Unit 3: Fundamentals of Digital Logic Design I

Introduction

- Digital logic is the foundation for digital computers. Knowing It helps you to understand the innards of computers.
- Digital logic has, relations to other kinds of logic including:
  - Formal logic - as taught by many philosophy departments
  - Fuzzy logic - a tool used to design control systems and many other systems.
  - So, in learning digital logic you learn something that helps you elsewhere.

What are You Going to Learn?

- There are at least two general areas you need to become familiar with.
- First, there's background you need to know - the basics of digital logic - things like zeros and ones (0s and 1s) and how you can represent signals as sequences of zeroes and ones. Eventually you will want to know how large arrays of zeroes and ones can be used in computer files to store information in pictures, documents, sounds and even movies and you'll want to learn about how information can be transmitted, between computers and digital signal sources.
- You will also need to know things about digital circuits - gates, flip-flops and memory elements and others - so that you can eventually design circuits to manipulate digital signals.

- Here is a short list of what you will learn in this:
  - ➢ Learn what logic signals look like
  - ➢ Model logic signals
  - ➢ Learn Boolean algebra for logic analysis
  - ➢ Learn about gates that process logic signals
  - ➢ Learn how to design some smaller logic circuits
  - ➢ Learn about flip-flops and memory elements that store logic signals

Objectives For This Lesson

- Given a system that uses logic signals, be able to specify what the output will be when the input is zero (0) and what the output will be when the input is one (1).
- Given an AND, OR, NAND or NOT gate, be able to determine the output of the gate given the input logic signals.
- Given a system that requires gates, you should be able to wire a chip correctly, and to check that the chip is functioning properly, and many more.

Logic Signals

- There are a number of different systems for representing binary information in physical systems. Here are a few.

i. A voltage signal with zero (0) corresponding to 0 volts and one (1) corresponding to five or three volts.
ii. A sinusoidal signal with zero(0) corresponding to some frequency, and one(1) corresponding to some other frequency.
iii. A current signal with zero corresponding to 4 milliamps and one(1) corresponding to 20 milliamps.
iv. And one last way is to use switches, OPEN for "0" and CLOSED for "1".

- (And there are more ways! : Look out for these as an assignment)

Characteristics of Logic Signals

- Let's pick a voltage signal as a working example. It can take on two values corresponding to 0 and 1.
- We can associate a variable with that logic signal, and we can assign a symbol to represent that variable - like the symbol A.

Lets think Binary!

- Assuming you have some sort of device that generates a logic signal:

1. It could be a telephone that converts your voice signal into a sequence of zeros and ones.
2. It could be the thermostat on the wall that generates a 1 when the temperature is too low, and a 0 when the temperature is above the set point temperature.

- The logic signal, A, takes on values of 0 (FALSE, OFF) or 1 (TRUE, ON). That signal might really be a voltage, a switch closure, etc. However, we want to think in terms of zeros and ones, not in terms of the values of the voltage.

Operations on Logic Signals

- Once we have the concept of a logic signal we can talk about operations that can be performed on logic signals. Begin by assuming we have two logic signals, A and B. Then assume that those two signals form an input set to some circuit that takes two logic signals as inputs, and has an output that is also a logic signal. That situation is represented below.
- The output, C, depends upon the inputs, A and B. There are many different ways that C could depend upon A and B. The output, C, is a function, - a logic function - of the inputs, A and B.

Logic Gates

- If we think of two signals, A and B, as representing a truth value of two different propositions, then A could be either TRUE (a logical 1) or FALSE (a logical 0). B can take on the same values. Now consider a situation in which the output, C, is TRUE only when both A is TRUE and B is TRUE. We can construct a truth table for this situation. In that truth table, we insert all of the possible combinations of inputs, A and B, and for every combination of A and B we list the output, C.

Logic Gates

- Gates are building blocks of digital computer circuits
- A gate is an electronic device that produces **a result** based on two or more input values.
- In reality, gates consist of one to six **transistors**, but digital designers think of them as a single unit.
- Integrated circuits contain collections of gates suited to a particular purpose. Gates are not sold individually; they are sold in units called integrated circuits (ICs)

Logic Gates

- A **chip** (a small silicon semiconductor crystal) is a small electronic device consisting of the necessary electronic components (transistors, resistors, and capacitors) to implement **various gates**.
- The first IC were called SSI chips and contained up to 100 electronic components per chip.
- We now have **ULSI** (ultra large-scale integration) with more than 1 million electronic components per chip.

An AND Example

Let's imagine a physician prescribing two drugs. For some conditions drug A is prescribed, and for other conditions drug B is prescribed. Taken separately each drug is safe. When used together dangerous side effects are produced.

Let :

A = Truth of the statement "Drug 'A' is prescribed.".

B = Truth of the statement "Drug 'B' is prescribed.".

C = Truth of the statement "The patient is in danger.".

Then, the truth table below shows when the patient is in danger.

AND GATES

- An AND function can be implemented electrically using a device known as an AND gate. You might imagine a system in which zero (0) is represented by zero (0) volts, and one (1) is represented by three (3) volts, for example. If we are going to use electrical devices we need some sort of symbolic representation. There is a standard symbol for an AND gate shown in fig 1(slide 27).

- Often in lab work it's helpful to use an LED to show when a signal is 0 or 1. Usually a 1 is indicated with an LED that is ON (i.e. glowing).

Practical: Simulation experiment

Note the following in the simulation (and you can use this in your lab experiments).

- To get a logical zero, connect the input of the gate to ground to have zero (0) volts input.
- To get a logical one, connect the input of the gate to a five (5) volts source to have five volts at the input.
- Each button controls one switch (two buttons - two switches) so that you can control the individual inputs to the gate.
- Each time you click a button, you toggle the switch to the opposite position.

Question1

- Q1. You have an AND gate. Both inputs are zero. What is the output?

- We now have two ways of representing an AND gate, the truth table and the circuit diagram. However, there is a third way of representing this information - a symbolic way - that will take us toward Boolean algebra.

- Let us consider our variables, A, B and C to be algebraic variables, but algebraic variables that can only take on two values, 0 and 1. Then we represent the AND function symbolically in either of two ways.

$$C = A \cdot B \quad \text{or } C = AB$$

- Some will prefer always to insert the dot between the variables so that the AND operation is clearly indicated. Many times, the context will allow you just to use AB, without a dot between A and B, but if there is a variable named AB, then confusion can arise.

Problems

- Assume we have an AND gate with two inputs, A and B. Determine the output, C, for the following cases.

i. P1. A = 1, B = 0
ii. P2. A = 0, B = 1
iii. P3. If either input is zero, what is the output?
iv. P4. A = 1, B = 1

AND GATE

Once we introduce Boolean variables, we can rethink the concept of a truth table. In the truth table 3.1, if A ,B and C) are truth tables and we have an AND gate with A and B as inputs and C as the output, the truth table would look like this.

The symbol for an AND gate is shown below Fig 1. in slide

INPUTS		OUTPUTS
A	B	C = (AB)
0	0	0
0	1	0
1	0	0
1	1	1

TABLE 3.1 The Truth Table for AND Gate

OR Gates

- Consider a case where a pressure can be high and a temperature can be high. Let's assume we have two sensors that measure temperature and pressure. The first sensor has an output, T, that is 1 when a temperature in a boiler is too high, and 0 otherwise. The second sensor produces an output, P, that is 1 when the pressure is too high, and 0 otherwise. Now, for the boiler, we have a dangerous situation when either the temperature or the pressure is too high. It only takes one. Let's construct a truth table for this situation. The output, D, is 1 when danger exists.

INPUTS		OUTPUTS
T	P	D
False	False	False
False	True	False
True	False	False
True	True	True

What we have done is defined an OR gate. An OR gate is a gate for which the output is 1 whenever one or more of the inputs is 1. The output of an OR gate is 0 only when all inputs are 0.

OR GATE

In terms of Boolean variables, the truth table for an OR gate looks like this. (Table3.3)

The OR operator is often referred to as a **Boolean sum**. The expression “A+B” is read “A OR B”.

OR AND gate is shown below Fig 1. in slide

INPUTS		OUTPUTS
A	B	C = (A+B)
0	0	0
0	1	1
1	0	1
1	1	1

TABLE 3.3: The Truth Table for OR Gate

Problems

Assume you have an OR gate with two inputs, A and B. Determine the output, C, for the following cases.

i. P5. A = 1, B = 0

ii. P6. A = 0, B = 1

iii. P7. If either input is one, what is the output?

NOT GATES

- The third important logical element is the NOT gates called inverter. An inverter does pretty much what it says. If the input is 0, the output is 1. Conversely, if the input is 1, the output is 0. The symbol for an inverter is shown below Fig 1 in slide.
- The truth table for an inverter is pretty simple since there is only one input. Call the input A, and the output C, and the truth table is:

Input	Output
A	C
0	1
1	0

-

Example Problem

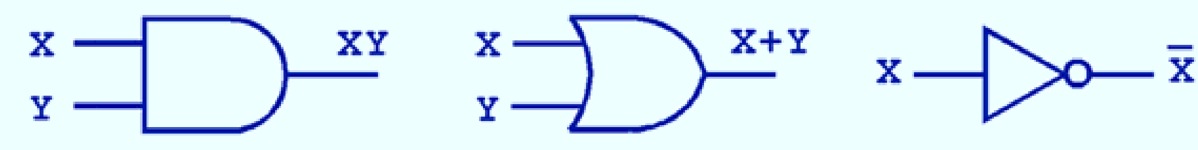
- You need to control two pumps that supply two different concentrations of reactant to a chemical process. The strong reactant is used when pH is very far from the desired value, and the weak reactant when pH is close to desired.
- You need to ensure that only one of the two pumps runs at any time. Each pump controller responds to standard logic signals, that is when the input to the pump controller is 1, the pump operates, and when that input is 0, the pump does not operate.
- You have a bunch of two-input AND gates (IC chips), OR gates and Inverters, and you need to design a logic circuit to control the pumps. You can generate a signal that is 1 when Pump S is ON, and 0 when Pump W is ON. Can you design the circuit?

Solution

In order to solve the problem, consider that the pump controls should receive logical inverse signals. When one pump signal is one, the other is zero. Given that recognition this circuit should work. Here, if X is 1, Pump S pumps, and if X = 0, .

Symbols for Logic Gates

- The three simplest gates are the AND, OR, and NOT gates.



X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT X

X	$\overline{X}$
0	1
1	0

Fig 1: The Three Basic Gates