

Conception et fabrication des microsystèmes

**Master Ingénierie des Systèmes
Embarqués**

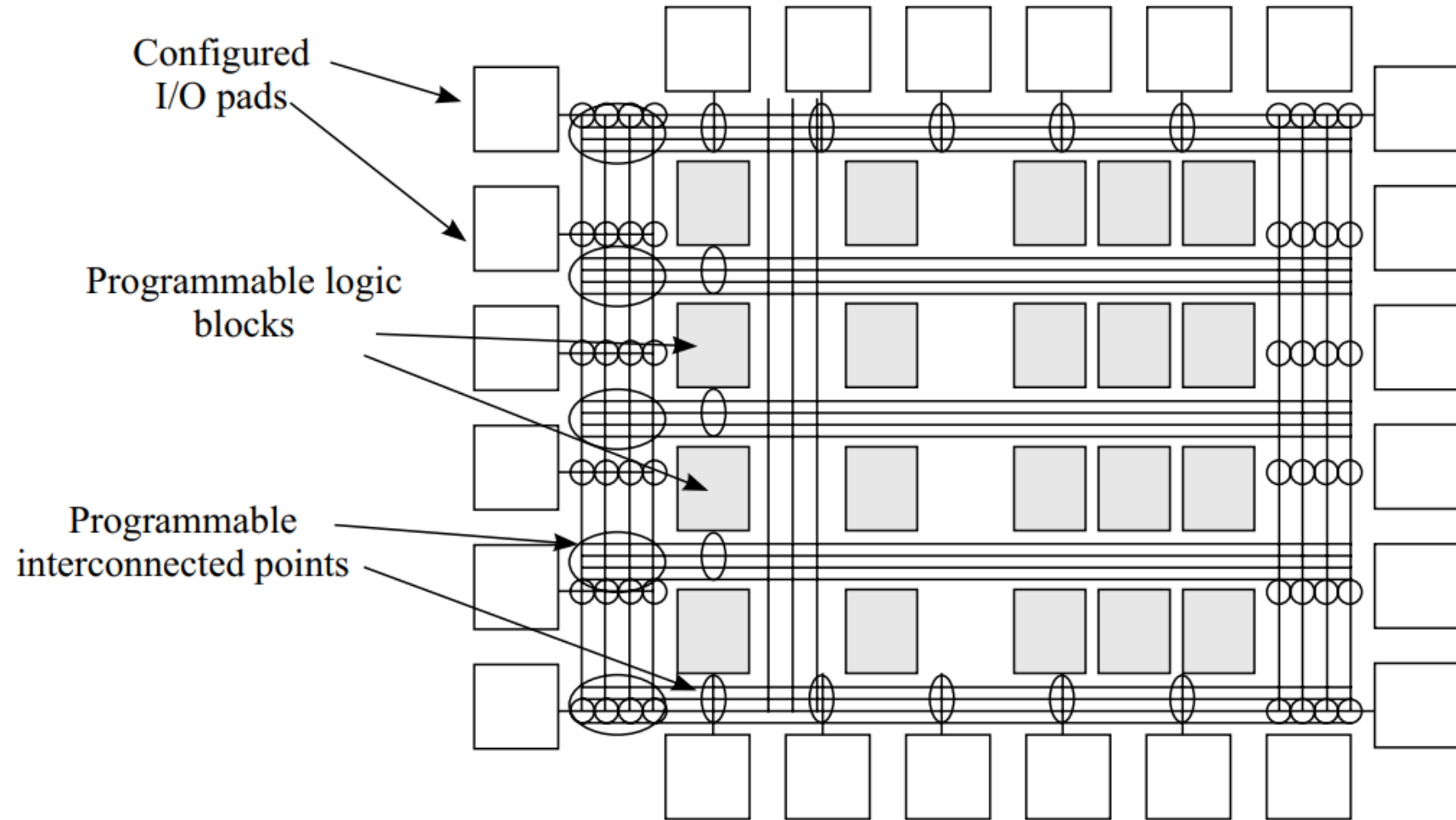
**Chapitre 9:
Conception des circuits FPGA**

Chapitre 9: Conception des circuits FPGA

- Architecture générale d'un circuit FPGA (Field Programmable Gate Array)

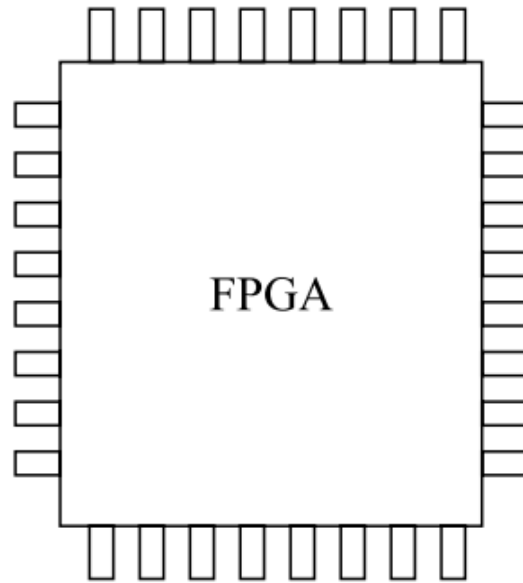
- La figure suivante illustre l'architecture générale d'un circuit FPGA. Elle est constitué de:

- Blocs logiques configurables le CLB (Configurable Logic Block)
- Des entrées / sorties configurables
- Et des interconnexions configurables

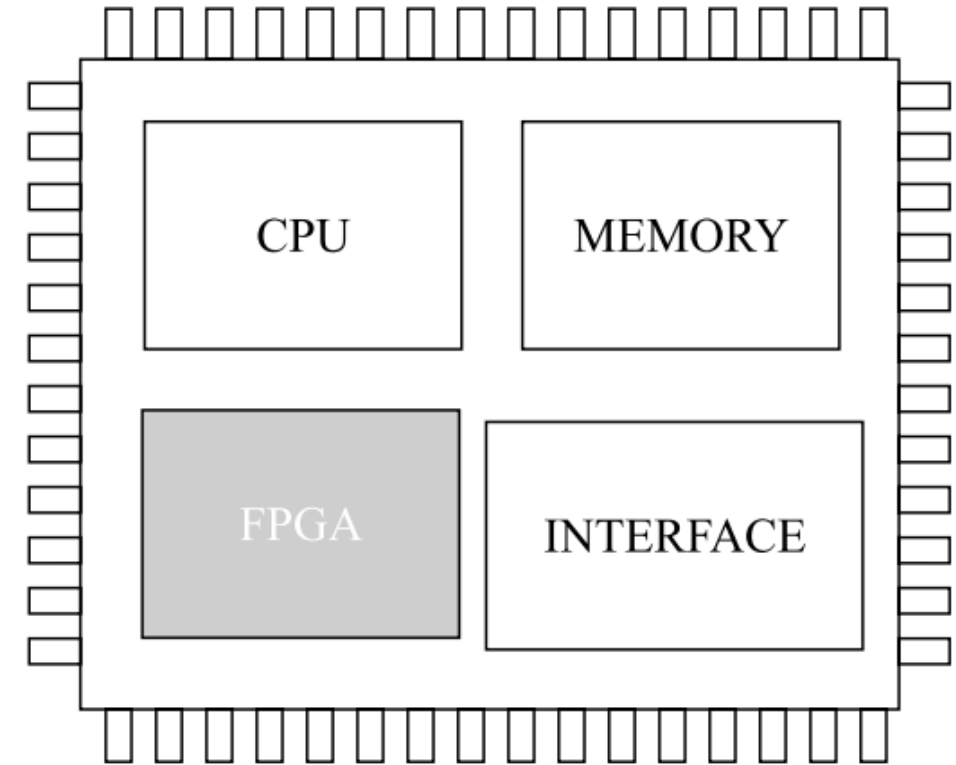


Chapitre 9: Conception des circuits FPGA

- Architecture générale d'un circuit FPGA (Field Programmable Gate Array)
- Les circuits FPGA peuvent parfois intégrer d'autres éléments tels que des microprocesseurs, des mémoires, des interfaces de communications, des oscillateurs (générateurs de fréquences)....



(a) Stand-alone FPGA



(b) FPGA internal block

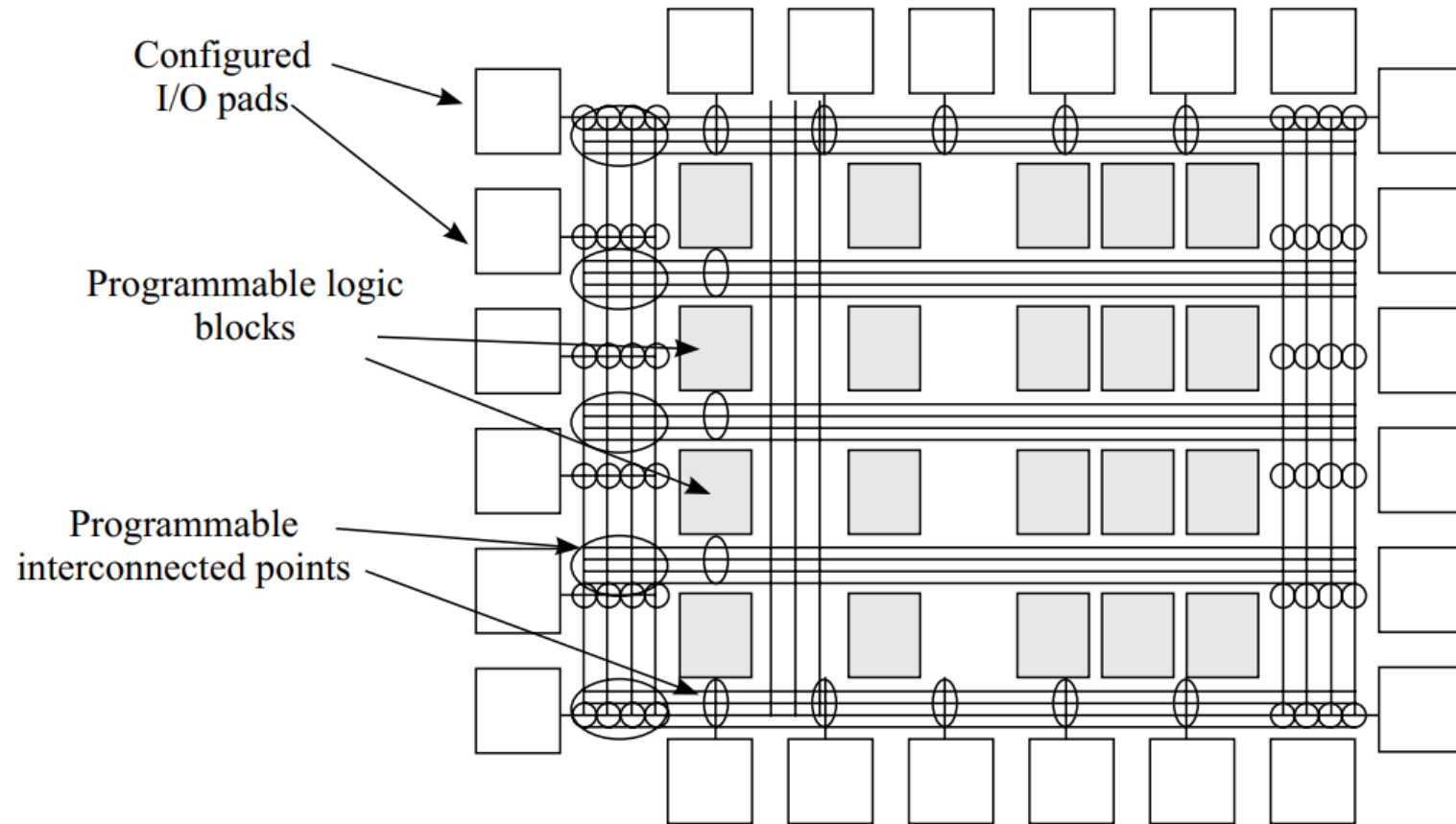
Chapitre 9: Conception des circuits FPGA

- Architecture générale d'un circuit FPGA (Field Programmable Gate Array)

- Le bloc logique configurable (CLB):**

le CLB constitue la cellule de base de tous les circuits FPGA. Il doit être capable d'implémenter toutes les fonctions logiques de base: AND, NAND, NOR, OR, XOR, XNOR....

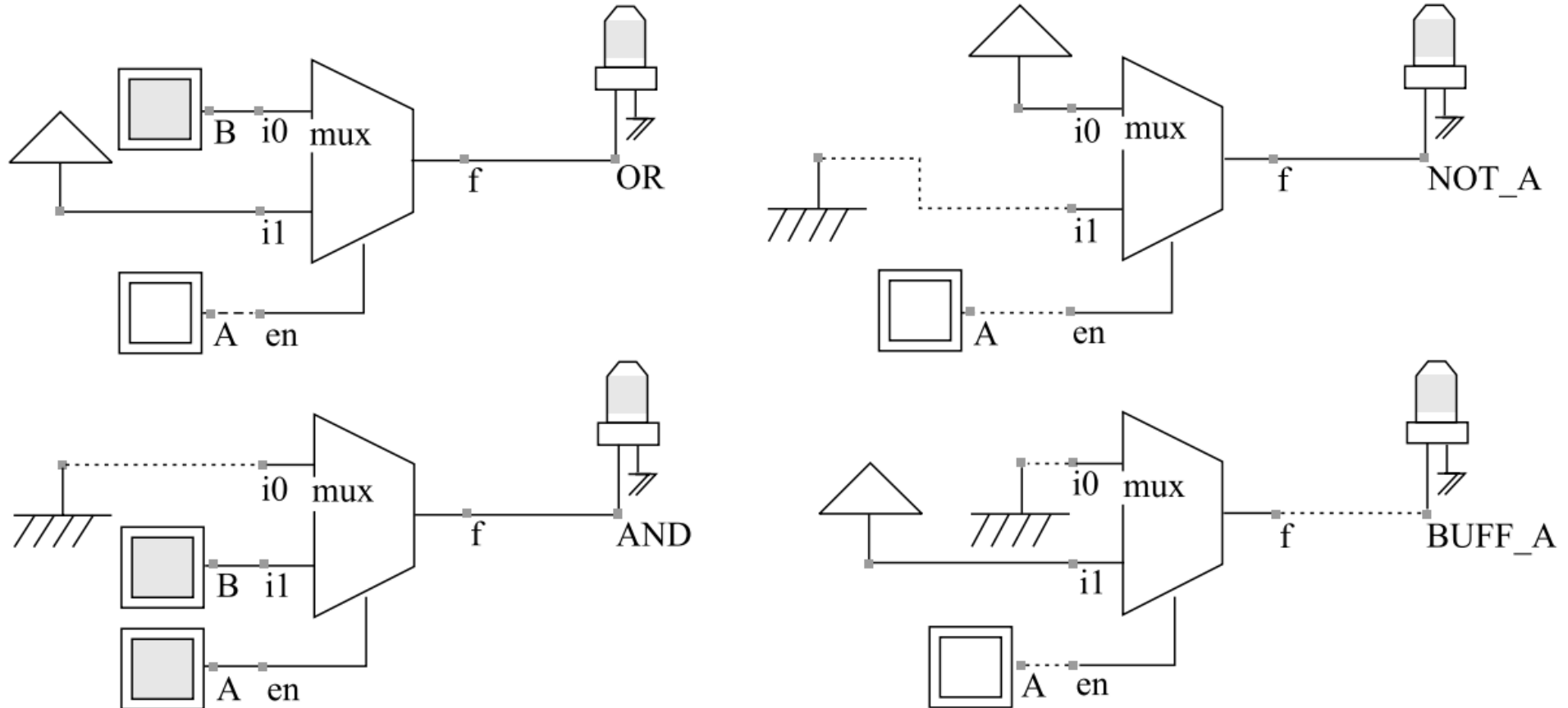
Les performances d'un circuit FPGA (sa puissance de calcul et de traitement) dépendent du nombre de CLB intégrés.



Chapitre 9: Conception des circuits FPGA

- Le bloc logique configurable (CLB) à base de multiplexeur:

Un multiplexeur 2 vers 1 peut être utilisé pour implémenter divers fonctions logiques



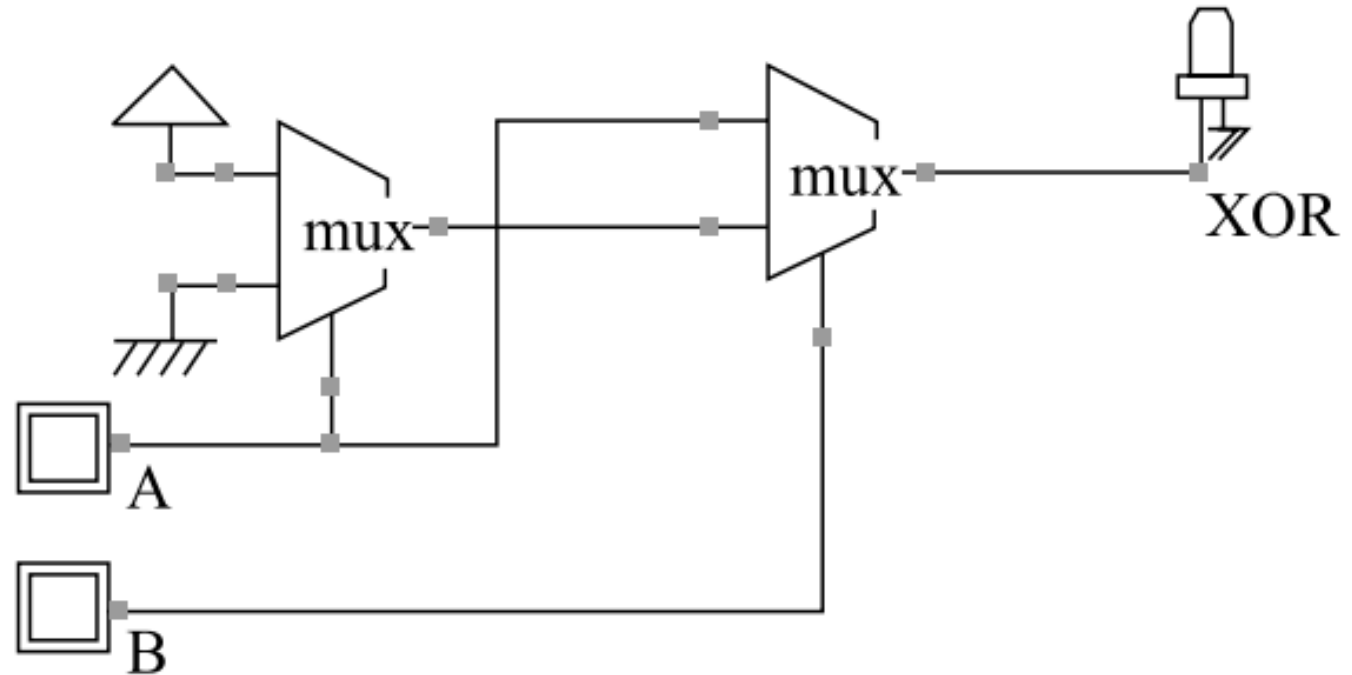
Chapitre 9: Conception des circuits FPGA

- **Le bloc logique configurable (CLB) à base de multiplexeur:**

Les fonctions NAND, NOR et XOR ne peuvent pas être réalisées directement à partir d'un seul multiplexeur 2 vers 1 mais elles nécessitent plusieurs MUX.

- **L'inconvénient des multiplexeurs:**

L'utilisation de plusieurs multiplexeurs pour la réalisation des portes logiques complexes **augmente le temps de propagation du signal** ce qui détériore les performances du circuit en termes de rapidité.

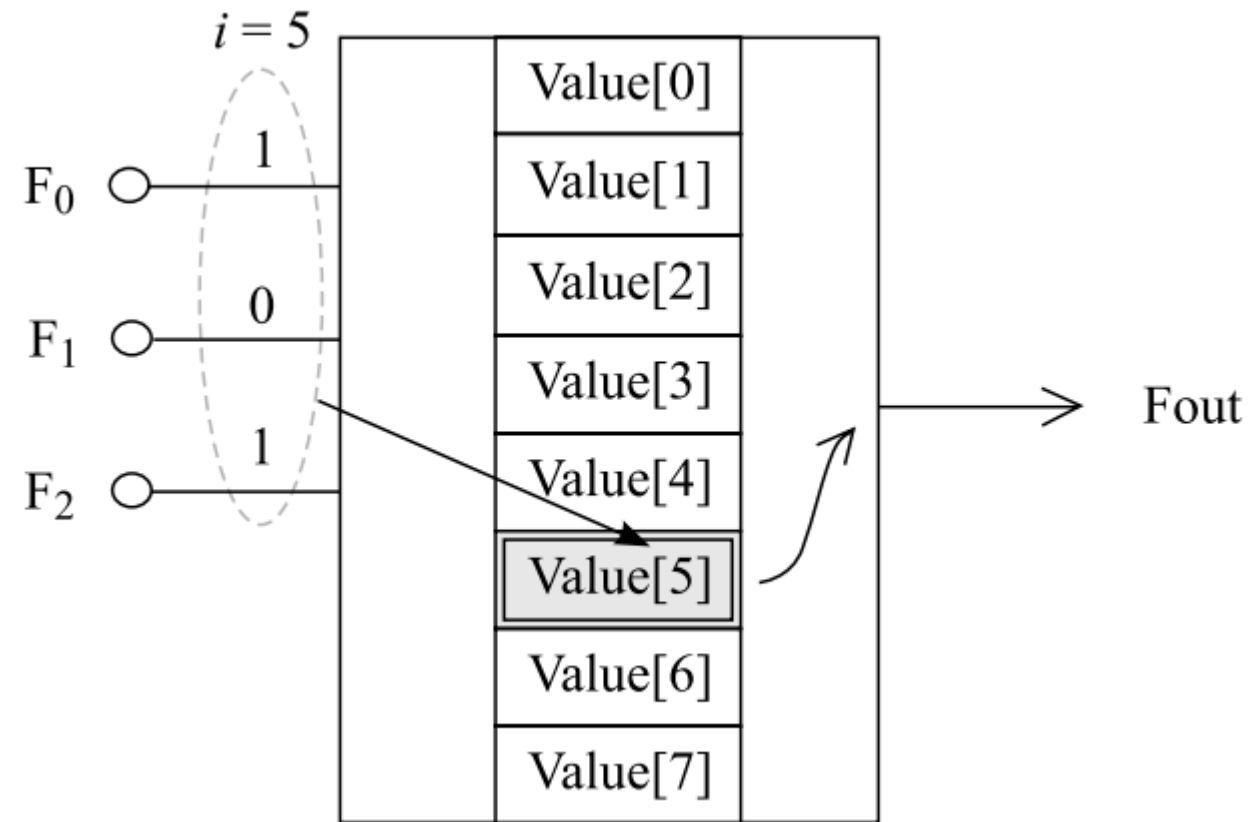


Chapitre 9: Conception des circuits FPGA

- **Le bloc logique configurable (CLB) à base de Look Up Table (LUT):**

Les LUT constituent la solution la plus adoptée pour la réalisation des CLB des circuits FPGA

- La figure suivante illustre un circuit LUT à trois entrées **F0**, **F1**, **F2** et une seule sortie **Fout**
- La table de vérité de la fonction **Fout = f(F0, F1, F2)** est stockée dans une **mémoire de 8bits** (car nous avons 8 combinaisons)
- En fonction des entrées **F0**, **F1**, **F2** une seule case mémoire sera sélectionnée et son contenu sera disponible sur la sortie **Fout**.

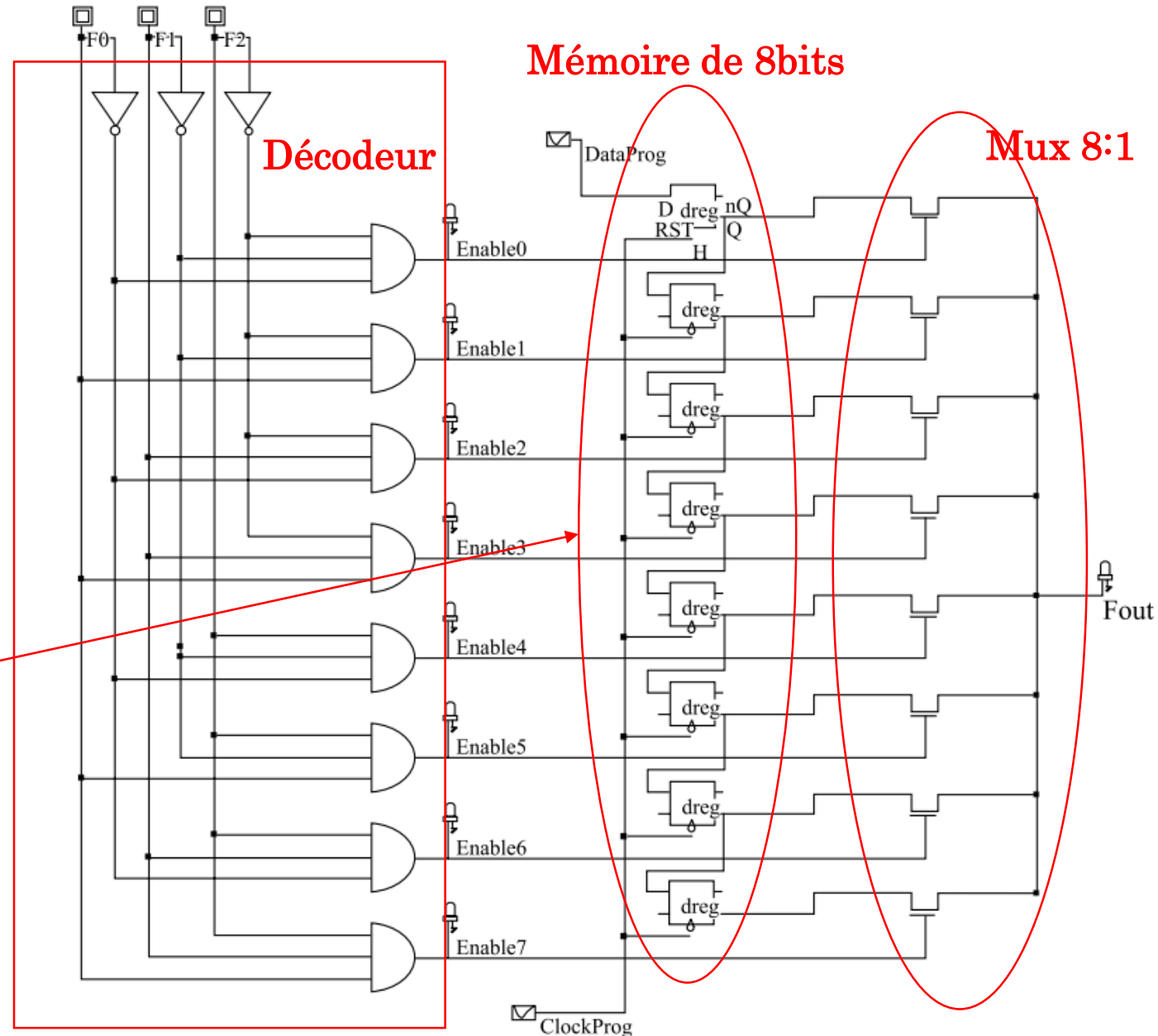


Chapitre 9: Conception des circuits FPGA

- Architecture d'une Look Up Table (LUT):

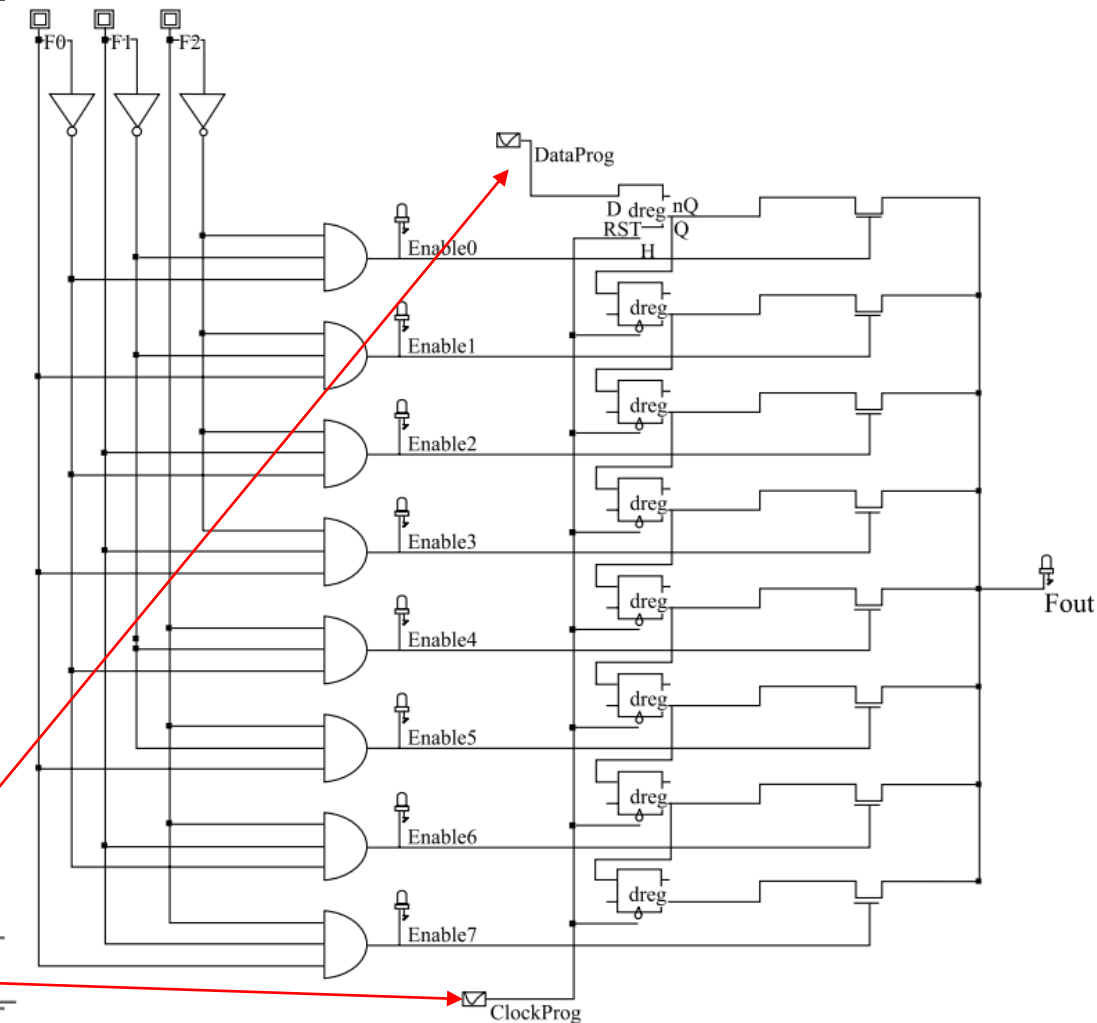
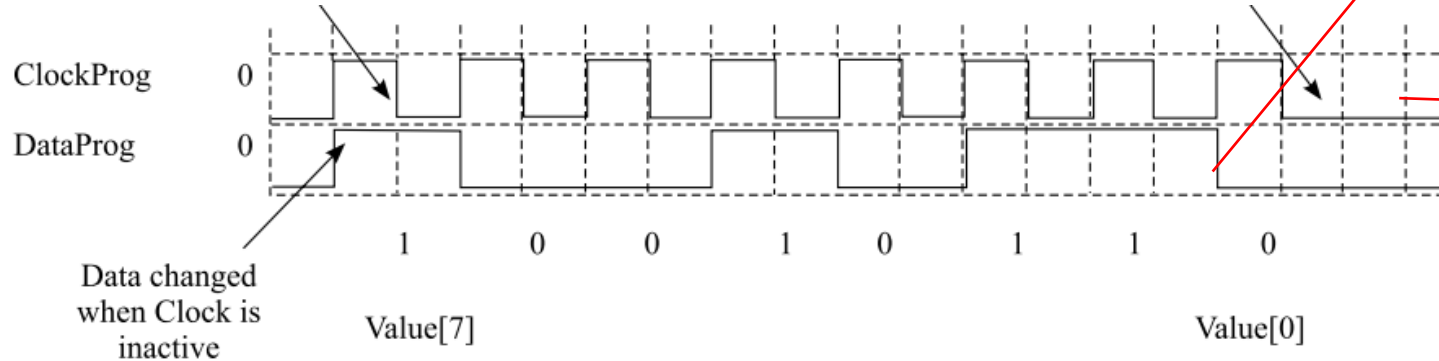
Un registre à décalage de 8 bascules D est utilisé pour mémoriser la fonction Fout

F0	F1	F2	Fout
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



Chapitre 9: Conception des circuits FPGA

- **Programmation d'une LUT**
- La programmation des registres se fait selon la manière suivante:
- La table de vérité de la fonction Fout est placée sur **DataProg** (donnée sur 8bits)
- Un signal d'horloge **ClockProg** est utilisé pour écrire le contenu de **DataProg** dans les bascules du registre



Chapitre 9: Conception des circuits FPGA

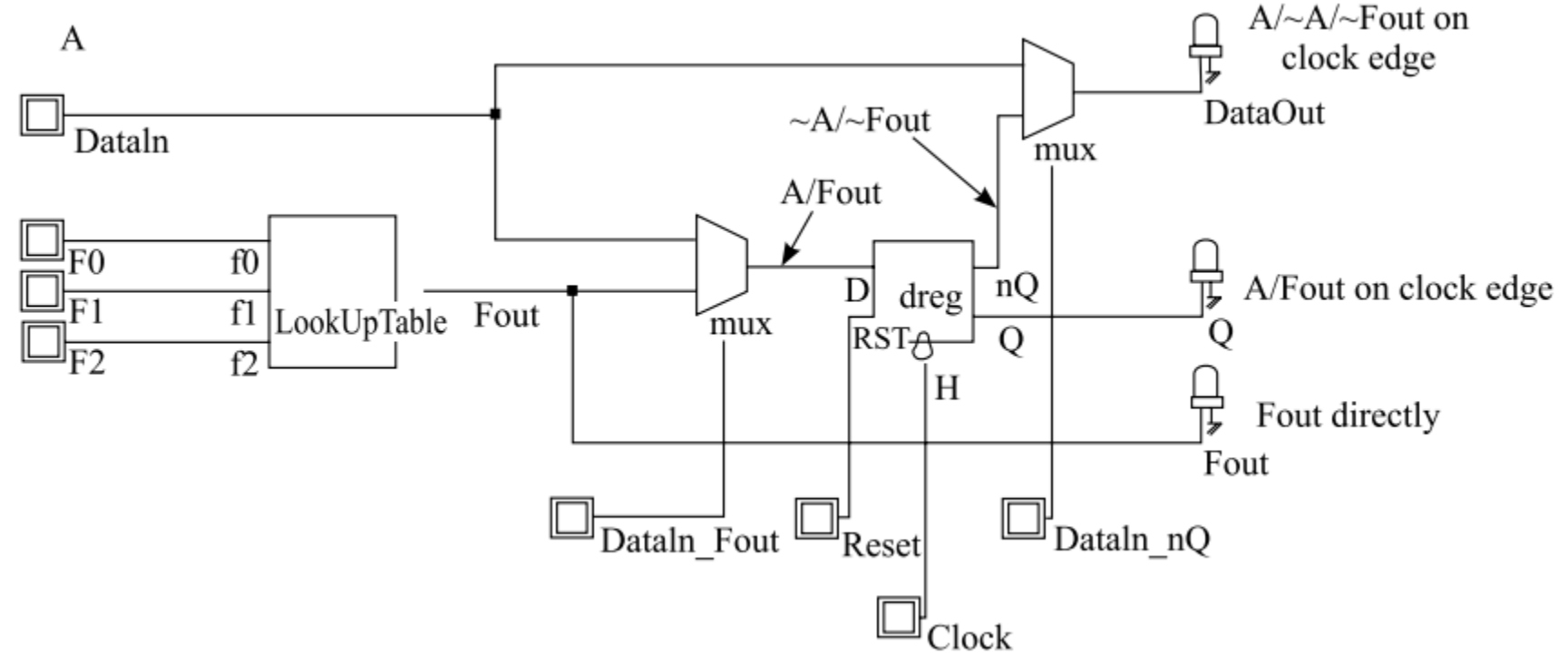
- **Architecture d'un bloc logique configurable (CLB) à base de Look Up Table (LUT):**

- L'architecture simplifiée d'un CLB est constituée d'une:

- LUT: à trois entrées dans notre cas (F0, F1, F2)

- Bascule D de stockage

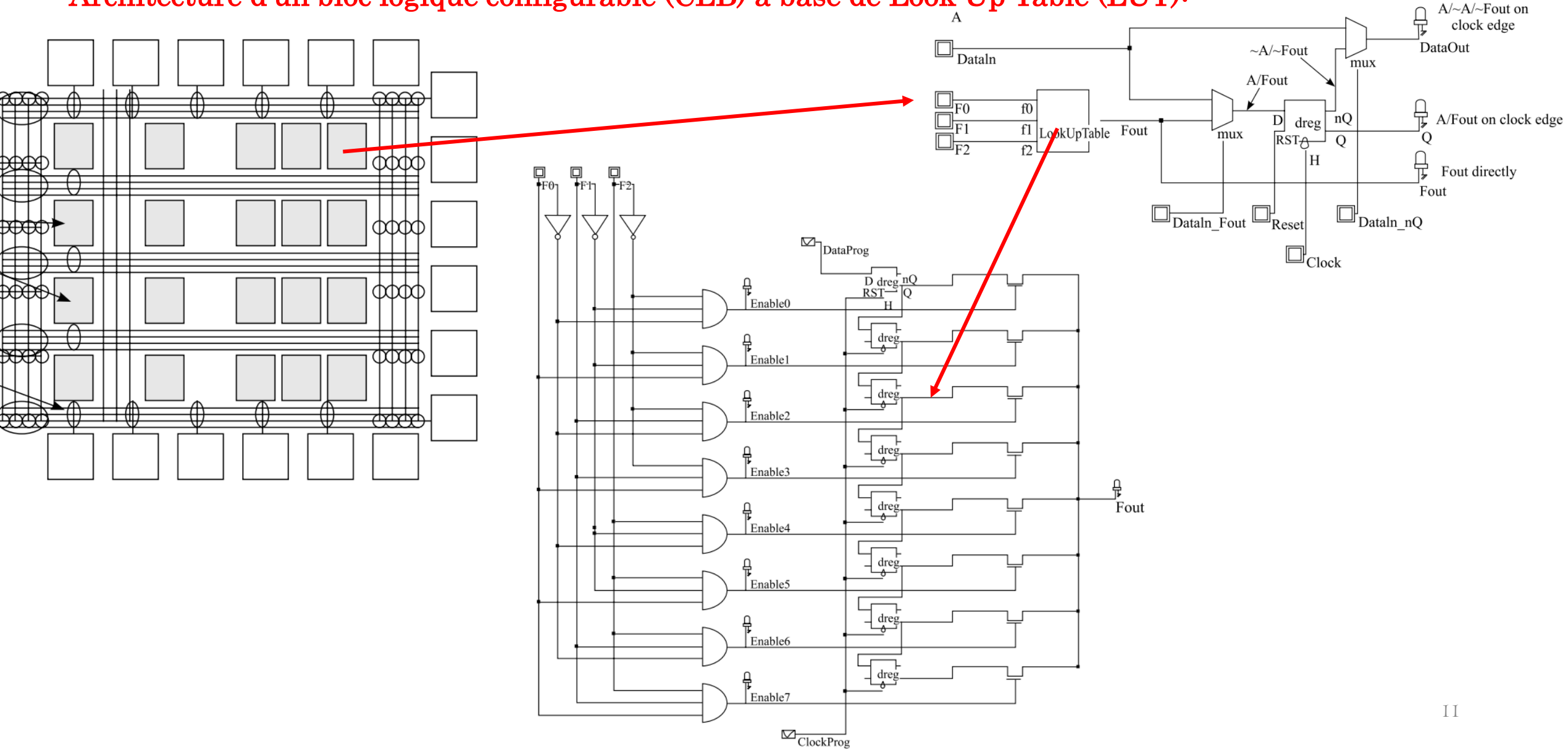
- De quelques multiplexeurs permettant de configurer plusieurs scénarios de configurations



- La sortie Fout peut être connectée directement vers la sortie ou bien stockée dans la bascule D
- Une autre entrée DataIn (A) provenant d'un autre CLB peut être aussi stockée dans la bascule D et aussi inversée.

Chapitre 9: Conception des circuits FPGA

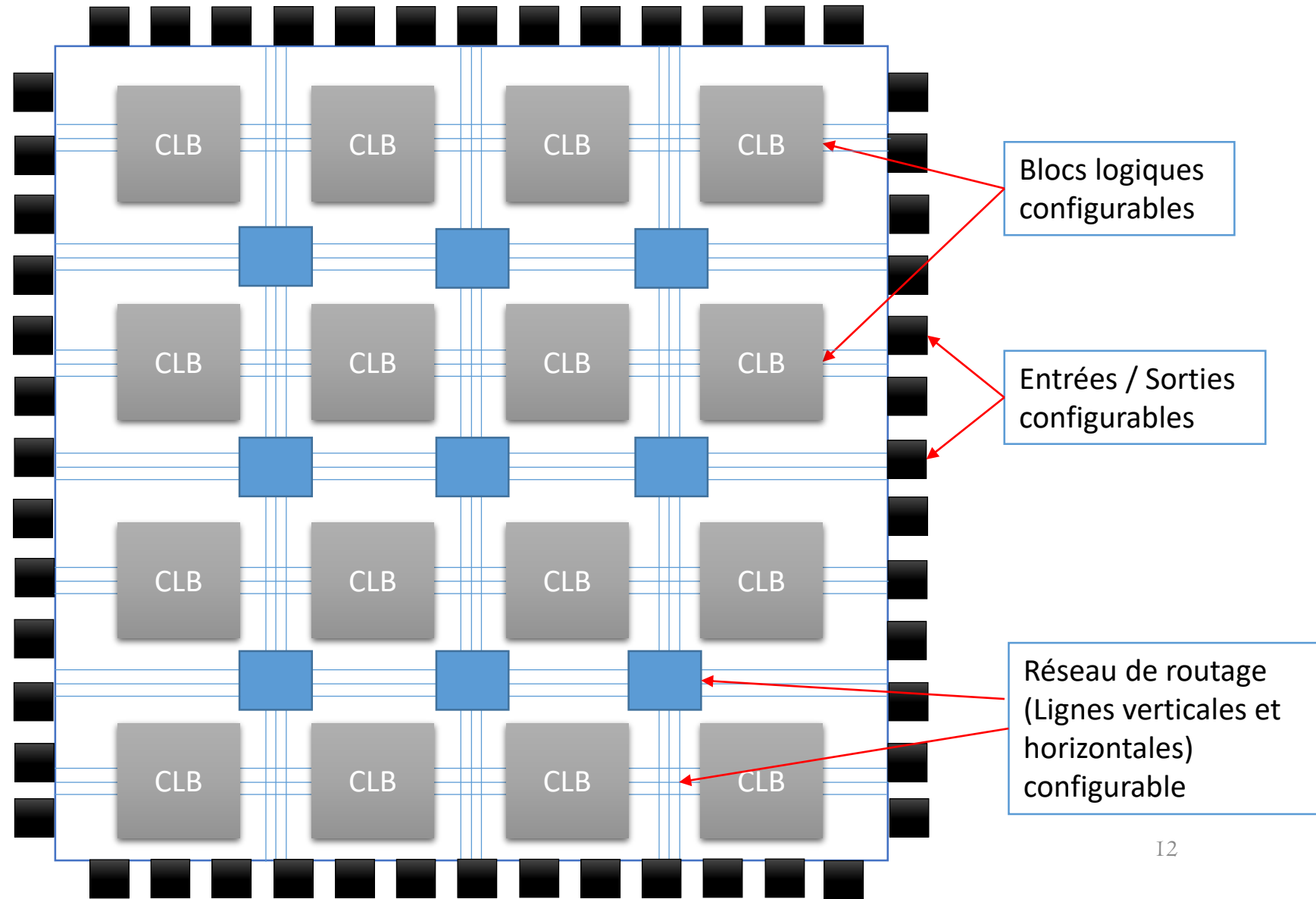
- Architecture d'un bloc logique configurable (CLB) à base de Look Up Table (LUT):



Chapitre 9: Conception des circuits FPGA

- Architecture d'une matrices de connexion (réseau de routage) entre les CLB

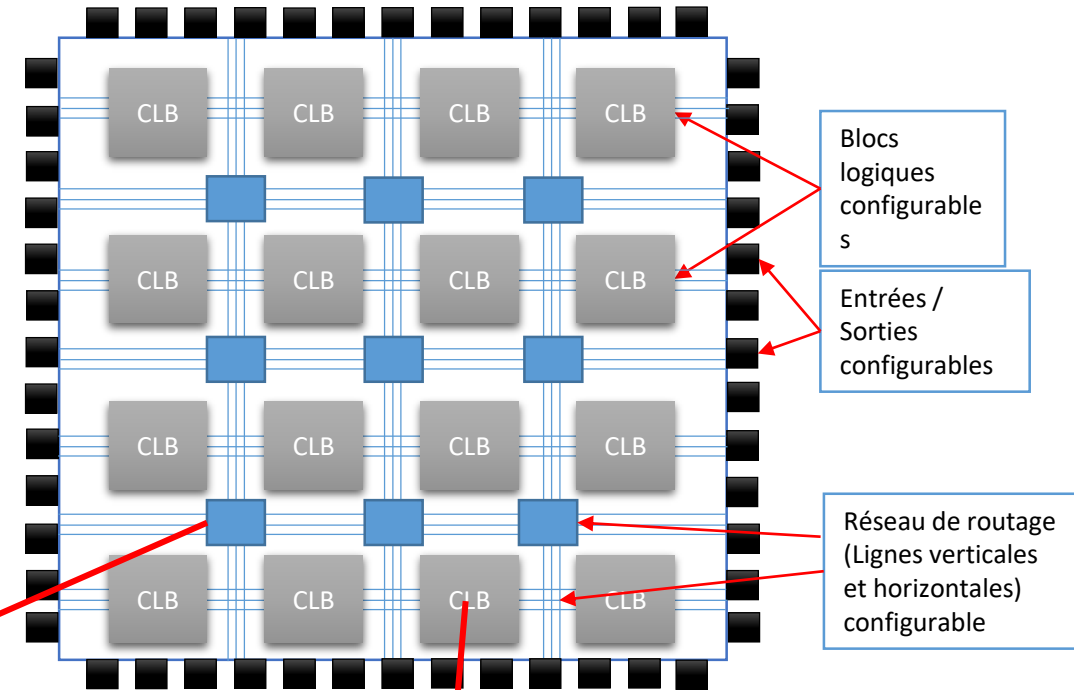
Les fonctions implantées dans les circuits FPGA vont de la simple porte logique jusqu'à une structure très complexe nécessitant d'interconnecter plusieurs blocs logiques entre eux. Les interconnexions utilisées dans les circuits FPGA se basent sur des portes de transmission



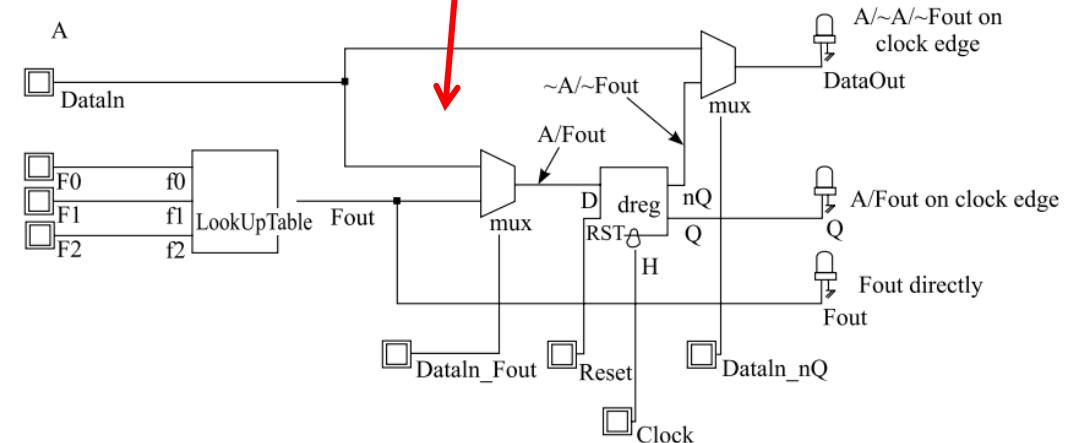
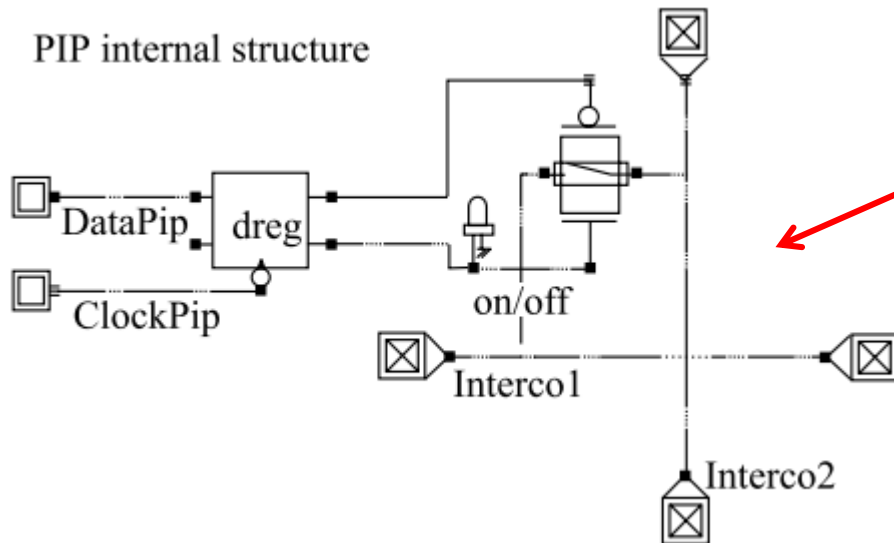
Chapitre 9: Conception des circuits FPGA

- **Architecture d'une matrices de connexion (réseau de routage) entre les CLB**

- Les interconnexions sont aussi programmables (PIP)
- Une porte de transmission et une bascule sont nécessaires pour interconnecter une seule ligne avec une colonne comme le montre la figure suivante.
- En fonction de l'état de la bascule (0 ou 1) on peut soit connecter la ligne soit la déconnecter de la colonne

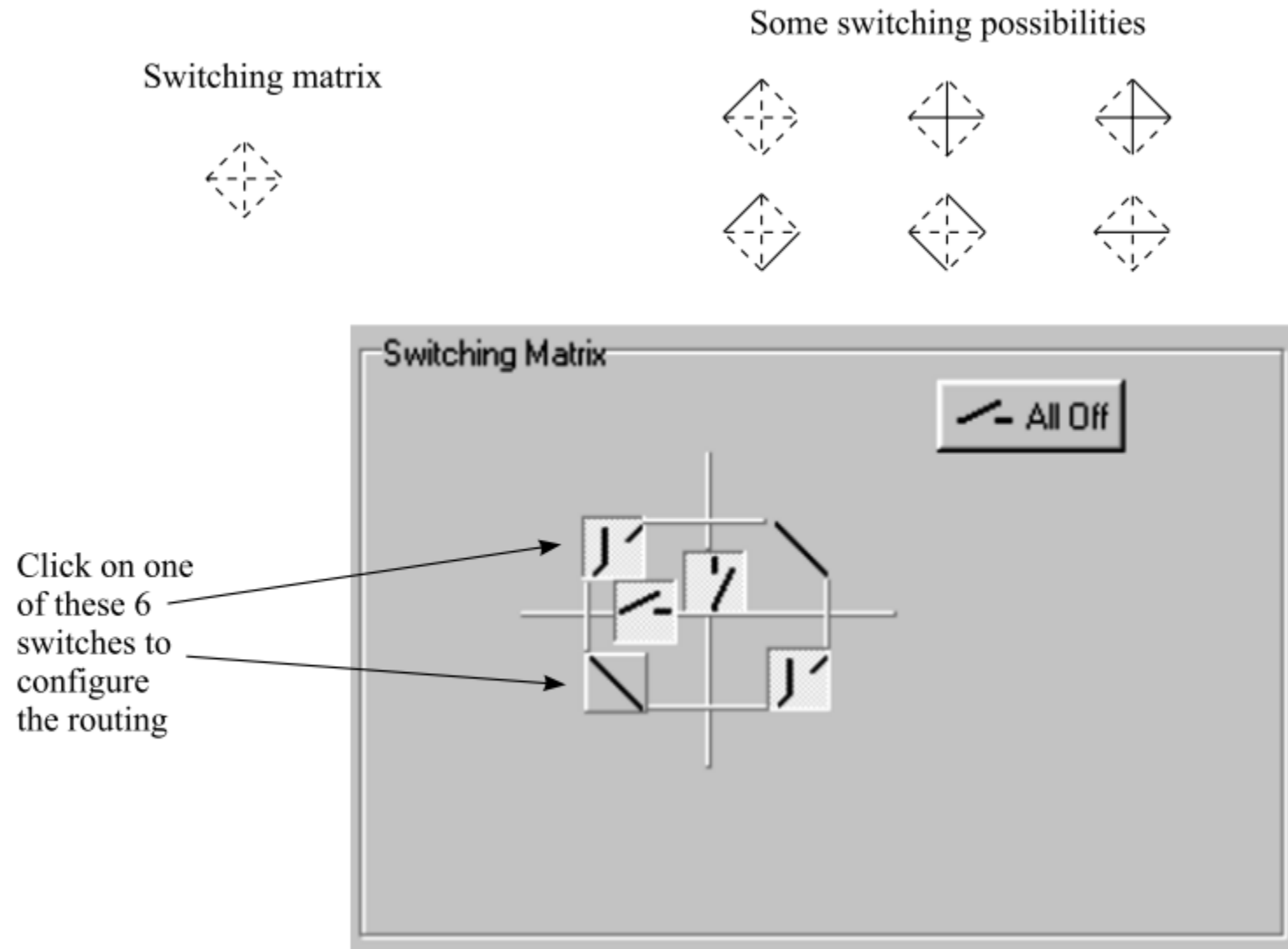


PIP internal structure



Chapitre 9: Conception des circuits FPGA

- **Architecture d'une matrices de connexion (réseau de routage) entre les CLB**
- Les matrices d'interconnexions constituent des points d'interconnexions sophistiqués
- Sa structure est basée sur 6 portes de transmission et 6 bascules D comme le montre la figure suivante.



Chapitre 9: Conception des circuits FPGA

- Du code VHDL vers le circuit FPGA

- La synthèse du code suivant se fait de la manière suivante:

1. Traduction du code VHDL/VERILOG en schéma logique (niveau RTL Register Transfert Level) (**la synthèse**)

2.)

```
Process (clk , reset)
```

```
Begin if (reset <= '1') then
```

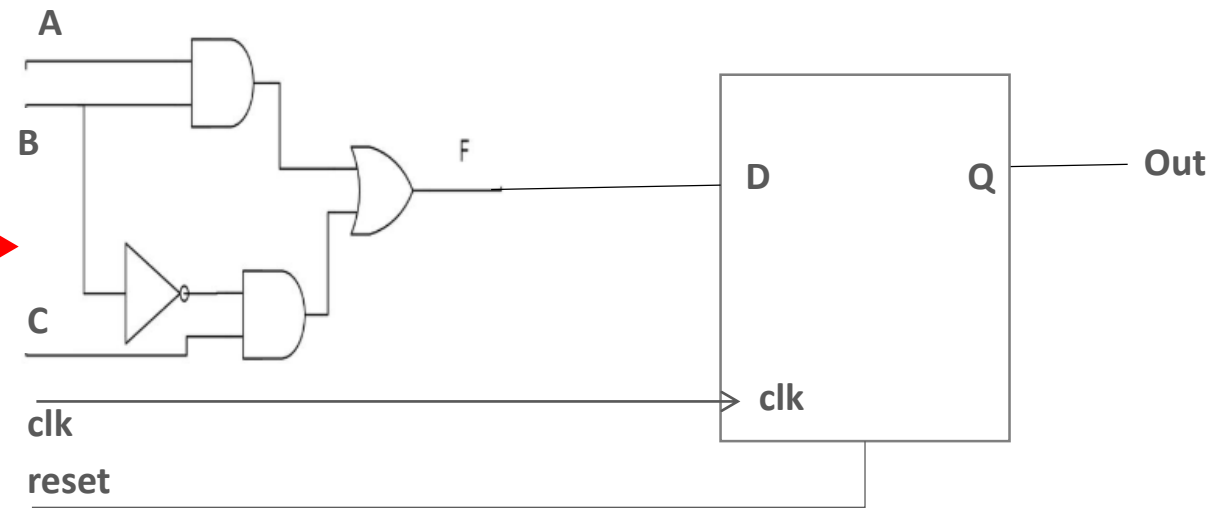
```
Out <= '0';
```

```
Else if clk 'event and clk = ' 1'  
then
```

```
Out <= A and B or not B and C;
```

```
End if;
```

```
End process;
```

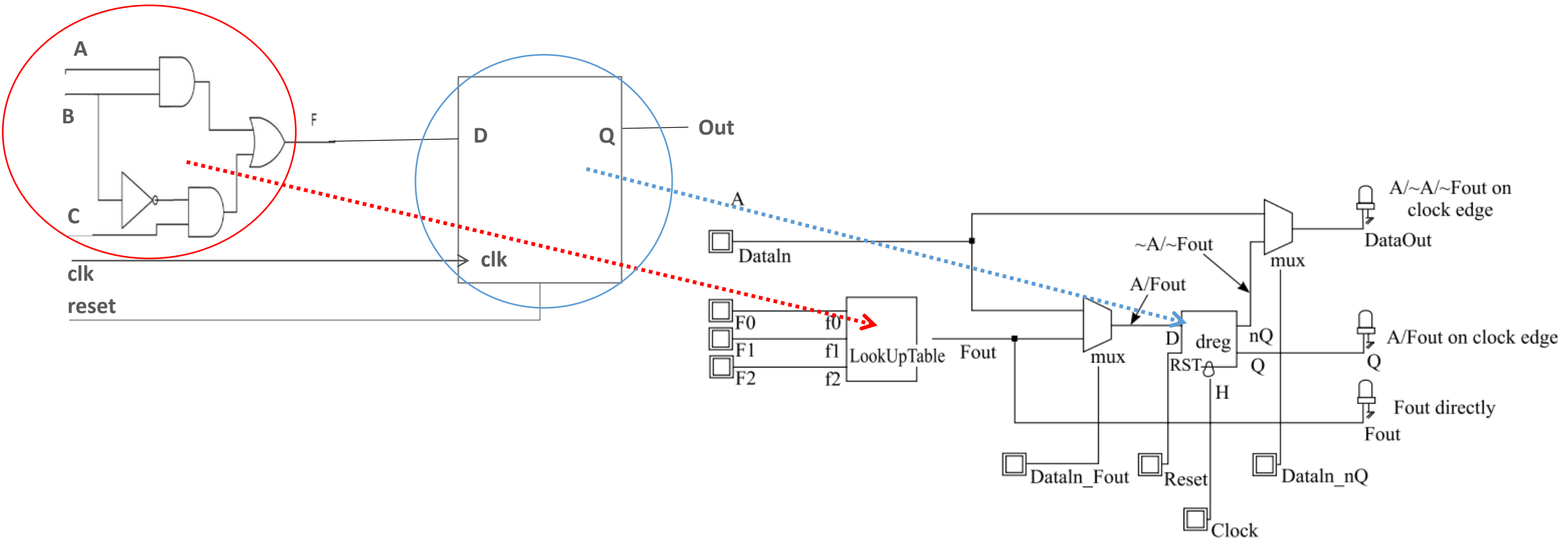


Chapitre 9: Conception des circuits FPGA

- Du code VHDL vers le circuit FPGA

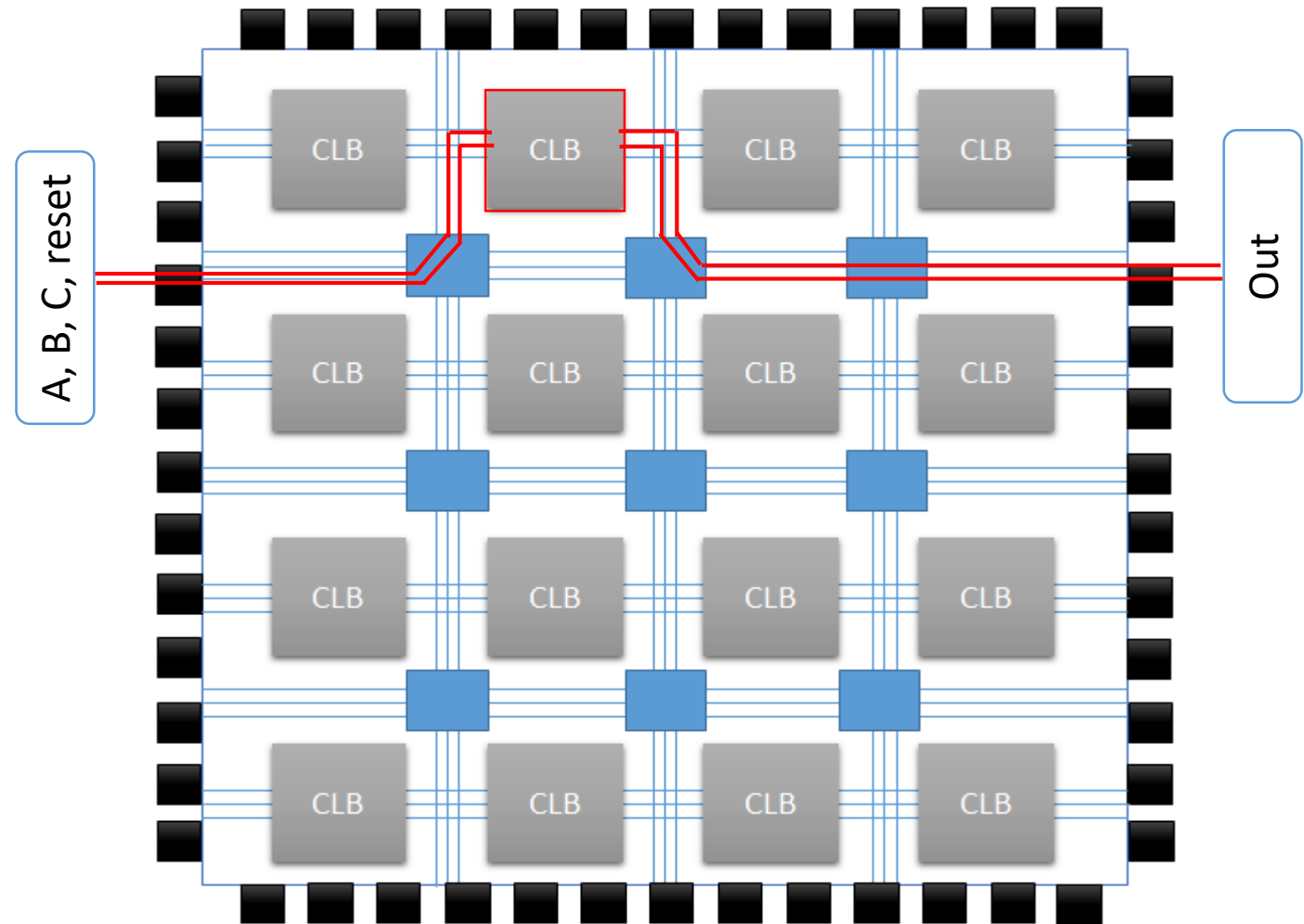
- La synthèse du code suivant se fait de la manière suivante:

1. Traduction du code VHDL/VERILOG en schéma logique (niveau RTL Register Transfer Level) (**la synthèse**)
2. Choix des CLB et implémentation des sous fonctions (**Map & Place**)



Chapitre 9: Conception des circuits FPGA

- Du code VHDL vers le circuit FPGA
- La synthèse du code suivant se fait de la manière suivante:
 1. Traduction du code VHDL/VERILOG en schéma logique (niveau RTL Register Transfer Level) (**la synthèse**)
 2. Choix des CLB et implémentation des sous fonctions (**Map & Place**)
 3. Configuration des matrices de switching et interconnexion des I/O (**Route**)



Chapitre 9: Conception des circuits FPGA

- **Circuit FPGA de la famille XILINX XC5200**
- Le tableau suivant illustre les caractéristiques des circuits FPGA de la famille XC5200

Table 1: XC5200 Field-Programmable Gate Array Family Members

Device	XC5202	XC5204	XC5206	XC5210	XC5215
Logic Cells	256	480	784	1,296	1,936
Max Logic Gates	3,000	6,000	10,000	16,000	23,000
Typical Gate Range	2,000 - 3,000	4,000 - 6,000	6,000 - 10,000	10,000 - 16,000	15,000 - 23,000
VersaBlock Array	8 x 8	10 x 12	14 x 14	18 x 18	22 x 22
CLBs	64	120	196	324	484
Flip-Flops	256	480	784	1,296	1,936
I/Os	84	124	148	196	244
TBUFs per Longline	10	14	16	20	24

Chapitre 9: Conception des circuits FPGA

- **Circuit FPGA de la famille XILINX XC5200**
- L'architecture du circuit XC5200 est constituée de:
- Des pins d'entrées/sorties programmables
- Des blocs logiques configurables (Versa-Block)
- Des matrices d'interconnexions programmables (GRM: General Routing Matrix)
- Des interconnexions programmables constituant une interface entre les pins d'entrées/sorties du circuit et les blocs logiques configurables (VersaRing)

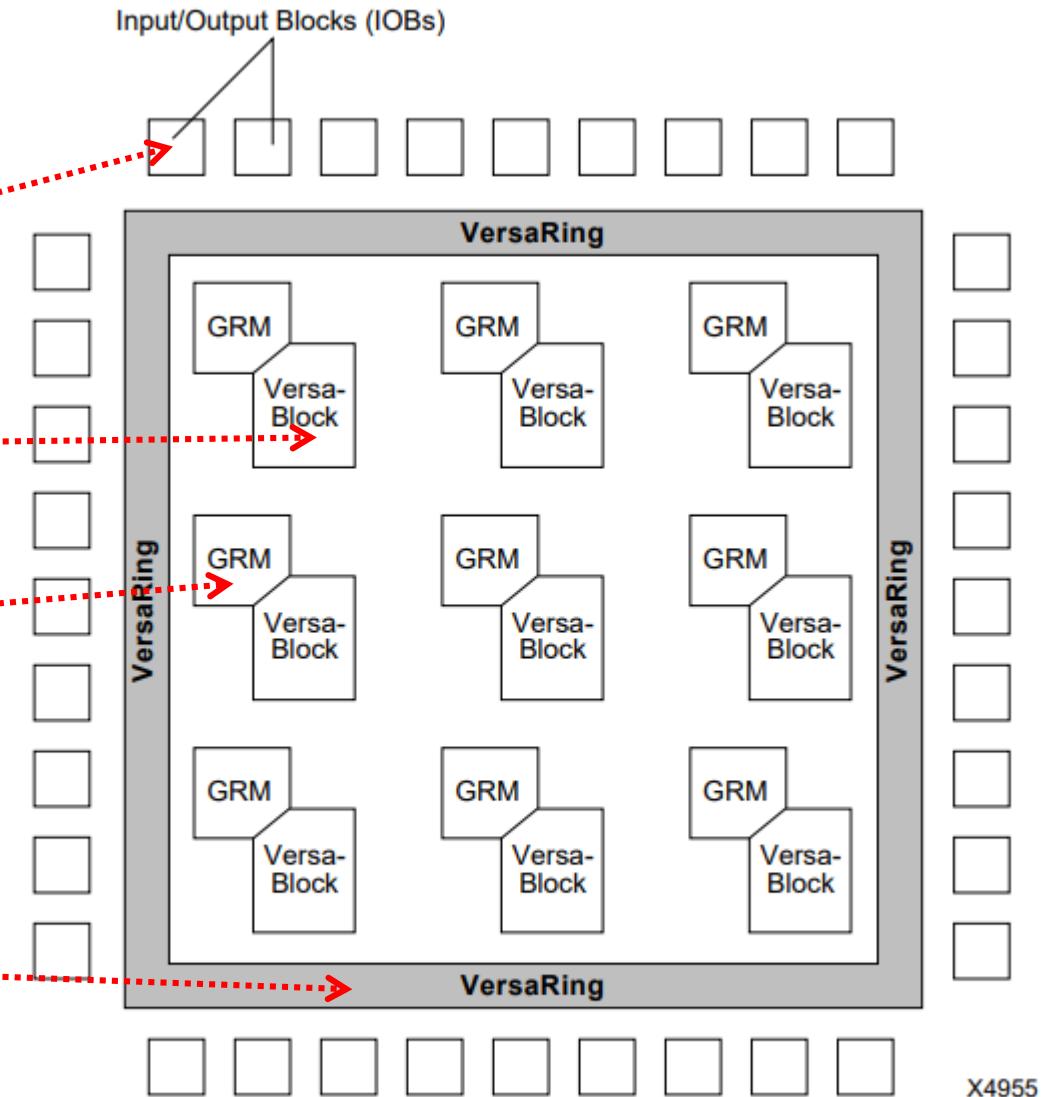


Figure 1: XC5200 Architectural Overview

Chapitre 9: Conception des circuits FPGA

- **Circuit FPGA de la famille XILINX XC5200**
- Chaque bloc configurable VersaBloc est constitué de quatre cellules logiques (LC : Logic Cell)
- Chaque LC est constituée d'une LUT à 4 variables F, de trois Mux 2:1 et d'une bascule de stockage FD
- Chaque LC a 5 entrées et trois sortie indépendantes
- d'autres signaux logiques de contrôle CI, CO (Carry In, Carry Out, Clock Enable, Clear)

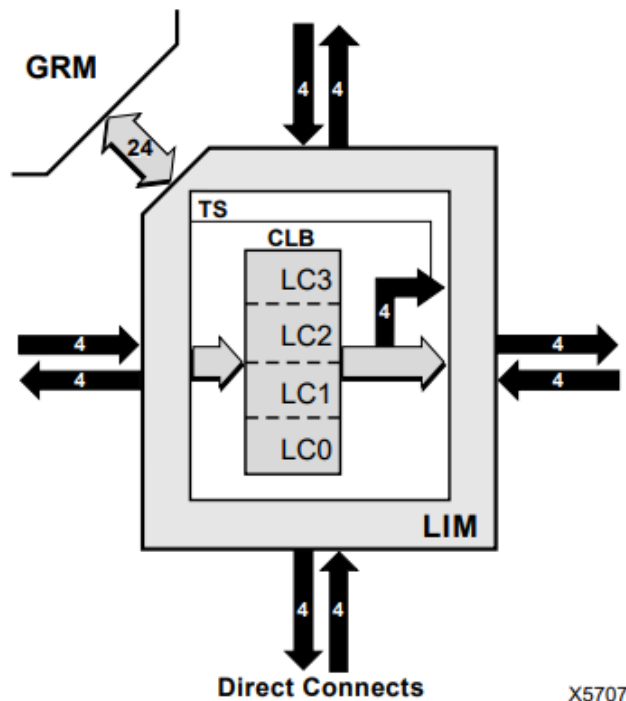


Figure 2: VersaBlock

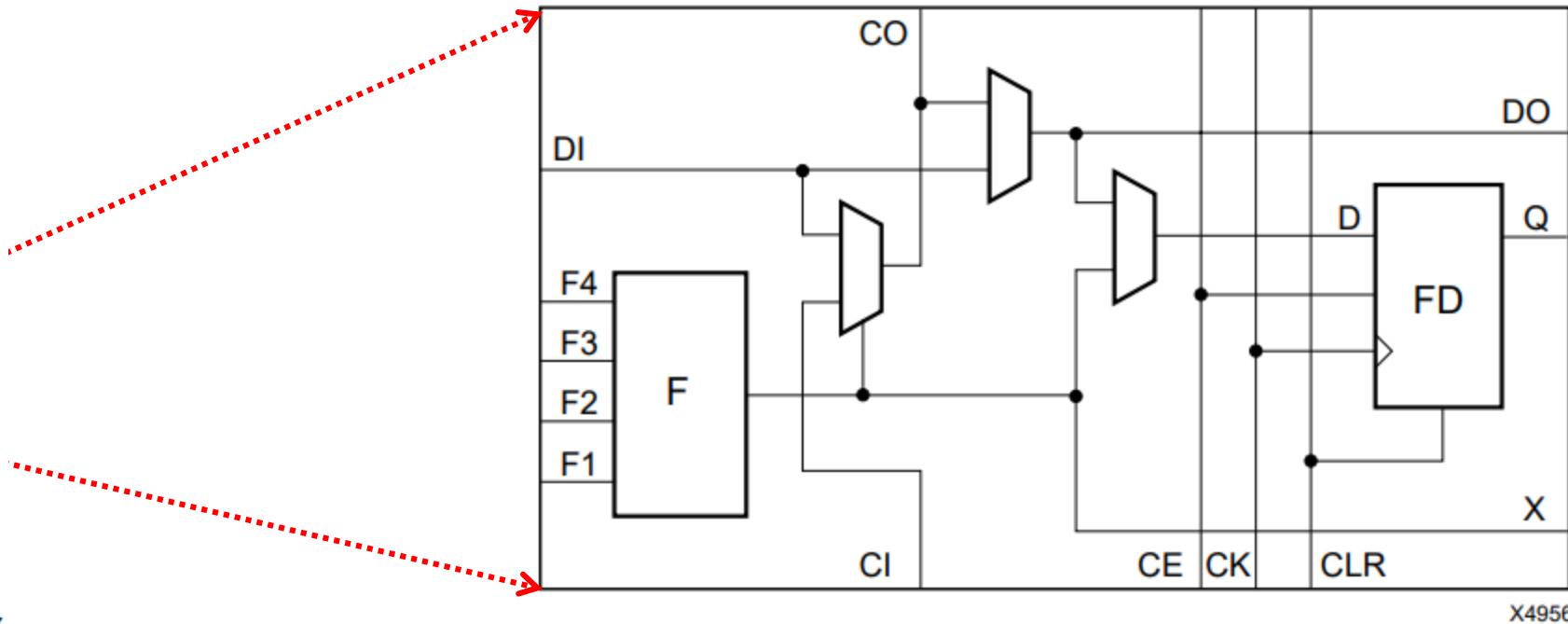


Figure 3: XC5200 Logic Cell (Four LCs per CLB)

Chapitre 9: Conception des circuits FPGA

- **Circuit FPGA de la famille XILINX XC5200**
- Chaque bloc logique configurable CLB du bloc VersaBloc est constitué de quatre cellules logiques (LC : Logic Cell)
- Chaque CLB de 4 LUT de 4 variables et quatre Bascules D
- Chaque CLB a 20 entrées indépendantes et 12 sorties indépendantes

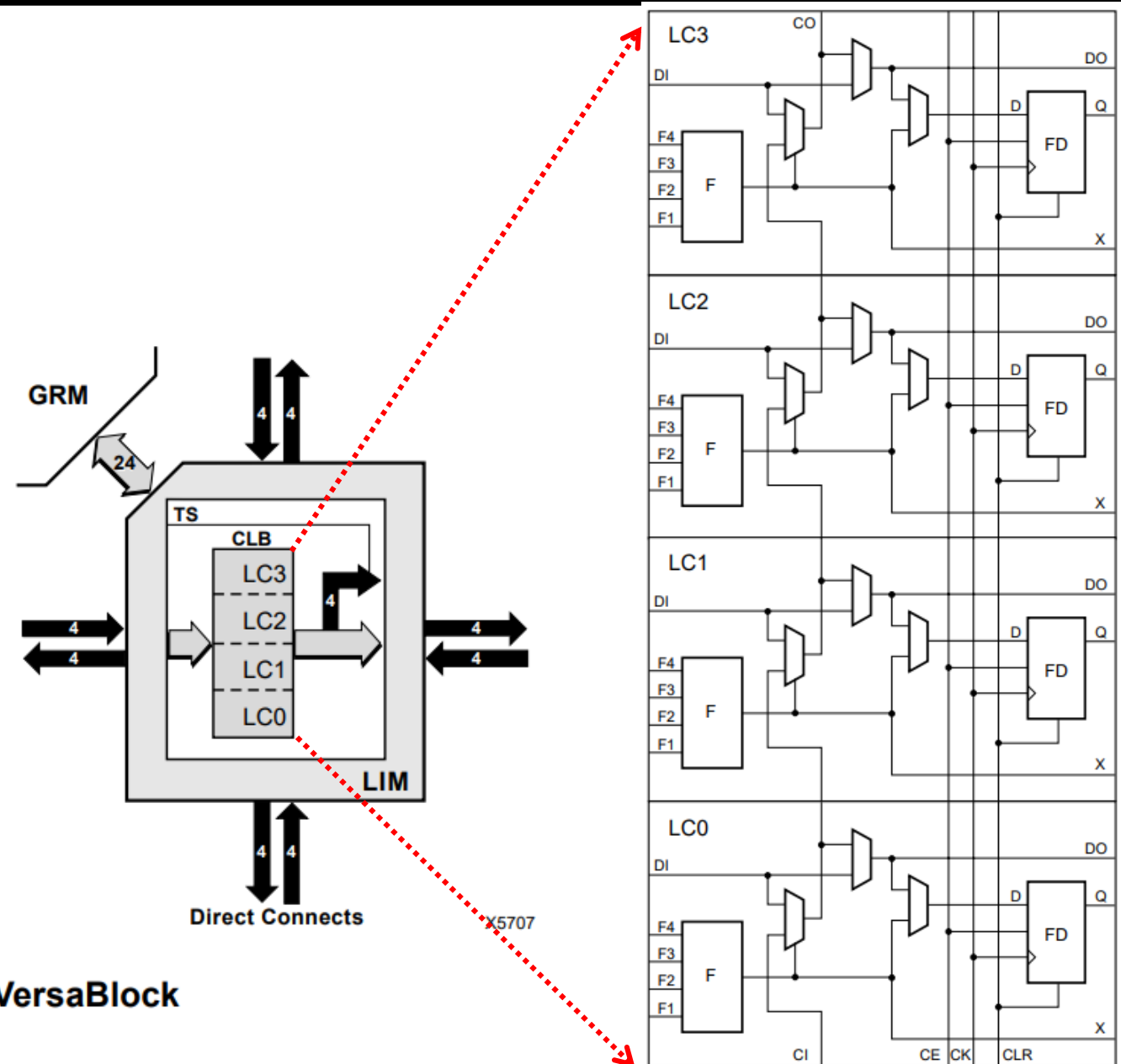
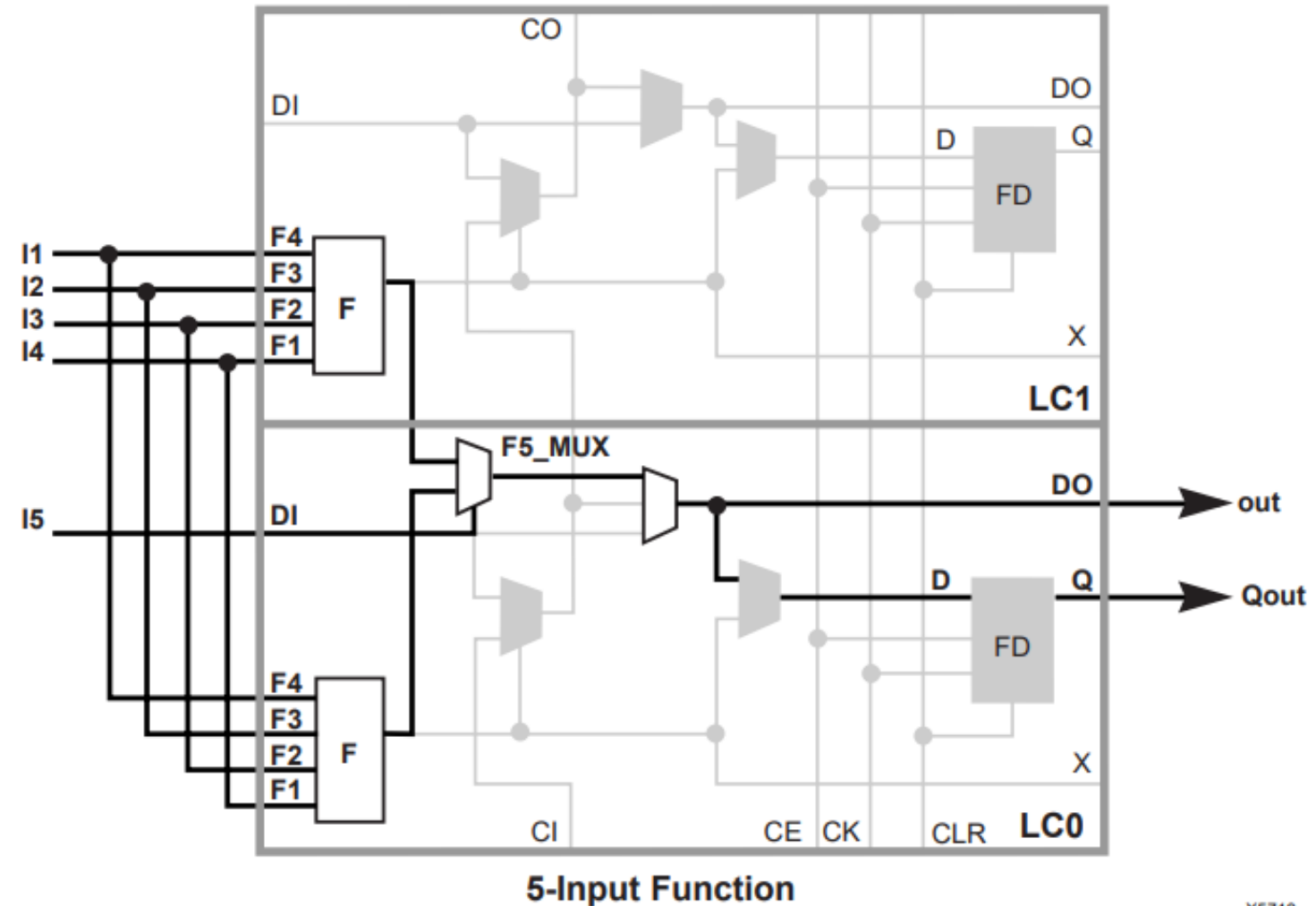


Figure 2: VersaBlock

Chapitre 9: Conception des circuits FPGA

- **Circuit FPGA de la famille XILINX XC5200**
- Avec ce type de CLB on peut par exemple implémenter des fonctions à 5 variables
- Deux LUT à quatre variables peuvent être utilisées comme le montre la figure suivante
- Les entrées I1...I4 sont communes pour les deux LUT
- La cinquième entrée I5 agit comme entrées sélection du F5_MUX
- La sortie peut être soit stockée dans la bascule FD soit connectée directement vers la sortie du CLB



X5710

Chapitre 9: Conception des circuits FPGA

- **Circuit FPGA de la famille XILINX XC5200**
- Chaque CLB contient quatre **inverseurs à trois états** contrôlés par un signal TS (Three State)
- La sortie de chaque inverseurs à trois états est connecté vers deux lignes horizontales et deux autres verticales
- **Le bloc LIM (Local Interconnect Matrix)** est constitué de 8 lignes horizontales et 8 autres verticales assurant les interconnexions nécessaires au sein d'un CLB

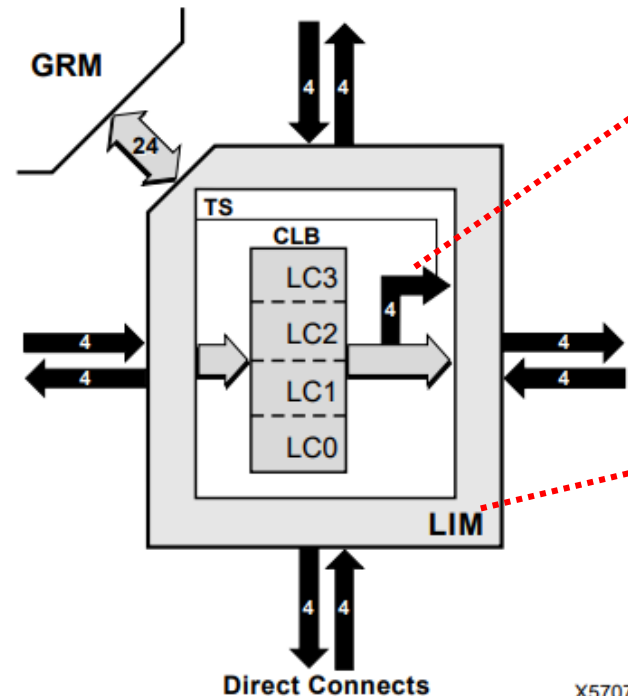


Figure 2: VersaBlock

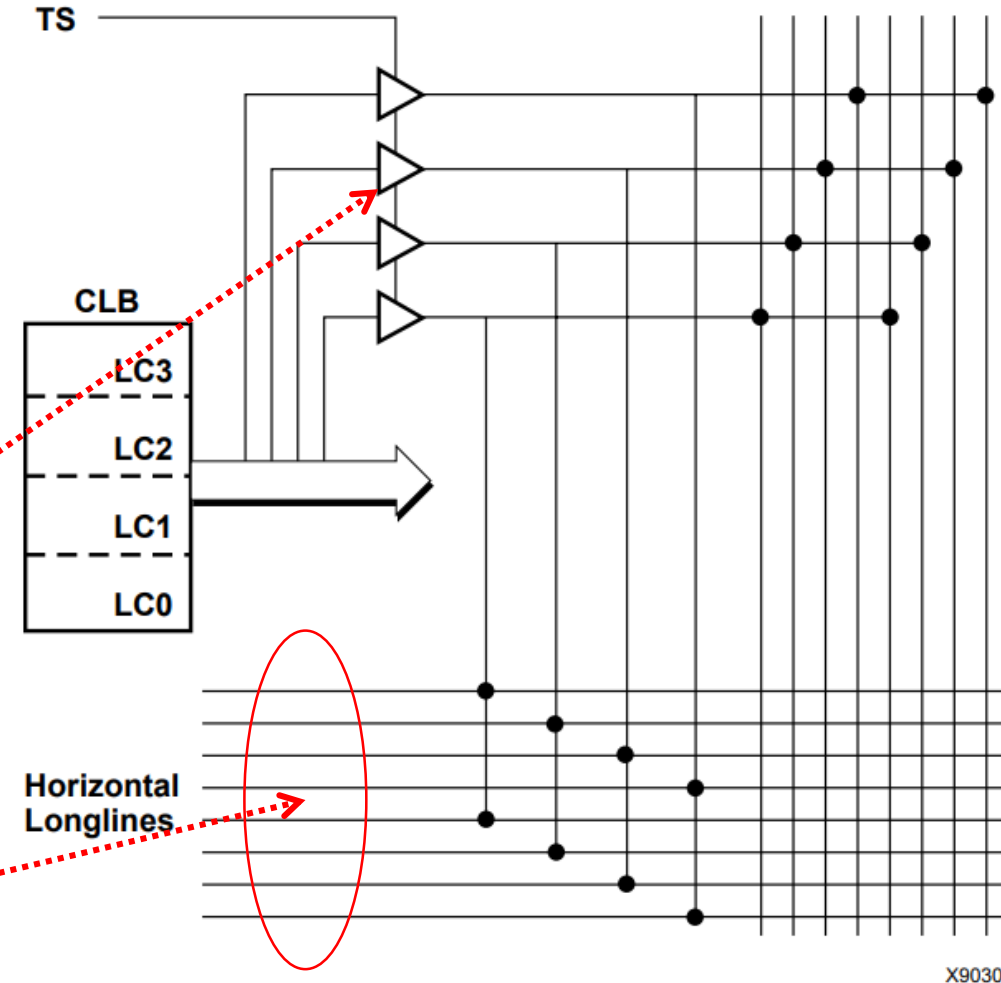
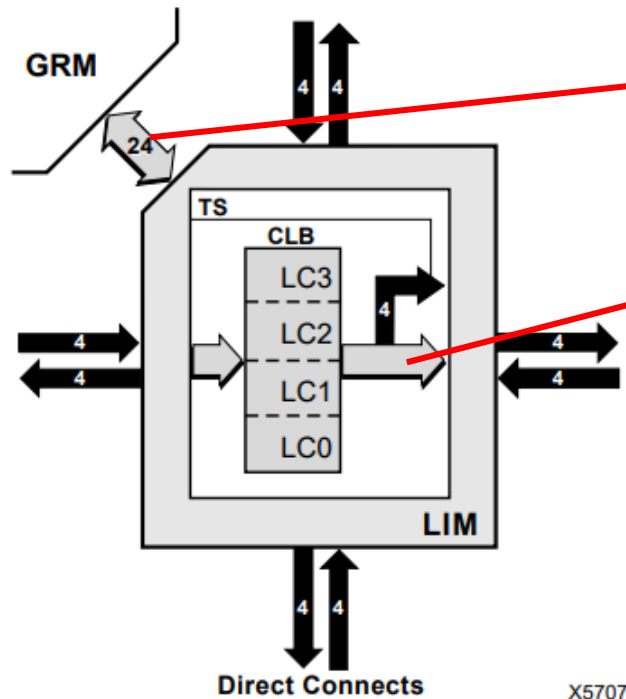


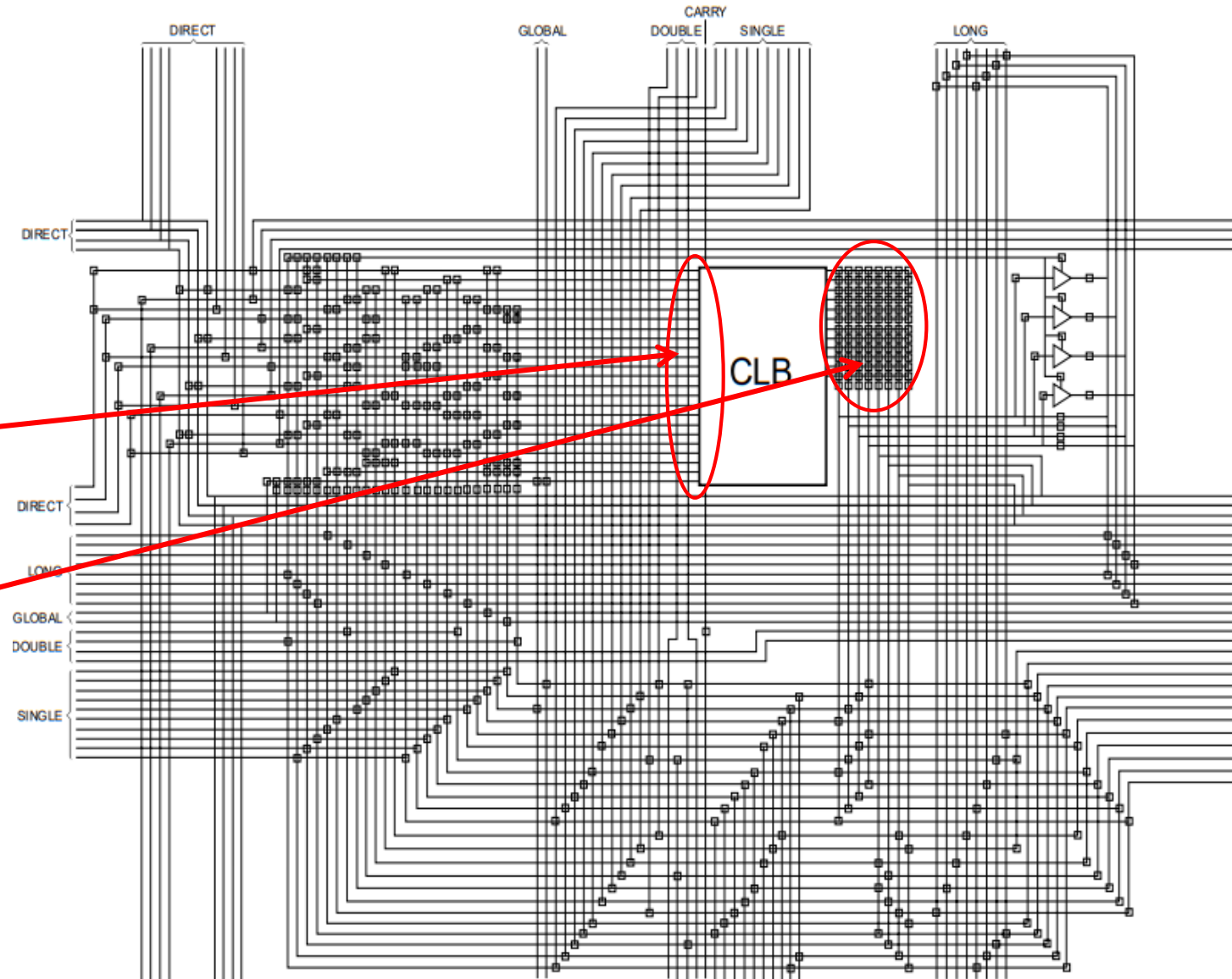
Figure 9: XC5200 3-State Buffers

Chapitre 9: Conception des circuits FPGA

- **Circuit FPGA de la famille XILINX XC5200**
- La figure suivante illustre les interconnexions associées à un seul bloc logique configurable CLB



X5707

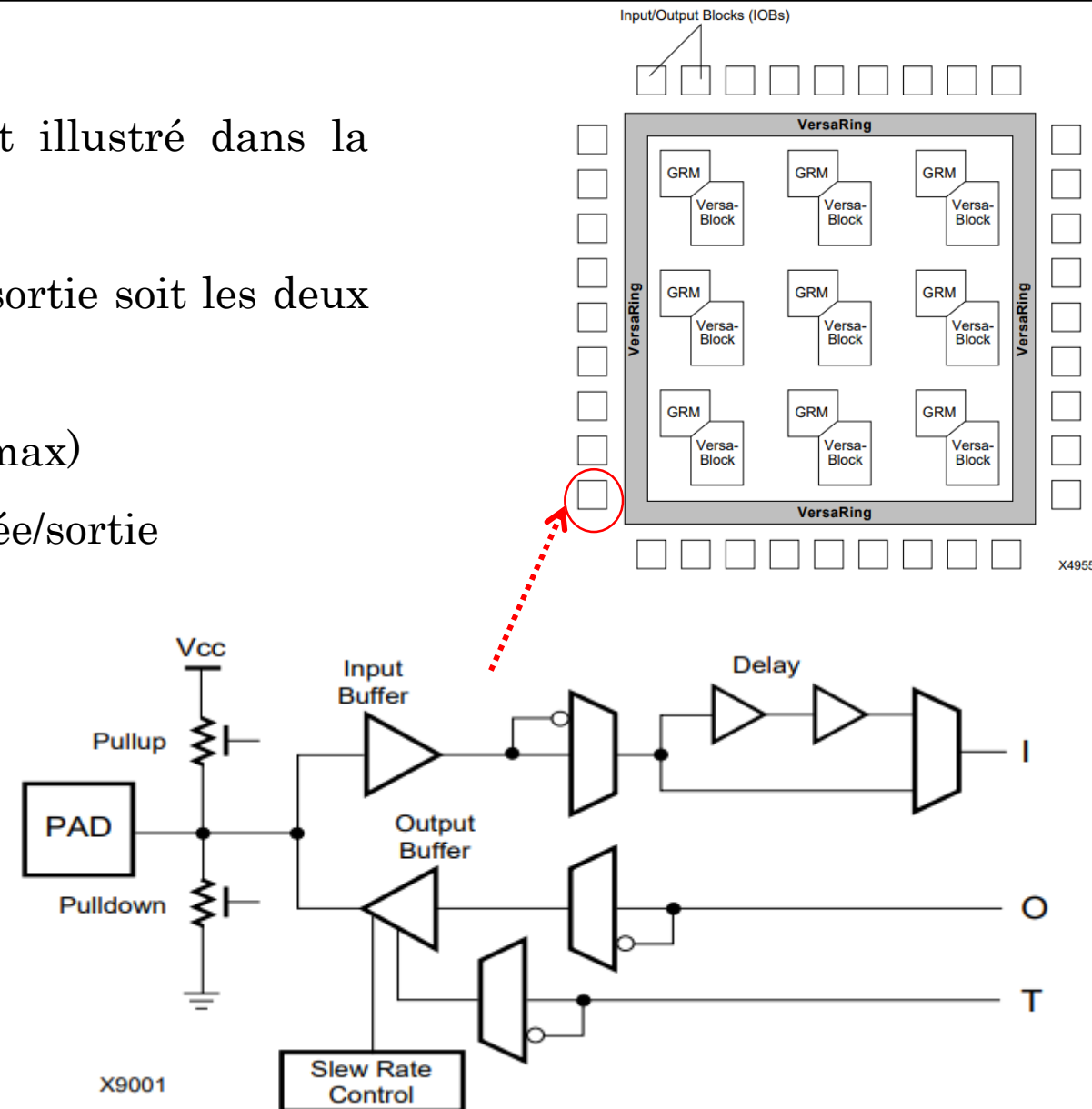


x9010

Figure 2: VersaBlock

Chapitre 9: Conception des circuits FPGA

- **Circuit FPGA de la famille XILINX XC5200**
- Le schéma basique de chaque port d'entrée/sortie est illustré dans la figure suivante.
- Chaque Port peut être configuré soit en entrée soit en sortie soit les deux (port bidirectionnel)
- Chaque port peut fournir un courant de sortie de **8mA** (max)
- Le bloc offre la possibilité d'inverser ou non chaque entrée/sortie
- Un délais optionnel peut être aussi appliqué dans les deux cas entrée / sortie
- Le port peut aussi être relié soit vers Vcc soit Vss à travers des résistances de Pullup/ Pulldown.
- Un autre signal T permet de mettre la sortie en état haute impédance



Chapitre 9: Conception des circuits FPGA

- **Circuit FPGA de la famille XILINX XC5200**

- Le bloc VersaRing est constitué de matrices d'interconnexion permettant d'assurer la connexion entre les ports I/O et la logique interne du circuit FPGA
- Chaque I/O bloc peut être connecté soit directement vers le bloc CLB le plus proche soit vers sa matrice GRM associée soit vers un autre CLB

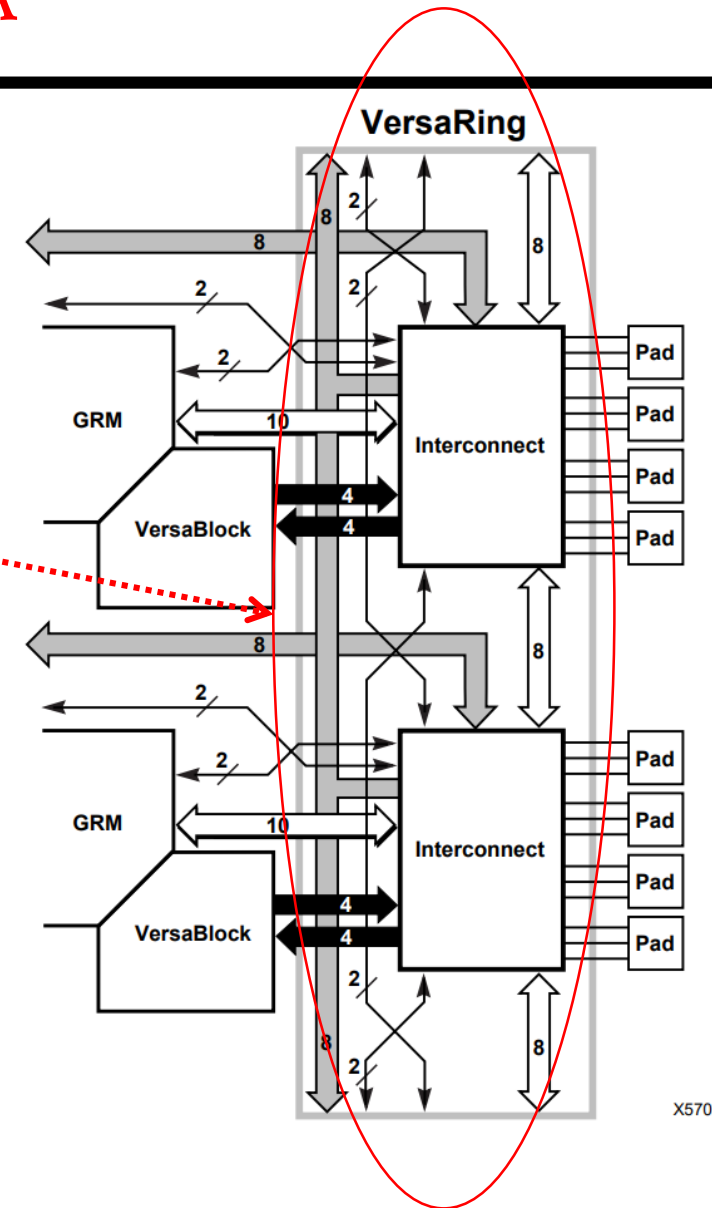
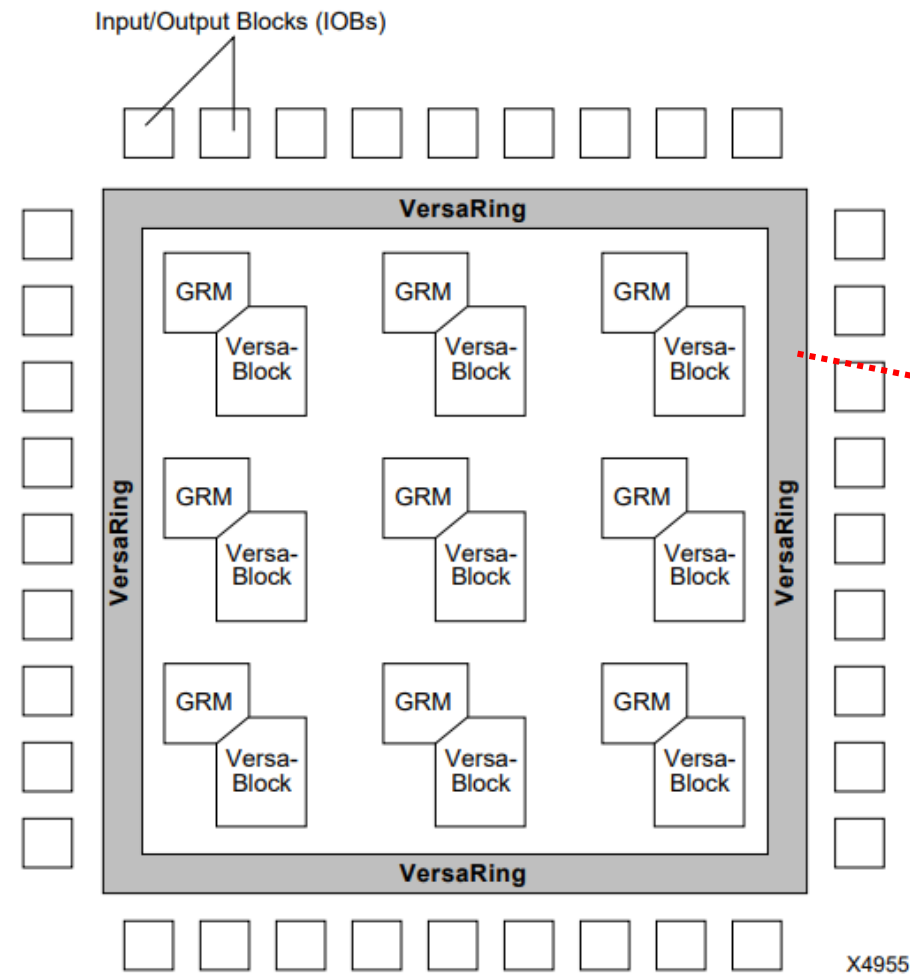
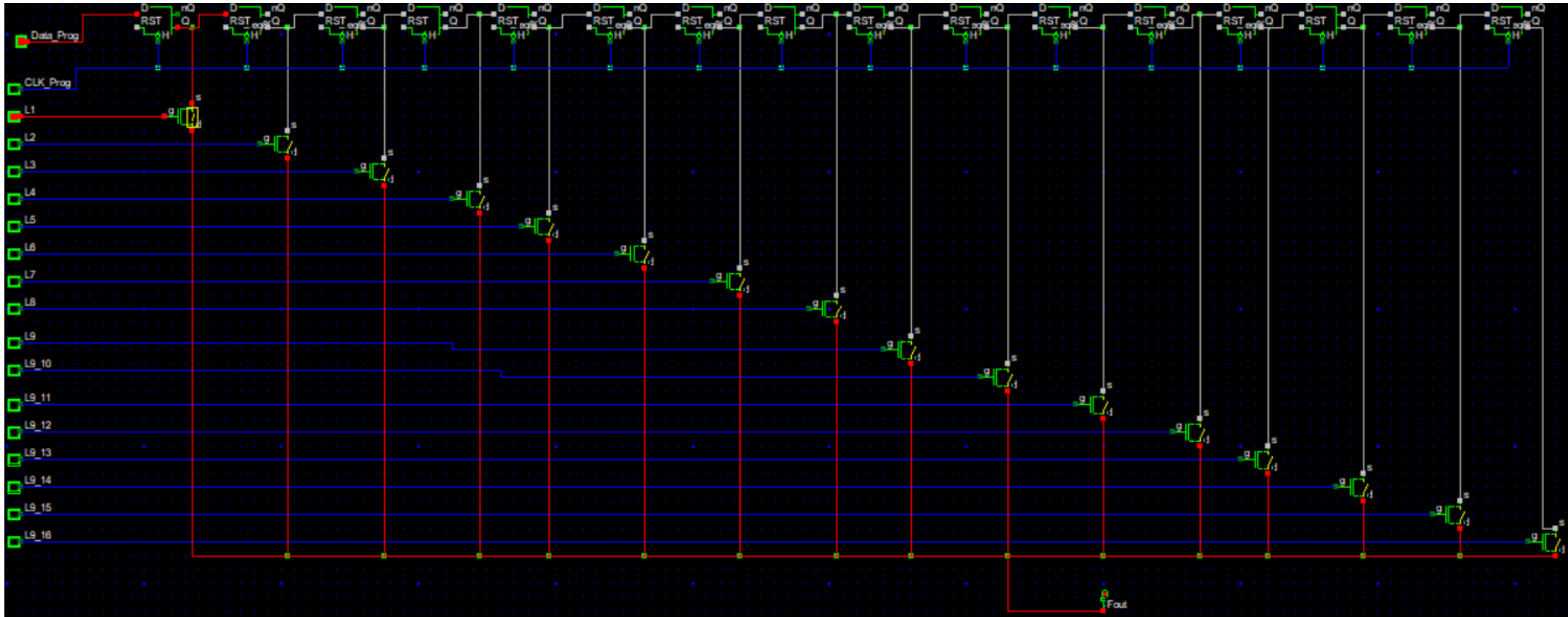


Figure 1: XC5200 Architectural Overview

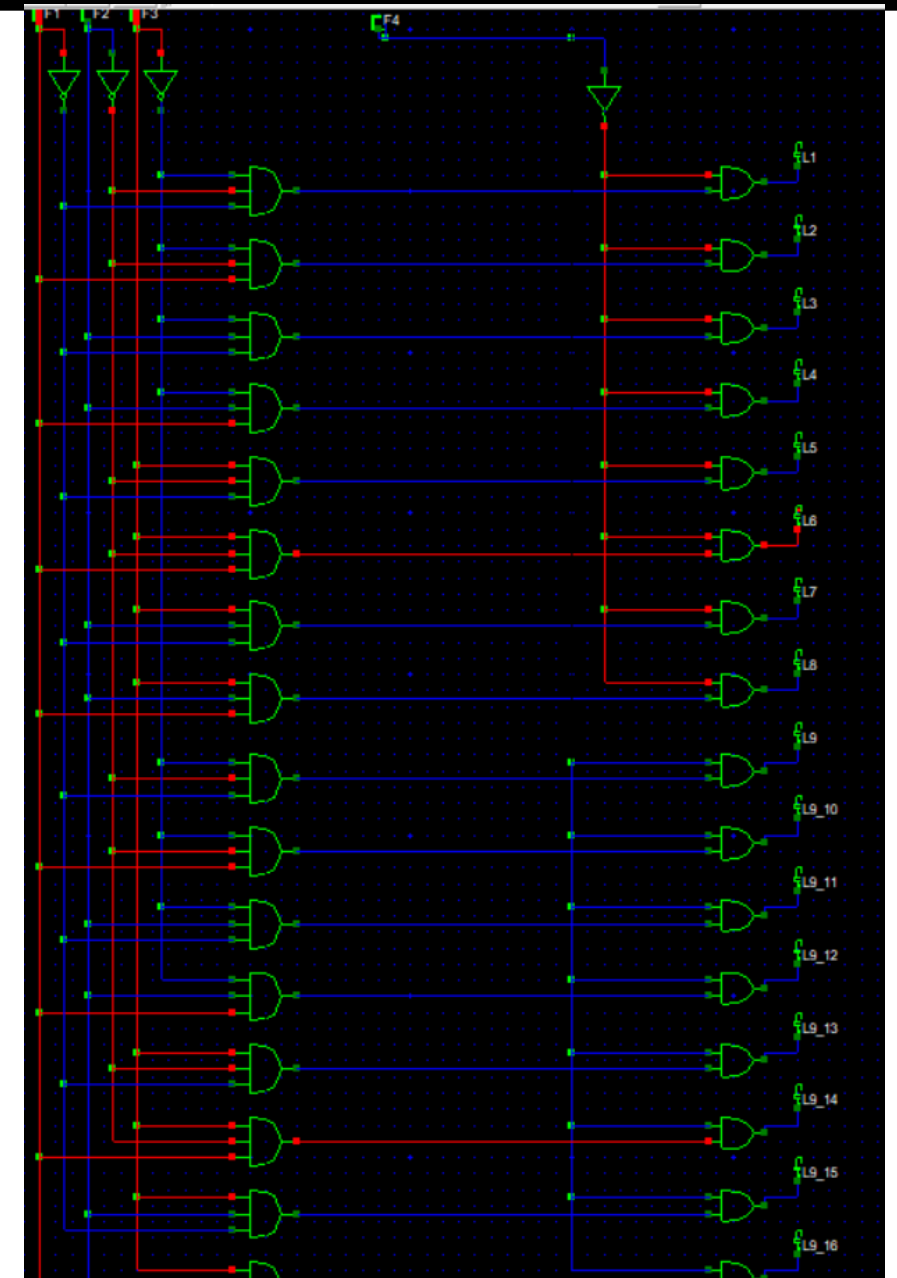
Chapitre 9: Conception des circuits FPGA

- Réaliser un LUT à 4 variables à base de registre à décalage comme le montre la figure suivante
- Simuler et vérifier son fonctionnement



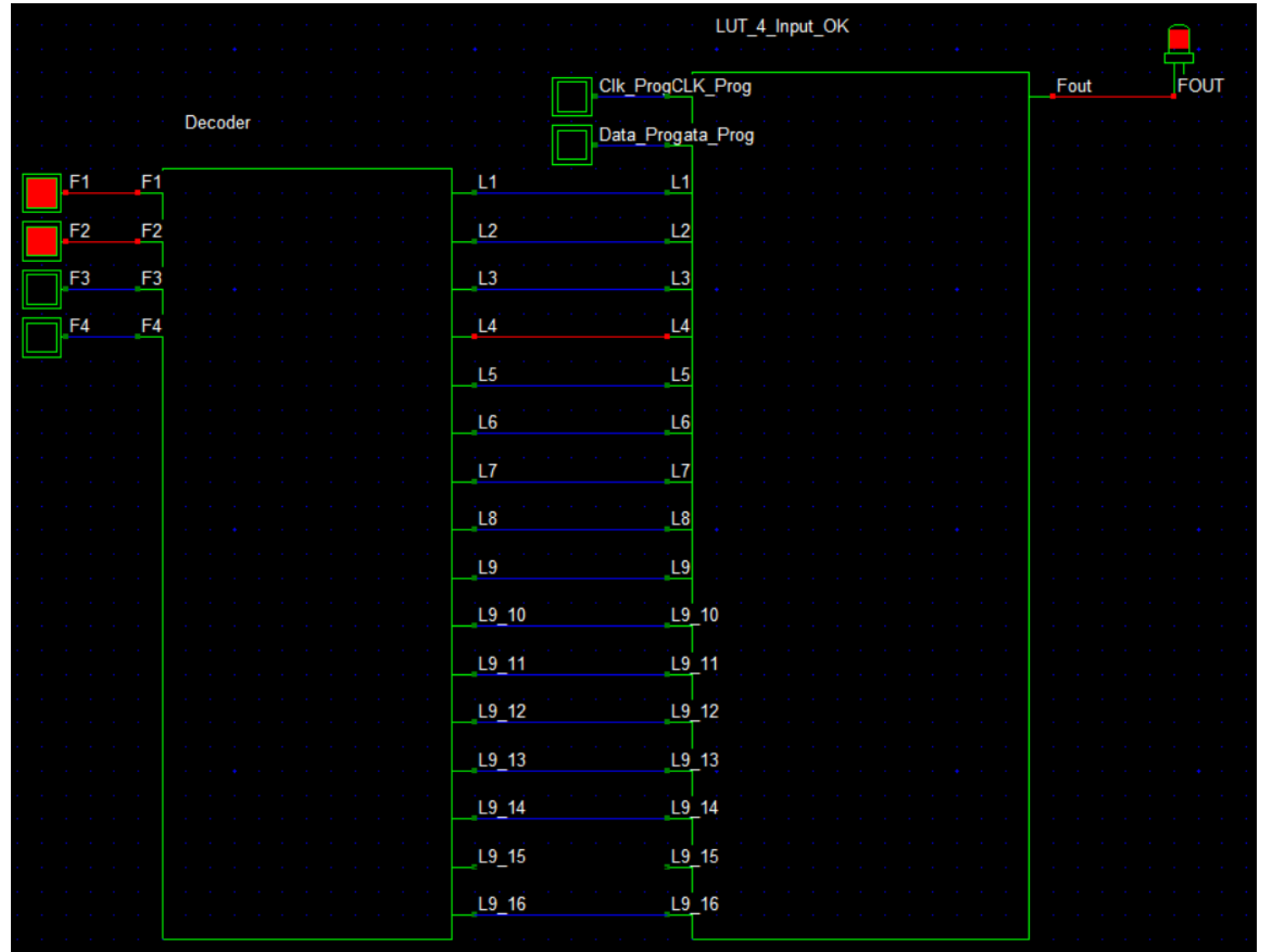
Chapitre 9: Conception des circuits FPGA

- Réaliser le décodeur 4 vers 16
comme le montre la figure
suivante



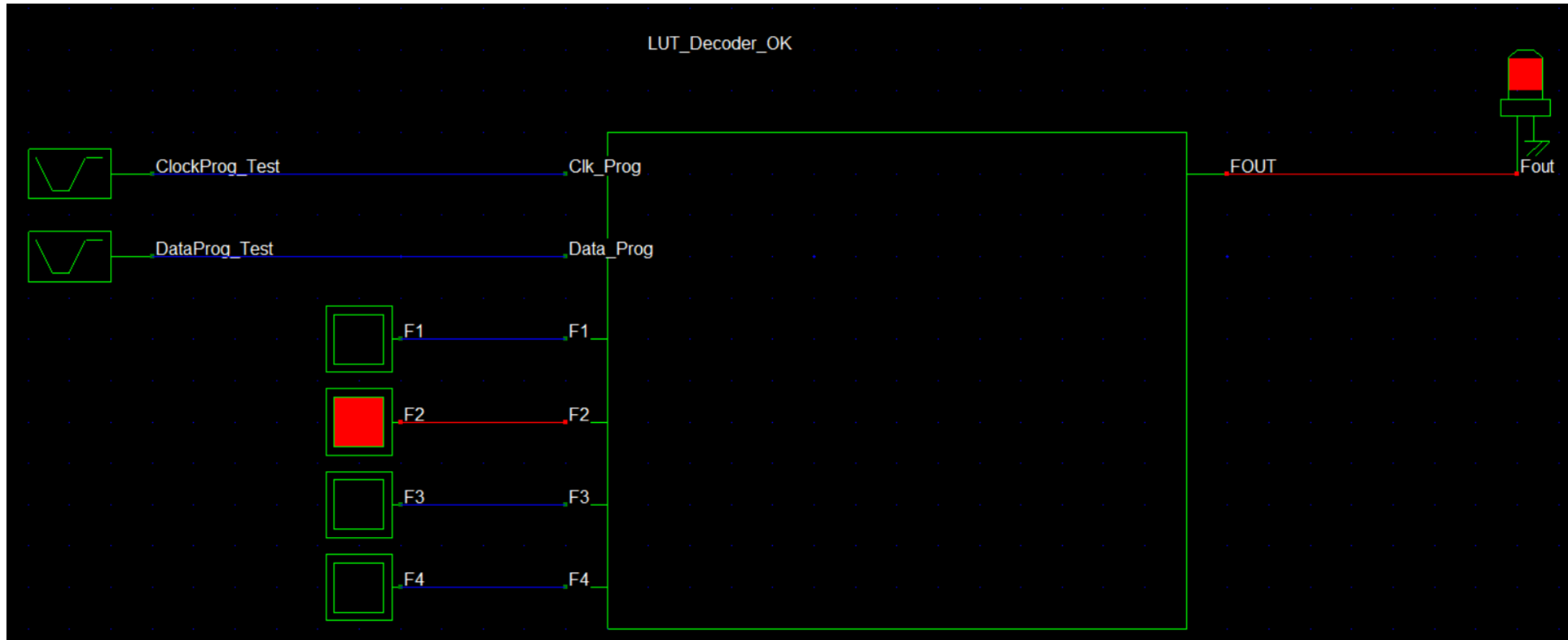
Chapitre 9: Conception des circuits FPGA

- Combiner le décodeur et la LUT et vérifier son fonctionnement comme le montre la figure suivante



Chapitre 9: Conception des circuits FPGA

- Générer le symbole du circuit LUT et vérifier son fonctionnement
- Utiliser les générateur de DATA et de l'horloge pour la configuration du circuit



Chapitre 9: Conception des circuits FPGA

- Utilisation de Data_Prog et Clock Prog

- Les bascules sont actives sur front descendant, donc la donnée sera décalée vers la droite chaque front descendant
- La donnée change quand le signal clk est inactif

Name : ClockProg_Test

Time(ns)	Value(1/0)
1.00	0
2.00	1
3.00	0
4.00	1
5.00	0
6.00	1
7.00	0

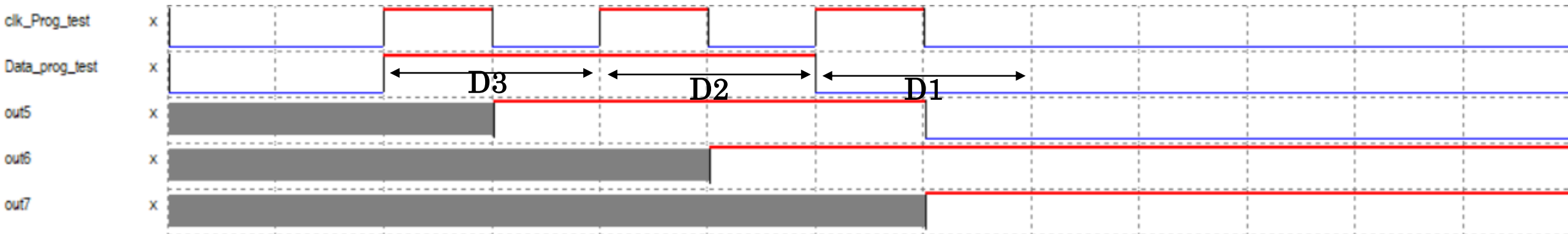
Premier Front descendant

Deuxième Front descendant

Troisième Front descendant

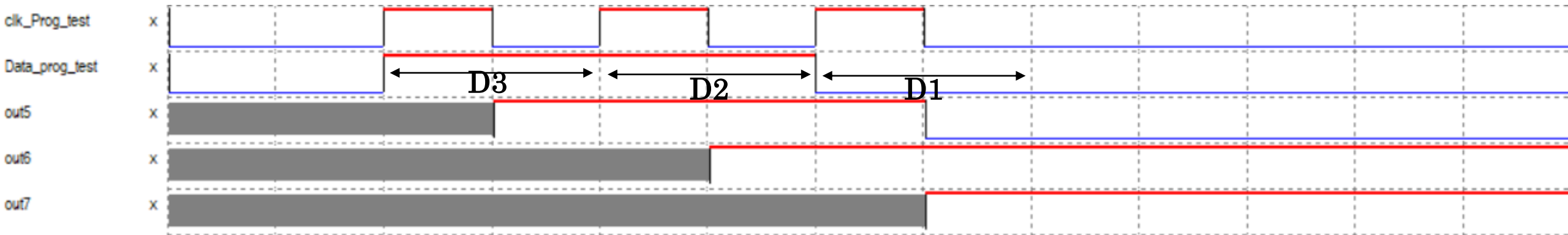
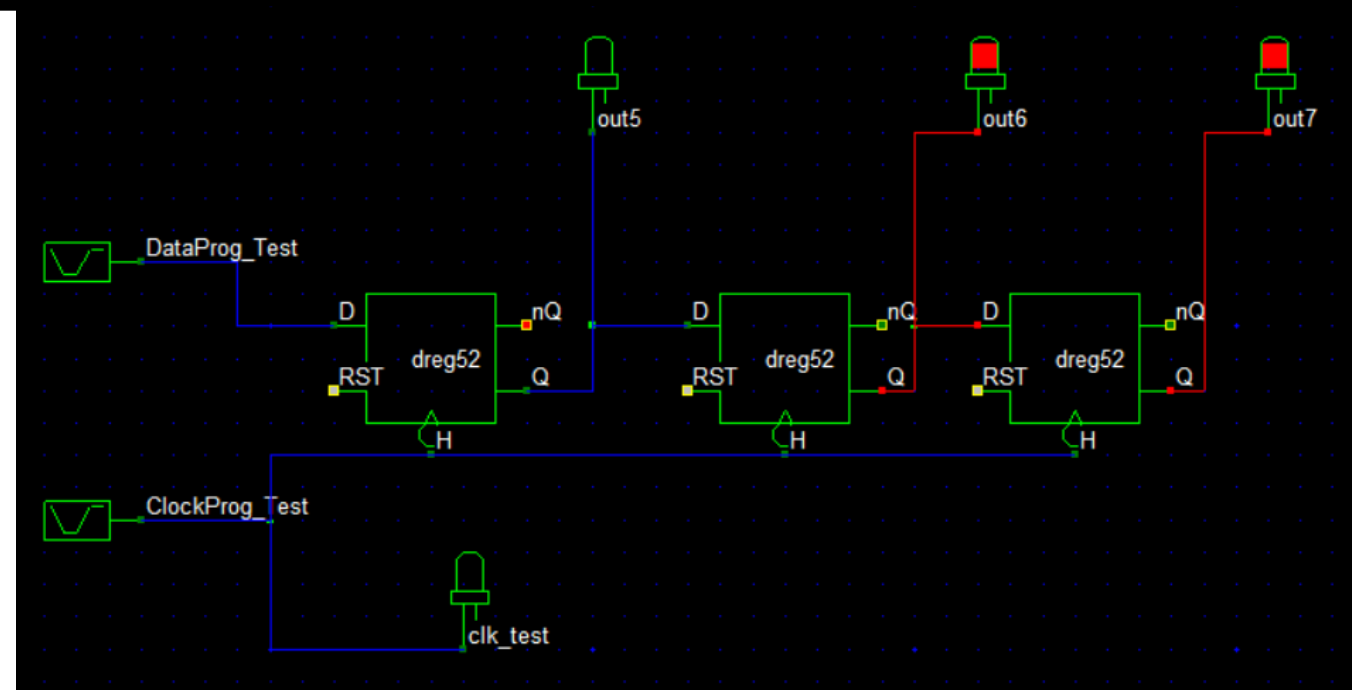
Name : DataProg_Test

Time(ns)	Value(1/0)	
1.00	0	
2.00	1	D3
3.00	1	
4.00	1	D2
5.00	1	
6.00	0	
7.00	0	D1



Chapitre 9: Conception des circuits FPGA

- Utilisation de Data_Prog et Clock Prog
- Utilisation de **Data_Prog** et **Clock Prog**
- Les bascules sont actives sur front descendant, donc la donnée sera décalée vers la droite chaque front descendant
- La donnée change quand le signal clk est inactif



Chapitre 9: Conception des circuits FPGA

- Réalisation du Logic Cell du XC5200
- Réaliser le circuit LC0 de la famille XC5200

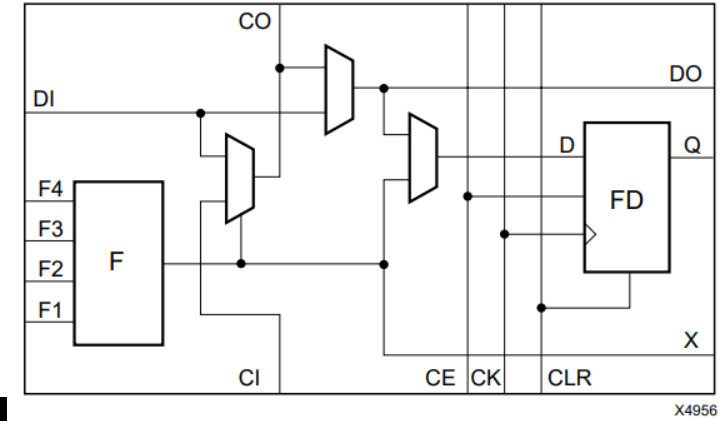


Figure 3: XC5200 Logic Cell (Four LCs per CLB)

