Engineering 378 Digital Systems Design

Exp. 4: Downloading to FPGA

Submitted by:

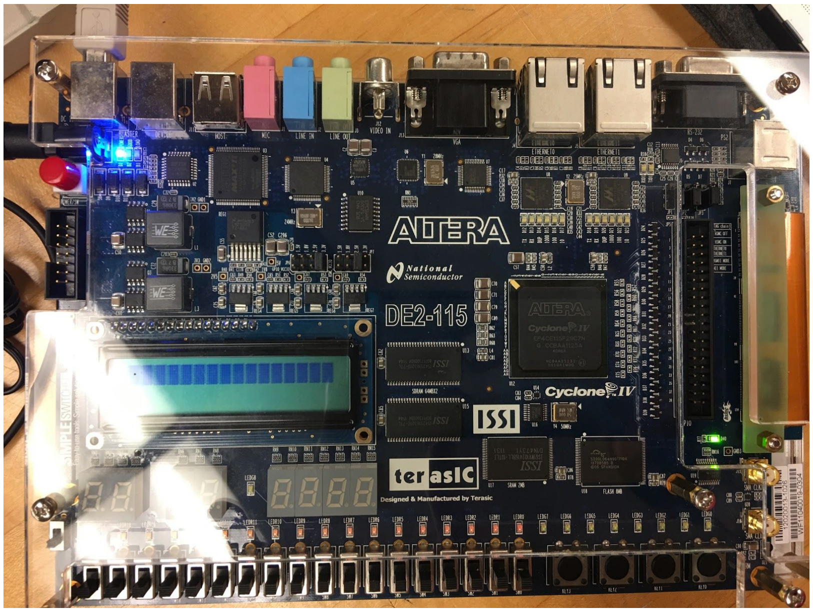
Norald Alejo
Farnam Adelkhani

October 13, 2016

Demonstration ……..…………………………………………………………………_/5
Report: Presentation ……..…………………………………………………_/2
Problem: analysis ……..…………………………………………………_/2
Design: work ………………………………………………………………_/5
Results ……..………………………………………………………………………_/5
Conclusion/Discussion ……..…………………………………………………_/1
Total ………………..………………………………………………………………_/20

Problem Analysis

This lab is an introduction to downloading programs to an FPGA device; in this particular situation the DE2-115 Board will be implemented. The user will model verilog code for a system that will convert 4-bit code into the following systems; hex, BCD and decimal number. The code should be entered utilizing 4 of the onboard switches available. The method of conversion will be selected using one of three buttons on the board, preferably BTN0 to BTN2. Code using one of the three on board switches, preferably SW0 to SW3. The challenge is to set up all the conditions for each binary output of the symbols indicated in the laboratory assignment. The same outputs then need to be programmed and registered to the indicated pin, this configuration is available through an excel on the ilearn website.



DE2-115 Board

Hardware Design

The pre-lab assignment was done, by choice, implemented the option to write up a pseudo-code for the eventual program. Considering the use of a Multi-code Converter Module, the task is to download the verilog module to convert the four bit code into one of these modes:

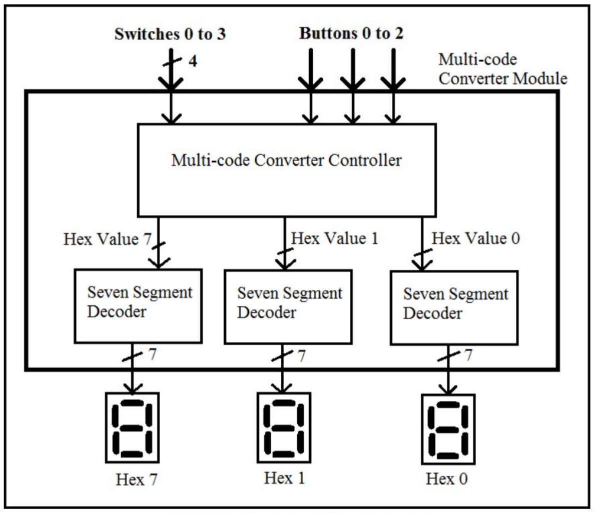


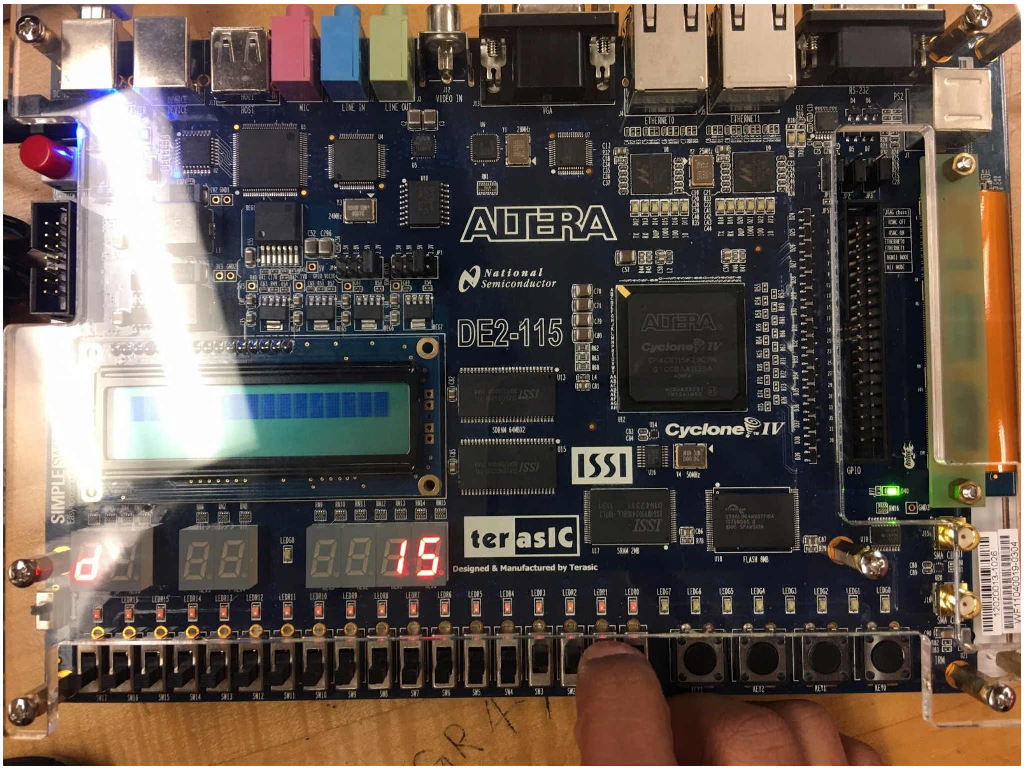
Figure 1: Multi-code Converter Module

i. A 1-digit Hex number: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

ii. A 1-digit decimal number (BCD): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (Display 9 for value 10+)

iii. A 2 digits decimal number: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

The verilog module must allow the 4 bit code to be entered by means of the four switches that are available on the board. The mode of conversion must be indicated by means of one of the three available push buttons of the board. The output of the conversion system should be displayed at the two rightmost 7-segment displays. The leftmost 7-segment display is required to display to the user the type of conversion, with letters corresponding to the mode. An FPGA is borrowed to implement the verilog model onto.


ALTERA
National Semiconductor
DE2-115
ISSI
Cyclone IV
terasIC
Designed & Manufactured by Terasic
d
15

Verilog Modeling:

```
module multCode(sw, bttn, hex7, hex1, hex0);



input [3:0] sw;

input [2:0] bttn;

output [6:0] hex7, hex1, hex0;

reg [6:0] hex7, hex1, hex0;

parameter       zero  = 7'b1000000;
parameter       one   = 7'b1111001;
parameter       two   = 7'b0100100;
parameter       three = 7'b0110000;
parameter       four  = 7'b0011001;
parameter       five  = 7'b0010010;
parameter        six  = 7'b0000010;
parameter       seven = 7'b1111000;
parameter       eight = 7'b0000000;
parameter        nine = 7'b0010000;
parameter letterA = 7'b0001000;
parameter letterb = 7'b0000011;
parameter letterC = 7'b1000110;
parameter letterd = 7'b0100001;
parameter letterE = 7'b0000110;
parameter letterF = 7'b0001110;
parameter letterH = 7'b0001001;

always @(bttn or sw)
begin
        hex1 = 7'b1111111;
        if (bttn == 3'b110)
        begin
                hex7 = letterH;
                case (sw)
                        4'b0000:
                                hex0 = zero;
                        4'b0001:
                                hex0 = one;
                        4'b0010:
                                hex0 = two;
                        4'b0011:
                                hex0 = three;
                        4'b0100:
                                hex0 = four;
                        4'b0101:
                                hex0 = five;
                4'b0110:
                        hex0 = six;
                4'b0111:
                        hex0 = seven;
                4'b1000:
                        hex0 = eight;
                4'b1001:
                        hex0 = nine;
                4'b1010:
                        hex0 = letterA;
                4'b1011:
                        hex0 = letterb;
                4'b1100:
                        hex0 = letterC;
                4'b1101:
                        hex0 = letterd;
                4'b1110:
                        hex0 = letterE;
                4'b1111:
                        hex0 = letterF;
        endcase
end
else if (bttn == 3'b101)
begin
        hex7 = letterb;
        case (sw)
                4'b0000:
                        hex0 = zero;
                4'b0001:
                        hex0 = one;
                4'b0010:
                        hex0 = two;
                4'b0011:
                        hex0 = three;
                4'b0100:
                        hex0 = four;
                4'b0101:
                        hex0 = five;
                4'b0110:
                        hex0 = six;
                4'b0111:
                        hex0 = seven;
                4'b1000:
                        hex0 = eight;
                default:
                        hex0 = nine;
        endcase
end
else
begin
        hex7 = letterd;
        case (sw)
                4'b0000:
                        hex0 = zero;
```

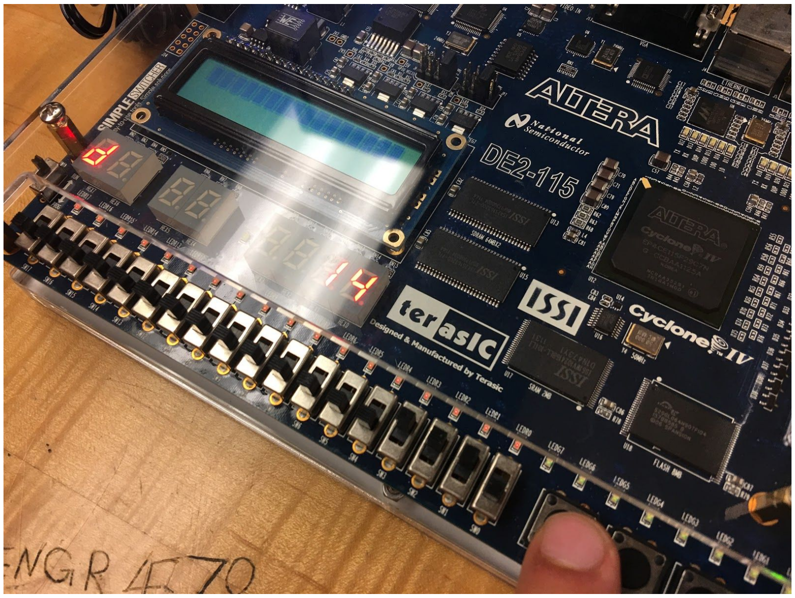
```
            4'b0001:
                hex0 = one;
            4'b0010:
                hex0 = two;
            4'b0011:
                hex0 = three;
            4'b0100:
                hex0 = four;
            4'b0101:
                hex0 = five;
            4'b0110:
                hex0 = six;
            4'b0111:
                hex0 = seven;
            4'b1000:
                hex0 = eight;
            4'b1001:
                hex0 = nine;
            4'b1010: begin
                hex1 = one;
                hex0 = zero;
                end
            4'b1011: begin
                hex1 = one;
                hex0 = one;
                end
            4'b1100: begin
                hex1 = one;
                hex0 = two;
                end
            4'b1101: begin
                hex1 = one;
                hex0 = three;
                end
            4'b1110: begin
                hex1 = one;
                hex0 = four;
                end
            4'b1111: begin
                hex1 = one;
                hex0 = five;
                end
        endcase
    end
end


endmodule
```

Results/Verification:

Verification of the functionality of the system is achieved by implementing the verilog model onto the FPGA board, which allows the user to test the verilog code by visual inspection, rather than relying on a test bench with outputs that are seen in plots.

The corresponding outputs should be expected when pressing the matching button or switch, the code is entered through the onboard switches while the type of conversion will be indicated by pressing on of the three available push buttons.



The 4-bit binary code is inputted using the switches. When the one of the button is pressed, the 4-bit code is converted to a Hexadecimal, Decimal, or BCD Decimal. For example, if the 4-bit code is 0110, when the *Key0* is pushed, the 4-bit code is converted to Hexadecimal.

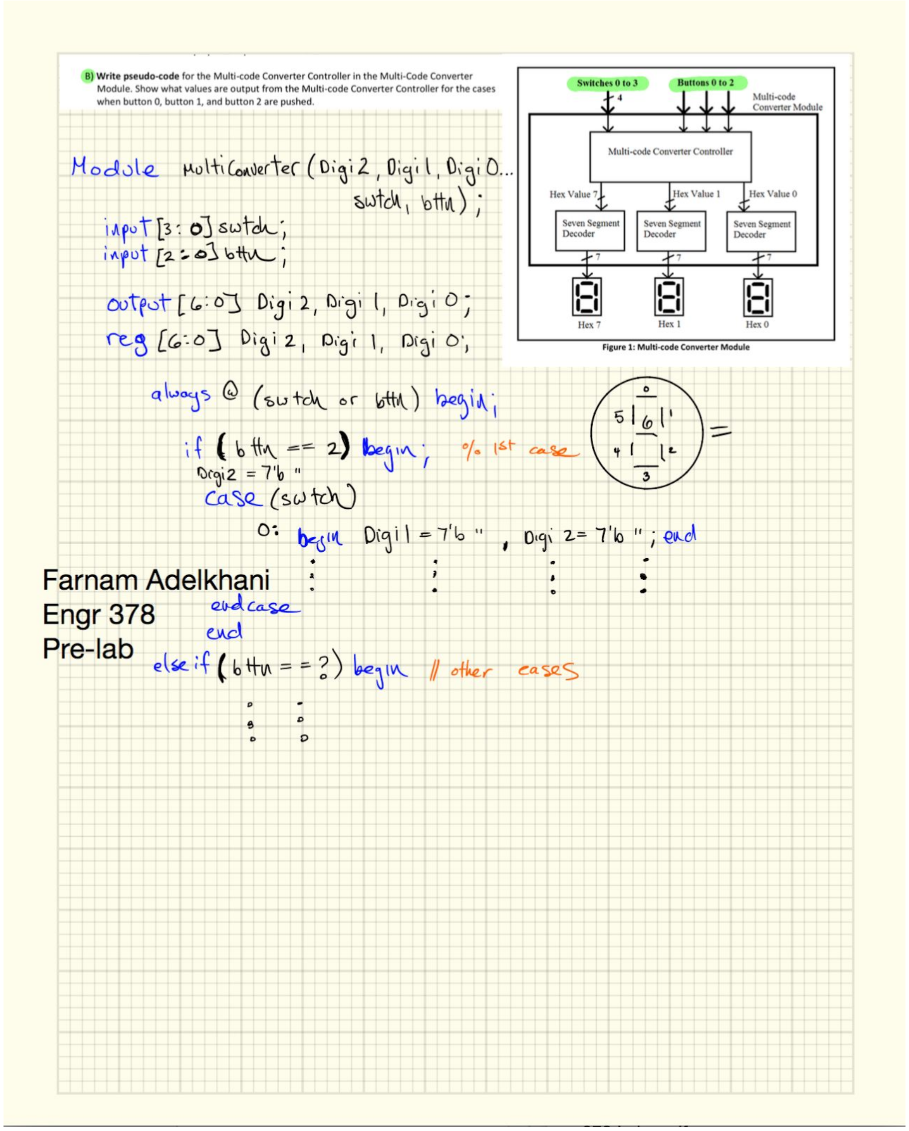
Conclusion and Discussion:

As an engineer student, we have been hearing about FPGA devices for a long time but having the ability to actually implement a system such as the 7-segment display device seems like a reward for learning to program. The FPGA is a powerful module to implemented a wide range of possible outcomes for practice and possible real-life application. The challenge in this assignment was to provide all the conversions to verilog in the model; this required the user to setup an array of inputs to the system through the verilog module that was implemented.

Work Breakdown

- Pre-lab assignments were completed individually.
- The laboratory assignment was completed as a team effort.

Prelabs



B) **Write pseudo-code** for the Multi-code Converter Controller in the Multi-Code Converter Module. Show what values are output from the Multi-code Converter Controller for the cases when button 0, button 1, and button 2 are pushed.

Switches 0 to 3
Buttons 0 to 2
4
Multi-code Converter Module
Multi-code Converter Controller
Hex Value 7
Hex Value 1
Hex Value 0
Seven Segment Decoder
Seven Segment Decoder
Seven Segment Decoder
7
7
7
Hex 7
Hex 1
Hex 0
0
5
6
1
4
2
3
=

Figure 1: Multi-code Converter Module

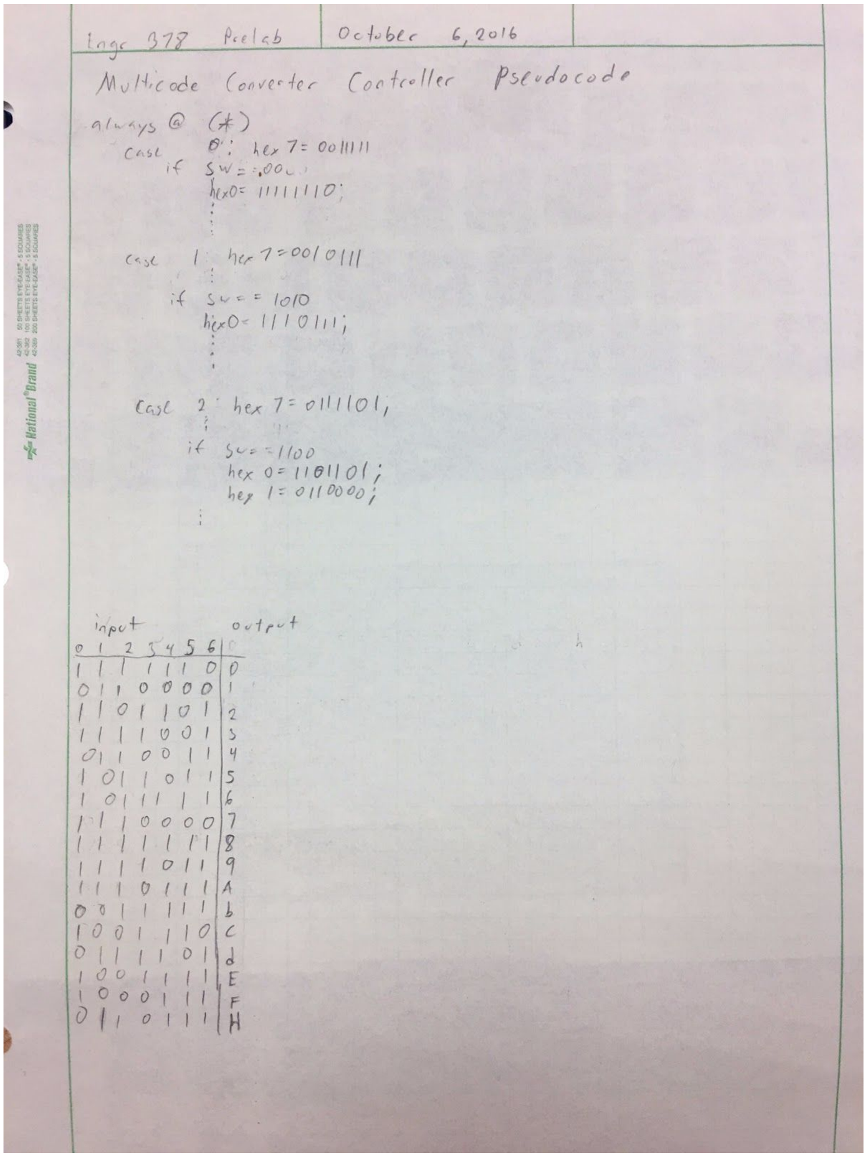
```
Module  MultiConverter (Digi2, Digi1, Digi0...
                                 swtch, bttn);
   input [3:0] swtch;
   input [2:0] bttn;

   output [6:0] Digi2, Digi1, Digi0;
   reg [6:0] Digi2, Digi1, Digi0;

      always @ (swtch or bttn) begin;

        if (bttn == 2) begin;    % 1st case
        Digi2 = 7'b "
         case (swtch)
              0:  begin  Digi1 = 7'b "  , Digi 2= 7'b " ; end
                  :          :          :          :
         endcase
         end
      elseif (bttn = = ?) begin  // other cases
              :     :
```

Farnam Adelkhani
Engr 378
Pre-lab



Engr 378 Prelab | October 6, 2016

Multicode Converter Controller Pseudocode

```
always @ (*)
    case    0:  hex 7= 0011111
        if  SW = =00
            hex0= 1111110;
            :
            :

    case   1:  hex 7=0010111
            :
        if  sw = = 1010
            hex0= 1110111;
            :
            :

     case  2:  hex 7= 0111101;
            :
          if  sw= =1100
              hex 0= 1101101;
              hex 1= 0110000;
            :
```

input							output
0	1	2	3	4	5	6	
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	1	1	9
1	1	1	0	1	1	1	A
0	0	1	1	1	1	1	b
1	0	0	1	1	1	0	C
0	1	1	1	1	0	1	d
1	0	0	1	1	1	1	E
1	0	0	0	1	1	1	F
0	1	1	0	1	1	1	H