

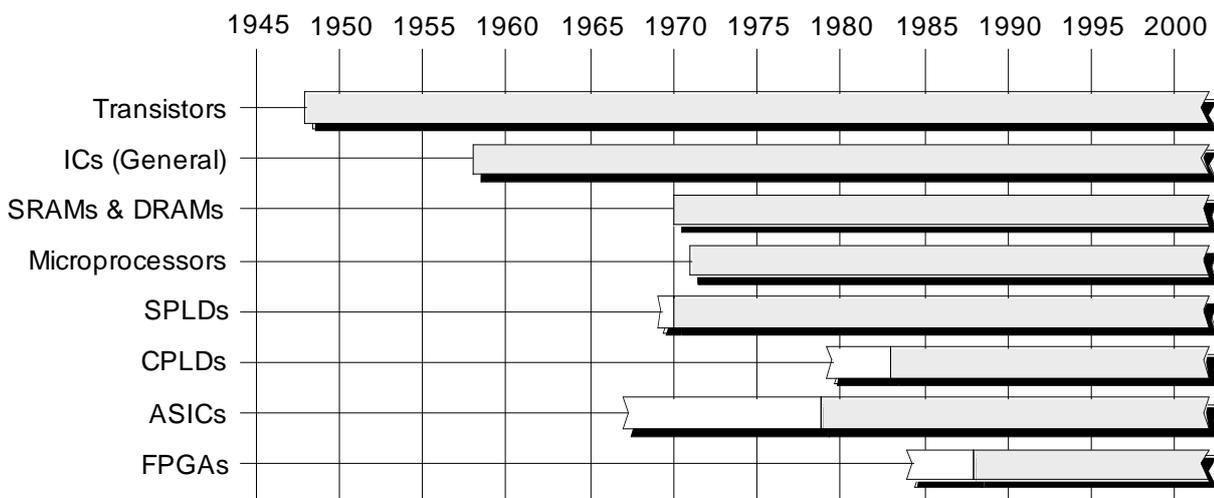
# Circuits reconfigurables: Les FPGAs

Eduardo Sanchez  
EPFL

## Types de circuits intégrés

- ASIC: *Application-Specific Integrated Circuits*
- ASSP: *Application-Specific Standard Parts*
- PLD: Programmable Logic Devices
  - SPLD: Simple PLD
  - CPLD: Complex PLD
- FPGA: Field Programmable Gate Array

- Les ASIC et les ASSP sont basés sur les mêmes processus de conception et technologies de fabrication. Les deux sont également conçus pour une application spécifique. Mais les ASIC sont destinés à une compagnie spécifique, tandis que les ASSP sont vendus à de multiples clients
- Comparés aux FPGA, les PLD contiennent un nombre relativement petit de portes logiques et les fonctions implémentables sont plus petites et plus simples
- En 2003, le nombre de nouvelles conceptions a été:
  - ASIC: 1'500-4'000
  - ASSP: 5'000
  - FPGA: 450'000



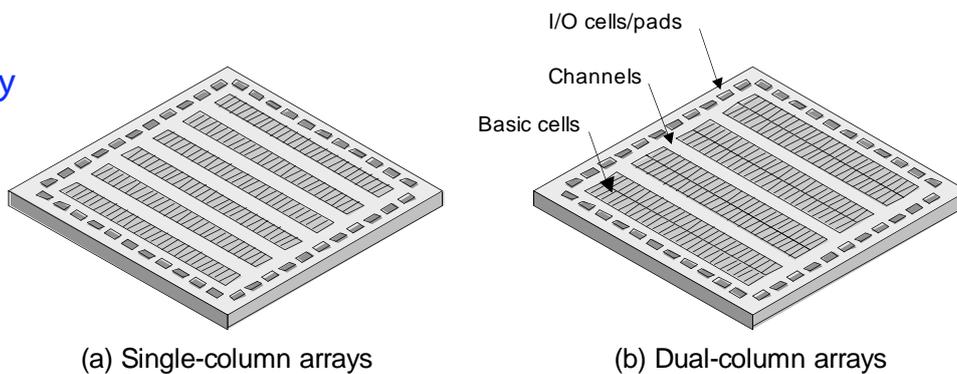
- Il y a quatre grandes familles d'ASIC:

- *full custom*: les concepteurs ont un contrôle total sur tous les masques utilisés pour fabriquer le chip
- *gate array*: une matrice de cellules de base. Chaque cellule est une collection de transistors et de résistances non connectés. Quand il n'y a pas de canaux d'interconnexion entre les colonnes de cellules, le circuit est nommé *sea-of-gates* ou *sea-of-cells*.

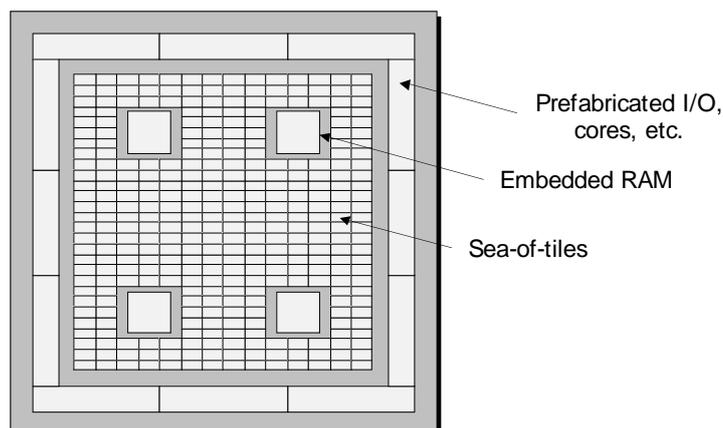
Le vendeur des ASIC définit un ensemble de fonctions logiques à mettre à disposition des ingénieurs de conception. Cet ensemble est appelé la bibliothèque de cellules (*cell library*)

- *standard cell*: le vendeur fournit une bibliothèque de cellules, mais aucun composant n'est pre-fabriqués dans le chip. Il est possible d'acheter aussi des blocs plus complexes: des IP (*Intellectual Property*)
- *structured*: le chip contient une matrice d'éléments pre-fabriqués, appelés tuiles (*tiles*), plus d'autres éléments spécialisés. Un ASIC de ce type utilise trois fois la surface et 2-3 fois la puissance d'un standard cell équivalent

gate array



structured

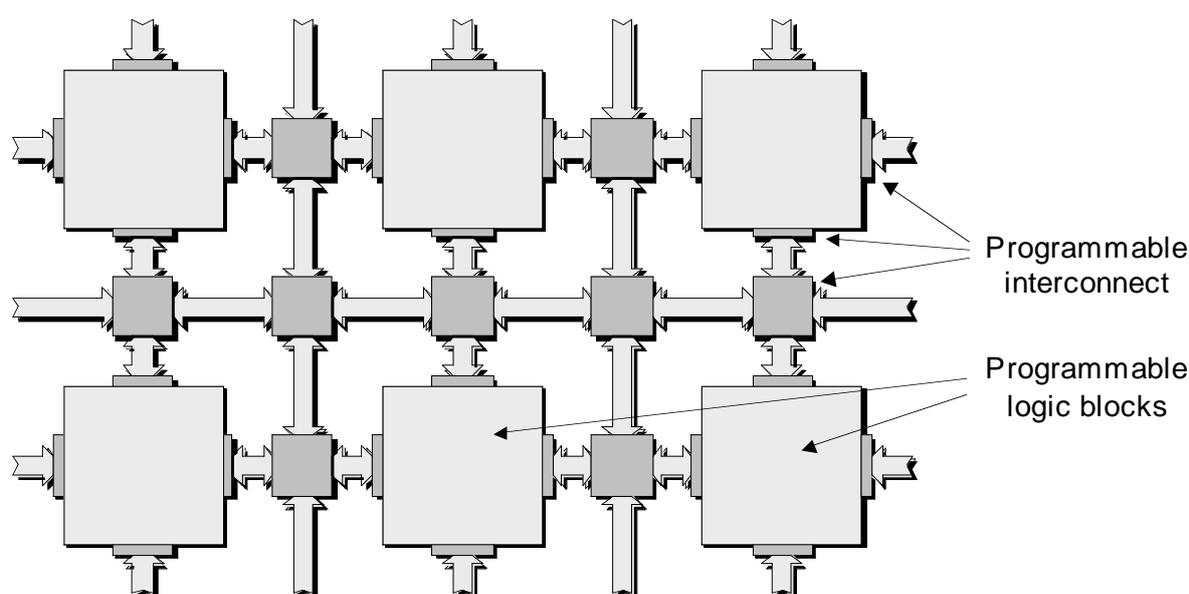


# Les FPGAs

- Les FPGAs (*Field Programmable Gate Arrays*) sont des circuits intégrés qui contiennent des blocs de logique configurables (programmables), ainsi que des interconnexions configurables entre ces blocs
- La configuration du circuit, c'est-à-dire la programmation de sa fonction, se fait sur place (*Field Programmable*), sans envoi chez un fabricant
- Les FPGA se placent à mi-chemin entre les PLD et les ASIC. Les avantages par rapport aux ASIC sont:
  - coût de développement plus bas: les dépenses en NRE (*nonrecurring engineering*) sont moins importantes
  - modifications plus simples à réaliser
  - *time-to-market* plus court

Eduardo Sanchez

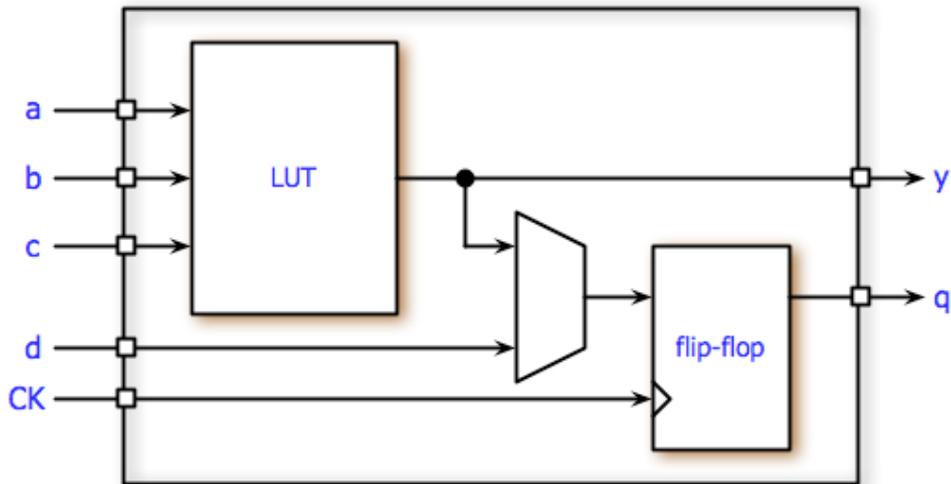
7



Eduardo Sanchez

8

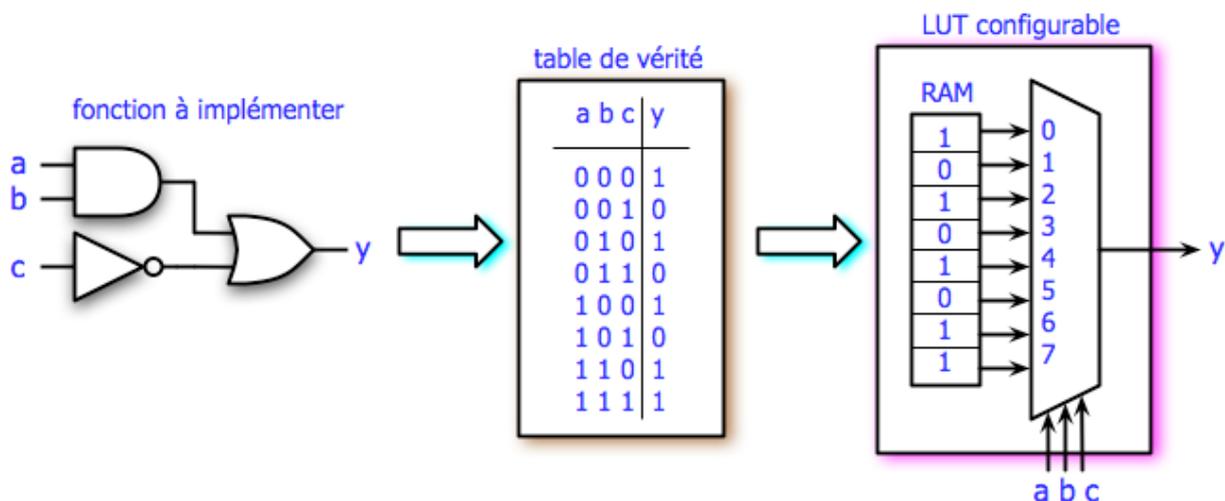
- La structure du bloc logique change selon le fabricant, la famille, etc
- Aujourd'hui, la structure la plus utilisée est basée sur une *look-up table* (RAM) pour implémenter une fonction combinatoire plus une bascule D



Eduardo Sanchez

9

- La fonction de la LUT est de stocker la table de vérité de la fonction combinatoire à implémenter dans la cellule



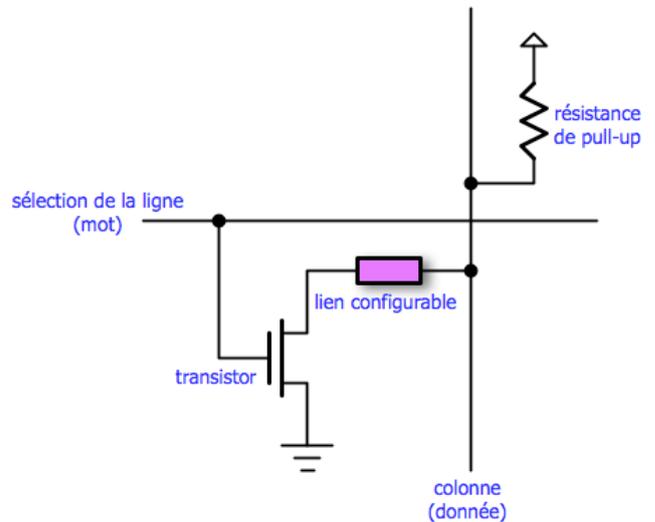
Eduardo Sanchez

10

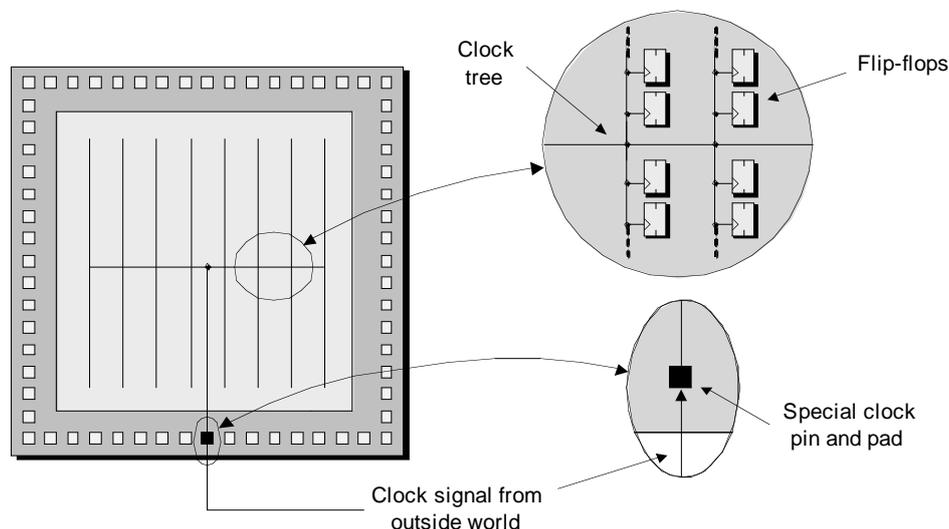
- Les technologies de configuration sont:

- fusible
- anti-fusible
- EPROM
- E2PROM/Flash
- SRAM

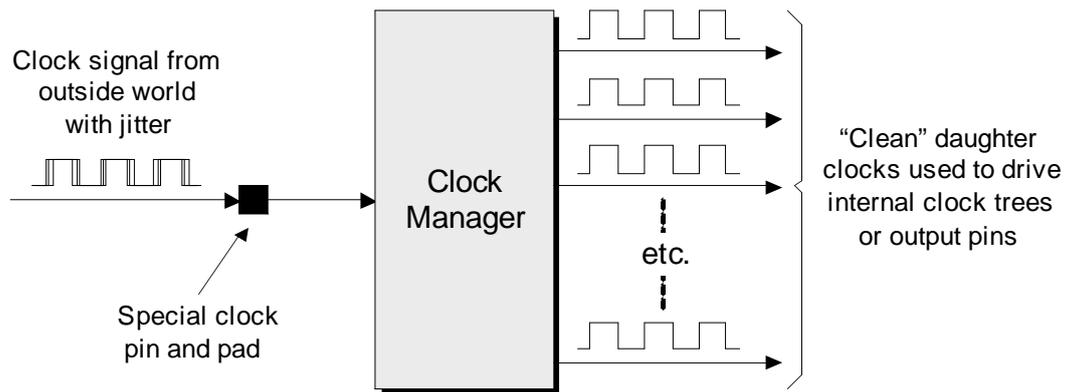
- Exemple: la figure montre la configuration d'un bit d'une ROM. Si le lien configurable existe, la colonne (donnée) est connectée à 0 via le transistor; dans le cas contraire, la colonne est connectée à 1 via la résistance



- Le signal d'horloge d'un FPGA vient généralement de l'extérieur. Il y a des entrées spécialisées pour recevoir les signaux d'horloge et les distribuer ensuite à l'intérieur du circuit
- La distribution du signal d'horloge se fait par le biais d'un arbre, minimisant ainsi les retards d'arrivée du signal aux flip-flops les plus éloignés (*skew*)



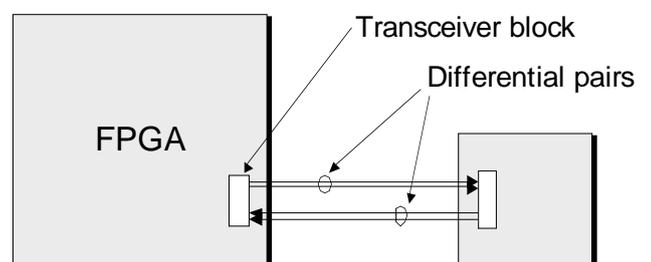
- Il est aussi courant d'avoir des blocs internes pour la gestion du signal d'horloge, des DCM (*Digital Clock Managers*)
- Les DCM génèrent d'autres fréquences d'horloge à partir du signal externe, en éliminant le *jitter* (retards entre les différents flancs des signaux)
- Les DCM sont basés sur des PLL (*phase-locked loops*) ou des DLL (*digital delay-locked loops*)



Eduardo Sanchez

13

- Les pins d'entrée/sortie des FPGA possèdent également des caractéristiques configurables:
  - type d'interface
  - impédance
  - voltage. Le voltage d'alimentation est en baisse constante:
    - 1998 (350 nm): 3.3 V
    - 1999 (250 nm): 2.5 V
    - 2000 (180 nm): 1.8 V
    - 2001 (150 nm): 1.5 V
    - 2003 (130 nm): 1.2 V
  - blocs pour la transmission de données en série à grande vitesse (*transceivers*). Ces blocs utilisent une paire de signaux différenciés (avec des valeurs complémentaires) pour transmettre des données (TX) et une autre paire pour recevoir (RX)

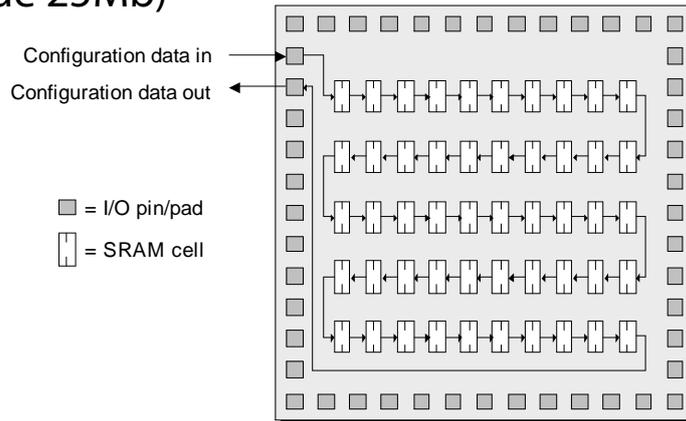


Eduardo Sanchez

14

# Configuration des FPGA

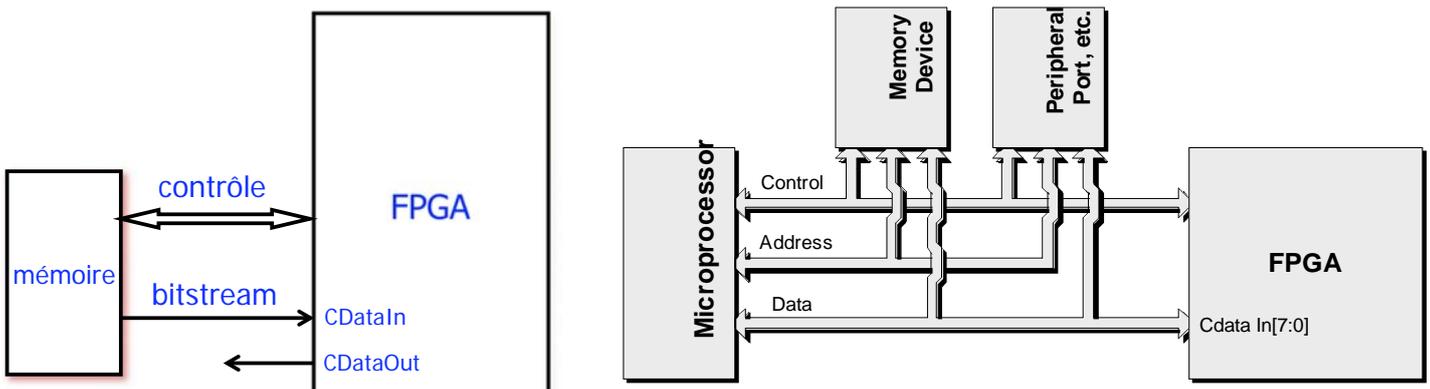
- La chaîne de bits de configuration (*bitstream*) est envoyée depuis l'extérieur et distribuée en série à l'intérieur du FPGA. On peut voir les SRAM de configuration comme un énorme registre à décalage (dans la réalité, c'est divisé en *frames* de 1024 bits et on utilise des latches!)
- Le temps de configuration peut être très long (on a aujourd'hui des bitstreams de 25Mb)



Eduardo Sanchez

15

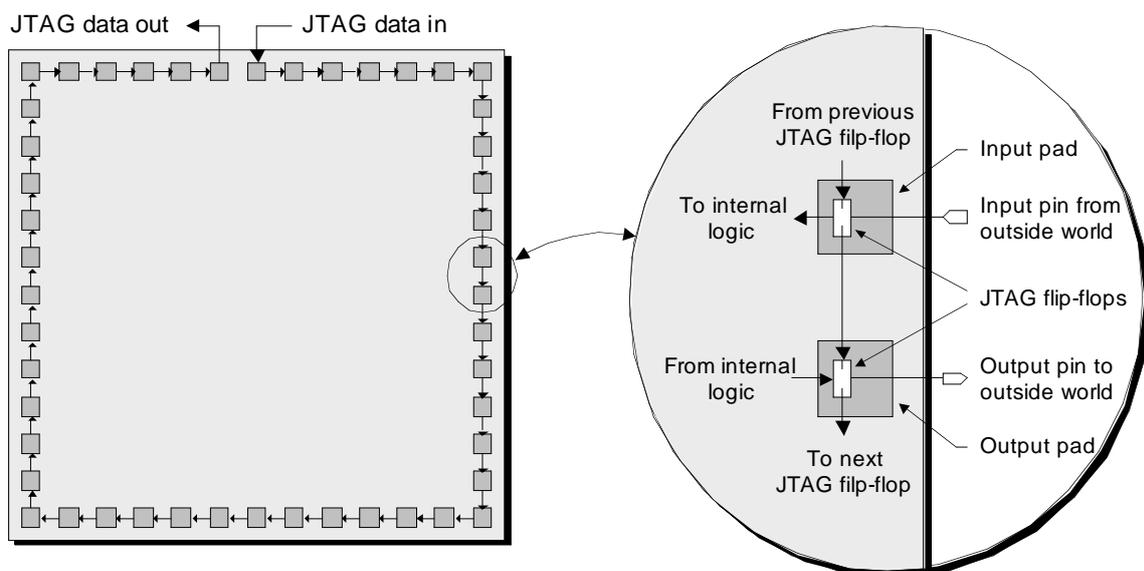
- Il existe en général 4 modes de configuration:
  - série, avec le FPGA en maître
  - série, avec le FPGA en esclave
  - parallèle, avec le FPGA en maître
  - parallèle, avec le FPGA en esclave



Eduardo Sanchez

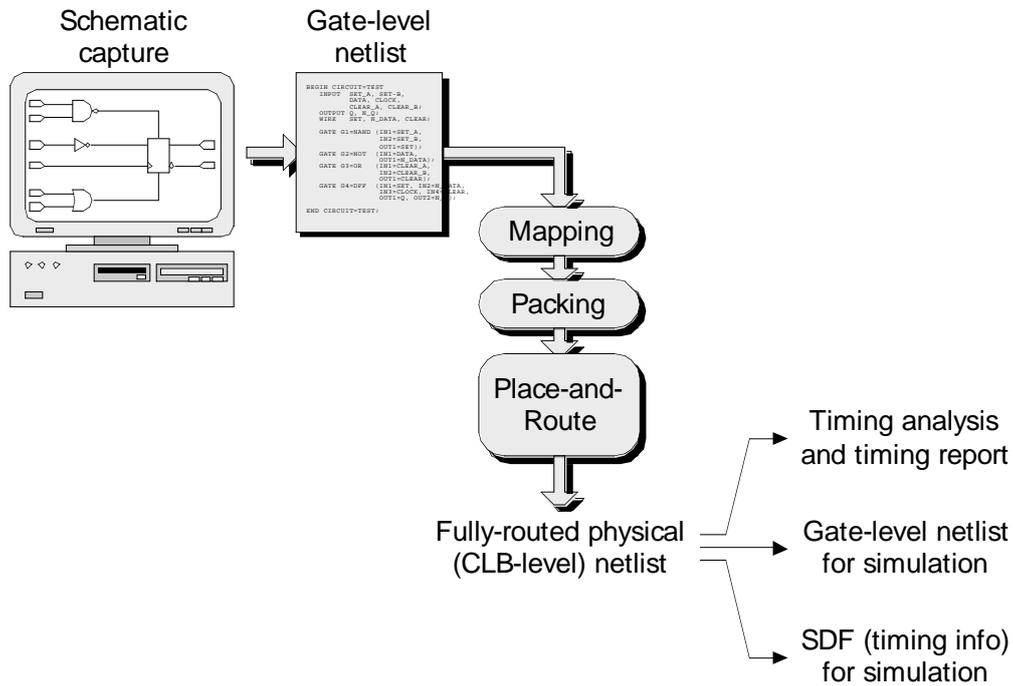
16

- Une autre façon de configurer le FPGA est l'utilisation du port JTAG (*Joint Test Action Group*)
- Normalement, ce port est utilisé pour tester le circuit intégré, grâce à la technique de *boundary scan*. Un pin est utilisé pour entrer les données JTAG et un autre pour les sortir. Tous les autres pins d'entrée/sortie disposent d'un flip-flop JTAG; tous ces flip-flops sont connectés en série
- En plus des commandes propres au test, le port JTAG des FPGA admet aussi des commandes de configuration, permettant le chaînage du registre de configuration au registre JTAG



# Processus de conception

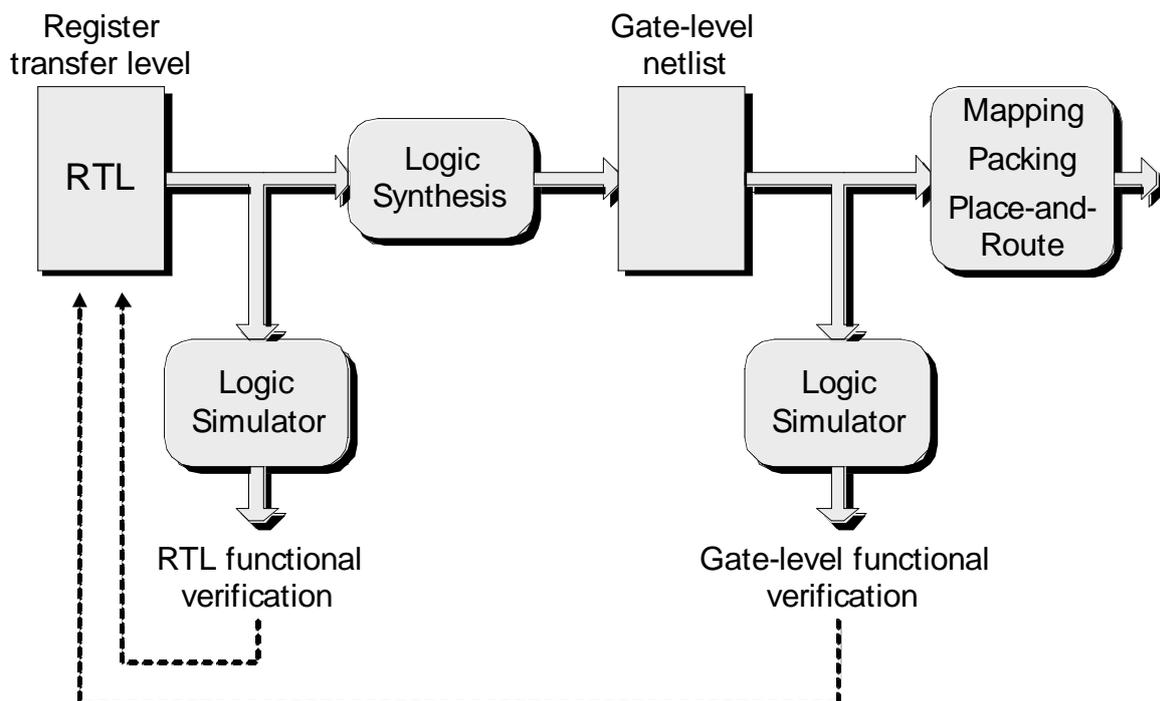
- A partir d'un schéma:



Eduardo Sanchez

19

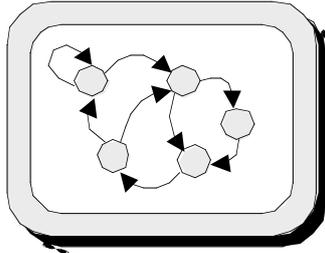
- A partir d'une description avec un langage de haut niveau (HDL = *Hardware Description Language*):



Eduardo Sanchez

20

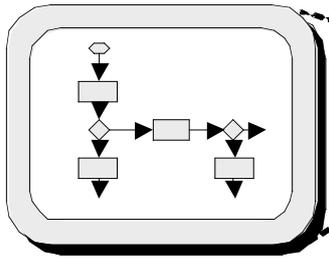
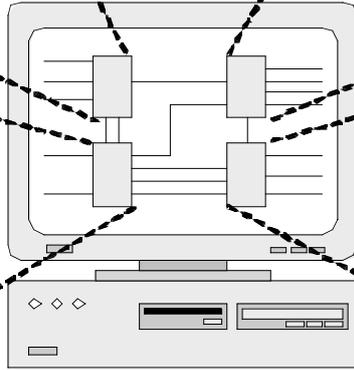
Graphical State Diagram



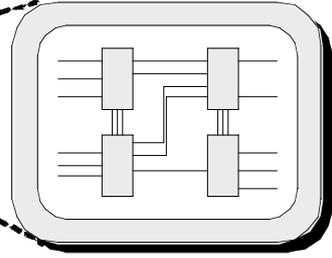
Textual HDL

```
When clock rises  
If (s = 0)  
then y = (a & b) | c;  
else y = c & !(d ^ e);
```

Top-level  
block-level  
schematic



Graphical Flowchart



Block-level schematic