Opérateurs arithmétiques

Mise en perspective

Éloi de Chérisey, Sylvain Guilley

Table des matières

Introduction

Représentation des nombres

- Représentation simple
- Conversion entre bases
- Représentation des nombres entiers (positifs ou négatifs)

Fonction arithmétiques

- L'additionneur
- Le soustracteur

Exercices

Table des matières

Introduction

Représentation des nombres

- Représentation simple
- Conversion entre bases
- Représentation des nombres entiers (positifs ou négatifs)

Fonction arithmétiques

- L'additionneur
- Le soustracteur

Exercices

Objectifs

- Savoir faire un changement de base.
- Connaître les bases 2, 16 et 10.
- Encodage des nombres dans la *vraie vie*.
- Addtionneur.
- Soustracteur.

Table des matières

Introduction

Représentation des nombres
- Représentation simple
- Conversion entre bases
- Représentation des nombres entiers (positifs ou négatifs)

Fonction arithmétiques
- L'additionneur
- Le soustracteur

Exercices

Les bases

Soit N un nombre décimal : $N = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + a_1 b^1 + a_0 b^0 + \ldots + a_{-1}b^{-1} + a_{-2}b^{-2} + \ldots + a_{-m}b^{-m}$

- a_{n-1} est le chiffre le plus significatif.
- a_{-m} est le moins significatif.

Bases (suite)

1. Base hexadécimale : $b = 16$,
   $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
2. Base octale : $b = 8$, $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$
3. binaire : $b = 8$, $a_i \in \{0, 1\}$

Exo

Tranformer les nombres de 1 à 8 en binaire
Sur combien de bits faut-il encoder ces nombres ?

Conversion entre bases

- Base b vers base 10 : Appliquer la formule

$$N = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + a_1 b^1 + a_0 \Delta b^0 + \ldots + a_{-1}b^{-1} + a_{-2}b^{-2} + \ldots + a_{-}$$

- Base 10 vers base b
  Algorithme d'Euclide (si mes souvenirs de 3e sont exacts)

Soit la conversion 57_{10} vers base 2 :

Division	Quotient	Reste
$57/2$	28	$1(a_0)$
$28/2$	14	$0(a_1)$
$14/2$	7	$0(a_2)$
$7/2$	3	$1(a_3)$
$3/2$	1	$1(a_4)$
$1/2$	0	$1(a_5)$

Le résultat est donc $(111001)_2$.

Partie fractionnaire

Exo

Transformer $0,57_{10}$ en base 2

Partie fractionnaire

Exo

Transformer $0,57_{10}$ en base 2

Soit la conversion $0,57_{10}$ vers base 2 :

Multiplication	Produit	Partie entière
$0,57 \cdot 2$	$1,14$	$1 (a_{-1})$
$0,14 \cdot 2$	$0,28$	$0 (a_{-2})$
$0,28 \cdot 2$	$0,56$	$0 (a_{-3})$
$0,56 \cdot 2$	$1,12$	$1 (a_{-4})$
$0,12 \cdot 2$	$0,24$	$0 (a_{-5})$
$0,24 \cdot 2$	$0,48$	$0 (a_{-6})$

Le résultat est donc $(0,100100)_2$.

Nombres négatifs - comment faire ?

Example

Comment représenter le nombre -5 ?

- Mettre un bit de signe au début en plus :

$$-5 = \mathbf{1}101$$

Question : quelles sont les problèmes liés à une telle représentation ?

Nombres négatifs - comment faire ?

Example

Comment représenter le nombre -5 ?

- Mettre un bit de signe au début en plus :

$$-5 = \mathbf{1}101$$

  Question : quelles sont les problèmes liés à une telle représentation ?
- Représentation en complément à 2 :

$$-5 = 1011$$

Complément à 2

- Sur n bits, on peut représenter 2^n nombres.
- 2^{n-1} négatifs, $2^{n-1}-1$ positifs et le 0.
- Si nombre positif ou nul (de 0 à $2^{n-1}-1$), on représente comme d'habitude.
- Si nombre négatif, (de -1 à 2^{-n}) alors, on prend le modulo à 2^n et on code sur ce modulo.
  Par exemple -5 = 16 - 5 modulo 16 = 11 modulo 16 = 1011

Exemple en C avec unsigned char

On compile le code suivant :

```c
#include <stdio.h>

int main(){

char i = 128;
unsigned char j = 128;

printf("i=%d,\tj=%d\n",i,j);

return(0);
}
~
~
```

Qu'obtient-on ?

Autres types de code

- Code de Gray
- Binary Coded Decimal
- Code à bit de parité

Exo

Représenter les nombres de 0 à 15 dans ces trois codes différents

Table des matières

Introduction

Représentation des nombres

- Représentation simple
- Conversion entre bases
- Représentation des nombres entiers (positifs ou négatifs)

Fonction arithmétiques

- L'additionneur
- Le soustracteur

Exercices

Création d'un additionneur

Étapes par étape...

1. Considérer uniquement l'addition de deux bits.
2. Table de vérité, puis schéma avec des portes logiques.
3. Puis, considérer, l'addition de deux bits en prenant en compte une retenue.
4. De même, table de vérité, tableau de Karnaugh si besoin et schéma des portes logiques.

A vous de jouer !

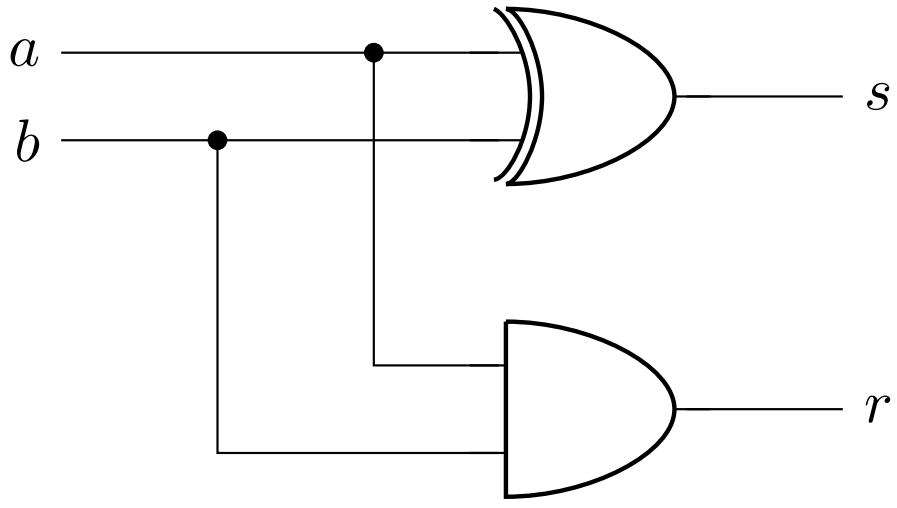
Solution pour l'additionneur - première étape

On considère deux bits a et b codés sur un bit à additionner. Les solutions sont donc $0, 1$ ou 2.
La solution étant représentée sur 1 bit, il faut donc propager une retenue r. On a donc la table de vérité suivante :

a	b	s	r
0	0	0	0
0	1	1	0
1	1	0	1
1	0	1	0

On a immédiatement $s = a \oplus b$ et $r = a.b$.

Première étape - portes logiques



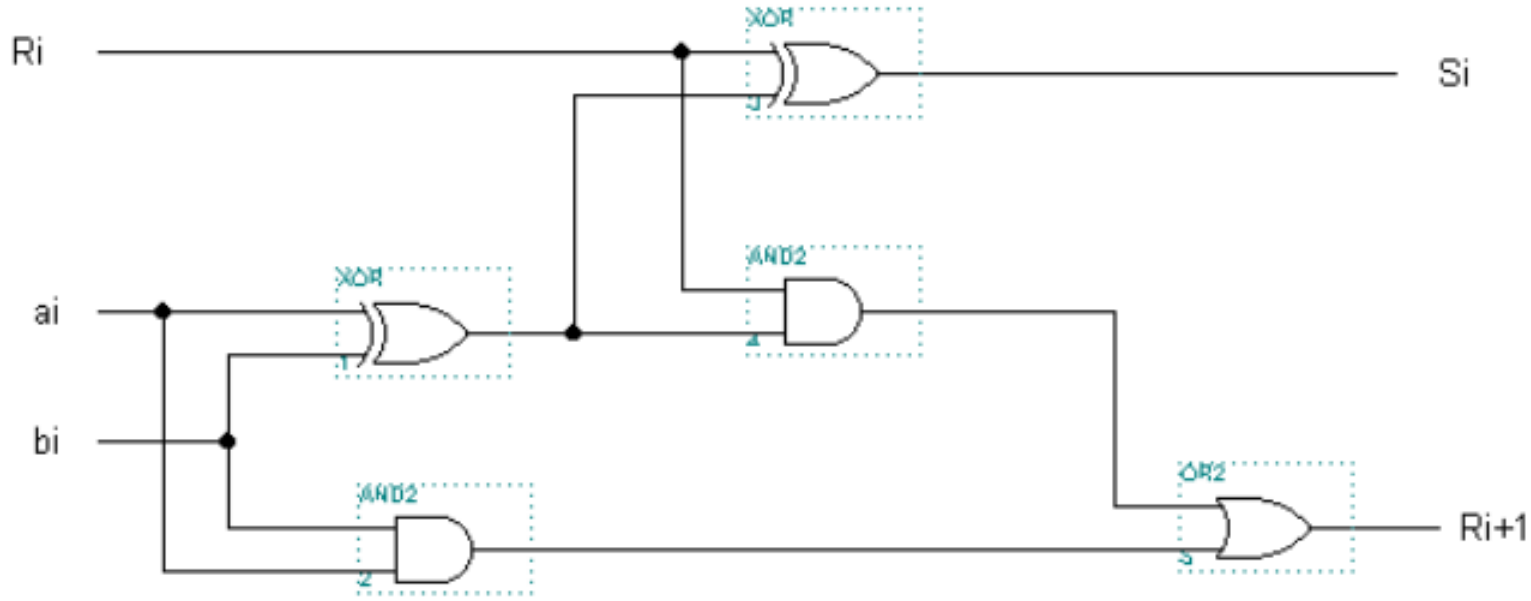
Solution pour l'additionneur - deuxième étape étape

On considère deux bits a_i et b_i codés sur un bit à additionner, avec une retenue r_i venant de l'étage précédant. Les solutions sont donc $0, 1, 2$ ou 3.

a_i	b_i	r_i	s	r_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	1	0	1
0	1	0	1	0
1	1	0	0	1
1	1	1	1	1
1	0	1	0	1
1	0	0	1	0

Schéma en portes logiques

On a $s_i = a_i \oplus b_i \oplus r_i$ et $r_{i+1} = a_i.b_i + r_i.(a_i \oplus b_i)$.



Le soustracteur

Faites de même pour le soustracteur !

Attention !

La soustraction n'est pas une loi commutative.

Table des matières

Introduction

Représentation des nombres

- Représentation simple
- Conversion entre bases
- Représentation des nombres entiers (positifs ou négatifs)

Fonction arithmétiques

- L'additionneur
- Le soustracteur

Exercices

Exercice 1

- Donner la représentation en complément à 2 des nombres suivants : +8, -8, -30, -52, +15
- Soit B un nombre codé en CA2 sur n bits, comment obtient-on la valeur de B à partir de sa représentation lorsqu'il est positif ? lorsqu'il est négatif ?
- Donner la représentation en CA2 des nombres +15, -12 sur 5 bits, puis sur 7 bits.
- D'une façon générale, comment peut-on étendre la représentation d'un nombre codé en CA2 sur n bits, à une représentation sur p bits, avec p plus grand que n ?

Exercice 2

1. Quel est l'intervalle de variation d'un nombre codé en CA2 sur n bits ?
2. Soient A et B deux nombres codés en CA2 sur n bits et S la somme de ces 2 nombres. Quel est l'intervalle de variation de S ? En déduire le nombre de bits nécessaires à son codage.
3. Réaliser en binaire les additions suivantes : 30 + 8, 30 + (-8), (-30) + 8, (-30) + (-8).

Exercice 3

On désire réaliser un opérateur capable d'effectuer la comparaison de 2 nombres positifs A et B codés sur 4 bits. La sortie S de l'opérateur vaut 1 si A est strictement inférieur à B, 0 sinon.

1. Proposer une solution à l'aide d'un soustracteur.
2. Une autre solution appelée *comparaison MSB en tête* consiste à comparer bit à bit les nombres A et B en commençant par les bits de poids forts. L'algorithme utilisé est le suivant :
   $S = 1$ si $(a_3 < b_3)$
   OU $((a_3 = b_3)$ ET $(a_2 < b_2))$
   OU $((a_3 = b_3)$ ET $(a_2 = b_2)$ ET $(a_1 < b_1))$
   OU $((a_3 = b_3)$ ET $(a_2 = b_2)$ ET $(a_1 = b_1)$ ET $(a_0 < b_0))$.

Exercice 3 - suite

- Construire l'opérateur élémentaire à 2 entrées a_i et b_i dont les sorties I_i(Inférieur) et E_i (Égal) vérifient :

$$I_i = 1 \text{ si } a_i < b_i$$
$$I_i = 0 \text{ sinon}$$
$$E_i = 1 \text{ si } a_i = b_i$$
$$E_i = 0 \text{ sinon}$$

- En utilisant l'opérateur construit précédemment, proposer le schéma complet du comparateur.
- Généraliser à n bits.

Exercice 4

1. Réaliser à la main l'opération 1001×1100 (9×12).
2. Proposer le schéma d'un multiplieur de 2 nombres positifs de 4 bits. On dispose pour cela d'additionneurs 4 bits.