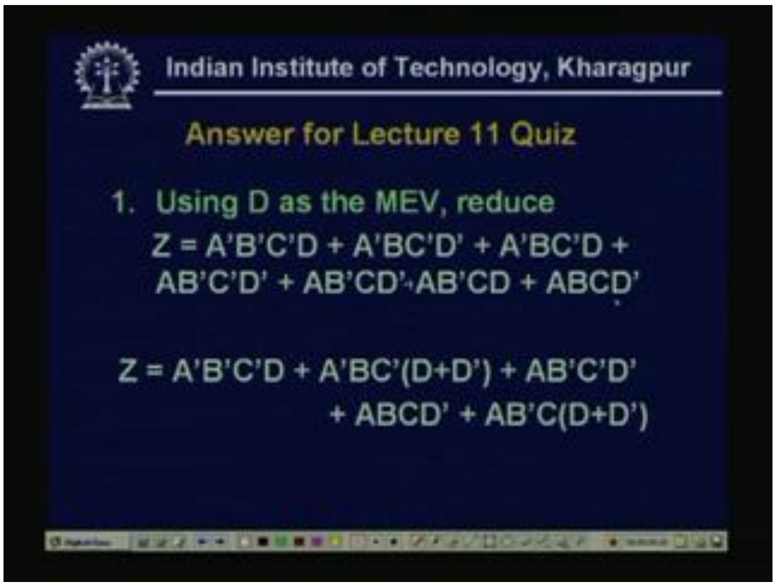
Digital Systems Design
Prof. D. Roychoudhury
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 12
Combinational Circuit Design

Today, we will read how to design the Combination Circuits. But before that, first I want to discuss the previous days quiz question.

(Refer Slide Time: 01:03)



Indian Institute of Technology, Kharagpur

Answer for Lecture 11 Quiz

1. Using D as the MEV, reduce
$Z = A'B'C'D + A'BC'D' + A'BC'D + AB'C'D' + AB'CD' + AB'CD + ABCD'$

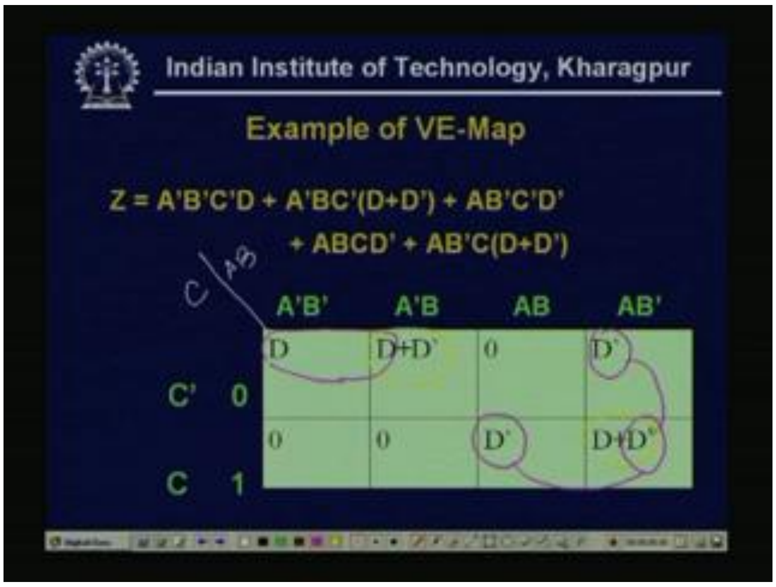
$Z = A'B'C'D + A'BC'(D+D') + AB'C'D' + ABCD' + AB'C(D+D')$

There was only one question and that was that using D as the MEV, that reduce the function Z equal to A dash, B dash, C dash, D plus A dash B C dash D dash plus A dash B C dash D plus A B dash C dash D dash plus A B dash C D dash plus A B dash C D plus A, B, C, D dash. Now, that last day we have discuss the map entered variable; that means, this is a technique that which reduces the larger number of variables to it is the lower one.

So, here the Z function is the four variable function A B C D, there are four variables. So, now we can rearrange this function like that, see the first means term A dash B dash C dash D is at the as it is. Now, that A dash B C dash, A dash B; that means the second term and A dash B, C dash D means the third term, we take the A dash B, C dash common, then it will be D plus D dash.

Now, A B dash C dash D dash, means the 4th minterm this is A B dash, C D dash fifth minterm and similarly the from the 6th and the 7th minterm. We take or we can write the 6th and 7th together by taking A B dash C common into D plus D dash.

(Refer Slide Time: 03:19)



Indian Institute of Technology, Kharagpur

Example of VE-Map

$$Z = A'B'C'D + A'BC'(D+D') + AB'C'D' + ABCD' + AB'C(D+D')$$

C \ AB		A'B'	A'B	AB	AB'
C'	0	D	D+D'	0	D'
C	1	0	0	D'	D+D'

Now, if we use the normal K map, then as it is a four variable, then it should be a 4 by 4 means 16 minterms should be there maximum possible 16 minterms. Now, we are reducing 1 and we have taking D as the MEV; that means that map entered variable. So, that it becomes now three variable A B C, so how we will be doing that thing. See that this is the K map for three variable and here this is for A B, the two variables and this is for C.

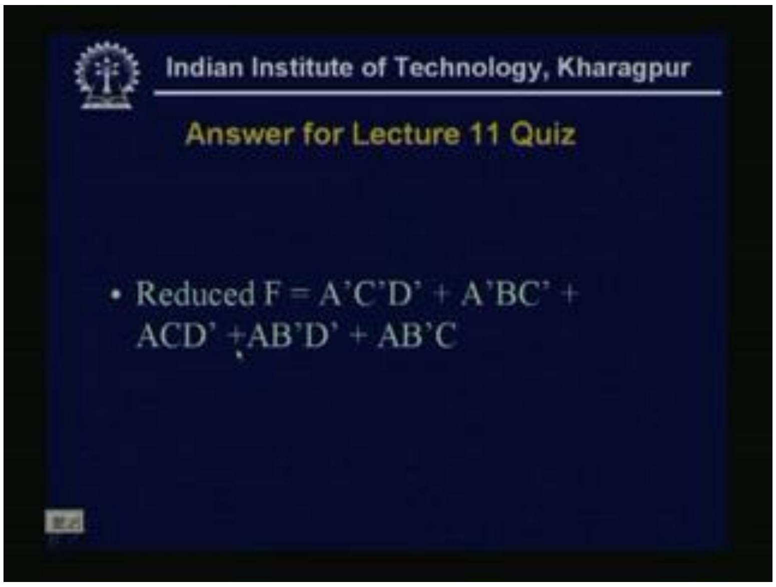
So, we can write the thing, that C, that A dash B dash C dash D, so the first one is A dash B dash C dash, so I can write the element first element as D. Similarly, the second thing, the second minterm, now A dash B C dash, so A dash B C dash and both D plus D dash are there, there is no minterm for AB C dash, so it is 0. Now, again A B dash C dash, the 4th one, it is the term is D dash, so I write D dash here.

Now, A B C D dash, so A, B, C, D dash, this is the second row third column, this is A, B, C, D dash, so A, B, C the D dash, I put here and the second row 4th column A B dash C into D plus D dash, so I put D plus D dash here. Now, if we try to select the couple, the same way that we have done in the K map, because our aim is to reduce the function.

So, how we can do that thing, see here these are two adjacent elements, where other than 0 value exists, a normal K map one was there. Now, see here both D and D dash exist, here one D is there, so I can put a couple here by doing D and D dash, but here D plus D dash, itself will make a couple. Similarly, here D plus D dash itself will make a couple, but then adjacent D dash, this adjacent D dash will make a couple with this D dash.

Similarly, here this D dash will make a couple with this D dash, so now I can get 1, 2, 3, 4, 5. Such couples, where the one element itself, there are two such element that D plus D dash, which make some reduction.

(Refer Slide Time: 07:09)



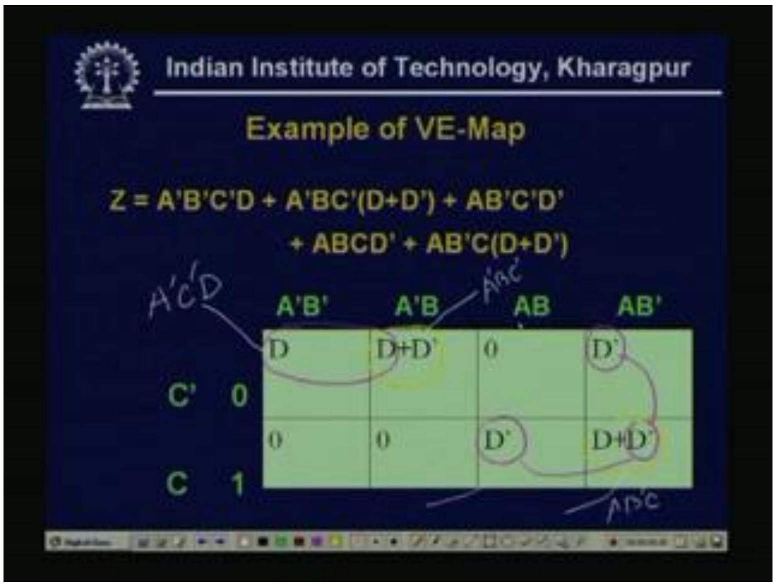
Indian Institute of Technology, Kharagpur

Answer for Lecture 11 Quiz

- Reduced F = A'C'D' + A'BC' + ACD' + AB'D' + AB'C

Now, I see if now we do that thing, see reduced function will be A dash C dash D dash, how this is A dash C dash D dash.

(Refer Slide Time: 07:22)



Indian Institute of Technology, Kharagpur

Example of VE-Map

$Z = A'B'C'D + A'BC'(D+D') + AB'C'D' + ABCD' + AB'C(D+D')$

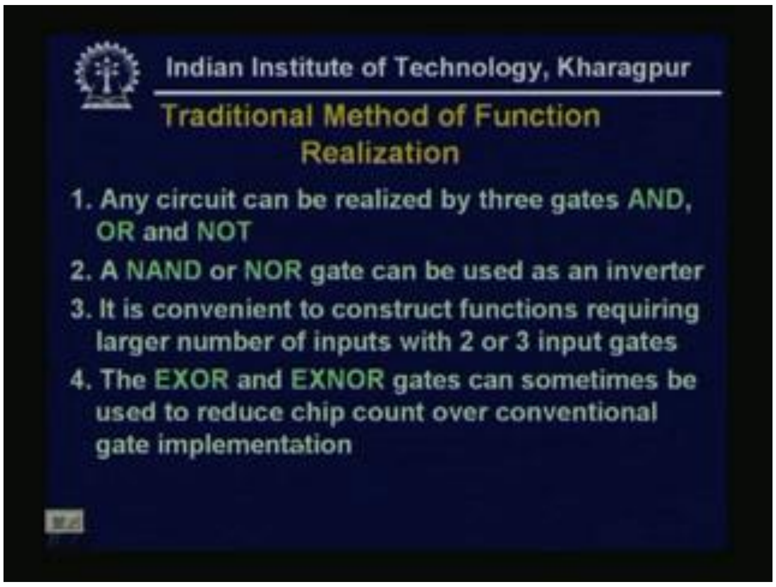
$A'C'D$ $\quad$ $A'BC'$ $\quad$ $AB'C$

		A'B'	A'B	AB	AB'
C'	0	D	D+D'	0	D'
C	1	0	0	D'	D+D'

See, when we will be taking this couple, that this D and adjacent D, then this becomes A dash C dash D, so this couple is A dash C dash D. Now, this one reduces to A dash B C dash, because D plus D dash is 1, so A dash B C dash. Similarly, that this one reduces to A B dash C, not this one, this one will make a couple with this D dash and they will form they will form D dash.

So, this will be A B, C A, A C, D dash and, this one and this one will form, because they are in same column, so A B dash D dash. So, 1, 2, 3, 4, 5 this will are the reduced function and see this is the A dash C dash D dash, A dash B C dash A C dash A B dash D dash plus A B dash C, this are the reduced function.

(Refer Slide Time: 09:40)



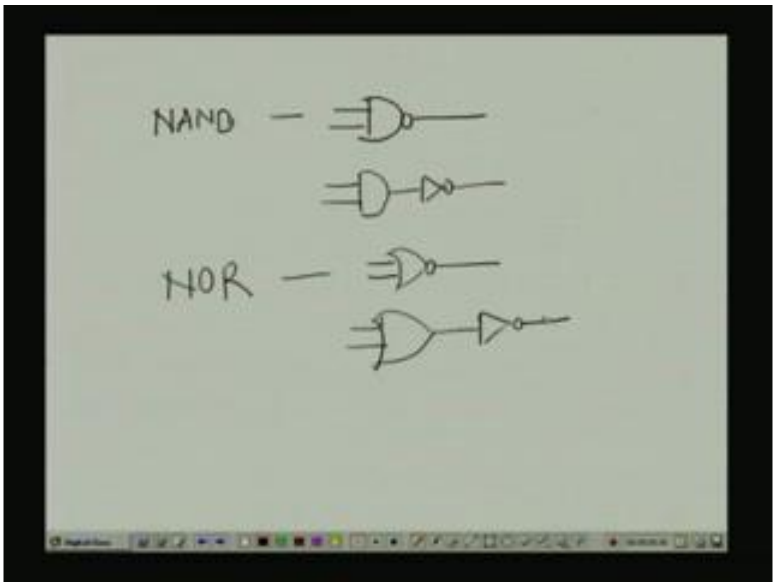
Indian Institute of Technology, Kharagpur

Traditional Method of Function Realization

1. Any circuit can be realized by three gates AND, OR and NOT
2. A NAND or NOR gate can be used as an inverter
3. It is convenient to construct functions requiring larger number of inputs with 2 or 3 input gates
4. The EXOR and EXNOR gates can sometimes be used to reduce chip count over conventional gate implementation

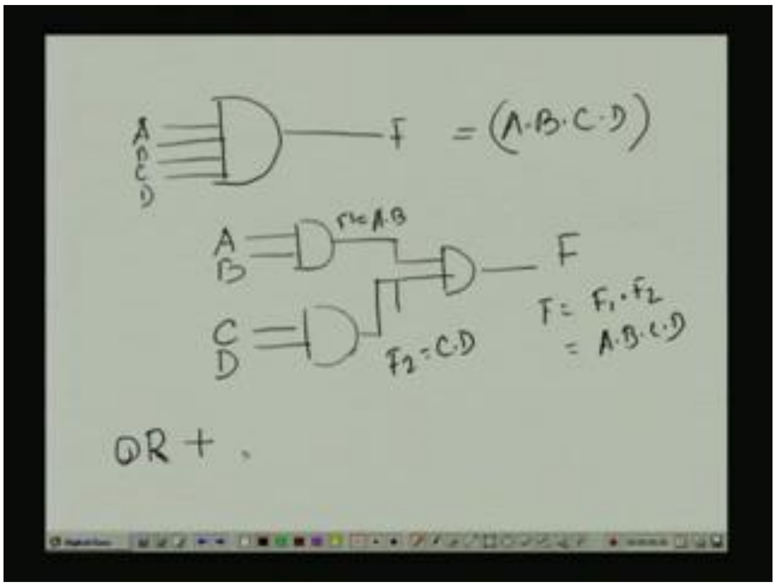
Now, we start the combinational circuit design, first we again we recapitulate the some of the traditional method of function realization. All ready, we have discuss that the Karnaugh map; that means, given a table or the some expression, Boolean expression from there, how we can reduce that thing, then how we can realize that functions. Now, normally that any circuit can be realized by three gates, AND, OR and NOT and these are same as that of our three operators, that dot, plus and the inversion or the complement.

(Refer Slide Time: 10:35)



Now, in NAND or NOR gate can be used as an inverter, because all ready we have discussed that NAND is nothing but the NAND is we can write like that, say this is a two input NAND, which is a two input AND, followed by a inverter NOT. Similarly, that NOR this is a two input, NOR that can be replaced by two input, OR followed by a inverter, so always the NOT can be replaced by NAND and NOR.

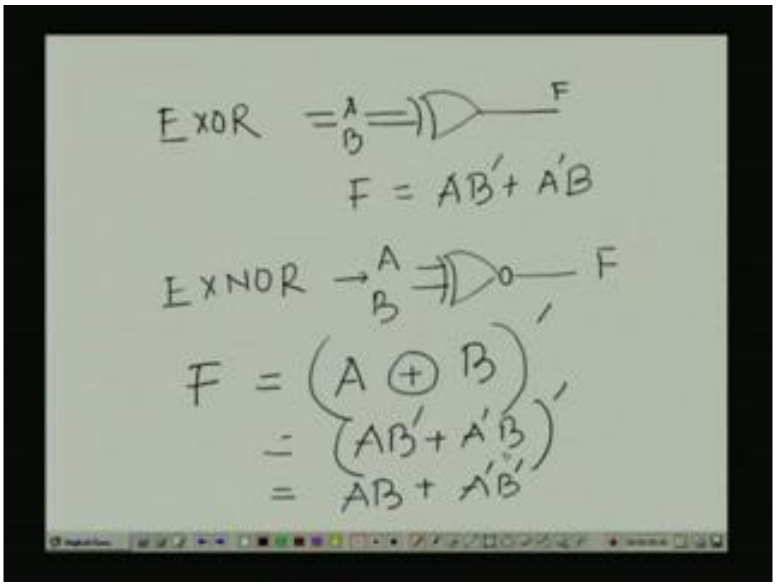
(Refer Slide Time: 11:39)



Now, it is convenient to construct functions requiring larger number of inputs with two or three input gates, what does it mean, say I have four input AND, gate, say I have some four input AND, gate, what I can do that I can make a two input AND gate. Another, two input AND gate, say these are A, B, C, D four inputs and these are two inputs A B, these are two inputs C D and again I can put another AND, gate here.

So, this will be F equal to A dot B dot C dot D and here it is A dot B say this is F 1, say this line is F 2 equal to C dot D and this is F equal to F 1 dot F 2 means A dot B dot C dot D. Similarly, the same thing will happen for OR or others operations source, now the EXOR and the EXNOR gates can sometimes be used to reduce chip count over conventional gate implementation.

(Refer Slide Time: 13:21)



$F = AB' + A'B$

$F = (A \oplus B)'$

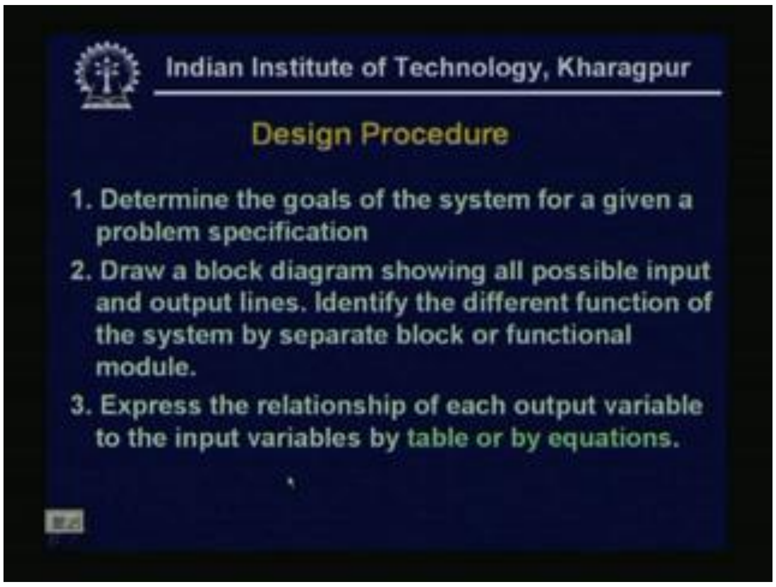
$= (AB' + A'B)'$

$= AB + A'B'$

So, what are the EXOR and EXNOR circuits, say EXOR circuits wire the notation is like that, say this is a two input EXOR, A, B and F where F is A B dash plus A dash B. Now, similarly EXNOR is say two input, then EXOR, one naught is there and again if it is a two input AND gate, so F is actually A normally, we give EXOR notation is like that.

So, A EXOR B complement and this is A B dash plus A dash B and this becomes, if we apply the Demorgan's law, then it becomes A B plus A dash B dash. So, this two sometimes, the function are realized by employing only this two gates, EXOR and EXNOR.

(Refer Slide Time: 15:02)



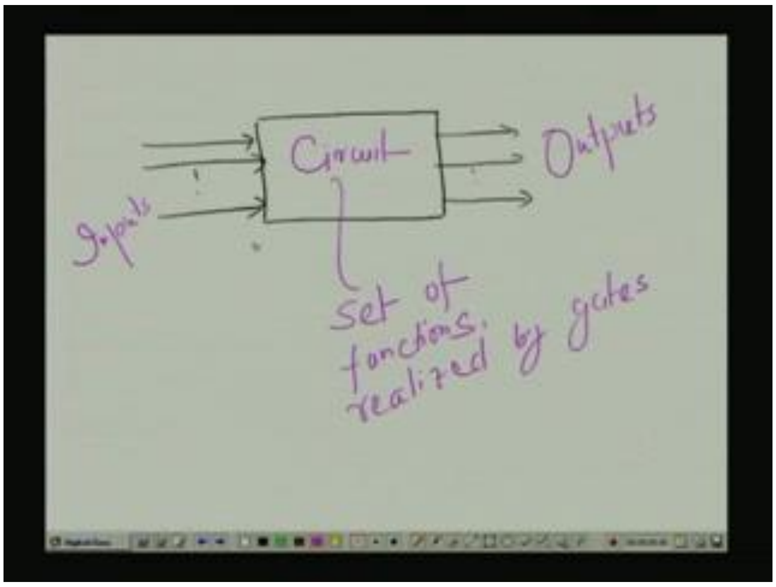
Indian Institute of Technology, Kharagpur

Design Procedure

1. Determine the goals of the system for a given a problem specification
2. Draw a block diagram showing all possible input and output lines. Identify the different function of the system by separate block or functional module.
3. Express the relationship of each output variable to the input variables by table or by equations.

Now, there are some design procedure, see first normally the combinations of gates the concept is very simple.

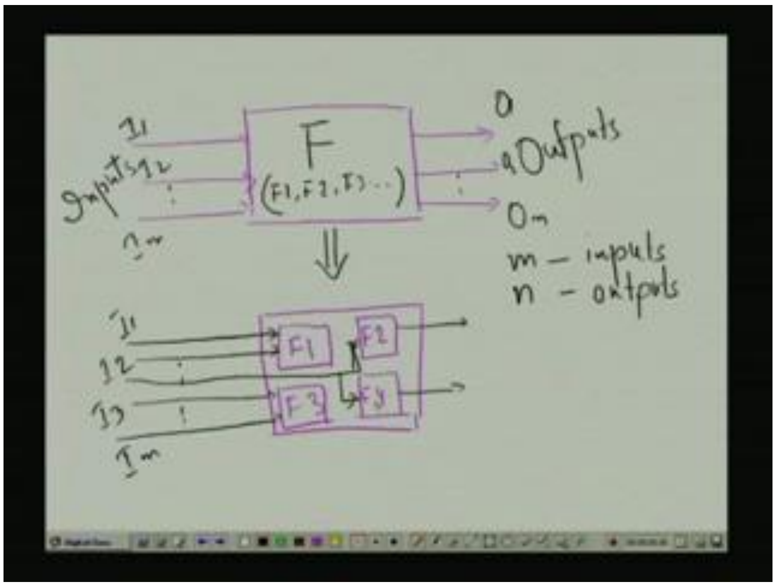
(Refer Slide Time: 15:16)



See here, the combination of circuit means, see this is one functions or the circuit to be realized, just we are considering as if this is a black box, this is a set of set of inputs and these are a set of outputs. So, these are inputs these are outputs, these are my circuit or what I can tell, this is a set of functions realized by gates. Now, here this function combinational circuit means, this function or the output behavior, the output are only depends on the inputs, the set of outputs depends on the set of inputs.

Now, how to design this combinational circuits, so first we see a simple design procedure. The first step is determine the goals of the system for a given problem specification, means given a problem first we will decide, that what will be the output of the system, what we have to compute or we have to determine. Then, second is that, draw a block diagram, showing all possible input and output lines, identify the different function of the system by a separate block or functional module.

(Refer Slide Time: 17:37)



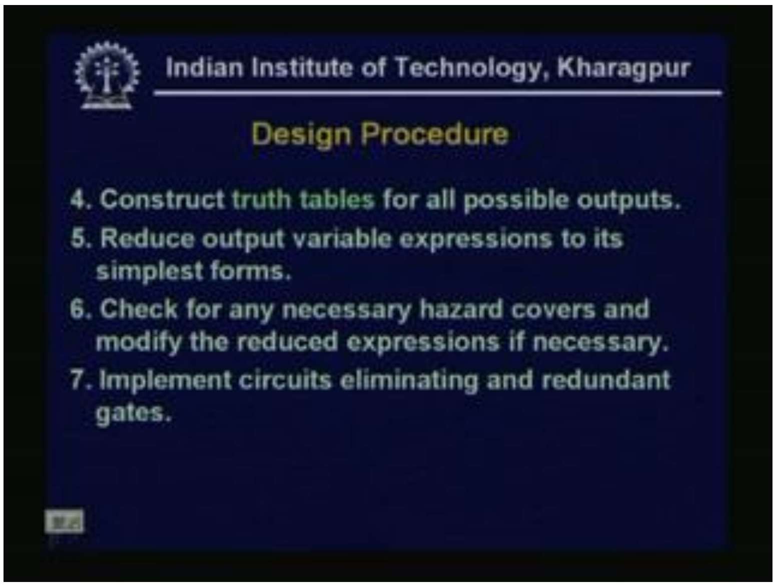
Now, say just now I mentioned, that is if I explain the second point, see we have, this is a block diagram level, there are set of inputs and set of outputs. Now, see that this is a function F realized, now this F can be a large circuit or just now I mentioned that, this is a set of functions, say this can be a set of functions F 1, F 2, F 3 like that. Now, what we can do, that as if these we can identify as separate blocks and but the primary inputs or the inputs are same.

Say, it may happen that, this I 1 say inputs are there are m inputs and say these are I 1, I 2, to I m. Similarly, the outputs are O 1, O 2, say O n number of outputs; that means, m inputs and n outputs. Now, say same say as if I 1, I 2 are fed to F 1, whereas say some I 3 or I m, that is fed to F 3 or one can be fed to F 2, some other can be fed to F 4 like that, similarly the outputs can be.

So, the original function is partition into smaller modules and identify that blocks and there are separate functional module. Now, the third step express the relationship of each output variable to the input variables by table or by equations, now what we are doing that for the whole function. Now, for the each smaller module, what we can do we that express the output variable to the input variables.

Because, truth table is nothing but all possible input output combinations or the all set of equations. If we consider; that means, for set of equations for all functional modules F 1 to F n, then they will actually constitute the whole system.

(Refer Slide Time: 21:24)



Indian Institute of Technology, Kharagpur

Design Procedure

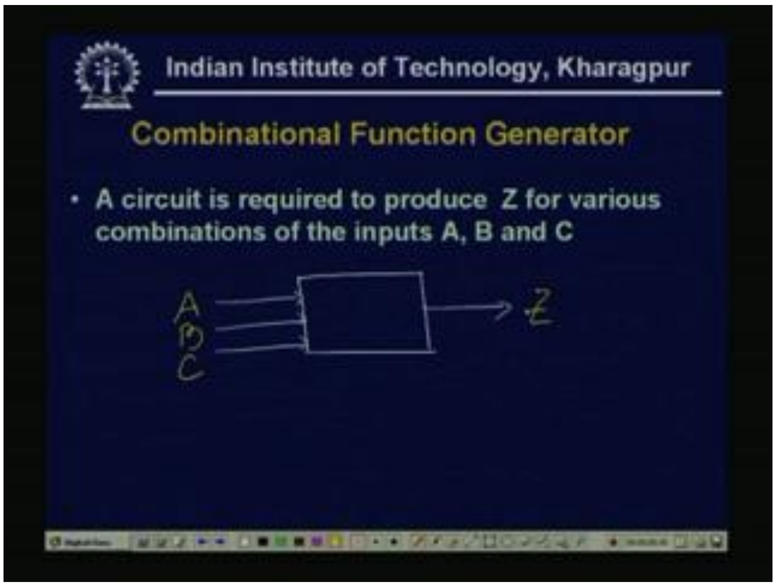
4. Construct truth tables for all possible outputs.
5. Reduce output variable expressions to its simplest forms.
6. Check for any necessary hazard covers and modify the reduced expressions if necessary.
7. Implement circuits eliminating and redundant gates.

So, we will now express the relationship of the output variable to the input variables, then we will construct the truth tables for all possible outputs and reduce output variable expression to it is simplest form. Already, we have discussed, how we can reduce that thing, so any using anyone of the method, we can reduce that expression. Now, check for any necessary hazard covers and modify the reduced expressions if necessary.

Now, we will say whether my expressions or the set of expressions reduced, whether they are there are some static hazards or dynamic hazards exist are not. Then, the last day, the method we discussed by introducing the redundant implicant, we can remove the hazards. So, we will modify that is why, that we have to modify the reduced expression, if necessary, means if the hazards are there.

Now, we have a set of equations and reduced equations and then implement the circuits eliminating and redundant gates. So, this is one of the simple design procedure, now 1 by 1, we will see.

(Refer Slide Time: 22:40)



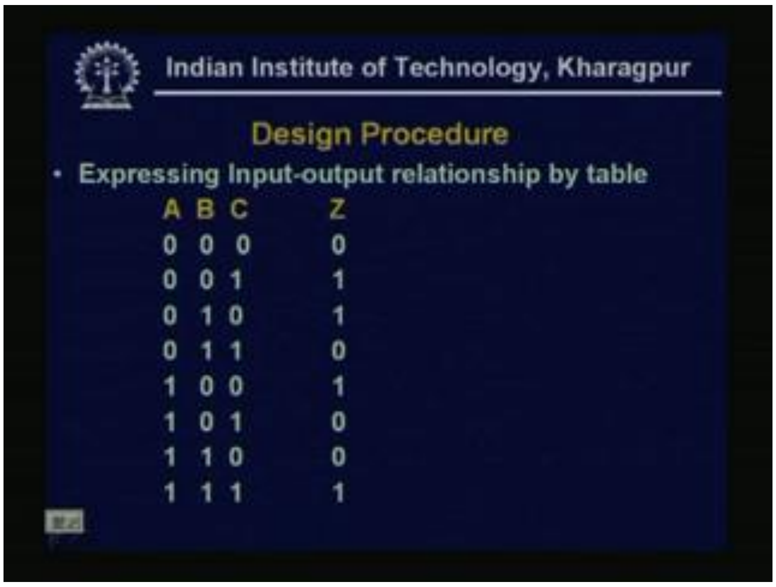
Indian Institute of Technology, Kharagpur

Combinational Function Generator

- A circuit is required to produce Z for various combinations of the inputs A, B and C

Now, first we see a combinational function generator, say a circuit is required to produce Z for various combinations of the A, B, C means say I am telling, I am taking one black box or say if this is my system, there are three inputs and one output. See inputs are A, B C and output is Z, now first I have to see that, what the function it realizes or what is the input output relationship.

(Refer Slide Time: 23:43)



Indian Institute of Technology, Kharagpur

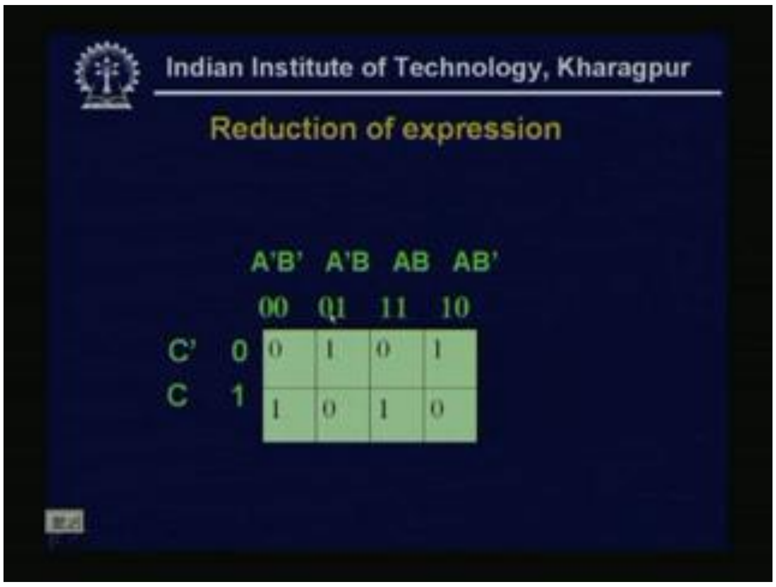
Design Procedure

- Expressing Input-output relationship by table

A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Now, expressing input output relationship by table, so here one table is given which actually gives that, if the A, B, C inputs are they take this values. Then, what will be the output, say this is one table, if it is this is 000, then output is 0, if it is 001 output is 1, similarly if it is 101, then A equal to 1, B equal to 0, C equal to 1, then the output Z is 0. So, this we have identified that this is the relationship; that means, output will be produced, if the inputs take this particular value.

(Refer Slide Time: 24:33)



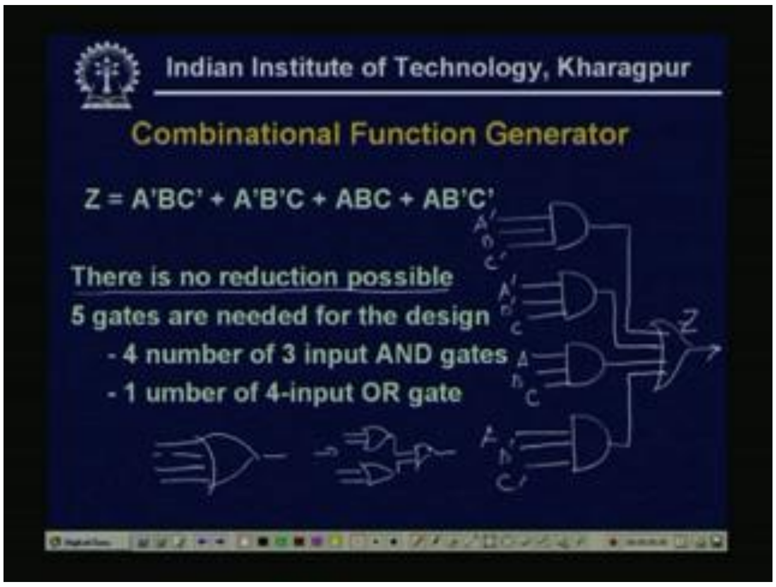
Indian Institute of Technology, Kharagpur

Reduction of expression

		A'B'	A'B	AB	AB'
		00	01	11	10
C'	0	0	1	0	1
C	1	1	0	1	0

Now, from this truth table, we can easily draw the Karnaugh map, as this is the three variable function, because the inputs are three, so we write that this three variables are A, B, C and see from the previous table that say for 000, it is 0. So, there are 1, 2, for 011 combination output is 0, for 101 means 0, this is 3, this is 4 and or 0, 3, 5, 6 for this 4 combinations or this four inputs, the output is 0. See that this for 000 means 0, this is 3, this is my 5 and this is 6, so there 0000 and rest are 1; that means, there are 1111.

(Refer Slide Time: 25:56)


Indian Institute of Technology, Kharagpur

Combinational Function Generator

Z = A'BC' + A'B'C + ABC + AB'C'

There is no reduction possible

5 gates are needed for the design

- 4 number of 3 input AND gates
- 1 umber of 4-input OR gate

Now, if we reduce this expression, then we will get that Z is A dash B C dash plus A dash B C plus A, B, C plus A B dash C, actually there is no reduction possible here, mainly we have the minterms, we have written. So, now if I want to realize this circuit, what will be the type of circuits, see there are 5 gates are needed for the design. Now, we are assuming that the variables and complimented as well as uncomplimented variables are available.

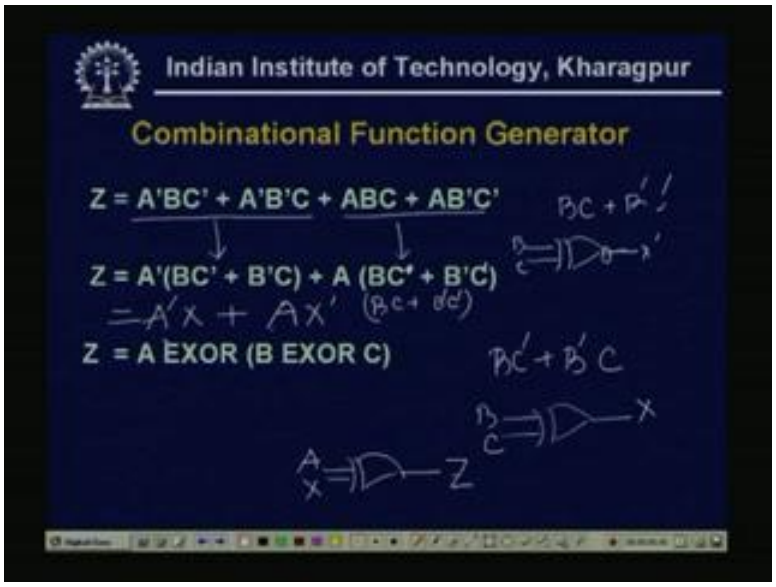
So, if we draw that this type of circuits there are 4, AND gate, there will be 4, AND gate and all have three inputs, here it is A dash B C dash, this is A dash B dash C, this is my A, B, C, this is my A B dash C dash. Now, these are sum of product forms, so this is OR and this will give my Z, so there are 4 gates needed, 4 number of three input AND gates, these are the 4 number of three input AND gates and 1 OR gate 1,4 input OR gate.

Now, typically or traditionally what we can do, instead of this three input AND gate, we can take two input AND gate. Instead of this four input OR gate, we can take two input

OR gate, followed by another input OR gate. That means, always this four input OR gate as all ready I mentioned, this can be realized to two inputs OR gate and again that will be another, so this will be the same thing.

Now, all ready we have seen from the Karnaugh map, that there was no reduction possible. But we can realize the circuit, because now we are discussing, how the expressions of the functions are realized by the gates.

(Refer Slide Time: 28:42)



Indian Institute of Technology, Kharagpur

Combinational Function Generator

$$Z = \underbrace{A'BC' + A'B'C}_{\downarrow} + \underbrace{ABC + AB'C'}_{\downarrow}$$

$$Z = A'(BC' + B'C) + A\,(BC' + B'C)$$

$$= A'X + AX' \quad (BC + B'C')$$

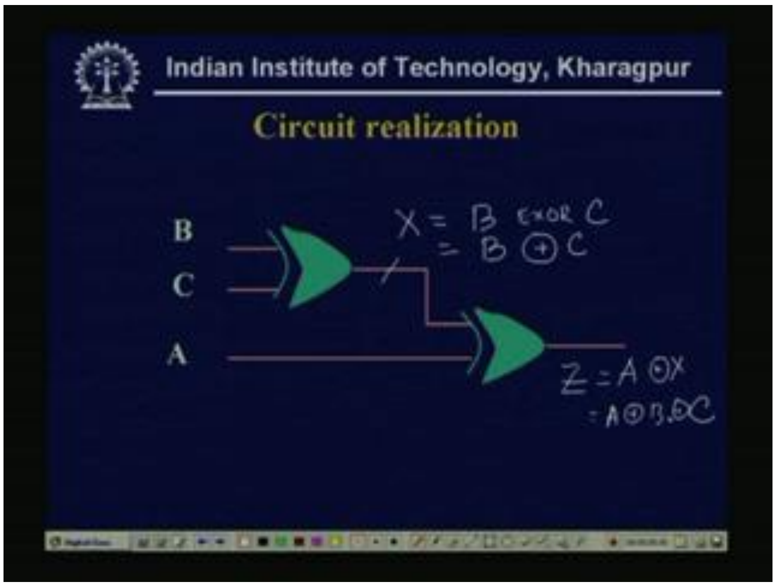
Z = A EXOR (B EXOR C)

So, even the reduction is not possible, but see this particular expression, I can take some common or terms a literal, say from the first two terms see from the first two terms, A dash B C dash and A dash B, B dash C, from here see I can take A dash common. So, this is BC dash plus B dash C, similarly the last two terms, I can take A common and this is BC dash plus B dash C.

Now, what is my B C dash, B C dash plus B dash C this is nothing but B EXOR C, so this is my B EXOR C, so this is X. So, how this expression becomes, then Z equal to A dash X and here this is A, what is this term B C dash plus B dash C dash this is nothing but EXNOR, all ready we have seen that this is EXNOR. That that means, this is my X dash, that B C plus B dash C dash. So, here BC dash will not be there this is BC plus B dash C dash this is BC plus B dash C dash, now this is my X dash.

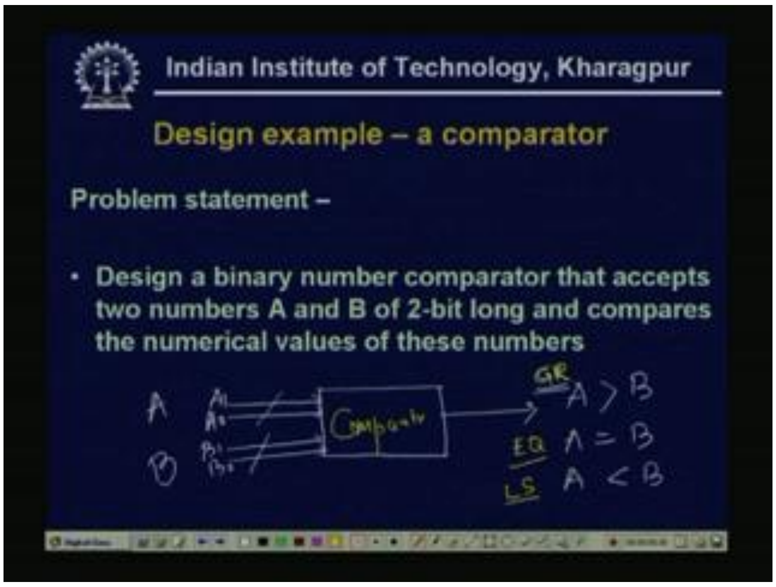
Now, A dash X plus A X dash, which is A EXOR which is A EXOR X, that means again I can take 1 and A and X and this is my Z, so this is my A EXOR. This expression becomes A EXOR X, so A EXOR and this is my A X means B EXOR C, so X is, this is my X B EXOR C.

(Refer Slide Time: 32:18)



So, the same function I can realize by this circuit, see this is my B EXOR C, so this is my X equal to B EXOR C or we can give a different notation, conventionally we use this thing, that B EXOR C. And this is my Z, this is again A EXOR X means A EXOR B EXOR C, now see if I use only EXOR, then instead of that 5 and OR gates that initially we have seen. The 5 AND OR gates, we have realized, there were only two EXOR gates are needed, so this can be a circuit realization of that particular circuit function.

(Refer Slide Time: 33:39)


Indian Institute of Technology, Kharagpur

Design example – a comparator

Problem statement –

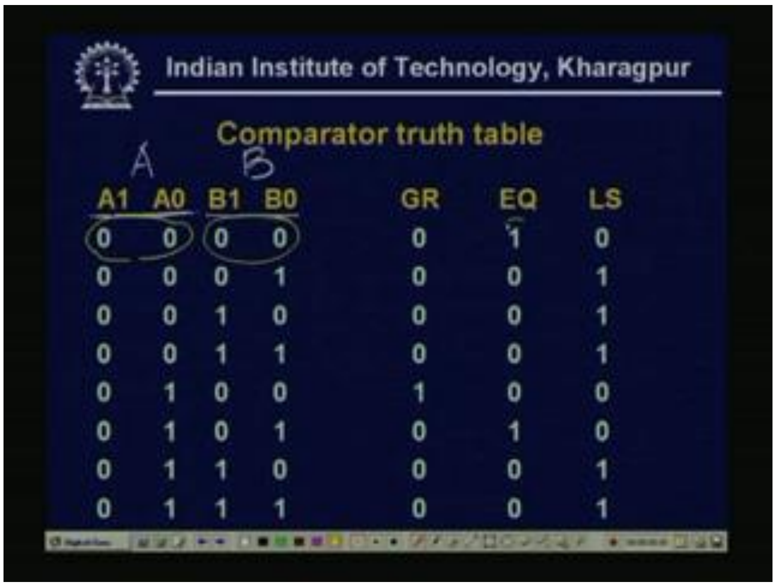
- Design a binary number comparator that accepts two numbers A and B of 2-bit long and compares the numerical values of these numbers

A
A_1
A_0
B
B_1
B_0
Comparator
GR $A > B$
EQ $A = B$
LS $A < B$

Now, we consider a bigger example, now see this is a comparator design, the problems statement first detail; that design a binary number comparator, that accepts two numbers A and B of two bit long and compares the numerical values of this numbers. So, if I draw that, say this system of that my comparator, there will be two inputs two bit. So, this is my A 1, A 0 and another two bit say B 1, B 0, so these are the two bits, two input A B each has two bits.

Then, output will be only 1, it will tell whether A greater than B, we are telling this is as a greater than A equal to B and whether A less than B, so this is my comparator, these we are telling EQ, these we are telling LS and this is my greater. So, 2 inputs and 1 output, 1 bit, so these we have to design, this is my comparator.

(Refer Slide Time: 35:55)


Indian Institute of Technology, Kharagpur

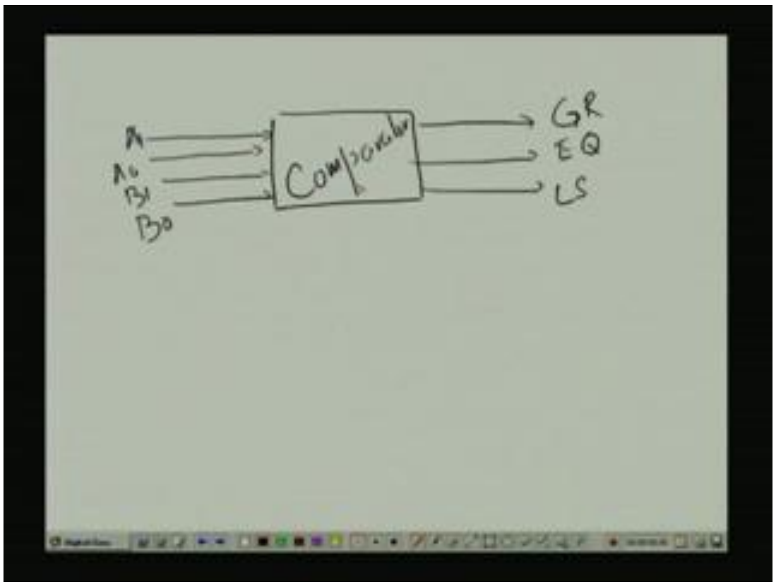
Comparator truth table

A1	A0	B1	B0	GR	EQ	LS
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1

Now, again first we try to identify the output input relationship, see that here, there are actually there are two, we have 2, 2 inputs, 2, 2 inputs and they are see this is one of the inputs A, this is another input B, both has two bits. So, as if we are thinking there are four variables, A 0, A 1, B 0, B 1, now we take all possible combinations, because they are the inputs.

So, all possible combinations of the four inputs and we see that what will be the output, means mainly we are trying to identify the output input relationship. See that, these are A 1is 00, B 1 is 00, the numerical value is 0, so these are equal, so that is why this is a 1 and greater and less than they are all 0.

(Refer Slide Time: 37:50)



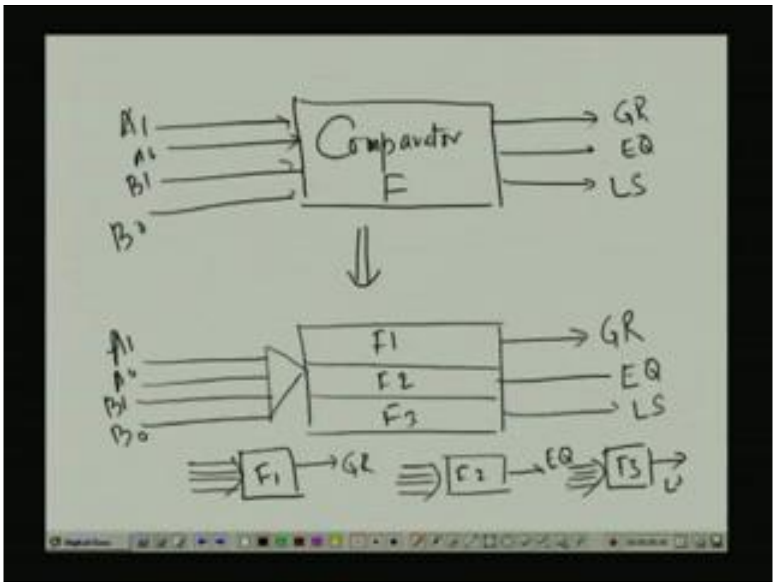
Now, what we can do then, that of instead of designing one output, what we can do, we can this type of thing also we can do that A, B, C, A not A, B, C that A 0, say A 0, A 1 means A 1, A 0, B 1, B 0. There are four inputs, I can make three outputs, one is greater equal and less, that also we can do and this is my comparator.

So, when see that 00, A 1, A 0 is 00 and B 1, B 0 is 01, so, what is the numerical value, A B has see this is 00 means, this is 0 value, this is 0 1 means 1 value, so A is less than B. So, that is why LS is producing a 1, whereas the GR line and the EQ line means greater and equal, they are giving 0. Similarly, we take this 1, say 0100, so 0100 see this 01, this is 1, this is 0, so it is A greater than A greater than B. So, this greater than is giving a 1, equal is giving a 0 and less than is giving a 0.

So, for all such conditions, now say the next 01, 01, 01, 01 this is 2, the value is 1, 01 the value is 1. So, A equal to B, means my EQ line means equal, that will produce a 1; that means, equal it is true and whereas, the greater and the lesser they are falls means, they are producing a 0. So, in this way we have put, what the three output lines will generate the value depending on a particular combination of the four inputs.

So, this is the whole combination or the all possibly inputs combinations are there and for them, what output values should be generated by the system; that is given. Again, for this all one values, these are A value is 3, B value is 3. So, A equal to B and so EQ is 1, whereas greater GR line and LS lines they are 0, so this two tables are given.

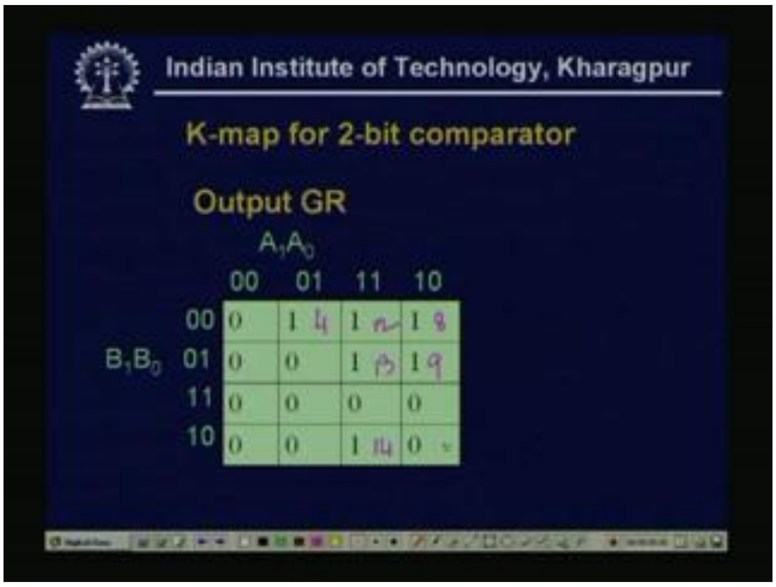
(Refer Slide Time: 41:45)



Now, all ready we have discussed, when that design procedures, I told, see as if here there is one comparator function F and there are four inputs A 1, A 0, B 1, B 0, there are three outputs GR EQ and LS. Now, I can replace or identify the sub blocks or sub modules, means this F I have as if partitioned into 3, the F 1, F 2 and F 3 and say this is producing say GR, F 2 is producing say EQ and this producing a LS.

Say, as if all inputs are fed to all the modules this is the situation and now this F 1, F 2, F 3, we have to design, so what we can do now, as if this we are taking we are partitioning this as a three sub problems. Actually there are four inputs and three outputs, as if we are partitioning four inputs F 1 has four input, one output, F 2 has four input, one output and F 3 as that means, if we take this 1. So, I can break, this is my F 1 and this is GR, I can break F 2 output is EQ, then I can break F 3 output is LS, now these functions, I have to design.

(Refer Slide Time: 44:23)



Indian Institute of Technology, Kharagpur

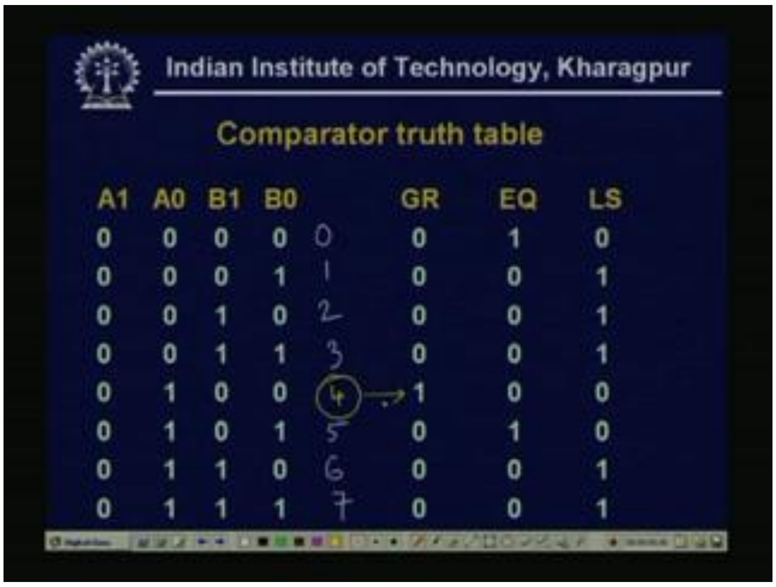
K-map for 2-bit comparator

Output GR

B_1B_0 \ A_1A_0	00	01	11	10
00	0	1 4	1 12	1 8
01	0	0	1 13	1 9
11	0	0	0	0
10	0	0	1 14	0

Now, we will see for output GR, now again we have to revisit that truth table that means, for which minterms, the output is 1, and for which minterms output is 0.

(Refer Slide Time: 44:50)



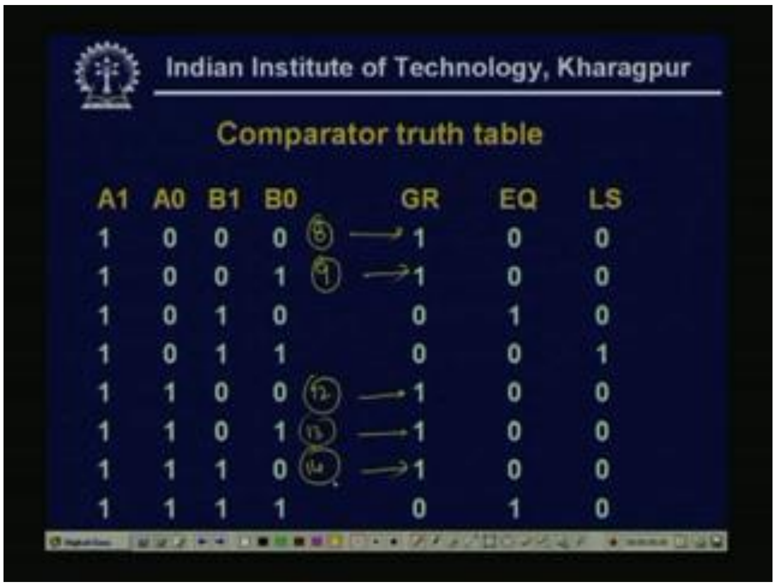
Indian Institute of Technology, Kharagpur

Comparator truth table

A1	A0	B1	B0		GR	EQ	LS
0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1
0	0	1	0	2	0	0	1
0	0	1	1	3	0	0	1
0	1	0	0	$\textcircled{4} \rightarrow$	1	0	0
0	1	0	1	5	0	1	0
0	1	1	0	6	0	0	1
0	1	1	1	7	0	0	1

So, now we will consider or we concentrate only on the GR, now see for the decimal value, if we write for 0, 1, 2, 3 then 5, 6, 7 except 4, so for 4, it is generating a 1.

(Refer Slide Time: 45:32)



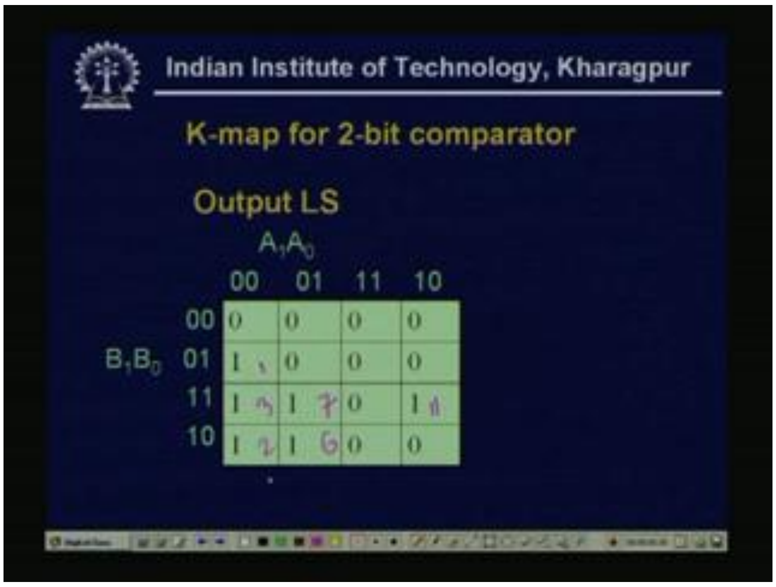
Indian Institute of Technology, Kharagpur

Comparator truth table

A1	A0	B1	B0		GR	EQ	LS
1	0	0	0	8 ⟶	1	0	0
1	0	0	1	9 ⟶	1	0	0
1	0	1	0		0	1	0
1	0	1	1		0	0	1
1	1	0	0	12 ⟶	1	0	0
1	1	0	1	13 ⟶	1	0	0
1	1	1	0	14 ⟶	1	0	0
1	1	1	1		0	1	0

Similarly, for 8, it is generating a 1, 9 it is generating a 1, then 12 it is generating a 1, 13 it is generating a 1, 14 it is generating a 1, so 4, 8, 9, 12, 13, 14. Now, we see this is, if I draw the Karnaugh map, see this is 0001, actually this is 0100. So, this is my 4, then this is 1100 means, this is 12, this is 8, then 13, 1001 means 9, 1110 means 14. So, 4, 8, 9, 12, 13, 14, so for this it is giving a just again we check that, if it is 4, 8, 9, 12, 13, 14, so this is the minterms of 1.

(Refer Slide Time: 48:07)



Indian Institute of Technology, Kharagpur

K-map for 2-bit comparator

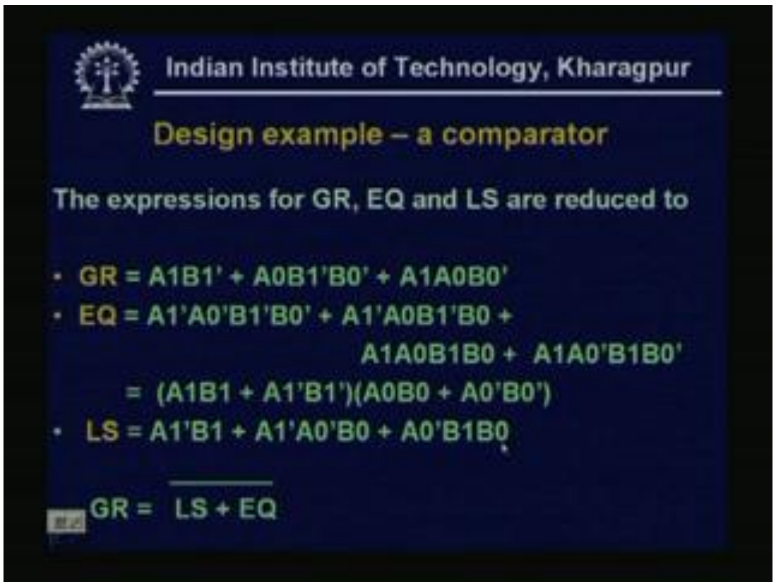
Output LS

B_1B_0 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	1 1	0	0	0
11	1 3	1 7	0	1 11
10	1 2	1 6	0	0

So, now if we the same thing, we see for LS, that see that A 1, A 0 is 00, B 1, B 0 is 01, so this is less so; that means, here it is, this is a lesser value A is less than B, so this is my 1. Again, 1010 is less than 11 means 2 is less than 3, so this is 1011 means 11 and similarly 00 is less than 1100 is less than 1001 is less than 11, now 0111 means 8, 4 or 4, 2, 1 this is 7 and this is 6, so for 1, 3, 1, 2, 3, 6, 7, 11 for this values are less.

Now, similarly we can write that equal also, see for LS, now if we see that, this is for this for greater 01 greater than 00, all ready we have seen these are greater. Similarly, we can draw that equal also.

(Refer Slide Time: 50:05)



Indian Institute of Technology, Kharagpur

Design example – a comparator

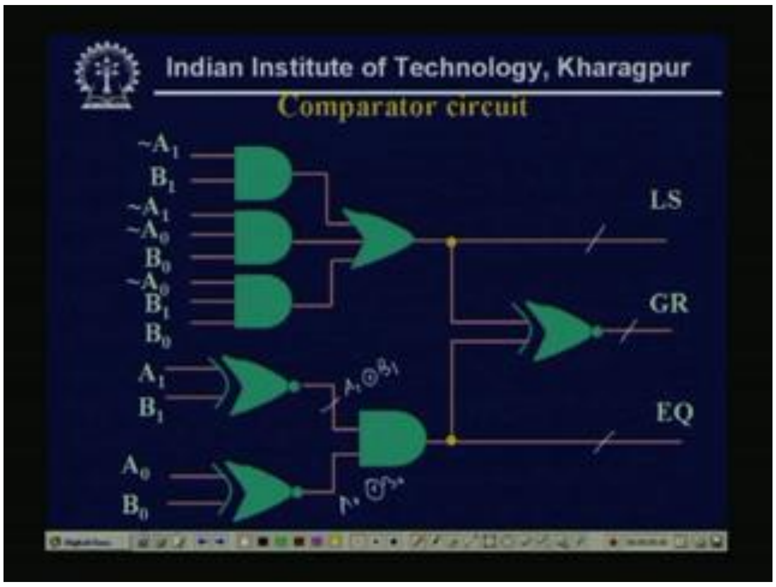
The expressions for GR, EQ and LS are reduced to

- GR = A1B1' + A0B1'B0' + A1A0B0'
- EQ = A1'A0'B1'B0' + A1'A0B1'B0 + A1A0B1B0 + A1A0'B1B0'
  = (A1B1 + A1'B1')(A0B0 + A0'B0')
- LS = A1'B1 + A1'A0'B0 + A0'B1B0

$GR = \overline{LS + EQ}$

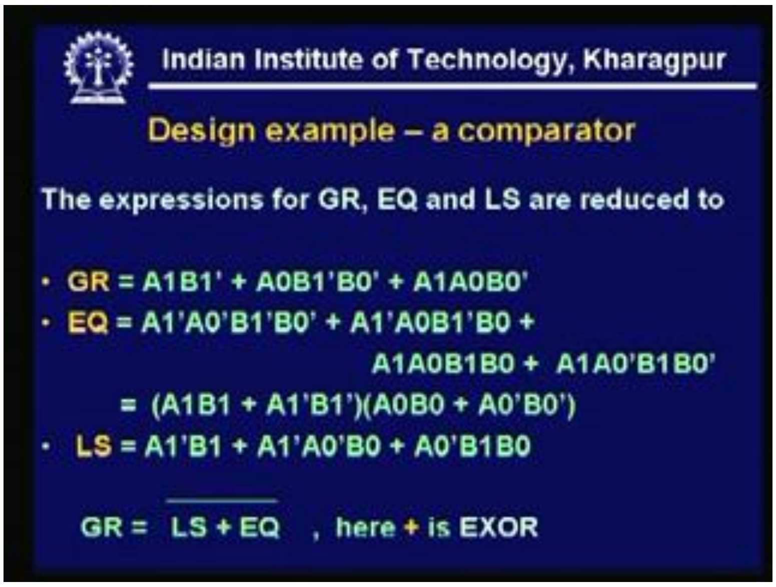
Now, if we write the expressions for that three Karnaugh maps, now that if we write the expressions for GR EQ and LS, they are reduce to GR is A 1, B 1 dash A 0, B 1 dash B 0 dash, A 1, A 0, B 0. Similarly, EQ will be A 1 dash A 0 dash B 1 dash B 0 dash, A 1 dash A 0, B 1 dash B 0, A 1, A 0, B 1, B 0 plus A 1, A 0 dash B 1, B 0 dash and this equal to A 1, B 1 plus A 1 dash B 1 dash A 0, B 0 plus A 0 dash B 0 dash. And the LS line means less that line will be A 1, dash B 1 A 1 dash A 0 dash B 0 plus A 0 dash B 1, B 0.

(Refer Slide Time: 51:01)



So, what we can do now if we want to realize this circuits, see for that, first we see that for that equal line, see equal line, this is A 1, EXOR B 1 and A 0 EXOR B 0 and then they are ended. Means that, this will be this is A 1, EXOR B 1, this is A 0 EXOR B 0 and then it will be my equal line. Similarly, the LS line will be that mainly the realized by the AND, OR circuits and the GR line will be AND, OR and realization by the AND, OR, EXORs, so in this way, we can design a combinational circuits.

(Refer Slide Time: 52:04)



Indian Institute of Technology, Kharagpur

Design example – a comparator

The expressions for GR, EQ and LS are reduced to

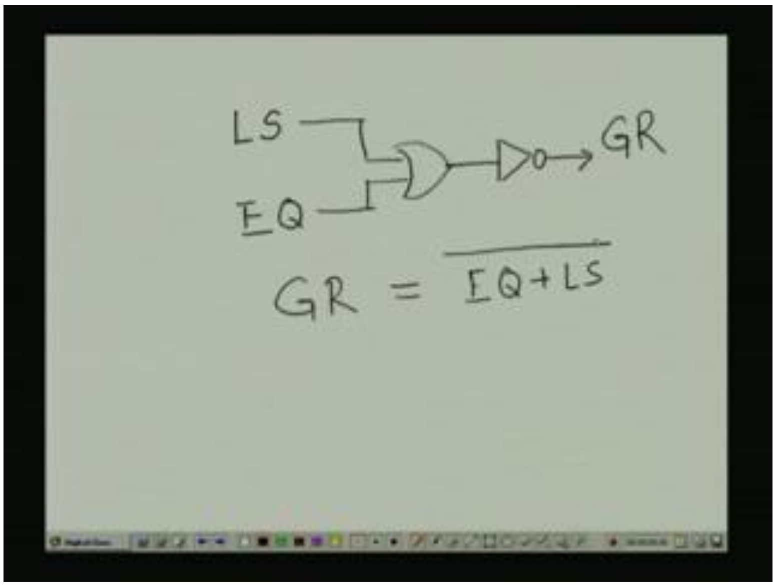
- GR = A1B1' + A0B1'B0' + A1A0B0'
- EQ = A1'A0'B1'B0' + A1'A0B1'B0 + A1A0B1B0 + A1A0'B1B0'
  = (A1B1 + A1'B1')(A0B0 + A0'B0')
- LS = A1'B1 + A1'A0'B0 + A0'B1B0

GR = $\overline{\text{LS + EQ}}$, here + is EXOR

Now, we see that the previous thing, now we see that the previous thing, see that if we this is one type of realization, where mainly the AND, OR and the EXOR, these are the three gates, we have employed, we have used. Now, if we study this greater equal and less than line, then we will be seeing that, this greater line will be less compliment of the less than and the equal line.

That means, that if we again see that circuit that here that, three separate the three outputs are designed separately. Instead of that what we can do that if the LS and equal this two are designed.

(Refer Slide Time: 53:22)

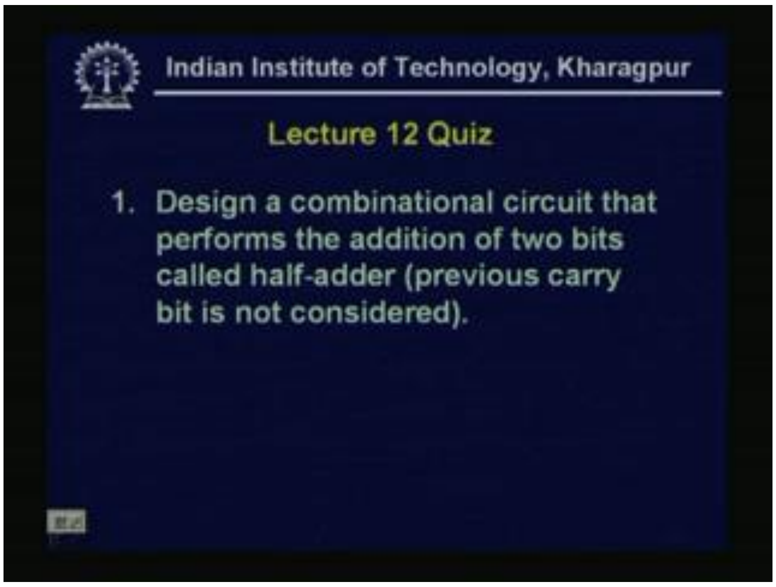


Say LS ones the LS and EQ, this two outputs are realized, then as if we can we use this or we can use this two outputs and this is OR and then actually this is a NOR, OR and then we can inverted that thing, which will give the greater line. So, this is that GR is EQ plus LS complement, so this can be in this way we can again still reduce that GR circuits. Now, see mainly this thing has been done here, that LS and EQ and this is by examine that thing.

So, in this way, that first we have to from the problems specification, we have to set our goals, that what we have to determine; that means, the outputs. Then, we have to identify the relation between the output and inputs, then we can reduce the expressions and using some Karnaugh maps, we can reduce the expression sets, then from that set of reduced expression, we can realize the gates.

Now, again the realization can be of different type, just now what we have discussed that can be simple AND, OR, EXOR, EXNOR circuits or that can be a different type of realization process, again we will discuss further.

(Refer Slide Time: 55:16)



Indian Institute of Technology, Kharagpur

Lecture 12 Quiz

1. Design a combinational circuit that performs the addition of two bits called half-adder (previous carry bit is not considered).

So, today's quiz is just design a combinational circuit that performs the addition of two bits called half adder, that means the previous carry bit is not considered, only that there are two operands A and B and we know the addition operator. Then, we have to design a combinational circuit which does this operation, so, this is the quiz for this lecture.

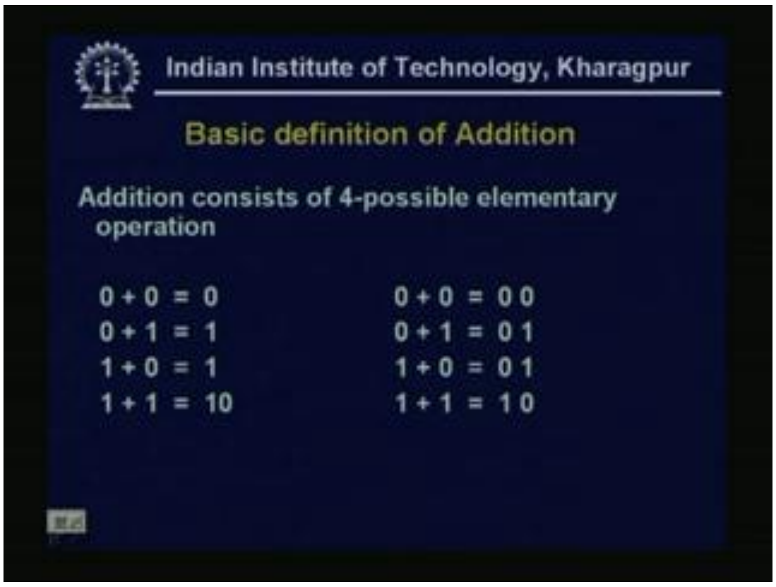
Thank you.

Digital Systems Design
Prof. D. Roychoudhury
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 13
Design of Adder Circuits

In the last class, we have read that, how we can design the combinational circuits, what is the conventional design procedure and we have seen a comparator design. Now, today we will see how the common circuits, such as some very popular circuits and very important circuits, that adder. The different type of adders can be designed, because this adder is nothing but the combinational one.

(Refer Slide Time: 56:46)



Indian Institute of Technology, Kharagpur

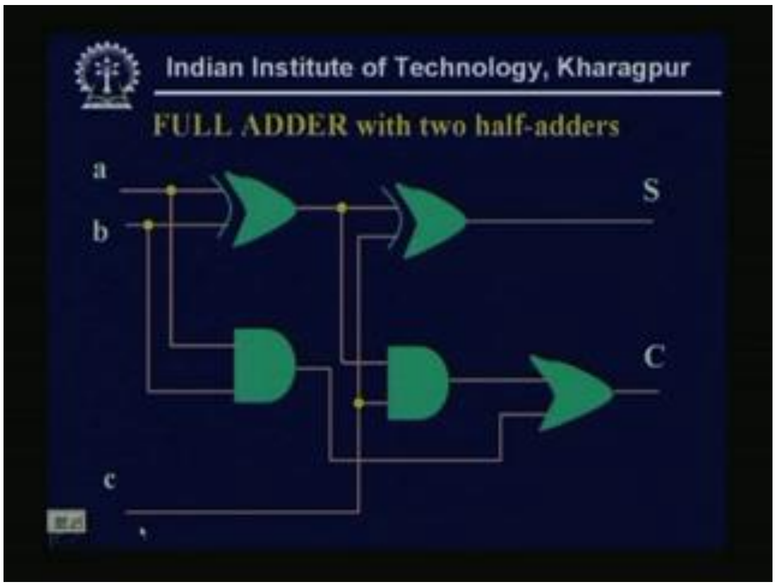
Basic definition of Addition

Addition consists of 4-possible elementary operation

0 + 0 = 0	0 + 0 = 0 0
0 + 1 = 1	0 + 1 = 0 1
1 + 0 = 1	1 + 0 = 0 1
1 + 1 = 10	1 + 1 = 1 0

So, as it is name implies the addition consists of the adder does the addition operator and this consists of four possible elementary operation.

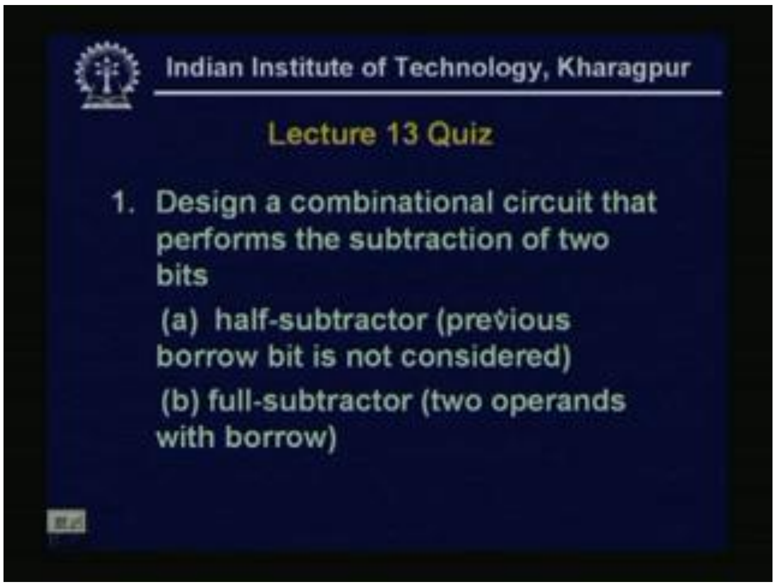
(Refer Slide Time: 57:17)


Indian Institute of Technology, Kharagpur
FULL ADDER with two half-adders
a
b
S
C
c

Say, there are two bits, say this is plus, what we have shown that for each type of adder half adder and full adder a number of different type of realization is possible and mainly depending on the application. That means, whether we want to reduce the area or we want to reduce the power, that we will select that type of realization consisting of AND, OR gates or EXOR gates or the NOT gates etcetera.

Now, today's this quiz question is again this is a combination circuit design and very similar type, that we have discuss the adder.

(Refer Slide Time: 58:18)



Indian Institute of Technology, Kharagpur

Lecture 13 Quiz

1. Design a combinational circuit that performs the subtraction of two bits
   (a) half-subtractor (previous borrow bit is not considered)
   (b) full-subtractor (two operands with borrow)

And now, the problem is design a combinational circuit that performs the subtraction of two bits, mainly the addition we have discussed and the subtraction for similar to the half adder and full adder. First, we have to do the half subtractor and then it is full subtractor, so first we have to define, what do you mean by half subtractor, half subtractor means that, again this will be a single bit minus, where that previous borrow bit, where if for addition, it was the carry bit and now it is a borrow bit, that is not considered.

And for full subtractor, the two operands as well as the borrow bit from the previous subtraction should be considered. So, these are the quiz question for this current lecture.

Thank you.