

**L3 STIC ESET**  
**FPGA AND VHDL (ETRS 501 et ETRS504)**  
**Session 2**  
**Date : 13/06/2024**  
**Durée : 1h30**

**Règles pour l'épreuve :**

- Tous les documents sont autorisés

**Exercice 1 :**

On souhaite décrire un circuit ayant la table de vérité suivante :

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

1. Ecrire l'architecture en description flot de données de ce circuit.
2. Décrire ce même circuit mais en utilisant une description comportementale.
3. Quelle est la différence ces deux descriptions ?
4. En VHDL, à quoi sert un process ?
5. Réécrire l'architecture suivante en utilisant un process

```
architecture Arch of Q2 is
begin
    with Sel select
        sortie <= a when "00"|"01",
                b when others;
end Arch;
```

6. Quelle est la différence entre l'instruction « with...select... » et l'instruction « <=...when... » ?
7. Sur la carte DE1, on trouve de la RAM à l'intérieur du FPGA, à l'extérieur du FPGA, ou à la fois à l'intérieur et à l'extérieur du FPGA ?
8. Sur la carte DE1, l'horloge à 50 MHz est-elle générée à l'intérieur du FPGA, à l'extérieur du FPGA, ou à la fois à l'intérieur et à l'extérieur du FPGA ?

Dans un FPGA, il existe des broches particulières pour appliquer une horloge extérieure.

9. En quoi ces broches sont différentes des broches « normales » ?

10. En TP vous pouviez utiliser le fichier DE1\_pin\_assignments.csv. A quoi sert-il ?

### **Exercice 2 :**

On étudie le code VHDL de l'annexe A.

Q1. Quelles sont les entrées et les sorties du circuit ? Quelle est leur taille ?

Q2. De combien de composants est composé ce circuit ?

Q3. Dessiner le schéma du circuit en y mettant le plus d'information possible (nom et taille des entrées/sorties, noms et taille des fils internes, nom et type des composants, ...).

Q4. Que fait ce circuit ?

Q5. Décrire le même circuit en VHDL mais sans utiliser de composants.

### **Exercice 3 :**

L'annexe B présente le code décrivant le circuit P\_GENERATOR.

1. Quelles sont les entrées et les sorties du circuit ?

2. Dessiner le graph de la machine à états décrite.

3. La remise à zéro implémentée est-elle synchrone ou asynchrone ? Quelle est la différence entre les deux ?

4. Dessiner les chronogrammes obtenus pour les entrées et les sorties après avoir lancé la simulation décrite dans l'annexe C. Préciser dans quel état se trouve le système tout au long des chronogrammes.

5. Quelle fonction réalise le circuit ? De quelle durée est l'impulsion ?

## Annexe A

```
library IEEE;
use IEEE.std_logic_1164.all;

entity adder4 is
    port( a,b : in std_logic_vector(3 downto 0); Ci : in std_logic;
          s : out std_logic_vector(3 downto 0); Co : out std_logic);
end adder4;

architecture structure of adder4 is
    component adder
        port(X,Y,Cin : in std_logic;
             Cout, Sum : out std_logic);
    end component;
    signal c : std_logic_vector(3 downto 1);
    begin
        fa0: adder port map (a(0), b(0), Ci, c(1), s(0));
        fa1: adder port map (a(1), b(1), c(1), c(2), s(1));
        fa2: adder port map (a(2), b(2), c(2), c(3), s(2));
        fa3: adder port map (a(3), b(3), c(3), Co, s(3));
    end structure;
```

```
library IEEE;
use IEEE.std_logic_1164.all;

entity adder is
    port(X,Y,Cin : in std_logic;
          Cout, Sum : out std_logic);
end adder;

architecture arch of adder is
begin
    Sum<=X xor Y xor Cin;
    Cout<=(Cin and (X xor Y)) or (X and Y);
end arch;
```

## ANNEXE B

```

library IEEE;
use IEEE.std_logic_1164.all;

entity P_GENERATOR is
port (
    CLK : in std_logic;
    RESET : in std_logic;
    TRIG : in std_logic;
    PULSE : out std_logic);
end P_GENERATOR;

architecture STATE_MACHINE of P_GENERATOR is
type PULSEGEN_STATE_TYPE is (IDLE, GEN_PULSE_A, GEN_PULSE_B, END_PULSE, RETRIGGER);
signal CURRENT_STATE, NEXT_STATE: PULSEGEN_STATE_TYPE;
signal COUNT : integer range 0 to 31;
constant WIDTH : integer range 0 to 31 := 4;

begin
STATE_MACH_PROC : process (CURRENT_STATE, TRIG, COUNT)
begin
    case CURRENT_STATE is
        when IDLE => if TRIG='1' then
            NEXT_STATE <= GEN_PULSE_A;
        end if;
        when GEN_PULSE_A => if COUNT = WIDTH then
            NEXT_STATE <= END_PULSE;
        elsif TRIG='0' then
            NEXT_STATE <= GEN_PULSE_B;
        end if;
        when END_PULSE => if TRIG = '1' then
            NEXT_STATE <= IDLE;
        end if;
        when GEN_PULSE_B => if TRIG = '1' then
            NEXT_STATE <= RETRIGGER;
        elsif COUNT=WIDTH then
            NEXT_STATE <= IDLE;
        end if;
        when RETRIGGER => NEXT_STATE <= GEN_PULSE_A;
        when OTHERS => NEXT_STATE <= NEXT_STATE;
    end case;
end process STATE_MACH_PROC;

PULSE_PROC : process (CLK, RESET) -- sensitivity list
begin
    if RESET = '1' then
        PULSE <= '0';
        COUNT <= 0;
        CURRENT_STATE <= IDLE;
    elsif (clk='1' and clk'event) then
        CURRENT_STATE <= NEXT_STATE;
        case NEXT_STATE is
            when IDLE => PULSE <= '0';
                COUNT <= 0;
            when GEN_PULSE_A => PULSE <= '1';
                COUNT <= COUNT + 1;
            when END_PULSE => PULSE <= '0';
                COUNT <= 0;
            when GEN_PULSE_B => PULSE <= '1';
                COUNT <= COUNT + 1;
            when RETRIGGER => COUNT <= 0;
            when OTHERS => COUNT <= COUNT;
        end case;
    end if;
end process PULSE_PROC;

end STATE_MACHINE;

```

## Annexe C

```
library IEEE;
use IEEE.std_logic_1164.all;

entity STATE_MACHINE_TB is
end STATE_MACHINE_TB;

architecture ARC_STATE_MACHINE_TB of STATE_MACHINE_TB is
    component P_GENERATOR
        port ( CLK : in std_logic;
              RESET : in std_logic;
              TRIG : in std_logic;
              PULSE : out std_logic);
    end component;

    signal CLK : std_logic;
    signal RESET : std_logic;
    signal TRIG : std_logic;
    signal PULSE : std_logic;

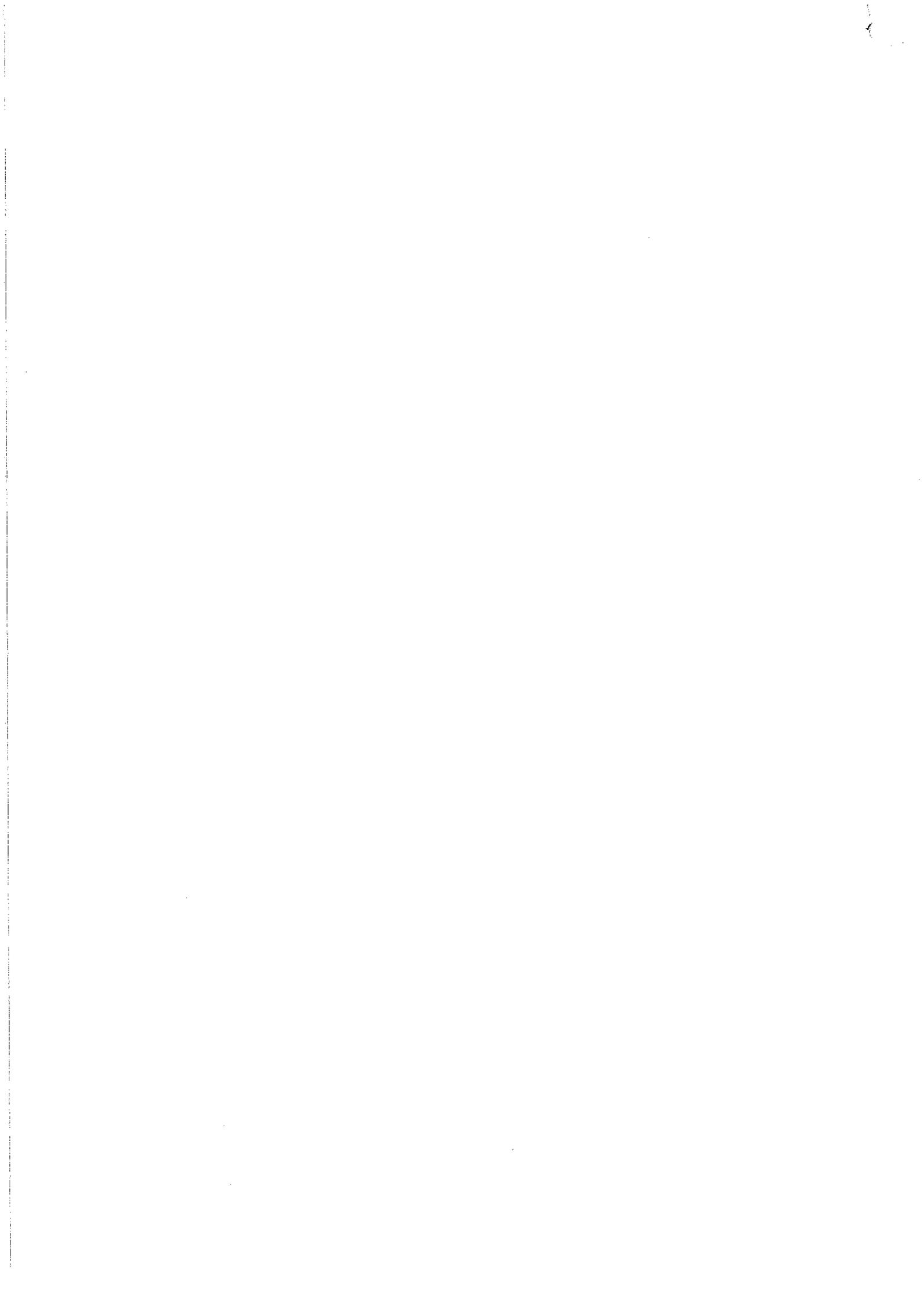
begin
    P_GENERATOR port map( CLK, RESET,TRIG,PULSE);

    RESET <='1','0' after 5 ns;

    CREATE_CLOCK: process (clk)
    begin
        if clk <= 'U' then clk <= '0' after 1 ns;
        else clk <= not clk after 1 ns;
        end if;
    end process CREATE_CLOCK;

    CREATE_PULSE: process (TRIG)
    begin
        TRIG <= '0' ,
            '1' after 10 ns,
            '0' after 21 ns;
    end process CREATE_PULSE;

end ARC_STATE_MACHINE_TB;
```



Épreuve du mardi 11 juin 2024 – session 2

Partie 1

EXERCICE 1 :

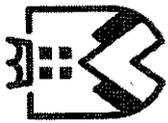
1. Trouver une primitive de  $A = \int \left( 4x^3 + \frac{1}{x^2} - \sqrt{x} \right) \cdot dx$
2. Trouver une primitive de  $B = \int (1+3x) \cdot e^x \cdot dx$
3. Après avoir fait une division de polynômes, donner une primitive de  $C = \int \frac{2x^3}{x^2-4} \cdot dx$

EXERCICE 2 :  $z$  est un nombre complexe et on pose  $j$  tel que  $j^2 = -1$

4. Soit  $z_1 = \frac{-4}{(1+j\sqrt{3})}$ , écrire  $z_1$  sous la forme  $a+jb$  puis sous sa forme trigonométrique et enfin sous sa forme exponentielle.
5. Calculez  $z_1^3$  puis placez le dans le plan complexe ainsi que  $z_1$ .

EXERCICE 3 :  $j$  est tel que  $j^2 = -1$ . Soit la fonction  $x \rightarrow H(x)$  où  $x$  est un réel et  $H(x)$  un nombre complexe, s'écrivant  $H(x) = \frac{jx}{1+jx}$

6. Déterminer le module et l'argument de  $H(x)$  que l'on nommera  $|H(x)|$  et  $\arg(H(x))$ .
7. Pour  $x=0, 1, 1, 2, 10$  et  $100$  donner les valeurs de  $|H(x)|$  et  $\arg(H(x))$ .
8. Pour  $x$  variant de  $0$  à l'infini étudier les valeurs limites et les asymptotes de  $|H(x)|$ . On veillera à bien donner l'expression analytique des asymptotes pour le module.
9. Donner l'expression analytique de  $20 \times \log_{10} |H(x)|$ . Donner ensuite l'expression analytique des asymptotes de  $20 \times \log_{10} |H(x)|$  lorsque  $x$  tend vers  $0$  et  $x$  tend vers l'infini.
10. Tracer  $|H(x)|$  en dB dans un diagramme de Bode à l'aide du papier semi-log joint à ce sujet pour  $x$  variant de  $0.01$  à  $100$ .



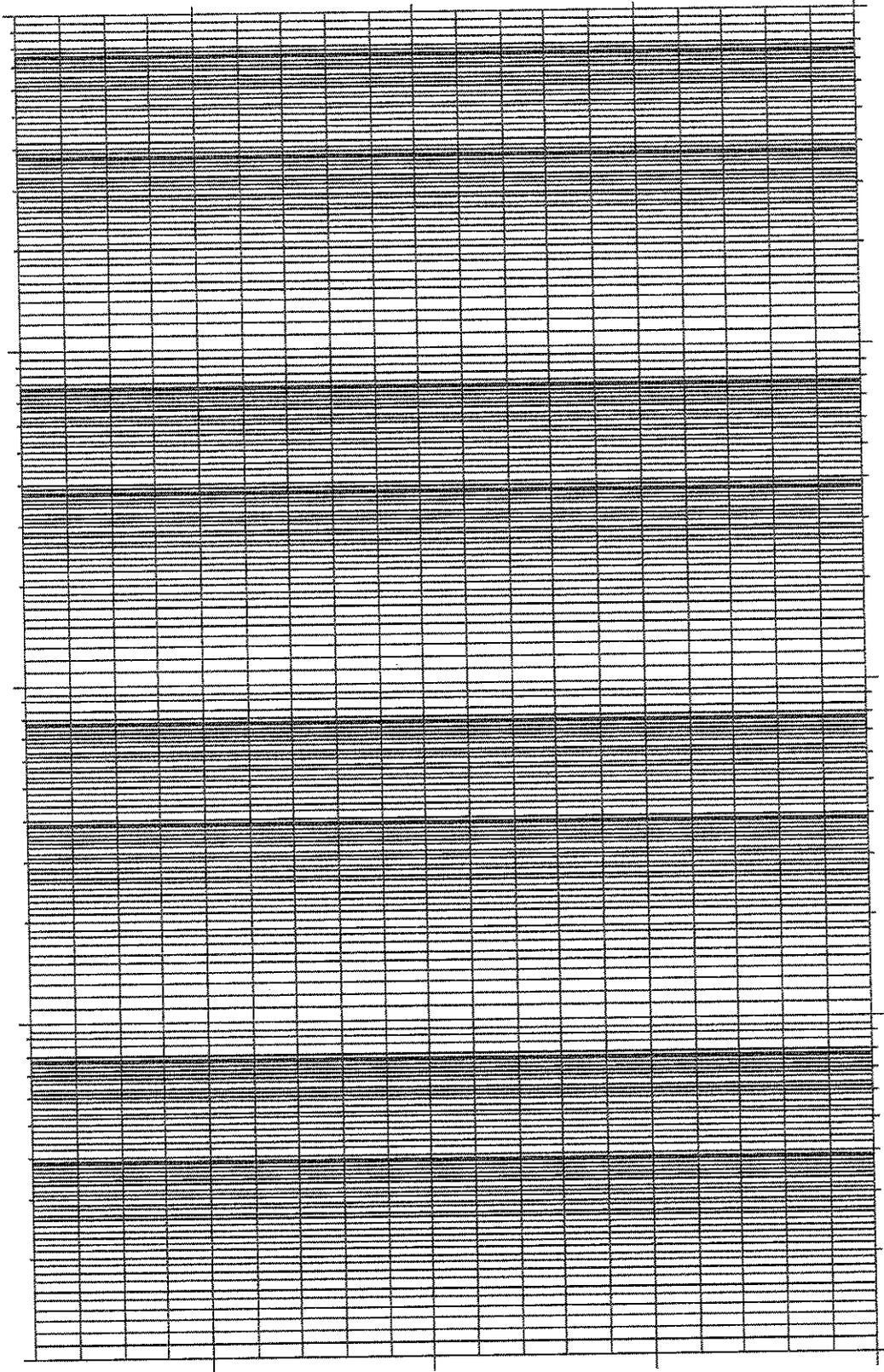
UNIVERSITÉ  
SAVOIE  
MONT BLANC

**Licence Sciences et technologies**

L3 ESET et TRI Rattrapage du Semestre 5

Unité d'enseignement UE 522

Module ETRS522-ETRS Bases des Mathématiques pour l'ingénieur



**Licence ESET 3<sup>ème</sup> année**

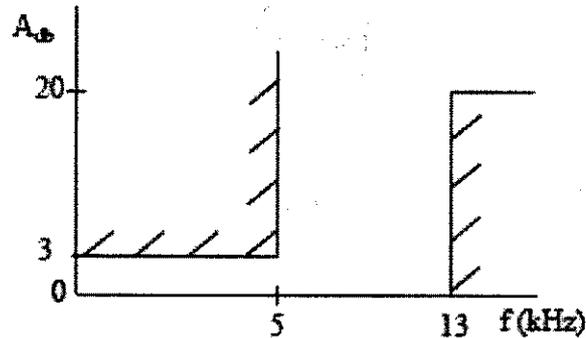
Epreuve de ETRS 602 - session 2

Date : mercredi 19 juin 2024

Durée : 1h30.

**1) Question de cours : Filtre de Butterworth**

Le but est de réaliser un filtre de **Butterworth** dont le gabarit est le suivant :



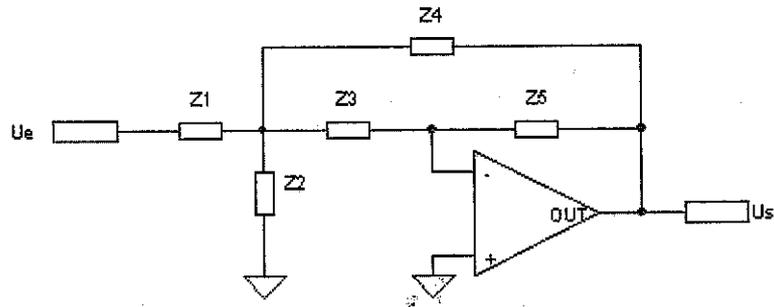
- De quel type de filtre s'agit-il ? Quelles sont les particularités d'un filtre de Butterworth ?
- Donner les paramètres  $A_{min}$ ,  $A_{max}$ ,  $f_p$  et  $f_a$  associés aux bandes passante et atténuée du filtre.
- Déterminer l'ordre du filtre correspondant au gabarit ci-dessus.
- Donner l'expression générale de la fonction de transfert d'un tel filtre.

On donne la table des polynômes suivante pour la valeur de  $\epsilon$  correspondant au cas présent :

N	$A(p)$ ( $p=j\omega$ ) pour $A_{max}=3$ dB
1	$p+1$
2	$(p^2+1.414 p +1)$
3	$(p+1) (p^2+ p +1)$
4	$(p^2+0.765 p +1) (p^2+1.848 p +1)$
5	$(p+1) (p^2+0.618 p +1) (p^2+1.618 p +1)$

## 2) Etude d'un filtre

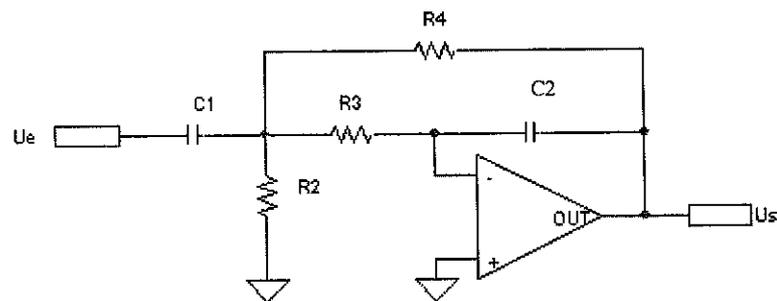
Soit la structure général d'un filtre :



- a) Montrer que la fonction de transfert  $H(j\omega) = \frac{V_s(j\omega)}{V_e(j\omega)}$  d'une telle structure s'exprime comme suit en fonction des impédances  $Z_i$  :

$$H(j\omega) = \frac{-1}{\frac{Z_3}{Z_5} + \frac{Z_1}{Z_5} + \frac{Z_1}{Z_4} + \frac{Z_1 Z_3}{Z_2 Z_5} + \frac{Z_1 Z_3}{Z_4 Z_5}}$$

On fabrique, sur la base de la structure précédente, le filtre suivant :



- b) Donner l'expression de la fonction de transfert complexe de ce filtre.  
 c) Faire l'étude asymptotique du module et de l'argument de  $H(j\omega)$ .

On prendra  $R_2=R_3=R_4=R$  et  $C=C_1=C_2$ .

- d) Tracer le module et l'argument de  $H(j\omega)$  dans le plan de Bode.  
 En déduire de quel type de filtre il s'agit.

**L3 STIC ESET**  
**FPGA et VHDL (ETRS 501 et ETRS504)**  
**Session 1**

**Date : 19/12/2023**

**Durée : 1h30**

**Règles pour l'épreuve :**

- Tous les documents papier sont autorisés

**Exercice 1 : Cherchez l'erreur !**

Un étourdi a écrit le code VHDL suivant et le compilateur lui signale 11 erreurs.

**Q1. Retrouvez au moins 8 erreurs dont vous êtes sûrs et expliquez pourquoi la syntaxe est fautive.**

```
library ieee;
use ieee.std_logic_1164.all;

entity exam is
port( clk : std_logic;
      Erreur : in bit ;
      a : in signed (3 downto 0);
      b : in signed (3 downto 0);
      operation : in bit;
      Nb_erreurs : out integer;
      add : out signed (3 downto 0));
end exam;

architecture 1A of exambis is
variable var : integer;

begin
  process(clk)
  begin
    if clk'event and clk=1 then
      Nb_erreurs <= Nb_erreurs + 1 when erreur = '1' else Nb_erreurs;
      if operation = '0' then
        var <= 2;
      else
        var := 6;
      end if;
      add <= a + b + var;
    end if;
  end process;
end 1A;
```

## **Exercice 2 : Etude d'un circuit**

On donne en annexe les fichiers vhdl d'un projet décrivant un circuit.

### 1. Etude du Top Level

**Q1.** Quel type de description est utilisé dans ce fichier ?

**Q2.** Dessinez de façon claire le schéma décrit par ce fichier. Faites apparaître les entrées et sorties du circuit et des composants. Précisez bien les tailles des bus utilisés et nommez tous les signaux.

### 2. Etude du composant bloc

**Q3.** Ce composant est décrit en utilisant un paramètre générique. Pourquoi utiliser un tel paramètre ?

**Q4.** Quelle est la valeur par défaut du paramètre générique ?

Pour comprendre le fonctionnement de ce bloc, on propose de le simuler. Pour cette simulation, on fixera le paramètre générique à 4.

**Q5.** Quelles sont les entrées/sorties de ce composant ? Indiquez leur taille.

**Q6.** Proposez des chronogrammes à appliquer sur toutes les entrées pour effectuer une simulation pertinente du composant.

**Q7.** Ecrivez le fichier vhdl permettant de simuler le composant selon les chronogrammes de la question précédente.

**Q8.** En considérant que l'on applique sur les entrées les chronogrammes de la question Q4, dessinez le(s) chronogramme(s) que l'on peut attendre sur la (les) sortie(s) après une simulation fonctionnelle.

**Q9.** Quelle est la fonction du composant bloc ?

### 3. Etude du composant FSM

**Q10.** Quels sont les états de cette machine à états ?

**Q11.** Dessinez le diagramme d'état.

**Q12.** Quelles lignes de code pourrait-on ajouter et à quel endroit du fichier si on voulait améliorer la machine avec un reset asynchrone qui la force dans l'état « attente » ?

### 4. Description du composant parité

**Q13.** En suivant les contraintes données par le Top Level, écrivez l'entity du composant parité.

**Q14.** Ecrivez l'architecture de ce composant sachant qu'il doit réaliser, sur front montant d'horloge, l'opération OU EXCLUSIF entre deux entrées. Un reset synchrone force la sortie à 0.

**Q15.** Dans une liaison série, à quoi sert le bit de parité ? Comment est-il déterminé ? Vous partirez du principe que la parité est paire.

**Q16.** Expliquez pourquoi un OU EXCLUSIF peut être utilisé pour calculer la parité des entrées.

### 5. Fonctionnement du circuit

**Q17.** Complétez les chronogrammes de l'annexe 4 (on négligera les temps de propagation) en considérant que  $D_{in} = "1010110"$  et que à  $t=0$  le circuit est dans l'état « attente ».

**Q18.** Expliquez le fonctionnement de ce circuit.