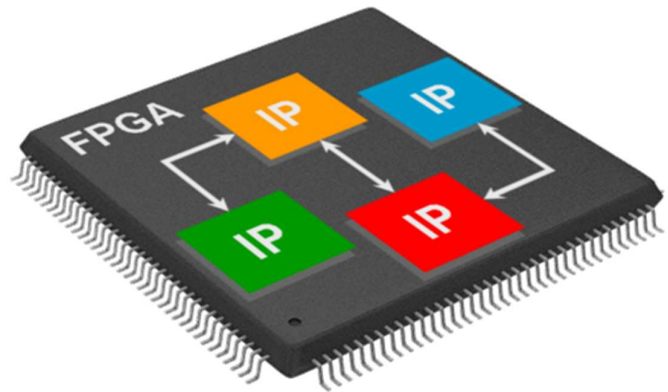
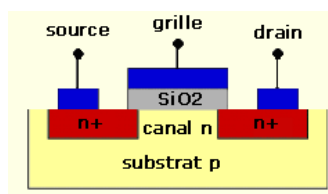


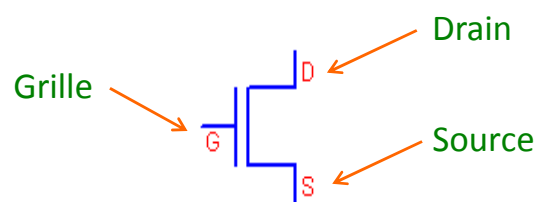
Le composant FPGA



1.1. Le transistor NMOS

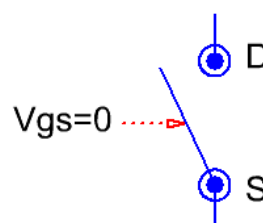
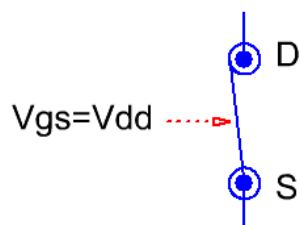


Vue en coupe

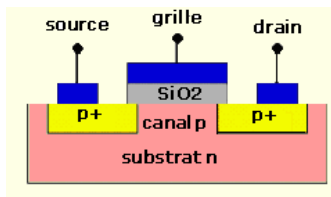


Symbole simplifié

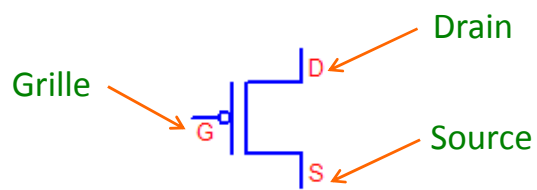
- ➔ En logique, le transistor est utilisé en interrupteur commandé.
- ➔ L'interrupteur se situe entre le drain et la source.
- ➔ La commande de l'interrupteur se fait par la tension V_{GS} (entre Grille et Substrat).



1.2. Le transistor PMOS

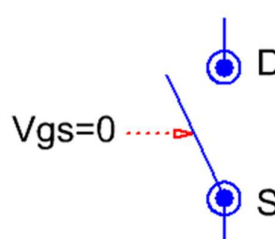
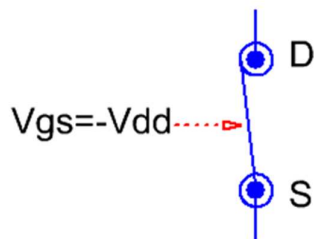


Vue en coupe



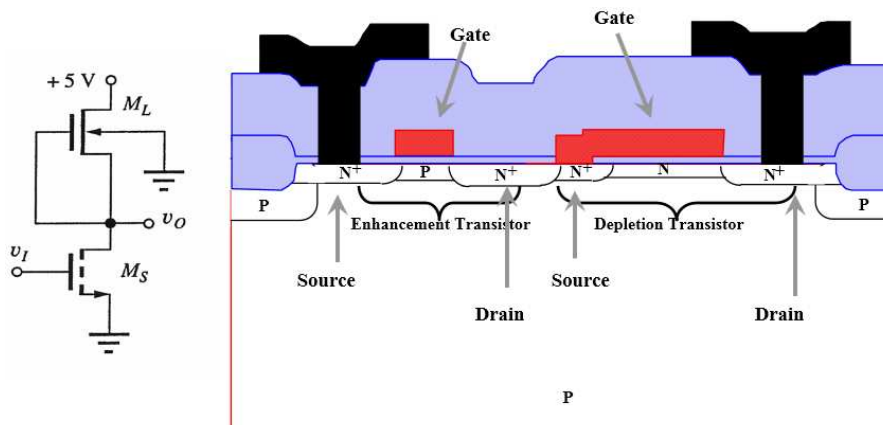
Symbole simplifié

- ➔ Comme son dual, le transistor PMOS est aussi utilisé en interrupteur commandé.
- ➔ Il y a une complémentarité entre les deux types, que ce soit au niveau de la structure ou au niveau du fonctionnement.
- ➔ La technologie CMOS (Complementary MOS) exploite cette complémentarité.



3

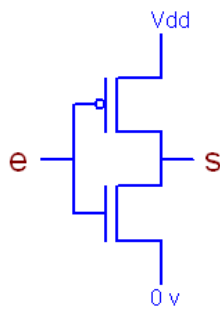
1.3. Technologie microélectronique



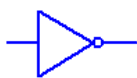
4

1.4. L'inverseur

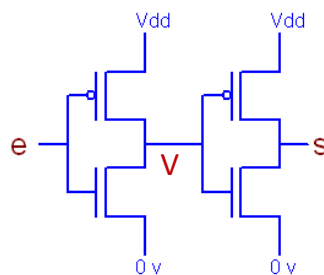
Illustration du temps de propagation quand e passe de 0 à 1



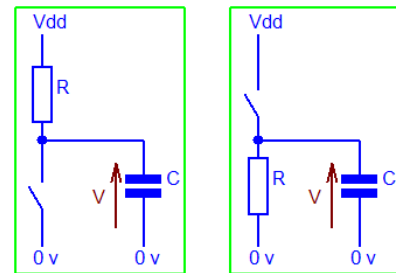
Structure



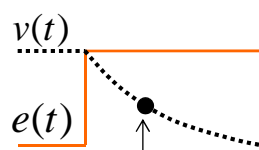
Symbole



Double inversion



Schémas équivalents



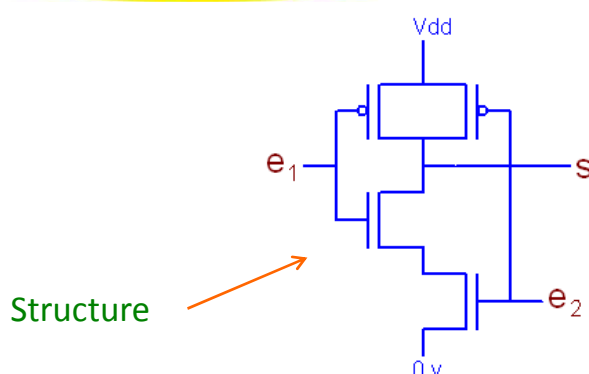
Changement d'état de $s(t)$ à $v(t)=Vdd/2$

$$v(t) = Vdd(e^{-\frac{t}{RC}})$$

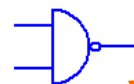
Il y a un temps de propagation dans tout circuit logique. Sa connaissance permet d'adapter la fréquence de travail.

5

1.5. La porte NAND



Structure

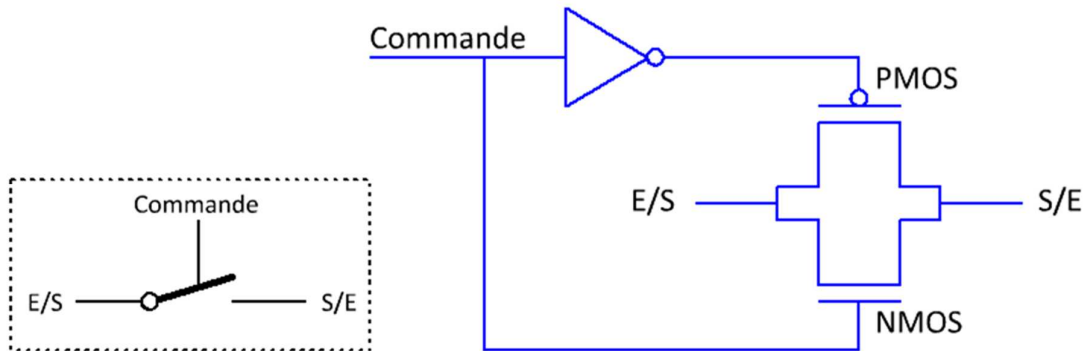


Symbole

- ➔ La porte NAND à 2 entrées, en technologie CMOS est constituée de 4 transistors.
- ➔ Toute fonction logique combinatoire peut être réalisée à partir de portes NAND. Ce qui en fait une porte universelle.
- ➔ La porte NAND est choisie comme unité de mesure pour évaluer la densité d'un circuit logique programmable.
- ➔ Exemple : un composant de densité 1000 portes peut embarquer une logique équivalente à 1000 portes NAND.

6

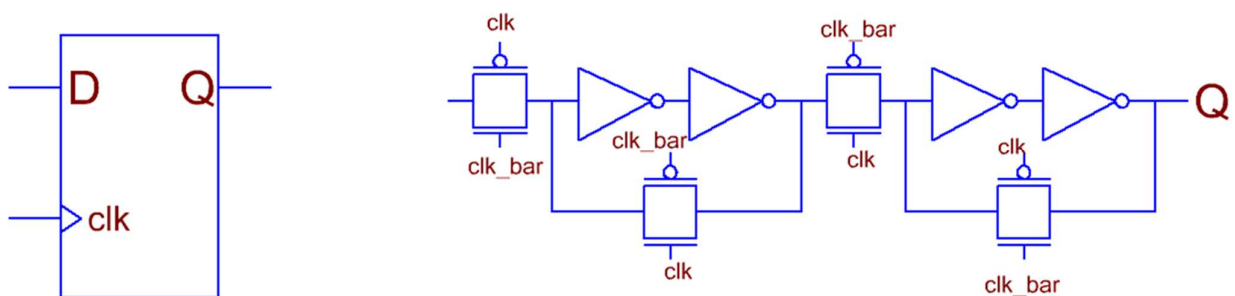
1.6. L'interrupteur analogique



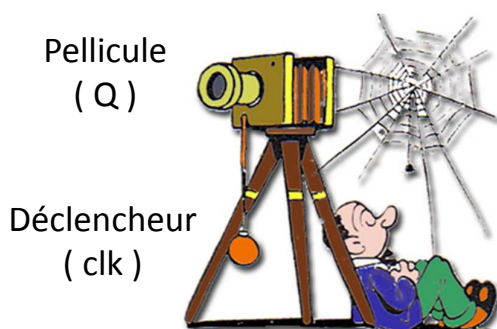
- ➔ Commande = 0L : les deux transistors présentent une impédance quasi infinie entre le drain et la source, le circuit se comporte comme un interrupteur ouvert
- ➔ Commande = 1L : les deux transistors présentent une faible résistance R_{ON} entre le drain et la source, le circuit se comporte comme un interrupteur fermé (parfaitement symétrique)

7

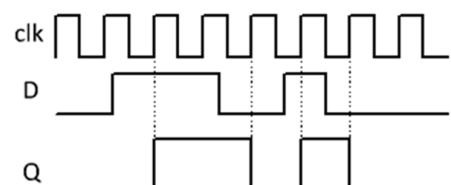
1.7. La bascule D



- ➔ La bascule D recopie l'entrée D sur la sortie Q sur front montant d'horloge clk.
- ➔ C'est un composant qui fige une situation comme le ferait un appareil photo.

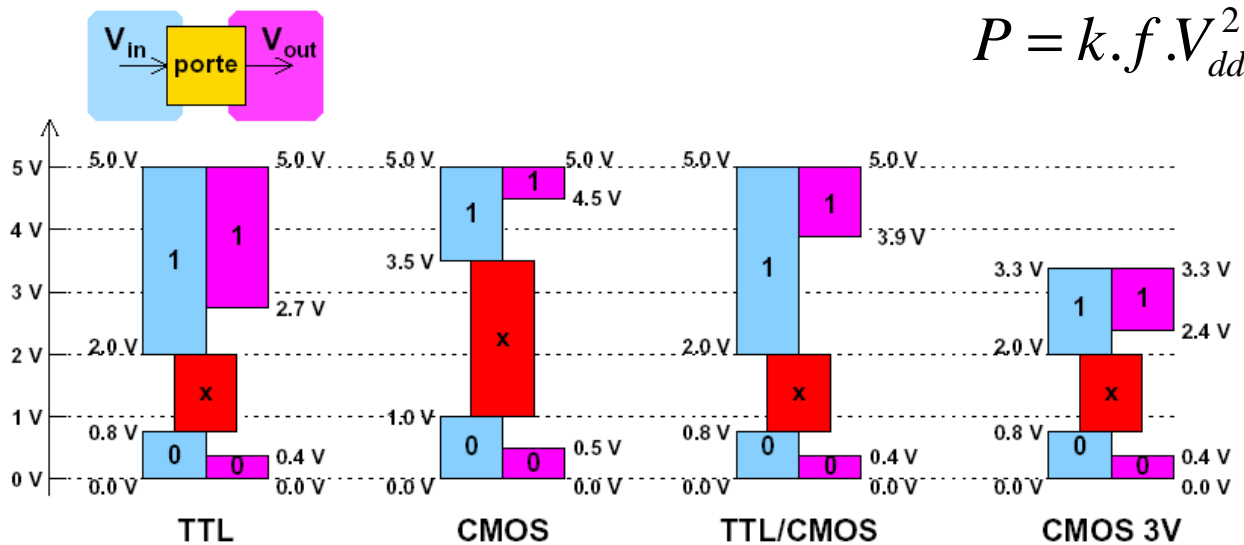


Sujet
(D)



8

2.1. Tension et niveaux logiques

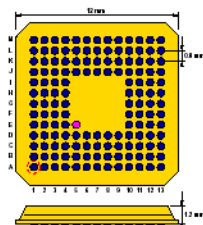


- ➔ Les fourchettes des niveaux logiques fixent les règles de dialogue entre les circuits tout en garantissant une immunité au bruit.
- ➔ Pour réduire la puissance consommée, on diminue la tension d'alimentation. Jusqu'à quelle valeur ?

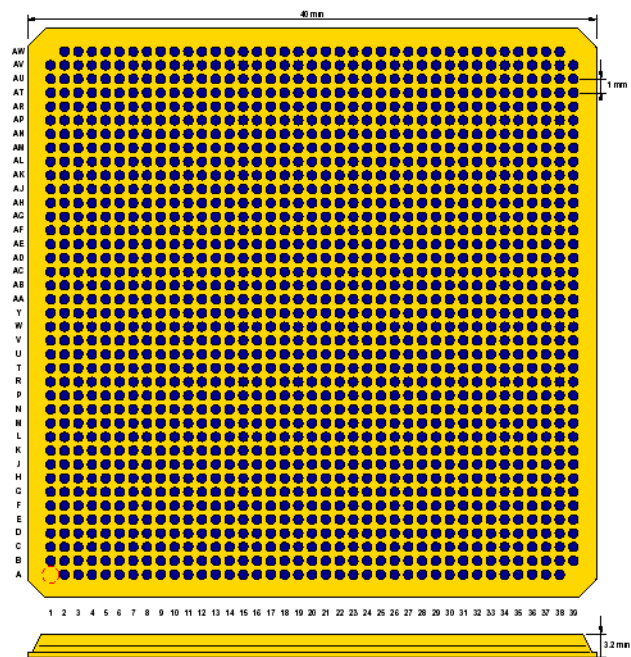
9

2.2. Les boîtiers BGA (Ball Grid Array)

CS144 Chip-Scale BGA



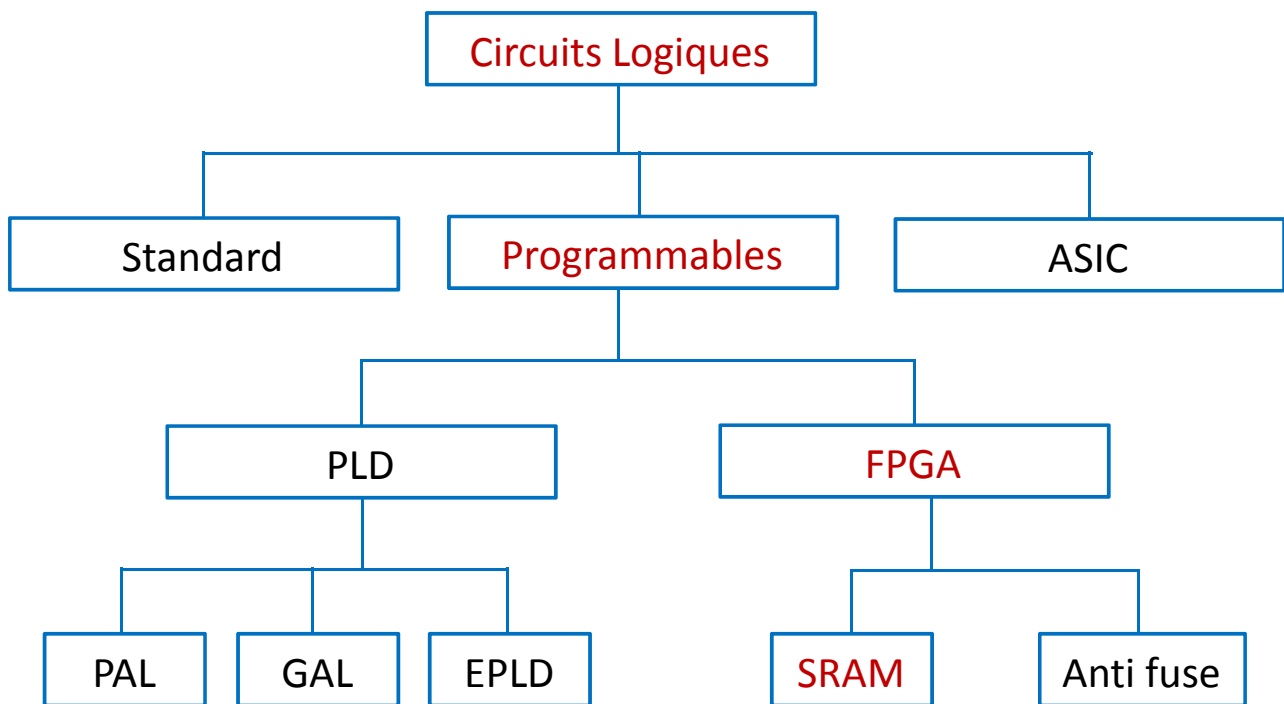
FF1517 Flip-Chip Fine-Pitch BGA



- ➔ Les boîtiers de type BGA ont révolutionné la micro électronique en faisant exploser le nombre de broches.
- ➔ Le montage en surface de ces boîtiers permet des économies non négligeables.
- ➔ La taille des composants se trouve réduite par la même occasion.

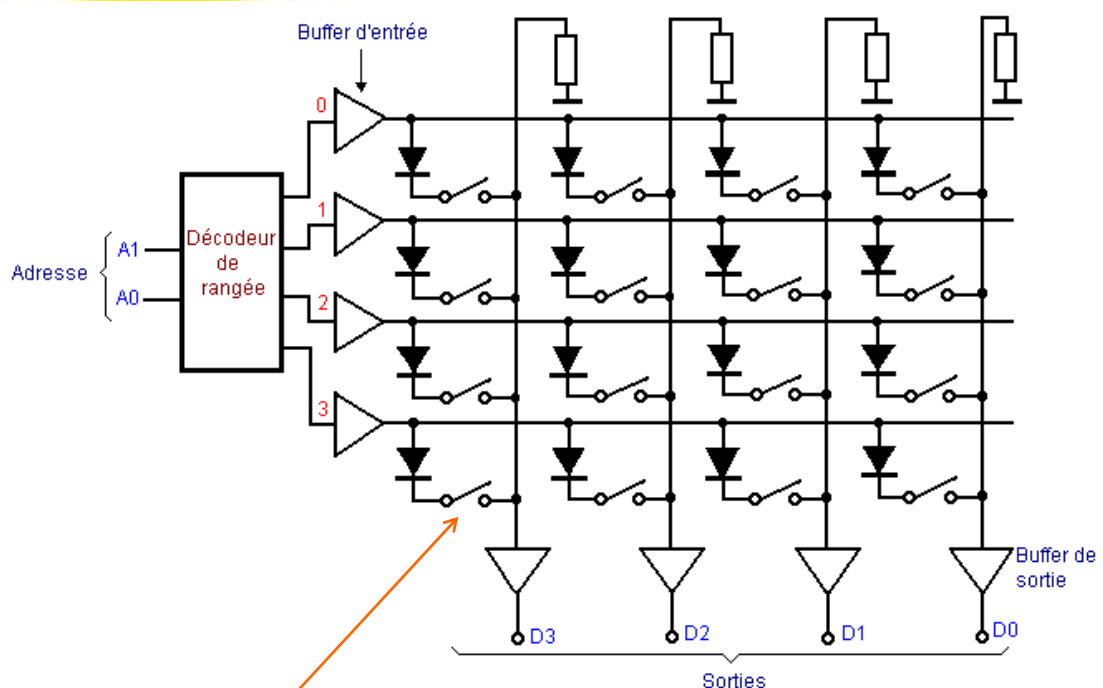
10

2.3. Les types de circuits logiques



11

2.4. Technique de programmation

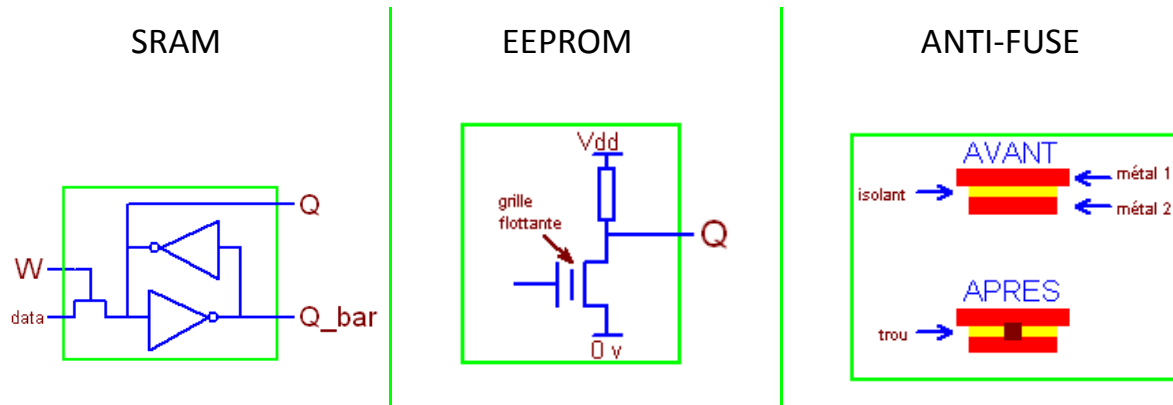
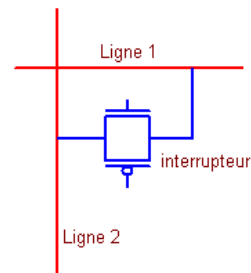


- ➔ La technique de programmation s'inspire de celle des mémoires : elle consiste à connecter des lignes horizontales avec des lignes verticales

12

2.5. Technologie de programmation

- ➔ La programmation d'un composant revient à fermer ou ouvrir l'interrupteur qui relie les 2 lignes.
- ➔ Différentes technologies sont mises en œuvre pour commander ou réaliser l'interrupteur.

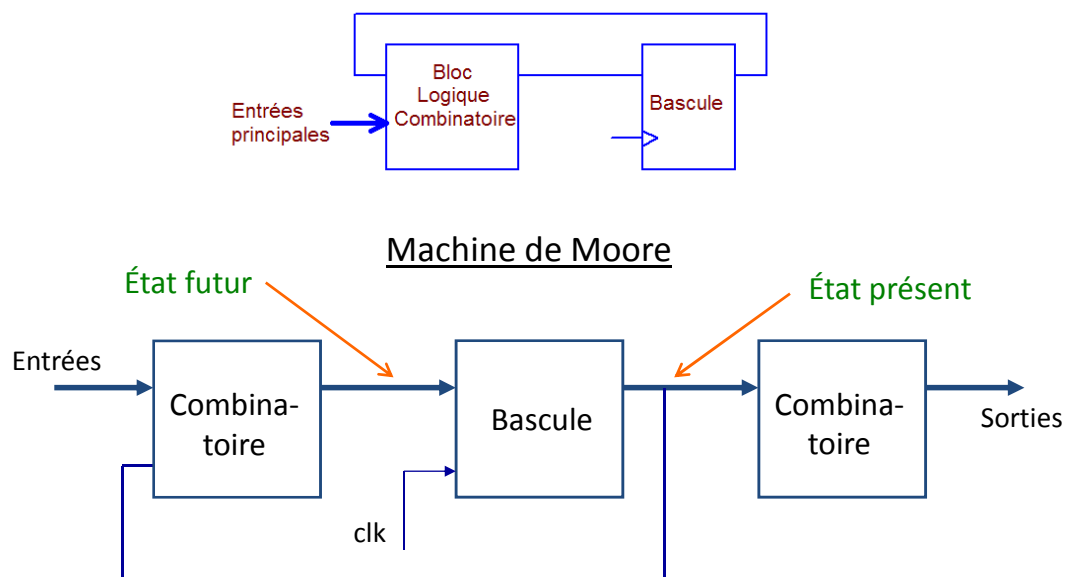


- ➔ Remarque : la technique la plus ancienne utilisait des fusibles !

13

3.1. Synthèse logique

- ➔ Les composants programmables exploitent tous la même technique d'implantation de fonctions combinatoires ou séquentielles : un bloc logique combinatoire + une bascule (mémoire). La bascule D suffit amplement.



Les sorties dépendent uniquement de l'état présent !

14

3.2. Synthèse dans les PAL

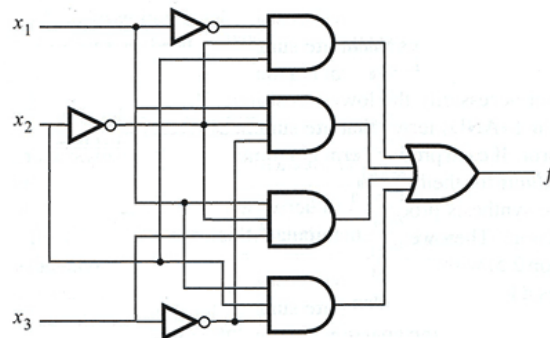
- ➔ Toute fonction combinatoire peut être décrite comme une somme de produits (min termes).

$$f = \sum \Pi$$

- ➔ Dans les composants de première génération, on traduisait l'équation précédente par des portes ET suivies de portes OU dans le composant (somme de produits).

| Row number | x_1 | x_2 | x_3 | $f(x_1, x_2, x_3)$ |
|------------|-------|-------|-------|--------------------|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 0 |

$$f(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3$$



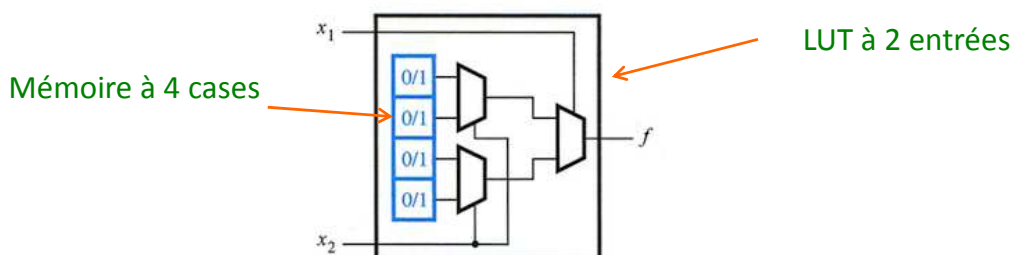
Somme de produits

- ➔ A l'usage, cette technique gaspille énormément de ressources dans le composant

15

3.3. Synthèse dans les FPGA

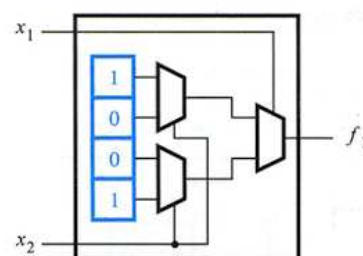
- ➔ Les composants FPGA utilisent la technique de la table de consultation (LUT).
- ➔ Cette technique utilise des petites mémoires pour réaliser des fonctions combinatoires.



| x_1 | x_2 | f_1 |
|-------|-------|-------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$f_1 = \bar{x}_1\bar{x}_2 + x_1x_2$$

Fonction combinatoire
à 2 entrées

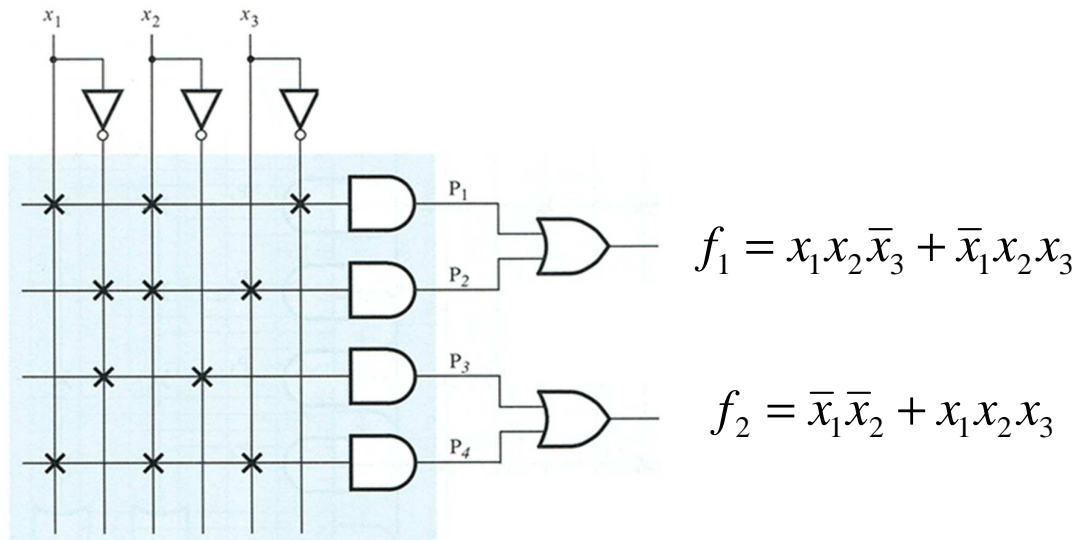


Réalisation dans une LUT
à 4 adresses

16

4.1. Architecture des PAL

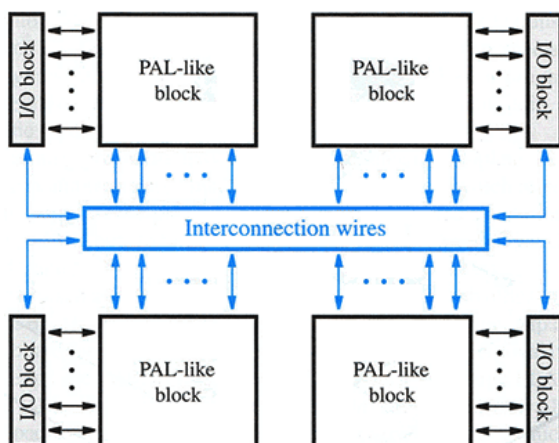
- Les circuits PAL ont une architecture qui convient très bien à l'implantation de fonctions combinatoires.
- les réseaux de ET logiques suivis de OU logiques qui constituent le PAL, correspondent parfaitement à la réalisation de fonctions combinatoires : $f = \sum \Pi$
- Les croix représentent les jonctions réalisées (programmées).



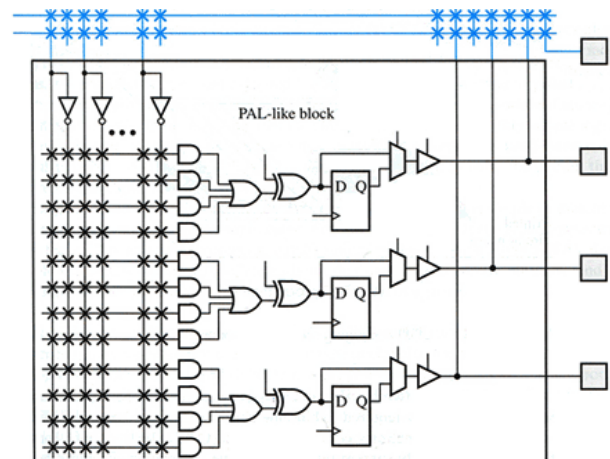
17

4.2. Architecture des CPLD

- Un CPLD (Complex Programmable Logic Device) embarque plusieurs circuits PAL et un réseau d'interconnexions complètement programmable.
- Un composant CPLD peut être considéré comme un gros circuit PAL.



Structure du CPLD

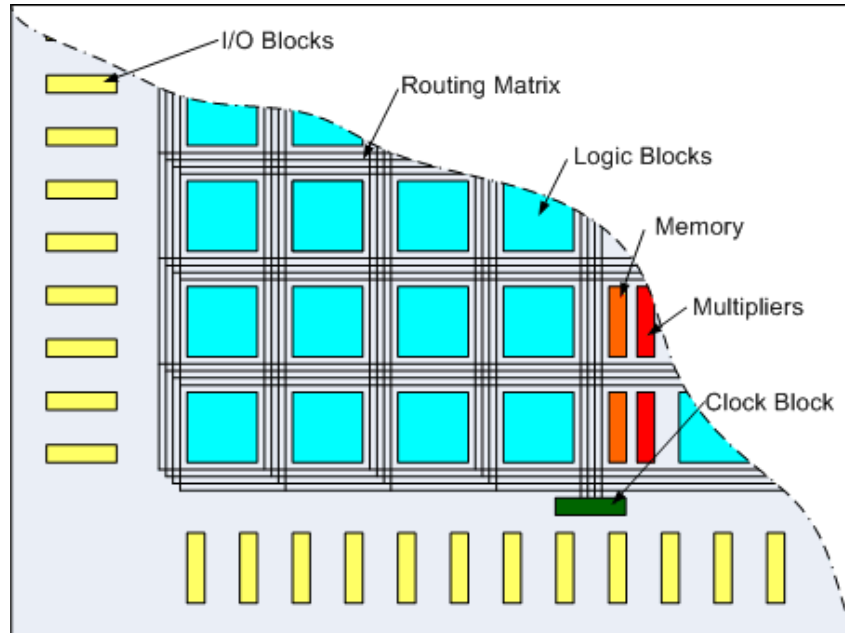


Zoom sur une section du CPLD

18

4.3. Architecture des FPGA

- ➔ Un circuit FPGA est une matrice symétrique comportant des ressources pour la logique, les entrées-sorties, les interconnexions et bien plus encore.
- ➔ Son architecture n'a rien à voir avec celles des PAL et des CPLD.



19

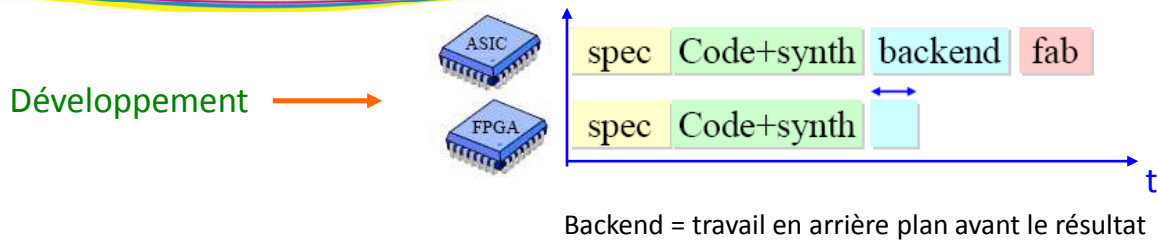
4.4. Marché des FPGA

| Rank 2007 | Rank 2008 | Company | Revenue (\$M) 2007 | Revenue (\$M) 2008 | Revenue Change 2007-2008 | Market Share 2008 |
|-----------|-----------|---|-----------------------|-----------------------|-----------------------------|----------------------|
| 1 | 1 | Xilinx | 1,809 | 1,906 | 5.4% | 51.2% |
| 2 | 2 | Altera | 1,216 | 1,323 | 8.8% | 35.5% |
| 3 | 3 | Lattice Semiconductor | 229 | 222 | -3.1% | 6.0 |
| 4 | 4 | Actel | 196 | 218 | 11.2% | 5.9% |
| 6 | 5 | QuickLogic | 28 | 23 | -17.9% | 0.6% |
| 5 | 6 | Cypress Semiconductor | 32 | 21 | -34.4% | 0.6% |
| 7 | 7 | Atmel | 14 | 9 | -35.7% | 0.2% |
| 8 | 8 | Chengdu Sino (China) Microelectronics System | 4 | 3 | -25.0% | 0.1% |
| | | Others | 0 | 0 | NM | 0.0% |
| | | Total Market | 3,528 | 3,725 | 5.6% | 100.0% |

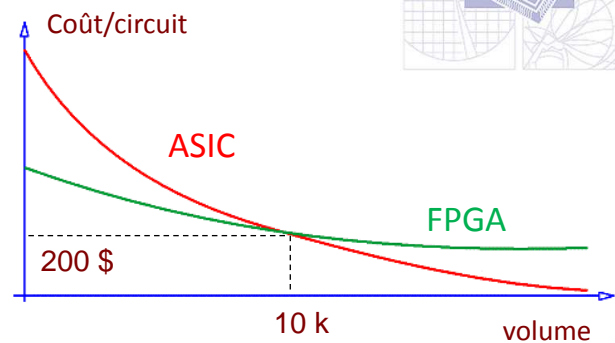
Source: Gartner

20

4.5. Comparaison FPGA /ASIC



- Temps de développement :
Avantage FPGA
- Coût:
Faible volume : Avantage FPGA
Fort volume : Avantage ASIC
- Performances, densité, consommation:
Avantage ASIC
- Evolution maintenance:
Avantage FPGA



21

4.6. Les différents FPGA Xilinx



Automotive Device Families

- XA Spartan-6 FPGAs
- XA Spartan-3A FPGAs
- XA Spartan-3A DSP FPGAs
- XA Spartan-3E FPGAs



Space-grade Device Families

- Space-grade Virtex-5QV FPGAs
- Space-grade Virtex-4QV FPGAs

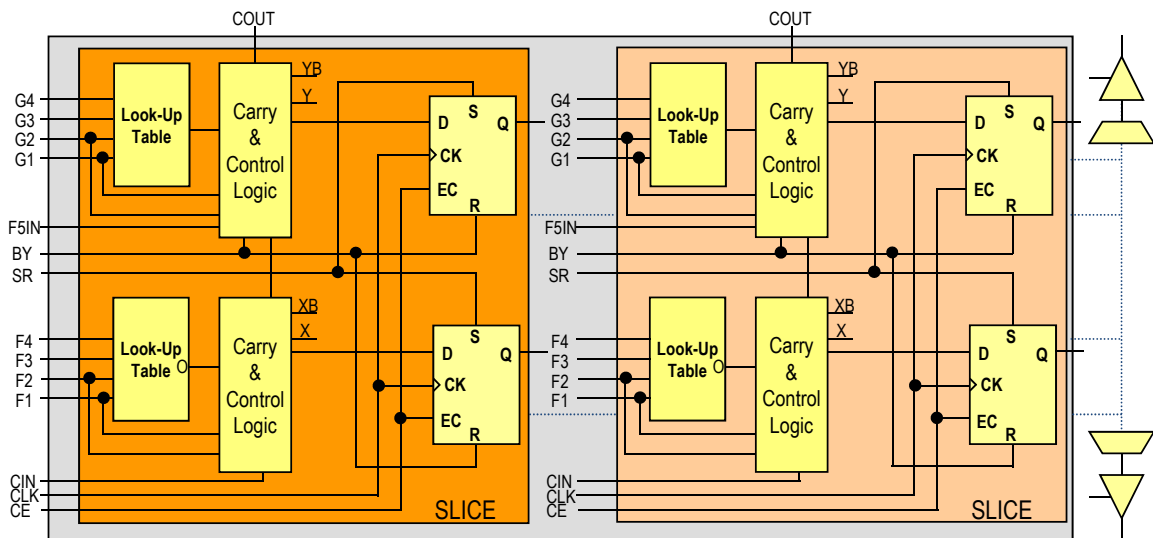


Defense-grade Device Families

- Defense-grade Artix-7Q FPGAs
- Defense-grade Kintex-7Q FPGAs
- Defense-grade Virtex-7Q FPGAs
- Defense-grade Virtex-6Q FPGAs
- Defense-grade Spartan-6Q FPGAs
- Defense-grade Virtex-5Q FPGAs
- Defense-grade Virtex-4Q FPGAs

22

5.1.1. Le bloc logique CLB

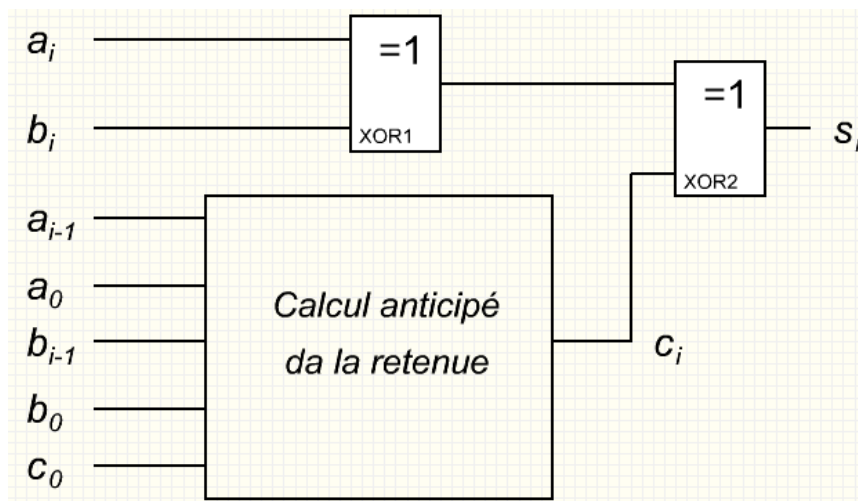


- Chaque CLB comporte 2 tranches (SLICE).
- Un slice contient 2 paires de (LUT + bascule + Retenue).
- 2 buffers 3 états (BUFT) sont associés à chaque CLB.

23

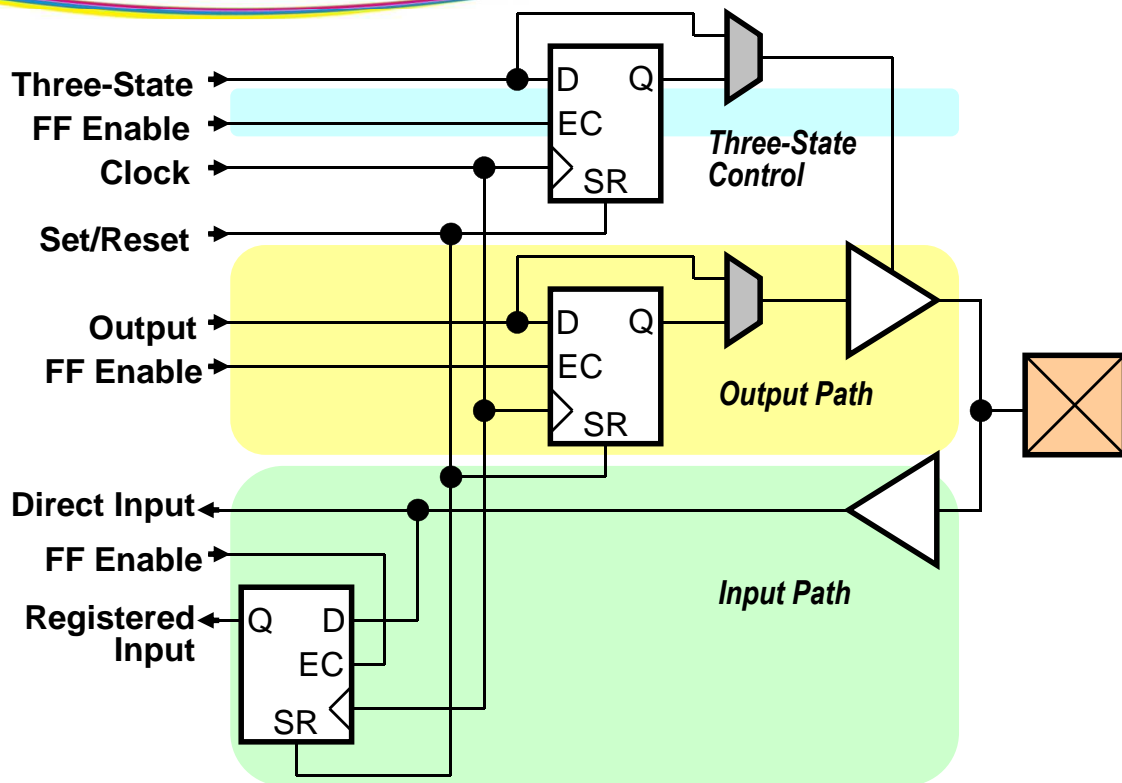
5.1.2. La retenue anticipée

- Dans le cas de l'additionneur complet, la somme est calculée dans une LUT
- La retenue est calculée par anticipation par un circuit câblé et optimisé pour raccourcir les temps de calcul sur des mots à plusieurs bits



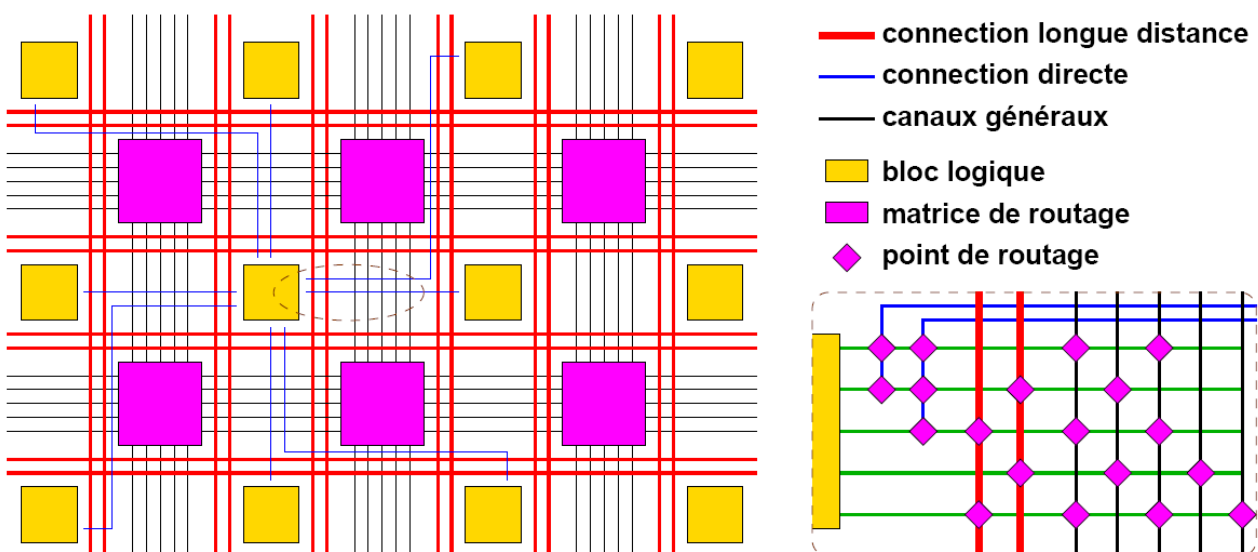
24

5.2. Le bloc d'entrée-sortie (IOB)



25

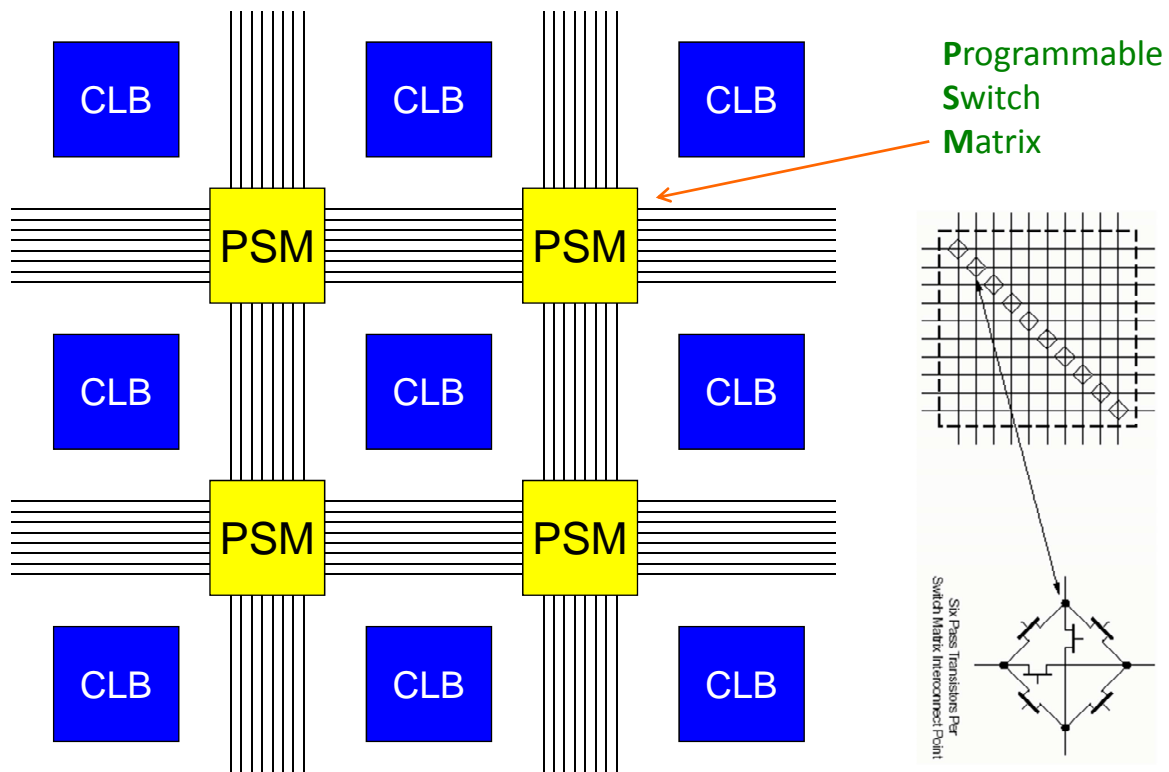
5.3. Les ressources de routage



- ➔ Les ressources de routage sont abondantes dans un FPGA car le routage est un point névralgique dans la compilation d'un design.
- ➔ C'est grâce à ces ressources qu'on atteint un taux de remplissage proche de 80%.

26

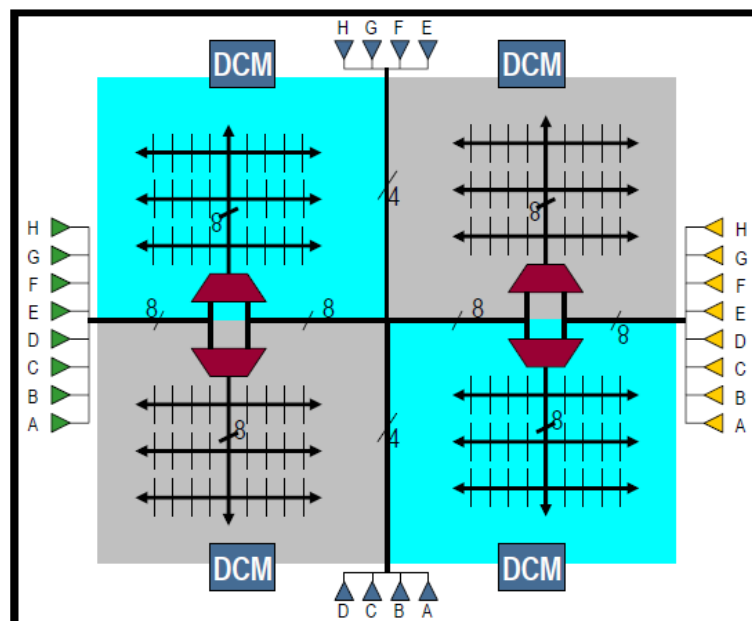
5.4. Les matrices d'aiguillage



27

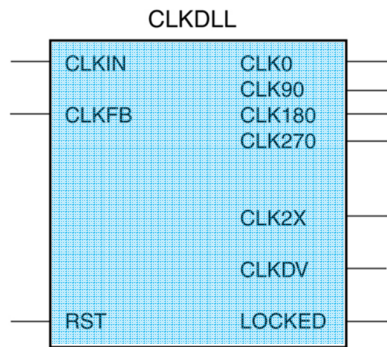
5.5. Routage de l'horloge

- ➔ Un réseau de routage est spécialement dédié à l'horloge pour lui permettre d'atteindre sans délai toutes les bascules du composants.

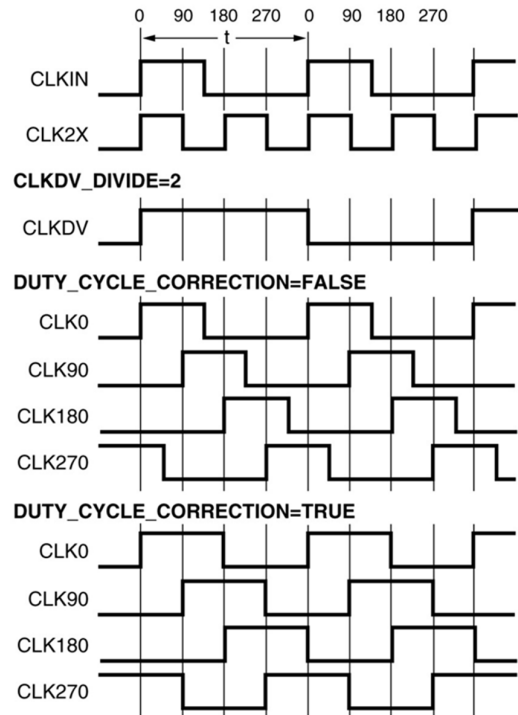


28

Générateur d'horloge 'DLL'

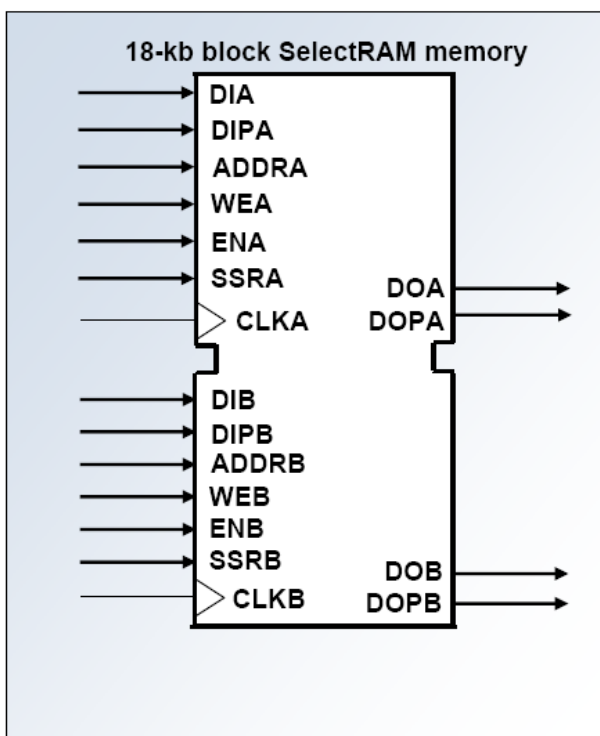


- ➔ Boucle à verrouillage.
- ➔ Elimine le délai .
- ➔ Permet des déphasages.
- ➔ Peut diviser la fréquence d'entrée.
- ➔ Permet aussi une multiplication.

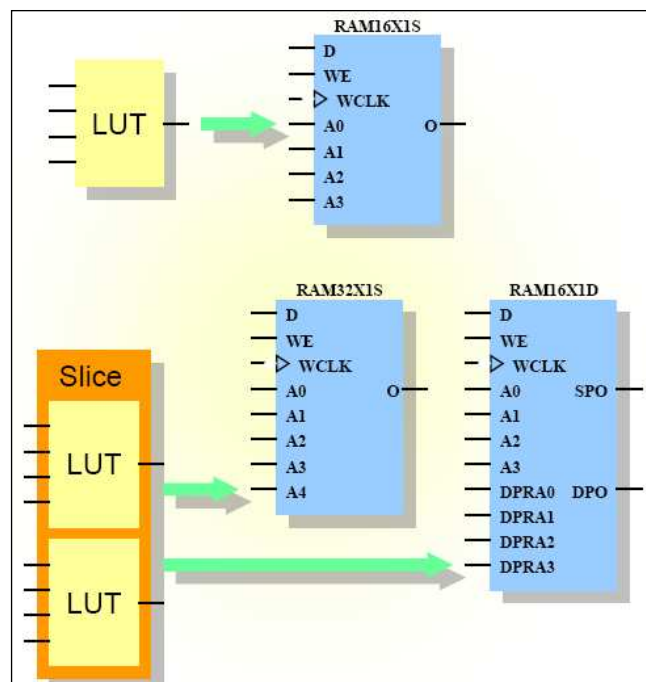


29

6. Bloc RAM et mémoire distribuée



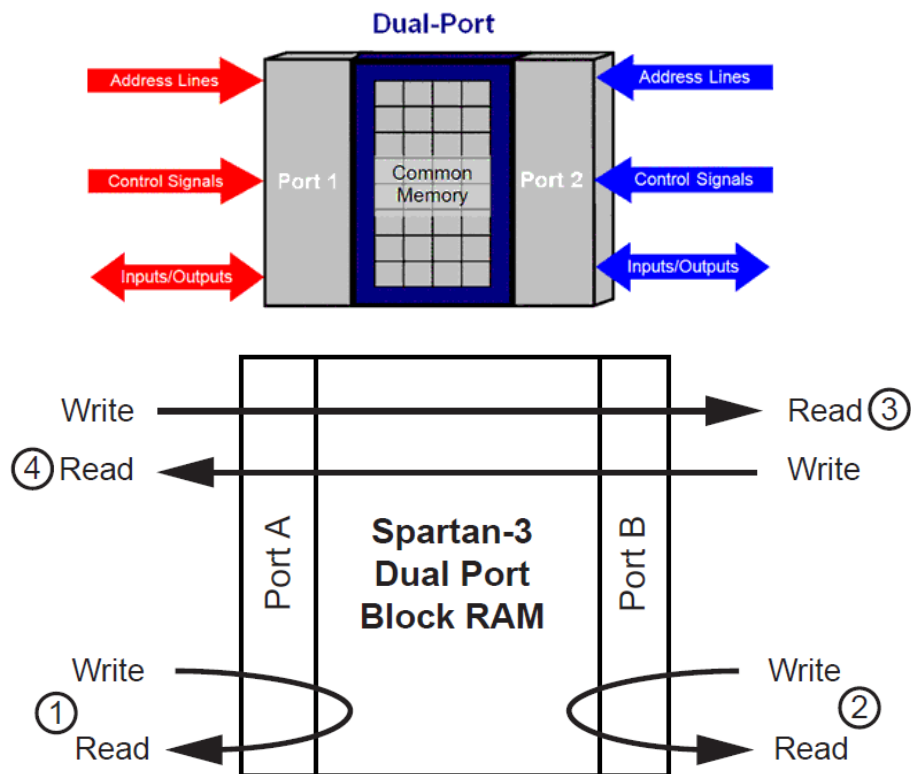
Bloc RAM



Mémoire distribuée

30

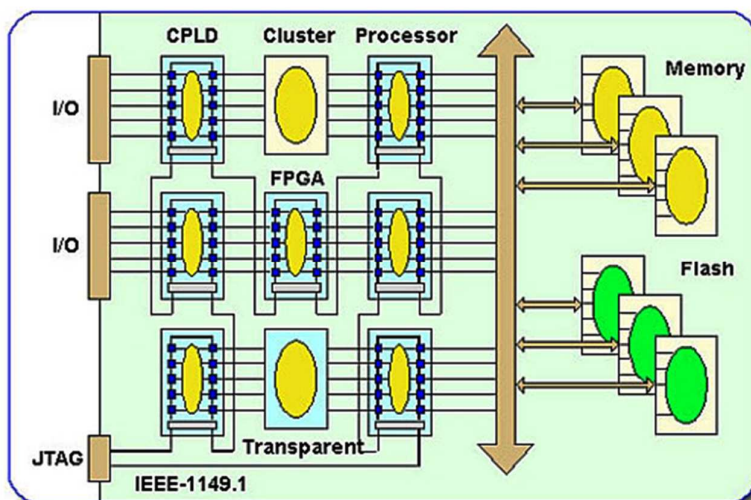
Dans une mémoire simple port, on ne peut accéder qu'à une donnée à la fois, alors que dans une mémoire à double port, on peut accéder à 2 données en même temps,



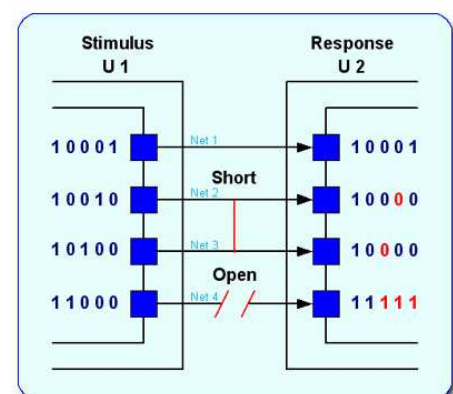
31

7. Le test 'Boundary Scan'

- Boundary Scan est un standard 'JTAG' pour les test physique (coul-circuits, coupures), et le test fonctionnel de cartes électroniques denses comportant des boîtiers de type 'BGA'.
- Il utilise 4 signaux (TDI : entrée, TDO : sortie, TCK : horloge, TMS : sélection du mode).



Carte avec une variété de composants



Exemple de test physique

32

8.1. Les familles de la série 7

| | ARTIX ⁷ | KINTEX ⁷ | VIRTEX ⁷ |
|-------------------------|-----------------------|-----------------------------------|---------------------------------------|
| Maximum Capability | Lowest Power and Cost | Industry's Best Price/Performance | Industry's Highest System Performance |
| Logic Cells | 20K – 355K | 70K – 480K | 285K – 2,000K |
| Block RAM | 12 Mb | 34 Mb | 65 Mb |
| DSP Slices | 40 – 700 | 240 – 1,920 | 700 – 3,960 |
| Peak DSP Perf. | 504 GMACS | 2,450 GMACS | 5,053 GMACS |
| Transceivers | 4 | 32 | 88 |
| Transceiver Performance | 3.75Gbps | 6.6Gbps and 12.5Gbps | 12.5Gbps, 13.1Gbps and 28Gbps |
| Memory Performance | 1066Mbps | 1866Mbps | 1866Mbps |
| I/O Pins | 450 | 500 | 1,200 |
| I/O Voltages | 3.3V and below | 3.3V and below 1.8V and below | 3.3V and below 1.8V and below |

33

8.2. Une architecture alignée

- Common elements enable easy IP reuse for quick design portability across all 7 series families
 - Design scalability from low-cost to high-performance
 - Expanded eco-system support
 - Quickest TTM

 **Logic Fabric**
LUT-6 CLB

 **Precise, Low Jitter Clocking**
MMCMs

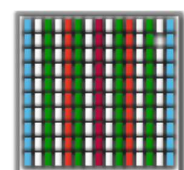
 **On-Chip Memory**
36Kbit/18Kbit Block RAM

 **Enhanced Connectivity**
PCIe® Interface Blocks

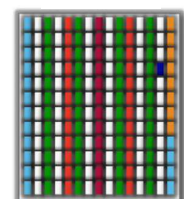
 **DSP Engines**
DSP48E1 Slices

 **Hi-perf. Parallel I/O Connectivity**
SelectIO™ Technology

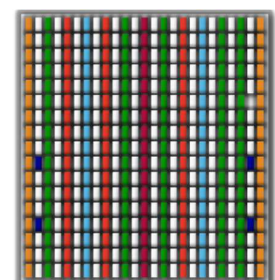
 **Hi-performance Serial I/O Connectivity**
Transceiver Technology



Artix™-7 FPGA



Kintex™-7 FPGA



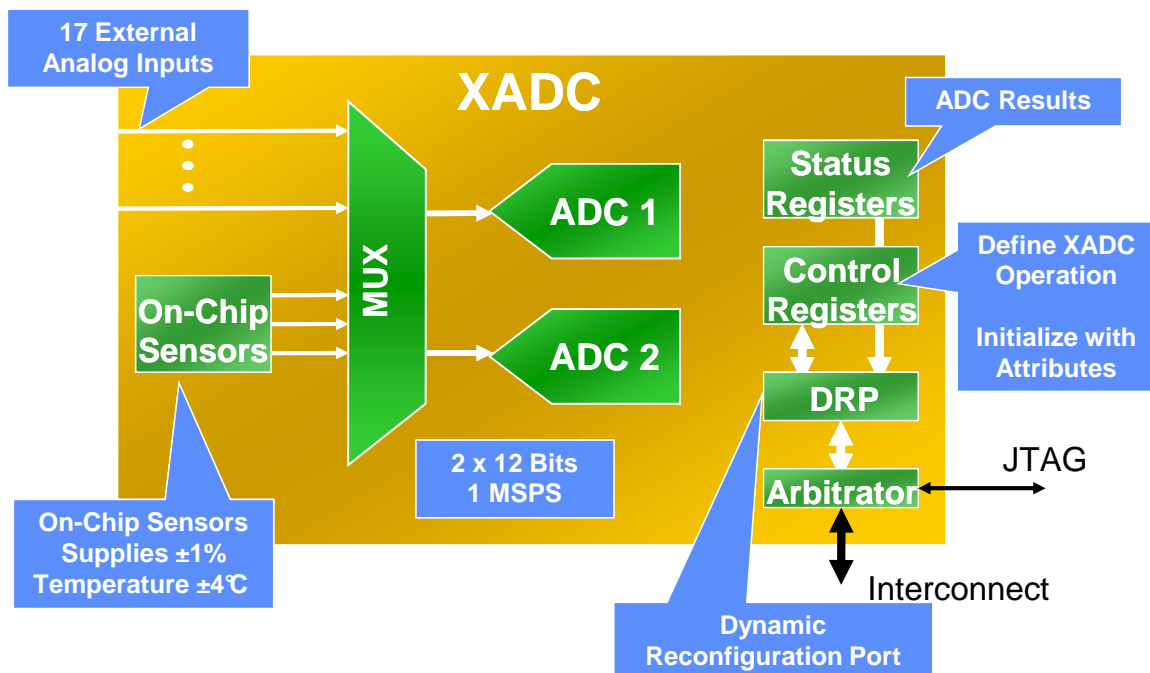
34

-



-

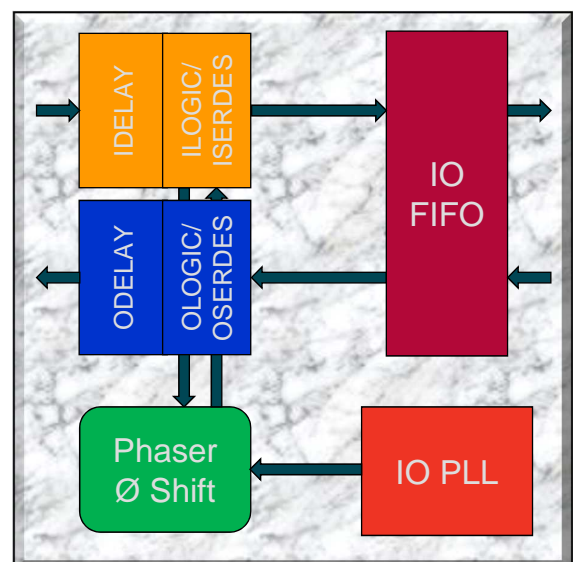
8.5. Le convertisseur « ADC »



37

8.6. Le bloc d'entrée/sortie

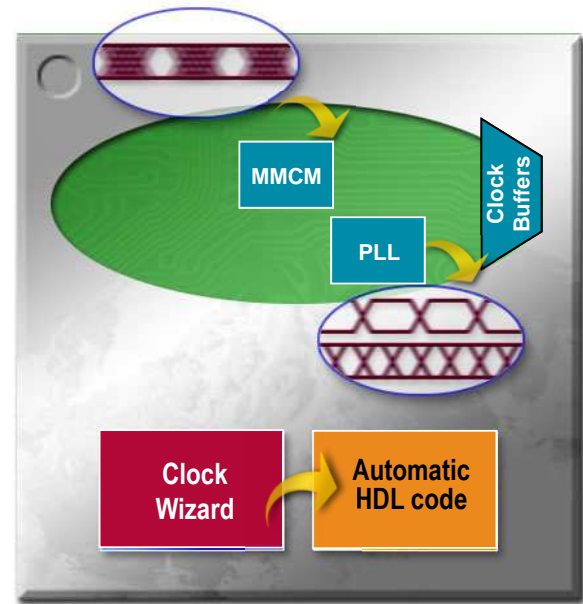
- Two distinct I/O types
 - High range: Supports standards up to 3.3V
 - High performance: Higher performance with more I/O delay capability
 - Supports I/O standards up to 1.8V
- Extension of logic layer functionality
 - Wider input/output SERDES (sérailise)
 - Addition of independent ODELAY
- New hardware blocks to address highest I/O performance
 - Phaser, IO FIFO, IO PLL



38

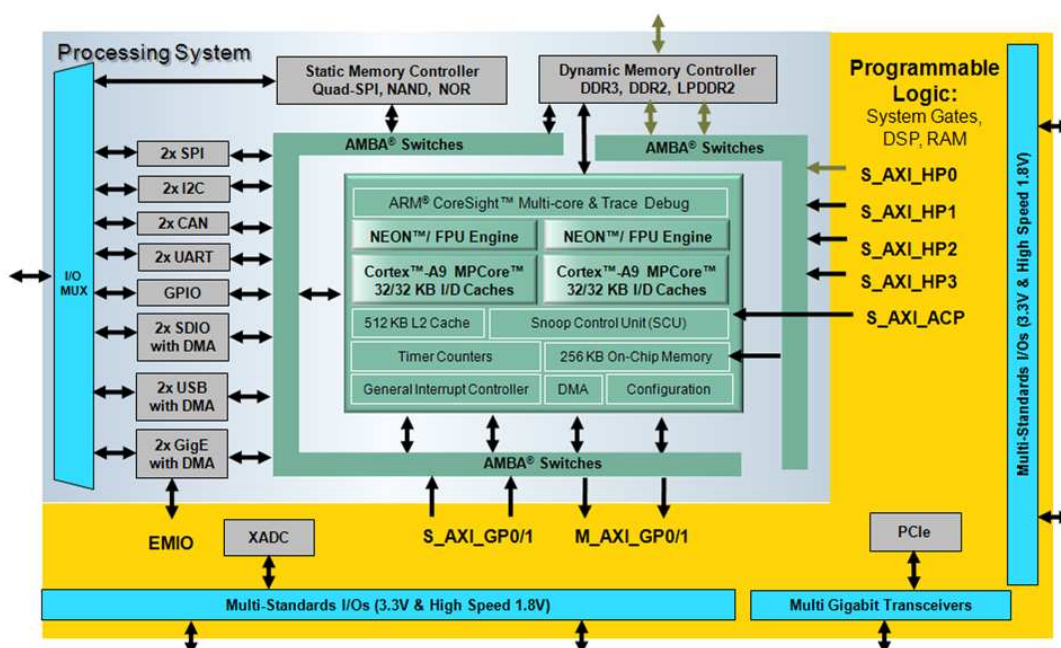
8.7. Générateur d'horloge

- Based on the established Virtex-6 FPGA clocking structure
 - All 7 series FPGAs use the same unified architecture
- Low-skew clock distribution
 - Combination of paths for driving clock signals to and from different locations
- Clock buffers
 - High fanout buffers for connecting clock signals to the various routing resources
- Clock regions
 - Device divided into clock regions with dedicated resources
- Clock management tile (CMT)
 - One MMCM and one PLL per CMT
 - Up to 24 CMTs per device



39

9. La famille Zync 7000



Le composant embarque un processeur ARM9 à 2 cœurs, des contrôleurs de différents types de mémoires, divers interfaces d'un côté, et de l'autre côté, de la logique programmable.

Ces 2 parties communiquent via un bus AXI. Circuit idéal pour les SOC (System On Chip)

40

10.1. Développement

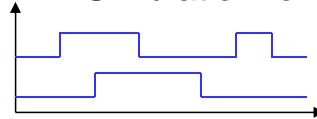
Design and implement a simple unit permitting to speed up encryption with RCS-similar cipher with fixed key set on 8031 microcontroller. Unlike in the experiment 5, this time your unit has to be able to perform an encryption algorithm by itself, executing 32 rounds.....

```
Library IEEE;  
use ieee.std_logic_1164.all;  
use ieee.std_logic_unsigned.all;  
  
entity RCS_core is  
  port(  
    clock, reset, encr_decr: in std_logic;  
    data_input: in std_logic_vector(31 downto 0);  
    data_output: out std_logic_vector(31 downto 0);  
    out_full: in std_logic;  
    key_input: in std_logic_vector(31 downto 0);  
    key_read: out std_logic;  
  );  
end RCS_core;
```

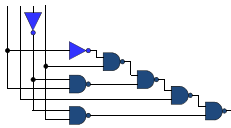
Specifications

Description VHDL

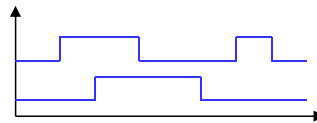
Simulation fonctionnelle



Synthèse logique



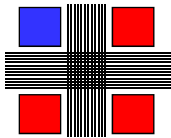
Simulation après synthèse



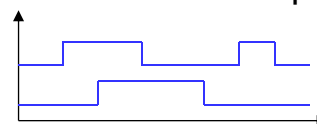
41

Développement (suite)

Implementation



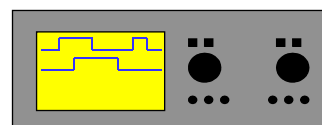
Simulation temporelle



Configuration

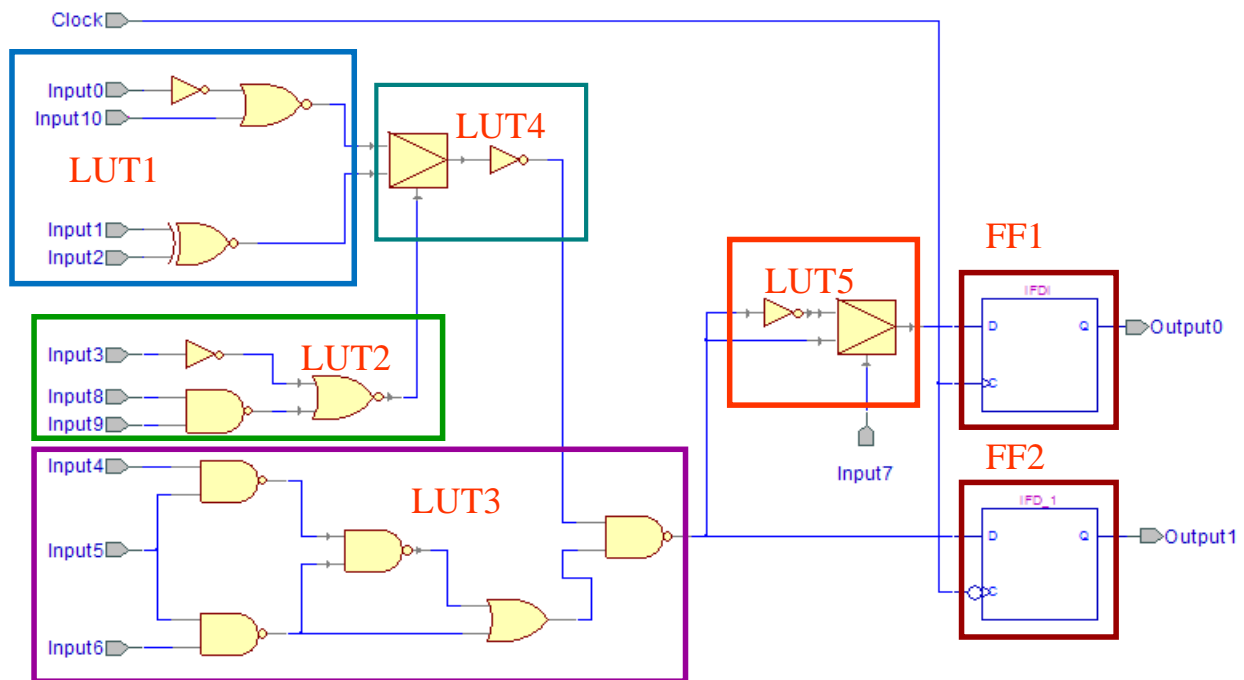


Test sur site



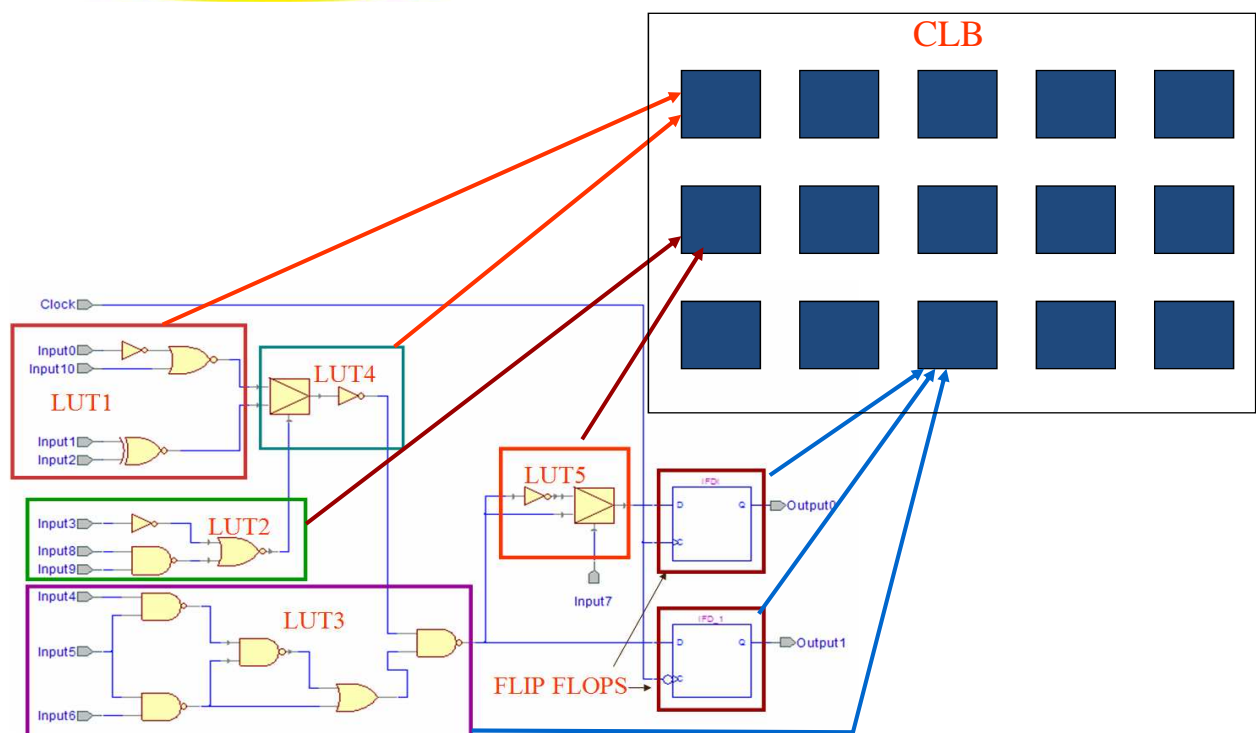
42

10.2. Implémentation « mapping »



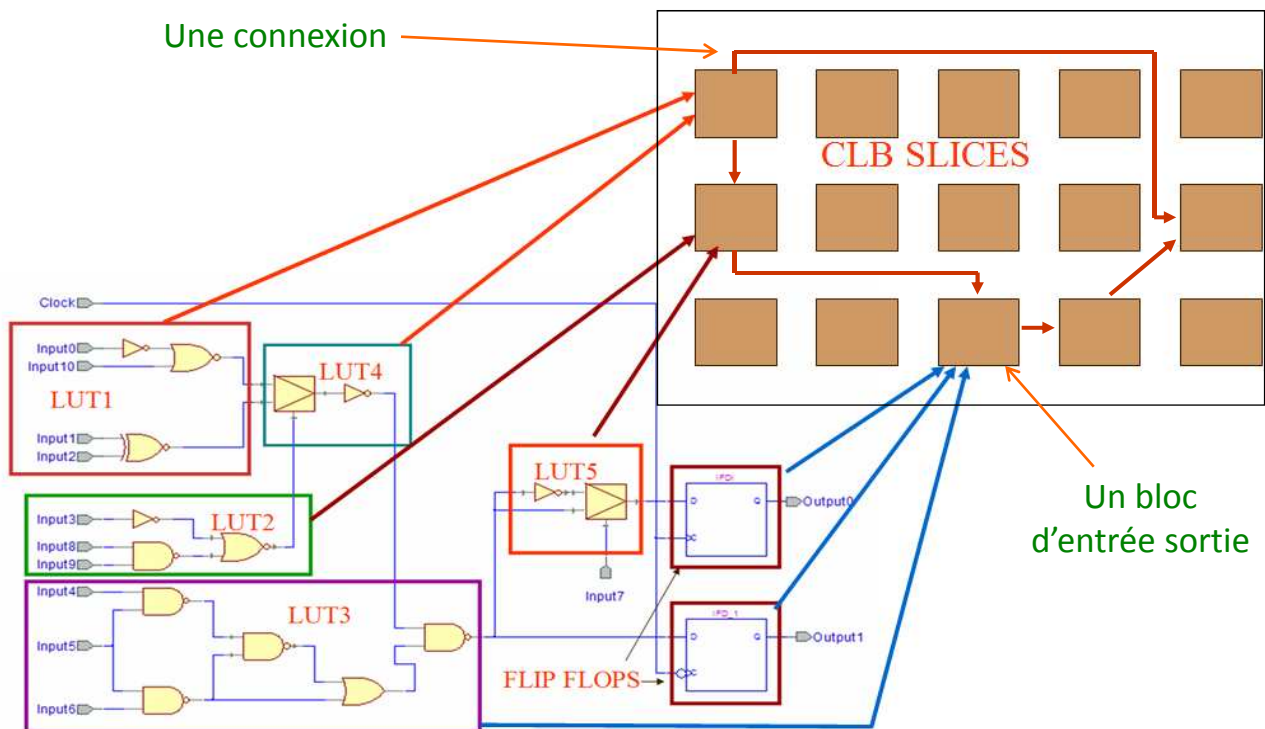
43

Implémentation « placement »



44

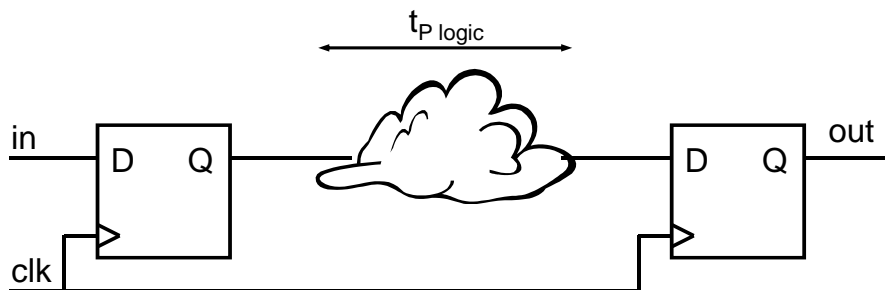
Implémentation « routage »



45

10.3. Le « timing »

- Chemin critique = chemin reliant deux bascules et ayant le plus long délai.
- C'est de lui que dépendra la fréquence maximale de l'horloge.



$$t_{\text{Critical}} = t_{P \text{ FF}} + t_{P \text{ logic}} + t_{S \text{ FF}}$$

- Période minimale d'horloge = délai du chemin critique.
- Celle-ci est calculée automatiquement par les outils de développement IDE.

46

10.4. La programmation

