

## 7. Mémoires vives statiques et dynamiques. Mémoires spéciales et protocoles d'échange (SPI, I2C, Microwire, 1-Wire).

### 7.1. Mémoires vives statiques et dynamiques.

La mémoire vive c'est la mémoire (*mémoire opérationnelle + mémoire de cache + mémoire morte*) avec laquelle le processeur puisse fonctionner directement. Cette mémoire sert de stocker tous les programmes exécutés dans un moment concret de l'ordinateur, les données traitées de ces programmes et les programmes de systèmes, nécessaires pour la gestion des processus calculateurs utilisés et assurance micro programmable (sur la mémoire morte/rémanente/non volatile).

Par rapport à la cellule principale construisant la mémoire vive on distingue :

#### 7.1.1. Mémoire vive statique :

Cette mémoire vive statique est avec un accès aléatoire d'écrire et lire. C'est une mémoire semi-conductrice, construite par sa cellule principale de mémoire - la bascule. Elle est utilisée pour la mémoire opérationnelle de l'ordinateur et aussi bien pour la mémoire de cache. Ses caractéristiques fonctionnelles comme *Rapidité*, *Capacitance* et *Densité* sont plus basses que chez les mémoires vives dynamiques.

#### 7.1.2. Mémoire vives dynamique:

Cette mémoire vive dynamique est aussi avec un accès aléatoire d'écrire et lire. C'est une mémoire aussi semi-conductrice, construite par sa cellule principale de mémoire - la capacitance de quelques *pF*, réalisé technologiquement par une *p-n* transition ou une couche semi-conductrice du type *MOS*. Elle est utilisée comme partie principale de la mémoire opérationnelle de l'ordinateur et aussi bien pour la mémoire de cache. Ses caractéristiques fonctionnelles comme *Rapidité*, *Capacitance* et *Densité* sont beaucoup plus hautes que chez les mémoires vives dynamiques.

### 7.2. Mémoires spéciales et protocoles d'échange (SPI, I2C, Microwire, 1-Wire).

#### 7.2.1. Mémoires spéciales:

Ce sont toutes sortes de cartes de mémoire, utilisées dans les différents dispositifs : photo cameras, phones mobiles, tablettes, jeux électroniques, etc. Actuellement leur capacitance commence de 1Gbyte to 128 Gbytes.

#### 7.2.2. Protocoles d'échange SPI:

➤ Une liaison *SPI* (pour *Serial Peripheral Interface*) est un bus de données série synchrone baptisé ainsi par **Motorola**, qui opère en mode *Full-duplex*. Les circuits communiquent selon un schéma maître-esclaves, où le maître s'occupe totalement de la communication. Plusieurs esclaves peuvent coexister sur un même bus, dans ce cas, la sélection du destinataire se fait par une ligne dédiée entre le maître et l'esclave appelée *chip select*.

➤ *Le bus SPI* utilise 4 signaux logiques :

- **SCLK** — Serial Clock, Horloge (généralisé par le maître)
- **MOSI** — Master Output, Slave Input (généralisé par le maître)
- **MISO** — Master Input, Slave Output (généralisé par l'esclave)
- **SS** — Slave Select, Actif à l'état bas (généralisé par le maître).

➤ *Fonctionnement du SPI* :

Une transmission SPI typique est une communication simultanée entre un maître et un esclave :

- Le maître génère l'horloge et sélectionne l'esclave avec qui il veut communiquer par l'utilisation du signal SS
- L'esclave répond aux requêtes du maître

À chaque coup d'horloge le maître et l'esclave s'échangent un bit. Après huit coups d'horloges le maître a transmis un octet à l'esclave et vice versa. La vitesse de l'horloge est réglée selon des caractéristiques propres aux périphériques.

➤ *Avantages*

- Communication **Full duplex**
- Débit assez important par rapport à I<sup>2</sup>C
- Flexibilité du nombre de bits à transmettre ainsi que du protocole en lui-même.
- Simplicité de l'interface matérielle
  - Aucun arbitre nécessaire car aucune collision possible
  - Les esclaves utilisent l'horloge du maître et n'ont donc pas besoin d'oscillateur propre
  - Pas de la couche physique du modèle OSI nécessaire
- Partage d'un bus commun pour l'horloge, MISO et MOSI entre les périphériques

➤ *Inconvénients*

- Monopolise plus de broches d'un boîtier que l'I<sup>2</sup>C ou une UART qui en utilisent seulement deux.
- Aucun adressage possible, il faut une ligne de sélection par esclave en mode non chaîné.
- Le protocole n'a pas d'acquittement. Le maître peut parler dans le vide sans le savoir.
- La plupart des implémentations ne tolèrent la présence que d'un seul maître SPI sur le bus. Néanmoins, on trouve des circuits intégrés supportant le mode "multi-master", permettant de partager le bus SPI entre plusieurs maîtres. Les implémentations pratiques de ce genre restent rares.
- Ne s'utilise que sur de courtes distances contrairement aux liaisons RS-232, RS-485 ou bus CAN (*Controller Area Network* est un bus système série très répandu dans beaucoup d'industries, notamment l'automobile. Il a été normalisé avec la norme **ISO 11898**).
- Néanmoins, il est possible d'utiliser des tampons de bus, comme des adaptateurs RS-232 (exemple : MAX208), RS485 pour relier un maître et un esclave avec un câble de quelques mètres. La mise en œuvre de ces circuits reste délicate car il devient nécessaire de procéder à des adaptations d'impédance pour éviter les oscillations parasites.

➤ *Outils de développement*

Il existe une grande variété d'instruments contrôlés par PC permettant d'agir comme maître et/ou analyseur de protocole SPI. Ces outils peuvent parfois être utilisés pour des protocoles série dérivant du protocole SPI, comme les protocoles sur 3 fils.

### 7.2.3. Protocoles d'échange I<sup>2</sup>C:

➤ La norme I<sup>2</sup>C (*Inter Integrated Circuit*) a été développée par **Philips** en 1982. Depuis elle est maintenue par NXP (ex-division semi-conducteurs de Philips). La version 1.0 a été publiée en 1992, la version 2.0 en 1998, la version 2.1 en 2000, la version 3.0 en 2007 et la version actuelle 4.0 en 2012. I<sup>2</sup>C est un bus de données qui a émergé de la « guerre des standards » lancée par les acteurs du monde électronique. Conçu par Philips pour les applications de *domotique* et d'électronique domestique, il permet de relier facilement un microprocesseur et différents circuits, notamment ceux d'une télévision moderne : récepteur de la télécommande, réglages des amplificateurs basses fréquences, tuner, horloge, gestion de la prise péritel, etc.

(La **domotique** est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans les bâtiments, plus ou moins « interopérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prises électriques, etc.). La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut retrouver dans les maisons, les hôtels, les lieux publics, etc.)

Il existe d'innombrables périphériques exploitant ce bus, il est même implémentable par logiciel dans n'importe quel microcontrôleur. Le poids de l'industrie de l'électronique grand public a permis des prix très bas grâce à ces nombreux composants. Ce bus porte parfois le nom de TWI (Two Wire Interface) chez certains constructeurs.

➤ *Le bus I<sup>2</sup>C* est réalisée par l'intermédiaire de 2 lignes actives:

- **SDA** - Serial Data Line: ligne de données bidirectionnelle,
- **SCL** - Serial Clock Line: ligne d'horloge de synchronisation bidirectionnelle.

Il ne faut également pas oublier la masse qui doit être commune aux équipements.

I<sup>2</sup>C est un bus *série synchrone bidirectionnel half-duplex*. Plusieurs équipements, soit maîtres, soit esclaves, peuvent être connectés au bus.

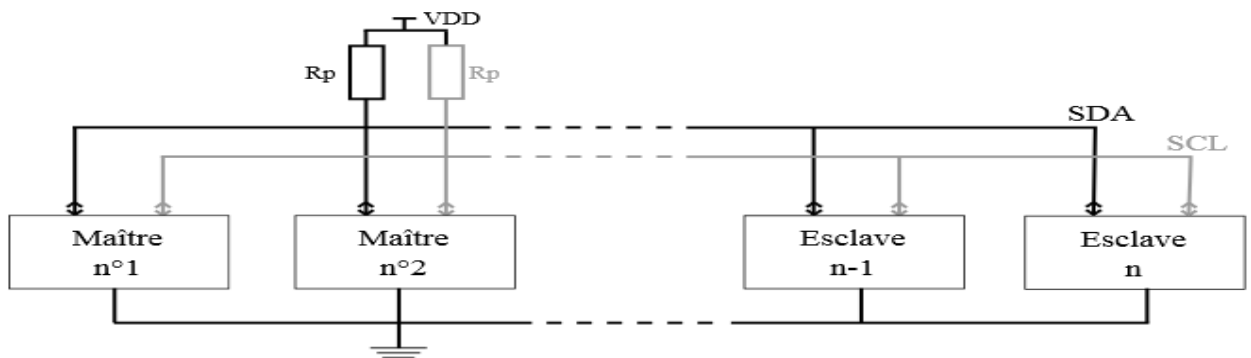


Figure 22. Protocoles d'échange I<sup>2</sup>C

Les 2 lignes sont tirées au niveau de tension  $V_{DD}$  à travers des résistances de pull-up ( $R_P$ ). Le nombre maximal d'équipements n'est pas spécifié, mais dépend de la capacité du bus ( $C_B$ ) maximum (dont dépend la vitesse maximale du bus).

➤ *Fonctionnement de l'I<sup>2</sup>C:*

Les échanges ont toujours lieu entre un seul maître et un (ou tous les) esclave(s), toujours à l'initiative du maître (jamais de maître à maître ou d'esclave à esclave). Cependant, rien n'empêche un composant de passer du statut de maître à esclave et réciproquement.

Les équipements connectés au bus le sont par des sorties de type drain ouvert (ou collecteur ouvert) sur les 2 lignes SDA et SCL. Les équipements sont donc câblés sur le bus par le principe du « ET câblé », ce qui veut dire qu'en cas d'émission simultanée de 2 équipements, la valeur 0 écrase la valeur 1.

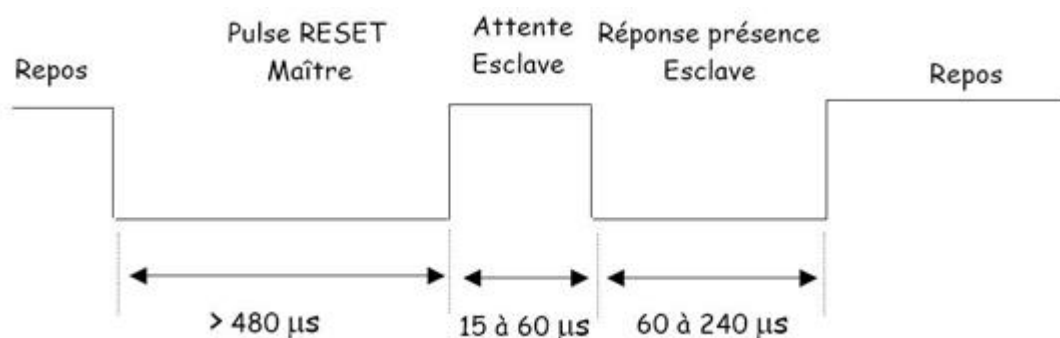
On dit donc :

- que l'état logique « 0 » ou « LOW » est l'état « dominant »,
- que l'état logique « 1 » ou « HIGH » est l'état « récessif ».

Lorsque le bus n'est pas utilisé, il est au niveau haut (à travers les résistances de pull-up  $R_P$ ).

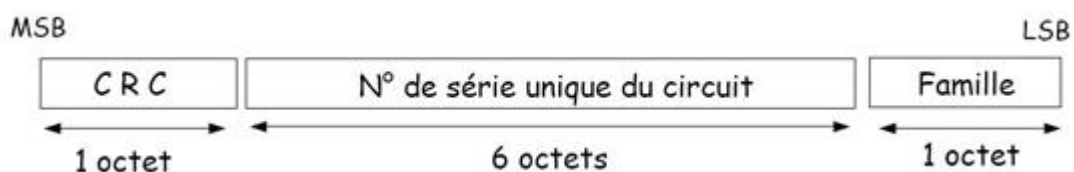
Il est généralement utilisé en domotique pour des thermomètres ou autres instruments de mesure météorologiques. Il est également très utilisé dans les circuits de gestion de l'énergie dans les batteries d'équipements électroniques et les chaînes d'onduleurs Ce système de bus utilise un seul maître, qui pourra dialoguer avec un ou plusieurs esclaves, mais ce protocole ne peut être utilisé entre deux appareils à la fois. Toutes les commandes et données sont envoyées avec le bit LSB en tête. L'état repos du bus est donc un état haut.

Avant toute communication, le maître met le bus à l'état bas (0V) pendant 480  $\mu s$  pour faire une initialisation (reset) des composants connectés. Après un délai de 15 à 60  $\mu s$ , le maître ou les esclaves connectés, forcent le bus à l'état bas pendant 60 à 240  $\mu s$  pour signaler leur présence. Le maître est alors informé des esclaves connectés sur le bus.



**Figure 23. Fonctionnement d'I<sup>2</sup>C**

Chaque circuit possède une adresse physique unique, gravée dans la puce à la fabrication. Cette adresse est constituée de 64 bits soit 8 octets. Le premier octet détermine le type de famille auquel appartient le circuit. Les 6 octets suivants, constituent le code propre du circuit. Le dernier octet est le CRC. C'est un octet de contrôle calculé à partir des 56 bits précédents.



**Figure 24. Adresse physique d'I²C**

Toute transaction entre un maître et un ou plusieurs esclaves, débute par une initialisation, constituée par l'envoi du pulse de Reset par le maître. Le maître doit ensuite envoyer une commande de type ROM qui est propre au *protocole 1-Wire*, et que tous les circuits de ce type vont reconnaître. Cela va permettre entre autre de sélectionner un circuit parmi les différents esclaves qui ont répondu présents au pulse de Reset. Le dialogue et l'échange de données pourra ensuite commencer, entre le maître et l'esclave sélectionné.

➤ *Temps et vitesses*

Il existe 5 vitesses de transmission :

- « *Standard mode* »  $\leq 100 \text{ kbit/s}$ ,
- « *Fast mode* »  $\leq 400 \text{ kbit/s}$ ,
- « *Fast plus mode* »  $\leq 1 \text{ Mbit/s}$ ,
- « *High-speed mode* »  $\leq 3,4 \text{ Mbit/s}$ ,
- « *Ultra-fast mode* »  $\leq 5 \text{ Mbit/s}$ .

**7.2.4. Protocoles d'échange MicroWire:**

➤ *MicroWire*, développé par National Semi-Conducteur est synchrone, maître type /esclave du protocole série. Ce protocole peut prendre en charge plusieurs périphériques esclaves, mais en raison de sa vitesse maximale lente, communément n'a qu'un seul esclave. La vitesse maximale est inférieure à 625 Kbps et la distance maximale recommandée entre le maître et l'esclave est au moins 3 m. Ce protocole prend en charge le mode *full-duplex 3Mbps*, série standard d'interface 3 fils définie par National Semi-Conducteur. Le protocole de MicroWire est essentiellement un sous-ensemble de la *SPI* interface. MicroWire est une série I / O Port sur microcontrôleurs, de sorte que le bus MicroWire sera également disponible sur EEPROM et autres puces périphériques.

➤ *Le bus MicroWire* est réalisée par l'intermédiaire de 2 lignes actives:

- *TR*, la ligne d'entrée en série au microcontrôleur, est donc la ligne de sortie en série, et
- *SK* est la ligne d'horloge en série. L'information est déplacée sur le front descendant de *SK*, et est évaluée sur le front montant.

Il ne faut également pas oublier la masse qui doit être commune aux équipements.

*SI* est décalée sur le front montant de *SK*. Le protocole est un bit d'état «1» suivi d'un mot de longueur variable instruction / adresse, suivi d'un champ de données de longueur variable. Ou l'appareil peut envoyer aucun bit de départ, juste un flux de bits de longueur variable.

Le temps de cycle d'horloge minimale est  $2\mu\text{S}$ ,  $1\mu\text{S}$  haut et bas  $1\mu\text{S}$ . La vitesse d'horloge est fonction du microcontrôleur, des vitesses typiques sont de 0 Hz à 250 kHz.

➤ Une amélioration supplémentaire de bus est appelé *MicroWire / Plus*. La principale différence entre les deux bus semble être que *MicroWire / Plus* architecture dans le microcontrôleur est plus complexe.

#### 7.2.5. Protocoles d'échange 1-Wire:

➤ Le protocole série *1-Wire* est asynchrone et doit avoir un maître et un esclave. Il est **half-duplex** et a une vitesse maximale de 16 Kbps. La distance maximale entre les deux appareils est de 300 m. Ce protocole ne peut être utilisé entre deux appareils à la fois.

➤ Le bus *1-Wire* permet de connecter et de faire dialoguer entre eux (en série, parallèle ou en étoile) des composants (des circuits) avec *seulement un fil de données*. Le fil unique du bus doit être tiré au +Vcc par une résistance de  $4,7\text{K}\Omega$ .

Le second fil est le fil de masse.

➤ *1-Wire* (aussi connu sous le nom de *bus Dallas* ou *OneWire*) est un bus conçu par **Dallas Semiconductor** qui de nature est similaire à I<sup>2</sup>C, il présente cependant des vitesses de transmission et un coût inférieurs.

## 8. Mémoires mortes - principes de sauvegarde de l'information. Programmation. PLD, FPGA.

### 8.1. Mémoires mortes - principes de sauvegarde de l'information. Programmation

Selon l'indice technologique et le moyen d'écriture / lire, les mémoires se divisent en:

**8.1.1. SRAM** – *Static Random Access Memory*: mémoire statique à l'accès aléatoire pour écrire et lire. C'est une mémoire semi-conductrice, construite de la cellule principale (de base) – la bascule. Elle est utilisée comme une partie de la mémoire opérationnelle de l'ordinateur, aussi bien dans les mémoires de cache.

**8.1.1. DRAM** – *Dynamic Random Access Memory*: mémoire dynamique à l'accès aléatoire pour écrire et lire. C'est une mémoire semi-conductrice, construite de la cellule principale (de base) – la capacitance, réalisée technologiquement comme  $p - n$  transfert ou comme part semi – conductrice **MOS**. Elle est utilisée comme une partie principale de la mémoire opérationnelle de l'ordinateur, aussi bien dans les mémoires super – rapide de cache.

**8.1.1. ROM** – *Read - Only Memory*: mémoire seulement de lire. C'est aussi une mémoire semi – conductrice. Elle est utilisée comme mémoire morte du processeur central. Par différence des SRAM/DRAM, ces mémoires sont mémoires non – volatiles et sont convenables pour stocker infiniment des programmes et des microprogrammes de gestion. Les mémoires mortes sont construites des circuits intégraux (CI) du type ROM. Tout circuit intégral possède des *entrées adressables*  $A_i$ , par lesquelles est causée l'adresse de lire d'un groupe de « n » bits et des *entrées de données*  $D_i$  pour les données lues. Il existe les variétés suivantes du type **ROM**:

❖ **EPROM** – *Erasable Programmable ROM*: variante de la mémoire type **ROM**. C'est une mémoire de lire qui puisse être mise à zéro. L'effacement des données, écrites sur elle, se réalise par éclairage du crystal semi – conducteur avec la lumière ultra-violette. Après cela on écrit électriquement les données nouvelles dans un dispositif spécial – *programmeur*, branché à l'une des interfaces standardisées de l'ordinateur;

❖ **EEPROM** – *Electrically Erasable Programmable ROM*: cette mémoire semi – conductrice, effaçable et programmable électriquement;

❖ **FLASH – mémoire**: variante d'une mémoire non – volatile ré – programmable du type **EEPROM** dans les ordinateurs contemporains.

Elle est installée sur la plaque principale à circuit intégré. Dans cette mémoire sont ré – enregistrées, à l'aide d'une mémoire externe, les versions nouvelles du BIOS, sans démonter le circuit intégré, dédié pour ce but, pour tous les modèles précédents d'ordinateurs. Actuellement cette technologie est en usage avec des mémoires de flashe à capacitance de déjà 32Gbytes.

La mémoire de flashe apparaît en 1988. Elle se réalise sur des circuits semi – conducteurs de la technologie à couches plates magnétiques. On produit deux types principaux de mémoire de flashe:

- Type **NOR** – logique NON plus OU,
- Type **NAND** – logique NON plus ET.

Pour les deux types de mémoires comme cellules élémentaires de stocker l'information sont usés les transistors de champ. La mémoire typique de flashe use un bit de la cellule élémentaire - la technologie *SBC (Single Bit per Cell)*. Actuellement la technologie *MLC (Multi-Level Cell)* est en usage, qui permet de stocker plusieurs (quelques) bits dans une cellule de cette mémoire. Elle se réalise sur une plaque à circuit intégré (c'est pour cela qu'on la rencontre aussi sous le nom de « carte de flashe »), sur laquelle sont montés deux circuits intégrés), le premier est la mémoire semi – inductrice du type *EEPROM* et le second représente le contrôleur de liaison avec l'ordinateur.

La mémoire de flashe peut être lit des fois arbitraires, mais l'enregistrement est limité - ordinairement environ 10 000 fois. La cause est que pour l'enregistrement il est nécessaire au début d'effacer une partie de la mémoire et cette partie puisse supporter un nombre limité d'effacements; Comme l'effacement est distribuée sur des parties entières, il n'est pas possible de changer un bit ou un byte, sans effacer la partie totale (entière). Cette dernière restriction est pour le type de mémoire de flashe le plus populaire – *NAND*.

L'avantage de la mémoire de flashe par rapport aux mémoires *RAM* et *DRAM* est sa volatilité – en déclenchant la tension, le contenu de la mémoire est sauvegardé. Son avantage en regard des dispositifs à disques est l'absence des parties électro – mécaniques. C'est pour cela que la mémoire de flashe est compacte, à bas prix et assure un accès plus rapide.

L'imperfection par comparaison des dispositifs à disques durs est la capacitance relative de moindre volume: le volume des cartes de flashe les plus grandes, appelées déjà dispositifs de flashe est jusqu'à 128 / 256 Gbytes. Le développement d'augmenter le volume continue.

Grace à la compacité, le prix bas et le manque de nécessité de tension, la mémoire de flashe est usée largement en cartes de flashe dans les dispositifs portatifs, fonctionnant en batteries – le caméra et le vidéo caméra digitaux, le dictaphone digital, MP3-dispositif, PDA (**Personnel** Digital Assistant), tablettes, mobile phones, etc. Sous la forme de *USB* carte de flashe (avec la sortie standardisée pour porte *USB*) on l'exploite pour stocker des données dans les ordinateurs. Le logiciel monté dans les différents dispositifs périphériques et de réseau (routeurs, imprimantes, scanners) est enregistré plus souvent déjà sur ce type de mémoire.

## **8.2. Programmation. PLD, FPGA**

Ce sont deux technologies de circuits logiques programmables. Un **circuit logique programmable**, ou **réseau logique programmable**, est un circuit intégré logique qui peut être reprogrammé après sa fabrication. Notons qu'il serait impropre de parler de programmation au sens logiciel (contrairement à un microprocesseur, il n'exécute aucune ligne de code). Ici, mieux vaudrait parler de « reconfiguration » plutôt que de reprogrammation (on modifie des connexions ou le comportement du composant, on connecte des portes logiques entre elles, *etc.*). Le verbe programmer est toutefois le plus fréquent, mais au sens de personnaliser. Il s'agit bel et bien de réseaux logiques reconfigurables et modifiables.

Il est composé de nombreuses cellules logiques élémentaires et bascules logiques librement



connectables (c'est justement la reconfiguration, ou programmation, du composant qui définit les connexions faites entre portes logiques).

Ce type de composant électronique est communément désigné par différents **SIGLES ANGLAIS** dont notamment :

- **FPGA** (*field-programmable gate array*, réseau de portes programmables *in situ*) ;
- **PLD** (*programmable logic device*, circuit logique programmable) ;
- **EPLD** (*erasable programmable logic device*, circuit logique programmable et effaçable) ;
- **CPLD** (*complex programmable logic device*, circuit logique programmable complexe) ;
- **PAL** (*programmable array logic*, réseau logique programmable) ;
- **PLA** (*programmable logic array*, réseau logique programmable).

Bien que fondamentalement synonymes, ces termes ne sont généralement pas interchangeables dans le vocabulaire commercial des fabricants : FPGA désigne plutôt des composants à technologie RAM, EPLD des composants à technologie FLASH, PAL des composants à technologie fusible (voir § procédés technologiques).

Les premiers brevets pour de tels composants datent des années 1980 à 1990 et c'est au début des années 1990 qu'ils se sont généralisés.

### 8.2.1. PLD- CPLD

On va discuter ici sur la technologie **CPLD** (*complex programmable logic device*, circuit logique programmable complexe) comme bien répandue aujourd'hui au lieu de **PLD** (*programmable logic device*, circuit logique programmable).

Les PAL, GAL, CPLD et EPLD, de conception plus ancienne, utilisent des « macrocellules » logiques, composées d'un réseau combinatoire de portes ET et OU afin d'implémenter des équations logiques. Des bascules sont disponibles seulement dans les blocs d'entrée-sortie. Un composant contient de quelques dizaines à quelques centaines de macrocellules.

Comme le routage est fixe, les temps de propagations sont bornés et permettent une fréquence de fonctionnement élevée et indépendante du design. Par contre, l'utilisation des ressources n'est pas optimale (tout terme non utilisé dans une équation logique équivaut à des portes perdues), avec des taux d'utilisation d'environ 25 %.

On distingue les CPLD des autres PLD car ils contiennent l'équivalent de plusieurs composants PLD, reliés par une matrice d'interconnexion.

### 8.2.2. FPGA

C'est une structure d'interconnexion programmable. La plupart des grands FPGA modernes sont fondés sur des cellules SRAM aussi bien pour le routage du circuit que pour les blocs logiques à interconnecter.

Un bloc logique est de manière générale constitué d'une table de correspondance (LUT ou *Look-Up-Table*) et d'une bascule (*Flip-Flop* en anglais). La LUT sert à implémenter des équations logiques ayant généralement 4 à 6 entrées et une sortie. Elle peut toutefois être considérée comme une petite mémoire, un multiplexeur ou un registre à décalage. Le registre permet de mémoriser un état (machine séquentielle) ou de synchroniser un signal (*pipeline*).

Les blocs logiques, présents en grand nombre sur la puce (de quelques milliers à quelques millions en 2007) sont connectés entre eux par une matrice de routage configurable. Ceci permet la reconfiguration à volonté du composant, mais occupe une place importante sur le silicium et justifie le coût élevé des composants FPGA. La topologie est dite « Manhattan », en référence aux rues à angle droit de ce quartier de New York.

Les densités actuelles ne permettent plus un routage manuel, c'est donc un outil de placement-routage automatique qui fait correspondre le schéma logique voulu par le concepteur et les ressources matérielles de la puce. Comme les temps de propagation dépendent de la longueur des liaisons entre cellules logiques, et que les algorithmes d'optimisation des placeurs-routeurs ne sont pas déterministes, les performances (fréquence max.) obtenues dans un FPGA sont variables d'un design à l'autre. L'utilisation des ressources est par contre très bonne, et des taux d'occupation des blocs logiques supérieurs à 90 % sont possibles.

Comme la configuration (routage et LUT's) est faite par des points mémoire volatils, il est nécessaire de sauvegarder le design du FPGA dans une mémoire non volatile externe, généralement une mémoire Flash série, compatible « JTAG ». Certains fabricants se distinguent toutefois par l'utilisation de cellules EEPROM pour la configuration, éliminant le recours à une mémoire externe, ou par une configuration par anti-fusibles (la programmation par une tension élevée fait « claquer » un diélectrique, créant un contact). Cette dernière technologie n'est toutefois pas reconfigurable.

Quelques fonctionnalités particulières disponibles sur certains composants :

- blocs de mémoire supplémentaires (hors des LUT), souvent double-port, parfois avec mécanisme de FIFO,
- multiplieurs câblés (coûteux à implémenter en LUT), voire blocs multiplieur-accumulateur pour traitements DSP,
- cœur de microprocesseur enfoui (dit *hard core*) comme par exemple des architectures PowerPC ou ARM,
- blocs PLL pour synthétiser ou resynchroniser les horloges,
- reconfiguration partielle, même en cours de fonctionnement,
- chiffrement des données de configuration,
- sérialiseurs / désérialiseurs dans les entrées-sorties, permettant des liaisons série haut-débit,
- impédance contrôlée numériquement dans les entrées-sorties, évitant de nombreux composants passifs sur la carte.
- couche MAC Ethernet
- couches matérielles.