Structure Machine1

Centre Universitaire Abdalhafid Boussouf Mila

2022/2023

Programme de la matière:

- **Matière : Structure machine 1**
- **Unité d'enseignement : Fondamentale**
- **Crédits : 5**
- **Coefficient : 3**

- **Objectifs de l'enseignement :**

Le but de cette matière est de présenter et d'approfondir les notions concernant Les différents systèmes de numération ainsi que la présentation de l'information quelle soit de type numérique ou caractère.

Les base le l'algèbre de Boole sont, eux aussi, abordés de façon approfondie.

Programme de la matière:

Contenu de la matière :

- **Chapitre 1 :**

  Introduction générale.

- **Chapitre 2 : Les systèmes de numération**

  Définition

- Présentation des systèmes:

  décimal, binaire, octal et hexadécimal.

- Conversion entre ces différents systèmes.
- Opérations de base dans le système binaire :

  Addition, Soustraction, Multiplication, Division.

Programme de la matière:

Chapitre 3 : La représentation de l’information

- **Représentation des nombres :**

  1- Nombres entiers : Représentation non signée, Représentation avec signe et valeur absolue, Complément à un , Complément à deux.

  2- Les nombres fractionnaires : Virgule fixe, flottante.

- **Le codage binaire :** Le codage binaire pur, Le code binaire réfléchi, Le code DCB , Le code excède de trois.
- **Représentation des caractères :** Code EBCDIC, ASCII, UTF.

Programme de la matière:

Chapitre 4 : L’algèbre de Boole binaire

- Définition de l’algèbre de Boole: (Théorèmes et propriétés).
- Les opérateurs logiques: (ET, OU, négation, NAND et NOR, Ou exclusif) et la Représentation schématique.
- Table de vérité, Expressions et fonctions logiques, Ecriture algébrique d’une fonction sous première et deuxième forme normale , Expression d’une fonction logique avec des circuits NANDs ou NOR).
- Schéma logique d’une fonction.
- Simplification d’une fonction logique :(Méthode algébrique, Tableaux de Karnaugh, Méthode de quine-mc cluskey).

Chapitre 1 : Systèmes de numération

- Introduction
- Codage d'information
- Les systèmes de numération
  - Le système décimal
  - Le système binaire , octal ethexadécimal
- Le transcodage (conversion de base).
- Les opérations arithmétiques.

Objectifs

- Comprendre c'est quoi le codage de l'information.
- Apprendre le transcodage (conversion d'une base à une autre).
- Apprendre à faire des opérations arithmétiques en binaire.

Introduction

Les informations traitées par les ordinateurs sont de différentes natures : nombres, texte, images, sons, vidéo, programmes, …

Dans un ordinateur, elles sont toujours représentées sous forme binaire
(**BIT : Binary digIT**) une suite de 0 et de 1,

Codage d'information :

Définition :

- le codage permet d'établir une correspondance qui permet sans ambiguïté de passer d'une représentation (dite externe) d'une information à une autre représentation (dite interne : sous forme binaire) de la même information, suivant un ensemble de règles précises.

Exemple :

- Le nombre 35 : 35 est la représentation externe du nombre trentecinq
- La représentation interne de 35 sera une suite de 0 et 1 (100011)

Nous avons pris l'habitude de représenter les nombres en utilisant dix symboles différents: 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9

- Ce système est appelé le système décimal (**déci** signifie dix).

- Il existe cependant d'autres formes de numération qui fonctionnent en utilisant un nombre de symboles distincts.
  - Exemple :
    - Système binaire (bi: deux),
    - Le système octal (oct: huit),
    - Le système hexadécimal (hexa: seize).
    - ….
- Dans un système de numération : le nombre de symboles distincts est appelé la base du système de numération.

Les systèmes de numération

Système de numération :

Un système de numération décrit la façon avec laquelle les nombres sont représentés.

- Un système de numération est défini par :

- Une base
- Un alphabet A : ensemble de symboles ou chiffres,
- Des règles de représentation des nombres.

Système de numération :

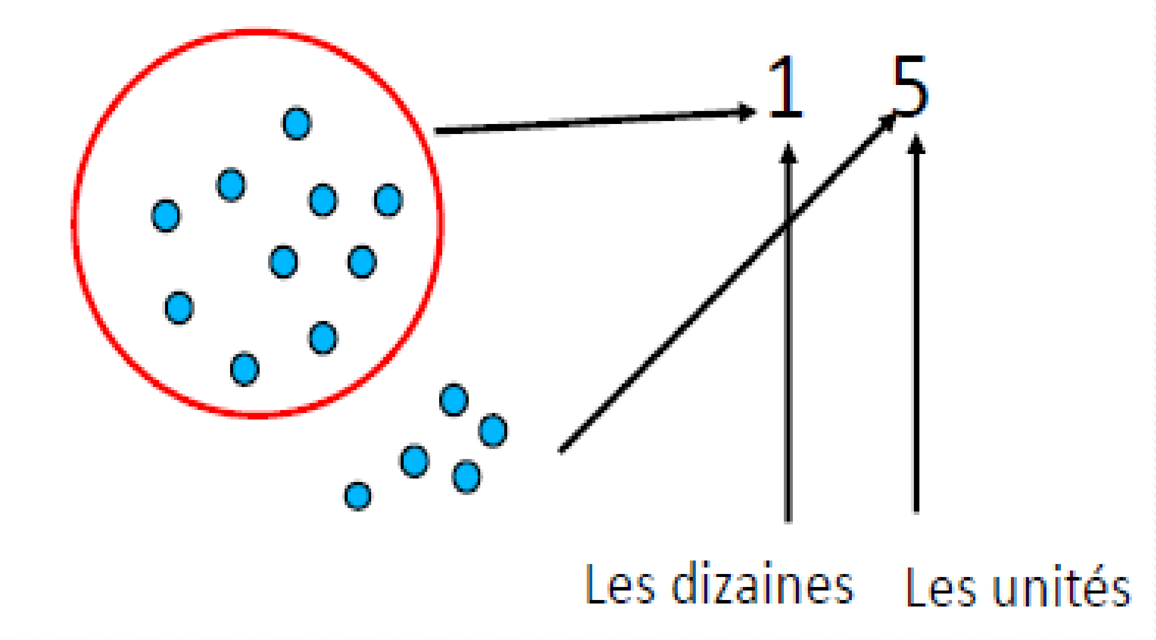
Les systèmes de numérations utilisés dans les domaines de **l'électronique numérique** et de **l'informatique** sont les suivants:

- Système binaire (Base 2)
- Système octal (Base 8)
- Système hexadécimal (Base 16)

En plus du Système décimal (Base 10) utilisé par l’homme pour communiquer avec la machine.

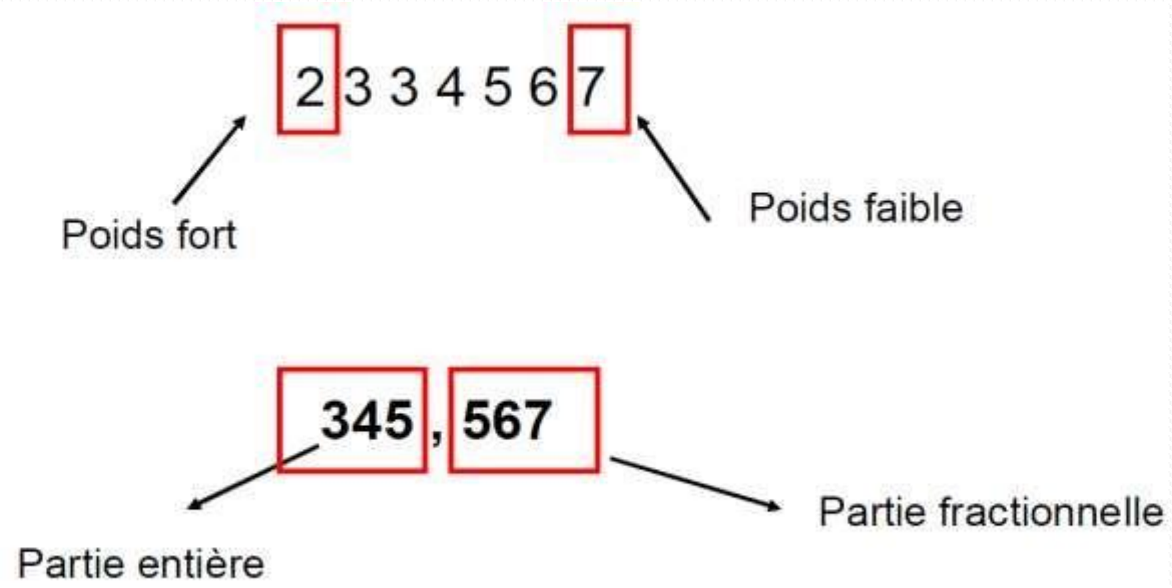
Le système décimal:

- Supposons qu'on a 15 jetons , si on forme des groupes de 10 jetons. On va obtenir **un** seul groupe et il reste **5** jetons:



Le système décimal:

L ’ alphabet de système décimal est composé de dix chiffres différents: **A**= { 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 } N’importe quelle combinaison des symboles:
{ 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 ,9 } nous donne **un nombre**.



Le système décimal:

- Soit le nombre 1982, ce nombre peut être écrit sous la forme suivante :

$(1982)_{10}=2+80+900+1000 =1*2+8*10+9*100+1*1000$

$$\mathbf{(1982)_{10}=2*10^0+8*10^1+9*10^2+1*10^3}$$

Cette forme s'appelle **la forme polynomiale**

Un nombre réel peut être écrit aussi sous la forme polynomiale $(978,265)_{10} = 8*10^0 + 7*10^1 + 9*10^2 + 2*10^{-1} + 6*10^{-2} + 5*10^{-3}$

Comptage décimal:

- Sur une seule position : 0 ,1,2,3,4,5,….9= 10^1-1
- Sur deux positions : 00 , 01,02, …..,99=10^2-1
- Sur trois positions : 000,001,……,999=10^3-1

- Sur n positions :
  - minimum 0
  - maximum 10^n-1
  - nombre de combinaisons 10^n

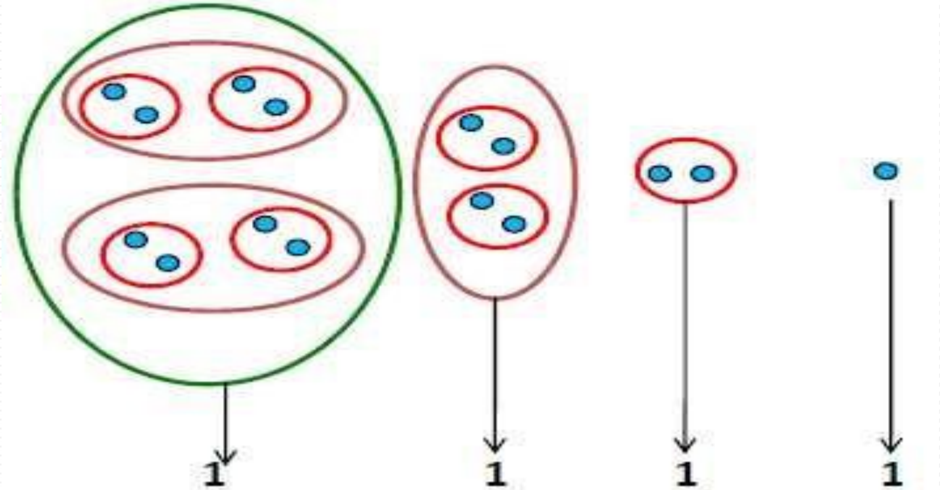
Le système binaire:

Toute communication à l'intérieur de l'ordinateur est faite avec des signaux électriques.

- Un signal électrique à deux états seulement :
  - 1 $\rightarrow$ absence de signal électrique
  - 2 $\rightarrow$ présence de signal électrique
- Une unité d'information (0 ou 1) est appelée bit (de l'anglais binary digit)

Le système binaire:

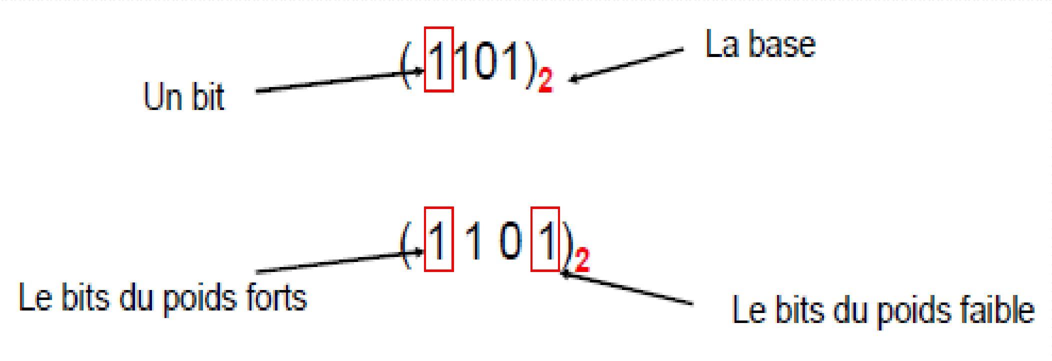
Supposons qu’on a 15 jetons , si on forme des groupes de 2 jetons, en suite des groupes de 2 à 2 consécutivement:



- Le nombre 1111 est la représentation du nombre décimal « 15 » dans la base 2

Le système binaire:

Dans le système binaire, pour exprimer n'importe quelle valeur on utilise uniquement 2 symboles : **A**= { 0 , 1}



Un nombre dans la base 2 peut être écrit aussi sous la forme polynomiale

$(1101)_2 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3$

$(110,101)_2 = 0*2^0 + 1*2^1 + 1*2^2 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3}$

Comptage en binaire

Exemple

- Sur **un seul bit** : 0 , 1
- **Sur 2 bits :**

Binaire	Décimal
00	0
01	1
10	2
11	3

4 combinaisons= 2^2

Sur 3 Bits

$2^2 2^1 2^0$

Binaire	Décimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

8 combinaisons= 2^3

Sur 4 Bits

Binaire	Décimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

16 combinaisons= 2^4

Le comptage binaire:

- Avec un bit nous pouvons coder deux états, avec 2 bits quatre états …. A chaque nouveau bit, le nombre de combinaisons possibles est doublé.
- En utilisant n bits, on peut former **2^n** nombres différents et le plus grand d'entre eux est égal à (**2^n-1**). Par exemple si n= 8, Nmax = (**2^8-1**) = 255, on peut former 256 nombres différents de: $(0)_{10} = (00000000)_2$ -à- $(255)_{10} = (11111111)_2$
- **Remarque:** Un groupe de huit bits est appelé un octet (byte : en anglais)

Le système octal:

C’est le système de base 8

8 symboles sont utilisés dans ce système:

A={ 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 }

- **Exemple de forme polynomiale :**

$(237)_8 = 7*8^0 + 3*8^1 + 2*8^2$

$(53,948)_8 = 3*8^0 + 5*8^1 + 9*8^{-1} + 4*8^{-2} + 8*8^{-3}$

- **Exemple 2 :**

Le nombre (1289) n’existe pas dans la base 8 puisque les symboles 8 et 9 n’appartiennent pas à la base octal .

Le système hexadécimal:

C'est le système de base 16

On utilise seize 16 symboles différents: {0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F}

- **Exemple de forme polynomiale :**

$(A4C)_{16} = 12*16^0 + 4*16^1 + 10*16^2$

$(14,2B)_{16} = 4*16^0 + 1*16^1 + 2*16^{-1} + 11*16^{-2}$

Décimal	Hexadécimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Généralisation: le système B

- Dans une base B , on utilise B symboles distincts pour représenter les nombres.
- La valeur de chaque symbole doit être strictement Inférieur à la base B.
- Chaque nombre dans une base B peut être écrit sous sa forme polynomiale.

$N_B = (a_{n-1}\, a_{n-2} \ldots\, a_2\, a_1\, a_0\, ,\ \mathbf{a_{-1} \ldots\, a_{-m}})_B = a_0.B^0 + \ \ldots + a_{n-2}.B^{n-2} + a_{n-1}.B^{n-1} + \mathbf{a_{-1}.B^{-1}} + \ldots + \mathbf{a_{-m}.B^{-m}} = N_{10}$

Le Transcodage: (conversion de base)

Définition du transcodage:

- **Le transcodage** (ou conversion de base) est l' opération qui permet de passer de la représentation d'un nombre exprimé dans une base à la représentation du même nombre mais exprimé dans une autre base.

Par la suite, on verra les conversions de bases suivantes:

- Décimale vers Binaire, Octale et Hexadécimale
- Binaire vers Décimale, Octale et Hexadécimale

Conversion d'une base B à la base 10:

- Cette conversion est assez simple puisque il suffit de faire le développement en polynôme de ce nombre dans la base B, et de faire la somme par la suite.

- **Exemples :**
- $(1101)_2 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3 = (13)_{10}$
- $(1A7)_{16} = 7*16^0 + 10*16^1 + 1*16^2 = (423)_{10}$
- $(1101,101)_2 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} = (13,625)_{10}$
- $(43,2)_5 = 3*5^0 + 4*5^1 + 2*5^{-1} = (23,4)_{10}$

Conversion du décimal à une base B:

- **Principe :**

- La **division** sur **B** pour la partie entière

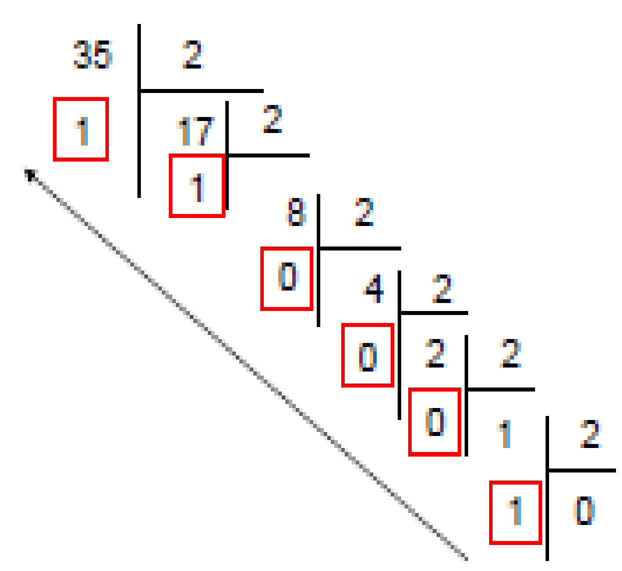
- La **multiplication** par **B** pour la partie fractionnelle.

Conversion de la base 10 à la base 2:

- Le principe consiste à faire des divisions successives du nombre sur 2 , et prendre le reste des divisions dans l'ordre inverse.

- Exemple : $(35)_{10}=(?)_2$

- Après division :



- on obtient :$(35)_{10}=(100011)_2$

Conversion de la base 10 à la base 2:

- (cas d'un nombre réel) : Un nombre réel est constitué de deux parties : la partie entière et la partie fractionnelle.
- La partie entière est transformée en effectuant des divisions successives.
- La partie fractionnelle est transformée en effectuant des multiplications successives par 2 .
- **Exemple 1:** 35,625=(?)2

0,625 * 2 =**1**,250

0,25* 2 = **0** ,50

0,5* 2 = **1** ,0,

P.E =35 = (100011)2

P.F= 0,625) = (0,101)2

Donc 35,625=(100011,101)2

Conversion de la base 10 à la base 2:

- **<u>Exemple 2:</u>** Effectuer la conversion suivante $(0,7)_{10}=(?)_2$ (cas d'un nombre réel)

$0,7 * 2 = 1,4$

$0,4 * 2 = 0,8$

$0,8 * 2 = 1,6$

$0,6 * 2 = 1,2$

$0,2 * 2 = 0,4$

$(0,7) = (0,10110)_2$

- Le nombre de bits après la virgule va déterminer la précision

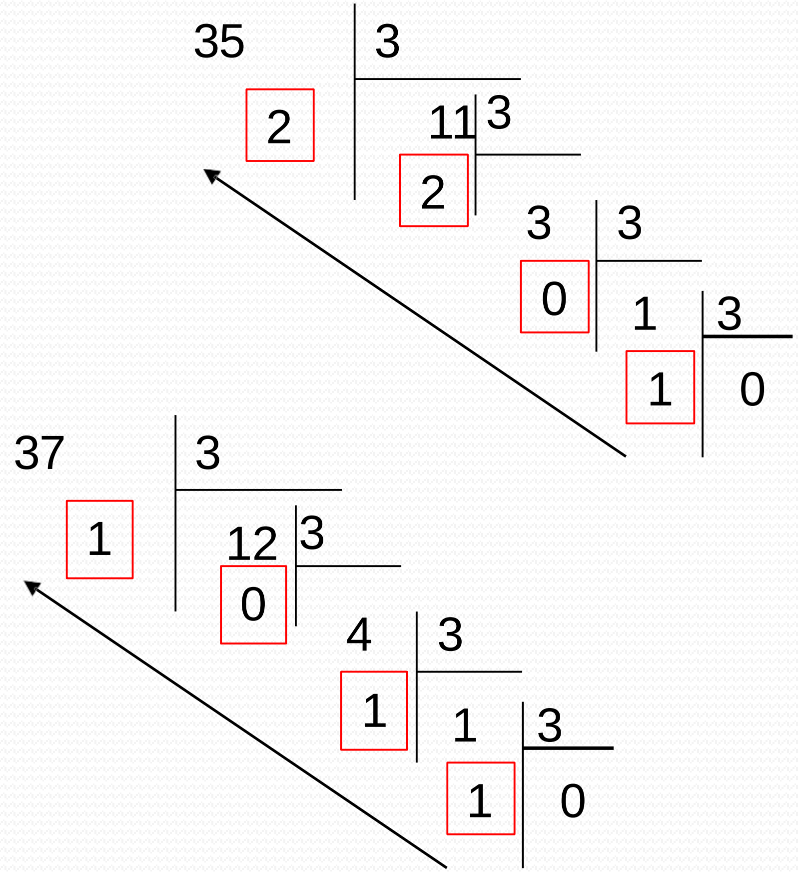
Conversion de la base 10 à la base 2:

- **Remarque :** Parfois en multipliant la partie décimale par la Base B, on n’arrive pas à convertir toute la partie entière. Ceci est dû essentiellement au fait que le nombre à convertir n’a pas un équivalent exact dans la Base B et sa partie décimale est cyclique.
- 0,1 5*2=0,3
- 0,3*2=0,6
- 0,6*2=1,2
- 0,2*2=0,4
- 0,4*2=0,8
- 0,8*2=1,6
- 0,6*2=1,2
- Le résultat est donc : (0,15)10= (0, 0010011001...)2. On dit (0,15)10 est cyclique dans la Base 2 de période 1001.
- Remarque : après plusieurs multiplication, on arrête les calculs.

Conversion du décimal à une base X

- La conversion se fait en prenant les restes des divisions successives sur la base X dans le sens inverse.

Exemple : effectuer les conversions suivantes:



$(35)_{10} = (?)_3$

$(37)_{10} = (?)_3$

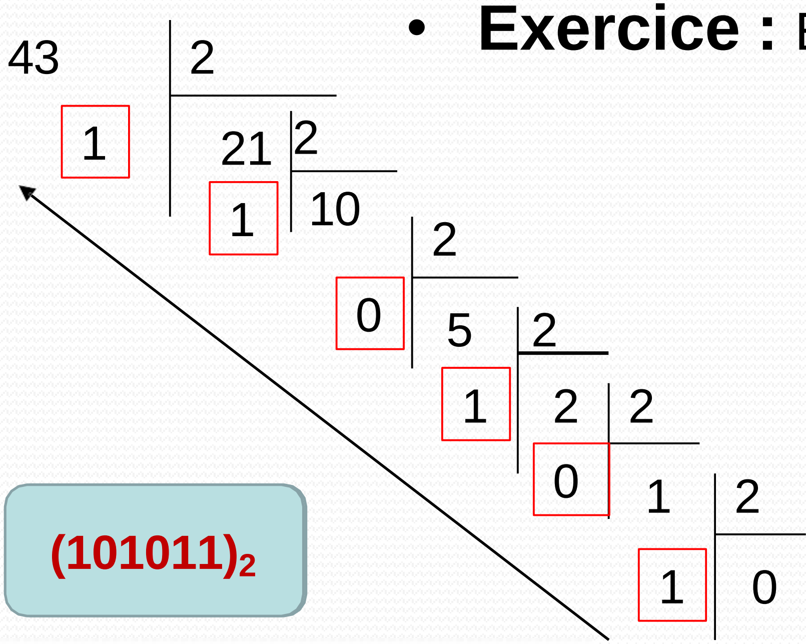
$(35)_{10} = (1022)_3$

$(37)_{10} = (1101)_3$

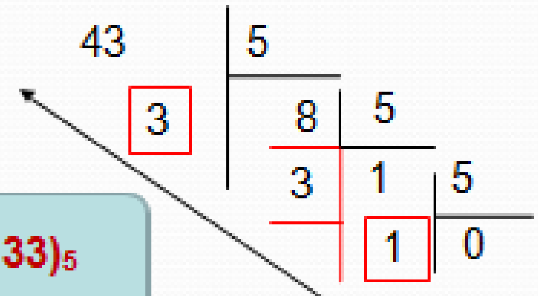
Conversion du décimal à une base X

- **Exercice :** Effectuer les transformations suivantes :

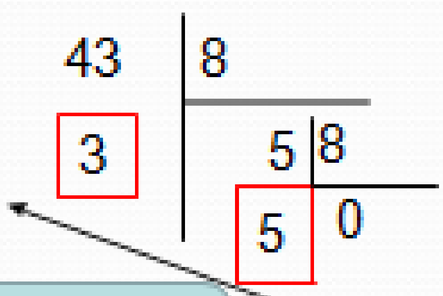
$$(43)_{10}=(?)_2=(?)_5=(?)_8=(?)_{16}$$



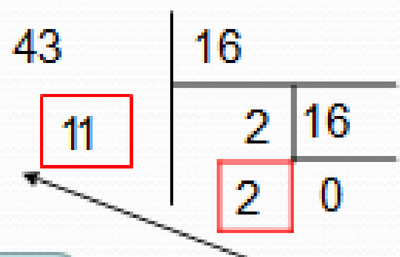
$(101011)_2$



$(133)_5$



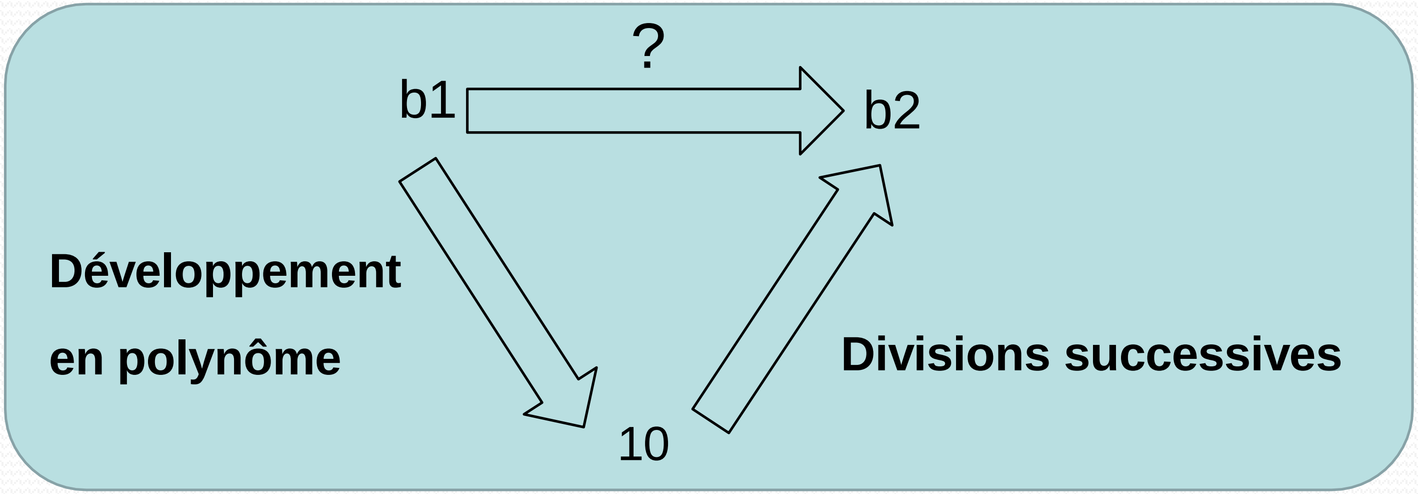
$(53)_8$



$(2B)_{16}$

Conversion d'une base b1 à une base b2

- Pour passer d'une **base b1** à une autre **base b2** directement (généralement il n'existe pas une méthode!!)
- L'idée est de convertir le nombre de la **base b1 à** la **base 10**, en suite convertir le résultat de la **base 10** à la **base b2**.

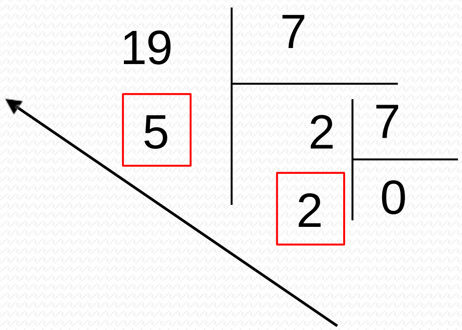


Conversion d'une base b1 à une base b2

Exercice : Effectuer la conversion suivante

$(34)_5 = (?)_7$

$$(34)_5 = 3*5^1 + 4*5^0 = 15 + 4 = (19)_{10} = (?)_7$$



$(34)_5$ $= (19)_{10} = (25)_7$ → $(34)_5 = (25)_7$

Conversion octal → binaire:

- En octal, chaque symbole de la base s'écrit **sur 3 bits en binaire**. L'idée de base est de remplacer chaque symbole dans la base octale par sa valeur en binaire sur **3 bits** **(faire des éclatements sur 3 bits).**
  - **Exemples :** $(345)_8$=?; $(65,76)_8$=?; $(35,34)_8$=?

Octal	Binaire
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

- $(345)_8 = (\underline{011}\ \underline{100}\ \underline{101})_2$
- $(65,76)_8 = (\underline{110}\ \underline{101},\ \underline{111}\ \underline{110})_2$
- $(35,34)_8 = (\underline{011}\ \underline{101}\ ,\ \underline{011}\ \underline{100})_2$

- **Remarque :** le remplacement se fait de droite à gauche pour la partie entière et de gauche à droite pour la partie fractionnelle .

Conversion binaire → octal:

- L'idée de base est de faire des **regroupements** de **3 bits** à partir du poids faible. Par la suite **remplacer** chaque regroupement par la valeur octal correspondante.
- **Exemples :** $(11001010010110)_2$ =?; $(110010100,10101)_2$ =?

$(11001010010110)_2 = (\underline{011}\ \underline{001}\ \underline{010}\ \underline{010}\ \underline{110})_2 = \mathbf{(31226)_8}$

$(110010100,10101)_2 = (\underline{110}\ \underline{010}\ \underline{100}\ ,\ \underline{101}\ 0\underline{10})_2 = \mathbf{(624,52)_8}$

<--- ---->

- **Remarque :**
- Le regroupement se fait de droite à gauche pour la partie entière et de gauche à droite pour la partie fractionnelle .

Conversion hexadécimal → binaire:

- En Hexadécimal, chaque symbole de la base s'écrit **sur 4 bits**. L'idée de base est de **remplacer** chaque symbole par sa valeur en binaire sur 4 bits (faire des éclatement sur 4 bits).
  - **Exemples :** $(757F)_{16}$=? ; $(BA3,5F7)_{16}$ =?

$(757F)_{16} = (\underline{0111}\ \underline{0101}\ \underline{0111}\ \underline{1111})_2$

$(BA3,5F7)_{16} = (\ \underline{1011}\ \underline{1010}\ \underline{0011}\ ,\ \underline{0101}\ \underline{1111}\ \underline{0111}\ )_2$

Hexa	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Conversion binaire → hexadécimal:

- L'idée de base est de faire des **regroupements** de **4 bits** à partir du poids faible. Par la suite **remplacer** chaque regroupement par la valeur Hexadécimal correspondante .

- **Exemples :**
- $(11001010100110)_2$=?
- $(110010100,10101)_2$ = ?

$(11001010100110)_2=(\underline{0011}\ \underline{0010}\ \underline{1010}\ \underline{0110})_2=(32A6)_{16}$

$(110010100,10101)_2 = (\underline{0001}\ \underline{1001}\ \underline{0100},\underline{1010}\ \underline{1000})_2 =(194,A8)_{16}$

Conversion d'une base B1 à une base B2

- Les deux bases sont des puissances de 2 (base 8 et 16)
- On utilise la base 2 comme base intermédiaire
  - Base B1 → Base 2 → BaseB2
- Les deux bases <u>ne sont pas</u> des puissances de 2
- On utilise la base 10 comme base intermédiaire
  - Base B1 → Base 10 →Base B2

Les opération arithmétiques

Les opération arithmétiques

- **Principe général:**

- Les règles des opérations arithmétiques en décimal sont aussi valables pour les opérations arithmétiques dans n'importe quelle base.

Addition binaire

- Pour additionner deux nombres binaires, on procède exactement comme en décimale, mais en prenant en compte la table d'addition élémentaire suivante :

- 0+0 = 0 retenue 0
- 0+1 = 1 + 0 = 1 retenue 0
- 1 + 1 = 0 retenue 1
- 1 + 1 + 1 = 1 retenue 1

<u>Exemple</u> :

		En binaire									En décimal		
Retenus→		*1*	*1*	*1*	*1*	*1*	*1*	*1*				*1*	
			1	1	1	0	0	1	1		1	1	5
	+				1	1	1	0	1	+		2	9
Résultat→	=	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	=	**1**	**4**	**4**

Addition binaire

- **Exemple:**

a	*b*	*S*	*R*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

S : la somme
R : la retenue

```
          1  1
   1 1 0 0 0 1 1
+ 1 0 0 0 1 0 1 1
------------------
  1 1 1 0 1 1 1 0
```

Opérations arithmétiques en binaire

- **Exercice:** Effectuer l'opération suivante

$(1100011)_2$ +$(10001011)_2$ =$(?)_2$:

$$\begin{array}{r} 0 \\ +\;0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ +\;1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\;0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\;1 \\ \hline 10 \end{array}$$

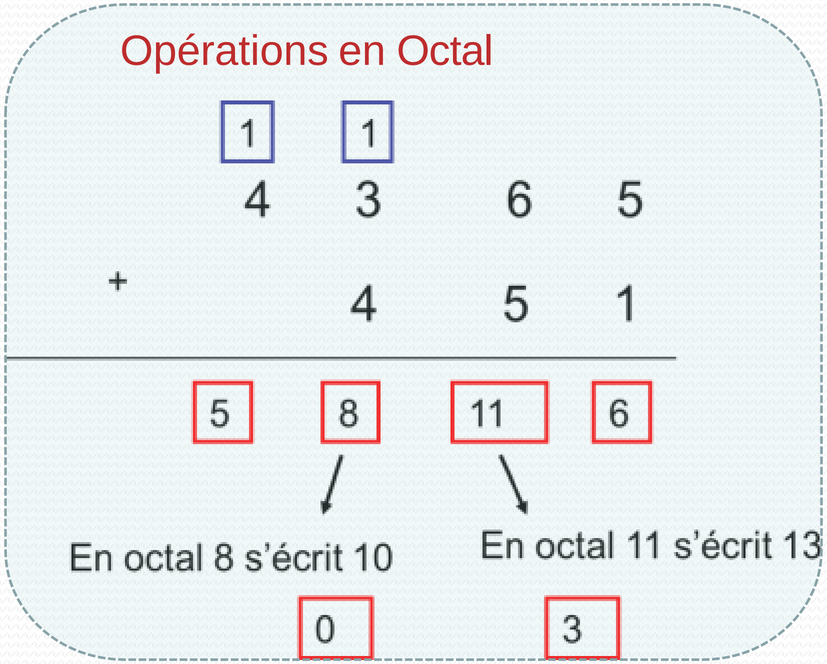
Le résultat : **$(11101110)_2$**

$$\begin{array}{rcccccccc} & & & & & 1 & & 1 & \\ & & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ + & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ \hline & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{array}$$

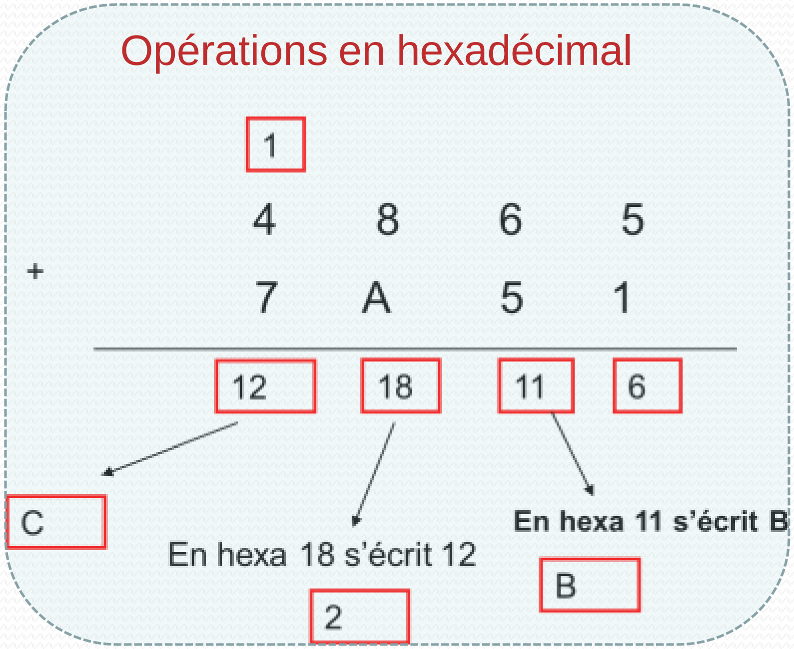
Opérations arithmétiques en octal

Exercice: Effectuer les opérations suivantes:

$(4365)_8 + (451)_8 = (?)_8$ **$(4865)_{16} + (7A51)_{16} = (?)_{16}$**



Le résultat : **$(5036)_8$**



Le résultat : **$(C2B6)_{16}$**

Soustraction binaire

- Dans la soustraction binaire, quand la quantité a soustraire est supérieure a la quantité dont on soustrait, on emprunte 1 au voisin de gauche. En binaire, ce 1 ajoute 2 a la quantité dont on soustrait, tandis qu'en décimal il ajoute 10.

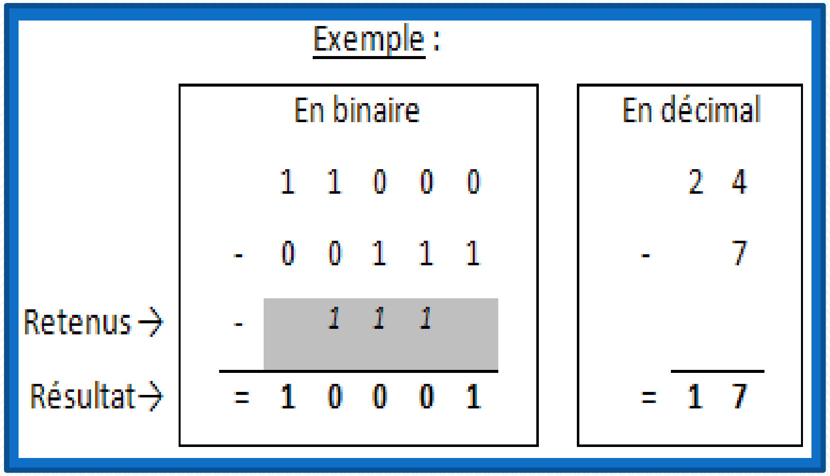
Table de soustraction binaire :

- 0 - 0 = 0 retenue 0
- 1 - 0 = 1 retenue 0
- 0 - 1 = 1 retenue 1 a soustraite

 au chiffre voisin de gauche

- 0 - 1 - 1 = 0 retenue 1 a soustraite

Au chiffre voisin de gauche



Exemple :

	En binaire	En décimal
	1 1 0 0 0	2 4
	- 0 0 1 1 1	- 7
Retenus →	- - - -	
Résultat →	=	=

Exemple :

		En binaire				
		1	1	0	0	$^{1}0$
	-	0	0	1	1	1
Retenus →	-		1	1	1	
Résultat →	=	1	0	0	0	1

	En décimal	
	2	4
-		7
=	1	7

Exemple :

En binaire

						2-1
		1	1	0	0	1 0
	-	0	0	1	1	1
Retenus →	-		1	1	1	
Résultat →	=	1	0	0	0	1

En décimal

	2	4
-		7
=	1	7

Exemple :

1-0 1-1 2-1-1 2-1-1 2-1

	En binaire	En décimal
	1 1 0 0 0	2 4
	- 0 0 1 1 1	- 7
Retenus →	- 1 1 1	
Résultat→	= 1 0 0 0 1	= 1 7

Soustraction binaire

- Exemple

a	*b*	*D*	*E*
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

D : Différence

E : Empreinte

$$
\begin{array}{rccccccccccc}
 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
- & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
\hline
 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1
\end{array}
$$

Multiplication binaire

- La multiplication binaire se réalise comme une multiplication décimale.

Voici les règles de calcul a utiliser :

- 0 x 0 = 0
- 0 x 1 = 0
- 1 x 0 = 0
- 1 x 1 = 1

Elle consiste à faire une suite d'additions avec le multiplicande décalé vers la gauche. Cette opération est répétée autant de fois qu'il y a d'éléments binaires (à 1) dans le multiplicateur.

- *Remarque : Lorsqu'une opération donne plus de deux produits partiels*, effectuez la somme de ces derniers 2 a 2 pour diminuer le risque d'erreur.

Exemple :

1101	multiplicande
x 1011	multiplicateur
0001101	
+ 001101	décalage 1 pas
+ 1101	décalage 3 pas
10001111	**résultat**

Multiplication binaire

a	*b*	*c*
0	0	0
0	1	0
1	0	0
1	1	1

c = a . b

- Exemple

```
          1 0 1 1 0 0 1 0 1
x                   1 0 1 1
          -------------------
          1 0 1 1 0 0 1 0 1
        1 0 1 1 0 0 1 0 1
      0 0 0 0 0 0 0 0 0
    1 0 1 1 0 0 1 0 1
    -------------------------
    1 1 1 1 0 1 0 1 0 1 1 1
```

Multiplication binaire

- Exemple

						1	0	1	0
x				1	0	1	0	1	0
						0	0	0	0
+				1	1	0	1	0	0
+				0	0	0	0	0	0
+		1	1	0	1	0	0	0	0
+		0	0	0	0	0	0	0	0
+	1	0	1	0	0	0	0	0	0
=	1	1	0	1	0	0	1	0	0

Division binaire

- La division binaire s'effectue a l'aide de soustractions
et de décalages, comme la division décimale, sauf que les chiffres du quotient ne peuvent être que 1 ou 0. Le bit du quotient est 1 si on peut soustraire le diviseur, sinon il est 0.

Division décimale	
1 6 5	1 1
- 1 1	1 5
5 5	
- 5 5	
0	

Division binaire	
1 0 1 0 0 1 0 1	1 0 1 1
- 0 0 0 0	0 1 1 1 1
1 0 1 0 0	
- 1 0 1 1	
1 0 0 1 1	
- 1 0 1 1	
1 0 0 0 0	
- 1 0 1 1	
0 1 0 1 1	
- 1 0 1 1	
0 0 0 0	

Division binaire

Exemple

```
  1 1 1 0 0 1 1 1          | 1 0 1
- 1 0 1                    |------------
  --------                 | 1 0 1 1 1 0
  0 1 0 0 0
-     1 0 1
    --------
    0 0 1 1 1
-         1 0 1
      ---------
          0 1 0 1
-           1 0 1
        ----------
            0 0 0 1
-             0 0 0
          ----------
              0 0 1
```

111100111 : 101 = 101110   reste 1.

Division binaire

Exemple

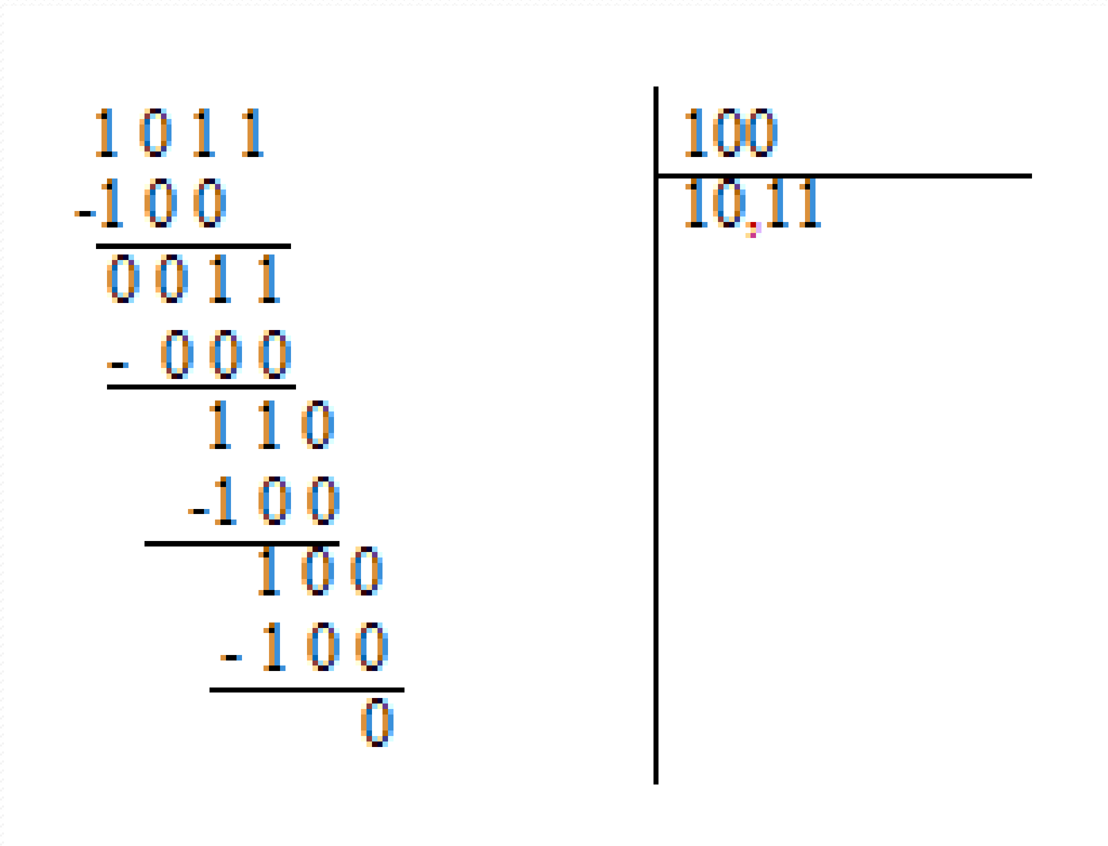
$$
\begin{array}{ccccc|ccc}
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ \cline{6-8}
1 & 0 & 1 & & & 1 & 0 & 1 \\ \cline{1-3}
 & 1 & 0 & 0 & 1 & & & \\
 & & 1 & 0 & 1 & & & \\ \cline{2-5}
 & & 1 & 0 & 0 & & &
\end{array}
$$

Il suffit en fait de soustraire 101 lorsqu'on le peut, et d'abaisser le chiffre suivant :

$11101 = 101 \times 101 + 100$

Division binaire

Exemple



Les opérations arithmétiques en Octal :

- **L’addition**
- Comme pour le système binaire, on applique les mêmes règles pour les nombres octaux. Toutefois, dans ce cas, on aura la retenue **«1» à gauche à chaque fois que la somme dépasse la valeur 7 car 7(8)+1(8)=10(8).**
- Exemple d'addition en base octale:

$$\begin{array}{cccc}
 & {}^{1}1 & {}^{1}2 & 3_{(8)} \\
+ & 4 & 5 & 7_{(8)} \\
\hline
 & 6 & 0 & 2_{(8)}
\end{array}$$

Les opérations arithmétiques en Octal :

La soustraction

- Comme en décimal mais en se limitant à 7 (octal).
- Exemple:

$$
\begin{array}{cccc}
 & {}^{1}1 & {}^{1}2 & {}_{1}3_{(8)} \\
- & {}_{1}4 & {}_{1}5 & 7_{(8)} \\
\hline
 & 4 & 4 & 4_{(8)}
\end{array}
$$

Les opérations arithmétiques en Hexadécimal

- **L’addition**
- Comme en décimal, l'addition s'effectue chiffre par chiffre. Toutefois, dans ce cas, on aura la retenue **«1» à gauche à chaque fois que la somme dépasse la valeur F car F16)+116)=10(16).**

$$
\begin{array}{cccc}
 & 1 & {}^{1}2 & A_{(16)} \\
+ & E & 5 & 7_{(16)} \\
\hline
 & F & 8 & 1_{(16)}
\end{array}
$$

Les opérations arithmétiques en Hexadécimal

- **La soustraction en hexadécimal:**
- **Exemple:**

$$
\begin{array}{cccc}
 & \mathbf{F} & {}^{1}\mathbf{2} & \mathbf{A}_{(16)} \\
- & {}_{1}\mathbf{E} & \mathbf{5} & \mathbf{7}_{(16)} \\
\hline
 & \mathbf{0} & \mathbf{D} & \mathbf{3}_{(16)}
\end{array}
$$