TD1 - Architecture de l'ordinateur et Circuits logiques
Partie 1 : Les circuits combinatoires

Exercice 1 (Codage des entiers positifs en base 2)

Question 1
Que valent (1101)2, (1010)2 et (0001010)2 ?

Question 2
Donnez un codage binaire de 5, 10 et 11.

Question 3
Expliquez comment obtenir un codage de 2 x *X* lorsque l'on connait déjà un codage de *X*.

Question 4
Expliquez comment obtenir un codage de 2 x *X* + 1 lorsque l'on connait un codage de *X*.

Question 5
Expliquez comment obtenir un codage de *Y* lorsque l'on connait un codage de *Y* ÷ 2. Vous pourrez distinguer les cas où *Y* est pair ou impair.

Question 6
Donnez un codage binaire de 72 à l'aide de divisions euclidiennes successives.

Exercice 2 (Construction d'un circuit combinatoire compteur)
On souhaite réaliser un circuit à trois entrées binaires A, B et C et une sortie S = (s_1; s_0) sur 2 bits qui a pour fonction d'indiquer en base 2 le nombre d'entrées qui sont égales à 1. Par exemple, si A = B = 1 et C = 0, on aura $s_1 = 1$ et $s_0 = 0$, puisque, d'une part, il y a dans ce cas deux entrées égales à 1 et, d'autre part, 2 s'écrit $(10)_2$ en binaire.

Question 1
Construire les tables de vérité de s_0 et s_1.

Question 2
Donner une équation correspondant à chacune des deux sorties s_0 et s_1.

Question 3
Vérifier que $s_0 = A \oplus B \oplus C$ et que $s_1 = AB + BC + CA$.

Question 4
Dessiner le circuit associé.

Exercice 3 (Construction d'un circuit combinatoire additionneur)
L'addition de nombres écrits en binaire procède de la même manière qu'en base 10 : en commençant par les unités et en conservant les retenues, avec la particularité que $1+1 = (10)_2$. Le calcul ci-dessous

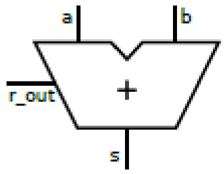
illustre l'addition de deux nombres x et y qui s'écrivent sur 4 bits : $x = (1011)_2$ et $y = (1110)_2$, les retenues étant indiquées en italique.

$$
\begin{array}{ccccccc}
 & & & \mathit{1} & \mathit{1} & & \\
 & & & 1 & 0 & 1 & 1 \\
+ & & & 1 & 1 & 1 & 0 \\
\hline
 & & 1 & 1 & 0 & 0 & 1
\end{array}
$$

Effectivement $x = (1011)_2 = 11$, $y = (1110)_2 = 14$ et $(11001)_2 = 25 = x + y$.

Question 1

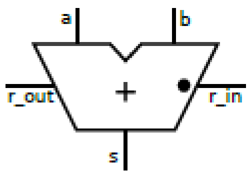
Construisez un circuit demi-additionneur (sur un bit). Ce circuit correspond au calcul réalisé sur la première colonne lors d'une addition : les entrées seront a et b (les deux bits à additionner) et les sorties s et r_{out} seront telles que la somme des 2 entrées vaut $(r_{out}s)_2$.



Par exemple, si a = b = 1 alors s = 0 et $r_{out} = 1$.

Question 2

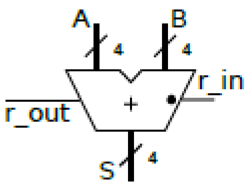
En utilisant ce demi-additionneur comme élément constitutif, construisez un circuit additionneur complet (sur un bit), c'est-à-dire un circuit qui correspond au calcul réalisé sur les autres colonnes lors d'une addition : les entrées seront a et b (les deux bits à additionner) et r_{in} (la retenue entrante); les sorties s et r_{out} seront telles que la somme des 3 entrées vaut $(r_{out}s)_2$.



Par exemple, si $a = b = r_{in} = 1$ alors s = 1 et $r_{out} = 1$.

Question 3

Réalisez un module additionneur à 4 bits en utilisant des additionneurs complets.



Les entrées seront :

- l'entier $A = (a_3a_2a_1a_0)_2$ sur 4 bits,
- l'entier $B = (b_3b_2b_1b_0)_2$ sur 4 bits
- et le bit r_{in} (la retenue entrante).

Les sorties seront :

- l'entier $S = (s_3s_2s_1s_0)_2$ sur 4 bits
- et le bit r_{out}: la retenue de dépassement

tels que $(r_{out}s_3s_2s_1s_0)_2 = A + B + r_{in}$, c'est-à-dire $2^4 \times r_{out} + S = A + B + r_{in}$.

Par exemple, si $A = (1011)_2$, $B = (1110)_2$et $r_{in} = 0$ alors $S = (1001)_2$ et $r_{out} = 1$.

Exercice 4 (Codage hexadécimal des entiers positifs)

Si dans la vie de tous les jours les nombres sont écrits en base 10 (avec 10 chiffres) et si au niveau du matériel les calculs sont réalisés en binaire (avec deux chiffres), l'informaticien utilise souvent le codage hexadécimal, avec 16 chiffres : les chiffres habituels pour les dix premiers chiffres et les lettres A à F pour les six suivants.

Ainsi, une suite de chiffres hexadécimaux (c_n; c_{n-1}; ... ; c_1; c_0) représente l'entier $X = c_0 + 16.c_1 + ... + 16^n.c_n$ où A vaut 10, B vaut 11, C vaut 12, D vaut 13, E vaut 14 et F vaut 15.

Par convention, ce nombre est noté $(c_nc_{n-1}...c_1c_0)_{16}$ ou encore $0xc_nc_{n-1}...c_1c_0$, le préfix "0x" servant à signaler, comme dans de nombreux langages informatiques, un codage en base 16.

Question 1
Que vaut l'entier 0x3F en décimal ?

Question 2
Que vaut 132 en hexadécimal ?

Question 3
Que vaut l'entier 0x5B en binaire ?

Question 4
Que vaut $(1010111100110101)_2$ en hexadécimal ?

Exercice 5 (Construction de portes logiques à l'aide de transistors CMOS)
Le transistor est la brique avec laquelle sont construits tous les circuits électroniques. La technologie actuellement utilisée pour fabriquer les microprocesseurs est la technologie MOS (Metal-Oxide-Semicon-ductor). Il existe deux types de transistors MOS : les transistors de type n (pour négatif) et les transistors de type p (pour positif). Les microprocesseurs actuels utilisent systématiquement des transistors des deux types. On parle donc de technologie CMOS (pour Complementary Metal-Oxide-Semiconductor).

Les transistors MOS disposent de 3 broches : une entrée principale, appelée grille (base), qui détermine son fonctionnement et deux ports secondaires, appelés respectivement source (collecteur) et drain (émetteur). Un transistor se comporte comme un interrupteur électrique entre la source et le drain qui serait commande par la grille.

Ainsi, un transistor peut être en mode bloquant ou passant (saturé) :
- en mode bloquant, le transistor correspond à un interrupteur ouvert, c'est-à-dire à une absence de connexion entre les deux ports secondaires.
- en mode passant, le transistor correspond à un interrupteur fermé, c'est-à-dire à une simple connexion entre les deux ports secondaires (le courant circule de la source vers le drain);

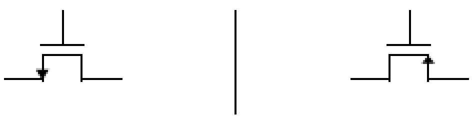


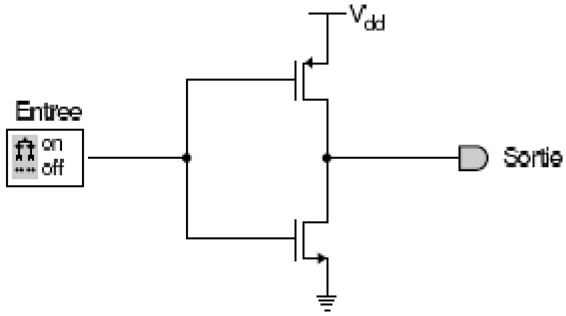
Figure 1: Symboles des deux transistors CMOS

Le transistor p-MOS (de type p) est en mode passant lorsque son entrée principale est à 0 et en mode bloquant lorsque son entrée principale est à 1. A l'inverse, le transistor n-MOS (de type n) est en mode passant lorsque son entrée principale est à 1 et en mode bloquant lorsque son entrée principale est à 0.

Les deux symboles de ces deux transistors sont représentés sur la figure 1. La grille correspond dans chaque cas à la broche verticale.

Question 1

Le circuit ci-dessous est composé d'une entrée binaire (à gauche), d'une sortie binaire (à droite), d'une alimentation positive V_{dd}, qui correspond à la valeur logique 1 (en haut), d'une masse (en bas), qui correspond à la valeur logique 0, et de deux transistors complémentaires, l'un positif, l'autre négatif, qui ont chacun un port relié à la sortie. La table de vérité de ce circuit est indiquée sur sa droite.

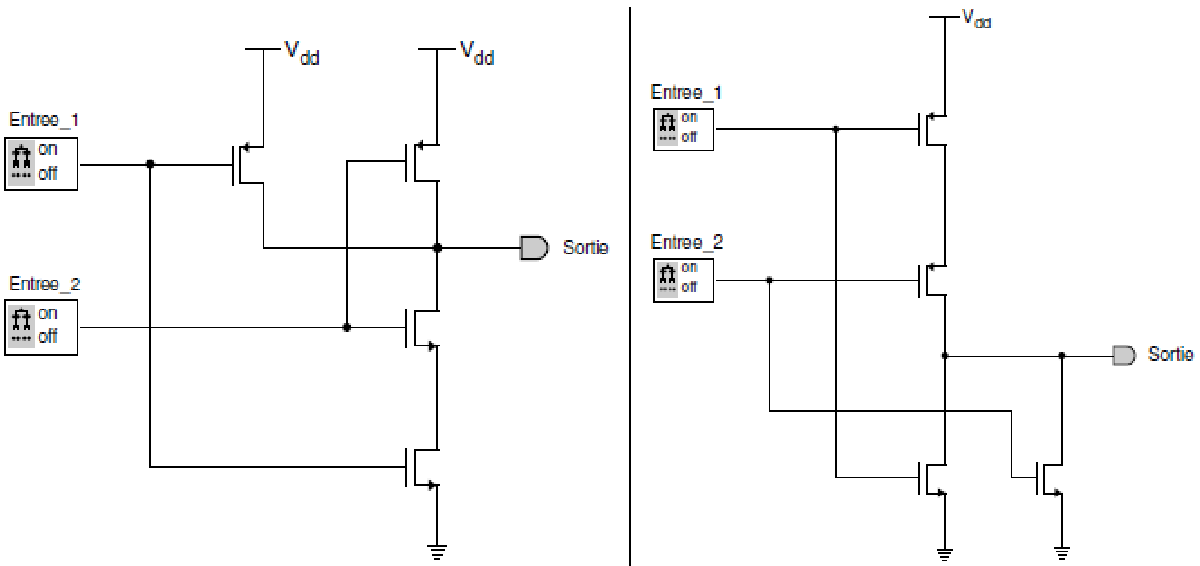


Entrée	Sortie
0	1
1	0

Déterminez quel est le transistor de type positif dans ce circuit.

Question 2

En déduire la table de vérité des deux circuits de la figure 2.



Entrée 1	Entrée 2	Sortie
0	0	
0	1	
1	0	
1	1	

Entrée 1	Entrée 2	Sortie
0	0	
0	1	
1	0	
1	1	

Figure 2: Tables à compléter de deux circuits mystérieux

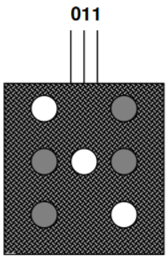
Exercice 6 (Construction de portes logiques élémentaires)

Construire, à l'aide de portes NOT (inverseurs), de portes NAND et de portes NOR :

- Une porte AND;
- Une porte OR;
- Une porte XOR.

Exercice 7 Le dé électronique

Le dé électronique possède trois entrées, par lesquelles arrivent des nombres de 1 à 6 codés en binaire. Donnez une fonction pour chacune des lampes qui indiquera si elle doit être allumée où éteinte et en déduire le logigramme associé.



Exercice 8 (Décodage 7 segments hexadécimal)

Un afficheur 7 segments est composé de 7 diodes électroluminescentes notés a, b, ..., g (voir figure 3). Un nombre est fourni, codé en binaire sur 4 bits $x_3x_2x_1x_0$, qui devra être affiché. Donnez l'expression des 7 fonctions $a(x_3x_2x_1x_0)$, $b(x_3x_2x_1x_0)$, ... et trouver le schéma du composant X.

Ses 4 entrées correspondent à la représentation binaire d'un chiffre entre 0 et 15. Il faut fournir en sortie les 7 signaux nécessaires à l'affichage du chiffre hexadécimal correspondant. On suppose qu'il faut un 0 pour allumer un segment, et un 1 pour l'éteindre.

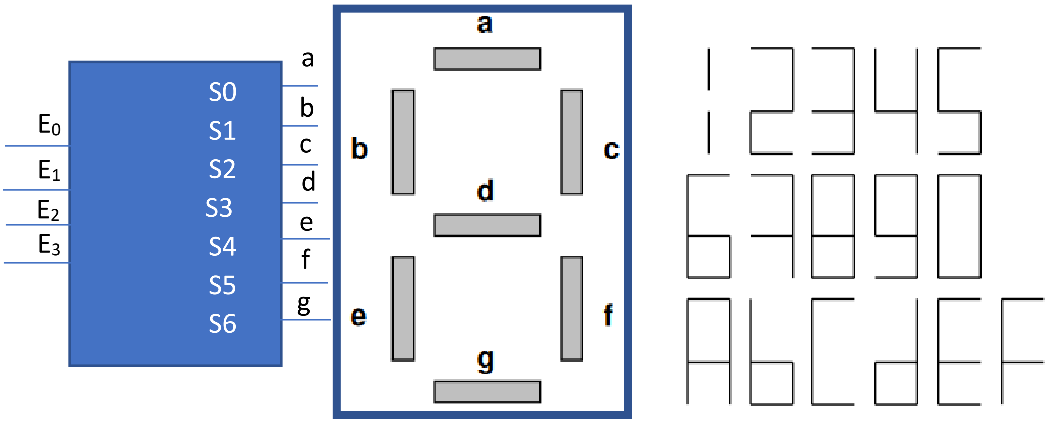


FIGURE 3.2 – Afficheur 7 segments, chiffres