

# NSI : LEÇON 7 , CODAGE DES ENTIERS RELATIFS.

## 1. RÉVISIONS : LE CODAGE HEXADÉCIMAL ET LE CODAGE BINAIRE.

### Exercice 1

Le site <https://htmlcolorcodes.com/fr/> permet d'obtenir les codes couleur en hexadécimal et en RVB d'une couleur de votre choix.

#	<b>4289C0</b>
R	<b>66</b>
G	<b>137</b>
B	<b>192</b>

1. Expliquer la correspondance entre ces deux codes couleurs différentes.

### Exercice 2

On considère les deux entiers a et b ci-dessous codés en binaire sur un octet.

a = 

0	0	1	0	1	0	1	1
---	---	---	---	---	---	---	---

0	0	1	0	1	0	1	1
---	---	---	---	---	---	---	---

b = 

1	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---

1	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---

a =                      =

a + b = 

--	--	--	--	--	--	--	--

b =                      =

1) Déterminer leur écriture décimale et hexadécimale. 2) Déterminer le résultat binaire de leur somme.

## 2. L'ÉCRITURE DES ENTIERS RELATIFS EN BINAIRE.

### a) Codage sur le bit de poids fort.

Pour codes des entiers relatifs sur un octet, on peut utiliser la convention suivante : le bit de poids fort indique le signe ( 0 pour un entier positif, 1 pour un entier négatif ). On code alors les entiers sur les 7 bits restants.

### Exercice 3

On considère les 2 entiers relatifs a et b codés sur un octet signé sur le bit de poids fort.

a =	<table border="1"><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	0	0	1	0	1	0	1	1		<table border="1"><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	0	0	1	0	1	0	1	1
0	0	1	0	1	0	1	1												
0	0	1	0	1	0	1	1												
b =	<table border="1"><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	1	0	1	1	1	1	1	0		<table border="1"><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	1	0	1	1	1	1	1	0
1	0	1	1	1	1	1	0												
1	0	1	1	1	1	1	0												
a =		a + b =	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																
b =																			

1) Déterminer la valeur décimale de a et b. 2) Déterminer le résultat en binaire de leur somme.

Cette méthode possède plusieurs désavantages :

- le 0 peut s'écrire de deux façons différentes :
- il est nécessaire pour les opérations élémentaires comme l'addition de mettre en place deux algorithmes de calculs différents suivant le signe des entiers.

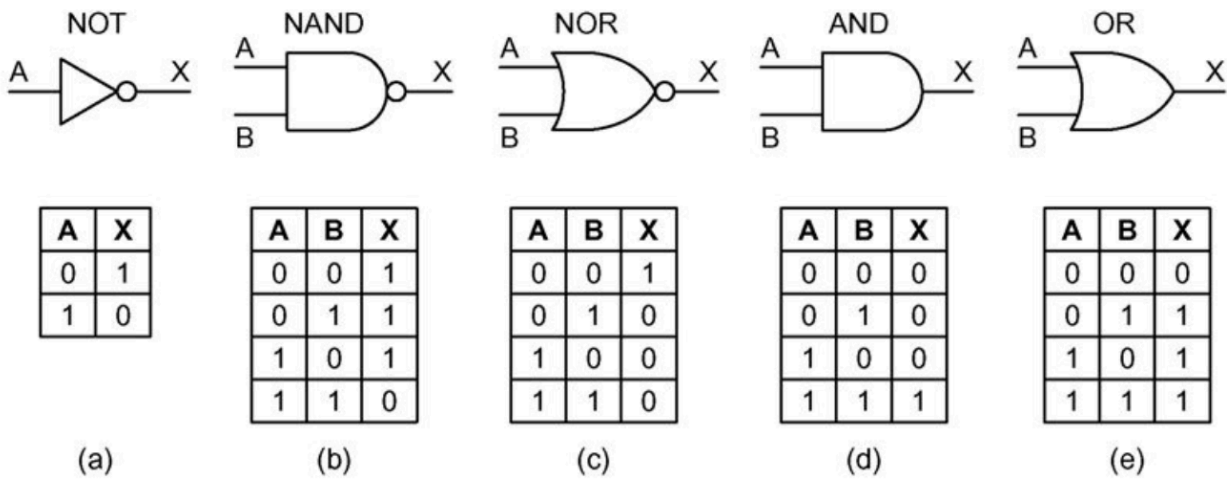
### b) Codage suivant la méthode du complément à 2.

#### 1. Un détour par les portes logiques et l'architecture d'un ordinateur.

Au niveau de l'unité arithmétique-logique du processeur d'un ordinateur sont implémentées différents circuits électroniques permettant d'opérer sur les nombres binaires.

On retrouve en particulier des circuits permettant d'opérer des additions et des multiplications.

On trouve également des circuits permettant d'opérer certaines opérations logiques :



Ces circuits correspondent aux opérateurs booléens : NON, NON(ET),NON(OU),ET, et OU.

On retrouve ces opérations aux niveaux des tests logiques implémentés dans les langages de programmation :

- le AND correspond au and en python et au && en javascript,
- le OR correspond au or en python et au || en javascript,
- le NOT correspond au not ou ! en python ou en javascript.

## 2. La méthode du complément à 2 pour écrire des entiers relatifs en binaire.

On désire écrire le nombre - 12 en binaire sur un octet selon la méthode du complément à 2.

- étape 1 : On écrit le nombre en binaire sur un octet :
- étape 2 : on effectue un NOT sur ce nombre ( on parle aussi de complément à 1 ) :
- étape 3 : on ajoute 1 au résultat obtenu :
- étape 4 : on peut vérifier que le résultat correspond bien à l'opposé de 12 en effectuant la somme en binaire de 12 et -12.

12 = 

0	0	0	0	1	1	0	0
---	---	---	---	---	---	---	---

12 = 

0	0	0	0	1	1	0	0
---	---	---	---	---	---	---	---

Not(12)= 

--	--	--	--	--	--	--	--

-12 = 

--	--	--	--	--	--	--	--

On ajoute 1 = 

--	--	--	--	--	--	--	--

12+(-12) = 

--	--	--	--	--	--	--	--

-12 = 

--	--	--	--	--	--	--	--

## Exercice 4

1) Ecrire selon la méthode du complément à 2 les entiers suivants : 27 , -27, -120.

2) On veut écrire le nombre -500 en binaire selon la méthode du complément à 2. De combien d'octets aura-t-on besoin ? Donner l'écriture recherchée.

3) Combien d'entiers relatifs différents peut-on écrire sur un octet selon la méthode du complément à 2.

### 3. Déterminer facilement l'écriture décimale d'un entier relatif codé en complément à 2.

On considère le nombre relatif suivant, codé sur un octet selon la méthode du complément à 2 : 11010010. Pour déterminer sa valeur décimale, il suffit de déterminer son complément à 2 et de ne pas oublier qu'il est négatif.

a =	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
Not(a)=	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
On ajoute 1 =	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
-a =	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	donc a =

### Exercice 5

Déterminer l'écriture décimale des nombres binaires suivants, codés suivant la méthode du complément à 2 : 10011111, 00011001

### 4. Un truc permettant de calculer l'opposé d'un nombre positif selon la méthode du complément à 2.

On trouve la méthode suivante dans la page Wikipédia :

[https://fr.wikipedia.org/wiki/Compl%C3%A9ment\\_%C3%A0\\_deux](https://fr.wikipedia.org/wiki/Compl%C3%A9ment_%C3%A0_deux)

- Prenons par exemple le nombre 20 : 00010100.
- On garde la partie à droite telle quelle : (00010**100**).
- On inverse la partie de gauche après le premier un : **1110**1100.
- Et voici -20 : 11101100.

1) Essayer de comprendre pourquoi cette méthode fonctionne,

2) Peut-on vraiment utiliser le terme inverse au point 3 de cette méthode ? Quel terme serait souhaitable d'utiliser ?