

# Laboratory Exercise 1

## Building Circuits using 7400-Series Chips

Revision of September 13, 2021

The purpose of this lab is to illustrate the process of building logic circuits by using chips that contain individual logic gates. Although circuits are no longer built this way in industry, it is useful to show how discrete gates are connected together to form a logic function. This is also a useful skill to have when making projects using boards such as Arduino, which often use a breadboard and discrete components.

In future, you will be writing Verilog code to describe circuits, but **you should always have in mind that the circuits will be constructed as you see in this first lab.** This mindset is critical to being successful at writing **working** Verilog code for hardware.

This lab exercise will have two components:

1. You will build some circuits using a simple schematic editor and simulator. This should be done before doing the in-lab component. We will use *logisim* and a library of 7400 chips that is available for *logisim* and you will wire up the chips in the *logisim* simulator. Note that you will not have to worry about wiring up power to these simulated chips!
2. There will be an in-lab component where you will build the circuits you simulated in the first component, but now using the actual physical 7400 chips. **You will have to worry about applying power to the real chips, i.e., make sure each chip has a connection to 5V and GND. If your circuit is not working, that's the first thing to check.**

If you want to install *logisim* on your own machine, read the *Guide to Tools for ECE241/ECE253*.

## 1 Lab Books

We encourage you to do your preparations in a **lab book**. This is good practice and will help you keep organized. In industry, you are often asked to keep your notes in lab books because this helps to document your work for various kinds of reporting and as evidence when filing patents where it is important to show dates and times of when ideas are conceived. This does not need to be a new book; feel free to use an existing lab book or keep an electronic

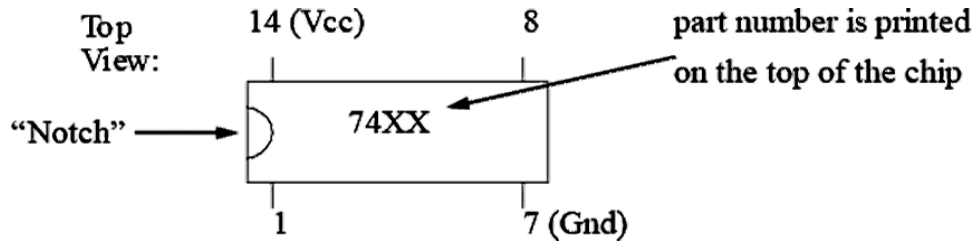


Figure 1: Typical DIP Package

notebook. What is important is that your notes are clear and legible and you can show and explain your work to a TA if asked.

## 2 7400-series Chip Packages

The chips that you will use in this lab are Small Scale Integration (SSI - meaning there are not many transistors in a single chip) 7400 series devices.

All of the chips you will use are *Dual In-line Packages* or DIPs. Most of the packages are 14 pins, and the pins are numbered from looking at the chip from the top: With the notch on the left end, then below the notch are pins 1 to 7, and above the notch are pins 14 to 8. NOTE: Pin 14 must always be connected to VCC (+5V) and pin 7 to ground (0V). Figure 1 shows the layout of a typical DIP package.

## 3 Preparation

Design all of the circuits in both Parts I and II using **only 7404 (NOT), 7408 (AND) and 7432 (OR)** series chips, as shown on the attached sheets. Choose the actual pin numbers of the chips that you will use when you build your circuit and show them on your circuit diagram, i.e., *schematic diagram*. This will make the construction of your circuit easier.

For example, to implement the following function:

$$out = ab + (c + b)$$

the **schematic diagram** is shown in Figure 2. Check the correspondence of the pin numbers on the schematic with the pin-out information sheet in Figure 4.

If you want to see the schematic for the DE1-SoC board we will be using in the lab, you can find it [here](#). Every chip and component on the board and all of the connections are shown

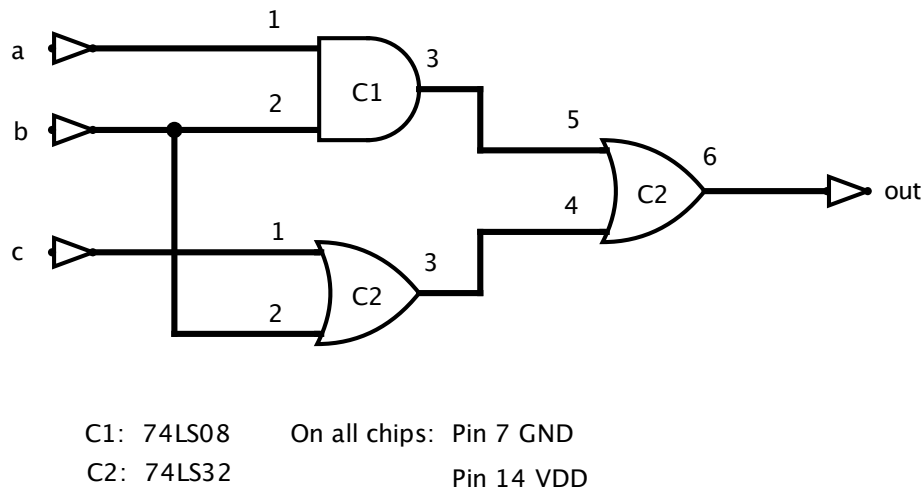


Figure 2: Example Schematic Diagram

on the schematic. A board designer will use the schematic to physically layout the board and draw where the wires should go on the board.

Note that you do not need to draw the entire chip; you only need to label which chip you used and which pin number on the chip you used. Each chip has a unique label, C1 and C2 in this case, and there is a legend to indicate the type of chip. This will be handy when you have larger circuits where you will have several chips of the same type. We can see that the AND gate (C1) uses one **7408** chip using pins 1 to 3 and one OR gate (C2) **7432** chip using pins 1 to 6. The power and ground connections are shown separately.

In each part below, show all of the steps required to go from the specification given to the final circuit, including: assigning variable names to inputs and outputs, deriving a truth table, the logic function, and then a schematic diagram of the final circuit, with pin numbers and chip types.

**Important:** You are allowed to use only the following packages: 7404 (NOT gates), 7408 (AND gates) and 7432 (OR gates). Figure 4 shows the pin-outs for these chips.

### 3.1 Loading the 7400 Library for *logisim*

You will need to download the library of 7400 parts for *logisim*. You can find them at <http://www.cburch.com/logisim/download/7400-lib.zip>. Load and unzip this file in a directory you can find easily. After starting *logisim* you can load the library by going to Project>Load Library>Logisim Library and navigating to the downloaded library file,

7400-lib.circ.

## 4 Part I

The multiplexer is a device that selects one of multiple inputs to be output. The following boolean function is a 2 to 1 multiplexer.

$$f = x\bar{s} + ys$$

As we can see, when the select signal  $s$  is 0, the signal  $x$  is shown at the output. However, when  $s$  is a 1, the  $y$  signal will show at the output. This is an extremely useful circuit with multiple applications such as in a datapath of a CPU that you will be implementing a part of in the future labs.

Perform the following steps.

1. Draw the 2 to 1 multiplexer schematic diagram using only the gates specified in Section 3.
2. Write out the truth table for the design.
3. Using *logisim* and the 7400 library, build the circuit according to your schematic and verify that it matches your truth table. Note that for this part you do not build the circuit using the gates in *logisim*. You instantiate the 7400-lib chip elements and wire up your circuit by connecting to appropriate pins on the chips.
4. Is there a cheaper implementation for your design, i.e., using fewer chips?

## 5 Part II

Build the gate-level implementation for the following Boolean expression:

$$f = \bar{b}(a\bar{c} + \bar{a}d) + a(b\bar{c} + \bar{b}d)$$

Perform the following steps, but **first read through all steps** as you may be able to save some time and effort after reading Step 6 .

1. Draw the schematic diagram for the function shown above using only the gates specified in Section 3.
2. Write out the truth table for the design.
3. Using *logisim* and the 7400 library, build the circuit according to your schematic.
4. Be sure to include the necessary input and output pins so that you can toggle the inputs to the circuit and observe the outputs.
5. Verify that your circuit matches your truth table.
6. Is there a cheaper implementation for your design, i.e., using fewer chips? If you can find a cheaper implementation, show the Boolean Algebra used to arrive at the simpler equation. You may then carry out Steps 1-5 with the simpler implementation, instead of the original form.

## 5.1 Before Going to the Lab

The next steps will be done in the lab. You should read through the next sections to be familiar with what you must do in the lab. If you are done your prep for Lab 1, you might also want to look at Section 9, which can give you a head start with Lab 2.

# 6 The Equipment

In the next parts, we will build the previous circuits using physical 7400 chips in the lab on a breadboard. We begin with a description of the different pieces of equipment you will use: the breadboard, logic probe, wire strippers and digital switch/light board.

## 6.1 Breadboard

The breadboard is for holding and connecting chips. As illustrated in Figure 3, chips are inserted across the middle *valley* in the breadboard. The set of holes in a vertical line above the valley are connected electrically, as are the vertically aligned holes below the valley. So, each pin of the chip in the board is connected to the holes above (or below) the pin. To make a connection to a specific pin, you need only make connections between the holes by plugging the bare end of a wire into the holes above or below the pins.

In Figure 3 the horizontal lines at the top and bottom of the board delineate holes that are connected horizontally; note that the space in the middle indicates a disconnection. The

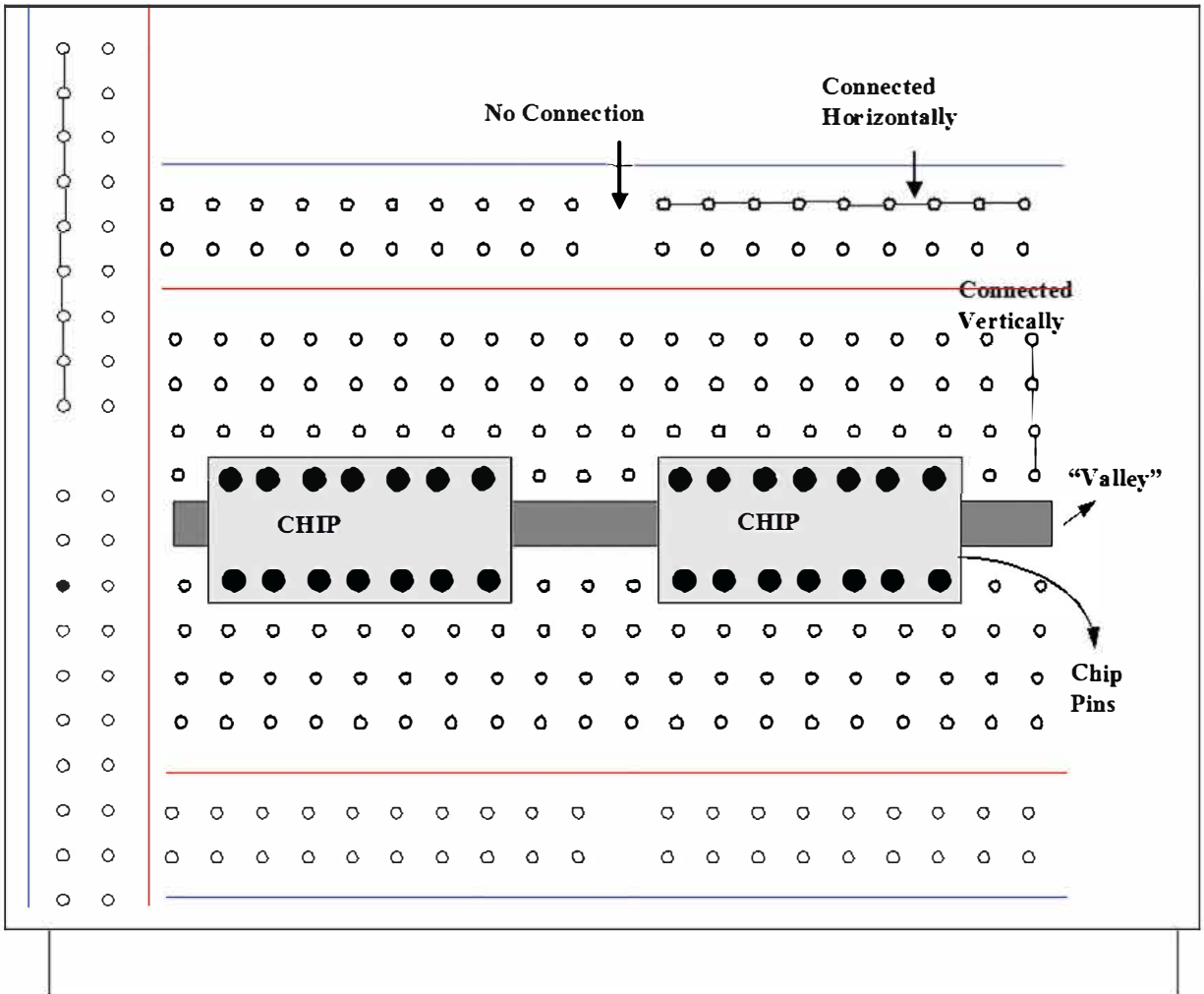


Figure 3: The Breadboard

horizontally-connected holes at the top and the vertically connected holes at the side are usually connected to the power and ground provided by the external connector. The power and ground of the chips are then connected to these strips of holes. The first thing you should do in the lab is connect power and ground to these horizontal and vertical strips.

## 6.2 Digital Switch Board

The digital switch board provides switches that have digital output (**5V = logic 1, 0V = logic 0**) and lights that can be driven by logic signals (logic 1 turns a light on, logic 0 turns it off). Test the board by connecting the switches to the lights. The board also provides a clock, which can have its frequency varied by inserting different capacitors into the holes next to it, and a seven-segment display.

## 6.3 Logic Probe

The logic probe is used for measuring the logic values of signals on the board. Be sure that it has power attached to the correct terminals. To test the probe, touch it to the +5V on the breadboard and ground to ensure that it correctly indicates the values high (1) and low (0) respectively.

Depending on exactly which chip you end up using in the lab you may have to set the logic probe to one of two settings: TTL or CMOS. This setting depends on the type of technology used for the transistors in the chips.

## 6.4 Wire Strippers and Chip Puller

The wire strippers are attached to each workstation to make sure they don't get lost. If you haven't ever stripped a wire, try it!

The chip puller should always be used to remove chips from the breadboard. Doing it with your fingers will bend the pins and ultimately break them, so please don't do this!

# 7 Part III

Wire your design for Part I on the breadboard and demonstrate the functionality to the TA. Your results should match your truth table from Part I.

## 8 Part IV

Wire your design for Part II on the breadboard and demonstrate the functionality to the TA. Your results should match your truth table from Part II.

## 9 Start to Learn Quartus

Starting with Lab 2, you will be using the Intel Quartus Prime tool for building designs on FPGAs. You should go to the Intel University web site at <https://software.intel.com/content/www/us/en/develop/topics/fpga-academic/learn/tutorials.html> and download the Verilog version of *Introduction to Intel Quartus Prime Software (standard or lite)*. Without the board, you can still do everything up to programming, configuring and testing the design on the FPGA.

If you want to install Quartus Prime on your own machine, read the *Guide to Tools for ECE241/ECE253*.



### Pin-out of Selected TTL Chips

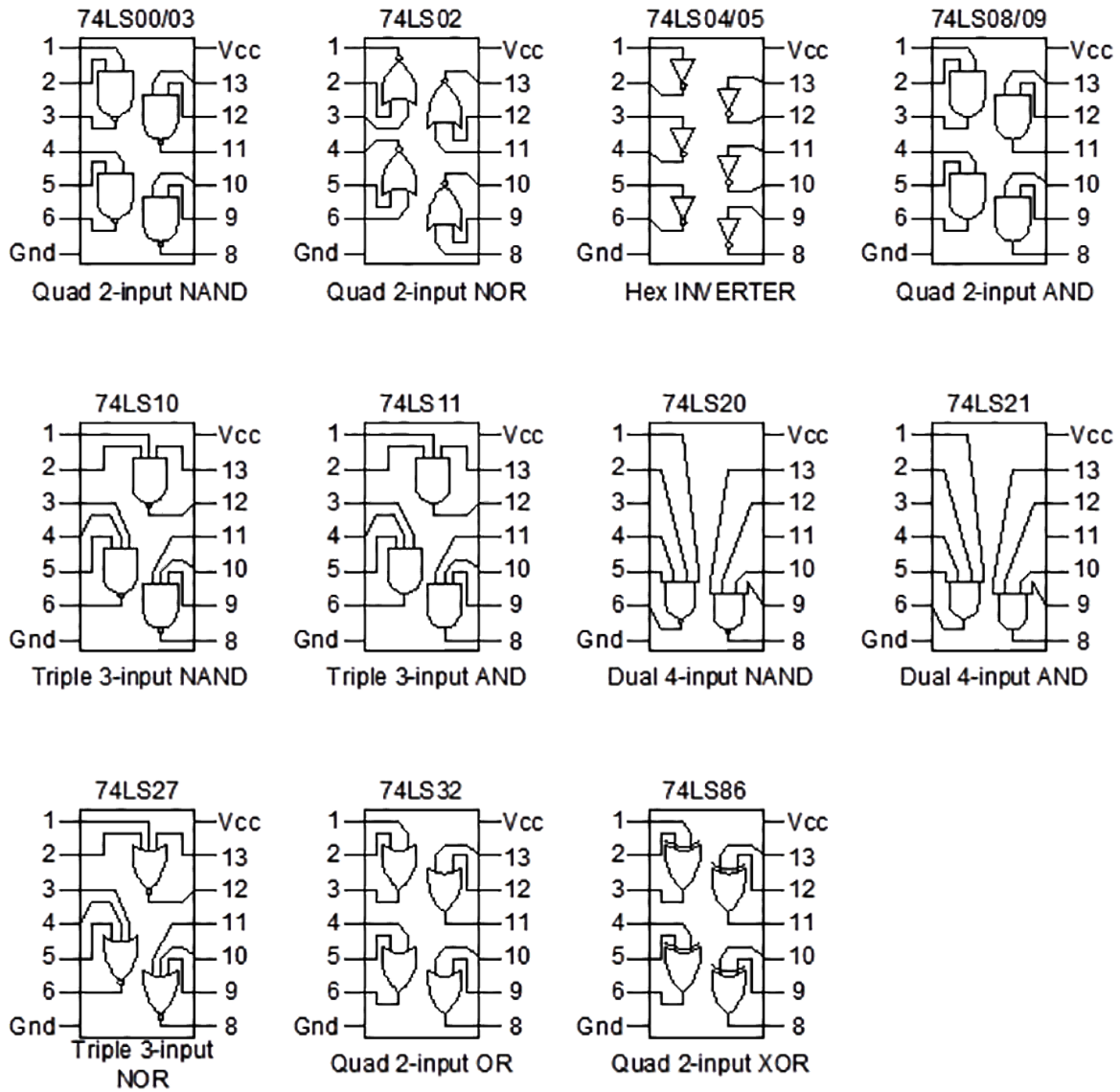


Figure 4: Pin-Out Information for 7400-series Chips

Digital Board Header Pin Assignment									
Pin#	Description					Description	Pin#		
1	Switch #1	o		o		Switch #2	2		
3	Switch #3	o		o		Switch #4	4		
5	Switch #5	o		o		Switch #6	6		
7	Switch #7	o		o		Switch #8	8		
9	Ground	o		o		NC	10		
11	Ground	o		o		NC	12		
13	Ground	o		o		NC	14		
15	Ground	o		o		NC	16		
17	LED #1	o		o		LED #2	18		
19	LED #3	o		o		LED #4	20		
21	LED #5	o		o		LED #6	22		
23	LED #7	o		o		LED #8	24		
25	Ground	o		o		NC	26		
27	Ground	o		o		NC	28		
29	Ground	o		o		NC	30		
31	Ground	o		o		NC	32		
33	Clock	o		o		NC	34		
35	NC	o		o		NC	36		
37	NC	o		o		Pulse Button	38		
39	NC	o		o		NC	40		

Figure 5: Digital Board Header Pin Assignment