Gerardo Ravago

Rushi Ganmukhi

November 15, 2012

EC551: Fall ‘12

Project Description

**YouTube clip at**: <http://youtu.be/fviGLG6tz5E>

**Goal:** To create a system that reads a stored image from block memory and displays it onto the monitor via VGA. Four button inputs will change the location of the image to one of four locations (top, bottom, right, left).

**Design:**

To implement the system we first used Matlab to create a coefficients file using our test image. Next, using CoreGen we created a block ram module and instantiated the memory with our .coe file. In our top module we called an FSM to determine which state we were in according to which button was pressed and the outputted this image in the correct location to the display over VGA.

**Contributions:**

*Rushi: VGA controller, image into block memory*

*Gerardo: FSM for image state, memory controller*

*Both: top\_module*

**Modules:**

*VGA controller:*

A VGA controller module was provided for us and it worked correctly. We simply called the VGA controller in our top module and inputted a divided clock and received horizontal and vertical sink and counter which we then used to determine where to display a simple image by setting the RGB registers.

*Block Memory Containing Image:*

To create we first used a matlab script to convert our image into a COE file. Next we used CoreGen to create the memory and initialize it with the COE file.

*FSM for Image State(vgaDraw):*

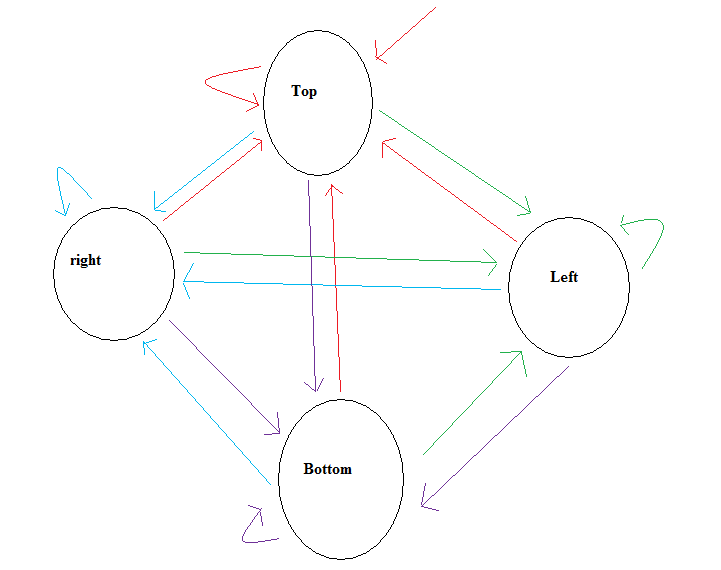
A FSM that outputs the starting coordinates of the image depending on state. Inputs are button presses. Four states corresponding to top,bottom, right,left.

In addition to the module above we implemented a denouncer for the buttons and a clock divider.

**Problems:** While conducting this lab we primarily ran into problems using the memory. To begin we struggled using the VGA controller, but once we realized we needed to divide the clock we fixed this issue. Next we struggled a little accessing the image stored in memory but after some quick reading we solved this problem. The main issue we had during this lab was moving the pictures around on button clicks. After some debugging we realized that this was an issue in our FSM not transitioning states correctly and we fixed it. The only conceptual problem we had during this lab was how to draw images in different locations. In the end we decided that the FSM would return a start value for the horizontal and verticle count. Using this we then wrote from the block memory to these locations until the length of the images and printed black everywhere else.

**Keyboard Interface:** Originally we wanted to use the keyboard as the input, but could not find a PS2 keyboard. Attached is the module for our keyboard controller that simulates. Instead of using the keyboard we used four push buttons on the FPGA. If we were to have a PS2 keyboard then we could easily implement our design by switching out the buttons for keyboard signals in our controller. The code for this is contained in getScanCode.v

**FSM Diagram**

****

*For clarity the transitions are below:*

-Red transitions occur on button zero presses, start/reset, or if currently in top state and no buttons are pressed

-Purple transitions occur on button one presses, or if currently in bottom state and no buttons are pressed

-Green transitions occur on button two presses, or if currently in left state and no buttons are pressed

-Blue transitions occur on button three presses, or if currently in right state and no buttons are pressed