```
1   `timescale 1ns / 1ps
2   /**********************************************************************************
3   *
4   * Engineer: Justin Maeder; 015906629
5   * Email:    Justin_maeder@hotmail.com
6   * Filename: VGA_controller.v
7   * Date:     May 5, 2019
8   * Version:  14.7
9   * Description: The purpose of this top-level VGA controller is to put together the
10  *              three modules AISO, VGA_sync, and pong_graph_st. On reset, the AISO
11  *              module creates a asyncronous-in synchronous-out signal which provides
12  *              a synchronous reset to all flops in the VGA controller to avoid
13  *              violating timing constraints (metastability). The Nexys4 DDR has a
14  *              clock, reset, btn_up, and btn_down input. And the VGA controller
15  *              outputs the hsync and vsync from the vga_sync module, and the 12-bit
16  *              rgb signals from the pixel generation module. The VGA controller
17  *              outputs are then interfaced with the VGA monitor.
18  *
19  **********************************************************************************/
20  module VGA_controller(clk, reset, hsync, vsync, rgb, btn_up, btn_down);
21     input wire clk, reset, btn_up, btn_down;
22     output wire hsync, vsync;
23     output wire [11:0] rgb;
24
25     wire [9:0] pixel_x, pixel_y;
26     wire AISO, video_on, pixel_tick;
27     reg  [11:0] rgb_reg;
28     wire [11:0] rgb_next;
29
30     // BODY
31     // INSTANTIATE AISO UNIT
32     AISO              AISO_unit (.clk(clk), .reset(reset), .rst_s(AISO));
33
34     // INSTANTIATE VGA SYNC UNIT
35     vga_sync         vsync_unit (.clk(clk),           .reset(AISO),
36                                  .hsync(hsync),       .vsync(vsync),
37                                  .video_on(video_on), .p_tick(pixel_tick),
38                                  .pixel_x(pixel_x),   .pixel_y(pixel_y));
39
40     // INSTANTIATE GRAPHIC GENERATOR
41     pong_graph_st     pong0(.clk(clk),         .reset(AISO),    .video_on(video_on),
42                             .btn_up(btn_up),   .btn_down(btn_down),
43                             .pixel_x(pixel_x), .pixel_y(pixel_y),
44                             .graph_rgb(rgb_next));
45     // RGB Buffer
46     always @ (posedge clk) if (pixel_tick) rgb_reg <= rgb_next;
47
48     // OUTPUT
49     assign rgb = rgb_reg;
50  endmodule
```