```
1   `timescale 1ns / 1ps
2   /**********************************************************************************
3   *
4   * Engineer: Justin Maeder; 015906629
5   * Email:    Justin_maeder@hotmail.com
6   * Filename: vga_sync.v
7   * Date:     April 4, 2019
8   * Version:  14.7
9   * Description: The Video Graphics Array (VGA) syncronization circuit outputs a hsync
10  *              and vsync signals which are the time required to transverse a row and
11  *              transverse the entire screen respectively. For this project we
12  *              implemented 640-by-480 VGA screen with a 25MHz pixel rate. This means
13  *              that 25M pixels are processed in one second. A ticker module was
14  *              implemented to take our on-board clock of 100MHz and create a 25MHz
15  *              pulse. Outside of the 640-by-480 visable screen, there are borders
16  *              to account for which create a 800-by-525 horizontal-by-vertical scan.
17  *              The video-on output indicates if the screen is on and is active when
18  *              horizontal and vertical video_on signals are active. pixel_x and
19  *              pixel_y are output wires that are used to display the static objects.
20  *
21  **********************************************************************************/
22  module vga_sync(clk, reset, hsync, vsync, video_on, pixel_x, pixel_y, p_tick);
23
24     // constant declaration
25     // VGA 640-by-480 sync parameters
26     localparam
27        HD = 640, // horizontal display area
28        HF = 48 , // h. front (left) border
29        HB = 16 , // h. back (right) border
30        HR = 96 , // h. retrace
31        VD = 480, // vertical display area
32        VF = 10 , // v. front (top) border
33        VB = 33 , // v. back (bottom) border
34        VR = 2  ; // v. retrace
35
36     // Input & Output Declarations
37     input wire clk, reset;
38     output wire hsync, vsync, video_on, p_tick;
39     output wire [9:0] pixel_x, pixel_y;
40     // Sync counters
41     reg [9:0] h_count_reg, h_count_next;
42     reg [9:0] v_count_reg, v_count_next;
43
44     // Output buffers
45     reg  v_sync_reg,  h_sync_reg;
46     wire v_sync_next, h_sync_next;
47
48     // video on signals
49     wire h_video, v_video;
50
51     // Status signal
52     wire h_end, v_end, pixel_tick;
53
54     // Instantiate ticker
55     ticker tick0(.clk(clk), .reset(reset),.k(2'd3), .tick(pixel_tick));
56
57     // Registers
```

```
58      always @(posedge clk, posedge reset)
59         if (reset) begin
60            h_count_reg <= 10'b0;
61            v_count_reg <= 10'b0;
62            h_sync_reg  <= 1'b0;
63            v_sync_reg  <= 1'b0;
64            end
65         else begin
66            h_count_reg <= h_count_next;
67            v_count_reg <= v_count_next;
68            h_sync_reg  <= h_sync_next;
69            v_sync_reg  <= v_sync_next;
70            end
71
72      // status signals
73      // end of horizontal counter (799)
74      assign h_end = (h_count_reg==(HD+HF+HB+HR-1)) ;
75      // end of vertical counter (524)
76      assign v_end = (v_count_reg==(VD+VF+VB+VR-1)) ;
77
78      // next-state logic of mod-800 horizontal sync counter
79      always @(*)
80         if (pixel_tick) // 25 MHz pulse
81            if (h_end) h_count_next = 10'b0;
82            else       h_count_next = h_count_reg + 1'b1;
83         else h_count_next = h_count_reg;
84
85      // next-state logic of mod-525 vertical sync counter
86      always @(*)
87         if (pixel_tick & h_end)
88            if (v_end) v_count_next = 10'b0;
89            else v_count_next = v_count_reg + 1'b1;
90         else v_count_next = v_count_reg;
91
92      // horizontal and vertical sync, buffered to avoid glitch
93      // h-svnc-next asserted between 656 and 751
94      assign h_sync_next = (h_count_next>=(HD+HB) && h_count_next<=(HD+HB+HR-1));
95      // vh-sync-next asserted between 490 and 491
96      assign v_sync_next = (v_count_next>=(VD+VF) && v_count_next<=(VD+VF+VR-1));
97
98      // Horizontal Video On, HIGH ACTIVE when horizontal scan 0 through 639
99      assign h_video = (h_count_reg>=10'd0 && h_count_reg <= (HD-1));
100
101     // Vertical Video On, HIGH ACTIVE when vertical count scan 0 through 479
102     assign v_video = (v_count_reg>=10'd0 && v_count_reg <= (VD-1));
103
104     // Video on/off
105     assign video_on = h_video && v_video;
106     // output
107     assign hsync = ~h_sync_reg;
108     assign vsync = ~v_sync_reg;
109     assign pixel_x = h_count_reg;
110     assign pixel_y = v_count_reg;
111     assign p_tick = pixel_tick;
112  endmodule
```