

Electronique Numérique et Synthèse Logique

Introduction à l'électronique numérique

Les circuits combinatoires

Les circuits séquentiels

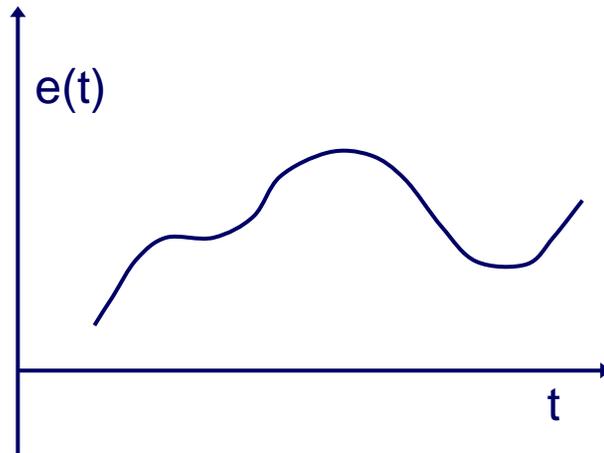
Les mémoires et circuits logiques programmables

Introduction à l'électronique numérique

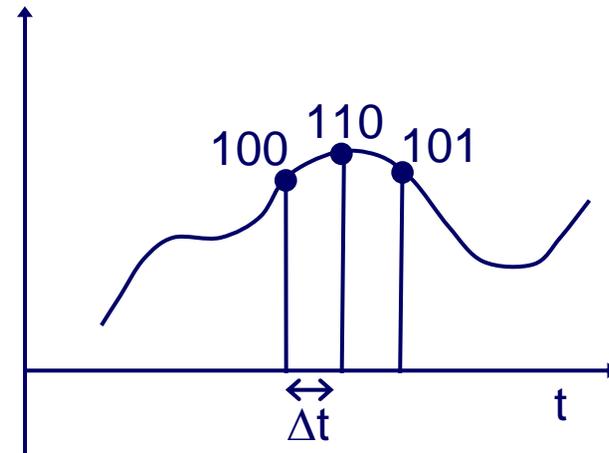
I – Signaux et systèmes numériques

I.1 – Définitions :

Un système numérique est un ensemble de dispositifs et de circuits permettant de réaliser des fonctions sur des grandeurs de natures numériques.



Signal analogique : Représentation continue au cours du temps.

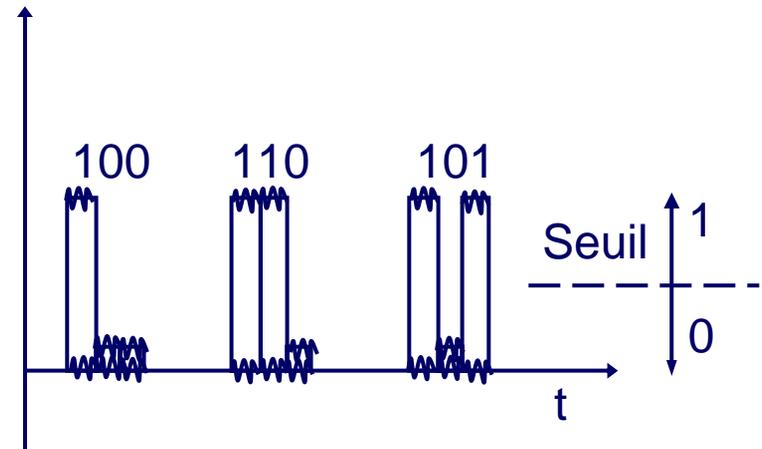
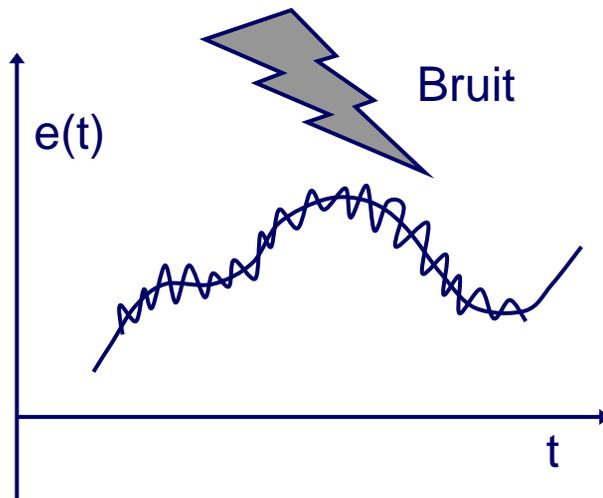


Signal numérique : Le signal est prélevé à des instants discrets et chaque amplitude est représentée par un nombre binaire (0 ou 1).

I – Signaux et systèmes numériques

Avantages :

- Circuits rapides et précis.
- Circuits programmables.
- Possibilité de garder les données en mémoire.
- Peu de sensibilités aux parasites.



Système hybride : Système dans lequel interviennent des grandeurs analogiques et numériques. Ils nécessitent des convertisseurs analogiques numériques et numériques analogiques.

II – Les systèmes de numérotation

II.1 – Le système décimal (base 10) :

Il possède 10 éléments de 0 à 9.

Chiffre de poids fort Chiffre de poids faible

 ↘ ↙

 453 = 4. 10² + 5. 10¹ + 3.10⁰

La virgule sépare les puissances positives des puissances négatives.

$$2745,214 = 2. 10^3 + 7. 10^2 + 4. 10^1 + 5. 10^0 + 2. 10^{-1} + 1. 10^{-2} + 4. 10^{-3}$$

Principe de comptage :

N chiffres ⇒ 10^N possibilités
(de 0 à 10^N-1).

0	10	20	100
1	11	21	101
2	12	22	102
3	13	23	103
4	14	24	104
5	15	25	105
6	16	26	106
7	17	27	107
8	18	28	108
9	19	29	109

II – Les systèmes de numérotation

II.2 – Le système binaire (base 2) :

Il possède 2 éléments 0 et 1 appelés bits (binary digit).

Bit de poids fort (Most Significant Bit)



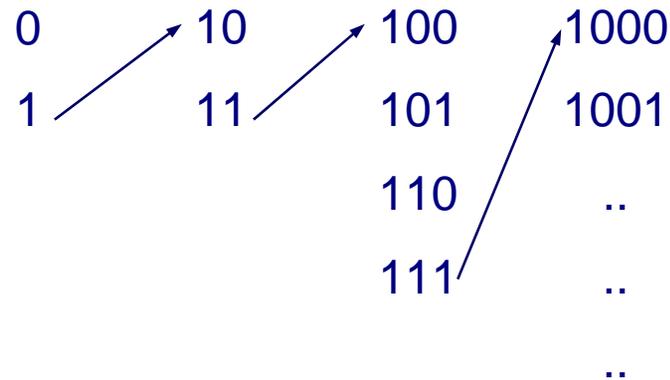
$$(1011,101)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$



Bit de poids faible (Least Significant Bit)

La virgule sépare les puissances positives des puissances négatives.

Principe de comptage :



N bits $\Rightarrow 2^N$ possibilités
(de 0 à 2^N-1).

II – Les systèmes de numérotation

II.3 – Le système hexadécimal (base 16) :

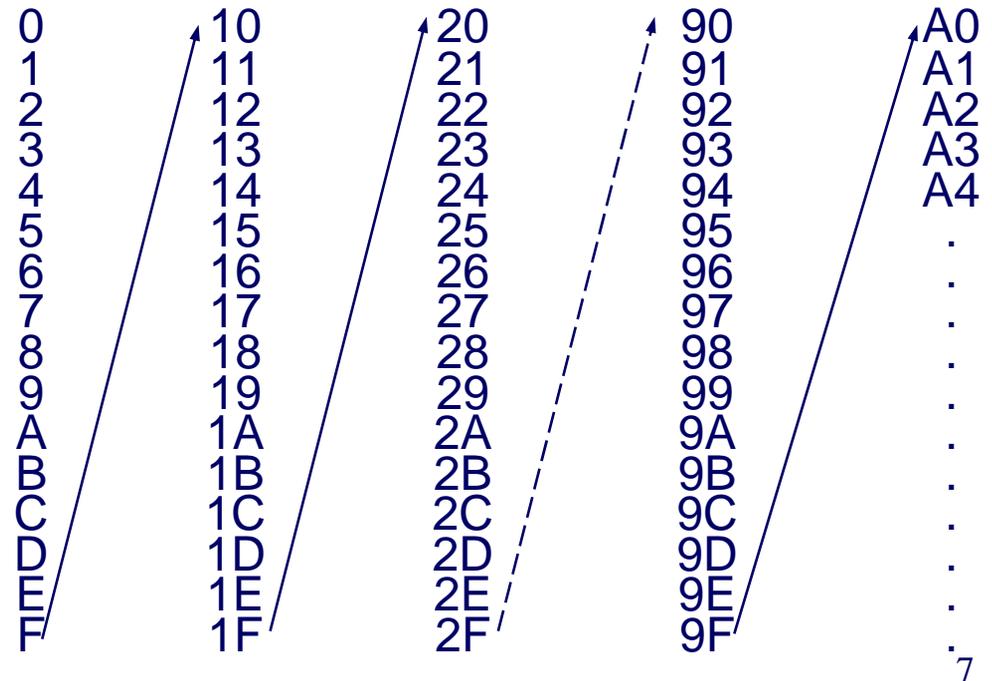
Il possède 16 éléments de 0 à F (0..9, A, B, C, D, E, F).

$$(1A,2)_{16} = 1 \cdot 16^1 + A \cdot 16^0 + 2 \cdot 16^{-1}$$

Les principes précédents restent valables.

Principe de comptage :

N chiffres $\Rightarrow 16^N$ possibilités (de 0 à 16^N-1).



II – Les systèmes de numérotation

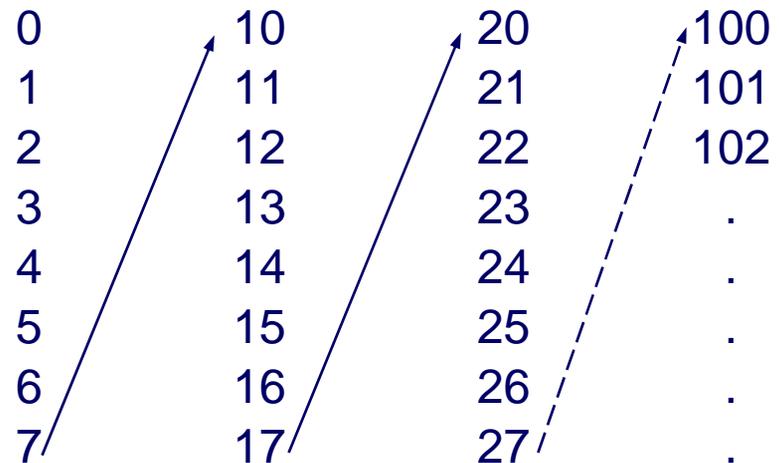
II.4 – Le système octal (base 8) :

Il possède 8 éléments de 0 à 7.

$$(372,4)_8 = 3 \cdot 8^2 + 7 \cdot 8^1 + 2 \cdot 8^0 + 4 \cdot 8^{-1}$$

Principe de comptage :

N chiffres $\Rightarrow 8^N$ possibilités
(de 0 à $8^N - 1$).



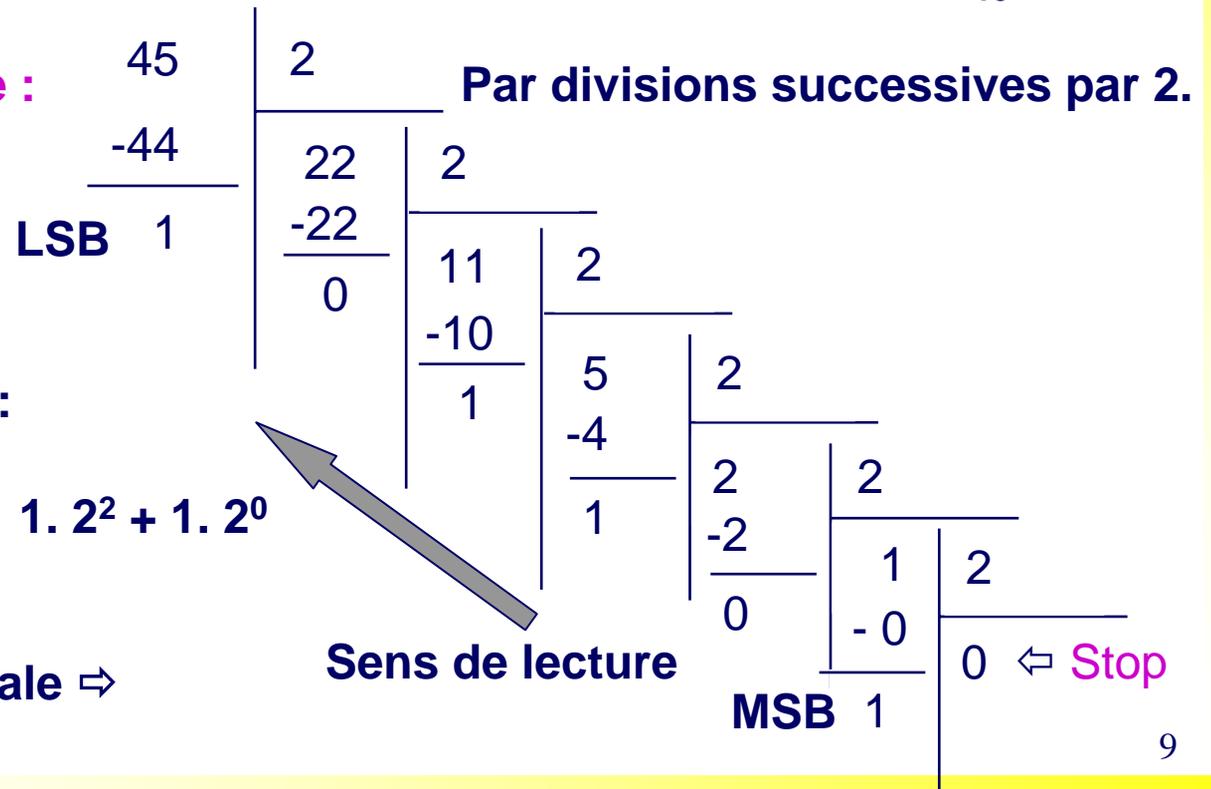
II – Les systèmes de numérotation

II.5 – Les conversions :

Binaire vers décimal : Par addition directe des poids binaires (2^x).

$$(11011,1)_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = (27,5)_{10}$$

Décimal vers binaire :



Ou par calcul direct :

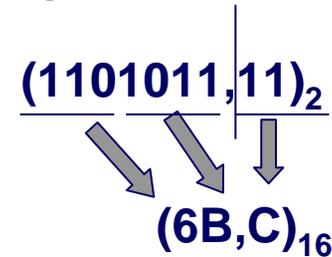
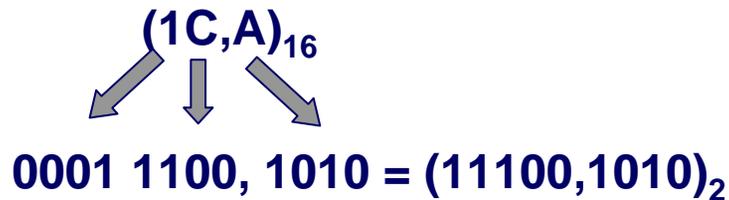
$$\begin{aligned} (45)_{10} &= 32 + 8 + 4 + 1 \\ &= 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 \\ &= (101101)_2 \end{aligned}$$

Pour la partie décimale \Rightarrow
multiplication par 2.

II – Les systèmes de numérotation

Binaire vers hexadécimal ou inversement :

$16 = 2^4 \Rightarrow$ Regroupements de 4 bits à partir de la virgule.



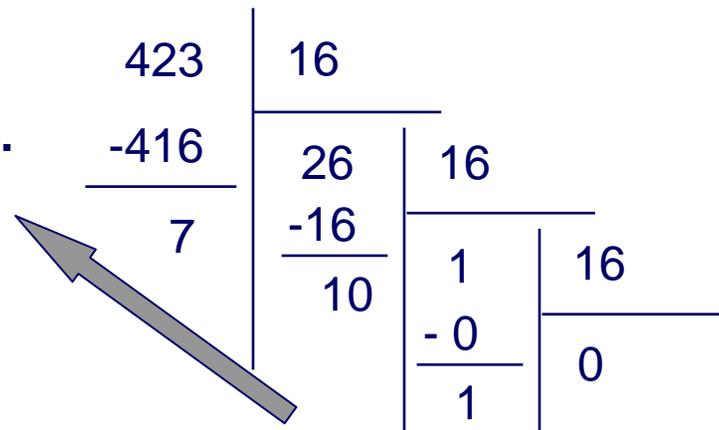
Hexadécimal vers décimal : Par addition directe des poids binaires (16^X).

$$(A27,4)_{16} = A \cdot 16^2 + 2 \cdot 16^1 + 7 \cdot 16^0 + 4 \cdot 16^{-1} = (2599,25)_{10}$$

Décimal vers hexadécimal :

Par divisions successives par 16.

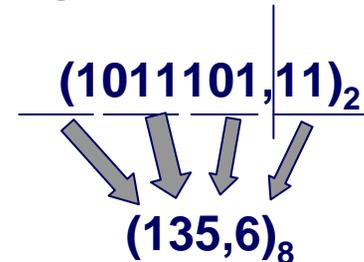
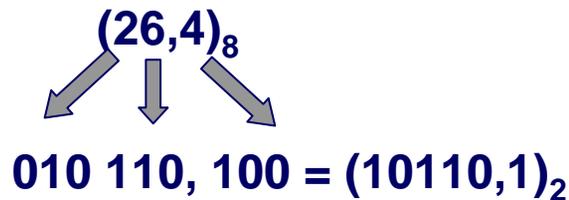
$$(423)_{10} = (1A7)_{16}$$



II – Les systèmes de numérotation

Binaire vers octal ou inversement :

$8 = 2^3 \Rightarrow$ Regroupements de 3 bits à partir de la virgule.



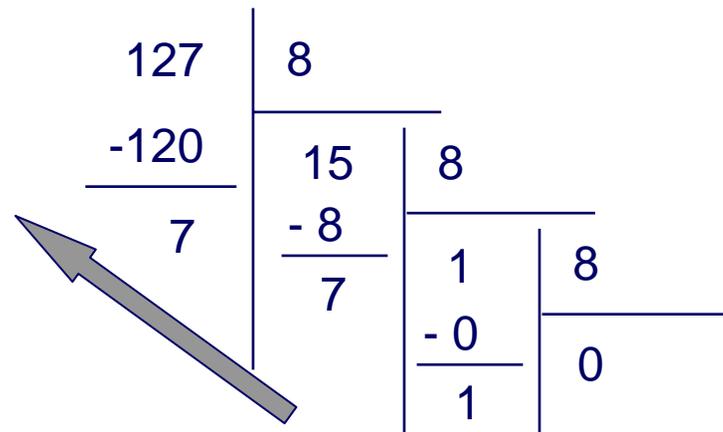
Octal vers décimal : Par addition directe des poids binaires (8^x).

$$(745,4)_8 = 7 \cdot 8^2 + 4 \cdot 8^1 + 5 \cdot 8^0 + 4 \cdot 8^{-1} = (485,5)_{10}$$

Décimal vers octal :

Par divisions successives par 8.

$$(127)_{10} = (177)_8$$



II – Les systèmes de numérotation

II.6 – Récapitulatif :

(Décimal) ₁₀	(Binaire) ₂	(Octal) ₈	(Hexadécimal) ₁₆
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

III – Les codes

Un code est une correspondance arbitraire entre un ensemble de symboles et un ensemble d'objets.

Les objets sont des lettres, des chiffres, des signes,... Les symboles sont des combinaisons de 0 et de 1.

III.1 – Les codes pondérés :

Chaque position de chiffres correspond à un poids (exemple : le code binaire naturel).

Le code D.C.B. (Décimal Codé Binaire) :

A chaque chiffre d'un nombre décimal (0 à 9), on associe son équivalent binaire (4 bits). Il correspond à la pression d'une touche.

	A	B	C	D
poids	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1


 $0100\ 0010\ 0001 \neq 110100101 = (421)_{10}$

III – Les codes

Le code AIKEN :

Il fonctionne comme le DCB pour des poids égaux à 2, 4, 2, 1.

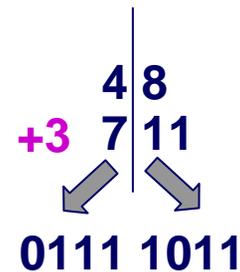


III.2 – Les codes non pondérés :

Il n'y a pas de poids binaires mais une correspondance entre chaque élément.

Le code Excédant 3 (XS3) :

A chaque chiffre d'un nombre décimal (0 à 9), on ajoute 3 et on code le résultat en binaire naturel (4 bits). Il est utilisé pour les calculs (soustraction).



	A	B	C	D
poids	2	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

	A	B	C	D
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0 ¹⁴

III – Les codes

Le code Gray ou binaire réfléchi :

Deux termes successifs ne diffèrent que par un seul bit.

0000
0001 1^{er} axe de symétrie
0011
0010 2^{eme} axe de symétrie
0110
0111
0101
0100 3^{eme} axe de symétrie
1100
1101
....

Aucun état temporaire indésirable !

7 ⇒ 8 : 0111 ⇒ 1000 (binaire)
0100 ⇒ 1100 (Gray)

	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

III – Les codes

Le code ASCII (American Standard Code for Information Interchange) :

Initialement, ce code sur 7 bits comprenait 87 caractères minimum :

- 26 minuscules.**
- 26 majuscules.**
- 10 chiffres.**
- 25 caractères spéciaux (+,£, \$,...).**

Il est à présent codé sur 8 bits et appelé ASCII étendu (256 caractères).

III.3 – Les codes détecteurs d'erreurs :

Ils sont destinés à donner des informations permettant au récepteur de vérifier le mot reçu.

Le code p parmi n :

A chaque chiffre décimal on fait correspondre n bits dont p sont à 1 et n-p à 0 (nécessité d'un poids nul).

III – Les codes

Le code 2 parmi 5 :

Deux séries de poids possibles.

Codage particulier →

	V	W	X	Y	Z	V	W	X	Y	Z
poids	7	4	2	1	0	8	4	2	1	0
0	1*	1	0	0	0	1*	1	0	0	0
1	0	0	0	1	1	0	0	0	1	1
2	0	0	1	0	1	0	0	1	0	1
3	0	0	1	1	0	0	0	1	1	0
4	0	1	0	0	1	0	1	0	0	1
5	0	1	0	1	0	0	1	0	1	0
6	0	1	1	0	0	0	1	1	0	0
7	1	0	0	0	1	1*	0	1	0	0
8	1	0	0	1	0	1	0	0	0	1
9	1	0	1	0	0	1	0	0	1	0

Codage particulier →

Ces codes permettent de détecter 0 ou 1 erreur mais ne permettent pas de les corriger.

III – Les codes

Le code biquinaire :

	1 ^{er} groupe (0, 5)			2 ^{eme} groupe (0, 1, 2, 3, 4)			
	A	B	C	D	E	F	G
	0	5	0	1	2	3	4
0	1	0	1	0	0	0	0
1	1	0	0	1	0	0	0
2	1	0	0	0	1	0	0
3	1	0	0	0	0	1	0
4	1	0	0	0	0	0	1
5	0	1	1	0	0	0	0
6	0	1	0	1	0	0	0
7	0	1	0	0	1	0	0
8	0	1	0	0	0	1	0
9	0	1	0	0	0	0	1

Il contient deux groupes contenant un seul 1 d'où la possibilité de détecter 2 erreurs (si elles ne sont pas dans le même groupe).

III – Les codes

Le code avec bit de parité :

Un bit de parité est un bit supplémentaire contrôlant le nombre de 1 dans l'information.

Parité paire : Le bit de parité est fixé de façon à ce que le nombre de 1 contenus dans l'information et le bit de parité est pair.

Parité impaire : Le nombre total de 1 (information et bit de parité) est impair.

Information globalement impaire. 

A	B	C	D	P
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1

La détection du bit erroné est possible si l'on code les informations émises par lignes et par colonnes.

III – Les codes

Table ASCII

Code	Caractère	Code	Caractère	Code	Caractère	Code	Caractère	
21	0100001	!	34	0110100	4	48	1001000	H
22	0100010	"	35	0110101	5	49	1001001	I
23	0100011	#	36	0110110	6	4A	1001010	J
24	0100100	\$	37	0110111	7	4B	1001011	K
25	0100101	%	38	0111000	8	4C	1001100	L
26	0100110	&	39	0111001	9	4D	1001101	M
27	0100111	'	3A	0111010	:	4E	1001110	N
28	0101000	(3B	0111011	;	4F	1001111	O
29	0101001)	3C	0111100	<	50	1010000	P
2A	0101010	*	3D	0111101	=	51	1010001	Q
2B	0101011	+	3E	0111110	>	52	1010010	R
2C	0101100	,	3F	0111111	?	53	1010011	S
2D	0101101	-	40	1000000	@	54	1010100	T
2E	0101110	.	41	1000001	A	55	1010101	U
2F	0101111	/	42	1000010	B	56	1010110	V
30	0110000	0	43	1000011	C	57	1010111	W
31	0110001	1	44	1000100	D	58	1011000	X
32	0110010	2	45	1000101	E	59	1011001	Y
33	0110011	3	46	1000110	F	5A	1011010	Z
			47	1000111	G	5B	1011011	[
5C	1011100	\	6E	1101110	n	6F	1101111	o
5D	1011101]	6F	1101111	o			
5E	1011110	^	70	1110000	p			
5F	1011111	_	71	1110001	q			
60	1100000	`	72	1110010	r			
61	1100001	a	73	1110011	s			
62	1100010	b	74	1110100	t			
63	1100011	c	75	1110101	u			
64	1100100	d	76	1110110	v			
65	1100101	e	77	1110111	w			
66	1100110	f	78	1111000	x			
67	1100111	g	79	1111001	y			
68	1101000	h	7A	1111010	z			
69	1101001	i	7B	1111011	{			
6A	1101010	j	7C	1111100				
6B	1101011	k	7D	1111101	}			
6C	1101100	l	7E	1111110	~			
6D	1101101	m	7F	1111111	DEL			

IV – Arithmétique binaire

IV.1 – L'addition binaire :

Seuls 4 cas sont possibles :

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \text{ et une retenue de } 1$$

Éventuellement : $1 + 1 + 1 = 1$ et une retenue de 1 !

$$\begin{array}{r} \\ \\ \hline \end{array}$$

$$\begin{array}{r} \\ \\ \hline \end{array}$$

Ne pas confondre l'addition binaire et l'opérateur logique OU !

IV – Arithmétique binaire

IV.2 – Nombres signés :

En arithmétique signé, un bit de signe **S** est nécessaire tel que :

- Si le nombre est positif, **S = 0**.
- Si le nombre est négatif, **S = 1**.

Notation en grandeur exacte :

	S								
+ 57	⇒	0	1	1	1	0	0	1	binaire naturel.
- 57	⇒	1	1	1	1	0	0	1	binaire naturel.

Notation en complément à 1 (restreint) :

	S								
+ 57	⇒	0	1	1	1	0	0	1	binaire naturel.
- 57	⇒	1	0	0	0	1	1	0	cpl1.

Le complément à 1 correspond à complémenter chaque bit du nombre négatif.

IV – Arithmétique binaire

Notation en complément à 2 (vrai) :

		S							
+ 57	⇒	0	1	1	1	0	0	1	binaire naturel.
- 57	⇒	1	0	0	0	1	1	1	cpl2.

cpl2 = cpl1 + 1.

Inversement, quand un nombre négatif est codé en complément à 2 (ou 1), il faut prendre son complément à 2 (ou 1) pour obtenir sa valeur.

Intervalle de variation : 7 bits de codage + 1 bit de signe.

De	0 1111111	à	1 0000000
De	+127	à	- 128

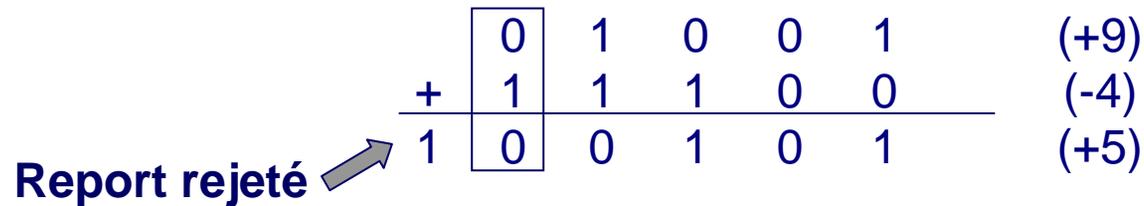
N bits de codage + 1 bit de signe : de $2^N - 1$ à -2^N .

IV – Arithmétique binaire

IV.4 – Soustraction et complément à 2 :

Soustraction = Addition en cpl2 ((+9) – (+4) = (+9) + (-4))

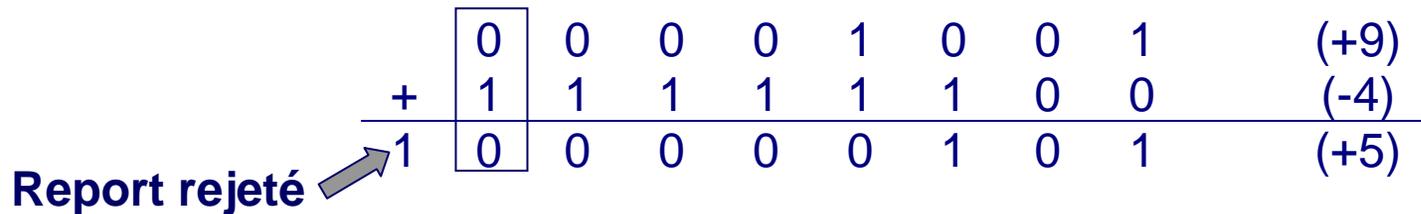
		0	1	0	0	1	(+9)	
	+	1	1	1	0	0	(-4)	
Report rejeté		1	0	0	1	0	(+5)	



IV.5 – Problèmes de dépassement :

Exemple : 7 bits de codage et 1 bit de signe.

		0	0	0	0	1	0	0	1	(+9)		
	+	1	1	1	1	1	1	0	0	(-4)		
Report rejeté		1	0	0	0	0	0	1	0	(+5)		



IV – Arithmétique binaire

$$\begin{array}{rcccccccc} & \boxed{0} & 0 & 1 & 1 & 0 & 0 & 0 & 1 & (+49) \\ + & \boxed{0} & 1 & 0 & 1 & 1 & 0 & 0 & 0 & (+88) \\ \hline 0 & \boxed{1} & 0 & 0 & 0 & 1 & 0 & 0 & 1 & (+137) \end{array}$$

← - 119 →

$$\begin{array}{rcccccccc} & \boxed{1} & 1 & 1 & 0 & 0 & 0 & 0 & 0 & (-32) \\ + & \boxed{1} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & (-127) \\ \hline 1 & \boxed{0} & 1 & 1 & 0 & 0 & 0 & 0 & 1 & (-159) \end{array}$$

← + 97 →

 **Report rejeté**

Le résultat doit pouvoir être codé sur le nombre de bits utilisés !

IV – Arithmétique binaire

IV.6 – La multiplication binaire :

$$\begin{array}{r}
 1 \ 0 \ 0 \ 1 (9) \\
 X 1 \ 0 \ 1 \ 1 (11) \\
 \hline
 1 \ 0 \ 0 \ 1 \\
 1 \ 0 \ 0 \ 1 \ . \\
 0 \ 0 \ 0 \ 0 \ . \\
 1 \ 0 \ 0 \ 1 \ . \\
 \hline
 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 (99)
 \end{array}$$

Multiplier par 2 = décaler à gauche.

La multiplication et le complément à 2 :

$$\begin{array}{r}
 X \begin{array}{|c|} \hline 0 \\ \hline \end{array} 0 \ 1 \ 1 (-3) \\
 \begin{array}{|c|} \hline 0 \\ \hline \end{array} 0 \ 1 \ 0 (-2) \\
 \hline
 \begin{array}{|c|} \hline 0 \\ \hline \end{array} 0 \ 0 \ 0 \\
 0 \begin{array}{|c|} \hline 0 \\ \hline \end{array} 1 \ 1 \ . \\
 \hline
 0 \begin{array}{|c|} \hline 0 \\ \hline \end{array} 1 \ 1 \ 0 (+6)
 \end{array}$$

$$\begin{array}{r}
 X \begin{array}{|c|} \hline 0 \\ \hline \end{array} 0 \ 1 \ 1 (-3) \\
 \begin{array}{|c|} \hline 0 \\ \hline \end{array} 0 \ 1 \ 0 (+2) \\
 \hline
 \begin{array}{|c|} \hline 0 \\ \hline \end{array} 0 \ 0 \ 0 \\
 0 \begin{array}{|c|} \hline 0 \\ \hline \end{array} 1 \ 1 \ . \\
 \hline
 0 \begin{array}{|c|} \hline 0 \\ \hline \end{array} 1 \ 1 \ 0 (-6)
 \end{array}$$

Nombre positif ⇒ Garder le nombre.
Nombre négatif ⇒ Prendre le cpl2.

Signe à ⇒ 1 0 1 0 en cpl2
Rectifier.

IV – Arithmétique binaire

IV.7 – La division binaire :

$$\begin{array}{r|l}
 1001 & 11 \\
 -011 & \hline
 11 & 0011 \\
 0 & \\
 \hline
 \end{array}
 \qquad
 \begin{array}{r|l}
 1010 & 100 \\
 -100 & \hline
 100 & 0010,1 \\
 0 & \\
 \hline
 \end{array}$$

Calcul identique à la base 10.

IV.8 – L'addition en DCB :

$$\begin{array}{rcccc}
 & 0 & 1 & 0 & 1 & \text{D.C.B de 5} \\
 + & 0 & 1 & 0 & 0 & \text{D.C.B de 4} \\
 \hline
 & 1 & 0 & 0 & 1 & \text{D.C.B de 9}
 \end{array}$$

Calcul identique à l'addition binaire. 

$$\begin{array}{rcccc}
 & 0 & 1 & 1 & 0 & \text{D.C.B de 6} \\
 + & 0 & 1 & 1 & 1 & \text{D.C.B de 7} \\
 \hline
 & 1 & 1 & 0 & 1 & \text{Invalide !}
 \end{array}$$


$$\begin{array}{rcccc}
 & 0 & 1 & 1 & 0 & \text{D.C.B de 6} \\
 + & 0 & 1 & 1 & 1 & \text{D.C.B de 7} \\
 \hline
 & 1 & 1 & 0 & 1 & \text{invalide} \\
 + & 0 & 1 & 1 & 0 & \text{correction} \\
 \hline
 & 1 & 0 & 0 & 1 & \text{D.C.B de 13}
 \end{array}$$

Code inexistant \Rightarrow Correction de +6
(6 valeurs inexistantes de 1010 à 1111).

IV – Arithmétique binaire

Position de la correction :

$$\begin{array}{r}
 \begin{array}{cccccccccccc}
 & 0 & 0 & 1 & 0 & & 0 & 1 & 1 & 1 & & 0 & 1 & 0 & 1 & 275 \\
 + & 0 & 1 & 1 & 0 & & 0 & 1 & 0 & 0 & & 0 & 0 & 0 & 1 & 641 \\
 \hline
 & 1 & 0 & 0 & 0 & & 1 & 0 & 1 & 1 & & 0 & 1 & 1 & 0 & \\
 + & & & & & & 0 & 1 & 1 & 0 & & & & & & \\
 \hline
 & 1 & 0 & 0 & 1 & & 0 & 0 & 0 & 1 & & 0 & 1 & 1 & 0 & 916
 \end{array}
 \end{array}$$

IV.9 – Les opérations en hexadécimal :

L'addition :

$$\begin{array}{r}
 \begin{array}{r}
 \begin{array}{r}
 5 \ 8 \quad (88) \\
 + 2 \ 4 \quad (36) \\
 \hline
 7 \ C \quad (124)
 \end{array}
 \end{array}
 \qquad
 \begin{array}{r}
 \begin{array}{r}
 5 \ 8 \quad (88) \\
 + 4 \ B \quad (75) \\
 \hline
 A \ 3 \quad (163)
 \end{array}
 \end{array}
 \end{array}$$

La soustraction (par complément à 2) :

$$\begin{array}{r}
 \begin{array}{r}
 C \ 1 \quad (193) \\
 - A \ 8 \quad (168) \\
 \hline
 1 \ 9 \quad (25)
 \end{array}
 \Rightarrow
 \begin{array}{l}
 A8 \quad : 10101000 \\
 \text{cpl1}(A8) : 01010111 \\
 \text{cpl2}(A8) : 01011000 = (58)
 \end{array}
 \Rightarrow
 \begin{array}{r}
 \begin{array}{r}
 C \ 1 \\
 + 5 \ 8 \\
 \hline
 1 \ 1 \ 9
 \end{array}
 \end{array}
 \end{array}$$


Report rejeté

V – Variables et fonctions binaires

V.1 – Algèbre de Boole :

C'est une algèbre proposée en 1854 par Gorges Boole ne faisant intervenir que des variables pouvant prendre que deux états 0 ou 1 (variable booléenne).

Une variable booléenne représente un niveau de tension sur un fil et peut être assimilée à un interrupteur :

- Si $V \in [0; 0,8V] \Rightarrow$ Niveau logique bas ou 0 ou faux (interrupteur ouvert).
- Si $V \in [2; 5V] \Rightarrow$ Niveau logique haut ou 1 ou vrai (interrupteur fermé).

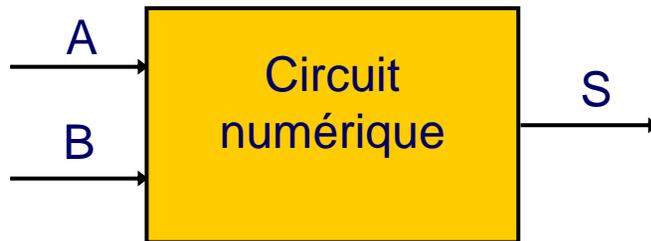
Si, à l'état logique 1, on associe le niveau de tension le plus élevé, le circuit travaille en logique positive.

Si, à l'état logique 1, on associe le niveau de tension le plus bas, le circuit travaille en logique négative.

V – Variables et fonctions binaires

V.2 – La table de vérité :

C'est un tableau représentant tous les états possibles des sorties en fonctions des entrées (N entrées $\Rightarrow 2^N$ lignes).



A	B	S
0	0	1
0	1	0
1	0	0
1	1	X

V.3 – Les opérateurs de bases :

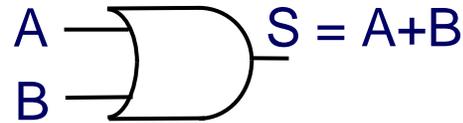
L'algèbre de Boole possède 3 opérateurs logiques permettant de définir tous les autres : OU (OR), ET (AND), NON (NOT).

V – Variables et fonctions binaires

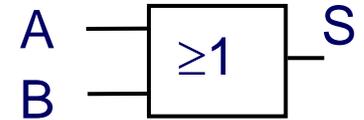
L'opérateur OU (OR) noté + :

A	B	$S = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Logigramme :



Internationale

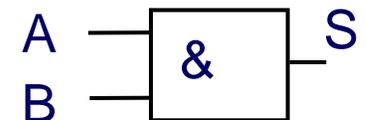
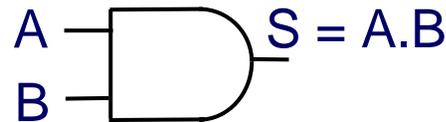


Française

Généralisation : $S = 1$ si au moins une des entrées est égale à 1.

L'opérateur ET (AND) noté . :

A	B	$S = A . B$
0	0	0
0	1	0
1	0	0
1	1	1

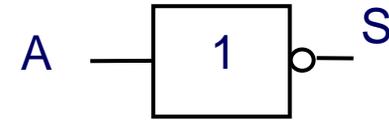
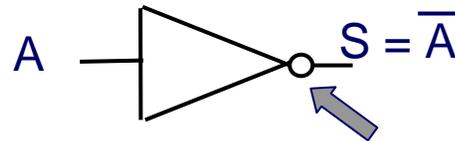


Généralisation : $S = 1$ si toutes les entrées sont à 1.

V – Variables et fonctions binaires

L'opérateur NON (NOT) noté $\bar{\quad}$:

	$S = \bar{A}$
0	1
1	0

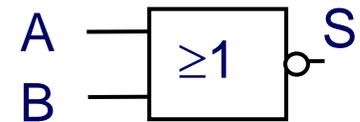
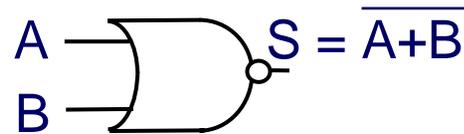


Sigle d'inversion

V.4 – Trois autres opérateurs logiques :

L'opérateur NON OU (NOR) (OU suivi d'une porte NON) :

A	B	$S = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

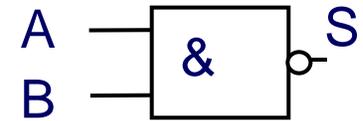
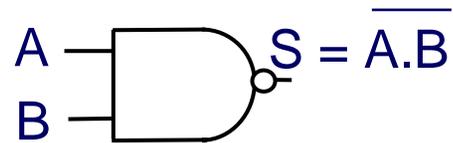


Généralisation : $S = 0$ si au moins une des entrées est égale à 1.

V – Variables et fonctions binaires

L'opérateur NON ET (NAND) (ET suivi d'une porte NON) :

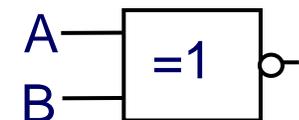
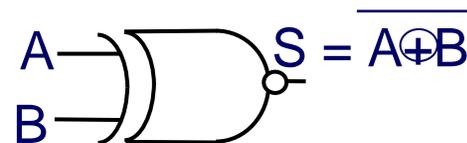
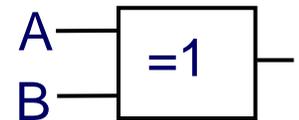
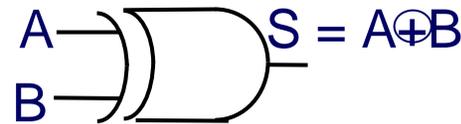
A	B	$S = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0



Généralisation : $S = 0$ si toutes les entrées sont à 1.

L'opérateur OU exclusif (XOR) :

A	B	$S = A \oplus B$ $\overline{A}B + A\overline{B}$	$S = \overline{A \oplus B}$ $\overline{\overline{A}B + A\overline{B}}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

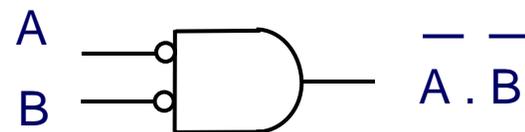
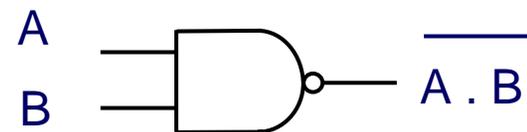
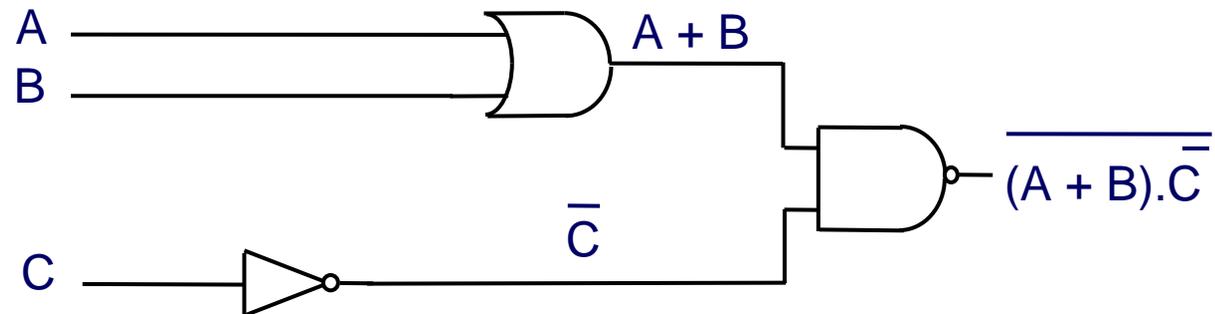
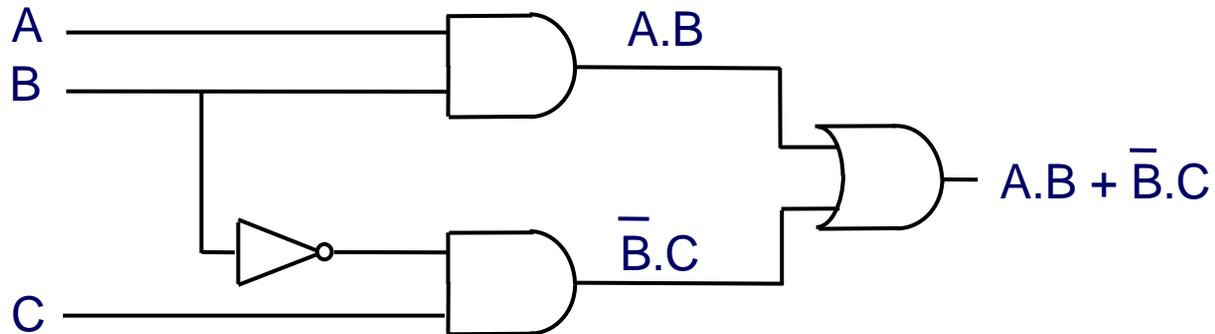


$A \oplus B = 1$ si :

- $A \neq B$.
- (A,B) contient un nombre impair de 1.

V – Variables et fonctions binaires

V.5 – Exemples :



V – Variables et fonctions binaires

V.6 – Théorèmes de l'algèbre de Boole :

Lois liées aux opérateurs de base :

$$A.0 = 0$$

$$A.1 = A$$

$$A.A = A$$

$$A.\overline{A} = 0$$

$$A+1 = 1$$

$$A+0 = A \quad \text{Elément neutre}$$

$$A+A = A \quad \text{Idempotence}$$

$$A+\overline{A} = 1 \quad \text{Inversion}$$

$$\overline{0} = 1$$

$$\overline{1} = 0$$

$$\overline{\overline{A}} = A$$

Lois de commutativité :

$$A+B = B+A$$

$$A.B = B.A$$

Lois d'associativité :

$$(A+B)+C = A+(B+C) = A+B+C$$

$$(A.B).C = A.(B.C) = A.B.C$$

Lois de distributivité :

$$A.(B+C) = A.B+A.C$$

$$A+(B.C) = (A+B).(A+C)$$

Lois d'absorption :

$$A+A.B = A.(1+B) = A$$

$$A.(A+B) = A+A.B = A$$

$$A+\overline{A}.B = (A+\overline{A}).(A+B) = A+B$$

Théorème du consensus : $A.C + \overline{B}.C + A.B = A.C + \overline{B}.C$

V – Variables et fonctions binaires

V.7 – Théorèmes de De Morgan :

Le complément d'une somme logique entre plusieurs variables est égal au produit logique des variables complémentées :

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

Le complément d'un produit logique entre plusieurs variables est égal à la somme logique des variables complémentées :

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

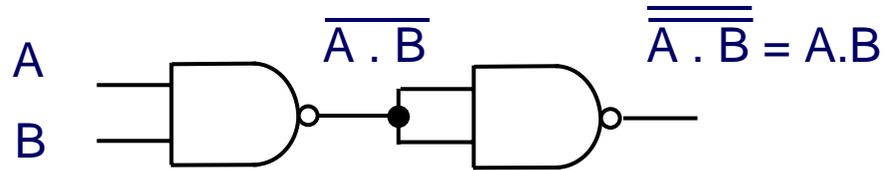
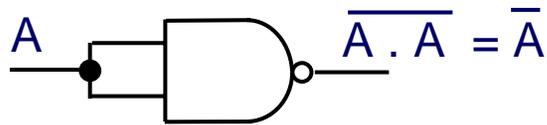
Ces relations s'appliquent à des variables binaires ou des expressions logiques.

Il devient toujours possible de transformer un opérateur ET en OU logique et inversement :

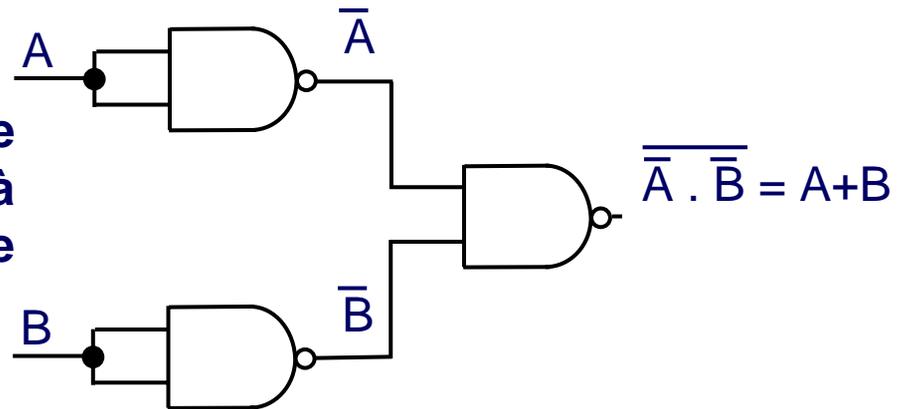
$$\begin{aligned} A \cdot B &= \overline{\overline{A \cdot B}} = \overline{\overline{A} + \overline{B}} \\ A + B &= \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}} \end{aligned}$$

V – Variables et fonctions binaires

Universalité des portes NON ET et NON OU :



Toute fonction logique peut être représentée exclusivement à l'aide de portes NON ET ou de portes NON OU.



V.8 – Simplification des fonctions binaires :

A l'aide des lois de l'algèbre de Boole

$$\begin{aligned} Z &= ABC + A\bar{B}\bar{C} + \bar{A}BC \\ &= ABC + A\bar{B}\bar{C} + \bar{A}BC + ABC \\ &= AB \quad \quad \quad + AC \\ &= A.(B+C) \end{aligned}$$

V – Variables et fonctions binaires

Par la table de vérité :

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

Minterm (produit logique de toutes les entrées directes ou complémentées).

$\bar{A} \bar{B} \bar{C}$

$\bar{A} B \bar{C}$

$\bar{A} B C$

$A \bar{B} \bar{C}$

$A B C$

Lecture selon les 1 :

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

Toute fonction logique peut être représentée par la somme logique de tous les minterms pour lesquels la sortie est à 1.

Lecture selon les 0 : $\bar{S} = \bar{A} B C + A \bar{B} C + A B \bar{C}$

$$S = (A + B + \bar{C}) \cdot (\bar{A} + B + \bar{C}) \cdot (\bar{A} + \bar{B} + C)$$

Maxterm.

Toute fonction logique peut être représentée par le produit logique de tous les maxterms (somme logique des entrées) pour lesquels la sortie est à 0.

V – Variables et fonctions binaires

Par la table de Karnaugh :

Table de vérité

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

Table de Karnaugh



Représentation identique à la table de vérité mais selon le code Gray (adjacence des cases).

A \ BC	00	01	11	10
0	1	0	1	1
1	1	0	1	0

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

$$\bar{A}BC + \bar{A}B\bar{C} = \bar{A}B$$

$$\bar{A}BC + ABC = BC$$

Forme simplifiée : $S = \bar{B}\bar{C} + BC + \bar{A}B$ Ou : $S = \bar{B}\bar{C} + BC + \bar{A}\bar{C}$

V – Variables et fonctions binaires

Règles :

- Lire tous les 1.
- Les regrouper par paquets de 2, 4, 8, 16, ...
- Un même 1 peut servir plusieurs fois.
- Lire les regroupements (ne pas tenir compte des variables qui changent d'état).
- Obtenir $S = \dots$

Ou :

- Lire tous les 0
- Obtenir $\overline{S} = \dots$

V – Variables et fonctions binaires

Formes pratiques des tables de Karnaugh :

A \ B	0	1
0		
1		

2 variables

A B \ CD	00	01	11	10
00				
01				
11				
10				

4 variables

A \ BC	00	01	11	10
0				
1				

3 variables

Cases adjacentes !

A B \ CDE	000	001	011	010	110	111	101	100
00								
01								
11								
10								

5 variables

V – Variables et fonctions binaires

Cas indéterminés :

Table de vérité

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

Table de Karnaugh



X : 0 ou 1 au choix (afin d'améliorer les regroupements).

A \ BC	00	01	11	10
0	1	X	1	1
1	1	0	1	X

\bar{C}

B

Forme simplifiée : $S = B + \bar{C}$

Les circuits combinatoires

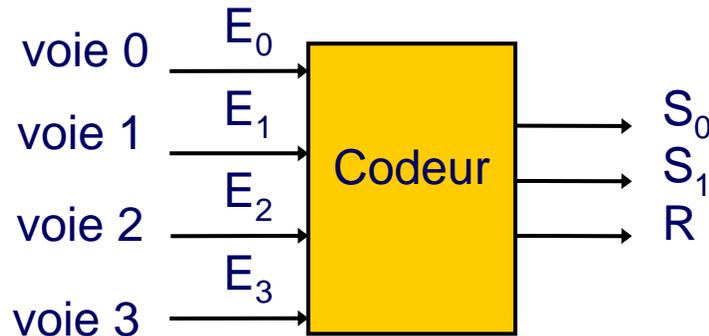
Un circuit est dit combinatoire quand ses sorties ne dépendent que de ses entrées et non des états antérieurs des sorties.

A chaque combinaison des entrées correspond toujours une seule et même combinaison des sorties.

I – Les circuits de transcodage

I.1 – Les codeurs (ou encodeurs) :

Exemple :



E_3	E_2	E_1	E_0	S_1	S_0	R
0	0	0	1	0	0	0
0	0	1	0	0	1	0
0	1	0	0	1	0	0
1	0	0	0	1	1	0
0	0	0	0	0	0	1

C'est un circuit à 2^N entrées binaires et N sorties.

Il code en binaire le numéro de l'entrée activée (codage d'une touche enfoncée).

Voie0 activée $\Rightarrow S_1 S_0 = 00$

Voie1 activée $\Rightarrow S_1 S_0 = 01$

...

$R = 1$ si aucune touche n'est activée.

**Voie activée : 1 et toutes les autres à 0
ou 0 et toutes les autres à 1.**

I – Les circuits de transcodage

$E_3E_2 \setminus E_1E_0$	00	01	11	10
00	0	0	X	0
01	1	X	X	X
11	X	X	X	X
10	1	X	X	X

$$S_1 = E_3 + E_2$$

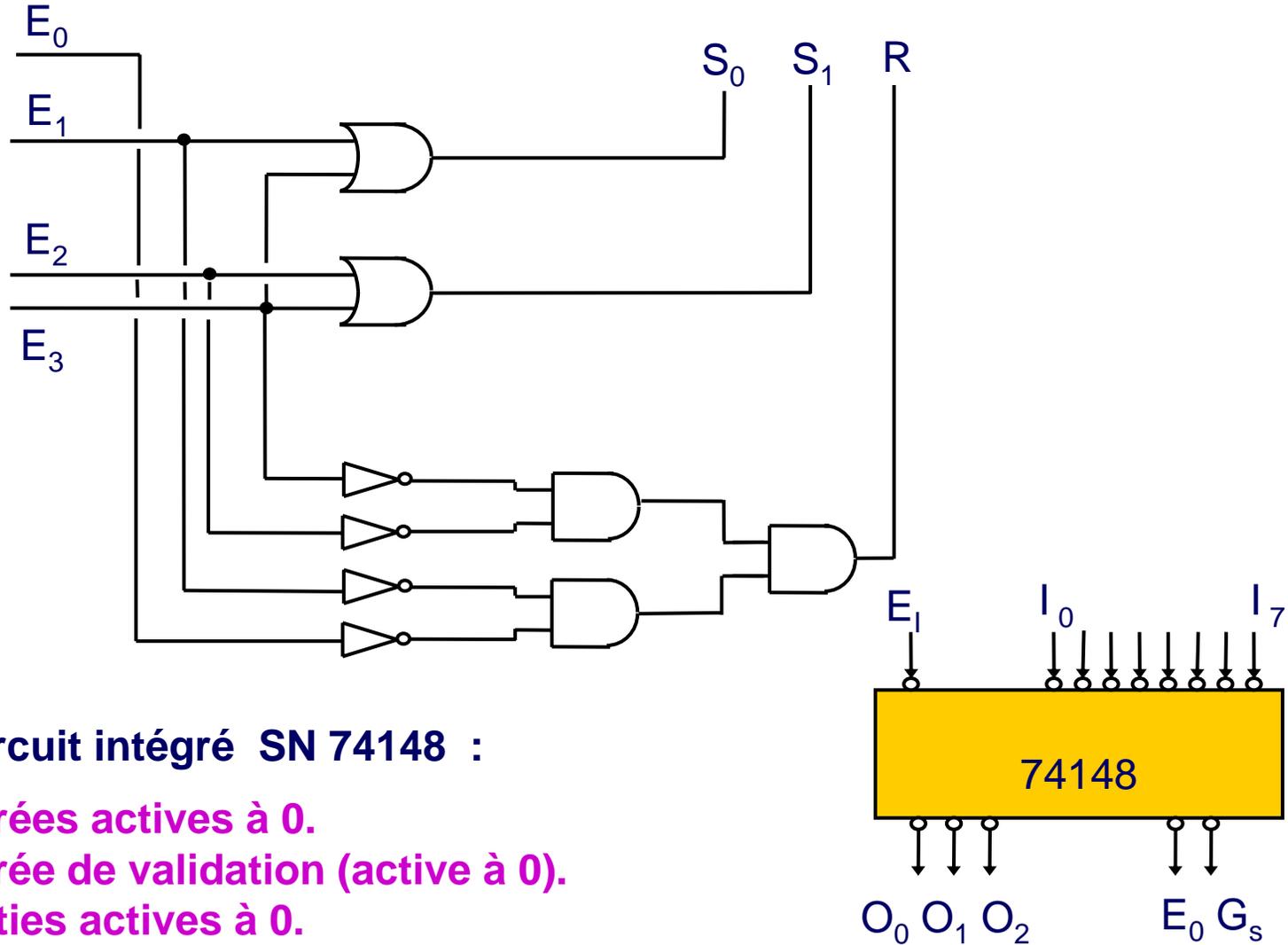
$E_3E_2 \setminus E_1E_0$	00	01	11	10
00	0	0	X	1
01	0	X	X	X
11	X	X	X	X
10	1	X	X	X

$$S_0 = E_3 + E_1$$

$E_3E_2 \setminus E_1E_0$	00	01	11	10
00	1	0	X	0
01	0	X	X	X
11	X	X	X	X
10	0	X	X	X

$$R = \overline{E_3} \overline{E_2} \overline{E_1} \overline{E_0}$$

I – Les circuits de transcodage



Le circuit intégré SN 74148 :

8 entrées actives à 0.

1 entrée de validation (active à 0).

3 sorties actives à 0.

2 sorties de contrôle (actives à 0).

I – Les circuits de transcodage

E_i	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	O_2	O_1	O_0	G_S	E_0
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	X	X	X	X	X	X	X	0	0	0	0	0	1
0	X	X	X	X	X	X	0	1	0	0	1	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	0	1	1	1	0	1	1	0	1
0	X	X	X	0	1	1	1	1	1	0	0	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

$E_i = 0$: Mode codeur.

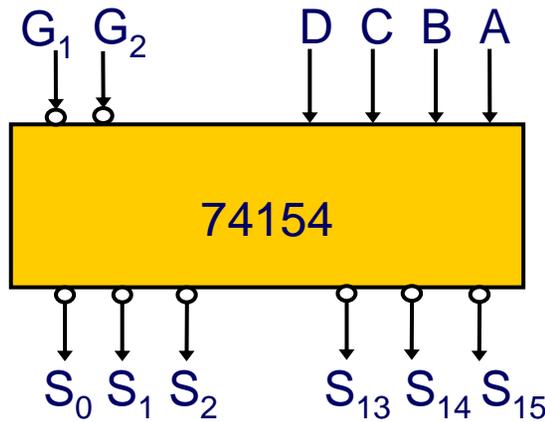
$G_S = 0$ et $E_0 = 1$: Activation (pour une mise en cascade) et codage d'une touche (en complément à 1).

Ce circuit est appelé encodeur de priorité : Seule la touche de numéro le plus élevé est codée.

I – Les circuits de transcodage

Le circuit intégré SN 74154 :

Décodeur 1 parmi 16 :



$G_1 = G_2 = 0$: Mode
décodeur.

Activation des sorties
à l'état bas.

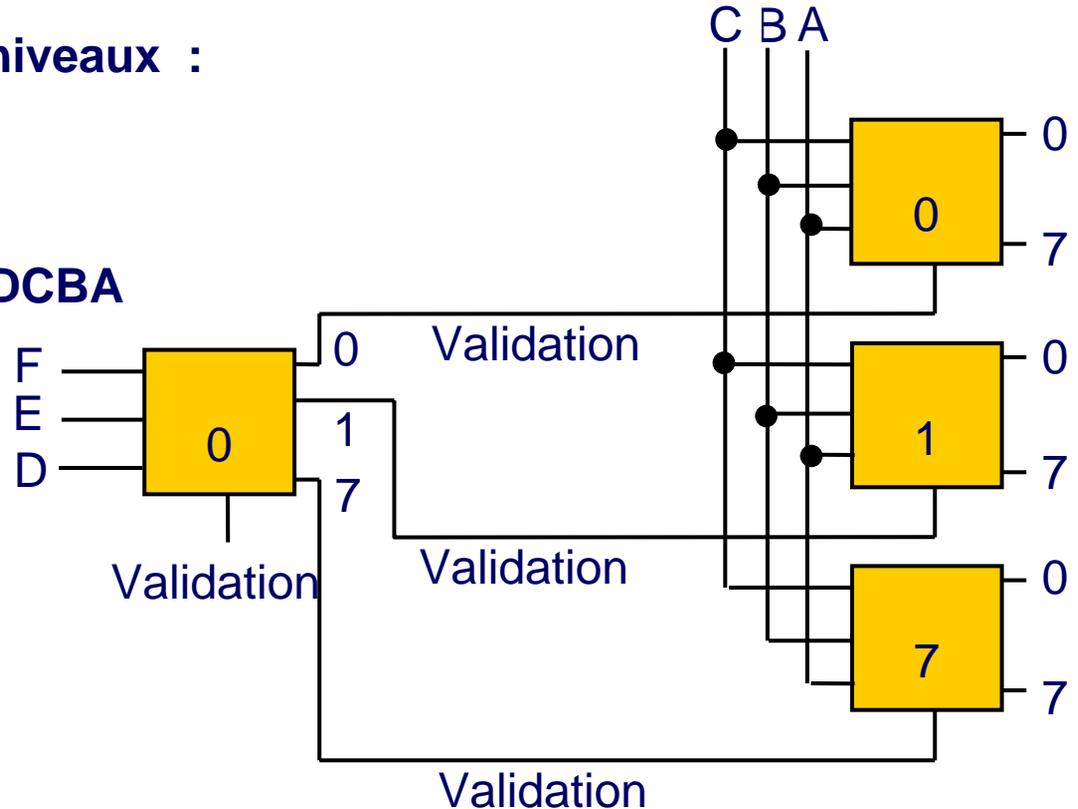
G_1	G_2	D	C	B	A	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

I – Les circuits de transcodage

Décodage à plusieurs niveaux :

Décodeur 1 parmi 64 :

6 entrées ($2^6 = 64$) : FEDCBA



8 décodeurs 1 parmi 8 + 1 de sélection :

FED sélectionnent un décodeur.

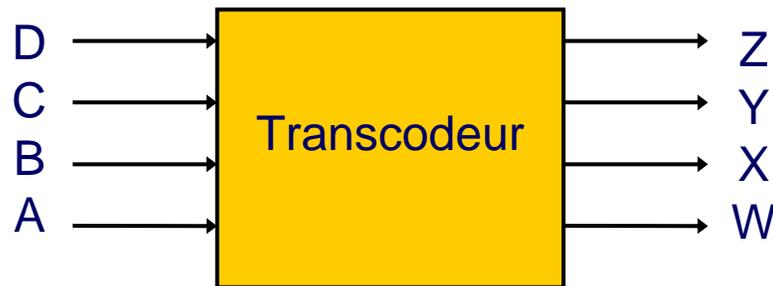
CBA sélectionnent une sortie sur le décodeur sélectionné.

Ce principe est également utilisé pour l'adressage des mémoires.

I – Les circuits de transcodage

I.3 – Les transcodeurs :

Exemple : Le transcodeur XS3 \Rightarrow AIKEN.



But : Exprimer les sorties W, X, Y et Z en fonction de D, C, B, A.

XS3				AIKEN			
D	C	B	A	Z	Y	X	W
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	1

I – Les circuits de transcodage

DC \ BA	00	01	11	10
00	X	X	0	X
01	0	0	0	0
11	1	X	X	X
10	1	1	1	1

$$Z = D$$

DC \ BA	00	01	11	10
00	X	X	0	X
01	0	0	1	0
11	1	X	X	X
10	0	1	1	1

$$Y = ABC + DC + DA + DB$$

DC \ BA	00	01	11	10
00	X	X	0	X
01	0	1	0	1
11	1	X	X	X
10	1	0	1	0

$$X = D\bar{A}\bar{B} + DAB + \bar{D}\bar{B}A + \bar{D}B\bar{A} = D \oplus B \oplus A$$

DC \ BA	00	01	11	10
00	X	X	0	X
01	1	0	0	1
11	1	X	X	X
10	1	0	0	1

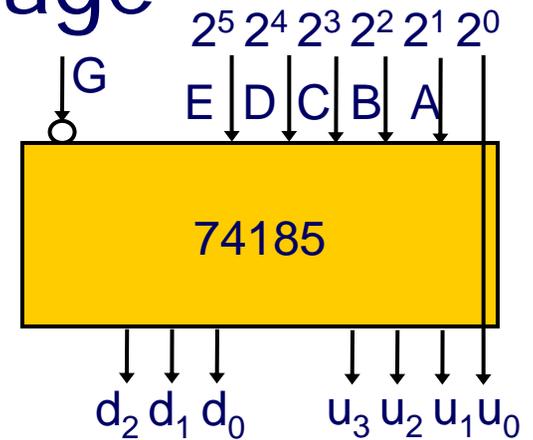
$$W = \bar{A}$$

I – Les circuits de transcodage

Exemples de transcodeurs :

Le convertisseur binaire \Rightarrow D.C.B. 74185.

$d_2 d_1 d_0$: Dizaines, $u_3 u_2 u_1 u_0$: Unités.



N	2^5	2^4	2^3	2^2	2^1	2^0	d_2	d_1	d_0	u_3	u_2	u_1	u_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
.
.
9	0	0	1	0	0	1	0	0	0	1	0	0	1
10	0	0	1	0	1	0	0	0	1	0	0	0	0
11	0	0	1	0	1	1	0	0	1	0	0	0	1
12	0	0	1	1	0	0	0	0	1	0	0	1	0
13	0	0	1	1	0	1	0	0	1	0	0	1	1
.

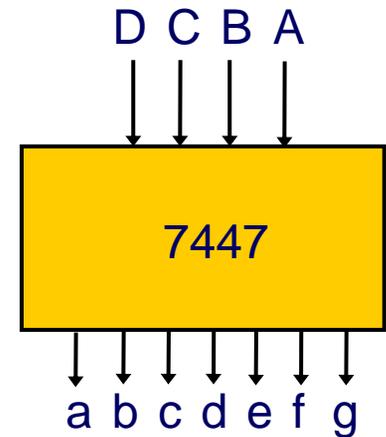
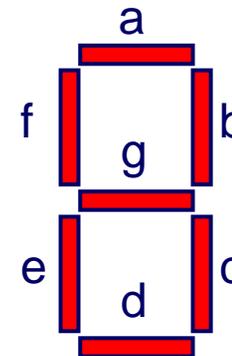
I – Les circuits de transcodage

Le transcodeur D.C.B. 7 segments 7447.

Ce transcodeur traduit visuellement un nombre binaire codé en entrée (4 bits).

Chaque segment (LED) est allumé ou éteint en fonction du nombre à visualiser.

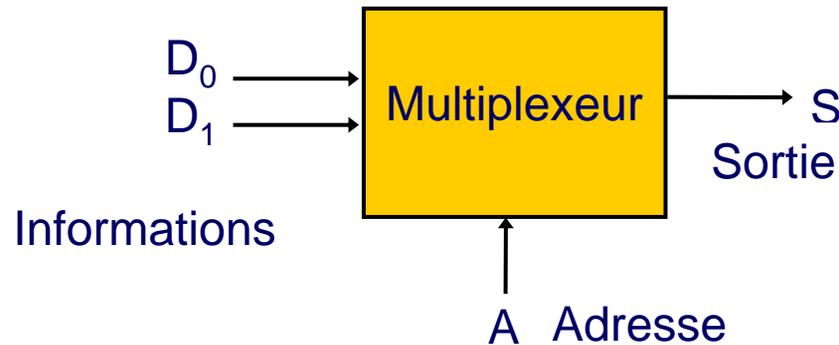
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
.



II – Les circuits d'aiguillage

II.1 – Les multiplexeurs :

Exemple : Le multiplexeur 2 vers 1.



C'est un circuit à 2^N entrées d'informations, N entrées d'adresses et une sortie.

Il redirige sur la sortie l'entrée sélectionnée à l'aide des bits d'adresses.

A	D ₀	D ₁	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

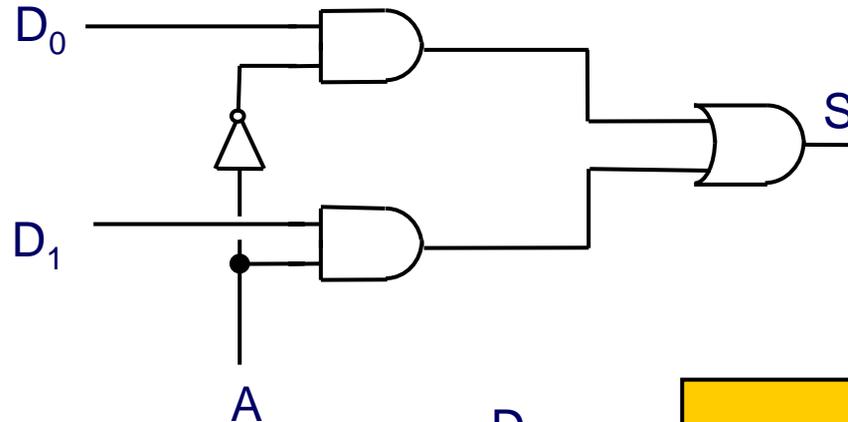
$$A = 0 \Rightarrow S = D_0$$

$$A = 1 \Rightarrow S = D_1$$

$$S = \overline{A} D_0 + A D_1$$

A \ D ₀ D ₁	00	01	11	10
0	0	0	1	1
1	0	1	1	0

II – Les circuits d'aiguillage



Le multiplexeur 4 vers 1 :

$B A = 00 \Rightarrow S = D_0$

$B A = 01 \Rightarrow S = D_1$

$B A = 10 \Rightarrow S = D_2$

$B A = 11 \Rightarrow S = D_3$



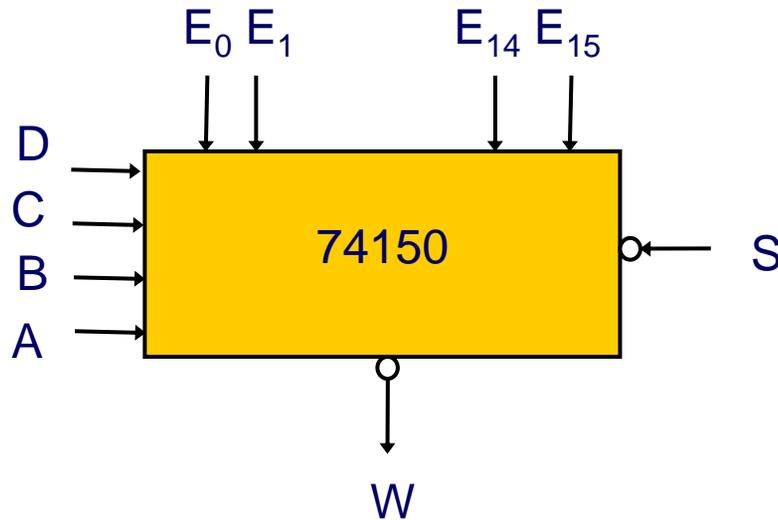
$B \ A \ V = 1$: Multiplexeur

$V = 0$: $S = 0$

$$S = V (\overline{B} \overline{A} D_0 + \overline{B} A D_1 + B \overline{A} D_2 + B A D_3)$$

II – Les circuits d'aiguillage

Le circuit intégré SN 74150 :



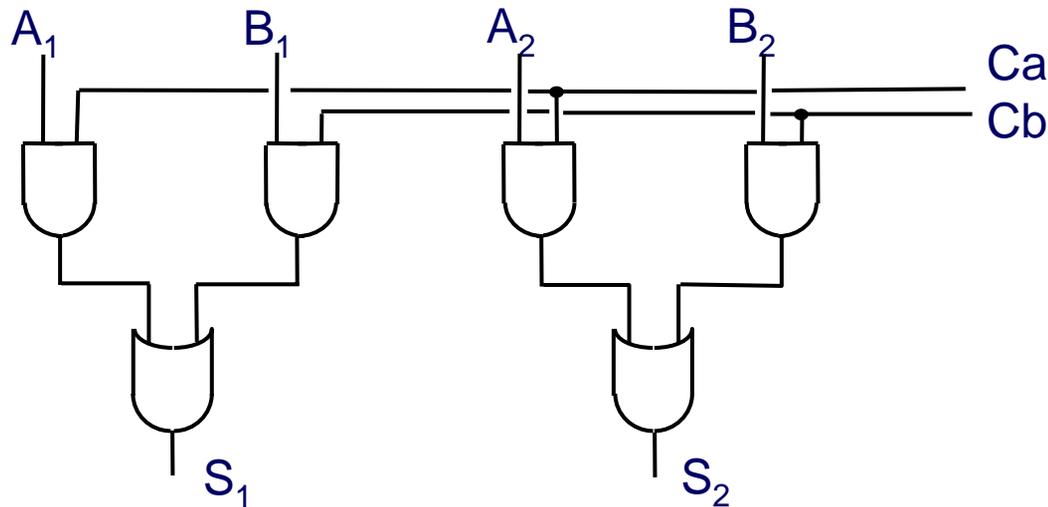
- 1 sortie complémentée.
- 1 entrée de validation (active à 0).
- 4 bits d'adresses.
- 16 entrées d'informations.

S = 0 : Mode multiplexeur.

D	C	B	A	S	W
X	X	X	X	1	1
0	0	0	0	0	$\overline{E_0}$
0	0	0	1	0	$\overline{E_1}$
0	0	1	0	0	$\overline{E_2}$
0	0	1	1	0	$\overline{E_3}$
0	1	0	0	0	$\overline{E_4}$
0	1	0	1	0	$\overline{E_5}$
0	1	1	0	0	$\overline{E_6}$
0	1	1	1	0	$\overline{E_7}$
1	0	0	0	0	$\overline{E_8}$
1	0	0	1	0	$\overline{E_9}$
1	0	1	0	0	$\overline{E_{10}}$
1	0	1	1	0	$\overline{E_{11}}$
1	1	0	0	0	$\overline{E_{12}}$
1	1	0	1	0	$\overline{E_{13}}$
1	1	1	0	0	$\overline{E_{14}}$
1	1	1	1	0	$\overline{E_{15}}$

II – Les circuits d'aiguillage

Les multiplexeurs à plusieurs sorties :

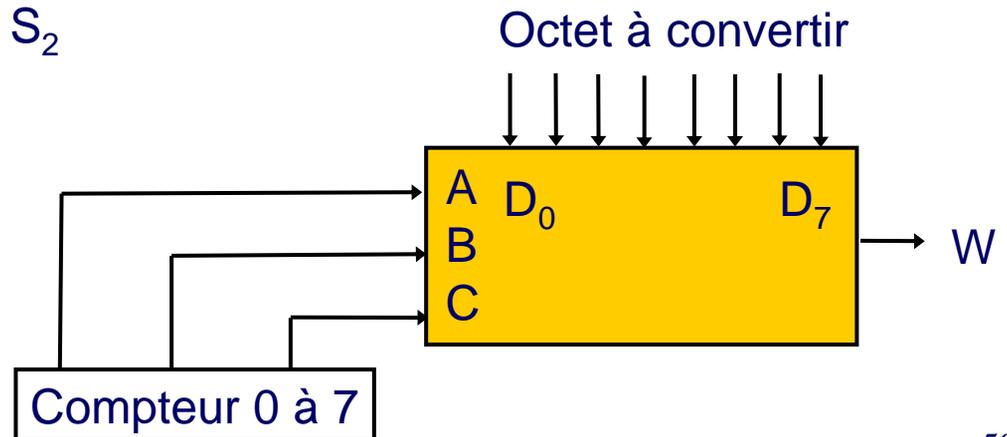


$$C_a = 1 \ (C_b = 0) \Rightarrow S_n = A_n$$
$$C_b = 1 \ (C_a = 0) \Rightarrow S_n = B_n$$

$$S_1 = C_a A_1 + C_b B_1$$
$$S_2 = C_a A_2 + C_b B_2$$
$$S_n = C_a A_n + C_b B_n$$

Applications :

Conversion parallèle série.

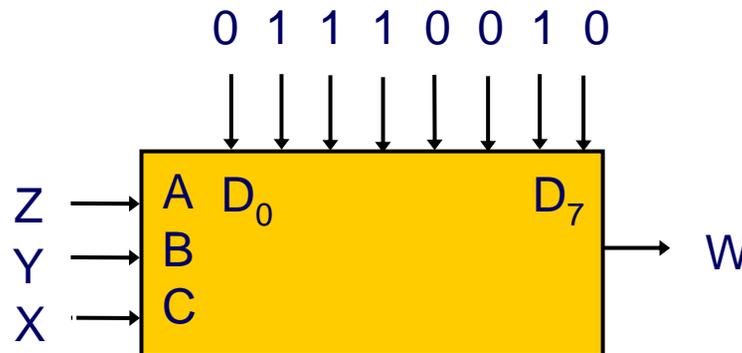


II – Les circuits d'aiguillage

Générateur de fonctions.

On veut réaliser : $\overline{X}\overline{Y}Z + \overline{X}Y\overline{Z} + \overline{X}YZ + XY\overline{Z}$

Sortie du multiplexeur : $W = D_0 \overline{C}\overline{B}\overline{A} + D_1 \overline{C}\overline{B}A + D_2 \overline{C}B\overline{A} + D_3 \overline{C}BA + D_4 C\overline{B}\overline{A} + D_5 C\overline{B}A + D_6 C B\overline{A} + D_7 CBA$

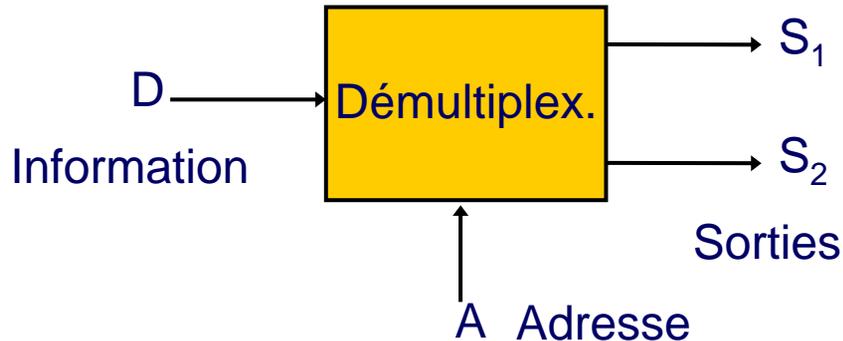


Aucun câblage de portes !

II – Les circuits d'aiguillage

II.2 – Les démultiplexeurs :

Exemple : Le démultiplexeur 1 vers 2.



A	D	S ₁	S ₂
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

C'est un circuit à 1 entrée d'informations, N entrées d'adresses et 2^N sorties.

Il redirige l'entrée d'information sur l'une des sorties sélectionnée à l'aide des bits d'adresses.

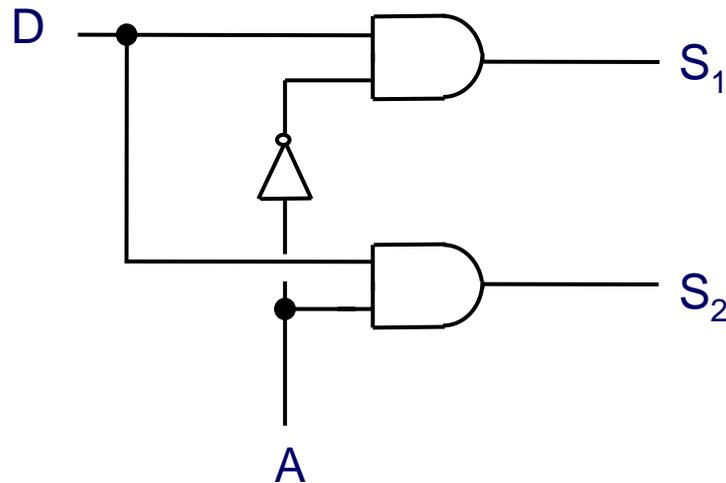
$$A = 0 \Rightarrow S_1 = D \text{ et } S_2 = 0$$

$$A = 1 \Rightarrow S_2 = D \text{ et } S_1 = 0$$

$$S_1 = \overline{A}D$$

$$S_2 = AD$$

II – Les circuits d'aiguillage



Le démultiplexeur 1 vers 4 :

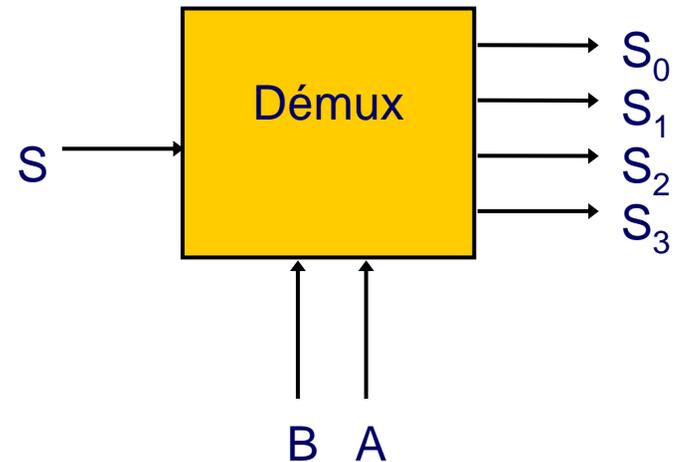
$$B A = 00 \Rightarrow S_0 = D$$

$$B A = 01 \Rightarrow S_1 = D$$

$$B A = 10 \Rightarrow S_2 = D$$

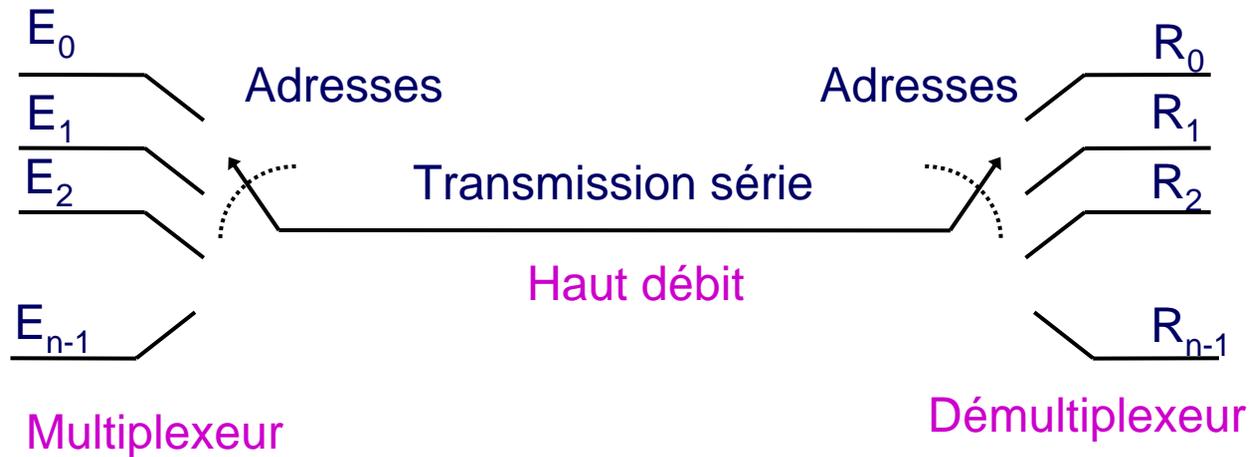
$$B A = 11 \Rightarrow S_3 = D$$

$$\begin{aligned} S_0 &= \overline{B} \overline{A} D \\ S_1 &= \overline{B} A D \\ S_2 &= B \overline{A} D \\ S_3 &= B A D \end{aligned}$$



II – Les circuits d'aiguillage

Application : **Liaison multiplex.**



III – Les circuits de comparaison

III.1 – Les comparateurs :

Exemple : Le comparateur à 2 éléments binaires (2 e.b.).

a	b	S_1 a > b	S_2 a < b	S_3 a = b
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

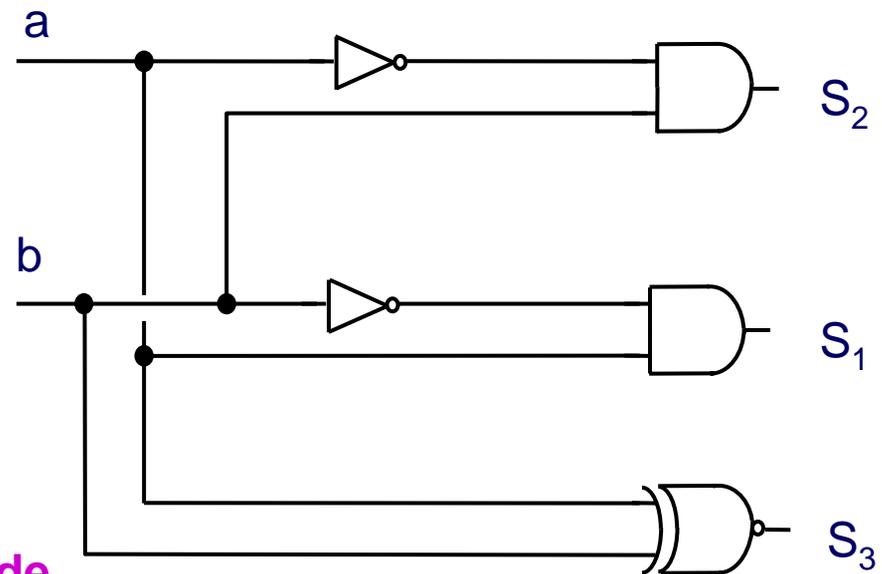
$$S_1 = a \overline{b}$$

$$S_2 = \overline{a} b$$

$$S_3 = a \oplus b$$

Méthode lourde à partir de
2 mots de 2 bits !

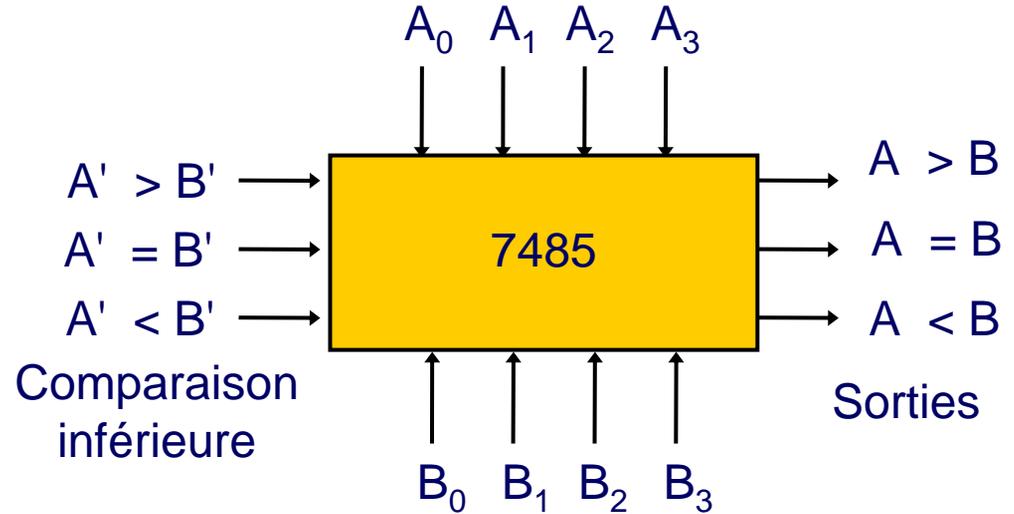
1 sortie « plus grand que ».
1 sortie « plus petit que ».
1 sortie « égalité ».



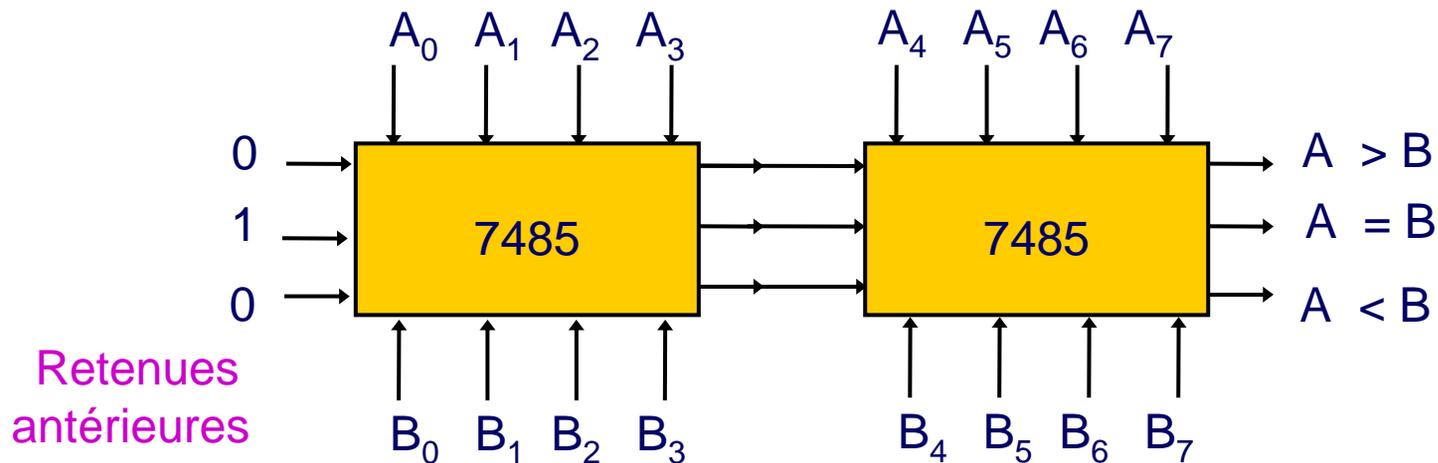
III – Les circuits de comparaison

Le circuit intégré 7485 :

Comparaison de 2 mots de 4 bits (A et B) avec entrées de mise en cascade.



Comparaison de deux octets A et B :



III – Les circuits de comparaison

III.2 – Les contrôleurs et générateurs de clé de parité :

Exemple : Information globalement impaire sur 2 bits.

a_1	a_0	S
0	0	1
0	1	0
1	0	0
1	1	1

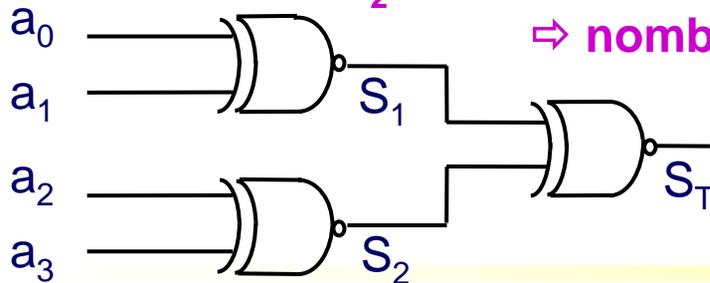
$$S = a \oplus b$$

Information globalement impaire sur 4 bits :

$S_1 = 0$: Nombre impaire de 1 dans (a_0, a_1)

$S_2 = 0$: Nombre impaire de 1 dans (a_2, a_3)

\Rightarrow nombre total : pair $\Rightarrow S_T = 1$

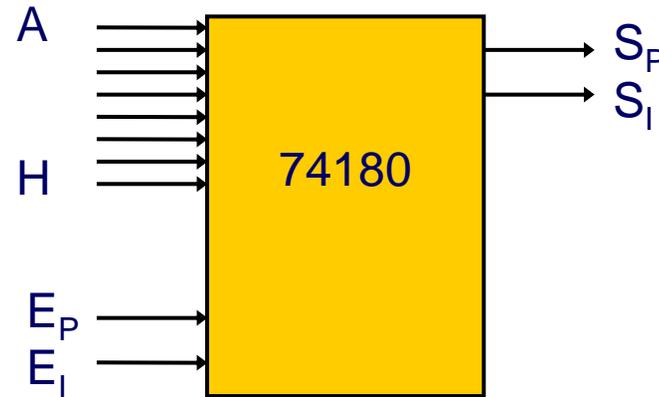


S_1	S_2	S_T
0	0	1
0	1	0
1	0	0
1	1	1

III – Les circuits de comparaison

Le circuit intégré 74180 :

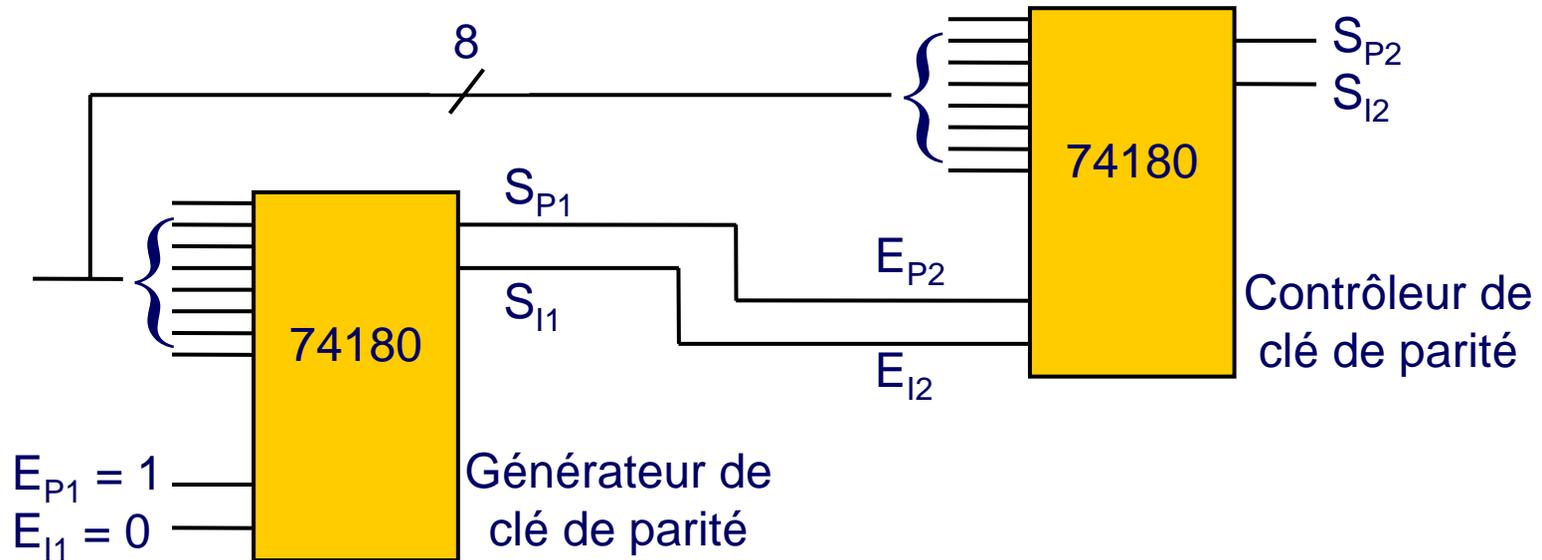
Contrôleur et générateur de clé de parité sur 8 bits :
8 bits d'informations.
2 entrées de parité.
2 sorties paire/impaire.



Nombre de 1 sur A,B,C...H	Entrée paire E_P	Entrée impaire E_I	Sortie paire S_P	Sortie impaire S_I
Pair	1	0	1	0
Impair	1	0	0	1
Pair	0	1	0	1
Impair	0	1	1	0
Quelconque	1	1	0	0
Quelconque	0	0	1	1

III – Les circuits de comparaison

Fonctionnement en contrôleur et générateur de clé de parité :



Émission : Nombre pair de 1 (8bits) ($E_{P1}=1$ et $E_{I1}=0$) $\Rightarrow S_{P1}=1$ et $S_{I1}=0$

Réception : Nombre pair de 1 ($E_{P2}=1$ et $E_{I2}=0$) $\Rightarrow S_{P2}=1$ et $S_{I2}=0$ OK
Nombre impair de 1 ($E_{P2}=0$ et $E_{I2}=0$) $\Rightarrow S_{P2}=0$ et $S_{I2}=1$ Erreur

Valable également pour un nombre impair de 1 sur les 8 bits.

IV – Les circuits arithmétiques

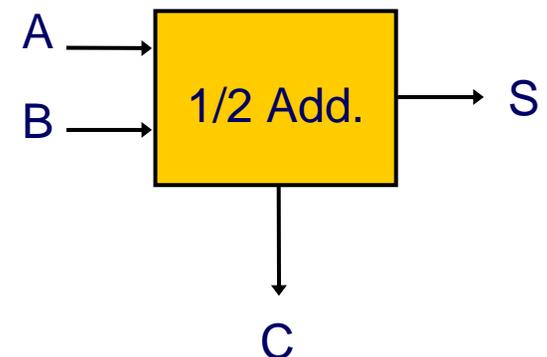
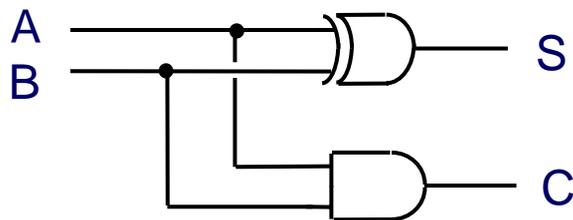
IV.1 – L'élément de base : Le demi-additionneur.

Calcul sur deux éléments binaires.

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

$$S = A \oplus B \quad \text{Somme}$$
$$C = A \cdot B \quad \text{Retenue}$$

Le demi additionneur additionne deux bits mais ne tient pas compte de la retenue précédente.



IV – Les circuits arithmétiques

IV.2 – L'additionneur complet :

C_{i-1}	A_i	B_i	S_i	C_i
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

Exemple :

	C_2	C_1	C_0	
		A_2	A_1	A_0
+		B_2	B_1	B_0
	C_2	S_2	S_1	S_0

L'additionneur complet additionne deux bits A_i et B_i plus la retenue précédente C_{i-1} et donne une somme S_i et une retenue C_i .

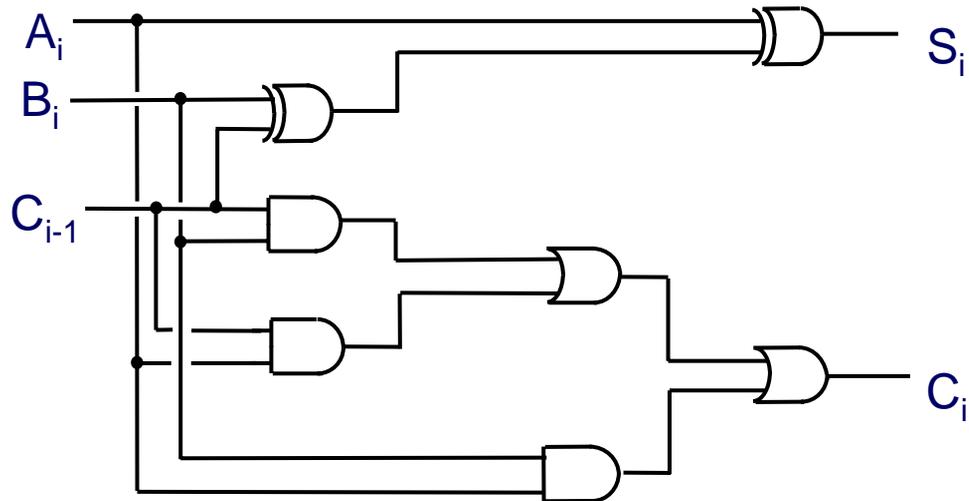
$C_{i-1} \setminus B_i A_i$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

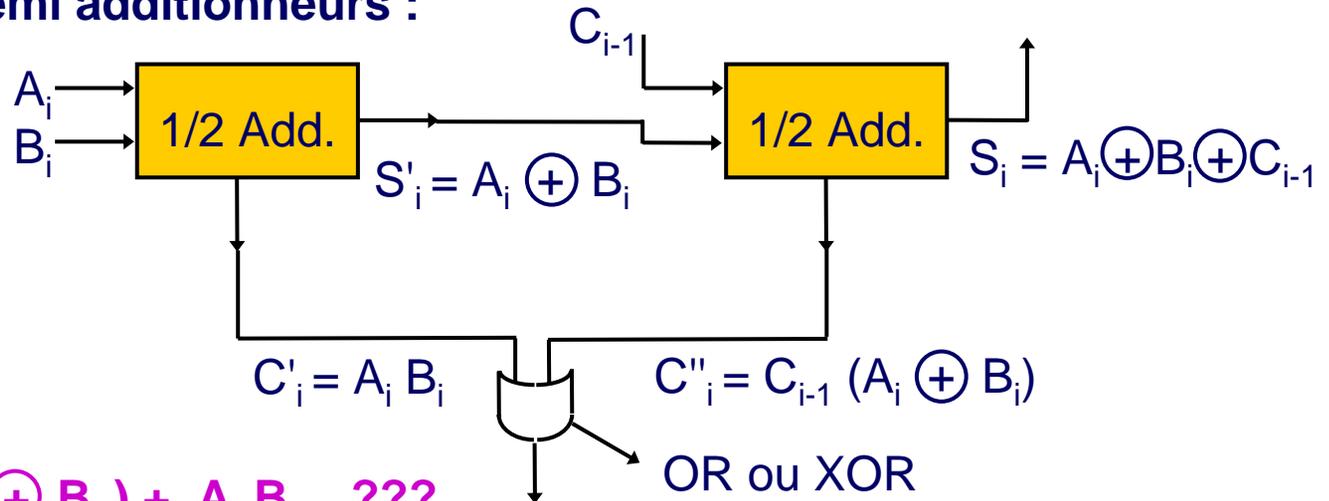
$C_{i-1} \setminus B_i A_i$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$C_i = C_{i-1} A_i + C_{i-1} B_i + A_i B_i$$

IV – Les circuits arithmétiques



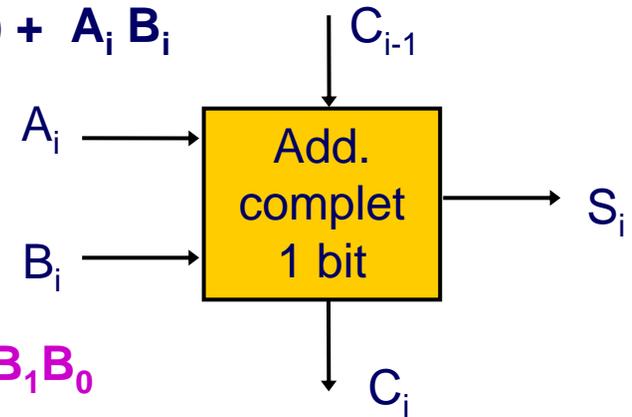
Structure en demi additionneurs :



$$C_i = C_{i-1} (A_i \oplus B_i) + A_i B_i \quad ???$$

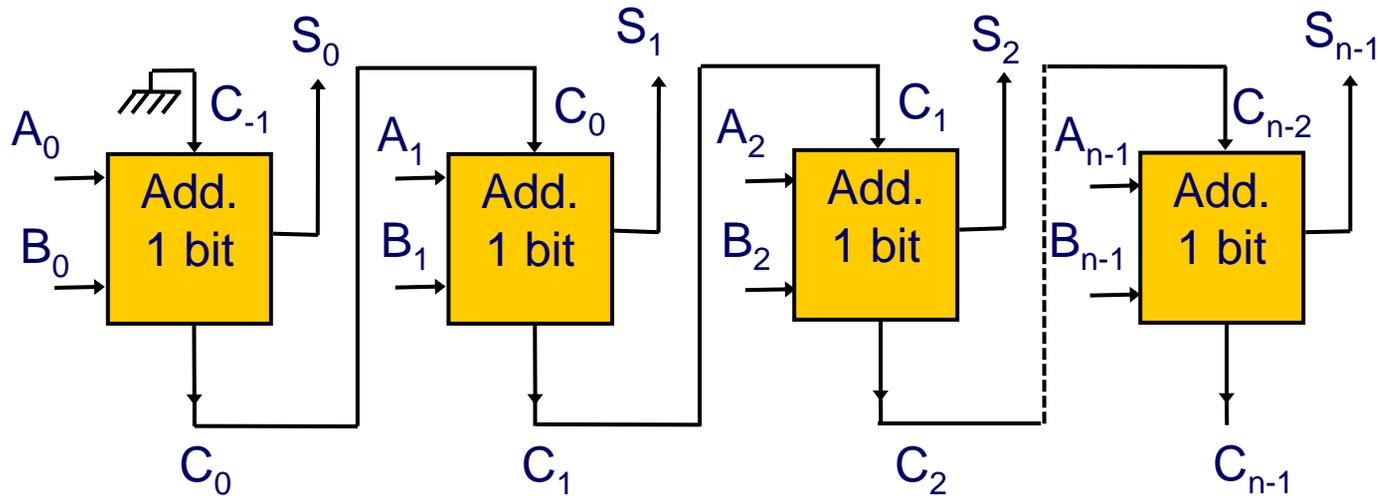
IV – Les circuits arithmétiques

$$C_i = C_{i-1} (A_i \oplus B_i) + A_i B_i \quad ?? = C_{i-1} (A_i + B_i) + A_i B_i$$



Addition sur n bits :

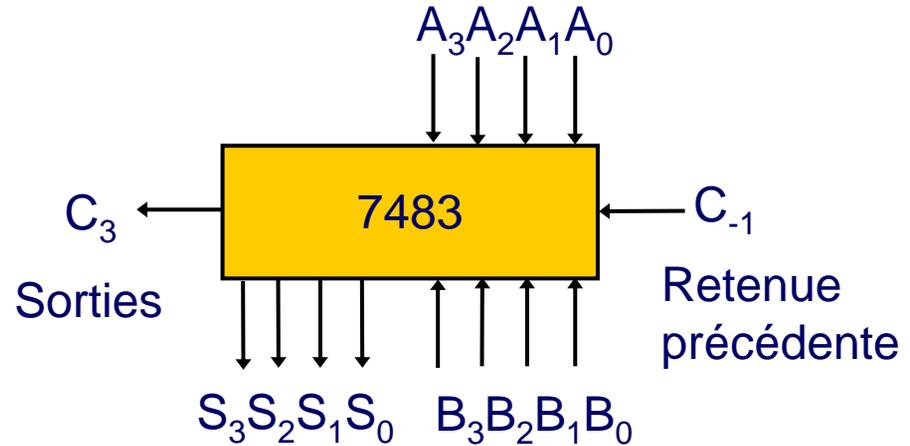
$$A_{n-1} A_{n-2} \dots A_1 A_0 + B_{n-1} B_{n-2} \dots B_1 B_0 = C_{n-1} S_{n-1} S_{n-2} \dots S_1 S_0$$



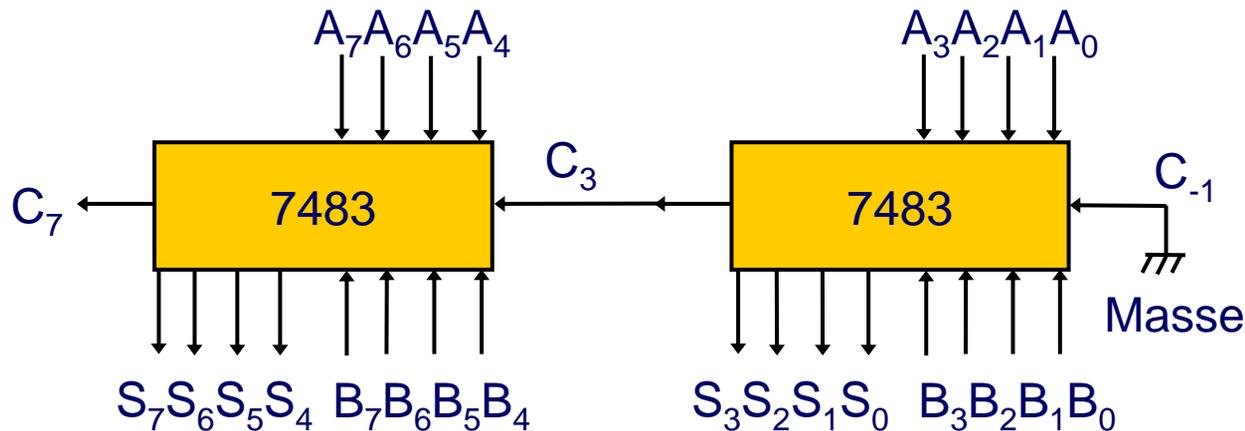
IV – Les circuits arithmétiques

Le circuit intégré 7483 (additionneur parallèle) :

Addition de 2 mots de 4 bits (A et B) avec entrées de mise en cascade.



Addition de deux octets A et B :



Le résultat est codé sur 9 bits S_0 à S_7 et C_7 .

IV – Les circuits arithmétiques

La retenue anticipée :

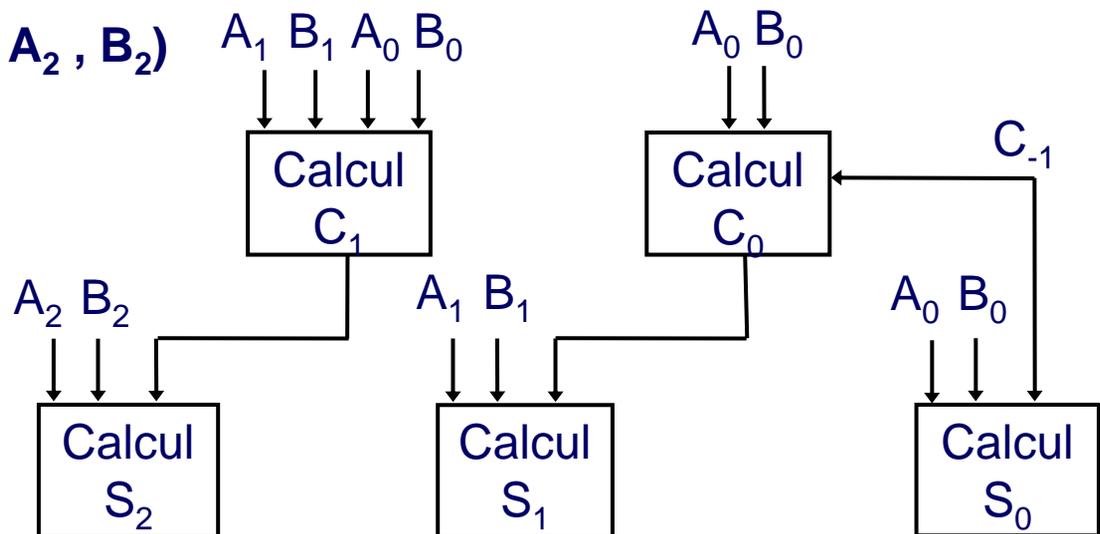
La propagation de la retenue ralentit le calcul de l'additionneur !

$$C_i = C_{i-1} (A_i + B_i) + A_i B_i$$

$C_0 = A_0 B_0 = F(A_0, B_0)$: Pas besoin de S_0 pour le calcul de C_0

$C_1 = C_0 (A_1 + B_1) + A_1 B_1 = F(A_0, B_0, A_1, B_1)$: Pas besoin de S_0, S_1

$$C_2 = f(A_0, B_0, A_1, B_1, A_2, B_2)$$



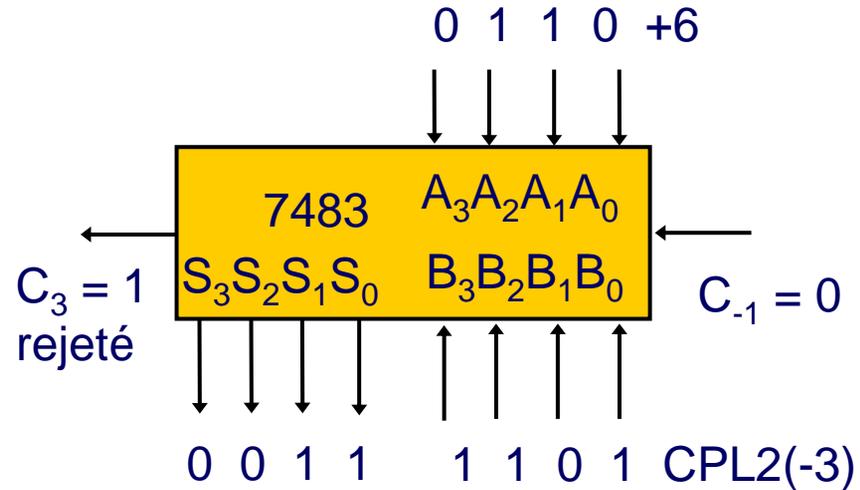
IV – Les circuits arithmétiques

IV.3 – Le soustracteur :

Un soustracteur se réalise à partir d'un additionneur à l'aide du complément à 2.

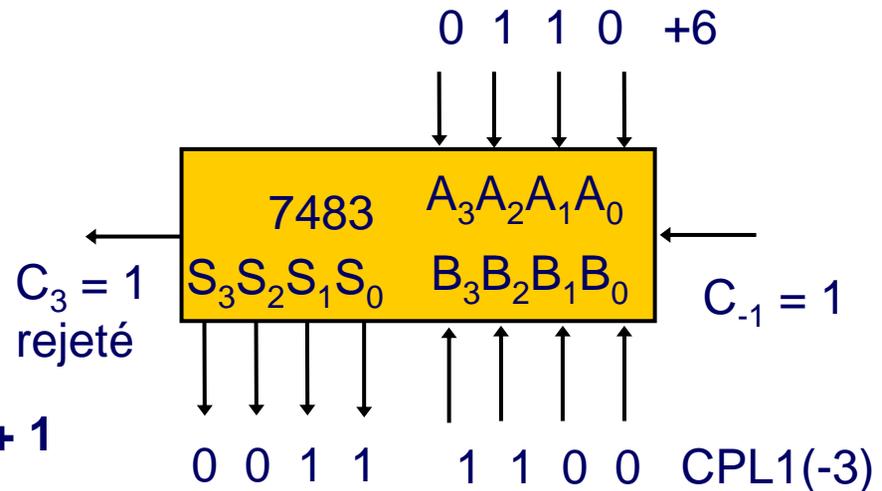
$$6 - 3 = 6 + (-3) = 6 + \text{CPL2}(-3)$$

$C_{-1} = 0$



$$6 - 3 = 6 + (-3) = 6 + \text{CPL1}(-3) + 1$$

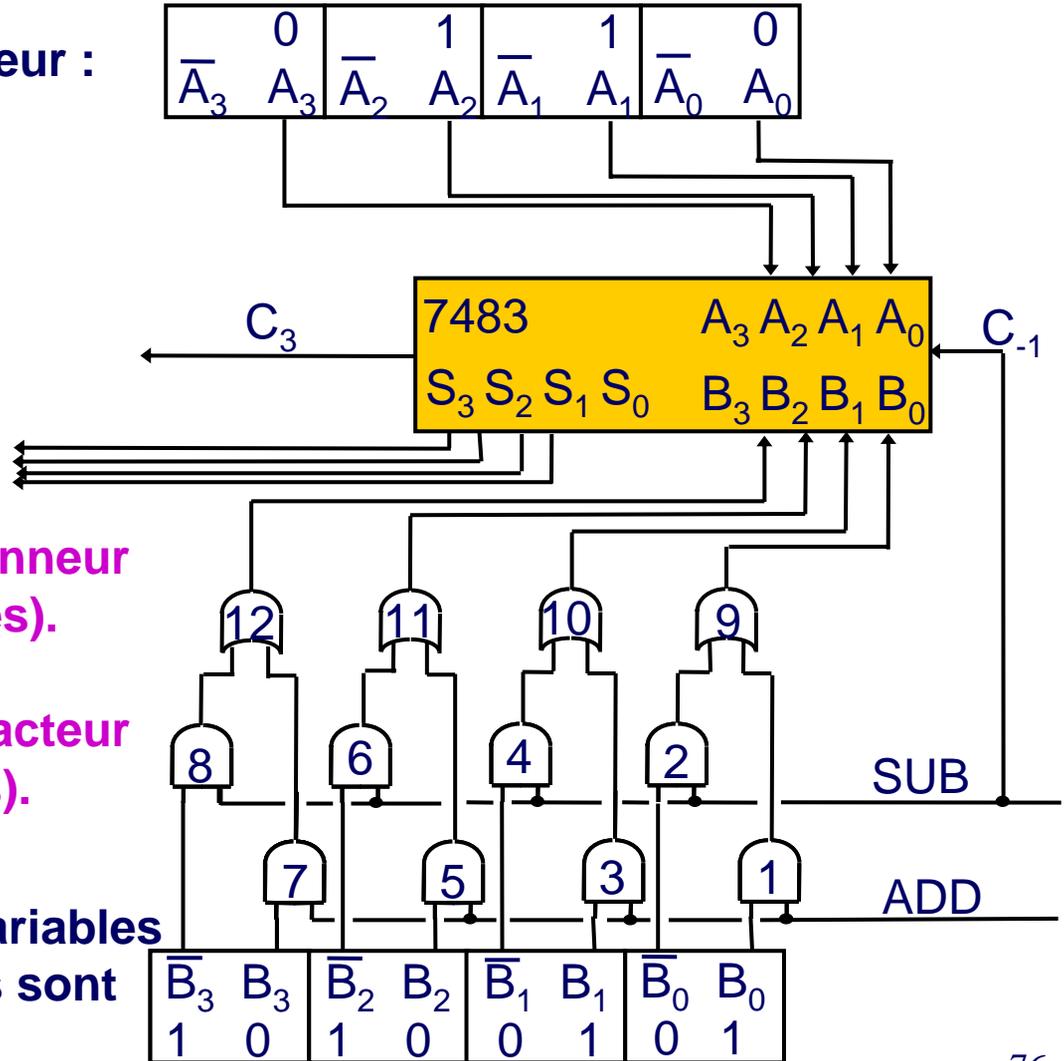
$C_{-1} = 1$



IV – Les circuits arithmétiques

L'additionneur / soustracteur :

$S = A+B$ ou $A-B$ selon les états de SUB et ADD.



ADD = 1, SUB = 0 \Rightarrow additionneur
(portes 2, 4, 6, 8 bloquées).

ADD = 0, SUB = 1 \Rightarrow soustracteur
(portes 1, 3, 5, 7 bloquées).

Registre de mémoire (les variables directes ou complémentées sont accessibles).

IV – Les circuits arithmétiques

IV.4 – L'additionneur DCB :

Si la somme est inférieure à 9 l'additionneur DCB est égal à l'additionneur binaire.

Sinon, une correction de + 6 est nécessaire.

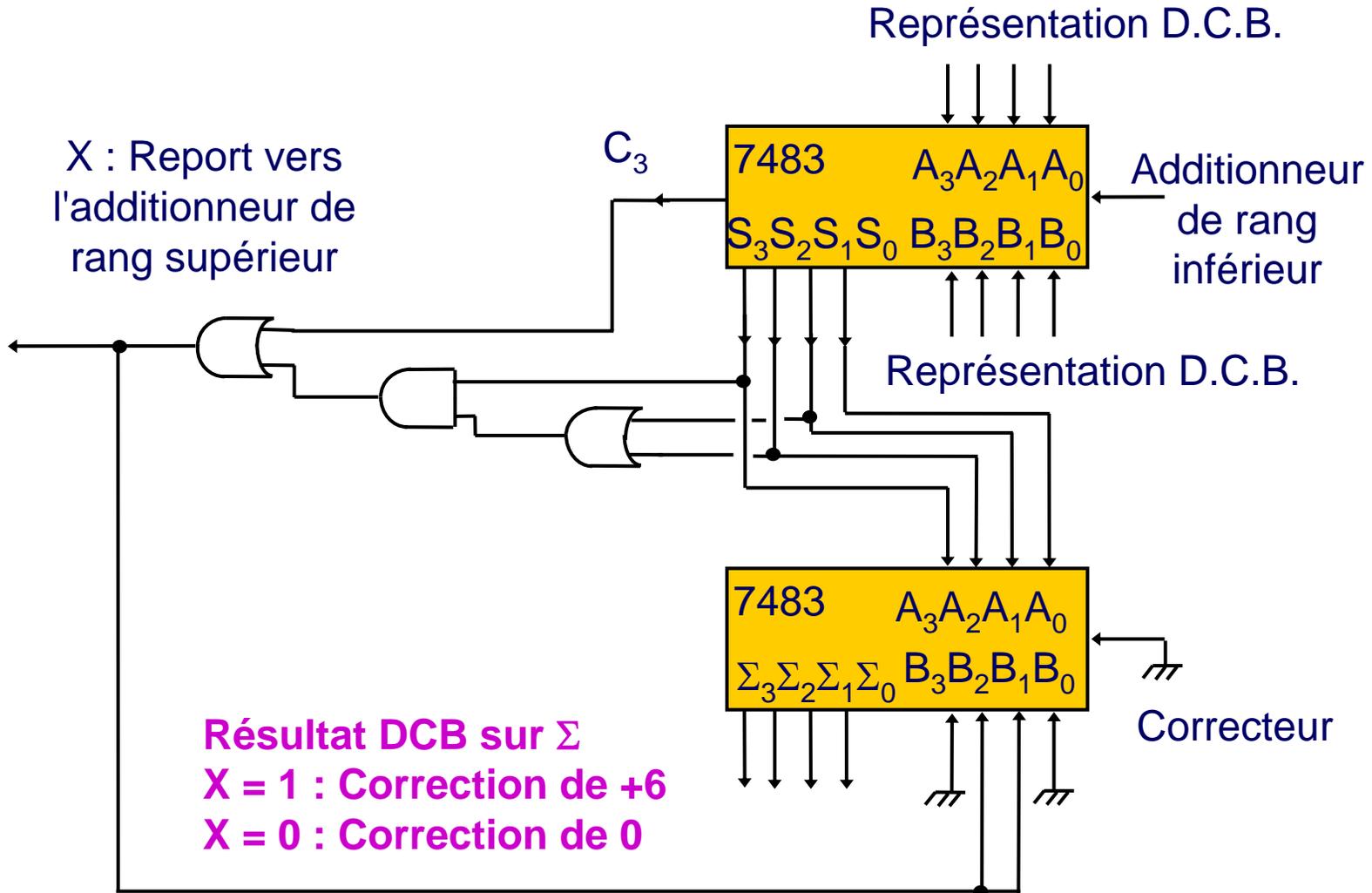
Cas en sortie de l'additionneur donnant lieu à une correction :

$X = 1 \Rightarrow$ Correction
 $\Rightarrow X = C_3 + S_3 (S_1 + S_2)$

Somme maximale en DCB \Rightarrow

C_3	S_3	S_2	S_1	S_0	
0	1	0	1	0	10
0	1	0	1	1	11
0	1	1	0	0	12
0	1	1	0	1	13
0	1	1	1	0	14
0	1	1	1	1	15
1	0	0	0	0	16
1	0	0	0	1	17
1	0	0	1	0	18

IV – Les circuits arithmétiques



IV – Les circuits arithmétiques

IV.5 – L'unité arithmétique et logique (ALU) :

Le circuit intégré 74181 :

Circuit permettant tous les calculs logiques et binaires sur 4 bits.

M = 1 ⇒ Logique.

M = 0 ⇒ Arithmétique.

S₃, S₂, S₁, S₀ : A configurer par soft.

SELECTION S ₃ S ₂ S ₁ S ₀	ACTIVE-HIGH DATA		
	M = H LOGIC FUNCTIONS	M = L : ARITHMETIC OPERATIONS	
		C _n = 0 C̄ _n = 1 = H	C _n = 1 C̄ _n = 0 = L
L L L L	F = \bar{A}	F = A	F = A PLUS 1
L L L H	F = $\overline{A+B}$	F = A + B	F = (A + B) PLUS 1
L L H L	F = $\bar{A}B$	F = A + \bar{B}	F = (A + \bar{B}) PLUS 1
L L H H	F = 0	F = MINUS 1 (2's COMPL)	F = ZERO
L H L L	F = \overline{AB}	F = A PLUS \overline{AB}	F = A PLUS \overline{AB} PLUS 1
L H L H	F = \bar{B}	F = (A + B) PLUS \overline{AB}	F = (A + B) PLUS \overline{AB} PLUS 1
L H H L	F = A ⊕ B	F = A MINUS B MINUS 1	F = A MINUS B
L H H H	F = \overline{AB}	F = \overline{AB} MINUS 1	F = \overline{AB}
H L L L	F = $\bar{A} + B$	F = A PLUS AB	F = A PLUS AB PLUS 1
H L L H	F = $\overline{A \oplus B}$	F = A PLUS B	F = A PLUS B PLUS 1
H L H L	F = B	F = (A + \bar{B}) PLUS AB	F = (A + \bar{B}) PLUS AB PLUS 1
H L H H	F = AB	F = AB MINUS 1	F = AB
H H L L	F = 1	F = A PLUS A*	F = A PLUS A PLUS 1
H H L H	F = A + \bar{B}	F = (A + B) PLUS A	F = (A + B) PLUS A PLUS 1
H H H L	F = A + B	F = (A + \bar{B}) PLUS A	F = (A + \bar{B}) PLUS A PLUS 1
H H H H	F = A	F = A MINUS 1	F = A

Les circuits séquentiels

Un circuit séquentiel est un circuit pour lequel un état des variables d'entrées peut engendrer plusieurs états des sorties (lié à un état mémoire par exemple).

L'état des sorties dépend des entrées mais également des états précédents du circuit.

I – La fonction mémoire

I.1 – Définitions :

Régime permanent stable : Le circuit reste dans le même état et ne peut être modifié que par une action sur les entrées.

Régime permanent instable : Le circuit évolue sans jamais atteindre d'états stables (oscillations 0, 1,).

Régime transitoire : Le circuit évolue vers un nouveau régime qui peut être permanent ou transitoire.

Pour qu'une fonction binaire soit une fonction mémoire, il faut et il suffit qu'il existe trois groupes distincts de combinaisons de variables tels que :

- Toute combinaison d'un groupe entraîne $S = 0$.
- Toute combinaison d'un groupe entraîne $S = 1$.
- Toute combinaison d'un groupe entraîne $S_{n+1} = S_n$.

Le circuit de mémorisation le plus simple est appelé bascule.

I – La fonction mémoire

La fonction de commutation est définie telle que :

$e_1(t)$	$e_2(t)$	$s(t)$	$s(t + \tau)$	φ
0	0	0	0	0
0	1	0	1	1
1	1	0	1	1
1	0	0	0	0
0	0	1	0	1
0	1	1	0	1
1	1	1	1	0
1	0	1	1	0

$\varphi = 1$ si $s(t+\tau) \neq s(t)$

$\varphi = 0$ si $s(t+\tau) = s(t)$
(fonction mémoire).

$s(t+\tau)$ et $s(t)$ symbolisent deux états de la même sortie à des instants différents (notés également S_n et S_{n+1}).

II – Les types de bascules

4 types de bascules :

E \Rightarrow Effacement.

I \Rightarrow Inscription.

E	I	S_{n+1}	
0	0	S_n	Mémoire
0	1	1	Mise à 1
1	0	0	Mise à 0
1	1	1	Inscription prioritaire

E	I	S_{n+1}	
0	0	S_n	Mémoire
0	1	1	Mise à 1
1	0	0	Mise à 0
1	1	0	Effacement prioritaire

E	I	S_{n+1}	
0	0	S_n	Mémoire
0	1	1	Mise à 1
1	0	0	Mise à 0
1	1	S_n	Sans priorité

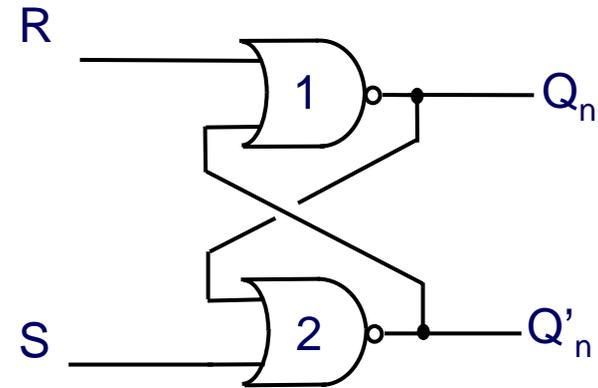
E	I	S_{n+1}	
0	0	S_n	Mémoire
0	1	1	Mise à 1
1	0	0	Mise à 0
1	1	$\overline{S_n}$	Changement prioritaire

II – Les types de bascules

II.1 – La bascule RS :

La bascule RS NOR.

$R = 1$ et $S = 0 \Rightarrow Q_{n+1} = 0$ et $Q'_{n+1} = 1$
 $R = 0$ et $S = 1 \Rightarrow Q'_{n+1} = 0$ et $Q_{n+1} = 1$
 $R = 0$ et $S = 0 \Rightarrow Q_{n+1} = Q_n$ et $Q'_n = \overline{Q_n}$
 $R = 1$ et $S = 1 \Rightarrow Q_{n+1} = 0$ et $Q'_{n+1} = 0$



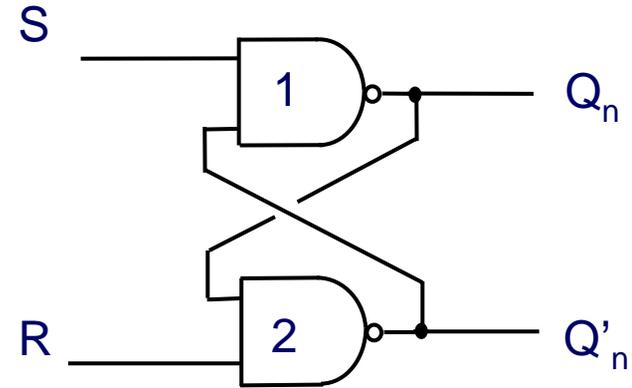
R	S	Q_{n+1}	Q'_{n+1}	
0	0	Q_n	$\overline{Q_n}$	Mémoire
0	1	1	0	Mise à 1
1	0	0	1	Mise à 0
1	1	0 *	0 *	Effacement prioritaire

- $R \Rightarrow$ Reset, $S \Rightarrow$ Set.
- Signaux actifs à 1.
- Sorties complémentées sauf pour $R = S = 1$.
- Cas ambiguë si les entrées passent de $R = S = 1$ à $R = S = 0$.

II – Les types de bascules

La bascule RS NAND.

$R = 1$ et $S = 0 \Rightarrow Q_{n+1} = 1$ et $Q'_{n+1} = 0$
 $R = 0$ et $S = 1 \Rightarrow Q'_{n+1} = 1$ et $Q_{n+1} = 0$
 $R = 1$ et $S = 1 \Rightarrow Q_{n+1} = Q_n$ et $Q'_n = \overline{Q_n}$
 $R = 0$ et $S = 0 \Rightarrow Q_{n+1} = 1$ et $Q'_{n+1} = 1$



\overline{R}	\overline{S}	Q_{n+1}	Q'_{n+1}
1	1	Q_n	$\overline{Q_n}$
1	0	1	0
0	1	0	1
0	0	1 *	1 *

Mémoire

Mise à 1

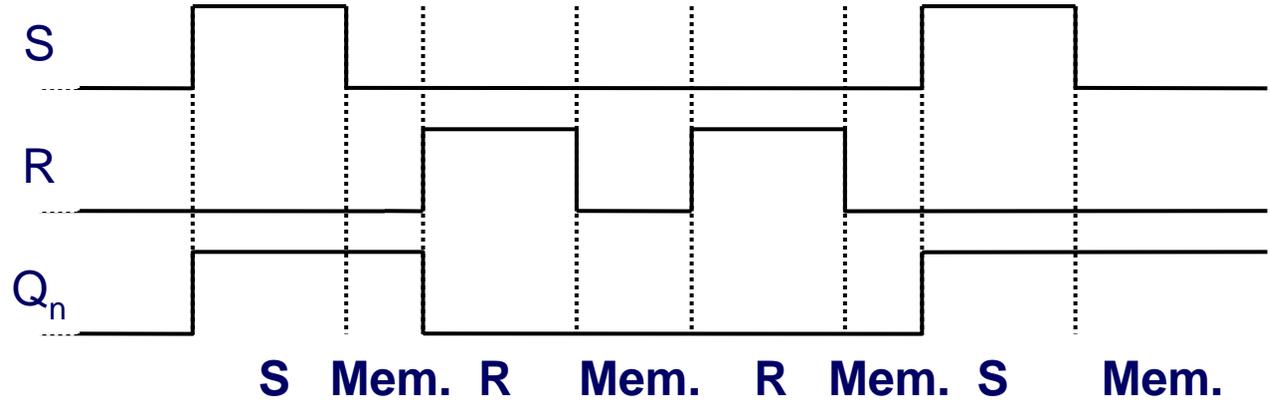
Mise à 0

Inscription
prioritaire

- $R \Rightarrow$ Reset, $S \Rightarrow$ Set.
- Signaux actifs à 0 (\overline{R} , \overline{S}).
- Sorties complémentées sauf pour $\overline{R} = \overline{S} = 0$.
- Cas ambiguë si les entrées passent de $\overline{R} = \overline{S} = 0$ à $\overline{R} = \overline{S} = 1$.

II – Les types de bascules

Chronogramme (RS NOR) :

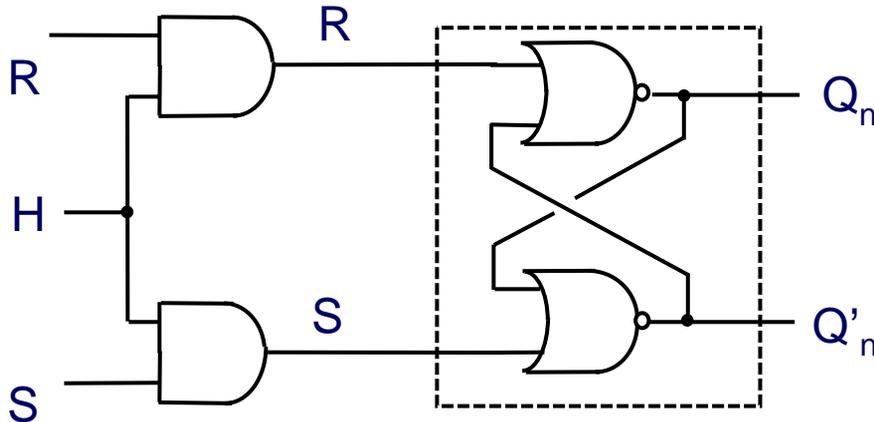


II.2 – La bascule RSH :

En portes NOR :

H = 1 ⇒ Bascule RS NOR.

H = 0 ⇒ Mémoire.

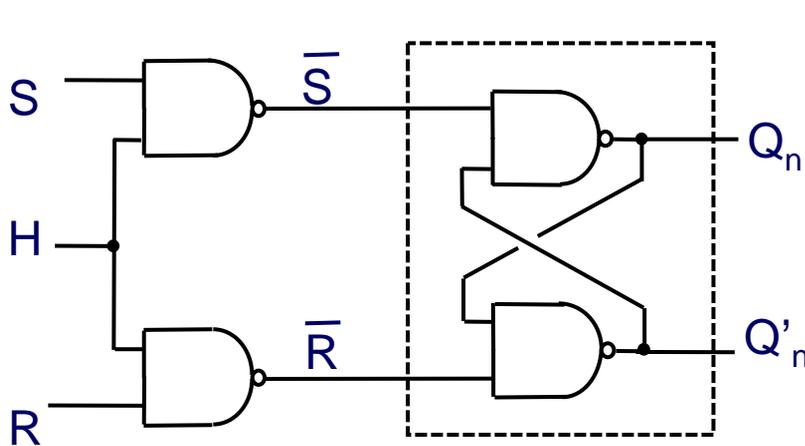


H	R	S	Q_{n+1}	Q'_{n+1}
1	0	0	Q_n	$\overline{Q_n}$
1	0	1	1	0
1	1	0	0	1
1	1	1	0 *	0 *
0	X	X	Q_n	$\overline{Q_n}$

H synchronise les entrées : R et S deviennent entrées synchrones.

II – Les types de bascules

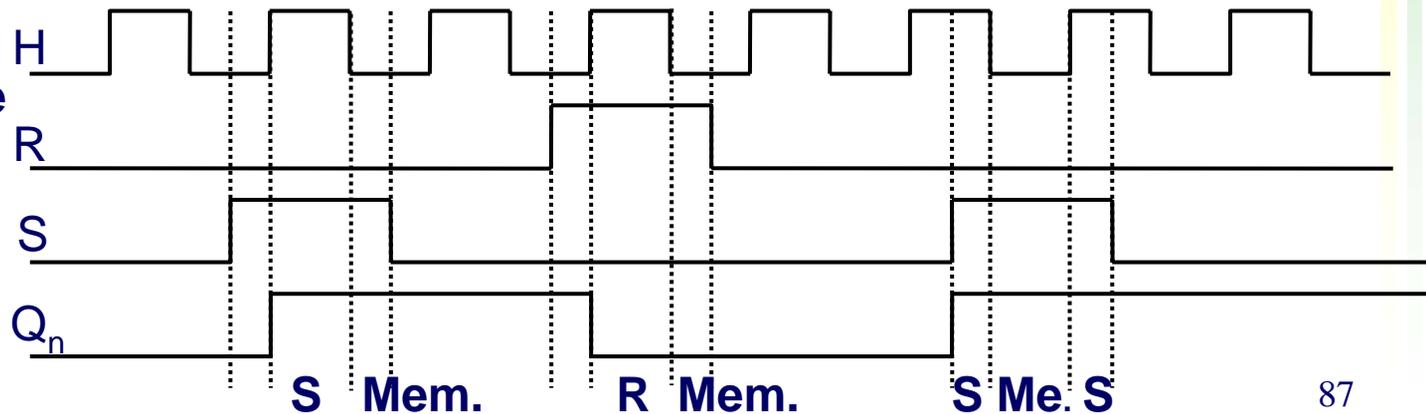
En portes NAND : $H = 1 \Rightarrow$ Bascule RS NAND.
 $H = 0 \Rightarrow$ Mémoire.



R et S redeviennent actifs à 1.

H	R	S	\overline{R}	\overline{S}	Q_{n+1}	Q'_{n+1}
1	0	0	1	1	Q_n	$\overline{Q_n}$
1	0	1	1	0	1	0
1	1	0	0	1	0	1
1	1	1	0	0	1*	1*
0	X	X	1	1	Q_n	$\overline{Q_n}$

Chronogramme (RSH NOR) :



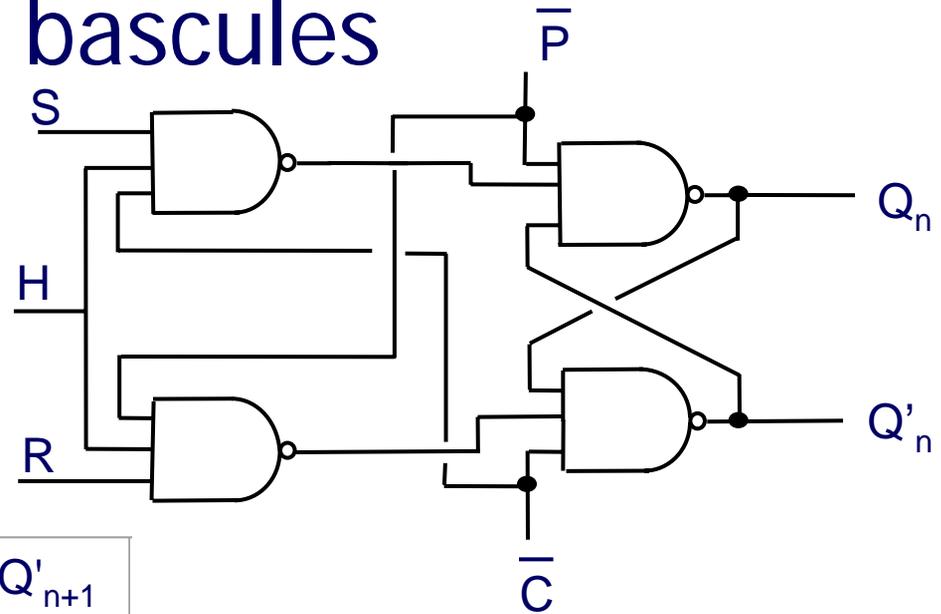
II – Les types de bascules

Entrées de forçages :

Clear (\overline{C}) et Preset (\overline{P}) :
Entrées asynchrones
(indépendantes de H).

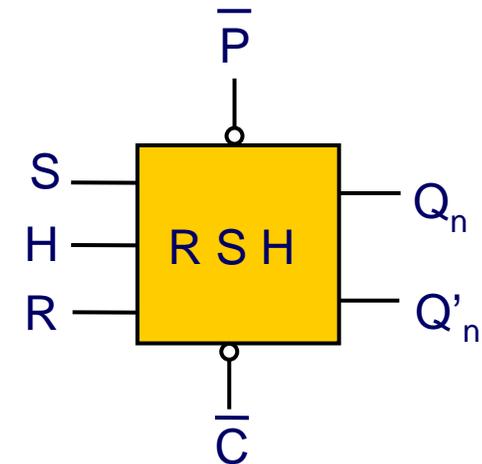
$$\overline{C} = 0 ; \overline{P} = 1 \Rightarrow Q_{n+1} = 0$$

$$\overline{P} = 0 ; \overline{C} = 1 \Rightarrow Q_{n+1} = 1$$



H	R	S	\overline{P}	\overline{C}	Q_{n+1}	Q'_{n+1}
1	0	0	1	1	Q_n	Q_n
1	0	1	1	1	1	0
1	1	0	1	1	0	1
1	1	1	1	1	1 *	1 *
X	X	X	0	1	1	0
X	X	X	1	0	0	1
X	X	X	0	0	1 *	1 *
0	X	X	1	1	Q_n	Q_n

Mémoire
Mise à 1
Mise à 0
Ambiguë
Forçage à 1
Forçage à 0
Interdit
Mémoire



II – Les types de bascules

II.3 – La bascule D Latch (verrou) :

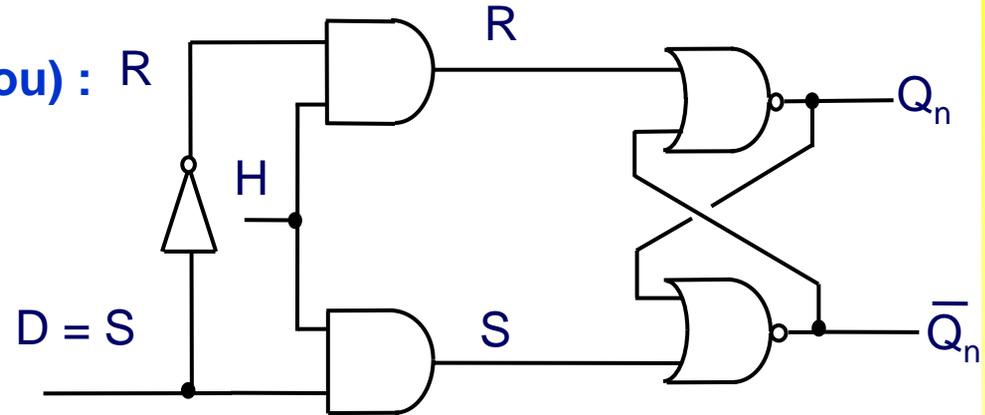
Bascule RSH avec $D = S = \overline{R}$

H	D	Q_{n+1}
0	X	Q_n
1	0	0
1	1	1

Mémoire

Mise à 0

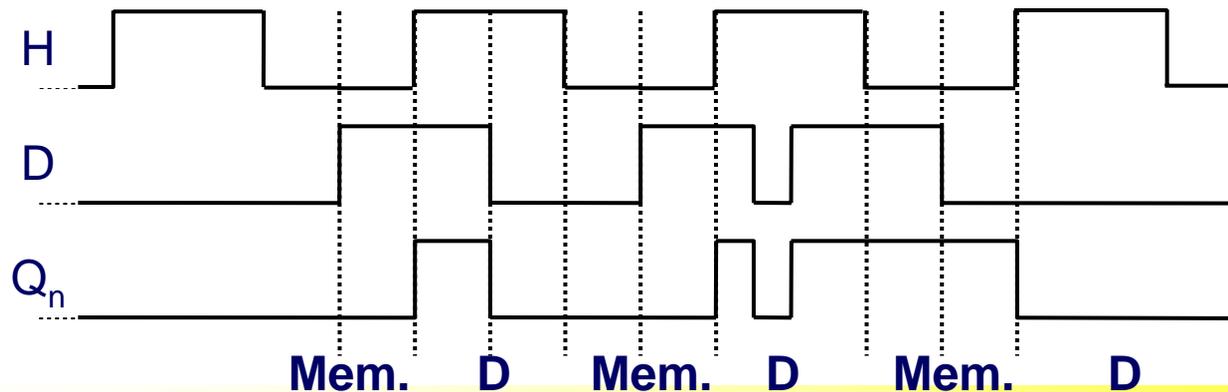
Mise à 1



$H = 1 \Rightarrow Q$ recopie D.
 $H = 0 \Rightarrow$ Mémoire.



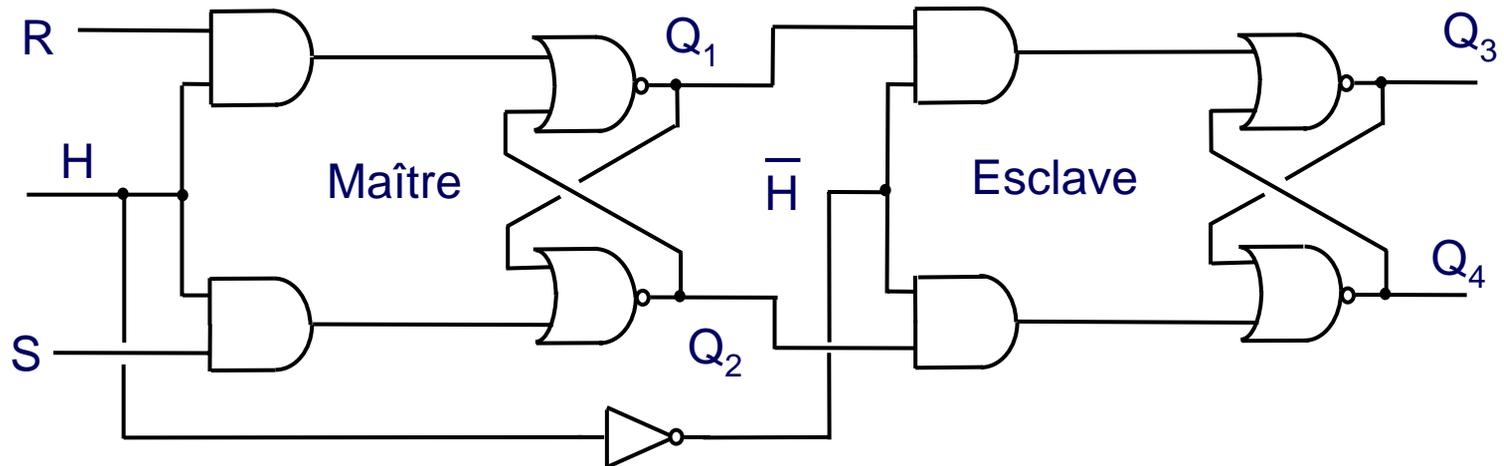
Chronogramme :



II – Les types de bascules

II.4 – La bascule RSH Maître Esclave :

Les bascules peuvent être mises en cascades afin de réaliser des compteurs ou des registres à décalages (exemple de la division par 2).



$H = 1 \Rightarrow$ Le maître est transparent : Les entrées R et S imposent les valeurs sur Q1 et Q2 (acquisition).

$\overline{H} = 0 \Rightarrow$ L'esclave est verrouillé (mémoire).

$H = 0 \Rightarrow$ Le maître est verrouillé (mémoire).

$\overline{H} = 1 \Rightarrow$ L'esclave recopie les sorties du maître (acquisition).

II – Les types de bascules

Exemple : $R = 1$ et $S = 0$.

$H = 1 \Rightarrow$ Maître transparent et esclave verrouillé.

$\Rightarrow Q_1 = 0; Q_2 = 1.$

$\Rightarrow Q_3$ et Q_4 inchangées.

$H = 0 \Rightarrow$ Maître verrouillé et esclave transparent.

$\Rightarrow Q_1 = 0; Q_2 = 1$ (inactives).

$\Rightarrow Q_3 = 1; Q_4 = 0.$

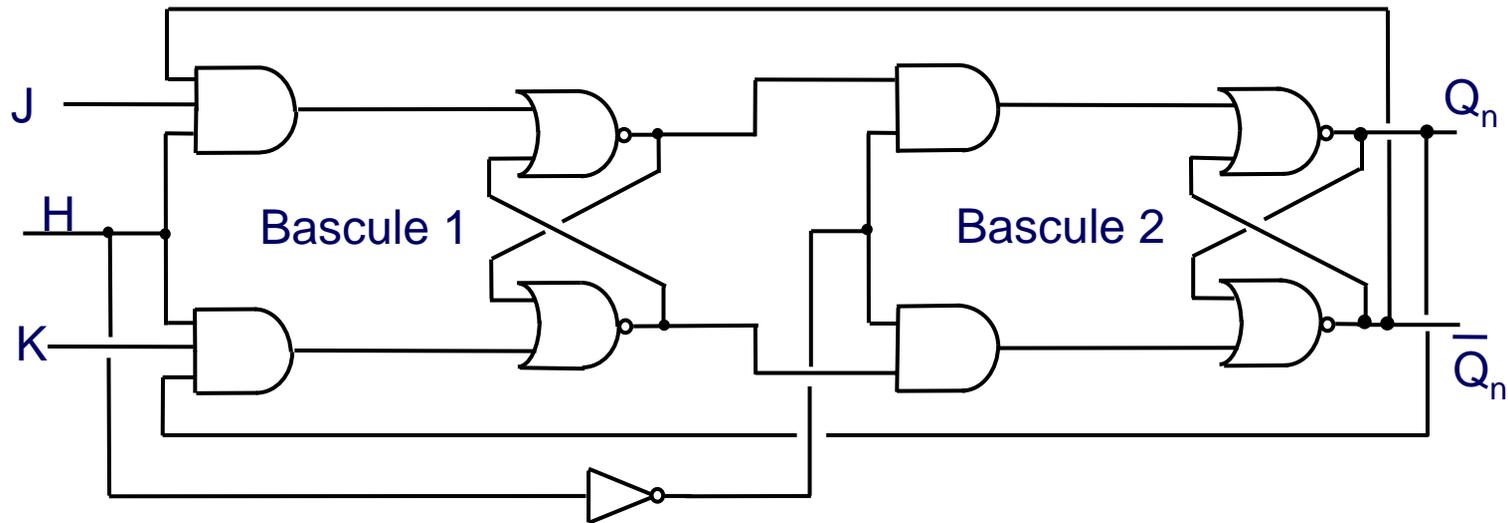
Le maître est sensible tant que $H = 1$ (des variations de R et S entraînent des variations sur Q_1 et Q_2). Par contre le changement d'état sur les sorties Q_3 et Q_4 a lieu sur le front descendant de H (dès que $H = 0$).

La structure de la bascule RSH Maître Esclave en portes NAND existe également et donne les mêmes résultats.

II – Les types de bascules

II.5 – La bascule JK :

C'est une bascule Maître Esclave avec deux entrées supplémentaires et un rebouclage des sorties sur les entrées.



- $J = 0$ et $K = 0 \Rightarrow Q_n$ et $\overline{Q_n}$ conservées à la prochaine impulsion de H.
- $J = 0$ et $K = 1 \Rightarrow Q_{n+1} = 0$ et $\overline{Q_{n+1}} = 1$ à la prochaine impulsion de H.
- $J = 1$ et $K = 0 \Rightarrow Q_{n+1} = 1$ et $\overline{Q_{n+1}} = 0$ à la prochaine impulsion de H.
- $J = 1$ et $K = 1 \Rightarrow Q_n$ et $\overline{Q_n}$ changent d'état à la prochaine impulsion de H.

II – Les types de bascules

J	K	H	Q_{n+1}	Q'_{n+1}
0	0	1	Q_n	$\overline{Q_n}$
0	1	1	0	1
1	0	1	1	0
1	1	1	$\overline{Q_n}$	Q_n
X	X	0	Q_n	$\overline{Q_n}$

Mémoire

Mise à 0

Mise à 1

Inversion (Toggle)

Mémoire

La structure à base de bascules Maître Esclave en portes NAND existe également et donne les mêmes résultats.

II.6 – Les bascules déclenchables :

Contrairement à la bascule JK Maître Esclave (sensible sur la durée d'un état de H), la bascule est sensible aux variations des entrées uniquement au moment du front d'horloge (montant ou descendant).

Ce principe permet d'éviter les perturbations et les effets des variations imprévues.

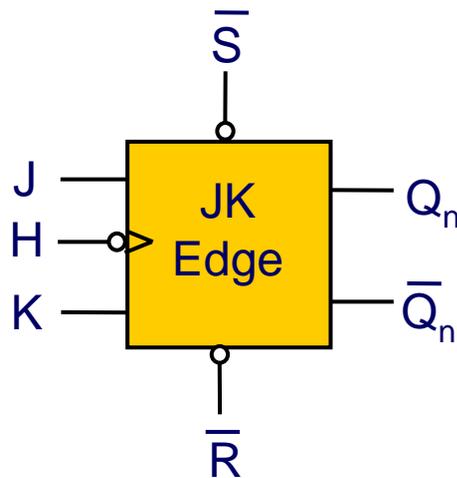
II – Les types de bascules

La bascule JK Edge Triggered :

J, K : Entrées synchrones.

H : Horloge active sur front descendant.

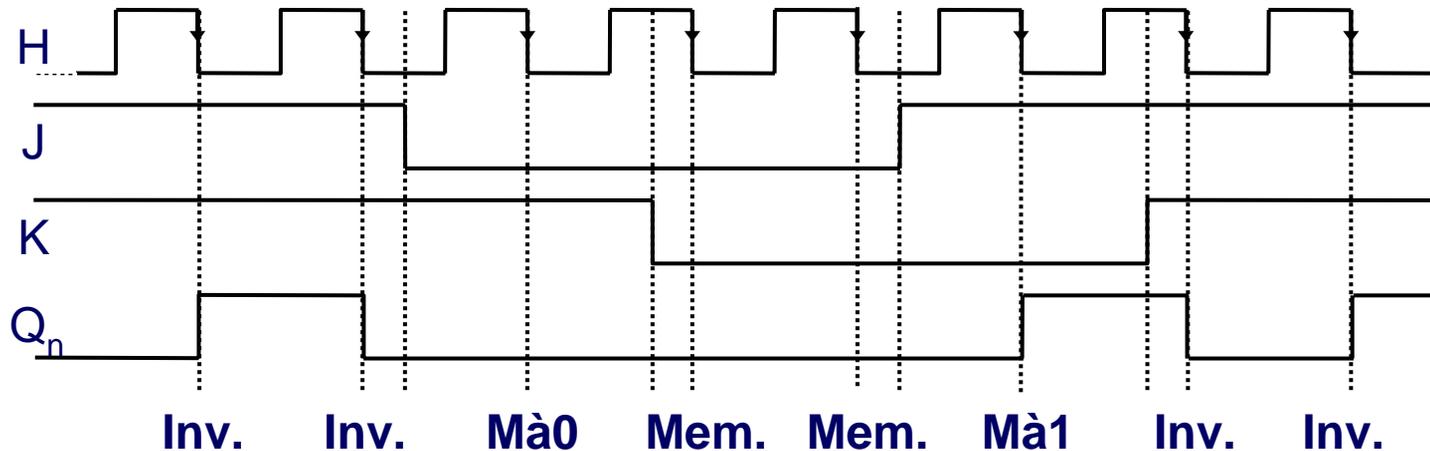
\overline{R} , \overline{S} : Entrées de forçage asynchrones actives à 0.



\overline{S}	\overline{R}	J	K	H	Q_{n+1}	Q'_{n+1}	
1	1	X	X	0	Q_n	$\overline{Q_n}$	Mémoire
1	1	X	X	1	Q_n	$\overline{Q_n}$	Mémoire
1	1	0	0	$\overline{\downarrow}$	Q_n	$\overline{Q_n}$	Mémoire
1	1	0	1	$\overline{\downarrow}$	0	1	Mise à 0
1	1	1	0	$\overline{\downarrow}$	1	0	Mise à 1
1	1	1	1	$\overline{\downarrow}$	$\overline{Q_n}$	Q_n	Inversion
0	1	X	X	X	1	0	Forçage à 1
1	0	X	X	X	0	1	Forçage à 0
0	0	X	X	X	*	*	Interdit

II – Les types de bascules

Chronogramme :



On ne tient compte des états de J et K que sur le front descendant de l'horloge.

On peut facilement remplacer une bascule JK en une bascule D en imposant :

$$D = J = \overline{K}$$

$J = 0 ; K = 1 \Rightarrow Q_{n+1} = 0$ et $Q'_{n+1} = 1$ au prochain front de H.
 $J = 1 ; K = 0 \Rightarrow Q_{n+1} = 1$ et $Q'_{n+1} = 0$ au prochaine front de H.

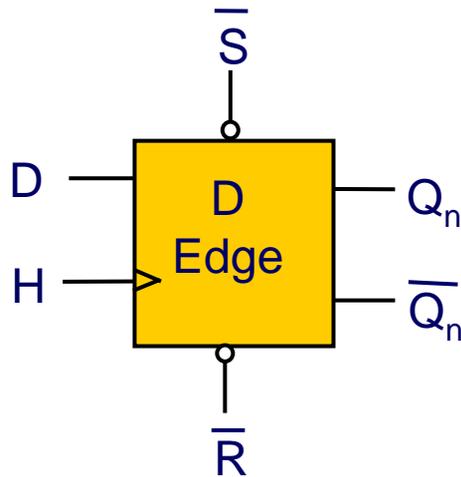
II – Les types de bascules

La bascule D Edge Triggered :

D : Entrée synchrone.

H : Horloge active sur front montant.

\bar{R} , \bar{S} : Entrées de forçage asynchrones actives à 0.

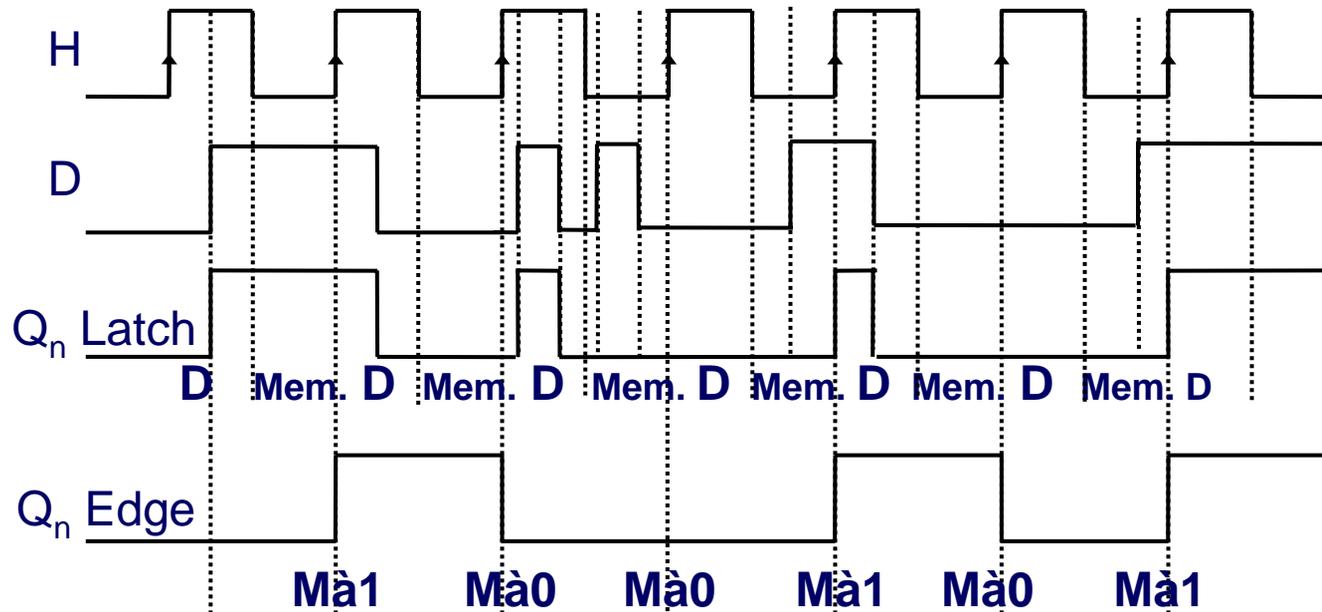


Q recopie D sur le front montant de H (mémoire sinon).

\bar{S}	\bar{R}	D	H	Q_{n+1}	Q'_{n+1}	
1	1	X	0	Q_n	$\overline{Q_n}$	Mémoire
1	1	X	1	Q_n	$\overline{Q_n}$	Mémoire
1	1	0	\uparrow	0	1	Mise à 0
1	1	1	\uparrow	1	0	Mise à 1
0	1	X	X	1	0	Forçage à 1
1	0	X	X	0	1	Forçage à 0
0	0	X	X	*	*	Interdit

II – Les types de bascules

Chronogramme :



Q recopie D sur le front montant de H \neq Q recopie D quand H = 1.

III – Les compteurs et décompteurs

III.1 – Définitions :

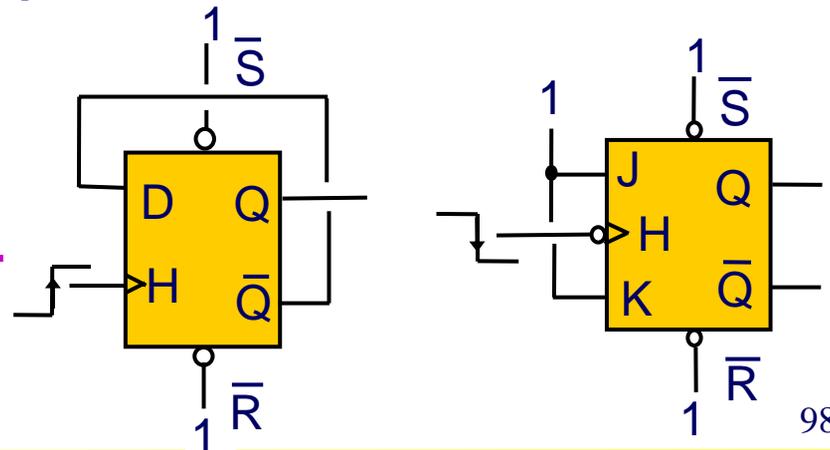
Un compteur par N (ou modulo N) est un système séquentiel possédant N états stables et pouvant passer de l'un à l'autre sous l'influence d'une impulsion.

Un compteur synchrone est un compteur pour lequel tous les basculements sont simultanés.

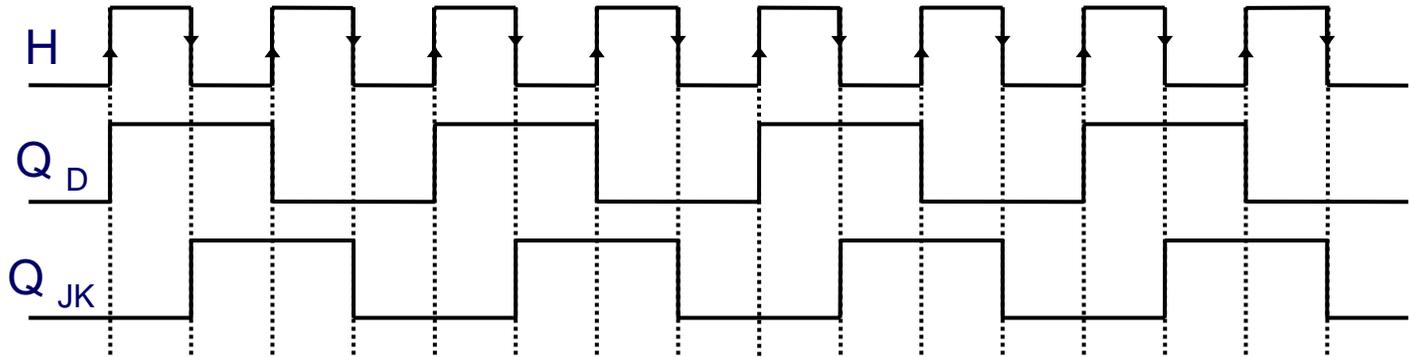
Un compteur asynchrone est un compteur pour lequel les étages basculent successivement les uns après les autres.

Plus petit compteur (bistable) :

1 bistable \Rightarrow Compteur modulo 2.
N bistables \Rightarrow Compteur modulo 2^N .



III – Les compteurs et décompteurs



La période des signaux est deux fois plus élevée que l'horloge. Ce montage est appelé diviseur par 2 (de fréquence) ou compteur par 2.

III.2 – Les compteurs asynchrones (à propagation) :

Dans un compteur asynchrone, chaque bascule est montée en diviseur par 2 (bistable).

Chaque bascule fait commuter la bascule suivante.

III – Les compteurs et décompteurs

Q _D	Q _C	Q _B	Q _A	imp.
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15
0	0	0	0	16

Compteur binaire asynchrone modulo 16 :

4 bascules ⇒ 16 états.

Q_A passe de 1 à 0 ⇒ Q_B change d'état.

Q_B passe de 1 à 0 ⇒ Q_C change d'état.

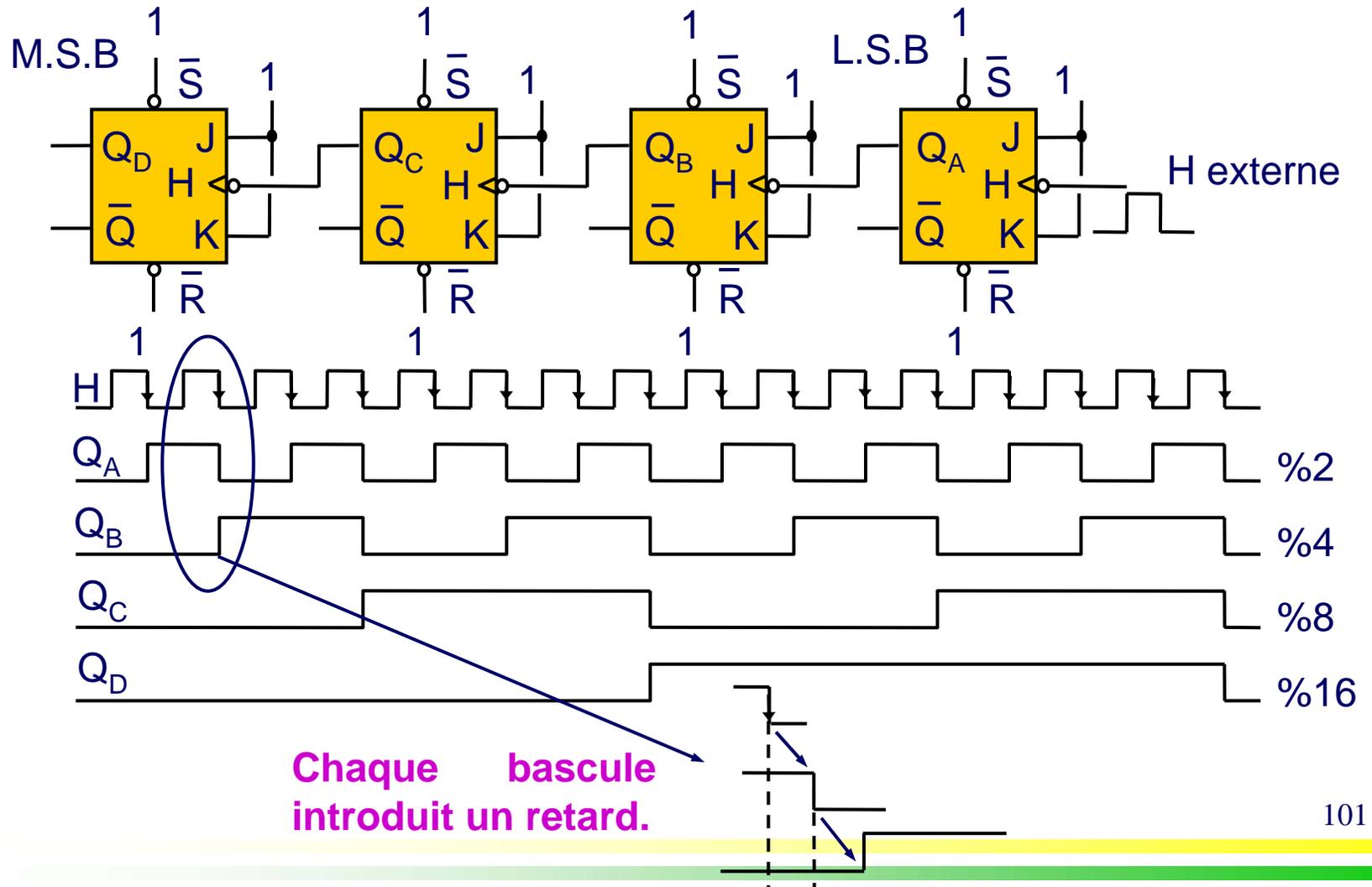
Q_C passe de 1 à 0 ⇒ Q_D change d'état.

Toutes les bascules sont montées en diviseur par 2.

⇐ **Recyclage**

III – Les compteurs et décompteurs

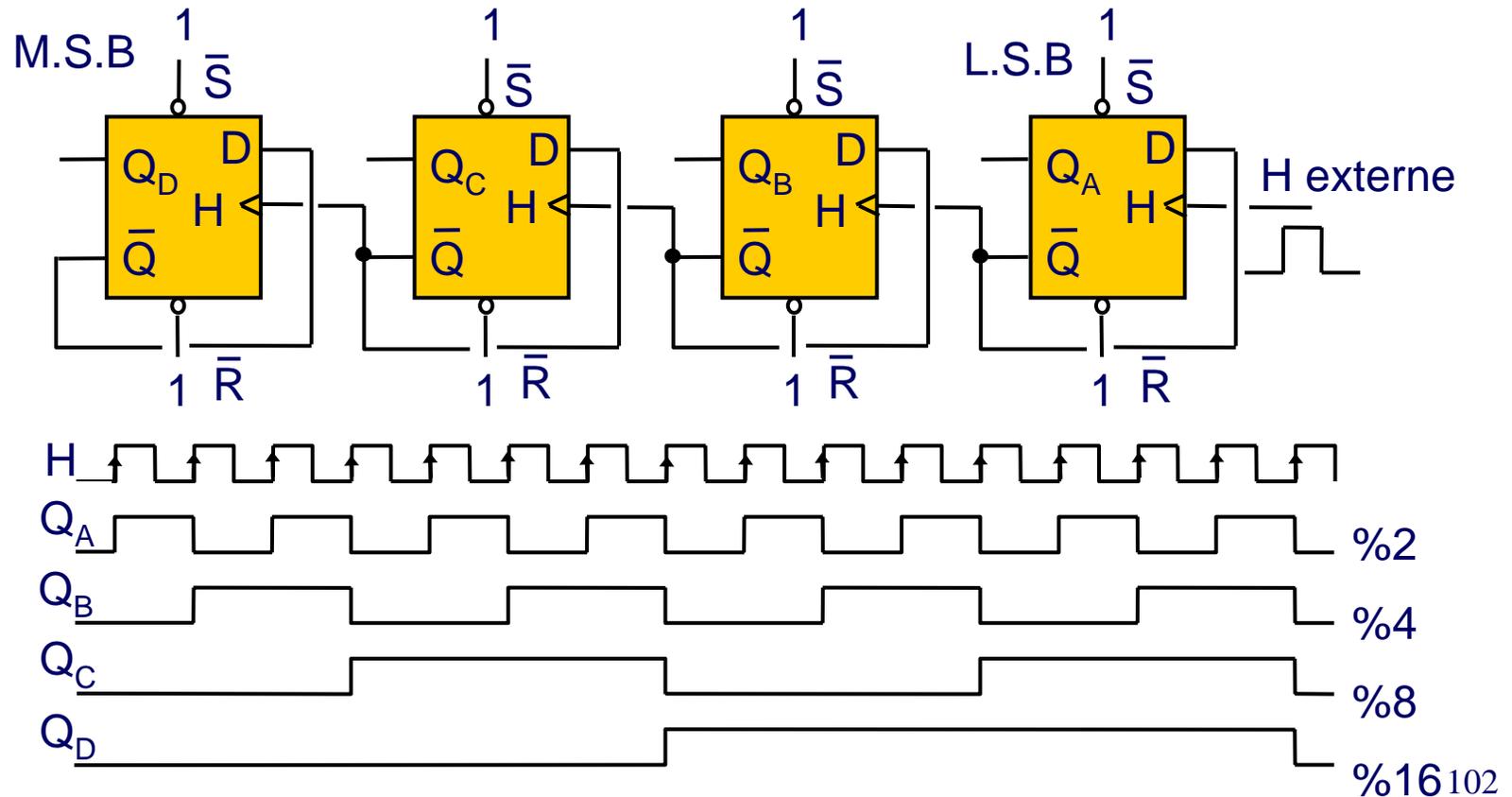
Structure en bascules JK actives sur front descendant :



III – Les compteurs et décompteurs

Structure en bascules D actives sur front montant :

On récupère le front montant actif pour les bascules sur les sorties complémentées.



III – Les compteurs et décompteurs

Compteur binaire asynchrone :

- Bascules montées en diviseur par 2.
- Front descendant actif $\Rightarrow Q$ reliée à H (bascule suivante).
- Front montant actif $\Rightarrow \overline{Q}$ reliée à H (bascule suivante).
- Comptage sur les sorties Q.

Compteur à cycle incomplet :

Il suffit de réaliser un compteur à cycle complet (2^N) et d'actionner les RAZ asynchrones.

Exemple : Modulo 6 (0 à 5).

On détecte l'état 6 et on actionne les entrées R.

Q_C	Q_B	Q_A	imp.
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7
0	0	0	8

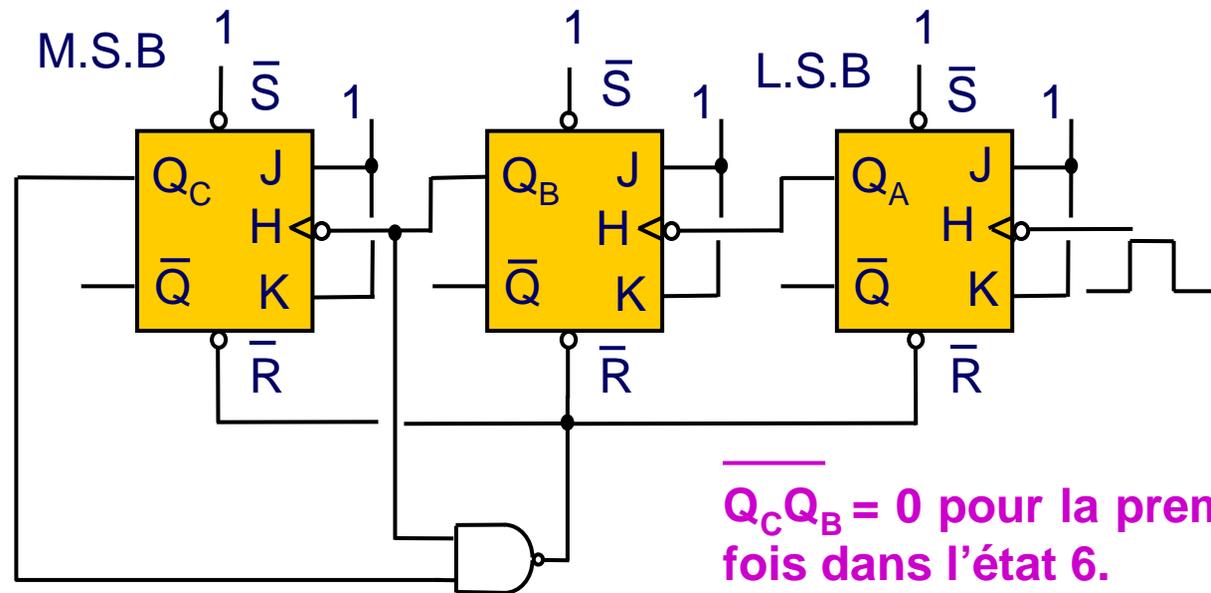
\Leftarrow Remise à 0

\Leftarrow Recyclage

III – Les compteurs et décompteurs

État 6 : $Q_C Q_B Q_A = 110 \Rightarrow$ On envoie $\overline{Q_C} \overline{Q_B} \overline{Q_A}$ vers les entrées \overline{R} .

Ou : Première fois que $Q_C = Q_B = 1 \Rightarrow \overline{Q_C} \overline{Q_B}$ vers les entrées \overline{R} .



Pour un compteur à décade (0 à 9) \Rightarrow 10 états stables \Rightarrow Remise à zéro dans l'état 10 (1010).

Attention si les entrées de Remise à zéro sont actives à l'état haut !

III – Les compteurs et décompteurs

Décompteur binaire asynchrone :

Q_C	Q_B	Q_A	imp.
1	1	1	0
1	1	0	1
1	0	1	2
1	0	0	3
0	1	1	4
0	1	0	5
0	0	1	6
0	0	0	7
1	1	1	8

Décompteur binaire asynchrone modulo 8.

3 bascules \Rightarrow 8 états.

Q_A passe de 0 à 1 $\Rightarrow Q_B$ change d'état.

Q_B passe de 0 à 1 $\Rightarrow Q_C$ change d'état.

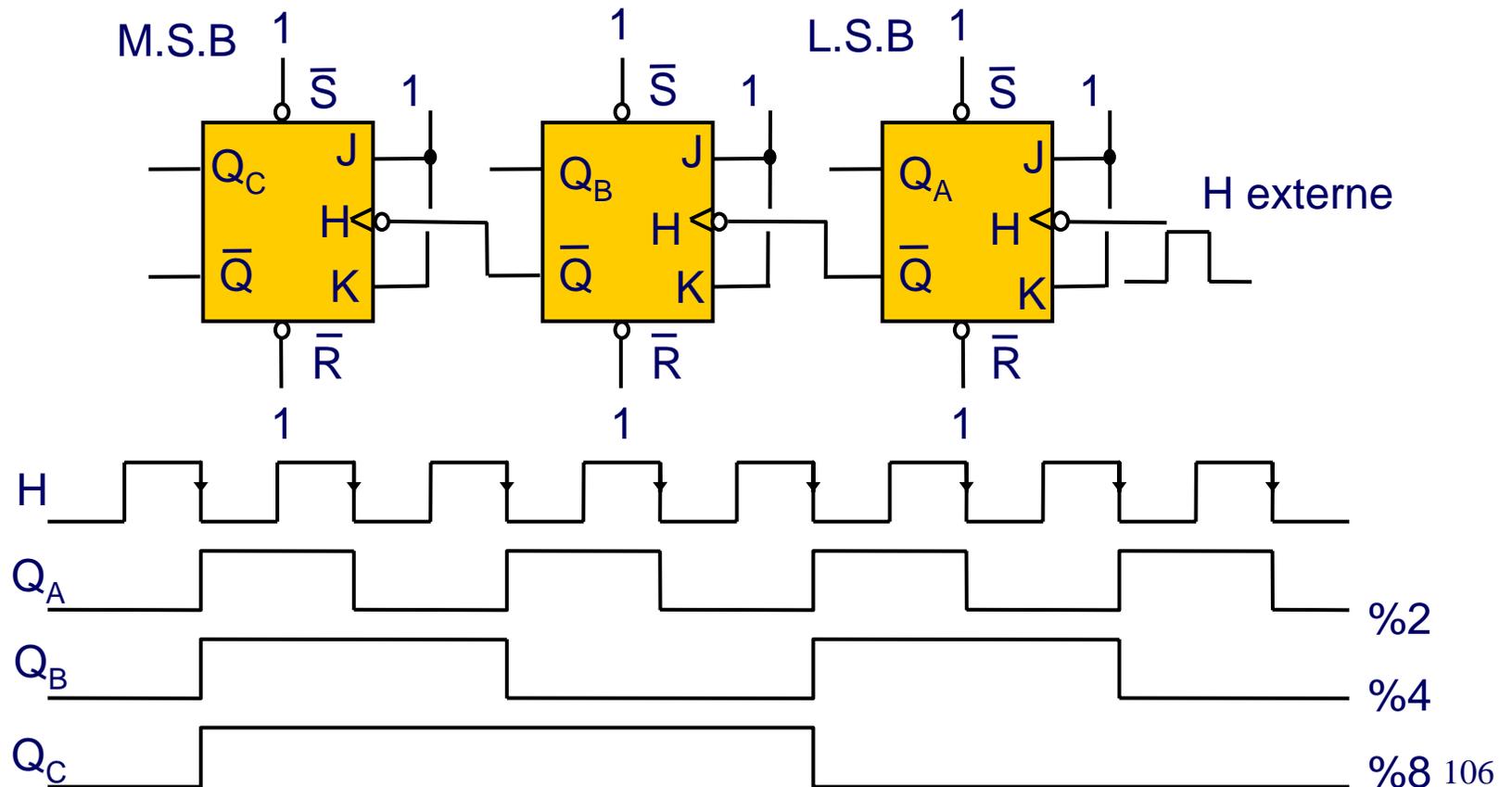
Toutes les bascules sont montées en diviseur par 2.

\Leftrightarrow Recyclage

III – Les compteurs et décompteurs

Structure en bascules JK actives sur front descendant :

On récupère le front descendant actif pour les bascules sur les sorties complémentées.



III – Les compteurs et décompteurs

Décompteur binaire asynchrone :

- Bascules montées en diviseur par 2.
- Front descendant actif $\Rightarrow \overline{Q}$ reliée à H (bascule suivante).
- Front montant actif $\Rightarrow Q$ reliée à H (bascule suivante).
- Décomptage sur les sorties Q.

Fréquence maximale de travail :

Chaque bascule introduit un retard t_B lors de sa commutation (le temps de propagation d'une bascule est de l'ordre de quelques dizaines de ns).

La période T_H de l'horloge doit donc être telle que :

$$T_H > N t_B \Rightarrow F_H < 1 / (N t_B) \quad \text{pour } N \text{ bascules.}$$

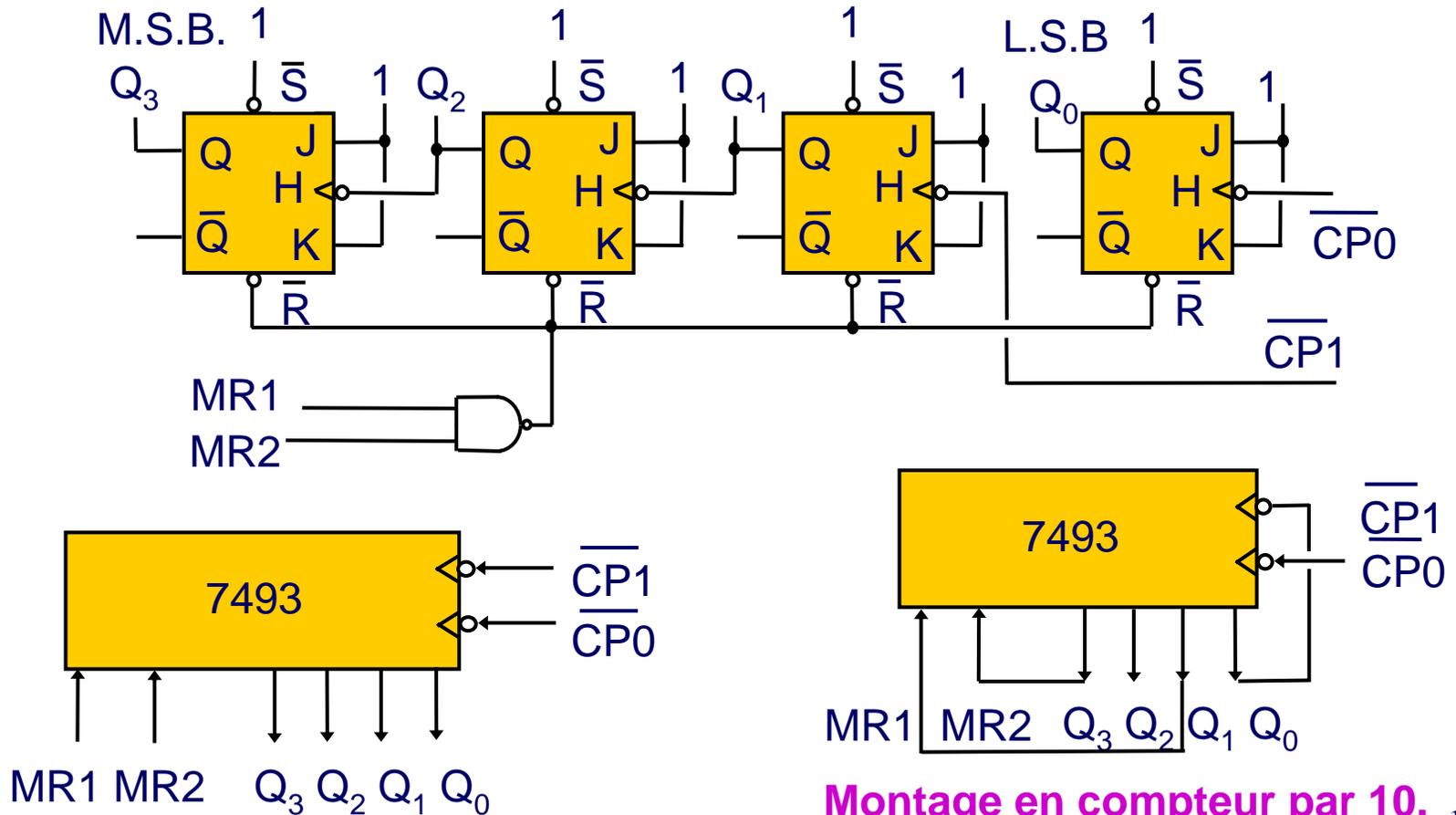
$$t_B = 24 \text{ ns} ; N = 4 \Rightarrow F_{\text{max}} = 10,4 \text{ MHz.}$$

$$t_B = 24 \text{ ns} ; N = 6 \Rightarrow F_{\text{max}} = 6,9 \text{ MHz.}$$

III – Les compteurs et décompteurs

Le circuit intégré 7493 :

Compteur asynchrone 4 bits avec remise à zéro.



Montage en compteur par 10.

III – Les compteurs et décompteurs

III.3 – Les compteurs synchrones (parallèle)

Dans un compteur synchrone, toutes les bascules commutent simultanément. Il faut donc anticiper les changements d'horloge.

Compteur binaire synchrone modulo 16 :

4 bascules \Rightarrow 16 états.

Q_A diviseur par 2.

Q_B conserve son état sauf quand $Q_A = 1$ (au prochain front, Q_B changera d'état).

Q_C conserve son état sauf quand $Q_A = Q_B = 1$ (au prochain front, Q_C changera d'état).
Recyclage \Rightarrow

Q_D	Q_C	Q_B	Q_A	imp.
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15
0	0	0	0	16 ⁰⁹

III – Les compteurs et décompteurs

Structure en bascules JK :

Le fait que la bascule soit active sur un front montant ou descendant n'intervient plus dans le raisonnement.

Toutes les bascules reçoivent le même signal d'horloge.

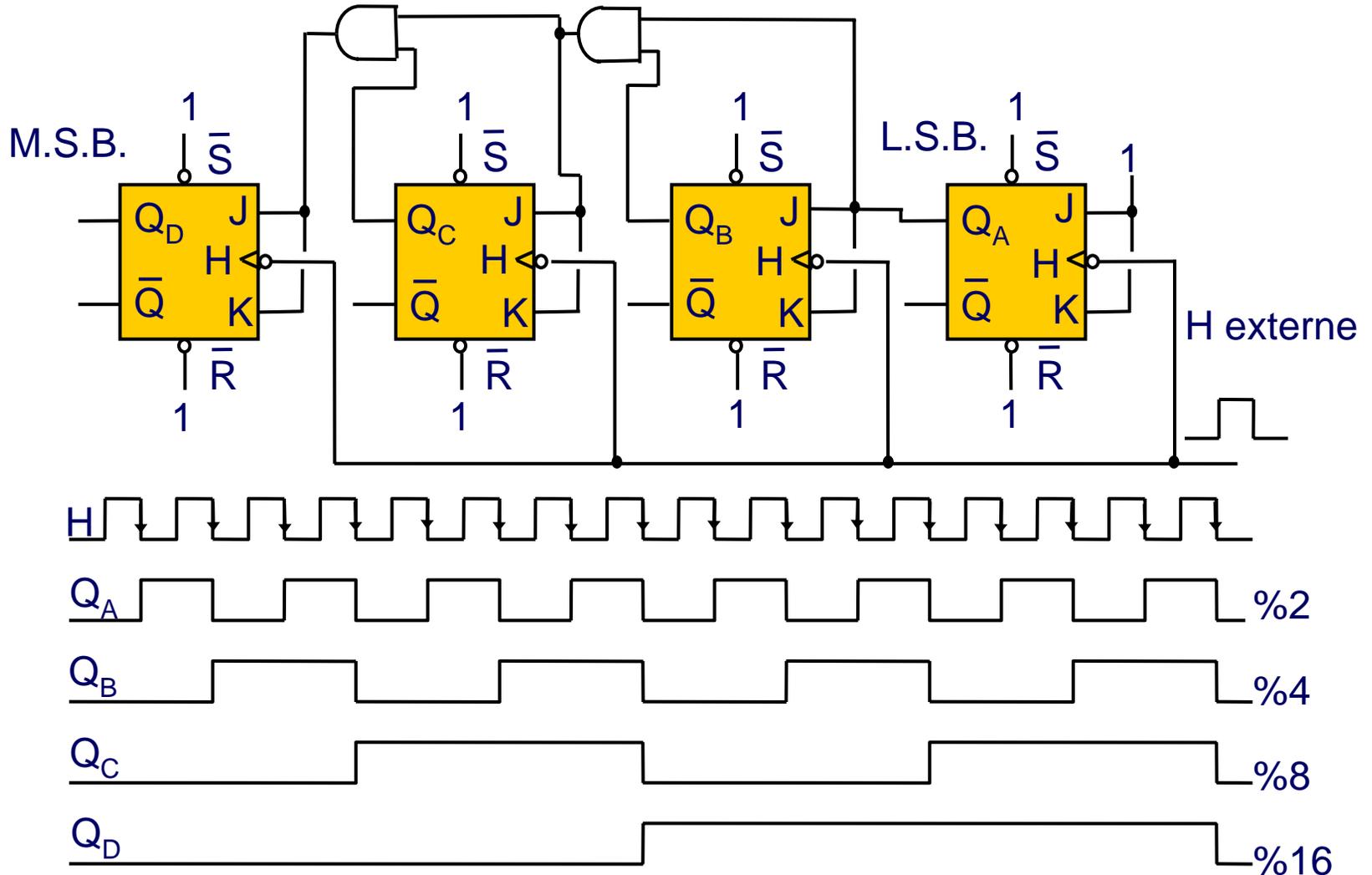
$J = K = 0 \Rightarrow$ La bascule conserve son état.

$J = K = 1 \Rightarrow$ La bascule s'inverse au prochain front.

On utilise des portes ET afin de préparer les conditions de basculement avant le front d'horloge.

Cette méthode ne permet pas la synthèse à l'aide de bascules D.

III – Les compteurs et décompteurs



Un seul retard t_B introduit quelque soit le nombre de bascules.

III – Les compteurs et décompteurs

Décompteur binaire synchrone modulo 16 :

4 bascules \Rightarrow 16 états.

Q_A diviseur par 2.

Q_B conserve son état sauf quand $Q_A = 0$ ($\overline{Q_A} = 1$) (au prochain front, Q_B changera d'état).

Q_C conserve son état sauf quand $Q_A = Q_B = 0$ ($\overline{Q_A} = \overline{Q_B} = 1$) (au prochain front, Q_C changera d'état).

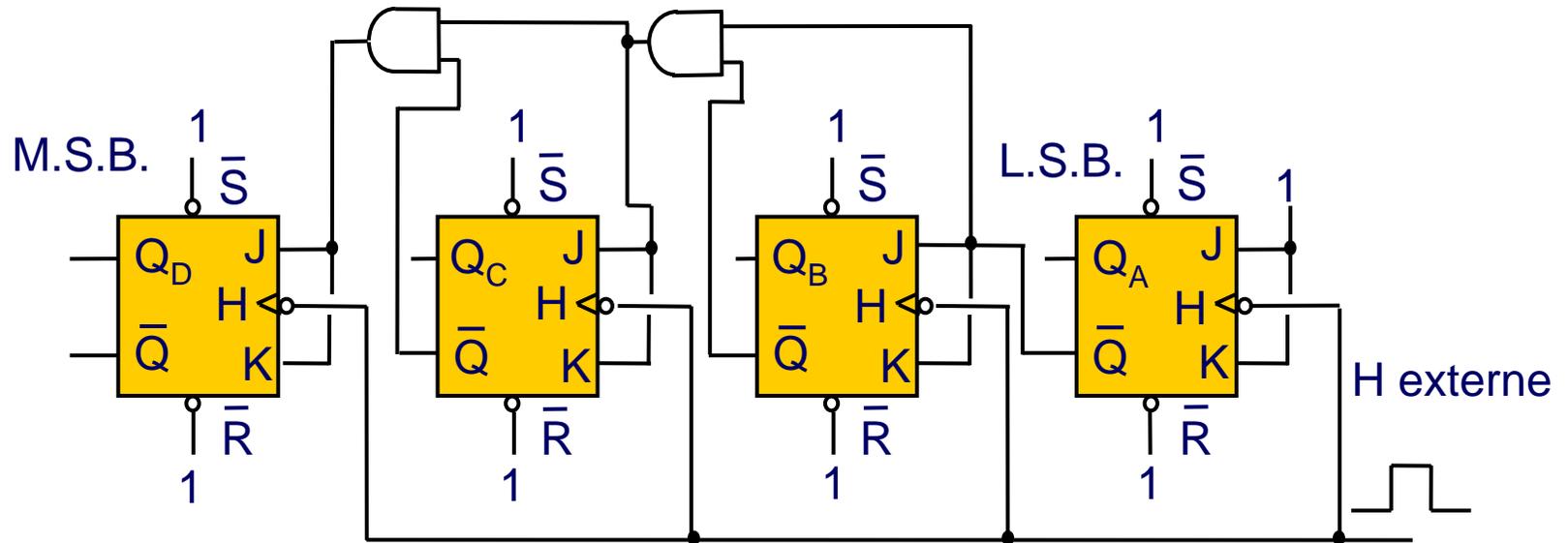
Idem pour Q_D .

Q_D	Q_C	Q_B	Q_A	imp.
1	1	1	1	0
1	1	1	0	1
1	1	0	1	2
1	1	0	0	3
1	0	1	1	4
1	0	1	0	5
1	0	0	1	6
1	0	0	0	7
0	1	1	1	8
0	1	1	0	9
0	1	0	1	10
0	1	0	0	11
0	0	1	1	12
0	0	1	0	13
0	0	0	1	14
0	0	0	0	15
1	1	1	1	16 ¹²

Recyclage \Rightarrow

III – Les compteurs et décompteurs

Structure en bascules JK :



Compteur binaire synchrone :

- Q_A, Q_B, Q_C, Q_D et leurs combinaisons en portes ET sur les entrées J et K des bascules (comptage sur les sorties Q).

Décompteur binaire synchrone :

- $\bar{Q}_A, \bar{Q}_B, \bar{Q}_C, \bar{Q}_D$ et leurs combinaisons en portes ET sur les entrées J et K des bascules (comptage sur les sorties Q).

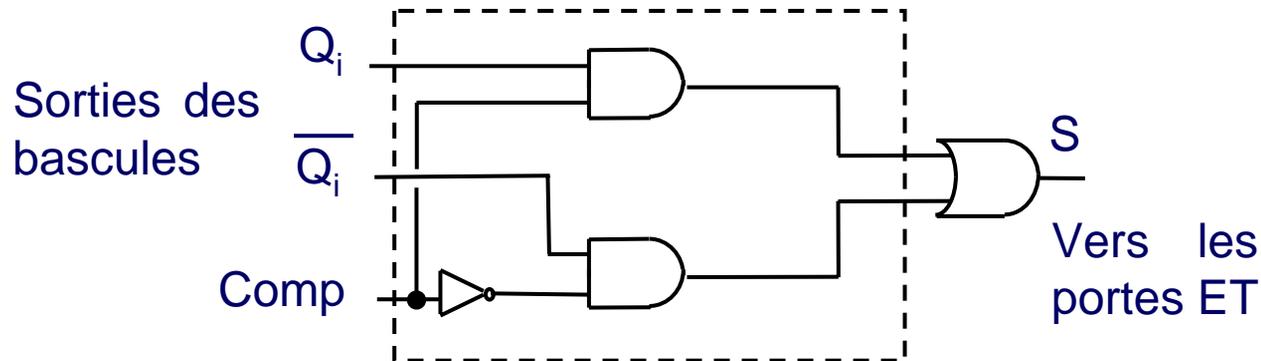
III – Les compteurs et décompteurs

Structure d'un compteur / décompteur binaire synchrone :

On ajoute une variable supplémentaire de comptage :

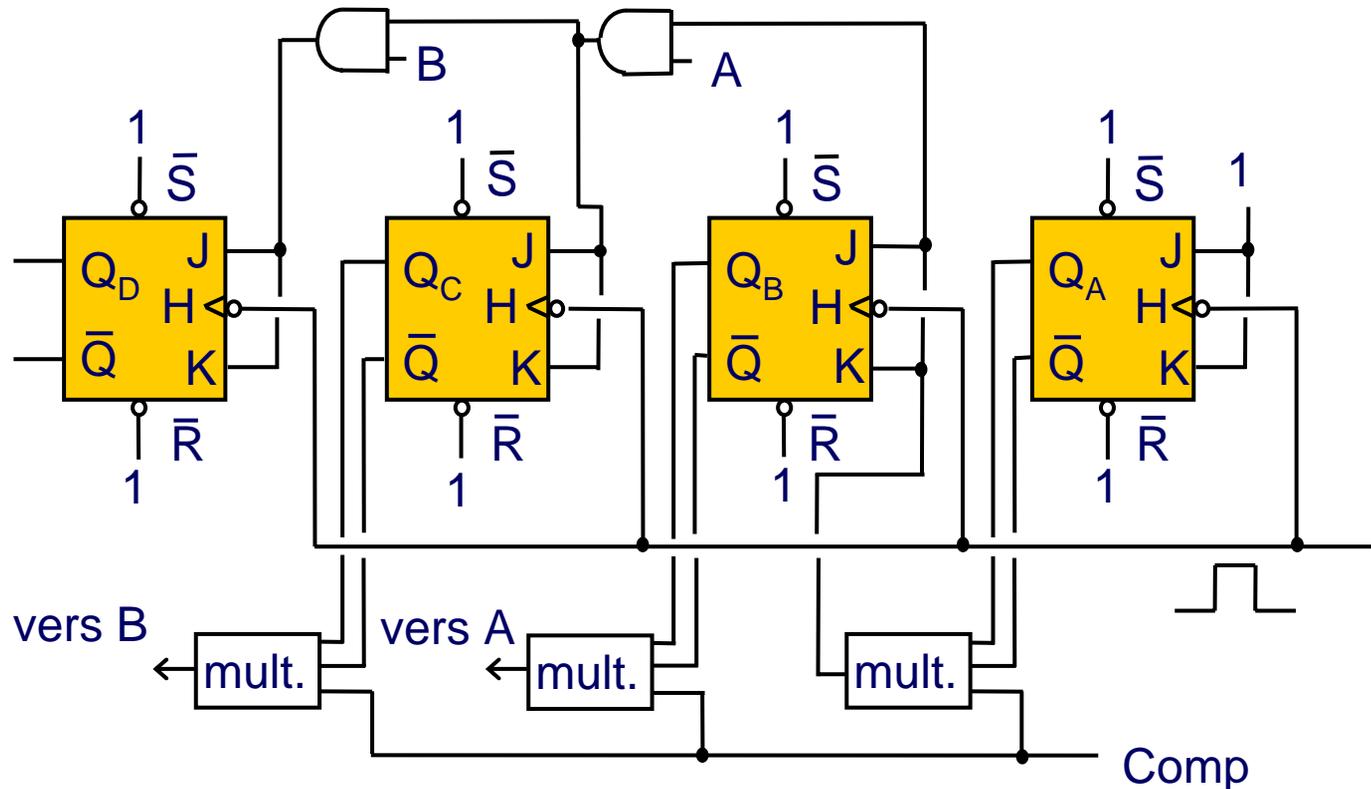
- $\text{Comp} = 1 \Rightarrow$ Comptage \Rightarrow Envoie des Q_i vers les portes ET.
- $\text{Comp} = 0 \Rightarrow$ Décomptage \Rightarrow Envoie des \overline{Q}_i vers les portes ET.

Le circuit de sélection est un multiplexeur 2 vers 1 :



III – Les compteurs et décompteurs

Compteur / décompteur binaire synchrone :



Seule l'action sur la variable **Comp** permet le passage du comptage au décomptage.

III – Les compteurs et décompteurs

Compteur pré-réglable :

$\overline{PL} = 1 \Rightarrow$ Fonctionnement synchrone.

$\overline{PL} = 0 \Rightarrow$ Chargement asynchrone (prioritaire) d'une valeur.

Chargement :

$P_c = 1$

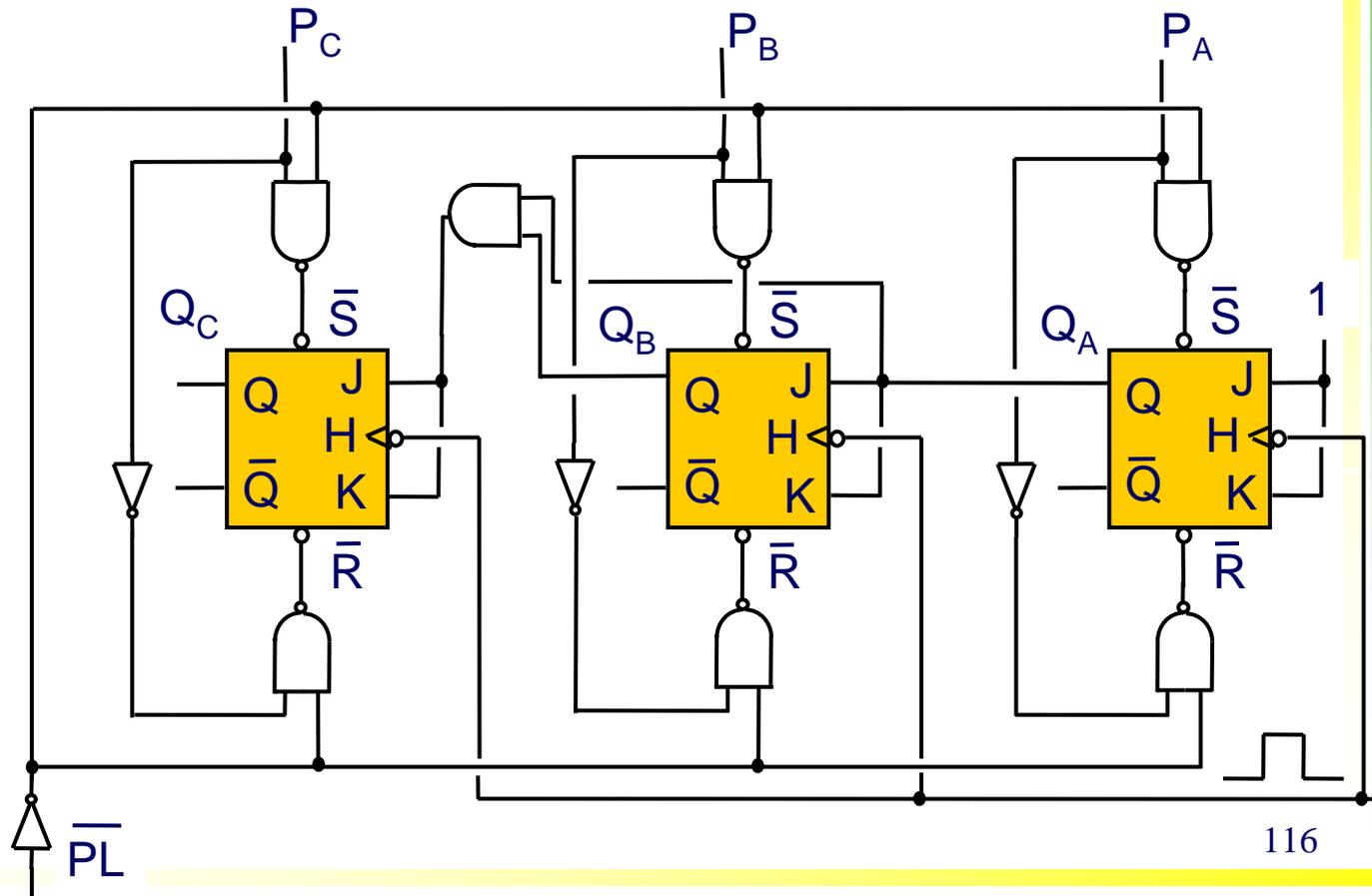
$\Rightarrow \overline{S} = 0, \overline{R} = 1$

$\Rightarrow Q_c = 1, \overline{Q}_c = 0$

$P_c = 0$

$\Rightarrow \overline{S} = 1, \overline{R} = 0$

$\Rightarrow Q_c = 0, \overline{Q}_c = 1$



III – Les compteurs et décompteurs

Le circuit intégré 74193 :

Compteur synchrone 4 bits avec remise à zéro et chargement parallèle.

CPU et CPD : Horloge de comptage et décomptage.

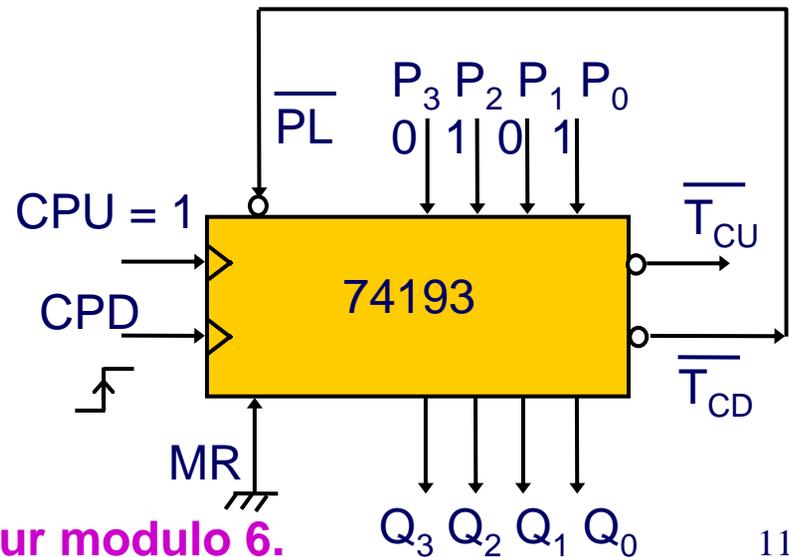
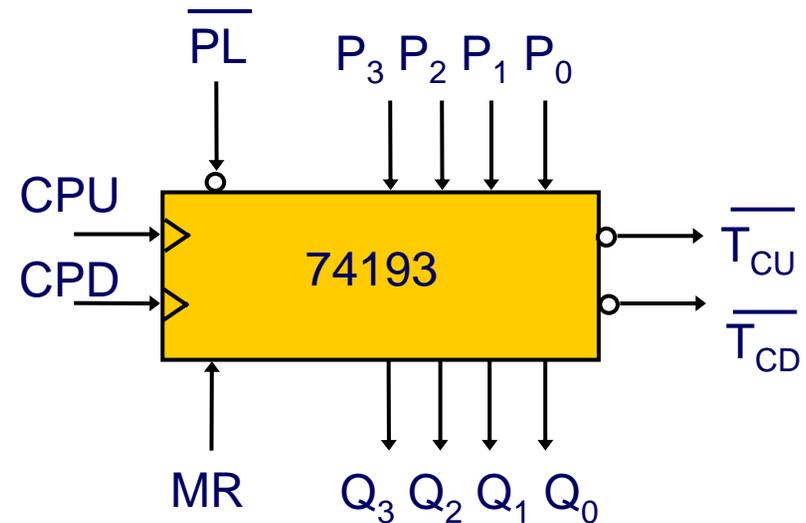
MR : Réinitialisation (MR = 1 \Rightarrow compteur bloqué à 0000).

$Q_3 \dots Q_0$: Sorties du compteur.

$P_3 \dots P_0$: Entrées parallèles.

\overline{T}_{CU} \overline{T}_{CD} : Retenues de comptage et de décomptage.

PL : Chargement parallèle asynchrone.



Montage en décompteur modulo 6.

III – Les compteurs et décompteurs

Méthode de synthèse d'un compteur synchrone :

Exemple : Compteur binaire modulo 6 (3 bascules).

Réalisation en bascules JK avec $J = K$.

Au prochain front :

$J = K = 0 \Rightarrow$ Mémoire.

$J = K = 1 \Rightarrow$ Changement d'état.

Remise à 0 synchrone

Q_C	Q_B	Q_A	J_C	J_B	J_A
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	1	0	1

But : déterminer les états de J et K à placer avant le front d'horloge \Rightarrow Problème combinatoire (exprimer les entrées J et K en fonction des sorties).

III – Les compteurs et décompteurs

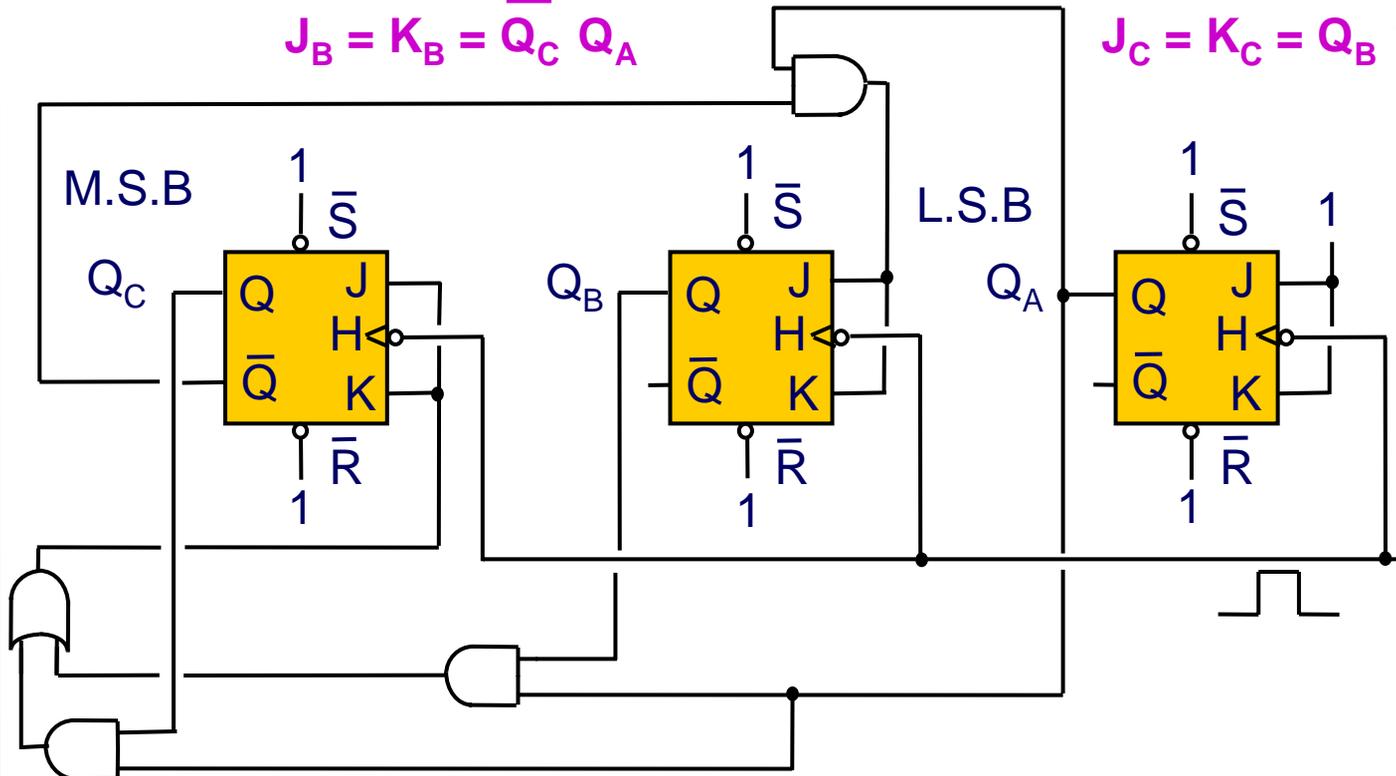
$Q_C \setminus Q_B Q_A$	00	01	11	10
0	0	1	1	0
1	0	0	X	X

$Q_C \setminus Q_B Q_A$	00	01	11	10
0	0	0	1	0
1	0	1	X	X

$$J_B = K_B = \overline{Q_C} Q_A$$

$$J_C = K_C = Q_B Q_A + Q_C Q_A$$

$$J_A = K_A = 1$$



III – Les compteurs et décompteurs

Réalisation en bascules JK avec J et K quelconques.

Au prochain front :

$J = 1, K = 1 \Rightarrow$ Inversion.

$J = 0, K = 0 \Rightarrow$ Mémoire.

$J = 0, K = 1 \Rightarrow$ Mise à 0.

$J = 1, K = 0 \Rightarrow$ Mise à 1.

Remise à 0
synchrone

Q_C	Q_B	Q_A	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	X	0	X	1	X
0	0	1	0	X	1	X	X	1
0	1	0	0	X	X	0	1	X
0	1	1	1	X	X	1	X	1
1	0	0	X	0	0	X	1	X
1	0	1	X	1	0	X	X	1

III – Les compteurs et décompteurs

$Q_C \setminus Q_B Q_A$	00	01	11	10
0	0	1	X	X
1	0	0	X	X

$$J_B = \overline{Q_C} Q_A$$

$Q_C \setminus Q_B Q_A$	00	01	11	10
0	X	X	1	0
1	X	X	X	X

$$K_B = Q_A$$

$Q_C \setminus Q_B Q_A$	00	01	11	10
0	0	0	1	0
1	X	X	X	X

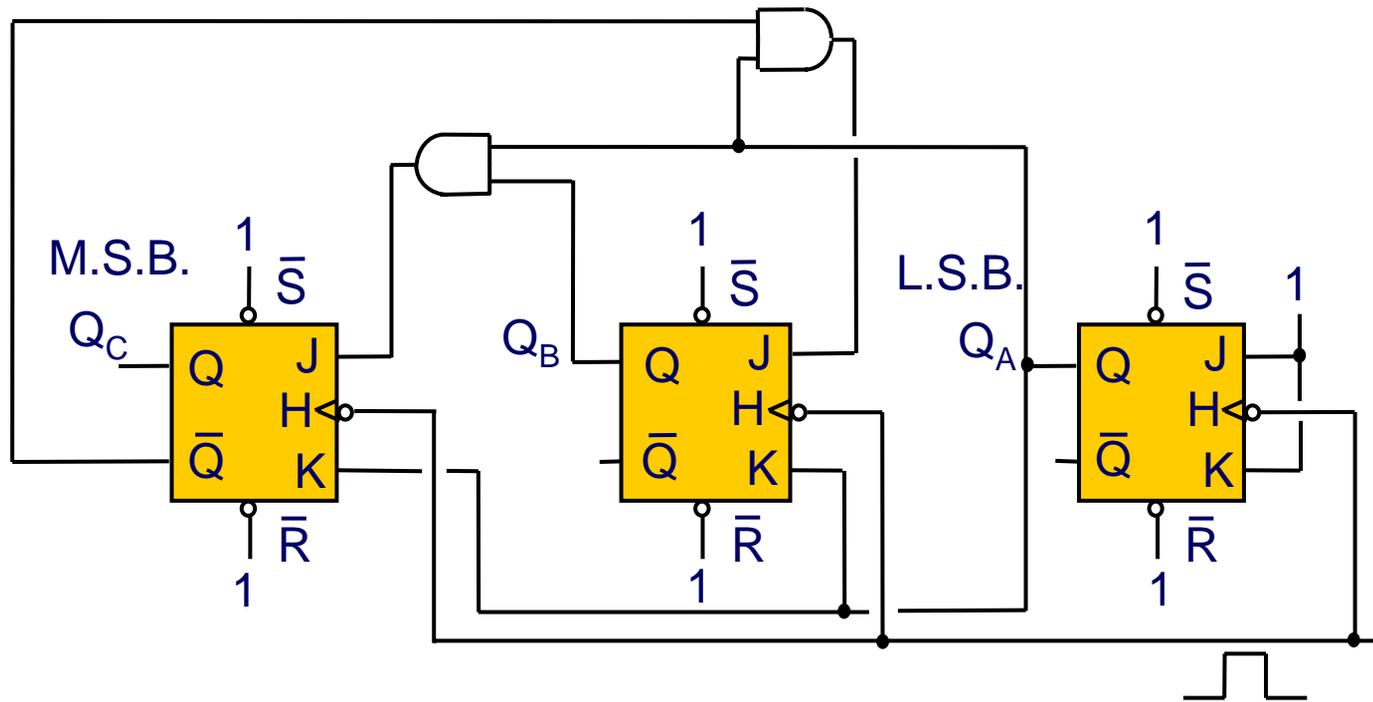
$$J_C = Q_B Q_A$$

$Q_C \setminus Q_B Q_A$	00	01	11	10
0	X	X	X	X
1	0	1	X	X

$$K_C = Q_A$$

$$J_A = K_A = 1$$

III – Les compteurs et décompteurs



La table de vérité est plus longue à établir mais le schéma logique est plus simple.

III – Les compteurs et décompteurs

Réalisation en bascules D.

Q recopie D au prochain front.

Remise à 0
synchrone

$$D_A = \overline{Q_A}$$

Q_C	Q_B	Q_A	D_C	D_B	D_A
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0

$Q_C \setminus Q_B Q_A$	00	01	11	10
0	0	0	1	0
1	1	0	X	X

$$D_C = Q_B Q_A + Q_C \overline{Q_A}$$

$Q_C \setminus Q_B Q_A$	00	01	11	10
0	0	1	0	1
1	0	0	X	X

$$D_B = Q_B \overline{Q_A} + \overline{Q_C} \overline{Q_B} Q_A$$

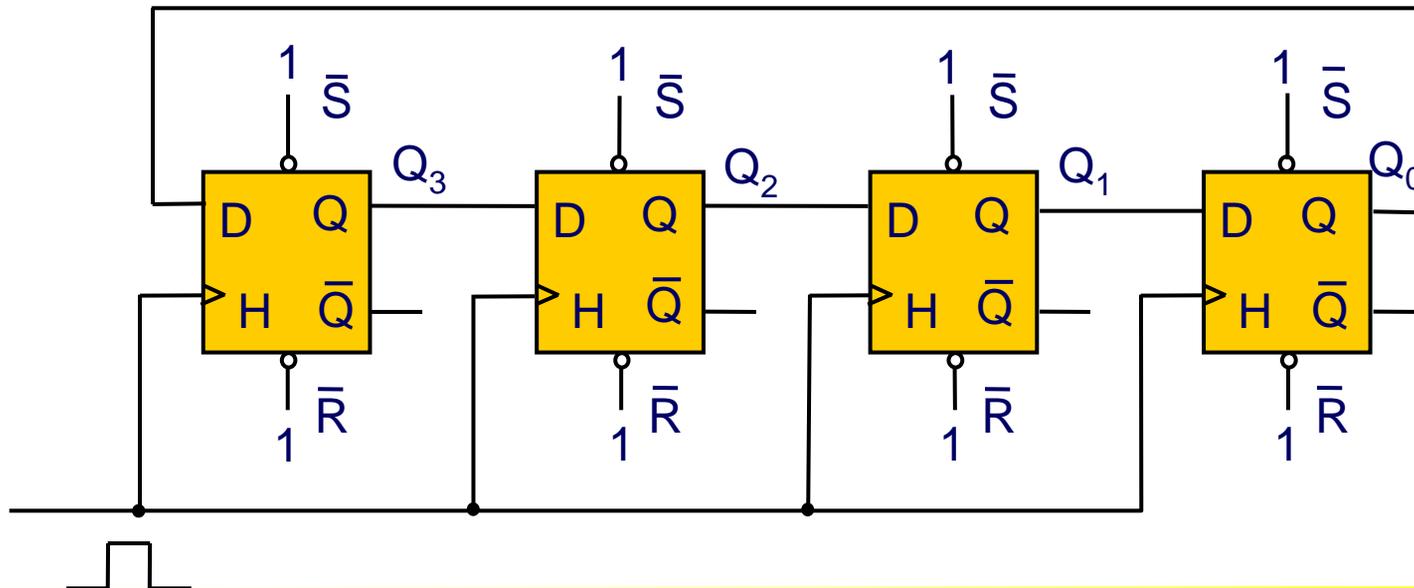
III – Les compteurs et décompteurs

Les compteurs circulaires :

On appelle compteur circulaire un compteur pour lequel la sortie de la dernière bascule est rebouclée sur l'entrée de la première.

Compteur en anneau \Rightarrow Rebouclage de la sortie Q.

Compteur en anneau à base de bascules D :



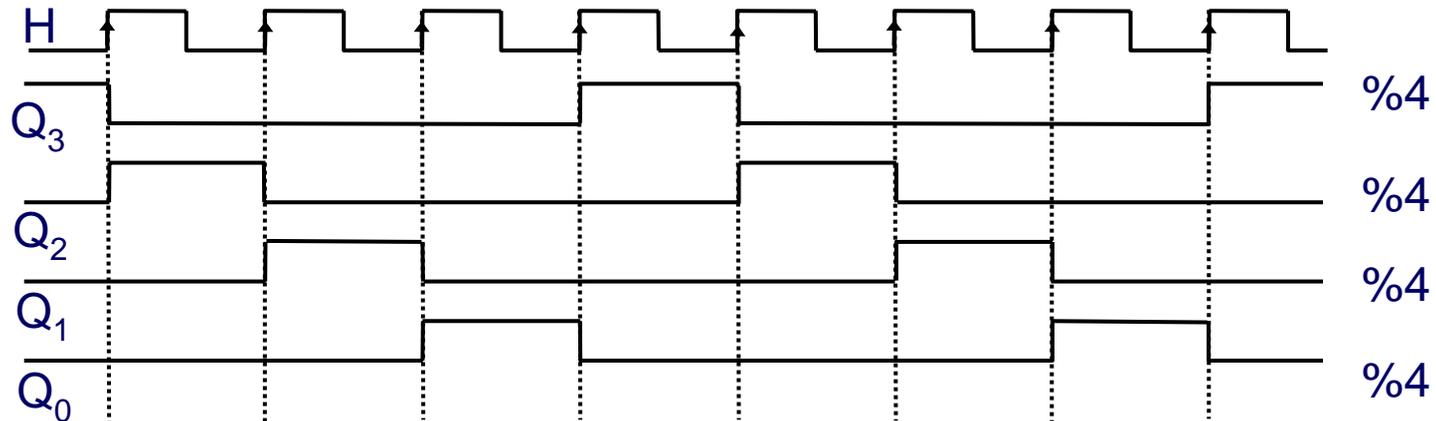
III – Les compteurs et décompteurs

Différences avec un compteur normal :

- N bascules pour un modulo N.
- Chaque sortie a pour fréquence F / N (F fréquence de H).

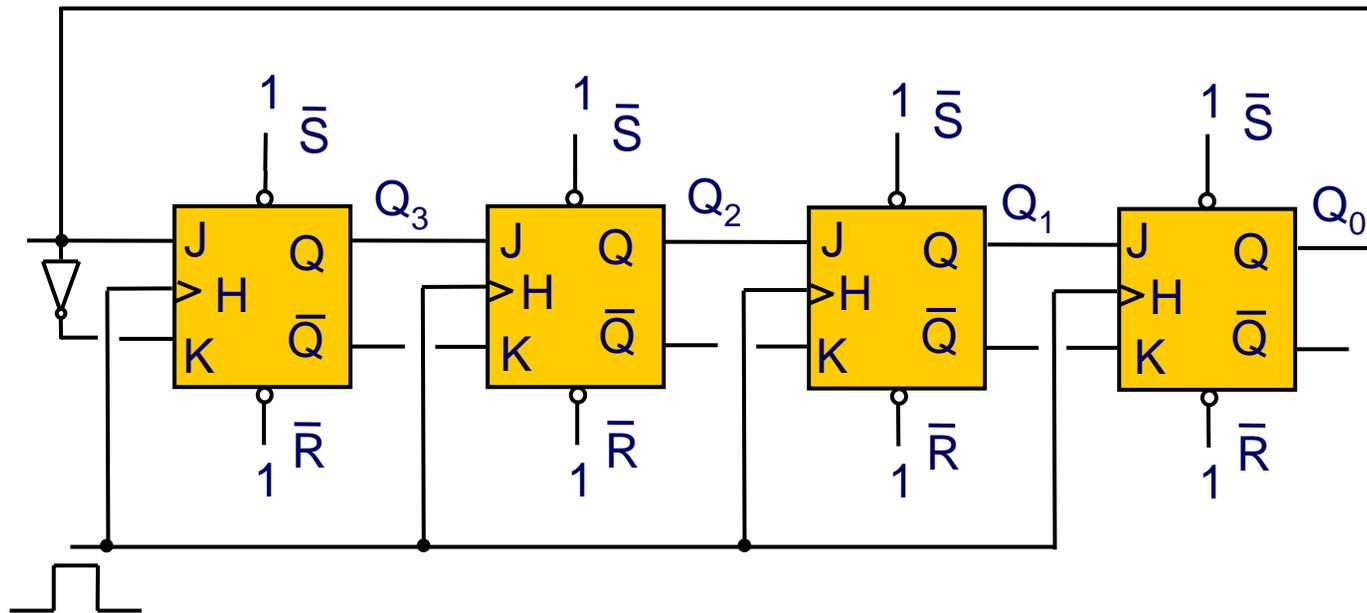
Q_3	Q_2	Q_1	Q_0	imp.
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4

Le problème de ce compteur réside dans le chargement initial d'un état 1.



III – Les compteurs et décompteurs

Compteur en anneau à base de bascules JK :



Compteur Johnson \Rightarrow Rebouclage de la sortie \bar{Q} .

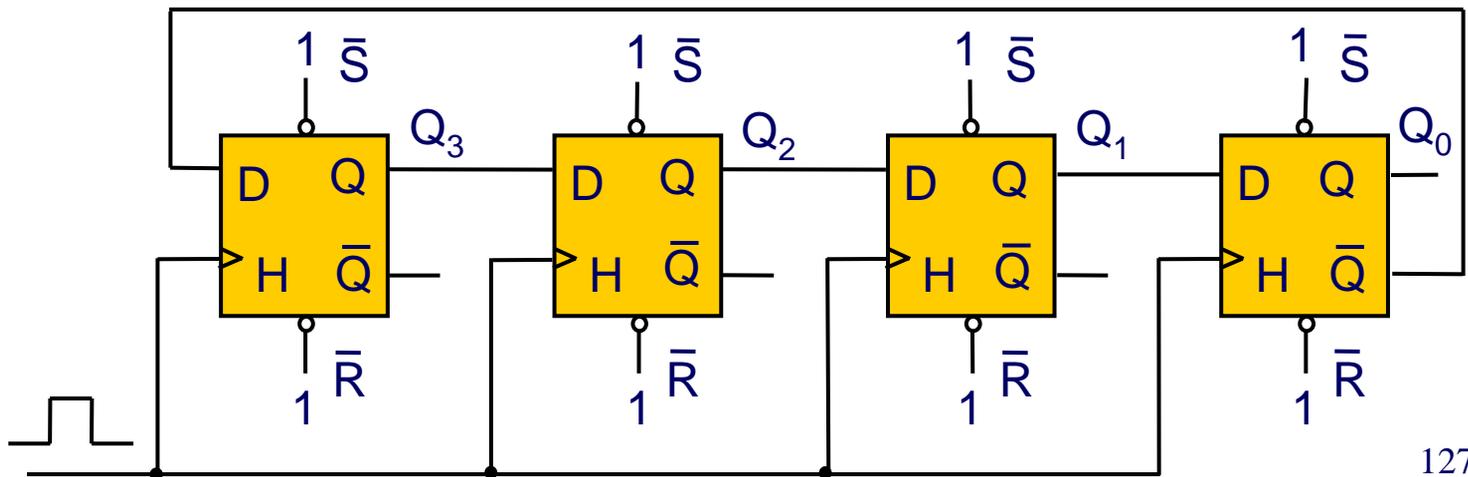
Pour cette structure, l'état logique 1 se charge directement à partir de la sortie complémentée.

III – Les compteurs et décompteurs

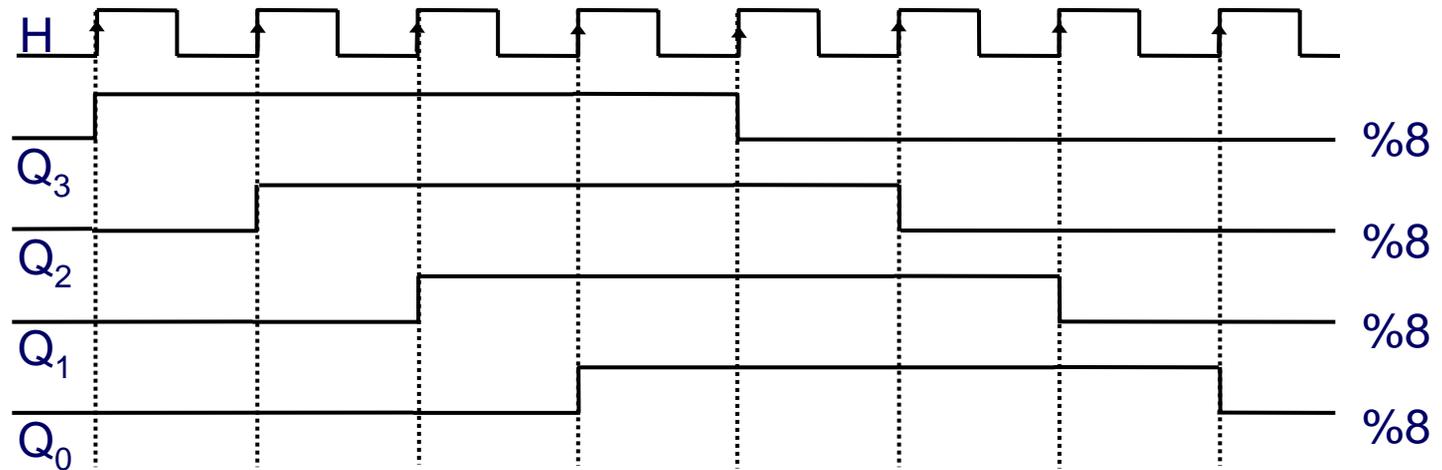
Q_3	Q_2	Q_1	Q_0	imp.
0	0	0	0	0
1	0	0	0	1
1	1	0	0	2
1	1	1	0	3
1	1	1	1	4
0	1	1	1	5
0	0	1	1	6
0	0	0	1	7
0	0	0	0	8

Différences avec un compteur normal :

- N bascules pour un modulo 2N.
- Chaque sortie a pour fréquence $F / 2N$ (F fréquence de H).



III – Les compteurs et décompteurs



Pour un compteur Johnson, si $N = 5$, on obtient un compteur modulo 10 encore appelé anneau de Regener. Chaque sortie a pour fréquence $F/10$.

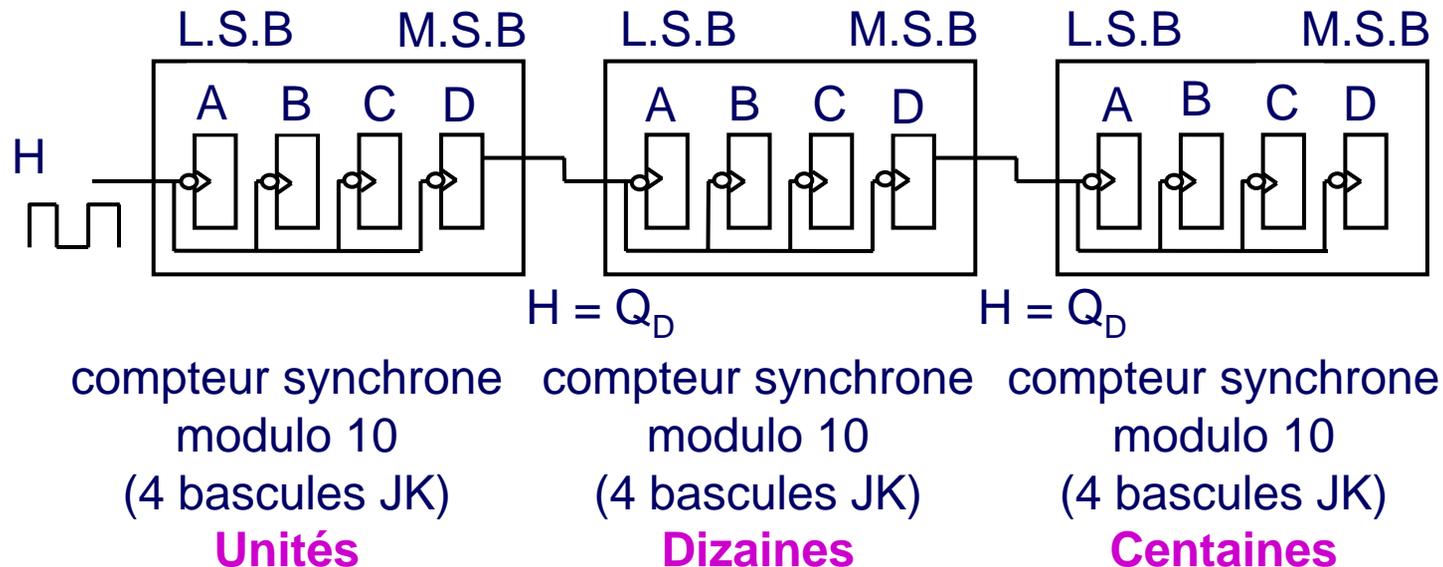
III – Les compteurs et décompteurs

III.4 – Les compteurs en cascades :

Exemple du compteur par 1000 :

Deux solutions sont possibles :

- 10 bascules en cascades ($2^{10} = 1024$) \Rightarrow Trouver les équations.
- 3 compteurs DCB en cascades \Rightarrow 1 afficheur 7 segments par compteur.



III – Les compteurs et décompteurs

Horloge externe : Horloge des unités.

MSB des unités : Horloge des dizaines.

MSB des dizaines : Horloge des centaines.

Fonctionnement :

- Le compteur des unités passe de 9 à 0 \Rightarrow le MSB crée un front descendant et incrémente le compteur des dizaines.

- Le compteur des dizaines passe de 9 à 0 \Rightarrow le MSB crée un front descendant et incrémente le compteur des centaines.

L'ensemble du compteur est asynchrone.

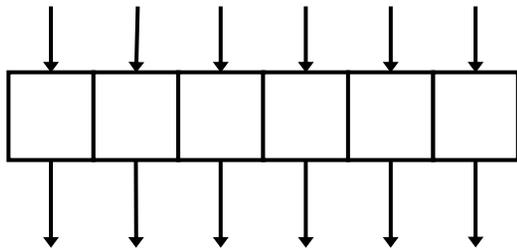
Pour des bascules actives sur front montant c'est la sortie complémentée du MSB qu'il faut relier aux horloges.

IV – Les Registres

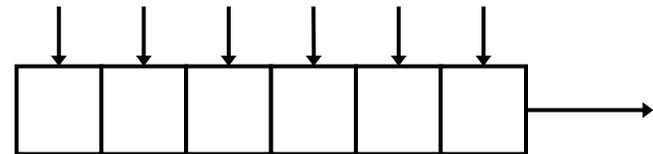
Un registre est une association linéaire de plusieurs bascules servant à mémoriser des informations ou à les décaler.

Chaque bascule matérialise une position mémoire pouvant recevoir un bit. Un registre se caractérise par :

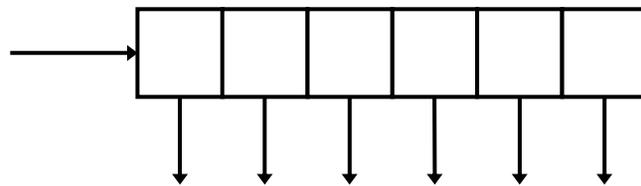
- Son nombre de positions mémoire.
- Son mode de lecture et d'écriture (chargement) des informations.



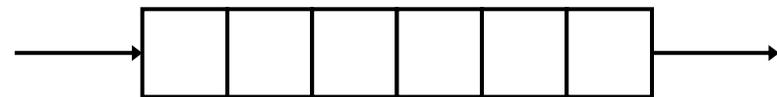
Ecriture parallèle; lecture parallèle



Ecriture parallèle; lecture série



Ecriture série; lecture parallèle



Ecriture série; lecture série

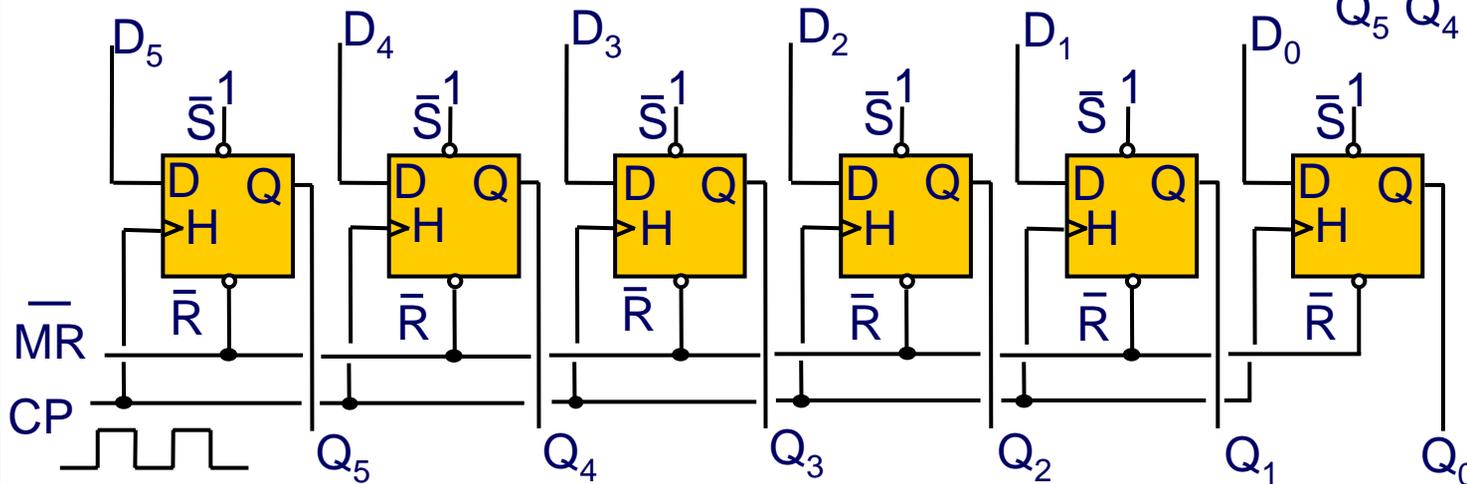
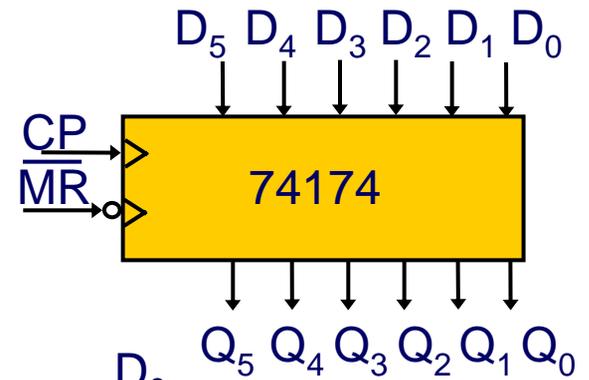
IV – Les Registres

IV.1 – Les registres de mémoire :

Pour ces registres, le chargement et la lecture s'effectuent en parallèle.

Le circuit intégré 74174 :

Registre de mémoire 6 bits avec remise à 0 :
Le chargement s'effectue sur un seul coup d'horloge.



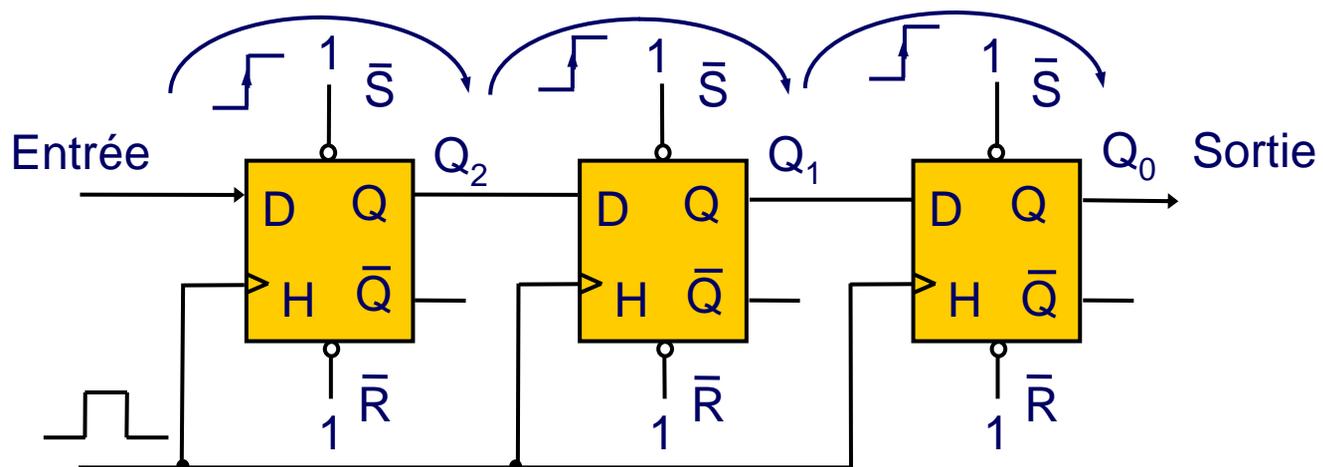
IV – Les Registres

IV.2 – Les registres à décalages :

Ce sont des circuits utilisés pour transférer des données d'un registre à un autre un bit à la fois : le contenu de la cellule i se transfère dans la cellule $i-1$.

Écriture série / lecture série :

A chaque front montant, la sortie d'une bascule recopie son entrée c'est à dire la sortie de la bascule précédente.

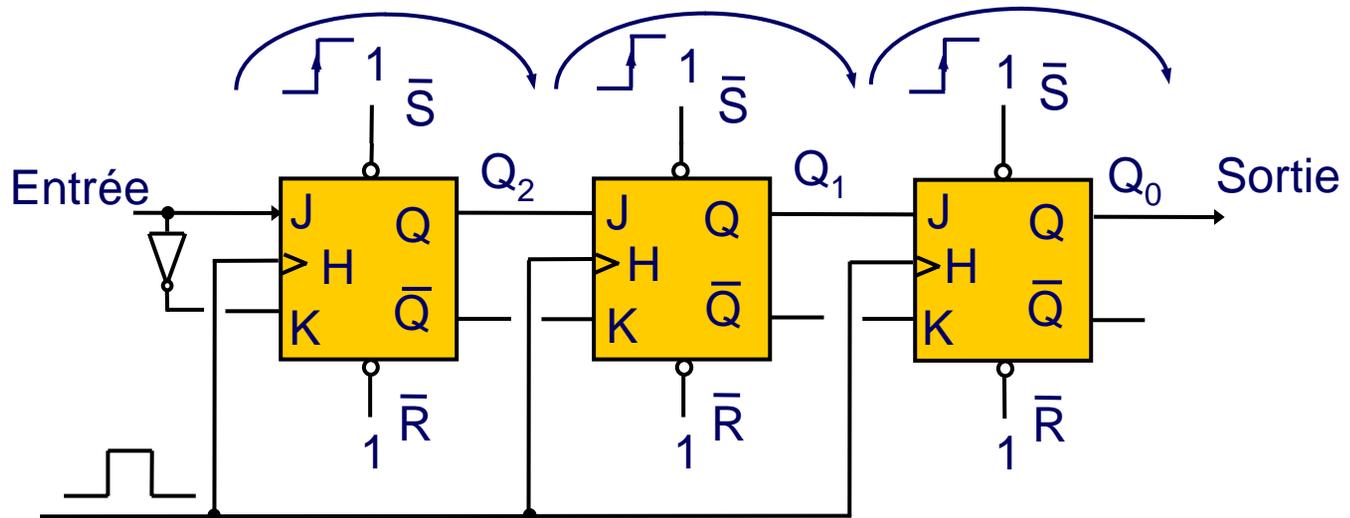


IV – Les Registres

Structure en bascules JK :

- $Q_2 = 0 \Rightarrow J_1 = 0; K_1 = 1 \Rightarrow Q_1 = 0$ au prochain front.
- $Q_2 = 1 \Rightarrow J_1 = 1; K_1 = 0 \Rightarrow Q_1 = 1$ au prochain front.

La porte inverseuse en entrée permet de charger la valeur d'entrée sur la sortie de la première bascule.

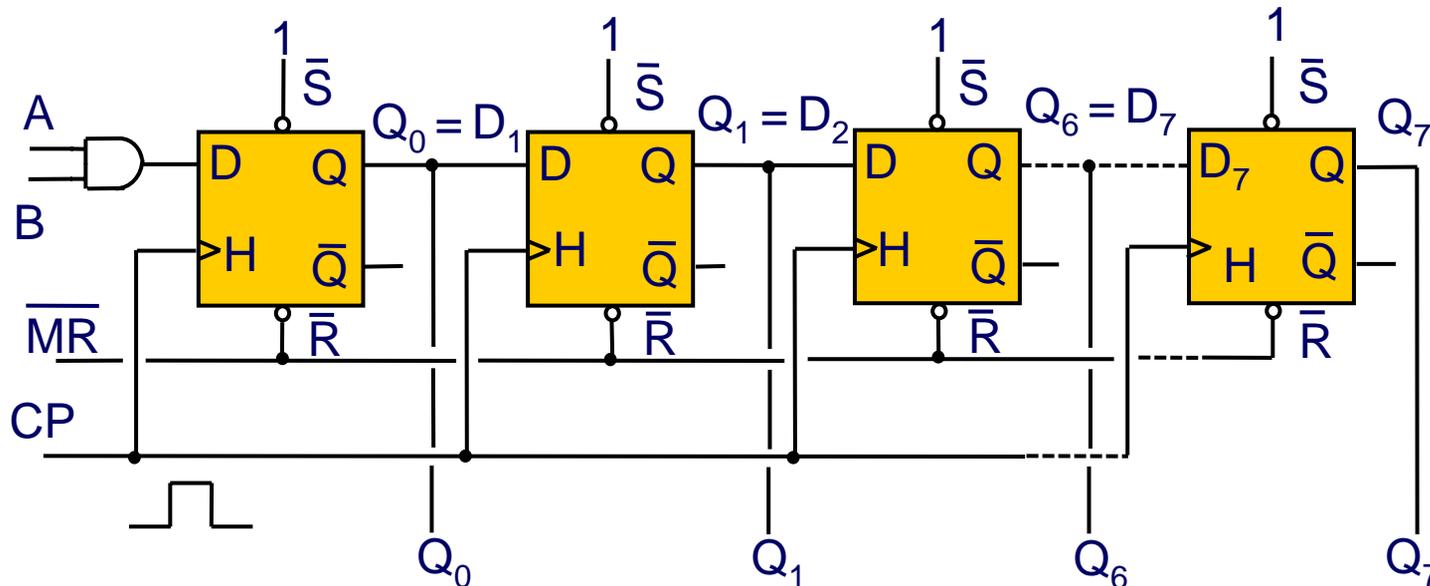
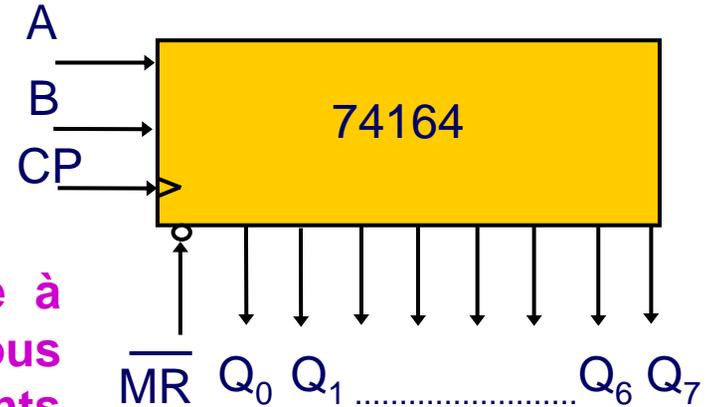


IV – Les Registres

Écriture série / lecture parallèle :

Le circuit intégré 74164 :

Registre à décalage 8 bits avec remise à zéro : Le produit logique AB se décale tous les fronts montants de CP. Il faut 8 fronts pour charger un octet.



IV – Les Registres

Ecriture parallèle / lecture série :

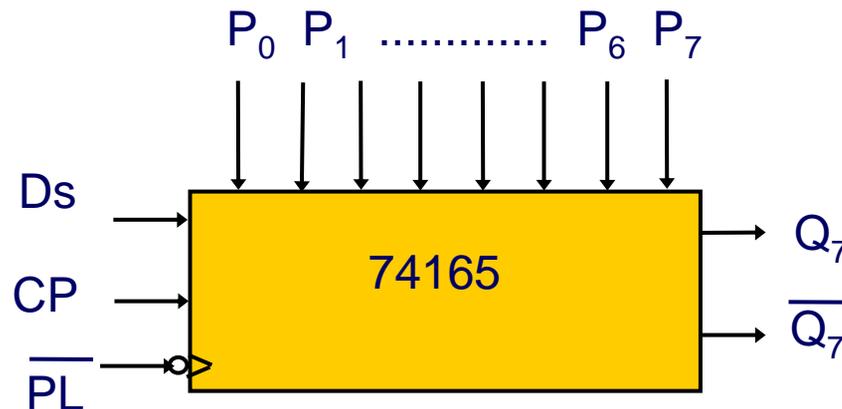
Le circuit intégré 74165 :

Registre à décalage 8 bits.

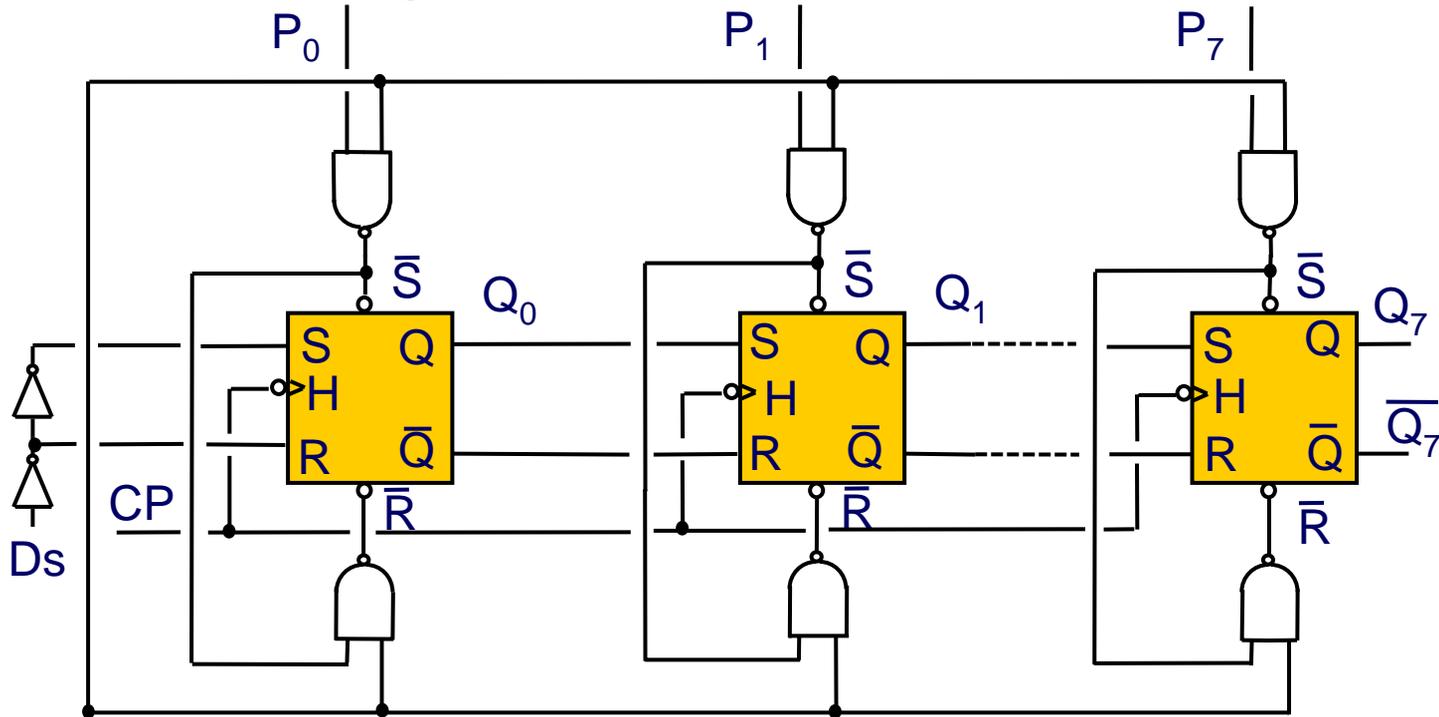
Le chargement parallèle s'effectue à l'aide des entrées asynchrones et est prioritaire.

Le décalage série est synchrone et s'effectue sur chaque front.

Ce registre possède également une entrée série Ds (synchrone).



IV – Les Registres



Chargement parallèle ($\overline{PL}=0$) :

$P_0 = 1$
 $\Rightarrow S=0, \overline{R}=1$
 $\Rightarrow Q_0=1, \overline{Q}_0=0$

$P_0 = 0$
 $\Rightarrow S=1, \overline{R}=0$
 $\Rightarrow Q_0=0, \overline{Q}_0=1$

Chargement et décalage série ($\overline{PL}=1$) :

$Ds = 1$
 $\Rightarrow S=1, R=0$
 $\Rightarrow Q_0=1, \overline{Q}_0=0$ (front)

$Ds = 0$
 $\Rightarrow S=0, R=1$
 $\Rightarrow Q_0=0, \overline{Q}_0=1$ (front)

IV – Les Registres

Ecriture parallèle / lecture parallèle :

Le fonctionnement de ce registre correspond à celui du registre de mémoire (exemple 74178).

Registre universel :

Un registre universel est un registre offrant toutes les possibilités de lecture et d'écriture :

- Entrée série.**
- Sortie série.**
- Entrées parallèles.**
- Sorties parallèles.**

Les mémoires et circuits logiques programmables

I – Les mémoires

On appelle mémoires des circuits pouvant enregistrer, stocker et restituer des informations binaires. Elles peuvent être à accès sélectif (temps d'accès identique pour toutes les cases mémoire) ou séquentiel (temps d'accès dépendant de la localisation des cases).

I.1 – Les mémoires vives (RAM) :

Les mémoires vives (Random Access Memory) sont des mémoires volatiles : l'information stockée est conservée que si le circuit est sous tension. La majorité des mémoires à semi-conducteurs est volatile, alors que la totalité des mémoires magnétiques est rémanente.

Dans une RAM, il est aussi facile de lire que d'écrire une donnée. On distingue :

- Des RAM statiques (SRAM) dont les cellules sont des bistables.
- Des RAM dynamiques (DRAM) dont les cellules sont des capacités (ce qui nécessite une opération de rafraîchissement).

I – Les mémoires

I.2 – Les mémoires mortes (ROM) :

Les mémoires mortes (Read Only Memory) sont des circuits pour lesquels le nombre d'opérations de lecture est très supérieur au nombre d'opérations d'écriture. Techniquement, les données dans une mémoire morte sont écrites (programmées) une seule fois et sont conservées en absence d'alimentation.

On distingue :

- **MROM (Masked ROM) : L'écriture des données est effectuée une fois pour toute par le fabricant suivant les spécifications du client. On utilise un masque pour établir les interconnexions électriques.**

- **PROM (Programmable ROM) : La programmation n'est pas faite en usine mais par l'utilisateur à l'aide d'un appareil. Une fois programmée, une PROM ne peut être ni effacée, ni reprogrammée.**

I – Les mémoires

Parmi les PROM, on distingue :

- **PROM à fusibles** : La programmation se réalise par destruction physique d'un ou plusieurs fusibles contenus sur la puce de silicium. Ces mémoires sont programmées une fois pour toutes.
- **EPROM (Erasable PROM)** : Le processus de programmation est réversible et la mémoire programmée peut à nouveau être rendue vierge à tout instant.
- **OTPROM (One Time PROM)** : Ce sont des EPROM dans lesquelles on a inhibé la possibilité d'effacement pour une raison technologique liée généralement au coût de fabrication. Ces mémoires ne sont pas effaçables (même si sur la puce elles en avaient la possibilité initialement).

I – Les mémoires

Parmi les EPROM, on distingue :

- **UV PROM (PROM effaçables aux UV) :** Ces mémoires se programment électriquement et s'effacent par exposition aux UV. Leur boîtier est muni d'une fenêtre en quartz permettant l'exposition.

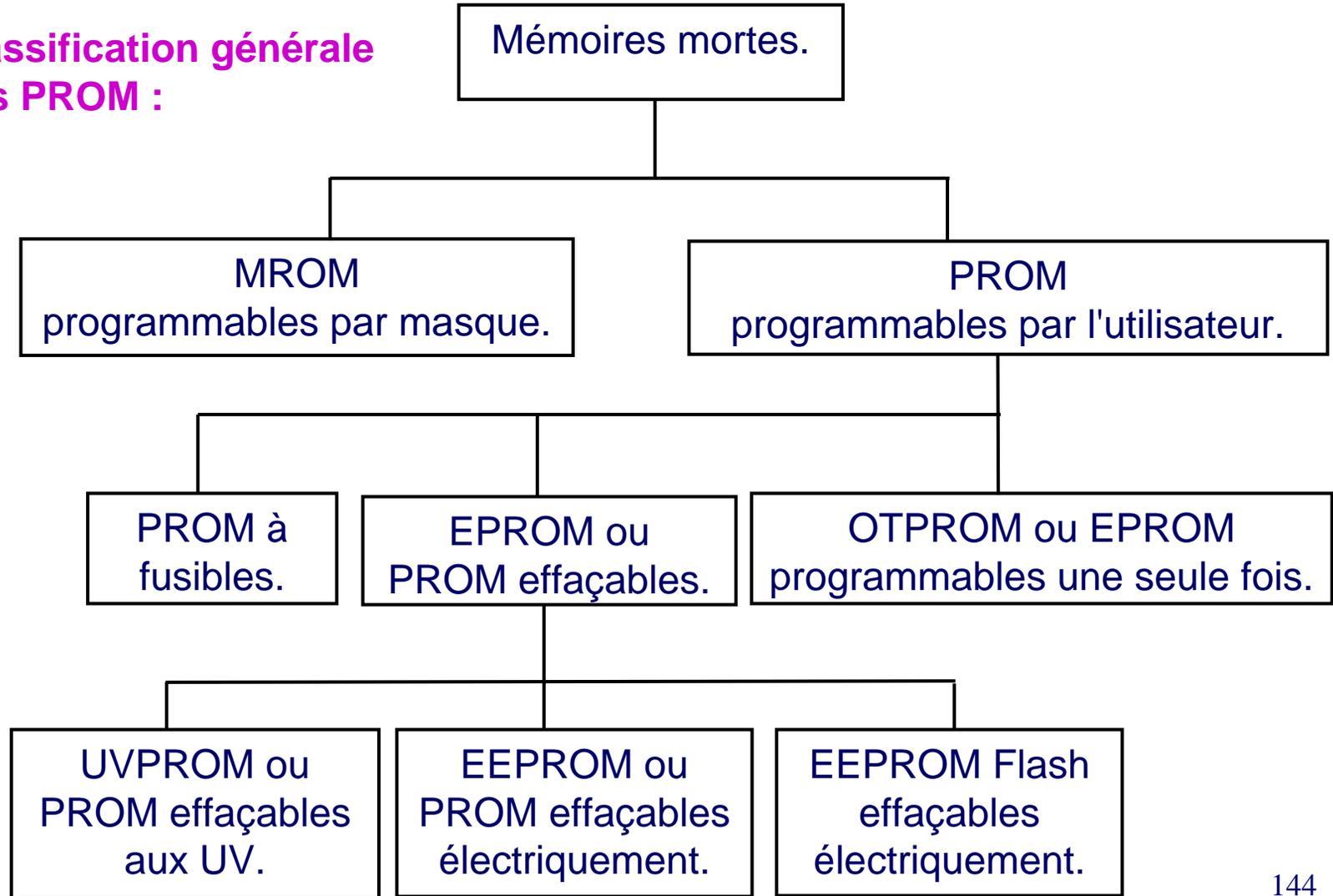
- **EEPROM (Electrically Erasable PROM) :** Ces mémoires se programment électriquement et s'effacent par niveaux de tensions. De plus, l'effacement dure environ 10 ms par rapport à 20 minutes pour une UV PROM.

- **EEPROM FLASH :** Ces mémoires sont effaçables et programmables électriquement. Par rapport aux EEPROM, elles sont plus rapides (en terme de programmation et d'effacement).

Les EEPROM autorisent l'effacement sélectif d'un octet quelconque, les FLASH EEPROM ne permettent que l'effacement total de la mémoire.

I – Les mémoires

Classification générale
des PROM :



I – Les mémoires

I.3 – Principes de fonctionnement :

Un boîtier mémoire possède :

- Des adresses (permettant de repérer la localisation physique de la donnée).
- Des données pouvant être mémorisées (écriture) ou récupérées (lecture).
- Des signaux de commandes pour lire ou écrire les informations.

Contenu	Adresse
mot 0	000
mot 1	001
mot 2	010
mot 3	011
mot 4	100
mot 5	101
mot 6	110
mot 7	111

On définit également la capacité de la mémoire qui est la quantité de bits que l'on peut mémoriser. Sa capacité est un multiple du Kilo ou du Méga :

- 1 KO = 2^{10} = 1024 octets.
- 1 MO = 2^{20} = 1 048 576 octets.
- 1 octet = 8 bits (1 byte).

I – Les mémoires

L'EPROM 2716 :

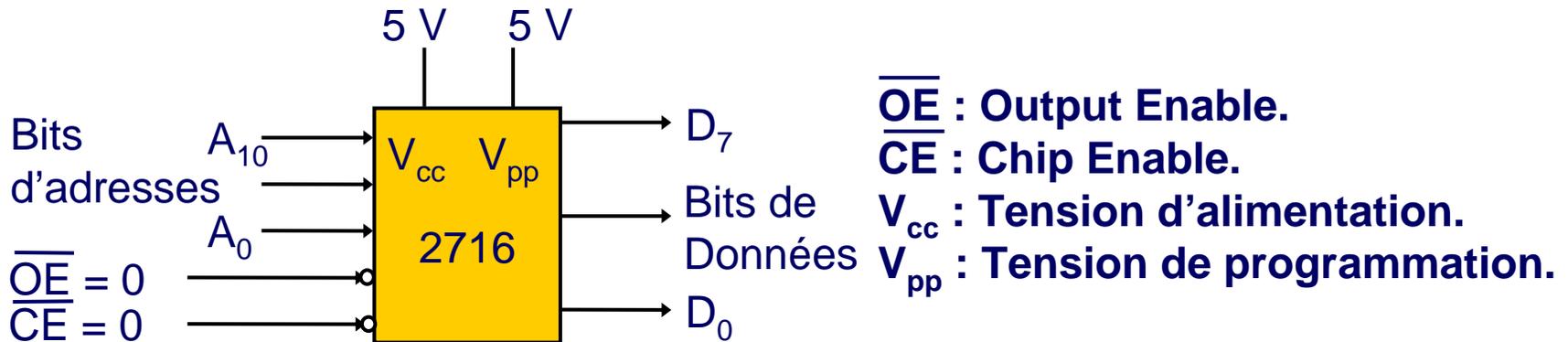
EPROM permettant de stocker 2 Kilo octets de données.

27 \Rightarrow EPROM.

16 \Rightarrow 8 bits de données (D_0 à D_7).

x 2 K de zone mémoire (11 bits d'adresses (A_0 à A_{10})).

(2K = 2048 = 2^{11}).



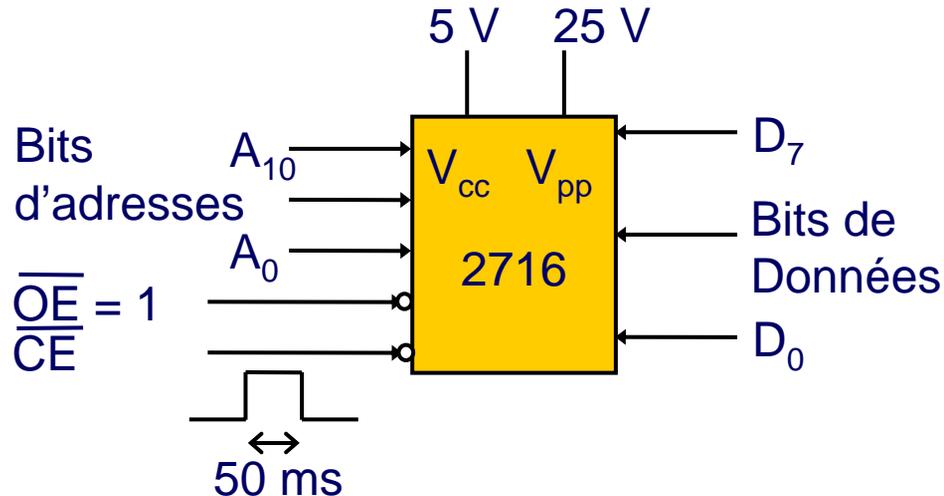
Lecture d'une donnée ($V_{pp} = 5V$) :

- Placer l'adresse désirée sur les lignes A_0 à A_{10} .
- Placer les signaux \overline{OE} et \overline{CE} à 0.
- L'octet placé à l'adresse est alors présent sur D_0 à D_7 .

I – Les mémoires

Ecriture d'une donnée ($V_{pp} = 25V$) :

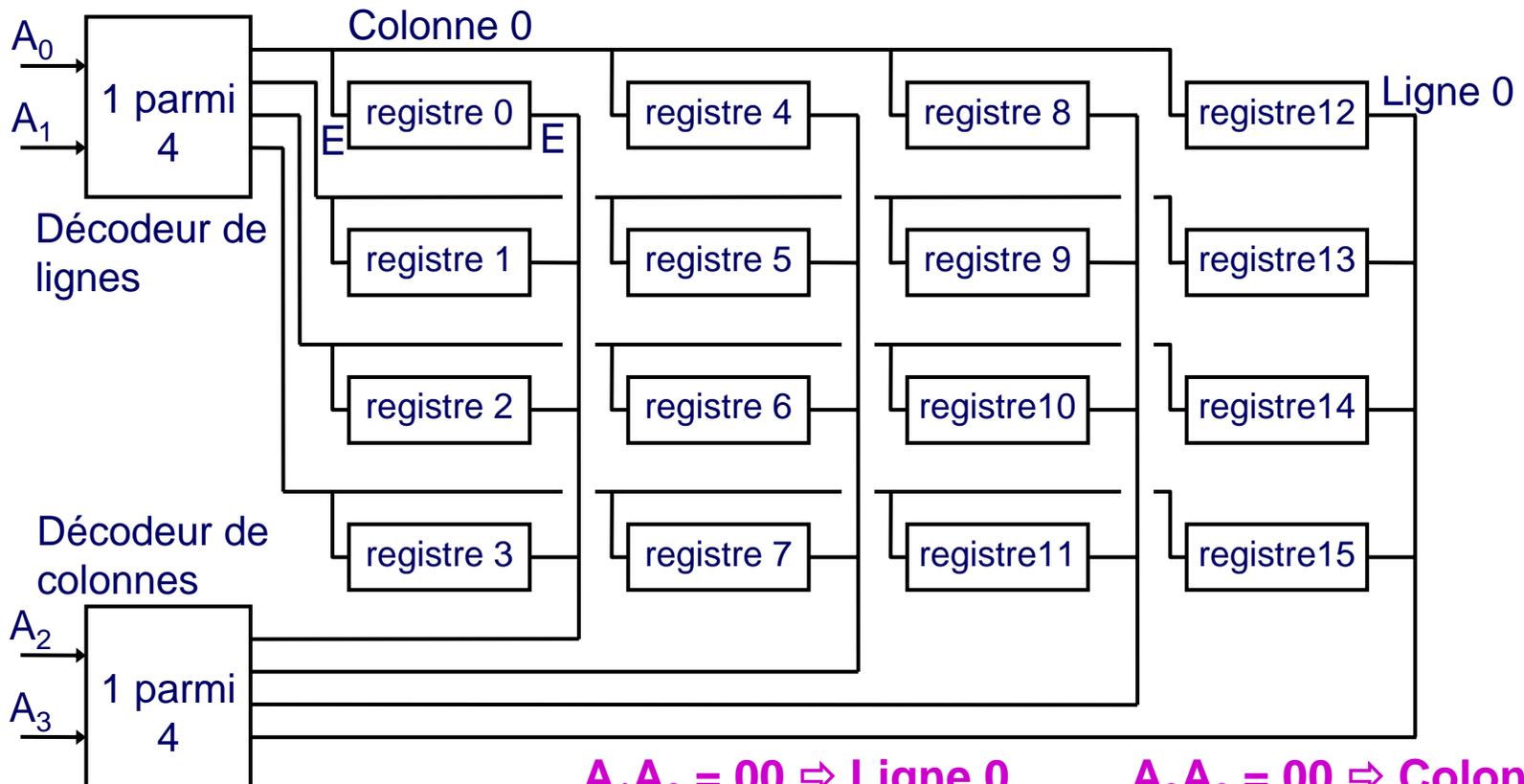
- Placer \overline{OE} à 1.
- Placer l'adresse désirée sur les lignes A_0 à A_{10} .
- Appliquer le mot de 8 bits sur D_0 à D_7 .
- Appliquer sur \overline{CE} une impulsion de 50 ms.



La zone mémoire varie de : 0000000000 à 1111111111.
000 à 7FF.

I – Les mémoires

Structure générale : Exemple d'un boîtier 16 x8.



- 16 registres 8 bits.
- 4 bits d'adresses.
- 2 entrées de validation.

$A_1A_0 = 00 \Rightarrow$ Ligne 0

$A_1A_0 = 01 \Rightarrow$ Ligne 1

$A_1A_0 = 10 \Rightarrow$ Ligne 2

$A_1A_0 = 11 \Rightarrow$ Ligne 3

$A_3A_2 = 00 \Rightarrow$ Colonne 0

$A_3A_2 = 01 \Rightarrow$ Colonne 1

$A_3A_2 = 10 \Rightarrow$ Colonne 2

$A_3A_2 = 11 \Rightarrow$ Colonne 3

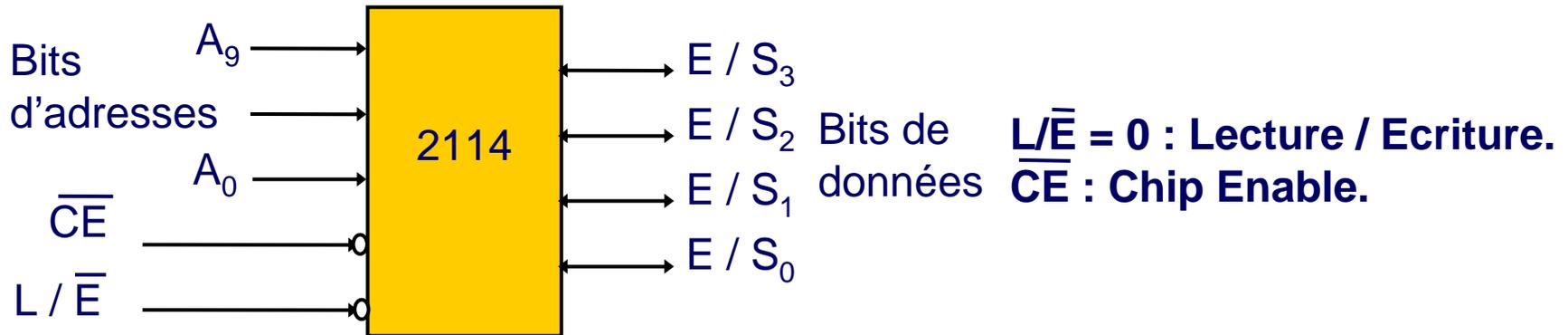
I – Les mémoires

La RAM 2114 :

RAM permettant de stocker 1 Kilo mot de 4 bits.

⇒ 4 bits de données (E/S₀ à E/S₃).

⇒ 1 K de zone mémoire (10 bits d'adresses (A₀ à A₉)).



Lecture d'une donnée :

- Placer L/ \overline{E} à 1.
- Placer \overline{CE} à 0.
- Placer l'adresse désirée sur les lignes A₀ à A₉.
- Le mot apparaît sur S₀ à S₃.

Principe identique en écriture avec L/ \overline{E} = 0 et le mot présent sur E₀ à E₃

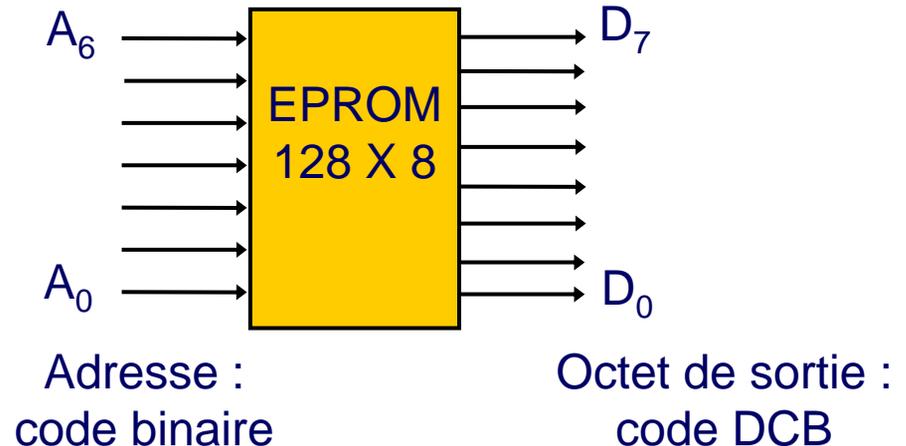
I – Les mémoires

Applications :

Stockage des données et des programmes informatiques.

Transcodage.

Exemple d'un transcodeur
binaire vers DCB (0 à 99) :

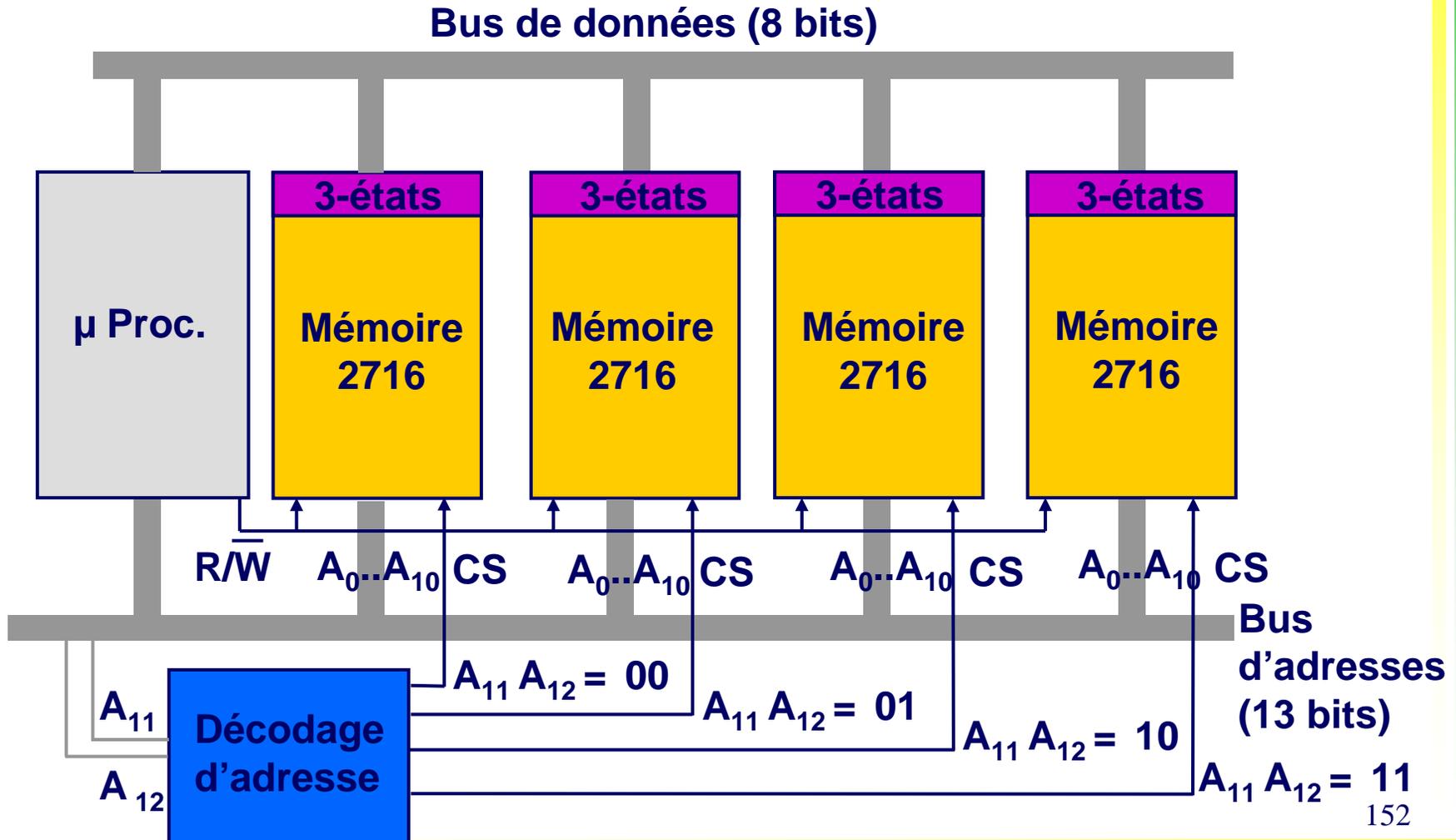


Entrée \Rightarrow Nombre binaire de $(0000000)_2$ à $(1100011)_2$ codés sur 7 bits.
Sortie \Rightarrow Equivalent DCB de (00000000) à (10011001) codés sur 8 bits.

L'intégralité du transcodeur peut être réalisé en programmant simplement une ROM 128 X 8.

I – Les mémoires

Structure générale :



II – Les circuits programmables

Les circuits logiques programmables (PLD pour Programmable Logic Device) sont des circuits permettant de réaliser des fonctions logiques sans aucun câblage de la part de l'opérateur.

On distingue :

- PAL (Programmable Array Logic) : Ces circuits sont constitués de réseaux de portes ET et OU (appelés PLA) câblées de façon à produire des sorties dont l'expression booléenne est une somme de produits de plusieurs entrées.

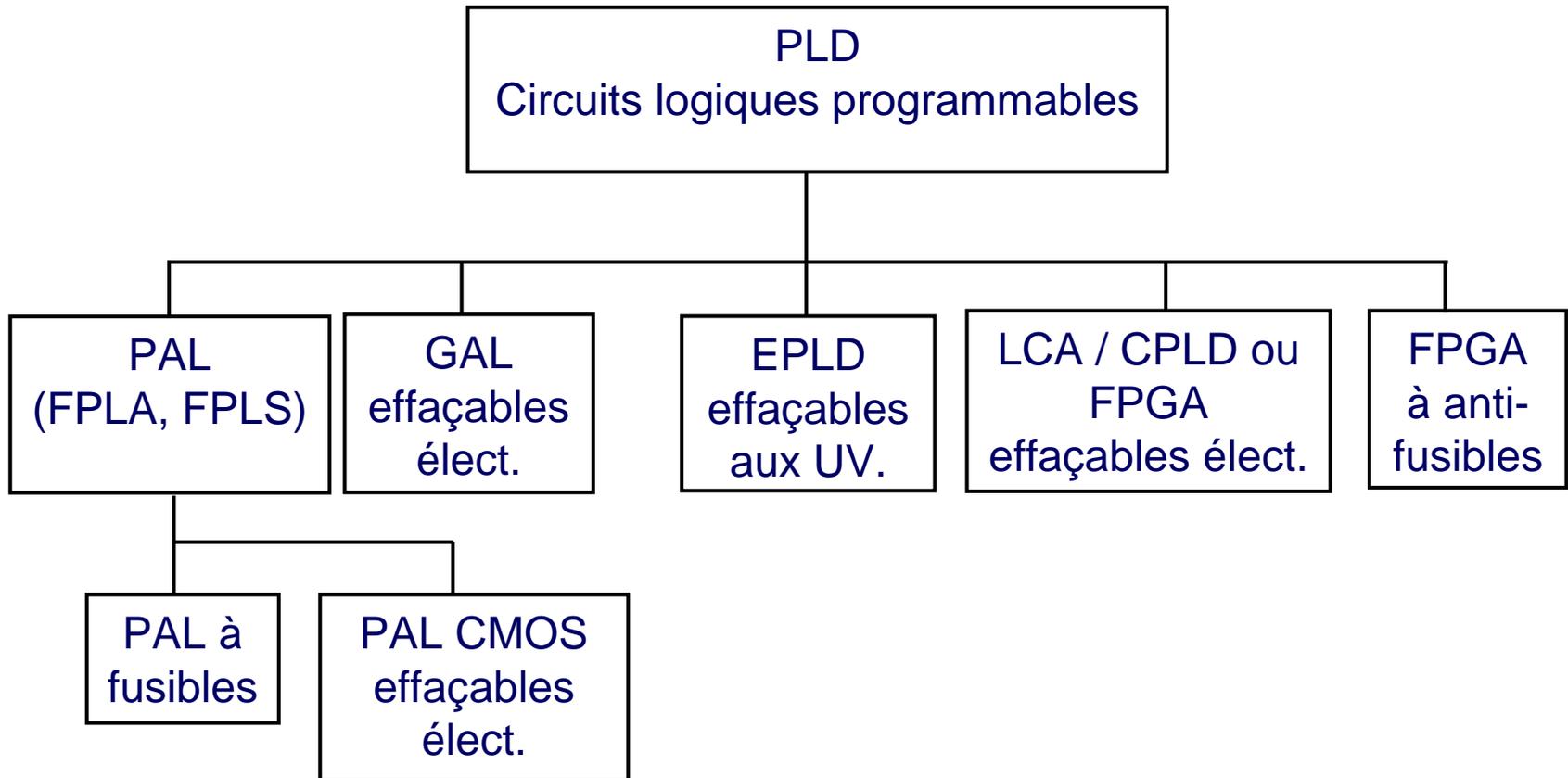
L'utilisateur peut détruire les liaisons entre les entrées logiques et les portes ET et OU de façon à obtenir la fonction désirée. Une fois programmé un PAL ne peut être ni effacé ni reprogrammé.

- PLA : Seule la matrice ET est programmable.
- FPLA (Field PLA) : Les matrices ET et OU sont programmables.
- FPLS (Field Programmable Logic Sequencer) : PAL contenant des registres (fonctionnement séquentiel).

II – Les circuits programmables

- **EPLD (Erasable Programmable Logic Device) : Ces circuits logiques sont programmables électriquement et effaçables aux ultraviolets.**
- **GAL (Generic Array Logic) : Ces circuits sont programmables et effaçables électriquement.**
- **LCA (Logic Cell Array) : Ce sont de gros ensembles de blocs logiques élémentaires (de 2000 à 10000 portes) que l'utilisateur peut interconnecter pour réaliser la ou les fonctions logiques voulues.**
- **FPGA (Field Programmable Gate Array) : Ce sont les circuits logiques les plus évolués et comprenant le plus de portes. Les circuits les plus récents sont les FPGA à anti-fusibles programmables électriquement mais non effaçables.**

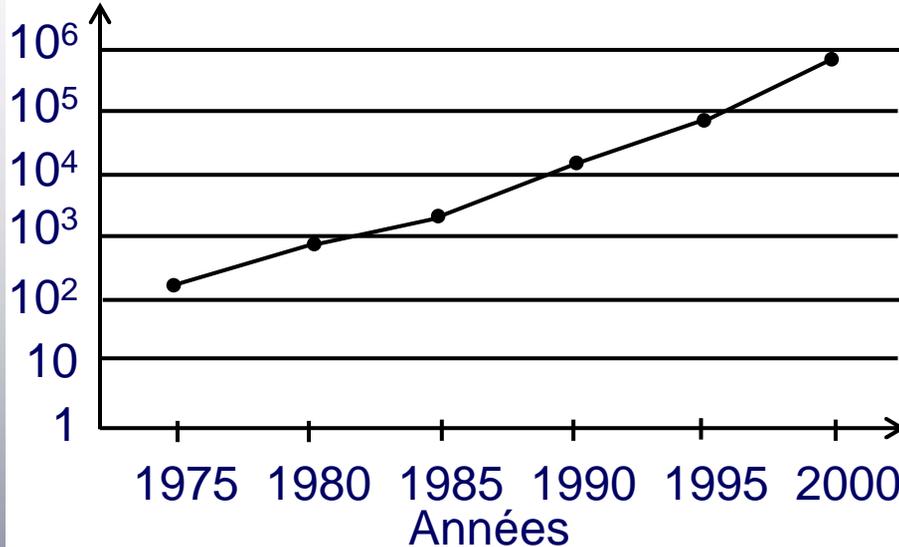
II – Les circuits programmables



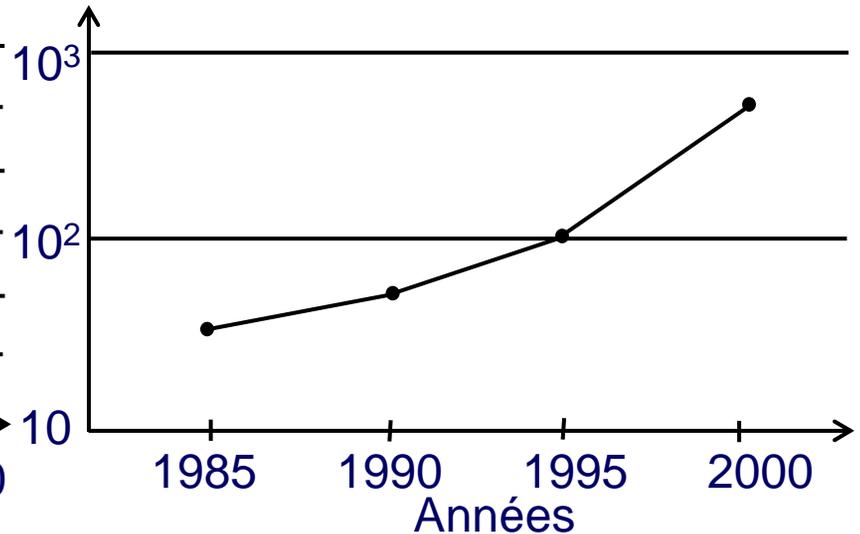
De façon générale, on parle de SPLD (Simple PLD) pour des circuits de type PAL, de CPLD (Complex PLD) pour des circuits faisant dialoguer entre eux des circuits de types SPLD et de FPGA pour les circuits de structures très complexes.

II – Les circuits programmables

Nombre de portes



Fréquence de travail (MHz)

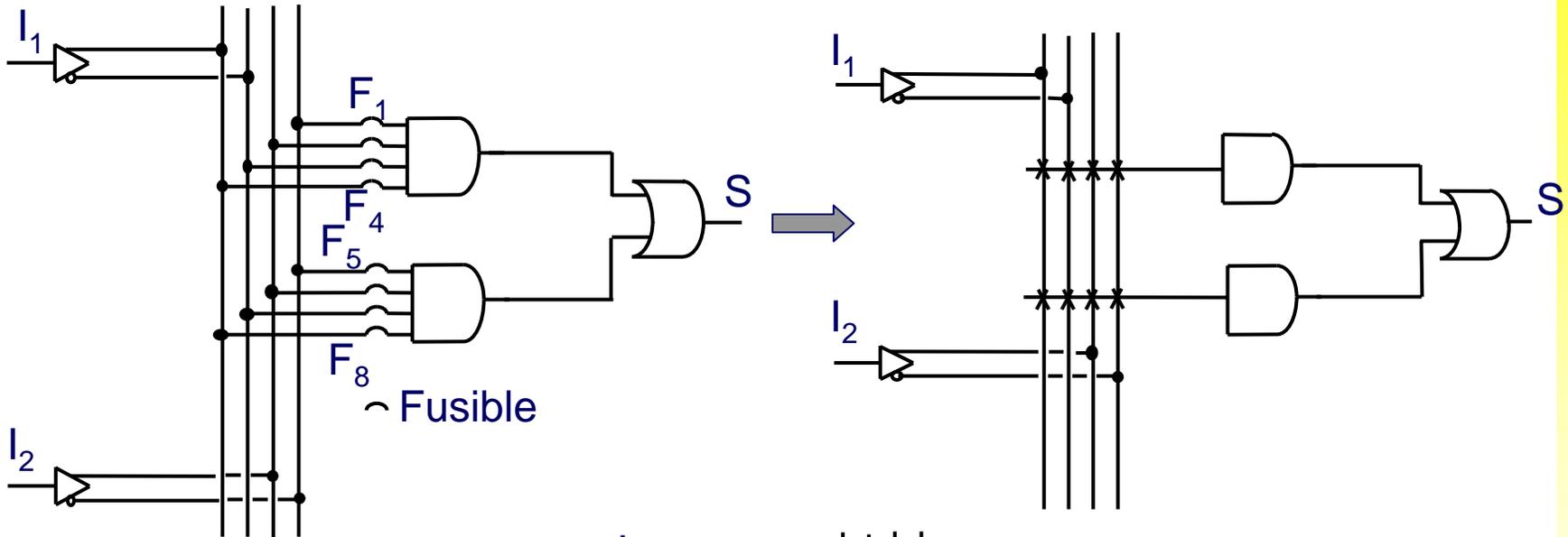


II.1 – Les circuits PAL :

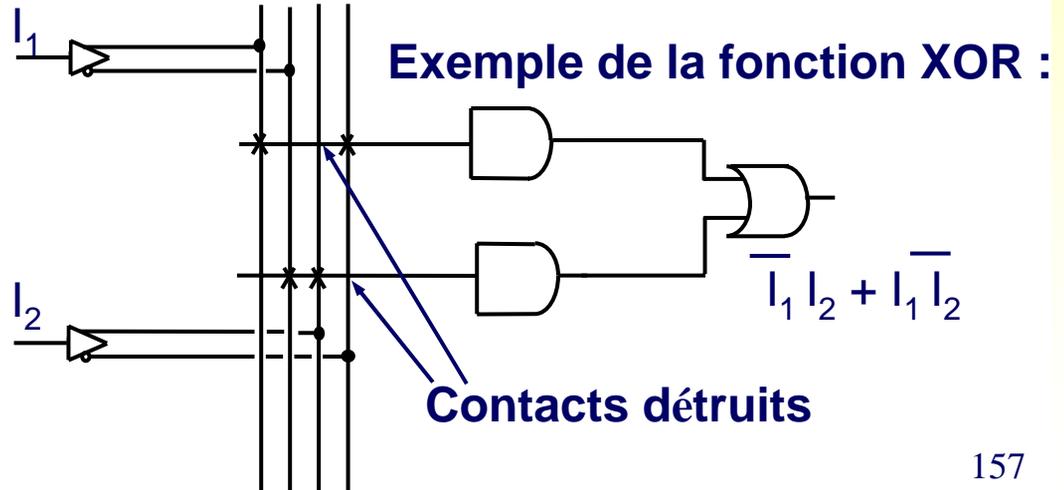
Les connexions entre les portes étant nombreuses, on adopte la convention suivante pour une meilleure lisibilité :



II – Les circuits programmables

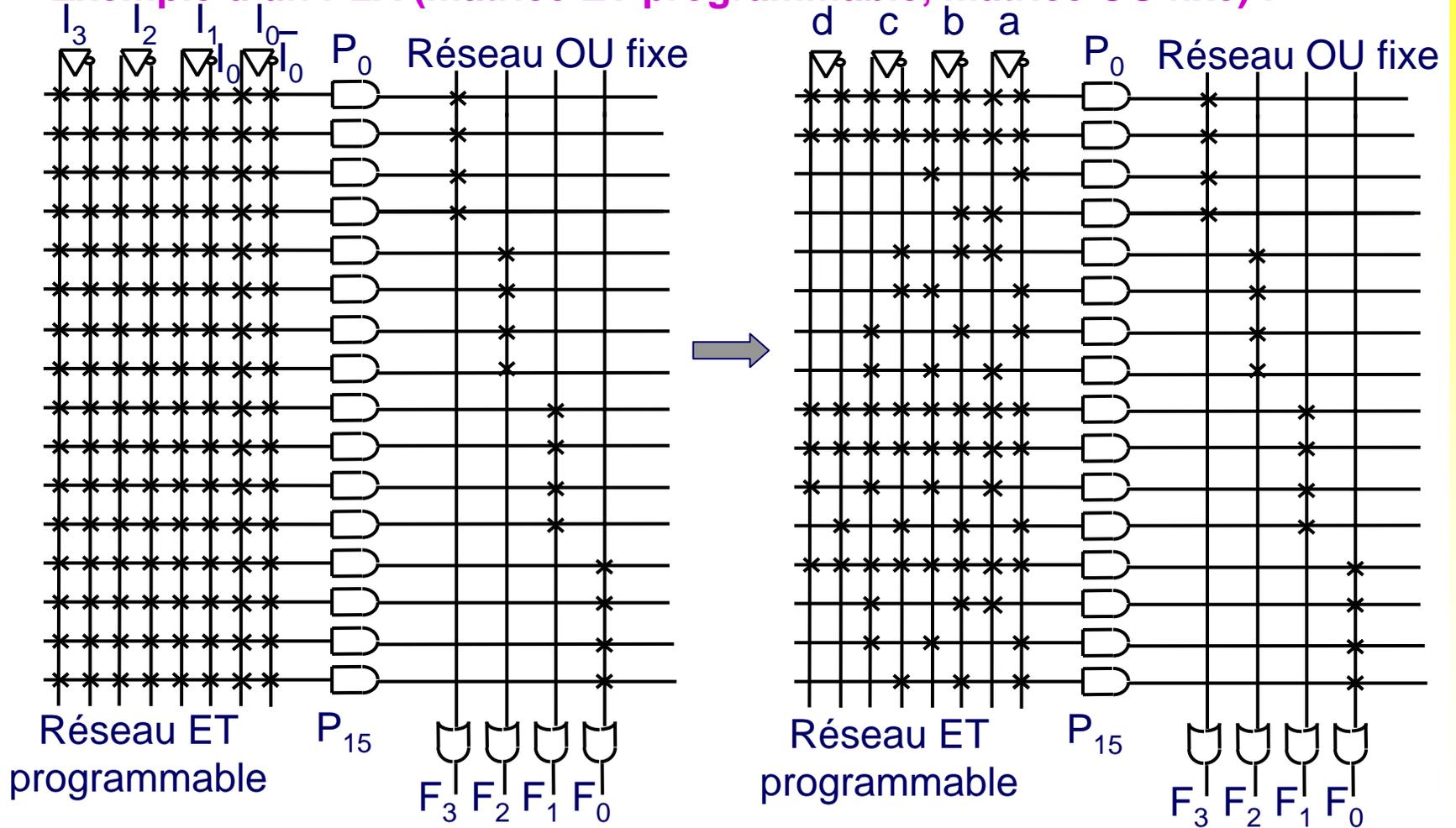


Initialement, tous les contacts sont établis et la programmation consiste à faire sauter les fusibles à l'aide d'une surtension.



II – Les circuits programmables

Exemple d'un PLA (matrice ET programmable, matrice OU fixe) :



$$F_3 = P_0 + P_1 + P_2 + P_3$$

$$F_2 = P_4 + P_5 + P_6 + P_7$$

$$F_1 = P_8 + P_9 + P_{10} + P_{11}$$

$$F_0 = P_{12} + P_{13} + P_{14} + P_{15}$$

$$F_0 = \bar{a}\bar{b}\bar{c} + \bar{a}bc + a\bar{b}\bar{c}$$

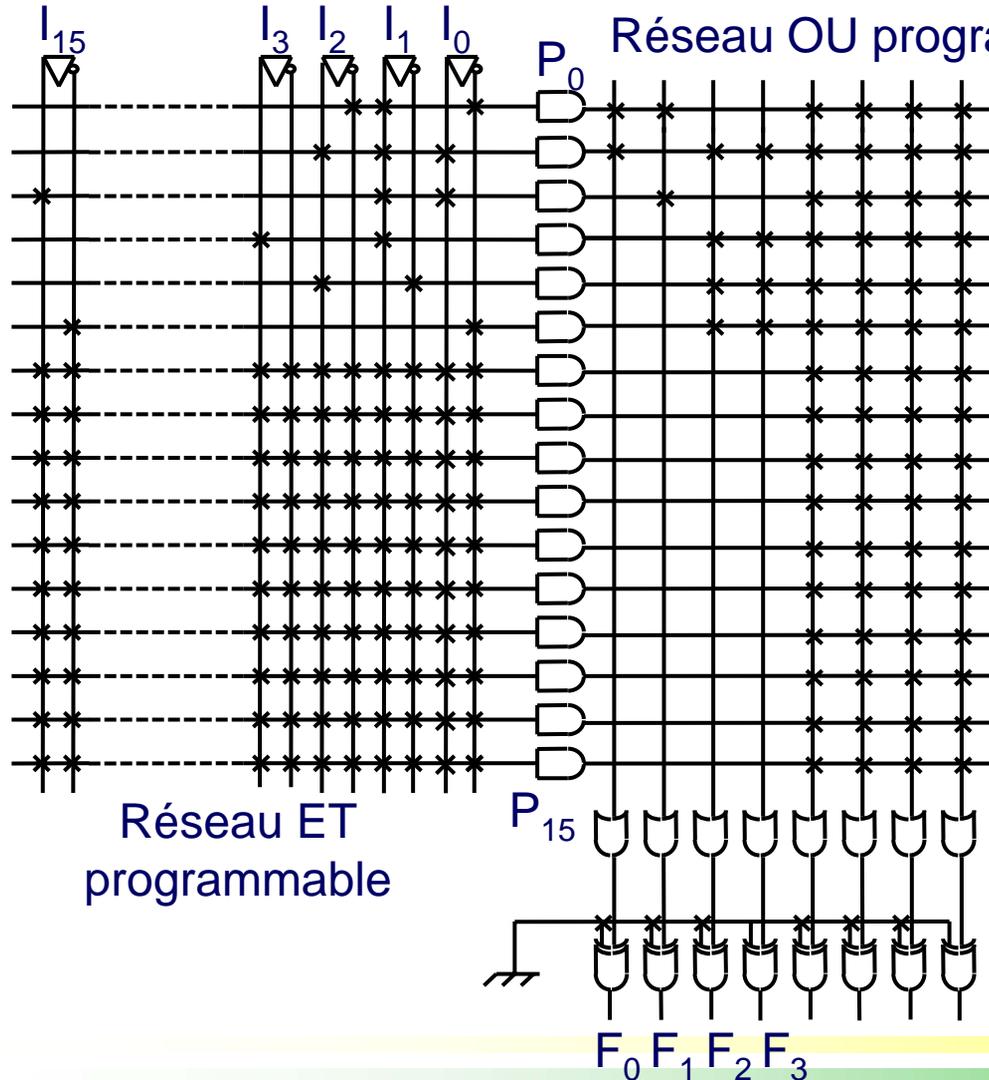
$$F_1 = \bar{a}\bar{b}cd + abcd$$

$$F_2 = abc + \bar{a}bc + a\bar{b}c + abc$$

$$F_3 = \bar{a}b + a\bar{b}$$

II – Les circuits programmables

Exemple d'un FPLA (matrice ET et OU programmables) :



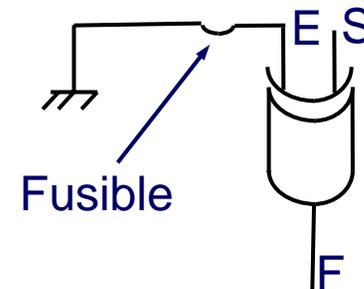
$$F_0 = \bar{I}_0 I_1 \bar{I}_2 + I_0 I_1 I_2$$

$$F_1 = I_0 I_1 I_{15} + \bar{I}_0 I_1 \bar{I}_2$$

$$F_2 = I_1 I_3 + \bar{I}_1 I_2 + \bar{I}_0 \bar{I}_{15} + I_0 I_1 I_2$$

$$F_3 = \bar{F}_2$$

Inverseur commandé :



$$F = \bar{S} E + S \bar{E}$$

Avec fusible $\Rightarrow E = 0 \Rightarrow F = \bar{S}$

Sans fusible $\Rightarrow E = 1 \Rightarrow F = S$

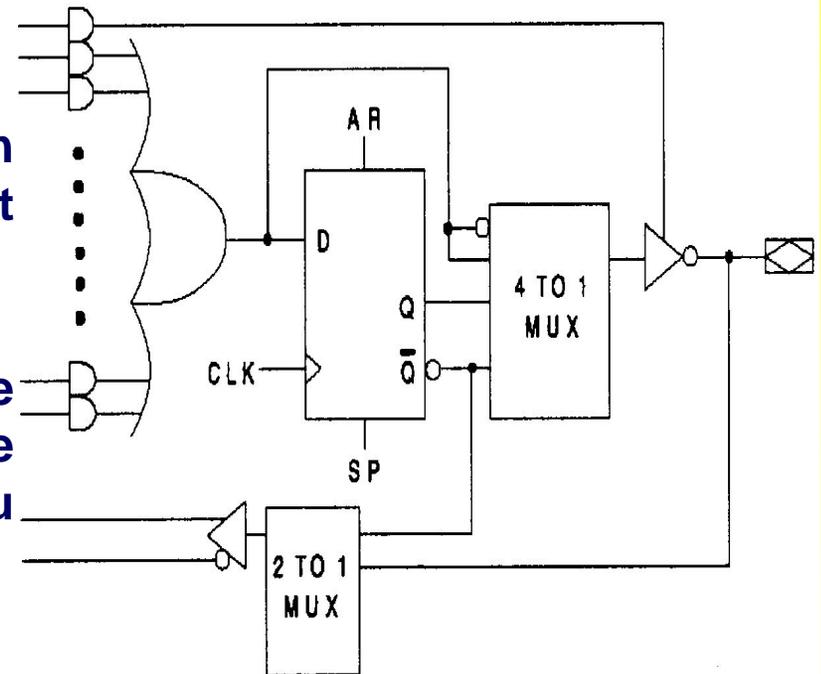
II – Les circuits programmables

Les circuits PAL sont constitués d'un réseau PLA se terminant par une structure plus ou moins complexe :

Exemple : Le circuit PAL 22V10.

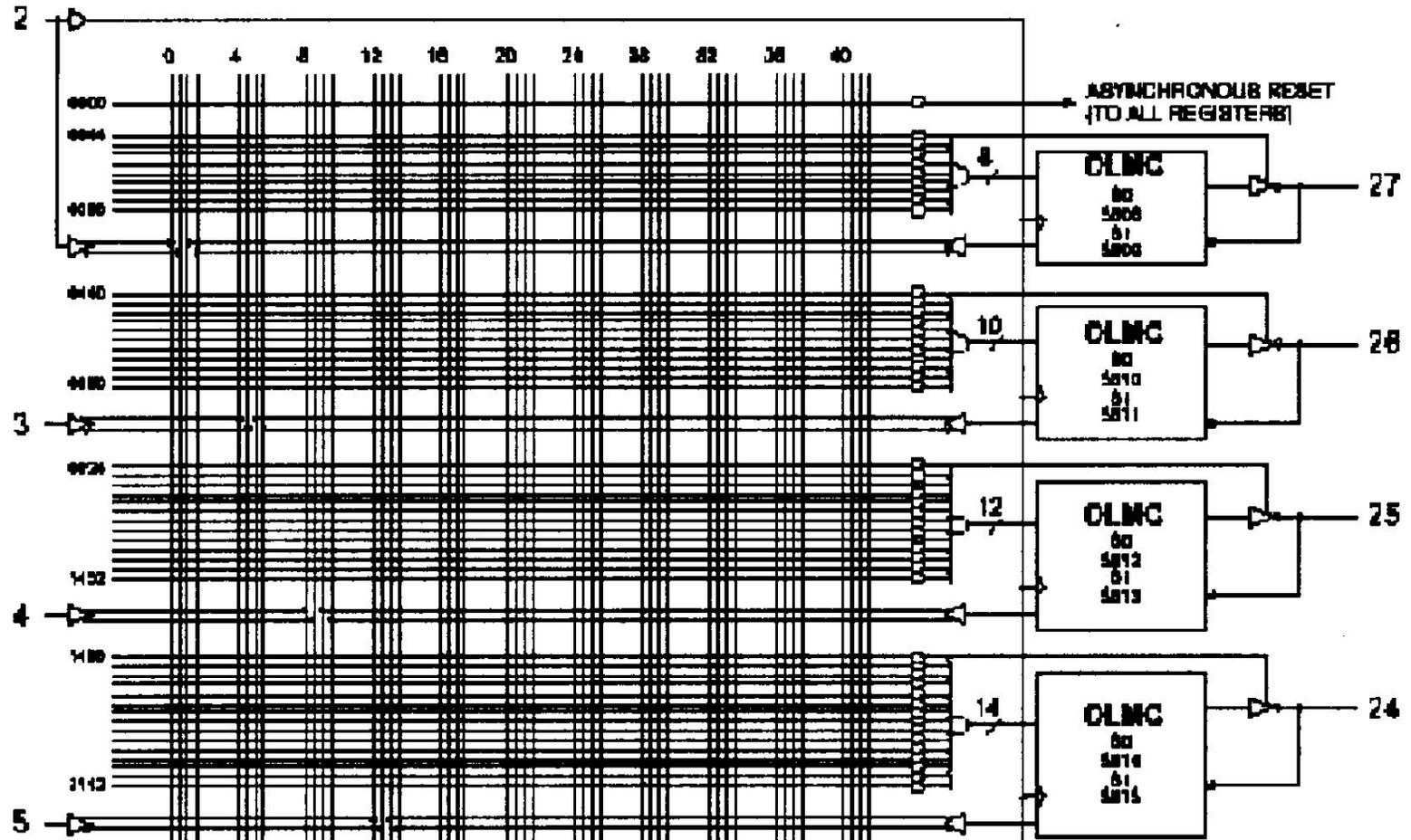
Le multiplexeur 4 vers 1 met en circuit ou non la bascule D et permet d'inverser ou pas les signaux.

Le multiplexeur 2 vers 1 permet de réinjecter soit la sortie, soit l'entrée du buffer de sortie vers le réseau programmable.



II – Les circuits programmables

Structure générale du 22V10 :



.....

II – Les circuits programmables

Décodage des références :

PAL	CE	XX	AB	YY	C	ZZ	DEF
-----	----	----	----	----	---	----	-----

PAL : Circuit PAL.

CE : Version MOS

XX : Nombre d'entrées.

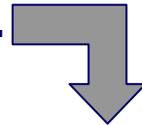
AB : Structure de sortie.

YY : Nombre de sorties.

C : Consommation.

ZZ : Vitesse.

DEF : Boîtier, gamme.



Type	Désignation	Type	Désignation
H	Combinatoire à logique positive	R	A registre
L	Combinatoire à logique négative	RA	A registre asynchrone
X	Registre ET OU exclusif	V	Versatile (multiplex.)

Exemples : Les circuit PAL 16R8H15PC ou PAL 22V10.

II – Les circuits programmables

II.2 – Les circuits GAL :

A la base, les technologies de connexions des circuits PAL ne permettent pas leurs reprogrammations.

La technologie des circuits GAL est appelée MOS à grille flottante. Ce sont donc des circuits programmables et effaçables électriquement.

Références du circuit :

GAL XX AB YY

XX : Nombre d'entrées.

AB : Structure de sorties.

YY : Nombre de sorties.

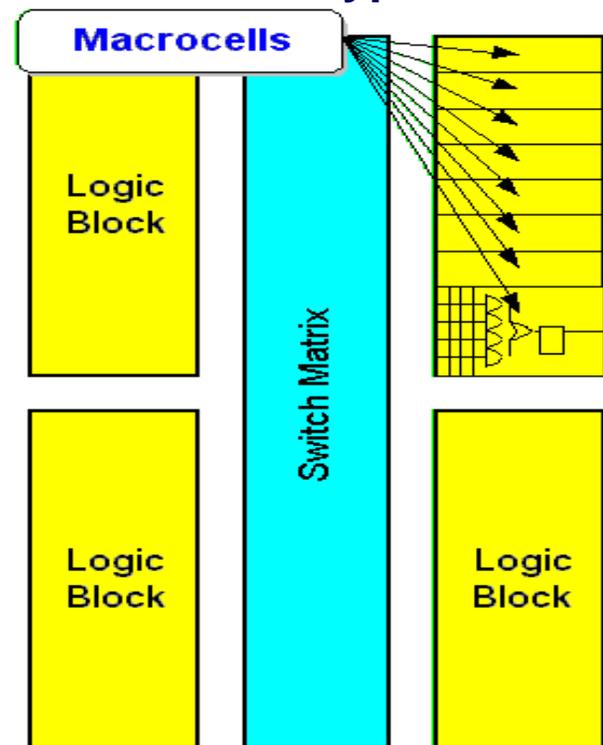
GAL est en fait une marque déposée de la société Lattice et ce genre de circuits devrait être appelé PAL MOS.

II – Les circuits programmables

II.3 – Les circuits CPLD :

Un CPLD est approximativement une association de plusieurs PLD connectés entre eux par une matrice d'interconnexions. Le taux d'intégration est donc plus élevé que pour un circuit de type PAL.

La technologie de connexion utilisée est généralement de l'EEPROM ou de l'EPROM FLASH ce qui rend ces circuits effaçables et programmables électriquement.

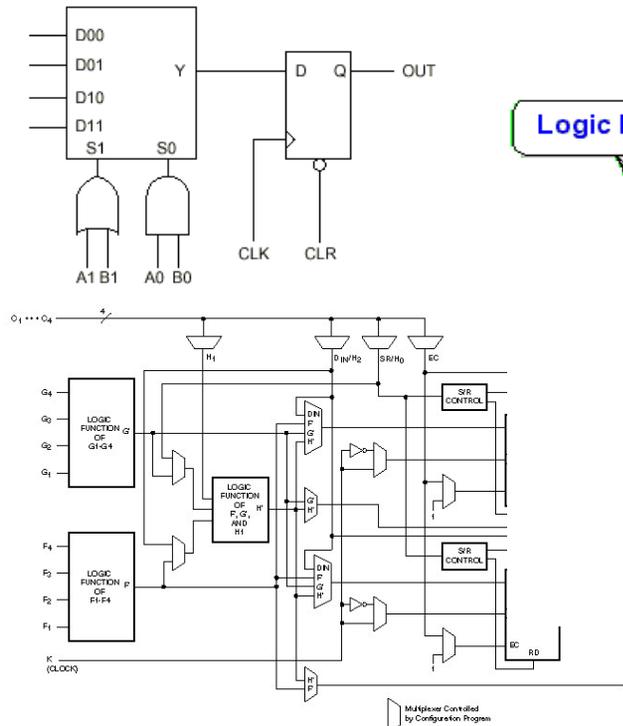


II – Les circuits programmables

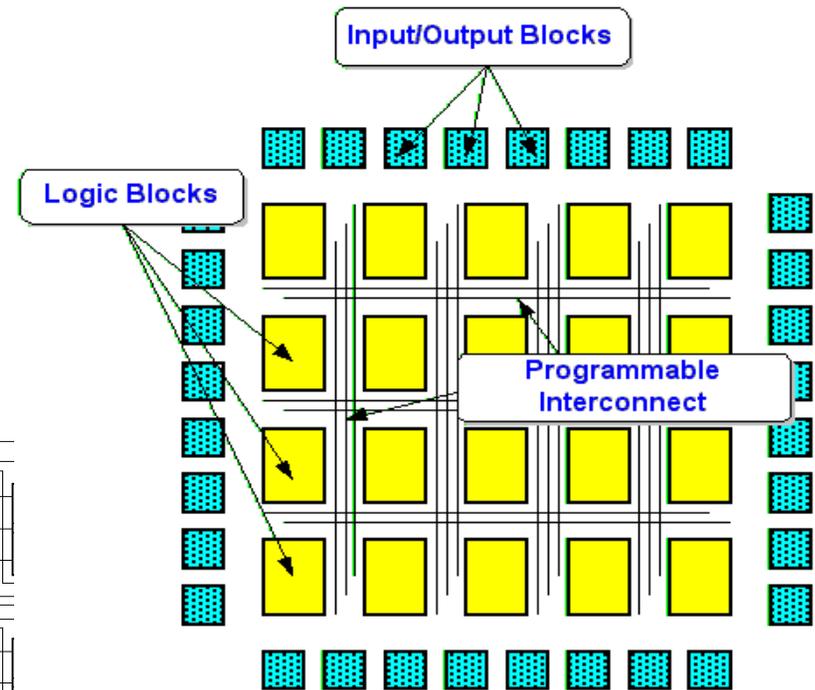
II.4 – Les circuits FPGA :

Ce sont les circuits qui ont le taux d'intégration le plus élevé (supérieur à 100 000 portes). Les blocs logiques sont plus nombreux mais plus simples que pour les CPLD. Ils nécessitent donc d'importantes ressources de routage.

Bloc logique Actel.



Bloc logique Xilinx.



II – Les circuits programmables

Deux technologies de connexion sont possibles :

- **FPGA SRAM** : La configuration des connexion est contenue dans une RAM et peut être modifiée lors du fonctionnement. Par contre, cette configuration est perdue hors alimentation.

- **FPGA à anti fusibles** : Ces circuits sont programmables électriquement mais non effaçables. La programmation consiste à enlever de la matière isolante. La configuration est maintenue sans alimentation.