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

--

-- FileName: vga\_controller.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;

ENTITY vga\_controller IS

GENERIC(

h\_pulse : INTEGER := 112; --horiztonal sync pulse width in pixels

h\_bp : INTEGER := 248; --horiztonal back porch width in pixels

h\_pixels : INTEGER := 1280; --horiztonal display width in pixels

h\_fp : INTEGER := 48; --horiztonal front porch width in pixels

h\_pol : STD\_LOGIC := '0'; --horizontal sync pulse polarity (1 = positive, 0 = negative)

v\_pulse : INTEGER := 3; --vertical sync pulse width in rows

v\_bp : INTEGER := 38; --vertical back porch width in rows

v\_pixels : INTEGER := 1024; --vertical display width in rows

v\_fp : INTEGER := 1; --vertical front porch width in rows

v\_pol : STD\_LOGIC := '1'); --vertical sync pulse polarity (1 = positive, 0 = negative)

PORT(

pixel\_clk : IN STD\_LOGIC; --pixel clock at frequency of VGA mode being used

reset\_n : IN STD\_LOGIC; --active low asycnchronous reset

h\_sync : OUT STD\_LOGIC; --horiztonal sync pulse

v\_sync : OUT STD\_LOGIC; --vertical sync pulse

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

column : OUT INTEGER; --horizontal pixel coordinate

row : OUT INTEGER; --vertical pixel coordinate

n\_blank : OUT STD\_LOGIC; --direct blacking output to DAC

n\_sync : OUT STD\_LOGIC); --sync-on-green output to DAC

END vga\_controller;

ARCHITECTURE behavior OF vga\_controller IS

CONSTANT h\_period : INTEGER := h\_pulse + h\_bp + h\_pixels + h\_fp; --total number of pixel clocks in a row

CONSTANT v\_period : INTEGER := v\_pulse + v\_bp + v\_pixels + v\_fp; --total number of rows in column

BEGIN

n\_blank <= '1'; --no direct blanking

n\_sync <= '0'; --no sync on green

PROCESS(pixel\_clk, reset\_n)

VARIABLE h\_count : INTEGER RANGE 0 TO h\_period - 1 := 0; --horizontal counter (counts the columns)

VARIABLE v\_count : INTEGER RANGE 0 TO v\_period - 1 := 0; --vertical counter (counts the rows)

BEGIN

IF(reset\_n = '0') THEN --reset asserted

h\_count := 0; --reset horizontal counter

v\_count := 0; --reset vertical counter

h\_sync <= NOT h\_pol; --deassert horizontal sync

v\_sync <= NOT v\_pol; --deassert vertical sync

disp\_ena <= '0'; --disable display

column <= 0; --reset column pixel coordinate

row <= 0; --reset row pixel coordinate

ELSIF(pixel\_clk'EVENT AND pixel\_clk = '1') THEN

--counters

IF(h\_count < h\_period - 1) THEN --horizontal counter (pixels)

h\_count := h\_count + 1;

ELSE

h\_count := 0;

IF(v\_count < v\_period - 1) THEN --veritcal counter (rows)

v\_count := v\_count + 1;

ELSE

v\_count := 0;

END IF;

END IF;

--horizontal sync signal

IF(h\_count < h\_pixels + h\_fp OR h\_count > h\_pixels + h\_fp + h\_pulse) THEN

h\_sync <= NOT h\_pol; --deassert horiztonal sync pulse

ELSE

h\_sync <= h\_pol; --assert horiztonal sync pulse

END IF;

--vertical sync signal

IF(v\_count < v\_pixels + v\_fp OR v\_count > v\_pixels + v\_fp + v\_pulse) THEN

v\_sync <= NOT v\_pol; --deassert vertical sync pulse

ELSE

v\_sync <= v\_pol; --assert vertical sync pulse

END IF;

--set pixel coordinates

IF(h\_count < h\_pixels) THEN --horiztonal display time

column <= h\_count; --set horiztonal pixel coordinate

END IF;

IF(v\_count < v\_pixels) THEN --vertical display time

row <= v\_count; --set vertical pixel coordinate

END IF;

--set display enable output

IF(h\_count < h\_pixels AND v\_count < v\_pixels) THEN --display time

disp\_ena <= '1'; --enable display

ELSE --blanking time

disp\_ena <= '0'; --disable display

END IF;

END IF;

END PROCESS;

END behavior;