LABORATORY # 3
LAB MANUAL

Combinatorial Logic

Objectives

Its purposes are to get familiar with:

1. Xilinx ISE Design, Synthesis and Basys Board FPGA Programming.

2. Learning Basys Board components and FPGA pin routing.

3. Understanding of Configuration files.

4. Synthesis and Implementation of combinational logic applications on FPGA.

5. Basys Board Programming

Equipment

- PC or compatible
- Digilent's Basys Spartan-3E FPGA Evaluation Board

Software

- Xilinx ISE Design Software Suite 10.1
- Digilent's Adept ExPort Software

Parts

N/A

Introduction

In all the labs we will adhere to the following industry standard design flow in applications development which utilizes FPGA devices.

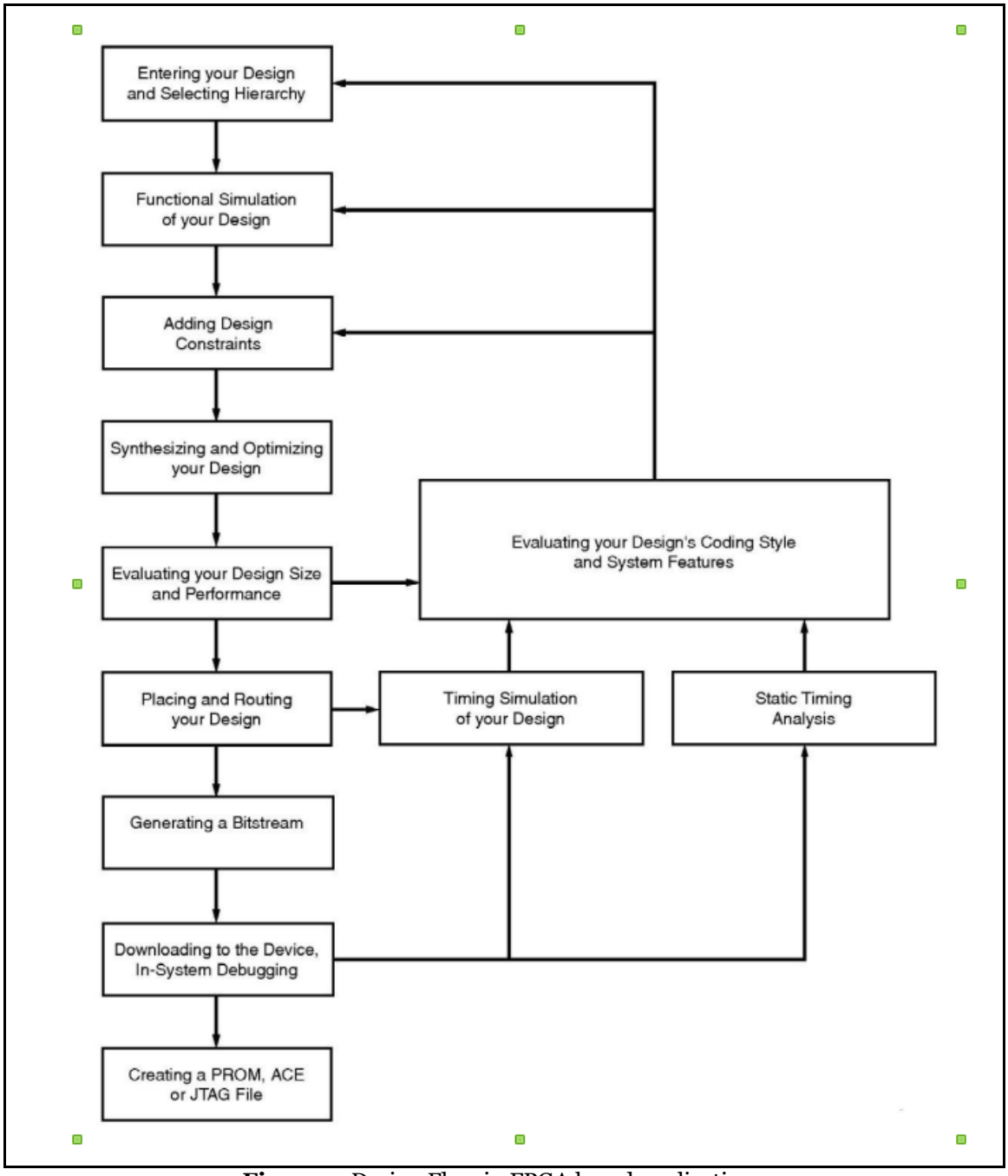

Figure 1. Design Flow in FPGA based applications

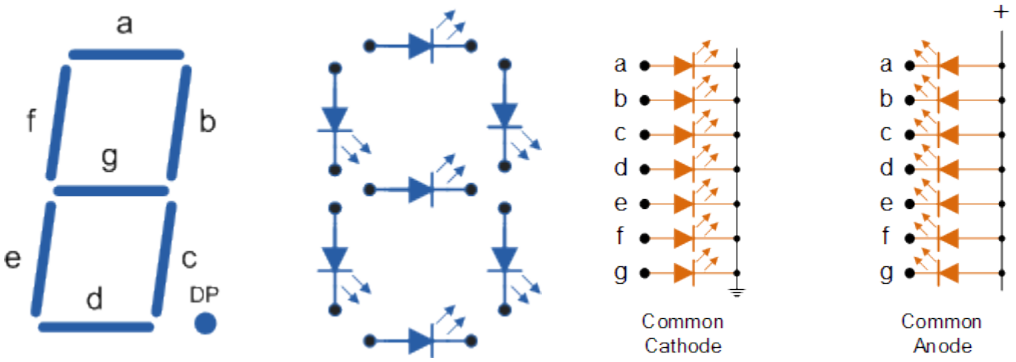

Figure2: (Left) Display image of a 7-segment LED design. (Right) Image of the common cathode and common anode configurations.

Please read and learn the 7-segment LED display from this link:
https://www.electronics-tutorials.ws/combination/comb_6.html

Briefly speaking, the 7- segments display consists of 7 independent LEDs (If you count the DP in, that will be 8 independent LEDs). You can control the combination in display numbers through 0-9.
In our lab, the 7-segment LEDs are common Anode, which means a=0 is light on, a=1 is light off. If you want to display number "1", the combination should be abcdefg = 4'b1001111, only segment b and segment c are light on.

BCD to 7 Segment LED Display

Specification

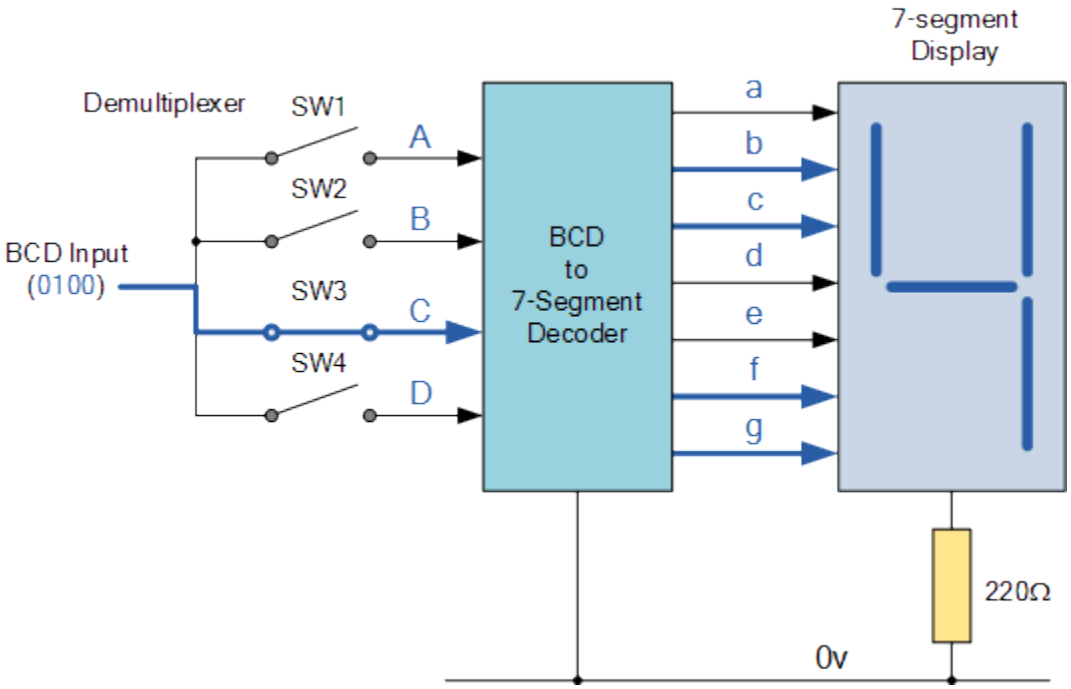

Figure 3: An example of the 4-bit BCD input (0100) representing the number "4".

You are required to implement the structural (slow to code) **or** behavioral (faster to code) description of the BCD-to-7seg decoder described above. The module you are about to implement should have the set of input and output ports given in listing 3. In here, we have provided a basic template that can be used as a starting point for the behavioral description, and please complete the missing code.

Note, the 7-seg display is active low (0 = on, 1 = off)

```
module bcd_to_7led_bh (
 input wire sw0 ,  // Switches
 input wire sw1 ,
 input wire sw2 ,
 input wire sw3 ,
 output reg a ,  // LED segments
 output reg b ,
 output reg c ,
 output reg d ,
 output reg e ,
 output reg f ,
 output reg g ,
 output reg an0,  // LED display control (To power on the entire 7-seg display)
 output reg an1,
```

```
 output reg an2,
 output reg an3
  );

// Internal wire
wire [3:0] bundle ;
assign bundle = {sw3,sw2,sw1,sw0 } ;

always @(*) begin

        // Setting the ANs signals (Set AN3 to 0 to turn it on)
        an0 = 1'b1;
        an1 = 1'b1;
        an2 = 1'b1;
        an3 = 1'b0; // Display in the module AN3

        // Setting the segments signals (Initialize all to off/1)
        a = 1'b1 ;
        b = 1'b1 ;
        c = 1'b1 ;
        d = 1'b1 ;
        e = 1'b1 ;
        f = 1'b1 ;
        g = 1'b1 ;

 case ( bundle )

        4'b0000 : begin // 0
                a = 1'b0 ;
                b = 1'b0 ;
                c = 1'b0 ;
                d = 1'b0 ;
                e = 1'b0 ;
                f = 1'b0 ;
                g = 1'b1 ;    //(Don't need to explicitly state that g is off here since
                              //  it is initialized to off already, but it doesn't hurt)
        end

        // Your code goes here for the other 8 numbers (1-9)


        endcase
end
```

```
endmodule
```

Listing 3. BCD-to-7seg decoder behavioral model

AN signals are required to be low for the 7-segments units to work (0 = on, 1 = off). In addition, to facilitate the implementation of the BCD to 7-Seg module, the following set of constraints are given.

```
# Inputs
NET "sw0" LOC = "p38";
NET "sw1" LOC = "p36";
NET "sw2" LOC = "p29";
NET "sw3" LOC = "p24";

# Outputs
NET "a" LOC = "p25";
NET "b" LOC = "p16";
NET "c" LOC = "p23";
NET "d" LOC = "p21";
NET "e" LOC = "p20";
NET "f" LOC = "p17";
NET "g" LOC = "p83";

// ANx
NET "an0" LOC = "p26";
NET "an1" LOC = "p32";
NET "an2" LOC = "p33";
NET "an3" LOC = "p34";
```

The testbench is also given if you do not use the board.

```
`timescale 1ns / 1ps

module bcdtoled_tb;

	// Inputs
	reg sw0;
	reg sw1;
	reg sw2;
	reg sw3;

	// Outputs
	wire a;
	wire b;
```

```
        wire c;
        wire d;
        wire e;
        wire f;
        wire g;

        // Instantiate the Unit Under Test (UUT)
        bcdto7led_bh uut (
                .sw0(sw0),
                .sw1(sw1),
                .sw2(sw2),
                .sw3(sw3),
                .a(a),
                .b(b),
                .c(c),
                .d(d),
                .e(e),
                .f(f),
                .g(g)
        );

        initial begin

                // Initialize Inputs
                sw3 = 0;        sw2 = 0;        sw1 = 0;        sw0 = 0;
                #100;
                $display("TC10 ");
                if ( {a,b,c,d,e,f,g} != 7'b0000001 ) $display  ("Result is wrong %b
", {a,b,c,d,e,f,g});

                // Your test cases go here (9 left)

end
endmodule
```

Demonstration

Provide the truth tables, circuit schematic and functionality of the design.

Procedures

1. Xilinx ISE Design and Synthesis environment;

2. Creation of Configuration files;
3. Usage of Adept ExPort download software;

Presentation and Report

Must be presented according to the general EE120A lab guidelines.

Prelab

1. Familiarize yourself with ISE tutorials.
2. Review Lectures.
3. Try to answer all the questions, prepare logic truth tables, do all necessary computations.