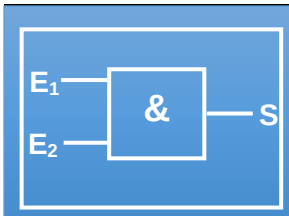


Les opérateurs booléens

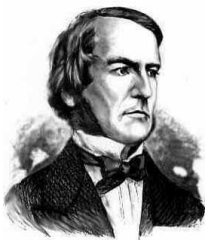
</>
01110

Objectifs pédagogiques :

- ✓ Comprendre la notion de valeurs booléennes (0,1)
- ✓ Savoir utiliser des opérateurs booléens :
- ✓ Dresser la table de vérité d'une expression booléenne

Les portes logiques sont des composants électroniques possédant 2 entrées et 1 sortie. Elles permettent de réaliser des opérations simples sur les bits (0,1) et leur association des opérations complexes. Elles servent à réaliser des calculs dans un ordinateur.

Algèbre de Boole et portes logiques



George BOOLE (1815-1864)

■ L'**algèbre de Boole,** ou calcul booléen, est la branche des mathématiques qui s'intéresse aux opérations et aux fonctions sur les **variables logiques**. Elle fut inventée par le mathématicien britannique George Boole. Aujourd'hui, l'algèbre de Boole trouve de nombreuses applications en informatique et dans la conception des circuits électroniques.
■ L'algèbre de Boole est une structure algébrique qui ne contient que deux éléments, que l'on appelle couramment **variables booléennes**. Ces variables ne peuvent avoir que deux états :

Variables booléennes		
0	**OU**	FAUX
1		VRAI

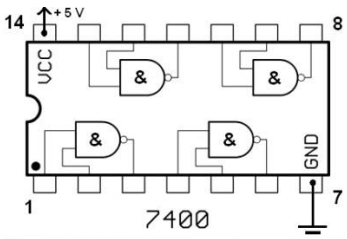
Document 1 : les variables booléennes

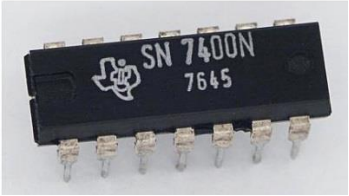
George Boole (1815-1864) est un logicien, mathématicien et philosophe britannique. Il est le créateur de la logique moderne, fondée sur une structure algébrique et sémantique, que l'on appelle algèbre de Boole en son honneur. Ses travaux ont contribué au développement de l'informatique.

■ L'algèbre de Boole utilise plusieurs **opérateurs** que l'on nomme de différentes façons : **opérateurs booléens**, **opérateurs logiques**, **fonctions logiques** ou **portes logiques** (terme plus propre à l'électronique). Un opérateur logique est une fonction mathématique qui donne des variables logiques en sortie à partir de variables logiques d'entrée.
■ Les 4 opérateurs logiques de base sont les suivants :

ET (AND)	OU (OR)	NON (NO)	OU exclusif (XOR)
▶ **ET** correspond à l'**intersection** de deux conditions. ▶ **a ET b (noté a x b) est VRAI** si et seulement si a est VRAI et b est VRAI	▶ **OU** correspond à la **réunion** de deux conditions. ▶ **a OU b (noté a + b) est VRAI** si et seulement si a est VRAI ou b est VRAI, ou si a et b sont VRAIS	▶ **NON** correspond au **contraire** d'une condition. ▶ Le **contraire** de a **(noté $\bar{a}$) est VRAI** si et seulement si a est FAUX.	▶ **XOR** (OU exclusif) correspond à l'intersection de deux conditions privées de la réunion de celles-ci. ▶ **a XOR b (noté a $\oplus$ b) est VRAI** si a OU b est VRAI sauf si a ET b sont VRAIS.

Document 2 : les opérateurs logiques (ou booléens)





Circuit intégré 7400 contenant 4 portes NAND.

■ L'informatique, l'électronique et différentes technologies numériques, utilisent des **circuits intégrés** comprenant des **portes logiques**. Les portes logiques permettent de combiner les **signaux logiques** selon les lois de l'algèbre de BOOLE. Il existe de nombreuses familles logiques basées sur deux technologies : les familles **TTL** qui utilisent des **transistors bipolaires** et les familles **MOS** qui utilisent des **transistors à effet de champ**.

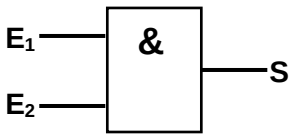
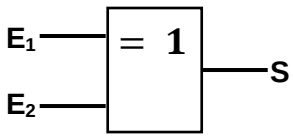
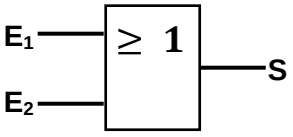
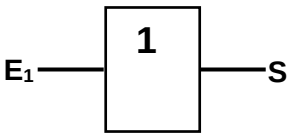
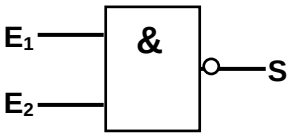
■ Une porte logique possède **deux entrées** notées E_1 et E_2 et une **sortie S**.

■ Les deux entrées E_1 et E_2 sont décrites par des **états électriques** « **haut** » (5,0 V par exemple) ou « **bas** » (0,0 V par exemple). On associe à ces deux états électriques les valeurs binaires « **0** » pour l'état « **bas** » et « **1** » pour l'état « **haut** ».

■ A chaque porte logique est associé une **table de vérité** qui donne l'état logique de la sortie S en fonction des états logiques des entrées E_1 et E_2.

Document 3 : notion de porte logique

Q1. Complétez les tables de vérité des portes logiques suivantes. Aidez-vous si besoin du simulateur en ligne de circuits logiques logic.ly afin de vérifier vos résultats.

Porte logique Fonction logique	Symbole normalisé français	Opération booléenne entre E_1 et E_2	Table de vérité
ET (AND)		$E_1 \times E_2$	E_1 E_2 S 0 0 0 1 1 0 1 1
OU (OR)		$E_1 + E_2$	E_1 E_2 S 0 0 0 1 1 0 1 1
OU exclusif (XOR)		$E_1 \oplus E_2$	E_1 E_2 S 0 0 0 1 1 0 1 1
NON (NO)		$\overline{E_1}$	E_1 S 0 1
NON-ET (NAND)		$\overline{E_1 \times E_2}$	E_1 E_2 S 0 0 0 1 1 0 1 1

Document 4 : symboles normalisés européens des portes logiques de base et tables de vérité associées

Quelques propriétés mathématiques des fonctions logiques

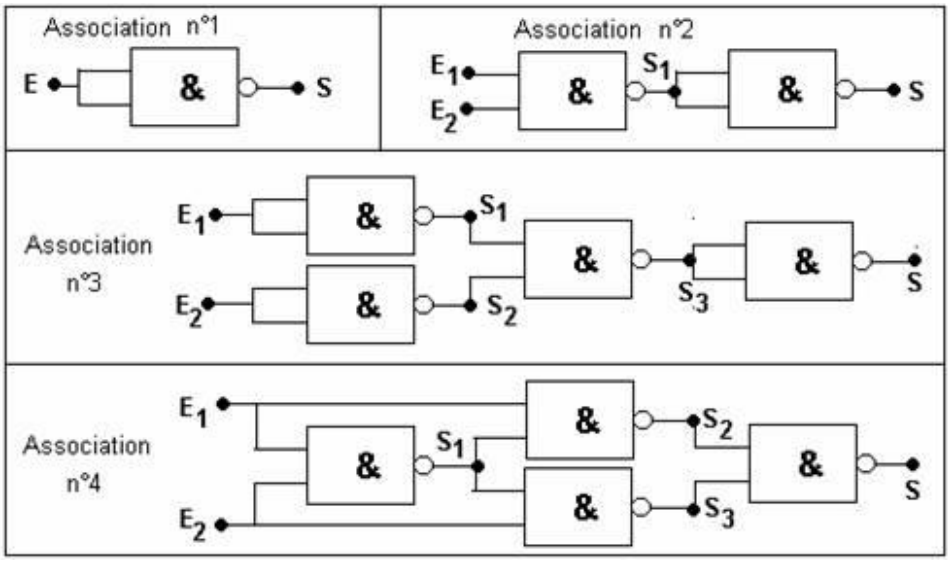
Soient E_1, E_2 et E_3 des variables logiques :

- **Commutativité :** $E_1 + E_2 = E_2 + E_1$; $E_1 \times E_2 = E_2 \times E_1$
- **Associativité :** $E_1 + (E_2 + E_3) = (E_1 + E_2) + E_3$; $E_1 \times (E_2 \times E_3) = (E_1 \times E_2) \times E_3$
- **Distributivité :** $E_1 \times (E_2 + E_3) = (E_1 \times E_2) + (E_1 \times E_3)$; $E_1 + (E_2 \times E_3) = (E_1 + E_2) \times (E_1 + E_3)$

- **Théorèmes de De Morgan :** $\overline{E_1 \times E_2} = E_1 + E_2$; $\overline{E_1 + E_2} = E_1 \times E_2$

Document 5 : quelques propriétés mathématiques des fonctions logiques

Association de portes logiques NON ET (NAND)



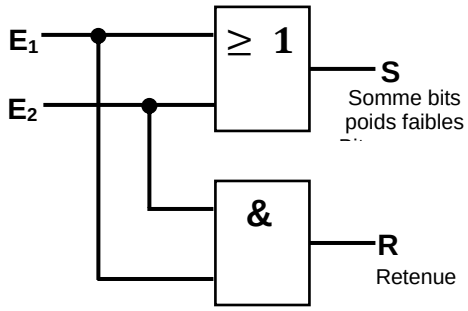
Q2. Déterminez la table de vérité de chacune des associations de portes NAND précédentes. Les états intermédiaires devront apparaître explicitement pour chacune d'entre elles.
Q3. Comment faut-il associer 3 portes NON ET pour créer une fonction OU ?
Q4. Pourquoi la fonction NON ET est-elle qualifiée d'universelle ?

Remarque : le circuit électronique en CMOS des portes logiques NON ET étant très simple à fabriquer, elles servent souvent de « brique de base » à des circuits intégrés beaucoup plus complexe.

Quelques applications des portes logiques

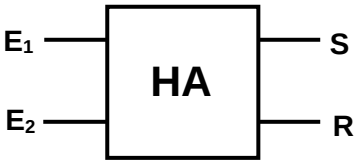
On utilise des associations de portes logiques pour réaliser des opérations mathématiques dans les calculatrices ou les ordinateurs.

> Le demi-additionneur



▶ Pour additionner deux nombres, il faut d'abord additionner les 2 bits de poids faible, puis additionner les bits suivants sans oublier les retenues.

▶ Un demi-additionneur (ou **H**alf **A**dder **HA**) est un circuit qui additionne les 2 bits de poids faibles pour lesquelles la retenue propagée n'est pas prise en compte.

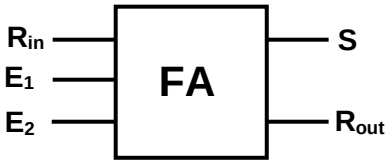


Q5. Complétez la table de vérité du demi additionneur. Aidez-vous si besoin du simulateur en ligne de circuits logiques logic.ly afin de vérifier vos résultats (astuce : reliez d'abord les portes logiques aux interrupteurs).

Q6. Pourquoi parle-t-on de demi-additionneur ?

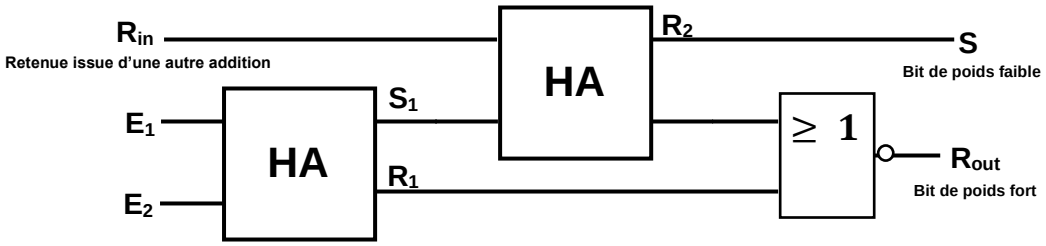
Entrées		Sorties	
E_1	E_2	R	S
0	0		
0	1		
1	0		
1	1		

> L'additionneur complet 1 bit



▶ Pour additionner les bits de poids supérieur et tenir compte de la propagation de la retenue, il faut recourir à un « Full Adder » ou additionneur complet qui nécessite une entrée supplémentaire par rapport au demi-additionneur.

▶ Un « Full Adder FA» peut additionner 2 bits (E_1,E_2) avec une retenue (R), soit trois entrées et deux sorties.



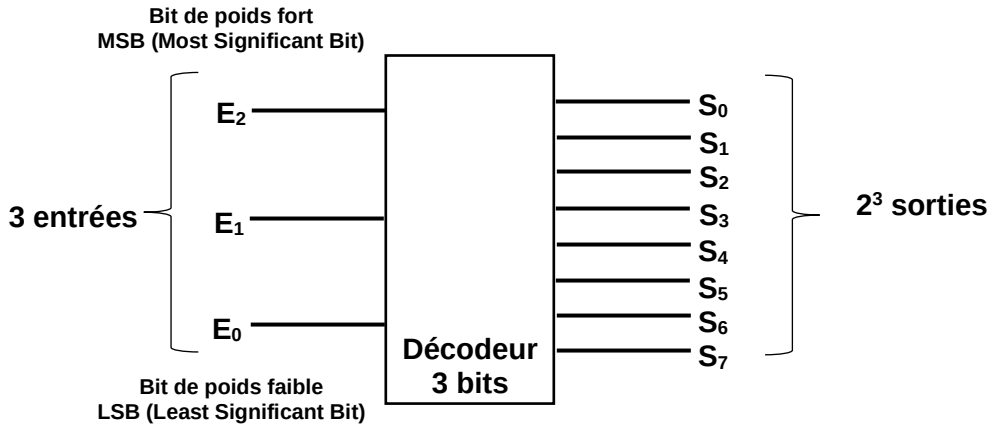
Q7. Complétez la table de vérité de l'additionneur complet 1 bit. Aidez-vous si besoin du simulateur en ligne de circuits logiques logic.ly afin de vérifier vos résultats (astuce : reliez d'abord les portes logiques aux interrupteurs).

Entrées			Sorties	
E_1	E_2	R_{in}	R_{out}	S
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

> Le décodeur 3 bits

■ Dans un système numérique les informations, telles que les nombres entiers, sont transportées dans les différents dispositifs électroniques sous forme de mots binaires. Par exemple un mot binaire de 3 bits code 2^3 chiffres entiers différents.

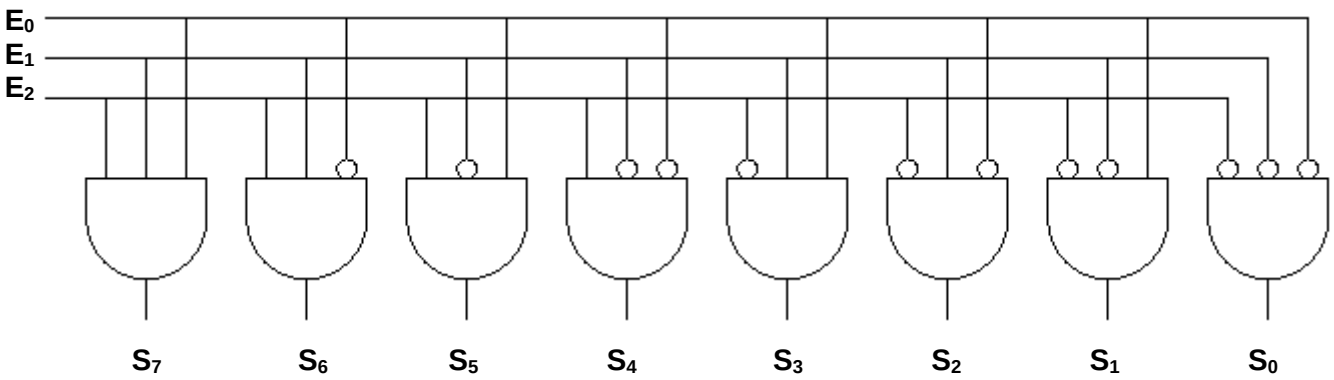
■ Un décodeur est un système numérique qui possède N entrées et 2^N sorties. Pour chacune des combinaisons possibles des entrées, seule une ligne de sortie est active. Un décodeur est constitué par une association de portes logiques plus ou moins complexe.



Entrées			Sorties								
E_2	E_1	E_0	S_0	S_1	S_2	S_3	S_4	S_6	S_6	S_7	DEC
0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	1
0	1	0	0	0	1	0	0	0	0	0	2
0	1	1	0	0	0	1	0	0	0	0	3
1	0	0	0	0	0	0	1	0	0	0	4
1	0	1	0	0	0	0	0	1	0	0	5
1	1	0	0	0	0	0	0	0	1	0	6
1	1	1	0	0	0	0	0	0	0	1	7

Dans le décodeur binaire 3 bits ci-dessus, le numéro de la sortie active donne le nombre en base 10 appliqué en entrée sous forme binaire. Ce décodeur permet donc la conversion d'un nombre binaire 3 bits en sa représentation en base 10.

Le décodeur 3 bits est constitué d'une association de portes logiques **ET** (symbole anglo-saxon ci-dessous) sur lesquelles on applique sur certaines entrées d'entre elles une fonction **NON**.



Document 6 : décodeur 3 bits BIN - DEC