EE 3610 Digital Systems

Lab 4

Title: Video Timing.

Objective: The student will be introduced to design element libraries and will use the Xilinx unified library to generate a 65 MHz clock. The student will also understand the basics of video timing.

Equipment: Spartan 3E Starter Board
VGA monitor that accepts the 1024x768 XGA format.

Background: In this lab, you are to design a system to generate a simple color pattern on a VGA monitor. You will have to generate signals for vertical sync, horizontal sync, red, green and blue.

A frame of VGA data is transmitted as a sequence of horizontal lines. Each line is, in turn, transmitted as a sequence of pixels. The color of the pixels is set by the red, green and blue signals. For the 1024x768 XGA format, a line consists of 1344 pixels: 1024 data pixels, a “front porch” of 24 pixels, a horizontal sync of 136 pixels and a “back porch” of 160 pixels. A frame consists of 806 lines: 768 data lines, a 3-line front porch, a 6-line vertical sync and a 29-line back porch. The front porch, sync pulse and back porch (both vertical and horizontal) are collectively known as the “blanking interval,” and during that time, the red, green and blue signals must be 0.

For the XGA format, pixels are transmitted at a frequency of 65MHz. This constitutes a challenge on the Spartan 3E starter board since the only external clock is 50MHz. Fortunately, the Spartan 3E FPGA contains multiple DCM (Digital clock manager) modules, one of which can be used to generate the 65 MHz clock. To access the DCM, it is necessary to use primitives from the Xilinx unified library. To use the library, you must include the following lines at the beginning of your VHDL file:

```
library UNISIM;
use UNISIM.VComponents.all;
```

You will need to use three primitive components: IBUFG, DCM_SP and BUFG. IBUFG is a buffer that routes the external clock to the DCM. DCM_SP is the DCM itself, and BUFG connects to the DCM output and distributes the clock to the flip-flops and registers. More information about these primitives can be found at:

http://www.xilinx.com/support/documentation/user_guides/ug331.pdf

Preparation: Write the title and a short description of this lab in your lab book. Make sure the page is numbered and make an entry in the table of contents for this lab.

Draw a schematic that includes only the components you will be using for this lab. Specifically, include the FPGA (but show only the pins you are using) and the VGA port (See J15 on page 2 the Spartan 3E Starter Board Schematic.) Include pin numbers or coordinates. Omit the power supply.

Design a VHDL module to generate the horizontal and vertical sync pulses for the VGA port. Your module should take two inputs: reset and 65 MHz clock. It should output horizontal sync, vertical sync and a blanking signal (indicating that the system is in a blanking interval). It should also generate a horizontal start pulse (asserted one cycle before the first data pixel in each horizontal line) and a 9-bit line number (which changes only during the horizontal sync pulse).

Design a second VHDL module to generate the 65MHz clock given a 50MHz external clock. The module should consist of one instance each of an IBUFG, a DCM_SP and a BUFG component. The DCM_SP module needs to be configured using a “generic map”. For example:

```
DCM_SP_inst : DCM_SP
  generic map (
    CLKDV_DIVIDE => 2.0,
    CLKFX_DIVIDE => 10, -- Can be any interger from 1 to 32
    CLKFX_MULTIPLY => 13, -- Can be any integer from 1 to 32
    CLKIN_DIVIDE_BY_2 => FALSE, -- don’t divide CLKIN by two
    CLKIN_PERIOD => 20.0, -- Specify period of input clock
    CLKOUT_PHASE_SHIFT => "NONE", -- No phase shift
    CLK_FEEDBACK => "1X", -- Specify "NONE", "1X" or "2X"
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS",
    DLL_FREQUENCY_MODE => "LOW", -- "HIGH" or "LOW"
    DUTY_CYCLE_CORRECTION => TRUE, -- Duty cycle correction
    PHASE_SHIFT => 0, -- Amount of fixed phase shift from -255 to 255
    STARTUP_WAIT => FALSE) -- Don’t wait until DCM_SP LOCK
  port map (
    CLK0 => Open, -- 0 degree DCM CLK output
    CLK180 => Open, -- 180 degree DCM CLK output
    CLK270 => Open, -- 270 degree DCM CLK output
    CLK2X => Open, -- 2X DCM CLK output
    CLK2X180 => Open, -- 2X, 180 degree DCM CLK out
    CLK90 => Open, -- 90 degree DCM CLK output
    CLKDV => Open, -- Divided DCM CLK out (CLKDV_DIVIDE)
    CLKFX => Clk2, -- DCM CLK synthesis out (M/D)
    CLKFX180 => Open, -- 180 degree CLK synthesis out
    LOCKED => Open, -- DCM LOCK status output
    PSDONE => Open, -- Dynamic phase adjust done output
    STATUS => Open, -- 8-bit DCM status bits output
    CLKFB => Open, -- DCM clock feedback
    CLKIN => Clk1, -- Clock input (from BUFG)
    PSCLK => Open, -- Dynamic phase adjust clock input
    PSEN => Open, -- Dynamic phase adjust enable input
    PSINCDEC => Open, -- Dynamic phase adjust increment/decrement
    RST => '0' -- DCM asynchronous reset input
  );
```

Note that in the code above, Clk1 and Clk2 are internal signals. The 50MHz clock is routed through an IBUFG buffer to Clk1, and Clk2 is routed through a BUFG buffer to the clock network on the FPGA.

Write a test bench for your VGA module. Simulate the design for 1 frame to verify proper horizontal and vertical sync pulse timings. Zoom in to show one line, print the waveform and affix it to your lab book.

Design a top level module that connects the 50MHz clock to an internal 65MHz clock via your clock module. It should also connect your VGA module to the horizontal and vertical sync signals on the Spartan board. Use the horizontal start pulse from your VGA module to reset a 3-bit free-running counter. Add the output of the counter to the least significant 3 bits of the line number and use the 3-bit result (gated by the blanking signal) to drive the red, green and blue outputs. (This will make a diagonal stripe pattern on the monitor.) Synthesize your design to verify that there are no syntax errors.

Bring your lab notebook and the Spartan board, above, to your lab period.

Set up: Connect the USB cable, VGA monitor and power supply to the Spartan board. Turn on power to the monitor and the Spartan board.

Procedure: Download your code. Verify that a stable diagonal stripe pattern appears on the monitor. You may need to auto-adjust the monitor. Demonstrate your system to the lab instructor.

Affix the final copies of your VGA and clock modules (in VHDL) to your lab book.

Conclusions: In the conclusion section, write a short summary of what you did, what you learned, and what could be done better.