Initiation aux Systèmes d'Information Numériques

DUT GEII 1ère année 2019-2020

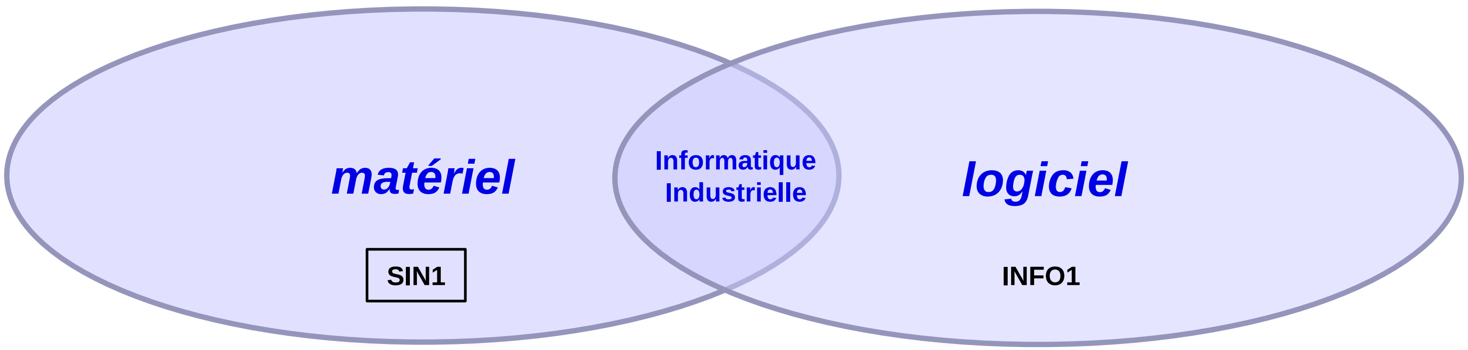
Thierry FIOL

thierry.fiol@umontpellier.fr

G.E.I.I. = Génie Electrique et Informatique Industrielle

L'Informatique industrielle est une branche de l'informatique appliquée qui couvre l'ensemble des techniques de conception, d'analyse et de programmation de systèmes informatiques à vocation industrielle…

Les appareils concernés contiennent en général au moins un microprocesseur ou un microcontrôleur, ainsi que des coupleurs d'interfaçage entre des machines ou appareillages industriels et de l'informatique. WIKIPÉDIA L'encyclopédie libre



INFO2,

ARS2

Planning prévisionnel

		Module SIN1 - Planning prévisionnel 2019-2020			
1°A	Mois	CM (1,5h)	TD (1,5h)	TP (1,5h)	Test court (1h)
S1.1	Sept.	Cours N°1: algèbre de Boole			
S1.2		Cours N° 2: Méthode de synthèse des systèmes logiques combinatoires			
S1.3		Cours N° 3: Méthode de Karnaugh	TD1: algebre de Boole / circuits à contacts		
S1.4		Cours N° 4:Fonctions incomplètement spécifiées	TD1 suite: Synthèse syst log. Comb		
S1.5	Oct.	Cours N° 5: Portes logiques électroniques	TD2: Simplif. Par Karnaugh	TP 1: Cedar Alg boole	TC1: combinatoire
S1.6		Cours N° 6: Systèmes de numération	TD3: Fn log. Inc. Spéc.	TP 2: Câblage / Cedar Alg Boole	
S1.7		Cours N° 7: Arithmetique binaire	TD3: Fn log. Inc. Spéc.	TP 3: Cedar Synthese syst log comb	
S1.8		Cours N° 8: Introduction aux systèmes logiques séquentiels	Test C/TD1	Test TP1	
S1.9	Nov.	Cours N° 9: Méthode d'Huffman	TD5 Logigrammes	TP4: Cedar / comb circuit mémoire	
S1.10		Cours N° 10: Méthode d'Huffman	TD6: Systèmes de numération	TP4: Cedar / comb circuit mémoire	TC2: Numération
S1.11		Cours N° 11: Bascules - Synthèse systèmes synchrones	TD6: Systèmes de numération	TP5: Huffman circuit mémoire	
S1.12		Cours N° 12: Bascules - Synthèse systèmes synchrones	Test C/TD2	TP6: Huffman circuit mémoire 2	
S1.13	Dec.		TD N° 7: Huffman N° 1	Test TP2	
S1.14			TD N° 8: Huffman N° 2	TP7 Mot pas-à-pas	
S1.15			Test C/TD3	TP8 FPGA1:	
	Janv.				
S1.16				TP8 FPGA1:	
S1.17				TP9 FPGA2:	
S1.18				Test TP2	

Quelques ressources utiles:

http://infoindustrielle.free.fr/Logique/page_guide.htm

http://thierryperisse.free.fr/documents/electronique_numerique/CNED/POLY%20CNED%201.pdf

http://www.iutenligne.net/

Chapitre 1: Etude des systèmes logiques combinatoires

- Information, système, traitement, signal …

- Variables et fonctions logiques

- Méthode de synthèse des systèmes logiques combinatoires

- Algèbre de Boole et simplification des fonctions logiques

- Application au câblage de circuits électriques commutés

- Application à réalisation de systèmes électroniques

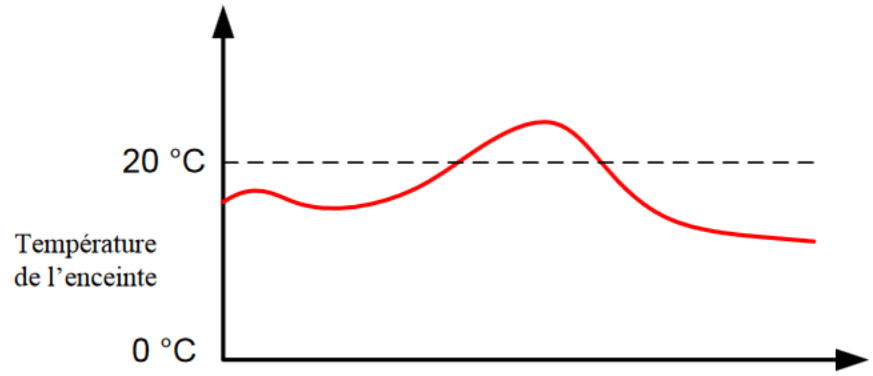
Définitions:

Information: Élément de connaissance susceptible d'être représenté à l'aide de conventions pour être conservé, traité ou communiqué.

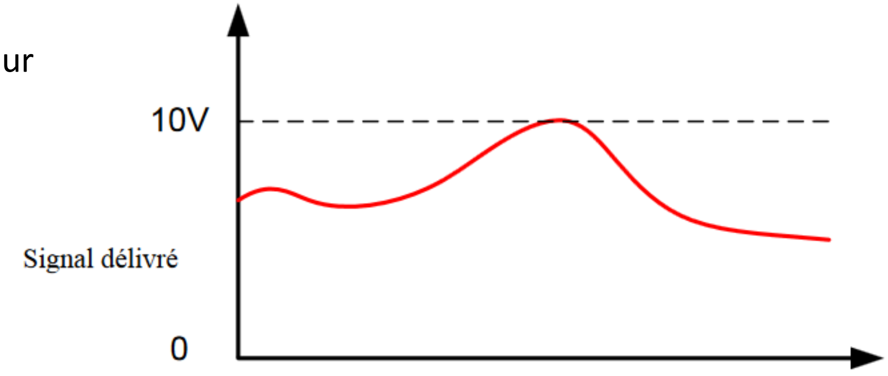
Signal: Variation d'une grandeur physique de nature quelconque, transportant de l'information, et grâce à laquelle, dans un système, un élément en influence un autre.

3) Signal analogique :

Un signal analogique varie de façon continue dans le temps. Il peut prendre une infinité de valeurs dans une plage donnée.

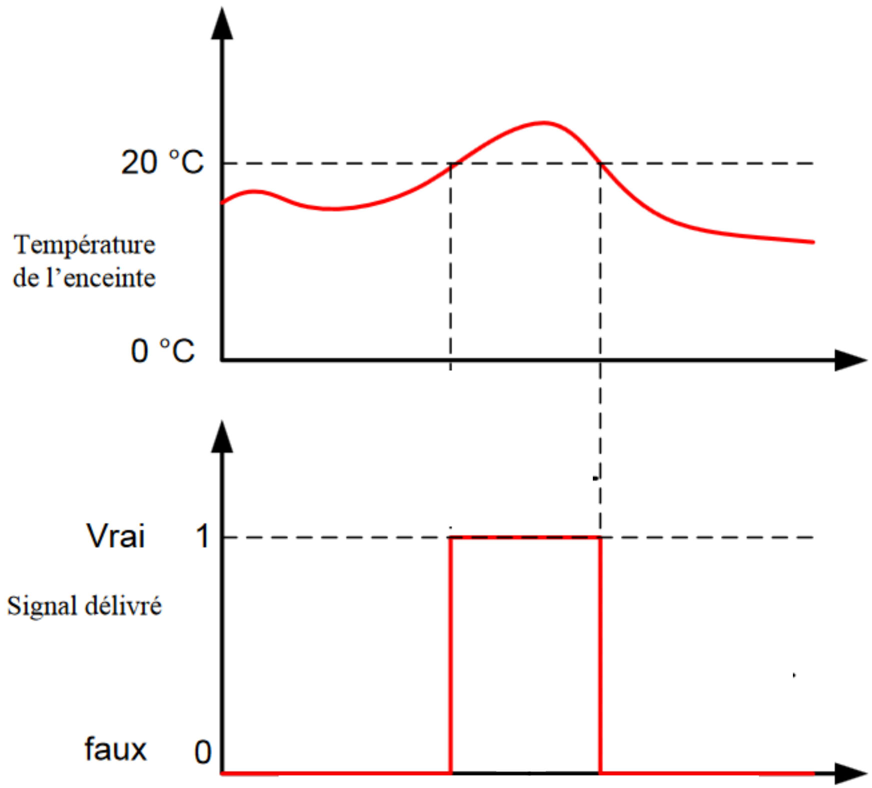


Exemple : la sortie (0/10V) d'un Capteur de température transmet l'image de la température de l'enceinte.



2) **Signal logique** :

Le signal logique ou Tout Ou Rien peut prendre deux valeurs.



Exemple : la sortie logique d'un thermostat transmet deux informations : la température est supérieure à la consigne (à la valeur attendue) ou la température est inférieure à la consigne.

Vision « Système »

système

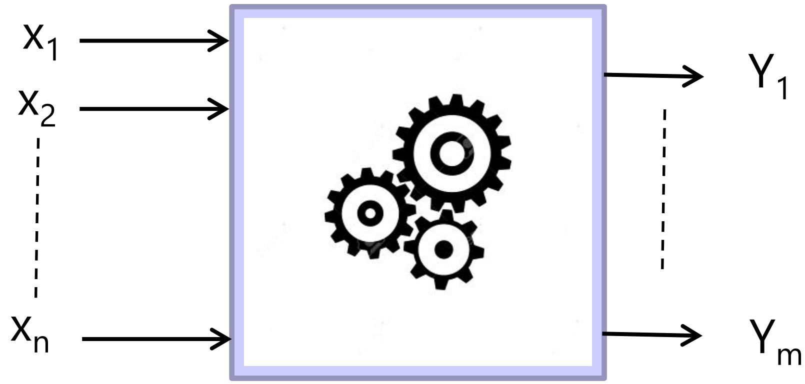
nom masculin

1. Ensemble abstrait dont les éléments sont coordonnés par une loi, une théorie.
   Le système astronomique de Copernic.
2. Ensemble de pratiques organisées en fonction d'un but.
   Le système de défense d'un accusé.
   synonymes : méthode
3. Ensemble de pratiques et d'institutions.
   Système politique, social.
4. **Esprit de système** tendance à organiser, à relier les connaissances en ensembles cohérents ; *péjoratif* tendance à faire prévaloir la conformité à un système sur une juste appréciation du réel.
5. Ensemble complexe d'éléments naturels de même espèce ou de même fonction.
   Le système solaire.
   synonymes : structure
6. Dispositif ou appareil complexe mis en œuvre pour aboutir à un résultat.
   Système de miroirs.
7. Ensemble structuré (de choses abstraites).

Système logique = opérant sur des informations logiques

informations logiques fournies par l'environnement du système

(variables logiques d'entrée)



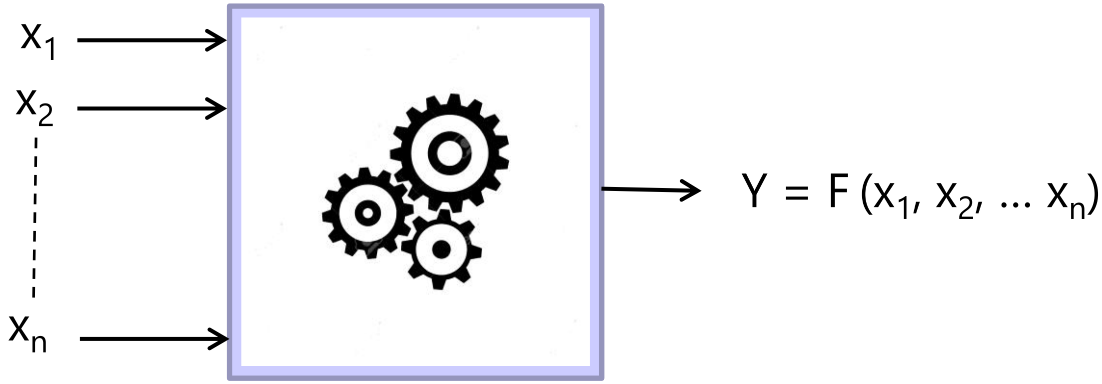
informations logiques produites par le système …

(variables logiques de sorties)

Variable logique: une variable qui ne peut prendre que deux valeurs (souvent 0/1)
On dit aussi variable binaire ou encore variable booléenne.

La signification de chacune des deux valeurs doit être conventionnellement définie pour chaque variable du système.

Fonctions logiques combinatoires

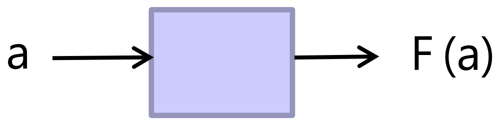


Une **fonction logique combinatoire** F des ***n*** variables logiques $(x_1, x_2, \ldots x_n)$, notée $Y = F(x_1, x_2, \ldots x_n)$, associe une valeur 0 ou 1 aux différentes combinaisons possibles des ***n*** variables logiques .

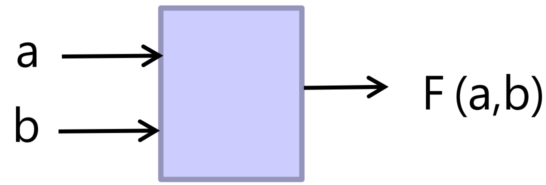
Chaque variable logique x_i pouvant prendre la valeur 0 ou 1, il y a au total 2^n combinaisons possibles des variables logiques $(x_1, x_2, \ldots x_n)$ et on définit complètement une fonction logique en donnant sa valeur pour chacune de ces combinaisons.

Une manière efficace de représenter le comportement d'une fonction logique combinatoire consiste à établir sa **table de vérité**.

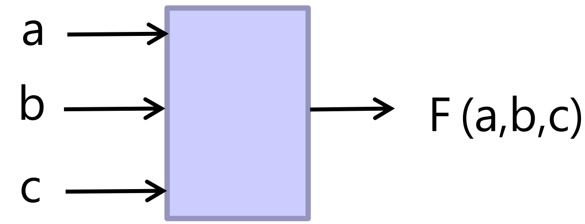
Enumération des 2^n combinaisons de valeurs des variables d'entrée et tables de vérités associées



a	F(a)
0	
1	

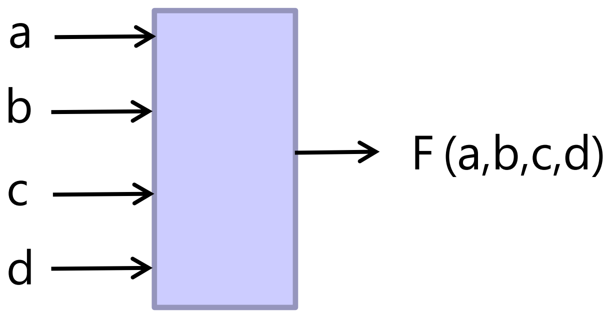


a	b	F(a,b)



a	b	c	F(a,b,c)

Enumeration des 2^n combinaisons de valeurs des variables d'entrée

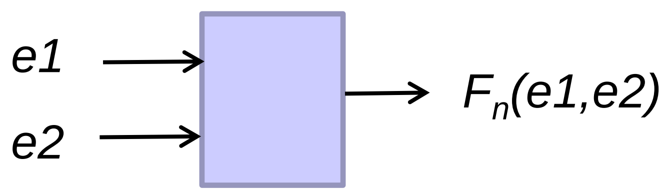


a	b	c	d	F(a,b,c,d)

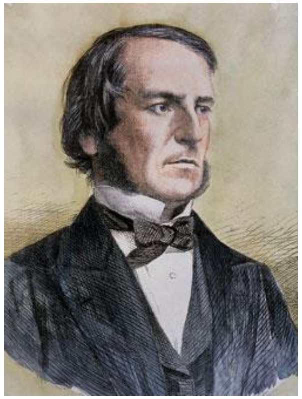
Liste des fonctions logiques combinatoires de deux variables

Il existe 16 fonctions logiques combinatoires de 2 variables …

e1	e2	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

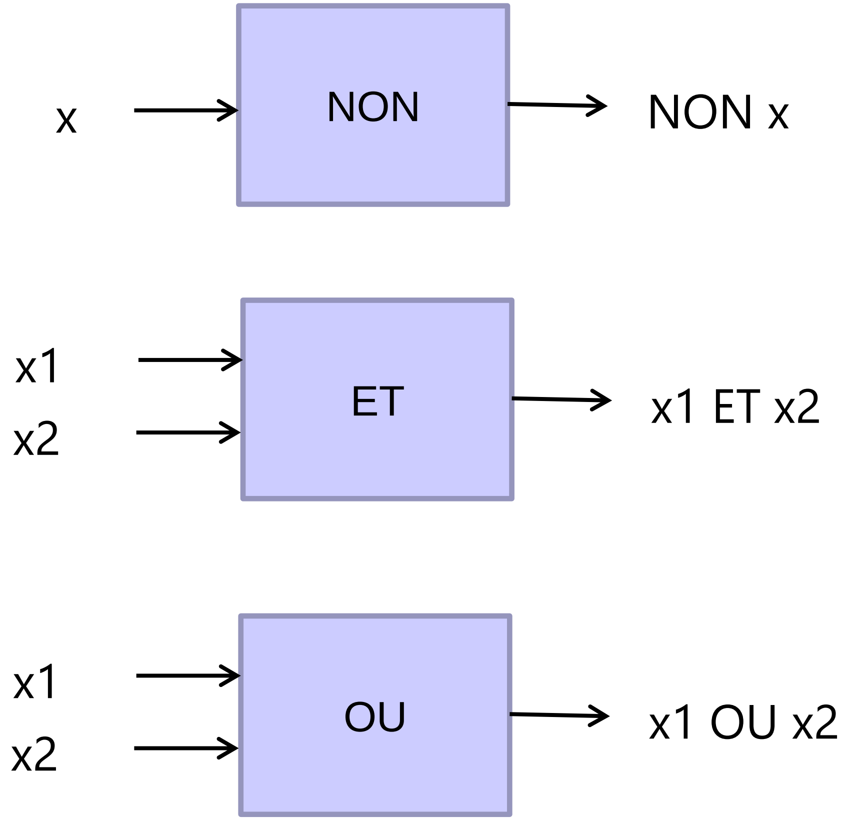


Algèbre de Boole ou algèbre de la logique

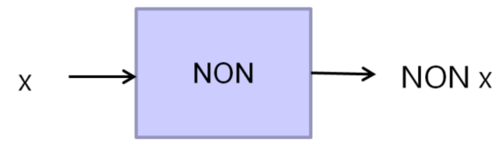


George Boole
1815-1864

Les lois de
la pensée



Fonctions logiques de base



Fonction inversion NON (NOT).

Cette fonction est également appelée complément

Notation : $F = \overline{x}$

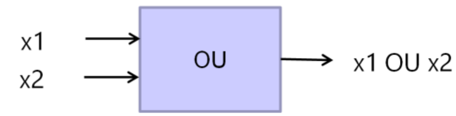
Table de vérité

x	$F = \overline{x}$
0	1
1	0

Fonctions logiques de base (2)

Fonction OU (OR).

C'est une fonction de deux variables également appelée somme logique



Notation : $F = x + y$

Table de vérité

x y	$F = x + y$
0 0	0
1 0	1
1 1	1
0 1	1

Relations caractéristiques :

$$x + 0 = x$$

$$x + 1 = 1$$

$$x + x = x$$

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

La fonction **OU** vaut 1 si au moins une des variables vaut 1.

Fonctions logiques de base (3)

Fonction ET (AND).

C'est une fonction de deux variables également appelée produit logique

Notation : $F = x \bullet y$

Table de vérité

x y	$F = x \bullet y$
0 0	0
1 0	0
1 1	1
0 1	0

Relations caractéristiques :

$$x \bullet 0 = 0$$

$$x \bullet 1 = x$$

$$x \bullet x = x$$

$$x \bullet \overline{x} = 0$$

La fonction **ET** ne vaut 1 que si toutes les variables valent 1.

Toute fonction peut être exprimée au moyen des 3 fonctions de base (et / ou / non)

e1	e2	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

$F1 = 0$

$F2 = \overline{e1}.\overline{e2}$

$F3 = \overline{e1}.e2$

$F4 = \overline{e1}$

$F5 = e1.\overline{e2}$

$F6 = \overline{e2}$

$F7 = \overline{e1}.e2 + e1.\overline{e2}$

$F8 = \overline{e1} + \overline{e2}$

$F9 = e1.e2$

$F10 = \overline{e1}.\overline{e2} + e1.e2$

$F11 = e2$

$F12 = \overline{e1} + e2$

$F13 = e1$

$F14 = e1 + \overline{e2}$

$F15 = e1 + e2$

$F16 = 1$

Postulats de l'algèbre de Boole

(1) $a + 1 = 1$

(2) $a.1 = a$

(3) $a + 0 = a$

(4) $a.0 = 0$

(5) $a + a = a$

(6) $a + \overline{a} = 1$

(7) $a.a = a$

(8) $a.\overline{a} = 0$

Associativité

(9) $a+(b+c)= (a+b)+c$

(10) $a.(b.c)=(a.b).c$

Commutativité

(11) $a+b=b+a$

(12) $a.b=b.a$

Distributivité

(13) $a.(b+c)=a.b+a.c$

(14) $a+(b.c)=(a+b).(a+c)$

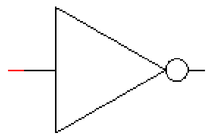
Théorème de DE MORGAN

(15) $\overline{a + b} = \overline{a} . \overline{b}$

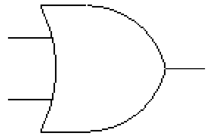
(16) $\overline{a . b} = \overline{a} + \overline{b}$

Portes logiques – symboles normalisés / Logigrammes

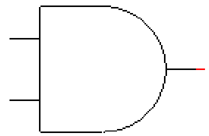
NON

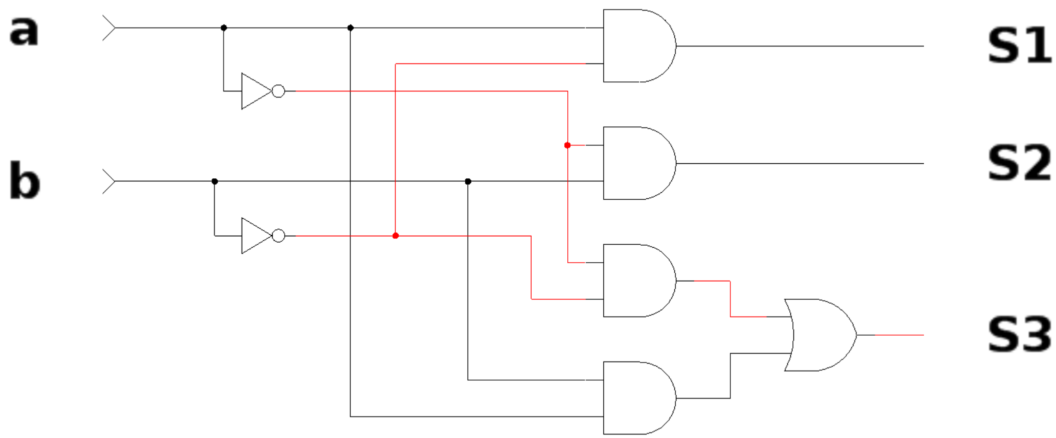


OU



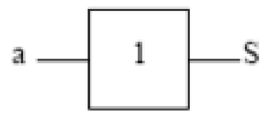
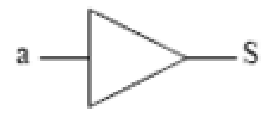
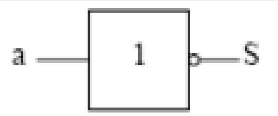
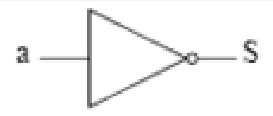
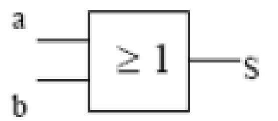
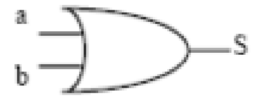
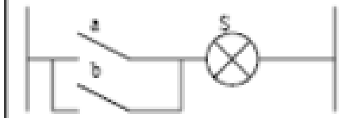
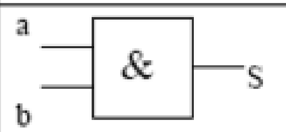
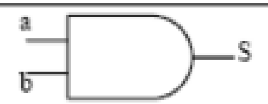
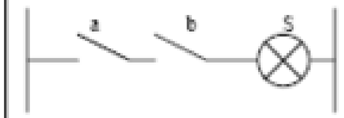
ET



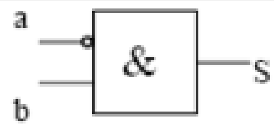
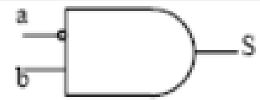
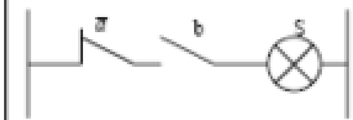
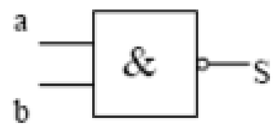
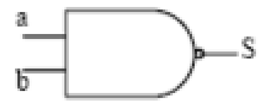
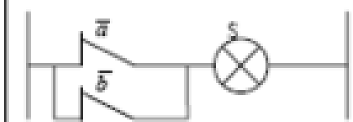
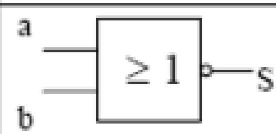
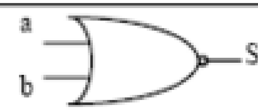
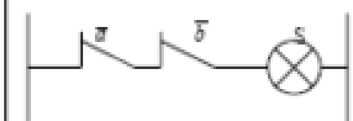
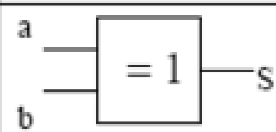
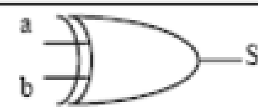
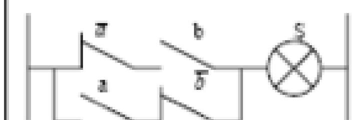
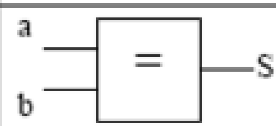
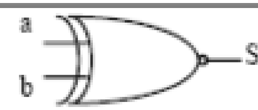
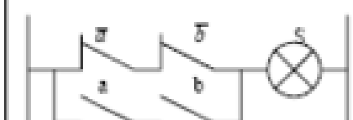


fonctions logiques.cdl

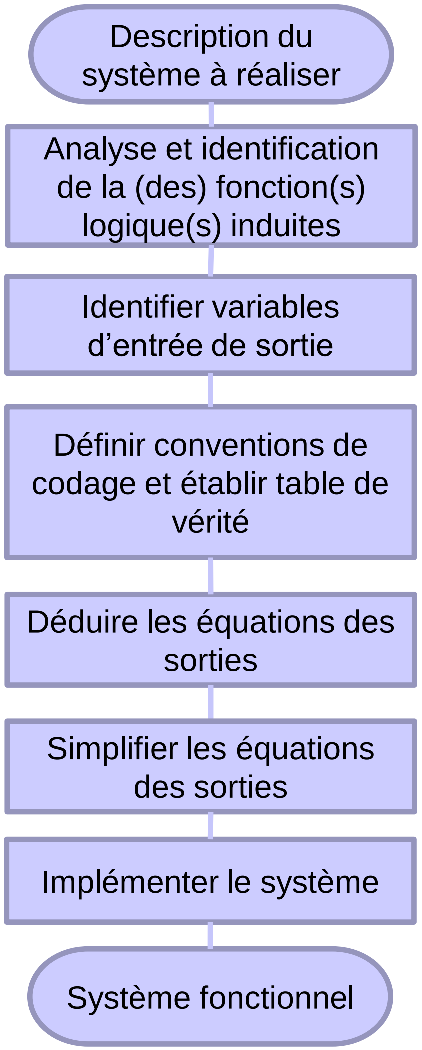
Portes logiques – symboles normalisés

Opérateur	équation logique	symbole AFNOR	symbole ASGS	table de vérité	schéma à contact
OUI	$S = a$			a S 0 0 1 1	
NON	$S = \overline{a}$			a S 0 1 1 0	
OU	$S = a + b$			a b S 0 0 0 0 1 1 1 0 1 1 1 1	
ET	$S = a.b$			a b S 0 0 0 0 1 0 1 0 0 1 1 1	

Portes logiques – symboles normalisés

Opérateur	équation logique	symbole AFNOR	symbole ASGS	table de vérité	schéma à contact
INHIBITION	$S = \overline{a}.b$			a b S 0 0 0 0 1 1 1 0 0 1 1 0	
NAND (NON ET)	$S = \overline{a.b} = \overline{a} + \overline{b}$			a b S 0 0 1 0 1 1 1 0 1 1 1 0	
NOR (NON OU)	$S = \overline{a + b} = \overline{a}.\overline{b}$			a b S 0 0 1 0 1 0 1 0 0 1 1 0	
OU EXCLUSIF	$S = a \oplus b$			a b S 0 0 0 0 1 1 1 0 1 1 1 0	
IDENTITE	$S = \overline{a \oplus b}$			a b S 0 0 1 0 1 0 1 0 0 1 1 1	

Synthèse des fonctions logiques combinatoires



Synthèse des fonctions logiques combinatoires

- Description du système à réaliser
- Analyse et identification de la (des) fonction(s) logique(s) induites
- Identifier variables d'entrée de sortie
- Définir conventions de codage et établir table de vérité
- Déduire les équations des sorties
- Simplifier les équations des sorties
- Implémenter le système
- Système fonctionnel

Conception d'un système de protection d'un motocycliste contre le risque de roulage avec béquille latérale non repliée …

objectif: couper l'alimentation du circuit d'allumage en cas de situation dangereuse

Synthèse des fonctions logiques combinatoires

Description du système à réaliser

Analyse et identification de la (des) fonction(s) logique(s) induites

Identifier variables d'entrée de sortie

Définir conventions de codage et établir table de vérité

Déduire les équations des sorties

Simplifier les équations des sorties

Implémenter le système

Système fonctionnel

Le système à réaliser doit agir sur…

- l' alimentation du circuit d'allumage

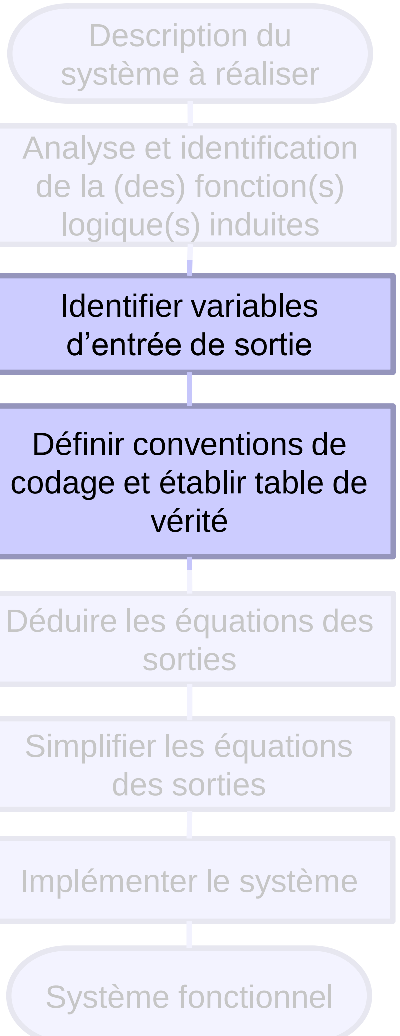
en fonction de…

-La position de la béquille,

-L'état de la boite de vitesses,

D'autre part, l'alimentation du circuit d'allumage nécessite la position '2' de la clef de contact

Synthèse des fonctions logiques combinatoires

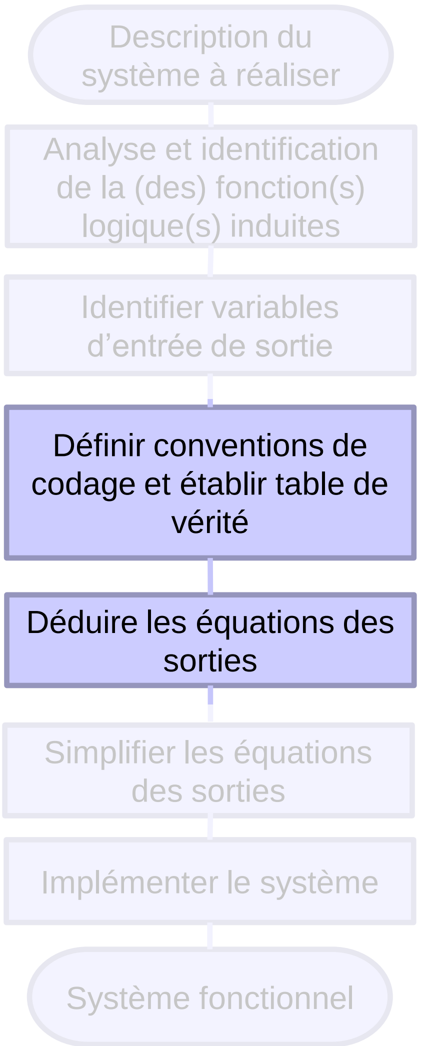


CCP2 = Clef de Contact en Position 2
BLD = Béquille Latérale Dépliée
PME = Point Mort Enclenché
CAA = Circuit d'Allumage Alimenté

1 ⇔ VRAI / OUI
0 ⇔ FAUX / NON

CCP2	BLD	PME	CAA

Synthèse des fonctions logiques combinatoires



CCP2 = Clef de Contact en Position 2
BLR = Béquille Latérale Dépliée
PME = Point Mort Enclenché
CAA = Alimenter le Circuit d'Allumage

1 ⇔ VRAI / OUI
0 ⇔ FAUX / NON

CCP2	BLD	PME	CAA
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

CAA =

Synthèse des fonctions logiques combinatoires

Description du système à réaliser

Analyse et identification de la (des) fonction(s) logique(s) induites

Identifier variables d'entrée de sortie

Définir conventions de codage et établir table de vérité

Déduire les équations des sorties

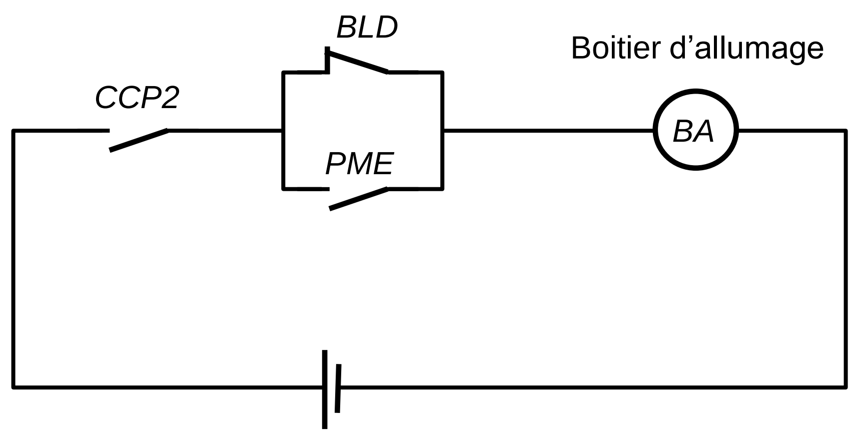
Simplifier les équations des sorties

Implémenter le système

Système fonctionnel

$$CAA = (CCP2.\overline{BLD}.\overline{PME}) + (CCP2.\overline{BLD}.PME) + (CCP2.BLD.PME)$$

Application de l'Algèbre de Boole à l'étude des circuits à contacts



$$BA = CCP2.\left(\overline{BLD}+PME\right)$$

Synthèse des fonctions logiques combinatoires

Description du système à réaliser

Analyse et identification de la (des) fonction(s) logique(s) induites

Identifier variables d'entrée de sortie

Définir conventions de codage et établir table de vérité

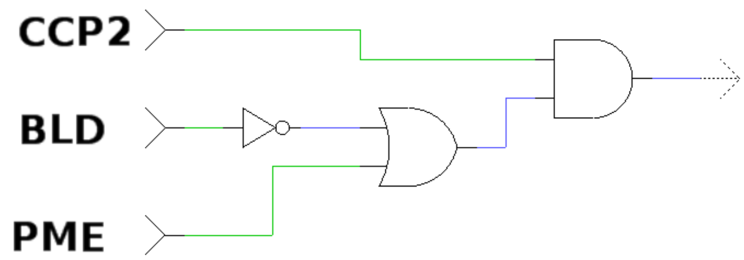
Déduire les équations des sorties

Simplifier les équations des sorties

Implémenter le système

Système fonctionnel

$$CAA = CCP2.\left(\overline{BLD}+PME\right)$$

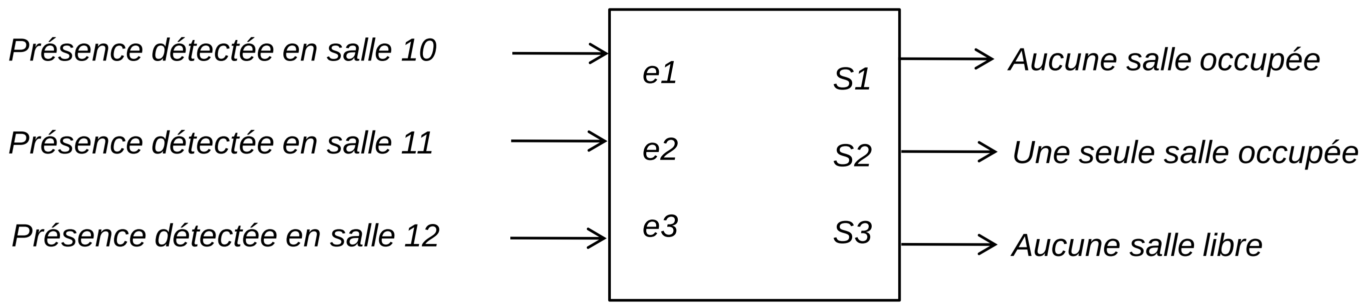


Signal d'activation du Boitier d'allumage

Cas des fonctions logiques multiples

Fonction logique multiple:

plusieurs fonctions logiques qui dépendent des mêmes variables d'entrée …



Chaque sortie est traitée comme une fonction logique simple

e1	e2	e3	S1
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

e1	e2	e3	S2
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

e1	e2	e3	S3
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

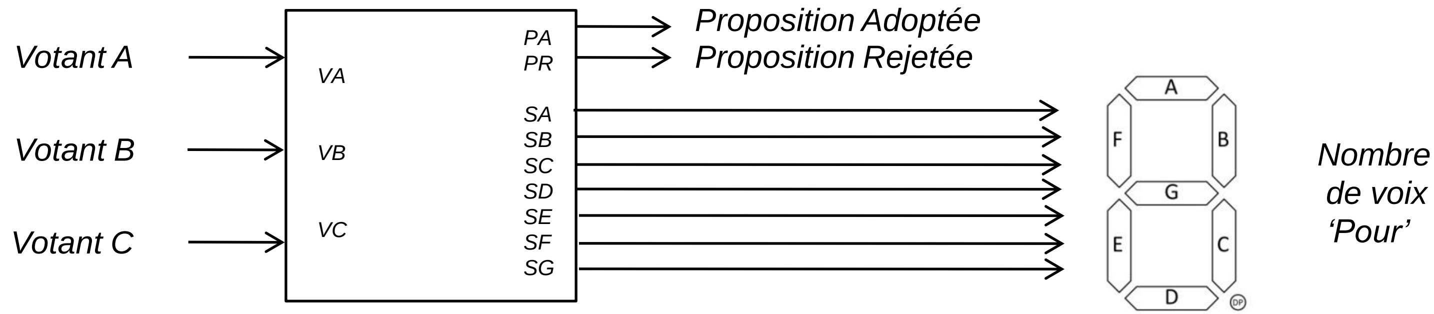
entrées			sorties		
e1	e2	e3	S1	S2	S3
0	0	0	1	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	0
1	0	0	0	1	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	1

$S1 = \overline{e1}.\overline{e2}.\overline{e3}$

$S2 = \overline{e1}.\overline{e2}.e3 + \overline{e1}.e2.\overline{e3} + e1.\overline{e2}.\overline{e3}$

$S3 = e1.e2.e3$

Cas des fonctions logiques multiples



entrées			sorties								
VA	VB	VC	PA	PR	SA	SB	SC	SD	SE	SF	SG
0	0	0									
0	0	1									
0	1	0									
0	1	1									
1	0	0									
1	0	1									
1	1	0									
1	1	1									

Les deux formes canoniques de l’expression d’une fonction

$$\boxed{F = \overline{a}.\overline{b}.c + \overline{a}.b.\overline{c} + \overline{a}.b.c + a.\overline{b}.\overline{c} + a.\overline{b}.c + a.b.c}$$

a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Les deux formes canoniques de l'expression d'une fonction

$$\boxed{F = \overline{a}.\overline{b}.c + \overline{a}.b.\overline{c} + \overline{a}.b.c + a.\overline{b}.\overline{c} + a.\overline{b}.c + a.b.c}$$

a	b	c	F	$\overline{F}$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

$$\overline{F} = \overline{a}.\overline{b}.\overline{c} + a.b.\overline{c}$$

Les deux formes canoniques de l'expression d'une fonction

$$F = \overline{a}.\overline{b}.c + \overline{a}.b.\overline{c} + \overline{a}.b.c + a.\overline{b}.\overline{c} + a.\overline{b}.c + a.b.c$$

a	b	c	F	$\overline{F}$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

$$\overline{F} = \overline{a}.\overline{b}.\overline{c} + a.b.\overline{c}$$

$$\overline{\overline{F}} = F = \overline{\overline{a}.\overline{b}.\overline{c} + a.b.\overline{c}}$$

Les deux formes canoniques de l'expression d'une fonction

$$\boxed{F = \overline{a}.\overline{b}.c + \overline{a}.b.\overline{c} + \overline{a}.b.c + a.\overline{b}.\overline{c} + a.\overline{b}.c + a.b.c}$$

a	b	c	F	$\overline{F}$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

$$\overline{F} = \overline{a}.\overline{b}.\overline{c} + a.b.\overline{c}$$

$$\overline{\overline{F}} = F = \overline{\overline{a}.\overline{b}.\overline{c} + a.b.\overline{c}}$$

$$\boxed{F = (a + b + c).(\overline{a} + \overline{b} + c)}$$

Simplification des fonctions logiques par la méthode de **Karnaugh**

Méthode graphique qui exploite la notion d'adjacence des monômes…

Monôme: expression algébrique qui ne contient que des ~~produits~~ « ET »

La première forme canonique est une « somme » de monômes

$$F = \overline{a}.\overline{b}.c + \overline{a}.b.\overline{c} + \overline{a}.b.c + a.\overline{b}.\overline{c} + a.\overline{b}.c + a.b.c$$

Monômes adjacents: deux monômes sont dits adjacents si:
-Ils sont composés strictement des mêmes variables
-Et si une seule variable change d'état entre les deux monômes

Exemple: $\overline{a}.\overline{b}.c$ et $\overline{a}.b.c$ sont deux monômes adjacents

Remarque: le « OU » entre deux monômes adjacents donne un monôme constitué avec une variable en moins ….

$$\bar{a}.\bar{b}.c + \bar{a}.b.c = \bar{a}.c.(\bar{b} + b) = \bar{a}.c.1 = \bar{a}.c$$

$$F = \bar{a}.\bar{b}.c + \bar{a}.b.\bar{c} + \bar{a}.b.c + a.\bar{b}.\bar{c} + a.\bar{b}.c + a.b.c$$

$$F = \bar{a}.b.\bar{c} + \bar{a}.c + a.\bar{b}.\bar{c} + a.c$$

$$F = \bar{a}.b.\bar{c} + a.\bar{b}.\bar{c} + c$$

Construction des tables de **Karnaugh**

Les tables de KARNAUGH sont des tables de vérité présentées sous la forme de matrices de 2^n cases, n étant le nombre de combinaisons des variables d'entrées.
Chaque case de cette matrice permet de représenter un point de la fonction.
Les cases sont disposées de façon à ce que deux cases adjacentes soient toujours associées à deux combinaisons adjacentes des variables d'entrées.

a	b	F(a,b)
0	0	P1
0	1	P2
1	0	P3
1	1	P4

Exemple de disposition des cases pour représenter les 4 points d'une fonction de deux variables

a \ b	0	1
0	P1	P2
1	P3	P4

F(a,b)

Exemples de tables de fonctions de deux variables

a	b	a.b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	1

a	b	F
0	0	0
0	1	0
1	0	1
1	1	1

a	b	F
0	0	1
0	1	0
1	0	1
1	1	1

a.b

a \ b	0	1
0	0	0
1	0	1

a+b

a \ b	0	1
0	0	1
1	1	1

F=

a \ b	0	1
0	0	0
1	1	1

F=

a \ b	0	1
0	1	0
1	1	1

Tables de KARNAUGH des fonctions de 3 variables

a	b	c	F(a,b,c)
0	0	0	P1
0	0	1	P2
0	1	0	P3
0	1	1	P4
1	0	0	P5
1	0	1	P6
1	1	0	P7
1	1	1	P8

c

b	0	1
0	P1	P2
1	P3	P4

F(a,b,c) quand a=0

c

b	0	1
0	P5	P6
1	P7	P8

F(a,b,c) quand a=1

c

	b	0	1
a=0	0	P1	P2
	1	P3	P4
a=1	1	P7	P8
	0	P5	P6

F(a,b,c)

c

a,b	0	1
00		
01		
11		
10		

F(a,b,c)

Tables de KARNAUGH des fonctions de 3 variables

a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

a,b \ c	0	1
00	0	1
01	1	1
11	0	1
10	1	1

F(a,b,c)

a \ b,c	00	01	11	10
0	0	1	1	1
1	1	1	1	0

F(a,b,c)

Tables de KARNAUGH des fonctions de 4 variables

a	b	c	d	F(a,b,c,d)
0	0	0	0	P1
0	0	0	1	P2
0	0	1	0	P3
0	0	1	1	P4
0	1	0	0	P5
0	1	0	1	P6
0	1	1	0	P7
0	1	1	1	P8
1	0	0	0	P9
1	0	0	1	P10
1	0	1	0	P11
1	0	1	1	P12
1	1	0	0	P13
1	1	0	1	P14
1	1	1	0	P15
1	1	1	1	P16

a,b \ c,d	00	01	11	10
00	P1	P2	P4	P3
01	P5	P6	P8	P7
11	P13	P14	P16	P15
10	P9	P10	P12	P11

F(a,b,c,d)

Règles d'exploitation des tables de KARNAUGH

Il faut regrouper tous les 1 par groupes de 2, 4, 8 ou 16 (!) les plus gros et les moins nombreux possibles.

Chaque case avec 1 peut appartenir à plusieurs regroupements différents

Il est généralement astucieux de commencer par les 1 les plus isolés.

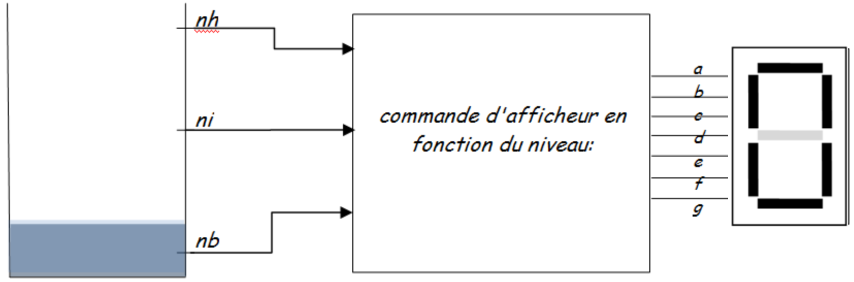
Il peut parfois être plus rapide de calculer $\overline{F}$ en regroupant les 0 puis de revenir à F en complémentant le résultat.

Fonctions logiques incomplètement spécifiées

Une fonction logique est dite incomplètement spécifiée lorsque sa valeur ne peut pas être définie pour une combinaison au moins de ses variables d'entrée.

Cette situation se produit souvent à cause de relations de dépendance entre les variables d'entrées.

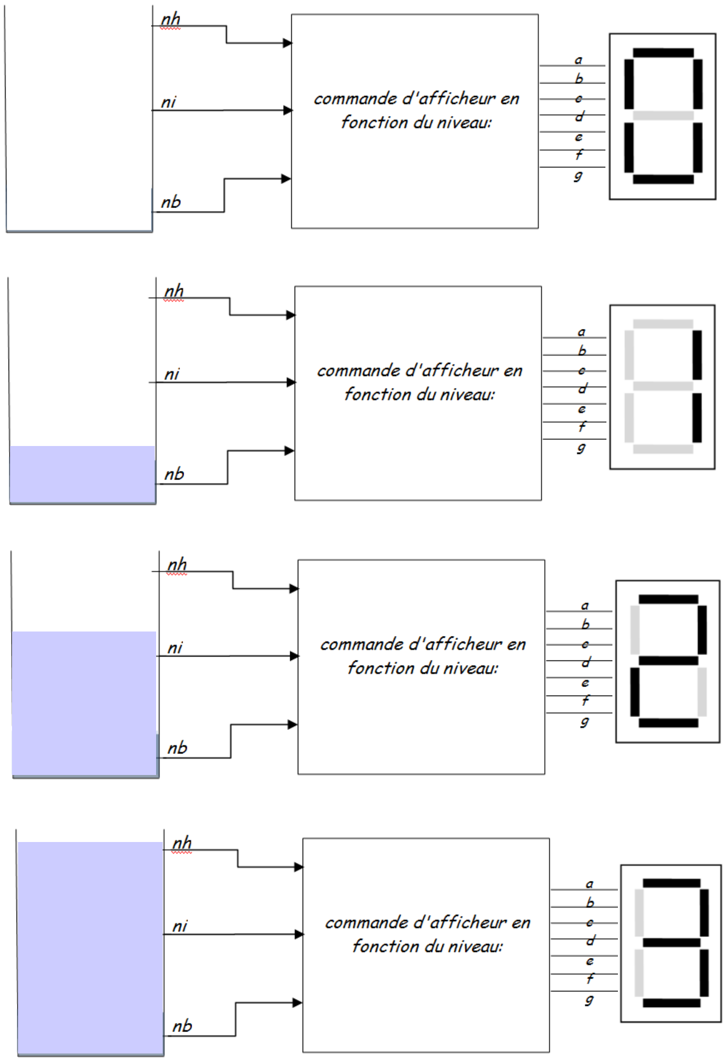
Exemple:

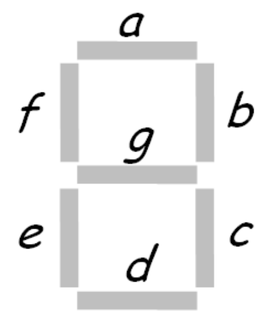


nh	ni	nb
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Combinaisons « impossibles »

Fonctions logiques incomplètement spécifiées

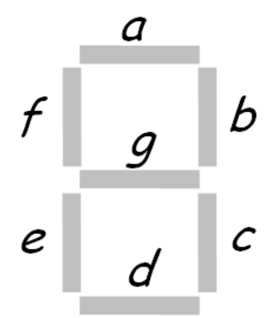




entrées			sorties						
nh	ni	nb	a	b	c	d	e	f	g
0	0	0	1	1	1	1	1	1	0
0	0	1	0	1	1	0	0	0	0
0	1	0							
0	1	1	1	1	0	1	1	0	1
1	0	0							
1	0	1							
1	1	0							
1	1	1	1	1	1	1	0	0	1

Fonctions logiques incomplètement spécifiées

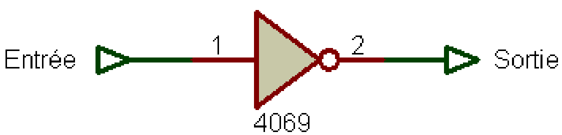
entrées				*sorties*					
nh	ni	nb	a	b	c	d	e	f	g
0	0	0	1	1	1	1	1	1	0
0	0	1	0	1	1	0	0	0	0
0	1	0							
0	1	1	1	1	0	1	1	0	1
1	0	0							
1	0	1							
1	1	0							
1	1	1	1	1	1	1	0	0	1

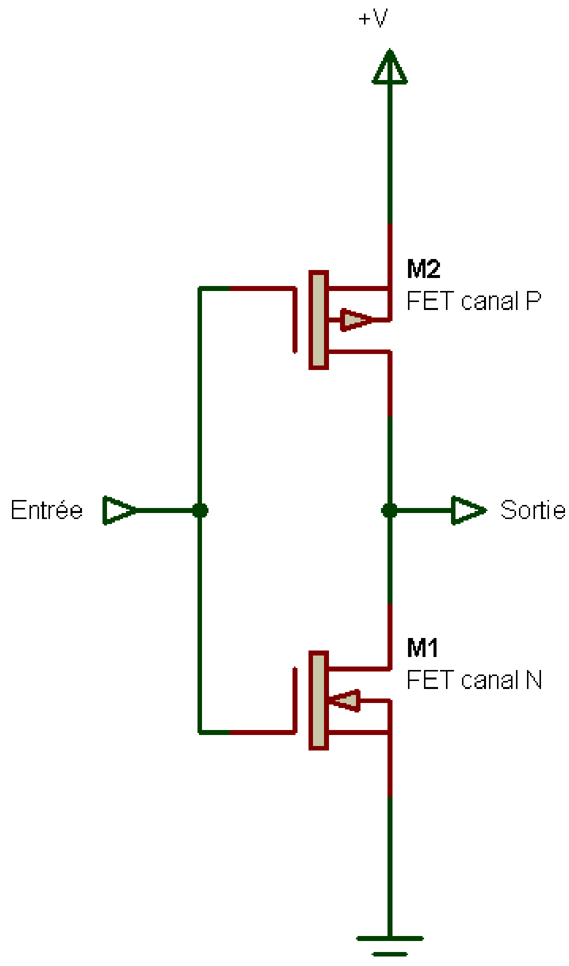


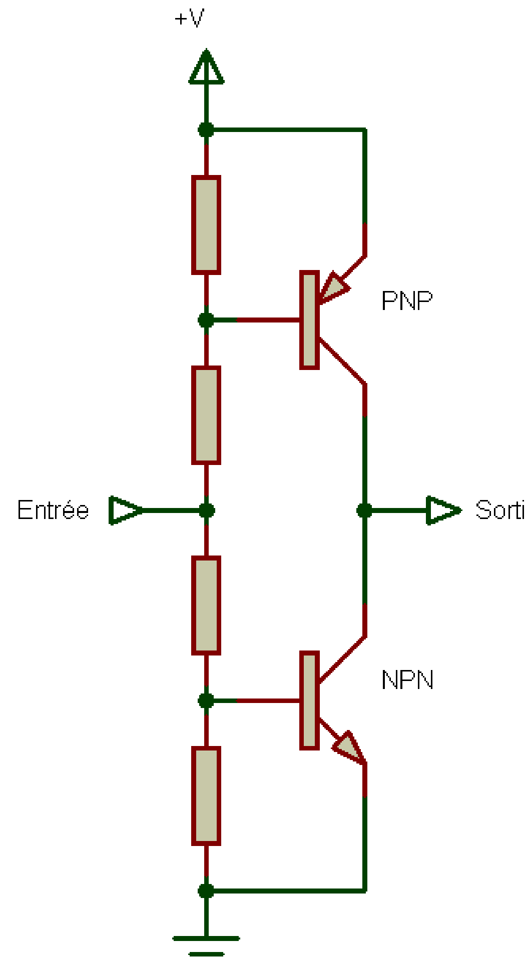
		ni, nb			
		00	01	11	10
nh	0	1	0	1	
	1			1	

a

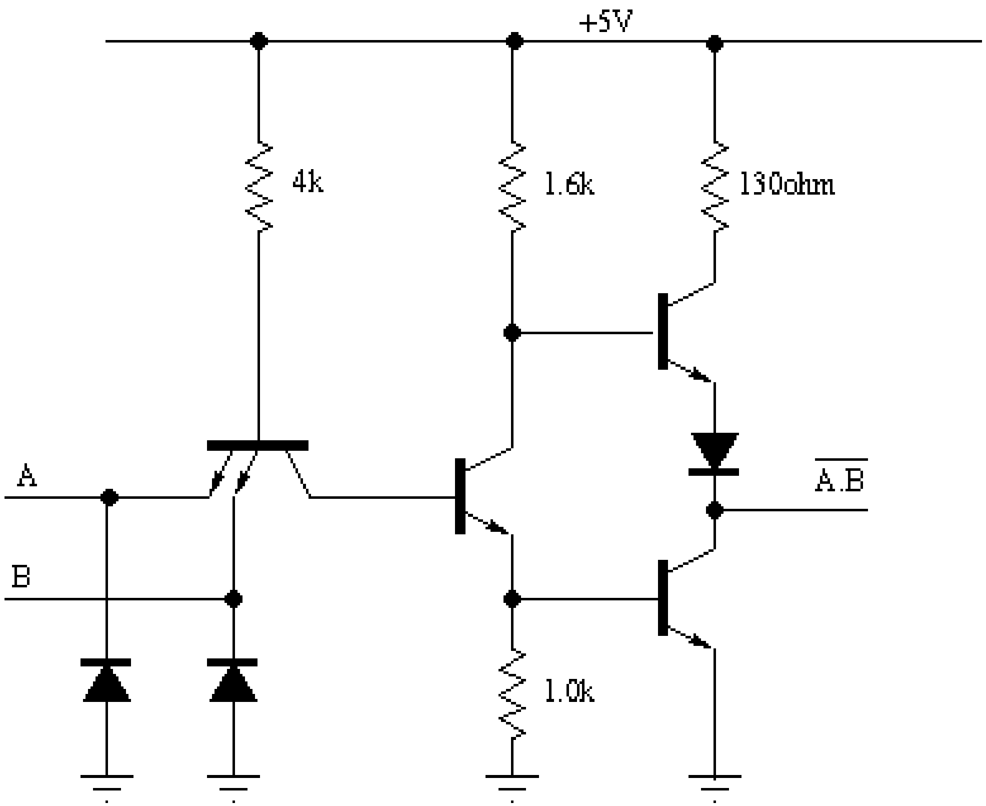
Porte électroniques en technologie électronique’

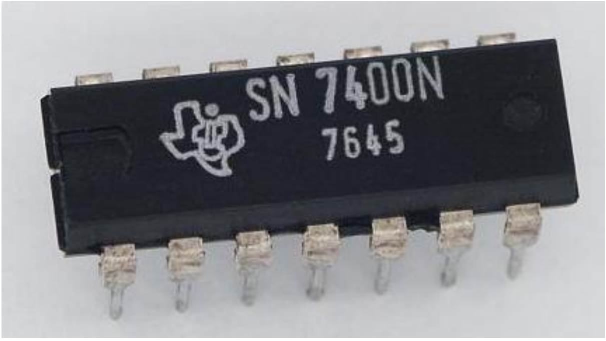


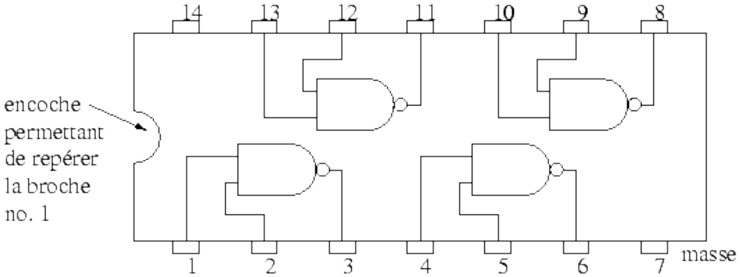




Porte électroniques en technologie électronique'





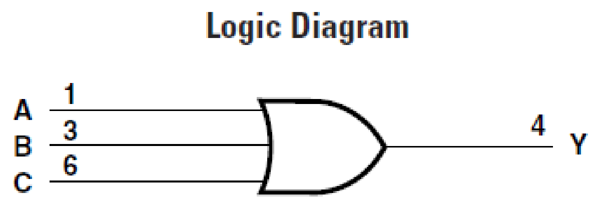


Porte électroniques en technologie électronique'

1G332

SINGLE 3-INPUT POSITIVE-OR GATE

- Y = A + B + C



FUNCTION TABLE

INPUTS			OUTPUT
A	B	C	Y
H	X	X	H
X	H	X	H
X	X	H	H
L	L	L	L

ELECTRICAL CHARACTERISTICS AND RECOMMENDED OPERATING CONDITIONS

PARAMETER	MAX or MIN	LVC 5V	LVC 3.3V	LVC 2.5V	LVC 1.8V	UNIT
I_{CC}	MAX	0.01	0.01	0.01	0.01	mA
I_{OH}	MAX	-32	-24	-8	-4	mA
I_{OL}	MAX	32	24	8	4	mA

SWITCHING CHARACTERISTICS

PARAMETER	INPUT	OUTPUT	MAX or MIN	LVC 5V	LVC 3.3V	LVC 2.5V	LVC 1.8V
t_{PLH}	A, B or C	Y	MAX	3.5	4.8	6.2	17.2
t_{PHL}				3.5	4.8	6.2	17.2

UNIT:ns

Porte électroniques en technologie électronique