

Travail écrit 2: Système Numérique 1

2016/2017

Filière : Télécommunication

Classe : T-2a, T-2d

Date : 22 décembre 2016, 13h00 à 14h35

Professeur : Fabio Cunha

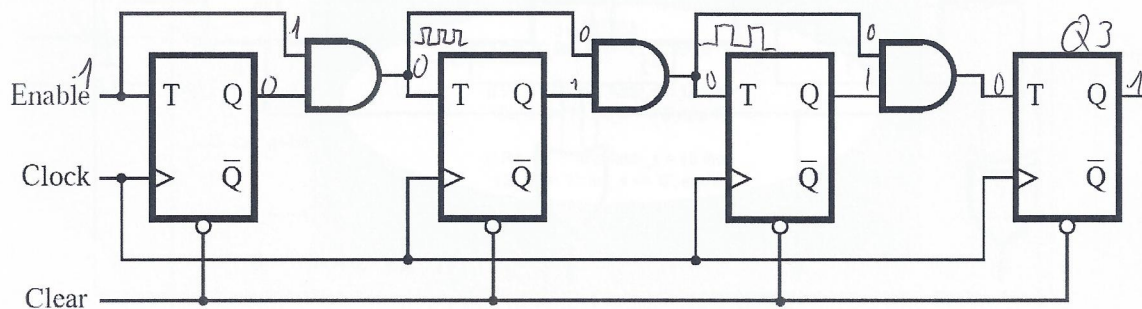
Nom et prénom : [REDACTED]

Points : 14/20

Note : 4.7

Problème 1: Structure based on Toggle Flip-flop (6 pts)

Soit la structure suivante, basée sur la bascule T, appelée bascule Toggle :

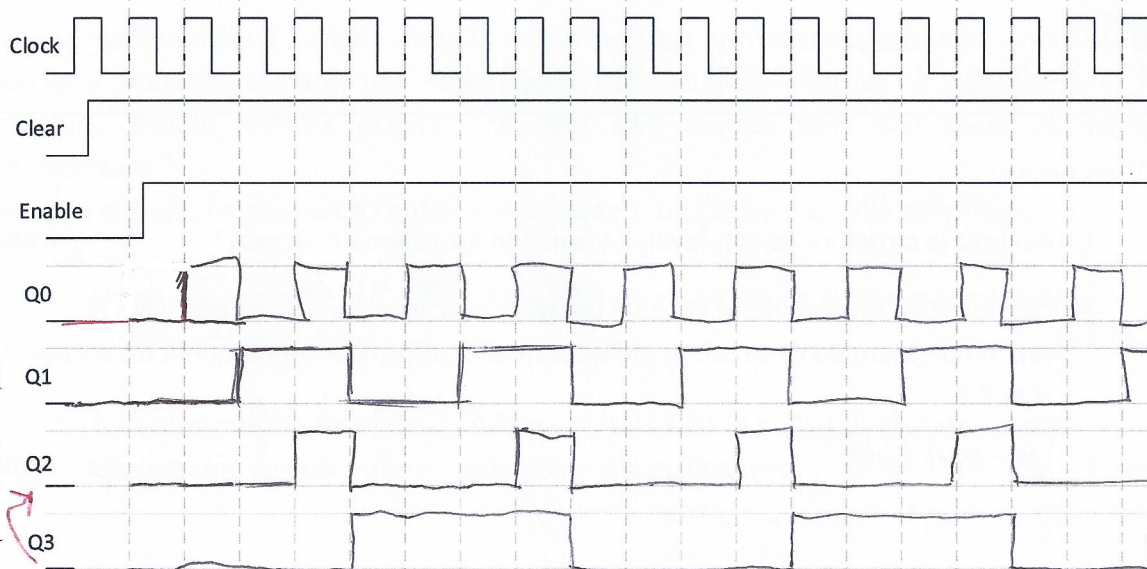


Vous devez :

1. Complétez le chronogramme ci-dessous.

2,5/4 pts

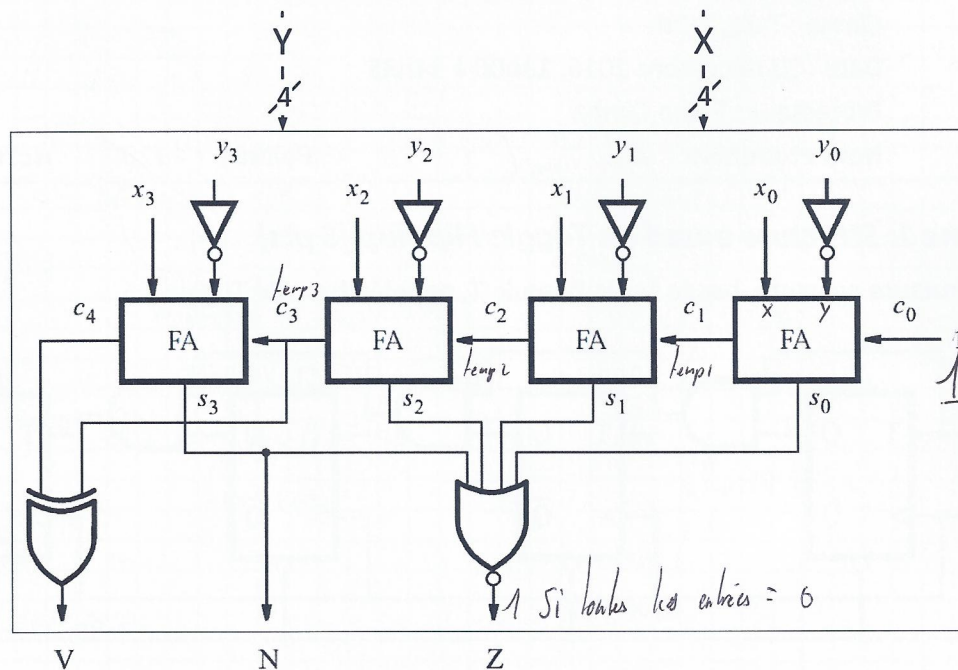
2. Quelle est la fonction du circuit ? Soyez clair dans votre réponse. => voir rec'de l'inter 2 pts



~~dans le cas, x=4 donc 2^4=16 donc~~

Problème 2: Structure based on full adder (6 pts) 3.5

Soit la structure suivante basée sur un composant que vous connaissez bien, le full-adder:



```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY fulladd IS
    PORT(Cin, x, y : IN STD_LOGIC;
          s, Cout : OUT STD_LOGIC);
END fulladd;

ARCHITECTURE LogicFunc OF fulladd IS
BEGIN
    s <= x XOR y XOR Cin;
    Cout <= (x AND y) OR (Cin AND x) OR (Cin AND y);
END LogicFunc;
    
```

Vous devez :

1. Réalisez la structure (le top-level) en fonction du schéma ci-dessus : 2.5 / 4 pts
2. Quelle est la fonction de ce circuit ? Décrivez avec le maximum de précision le fonctionnement de ce circuit et indiquez quelles informations indiquent les sorties V, N et Z ?

addition deux bits quatre bits

opération $x + y$

$Z = 1$ si l'addition = 0000... \rightarrow zero

$N =$ msb du résultat \rightarrow négatif

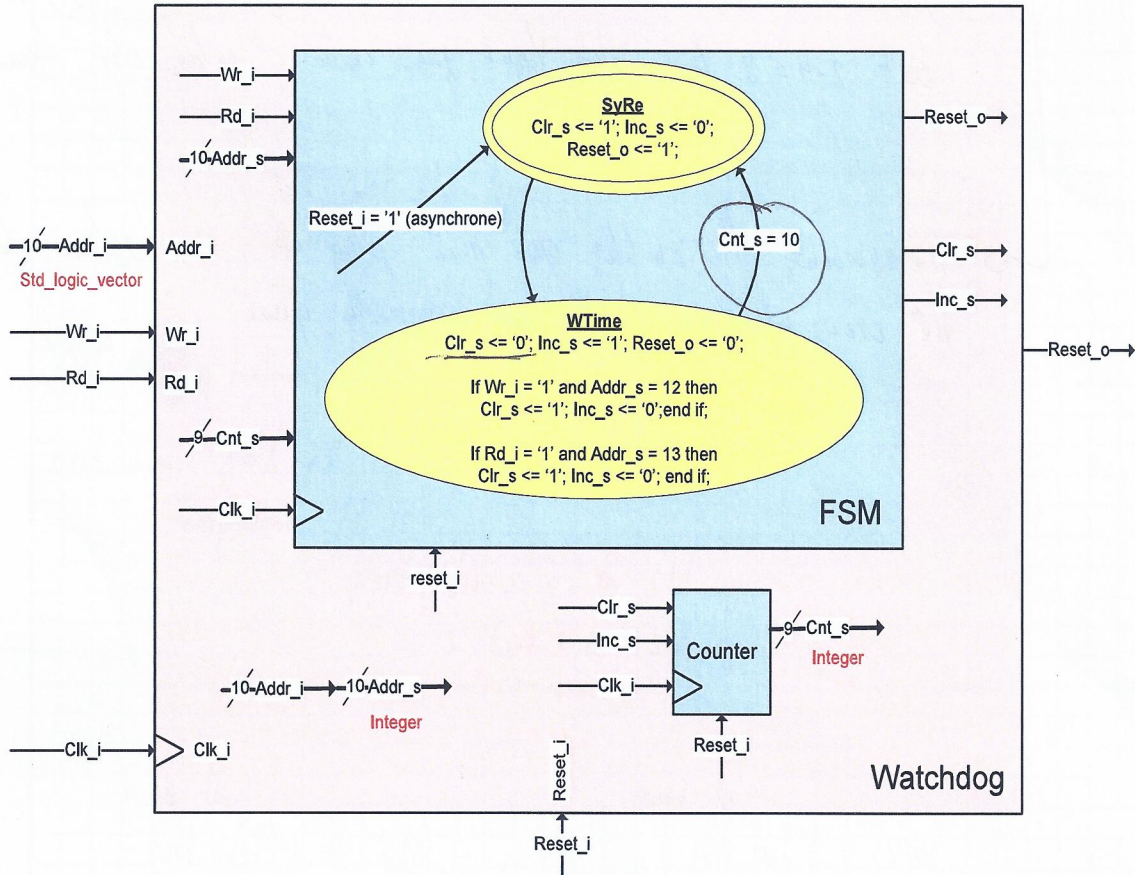
$V = 0$ si le résultat tient sur trois bits \rightarrow overflow

1 / 2 pts

Problème 3: FSM Watchdog (8 pts)

que architecture

Soit le circuit numérique ci-dessous :



Cette machine d'états « FSM », avec le « Counter » est un Watchdog modélisé en VHDL. Le circuit « Watchdog » n'a qu'une seule entité. Tous les blocs internes (le « Counter » et la machine d'états « FSM » proprement dite) sont décrits dans une seule et même architecture.

Pour ce qui suit, ne pas écrire l'entité « Watchdog », on suppose qu'elle est déjà écrite.

1. Est-ce que la machine d'états « FSM » est un automate de Moore ou de Mealy ?

Justifiez votre réponse.

1/ 1 pt

2. Écrivez le modèle VHDL de la machine d'états FSM (6 pts) et du Counter (1 pt).

Déclarez les signaux internes spécifiques à l'architecture.

5/ 7 pts

Réponse ex 1 : le but est d'avoir un nombre de toggle x pairs. le ~~clock~~ signal de sortie de la dernière bascule est un clock de période $2x$ fois plus lent que le clock d'entrée des toggle.

Dans notre cas, $x = 4$ bascules donc le clock de sortie est $2 \cdot 4 = 8$ fois plus lent que celui d'entrée des bascules.

↳ néanmoins, il semble que avec plus de 4 toggle, cela ne crée plus un clock mais n'importe quoi

(✓)

~~entity iee~~ ;

library ieee;

use ieee.std_logic_1164.all;

use ieee.numeric_std.all;

entity toplvl is

port (y: in std_logic_vector (3 down 0);
x: " " " " " " ;

V: out std_logic;

N: " " " " ;

Z: out std_logic);

end ~~entity~~ toplvl;

architecture arch of toplvl is

component fa is

port (Cin, x, y: in std_logic;

S, Cout: out std_logic);

end component;

begin

map1: fa

port map (X => X(0),

y => ^{not} y(0),

Cin => '1',

Cout => temp1,

S => a);

map2: fa

port map (X => X(1),

y => y(1), Cout => temp2,

Cin => temp1,

S => b);

map3: fa

port map (X => X(2), y => y(2),

Cin => temp2, Cout => temp3,

S => c);

map4: fa

port map (X => X(3), y => y(3),

Cin => temp3, Cout => temp4,

S => d);

Z <= not (a or b or c or d);

N <= ~~d~~; ~~c~~; ~~-1~~

V <= temp4 XOR temp3;

end architecture;

Signal temp1, temp2, temp3, temp4,
a, b, c, d: std_logic;

3)

1) Machine de Mealy car les entrées sont sur la fonction combinatoire de sortie.

2) architecture arch of fsm is

type elat is (SyRe, WTime);
signal futur, current : elat;
signal inc-s, clr-s;
signal cnt-s

```
begin
  Addr-s <= integer(unsigned(Addr-i))
  process (reset-i, clk-i)
  begin
    if reset-i = '1' then
      futur <= SyRe;
    elsif rising-edge(clk-i) then
      futur <= current;
    end if;
  end process;
```

```
process (cnt-s)
begin
  if cnt-s = 10
    futur <= SyRe;
  else
    futur <= WTime;
  end if;
end process;
```

```
process (Wr-i, Addr-s, Rcl-i)
begin
  case current is
    when SyRe =>
      Clr-s <= '1';
      Inc-s <= '0';
      Reset-o <= '1';
    when WTime =>
      if (Wr-i = '1' and Addr-s = 12)
        or (Rcl-i = '1' and Addr-s = 13) then
        Clr-s <= '1'; Inc-s <= '0';
      else
        Clr-s <= '0'; Inc-s <= '1';
      end if;
      reset-o <= 0;
    end case;
  end process;
```

```
process (Reset-i, CLK-i)
begin
  if Reset-i = '1' then
    cnt-s <= 0;
  elsif rising-edge(CLK-i) then
    if Clr-s = '1' then
      cnt-s <= 0;
    elsif Inc-s = '1' then
      cnt-s <= cnt-s + 1;
    end if;
  end if;
end process;
```

end architecture;