ESIR – CUPGE 1ère année

ELECTRONIQUE NUMERIQUE

Semestre 1 – J. Bézy-Wendling

SeqA – Chap1
01/11/2021

CODAGE DE L'INFORMATION
ET
OPERATIONS ARITHMETIQUES

A. CODAGE DE L'INFORMATION

1. Nombres entiers

2. Changement de base de représentation

B. OPERATIONS ARITHMETIQUES

Addition de deux nombres binaires

A. CODAGE DE L'INFORMATION

1. Nombres entiers

- Plusieurs façons de représenter les nombres entiers :

✓ **Base 2** (binaire) avec deux symboles (bits) : 0 et 1
✓ **Base 8** (octal) avec huit symboles : 0,1,2,3,4,5,6,7
✓ **Base 10** (décimal) avec dix symboles : 0,1,2,3,4,5,6,7,8,9
✓ **Base 16** (héxadécimal) avec seize symboles : 0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F les symboles lettres valant 10, 11, 12, 13, 14 et 15 en décimal.

- Dans les quatre représentations (pondérées) :
  - ✓ Nombre N s'exprime en fonction de la base (B) et des symboles a_i :

$$\boldsymbol{N = \sum_{i=0}^{n-1} a_i B^i}$$

- En pratique : symboles juxtaposés (gauche : **poids forts**, droite : **poids faibles**)

- Exemples :

$$N_2 = 10110_2 = 1.2^4 + 0.2^3 + 1.2^2 + 1.2^1 + 0.2^0 = 16 + 4 + 2 = 22_{10}$$

$$N_8 = 572_8 = 5.8^2 + 7.8^1 + 2.8^0 = 5 * 64 + 7 * 8 + 2 * 1 = 378_{10}$$

$$N_{10} = 782_{10} = 7.10^2 + 8.10^1 + 2.10^0 = 700 + 80 + 2 = 782_{10}$$

$$\begin{aligned} N_{16} = 2A9F_{16} &= 2.16^3 + 10.16^2 + 9.16^1 + 15.16^0 \\ &= 2 * 4096 + 10 * 256 + 9 * 16 + 15 * 1 = 10911_{10} \end{aligned}$$

Comptage (succession de nombres) dans les différents systèmes de représentation

Décimal	Binaire	Octal	Hexadécimal
00	000 000	00	00
01	000 001	01	01
02	000 010	02	02
03	000 011	03	03
04	000 100	04	04
05	000 101	05	05
06	000 110	06	06
07	000 111	07	07
08	001 000	10	08
09	001 001	11	09
10	001 010	12	0A
11	001 011	13	0B
12	001 100	14	0C
13	001 101	15	0D
14	001 110	16	0E
15	001 111	17	0F
16	010 000	20	10
17	010 001	21	11
18	010 010	22	12
19	010 011	23	13

..etc

2. Changement de bases de représentation

- Soit un nombre N, exprimé dans une base B_1, à **convertir** dans une base B_2
- Méthode dépend des bases de départ et d'arrivée

2.1. Binaire ↔ Octal / Hexadécimal

- Méthode **Binaire → Octal** :

  ✓ diviser le nombre en groupes de 3 bits en partant de la virgule (parfois nécessaire d'ajouter un 0 au début ou à la fin d'un groupe de 3 bits)

  ✓ remplacer chaque groupe de 3 bits par un nombre de 0 à 7 :

  000 → 0, 001 → 1, …, 110 → 6, 111 → 7

- Méthode **Octal → Binaire** :

  ✓ remplacer chaque chiffre octal (0……7) par un groupe de 3 bits :

  0 → 000, 1 → 001, …, 7 → 111

- Méthode **Binaire ↔ Hexadécimal** :
  - ✓ Même principe avec des groupes de 4 bits.

Exercice d’application

Donner les représentations en base 8, puis en base 16 des deux nombres suivants représentés en base 2.

- 0010101110011010_2
- $1101011101010001{,}101_2$

Correction

- **1er nombre :**
  - ✓ *En octal*

    000 010 101 110 011 010 (2)

    0 2 5 6 3 2 (8)

  - ✓ *En hexadécimal*

    0010 1011 1001 1010 (2)

    2 B 9 A (16)

- **2ème nombre :**
  - ✓ *En octal*

    001 101 011 101 010 001,101 2

    1 5 3 5 2 1 , 5 (8)

  - ✓ *En hexadécimal*

    1101 0111 0101 0001 , 1010 2

    D 7 5 1 , A (16)

2.2. Binaire ↔ Décimal

- Deux méthodes pour convertir un **nombre décimal entier** vers son équivalent binaire

- **Méthode 1** (pour les petits nombres) :
  - ✓ Exprimer le nombre comme une somme de puissances de 2
  - ✓ Inscrire des 1 et des 0 en face des positions binaires
  - ✓ Exemple : $39_{10}=32+4+2+1 \quad =1\text{x}32+0\text{x}16+0\text{x}8+1\text{x}4+1\text{x}2+1\text{x}1$

    $=1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1_2$
- **Méthode 2** : divisions successives

  - ✓ Convient mieux aux grands nombres
  - ✓ Division du quotient par 2, **jusqu'à ce qu'il soit nul**
  - ✓ Nombre binaire formé par les restes des divisions (le premier reste obtenu en position poids faible, et le dernier reste obtenu en poids fort)

- **Exemple Méthode 2 : (on prend les restes de bas en haut mais on les écrit de gauche à droite.**

$\frac{34}{2} = 17 + reste\ de\ 0$ $\frac{17}{2} = 8 + reste\ de\ 1$ $\frac{8}{2} = 4 + reste\ de\ 0$ $\frac{4}{2} = 2 + reste\ de\ 0$ $\frac{2}{2} = 1 + reste\ de\ 0$ $\frac{1}{2} = 0 + reste\ de\ 1$	**Résultat :** **$34_{10}=100010_2$**

2.3. Décimal ↔ Octal

- **Octal → Décimal :**
  - ✓ Multiplier chaque chiffre octal par son **poids** positionnel (**puissance de 8**)
  - ✓ Additionner les nombres ainsi obtenus
  - ✓ Exemple : $234_8=2x8^2+1x8^1+4x8^0=2x64+8+4=140_{10}$
  - ✓ Exemple : $13,6_8=1x8^1+3x8^0+6x8^{-1}=8+3+0,75=11,75_{10}$

- **Décimal → Octal :**
  - ✓ Passage par le binaire (divisions par 2 puis conversion binaire→Octal)
  - ✓ Divisions successives par 8

$\frac{150}{8} = 18 + reste\ de\ 6$ $\frac{18}{8} = 2 + reste\ de\ 2$ $\frac{2}{8} = 0 + reste\ de\ 2$	**Résultat :** $\mathbf{150_{10}=226_8}$

2.4. Décimal ↔ Hexadécimal

- **Hexadécimal → Décimal :**
  - ✓ Multiplier chaque chiffre hexadécimal par son **poids** positionnel (**puissance de 16**)
  - ✓ Additionner les nombres ainsi obtenus
  - ✓ Exemple : $A3C_{16}=10x16^{2}+3x16^{1}+12x16^{0}=2560+48+12=2620_{10}$
  - ✓ Exemple : $2B,8_{16}=2x16^{1}+11x16^{0}+8x16^{-1}=32+11+0,5=43,5_{10}$
- **Décimal →Hexadécimal:**
  - ✓ Passage par le binaire (divisions par 2 puis conversion binaire→Hexa)
  - ✓ Divisions successives par 16

$\frac{\mathbf{287}}{\mathbf{16}} = \mathbf{17} + \boldsymbol{reste\ de}\ \mathbf{15}$ $\frac{\mathbf{17}}{\mathbf{16}} = \mathbf{1} + \boldsymbol{reste\ de}\ \mathbf{1}$ $\frac{\mathbf{1}}{\mathbf{16}} = \mathbf{0} + \boldsymbol{reste\ de}\ \mathbf{1}$	**Résultat :** **$287_{10}=11F_{16}$**

B. OPERATIONS ARITHMETIQUES

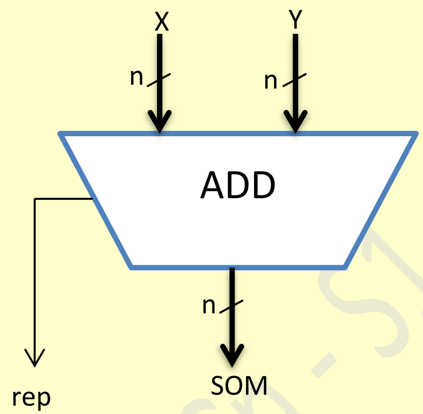
Addition de deux nombres binaires (<u>non signés)</u>

Principe de l'addition

- Soit deux nombres A et B représentés sur quatre bits
- Somme de A et B : addition bit par bit, de la droite vers la gauche en propageant la retenue au rang immédiatement supérieur.

Nombre A	a_4	a_3	a_2	a_1
Nombre B	b_4	b_3	b_2	b_1
Somme A+B	S_4	S_3	S_2	S_1
Retenues	$\leftarrow r_4$	r_3	r_2	r_1

- Opérateur ADD sur mots de n bits :



- Fonctionnement défini par un algorithme : algorithme d'addition binaire

x_i y_i r_i	r_{i+1}	s_i
0 0 0	0	0
0 0 1	0	1
0 1 0	0	1
1 0 0	0	1
0 1 1	1	0
1 1 0	1	0
1 0 1	1	0
1 1 1	1	1

Appliqué itérativement avec $r_0=0$

$SOM=s_{n-1}\ s_{n-2}\ldots s_0$

$rep=r_n$ ("report ou retenue")

Exemple :

$$\begin{array}{lr} X & 1011 \\ Y & \underline{0011} \\ & 0\ \ 1110 \\ & \text{rep}\ \ \text{SOM} \end{array}$$

ESIR ECOLE SUPERIEURE D'INGENIEURS DE RENNES

Exercice d’application

1) Additionner les nombres binaires suivants :

$1011\ 0111_{(2)} + 1100\ 1000_{(2)}$

$0111\ 1010_{(2)} + 1000\ 0101_{(2)}$