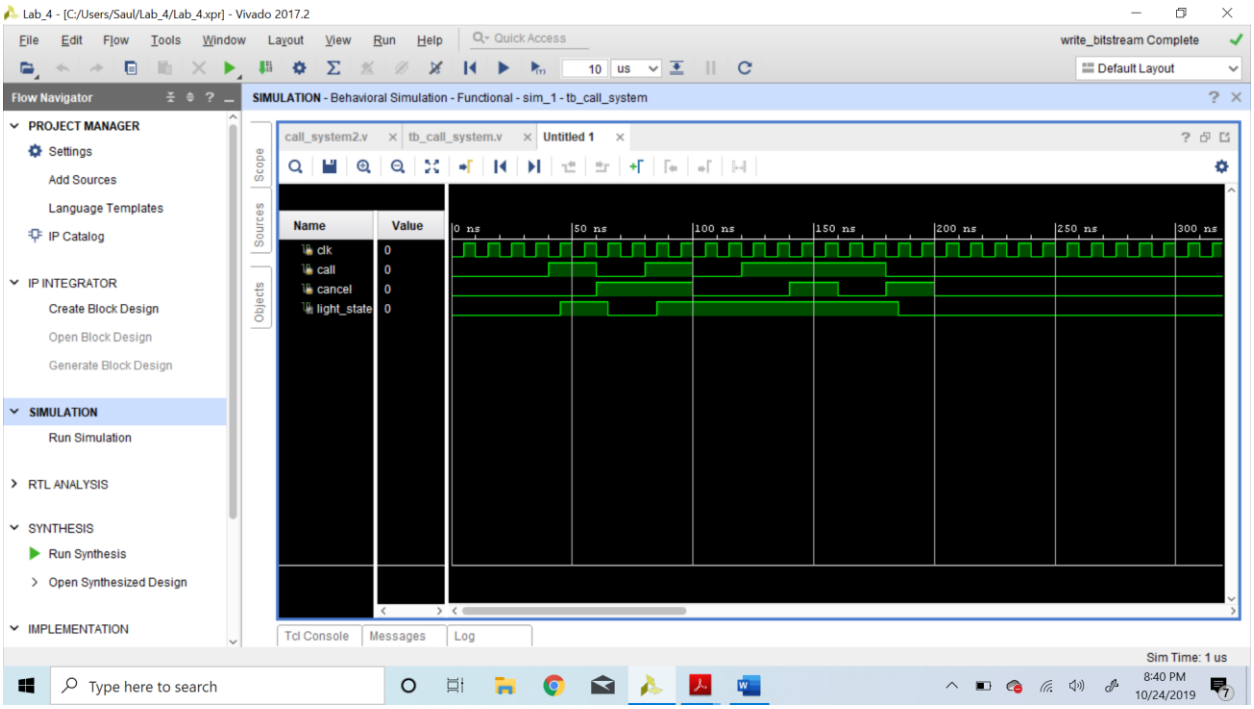
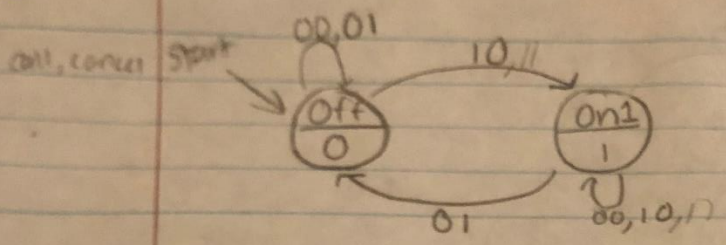
Part A




```
`timescale 1ns / 1ps
module call_system2(
    input clk,
    input call,
    input cancel,
    output reg light_state  //Current state
    );
    wire next_state; //Holds next state
    initial light_state = 0; //Initializes output/current
    assign next_state = call | (~call & ~cancel & light_state);
    always @(posedge clk) begin //Sensitivity list triggers on rising edge of clk
        light_state <= next_state; //Next state gets clocked in on clk
    end
endmodule
```



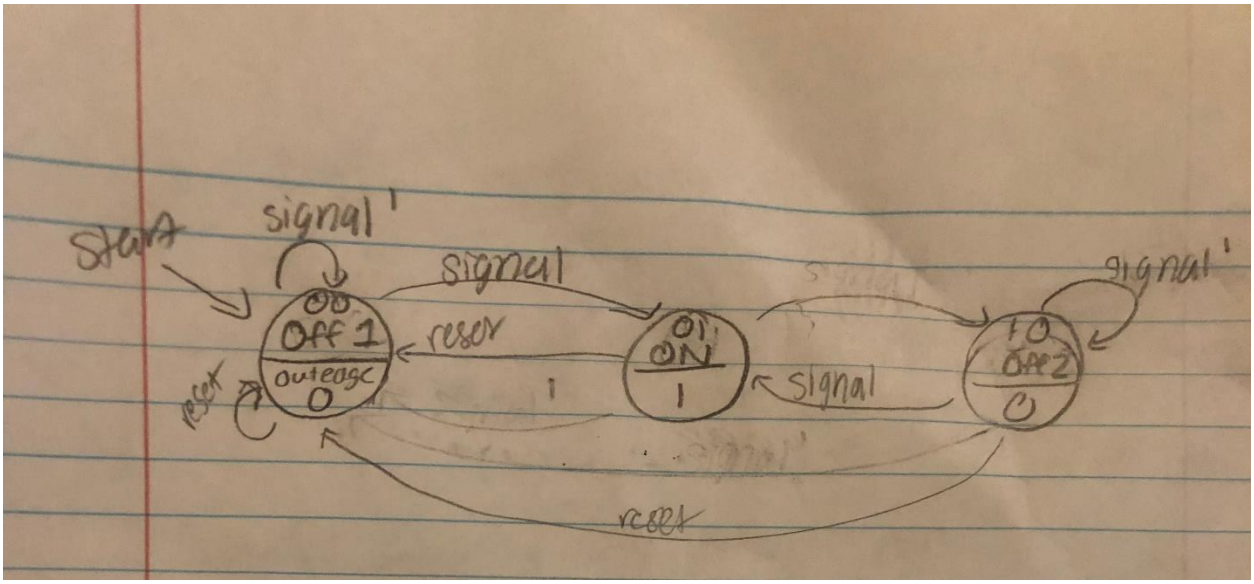
Current	input	next state	output
0	00	0	0
1	00	1	1
0	01	0	0
1	01	0	0
0	10	1	1
1	10	1	1
0	11	1	1
1	11	1	1
	S_2 S_1		S_0

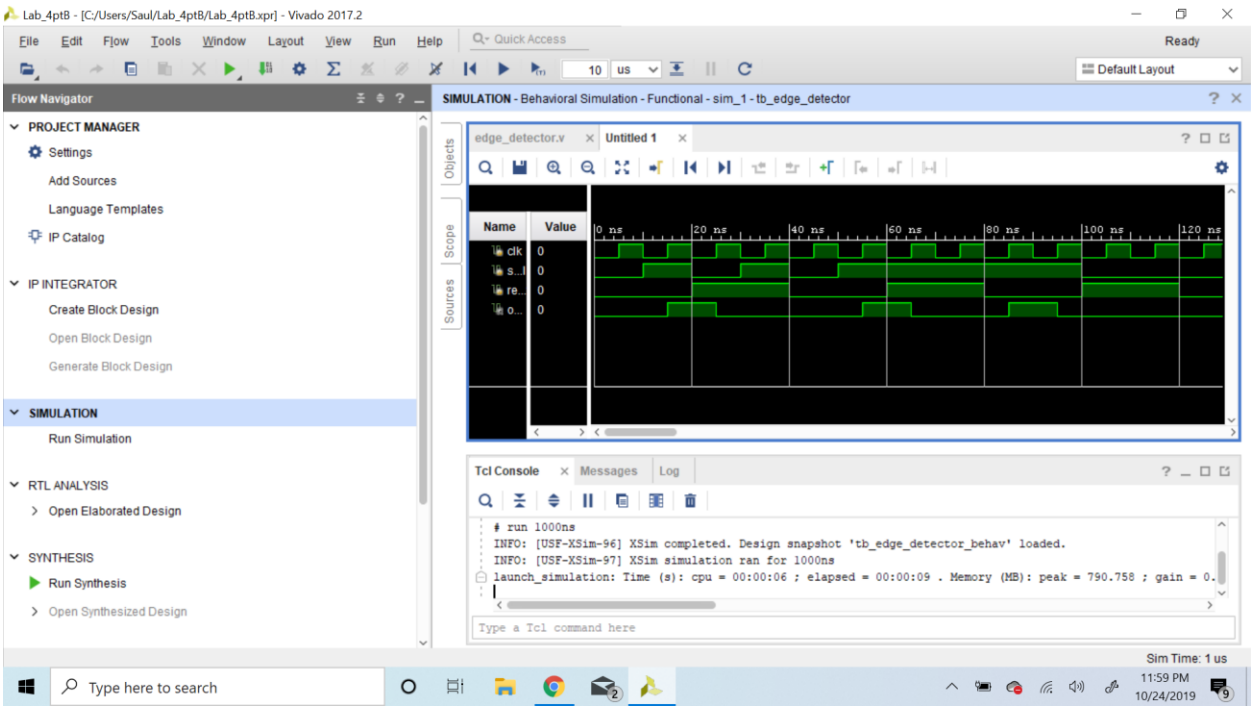
S_2 \ S_1S_0	00	01	11	10
0	0	1	0	0
1	1	1	1	1

S_0 = light
S_1 = cancel
S_2 = call

$next\_state = S_2 + \overline{S_1} S_0$

Part B


start
signal'
signal
reset
signal
signal'
reset
reset


Lab_4ptB - [C:/Users/Saul/Lab_4ptB/Lab_4ptB.xpr] - Vivado 2017.2
File Edit Flow Tools Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_edge_detector
edge_detector.v
Untitled 1
Name
Value
clk
0 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
Tcl Console
Messages
Log
run 1000ns
INFO: [USF-XSim-96] XSim completed. Design snapshot 'tb_edge_detector_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:06 ; elapsed = 00:00:09 . Memory (MB): peak = 790.758 ; gain = 0.
Type a Tcl command here
Sim Time: 1 us
Type here to search
11:59 PM
10/24/2019

Edge_detector

```
`timescale 1ns / 1ps

module edge_detector(
   input clk,
   input signal,
   input reset,
   output reg outedge
   );
   wire slow_clk;
   reg [1:0] state;
   reg [1:0] next;



   clkdiv c1(clk, reset, slow_clk);

   always @(*) begin
      case(state)
         2'b00 : begin
            outedge = 1'b0;
            if (~signal | reset) next = 2'b00;
            else next = 2'b01;
            end

         2'b01 : begin
            outedge = 1'b1;
```

```
            if (reset) next = 2'b00;
            if (~signal) next = 2'b10;
            else next = 2'b10;
            end

         2'b10 : begin
            outedge = 1'b0;
            if (reset) next = 2'b00;
            if(~signal) next = 2'b10;
            else next = 2'b01;
            end

         default : begin
         next = 2'b00;
         outedge = 1'b0;
         end
      endcase
   end

   always @(posedge slow_clk) begin
      if(reset) state <= 2'b00;
      else state <= next;
   end
endmodule
```

Testbench

```
`timescale 1ns / 1ps
```

```
module tb_edge_detector;

    reg clk;
    reg signal;
    reg reset;

    wire outedge;

    edge_detector uut(
    .clk(clk),
    .signal(signal),
    .reset(reset),
    .outedge(outedge)
    );

    initial begin

        clk = 0;
        signal = 0;
        reset = 0;

        #10;

        signal = 1;
        reset = 0;

        #10;

        signal = 0;
```

```
reset = 1;

#10;

signal = 1;
reset = 1;

#10;

signal = 0;
reset = 0;

#10;

signal = 1;
reset = 0;

#10

reset = 1;

#20;

reset = 0;

#20;

signal = 0;
reset = 1;
```

```
    #20;

    signal = 0;
    reset = 0;

  end
  always
    #5 clk = ~clk;
Endmodule
```

Clock Division

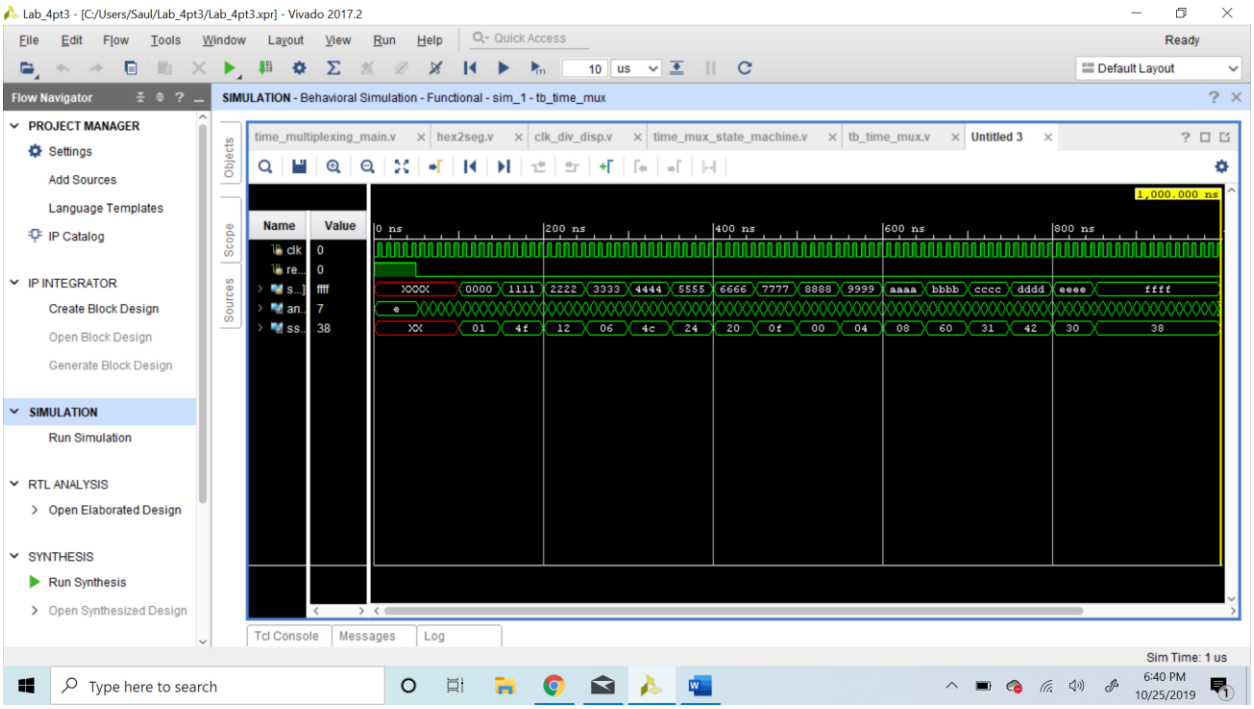
```
`timescale 1ns / 1ps

module clkdiv(
  input clk,
  input reset,
  output clk_out
  );

  reg [26:0] count = 0;
    //clk_out = count[15];
    assign clk_out = count[24];

    always @(posedge clk)
    begin
    if (reset)count = 0;
    else count = count + 1;
```

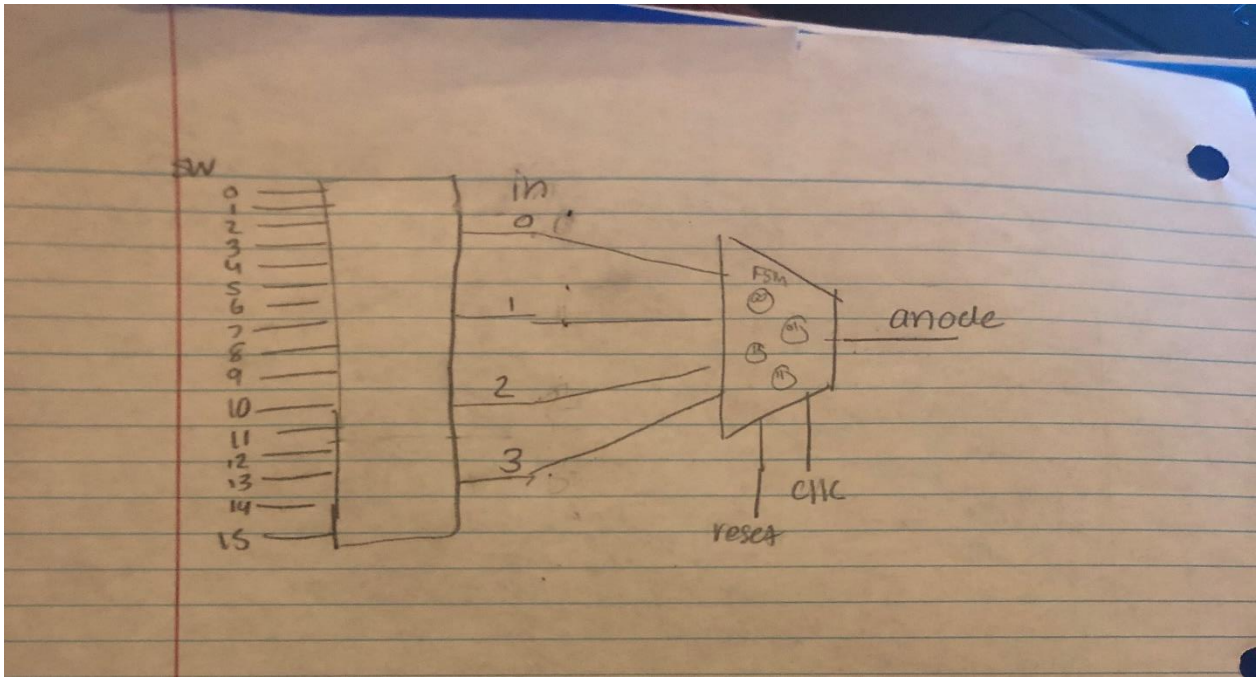
```
    end

endmodule
```

Constraints

```
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} [get_ports {clk}]

set_property PACKAGE_PIN V17 [get_ports {signal}]
set_property IOSTANDARD LVCMOS33 [get_ports {signal}]

set_property PACKAGE_PIN U18 [get_ports reset]
set_property IOSTANDARD LVCMOS33 [get_ports reset]


set_property PACKAGE_PIN U16 [get_ports {outedge}]
set_property IOSTANDARD LVCMOS33 [get_ports {outedge}]
```

Part C


Lab_4pt3 - [C:/Users/Saul/Lab_4pt3/Lab_4pt3.xpr] - Vivado 2017.2
File Edit Flow Tools Window Layout View Run Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
SIMULATION - Behavioral Simulation - Functional - sim_1 - tb_time_mux
time_multiplexing_main.v
hex2seg.v
clk_div_disp.v
time_mux_state_machine.v
tb_time_mux.v
Untitled 3
Name
Value
clk
0
re...
0
s...]
ffff
an...
7
ss...
38
0 ns
200 ns
400 ns
600 ns
800 ns
1,000.000 ns
0000 1111 2222 3333 4444 5555 6666 7777 8888 9999 aaaa bbbb cccc dddd eeee ffff
01 4f 12 06 4c 24 20 0f 00 04 08 60 31 42 30 38
Tcl Console
Messages
Log
Sim Time: 1 us
Type here to search
6:40 PM
10/25/2019


SW
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
in
0
1
2
3
FSM
anode
clk
reset

div 2 =	count [0]	= 50 MHz
div 4	[1]	= 25 MHz
8	[2]	12.5
16	[3]	6.25
32	[4]	3.125
64	[5]	1.562S
128	[6]	781.250 kHz
256	[7]	390,625 kHz
512	[8]	195.3125 kHz
1024	[9]	97,656 Hz
2048	[10]	48,828
4096	[11]	24,414
8192	[12]	12.207
16,384	[13]	6,103
32,768	[14]	3.051
65,536	[15]	1.525
131,072	[16]	762 Hz
262144	[17]	381.
524,288	[18]	190.
1,048,576	[19]	95.367 Hz
2,097,152	[20]	47
4,194,304	[21]	23
8,388,608	[22]	11
16,777,216	[23]	5.96 Hz
33,554,432	[24]	2.98 Hz

Main

```
`timescale 1ns / 1ps

module time_multiplexing_main(
    input clk,
    input reset,
    input [15:0] sw,
    output [3:0] an,
    output [6:0] sseg
    );

    wire [6:0] in0, in1, in2, in3;
    wire slow_clk;

    hex2seg c1 (.x(sw[3:0]), .r(in0));
    hex2seg c2 (.x(sw[7:4]), .r(in1));
    hex2seg c3 (.x(sw[11:8]), .r(in2));
    hex2seg c4 (.x(sw[15:12]), .r(in3));

    clk_div_disp c5(.clk(clk), .reset(reset), .clk_out(slow_clk));

    time_mux_state_machine c6(
    .clk (slow_clk),
    .reset (reset),
    .in0 (in0),
    .in1 (in1),
    .in2 (in2),
    .in3 (in3),
```

```
   .an (an),
   .sseg (sseg)
   );
Endmodule
```

Hex2seg

```
`timescale 1ns / 1ps

module hex2seg(
   input [3:0] x,
   output reg [6:0] r
   );

   always @(*)
      case(x)
         4'b0000: r = 7'b0000001;
         4'b0001: r = 7'b1001111;
         4'b0010: r = 7'b0010010;
         4'b0011: r = 7'b0000110;
         4'b0100: r = 7'b1001100;
         4'b0101: r = 7'b0100100;
         4'b0110: r = 7'b0100000;
         4'b0111: r = 7'b0001111;
         4'b1000: r = 7'b0000000;
         4'b1001: r = 7'b0000100;
         4'b1010: r = 7'b0001000;   //A
         4'b1011: r = 7'b1100000;
         4'b1100: r = 7'b0110001;
         4'b1101: r = 7'b1000010;
```

```
        4'b1110: r = 7'b0110000;
        4'b1111: r = 7'b0111000;
    endcase

endmodule
```

Mux

```
`timescale 1ns / 1ps

module time_mux_state_machine(
    input clk,
    input reset,
    input [6:0] in0,
    input [6:0] in1,
    input [6:0] in2,
    input [6:0] in3,
    output reg [3:0] an,
    output reg [6:0] sseg
    );

    reg [1:0] state;
    reg [1:0] next_state;

    always @ (*) begin
        case(state)
            2'b00: next_state = 2'b01;
            2'b01: next_state = 2'b10;
            2'b10: next_state = 2'b11;
            2'b11: next_state = 2'b00;
```

```
    endcase
  end

  always @(*) begin
    case (state)
      2'b00 : sseg = in0;
      2'b01 : sseg = in1;
      2'b10 : sseg = in2;
      2'b11 : sseg = in3;
    endcase
  //end
  //always @ (*) begin
    case (state)
      2'b00 : an = 4'b1110;
      2'b01 : an = 4'b1101;
      2'b10 : an = 4'b1011;
      2'b11 : an = 4'b0111;
    endcase
  end

  always @(posedge clk or posedge reset) begin
    if(reset)
      state <= 2'b00;
    else
      state <= next_state;
  end
endmodule
```

Clock

```
`timescale 1ns / 1ps

module clk_div_disp(
   input clk,
   input reset,
   output clk_out
   );

   reg [26:0] COUNT;

      assign clk_out = COUNT[26];

   always @(posedge clk)
   begin
      if (reset)
         COUNT = 0;
      else
         COUNT = COUNT + 1;
      end
endmodule
```

Testbench

```
`timescale 1ns / 1ps

module tb_time_mux;

reg clk;
```

```
reg reset;
reg [15:0] sw;
wire [3:0] an;
wire [6:0] sseg;



time_multiplexing_main uut(
    .clk(clk),
    .reset(reset),
    .sw(sw),
    .an(an),
    .sseg(sseg)
    );

initial
    begin

    clk = 0;
    reset = 1;

    #50;

    reset = 0;

    #50;
    ////

    sw = 16'h0000;
```

```
#50

sw = 16'h1111;

#50

sw = 16'h2222;

#50

sw = 16'h3333;

#50

sw = 16'h4444;

#50

sw = 16'h5555;

#50

sw = 16'h6666;

#50

sw = 16'h7777;

#50
```

```
sw = 16'h8888;

#50

sw = 16'h9999;

#50

sw = 16'haaaa;

#50

sw = 16'hbbbb;

#50

sw = 16'hcccc;

#50

sw = 16'hdddd;

#50

sw = 16'heeee;

#50
```

```
    sw = 16'hffff;

    end

    always
    #5 clk = ~clk;

    Endmodule
```

Constraints

```
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} [get_ports {clk}]

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]}]


set_property PACKAGE_PIN W7 [get_ports {sseg[6]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[6]}]
set_property PACKAGE_PIN W6 [get_ports {sseg[5]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[5]}]
set_property PACKAGE_PIN U8 [get_ports {sseg[4]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[4]}]
set_property PACKAGE_PIN V8 [get_ports {sseg[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[3]}]
set_property PACKAGE_PIN U5 [get_ports {sseg[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[2]}]
```

```
set_property PACKAGE_PIN V5 [get_ports {sseg[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[1]}]
set_property PACKAGE_PIN U7 [get_ports {sseg[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {sseg[0]}]

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]}]

set_property PACKAGE_PIN U18 [get_ports {reset}]
   set_property IOSTANDARD LVCMOS33 [get_ports {reset}]
```