

E7.1. But de la manipulation

- Comprendre et être capable de construire des fonctions logiques
- Construire des circuits électroniques basés sur ces notions de fonctions logiques

E7.2. Introduction**E7.2.1. Fonctions logiques**

Une **variable binaire** est une variable ne pouvant prendre que deux valeurs, conventionnellement 1 ou 0. Une **fonction logique** est une fonction d'une ou de plusieurs variables binaires; ses valeurs, 1 ou 0 également, sont données, pour toutes les combinaisons possibles des valeurs des variables binaires dont elle dépend, par un tableau appelé **table de vérité**.

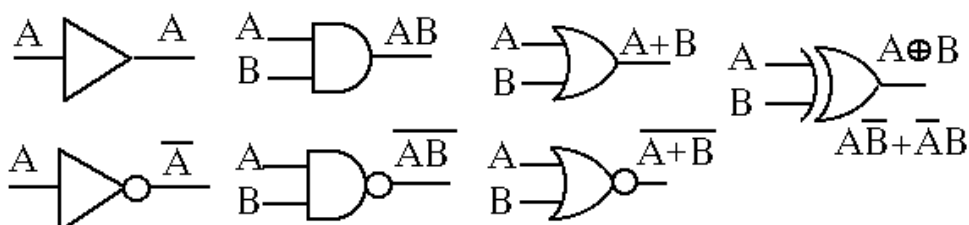
E7.2.2. Fonctions logiques élémentaires

Tables de vérité					
Opérateur NOT (NON)			Opérateur AND (ET)		
A		\bar{A}	A	B	$A \cdot B$
1		0	0	0	0
0		1	0	1	0
			1	0	0
			1	1	1
Opérateur OR (OU)			Opérateur XOR (OU exclusif)		
A	B	$A+B$	A	B	$A \oplus B$
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Les tables de vérité ci-dessus présentent quatre fonctions logiques élémentaires:

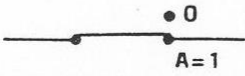

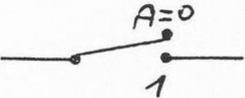

- la fonction NOT (NON) : \bar{A} ;
- la fonction AND (ET) c'est-à-dire la fonction $A \cdot B$ (produit logique);
- la fonction OR (OU) c'est-à-dire la fonction $A+B$ (somme logique);
- la fonction XOR (OU exclusif) c'est-à-dire la fonction $A \oplus B = (A \cdot \bar{B}) + (\bar{A} \cdot B)$

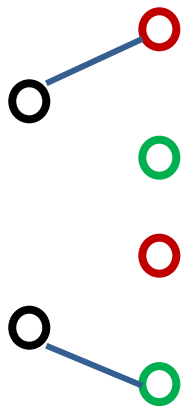
S'y ajoutent les fonctions NAND (NON-ET) et NOR (NON-OU).



E7.2.3 Circuits logiques

■ Un circuit logique est un circuit électrique ou électronique matérialisant une fonction logique. Ainsi les deux positions d'un interrupteur à deux directions (interrupteur ouvert ou fermé) peuvent-elles représenter les deux valeurs d'une variable binaire; il en est de même des deux états de fonctionnement (conducteur ou non-conducteur) d'une diode, plus généralement de deux états différents de potentiel électrique en un point donné d'un certain circuit (voir ci-dessous).

Valeur logique	Interrupteur 2 directions	Diode électroluminescente
1		
0		



interrupteur en position "haute"
peut représenter 1 ou 0

interrupteur en position "basse"
peut représenter 0 ou 1

LED de contrôle (output)



LED éteinte = 0

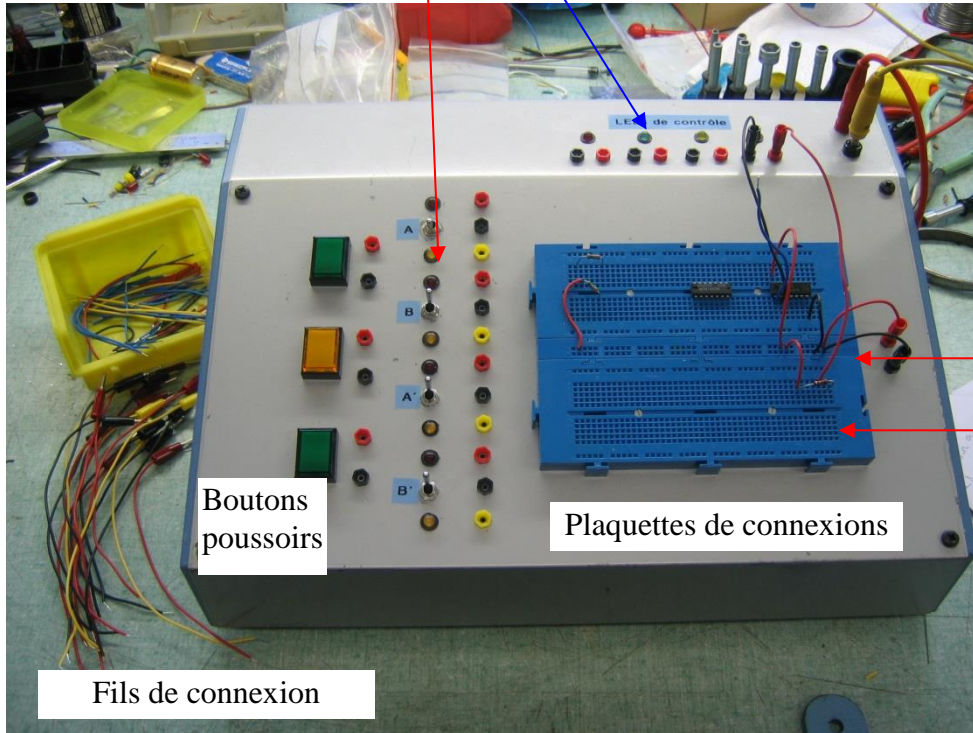


LED allumée = 1

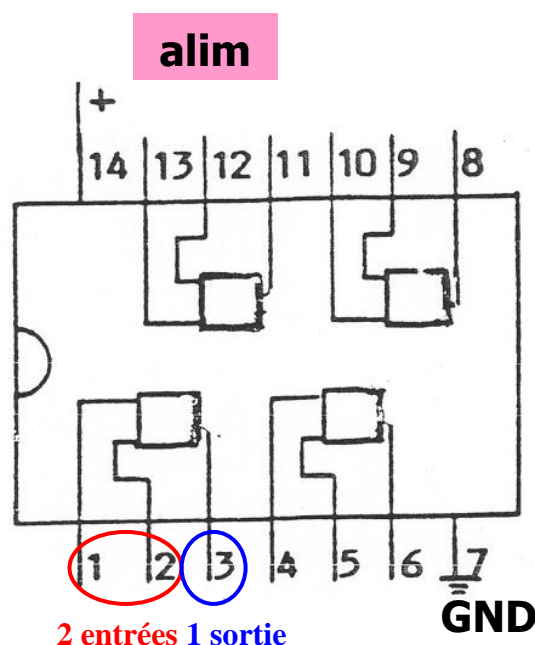
E7.3. Dispositif expérimental

On dispose d'un boîtier (cf. photo) muni notamment de :

- **4 interrupteurs à 2 directions** (Normalement Ouvert "0" / Normalement Fermé "1") et munis de LEDs indiquant l'état "haut" ou "bas", à définir en tant que "0" ou "1",
- **4 LEDs de contrôle** (Light Emitting Diodes, diodes électroluminescentes en Français) de couleur différente et indépendantes dont la fonction est de déterminer l'état du circuit établi.



- De plus, on dispose de: résistances, fils de connexion, source de tension extérieure continue à régler sur + 5V.
- L'élément principal à étudier est un circuit intégré SN7400 comportant 4 circuits logiques simples (cf. schéma); ces circuits sont tous identiques et donc représentés par un même pictogramme (rectangle dessiné à main levée). Un des buts de la manipulation est justement de déterminer le type de fonction logique réalisée par chacun de ces circuits de base.



Schématisation
circuit intégré
SN7400

E7.4. manipulation

E6.4.1 Préambule

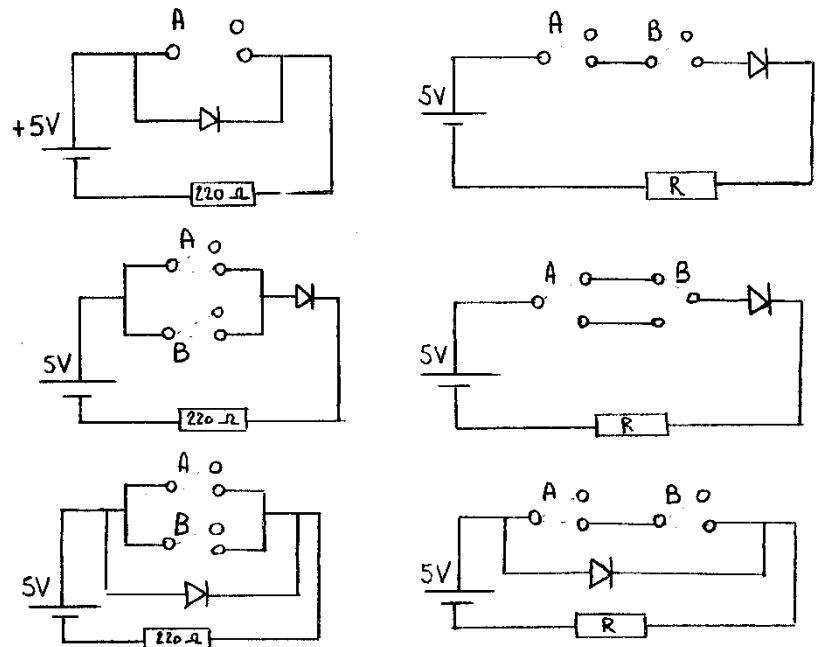
- 1) Démontrer que : $A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$
- 2) Construire la table de vérité de la fonction NAND et démontrer aussi que $\overline{A \cdot B} = \overline{A} + \overline{B}$
- 3) Construire la table de vérité de la fonction NOR et démontrer aussi que $\overline{A + B} = \overline{A} \cdot \overline{B}$

E7.4.2 Première expérience : étude de circuits logiques simples

Dans les six circuits logiques schématisés ci-contre, un ou deux interrupteurs sont utilisés comme variables binaires.

Une convention possible pour la valeur de la variable représentée par les positions des interrupteurs peut être : 0 en "haut", 1 en "bas".

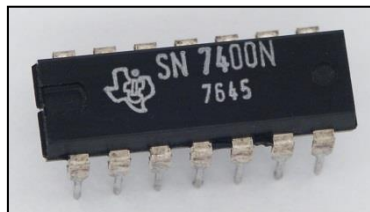
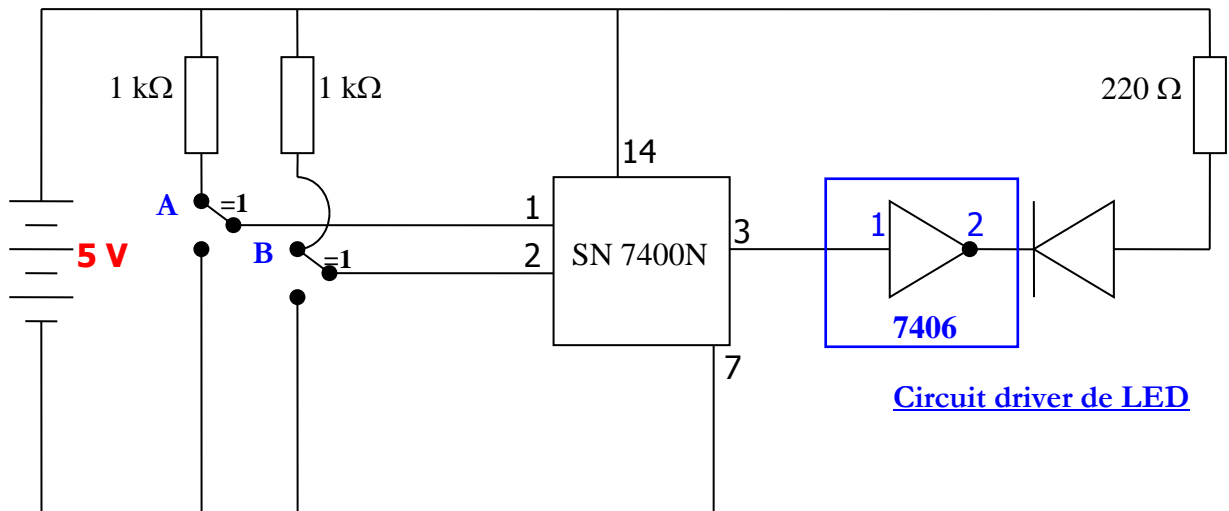
L'état de fonctionnement d'une LED de contrôle est utilisée comme fonction logique de résultat ou de sortie (*output*). La valeur 1 est attribuée à l'état conducteur de la diode.



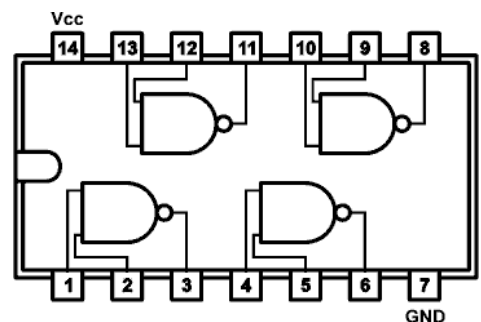
- 1) Quelle est la fonction logique réalisée par le premier circuit ne comportant qu'un seul interrupteur "A" ? Réalisez le circuit et vérifiez cette prédiction. **Couper l'alimentation en fin de réalisation.**
- 2) Pour chacun des cinq autres circuits, essayez de prévoir la fonction logique que le circuit représente (Ex : AND, OR, XOR, NAND,). Réalisez alors chacun des circuits et déterminez expérimentalement le circuit établi (vérifiez ainsi vos prévisions). **ATTENTION : couper l'alimentation entre les différents montages !**
- 3) La convention relative aux interrupteurs est-elle importante ?

E7.4.3 Seconde expérience: étude de circuits intégrés logiques

- 1) Réalisez le circuit schématisé à la figure ci-dessous et utilisant une des quatre fonctions logiques identiques disponibles sur le circuit intégré "SN7400". Identifiez cette fonction. La valeur 1 des deux variables binaires A et B correspond, pour l'une comme pour l'autre et comme indiqué, à la fermeture d'un interrupteur sur l'élément intégré (entrées 1 et 2 de cet élément). La valeur 1 de la fonction logique "de sortie" correspond à nouveau à l'état conducteur d'une diode électroluminescente ou LED "allumée".

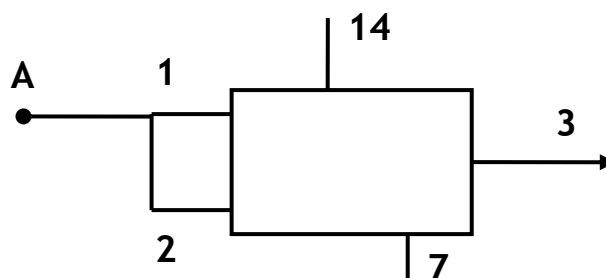


A	B	S
1	1	
1	0	
0	1	
0	0	



- 2) A partir du résultat ci-dessus, **prévoir** ce qu'il se passe si on court-circuite les entrées 1 et 2 comme schématisé ci-dessous ?

A	S
1	
0	

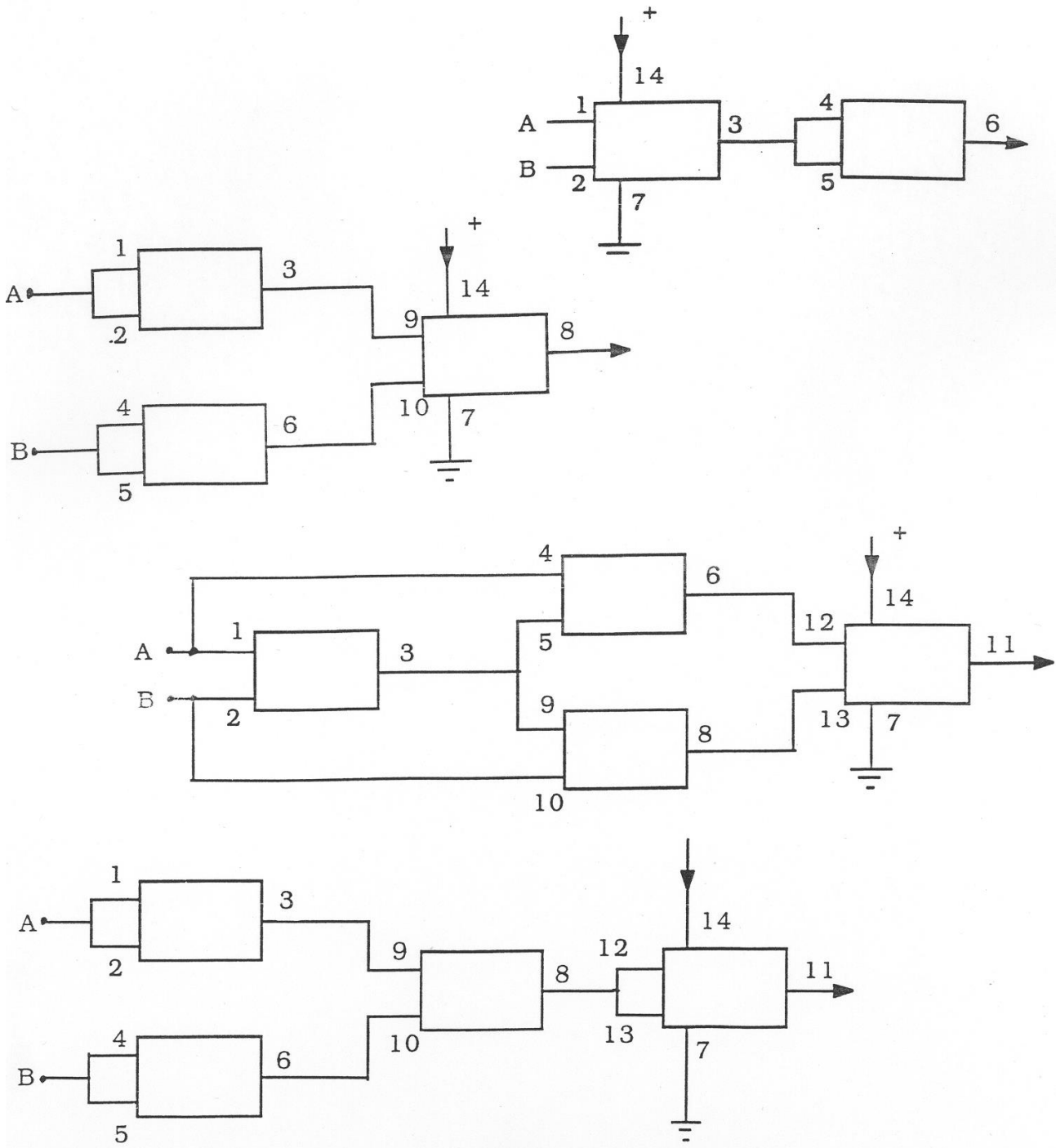


- 3) Cette page présente 4 montages connectant entre eux plusieurs des quatre éléments intégrés identiques dont on dispose sur le SN7400. Réalisez TOUS les montages présentés afin de déterminer les fonctions logiques ainsi réalisées.

Si possible, expliquez. Pour vous aider, vous pouvez consulter vos autres cours ou des sites d'électronique comme par exemple:

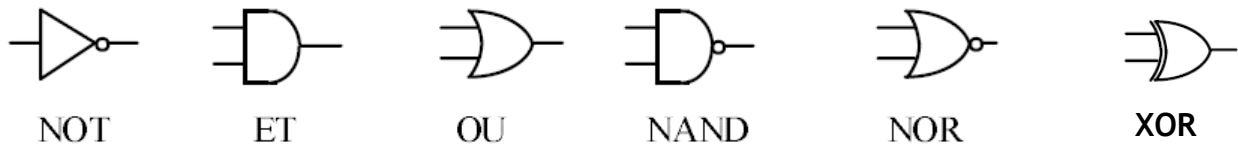
<http://www.univ-lemans.fr/enseignements/physique/02/electro/mnueltro.html>

Clic sur "portes logiques" et la suite.



Annexe :

Représentation symbolique des opérateurs sous forme de portes logiques

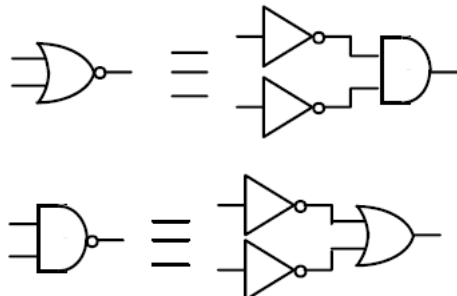


L'inverseur (NOT) correspond à la fonction complémentation. Les autres portes ont le même nom que les fonctions logiques correspondantes.

Théorèmes de De Morgan

$$\text{NAND}(a, b) = \overline{a \cdot b} = \bar{a} + \bar{b}$$

$$\text{NOR}(a, b) = \overline{a + b} = \bar{a} \cdot \bar{b}$$

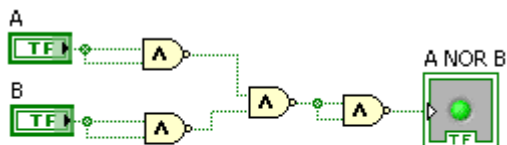


non (A et B) = (non A) ou (non B)

non (A ou B) = (non A) et (non B)

Application: réalisation de toutes les fonctions logiques à partir de portes NAND

Le théorème de Morgan permet de transformer toutes les portes logiques en portes NAND ou NOR. A contrario, toutes les fonctions logiques (i.e., AND, OR, NOT, XOR, XNOR) peuvent être réalisées à partir d'un réseau de portes NAND (ou de portes NOR). Historiquement, les portes NAND étaient plus faciles à construire en utilisant la technologie MOS¹ et donc les portes NAND ont constitué le premier pilier de la logique booléenne dans le calcul électronique.



A NOR gate can be constructed from this combination of 4 NAND gates

Exemple d'implémentation de la fonction NOR sous forme de portes NAND

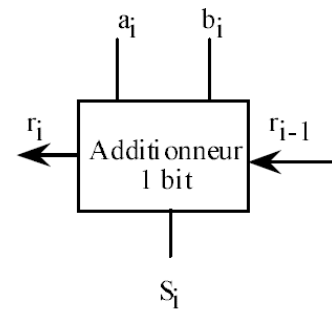
¹ **MOS Technology**, également connue sous le nom de **Commodore Semiconductor Group**, fut une société américaine d'électronique et un fabricant de [microprocesseurs](http://fr.wikipedia.org/wiki/MOS_Technology_6502). La société est particulièrement connue pour avoir conçu la gamme des microprocesseurs [MOS Technology 6502](http://fr.wikipedia.org/wiki/MOS_Technology_6502) (8-bits μ pro conçu en 1975) qui équipa notamment les micro-ordinateurs [Commodore 64](http://fr.wikipedia.org/wiki/Commodore_64). C'était à l'époque le moins coûteux des unités CPU sur le marché (1/6 du prix par rapport aux concurrents comme [Motorola](http://fr.wikipedia.org/wiki/Motorola) et [Intel](http://fr.wikipedia.org/wiki/Intel)).

En voir plus sur : http://fr.wikipedia.org/wiki/MOS_Technology_6502

Application : réalisation de l'additionneur 1 bit à l'aide de portes NAND

L'addition de deux bits a_i et b_i avec la retenue entrante r_{i-1} fournit une somme S_i et une retenue r_i selon la table ci-dessous. Le schéma logique de l'additionneur correspondant, appelé additionneur 1 bit, est donné à la figure à coté du tableau.

r_{i-1}	a_i	b_i	r_i	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Cet additionneur peut être implanté avec des portes logiques de type NAND. En effet, on a les relations :

$$S_i = \overline{r_{i-1}} \cdot \overline{a_i} \cdot \overline{b_i} + \overline{r_{i-1}} \cdot a_i \cdot \overline{b_i} + r_{i-1} \cdot \overline{a_i} \cdot b_i + r_{i-1} \cdot a_i \cdot b_i \quad \& \quad r_i = a_i \cdot b_i + r_{i-1} \cdot a_i + r_{i-1} \cdot b_i$$

Un schéma d'implémentation directe de la table de vérité de l'additionneur 1 bit avec des portes NAND (à 2 ou plusieurs entrées) est donné ci-dessous.

