Université Tunis El-Manar	**Année Universitaire** : 2016-2017
Faculté des Sciences de Tunis	**Module :** Synthèse en VHDL de systèmes embarqués

TD N° 2

Exercice 1.

1)

Description VHDL par flot de données d'un additionneur complet 1-bit

```
library ieee;
use ieee.std_logic_1164.all;
Entity add1bit is port(
      A,B,Cin : in std_logic;
      Cout,S : out std_logic);
end Entity;
Architecture arch_add1bit of add1bit is
Begin
      Cout<=(A and B) or (B and Cin) or (A and
      Cin);
      S <= A xor B xor Cin;
End Architecture;
```

2)

Description structurelle générique d'un additionneur série N bits.

```
library ieee;
use ieee.std_logic_1164.all;
Entity AdditionneurSerie is
      generic(N : natural range 1 to 128:=8);
      port (
            A,B : in std_logic_vector(N-1 downto 0);
            S : out std_logic_vector(N-1 downto 0);
            Cout : out std_logic);
end Entity;
Architecture arch_addserie of AdditionneurSerie is
component add1bit port (
      A,B,Cin : in std_logic;
      Cout,S : out std_logic);
end component;
signal i_c : std_logic_vector(N-2 downto 0);
Begin
      add0 : add1bit port
map(A=>A(0),B=>B(0),Cin=>'0',Cout=>i_c(0),S=>S(0));
      addN_1 : add1bit port
map(A=>A(0),B=>B(0),Cin=>i_c(N-2),Cout=>Cout,S=>S(N-
1));
b_for: for i in 1 to N-2 generate

addi: add1bit port map(A=>A(i),B=>B(i),Cin=>i_c(i-
1),Cout=>i_c(i),S=>S(i));
end generate;
End Architecture;
```

Exercice 2.

1) Simulation impossible car il s'agit de deux instructions concurrentes d'écriture sur le même signal.

2) Dans un process, les instructions sont interprétées séquentiellement. Par conséquent, F est toujours égal à "not A".

Exercice 3.

```
library ieee;
use ieee.std_logic_1164.all;
entity RegistreGen is
      generic(t : integer range 1 to 256:=8);
      port (
            LOAD,AUT,clk,Rst,SDI,DIR : in std_logic;
            D : in std_logic_vector(t-1 downto 0);
            SDO : out std_logic;
            Q : out std_logic_vector(t-1 downto 0)
      );
end entity;
architecture arch_reggen of RegistreGen is
signal temp : std_logic_vector(t-1 downto 0);
Begin
```

```
      Q<=temp;
      process(clk,Rst)
      Begin
            if Rst='1' then
                  temp<=(others=>'0');SDO<='0';
            elsif clk'event and clk='1' then
                  if AUT='1' then
                        if LOAD='1' then
                              temp<=D;
                        elsif DIR='1' then
                              SDO<=temp(t-1);
                              temp<=temp(t-2  downto  0)
& SDI;
                        else
                              SDO<=temp(0);
                              temp<=SDI    &    temp(t-1
downto 1);
                        end if;
                  end if;
            end if;
      End process;
End architecture;
```