DDV3101-1 Datamaskinarkitektur og VHDL programering

Ingvar A. Wedervang - 220933

November 2020

Contents

1 Introduction

This report give an overview of the denouncing circuit design with implicit data path, which is the semester assignment in DDV3101-1 Datamaskinarkitektur og VHDL programering, autumn 2020 at USN(University of Southeastern-Norway). The report contains the design of the circuit, test-bench and a link to a youtube video that I made, which demonstrates the circuit on a Basys3 card: https://www.youtube.com/watch?v=8GmLg2w02yYfeature=youtu.be

2 System overview

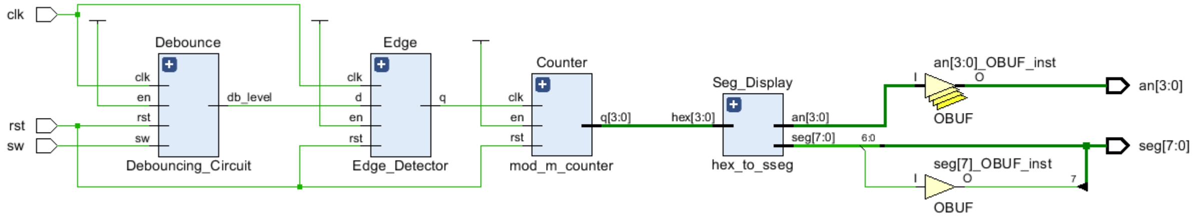


Figure 1: The entire project in elaborated form

3 Debouncing circuit

When a the button T18 on the Basys3 is pressed, the system shall increment a counter (mod 10) by 1, and show the current count on a SEGG display. Figure 1 shows the entire system. A button is connected to the sw signal, which is sent to the debounce part of the system. Further the signal is injected to an edge detector, that synchronizes the sw and clock signal before it reaches the counter. The counter is visualized by a segg display.

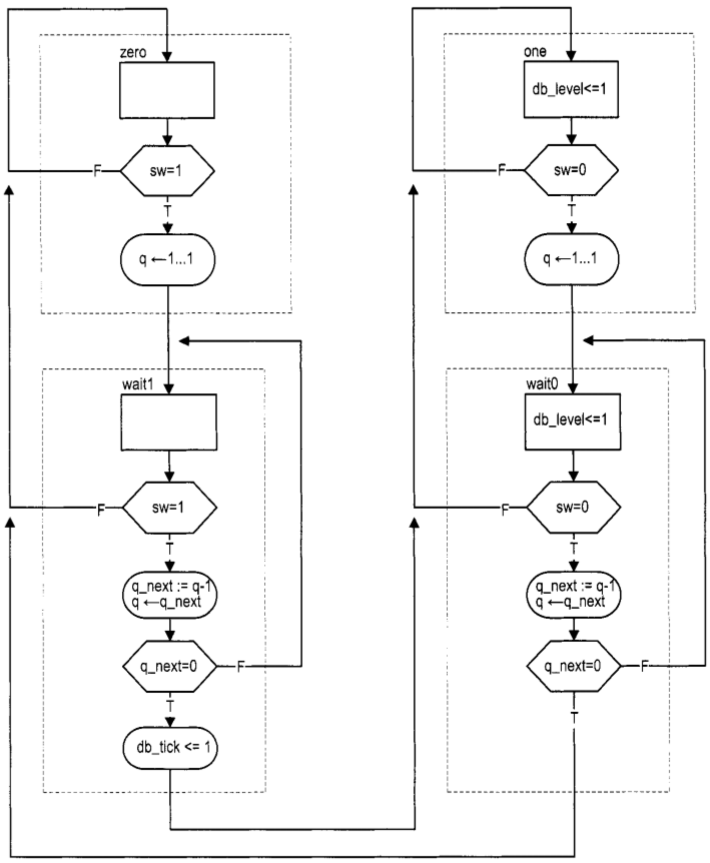
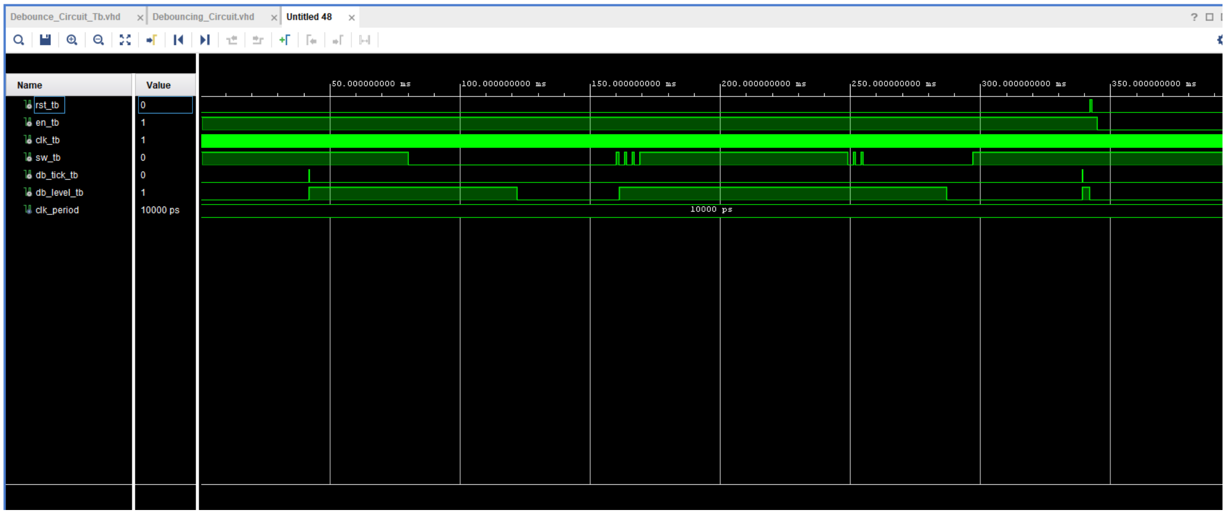


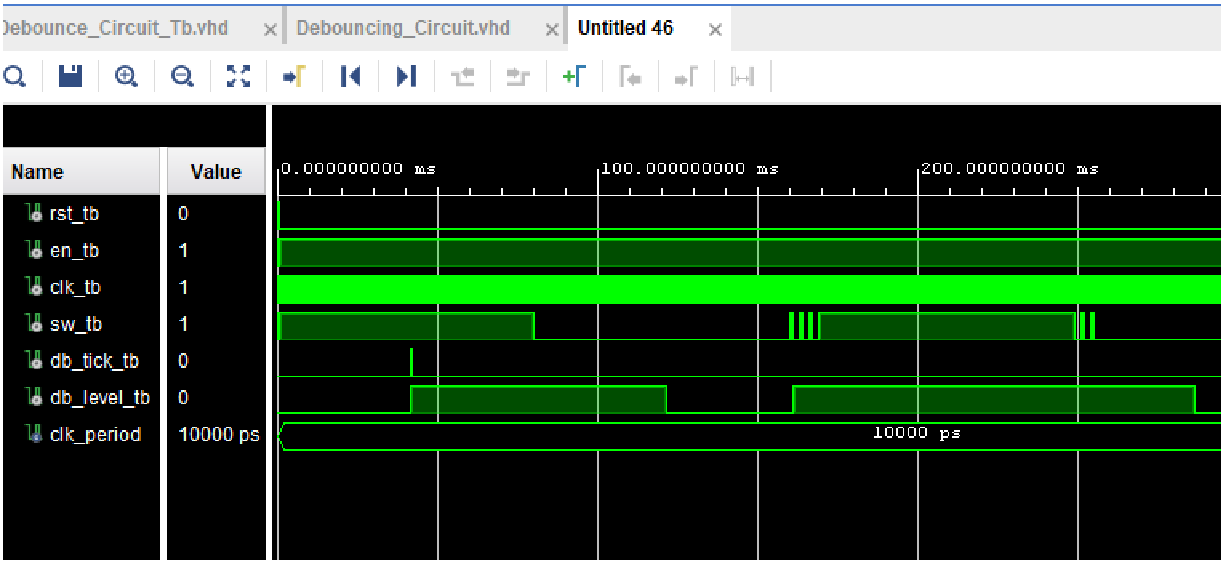
Figure 2: ASMD chart from[1]

The ASMD chart from[1] was used when programming the debouncing circuit. The code has implicit datapath.



Name	Value
rst_tb	0
en_tb	1
clk_tb	1
sw_tb	0
db_tick_tb	0
db_level_tb	1
clk_period	10000 ps

Figure 3: The entire test



Name	Value
rst_tb	0
en_tb	1
clk_tb	1
sw_tb	1
db_tick_tb	0
db_level_tb	0
clk_period	10000 ps

Figure 4: Debounce part of the test

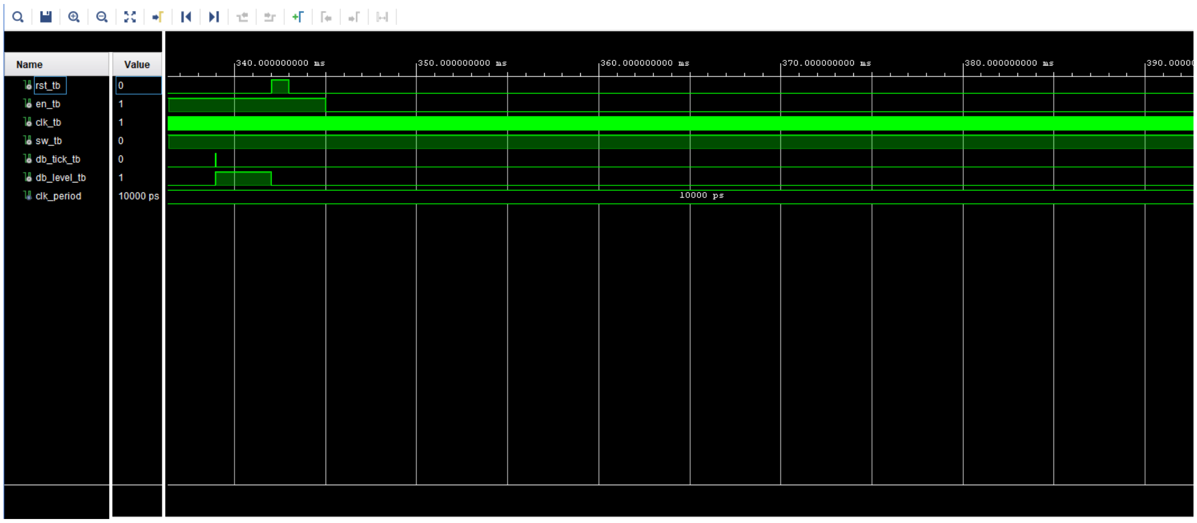


Figure 5: Testing En and RST signals

the test bench in figure 3 shows the entire testing of the circuit. The test is confirming that the debounce circuit works, including the rst and en signals. Figure 4 and 5 is from the test.

4 Edge detector

The edge detector is making sure that signals reaching the counter is synchronized with the clk signal. The edge detector is a simple register, synchornized with a clock.

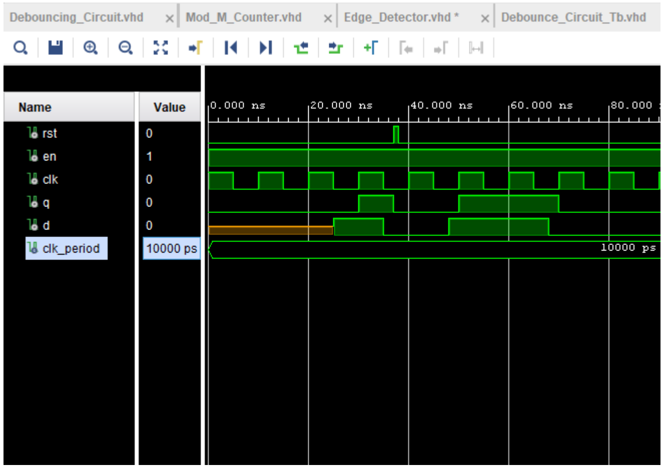


Figure 6: test bench for edge detector

5 XDC & testing on Basys3

The all the different modules in the system was collected in a queue entity, shown in figure 1. The system was sent to the Basys3 card. The following link is a demonstration video I published on youtube: https://www.youtube.com/watch?v=8GmLg2w02yYfeature=you

6 Sources

1. P. P. Chu, ”FPGA prototyping by VHDL examples”

7 Appendix

7.1 Debouncing circuit

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

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

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Debouncing_Circuit is
    Port ( rst                                  : in STD_LOGIC;
           en                                   : in STD_LOGIC;
           clk                                  : in STD_LOGIC;
           sw                                   : in STD_LOGIC;
           db_tick, db_level    : out STD_LOGIC);
                   -- db_tick, high on transition from 0 to 1
end Debouncing_Circuit;

architecture Behavioral of Debouncing_Circuit is


---------------signal declaration-----


type debounce_states is                 (zero, wait1, one, wait0);
constant N                      : integer := 21;
                                -- 40 ms deleay. 2^20 * 2 ns = 40 ms;
signal r_reg, r_next            : unsigned (N downto 0);
signal state_reg, state_next    : debounce_states;
--signal db_tick_s, db_level_s    : std_logic;


---------------sequential part--------

begin

process(en, rst, clk, state_reg, state_next, r_reg, r_next)
begin
```

```
        if(rst = '1') then
                state_reg <= ZERO;
                r_reg <= (others => '0');
        elsif(rising_edge(clk)) then
                if(en = '1') then
                        state_reg <= state_next;
                        r_reg <= r_next;
                end if;
        end if;
end process;


---------------next state logic-------


process(clk, state_reg, state_next, r_reg, r_next, sw)
begin
        state_next      <= state_reg;
        r_next          <= r_reg;
        db_tick         <= '0';
        db_level        <= '0';


case state_reg is
        when zero =>
                if(sw = '1') then
                r_next <=  (others => '1'); -- reg is filled with all 1
                state_next <= wait1;
                end if;

        when wait1 =>
                if(sw = '1') then
                        r_next <= r_reg - 1;
                        if(r_next = 0) then
                                state_next <= one;
                                db_tick <= '1';
                                -- Tick on transition from 1 to 0
                        end if;
                else
                        state_next <= one;
                end if;

        when one =>
                db_level <= '1';
                if (sw = '0') then
                        r_next <=  (others => '1');
                        state_next <= wait0;
                end if;

        when wait0 =>
                db_level <= '1';
                if(sw = '0') then
                        r_next <= r_reg -1;
                        if(r_next = 0) then
                                state_next <= zero;
                        end if;
                end if;

        end case;

end process;

end behavioral;
```

7.2 Debouncing circuit test bench

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

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

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Debounce_Circuit_Tb is
--  Port ( );
end Debounce_Circuit_Tb;

architecture Behavioral of Debounce_Circuit_Tb is


component Debouncing_Circuit is
    Port ( rst                                  : in STD_LOGIC;
           en                                   : in STD_LOGIC;
           clk                                  : in STD_LOGIC;
           sw                                    : in STD_LOGIC;
           db_tick, db_level   : out STD_LOGIC);
                                                                -- db_tick, 
end component Debouncing_Circuit;

constant clk_period                 :  time := 10 ns;
signal rst_tb                                   :  STD_LOGIC;
signal en_tb                                        :
STD_LOGIC;
signal clk_tb                                   :  STD_LOGIC;
signal sw_tb                                    :  STD_LOGIC;


signal db_tick_tb                   :  STD_LOGIC;
signal db_level_tb                  :  STD_LOGIC;


begin

UUT: Debouncing_Circuit
port map(rst => rst_tb,en => en_tb,clk => clk_tb,sw => sw_tb,db_tick => db_t



process

begin
loop
        clk_tb  <= '1';
        wait for clk_period/2;
        clk_tb  <= '0';
        wait for clk_period/2;
        end loop;
end process;

process

begin
wait for 20 ns;
rst_tb <= '1';
en_tb <= '0';
sw_tb <= '0';
wait for 20 ns;
en_tb <= '1';
rst_tb <= '0';
sw_tb <= '0';
```

```
wait for 20 ns; ----INITILIZATION FINISHED

sw_tb <= ’1’; ------ ON AND OFF
wait for 80 ms;
sw_tb <= ’0’;
wait for 80 ms;

sw_tb <= ’1’; ------DOUBNCE TO ON
wait for 1 ms;
sw_tb <= ’0’;
wait for 2 ms;
sw_tb <= ’1’;
wait for 1 ms;
sw_tb <= ’0’;
wait for 2 ms;
sw_tb <= ’1’;
wait for 1 ms;
sw_tb <= ’0’;
wait for 2 ms;

sw_tb <= ’1’; -----DEBOUNCE to OFF
wait for 80 ms;
sw_tb <= ’0’;
wait for 2 ms;
sw_tb <= ’1’;
wait for 1 ms;
sw_tb <= ’0’;
wait for 2 ms;
sw_tb <= ’1’;
wait for 1 ms;
sw_tb <= ’0’;
wait for 2 ms;

sw_tb <= ’0’;
wait for 40 ms;

sw_tb <= ’1’;
wait for 45 ms;
rst_tb <= ’1’;

wait for 1 ms;
rst_tb <= ’0’;
wait for 2 ms;
en_tb <= ’0’;
wait for 50 ms;
```

```
end process;


end Behavioral;
```

7.3 Edge detector

```
----------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05.11.2020 12:55:02
-- Design Name:
-- Module Name: Edge_Detector - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

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

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Edge_Detector is
    Port ( clk  : in STD_LOGIC;
           en    : in STD_LOGIC;
           rst   : in STD_LOGIC;
           d     : in STD_LOGIC;
           q     : out STD_LOGIC);
end Edge_Detector;

architecture Behavioral of Edge_Detector is

type states is (ZERO, EDGE, ONE);
signal state_reg, state_next : states;

begin
process(rst, clk, en, state_reg, state_next)
begin

if (rst = '1') then
        state_reg <= ZERO;
elsif(rising_edge(clk)) then
        if (en = '1') then
                state_reg <= state_next;
        end if;
end if;

end process;


process(state_reg, state_reg, d)
begin
state_next <= state_reg;
--q <= (others =>'0');
q <= '0';

case state_reg is

when ZERO =>
        if(d = '1') then
                state_next <= EDGE;

        end if;

when EDGE =>
        q <= '1';
        if(d = '0') then
                state_next <= ZERO;
        else
                state_next <= ONE;
        end if;


when ONE =>
        q <= '1';
        if(d = '0') then
                state_next <= ZERO;
        end if;

        end case;

end process;

end Behavioral;
```

7.4 Edge detector test bench

```
----------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 20.11.2020 12:45:43
-- Design Name:
-- Module Name: Edge_Detector_Tb - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

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

```
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Edge_Detector_Tb is
--  Port ( );
end Edge_Detector_Tb;

architecture Behavioral of Edge_Detector_Tb is

component Edge_Detector is
    Port ( clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           en : in STD_LOGIC;
            d : in STD_LOGIC;
           q : out STD_LOGIC);
end component Edge_Detector;


constant clk_period                  :  time := 10 ns;
signal rst                                       :  STD_LOGIC;
signal en                                            :
STD_LOGIC;
signal clk                                       :  STD_LOGIC;
signal q, d : STD_logic;


begin

UUT: Edge_Detector
port map(en => en, rst => rst, clk => clk, q => q, d => d);

process

begin
loop  -------10 ns CLK pulse
        clk  <= '1';
        wait for clk_period/2;
        clk  <= '0';
        wait for clk_period/2;
        end loop;
end process;

process
```

```
begin

en <= ’1’;
rst <= ’0’;
wait for 20 ns; -- test begin at 20 ns

wait for 5 ns; -- ZERO -> TICK -> ZERO
d <= ’1’;
wait for 5 ns;
d <= ’0’;

d <= ’1’; ----RST
wait for 5 ns;
d <= ’0’;
wait for 2 ns;
rst <= ’1’;
wait for 1 ns;
rst <= ’0’;

wait for 10 ns;

d <= ’1’; --------- ZERO -> TICK -> ONE -> ZERO
wait for 20 ns;
d <= ’0’;

end process;



end Behavioral;
```

7.5 mod M counter

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

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

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
```

```
--use UNISIM.VComponents.all;

entity mod_m_counter is
    generic(
    N : integer := 16;
    M : integer := 4);

    Port ( rst      : in STD_LOGIC;
           en       : in STD_LOGIC;
           clk      : in STD_LOGIC;
           q        : out STD_LOGIC_VECTOR (M-1 downto 0);
           max_tick : out std_logic);
end mod_m_counter;

architecture Behavioral of mod_m_counter is
signal current_state, next_state : unsigned(M -1 downto 0);

begin
        process(rst, clk)
        begin
                if(rst = '1') then
                        current_state <= (others => '0');
                elsif (rising_edge(clk)) then --(clk'event and clk = '1')
                        if(en = '1' ) then
                                current_state <= next_state;
                        end if;
                end if;
        end process;

--next state logic

------input : r_reg
------output : r_next
  next_state <= (others => '0') when current_state = N-1 else
                          current_state +1;

 -------r_reg and external input signal
         q <= std_logic_vector(current_state);
         max_tick <= '1' when current_state = N-1 else
                                '0';

end Behavioral;
```

7.6 queue system

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

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

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity mod_m_counter is
    generic(
    N : integer := 16;
    M : integer := 4);

    Port ( rst      : in STD_LOGIC;
           en       : in STD_LOGIC;
           clk      : in STD_LOGIC;
           q        : out STD_LOGIC_VECTOR (M-1 downto 0);
           max_tick : out std_logic);
end mod_m_counter;

architecture Behavioral of mod_m_counter is
signal current_state, next_state : unsigned(M -1 downto 0);

begin
        process(rst, clk)
        begin
                if(rst = ’1’) then
                        current_state <= (others => ’0’);
                elsif (rising_edge(clk)) then --(clk’event and clk = ’1’)
                        if(en = ’1’ ) then
                                current_state <= next_state;
                        end if;
                end if;
        end process;

--next state logic

------input : r_reg
------output : r_next
  next_state <= (others => ’0’) when current_state = N-1 else
```

```
                          current_state +1;

 -------r_reg and external input signal
         q <= std_logic_vector(current_state);
         max_tick <= '1' when current_state = N-1 else
                                '0';

end Behavioral;
```

7.7 Hex to seg

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

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

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;****

entity hex_to_sseg is
    Port ( hex  : in STD_LOGIC_VECTOR (3 downto 0);
           seg  : out STD_LOGIC_VECTOR (7 downto 0);
           an   : out STD_LOGIC_VECTOR (3 downto 0));
end hex_to_sseg;

architecture Behavioral of hex_to_sseg is

begin


        an               <= "0000";       --- use all for SEG
        seg(7)  <= '1';          --- dp is turned off
        process(hex)
        begin
                case hex is
                        when "0000" => --0
                                seg(6 downto 0) <= "1000000";
                        when "0001" => --1
                                seg(6 downto 0) <= "1111001";
                        when "0010" => --2
```

```
                                seg(6 downto 0) <= "0100100";
                        when "0011" => --3
                                seg(6 downto 0) <= "0110000";
                        when "0100" => --4
                                seg(6 downto 0) <= "0011001";
                        when "0101" => --5
                                seg(6 downto 0) <= "0010010";
                        when "0110" => --6
                                seg(6 downto 0) <= "0000010";
                        when "0111" => --7
                                seg(6 downto 0) <= "1111000";
                        when "1000" => --8
                                seg(6 downto 0) <= "0000000";
                        when "1001" => --9
                                seg(6 downto 0) <= "0010000";
                        when "1010" => --10 | A
                                seg(6 downto 0) <= "0001000";
                        when "1011" => --11 | B
                                seg(6 downto 0) <= "0000011";
                        when "1100" => --12 | C
                                seg(6 downto 0) <= "1000110";
                        when "1101" => --13 | D
                                seg(6 downto 0) <= "0100001";
                        when "1110" => --14 | E
                                seg(6 downto 0) <= "0000110";
                        when "1111" => --15 | F
                                seg(6 downto 0) <= "0001110";

                end case;
        end process;


end Behavioral;
```

7.8 XDC

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the

## Clock signal
set_property PACKAGE_PIN W5 [get_ports clk]
        set_property IOSTANDARD LVCMOS33 [get_ports clk]
        create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [g
```

```
## Switches
#set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
#set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
#set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
#set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
#set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
#set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
#set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
#set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
#set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
#set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
#set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
#set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
#set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
#set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
#set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
#set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]


## LEDs
#set_property PACKAGE_PIN U16 [get_ports {led[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
#set_property PACKAGE_PIN E19 [get_ports {led[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
#set_property PACKAGE_PIN U19 [get_ports {led[2]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
#set_property PACKAGE_PIN V19 [get_ports {led[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
#set_property PACKAGE_PIN W18 [get_ports {led[4]}]
```

```
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
#set_property PACKAGE_PIN U15 [get_ports {led[5]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
#set_property PACKAGE_PIN U14 [get_ports {led[6]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
#set_property PACKAGE_PIN V14 [get_ports {led[7]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
#set_property PACKAGE_PIN V13 [get_ports {led[8]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
#set_property PACKAGE_PIN V3 [get_ports {led[9]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
#set_property PACKAGE_PIN W3 [get_ports {led[10]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
#set_property PACKAGE_PIN U3 [get_ports {led[11]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
#set_property PACKAGE_PIN P3 [get_ports {led[12]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
#set_property PACKAGE_PIN N3 [get_ports {led[13]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
#set_property PACKAGE_PIN P1 [get_ports {led[14]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
#set_property PACKAGE_PIN L1 [get_ports {led[15]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]


##7 segment display
set_property PACKAGE_PIN W7 [get_ports {seg[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {seg[0]}]
set_property PACKAGE_PIN W6 [get_ports {seg[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
set_property PACKAGE_PIN U8 [get_ports {seg[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
set_property PACKAGE_PIN V8 [get_ports {seg[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
set_property PACKAGE_PIN U5 [get_ports {seg[4]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
set_property PACKAGE_PIN V5 [get_ports {seg[5]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
set_property PACKAGE_PIN U7 [get_ports {seg[6]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]

set_property PACKAGE_PIN V7 [get_ports {seg[7]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {seg[7]}]

set_property PACKAGE_PIN U2 [get_ports {an[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
set_property PACKAGE_PIN U4 [get_ports {an[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
set_property PACKAGE_PIN V4 [get_ports {an[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
set_property PACKAGE_PIN W4 [get_ports {an[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]


##Buttons
        set_property IOSTANDARD LVCMOS33 [get_ports rst]
set_property PACKAGE_PIN T18 [get_ports sw]
        set_property IOSTANDARD LVCMOS33 [get_ports sw]
set_property PACKAGE_PIN W19 [get_ports btnL]
        #set_property IOSTANDARD LVCMOS33 [get_ports btnL]
#set_property PACKAGE_PIN T17 [get_ports btnR]
        #set_property IOSTANDARD LVCMOS33 [get_ports btnR]
#set_property PACKAGE_PIN U17 [get_ports btnD]
        #set_property IOSTANDARD LVCMOS33 [get_ports btnD]



##Pmod Header JA
##Sch name = JA1
#set_property PACKAGE_PIN J1 [get_ports {JA[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JA[0]}]
##Sch name = JA2
#set_property PACKAGE_PIN L2 [get_ports {JA[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JA[1]}]
##Sch name = JA3
#set_property PACKAGE_PIN J2 [get_ports {JA[2]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JA[2]}]
##Sch name = JA4
#set_property PACKAGE_PIN G2 [get_ports {JA[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JA[3]}]
##Sch name = JA7
#set_property PACKAGE_PIN H1 [get_ports {JA[4]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JA[4]}]
##Sch name = JA8
#set_property PACKAGE_PIN K2 [get_ports {JA[5]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JA[5]}]
##Sch name = JA9
#set_property PACKAGE_PIN H2 [get_ports {JA[6]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JA[6]}]
##Sch name = JA10
#set_property PACKAGE_PIN G3 [get_ports {JA[7]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JA[7]}]
```

```
##Pmod Header JB
##Sch name = JB1
#set_property PACKAGE_PIN A14 [get_ports {JB[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JB[0]}]
##Sch name = JB2
#set_property PACKAGE_PIN A16 [get_ports {JB[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JB[1]}]
##Sch name = JB3
#set_property PACKAGE_PIN B15 [get_ports {JB[2]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JB[2]}]
##Sch name = JB4
#set_property PACKAGE_PIN B16 [get_ports {JB[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JB[3]}]
##Sch name = JB7
#set_property PACKAGE_PIN A15 [get_ports {JB[4]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JB[4]}]
##Sch name = JB8
#set_property PACKAGE_PIN A17 [get_ports {JB[5]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JB[5]}]
##Sch name = JB9
#set_property PACKAGE_PIN C15 [get_ports {JB[6]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JB[6]}]
##Sch name = JB10
#set_property PACKAGE_PIN C16 [get_ports {JB[7]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JB[7]}]



##Pmod Header JC
##Sch name = JC1
#set_property PACKAGE_PIN K17 [get_ports {JC[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JC[0]}]
##Sch name = JC2
#set_property PACKAGE_PIN M18 [get_ports {JC[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JC[1]}]
##Sch name = JC3
#set_property PACKAGE_PIN N17 [get_ports {JC[2]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JC[2]}]
##Sch name = JC4
#set_property PACKAGE_PIN P18 [get_ports {JC[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JC[3]}]
##Sch name = JC7
#set_property PACKAGE_PIN L17 [get_ports {JC[4]}]
```

```
        #set_property IOSTANDARD LVCMOS33 [get_ports {JC[4]}]
##Sch name = JC8
#set_property PACKAGE_PIN M19 [get_ports {JC[5]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JC[5]}]
##Sch name = JC9
#set_property PACKAGE_PIN P17 [get_ports {JC[6]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JC[6]}]
##Sch name = JC10
#set_property PACKAGE_PIN R18 [get_ports {JC[7]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JC[7]}]


##Pmod Header JXADC
##Sch name = XA1_P
#set_property PACKAGE_PIN J3 [get_ports {JXADC[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[0]}]
##Sch name = XA2_P
#set_property PACKAGE_PIN L3 [get_ports {JXADC[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[1]}]
##Sch name = XA3_P
#set_property PACKAGE_PIN M2 [get_ports {JXADC[2]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[2]}]
##Sch name = XA4_P
#set_property PACKAGE_PIN N2 [get_ports {JXADC[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[3]}]
##Sch name = XA1_N
#set_property PACKAGE_PIN K3 [get_ports {JXADC[4]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[4]}]
##Sch name = XA2_N
#set_property PACKAGE_PIN M3 [get_ports {JXADC[5]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[5]}]
##Sch name = XA3_N
#set_property PACKAGE_PIN M1 [get_ports {JXADC[6]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[6]}]
##Sch name = XA4_N
#set_property PACKAGE_PIN N1 [get_ports {JXADC[7]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[7]}]



##VGA Connector
#set_property PACKAGE_PIN G19 [get_ports {vgaRed[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[0]}]
#set_property PACKAGE_PIN H19 [get_ports {vgaRed[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[1]}]
#set_property PACKAGE_PIN J19 [get_ports {vgaRed[2]}]
```

```
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[2]}]
#set_property PACKAGE_PIN N19 [get_ports {vgaRed[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[3]}]
#set_property PACKAGE_PIN N18 [get_ports {vgaBlue[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[0]}]
#set_property PACKAGE_PIN L18 [get_ports {vgaBlue[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[1]}]
#set_property PACKAGE_PIN K18 [get_ports {vgaBlue[2]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[2]}]
#set_property PACKAGE_PIN J18 [get_ports {vgaBlue[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[3]}]
#set_property PACKAGE_PIN J17 [get_ports {vgaGreen[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[0]}]
#set_property PACKAGE_PIN H17 [get_ports {vgaGreen[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[1]}]
#set_property PACKAGE_PIN G17 [get_ports {vgaGreen[2]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[2]}]
#set_property PACKAGE_PIN D17 [get_ports {vgaGreen[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[3]}]
#set_property PACKAGE_PIN P19 [get_ports Hsync]
        #set_property IOSTANDARD LVCMOS33 [get_ports Hsync]
#set_property PACKAGE_PIN R19 [get_ports Vsync]
        #set_property IOSTANDARD LVCMOS33 [get_ports Vsync]


##USB-RS232 Interface
#set_property PACKAGE_PIN B18 [get_ports RsRx]
        #set_property IOSTANDARD LVCMOS33 [get_ports RsRx]
#set_property PACKAGE_PIN A18 [get_ports RsTx]
        #set_property IOSTANDARD LVCMOS33 [get_ports RsTx]


##USB HID (PS/2)
#set_property PACKAGE_PIN C17 [get_ports PS2Clk]
        #set_property IOSTANDARD LVCMOS33 [get_ports PS2Clk]
        #set_property PULLUP true [get_ports PS2Clk]
#set_property PACKAGE_PIN B17 [get_ports PS2Data]
        #set_property IOSTANDARD LVCMOS33 [get_ports PS2Data]
        #set_property PULLUP true [get_ports PS2Data]


##Quad SPI Flash
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it u
##STARTUPE2 primitive.
#set_property PACKAGE_PIN D18 [get_ports {QspiDB[0]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[0]}]
```

```
#set_property PACKAGE_PIN D19 [get_ports {QspiDB[1]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[1]}]
#set_property PACKAGE_PIN G18 [get_ports {QspiDB[2]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[2]}]
#set_property PACKAGE_PIN F18 [get_ports {QspiDB[3]}]
        #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[3]}]
#set_property PACKAGE_PIN K19 [get_ports QspiCSn]
        #set_property IOSTANDARD LVCMOS33 [get_ports QspiCSn]
```