**HW6**

**-Daksh Thapar 2018137**

**-Tanmaya Gupta 2018200**

Verilog Codes

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 24.09.2019 19:59:42

// Design Name:

// Module Name: top

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module top(

input clk\_100M,

inout [7:0] JA,

output [3:0] anode,

output [7:0] cathode,

input enter

);

wire clk\_5ms, clk\_1K, clk\_7M;

wire [3:0] decode;

clk\_wiz\_0 c11(.clk\_in1(clk\_100M),.clk\_out1(clk\_7M));

clk\_div #(13) u1( .clk\_in(clk\_7M) , .clk\_out(clk\_1K));

clk\_div #(15) u2(.clk\_in(clk\_7M),.clk\_out(clk\_5ms));

Decoder s11(.clk(clk\_100M),.Row(JA[7:4]),.Col(JA[3:0]),.DecodeOut(decode),.clking(bouncer),.enter(enter));

wire temp\_bouncer;

debouncer d1(.clk\_in(clk\_5ms),.clr\_o(temp\_bouncer),.clr\_i(bouncer));

wire [7:0] number;

wire [3:0] ones,tens,hundreds,thousands;

fsm\_moore\_ccd moore(.clk(temp\_bouncer),.din(decode),.present\_state(number),.enter(enter));

reg [3:0] first,second,third,fourth;

always @( \* )

if ( number == 8'd6)

first <= 4'd1;

else if ( number == 8'd4)

first<= 4'd0;

else if (number == 8'd5 )

first <= 4'd7;

else

first <= 4'd0;

always @( \* )

if ( number == 8'd6)

second <= 4'd2;

else if ( number == 8'd4)

second<= 4'd4;

else if (number == 8'd5 )

second <= 4'd2;

else

second <= 4'd0;

always @( \* )

if ( number == 8'd6)

third <= 4'd3;

else if ( number == 8'd4)

third<= 4'd5;

else if (number == 8'd5 )

third <= 4'd4;

else

third <= 4'd0;

always @( \* )

if ( number == 8'd6)

fourth <= 4'd3;

else if ( number == 8'd4)

fourth<= 4'd6;

else if (number == 8'd5 )

fourth <= 4'd8;

else

fourth <= 4'd0;

sevenSeg s1(.clk(clk\_1K),.ones(fourth),.tens(third),.thousands(first),.hundreds(second),.anode(anode),.cathode(cathode));

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 24.09.2019 20:02:17

// Design Name:

// Module Name: Decoder

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module Decoder(

input clk,

input [3:0] Row,

output reg [3:0] Col,

output clking,

output reg [3:0] DecodeOut,

input enter

);

reg [3:0] temp;

reg [19:0] sclk = 0 ;

always @ (posedge clk)

begin

if (sclk == 20'b00011000011010100000)

begin

Col <= 4'b0111;

sclk <= sclk + 1'b1;

temp[0] <= 1'b1;

end

else if(sclk ==20'b00011000011010101000)

begin

if (Row==4'b0111) begin

DecodeOut <= 4'b0001;

end

else if (Row == 4'b1011) begin

DecodeOut <= 4'b0100;

end

else if (Row == 4'b1101) begin

DecodeOut <= 4'b0111;

end

else if (Row == 4'b1110) begin

DecodeOut <= 4'b0000;

end

else begin

temp[0] = 1'b0;

end

sclk <= sclk + 1'b1;

end

else if (sclk == 20'b00110000110101000000) begin

Col <= 4'b1011;

sclk <= sclk +1'b1;

temp[1]<= 1'b1;

end

else if (sclk == 20'b00110000110101001000) begin

if (Row==4'b0111) begin

DecodeOut <= 4'b0010;

end

else if (Row == 4'b1011) begin

DecodeOut <= 4'b0101;

end

else if (Row == 4'b1101) begin

DecodeOut <= 4'b1000;

end

else if (Row == 4'b1110) begin

DecodeOut <= 4'b1111;

end

else

begin

temp[1] <= 1'b0;

end

sclk <= sclk + 1'b1;

end

else if (sclk == 20'b01001001001111100000) begin

Col <= 4'b1101;

sclk <= sclk +1'b1;

temp[2]<= 1'b1;

end

else if (sclk == 20'b01001001001111101000) begin

if (Row==4'b0111) begin

DecodeOut <= 4'b0011;

end

else if (Row == 4'b1011) begin

DecodeOut <= 4'b0110;

end

else if (Row == 4'b1101) begin

DecodeOut <= 4'b1001;

end

else if (Row == 4'b1110) begin

DecodeOut <= 4'b1110;

end

else

begin

temp[2]<=1'b0;

end

sclk <= sclk + 1'b1;

end

else if (sclk == 20'b01100001101010000000) begin

Col <= 4'b1110;

sclk <= sclk +1'b1;

temp[3]<= 1'b1;

end

else if (sclk == 20'b01100001101010001000) begin

if (Row==4'b0111) begin

DecodeOut <= 4'b1010;

end

else if (Row == 4'b1011) begin

DecodeOut <= 4'b1011;

end

else if (Row == 4'b1101) begin

DecodeOut <= 4'b1100;

end

else if (Row == 4'b1110) begin

DecodeOut <= 4'b1101;

end

else

begin

temp[3] <= 1'b0;

end

sclk <= 20'b00000000000000000000;

end

else begin

sclk <= sclk + 1'b1;

end

end

assign clking = temp[0]|temp[1]|temp[2]|temp[3]|enter;

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 24.09.2019 20:04:50

// Design Name:

// Module Name: debouncer

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module debouncer(

input clk\_in,

input clr\_i,

output clr\_o

);

reg d1, d2, d3;

always @(posedge clk\_in)

begin

d1 <= clr\_i;

d2 <= d1;

d3 <= d2;

end

assign clr\_o = d1 && d2 && !d3;

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 24.09.2019 20:06:18

// Design Name:

// Module Name: fsm\_moore\_ccd

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module fsm\_moore\_ccd(

input clk,

input enter,

input [3:0] din,

output reg [3:0] present\_state

);

reg [3:0] next\_state;

parameter s0=4'd0, s1=4'd1, s2=4'd2, s3=4'd3, s4=4'd4, s5=4'd5, s6=4'd6, s7=4'd7, s8=4'd8, s9=4'd9, s10=4'd10, s11=4'd11, s12=4'd12;

// s4- invalid

// s6- pass

// s5- fail

always @(posedge clk)

begin

present\_state <= next\_state;

end

always @( present\_state )

begin

case ( present\_state )

s0 :if ( enter )

next\_state <= s4;

else if (din == 4'd12)

next\_state <= s1;

else

next\_state <= s7;

s1 :if ( enter )

next\_state <= s4;

else if (din == 4'd12)

next\_state <= s2;

else

next\_state <= s8;

s2 :if ( enter )

next\_state <= s4;

else if ( din == 4'd13 )

next\_state <= s3;

else

next\_state <= s9;

s3: if ( enter )

next\_state <= s6;

else

next\_state <= s10;

s7 : if ( enter )

next\_state <= s4;

else

next\_state <=s8;

s8: if ( enter )

next\_state <= s4;

else

next\_state <= s9;

s9 :if ( enter )

next\_state <= s5;

else

next\_state <= s10;

s10 : if ( enter )

next\_state <= s4;

else

next\_state <= s10;

s4 : if ( din == 4'd12 )

next\_state <= s1;

else

next\_state <= s7;

s6 : if ( din == 4'd12 )

next\_state <= s1;

else

next\_state <= s7;

s5 : if ( din == 4'd12 )

next\_state <= s1;

else

next\_state <= s7;

default : next\_state <= s0;

endcase

end

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 24.09.2019 20:11:00

// Design Name:

// Module Name: clk\_div

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module clk\_div

#(

parameter divide\_by = 19

)

( input clk\_in,

output clk\_out );

reg [divide\_by-1:0] count = 0 ;

always @ (posedge clk\_in)

begin

count = count+1 ;

end

assign clk\_out = count[divide\_by-1] ;

endmodule

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 24.09.2019 20:12:43

// Design Name:

// Module Name: sevenSeg

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module sevenSeg(

input clk,

input [3:0] ones,

input [3:0] tens,

input [3:0] hundreds,

input [3:0] thousands,

output [3:0] anode,

output [7:0] cathode

);

reg [6:0] cathode\_temp = 7'b1111111;

reg [3:0] anode\_temp = 4'b1110;

reg [1:0] count = 0;

always @(posedge clk)

begin

count = count + 1;

end

always @ (posedge clk)

begin

case(count)

2'b00:

begin

case(ones)

4'd1:cathode\_temp = 7'b0011000;

4'd2:cathode\_temp = 7'b0001000;

4'd3:cathode\_temp = 7'b0100100;

4'd4:cathode\_temp = 7'b1001111;

4'd5:cathode\_temp = 7'b1101010;

4'd6:cathode\_temp = 7'b1000001;

4'd7:cathode\_temp = 7'b0111000;

4'd8:cathode\_temp = 7'b1110001;

default:cathode\_temp = 7'b1111111;

endcase

anode\_temp = 4'b1110;

end

2'b01:

begin

case(tens)

4'd1 : cathode\_temp = 7'b0011000;

4'd2:cathode\_temp = 7'b0001000;

4'd3:cathode\_temp = 7'b0100100;

4'd4:cathode\_temp = 7'b1001111;

4'd5:cathode\_temp = 7'b1101010;

4'd6:cathode\_temp = 7'b1000001;

4'd7:cathode\_temp = 7'b0111000;

4'd8:cathode\_temp = 7'b1110001;

default:cathode\_temp = 7'b1111111;

endcase

anode\_temp = 4'b1101;

end

2'b10:

begin

case(hundreds)

4'd1 : cathode\_temp = 7'b0011000;

4'd2:cathode\_temp = 7'b0001000;

4'd3:cathode\_temp = 7'b0100100;

4'd4:cathode\_temp = 7'b1001111;

4'd5:cathode\_temp = 7'b1101010;

4'd6:cathode\_temp = 7'b1000001;

4'd7:cathode\_temp = 7'b0111000;

4'd8:cathode\_temp = 7'b1110001;

default:cathode\_temp = 7'b1111111;

endcase

anode\_temp = 4'b1011;

end

2'b11:

begin

case(thousands)

4'd1 : cathode\_temp = 7'b0011000;

4'd2:cathode\_temp = 7'b0001000;

4'd3:cathode\_temp = 7'b0100100;

4'd4:cathode\_temp = 7'b1001111;

4'd5:cathode\_temp = 7'b1101010;

4'd6:cathode\_temp = 7'b1000001;

4'd7:cathode\_temp = 7'b0111000;

4'd8:cathode\_temp = 7'b1110001;

default:cathode\_temp = 7'b1111111;

endcase

anode\_temp = 4'b0111;

end

endcase

end

assign anode = anode\_temp;

assign cathode = {cathode\_temp,1'b1};

endmodule

## This file is a general .xdc for the Basys3 rev B board

## To use it in a project:

## - uncomment the lines corresponding to used pins

## - rename the used ports (in each line, after get\_ports) according to the top level signal names in the project

# Clock signal

set\_property PACKAGE\_PIN W5 [get\_ports clk\_100M]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk\_100M]

# create\_clock -add -name sys\_clk\_pin -period 10.00 -waveform {0 5} [get\_ports clk]

## Switches

#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]}]

## LEDs

#set\_property PACKAGE\_PIN U16 [get\_ports {led[0]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[0]}]

#set\_property PACKAGE\_PIN E19 [get\_ports {led[1]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[1]}]

#set\_property PACKAGE\_PIN U19 [get\_ports {led[2]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[2]}]

#set\_property PACKAGE\_PIN V19 [get\_ports {led[3]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[3]}]

#set\_property PACKAGE\_PIN W18 [get\_ports {led[4]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[4]}]

#set\_property PACKAGE\_PIN U15 [get\_ports {led[5]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[5]}]

#set\_property PACKAGE\_PIN U14 [get\_ports {led[6]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[6]}]

#set\_property PACKAGE\_PIN V14 [get\_ports {led[7]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[7]}]

#set\_property PACKAGE\_PIN V13 [get\_ports {led[8]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[8]}]

#set\_property PACKAGE\_PIN V3 [get\_ports {led[9]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[9]}]

#set\_property PACKAGE\_PIN W3 [get\_ports {led[10]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[10]}]

#set\_property PACKAGE\_PIN U3 [get\_ports {led[11]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[11]}]

#set\_property PACKAGE\_PIN P3 [get\_ports {led[12]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[12]}]

#set\_property PACKAGE\_PIN N3 [get\_ports {led[13]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[13]}]

#set\_property PACKAGE\_PIN P1 [get\_ports {led[14]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[14]}]

#set\_property PACKAGE\_PIN L1 [get\_ports {led[15]}]

# set\_property IOSTANDARD LVCMOS33 [get\_ports {led[15]}]

##7 segment display

set\_property PACKAGE\_PIN W7 [get\_ports {cathode[7]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {cathode[7]}]

set\_property PACKAGE\_PIN W6 [get\_ports {cathode[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {cathode[6]}]

set\_property PACKAGE\_PIN U8 [get\_ports {cathode[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {cathode[5]}]

set\_property PACKAGE\_PIN V8 [get\_ports {cathode[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {cathode[4]}]

set\_property PACKAGE\_PIN U5 [get\_ports {cathode[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {cathode[3]}]

set\_property PACKAGE\_PIN V5 [get\_ports {cathode[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {cathode[2]}]

set\_property PACKAGE\_PIN U7 [get\_ports {cathode[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {cathode[1]}]

set\_property PACKAGE\_PIN V7 [ get\_ports {cathode[0] }]

set\_property IOSTANDARD LVCMOS33 [get\_ports {cathode[0] }]

set\_property PACKAGE\_PIN U2 [get\_ports {anode[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {anode[0]}]

set\_property PACKAGE\_PIN U4 [get\_ports {anode[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {anode[1]}]

set\_property PACKAGE\_PIN V4 [get\_ports {anode[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {anode[2]}]

set\_property PACKAGE\_PIN W4 [get\_ports {anode[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {anode[3]}]

##Buttons

set\_property PACKAGE\_PIN U18 [get\_ports enter]

set\_property IOSTANDARD LVCMOS33 [get\_ports enter]

#set\_property PACKAGE\_PIN T18 [get\_ports btnU]

#set\_property IOSTANDARD LVCMOS33 [get\_ports btnU]

#set\_property PACKAGE\_PIN W19 [get\_ports btnL]

#set\_property IOSTANDARD LVCMOS33 [get\_ports btnL]

#set\_property PACKAGE\_PIN T17 [get\_ports btnR]

#set\_property IOSTANDARD LVCMOS33 [get\_ports btnR]

#set\_property PACKAGE\_PIN U17 [get\_ports btnD]

#set\_property IOSTANDARD LVCMOS33 [get\_ports btnD]

##Pmod Header JA

##Sch name = JA1

set\_property PACKAGE\_PIN J1 [get\_ports {JA[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[0]}]

#Sch name = JA2

set\_property PACKAGE\_PIN L2 [get\_ports {JA[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[1]}]

#Sch name = JA3

set\_property PACKAGE\_PIN J2 [get\_ports {JA[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[2]}]

#Sch name = JA4

set\_property PACKAGE\_PIN G2 [get\_ports {JA[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[3]}]

#Sch name = JA7

set\_property PACKAGE\_PIN H1 [get\_ports {JA[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[4]}]

#Sch name = JA8

set\_property PACKAGE\_PIN K2 [get\_ports {JA[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[5]}]

#Sch name = JA9

set\_property PACKAGE\_PIN H2 [get\_ports {JA[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[6]}]

#Sch name = JA10

set\_property PACKAGE\_PIN G3 [get\_ports {JA[7]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {JA[7]}]

##Pmod Header JB

##Sch name = JB1

#set\_property PACKAGE\_PIN A14 [get\_ports {JB[0]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[0]}]

##Sch name = JB2

#set\_property PACKAGE\_PIN A16 [get\_ports {JB[1]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[1]}]

##Sch name = JB3

#set\_property PACKAGE\_PIN B15 [get\_ports {JB[2]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[2]}]

##Sch name = JB4

#set\_property PACKAGE\_PIN B16 [get\_ports {JB[3]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[3]}]

##Sch name = JB7

#set\_property PACKAGE\_PIN A15 [get\_ports {JB[4]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[4]}]

##Sch name = JB8

#set\_property PACKAGE\_PIN A17 [get\_ports {JB[5]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[5]}]

##Sch name = JB9

#set\_property PACKAGE\_PIN C15 [get\_ports {JB[6]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[6]}]

##Sch name = JB10

#set\_property PACKAGE\_PIN C16 [get\_ports {JB[7]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JB[7]}]

##Pmod Header JC

##Sch name = JC1

#set\_property PACKAGE\_PIN K17 [get\_ports {JC[0]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[0]}]

##Sch name = JC2

#set\_property PACKAGE\_PIN M18 [get\_ports {JC[1]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[1]}]

##Sch name = JC3

#set\_property PACKAGE\_PIN N17 [get\_ports {JC[2]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[2]}]

##Sch name = JC4

#set\_property PACKAGE\_PIN P18 [get\_ports {JC[3]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[3]}]

##Sch name = JC7

#set\_property PACKAGE\_PIN L17 [get\_ports {JC[4]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[4]}]

##Sch name = JC8

#set\_property PACKAGE\_PIN M19 [get\_ports {JC[5]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[5]}]

##Sch name = JC9

#set\_property PACKAGE\_PIN P17 [get\_ports {JC[6]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[6]}]

##Sch name = JC10

#set\_property PACKAGE\_PIN R18 [get\_ports {JC[7]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JC[7]}]

##Pmod Header JXADC

##Sch name = XA1\_P

#set\_property PACKAGE\_PIN J3 [get\_ports {JXADC[0]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[0]}]

##Sch name = XA2\_P

#set\_property PACKAGE\_PIN L3 [get\_ports {JXADC[1]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[1]}]

##Sch name = XA3\_P

#set\_property PACKAGE\_PIN M2 [get\_ports {JXADC[2]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[2]}]

##Sch name = XA4\_P

#set\_property PACKAGE\_PIN N2 [get\_ports {JXADC[3]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[3]}]

##Sch name = XA1\_N

#set\_property PACKAGE\_PIN K3 [get\_ports {JXADC[4]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[4]}]

##Sch name = XA2\_N

#set\_property PACKAGE\_PIN M3 [get\_ports {JXADC[5]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[5]}]

##Sch name = XA3\_N

#set\_property PACKAGE\_PIN M1 [get\_ports {JXADC[6]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[6]}]

##Sch name = XA4\_N

#set\_property PACKAGE\_PIN N1 [get\_ports {JXADC[7]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {JXADC[7]}]

##VGA Connector

#set\_property PACKAGE\_PIN G19 [get\_ports {vgaRed[0]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaRed[0]}]

#set\_property PACKAGE\_PIN H19 [get\_ports {vgaRed[1]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaRed[1]}]

#set\_property PACKAGE\_PIN J19 [get\_ports {vgaRed[2]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaRed[2]}]

#set\_property PACKAGE\_PIN N19 [get\_ports {vgaRed[3]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaRed[3]}]

#set\_property PACKAGE\_PIN N18 [get\_ports {vgaBlue[0]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaBlue[0]}]

#set\_property PACKAGE\_PIN L18 [get\_ports {vgaBlue[1]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaBlue[1]}]

#set\_property PACKAGE\_PIN K18 [get\_ports {vgaBlue[2]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaBlue[2]}]

#set\_property PACKAGE\_PIN J18 [get\_ports {vgaBlue[3]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaBlue[3]}]

#set\_property PACKAGE\_PIN J17 [get\_ports {vgaGreen[0]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaGreen[0]}]

#set\_property PACKAGE\_PIN H17 [get\_ports {vgaGreen[1]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaGreen[1]}]

#set\_property PACKAGE\_PIN G17 [get\_ports {vgaGreen[2]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaGreen[2]}]

#set\_property PACKAGE\_PIN D17 [get\_ports {vgaGreen[3]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {vgaGreen[3]}]

#set\_property PACKAGE\_PIN P19 [get\_ports Hsync]

#set\_property IOSTANDARD LVCMOS33 [get\_ports Hsync]

#set\_property PACKAGE\_PIN R19 [get\_ports Vsync]

#set\_property IOSTANDARD LVCMOS33 [get\_ports Vsync]

##USB-RS232 Interface

#set\_property PACKAGE\_PIN B18 [get\_ports RsRx]

#set\_property IOSTANDARD LVCMOS33 [get\_ports RsRx]

#set\_property PACKAGE\_PIN A18 [get\_ports tx]

#set\_property IOSTANDARD LVCMOS33 [get\_ports tx]

#USB HID (PS/2)

##Quad SPI Flash

##Note that CCLK\_0 cannot be placed in 7 series devices. You can access it using the

##STARTUPE2 primitive.

#set\_property PACKAGE\_PIN D18 [get\_ports {QspiDB[0]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {QspiDB[0]}]

#set\_property PACKAGE\_PIN D19 [get\_ports {QspiDB[1]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {QspiDB[1]}]

#set\_property PACKAGE\_PIN G18 [get\_ports {QspiDB[2]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {QspiDB[2]}]

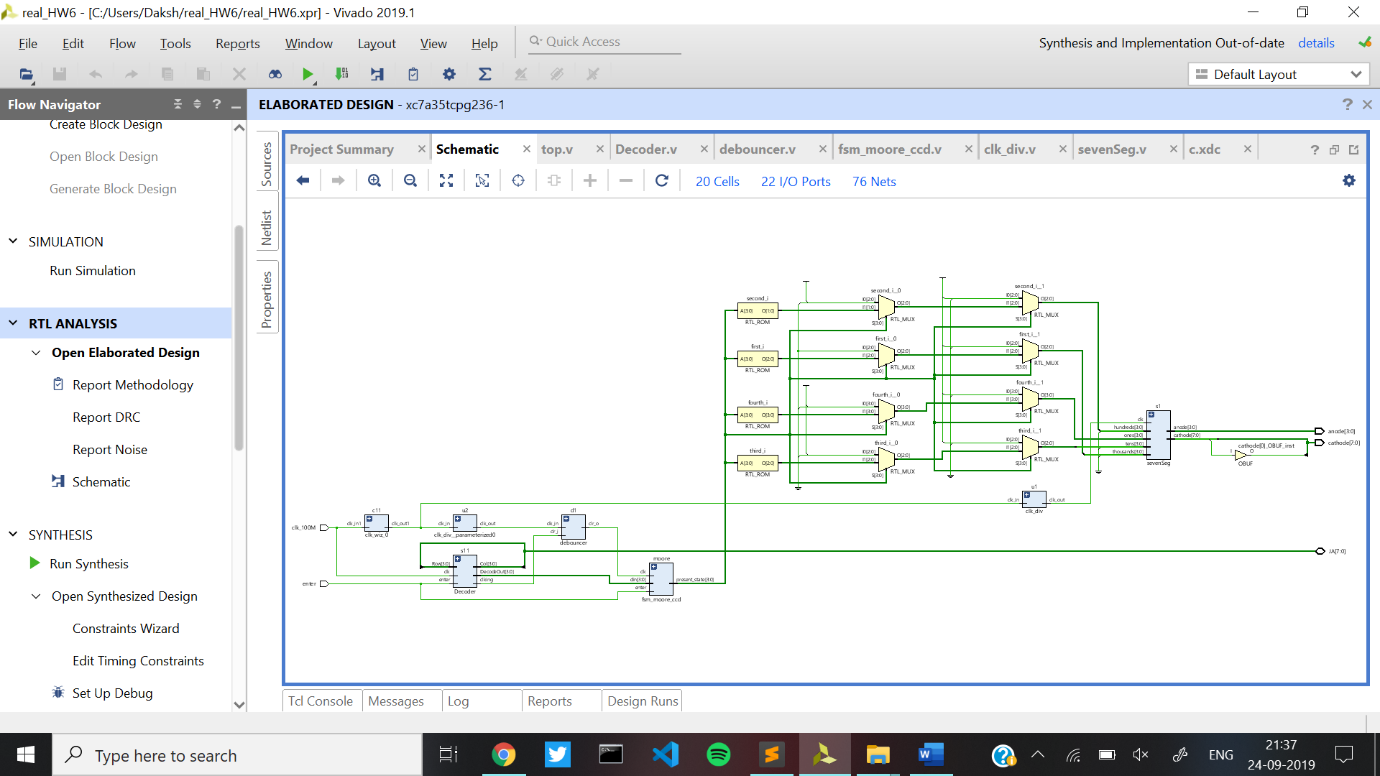
#set\_property PACKAGE\_PIN F18 [get\_ports {QspiDB[3]}]

#set\_property IOSTANDARD LVCMOS33 [get\_ports {QspiDB[3]}]

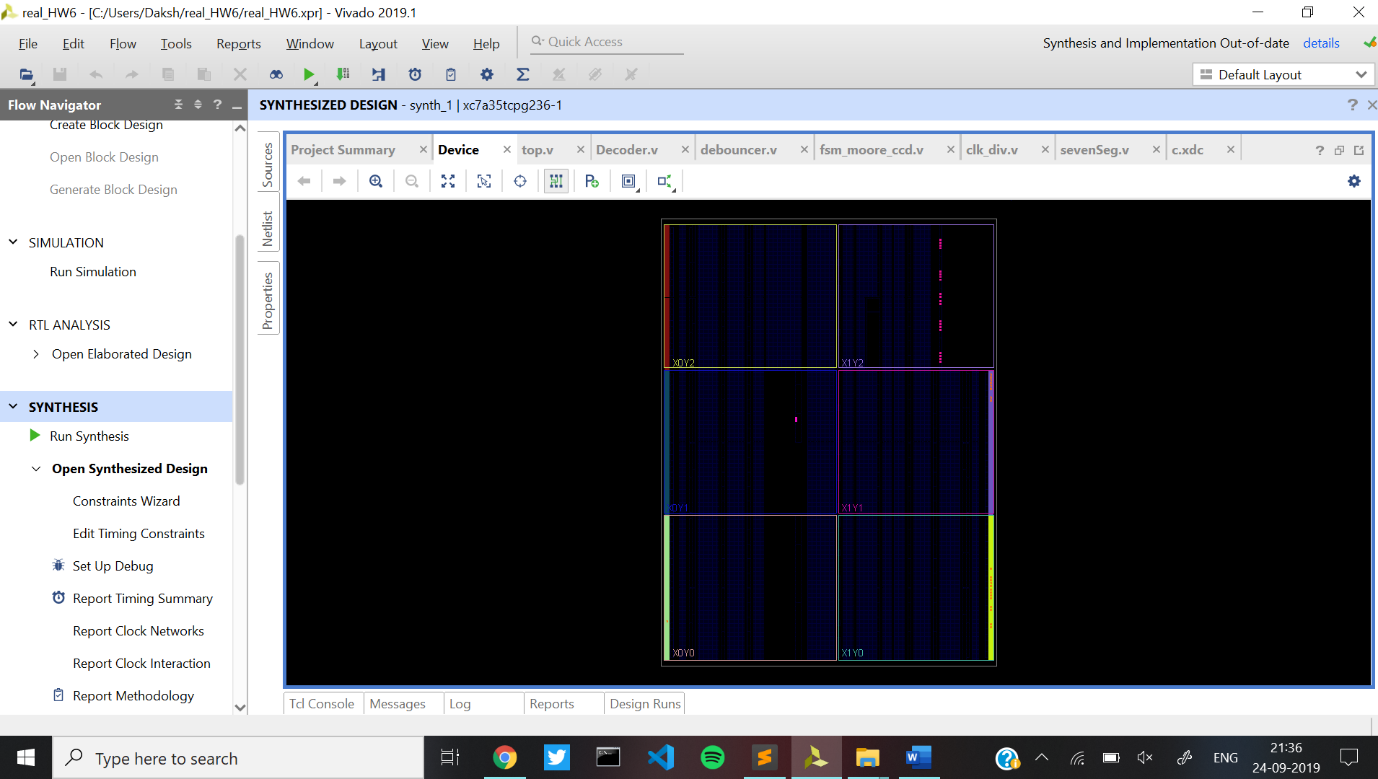
#set\_property PACKAGE\_PIN K19 [get\_ports QspiCSn]

#set\_property IOSTANDARD LVCMOS33 [get\_ports QspiCSn]

Elaborated Design



Synthesised Design-



Implemented Design

