

# ASIC et composants à réseaux logiques programmables : PAL, PLD, CPLD, FPGA.

L'électronique moderne se tourne de plus en plus vers le numérique qui présente de nombreux avantages sur l'analogique : grande insensibilité aux parasites et aux dérives diverses, modularité et (re)configurabilité, facilité de stockage de l'information etc...

Les circuits numériques nécessitent par contre une architecture plus lourde et leur mode de traitement de l'information met en oeuvre plus de fonctions élémentaires que l'analogique d'où découle des temps de traitement plus long.

Aussi les fabricants de circuits intégrés numériques s'attachent-ils à fournir des circuits présentant des densités d'intégration toujours plus élevée, pour des vitesses de fonctionnement de plus en plus grandes.

D'abord réalisées avec des circuits SSI (Small Scale Integration) les fonctions logiques intégrées se sont développées avec la mise au point du transistor MOS dont la facilité d'intégration a permis la réalisation de circuits MSI (Medium Scale Integration) puis LSI (Large Scale Integration) puis VLSI (Very Large Scale Integration). Ces deux dernières générations ont vu l'avènement des microprocesseurs et microcontrôleurs.

Bien que ces derniers aient révolutionné l'électronique numérique par la possibilité de réaliser n'importe quelle fonction par programmation d'un composant générique, ils traitent l'information de manière séquentielle (du moins dans les versions classiques), ne répondant pas toujours aux exigences de rapidité.

Au début des années 70 sont apparus les premiers composants (en technologie bipolaire) entièrement configurable par programmation. La nouveauté résidait dans le fait qu'il était maintenant possible d'implanter physiquement par simple programmation, au sein du circuit, n'importe quelle fonction logique, et non plus de se contenter de faire réaliser une opération logique par un microprocesseur dont l'architecture est figée.

D'abord dédiés à des fonctions simples en combinatoire (décodage d'adresse par exemple), ces circuits laissent aujourd'hui au concepteur la possibilité d'implanter des composants aussi divers qu'un inverseur et un microprocesseur au sein d'un même boîtier ; le circuit n'est plus limité à un mode de traitement séquentielle de l'information comme avec les microprocesseurs. L'intégration des principales fonctions numériques d'une carte au sein d'un même boîtier permet de répondre à la fois aux critères de densité et de rapidité (les capacités parasites étant plus faibles, la vitesse de fonctionnement peut augmenter).

La plupart de ces circuits sont maintenant **programmés à partir d'un simple ordinateur type PC** directement **sur la carte** où ils vont être utilisés. En cas d'erreur, ils sont reprogrammables électriquement sans avoir à extraire le composant de son environnement.

De nombreuses familles de circuits sont apparues depuis les années 70 avec des noms très divers suivant les constructeurs : des circuits très voisins pouvaient être appelés différemment par deux constructeurs concurrents, pour des raisons de brevets et de stratégies commerciales. De même une certaine inertie dans l'évolution du vocabulaire a fait que certains circuits technologiquement différents ont le même nom.

Le terme même de circuit programmable est ambigu, la programmation d'un FPGA ne faisant pas appel aux mêmes opérations que celle d'un microprocesseur. Il serait plus juste de parler pour les PLD, CPLD et FPGA de circuits à architecture programmable ou encore de circuits à réseaux logiques programmables.

Ce domaine de l'électronique est aussi celui qui certainement a vu la plus forte évolution technologique ces dernières années :

- en moins de 15 ans la **densité d'intégration a été multipliée par 200** (2000 à 20 000 portes en 85 pour 72 000 à 4 000 000 en 2000)
- en moins de 10 ans la **vitesse de fonctionnement par 6** (40 MHz en 91 pour 240 MHz en 2000)

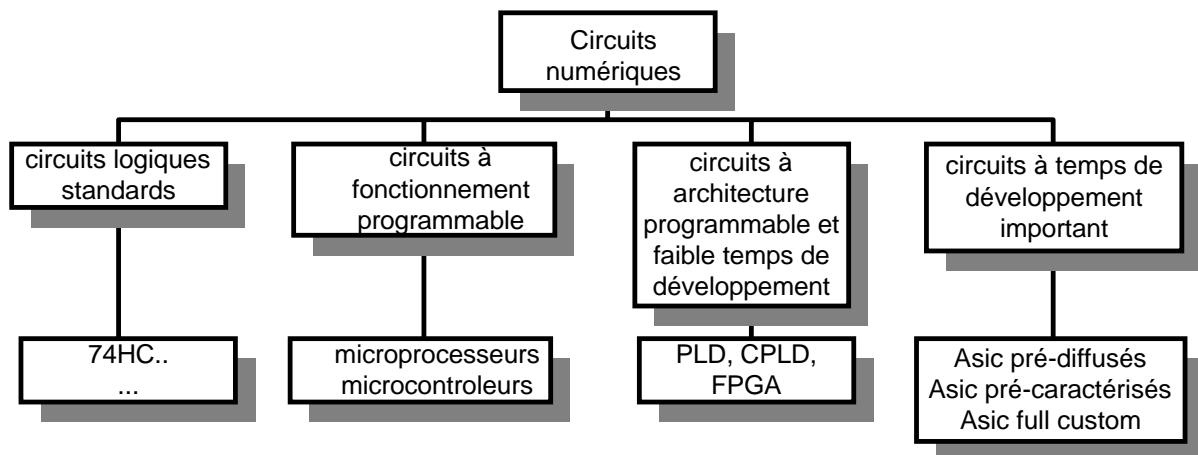
- la taille d'un transistor est passée **de 1,2  $\mu\text{m}$  en 91 à 0,18  $\mu\text{m}$**  en 2000.
- les technologies de conception ont fortement évolué, tel constructeur initiateur d'un procédé l'abandonne pour un autre, alors que le concurrent le reprend à son compte.
- la tension d'alimentation est passée **de 5 V à 1,8 V** diminuant ainsi la consommation.

Aussi est-il très difficile de s'y retrouver et de donner des ordres de grandeurs qui puissent être comparés. Nous tenterons dans cet exposé une clarification des choses dont la volonté de simplification pourra être facilement prise en défaut.

Parallèlement à ces circuits, on trouvera les ASIC (Application Specific Integrated Circuits) qui sont des composants où le concepteur intervient au niveau du dessin de la pastille de silicium en fournissant des masques à un fondeur. On ne peut plus franchement parler de circuits programmables. Les temps de développement long ne justifient l'utilisation que pour des grandes séries.

Les PLD, CPLD et FPGA sont parfois considérés comme des ASIC par certains auteurs.

Le tableau ci-après tente une classification possible des circuits numériques :



Nous nous intéresserons surtout aux circuits à architecture programmable à faible temps de développement. Le principe de base des circuits nous intéressant ici consiste à réaliser des connexions logiques programmables entre des structures présentant des fonctions de bases. Le premier problème va donc être d'établir ou non suivant la volonté de l'utilisateur, un contact électrique entre deux points. Aussi, nous intéresserons nous, avant de passer aux circuits proprement dits et à leur programmation, à ces technologies d'interconnexion.

Mais avant toutes choses, rappelons comment coder une fonction logique.

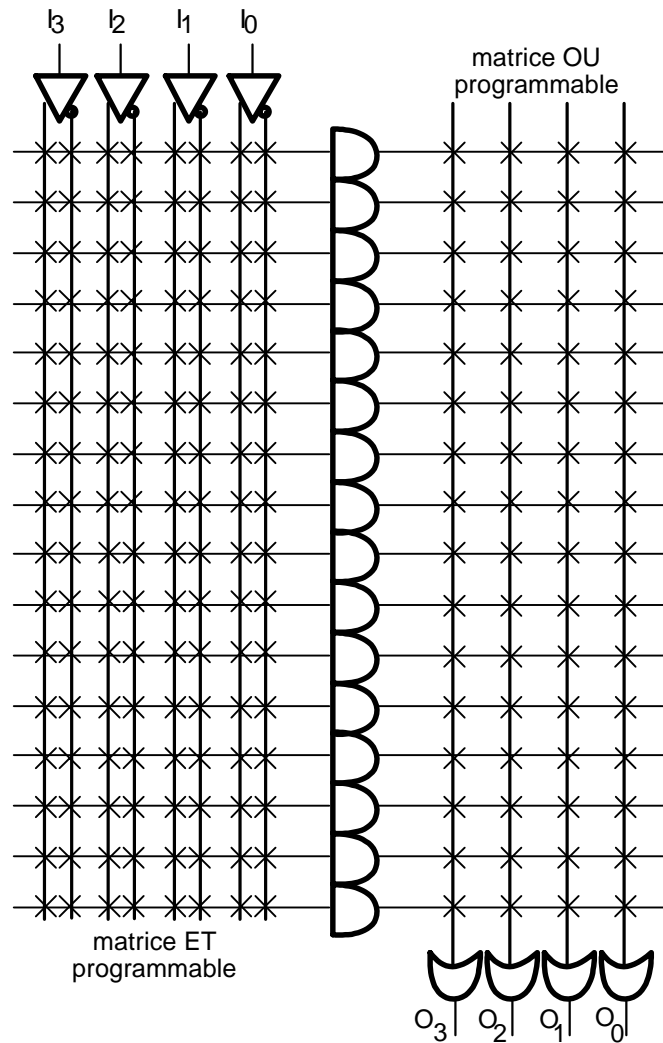
## 1. Codage d'une fonction logique

La base d'une fonction logique, est toujours une fonction combinatoire. Pour obtenir une fonction séquentielle, il suffira ensuite de réinjecter les sorties sur les entrées, ce qui donnera alors un système asynchrone. Si on souhaite un système synchrone, on intercalera avant les sorties, une série de bascules à front, dont l'horloge commune synchronisera toutes les données et évitera bien des aléas de fonctionnement.

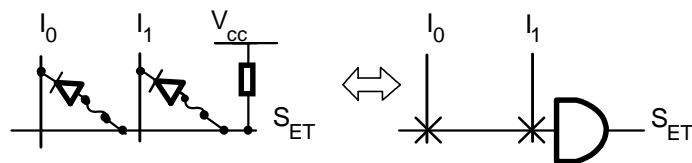
Pour coder une fonction combinatoire, trois solutions sont classiquement utilisées.

### 1.1. Sommes de produits, produits de somme et matrice PLA (Programmable Logic Array)

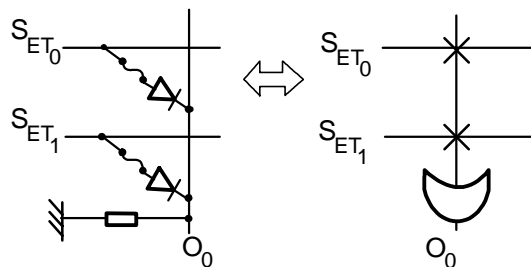
N'importe quelle fonction peut être codée par une somme de produit, par un produit de somme ou un mélange des deux. On peut immédiatement en déduire une structure de circuits, appelé matrice PLA (Programmable Logic Array). La figure suivante représente une matrice PLA à 4 entrées et 4 sorties :



Chacune des 4 entrées et son complémentaire arrive sur une des  $2^4=16$  portes ET à  $2 \times 4=8$  entrées. Afin de simplifier la représentation, les 8 lignes ont été représentées par une seule, chaque croix représentant une connexion programmable (un fusible par exemple). La figure suivante propose un principe de réalisation des fonctions de la matrice ET ; a mise au niveau logique 0 (NL0) d'une des entrées  $l_x$  impose un NL0 en sortie :



et de la matrice OU, sur laquelle une entrée au NL1 impose un NL1 en sortie :

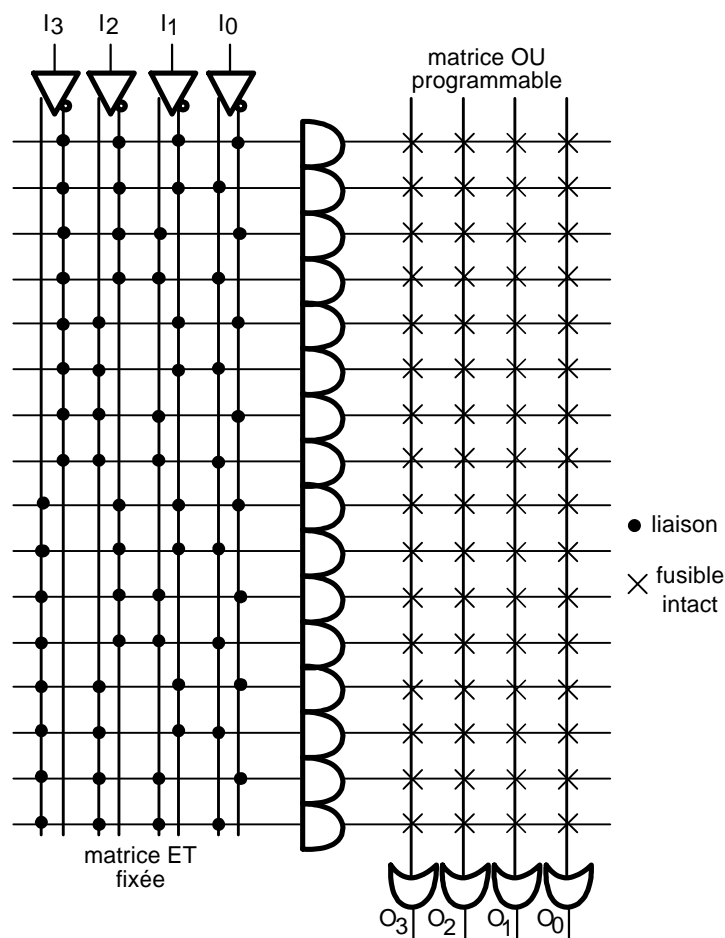


Ce type de structure est utilisé dans certains circuits ASIC (Application Specific Integrated Circuit) et demande une densité d'intégration importante : en effet pour  $n$  variables en entrées, il faut  $2^n$  fonctions ET à  $2n$  entrées et au moins un OU à  $2^n$  entrées (il y a en effet  $2^n$  combinaisons possibles, chaque combinaison dépendant de l'entrée et de son complémentaire).

La plupart des applications n'exigent pas une telle complexité et on peut se contenter d'une matrice ET programmable et d'une matrice OU figée. De même, il est peu probable d'utiliser tous les termes produits et on peut alors limiter le nombre d'entrées de la fonction OU. C'est le principe utilisé par les circuits programmable, appelés au début PAL (Programmable Array Logic), mais plus communément désigné aujourd'hui sous le terme PLD (Programmable Logic Device).

## 1.2. Mémoires

Une fonction combinatoire associe à chacune de ces combinaisons d'entrée une valeur en sortie décrite par sa table de vérité. C'est le principe de la mémoire où pour chaque adresse en entrée, on associe une valeur en sortie, sur un ou plusieurs bits. La structure physique des mémoires fait appelle à une matrice PLA dont la matrice ET est figée et sert de décodeur d'adresse et dont la matrice OU est programmée en fonction de la sortie désirée

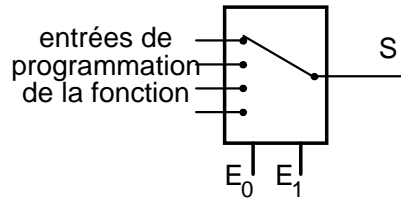


Lorsqu'une adresse est présentée, par exemple  $I_1 I_2 I_3 I_4 = 1111$ , la porte ET concernée passe au NL1 (celle du bas dans l'exemple) et suivant les fusibles laissés intacts sur la matrice OU, on a un mot différent en  $O_0 O_1 O_2 O_3$  (0000 si tous les fusibles sont "grillés" par exemple).

Ce principe utilisé pour les mémoires, l'est aussi dans les CPLD (Complex Programmable Logic Device), mais surtout dans les FPGA (Field Programmable Gate Array) sous le nom de LUT (Look Up Table).

### **1.3. Multiplexeur**

Le multiplexeur permet également de coder une fonction combinatoire, comme le montre la figure suivante :



A chaque valeur des entrées  $E_0$  et  $E_1$  est associé un niveau logique défini dans la table de vérité pour la sortie  $S$ . Ce niveau logique est imposé sur l'entrée de programmation correspondante. Ce principe est utilisé dans les FPGA.

## **2. Technologie d'interconnexions**

Comme nous venons de le voir, l'un des éléments clé des circuits étudié est la connexion programmable.

Du choix d'une technologie dépendra essentiellement :

- la densité d'intégration
- la rapidité de fonctionnement une fois le composant programmé, fonction de la résistance à l'état passant et des capacités parasites
- la facilité de mise en oeuvre (programmation sur site, reprogrammation etc)
- la possibilité de maintien de l'information.

Passons en revue quelques technologies classiquement utilisées et leurs caractéristiques. Ces technologies sont ou pourraient être utilisées pour la réalisation de mémoires. Il faut cependant garder à l'esprit qu'hormis quelques cas particuliers (circuit reprogrammés en cours d'utilisation), le temps d'écriture reste secondaire, le circuit étant habituellement programmé une fois pour toutes avant utilisation.

### **2.1. Connexions programmable une seule fois (OTP : One Time Programming)**

#### **2.1.1. Cellules à fusible**

Ce sont les premières à avoir été utilisées et elles ont aujourd'hui disparu au profit de technologies plus performantes. Leur principe consistait à détruire un fusible conducteur par passage d'un courant fourni par une tension supérieure à l'alimentation (12 à 25 V)

#### **2.1.2. Cellules à antifusible**

En appliquant une tension importante (16 V pendant 1 ms) à un isolant entre deux zones de semi-conducteur fortement dopées, ce dernier diffuse dans l'isolant et le rend conducteur. Chaque cellule occupe environ  $1,8 \mu\text{m}^2$  ( $700 \mu\text{m}^2$  pour un fusible) ; cette technologie très en vogue permet une haute densité d'intégration.

Hormis la non reprogrammabilité, c'est la meilleure technologie (vitesse et surtout densité d'intégration).

## **2.2. Cellules reprogrammables**

### **2.2.1. Cellule à transistor MOS à grille flottante et EPROM (Erasable Programmable Read Only Mémoire)**

L'apparition du transistor MOS à grille flottante a permis de rendre le composant bloqué ou passant sans application permanente d'une tension de commande. Le principe consiste à piéger ou non (à l'aide d'une tension supérieure à la tension habituelle d'alimentation) des électrons dans la grille.

L'extraction éventuelle des électrons piégés permet le retour à l'état initial. Plusieurs technologies EPROM sont en concurrence :

#### 2.2.1.1. UV-EPROM

Les connexions sont réinitialisable par une exposition à un rayonnement ultraviolet d'une vingtaine de minutes, une fenêtre étant prévue sur le composant. L'effacement est de mise en oeuvre lourde, n'est pas sélectif et ne peut se faire sur site. Ce principe n'est pas utilisé pour les circuits qui nous intéressent.

#### 2.2.1.2. EEPROM (Electrically EPROM)

L'effacement et la programmation se font cette fois électriquement avec une tension de 12 V et peuvent se faire de manière sélective (la reprogrammation de tout le composant n'est pas nécessaire). Une cellule demande toutefois 5 transistors pour sa réalisation, ce qui conduit à une surface importante (75 à 100  $\mu\text{m}^2$  en CMOS 0,6  $\mu\text{m}$ ) et réduit la densité d'intégration possible.

D'autre part le nombre de cycles de programmation est limité à un nombre de 100 (en CMOS 0,6  $\mu\text{m}$ ) à 10 000 (en CMOS 0,8  $\mu\text{m}$ ) à cause de la dégradation des isolants.

La programmation ou l'effacement d'une cellule dure quelques millisecondes.

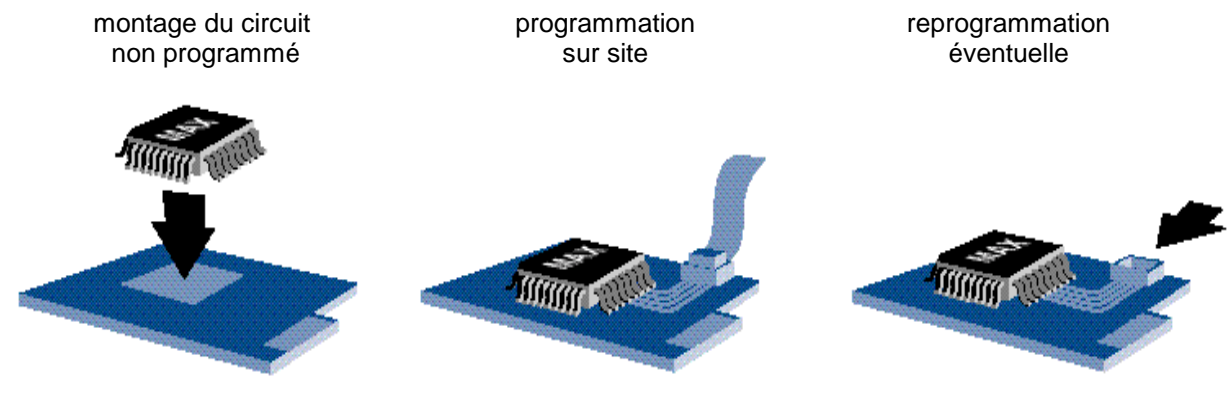
#### 2.2.1.3. Flash EPROM

L'utilisation de deux transistors par cellule uniquement (5 pour l'EEPROM) et une structure verticale permettent une densité d'intégration importante (25  $\mu\text{m}^2$  par cellules en CMOS 0,6  $\mu\text{m}$ ) trois à quatre fois plus importante que l'EEPROM, mais quand même 10 fois moins que la technologie à antifusible. Le nombre de cycle d'écriture ( $10^4$  à  $10^6$ ) est également plus grand que pour l'EEPROM car l'épaisseur de l'isolant est plus importante.

Par contre, la simplicité de la cellule élémentaire n'autorise pas une reprogrammation sélective (éventuellement par secteur), ce qui n'est pas gênant pour le type de circuits qui nous intéresse..

La tension de programmation et d'effacement est de 12 V, avec un temps de programmation de quelques dizaines de  $\mu\text{s}$  pour un temps d'effacement de quelques millisecondes.

Un des inconvénients des cellules flash et EEPROM de nécessiter une alimentation supplémentaire pour la programmation et l'effacement est pallié par les constructeurs en intégrant dans le circuit un système à pompe de charge fournissant cette alimentation. Le composant peut alors être programmé directement sur la carte où il est utilisé. On parle alors de composants ISP : In Situ Programming ou encore suivant les sources, In System Programming .



### **2.2.2. Cellules SRAM à transistors MOS classique**

Ce principe est classiquement choisi pour les FPGA.

Le fait d'utiliser une mémoire de type RAM (donc volatile) impose la recharge de la configuration à chaque mise sous tension : une PROM série mémorise généralement les données.

Ce qui peut paraître un inconvénient devient un avantage si on considère l'aspect évolutif du système qui peut s'adapter à un environnement extérieur changeant et modifier sa configuration en fonction des besoins. On pourra d'autre part facilement intégrer de la mémoire RAM dans le circuit.

Le choix d'une cellule SRAM (Static Read Only Memory) à 6 transistors permet de bénéficier d'un accès sélectif et rapide (quelques ns) en cours d'utilisation.

La taille d'une cellule n'est que deux fois plus forte (50  $\mu\text{m}^2$  par cellule) qu'avec une flash EEPROM.

Cette technologie, utilisée pour les autres circuits VLSI (contrairement aux EEPROM et Flash EPROM et leurs transistors à grille flottante) permet de bénéficier directement des progrès importants réalisés dans ce domaine.

Comme nous venons de la voir, la programmation sur site de ces circuits est une nécessité absolue.

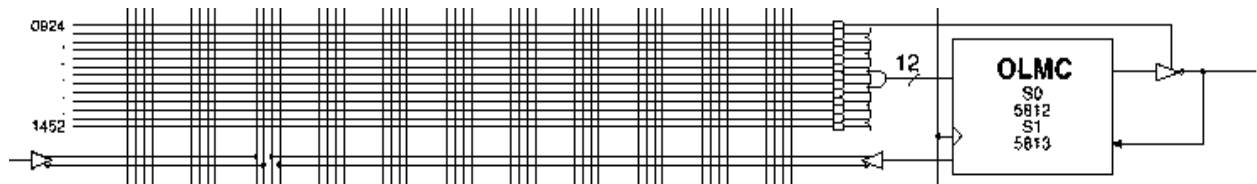
### 3. Architectures utilisées

#### 3.1. PLD (Programmable Logic Device)

Comme nous l'avons vu, d'abord appelés PAL lors de sa sortie, ce circuit utilise le principe de la matrice PLA à réseau ET programmable. Bien que pas très anciens pour les dernières générations, les PLD ne sont presque plus utilisés pour une nouvelle conception. L'un de leur avantage qu'était la rapidité a disparu, les efforts de recherche des constructeurs portant plutôt sur les circuits à plus forte densité d'intégration que sont les CPLD et les FPGA.

Le fait que les PLD soient à la base de la conception des CPLD, très en vogue aujourd'hui, justifie cependant leur étude.

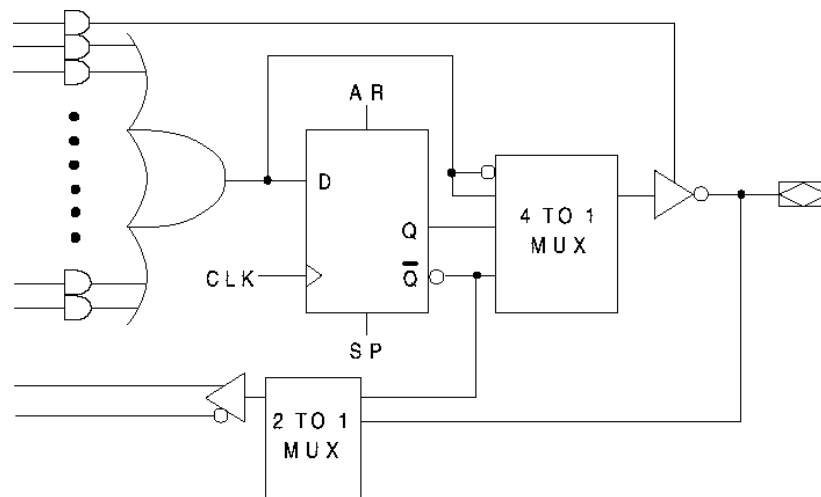
Initialement bipolaire, les cellules de connexions sont aujourd'hui réalisées en technologie MOS à grille flottante. La structure de base comprend un circuit PLA dont seule la matrice ET est programmable.



La partie nommée OLMC (Output Logic MacroCell, dénomination Lattice) sur la figure peut être :

- combinatoire, une simple connexion relie alors la sortie du OU à l'entrée du buffer de sortie, dont la sortie est réinjectée sur le réseau programmable ;
- séquentielle, le bloc OLMC étant alors une simple bascule D ;
- versatile, il est alors possible par programmation de choisir entre les deux configurations précédentes.

Les PLD de dernière génération utilisent des OLMC versatiles, dont on donne ci-après la structure :



Le multiplexeur 4 vers 1 permet de mettre en circuit ou non la bascule D, en inversant ou pas les signaux. Le multiplexeur 2 vers 1 permet de réinjecter soit la sortie, soit l'entrée du buffer de sortie vers le réseau programmable.

**3.1.1. Désignation**

Elle est de la forme PAL EE T SS où EE représente le nombre d'entrées, SS le nombre de sorties et T le type du PAL :

Type	désignation
H	combinatoire à logique positive (état actif à 1)
L	combinatoire à logique négative
P	combinatoire à logique à polarité programmable
C	combinatoire à logique complémentaire
R	à registre
RA	à registre asynchrone
RP	à registre programmable
V	versatile

exemple : PAL 22V10.

Pour les dernières générations, on trouvera plutôt la référence de type GAL EE T SS, (GAL pour Generic Array Logic) pour désigner un circuit à transistor MOS à grille flottante, donc reprogrammable électriquement.

**3.1.2. Programmation**

Les étapes de programmation sont assistées par ordinateur et présentent la chronologie suivante :

- description de la fonction souhaitée par entrée schématique ou syntaxique ; dans ce dernier cas on utilise un langage approprié appelé HDL (Hardware Description Language) comme le langage ABEL ou éventuellement VHDL (Very High speed integrated circuit HDL). On en profite pour définir des vecteurs de test de la fonction réalisée
- simulation logique puis temporelle de la fonction réalisée et éventuellement retour à l'étape précédente.
- compilation et génération d'un fichier de programmation (fichier au standard JEDEC)
- programmation et test physique du composant.

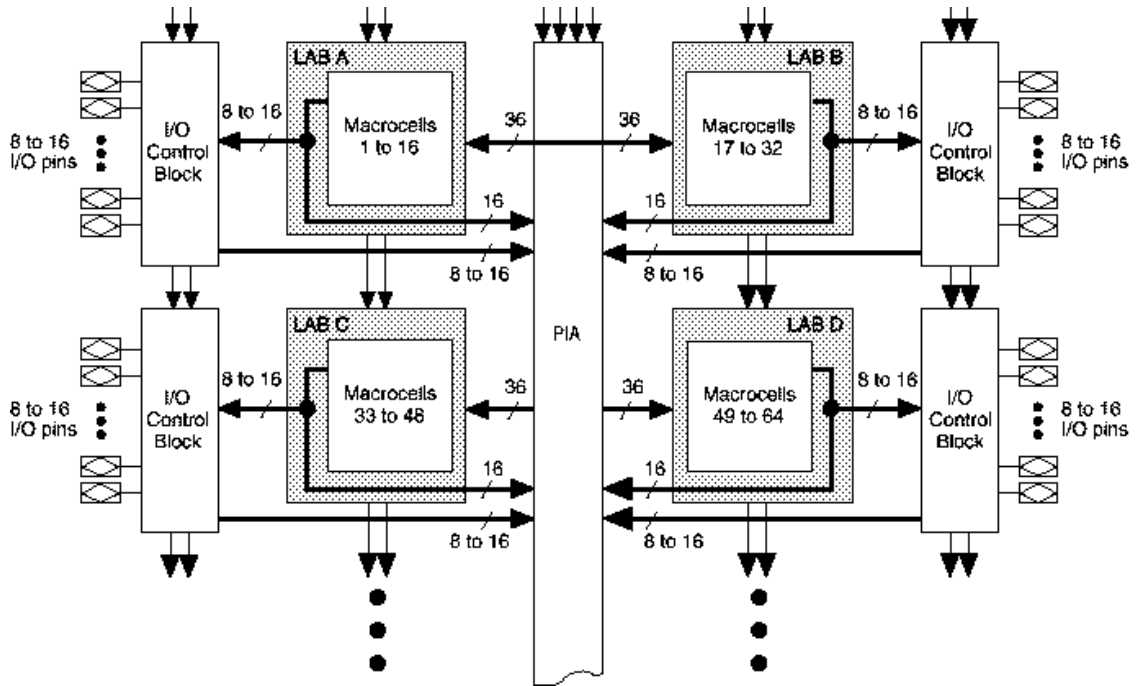
**3.2. CPLD (Complex Programmable Logic Device)**

La nécessité de placer de plus en plus de fonctions dans un même circuit a conduit tout naturellement à intégrer plusieurs PLD (blocs logiques) sur une même pastille, reliés entre eux par une matrice centrale.

Sur la figure suivante chaque bloc LAB (Logic Array Block) de 16 macrocellules est l'équivalent d'un PLD à 16 OLMC. Ils sont reliés entre eux par une matrice d'interconnexion (PIA pour Programmable Interconnect Array).

Un seul point de connexion relie entre eux les blocs logiques. Les temps de propagation d'un bloc à l'autre sont donc constants et prédictibles.

La phase de placement des différentes fonctions au sein des macrocellules n'est donc pas critique sur un CPLD, l'outil de synthèse regroupant au maximum les entrées sorties utilisant des ressources communes.



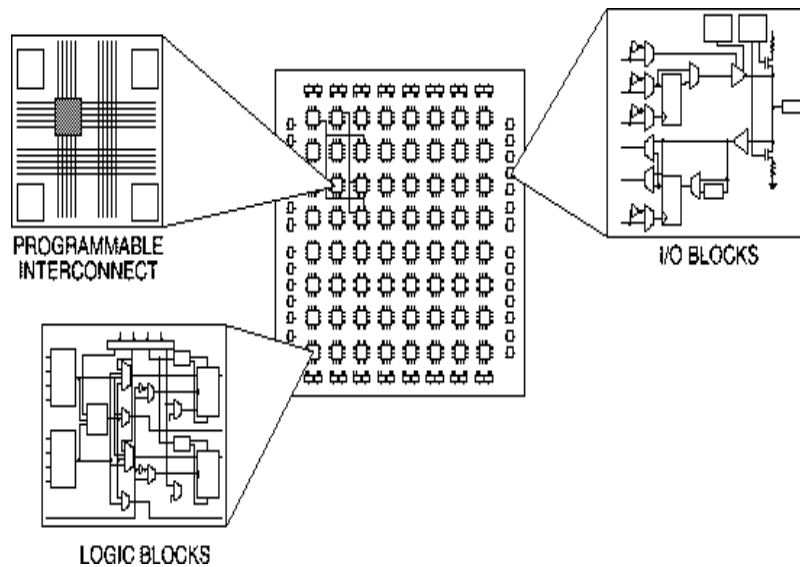
L'établissement des liaisons (routage) entre les différentes macrocellules est encore moins critique : un seul point de connexion -cause du retard- relie les LAB entre eux. Le temps de propagation des signaux est parfaitement prédictible avant que le routage ne soit fait. Ce dernier n'influence donc pas les performances du circuit programmé.

Dans l'outil de synthèse, la partie s'occupant du placement et du routage est appelée le "fitter" (to fit : placer, garnir).

La technologie de connexion utilisée est généralement l'EEPROM (proche de celle des PLD) ou EEPROM flash.

### 3.3. FPGA (Field Programmable Gate Array)

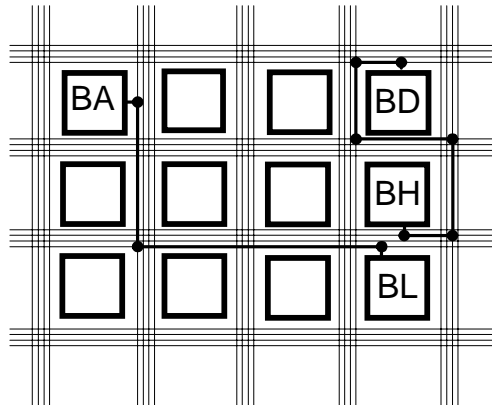
Les blocs logiques sont plus nombreux et plus simple que pour les CPLD, mais cette fois les interconnexions entre les blocs logiques ne sont pas centralisées.



Le passage d'un bloc logique à un autre se fera par un nombre de point de connexion (responsables des temps de propagation) fonction de la position relative des deux blocs logiques et de l'état "d'encombrement" de la matrice. Ces délais ne sont donc pas prédictibles (contrairement aux CPLD) avant le placement routage.

De la phase de placement des blocs logiques et de routage des connexions dépendront donc beaucoup les performances du circuit en terme de vitesse. Sur la figure suivante, pour illustrer le phénomène, on peut voir

- une liaison entre deux blocs logiques (BA et BL) éloignés, mais passant par peu de points de connexion, donc introduisant un faible retard,
- une liaison entre deux blocs proches (BD et BH) mais passant par de nombreux points de connexion, donc introduisant un retard important.



L'utilisation optimale des potentialités d'intégration d'un FPGA conduira à un routage délicat et pénalisant en termes de vitesse.

Il convient de noter que ces retards sont dus à l'interaction de la résistance de la connexion et de la capacité parasite ; cela n'a rien à voir avec un retard dû à la propagation d'un signal sur une ligne tel qu'on le voit en haute fréquence.

Ces composants permettent une forte densité d'intégration. La petitesse des blocs logiques autorise une meilleure utilisation des ressources du composant (au prix d'un routage délicat)

Il devient alors possible d'implanter dans le circuit des fonctions aussi complexes qu'un microcontrôleur. Ces fonctions sont fournies sous forme de programme par le constructeur du composant et appelées "megafonction" ou "megacore". Le terme générique classiquement utilisé pour les désigner est "propriété intellectuelle" ou IP (Intellectual Property).

Les FPGA utilisent généralement les technologies SRAM ou antifusible.

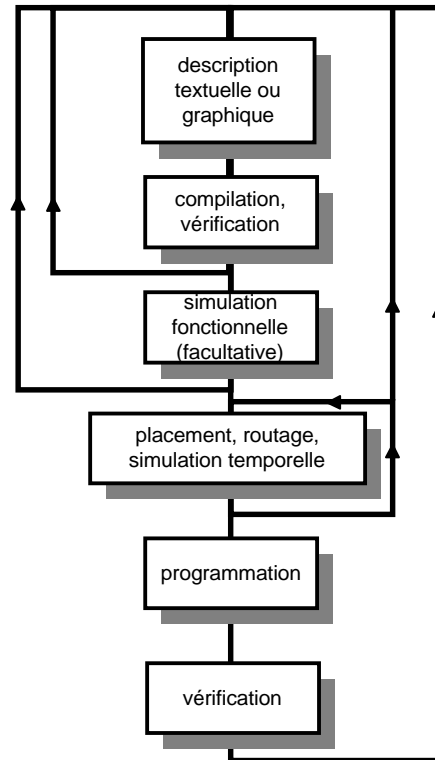
### **3.4. Les outils de développement des CPLD et FPGA**

Cet outil va permettre au concepteur de programmer le circuit à partir de la description de la fonction à réaliser. Cette description peut être textuelle (VHDL, Verilog) ou graphique (symboles de fonction, graphe d'état, chronogrammes).

La compilation va permettre dans un premier temps vérifier la cohérence de la description et la syntaxe du langage utilisé, puis d'effectuer une simulation fonctionnelle dans un premier temps

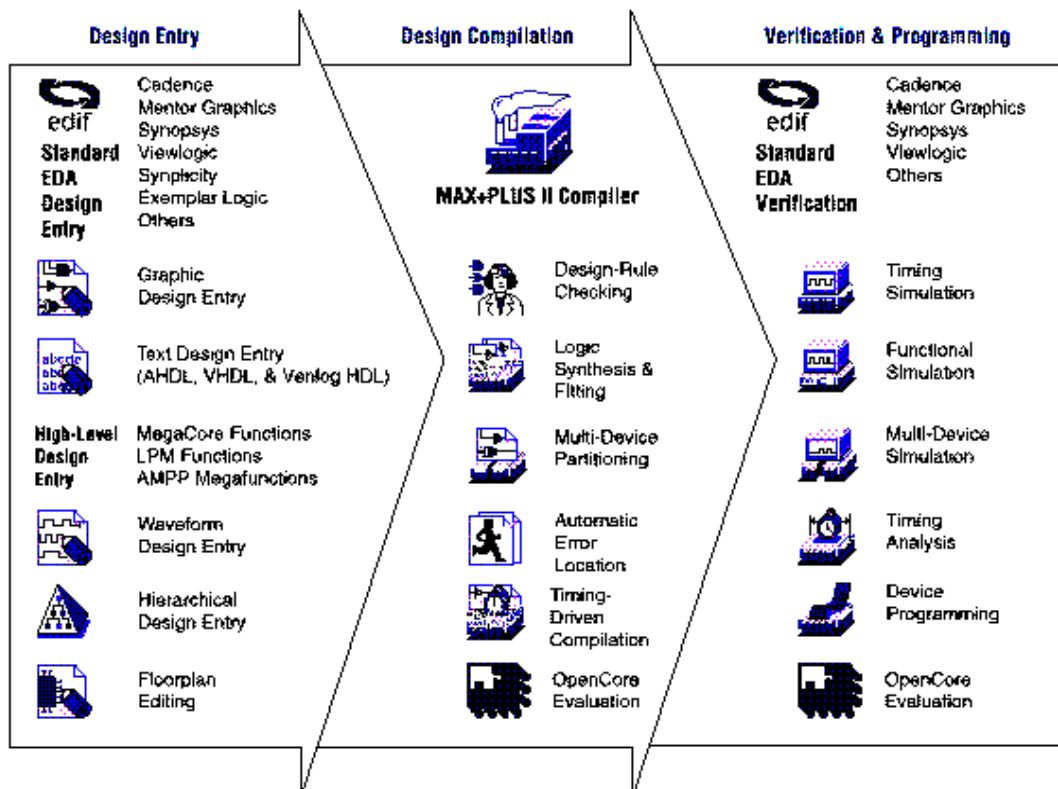
Après avoir choisi le circuit cible, le "fitter" effectue ensuite le placement et routage des blocs logiques. Comme nous l'avons vu cette opération est capitale pour une utilisation optimale d'un FPGA : un circuit performant ne sera pas utilisé de manière optimale si l'outil de synthèse est mauvais.

Vient enfin la programmation du circuit et la vérification du fonctionnement.



Le graphe ci-après donne une vue d'ensemble des différents outils utilisés (obligatoirement ou facultativement) par le concepteur. Il s'agit ici du logiciel Max+plus II de chez Altera.

La partie gauche de la figure énumère les divers outils d'entrée : les différents éditeurs dont nous avons parlé, avec comme alternative l'utilisation d'outils autre comme les logiciels d'autres marques (Synopsys, Mentors Graphics...). On remarque également la possibilité d'utiliser les IP, des outils d'aide au routage (Floorplan Editor) ou encore de gestion de projet (Hierachical Design Entry).



Suivent ensuite les outils de compilation, vérification, placement, routage, simulation fonctionnelle et temporelle et programmation.

Le "multi-device partitionning" consiste à répartir un projet sur 2 ou plusieurs circuits.

### **3.5. ASIC (Application Specific Integrated Circuit)**

Si les composants précédents pouvaient être développés avec un simple ordinateur, ceux que nous abordons maintenant nécessitent l'intervention d'un fondeur qui produira le circuit demandé à partir des masques fournis par son client. Ici encore, le terme programmable n'est pas des plus judicieux, les connexions entre les éléments étant dessinées sur les masques.

Les temps et coûts de productions sont importants. On distingue trois types d'ASIC classé par ordre croissant de configurabilité.

#### **3.5.1. Les prédiffusés (gate arrays)**

Ils contiennent une nébuleuse de transistors ou de portes à interconnecter avec les problèmes de routage et de délais que cela comporte.

#### **3.5.2. Les précactérisés (standard cell)**

On utilise cette fois des bibliothèques de cellules standards à placer sur le semi-conducteur

#### **3.5.3. Les "fulls customs"**

Ils sont entièrement définissables par le client. Ces circuits conduisent à la réalisation de tous les composants VLSI comme les microprocesseurs.

### **3.6. Comparaison et évolution**

Il est difficile de comparer les ASIC et les CPLD/FPGA. Les méthodes et temps de développement ne sont pas du tout les mêmes. L'utilisation des ASIC va surtout être justifié par la production en grande série du circuit à réaliser. Il faut cependant noter que très marginaux en 1990, les CPLD et FPGA ont pris en 2000 près de 95% du marché des ASIC.

En ce qui concerne les CPLD et FPGA uniquement, comme nous l'avons précisé au début, il est très difficile de donner des ordres de grandeurs et des éléments de comparaison dans un domaine qui évolue aussi rapidement, et où la concurrence entre les constructeurs a tendance à brouiller les pistes.

Les principaux critères de choix seront :

- la vitesse de fonctionnement
- le nombre de portes
- la consommation
- le prix

Un bon résumé de la situation présente les FPGA comme des circuits à forte densité d'intégration et les CPLD comme des circuits rapides. Le tableau ci-après donne les principales caractéristiques de chaque catégorie :

	circuits MSI (à titre de comparaison)	PLD	CPLD	FPGA
nombre de portes (ordre de grandeur)	100	150	40 000	4 000 000
vitesse de fonctionnement (ordre de grandeur)	100 MHz	200 MHz	240 MHz	100 MHz
technologie de connexion		MOS à grille flottante	MOS à grille flottante	SRAM et antifusible
codage des fonctions		PLA	PLA et LUT	LUT et MUX

La notion de nombre de portes supposant implicitement que toutes sont utilisées, on préfère souvent

parler de portes équivalent ou utilisables (usable gates) pour caractériser la densité d'un circuit. Il existe un rapport 2 à 4 entre ces deux termes.

L'utilisation des propriétés intellectuelles (IP) gratuite ou payante se généralise de plus en plus, et permet facilement l'intégration de fonctions complexes comme les microprocesseurs au sein des circuits. Ces fonctions se présentent comme un programme, optimisé ou non pour un composant. De même, à l'intérieur des FPGA on trouve maintenant des zones optimisées pour implanter de la mémoire (20 Kbits dans un FLEX10K par exemple) et des microprocesseurs.

L'utilisation de FPGA SRAM reconfiguré en cours d'utilisation pour s'adapter à l'évolution de l'environnement devient également courante.

## **Bibliographie**

Logique programmable par L. Dutrieux et D. Demigny chez Eyrolles

Sites constructeurs : <http://www.altera.com> , <http://www.xilinx.com> , <http://www.latticesemi.com> , <http://www.actel.com> .

Le site de J Weiss <http://www.supelec-rennes.fr/ren/rd/etscm/base>

Le site du centre national de formation en micro-électronique <http://www.cnfm.fr> et en particuliers l'adresse <http://www.cnfm/altera/index/index.html>

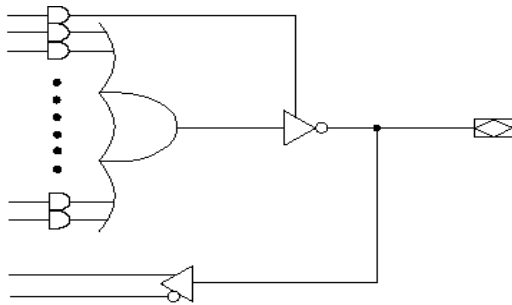
Les circuits programmables : faisons le point sur

<http://www.electron.cndp.fr/documents/ressources/contributions/pellet.azoulay/pld.exe>

### **Annexe 1 : GAL 22V10**

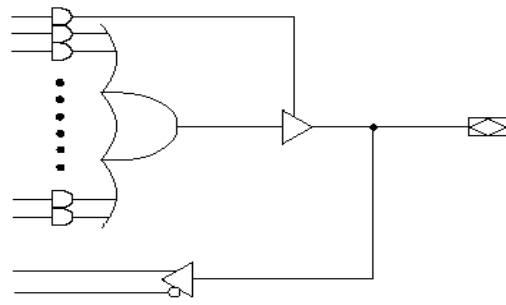
Ce circuit présente 10 OLMC (output logic macrocell) reliés par un PLA à matrice OU programmable. Le délais de propagation des signaux est de 4 ns, autorisant une fréquence de fonctionnement jusqu'à 230 MHz. L'alimentation se fait en 5 V ou 3,3 V suivant les versions.

Les OLMC peuvent être programmées, suivant les niveaux imposés aux multiplexeurs (voir chapitre sur les PLD) pour fonctionner en combinatoire,



**ACTIVE LOW**

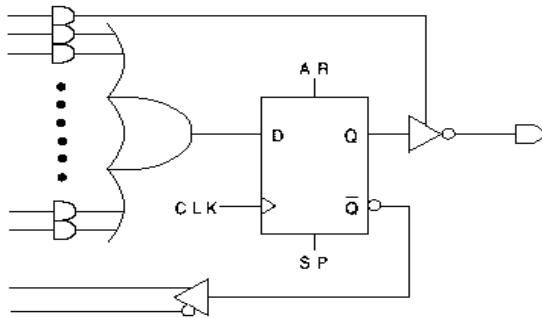
$S_0 = 0$   
 $S_1 = 1$



**ACTIVE HIGH**

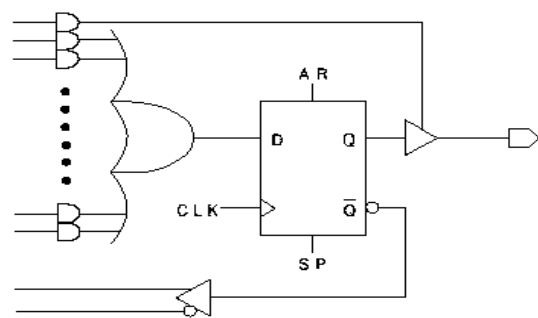
$S_0 = 1$   
 $S_1 = 1$

ou en séquentielle,



**ACTIVE LOW**

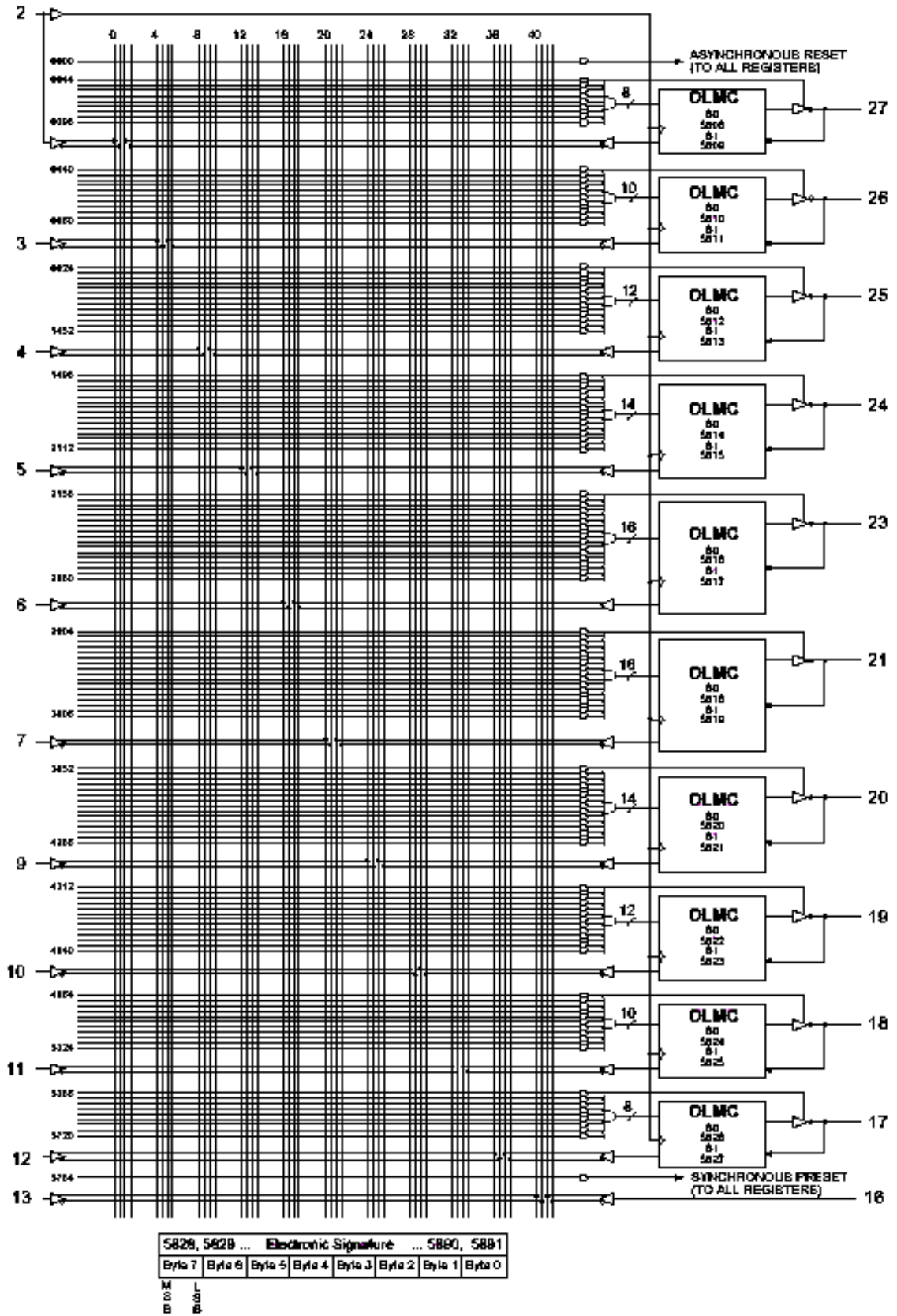
$S_0 = 0$   
 $S_1 = 0$



**ACTIVE HIGH**

$S_0 = 1$   
 $S_1 = 0$

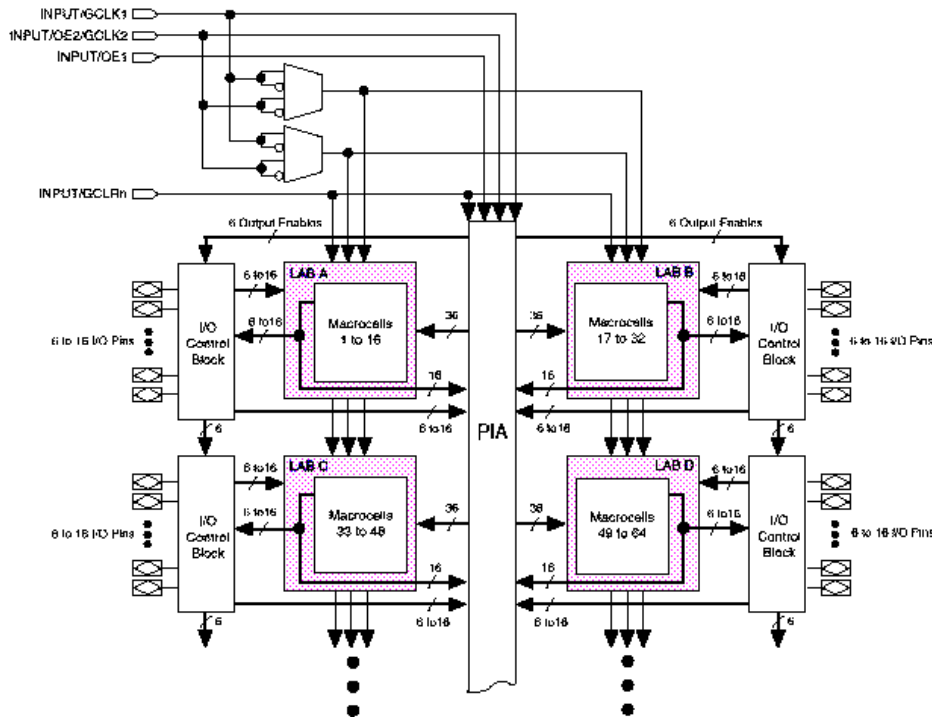
La figure suivante donne une vue d'ensemble du circuit :



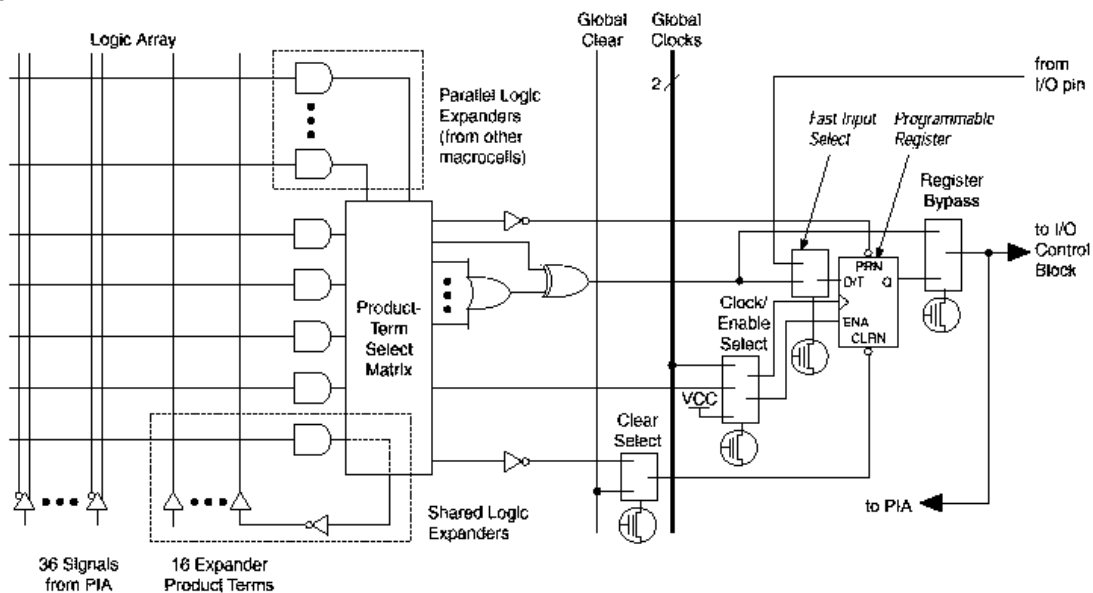
**Annexe 2 : Famille MAX7000 de CPLD de chez Altera**

Cette famille mise sur le marché en 92 comprend des circuits de 600 à 20 000 portes (600 à 5000 portes utilisables). Elle utilise une technologie EEPROM, et autorise pour les circuits les plus rapides des temps de propagations broche à broche de 5 ns, permettant un fonctionnement en compteur jusqu'à 175 MHz.

On retrouve sur la structure interne un ensemble de blocs logiques (LAB: logic array block) interconnectés par une matrice (PIA :programmable interconnect array).

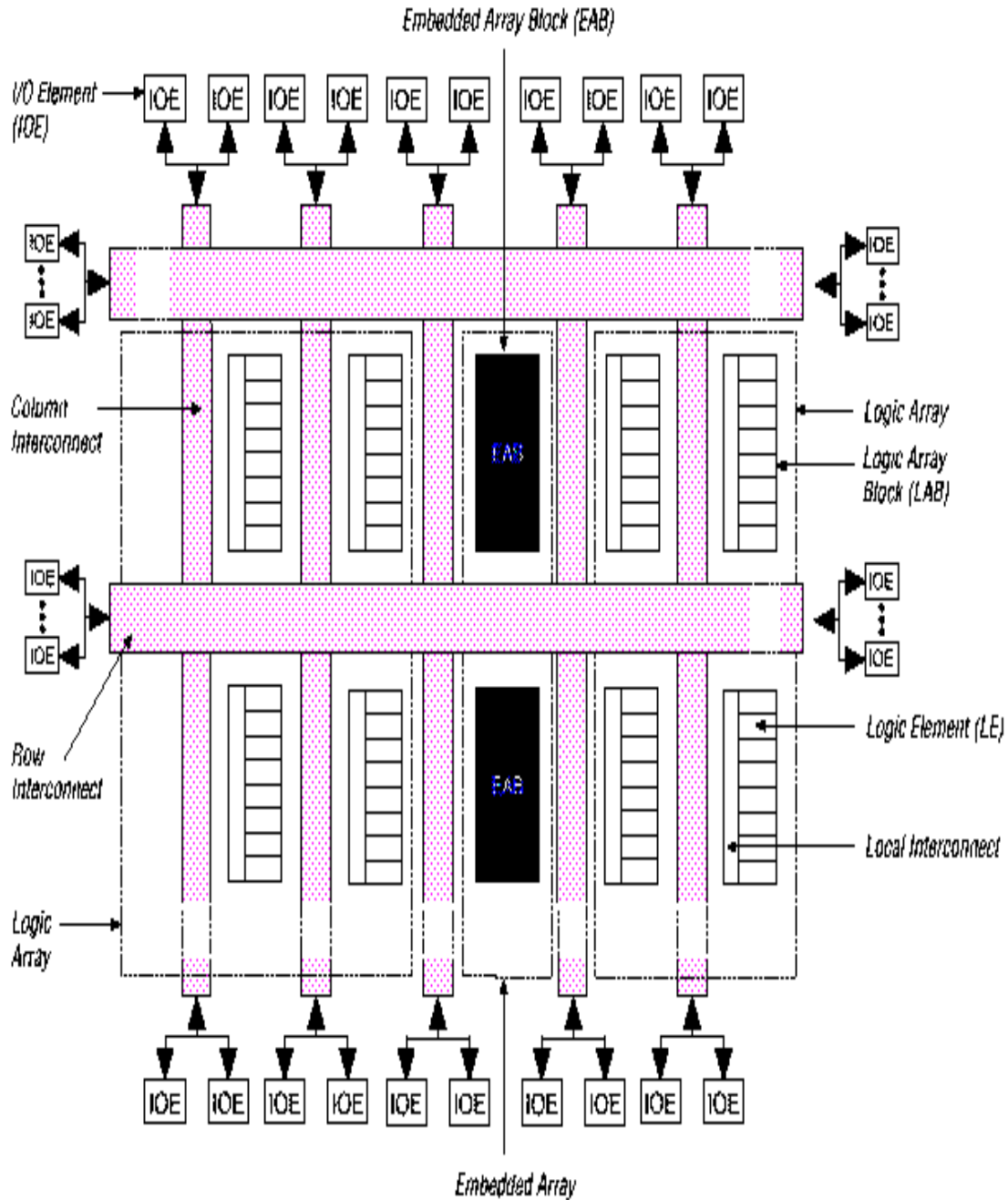


Chaque LAB est très similaire au PLD précédent et comprend 16 macrocellules, chacune étant constituée d'une matrice ET programmable, d'une matrice OU fixe et d'une bascule configurable en sortie.

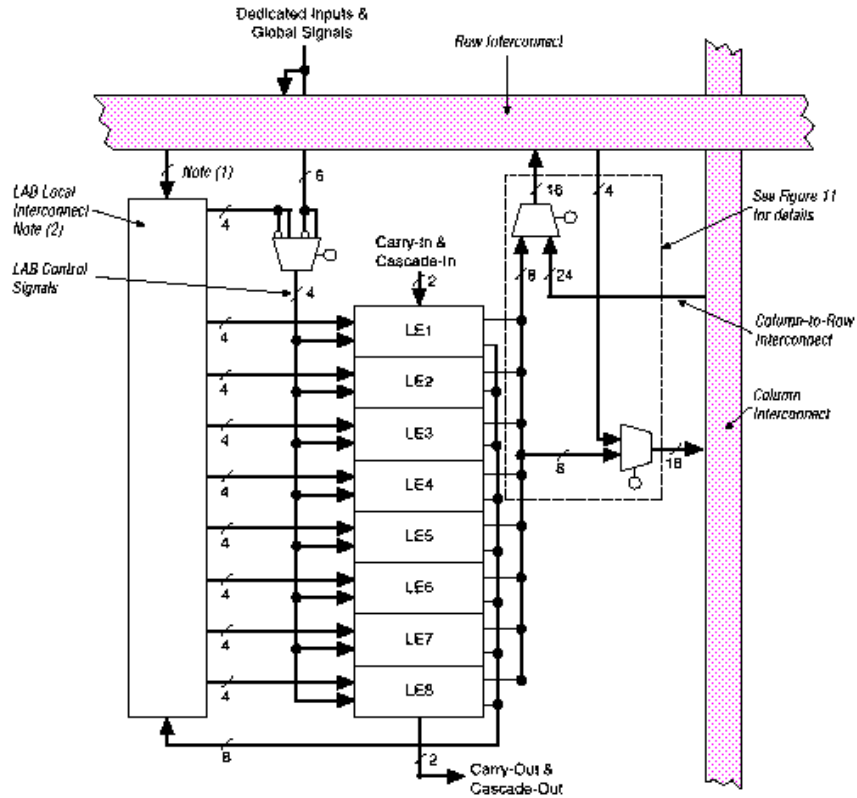


**Annexe 3 : famille FLEX10K de chez Altera, un hybride CPLD, FPGA**

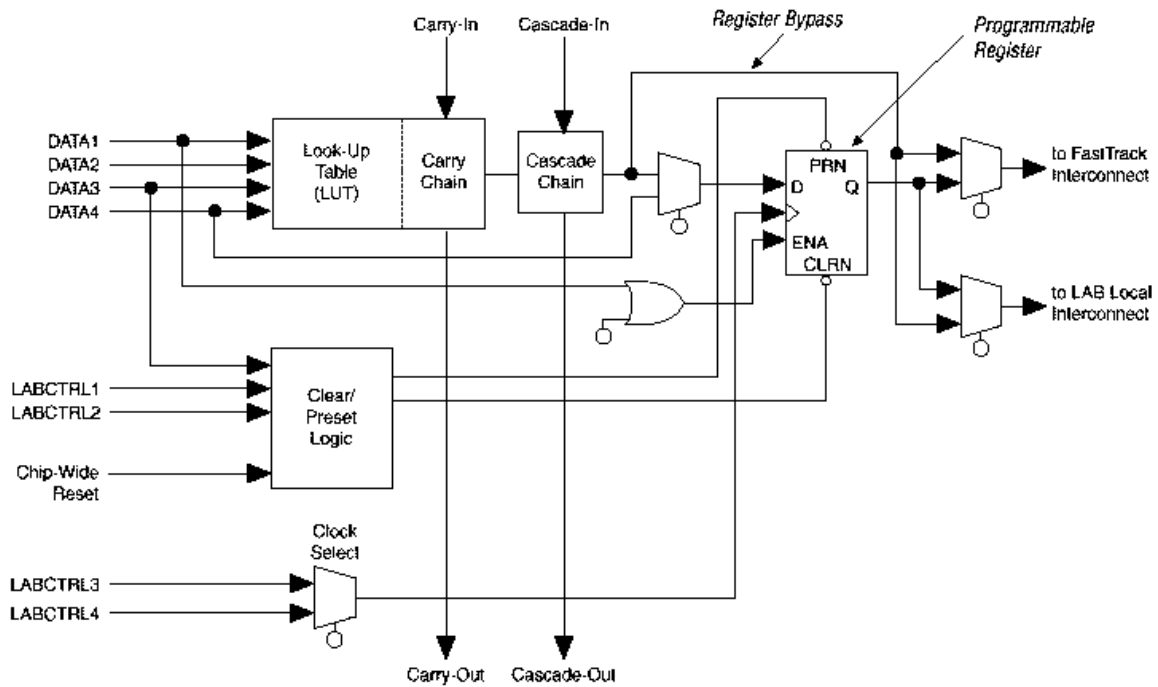
Les circuits de cette famille intègrent de 10 000 à 250 000 portes et jusqu'à 40 Kbits de RAM, le tout en technologie SRAM. La configuration étant volatile, le composant doit être associé à une PROM série ou parallèle. La structure du circuit ressemble beaucoup à un CPLD classique, une matrice d'interconnexion reliant des blocs logiques.



Comme pour les CPLD, les retards dus aux interconnexions de blocs logiques sont prédictibles. Par contre, comme un FPGA le circuit contient une partie dédiée à l'intégration de RAM, les EAB (Embedded Array Blocks), et comme un FPGA les connexions sont réalisées en technologie SRAM. La figure suivante représente un LAB.

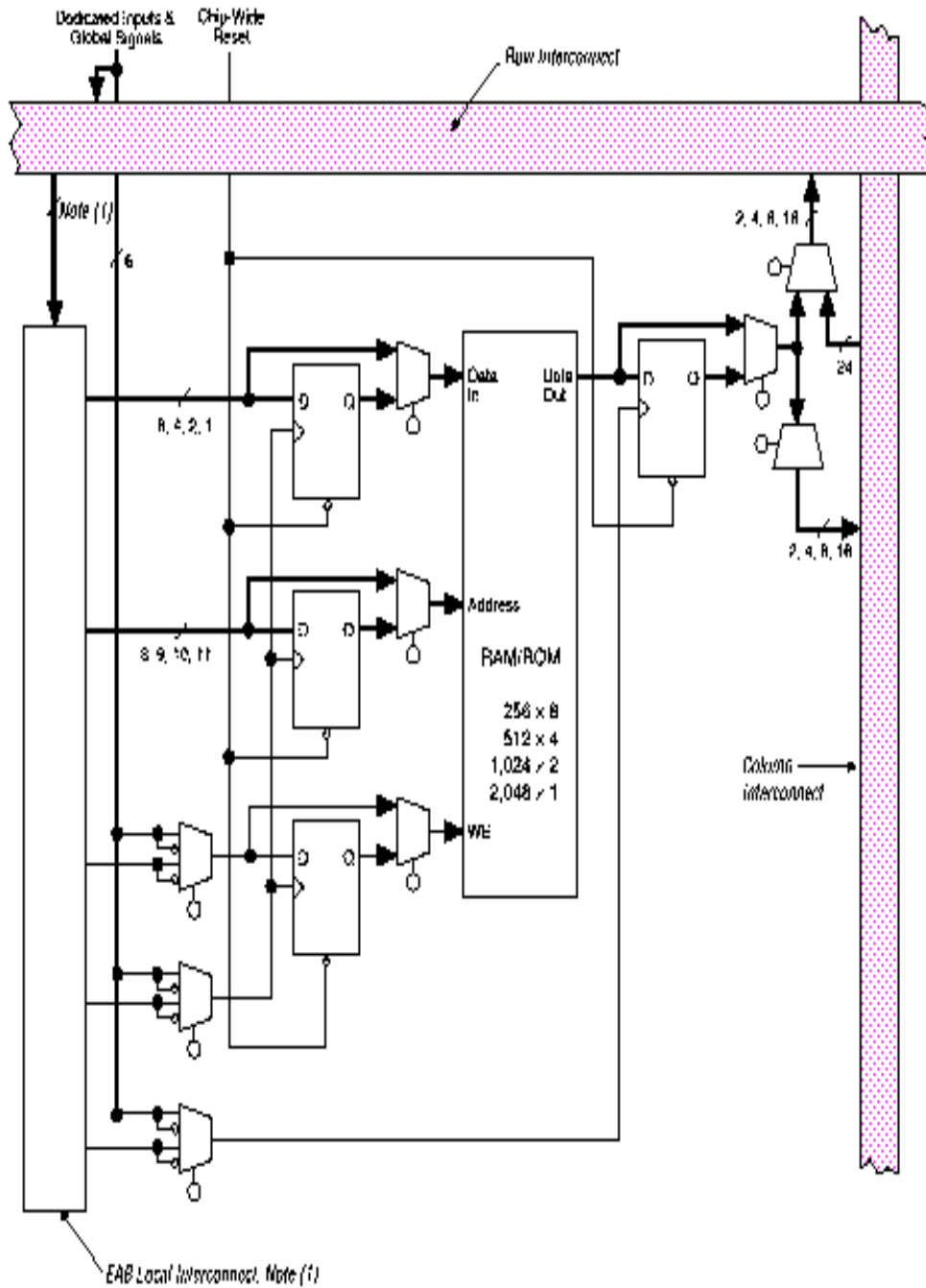


Chaque LAB contient 8 éléments logiques (LE), un de ces éléments étant décrit ci-après :



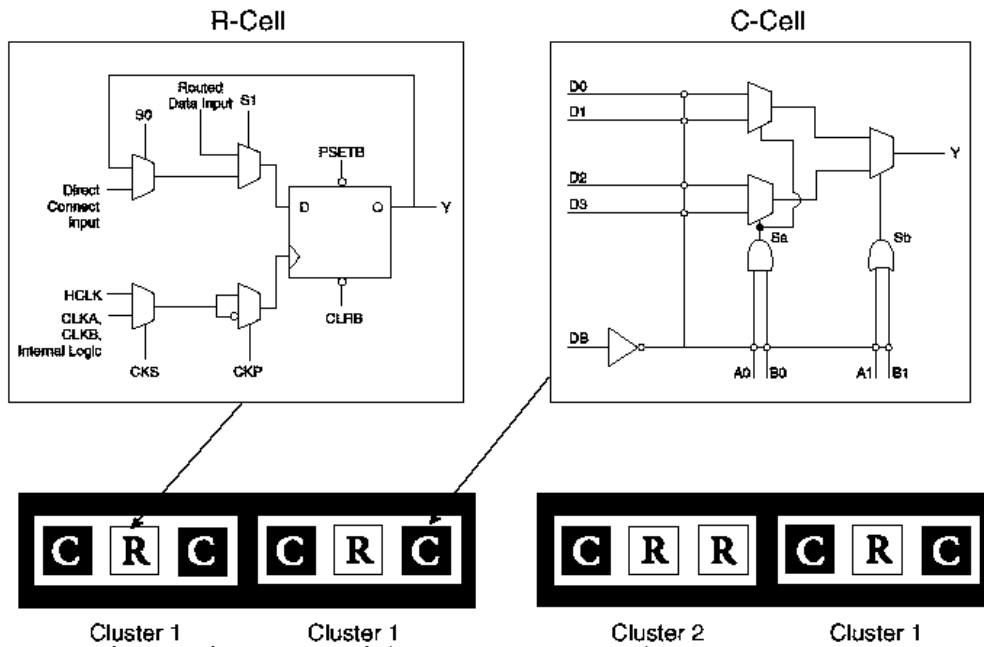
Comme on peut le voir, la partie combinatoire est réalisée par une petite mémoire à 4 bits d'adresse et 1 bit de donnée (nommée LUT), celle-ci attaquant une bascule.

La figure suivante représente un EAB destiné à l'implantation de la RAM, ce dernier pouvant aussi être utilisé pour la réalisation d'autres fonctions.

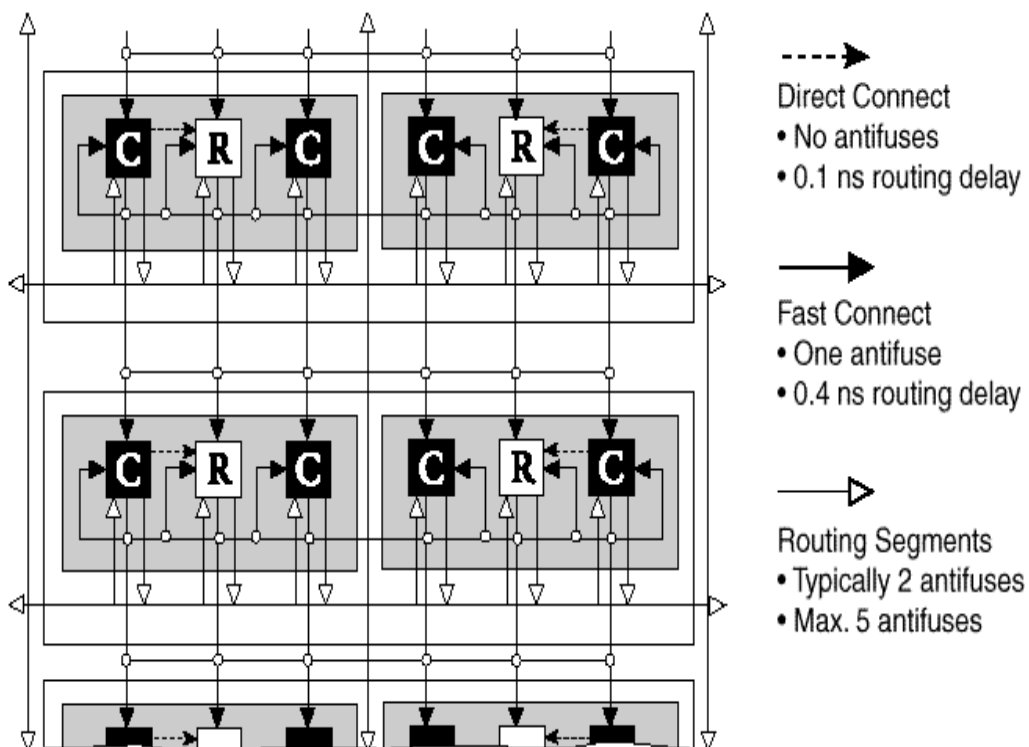


**Annexe 4 : famille 54SX de FPGA de chez Actel**

Il s'agit d'une famille de FPGA à antifusibles, intégrant jusqu'à 40 000 portes et pouvant fonctionner jusqu'à 320 MHz en interne (les plus rapides étant les moins denses).  
 Les modules logiques de base sont extrêmement simple, certains réalisant une fonction combinatoire (modules notés C), d'autres une fonction séquentielle (modules notés S). Les fonctions combinatoires sont réalisées au moyen de multiplexeurs 2 vers 1.



Ces modules sont regroupés en "cluster" de deux types suivant l'association réalisée. Les clusters sont interconnectés entre eux par des liaisons à antifusible.



## **Annexe 5 : Lexique**

**ABEL** : langage de programmation des circuits de faible densité d'intégration.

**ASIC (Application Specific Integrated Circuit)** : circuit non programmable configuré lors de sa fabrication pour une application spécifique.

**CPLD (Complex Programmable Logic Device)** : circuit intégrant plusieurs PLD sur une même pastille.

**EEPROM ou E2PROM (Electrical Erasable Programmable Read Only Memory)** : mémoire ROM programmable et effaçable électriquement.

**E2PAL (Electrical Erasable PAL)** : voir GAL

**EPLD (Erasable PLD)** : voir GAL.

**EPROM (Erasable PROM)** : PROM effaçable par UV.

**Flash EEPROM** : EEPROM utilisant 2 transistors par point mémoire ; utilisé pour les connexions dans les CPLD.

**FPGA (Field Programmable Logic Array)** : réseau programmable à haute densité d'intégration.

**FPLD (Fiel Programmable Logic Device)** : terme générique pour les CPLD et FPGA.

**FPLS (Fiel Programmable Logic Sequencer)** : ancien nom des PAL à registre.

**GAL (Generic Array Logic)** : PLD programmable et effaçable électriquement

**ISP (In Situ Programmable)** : caractérise un circuit reprogrammable sur l'application.

**JEDEC** : organisme de normalisation, donnant son nom aux fichiers de programmation des PLD.

**LSI (Large Square Integration)** : circuits intégrant quelques centaines à quelques milliers de transistors.

**LUT (Lock Up Table)** : nom donné aux cellules mémoire réalisant les fonctions combinatoires dans les CPLD et FPGA.

**MSI (Medium Square Integration)** : circuits intégrant quelques centaines de transistors.

**MUX** : abréviation pour multiplexeur

**PAL (Programmable Array Logic)** : ancien nom des PLD.

**PLA (Programmable Logic Array)** : réseau à matrice ET et OU permettant la réalisation de fonctions combinatoires.

**PLD (Programmable Logic Device)** : circuit logique programmable intégrant une matrice ET programmable, une matrice OU fixe et plusieurs cellules de sortie.

**PROM (Programmable Read Only Memory)** : mémoire ROM programmable.

**SPLD (Simple PLD)** : par opposition aux FPLD, voir PLD.

**SRAM (Static Random Access Memory)** : technologie utilisée pour les connexions dans les CPLD et FPGA.

**SSI (Small Square Integration)** : circuits intégrant quelques portes.

**Verilog** : langage de synthèse des circuits numériques

**VHDL (Very high speed or scale integrated circuits Hardware Description Language)** : langage de modélisation et de synthèse des circuits numérique.