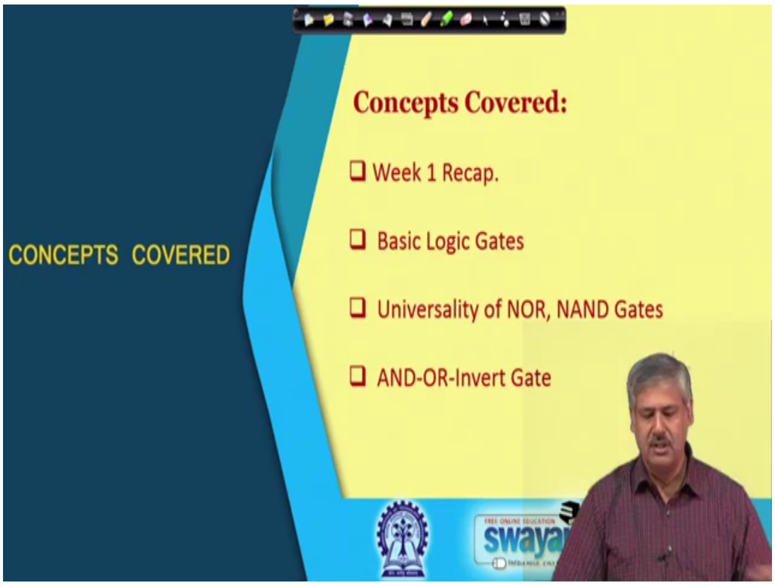
Digital Electronic Circuits
Prof. Goutam Saha
Department of E and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 06
Basic Gates and Their Representations

Hello everybody, we enter week 2 of this course.

(Refer Slide Time: 00:19)



CONCEPTS COVERED

Concepts Covered:

❑ Week 1 Recap.

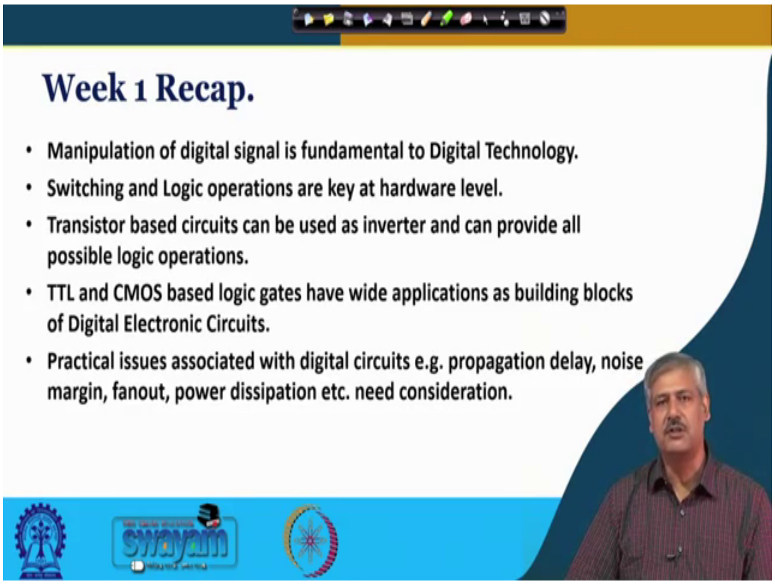
❑ Basic Logic Gates

❑ Universality of NOR, NAND Gates

❑ AND-OR-Invert Gate

In the 1st lecture of this week we shall have a quick recap of what we discussed in week 1, then we shall discuss the various representations of basic logic gates. We shall also discuss universality of NOR and NAND gates it is usefulness and also usefulness of AND OR invert gate.

(Refer Slide Time: 00:41)



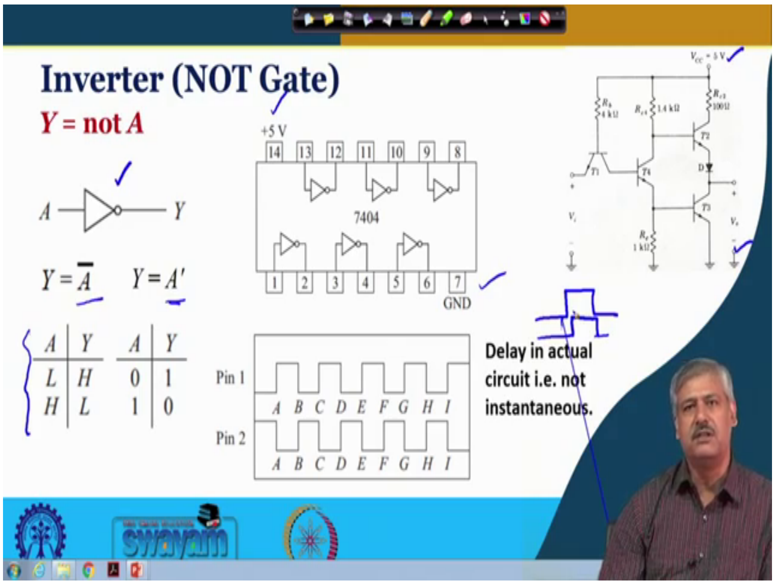
Week 1 Recap.

- Manipulation of digital signal is fundamental to Digital Technology.
- Switching and Logic operations are key at hardware level.
- Transistor based circuits can be used as inverter and can provide all possible logic operations.
- TTL and CMOS based logic gates have wide applications as building blocks of Digital Electronic Circuits.
- Practical issues associated with digital circuits e.g. propagation delay, noise margin, fanout, power dissipation etc. need consideration.

If we recollect in brief we had a look at in week 1 of following concepts. Manipulation of digital signal is fundamental to digital technology switching and logic operations are key to key at hardware level implementation at of digital. Technology transistor based circuits are important for realization of this switching and logic operations, because we get inverting up operation through transistors.

TTL and CMOS based logic gates have wide applications as building blocks of digital electronic circuits. And you also remember that there are practical issues like propagation delay, noise margin, fan out, power dissipation so on and so forth which need consideration when we develop digital circuits are complex digital building blocks.

(Refer Slide Time: 01:49)



Inverter (NOT Gate)

$Y = \text{not } A$

$Y = \overline{A}$ $Y = A'$

A	Y
L	H
H	L

A	Y
0	1
1	0

Delay in actual circuit i.e. not instantaneous.

To start with we recollect that we had discussed inverter through CMOS technology, through TTL technology, and you have given a circuit of the inverter using TTL at the right hand side which you can connect to. But, when we use these as a building block we shall not present either CMOS circuit or TTL circuit everywhere, because that will make it very very complex I mean the whole circuit will look very complex.

Instead of a inverter depending on whether it is a CMOS technology or inverter or a TTL technology or any other technology we shall put one symbol as a building block we will know that inside depending on the logic family that is used one circuit or the other circuit will be there ok. And that is represented through a symbol the symbol over here you see is this one.

So, this is the symbol that we shall be using for NOT gate we are all we may be already familiar with this, but to formalize this discussion we put it over here. So, this is one way of representing inverting operation, the other way of representing inverting operation is through Boolean expression. So, this is Y is equal to A compliment which represents inverting operation and it is represented through A bar or A prime both are equivalent.

So, you shall used interchangeably this two notations this can be you represented through the truth table also this is another representation. So, if you see a truth table which is like this that is a low connected is associated with high a high associated with low 0 associated with 1 I mean at the input is associated with 1 at the output and 1 is associated

1 at the input is associated with 0 at the output we know that we are talking about inverting operation. That is the inverter circuit is a digital circuit there is a inverting operation that is involved.

The 4th representation that we see is through timing diagram and for timing diagram we just look at one before looking at the timing diagram we look at tone IC which your people use in the lab; lab we use TTL family because of issues with CMOS circuits the lot of protections are required. So, mostly we will be using TTL circuits in the lab TTL ICs. So, IC 7 4 0 4 has got 6 NOT gates and these 6 NOT gates are connected the input and output pins are connected like this.

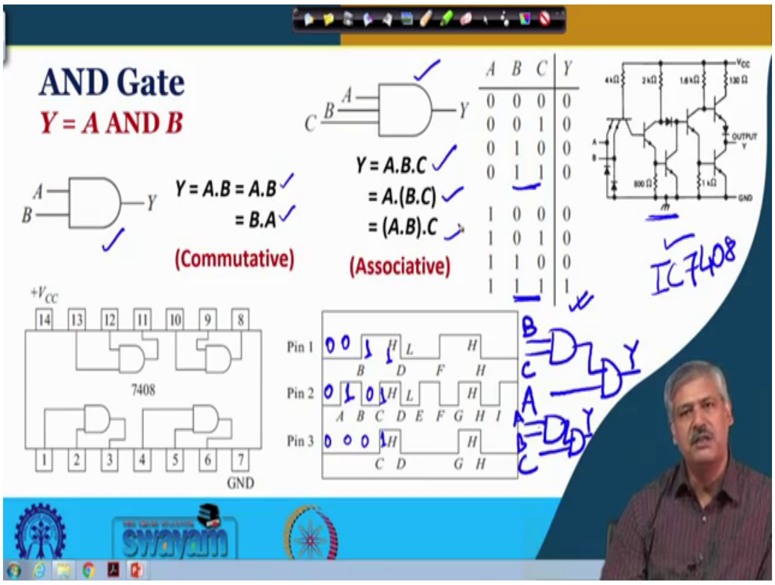
So, and there is a VCC and there is a ground which is required you know as power supply over here you see in this power supply this power supply. So, this power supply is are associated over here and there are 6 such NOT gates inside it ok. And we just if you give at the input of pin 1 a signal like which is low and high you know 0 volt 5 volt it is just you know there are rectangular signal like this then what will happen at the output.

Whenever, there is a low there will be corresponding high at the output. So, this is pin 1 and pin 2 that you see and that way it will continue right. So, depending on how long these input signal is changed you will see you will see a corresponding timing diagram at the output. Now, all these things that we have you know so, discuss for these representations in each of these it appears it appears that the output is instantaneous.

Like; if you look at the Boolean expression or the truth table you have given input immediately you are you know getting output that is the kind of you know understanding one may have from this kind of representation. But, if you have timing diagram and if you explain the time scale then you can have the propagation delay included in the timing diagram ok. So, depending on the time scale alright; if it is in the order of second or millisecond you might not be able to visualize it, if it is in the order of microsecond or nanosecond then you can see a nanosecond order of propagation delay inward.

So, that way the propagation delay based in presentations are useful, and this propagation that we shall discuss and use later in some other discussion.

(Refer Slide Time: 06:29)



AND Gate

Y = A AND B

$Y = A.B = A.B$

$= B.A$

(Commutative)

$Y = A.B.C$

$= A.(B.C)$

$= (A.B).C$

(Associative)

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Now, we look at representation of AND gate ok. So, AND gate again if you look at the circuit diagram will be having a circuit in the TTL TTL space which is which looks something like this ok. And remember when we talk about design you know circuit of AND gate at circuit level it is not a NAND gate then again a NOT gate is pushed put.

So, in a NAND gate circuit itself a inverting transistor a block can we put with some associate additional circuitry by which you can get NAND and NOT together and operation. So, that is the takes as instrument you know these data sheet from that the circuit has been shown this is a practical circuit which is in used for IC 7408 IC 7408 that uses this internal circuitry.

Coming to the again when we talk about and gate and is it is use then we will not be using the whole circuitry in various represents rather it is block. So, a 2 input and gate the block diagram the symbol representation is like this and 3 input it is like this. So, for 2 input how this input and output in the Boolean representation would look like it would look like this Y is equal to A and B and what about it is A and B; you can write in another manner B and A with I mean whether the operator if you just change the position you it does not make a difference.

So, this is nothing, but what is known as commutative I mean the operation is commutative. That means, the operands if the change their order in the operation there is no change in the output. So, this AND operation is commutative ok. So, this is another presentation. So, IC 7408 this is the structure in which I mean this is the arrangement in which 4 AND gates are there VCC ground and these 1 2 is the input and 3 is the output.

And if you send a signal rectangular signal like this in which the arrangement is such that there are these possibility 0 0 then this is 0, and this is 1, this is 1 and this is 0 and this is 1 and this is 1 ok. Then what we expect for and logic when both the inputs are 1 the output will be high. So, that is what you see for each of these case this is 0 0 0 this is 1.

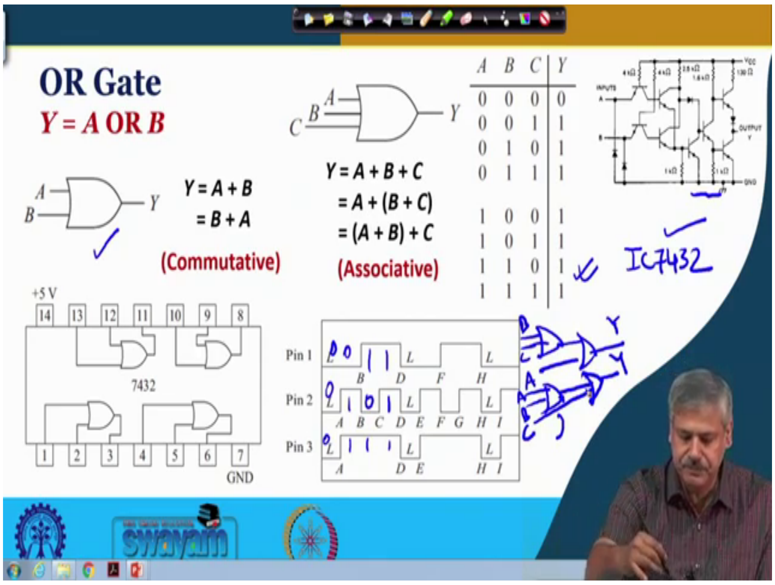
So, if you look I have a look at some such timing diagram you can understand the circuit involved in between this 2 input and the corresponding output is nothing, but an AND circuit or AND logic is involved. So, it can be represented through truth table also way to you know the way we had done there will be you know 2 inputs. And there will be 1 output and when both the inputs are 1 corresponding output in the truth table will be 1 and for rest of the cases it will be 0 ok.

Now, what happens in this case when the in the case of 3 input and gate we have you know similar such truth table the kind of truth table that you can expect is like this. So, 3 input so, there are 8 different possibilities 2 to the power 3 8 different possibilities of the input arrangement. And for each of these cases except when all the inputs are 1 output is 0 alright.

So, that is what you see over here. So, that is the operation and Y is equal to ABC all right and one important thing over here if you 1st and BC all right which will generate a output 1 when both B and C are 1. And then this output you and you with A right then the output will be 1 when A is also 1 so, this is the case. So, BC is 1 just if you think of this is B and this is C this output is fed to another AND gate. So, BC is 1 here as well as here alright these two cases, but when A is 0 all right output is 0 because it is again getting AND gate and when A is 1 output is 1.

So, if you group BC if you and them together all right and then and with A the output remain same as that of Y is equal to ABC. And similarly if you group A and B together and then you and it with C; that means, A and B then this is AND gate with C right the result remains the same the result remains the same. So, whether this grouping or association is done this way or the other way result remains the same so, AND operation is associated.

(Refer Slide Time: 12:13)



OR Gate

$Y = A \text{ OR } B$

$Y = A + B$
$= B + A$

(Commutative)

$Y = A + B + C$
$= A + (B + C)$
$= (A + B) + C$

(Associative)

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Now, we look at or operation. So, OR operation again the circuit if you look at the circuit level the circuit will be this is IC 7432 that inside circuitry that is there again you will see that it is not OR NOR gate which we discussed earlier. And NOT gate which is put in succession or in cascade rather within the not structure itself one inverting transistor is produced with associated circuitry with which you get OR operation ok.

That reduces the transistor count ok. So, again the whole circuitry we shall not be using for our subsequent discussions or developing more complex digital circuits we shall be using symbols. So, this 2 input or gate symbol is like this where A and B the inputs that you see over here if you just interchange the order or the operation the way you had done for AND operation there is no change in the output ok.

So, that is why this OR operation is commutative right, and if you look at other representations like timing diagram. So, for that if we use the IC 7432 IC 7432 is again a TTL IC which has got 4 OR gates inside it and this 1 2 is the input and 3 is the output and similarly if we look at 4 possible combinations at the input the way the signal is you know presented to the input pins.

So, you have got this 0 0 over here 0 1 over here 1 0 over here and 1 1 over here. So, for OR logic this is 0; and for you will see at the output of the pin if you just look at the wave form all right or the signal you will see that the output is 1 when any of the input is

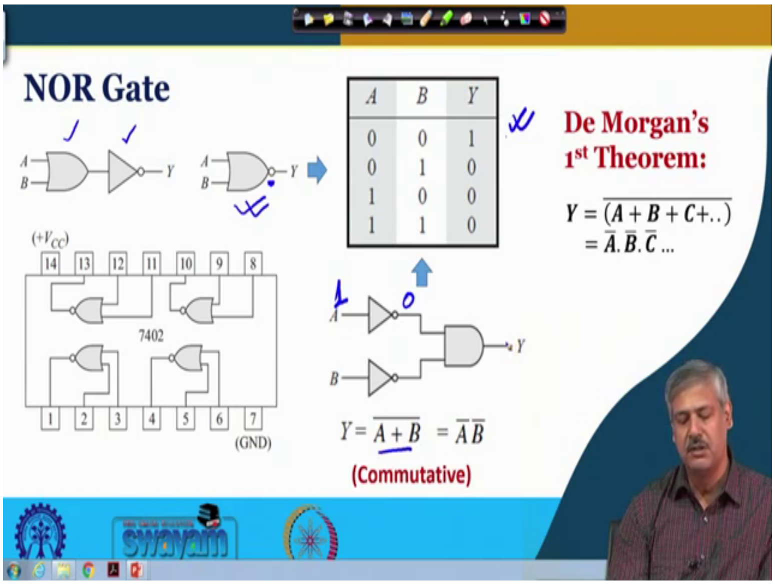
1 that is what OR logic is. Again I have noted that that before that if the time scale is expanded which by which you can visualize nanosecond order you know time indices.

Then you can see the propagation delay involved in this timing diagram based representation and of course, there is a truth table base representation the where for B and C for A and pin 1 and 2 if they are any of them is 1 in the corresponding output in the truth table will be 1 and when both of them are 0 the output will be 0.

Now, similar to AND gate we can have multiple input OR gate also multiple input OR gate for which we is just show a an example of 3 input or gate. So, this is the 3 input OR gate and the corresponding logic diagram is any of the input 1 the output will be 1 any of the input will be 1 output will be 1 that is what you see in this truth table right. And similar to AND OR is also associated; that means, whether you group B and C first you do B and C oring first and then you or the output of this B and C with A.

Whatever result you get if you OR B A and B. And then it is output you OR with C the result will be the same I mean the what I had seen that is what we discussed in case of AND gate ok. So, similar thing is also applicable over here. So, whatever is this same thing instead of if you have got A and B and then you are then you are oring with C the results remain the same so, this is associated.

(Refer Slide Time: 16:23)



NOR Gate

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

$Y = \overline{A+B} = \bar{A}\bar{B}$

(Commutative)

De Morgan's 1st Theorem:

$$Y = \overline{(A+B+C+..)}$$
$$= \bar{A}.\bar{B}.\bar{C}\ ...$$

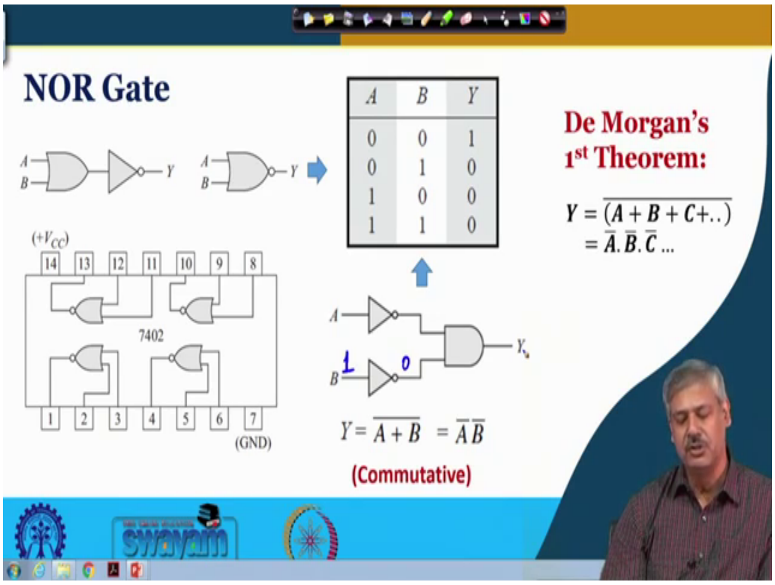
Now, we come to NOR gate. So, corresponding IC is 7402 in the TTL family NOR gate circuit we have already seen before in earlier previous way and for that we are using the symbol the one that you see over here this symbol ok. This symbol we are not using a OR and then a NOR gate rather we have putting this bubble over here. So, this bubble this bubble whenever in relation to logic operation is present you have to you can consider that there is an inversion associated with that bubble ok.

So, put together it looks more compact. So, this is your NOR gate and the NOR gate truth table we know it is just opposite of OR right opposite any of the input was 1 for or output was 1 here any of the input 1 output will be then 0 all right. And when both the input are 0 the output will be 1 ok. So, this is what you see for the OR gate. So, I am not drawn timing diagram representation of this. So, the similar thing can be drawn for the way we have done for possible combinations and corresponding output can be drawn where both of them 0 the output will be 1 ok

And note: in 7402 it is this NOR gate IC the 2 3 is input and 1 is output unlike the previous cases 1 2 was input and 3 was the output for AND gate and OR gate that we have seen before all right. So, this is something for practical purposes you need to take note of. And Boolean expression we just use this is OR and then a compliment or a bar that represents the NOR operation. So, this is the NOR operation in the Boolean expression of it.

Now, one thing is interesting over here is that whatever NOR operation you have seen that kind of truth table that you have got. If you have the same inputs complimented and then AND it alright, we get a equivalent representation. So, let us consider this A and B this is AND operation all right. So, any of the input is 1. So, this is 1 then this is 0. So, that will make the output 0 A is 1; that means, output is 0 right because for the AND gate 0 is the forcing input. So, 0 means output will be 0.

(Refer Slide Time: 19:01)



NOR Gate

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

De Morgan's 1st Theorem:

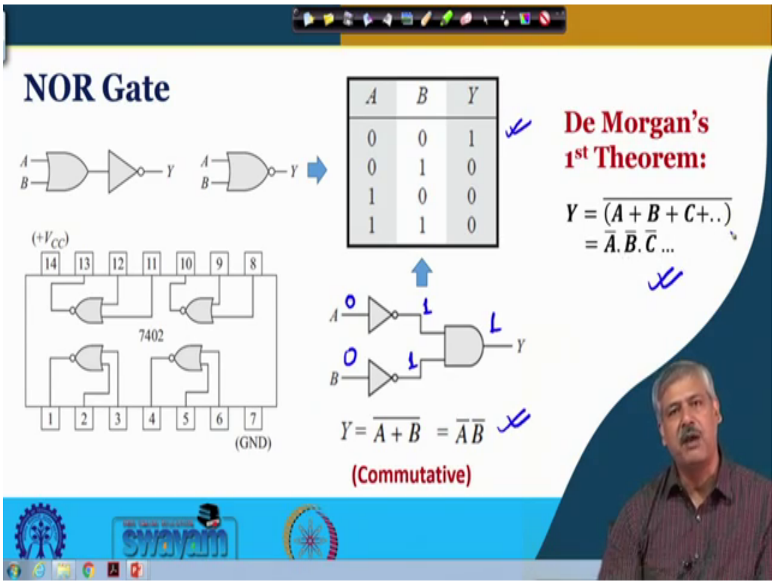
$$Y = \overline{(A+B+C+..)}$$
$$= \bar{A}.\bar{B}.\bar{C}\,...$$

$Y = \overline{A+B} \;= \bar{A}\bar{B}$

(Commutative)

So, instead of that if B is 1 this output will be 0 which will force the Y to be 0.

(Refer Slide Time: 19:07)



NOR Gate

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

De Morgan's 1st Theorem:

$$Y = \overline{(A+B+C+..)}$$
$$= \bar{A}.\bar{B}.\bar{C}\,...$$

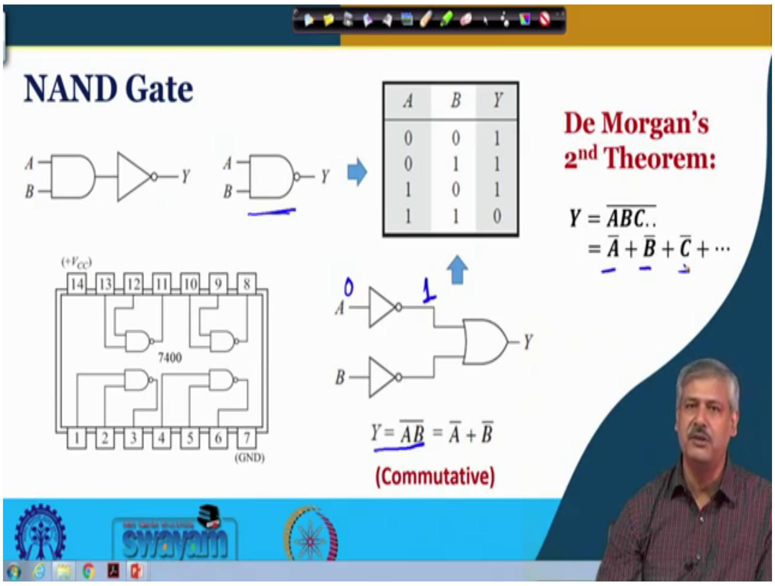
$Y = \overline{A+B} \;= \bar{A}\bar{B}$

(Commutative)

Only when both of them are 0 so, this is 1 and this is 1 these output is 1. So, this is this case this is nothing, but NOR operation right. So, we can write this A plus B prime as A prime and B prime as well and we can see that this is also commutative the way we have seen and extending this one for more number of variables ok.

So, instead of A B if we have got say C D and all then you can extend the same thing the way you have done for 2 variable cases and we arrive at De Morgan’s theorem was theorem which is saying which is represented in this manner this is clear.

(Refer Slide Time: 19:59)



NAND Gate

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

De Morgan’s 2nd Theorem:

$$Y = \overline{ABC..}$$
$$= \bar{A} + \bar{B} + \bar{C} + \cdots$$

$$Y = \overline{AB} = \bar{A} + \bar{B}$$

(Commutative)

Now, we look at NAND operation ok. So, NAND operation a similar again the way we had discussed NOR we are not having a 2 different you know symbols that is AND and NOR a not put together rather we are putting 1 single logic gate with bubble means and then there is a inversion associated ok.

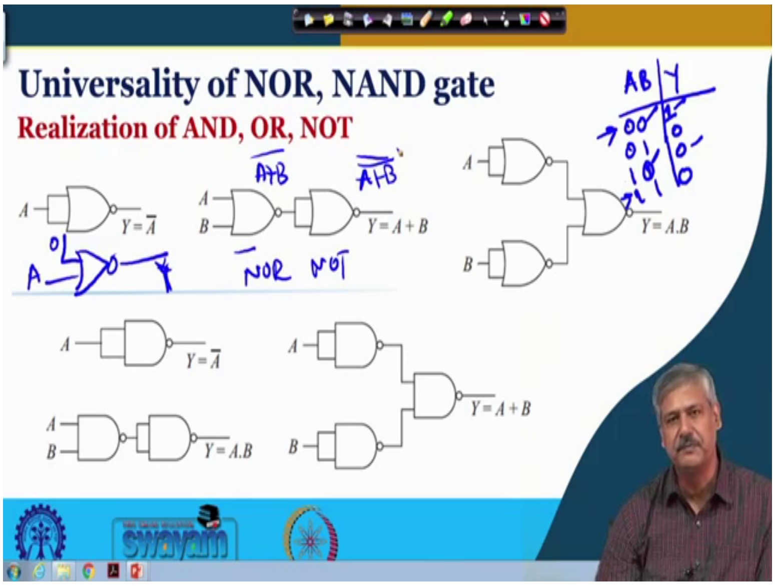
And the corresponding truth table is output is 1 when any of the input is 0. Any of the input is 0 for and output is 0. So, in this case any of the input is 0 output is 1 only when both the inputs are 1 the output will be 0 and the corresponding the IC that we use in there TTL family is 7400. So, 1 2 is the input and 3 is the output. So, 4 such this gates are there inside the IC and you can have a corresponding timing diagram ok.

And for NAND gate what we had seen in case of NOR gate a similar thing occurs over here. While the NAND representation this A B this is the Boolean representation complimented that is what is the NAND operation over here. The same thing we can obtain where the inputs are complimented and then OR.

So, A bar plus B bar how is it done. So, we just you consider any of the input as 0 this is symmetric so, it will do. So, then this output is 1 for OR gate 1 is the forcing input. So, if the OR gate input is 1 the output will be 1 right so that is what is happening over here and if you extended to more number of variable like ABC over here right. So, then the corresponding expanded version in this form will be A bar plus B bar plus C bar all right and so on and so forth.

So, that is giving you De Morgan’s second theorem which is also very very useful which is which we shall discuss more later on ok.

(Refer Slide Time: 22:03)



Universality of NOR, NAND gate

Realization of AND, OR, NOT

Now, universality of NOR and NAND gate this is something interesting this is useful in; since if you have only one, if you think of only one variety of you know gate by which you would like to get many different logic operations or Boolean an expression realization of that then you have to pick up I either of NOR or NAND, because they can be converted to any other logic operation by different combinations of themselves.

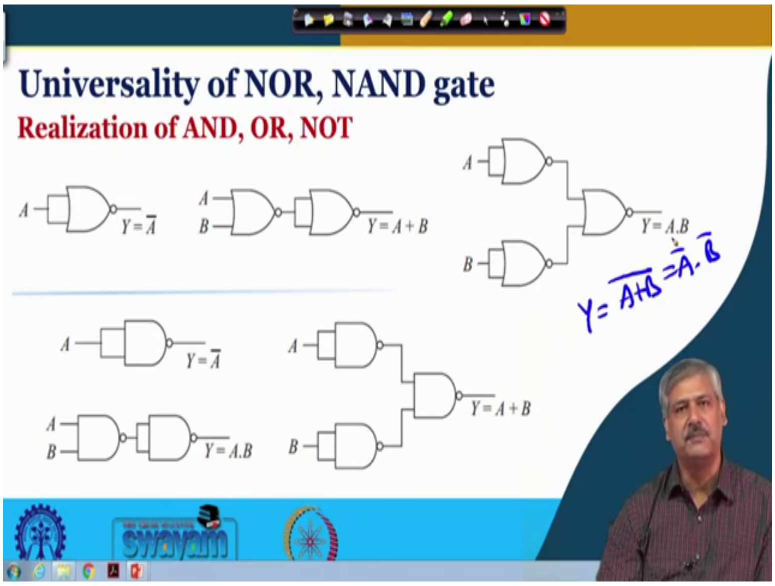
That is the good thing about it. So, you can have only one variety of it one kind of fabrication process which is useful otherwise for AND gate and or gate to different variety there are different kind of circuitry that we have seen before. So, NAND you just having only one kind of circuitry and how we can achieve it so, we just give some examples of basic you know Boolean operations.

So, far from NOR gate if you want to get a if you want to get a not operation all right. So, you just need to short both the inputs and then the corresponding output will be the invert of this I mean if you remember the NOR logic. So, this is A B and Y; so when 0 0 0 1 1 1 and sorry 1 0 and 1 1 ok.

So, NOR logic any of the input is 1 output is 0 right and when both the input are 1 output is 1 right. So, if you just short both of them A and B. So, common thing common one then what is happening at that time. So, either 0 0 case is there or 1 1 case is there when 0 0 is there output is 1 and 1 1 output is 0 just opposite of it right. The other thing for NOR gate we know 0 is the non forcing input all right. So, if you have a NOR gate right. So, which is you connect to 0 and this you connect to A how the output will be real you know behaving alright. So, B is 0 so, this is the case and this is the case these are the two cases right and at that time if A is 0 output is 1 if A is 1 output is 0. So, that is also giving you a inversion is not it alright.

So, that is the two ways you can get NOR operation NOT operation from A NOR gate to input NOR gate and how you can get or operation. So, NOR and followed by a NOR followed by not you get or operation all right just inversion of so this was A plus B bar. So, this is A plus B bar and again another B bar. So, which we will get A plus B clear and how you AND operation?

(Refer Slide Time: 25:29)

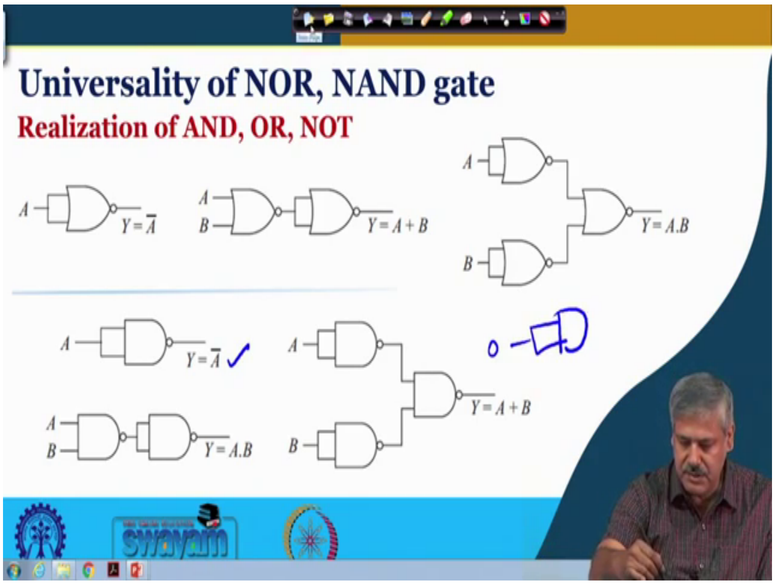


Universality of NOR, NAND gate

Realization of AND, OR, NOT

Earlier we have seen Y is equal to A plus B bar is A bar B bar right. So, at the input if you send complemented A and complimented B then you get A and B. So, that that is how we are getting AND operation over here clear.

(Refer Slide Time: 25:59)

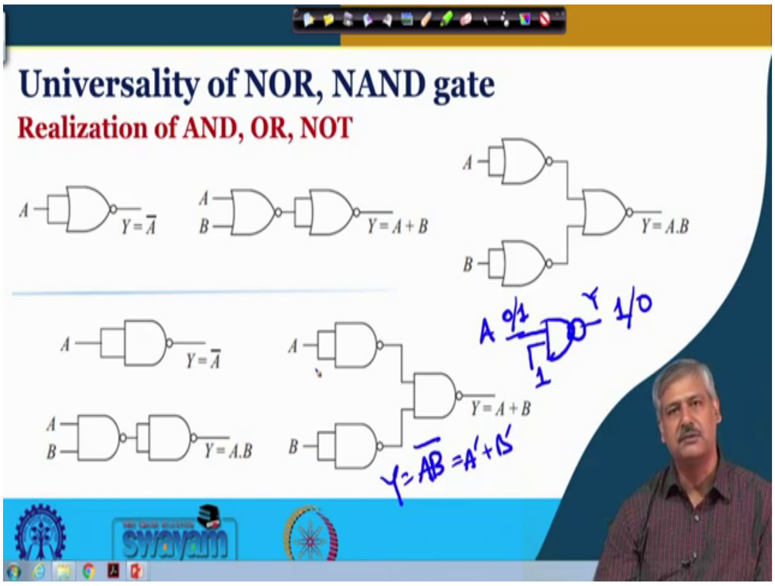


Universality of NOR, NAND gate

Realization of AND, OR, NOT

So, similar thing would happen in case of NAND gate also you can get any other operation like and like inversion. For inversion in this case right what is the other so, this is one option.

(Refer Slide Time: 26:11)



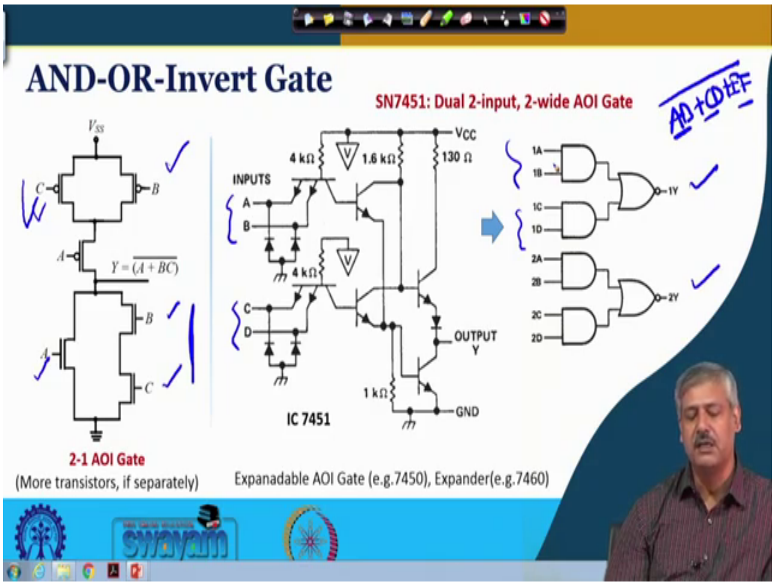
Universality of NOR, NAND gate

Realization of AND, OR, NOT

That is this is one option we have already noted over here right, the other option is non forcing input for a NAND gate is 1 right. So, if you give this as A and this is the corresponding output will be.

So, this is if this is 0 0 1 output will be 1 right. So, if it is 1 1 1 output will be 0 right. So, you get the inversion for some such arrangement as well all right and AND operation is NAND followed by NOT and OR operation again we follow the De Morgan's theorem alright Y is equal to A B prime is equal to A prime plus B prime. So, if you send primed version of it complimented version of it you get OR operation clear.

(Refer Slide Time: 27:07)



AND-OR-Invert Gate

SN7451: Dual 2-input, 2-wide AOI Gate

2-1 AOI Gate
(More transistors, if separately)

Expanadable AOI Gate (e.g.7450), Expander(e.g.7460)

So, the last topic that we discuss in this lecture is the usefulness of AND OR invert gate ok. So, when we realize Boolean expression we shall discuss it in future classes. So, there could be many different cases where you are having relationship like A B plus CD plus EF right inverted a kind of thing. So, basically there is a AND operation involved this is AND AND AND then this is OR and then finally and inversion is taking place ok.

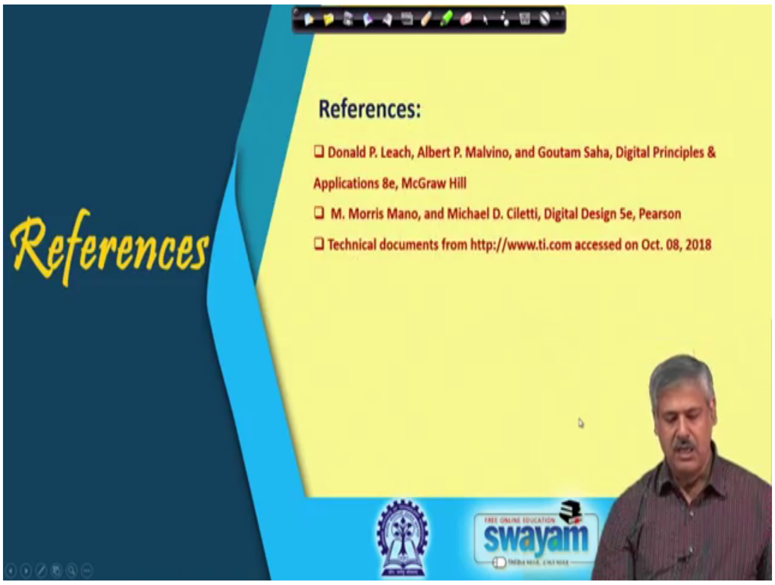
So, this is called AND OR invert. So, if you want to realize this separately right the number of transistors you know involved will be you know as it is required for each of these realization. But, within the logic family there are you know gates or the arrangements can be made by which this AND OR invert operation can be done with lesser number of transistor.

So, for CMOS it is very easy you can see that this is an example of two one AND OR in invert gate. So, these are the 2 inputs these are the 2 inputs over here and so, that is this is the 2 and this is the corresponding 1 input. So, these 2 are ended B and C are ended and this is OR with a and corresponding these are the NMOS side and these are the corresponding PMOS side where B and C are in parallel. These are in series here then NMOS side it is serious PMOS side it is in parallel and a will be in this side is in parallel, because of the or operation and in this side this is in series that we already know how we get CMOS circuits.

So, very very convenient way of getting it and we can get OR and invert gate also in similar manner for in a CMOS circuitry ok. TTL family also has similar such gates and or input gates. So, one example is 7451 which is dual; that means, two such gates is there 1 and another 2 input; that means these are the 2 inputs that you see 2 inputs 2 inputs. 2 wide means 2 such blocks are there right by which you can get and or invert gates and this is the corresponding circuitry secretary in the TTL family from the TTL data sheet.

And you can have you know expanders also arrangement by which more such gates can be attend added and more complex circuits can be arrived at this saves in a transistor count ok. So, these are some basic gates which are useful in realization impractical circuits or in your laboratory experiments which you will be undertaking.

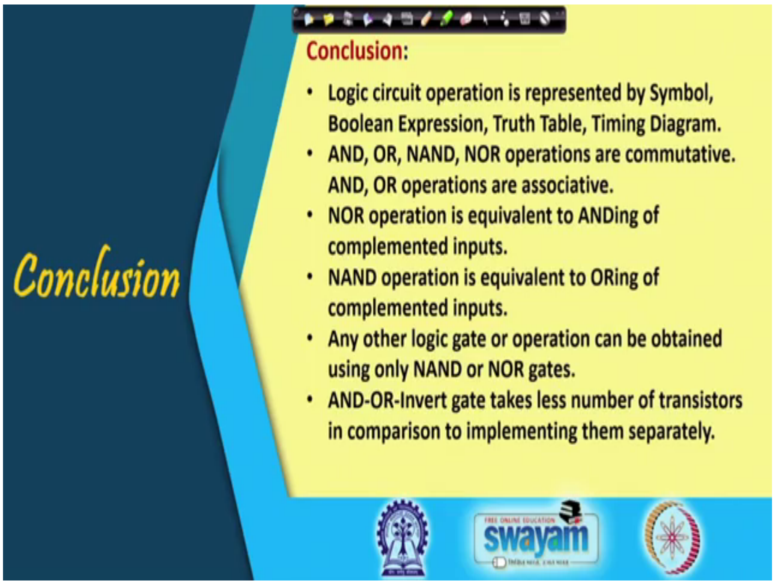
(Refer Slide Time: 30:07)



References:

- ❑ Donald P. Leach, Albert P. Malvino, and Goutam Saha, Digital Principles & Applications 8e, McGraw Hill
- ❑ M. Morris Mano, and Michael D. Ciletti, Digital Design 5e, Pearson
- ❑ Technical documents from http://www.ti.com accessed on Oct. 08, 2018

(Refer Slide Time: 30:09)



Conclusion

Conclusion:

- Logic circuit operation is represented by Symbol, Boolean Expression, Truth Table, Timing Diagram.
- AND, OR, NAND, NOR operations are commutative. AND, OR operations are associative.
- NOR operation is equivalent to ANDing of complemented inputs.
- NAND operation is equivalent to ORing of complemented inputs.
- Any other logic gate or operation can be obtained using only NAND or NOR gates.
- AND-OR-Invert gate takes less number of transistors in comparison to implementing them separately.

And so to conclude, the logic circuit operation can be represented by symbol Boolean expression truth table timing diagram we have seen that. AND OR NAND NOR operations are commutative, but AND OR operations are associative the corresponding NAND NOR OR NOT associative which you shall discuss more later on.

NOR operation is equivalent to ending of complimented inputs De Morgan's theorem we have seen one version, NAND operation is equivalent to oring of complemented inputs another De Morgan's theorem. Any other logic gate or operation can be obtained using only NAND or NOR gates which is universality of this two gates. And AND OR invert gate takes less number of transistors in comparison to implementing them separately which is useful in many cases ok.

Thank you.