```
1   `timescale 1ns / 1ps
2   /*********************************************************************************
3   *
4   * Engineer: Justin Maeder; 015906629
5   * Email:    Justin_maeder@hotmail.com
6   * Filename: VGA_controller_tf.v
7   * Date:     May 5, 2019
8   * Version:  14.7
9   * Description: This is a self-checking test bench. All inputs are registers, all
10  *              outputs are wires. First, I instantiated the VGA_controller which is
11  *              the design under test or DUT. I then imported signals as wires from
12  *              lower level modules. Then I created a clock. I set the inputs clock
13  *              and reset and deactived reset to 0 after 10 nS. The purpose of
14  *              this test bech is to go through the given requirements and print any
15  *              discrepancies that might have occured in the design.
16  *
17  *********************************************************************************/
18
19  module VGA_controller_tf;
20
21     // Inputs
22     reg clk;
23     reg reset;
24     reg btn_up, btn_down;
25
26     // Outputs
27     wire hsync;
28     wire vsync;
29     wire [11:0] rgb;
30
31     // Pullout signals from ticker module
32     wire tick = pong.vsync_unit.tick0.tick;
33     wire [1:0] count = pong.vsync_unit.tick0.count;
34     // Pullout signals from vsync_unit module
35     wire [9:0] pixel_x = pong.vsync_unit.pixel_x, pixel_y = pong.vsync_unit.pixel_y;
36     wire [9:0] h_count = pong.vsync_unit.h_count_reg;
37     wire [9:0] v_count = pong.vsync_unit.v_count_reg;
38     wire h_video = pong.vsync_unit.h_video, v_video = pong.vsync_unit.v_video,
39          h_end = pong.vsync_unit.h_end, v_end = pong.vsync_unit.v_end,
40          video_on = pong.vsync_unit.video_on;
41     // Pullout signals from pong_grf_unit
42     wire [11:0] graph_rgb = pong.pong0.graph_rgb;
43     wire wall_on = pong.pong0.wall_on,
44           bar_on = pong.pong0.bar_on,
45           sq_ball_on = pong.pong0.sq_ball_on;
46
47     // Instantiate the Design Under Test (DUT)
48     VGA_controller pong(.clk(clk),
49                         .reset(reset),
50                         .hsync(hsync),
51                         .vsync(vsync),
52                         .rgb(rgb),
53                         .btn_up(btn_up),
54                         .btn_down(btn_down));
55
56     // 1. Instantiate a clock
57     always #5 clk = ~clk;
```

```
58
59      initial begin
60         // Initialize Inputs
61         clk = 0;
62         reset = 1;
63
64         // Wait 10 ns for global reset to finish
65         #15;
66         reset = 0;
67      end
68
69      // Add stimulus here
70      always @ (posedge clk, posedge reset) begin
71
72         // Requirement  1: Reset shall bring the VGA Synchronization circuit to a
73         //                 known state with all outputs inactive.
74         if( reset )
75            if ( (hsync === 1'b0) || (vsync === 1'b0) || (rgb != 12'b0) ) begin
76               $display("Reset does not bring circuit to a known state");
77               $stop;
78               end
79
80         // Requirement  2: The VGA synchronization Logic shall be updated at a 25
81         //                 MHz rate.
82         if( (tick === 1'b1) && (count !== 2'd3) ) begin
83            $display("VGA was not updated at 25MHz");
84            $stop;
85         end
86
87         // Requirement  3: The Horizontal Scan Count shall be updated at the 25 MHz
88         //                 rate.
89         if( (tick === 1'b1) && (h_count === h_count + 10'b1) ) begin
90            $display("Horizontal Scan Count does not update at the 25 MHz rate");
91            $stop;
92         end
93
94         // Requirement  4: The Horizontal Scan Count shall range from 0 to 799.
95         if( !( (h_count >= 10'd0) && (h_count <= 10'd799) ) ) begin
96            $display("%d is not within the Horizontal Scan Count 0 - 799",h_count);
97            $stop;
98         end
99
100        // Requirement  5: The Horizontal Sync signal shall be LOW ACTIVE and shall
101        //                 be active from Horizontal Scan Count 656 through 751.
102        if( (hsync === 1'b1) && (h_count >= 10'd656) && (h_count <= 10'd751) ) begin
103              $display("Horizontal Sync Signal was not LOW active from Horizontal");
104              $display("Scan Count 656 - 751 --> %d h_count is the problem",h_count);
105              $stop;
106        end
107
108        // Requirement  6: The Horizontal Video On signal shall be HIGH ACTIVE and
109        //                 shall be active from Horizontal Scan Count 0 through 639.
110        if( (h_count >= 10'd0) && (h_count <= 10'd639) && (h_video === 1'b0) ) begin
111           $display("Horizontal Video On signal not active on scan count %d",h_count);
112           $stop;
113        end
114
```

```
115         // Requirement  7: The Vertical Scan Count shall be updated at the
116         //                 completion of a Horizontal Scan.
117         if( (tick === 1'b1) && (h_end === 1'b1) && (v_count ==  v_count + 1'b1) )begin
118            $display("Vertical Scan count was not updated after Horizontal scan");
119            $stop;
120         end
121
122         // Requirement  8: The Vertical Scan Count shall range from 0 to 524.
123         if( !((v_count >= 10'd0) && (v_count <= 10'd524)) ) begin
124            $display("Vertical Scan Count is not in range, count = %d", v_count);
125            $stop;
126         end
127
128         // Requirement  9: The Vertical Sync signal shall be LOW ACTIVE and shall be
129         //                 active from Vertical Scan Count 490 through 491.
130         if( (v_count >= 10'd490) && (v_count <= 10'd491) && (vsync === 1'b1) ) begin
131            $display("Vertical Sync is active incorrectly = %d", v_count);
132            $stop;
133         end
134
135         // Requirement 10: The Vertical Video On signal shall be HIGH ACTIVE and
136         //                 shall be active from Vertical Scan Count 0 through 479.
137         if( (v_count >= 10'd0) && (v_count <= 10'd479) && (v_video !== 1'b1) ) begin
138            $display("Vertical video on signal is active incorrectly = %d",v_count);
139            $stop;
140         end
141
142         // Requirement 11: The Video On signal shall be HIGH ACTIVE and shall be
143         //                 active when Horizontal Video On and Vertical Video On
144         //                 are active at the same time.
145         if( (h_video === 1'b1) && (v_video === 1'b1) && (video_on !== 1'b1) ) begin
146            $display("Video On signal is not active when vertical and horizontal on");
147            $stop;
148         end
149
150         // Requirement 12: The RGB signals shall be driven while the Video On
151         //                 signal is ACTIVE. When the Video On signal is INACTIVE
152         //                 the RGB signals shall be held at 0.
153         if( (video_on === 1'b1 && graph_rgb === 12'h000) ||
154             (video_on === 1'b0 && graph_rgb !== 12'h000) ) begin
155            $display("Video On and RGB signals are incorrectly driven");
156            $stop;
157         end
158
159         // ____________________________OBJECT MAPPING REQUIREMENTS_____________________________
160
161         // WALL requirements for placement and rgb
162         if( (pixel_x >= 10'd32) && (pixel_x <= 10'd35) &&
163             (pixel_y >= 10'd0) && (pixel_y <= 10'd480)  &&
164             (graph_rgb === 12'h00F) && (wall_on !== 1'b1) ) begin
165            $display("Wall was not properly displayed");
166            $stop;
167         end
168
169         // BAR requirements for placement and rgb
170         if( (pixel_x >= 10'd600) && (pixel_x <= 10'd603) &&
171             (graph_rgb === 12'hF00) && (bar_on !== 1'b1) ) begin
```

```
172             $display("Bar was not properly displayed");
173             $display("%d %d %d", pixel_x, pixel_y, bar_on);
174             $stop;
175         end
176 
177         // BALL requirements for placement and rgb
178         if( (graph_rgb === 12'h777) && (sq_ball_on !== 1'b1) ) begin
179             $display("Ball was not properly displayed");
180             $stop;
181         end
182       end // end always @
183   endmodule
```