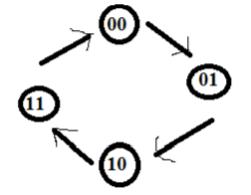
Lab 8 – VHDL

Name: Nhut Quang Tran

Group: NTIVIS14

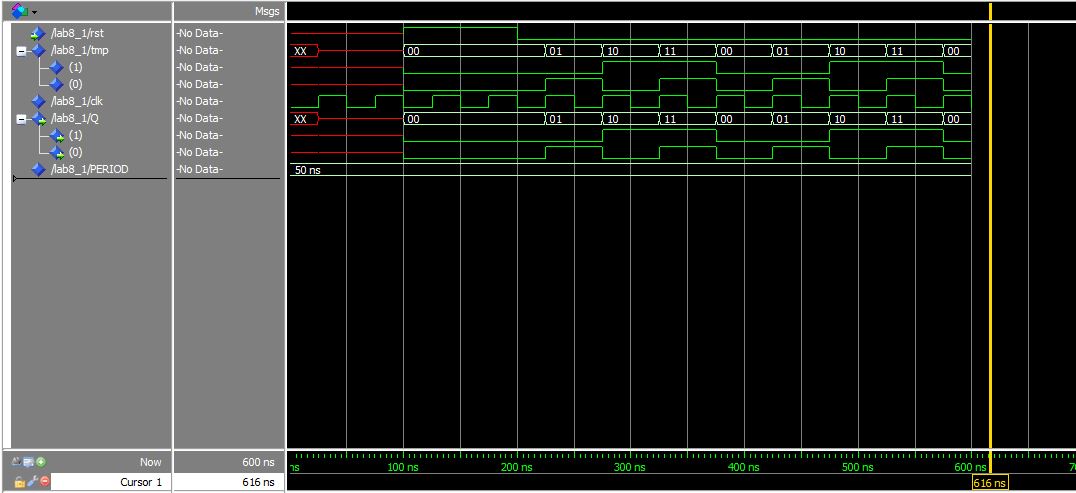
TURKU UNIVERSITY OF APPLIED SCIENCES

**Exercise 1: With the help of these model try build a VHDL model of 2-bit binary counter.**

**The code:**

|  |
| --- |
| LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  use IEEE.NUMERIC\_STD.all;  use IEEE.std\_logic\_unsigned.all;  ENTITY Lab8\_1 IS  PORT (rst: IN STD\_LOGIC;  Q: OUT STD\_LOGIC\_VECTOR(1 DOWNTO 0));  END Lab8\_1;  ARCHITECTURE BitCounter OF Lab8\_1 IS  constant PERIOD: time := 50ns;  signal clk : std\_logic := '0';  signal tmp:STD\_LOGIC\_VECTOR (1 downto 0);  BEGIN  P0:PROCESS(rst,clk)  BEGIN  clk <= not clk after PERIOD/2; -- Generating clock, the period = 50ns.  IF(rst = '1') then  tmp<="00";  ELSIF (clk'event and clk='1') THEN  tmp <= tmp + 1;  END IF;  END PROCESS P0;  Q <= tmp;  END BitCounter; |

**The simulating picture:**

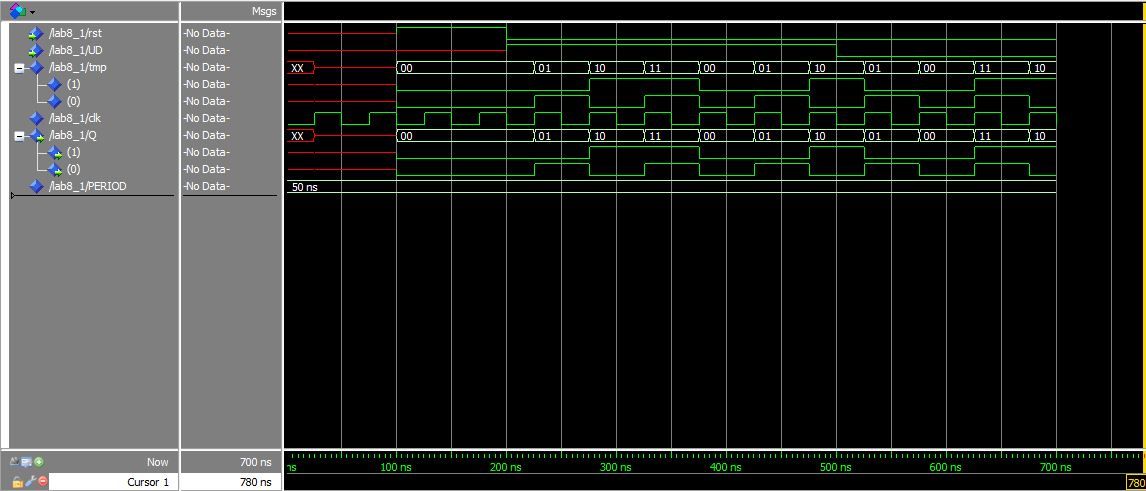
****

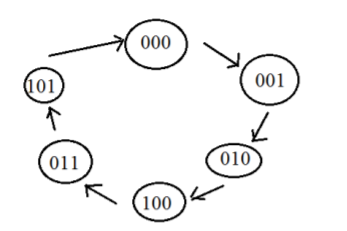
**Exercise 2:** **Build a VHDL model for 2-bit up/down counter.**

**The code:**

|  |
| --- |
| LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  use IEEE.NUMERIC\_STD.all;  use IEEE.std\_logic\_unsigned.all;  ENTITY Lab8\_2 IS  PORT (UD,rst: IN STD\_LOGIC;  Q: OUT STD\_LOGIC\_VECTOR(1 DOWNTO 0));  END Lab8\_2;  ARCHITECTURE BitCounter OF Lab8\_2 IS  constant PERIOD: time := 50ns;  signal clk : std\_logic := '0';  signal tmp:STD\_LOGIC\_VECTOR (1 downto 0);  BEGIN  P0:PROCESS(rst,UD,clk)  BEGIN  clk <= not clk after PERIOD/2; -- Generating clock, the period = 50ns.  IF(rst = '1') then  tmp<="00";  ELSIF (clk'event and clk='1') THEN  IF( UD = '1' ) THEN  tmp <= tmp + 1;    ELSE  tmp <= tmp - 1;    END IF;  END IF;  END PROCESS P0;  Q <= tmp;  END BitCounter; |

**The simulating picture:**

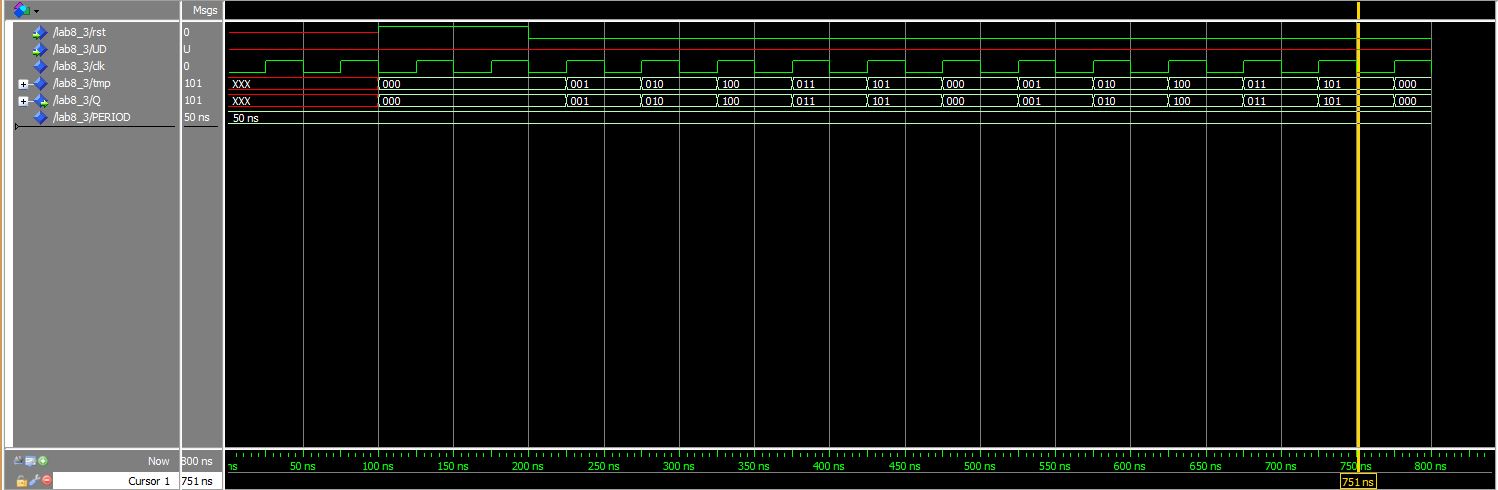
****

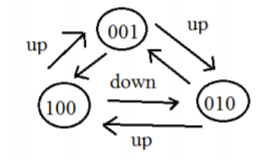
**3. Build a VHDL model for 3-bit counter which counting from 0 to 5 according to picture below.**

**The code:**

|  |
| --- |
| LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  use IEEE.NUMERIC\_STD.all;  use IEEE.std\_logic\_unsigned.all;  ENTITY Lab8\_3 IS  PORT (rst: IN STD\_LOGIC;  Q: OUT STD\_LOGIC\_VECTOR(2 DOWNTO 0));  END Lab8\_3;  ARCHITECTURE BitCounter OF Lab8\_3 IS  constant PERIOD: time := 50ns;  signal clk : std\_logic := '0';  signal tmp:STD\_LOGIC\_VECTOR (2 downto 0);  BEGIN  P0:PROCESS(rst,clk)  BEGIN  clk <= not clk after PERIOD/2; -- Generating clock, the period = 50ns.  IF(rst = '1') then  tmp<="000";  ELSIF (clk'event and clk='1') THEN  IF(tmp = "000") THEN  tmp <= tmp + 1;  ELSIF(tmp >= "001" ) THEN  tmp(1) <= tmp(0);  tmp(2) <= tmp(1);  tmp(0) <= tmp(2);  IF(tmp = "100") THEN  tmp <= "011";  ELSIF(tmp = "011") THEN  tmp <= "101";  ELSIF(tmp = "101") THEN  tmp <= "000";  END IF;  END IF;  END IF;  END PROCESS P0;  Q <= tmp;  END BitCounter; |

**The simulating picture:**

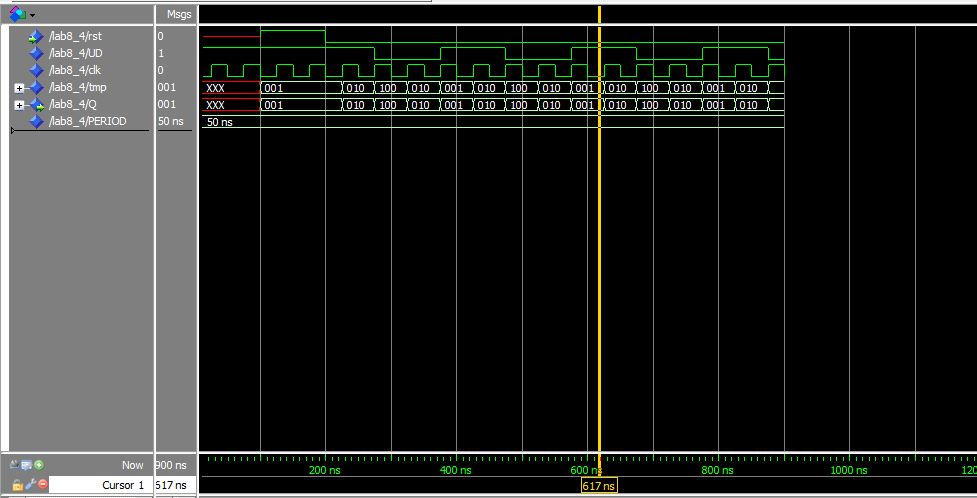
****

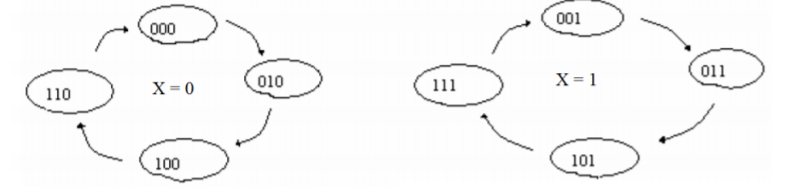
**4. Example code for 4-bit Johnson counter is below. Try to modify that VHDL model so, that your model works as 3-bit Johnson Up/Down counter**

**The code:**

|  |
| --- |
| LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  use IEEE.NUMERIC\_STD.all;  use IEEE.std\_logic\_unsigned.all;  ENTITY Lab8\_4 IS  PORT (rst: IN STD\_LOGIC;  Q: OUT STD\_LOGIC\_VECTOR(2 DOWNTO 0));  END Lab8\_4;  ARCHITECTURE JohnsonCounter OF Lab8\_4 IS  constant PERIOD: time := 50ns;  signal clk : std\_logic := '0';  signal tmp:STD\_LOGIC\_VECTOR (2 downto 0);  signal UD : std\_logic:= '1';  BEGIN  P0:PROCESS(rst,UD,clk)  BEGIN  clk <= not clk after PERIOD/2; -- Generating clock, the period = 50ns.  IF(rst = '1') then  tmp<="001";  ELSIF (clk'event and clk='1') THEN  IF( UD = '1' ) THEN  tmp(1) <= tmp(0);  tmp(2) <= tmp(1);  tmp(0) <= tmp(2);  IF(tmp(1) = '1') THEN  UD <= not UD;  END IF;    ELSE  tmp(1) <= tmp(2);  tmp(2) <= tmp(0);  tmp(0) <= tmp(1);  IF(tmp = "010") THEN  UD <= not UD;  END IF;  END IF;  END IF;  END PROCESS P0;  Q <= tmp;  END JohnsonCounter; |

**The simulating picture:**

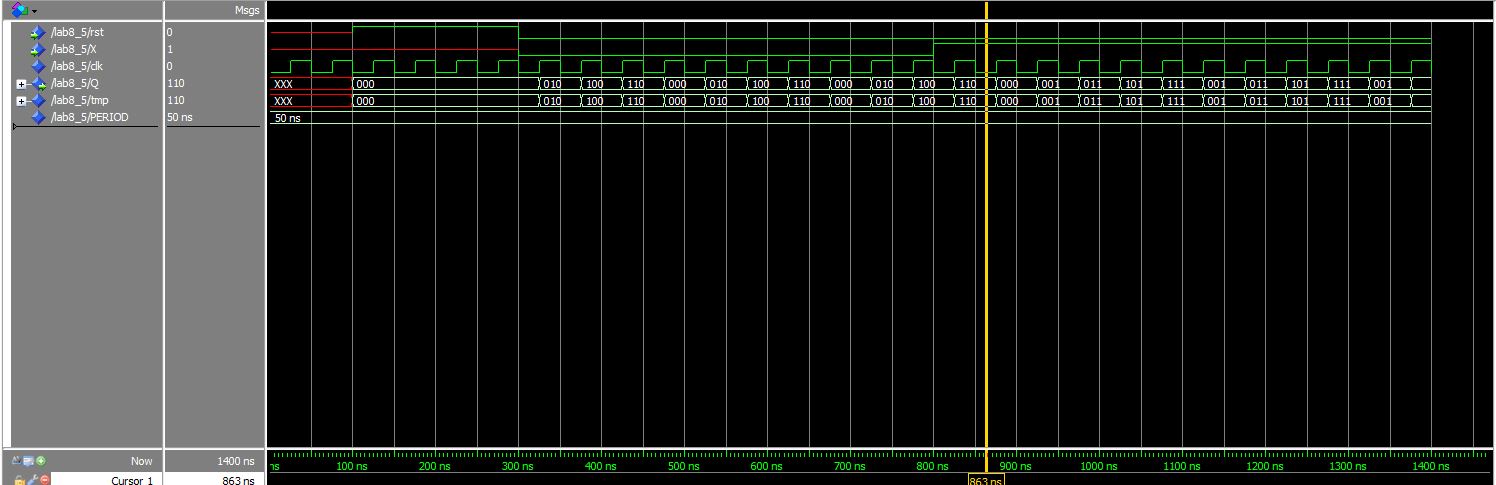
****

**5. Build a VHDL-model of a 3-bit counter which is counting only the even values when X = 0 and Odd values, when X = 1.**

**The code:**

|  |
| --- |
| LIBRARY ieee ;  USE ieee.std\_logic\_1164.all ;  use IEEE.NUMERIC\_STD.all;  use IEEE.std\_logic\_unsigned.all;  ENTITY Lab8\_5 IS  PORT (X,rst: IN STD\_LOGIC;  Q: OUT STD\_LOGIC\_VECTOR(2 DOWNTO 0));  END Lab8\_5;  ARCHITECTURE BitCounter OF Lab8\_5 IS  constant PERIOD: time := 50ns;  signal clk : std\_logic := '0';  signal tmp:STD\_LOGIC\_VECTOR (2 downto 0);  BEGIN  P0:PROCESS(X,clk,tmp,rst)  BEGIN  clk <= not clk after PERIOD/2; -- Generating clock, the period = 50ns.  IF(rst = '1') then -- After press rst, tmp = 000  tmp<="000";  ELSIF(X = '0') THEN -- When X = 0, tmp = 000 and count to 110    IF (clk'event and clk='1') THEN  tmp <= "000";  IF(tmp >= "000") THEN  tmp <= tmp + 2;  END IF;  END IF;    ELSIF( X = '1' ) THEN -- When X = 1 change tmp = 001 and count to 111    IF (clk'event and clk='1') THEN  tmp <= "001";  IF(tmp >= "001") THEN  tmp <= tmp + 2;  END IF;  END IF;  END IF;  END PROCESS P0;  Q <= tmp; -- The Output  END BitCounter; |

**The simulating picture:**

****