Université Ibn Khaldoun Tiaret 2018/2019
Faculté de Mathématiques et de l'Informatique
Dept d'Informatique
Module: Structure des Machines 1

TD1

EXO1

Exprimez le nombre décimal 100 dans toutes les bases de 2 à 9 et en base 16.

EXO2

Exprimez les nombres décimaux 94, 141, 163 et 197 en base 2, 8 et 16

EXO3

Donnez la valeur en décimal des nombres positifs suivants:

a. $(4032.023)_5$ b. $(7AE.F)_{16}$ c. $(457703.17)_8$ d. $(8065)_9$

EXO4

$(0,8254)_{10}=(\ ?)_2$ $214,45_{10}=(\ ?)_2$

Ecrire 88 en base 2 et 3

EXO5

a- $(9541)_{10} = (\ ?)_{DCB}$ $(\ 849529)_{10}= (\ ?)_{DCB}$

b- $(100110000100)_{DCB}=(\ ?)_{10}$ $(111100100010110)_{DCB}=(\ ?)_{10}$ $(110110010001)_{DCB}=(\ ?)_{10}$

c- $(100110000100)_{DCB}=(\ ?)_2=(\ ?)_8=(\ ?)_H$ $(111100100010110)_{DCB}=(\ ?)_8 =(\ ?)_2=(\ ?)_H$

EXO6

Donnez sur 8 bits les représentations « signe et valeur absolue » et complément à 2 des nombres décimaux 45, 73, 84, –99, –102 et –118.

EXO7

1) Effectuez la soustraction 122 – 43 dans la représentation en complément à 2 en n'utilisant que l'addition.

2) Multipliez les nombres binaires 10111 et 1011 ; vérifiez le résultat en décimal.

EXO8

On cherche à déterminer les cas de débordement lors d'une addition signée. On dit qu'il y a débordement lorsque le résultat obtenu (limité à la taille autorisée) n'est pas correct.

1) Effectuez en binaire sur 8 bits, en représentation en complément à 2, les opérations suivantes :

100 + 100, (–1) + (–2), (–1) + 16, (–100) + (–100). Dans quel(s) cas y a-t-il débordement ?

2) On additionne 8 bits $A = a_7 \ldots a_0$ avec les 8 bits $B = b_7 \ldots b_0$, en obtenant un résultat $S = s_7 \ldots s_0$ et une retenue r. Si A et B sont de signes contraires, peut-il y avoir débordement ? Si A et B sont de même signe (distinguez), quelles valeurs peuvent prendre s_7 et r ? Dans quel(s) cas y a-t-il débordement ? On s'intéresse maintenant aux deux retenues de poids fort : r et r_6 (qui provient de l'addition de a_6, b_6 et r_5, et s'ajoute à a_7 et b_7 pour donner s_7 et r). En reprenant la question précédente, donnez l'expression du débordement en fonction de r_6 et r.

EXO9

On considère une représentation simplifiée des réels en virgule flottante. Un nombre réel X est représenté par 10 bits *s eeee mmmmm* où $X = (-1)^s \times 1, m \times 2^{e-7}$ avec un exposant sur 4 bits ($0 < e \leq 15$, un exposant égal à zéro désigne le nombre zéro quelle que soit la mantisse) et une pseudo-mantisse sur 5 bits (représentant les puissances négatives de 2).

1) Quels sont le plus grand nombre et le plus petit nombre strictement positifs représentables ?

2) Comment se représente le nombre 1 ? La précision ε d'une représentation est l'écart entre 1 et le nombre représentable suivant. Combien vaut ε pour cette représentation ?

3) Peut-on représenter 7,2 et 57,6 ? Quels sont les nombres qui encadrent 7,2 et 57,6 dans cette représentation ?

4) Que donne en décimal la multiplication de 7,25 et 28,5 ? Écrivez 7,25 et 28,5 dans la représentation proposée et effectuez la multiplication. Quels sont les deux arrondis possibles pour le résultat et leur valeur décimale ?

EXO10

Il s'agit maintenant d'imiter ce que l'on appelle la notation scientifique. On va écrire le nombre voulu sous la forme $\pm 1,M \times 2^E$, où 1,M s'appelle la mantisse du nombre et E l'exposant. Comme la mantisse commence toujours par une partie entière égale à 1, on ne l'écrit pas et on n'exprime que la partie fractionnaire, M, appelée « pseudo-mantisse ». Puisqu'il y a plusieurs manières d'écrire les différents champs, une normalisation a été adoptée qui définit deux principaux types de nombres en virgule flottante : la simple précision sur 32 bits et la double précision sur 64 bits. Les nombres en simple précision possèdent une pseudo-mantisse sur 23 bits (correspondant aux puissances négatives de 2, de 2^{-1} à 2^{-23}), un exposant sur 8 bits et un bit de signe. Les nombres en double précision ont une pseudo-mantisse sur 52 bits, un exposant sur 11 bits et un bit de signe

La pseudo-mantisse est la partie fractionnaire du nombre et il suffit d'additionner les puissances négatives de 2 pour calculer le résultat. Afin que l'on puisse exprimer les exposants négatifs, la valeur binaire dans le champ exposant est la valeur réelle décalée par excès d'un biais de 127 (1 023 pour les nombres en double précision). Il faut donc retrancher ce biais de la valeur indiquée pour obtenir l'exposant réel. Le bit de poids fort est un bit de signe, par convention à 1 quand le nombre est négatif.

Représenter les nombres suivants en virgule flottante avec la norme IEEE 754 sur 32 bits

500 ; -3,5 et 0,40625

EXO11.

Compléter les tables :

Addition en octal

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
. . .								
7								

Multiplication en octal

*	0	1	2	3	4	5	6	7
0								
1								
. . .								
7	0							61

EXO12.

Effectuer les opérations suivantes

A	b	a + b	a — b	a * b
$(11011)_2$ $(45)_8$ $(A2A)_{16}$	$(1101)_2$ $(32)_8$ $(92)_{16}$			

EXO13 Effectuer les opérations suivantes (base2 et base 16) en 8 bits non signés et signés

01000111	1001101	0A	C8
-	-	-	-
00011001	0000101	17	5B

EXO14

Donnez la valeur des nombres suivants exprimés en code de Gray

a) 10010 b) 011 c) 1001 d) 00001 e) 11111 f) 10

EXO15

Donnez le résultat du bit de la parité des mots suivants pour travailler avec la parité paire:

a) 11101001 b) 10011010 c) 10101100 d) 01100100 e) 10000000 f) 11111111111111

g) 11110010 h) 01000001 i) 110011 j) 1100001

EXO16

a- Ecrire un programme en langage C pour passer de l'écriture suivant les puissances décroissantes à celle suivant les puissances croissantes ? Par exemple on veut passer du tableau ad[L] contenant 10110 (avec ici L = 5) au tableau ac[L] contenant 01101, avec les indices des cases de 0 à L – 1.

b- Comment faire la même chose, mais avec un seul tableau a[L], contenant au départ l'écriture suivant les puissances décroissances, et à la fin suivant les puissances croissantes ? Il s'agit de mettre à l'envers le tableau a[L] contenant par exemple 10110 pour L = 5.

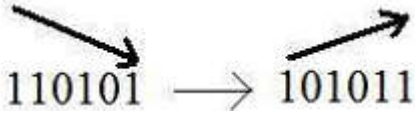
EXO17

A-Conversion d'un nombre (en binaire) en nombre (en base 10) par programmation

On sait déjà faire cela manuellement, par exemple ce nombre écrit suivant ses puissances
décroissantes :

$110101 = 2^5 + 2^4 + 2^2 + 2^0 = 32 + 16 + 4 + 1 = 53$

On peut faire de même par programme. On commence par placer le nombre binaire dans un tableau a[L] avec L donné. Par souci de simplicité (mais ce n'est pas obligatoire) réécrivons le nombre suivant ses puissances croissantes, en le mettant à l'envers, toujours dans le même tableau,



par exemple

Enfin on parcourt le tableau a[L] en additionnant les termes successifs, tous de la forme a[i] * 2i, avec i de 0 à L – 1.

Donner ce programme en langage C

B-Passage d'un nombre en base 10 au nombre en binaire par programmation

a- Recherche répétée de la plus forte puissance de 2 inférieure ou égale au nombre

Prenons par exemple le nombre 13. La plus forte puissance de 2 qui est dans 13 est 8 = 2^3.

$13 = 2^6 + 5$. Puis on recommence avec 5 : $5 = 2^2 + 1$. On recommence avec 1

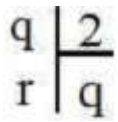
$1 = 2^0$. C'est fini.

Finalement $13 = 2^3 + 2^2 + 2^0 = 1101$ en binaire décroissant

Donner le programme en C qui fait ce travail

b- Deuxième méthode, par divisions successives par 2

Reprenons le nombre 13 qui s'écrit 1101 en binaire décroissant. Comment passer de 13 à 1101 ? On commence par diviser 13 par 2, ce qui donne comme quotient 6 et comme reste 1. Ce reste est le dernier chiffre de l'écriture en binaire. Puis on recommence avec 6 et ainsi de suite On s'arrête lorsque le quotient est 0. La lecture des restes successifs donne 1011, il s'agit de l'écriture en binaire mais suivant les puissances croissantes de 2, ce qui s'écrit 1011. En le lisant à l'envers, on a l'écriture 1101 suivant les puissances décroissantes. Chaque division est de la forme



avec le nouveau quotient q obtenu à partir de l'ancien quotient q, sauf au départ, où l'on part du nombre 13. Pour des besoins d'unification on posera en conditions initiales q = 13 (en général q = N).

Déduire le programme, où l'on place les restes successifs dans un tableau de façon à pouvoir le lire à partir de la fin (suivant les puissances décroissantes) .