REPUBLIQUE TUNISIENNE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

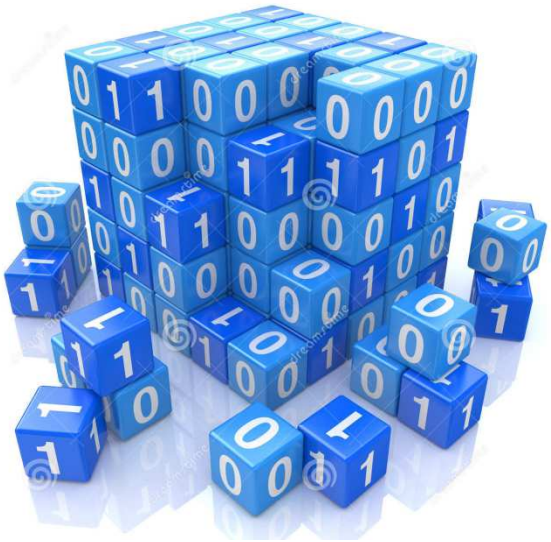
INSTITUT SUPERIEUR DES ÉTUDES TECHNOLOGIQUES DE KAIROUAN

DÉPARTEMENT GÉNIE ÉLECTRIQUE

Mr. SAMIR ARFA

COURS SYSTEMES LOGIQUES : S1+S2

GETC1



2018

Avant propos

Ce cours correspond aux programmes de l'enseignement supérieur de la matière systèmes logiques du réseau des Instituts Supérieurs des Etudes Technologiques (ISETs) pour lés étudiants de la 1ère année tronc commun génie électrique (S1+ S2).

Ce document est structuré en six chapitres qui couvrent le programme officiel de la matière.

Fiche matière

FICHE MATIERE

Unité d'enseignement : Traitement de données 1 — Code UE : 1.4

ECUE n° 2 : Systèmes logiques 1. — Code ECUE : 1.4.2

Domaine de formation : **Sciences appliquées et technologie**	Mention :	GE
	Spécialité	
Domaine et parcours : **Licence appliquée GE**	Semestre	S1.
Parcours : Tronc commun.		

Nombre d'heures/semestre		Coefficient	Crédits	Système d'évaluation
Cours integre	TP			
21	0	2	1.5	Continu

OBJECTIFS DE L'ENSEIGNEMENT :

- Connaître les différentes structures de la logique combinatoire.

PRE - REQUIS :

Notions élémentaires de l'algèbre binaire.

Contenu théorique :

- Notions sur les systèmes de numérisation et les codes ;
- Les fonctions logiques :
  - Rappel des fonctions élémentaires,
  - Théorèmes de DEMORGAN,
  - Formes canoniques de Shannon,
  - Simplification des expressions logiques algébriquement et graphiquement (tableau de KARNAUGH).
- Systèmes de logique combinatoires
- Methode de synthèse appliquée à des exemples à caractères industriels ;
- Fonctions intégrés de logique combinatoire :
  (Codage, decodage, transcodage, circuits de transfert d'informations, circuits arithmétiques).
- Synthèse des systèmes de logiques combinatoires par l'utilisation des fonctions intégrées
- Technologie des circuits intégrés numériques : (Familles des circuits intégrés, TTL, CMOS, niveau d'intégration, boitier « package », Alimentation, Niveaux logiques, Retards de propagations)

PROLONGEMENT :

- Les systèmes séquentiels

Bibliographie :

Réhabilitation LAGE_TC 2013-2017 Page 61

Table des matières :

Avant propos 3

Fiche matière 4

Semestre 1 6

Chapitre 1: Systèmes de numérations 7

Chapitre 2 : portes logiques et algèbre de Boole 19

Chapitre 3: circuit logiques combinatoires 27

Semestre 2 40

Chapitre 4 : bascules : R.S-R.S.H -J.K-D- T 41

Chapitre 5: compteurs et décompteurs 47

Chapitre 6: Les Registres 56

Bibliographie: 65

Sites Web: 65

Semestre 1

Chapitre 1: Systèmes de numérations

I. Introduction :

On utilise généralement les systèmes de numération suivants: **Le système décimal, le système binaire, le système octal, le système hexadécimal.**

D'une manière générale, on peut exprimer un nombre (N) dans le système de numération de base B:

$$(N)_B = (a_{n-1}a_{n-2}...a_1a_0a_{-1}...a_{-m})_B = (a_{n-1}B^{n-1} + a_{n-2}B^{n-2} + ... + a_1B^1 + a_0B^0 + a_{-1}B^{-1} + ... + a_{-m}B^{-m})_B$$

Cette forma s'appelle la forme polynomiale.

Exemple : $(2580,75)_{10}$:

$$2580,75 = 2.10^3 + 5.10^2 + 8.10^1 + 0.10^0 + 7.10^{-1} + 5.10^{-2} = 2000 + 500 + 80 + 0 + 0,7 + 0,05$$

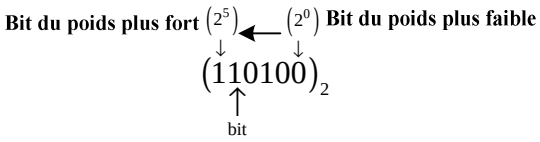
II. Différentes types de système de numération

1. Système Binaire : $B = 2$ et $a_i \subset \{0;1\}$

Dans le système binaire $(B = 2)$, pour exprimer n'importe quelle valeur on utilise uniquement deux symboles : {0, 1}.

Exemple :

$$(110100)_2 = 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0 = (52)_{10} \quad (110100)_2$$



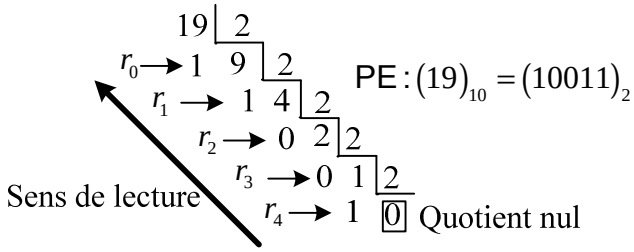
❖ **Conversion décimal $\rightarrow$ binaire :**

Exemple 1: cas d'un nombre réel $(19,25)_{10}$

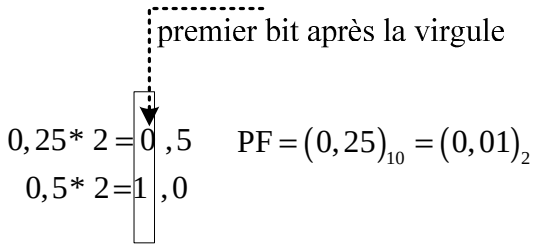
Un nombre réel est constitué de deux parties :

$$\underbrace{19}_{\text{Partie entière(PE)}},\underbrace{25}_{\text{Partie fractionnelle(PF)}}$$

- La partie entière est transformée en effectuant des divisions successives.



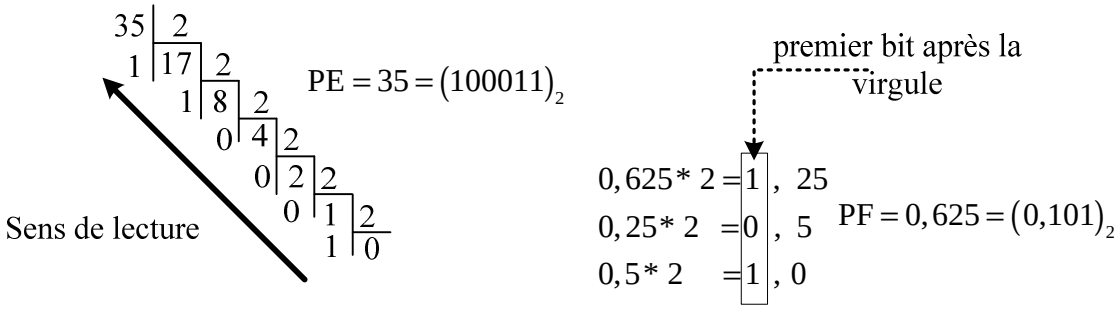
- La partie fractionnelle est transformée en effectuant des multiplications successives par 2.



Finalement : $(19,25)_{10} = (10011,01)_2$

Exemple 2 :

$35,625 = (?)_2$



Donc: $35,625 = (100011,101)_2$

❖ **Conversion binaire $\rightarrow$ décimal :**

Un nombre dans la base 2 peut être écrit aussi sous la forme polynomiale :

$(10011,01)_2 = (?)_{10}$

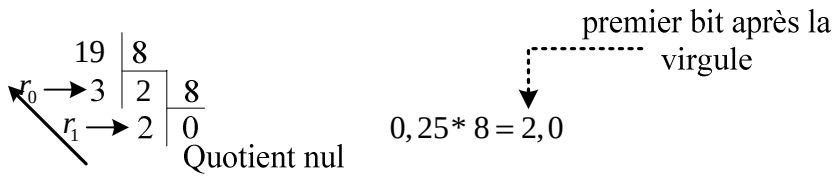
Cette conversion est assez simple puisque il suffit de faire le développement en polynôme de ce nombre dans la base B, et de faire la somme par la suite.

$$n=5 \Rightarrow (10011)_2 = 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0 + 0*2^{-1} + 1*2^{-2}$$
$$= 16 + 0 + 0 + 2 + 1 + 0 + \frac{1}{2^2} = 19,25$$

2. Système octal : $B = 8$ et $a_i \subset \{0;1; .....;7\}$

Exemple : $(23,2)_8 = 2*8^1 + 3*8^0 + 2*8^{-1} = (19,25)_{10}$

❖ **Conversion décimal $\rightarrow$ octal** $(19,25)_{10}$ **:**



D'ou $(19)_{10} = (23,2)_8$

❖ **Conversion octal $\rightarrow$ décimal**

Exemple : Convertir le nombre octal $(23,2)_8$ dans la base 10 :

$(23,2)_8 = 2.8^1 + 3.8^0 + 2.8^{-1} = (19,25)_{10}$

❖ **Conversion octal $\rightarrow$ binaire :**

En octal, chaque symbole de la base s'écrit sur 3 bits en binaire.

L'idée de base est de remplacer chaque symbole dans la base octal par sa valeur en binaire sur 3 bits (éclatements sur 3 bits).

Exemple : $(23,2)_8$

Chiffres octaux	2	3	,2
Equivalent binaire	010	011	,010

D'ou $(23)_8 = (010011,010)_2$

❖ **Conversion : Binaire $\rightarrow$ Octal**

L'idée de base est de faire des regroupements de 3 bits à partir du poids faible, par la suite remplacer chaque regroupement par la valeur octal correspondante.

Remarque : le regroupement se fait de droit à gauche pour la partie entière et de gauche à droite pour la partie fractionnelle.

Exemple : $(10011,01)_2 = (?)_8$

$(10011,01)_2 = (\underline{010}\,\underline{011},\underline{010})_2 = (\underline{23},\underline{2})_8$

3. Système hexadécimal : $B = 16$ et $a_i \subset \{0,1,.....9, A, B, C, D, E, F\}$

Exemple : $(13,4)_{16} = 1*16^1 + 3*16^0 + 4*16^{-1} = (19,25)_{10}$

❖ **Conversion décimal $\rightarrow$ hexadécimal**

❖ **Exemple :** Soit à convertir le nombre suivant en hexadécimal: $(19,25)_{10} = (?)_{16}$

$$\begin{array}{lll} & 19 & \lfloor 16 \\ r_0 \rightarrow & 3 & 1 \lfloor 16 \\ r_1 \rightarrow & 1 & \boxed{0} \text{ Quotient nul} \end{array}$$

D'ou $(19)_{10} = (13)_{16}$

$0,25 * 16 = 4,0$ premier bit après la virgule

$$(19,25)_{10} = (13,4)_{16}$$

❖ **Conversion hexadécimal $\rightarrow$décimal:** $(13,4)_{16}$

$(13,8)_{16} = 1*16^1 + 3*16^0 + 4*16^{-1} = 19,5$

❖ **Conversion hexadécimal $\rightarrow$binaire :**

En Hexadécimal chaque symbole de la base s'écrit sur 4 bits, l'idée de base est de remplacer chaque symbole par sa valeur en binaire sur 4 bits (éclatement sur 4 bits).

- **Exemple :** $(13,4)_{16}$

Chiffres hexadécimals	1	3	,4
Equivalent binaire	0001	0011	,0100

$(13,4)_{16} = (10011,01)_2$

- **Conversion binaire $\rightarrow$ hexadécimal**

L'idée de base est de faire des regroupements de 4 bits à partir du poids faible, par la suite remplacer chaque regroupement par la valeur hexadécimale correspondante.

- **Exemple :** $(10011,01)_2 = (?)_{16}$

$(10011,01)_2 = (\underline{0001}\,\underline{0011},\underline{0100})_2 = (\underline{13},\underline{4})_{16}$

- **Conversion hexadécimal $\rightarrow$ octal: (Passage par la base 2)**

Exemple:

Convertir le nombre hexadécimal $(13,4)_{16} = (?)_8$

$$(13,4)_{16} = (\underline{0001}\,\underline{0011},\underline{0100})_2 = (10011,01)_2$$
$$= (\underline{010}\,\underline{011},\underline{010})_2 = (\underline{2}\underline{3},\underline{2})_8$$

III. Les codes

En binaire, on distingue trois principaux systèmes de codage :

- Code binaire naturel : (1-2-4-8-…);
- Code binaire réfléchi : (code GRAY);
- Code décimal binaire (D.C.B) ou (B.C.D).
- Code alphanumériques.
- Code à barres.

1. Code binaire naturel

Le code binaire naturel est le code dans lequel on exprime un nombre selon le système de numération binaire.

Exemple : $n = 4\text{bits} \Rightarrow 2^n = 16$: 16 combinaisons :

Décimal	Binaire naturel			
N	2^3	2^2	2^1	2^0
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
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

2. Binaire réfléchi (code Gray)

Pour le binaire réfléchi à chaque augmentation d'une unité du chiffre décimal, on a un seul bit du nombre binaire équivalent qui change de valeur par rapport au nombre binaire précédent.

Exemple : $n = 4\text{bits} \Rightarrow 2^n = 16$: 16 combinaisons :

Décimal	Binaire naturel				Binaire réfléchi			
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

- **Conversion binaire $\rightarrow$ réfléchi (Gray)**

La conversion d'un nombre binaire en binaire réfléchi, s'effectue suivant la méthode pratique suivante :

- Ecrire le nombre binaire à convertir.
- Reproduire le bit binaire de plus fort bit, pour obtenir celui du code réfléchi.
- Le reste des bits du code de réfléchi sont obtenus en additionnant les bits binaires deux à deux, à partir de la gauche, sans tenir compte de la retenue.

Exemple 1:

Convertir en code réfléchi le nombre binaire suivant : $(110100)_2 = (?)_G$

$$\begin{array}{lcccccc} \text{Code binaire:} & 1 \xrightarrow{+} & 1 \xrightarrow{+} & 0 \xrightarrow{+} & 1 \xrightarrow{+} & 0 \xrightarrow{+} & 0 \\ & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow \\ \text{Code réfléchi:} & 1 & 0 & 1 & 1 & 1 & 0 \end{array}$$

$(110100)_2 = (101110)_{Gray}$

❖ **Conversion binaire réfléchi $\rightarrow$ binaire naturel**

La conversion d'un nombre codé en réfléchi en un nombre binaire, s'effectue comme suit :

- Ecrire le nombre en réfléchi à convertir.
- Reproduire le bit de plus fort poids du nombre réfléchi, pour obtenir celui du code de binaire.
- Le reste des bits du code binaire sont obtenus en additionnant les bits sans retenue, deux à deux, à partir de la gauche, le bit du rang i du code binaire avec celui du rang i-1, est ainsi de suite.

Exemple :

Convertir en code binaire le nombre codé en Gray suivant : $(101110)_{Gray}$

$$\begin{array}{lcccccc} \text{Code réfléchi:} & 1 & 0 & 1 & 1 & 1 & 0 \\ & \Downarrow \nearrow + & \Downarrow \nearrow + & \Downarrow \nearrow + & \Downarrow \nearrow + & \Downarrow \nearrow + & \Downarrow \\ \text{Code binaire:} & 1 & 1 & 0 & 1 & 0 & 0 \end{array}$$

3. Code Décimal codé binaire (B.C.D)

Dans ce code, le nombre décimal compris entre 0 et 9 est codé chiffre par chiffre par des combinaisons de quatre bit.

Décimal	BCD (8421)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Exemple : coder en BCD le nombre décimal 238 :

$$\begin{array}{cccl} 2 & 3 & 8 & \leftarrow \text{en décimal} \\ 0010 & 0011 & 1000 & \leftarrow \text{en BCD} \end{array}$$

$$238_{(10)} = (\underline{0010}\,\underline{0011}\,\underline{1000})_{BCD}$$

4. Les codes alphanumériques

- 26 caractères pour les lettres minuscules
- 26 caractères pour les lettres majuscules
- 10 pour les chiffres décimaux
- 30 pour les caractères spéciaux (><%§+_-...)

Table des caractères ASCII

Table ASCII standard (codes de caractères de 0 à 127)

000		(nul)	016	►	(dle)	032	sp	048	0	064	@	080	P	096	`	112	p
001	☺	(soh)	017	◄	(dc1)	033	!	049	1	065	A	081	Q	097	a	113	q
002	☻	(stx)	018	↕	(dc2)	034	"	050	2	066	B	082	R	098	b	114	r
003	♥	(etx)	019	‼	(dc3)	035	#	051	3	067	C	083	S	099	c	115	s
004	♦	(eot)	020	¶	(dc4)	036	$	052	4	068	D	084	T	100	d	116	t
005	♣	(enq)	021	§	(nak)	037	%	053	5	069	E	085	U	101	e	117	u
006	♠	(ack)	022	▬	(syn)	038	&	054	6	070	F	086	V	102	f	118	v
007	•	(bel)	023	↨	(etb)	039	'	055	7	071	G	087	W	103	g	119	w
008	◘	(bs)	024	↑	(can)	040	(	056	8	072	H	088	X	104	h	120	x
009		(tab)	025	↓	(em)	041	)	057	9	073	I	089	Y	105	i	121	y
010		(lf)	026		(eof)	042	*	058	:	074	J	090	Z	106	j	122	z
011	♂	(vt)	027	←	(esc)	043	+	059	;	075	K	091	[	107	k	123	{
012	♀	(np)	028	∟	(fs)	044	,	060	<	076	L	092	\	108	l	124	|
013		(cr)	029	↔	(gs)	045	-	061	=	077	M	093	]	109	m	125	}
014	♫	(so)	030	▲	(rs)	046	.	062	>	078	N	094	^	110	n	126	~
015	☼	(si)	031	▼	(us)	047	/	063	?	079	O	095		111	o	127	⌂

5. Code à barres :

Le code à barre qui figure sur la plupart des emballages des produits de consommation courante est la fiche d'identité du produit. Ce code imprimé sur l'emballage se présente sous la forme d'une étiquette rectangulaire collée. Il est composé de barres et d'espaces large ou étroits dont le nombre correspond à un ensemble de donnés numériques ou alphanumériques.

Le marquage comporte un certain nombre de barres verticales ainsi que 13 chiffres :

- Le 1er chiffre désigne le pays d'origine : 3 = France, 4 = Allemagne, 0 = U.S.A, Canada etc.

- Les cinq suivants sont ceux du code « fabricant »,

- Les six autres sont ceux du code de l'article,

- Le dernier étant une clé de contrôle.

IV. Notion d'arithmétique binaire

1. Opérations arithmétiques

addition	retenue		**soustraction**	retenue
0 + 0 = 0	**0**		**0 - 0 = 0**	**0**
0 + 1 = 1	**0**		**0 - 1 = 1**	**1**
1 + 0 = 1	**0**		**1 - 0 = 1**	**0**
1 + 1 = 0	**1**		**1 - 1 = 0**	**0**

La méthode d'addition des nombres binaires signés, consiste à écrire les nombres positifs en binaire avec un bit de signe 0, et à remplacer les nombres négatifs par leur complément à 2 avant addition.

Si le résultat est positif, il est en notation binaire, s'il est négatif il est en notation complément à 2.

2. Représentation en complément à 2 :

- En notation en complément à 2 et avec n bits, on représente les nombres signés compris dans l'intervalle $[-2^n, 2^{n-1}-1]$.
- Pour revenir d'un nombre complémenté à 2 au nombre binaire initial, il faut de nouveau complémenter à 2.

$$N^* = \overline{N} + 1$$

Exemple :

Le complément à 2 sur n=4 bits.

Décimal	Nombre binaire en complément à 2 (N^*)
0	0000
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001

a. Addition de deux nombres binaires :

a) **Exemple :** sur n=4 bits (intervalle $[-8,+7]$)

$7+(-4)=3$

- Complément à 2 de 4 c.à.d. (-4)

$N=4=0100$:

$N^{*}=\overline{N}+1=1011_{1}+1=1100$

$$\begin{array}{rl} & 0111 \leftarrow 7 \\ + & 1100 \leftarrow \text{Complément à 2 de 4} \\ \hline & \underset{\uparrow}{1}0011 \leftarrow 3 \end{array}$$

Bit à rejéter

b) **Exemple :** $-9-4=-13$

Sur n=5 bits : (intervalle $[-16,+15]$)

- Complément à 2 de 9 c.à.d. (-9):

$N=9=(01001)_{2}$:

$N^{*}=\overline{N}+1=10110+1=10111$

- Complément à 2 de 4 c.à.d. (-4)

$N=4=00100$:

$N^{*}=\overline{N}+1=11011_{1}+1=11100$

$$\begin{array}{rl} & 10111 \leftarrow \text{Complément à 2 de 9} \\ + & 11100_{11} \leftarrow \text{Complément à 2 de 4} \\ \hline & \underset{\uparrow}{1}10011 \leftarrow \text{Complément à 2 de 13} \end{array}$$

Bit à éliminer

b. Multiplication binaire :

On multiplie les nombres binaires de la même façon qu'on multiplie les nombreux décimaux. Si les nombreux à multiplier sont négatifs, ce sont leurs compléments à 2 qui sont pris en compte avant la multiplication.

Si l'un des nombres est négatif, on prend son complément à 2 avant la multiplication, le résultat est le complément à 2 du produit cherché.

Remarque importante : il faut toujours vérifier que le résultat tient dans l'intervalle $\left[-2^{n-1},\ 2^{n-1}-1\right]$ avec des nombres signés.

Exemple : effectuer l'opération suivante :

$$\begin{array}{r} 011 \\ \times\ 010 \\ \hline 000 \\ 011 \\ 000 \\ \hline 000110 = 110 \end{array}$$

c. Division binaire $\left(a/b\right)$:

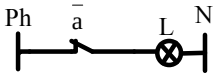
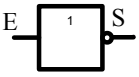
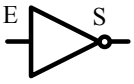
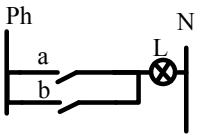
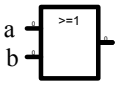
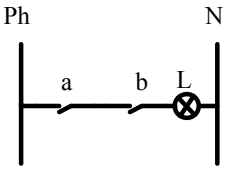
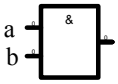
La méthode de division binaire est identique à celle de la division de deux nombres décimaux de gauche vers la droite.

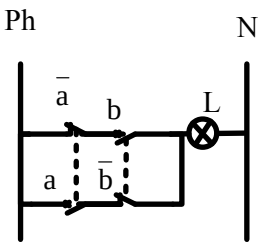
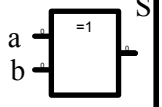
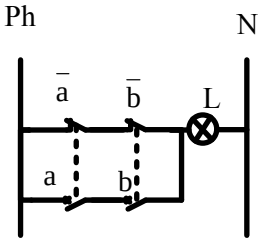
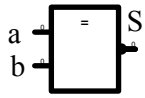
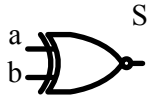
Exemple :

$$\begin{array}{r|l} 11 & 4 \\ \cline{2-2} 30 & 2,75 \\ 28 & \\ \cline{1-1} 20 & \\ 0 & \end{array} \qquad \begin{array}{r|l} 1011 & 100 \\ \cline{2-2} 100 & 10,1100 \\ \cline{1-1} 0011 & \\ 000 & \\ \cline{1-1} 0110 & \\ 100 & \\ \cline{1-1} 0100 & \\ 100 & \\ \cline{1-1} 0000 & \end{array}$$

Chapitre 2 : portes logiques et algèbre de Boole

I. Portes logiques de bases

Fonction	Schéma a contact	Symboles		Table de vérité et équation logique	Conclusion
		Francais	American		
NON/ NOT			Circuit intégré : 74LS04 	E S 0 1 1 0 $S = \bar{E}$	La sortie est égale à 1 si, et seulement si, l'entrée est à l'état 0.
SOMME /OU/OR			Circuit intégré- 74LS32	b a S 0 0 0 0 1 1 1 0 1 1 1 1 $S = a + b$	La sortie est égale à 1 si, et seulement si, une ou plusieurs entrées sont à l'état 1.
PRODUI T/ET/ AND			Circuit intégré- 74LS08	b a S 0 0 0 0 1 0 1 0 0 1 1 1 $S = a.b$	La sortie est égale à 1 si, et seulement si, toutes les entrées sont à l'état 1.

OU-exclusif /XOR			Circuit intégré- 74LS86	$S = a \oplus b$
NI-exclusif / XNOR(coïncidence)				$S = a \odot b$

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

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

Propriétés de XOR et XNOR :

Les portes XOR et XNOR vérifient les propriétés suivantes :

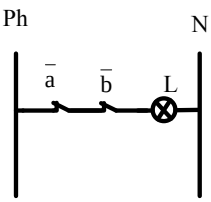
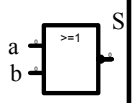
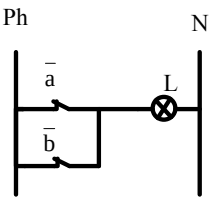
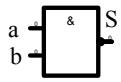
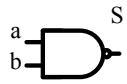
$$\left.\begin{array}{l} x \oplus y = y \oplus x \\ x \odot y = y \odot x \end{array}\right\} \text{ : } \textbf{Commutativité}$$

$$\left.\begin{array}{l} x \oplus (y \oplus z) = (x \oplus y) \oplus z \\ x \odot (y \odot z) = (x \odot y) \odot z \end{array}\right\} \text{ : } \textbf{Associativité}$$

$$\boxed{\begin{array}{l} x \oplus 1 = \bar{x} \\ x \odot 1 = x \end{array}} \qquad \boxed{\begin{array}{l} x \oplus 0 = x \\ x \odot 0 = \bar{x} \end{array}}$$

II. Fonction logiques universelles

Une fonction est dite universelle si et seulement si elle permet de réaliser toutes les opérateurs logiques de base.

Fonction	Schéma a contact	Symboles		Table de vérité et équation logique	Conclusion
		Francais	American		
NON-OU/NI/NOR			Circuit intégré-74LS02 	b a S 0 0 1 0 1 0 1 0 0 1 1 0 $S=\bar{a}\bar{b}=\overline{a+b}$ $=a\downarrow b$	La sortie est égale à '1', si et seulement si, toutes les entrées sont à l'état '0'.
NI-exclusif / XNOR(coïncidence)			Circuit intégré-74LS00 	b a S 0 0 1 0 1 1 1 0 1 1 1 0 $S=a/b$	La sortie est à l'état 0, si et seulement si, toutes les entrées sont à l'état 1.

3. Propriétés des fonctions universelles

Fonction NI/NOR	Fonction NAND/ON
$\bar{a}\bar{b}=\bar{b}\bar{a}\Rightarrow \quad a\downarrow b=b\downarrow a$ **Commutative**	$a/b=b/a \quad car: \bar{a}+\bar{b}=\bar{b}+\bar{a}$ **Commutative**
$a\downarrow b\downarrow c\neq(a\downarrow b)\downarrow c\neq a\downarrow(b\downarrow c)$ **Non Associative**	$(a/b)/c\neq a/(b/c)\neq a/b/c$ **Non Associative**

<u>Vérification</u> :

- $a\downarrow b\downarrow c=\overline{a+b+c}=\bar{a}.\bar{b}.\bar{c}$
- $(a\downarrow b)\downarrow c=\overline{(\overline{a+b})+c}=\overline{(\overline{a+b})}\bar{c}=(a+b)\bar{c}=a\bar{c}+b\bar{c}$
- $a\downarrow(b\downarrow c)=\overline{a+(\overline{b+c})}=\bar{a}.\overline{(\overline{b+c})}=\bar{a}.(b+c)=\bar{a}b+\bar{a}c$

Donc : $\bar{a}.\bar{b}.\bar{c}\neq a\bar{c}+b\bar{c}\neq\bar{a}b+\bar{a}c$

III .Propriétés de l'algèbre de Boole

Pour effectuer tout calcul booléen on utilise un ensemble de propriétés:

Commutativité	$a.b = b.a$ $a + b = b + a$
Associativité	$a + b + c = a + (b + c) = (a + b) + c$ $a.b.c = a(b.c) = (a.b).c$
Distributivité	$a.(b + c) = a.b + a.c$ $(a + b)(a + c) = a + b.c$
Idempotence	$a + a + a + a = a$ $a.a...a = a$

- $a + 1 = 1$
- $a + \bar{a} = 1$
- $a\bar{a} = 0$
- $a + \bar{a}b = a + b$ $\qquad \left(a + b = a + b\left(a + \bar{a}\right) = a + ab + \bar{a}b = a + \bar{a}b\right)$

Théorèmes de DE MORGAN

► **Théorème 1 :** Le complément d'un produit de variables est égal à la somme des compléments des variables : $\overline{a.b} == \bar{a} + \bar{b}$

► **Théorème 2 :** Le complément d'une somme de variables est égal au produit des compléments des variables : $\overline{a + b} = \bar{a}.\bar{b}$

Remarque :

Le théorème de DE MORGAN peut être généralisé à plusieurs variables :

$$\overline{a + b + c + ... + z} = \bar{a}\,\bar{b}\,\bar{c}...\bar{z}$$

$$\overline{abc...z} = \bar{a} + \bar{b} + \bar{c} + ... + \bar{z}$$

IV .Simplification des équations logiques

Minimiser une fonction revient à diminuer le nombre de terme qui intervient dans sa définition et ainsi on réduit le nombre de circuits nécessaires à sa réalisation.

a. Méthode algébrique de simplification

Il s'agit d'appliquer les théorèmes et les propriétés de l'algèbre de Boole pour obtenir une expression plus simple de la fonction.

Exemple :

Simplifier les fonctions logiques suivantes par la méthode algébrique :

$$F_1 = a + \bar{a}.b$$

$$F_2 = ab + a\bar{b} + \bar{a}b$$

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

Solution :

$$F_1 = a + \bar{a}.b = \left(a + \bar{a}\right)\left(a + b\right) = a + b$$

$$F_2 = ab + a\bar{b} + \bar{a}b = \left(b + \bar{b}\right)a + \bar{a}b = a + \bar{a}b = a + b$$

$$F_3 = \bar{a}\bar{b} + \bar{a}b + a\bar{b} = \bar{a}\left(\bar{b} + b\right) + a\bar{b} = \bar{a} + b\bar{a} = \bar{a} + \bar{b}$$

b. Simplification a l'aide du tableau de Karnaugh

Le tableau de Karnaugh permet de simplifier les fonctions logiques par des groupements de cases adjacentes affectés par 1 ou 0.

Pour obtenir la forme minimale d'une fonction logique par l'utilisation des tableaux de Khrnaugh, il faut respecter les règles suivantes :

- On ne peut regrouper que 2^n cases: c.à.d. (2, 4, 8,…)
- Grouper le maximum de cases.
- Respecter les adjacents est les symétries.
- Réaliser le minimum de groupements.

Exemple 1:

$n = 2$; Entrées : a et b ; $2^2 = 4$ cases

b \ a	0	1
0	0	1
1	1	1

$S = a + b$

$n = 3$; Entrées a, b et c; $2^3 = 8$ cases

c \ ba	00	01	11	10
0	1	1	1	1
1	0	1	1	0

$S = \bar{c} + a$

$n = 4$; Entrées a, b ; c et d : $2^4 = 16$ cases

dc \ ba	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	0	1	1	1
10	0	1	0	0

$S = ac + c\bar{b}\bar{d} + a\bar{b}d + bdc$

Exemple 2:

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

dc \ ba	00	01	11	10
00	1	1	0	1
01	1	1	0	0
11	0	0	0	0
10	1	0	0	1

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

Exemple 3: (Remplissage du Tableau de Karnaugh à partir de la table de vérité)

Soit la table de vérité d'un détecteur des chiffres impairs compris entre 0 et 9 :

N°	d	c	b	a	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1

$2^3 2^2$ \ $2^1 2^0$ (dc \ ba)		00 (0)	01 (1)	11 (3)	10 (2)
0	00	0	1	1	0
4	01	0	1	1	0
12	11	ϕ	ϕ	ϕ	ϕ
8	10	0	1	ϕ	ϕ

F = a

V. Formes canoniques

Une fonction est sous **forme canonique** (ou **normale**) si chaque terme contient toutes les variables. L'écriture sous forme canonique est unique.

Exemple :

Soit la table de vérité d'un détecteur des chiffres impairs compris entre 0 et 7 :

N°	c	b	a	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

- **1ère forme canonique :** somme de produits (groupement des « 1 »)

$$F = f(a,b,c) = \sum(1,3,5,7) = a\bar{b}\bar{c} + ab\bar{c} + a\bar{b}c + abc$$

$$F = \underbrace{a\bar{b}\bar{c}}_{\uparrow} + ab\bar{c} + a\bar{b}c + abc$$

Minterme

Remarque 1: A partir de la **1ère Forme canonique** on peut facilement représenter la fonction F par des opérateurs **NAND** seulement.

$$F = a\bar{b}\bar{c} + ab\bar{c} + a\bar{b}c + abc = \overline{a\bar{b}\bar{c}} / \overline{ab\bar{c}} / \overline{a\bar{b}c} / \overline{abc} =$$

$$= \left(a/\bar{b}/\bar{c}\right)/\left(a/b/\bar{c}\right)/\left(a/\bar{b}/c\right)/\left(a/b/c\right) = \left(a/b/c\right)/\left(a/b/\bar{c}\right)/\left(a/\bar{b}/c\right)/\left(a/\bar{b}/\bar{c}\right)$$

- **Deuxième forme canonique :** produit de sommes (groupement des « 0 »)

$$\bar{F} = f(a,b,c) = \sum(0,2,4,6) = \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}\bar{b}c + \bar{a}bc$$

$$\bar{\bar{F}} = \overline{\bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}\bar{b}c + \bar{a}bc} = \left(\overline{\bar{a}\bar{b}\bar{c}}\right)\left(\overline{\bar{a}b\bar{c}}\right)\left(\overline{\bar{a}\bar{b}c}\right)\left(\overline{\bar{a}bc}\right) = (a+b+c)\left(a+\bar{b}+c\right)\left(a+b+\bar{c}\right)\left(a+\bar{b}+\bar{c}\right)$$

$$F = \underbrace{(a+b+c)}_{\uparrow}\left(a+\bar{b}+c\right)\left(a+b+\bar{c}\right)\left(a+\bar{b}+\bar{c}\right)$$

Maxterme

Remarque 2: A partir de la **2ème Forme canonique** on peut facilement représenter la fonction F par des opérateurs **NOR** uniquement :

$$F = (a+b+c)\left(a+\bar{b}+c\right)\left(a+b+\bar{c}\right)\left(a+\bar{b}+\bar{c}\right) = \left(\overline{a+b+c}\right)\downarrow\left(\overline{a+\bar{b}+c}\right)\downarrow\left(\overline{a+b+\bar{c}}\right)\downarrow\left(\overline{a+\bar{b}+\bar{c}}\right)$$

$$= \left(a\downarrow b\downarrow c\right)\downarrow\left(a\downarrow\bar{b}\downarrow c\right)\downarrow\left(a\downarrow b\downarrow\bar{c}\right)\downarrow\left(a\downarrow\bar{b}\downarrow\bar{c}\right)$$

Chapitre 3: circuit logiques combinatoires

I. Additionneur :

1) Demi-additionneur

Un demi-additionneur fourni la somme S et la retenue C de deux bits a et b sans tenir compte de la retenue de l'étage précédent.

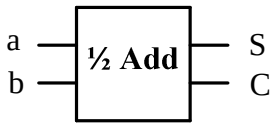


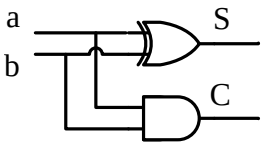
Schéma bloc

Table de vérité

a	b	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = \bar{a}b + a\bar{b} = a \oplus b$

$C = ab$



Logigramme

2) Additionneur :

L'additionneur complet est le circuit de base pour effectuer la somme de nombres de plusieurs bits.

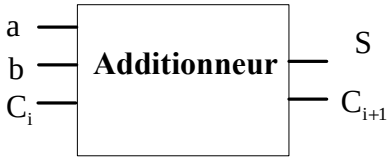


Schéma bloc

Table de vérité

a	b	C_i	S	C_{i+1}
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

L'expression de la somme S est alors**:**

$$S = \bar{a}\bar{b}C_i + \bar{a}b\bar{C}_i + a\bar{b}\bar{C}_i + abC_i$$
$$= C_i\left(\bar{a}\bar{b} + ab\right) + \bar{C}_i\left(\bar{a}b + a\bar{b}\right) = C_i\left(\overline{a \oplus b}\right) + \bar{C}_i\left(a \oplus b\right)$$

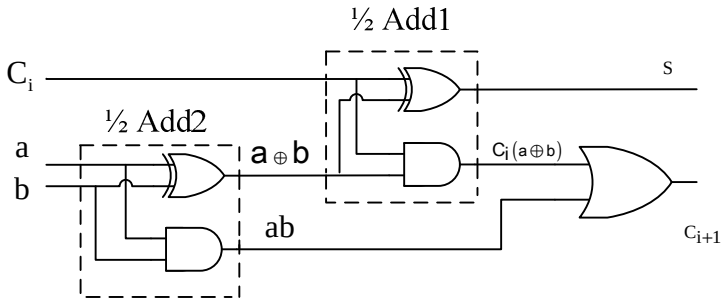
$$\boxed{S = C_i \oplus (a \oplus b)}$$

L'expression de la retenue C_{i+1} est alors **:**

$$C_{i+1} = \bar{a}bC_i + a\bar{b}C_i + ab\bar{C}_i + abC_i$$
$$= C_i\left(\bar{a}b + a\bar{b}\right) + ab\left(\bar{C}_i + C_i\right) = C_i\left(a \oplus b\right) + ab$$

$$\boxed{C_{i+1} = ab + C_i(a \oplus b)}$$

Ce qui conduit au schéma logique relatif à l'additionneur complet :

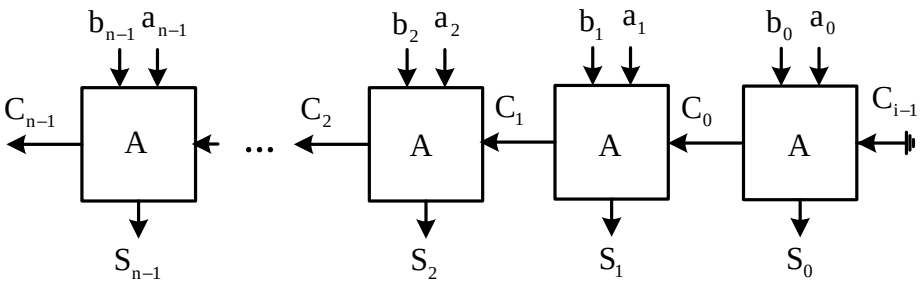


Logigramme d'un additionneur

3) Additionneur parallèle à deux nombres:

A partir des additionneurs il est facile d'effectuer l'addition de deux nombres binaires à n bits. Cette addition est réalisée par la mise en cascade de n additionneur complets, comme le montre le montage de la figure ci-dessous, il s'agit d'un additionneur parallèle, parce qu'on additionne toutes les colonnes en même temps.

$$
\begin{array}{rcccccc}
 & C_{n-1} \ldots & & & C_1 & C_0 & \\
 & & a_{n-1} & \ldots & a_2 & a_1 & a_0 \\
+ & & & & & & \\
 & & b_{n-1} & \ldots & b_2 & b_1 & b_0 \\
\hline
 & S_n & S_{n-1} & \ldots & S_2 & S_1 & S_0
\end{array}
$$



Additionneur parallèle de deux nombres binaire à N bits

4) Soustracteur :

Exemple : a – b

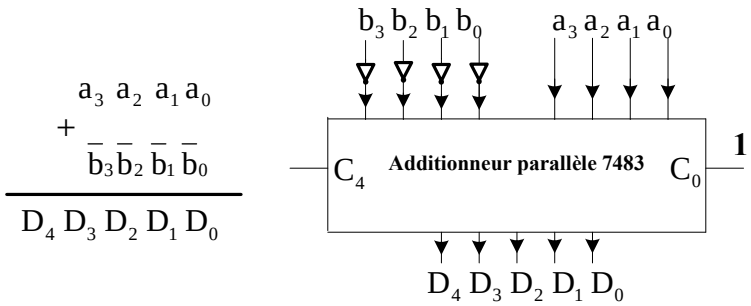
Table de vérité

a	b	D	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D = \bar{a}b + a\bar{b} = a \oplus b$$
$$C = \bar{a}b$$

Remarque 1 : La soustraction se résume à une opération d'addition, si on exprime les nombres négatifs selon la notation en complément à 2.

Pour effectuer cette soustraction en utilisant l'additionneur 7483, il faut tout d'abord effectuer le complément à 2 de $b_3\, b_2\, b_1\, b_0$, pour cela on prend le complément à 1 à l'aide des inverseurs et on ajoute ensuite 1 au bit de poids le plus faible à C_0



Soustracteur à l'aide de l'additionneur 7483

II. Comparateur

a) Comparateur de deux nombres binaires à 1 bit

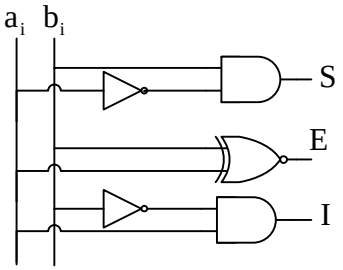
Soient deux nombres binaires à 1 bit a et b, le comparateur permet d'affirmer que a est supérieur (S), inférieur (I) ou égal (E) à b.

Table de vérité

a	b	S(⟩)	I(⟨)	E(=)
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

Logigramme

$$S = a\bar{b}$$
$$I = \bar{a}b$$
$$E = \bar{a}\bar{b} + ab = \overline{a \oplus b}$$



Comparateur de deux nombres binaires à 1 bit

b) Comparateur de deux nombres binaires à n bits

Pour comparer deux nombres binaires de n bits, il faut effectuer une comparaison bit par bit, en commençant par les bits de poids le plus fort, s'ils sont égaux on passe aux bits de poids immédiatement inférieur et ainsi de suite….etc.

Soit A et B les deux nombres à n bits

$A = a_{n-1}a_{n-2}...a_1a_0$ et $B = b_{n-1}b_{n-2}...b_1b_0$

- $(A = B)$

Il faut que l'on ait :

$$(a_{n-1} = b_{n-1}) \text{ et } (a_{n-2} = b_{n-2})... \text{ et } (a_1 = b_1) \text{ et } (a_0 = b_0)$$

Donc l'équation booléenne de E est :

$$E = \left(\overline{a_{n-1} \oplus b_{n-1}}\right)\left(\overline{a_{n-2} \oplus b_{n-2}}\right)...\left(\overline{a_0 \oplus b_0}\right)$$

- $(A \rangle B)$

$(a_{n-1} \rangle b_{n-1}) \Rightarrow (S_{n-1} = 1)$ Ou $(a_{n-1} = b_{n-1})$ et $(a_{n-2} \rangle b_{n-2}) \Rightarrow (S_{n-2} = 1)$

……

$$\text{Ou}\left(a_{n-1}=b_{n-1}\right)\ \text{et}\ \left(a_{n-2}=b_{n-2}\right)\ \ldots\ \text{et}\left(a_1=b_1\right)\text{et}\ \left(a_0\rangle\ b_0\right)\Rightarrow\left(S_0=1\right)$$

Donc l'équation booléenne de S est :

$$S=a_{n-1}\bar{b}_{n-1}+\left(\overline{a_{n-1}\oplus b_{n-1}}\right)a_{n-2}\bar{b}_{n-2}+\ldots+\left(\overline{a_{n-1}\oplus b_{n-1}}\right)\ldots\left(\overline{a_1\oplus b_1}\right)a_0\bar{b}_0$$

- $\underline{(A\langle B)}$

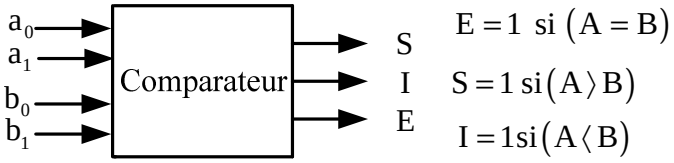
En utilisant le même raisonnement que précédent on a :

$$I=\bar{a}_{n-1}b_{n-1}+\left(\overline{a_{n-1}\oplus b_{n-1}}\right)\bar{a}_{n-2}b_{n-2}+\ldots+\left(\overline{a_{n-1}\oplus b_{n-1}}\right)\ldots\left(\overline{a_1\oplus b_1}\right)\bar{a}_0b_0$$

<u>Exercice d'application</u>

Comparateur de deux nombres à deux bits (n=2) :

$A=a_1a_0$ et $B=b_1b_0$



Déterminez les expressions de :

a) $E=1\ \text{si}\ (A=B)$

b) $S=1\ \text{si}(A\rangle B)$

c) $I=1\ \text{si}(A\langle B)$

<u>Eléments de corrections</u>

a) $E=\left(\overline{a_1\oplus b_1}\right)\left(\overline{a_0\oplus b_0}\right)$

b) $S=a_1\bar{b}_1+\left(\overline{a_1\oplus b_1}\right)a_0\bar{b}_0$

c) $I=\bar{a}_1b_1+\left(\overline{a_1\oplus b_1}\right)\bar{a}_0b_0$

III. Codeurs et décodeurs

1) Codeur :

Un codeur (encodeur) est un circuit logique qui possède 2^n voies d'entrée dont une seule active et n voies de sortie.

Un codeur (encodeur) est une "boîte noire", avec 2^n entrées, dont un seul est active à la fois, et n sorties. L'état des sorties indique quelle entrée est active.

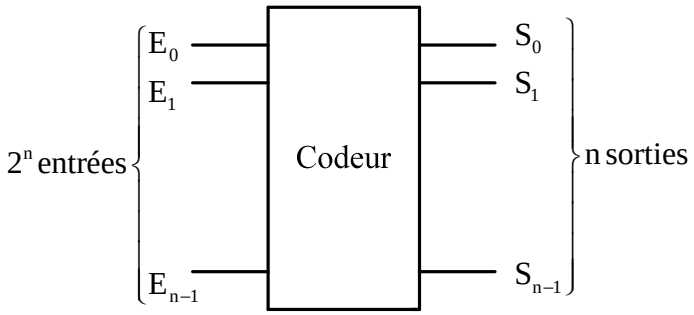


Schéma fonctionnel d'un Codeur

Exemple : Codeur B.C.D:

	Sorties		
Entrée à 1	S_2	S_1	S_0
E_0	0	0	0
E_1	0	0	1
E_2	0	1	0
E_3	0	1	1
E_4	1	0	0
E_5	1	0	1
E_6	1	1	0
E_7	1	1	1

Table de véritédu codeur B.C.D

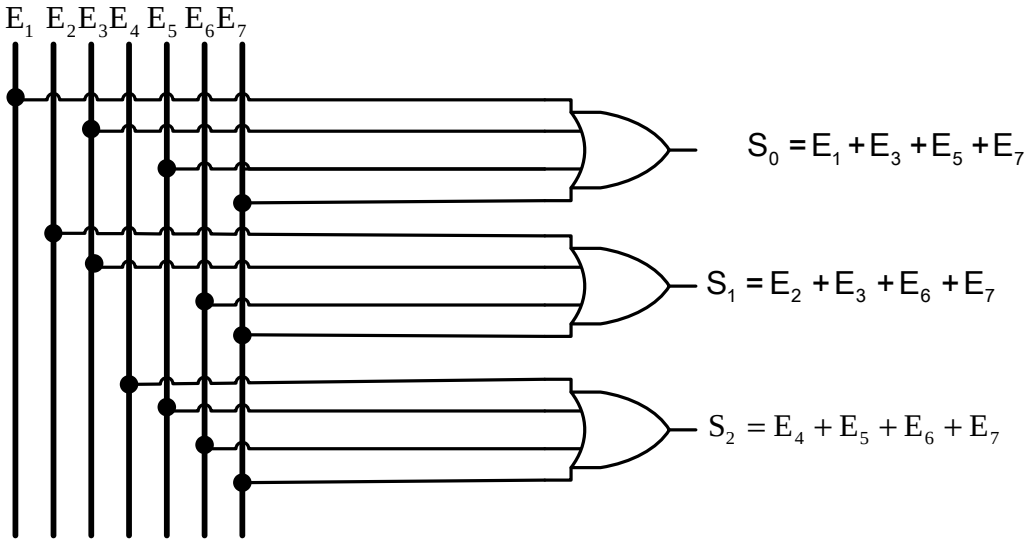
Les équations logiques associées se déduisent facilement à partir de la table de vérité :

On a alors :

$$S_0 = E_1 + E_3 + E_5 + E_7$$

$$S_1 = E_2 + E_3 + E_6 + E_7$$

$S_2 = E_4 + E_5 + E_6 + E_7$



2) Codeur de priorité :

Dans le cas ou plusieurs entrées seraient simultanément activités, on fait appel à un codeur de priorité pour éviter toute ambiguïté. C'est un dispositif qui réalise le codage du numéro le plus élevé dans le cas ou plusieurs entrées seraient actionnées.

	Entrées							Sorties		
Chiffre décimal	E_7	E_6	E_5	E_4	E_3	E_2	E_1	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	1
2	0	0	0	0	0	1	X	0	1	0
3	0	0	0	0	1	X	X	0	1	1
4	0	0	0	1	X	X	X	1	0	0
5	0	0	1	X	X	X	X	1	0	1
6	0	1	X	X	X	X	X	1	1	0
7	1	X	X	X	X	X	X	1	1	1

Table de vérité du codeur de priorité B.C.D

3) Décodeur :

Un décodeur est un circuit numérique qui possède n entrées et 2^n sorties. Pour chacune des combinaisons possibles des entrées, seule une ligne de sortie est validée (soit à un niveau haut, soit à un niveau bas selon le type de décodeur).

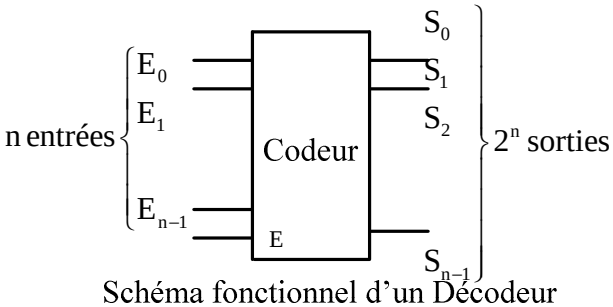


Schéma fonctionnel d'un Décodeur

Synthèse d'un décodeur :

Entrées			Sorties							
C	B	A	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Table de vérité d'un décodeur 3 bits

On en déduit :

$$S_0 = \overline{A}\,\overline{B}\,\overline{C}\ ,S_1 = A\overline{B}\,\overline{C}, S_2 = \overline{A}B\overline{C}, S_3 = AB\overline{C}, S_4 = \overline{A}\,\overline{B}C, S_5 = A\overline{B}C, S_6 = \overline{A}BC, S_7 = ABC$$

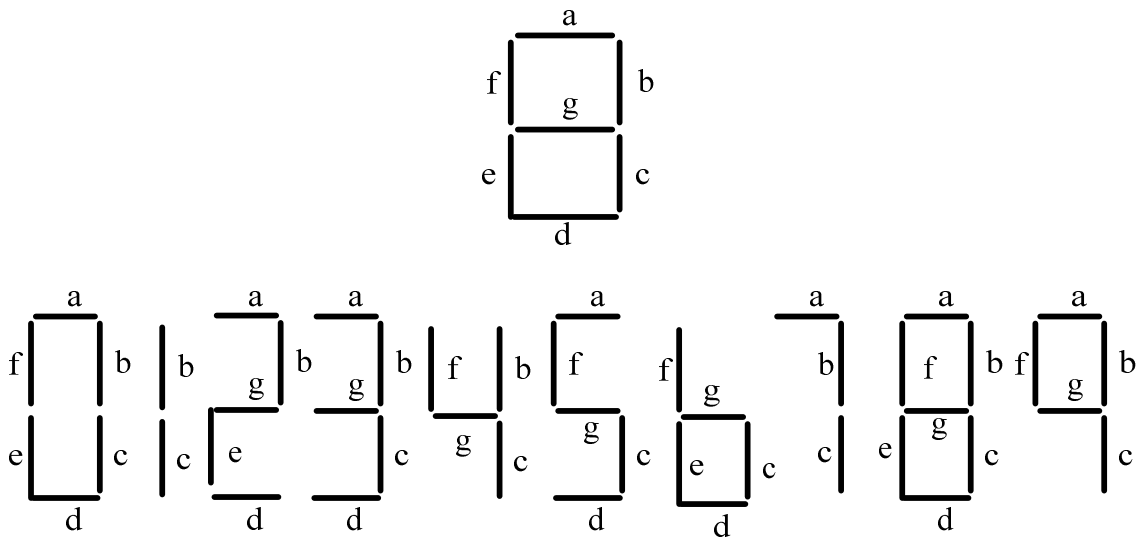
III. Transcodeurs

Un transcodeur est un dispositif qui permet de faire passer une information écrite dans le code C_1 à un code C_2.

Les deux plus importantes applications des transcodeurs sont : la conversion de code et l'affichage par segments.

Exemple 1: Transcodeur BCD-7 segments

Les dix chiffres 0 à 9 sont affichés au moyen d'un dispositif appelé afficheur 7 segments lumineux qui sont des diodes électroluminescentes(DEL)



Afficheur à 7 ségment

Exemple 2 : transcodeur à deux bits : **Binaire-Gray**.
La table de vérité est donnée sur la figure ci-dessous

B_1	B_0	G_1	G_0
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Table de vérité d'un transcodeur 2 bits Binaire-Gray

On déduit : $G_0 = B_1$ et $G_0 = \overline{B}_1 B_0 + B_1 \overline{B}_0 = B_0 \oplus B_1$

Exemple 3 :(transcodeur Gray-binaire) :

Chercher le circuit logique **transcodeur Gray-binaire** qui permet de convertir le code **gray** à 4 bits, $(G_3G_2G_1G_0)$, en code **binaire** $(B_3B_2B_1B_0)$ (utiliser un tableau).

1) Montrer en utilisant le tableau de Karnaugh que :

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

$$B_1 = G_3 \oplus G_2 \oplus G_1$$

$$B_2 = G_3 \oplus G_2$$

$B_3 = G_3$

2) Tracer le logigramme de ce transcodeur.

V. Multiplexeur et Démultiplexeur

1. Multiplexeur

Un multiplexeur est un circuit logique ayant :

- n fils d'adresse.
- 2^n fils d'entrée
- Une seule sortie S.

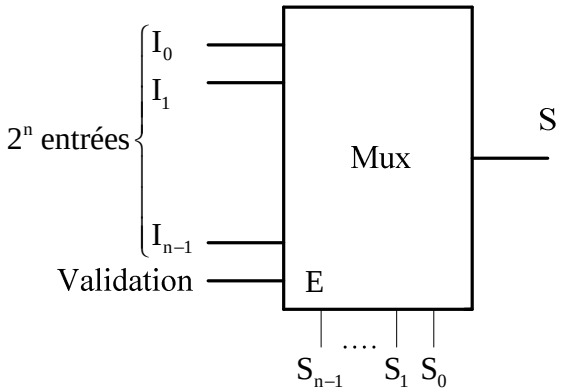
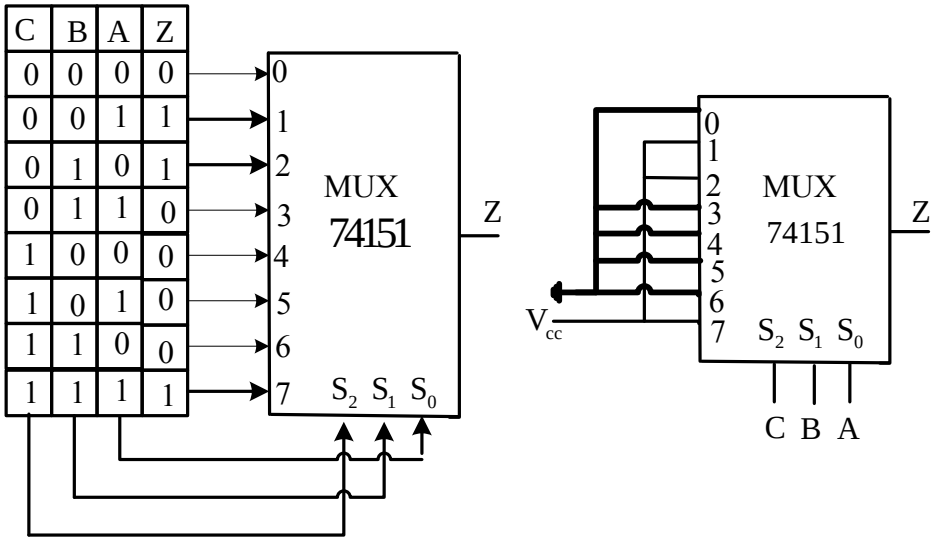


Schéma fonctionnel d'un multiplexeur

<u>Exemple</u> : Synthèse d'une fonction logique à l'aide d'un multiplexeur :



Synthèse d'une fonction logique à l'aide d'un multiplexeur

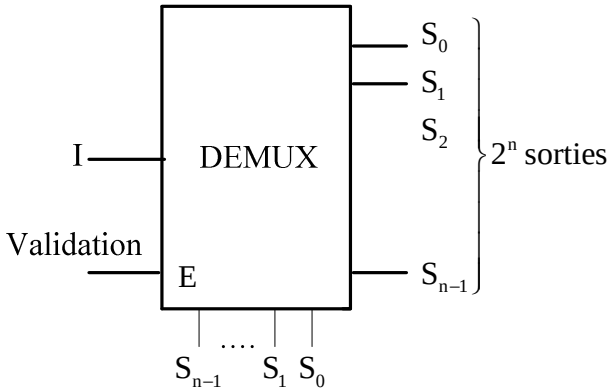
Les variables A B et C sont raccordées respectivement à S_0 S_1 et S_2

$$Z = A\overline{B}\,\overline{C} + \overline{A}B\overline{C} + ABC$$

2. **Démultiplexeur**

Un démultiplexeur est un circuit logique ayant :

- n fils d'adresse.
- 2^n fils de sortie
- Une seule entrée.

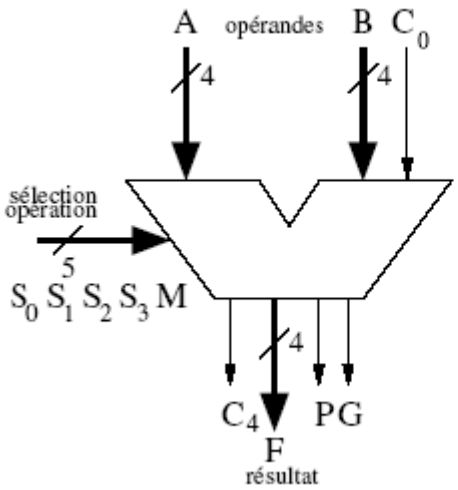


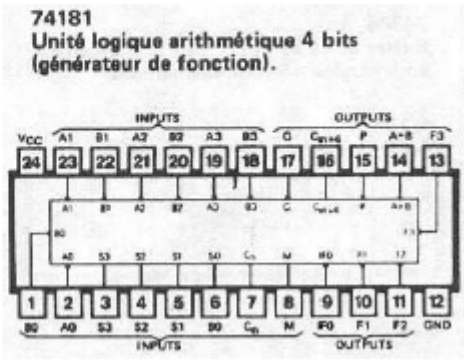
IV Unité arithmétique et logique(U.A.L) :74181

Fonctionnement de l'U.A.L 74181

L'unité arithmétique et logique, abrévié **UAL** (ALU, Arithmetic Logic Unit en anglais), est l'organe de l'ordinateur chargé d'effectuer les calculs. Le plus souvent, l'UAL est incluse dans l'unité centrale ou le microprocesseur.

C'est un circuit intégré capable d'effectuer des opérations logiques ou arithmétiques sur des mots de 4 bits :





Il dispose de :

- Huit entrées relatives aux deux nombres binaires à traiter $A = A_3A_2A_1A_0$ et $B = B_3B_2B_1B_0$
- Une entrée de retenue C_n
- Cinq entrées de sélections $S_3S_2S_1S_0$ complétées par une entrée M
- Si M=1, l'U.A.L effectue une opération logique selon le code de sélection.
- Si M=0, l'U.A.L effectue une opération arithmétique selon le code de sélection.
- Quatre sorties $F = F_3F_2F_1F_0$ pour afficher le résulta de sortie.
- Une sortie de la retenue C_{n+4}.
- L'U.A.L comprend un comparateur qui met la sortie A=B, chaque fois que les entrées A et B sont égales et cela indépendamment de résulta F.
- Deux sorties relatives aux termes de propagation $\overline{P}$ (ou X) et de génération $\overline{G}$ (ou Y).

Semestre 2

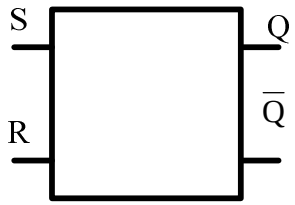
Chapitre 4 : bascules : R.S-R.S.H -J.K-D- T

1) Introduction

- La bascule est l'élément de base de la logique séquentielle.
- Une bascule est un circuit bistable (deux états stables) pouvant prendre deux états logiques : 0 et 1.
- L'état de la bascule peut-être modifié en agissant sur une ou plusieurs entrées.
- Le nouvel état de la bascule dépend non seulement des combinaisons des entrées mais également de l'état précédent : la bascule à la capacité de conserver son état (état des sorties) : elle est utilisé come mémoire.

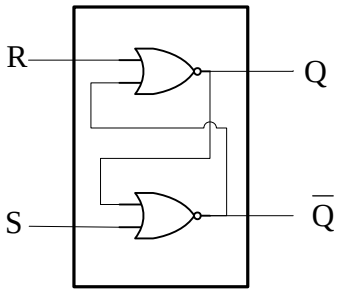
2) BASCULE ASYNCHRONE (R-S)

Symbole et table de vérité

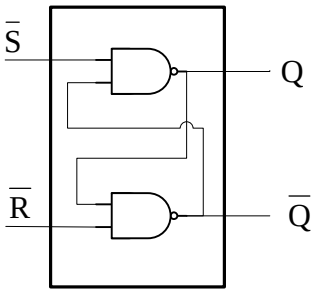


S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Interdit

Technologie NOR

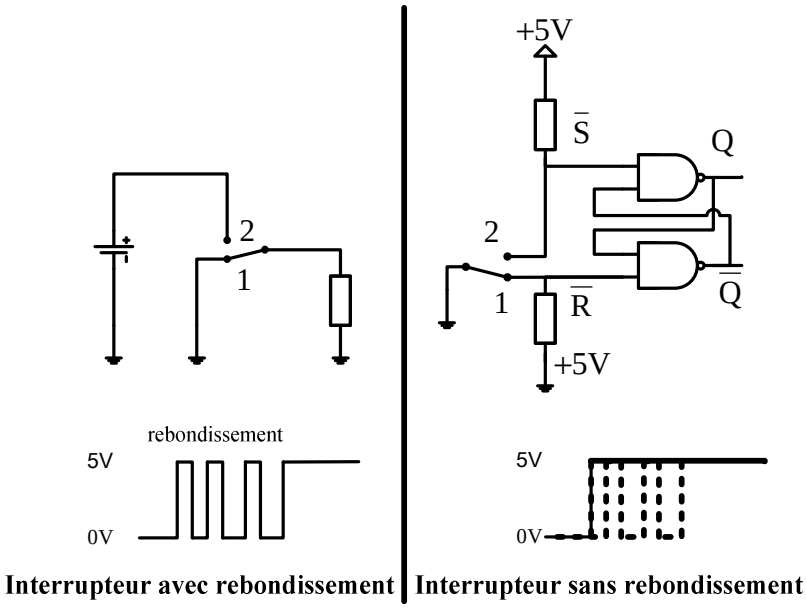


Technologie NAND



Application : (anti-rebond)

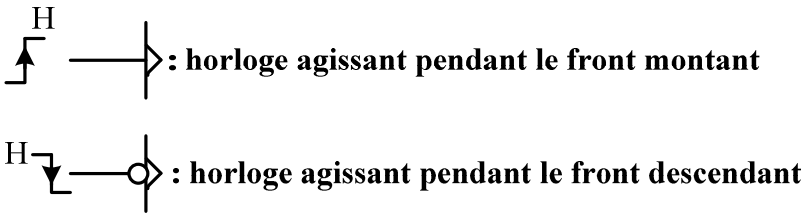
Pour éviter les rebondissements on insère un circuit anti-rebond à base des bascules RS immédiatement après l'interrupteur de commande.



3. Bascules synchrones :

La bascule RS ne permet pas de contrôler les instants de commutations des sorties Q et $\overline{Q}$. Une bascule synchrone sera pilotée par un signal externe sera piolé Horloge et qui lui délivre des impulsions périodiques. On dit que qu'il y a synchronisation des basculements sur les impulsions d'horloge.

Toutes les bascules synchrones disposent d'une entrée d'horloge H ou T (trigger) ou CLK et d'une ou deux entrée de donnés.



a. Bascule synchrone RSH

L'état de sortie est toujours déterminé par les entrées R et S mais s'apparaît qu'au moment ou se produit la transition dans le signal d'horloge.

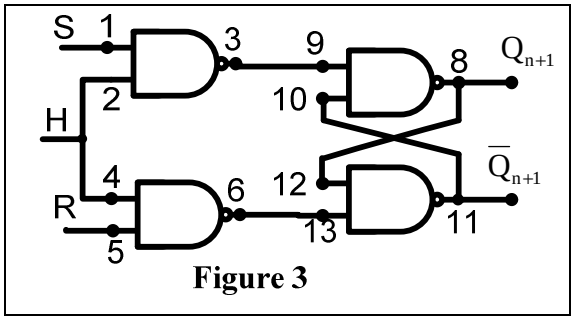
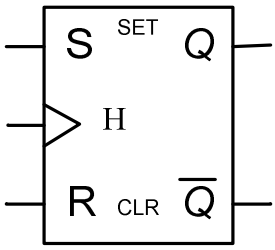


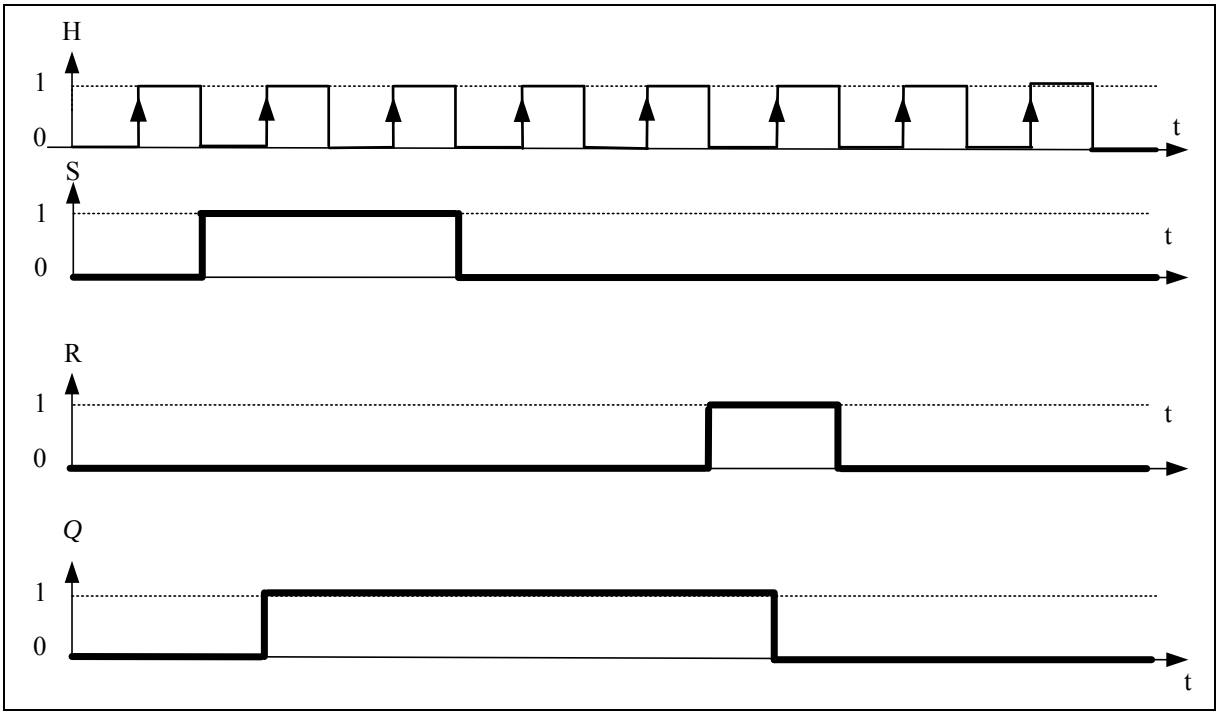
Figure 3



SET: mise à 1
CLR: mise à 0

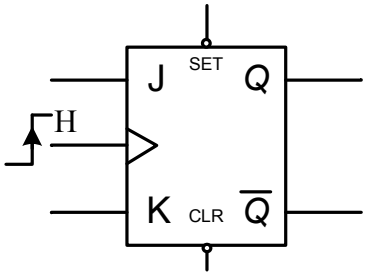
Bascule RSH

Chronogramme pour le front montant :



b. Bascule synchrone JK

La bascule JK est une mémoire bistable commandée par deux entrées J : (entrée d'enclenchement) et K : (entrée de déclenchement) Cette bascule élimine la condition indéterminée de la bascule précédente(SR) en effet si JK=11 on aura $\boxed{Q_{n+1} = \overline{Q}_n}$.

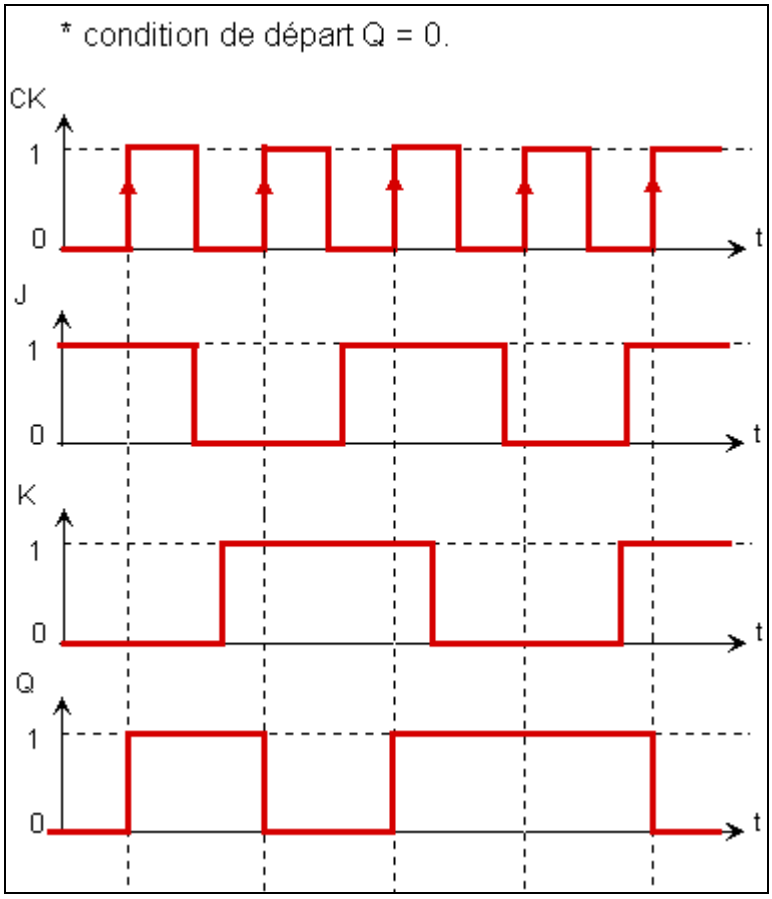


SET et CLR sont :

- Des entrées de forçages asynchrones.
- Actives niveau bas.
- SET : forçages à 1
- CLR forçages à 0

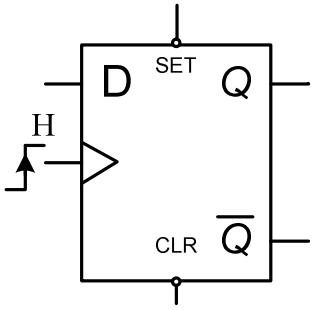
H	J	K	Q_n
0	∅	∅	Q_{n-1}
↑	0	0	Q_{n-1}
↑	0	1	0
↑	1	0	1
↑	1	1	$\overline{Q}_{n-1}$

Table de vérité



c. Bascule synchrone D

La bascule D synchrone est une mémoire bistable commandée par une seule entrée. Elle recopie la valeur de D sur Q_n à chaque front d'horloge.

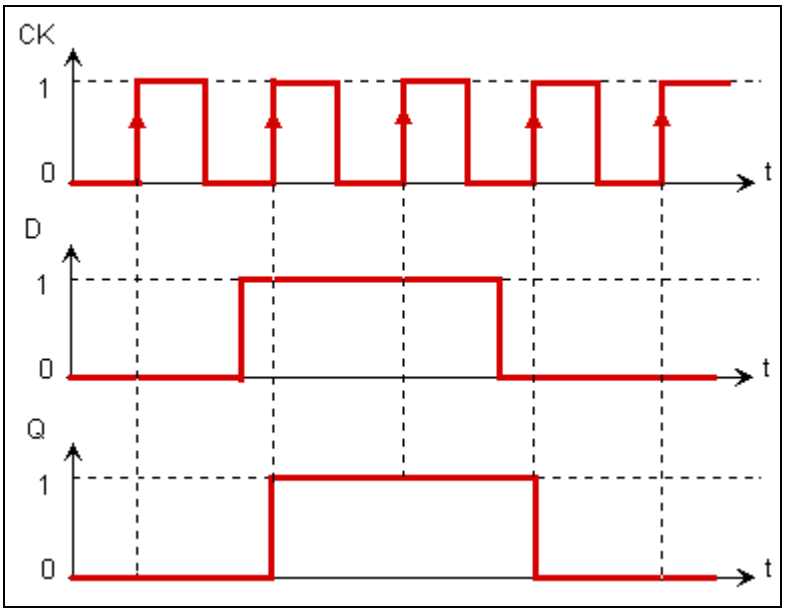


H	D	Q_n
0	∅	Q_{n-1}
↑	0	0
↑	1	1

Table de vérité

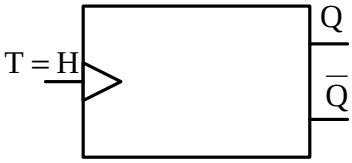
SET et CLR sont :

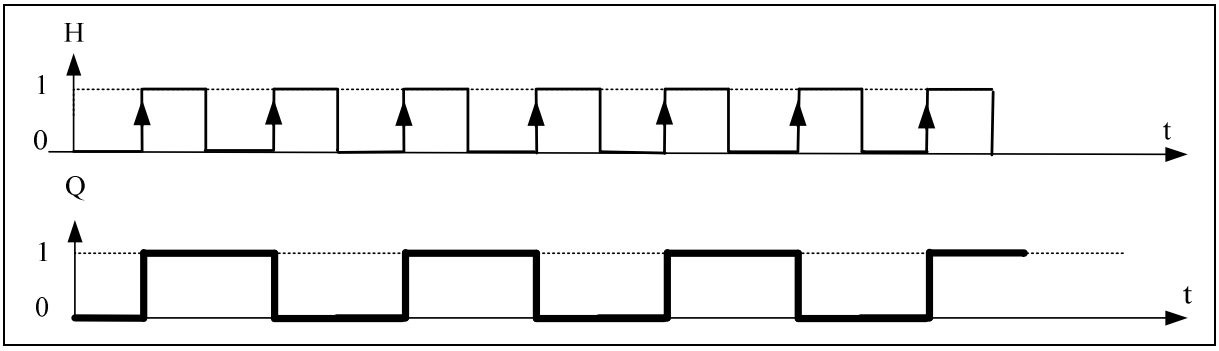
- Des entrées de forçages asynchrones.
- Actives niveau bas.
- SET : forçage à 1
- CLR forçage à 0



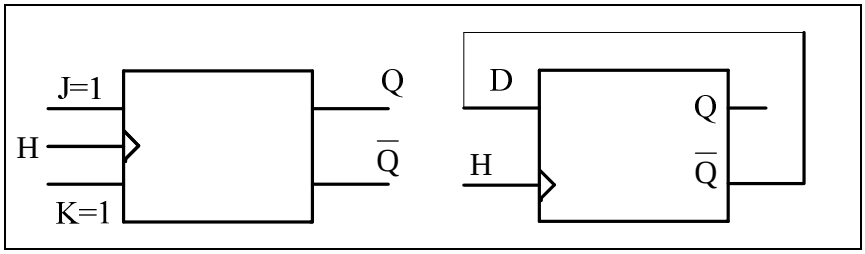
d. Bascule synchrone T

Cette bascule est commandée par une seule entrée T (timing), elle change d'état à chaque front d'horloge.





Application : On peut réaliser une bascule T synchrone à partir de la bascule JK ou de la bascule D. Ces bascules jouent le rôle d'un diviseur par 2 de fréquence.



La **bascule JK** est la plus évoluée, son rôle est essentiel au comptage. C'est une bascule RS maître esclave avec une rétroaction croisée entre les sorties et les entrées. Elle permet d'effectuer du comptage et de prépositionné, par ses entrées J et K, le départ du comptage et son arrêt.

Chapitre 5: compteurs et décompteurs

1) Compteur asynchrone

Un compteur asynchrone est constitué de plusieurs bascules en cascade. La première bascule reçoit le signal d'horloge CLK, le deuxième reçoit comme signal d'horloge le signal de sortie de la bascule précédente et ainsi de suite.

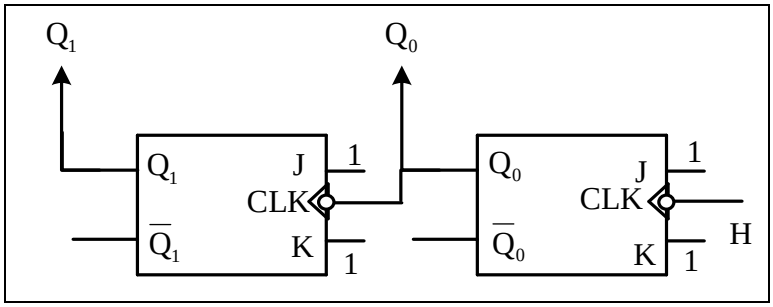
a) Compteur à cycle complet :

Exemple 1 : Compteur asynchrone modulo 4, avec des bascules JK à front descendant.

La capacité de comptage (**N**) fixe le nombre de bascules (**n**) : $2^{n-1} \langle N \leq 2^n$

$N = 4 = 2^{n=2} \Rightarrow$ n=2 bascules.

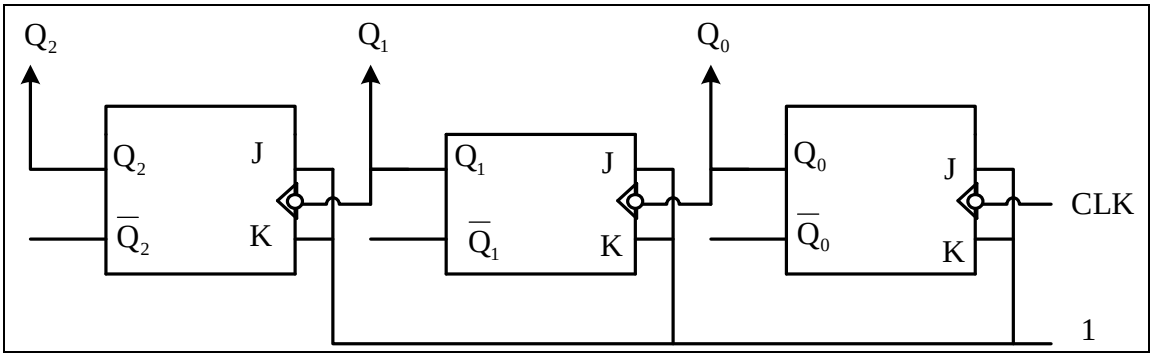
Bascules à front **descendant** donc $H_0 = H$ Et $H_i = Q_{i-1}$

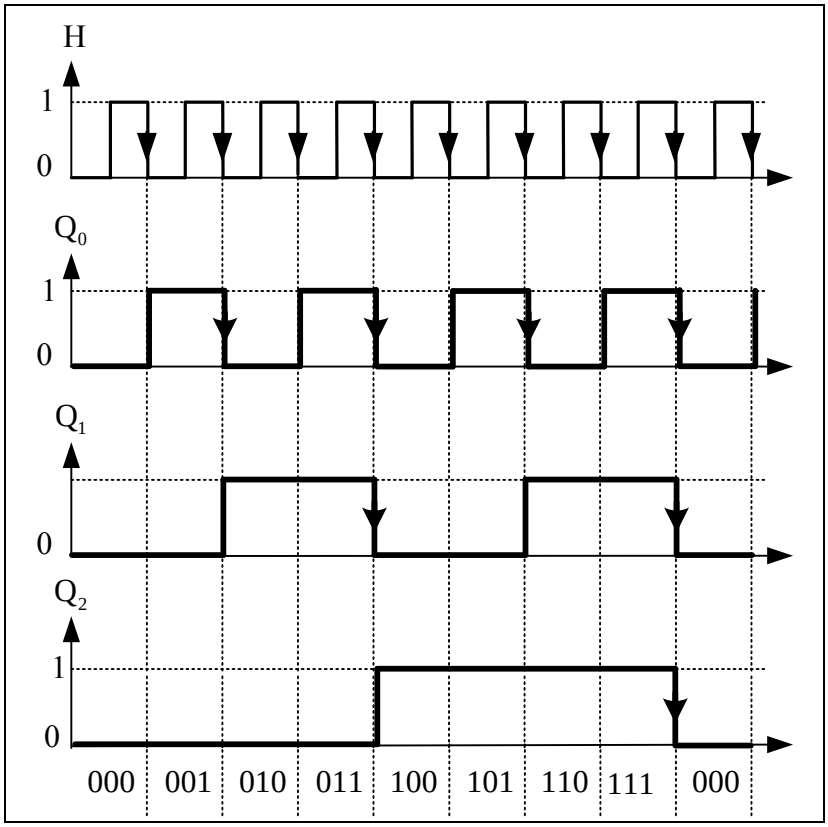


Exemple 2 : Compteur asynchrone modulo 8, avec des bascules JK à front descendant.

Le nombre de bascules: $2^{n-1} \leq N \leq 2^n \Rightarrow N = 8 = 2^{n=3}$

Bascules à front **descendant** donc $H_0 = H = CLK$ Et $H_i = Q_{i-1}$





Remarque :

Pour les Bascules à front montant $\boxed{H_0 = H}$ **Et** $\boxed{H_i = \overline{Q}_{i-1}}$

b) Décompteur asynchrone

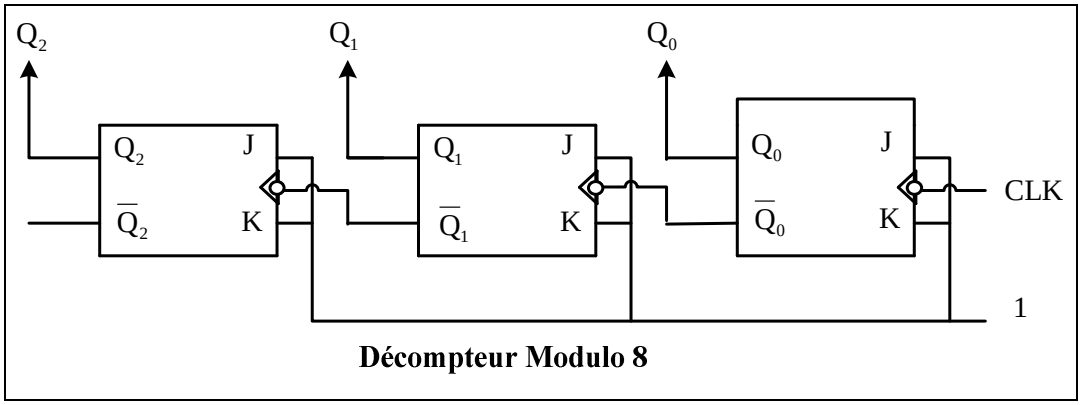
Dans les décompteurs asynchrones l'information à décompter est reliée à l'entrée d'horloge de la première bascule.

- Si le front est descendant alors la sortie $\overline{Q}$ de la première bascule est reliée à l'entrée d'horloge de la bascule suivante est ainsi de suite ;
- Si le front est montant alors la sortie Q de la première bascule est reliée à l'entrée d'horloge de la bascule suivante est ainsi de suite.

Exemple : décompteur asynchrone modulo (N=8) à base de bascule JK

Bascule : JK à font descendant

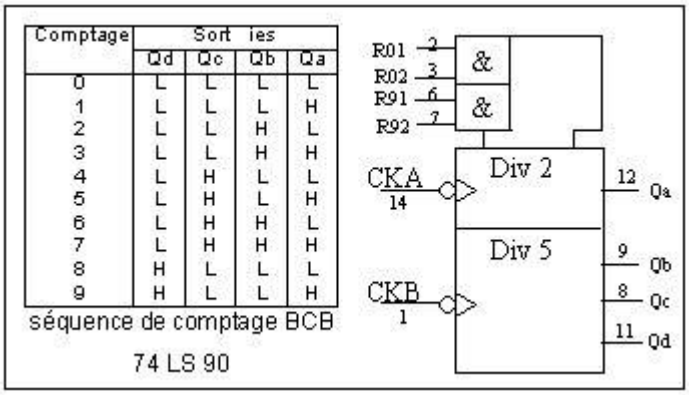
$H_0 = H$ Et $H_i = \overline{Q}_{i-1}$



Décompteur Modulo 8

3) Compteurs asynchrones en circuit intégrés

- **Compteur asynchrone décimal (7490) :**



Comptage	Sortie			
	Qd	Qc	Qb	Qa
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

séquence de comptage BCB

74 LS 90

- **Compteur asynchrone 7493**

Ce compteur peut fonctionner en diviseur par 8 (MDULO 8) en présentant l'horloge sur l'entrée INPUTB ou en diviseur par 16 en présentant l'horloge sur l'entrée INPUT A et en reliant la sortie Q_A à l'entrée INPUT B.

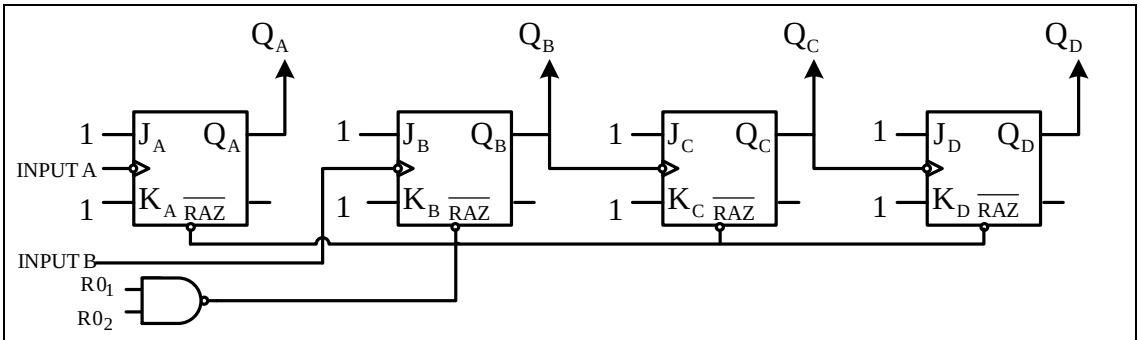


Schéma interne du compteur :7493

Une remise à zéro générale asynchrone du compteur est possible grâce aux entrées $R0_1$ et $R0_2$. Pour cela les deux entrées $R0_1$ et $R0_2$ doivent être simultanément à «1».

Entrées asycnrones		sorties			
$R0_1$	$R0_2$	Q_D	Q_C	Q_B	Q_A
1	1	0	0	0	0
0	×	comptage			
×	0	comptage			

4) Compteur synchrone

Pour ce type de compteur une horloge commune est appliquée simultanément à toutes les entrées des différentes bascules.

La capacité de comptage (**N**) fixe le nombre de bascules (**n**) : $2^{n-1} \leq N \leq 2^n$

a) Synthèse d'un compteur synchrone

La table d'excitation de la bascule JK.

Q_n	Q_{n+1}	J	K
0	0	0	∅
0	1	1	∅
1	0	∅	1
1	1	∅	0

∅ = 0 ou 1

Table d'excitation de la bascule JK

La synthèse d'un compteur synchrone consiste à calculer chaque entrée des bascules constituant le compteur afin de conditionner leurs évolutions lors de la prochaine impulsion d'horloge.

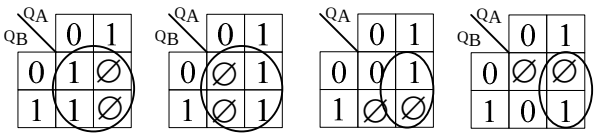
A l'instant, les sorties des bascules du compteur sont à l'état Q_n, quelles sont les valeurs à appliquer aux entrées J_i et K_i pour qu'a l'impulsion d'horloge suivante, les sorties prennent les valeurs de Q_{n+1} imposées par le cycle de comptage désiré.

Exemple1 : compteur synchrone modulo-4.

La table d'excitation permet la détermination des entrées J_i et K_i à appliquer pour obtenir les sorties avant l'impulsion d'horloge.

Etats présents(n)		Etats futurs(n+1)		Entrées: $(J_i K_i)$			
Q_B	Q_A	Q'_B	Q'_A	J_A	K_A	J_B	K_B
0	0	**0**	1	1	∅	0	∅
0	1	**1**	0	∅	1	1	∅
1	0	**1**	1	1	∅	∅	0
1	1	**0**	0	∅	1	∅	1

$\varnothing = 0$ ou 1



$J_A = 1$ $K_A = 1$ $J_B = Q_A$ $J_B = Q_A$

Exemple 2 : compteur synchrone modulo-8

	n			n+1			Entrées: $(\varnothing = 0 \text{ ou } 1)$					
N°	Q_C	Q_B	Q_A	Q'_C	Q'_B	Q'_A	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	0	1	1	$\varnothing$	0	$\varnothing$	0	$\varnothing$
1	0	0	1	0	1	0	$\varnothing$	1	1	$\varnothing$	0	$\varnothing$
2	0	1	0	0	1	1	1	$\varnothing$	$\varnothing$	0	0	$\varnothing$
3	0	1	1	1	0	0	$\varnothing$	1	$\varnothing$	1	1	$\varnothing$
4	1	0	0	1	0	1	1	$\varnothing$	0	$\varnothing$	$\varnothing$	0
5	1	0	1	1	1	0	$\varnothing$	1	1	$\varnothing$	$\varnothing$	0
6	1	1	0	1	1	1	1	$\varnothing$	$\varnothing$	0	$\varnothing$	0
7	1	1	1	0	0	0	$\varnothing$	1	$\varnothing$	1	$\varnothing$	1

Cherchons l'expression booléenne de chaque entrée J_i et K_i en utilisant le diagramme de Karnaugh :

Q_C \ Q_BQ_A	00	01	11	10
0	1	$\varnothing$	$\varnothing$	1
1	1	$\varnothing$	$\varnothing$	1

$J_A = 1$

Q_C \ Q_BQ_A	00	01	11	10
0	$\varnothing$	1	1	$\varnothing$
1	$\varnothing$	1	1	$\varnothing$

$K_A = 1$

Q_C \ Q_BQ_A	00	01	11	10
0	0	1	$\varnothing$	$\varnothing$
1	0	1	$\varnothing$	$\varnothing$

$J_B = Q_A$

Q_C \ Q_BQ_A	00	01	11	10
0	$\varnothing$	$\varnothing$	1	0
1	$\varnothing$	$\varnothing$	1	0

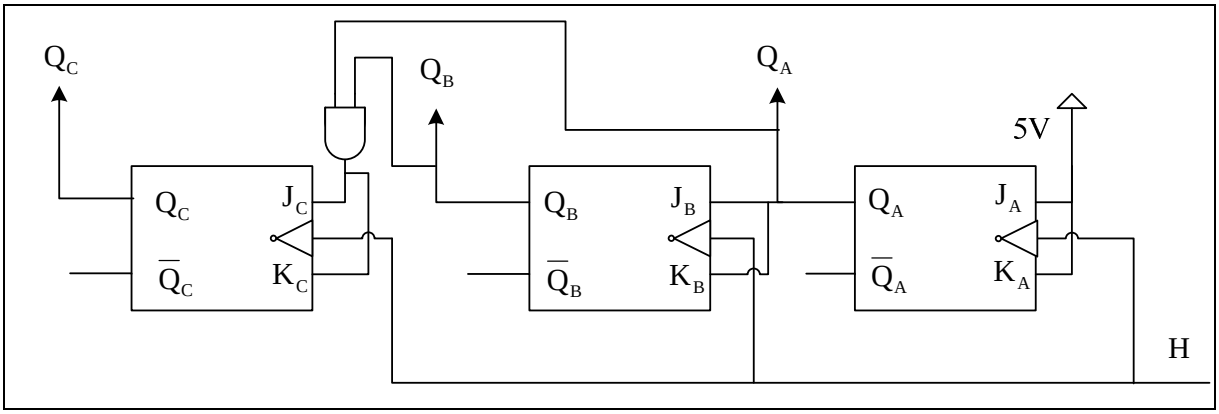
$K_B = Q_A$

Q_C \ Q_BQ_A	00	01	11	10
0	0	0	1	0
1	∅	∅	∅	∅

$J_C = Q_A Q_B$

Q_C \ Q_BQ_A	00	01	11	10
0	∅	∅	∅	∅
1	0	0	1	0

$K_C = Q_A Q_B$



Dans le cas d'un compteur synchrone, toutes les bascules changent d'état en même temps c'est-à-dire quelles sont synchrones sur le front descendant du signal d'horloge.

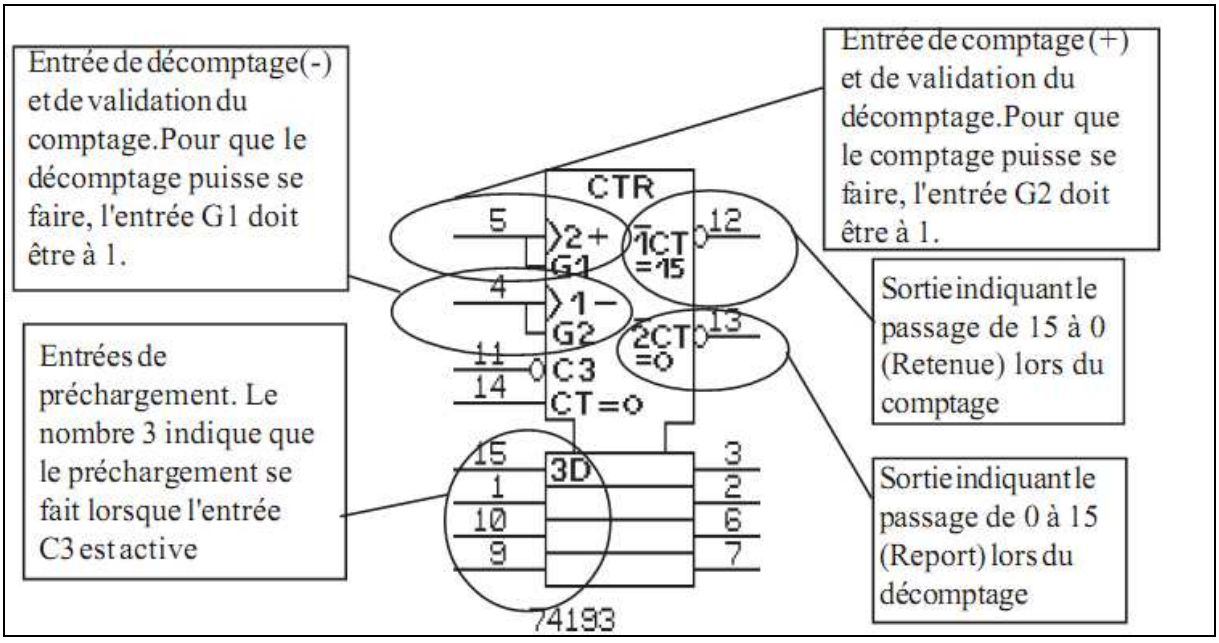
3) Compteurs synchrone en circuit intégrés

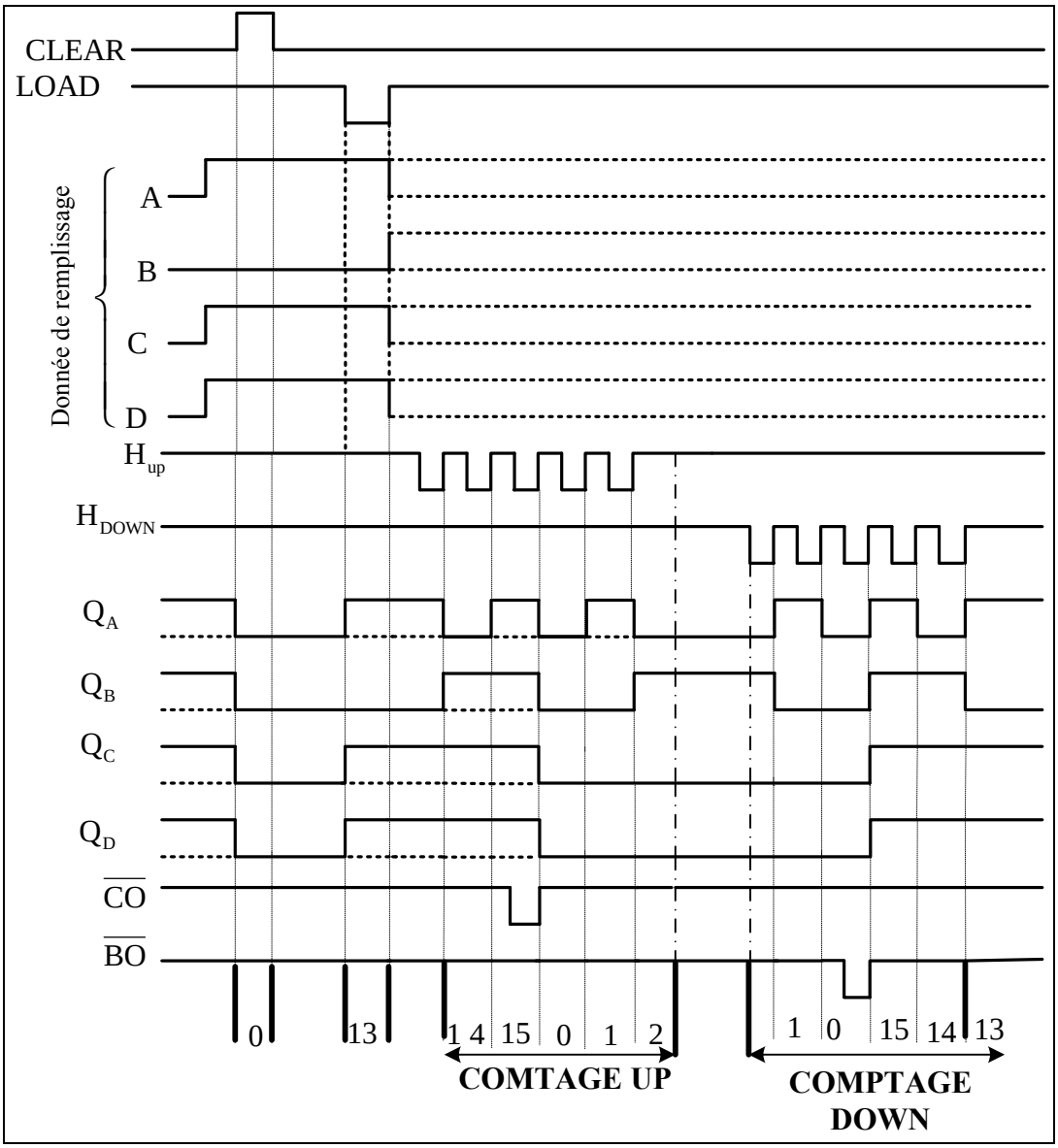
Le catalogue en circuits intégrés contient plusieurs types de compteurs synchrones. On résume dans le tableau les compteurs synchrones de la série qui sont les plus utilisés.

Référence	Type	Chargement	Réversible	Mise en cascade
74190 74191	B.C.D Hexadécimal	Asynchrone	Oui	Asynchrone ou synchrone
74192 74193	B.C.D Hexadécimal			Asynchrone
74160 74161	B.C.D Hexadécimal	Synchrone	Non	Synchrone

Exemples de compteurs synchrone de la famille 74

EXEMPLE D'UN COMPTEUR PROGRAMMABLE LE 74193 :


Entrée de décomptage(-) et de validation du comptage.Pour que le décomptage puisse se faire, l'entrée G1 doit être à 1.
Entrée de comptage (+) et de validation du décomptage.Pour que le comptage puisse se faire, l'entrée G2 doit être à 1.
Sortie indiquant le passage de 15 à 0 (Retenue) lors du comptage
Entrées de préchargement. Le nombre 3 indique que le préchargement se fait lorsque l'entrée C3 est active
Sortie indiquant le passage de 0 à 15 (Report) lors du décomptage
CTR
74193


CLEAR
LOAD
Donnée de remplissage
A
B
C
D
Hup
HDOWN
QA
QB
QC
QD
CO
BO
0 13 14 15 0 1 2
COMTAGE UP
1 0 15 14 13
COMPTAGE DOWN

Un compteur décompteur programmable comporte :

- des entrées de contrôle pour
  - le pré-chargement (associée à A, B, C, D)
  - le comptage UP
  - le décomptage : DOWN
  - la remise à zéro : CLEAR
  - une entrée LOAD de remplissage.
- Des sorties (Q_A, Q_B, Q_C, Q_D)
- $\overline{CO}$ Sortie indiquant le passage de 15 à 0 (Retenue) lors du comptage
- $\overline{BO}$ Sortie indiquant le passage de 0 à 15 (Report) lors du décomptage

Chapitre 6: Les Registres

I. Principe

Un registre est un ensemble de bascules, synchronisées par la même horloge qui permet la mémorisation de n bits en même temps. Il existe dans un ordinateur plusieurs variétés de registres, les registres parallèles, les registres à décalage (décalage à droite ou décalage à gauche) les registres séries.

Les bascules de type D sont les plus utilisées pour construire des registres de différents types en fonction de la disposition des entrées et des sorties des bascules :

Les registres à entrée série/sortie série, à entrée série/sortie parallèle, à entrée parallèle/sortie parallèle, à entrée parallèle/sortie série.

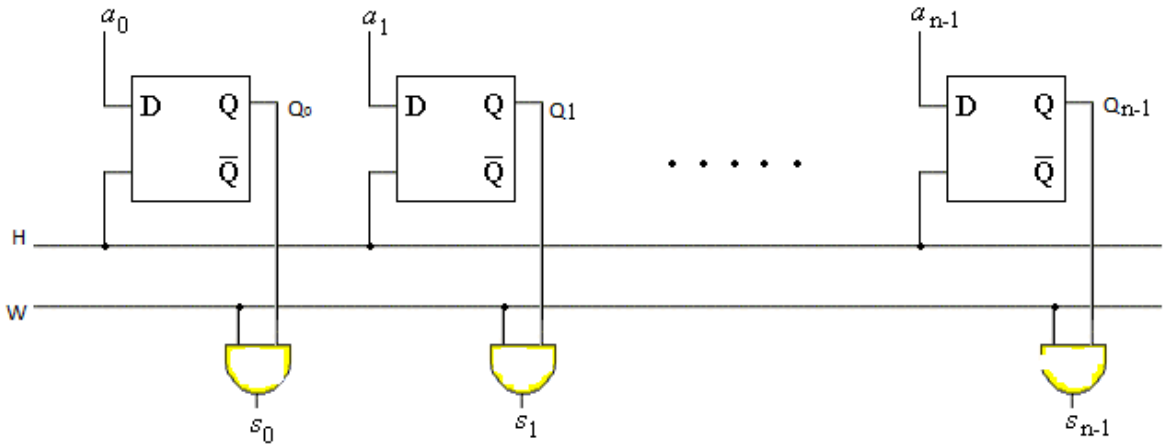
On distingue deux types de registres :

- Registre à décalage
- Registre de mémorisation

II. Registre de mémorisation (Registre parallèle)

Un registre de mémorisation (ou registre de données) est un registre capable de réaliser la fonction de mémorisation en emmagasinant une information binaire sous forme d'un mot de n bits.

Voici un exemple de registre à n entrées parallèles (a_0, a_1,…, a_{n-1}) et à n sorties parallèles (s_0, s_1,…, s_{n-1}) construit avec des bascules de type D :



Examinons le fonctionnement de ce « registre » :

- C'est un registre parallèle à n bits :

- La ligne H fournit le signal d'horloge, et permet de charger les n bits

- La ligne w permet de lire l'information sur n bits
  - ✓ Lorsque w= 0 on a (s_0=0, s_1=0, s_{n-1}=0)
  - ✓ Lorsque w = 1 on a (s_0= Q_0, s_1= Q_1, …, s_{n-1}= Q_{n-1})

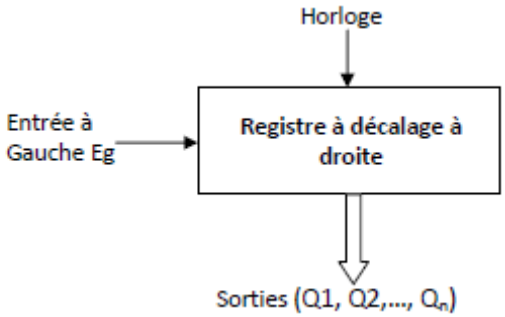
III. Registre à décalage

Un registre à décalage est un registre ayant la possibilité de décaler à droite ou à gauche ou réversible son contenu.

Ce type de registre est principalement utilisé comme mémoire d'information dynamique ; la fonction de décalage consiste de faire glisser l'information de chaque cellule élémentaire dans une autre cellule élémentaire adjacente.

III.1. Registre à décalage à droite

Il est composé de n bascules interconnectées de façon à ce que l'état logique de la sortie $\mathbf{Q_i}$ de la $\mathbf{i^{ème}}$ bascule soit reproduit à la sortie $\mathbf{Q_{i+1}}$ de la $\mathbf{(i+1)^{ème}}$ bascule quand un signal d'horloge est appliqué à l'ensemble des bascules. Ce type de registre à décalage possède une seule entrée à gauche **Eg** et n sorties **(Q_1, Q_2,…, Q_n)**.



Exemple : Registre à décalage à droite formé de quatre bascules type D à front montant.

Les expressions algébriques des variables des variables d'entrée :

$Q_i = Q_{i+1}^{+}$:(Caractéristique du décalage à droite)

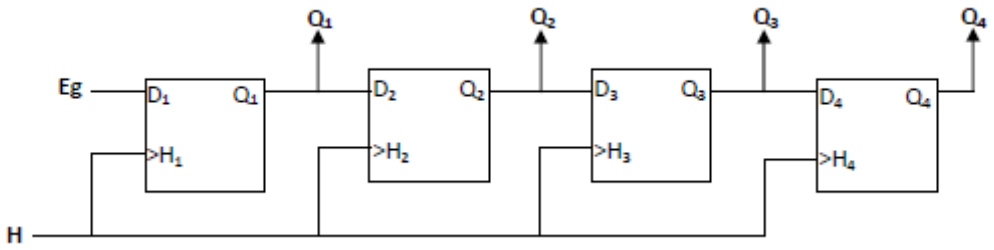
$Q_{i+1}^{+} = D_{i+1}$: (Caractéristique de la bascule D)

(A partir de ces deux égalités, on déduit que :$D_{i+1} = Q_i$

D'où :

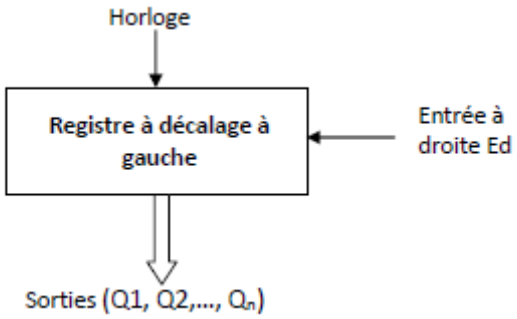
- $D_4 = Q_3$
- $D_3 = Q_2$
- $D_2 = Q_1$
- $D_1 = E_g$

Le circuit :



III .2. Registre à décalage à gauche

Il est composé de n bascules interconnectées de façon à ce que l'état logique de la sortie $\mathbf{Q_{i+1}}$ de la **(i+1)**^{ème} bascule soit reproduit à la sortie $\mathbf{Q_i}$ de la **i**^{ème} bascule quand un signal d'horloge est appliqué à l'ensemble des bascules. Ce type de registre à décalage possède une seule entrée à droite **Ed** et n sorties **(Q_1, Q_2,…, Q_n)**.



Exemple : Registre à décalage à gauche formé de quatre bascules type D à front montant.

Les expressions algébriques des variables des variables d'entrée :

Nous avons :

Les expressions algébriques des variables des variables d'entrée :

$Q_{i+1} = Q_i^+$:(Caractéristique du décalage à gauche)

$Q_i^+ = D_i$: (Caractéristique de la bascule D)

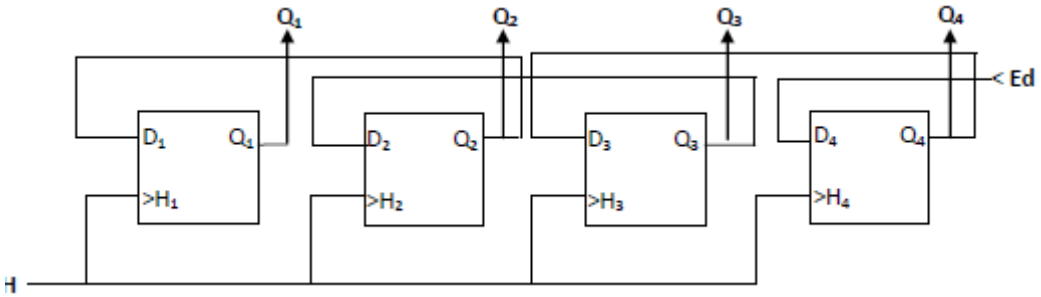
(A partir de ces deux égalités, on déduit que :

$D_i = Q_{i+1}$

D'où :

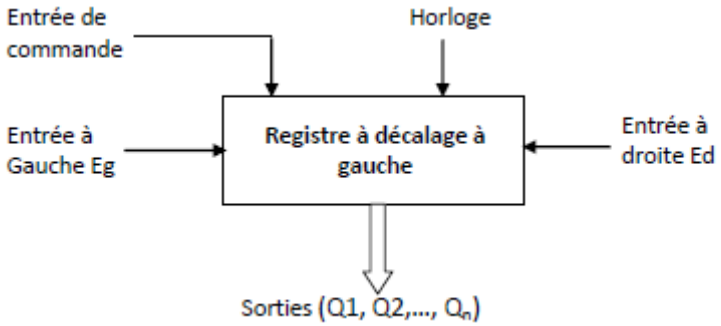
- $D_1 = Q_2$
- $D_2 = Q_3$
- $D_3 = Q_4$
- $D_4 = E_d$

Le circuit :



III.3. Registre à décalage réversible

C'est la composition des deux registres précédents en ajoutant une entrée supplémentaire pour la sélection du sens de décalage. Ce type de décalage possède n sorties **(Q1, Q2,…, Qn)** et deux entrées **Ed** et **Eg** telles que **Ed** est l'entrée à droite et **Eg** est l'entrée à gauche.



Exemple : Registre à décalage à gauche ou à droite formé de quatre bascules type D à front montant.

Un tel registre possède une entrée de commande x pour la sélection du sens de décalage, telle que :

- ✓ x=0, correspond au décalage à droite
- ✓ x=1, correspond au décalage à gauche

Rappelons que :

- ✓ Les équations d'un registre à décalage à droite, formé de 4 bascules D à front montant sont :

- ➢ $D_4 = Q_3$
- ➢ $D_3 = Q_2$
- ➢ $D_2 = Q_1$
- ➢ $D_1 = E_g$

- ✓ Les équations d'un registre à décalage à gauche, formé de 4 bascules D à front montant sont :

- ➢ $D_1 = Q_2$
- ➢ $D_2 = Q_3$
- ➢ $D_3 = Q_4$

- $D_4 = E_d$

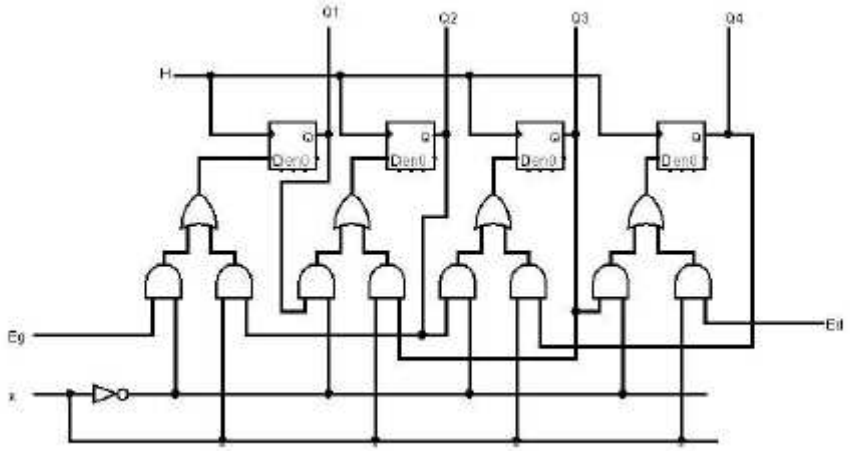
D'après la table de vérité :

x	D_1	D_2	D_3	D_4
0	E_g	Q_1	Q_2	Q_3
1	Q_2	Q_3	Q_4	E_d

Les équations d'un registre à décalage à droite ou à droite, formé de 4 bascules D à front montant sont :

- $D_1 = \bar{x}.E_g + x.Q_2$
- $D_2 = \bar{x}.Q_1 + x.Q_3$
- $D_3 = \bar{x}.Q_2 + x.Q_4$
- $D_4 = \bar{x}.Q_3 + x.E_d$

Le circuit :



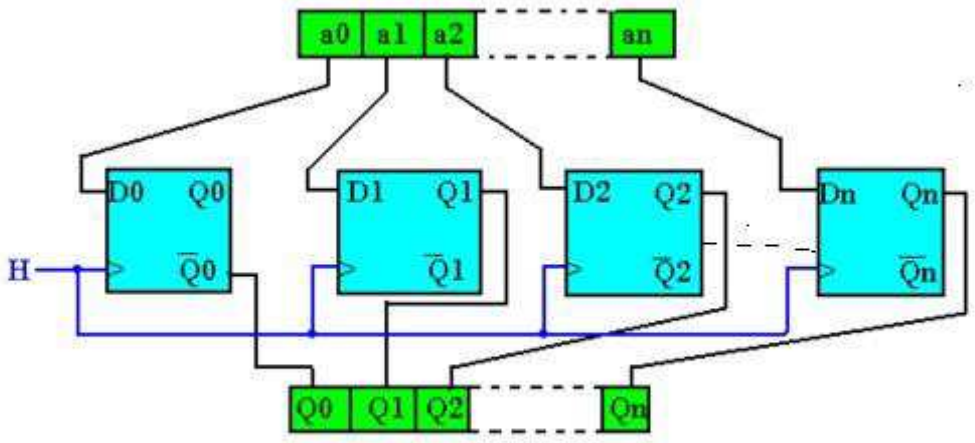
IV. Différents types de registres à décalage

On distingue quatre (04) types de registres à décalage :

- Registre à entrée parallèle/ sortie parallèle (Parallel In - Parallel Out) : PIPO
- Registre à entrée série/sortie série (Serial In Serial Out) : SISO
- Registre à entrée série/sortie parallèle (Serial In - Parallel Out) : SIPO
- Registre à entrée parallèle/ sortie série (Parallel In - Serial Out) : PISO

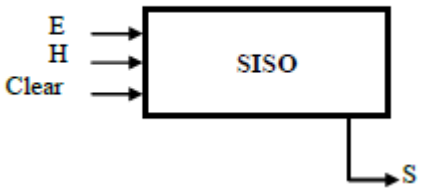
1) Registres à entrées parallèles /sorties parallèles « PIPO »

Dans ce type de registre, l'information est introduite en un seul bloc et récupérée de la même façon.

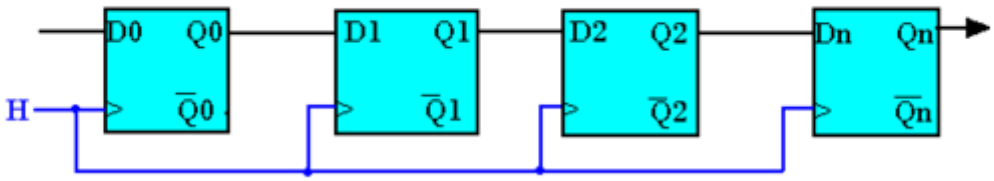


Ce type de registre est aussi appelé registre tampon. Il est souvent utilisé pour la mémorisation de données de durée brève ou pour le transfert de données.

2) Registres à entrées séries/sorties séries «SISO»



Dans ce type de registre, l'information est introduite bit par bit à l'entrée de la première bascule et se propagent à travers le registre à chaque impulsion d'horloge, pour sortir par la dernière bascule, c'est donc une sortie série.

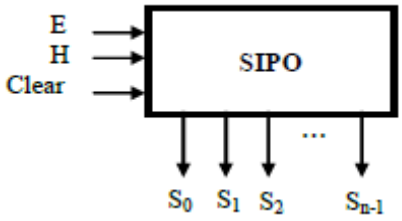


Remarques :

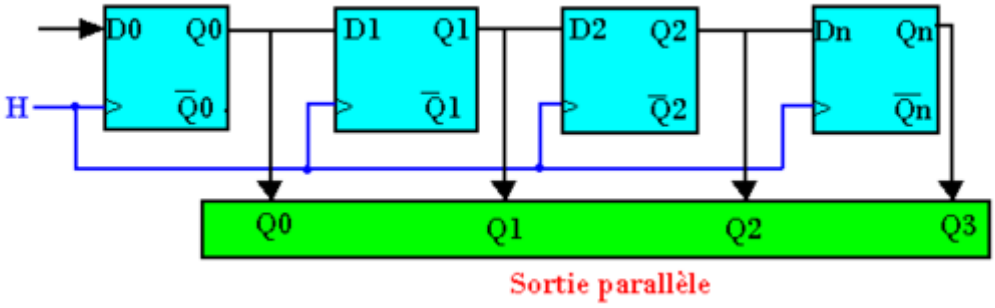
En rebouclant la sortie de la dernière bascule sur l'entrée de la première, on obtient ce qu'on appelle un "**compteur en anneau**". Pour charger une donnée 4 bits initiale sur les entrées D des bascules, il faut ajouter une logique de commande composée de quelques portes supplémentaires. Cette donnée se retrouver cycliquement sur les mêmes bascules.

En rebouclant la sortie complémentée $\bar{Q}$ de la dernière bascule sur l'entrée de la première, on obtient ce que l'on appelle un "**compteur Johnson**". Ce compteur possède un modulo égal à 2n, où n est le nombre de bascules.

3) Registres à entrées séries/sortie parallèle « SIPO »



C'est un type de registre dans lequel les données arrivent en série (bit par bit) et ressortent en parallèle (en un seul bloc).

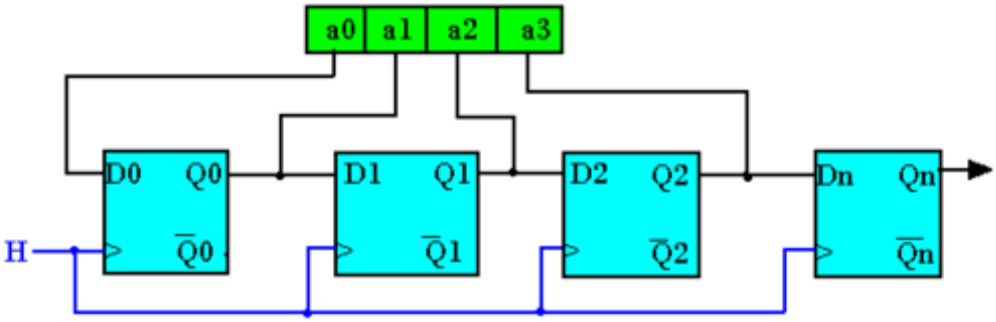


La transmission parallèle des informations d'un registre à un autre est la plus facile. La transmission série utilise peux d'élément donc peux coûteux.

4) Registres à entrées parallèles/ sortie série « (PISO »

Dans ce type de registre, l'information ou la donnée est introduite en un seul bloc, mais ne peut être récupérée que bit par bit.

Ces registres peuvent être utilisés pour faire une transformation parallèle-série des données.



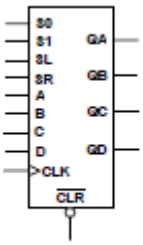
V. Registres universels

Il existe des circuits intégrés regroupant les quatre types de registres présentés ci-dessus.

Ils permettent les modes de fonctionnement suivants :

- chargement et lecture parallèles,
- chargement série et décalages à droite ou à gauche, lecture série ou parallèle,
- chargement parallèle et décalages à droite ou à gauche, lecture série ou parallèle.

Par exemple, le circuit intégré de référence 74194 possède la représentation symbolique suivante :



Les entrées A, B, C, D sont les entrées parallèles. Les entrées SL et SR sont respectivement les entrées/sorties séries gauches et droites. Les entrées S0 et S1 permettent de choisir le mode de fonctionnement de ce registre (blocage, décalage à droite, décalage à gauche, chargement parallèle). L'entrée CLR (active sur niveau bas) permet une remise à zéro asynchrone des sorties. L'entrée CLK est l'entrée horloge de synchronisation. Les sorties sont QA, QB, QC, QD.

VI. Application des registres

Un registre permet de mémoriser de façon temporaire un mot de N bits en attendant son traitement ultérieur. Chacune de ses bascules permet de mémoriser 1 bit.

Les registres sont utilisés dans les microprocesseurs pour des mémorisations temporaires des données.

Un registre de huit bits est constitué de 8 bascules synchronisées par la même horloge.

Un registre est un circuit permettant :

- de stocker une information binaire,
- de transférer une information dans certaines conditions,
- de faire des traitements simples sur les éléments binaires comme des décalages ou des rotations.

- Deux types de registres (PISO et SIPO) sont utilisés dans les liaisons séries ; ils forment la base des modems. Par exemple, si on veut transmettre une information

entre deux ordinateurs distants de quelques dizaines de mètres. Transmettre l'information en parallèle nécessite beaucoup de fils et très couteux. La solution est alors d'utiliser un registre PISO pour envoyer les bits sur une seule ligne. Au bout de laquelle, un registre SIPO reçoit ces bits et reconstitue des octets qui sont transmis à l'ordinateur de destination.

Bibliographie:

[1] : Circuits Numériques Théorie et Applications. υ Auteur : Ronald J. Tocci. Editeur : Reynald Goulet inc. Année : 1996. ISBN : 2-89377-108-4.

[2] : Cours et Problèmes d'Electronique Numérique. Auteur : Jean-Claude Laffont, Jean-Paul Vabre. Editeur : Edition Marketing. Année : 1986. ISBN :

[3] : Logique Combinatoire et Technologie. Auteurs : Marcel Gindre, Denis Roux. Editeur : BELIN. Année : 1984. ISBN : 2-7011-0857-8.

[4] : Systèmes Numériques. Auteurs : Jaccob Millman, Arvin Grabel. Editeur : McGRAW-HILL. Année : 1989. ISBN : 2-7042-1182-5.

[5] : Electronique Numérique. Auteurs : Rached Tourki. Editeur : Centre de publication Universitaire. Année : 2005. ISBN : 9973-37-019-8.

[6] : Support de cours de Systèmes Logiques. Auteurs : Mohamed Habib BOUJMIL. Année : 2004/2005.

[7] : Support Pédagogique de Systèmes Logiques.{ Auteurs : Fedia DOUIRI. Année : 2011/2012.

Sites Web:

http://didier.villers.free.fr/STI-2D/tronc-commun-activites.htm

http://pageperso.lif.univ-mrs.fr/~severine.fratani/enseignement/lib/exe/fetch.php?media=archi:td4- seq.pdf http://users.polytech.unice.fr/~fmuller/doc/ens/Peip2-SujetTP.pdf http://ensa-mecatronique.e-monsite.com/medias/files/compteurs-cor.pdf http://sebastien.bernard.free.fr/cours-tp-td-exo/TD-E-Logique-sequentielle-Fonction-Comptage.pdf

http://ressource.electron.free.fr/cours/Exercice_de_logique_sequentielle.pdf