www.RetroMicro.com

Created by: Douglas Hodson

If you find this project useful, please email me at

doug@RetroMicro.com

Project Name: xsocVGABitmap

Creation Date: July 2003
Development Board: XESS XSA-100 Plus XStend Version 2
Development Software: Xilinx ISE Version 5.1.03i

Description

This project generates a vga signal and displays a monochrome bitmap by reading data from SDRAM memory. This project also demonstrates how to mix both VHDL and Verilog code together in the same project. The project uses the vga circuit design provided in Jan Gray's XSOC project (see references). Basically is generates the correct timing signals needed and shifts through a 16-bit word one bit at a time (monochrome) for pixel information. As data is requested, SDRAM memory is accessed and data is retrieved.

Accessing SDRAM memory may sound easy, but it's more complicated than SRAM. XESS has produced a special SDRAM "controller" just for this purpose. For full details about the controller, refer to the XESS site.

To generating an image for display, use the "bit2xes" program. It loads a bitmap image into memory and writes it out in xes format. The XESS gxsload tool can be used to download this memory file to SDRAM.

This project must be run at 25MHz due to the fact that this is the minimum freq that SDRAM will remain fresh. The clock in the design is divided in half to drive the vga circuit at 12.5 MHz. Since the vga circuit is running at half the rate of the SDRAM controller, when data is received from the controller, it must be made available to the vga circuit for one extra clock cycle (or twice as long) to ensure vga will "see" it in it's timeframe. This is the reason for the MEMORY_WAIT_FOR_VGA state.

The XStend board is not required in this project.

The XSA-100 board frequency must be set to 25 MHz!

Note: The vga circuit as provided in the XSOC project has a bug. It reads 576x455 bits of data for image display, but only displays 560x455. The last 16-bit word of pixel data read for display on each horizontal line is not displayed. In other words, the right side of the image is cut off. This problem has been corrected in the version supplied.

A Little Bit About Clocks and the SDRAM Controller

When interfacing to SDRAM, you drive the controller with a clock signal wired to the clkin port. After that the controller provides three clock signals back for the design to use; bufclk, clk0 and clk2x. They are defined something like this:

- bufclk : simply a buffered version of the "clkin" signal you are feeding to controller. In other words, the clock signal direct from a pin on the fpga chip with a buffer in the stream.

- clk0 : a clock signal generated within the sdram controller itself to account for the printed circuit board (PCB) delays in interfacing to the externally driven sdram memory.

- clk2x : a doubled version of clk0

When using the SDRAM controller, you supply clkin direct from fpga clock pin, then you should use the clk0 signal as the master clock for your design. You should also use bufclk in conjunction with another sdram provided signal, "lock" to determine when clk0 has "locked" on to the master clock you have provided. This is accomplished with a clkdll primitive.

This project uses the latest version of the SDRAM controller provided by XESS. It should be noted that I have found this version substantially better and more reliable than an earlier version.

Project Directory Structure

This project is organized into the following directories:

./ - contains all the files need to synthesize the design.

./config – effectively a backup of all main files in the root directory

./docs – this PDF file. Source files used to build this PDF are located in ./docs/src.

./bit2xes – program to convert bmp files to xes format.

./images – example image to download to XSA board for display

./src – source directory for all HDL files

./temp – temporary directory used during the build process.

Synthesis

The project is built and maintained using two windows based batch files. The first is “make.bat”. It simply issues are the commands required to “compile” all the source files in the ./src directory and eventually generate the chipIO.bit file to downloaded to the fpga.

The second batch file is called “clean.bat”. Its sole purpose is to delete most of the unwanted files generated during the build process.

References

The primary reference for building this project is Jan Gray’s XSOC project. This project is an excellent place to start on how to build a fully pipelined risc based cpu interfaced with a video display and other devices. See the www.fpgacpu.org website for the complete XSOC project and information.