Digital Circuits and Systems

Prof. S. Srinivasan
Department of Electrical Engineering
Indian Institute of Technology, Madras

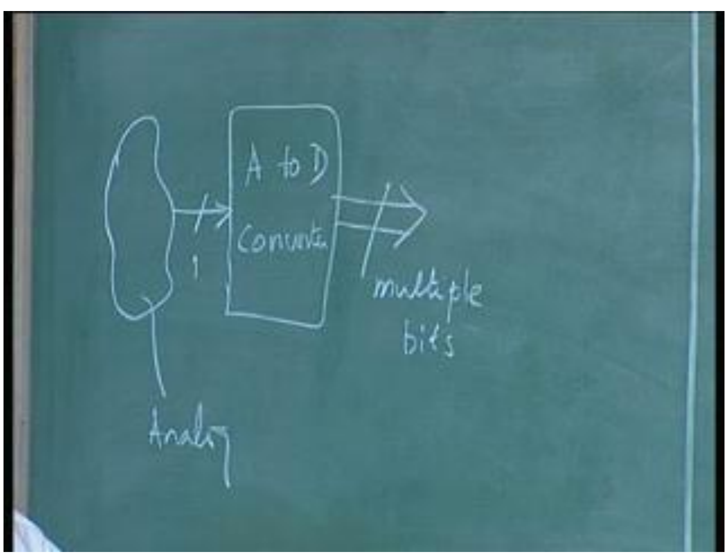
Lecture –3

Combinational Logic Basics

We defined digital signal as a signal which can only take discrete levels and in order to realize it using simple devices like switches we restricted to two levels 0 and one. In terms of voltages that will be 0V and 5V as a general thing but nowadays we have other voltages also. Some other practical devices use these two levels as 0V and 5V. So we have a two level digital signal also called binary signals. but since we have only two levels we cannot represent a very precisely or accurately so we wanted to have more of these binary digits so any given signal can be represented by several binary digits and binary digits are also called bits. So a signal can be represented by several bits. Number of bits depends on the precision or accuracy with which you want to represent the signal, this we have seen earlier.

If you remember when I we drew that analog to digital converter diagram in the last lecture I said the analog real world signal and I said one line is going, analog signal is a single variation that can take any number of values any value between the two limits. On the other hand when I said digital analog to digital converter if you remember I used a thick arrow here (Refer Slide Time: 4:43), thick arrow do you remember this? I told you I will tell you the reason that is what it is.

(Refer Slide Time 6:09)



Now analog signal can take any value over a range. Digital is discrete levels and in order to make it simple or in order to be able to realize this digital signals with simple devices like switches for example we restricted to two levels so they became binary signals but if we realize that in binary signals two levels are not enough to represent all the variations that a signal can have so we decided to have several of these binary digits or bits to represent each signal and the number of bits depends on the accuracy and the computational complexity you want to have, accuracy you want to have etc.

So this is a single line, these are multiple lines (Refer Slide Time: 5:42) in this thick arrow there are multiple bits we don't know the number it can be 1, it can be 2, it can be 4, it can be 8 usually it is integral power of 2 not necessarily though there is no such restriction but generally we will find a 8 bit processor, we will buy a 16 bit processor, we will have a 32 bit processor and so forth. Hence these are the type of things we will have.

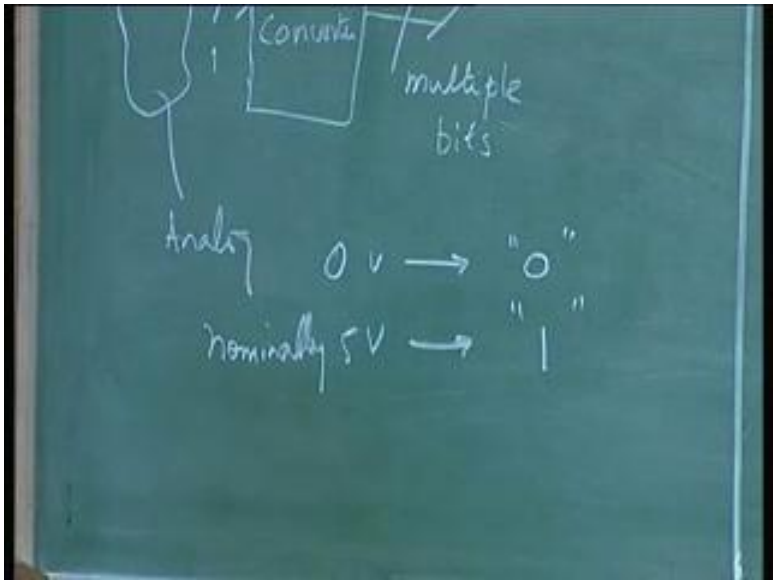
And normally I said normally unless otherwise stated I think told you the other day unless otherwise stated these signals we will assume as two levels being 0V and 5V but today this is a conventional value 0 and 5V have been there for a while, the electronic switches. When I say switches I mean as I said even a primitive thing non-electrical quantity like a flag waving can be used but in electronics the switches are simple devices for which we know that you have a switch at home to turn on and turn off a light but a switch is little more complicated, electronic switching, because we want to have a smaller value and all the time a smaller size.

Earlier they used a switch between 0 and 5V but today because of the advances in technology and because we wanted to save power because whenever we are operating with lower voltage we can consume less power because power is V square by R for a given resistance if you reduce the value of V power dissipation is reduced. So today the technology is power saving technology because you want to have everything handheld.

I told you the other day what are the various devices you can have in digital domain like a handheld calculator or a cell phone or a hearing aide or any of those things which can be operated by battery and in order for the battery to last for a long period of time I would want the energy consumption to be very little consumed by the circuit and that is one of the ways but there are several techniques used to reduce the power consumption one technique is to reduce the supply voltage.

So even though we assume 0 as 0V and 5 as 5V for theoretical purpose for all our understanding purposes in this course, today's technology is all different. Today we talk of 0V and 3.3V, 0V and 1.2V also now they are trying to get. Why 1.2V? With a single battery I should be able to operate a walkman or any electronic device which is a very complex device so they want to have a 1.2V and then they want to have less than 1V. Of course it is not that easy is not scales voltage scaling down is not going to directly affect the circuit it is going to have several other problems. So voltage scaling will have effects on circuit design all those things are not the scope of this lecture series. So let us assume this 0 and 5V, 0V as 0 level or nominally I will say nominally even though it is not known 5V as 1. I just want to make one point here, as I said 5V can be anything.

(Refer Slide Time 9:35)

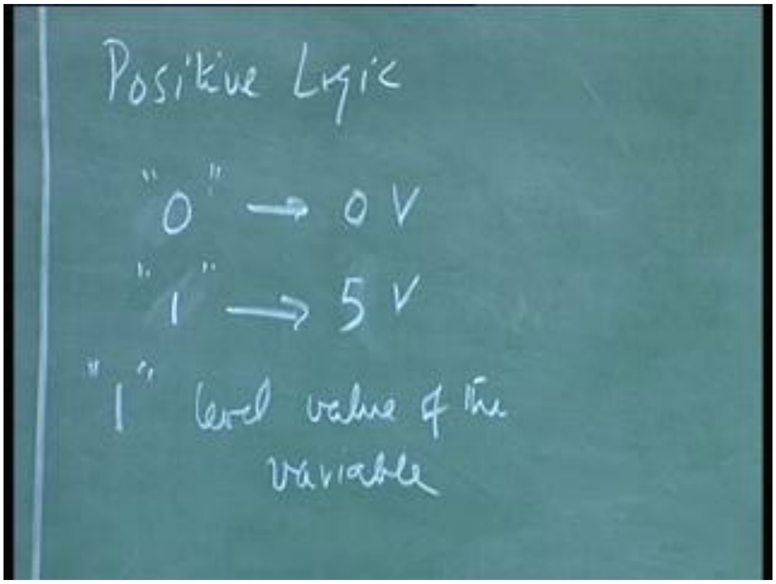


Sometimes it can be even more than 5V. Some technologies for some reason we want to have enough margin between this 0 and 1 condition. You are not sure about signal being coming or not coming, ambiguity. You want to have no ambiguity then you can have a larger range also. It is also possible to have 0V and 10V, 0V and 15V is also possible but rarely. Generally it is 5V or less and today as I said we are turning towards less than 5V operation.

But one thing I want to tell you is there is positive logic and negative logic. In a positive logic circuit 0 level voltage corresponds 0 corresponds to 0V and 1 corresponds to 5V or any other nominal value like 3V, 3.3V etc this is called positive logic. That means one level value the variable whether it is a voltage or current doesn’t matter voltage 5V 0V and current it can be 1 ampere and 0 amperes, it can be current also.

Binary variable signal need not be only voltage signal it can be current, power or whatever. As long as the value of the 0 level is greater than 0 level values such a signal is called a positive logic signal. That means presence of a signal is represented by a higher value, presence of a signal is represented by a higher value is called positive logic, obviously so positive, occasionally though we may have an absence of a signal represented by a higher voltage. Again this 5V is nominal it can be anything, it can be 3V, it can be 1.3V, 3.3V or whatever, even 10V for that matter.

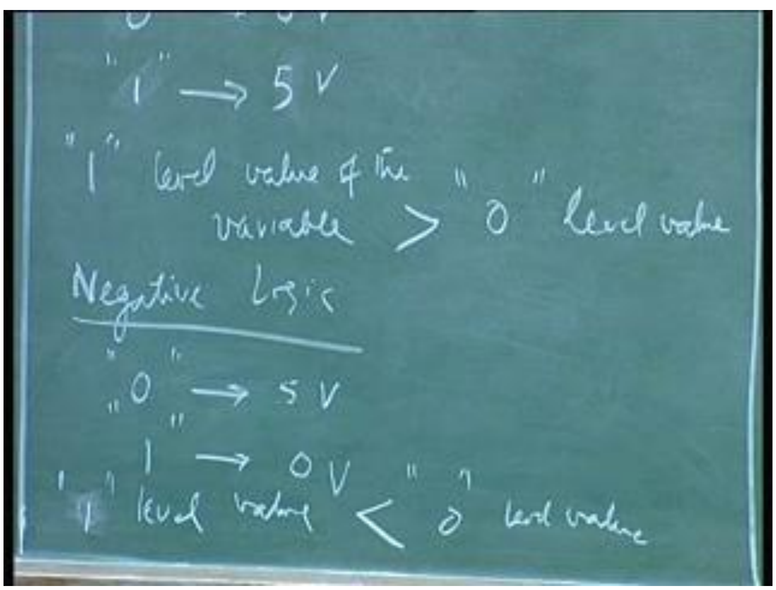
(Refer Slide Time 11:03)



The absence of the presence of a signal is represented by a value which is less than the absence of a signal, the value of this parameter the absence of the signal. The value of the parameter in the presence of a signal is less than the value of the parameter in the absence of signal such logic is called a negative logic. So a 1 level value is less than 0 level.

I will not write the detail you can write it down yourself it can be any parameter as I said it could be any parameter or it can be any value it can be voltage, current, whatever.

(Refer Slide Time 12:50)



And one other thing is it need not be always positive 0V and 5V. I can have for example – 5V and 0V. Here again it is greater 0 is – 5 one is 0 here on the other hand I can have 0 as 0V and 1 as – 5. Any time the presence or one value one level is more it is called positive logic and whenever one level is less it is called negative logic.

These are all concepts I am trying to tell you because these things in the introductory course I don't want to confuse with so many variations and parameters positive logic negative logic, I will not give you one problem with 5V as one level and another problem with 3.3V there is no point in doing it just for understanding concepts. We will stick with positive logic 0V as 0 level and 5V as one level. That will be the standard thing unless otherwise told. These are for reasons for a particular circuit to be explained properly or to give you a complexity in examination or whatever. But unless otherwise stated you don't have to keep on asking me are you talking about positive logic negative logic you don't have to keep on asking me do you mean 1 = 5V always.

In this course we have set the ground rules we discussed all about what is digital, how is the digital signal different from analog signal, how we can have only two levels and still have whatever representation you want in terms of the range of the value or the accuracy of the value. we also said the advantage of digital and discussed why we still need to have analog part of the system, we have set the ground rules, we talked about the positive logic and negative logic, we also said how to represent a number which we are familiar with normally like digital representation, decimal representation to binary representation and similarly a binary representation back to decimal representation. This is something you are supposed to know for submitting the assignment.

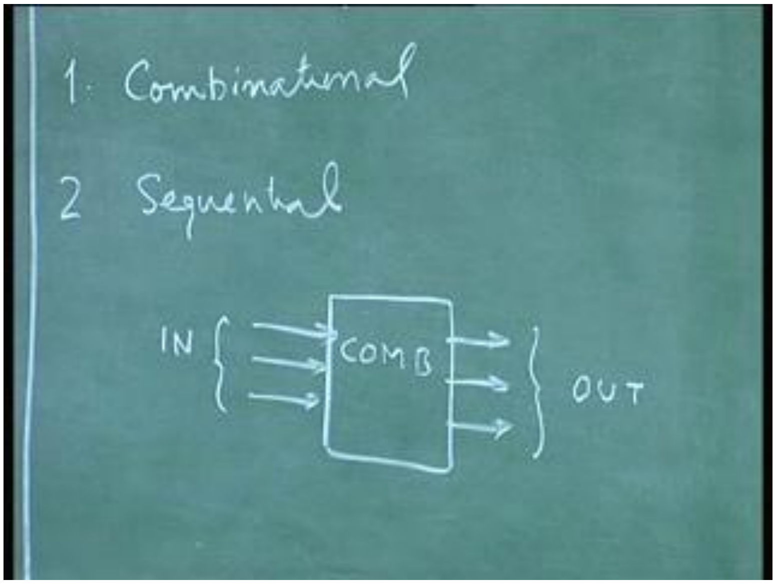
If you remember hierarchically we said system, subsystem, modules, functions, logic levels or whatever gate levels or circuit level and what we will not do is draw transistor level, circuit and analyze currents and voltages.

We will also mention currents and voltages as I said now in this case 0V and 5V. You know it should have an approximation, we are electrical engineers after all and we are going to finally deal with voltages and currents in the lab so you should know that. So finally we will talk about voltage. We already mentioned about 0 and 5V and what are the other possibilities and what are the currents that are flowing in these gates or these digital circuits. But for that we will stop with the functional level. We will define a functional block and analyze various functional blocks and use these functional blocks in the design of digital circuits that is our goal and to the extent possible we will build subsystems and possibly systems.

With this introduction let us get into the types of functional blocks they all call it logic even though they will deal with numbers and add and subtract whole thing is your logic because the basis of all these circuitry is what is known as a gate. These gates are called as logic gates because they operate on some logic concepts. Any circuit arithmetic circuit or even any other system that you can design including a very complex micro processor of today you can theoretically break it down to these gates which are logically logical functional blocks. So some people call them types of logic. Hence instead of functional blocks we keep on using it we can say logic. There are two types of logic functions; one is called the combinational logic and the other is called the sequential.

In any digital system especially practical system there will be combinational elements and sequential elements. Even though in principle you can define a purely combinational circuit, sell them, you have very limited use for combinational blocks and sequential block necessarily will have combinational logic. We will see why a little later on in this course but just to give you an overview definition, a first order definition, it is not a crude definition not a wrong definition but it is a very first order very crude definition.

(Refer Slide Time 19:07)



Combinational circuit or combinational logic or combinational functional block is a functional block with inputs and outputs of course all circuits have inputs and outputs.
Outputs of this block depend only on the inputs at that time. If we give a set of inputs a set of outputs are defined for this combinational block I have to implement it, we can analyze to get what it is or given this input output relationship I can build this combinational logic by designing it.

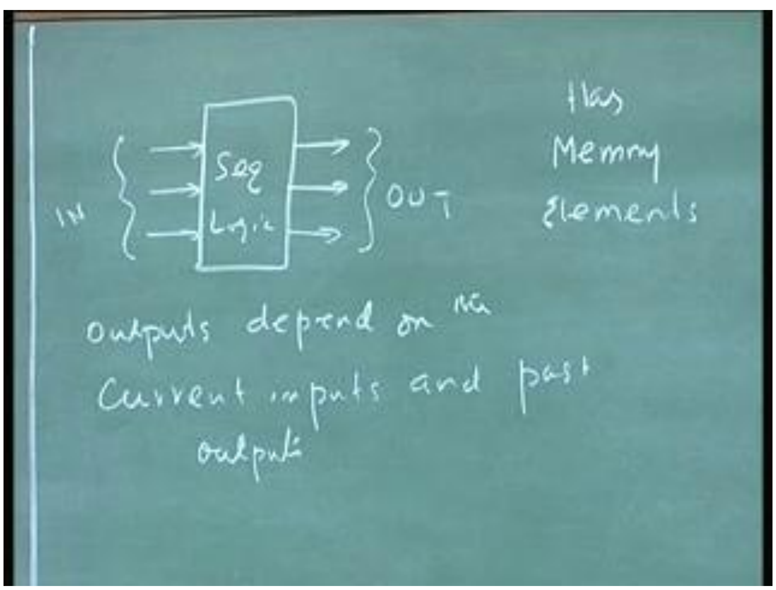
The concept is in combinational block the output depends only on inputs at any given time. Outputs depend only on the input at the current inputs, it doesn't store the history of the circuit. For example, suppose I have a set of switches and these are set of lights and a particular combination of switches will make one of these lights glow. If I change the switch configuration another light will glow. So by changing the inputs I can get the change in the output without having to worry about what it was earlier. So the output is purely based on what is the input. This is combinational logic (Refer Slide Time: 20:24) and on the other end is sequential logic and again we will have input output, the output not only depends on the current input but also on the past outputs or past history of the circuit.

So outputs depend on the current inputs and past outputs. Take an example of a counter; if I want to count one, count one more what is the new value depends on what is the old value, if I had 6 earlier now it will be 7, 13 earlier now it will be 14 so the counter has to be a sequential circuit because it has to have a sequence of things.

A micro processor or a computer has to be sequential. You give an input it has to take it and then consider something and then give the output. Some sort of sequence of operations takes place in order for you to get a proper result. So sequential circuit not only has to consider the present inputs but also consider the previous outputs or the past history of the circuit in other words before it decides what this output is. The current output depends on the current input as well as the previous history. Previous history is represented by the past output so it needs a memory which is not required in the combinational logic that is the only difference. this circuit need not remember anything like you guys you come here listen and go you don't you don't worry about this course right. Only for the exam you will start looking at what this man has been talking so many days let us see what is there in this course.

In a sequential circuit on the other hand the behavior depends on the memory what is the thing like movies, cricket statistics you have memory right you want to watch a match you will have to know the previous score of all the players, why did Sachin Tendulkar fail seven innings in a row, that history is required, so it is the past history, memory. This circuit (Refer Slide Time: 23:14) does not need a memory and this circuit has memory element.

(Refer Slide Time 23:24)



How would your memory element be built will be seen later when you come to sequence circuits. But I want tell you as a broad classification of this. And most digital circuit systems are sequential, of course there is no use.

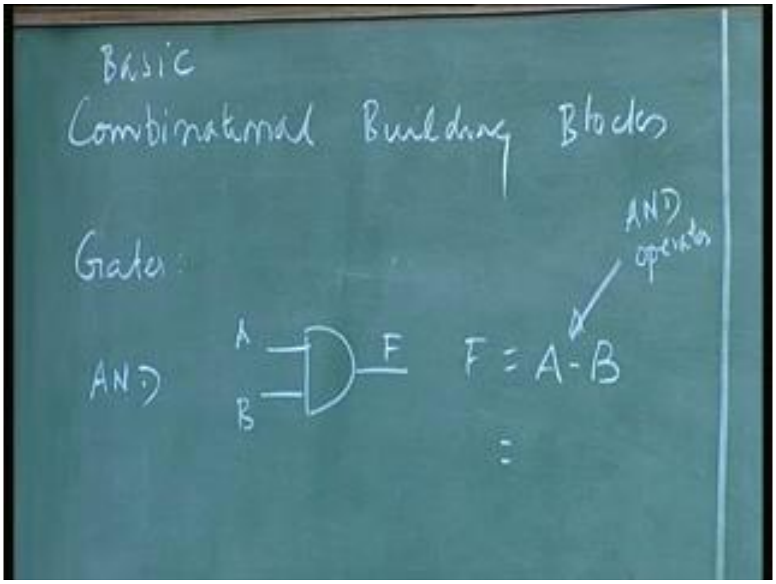
For example, you are going to have an elevator, lift you have to press the up switch, you are in third floor and you want to go to the fifth floor so you press it and if it only knows you are in three it will add two more and then go to 5.

Let us say there is only one floor at a time it can move you have a very simple elevator which can move only one floor at a time so if you switch up press up it has to go from three to four, traffic light also in a way has to be sequential because it should know there may be cars but then the signal will just change from green to red or red to green in one line just because a car is present you don't want to change it again you have to wait, have to spent enough time for this street to have the green signal, it has not been having green signal for a given amount of time there is no point in changing it now so wait, some sort of a behavior model is required which is not required in the combinational logic.

After having said that let us really get into the combinational building blocks. We will start with the combinational building block that is why I said. We will have to do quickly all these things because we have to do combinational sequential blocks and sequential blocks and then put them together in circuits and then make subsystems and then make systems a large agenda. We will see how much we can do.

Combinational building blocks or combinational I will not even say combinational building blocks but I will say basic combinational building blocks. That means in basic building blocks there are little more complex building blocks. We will start with basics.
The basic combinational building blocks are called gates all of you know this most probably from your high school physics. I don't know why they teach these things in high school physics. They would rather teach you good physics. Combinational building block is just by giving one chapter in electronics.

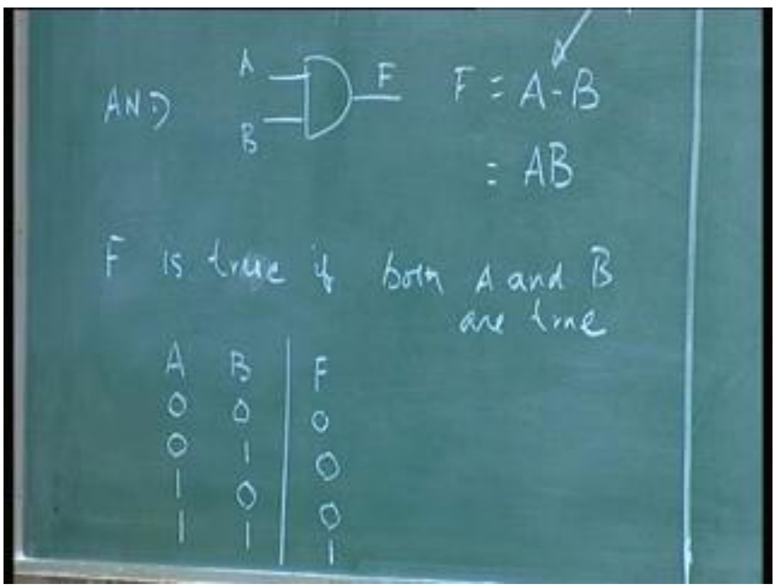
(Refer Slide Time 27:07)



Anyway since most of you know what it is gates are the combinational building blocks. Gate is a logical function that's all. That means it decides an output based on the input conditions and a definition of the function of the circuit. So what are the basic gates we are familiar with? AND gate of course all of us know that this is the symbol for the AND gate.

Let us say there are two inputs A and B. Let us say F is the output, F is A and B. This dot you see here (Refer Slide Time: 26:50) is not a product or a multiplication dot it is AND operation this is an operator called AND and the dot is called an operator AND operator. But as you go along keep on writing things you will drop this dot. Many of us write this will like this even in books (Refer Slide Time: 27:18) you won't find the dot in between. That means it is A and B. Even though we can read it is AB it should be strictly be read as A AND B. What does it mean? An output F is present only if both A and B are present, AND, that is why it is called as AND operation, both A AND B are present. Even if one of them is absent output is absent.

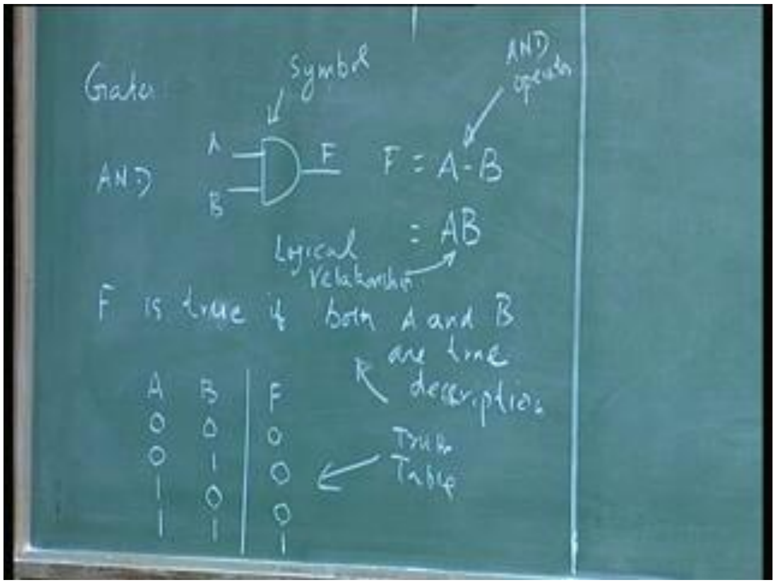
(Refer Slide Time 28:52)



So F is true, the presence or absence instead of saying all that because signal can be present or absent we will always say whether it is true or false, F is true, if both A and B are true they may be equivalent to 0V and 5V, 0V and 3.3V. We decided that we will only symbolically represent this level as 0 and 1. that means I can also say it this way, I can say it this way, I can say it this way (Refer Slide Time: 28:28) and I can it this way, I can it also in one more way I can write a simple table. if both A level and B level are 0 levels F is 0, if A is 0, B is 1 then also F is 0, A is 1, B is 0 then also F is 0 and if A and B are both 1 then F is one. So there are four ways of saying it. We can say F = A and B you know what it means, if I draw this symbol you know what it means, if I say F is true if both A and B are true this is an operational description, this is a symbol (Refer Slide Time: 29:17) and this is a logical equation or relationship, this is a description and this is a table. This table is also called a truth table.

Truth table gives you the tabular representation of a particular function. F is true or F is present only if both A and B is present or F is true if both A and B is true otherwise not.

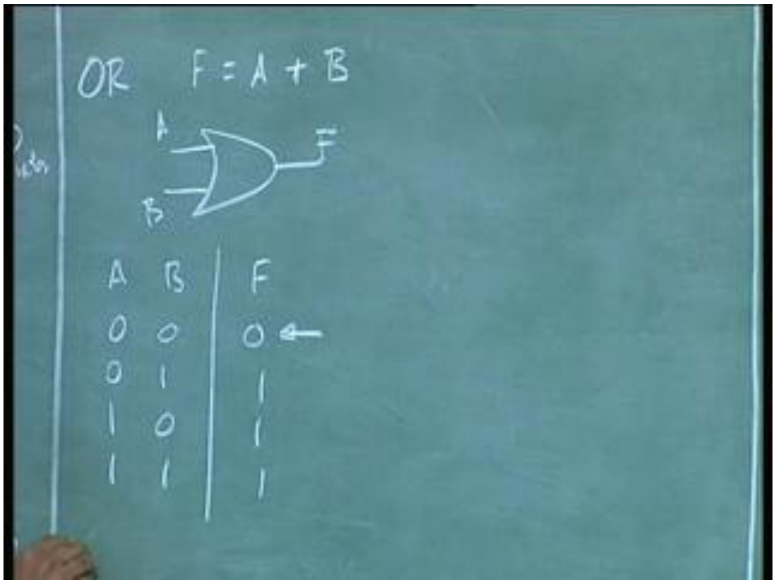
(Refer Slide Time 30:19)



So this is called a truth table of AND gate, this (Refer Slide Time: 30:16) is the description of AND gate, this is a logical representation of AND gate and this is the symbol of the AND gate. Likewise there are other gates. We will quickly finish some of them today.

OR is equal to A OR B this is OR operator, this is not plus don't read this as + B A OR B. That means if one of them A OR B is true output is true. Of course for both, if both A and B are true then also output is true. I will not give the word description you can write it yourself; F is true if either one of A OR B or A AND B or both true output is true. We can say at least one of A and B is true output is true whatever way and the symbolic representation as I said this is the thing the gate is represented like this (Refer Slide Time: 31:20) and then the truth table is this, and the only case for which the output is false is when both A and B are false.

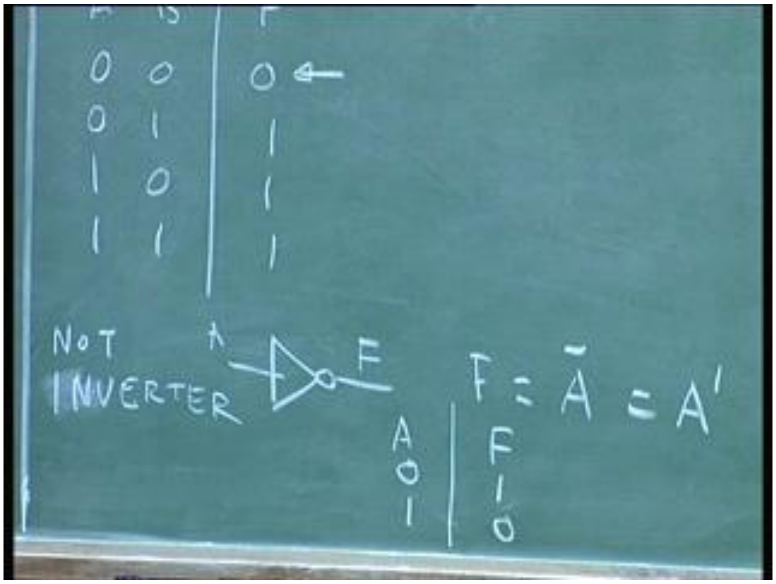
(Refer Slide Time 31:55)



If at least one of them is true output is true. There is a most widely used gate called a NOT gate which simply gives the opposite of complement they call it.

NOT A, suppose I have a gate whose input is A output is F this is the symbol, F is A bar, bar is negation or a complementing operation. This is called NOT we can say A NOT or NOT A, F is NOT A or some people call it A bar because there is a bar on the top. Some people write it as a prime, A prime. strictly speaking it should be read as NOT A, F = NOT A that is the correct of saying but as long as you know what we are talking about many people start calling it as F = A bar, F = A bar means A gets complemented or A gets inverted that is why it is called as inverter NOT gate is also called as an inverter that means if A has only one input A, one output F, if A is 0, F is 1, A is 1, F is 0, this is the truth table of an inverter.

(Refer Slide Time 33:28)



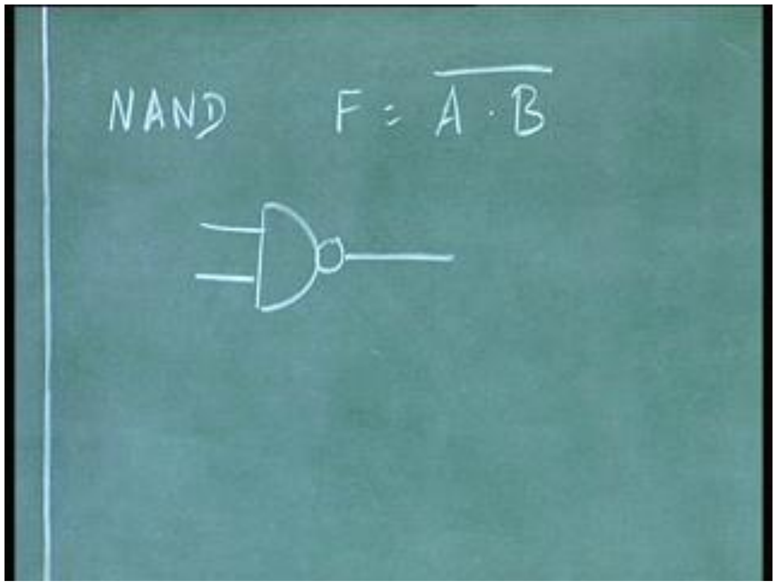
It can be shown we will see it in future classes that these three gates are enough complex it is any digital system however complex it is can be implemented using AND or OR Inverters there is a procedure for doing it we will see it. It may not be always optimum efficient in terms of the number of gates used, the area it occupies and the power it consumes, speed of operation because today we are talking about high speed operation, low power operation, low cost operation, small value. Take any electronic gadget it has to be a very small like a cell phone as I said or a hearing aide you put in your ear, size is important, energy is important, power consumption is important, cost of course is affordable, every consumer has everything as I told you the other day about how many computers we all use so cost, energy or power size of course and it should be has high speed as possible like internet.

You don't have even patience to click the mouse and wait for the response of the screen right? If you don't get it you will start double clicking, and double click and finally it says error message and it says you have to reboot it and start all over again. So we want everything very fast, it should be of very high speed that is we want very high speed, very low cost, very low power and very small size. Therefore AND OR inverter may be the best thing, it has the possibility of doing everything however complex things are, including sequential memory component we will come to that little later on.

Including sequential memory component can be designed using this but there may be efficiency you may have to think of some other way of doing it, that we will be see later. First we have to see how to make simple circuits using AND OR inverters and then we will see make them how we can make them more efficient by introducing more components.

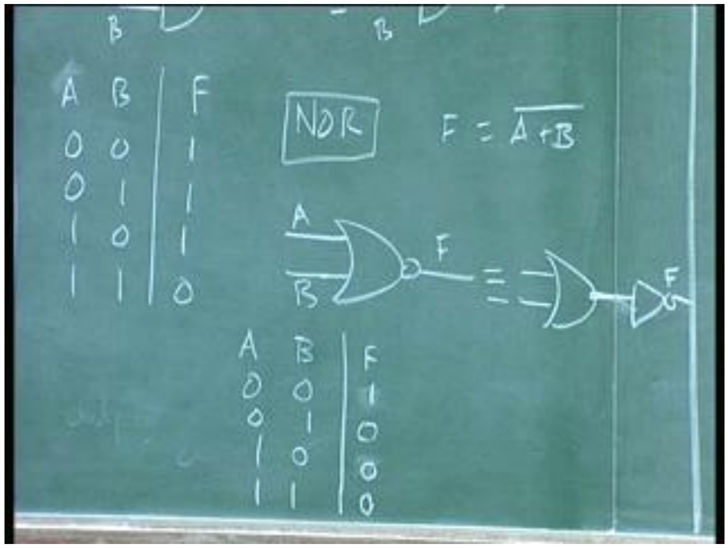
Now to complete the gates, even though these are the called the complete set of gates few other gates we have to mention. One is called a NAND gate as the word suggests NAND it is not AND invert of AND, invert of AND is NAND.

(Refer Slide Time 36:38)



So F is A AND B bar, not of A AND B you want to read it correctly, F is not A AND B that is exactly opposite to the truth table. Symbol is this bubble is identified as the inverter bubble (Refer Slide time: 36:28) this bubble you have in the inverter this bubble you will see whenever there is a negation function. Whenever you have a negation function in the digital logic you will find that bubble. So this is A and B i.e. AB, F so truth table is, here all you have to do is write the opposite to F AND gate, complement of AND truth table, this is 1 1 1 0 AND was exactly opposite of this complement, we don’t call it opposite of it, we call it complement. So, it can also be written as, I can have a NAND gate by putting AND gate and inverter together. So this is AND gate, OR gate, NOT or INVERTER, this is NAND gate then I am going to have a NOR gate which is to be expected as complement of NOT A OR B which is a logic function with a bubble. It can also be written as (Refer Slide Time: 38:13) and the truth table is complement of OR truth table and finally we have one more gate, these AND, OR, INVERT, NAND and NOR are the basic gates and I will also add one more to this called the Exclusive OR.

(Refer Slide Time 38:42)



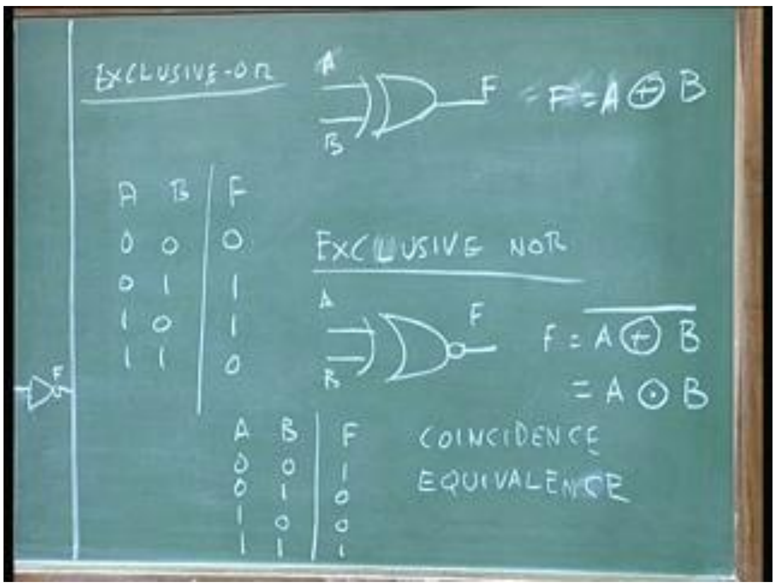
Even though these three gates AND, OR, Invert are enough, they shake the entire world. We can build any circuit, any complex digital logic be it sequential or combinational can be proved to be implemented using AND OR Invert but we will not use it, we will use this in the beginning and later on we will go to the higher level. But still NAND and OR why are we studying because of convenience. Whenever we have AND and Inverter why don't you combine and have a single gate instead of two gates. So we need this type of extra gates in order to simplify our circuitry we always have these four goals in mind; low cost, low power, small size, high speed. High speed can also be replaced by a high performance because speed is only one aspect of performance. So performance in terms of speed in the case of computers but high performance, low cost, low power, small size. Hence in order to do that we need all these extra, extra things.

Now Exclusive OR is another gate exclusive OR is same as OR gate except that in OR gate we said either A or B or both true the output is true. Output is true if A is true, B is true or both are true. If we remove that term both class then the output is true only if either A is true or B is true otherwise it is false not when both are true that's why it is Exclusive of, that means when you exclude the possibility of both being one that's called Exclusive OR gate, the symbol is this (Refer Slide Time: 40:49) same as OR symbol with a little arc in front and the same equation, there is no bubble here (Refer Side Time: 41: 17), A exclusive OR B there is no bubble.

Therefore my truth table is very simple, I use it sometimes, I have two signals if one of them is present I have an output. I am expecting only one of the signals, I am not expecting both the signals at the same time. This is a good signal detector, there are two signals and one of them is present, we have only a combination of either one of them is present or none of them are present, in that case this will be a good thing. When none of them are present or both of them are present also it is not required but the complement of exclusive OR has to be exclusive NOR.

Now we have the bubble. This is F the symbol is F A, this is the Exclusive OR symbol with plus in the circle. Since it is an Exclusive NOR complement the truth table is complement of this truth table. that means if both of the signals or present or absent or both signals are same if both A and B are same whether it is 0 or 1 if both are present or both are absent then the output is 1 otherwise output is 0. That means the truth table is, if both signals are present or both signals are absent. Or in other words if both A and B are same output is 1, if A and B are different output is 0.

(Refer Slide Time 44:41)



So it is also called as coincidence detector. If A and B are same it is called coincidence. A and B are incident or non incident coincident or equivalence A and B are same A = B either 1 or 0 so this gate is also called coincidence gate or an equivalence gate coincidence or equivalent or equivalence. And sometimes they also write this symbol as A circle remove this bar put a circle with a dot inside that is sometimes used as a symbol for Exclusive OR or coincidence gate as you said, you can also build this within Exclusive OR and an Inverter, it is all the same. So we have seen the basic gates out of this these three AND, OR, Invert can alone be used these three are alone enough to conquer the world, any digital signal can be built.

We are going to see in the next class NAND and NOR by themselves are also enough each one of them not together. What AND OR Invert can do I said AND OR Invert are enough to do everything and we have NAND, NOR etc to simplify logic that’s what I said but I can show you in the next lecture that NAND alone can design any logic circuit that you want, similarly NOR alone can design anything. So NAND and NOR are called universal gates for that reason. The whole thing is called basic gates and out of these basic gates AND OR Invert are fundamental gates because they are required to make any operation possible in digital systems. The same concept of AND OR Invert can be implemented using NAND gate alone or NOR gate alone so because of that reason NAND and NOR are called universal gates.

Tomorrow I will give you a simple example to tell you how this can be done and then proceed further. Using these gates how to implement logic functions of significance will be understood. What do you mean by having these gates, what do you want to do with these gates. We will start building circuits using these gates and then first analyze them and later on understand given the function how to synthesize or how to design the circuits using these gates. So first you have to understand the logic of these universal gates and then we will have to see how to build simple functions and analyze them, how does it work and then we will see and how a given function can be implemented using these gates.