BASYS 2 FPGA – Number Generation (Seven Segment)

Santosh Kumar Dubey - 5009 7059

Contents

Objective

The Objective of this project was to program BASYS2 Board and perform desired operations by FPGA and peripherals i.e. 7-Segment display (Available : 4 and Used : 2) , Slide Switches (Available : 7 and Used : 7) and Buttons (Available : 4 and Used : 2) , We have to perform following tasks by programming FPGA .

1. If slide switches are ON then corresponding LED should glow (i.e. 8^{th} and 1^{st} is ON then 8^{th} and 1^{st} LED should Glow)
2. Any number of switch can be ON at a time ,By Pressing BTN0 corresponding switch numbers should be displayed on 2 Digits (Seg1, Seg0) of available 4 7-Segments Display rest two will be OFF (i.e. if 8^{th} and 1^{st} Slide Switch is ON , then 2 Digits of 7-Segment Display will show 81)
3. If we Press BTN1 then 7-Segment Display should go to OFF.
4. Display on 7-Segment should be updated after pressing BTN0 and reset on pressing BTN1 (i.e. if slide switch 7 and 2 are on and Value on 7-Segment should be 72 after pressing BTN0, now we have change the slide switch positions from 7 and 2 to 6 and 3 but 7-Segment should show value 72 only, after pressing BTN0 value should be updated to 63).

Introduction

In this project we have to use Xilinx FPGA and after writing code we have to test the functionality of code in simulation by using Testbench and Synthesize the code (except Testbench) and check the timing in synthesis and Generate a BIT file , which should be programmed into FPGA by using BASYS2 Software tool. The main block of this project1 is decoder block which actually decodes the switch values into corresponding 7-Segment values and other is controller block which exactly control the 7-Segment
Cathode and Anode timing and refresh rate and control block also controls the buttons inputs, as motioned in project objective , by pressing button 0 (BTN0) value of switch should be updated in 7-segment and by pressing button 1 (BTN1) 7-segment should reset as well system will also reset .
But there are few corner cases of slide switches those are following

1. If No Slide Switch is ON then what will happen on 7-Segment Display.
2. If One Slide Switch is ON then what will happen on 7-Segment Display.
3. If more than Two Slide Switches are ON then what will happen on 7-Segment Display.

I have taken care of above corner cases in the Design ,
In case #1 7-Segment will Display “00” which will give idea to user No 7-Segment Display is ON ,
In case #2 7-Segment will Display number of that switch , we have divided 4 and 4 switches to 2 digits , support 4^{th} switch is ON then 10 will be displayed on seaven segment
In case #3 it will also display corresponding number support 0^{th} ,1^{st} and 4^{th} is ON then it will display “13”.

Features

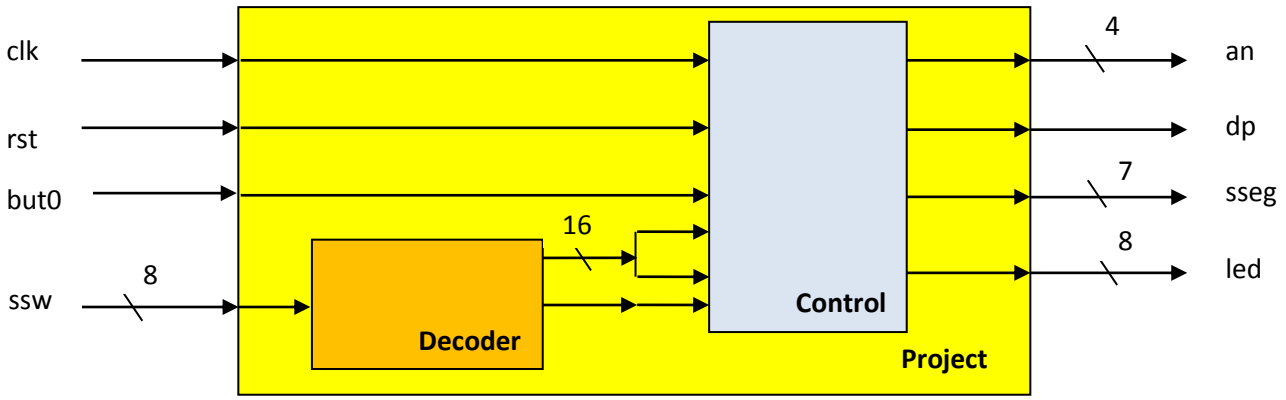
- Works on 50 MHz
- Error (Er) Implementation on wrong input of slider switch.
- Display and System Reset is same both are Asynchronous.
- Optimized logic used for the project

Design

As I have described in introduction that design has two major module control and switch_deocde, but we have created a top module which has instances of both the module , we have describe the basic functionality input and output which pin diagram in following sections :

Top Module (project1.v)

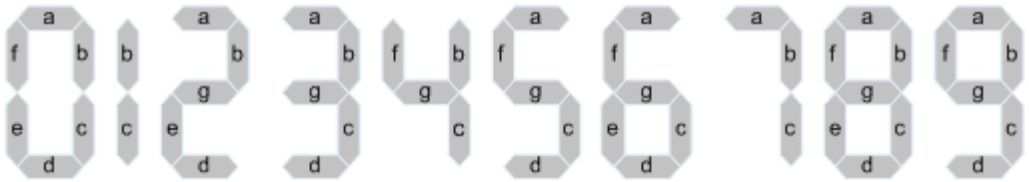
This module has only instantiation of decoder (switch_decode.v) and control (ctrl.v) block , Only one simple logic is there output digits (14 bit) is splitted into digit0 (6:0) and digit1 (13:7) , where digit0 and digit1 are input of Control block.



Signal Name	Bit	I/O	Description
Clk	1	I	System Clock (50Mhz)
Rst	1	I	System/Display Reset When “1” : System Reset When “0” : System Mode
But0	1	I	Button 0 when press switch should updated on 7-Segment display When “1” : Button is ON When “0” : Button is OFF
ssw	8	I	Slide switch Input When “1” : Switch is ON When “0” : Switch is OFF
Led	8	O	Output of Corresponding LEDs , it should be same as SSW
sseg	7	O	Input to 7-Segment Display (Cathode to 7-Segment Display)
An	4	O	Input to 7-Segment Display with (Anode Values for all 4 Display , but Only 2 is Active Other are High)
Dp	1	O	Decimal Point to 7-Segment High Always to Display decimal points

Decoder (switch_decode.v)

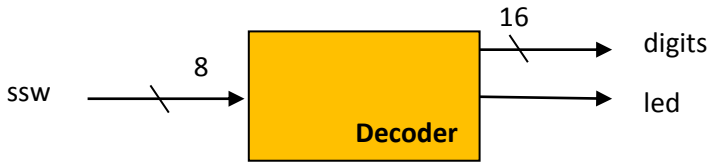
This decoder decodes the number and words corresponding to switch following are the truth table of seven segment decoder but we have implemented for two digits which as we can see in the following diagram



Truth Table for a 7-segment display

Individual Segments							Display
a	b	c	d	e	f	g	
0	0	0	0	0	0	1	0
1	0	0	1	1	1	1	1
0	0	1	0	0	1	0	2
0	0	0	0	1	1	0	3
1	0	0	1	1	0	0	4
0	1	0	0	1	0	0	5
0	1	0	0	0	0	0	6
0	0	0	1	1	1	1	7

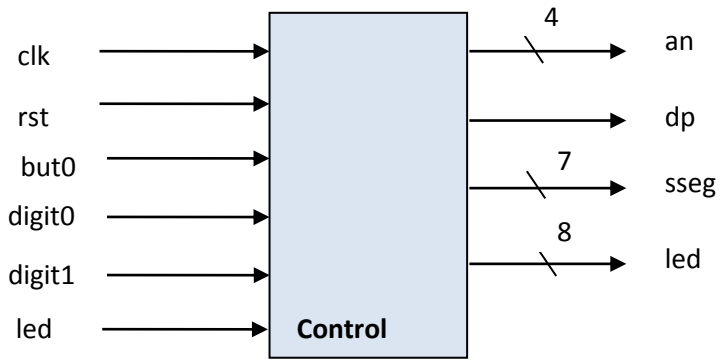
Individual Segments							Display
a	b	c	d	e	f	g	
0	0	0	0	0	0	0	8
0	0	0	0	1	0	0	9
0	0	0	1	0	0	0	A
1	1	0	0	0	0	0	b
0	1	1	0	0	0	1	C
1	0	0	0	0	1	0	d
0	1	1	0	0	0	0	E
0	1	1	1	0	0	0	F



Signal Name	BITS	I/O	Description
ssw	8	I	Slide switch input value , Valid Condition: Only two bit should be High at a time. Bit value 1 : “Switch is ON” Bit value 0 : “Switch is OFF”
digits	16	O	Output value for Both 7-Segment Bits corresponding to switches
led	8	O	Same Values as Switches

Controller (ctrl.v)

Basically Controller is actual controller of the project it will control the 7-segment display digits, leds and act on button 0 and button 1, this block has 50 MHz clock input and rst which is connected with button 1 whenever button 1 is pressed then system/display will reset. When we press button 0 then current value of slide switch will update on 7-Segment Display, there is Internal counter in control block which actually generates the an[4] signals . sseg is 7-Segment output and led is output which glows corresponding leds to Slide switches.



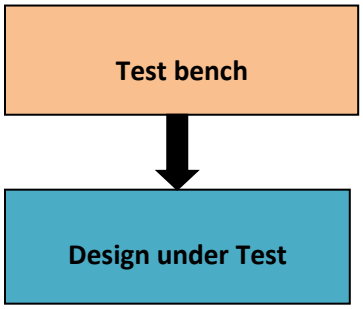
Signal Name	BITS	I/O	Description
clk	1	I	50 MHz clock signals
rst	1	I	System/display reset signal
but0	1	I	Input button which updates value on 7-Segment Display
digit0	7	I	Input for Digit 0
digit1	7	I	Input for Digit 1
led	8	I	LED input
an	4	O	Anode for 7-Segment Display
dp	1	O	Decimal point (Hardcoded “1” Disabled)
sseg	7	O	Cathode for 7-Segment Display
led	8	O	LED Output same as input

Verification

As we know verification is more important than design because poor design is not acceptable by anyone so we have described perfect verification environment , and it is always good to provide auto check mechanism which will inform user for pass and failure ,

Test bench (tb.v)

As we in the below diagram test bench and DUT connected with input and output input should be exercised by DUT and output should be compared by Test bench.



Clock generation

```
always begin
clk1 = 0;
forever #1 clk1 = ~clk1;
end
```

tasks

There are two task in the test bench initialize and switch_set , initialize will initialize all the inputs and reset the system , however switch_set is used to supply value of slide switches i.e.slide_set(42) means 4th and 2nd switch is ON.

```
task initialize;
begin
rst  = 1'b1;
but0 = 1'h0;
but1 = 1'b0;
ssw = 7'b0;
#10 rst =1'b0;
end
endtask

task switch_set;
input [7:0] slide_switch1;
begin
ssw=slide_switch1;
repeat(2)@(posedge clk1);
but0=1'b1;
repeat(5)@(posedge clk1);
but0=1'b0;
wait(pro.ctrl0.temp);
wait(~pro.ctrl0.temp);
end
endtask
```

Auto check feature

This is major feature of Our testbench , this feature will take output and re-convert into format of input and compare with actual input and this will also check for error scenario's in the design.

Logfile

Following log file is printed for all the tests in simulation

```
--Tests Start--
Finished circuit initialization process.
--Basic Tests--
At Time : 196617  Result : <PASS> Actual Data : 01 Expected Data : 01
At Time : 720905  Result : <PASS> Actual Data : 02 Expected Data : 02
At Time : 1245193  Result : <PASS> Actual Data : 03 Expected Data : 03
At Time : 1769481  Result : <PASS> Actual Data : 04 Expected Data : 04
At Time : 2293769  Result : <PASS> Actual Data : 05 Expected Data : 05
At Time : 2818057  Result : <PASS> Actual Data : 06 Expected Data : 06
At Time : 3342345  Result : <PASS> Actual Data : 07 Expected Data : 07
At Time : 3866633  Result : <PASS> Actual Data : 08 Expected Data : 08
At Time : 4390921  Result : <PASS> Actual Data : 09 Expected Data : 09
At Time : 4915209  Result : <PASS> Actual Data : 0a Expected Data : 0a
At Time : 5439497  Result : <PASS> Actual Data : 0b Expected Data : 0b
At Time : 5963785  Result : <PASS> Actual Data : 0c Expected Data : 0c
At Time : 6488073  Result : <PASS> Actual Data : 0d Expected Data : 0d
At Time : 7012361  Result : <PASS> Actual Data : 0e Expected Data : 0e
At Time : 7536649  Result : <PASS> Actual Data : 0f Expected Data : 0f
At Time : 8060937  Result : <PASS> Actual Data : 10 Expected Data : 10
At Time : 8585225  Result : <PASS> Actual Data : 11 Expected Data : 11
At Time : 9109513  Result : <PASS> Actual Data : 12 Expected Data : 12
At Time : 9633801  Result : <PASS> Actual Data : 13 Expected Data : 13
At Time : 10158089  Result : <PASS> Actual Data : 14 Expected Data : 14
At Time : 10682377  Result : <PASS> Actual Data : 15 Expected Data : 15
At Time : 11206665  Result : <PASS> Actual Data : 16 Expected Data : 16
At Time : 11730953  Result : <PASS> Actual Data : 17 Expected Data : 17
At Time : 12255241  Result : <PASS> Actual Data : 18 Expected Data : 18
At Time : 12779529  Result : <PASS> Actual Data : 19 Expected Data : 19
At Time : 13303817  Result : <PASS> Actual Data : 1a Expected Data : 1a
At Time : 13828105  Result : <PASS> Actual Data : 1b Expected Data : 1b
At Time : 14352393  Result : <PASS> Actual Data : 1c Expected Data : 1c
At Time : 14876681  Result : <PASS> Actual Data : 1d Expected Data : 1d
At Time : 15400969  Result : <PASS> Actual Data : 1e Expected Data : 1e
At Time : 15925257  Result : <PASS> Actual Data : 1f Expected Data : 1f
At Time : 16449545  Result : <PASS> Actual Data : 20 Expected Data : 20
At Time : 16973833  Result : <PASS> Actual Data : 21 Expected Data : 21
At Time : 17498121  Result : <PASS> Actual Data : 22 Expected Data : 22
At Time : 18022409  Result : <PASS> Actual Data : 23 Expected Data : 23
At Time : 18546697  Result : <PASS> Actual Data : 24 Expected Data : 24
At Time : 19070985  Result : <PASS> Actual Data : 25 Expected Data : 25
At Time : 19595273  Result : <PASS> Actual Data : 26 Expected Data : 26
At Time : 20119561  Result : <PASS> Actual Data : 27 Expected Data : 27
At Time : 20643849  Result : <PASS> Actual Data : 28 Expected Data : 28
At Time : 21168137  Result : <PASS> Actual Data : 29 Expected Data : 29
At Time : 21692425  Result : <PASS> Actual Data : 2a Expected Data : 2a
At Time : 22216713  Result : <PASS> Actual Data : 2b Expected Data : 2b
At Time : 22741001  Result : <PASS> Actual Data : 2c Expected Data : 2c
At Time : 23265289  Result : <PASS> Actual Data : 2d Expected Data : 2d
At Time : 23789577  Result : <PASS> Actual Data : 2e Expected Data : 2e
At Time : 24313865  Result : <PASS> Actual Data : 2f Expected Data : 2f
At Time : 24838153  Result : <PASS> Actual Data : 30 Expected Data : 30
At Time : 25362441  Result : <PASS> Actual Data : 31 Expected Data : 31
At Time : 25886729  Result : <PASS> Actual Data : 32 Expected Data : 32
At Time : 26411017  Result : <PASS> Actual Data : 33 Expected Data : 33
At Time : 26935305  Result : <PASS> Actual Data : 34 Expected Data : 34
At Time : 27459593  Result : <PASS> Actual Data : 35 Expected Data : 35
At Time : 27983881  Result : <PASS> Actual Data : 36 Expected Data : 36
At Time : 28508169  Result : <PASS> Actual Data : 37 Expected Data : 37
At Time : 29032457  Result : <PASS> Actual Data : 38 Expected Data : 38
At Time : 29556745  Result : <PASS> Actual Data : 39 Expected Data : 39
At Time : 30081033  Result : <PASS> Actual Data : 3a Expected Data : 3a
At Time : 30605321  Result : <PASS> Actual Data : 3b Expected Data : 3b
At Time : 31129609  Result : <PASS> Actual Data : 3c Expected Data : 3c
At Time : 31653897  Result : <PASS> Actual Data : 3d Expected Data : 3d
At Time : 32178185  Result : <PASS> Actual Data : 3e Expected Data : 3e
At Time : 32702473  Result : <PASS> Actual Data : 3f Expected Data : 3f
At Time : 33226761  Result : <PASS> Actual Data : 40 Expected Data : 40
At Time : 33751049  Result : <PASS> Actual Data : 41 Expected Data : 41
At Time : 34275337  Result : <PASS> Actual Data : 42 Expected Data : 42
At Time : 34799625  Result : <PASS> Actual Data : 43 Expected Data : 43
At Time : 35323913  Result : <PASS> Actual Data : 44 Expected Data : 44
At Time : 35848201  Result : <PASS> Actual Data : 45 Expected Data : 45
```

```
At Time : 36372489  Result : <PASS> Actual Data : 46 Expected Data : 46
At Time : 36896777  Result : <PASS> Actual Data : 47 Expected Data : 47
At Time : 37421065  Result : <PASS> Actual Data : 48 Expected Data : 48
At Time : 37945353  Result : <PASS> Actual Data : 49 Expected Data : 49
At Time : 38469641  Result : <PASS> Actual Data : 4a Expected Data : 4a
At Time : 38993929  Result : <PASS> Actual Data : 4b Expected Data : 4b
At Time : 39518217  Result : <PASS> Actual Data : 4c Expected Data : 4c
At Time : 40042505  Result : <PASS> Actual Data : 4d Expected Data : 4d
At Time : 40566793  Result : <PASS> Actual Data : 4e Expected Data : 4e
At Time : 41091081  Result : <PASS> Actual Data : 4f Expected Data : 4f
At Time : 41615369  Result : <PASS> Actual Data : 50 Expected Data : 50
At Time : 42139657  Result : <PASS> Actual Data : 51 Expected Data : 51
At Time : 42663945  Result : <PASS> Actual Data : 52 Expected Data : 52
At Time : 43188233  Result : <PASS> Actual Data : 53 Expected Data : 53
At Time : 43712521  Result : <PASS> Actual Data : 54 Expected Data : 54
At Time : 44236809  Result : <PASS> Actual Data : 55 Expected Data : 55
At Time : 44761097  Result : <PASS> Actual Data : 56 Expected Data : 56
At Time : 45285385  Result : <PASS> Actual Data : 57 Expected Data : 57
At Time : 45809673  Result : <PASS> Actual Data : 58 Expected Data : 58
At Time : 46333961  Result : <PASS> Actual Data : 59 Expected Data : 59
At Time : 46858249  Result : <PASS> Actual Data : 5a Expected Data : 5a
At Time : 47382537  Result : <PASS> Actual Data : 5b Expected Data : 5b
At Time : 47906825  Result : <PASS> Actual Data : 5c Expected Data : 5c
At Time : 48431113  Result : <PASS> Actual Data : 5d Expected Data : 5d
At Time : 48955401  Result : <PASS> Actual Data : 5e Expected Data : 5e
At Time : 49479689  Result : <PASS> Actual Data : 5f Expected Data : 5f
At Time : 50003977  Result : <PASS> Actual Data : 60 Expected Data : 60
At Time : 50528265  Result : <PASS> Actual Data : 61 Expected Data : 61
At Time : 51052553  Result : <PASS> Actual Data : 62 Expected Data : 62
At Time : 51576841  Result : <PASS> Actual Data : 63 Expected Data : 63
At Time : 52101129  Result : <PASS> Actual Data : 64 Expected Data : 64
At Time : 52625417  Result : <PASS> Actual Data : 65 Expected Data : 65
At Time : 53149705  Result : <PASS> Actual Data : 66 Expected Data : 66
At Time : 53673993  Result : <PASS> Actual Data : 67 Expected Data : 67
At Time : 54198281  Result : <PASS> Actual Data : 68 Expected Data : 68
At Time : 54722569  Result : <PASS> Actual Data : 69 Expected Data : 69
At Time : 55246857  Result : <PASS> Actual Data : 6a Expected Data : 6a
At Time : 55771145  Result : <PASS> Actual Data : 6b Expected Data : 6b
At Time : 56295433  Result : <PASS> Actual Data : 6c Expected Data : 6c
At Time : 56819721  Result : <PASS> Actual Data : 6d Expected Data : 6d
At Time : 57344009  Result : <PASS> Actual Data : 6e Expected Data : 6e
At Time : 57868297  Result : <PASS> Actual Data : 6f Expected Data : 6f
At Time : 58392585  Result : <PASS> Actual Data : 70 Expected Data : 70
At Time : 58916873  Result : <PASS> Actual Data : 71 Expected Data : 71
At Time : 59441161  Result : <PASS> Actual Data : 72 Expected Data : 72
At Time : 59965449  Result : <PASS> Actual Data : 73 Expected Data : 73
At Time : 60489737  Result : <PASS> Actual Data : 74 Expected Data : 74
At Time : 61014025  Result : <PASS> Actual Data : 75 Expected Data : 75
At Time : 61538313  Result : <PASS> Actual Data : 76 Expected Data : 76
At Time : 62062601  Result : <PASS> Actual Data : 77 Expected Data : 77
At Time : 62586889  Result : <PASS> Actual Data : 78 Expected Data : 78
At Time : 63111177  Result : <PASS> Actual Data : 79 Expected Data : 79
At Time : 63635465  Result : <PASS> Actual Data : 7a Expected Data : 7a
At Time : 64159753  Result : <PASS> Actual Data : 7b Expected Data : 7b
At Time : 64684041  Result : <PASS> Actual Data : 7c Expected Data : 7c
At Time : 65208329  Result : <PASS> Actual Data : 7d Expected Data : 7d
At Time : 65732617  Result : <PASS> Actual Data : 7e Expected Data : 7e
At Time : 66256905  Result : <PASS> Actual Data : 7f Expected Data : 7f
At Time : 66781193  Result : <PASS> Actual Data : 80 Expected Data : 80
At Time : 67305481  Result : <PASS> Actual Data : 81 Expected Data : 81
At Time : 67829769  Result : <PASS> Actual Data : 82 Expected Data : 82
At Time : 68354057  Result : <PASS> Actual Data : 83 Expected Data : 83
At Time : 68878345  Result : <PASS> Actual Data : 84 Expected Data : 84
At Time : 69402633  Result : <PASS> Actual Data : 85 Expected Data : 85
At Time : 69926921  Result : <PASS> Actual Data : 86 Expected Data : 86
At Time : 70451209  Result : <PASS> Actual Data : 87 Expected Data : 87
At Time : 70975497  Result : <PASS> Actual Data : 88 Expected Data : 88
At Time : 71499785  Result : <PASS> Actual Data : 89 Expected Data : 89
At Time : 72024073  Result : <PASS> Actual Data : 8a Expected Data : 8a
At Time : 72548361  Result : <PASS> Actual Data : 8b Expected Data : 8b
At Time : 73072649  Result : <PASS> Actual Data : 8c Expected Data : 8c
At Time : 73596937  Result : <PASS> Actual Data : 8d Expected Data : 8d
At Time : 74121225  Result : <PASS> Actual Data : 8e Expected Data : 8e
At Time : 74645513  Result : <PASS> Actual Data : 8f Expected Data : 8f
At Time : 75169801  Result : <PASS> Actual Data : 90 Expected Data : 90
At Time : 75694089  Result : <PASS> Actual Data : 91 Expected Data : 91
```

```
At Time : 76218377  Result : <PASS> Actual Data : 92 Expected Data : 92
At Time : 76742665  Result : <PASS> Actual Data : 93 Expected Data : 93
At Time : 77266953  Result : <PASS> Actual Data : 94 Expected Data : 94
At Time : 77791241  Result : <PASS> Actual Data : 95 Expected Data : 95
At Time : 78315529  Result : <PASS> Actual Data : 96 Expected Data : 96
At Time : 78839817  Result : <PASS> Actual Data : 97 Expected Data : 97
At Time : 79364105  Result : <PASS> Actual Data : 98 Expected Data : 98
At Time : 79888393  Result : <PASS> Actual Data : 99 Expected Data : 99
At Time : 80412681  Result : <PASS> Actual Data : 9a Expected Data : 9a
At Time : 80936969  Result : <PASS> Actual Data : 9b Expected Data : 9b
At Time : 81461257  Result : <PASS> Actual Data : 9c Expected Data : 9c
At Time : 81985545  Result : <PASS> Actual Data : 9d Expected Data : 9d
At Time : 82509833  Result : <PASS> Actual Data : 9e Expected Data : 9e
At Time : 83034121  Result : <PASS> Actual Data : 9f Expected Data : 9f
At Time : 83558409  Result : <PASS> Actual Data : a0 Expected Data : a0
At Time : 84082697  Result : <PASS> Actual Data : a1 Expected Data : a1
At Time : 84606985  Result : <PASS> Actual Data : a2 Expected Data : a2
At Time : 85131273  Result : <PASS> Actual Data : a3 Expected Data : a3
At Time : 85655561  Result : <PASS> Actual Data : a4 Expected Data : a4
At Time : 86179849  Result : <PASS> Actual Data : a5 Expected Data : a5
At Time : 86704137  Result : <PASS> Actual Data : a6 Expected Data : a6
At Time : 87228425  Result : <PASS> Actual Data : a7 Expected Data : a7
At Time : 87752713  Result : <PASS> Actual Data : a8 Expected Data : a8
At Time : 88277001  Result : <PASS> Actual Data : a9 Expected Data : a9
At Time : 88801289  Result : <PASS> Actual Data : aa Expected Data : aa
At Time : 89325577  Result : <PASS> Actual Data : ab Expected Data : ab
At Time : 89849865  Result : <PASS> Actual Data : ac Expected Data : ac
At Time : 90374153  Result : <PASS> Actual Data : ad Expected Data : ad
At Time : 90898441  Result : <PASS> Actual Data : ae Expected Data : ae
At Time : 91422729  Result : <PASS> Actual Data : af Expected Data : af
At Time : 91947017  Result : <PASS> Actual Data : b0 Expected Data : b0
At Time : 92471305  Result : <PASS> Actual Data : b1 Expected Data : b1
At Time : 92995593  Result : <PASS> Actual Data : b2 Expected Data : b2
At Time : 93519881  Result : <PASS> Actual Data : b3 Expected Data : b3
At Time : 94044169  Result : <PASS> Actual Data : b4 Expected Data : b4
At Time : 94568457  Result : <PASS> Actual Data : b5 Expected Data : b5
At Time : 95092745  Result : <PASS> Actual Data : b6 Expected Data : b6
At Time : 95617033  Result : <PASS> Actual Data : b7 Expected Data : b7
At Time : 96141321  Result : <PASS> Actual Data : b8 Expected Data : b8
At Time : 96665609  Result : <PASS> Actual Data : b9 Expected Data : b9
At Time : 97189897  Result : <PASS> Actual Data : ba Expected Data : ba
At Time : 97714185  Result : <PASS> Actual Data : bb Expected Data : bb
At Time : 98238473  Result : <PASS> Actual Data : bc Expected Data : bc
At Time : 98762761  Result : <PASS> Actual Data : bd Expected Data : bd
At Time : 99287049  Result : <PASS> Actual Data : be Expected Data : be
At Time : 99811337  Result : <PASS> Actual Data : bf Expected Data : bf
At Time : 100335625  Result : <PASS> Actual Data : c0 Expected Data : c0
At Time : 100859913  Result : <PASS> Actual Data : c1 Expected Data : c1
At Time : 101384201  Result : <PASS> Actual Data : c2 Expected Data : c2
At Time : 101908489  Result : <PASS> Actual Data : c3 Expected Data : c3
At Time : 102432777  Result : <PASS> Actual Data : c4 Expected Data : c4
At Time : 102957065  Result : <PASS> Actual Data : c5 Expected Data : c5
At Time : 103481353  Result : <PASS> Actual Data : c6 Expected Data : c6
At Time : 104005641  Result : <PASS> Actual Data : c7 Expected Data : c7
At Time : 104529929  Result : <PASS> Actual Data : c8 Expected Data : c8
At Time : 105054217  Result : <PASS> Actual Data : c9 Expected Data : c9
At Time : 105578505  Result : <PASS> Actual Data : ca Expected Data : ca
At Time : 106102793  Result : <PASS> Actual Data : cb Expected Data : cb
At Time : 106627081  Result : <PASS> Actual Data : cc Expected Data : cc
At Time : 107151369  Result : <PASS> Actual Data : cd Expected Data : cd
At Time : 107675657  Result : <PASS> Actual Data : ce Expected Data : ce
At Time : 108199945  Result : <PASS> Actual Data : cf Expected Data : cf
At Time : 108724233  Result : <PASS> Actual Data : d0 Expected Data : d0
At Time : 109248521  Result : <PASS> Actual Data : d1 Expected Data : d1
At Time : 109772809  Result : <PASS> Actual Data : d2 Expected Data : d2
At Time : 110297097  Result : <PASS> Actual Data : d3 Expected Data : d3
At Time : 110821385  Result : <PASS> Actual Data : d4 Expected Data : d4
At Time : 111345673  Result : <PASS> Actual Data : d5 Expected Data : d5
At Time : 111869961  Result : <PASS> Actual Data : d6 Expected Data : d6
At Time : 112394249  Result : <PASS> Actual Data : d7 Expected Data : d7
At Time : 112918537  Result : <PASS> Actual Data : d8 Expected Data : d8
At Time : 113442825  Result : <PASS> Actual Data : d9 Expected Data : d9
At Time : 113967113  Result : <PASS> Actual Data : da Expected Data : da
At Time : 114491401  Result : <PASS> Actual Data : db Expected Data : db
At Time : 115015689  Result : <PASS> Actual Data : dc Expected Data : dc
At Time : 115539977  Result : <PASS> Actual Data : dd Expected Data : dd
```

```
At Time : 116064265  Result : <PASS> Actual Data : de Expected Data : de
At Time : 116588553  Result : <PASS> Actual Data : df Expected Data : df
At Time : 117112841  Result : <PASS> Actual Data : e0 Expected Data : e0
At Time : 117637129  Result : <PASS> Actual Data : e1 Expected Data : e1
At Time : 118161417  Result : <PASS> Actual Data : e2 Expected Data : e2
At Time : 118685705  Result : <PASS> Actual Data : e3 Expected Data : e3
At Time : 119209993  Result : <PASS> Actual Data : e4 Expected Data : e4
At Time : 119734281  Result : <PASS> Actual Data : e5 Expected Data : e5
At Time : 120258569  Result : <PASS> Actual Data : e6 Expected Data : e6
At Time : 120782857  Result : <PASS> Actual Data : e7 Expected Data : e7
At Time : 121307145  Result : <PASS> Actual Data : e8 Expected Data : e8
At Time : 121831433  Result : <PASS> Actual Data : e9 Expected Data : e9
At Time : 122355721  Result : <PASS> Actual Data : ea Expected Data : ea
At Time : 122880009  Result : <PASS> Actual Data : eb Expected Data : eb
At Time : 123404297  Result : <PASS> Actual Data : ec Expected Data : ec
At Time : 123928585  Result : <PASS> Actual Data : ed Expected Data : ed
At Time : 124452873  Result : <PASS> Actual Data : ee Expected Data : ee
At Time : 124977161  Result : <PASS> Actual Data : ef Expected Data : ef
At Time : 125501449  Result : <PASS> Actual Data : f0 Expected Data : f0
At Time : 126025737  Result : <PASS> Actual Data : f1 Expected Data : f1
At Time : 126550025  Result : <PASS> Actual Data : f2 Expected Data : f2
At Time : 127074313  Result : <PASS> Actual Data : f3 Expected Data : f3
At Time : 127598601  Result : <PASS> Actual Data : f4 Expected Data : f4
At Time : 128122889  Result : <PASS> Actual Data : f5 Expected Data : f5
At Time : 128647177  Result : <PASS> Actual Data : f6 Expected Data : f6
At Time : 129171465  Result : <PASS> Actual Data : f7 Expected Data : f7
At Time : 129695753  Result : <PASS> Actual Data : f8 Expected Data : f8
At Time : 130220041  Result : <PASS> Actual Data : f9 Expected Data : f9
At Time : 130744329  Result : <PASS> Actual Data : fa Expected Data : fa
At Time : 131268617  Result : <PASS> Actual Data : fb Expected Data : fb
At Time : 131792905  Result : <PASS> Actual Data : fc Expected Data : fc
At Time : 132317193  Result : <PASS> Actual Data : fd Expected Data : fd
At Time : 132841481  Result : <PASS> Actual Data : fe Expected Data : fe
At Time : 133365769  Result : <PASS> Actual Data : ff Expected Data : ff
```

FPGA

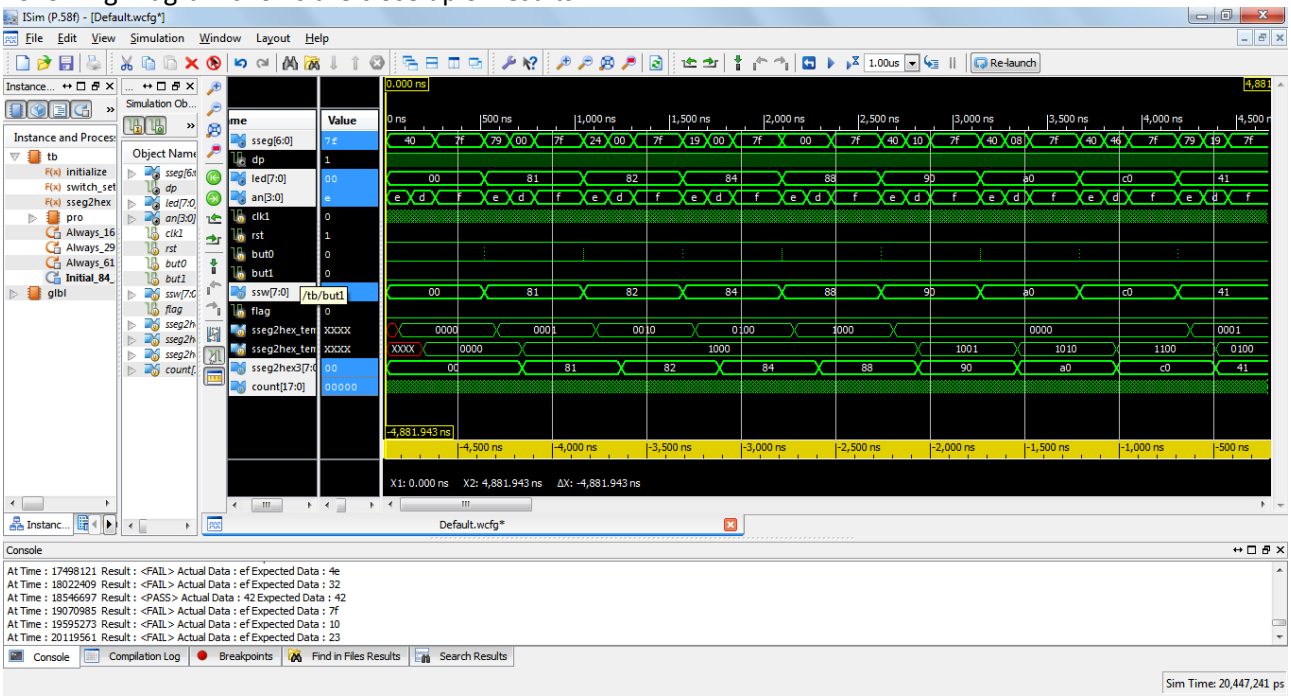
We have used Spartan 3E series of FPGA

Results

Simulation

Following is screenshot of TB Simulation Results

Following Diagram shows the close up of results



Synthesis

```
Following is the Synthesis report which gives idea about resources used in Xilinx Spartan
3E FPGA
=========================================================================
HDL Synthesis Report

Macro Statistics
# ROMs                                                 : 1
 4x4-bit ROM                                          : 1
# Counters                                             : 1
 18-bit up counter                                    : 1
# Registers                                            : 2
 7-bit register                                       : 2
# Multiplexers                                         : 1
 7-bit 4-to-1 multiplexer                             : 1

=========================================================================

=========================================================================
*                        Advanced HDL Synthesis                         *
=========================================================================

=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# ROMs                                                 : 1
 4x4-bit ROM                                          : 1
# Counters                                             : 1
 18-bit up counter                                    : 1
# Registers                                            : 14
 Flip-Flops                                           : 14
# Multiplexers                                         : 1
```

```
 7-bit 4-to-1 multiplexer                          : 1
=========================================================================

=========================================================================
*                        Low Level Synthesis                            *
=========================================================================

Optimizing unit <project1> ...

Optimizing unit <ctrl> ...

Mapping all equations...
Building and optimizing final netlist ...
Found area constraint ratio of 100 (+ 5) on block project1, actual ratio is 6.

Final Macro Processing ...

=========================================================================
Final Register Report

Macro Statistics
# Registers                                       : 32
 Flip-Flops                                       : 32

=========================================================================

=========================================================================
*                           Partition Report                            *
=========================================================================

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

-------------------------------
=========================================================================
*                            Final Report                               *
=========================================================================
Clock Information:
------------------
-----------------------------------+------------------------+-------+
Clock Signal                       | Clock buffer(FF name)  | Load  |
-----------------------------------+------------------------+-------+
clk                                | BUFGP                  | 32    |
-----------------------------------+------------------------+-------+
Asynchronous Control Signals Information:
----------------------------------------
-----------------------------------+------------------------+-------+
Control Signal                     | Buffer(FF name)        | Load  |
-----------------------------------+------------------------+-------+
rst                                | IBUF                   | 32    |
-----------------------------------+------------------------+-------+
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: 3.779ns (Maximum Frequency: 264.655MHz)
   Minimum input arrival time before clock: 9.336ns
   Maximum output required time after clock: 5.554ns
   Maximum combinational path delay: 5.339ns

=========================================================================
Process "Synthesize - XST" completed successfully
```

Appendix A (Soruce Codes)

Project1.v

```
//Description : This Module just decode Slide Switches only
//Code is Documented line by line .
//Top module of project where all both the instance
//sd0 and ctrl0 are intanciated.
module project1(rst,clk,but0,ssw,sseg,an,dp,led);

input        rst          ; //Reset Mapped with Button 3
input        clk          ; //System Clock
input        but0         ; //Button 0
input  [7:0] ssw          ; //Slide Switch

output [3:0] an           ; //Seven Segment Enable 0
output       dp           ; //Decimal Enable ( "1" Disabled all the time )
output [6:0] sseg         ; //Seven Segment Output
output [7:0] led          ; //Led Output

wire   [13:0] digits      ;
wire   [ 3:0] an          ;
wire   [ 6:0] sseg        ;

//Switch Decode Logic
switch_decode  sd0( .slide_switch(ssw)
                   ,.outp(digits)
                   ,.ledout(led)
                                          );

//Main Control State Machine
ctrl  ctrl0( .clk(clk)
            ,.rst(rst)
            ,.but0(but0)
            ,.digit0(digits[6:0])
            ,.digit1(digits[13:7])
            ,.dp(dp)
            ,.an(an)
            ,.sseg(sseg)
                           );

endmodule
```

switch_decode.v

```
// Description : This Module will decode and control the slide switch part
// and Output of this Module should be Sent to hex2seven Segment Decodar Module
// Code is Documented  line by line .
module switch_decode(slide_switch,outp,ledout);

input   [7:0] slide_switch ;// Input Connected to slide switches
output  [13:0] outp        ;// Output Connected to Seaven Segment
output  [7:0] ledout       ;// Output for LED

wire     [13:0] outp        ;


assign ledout = slide_switch ; // assignment of LED to slide switch


function [6:0] decode;

input [3:0] ss_temp;
begin
```

```
  case(ss_temp)
       4'b0000: decode = 7'b1000000    ; // Start Condition "00" No Switch is ON
                4'b0001: decode = 7'b1111001    ; // Switch 1 is ON
                4'b0010: decode = 7'b0100100    ; // Switch 2 is ON
                4'b0011: decode = 7'b0110000    ; // Switch 3 is ON
                4'b0100: decode = 7'b0011001    ; // Switch 4 is ON
                4'b0101: decode = 7'b0010010    ; // Switch 5 is ON
                4'b0110: decode = 7'b0000010    ; // Switch 6 is ON
                4'b0111: decode = 7'b1111000    ; // Switch 7 is ON
                4'b1000: decode = 7'b0000000    ; // Switch 8 is ON
                4'b1001: decode = 7'b0010000    ; // Switch 9 is ON
                4'b1010: decode = 7'b0001000    ; // Switch A is ON
                4'b1011: decode = 7'b0000011    ; // Switch B is ON
                4'b1100: decode = 7'b1000110    ; // Switch C is ON
                4'b1101: decode = 7'b0100001    ; // Switch D is ON
                4'b1110: decode = 7'b0000110    ; // Switch E is ON
                4'b1111: decode = 7'b0001110    ; // Switch F is ON
            default : decode = 7'b0101111    ; // Error Condition "r"
endcase
end
endfunction

assign outp = {decode(slide_switch[7:4]),decode(slide_switch[3:0])};

endmodule
```

Ctrl.v

```
//Description : Control State machine which will latch
//Value of 7-Seg Display Until the Next Switch is pressed.
module ctrl(clk,rst,but0,digit0,digit1,dp,an,sseg);

input       clk    ; // Connected with Systme Clock
input       rst    ; // Connected with Button 1
input       but0   ; // Connected with Button 0
input [6:0] digit0 ; // Connected with Slide switch Digit0
input [6:0] digit1 ; // Connected with Slide switch Digit1

output       dp    ; // Decimal Point Output which is Hardcoded to 1
output [3:0] an    ; // AN ( 4 Anode Signals for 7-seg display)
output [6:0] sseg  ; // 7-Segment Display Output

localparam N = 18  ; // to Control Refreshing rate count

wire         dp    ;
reg [3:0 ] an      ;
reg [6:0 ] sseg    ;
reg [6:0 ] sseg3, sseg2, sseg1, sseg0; // All four Display Values which lated on
used by mux
reg [N-1:0]count   ;
reg        temp    ;

assign dp     = 1'b1 ; // Hard Code Decimal Point to 1 ( Dot will not be Enabled
)

// Assign Digits to 7-Segments whenever but0 is pressed
always@(posedge clk or posedge rst) begin
  if(rst) begin
     sseg3 = 7'b1111111;
     sseg2 = 7'b1111111;
     sseg1 = 7'b1111111;
     sseg0 = 7'b1111111;
  end else if(but0) begin
     sseg3 = 7'b1111111;
     sseg2 = 7'b1111111;
     sseg1 = digit1; // Slide Switch Values
```

```
     sseg0 = digit0; // Slide Switch Values
  end
end

always @ (posedge clk or posedge rst) begin
  if (rst)
   count <= 1'b0     ; // Count Numbers
  else
   count <= count + 1; // Count Numbers
end

//Generic Mux for 7-Seg Display
//Where only Segment 0 and Segment 1 is Enabled

always @ (*) begin

  case(count[N-1:N-2]) // Count(17-16)

   2'b00 :  begin
                temp = 1'b0;
            sseg = sseg0  ;// seg0
            an   = 4'b1110;// an<0>
            end
   2'b01:   begin
                temp = 1'b0  ;
            sseg = sseg1  ;// seg1
            an = 4'b1101  ;// san<1>
            end
   2'b10:   begin
                temp = 1'b0  ;
            sseg = sseg2  ;// Parmanently OFF
            an   = 4'b1111;// Parmanently Disabled
            end
   2'b11:   begin
                temp = 1'b1   ;
            sseg = sseg3  ;// Parmanently OFF
            an   = 4'b1111;// Parmanently Disabled
            end
  endcase
 end
endmodule
```

tb.v

```
`timescale 1ps/1ps
module tb;

reg clk1;
reg rst ;

reg but0,but1;
reg [7:0] ssw;

wire [6:0] sseg;
```

```
wire       dp;
wire [7:0] led ;
wire [3:0] an  ;

always begin
clk1 = 0;
forever #1 clk1 = ~clk1;
end

reg       flag=0;

reg [3:0] sseg2hex_temp1;
reg [3:0] sseg2hex_temp2;
reg [7:0] sseg2hex3=7'b0;

reg [17:0]count   ;

always @ (posedge clk1 or posedge rst) begin
  if (rst)
   count <= 1'b0     ; // Count Numbers
  else
   count <= count + 1; // Count Numbers
end

task initialize;
begin
rst  = 1'b1;
but0 = 1'h0;
but1 = 1'b0;
ssw = 7'b0;
#10 rst =1'b0;
end
endtask



task switch_set;
input [7:0] slide_switch1;
begin
ssw=slide_switch1;
repeat(2)@(posedge clk1);
but0=1'b1;
repeat(5)@(posedge clk1);
but0=1'b0;
wait(pro.ctrl0.temp);
wait(~pro.ctrl0.temp);
end
endtask

always@(posedge count[15])begin

if(count[17:16]==00) begin
sseg2hex_temp1=sseg2hex(sseg);
end else if (count[17:16]==01) begin
sseg2hex_temp2=sseg2hex(sseg);
sseg2hex3={sseg2hex_temp2,sseg2hex_temp1};
if(sseg2hex3==ssw)
$display("At Time : %0t  Result : <PASS> Actual Data : %h Expected Data : %h"
,$time,sseg2hex3,ssw);
else
```

```
$display("At Time : %0t  Result : <FAIL> Actual Data : %h Expected Data : %h"
,$time,sseg2hex3,ssw);
end
end

project1 pro ( .rst  (rst)
              ,.clk  (clk1)
              ,.but0 (but0)
              ,.ssw  (ssw)
              ,.sseg (sseg)
              ,.an   (an)
              ,.dp   (dp)
              ,.led  (led));

initial begin
$display("--Tests Start--");
initialize;
$display("--Basic Tests--");
switch_set(8'h01);
switch_set(8'h02);
switch_set(8'h03);
switch_set(8'h04);
switch_set(8'h05);
switch_set(8'h06);
switch_set(8'h07);
switch_set(8'h08);
switch_set(8'h09);
switch_set(8'h0A);
switch_set(8'h0B);
switch_set(8'h0C);
switch_set(8'h0D);
switch_set(8'h0E);
switch_set(8'h0F);
switch_set(8'h10);
switch_set(8'h11);
switch_set(8'h12);
switch_set(8'h13);
switch_set(8'h14);
switch_set(8'h15);
switch_set(8'h16);
switch_set(8'h17);
switch_set(8'h18);
switch_set(8'h19);
switch_set(8'h1A);
switch_set(8'h1B);
switch_set(8'h1C);
switch_set(8'h1D);
switch_set(8'h1E);
switch_set(8'h1F);
switch_set(8'h20);
switch_set(8'h21);
switch_set(8'h22);
switch_set(8'h23);
switch_set(8'h24);
switch_set(8'h25);
switch_set(8'h26);
switch_set(8'h27);
switch_set(8'h28);
switch_set(8'h29);
switch_set(8'h2A);
switch_set(8'h2B);
switch_set(8'h2C);
switch_set(8'h2D);
switch_set(8'h2E);
switch_set(8'h2F);
switch_set(8'h30);
switch_set(8'h31);
switch_set(8'h32);
switch_set(8'h33);
switch_set(8'h34);
switch_set(8'h35);
switch_set(8'h36);
```

```
switch_set(8'h37);
switch_set(8'h38);
switch_set(8'h39);
switch_set(8'h3A);
switch_set(8'h3B);
switch_set(8'h3C);
switch_set(8'h3D);
switch_set(8'h3E);
switch_set(8'h3F);
switch_set(8'h40);
switch_set(8'h41);
switch_set(8'h42);
switch_set(8'h43);
switch_set(8'h44);
switch_set(8'h45);
switch_set(8'h46);
switch_set(8'h47);
switch_set(8'h48);
switch_set(8'h49);
switch_set(8'h4A);
switch_set(8'h4B);
switch_set(8'h4C);
switch_set(8'h4D);
switch_set(8'h4E);
switch_set(8'h4F);
switch_set(8'h50);
switch_set(8'h51);
switch_set(8'h52);
switch_set(8'h53);
switch_set(8'h54);
switch_set(8'h55);
switch_set(8'h56);
switch_set(8'h57);
switch_set(8'h58);
switch_set(8'h59);
switch_set(8'h5A);
switch_set(8'h5B);
switch_set(8'h5C);
switch_set(8'h5D);
switch_set(8'h5E);
switch_set(8'h5F);
switch_set(8'h60);
switch_set(8'h61);
switch_set(8'h62);
switch_set(8'h63);
switch_set(8'h64);
switch_set(8'h65);
switch_set(8'h66);
switch_set(8'h67);
switch_set(8'h68);
switch_set(8'h69);
switch_set(8'h6A);
switch_set(8'h6B);
switch_set(8'h6C);
switch_set(8'h6D);
switch_set(8'h6E);
switch_set(8'h6F);
switch_set(8'h70);
switch_set(8'h71);
switch_set(8'h72);
switch_set(8'h73);
switch_set(8'h74);
switch_set(8'h75);
switch_set(8'h76);
switch_set(8'h77);
switch_set(8'h78);
switch_set(8'h79);
switch_set(8'h7A);
switch_set(8'h7B);
switch_set(8'h7C);
switch_set(8'h7D);
switch_set(8'h7E);
switch_set(8'h7F);
switch_set(8'h80);
switch_set(8'h81);
switch_set(8'h82);
```

```
switch_set(8'h83);
switch_set(8'h84);
switch_set(8'h85);
switch_set(8'h86);
switch_set(8'h87);
switch_set(8'h88);
switch_set(8'h89);
switch_set(8'h8A);
switch_set(8'h8B);
switch_set(8'h8C);
switch_set(8'h8D);
switch_set(8'h8E);
switch_set(8'h8F);
switch_set(8'h90);
switch_set(8'h91);
switch_set(8'h92);
switch_set(8'h93);
switch_set(8'h94);
switch_set(8'h95);
switch_set(8'h96);
switch_set(8'h97);
switch_set(8'h98);
switch_set(8'h99);
switch_set(8'h9A);
switch_set(8'h9B);
switch_set(8'h9C);
switch_set(8'h9D);
switch_set(8'h9E);
switch_set(8'h9F);
switch_set(8'hA0);
switch_set(8'hA1);
switch_set(8'hA2);
switch_set(8'hA3);
switch_set(8'hA4);
switch_set(8'hA5);
switch_set(8'hA6);
switch_set(8'hA7);
switch_set(8'hA8);
switch_set(8'hA9);
switch_set(8'hAA);
switch_set(8'hAB);
switch_set(8'hAC);
switch_set(8'hAD);
switch_set(8'hAE);
switch_set(8'hAF);
switch_set(8'hB0);
switch_set(8'hB1);
switch_set(8'hB2);
switch_set(8'hB3);
switch_set(8'hB4);
switch_set(8'hB5);
switch_set(8'hB6);
switch_set(8'hB7);
switch_set(8'hB8);
switch_set(8'hB9);
switch_set(8'hBA);
switch_set(8'hBB);
switch_set(8'hBC);
switch_set(8'hBD);
switch_set(8'hBE);
switch_set(8'hBF);
switch_set(8'hC0);
switch_set(8'hC1);
switch_set(8'hC2);
switch_set(8'hC3);
switch_set(8'hC4);
switch_set(8'hC5);
switch_set(8'hC6);
switch_set(8'hC7);
switch_set(8'hC8);
switch_set(8'hC9);
switch_set(8'hCA);
switch_set(8'hCB);
switch_set(8'hCC);
switch_set(8'hCD);
switch_set(8'hCE);
```

```
switch_set(8'hCF);
switch_set(8'hD0);
switch_set(8'hD1);
switch_set(8'hD2);
switch_set(8'hD3);
switch_set(8'hD4);
switch_set(8'hD5);
switch_set(8'hD6);
switch_set(8'hD7);
switch_set(8'hD8);
switch_set(8'hD9);
switch_set(8'hDA);
switch_set(8'hDB);
switch_set(8'hDC);
switch_set(8'hDD);
switch_set(8'hDE);
switch_set(8'hDF);
switch_set(8'hE0);
switch_set(8'hE1);
switch_set(8'hE2);
switch_set(8'hE3);
switch_set(8'hE4);
switch_set(8'hE5);
switch_set(8'hE6);
switch_set(8'hE7);
switch_set(8'hE8);
switch_set(8'hE9);
switch_set(8'hEA);
switch_set(8'hEB);
switch_set(8'hEC);
switch_set(8'hED);
switch_set(8'hEE);
switch_set(8'hEF);
switch_set(8'hF0);
switch_set(8'hF1);
switch_set(8'hF2);
switch_set(8'hF3);
switch_set(8'hF4);
switch_set(8'hF5);
switch_set(8'hF6);
switch_set(8'hF7);
switch_set(8'hF8);
switch_set(8'hF9);
switch_set(8'hFA);
switch_set(8'hFB);
switch_set(8'hFC);
switch_set(8'hFD);
switch_set(8'hFE);
switch_set(8'hFF);
$finish;
end


function [3:0] sseg2hex;

input [6:0] sseg;
begin
case(sseg)
   7'b1000000    : sseg2hex = 4'b0000 ;   //0
   7'b1111001    : sseg2hex = 4'b0001 ;   //1
   7'b0100100    : sseg2hex = 4'b0010 ;   //2
   7'b0110000    : sseg2hex = 4'b0011 ;   //3
   7'b0011001    : sseg2hex = 4'b0100 ;   //4
   7'b0010010    : sseg2hex = 4'b0101 ;   //5
   7'b0000010    : sseg2hex = 4'b0110 ;   //6
   7'b1111000    : sseg2hex = 4'b0111 ;   //7
   7'b0000000    : sseg2hex = 4'b1000 ;   //8
   7'b0010000    : sseg2hex = 4'b1001 ;   //9
   7'b0001000    : sseg2hex = 4'b1010 ;   //A
   7'b0000011    : sseg2hex = 4'b1011 ;   //b
```

```
    7'b1000110    : sseg2hex = 4'b1100 ;    //C
    7'b0100001    : sseg2hex = 4'b1101 ;    //d
    7'b0000110    : sseg2hex = 4'b1110 ;    //E
    7'b0001110    : sseg2hex = 4'b1111 ;    //F
    7'b0101111    : sseg2hex = 4'b1111 ;
endcase
end
endfunction
endmodule
```

Appendix B (Tools /Refrences)

- www.xilinx.com
- www.digilent.com

Appendix C (UCF File)

```
# clock pin for Basys2 Board
NET "clk" LOC = "B8"; # Bank = 0, Signal name = clk

# Pin assignment for DispCtl
# Connected to Basys2 onBoard 7seg display
NET "sseg<0>" LOC = "L14"; # Bank = 1, Signal name = CA
NET "sseg<1>" LOC = "H12"; # Bank = 1, Signal name = CB
NET "sseg<2>" LOC = "N14"; # Bank = 1, Signal name = CC
NET "sseg<3>" LOC = "N11"; # Bank = 2, Signal name = CD
NET "sseg<4>" LOC = "P12"; # Bank = 2, Signal name = CE
NET "sseg<5>" LOC = "L13"; # Bank = 1, Signal name = CF
NET "sseg<6>" LOC = "M12"; # Bank = 1, Signal name = CG
NET "dp"      LOC = "N13"; # Bank = 1, Signal name = DP

NET "an<3>" LOC = "K14"; # Bank = 1, Signal name = AN3
NET "an<2>" LOC = "M13"; # Bank = 1, Signal name = AN2
NET "an<1>" LOC = "J12"; # Bank = 1, Signal name = an1
NET "an<0>" LOC = "F12"; # Bank = 1, Signal name = an0

# Pin assignment for LEDs
NET "led<7>" LOC = "G1" ; # Bank = 3, Signal name = LD7
NET "led<6>" LOC = "P4" ; # Bank = 2, Signal name = LD6
NET "led<5>" LOC = "N4" ;  # Bank = 2, Signal name = LD5
NET "led<4>" LOC = "N5" ;  # Bank = 2, Signal name = LD4
NET "led<3>" LOC = "P6" ; # Bank = 2, Signal name = LD3
NET "led<2>" LOC = "P7" ; # Bank = 3, Signal name = LD2
NET "led<1>" LOC = "M11" ; # Bank = 2, Signal name = LD1
NET "led<0>" LOC = "M5" ;  # Bank = 2, Signal name = LD0

# Pin assignment for Slide Switches(SWs)
NET "ssw<7>" LOC = "N3";  # Bank = 2, Signal name = SW7
NET "ssw<6>" LOC = "E2";  # Bank = 3, Signal name = SW6
NET "ssw<5>" LOC = "F3";  # Bank = 3, Signal name = SW5
NET "ssw<4>" LOC = "G3";  # Bank = 3, Signal name = SW4
NET "ssw<3>" LOC = "B4";  # Bank = 3, Signal name = SW3
NET "ssw<2>" LOC = "K3";  # Bank = 3, Signal name = SW2
NET "ssw<1>" LOC = "L3";  # Bank = 3, Signal name = SW1
NET "ssw<0>" LOC = "P11";  # Bank = 2, Signal name = SW0

# Pin assignment for Buttons 0/1
NET "rst"  LOC = "C11"; # Bank = 2, Signal name = BTN1
NET "but0" LOC = "G12"; # Bank = 0, Signal name = BTN0
```