Architecture des ordinateurs - TD 01

16 septembre 2014

1 Conversions de base

1. Convertissez les nombres décimaux suivants en binaire :
   255_{10}, 104_{10}, 2010_{10}

   > **Solution:** $255_{10} = 11111111_2$
   > $104_{10} = 1101000_2$
   > $2010 = 11111011010_2$

2. Convertissez les nombres décimaux suivants en base 5 :
   250_{10}, 78_{10}, 33_{10}, 622_{10},

   > **Solution:** $250_{10} = 2000_5$ $78_{10} = 303_5$ $33_{10} = 113_5$ $622_{10} = 4442_5$

3. Convertissez les nombres suivants en décimal :
   1234_5, 1234_7, 1234_9

   > **Solution:** $1234_5 = 194_{10}$ $1234_7 = 466_{10}$ $1234_9 = 922_{10}$

4. Convertissez les nombres binaires suivants en hexadécimal puis en octal :
   110_2, 1011_2, 11110101100_2, 1100000011011110_2

   > **Solution:** $110_2 = 6_{16} = 6_8$
   > $1011_2 = B_{16} = 13_8$
   > $1100000011011110_2 = C0DE_{16} = 140336_8$
   > $11110101100_2 = 7AC_{16} = 3654_8$

5. Convertissez les nombres hexadécimaux suivants en décimal :
   400_{16}, FFF_{16}, $7FF_{16}$, $A000_{16}$

   > **Solution:** $400_{16} = 1024_{10}$
   > $FFF_{16} = 4095_{10}$
   > $7FF_{16} = 2047_{10}$
   > $A000_{16} = 40960_{10}$

6. Quel est le plus grand entier positif codable sur 9 bits en binaire ? Combien faut-il de chiffres pour l'écrire en octal ? Et en hexadécimal ?

   > **Solution:** $511_{10}(111111111_2)$. Il faut 3 chiffres pour l'écrire en octal et 3 en hexadécimal.

2 Additions et Multiplications en binaire

1. Additionnez les entiers positifs suivants directement en binaire, indiquez les cas qui produisent un overflow de la représentation 8 bits :
   $00101001_2 + 11001010_2$
   $10101011_2 + 11001010_2$
   $11111111_2 + 11111111_2$

   Solution: $00101001_2(41_{10}) + 11001010_2(202_{10}) = 11110011_2(243_{10})$.
   $10101011_2(171_{10}) + 11001010_2(202_{10}) = 101110101_2(303_{10})$ le résultat produit un overflow.
   $11111111_2(255_{10}) + 11111111_2(255_{10}) = 111111110_2(510_{10})$ le résultat produit un overflow.

2. Multipliez les entiers positifs suivants, indiquez les cas qui produisent un overflow de la représentation 8 bits :
   $00001001_2 \times 00001010_2$
   $10101011_2 \times 11001010_2$

   Solution: $00001001_2(9_{10}) \times 00001010_2(10_{10}) = 01011010_2(90_2)$.
   $10101011_2(171_{10}) \times 11001010_2(202_{10}) = 1000011011101110_2(34542_{10})$ le résultat produit un overflow.

3. Quelle est l'entier le plus grand représentable sur n bits ?

   Solution: $2^n - 1$

4. Additionnons un entier x codé en n bits et un entier y codé en m bits, avec $n \geq m$, sur combien de bits faut-il coder le résultat pour éviter un overflow ?

   Solution: Supposons sans perte de généralité que $n \geq m$. Prenons $x_{max} = 2^n - 1$ et $y_{max} = 2^m - 1$. Or $x + y \leq x_{max} + y_{max} = 2^n + 2^m - 2 = 2^n.(1 + 2^{m-n} - 2^{1-n}) < 2^n.2$, il faut donc $n + 1 = max(n, m) + 1$ bits.

5. Soit un entier x codé en n bits et un entier y codé en m bits avec $n \geq m$. Montrez que $n + m$ bits suffisent pour représenter $x \times y$.
   (Bonus plus difficile : montrez que le résultat ne peut pas être codé sur $n + m - 1$ bits si $m \geq 2$.

   Solution: Prenons $x_{max} = 2^n - 1$ et $y_{max} = 2^m - 1$. Or $x \times y \leq x_{max} \times y_{max} = 2^{n+m} - 2^n - 2^m + 1 < 2^{n+m}$, il faut donc au maximum $n + m$ bits.
   Montrons que l'on ne peut pas coder le résultat en $n + m - 1$ bits, soit $2^{n+m} - 2^n - 2^m + 1 > 2^{n+m-1}$.
   On sait $2^n + 2^m \leq 2^n + 2^n \leq 2^{n+1} \leq 2^{n+m-1}$ car $n \geq m \geq 2$.
   $2^n + 2^m + 2^{n+m-1} \leq 2^{n+m-1} + 2^{n+m-1}$
   $2^n + 2^m + 2^{n+m-1} \leq 2^{n+m}$
   $2^n + 2^m + 2^{n+m-1} - 1 < 2^{n+m}$

3 Représentation des binaires codés en décimal

1. Donnez la représentation en binaire codé en décimal des nombres suivants :
   89
   2048
   1984

Solution: $89 = 10001001$
$2048 = 0010000001001000$
$1984 = 0001100110000100$

2. Donnez la valeur décimale des nombres codés en binaire suivants :
   01000010, 0010000000010001, 010100010010

Solution: $01000010 = 42$
$0010000000010001 = 2011$
$010100010010 = 512$

3. Estimez la rapport entre le nombre de bits nécessaires pour coder un nombre en binaire et en binaire codé décimal. Rappel de cours : pour coder l'entier N, il faut $\lceil log_b(N+1) \rceil$ bits en base b. Quel est la représentation la plus compacte ?

Solution: Binaire : Il faut $\lceil log_2(n+1) \rceil$ bits.
BCD : Il faut $4 * \lceil log_{10}(n+1) \rceil$ bits.
En applicant la formule $log_b(n) = ln(n)/ln(b)$ à ces équations, on obtient alors un ratio d'environ $4 * ln(2)/ln(10)$, soit autour de $1,2$.

4 Représentation en virgule fixe

Dans la représentation en virgule fixe, un nombre décimal s'écrit avec n chiffres pour la partie entière et q chiffres pour la partie fractionnaire. Voici quelques nombres décimaux lorsque $n = 2$ et $q = 3$: $12,345$, $05,217$.

1. Arrondissez les rationnels suivants à la valeur décimale la plus proche en virgule fixe ($n = 2$ et $q = 3$) : 1/3 15/7

Solution: $0,333$ $2,143$

2. Quelle est l'erreur maximale obtenue lors d'un arrondi dans la représentation décimale en virgule fixe (n,q) ?

Solution: 0.5×10^{-q}

3. Quel est le résultat de l'opération suivante : dans les réels ? en virgule fixe ($n = 2$ et $q = 3$) ?
   $0,14 \times 0,99$

Solution: $0,1386$
$0,139$