Le Binaire

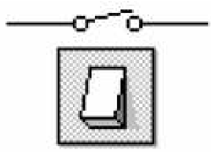
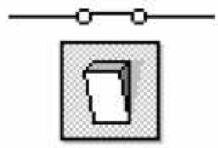
I) Le codage binaire

1) Présentation du binaire

L'informatique utilise des courants électriques, des aimantations, des rayons lumineux...
Chacun de ces phénomènes met en jeu deux états possibles :
- tension nulle (0 V) ou tension non nulle (5 V par exemple),
- aimantation dans un sens ou dans l'autre sens,
- lumière ou pas de lumière...

Il suffit de deux chiffres pour traduire ces deux états : c'est la numération binaire qui utilise les chiffres 0 et 1.
Un rayon de lumière peut parfaitement traduire ces deux valeurs :
- 1 = lumière
- 0 = pas de lumière

Question 1 :
Sur votre compte-rendu, indiquer les réponses pour les positions a et b des interrupteurs.

	Position a : Interrupteur ..... Le courant		Position b : Interrupteur ..... Le courant	
Attribution à cet état	de la valeur		de la valeur	

2) Définition de l'élément binaire (BIT)

L'état de cette grandeur binaire est codé sur un élément binaire (en anglais : BInary digiT, ou BIT), c'est-à-dire avec 0 ou 1, ce qui donne deux possibilités.
Reprenons l'exemple des interrupteurs électriques, qui peuvent être ouverts (ce qui correspond à l'élément binaire 0), ou fermés (ce qui correspond à l'élément binaire 1).
S'il n'y a qu'un interrupteur, on peut donc avoir 2 positions possibles : ouvert (0) ou fermé (1).
Si nous disposons de deux interrupteurs en dérivation, on peut avoir 4 combinaisons de positions possibles :

Interrupteur n° 2		Interrupteur n° 1	
Ouvert	0	Ouvert	0
Ouvert	0	Fermé	1
Fermé	1	Ouvert	0
Fermé	1	Fermé	1

Nous disposons donc des 4 combinaisons suivantes : 00 01 10 11
De même, si nous disposons de 3 interrupteurs en dérivation, on peut avoir 8 combinaisons de positions possibles :

000 001 010 011 100 101 110 111

On peut raisonner directement avec les bits : plus le nombre de bits formant le nombre binaire est important, plus le nombre de combinaisons différentes est important.

Question 2 :
Écrire les combinaisons différentes qu'on peut réaliser avec un nombre binaire constitué de quatre bits, ce qui revient à utiliser 4 interrupteurs en dérivation (exemple de nombre binaire : 0101). Combien a-t-on de combinaisons différentes en tout ?

Question 3 :

Combien y a-t-il de combinaisons différentes pour écrire un nombre binaire constitué de huit bits, ce qui revient à utiliser 8 interrupteurs en dérivation (exemple de nombre binaire : 0111 1001) ? (On ne demande surtout pas d'écrire toutes les combinaisons possibles)

Conclusion : en utilisant n éléments binaires, on peut former 2^n nombres différents et le plus grand d'entre eux est égal à $2^n - 1$.

3) Application aux capacités de stockage ou de communication en informatique

Pour évaluer les capacités de stockage de données, on utilise l'octet comme unité, et surtout ses multiples. L'octet est un nombre binaire formé de 8 éléments binaires.

Remarque : histoire de faire de jolies confusions, l'octet s'appelle « byte » en anglais…

Question 4 :

Compléter sur le compte-rendu les valeurs suivantes, en se servant éventuellement de la calculatrice.

Le kilo-octet (Ko) : 1 Ko $= 2^{10}$ octets = ………… octets
Le méga-octet (Mo) : 1 Mo$= 2^{20}$ octets = ………… octets
Le giga-octet (Go) : 1 Go$= 2^{30}$ octets = ………… octets

Question 5 :

En se servant de l'une des questions précédentes, rappeler combien il y a de combinaisons différentes pour écrire un octet.

Rappel historique : le premier ordinateur familial (ZX Sinclair) avait 1 Ko de RAM alors que les ordinateurs actuels ont environ ………………

Les disquettes 5 $\frac{1}{4}$ pouces stockaient 360 Ko puis 720 Ko. Les disquettes 3 $\frac{1}{2}$ pouces stockent 720 Ko puis 1,44 Mo Les disquettes ZIP stockent 100 Mo puis 250 Mo, tandis que les bandes atteignent 50 Go et les CD-Rom 650 Mo et les disques durs environ ………………o.

4) La notation hexadécimale

En informatique, les données sont également codées en hexadécimal utilisant les 10 chiffres habituels auxquels on ajoute les 6 premières lettres de l'alphabet en majuscules. Soit 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14), F (15). La base utilisée est 16. la méthode est similaire à celle utilisée en décimal, en remplaçant 10 par 16.

Exemple : $\mathrm{FAC} = \mathrm{F} \times 16^2 + \mathrm{A} \times 16^1 + \mathrm{C} \times 16^0 = 15 \times 16^2 + 10 \times 16^1 + 12 \times 16^0 = 4012$.

Voici pourquoi l'hexadécimal a été choisi : un processeur n'utilise seulement que des 0 et des 1 (binaire). Mais c'est lourd à traduire : 1000 en hexadécimal s'écrit 1000000000000 en binaire.

On a choisi une écriture plus condensée : il s'agit de grouper les quartets (donc des blocs de 4 symboles consécutifs) du code binaire 4 par 4.

Exemple : le décimal 125 est en fait codé en mémoire par l'octet 01111101.

Question 6 :

Vérifier cette valeur.

Séparons-le en 2 quartets : 0111 1101. Chaque quartet binaire représente un nombre décimal : 7 et 13 (souvenez-vous du Bibinaire), ce qui donne 7D en hexadécimal.

Application pratique : la couleur « vert d'eau » sur un moniteur est codée en hexadécimal 82DAB7, soit 8-2-13-10-11-7 (tableau Bibinaire), c'est à dire 1000-0010-1101-1010-1011-0111, donc en binaire 100000101101101010110111 en mémoire, ce qui correspond au décimal 8 575 671.

Question 7 :

Le noir est codé FFFFFF, ce qui correspond à quel binaire ? à quel décimal ? Justifier l'appellation 16 millions de couleurs pour nos « micros » actuels.

Voici des informations, telles que l'on pourrait les voir à l'intérieur de la mémoire d'un ordinateur :

001111000010101000100011011000110110010101100011
0110100100100000100101000011001010111001101110100
001010010010011001110101011011100011111100101011
011011010110010101110011011100110110000101100111
0110010100100000100101101011100110111010101100010
011011000110100101101101011010010110111001100001
0110110000100011001010101000111110

Quelle signification accorder aux informations enregistrées ? La réponse dépend du codage qui a été employé pour les enregistrer.

Question 8 :

Exercice de codage :

Décoder le « message » ci-dessus en considérant successivement :
– que des entiers naturels sont codés sur 8 bits,
– que des entiers naturels sont codés sur 16 bits.
On se limitera à la première ligne du « message ».

Éléments de réponse : on suppose que sont codés des entiers naturels sur 8 bits. Pour décoder, il suffit de regrouper les bits 8 par 8, et de calculer pour chacun des octets ainsi obtenus, le nombre qui y est codé : le premier octet 00111100 code l'entier 60 . . .

On suppose que les entiers naturels sont codés sur 16 bits. Le décodage est le même, mais en regroupant les bits 16 par 16. Que donne alors le décodage de la première ligne du « message » ?

5) Poids des bits

Rappel : Qu'est-ce qu'une base ? C'est le nombre total de chiffres (ou de caractères) qui servent à écrire un nombre.

Exemples :
– En base 2, on utilise les 2 chiffres 0 et 1 ; c'est le système binaire.
– En base 10, on utilise les 10 chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9 ; c'est le système décimal.

Rappel : Qu'est-ce que le « poids » d'un chiffre ? C'est la valeur associée à la position de ce chiffre dans un nombre.

Par exemple, on peut écrire 4 138 comme $(4 \times 1\,000) + (1 \times 100) + (3 \times 10) + (8 \times 1)$

Donc 4 138 peut s'écrire : $(4 \times 10^3) + (1 \times 10^2) + (3 \times 10^1) + (8 \times 10^0)$

Le chiffre 4 a un « poids » dix fois plus grand (il vaut dix fois plus) que s'il était en rang 2, cent fois plus que s'il était en rang 1 et 1 000 fois plus que s'il était en rang 0. On appelle le chiffre dont le « poids » est le plus élevé le « bit de poids fort » (en anglais « Most Significant Bit » ou MSB). Ici, c'est le chiffre 4. Ce chiffre est toujours placé le plus à gauche du nombre.

On appelle le chiffre dont le « poids » est le plus faible le « bit de poids faible » (en anglais « Least Significant Bit » ou LSB). Ici, c'est le chiffre 8. Ce chiffre est toujours placé le plus à droite du nombre.

De même, chaque élément binaire n'a pas le même poids, c'est-à-dire la même importance en terme de valeur. On distingue notamment l'élément binaire de poids le plus fort (ou MSB) et l'élément binaire de poids le plus faible (ou LSB) : voir figure 1.

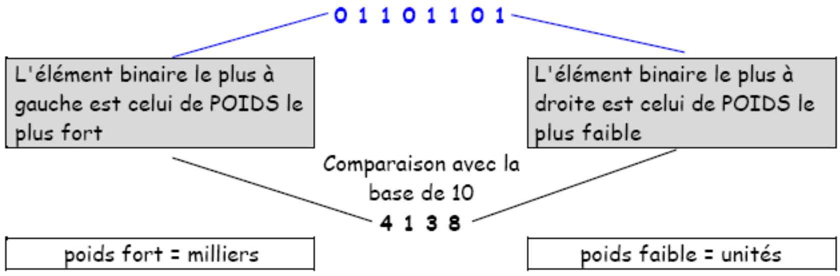


FIG. 1 – Poids le plus fort et poids le plus faible

6) Principe de la conversion d'un nombre binaire en nombre décimal

Un nombre binaire est formé d'une succession de 1 et de 0.

Exemple : Un octet est formé de huit éléments binaires, comme le nombre 1101 1010.

Pour éviter toute confusion, on indique la base d'un nombre de la manière suivante : on écrit le nombre entre parenthèses et la base en indice.

Exemple : l'octet de l'exemple précédente sera écrit dans le système binaire $(1101\ 1010)_2$.

Comme on travaille en base 2 pour le binaire, le poids d'un bit sera proportionnel à une puissance de 2.

Pour un dispositif à n bits, le poids de chaque bit est croissant de 2^0 à 2^n quand on lit les chiffres de droite à gauche.

Exemple : le nombre binaire 1101 1010 représente

$$(1 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$$

On pourrait en effet tracer le tableau ci-dessous :

Chiffre	1	1	0	1	1	0	1	0
Rang de l'élément binaire	7	6	5	4	3	2	1	0
Poids de l'élément binaire	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Valeur en base 10	$1 \times 2^7 = 128$	$1 \times 2^6 = 64$	$0 \times 2^5 = 0$	$1 \times 2^4 = 16$	$1 \times 2^3 = 8$	$0 \times 2^2 = 0$	$1 \times 2^1 = 2$	$0 \times 2^0 = 0$

On en déduit la valeur décimale de ce nombre :

$$128 + 64 + 0 + 16 + 8 + 0 + 2 + 0 = (218)_{10}$$

On écrit l'égalité des deux nombres dans les bases binaire et décimale de la manière suivante :

$$(1101\ 1010)_2 = (218)_{10}$$

Question 9 :

Convertir en décimal les nombres binaires suivants : $(0110)_2$; $(1001\ 0101)_2$; $(1111\ 1111)_2$

7) Principe de la conversion d'un nombre décimal en nombre binaire

Méthode :

– Diviser le nombre décimal à convertir par 2 ; le quotient obtenu doit être ENTIER.
– Conserver le reste ENTIER de la division.

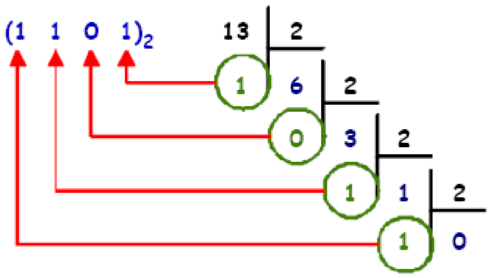


FIG. 2 – Méthode des divisions successives

- Le quotient obtenu est à nouveau divisé par 2 ; le nouveau quotient doit être ENTIER.
- Conserver le reste ENTIER de la division.
- Répéter l'opération sur chaque quotient obtenu jusqu'à ce que le quotient de la division soit égal à 0.

Le résultat en binaire est écrit en notant les restes successifs, en commençant par le dernier, *de la gauche vers la droite*. Cette méthode est dite « Méthode des divisions successives ».

Exemple : Convertir le nombre décimal : $(13)_{10}$ en nombre binaire : voir figure 2. Le premier reste donne le chiffre de rang le plus bas (LSB). Il faut continuer à diviser par deux jusqu'à ce que le quotient vaille 0. On obtient ici : $(1101)_2$.

Vérification :

Chiffre binaire	1	1	0	1
Rang de l'élément binaire	3	2	1	0
Poids de l'élément binaire	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Valeur en base 10	$1 \times 2^3 = 8$	$1 \times 2^2 = 4$	$0 \times 2^1 = 0$	$1 \times 2^0 = 1$

On obtient bien en système décimal : $8 + 4 + 0 + 1 = 13$.

Question 10 :

Convertir en binaire les nombres décimaux suivants :

$$(37)_{10}\ ;\ (51)_{10}\ ;\ (189)_{10}\ ;\ (205)_{10}\ ;\ (2313)_{10}$$

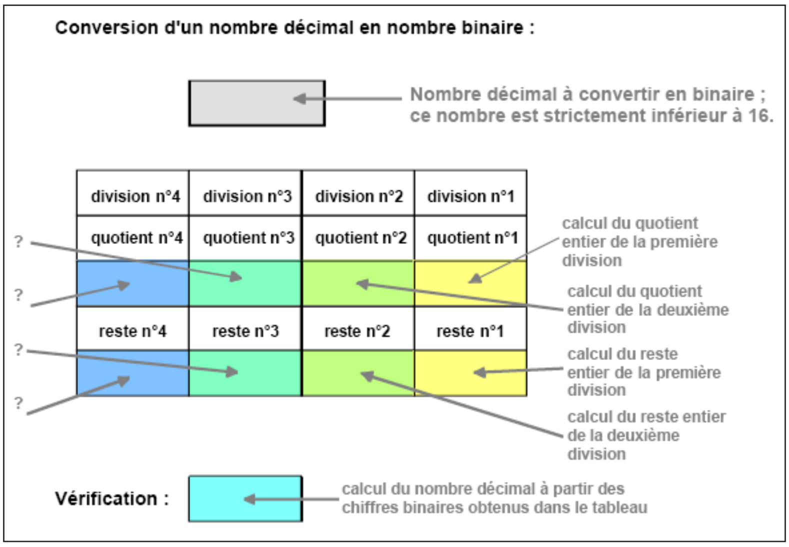
8) Utilisation d'un tableur pour les deux conversions

But : Réaliser une conversion automatique d'un nombre décimal en nombre binaire et vérifier le résultat par deux méthodes ; la première est de reconvertir le nombre binaire en nombre décimal « à la main » et la seconde est d'utiliser les fonctions de calcul du tableur directement.

On souhaite obtenir ceci dans la feuille de calcul du tableur (« classeur » d'Open Office) : voir figure 3.

En cliquant sur le bouton  (« Assistant Fonctions ») dans la barre du haut de la feuille de classeur d'Open Office, vous disposez des fonctions suivantes dans la liste des fonctions mathématiques proposées :

- QUOTIENT : QUOTIENT$(x; y)$ donne le quotient entier d'un nombre x divisé par le nombre y.
- MOD : MOD$(x; y)$ donne le reste entier d'un nombre x divisé par le nombre y.



Légende :
- en noir : ce qui doit être écrit dans la feuille de calcul ;
- en gris : ce qui doit être fait.

FIG. 3 – Feuille de calcul du tableur

Les nombres x et y peuvent être remplacés par des cellules (cliquer alors sur la cellule).

Question 11 :
- Faire dans la feuille de calcul une présentation comme dans la figure 3.
- Écrire les formules de calcul.
- Essayer si votre feuille de calcul fonctionne avec tout nombre décimal compris entre 0 et 15.
- Faire vérifier par votre professeur, et en discuter avec lui pour plus d'explications.

II) Opérations dans les systèmes décimal et binaire

1) Principe d'une addition

Soit deux additions dans le système décimal :

		8
+		5
=	1	3

		3	4
+		8	7
=	1	2	1

Question 12 :
Que se passe-t-il lorsque la somme des nombres placés dans la même colonne dépasse 9 ?

Dans le système binaire, on raisonne de la même manière. Soit les deux additions suivantes :

	0	1	1	1
+	0	1	0	0
=	1	0	1	1

	0	1	0	1
+	0	0	1	1
=	1	0	0	0

Question 13 :
Que se passe-t-il lorsque la somme des nombres placés dans la même colonne dépasse 1 ?

Question 14 :
Quel est le résultat de l'addition suivante : $(1101)_2 + (0100)_2$?

2) Principe d'une multiplication

Soit les multiplications suivantes dans le système décimal :

			7
×		2	5
		3	5
	1	4	
=	1	7	5

				7
×		2	0	5
			3	5
	1	4		
=	1	4	3	5

Question 15 :

– Que se passe-t-il lorsqu'on multiplie 7 par le chiffre 2 des dizaines du nombre 25 ?
– Que se passe-t-il lorsqu'on multiplie 7 par le chiffre 0 des dizaines du nombre 205 ?
– Que se passe-t-il lorsqu'on multiplie 7 par le chiffre 2 des centaines du nombre 205 ?
– Résumer la méthode utilisée pour faire une multiplication.

Question 16 :

Reproduire le tableau suivant dans le compte-rendu et faire la multiplication en binaire.

				1	0	1	0
×				1	0	1	1
=							

III) Exercices

1) Conversion

Convertir les nombres binaires suivants en base 10 :
– $(1001)_2 = (......)_{10}$;
– $(1000\ 1111)_2 = (......)_{10}$.
Convertir les nombres décimaux suivants en base 2 :
– $(52)_{10} = (......)_2$;
– $(125)_{10} = (......)_2$.
Sur combien de bits écrit-on ces deux nombres en base 2 ?

2) Compter

– Écrire les valeurs binaires correspondant à la suite des nombre décimaux 0, 1, 2, . . ., 8.
– Calculer les valeurs des puissances de 2 suivantes : 2^8, 2^9, 2^{10}, 2^{11}, . . ., 2^{16}.

3) Faire des opérations en binaire

– Additionner les nombre $(1101)_2$ et $(0101)_2$.
Doit-on écrire le résultat sur 4 ou sur 5 bits ?
– Multiplier les nombres $(10)_2$ et $(0110)_2$.
Sur combien de bits devez-vous écrire le résultat ?

4) Évaluer

a) Un écran comporte une palette de 256 couleurs. Combien faut-il d'octet pour les représenter ?

b) Grâce à un scanner, on enregistre une photo numérique d'un format 1 024 par 1 024 points.

N.B. : 1024 octets = 1 Ko (ne pas confondre avec k qui signifie 1 000 : 1 kg = 1 000 g).

Quelle quantité d'octets faut-il ? Traduire ce chiffre en mégaoctet (Mo).

Remarque : Pour représenter une photo numérique dont les points peuvent être d'une couleur à choisir parmi 256 couleurs, il faut, pour chaque point de l'image, mémoriser sa couleur : un octet suffit pour un point.

c) Pour représenter un texte écrit (sans information de mise en page), que l'ordinateur considère comme une suite de caractères (alphabétiques, symboles de ponctuation, espaces et passages à la ligne), sur une page, combien faut-il d'octets, sachant qu'une page (dactylographiée) contient environ 2 000 caractères ?

Remarque : Pour représenter un texte écrit il faut autant d'octets que le texte comporte de caractères.

d) À quelle quantité de pages correspond une photo numérisée ?

e) Si l'on numérise, par un scanner couleur, un livre de 1 000 pages, combien d'octets obtient-on ?

IV) Le jeu de Marienbad

Variante du jeu de Nim apparaissant dans le film *L'année dernière à Marienbad* d'Alain Resnais, le jeu de Marienbad se joue à deux de la façon suivante :

- On répartit 16 objets identiques (jetons, allumettes, pièces, ...) en 4 tas contenant respectivement 1, 3, 5 et 7 objets.
- Les deux joueurs décident qui commence et si celui qui ramassera le dernier objet sera le gagnant ou le perdant.
- Chaque joueur, à tour de rôle, ramasse autant d'objets qu'il veut, mais dans un seul des 4 paquets, jusqu'à ce que tous les objets aient été pris.

Il existe une stratégie qui gagne — presque — toujours (le personnage d'Alain Resnais, qui la connaît, affirme : « Je peux perdre, mais je gagne toujours. »). Elle utilise les nombres binaires. Saurez-vous la découvrir ?