ECE 152A LABORATORY 1

Preliminaries: The Seven-Segment Indicator

The seven-segment indicator is a display with 7 segments each of which can be lit independently of the other six segments. This is shown in Figure 1.

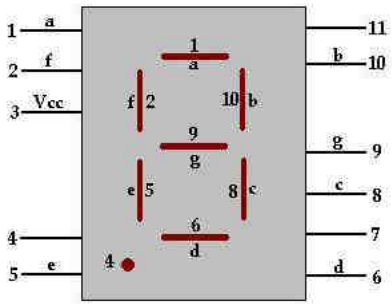

Figure-1

This indicator can be used to display the digits 0 through 9 (just like on old calculators), or it can be used to display some letters (such as on pagers that have alphanumeric characters). In this lab, we will use the seven-segment indicator to display letters.

Objective: Design a circuit that displays the letters A through J on a seven-segment indicator. The circuit has four inputs W, X, Y, Z, which represent the last 4 bits of the ASCII code for the letter to be displayed. For example, if WXYZ = 0001, "A" will be displayed. The letters should be displayed in the following form:

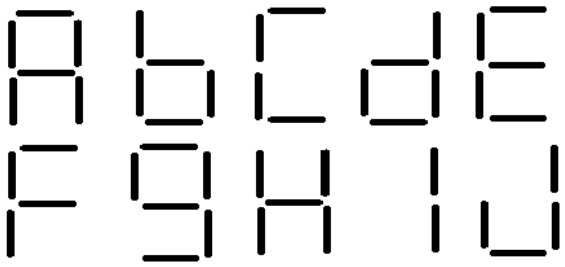

Figure 2

Deliverables and Due Dates:

Lab # 1 Problem Set is due beginning of your lab section. This comprises completing the Steps 1 and 2 below. (See the "Design Steps in Detail".)

In addition, you HAVE to come to the lab having read the data sheets for this lab. Please go to the course web page to read and print the TTL data sheets that you may need to refer to for this lab.

Lab # 1 check-out is due during your lab section in the following week. This comprises completing all of the remaining steps in this lab and demonstrating the correct functionality of your final circuit.

Grading will be done as follows for steps completed on time:

Step 1:	20 %	(due the beginning of Lab # 1 session)
Step 2:	20 %	(due the beginning of Lab # 1 session)
Step 3:	20 %	(due the following week: check-out)
Step 4:	40 %	(due the following week: check-out)

NOTE: PLEASE SKIP STEP 3, IF YOU FIND THAT THE MODELSIM SCHEMATIC EDITOR IS NOT WORKING ON THE DIGILAB MACHINES. CONSULT YOUR LAB SECTION T.A. (STEP 3 WILL NOT BE GRADED IN THIS CASE.)

Design Strategy (Outline):

In this lab, we will carry out the above design objective in several stages:

Step 1: **Map word problem to Boolean logic and minimize logic**

We map the word problem to Boolean logic. The result of this step is a set of logic equations, one for each output of the circuit. Then, you will minimize these equations using Karnaugh maps.

Step 2: Map Boolean logic to Schematic

We map the Boolean equations to a schematic, using the schematic tool Xilinx, Project Navigator, Schematic Editor.

Step 3: Simulation (ModelSim)

Simulation tools are computer-aided design tools that can take a circuit schematic as input, simulate the circuit and generate waveforms for the outputs of the circuit. We simulate our design before implementation in order to catch any errors in the design at an early stage. Once the simulation is seen to be correct, we will be ready to implement the circuit.

Step 4: TTL Implementation

Finally, we implement this design using TTL chips. An example of a TTL chip is shown in Figure 3 below. This chip 74LS00 has 4 2-input NAND gates in it. In order to make this chip functional, it is necessary to connect the VCC terminal to the VDD (e.g. 5 Volts) on your breadboard, and connect the GND terminal to GND on your breadboard.

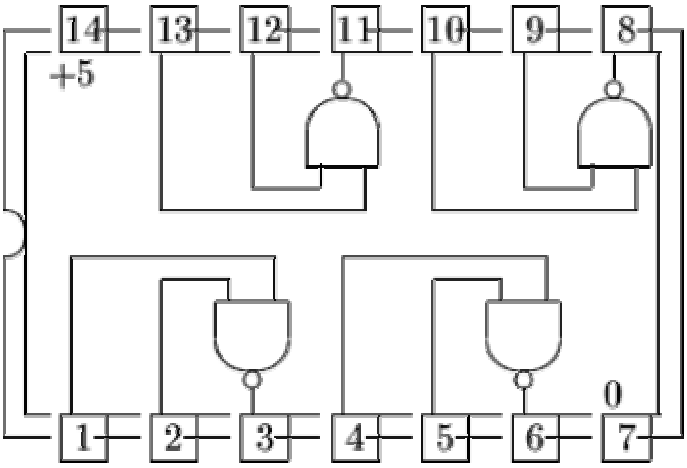

Figure 3. 74LS00 : Quad 2-input NAND gates.

Design Steps In Detail:

Step 1: Map word problem to Boolean logic, minimize logic

In this step, we map the word problem to Boolean logic.

1.1. What are the inputs to this design? What are the outputs? (Keep in mind that your output is the 7-segment display so make sure to label all seven segments separately as shown in Figure 1.)

1.2. Write down the truth table for this design. For example, the input WXYZ = 0001 will refer to "A" so the output in 7 segments is "1110111".

1.3. By using the truth table, write down a Boolean equation for each output (in any form you wish).

1.4. By using a Karnaugh map for each output variable, find a minimum sum of products expression for each of the outputs.

Step 2: Map Boolean logic to Schematic

Using Xilinx Project Navigator's Schematic Editor, draw a schematic diagram to implement the digital circuit built in Step 1.4. Note that in Step 1.4, your solution uses AND, OR and NOT gates. Before you map this solution to a schematic, you need to think about which TTL chips will be used for your design. For example, if you think you will be using NAND gates, then you will need the TTL chip that contains NAND gates (i.e. the TTL chip 74LS00).

At this point, take a look at all gates that are available and decide what chips you will need for this implementation. Then, convert the Boolean equations that you have from Step 1.4 to a Boolean circuit schematic that is implementable using the TTL chips you have in mind. Then, draw the schematic using the Schematic Editor tool.

Note: TAs will be providing assistance on how to use Xilinx Schematic Editor tools. There is also a tutorial xilinxsoft.pdf for Xilinx and Modelsim.pdf which can be found on the course web page.

Step 3: Simulation (Modelsim)

Before simulating your design with Modelsim, we need to create a test bench in order to check for its correct functionality. From Xilinx, Project Navigator, create a new source (test bench waveform) and fill in the correct output values. Once this is complete, you can save and exit. You will use the test bench waveform to compare with Modelsim waveform in order to verify the correctness of the design. You will use Modelsim to generate a simulation of your design. Verify that it exhibits correct operation on every input pattern.

Step 4: TTL Implementation

First, decide where you will place components on the breadboard. On paper, you may make a drawing that shows how the gates in Step 2 will be mapped to TTL chips in this step. On paper, lay out where you TTL chips will go.

Then, wire this circuit on the breadboard. Finally, verify correct operation by testing the circuit for every input combination.