

Notions vues au cours No 1,
 au TP 1 et en TD
 (exercice 26 du chapitre 2)

Exercice 1

$$x = (-1)^s (1+f) 2^E$$

a). Si $|x| \geq 2$ alors $E \geq 1$, donc $E = 127 + l = 128 + (l-1) = 2^7 + (l-1)$
 avec $l-1 \geq 0$. Pour obtenir la représentation binaire de E ,
 il suffit donc de prendre la représentation binaire de $l-1$
 et de rajouter 128, c'est à dire de mettre le bit 7 à 1.

• La partie fractionnaire f est codée sur 23 bits, Pour obtenir
 ces 23 bits, on peut utiliser un algorithme d'Euclide et faire
 23 multiplications par 2 : ce sera long. Pour aller plus vite, on
 peut faire 5 multiplications par 16, qui donneront chacune 4 bits,
 puis une multiplication par 8 pour avoir les 3 derniers bits.
 Pour aller encore plus vite, on peut obtenir les 3 premiers blocs
 d'un coup en multipliant par $4096 = 2^{12} = 2^{3 \times 4}$. Il reste alors
 2 blocs de 4 bits et 1 bloc de 3 bits, soit 11 bits, que l'on
 peut obtenir en multipliant par $2^{11} = 2048$

b) $25 < 30 < 36$ donc $5 < \sqrt{30} < 6$ donc $4 < \sqrt{30} < 8$

Donc $l = 2$ $m = \frac{\sqrt{30}}{2^2} = \frac{\sqrt{30}}{4}$ et $f = m - 1 = \frac{\sqrt{30}}{4} - 1$

Donc $E = 127 + l = 128 + (l-1) = 128 + 1 = \underbrace{(1000\ 000\ 1)}_{128\ 1}_2$

$4096 f = 1512,679 \rightarrow 1512 = 1024 + 256 + 128 + 32 + 8$ $f' = 0,679$

$2048 f' = 1390,56 \rightarrow 1391 = 1024 + 256 + 64 + 32 + 8 + 4 + 2 + 1$

et $x > 0$ donc $s = 0$ donc

$(0 | 1000\ 000\ 1 | \underbrace{010110101000}_{1512} \underbrace{10101101111}_{1391})_{IEEE754} \approx \sqrt{30}$

exercice 2

a) $(987)_{10} = (\underbrace{1001}_{\downarrow} \underbrace{1000}_{\downarrow} \underbrace{0111}_{\downarrow})_{dcb}$
 $(493)_{10} = (\underbrace{0100}_{\downarrow} \underbrace{1001}_{\downarrow} \underbrace{0011}_{\downarrow})_{dcb}$
 $(246)_{10} = (\underbrace{0010}_{\downarrow} \underbrace{0100}_{\downarrow} \underbrace{0110}_{\downarrow})_{dcb}$
 $(123)_{10} = (\underbrace{0001}_{\downarrow} \underbrace{0010}_{\downarrow} \underbrace{0011}_{\downarrow})_{dcb}$

Ces nombres sont les quotients successifs de divisions par 2

Code DCB :
vu au cours N°1, au TP3 et en TD (exo 40)

b) Si $D = (d_3 d_2 d_1 d_0)_2$ alors

$$D = 8d_3 + 4d_2 + 2d_1 + d_0$$

$$= 2(4d_3 + 2d_2 + d_1) + d_0$$

$$= 2Q + R$$

(décomposition sur les puissances successives de 2)

cours N°1, TP1
exo 2, 3 vu en TD

avec $Q = 4d_3 + 2d_2 + d_1 = (d_3 d_2 d_1)_2$
et $R = d_0$

On peut aussi montrer ce résultat en faisant tous les cas possibles

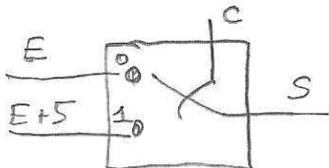
c)

E	e_2	e_1	e_0	R	r_3	r_2	r_1	r_0
0	0	0	0	5	0	1	0	1
1	0	0	1	6	0	1	1	0
2	0	1	0	7	0	1	1	1
3	0	1	1	8	1	0	0	0
4	1	0	0	9	1	0	0	1

Tableau de vérité avec seulement 5 lignes (spécification incomplète)

On peut aussi utiliser la méthode des tableaux de Karnaugh

$$S = \begin{cases} E & \text{si } c = 0 \\ E+5 & \text{si } c = 1 \end{cases}$$



correspond à la sortie d'un multiplexeur à 2 entrées

vu au TP4

Donc

$$s_3 = 0 \cdot \bar{c} + r_3 \cdot c = r_3 \cdot c$$

$$s_2 = e_2 \cdot \bar{c} + r_2 \cdot c$$

$$s_1 = e_1 \cdot \bar{c} + r_1 \cdot c$$

$$s_0 = e_0 \cdot \bar{c} + r_0 \cdot c = e_0 \cdot \bar{c} + \bar{e}_0 \cdot c = e_0 \oplus c$$

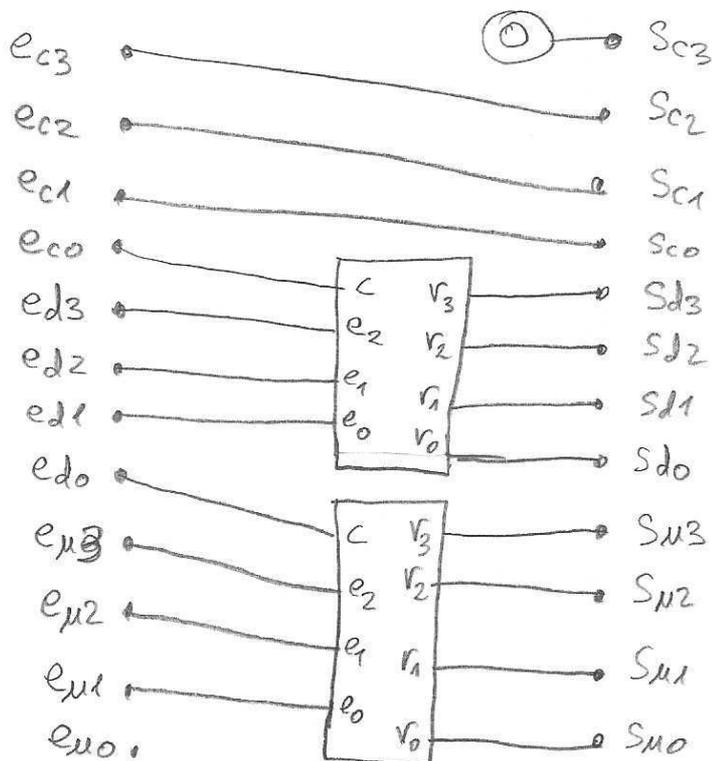
d) En regardant les résultats de la question a, on peut voir que

- Pour le chiffre des centaines, on fait simplement une division par 2 à l'aide d'un décalage à droite de 1 bit.
- Pour le chiffre des dizaines
 - soit le chiffre des centaines est pair, il suffit alors de diviser le chiffre des dizaines par un décalage à droite de 1 bit.
 - soit le chiffre des centaines est impair, il suffit alors de rajouter 10 au chiffre des dizaines avant de diviser par 2 c'est à dire ajouter 5 à la division par 2 du chiffre des dizaines
 exemple $18 / 2 = (10 + 8) / 2 = 5 + 8 / 2 = 9$
 $13 / 2 = (10 + 3) / 2 = 5 + 3 / 2 = 6$

c'est justement ce que fait le circuit de la question c

• Pour le chiffre des unités, c'est pareil.

D'où le schéma suivant



exercice 3

Voir feuilles ci-jointes

Cet exercice reprend les notions vues aux cours 2 et 4, aux TP 2 et 3 et en TD (exercices du chapitre 5).

exercice 4

Cet exercice reprend des notions vues aux cours 3 et 4, aux TP 3 et en TD (exercices du chapitre 6 et 7)

Je rappelle qu'en cours, j'ai traité l'exercice 10 p 92, en expliquant assez clairement pourquoi...

a) On veut compter de 0 à $4 = (100)_2$. On a donc besoin pour cela de 3 chiffres binaires, donc 3 bascules JK.

b)

Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	X	0	X	1	X
0	0	1	0	X	1	X	X	1
0	1	0	0	X	X	0	1	X
0	1	1	1	X	X	1	X	1
1	0	0	X	1	0	X	0	X

J	K	Q_n
0	0	Q_{n-1}
0	1	0
1	0	1
1	1	$\overline{Q_{n-1}}$

	J	K
$0 \rightarrow 0$	0	X
$0 \rightarrow 1$	1	X
$1 \rightarrow 0$	X	1
$1 \rightarrow 1$	X	0

On applique la méthode expliquée 3 fois en cours.

On obtient

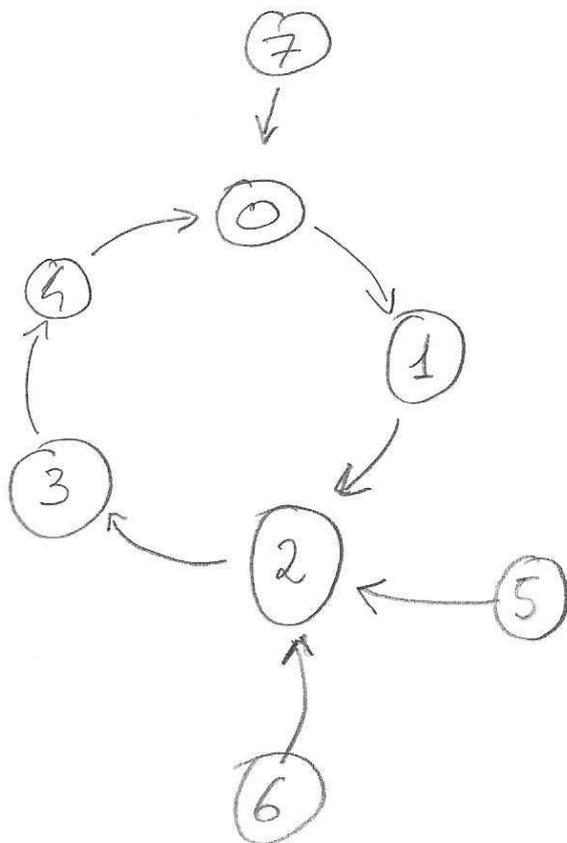
$$K_0 = 1$$
$$J_0 = \overline{Q_2}$$
$$K_1 = Q_0$$
$$J_1 = Q_0$$
$$K_2 = Q_2 \text{ (ou } 1)$$
$$J_2 = Q_1 \cdot Q_0$$

d) Pour savoir ce qui se passe si le circuit "sort" de son cycle nominal, on calcule dans chaque cas les niveaux logiques présents aux entrées des 3 bascules, pour en déduire l'état suivant

Si $(Q_2 Q_1 Q_0)_2 =$
 $(1 0 1)_2 = 5$, alors $\underbrace{J_2=0 \quad K_2=1}_{\Downarrow} \quad \underbrace{J_1=1 \quad K_1=1}_{\Downarrow} \quad \underbrace{J_0=0 \quad K_0=1}_{\Downarrow}$
 $Q_2=0 \quad Q_1=1 \quad Q_0=0$
 \rightarrow on retourne à $(010)_2 = 2$

Si $(Q_2 Q_1 Q_0)_2 =$
 $(1 1 0)_2 = 6$, alors $\underbrace{J_2=0 \quad K_2=1}_{\Downarrow} \quad \underbrace{J_1=0 \quad K_1=0}_{\Downarrow} \quad \underbrace{J_0=0 \quad K_0=1}_{\Downarrow}$
 $Q_2=0 \quad Q_1=1 \quad Q_0=0$
 \rightarrow on retourne aussi à $(010)_2 = 2$

Si $(Q_2 Q_1 Q_0)_2 =$
 $(1 1 1)_2 = 7$, alors $\underbrace{J_2=1 \quad K_2=1}_{\Downarrow} \quad \underbrace{J_1=1 \quad K_1=1}_{\Downarrow} \quad \underbrace{J_0=0 \quad K_0=1}_{\Downarrow}$
 $Q_2=0 \quad Q_1=0 \quad Q_0=0$



Ce circuit est stable :
 si on le sort de sa position
 d'équilibre, il y revient

```

-----
-- DivideBy2Bcd3

library IEEE;
use IEEE.std_logic_1164.all;

entity AddFiveOrNot is
  port( c : in std_logic
        E : in std_logic_vector(2 downto 0) ;
        S : out std_logic_vector(3 downto 0));
end AddFiveOrNot;

-----
-- arch_DivideBy3_Ver2

architecture arch_AddFiveOrNot of AddFiveOrNot is

signal r3, r2, r1 : std_logic;

begin
  r3 <= E(2) or (E(1) and E(0)) ; -- calcul de r3, r2 et r1
  r2 <= not(r3) ; -- r0 est inutile
  r1 <= E(1) xor E(0) ; -- car r0=not(E(0))

  S(3) <= c and r3 ;
  S(2) <= r2 when c='1' else E(2) ; -- multiplexeurs a deux entrees
  S(1) <= r1 when c='1' else E(1) ;
  S(0) <= c xor E(0) ;
end arch_AddFiveOrNot;

-----
-- DivideBy2Bcd3

library IEEE;
use IEEE.std_logic_1164.all;

entity DivideBy2Bcd3 is
  port( E2, E1, E0 : in std_logic_vector(3 downto 0) ;
        S2, S1, S0 : out std_logic_vector(3 downto 0));
end DivideBy2Bcd3;

-----
-- Arch_DivideBy2Bcd3

architecture Arch_DivideBy2Bcd3 of DivideBy2Bcd3 is

begin
  S2 <= '0' & E2(3 downto 1); -- division par 2 par decalage a droite

  digit1 : entity work.AddFiveOrNot
    port map( E2(0), E1(3 downto 1), S1); -- E1/2 ou 5+E1/2 suivant E2(0)

  digit0 : entity work.AddFiveOrNot
    port map( E1(0), E0(3 downto 1), S0); -- E0/2 ou 5+E0/2 suivant E1(0)
end Arch_DivideBy2Bcd3;

-----
-- TestDivideBy2Bcd3

library IEEE;
use IEEE.std_logic_1164.all;

entity TestDivideBy2Bcd3 is
  port( reset, button : in std_logic
        N2, N1, N0 : out std_logic_vector(3 downto 0));
end TestDivideBy2Bcd3;

-----
-- Arch_TestDivideBy2Bcd3

architecture Arch_TestDivideBy2Bcd3 of TestDivideBy2Bcd3 is

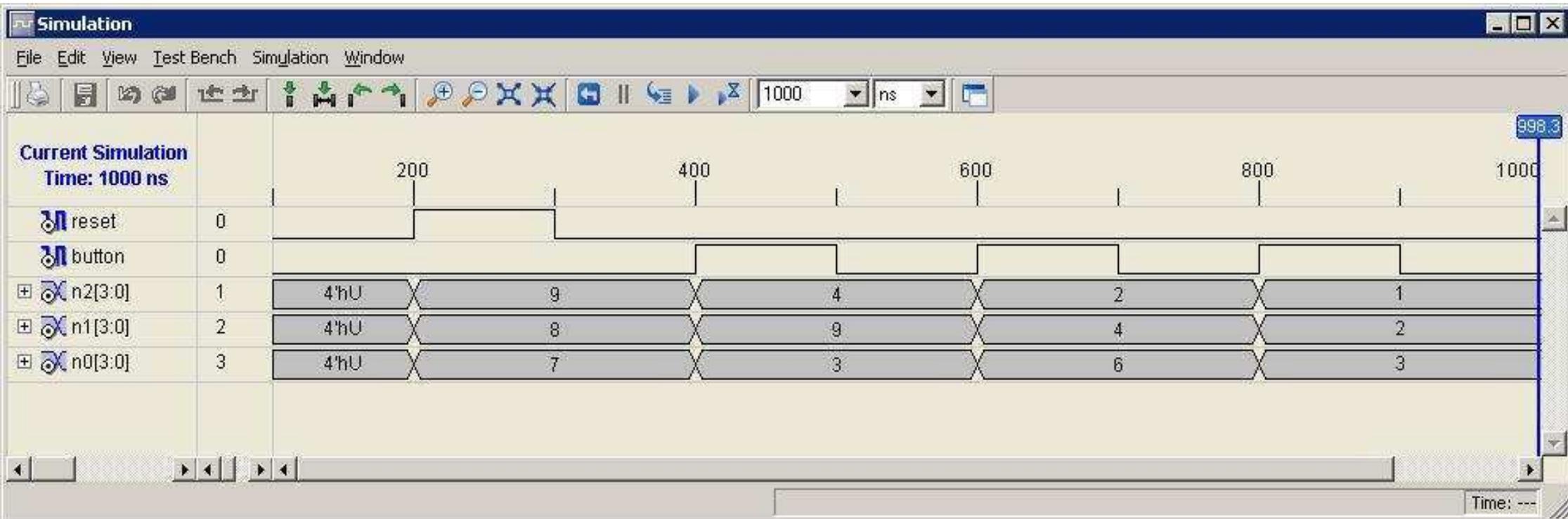
signal N2i, N1i, N0i, Ndiv2_2, Ndiv2_1, Ndiv2_0 : std_logic_vector(3 downto 0);

begin
  label0 : entity work.DivideBy2Bcd3
    port map(N2i, N1i, N0i, Ndiv2_2, Ndiv2_1, Ndiv2_0);

  process (reset, button)
  begin
    if (reset='1') then -- reinitialisation
      N2i <= "1001" ;
      N1i <= "1000" ;
      N0i <= "0111" ;
    elsif (button'event and button='1') then -- on divise par 2 a
      N2i <= Ndiv2_2; -- chaque appui sur le bouton
      N1i <= Ndiv2_1;
      N0i <= Ndiv2_0;
    end if;
  end process;

  N2 <= N2i; N1 <= N1i; N0 <= N0i;
end Arch_TestDivideBy2Bcd3;

```



2^{ème} Devoir surveillé d'informatique d'instrumentation I

Semestre 2, 2008/2009. Durée : 1 **heure 45**.

Les exercices 1, 2 et 4 sont indépendants. Si vous joignez cet énoncé à votre copie, indiquez ci-dessous votre nom, prénom et groupe.	
nom, prénom	groupe

1. (4 points)

(a) Valider et justifier l'affirmation suivante :

Pour obtenir rapidement la représentation au format IEEE 754 simple précision d'un nombre réel dont la valeur absolue est supérieure à 2, de la forme $x = (-1)^s (1 + f) 2^l$, il suffit :

- de coder l'exposant l par l'entier naturel $E = 128 + (l - 1)$;
- d'obtenir la représentation binaire de la partie fractionnaire f à l'aide d'une multiplication par 4096 suivie d'une multiplication par 2048.

(b) Mettre cette affirmation en pratique pour trouver la représentation au format IEEE 754 simple précision de $x = \sqrt{30}$.

2. (6 points) Un compteur DCB est utilisé pour mesurer le temps que met un charriot téléguidé pour faire l'aller-retour entre deux endroits. Malheureusement, c'est la durée d'un trajet qui est souhaitée, ce qui implique de diviser par 2 le résultat fourni par le compteur. Pour faire cela, on souhaite concevoir un circuit qui, à partir d'un nombre entier E compris entre 0 et 999 et codé en DCB, fournit un nombre S égal au quotient de la division par 2 de E , codé en DCB.

nombre	c_3	c_2	c_1	c_0	d_3	d_2	d_1	d_0	u_3	u_2	u_1	u_0
987												
493												
246												
123												

Figure 1: tableau utilisé dans l'exercice 2.

- (a) Donner les codes DCB des nombres 987, 493, 246, 123, en complétant le tableau de la figure 1.
- (b) Soit $D = (d_3 d_2 d_1 d_0)_2$ un nombre compris entre 0 et 9 représenté en binaire sur 4 bits. Montrer que le quotient de la division par 2 de D est¹ $Q = (d_3 d_2 d_1)_2$ et que le reste est d_0 .
- (c) Soit $E = (e_2 e_1 e_0)_2$ un nombre compris entre 0 et 4 représenté en binaire sur 3 bits. Faire un tableau de vérité puis donner des expressions logiques simples des variables r_3, r_2, r_1 et r_0 telles

¹Une telle opération correspond à un décalage des chiffres d'une position vers la droite et est appelée un décalage à droite (*right shift*) de 1 bit.

que $R = (r_3 r_2 r_1 r_0)_2 = E + 5$. En déduire ensuite les équations internes du circuit qui, à partir des variables d'entrée c, e_2, e_1 et e_0 , fournit des variables s_3, s_2, s_1 et s_0 telles que

$$S = (s_3 s_2 s_1 s_0)_2 = \begin{cases} E & \text{si } c = 0 \\ E + 5 & \text{si } c = 1 \end{cases}, \quad \text{avec } E = (e_2 e_1 e_0)_2$$

- (d) À l'aide des résultats des questions précédentes, réaliser le schéma d'un circuit qui, à partir d'un nombre $E = (e_{c3} e_{c2} e_{c1} e_{c0} e_{d3} e_{d2} e_{d1} e_{d0} e_{u3} e_{u2} e_{u1} e_{u0})_{\text{dcb}}$ compris entre 0 et 999 et codé en DCB, fournit un nombre $S = (s_{c3} s_{c2} s_{c1} s_{c0} s_{d3} s_{d2} s_{d1} s_{d0} s_{u3} s_{u2} s_{u1} s_{u0})_{\text{dcb}}$ codé en DCB et égal au quotient de la division par 2 de E .

3. (5 points)

- (a) Écrire l'architecture de la description en VHDL du circuit de la question 3 de l'exercice 2, associée à l'entité

```
entity AddFiveOrNot is
  port( c : in  std_logic          ;
        E : in  std_logic_vector(2 downto 0) ;
        S : out std_logic_vector(3 downto 0));
end AddFiveOrNot;
```

- (b) Écrire l'architecture de la description en VHDL du circuit de la question 4 de l'exercice 2, associée à l'entité

```
entity DivideBy2Bcd3 is
  port( E2, E1, E0 : in  std_logic_vector(3 downto 0) ;
        S2, S1, S0 : out std_logic_vector(3 downto 0));
end DivideBy2Bcd3;
```

- (c) Écrire l'architecture associée à l'entité

```
entity TestDivideBy2Bcd3 is
  port( reset, button : in  std_logic          ;
        N2, N1, N0    : out std_logic_vector(3 downto 0));
end TestDivideBy2Bcd3;
```

décrivant en VHDL un circuit qui

- réinitialise à 987 le nombre N dont le code DCB est formé des chiffres $N2, N1$ et $N0$, lorsque la variable `reset` est au niveau logique haut.
- remplace N par $N/2$ lorsque la variable `button` passe au niveau logique haut.

4. (5 points) On souhaite concevoir un compteur synchrone modulo 5 (qui compte donc périodiquement de 0 à 4, en représentation binaire : 0, 1, 2, 3, 4, 0, 1, 2 ...).

- (a) Combien de bascules JK sont-elles nécessaires ?
- (b) Faire un tableau des niveaux logiques qui doivent se trouver aux entrées des bascules pour obtenir les transitions désirées.
- (c) En déduire des expressions logiques simples de ces variables d'entrée.
- (d) Que se passe-t-il si, lors de la mise sous tension du circuit ou à la suite d'une perturbation, la sortie du circuit n'est pas égale à une des valeurs du cycle nominal ?