

Computer Architecture

Keyboard Input Lab

By: Hannah Gamiel, Jonathan Burge, Michael Lindahl, Edited slightly by Kent Jones

Introduction

In the VGA lab, you became familiar with displaying colors on a monitor. The finished project from this lab will display colors based on user input from a keyboard. This lab illustrates how to handle input and output using an FPGA. To obtain a basic understanding of keyboard input, we will first view an example project provided by XESS. We will then use this understanding of keyboard input to change the VGA output from the FPGA.

Learning Goals

- Develop a simple VGA display.
- Develop VHDL keyboard scanner that retrieves key scan codes.

PART 1: VGA Review

This lab will begin by reviewing and running the VGA lab we did last semester. First, modify that project and make each color that displays fill the entire screen. (This should only require the modification of one line of code.) Now change the color to be a different color of your choice.

PART 2: Keyboard Input Lab (XESS)

For this lab you will need to use a PS2 keyboard, or an adapter that can convert a regular usb keyboard to a PS2 port.

Read XESS' [Application Note "PS/2 Keyboard Interface for the XSA Boards"](#) document to gain a better understanding of how keyboard input works.

1. What are the two signals that a PS/2 keyboard uses to connect to the XSA Board?

2. What scancodes are used to indicate when a key released on the keyboard?

Download the "PS/2 Keyboard Test Application" found in the document. Unzip and save the folder to your student drive. Read and follow the instructions in the "readme" text file inside of the unzipped folder.

Keyboard Input Lab

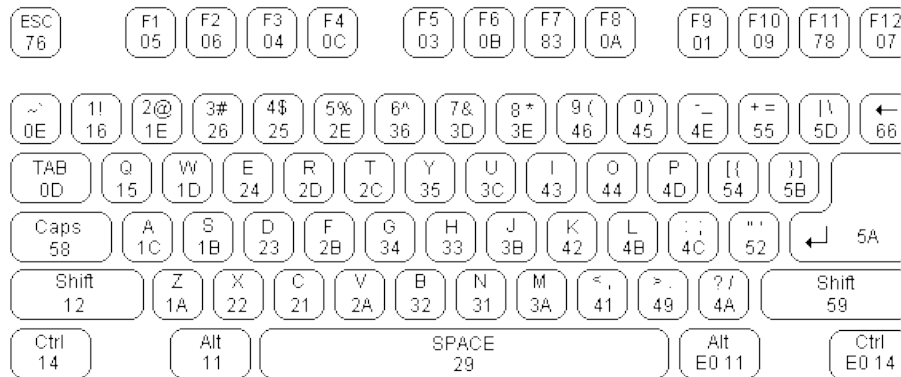
3. What needs to be done for our boards concerning clock frequency?

4. What happens when you press any other key besides 0-9?

Follow the instructions in the readme to run the keyboard application. As a reminder, will be using an XSA-3S1000 board. Test out the application and verify that it works.

Take a look at the .UCF file to gain a better understanding of how communication between the board and the architecture files (test_kbd.vhd) works. How does this communication work?

Each key on the keyboard is assigned a hexadecimal number. Figure 2 shows what these values are on the keyboard. Convert the numbers from keys 0-9 into binary.



0 _____
1 _____
2 _____
3 _____
4 _____
5 _____
6 _____
7 _____
8 _____
9 _____

Figure 2. PS2 Keyboard Scancodes

Note how these numbers match the scancodes in the architecture file. These scancodes are used to detect when certain keys are pressed. It is important to understand how the XESS project works for when we combine the two projects together.

Keyboard Input Lab

PART 3: VGA and Keyboard Combined

This example combines the information from the keyboard application lab and the VGA test lab to create an application that:

1. The test_kbd hardware implements a ps2_kbd object that takes input from the keyboard in the form of a scancode. This scancode is transferred from the test_kbd object to a vgatest object via the scancode_bus signal.
2. The vgatest object simply looks at the bottom three bits of the scancode and assigns these bits to the red_out signal.

When choosing colors, it is important to note that the color system we are using is base 8 (binary values 000 through 111). This was illustrated earlier in the VGA lab when we learned that you received the maximum signal at 0.7 volts and no signal at 0 volts. Thus, when each signal is at 0.7 volts, a white color is produced. In the VGA lab, we used a 3-bit value to represent each color component of the RGB value.

Experiment

Modify this lab. See if you can change the color based on a particular key press. For example, press r to get red, g to get green and b to set blue, then, use digits 0, 1, 2, ... 7 to set the intensity of the currently selected component.

Challenges

- Research and attempt to develop a more advanced keyboard scanner that sends keyboard information to the XESS boards SDRAM using the dual port SDRAM driver provided by XESS that we looked at in class last time.
- Create a text generator that scans the keyboard buffer in memory and writes pixels to the VGA display generator so that keyboard information is visible.
- Integrate the mips processor, SDRAM, and keyboard and VGA generator all into one application.