Courses : **CE312**	**Report** **TM2 - Modulo N Counter**
Date :	24/11/2020
Students :	CELLARD Rémi - PIERSON-MAURY Damien
Subject:	Modulo N Counter

1. Introduction

In this report, we'll see how we can create a modulo N with a custom bits counter with VHDL code and simulate it in Vivado.

2. Code of the modulo N counter component

All the code for the component and the testbench are included at the end of the report, in the Annexes 1 and 2.
It's also included in the Annex 3, the manual chronogram of the testbench and the chronogram who's simulated in the Vivado environment at the Annex 4.

Annex 1 - Modulo N Counter code of the component

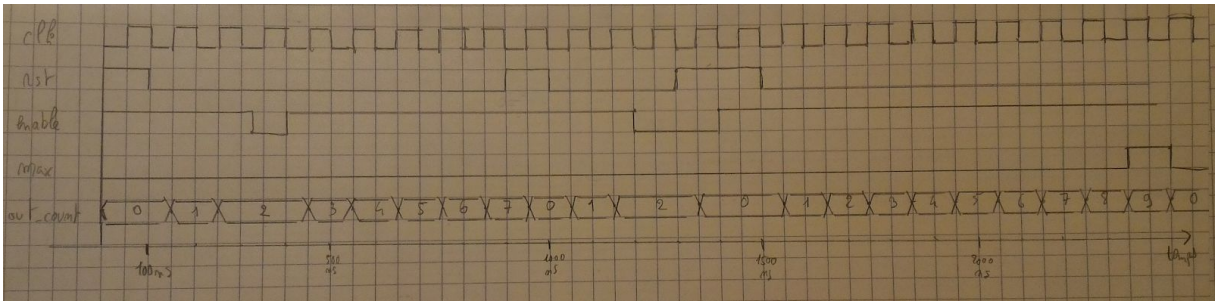
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity compteur_N is
generic (
        C_NB_BIT_COUNTER : integer;
        C_MODULO : integer
);
port (
        clk : in STD_LOGIC; -- clock signal
        rst : in STD_LOGIC; -- reset signal
        enable : in STD_LOGIC; -- enable signal
        max  : out STD_LOGIC; -- bit overflow
        out_count : out STD_LOGIC_VECTOR (C_NB_BIT_COUNTER-1 downto 0)
        -- count signal
);
end entity;

architecture archcompteur of compteur_N is
  signal maxtmp : STD_LOGIC := '0';
  signal outtmp : STD_LOGIC_VECTOR (C_NB_BIT_COUNTER-1 downto 0) :=
(others=>'0');
  -- two signal for made operations on it - because we cant on out an in signal
  begin
        process(clk) -- active when event happen on signal clk
        begin
        if (clk'event and clk='1') then
        maxtmp <= '0';
        if rst='1' then
        outtmp <= (others=>'0');
        elsif (enable='1') then -- no reset, counter enable
        outtmp <= std_logic_vector(unsigned(outtmp)+ 1); -- add 1 to unsigned
        if (unsigned(outtmp) = (C_MODULO-2)) then  -- max is reach
        maxtmp <= '1';
        elsif(unsigned(outtmp) = (C_MODULO-1)) then -- restart the counter
        outtmp <= (others=>'0');
        end if;
        end if;
        end if;
   end process;
        -- output assignment
        max <= maxtmp;
        out_count <= outtmp;
end architecture;
```

Annex 2 - Testbench code for the Modulo 16 Counter component

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity tb_compteur_N is
end tb_compteur_N;

architecture tb of tb_compteur_N is

  constant C_MODULO : integer := 10;
  constant C_NB_BIT_COUNTER : integer := 4;
  component compteur_N
  generic (
          C_NB_BIT_COUNTER : integer;
          C_MODULO : integer
  );
  port (
          clk : in STD_LOGIC;
          rst : in STD_LOGIC;
          enable : in STD_LOGIC;
          max  : out STD_LOGIC; -- bit overflow
          out_count : out STD_LOGIC_VECTOR (C_NB_BIT_COUNTER-1 downto 0)
          -- le compteur temporaire
  );
  end component;

  signal clk_tb, rst_tb, enable_tb, max_tb : STD_LOGIC;
  signal out_count_tb : STD_LOGIC_VECTOR (C_NB_BIT_COUNTER-1 downto 0);

  begin
          compteur: compteur_N generic map (C_NB_BIT_COUNTER=>C_NB_BIT_COUNTER,
C_MODULO=>C_MODULO) port map (clk=>clk_tb, rst=>rst_tb, enable=>enable_tb,
max=>max_tb, out_count=>out_count_tb);

          clock_10mhz_generation : process -- simulation of the clock signal
          begin
          clk_tb <= '0';
          wait for 50 ns;
          clk_tb <= '1';
          wait for 50 ns;
          end process;

          procreset: process -- simulation of the reset signal
          begin
          rst_tb <= '1';
          wait for 100 ns; -- we maintain the rst_tb signal on 100 ns
          rst_tb <= '0';
          wait for 800ns;
```

```
        rst_tb <= '1';
        wait for 100 ns; -- we maintain the rst_tb signal on 100 ns
        rst_tb <= '0';
        wait for 300ns;
        rst_tb <= '1';
        wait for 200ns;
        rst_tb <= '0';
        wait; -- puis on l'enlève définitivement
        end process;

        procenable: process -- simulation of the enable signal
        begin
        enable_tb <= '1';
        wait for 325 ns;
        enable_tb <= '0';
        wait for 75 ns;
        enable_tb <= '1';
        wait for 800ns;
        enable_tb <= '0';
        wait for 200 ns;
        enable_tb <= '1';
        wait;
        end process;

end architecture;
```

Annex 3 - Manual chronogram of the Testbench



Annex 4 - Chronogram of the Testbench in Vivado after simulation