

# Lecture Notes - Introduction to Discrete Digital Logic.

## Basic Logic Gates

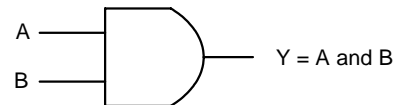
### AND Gate

The notation for the AND of A and B is  $A \cdot B$  or  $AB$ .

Truth Table

A	B	$Y=AB$
0	0	0
0	1	0
1	0	0
1	1	1

Symbol



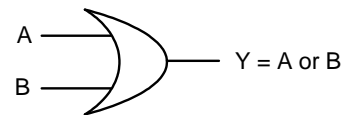
### OR Gate

The notation for the OR of A and B is  $A+B$ .

Truth Table

A	B	$Y=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

Symbol



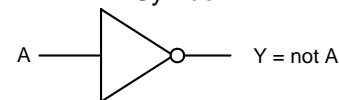
### Inverter

The notation for the COMPLEMENT or NOT of A is  $A'$ .

Truth Table

A	$Y=A'$
0	1
1	0

Symbol



## TTL and CMOS technology

TTL stands for Transistor-to-Transistor Logic. In this lab TTL chips will be used to implement basic logic gates such as AND, OR, NAND and others. On TTL technology gates are made of bipolar transistors (see figure 1). Main characteristics of these gates include their relatively high switching speed, immunity to noise and high consumption of power.

TTL chips can be identified by its code, which will always start with the numbers 74 or 54 followed by some more numbers or characters that will serve to uniquely identify the chip.

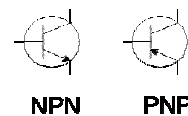


Figure 1. Two kinds of bipolar transistors.

In order to operate this chip it is necessary to take a look at their datasheet. Basic information to notice is the truth table and schematic of the chip as well as maximum and minimum voltages and current rates. Figure 2 shows a sample of this information.

**recommended operating conditions**

	SN5400			SN7400			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V <sub>CC</sub> Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V <sub>IH</sub> High-level input voltage	2			2			V
V <sub>IL</sub> Low-level input voltage			0.8			0.8	V
I <sub>OH</sub> High-level output current			-0.4			-0.4	mA
I <sub>OL</sub> Low-level output current			16			16	mA
T <sub>A</sub> Operating free-air temperature	-55		125	0		70	°C

Figure 2. Example of information from 7400 TTL datasheet

CMOS stands for Complementary Metal-Oxide Semiconductor. Chips implementing gates with this technology use MOS transistors. Figure 3 shows the symbol for these kinds of transistors. In comparison with TTL technology, CMOS power consumption is small, which makes them ideal for most of today's digital applications.

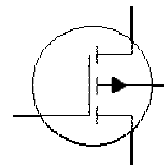


Figure 3. MOS transistor

Another important difference with TTL and CMOS chips is that CMOS can support a wider range of voltage. For instance, while TTL input digital zero levels range from 0 to 0.8 volts, CMOS can go up to 4 volts. Figure 4 shows the voltage ranges for CMOS and TTL.

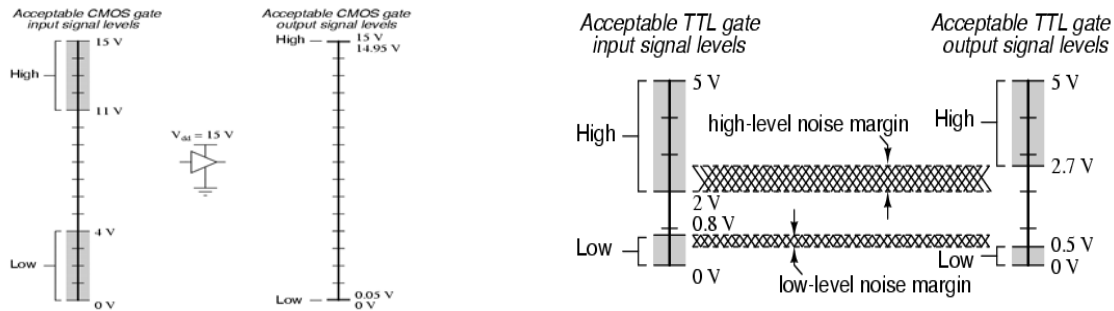


Figure 4. Example of CMOS and TTL voltage levels

## Logic Trainer

A description of basic features for the Logic Trainer (Figure 5) follows:

1. The power switch is in the top left-hand corner. Leave the power off while building and modifying the circuit. If you blow a fuse, notify your TA
2. Sources for +5 Volts and Ground are in the top right-hand corner.
3. Switches for inputs are along the bottom of the Logic Trainer. Up is true (logic 1). Use jumper wires to connect the switches to the main surface area of the Logic Trainer.

4. The main surface area is made up of white plastic pieces with lots of holes in them. This area is where the TTL chips go. See below for details on TTL chip placement. TTL chips are connected with jumper wires to input switches, other TTL chips, and output LEDs.
5. LEDs for output are on the right side of the Logic Trainer. They light for true (logic 1).
6. There is a logic probe that is used to check for the presence of a high (5 volts) or low (0 volts). The probe is your best tool for verifying that you have voltage where you think you have it. Below (figure 6) is an indication of five volts at the test point.



Figure 5. Logic Trainer

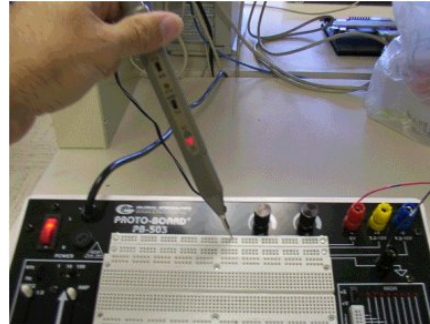


Figure 6. Use of the probe

## Elementary Theorem - Identity

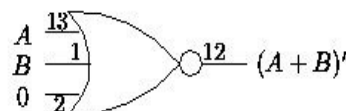
The Identity Theorem specifies that when performing an operation involving any element and the *identity element*, the result is always the original element. For multiplication, the identity element is 1; therefore  $x * 1 = x$ . For addition, the identity element is 0; therefore  $x + 0 = x$ .

## Creating a 2-Input NOR from a 3-Input NOR

This example involves a 2-input NOR function. However, the 2-input function will be implemented using a 3-input NOR gate. In order to determine what the third input of the gate should be we must create a truth table for the 2-input function and a truth table for two 3-input functions one with a third input of 0 and the other with a third input of 1. Comparing the truth table for the two 3-input functions to the truth table for the 2-input function will allow us to find the identity element of the OR operation. The identity element will be the third input of the NOR gate. Here is the truth table for  $(A+B)'$ ,  $(A+B+0)'$ , and  $(A+B+1)'$ .

AB	A+B	$(A+B)'$	A+B+0	$(A+B+0)'$	A+B+1	$(A+B+1)'$
00	0	1	0	1	1	0
01	1	0	1	0	1	0
10	1	0	1	0	1	0
11	1	0	1	0	1	0

Column 3 is the function to be implemented. Column 5 is the result of tying one input of a 3-input NOR gate to 0. Column 7 is the result of tying one input of a 3-input NOR gate to 1. By inspection of the truth tables, a 3-input NOR gate with one input tied to 0 will implement a 2-input NOR gate. Here is the logic diagram.



If you do not know how to convert an AND/OR circuit to NAND or NOR click here for instructions.

## K-Maps

This section will show you how to transition from complex (larger) equations to smaller equations and then to hardware. Let's start with a three variable problem and then progress to a four variable.

### Three Variables

Here is a Boolean function of three variables:  $F3 = (Y + Z') X + X'YZ$

#### a. Truth table for F3

X	Y	Z	X'	Z'	Y + Z'	(Y + Z') X	X' Y Z	F3
0	0	0	1	1	1	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	1	1	1	0	0	0
0	1	1	1	0	1	0	1	1
1	0	0	0	1	1	1	0	1
1	0	1	0	0	0	0	0	0
1	1	0	0	1	1	1	0	1
1	1	1	0	0	1	1	0	1

The most important information that this truth table gives you is contained in the first and last columns. The first column represents each distinct set of inputs. The last column represents the behavior of the output to each set of inputs. The other columns represent an isolated part of the equations and are useful in determining the final output.

#### b. K-map for F3

		Y Z			
		00	01	11	10
X	0	0	0	1	0
	1	1	0	1	1

X, Y and Z are the inputs. The values in the yellow area are the outputs. You fill in the K-Map by placing each output in the box corresponding to its inputs.

#### c. Simplify the Boolean function.

The K-Map will allow you to simplify the function. Combine the adjacent 1's in multiples of two. You cannot combine ones along the diagonal. However, the edges are adjacent, so you can combine ones along the edges.

		Y Z			
		00	01	11	10
X	0	0	0	1	0
	1	1	0	1	1

$$F3 = YZ + XZ'$$

The simplified equations results from two pairs of ones. Each term in the equation is color coded to match the pair that represents it in the K-Map.

**d. Design an AND/OR circuit.**

- a. 3 inputs: X, Y and Z
- b. 1 output: F3
- c. Draw a Logic diagram (figure 7)
- d. Draw a Layout diagram giving the relative position of the chips on the breadboard (figure 8)

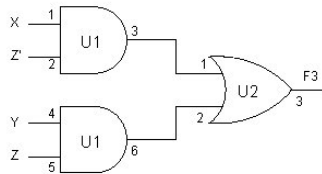


Figure 7. Logic Diagram

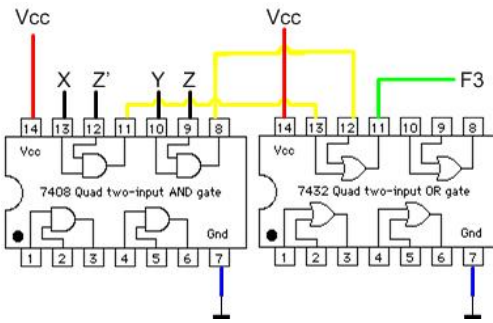


Figure 8. Layout Diagram

**Four Variables**

Here is a Boolean function of four variables: **F4 = B'CE'+ A'B'E'+ AC'E'+ ABC'E**

**a. Truth table for F4**

A	B	C	E	B'CE'	A'B'E'	AC'E'	ABC'E	F4
0	0	0	0	1	1	0	0	1
0	0	0	1	1	1	0	0	0
0	0	1	0	1	1	0	0	1
0	0	1	1	1	1	0	0	0
0	1	0	0	1	0	0	0	0
0	1	0	1	1	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	1	1	0	0	0	0
1	0	0	0	1	1	0	1	1
1	0	0	1	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	0	1	1	1	0	0	0	0
1	1	0	0	0	1	0	1	1
1	1	0	1	0	0	0	1	1
1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0

**b. K-maps for F4**

		C E			
		00	01	11	10
A B	00	1	0	0	1
	01	0	0	0	0
	11	1	1	0	0
	10	1	0	0	1

**c. Simplify the Boolean function**

		C E			
		00	01	11	10
A B	00	1	0	0	1
	01	0	0	0	0
		11	1	1	0
		10	1	0	0

$$F4 = B'E' + ABC'$$

**d. Design a NAND circuit.**

- 4 inputs: A, B, C and E
- 1 output: F4
- Draw a Logic diagram (figure 9)
- Draw a Layout diagram giving the relative position of the chips on the breadboard. (figure 10)

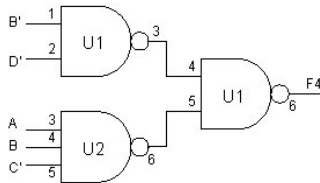


Figure 9. Logic Diagram

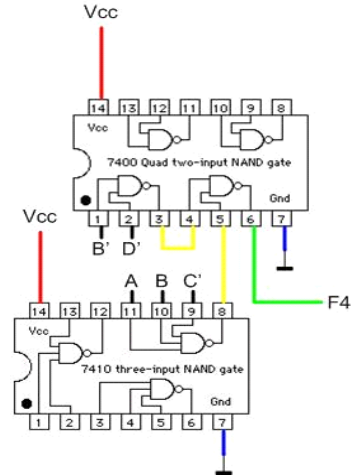


Figure 10. Layout Diagram