Représentation des nombres en machine

Cours BTS SIO 2ème année

Année 2025/2026

Plan du cours

Représentation d'un entier relatif

Addition binaire

- L'addition binaire répond aux règles suivantes :

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0 + 0 = 0 \qquad 1 + 0 = 0 + 1 = 1 \qquad 1 + 1 = 10$$

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0+0=0 \qquad 1+0=0+1=1 \qquad 1+1=10$$

- Exemple (sans retenue) :

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

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0 + 0 = 0 \qquad 1 + 0 = 0 + 1 = 1 \qquad 1 + 1 = 10$$

- Exemple (sans retenue) :

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

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0+0=0 \qquad 1+0=0+1=1 \qquad 1+1=10$$

- Exemple (sans retenue) :

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

- Exemple (avec retenue) :

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

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0+0=0 \qquad 1+0=0+1=1 \qquad 1+1=10$$

- Exemple (sans retenue) :

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

- Exemple (avec retenue) :

$$\begin{array}{ccccc} \color{red}{1} & & & \color{red}{1} & \\ & 1 & 1 & 0 & 1 \\ + & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$$

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0+0=0 \qquad 1+0=0+1=1 \qquad 1+1=10$$

- Exemple (sans retenue) :

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

- Exemple (avec retenue) :

$$\begin{array}{ccccc} \color{red}{1} & & & \color{red}{1} & \\ & 1 & 1 & 0 & 1 \\ + & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$$

☞ *Exercice 1*

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

- Exemple (sur 4 bits) :

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

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

- Exemple (sur 4 bits) :

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

- Le 5ème bit n'est pas pris en compte

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

- Exemple (sur 4 bits) :

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

- Le 5ème bit n'est pas pris en compte
- Équivalent à : $13 + 9 = 6$ (!)

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

- Exemple (sur 4 bits) :

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

- Le 5ème bit n'est pas pris en compte
- Équivalent à : $13 + 9 = 6$ (!)

☞ *Exercice 2*

Cas particulier

- Exemple (sur 8 bits) : $\begin{array}{ccccccccc} & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ + & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ \hline \end{array}$

Cas particulier

- Exemple (sur 8 bits) :

	0	0	1	0	0	1	0	1
+	1	1	0	1	1	0	1	1
1	0	0	0	0	0	0	0	0

Cas particulier

- Exemple (sur 8 bits) :
$$\begin{array}{ccccccccc} & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ + & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$
- Équivalent à $37 + 219 = 0$

Cas particulier

- Exemple (sur 8 bits) :

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

- Équivalent à $37 + 219 = 0$
- 219 équivaut donc à -37 (car $37 + (-37) = 0$)...

Cas particulier

- Exemple (sur 8 bits) : $\begin{array}{ccccccccc} & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ + & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$
- Équivalent à 37 + 219 = 0
- 219 équivaut donc à -37 (car 37 + (-37) = 0)...
- ... ou bien 37 équivaut à -219 (car (-219) + 219 = 0)

Cas particulier

- Exemple (sur 8 bits) :

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

- Équivalent à $37 + 219 = 0$
- 219 équivaut donc à -37 (car $37 + (-37) = 0$)...
- ... ou bien 37 équivaut à -219 (car $(-219) + 219 = 0$)

Règle

Dans la représentation des entiers relatifs, le bit de poids fort est un ***bit de signe*** *:*

- *0 indique un entier positif ou nul*
- *1 indique un entier négatif*

Cas particulier

- Exemple (sur 8 bits) :

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

- Équivalent à 37 + 219 = 0
- 219 équivaut donc à -37 (car 37 + (-37) = 0)...
- ... ou bien 37 équivaut à -219 (car (-219) + 219 = 0)

Règle

Dans la représentation des entiers relatifs, le bit de poids fort est un ***bit de signe*** *:*

- *0 indique un entier positif ou nul*
- *1 indique un entier négatif*

☞ *D'après la règle, c'est donc 219 qui équivaut à -37*

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais 37 + 219 = 256 = 2^8
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais 37 + 219 = 256 = 2^8
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais 37 + 219 = 256 = 2^8
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

- Exemple : opposé de 55 (sur 8 bits)

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais 37 + 219 = 256 = 2^8
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

- Exemple : opposé de 55 (sur 8 bits)
  - On calcule $2^8 - 55 = 256 - 55 = 201$

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais $37 + 219 = 256 = 2^8$
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

- Exemple : opposé de 55 (sur 8 bits)
  - On calcule $2^8 - 55 = 256 - 55 = 201$
  - En binaire, $201 = (1100\ 1001)_2$

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais $37 + 219 = 256 = 2^8$
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

- Exemple : opposé de 55 (sur 8 bits)
  - On calcule $2^8 - 55 = 256 - 55 = 201$
  - En binaire, $201 = (1100\ 1001)_2$
  - -55 est donc représenté par $(1100\ 1001)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$
  - On inverse tous les bits : $(1100\ 1000)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$
  - On inverse tous les bits : $(1100\ 1000)_2$
  - On ajoute 1 : $(1100\ 1001)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$
  - On inverse tous les bits : $(1100\ 1000)_2$
  - On ajoute 1 : $(1100\ 1001)_2$
- -55 est donc représenté par $(1100\ 1001)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$
  - On inverse tous les bits : $(1100\ 1000)_2$
  - On ajoute 1 : $(1100\ 1001)_2$
- -55 est donc représenté par $(1100\ 1001)_2$

☞ *Exercices 3,4,5,6*

Représentation d'un nombre à virgule

- En base 10, on distingue deux types de nombres à virgule :

Nombres dyadiques

- En base 10, on distingue deux types de nombres à virgule :
  - ceux qui ont un nombre **fini** de chiffres après la virgule, comme 1,618 ; on les appelle **nombres décimaux**

Nombres dyadiques

- En base 10, on distingue deux types de nombres à virgule :
  - ceux qui ont un nombre **fini** de chiffres après la virgule, comme 1,618 ; on les appelle **nombres décimaux**
  - ceux qui ont un nombre **infini** de chiffres après la virgule ; parmi eux, on trouve :
    - des **nombres rationnels** : comme $\frac{1}{3} = 0,333...$
    - des **nombres irrationnels** : comme $\sqrt{2} = 1,414213...$ ou $\pi = 3,14159265...$

Nombres dyadiques

- En base 10, on distingue deux types de nombres à virgule :
  - ceux qui ont un nombre **fini** de chiffres après la virgule, comme 1,618 ; on les appelle **nombres décimaux**
  - ceux qui ont un nombre **infini** de chiffres après la virgule ; parmi eux, on trouve :
    - des **nombres rationnels** : comme $\frac{1}{3} = 0,333...$
    - des **nombres irrationnels** : comme $\sqrt{2} = 1,414213...$ ou $\pi = 3,14159265...$
- En base 2, on appelle **nombres dyadiques** les nombres possédant un nombre fini de chiffres après la virgule, comme $10,001$.

Nombres dyadiques

- En base 10, on distingue deux types de nombres à virgule :
  - ceux qui ont un nombre **fini** de chiffres après la virgule, comme 1,618 ; on les appelle **nombres décimaux**
  - ceux qui ont un nombre **infini** de chiffres après la virgule ; parmi eux, on trouve :
    - des **nombres rationnels** : comme $\frac{1}{3} = 0,333...$
    - des **nombres irrationnels** : comme $\sqrt{2} = 1,414213...$ ou $\pi = 3,14159265...$
- En base 2, on appelle **nombres dyadiques** les nombres possédant un nombre fini de chiffres après la virgule, comme $10,001$.
- Que représentent de tels nombres ?

On décale la virgule...

- En numération décimale, diviser par 10^n revient à décaler la virgule de n rangs vers la gauche ; un nombre à virgule est donc le résultat d'un nombre entier divisé par une puissance de 10 :

$$123,45 = 12\ 345/10^2$$

$$7\ 392,136 = 7\ 392\ 136/10^3$$

On décale la virgule...

- En numération décimale, diviser par 10^n revient à décaler la virgule de n rangs vers la gauche ; un nombre à virgule est donc le résultat d'un nombre entier divisé par une puissance de 10 :

$$123,45 = 12\ 345/10^2$$

$$7\ 392,136 = 7\ 392\ 136/10^3$$

- En binaire, diviser par 2^n revient à décaler la virgule de n rangs vers la gauche ; un nombre à virgule est un nombre entier divisé par une **puissance de 2** :

$$(101,01)_2 = (1\ 0101)_2/2^2$$

$$(1001,1010)_2 = (1001\ 1010)_2/2^4$$

On décale la virgule...

- En numération décimale, diviser par 10^n revient à décaler la virgule de n rangs vers la gauche ; un nombre à virgule est donc le résultat d'un nombre entier divisé par une puissance de 10 :

$$123,45 = 12\ 345/10^2$$

$$7\ 392,136 = 7\ 392\ 136/10^3$$

- En binaire, diviser par 2^n revient à décaler la virgule de n rangs vers la gauche ; un nombre à virgule est un nombre entier divisé par une **puissance de 2** :

$$(101,01)_2 = (1\ 0101)_2/2^2 = 21/4 = \boxed{5,25}$$

$$(1001,1010)_2 = (1001\ 1010)_2/2^4 = 154/16 = \boxed{9,625}$$

On décale la virgule...

- En numération décimale, diviser par 10^n revient à décaler la virgule de n rangs vers la gauche ; un nombre à virgule est donc le résultat d'un nombre entier divisé par une puissance de 10 :

$$123,45 = 12\ 345/10^2$$

$$7\ 392,136 = 7\ 392\ 136/10^3$$

- En binaire, diviser par 2^n revient à décaler la virgule de n rangs vers la gauche ; un nombre à virgule est un nombre entier divisé par une **puissance de 2** :

$$(101,01)_2 = (1\ 0101)_2/2^2 = 21/4 = \boxed{5,25}$$

$$(1001,1010)_2 = (1001\ 1010)_2/2^4 = 154/16 = \boxed{9,625}$$

- Chaque bit après la virgule représente l'**inverse d'une puissance de 2**

On décale la virgule...

- En numération décimale, diviser par 10^n revient à décaler la virgule de n rangs vers la gauche ; un nombre à virgule est donc le résultat d'un nombre entier divisé par une puissance de 10 :

$$123,45 = 12\ 345/10^2$$

$$7\ 392,136 = 7\ 392\ 136/10^3$$

- En binaire, diviser par 2^n revient à décaler la virgule de n rangs vers la gauche ; un nombre à virgule est un nombre entier divisé par une **puissance de 2** :

$$(101,01)_2 = (1\ 0101)_2/2^2 = 21/4 = \boxed{5,25}$$

$$(1001,1010)_2 = (1001\ 1010)_2/2^4 = 154/16 = \boxed{9,625}$$

- Chaque bit après la virgule représente l'**inverse d'une puissance de 2**

☞ *Exercices 7,8*

Du décimal au binaire

Pour convertir un nombre à virgule de la base 10 à la base 2, on utilise l'algorithme suivant :

Algorithme

Du décimal au binaire

Pour convertir un nombre à virgule de la base 10 à la base 2, on utilise l'algorithme suivant :

Algorithme

- Séparer la **partie entière** et la **partie décimale** du nombre considéré

Du décimal au binaire

Pour convertir un nombre à virgule de la base 10 à la base 2, on utilise l'algorithme suivant :

Algorithme

- Séparer la **partie entière** et la **partie décimale** du nombre considéré
- Convertir la partie entière en binaire

Du décimal au binaire

Pour convertir un nombre à virgule de la base 10 à la base 2, on utilise l'algorithme suivant :

Algorithme

- Séparer la **partie entière** et la **partie décimale** du nombre considéré
- Convertir la partie entière en binaire
- Pour la partie décimale :
    - Multiplier la partie décimale par 2
    - Si le résultat est inférieur à 1, écrire un 0, puis recommencer
    - Si le résultat est supérieur à 1, écrire un 1, soustraire 1 à la partie décimale, puis recommencer
    - Si le résultat est égal à 1, écrire un 1, puis arrêter

Du décimal au binaire

- Exemple : 12,25

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: $\mathbf{0}$

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: **0**
    - $0,5 \times 2 \rightarrow 1$: 0**1**

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \to 0,5 < 1$: **0**
    - $0,5 \times 2 \to 1$: 0**1**
  - $\boxed{12{,}25 = (1100{,}01)_2}$

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: **0**
    - $0,5 \times 2 \rightarrow 1$: 0**1**
  - $\boxed{12{,}25 = (1100{,}01)_2}$

- Exemple : 37,5625

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: **0**
    - $0,5 \times 2 \rightarrow 1$: 0**1**
  - $\boxed{12{,}25 = (1100{,}01)_2}$

- Exemple : 37,5625
  - Partie entière : $37 = (10\ 0101)_2$

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: **0**
    - $0,5 \times 2 \rightarrow 1$: 0**1**
  - $\boxed{12{,}25 = (1100{,}01)_2}$

- Exemple : 37,5625
  - Partie entière : $37 = (10\ 0101)_2$
  - Partie décimale : 0,5625

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: $\mathbf{0}$
    - $0,5 \times 2 \rightarrow 1$: $0\mathbf{1}$
  - $\boxed{12{,}25 = (1100{,}01)_2}$

- Exemple : 37,5625
  - Partie entière : $37 = (10\ 0101)_2$
  - Partie décimale : 0,5625
    - $0,5625 \times 2 \rightarrow 1,125 > 1$: $\mathbf{1}$
    - On soustrait 1 : $1,125 \rightarrow 0,125$

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: **0**
    - $0,5 \times 2 \rightarrow 1$: 0**1**
  - $\boxed{12{,}25 = (1100{,}01)_2}$

- Exemple : 37,5625
  - Partie entière : $37 = (10\ 0101)_2$
  - Partie décimale : 0,5625
    - $0,5625 \times 2 \rightarrow 1,125 > 1$: **1**
    - On soustrait 1 : $1,125 \rightarrow 0,125$
    - $0,125 \times 2 \rightarrow 0,25 < 1$: 1**0**

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: **0**
    - $0,5 \times 2 \rightarrow 1$: 0**1**
  - $\boxed{12{,}25 = (1100{,}01)_2}$

- Exemple : 37,5625
  - Partie entière : $37 = (10\ 0101)_2$
  - Partie décimale : 0,5625
    - $0,5625 \times 2 \rightarrow 1,125 > 1$: **1**
    - On soustrait 1 : $1,125 \rightarrow 0,125$
    - $0,125 \times 2 \rightarrow 0,25 < 1$: 1**0**
    - $0,25 \times 2 \rightarrow 0,5 < 1$: 10**0**

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: **0**
    - $0,5 \times 2 \rightarrow 1$: 0**1**
  - $\boxed{12{,}25 = (1100{,}01)_2}$

- Exemple : 37,5625
  - Partie entière : $37 = (10\ 0101)_2$
  - Partie décimale : 0,5625
    - $0,5625 \times 2 \rightarrow 1,125 > 1$: **1**
    - On soustrait 1 : $1,125 \rightarrow 0,125$
    - $0,125 \times 2 \rightarrow 0,25 < 1$: 1**0**
    - $0,25 \times 2 \rightarrow 0,5 < 1$: 10**0**
    - $0,5 \times 2 \rightarrow 1$: 100**1**

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: **0**
    - $0,5 \times 2 \rightarrow 1$: 0**1**
  - $\boxed{12{,}25 = (1100{,}01)_2}$

- Exemple : 37,5625
  - Partie entière : $37 = (10\ 0101)_2$
  - Partie décimale : 0,5625
    - $0,5625 \times 2 \rightarrow 1,125 > 1$: **1**
    - On soustrait 1 : $1,125 \rightarrow 0,125$
    - $0,125 \times 2 \rightarrow 0,25 < 1$: 1**0**
    - $0,25 \times 2 \rightarrow 0,5 < 1$: 10**0**
    - $0,5 \times 2 \rightarrow 1$: 100**1**
  - $\boxed{37{,}5625 = (10\ 0101{,}1001)_2}$

Du décimal au binaire

- Exemple : 12,25
  - Partie entière : $12 = (1100)_2$
  - Partie décimale : 0,25
    - $0,25 \times 2 \rightarrow 0,5 < 1$: **0**
    - $0,5 \times 2 \rightarrow 1$: 0**1**
  - $\boxed{12{,}25 = (1100{,}01)_2}$

- Exemple : 37,5625
  - Partie entière : $37 = (10\ 0101)_2$
  - Partie décimale : 0,5625
    - $0,5625 \times 2 \rightarrow 1,125 > 1$: **1**
    - On soustrait 1 : $1,125 \rightarrow 0,125$
    - $0,125 \times 2 \rightarrow 0,25 < 1$: 1**0**
    - $0,25 \times 2 \rightarrow 0,5 < 1$: 10**0**
    - $0,5 \times 2 \rightarrow 1$: 100**1**
  - $\boxed{37{,}5625 = (10\ 0101{,}1001)_2}$

☞ *Exercices 9,10*