```
----------------------------------------------------------------------------------
-- Engineer: Taggart Bonham and Will Chisholm
--
-- Create Date: 05/23/2017 09:05:13 PM
-- Design Name:
-- Module Name: VGA_SYNC - Behavioral
-- Project Name: Etch-a-Sketch final project
-- Target Devices: Digilent Basys3 Board
-- Tool Versions: Vivado 2016.1
-- Description:

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

-- Code your design here
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

ENTITY VGA_SYNC IS
PORT (  clk     :   in  STD_LOGIC; --100 MHz clock
        V_sync  :   out STD_LOGIC;
        H_sync  :   out STD_LOGIC;
        video_on:   out STD_LOGIC;
        pixel_x :   out STD_LOGIC_vector(9 downto 0);
        pixel_y :   out STD_LOGIC_VECTOR(8 downto 0) );
end VGA_SYNC;


architecture behavior of VGA_SYNC is


--video on signals
signal H_video_on : STD_LOGIC := '1';
signal V_video_on : STD_LOGIC := '1';

--internal signals to determine when last Hsync went
signal h_complete : STD_LOGIC := '1'; --used to take place of rising_edge(H_sync)

--VGA Constants

constant left_border : integer := 48;
constant h_display : integer := 640;
constant right_border : integer := 16;
constant h_retrace : integer := 96;
constant HSCAN : integer := left_border + h_display + right_border + h_retrace - 1; --number of
PCLKs in an H_sync period

--counts
signal H_CNT : integer:= 0;
signal V_CNT : integer:= 0;

constant top_border : integer := 29;
constant v_display : integer := 480;
constant bottom_border : integer := 10;
constant v_retrace : integer := 2;
constant VSCAN : integer := top_border + v_display + bottom_border + v_retrace - 1; --number of
H_syncs in an V_sync period

BEGIN

hCounter: process(clk)
begin
if rising_edge(clk) then
    if (H_CNT < HSCAN) then
```

```
        H_CNT <= H_CNT + 1;
    elsif(H_CNT = HSCAN) then
        H_CNT <= 0;
    end if;
end if;
end process;

vCounter: process(clk)
begin
if rising_edge(clk) then
    if (H_complete = '1') then -- works as rising clk on h
         if (V_CNT < VSCAN) then
            V_CNT <= V_CNT + 1;
        elsif(V_CNT = VSCAN) then
             V_CNT <= 0;
    end if;
end if;
end if;
end process;

H_Logic: process (H_CNT)
begin
    H_sync <= '1';
    H_video_on <= '1';
    H_complete <= '0';

    pixel_x <=STD_LOGIC_VECTOR(to_unsigned(H_CNT, 10));

       if (H_CNT <  h_display) then
         H_video_on <= '1';
         H_sync <= '1';
           H_complete <= '0';

         elsif (H_CNT < h_display + right_border) then
            H_video_on <= '0';
             H_sync <= '1';
           H_complete <= '0';

         elsif ( H_CNT <  h_display + right_border + h_retrace) then
            H_video_on <= '0';
             H_sync <= '0';
             H_complete <= '0';

          elsif (H_CNT < HSCAN) then
                    H_video_on <= '0';
                     H_sync <= '1';
                H_complete <= '0';

          elsif (H_CNT = HSCAN) then
            H_video_on <= '0';
             H_complete <= '1';
              H_sync <= '1';

          end if;
          end process;



--V_sync generating process
Vert_logic : process(V_CNT)
begin


   V_sync <= '1';
```

```
   V_video_on <= '1';

   pixel_y <=STD_LOGIC_VECTOR(to_unsigned(V_CNT, 9));

      if (V_CNT <  v_display) then
        V_video_on <= '1';
        V_sync <= '1';

        elsif (V_CNT < v_display + bottom_border) then
           V_video_on <= '0';
            V_sync <= '1';

        elsif ( V_CNT <  v_display + bottom_border + v_retrace) then
           V_video_on <= '0';
            V_sync <= '0';

         elsif (V_CNT <= VSCAN) then
                   V_video_on <= '0';
                    V_sync <= '1';

         end if;

end process;

video_on <= H_video_on AND V_video_on;

end behavior;
```