



COMPTE RENDU TP2

Circuit Programmable FPGA

MODELSIM

ModelSim est un outil de Mentor Graphics. Il fournit un environnement complet de simulation et débogage pour les designs complexes en ASIC et en FPGA. Il supporte plusieurs langages de description, dont le Verilog, le SystemVerilog, le VHDL et le SystemC.

Cedric bassole

BASCULE_RS

• Code

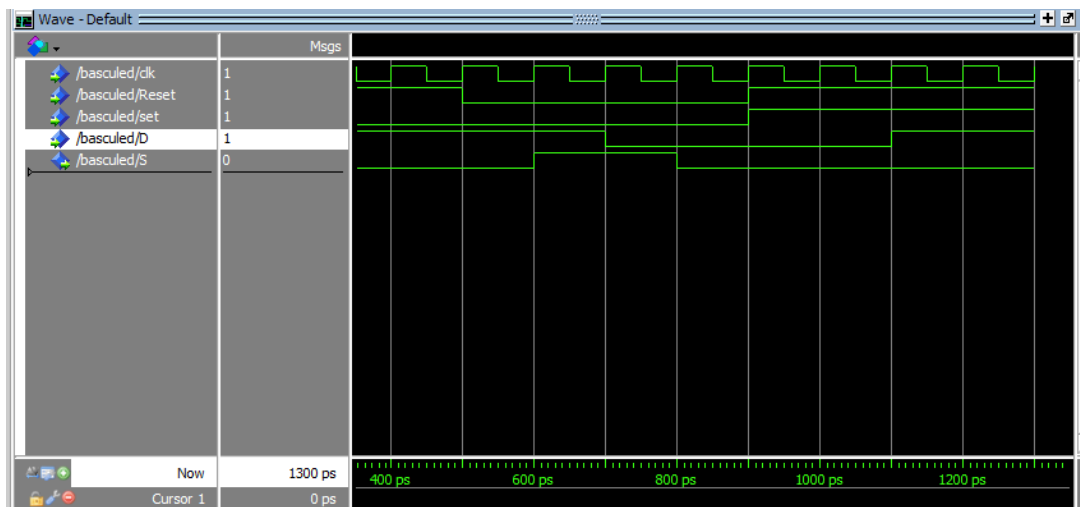
Une bascule ou un verrou est un circuit logique doté d'une ou plusieurs entrées et d'une ou deux sorties. Les changements d'état de la sortie sont déterminés par les signaux appliqués aux entrées et le type d'opérateur mis en œuvre.

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

entity BasculeD is
    port(
        clk:in std_logic;
        Reset:in std_logic;
        set:in std_logic;
        D:in std_logic;
        S: out std_logic);
end BasculeD;

architecture behavioral of BasculeD is
begin
    process(D,clk,set,Reset)
    begin
        if Reset='1' then
            S<='0';
        elsif set='1' then
            S<='1';
        elsif (clk'event and clk='1') then
            S <=D;
        end if;
    end process;
end behavioral;
```

• Simulation Wave



4 - Non, il n'est pas possible d'écrire le code VHDL de ce circuit sans process. Le process est utilisé pour modéliser le comportement asynchrone de la bascule.

5- Dans le cas de process, les sensibilités sont spécifiées par la liste des signaux qui sont utilisés dans la clause de sensibilité. Dans ce cas, la clause de sensibilité est process (clk, reset, set), ce qui signifie que le process est déclenché à chaque fois qu'il y a un changement sur l'un de ces signaux.

6- Le RESET est synchrone car il est pris en compte lors du front montant de l'horloge.

7- Voici les chronogrammes des entrées / sorties pour différentes possibilités :

- Si D est maintenu à 0 et SET et RESET sont à 0, alors la sortie S ne change pas et prend reste D=0.
- Si D est maintenu à 1 et SET et RESET sont à 0, alors la sortie S prend la Valeur de D au front montant de clk
- Si RESET est activé (passe à 1), alors la sortie S est réinitialisée à 0.
- Si SET est activé (passe à 1), alors la sortie Q est positionnée à 1.

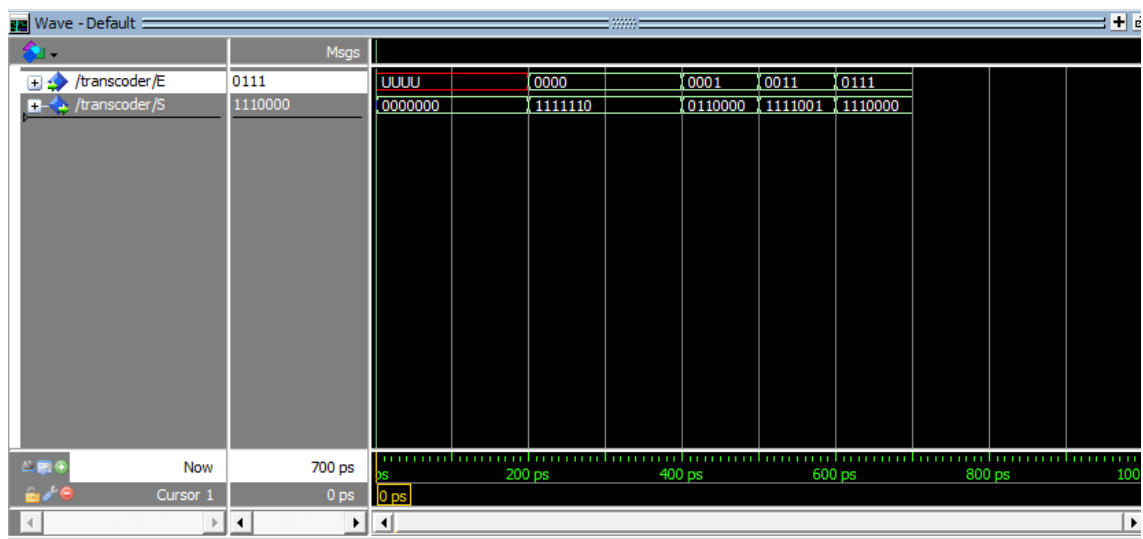
Transcodeur

• Code

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

entity Transcoder is
    port(
        E: in std_logic_vector(3 downto 0);
        S: out std_logic_vector(6 downto 0));
end Transcoder;
architecture behavioral of Transcoder is
begin
    process(E)
    begin
        case E is
            when "0000" => S<="1111110";
            when "0001" => S<="0110000";
            when "0010" => S<="1101111";
            when "0011" => S<="1111001";
            when "0100" => S<="0110011";
            when "0101" => S<="1011011";
            when "0110" => S<="1011111";
            when "0111" => S<="1110000";
            when "1000" => S<="1111111";
            when "1001" => S<="1111011";
            when others => S<="0000000";
        end case;
    end process;
end behavioral;
```

• Simulation Wave



Compteur Transcodeur

- Code Transcodeur incrémental

```
library ieee;

use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_unsigned.all;

entity Counting_Transcoder is
    port(
        H: in std_logic;
        O: out std_logic_vector(6 downto 0));
end Counting_Transcoder;

architecture behavioral of Counting_Transcoder is
    signal C: std_logic_vector(3 downto 0);

    component Horloge
        port(
            H: in std_logic;
            S: out std_logic_vector(4 downto 1):="0000");
    end component;

    component Transcoder
        port(
            E: in std_logic_vector(3 downto 0);
            S: out std_logic_vector(6 downto 0));
    end component;

    begin
        u1: Horloge port map (H,C);
        u2: Transcoder port map (C,O);
    end behavioral;
```

- Simulation

