Cours 1: Notions de VHDL

I) Introduction

- VHDL :

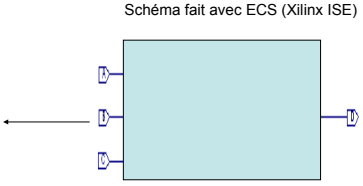
Very high speed integrated circuit **H**ardware **D**escription **L**anguage

- Langage de description de circuit et de tests de circuits
- Les éléments décrits fonctionnent simultanément, ce qui est très différent des langages séquentiels que vous connaissez!!!

I.1.a Description d'interconnexions

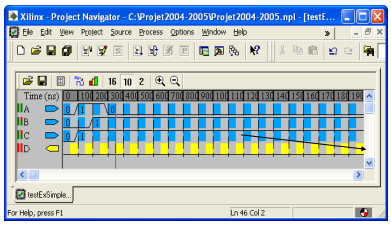
```
entity exsimple is
  port ( A : in   std_logic;
         B : in   std_logic;
         C : in   std_logic;
         D : out  std_logic);
end exsimple;

architecture BEHAVIORAL of exsimple is
   attribute BOX_TYPE   : STRING ;
   signal S1 : std_logic;
   component XOR2
      port ( I0 : in   std_logic;
             I1 : in   std_logic;
             O  : out  std_logic);
   end component;

   component AND2
      port ( I0 : in   std_logic;
             I1 : in   std_logic;
             O  : out  std_logic);
   end component;

begin
  BLOC1 : XOR2
      port map (I0=>B, I1=>A, O=>S1);
   BLOC2 : AND2
      port map (I0=>C, I1=>S1, O=>D);

end BEHAVIORAL;
```

Schéma fait avec ECS (Xilinx ISE)



I.2.b Description de tests

```
LIBRARY IEEE;USE IEEE.STD_LOGIC_1164.ALL;USE
IEEE.NUMERIC_STD.ALL;USE IEEE.STD_LOGIC_TEXTIO.ALL;
USE STD.TEXTIO.ALL;

ENTITY testExSimple IS
END testExSimple;
```



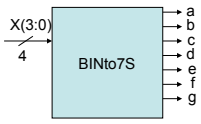

```
BEGIN
-- ------------------
A <= '0';
B <= '0';
C <= '0';
-- ------------------
WAIT FOR 100 ns;
A <= '1';
C <= '1';
-- ------------------
WAIT FOR 100 ns; -- Time=200 ns
B <= transport '1';
-- ------------------
WAIT FOR 100 ns; -- Time=300 ns
A <= transport '0';
-- ------------------
WAIT FOR 150 ns; -- Time=450 ns
-- ------------------
...
```

II Description fonctionnelle

a. Description vue de l'extérieur (ENTITY)

- Nom
- Entrées et sorties

Exemple

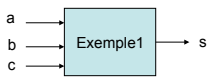


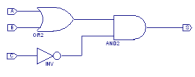

```
ENTITY BINto7S is
  PORT (
        X : in std_logic_vector(3 downto 0) ;
        A,B,C,D,E,F,G : out std_logic) ;
END BINto7S;
```

II.2 Description d'un composant combinatoire

a. Exemple simple

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY  Exemple1 IS
   Port ( a : in std_logic;
          b : in std_logic;
          c : in std_logic;
          s : out std_logic );
end Exemple1;
architecture Behavioral of Exemple1 is
begin
          s<=(a or b) and not c;
end Behavioral;
```





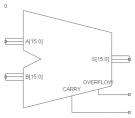
b. MUX 4 vers 1, 16 bits

```
library IEEE;
…
entity MUX4_1_16bits is
 Port(E0,E1,E2,E3 : in std_logic_vector(15 downto 0);
      SEL : in std_logic_vector(1 downto 0);
      S : out std_logic_vector(15 downto 0));
end MUX4_1_16bits;
architecture Behavioral of MUX4_1_16bits is
begin
 process (E0,E1,E2,E3,SEL)
 begin
          CASE SEL IS
          WHEN "00" => S <= E0;
          WHEN "01" => S <= E1;
          WHEN "10" => S <= E2;
          WHEN OTHERS => S <= E3;
          end case;
 end process;
end Behavioral;
```

SEL	S
00	E0
01	E1
10	E2
11	E3

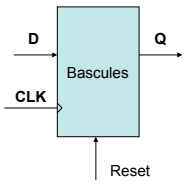
c. Additionneur signé

```
library IEEE;
…
entity addsub16 is
   Port ( A : in std_logic_vector(15 downto 0);
          B : in std_logic_vector(15 downto 0);
          S : out std_logic_vector(15 downto 0);
          CARRY : out std_logic;
          OVERFLOW : out std_logic);
end addsub16;
architecture Behavioral of addsub16 is
signal S_interne : std_logic_vector(16 downto 0);
begin
     CARRY<=S_interne(16);
     S<=S_interne(15 downto 0);
     S_interne <= (A(15) & A)  + (B(15) & B) ;
     OVERFLOW <= (A(15) AND B(15) AND NOT S_interne(15)) OR (NOT A(15)
     AND NOT B(15) AND S_interne(15));
end Behavioral;
```



3. Bascule D

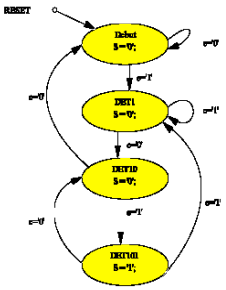
```
library IEEE;
…
entity basculeD is
   Port ( D : in std_logic_vector(15 downto 0);
          CLK,Reset : in std_logic;
       Q : out std_logic_vector(15 downto 0));
end basculeD;
architecture Behavioral of basculeD is
begin
      process (CLK,Reset)
      begin
           if (Reset='1') then
                   Q<=(others =>'0');
         elsif (CLK'event and CLK='1') then
                   Q<=D;
           end if;
      end process;
end Behavioral;
```

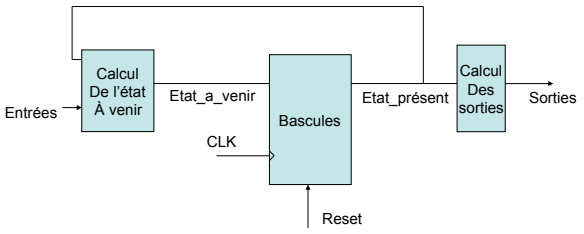


Reset	CLK	Q
1	X	0
0	Front	D
0	Cte	Qp

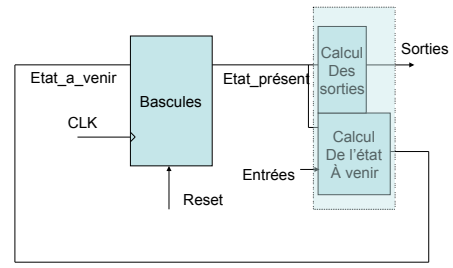
4 Automate

- Exemple, détection de 101 avec recouvrement
- Etape 1 : Graphe



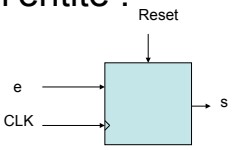


Réalisation



Réalisation modifiée, 2 blocs à décrire

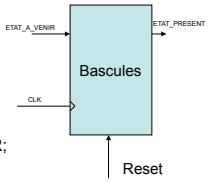
Description de l'entité :




```
library IEEE;
...
entity Detection101 is
   Port (      e : in std_logic;
               CLK : in std_logic;
          RESET : in std_logic;
               s : out std_logic);
end Detection101;
```

Description du comportement

```
architecture Behavioral of Detection101 is
   type StateType is (debut,det1,det10,det101);
   signal ETAT_A_VENIR,ETAT_PRESENT : StateType;
begin
BASCULES :
   process (CLK,RESET)
   begin
      if (RESET='1') then
            ETAT_PRESENT<=debut;
      elsif (CLK'EVENT AND CLK='1') THEN
            ETAT_PRESENT<=ETAT_A_VENIR;
      end if;
   end process BASCULES;
```




```
CALCUL_SORTIES_ET_ETAT_A_VENIR :
   process (ETAT_PRESENT,e)
   begin
      case ETAT_PRESENT is
            when debut =>
                  s<='0';
                  if (e='1') then
                        ETAT_A_VENIR<=det1;
                  else
                        ETAT_A_VENIR<=debut;
                  end if;
            when det1 =>
                  s<='0'; …

      end case;
   end process CALCUL_SORTIES_ET_ETAT_A_VENIR;
end Behavioral;
```