Intro to Digital Logic, Lab 8
Final Project

Lab Objectives

Now that you are an expert logic designer, it's time to prove yourself. You have until about the end of the quarter to do something cool with the DE1 board and your digital design skills.

Note that this lab is MUCH more complex than previous labs, and will take a LOT of time. Read the entire lab carefully, and START EARLY. You will need the time!

Ground Rules

You will design a significant project on the DE1 board. In some cases you may want to use the breadboard as well – note that all of the pins at the bottom of the breadboard are labeled with the pin they talk to on the FPGA, and thus are usable. This will be most useful to people who want to connect up to more interesting inputs and/or outputs for their design, since all of the digital logic should be done inside the FPGA. You can access these pins by having a bus connection "inout [35:0] GPIO_0" to your top-level module (like you used KEY, SW, LEDR, etc. in other labs). GPIO_0[0] is the leftmost connection (AC18), and GPIO_0[35] is the rightmost connection (AJ21).

You may use the clock_divider circuit if you wish. However, your ENTIRE circuit should be based off of EXACTLY ONE clock. If you need two clock speeds, use the clock divider to generate the faster clock. Then, use a counter as a timer to generate an enable signal at the slower rate – all slower elements will still use the fast clock, but will only change state when the slower counter signal occurs. Take a look at the traffic light controller and the timer for an example.

Grading

Because this lab is significantly more complex than others, it will be weighted much more significantly than a normal weekly lab.

You will be graded on correctness, style, testing, etc (projects 1,3,6,8: 140 points; project 5: 120 points; projects: 4,7: 100 points; project 2: 0 points/not offered this quarter; project 9: TBD, talk to me). Your bonus goals is cool/interesting features. For each of these projects, there are lots of obvious ways to make it more useful, more efficient, and more fun. TAs will give up to 20 bonus points for any of these. However, you will get MUCH more credit for a working, simple design than a not-working, awesome system. **The best plan is to get the basic system working, then add any frills if/when you have extra time.**

Extra Credit - Early Finish

All labs are due by the last day of class (December 11, 2015). Anything turned in after that time will be considered late, based upon the standard class late penalty.

To encourage people to get started early, and leave time to fix things when the inevitable problems come in, all students should attempt to get their projects finished a week earlier (December 4, 2015). There is a 15 point extra credit bonus for finishing your lab on this date, decreasing by 3 points per school day after that. Thus, turning in the lab on the due date gets you 0 extra points, 1 schoolday before that you get 3, etc. Demos are done first-come-first-served in the TA's lab hours – if you show up right as lab hours are ending for the day there is no guarantee the TA will be able to demo it that day, so show up early when possible.

Projects

To keep things interesting, we've listed a large number of different possible designs. You can pick any of these you prefer, and in fact can pick the "Venture Capitalist" design that lets you make up your own project. All of these projects are meant to be approximately the same difficulty to complete, though there are no guarantees that we have guessed correctly on this.

Project 1 – Frogger

The urban horticulture program on campus thinks we can eliminate road-kill accidents by training the local wildlife to avoid cars. Your job is to develop a high-tech traffic simulator so they can learn how to safely cross the road (see Wikipedia under "frogger").

You'll need to have several rows of red LEDs, with moving lights to show where the cars are (they'll likely be some basic pattern, with the cars in a row slowly moving to the right or left). You'll also need a green LED for each location as well, to show the position of your frog. Then, the user should have left/right/back/forward buttons to move their frog around. The goal is to get the frog from one side of the board to the other without being in the same square as a car (squish!). They win, they play again. They lose, they become road-kill-souffle'.

You'll need a reasonable sized board (say 4x6 or so), and the frog should be able to move faster than the cars (otherwise there's no hope).

Project 2 – Mastermind: NOT OFFERED THIS QUARTER

The guys down at SafeCrackersInternational need to practice their skills on electronic locks, and have hired you. Your job is to simulate the ultra-powerful multi-bit encryption locks they'll be facing in the real world.

Your system should pick a random passcode, consisting of a 4-digit number in base-4 (i.e. 0, 1, 2, or 3). Our budding safecracker will then guess the code. If they're right, they win. If not, they die… well, maybe not. These guys aren't the best in the business, so we'll give them multiple tries. But to encourage them, you should tell them how close they came. You'll have to tell them how many digits were correct (right number in right place), and how many digits were misplaced (correct number in the wrong place). Of course, don't tell them which digits were right, just how many of each type there were.

To help grade these burglars-in-training, we'll also want a count of how many attempts they make before they get the code.

Project 3 – Conway's Game of Life

I'm sick of liberal arts majors saying we should get a life – we've got Conway's Game of Life: <http://en.wikipedia.org/wiki/Conway's_Game_of_Life>. This is a simple computation that can give rise to very interesting behaviors over time. You need to implement at least an 8x8 array of the life board, and have ways for a player to input a new pattern and then start it running.

Project 4 – Elevator

The EE building elevator is down again. The faculty and staff have decided that a student should design a system that will work more reliably than the system currently in use. Your task is to make the elevator work. You will need to determine the basic system you would like to implement. You may also want to add features to the current system.

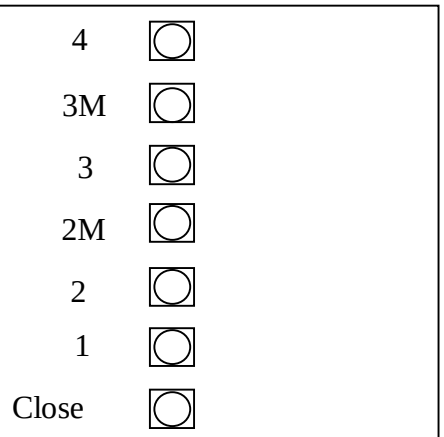

Figure 2: Elevator selection buttons for floor, door and direction.

The system, as it currently stands, will go to the floor specified based on the floors that are pressed and by the direction the elevator is already going. You do not have to worry about calling the elevator. The department has decided that our Undergraduate Advisor will be able to see many more students by running the elevator. What you do have to deal with is a passenger entering the elevator should be able to press the button for the floor he or she wishes to go to.

The EE building elevator has an East and West door. You will not want the elevator doors to open when there is not an exit available, nor if the floor is not one that was selected. However, we assume that on floors with both an East and West door both will be opened.

You will need to consider any possible conflicts. For example, if I get on the elevator with my friend at floor two and both the elevator and my friend are heading down to floor 1, the elevator should continue down to the first floor before heading up towards floor 3. On the same account, for the same situation, if the elevator had been heading up, I would have exited on the third floor before my friend exited on the first.

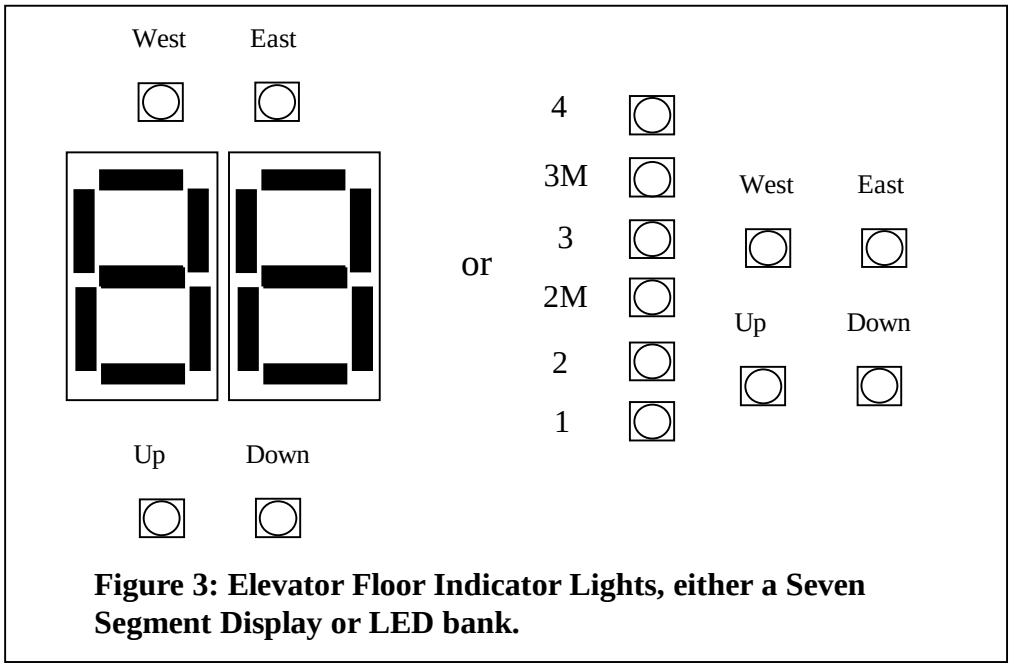

Figure 3: Elevator Floor Indicator Lights, either a Seven Segment Display or LED bank.

Figure 2 and 3 below should give you some idea how you will want to take inputs in as well as how to display your elevator. You will need to make it possible for the user to indicate which floor. You will want to indicate which floor the elevator is currently on as well as the direction it is heading in. When the elevator reaches the destination floor, you will want to show which door(s) are opening. Each time a selected floor is reached, the close door button will need to be pushed in order to allow the elevator to move on to another floor.

The inputs and display outlined above are the minimum requirements for this project. You are encouraged to use your own creativity on this part of the design.

Project 5 – Dancing with your Thumbs

Implement a simplified version of "Dance-Dance Revolution". Your machine will have four banks of lights, each bank with at least 5 lights. The 2nd to top light in each bank is a different color. Each bank of lights also has a button. The system will randomly turn on a light at the bottom of a bank, which will quickly move up that bank (one light on at a time, going from bottom to top). The goal is for the user to press the button on a bank of lights right when the light hits the 2nd to top position. If the user times it right, they get 2 points. If they get it in a position next to that goal position, they get 1 point. Pressing at any other point, or letting the light go off the top, you lose 2 points. Your system should keep track of the score over time, to see how well the player is doing. To make it interesting, multiple lights can go at once, and speed can be adjusted by speeding up/slowing down the machine (manually by the user).

Project 6 – Flappy Bird

The iPhone game Flappy Bird ruined the author's life – why shouldn't it do the same to yours? Flappy bird has our hero the Caped Cardinal (otherwise known as a red dot) flapping through a maze of "definitely NOT a Mario Brothers Ripoff" pipes (otherwise known as green vertical lines). Press the button and the CC goes up, release the button and CC goes down. Avoid the pipes by flying into the holes as you fly sideways through the pipe maze. You need to also

keep track of score, which will be a decimal value of up to at least 999, shown on the hex displays of the board.

Project 7 – D.O.T.lite

The traffic lights are wearing out by U-Village and somebody better fix 'em fast! At the corner of Pend Oreille and 25th Ave (NorthEast corner of campus) there's a 4-way intersection. You'll need to be able to handle traffic on both 25th and Pend Orielle, including the turn lane for 25th. BTW, just in case you thought it was too easy, the turn lights on 25th are different in the two directions, and activated by a sensor in the road.

Note that your Professor drives through that intersection daily, so no 4-way greens!

Project 8 – Snakes & Apples

The experts at WSU have a great idea for dealing with Washington State's bumper crop of apples – feed 'em to snakes! (They thought about making them into tea sets, but everyone knows Cougs can't get their hands on an Apple Cup). Let's see what happens.

Develop an implementation of the snake game (see Wikipedia under "Snake (video game)"). The snake (a green dot head plus 2 body segments) moves around the board looking for apples (red dots). When the head hits the apple, the snake scores a point, and grows longer. If the snake ever runs over its own tail, it dies.

Your system should keep score, and should allow the snake to grow long enough for the game to become challenging. Note that the easiest way to build this game is likely to build it like Tug of War – design a cell for each board position, and have it know how long to keep the light on once the snake head passes over that location.

Project 9 - Venture Capitalist

This is a project of your choice. It must be approved by the venture capitalist guys... that would be your Professor... before you can start. Set up a time to meet with your Professor to discuss your plans well in advance. Note: For the Venture Capitalist if you do not have a written project spec with the Professor's signature on it, in advance, you will not get credit.

Lab Demonstration/Turn-In Requirements

A TA needs to "Check You Off" for each of the tasks listed below.

- Turn in a block diagram of your entire system, and a detailed description of how it should operate (i.e. a User's Manual)
- Turn in all design files for your design, including testbenches of the submodules and the system as a whole.
- Demonstrate your complete system to the TA.
- Tell the TA how many hours (estimated) it took to complete this lab, including reading, planning, design, coding, debugging, testing, etc. Everything related to the lab (in total).