

Projet de VHDL

Réalisation d'un Fréquencemètre

JÉRÉMIE FOURMANN (Promo 2013 - Électronique)

MAXIME MORIN (Promo 2013 - Électronique)

26 mai 2012



Plan

1	Objectifs	2
1.1	Rappel du cahier des charges	2
1.2	Nos choix	2
2	Conception du système	3
2.1	Présentation du système	3
2.2	Présentation des sous Modules	3
2.2.1	Schéma bloc	3
2.2.2	Machine d'état	3
2.2.3	Résultat de simulation	3
2.3	Performance du système	3
3	Bilan	4
A	Manuel d'utilisateur	5
A	Extrait de code VHDL	6

1 Objectifs

1.1 Rappel du cahier des charges

1.2 Nos choix

2 Conception du système

2.1 Présentation du système

2.2 Présentation des sous Modules

2.2.1 Schéma bloc

2.2.2 Machine d'état

2.2.3 Résultat de simulation

2.3 Performance du système

3 Bilan

A Manuel d'utilisateur

A Extrait de code VHDL

```
1  -----
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:      18:18:06 05/06/2012
6  -- Design Name:
7  -- Module Name:      nombre_24bits_generateur - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Poue le test de notre module d'affichage
14 --
15 -- affiche une puissance de deux par seconde
16 --
17 -----
18 library IEEE;
19 use IEEE.STD_LOGIC_1164.ALL;

21 -- Uncomment the following library declaration if using
22 -- arithmetic functions with Signed or Unsigned values
23 --use IEEE.NUMERIC_STD.ALL;

25 -- Uncomment the following library declaration if instantiating
26 -- any Xilinx primitives in this code.
27 --library UNISIM;
28 --use UNISIM.VComponents.all;

29
30 entity nombre_24bits_generateur is
31     Port ( rst : in  STD_LOGIC;
32           clk : in  STD_LOGIC;
33           nombre : out STD_LOGIC_VECTOR (23 downto 0));
34 end nombre_24bits_generateur;

35
36 architecture Behavioral of nombre_24bits_generateur is
37 type etat is (init, calc_nombre, attente);

38
39 signal etatf : etat; --etat futur
40 signal etatp : etat; --etat present
41
42 signal timer : STD_LOGIC_VECTOR (23 downto 0);
43
44 begin
45
46     --Bloc F
47     process(etatp, rst)
48     begin
49         if rst='0' then etatf <= init ;
50         else
51             case etatp is
52             when init =>
53
54                 when others => etatf <= init;
55
56             end case;
57         end if;
58     end process;
59
60     --Bloc M
61     process(clk, rst)
62     begin
63         if (clk'event and clk = '1') then etatp <= etatf ;
64         end if;
```

```

65 end process;

67 --Bloc G
process(clk)
69 begin
    if(clk'event and clk='1') then
71
        case etatp is
73         when init =>

75         when others =>

77         end case;

79 end if;
end process;

81 --autres process
83 process(step)
begin
85
end process;

87

89 end Behavioral;

```