University of Essex

CE869 High Level Logic Design Assignment 1

Name: Ye Yihao

Student Number: 1305597

Tutor: Luca Citi

Lecture: High Level Logic Design

Submission Date: 07 / 02 / 2014

1. Description of Program Design

The propose of the project is to design a timer with countdown and initial time set function depending on a XILINX FPGA. The project is divided into four part: one digit unit, four digits unit, countdown and cnt. Applying the high level design method in this project, the first part finished is one digit unit. This is the basic function and should not be changed later. The one digit unit use WHEN ... SELECT statement to achieve the requirement that display 0 to 9 on the seven segments display component by link the switch binary value with the cathode value which show particular number (each segment of display is corresponding to a certain bit of the value of cathode, e.g. number "one" is corresponding to "1111001"). The port of the one digit entity consist of cathode output and digit input (switch).

The next entity is four digits unit. This entity is based on one digit unit and a clock. The clock used here is divided MCLK provided by higher level entity (countdown). The statement RISING_EDGE() used here is to count the clock and plus the variable cnt, cnt is a two bite unsigned signal, when it change its value, the corresponding anode is selected, the statement applied to achieve this function is CASE ... WHEN statement. As the frequency of the divided MCLK is fast enough, four seven segment displays will seems to display different number at the same time. In order to display a decimal point after the second number, DP is assigned to value zero (open) when the third anode is selected.

The countdown entity is designed to complete the task of countdown and initial time setup function. This entity is based on four digits unit. A clock divied function called fr_1kHz process is executed in this entity to provide the frequency which is requested in four digits unit to shift the lightening of four seven segment displays at a appropriate speed, since a higher frequency will result in blip in display while a lower frequency will make the displays flicker. The frequency of clock used here is MCLK divided by 2048. After the clock divied function, several WHEN statements are applied to deliver the value which is setted by four switches to variable Ax (A0 - A3), this variable will be called in the cnt entity for the initial time set.

The cnt entity is a generic entity. According to there are the four seven segment displays, four cnt entities are executed in the countdown entity. There are four input port (i.e. CK, c_in, SW7, Ax) and two buffer port (i.e. q, c_out) in each cnt entity. The variable q decide the number appears on the display. As SW7 equals to zero (switch 7 turned down), the value of q will be set by the Ax. When SW7 equals to one (switch 7 turned up), c_out equals zero (no carry to higher level bit) and c_in equals one (borrow detected from lower level bit), q minus itself as a rising edge of CK (RCCK) is detected. When c_out equals one and SW7 equals to one, q will be reset by the value of modulo (generic parameter). The value of c_out depends on the value of c_in and q, when c_in equals one and q equals 0000, c_out equals one.

2. Revision During the Project

There are some mistakes in the process of the programming, particularly in the countdown entity, it is difficult to combine the initial time set function and the countdown function. When both of them effect the value of D0 - D3 (value of displays setted by switches), the complier will show the error that these variables are drive by multiple objects. The solution is create new variable A0 - A3 to restore the switches value and implement both two functions in cnt entity.

3. Translation between Two Type of Statement

In the VHDL, there are two type of statement, concurrent statements for the combinatorial circuits and sequential code for the sequential circuits. A representative case can use two different ways to implement the same functionality is anode selected function of four digits unit, when the functionality is implemented in concurrent statement the WITH ... SELECT ... WHEN statement should be applied, otherwise the CASE statement should be applied while sequential statement is required.

3.1 Concurrent Statements:

```
WITH cnt SELECT
    DIGIT <= D0 WHEN “00”,
             D1 WHEN “01”,
             D2 WHEN “10”,
             D3 WHEN OTHERS;
WITH cnt SELECT
    ANODES   <= "1110" WHEN “00”,
                "1110" WHEN “01”,
                "1110" WHEN “10”,
                "1110" WHEN OTHERS;
WITH cnt SELECT
    DP   <= "1" WHEN “00”,
            "1" WHEN “01”,
             "0" WHEN “10”,
             "1" WHEN OTHERS;
```

3.2 Sequential Statement:

```
PROCESS
```

```
BEGIN
    CASE cnt IS
    WHEN "00" =>
        DIGIT <= D0; ANODES <= "1110"; DP <= '1';
    WHEN "01" =>
        DIGIT <= D1; ANODES <= "1101"; DP <= '1';
    WHEN "10" =>
        DIGIT <= D2; ANODES <= "1011"; DP <= '0';
    WHEN OTHERS =>
        DIGIT <= D3; ANODES <= "0111"; DP <= '1';
    END CASE;
END PROCESS;
```

Reference:

[1] Luca, Citi, Lecture Notes of CE869: High Level Logic Design , University of ESSEX, February, 2014.

[2] Vijay, Polavarapu, "VHDL Programming" available: http://cis.poly.edu/cs2204/vhdlprg.pdf, [2013]

[3] http://en.wikipedia.org/wiki/VHDL#Synthesizable_constructs_and_VHDL_templates [Feb. 03, 2014].

APPENDIX:

1. One Digit:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity one_digit is
   Port ( DIGIT : in  UNSIGNED (3 downto 0);
          CATHODES : out  STD_LOGIC_VECTOR (6 downto 0));
end one_digit;

architecture Behavioral of one_digit is

begin
   WITH DIGIT SELECT
      CATHODES <= "1000000" WHEN "0000",
                  "1111001" WHEN "0001",
                  "0100100" WHEN "0010",
                  "0110000" WHEN "0011",
                  "0011001" WHEN "0100",
                  "0010010" WHEN "0101",
                  "0000010" WHEN "0110",
                  "1111000" WHEN "0111",
                  "0000000" WHEN "1000",
                  "0010000" WHEN "1001",
                    "0111111" WHEN OTHERS;

end Behavioral;
```

2. Four Digits:

```
-------------------------
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity four_digits is
   Port (D0, D1, D2, D3 : in  UNSIGNED (3 downto 0);
          CK : in  STD_LOGIC;
          DP : out  STD_LOGIC;
          CATHODES : out STD_LOGIC_VECTOR (6 downto 0);
          ANODES   : out STD_LOGIC_VECTOR (3 downto 0));
end four_digits;

architecture Behavioral of four_digits is
   signal DIGIT : UNSIGNED (3 downto 0);
   signal cnt : UNSIGNED (1 downto 0);
```

```
begin

    one_digit_unit : entity one_digit(Behavioral)
        port map (DIGIT => DIGIT,
            CATHODES => CATHODES);
-- one_digit_unit calling.

    PROCESS
    BEGIN

        WAIT UNTIL RISING_EDGE(CK);
        cnt <= cnt + 1;

    END PROCESS;
-- Button Pressed

    PROCESS
    BEGIN
        CASE cnt IS
        WHEN "00" =>
            DIGIT <= D0; ANODES <= "1110"; DP <= '1';
        WHEN "01" =>
            DIGIT <= D1; ANODES <= "1101"; DP <= '1';
        WHEN "10" =>
            DIGIT <= D2; ANODES <= "1011"; DP <= '0';
        WHEN OTHERS =>
            DIGIT <= D3; ANODES <= "0111"; DP <= '1';
        END CASE;
```

```
   END PROCESS;
-- Display

end Behavioral;
```

3. Countdown

```
-------------
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity countdown is
   Port ( CATHODES : out STD_LOGIC_VECTOR (6 downto 0);
             ANODES : out STD_LOGIC_VECTOR (3 downto 0);
             BTN : in  STD_LOGIC_VECTOR (3 downto 0);
             SW : in  UNSIGNED (3 downto 0);
             SW7 : in STD_LOGIC;
             DP : out STD_LOGIC;
             CK : in STD_LOGIC;
             MCLK : in  STD_LOGIC);
```

```
end countdown;




architecture Behavioral of countdown is
    signal A0, A1, A2, A3 : UNSIGNED (3 downto 0);
    signal D0, D1, D2, D3 : UNSIGNED (3 downto 0);
    SIGNAL c01, c02, c03, disp_ck : STD_LOGIC;
    SIGNAL fr_cnt : UNSIGNED (10 DOWNTO 0);
    --SIGNAL c00 : STD_LOGIC;
begin

    four_digits_unit : ENTITY four_digits(Behavioral)
        PORT MAP (d3 => d3, d2 => d2, d1 => d1, d0 => d0, DP
=> DP,
        ck => disp_ck, cathodes => cathodes, anodes => anodes);

    fr_1kHz : PROCESS
    BEGIN
    WAIT UNTIL RISING_EDGE(mclk);
    disp_ck <= fr_cnt(fr_cnt'HIGH);
    fr_cnt <= fr_cnt + 1;
    END PROCESS;
    --divide the mclk clock



    --c00 <= '0' WHEN SW7 = '1';
    --c00 <= '1' WHEN SW7 = '0';

        A0 <= SW WHEN BTN(0) = '1' AND SW7 = '0';
        A1 <= SW WHEN BTN(1) = '1' AND SW7 = '0';

        A2 <= SW WHEN BTN(2) = '1' AND SW7 = '0';

        A3 <= SW WHEN BTN(3) = '1' AND SW7 = '0';



    cnt_d0 : ENTITY work.cnt_gen GENERIC MAP (width => 4,
modulo => 10)

    PORT MAP (ck => ck, c_in => '1', q => d0, c_out => c01,
SW7 => SW7, Ax => A0);

    cnt_d1 : ENTITY work.cnt_gen GENERIC MAP (width => 4,
modulo => 6)

    PORT MAP (ck => ck, c_in => c01, q => d1, c_out => c02,
SW7 => SW7, Ax => A1);

    cnt_d2 : ENTITY work.cnt_gen GENERIC MAP (width => 4,
modulo => 10)

    PORT MAP (ck => ck, c_in => c02, q => d2, c_out => c03,
SW7 => SW7, Ax => A2);

    cnt_d3 : ENTITY work.cnt_gen GENERIC MAP (width => 4,
modulo => 10)

    PORT MAP (ck => ck, c_in => c03, q => d3, c_out => open,
SW7 => SW7, Ax => A3);


    --d2 <= "0000";

    --d3 <= "0000";


end Behavioral;
```

4. Cnt:

```
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 primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;


ENTITY cnt_gen IS

   GENERIC (width : NATURAL := 4;

             modulo : NATURAL := 10);

      PORT (ck, c_in : IN STD_LOGIC;

             q : BUFFER UNSIGNED ((width-1) downto 0);

             SW7: in STD_LOGIC;

             Ax: IN UNSIGNED (3 downto 0);

             c_out : BUFFER STD_LOGIC);

END cnt_gen;



ARCHITECTURE behav OF cnt_gen IS

BEGIN

   PROCESS

   BEGIN


   IF SW7 = '0' THEN

      q <= Ax;

   END IF;


   WAIT UNTIL RISING_EDGE(ck);

   IF c_out = '1' AND SW7 = '1' THEN
```

```
        q <= TO_UNSIGNED((modulo-1),4);
    ELSIF c_in = '1' AND SW7 = '1' THEN
        q <= q - 1;
    END IF;

    END PROCESS;
    c_out <= '1' WHEN (c_in = '1') AND (q = "0000") ELSE '0';

END behav;
```