



Master 1 – SEM

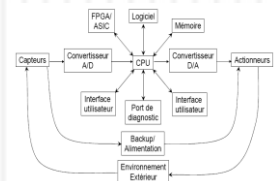
Processeurs Embarqués

Cours 2

Éléments logiques programmables - PLD

Année 2022-2023

Pr R. BOUDOUR



Circuits logiques

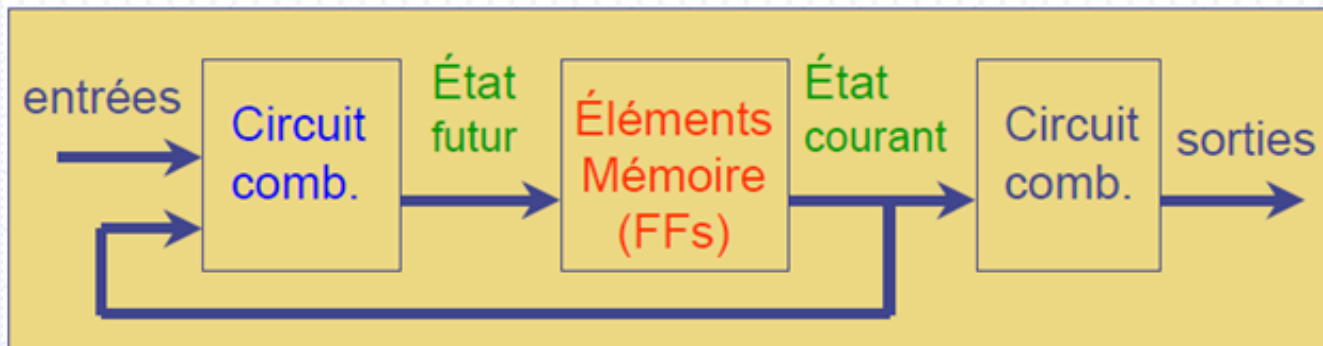
2

- ❑ Les circuits logiques sont élaborés à partir de composants électroniques : transistors
- ❑ Types de circuits logiques :
 - ❑ combinatoires
 - ❑ séquentiels
- ❑ Circuits combinatoires
 - ❑ Les sorties ne dépendent que des valeurs des entrées
- ❑ Circuits séquentiels
 - ❑ Ajout des notions d'état et de mémoire
 - ❑ Ajout de la notion de temps (horloge)
- ❑ Théories utilisées pour étudier ou spécifier les différents types de circuits :
 - ❑ circuits combinatoires : Algèbre de Boole
 - ❑ circuits séquentiels : Théorie des automates finis

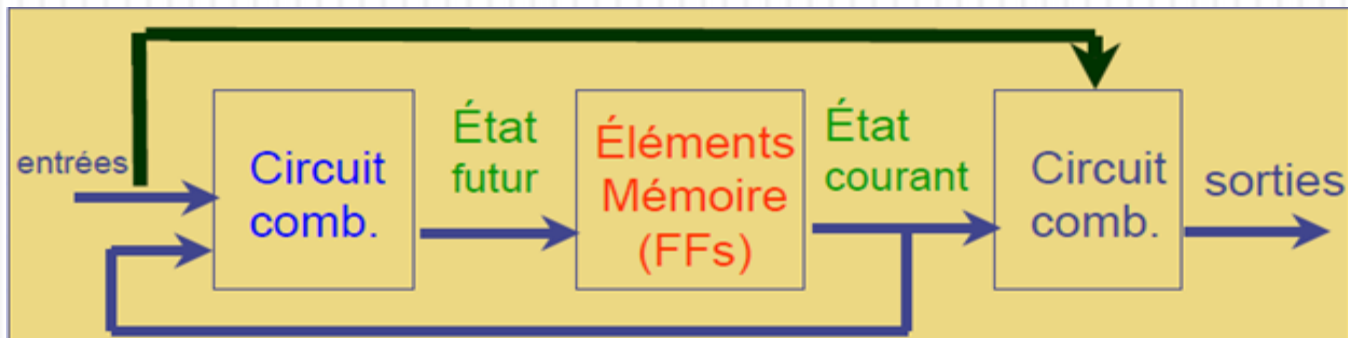
Automates Moore / Mealy

3

□ Automate de type Moore



□ Automate de type Mealy

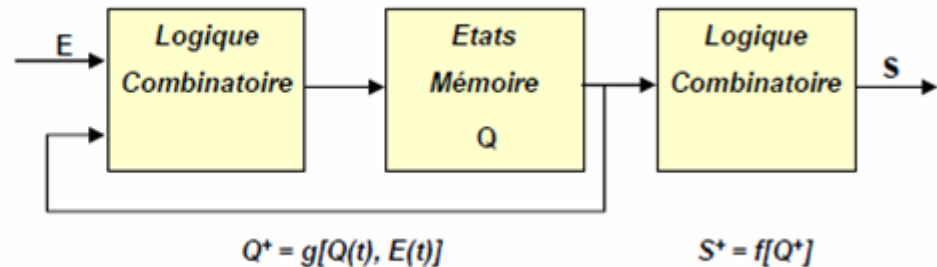


Automates Moore/Mealy

4

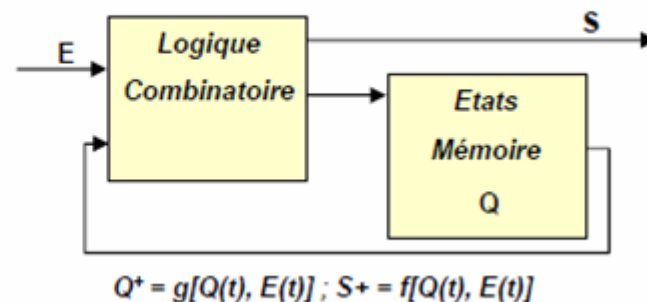
- **Automate de Moore**

On calcule d'abord les nouveaux états puis les sorties



- **Automate de Mealy**

On calcule en parallèle les nouveaux états et les nouvelles sorties. Plus général que Moore



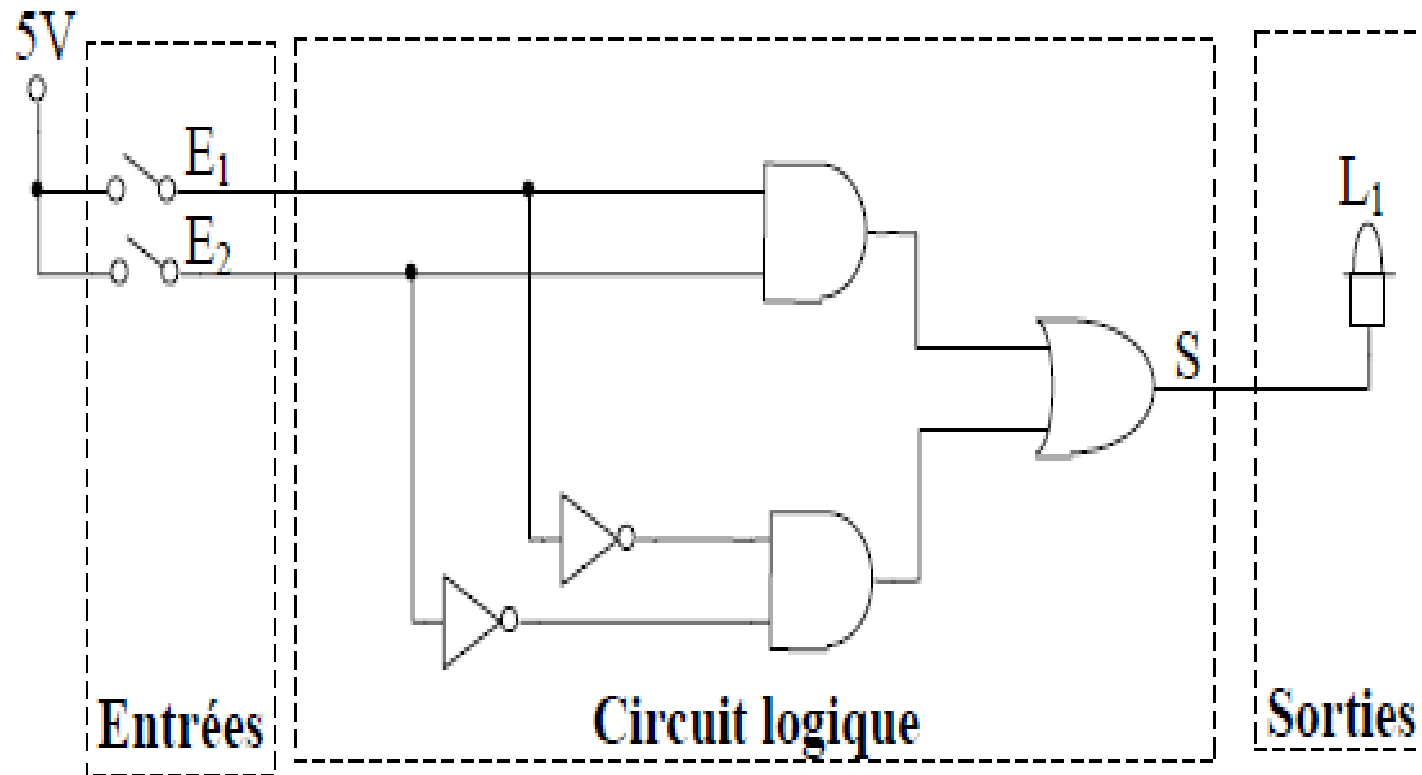
Machine Moore vs. Machine Mealy

5

S.No.	Mealy Machine	Moore Machine
1	Outputs depend on the present state and present inputs.	Outputs depend only on the present state.
2	The output changes asynchronously with the enabling clock edge.	Since the outputs change when the state changes and the state change is synchronous with the enabling clock edge, outputs change synchronously with this clock edge in Moore machine.
3	A counter is not a Mealy machine.	A counter is a Moore machine.
4	Will have the same or fewer states	The number of states will be equal or more.

Exemple de circuit logique simple

6



Analyse de circuit logique

7

☐ Trouver sa fonction logique

☐ Principe

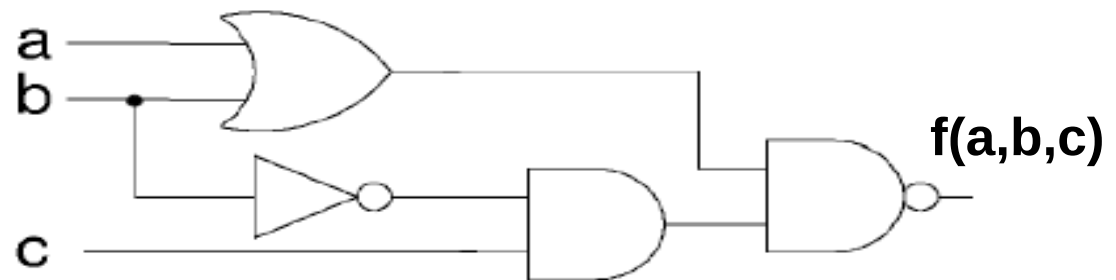
- ☐ Donner l'expression des sorties de chaque porte-composant en fonction des valeurs de ses entrées
- ☐ En déduire au final la ou les fonctions logiques du circuit
- ☐ on peut ensuite
 - ☐ Déterminer la table de vérité du circuit
 - ☐ Simplifier la fonction logique

Analyse de circuit logique

8

3 entrées, 1 sortie

Composé uniquement de portes logiques OU, ET et NON



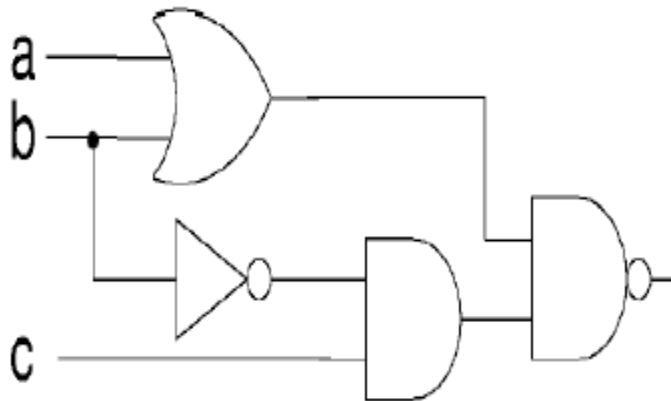
A partir de son logigramme

$$f(a,b,c) = \overline{(a + b) \cdot (\bar{b} \cdot c)}$$

Analyse de circuit logique

9

$$f(a,b,c) = \overline{(a+b) \cdot (\bar{b} \cdot c)}$$



a	b	c	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Synthèse d'un circuit logique

10

A partir d'une fonction logique trouver le logigramme correspondant à cette fonction

Principe

- Simplifier la fonction logique avec 2 méthodes
 - La méthode algébrique (algèbre de Boole)
 - La méthode des tableaux de Karnaugh
- En déduire le logigramme correspondant

Portes à 3 états

11

Extension de la logique à 2 états avec un troisième état : « état indéfini »

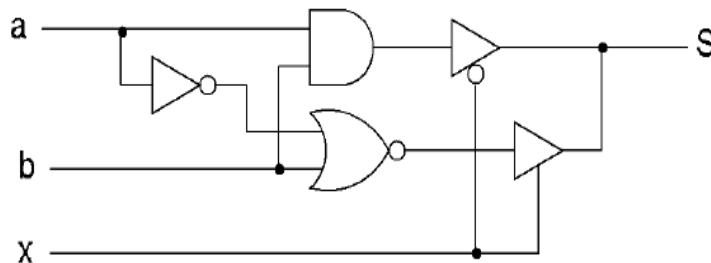
Utilité principale : activer ou désactiver des parties d'un circuit liées aux mêmes sorties ou éléments



Pour (1) : si $E = 1$ alors $o = i$, si $E = 0$, alors $o = ?$ (indéfini)

Pour (2) : si $E = 0$ alors $o = i$, si $E = 1$, alors $o = ?$ (indéfini)

Exemple de circuit



Selon la valeur de x , S correspond à la sortie d'une des 2 portes

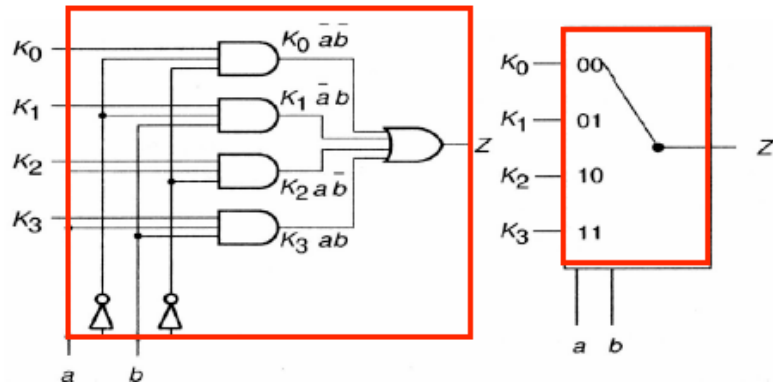
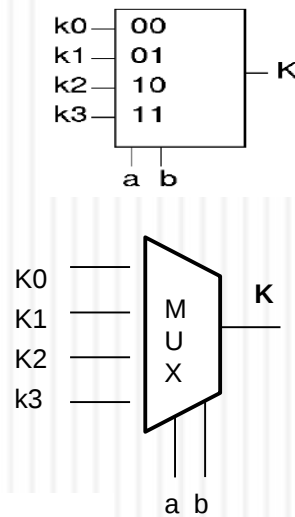
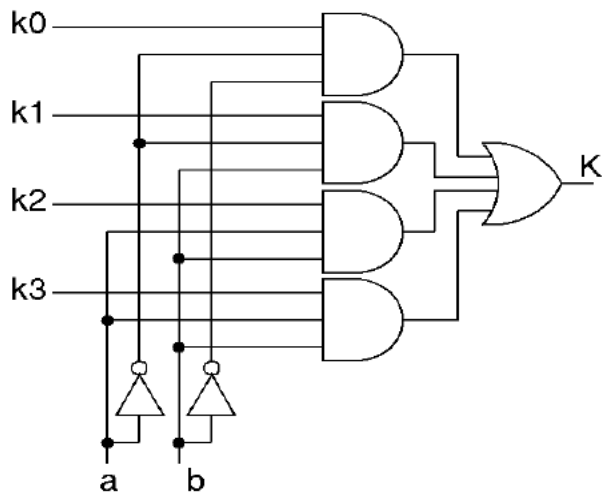
Si $x = 0$ alors $S = a b$

Si $x = 1$ alors $S = \overline{a} + \overline{b} = a \overline{b}$

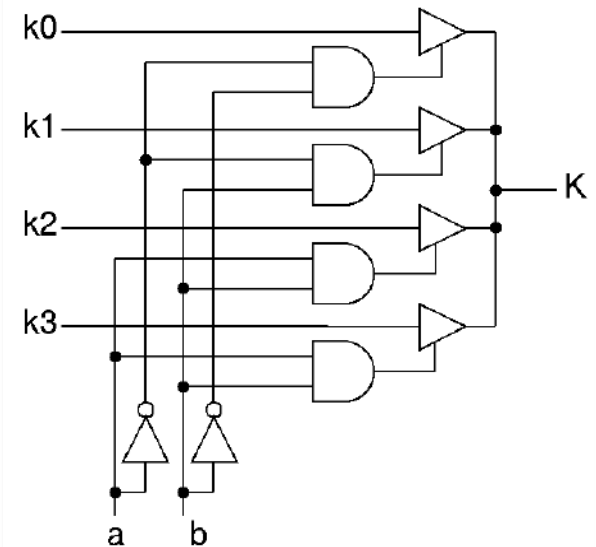
Quelques circuits combinatoires

12

❑ Multiplexeur



Variante avec logique à 3 états

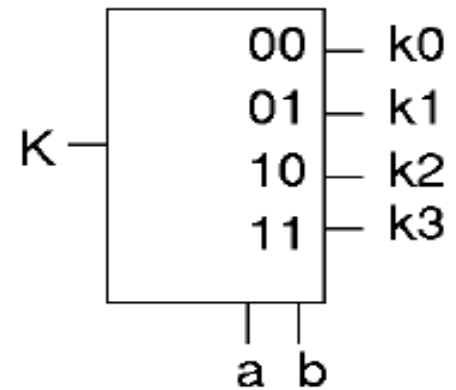
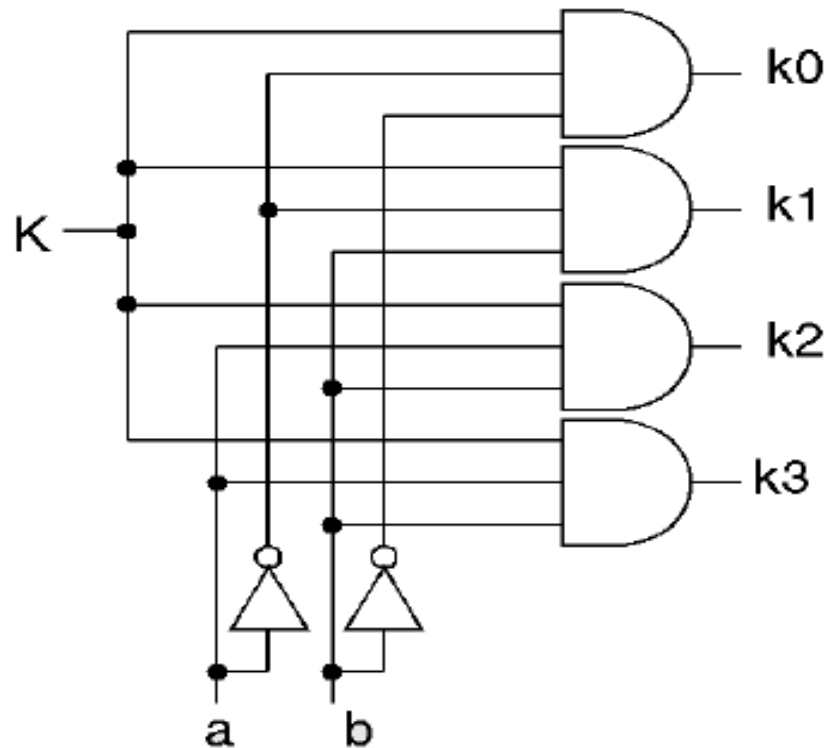


Selon la valeur de a et b , on ne redirige qu'un des quatre k_x vers K

Quelques circuits combinatoires

13

❑ Démultiplexeur



Demi additionneur et additionneur

14

❑ Demi additionneur 1 bit

❑ $S = a \text{ xor } b$

❑ $Rs = a.b$

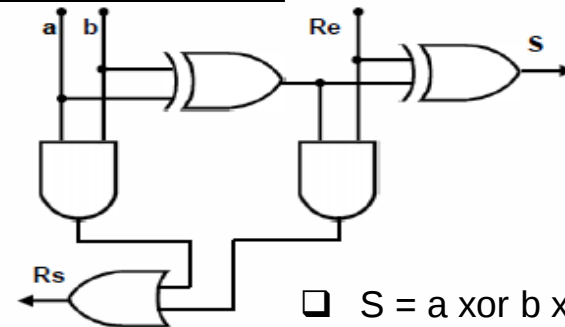
● Additionneur

● Cellule addition 1 bit

a	b	Re	S	Rs
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

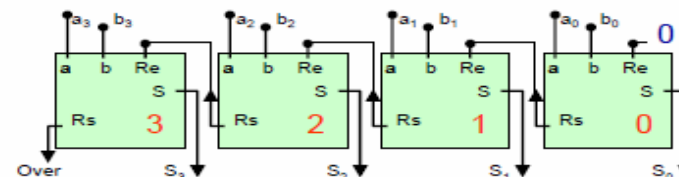
● Additionneur 4 bits

a	b	S	Rs
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



❑ $S = a \text{ xor } b \text{ xor } Re$

❑ $Rs = a.b + Re(a \text{ xor } b)$



Quelques circuits combinatoires

15

□ Additionneur 4 bits (avec bits d'état)

$$\begin{array}{r} 0100 \\ + 0011 \\ \hline = 0111 \end{array}$$

$C=0$

$$O = r_2 + r_3 = 0$$

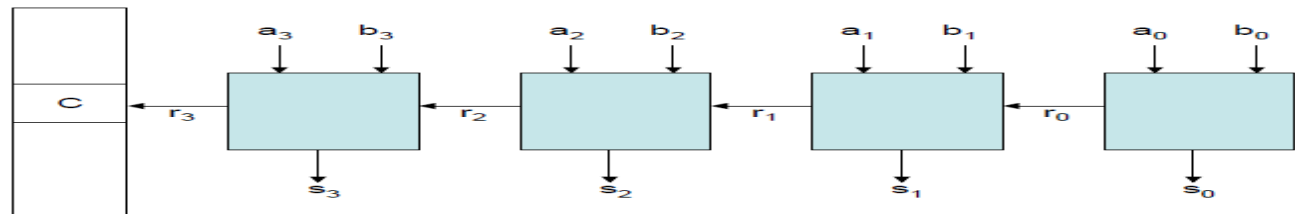
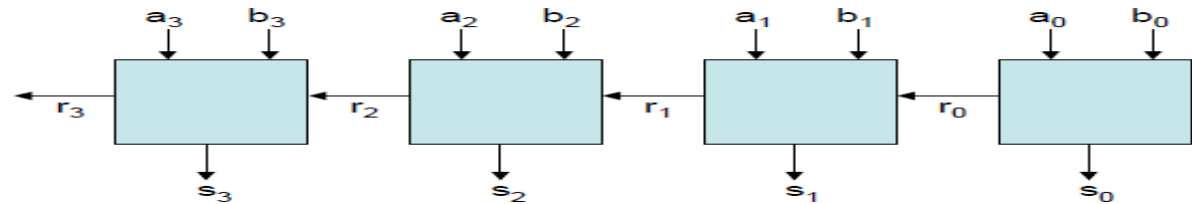
Pas de débordement

$$\begin{array}{r} 0101 \\ + 0110 \\ \hline = 1011 \end{array}$$

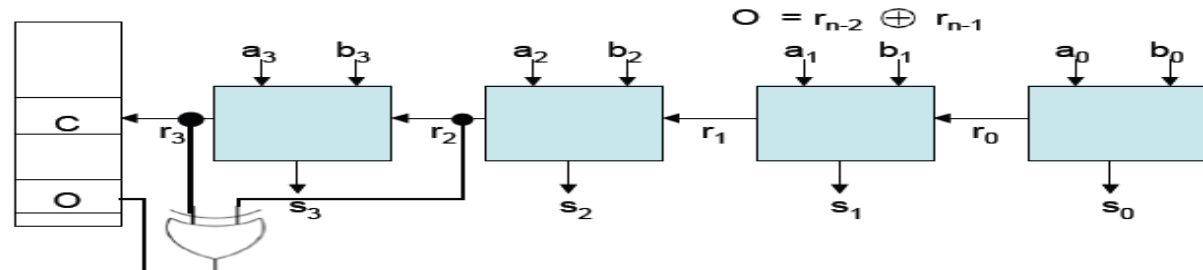
$C=0$

$$O = r_2 + r_3 = 1$$

Avec débordement



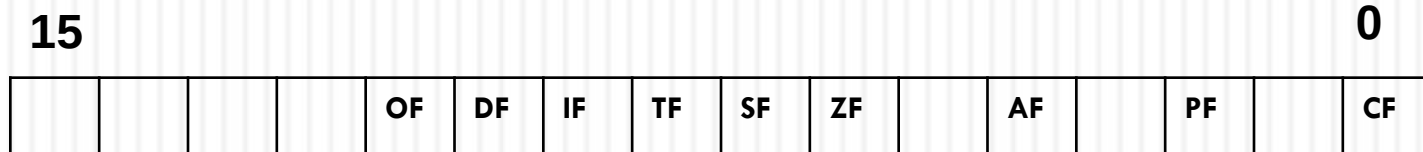
Registre d'état



Registre d'état

16

- Les drapeaux sont des indicateurs qui annoncent une condition particulière suite à une opération arithmétique ou logique.
- Le registre d'état du 8086 est formé par les bits suivants



CF : Retenue

PF : Parité

AF : Retenue auxiliaire

ZF : Zéro

SF : Signe

OF : Débordement

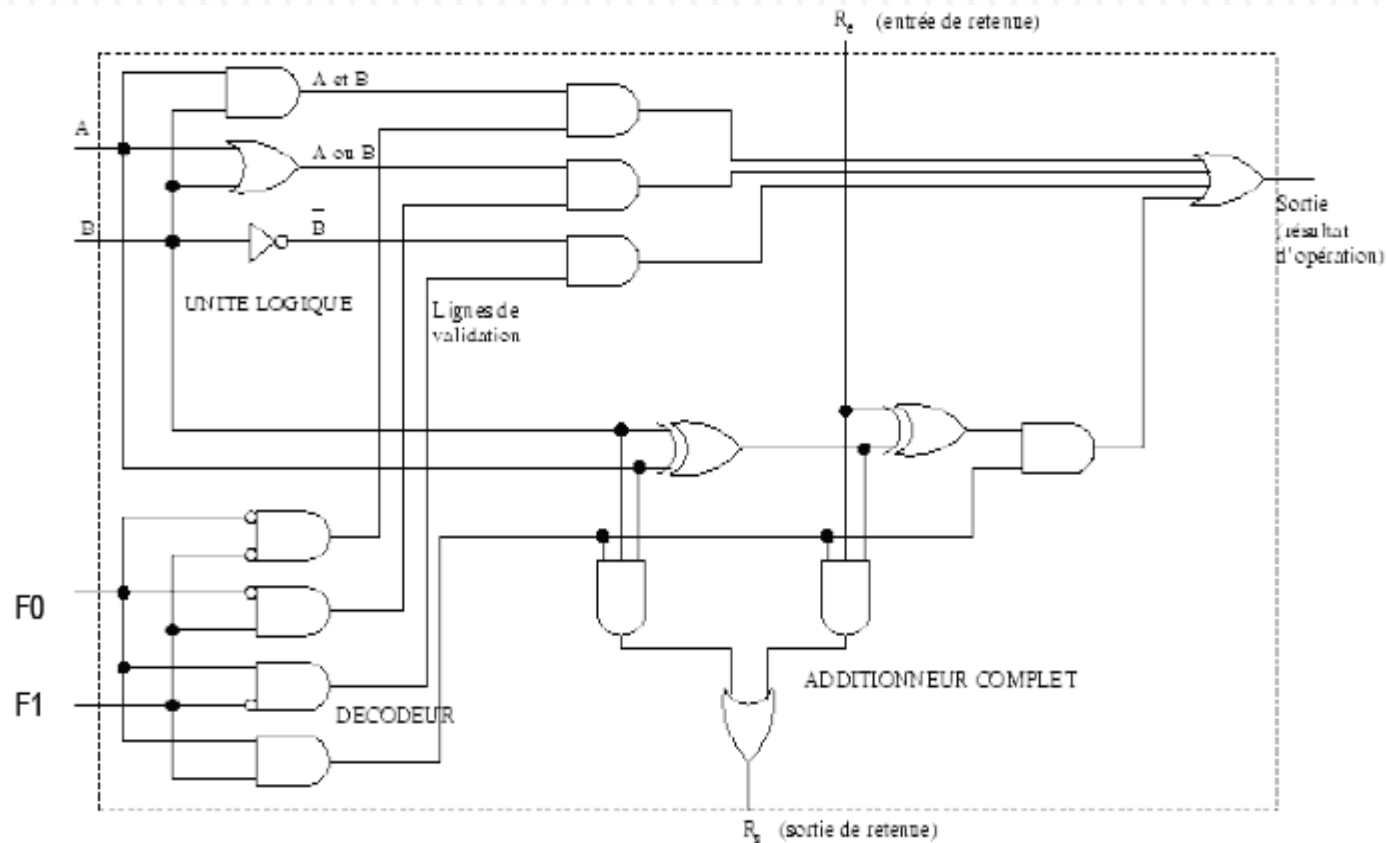
IF : Interruption

TF : Trappe (Mode pas à pas)

Quelques circuits combinatoires

17

□ UAL à 1 bit



Bascules

18

De gauche à droite

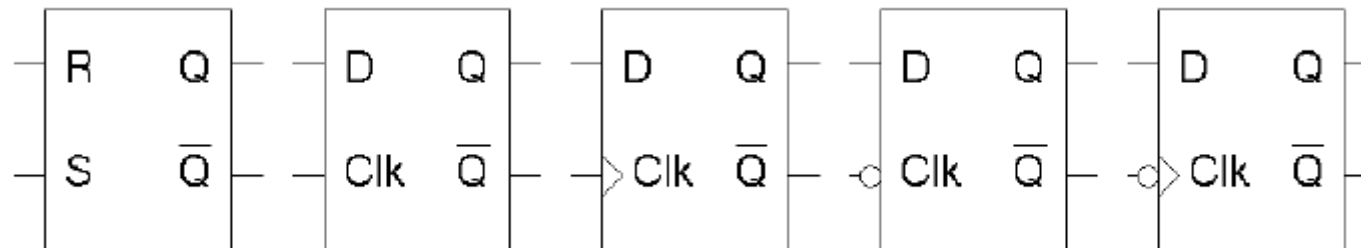
Bascule RS asynchrone

Bascule D latch sur niveau 1

Bascule D flip-flop sur front montant

Bascule D latch sur niveau 0

Bascule D flip-flop sur front descendant



Utilisation des bascules

19

Utilisation des bascules pour créer des circuits ayant un état

Compteurs

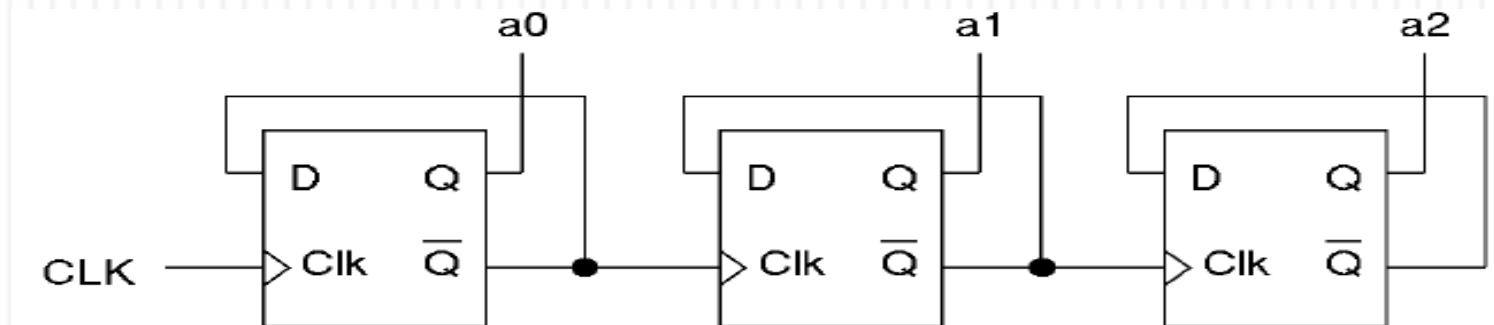
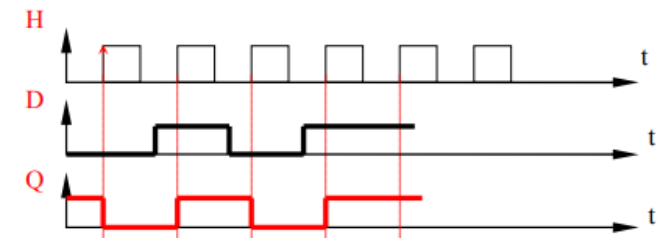
Registres

Mémoires

Table de vérité

H	D	Qn
0	X	Qn-1
↑	0	0
↑	1	1

Chronogramme



3 signaux d'horloge à 3 fréquences différentes

Représente les combinaisons de bits pour les valeurs de 0 à 7

Mémoire

20

● Mémoire Centrale

● mémoire

4 mots de 3 bits

● contrôleur

➤ instructions

□ écriture

□ lecture

Décodeur

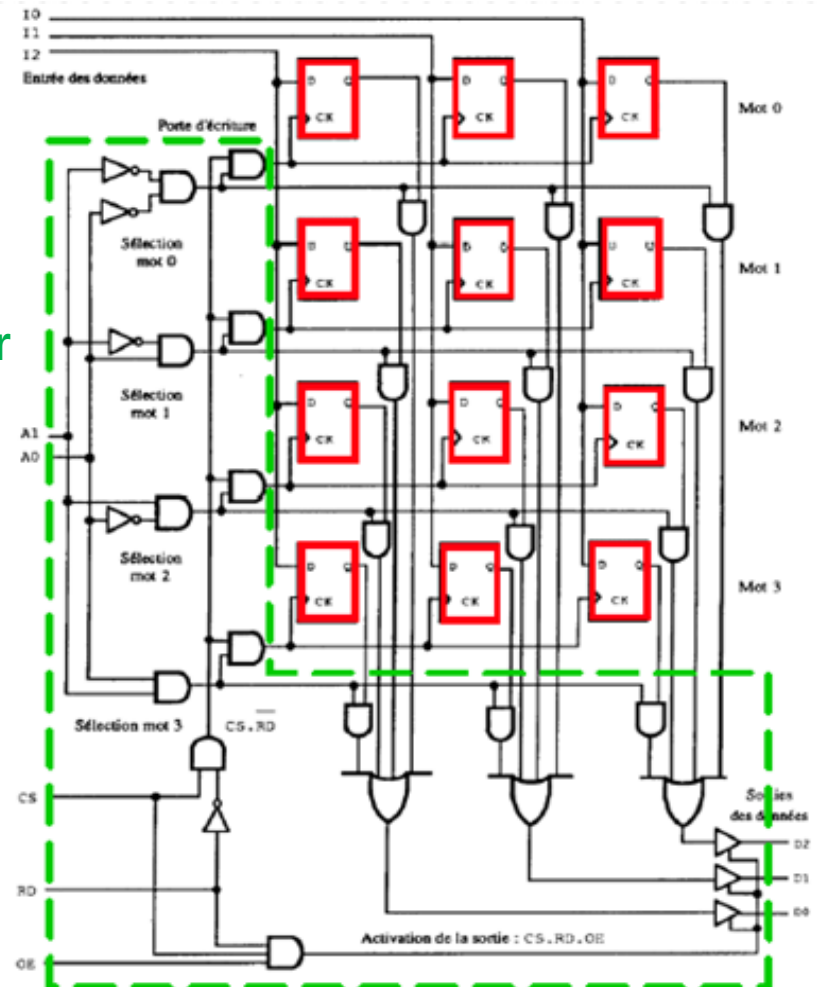
Légende

CS Chip Select

RD Read (Write)

OE Output Enable

(d'après Tannenbaum)



Éléments logiques programmables

21

Fonction logique : $F = ABCD + \overline{A}BCD + AB\overline{C} + \overline{A}B\overline{C}D$

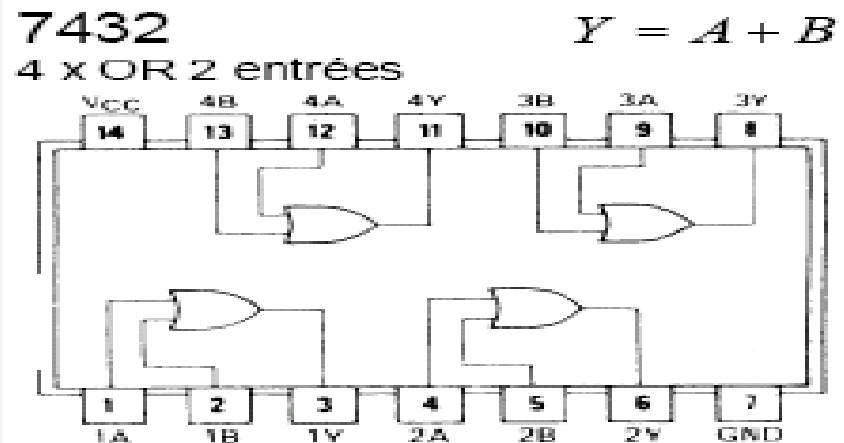
Avant :

- ☐ Table de vérité
- ☐ Tableau de Karnaugh
- ☐ Réalisation à l'aide de portes logiques

- ☐ Ou avec des CI

Aujourd'hui :

- ☐ Réalisation à l'aide d'un PLD



Eléments logiques programmables

22

- ❑ Les concepteurs logiciens disposent d'un large éventail de CI dans lesquels sont matérialisés des fonctions et des montages logiques
- ❑ Les CI sont fabriqués par différents constructeurs
- ❑ Ils sont très bon marché



Des problèmes se posent quand on construit des circuits et des systèmes avec seulement des CI existants

- ❑ Certaines solutions peuvent avoir besoin de centaines voire des milliers de CI
- ❑ Une plaquette imprimée de grande surface pour ce grand nombre de CI
- ❑ Un temps important pour l'insertion, le soudage et l'essai de toutes ces puces
- ❑ Le fabricant doit conserver un important stock de tous ces CI

Eléments logiques programmables

23

- Réduire le nombre de CI dans une conception peut procurer des avantages :
 - ▣ surface de plaquette moindre, donc circuits imprimés réduits et châssis moins encombrants
 - ▣ besoins en puissance plus faibles (alimentations plus petites)
 - ▣ procédés d'assemblage plus rapides
 - ▣ fiabilité accrue (moins de CI, risque de rupture d'une connexion diminuée)
 - ▣ dépannage plus aisé



Mettre un nombre de plus en plus important de fonctions dans un CI

Eléments logiques programmables

24

□ Remarques :

- Les circuits LSI, VLSI incorporent des fonctions courantes telles que : mémoire, microprocesseur, calculatrice, etc.
- Ils possèdent des centaines voire des milliers de portes, toutes câblées en interne afin de matérialiser un fonctionnement prévisible
- Mais dans de nombreux projets, il n'y a pas de solution LSI et VLSI toute faite



Pour y parvenir, les concepteurs doivent assembler des puces SSI et MSI pour bâtir le système envisagé ou utiliser un **circuit particulier** :

Un **élément logique programmable** pour remplacer plusieurs CI existants (SSI et MSI) par un seul CI

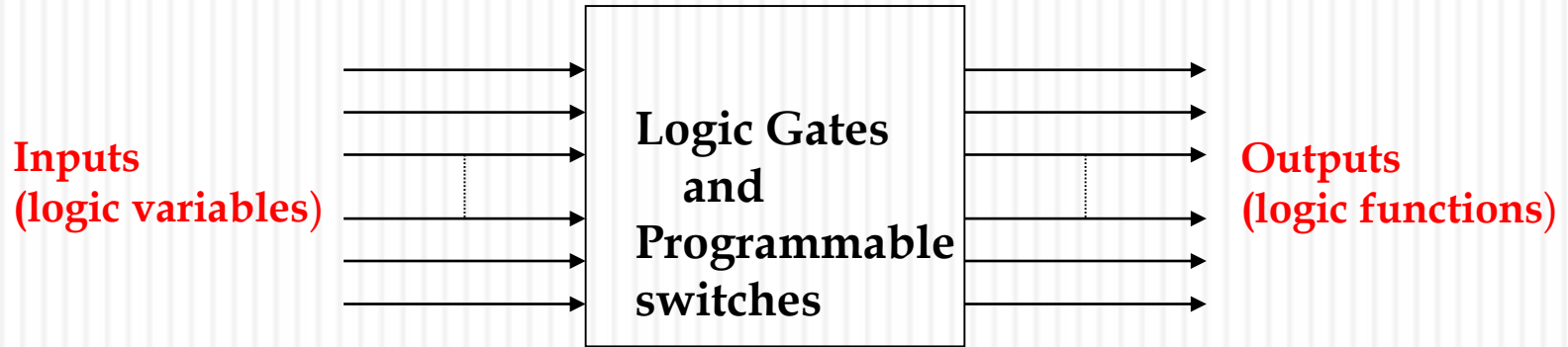
Eléments logiques programmables

25

- ❑ Les circuits logiques programmables ou PLD (Programmable Logic Devices) apparaissent à la fin des années 1970.
- ❑ Ces circuits permettent de réaliser une ou plusieurs fonctions logiques ou/et fonctions séquentielles faisant intervenir une ou plusieurs variables d'entrée.
- ❑ Un circuit est dit **configurable (ou programmable)** lorsque sa fonctionnalité n'est pas prédéfinie lors de sa fabrication mais peut être spécifiée plus tard par une **configuration (un programme)**.
- ❑ La programmation comme mesure de flexibilité d'un circuit.

Structure de base d'un PLD

26



Programmable Logic Device comme boîte noire

- ☐ Un PLD est caractérisé par :
 - ☐ Le nombre d'entrées
 - ☐ Le nombre de sorties
 - ☐ Le nombre de termes produits par sortie.
 - ☐ Le retard de propagation (vitesse).
 - ☐ La consommation de puissance.
 - ☐ La technologie.

Structure de base d'un PLD

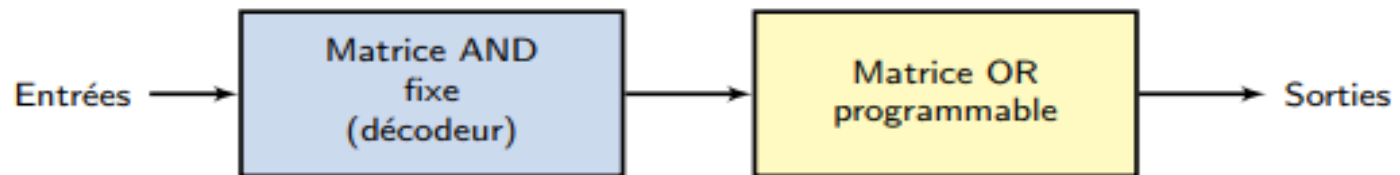
27

- La plupart des PLDs suivent la structure suivante :
 - ▣ Un ensemble d'opérateurs « ET » sur lesquels viennent se connecter les variables d'entrée et leurs compléments.
 - ▣ Un ensemble d'opérateurs « OU » sur lesquels les sorties des opérateurs « ET » sont connectées.
 - ▣ Une éventuelle structure de sortie (Portes inverseuses, logique 3 états, registres...).
- Les deux premiers ensembles forment chacun ce qu'on appelle une **matrice**. Les interconnexions de ces matrices doivent être programmables. C'est la raison pour laquelle elles sont assurées par des **fusibles** qui sont « grillés » lors de la programmation. Lorsqu'un PLD est vierge toutes les connexions sont assurées.

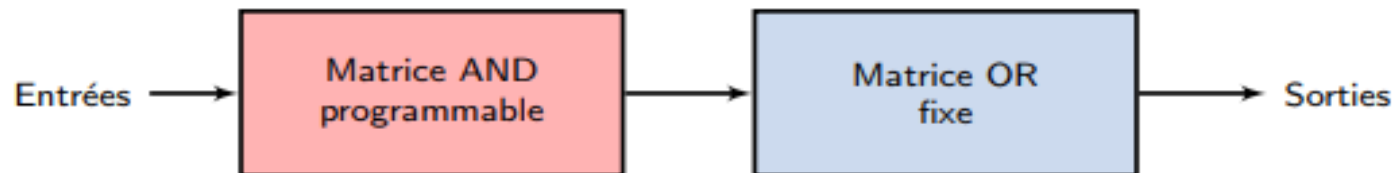
PLD principaux

28

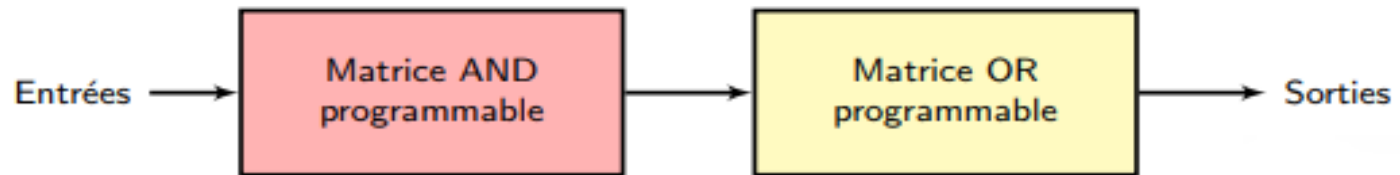
Trois types principaux de PLD: le PROM, le PAL et le PLA. Le PLA offre plus de flexibilité.



a) PROM



b) PAL



c) PLA

Familles de PLD

29

- ❑ Il existe plusieurs familles de PLD qui sont différenciées par leur structure interne. Le tableau suivant présente certaines de ces familles.

Type	Nombre de portes intégrées	Matrice ET	Matrice OU	Effaçable
PROM	2 000 à 500 000	Fixe	Programmable	Non
PAL	10 à 100	Programmable	Fixe	Non
GAL	10 à 100	Programmable	Fixe	Electriquement
EPLD	100 à 3000	Programmable	Fixe	Aux U-V
FPGA	2000 à 3000	Programmable	Programmable	Electriquement

Différences : Matrice programmable/Fixe - Type effacement

GAL : Generic Array Logic

30

- L'inconvénient majeur des PALs 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, il y a quelques années, à remplacer les fusibles irréversibles des PALs par des transistors MOS FET pouvant être régénérés. Ceci a donc donné naissance aux GALs que l'on pourrait traduire par « Réseau logique Générique ».
- Les circuits GAL peuvent donc être **reprogrammés à volonté** sans pour autant avoir une durée de vie restreinte.
- On peut aussi noter que dans leur structure interne les GALs sont constitués de **transistor CMOS** alors que les PALs classiques sont constitués de **transistors bipolaires**.
- La **consommation** des GALs est donc beaucoup plus faible.

Éléments logiques programmables

31

- Soit à implémenter les 4 fonctions logiques O_i sur chacun des 3 éléments PLD :

$$O3 = ab + \bar{c}\bar{d}$$

$$O2 = a\bar{b}c$$

$$O1 = ab\bar{c}\bar{d} + \bar{a}\bar{b}cd$$

$$O0 = a + b\bar{d} + c\bar{d}$$

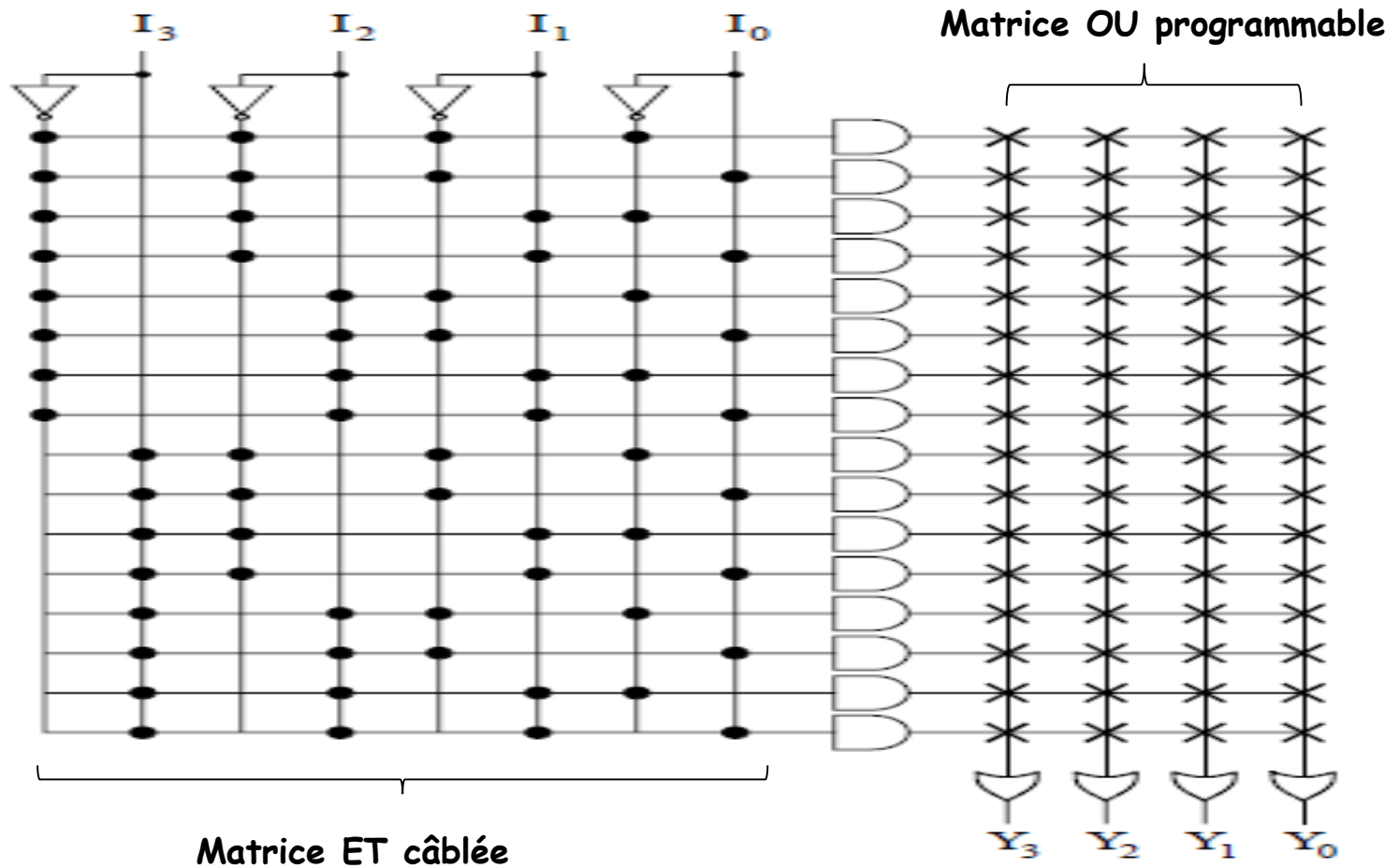
❑ PROM

Les mémoires de type PROM (programmable memory) sont programmables une seule fois au moyen d'un appareil spécial, le programmeur. Les données qui y sont inscrites ne sont pas modifiables. Elles conservent les informations quand l'alimentation est interrompue. Leur inconvénient majeur est l'impossibilité de modifier les informations qu'elles contiennent.

- ❑ Le schéma ci-après donne une représentation symbolique simplifiée d'une PROM de 16 mots de 4 bits.
 - ❑ Les portes ET reçoivent 4 variables directes ou inverses.
 - ❑ Les connexions de gauche, correspondant au décodeur et représentées par des disques noirs, sont figées.
 - ❑ Les croix à droite représentent des connexions programmables.
 - ❑ Les portes OU peuvent recevoir jusqu'à 16 variables.
 - ❑ Chaque sortie d'une PROM est ainsi représentée comme une somme canonique de produits.

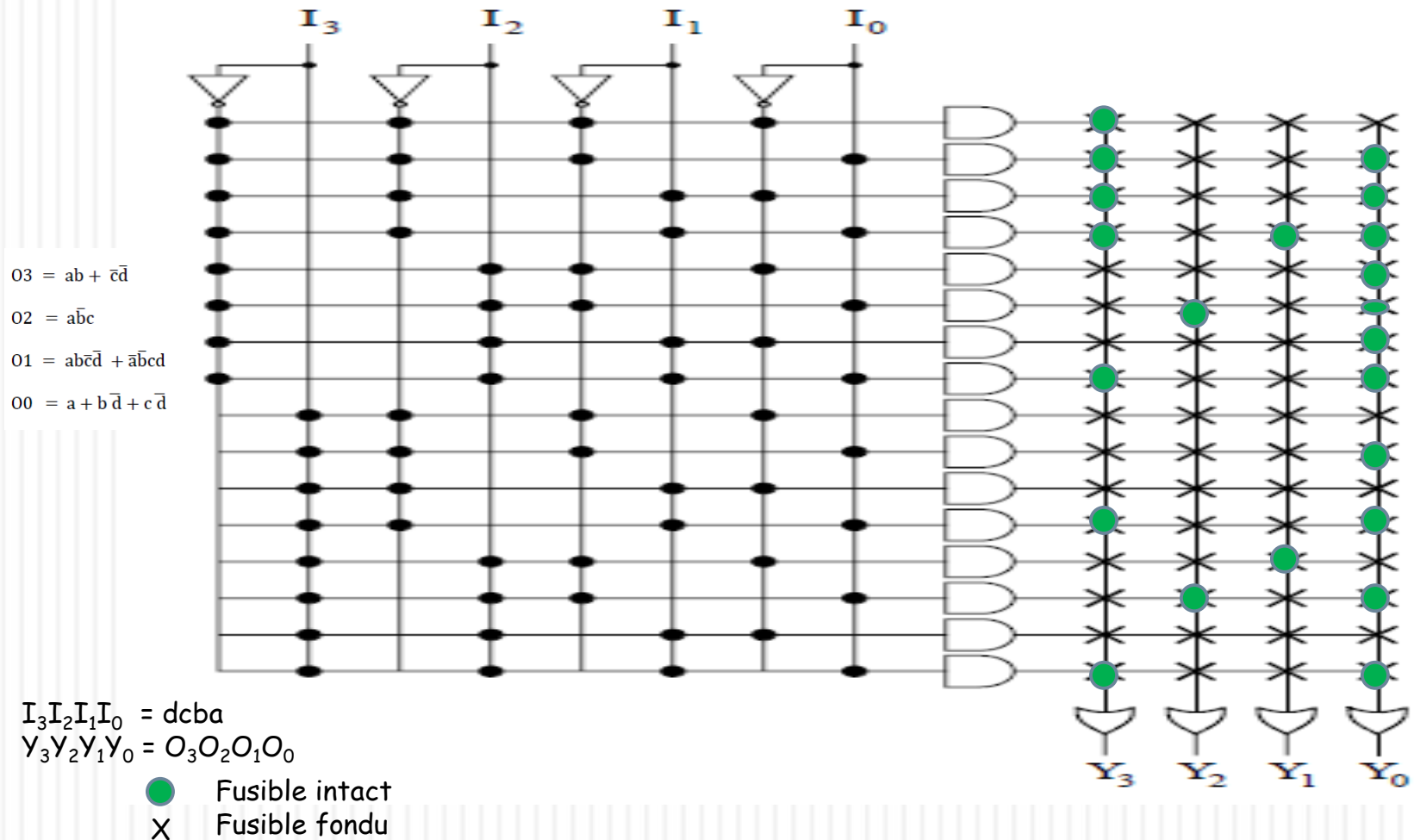
PROM

33



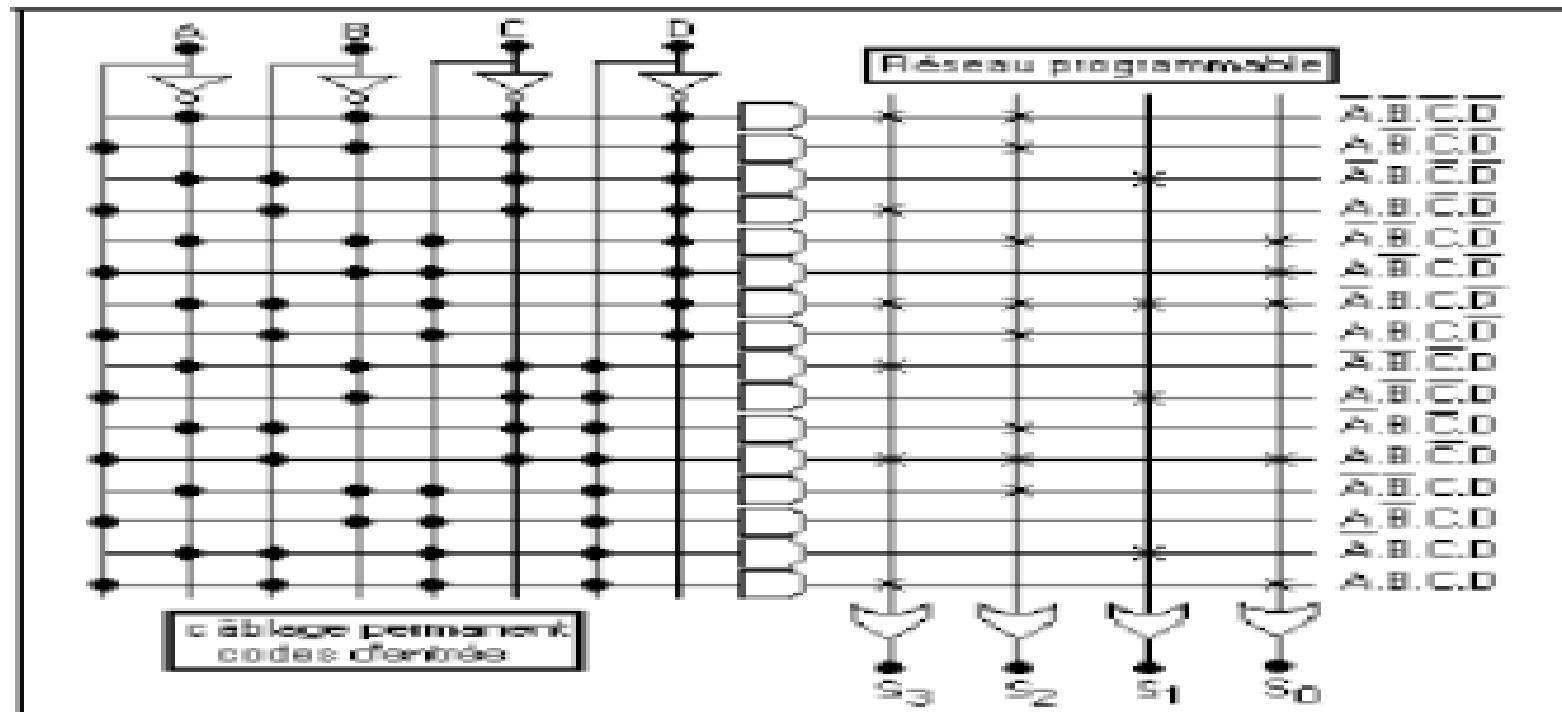
PROM

34



PROM : Autre exemple

35



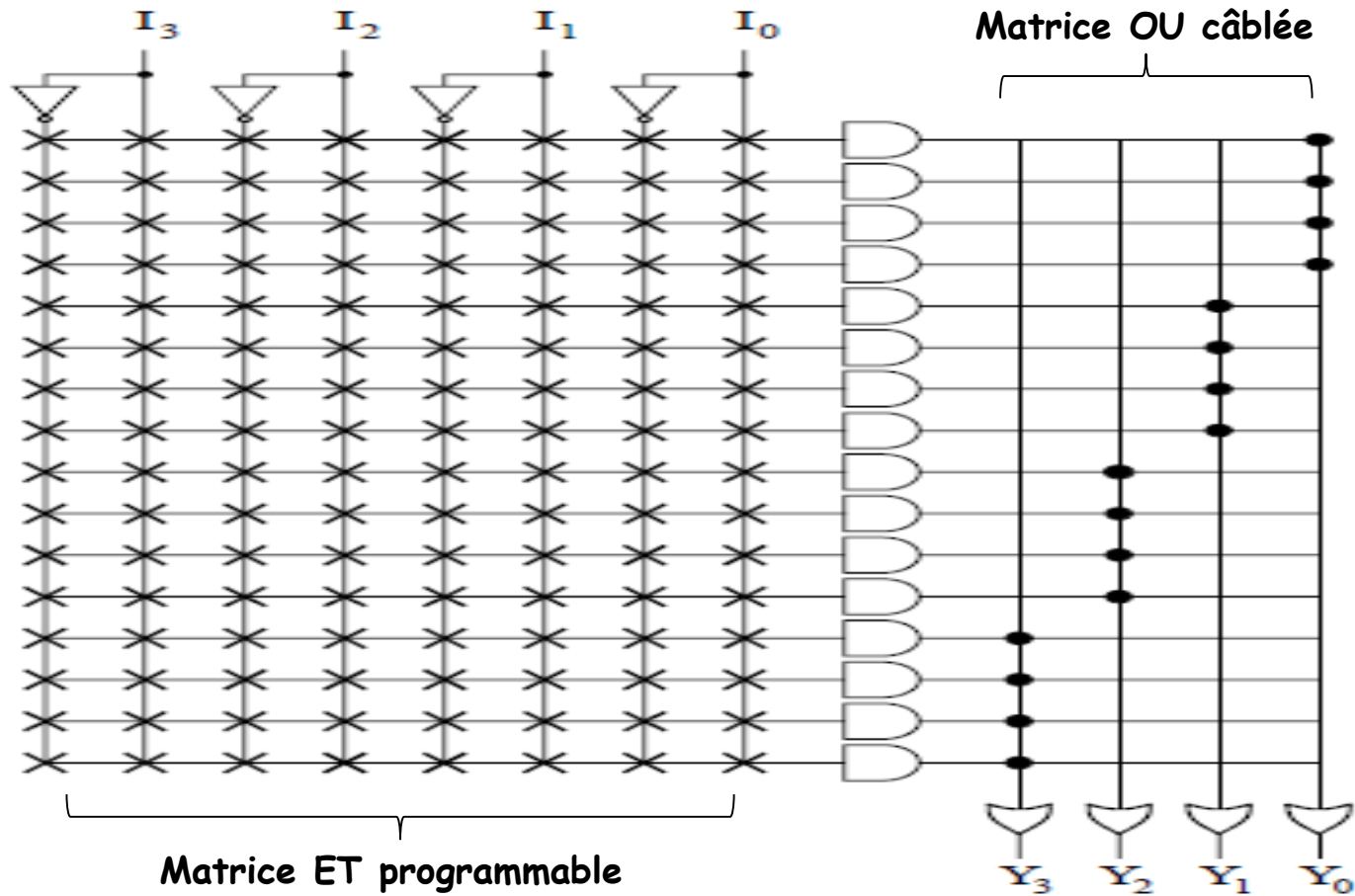
$$S_0 = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}D + ABCD$$

$$S_1 = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}BCD$$

$$S_2 = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}BC\bar{D} + ABC\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + \bar{A}B\bar{C}D$$

$$S_3 = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}D + \bar{A}B\bar{C}D + \bar{A}B\bar{C}D + \bar{A}B\bar{C}D$$

- Ce sont les circuits logiques programmables les plus anciens. Les PAL sont programmés par destruction de fusibles. Ils ne sont donc **programmables qu'une fois**, ce qui peut être gênant en phase de développement. Un PAL permet de remplacer jusqu'à 10 boîtiers SSI ou 2 à 3 boîtiers MSI.



Dans un Programmable Array Logic (PAL), la matrice des connexions OU est figée et la matrice des connexions ET est programmable

PAL

38

$$O_3 = ab + \bar{c}\bar{d}$$

$$O_2 = a\bar{b}c$$

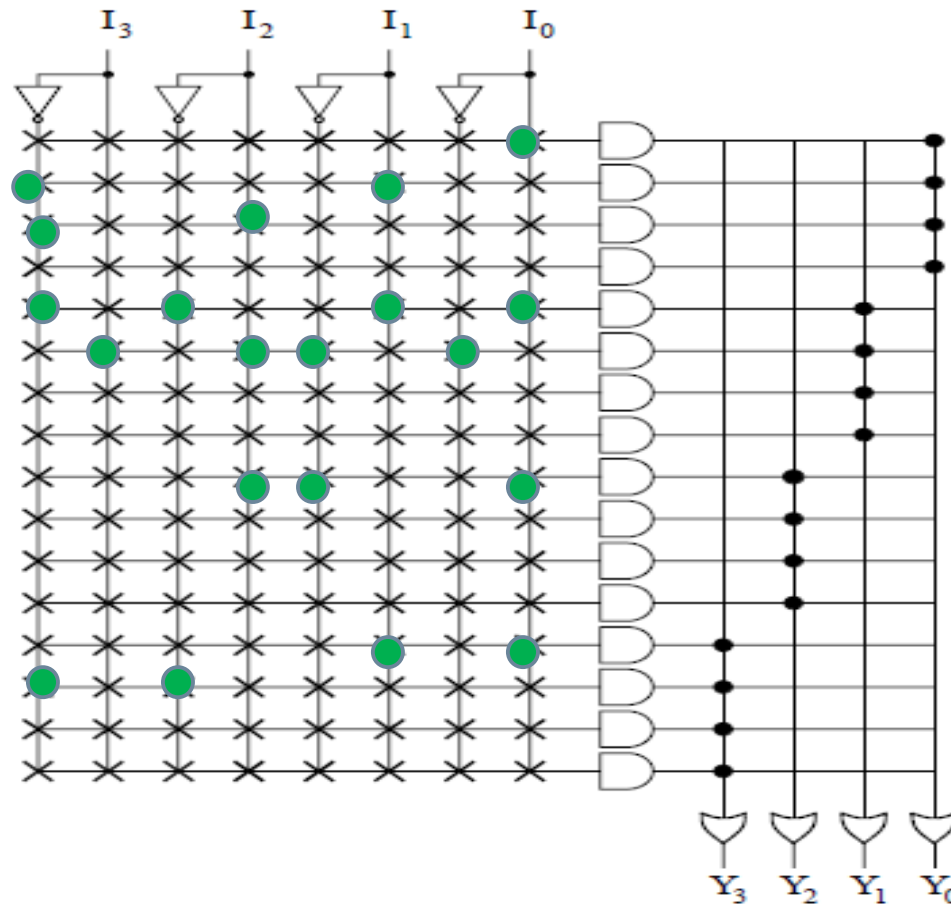
$$O_1 = ab\bar{c}\bar{d} + \bar{a}\bar{b}cd$$

$$O_0 = a + b\bar{d} + c\bar{d}$$

$$I_3I_2I_1I_0 = dcba$$

$$Y_3Y_2Y_1Y_0 = O_3O_2O_1O_0$$

● Fusible intact
X Fusible fondu



PAL : Autre exemple

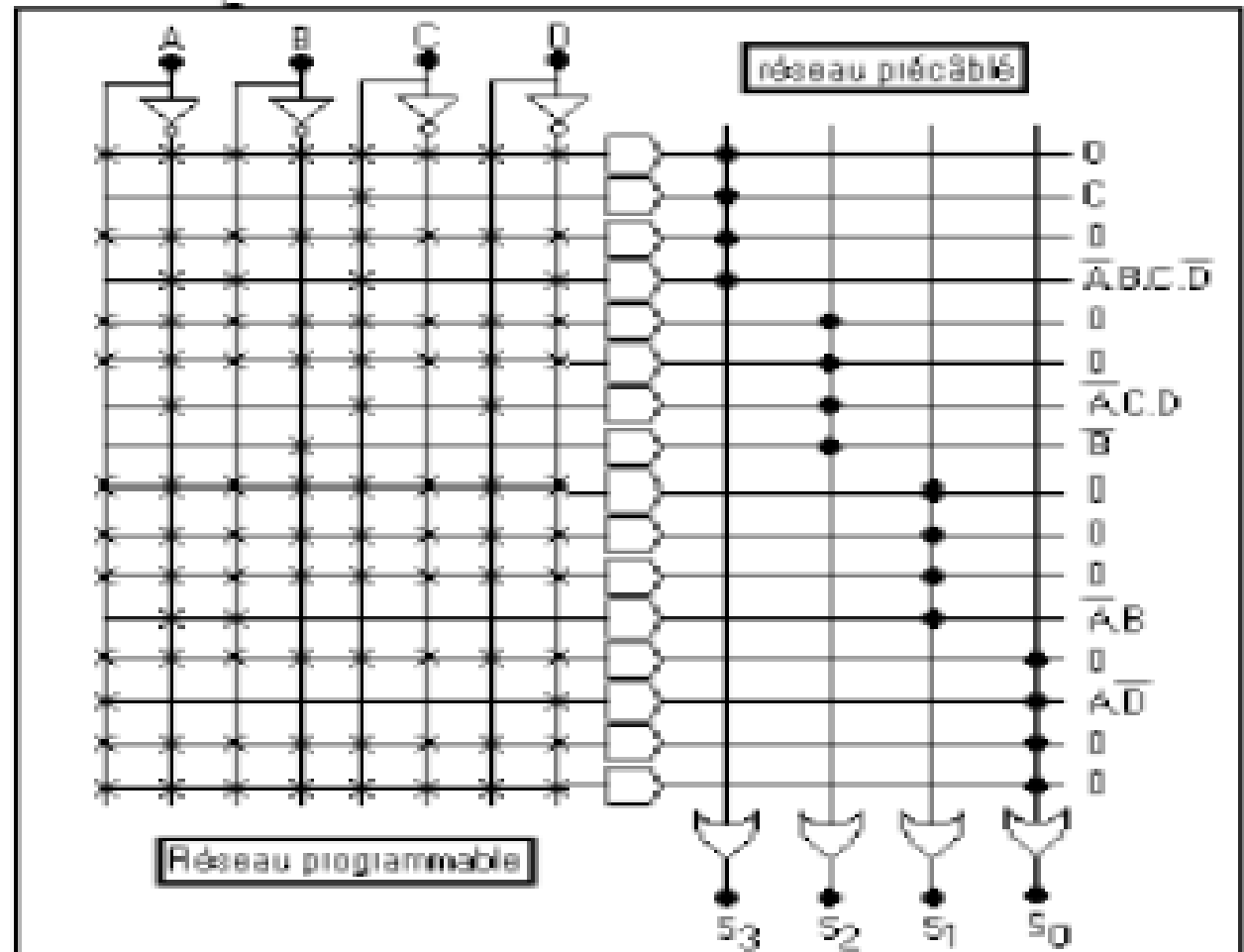
39

$$S_0 = A\bar{D}$$

$$S_1 = \bar{A}B$$

$$S_2 = \bar{A}CD + \bar{B}$$

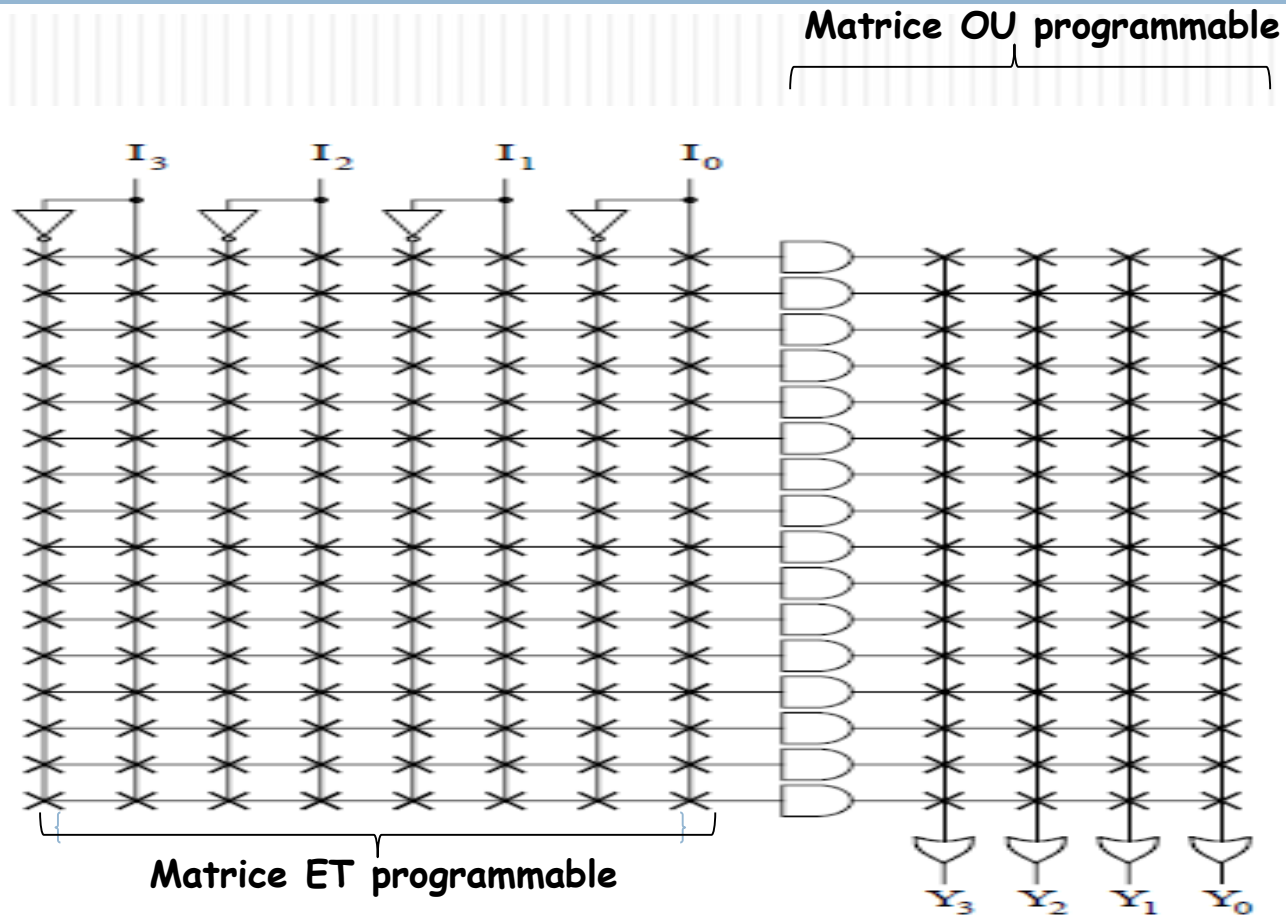
$$S_3 = C + \bar{A}BC\bar{D}$$



Un réseau logique programmable (PLA, pour programmable logic array) utilise le fait que toute fonction logique combinatoire peut se mettre sous forme d'une somme (OU logique) de produits (ET logique), c'est ce que l'on appelle classiquement la première forme normale, ou forme disjonctive.

PLA

41



Un Programmable Logic Array (PLA ou FPLA : Field Programmable Logic Array) offre le maximum de souplesse car les deux matrices des OU et ET sont programmables

14/10/2022 16:08:28

PLA

42

$$O_3 = ab + \bar{c}\bar{d}$$

$$O_2 = a\bar{b}c$$

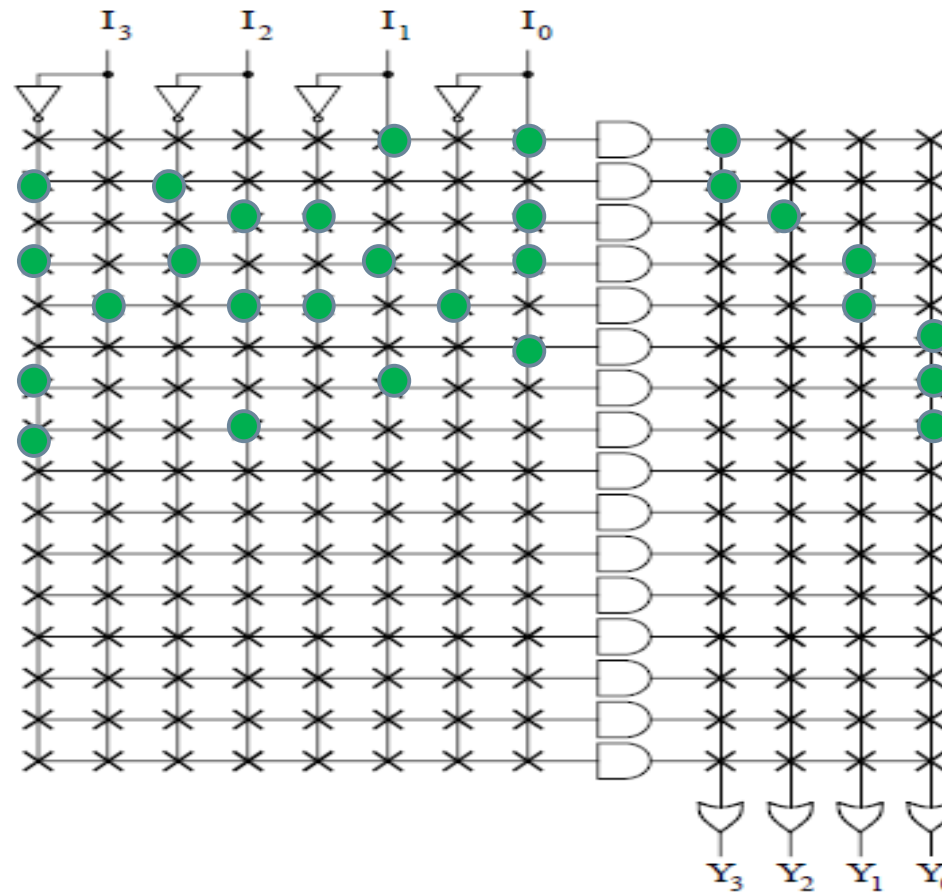
$$O_1 = ab\bar{c}\bar{d} + \bar{a}\bar{b}cd$$

$$O_0 = a + b\bar{d} + c\bar{d}$$

$$I_3I_2I_1I_0 = dcba$$

$$Y_3Y_2Y_1Y_0 = O_3O_2O_1O_0$$

● Fusible intact
X Fusible fondu



Autres exemples

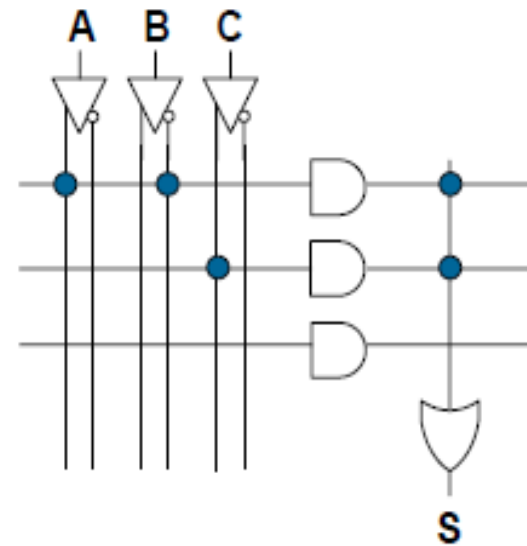
43

□ Exemple 1 :

$$S = A.\bar{B} + C$$

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Implémentation sur un
PLA à 3 variables d'entrée



Autres exemples

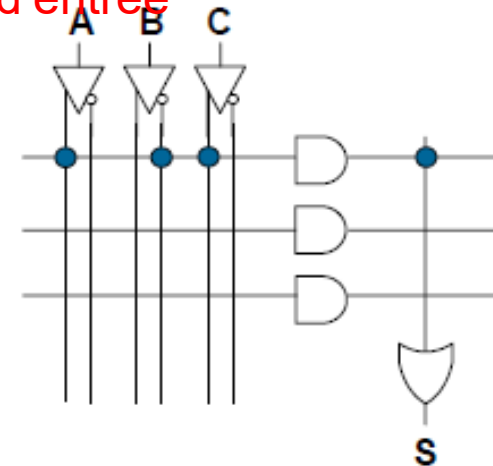
44

□ Exemple 2

$$S = A \cdot \bar{B} \cdot C$$

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Implémentation sur un
PLA à 3 variables
d'entrée



Autres exemples

45

❑ Exemple 3 :

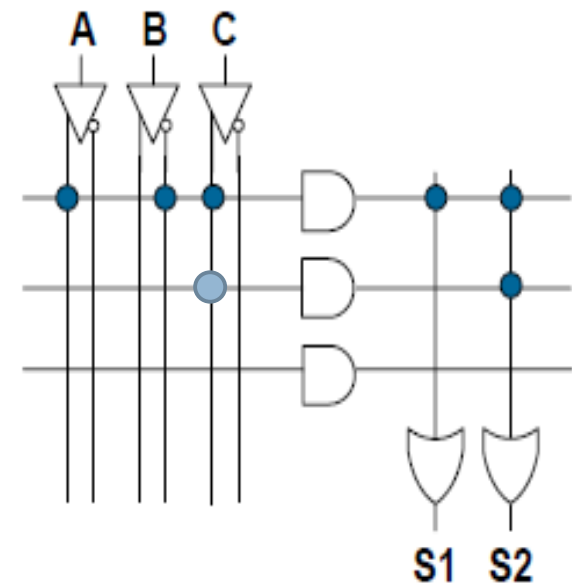
$$S1 = A \cdot \overline{B} \cdot C$$

$$S2 = A \cdot \overline{B} + C$$

A	B	C	S1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

A	B	C	S2
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Implémentation sur un
PLA à 3 variables d'entrée

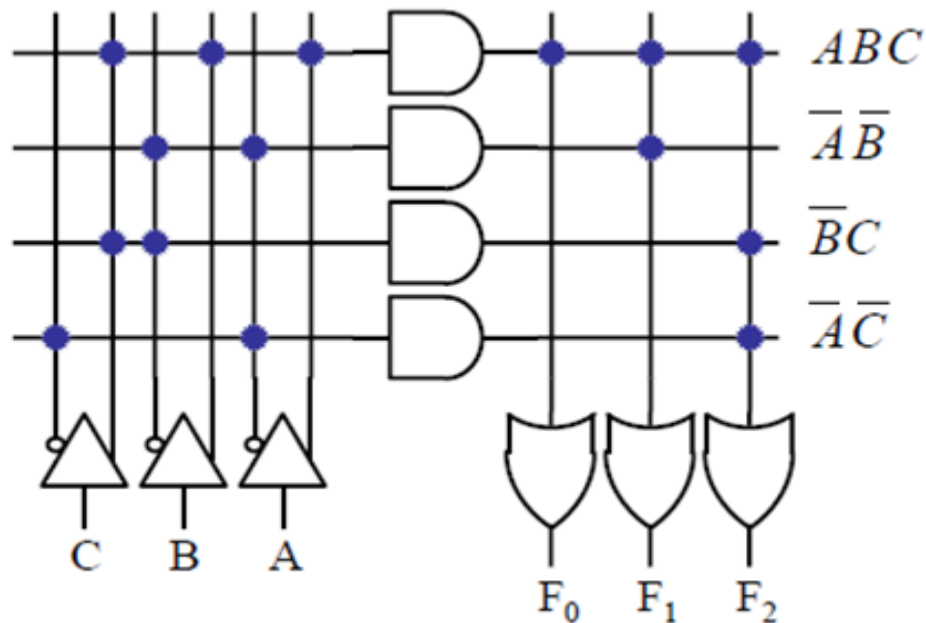


Autres exemples

46

❑ Exemple 4

- ❑ PLA : Programmable Logic Arrays (3 variables d'entrée et 3 sorties)
- ❑ Matrices ET-OU : Somme de produits



$$F_0 = ABC$$

$$F_1 = ABC + \bar{A}\bar{B}$$

$$F_2 = ABC + \bar{B}C + \bar{A}\bar{C}$$

Compromis

47

	Flexibilité	Performance
PROM	+	++
PAL	+	++
PLA	++	-

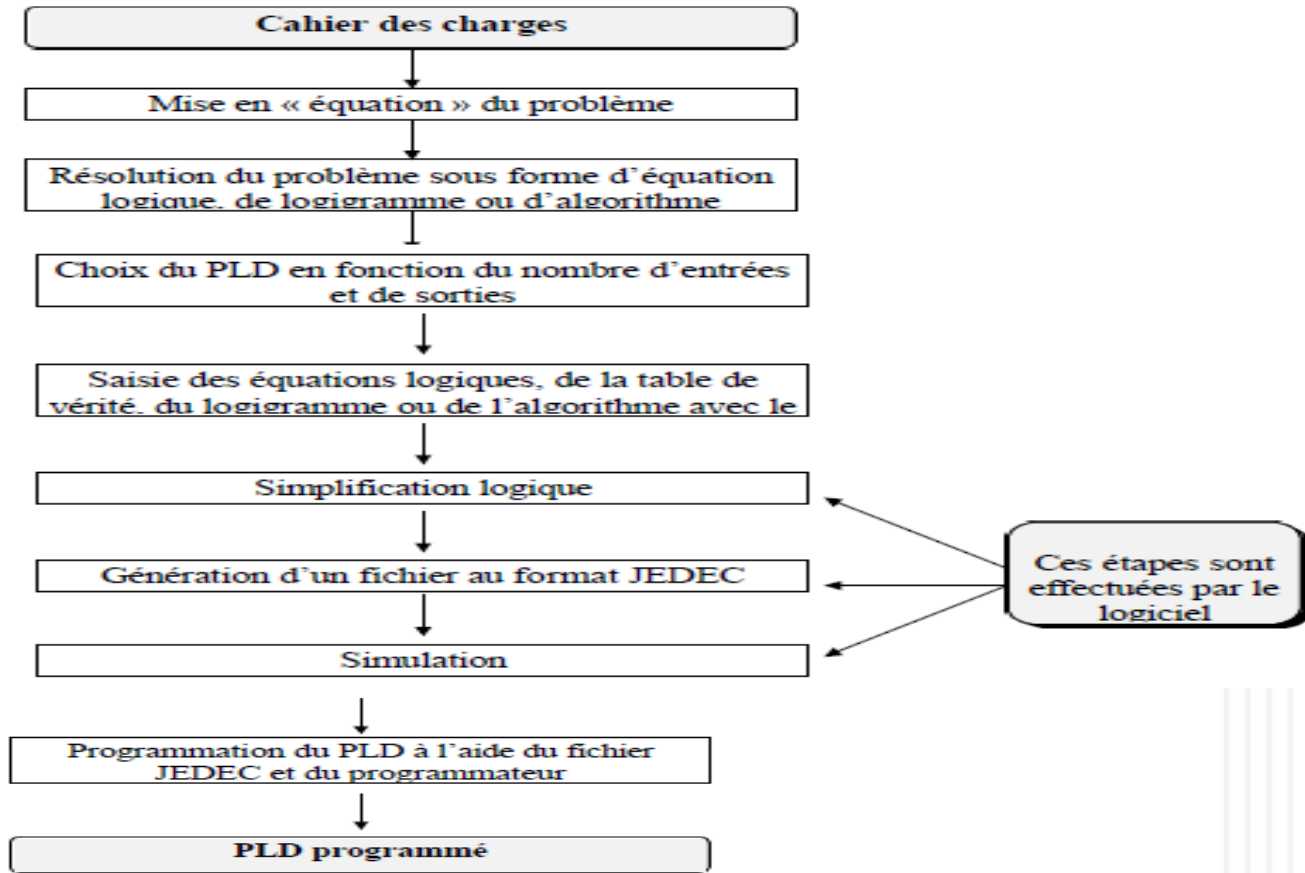
Flot de conception

48

- ❑ La programmation des PLDs nécessite un logiciel adapté pour le développement du programme et un programmeur permettant de « griller » les fusibles du circuit.
- ❑ En outre il est conseillé de suivre la démarche décrite par l'organigramme suivant :

Flot de conception

49



Systèmes de développement

50

- ❑ Ces systèmes produisent une table représentant les fusibles à détruire en fonction :
 - ❑ des équations logiques,
 - ❑ diagramme d'états
 - ❑ et tables de vérités écrit dans le langage propre au système, c'est le rôle du compilateur ou synthétiseur.
- ❑ La description du fonctionnement des circuits peut se faire de plusieurs façons, soit :
 - ❑ Par un schéma à base de fonctions logiques élémentaires (Portes ET,OU,NON, ... bascules, compteurs, registres à décalages).
 - ❑ En utilisant un langage de description comportementale H.D.L. (Hardware Description Language).

Systèmes de développement

51

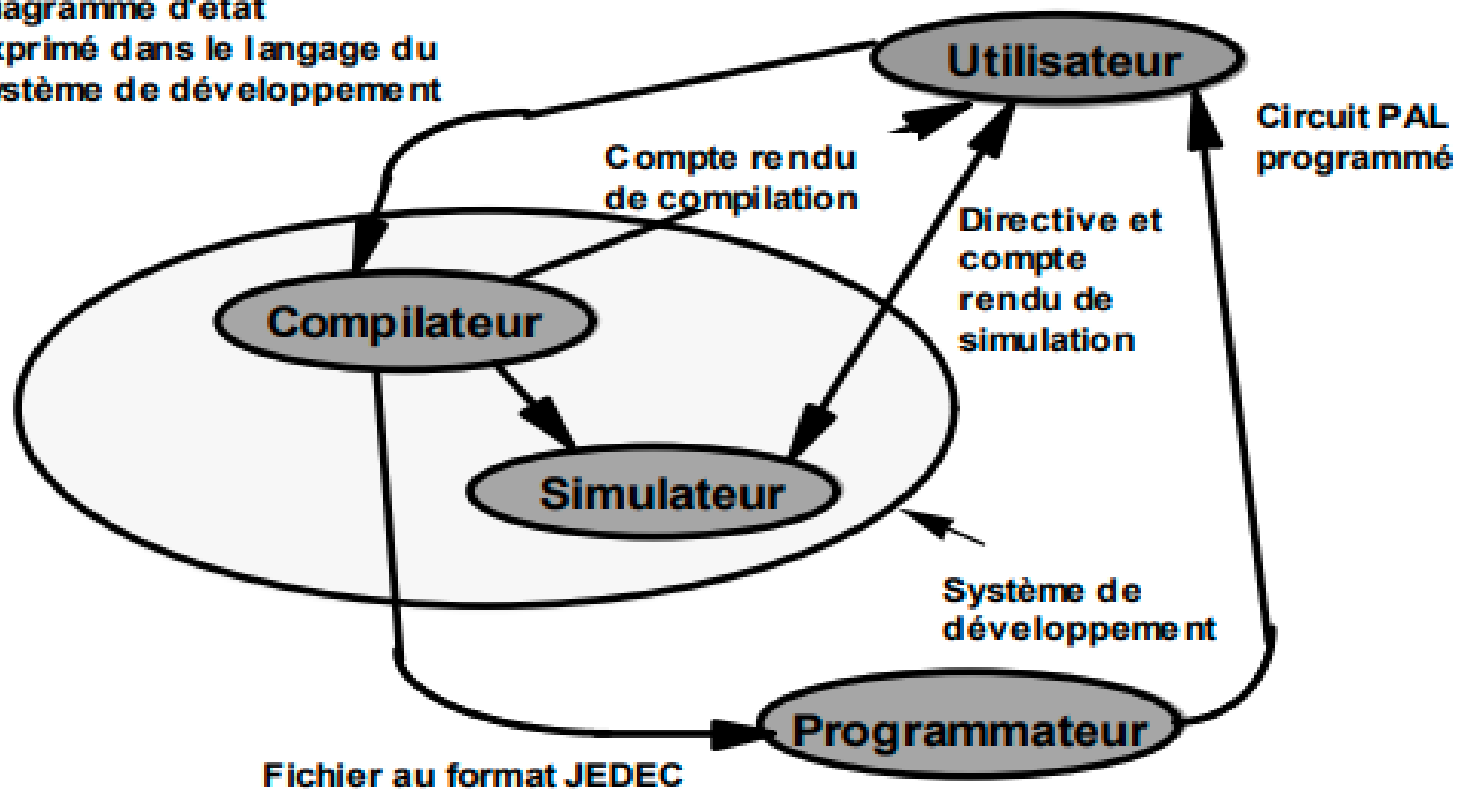
Les plus anciens sont PALASM, ORCAD/PLD et le plus connu et utilisé est sans conteste ABEL (utilisé par la plupart des systèmes de développement). Enfin les langages dits de haut niveau, VHDL (Very high speed Hardware Description Language) et VERILOG sont en général utilisés pour des circuits complexes. Le langage VHDL est très utilisé en Europe.

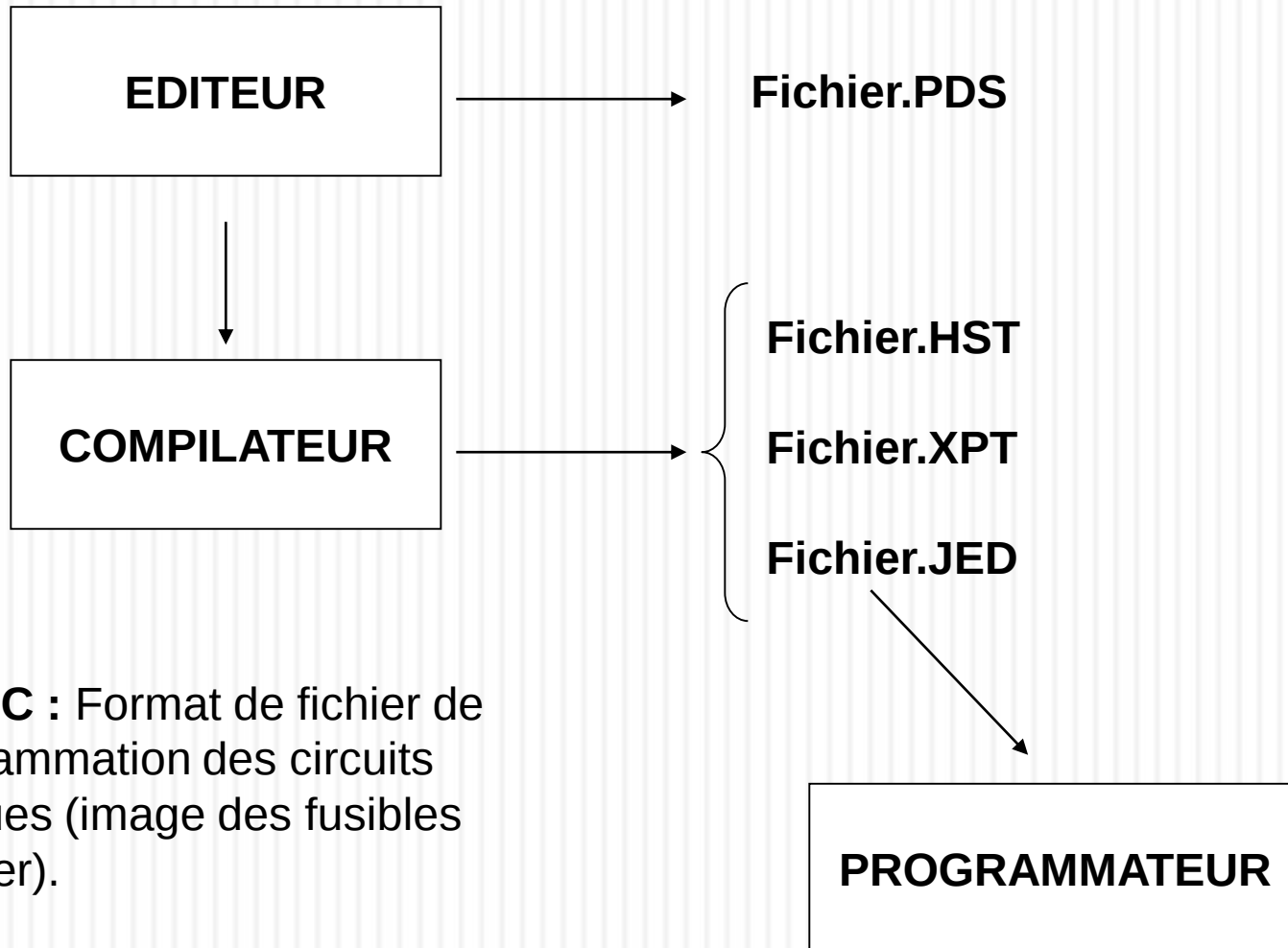
- Par l'utilisation de graphes d'états. (FSM : Flow States Machines)

Schéma fonctionnel d'un outil de développement de PAL

52

Equation, table de vérité,
Diagramme d'état
exprimé dans le langage du
système de développement





JEDEC : Format de fichier de programmation des circuits logiques (image des fusibles à griller).

Palasm

54

- Le logiciel de développement permet de simplifier les équations et de générer un fichier JEDEC à partir des données rentrées par l'opérateur.
- Il simule aussi le fonctionnement du PLD avec le programme obtenu.
- Le logiciel « PALASM » développé par « AMD » permet d'effectuer ces opérations pour la programmation de PAL ou de GAL.
- 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. Il est en général associé à un logiciel de pilotage.
- Les programmeurs utilisés sont les mêmes que ceux permettant la programmation des EPROM.

Quelques systèmes de développement

55

- **Synario** de LATTICE SEMI CONDUCTOR (Schéma, ABEL et VHDL)
WEB : <http://www.latticesemi.com>
- **Express** d'ORCAD (Schémas, VHDL) WEB : <http://www.orcad.com/> pour la France WEB : <http://www.alsdesign.fr/>
- **ViewPLD** de VIEW LOGIC (schémas, ABEL et VHDL) WEB :
<http://www.viewlogic.com>
- **StateCad** de STATECAD (Graphes d'états). WEB :
<http://www.statecad.com/>
- **Active VHDL** de ALDEC (Schémas, VHDL et Graphes d'états). WEB :
<http://www.aldec.com/>
- **Warp** de CYPRESS (VHDL et graphes d'états). WEB :
<http://www.cypress.com/>
- **MaxPlus** d'ALTERA (Schémas, VHDL et Graphes d'états). WEB :
<http://www.altera.com/>
- **Foundation** d'XILINX (Schémas, VHDL et Graphes d'états). WEB :
<http://www.xilinx.com/>

Programmateur PLD : Leaper-56

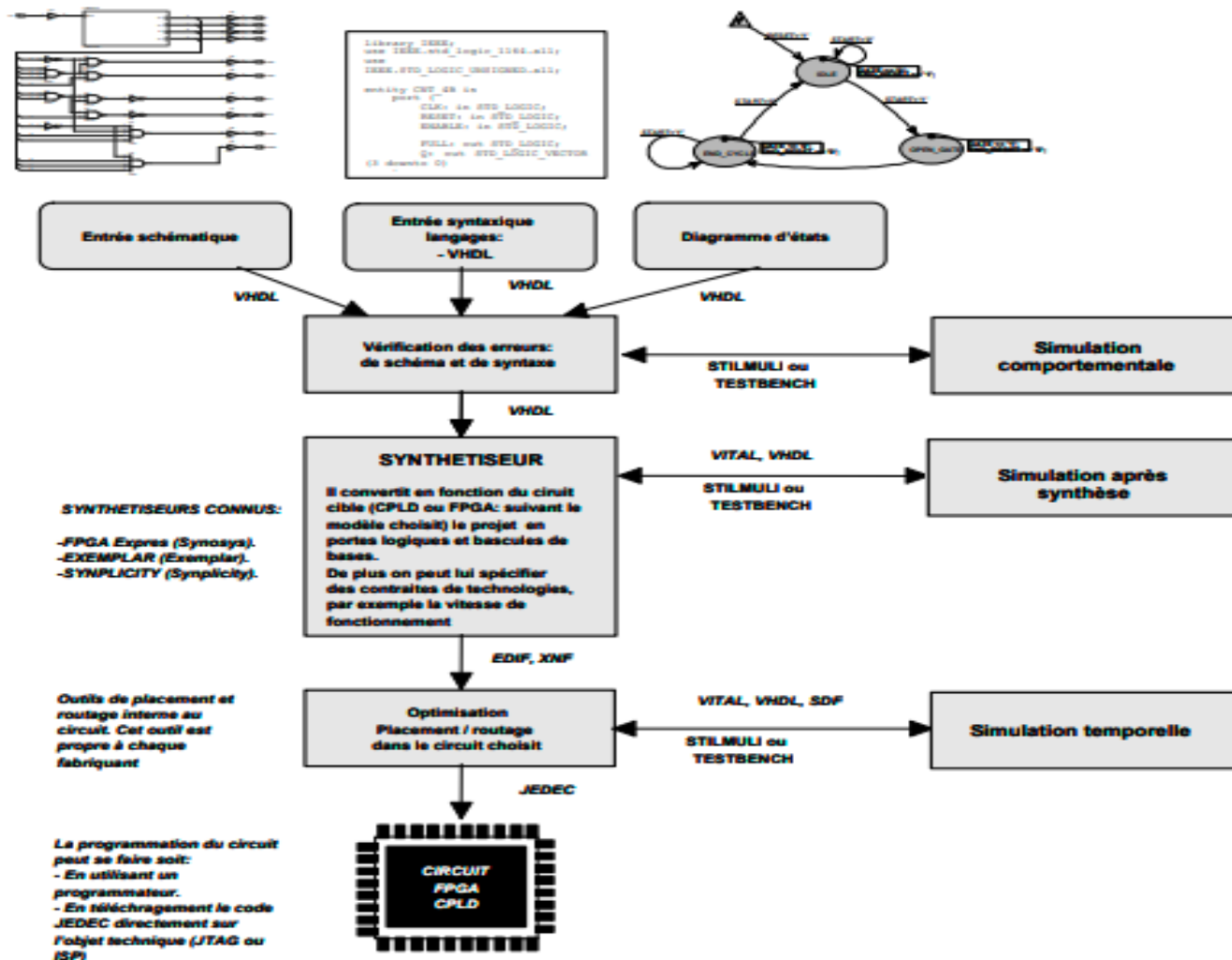
56

- Le Leaper-56 est un programmeur universel idéal pour une utilisation avec un ordinateur portable.
- Il se connecte directement sur un port USB 2.0 et ne nécessite pas d'alimentation externe pour la programmation.
- Il gère les EPROMs, EEPROMs, Mémoires FLASHs, PLD et les microcontrôleurs.
- Equipé d'un processeur pouvant gérer la programmation des composants jusqu'à une fréquence 75MHz, soit 2,5ns pour les signaux logiques, ce programmeur est extrêmement rapide pour la programmation des composants.
- Il teste également les pattes des composants insérés afin de détecter les broches défectueuses.
- Fonctionne sous Windows XP / Windows VISTA / Windows 7/8/10 (32 et 64 bits).



Schéma fonctionnel d'un outil de développement de FPGA/CPLD

57



Procédés technologiques

58

- ❑ **SRAM** - (Static Random Access Memory). Programmables à volonté et in-situ. technologie CMOS.
- ❑ **EPROM** (UVPROM) - (Erasable Programmable Read-Only Memory). Effaçables par exposition aux rayons ultra-violets. Technologie CMOS, disparition au profit de l'EEPROM.
- ❑ **EEPROM** - (Electrically Erasable Programmable Read-Only Memory). effacés et reprogrammés à volonté. peuvent être programmés in-situ (souvent par une connexion JTAG). Technologie CMOS.
- ❑ **Flash** - (Flash-erase EPROM). Mêmes propriétés que EEPROM mais avec une densité supérieure (coût inférieur pour une complexité donnée). Technologie CMOS.
- ❑ **Fusible** - Programmables une seule fois. Technologie bipolaire.
- ❑ **Anti-fusible** - Ne sont programmables qu'une seule fois. Technologie CMOS

Lexique

59

- **CPLD (Complex Programmable Logic Device)** : Désigne des PLD ayant un haut niveau d'intégration.
- **EEPROM ou E2PROM (Electrical Erasable Programmable Read-Only Memory)** : Mémoire programmable à lecture seule, effaçable électriquement.
- **EPLD (Erasable Programmable Logic Device)** : Circuits logiques reprogrammables.
- **EPROM (Erasable Programmable Read-Only Memory)** : Mémoire programmable à lecture seule, effaçable par ultraviolets.
- **FPGA (Forecasting Programmable Gate Array)** : Réseau de portes programmables à la demande. Technologie qui utilise des circuits encapsulés comportant des réseaux de portes logiques non reliées : l'utilisateur réalise les interconnexions nécessaires par programmation.
- **GAL (Generic Array Logic)** : Circuits logiques PAL reprogrammables à technologie CMOS.
- **ISP (In System Programmable)** : Circuit que l'on peut programmer (et donc effacer) même lorsqu'il est en place sur l'application.
- **JEDEC** : Format de fichier de programmation des circuits logiques (image des fusibles à griller).
- **PAL (Programmable Array Logic)** : Circuits logiques programmables dans lesquels seules les fonctions ET sont programmables, les fonctions OU ne le sont pas.
- **PLD (Programmable Logic Device)** : Famille des circuits programmables qui comprend les PAL, GAL, EPLD et FPGA.
- **PROM (Programmable Read-Only Memory)** : Mémoire programmable une seule fois à lecture seule.
- **VHDL** : Langage de programmation utilisé pour programmer les PLD.

14/10/2022 16:08:29

Récréation

60

- Q1. Quels types d'application s'y prêtent le mieux pour les PROM
Donner un exemple.
- Q2. Donner un inconvénient de l'élément PAL et PLA
- Q3. Citer une autre méthode de programmation de PLD
- Q4. Réaliser un circuit qui reçoit en entrée le code binaire d'un mois de l'année (janvier = 1, ..., décembre = 12) et active la sortie O_0 uniquement si c'est un mois d'hiver (décembre, janvier, février). O_1 uniquement si c'est un mois de printemps (mars, avril, mai) etc.
 - a) Dresser la table de vérité
 - b) A partir de cette table de vérité, marquer les fusibles à détruire dans la partie programmable de l'élément PAL.

Conclusion

61

- ❑ Les PLD ont connu des succès remarquables.
- ❑ Ces succès sont dus à la préconception des fonctions élémentaires (ET, OU, Bascule, etc.),
- ❑ Facilite la conception, réduit le temps de mise sur le marché et implique un faible coût de fabrication.
- ❑ Inconvénients :
 - ❑ limitation du nombre de portes par circuit.
 - ❑ La conception des circuits par PLD est basée sur la synthèse des fonctions sous la forme de sommes des produits SdP (SoP).

Thanks