```
1   `timescale 1ns / 1ps
2   /*********************************************************************
3    * File Name: vga_controller_tb.v
4    * Project: VGA Object Mapped
5    * Designer: Marc Cabote
6    * Email: marcdominic011@gmail.com
7    * Rev. Date: 11 October, 2017
8    *
9    * Purpose: The purpose of this module is to test if the the vga top
10   *          will behave the way it is expected to behave. From this
11   *          we will see that proper rgb values will be set depending
12   *          on the pixel location (x or y).
13   *
14   * Notes: - 25 Mhz Clock, Vcount and Hcount were copied from top level
15   *          to sync the clocks
16   *        - remove comments on display to see where the error is if
17   *          error counter is not zero
18   *        - if error counter stays zero then the top level is verified
19   *********************************************************************/
20
21  module vga_controller_tb;
22     // Inputs
23     reg clk;
24     reg rst;
25
26     // Outputs
27     wire hsync;
28     wire vsync;
29     wire [11:0] rgb;
30
31     //Signals
32     wire [9:0] pixel_x, pixel_y;
33     wire video_on;
34
35     //Variables
36     integer errorCount = 0;//counter for error
37
38     /*********************************
39     *Counter to generate 25Mhz clock
40     *********************************/
41     reg [1:0] count;
42     wire tick;
43
44     assign tick = (count == 2'b11);
45
46     always @ (posedge clk, posedge rst)
47        if (rst)  count <= 2'b0; else
48        if (tick) count <= 2'b0; else
49                  count <= count + 2'b1;
50
51
52     /*********************************
53     *Horizontal count 0-799
54     *********************************/
55     reg [9:0] hcount;
56     wire endh;
57
```

```
58       assign pixel_x = hcount;
59       assign endh = (hcount == 799);
60
61       always @ (posedge clk, posedge rst)
62          if (rst) hcount <= 10'b0; else
63          if (tick)
64             if (endh) hcount <= 10'b0; else
65                       hcount <= hcount +10'b1;
66
67       assign hsync = ~(hcount >= 656 & hcount <= 751);
68
69       /*********************************
70       *Vertical count 0-524
71       *********************************/
72       reg [9:0] vcount;
73       wire endv;
74
75       assign pixel_y = vcount;
76       assign endv = (vcount == 524);
77
78       always @ (posedge clk, posedge rst)
79          if (rst) vcount <= 10'b0; else
80          if (tick)
81             if(endh)
82                if(endv) vcount <= 10'b0; else
83                         vcount <= vcount + 10'b1;
84
85       assign vsync = ~(vcount >= 490 & vcount <= 491);
86
87       assign video_on = ((hcount<656) && (vcount<490));
88       // Instantiate the Unit Under Test (UUT)
89       vga_controller uut (
90          .clk(clk),
91          .rst(rst),
92          .hsync(hsync),
93          .vsync(vsync),
94          .rgb(rgb)
95       );
96
97       always #1 clk = ~clk;
98
99       initial begin
100         // Initialize Inputs
101         clk = 0;
102         rst = 1;//check reset
103
104         // Wait 100 ns for global reset to finish
105         #1;
106         rst = 0;
107
108         // Add stimulus here
109
110      end
111
112      always @ (posedge clk, posedge rst)
113      begin
114
```

```
115         if (tick && video_on) begin
116             //Verify Wall
117             if (pixel_x >= 32 && pixel_x <= 35
118                  && (rgb != 12'hF00))
119                 //$display ("Error Wall");
120                 errorCount = errorCount + 1;
121 
122             //Verify Bar
123             else if ((pixel_x >= 600) && (pixel_x <= 603)
124                  && (pixel_y >= 204)  && (pixel_y <= 276)
125                  && (rgb != 12'h0F0))
126                  //$display ("Error Bar");
127                  errorCount = errorCount + 1;
128 
129             //Verify Ball
130             else if ((pixel_x >= 580) && (pixel_x <= 588)
131                   && (pixel_y >= 238) && (pixel_y <= 246)
132                   && (rgb != 12'h00F))
133                  //$display ("Error Ball");
134                   errorCount = errorCount + 1;
135 
136             //Verify Background
137             else if (!(pixel_x >= 32) && !(pixel_x <= 35)
138                  && !(pixel_x >= 600) && !(pixel_x <= 603)
139                  && !(pixel_y >= 204) && !(pixel_y <= 276)
140                  && !(pixel_x >= 580) && !(pixel_x <= 588)
141                  && !(pixel_y >= 238) && !(pixel_y <= 246)
142                  && (rgb != 12'h000))
143                  //$display ("Error BG");
144                  errorCount = errorCount + 1;
145 
146            //Else display error
147            else if (errorCount != 0)
148               $display ("Error Count: ",errorCount);
149         end
150 
151         if (!(video_on) && !(rgb == 12'h00))
152            //$display ("Error video");
153            errorCount = errorCount + 1;
154         else if (errorCount != 0)
155            $display ("Error Count: ",errorCount);
156 
157       end
158 
159   endmodule
160 
161 
```