

**L3 STIC ESET
FPGA ET VHDL (ETRS 501_ESET)
FPGA (ETRS 507_ESET)**

Date : 05/01/2021

Durée : 2h

Règles pour l'épreuve :

- Tout document papier autorisé

Exercice 1 : Questions de cours

- Q1. 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?
- Q2. Sur la carte DE1, l'horloge à 50 MHz est générée à l'intérieur du FPGA, à l'extérieur du FPGA, ou à la fois à l'intérieur et à l'extérieur du FPGA?
- Q3. En TP vous pouviez utiliser le fichier DE1_pin_assignments.csv. A quoi sert-il ?
- Q4. Qu'est-ce qu'un « Top Level » ?
- Q5. Quelle est la différence entre l'instruction « with...select... » et l'instruction « <=...when... » ?
- Q6. On souhaite concevoir un circuit ayant la table de vérité suivante :

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

- Ecrire l'architecture en description flot de données de ce circuit.
- Décrire ce même circuit mais en utilisant une description comportementale.

Exercice 2 : Machine à état et simulation

On étudie le circuit dont le code vhdl est en annexe A.

- Q1. Dessiner le diagramme d'état.
- Q2. On décide de tester ce circuit avec une simulation fonctionnelle. Dessiner les chronogrammes (Annexe C) de Clk, reset, X et S correspondant au testbench en Annexe B. Vous préciserez dans quel état se trouve le système à chaque coup d'horloge.

Q18. Que fait ce circuit ?

La simulation étant correcte, le projet est téléchargé dans la carte et testé. Ca ne fonctionne pas comme prévu.

Q19. Proposer une explication possible à ce dysfonctionnement.

Une fois tous les problèmes réglés et ayant constaté que son circuit est opérationnel, l'auteur décide de le modifier en remplaçant le composant compteur par le composant compteur_bis décrit en annexe H (les autres fichiers restant identiques).

Q20. Compléter les chronogrammes en annexe I en prenant en compte ce changement.

ANNEXE A

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

entity FSM is
port(Clk,X,reset : in std_logic;
      S : out std_logic_vector(1 downto 0));
end FSM;

ARCHITECTURE arch of FSM is
  Type mes_etats is (E0,E1,E2,E3);
  Signal etat: mes_etats;
  BEGIN
    process (clk)
    begin
      if (clk' event and clk ='1') then
        if reset='1' then
          etat <= E0;
        else
          case etat is
            when E0 => if X='1' then etat <=E1;
                       else etat <=E0; end if;
            when E1 => if X='1' then etat <=E2;
                       else etat <=E1; end if;
            when E2 => if X='1' then etat <=E3;
                       else etat <=E2; end if;
            when E3 => if X='1' then etat <=E0;
                       else etat <=E3; end if;
          end case;
        end if;
      end if;
    end process;

    S <= "00" when etat = E0 else "01" when etat = E1 else "10" when
etat = E2 else "11";
  end arch;
```

ANNEXE B

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

entity fsm_tb
end fsm_tb;

architecture Arch of fsm_tb is

component fsm
port(Clk,X,reset : in std_logic;
      S : out std_logic_vector(1 downto 0));
end component;

signal sig_Clk,sig_X,sig_reset : std_logic := '0';
signal sig_S : std_logic_vector(1 downto 0);

begin

    test : fsm
    port map (sig_Clk, sig_X, sig_reset,sig_S);

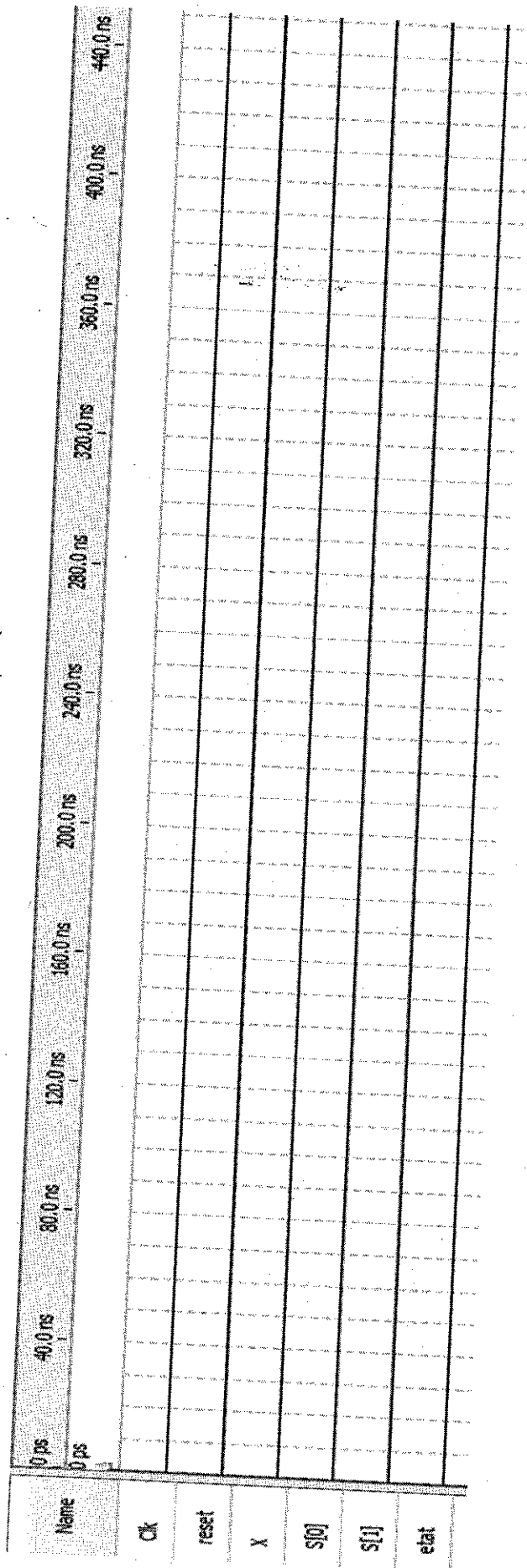
    process(sig_Clk)
    begin
        wait for 20 ns;
        sig_Clk<=not(sig_Clk);
    end process;

    sig_reset<='1' after 1 ns, '0' after 50 ns;
    sig_X<='0', '1' after 90 ns, '0' after 110 ns, '1' after 210 ns,
    '0' after 310 ns, '1' after 370 ns, '0' after 390 ns ;

    assert (now<400 ns) report "fin de simulation" severity failure;

end Arch;
```

ANNEXE C



ANNEXE D : application.vhd

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

entity application is
port(      CLOCK_50 : in std_logic;
          KEY : in std_LOGIC_VECTOR(3 downto 0);
          numero : out integer range 0 to 9;
          LEDR : out std_logic_vector(9 downto 0));
end application;

architecture arch of application is
    component tick_gen
    port( clk, reset : in std_logic;
          tick : out std_logic);
    end component;

    component compteur
    port( clk, c_enable,reset : in std_logic;
          numero : out integer range 0 to 9);
    end component;

    component ROM
    port(      address : in integer range 0 to 15;
          clock : in std_logic;
          q : out std_logic_vector (9 downto 0));
    end component;

    signal enable : std_logic;
    signal adresse : integer range 0 to 15;
    signal reset : std_logic;

begin

    reset<= not (KEY(0));

    comp1 : tick_gen
    port map(CLOCK_50,reset,enable);

    comp2 : compteur
    port map (CLOCK_50,enable,reset,adresse);

    comp3 : ROM
    port map(adresse, CLOCK_50,LEDR);

end arch;
```

ANNEXE E : Tick_gen.vhd

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

entity Tick_gen is
port(      clk, reset : in std_logic;
        tick : out std_logic );
end Tick_gen;

architecture arch of Tick_gen is
constant TICK_MAX : integer := 3;
begin
    process(clk)
        variable tickCount : integer range 0 to TICK_MAX-1;
    begin
        if (clk'event and clk='1') then
            if (reset = '1') then
                tickCount := 0;
                tick <= '0';
            elsif (tickCount = TICK_MAX-1) then
                tickCount := 0;
                tick <= '1';
            else
                tickCount := tickCount + 1;
                tick <= '0';
            end if;
        end if;
    end process;
end arch;
```

ANNEXE F : compteur.vhd

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

entity compteur is
port(
    clk, c_enable, reset : in std_logic;
    numero : out integer range 0 to 9;
);
end compteur;

architecture arch of compteur is
begin
    if rising_edge(clk) then
        if c_enable = 1 then
            numero <= numero + 1;
        end if;
    end if;
end arch;
```

ANNEXE G

	0 ps	80.0 ns	160.0 ns	240.0 ns	320.0 ns	400.0 ns	480.0 ns	560.0 ns	640.0 ns	720.0 ns	800.0 ns	880.0 ns	960.0 ns
Name													
CLOCK_50													
KE[0]													
Tick													
LEDX[9]													
LEDX[8]													
LEDX[7]													
LEDX[6]													
LEDX[5]													
LEDX[4]													
LEDX[3]													
LEDX[2]													
LEDX[1]													
LEDX[0]													

ANNEXE H

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

entity compteur_bis is
port( clk, c_enable, reset : in std_logic;
      numero : out integer range 0 to 9);
end compteur_bis;

architecture arch of compteur_bis is
begin
    process(clk)
        variable var : integer range 0 to 9;
        variable sens : std_logic := '0';
    begin
        if rising_edge(clk) then
            if c_enable = '1' then
                if sens = '0' then
                    var := var + 1;
                    if var = 9 then
                        sens := '1';
                    end if;
                else
                    var := var - 1;
                    if var = 0 then
                        sens := '0';
                    end if;
                end if;
            end if;
        end if;
    end process;
end arch;
```

ANNEXE I

