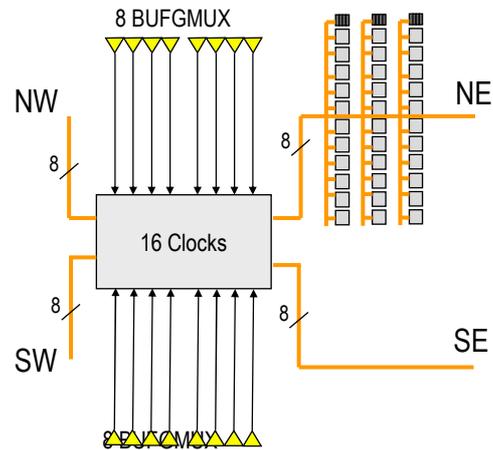
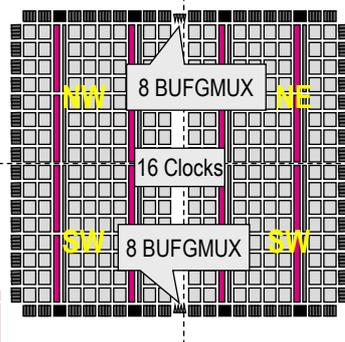
## Exemple ALTERA Stratix



## Interconnexion : Réseau d'horloges

### Exemple : XILINX VIRTEX II

16 équipotentielles réparties sur 4 quadrants



## Cellule E/S

### Cellule configurable

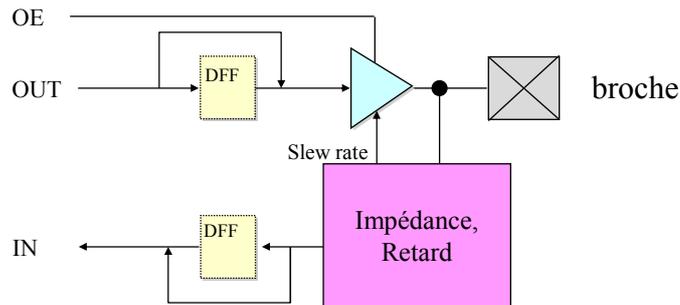
#### De nombreux paramètres et standards d'interface

- Mode Terminaison simple
- Mode Différentiel
- Synchronisation avec les horloges
- Sérialisation/désérialisation
- Ajustement du Slew Rate
- Adaptation d'impédance
- Adaptation du retard

### Tension d'alimentation spécifique

## Architecture cellule E/S

### Mode simple



## Les processeurs intégrés

### Peu d'offre en IP Hard :

- ❑ *XILINX VIRTEX : IBM PPC405*

### Beaucoup en Soft :

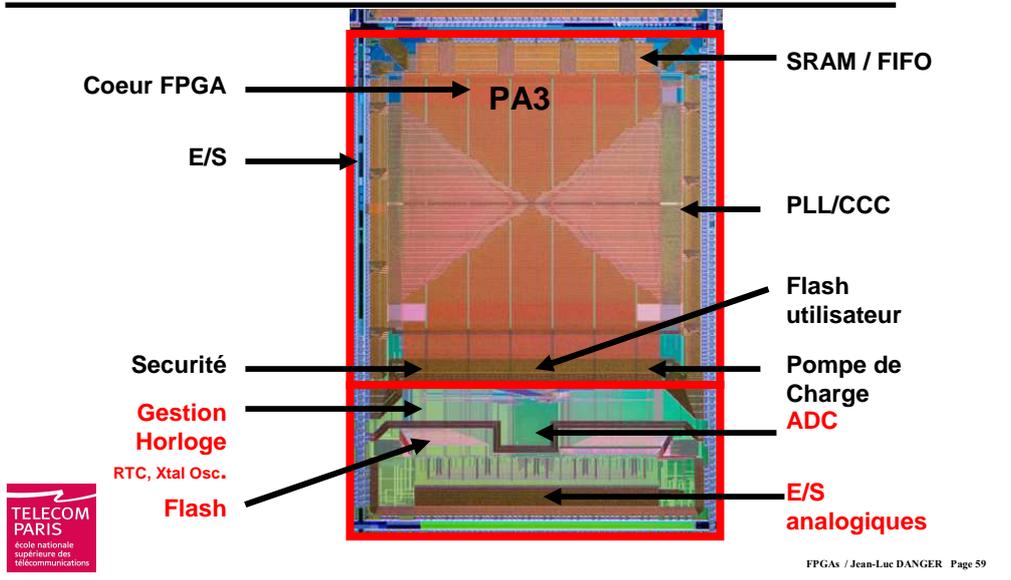
#### ❑ *Propriétaires*

- XILINX : MicroBlaze 32 bits
- ALTERA : NIOS 32 bits
- ACTEL+ARM : ARM7

#### ❑ *Libres. Exemples :*

- 32 bits RISC : Leon (SPARC 32 bits)
- 8 bits CISC simple : 6502

## Blocs embarqués analogiques : ACTEL Fusion



## Les IPs softs

### Bibliothèques très riches

Communications	Bus Interface	DSP	Cœur de Processeurs, Périphériques
ADPCM (u-law, a-law)	PCI Target	Color Space Converter	Nios™ Processor
ATM Controller	PCI Master-Target	Correlator	Tensilica X-tensa Processor
CRC	PCI-X	Digital Modulator	PalmChip Bus
Ethernet MAC (10/100/Gigabit)	CAN Bus	Discrete Cosine Transform	SDRAM Controller
HDLC Protocol Core	IIC Master & Slave	Fast Fourier Transform	DDR-SDRAM Controller
IMA Controller	IEEE 1394	FIR Compiler	QDR-SDRAM Controller
SONET/SDH Framer	PowerPC Bus Arbiter	IIR Filter	8237 DMA Controller
T3/E3 Framer	PowerPC Bus Master	Image Processing Library	8255 Peripheral Interface
Packet Over SONET Processor	PowerPC Bus Slave	NCO	8259 Interrupt Controller
Telephony Tone Generator	USB Function Controller	Reed Solomon Encoder/Decoder	8254 Timer/Counter
Utopia Master & Slave	USB Host Controller	Interleaver/Deinterleaver	8051, 6502, Z80
POS-PHY Interface	...	Viterbi Decoder	...
		Turbo Decoder	
		...	

TELECOM PARIS  
école nationale supérieure des télécommunications

FPGAs / Jean-Luc DANGER Page 60

# PLAN

---

- ☞ **Place des FPGAs dans les Filières technologiques**
- ☞ **Architectures de la logique**
- ☞ **Architectures des blocs embarqués**
- ➔ ☞ **Architectures d'interconnexion**
- ☞ **Architectures de configuration**
- ☞ **Consommation et robustesse**
- ☞ **Méthodes de développement**
- ☞ **Analyse des architectures des constructeurs**

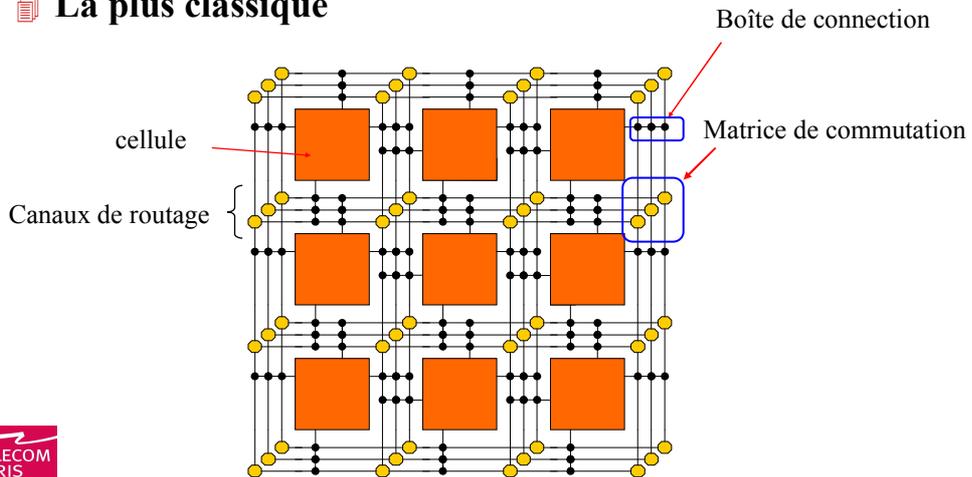
# Principales topologies d'interconnexion

---

- ☞ **Matrice (ou Mesh)**
- ☞ **Matrice hiérarchique (ou Island)**
- ☞ **En arbre**
- ☞ **Mixte Matrice/arbre**

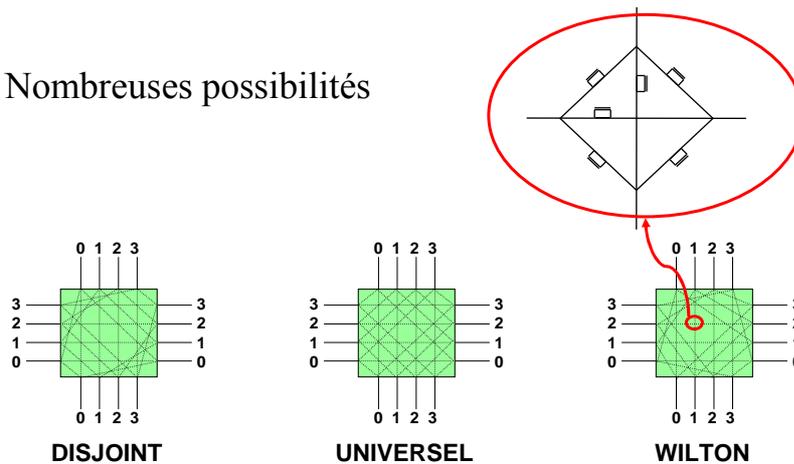
# Topologie d'interconnexion en matrice

## La plus classique



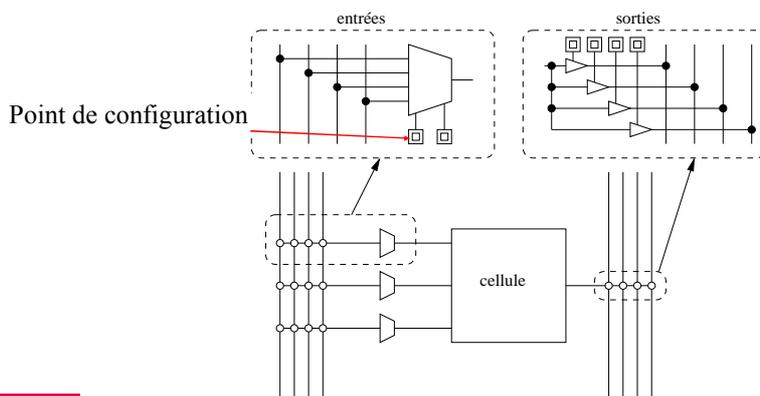
# La matrice de commutation

Nombreuses possibilités



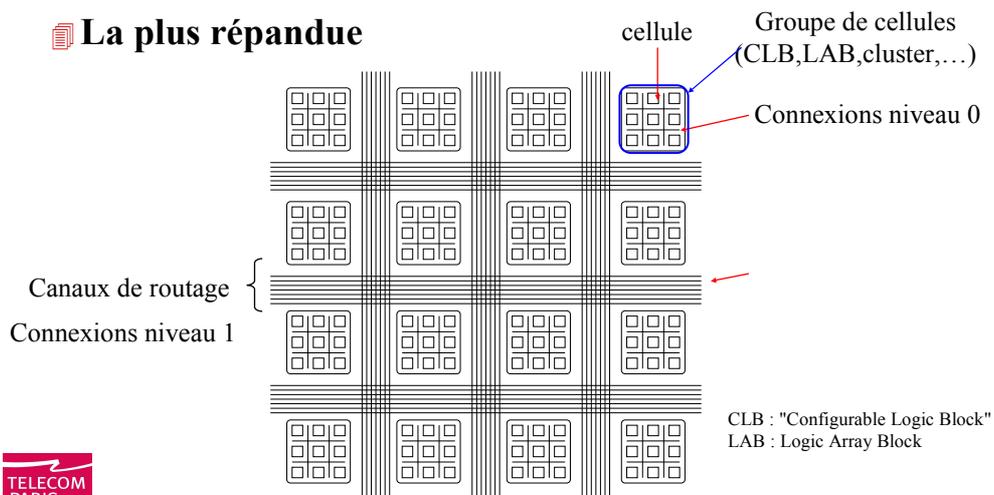
## La boîte de connexion

Les points de configuration des entrées sont en  $\log(N)$  pour N entrées

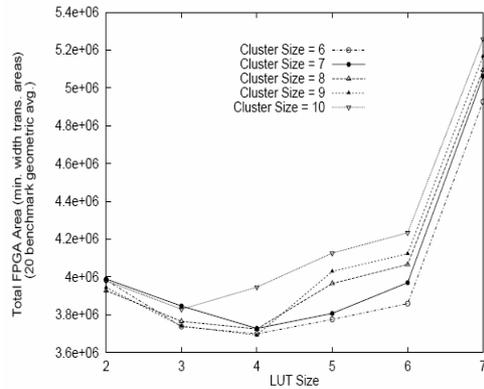


## Topologie de type matrice hiérarchique

La plus répandue



## Taille des clusters



Un grande taille de cluster facilite les contraintes de timings :  
=> taille d'au moins 8

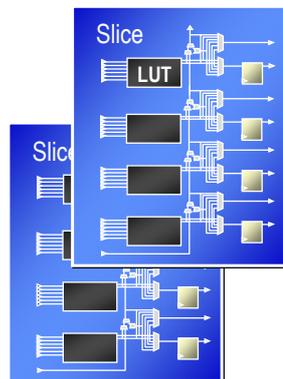
Taille du FPGA pour différentes tailles de LUT et cluster



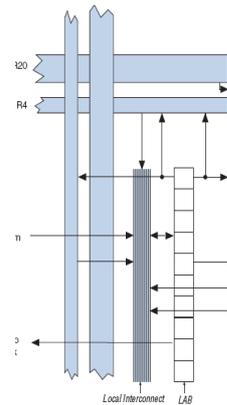
Elias Ahmed, Jonathan Rose: The effect of LUT and cluster size on deep-submicron FPGA performance and density. IEEE Trans. VLSI Syst. 12(3): 288-298 (2004)

FPGAs / Jean-Luc DANGER Page 67

## Exemple de clusters



**Virtex5 : CLB = 2 slices \* 4 LUT6**

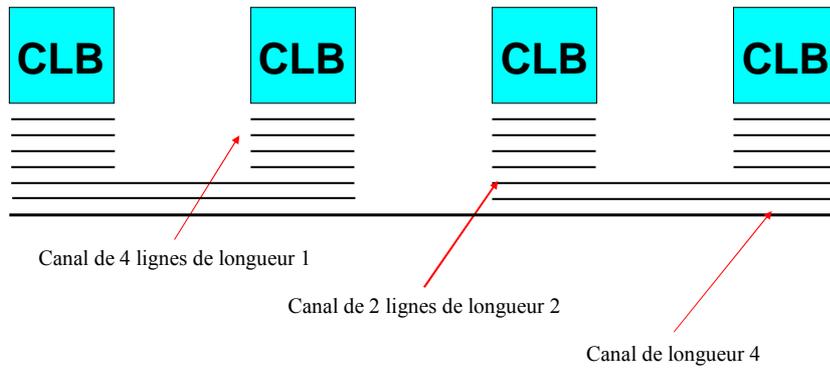


**StratixIII : LAB = 8 ALM**



FPGAs / Jean-Luc DANGER Page 68

## Matrice : interconnexions segmentées

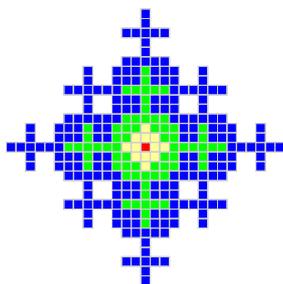


Un canal faisant toute la largeur sert pour les signaux globaux (horloge, reset,...)

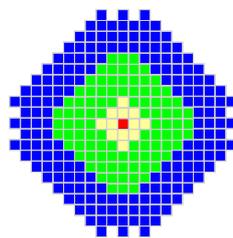
## Effet de la segmentation : gain en vitesse

1 hop = passage par un commutateur

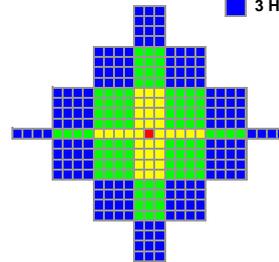
Intra-LAB  
1 Hop  
2 Hop  
3 Hop



XILINX  
Virtex 4

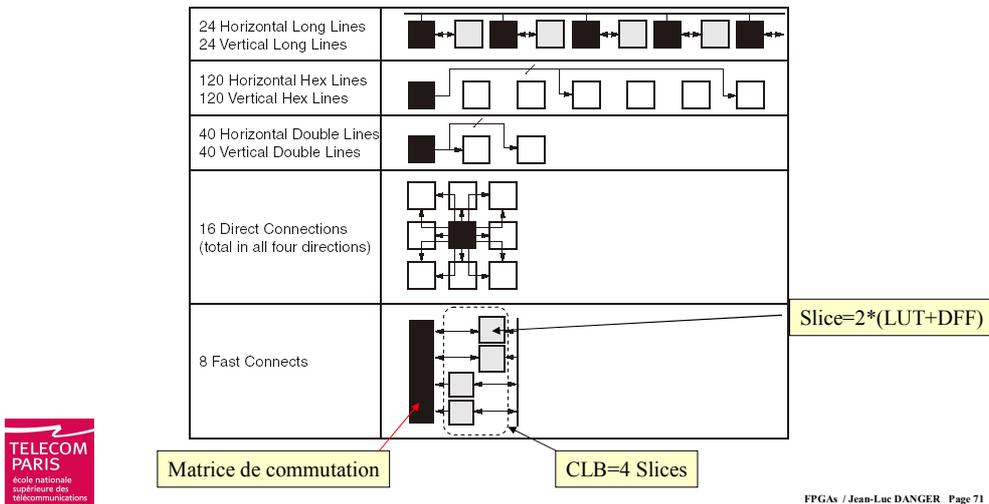


XILINX  
Virtex 5  
Segmentation diagonale

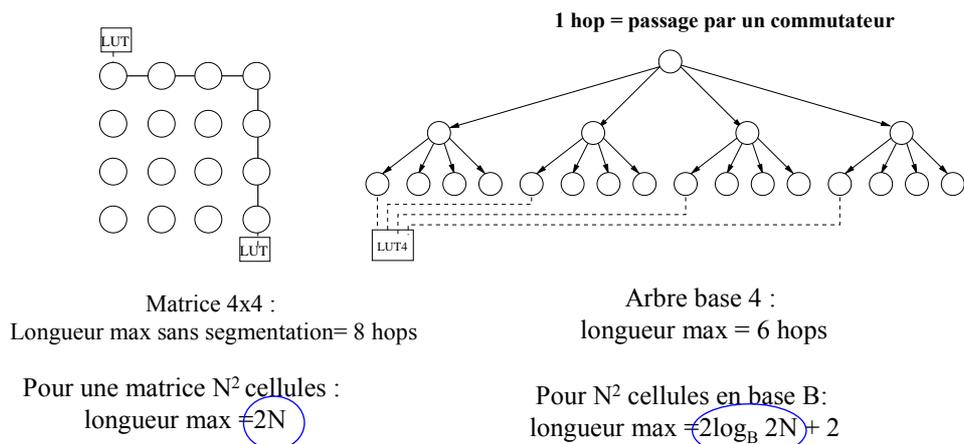


ALTERA  
StratixIII  
Segmentation très forte

## Matrice hiérarchique : XILINX VirtexII



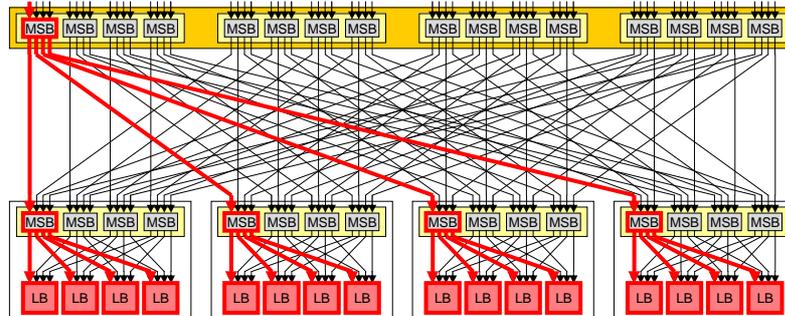
## Topologie en arbre



**Gros avantage pour la topologie en arbre.**  
**La matrice est toujours segmentée pour compenser ce handicap**

## Topologie en arbre : difficulté de routage

Réseau descendant sur 2 niveaux en base 4 :



Il faut aussi rajouter le niveau montant (mais base fois moins compliqué)

## PLAN

- ☞ **Place des FPGAs dans les Filières technologiques**
- ☞ **Architectures de la logique**
- ☞ **Architectures des blocs embarqués**
- ☞ **Architectures d'interconnexion**
- ➔ ☞ **Architectures de configuration**
- ☞ **Consommation et robustesse**
- ☞ **Méthodes de développement**
- ☞ **Analyse des architectures des constructeurs**

# Programmation

## ☞ Technologie interne

☐ *Volatile : Mémoire SRAM. Un "bitstream" est stocké à l'extérieur*

- On parle de configuration car la reprogrammation est possible

☐ *Nonvolatile :*

- Mémoire FLASH
- Antifusible

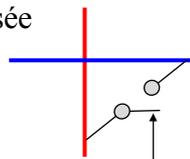
## ☞ Méthode de configuration pour SRAM

- ☐ *Interface spécifique*
- ☐ *Configuration totale/partielle*
- ☐ *sécurisation du transfert*

# Technologies de programmation

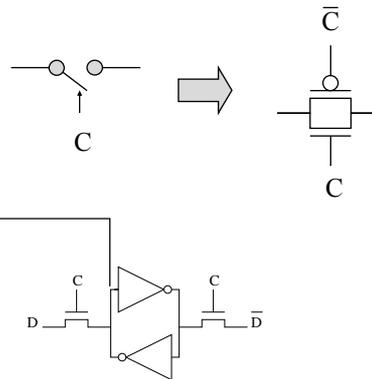
## ☞ RAM statique

La plus utilisée



Point mémoire SRAM

interconnexions par porte de transmission

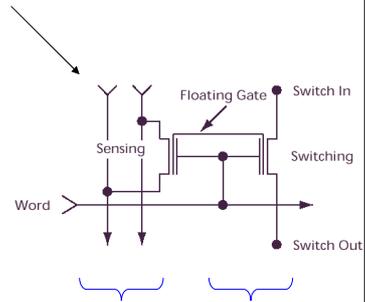
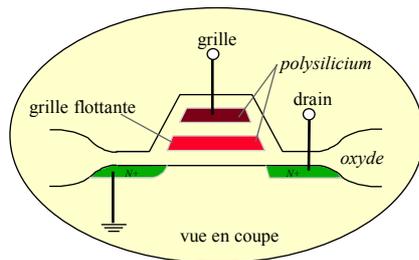


## Technologies de programmation

### EEPROM/FLASH

Utilisée pour les EPLD et certains FPGA (ACTEL ProASIC)

Transistor à grille flottante

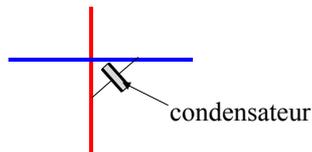


Programmation/vérification interconnexion

## Technologies de programmation

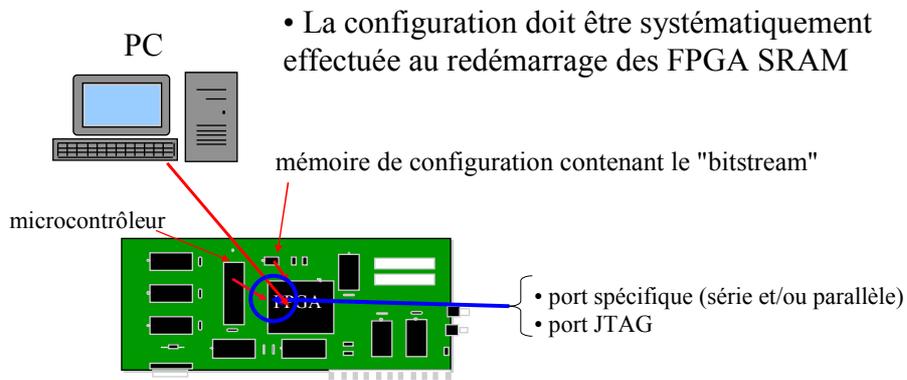
### Antifusible

Utilisés pour les FPGAs ACTEL AXcelerator



- La programmation consiste à faire fondre l'isolant du condensateur. L'opération est irréversible (programmation OTP)
- Excellente résistances aux rayons cosmiques (marché spatial)

## Interface de Configuration



Le FPGA peut être Maître (il pilote la configuration) ou Esclave

## Exemple de configuration du Virtex5

	Virtex-5 Modes	Bus width in bits			
		1	8	16	32
Maître	Master Serial	✓			
	Master SPI Flash	✓			
	Master Flash Up		✓	✓	
	Master Flash Down		✓	✓	
	Master SelectMAP		✓	✓	✓
Esclave	JTAG	✓			
	Slave SelectMAP		✓	✓	✓
	Slave Serial	✓			

série      parallèle

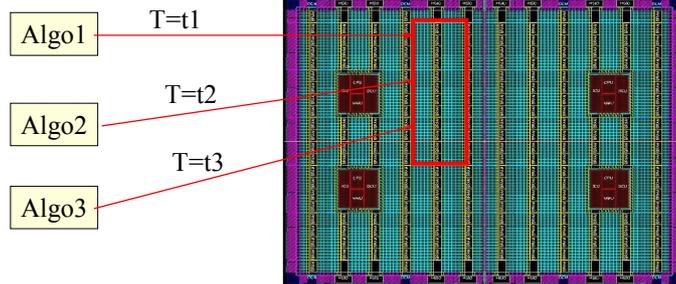
Si échec de la configuration : essai avec un nouveau bitstream => "Fallback"

## Configuration partielle

☞ Permet de minimiser le matériel

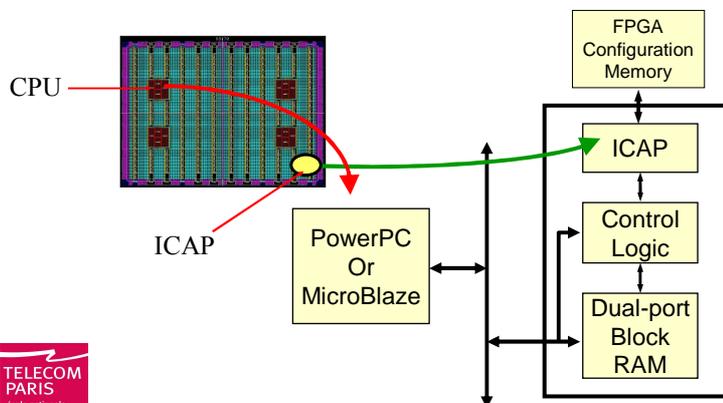
☞ Possibilité Xilinx

Fichiers de configuration



## Auto reconfiguration

☞ Virtex5 : Utilisation du port de configuration interne : ICAP et d'un microprocesseur local



# PLAN

---

- 📄 Place des FPGAs dans les Filières technologiques
- 📄 Architectures de la logique
- 📄 Architectures des blocs embarqués
- 📄 Architectures d'interconnexion
- 📄 Architectures de configuration
- ➔ 📄 **Consommation et robustesse**
- 📄 Méthodes de développement
- 📄 Analyse des architectures des constructeurs



# Consommation dans les FPGAs

---

## 📄 Consommation dynamique

- ❑ *En CMOS, l'essentiel de la consommation provient des courants de charge et décharge des capacités réparties*

$$P = \frac{1}{2} CV_{dd}^2 f$$

vdd ↘ f → c ↗ ⇒ P →

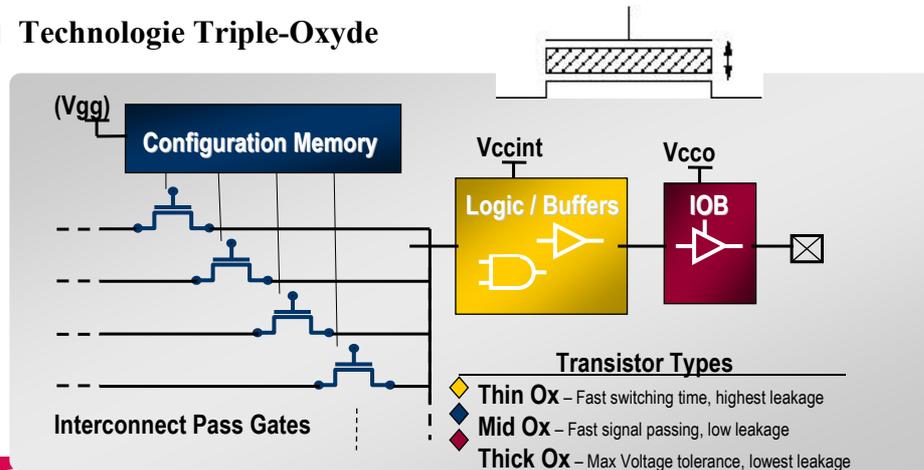
## 📄 Consommation statique

- ❑ *Due aux courants de fuite. Quasi nulle dans les technologies >130nm mais devient non négligeable si <130nm*



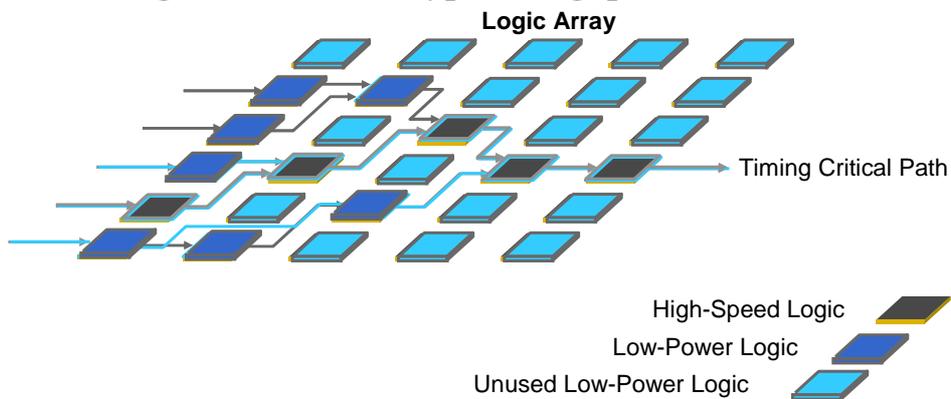
## Stratégie pour réduire la puissance statique

### Technologie Triple-Oxyde



## Stratégie pour réduire la puissance dynamique

### Programmation du type de logique : StratixIII



# Robustesse dans les FPGAs

## 2 problèmes

### Les rayonnements cosmiques

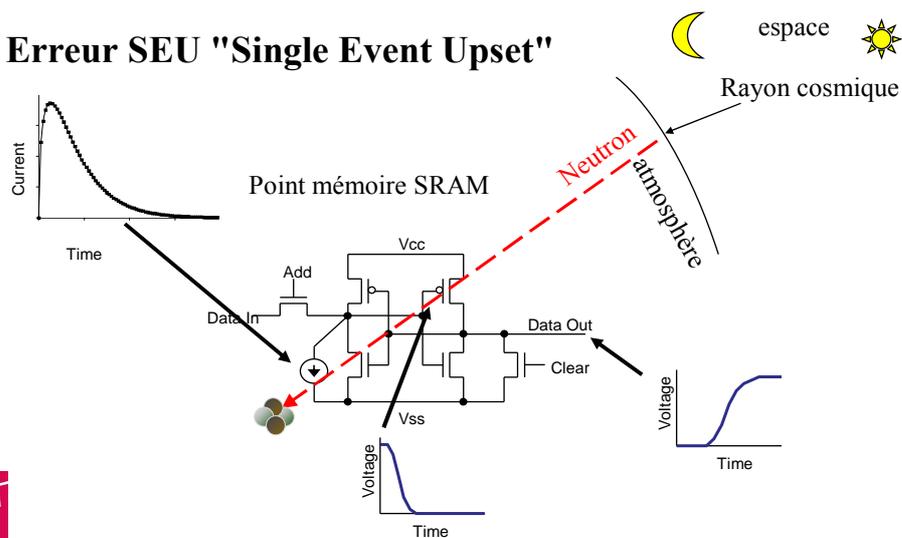
- Affecte les mémoires SRAM
  - Mémoires embarquées
  - Mémoire de configuration (à part Flash et anti-fusible)

### Le bruit interne détériorant l'intégrité du signal

- Interférences
- Bruit de masse (Simultaneous Switching Noise)
- Mauvaises adaptations d'impédance

# Rayonnement cosmique

## Erreur SEU "Single Event Upset"



## Parades au rayonnement cosmique

---

### ☞ Mémoire de configuration

- ❑ *Signature CRC dans le bitstream*
- ❑ *Mesure régulière du CRC (Xilinx) ou utilisation d'un processeur pour mesurer la "criticalité" (Altera)*
- ❑ *Utilisation de technologies robustes (Actel : FLASH, Antifusibles)*

### ☞ Mémoire utilisateur

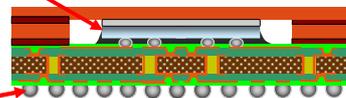
- ❑ *Code correcteur d'erreur ECC associé aux mémoires embarquées*

## Amélioration de l'intégrité du signal

---

### ☞ Action au niveau des E/S

- ❑ *Ajustement du slew rate*
- ❑ *Ajustement de la puissance des sorties*
- ❑ *Adaptation d'impédance*
- ❑ *Ajustement du retard*



### ☞ Action au niveau du boîtier

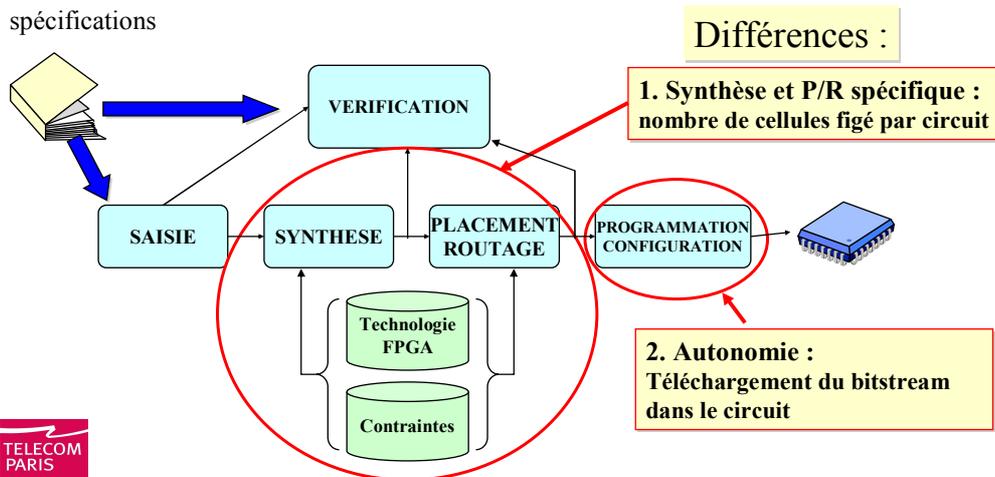
- ❑ *Respecter le ratio Vdd/Gnd/ E/S (8)*
- ❑ *Mettre des capacités de découplage ou utiliser des boîtiers prééquipés*

# PLAN

- 📄 Place des FPGAs dans les Filières technologiques
- 📄 Architectures de la logique
- 📄 Architectures des blocs embarqués
- 📄 Architectures d'interconnexion
- 📄 Architectures de configuration
- 📄 Consommation et robustesse
- ➔ 📄 Méthodes de développement
- 📄 Analyse des architectures des constructeurs

# Flot de conception FPGA

Très similaire à celle des circuits ASICs :



## Saisie

---

### Utilisation d'un langage HDL

- ❑ *Verilog, VHDL, SystemVerilog*

### Utilisation de la logique synchrone

- ❑ *Chaque cellule dispose "gratuitement" d'une DFF*
- ❑ *Une horloge unique ou plusieurs horloges corrélées sont gérées par les PLL et les réseaux d'interconnexions dédiés*
- ❑ *Facilité à sérialiser*
- ❑ *Facilité à paralléliser*
  - Duplication des blocs
  - Mise en pipeline
- ❑ *Facilité à analyser les timings et à fiabiliser*

## Intérêts des langages HDL

---

### Modélisation formelle

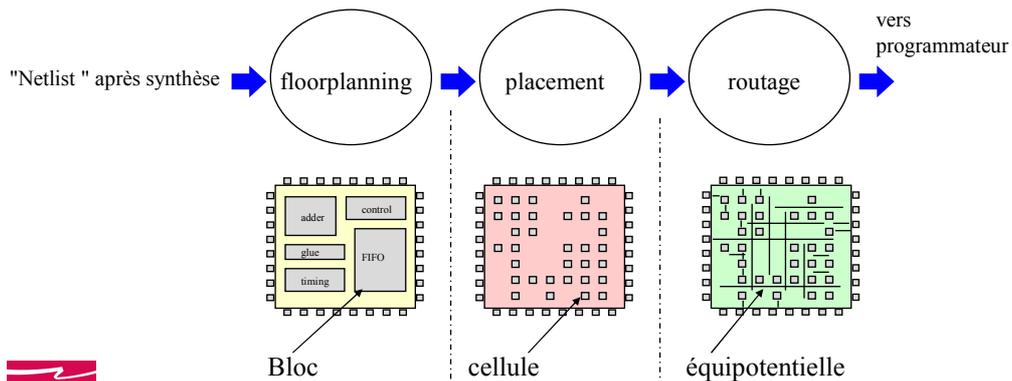
- ❑ *Simulation*
  - Ecriture de testbench avec circuit à tester
  - Vérification des résultats
- ❑ *Conception*
  - Synthèse logique

### Pour les gros circuits plusieurs modèles sont nécessaires :

- ❑ *Un modèle abstrait non synthétisable (C, systemC, HDL)*
- ❑ *Un modèle synthétisable en HDL au niveau RTL*

## Floorplanning-Placement-Routage

Chaque vendeur de CLP fournit son outil de P/R dédié



## Quartus II (ALTERA) et ISE (XILINX)

### 📁 Outil clés en main pour la compilation :

- ❑ *Outils de base : Saisie, synthèse, floorplan P/R, programmation*
- ❑ *Simulation post compilation*
- ❑ *compilation incrémentale*
- ❑ *Analyse de timing et de consommation*
- ❑ *Analyse logique*
- ❑ *SoC design*
- ❑ *Bibliothèque d'IPs*
- ❑ *Ligne de commande pour scripts*
- ❑ *Liens avec outils EDA*
- ❑ *Liens avec MATLAB*

### 📁 Nécessité d'un simulateur VHDL ou Verilog associé

## Points importants du Flot de Conception

---

- 📄 **Flooplanning**
- 📄 **Traitement macro-blocs (MC) et mémoires**
  - *Inférence/instantiation de MC*
- 📄 **Retiming**
- 📄 **Précision de l'analyse de timing**
- 📄 **Estimation de la consommation**
- 📄 **Débogage par analyse logique**
- 📄 **Compilation incrémentale**
- 📄 **Langage de script**

