CSE 370 Introductory Laboratory Assignment

Constructing Simple Logic Circuits - II

Assigned: Friday, October 3, 2008
Due: End of Lab Section

Objectives

This second laboratory assignment continues the introduction to the prototyping breadboard and Aldec's Active HDL. We will be using Aldec's Active-HDL as our primary design tool this quarter - both for schematic entry (drawing circuits) and simulating them (verifying that they will operate correctly when we wire them up). In this assignment you will learn how to do both these things for some basic circuits. As you gain more practice with Active-HDL, we will expect you to use this tool to do a large part of your written assignments throughout the quarter. This means, you will see Active-HDL assignments incorporated into your homework, so don't skimp on these labs, they're important!

By the end of this lab you should feel comfortable entering a schematic for a design, setting up simulation stimuli, and verifying that your circuit is correct. The overall goal for this lab will be to construct a circuit which performs the Boolean combination you derive, of course the circuit will have to only use the chips provided for you. By the end of this lab you should know how to construct and verify the operation of a small circuit (3-6 logic gates) in both Active-HDL and on your prototyping board using switches and LEDs as I/O devices.

Since this is another early lab that builds important skills, it's important to work on this lab by yourself. We don't mind if you want to share ideas and tips with each other, but all of the turn-in items should be completed individually.

Tasks

1. Complete Tutorial #1 for Aldec's ActiveHDL. Make sure to complete the entire tutorial. It is quite long but very informative and thorough. Feel free to get a head start on the tutorial before your lab session so you will have plenty of time to complete the lab. Please remember to create a new design and/or workspace for various class assignments, and remember when you name things, do not start the name with numbers, and don't use spaces or odd characters. Stick to letters. This quarter you will be using Active-HDL for both laboratory and written assignments to generate many different schematics. Separating the files into different designs based off the assignment/project will help prevent compiling bugs and help the course staff to diagnose your problem faster.
2. In Chapter 1, problems 1.3 and 1.4, you are asked to realize a circuit to detect the "jack of diamonds". For this lab you will design a circuit to detect the "queen of hearts" using the encoding scheme below and only the gates that are available in your lab kit. The gates are

available for use in your schematics in lib370 as "c74 ". If you don't remember the package number, refer to the lab kit chip map.

Problem 1.3) Encoding for a deck of cards:

The first step to encoding is to determine how many bits we need to encode each piece of data. For a standard deck of cards, we have 13 different values (2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A). So that means we need at least 4 bits to represent the 13 different values. Since two to the forth power gives 16, which is more than 13 so 4 bits should be enough. Finally we have 4 different suits, (Clubs, Diamonds, Hearts, and Spades), which means we'll need 2 bits; since two to the second power is 4, to represent the suits. So the value 0 will be assigned to clubs, 1 to diamonds, 2 to hearts, and 3 to spades. This means we'll need a total of 6 bits to encode an entire standard deck of cards.

We'll be using a specific encoding, so make sure you follow it when you do this lab assignment. First off, we won't be using the values of 0, 14, and 15. Finally the encoding will done in the following order:

$V_3V_2\ V_1\ V_0\ S_1\ S_0$

In general, note that the higher-numbered subscripts denote higher- order (more significant) bits. By convention, bits progress, left to right, from most significant to least significant. So now, the encoding is as follows in table form:

Value/Decimal Value	Binary Encoding
Ace / 1	0001
2 / 2	0010
3 / 3	0011
4 / 4	0100
5 / 5	0101
6 / 6	0110
7 / 7	0111
8 / 8	1000
9 / 9	1001
10 / 10	1010
Jack / 11	1011
Queen / 12	1100
King / 13	1101

Suit/Value	Binary Encoding
Clubs / 0	00
Diamonds / 1	01
Hearts / 2	10
Spaces / 3	11

Problem 1.4) An Example Encoding

As an example for the encoding above we'll encode Jack of Diamonds. Remember, your assignment is Queen of Hearts, so don't build the Jack of Diamonds! So first off we look at the value of the card, Jack corresponds to the value of 11, which means the encoding for the value should be 1011. Next the suit is Diamond, which means the encoding should be 01. So if we put it all together in the format described above we get: 101101. This can be represented as:

V_3 and V_2' and V_1 and V_0 and S_1' and S_0

You will have to design a circuit that will light up a light when the values for Queen of Hearts is entered via the switches, one switch for each bit. Since you have to use only the chips provided make sure your design uses NAND and/or NOR gates since your lab kit does not have AND gates. And remember, the smaller and more efficient the circuit, the easier it is to debug and build!

3. Draw a schematic for your **"Queen of Hearts"** circuit using Active-HDL and the chip parts (parts starting with c74) from the lib370 library. Using stimulators as shown in Tutorial #1, simulate this circuit for all possible input values, and show that your circuit recognizes the "Queen of Hearts". Show your schematic and the result of your simulation to a TA at checkoff before you proceed to step 4. We want to make sure that your logic is correct before you construct the circuit. There is no need to print anything.

   Remember: When you are designing this in Active-HDL be sure to use only the 370 parts in the 370 part library. You will always use the 370 part library! Finally make sure you use the right parts, don't use individual logic gates, instead use the chip parts that are included. In the library virtual copies of the chips in your kit are included. These begin with C74. When you do this it will make building the circuit much easier because you're using the actual chips instead of broken up gates. Lastly, if you're having trouble with the simulation make sure your chips are powered. They need both a VDD and a Ground in order to function properly, even in simulation!

4. Using your design from task 3 you will need to construct a circuit that will identify the queen of hearts on the prototyping board by connecting the inputs to switches and a confirmatory output to an LED. Since we use 6 bits to do the encoding you will need 6 switches. Please use SW0 and SW1 for the suit and SW2, SW3, SW4, and SW5 for the value of the card. Remember since the bits work left to right as most significant, SW5 will be V_3, SW4 will be V_2, SW3 will be V_1, and SW2 will be V_0. Finally SW1 will be S_1 and SW0 will be S_0. Demonstrate your circuit to the TA and make sure everything is working properly. You know it's working properly when the LED only comes on when the input for the Queen of Hearts is given.

Hints: If you have any problems with your circuit you can trace through your circuit with the logic probe to make sure all the values are what you expect.
Since we're allowing you to design your Queen of Hearts circuit, you actually have a little bit of free will in terms of how you want your circuit to react. You can choose to make your circuit active high or active low. Active High means the LED will come on when the Queen of Hearts input is given by the switches, and Active Low means that the LED is always on and will only turn off when the Queen of Hearts input is given. It is up to you how you want to design your circuit, just let your TA know when they check you off.

Lab Demonstration/Turn-In Requirements

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

1. Show your schematic and simulation output (Waveform) from Active-HDL from Task 3.
2. Demonstrate your working "Queen of Hearts" circuit to your TA.

Comments to:cse370- webmaster@cs.washington.edu