

C) Les réseaux logiques programmables.

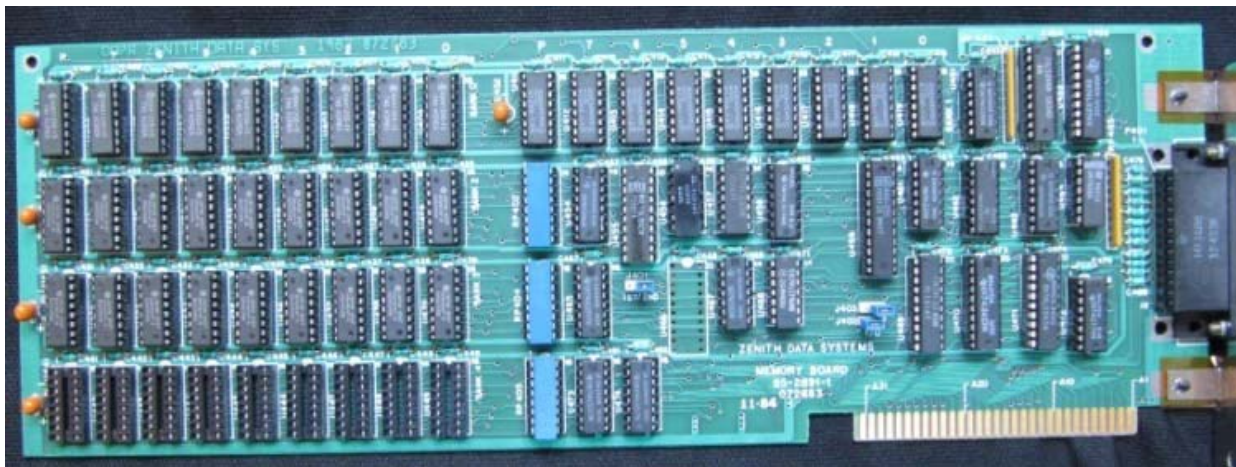
PLDs (**P**rogrammable **L**ogic **D**evelopments).

Quel est le problème qu'essaye de résoudre les PLDs?

Eviter de réaliser les fonctions logiques en utilisant les portes logiques de base, parce que ceci donne naissance à des solutions non optimisées sur divers plans:

- Espace
- Souplesse de modification.
- Coût.

L'idéal serait d'avoir ce qui est appelé "**The single chip solution**", **modifiable à volonté, là où le circuit réside.**



C) Les réseaux logiques programmables. Les réponses.

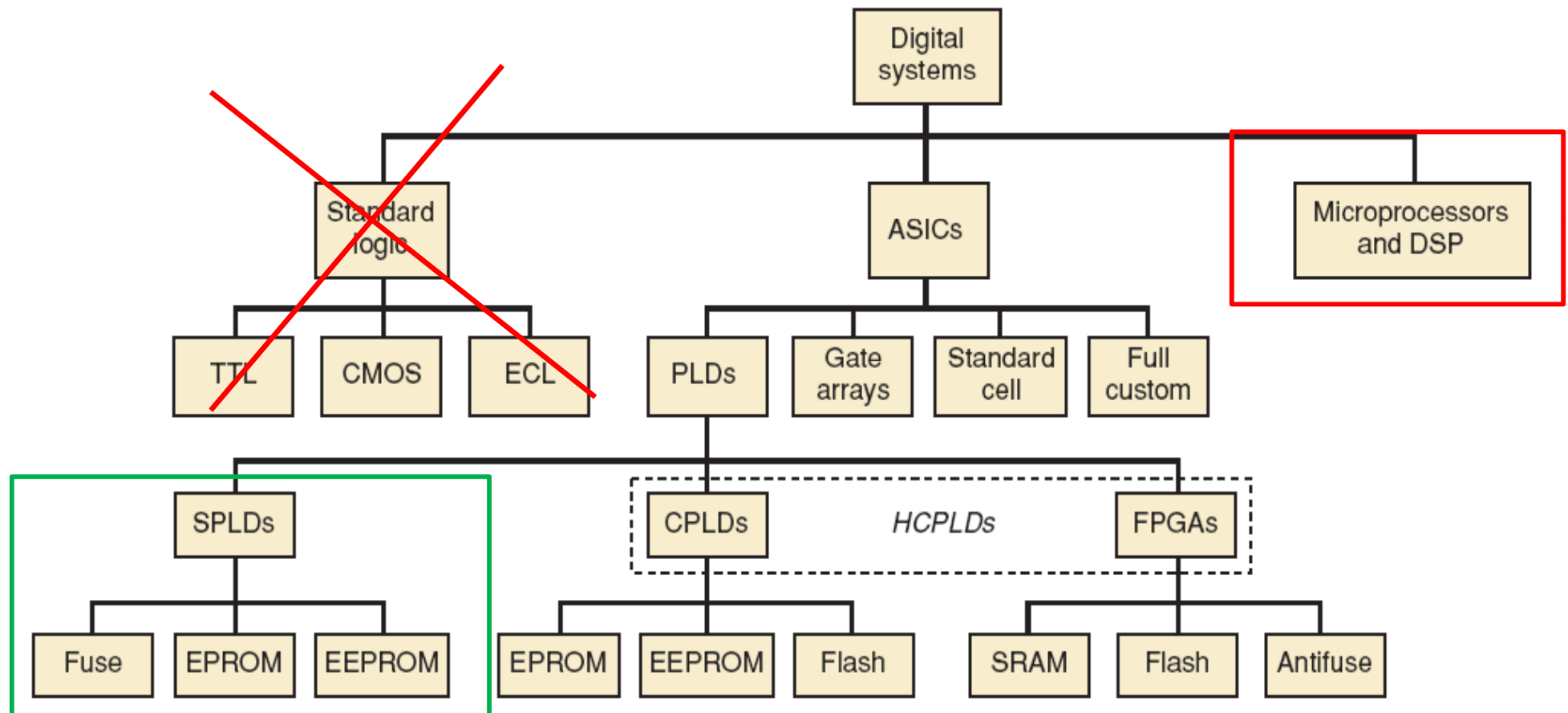


FIGURE 13-1 Digital system family tree.

Source: Ronald J. Tocci's Book

C) Les réseaux logiques programmables.

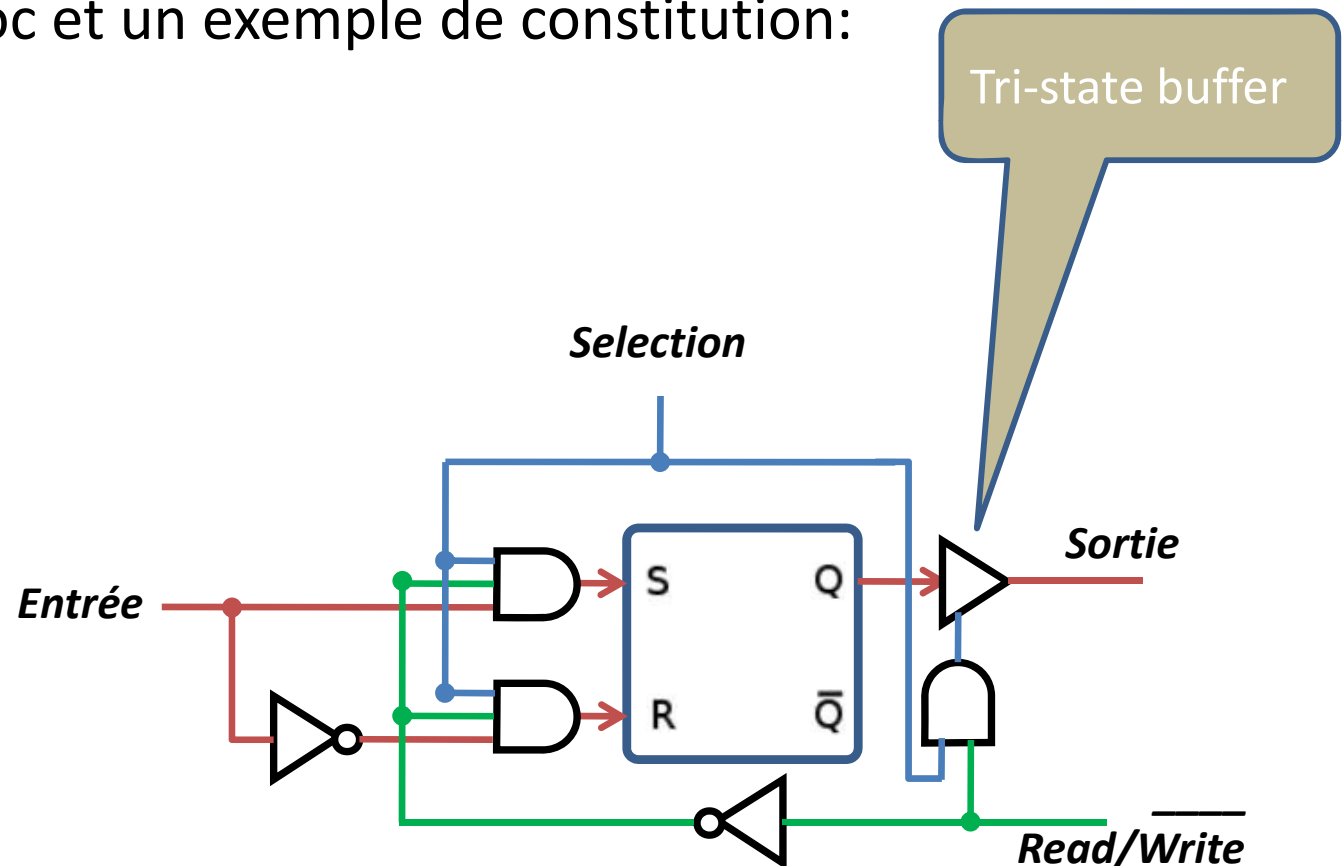
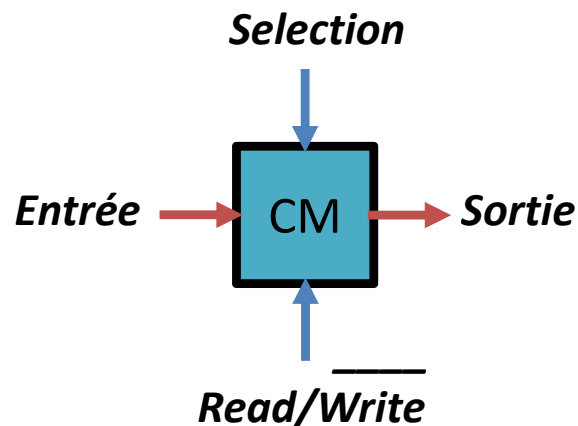
C-1: Les Mémoires (à semi-conducteurs. Il n'est pas question de disques).

Question: c'est quoi **un circuit** mémoire?

Un circuit mémoire est un circuit dans lequel on peut **écrire** des données, et à partir duquel on peut **lire** les **dernières** données écrites.

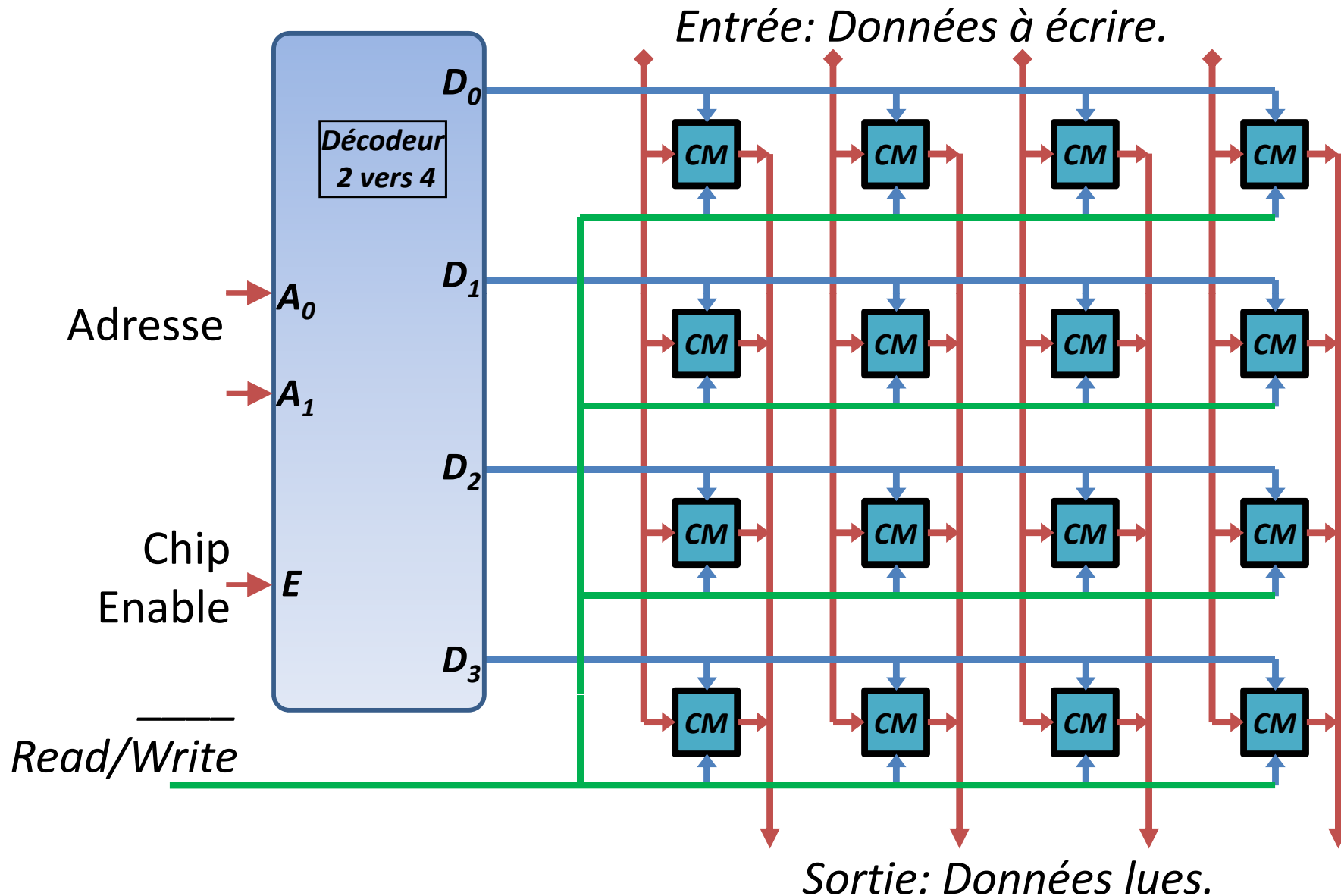
À la base il y a la **Cellule Mémoire** (CM)(1 bit)

Ci-après son schéma bloc et un exemple de constitution:



C) Les réseaux logiques programmables.

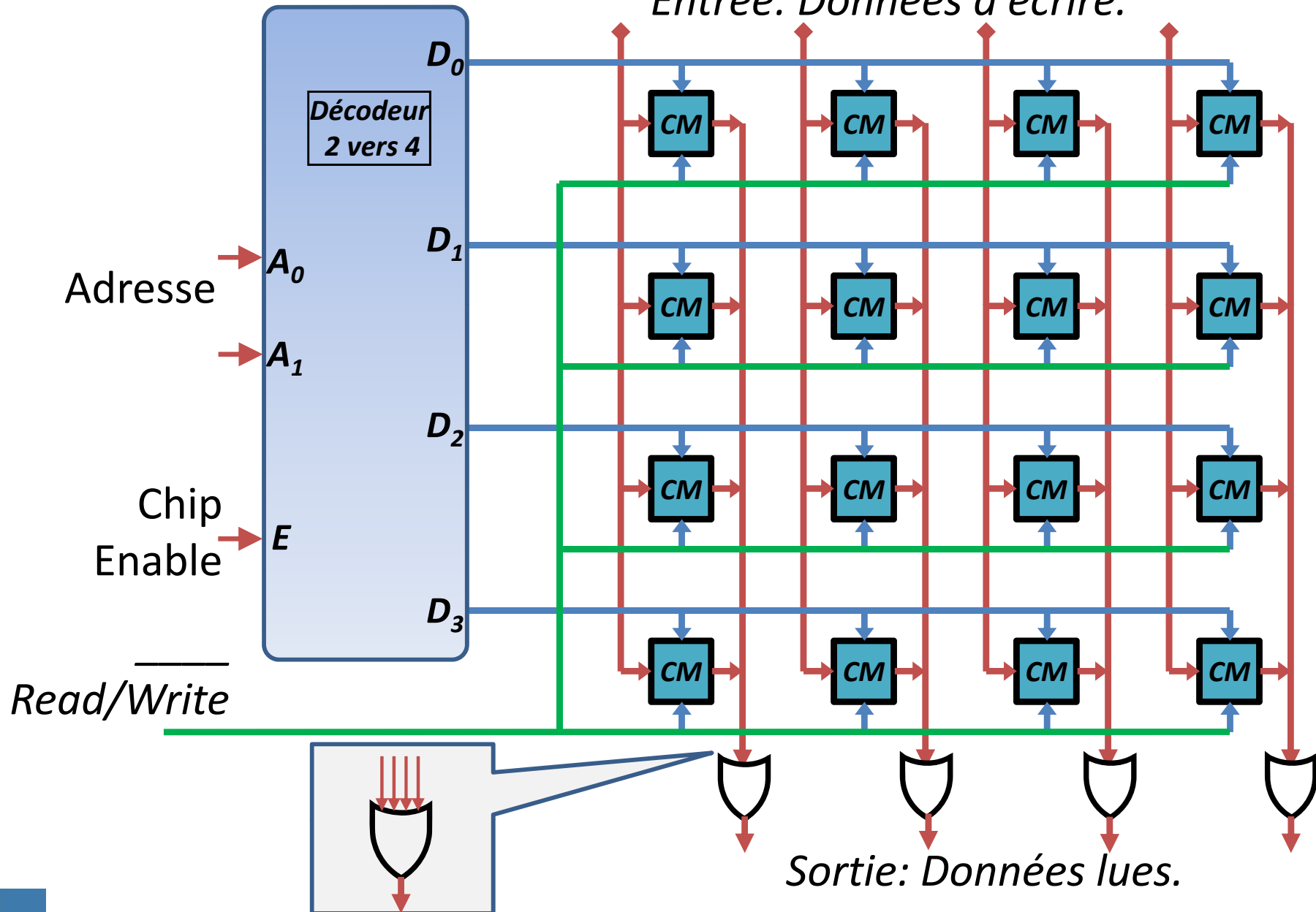
C-2: Exemple d'un circuit mémoire (RAM): 4 Mots de 4 bits chacun.



C) Les réseaux logiques programmables.

C-3: Exemple d'un circuit mémoire (RAM): 4 Mots de 4 bits chacun.

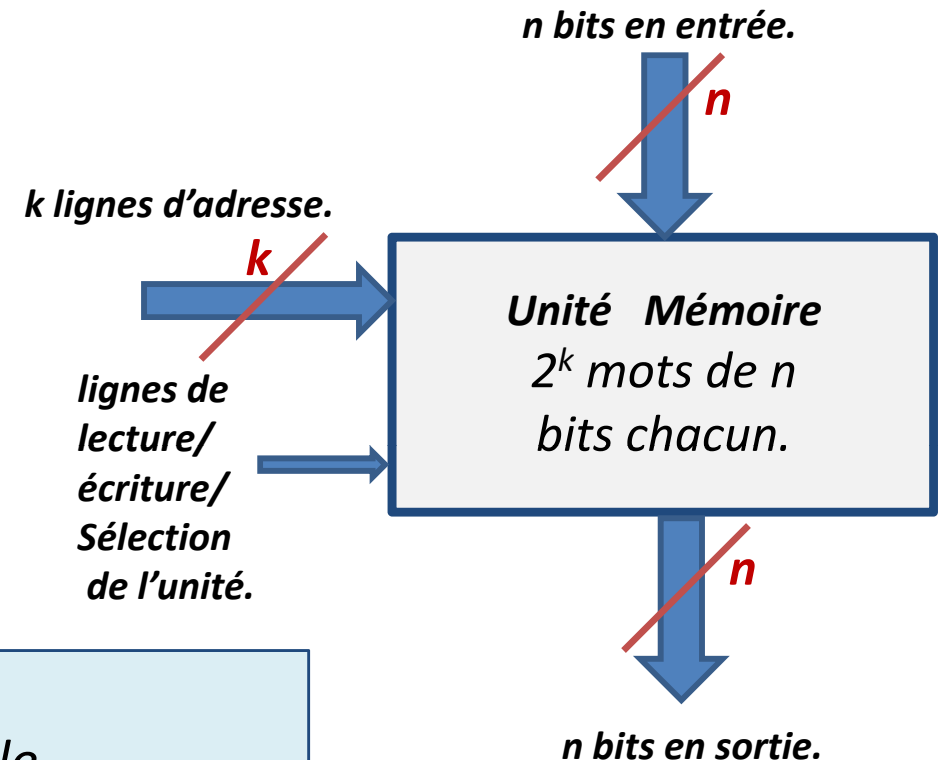
Entrée: Données à écrire.



C) Les réseaux logiques programmables.

C-4: type de mémoire.

- ✓ n lignes de données
appelées généralement bus de donnés.
- ✓ k lignes d'adresse
appelées généralement bus d'adresse.
- ✓ Les bus de donnés en entrée et en sortie
peuvent être séparés ou confondus
(bidirectionnels)
- ✓ Capacité ou taille : $2^k \text{ mots} \times n \text{ bits}$.



- Deux types de mémoire
 - ❖ La **RAM** (Random Access Memory) *volatile*.
 - ❖ La **ROM** (Read Only Memory). *Non volatile*.
 - Factory rom ou **mask-programmed ROM**.
 - PROM Programmable ROM.
 - EPROM REEPROM Erasable Reprogrammable ROM
 - EEROM Electrically Erasable Programmable ROM.

C) Les réseaux logiques programmables.

C-5: Exemple de contenu mémoire. (différenciez adresse et contenu).

Chapter 7 Memory and Programmable Logic

Memory address		Memory content
Binary	Decimal	
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
	*	*
	*	*
	*	*
	*	*
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

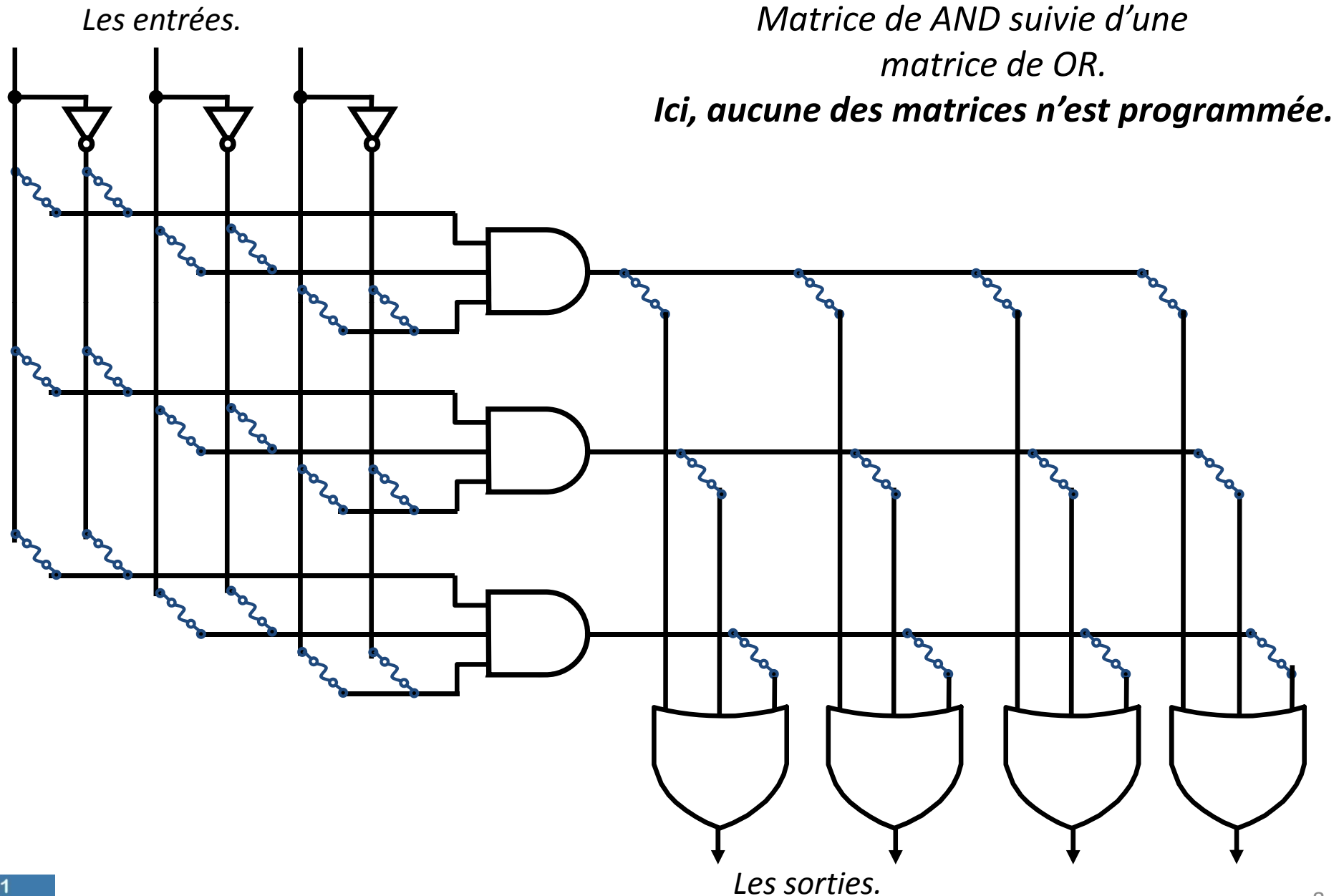
Source:
Mano's
Book.

FIGURE 7.3

Contents of a 1024×16 memory

C) Les réseaux logiques programmables.

C-6: Schéma générique, détaillé, d'un PLD simple.

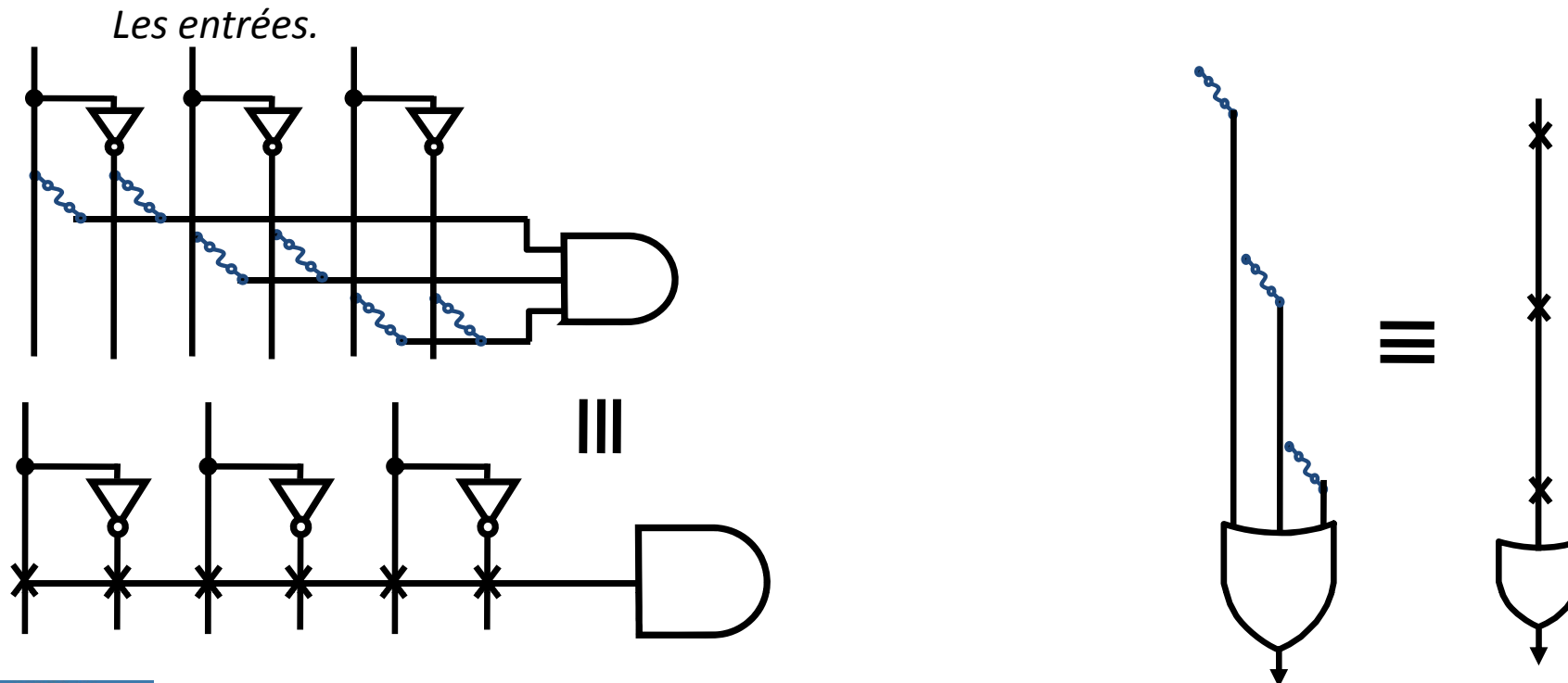


C) Les réseaux logiques programmables.

C-7: Schéma générique d'un PLD simple et équivalence.

- ✓ Un fusible intacte, dans la matrice des AND, fait intervenir une entrée ou son complément dans le minterme généré par la porte AND.
- ✓ Son absence est équivalent à un 1 logique, élément neutre dans le produit logique.
- ✓ Un fusible intacte, dans la matrice des OR de sortie, fait intervenir un minterme dans la somme générée par la porte OR.
- ✓ Son absence est équivalent à un 0 logique, élément neutre dans la somme logique.

les schémas des PLD simples seront représentés comme suit:



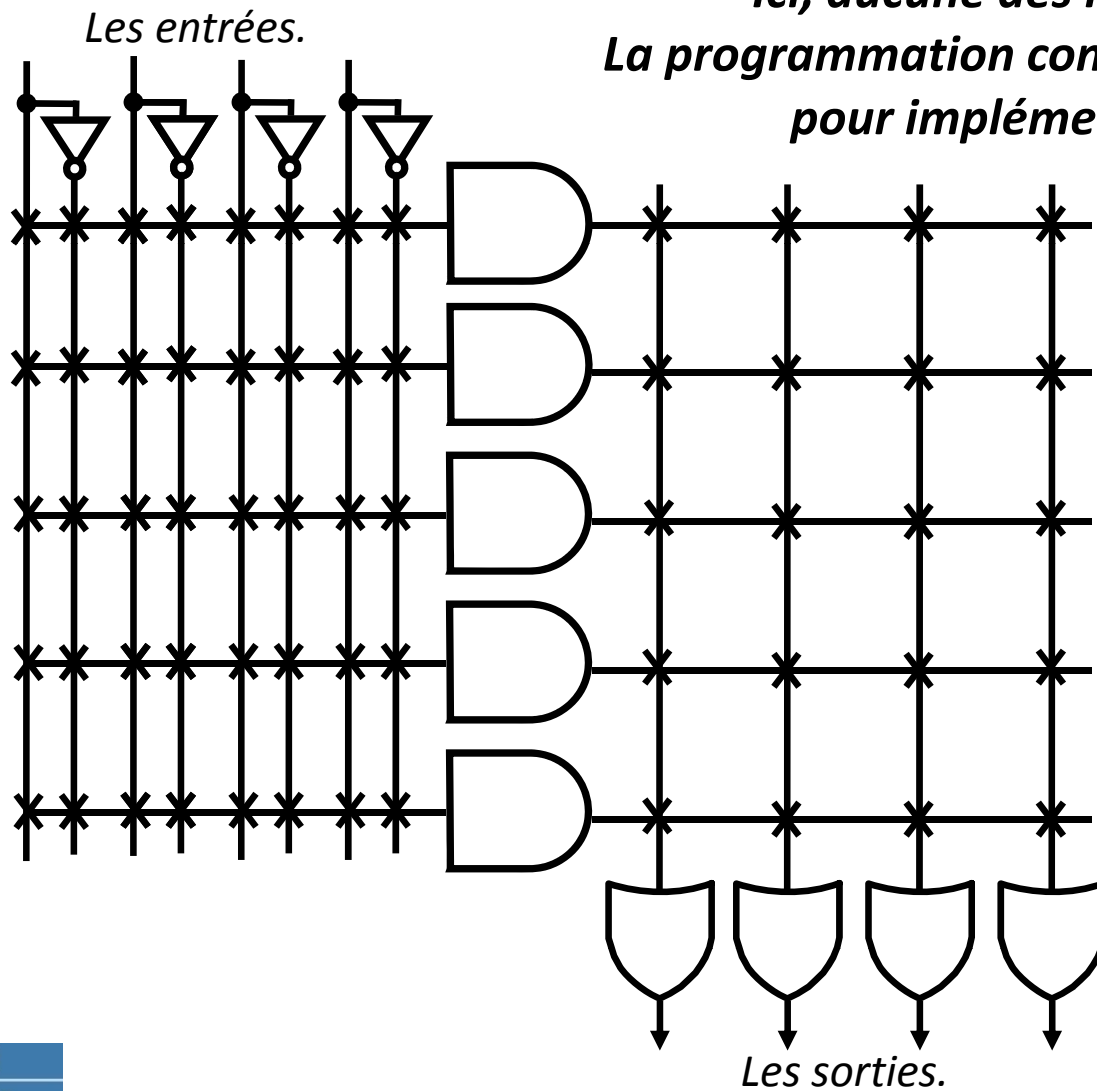
C) Les réseaux logiques programmables.

C-8: Schéma générique, succinct, d'un PLD simple.

*Matrice de AND suivie d'une
matrice de OR.*

Ici, aucune des matrices n'est programmée.

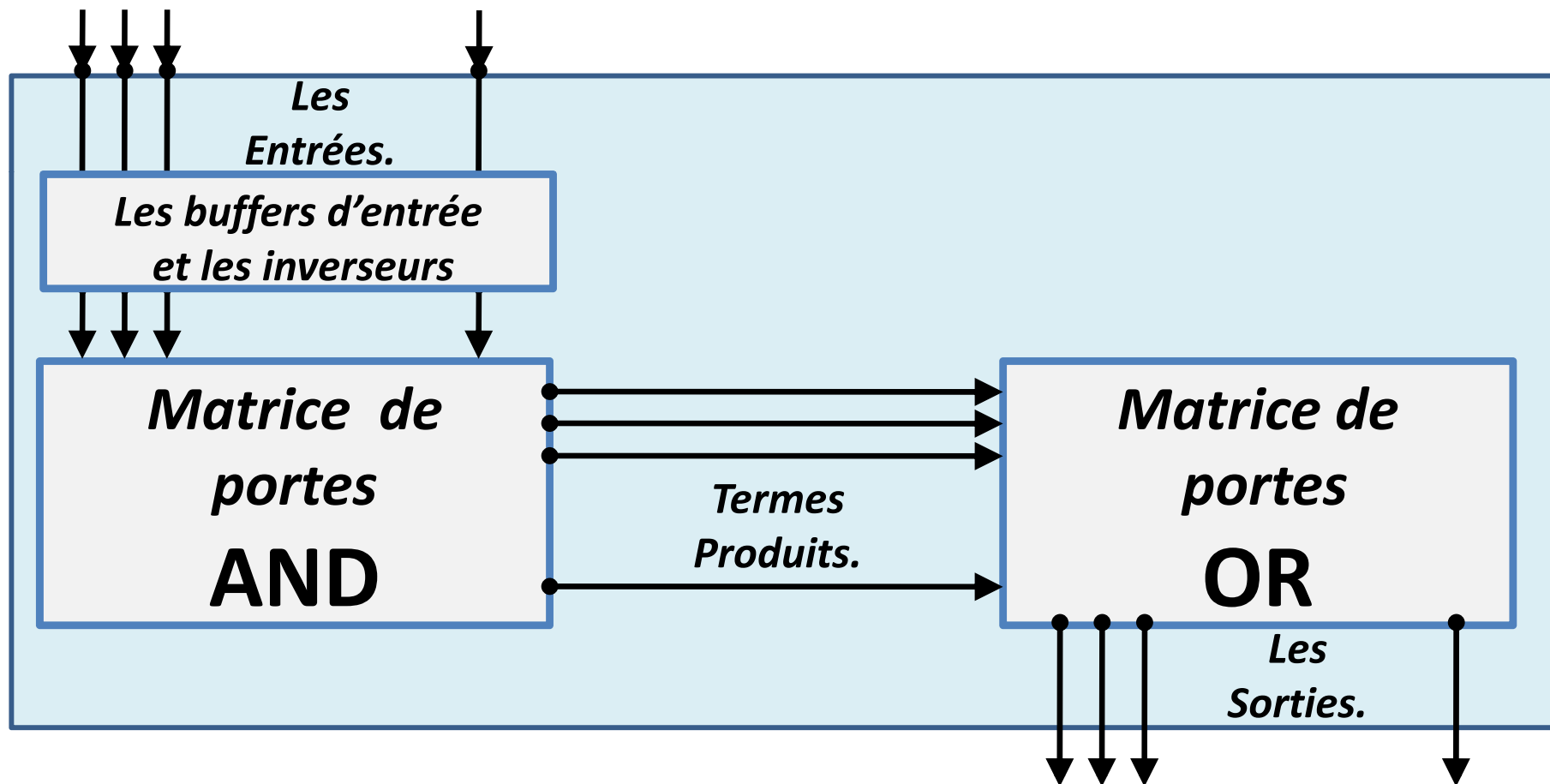
*La programmation consiste à garder ou bruler un fusible
pour implémenter une fonction logique.*



C) Les réseaux logiques programmables. Les SPLDs.

- Simple PLDs

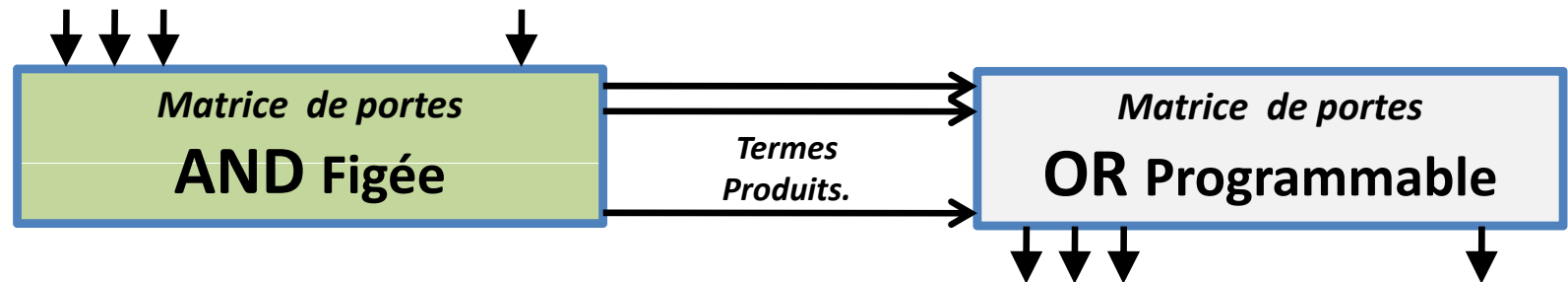
C-9: Structure générale.



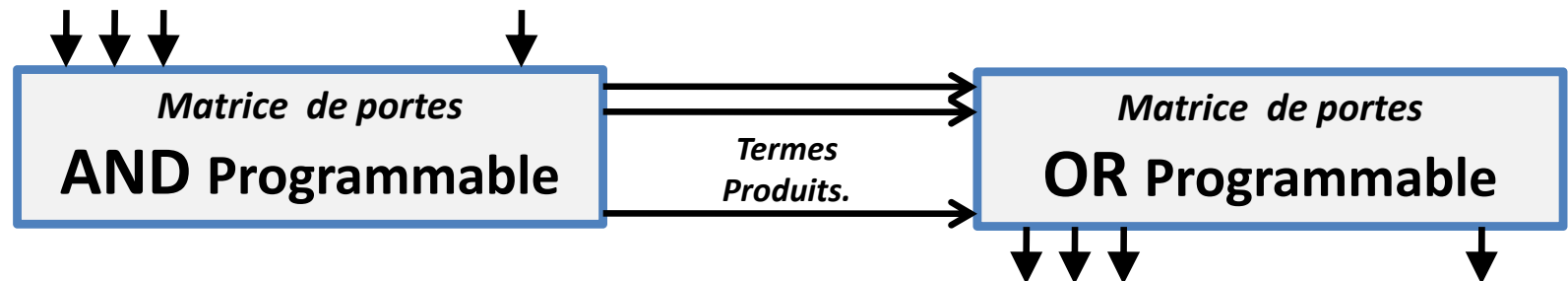
C) Les réseaux logiques programmables. Les SPLDs.

C-10: Les variantes.

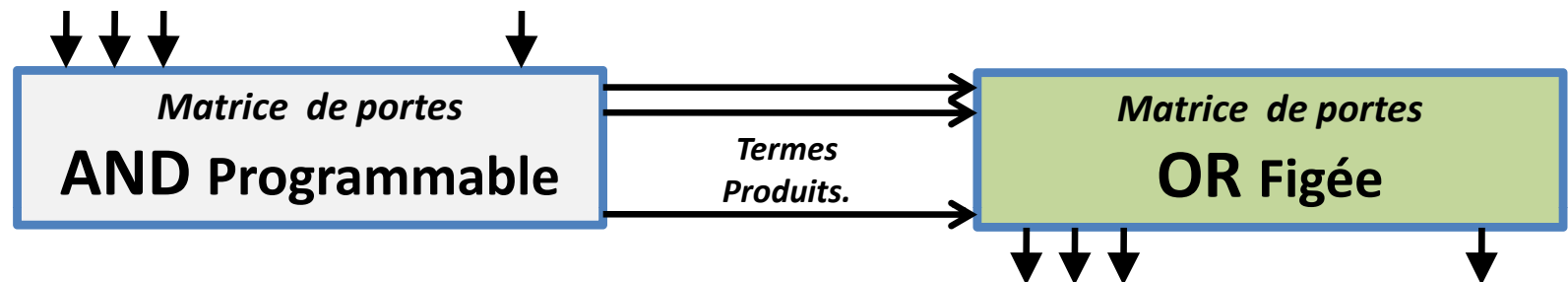
A) Les **PROMs** *Proqammable Read Only Memory.*



B) Les **PLAs** *Programmable Logic Array.*



C) Les **PALs** *Programmable Array Logic.*

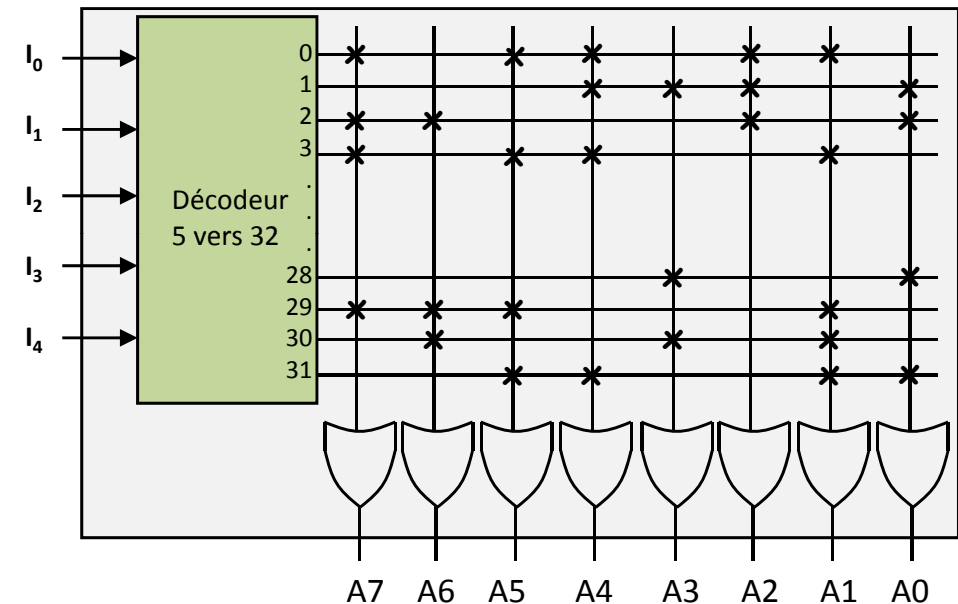


C) Les réseaux logiques programmables.

C-11: Utilisation des PROMs pour réaliser des fonctions logiques.

La table de vérité des fonctions est reportée en contenu de la ROM.

Entrées					Sorties							
I4	I3	I2	I1	I0	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		.						.				
		.						.				
		.						.				
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1



Source:
Roth's
Book.

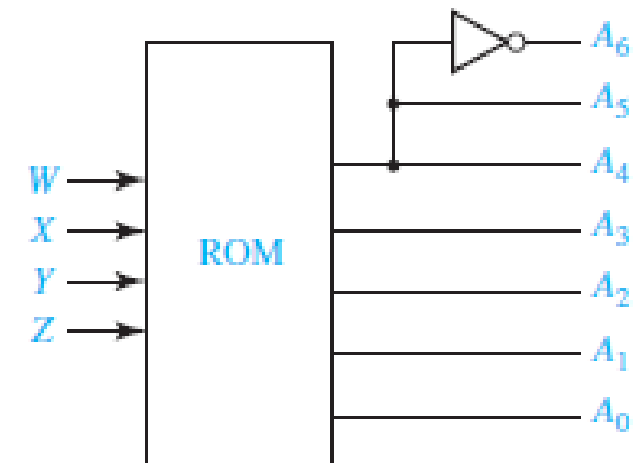
C) Les réseaux logiques programmables.

C-11: Utilisation des PROMs pour réaliser des fonctions logiques.

Exemple 1. Conversion hexadécimal – ASCII.

FIGURE 9-22
Hexadecimal-to-
ASCII Code
Converter

Input				Hex Digit	ASCII Code for Hex Digit						
W	X	Y	Z		A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	1	1	0	0	0	1
0	0	1	0	2	0	1	1	0	0	1	0
0	0	1	1	3	0	1	1	0	0	1	1
0	1	0	0	4	0	1	1	0	1	0	0
0	1	0	1	5	0	1	1	0	1	0	1
0	1	1	0	6	0	1	1	0	1	1	0
0	1	1	1	7	0	1	1	0	1	1	1
1	0	0	0	8	0	1	1	1	0	0	0
1	0	0	1	9	0	1	1	1	0	0	1
1	0	1	0	A	1	0	0	0	0	0	1
1	0	1	1	B	1	0	0	0	0	1	0
1	1	0	0	C	1	0	0	0	0	1	1
1	1	0	1	D	1	0	0	0	1	0	0
1	1	1	0	E	1	0	0	0	1	0	1
1	1	1	1	F	1	0	0	0	1	1	0



Source:
Roth's
Book.

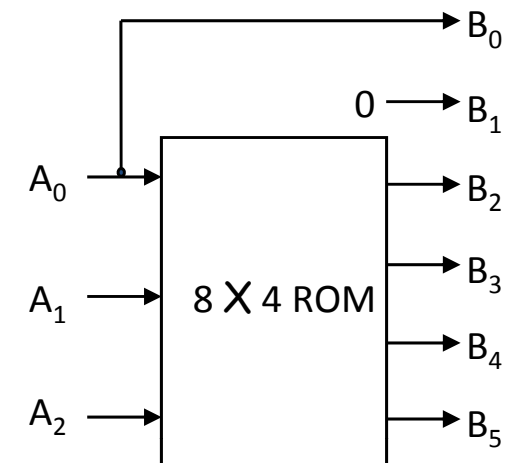
C) Les réseaux logiques programmables.

C-11: Utilisation des PROMs pour réaliser des fonctions logiques.

Exemple 2. génération des carrés de nombres de 0 à 7.

Entrées			Sorties						
A2	A1	A0	B5	B4	B3	B2	B1	B0	Carré
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

Source:
Roth's
Book.



C) Les réseaux logiques programmables.

C-11: Utilisation des PROMs pour réaliser des fonctions logiques.

Exemple 3. Calcul de la taille d'une ROM.

Problème: Trouvez la taille de la ROM (Nombre de mots mémoire et le nombre de bits par mot mémoire) pour réaliser, un additionneur soustracteur de deux données codées chacune sur 8 bits , avec prise en compte des deux retenues (entrante/sortante). Le circuit doit générer la somme et la différence.

- ✓ Les entrées (8 Bits) + (8 Bits) + retenue entrante font 17
Donc le nombre de mots de la ROM est $2^{17} = 2^7 * 2^{10} = 128 \text{ K mots}$.
- ✓ Les sorties (8 bits de somme) + (8 bits de différence) + retenue sortante font 17 bits par mot.

La capacité de la ROM: 128 K mots de 17 bits chacun.

C) Les réseaux logiques programmables.

C-12: Utilisation des **PLAs** pour réaliser des fonctions logiques.

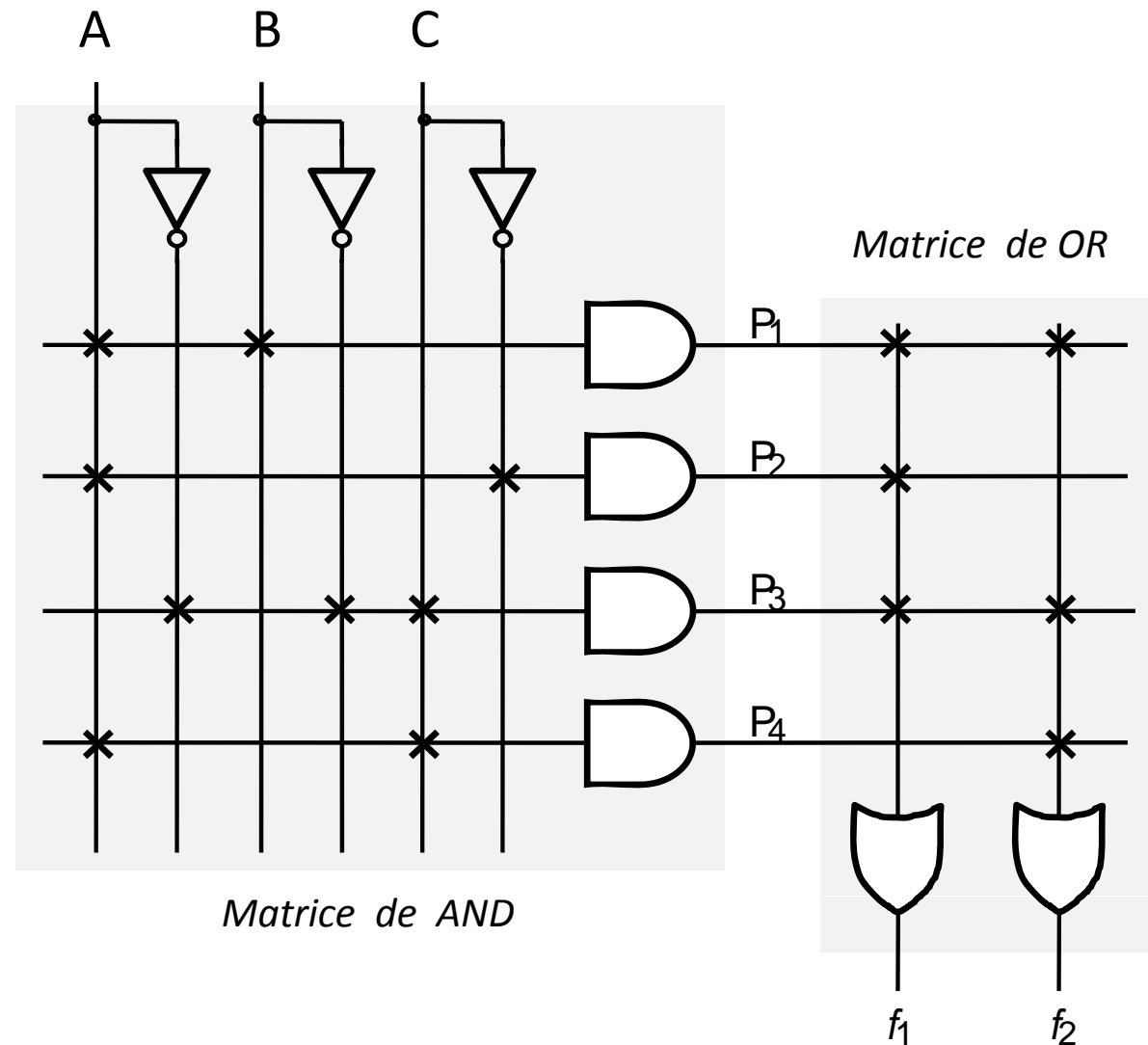
Exemple 1.

Soit à réaliser les deux fonctions suivantes:

$$f_1 = A.B + A.\bar{C} + \bar{A}.\bar{B}.C$$

$$f_2 = A.B + \bar{A}.\bar{B}.C + A.C$$

Certains produits (P_1 et P_3) sont communs aux deux fonctions, on cherchera à les faire apparaître.



C) Les réseaux logiques programmables.

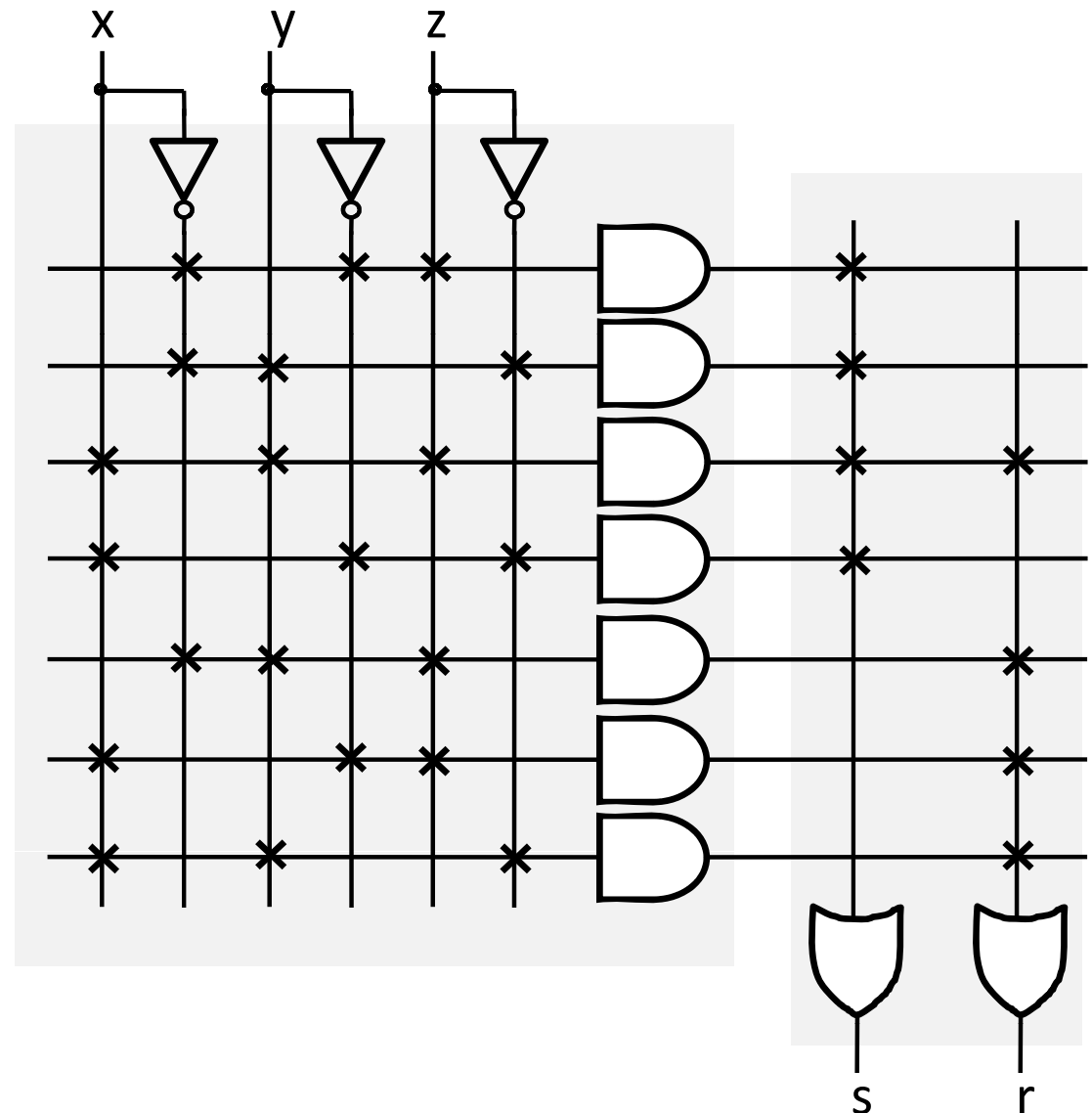
C-12: Utilisation des **PLAs** pour réaliser des fonctions logiques.

Exemple 2. réalisation de l'additionneur complet.

Rappel des équations logiques
des sorties de l'additionneur complet:

$$s = \bar{x}.\bar{y}.z + \bar{x}.y.\bar{z} + x.y.z + x.\bar{y}.\bar{z}$$

$$r = \bar{x}.y.z + x.\bar{y}.z + x.y.\bar{z} + x.y.z$$



C) Les réseaux logiques programmables.

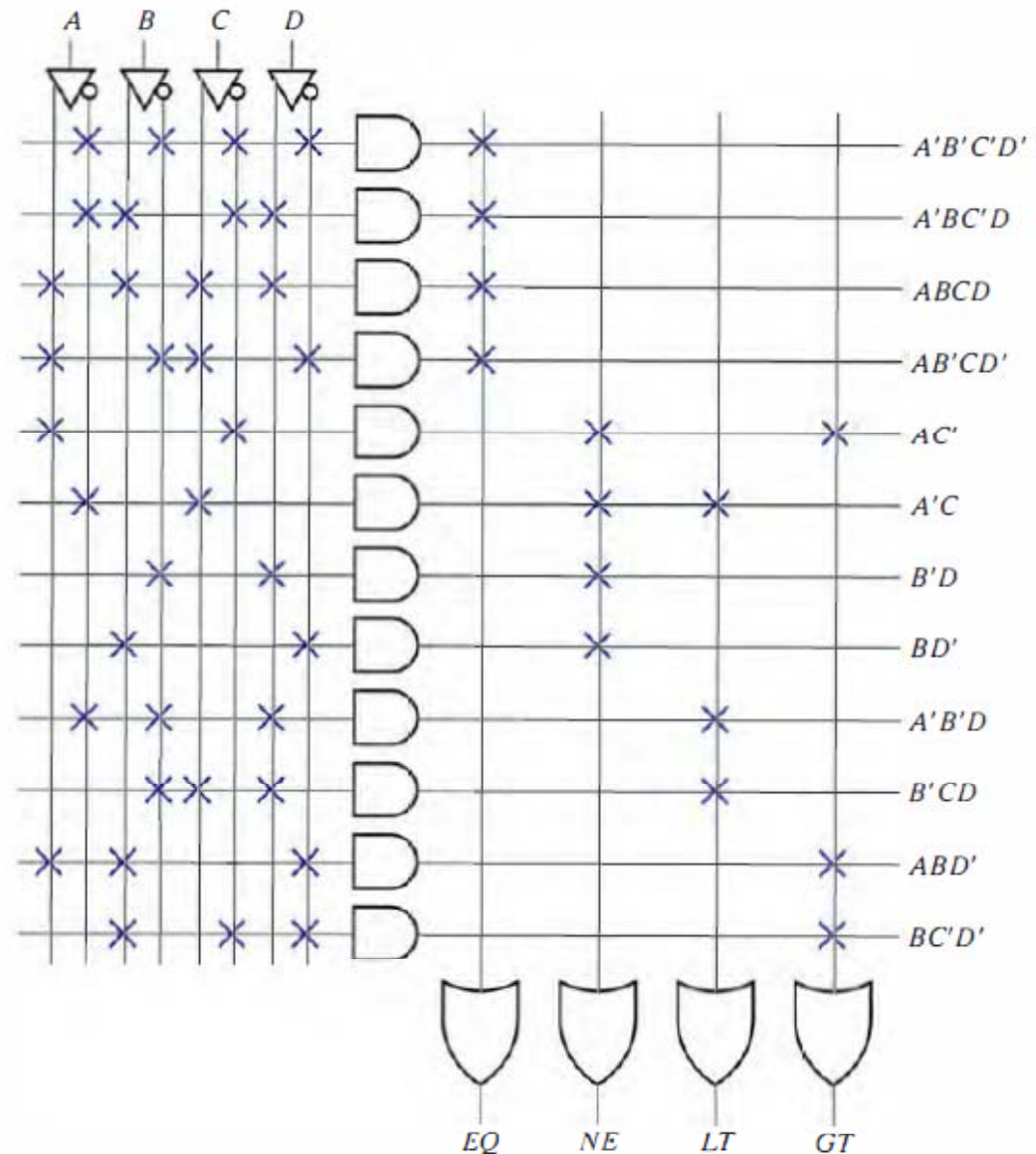
C-12: Utilisation des **PLAs** pour réaliser des fonctions logiques.

Exemple 3. Comparateur

Comparateur de deux
nombres formés de
deux bits chacun

$$N1=(AB)_2$$

$$N2=(CD)_2$$



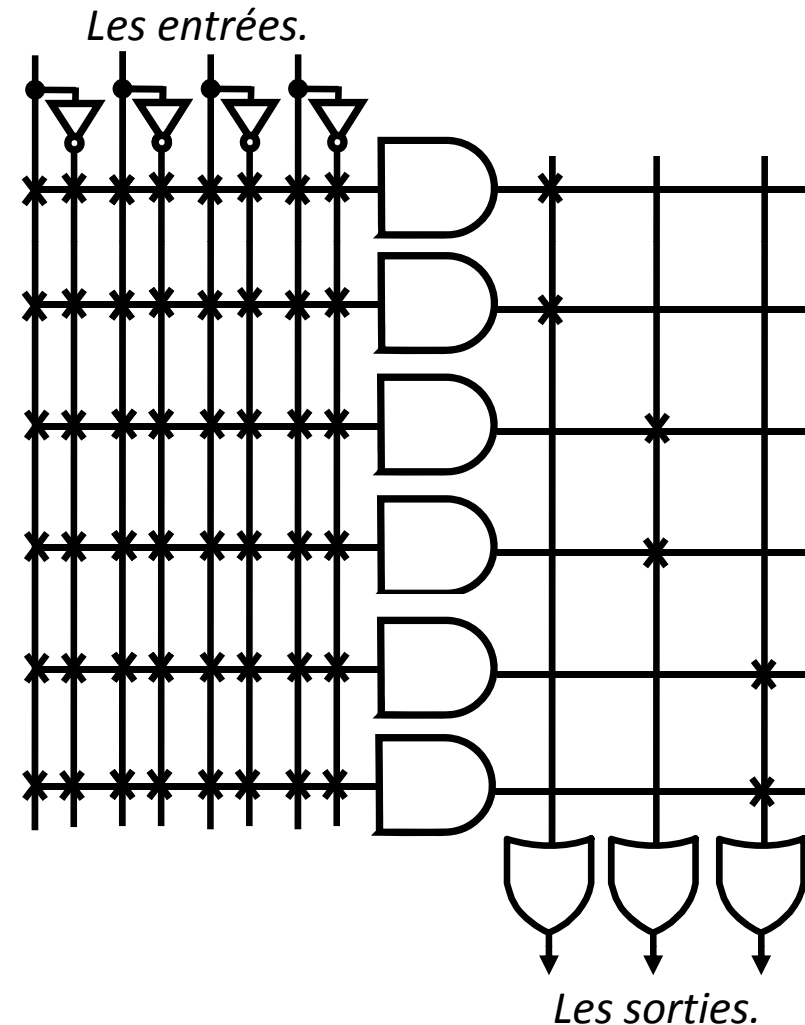
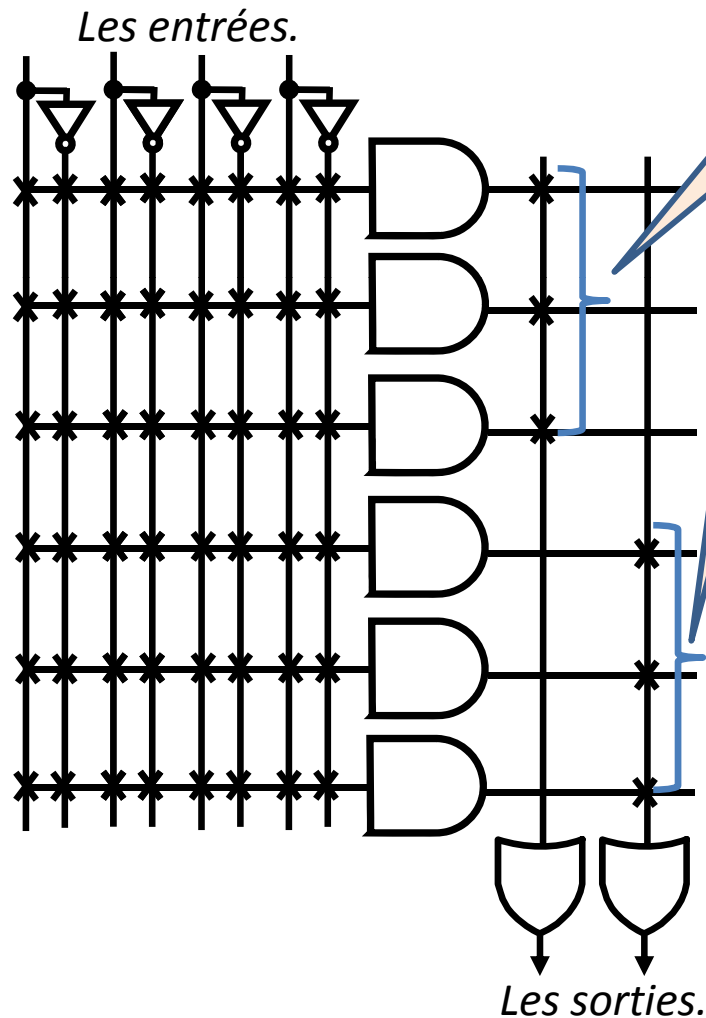
Source: Randy's Book

Figure 4.40 PLA implementation of magnitude comparator.

C) Les réseaux logiques programmables.

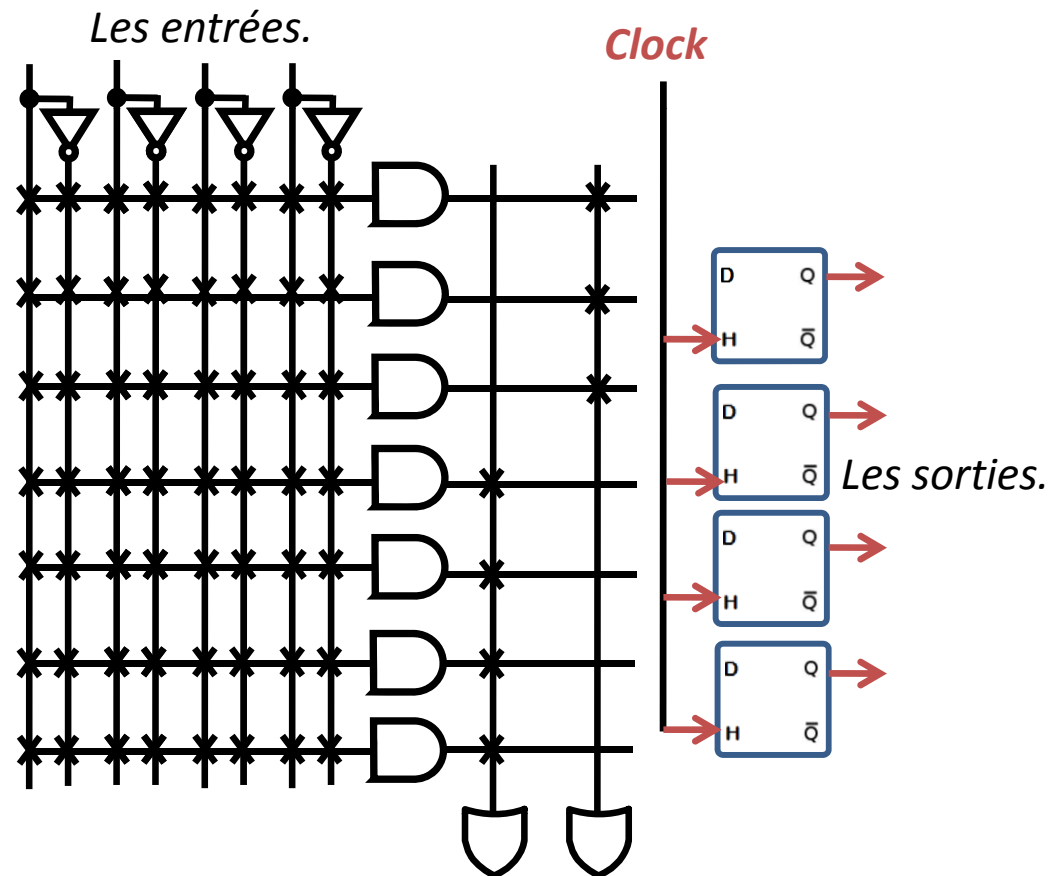
C-13: Schéma générique, succinct, d'un PAL.

Matrice de AND programmable suivie d'une matrice de OR figée.



C) Les réseaux logiques programmables.

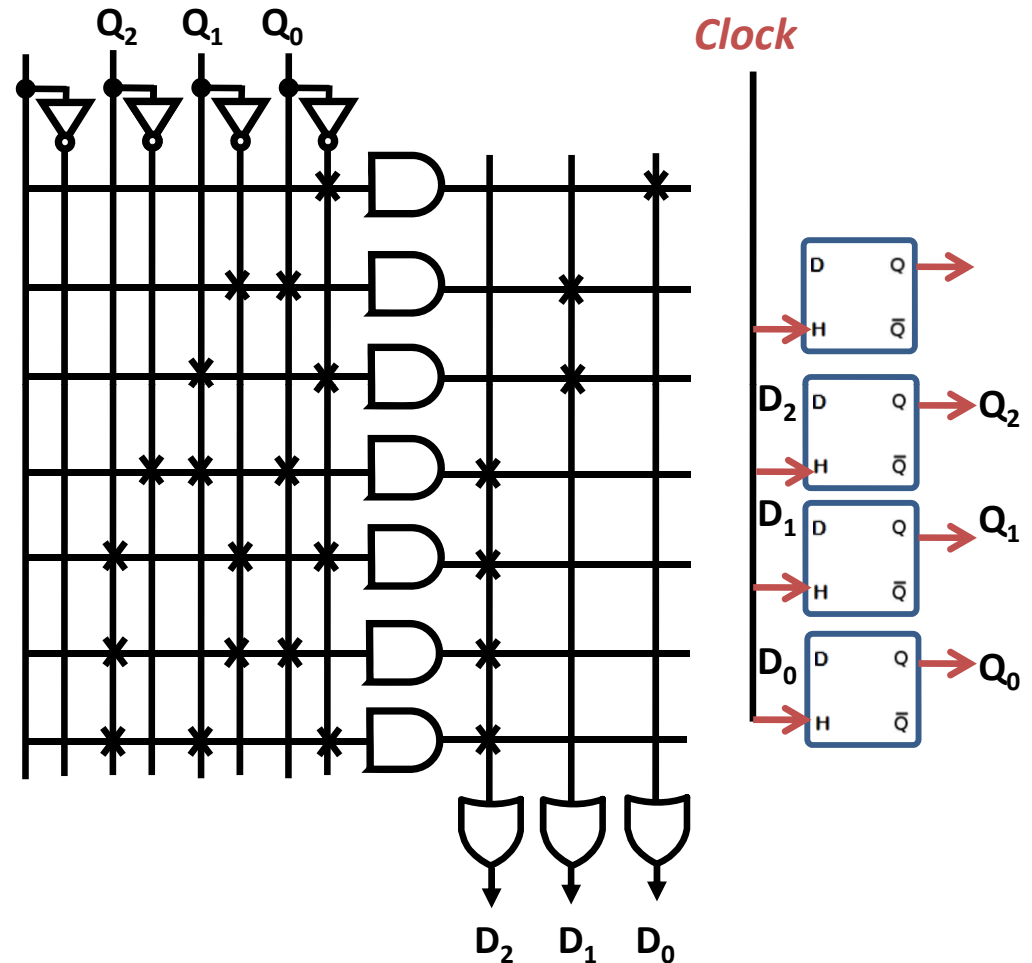
C-13: Schéma générique, succinct, d'un PAL (*séquentiel*) avec des éléments de mémorisation (des bistables D). Les connexions aux entrées D sont également programmables à partir des sorties des OR.



C) Les réseaux logiques programmables.

C-14: Exemple: Compteur synchrone Modulo 8.

Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	D_2	D_1	D_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0



$$D_0 = \overline{Q_0}$$

$$D_1 = \overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0}$$

$$D_2 = \overline{Q_2} \cdot Q_1 \cdot Q_0 + Q_2 \cdot \overline{Q_1} \cdot \overline{Q_0} + Q_2 \cdot \overline{Q_1} \cdot Q_0 + Q_2 \cdot Q_1 \cdot \overline{Q_0}$$