Correction du DS info instrum 1 14 mai 2009

F. Auger, 21/05/2009

exercice 1

a) $(1110100)_{-2} = (-2)^2 + (-2)^4 + (-2)^5 + (-2)^6$

6 5 4 3 2 1 0

$= 4 + 16 - 32 + 64 = (52)_{10}$

b)

$86 = -2 \times -43 + 0$

$-43 = -2 \times \underline{22} + 1$ ← attention !

$22 = -2 \times -11 + 0$

$-11 = -2 \times 6 + 1$

$6 = -2 \times -3 + 0$

$-3 = -2 \times 2 + 1$

$2 = -2 \times -1 + 0$

$-1 = -2 \times 1 + 1$

$1 = -2 \times \boxed{0} + 1$

donc $(86)_{10} = 110101010$

8 7 5 3 1

$= (-2)^8 + (-2)^7 + (-2)^5 + (-2)^3 + (-2)^1$

$= 256 - 128 - 32 - 8 - 2$

c)

0000	0	0100	4	1000	-8
0001	1	0101	5	1001	-7
0010	-2	0110	2	1010	-10
0011	-1	0111	3	1011	-9

1100	-4
1101	-3
1110	-6
1111	-5

On constate sur un tableau que sur 1 bit, on compte de 0 à 1

2 bits -2 à 1

3 bits -2 à 5

4 bits -10 à 5

Ces résultats sont bien conformes exactement aux expressions de l'énoncé :

	n = 1 (impair)	n = 2 (pair)	n = 3 (impair)	n = 4 (pair)
min	$-\frac{2}{3}(2^{1-1}-1) = 0$	$-\frac{2}{3}(2^2-1) = -2$	$-\frac{2}{3}(2^{3-1}-1) = -2$	$-\frac{2}{3}(2^4-1) = -10$
max	$\frac{1}{3}(2^{1+1}-1) = 1$	$\frac{1}{3}(2^2-1) = 1$	$\frac{1}{3}(2^{3+1}-1) = 5$	$\frac{1}{3}(2^4-1) = 5$

exercice 2

$x = \frac{120}{360} \times 64 = \frac{64}{3}$ sera représenté en virgule fixe avec 8 chiffres après la virgule par la fraction $\frac{y}{2^8} = \frac{y}{256}$ la plus proche de x.

$\frac{64}{3} \approx \frac{y}{256} \Rightarrow y \approx \frac{64 \times 256}{3} \Rightarrow y \approx 5461,33$

L'entier le plus proche est 5461 donc $y = 5461$

$5461 = 4096 + 1024 + 256 + 64 + 16 + 4 + 1$

donc $\frac{64}{3} \approx (0001010101010101)_{q8}$

x est positif donc sa notation en complément à 2 est identique à sa représentation binaire.

$x = \frac{240}{360} = \frac{128}{3}$ sera représenté par $\frac{y}{256}$ avec $y \approx \frac{128 \times 256}{3}$

$\Rightarrow y \approx 10922,66$ l'entier le plus proche (arrondi) est 10923

$10923 = 8192 + 2048 + 512 + 128 + 32 + 8 + 2 + 1$

donc $\frac{128}{3} \approx (0010101010101011)_{q8}$

exercice 3

$s_1 = \overline{\overline{a.b}\,.\,c} = \overline{\overline{a.b}} + \overline{c} = (a.b) + \overline{c}$

$s_2 = \overline{a.\,\overline{b.c}} \underbrace{= \overline{a} + \overline{\overline{b.c}} = \overline{a} + (b.c)}_{\text{en option, mais si vous l'écrivez, écrivez le correctement}}$

b)

a	b	c	s_1	s_2	s_3
0	0	0	1	1	1
0	0	1	0	1	1
0	1	0	1	1	1
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	0	1
1	1	1	1	1	0

$\bar{c}$ $a.b$ $b.c$ $\bar{a}$

écrire les lignes dans l'ordre du <u>code binaire naturel</u>.

Ceci montre que
(a nand b) nand c $\neq$ a nand (b nand c)
$\neq$ not (a and b and c)

L'opérateur NON ET <u>n'est pas associatif</u>, donc on n'obtient **PAS** une porte NON ET à 3 entrées en mettant 2 portes non ET à 2 entrées en série.

Écrire a nand b nand c n'a donc pas de sens, car il faut absolument écrire l'ordre dans lequel on fait les opérations.

<u>exercice 4</u>

a) les lignes 4 à 7 décrivent l'<u>entité</u> du circuit c'est à dire ce qu'il est nécessaire de connaître pour utiliser ce circuit
- son nom
- ses ~~p~~variables d'entrée (et leur type)
- ses variables de sortie (et leur type)

b) les lignes 9 à 16 décrivent l'<u>architecture</u> du circuit, c'est à dire la manière dont il est réalisé. On y trouve
- des déclarations de signaux logiques internes au circuit
- les opérations qui élaborent les sorties à partir des entrées.

Attention ! <= ne veut pas dire "inférieur ou égal" en VHDL
Il correspond au symbole d'affectation. Il représente une flèche ← et se lit "reçoit".

c) le tableau de vérité déduit du chronogramme montre que la fonction réalisée correspond à la fonction s_1 de l'exercice 3. Attention ! Ce n'est pas parce qu'on fait un chronogramme que le circuit étudié est séquentiel !

exercice 5

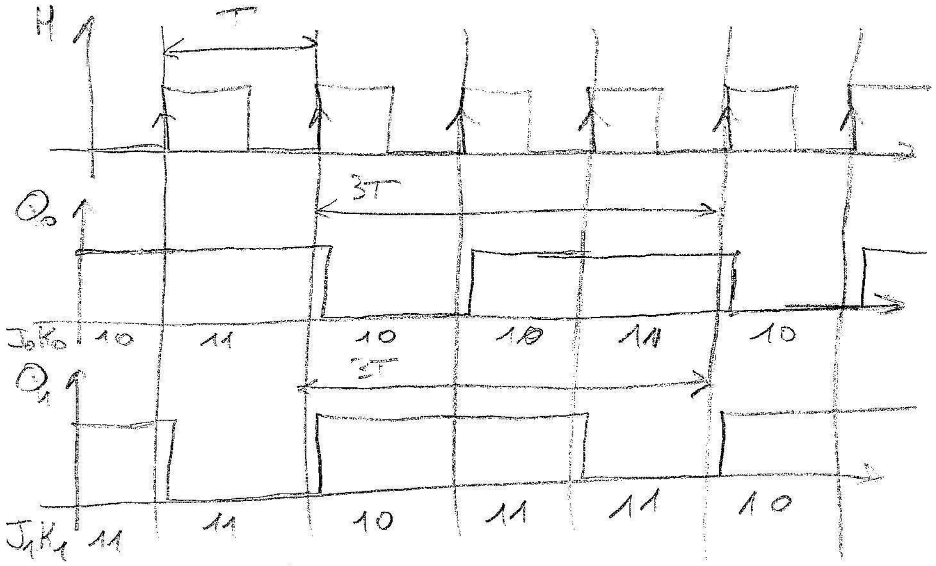
J	K	Q
0	0	Q_{prec}
0	1	0
1	0	1
1	1	$\overline{Q_{prec}}$

On ne travaille que sur ces 2 lignes

$J_0 = 1$ $\quad$ $J_1 = 1$

$K_0 = \overline{Q_1}$ $\quad$ $K_1 = Q_0$

Les deux bascules sont <u>synchrones</u> et changent d'état sur un front <u>montant</u> du signal d'horloge (avec un petit retard de propagation)



Q_1Q_0 11 01 10 11 01 10

Les deux signaux Q_0 et Q_1 sont – periodiques de période 3T

– rapport cyclique $2/3 \simeq 66\%$

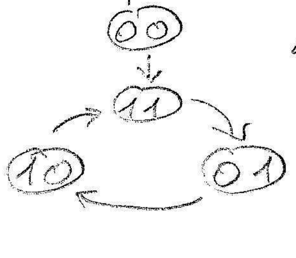
– déphasés de 120°

Ce circuit fabrique 2 signaux de fréquence $1/3T$ à partir d'un signal de fréquence $1/T$: c'est un diviseur de fréquence par 3

c) Si $Q_1 = Q_0 = 0$ alors $J_0 = 1$ $K_0 = 1$ donc Q_0 passe à 1

$J_1 = 1$ $K_1 = 0$ donc Q_1 passe à 1

Si $(Q_1Q_0)_2 = (00)_2$, le circuit passe à $(Q_1Q_0)_2 = (11)_2$ au coup d'horloge suivant et retournera à son cycle d'oscillation à 3 temps. C'est un circuit <u>stable</u>, qui n'a pas besoin d'être <u>initialisé</u> à une valeur bien précise.



Université de Nantes - IUT de Saint Nazaire

Département de Mesures Physiques

Devoir surveillé d'informatique d'instrumentation I

Semestre 2, 2008/2009. Durée : 1 **heure 30**.

Les cinq 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)** La représentation *négabinaire* est une autre technique de codage en binaire des nombres entiers relatifs[1]. Elle consiste à décomposer tout nombre en une somme de puissances de $b = -2$, affectés de coefficients multiplicatifs 0 ou 1 :

$$\begin{aligned}(b_{n-1}\, b_{n-2} \ldots b_1\, b_0)_{-2} &= b_{n-1}\,(-2)^{n-1} + b_{n-2}\,(-2)^{n-2} + \ldots + b_2\,(-2)^2 + b_1\,(-2)^1 + b_0 \\ &= \sum_{i=0}^{n-1} b_i\,(-2)^i\end{aligned}$$

Par exemple $(1011)_{-2}$ correspond à $1-2-8=-9$ et $(110)_{-2}$ correspond à $-2+4=2$. Pour trouver la représentation négabinaire d'un nombre, on peut utiliser un algorithme d'Euclide en effectuant une succession de divisions entières par -2, avec des restes de 0 ou 1. La représentation est alors formée par les restes successifs de ces divisions, écrits en commençant par le dernier. Par exemple :

$$\begin{array}{rcrcrcl} 6 &=& -2 &\times& -3 &+& 0, \\ -3 &=& -2 &\times& 2 &+& 1, \\ 2 &=& -2 &\times& -1 &+& 0, \\ -1 &=& -2 &\times& 1 &+& 1, \\ 1 &=& -2 &\times& 0 &+& 1, \end{array}$$

ce qui conduit à $(6)_{10} = (11010)_{-2} = 16-8-2$.

   (a) Quelle est la représentation décimale du nombre dont la représentation négabinaire est $(1110100)_{-2}$?

   (b) Quelle est la représentation négabinaire de $(86)_{10}$?

   (c) Compléter le tableau de la figure 1 en indiquant le nombre dont la représentation négabinaire correspond à chacun de ces 16 codes à 4 bits. Utiliser ce tableau pour vérifier, pour $n = 1$, 2, 3 et 4, que la représentation négabinaire sur n bits permet de coder des nombres compris

   - entre $-\frac{2}{3}\,(2^n-1)$ et $\frac{1}{3}\,(2^n-1)$ quand n est pair,
   - entre $-\frac{2}{3}\,(2^{n-1}-1)$ et $\frac{1}{3}\,(2^{n+1}-1)$ quand n est impair.

   Quel est l'intervalle des nombres entiers relatifs que l'on peut coder sur 8 bits avec cette technique ?

   (d) La représentation négabinaire vous parait-elle plus intéressante que la notation en complément à 2 ?

2. **(3 points)** Écrire les représentations en virgule fixe sur 16 bits avec 8 bits après la virgule de $\frac{120}{360}\times 64$ et $\frac{240}{360}\times 64$.

[1] Le représentation des nombres sur des bases négatives a été étudiée pour la première fois par V. Grunwald en 1885. Voir `http://fr.wikipedia.org/wiki/Systeme_negabinaire`

représentations							
négabinaire	décimale	négabinaire	décimale	négabinaire	décimale	négabinaire	décimale
0000		0100		1000		1100	
0001		0101		1001		1101	
0010		0110	2	1010		1110	
0011		0111		1011	-9	1111	

Figure 1: Représentation décimale des nombres ayant une représentation négabinaire à 4 chiffres, étudiée dans l'exercice 1.

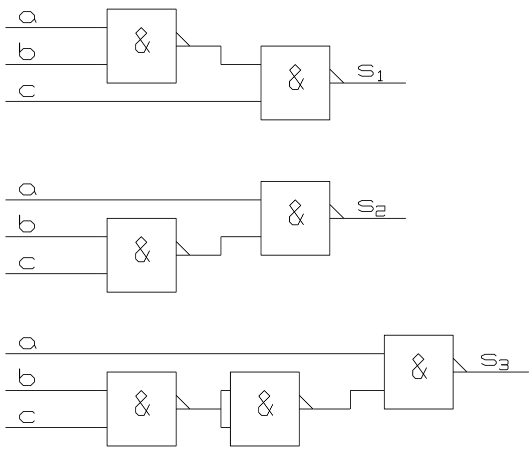


Figure 2: circuits étudiés dans l'exercice 3.

3. **(4 points)** Les circuits de la figure 2 fournissent trois variables s_1, s_2 et s_3 à partir des variables logiques a, b et c.

   (a) Quelles sont les expressions de s_1, s_2 et s_3 ?

   (b) Faire un tableau de vérité de ces trois variables. Ces trois fonctions sont-elles égales ?

4. **(5 points)** Le code VHDL ci-dessous décrit un circuit qui fabrique une variable `s` à partir de 3 variables `e1`, `e2` et `e3` :

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity InfoInstrum1_2009 is
5   port ( e1, e2, e3 : in  std_logic ;
6          s          : out std_logic);
7  end InfoInstrum1_2009;
8
9  architecture arch_InfoInstrum1_2009 of InfoInstrum1_2009 is
10
11 signal si : std_logic ;
12
13 begin
14  si <= e1 nand e2 ;
15  s  <= si nand e3 ;
16 end arch_InfoInstrum1_2009;
```

   (a) À quoi correspondent les lignes 4 à 7, et quelles informations indiquent-elles ?

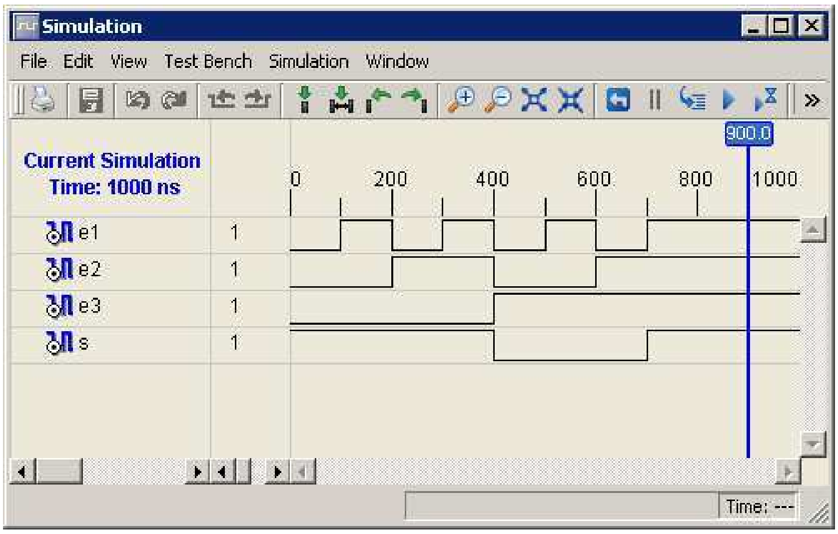


Figure 3: Chronogramme résultant de la simulation du circuit étudié dans l'exercice 4.

(b) À quoi correspondent les lignes 9 à 16, et quelles informations indiquent-elles ?

(c) La figure 3 montre un chronogramme des variables logiques `e1`, `e2`, `e3` et `s` obtenu par simulation. Quelle est la fonction logique réalisée par ce circuit ?

5. **(3 points)** Le circuit[2] de la figure 4 utilise deux bascules JK dont les entrées d'horloge sont toutes reliées au même signal logique H, périodique de période T et de rapport cyclique égal à 50 %. Leurs entrées J sont toutes les deux forcées au niveau logique haut.

(a) Tracer sur un chronogramme les signaux logiques H, Q_0 et Q_1, sur au moins 6 périodes de H, en tenant compte du retard de propagation des bascules et en supposant qu'à l'instant initial $Q_0 = 1$ et $Q_1 = 1$.

(b) Quelles sont les valeurs successives de $(Q_1\, Q_0)_2$? Quel est l'intérêt du circuit ?

(c) Que se passe t-il si $(Q_1\, Q_0)_2 = 0$ à l'instant initial ?

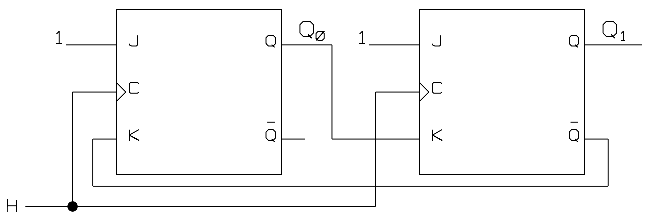


Figure 4: circuit étudié dans l'exercice 5.

[2]Voir J.F. Wakerly, "Digital design, principles and practices", third edition (2001), p 648.