

LES SYSTEMES LOGIQUES

I. LES CIRCUITS STANDARDS

- Ils donnent les fonctions fixes d'usage courant.
- Il existe trois types de circuits standards :
 - Les circuits à faible degré d'intégration (SSI) (ET, OU, Bascules,...,etc)
 - Les circuits à densité moyenne d'intégration (multiplexeurs, décodeurs, encodeurs, registres a décalage, compteurs, ...,etc
 - Les circuits à grande densité d'intégration (LSI) (mémoires a semi-conducteurs (RAM, ROM, PROM, EPROM, ...,etc)).
- Ces circuits sont caractérisés par le fait qu'ils ne réalisent qu'une seule fonction.

II. LES CIRCUITS SPECIFIQUES : (ASIC : APPLICATION SPECIFIC INTEGRATED CIRCUITS)

II.1. LES RESEAUX LOGIQUES PROGRAMMABLES

- Ils permettent à l'utilisateur de programmer ses propres fonctions (combinatoires ou séquentielles).
- La programmation se fait par fusibles avec des circuits tels que les PAL, PLD, FPLA...etc ou sans fusibles avec des circuits comme les GAL, EPLD...etc. Ces circuits se présentent comme des réseaux d'opérateurs ET-OU ou des bascules associées à des opérateurs ET-OU. Un circuit programmable peut donc substituer quelques boîtiers SSI ou MSI.
- La programmation de ce type de circuits s'effectue à l'aide d'un logiciel spécifique (PALASM, PLDesigner...etc) et d'un appareil adéquat appelé programmeur.

II.2. LES RESEAUX REDIFFUSES (GATE ARRAY) (FPGA OU LCA)

- A l'état vierge, un tel circuit comprend un grand nombre de cellules. Chaque cellule contient soit des portes logiques, soit des transistors et des résistances. Ces portes ou ces éléments ne sont pas interconnectés. La programmation de ce type de circuits revient à assurer la connexion entre ses différents composants.
- Le développement de ce type de circuits nécessite l'utilisation d'un outil d'IAO (Interconnexion Assistée par Ordinateur).

II.3. LES CELLULES PRE CARACTERISEES : (CELL ARRAY OU STANDARD CELLS)

- Dans ce type de circuits il n'y a pas de diffusion préalable d'éléments actifs sur le silicium. La programmation de ces circuits se fait à l'aide d'un outil de CAO qui doit être très puissant.

III. LES RESEAUX LOGIQUES COMBINATOIRES PROGRAMMABLES

Permettent la réalisation pratique d'un système logique dont la fonction est adaptable (programmable) par l'utilisateur.

On distingue trois principaux types de fonctions à circuits programmables :

III.1. STRUCTURE DES RESEAUX LOGIQUES COMBINATOIRES

La plupart de ces composants sont constitués d'une *matrice d'opérateurs ET* qui génère les produits des variables d'entrées et de leurs compléments et d'une *matrice d'opérateurs OU* qui somme les produits. Suivant le type de circuits, l'une ou l'autre ou les deux matrices sont programmables.

☞ Toute fonction logique se représente sous forme d'une expression de **somme de produits** ou de **produit sommes**.

III.2. CLASSIFICATION DES RESEAUX LOGIQUES COMBINATOIRES

III.2.1. LES PLE

Ces circuits disposent :

- D'une matrice *ET fixe* et complète. Donc le nombre de termes produits est égal à $2n$ pour n entrées. La matrice ET a la structure d'un décodeur binaire à n bits.
- D'une matrice *OU programmable*.
- D'un *circuit de sortie trois états à commande unique*.

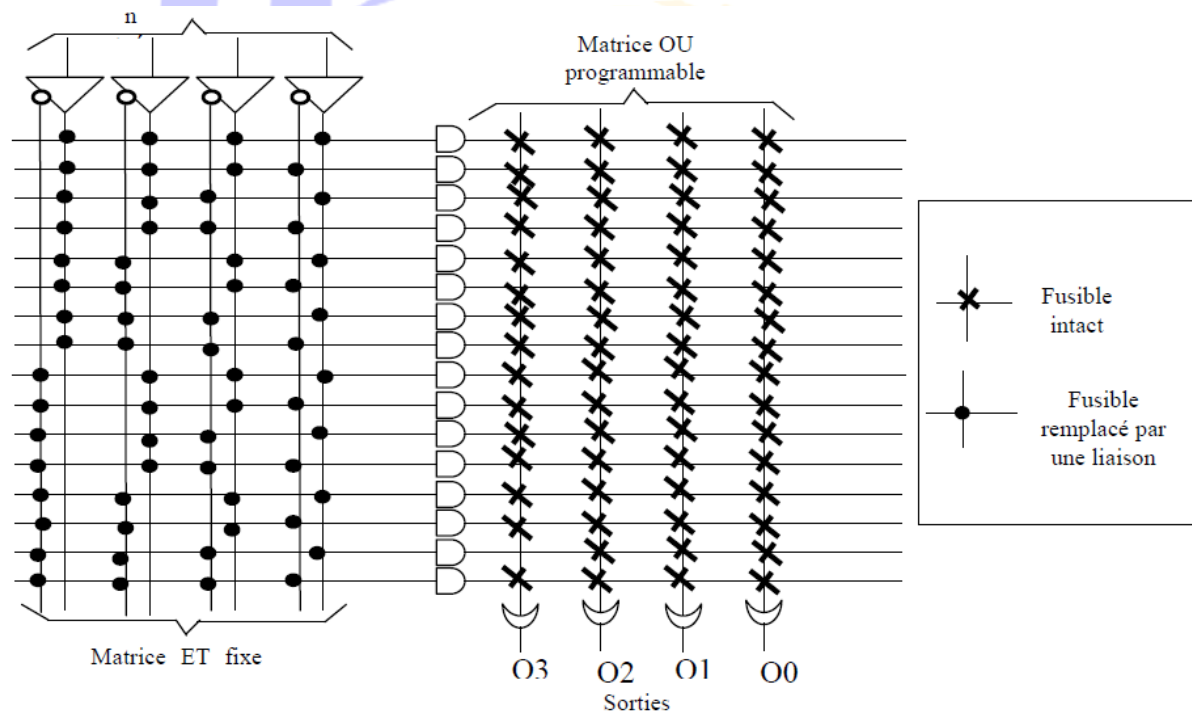


Figure I.1 : Structure interne d'un PLE

☞ Une fonction F est donc réalisée en programmant sa table de vérité. Donc en mémorisant la valeur de F pour toutes les combinaisons des entrées.

III.2.2.LES PAL (PROGRAMMABLE ARRAY LOGIC)

☞ Les PAL sont des composants programmables une seule fois : ceci est dû au fait que la programmation est réalisée par un réseau de fusibles irréversibles.

☞ La structure de base d'un PAL est l'opposée de celle d'une PLE. Il dispose d'une matrice *ET fixe programmable* et d'une matrice *OU figée*.

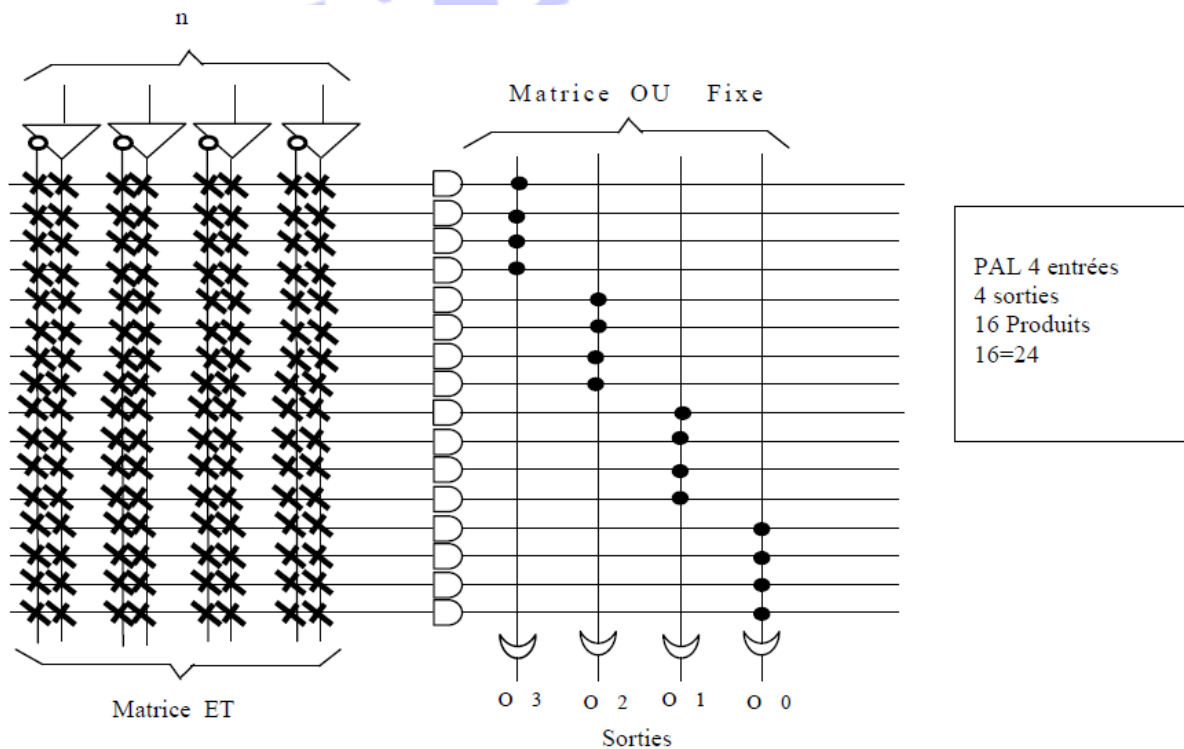


Figure I.2 : Structure interne d'un PAL

III.2.3.LES PLA OU FPLA (FIELD PROGRAMMABLE LOGIC ARRAY)

L'architecture des PLA est donnée ci-dessous :

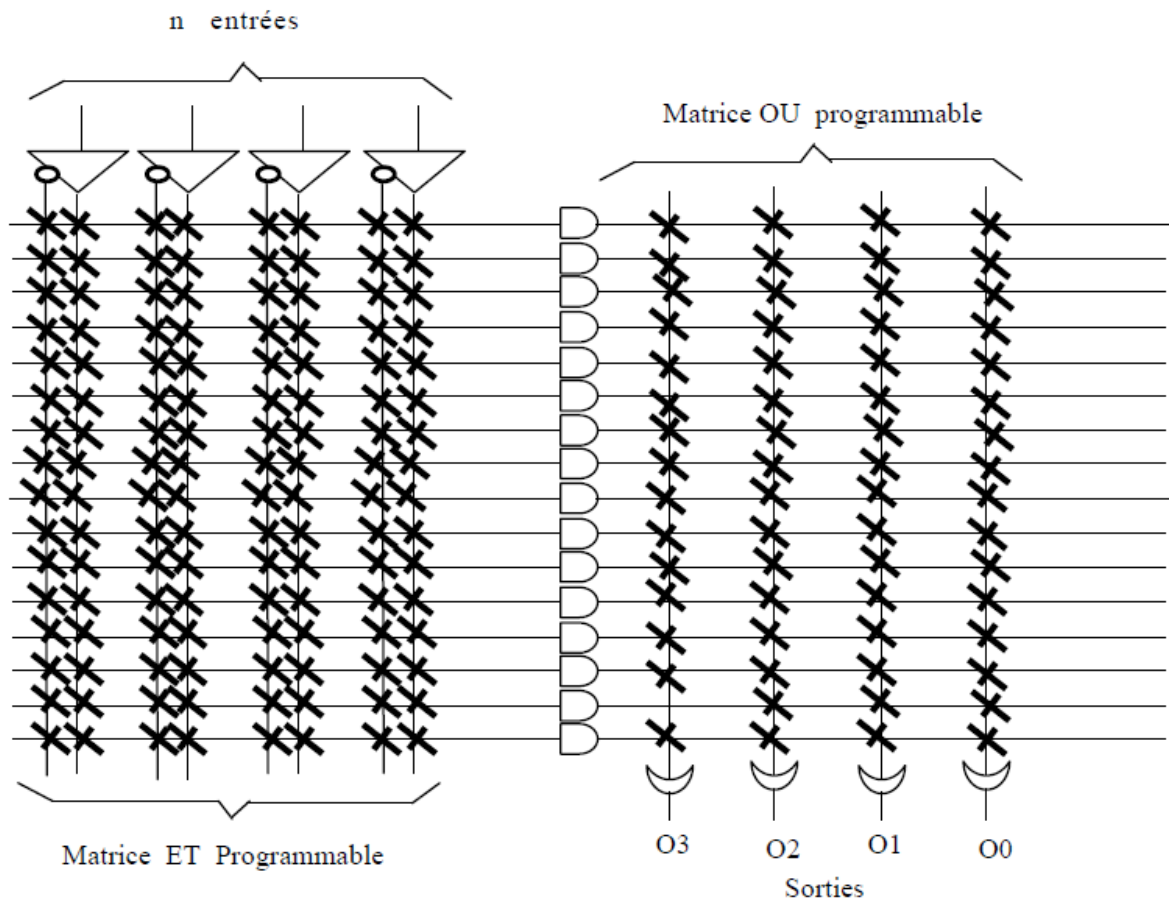


Figure I.3 : FPLA 4 Entrées 4 Sorties 16 Produits

- Les matrices ET et OU sont programmables indépendamment, ce qui offre une très grande souplesse. Ces circuits disposent également d'un circuit de sortie programmable (circuit d'inversion).

III.2.4. LES CIRCUITS LOGIQUES SEQUENTIELS PROGRAMMABLES

Les constructeurs de PAL combinatoires ont développé des réseaux logiques programmables adaptés aux systèmes séquentiels. On en trouve deux types :

- Les FPLS (Field Programmable Logic Sequenser) : Ils intègrent, en plus des matrices ET-OU classiques, des bascules dont la fonctionnalité est programmable.
- Les PAL séquentiels : Ce sont des circuits PAL dont la matrice ET-OU est identique à celle des PAL combinatoires et qui disposent en sortie de bascules D qui échantillonnent le résultat combinatoire aux fronts montants d'un signal d'horloge.

III.2.5. LES GAL (GENERIC ARRAY LOGIC)

L'inconvénient majeur des PAL est qu'ils ne sont programmables qu'une seule fois. Ceci impose un gaspillage important de ces circuits lorsqu'on veut développer un nouveau produit. La firme LATTICE a donc pensé à remplacer les fusibles irréversibles des PAL par des transistors MOS FET pouvant être régénérés. Ceci a donc donné naissance aux GAL que l'on

pourrait traduire par «Réseau logique Générique». Ces circuits peuvent donc être reprogrammés à volonté sans pour autant avoir une durée de vie restreinte.

III.2.6. PROGRAMMATION DES PLD

La programmation des PLD nécessite un logiciel adapté pour le développement du programme et un programmeur permettant de programmer le circuit.

☞ Le logiciel de développement permet de générer un fichier JEDEC à partir du programme rentré par l'opérateur. Il simule aussi le fonctionnement du PLD avec ce programme.

☞ Le fichier JEDEC est un ensemble de données binaires indiquant au programmeur les fusibles à « griller ».

☞ Le programmeur permet de « griller » les fusibles du PLD en fonction des données du fichier JEDEC.

III.2.7. LES CPLD (COMPLEX PROGRAMMABLE LOGIC DEVICES)

Les PLD décrits ci-dessus (Les ROM, PLA, PAL et GAL) sont parfois appelés des circuits logique programmable simples (Simple Programmable Logic Devices – SPLD). Ils sont limités par le nombre de pattes pouvant être placées sur la puce.

Pour réaliser des fonctions de plus en plus complexes, plusieurs PLA ou PAL sont intégrés sur la même puce, avec un réseau d'interconnexions, et ont donné naissance à un CPLD. Un CPLD est donc une extension naturelle des circuits PAL.

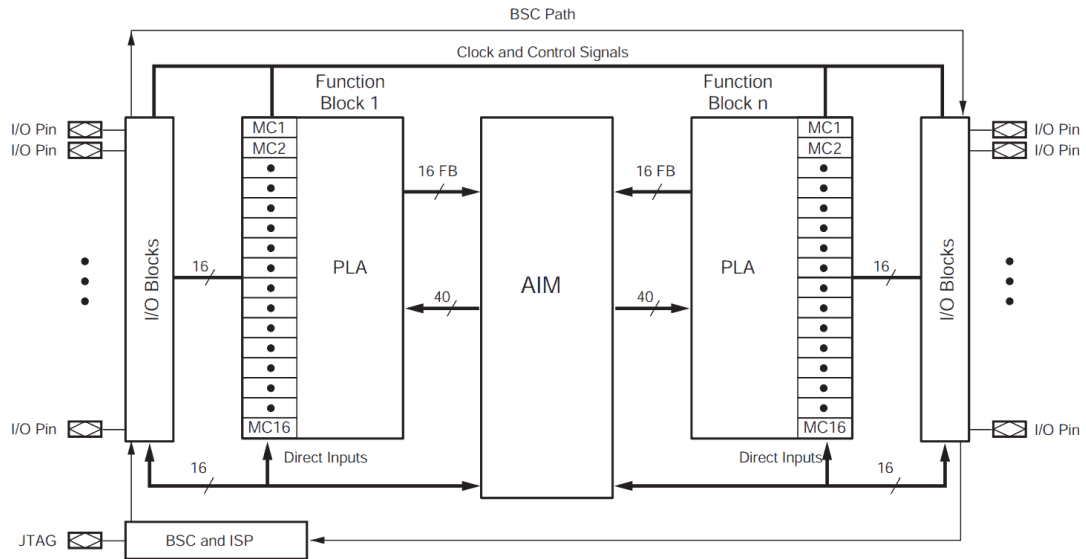
Le réseau d'interconnexions permet de relier les pattes de la puce à différents blocs internes, mais aussi à relier les blocs entre eux. Il est donc possible de composer des fonctions logiques très complexes incluant des machines à états et de petites mémoires.

Pour voir de près la composition d'un CPLD, jetons un œil sur le CPLD, Xilinx CoolRunner-II comme exemple pratique. Il s'agit d'une famille CPLD très populaire, fabriquée par Xilinx. La figure ci-dessous représente l'architecture de ce composant.

Voyons quelques fonctionnalités de cette figure:

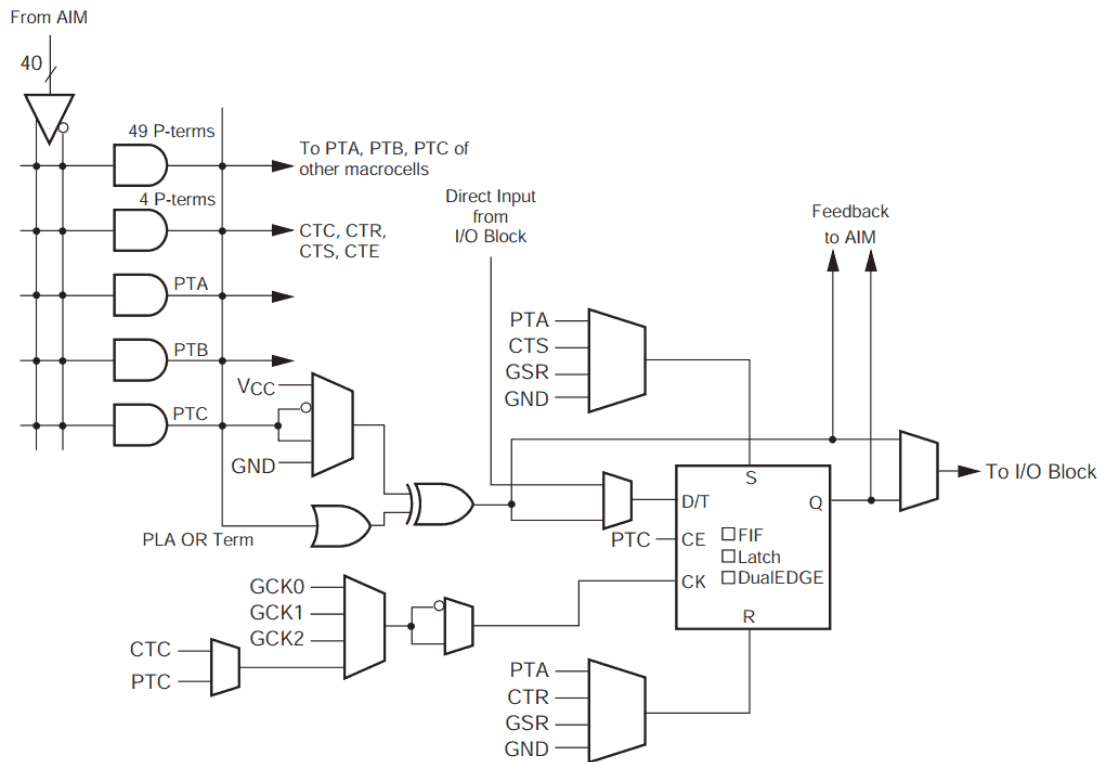
- Ce CPLD a peu de blocs fonctionnels allant du “Function Block 1” au “Function Block n” (ce nombre varie de 2 à 32 dans le cas des CPLD CoolRunner-II)
- 16 lignes de sortie de chaque “Function Block” entrent dans l'AIM (Advanced Interconnect Matrix), tandis que 40 lignes de signaux de sortie depuis l'AIM entrent dans les “Function Blocks”.
- Chaque “Function Block” dispose de 16 MacroCells (Macro-Cellules) (MC1 à MC16).
- Chaque MacroCell a accès à 16 signaux provenant des blocs d'E/S.

- La structure de MacroCell est illustrée ci-dessous. Chaque macro cellule contient 1 Flip-Flop et un tableau PLA de somme de produits que les utilisateurs peuvent utiliser pour créer une logique combinatoire ou séquentielle. Le nombre maximum de Flip-Flop disponibles dans le CLPD CoolRunner-II (XC2C512) est donc de 512.
- L'AIM peut être programmé pour interconnecter les signaux comme nous le voulons. Mais comme le nombre de signaux, pour chaque "Function Block", est limité (deuxième point ci-dessus), cela donnerai un nombre d'interconnexions (configuration) bien limité.



DS090_01_121201

Figure 1: CoolRunner-II CPLD Architecture



DS090_03_121201

Figure 3: CoolRunner-II CPLD Macrocell

III.2.8.LES FPGA (FIELD PROGRAMMABLE GATE ARRAY)

Les FPGA se distinguent des CPLD par le fait que les blocs de logique programmable (Configurable Logic Block - CLB) des FPGA sont beaucoup plus nombreux (peuvent atteindre quelque millions) et beaucoup plus complexes que les Macrocells des CPLD, et que les bascules (Flip-Flops) des FPGA sont également beaucoup plus nombreux (peuvent atteindre quelque millions), comparer au nombre réduit de Flip-Flops dans les CPLD (quelques centaines ≈ 512). Cette différence assez large permet de favoriser les FPGA pour l'implémentation des fonctions logiques beaucoup plus complexes ainsi que la réalisation et l'intégration d'une multitude de circuits indépendants.

Les FPGA sont principalement constitués de tables de correspondance (Look Up Table: LUT), de multiplexeurs et de bascules (Flip-Flops). Alors qu'au début ils permettaient de remplacer quelques composantes discrètes de façon plus efficace que les PAL et les PLA, les FPGA concurrencent de nos jours certains microprocesseurs et microcontrôleurs en complexité et en performance.

Un FPGA est composé à la base de :

- un réseau de blocs de logique programmable (Configurable Logic Block - CLB), chaque bloc pouvant réaliser des fonctions complexes de plusieurs variables, et comportant des éléments à mémoire;
- un réseau d'interconnexions programmables entre les blocs;
- des blocs spéciaux d'entrée et de sortie avec le monde extérieur (Input/Output Block – IOB).

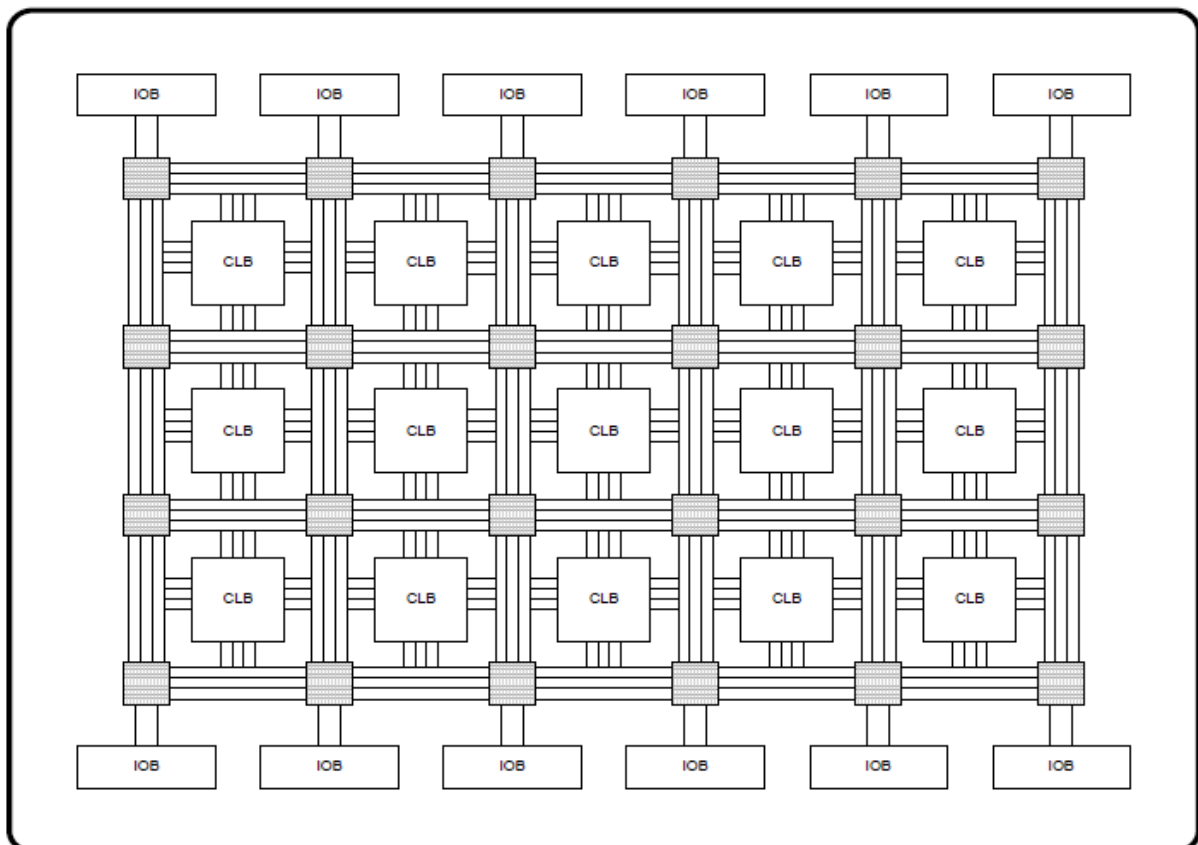


Figure I.5 : Modèle d'un FPGA

Ils sont également équipés de structures dédiées pour diverses fonctions telles que la RAM, les émetteurs-récepteurs série haute vitesse, les blocs de traitement du signal numérique (DSP) ...etc. En termes de variété de choix et de flexibilité, les CPLD ne sont pas à la hauteur des FPGA.

Le tableau ci-dessous opère une comparaison fonctionnelle entre un CPLD et un FPGA.

Où utiliser les CPLD ou les FPGA?

Lorsqu'une conception nécessite une simple logique ou une fonctionnalité qui n'a pas besoin d'être modifiée fréquemment, ou lorsque vous avez besoin d'un fonctionnement instantané, optez pour les CPLD. Sinon, pour la plupart des autres applications, les FPGA sont généralement préférés. Parfois, vous pouvez trouver les deux CPLD + FPGA dans une même conception. Dans ces conceptions, les CPLD réalisent généralement une simple logique, et ils sont responsables du

«démarrage» du **FPGA** ainsi que du contrôle de la réinitialisation et de la séquence de démarrage de la carte complète.

IV. TERMINOLOGIE : LE, LAB, ALM, SLICE, CLB

Pour des raisons internes aux différents fabricants, plusieurs termes sont utilisés pour parler de l'architecture interne des **FPGA**.

↘ Fabricant **Altera**

- Pour les **FPGA** de la famille **Cyclone**, **Altera** utilise le terme **Logic Element (LE)** pour une cellule de base incluant une table de conversion, un additionneur et un registre. Un **Logic Array Bloc (LAB)** regroupe 10 **LE**.
- Pour la famille **Stratix**, **Altera** a remplacé les **LEs** par des blocs plus complexes, les **Adaptive Logic Modules (ALM)**. Un **ALM** comprend deux tables de conversion, deux additionneurs et deux registres. Pour la famille **Stratix**, un **LAB** regroupe 10 **ALM**.

↘ Fabricant **Xilinx**

- Pour les **FPGA** des familles **Spartan** et **Virtex**, **Xilinx** utilise le terme **slice** pour une cellule de base incluant deux tables de conversion, deux additionneurs et deux registres. Un **Configurable Logic Block (CLB)** regroupe deux ou quatre **slices**, selon la famille de **FPGA**.

V. BLOCS DE MEMOIRE INTEGREE

Les blocs de mémoire ont été parmi les premiers modules ajoutés à cause du grand besoin en mémoire de la plupart des applications. L'avantage important à intégrer des blocs de mémoire près de la logique configurable est la réduction significative des délais de propagation et la possibilité de créer des canaux de communication parallèle très larges. La Figure ci-dessous illustre l'intégration de blocs de mémoire sous la forme d'une colonne entre les **CLB** d'un **FPGA**.

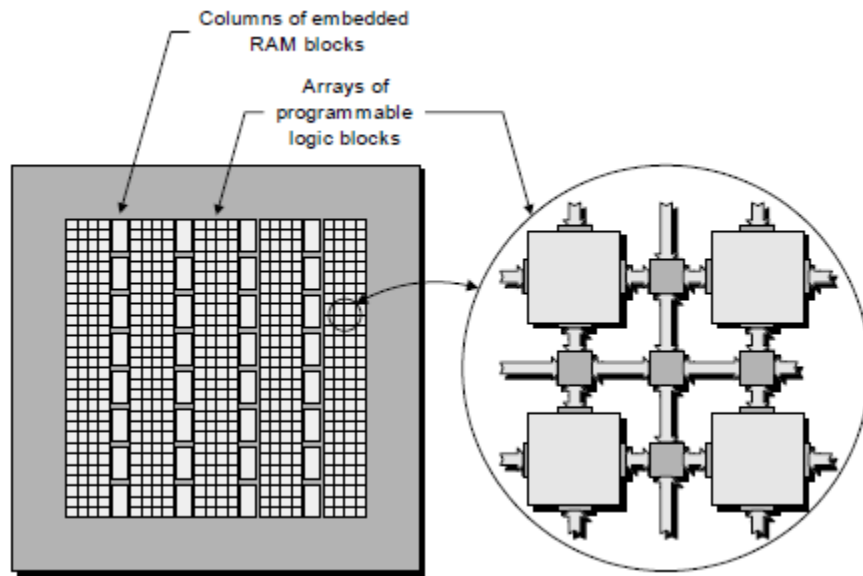


Figure I.6 : Mémoire RAM intégrée

- La quantité de mémoire présente dans les blocs de RAM varie à travers les différentes familles de FPGA, mais on peut retrouver jusqu'à 10 Méga bits de mémoire dans les plus gros et plus récents modèles. Les blocs peuvent être utilisés indépendamment ou en groupes, offrant une versatilité rarement rencontrée dans les systèmes numériques. De plus, les blocs de mémoire peuvent être utilisés pour implémenter des fonctions logiques, des machines à états, des registres à décalage très larges, etc.

VI. FONCTIONS ARITHMETIQUES AVANCEES

La multiplication est une opération fondamentale dans les applications de traitement du signal pour lesquelles les FPGA sont très populaires. Les fabricants de FPGA ont donc ajouté à leurs architectures des blocs spéciaux pour réaliser cette opération. Les multiplieurs sont en général disposés près des blocs de mémoire RAM pour des raisons d'efficacité de routage des signaux et de disposition des ressources du FPGA sur une puce. La figure ci dessous illustre un tel arrangement.

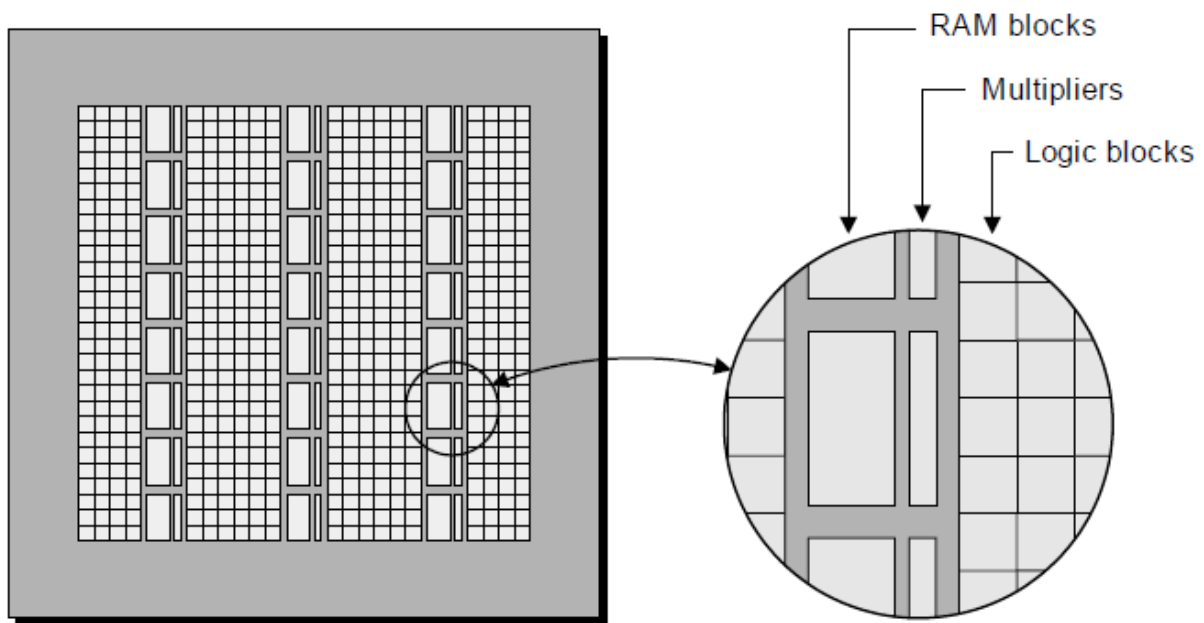
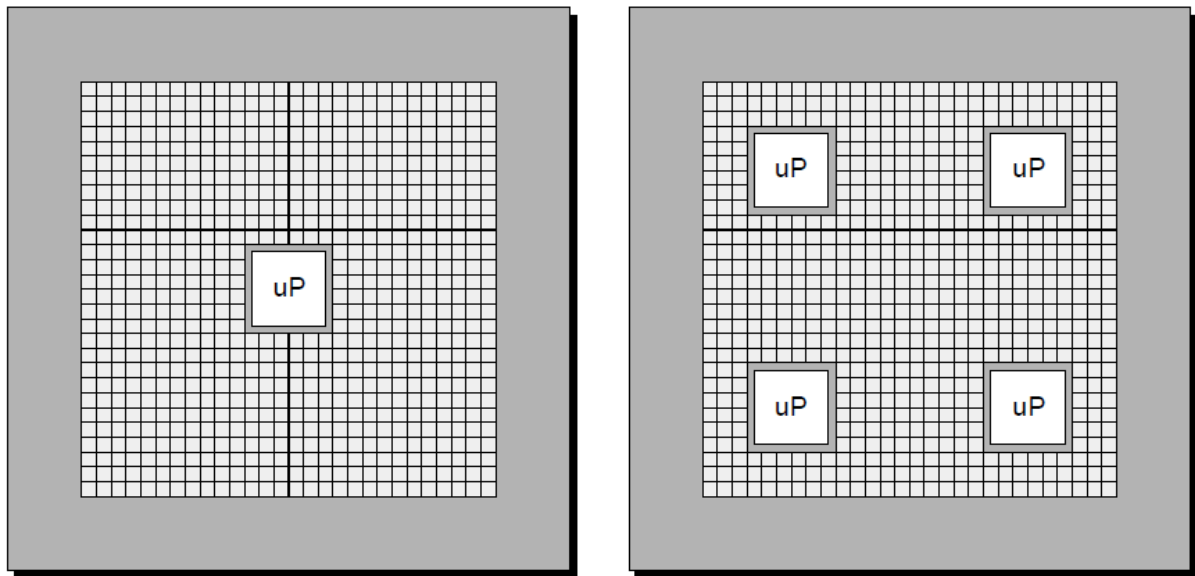


Figure I.7 : Multiplicateurs intégrés

- Les paramètres d'opération des multiplicateurs varient selon les familles de **FPGA**. Un format populaire est un multiplicateur signé de **18×18 bits**, ce qui est suffisant pour beaucoup d'applications de traitement du signal (le son est encodé avec 8 bits sur une ligne de téléphone et avec 16 bits par canal pour un CD audio). Pour les applications devant traiter des opérandes plus larges, il est possible de regrouper plusieurs multiplicateurs. Le nombre de multiplicateurs varie, mais on en retrouve facilement des dizaines même sur les **FPGA** les plus modestes.

VII. MICROPROCESSEURS FIXES

Avec le nombre de transistors intégrables sur une puce doublant tous les 18 mois, et les besoins important en contrôle dans plusieurs applications, les fabricants de **FPGA** intègrent des processeurs à plusieurs familles de **FPGA**. La figure ci-dessous illustre deux variations, avec un et quatre processeurs fixes.



(a) One embedded core

(b) Four embedded cores

Figure I.8 : Processeur intégré

- Les microprocesseurs intégrés ne sont pas réalisés à l'aide de blocs de logique ou de mémoire du **FPGA**. Ce sont plutôt des régions de la puce optimisées comme si le microprocesseur était réalisé par lui-même dans un circuit intégré.
- En intégrant un ou plusieurs microprocesseurs à un **FPGA**, on obtient un tissu de calcul d'une puissance très intéressante. En effet, une application peut être divisée efficacement entre des parties matérielles et logicielles; les parties matérielles sont réalisées avec les ressources configurables du **FPGA**, et les parties logicielles sont réalisées avec les microprocesseurs.
 - Là où le parallélisme des calculs l'exige, les blocs de logique configurable peuvent accélérer les calculs par un facteur de **10x**, **100x** ou plus.
 - Là où l'application risque d'être modifiée, ou si elle nécessite beaucoup de contrôle ou d'interface avec le monde extérieur ou entre des modules, une solution logicielle peut être obtenue plus facilement.
- Avec les microprocesseurs entourés des ressources configurables du **FPGA**, on peut atteindre des taux d'échange de données très élevés. Tout d'abord, la proximité physique des dispositifs réduit les délais. Ensuite, il est possible de mettre en place des liens de communication très larges.

VIII. GENERATION ET DISTRIBUTION D'HORLOGE

Un problème important lors de la conception et réalisation d'un circuit numérique concerne la génération et la distribution du signal d'horloge. Dans une application de communications et de traitement du signal, un même circuit peut nécessiter une dizaine d'horloges de fréquences et de phases différentes.

Pour faciliter la tâche des concepteurs, les **FPGA** incluent maintenant des circuits spécialisés de génération, régulation et distribution de l'horloge. Ce type de circuit accepte en entrée une horloge externe et génère une ou plusieurs horloges internes. Les horloges internes peuvent avoir des fréquences et des phases différentes.

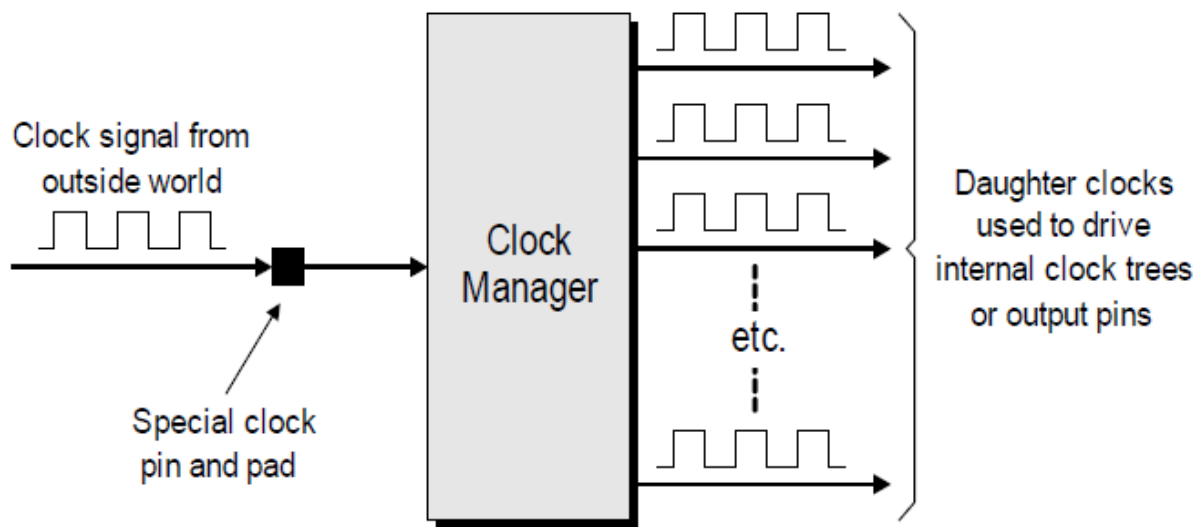


Figure I.9 : Génération de signaux d'horloge à partir d'une référence externe

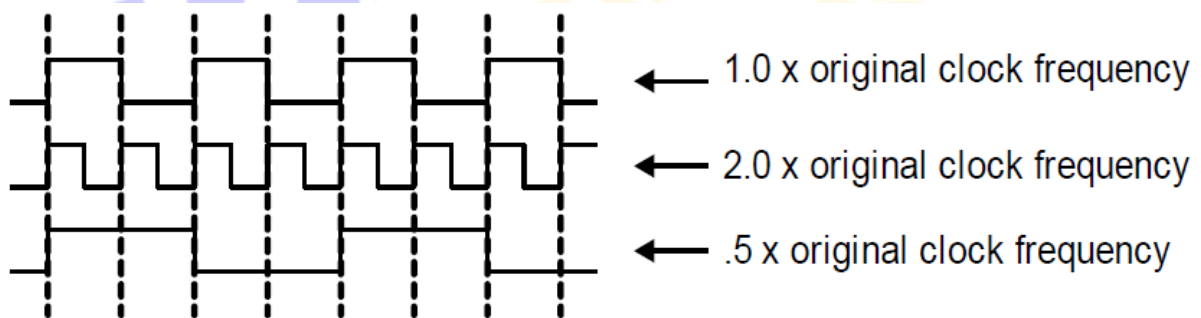


Figure I.10 : Varier la fréquence d'horloge

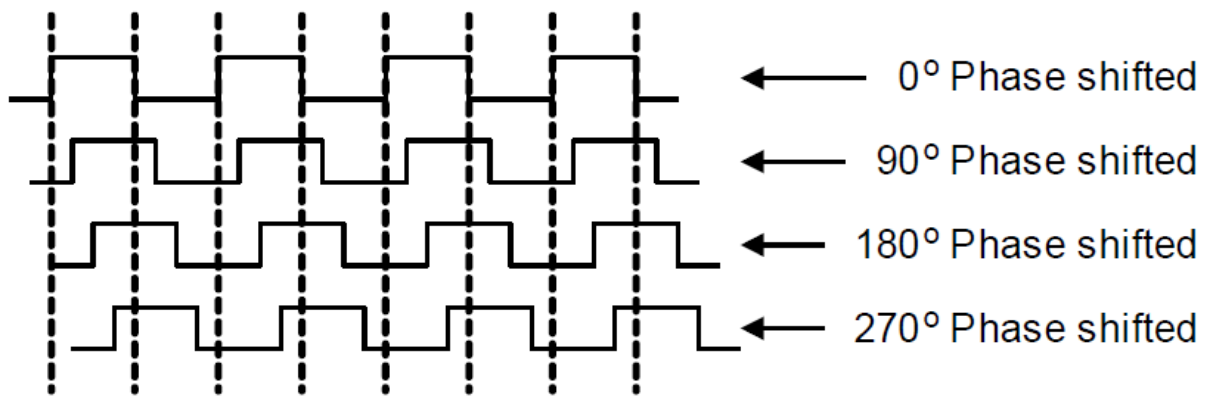


Figure I.11 : Varier la phase de l'horloge

Pour distribuer l'horloge à travers la puce tout en minimisant le déphasage, on utilise un réseau en arbre dédié (*Clock tree*). Ce réseau est alimenté soit par une patte spéciale du FPGA à laquelle est associé un amplificateur dédié, ou bien par l'entremise de signaux internes pouvant être routés au même amplificateur. La figure suivante illustre un tel arrangement.

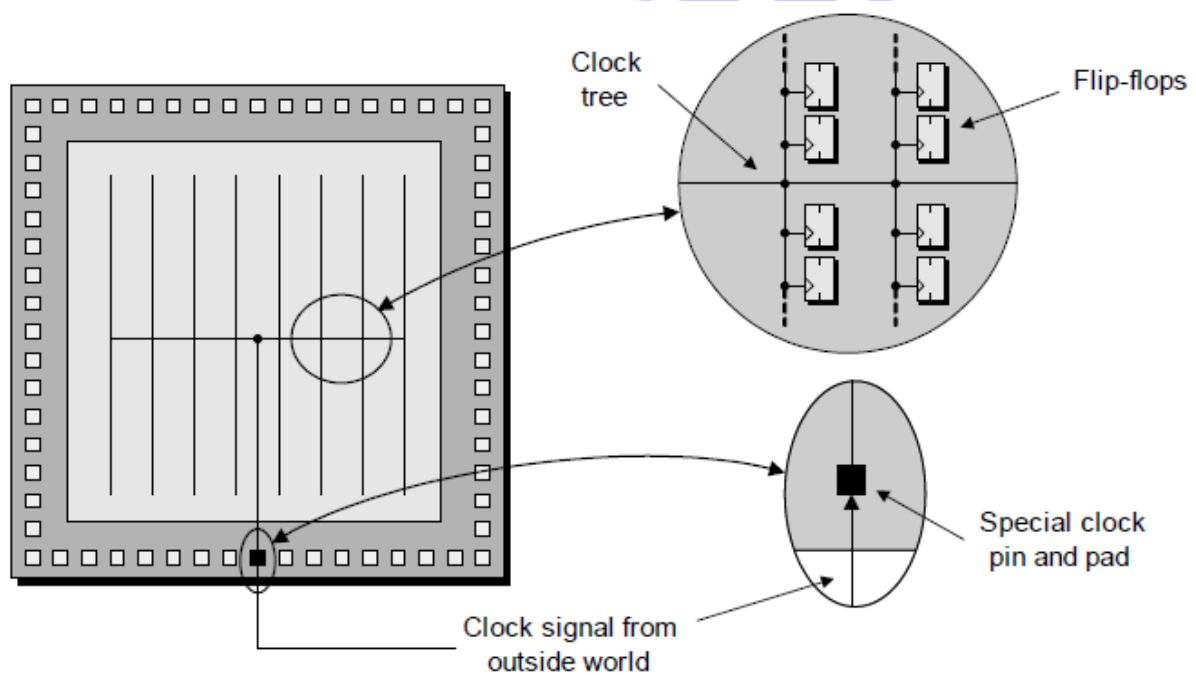


Figure I.12 : Arbre de distribution d'horloge