For the following project design problem, please work together in the following groups:
A. Amol, Marianne and Kaustubh
B. Ashish and Antoine

Hand in your design electronically by tar'ing the design files and mailing the tar file to Charles. We will also want to see a demo of your circuit.

1. You are to design a circuit that reads an image stored in memory and displays it on a VGA monitor. This circuit will become part of your image stabilization project. The image is a 512x512 color image in 16-bit color format. Each 16-bit pixel is in 5:6:5 bit format with 5 bits of red, 6 of green and 5 of blue. The pixels are stored this way in memory, so they can be sent straight to the display without modification. The image is stored in row-major order starting at address 0 and uses the first 256K of the 512K RAM bank. (We will be using the "left" 512Kx16 SRAM bank). The pixels are sent to a RAMDAC chip that converts the 16-bit color format to the analog RGB signals that are sent to the monitor.

We will start you with a framework that does the straight image display, except that it displays the entire 800x600 VGA screen by re-displaying some of the image to fill it up. You should start by understanding how this design works. It uses three library modules:
- The **vga** module provides the timing for displaying images to the VGA monitor. It generates the current horizontal and vertical screen coordinates on every cycle. The circuit must then determine what data should be displayed at that pixel location. In this design, these coordinates are just used to read the pixel value in memory. The vga module also provides the VGA HSYNC, VSYNC and BLANK signals.
- The **ramdac** module provides the pixel data interface to the RAMDAC chip, which drives the VGA display in 16-bit color mode. Each cycle, the pixel to be displayed is written to the ramdac module.
- The **memif2port** is a dual-ported memory interface that allows reads and writes to the external SRAM. This design only reads the memory. Later designs will write new images into memory as produced by the camera.

Documentation for these modules is provided on the project Web page under Hardware. All these modules run at 50MHz, and the last two in particular have been carefully designed to meet the external interface timing requirements. You do not need to understand how these modules work internally, but you do need to understand their interfaces and how to use them.

You must modify this circuit to display only a 384x384 subimage of the 512x512 image. This subimage should start out centered in the full image; that is, the origin of the subimage starts at (64,64) and a strip 64 pixels wide around the image is not displayed. This origin is then changed by the user using the four push-button switches. The first two switches move the origin left and right, while the last two switches move the origin up and down. This allows the user to scroll the subimage within the larger image. (Think carefully about how to do this.) You can let the origin wrap around, or you can constrain it so that scrolling stops at some min and max value (64). Later on, you will use a circuit like this to stabilize the image display by simply changing the origin to follow the motion vector that you will be computing.
The origin should be allowed to change only once every frame time to keep the scrolling rate reasonable. Moreover, the origin should change only during vertical retrace time, i.e. when VSYNC is asserted.

Use simulation to determine that your design works correctly. As before, you will have to change the parameters of the modules so that you don’t have to simulate millions of clock cycles. The vga and syncgen modules in particular contain a set of commented values that should allow reasonable simulation. The class Web page also has a description of how to use vcs, “Verilog Compiled Simulation” along with its GUI. Vcs is a much faster simulator than Verilog-XL, but it takes longer to start up. Thus it is good for long simulations, but the timing diagrams in the GUI are also useful for seeing a complete simulation.