

M1214

AUTOMATISMES – SEMESTRE 1

Organisation M1214

Automatismes Semestre 1

6 CMs x 1h30

1 TP x 1h30

Évaluation

- La note du DS en période DS2 (janvier)
- La note du TP

Systeme automatisé ?

Comme tout objet technique (OT), un système automatisé répond à un besoin - il découle de la même démarche de projet :

- identification précise du besoin

- C'est ce qui garantit une valeur ajoutée à la matière d'oeuvre entrante pour donner la matière d'oeuvre sortante.

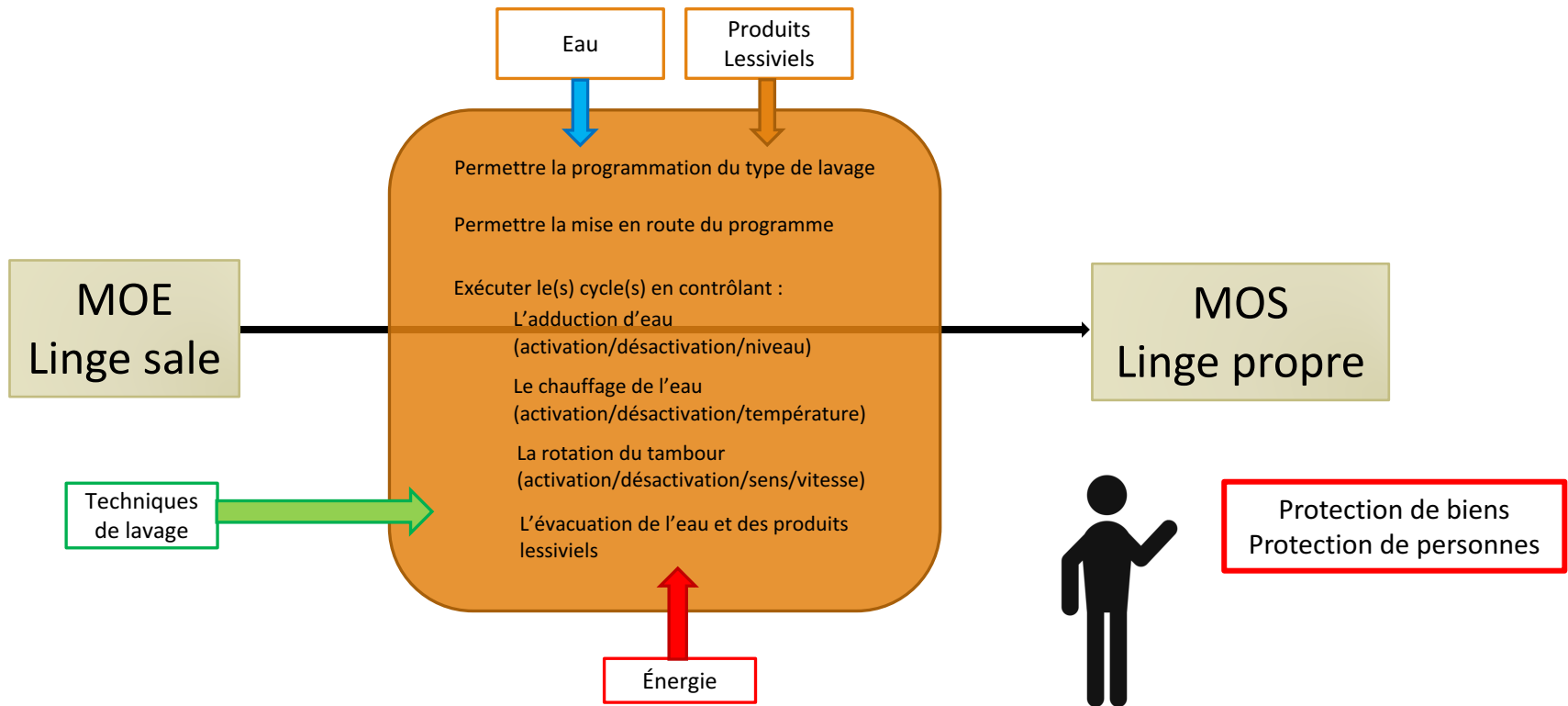
- identification des contraintes

- Relations entre l'OT dans son environnement ;
- Respect des normes.

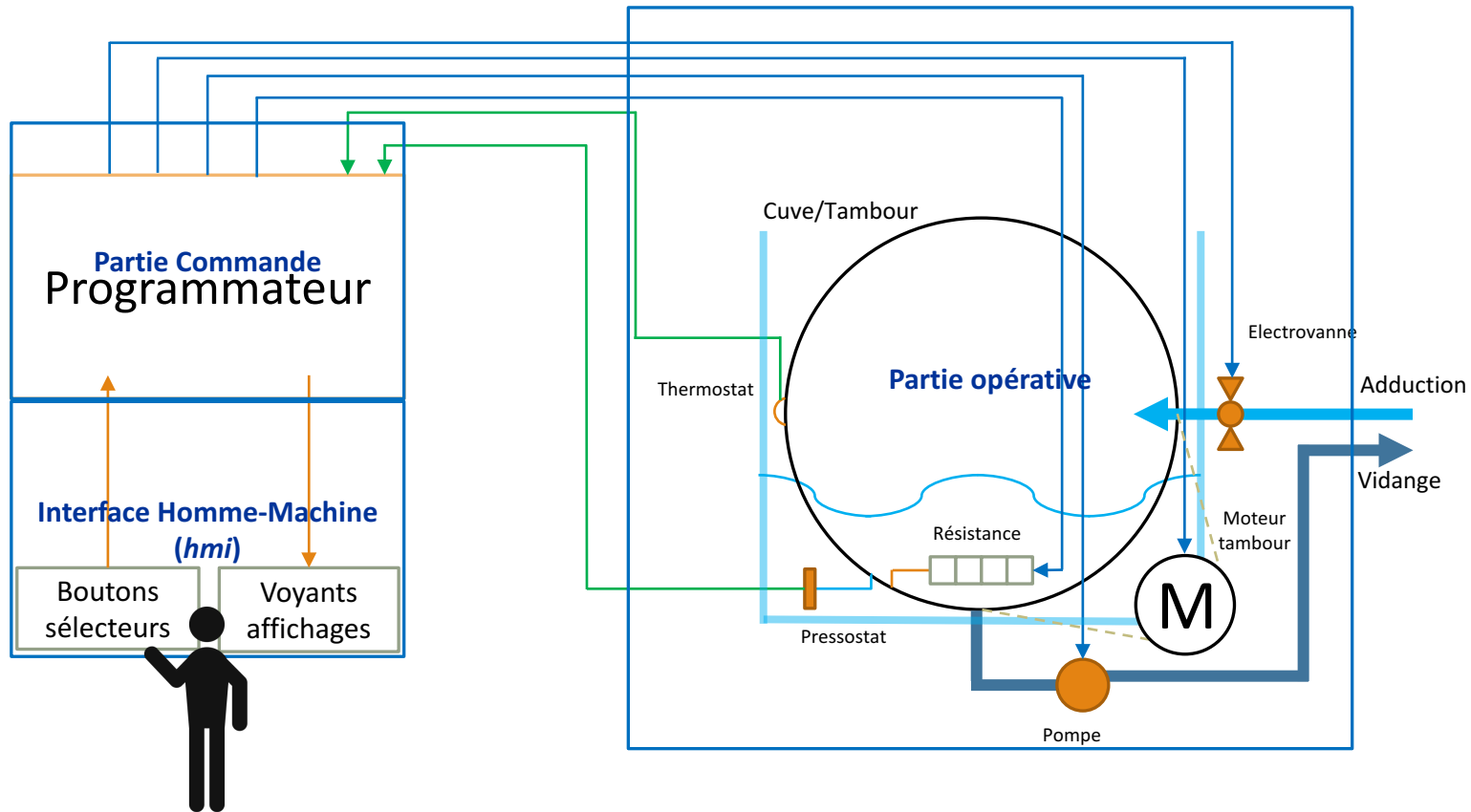
- effectuer une analyse fonctionnelle (fonctions à réaliser)

- rédiger un cahier des charges fonctionnel (NF X50-151)

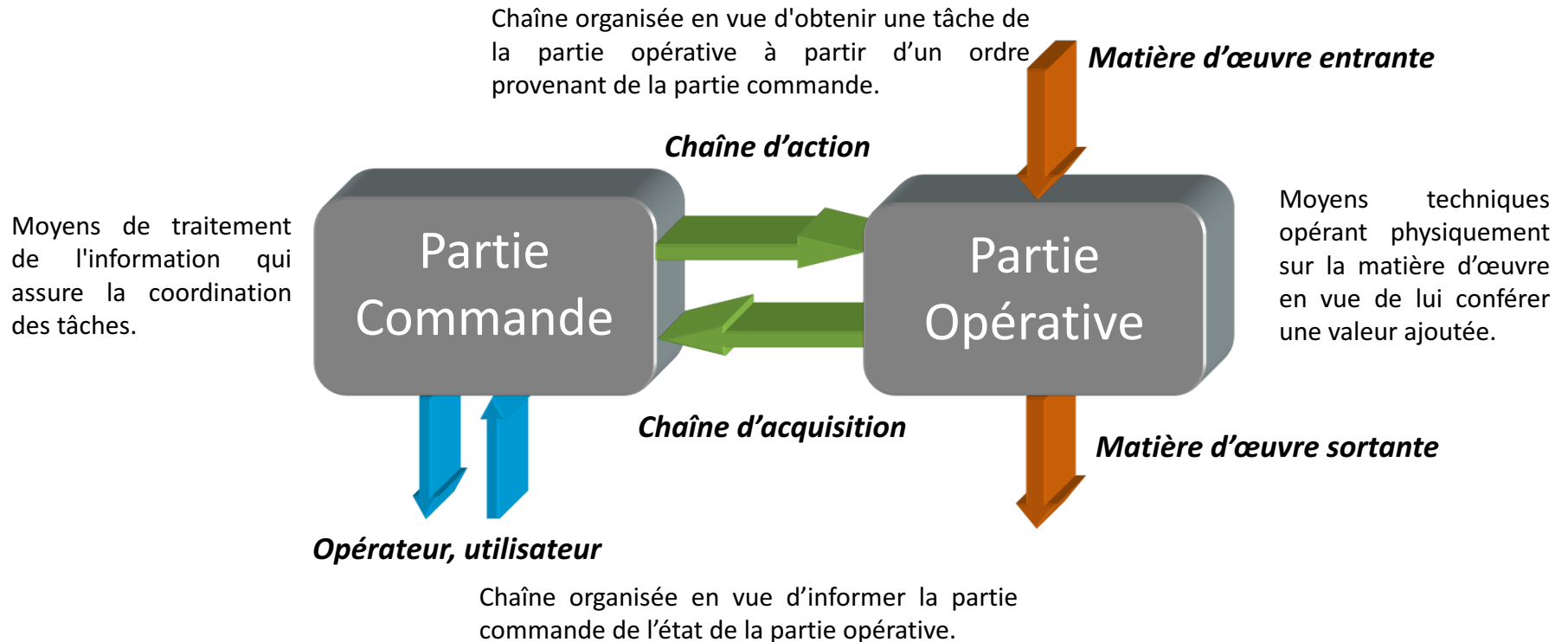
Un S.A. : Le lave-linge 1/2



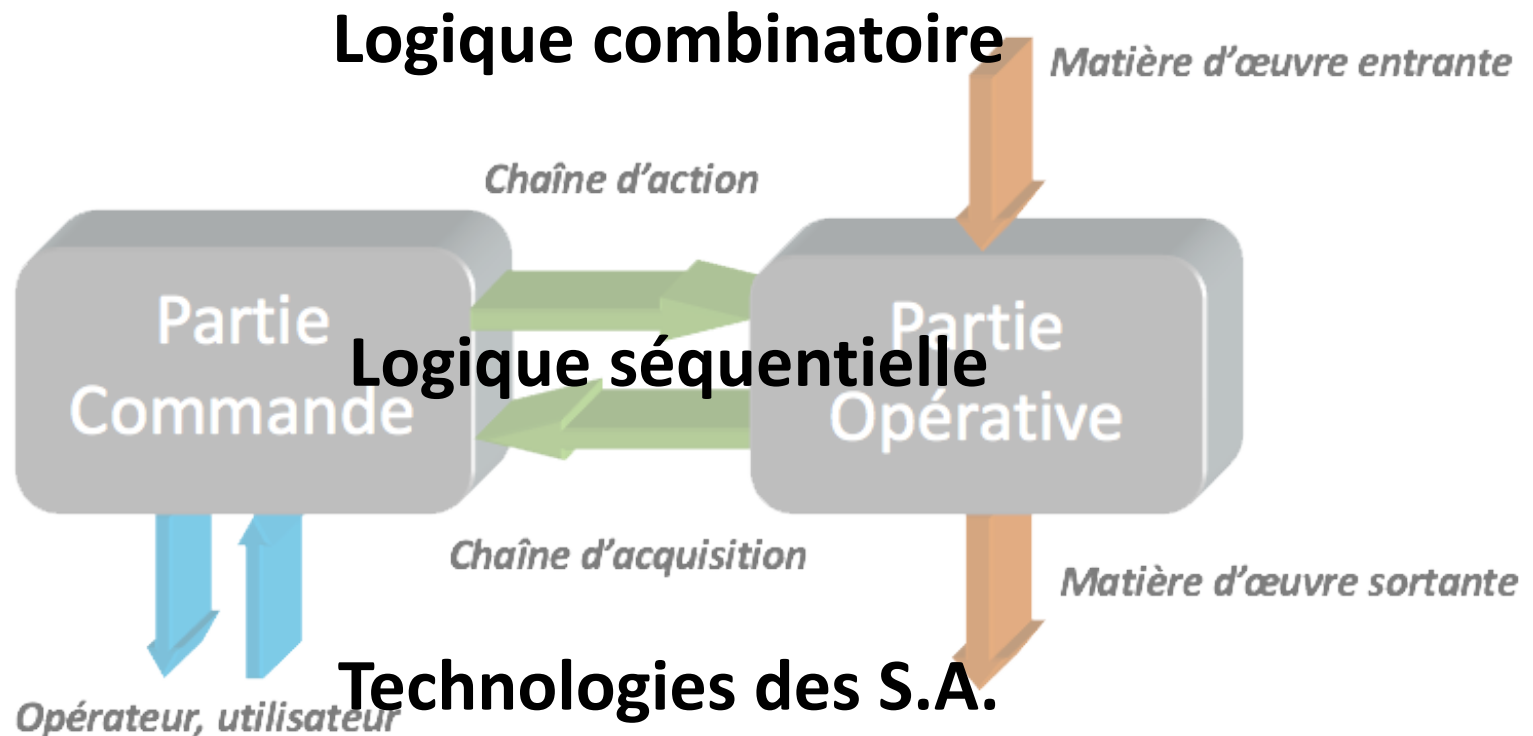
Un S.A. : Le lave-linge 2/2



Structure générale d'un système automatisé



Analyser, comprendre, concevoir



Logique Combinatoire

GMP3 – M1214

Plan du chapitre

- Circuits combinatoires
- Systèmes de codage – Conversion
- Algèbre de BOOLE
- Les tableaux de KARNAUGH
- Théorème de DE MORGAN
- Simplification par la fonction XOR
- Réalisation à partie de circuits logiques

Bibliographie

Cours et problèmes d'électronique Numérique

Jean Claude LAFONT & Jean Paul VABRE

Édition 1996 - ELLIPSES

Automatique et informatique industrielle

J.PERRIN & F BINET

Édition 2004 NATHAN technologie

Ce qu'il faut savoir sur les automatismes

Édition 2008 Technosup, <http://www.amazon.fr/>

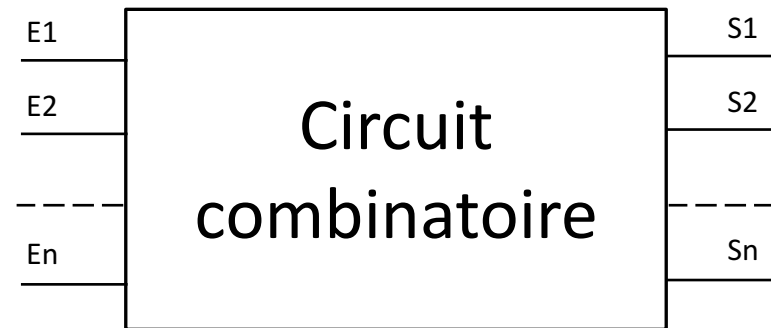
Philippe GRARE & Imed KACEM

Logique combinatoire

Un circuit combinatoire détermine une relation unique entre des sorties (S) et des entrées (E). Il en découle que, pour **une valeur donnée de E1, E2, ..,En**, il n'y a **qu'une valeur possible et invariable de S1, S2, ..Sn**.

Ces entrées et sorties peuvent prendre 2 valeurs : 1 **OU** 0.

On parle de variable booléennes.



$$S1, \dots, S_n = f(E1, \dots, E_n)$$

Variables booléennes 1/2



Une variable booléenne ne prend que 2 valeurs possibles et distinctes : **0** ou **1**.

On peut également les représenter par les mnémoniques **true** et **false**.

Ce ne sont pas des nombres entiers !!

Ces 2 états (0 , 1) sont souvent rencontrés dans les S.A. :

- un contact électrique fermé ou ouvert ;
- Un transistor passant ou bloqué;
- Une bobine de relais alimentée ou non ;
- une pression d'air élevée ou faible ;
- Etc..

Variables booléennes 2/2



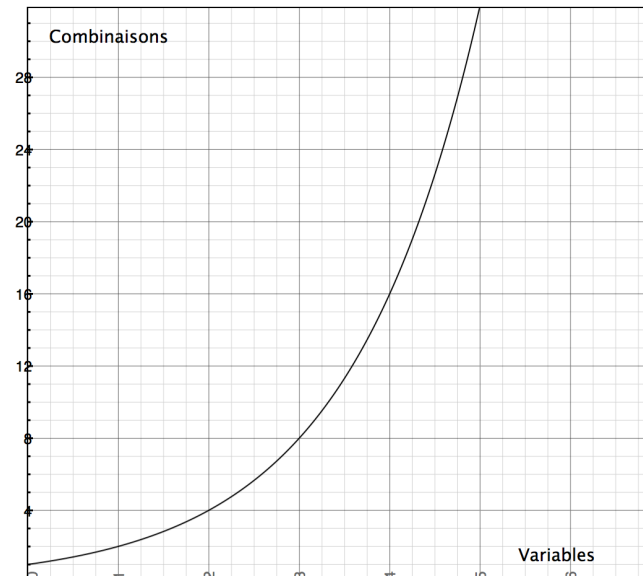
Le nombre de combinaisons (c) dépend du nombre de variables (n) :

$$c = 2^n$$

Pour 2 variables a et b , on peut réaliser le tableau suivant :

a	b	
0	0	1
0	1	2
1	0	3
1	1	4

$$c = 2^2 = 4$$



La croissance de c est exponentielle !

Systeme de codage – binaire pur

Le « binaire pur » est le langage de base des ordinateurs numériques. C'est aussi l'espace mathématique de l'algèbre de Boole {0, 1} ou base 2. Dans cet espace, tout nombre sera donc représenté par une suite de 0 et de 1.

Quelques exemples d'entiers en base 10 et leur représentation en base 2 ou binaire pur :

b_{10}	b_2
0	0
1	1
2	10
3	11
4	100

En généralisant :

$$d \text{ en décimal} = a_0.2^0 + a_1.2^1 + a_2.2^2 + \dots + a_n.2^n$$

Une information binaire est appelée en anglais **binary digit**, le mot-valise créé (et universellement connu) est **bit**.

Conversion Décimal - Binaire

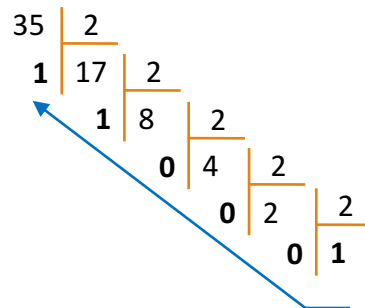
Pour convertir un décimal d (B_{10}) en binaire pur b (B_2)

On effectue les divisions successives de n par 2 avec :

- n prenant comme valeur initiale d
- puis n prenant les valeurs des quotients successifs jusqu'à $q = 1$

On reconstitue le binaire en prenant la dernière valeur de q (1 ou 0) et les restes successifs en remontant.

Exemple : 35 en décimal à convertir en binaire pur :



$$35_{10} = 100011_2$$

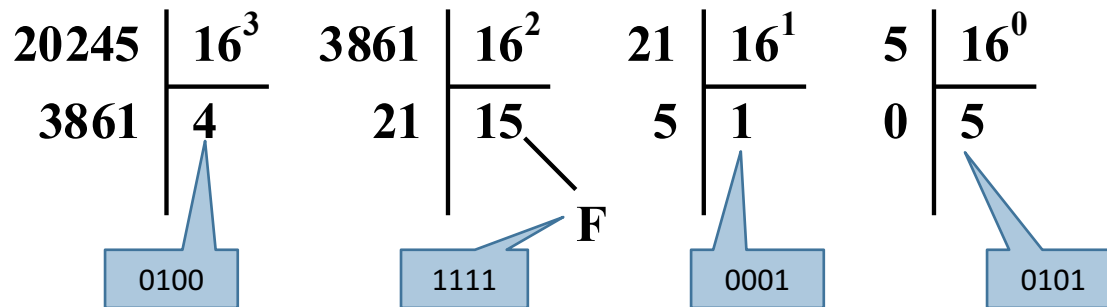
Codage h xad cimal

Utilis  par les microprocesseurs, mais aussi en informatique en g n ral, ce codage comporte 16 signes :

- Les signes 0   9 comme en d cimal
- Auxquels on a ajout  les signes A   F

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

Chaque « chiffres » (0   F) est repr sent  par 4 bits, de 0 (0000)   F (1111).



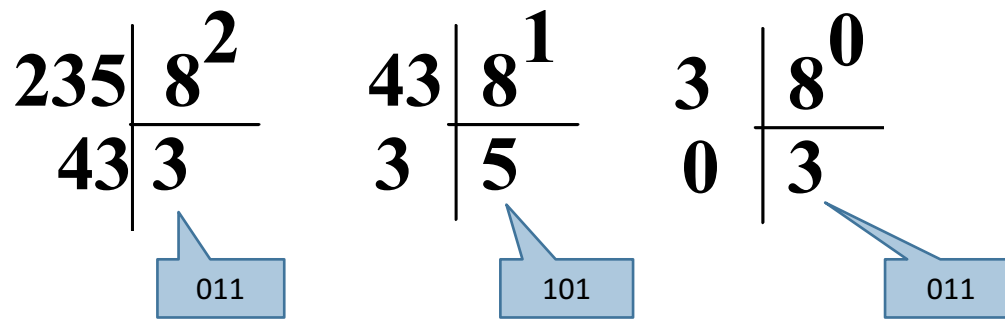
$$20245_{10} = 4F15_{16} = 0100\ 1111\ 0001\ 0101_2$$

Codage octal

Codage en base 8, utilisé aussi pour la programmation de micro-processeurs.

0	1	2	3	4	5	6	7	8	0	1	1	1	1	1	1	1
									0	1	2	3	4	5	6	
0	1	2	3	4	5	6	7	1	1	1	1	1	1	1	1	2
								0	1	2	3	4	5	6	7	0

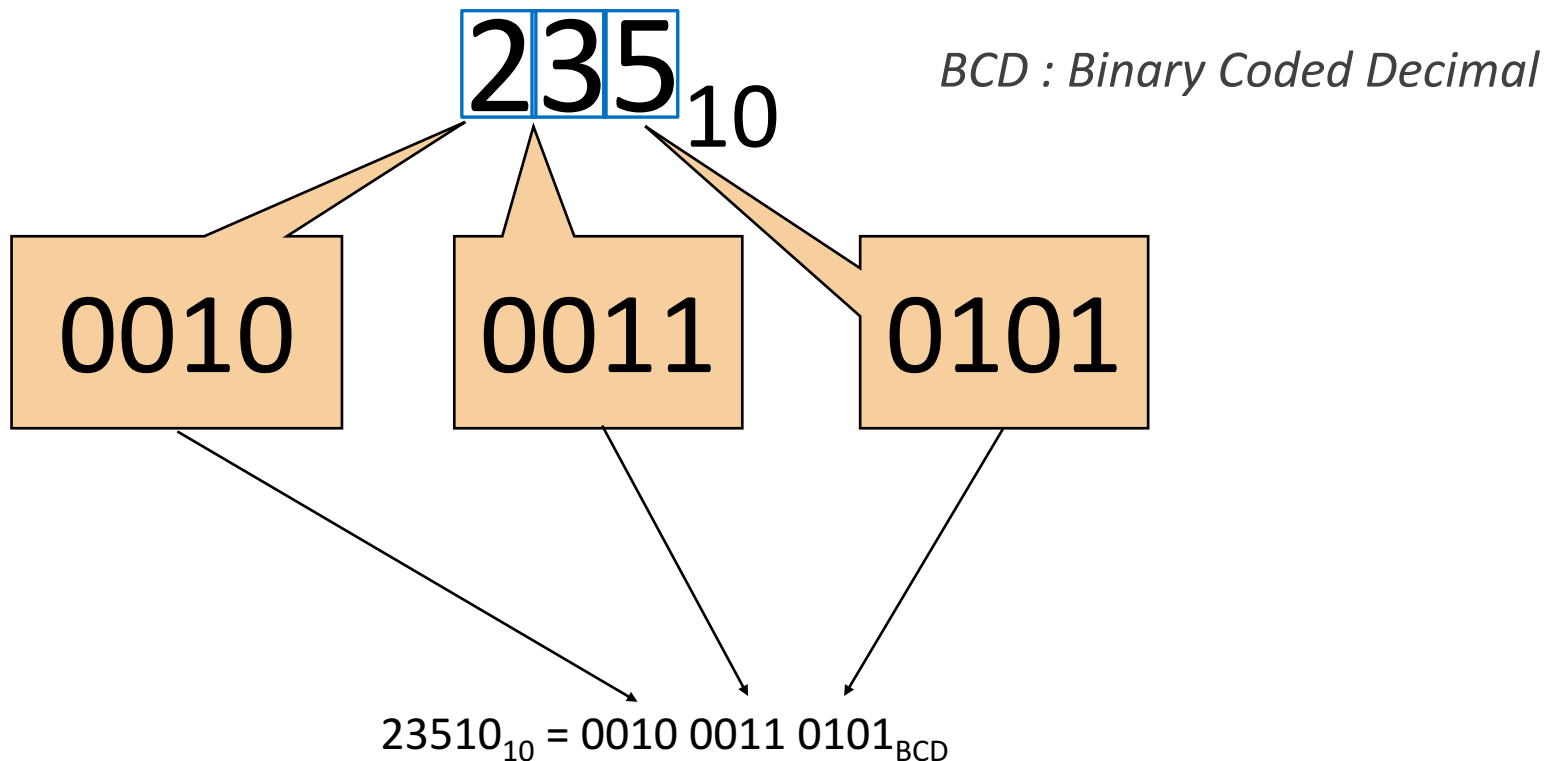
Chaque « chiffres » (0 à 7) est représenté par 3 bits, de 0 (000) à 7 (111).



$$235_{10} = 353_8 = 011\ 101\ 011_2$$

Décimal Codé Binaire (*BCD*)

C'est le résultat de la décomposition de chaque chiffre (., milliers, centaines, dizaines, unités) d'un nombre décimal en « mots » 4 bits. Les mots sont ensuite simplement concaténés pour donner le mot BCD.



Conversion base m - base n 1/3

C'est la mise en application de

$$d_m = a_0.n^0 + a_1.n^1 + a_2.n^2 + \dots + a_p.n^p$$

Les valeurs successives $a_p \dots a_0$ forment la représentation dans la base n.

On réalise des **divisions entières** successives dont :

- le numérateur de la première division est m
- le dénominateur de la première division est n^p immédiatement inférieur ou égal à m

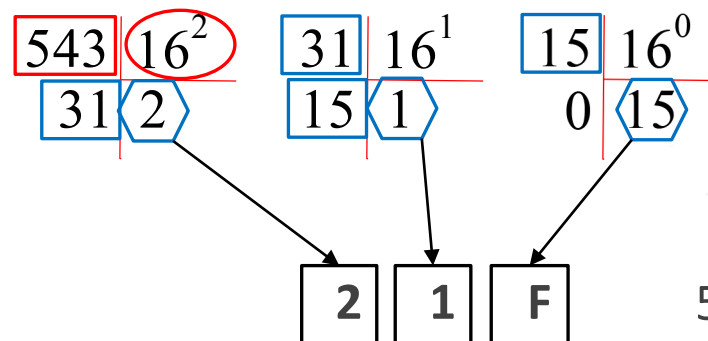
Pour les divisions suivantes :

- le numérateur devient le reste de la division précédente
- le dénominateur est n^{p-q} jusqu'à $p-q = 0$

Conversion base m - base n 2/3

Exemple : 543_{10} en hexadécimal

1. On cherche 16^n immédiatement inférieur à 543 : $16^2 \leq 543_{10} \leq 16^3$
2. On peut effectuer les divisions entières par 16^2 jusqu'à 16^0 :



Rem1 : $a^0 = 1$ si $a \neq 0$

Rem2 : $15_{10} = F_{16}$

$$543_{10} = 21F_{16}$$

Vérification :

$$\begin{aligned} 21F_{16} &= 2 \times 16^2 + 1 \times 16^1 + 15 \times 16^0 \\ &= 512 + 16 + 15 = 543_{10} \end{aligned}$$

Conversion base m - base n 3/3

Exemple : 543_{10} en octal

1. On cherche 8^n immédiatement inférieur à 543 : $8^3 \leq 543_{10} \leq 8^4$
2. On peut effectuer les divisions entières par 8^3 jusqu'à 8^0 :

$$\begin{array}{r|l} 543 & 8^3 \\ \hline 31 & 1 \end{array} \quad \begin{array}{r|l} 31 & 8^2 \\ \hline 31 & 0 \end{array} \quad \begin{array}{r|l} 31 & 8^1 \\ \hline 7 & 3 \end{array}$$

Rem : la dernière division sera toujours $r/1$ avec r , reste de l'avant-dernière division.

On peut donc directement prendre ce reste pour la conversion.

3. On prend les quotients successifs :

$$543_{10} = 1037_8$$

Vérification :

$$\begin{aligned} 1037_8 &= 1 \times 8^3 + 0 \times 8^2 + 3 \times 8^1 + 7 \times 8^0 \\ &= 512 + 0 + 24 + 7 = 543_{10} \end{aligned}$$

Binaire réfléchi ou code Gray

Le code binaire réfléchi est un code binaire qui, au passage entre deux nombres consécutifs, n'autorise le changement que d'un seul élément binaire (bit) à la fois.

Rappel des premières valeurs en binaire pur :

Au passage de 1 à 2, 2 valeurs de bit sont permutées.

Au passage de 3 à 4, ce sont cette fois 3 valeurs de bit sont permutées.

b_{10}	b_2
0	000
1	001
2	010
3	011
4	100

2 bits changent

3 bits changent

Une permutation à la fois = Code Gray

Les codages ne sont pas seulement inversés, mais ils sont aussi différents ($2 = 10_2 = 11_{\text{Gray}}$) puisque l'organisation croissante des décimaux ne changent pas. C'est pourquoi on parle de binaire **pur** d'une part, et de binaire **réfléchi** ou Gray d'autre part.

b_{10}	Gray
0	000
1	001
2	011
3	010
4	110

1 seul changement à chaque passage !

Nous verrons que certaines technologies de capteurs utilisent ce code.

Quelques règles d'écriture

Les variables désignant des entrées (appelées plus simplement « entrées ») sont le plus souvent représentées par des lettres en minuscules (a, b, c, x, y, etc..)

Les variables désignant des sorties (appelées plus simplement « sorties ») sont le plus souvent représentées par des lettres en majuscules (S, etc..)

On peut trouver aussi pour les entrées :

- E1, E2, E3.. (dénomination allemande)
- I1, I2, I3.. (dénomination internationale : I pour *Input*)

Et pour les sorties :

- A1, A2, A3.. (dénomination allemande)
- Q1, Q2, Q3.. (dénomination internationale : O pour *Output* devenu Q pour ne pas être confondu avec le zéro 0 !)

Enfin, pour toute variable booléenne **a**, il existe une variable booléenne \bar{a} (prononcé a barre) tel que si $a = 1$ alors $\bar{a} = 0$, et si $a = 0$ alors $\bar{a} = 1$.

On peut aussi écrire : $\bar{a} = \text{non } a$ ou encore $a = \overline{\bar{a}}$

La suite de ce cours va permettre de mettre en évidence un tel concept.

Fonctions booléennes 1/2

Fonctions booléennes pour deux variables a et b.

On a donc 4 (2^2) combinaisons possibles pour ab (00, 01, 10, 11) et 2^4 fonctions différentes suivant les valeurs des combinaisons :

F _n	ab 00	ab 01	ab 10	ab 11	Nom de la fonction	Équation équivalente
F0	0	0	0	0	Nulle	$S = 0$
F1	0	0	0	1	ET / AND	$S = a.b$
F2	0	0	1	0	Se réduit à	$S = a.\bar{b}$
F3	0	0	1	1	b disparaît	$S = a$
F4	0	1	0	0	Se réduit à	$S = \bar{a}.b$
F5	0	1	0	1	a disparaît	$S = b$
F6	0	1	1	0	OU Exclusif / XOR	$S = \bar{a}.b + a.\bar{b}$
F7	0	1	1	1	OU / OR	$S = a + b$

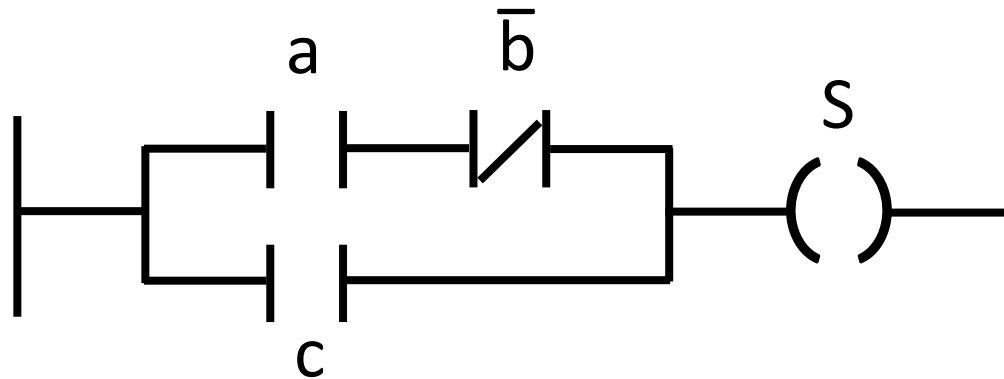
Fonctions booléennes 2/2

F _n	ab 00	ab 01	ab 10	ab 11	Nom de la fonction	Équation équivalente
F8	1	0	0	0	NON OU / NOR	$S = \overline{a + b}$
F9	1	0	0	1	Équivalence	$a = b$
F10	1	0	1	0	a disparaît	$S = \overline{b}$
F11	1	0	1	1	se réduit à	$S = a + \overline{b}$
F12	1	1	0	0	b disparaît	$S = \overline{a}$
F13	1	1	0	1	se réduit à	$S = \overline{a} + b$
F14	1	1	1	0	NON ET / NAND	$S = \overline{a \cdot b}$
F15	1	1	1	1	Toujours VRAIE	$S = 1$

Outils de représentation 1/3

La représentation du fonctionnement d'un S.A. en logique combinatoire peut se faire grâce à différents outils et langages.






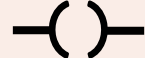
1. Schéma à contact : les variables y sont matérialisées par des interrupteurs ouverts ou fermés au repos. Les produits booléens deviennent des circuits en série tandis que les sommes booléennes deviennent des circuits parallèles. L'exemple ci-dessous représente en schéma à contact de la fonction $S = a.b + c$. La sortie est représentée le plus souvent par une bobine (), qui est l'élément électrique d'excitation d'un relais ou d'une électrovanne.



Outils de représentation 2/3

Le schéma à contact n'est pas un schéma d'installation électrique. Il n'exprime que des équations logiques et n'intègre par exemple pas les définitions matérielles des appareils, non plus que les énergies utilisées.

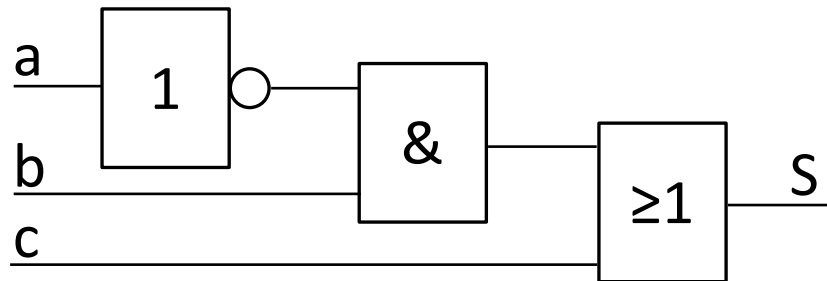
Plusieurs normes coexistent dans la représentation des variables d'entrées (contacts) et de sorties (bobines) :

	Contact NO/NO	Contact NF/NC	Bobine
Norme ISO			
Norme US			

Lorsque le nombre de variables augmentent et/ou que les fonctions à représenter deviennent trop complexes, les schémas à contact deviennent eux-mêmes difficiles à réaliser et lire. On leur préfère alors les schémas à portes logiques.

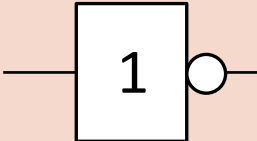
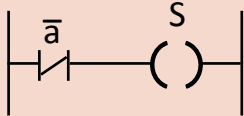
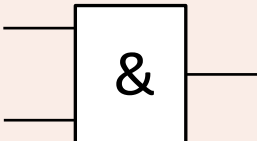
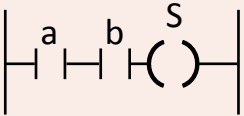
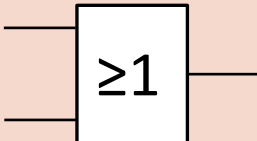
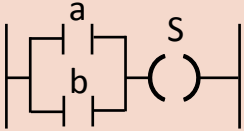
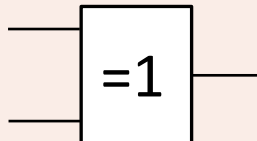
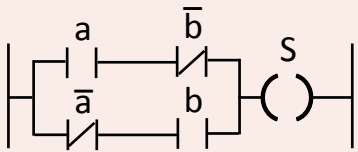
Outils de représentation 3/3

2. Schéma à portes logiques : les portes logiques réalisent les fonctions les plus intéressantes parmi les fonctions booléennes vues précédemment. Ces fonctions existent sous formes de circuits intégrés (discrets ou montés en surface). Ci-dessous le schéma à portes logiques de la fonction $S = a.\bar{b} + c$.

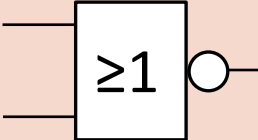
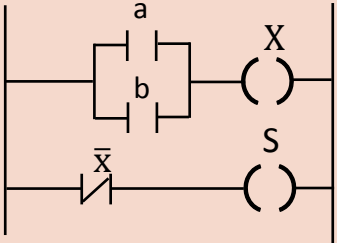
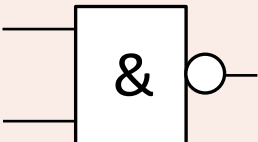
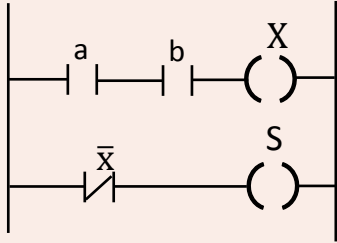


Les portes peuvent présenter une ou plusieurs entrées suivant la fonction réalisée. La présence d'un cercle (en entrée ou en sortie) correspond à la négation de la variable (comme ici pour \bar{a}).

Portes logiques 1/2

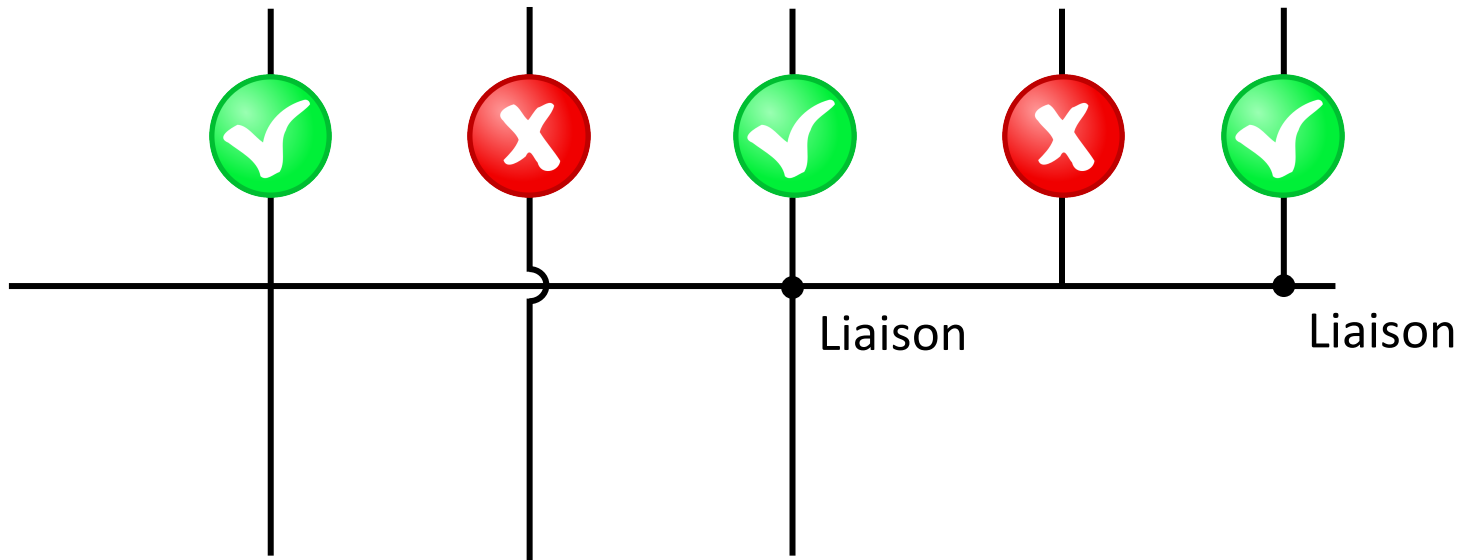
Fonction	Symbole	Schéma Contact équivalent
NON/ <i>NOT</i>		$S = \bar{a}$ 
ET/ <i>AND</i>		$S = a \cdot b$ 
OU/ <i>OR</i>		$S = a + b$ 
OU Exclusif/ <i>XOR</i>		$S = a \oplus b$ 

Portes logiques 2/2

Fonction	Symbole	Schéma Contact équivalent
NON OU/ <i>NOR</i>		$S = \overline{a + b}$ 
NON ET/ <i>NAND</i>		$S = \overline{a \cdot b}$ 

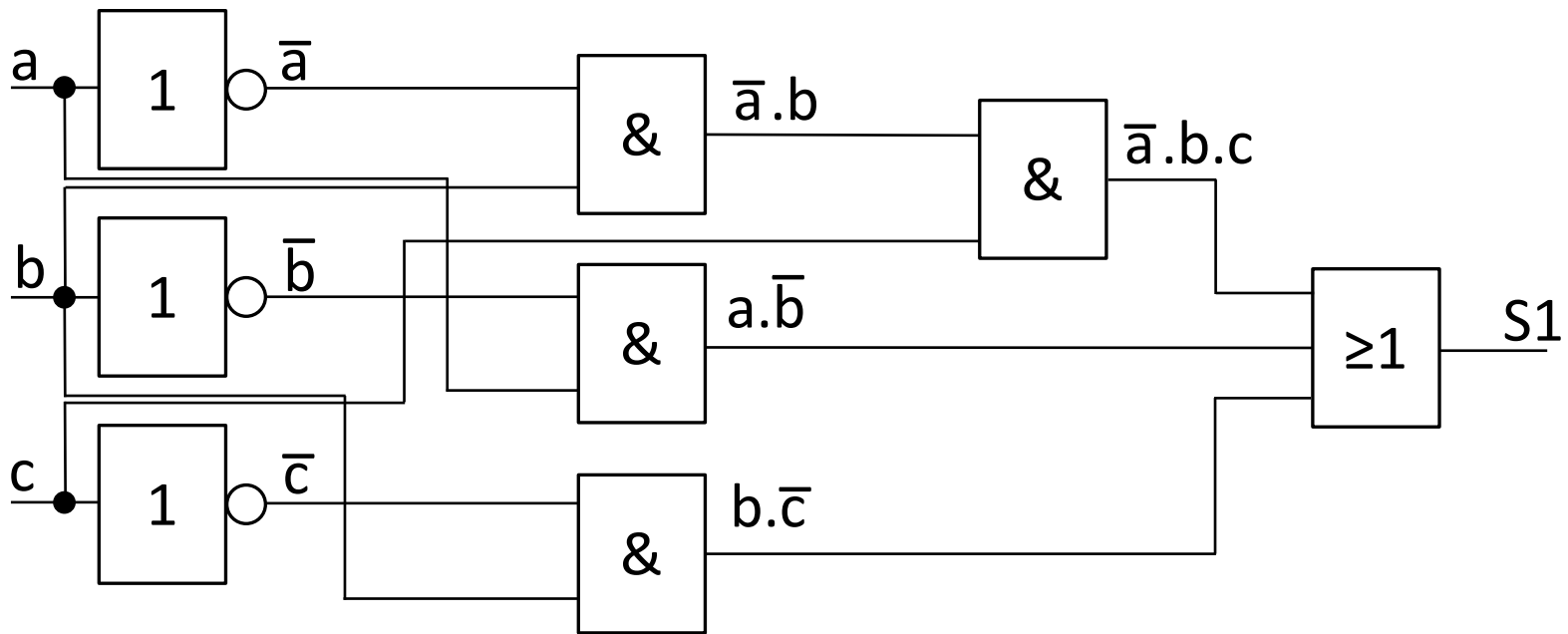
A propos des schémas

Règles de représentation des liaisons :



Un exemple $S1 = \bar{a}.b.c + a.\bar{b} + b.\bar{c}$

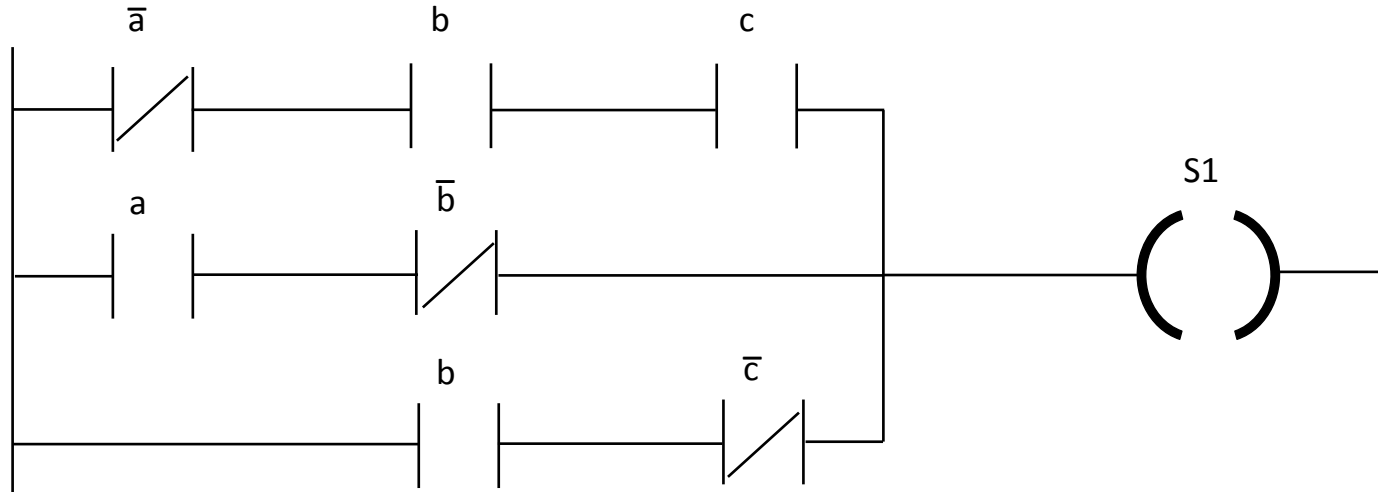
Schéma portes logiques :



Un exemple $S1 = \bar{a}.b.c + a.\bar{b} + b.\bar{c}$

Schéma à contacts

On retrouve les 3 circuits en parallèle :



Les contacts de même nom, ainsi que leur négation (par exemple b et \bar{b}), fonctionnent simultanément.

Algèbre de Boole 1/3



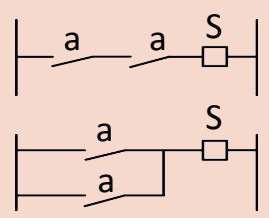
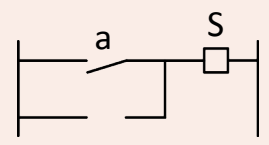
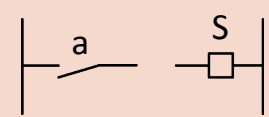
Propriétés de base.

Les produits et sommes booléens fonctionnent comme les produits et sommes réels. C'est pourquoi on peut utiliser les mêmes opérateurs. De même, on retrouve les mêmes propriétés :

Propriétés	Équation	Schéma contact
Commutativité	$a.b = b.a$ $a + b = b + a$	
Associativité	$a.(b.c) = (a.b).c$ $a+(b+c) = (a+b)+c$	
Distributivité	$a.(b+c) = a.b + a.c$	
Complément	$a + \bar{a} = 1$ $a.\bar{a} = 0$	

Algèbre de Boole 2/3



Propriétés	Équation	Schéma contact
Indempotence	$a.a = a$ $a + a = a$	
Éléments neutres	$a + 0 = a$ $a.1 = a$	
Éléments absorbants	$a.0 = 0$ $a + 1 = 1$	

Algèbre de Boole 3/3



Propriétés	Équation	Schéma contact
Absorption	$a.x + a = a \quad (\text{qqs } x)$ $a.b + \bar{b} = a + \bar{b}$ $a.\bar{b} + b = a + b$ $\bar{a}.b + \bar{b} = \bar{a} + \bar{b}$ $\bar{a}.\bar{b} + b = \bar{a} + b$	Ces propriétés seront démontrées ultérieurement !

Tables de vérités

Une table de vérité exprime le cahier des charges sous la forme d'un tableau, et ce du point de vue des relations sorties/entrées. Si un système possède 3 entrées et une sortie, le nombre de cas (donc de lignes du tableau) à examiner est de $2^3 = 8$. Ci-dessous un exemple de table de vérité à 3 variables.

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

Chaque ligne indique l'état de la sortie (0/1) pour une combinaison de a,b et c.

$$\bar{a}.\bar{b}.c$$

On « ajoute » ensuite les lignes où S vaut 1.

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

Simplification par calcul :

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

$$\bar{c} + c = 1 \text{ et } x.1 = x$$

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

$$a.b.\bar{c}$$

$$a.b.c$$