

# Travail écrit 1: Système Numérique 2

2016/2017

Filière : Télécommunication

Classe : T-2a, T-2d

Date : 26 avril 2017, 13h00 à 14h35

Professeur : Fabio Cunha

Nom et prénom :

Points : 12.5 / 20

Note : 4.1

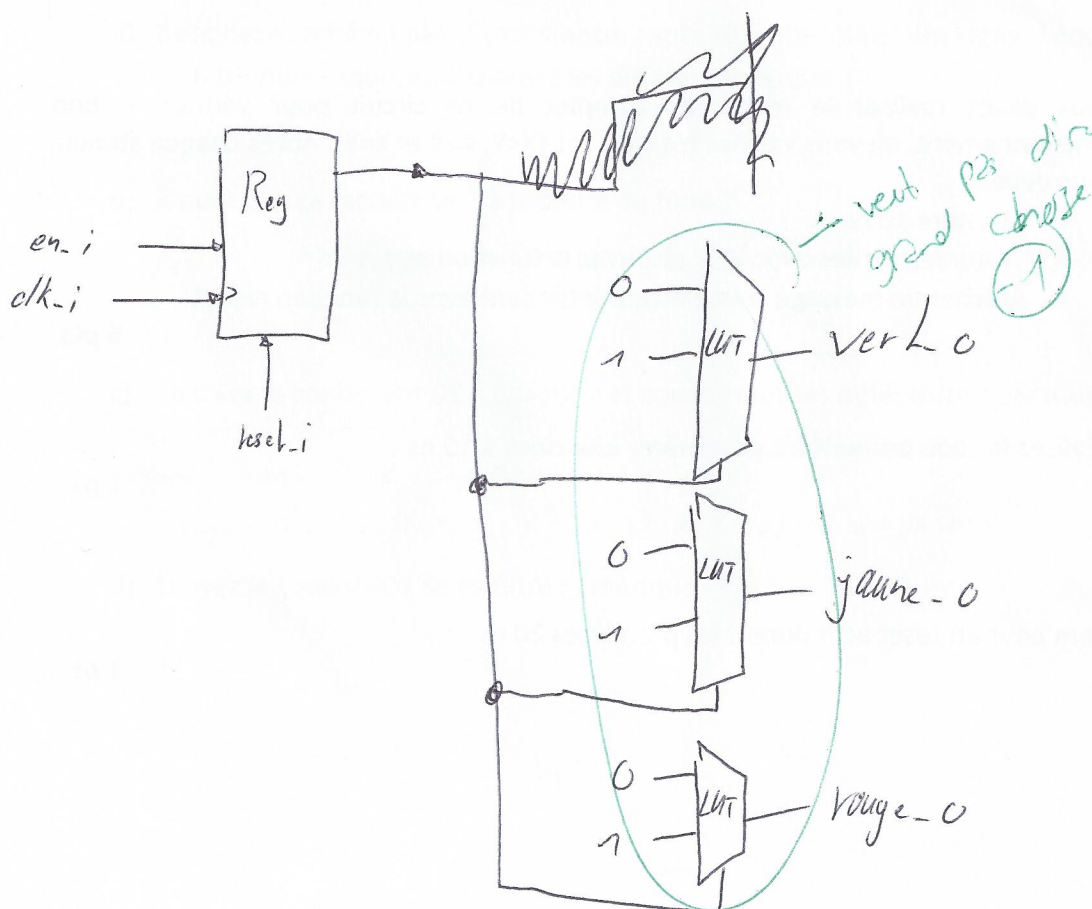
## Problème 1: Feu de carrefour routier (5 pts)

On veut gérer la transition **vert** → **jaune** → **rouge** → **jaune\_et\_rouge** → **vert** → **jaune...etc.** d'un feu de carrefour routier.

On dispose d'une horloge **clk\_i** de 1Hz. Le temps sur **vert** ou **rouge** doit être de 1 minutes, le temps sur **jaune** de 5s et le temps sur **jaune\_et\_rouge** de 3s. Le circuit numérique réalisant cette fonction a trois entrées : **en\_i**, **clk\_i** et **reset\_i**. Il a trois sorties : **vert\_o**, **jaune\_o** et **rouge\_o**.

- Ecrire le code VHDL complet.
- Dessiner le schéma logique à partir de votre code

3.5 / 5 pts  
1 / 2 pts



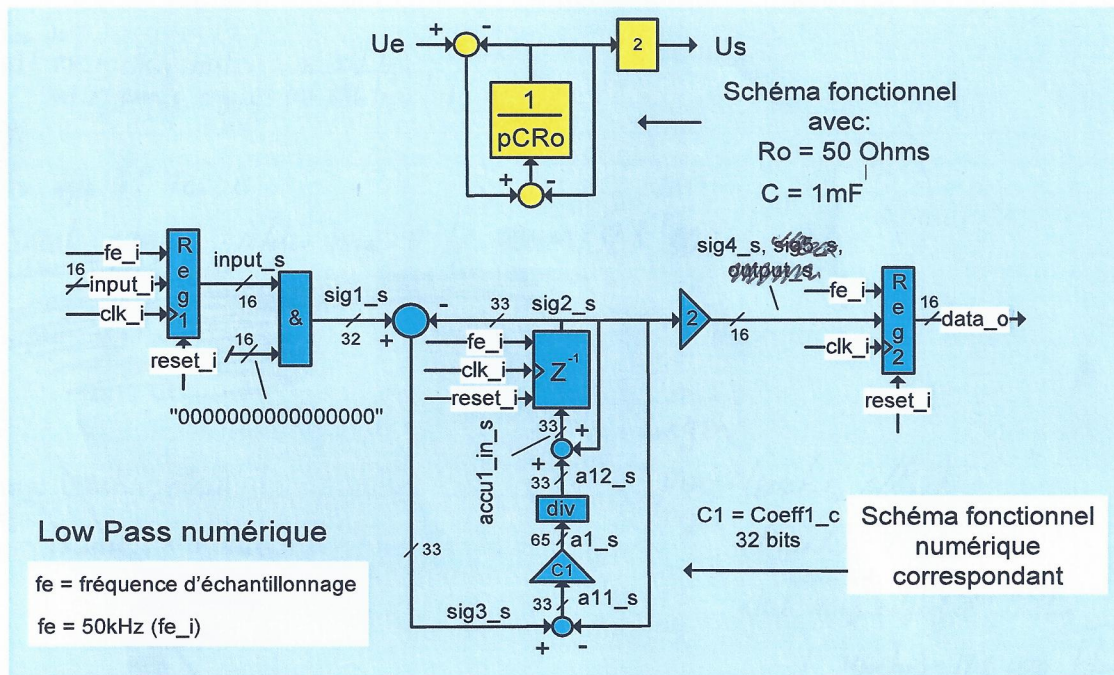




### Problème 3 : Filtre à intégrateur (6 pts)

2

Soit le filtre numérique ci-dessous :



- a) Dessinez le schéma passif (résistance, capacité, inductance) analogue équivalent de ce filtre numérique, en illustrant les différentes étapes.

6/ 1 pt

- b) A quoi sert ce facteur de 2 à la sortie du filtre ?

Pour compenser la perte de 6dB sur le diviseur 1/1 pt

- c) Calculez le coefficient C1 = Coeff1\_c et commentez les différentes opérations

en virgule flottante :  $C_1 = \frac{T_F}{T_n} = \frac{50kHz}{0.05s} = 1'000'000$

constant de temps :  $\tau_n = C \cdot R_o = 1mF \cdot 50\Omega = 0.05s$

en virgule fix :  $coeff_1 = C_1 \cdot 2^{32} = 42'99'96'67'29'60'00'000$

0/ 1 pt

- d) Ecrivez le code VHDL de ce filtre numérique

1/ 3 pts

$$1m = 60'000$$

$$55 = 5000$$

$$T = \frac{1}{F} = 1ms \Rightarrow 35 = 3000$$

## Probleme 1

```
library ieee;
use ieee.std-logic-1164.all;
use ieee.numeric_std.all;
```

```
entity Fen is
port ( en-1, clk-i, reset-i : in std-logic;
      vert-o, jaune-o, rouge-o : out std-logic );
end entity;
```

architecture logic of Fen is

```
signal counter3 : integer range 0 to 128'000;
signal counter5 : integer range 0 to 5000;
signal counter60 : integer range 0 to 60000;
signal en3 : std-logic;
en5;
en60;
```

juste 127  
-1

```
begin
process (reset-i, clk-i)
begin
if reset-i = '1' then
counter3 <= 0;
counter5 <= 0;
counter60 <= 0;
else if rising-edge (clk-i) then
if en3 = '1' or counter3 = 3000 then
counter3 <= 0;
else if en5 = '1' then
counter3 <= counter3 + 1;
end if;
if en5 = '0' or counter5 = 5000 then
counter5 <= 0;
else if en5 = '1' then
counter5 <= counter5 + 1;
end if;
if en60 = '0' or counter60 = 60000 then
counter60 <= 0;
else if en60 = '1' then
counter60 <= counter60 + 1;
end if;
end if;
end process;
```

begin

```
process (reset-i, clk-i)
begin
```

```
if reset-i = '1' then
counter3 <= 0;
else if rising-edge (clk-i) then
if counter3 = 128'000 then
counter3 <= 0;
else
counter3 <= counter3 + 1;
end if;
end if;
```

end if;

end process;

```
vert-o <= '1' when counter3 < 60'000 else '0';
jaune-o <= '1' when (counter3 >= 60'000 and
counter3 < 65'000) OR
counter3 >= 125'000 else '0';
```

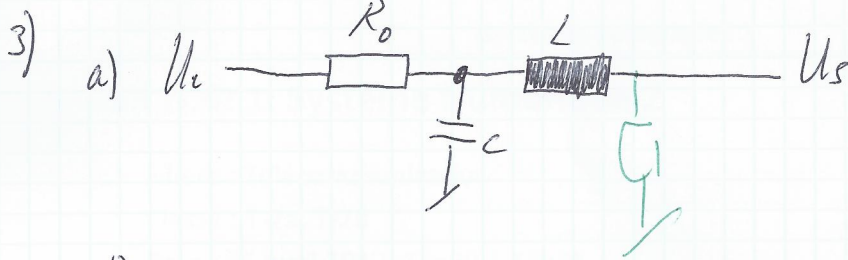
```
rouge-o <= '1' when counter3 >= 65'000 else '0';
end architecture;
```

en-1 - 1/2

✓







d) entity filter is

port(  
 $re-i, clk-i, reset-i$  : in std\_logic;  
 $input-i$  : in std\_logic\_vector (15 downto 0);  
 $data-o$  : out std\_logic\_vector (15 downto 0));

end entity;

architecture logic of filter is

signal input-s, ~~reset~~,  $sig4-s$  : std\_logic\_vector (15 downto 0);  
 $sig1-s$  : std\_logic\_vector (31 downto 0);  
 $sig2-s, sig3-s, sig12-s$  : std\_logic\_vector (32 downto 0);

begin

process ( $reset-i, clk-i$ )

begin

if  $reset-i = '1'$  then

$input-s \leq (\text{others} \Rightarrow '0')$ ;  $data-o \leq (\text{others} \Rightarrow '0')$ ;  $sig2-s \leq (\text{others} \Rightarrow '0')$ ;

elsif rising-edge( $clk-i$ ) then  $data-o \leq sig2-s$ ;

$input-s \leq input-i$ ;

end if;

$sig1-s \leq "0000\ 0000\ 0000\ 0000"$  and  $input-s$ ;

$sig3-s \leq sig1-s - sig2-s$ ;

end architecture;

signed

(✓)

incomplete (-2)