

Polynômes Binaires I : Représentation sur machine

Notation hexadécimale.

La numération hexadécimale remplace avantageusement la numération binaire. Quatre chiffres binaires sont regroupés pour former un unique chiffre hexadécimal.

Dcimal	Hexadécimal	Binaire
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

En C, pour indiquer la numération hexadécimale on fait précéder les chiffres par 0x. Par exemple $0x12 = 0001\ 0010_2 = 1 \times 16 + 2 = 18_{10}$.

Dans la plupart des compilateurs, le type `long unsigned int` représente des entiers codes avec 32 bits (chiffres binaires) : $(b_0, b_1, \dots, b_{31})$ représente l'entier $b_0 + 2b_1 + \dots + 2^{31}b_{31}$. Les entiers représentables sont donc compris entre 0 et $2^{32} - 1$. On peut noter avec 32 chiffres binaires ou 8 chiffres hexadécimaux.

On utilise ce type pour représenter un polynôme coefficients dans \mathbb{F}_2 : $(b_0, b_1, \dots, b_{31})$ représente le polynôme $b_0 + b_1X + \dots + b_{31}X^{31}$. Par exemple $0x123 = 000100100011$ représente le polynôme $1 + X + X^5 + X^8$. b_0 est le chiffre binaire de poids faible, c'est-à-dire celui qui est écrit le plus droite (ici 1). Tous les polynômes binaires de degré ≤ 31 sont représentables sur un mot de 32 bits.

Exercice 1.

- (1) Écrire les polynômes qui correspondent $0x12ab$, $0xff$.
- (2) Écrire en notation hexadécimale les polynômes
 - $1 + X + X^7 + X^8 + X^{10}$,
 - $X + X^3 + X^4 + X^9$,

$$- 1 + X + X^2 + X^3 + X^{16}.$$

Pour accéder aux différents chiffres binaires d'un entier, on procède par *décalage* et *masque*. Par exemple `(x >> 7) & 1` décale la variable `x` de 7 positions vers la droite, puis réalise un *et* logique avec la valeur 1. Le résultat est précisément égal au bit b_7 de `x`.

L'expression `(x & 0x80)` a également un résultat non nul si et seulement si le bit b_7 de `x` vaut 1. Le résultat de cette expression est 0 si b_7 est nul et 0x80 si b_7 vaut 1.

Exercice (sur machine) 1.

En utilisant des décalages et des masques, écrire une procédure `void PrintBinary(long unsigned x)` qui affiche la représentation d'un entier en base 2. Attention, il faut afficher le bit de poids fort en premier. Quelle est l'écriture binaire de 121 ?

Exercice (sur machine) 2.

Écrire une procédure `void PrintPol(long unsigned x)` qui affiche le polynôme binaire représenté par un entier donné sous sa forme usuelle. Par exemple `PrintPol(0x123)` devra afficher $1+X+X^5+X^8$. Ici, le parcours des termes va du poids faible vers le poids fort : extraire le bit de poids faible en masquant avec la valeur 1, puis décaler d'un rang vers la droite chaque itération.

Exercice (sur machine) 3.

Le degré d'un polynôme binaire est l'indice le plus élevé d'un bit non nul. Par exemple, le polynôme $1 + X + X^9$, représenté en hexadécimal par `0x203 = 001000000011` est de degré 9, ce qui correspond à l'indice de son premier chiffre égal 1.

Pour trouver le degré d'un polynôme représenté par un entier `x`, on peut décaler `x` d'un rang vers la droite jusqu'à obtenir un résultat nul. Le nombre de décalages effectués correspond alors exactement au degré.

Écrire une fonction `int Degree(long unsigned a)` qui renvoie le degré d'un polynôme binaire représenté par un `long unsigned`.