Université Moulay Ismaïl
Faculté des Sciences et Techniques
Département de Physique
LST Sciences de l'Ingénieur – Énergies Renouvelables

Travaux Dirigés d'électronique numérique
Année Universitaire : 2020−2021
Pr. El− Hanaoui

Feuille de TD N° 1 : Systèmes de numération

Exercice 1 : Conversion

1. Convertir les mots binaires $(11010111)_2$, et $(1101101)_2$ en décimal.
2. Même question pour le mot $(101111101011110000010000000)_2$.
3. Convertir $(21)_{10}$ et $(255)_{10}$ en binaire.
4. Convertir $(1110010100101011 1)_2$, et $(1111110010100101011 1)_2$ en hexadécimal.
5. Convertir $(12A5)_{16}$, $(FC9E)_{16}$, $(CF9E)_{16}$, et $(8372)_{16}$ en binaire puis en décimal.

Exercice 2 : Nombres signés

Soient les quatre nombres hexadécimaux codés sur 8 bits suivants :
$(46)_{16}$, $(C6)_{16}$, $(24)_{16}$, $(CB)_{16}$

1. Convertir ces nombres en décimal en considérant les deux cas : $a)$. non signés, et $b)$. signés.
2. Convertir ces nombres sur 16 bits en considérant les cas précités.

Exercice 3 : Arithmétique

1. Effectuer les opérations suivantes en limitant le résultat à quatre chiffres significatifs et en indiquant l'état de la retenue : $(1253 + 7253)_{10}$, $(2345 + 8765)_{10}$, $(7854 – 2345)_{10}$, $(2345 – 7854)_{10}$. Commenter les résultats obtenus.
2. Effectuer les opérations suivantes (tous les nombres sur 8 bits en CA 2) : $(56 + 2C)_{16}$, $(56 - 2C)_{16}$, $(2C - 56)_{16}$, $(8C - 24)_{16}$, $(24 - 8C)_{16}$. Indiquer les valeurs des retenues C_6 et C_7 ainsi que de l'overflow.
3. Représenter en code **BCD** les nombres : 199, et 124, puis effectuer leur somme.

Exercice 4 : Flottants, Norme IEEE 754

1. Quels sont les plus petit et grand nombres réels représentables selon la norme IEEE 754 simple précision ?
2. Coder les réels suivants selon la norme IEEE 754 32 bits : $8,\ 9,\ 1.5,\ 3.14, -6.625,\ et\ 125$.
3. En virgule fixe, décoder le nombre binaire 11.011.
4. En virgule flottante normalisée, coder en binaire au format simple précision le réel 12.575 , puis effectuer le codage inverse.

5. Convertir en décimal, les nombres hexadécimaux réels données sous format IEEE 754 32 bits : $42E48000$, $3F880000$, $C7F00000$ $BFC00000$, $C0900000$ 80000008 .
6. Étant donnés les nombres $(0.10010 \cdot 10^{101})_2$, et $(0.11010 \cdot 10^{1})_2$, effectuer leurs somme et produit en virgule flottante.

Corrigé de Feuille de TD N° 1 : Systèmes de numération

Exercice 1 : Conversion

1. Convertissons en décimal les mots binaires suivants, en adoptant l'écriture polynomiale $(11010111)_2$, et $(1101101)_2$.

1.$a-$ On obtient :

$$11010111 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$\boxed{\textbf{11010111 = 215}}$$

1.$b-$ De même,

$$1101101 = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0$$

$$\boxed{\textbf{1101101 = 109}}$$

2. Même question pour le mot $(10111110101111000010000000)_2$.
On écrit ce mot sous sa forme polynomiale de base 16,
on obtient :

$$110\ 1111\ 1010\ 1111\ 0000\ 1000\ 0000 = 6FAF080_{16}$$

$$110\ 1111\ 1010\ 1111\ 0000\ 1000\ 0000 = 6 \cdot 16^6 + 15 \cdot 16^5 + 10 \cdot 16^4 + 15 \cdot 16^3 + 8 \cdot 16^1$$

$$\boxed{\textbf{110 1111 1010 1111 0000 1000 0000 = 50M}}$$

Cette valeur correspond à la fréquence du signal d'horloge CLK, de la carte DE1 de FPGA (CLS : Voir plus loin).

3. La conversion de $(21)_{10}$ et $(255)_{10}$ en binaire est issue des divisions successives par 2. Les reports r_i représenteront les bits associés à ces nombres ; en général,

$$\boxed{\mathbf{N}_{10} = (q_{k-1}r_{k-1}\ldots r_1r_0)_2}$$

On illustre cette conversion par le tableau 1. Le tableau 1 montre que :

résultats de $\div 2$:q_i	Nombre	Reports	r_i
	$N_{10} = 21$	1	r_0
q_0	10	0	r_1
q_1	5	1	r_2
q_2	2	0	r_3
q_3	1	1	q_3

TABLE 1 –

$$\boxed{\mathbf{21}_{10} = 10101_2}$$

De même pour

$$\boxed{\mathbf{255}_{10} = 11111111_2}$$

(Remarquer ici que ts les bits $= 1$; en effet $255_{10} = 2^n - 1$ avec $n = 8$).

DEC	BIN	HEX
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

TABLE 2 –

4. Il s'agit ds cette question de convertir en hexadécimal, les mots binaires suivants : $(11100101001010111)_2$, $(1111110010100101011 1)_2$.
   L'idée ici consiste à faire des regroupements de 4 bits à partir du poids faible. Ensuite, on remplace chaque regroupement par la valeur Héxa correspondante .
   On rappelle la table de conversion 2, qui montre le passage entre systèmes DEC$-$BIN$-$HEX.
   On tire alors,

   1 1100 1010 0101 0111 = 1CA57

   1111 1100 1010 0101 0111 = FCA57

5. Faisons l'opération inverse, qui permet de passer de l'hexadécimal : , $(FC9E)_{16}$, $(CF9E)_{16}$, et $(8372)_{16}$ en binaire puis en décimal. On obtient :

   $$\textbf{(12A5)}_{16} \;=\; 0001\ 0010\ 1010\ 0101 \;=\; 4773$$

   Idem pour les autres cas.

Exercice 2 : Nombres non signés, signés

$1-$ Soient 4 nombres hexadécimaux de taille 8 bits : (46) , $(C6)$, (24) , et (CB) . Convertissons $-$ les en décimal en considérant les cas :

$1.a-$ **non signé (Positif)** : La dynamique des nombres N représentables dans ce cas, est donnée par l'encadrement :

$$0 \leq N \leq 2^{n=8} - 1 \;\; ; \quad i.e \;\; 0 \leq N \leq 255$$

Les résultats de cette conversion sont portés dans le tableau 3.

$1.b-$ **Signé (positif ou négatif)** : L'intervalle des valeurs qu'on peut représenter en **CA2** est tel que :

$$-2^{n-1} \leq N \leq 2^{n-1} - 1 \;\; ; \quad i.e \;\; -128 \leq N \leq 127$$

Les résultats de cette conversion sont portés dans le tableau 4. (En effet, $N + 2^8 \;=\; N \;\; (modlo \;\; 2^8)$)

Nombre Hexa	Nombre DEC correspondant
(46)	70
$(C6)$	198
(24)	36
(CB)	203

TABLE 3 –

Nombre Hexa	Nombre DEC correspondant
(46)	70
$(C6)$	58
(24)	36
(CB)	53

TABLE 4 –

2. Changeons la taille des nombres de 8 à 16 bits. On complète à partir du bit de poids fort (MSB= bit de signe) avec des 0 si N est positif, avec des 1 dans le cas contraire. Les résultats sont résumés dans les tableau 5.

Nombre Hexa	Nombre Hexa (Cas NS)	Nombre Hexa (S)
(46)	(0046)	(0046)
$(C6)$	$(00C6)$	$(FFC6)$
(24)	(0024)	(0024)
(CB)	$(00CB)$	$(FFCB)$

TABLE 5 –

Exercice 3 : Arithmétique

1. Effectuons les opérations classiques suivantes en limitant le résultat à 4 chiffres significatifs, et en indiquant l'état de la retenue Co :

1253 + 7253 = 8506 (Co = 0 ; resultat Juste) ;

2345 + 8765 = 1110 (Co $\neq 0; resultat\ \ Faux$**)**

7854 - 2345 = 5509 (Co = 0 ; resultat V) ;

2345 - 7854 = 4491 (Co $\neq 0; resultat\ \ F$**)**

N.B : Ce résultat est rectifié en prenant le complément de 4491, affecté du signe $-$; soit : $-\,(10^4\ -\ 4491)\ =\ -5509$

2. Effectuons maintenant les opérations suivantes, tout en indiquant les valeurs des retenues r_6 et r_7 ainsi que de l'overflow (débordement de calculs OVF). Tous les nombres hexa sont de taille 8 bits, et représentés en système CA 2 . Le CA 2 permet en effet de transformer une soustraction en addition.

$$(56)\ +\ (2C)\ =\ 0101\,0110\ +\ 0010\,1100\ =\ 1000\,0010$$

(56) + (2C) = (82) ; $\mathbf{r}_6\ =\ 1;\ \ r_7\ =\ 0;\ \ OVF\ =\ 1$

$$(56) \; - \; (2C) \; = \; 0101\,0110 \; + \; 1101\,0011 \; + \; 1 \; = \; 0010\,1010$$

$$\boxed{\textbf{(56) - (2C) = (2A) ;} \;\; \mathbf{r}_6 \; = \; 1; \;\; r_7 \; = \; 1; \;\; OVF \; = \; 0}$$

$$(2C) \; - \; (56) \; = \; 0010\,1100 \; + \; 1010\,1001 \; + \; 1 \; = \; 1101\,0110$$

$$\boxed{\textbf{(2C) - (56) = (D6) ;} \;\; \mathbf{r}_6 \; = \; 0; \;\; r_7 \; = \; 0; \;\; OVF \; = \; 0}$$

Idem pour les autres cas,

$$\boxed{\textbf{(8C) - (24) = (68) ; OVF = 1 ; (24) - (8C) = (98) ; OVF = 1}}$$

3. Représentons en code **BCD** (Binary Coded Decimal) les nombres : 199, et 124, on obtient facilement :

$$199 \; = \; 0001\;1001\;1001$$

$$124 \; = \; 0001\;0010\;0100$$

Effectuons ensuite leur somme, càd :

$$\boxed{\textbf{199 + 124 = 11 0010 0011 = 323}}$$

N.B : Un ajout de 6 étant appliqué aux codes invalides 1101 , et 1100 , apparus suite à l'opération d'addition bit à bit.

Exercice 4 : Flottants, Norme IEEE 754

1. Déterminons l'intervalle I des nombres réels représentables selon la norme **IEEE** $-754-1985$ simple précision. X : Nombre flottant.

$$X \; = \; (-1)^s \cdot 2^{E_x - E_0} \cdot 1, F \cdots$$

où s est le signe de X, $E_x - E_0$ est l'exposant entier signé, codé en binaire décalé ; $E_0 \; = \; 2^{m_E - 1} - 1$ et $F \cdots$ est la partie fractionnaire de la valeur absolue de la mantisse. Le format de virgule flottante Simple précision étant sur 32 bits : $(m_E, m_F) = (8, 23)$. On a :

$$1 \; \leq \; E_X \; \leq \; 2^{m_E} \; - \; 2 \; = \; 2 \cdot E_0$$

$$1 \; - \; E_0 \; \leq \; E_X \; - \; E_0 \; \leq \; E_0$$

Et,

$$1 \; \leq \; 1, F \; \leq \; 2 \; - \; 2^{-m_F}$$

Ce qui permet d'écrire,

$$2^{1 \, - \, E_0} \; \leq \; |X| \; \leq \; \left(2 \; - \; 2^{-m_F}\right) \cdot 2^{E_0}$$

$$\boxed{\textbf{I =} \left[2^{1 \, - \, E_0} \; , \; 2^{1 \, + \, E_0}\right[\;\; ; \; E_0 \; = \; 127}$$

2. Représentons les réels suivants selon la norme IEEE 754 32 bits : $8, \; 9, \; 1.5, \; 3.14, -6.625, \; et \; 125$.

◇ **Le réel** 8 **est positif, le bit de signe est** 0 **;**

◇ **On convertit** 8 **(sans signe) en binaire,on obtient :** $1000,0 \; = \; 1,0 \cdot 2^3$

◇ **La partie après virgule, remplie de** 0 **à droite pour obtenir** 23 **bits ; cela donne** 00000000000000000000000

◇ **L'exposant est égal à** 3**, et on doit décaler puis convertir en binaire :** $3 + 127 = 130$ **codé par** 10000010 **;**

◇ **Au final** 8 **est codé par**

$$\boxed{\textbf{0 10000010 00000000000000000000000}}$$

▽ **Le réel** $3,14$ **a est** 0 **comme bit de signe ;**

▽ **Conversion en binaire donne :** $11,001 \ = \ 1,1001 \cdot 2^1$

▽ **La partie fractionnaire remplie de** 0 **à droite pour obtenir** 23 **bits ; cela donne** 10010000000000000000000

▽ **L'exposant est égal à** 1**, et on doit décaler puis convertir en binaire :** $1 + 127 = 128$ **codé par** 10000000 **;**

▽ **Au final** $3,14$ **est codé par,** (Idem pour le reste des cas.)

0 10000000 10010000000000000000000

3. Décodons en virgule fixe, le nombre binaire 11.011. Pour ce faire, on écrit (forme polynomiale) :$11,011 \ = \ 1 \cdot 2^1 \ + \ 1 \cdot 2^0 \ + \ 1 \cdot 2^{-2} \ + \ 1 \cdot 2^{-3}$. On obtient,

11,001 = 3,375 !

4. Je vous laisse le soin de mq la représentation au format SP du réel 12.575 est :

0 01000010 10010000000000000000000

Effectuer ensuite le codage inverse.

5. Convertissons en décimal, les hexadécimaux réels donnés sous format IEEE 754 32 bits : $42E48000$, $3F880000$, $C7F00000$ $BFC00000$, $C0900000$ 80000008 .

◇ **On convertit** $X \ = \ 42E48000$ **en binaire :**$X \ = \ 0 \ 10000101 \ 11001001000000000000000$

◇ **Le signe de l'hexadécimal réel :** $S = 0$ **;**

◇ **L'exposant en binaire décalé est égal à** 133**, Soit :** $E_X = 133 - 127 = 6$ **;**

◇ **La partie après la virgule, sur** 23 **bits est** $F = 2^{-1} + 2^{-2} + 2^{-5} + 2^{-7}$

◇ **Au final** $(42E48000)_{16}$ **représente le décimal**

42E48000 = (-1)$^0 \ 2^6 \ (1 + 2^{-1} + 2^{-2} + 2^{-5} + 2^{-7}) \ = \ 114,25$

À vous de faire des efforts !

6. Étant donnés les nombres $(\ 0.10010 \ \cdot \ 10^{101} \ )_2$, et $(\ 0.11010 \ \cdot \ 10^{1} \ )_2$, vérifier que leurs somme et produit en virgule flottante, sont respectivement : $(\ 0.10011101 \ \cdot \ 10^{101} \ )_2$, **et** $(\ 0.1110101 \ \cdot \ 10^{11} \ )_2$.