Université Toulouse III – Paul sabatier
L2 Informatique

Architecture des systèmes Informatiques — TD

Semestre 3

Toulouse, le 17 octobre 2012

Chapitre 1

Numérotation et codage TD optionnel

1.1 Numérotation

1.1.1 Réaliser l'opération suivante en binaire : $(1101011 - 10110) \times 11001$

$$\begin{aligned}(110\ 1011)_2 &= (107)_{10}\\ (1\ 0110)_2 &= (22)_{10}\\ (1\ 1001)_2 &= (22)_{10}\end{aligned}$$

$$\begin{array}{cccccccc} & 1 & 1 & 0 & 1 & 0 & 1 & 1\\ - & & & 1 & 0 & 1 & 1 & 0\\ \hline & 1 & 0 & 1 & 0 & 1 & 0 & 1\end{array} \text{ Ou } \begin{array}{cccccccc} & 1 & 1 & 0 & 1 & 0 & 1 & 1\\ + & 1 & 1 & 1 & 0 & 1 & 0 & 1\\ \hline & 1 & 0 & 1 & 0 & 1 & 0 & 1\end{array}$$

$$(1010101)_2 = (85)_{10}$$

$$\begin{array}{cccccccccccc} & & & & & 1 & 0 & 1 & 0 & 1 & 0 & 1\\ & & & & \times & & 1 & 1 & 0 & 0 & 1 & \\ \hline & & & & & 1 & 0 & 1 & 0 & 1 & 0 & 1\\ & & & 1 & 0 & 1 & 0 & 1 & 0 & 1 & . & . & .\\ & & 1 & 0 & 1 & 0 & 1 & 0 & 1 & . & . & . & .\\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1\end{array}$$

1.1.2 Réaliser les opérations suivantes en hexadécimal : $(389A+7293)-EB2$

$$\begin{array}{ccccc} & 3 & 8 & 9 & \text{A}\\ + & 7 & 2 & 9 & 3\\ \hline & \text{A} & \text{B} & 2 & \text{B}\\ - & \text{E} & \text{B} & 2 & \text{B}\\ \hline & 9 & \text{C} & 7 & \text{B}\end{array}$$

$$\begin{aligned}(389A)_{16} &= (0011\ 1000\ 1001\ 1010)_2\\ (7293)_{16} &= (0111\ 0010\ 1001\ 0011)_2\\ (AB2B)_{16} &= (1010\ 1011\ 0010\ 1101)_2\\ (EB2B)_{16} &= (0000\ 1110\ 1011\ 0010)_2\\ (9C7B)_{16} &= (1001\ 1100\ 0111\ 1011)_2\end{aligned}$$

1.1.3 Effectuer les conversions ci-dessous

1.1.3.1 $(1447.140625)_{10} = (??)_2 = (??)_{16}$

$$\begin{aligned}
1447 \div 16 &= 90\ R = 7 \\
90 \div 16 &= \ 5\ R = A \\
5 \div 16 &= \ 0\ R = 5 \\
(1447)_{10} &= (5A7)_{16} \\
\\
0.140625 \times 16 &= 2.25 \\
0.25 \times 16 &= 4.00 \\
(0.140625)_{10} &= (0.24)_{16} \\
\\
(1447.140625)_{10} = (5A7.24)_{16} &= (0101\ 1010\ 0111.0010\ 0100)_2
\end{aligned}$$

1.1.3.2 $(1111100101.01011)_2 = (??)_{10} = (??)_{16}$

$$\begin{aligned}
(11\ 1110\ 0101.0\ 1011)_2 &= (3E5; 58)_{16} \\
3E5 &= (3 \times 16^2 + 14 * 16 + 5 + 5 \times 16^{-1} + 8 \times 16^{-2} \\
&= (997, 34375)_{10}
\end{aligned}$$

1.2 Codage

1.2.1 Codage de nombres entiers relatifs

On veut coder des entiers relatifs sur 16 chifffres binaires (deux octets).

	Valeur absolue		Valeur relative	
Base 10	**Base 2**	**Base 16**	**Valeur absolue + signe**	**Complément à 2**
35671	10000 010 0101 0111	75B8	Hors intervalle	Hors intervalle
−32768	1000 0000 0000 0000	8000	Hors intervalle	1000 0000 0000
46443	1011 0101 0110 1011	B56B	Hors intervalle	Hors intervalle
−19536	0100 1100 0110 0100	4C64	1100 1100 0110 0100	1011 0011 1001 1100
−19040	0100 1010 0110 0000	4A60	1100 1010 0110 0000	1011 0101 1010 0000

1.2.1.1 Calcul de la valeur absolue en base 2

$$\begin{aligned}
3567 \div 16 &= 229 \text{ et reste } 7\\
2229 \div 16 &= 139 \text{ et reste } 5\\
139 \div 16 &= 8 \text{ et reste B}\\
8 \div 16 &= 0 \text{ et reste } 8
\end{aligned}$$

$$\Rightarrow (35671)_{10} = (8B57)_{16} = (1000\ 1010\ 0101\ 0111)_2$$

R Afin de pouvoir représenter un nombre celui-ci ne doit pas dépasser un certain intervalle :

Entier naturel $[0; 2^{16-1}] = [0; 65535]$

Valeur absolue + signe $[-2^{15-1}; 2^{15-1}] = [-16384; 16384]$

Complément à deux $[-2^{15}; 2^{15}] = [-32768; 32768]$

1.2.2 Convertir un nombre flottant en décimal

0	1000 0001	1110 0000 0000 0000 0000 0000

$$C = E + biais$$

En simple précision biais = 127.

$$\begin{aligned}
S &= 0 \rightarrow \text{ positif}\\
C &= 129\\
E &= C - 127 = 2\\
1.M &= 1.111 \Rightarrow 1.111 \times 2^2 = 111.1 \times 2^0 = (7.5)_{10}
\end{aligned}$$

1.2.3 Convertir un nombre décimal en flottant

$$
\begin{aligned}
(35.5)_{10} &= ? \\
100011.1 &= 10000111 \times 2^5 \\
\text{Nombre positif donc } S &= 0 \\
E &= 5 \\
C &= E + 127 = 132 = 128 + 4 = (10000100)_2 \\
1.M &= 1.00011 \\
M &= 0011
\end{aligned}
$$

0	1000 0001	0100 0011 0000 0000 0000 0000

Chapitre 2

Algèbre de Bool

2.1 Table de vérité des opérateurs classiques

2.1.1 Exercice 1

2.1.1.1 A – Démontrer que les opérateurs NAND et NOR sont des opérateurs complets

$$
\begin{aligned}
\bar{A} &= A|A \\
A.B &= A\bar{\bar{.}}B = \bar{A} \bar{+} \bar{B} = A\bar{/}B = (A|B)|(A|B) \\
A+B &= A \bar{\bar{+}} B = \bar{A} \bar{+} \bar{B} = (A|B)|(B|B)
\end{aligned}
$$

$$\bar{A} = A \bar{+} A = A \downarrow A$$
$$A.B = A\bar{\bar{.}}B = \bar{A} \bar{+} \bar{B} = A \downarrow B \downarrow (\downarrow B \downarrow B)$$

2.1.1.2 B –

1. $f(A,B,C,D) = \bar{A}BD + B\bar{C} + A\bar{C}D = \bar{A}B\bar{D} + \bar{\bar{B}}\bar{C} + A\bar{C}D = (\bar{B}\bar{D}).\bar{\bar{(}}A\bar{\bar{C}}D) = ((A|A)|B((D|D)))|(B|($

<++>