LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

ENTITY hw\_image\_generator IS

PORT(

ps2\_code : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

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

END hw\_image\_generator;

ARCHITECTURE behavior OF hw\_image\_generator IS

signal x\_position : integer range 0 to 1280;

signal y\_position : integer range 0 to 1024;

-- signal x\_box : integer range 0 to 1280 := 20;

-- signal y\_box : integer range 0 to 1024 := 20;

BEGIN

PROCESS(disp\_ena, row, column, x\_position, y\_position) --x\_box, y\_box)

BEGIN

IF(disp\_ena = '1') THEN --display time

IF(row < 512 AND column > 231 AND column < 281) THEN

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

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

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

ELSIF(row > 743 AND row < 793 AND column < 512) THEN

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

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

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

ELSIF(row > 1024 AND column > 231 AND column < 281) THEN

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

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

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

ELSIF(row > 256 AND row < 1024 AND column > 487 AND column < 537) THEN

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

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

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

ELSIF(row > 487 AND row < 537 AND column > 536 AND column < 768) THEN

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

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

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

ELSIF(row > 231 AND row < 281 AND column > 768) THEN

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

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

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

ELSIF(row > 743 AND row < 793 AND column > 768) THEN

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

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

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

ELSIF(row > 974 AND row < 1024 AND column > 536 AND column < 768) THEN

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

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

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

ELSE

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

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

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

END IF;

IF(ps2\_code = "00101001") THEN -- space bar

IF(row > 20 AND row < 70 AND column > 20 AND column < 70) THEN

x\_position <= row;

y\_position <= column;

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

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

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

END IF;

END IF;

IF(ps2\_code = "00011101") THEN --W key

y\_position <= y\_position + 10;

x\_position <= x\_position;

IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

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

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

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

END IF;

END IF;

IF(ps2\_code = "00011011") THEN --S key

y\_position <= y\_position - 10;

x\_position <= x\_position;

IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

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

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

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

END IF;

END IF;

IF(ps2\_code = "00100011") THEN --D key

x\_position <= x\_position + 10;

y\_position <= y\_position;

IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

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

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

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

END IF;

END IF;

IF(ps2\_code = "00011100") THEN --A key

x\_position <= x\_position - 10;

y\_position <= y\_position;

IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

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

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

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

END IF;

END IF;

-- IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

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

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

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

-- END IF;

-- IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

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

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

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

-- END IF;

-- IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

-- IF(ps2\_code = "00011101") THEN --W key

-- y\_position <= y\_position + 10;

-- x\_position <= x\_position;

--

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

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

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

-- END IF;

-- END IF;

--

-- IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

-- IF(ps2\_code = "00011011") THEN --S key

-- y\_position <= y\_position - 10;

-- x\_position <= x\_position;

--

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

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

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

-- END IF;

-- END IF;

--

-- IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

-- IF(ps2\_code = "00100011") THEN --D key

-- x\_position <= x\_position + 10;

-- y\_position <= y\_position;

--

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

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

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

-- END IF;

-- END IF;

--

-- IF(row > x\_position AND row < x\_position+50 AND column > y\_position AND column < y\_position+50) THEN

-- IF(ps2\_code = "00011100") THEN --A key

-- x\_position <= x\_position - 10;

-- y\_position <= y\_position;

--

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

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

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

-- END IF;

-- END IF;

-- END IF;

ELSE --blanking time

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

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

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

END IF;

END PROCESS;

END behavior;