

Correction du DS informatique d'instrumentation No 1

F. Auger, 7/5/2015

exercice 1

a) Sur 3 bits, on peut compter de 0 à $(111)_2 = 4+2+1 = 7 = 2^3 - 1$
 Sur 31 bits, on peut compter de 0 à $2^{31} - 1 = 2,147,483,647$

b) Sur 63 bits, on peut compter de 0 à $2^{63} - 1 \approx 9,223 \cdot 10^{18}$

c) $N = 2320929609$

Pour obtenir rapidement sa représentation hexadécimale, on peut utiliser des divisions euclidiennes par $2^{12} = 4096$, ce qui donnera 3 chiffres hexadécimaux (car $4096 = 2^{12}$ et $12 = 3 \times 4$)

$$N = 566633 \times 4096 + 841 \quad \uparrow$$

$$566633 = 138 \times 4096 + 1385$$

$$138 = 0 \times 4096 + 138$$

$$138 = 8 \times 16 + 10 = (8A)_{16}$$

$$1385 = 5 \times 256 + 6 \times 16 + 9 = (569)_{16}$$

$$841 = 3 \times 256 + 4 \times 16 + 9 = (349)_{16}$$

$$\text{Donc } N = (8A569349)_{16}$$

$$= (10001010010101101001001101001001)_2$$

exercice 2

$$\text{Soit } N = (ABCDEF)_{16} = (101010111100 | 110111101111)_2$$

$$= \begin{array}{cccc} 101010 & 111100 & 110111 & 101111 \\ \begin{array}{cccc} 32 & 16 & 8 & 4 & 2 & 1 \end{array} & & & \end{array}$$

$$= /42 / 60 / 55 / 47/$$

$$= (983v)_{\text{base } 64}$$

$$(1997)_{16} = (000110011001 | 0111)_2 \leftarrow 2 \text{ octets} = 4 \times 4 = 2 \times 8 \text{ bits}$$

$$= /000110 / 011001 / 011100 / =$$

$$= /6 / 25 / 28 / = (GZC=)_{\text{base } 64}$$

$$(07|05|20|15)_{16} = (0000\ 0111\ 0000\ |0101\ 0010\ 0000\ |0001\ 0101)_{2}$$

4 octets = 3+1

$$= /000001/110000/010100/100000/000101/010000/==$$

$$= /1/48/20/32/5/16/==$$

$$= (BwVgFQ==)_{base\ 64}$$

b) (aHREbD4) = (26/7/17/45/27/3/56/E)

2 octets seuls à la fin

$$= /011010/000111/010001/101101/011011/000011/$$

$$56 = 32 + 16 + 8$$

48 56

111000 = /
on enlève 2 bits

$$= \begin{array}{|c|c|c|c|c|c|c|} \hline 0110 & 1000 & 0111 & 0100 & 0110 & 1101 & 0110 & 1100 \\ \hline 6 & 8 & 7 & 4 & 6 & D & 6 & C \\ \hline \end{array}$$

$$= (68746D6C3E)_{16}$$

3 octets 2 octets

0011/1110/
3 E

(dG1sPg==) base 64 = (29/6/53/44/15/32/==)

1 octet seul à la fin

$$= /011101\ 000110\ |110101\ 101100\ |001111\ 100000/$$

$$= (746D6C3E)_{16}$$

3 octets 1 octet

on enlève 4 bits

Devoir surveillé d'informatique d'instrumentation

Semestre 2, année 2014/2015. Durée : 1 heure 30.

Les trois exercices sont indépendants. Si vous joignez cet énoncé à votre copie, indiquez ci-dessous votre nom, prénom et groupe.

nom, prénom	groupe

1. (5 points)

- 1
1
1
- (a) Jusqu'en décembre 2014, le compteur qui permettait de savoir combien de fois une vidéo a été regardée sur *YouTube* était codé en binaire sur 31 bits. Quelle est la valeur maximale que permettait ce codage ?
- (b) Cette valeur a été dépassée par la vidéo de la chanson *Gangnam Style* du chanteur sud-coréen *PSY*, ce qui a conduit YouTube à coder tous ses compteurs sur 63 bits. Quelle est la valeur maximale que permet ce nouveau codage ?
- (c) À 18h55 le 03/05/2015, cette vidéo a été regardée 2 320 929 609 fois. Quelles sont les représentations hexadécimales et binaires de ce nombre ? Combien de bits utiliserait le codage en DCB de ce nombre ?

2. (9 points)

Présenté aussi la méthode utilisée pour obtenir les représentations (je suis sûr de votre réponse)

Le système de codage d'information¹ base64 est utilisé en informatique pour la transmission en ASCII des pièces jointes des courriers électroniques (images, vidéos, enregistrements sonores ...). Il utilise 64 caractères pour représenter un nombre N compris entre 0 et 63, c'est à dire un bloc de 6 bits. Le caractère C correspondant à chacun des 64 nombres N est indiqué sur le tableau de la figure 1.

Ce processus de codage consiste à coder chaque groupe de trois octets² en une succession de quatre caractères (car $3 \times 8 = 24 = 4 \times 6$). Les octets sont pris de gauche à droite, c'est à dire au fur et à mesure de leur production. Si le nombre d'octets à transmettre n'est pas un multiple de 3, alors

- si il reste un seul octet à la fin, soit 8 bits, on rajoute quatre 0 à droite pour avoir deux blocs de 6 bits (car $8 + 4 = 12 = 2 \times 6$) et on termine le dernier bloc de quatre caractères par deux caractères = ;
- si il reste deux octets à la fin, soit 16 bits, on rajoute deux 0 à droite pour avoir trois blocs de 6 bits (car $16 + 2 = 18 = 3 \times 6$) et on termine le dernier bloc de quatre caractères par un caractère =.

2 2 2

(a) Quels sont les codages en base64 des données dont les représentations hexadécimales sont *ABCDEF*, *1997* et *07052015* ? On fournira tout d'abord le nombre d'octets de ces trois données, leur représentation binaire et la représentation décimale de chacun des blocs de 6 bits qu'ils contiennent.

1,5/1,5

(b) Quelle est la représentation hexadécimale des données dont le code en base64 est *aHRtbdD4=* et *dG1sPg==* ?

3. (6 points)

Un système de contrôle qualité permet de vérifier les caractéristiques de pièces mécaniques fabriquées dans une usine. Soient P , L_o , L_a et E les variables logiques qui indiquent si le poids (P), la longueur (L_o), la largeur (L_a) et l'épaisseur (E) sont correctes. Elles sont au niveau logique 1 quand la caractéristique correspondante est correcte, et au niveau logique 0 quand elle est incorrecte. À partir de ces quatre variables logiques, on souhaite déduire trois variables logiques Q_1 , Q_2 et Q_3 qui vont permettre de trier les pièces en trois catégories :

- la variable Q_1 est au niveau logique 1 si au moins trois des quatre caractéristiques sont correctes ;
- la variable Q_2 est au niveau logique 1 si le poids est correct et si au moins deux des trois dimensions sont incorrectes ;
- la variable Q_3 est au niveau logique 1 si le poids est incorrect et si au moins une des trois dimensions est incorrecte.

¹Voir <http://fr.wikipedia.org/wiki/Base64>.

²Rappelons qu'un octet est un groupe de 8 bits.

N	C	N	C	N	C	N	C	N	C
0	A	13	N	26	a	39	n	52	0
1	B	14	O	27	b	40	o	53	1
2	C	15	P	28	c	41	p	54	2
3	D	16	Q	29	d	42	q	55	3
4	E	17	R	30	e	43	r	56	4
5	F	18	S	31	f	44	s	57	5
6	G	19	T	32	g	45	t	58	6
7	H	20	U	33	h	46	u	59	7
8	I	21	V	34	i	47	v	60	8
9	J	22	W	35	j	48	w	61	9
10	K	23	X	36	k	49	x	62	+
11	L	24	Y	37	l	50	y	63	/
12	M	25	Z	38	m	51	z		

Figure 1: Tableau des caractères utilisés par le codage base64 défini dans le RFC 4648, utilisé dans l'exercice 2. Comme le montre clairement ce tableau, ce codage utilise les 26 lettres de l'alphabet écrites en majuscules puis en minuscules, les dix chiffres décimaux et les deux caractères '+' et '/'.

0 correspond à la représentation décimale d'un bloc de 6 chiffres binaires

- 1 1 1 (a) En complétant le tableau de la figure 2, faire le tableau de vérité des fonctions Q_1, Q_2 et Q_3 .
 1 1 1 (b) À l'aide si besoin de tableaux de Karnaugh, en déduire des expressions minimales simplifiées des trois fonctions Q_1, Q_2 et Q_3 .

P	L_0	L_a	E	Q_1	Q_2	Q_3
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	0	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	1	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

Q_1		L_a, E			
P, L_0	00	01	11	10	
00					
01			1		
11		1	1	1	
10			1		

Q_2		L_a, E			
P, L_0	00	01	11	10	
00					
01					
11	1				
10		1			1

Q_3		L_a, E			
P, L_0	00	01	11	10	
00	1	1	1	1	
01	1	1	0	1	
11					
10					

$$Q_1 = \overline{L_0} \cdot L_a \cdot E + P \cdot (L_0 \cdot L_a + L_0 \cdot E + L_a \cdot E) = P \cdot L_0 \cdot (L_a + E) + L_a \cdot E \cdot (P + L_0)$$

$$Q_2 = P \cdot (\overline{L_a} \cdot \overline{E} + \overline{L_0} \cdot \overline{L_a} + \overline{L_0} \cdot \overline{E}) = P \cdot (\overline{L_a + E} + \overline{L_0 + L_a} + \overline{L_0 + E}) = P + (L_a + E) \cdot (L_0 + L_a) \cdot (L_0 + E)$$

$$Q_3 = \overline{P} \cdot \overline{L_0} \cdot L_a \cdot E = \overline{P} \cdot (\overline{L_0} + \overline{L_a} + \overline{E}) = \overline{Q_1 + Q_2} = P + L_0 \cdot L_a \cdot E$$

Figure 2: Tableaux utilisés dans l'exercice 3.

Q_1 : les 3 variables dimensionnelles ou P et 2 variables dimensionnelles
 Q_2 : P et au moins 2 variables dimensionnelles à zéro
 Q_3 : P à zéro et au moins une variable dimensionnelle à zéro