

Name _____

EET 131 LAB # 3

This Lab Activity has been designed to familiarize the beginning student to logic gates and IC chips, breadboarding and testing of gates and logic circuits.

- The first thing to know is which gates are inside a given chip number. What are the pin numbers on the chip for gate inputs/outputs, V_{CC} , and GND so that you can run wires to connect them as needed.
- Next you will test gates using *Multisim* Simulation Software.
- Next you will learn to breadboard ICs in order to test how the gates inside them work. You will test various basic gates (using correct chip numbers) and come up with their truth tables through this hands-on activity.

There are related questions at the end of this activity. Give complete answers. Use diagrams if needed for clarity.

OBJECTIVES:

- 1) Familiarization with various logic gate chips and their pin numbers.
- 2) Verifying NOT (Inverter), OR, AND, NOR, NAND logic by:
 - (i) Breadboarding and taking voltage measurements
 - (ii) Using simulation software (*Multisim*)
- 3) Learn to use gates for Enable/Disable purposes.
- 4) Learn to expand gates.

EQUIPMENT REQUIRED:

- ICs: 7400, 7402, 7404, 7408, 7432, 7410
- Digital Circuit Trainer
- Logic Probe
- *Multisim* Software

This lab has two main parts:

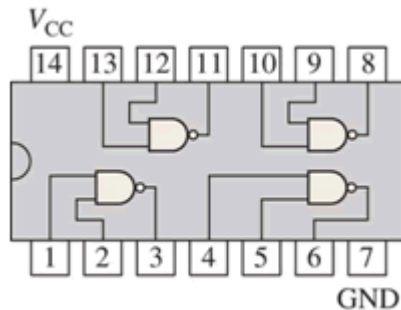
Part 1 is about how to look up pin-outs and other information about any chip from a data book or from a company's web site.

Part 2 is about testing of various logic gates/circuits.

Reading IC Pin Numbers

Reading IC pin numbers correctly is crucial to all lab work. In order to make correct connections in any circuit using ICs, you must be able to identify correct pin numbers, and identify each IC's power and ground pins.

An example using the 7400 chip is shown below. This IC has four 2-input NAND gates inside it. For one of these NAND gates, pins 1 and 2 are the inputs and pin 3 is the output.



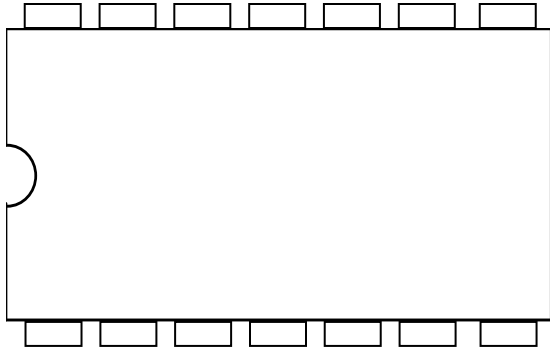
TOP VIEW

No matter who the manufacturer is, pin numbers are standard. The 7400 shown above is in a 14-pin Dual-in-Line package (DIP). The pin numbers are read 1-to-7 on your side (pin 1 is on the left), as shown above, as you read the chip number printed on the top correctly. Pin 8 is right across from pin 7 and you read pins 8-to-14 on the other side, from right to left.

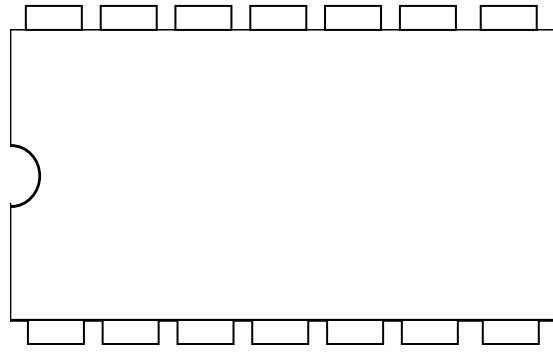
Notice that the notch (or sometimes a dot) on one end of the DIP helps you orient it, as shown above. Sometimes pin 1 may be printed as '1' on the package.

Part 1

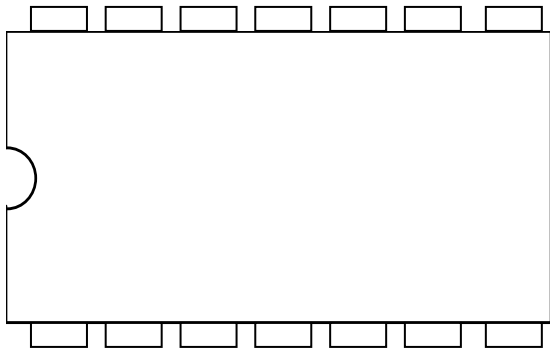
- (i) Using a TTL Data Book , or your textbook, or [www.TI.com](http://www.ti.com), draw gate symbols inside the IC outlines given below. Mark the correct pin numbers on each.



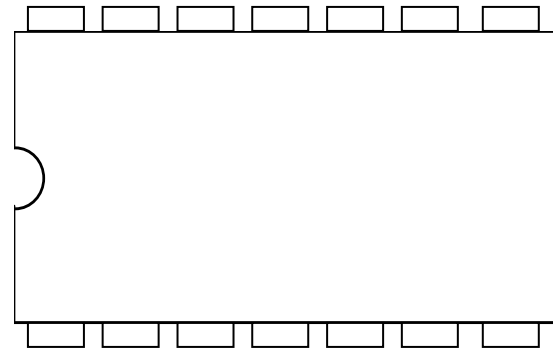
7400



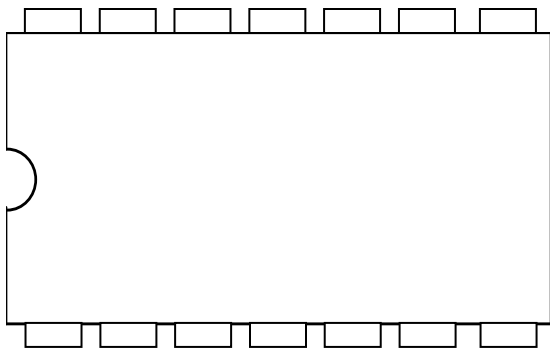
7402



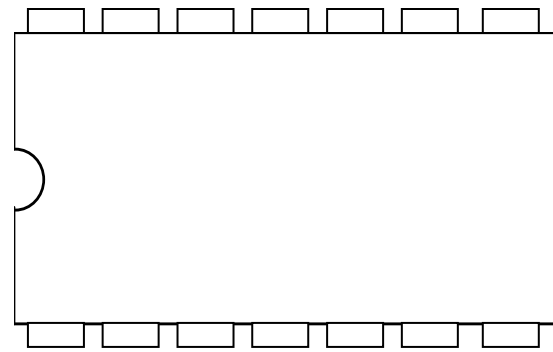
7404



7408



7432



7410

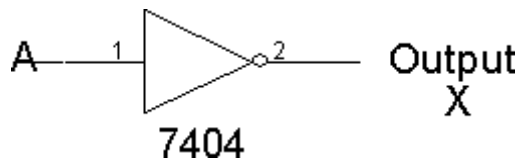
Part 2

Testing various gates / circuits and obtaining their truth tables experimentally.

PROCEDURE:

- For each case below you will breadboard the gate/circuit and test it.
- Use the trainer's data switches to apply HIGH (1) or LOW (0) inputs to the gates.
- Use the logic probe to measure the outputs from the gates.

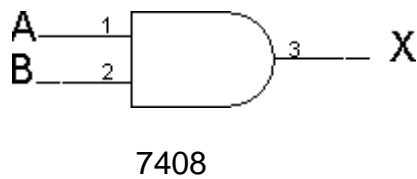
[A] Breadboard the NOT gate and test it. Input **A** is pin 1. Output **X** is pin 2. Complete Truth Table 1.



Truth Table 1

Input A	Output X
0	
1	

[B] (i) Breadboard and test the AND gate. Complete Truth Table 2.



Truth Table 2

A	B	X
0	0	
0	1	
1	0	
1	1	

(ii) Study the AND gate's Enable/Disable function.

Set pin 2 switch LOW (0). Now note the following:

With pin 1 set to HIGH (1), output = _____

With pin 1 set to LOW (0), output = _____

Now set pin 2 switch HIGH (1). Note the following:

With pin 1 set to HIGH (1), output = _____

With pin 1 set to LOW (0), output = _____

NOTE :

A gate is said to be **enabled** when its output depends on the input data. Otherwise the gate is **disabled**.

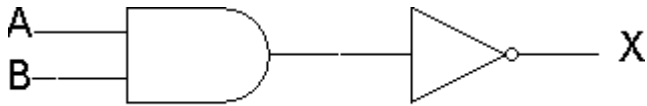
Input data is a string of HIGHS and LOWs applied to an input pin [pin 1 in this case].

Logic level on the control pin [pin 2 in this case] determines if a gate is enabled or disabled.

Whenever a gate is enabled, data gets through (both HIGHS and LOWs).

Whenever a gate is disabled, data is blocked and its output is constant HIGH or LOW depending upon the gate.

[C] In the diagram shown below, label each gate with its **chip number** and **pin numbers**. Then breadboard the circuit and complete Truth Table 3.



Truth Table 3

A	B	X
0	0	
0	1	
1	0	
1	1	

Label the gate shown below with its **chip number** and **pin numbers**. Then breadboard the gate, test it, and draw a complete truth table for it (Truth Table 4).



[D] Draw the symbol for a **3-input NAND gate**.

- Write the chip number and pin numbers on the symbol.
- Draw a truth table (Truth Table 5) for this gate.
- Now breadboard this gate for testing and complete Truth Table 5.

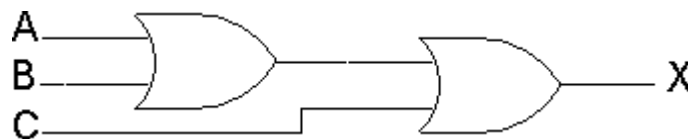
[E] Label the gate shown below with its **chip number** and **pin numbers**. Then breadboard the gate, test it, and complete Truth Table 6.



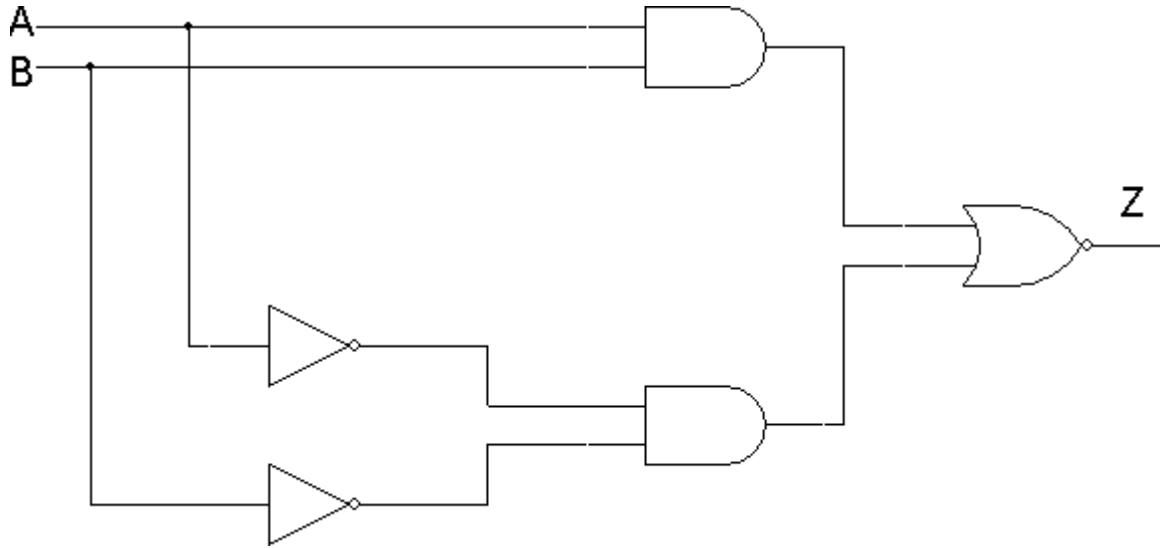
Truth Table 6

A	B	X

[F] In the diagram below, label each gate with its chip number and pin numbers. Breadboard and test the circuit. Draw and complete a truth table (Truth Table 7) for it.



[G] In the diagram below, label each gate with its chip number and pin numbers. Then breadboard the circuit, test it, and complete Truth Table 8.



Truth Table 8

A	B	Z

When your circuit works correctly, ask me to check it.

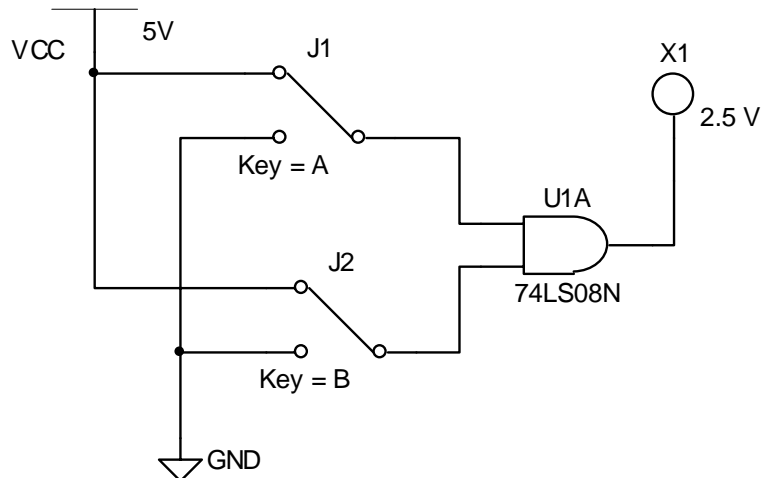
Circuit works correctly? _____ DIPs inserted correctly? _____

Using power/ground busses? _____ Wire colors? _____

Wire lengths? _____ Wire ends trimmed? _____ DIPs accessible? _____

Testing Gates Using *Multisim*

Let's say you want to test a **2-input AND gate** and obtain its truth table in *Multisim*. You will build the circuit as shown below, simulate, and test to complete the truth table:



Truth table

A	B	X1
0	0	
0	1	
1	0	
1	1	

- In the same way, test a **2-input NAND gate** and a **2-input NOR gate** using *Multisim*.
- Get printouts showing your NAND gate and NOR gate circuits. Each printout should include the following typed (not handwritten) information:
 - Your name
 - Course number and section
 - Date
 - Truth table for the gate
- Staple these two printouts to the lab when you turn it in.

Based on your observations, answer the following questions.

1. How many NAND gates are there in one 7400 chip? _____
2. How many NAND gates are there in one 7410 chip? _____
3. Of the following four chips, which one is different from the other three in how the gates are laid out on the chip, and how would you describe this difference?
7400, 7402, 7408, and 7432
4. What are the pin numbers for V_{CC} and GND connections on each chip you used?
5. State in words how an AND gate works.
6. From observations of Part – 2 [B] explain how an AND gate can be used to pass or stop data applied to one of the inputs (pin 1).
7. Now explain how can an OR gate be used to pass or stop data applied to one pin by using the other pin as control.
8. Are the two truth tables in Part – 2 [C] different or identical? Why?
9. In Part – 2 [E], if you didn't have any 7402 chips, which **two** chips would you use to build a circuit that gives the same output?

10. (a) Write the truth table for a 3-input OR gate.

(b) How does the truth table of a 3-input OR gate above compare with that of Part II (F)? Explain.