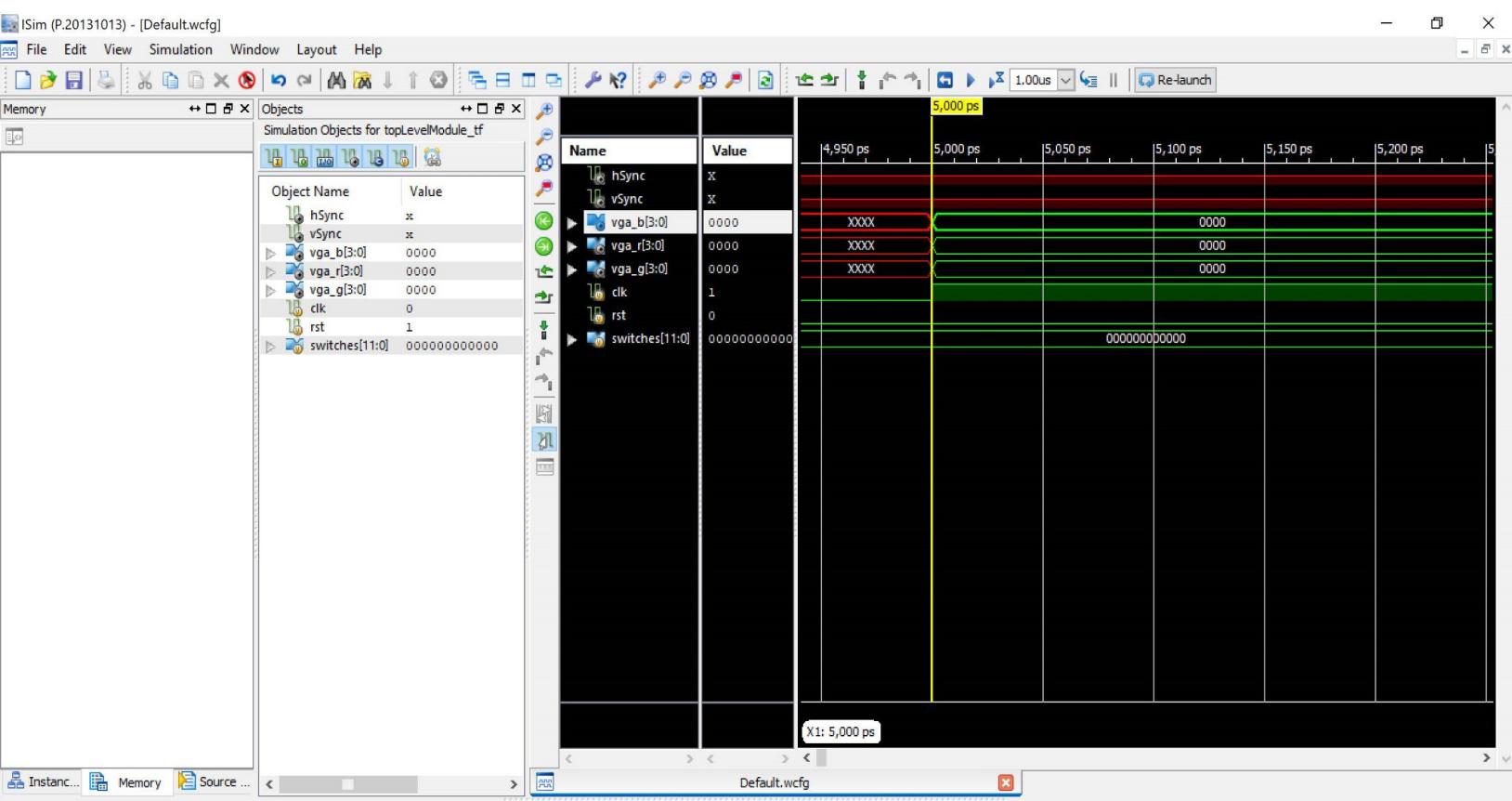


```
1 `timescale 1ns / 1ps
2 //***** File name: topLevelModule_tf.v *****
3 //
4 //
5 // Created by      Paul Valenzuela on 10/11/19.
6 // Copyright (C) 2018      Paul Valenzuela. All rights reserved.
7 //
8 //
9 //      In submitting this file for class work at CSULB
10 //      I am confirming that this is my work and the work
11 //      of no one else. In submitting this code I acknowledge that
12 //      plagiarism in student project work is subject to dismissal
13 //      from the class
14 //*****
15 //*****
16 //          topLevelModule_tf.v
17 //
18 //
19 //      The topLevelModule test fixture block tests the capabilities of the
20 //      topLevelModule. It ensures that reset resets the rgb values
21 //
22 module topLevelModule_tf;
23
24     // Inputs
25     reg clk;
26     reg rst;
27     reg [11:0] switches;
28
29     // Outputs
30     wire hSync;
31     wire vSync;
32     wire [3:0] vga_b;
33     wire [3:0] vga_r;
34     wire [3:0] vga_g;
35
36     // Instantiate the Unit Under Test (UUT)
37     topLevelModule uut (
38         .clk(clk),
39         .rst(rst),
40         .switches(switches),
41         .hSync(hSync),
42         .vSync(vSync),
43         .vga_b(vga_b),
44         .vga_r(vga_r),
45         .vga_g(vga_g)
46     );
47
48     //100Mhz Clock
49     always #5 clk = ~clk;
50
51     initial begin
52         // Initialize Inputs
53         clk = 0;
54         rst = 0;
55         switches = 0;
56
```

```
57
58      // Wait 20 ns for global reset to finish
59      #20;
60
61      rst = 1;
62
63      #10
64
65      //Test
66      if ((vga_r == 8'b0) && (vga_g == 8'b0) && (vga_b == 8'b0)) $display ("Reset
Test Sucessful, All Outputs Set to 0",vga_b);
67
68
69      end
70
71  endmodule
72
73
```



Console

WARNING: Please use Xilinx License Configuration Manager to check out a full IISim license.
 WARNING: IISim will run in Lite mode. Please refer to the IISim documentation for more information on the differences between the Lite and the Full version.
 This is a Lite version of IISim.
 Time resolution is 1 ps
 Simulator is doing circuit initialization process.
 Finished circuit initialization process.
 Reset Test Successful, All Outputs Set to 0 0

IISim>

Console Compilation Log Breakpoints Find in Files Results Search Results

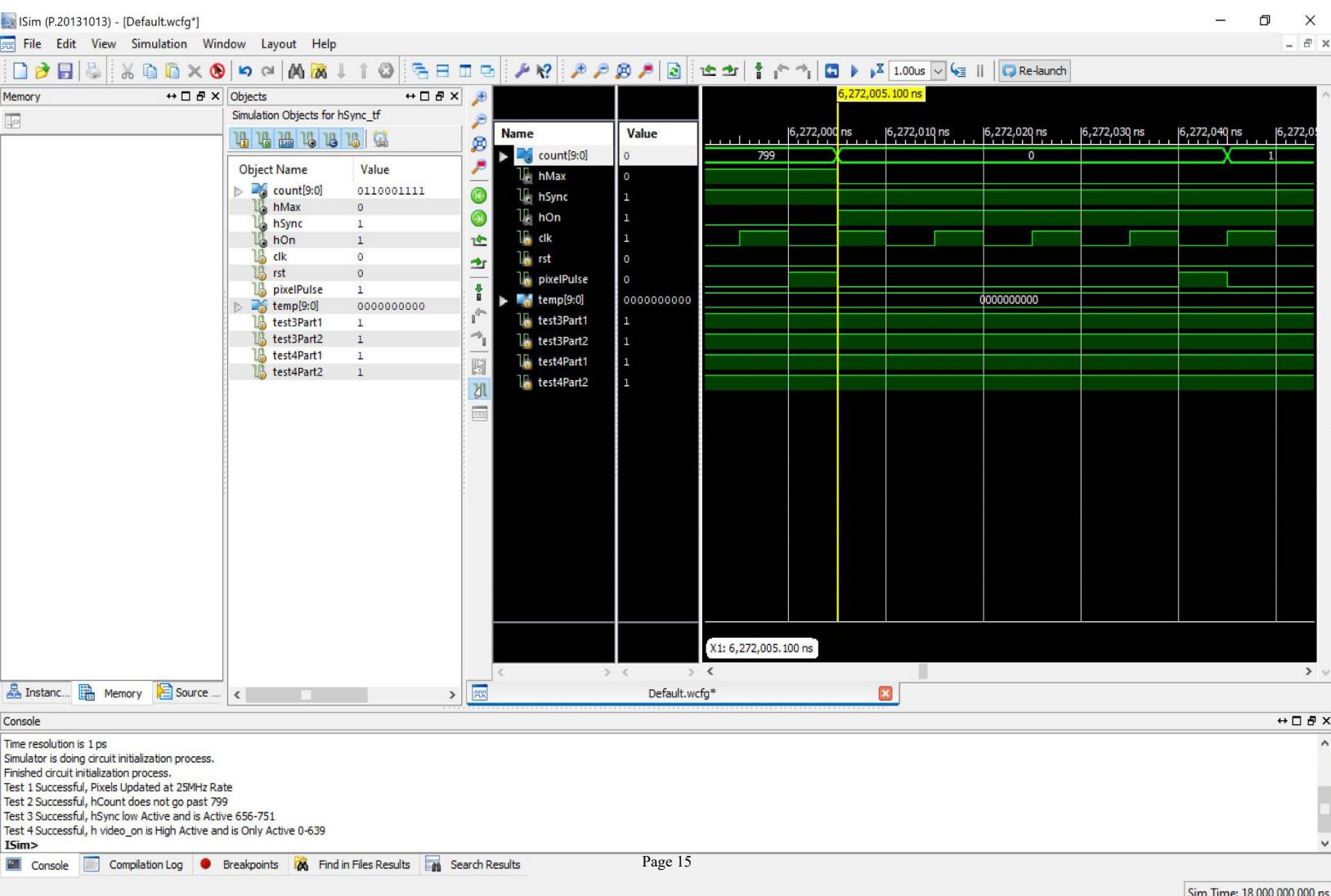
Page 12

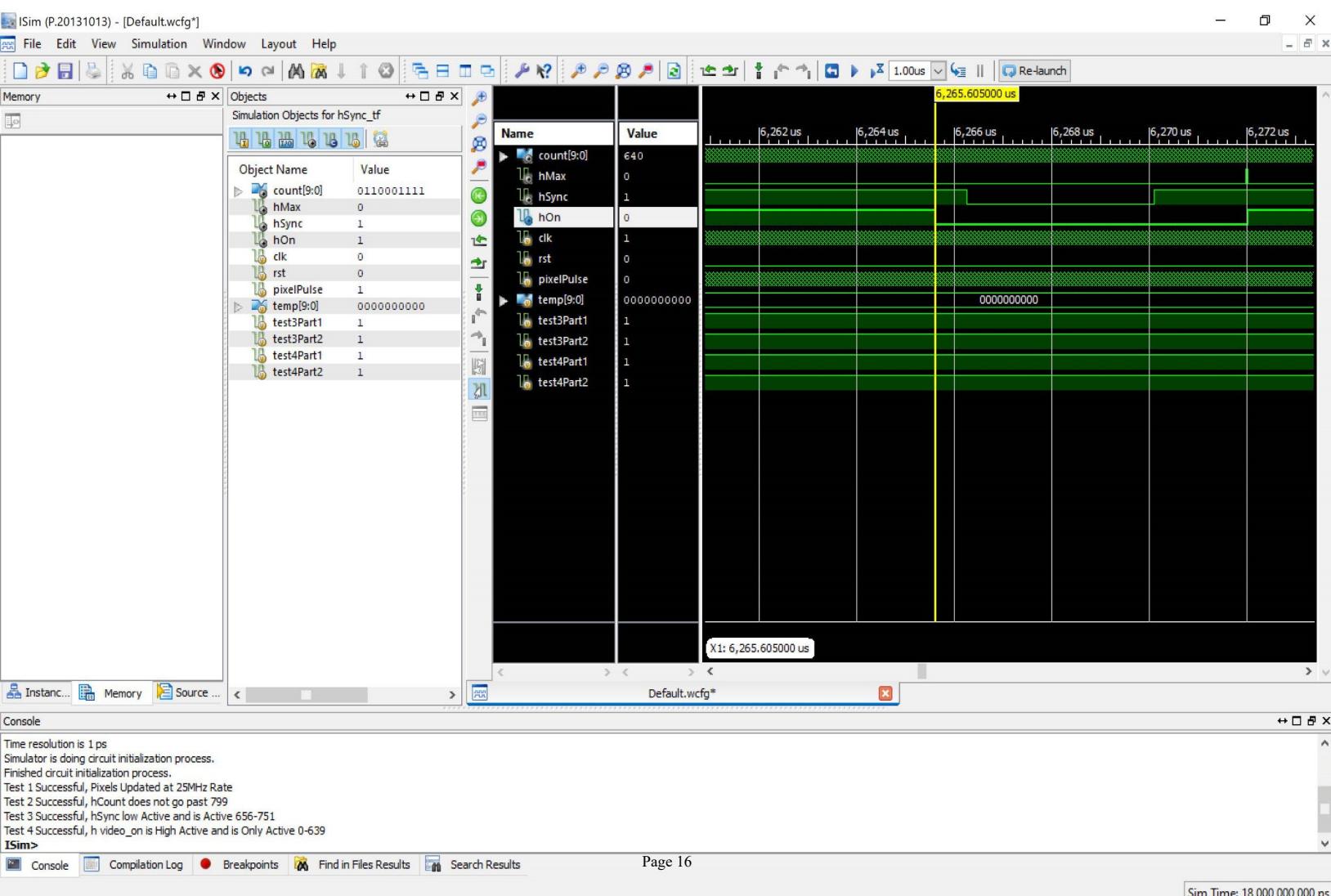
Microsoft To-Do

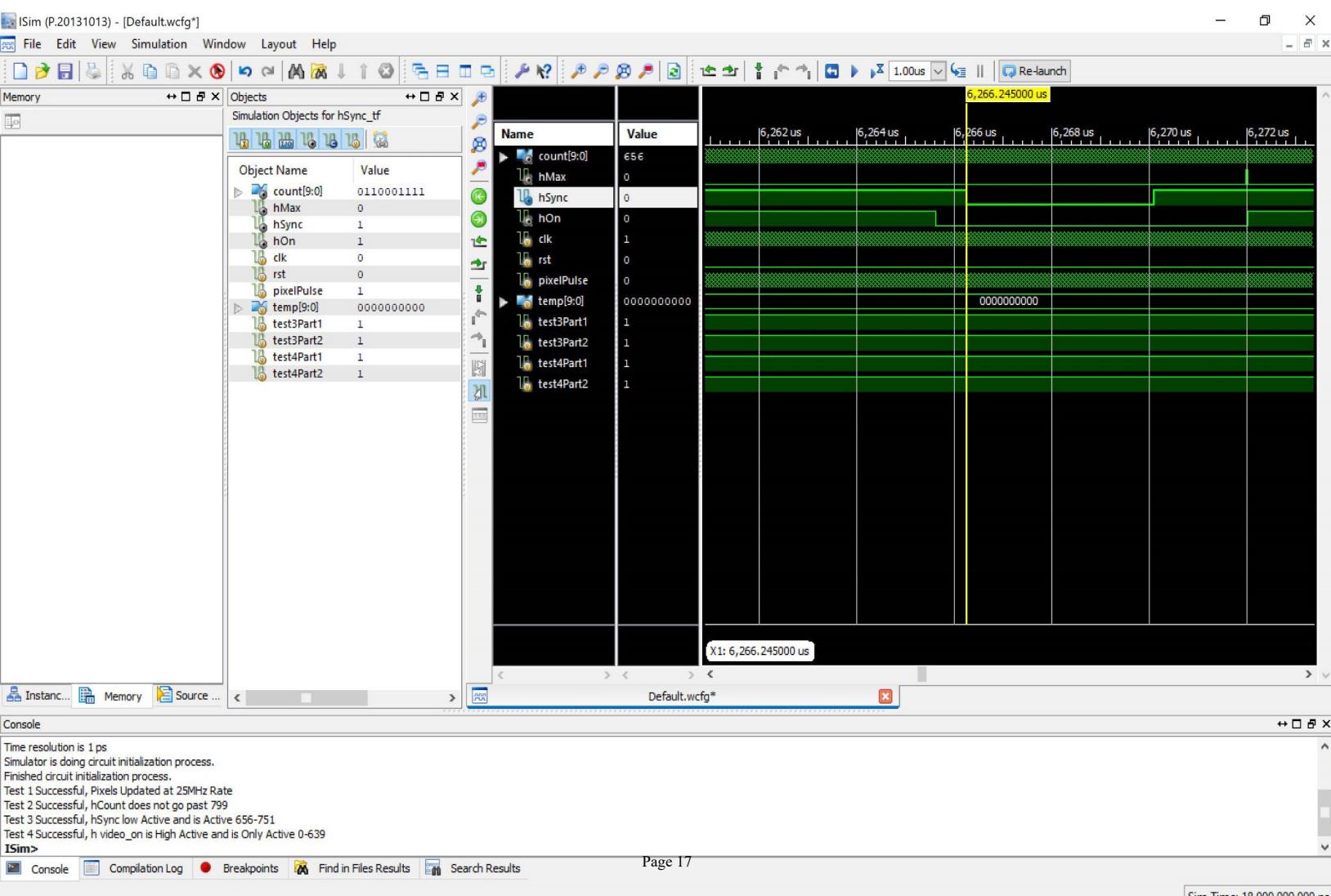
Sim Time: 18,000,000,000 ps

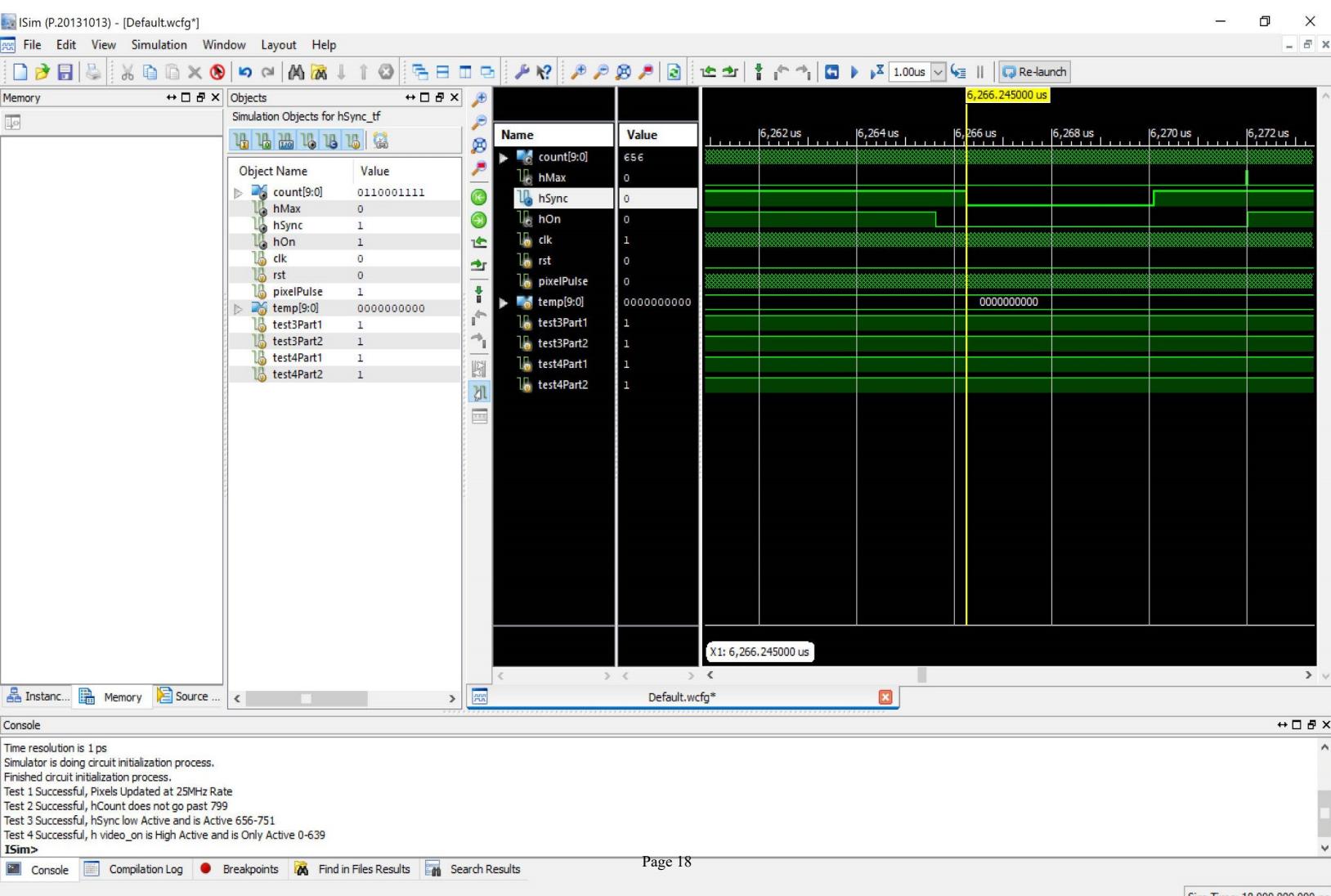
```
1 `timescale 1ns / 1ps
2 //***** File name: hSync_tf.v *****
3 //
4 //
5 // Created by      Paul Valenzuela on 10/11/19.
6 // Copyright (C) 2018      Paul Valenzuela. All rights reserved.
7 //
8 //
9 //      In submitting this file for class work at CSULB
10 //      I am confirming that this is my work and the work
11 //      of no one else. In submitting this code I acknowledge that
12 //      plagiarism in student project work is subject to dismissal
13 //      from the class
14 //***** hSync_tf.v *****
15 //*****
16 //      hSync_tf.v
17 //
18 //      The hSync test fixture block tests the capabilities of the hSync block
19 //      It ensures that hCount updates at 25MHz, and does not go past 799
20 //      In addition, it ensures hSync is low active when 656-751, and hOn
21 //      is high active when 0-639
22 //
23 //*****
24 module hSync_tf;
25
26     // Inputs
27     reg clk;
28     reg rst;
29     reg pixelPulse;
30
31     // Outputs
32     wire [9:0] count;
33     wire hMax;
34     wire hSync;
35     wire hOn;
36
37     // Instantiate the Unit Under Test (UUT)
38     hSync uut (
39         .clk(clk),
40         .rst(rst),
41         .pixelPulse(pixelPulse),
42         .count(count),
43         .hMax(hMax),
44         .hSync(hSync),
45         .hOn(hOn)
46     );
47
48     reg [9:0] temp;
49     reg test3Part1, test3Part2, test4Part1, test4Part2;
50
51     // 100 MHz Clock
52     always #5 clk = ~clk;
53
54     // 25 MHz Pulse
55     always #40 pixelPulse = 1;
56
57     // Pulse only lasts one clock cycle
```

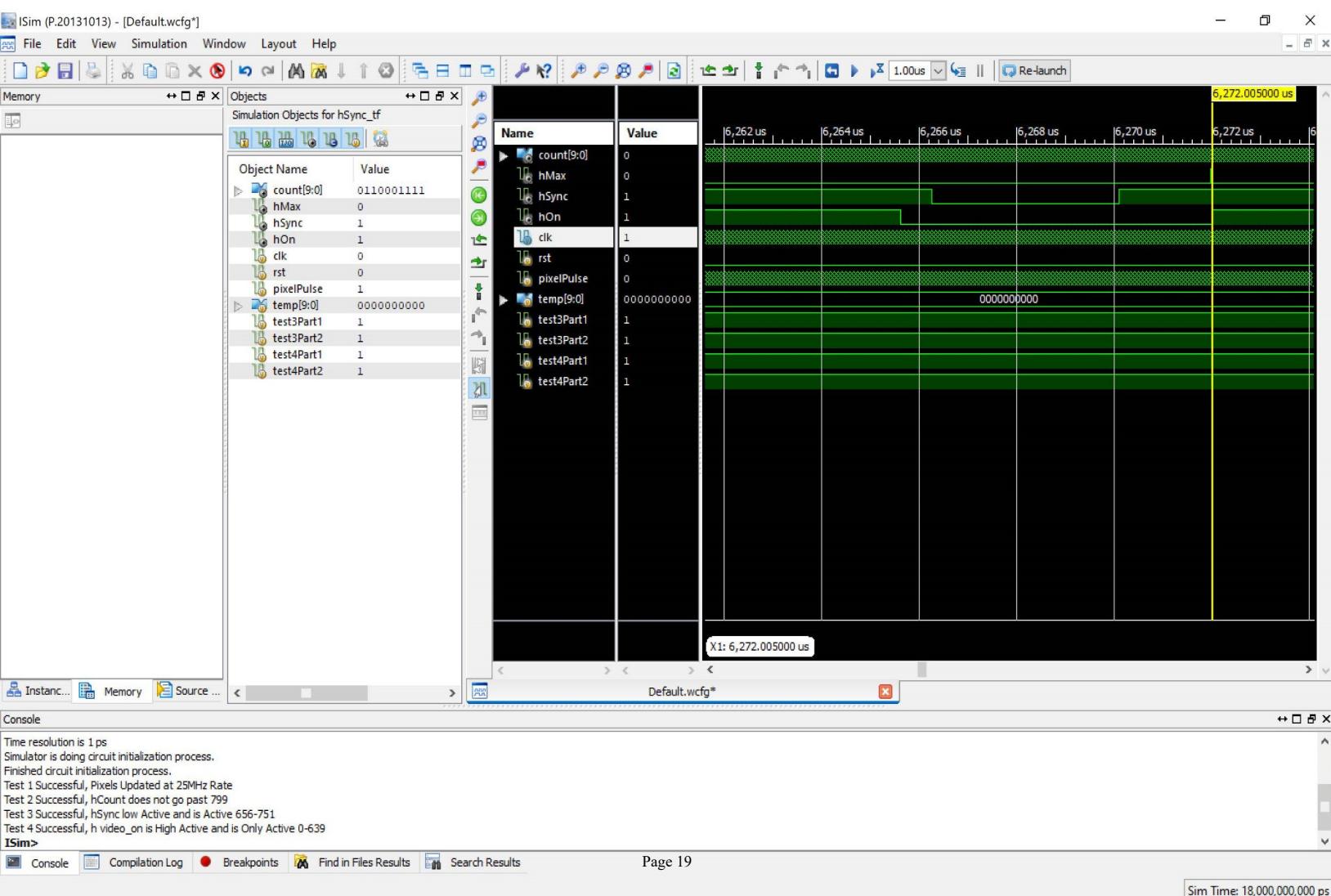
```
58     always@(posedge clk)
59         if(pixelPulse)
60             pixelPulse = 0;
61
62     // Test 3
63     // Make sure hSync is low active between 656-751
64     always@(*)
65         if(count == 656 && hSync == 0) test3Part1 = 1'b1;
66         else if(count == 751 && hSync == 0) test3Part2 = 1'b1;
67
68     // Test 4
69     // Make Sure hOn is high active 0-639
70     always@(*)
71         if(count == 0 && hOn == 1) test4Part1 = 1'b1;
72         else if(count == 640 && hOn == 0) test4Part2 = 1'b1;
73
74
75
76     initial begin
77         // Initialize Inputs
78         clk = 0;
79         rst = 1;
80         pixelPulse = 0;
81
82         // Wait 40s for global reset to finish
83         #40;
84
85         rst = 0;
86
87
88         // Test 1
89         temp = count;
90
91         #40;
92
93         if((pixelPulse == 1'b1) && (temp != count)) $display ("Test 1 Successful,
Pixels Updated at 25MHz Rate");
94
95
96
97         // Test 2
98         // 40 x 800 - 40 = 31960
99         #31960;
100        if((count < 800)) $display("Test 2 Successful, hCount does not go past 799");
101
102        // Test 3
103        if(test3Part1 && test3Part2) $display("Test 3 Successful, hSync low Active and
is Active 656-751");
104
105        // Test 4
106        if(test4Part1 && test4Part2) $display("Test 4 Successful, h video_on is High
Active and is Only Active 0-639");
107
108
109    end
110
111 endmodule
```







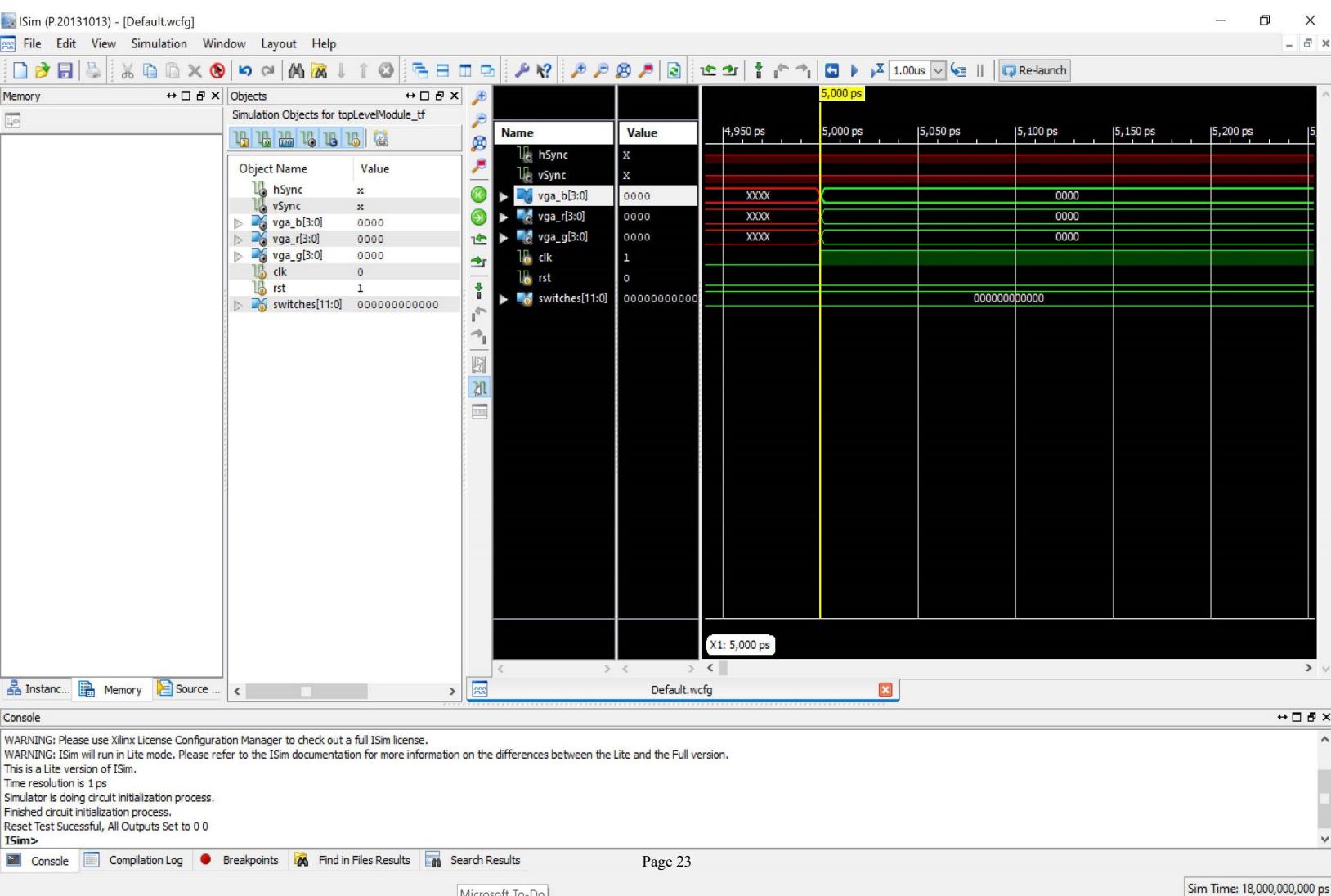


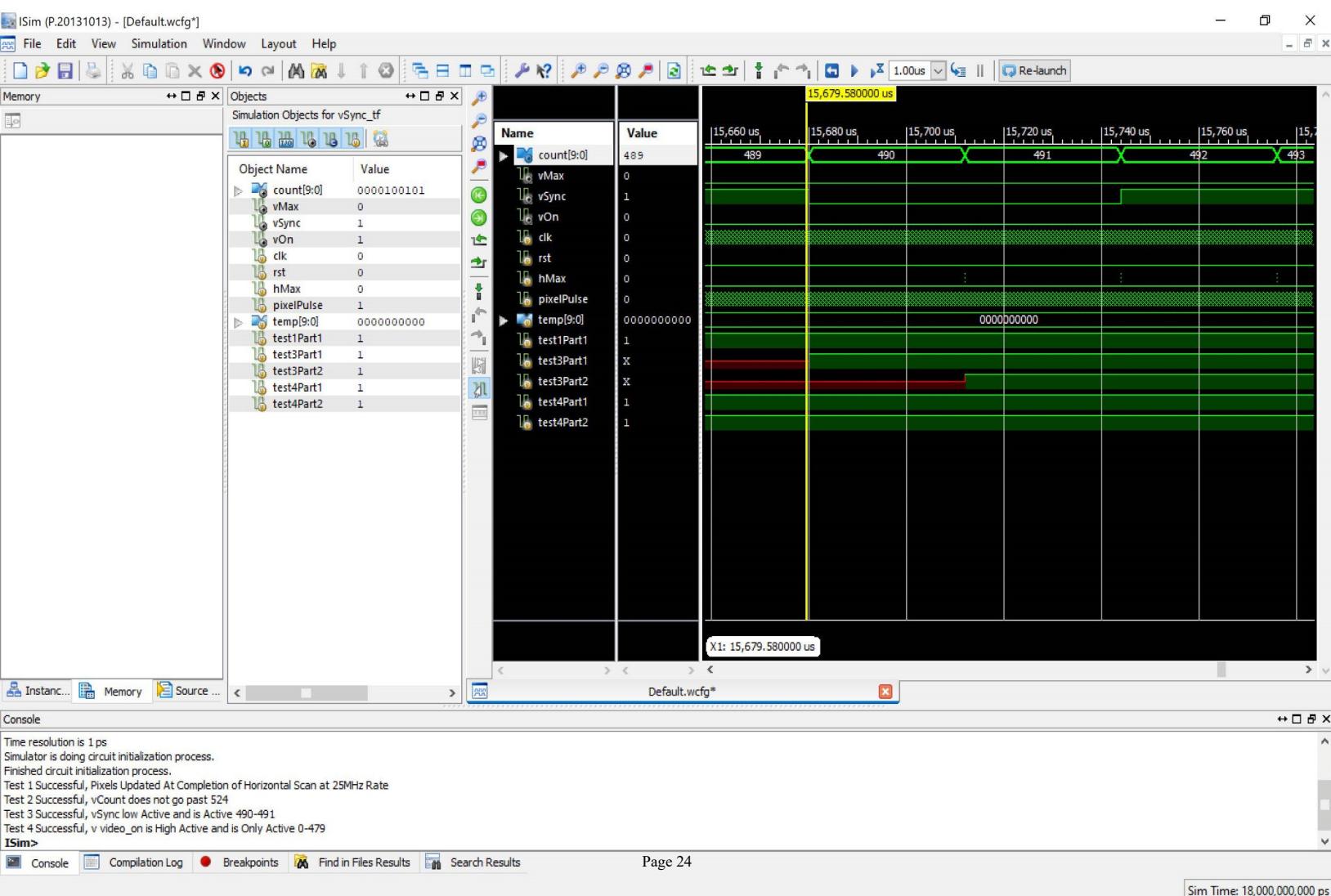


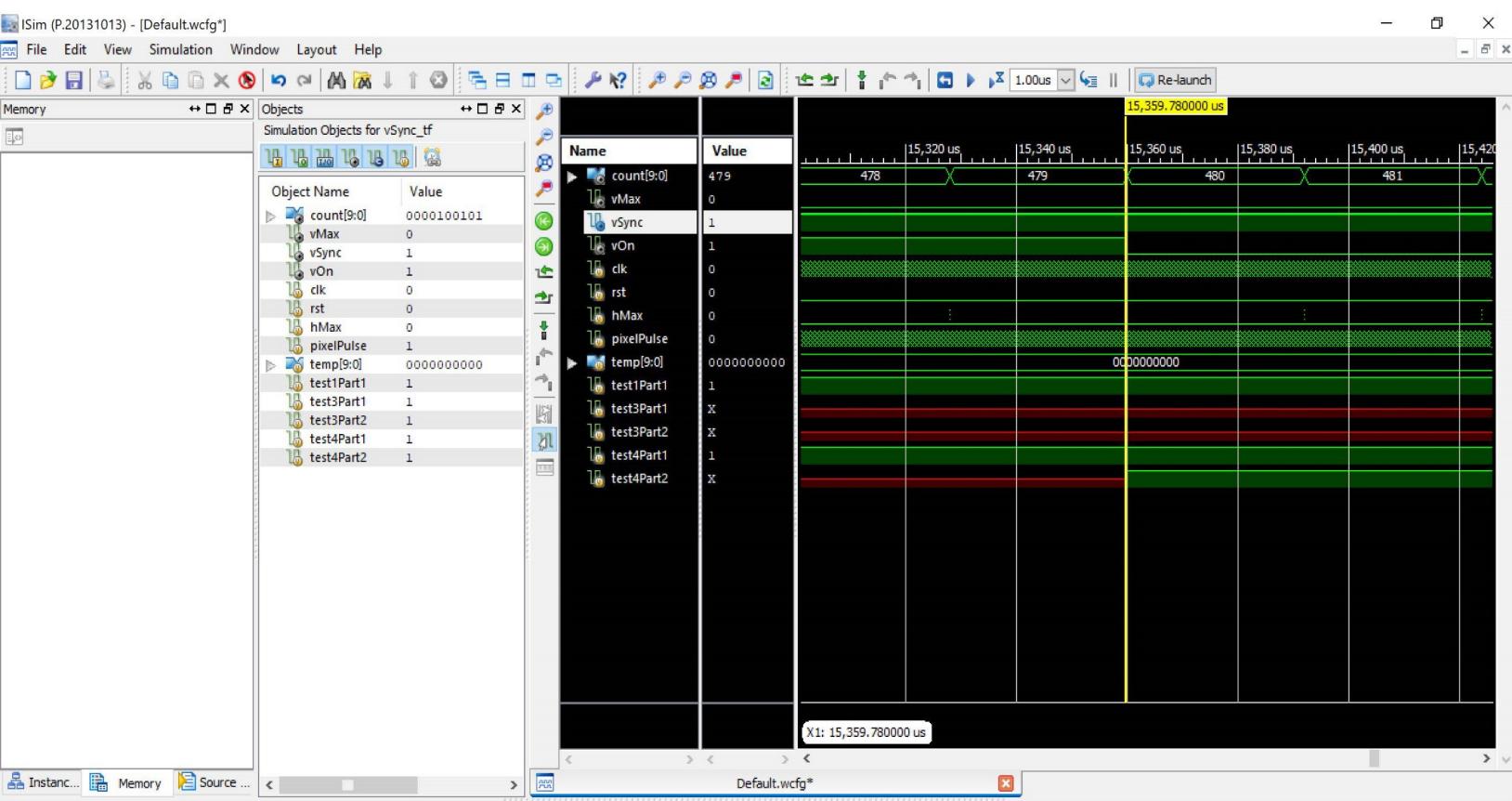
```
1 `timescale 1ns / 1ps
2 //***** File name: vSync_tf.v *****
3 //
4 //
5 // Created by      Paul Valenzuela on 10/24/19.
6 // Copyright (C) 2018      Paul Valenzuela. All rights reserved.
7 //
8 //
9 //      In submitting this file for class work at CSULB
10 //      I am confirming that this is my work and the work
11 //      of no one else. In submitting this code I acknowledge that
12 //      plagiarism in student project work is subject to dismissal
13 //      from the class
14 //***** vSync_tf.v *****
15 //*****
16 //      vSync_tf.v
17 //
18 //      The vSync test fixture block tests the capabilities of the vSync block
19 //      It ensures that hCount updates at 25MHz and hMax, and does not go past 524
20 //      In addition, it ensures hSync is low active when 490-491, and hOn
21 //      is high active when 0-479
22 //
23 //*****
24 module vSync_tf;
25
26     // Inputs
27     reg clk;
28     reg rst;
29     reg hMax;
30     reg pixelPulse;
31
32     // Outputs
33     wire [9:0] count;
34     wire vMax;
35     wire vSync;
36     wire vOn;
37
38     // Instantiate the Unit Under Test (UUT)
39     vSync uut (
40         .clk(clk),
41         .rst(rst),
42         .hMax(hMax),
43         .count(count),
44         .pixelPulse(pixelPulse),
45         .vMax(vMax),
46         .vSync(vSync),
47         .vOn(vOn)
48     );
49
50     reg [9:0] temp;
51     reg test1Part1, test3Part1, test3Part2, test4Part1, test4Part2;
52
53     // 100MHz Clock
54     always #5 clk = ~clk;
55
56     // 25MHz Pulse
57     always #40 pixelPulse = 1;
```

```
58
59      // hMax
60      always #32000 hMax = 1;
61
62      // hMax only lasts one clock cycle
63      always@(posedge clk)
64          if(hMax)
65              hMax = 0;
66
67
68      // Pulse only lasts one clock cycle
69      always@(posedge clk)
70          if(pixelPulse)
71              pixelPulse = 0;
72
73      // Test 1
74      // vCount only increments at hMax and pixelPulse
75      always@(*)
76          if(pixelPulse && hMax)
77              test1Part1 = 1'b1;
78
79      // Test 3
80      // Make sure vSync is only low active 490-491
81      always@(*)
82          if(count == 490 && vSync == 0) test3Part1 = 1'b1;
83          else if(count == 491 && vSync == 0) test3Part2 = 1'b1;
84
85      // Test 4
86      // Make sure vOn is only high active 0-479
87      always@(*)
88          if(count == 0 && vOn == 1) test4Part1 = 1'b1;
89          else if(count == 480 && vOn == 0) test4Part2 = 1'b1;
90
91
92
93      initial begin
94          // Initialize Inputs
95          clk = 0;
96          rst = 1;
97          pixelPulse = 0;
98
99          // Wait 100s for global reset to finish
100         #40;
101
102         rst = 0;
103
104
105         // Test 1
106         temp = count;
107         #32000;
108         if((test1Part1 == 1'b1) && (temp != count)) $display ("Test 1 Successful,
Pixels Updated At Completion of Horizontal Scan at 25MHz Rate");
109
110
111         // Test 2
112         // 40 * 800 * 524 = 16768000
113         #16800000;
```

```
114      if((count < 525)) $display("Test 2 Successful, vCount does not go past 524");
115
116
117      // Test 3
118      if(test3Part1 && test3Part2) $display("Test 3 Successful, vSync low Active and
119      is Active 490-491");
120
121      // Test 4
122      if(test4Part1 && test4Part2) $display("Test 4 Successful, v video_on is High
123      Active and is Only Active 0-479");
124
125 end
126
127 endmodule
```







Console

Time resolution is 1 ps
 Simulator is doing circuit initialization process.
 Finished circuit initialization process.
 Test 1 Successful, Pixels Updated At Completion of Horizontal Scan at 25MHz Rate
 Test 2 Successful, vCount does not go past 524
 Test 3 Successful, vSync low Active and is Active 490-491
 Test 4 Successful, v video_on is High Active and is Only Active 0-479

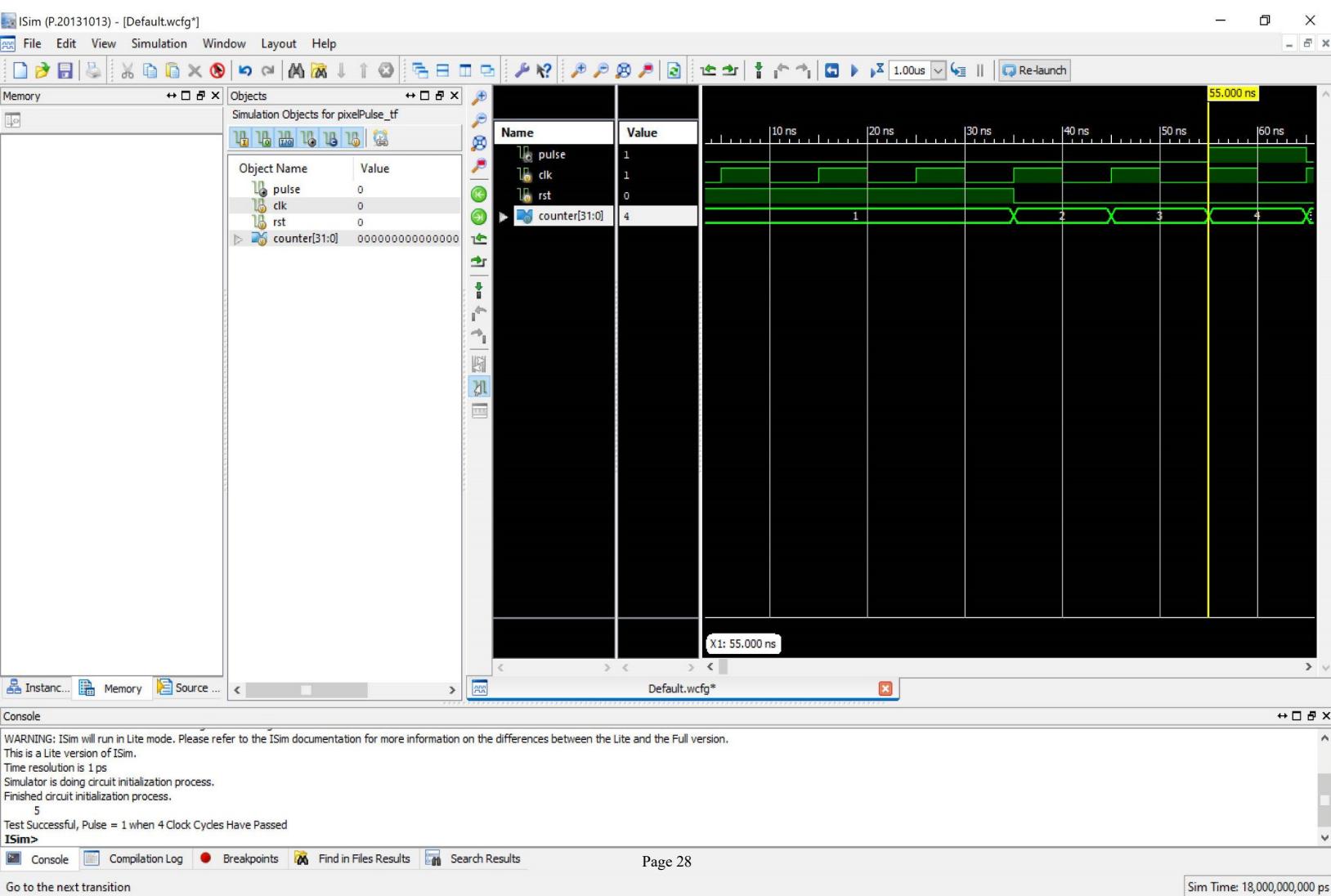
IISim>

Console Compilation Log Breakpoints Find in Files Results Search Results Page 25

Redo the previously undone action Sim Time: 18,000,000,000 ps

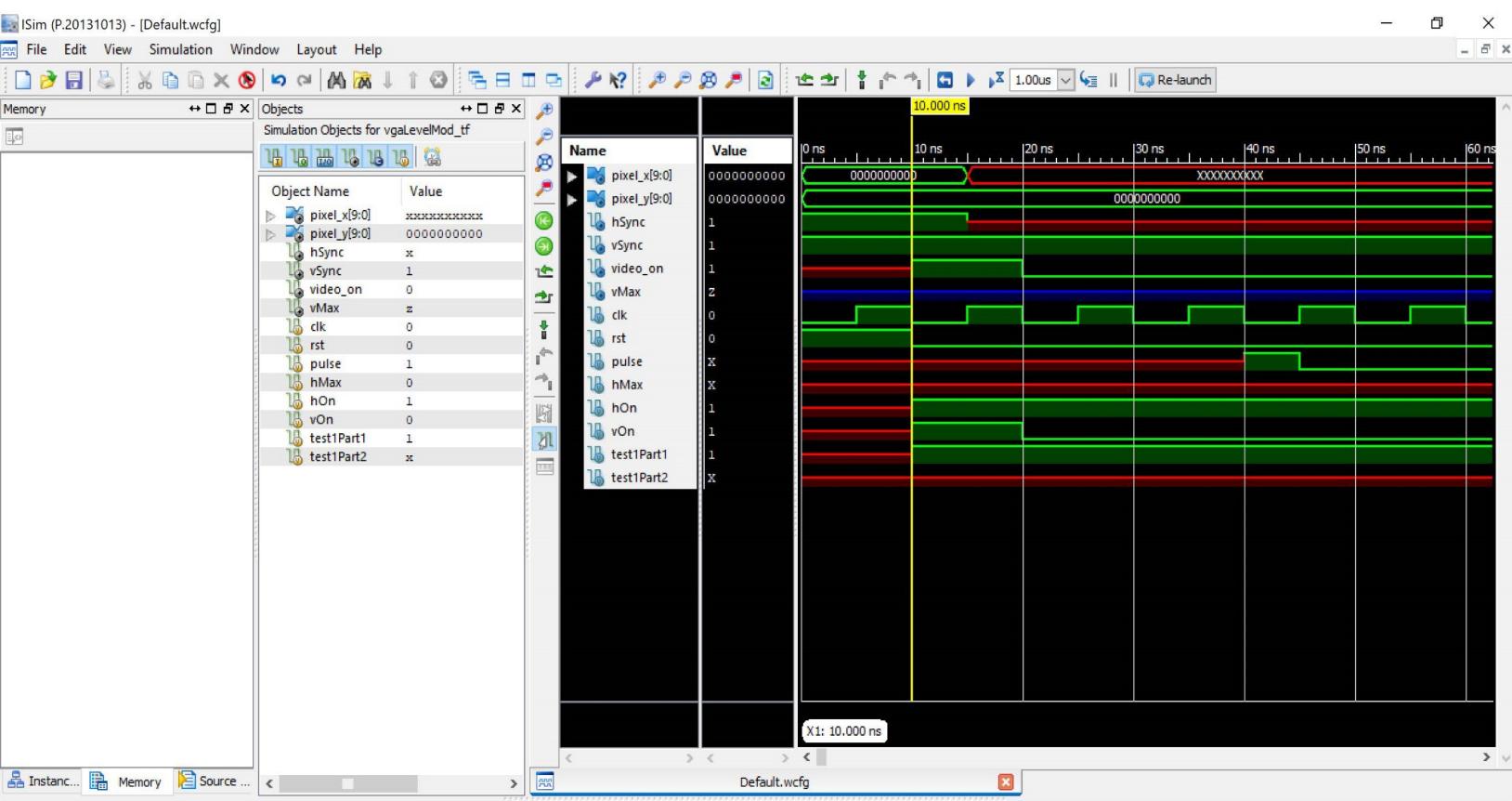
```
1  `timescale 1ns / 1ps
2  //***** File name: pixelPulse_tf.v *****
3  //
4  //
5  // Created by      Paul Valenzuela on 10/11/19.
6  // Copyright (C) 2018      Paul Valenzuela. All rights reserved.
7  //
8  //
9  //      In submitting this file for class work at CSULB
10 //      I am confirming that this is my work and the work
11 //      of no one else. In submitting this code I acknowledge that
12 //      plagiarism in student project work is subject to dismissal
13 //      from the class
14 //***** pixelGenerator_tf.v *****
15 //*****
16 //      pixelGenerator_tf.v
17 //
18 //
19 //      The pixelPulse test fixture block tests the capabilities of the
20 //      pixelPulse block. It ensures that there is a clock with a 25Mhz signal
21 //*****
22 module pixelPulse_tf;
23
24     // Inputs
25     reg clk;
26     reg rst;
27
28     // Outputs
29     wire pulse;
30
31     // Instantiate the Unit Under Test (UUT)
32     pixelPulse uut (
33         .clk(clk),
34         .rst(rst),
35         .pulse(pulse)
36     );
37
38     // 100Mhz Clock
39     always #5 clk = ~clk;
40
41     integer counter = 1;
42
43     initial begin
44         // Initialize Inputs
45         clk = 0;
46         rst = 1;
47
48         // Wait 35 ns for global reset to finish
49         #35;
50
51         rst = 0;
52
53         // Allow 4 Clock Cycles To Pass, See if Pulse Is On
54         repeat(4) @(posedge clk) counter = counter + 1;
55         $display(counter);
56         if((pulse == 1'b1) && (counter == 4)) $display("Test Successful, Pulse = 1 when
```

```
        4 Clock Cycles Have Passed");  
57  
58  
59      end  
60  
61  endmodule  
62  
63
```



```
1  `timescale 1ns / 1ps
2  //***** File name: vgaLevelMod_tf.v *****
3  //
4  //
5  // Created by      Paul Valenzuela on 10/11/19.
6  // Copyright (C) 2018      Paul Valenzuela. All rights reserved.
7  //
8  //
9  //      In submitting this file for class work at CSULB
10 //      I am confirming that this is my work and the work
11 //      of no one else. In submitting this code I acknowledge that
12 //      plagiarism in student project work is subject to dismissal
13 //      from the class
14 //***** vgaLevelMod_tf.v *****
15 //*****
16 //      vgaLevelMod_tf.v
17 //
18 //      The vgaLevelMod test fixture block tests the capabilities of the
19 //      vgaLevelMod. It ensures that video_on is only active when h_on and vOn
20 //      are active
21 //
22 //*****
23 module vgaLevelMod_tf;
24
25     // Inputs
26     reg clk;
27     reg rst;
28     reg pulse;
29
30     // Outputs
31     wire [9:0] pixel_x;
32     wire [9:0] pixel_y;
33     wire hSync;
34     wire vSync;
35     wire video_on;
36     reg hMax;
37     reg hOn;
38     reg vOn;
39     wire vMax;
40
41     // hMax
42     always #32000 hMax = 1;
43
44     // hMax only lasts one clock cycle
45     always@ (posedge clk)
46         if (hMax)
47             hMax = 0;
48
49     reg test1Part1, test1Part2;
50
51     // 100Mhz Clock
52     always #5 clk=~clk;
53
54     // 25MHz Pulse
55     always #40 pulse = 1;
56
57     // Pulse only lasts one clock cycle
```

```
58      always @ (posedge clk)
59          if(pulse)
60              pulse = 0;
61
62      always @(*)
63          if(video_on)
64              test1Part1 = 1;
65
66
67 // Instantiate the Unit Under Test (UUT)
68 vgaLevelMod uut (
69     .clk(clk),
70     .rst(rst),
71     .pixel_x(pixel_x),
72     .pixel_y(pixel_y),
73     .hSync(hSync),
74     .vSync(vSync),
75     .pulse(pulse),
76     .video_on(video_on),
77     .vvOn(vOn),
78     .hhOn(hOn)
79 );
80
81
82 initial begin
83
84
85     // Initialize Inputs
86     clk = 0;
87     rst = 1;
88
89     // Wait 10 ns for global reset to finish
90     #10;
91
92     rst = 0;
93     hOn = 1;
94     vOn = 1;
95
96     #10
97
98     vOn = 0;
99
100
101
102     //Test 1
103     if(test1Part1) $display("Test 1 Successful, Video On Is Only On When hOn and
104     vOn are Active");
105
106 end
107
108 endmodule
109
110
```



Console

WARNING: Please use Xilinx License Configuration Manager to check out a full IISim license.
 WARNING: IISim will run in Lite mode. Please refer to the IISim documentation for more information on the differences between the Lite and the Full version.
 This is a Lite version of IISim.
 Time resolution is 1 ps
 Simulator is doing circuit initialization process.
 Finished circuit initialization process.
 Test 1 Successful, Video On is Only On When hOn and vOn are Active
IISim>

Console Compilation Log Breakpoints Find in Files Results Search Results Page 31 Sim Time: 18,000,000,000 ps

```
1 `timescale 1ns / 1ps
2 //*****//*****
3 // File name: topLevelMod.v ////
4 //
5 // Created by      Paul Valenzuela on 10/16/19. ////
6 // Copyright (C) 2018      Paul Valenzuela. All rights reserved. ////
7 //
8 //
9 //      In submitting this file for class work at CSULB ////
10 // I am confirming that this is my work and the work ////
11 // of no one else. In submitting this code I acknowledge that ////
12 // plagiarism in student project work is subject to dismissal ////
13 // from the class ////
14 //*****//*****
15 //*****//*****
16 //          topLevelMod.v ////
17 //
18 //      The topLevelMod block instantiates all the other blocks and connects ////
19 // them in a manner so that they all contribute to the top level design. ////
20 // Each component is essential and is instantiated at a specific moment in ////
21 // the execution process ////
22 //
23 // @input    clk, rst, switches[11:0] ////
24 // @output   [3:0] vga_r, [3:0] vga_g, [3:0] vga_b, hSync, vSync ////
25 //
26 //*****//*****
27 module topLevelModule(clk,rst,switches,hSync,vSync, vga_b, vga_r, vga_g, topButton,
bottomButton);
28
29     // Input From Nexys A7
30     input wire clk, rst, topButton, bottomButton;
31     input wire [11:0] switches;
32
33     // Output That Designates Which Color To Output
34     output wire [3:0] vga_r, vga_g, vga_b;
35
36     // Ouput That Tells Pixels When To Update
37     output wire hSync, vSync;
38
39     // Wires That Connect Instantiations
40     wire mRst, pulse, video_on;
41     wire [9:0] pixel_x, pixel_y;
42
43     // Synchronized Reset
44     AISO mReset(.clk(clk), .rst(rst), .mRst(mRst));
45
46     // 25MHz Clock Signal
47     pixelPulse pulser (.clk(clk), .rst(mRst), .pulse(pulse));
48
49     // VSync Block Used To Synchronize Pixels
50     vgaLevelMod vgaConnect(.clk(clk), .rst(mRst), .pixel_x(pixel_x[9:0]),
51         .pixel_y(pixel_y[9:0]), .pulse(pulse), .hSync(hSync), .vSync(vSync),
52         .video_on(video_on));
53
54     // PixelGenerator Block That Designates Color Assignment
55     pixelGenerator generateVGA(.clk(clk), .rst(mRst), .pulse(pulse),
56         .pixel_x(pixel_x[9:0]), .pixel_y(pixel_y[9:0]),
```

```
57      .switches(switches[11:0]), .video_on(video_on),  
58      .rgb({vga_r[3:0],vga_g[3:0],vga_b[3:0]}), .topButton(topButton), .bottomButton(  
bottomButton));  
59  
60 endmodule  
61
```