--------------------------------------------------------------------------------

--

-- FileName: hw\_image\_generator.vhd

-- Dependencies: none

-- Design Software: Quartus II 64-bit Version 12.1 Build 177 SJ Full Version

--

-- HDL CODE IS PROVIDED "AS IS." DIGI-KEY EXPRESSLY DISCLAIMS ANY

-- WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT

-- LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A

-- PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL DIGI-KEY

-- BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL

-- DAMAGES, LOST PROFITS OR LOST DATA, HARM TO YOUR EQUIPMENT, COST OF

-- PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES, ANY CLAIMS

-- BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF),

-- ANY CLAIMS FOR INDEMNITY OR CONTRIBUTION, OR OTHER SIMILAR COSTS.

--

-- Version History

-- Version 1.0 05/10/2013 Scott Larson

-- Initial Public Release

--

--------------------------------------------------------------------------------

LIBRARY ieee;

use IEEE.STD\_LOGIC\_1164.all;

use IEEE.STD\_LOGIC\_ARITH.all;

use IEEE.STD\_LOGIC\_UNSIGNED.all;

ENTITY hw\_image\_generator IS

GENERIC(

pixels\_y : INTEGER := 940; --row that first color will persist until

pixels\_x : INTEGER := 400); --column that first color will persist until

PORT(

disp\_ena : IN STD\_LOGIC; --display enable ('1' = display time, '0' = blanking time)

row : IN INTEGER; --row pixel coordinate

column : IN INTEGER; --column pixel coordinate

red : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0) := (OTHERS => '0'); --red magnitude output to DAC

green : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0) := (OTHERS => '0'); --green magnitude output to DAC

blue : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0) := (OTHERS => '0'); --blue magnitude output to DAC

p1s1 : IN STD\_LOGIC; --player 1 switch 1 (sw17)

p1s2 : IN STD\_LOGIC; --SW16

p1s3 : IN STD\_LOGIC; --SW15

p1s4 : IN STD\_LOGIC; --SW14

p1s5 : IN STD\_LOGIC; --SW13

p1s6 : IN STD\_LOGIC; --SW12

p1s7 : IN STD\_LOGIC; --SW11

p1s8 : IN STD\_LOGIC; --SW10

p1s9 : IN STD\_LOGIC; --SW9

p2s1 : IN STD\_LOGIC; --player 2 switch 1 (sw8)

p2s2 : IN STD\_LOGIC; --SW7

p2s3 : IN STD\_LOGIC; --SW6

p2s4 : IN STD\_LOGIC; --SW5

p2s5 : IN STD\_LOGIC; --SW4

p2s6 : IN STD\_LOGIC; --SW3

p2s7 : IN STD\_LOGIC; --SW2

p2s8 : IN STD\_LOGIC; --SW1

p2s9 : IN STD\_LOGIC; --SW0

startbutton : IN STD\_LOGIC; --start button

CLK1 : IN STD\_LOGIC; --clock for counter (did not use)

p1wins : IN STD\_LOGIC; --count for player one win

p2wins : IN STD\_LOGIC); --count for player two wins

END hw\_image\_generator;

ARCHITECTURE behavior OF hw\_image\_generator IS

BEGIN

PROCESS(disp\_ena, row,column, p1s1, p1s2, p1s3, p1s4, p1s5, p1s6, p1s7, p1s8, p1s9, p2s1, p2s2, p2s3, p2s4, p2s5, p2s6, p2s7, p2s8, p2s9, startbutton, p1wins, p2wins)

variable p1count : integer := 0;

variable winsp1 : integer := 0;

variable winsp2 : integer := 0;

BEGIN

IF(startbutton = '0') THEN

p1count := 1;

END IF;

IF(disp\_ena = '1' AND p1count = 1) THEN

IF((row < (940) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 940 AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSE --background

red <= (OTHERS => '0');

green <= (OTHERS => '0');

blue <= (OTHERS => '0');

END IF;

--BOXES WHEN SWTICHED COORDINATES PLAYER 1

--SW17 = BOX 1 p1s1

--SW16 = BOX 2 p1s2

--SW15 = BOX 3 p1s3

--SW14 = BOX 4 p1s4

--SW13 = BOX 5 p1s5

--SW12 = BOX 6 p1s6

--SW11 = BOX 7 p1s7

--SW10 = BOX 8 p1s8

--SW9 = BOX 9 p1s9

IF(disp\_ena = '1' AND p1s1 = '1') THEN --switch one box 1

IF((row < 360 AND row > 90) AND (column < (290) AND column > 10)) THEN --top left box

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p1s2 = '1') THEN --box 2

IF ((row < 650 AND row > 400) AND (column < (290) AND column > 10)) THEN --top middle box

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p1s3 = '1') THEN --BOX 3

IF ((row < 930 AND row > 690) AND (column < (290) AND column > 10)) THEN --top right box

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p1s4 = '1') THEN --BOX 4

IF ((row < 360 AND row > 90) AND (column < (610) AND column > 330)) THEN --middle left box

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p1s5 = '1') THEN --BOX 5

IF ((row < 650 AND row > 400) AND (column < (610) AND column > 330)) THEN --middle middle box

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p1s6 = '1') THEN --BOX 6

IF ((row < 930 AND row > 690) AND (column < (610) AND column > 330)) THEN --middle right box

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p1s7 = '1') THEN --BOX 7

IF ((row < 360 AND row > 90) AND (column < (930) AND column > 650)) THEN --bottom left box

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p1s8 = '1') THEN --BOX 8

IF ((row < 650 AND row > 400) AND (column < (930) AND column > 650)) THEN --bottom middle box

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p1s9 = '1') THEN --BOX 9

IF ((row < 930 AND row > 690) AND (column < (930) AND column > 650)) THEN --bottom right box

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

--PLAYER TWO BOXES

--SW8 = BOX 1 p2s1

--SW7 = BOX 2 p2s2

--SW6 = BOX 3 p2s3

--SW5 = BOX 4 p2s4

--SW4 = BOX 5 p2s5

--SW3 = BOX 6 p2s6

--SW2 = BOX 7 p2s7

--SW1 = BOX 8 p2s8

--SW0 = BOX 9 p2s9

IF(disp\_ena = '1' AND p2s1 = '1') THEN --switch one box 1

IF((row < 360 AND row > 90) AND (column < (290) AND column > 10)) THEN --top left box

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p2s2 = '1') THEN --box 2

IF ((row < 650 AND row > 400) AND (column < (290) AND column > 10)) THEN --top middle box

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p2s3 = '1') THEN --BOX 3

IF ((row < 930 AND row > 690) AND (column < (290) AND column > 10)) THEN --top right box

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p2s4 = '1') THEN --BOX 4

IF ((row < 360 AND row > 90) AND (column < (610) AND column > 330)) THEN --middle left box

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p2s5 = '1') THEN --BOX 5

IF ((row < 650 AND row > 400) AND (column < (610) AND column > 330)) THEN --middle middle box

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p2s6 = '1') THEN --BOX 6

IF ((row < 930 AND row > 690) AND (column < (610) AND column > 330)) THEN --middle right box

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p2s7 = '1') THEN --BOX 7

IF ((row < 360 AND row > 90) AND (column < (930) AND column > 650)) THEN --bottom left box

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p2s8 = '1') THEN --BOX 8

IF ((row < 650 AND row > 400) AND (column < (930) AND column > 650)) THEN --bottom middle box

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND p2s9 = '1') THEN --BOX 9

IF ((row < 930 AND row > 690) AND (column < (930) AND column > 650)) THEN --bottom right box

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

--POSSIBILITIES THAT PLAYER ONE WINS

IF(disp\_ena = '1' AND (p1S1 = '1' AND p1s2 = '1' AND p1s3 = '1')) THEN --TOP ROW WINNING HORIZONTAL

p1count:= 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p1S4 = '1' AND p1s5 = '1' AND p1s6 = '1')) THEN --middle row winning HORIZONTAL

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row >80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p1S7 = '1' AND p1s8 = '1' AND p1s9 = '1')) THEN --bottom row winning HORIZONTAL

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p1S1 = '1' AND p1s4 = '1' AND p1s7 = '1')) THEN --left row winning vertical

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p1S2 = '1' AND p1s5 = '1' AND p1s8 = '1')) THEN --middle row winning vertical

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p1S3 = '1' AND p1s6 = '1' AND p1s9 = '1')) THEN --right row winning vertical

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p1S1 = '1' AND p1s5 = '1' AND p1s9 = '1')) THEN --left vertical winning

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p1S3 = '1' AND p1s5 = '1' AND p1s7 = '1')) THEN --right vertical winning

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

END IF;

END IF;

--POSSIBILITIES THAT PLAYER TWO WINS

IF(disp\_ena = '1' AND (p2S1 = '1' AND p2s2 = '1' AND p2s3 = '1')) THEN --TOP ROW WINNING HORIZONTAL

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p2S4 = '1' AND p2s5 = '1' AND p2s6 = '1')) THEN --middle row winning HORIZONTAL

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p2S7 = '1' AND p2s8 = '1' AND p2s9 = '1')) THEN --bottom row winning HORIZONTAL

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p2S1 = '1' AND p2s4 = '1' AND p2s7 = '1')) THEN --left row winning vertical

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p2S2 = '1' AND p2s5 = '1' AND p2s8 = '1')) THEN --middle row winning vertical

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p2S3 = '1' AND p2s6 = '1' AND p2s9 = '1')) THEN --right row winning vertical

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p2S1 = '1' AND p2s5 = '1' AND p2s9 = '1')) THEN --left vertical winning

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

IF(disp\_ena = '1' AND (p2S3 = '1' AND p2s5 = '1' AND p2s7 = '1')) THEN --right vertical winning

p1count := 2;

IF((row < (pixels\_y) AND row > 80) AND (column < (320) AND column > 300)) THEN --first horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < pixels\_y AND row > 80) AND (column < 640 AND column > 620)) THEN --second horizontal line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 390 AND row >370) AND column < 940) THEN --first vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

ELSIF ((row < 680 AND row >660) AND column < 940) THEN --second vertical line

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

END IF;

END IF;

END IF;

--Press to show that P1 has won both games

IF(disp\_ena = '1' AND p1wins = '0') THEN

winsp1 := 1;

END IF;

IF(winsp1 = 1) THEN

IF(row < 1280 AND column < 1024) THEN

red <= (OTHERS => '0');

green <= (OTHERS => '1');

blue <= (OTHERS => '1');

winsp1 := 2;

END IF;

END IF;

--Press to show that P2 has won both games

IF(disp\_ena = '1' AND p2wins = '0') THEN

winsp2 := 1;

END IF;

IF(winsp2 = 1) THEN

IF(row < 1280 AND column < 1024) THEN

red <= (OTHERS => '1');

green <= (OTHERS => '0');

blue <= (OTHERS => '1');

winsp2 := 2;

END IF;

END IF;

END PROCESS;

END behavior;