Laboratory Exercise 1

Building Circuits using 7400-Series Chips
September 9, 2018

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

Below is a description of the different pieces of equipment you will use: the protoboard, logic probe, wire strippers and digital switch/light board. **BEFORE the lab**, do the preparations in a **lab book**. You are welcome to use one of your existing lab books. During your lab, read through these sections and do the actions.

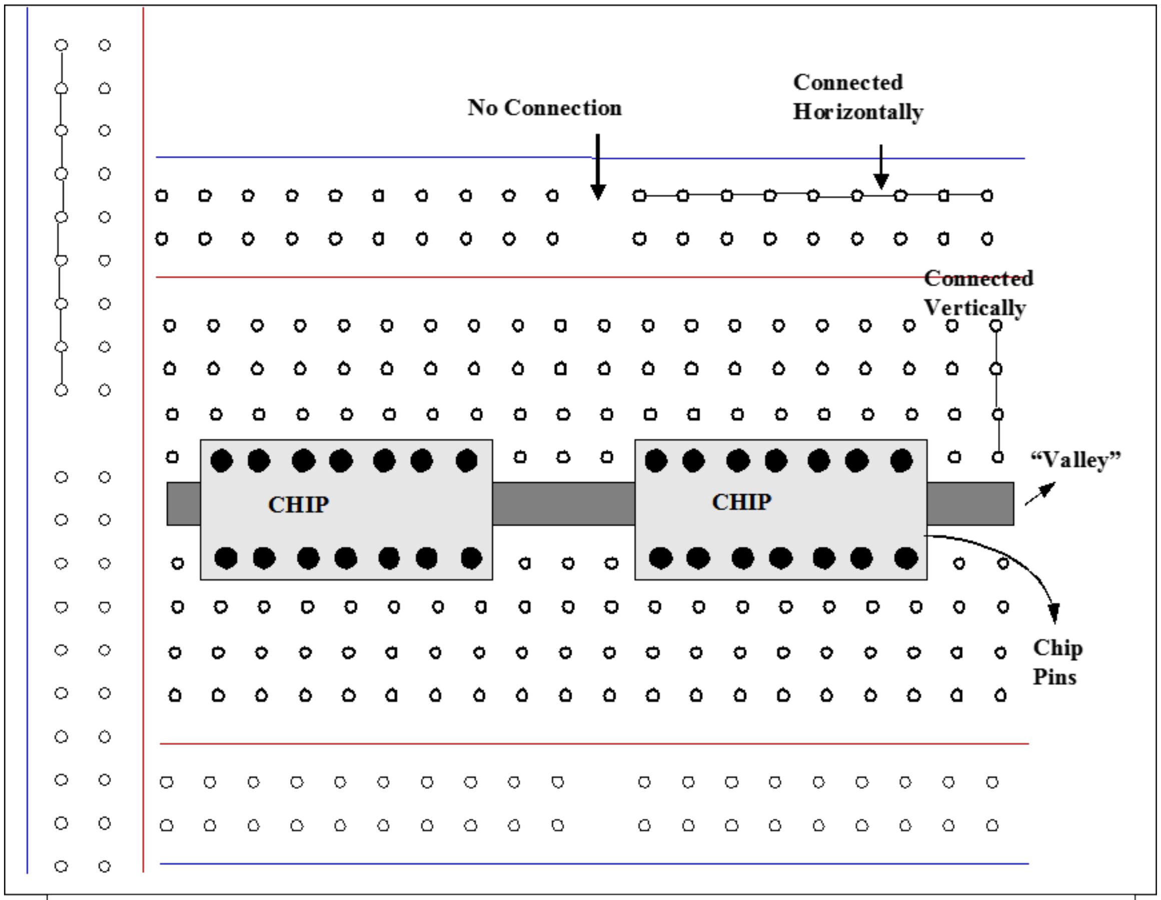


Figure 1 - Picture of the Protoboard

Protoboard:

The protoboard (breadboard) is for holding and connecting chips. As illustrated in Figure 1, chips are inserted across the middle valley in the protoboard. 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 the figure 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 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.

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.

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 protoboard and ground, to ensure that it correctly indicates the values high (1) and low (0) respectively.

Wire Strippers and Chip Puller:

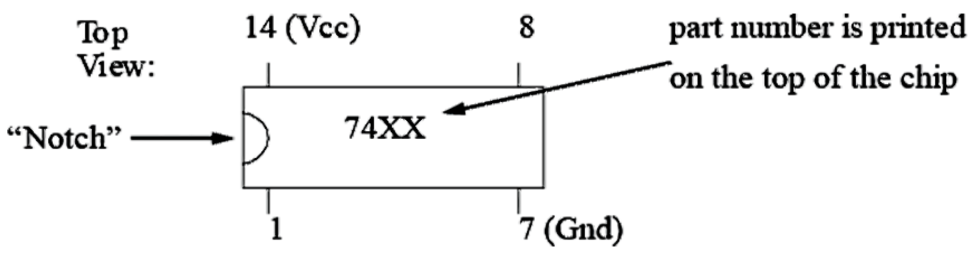
The wire strippers are attached to each workstation to make sure they don't get lost. If you haven't ever stripped a wire, try it!

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

7400-series Chip Packages:

The chips that you will use in this lab are Small Scale Integration (SSI - meaning there's not much logic on a single chip) 7400 series. 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.

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: Below the notch is pin 1 to pin 7, and above the notch is pin 14 down to 8.



NOTE: Pin 14 must always be connected to VCC (+5V) and pin 7 to ground (0V).

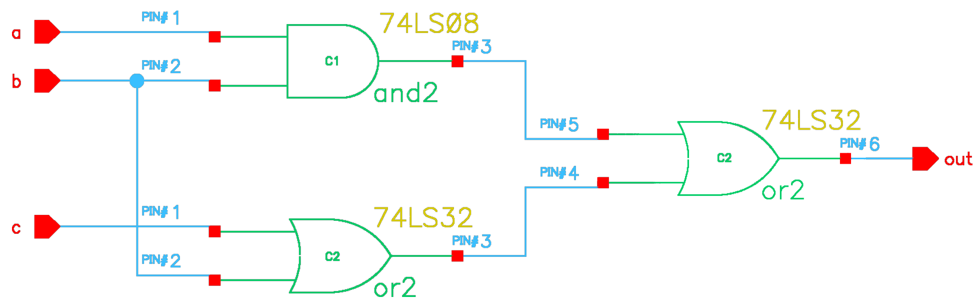
Preparation Before the Lab

Design all of the circuits in both Part I and II using **only 74LS04 (NOT), 74LS08 (AND) and 74LS32 (OR) series chips**, as given 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 - this will make the construction of your circuit easier.

For example, to implement the following function:

$$\textbf{out} = ab + (c + b)$$

The **schematic diagram** for the circuit to implement **out** will look like this:



CHIPS USED:
C1 – 74LS08
C2 – 74LS32

CONNECTED TO ALL CHIPS:
PIN# 7 – GND
PIN# 14 – VDD

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 say what 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 **74LS08** chip using pins 1 to 3 and one OR gate (C2) **74LS32** 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 (see sheet attached): 74LS04 (NOT gates), 74LS08 (AND gates) and 74LS32 (OR gates).

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.

$$\mathbf{f = xs' + ys}$$

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 the gates specified above. Show this design to your TA as part of your prelab to verify that the design is correct.
2. Write out the truth table for the design and show it to the TA as another part of the prelab.
3. Wire your design on the protoboard and demonstrate the functionality to the TA. Your results should match your truth table from the prelab.
4. Is there a cheaper implementation for your design? (using less chips)

Part II

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

$$\mathbf{f = a'bc' + b'(a'c' + ad) + a'b'd}$$

Perform the following steps, but **first read through all steps** as you may be able to save some time and effort after reading Step 4.

1. Draw the schematic diagram for the function shown above using the gates specified in the lab preparation. Show this design to your TA as part of your prelab to verify that the design is correct.
2. Write out the truth table for the design and show it to the TA as another part of the prelab.
3. Wire your design on the protoboard and demonstrate the functionality to the TA. Your results should match your truth table from the prelab.
4. 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 1-3 with the simpler implementation, instead of the original form.

Part III

The Altera University web site at `https://www.intel.com/content/www/us/en/programmable/support/training/university/materials-tutorials.html` has tutorials that you should read on your own. For this part, you can look at *Quartus Introduction Using Schematic Designs*. Note that the latest Quartus tutorial version available is 17.0 and we are using 18.0. For what you are doing, we do not expect significant changes from 17.0 so just use the latest available, 17.0, as of this writing.

For Lab 2, you should do the Verilog version of *Quartus Introduction* (Standard Edition, Verilog) from the Altera university web site so while you are there, you can download the introduction.

Perform the tutorial steps outside of the lab using simulation only.

In this section, you will start on your first Quartus project using the schematic builder. You are to design Parts I and II and compare them with the truth tables from your prelab.

Perform the following steps for each of the logic expressions given in Parts I and II and test the circuits on the DE1-SoC board.

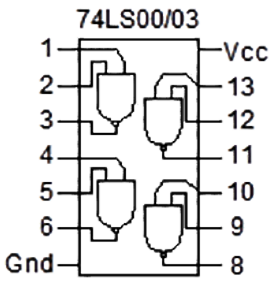
1. Open Quartus and go to File > New... and select New Quartus Project.
2. Click Next and under **Directory, Name, Top-Level Entity** select your working directory and type the name of your project. The top-level design will automatically fill out to be the same name as your project.
3. Click Next until you reach **Family & Device Settings** and select the chip 5CSEMA5F31C6 under Available Devices and then click Finish.
4. Click File > New... again and select Block Diagram/Schematic File. This should automatically open a schematic view window.
5. To place an object click on Symbol Tool (shown as an AND gate) and expand the library till you get to primitives > logic, where you can find all of the gates you need. Place all the gates you need.
6. You will also need to define the inputs and outputs of your circuit. Use the drop down Pin Tool to the right of the Symbol Tool to place three inputs (four in the case of the circuit from Part II) and one output symbol.
7. To connect the symbols together, use the Orthogonal Node Tool (shown as a thin 90 degree corner).
8. Obtain a copy of the `DE1_SoC.qsf` file available under Resources on Piazza and place it in your design directory. This file associates signal names to pins on the chip. If you use these exact signal names for the inputs and outputs in your design, the tool will connect those signals to the appropriate pins. You can examine the file in an editor to see the names and pin numbers.
9. Click on Assignments > Import Assignments... and import the `DE1_SoC.qsf` file.
10. If you open Assignments > Pin Planner, you can see all the assignments of signal names to pin numbers (eg. SW[0] to pin number PIN_AB12).
11. Name the inputs of your design with *SW[0]*, *SW[1]*, and *SW[2]* and your output as *LEDR[0]*. You can do this by double-clicking on the input and output symbols. Note that to implement the circuit from Part II, you will also use *SW[3]*.
12. Once you have completed your design, click Processing > Start Compilation.
13. When compilation is done, click Tools > Programmer and a window will appear.
14. Go to Hardware Setup and ensure Currently Selected Hardware is DE1-SoC [USB-x] and close the window.
15. Click Auto Detect and select *5CSEMA5* and click OK.

16. Double click <*none*> for device *5CSEMA5* and load SOF file (usually under folder "output_files") and device will change to *5CSEMA5F31*.
17. Ensure Program/Configure for device "*5CSEMA5F31* is checked and click Start.
18. Verify that your design is correct by matching it to the truth table from your prelab. That is, you should toggle the switches and observe the expected behaviour on the red LED.
19. Demonstrate the circuit to your TA.

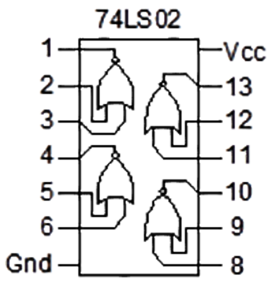
Pin-Out Information for 7400-series Chips and Digital Board

Here are the Pin-out numbers and schematics for all of the chips used in Lab 1:

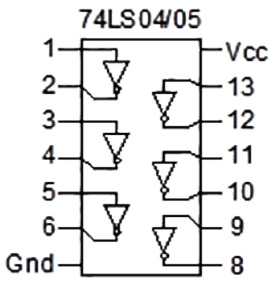
Pin-out of Selected TTL Chips



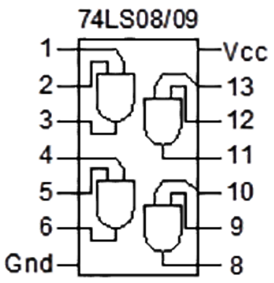
Quad 2-input NAND



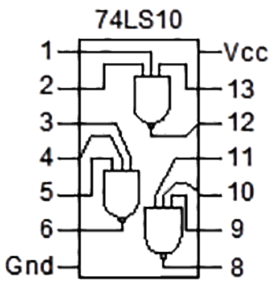
Quad 2-input NOR



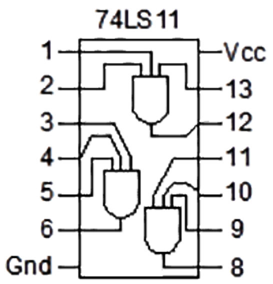
Hex INVERTER



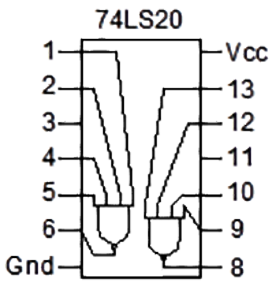
Quad 2-input AND



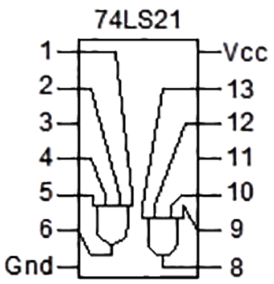
Triple 3-input NAND



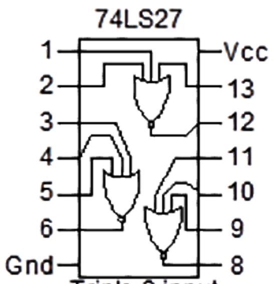
Triple 3-input AND



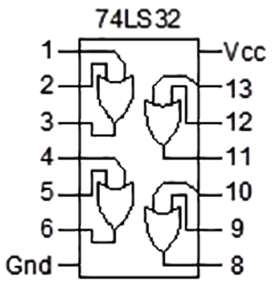
Dual 4-input NAND



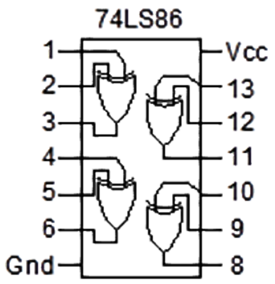
Dual 4-input AND



Triple 3-input NOR



Quad 2-input OR



Quad 2-input XOR

Here is the pin out connections for the header on the digital switch board:

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