Algèbre de Boole

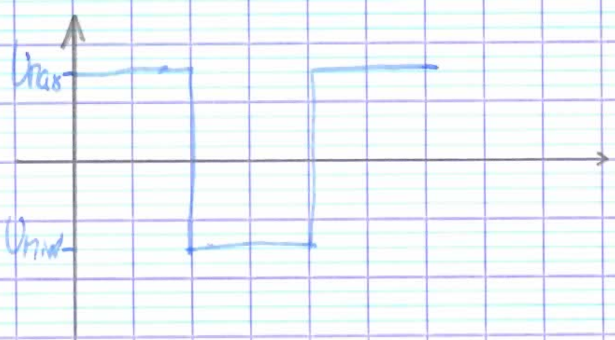
I Définitions

1- Variable binaire

: Variable susceptible de prendre 2 valeurs distinctes non simultanées

Ex : Porte ouverte ou fermée
Lampe allumée ou éteinte
Vrai / Faux
0 / 1

2- Types de Logique



- Logique positive :

* État 1 associé à U_{max}
* État 0 associé à U_{min}

- Logique négative :

* État 0 associé à U_{max}
* État 1 associé à U_{min}

II Opérateurs fondamentaux (Soient x, y, z variables binaires)

1- Opérateur NON

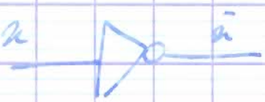
Notation : $S = \bar{x}$

Table de vérité :

x	$\bar{x}$
0	1
1	0

Symbole logique :

norme américaine :



(norme européenne :

)

2- Opérateur OU (inclusif) Somme logique

Notation : $S = x + y$

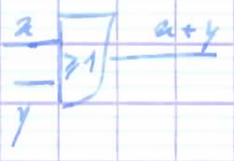
Table de vérité :

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

Symbole logique :

norme américaine :

(norme européenne :



)

3- Opérateur ET (exclusif) produit logique

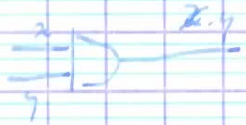
Notation : $S = x.y = xy$

Table de vérité

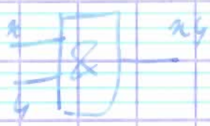
x	y	$x.y$
0	0	0
0	1	0
1	0	0
1	1	1

Symbole logique :

- norme américaine :



(norme européenne :



)

4- Propriétés des opérateurs fondamentaux

a- Élément neutre

. OU logique 0

. ET logique 1

b- Élément absorbant

. OU logique 1

. Et logique 0

c- Complémentarité

. $x + \bar{x} = 1$

. $x.\bar{x} = 0$

d- Idempotence

- $x + x = x$
- $x \cdot x = x$

e- Commutativité

- $x + y = y + x$
- $x \cdot y = y \cdot x$

f- Associativité

- $(x+y)+z = x+(y+z) = x+y+z$
- $x.(yz) = (xy)z = xyz$

g- Distributivité

- Distributivité du ET sur le OU :

$x.(y+z) = x.y + xz$

- Distributivité du OU sur le ET

$$x + y \cdot z = (x+y)(x+z)$$
$$= x.x + x.z + y.x + y.z$$
$$= x.1 + x.z + y.x + y.z$$
$$= x.\underbrace{(1+z+y)}_{1} + yz$$
$$= x + yz$$

4- Autres propriétés

- $\bar{\bar{x}} = x$
- Loi d'absorption : $x + xy = x = x.(x+y)$
- Multiples de complément

$$(x+\bar{y}).y = x.y$$
$$x\bar{y} + y = x + y$$

- Théorème de De Morgan :

$$\overline{x.y} = \bar{x} + \bar{y}$$
$$\overline{x+y} = \bar{x}.\bar{y}$$

Rq : Si $x + y = x + z \Rightarrow y = z$ Si $x = 1$, $y = 0$, $z = 1$

Si $x.y = x.z \not\Rightarrow y = z$

Si $x = 0$, $y = 0$, $z = 1$

Ex : Simplifier $S = (x+\bar{y})(x+y) + z(\bar{z}+y)$

$$S = \underbrace{xx}_{x} + xy + \bar{y}x + \underbrace{\bar{y}y}_{0} + z\bar{x} + zy$$ ← absorption

$$S = x + z\bar{x} + zy$$

$$S = (x+z)\underbrace{(x+\bar{x})}_{1} + zy$$ ← Multiple de complément de distributivité du ou sur le ET

$$S = x + \underbrace{z + zy}_{z}$$

$$S = x + z$$

III Fonctions usuelles

1) Fonction logique (= Fonction Combinatoire)

C'est une fonction qui donne l'état d'une ou plusieurs variables de sortie en fonction d'une combinaison des variables d'entrée.

Elle est définie à partir :

-> d'équations logiques écrites à l'aide des opérateurs fondamentaux ;

-> d'une table de vérité

-> d'un logigramme (= circuit logique dessiné avec les symboles logiques fondamentaux)

2) Fonctions usuelles

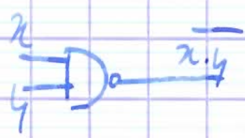
a) Fonction NON ET : Porte NAND

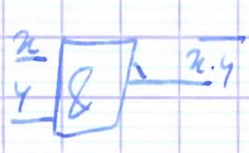
Équation : $S = \overline{x \cdot y}$

Table de vérité :

x	y	$\overline{x \cdot y}$
0	0	1
0	1	1
1	0	1
1	1	0

Symbole logique





b) Fonction NON OU : Porte NOR

Équation : $S = \overline{x+y}$

Table de vérité

x	y	$\overline{x+y}$
0	0	1
0	1	0
1	0	0
1	1	0

Symbole logique

c) Fonction OU EXCLUSIF : Porte XOR

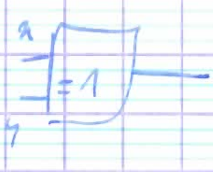
Équation et notation

$S = x \oplus y = x\bar{y} + \bar{x}y$

Table de vérité :

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Symbole logique :



d) Fonction ET INCLUSIF : Porte XNOR

équation et Notation :

$$S = x \odot y = \overline{x \oplus y} = \overline{x\bar{y} + \bar{x}y} = (\overline{x\bar{y}}).(\overline{\bar{x}y})$$
$$= (\bar{x}+y).(x+\bar{y}) = \bar{x}\bar{y} + xy$$

Table de vérité

x	y	$x \odot y$
0	0	1
0	1	0
1	0	0
1	1	1

Symbole logique :

IV Différents modes d'écriture d'une fonction

1) Forme Canonique

Toute fonction logique peut s'écrire sous la forme :

→ D'une somme logique de produits logiques, chaque produit devant contenir toutes les variables complémentées ou non. (1ère forme canonique)

→ D'un produit logique de sommes logiques, chaque somme devant contenir toutes les variables complémentées ou non (2e forme canonique)

Ex : $S = xyz + \bar{x}y\bar{z} + xy\bar{z}$

$S = x\bar{y} + \bar{x}y\bar{z}$

Q? Que représentent les produits de la 1ère forme canonique et les sommes de la 2e ?

1ère forme canonique :

1 pdt = 1 combinaison de variables pour laquelle la fonction vaut 1.

2ème forme canonique :

Somme = 1 combinaison de variables pour laquelle la fonction vaut 0

Ex :

x	y	S
0	0	1
0	1	0
1	0	0
1	1	1

1ère forme canonique :

$S = \bar{x} \cdot \bar{y} + xy$

2ème forme canonique :

$S = (x + \bar{y}) \cdot (\bar{x} + y)$

a	b	c	S'	
0	0	0	0	⑤
0	0	1	1	①
0	1	0	1	②
0	1	1	0	⑥
1	0	0	0	⑦
1	0	1	0	⑧
1	1	0	1	③
1	1	1	1	④

1ère forme canonique

$$S_1 = \underbrace{\bar{a}\bar{b}c}_{①} + \underbrace{\bar{a}b\bar{c}}_{②} + \underbrace{ab\bar{c}}_{③} + \underbrace{abc}_{④}$$

2ème forme canonique

$$S_1 = \underset{⑤}{(a+b+c)}\,\underset{⑥}{(a+\bar{b}+\bar{c})}\,\underset{⑦}{(\bar{a}+b+c)}\,\underset{⑧}{(\bar{a}+b+\bar{c})}$$

2) Images décimales

2^2	2^1	2^0	
a	b	c	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$S = (1, 2, 6, 7)_1$$

$$S = (0, 3, 4, 5)_0$$

Ex : $S(a, b, c, d) = (0, 2, 7, 9, 12)_1$

$$\begin{matrix} a & b & c & d \\ 2^3 & 2^2 & 2^1 & 2^0 \end{matrix}$$

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

3) Forme simplifiée

2 façons d'obtenir la forme simplifiée d'une fonction :

a) Simplification algébrique

→ On utilise les propriétés des opérateurs

b) Simplification géométrique

→ Méthode des tableaux de Karnaugh

. Tableau de Karnaugh

= table de vérité "dessinée" sous la forme d'un tableau à double entrée, chaque case doit être logiquement adjacente à ses voisines.

• Cases logiquement adjacentes :

= 2 cases sont dites adjacentes si les combinaisons des variables d'entrées qui leur correspondent ne diffèrent que d'un seul bit.

Ex : Tableau de Karnaugh à 3 variables

a \ bc	00	01	11	10
0				
1				

→

Remplissage du tableau de Karnaugh

a	b	c	d
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

a \ bc	00	01	11	10
0	0 (0)	1 (1)	0 (3)	1 (2)
1	0 (4)	0 (5)	1 (7)	1 (6)

$\bar a \bar b c$ $\qquad$ $b\bar c$

$a b c + a b \bar c = ab(c+\bar c)$
$= ab$

$\Rightarrow S = ab + b\bar c + \bar a \bar b c$
$= b(\underbrace{a+\bar c}_{\overline{\bar a c}}) + \bar b(\bar a c)$
$= b \oplus (\bar a . c)$

⚠ Nombre de cases que l'on peut regrouper = Puissance de 2

ab \ cd	00	01	11	10
00	1 (0)	1 (1)	0 (3)	1 (2)
01	0 (4)	1 (5)	0 (7)	0 (6)
11	0 (12)	1 (13)	1 (15)	1 (14)
10	1 (8)	0 (9)	1 (11)	1 (10)

$S = \bar b \bar d + ac + \bar a \bar c d + abd$

Cas des fonctions incomplètement définies

Si certaines combinaisons de variables d'entrée ne peuvent pas se produire, on peut alors choisir, pour ces combinaisons la valeur de sortie que l'on veut

$\Rightarrow$ État indifférent : X

Ex :

D	C	B	A	&
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1

DC \ BA	0 0	0 1	1 1	1 0
0 0	0_{0}	0_{1}	1_{3}	0_{2}
0 1	0_{4}	0_{5}	0_{7}	1_{6}
1 1	X_{12}	X_{13}	X_{15}	X_{14}
1 0	0_{8}	1_{9}	X_{11}	X_{10}

$$S = DA + CB\bar{A} + \bar{C}BA$$
$$= DA + B(C\bar{A} + \bar{C}A)$$
$$= DA + B(C \oplus A)$$

V Résolution de problèmes

Etapes de résolution :

- Identification des variables d'entrée et de sortie
- Traduction du cahier des charges en table de vérité
- Recherche des équations simplifiées des sorties
- Conception du circuit logique.

Ex : 1) Remplissage des soutes d'un pétrolier

Variables d'entrée a, b, c
Variables de sortie

a	b	c	V
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

a \ bc	00	01	11	10
0	1	0	0	1
1	0	1	1	0

$V = ac + \bar{a}\bar{c}$

$= a \odot c$

2)

a	b	c	d	V
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

ab \ cd	00	01	11	10
00	0	0	1	0
01	0	0	1	1
11	0	1	1	1
10	0	0	1	1

$S = cd + bc + ac + abd$

a	b	c	d	V_1	V_2	V_3
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	X	X	X
0	1	0	1	0	0	1
0	1	1	0	X	X	X
0	1	1	1	0	0	1
1	0	0	0	X	X	X
1	0	0	1	X	X	X
1	0	1	0	0	0	1
1	0	1	1	0	0	1
1	1	0	0	X	X	X
1	1	0	1	X	X	X
1	1	1	0	X	X	X
1	1	1	1	1	0	0

R_1 + a, + c R_2 + b, + d

a = 1 et c = 0 ⇒ Impossible
b = 1 et d = 0

V_1

ab \ cd	00	01	11	10
00	0	0	0	0
	X	0	0	X
	X	X	1	X
	X	X	0	0

V_2 $V_1 = ab$

ab \ cd	00	01	11	10
00	1	0	0	0
01	X	0	0	X
11	X	X	0	X
10	X	X	0	0

$V_2 = \bar{c}\bar{d}$

V_3 $= \overline{V_1 + V_2} = \bar{V_1} \cdot \bar{V_2}$

ab \ cd	00	01	11	10
00	0	1	1	1
01	X	1	1	X
11	X	X	0	X
10	X	X	1	1

$V_3 = \bar{a}d + \bar{b}c$

$= (\bar{a} + \bar{b})(c + d)$

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

$= \bar{a}(c+d) + \bar{b}(c+d)$

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

Solutions évidentes

$H' = G' = A' = 0$

F'

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

$F' = DC + DB$

E'

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

$E' = \bar{D}CA + \bar{D}CB + CBA + D\bar{C}\bar{B}$

D'

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

$D' = \bar{D}C\bar{B}\bar{A} + DCB\bar{A}$

C'

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

$C' = \bar{D}\bar{C}B + \bar{D}AB + DC\bar{B} + D\bar{A}\bar{B}$

B'

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

$B' =$

VI Fonctions combinatoires

1) Définition

C'est une fonction où les sorties dépendent uniquement des combinaisons des variables d'entrée

-> pas d'effet mémoire

-> le temps n'intervient pas.

2) Fonctions arithmétiques

Cf. TD

3) Fonctions de Transcodage

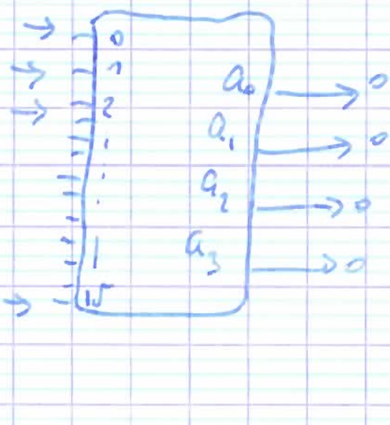
Def : Fonction qui transfère une information donnée en entrée sous 1 forme A (code A) en son équivalent en sortie sous 1 forme B (code B).

a) Codeur

= codage binaire d'une information décimale (code A)
(code B)

=> Nb. d'entrée 2^n

Nb de sortie : n



b) Décodeur

= Décodage décimal (ou hexa) (= code B) d'une info binaire (code A)

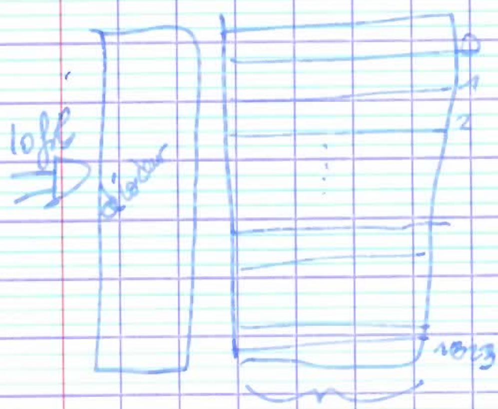
n entrées ⇒ 2^n sorties

Ex d'application : Adressage d'une mémoire

Mémoire = Tableau d'éléments binaires

1 mot-mémoire est contenu dans 1 ligne.

1 mot est repéré par 1@



Mémoire de 1024 mots

c) Transcodeur

= Convertisseur d'un code quelconque à un autre code.

Décimal ↔ Hexa

Binaire ↔ BCD

Code Grey ↔ Binaire

BCD ↔ Afficheur 7 segments

(Permet de définir les fonctions d'allumage des 7 segments)

4) Fonctions d'aiguillage

1) Définitions

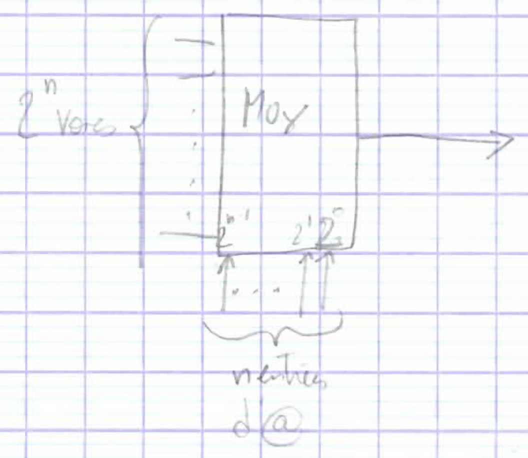
Multiplexage :

Regroupe, en série, sur 1 voie, les signaux venant de n voies en //.

Démultiplexage :

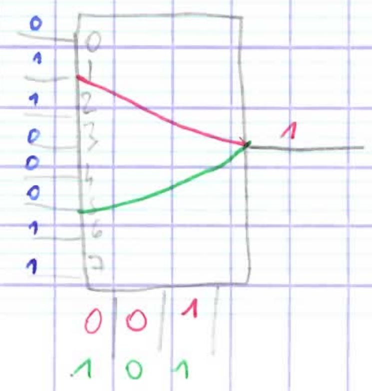
Aiguille vers n voies en //, les signaux venant en série d'une seule voie

2) Multiplexeur

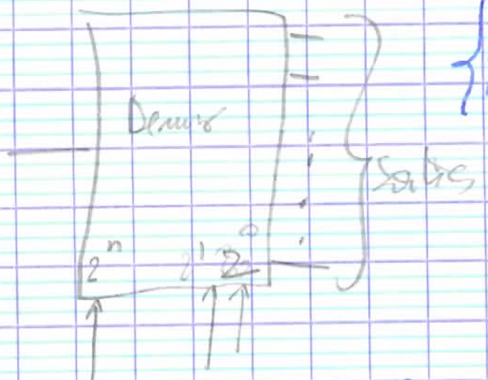


$\left\{ \begin{array}{l} 2^n \text{ entrées d'informations} \\ 1 \text{ sortie} \\ n \text{ entrées d'@} \end{array} \right.$

Ex : Mux 8/1 $\Rightarrow$ $\left\{ \begin{array}{l} 8 \text{ entrées d'info} \\ 1 \text{ sortie} \\ 3 \text{ entrées d'@} \end{array} \right.$



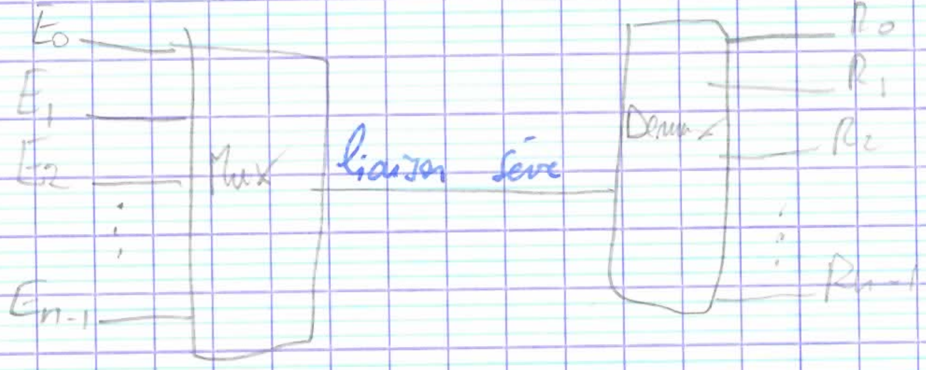
3) Démultiplexeur



- 1 entrée d'info
- 2^n sorties
- n entrées d'@

Ex : Demux 1/16 :
- 1 entrée
- 16 sorties
- 4 entrées d'@

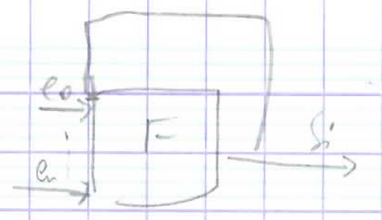
4) Application : Transmission série



VII Fonctions Séquentielles

1) Définition

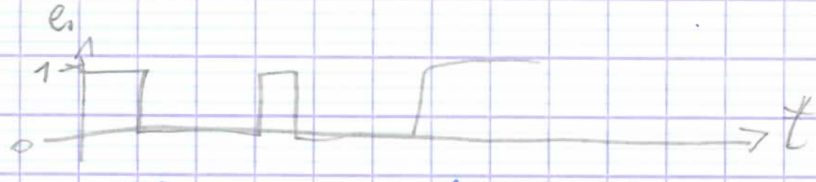
= Circuit où les sorties dépendent des combinaisons d'entrée et de l'historique des sorties.
À une combinaison de variables d'entrée peuvent correspondre plusieurs combinaisons de sortie. Le temps intervient.
→ Effet mémoire.



$$S(t+\varepsilon) = f(e_i, s(t))$$

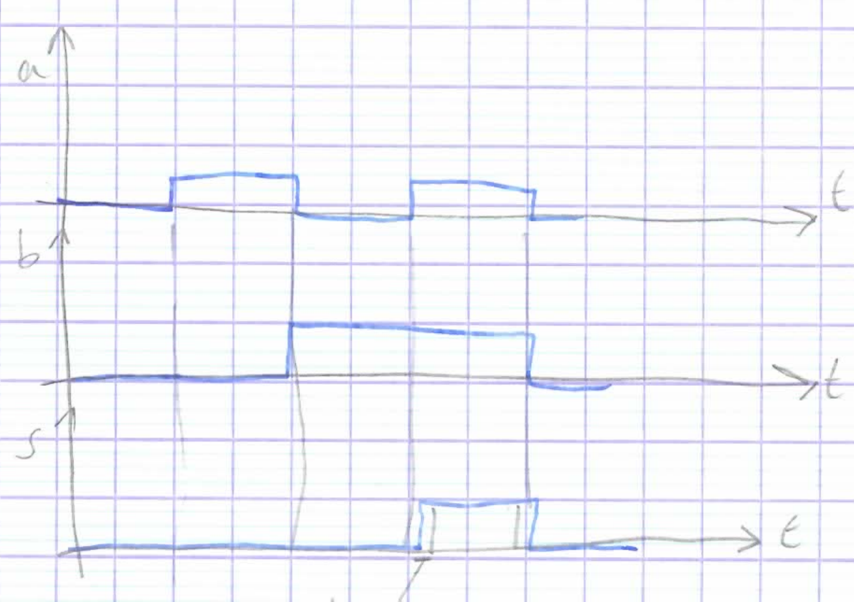
. Chronogramme

- Représentation de l'évolution des variables d'entrée et de sortie en fonction du tps :



Ex : Chronogramme fonction ET

$S = a.b$



temps de réponse de la porte ET

* Modes asynchrones et synchrones :

Mode asynchrone

Les ordres appliqués sur les entrées provoquent immédiatement le changement d'état correspondant en sortie

Mode synchrone

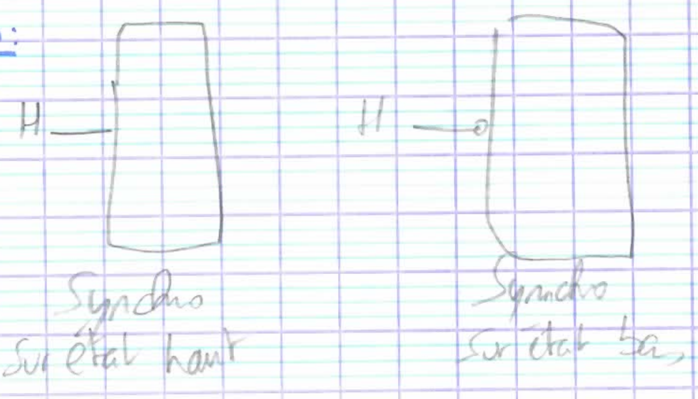
L'exécution de l'ordre n'intervient qu'avec 1 signal de synchronisation l'horloge (H ou CK)

∃ ≠ modes de synchronisation

→ Synchronisation sur état

La prise en compte et l'exécution de l'ordre en entrée s'effectue quand :
. H = 1 (synchro sur état haut)
. H = 0 (synchro sur état bas).

Notation :

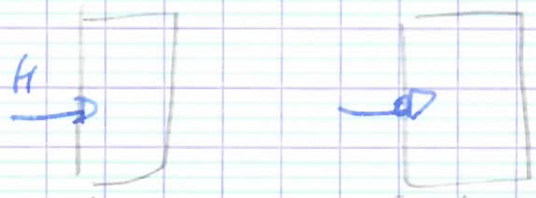


→ Synchronisation sur front

Front = transition État Bas/État haut.
(front montant).
ou État Haut/État bas
(front descendant).

La prise en compte et l'exécution de l'ordre s'effectue lors
. des fronts montants
. des fronts descendants

Symbole :



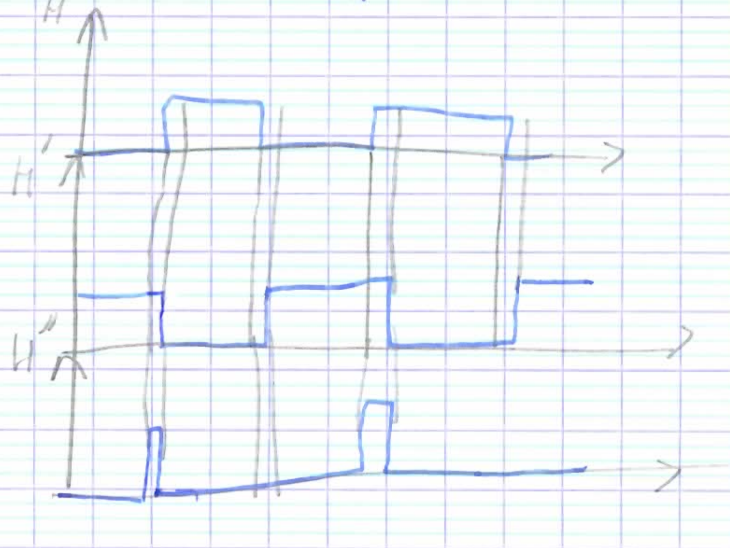
Synchro sur front montant

Synchro sur front descendant

Rq

$H' = \bar{H}$

$H'' = \bar{H} . H = 0$



→ Synchronisation sur impulsion
(Circuit Maître - Esclave)

La prise en compte de l'ordre se fait sur 1 front (montant ou descendant) mais l'exécution se fait qu'au front suivant (descendant ou montant)

Mem : Front ↑ App. front ↓

⇒ Synchro sur impulsion > 0

Mem : Front ↑ App front ↑

⇒ Synchro sur impulsion < 0

Bascules : Circuits séquentiels de base

Symboles :

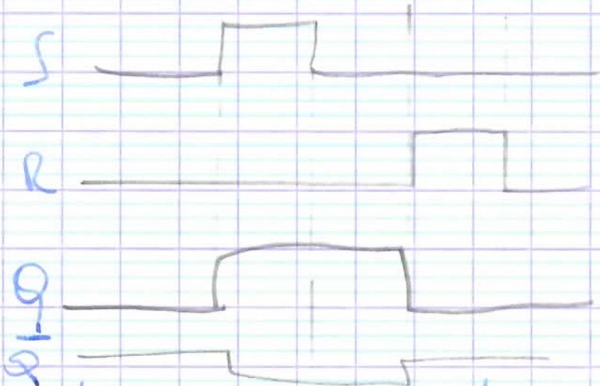
synchro sur impulsion > 0 — synchro sur impulsion < 0

. Entrées synchrones et asynchrones
_ Entrées synchrones : nécessitent le signal d'horloge pour être prises en compte
_ Entrées asynchrones : les ordres appliqués sur les entrées sont pris en compte dès leur apparition ∀ H
∀ état des entrées synchrones
On les appelle aussi entrées de forçage.
Elles sont prioritaires sur les autres entrées.

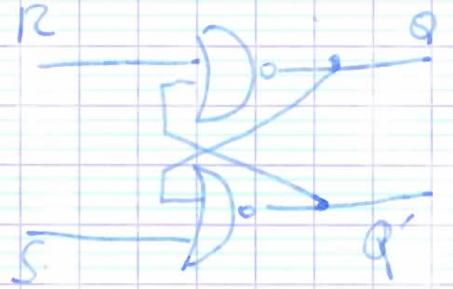
2) Bascule RS asynchrone

principe de fonctionnement

- 2 entrées : R et S
- 2 sorties : Q et Q'



a) Réalisation avec des portes NOR

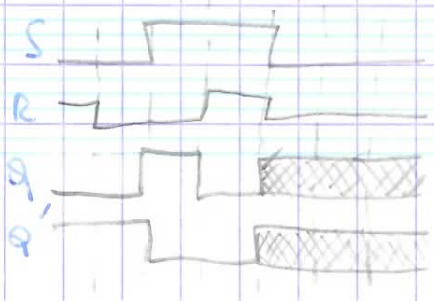


$Q' = \overline{Q}$

R	S	Q(n+1)	Q'(n+1)	
0	0	Q(n)	$\overline{Q}$(n)	← Etat mémoire
0	1	1	0	← Mise à 1
1	0	0	1	← Mise à 0
1	1	0	0	← Etat Interdit.

Q(n+1) = Sortie Q à l'instant (n+1)

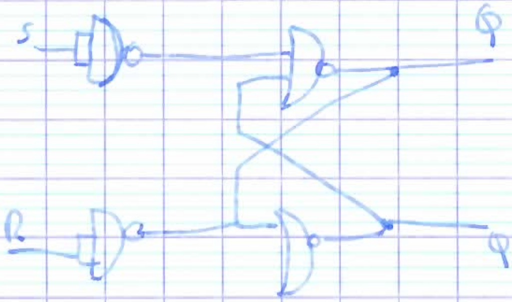
Rq :



État mémoire qui succède à 1 état interdit
→ Cas indéterminé

- Bascule RS réalisée avec des portes NOR = Bascule à arrêt prioritaire

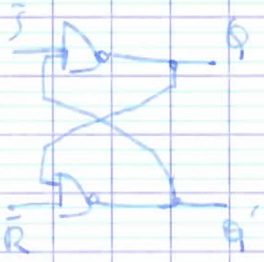
b) Réalisation avec des portes NAND



$Q' = \overline{Q}$

R	S	Q(n+1)	Q'(n+1)	
0	0	Q(n)	$\overline{Q}$(n)	← État mémoire.
0	1	1	0	← Mise à 1
1	0	0	1	← Mise à 0
1	1	1	1	← État interdit

Rq : Bascule RS réalisée avec des portes NAND = Bascule à marche prioritaire

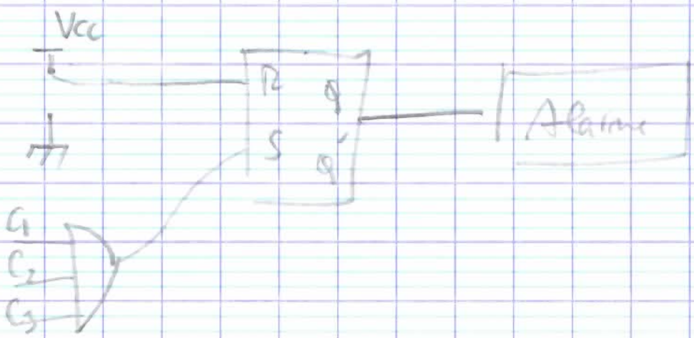


Bascule $\bar{S}\bar{R}$

c) Application : Réalisation d'une alarme

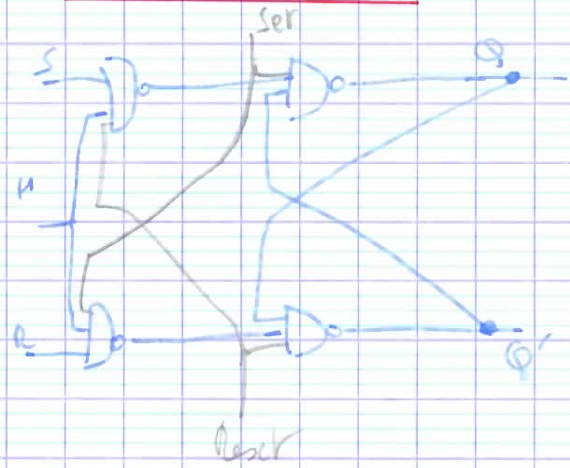
Cahier des charges
- Elle doit disposer de 3 capteurs C_1, C_2, C_3. Dès que l'un d'entre eux est activé, l'alarme doit se déclencher.
- L'alarme s'arrête quand 1 bouton poussoir est appuyé, sauf si 1 des capteurs est encore actif.

Choix de la bascule : Marche prioritaire



3) Bascules RS synchrones

a. Synchro sur état



SR = Entrées synchrones

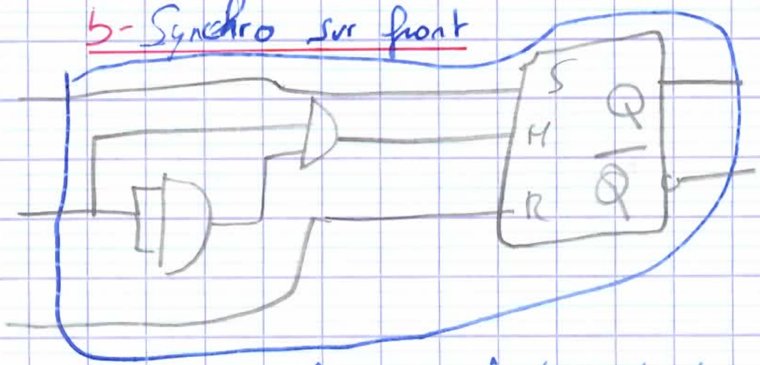
Set, Reset = Entrées asynchrones

H = Entrée de synchronisation

fonctionnement synchrone de la bascule

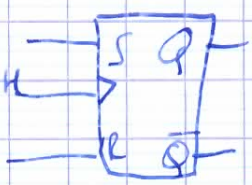
H	S	R	Set	Reset	$Q(n+1)$	$\bar{Q}(n+1)$	
0	X	X	1	1	$Q(n)$	$\bar{Q}(n) = \overline{Q(n)}$	
1	0	0	1	1	$Q(n)$	$\bar{Q}(n)$	Bascule RS
1	0	1	1	1	0	1	
1	1	0	1	1	1	0	
1	1	1	1	1	1	1	
X	X	X	0	1	1	0	← Forçage à 1
X	X	X	1	0	0	1	← Forçage à 0
X	X	X	0	0	1	1	← État Interdit

b- Synchro sur front



RS Synchro sur front montant

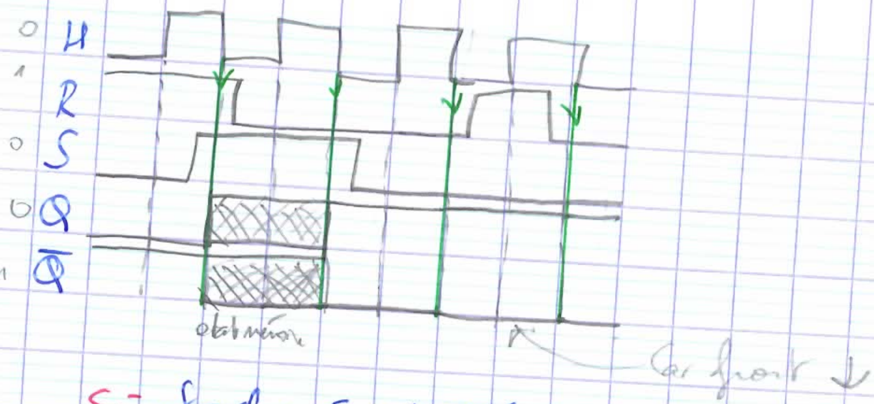
Symbole :



Synchro sur front ↑

H	R	S	Q_{n+1}	$\bar{Q}_{n+1}$
$\ast$ (H sauf un front montant)	X	X	Q_n	$\bar{Q}_n$
↑	0	0	Q_n	$\bar{Q}_n$
↑	0	1	1	0
↑	1	0	0	1
↑	1	1	État	Interdit

Ex : Bascule Synchronisée sur un front descendant (marche prioritaire)

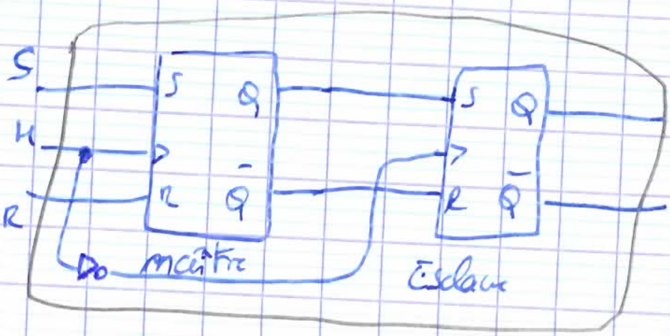


c - Synchro sur impulsion

Ex : Synchro sur impulsion >0

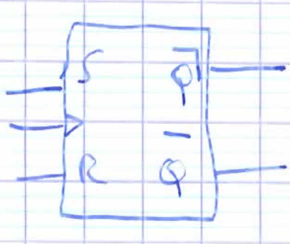
Rappel : Mémorisation → Front ↑

Exécution → front ↓



Bascule RS synchro sur impulsion >0

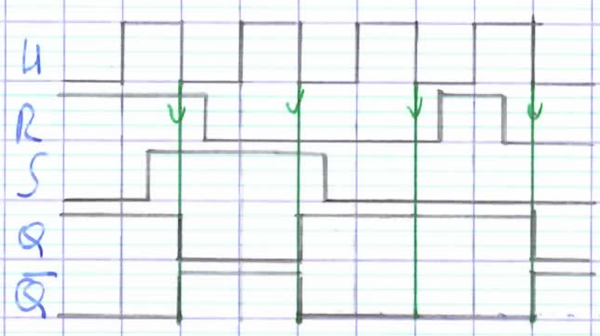
Symbole :



Synchro sur impulsion >0

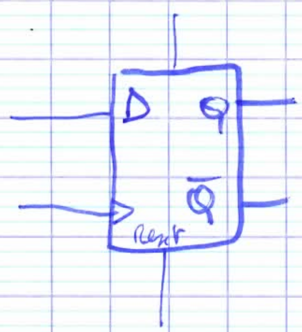
Synchro sur impulsion <0

Ex : Bascule synchronisée sur impulsion > 0



4) Bascules D

Symbole :



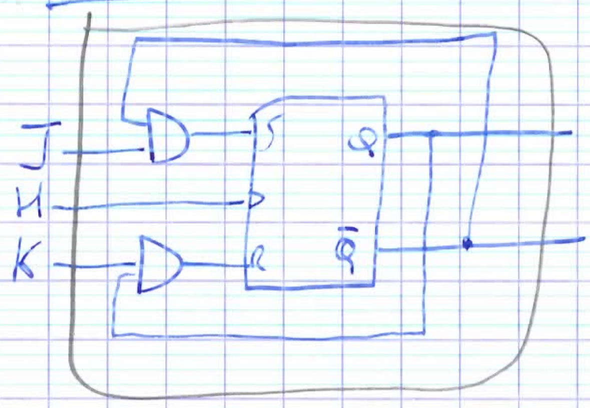
Cas d'un bascule D synchronisée sur front ↑ avec entrées asynchrones actives au niveau haut

Table de vérité

H	D	Set	Reset	Q_{n+1}	$\bar{Q}_{n+1}$
$\not\uparrow$	x	0	0	Q_n	$\bar{Q}_n$
↑	0	0	0	0	1
↑	1	0	0	1	0
X	X	0	1	0	1
X	X	1	0	1	0
X	X	1	1	État	Interdit

A chaque signal actif de H, la sortie recopie l'entrée

Bascules JK



2 entrées synchrones
2 Sorties
1 Signal d'horloge

Table de vérité

	H	J	K	Q_{n+1}	$\overline{Q_{n+1}}$
	$\not\uparrow$	X	X	Q_n	$\overline{Q_n}$
Etat mémoire	$\uparrow$	0	0	Q_n	$\overline{Q_n}$
mise à 0	$\uparrow$	0	1	0	1
mise à 1	$\uparrow$	1	0	1	0
Commutation	$\uparrow$	1	1	$\overline{Q(n)}$	$Q(n)$

. $J = K = 0 \Rightarrow S = R = 0$

↳ État mémoire

. $\left.\begin{array}{l} J = 0 \\ K = 1 \end{array}\right\} \Rightarrow \begin{array}{l} S = 0 \\ R = Q_n \end{array}$ ↗ $Q_n = 0$ état mémoire $Q_{n+1} = 0$

↘ $Q_n = 1$ mise à 0 $\Rightarrow Q_{n+1} = 0$

. $\left.\begin{array}{l} J = 1 \\ K = 0 \end{array}\right\} \Rightarrow \left\{\begin{array}{l} S = \overline{Q_n} \\ R = 0 \end{array}\right.$ ↗ $\overline{Q_n} = 0 \Rightarrow$ état mémoire $Q_{n+1} = 1$ ($Q_n = 1$)

↘ $\overline{Q_n} = 1 \Rightarrow$ Mise à 1 $Q_{n+1} = 1$

. $J = K = 1$: $S = \overline{Q_n}$, $R = Q_n$

↗ Si $Q_n = 0$ $\left\{\begin{array}{l} S = 1 \\ R = 0 \end{array}\right.$ $\Rightarrow$ Mise à 1 $Q_{n+1} = 1$

↘ Si $Q_n = 1$ $\left\{\begin{array}{l} S = 0 \\ R = 1 \end{array}\right.$ $\Rightarrow$ Mise à 0 $Q_{n+1} = 0$