Remember to bring this along to your lab!

CS2100: Computer Organisation

Lab #6: Introduction to Logic Trainer

(Week 9: 14 – 18 October 2024)

[This document is available on Canvas and course website https://www.comp.nus.edu.sg/~cs2100]

Objectives:

1. To familiarise yourself with the lab equipment.
2. To learn how to use the logic trainer.
3. To learn how to connect a simple logic circuit.

No submission is required for this lab, but attendance is given 5 marks.

IC chip: One 74LS00 chip.

Procedure:

1. Please read the accompanying document "CS2100 Logic Trainer Guidelines" given to you before attending this lab.
2. Your lab TA will explain how to use the equipment.
3. A demonstration on how to connect a simple logic circuit on the breadboard of the logic trainer will be given. Please be punctual for this lab or you may miss the demonstration.

Logic Trainer/Breadboard

(The following text is adapted from Appendix A of "Introduction to Logic Design" by Alan B. Marcovitz).

An IDL-800 logic trainer is shown in Figure 1 below. It consists of a breadboard where small and medium size logic circuits can be connected.

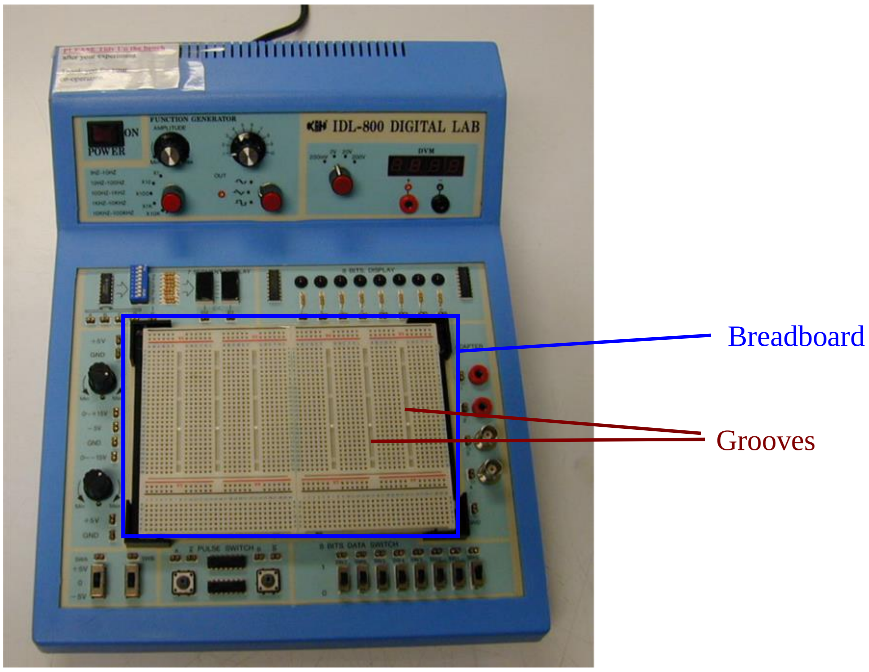


Figure 1. Logic trainer.

Circuits are wired on the breadboard, a small portion of which is shown in Figure 2. The main part of the breadboard has a number of grooves, over which an integrated circuit (IC) chip fits, as shown in Figure 2. Each pin of the chip is inserted into one of the 6 horizontal holes, all of which are connected together internally. Thus, to make a connection to a pin, you insert a wire into one of the remaining 5 holes that belong to the same set of 6 holes.

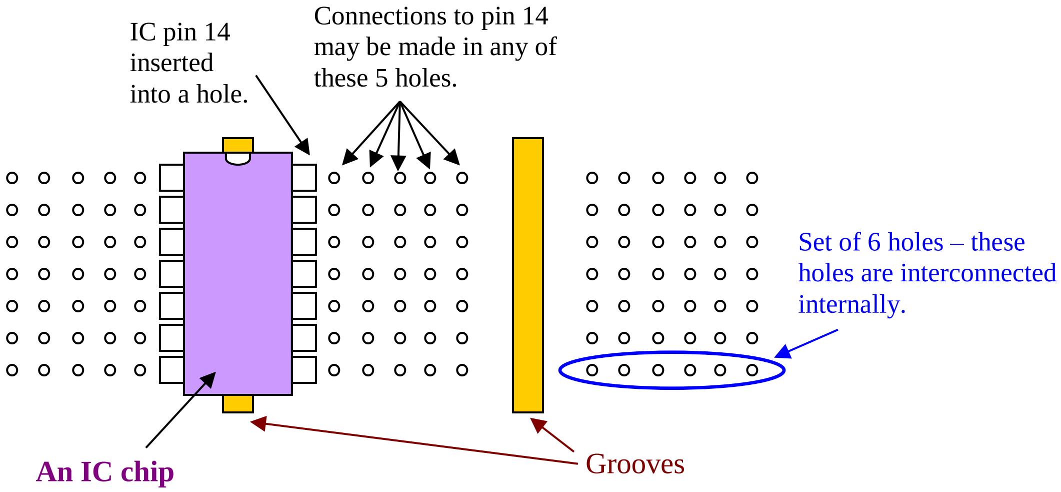
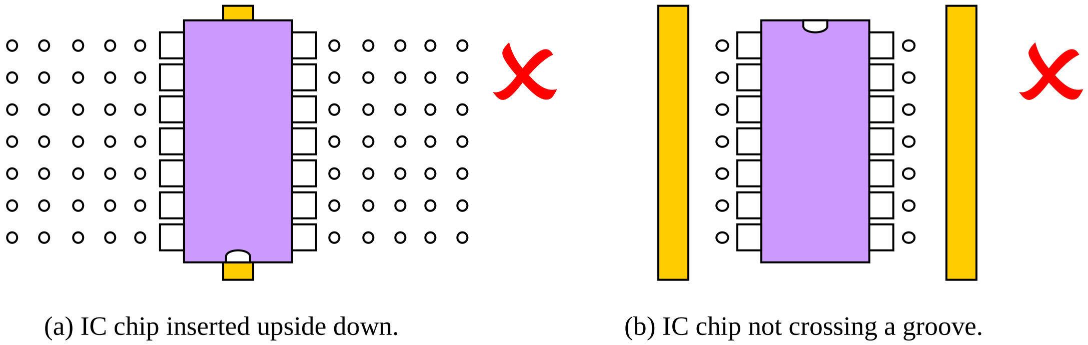

IC pin 14 inserted into a hole.
Connections to pin 14 may be made in any of these 5 holes.
Set of 6 holes – these holes are interconnected internally.
An IC chip
Grooves

Figure 2. Detail of the breadboard.

The IC chip shown here has 14 pins. The orientation of the chip is specified by the semicircle (at the top of the chip). Typically, that is just an indentation in the plastic shell of the chip. Pins are numbered from 1 (at the top-left) to 7 down the left side and then from 8 to 14 up the right side. See Figure 4 on the next page for an example. (If there are more pins, the numbering is the same, starting at the top-left and continuing down the left side and up the right side.)

Common Mistakes

Place the IC chip on the breadboard with its semicircle notch at the **TOP**, not bottom. Otherwise, you may burn the chip when you switch on the power. Also, insert the IC chip **over** the groove, such that a pin on the left side of the chip and a pin on the right side of the chip do not lie on the same set of holes. Figure 3 below shows these two common mistakes. Double-check before you switch on the power.



(a) IC chip inserted upside down.

(b) IC chip not crossing a groove.

Figure 3. Common mistakes.

IC Chip

The IC chip is identified by a code, for example 74LS00. The prefix “74LS” indicates the technology. The last two digits (sometimes 3 digits for bigger chips) indicate the type of gate/device. Sometimes there is a letter suffix, which you may ignore.

For example, the 74LS00 chip contains 4 two-input NAND gates, as shown in Figure 4. Figure 5 shows the truth table of a NAND operation.

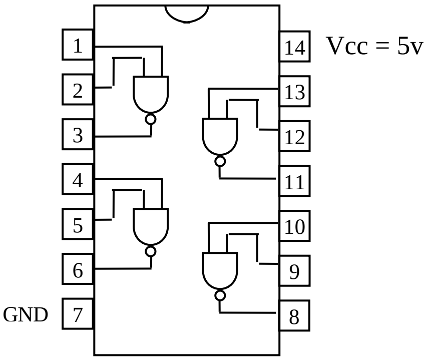


Figure 4. Layout of 74LS00.

Inputs		Output
0	0	1
0	1	1
1	0	1
1	1	0

Figure 5. Truth table for NAND.

Implementing a Function

Consider the implementation of this Boolean function: ***E* = ((*A*·*B*)' · *A*)'**

using only NAND gates. The circuit for this function is shown in the logic diagram in Figure 6.

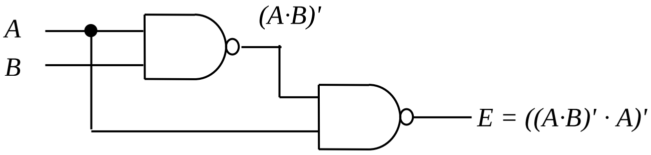


Figure 6. Logic diagram of a NAND gate circuit.

In order to wire this circuit, we need to associate each gate with an appropriate pin on the chip and find the pin number of the chosen pin. The circuit is redrawn in Figure 7 with the pin numbers shown. (These pin numbers are for your own reference. You do not need to write them in a logic diagram.)

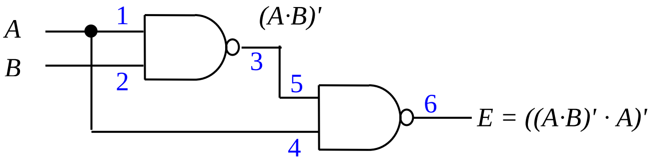


Figure 7. Logic diagram of a NAND circuit with pin numbers shown.

(Please follow the convention used in CS2100. Write ‘·’ for AND and ‘+’ for OR. Penalty will be given if you omit writing ‘·’ for AND. Write A' or $\bar{A}$ for the complement of A, instead of $\sim A$ or $\neg A$.)

Now the connections can be made on the breadboard. The circuit of Figure 8 shows the appropriate connections. Note that it does not matter which one of the five holes is used to connect to a chip's pin. However, only one wire can fit in a hole; thus, when the same signal goes to several points, more holes need to be used. For example, input *A* goes to pins 1 and 4. In Figure 8, input *A* is connected directly to pin 1; then another wire is run from pin 1 to pin 4.

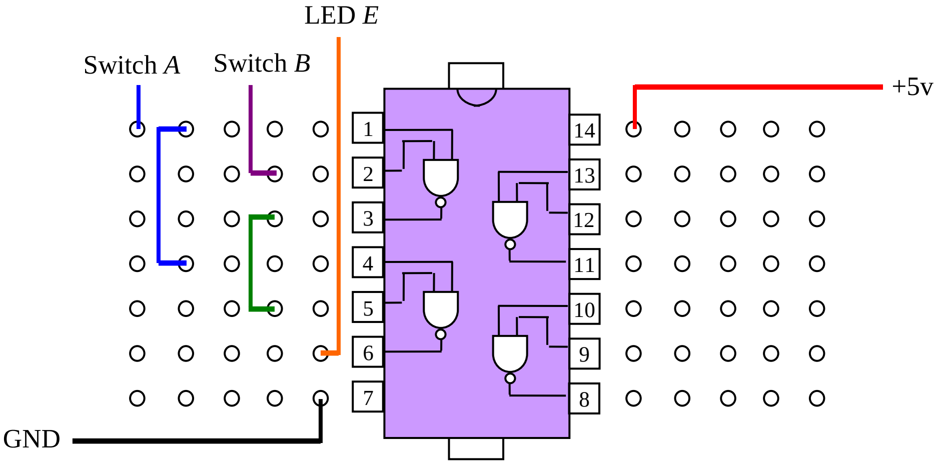


Figure 8. Wiring for the circuit.

Inputs are usually connected to the switches (labelled as SW0 to SW7) on the logic trainer. Outputs are usually connected to the LED lights (0 to 7) on the logic trainer.

Note that in Figure 8, we use wires of different colours for different purposes (blue for input *A*, purple for input *B*, green for the intermediate output, orange for the final output *E*, black for ground, and red for Vcc; refer to the softcopy to view the colours). We use two blue wires because input *A* is connected to both pins 1 and 4. Such colour scheme makes it easier for us to check the connection and make correction if necessary.

After the breadboard is wired, turn on the power and test the system. (Caution: **DO NOT** insert or remove wires or chips while the power is on.)

As an introduction to the system, build this circuit and test it. To test it, start with the two switches all in the 0 position and observe the output light. Repeat for each of the 4 input combinations. Record your observation into the truth table in Figure 9, and verify it using Boolean algebra. Using Boolean algebra, write down the simplified expression for *E*.

Inputs		Output
A	*B*	*E*
0	0	
0	1	
1	0	
1	1	

Write down your simplified expression:

E = ____________________

Figure 9. Truth table for function *E* = ((*A*·*B*)' ·*A*)'