

PS2

Introduction

Students will be introduced to the PS2 interfaces and controllers in this recitation.

Understanding the interaction between the interfaces and controllers is key to understanding how to utilize different functional modules. Students will study the PS2 controller provided and then alter the code so that pressing a key on a PS2 keyboard will cause the FPGA to display the corresponding character on the built-in LCD screen.

Equipment

- Altera DE2-115 FPGA
- PS2 keyboard

Lab Tasks

To receive credit for this lab, you must complete:

- ☐ Task 1: Integrate the PS2 input and display results on the FPGA LCD
- ☐ Task 2: Clean up workstation

You must complete all parts of this lab to receive credit. A TA must sign-off on the completion of each task. Ensure that the TA marks the completion of the tasks in Sakai.

Pre-Lab Instructions

There is no pre-lab.

Lab Instructions

NOTE: Items shown in red should be shown to the TA before continuing with the lab.

PS2 Input and FPGA LCD

In this section of the lab, you will be using the file provided on Sakai to learn about how to integrate a PS2 input, and how to use the LCD built into the FPGA. Additionally, you will write a skeleton to take in an input (use a character from the range a-z) and display it on the LCD on the FPGA.

To begin this lab, download the QAR file from Sakai. Open the “PS2_Interface.v” and “PS2_Controller.v” modules. In this first step, you will look at how these modules

interact with one another and write-up a quick description. Focus on the PS2_Interface.v file first; in this file, there are multiple sections, and it's important to understand what each section does:

Port Declarations	This section contains the port declarations and definitions used for the module.
Constant Declarations	This section contains a list of local parameters whose scope is that of the module they are contained in. These 'localparam' values are protected from being altered by any module that calls an instance of the PS2_Controller.v. Leave this section as is
Internal Wires and Registers Declarations	This section has the internal wire and register declarations used throughout the module. Leave this section alone.
Finite State Machine(s)	<p>This section contains a simple FSM which controls the current state of the PS2_Controller. In the first always block, the reset condition is defined so that if a reset signal is given, the controller moves to the default 'PS2_STATE_0_IDLE' state. Else, it allows the current state of the ps2_transceiver to be assigned the correct next ps2_transceiver state.</p> <p>In the next always block, the conditions for which state the controller will move into are defined. Using a case statement, the logic is defined for what the next state should be based on the inputs being seen. The states are: PS2_STATE_0_IDLE, PS2_STATE_1_DATA_IN, PS2_STATE_2_COMMAND_OUT, PS2_STATE_3_END_TRANSFER, PS2_STATE_4_END_DELAYED.</p> <p>S_ps2_transceiver is in parenthesis at the beginning of the case statement because the case statement is sensitive to that signal; this means whenever the value of this changes, the case statement is entered.</p> <p>The correct current state that the PS2_transceiver should be in is updated on every clock cycle.</p>
Sequential Logic	This section contains the sequential logic of the controller. The always statement uses the clock signal to control when the values of certain registers are updated. DON't worry about this section too much.
Combinational Logic	This section contains the logic used to determine what the positive and negative edges of the clock signal are. In addition, some basic assignments are made for asserting the correct

	values on the module ports.
Internal Modules	These are internal instances of Altera's IP blocks that handle communication with the internal PS2 hardware of the FPGA.

For this activity, it is enough to understand how the PS2 works down to the level of the controller; understanding how each of the internal modules of the PS2_Controller.v file work is unnecessary, but if you're curious you should look at it outside of lab.

The PS2_Interface.v file simply calls an instance of the controller; your skeleton will pass data to the controller through the interface, and the controller will control when its internal modules are communicating with the PS2 core hardware in the FPGA. Then, the interface will output data to be fed into the LCD module. Connect a PS2 keyboard to the FPGA and observe the output on the LCD screen when a key is pressed.

For this part, modify the skeleton so that a given key press will have the correct character displayed on the LCD screen. Choose 4 characters to implement.

Show your TA when you are done. Note, we understand that the LCD module and PS2 are complicated, and expect this to take a bit of time. Start with focusing on getting one character to be correct and expand to 4 once that one is working. A good starting point is to monitor the data being passed to the LCD module from the PS2 interface; the data passed is the scancode of the key pressed on the keyboard.

Clean Up Workstation

Clean up your workstation, making sure you put any broken pieces in the black "discard" bin. Put probes back on the wall, wires back in the wire bin, breadboards on top of the workstation, and all circuit pieces back in their respective bins. **You will not be able to pass this lab if you do not clean up after yourself.**

Grading

- Completing Lab Tasks: 100 points (pass/fail)
- **Due: Friday, Oct 8, 2021 by 11:59AM (Beijing time)**