INFORMATIQUE DU BÂTIMENT

Mathématiques 1er semestre

Systèmes de numération

Table des matières

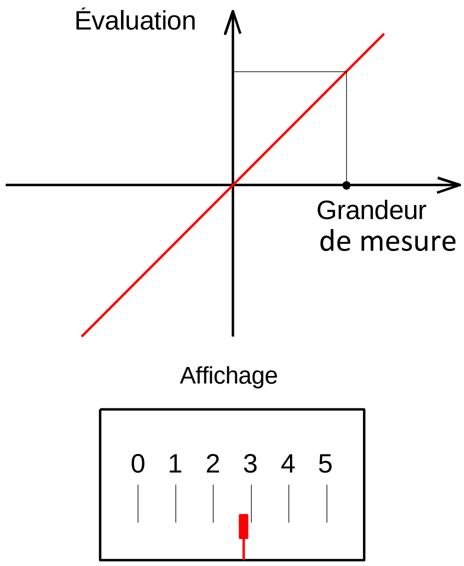
1 Introduction

1.1 Signaux analogiques et numériques

Un **signal** est une grandeur variable dans le temps (p. ex. une tension électrique) qui porte une **information** qui lui est propre.

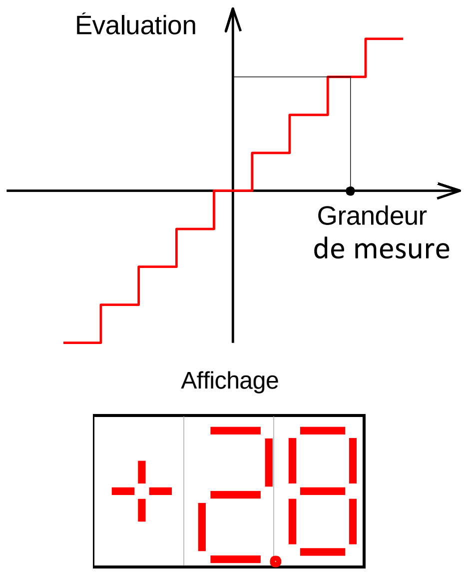
Comparaison:

Signal analogique



Les signaux analogiques peuvent prendre **n'importe quelle** valeur dans une certaine plage.

Signal numérique



Les signaux numériques peuvent uniquement varier de manière discrète, autrement dit, **quantifiée**.

1.2 Signaux binaires

Un signal binaire est un signal numérique qui n'est reproduit que sur 2 valeurs (niveau logique).

- 0 logique ou LO (engl. low)
- 1 logique ou HI (engl. high)

Un interrupteur par exemple est un élément binaire. Dans le cas le plus simple, il ne peut adopter que 2 états. Nous pouvons par exemple procéder à l'attribution suivante:

Interrupteur ouvert → 0 logique

Interrupteur fermé → 1 logique

Les signaux binaires présentent entre autres les avantages suivants:

- haute résistance aux interférences et reproductibilité
- possibilité de détecter/corriger les erreurs

2 Systèmes de numération

2.1 Système décimal:

Nous calculons généralement dans le *système décimal* ou *système de base 10*. La valeur de chaque chiffre dépend de sa position. On parle alors d'un **système de position**.

Exemple: *MSD*[1] *LSD*

2573_{10}	2	5	7	3	← suite de chiffres
Base 10	10^3	10^2	10^1	10^0	← valeur

$2573_{10} = 2\cdot10^3 + 5\cdot10^2 + 7\cdot10^1 + 3\cdot10^0$

$2000 + 500 + 70 + 3 = \underline{\underline{2573_{10}}}$

2.2 Système binaire:

Le *système binaire* est un système de numération à 2 valeurs. Il ne comprend que les chiffres 0 et 1.

Exemple: *MSB*[2] *LSB*

1101_2	1	1	0	1	← suite de chiffres
Base 2	2^3	2^2	2^1	2^0	← valeur

$1101_2 = 1\cdot2^3 + 1\cdot2^2 + 0\cdot2^1 + 1\cdot2^0$

$8 + 4 + 0 + 1 = \underline{\underline{13_{10}}}$

- *Les systèmes de numération* construits selon le système de position possèdent une réserve de chiffres correspondant à la valeur de la base (p.ex. système binaire = 2 chiffres).
- La valeur du chiffre le plus élevé est inférieure de 1 à la base.
- La *valeur* du voisin de gauche d'un chiffre est plus grande du facteur de la base, celle du voisin de droite est plus petite du même facteur.

[1] *MSD, LSD* engl. most significant, least significant digit (chiffre le plus ou le moins significatif)

[2] *MSB, LSB* engl. most significant, least significant bit (bit le plus ou le moins significatif)

2.3 Système octal et hexadécimal

Le *système binaire* est idéal pour un ordinateur mais pour nous, plutôt inconfortable. Pour simplifier l'utilisation, nous regroupons donc dans le *système octal*, 3 chiffres binaires pour former un **chiffre octal** ou dans le *système hexadécimal*, nous regroupons 4 chiffres binaires pour former un **chiffre HEX**.

	Base	Réserve de chiffres
Système octal	8	0 .. 7
Système HEX[3]	16	0 .. 9, A .. F

Exemple: donné: chiffre binaire = 1010101001_2
recherché: a) chiffre octal
b) chiffre HEX

a) binaire → octal

001	010	101	001_2	Groupes de 3
	$\boxed{2^2}$	$\boxed{2^1}$	$\boxed{2^0}$	Pondération
1	2	5	1	Chiffre octal

b) binaire → HEX

0010	1010	1001_2	Groupes de 4
$\boxed{2^3}$ $\boxed{2^2}$ $\boxed{2^1}$ $\boxed{2^0}$			Pondération
2	10 → A	9	Chiffre HEX

Essai:

a) $1251_8 = 1\cdot8^3 + 2\cdot8^2 + 5\cdot8^1 + 1\cdot8^0 \quad = \underline{681_{10}}$

b) $2A9_{16} = 2\cdot16^2 + 10\cdot16^1 + 9\cdot16^0 \quad = \underline{681_{10}}$

[3] les chiffres HEX sont souvent désignés par un H placé à la suite, p. ex: 0FH

2.4 Comparaison des systèmes de numération

Exercice: Complétez le tableau suivant!

Décimal	Binaire	Octal	HEX
0	00000	00	00
1	00001	01	01
2	00010	02	02
3	00011	03	03
4	00100	04	04
5	00101	05	05
6	00110	06	06
7	00111	07	07
8	01000	10	08
9	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F

3 Conversion de nombres

3.1 Décimal ← → Binaire

Les nombres décimaux sont transformés en *nombres binaires* par une division continue par la base 2.

Exemple: donné: 91_{10}
rech.: x_2

91	:	2 =	45	reste	1	← LSB
45	:	2 =	22	reste	1	
22	:	2 =	11	reste	0	
11	:	2 =	5	reste	1	
5	:	2 =	2	reste	1	
2	:	2 =	1	reste	0	
1	:	2 =	0	reste	1	← *MSB*

x2 = 101'1011

Exercice: Représenter le nombre 200_{10} en nombre binaire!

200	: 2	= 100	Reste	0
100	: 2	= 50	Reste	0
50	: 2	= 25	Reste	0
25	: 2	= 12	Reste	1
12	: 2	= 6	Reste	0
6	: 2	= 3	Reste	0
3	: 2	= 1	Reste	1
1	: 2	= 0	Reste	1

Solution $200_{10} = \underline{1100'1000_2}$

La conversion **binaire → décimal** s'obtient en additionnant les puissances de 2.

Exemple: 10110110_2 $1\cdot2^7 + 1\cdot2^5 + 1\cdot2^4 + 1\cdot2^2 + 1\cdot2^1 = \underline{182_{10}}$

3.2 Décimal ← → HEX

Pour la *conversion décimal-HEX* nous effectuons une division continue d'entiers en **base 16**. Les restes qui en résultent représentent les chiffres HEX.

Exemple: convertir le nombre 1000_{10} en un nombre hexadécimal!

1000	: 16 =	62	Reste	8
62	: 16 =	3	Reste	14
3	: 16 =	0	Reste	3

Résultat = $3E8_{16}$ ou: $3E8_H$

Exercices

1. Convertissez le nombre 520_{10} en format HEX!

520 : 16 = 32 reste 8

32 : 16 = 2 reste 0

2 : 16 = 0 reste 2

Solution: 520_{10} = 208H

2. Soit le nombre hexadécimal A52H. Quel est le nombre décimal correspondant?

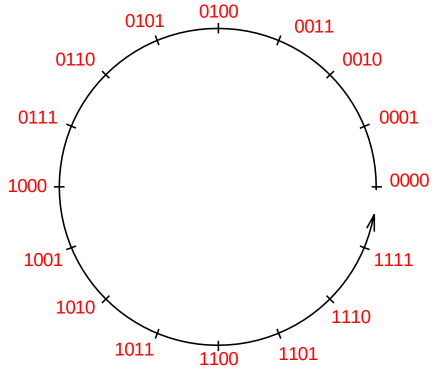
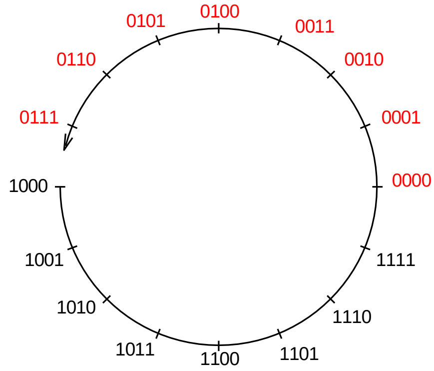
La conversion **HEX → Décimal** s'obtient par addition des puissances de 16:

$A52_H = 10 \cdot 16^2 + 5 \cdot 16^1 + 2 \cdot 16^0 =$

$2560 + 80 + 2 = 2642_{10}$

4 Calculer avec des nombres binaires

4.1 Représentation des nombres négatifs

Plage de nombres	Nombres positifs	Nombres positifs et négatifs
Arithmétique 4 bits		
Arithmétique 8 bits	255_{10} 1111'1111 FF_H 128_{10} 1000'0000 80_H 127_{10} 0111'1111 $7F_H$ 0_{10} 0000'0000 00_H	127_{10} 0111'1111 $7F_H$ 0_{10} 0000'0000 00_H -1_{10} 1111'1111 FF_H -128_{10} 1000'0000 80_H

→ dans la représentation des nombres positifs et négatifs,

le bit (MSB) indique le signe.

L'ordinateur gère les nombres négatifs de manière binaire en complément à deux.

Exemple: à quoi correspond le nombre **-2₁₀** eines d'un ordinateur 4 bits en complément à 2?

Z	0 0 1 0	chiffre positif
$\overline{Z}$	1 1 0 1	complément à 1
$-Z$	1 1 1 0	complément à 2

En général:

$$-Z = \overline{Z} + 1$$

compl. à 2 =compl. à 1 +1

Le *complément à 2* s'obtient en ajoutant 1 au *complément à 1.*

4.2 Addition et soustraction

Les nombres binaires s'additionnent ou se soustraient de la même manière que les nombres décimaux.

Règles de calcul

Opération	résultat	report
0 + 0	0	0
0 + 1	1	0
1 + 0	1	0
1 + 1	0	1
1 + 1 + 1	1	1

Opération	résultat	report
0 − 0	0	0
0 − 1	1	-1
1 − 0	1	0
1 − 1	0	0
0 − 1 − 1	0	-1
1 − 1 − 1	1	-1

Exemples d'addition et de soustraction directe

1.

	décimal	*binaire*
	11	1011
report (carry):	+ 19	+10011
résultat:	30	11110

2.

	décimal	*binaire*
	30	11110
	- 11	- 1011
report (borrow):		
résultat:	19	10011

3.

	décimal	*binaire*
	7	111
	- 8	- 1000
report (borrow):		
résultat:	-1	1 1111

4.3 Soustraction par addition du complément à 2

Les ordinateurs effectuent généralement une soustraction sous la forme d'une addition avec un nombre négatif:

$$Z1 - Z2 = Z1 + (-Z2)$$

chiffre négatif,
représentation en
complément à 2.

Exemple: Un ordinateur 8 bits doit effectuer la soustraction 10111 – 110.

Procédure:
1. *Complément à la largeur de bit de l'ordinateur, ici 8 chiffres*
2. *Formation du complément à 2 du soustracteur*
3. *Exécution de l'addition*

1. *complément pour atteindre 8 positions*

```
10111 → 00010111
  110 → 00000110
```

2. *Formation du complément à 2*

```
 00000110 (chiffre)
11111001 (compl. à 1)
      11111010 (compl. à 2 = compl. à 1+1)
```

3. *soustraction / addition*

	direct	*indirect*
	00010111	00010111
	- 00000110	+ 11111010
résultat:	00010001	1 00010001

Exercice: effectuer le calcul 110 – 10111 à l'aide du complément à 2!

1. *Complément pour atteindre 8 positions*

```
  110 → 00000110
10111 → 00010111
```

2. *formation du complément à 2*

```
00010111 (chiffre)
11101000 (compl.à 1)
11101001 (compl.à 2
= compl.à 1 +1)
```

3. *soustraction / addition*

```
  00000110
+ 11101001
```

report (borrow):

résultat: 11101111

4.4 Multiplication et division

Règles de calcul

Multiplication

$$\boxed{\begin{array}{ccccc} 0 & \cdot & 0 & = & 0 \\ 0 & \cdot & 1 & = & 0 \\ 1 & \cdot & 0 & = & 0 \\ 1 & \cdot & 1 & = & 1 \end{array}}$$

Division

$$\boxed{\begin{array}{ccccc} 0 & : & 1 & = & 0 \\ 1 & : & 1 & = & 1 \end{array}}$$

Exemples:

1. *Décimal*

$$\begin{array}{r} \underline{20 \cdot 11} \\ 0 \\ \underline{22\bullet} \\ \underline{\underline{220}}_{10} \end{array}$$

Binaire

$$\begin{array}{r} 1\;0\;1\;0\;0 : 1\;0\;1\;1 \\ 0 \\ 0\;\bullet \\ 1\;0\;1\;1\;\bullet\;\bullet \\ 0\;\bullet\;\bullet\;\bullet \\ \underline{1\;0\;1\;1\;\bullet\;\bullet\;\bullet\;\bullet} \\ \underline{\underline{1\;1\;0\;1\;1\;1\;0\;0}}_{2} \end{array}$$

$(2^7+2^6+2^4+2^3+2^2=220_{10})$

2. *Décimal*

$33 : 11 = \underline{\underline{3}}_{10}$

Binaire

$$\begin{array}{l} 1\;0\;0\;0\;0\;1 : 1\;0\;1\;1 = \underline{\underline{11}}_2 = 3_{10} \\ -\underline{\;\;1\;0\;1\;1} \\ \;\;0\;0\;1\;0\;1\;1 \\ \;\;\;\;\;-\underline{1\;0\;1\;1} \\ \;\;\;\;\;\;\;0\;0\;0\;0 \end{array}$$

Fractions binaires

Des chiffres après la virgule peuvent apparaître après la division.

Exemple: $0{,}111_2 =$ $2^{-1} + 2^{-2} + 2^{-3} = \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} = {}^1/_2 + {}^1/_4 + {}^1/_8 = \underline{\underline{0{,}875}}_{10}$

5 Exercices divers

	donné:		recherché:	
	nombre	système	système	nombre
1.	145’999	déc	hd	
2.	4832	déc	hd	
3.	2623	déc	hd	
4.	9CA	hd	déc	
5.	123F7	hd	déc	
6.	ACDC	hd	déc	
7.	1346	déc	2er	
8.	1537	déc	2er	
9.	63	déc	bin	
10.	1023	déc	dual	
11.	1010101	2	10	
12.	111000111	2	10	
13.	100100100	2	10	
14.	1111011101101	2	10	
15.	1000	déc	oct	
16.	1985	déc	oct	
17.	1234	oct	déc	
18.	5670	oct	déc	
19.	BBC	hd	déc	
20.	BCD	hd	déc	