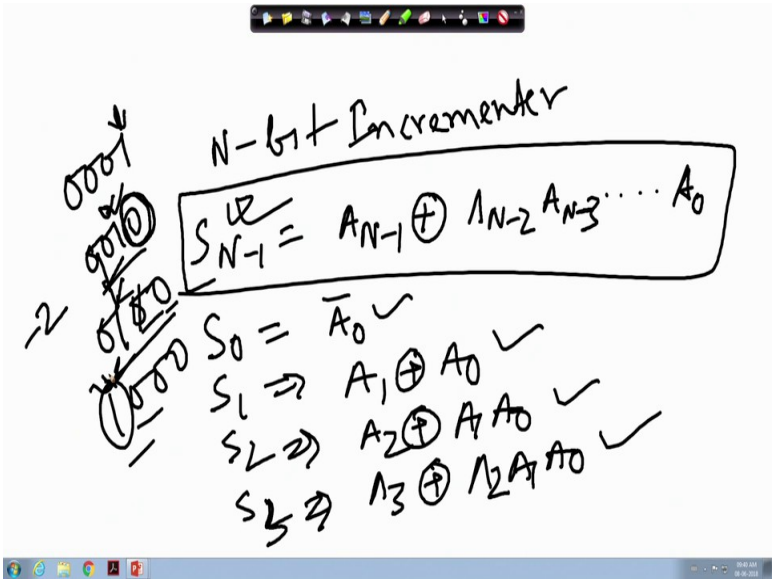
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 05
Algorithm to Efficient Architecture Mapping (Contd.)

Hello, everyone. Welcome back, to the course on Architecture Design of ICs. So, in the last class we have seen that means how efficiently we can implement the algorithm into the architecture. So, as an example of that we have seen the design technique for one; that means, N bit generic incrementer design, ok. So, initially we have started with what? Initially we have started with 4 bit incrementer design, ok. So, whenever we have we have initially we have followed the general method for designing any circuit that means, we have drawn the operation tables then from the operation table using this Karnaugh map; we have tried to find out each of the expression for the output bits.

And then we have realized that for 4 bit it is still manageable, ok. So, beyond 4 bit; that means, if I consider 6 bit or 8 bit or more than that incrementer design so, at that time by following that method that becomes too much complex, ok. So, for that reason what we what we did is that we have try to find out different method, ok. So, as we know that this is nothing, but; that means, one incrementer means that is the addition of 2 numbers where 1 number is fixed to 001. So, from that we have seen; that means, that is for added with 1 so, again we can do; that means, if it is if I want to add it by 2 and if I want to add it by 4 or if I add 1 to increment that is that means, the circuit by 8 also.

(Refer Slide Time: 02:16)



So, at that time what will happen? So, if I just take the example then. So, at that time what will happen? So, in an incrementer design so, for N bit for N bit incrementer design, what was the expression? The expression was something like this N minus, 1 if I just that was A N minus 1 XOR with A N minus 2 A N minus 3 up to A 0, ok. So, this was the, that means, expression for N bit incrementer design ok. So, now, this is only for if I want to add it by 1. So, we have seen that the for 4 bit the S 0 bit was nothing, but A 0 bar and S 1 was A 1 XOR with A 0 and S 2 was A 2 XOR with A 1 and A 0, and S 3 was S 3 was something like A 3 XOR with A 2 A 1 and A 0.

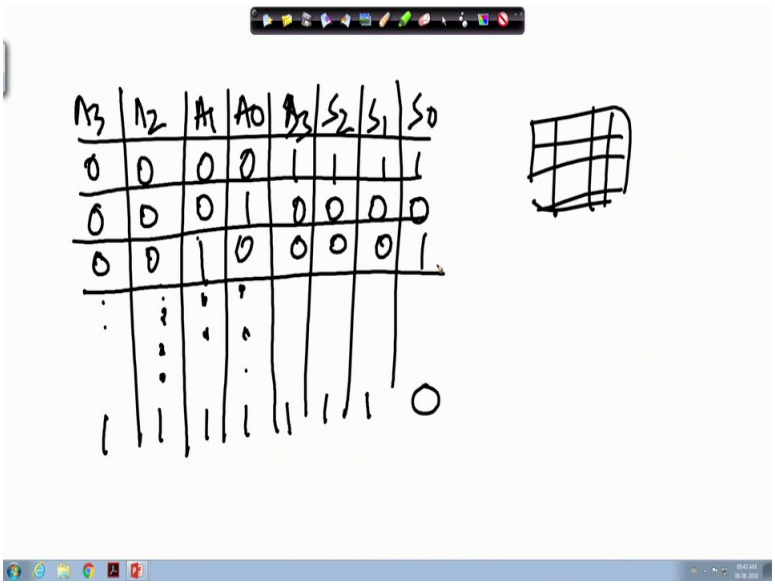
So, this is for if I want to add it by 1. So, if I just want to add it by 2; that means, 0 0 1 0 at that time what will happen; that means, at that time only this particular for 0 0 0 1 whenever this position is basically 1, so, that means, that is effecting the bits which are on the left hand side; that means, from towards the MSB side, ok. So, as I am adding at the LSB position I am adding 0 here; that means, that bit will be remain unchanged and from where we are adding 1, from there it will try to follow this particular equation.

Same things will happen for if I want to add it by 4, so, at that time this 2 bit position is 00. So, this will remain unchanged and the; that means, this corresponding this particular equation will start from this particular bit position, ok. The same thing will happen for A 2; that means, at that time this bit positions will be remain unchanged and the changes

will be occurred from this particular bit position. So; that means, now I can develop or I can design any N bit incrementer design which can increment by 1 by 2 by 4 or by 8, ok.

So, next we will see next we will see another example, ok. So, that is this next we will see another example that is 4 bit decrementer design, ok. 4 bit in incrementer what we are doing in incrementer we are just adding it by 1. In decrementer what we will do we will just decrement it by 1, ok. So, here we will again we will follow the same procedure like here we will try to find out the expression for 4 bit first, then we will try to find out can I apply those expression in terms of; that means, N bit representation, ok; So, that we will try to follow.

(Refer Slide Time: 06:16)

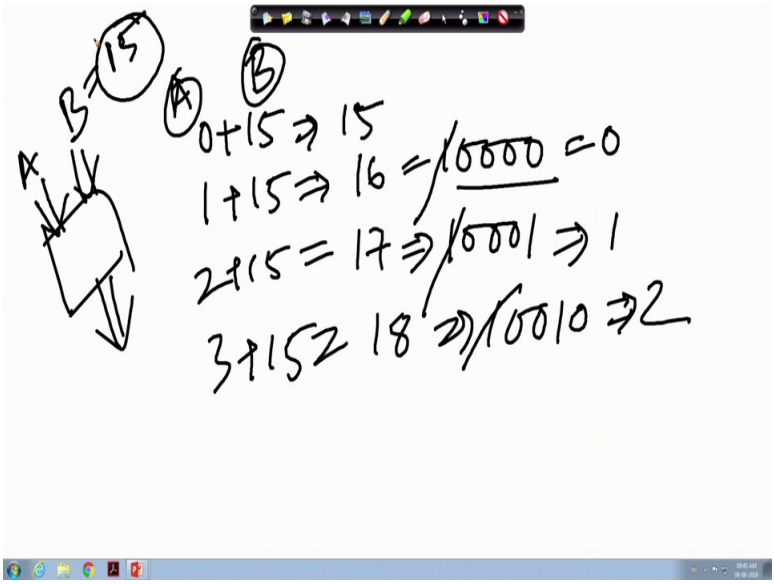


So, in a decrementer design, ok, if I just draw the corresponding so, if I just want to draw it so, in according to the 4 bit decrementer design what will happen? So, here at this particular side this will be the input variables sorry this is S 3, S 2, S 1 and S 0. So, for 0 0 0 0 all will be 1, ok; for 0 0 0 1 all will be 0; then 0 0 1 0 it will be 0 0 0 1. So, in this fashion this will be continued. So, this will be continued so, up to 1 1 1 1. So, whenever I will find 1 1 1 1 at that time I will get it like 1 1 1 0. So, then again using the same procedure using this drawing the Karnaugh map using drawing the Karnaugh map and then try I will try to find out the corresponding expression for that.

So, instead of doing that as I said that what where the problem lies with this technique, so, what we can do, what from these whenever we are doing decrementer design; that

means, whenever we are designing it or calling it as a decrementer at that time what we can write if you just that means, from the that previous things if you see that is 0 0 0 then at that time it is 15; 0 0 0 1 at that time it is 0, for 2 this is 1. So, how we can just do it if I just in incrementer design what we have done 1 bit I fixed to 0 0 0 1. So, in decrementer the same thing if I just add it by 15, as I am considering 4 bit here, so, if I just add it by instead of 1 if I just add it by 15 at that time it will work same as the corresponding decrementer design.

(Refer Slide Time: 08:29)

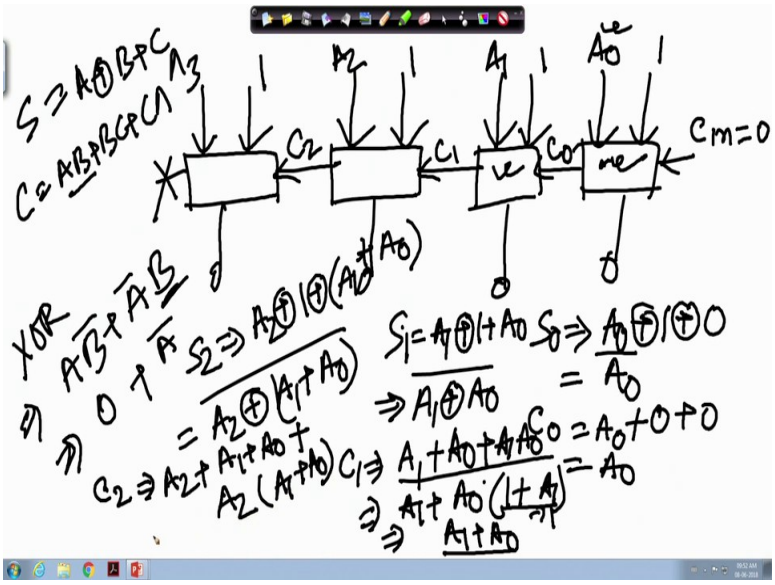


How? So, initially if you consider 0 plus 15 that means, it will give you 15. So, then 1 plus 15 it will give you 16. So, 16 means if I am considering the output of 4 16 means what in binary presentation it is 1 0 0 0 0. So, if I consider only the, these particular 4 bit, I discard this particular MSB. So, at that time this will become 0. So, then 2 plus 15 means equals to 17, so that means, this is 1 0 0 0 1. So, this is this will be discarded. So, this will be 1 then in this fashion for 3 plus 15 18, 18 means 1 0 0 1 0. So, here this is. So, this will be 2.

So; that means, if I just with this is my input A, and this is fixed, this is my input B. If B is fixed to ah; that means, 15 then the corresponding addition; that means, the full adder circuit if I just put it like this my A and this is B if I B instead of in incrementer design what you have done, I have set this B values to 1 and we have got the corresponding expression for incrementer, but here what we have to do we have to add it by 15. So,

then this circuit the same circuit will be that means, the act as a decrementer design. So, whenever we are putting this 15, so, at that time will that be the change in the logical expression, that we will see now, ok. So, how we will see?

(Refer Slide Time: 10:27)



So, for that what we have done in the incrementer design. So, do the same thing we have to do in this case also. So, that means, here if I consider; that means, each of this as that means, 4 bit means 4 full adder cell I require. So, this is A 3 this is A 2 this is A 1 this is A 0 and this is the carry which is been forwarded to the MSB side and this carry is basically cancelled, that means, I am discarding this carry out.

So, this is the carry in so, C in equals to 0. So, and all this B bits they are set to 1 ok. So, now, for S 0 what will be the expression that is what we know that that is for any sum the expression for that that is A XOR B XOR C and for carry that is AB plus BC plus CA, ok. So, here for S 0 what will be the value? A 0 XOR with 1, XOR with 0. So, these means whenever in XOR gate if any of the input is 1, so, at that time if I consider two inputs XOR gate if one of the input is 1 so, at that time it will produce you the inverted of that. Why is so? Because, if you just see the XOR that means thus, expression that is what A B bar plus A bar B, correct; So, if I fix this B as 1, ok, so, at a time what will happen this as this is 0 and here B is 1. So, it will be A bar.

So, that means, if I fix any values any of the input to 1. So, at that time you will get the inverted of that in the XOR gate output, ok. So, that means, now it will be what A 0 bar

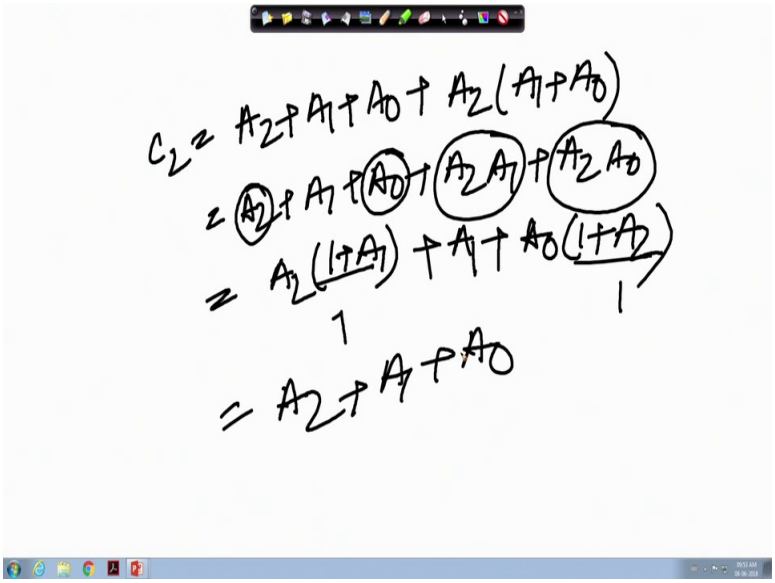
and we have to find out the C 0 and C 0 is what? C 0 is C 1 and this is C 2. Why I need to calculate this C 1 and C 2 because these carry you this bit is 1, ok. So, at that time that carry which is being generated in this particular full adder this particular full adder that will affect the corresponding next of it, the adjacent full adder cell it will the carry will affect. So, that we adjust basically traversing to the, that carry is propagated towards the MSB side, ok.

So, now, for carry we have to also calculate for this particular full adder cell. So, carry will be what A 0; that means, A plus B means A B means here B is one. So, A 0 plus for B and C this is 0. So, for C and A that is 0, ok. So, carry is A 0 in this particular case. So, then next for this particular full adder cell; what will be the expression? So, S 1 is here A 1 XOR with 1, then here carry is what? Carry is not 0 here, carry is basically A 0. So, this is A 0. So, I can write S 0 S 1 as XOR with A 0 bar, ok.

So, then again what will be the carry 1? Carry 1 will be A 1 A B means A 1 plus BC; BC means what? A 0 plus CA; CA means what? here a A 1 and A 0 ok; So, A 1 and A 0 if I just take it common something like this A 0 into 1 plus A 1 what I can write this expression as A 1 plus A 0. Why I can write this A 1 plus A 0, because 1 plus A 1 if you follow this DeMorgan's theorem 1 plus A 1 means that is nothing, but 1, ok. So, that means, A 0 into 1 means that is A 0; that means, that is A 0.

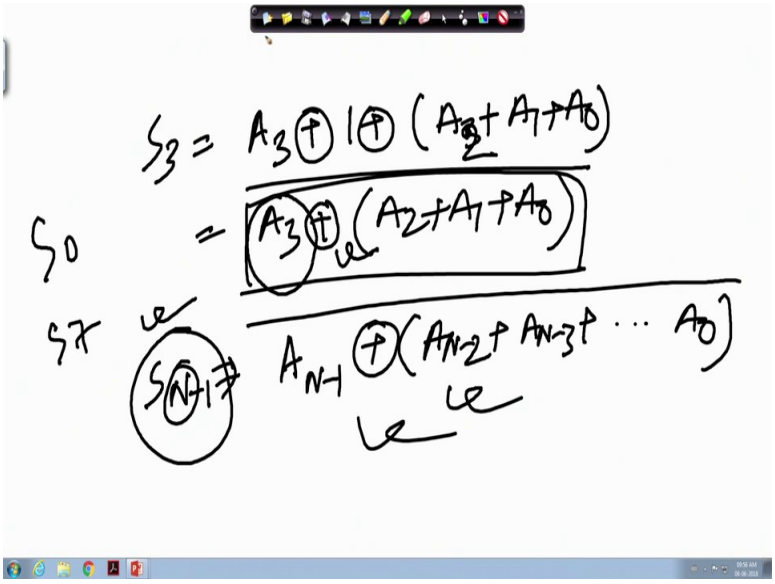
So, this expression I have simplified or I have if I have minimize this particular expression I will get that is as a A 1 plus A 0, ok. So, that means, carry 1 is now A 1 plus A 0. So, what will be the expression for S 2 now the S 2 expression is here A 2 XOR with 1, then XOR with A 1 plus A 0. So, this S 2 expression what I can write A 2 XOR with A 1 plus A 0 whole bar, ok. So, then again if I try to find out the corresponding carry out expression. So, carry out expression now what we will be; that means, for AB that is A 2 plus for BC that is A 1 plus A 0 plus what that is for AC that is A 2 into A 1 plus A 0, ok.

(Refer Slide Time: 16:53)



So, then the same thing using this that means, if I just consider a new page, if I just take this C 2 as what A 2 plus A 1 plus A 0 plus A 2 into A 1 plus A 0. So, if I just want to simplify this particular expression then what will happen A 2 plus A 1 plus A 0 plus A 2 into A 1 plus A 2 into A 0. So, then again if take it common as A 2 common; that means, within this and this if I take commons. So, A 2 plus A2 into A 1 plus if I take common of this A 0 and this; so, then at the time what will happen? Here A 1 is remain same. So, A 0 is common 1 plus A 2. So, 1 plus A 1 is nothing, but 1 and 1 plus A 2 is nothing, but 1. So, that means now what I can write this expression as, so that means, now C 2 becomes A 2 plus A 1 plus A 0, ok.

(Refer Slide Time: 18:03)



So, now next again then for S 3 what will happen then, for S 3 what will happen the bit is A 3 then with 1 and then the C 2 value is a sorry this is A 2 plus A 1 plus A 0. So, this means this is A 3 plus A 2 plus A 1 plus A 0 whole bar because of this 1. Then, that means, now we have find out the expression for what that 4 S value the 4 output value. How? Here this is the expression for this S 3, ok. If I just go back this is the expression for S 2, this is the expression for S 1, this is the expression for S 0. So, that means, now I have find out S 0, S 1, S 2 and S 4 by this particular technique.

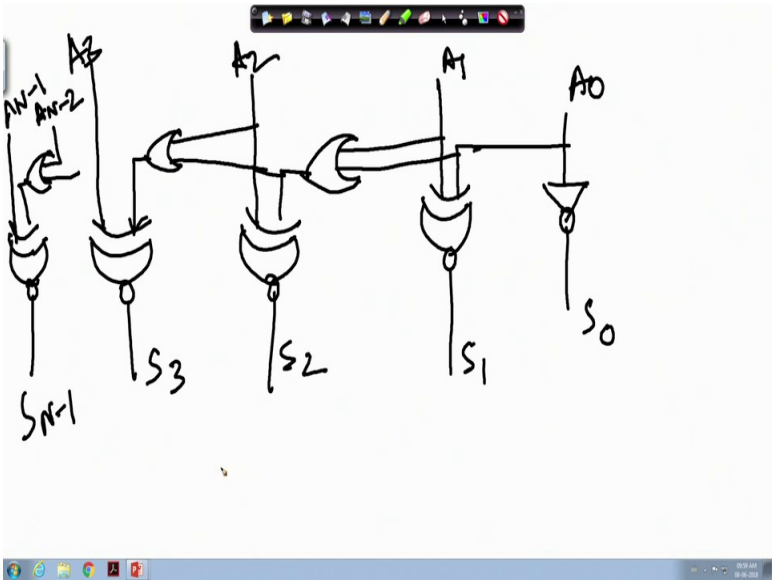
Now, so, from this can I generalize this particular equation? If you follow that whenever we are just; that means, traversing to the MSB side the corresponding bit is, sorry this is XOR, that is XOR with the lower side that means, the next corresponding bit which are OR to each other and then inversion of that. That means, from this particular equation if I want to make it generalizing generalize form so, at that time what will happen or what I can write that is S N equals to sorry S N minus 1 that will be A N minus 1 XOR with A N minus 2 plus A N minus 3 plus up to up to A 0, this is or and whole bar of it. So, this is the generic expression for any N bit decrementer.

So, now you can put this N value to any value; that means, whether this 4, 6, 8, 16 whatever value, you just put it and get each of the corresponding; that means, sum value that is if I consider 8, so, sum 0, 2 sum 7 you just find out and in this in this using this

particular equation you just find out the expression for each of that and then you just map it to the architecture, ok.

So, now if I just want to that means, this 4 bit this expression we have got now if I got just want to map it to the architecture; that means, whenever you have got this particular expression now you can write this expression using any HDL language. So, whenever you do the synthesis at that time that will be converted into the gate corresponding gate level net list; that means, that will be implemented using the corresponding gates.

(Refer Slide Time: 21:39)



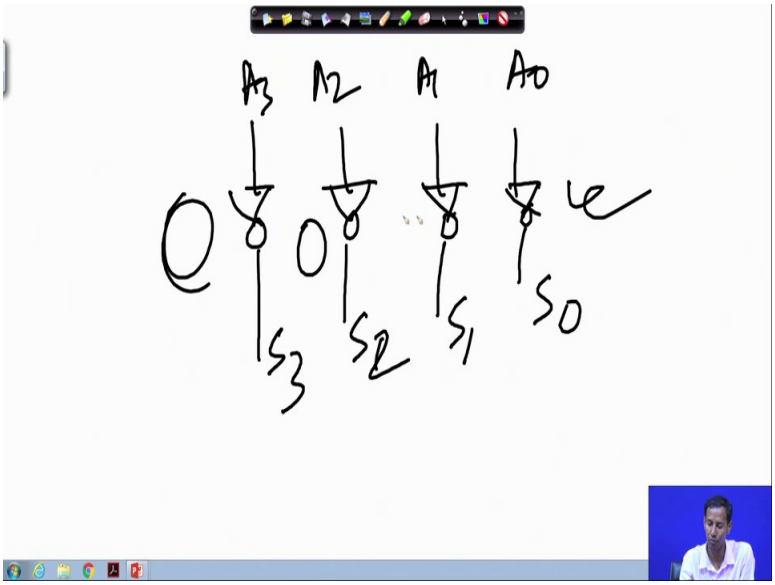
So, now if I just; that means, draw the corresponding that architecture for that so, at that time how it will look like? So, it will look like for S 0 what is there for this is A 0 and this is S 0 that is the inversion of that. So, in the next that is A 1 XNOR with A 0 that is S 1 ok. So, then what is there? I need OR gate, in incrementer I was needing AND gate, but here I need OR gate. So, that means, OR gate that has to be this is A 2 which is OR with A 1 and A 0 and; that means, sorry XNOR with A 1 plus A 0 then again for S 3 this is S 3, and here also I need one OR gate. So, this is one of the input is this one and another input is this one. So, here what I am doing A 2 plus A 1 plus A 0 which are XNOR with A 3 and this will give me S 3 value, this will give me S 2 value.

So, if I just want to make it for N bit. So, N bit means here what will happen then this will be this. So, this will be connected to A N minus 2 and this will come from the previous OR gate output and then it will be XNOR with the corresponding A N minus 1.

So, this this will give you S N minus 1 value, ok. So, this is the; that means, architecture for the 4 bit decrementer ok, understood?

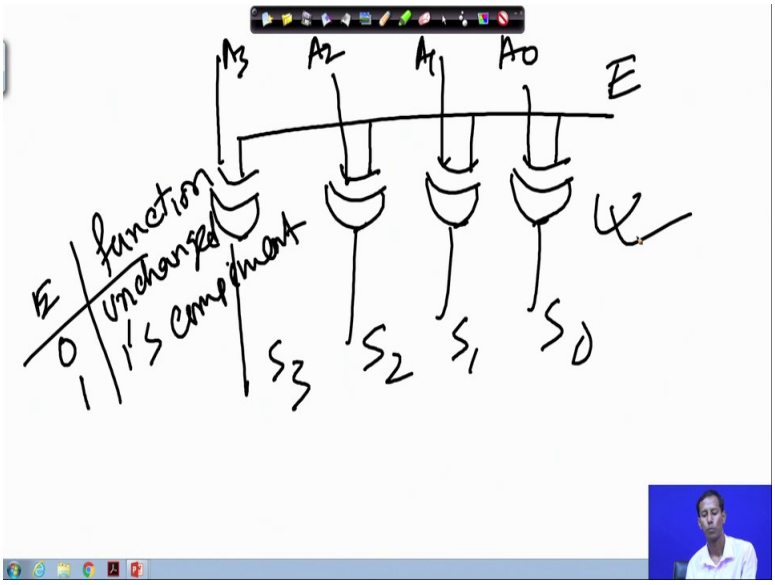
Then, again if I just want to just go back to the slide so; that means, this now in this; that means, we have learnt how to design any that means, any N bit decrementer design. So, now, next what we have to do we have to see 1's complement circuit design, ok. So, 1's complement; in 1's complement what we do? In 1's complement we do basically suppose whatever input we gave it will be just inversion of that.

(Refer Slide Time: 24:40)



That means, draw it that means in 1's complement circuit if I consider 4 bit so, at that time what will happen? So, this is A 3 A 2 A 1 A 0, this is S 3, S 1 sorry S 2, S 1 and S 0 so, this there nothing, but 1's complement circuit. But, I need controlled operation of that; that means, if I put enable signal then only it will give me this 1's complement circuit otherwise it will remain unchanged. So, can I for that what will be the changes in these, what will be the changes in these particular circuits? So, in the in the just previous case what we have seen in XOR gate if one of the input is 1 so, at that time the next the output of that particular XOR gate XOR gate is the inversion of the other input.

(Refer Slide Time: 26:00)



So, instead of this inverter over here if I just put it the XOR gate for each of this particular instead of this inverters, sorry I am designing XNOR and here each of this input is like this is A 3 this is A 2 this is A 1 and this is A 0 and this is connected with this is S 3, this is S 2, this is S 1 and this is S 0 and this is the enable signal. So, according to the logic what will happen if enables equals to 0, if enable equals to 0 at that time the it is unchanged ; that means, this is the functionality and whenever it is 1, so, at that time it will give me 1's complement, ok. So, this is the circuit where I can do this controlled 1's complement circuit.

So, if now if I want to make 2's complement; So, how we will do in 2's complement initially how we can do initially is that we do 1's complement then with that 1's complement if you add 1, then that will become 2's complement. So, that means, there is an option where or by which I can do in one particular circuit I can implement thus the both the things; that means, 1's complement as well as 2's complement; So, that we will discuss in the next class.

Thank you for today's class.