

```

1 -----
2 -- Engineer: Zachary Montoya
3 -- Submitted Date: 12-15-22
4 -- Module Name: Top - Behavioral
5 -- Project Name: Asteroids
6 -- Target Devices: Zybo Z7-10
7 -- Tool versions: Vivado 2020.2
8 --
9 -- Comment:
10 -- This file was entirely created the engineer listed above.
11 --
12 -----
13
14 --
15 -----
16 -----
17 library ieee;
18 use ieee.std_logic_1164.all;
19 use ieee.numeric_std.all;
20
21 entity walls is
22     port(
23         clk, reset: in std_logic;
24         -- btn: in std_logic_vector(3 downto 0); --R2 increasing the BTN array to 3
25         video_on: in std_logic;
26         pixel_x, pixel_y: in std_logic_vector(9 downto 0);
27         wall1_on_out, wall2_on_out, wall3_on_out, wall4_on_out: out std_logic;
28         wall1_rgb_out, wall2_rgb_out, wall3_rgb_out, wall4_rgb_out: out
29         std_logic_vector(2 downto 0);
30         Left_Wall_Out, Right_Wall_Out, Bottom_Wall_Out, Top_Wall_Out: out
31         std_logic_vector(9 downto 0)
32     );
33 end walls;
34
35 --
36 -----
37
38 architecture rtl of walls is
39
40     -- Signal used to control speed of ball and how often pushbuttons are checked for
41     -- paddle movement.
42     signal refr_tick: std_logic;
43
44     -- x, y coordinates (0,0 to (639, 479)
45     signal pix_x, pix_y: unsigned(9 downto 0);
46     signal wall1_on, wall2_on, wall3_on, wall4_on: std_logic;
47     signal wall1_rgb, wall2_rgb, wall3_rgb, wall4_rgb: std_logic_vector(2 downto 0);
48     signal Left_Wall, Right_Wall, Bottom_Wall, Top_Wall: std_logic_vector(9 downto 0);
49
50     -- Screen dimensions
51     constant MAX_X: integer := 640;
52     constant MAX_Y: integer := 480;
53
54     -- WALL1 - LEFT
55     constant WALL1_X_L: integer := 0;
56     constant WALL1_X_R: integer := 20;
57
58     -- WALL2 - RIGHT
59     constant WALL2_X_L: integer := 619;
60     constant WALL2_X_R: integer := 639;
61
62     -- WALL3 - BOTTOM
63     constant WALL3_X_T: integer := 409;
64     constant WALL3_X_B: integer := 479;
65
66     -- WALL4 - TOP
67     constant WALL4_X_T: integer := 0;
68     constant WALL4_X_B: integer := 20;
69
70     --
71     -----
72     begin
73         wall1_on_out <= wall1_on;

```

```

68 wall2_on_out <= wall2_on;
69 wall3_on_out <= wall3_on;
70 wall4_on_out <= wall4_on;
71 -- bar_on_out <= rom_bar_on;
72 -- asteroid2_on_out <= rd_ball_on;
73
74 wall1_rgb_out <= wall1_rgb;
75 wall2_rgb_out <= wall2_rgb;
76 wall3_rgb_out <= wall3_rgb;
77 wall4_rgb_out <= wall4_rgb;
78 Left_Wall_Out <= Left_Wall;
79 Right_Wall_Out <= Right_Wall;
80 Bottom_Wall_Out <= Bottom_Wall;
81 Top_Wall_Out <= Top_Wall;
82 Left_Wall <= std_logic_vector(to_unsigned(WALL1_X_R,10));
83 Right_Wall <= std_logic_vector(to_unsigned(WALL2_X_L,10));
84 Bottom_Wall <= std_logic_vector(to_unsigned(WALL3_X_T,10));
85 Top_Wall <= std_logic_vector(to_unsigned(WALL4_X_B,10));
86
87 --
=====
88 pix_x <= unsigned(pixel_x);
89 pix_y <= unsigned(pixel_y);
90
91 -- Refr_tick: 1-clock tick asserted at start of v_sync, e.g., when the screen is
refreshed -- speed is 60 Hz
92 refr_tick <= '1' when (pix_y = 1) and (pix_x = 1) else '0';
93
94 --
=====
95 -- wall1 left vertical stripe
96 wall1_on <= '1' when (WALL1_X_L <= pix_x) and (pix_x <= WALL1_X_R) else '0'; --
convert pix_x to pix_y to make horizontal
97 wall1_rgb <= "011"; -- paddle colors blue
98 -- wall2 right vertical stripe
99 wall2_on <= '1' when (WALL2_X_L <= pix_x) and (pix_x <= WALL2_X_R) else '0';
100 wall2_rgb <= "011"; -- paddle colors blue
101 -- wall3 left vertical stripe
102 wall3_on <= '1' when (WALL3_X_T <= pix_y) and (pix_y <= WALL3_X_B) else '0'; --
convert pix_x to pix_y to make horizontal
103 wall3_rgb <= "011"; -- paddle colors blue
104 -- wall4 right vertical stripe
105 wall4_on <= '1' when (WALL4_X_T <= pix_y) and (pix_y <= WALL4_X_B) else '0';
106 wall4_rgb <= "011"; -- paddle colors blue
107
108 end rtl;

```