Université Sultan Moulay Slimane
Ecole Supérieure de Technologie
Département Mécatronique

Electronique Numérique

Chapitre 2 : Opérations arithmétiques binaires

Pr. Assia ARSALANE

Plan du chapitre

I. Introduction

II. Addition binaire

III. Débordement

IV. Multiplication binaire

V. V. Multiplication ou division par une puissance de 2

VI. Soustraction binaire

VII. Addition/soustraction binaire en signé

VIII. Addition/soustraction binaire en complément à 2 : débordement

IX. Addition/soustraction binaire en complément à 2 : Exemples

I. Introduction

- Les opérations arithmétiques (addition, soustraction, multiplication, division) sont réalisables dans toute base B
  - Avec mêmes règles que pour la base décimale
  - Retenues également mais dépendant de la base
  - Quand on additionne 2 chiffres a et b dans la base B
    - Si la somme des valeurs décimales de a et b dépasse ou égale B alors il y a une retenue
- Exemple : principes de l'addition binaire
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 1 = 10 soit 0 avec une retenue de 1
  - 1 + 1 + 1 = 11 soit 1 avec une retenue de 1

II. Addition binaire

- Exemple : 10 + 1011

```
    0010    = 2
 +  1011    = 11
   -------
    1101    = 13
```

- Exemple : 1101 + 1010

```
    1101    = 13
 +  1010    = 10
   -------
   10111    = 23
```

- Addition de 2 nombres de 4 bits : on a besoin dans cet exemple de 5 bits
- Potentiel problème de débordement

III. Débordement

- Débordement : la taille allouée (8, 16 ... bits) au codage d'un entier est trop petite pour coder ou stocker le résultat d'un calcul
- Exemple avec addition, sur 8 bits, non signé :
  - 10110011 + 10000101 = 100111000
  - Besoin de 9 bits pour coder le nombre
  - Stockage du résultat impossible sur 8 bits
- Exemple avec addition, sur 8 bits, signé :
  - **0**1110011 + **0**1000101 = **1**0111000
  - Addition de 2 positifs donne un négatif !

III. Débordement

Exercice à faire :

- Faire Addition binaire sur 8 bits des deux nombres a et b
  - a = 11000101
  - b = 11100011
- Représenter a, b en hexadécimal et faite la somme en hexadécimal.
- Si on effectue la somme avec a et b codés sur 16 bits. Quel est l'état de C ? Justifiez votre réponse

IV. Multiplication binaire

- Comme en décimal
- N'utilise que du décalage de bits et additions
- Exemple : 101 x 110 :

```
        101      = 5
    x   110      = 6
        -----
        000
  +    1010
  +   10100
     --------
      11110      = 30
```

- Décalage d'un bit vers la gauche = multiplication

- Décalage d'un bit vers la droite = division entière par 2

V. Multiplication ou division par une puissance de 2

1. Multiplication par 2^k

- La multiplication d'un entier **a** codé sur **n bits** par $\mathbf{2^k}$ est obtenue par un décalage de la suite de bits de **a** de **k** positions vers la gauche.
- Les bits de plus faible poids sont remplacés par des '0'.
- Si le résultat n'est représentable, un débordement est généré.
- Exemple : soit l'opération 10*4 = 40. Cette opération réalisée sur 8 bits donne :
  - 10 = 0000 1010
  - 0000 1010 décalé à gauche de 2 postions donne : 0010 1000 soit 40
- Exercice à faire : à quoi correspond l'opération 16*3.

V. Multiplication ou division par une puissance de 2

2. Division par 2^k

- La division d'un entier **a** codé sur **n bits** par **2^k** est obtenue par un décalage de la suite de bits de **a** de **k** positions vers la droite.
- Le bit de signe est recopié dans les bits de poids le fort
- Ce type de décalage est dit arithmétique, dans le décalage dit logique, les **k** positions de poids fort sont remplies de 0.
- Exemple : Réaliser l'opération $\frac{20}{2} = 10$
- 20 = 16 + 4 = 0001 0100 et 0001 0100 décalé à droite d'une position donne 0000 1010 soit 10.
- Exercice à faire : à quoi correspond l'opération $\frac{30}{2} = 15$

VI. Soustraction en binaire

- Soustraction binaire : peut se faire comme en décimal

0 – 0 = 0

1 – 0 = 1

0 – 1 = 1 on emprunt 1

1 – 1 = 0

Exemple : 1101 – 1011 :

	1101	= 13
–	1011	= 11
	0010	= 2

- Autre technique
  - Utiliser les compléments à 2 et ne faire que des additions

VII. Addition/soustraction binaire en signé

- Codage en complément à 2 (CV)
  - Simplifie les additions et soustractions
  - On peut additionner directement des nombres, quels que soient leurs signes, le résultat sera directement correct (si pas de débordement) et « bien codé »
  - Soustraction d'un nombre = addition de son complément à 2
    - A – B = A + CV(B)
    - Valable dans tous les cas, quels que soient les signes de A et B
    - Là aussi le résultat est directement valide si pas de débordement

VIII. Addition/soustraction binaire en complément à 2 : débordement

- Gestion des débordements différent de l'addition non signée
- Une retenue sur un bit supplémentaire par rapport à la précision ne veut pas forcément dire que le résultat n'est pas stockable avec la précision utilisée
- On regarde les retenues des deux derniers bits (poids forts) additionnés pour savoir s'il y a eu débordement
  - Si retenues identiques (00 ou 11) : pas de débordement
  - Si retenues différentes (01 ou 10) : débordement
- On néglige systématiquement la retenue sur le bit supplémentaire pour déterminer le résultat final
  - S'il n'y a pas de débordement, le résultat tient dans la précision requise, la retenue n'a aucune signification

VIII. Addition/soustraction binaire en complément à 2 : débordement

- Débordement (suite)
  - Règles valables pour toute addition ou soustraction utilisant des entiers signés codés en complément à 2
    - Avec ce mode de codage des nombres signés, le débordement est différent du codage des entiers non signés
- Signe du résultat : on regarde le bit de poids fort
  - Si 0 : résultat est un nombre positif
  - Si 1 : nombre négatif
    - Le résultat est directement codé en complément à 2
    - Sa valeur absolue est trouvée par le calcul de la Grandeur Exacte.

IX. Addition/soustraction binaire en complément à 2 : Exemples

- Exemples de calcul avec codage des entiers signés en complément à 2, précision de 5 bits
  - 9 = $(01001)_2$ 8 = $(01000)_2$ 5 = $(00101)_2$
  - -9 = $(10111)_2$ -8 = $(11000)_2$ -5 = $(11011)_2$
  - Calcul de 9 + 8 :

01 → retenues

```
   01001  = 9
+  01000  = 8
  --------
  010001
```

- Résultat tient sur 5 bits mais calcul faux
- Car 2 dernières retenues sont différentes

IX. Addition/soustraction binaire en complément à 2 : Exemples

- Calcul de 9 – 8 = calcul de 9 + CV (8)

  = calcul de 9 + (-8)

11 → retenues

```
   01001  =  9
+  11000  = - 8
  --------
  100001
```

- Résultat ne tient pas sur 5 bits mais calcul correct
- Car 2 dernières retenues sont identiques
- Le bit de débordement (le 6ème bit) est à ignorer, il n'a aucune signification
- Le 5ème bit = 0 : nombre positif
  - Résultat = $(00001)_2$ = 1

IX. Addition/soustraction binaire en complément à 2 : Exemples

- Calcul de 5 – 8 = calcul de 5 + CV (8)

  = calcul de 5 + (-8)

```
  00 → retenues
   00101  =  5
+  11000  = - 8
  --------
  011101
```

  - Calcul correct car 2 dernières retenues sont identiques
  - Le 5$^{\text{ème}}$ bit = 1 : nombre négatif
  - Valeur absolue du résultat = CV(11101) = $(00011)_2$ = 3
  - Donc résultat de 5 – 8 = - 3

IX. Addition/soustraction binaire en complément à 2 : Exemples

- Calcul de -9 – 8 = calcul de CV(9) + CV(8)

  = calcul de (-9) + (-8)

  10 → retenues

```
   10111  = - 9
+  11000  = - 8
  --------
  101111
```

- Calcul incorrect car 2 dernières retenues sont différentes

IX. Addition/soustraction binaire en complément à 2 : Exemples

- Calcul de -5 – 8 = calcul de CV(5) + CV(8)

  = calcul de (-5) + (-8)

  11 → retenues

$$
\begin{array}{rl}
 & 11011 \quad = -5 \\
+ & \underline{11000} \quad = -8 \\
 & 1\underline{1}0011
\end{array}
$$

  - Calcul correct car 2 dernières retenues sont identiques
  - Le $5^{\text{ème}}$ bit = 1 : nombre négatif
  - Valeur absolue du résultat = CV(10011) = $(01101)_2$ = 13
    - On ignore systématiquement le $6^{\text{ème}}$ bit
  - Donc résultat de - 5 – 8 = - 13