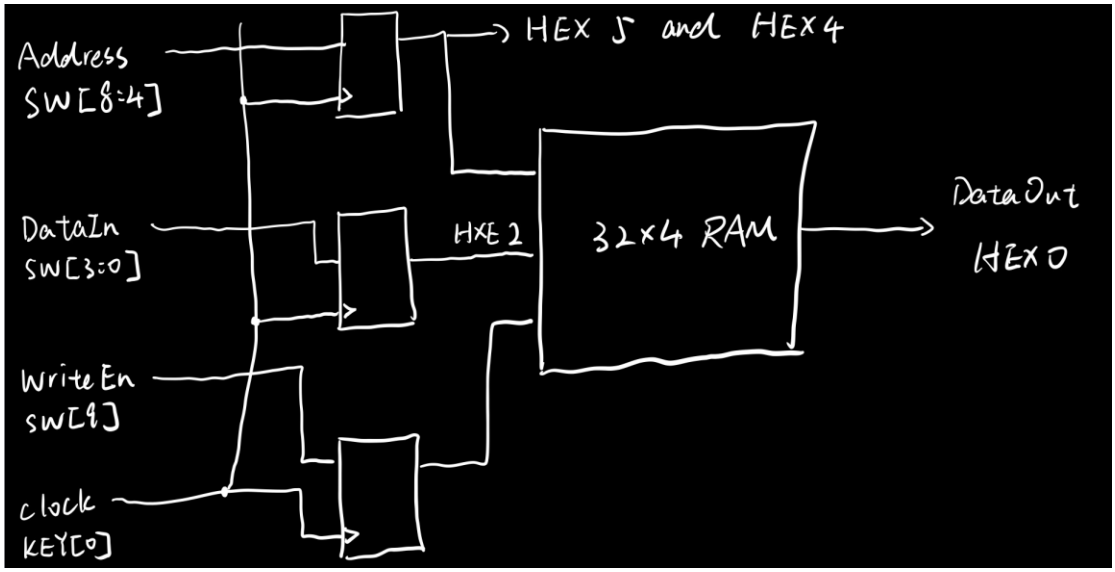
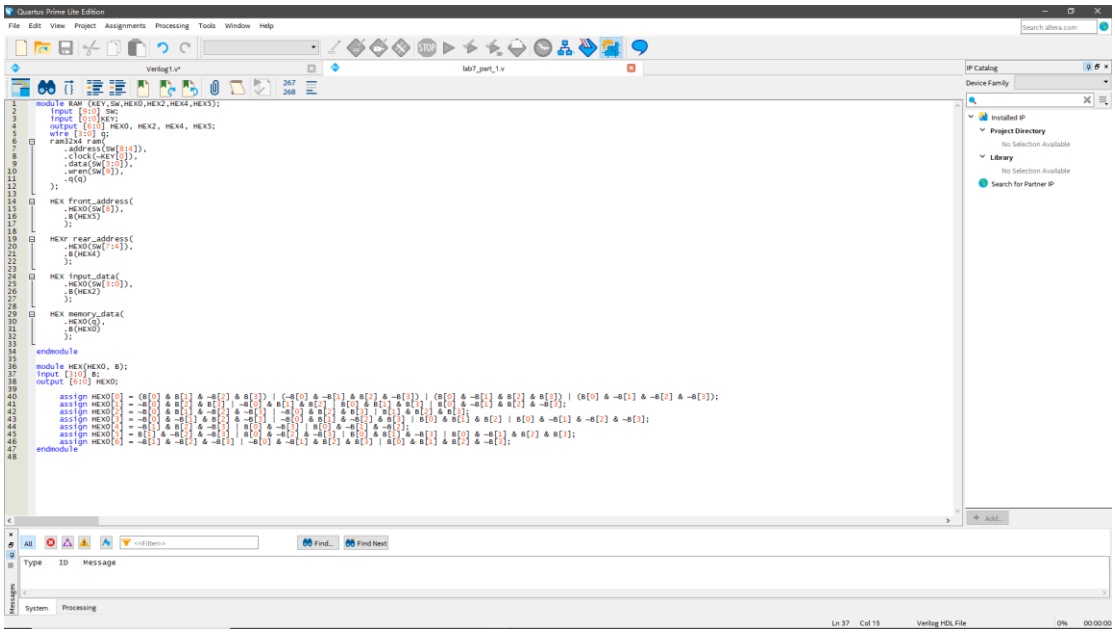
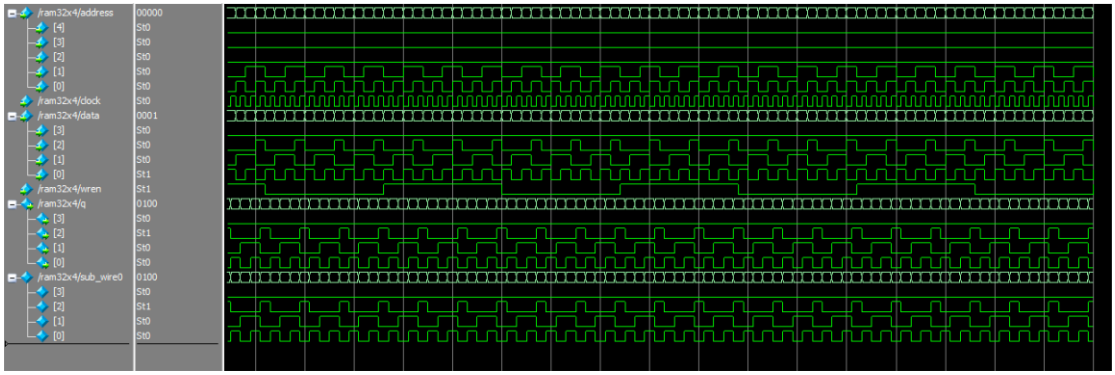
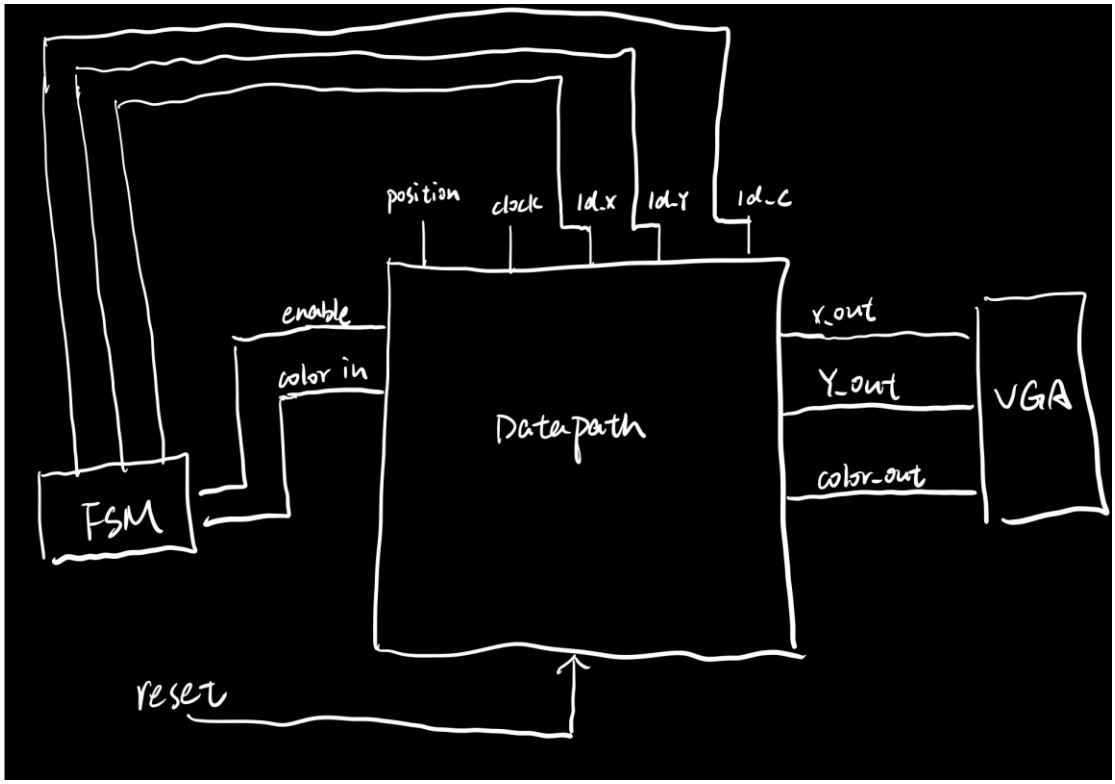
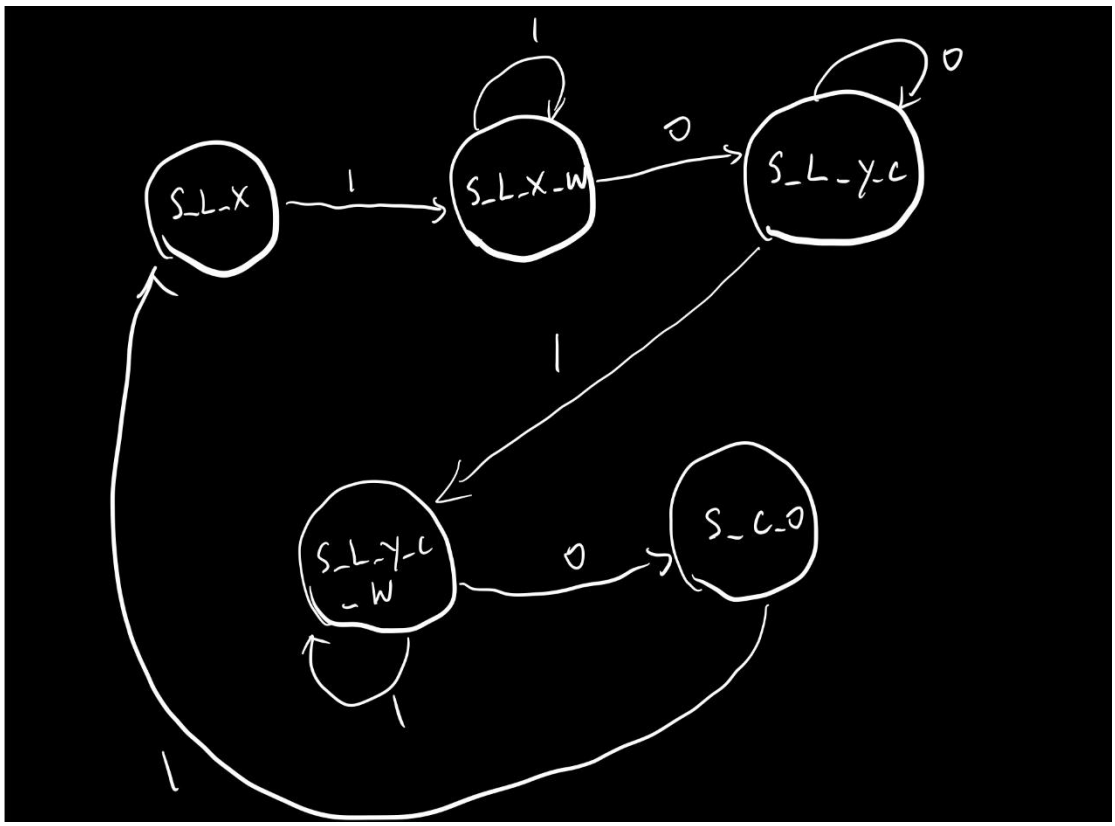
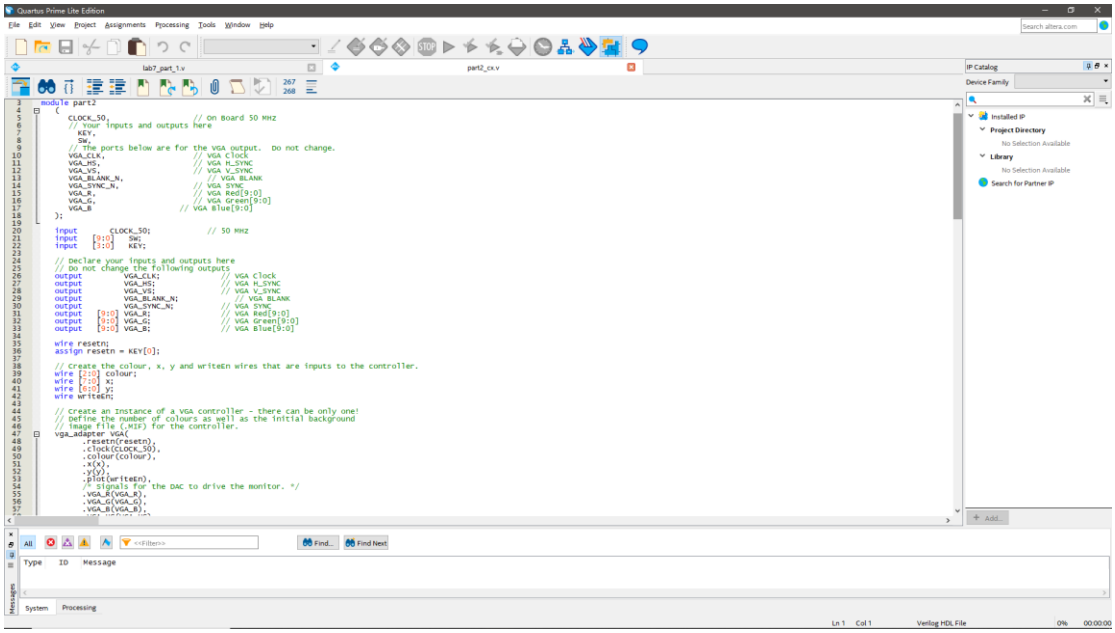

Address
SW[8:4]
DataIn
SW[3:0]
WriteEn
SW[9]
clock
KEY[0]
HEX5 and HEX4
HEX2
32x4 RAM
DataOut
HEX0

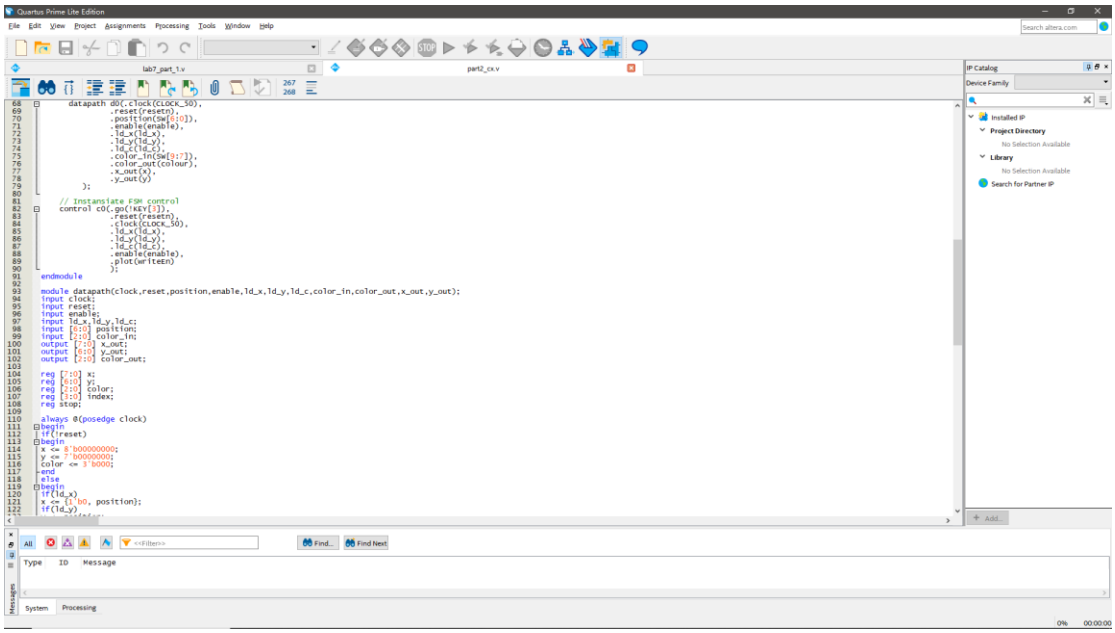

Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
Verilog1.v*
lab7_part_1.v
IP Catalog
Device Family
Installed IP
Project Directory
No Selection Available
Library
No Selection Available
Search for Partner IP
module RAM (KEY,SW,HEX0,HEX2,HEX4,HEX5);
input [9:0] SW;
input [3:0]KEY;
output [6:0] HEX0, HEX2, HEX4, HEX5;
wire [3:0] q;
ram32x4 ram(
.address(SW[8:4]),
.clock(~KEY[0]),
.data(SW[3:0]),
.wren(SW[9]),
.q(q)
);
HEX front_address(
.HEX0(SW[8]),
.B(HEX5)
);
HEXr rear_address(
.HEX0(SW[7:4]),
.B(HEX4)
);
HEX input_data(
.HEX0(SW[3:0]),
.B(HEX2)
);
HEX memory_data(
.HEX0(q),
.B(HEX0)
);
endmodule
module HEX(HEX0, B);
input [3:0] B;
output [6:0] HEX0;
assign HEX0[0] = (B[0] & B[1] & ~B[2] & B[3]) | (~B[0] & ~B[1] & B[2] & ~B[3]) | (B[0] & ~B[1] & B[2] & B[3]) | (B[0] & ~B[1] & ~B[2] & ~B[3]);
assign HEX0[1] = ~B[0] & B[2] & B[3] | ~B[0] & B[1] & B[3] | B[0] & B[1] & ~B[2] | B[0] & B[1] & B[3] | B[0] & ~B[1] & B[2] & ~B[3];
assign HEX0[2] = ~B[0] & B[1] & ~B[2] & ~B[3] | ~B[0] & B[2] & B[3] | B[0] & ~B[1] & B[2] & B[3];
assign HEX0[3] = ~B[0] & ~B[1] & B[2] & ~B[3] | ~B[0] & B[1] & ~B[2] & B[3] | B[0] & B[1] & B[2] | B[0] & ~B[1] & ~B[2] & ~B[3];
assign HEX0[4] = ~B[1] & B[2] & ~B[3] | B[0] & ~B[2] | B[0] & ~B[1] & ~B[3];
assign HEX0[5] = B[1] & ~B[2] & ~B[3] | B[0] & ~B[2] | ~B[0] & B[1] & ~B[3] | B[0] & B[1] & ~B[3] | B[0] & ~B[1] & B[2] & B[3];
assign HEX0[6] = ~B[1] & ~B[2] & ~B[3] | ~B[0] & ~B[1] & B[2] & B[3] | B[0] & B[1] & B[2] & ~B[3];
endmodule
All
Find...
Find Next
Type ID Message
System Processing
Ln 37 Col 15
Verilog HDL File
0%
00:00:00

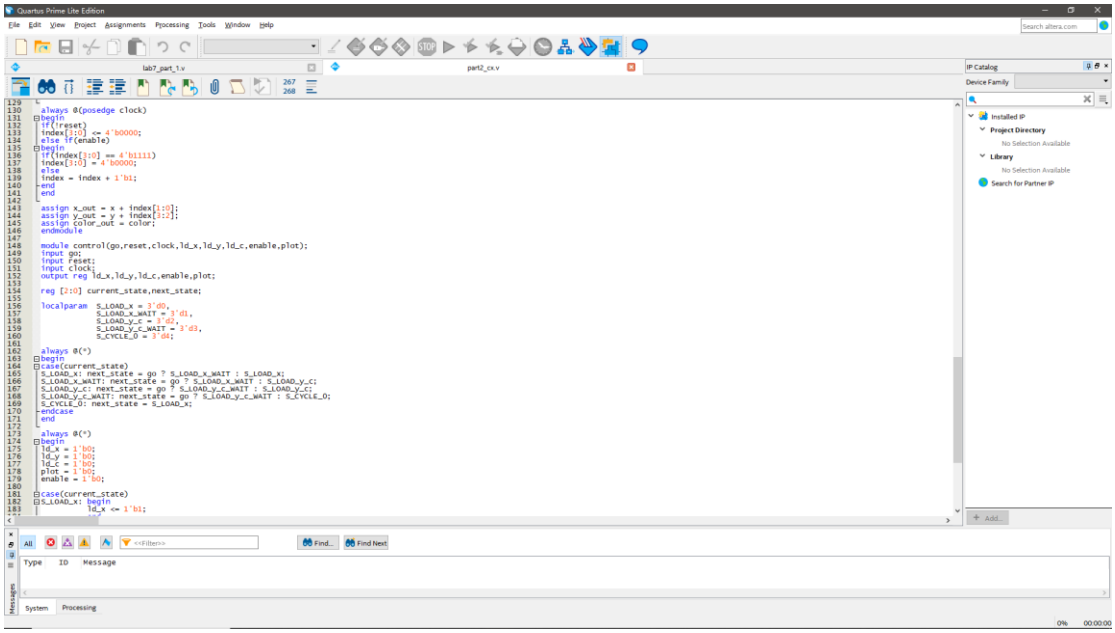

/ram32x4/address
/ram32x4/clock
/ram32x4/data
/ram32x4/wren
/ram32x4/q
/ram32x4/sub_wire0

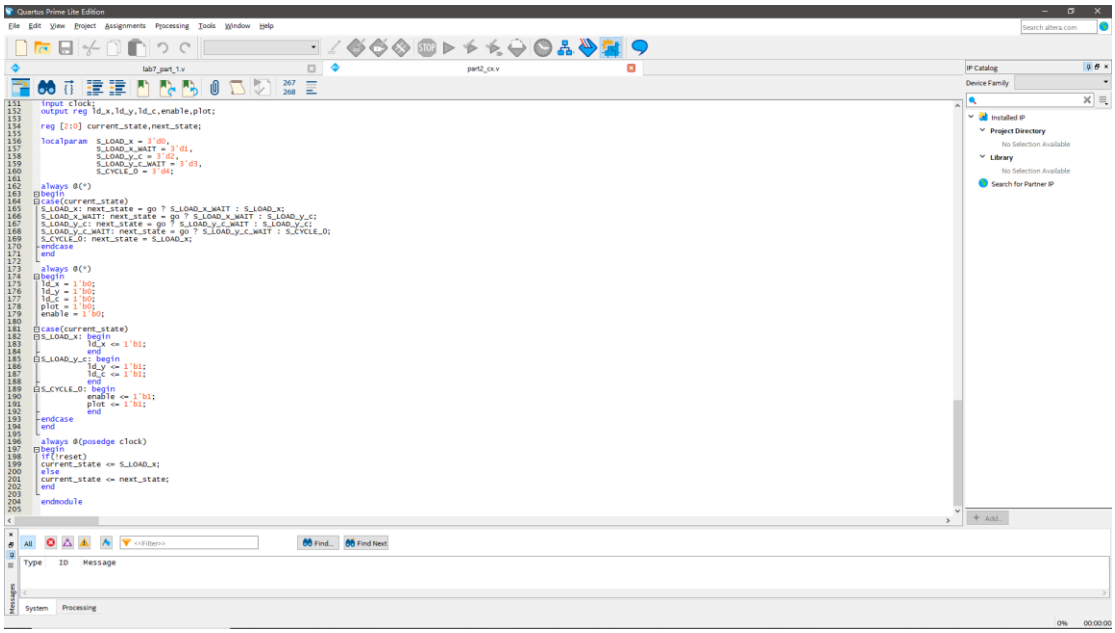

position
clock
ld_x
ld_y
ld_c
enable
color in
Datapath
x_out
Y_out
color_out
VGA
FSM
reset

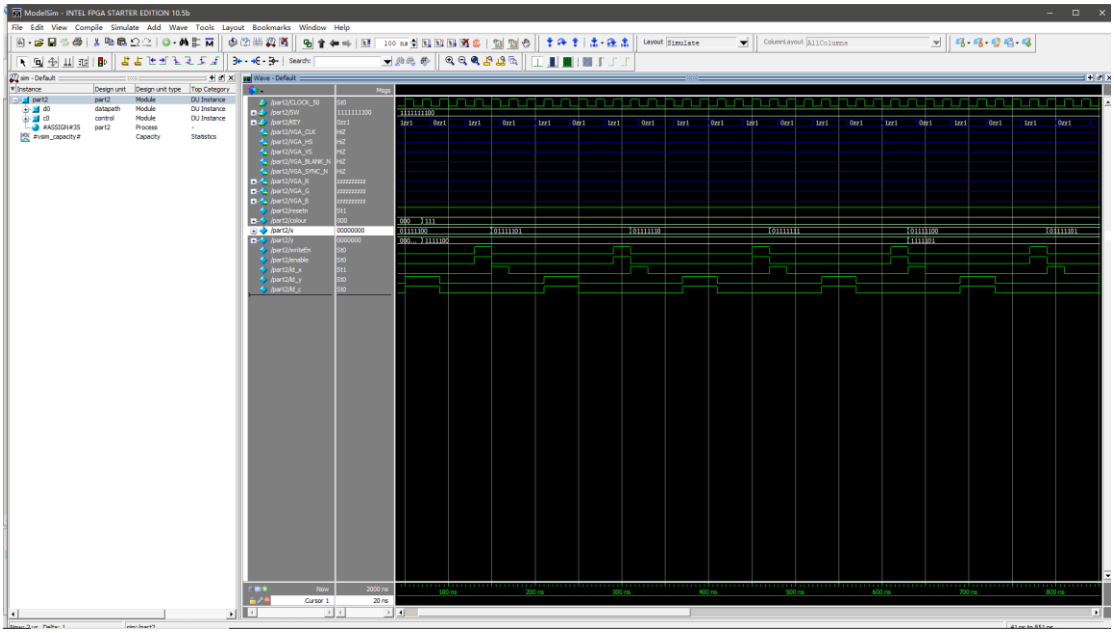

S_L_X
S_L_X_W
S_L_Y_C
S_L_Y_C_W
S_C_D
1
0
1
0
1
1


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
lab7_part_1.v
part2_cx.v
IP Catalog
Device Family
Installed IP
Project Directory
No Selection Available
Library
No Selection Available
Search for Partner IP
module part2
CLOCK_50, // On Board 50 MHz
// Your inputs and outputs here
KEY,
SW,
// The ports below are for the VGA output. Do not change.
VGA_CLK, // VGA Clock
VGA_HS, // VGA H_SYNC
VGA_VS, // VGA V_SYNC
VGA_BLANK_N, // VGA BLANK
VGA_SYNC_N, // VGA SYNC
VGA_R, // VGA Red[9:0]
VGA_G, // VGA Green[9:0]
VGA_B // VGA Blue[9:0]
);
input CLOCK_50; // 50 MHz
input [9:0] SW;
input [3:0] KEY;
// Declare your inputs and outputs here
// Do not change the following outputs
output VGA_CLK; // VGA Clock
output VGA_HS; // VGA H_SYNC
output VGA_VS; // VGA V_SYNC
output VGA_BLANK_N; // VGA BLANK
output VGA_SYNC_N; // VGA SYNC
output [9:0] VGA_R; // VGA Red[9:0]
output [9:0] VGA_G; // VGA Green[9:0]
output [9:0] VGA_B; // VGA Blue[9:0]
wire resetn;
assign resetn = KEY[0];
// Create the colour, x, y and writeEn wires that are inputs to the controller.
wire [2:0] colour;
wire [7:0] x;
wire [6:0] y;
wire writeEn;
// Create an Instance of a VGA controller - there can be only one!
// Define the number of colours as well as the initial background
// image file (.MIF) for the controller.
vga_adapter VGA(
.resetn(resetn),
.clock(CLOCK_50),
.colour(colour),
.x(x),
.y(y),
.plot(writeEn),
/* Signals for the DAC to drive the monitor. */
.VGA_R(VGA_R),
.VGA_G(VGA_G),
.VGA_B(VGA_B),
Find...
Find Next
Type ID Message
System
Processing
Ln 1 Col 1
Verilog HDL File
0%
00:00:00


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
lab7_part_1.v
part2_cx.v
IP Catalog
Device Family
Installed IP
Project Directory
No Selection Available
Library
No Selection Available
Search for Partner IP
datapath d0(.clock(CLOCK_50),
.reset(resetn),
.position(SW[6:0]),
.enable(enable),
.ld_x(ld_x),
.ld_y(ld_y),
.ld_c(ld_c),
.color_in(SW[9:7]),
.color_out(colour),
.x_out(x),
.y_out(y)
);
// Instansiate FSM control
control c0(.go(!KEY[3]),
.reset(resetn),
.clock(CLOCK_50),
.ld_x(ld_x),
.ld_y(ld_y),
.ld_c(ld_c),
.enable(enable),
.plot(writeEn)
);
endmodule
module datapath(clock,reset,position,enable,ld_x,ld_y,ld_c,color_in,color_out,x_out,y_out);
input clock;
input reset;
input enable;
input ld_x,ld_y,ld_c;
input [6:0] position;
input [2:0] color_in;
output [7:0] x_out;
output [6:0] y_out;
output [2:0] color_out;
reg [7:0] x;
reg [6:0] y;
reg [2:0] color;
reg [3:0] index;
reg stop;
always @(posedge clock)
begin
if(!reset)
begin
x <= 8'b00000000;
y <= 7'b0000000;
color <= 3'b000;
end
else
begin
if(ld_x)
x <= {1'b0, position};
if(ld_y)
Find...
Find Next
Type ID Message
System
Processing
0%
00:00:00


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
lab7_part_1.v
part2_cx.v
IP Catalog
Device Family
Installed IP
Project Directory
No Selection Available
Library
No Selection Available
Search for Partner IP
always @(posedge clock)
begin
if(reset)
index[3:0] <= 4'b0000;
else if(enable)
begin
if(index[3:0] == 4'b1111)
index[3:0] = 4'b0000;
else
index = index + 1'b1;
end
end
assign x_out = x + index[1:0];
assign y_out = y + index[3:2];
assign color_out = color;
endmodule
module control(go,reset,clock,ld_x,ld_y,ld_c,enable,plot);
input go;
input reset;
input clock;
output reg ld_x,ld_y,ld_c,enable,plot;
reg [2:0] current_state,next_state;
localparam S_LOAD_x = 3'd0,
S_LOAD_x_WAIT = 3'd1,
S_LOAD_y_c = 3'd2,
S_LOAD_y_c_WAIT = 3'd3,
S_CYCLE_0 = 3'd4;
always @(*)
begin
case(current_state)
S_LOAD_x: next_state = go ? S_LOAD_x_WAIT : S_LOAD_x;
S_LOAD_x_WAIT: next_state = go ? S_LOAD_x_WAIT : S_LOAD_y_c;
S_LOAD_y_c: next_state = go ? S_LOAD_y_c_WAIT : S_LOAD_y_c;
S_LOAD_y_c_WAIT: next_state = go ? S_LOAD_y_c_WAIT : S_CYCLE_0;
S_CYCLE_0: next_state = S_LOAD_x;
endcase
end
always @(*)
begin
ld_x = 1'b0;
ld_y = 1'b0;
ld_c = 1'b0;
plot = 1'b0;
enable = 1'b0;
case(current_state)
S_LOAD_x: begin
ld_x <= 1'b1;
Find... Find Next
Type ID Message
System Processing
0% 00:00:00


Quartus Prime Lite Edition
File Edit View Project Assignments Processing Tools Window Help
lab7_part_1.v
part2_cx.v
IP Catalog
Device Family
Installed IP
Project Directory
No Selection Available
Library
No Selection Available
Search for Partner IP
input clock;
output reg ld_x,ld_y,ld_c,enable,plot;
reg [2:0] current_state,next_state;
localparam S_LOAD_x = 3'd0,
S_LOAD_x_WAIT = 3'd1,
S_LOAD_y_c = 3'd2,
S_LOAD_y_c_WAIT = 3'd3,
S_CYCLE_0 = 3'd4;
always @(*)
begin
case(current_state)
S_LOAD_x: next_state = go ? S_LOAD_x_WAIT : S_LOAD_x;
S_LOAD_x_WAIT: next_state = go ? S_LOAD_x_WAIT : S_LOAD_y_c;
S_LOAD_y_c: next_state = go ? S_LOAD_y_c_WAIT : S_LOAD_y_c;
S_LOAD_y_c_WAIT: next_state = go ? S_LOAD_y_c_WAIT : S_CYCLE_0;
S_CYCLE_0: next_state = S_LOAD_x;
endcase
end
always @(*)
begin
ld_x = 1'b0;
ld_y = 1'b0;
ld_c = 1'b0;
plot = 1'b0;
enable = 1'b0;
case(current_state)
S_LOAD_x: begin
ld_x <= 1'b1;
end
S_LOAD_y_c: begin
ld_y <= 1'b1;
ld_c <= 1'b1;
end
S_CYCLE_0: begin
enable <= 1'b1;
plot <= 1'b1;
end
endcase
end
always @(posedge clock)
begin
if(reset)
current_state <= S_LOAD_x;
else
current_state <= next_state;
end
endmodule
Find... Find Next
Type ID Message
System Processing
0% 00:00:00


ModelSim - INTEL FPGA STARTER EDITION 10.5b
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
sim - Default
Instance
Design unit
Design unit type
Top Category
Wave - Default
Msgs
/part2/CLOCK_50
/part2/SW
/part2/KEY
/part2/VGA_CLK
/part2/VGA_HS
/part2/VGA_VS
/part2/VGA_BLANK_N
/part2/VGA_SYNC_N
/part2/VGA_R
/part2/VGA_G
/part2/VGA_B
/part2/resetn
/part2/colour
/part2/x
/part2/y
/part2/writeEn
/part2/enable
/part2/M_x
/part2/M_y
/part2/M_c
Now
Cursor 1
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns