Université Paris 7- Denis Diderot
L1 d'Informatique — Examen du 8 janvier 2013

Principes de fonctionnement des machines binaires

Durée : 3h. Les documents et les appareils électroniques de toutes sortes ne sont pas autorisés.

Toutes vos réponses doivent être justifiées. Les exercices sont indépendants, et peuvent être traités dans un ordre quelconque. Il est demandé de traiter tous les exercices de la partie obligatoire, et deux exercices au choix parmi les exercices de la partie avec choix.

Partie obligatoire

Exercice 1 :

Question 1 : Soit le nombre dont la représentation en base 10 est 5137. Donner ses représentations dans les bases 2, 4, 8 et 16. Parmi les types entiers de Java, lesquels permettent de le stocker sans perte ?

$5137_{10} = 4096 + 1024 + 16 + 1 = 1010000010001_2$

$\overline{1}\ \overline{01}\ \overline{00}\ \overline{00}\ \overline{01}\ \overline{00}\ \overline{01}_2 = 1100101_4$

$\overline{1}\ \overline{010}\ \overline{000}\ \overline{010}\ \overline{001}_2 = 12021_8$

$\overline{1}\ \overline{0100}\ \overline{0001}\ \overline{0001}_2 = 1411_{16}$

La représentation de 5137_{10} en base 2 prend 13 chi□res : il faut donc 14 bits pour le représenter sous forme d'entier signé. Il est donc possible de représenter 5137_{10} sans perte sous forme de `short`, de `int` et de `long`.

Question 2 : On considère maintenant `-5137`. Quel est le plus petit type Java permettant sa représentation sans perte ? Donner cette représentation. Comment peut-on ensuite obtenir la représentation de `-5137` dans les types Java plus grands ?

De même, il est possible de représenter -5137_{10} sans perte sous forme de `short`, de `int` et de `long` : le plus petit tel type est donc `short`.

La représentation de 5137_{10} en tant que `short` est `0001010000010001`, donc celle de -5137_{10} est `1110101111101111`. Puis, pour représenter -5137_{10} sous forme de `int` et de `long`, il su□t de rajouter des 1 comme bits de poids fort (on en rajoute 16 pour faire un `int` et 48 pour faire un `long`).

Question 3 : Soit le nombre dont la représentation en base 10 est `27837`. Parmi les trois propositions suivantes, retrouver quel est son codage en tant que `short` Java, en expliquant pourquoi il n'est pas nécessaire de mener tous les calculs.

1. `1010100110011101`
2. `0110110010111101`

3. `0110101100110110`

27837 est un entier positif impair, donc son bit de poids fort est un `0` et sont bit de poids faible est un `1`. Ainsi, seule la proposition n°2 convient.

Question 4 : Soit x le nombre dont la représentation hexadécimale est `1AE2B67F1C0`. Donner une valeur approchée de x sous forme d'une puissance de 2 puis d'une puissance de 10 (rappel : $2^{10} \approx 10^3$). Quelle est la plus grande puissance de 2 dont x est le multiple ?

La représentation binaire de x est `1101011100010101101100111111100011100000`, donc on remarque que $3 \times 2^{39} < x < 2^{41}$. On peut donc considérer x comme approximativement égal à 2^{41}, ou encore à 10^{12}.

En outre, l'écriture de x en base 2 se termine par six 0, donc x est un multiple de 2^6 mais pas de 2^7.

Exercice 2 :

Établir les tables d'addition et de multiplication de la base 5. E□ectuer dans cette base l'opération $(321)_5 \times (4440)_5$.

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	10
2	2	3	4	10	11
3	3	4	10	11	12
4	4	10	11	12	13

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	11	13
3	0	3	11	14	22
4	0	4	13	22	31

				3	2	1
×			4	4	4	0
		2	3	3	4	
	2	3	3	4		
2	3	3	4			
3	2	0	1	2	4	0

Exercice 3 :

Soit n le nombre rationnel dont la représentation en base 10 est $n = -65432$.

Question 1 : Donner la représentation binaire de l'entier n.

On remarque que $2^{16} = 65536$ et donc que $n + 2^{16} = 104 = 64 + 32 + 8$. Ainsi, $n = -(1111111110011000)_2$.

Question 2 : Donner sa représentation en machine, comme valeur de type `int`.

On déduit immédiatement de la question précédente que n est représenté par l'`int` `11111111111111110000000001101000`.

Question 3 : Donner sa représentation en machine, comme valeur de type `float`, et exprimer celle-ci sous forme hexadécimale.

Puisque $n = (-1) \times 2^{15} \times (1,111111110011)_2$, il est représenté par le `float` $11000111011111111001100000000000_2 = \texttt{C77F9800}_{16}$.

Soit r le nombre rationnel dont la représentation en base 10 est $(77,7(7)^\omega)_{10}$.

Question 4 : Donner sa représentation en base 2.

On remarque que $10r - r = 700$, donc que $r = \frac{700}{9} = 77 + \frac{7}{9}$. Or, $77_{10} = (1001101)_2$ et $\frac{7}{9} = \frac{9+5}{2\times 9} = 2^{-1} + \frac{5}{2\times 9} = 2^{-1} + \frac{9+1}{2^2\times 9} = 2^{-1} + 2^{-2} + \frac{1}{2^2\times 9} = 2^{-1} + 2^{-2} + \frac{9+7}{2^6\times 9} =$

$2^{-1} + 2^{-2} + 2^{-6} + \frac{7}{2^6 \times 9}$, donc $\frac{7}{9} = (0, (110001)^\omega)_2$.
Ainsi, $r = (1001101, (110001)^\omega)_2$.

Question 5 : Donner sa représentation en machine, comme valeur de type `float`.

Puisque $r = 2^6 \times (1, 00110111000111000111000\underline{1}\ldots)_2$ — on a écrit 24 bits après la virgule — il est représenté par le `float` $11000010100110111000111000111001_2 =$ $\texttt{C29B8E39}_{16}$.

Exercice 4 :

Un mot mémoire de 32 bits contient la valeur hexadécimale suivante : `706F7000`.

Question 1 : Quelle est la valeur de chacun des 32 bits ? Le mot contient la valeur binaire $01110000011011110111000000000000_2$.

Question 2 : Si on suppose que dans cette mémoire est codée la représentation machine d'une suite de caractères `ASCII`, quelle est la valeur de cette chaîne ?

Le mot, lu comme suite de caractères ASCII, représente le texte `pop`.

Question 3 : Si on suppose que dans cette mémoire est codée la représentation machine de deux entiers en format `short`, quelles sont les valeurs de ces entiers ?

Le mot, lu comme paire de `short`, représente la paire (28783,28672).

Question 4 : Si on suppose que dans cette mémoire est codée la représentation machine d'un réel en format `float`, donner la valeur de ce réel sous la forme $m \times 2^e$, où m et e sont deux entiers exprimés en base 2. Donner un ordre de grandeur de ce nombre.

Le mot, lu comme `float`, représente le réel $2^{95} \times (1, 10111101111000000000000)_2 = m \times 2^e \approx 2^{96} \approx 7 \times 10^{28}$, où $m = (111011110111)_2$ et $e = (1010100)_2$.

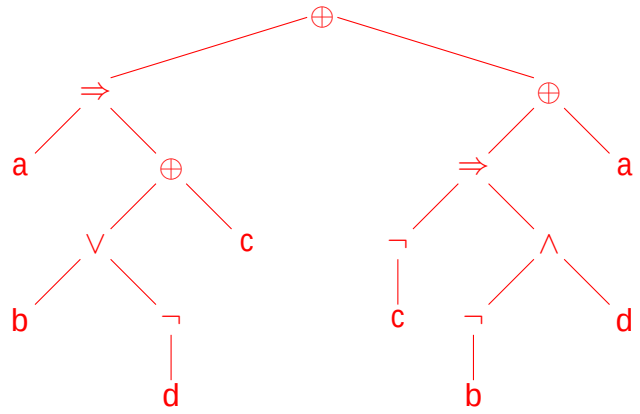
Rappel :

Le standard ASCII code les caractères alphanumériques de 0 à 127. Les caractères sont alignés sur 8 bits. On y trouve notamment :
– 0 : le caractère terminateur de chaînes de caractères
– 45 : le signe moins -
– 46 : le point.
– de 48 à 57 : les chi□res de 0 à 9
– de 65 à 90 : l'alphabet des majuscules (A à Z)
– de 97 à 122 : l'alphabet des minuscules (a à z).

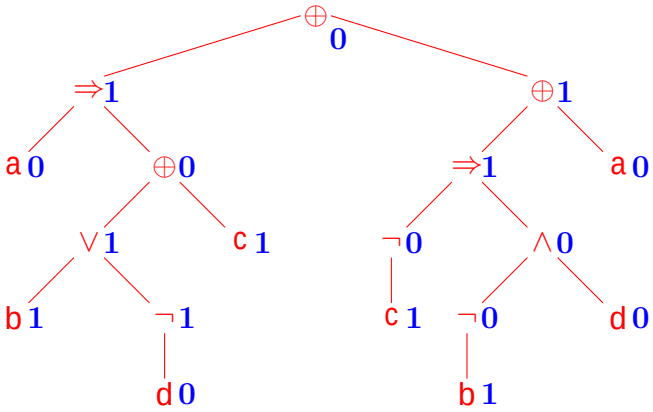
Exercice 5 :

On considère l'expression logique suivante : $(a \Rightarrow ((b \vee \neg d) \oplus c)) \oplus ((\neg c \Rightarrow (\neg b \wedge d)) \oplus a)$. (On rappelle que $\oplus$ est le connecteur du "ou" exclusif. Le connecteur $\Rightarrow$ est aussi noté $\subset$).

Question 1 : Construire l'arbre associé à cette expression. Chaque nœud de l'arbre est la racine d'un sous-arbre qui correspond à une sous-expression de l'expression de départ.



Question 2 : Évaluer en chaque nœud de l'arbre la valeur de la sous-expression lui correspondant, pour $a = d = 0$, $b = c = 1$.



Question 3 : Écrire le mot w obtenu par le parcours gauche préfixe de cet arbre et le mot f obtenu par le parcours gauche postfixe de cet arbre.

$w = \oplus \Rightarrow$a$\oplus\vee$b$\neg$dc$\oplus \Rightarrow \neg$c$\wedge\neg$bda

$f =$abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$

Question 4 : En détaillant les valeurs successives de la pile et les calculs e□ectués utilisant le mot f, donner la valeur de l'expression pour $a = b = 1$ et $c = d = 0$.

On parcourt f et on calcule, au fur et à mesure, la valeur de la pile mémorisée par l'ordinateur.

– on part de la pile vide.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 — on ajoute $a = 1$.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 1 — on ajoute $b = 1$.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 1 0 — on ajoute $d = 0$.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 1 1 — on supprime 0 puis on ajoute $\neg 0 = 1$.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 1 — on supprime 1 1 puis on ajoute $1 \vee 1 = 1$.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 1 0 — on ajoute $c = 0$.
– abd$\neg\vee$c$\oplus\Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 1 — on supprime 1 0 puis on ajoute $1 \oplus 0 = 1$.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 — on supprime 1 1 puis on ajoute $1 \Rightarrow 1 = 1$.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 0 — on ajoute $c = 0$.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 1 — on supprime 0 puis on ajoute $\neg 0 = 1$.
– abd$\neg\vee$c$\oplus \Rightarrow$c$\neg$b$\neg$d$\wedge \Rightarrow$a$\oplus\oplus$: 1 1 1 — on ajoute $b = 1$.

– abd¬∨c⊕ ⇒c¬b¬d∧ ⇒a⊕⊕ : 1 1 0 — on supprime 1 puis on ajoute $\neg 1 = 0$.
– abd¬∨c⊕ ⇒c¬b¬d∧ ⇒a⊕⊕ : 1 1 0 0 — on ajoute $d = 0$.
– abd¬∨c⊕ ⇒c¬b¬d∧⇒a⊕⊕ : 1 1 0 — on supprime 0 0 puis on ajoute $0 \wedge 0 = 0$.
– abd¬∨c⊕ ⇒c¬b¬d∧ ⇒a⊕⊕ : 1 0 — on supprime 1 0 puis on ajoute $1 \Rightarrow 0 = 0$.
– abd¬∨c⊕ ⇒c¬b¬d∧ ⇒a⊕⊕ : 1 0 1 — on ajoute $a = 1$.
– abd¬∨c⊕ ⇒c¬b¬d∧ ⇒a⊕⊕ : 1 1 — on supprime 1 0 puis on ajoute $0 \oplus 1 = 1$.
– abd¬∨c⊕ ⇒c¬b¬d∧ ⇒a⊕⊕ : 0 — on supprime 1 1 puis on ajoute $1 \oplus 1 = 0$.

Ainsi, notre expression logique vaut 0 quand $a = b = 1$ et $c = d = 0$.

Question 5 : Dresser la table de vérité de l'expression.

La table de vérité de notre expression E est :

a	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
b	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
c	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
d	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
E	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0

Question 6 : En donner la forme normale disjonctive.

La forme normale disjonctive de notre expression E est : $\overline{a}\overline{b}\overline{c}\overline{d} + \overline{a}b\overline{c}\overline{d} + \overline{a}b\overline{c}d + a\overline{b}cd$.

Question 7 : Donner sa forme normale conjonctive.

La forme normale conjonctive de notre expression E est : $(a+b+c+\overline{d})(a+b+\overline{c}+d)(a+b+\overline{c}+\overline{d})(a+\overline{b}+\overline{c}+d)(a+\overline{b}+\overline{c}+\overline{d})(\overline{a}+b+c+d)(\overline{a}+b+c+\overline{d})(\overline{a}+b+\overline{c}+d)(\overline{a}+\overline{b}+c+d)(\overline{a}+\overline{b}+c+\overline{d})(\overline{a}+\overline{b}+\overline{c}+d)(\overline{a}+\overline{b}+\overline{c}+\overline{d})$.

Partie avec choix

Exercice 6 :

On veut scanner une image en RGB (couleurs vraies) à 100 dpi.

Question 1 : Quelle surface (en inch^2) peut-on scanner pour un poids de 3 Mo ?

Chaque inch^2 contient $100^2 = 10^4$ pixels, dont chacun pèse 3 octets. On a droit à 3×10^6 octets. On peut donc scanner une surface de 100 inch^2.

Question 2 : Si l'image a un ratio de $4/3$, donner (pour la surface trouvée à la question précédente) une valeur approchée de sa diagonale (en inch).

L'image a une longueur L et une hauteur H telles que $H \times L = 100$ inch^2 et $\frac{L}{H} = \frac{4}{3}$. Donc $H^2 = \frac{3}{4} \times 100 = 75$ inch^2, $L^2 = \frac{4}{3} \times 100 \approx 133$ inch^2, et la diagonale mesure $D = \sqrt{H^2 + L^2} \approx \sqrt{208} \approx 14,5$ inch.

Question 3 : En faisant varier les données de l'énoncé, donner deux moyens de scanner davantage de surface pour un poids de 3 Mo.

On peut, au choix :
– scanner l'image avec une résolution moindre ;
– scanner l'image en 256 couleurs, en niveaux de gris, ou encore en noir et blanc ;
– utiliser un algorithme de compression d'images.

Exercice 7 :

Soit le programme suivant : `byte x = 1 ;`
`byte y = -128 ;`

Question 1 : Donner la représentation machine de x et y.

x = 00000001_2 et y = 10000000_2.

Question 2 : Donner la représentation machine de x << 3 et y >> 3, puis donner leur valeur.

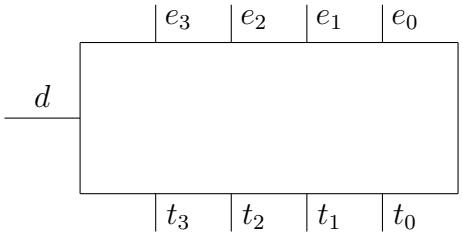
x << 3 = $00001000_2 = 8_{10}$ et y >> 3 = $11110000_2 = -16_{10}$.

Question 3 : Quelle est la valeur de ((y >> 1) >>> 3) | (x << 1) ?

y >> 1 = 11000000_2, (y >> 1) >>> 3 = 00011000_2, x << 1 = 00000010_2 donc ((y >> 1) >>> 3) | (x << 1) = $00011010_2 = 26_{10}$.

Exercice 8 :

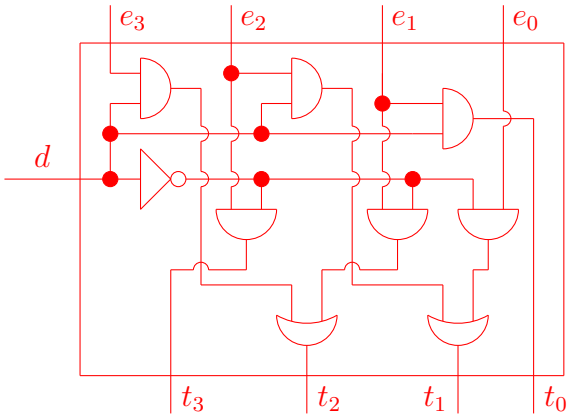
On se propose de construire un circuit qui prend en entrée 4 bits e_0, e_1, e_2 et e_3 ainsi qu'une variable booléenne d, et donne en sortie 4 bits t_0, t_1, t_2 et t_3 qui ont les mêmes valeurs que ceux en entrée, mais avec un décalage (d'un cran) vers la gauche si $d = 0$, ou vers la droite si $d = 1$ (l'une des valeurs est perdue, et on introduit un 0). Le schéma est le suivant :



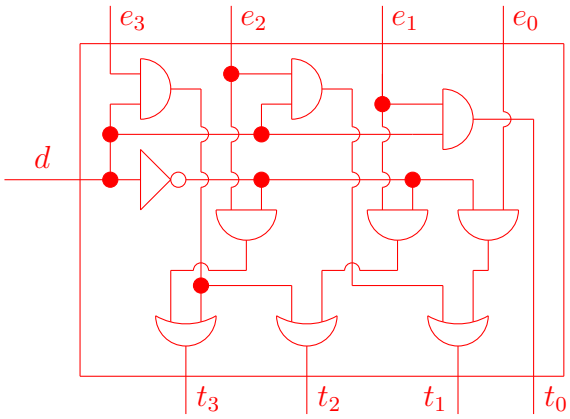
Question 1 : Donner pour tout entier i une fonction logique de t_i.

$t_0 = de_1$, $t_1 = \overline{d}e_0 + de_2$, $t_2 = \overline{d}e_1 + e_3$ et $t_3 = \overline{d}e_2$.

Question 2 : Dessiner un circuit C réalisant ces fonctions.



Question 3 : Modifier ce circuit pour que le décalage à droite introduise à gauche la valeur de e_3 (au lieu de 0). Quelle pourrait en être l'utilité ?



Question 4 : Modifier le circuit C en ajoutant une sortie supplémentaire qui vaut 1 si on fait un décalage à gauche et la valeur de e_3 est 1. Quelle pourrait en être l'utilité ?