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

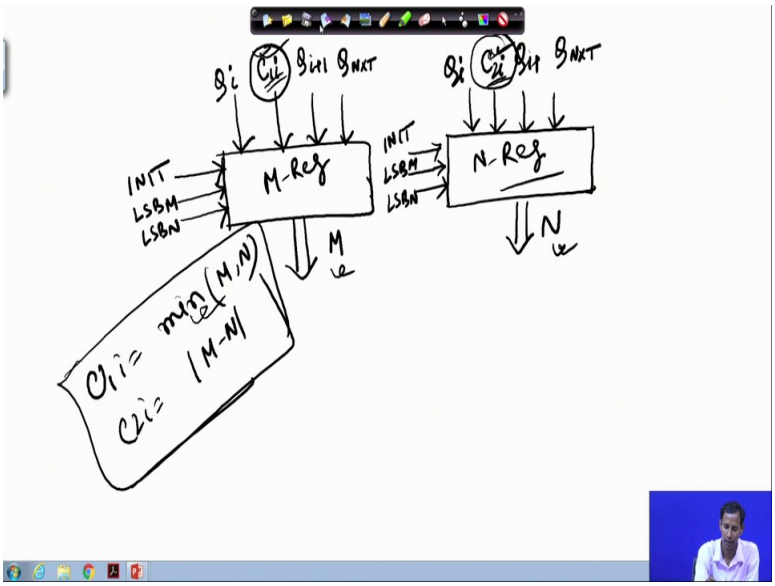
Lecture – 11
Algorithm to Efficient Architecture Mapping (Contd.)

Welcome back, to the course on Architectural Design of ICs. So, here we are continuing with the efficient architecture design from the algorithm different algorithms; how efficiently we can implement the architecture, different architecture, there are different styles of implementation; that means, the architecture will be different if we are considering for this speed efficient or power as a major constraint or a area as a major constraint.

So, the architectural will different for the same algorithm or for the same function; I will have or we will have different architecture for different different perspective, ok. So, that we are seeing in the last few classes and we will continue more to onto the same flow.

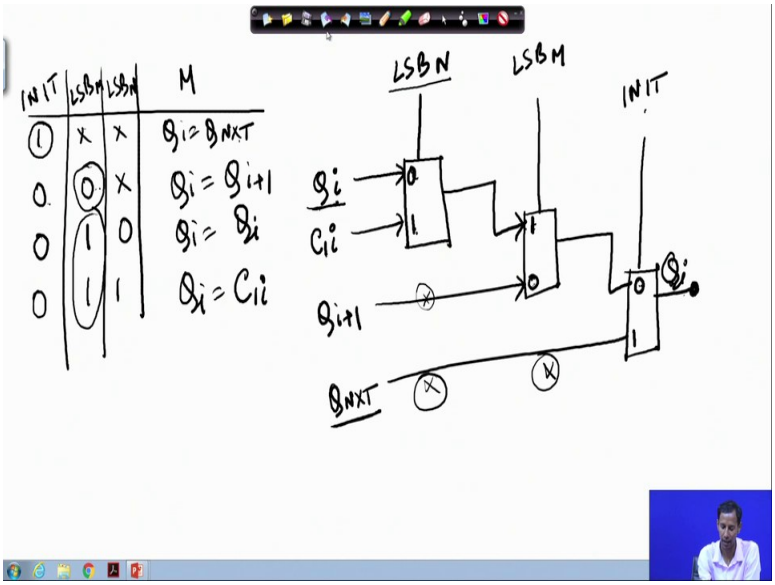
So, in the last class what we are discussing we are discussing about the circuit which can implement the GCD of two numbers, ok. So, we have built the circuit; the basic components of the circuit we have already built. So, now we will try to find out the final architecture which will give me the GCD of two numbers, ok.

(Refer Slide Time: 01:44)



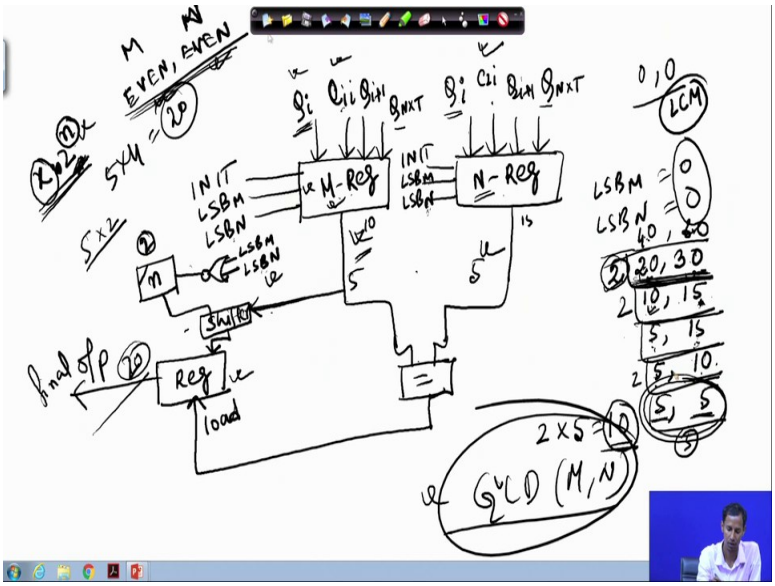
So, then again come back to the; that means, so, here in the last few class; that means, last class what we are seeing that we have built this M-registers and N-registers.

(Refer Slide Time: 01:52)



And, then how we can build this M and N that also we have seen and.

(Refer Slide Time: 02:15)



Now, next is that again if we that means, we have we have already drawn or we have already defined the corresponding this was the M-registers and this was the N-registers, right.

So, if this is a Q i this is C 1i this is Q i plus 1 and this is Q next this is also Q i this is C 2i this is Q i plus 1 this is Q next and this is for M value and this is for N value; that means, here I am mentioning Q i. Q i means that is the value for Q M that is not value for; that means, this Q i and this Q i is not same. This is the that means, same input which will connected to the output for N values and this is the same output and input will be connected for M values, ok.

So, that means, this Q i and this Q i or this Q next and this Q next is not same this is for this is for M and this is N, and what was the that means, control signal? The control signal was INIT LSB of M and LSB of N. So, here also the same thing INIT LSB of M and LSB of N. So, we have already in the last class we have already developed this circuit for this C 1 and C 2 that also we have already developed.

Now, we have to build the final architecture. So, I need this two. Basically, what was the logic? That means, whenever we are finding out the GCD using this particular algorithm. So, at that time the final results how it will be it will be x into 2 to the power N. So, where n is what whenever we find out this even, even, even positions for M and N, sorry M and N. So, at that time that has to store somewhere and that information will come and whenever this; that means, what is the terminating condition for this; that means, whenever M and N value is same at that time this loop will terminate.

So, that means, I need one this comparator block equals to comparator block which says that this a M and N both are same. So, now computation will be over, ok. So, that means, I need one register where this will indicate, ok. So, this is an output final output. So, this will indicate this registers that you produce the results at the output bit, but at that time whatever is the value of whenever this is M and N, so, at the time that is the end point. So, now, that value with that value I have to multiply with this; that means, how many times I find out this even, even position.

So, even even position how can I determine that these are the even even position; that means, when LSB of M and LSB of a N both are 0, 0. So, I need one NOR gate where this is LSB of M and this is LSB of N which will give me that ok, this is the position where I am finding out this both as 0. So, this will be connected to where? This will be connected to a counter; so, that means, whenever how many times I am finding this

during this iterations how many times I am finding this even even that will be recorded via this LSB M and LSB of N.

So, LSB of M and LSB of N means that we will explain with the example later. So, here basically this is one N registers, ok. So, now, that will be connected with what that will be connected to a shifter and this shifter value is coming with this and this shifter output will be connected to this; that means, this is the load. So, what is output of this, why I need the shifter? I need to implement this particular function that x into 2 to the power of N.

So, that means, this is x this is the; that means, terminating values and that is to multiplied with how many times I am finding this even, even position ok. So, now, if I just want to explain with the corresponding x; that means, example. So, if I consider this 20 and 30 case. So, what was the expression? So, 2, here it was 10 and then it was 15, then it was 2, it was 5 and it was 15, then it was 5 and then it was 10 then again I have to; that means, in the next it was 5 and it was 5. So, both are same. So, this is 5.

So; that means, here only for this particular position I am getting this both are even. So, that means, 2 into this is the that means, terminating condition so, 2 into 5. So, the GCD of this 20 and 30, that it has to be 10. So, now, you check whether this circuit is giving me the same results or not. So, in the in this particular case this 20 and 30 is loaded over here, ok.

So, based on that this LSBM and LSBN, where for this two, it is 0 and 0 ok. So, now, then you see this is 1. So, this has incremented by 1. So, that means, now this counter value has been incremented by one then in the next it is this next value is again comes over here. So, the newer value of M and N this is M value is ah; that means, even and N value is odd. So, from here based on this will be now the corresponding value will be loaded to this, that means, initially the value was 20 and 30 we have stored this then this value is 10 and this value is 15.

So, it will check whether that 10 and 15 both are same? No, ok. So, that means, this signal is 0 at that time so; that means, the register will not produce at the output. So, then what will happen, this particular case. So, whether this LSBM and LSBN is getting the newer; that means, incremented by once it found that this is not 0 0; that means, this is 0, this is 1. So, that means, this operation is not at all performed at that cycle.

So, again then what will happen, 10 and 15 case for this 10 and 15 case for this 0 and 1 for 0 and 1 here what will happen it will try to select it will do this M value by 5 and N will remain same. So, then again if finds that in the next cycle what will happen it finds that it is 1 and 1; that means, both are odd. So, for both are odd this will not change. Is 5 and 15 is same? No, that means, this particular operation will not performed. Only at that time what will happen again as this is 1 and 1 case. So, this will choose this C 1i and this will choose C 2i. So, C 1i and C 2i what will happen this will choose minimum of this, which is 15 and 5 which is minimum of this is 5 and this is mod M minus N, that is 10, ok.

So, then again in the next cycle what will happen again this will this is even this is odd this is even, right. So, whenever this is even; that means, it will be divided by 2 and M will remains the same. So, that means, M will remains same and this will be divided by 2 means this will be 5. So, and again as this is not even, even. So, this will not even and 5 and 10 is not same. So, again this power this will not happen.

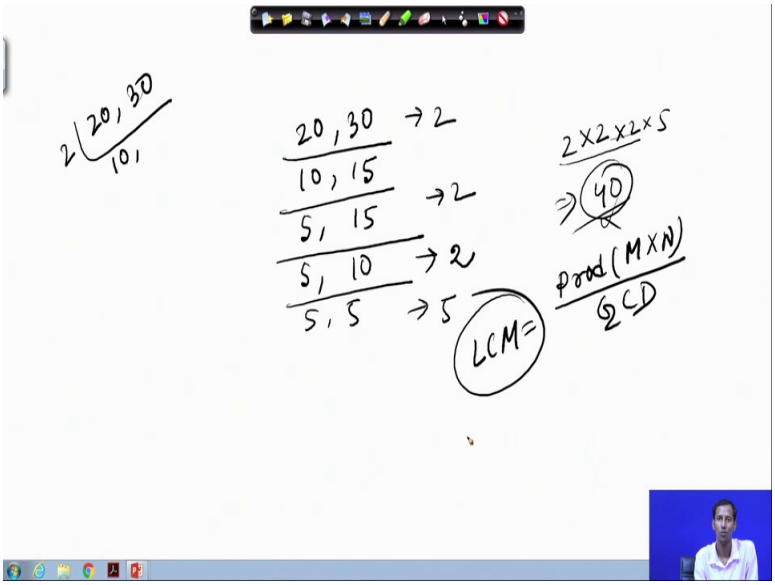
So, once I reach to this 5 and 5. So, at that time what will happen 5 and 5 means this com this is 1, ok. So, 1 and at that time this is 5, this is 5; that means, at this particular position this is 5, this is 5. So, 5 into 2 here only 1, this counter has been increased by only one times. So, that means, from here it will 5 which will be multiplied with 2, that means, then at that time it will be this results will be 10 and it will produce the output as 10. The same thing will happen if I consider that as 40 and 60, at that time what will happen? Two times I will find even, even case for 40 and 60, it will be even, even. For 20 and 30 it will be even, even. So, that means, for two times it at that time this counter value this will be 2. So, that means, 5 into 4; that means, it will be 20.

So, that means, the GCD of this 40 and 60, that is 20 and I will get the 20 at the time as the output, ok. So, this is the circuit which gives me the GCD of two numbers, ok. So, now, right from this particular circuit ok, we know that this is the GCD means the greatest common divisor among two numbers if you just see the LCM which is the lowest common multiplication factor.

So, can I that means, try to say that the that number also using the this particular circuit can I implement if I have tried can I that means, the logic if I find out can I do that. So,

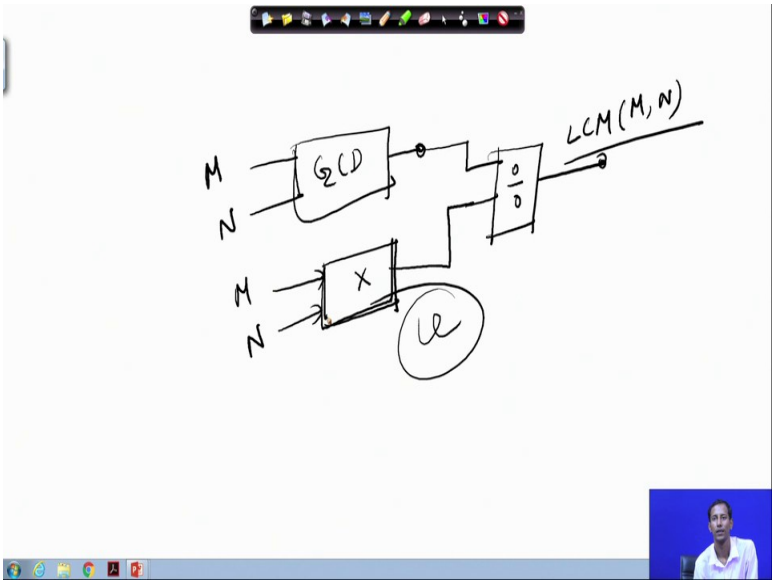
here if you see in each of this function whenever I am getting this the LCM of this if I just go to this then 20 and 30.

(Refer Slide Time: 14:17)



If I consider so, initially what are that is and 10 and sorry. So, 20 and 30, that means, here it is 10 it is 15. So, that means, here I am doing 2. So, then this is 5 and this is 15. So, here I am going to then this is 5 and 10; that means, here if 2 divide by 2. So, that means, 5 and 5. So, here from here if I get 5, so; that means, 2 into 2 into 2 into 5, what is that? 5 into 8 that is 40. So, using this particular circuit I cannot implement the LCM of that. If I want to calculate the LCM of this at that time I have to follow or I have to design one another algorithm to find out the LCM and then how can I do because LCM means what LCM means what the product of M and N divide by the GCD. So; that means, how can I find out the LCM from this particular circuit ?

(Refer Slide Time: 15:53)



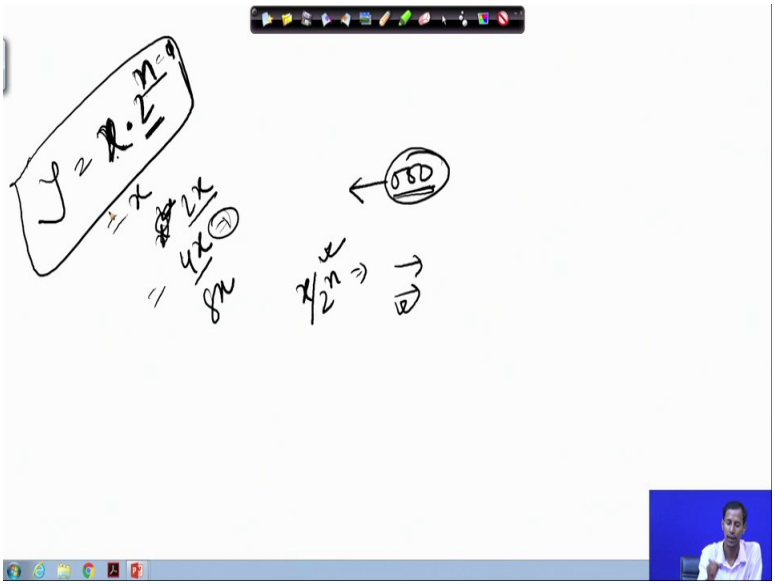
I am having this GCD circuit, right, where I am putting this M and N I need another particular multiplication operation if I produce one divider circuit over here which gives me this M and N divide by this GCD. So, at that time I can find out the LCM of M and N ok, understood. That means now I need from this particular GCD function I can; that means, this I have to start with this GCD then I can find out the LCM too by using this particular logic.

So, this is the; that means, this is we have to start from the basic then from the basic now we can try to keep on building the cell, ok. So, this is the; that means, the implementation of GCD function and whenever you are following this course at that time please actually let me know if you are having that means, any of the circuit; that means, or for this particular system. So, it is your thought how you basically how do you think that if I implement in this way so, it I can get actually it depends upon your requirement; that means, whether you are choosing speed as a constraint or power as a constraint or area as a constraint, right. It is not at all through that whatever circuit or actually here I am showing you the different implementation style for different application, ok.

So, whenever you are following this course at that time if you find me if you are having any problem related to your research or your; that means, your design any type of circuit doubts if you are having at that time please let me know in the forum ok. So, the you just

post that questions on the forum. Obviously, I will try my level best to solve that problem, ok. So, then again continue with the course.

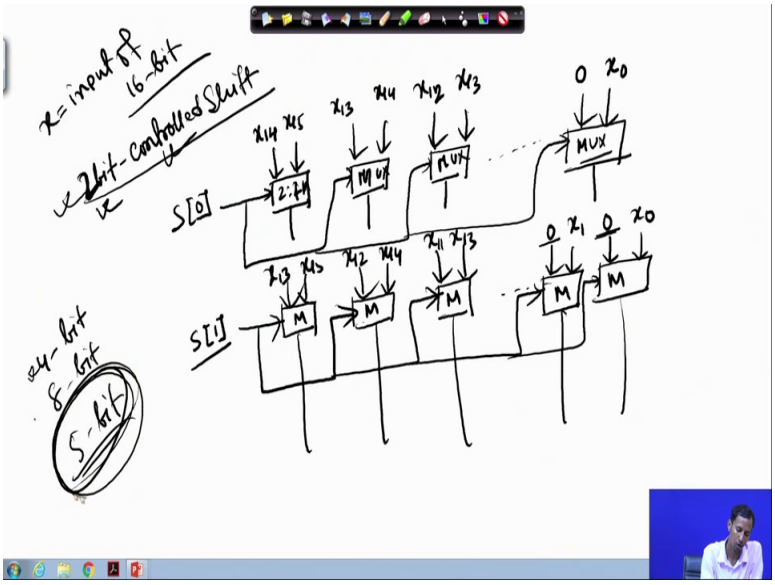
(Refer Slide Time: 18:41)



The next circuit which I want to implement that is let us consider I want a function which will give me this, ok. So, what I said in the that means, we have to implement this kind of function right, x into 2 to the power of N. So, if I want to ah; that means, implement this kind of architecture that means, I want a function which will give me this x into 2 to the power of N. So, what will be the circuit for this. So, then, what I need, what I need? Basically this 2 to the N means what that is the multiplication operation of x; that means, how if N value is 0; that means, 1. So, if it is if it is x if N will equals to 1. So, that means, this is x square sorry 2 x.

If it is 2, so, that means, at that time it is 4x so, that means, 2x, 4x, 8x means what I am just putting or adding zeros at the end of this x values. Whenever we are dividing so, x if I just; that means, divide by 2 to the power n, that means, I am discarding the LSB position and whenever we are multiplying with 2 to the power n at that time means it means what I am adding zeros at the LSB position to increase the as this is binary weighted values. So, that is why we are increasing the value by putting extra zeros to the LSB position, ok.

(Refer Slide Time: 20:49)



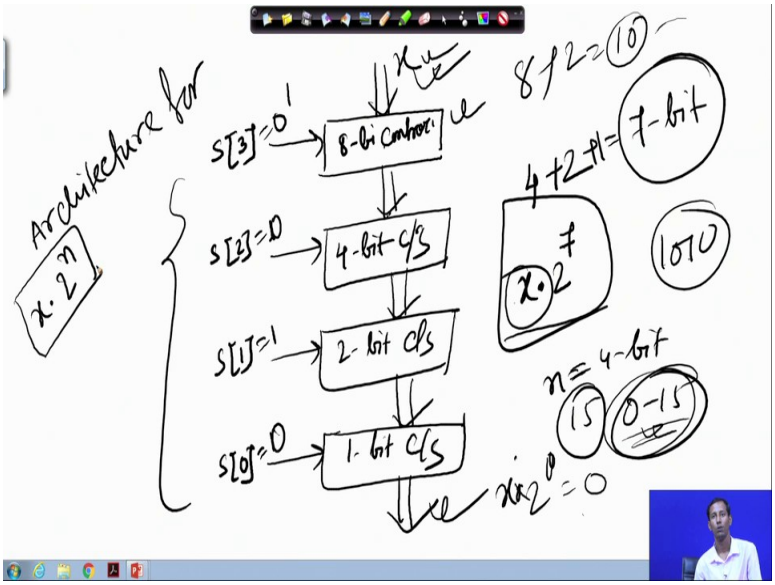
So, for this particular function for implementing what I require is that I require 1 bit controlled shift operation. So, 1 bit-controlled shift operation means what I need 2 is to 1 multiplexers. So, if I consider suppose this x input x is the input of 16 bit, let us consider that and I need 1 bit-controlled shift of that. So, at the time what will be, so, that means, here it will be up to x to the power 14 x to the power 15, x 15, x 13, x 14, x 12, x 13 here it will be 0 and it will be x 0, ok.

So, now this will each of this will be connected something like this is the select line or S 0 if I select. So, this is that means, this is 2 is to 1 2 is to 1 MUX. So, this is the MUX ok. So, this is 1 bit-controlled shift operation. If I wanted 2 bit, 2 bit so, at that time only the bit position will change, right. So, how the bit position will change? So, if I considered 2 at this particular stage. So, if I consider this as a select line at that time each of this is MUX, 2 is to 1 MUX. So, at the time what will happen this will be x 13, x 15 this will be x 12, x 14, this will be x 11, x 13 this will be 0 x 1, 0, x 0, and this is for x 1 ok. So, that means, whenever I need 2 bits at that time this two zeros will be padded to the LSB position.

So, the same thing if I need 4 bit-controlled shift operation. So, at that time 4 zeros will be added for 8 bit operation I need this kind of; that means, a up to; that means, 8 LSB position will be padded with 0. So, now, if I that means, if I just this is for 1 bit, 2 bit, sorry 1 bit, 2 bit, 4 bit and 8 bit if I need let us consider 5 bit. So, at that time how can I

do? I can if I just implement it in a structured way. So, at that time I can do it any values of this controlled shift operation how can do that.

(Refer Slide Time: 24:38)



So, if I just added something like this if this is my x. So, this is my x, this is let us say S 3, this is S 2, this is S 1, and this is S 0. So, this is let us consider 8 bit controlled shift, this is 4 bit controlled shift, this is 2 bit controlled shift, this is 1 bit control controlled shift.

So, if I if you just put these value to that means, if you just change this corresponding bit values to the select line then you will get that number of that means, shift operation of the x input. How? If you just select this as 1, this as 1 and this as 1 and this as 0. So, at that time how many bit how many shift you require you will get in terms of this x this time as this is 0; that means, there will be no shift at this part at this particular stage.

So, x bit will be coming to coming over here. So, here it is shifted by 4; that means, initially this x is shifted by 4, then again that value will be shifted by 1 means it will be shifted by 2, then again that value will come here and that will be shifted by 1. So, that means, total 7 bit shift I will get of x or 7 bit x shift means that is x into 2 to the power of 7 ok.

So, if I need, so that means, n bit n value is if it is of 4 bit, so, that means, total 15 from 0 to 15 I can choose any value or of shifting of x. So, by this structured manner I can shift

within this any values of this within the range of 0 to 15. So, if I put; that means, 10. So, at that time I have to select that as 0 here 1, 0. So, that means, here it will be shifted by 8 and here it will be sorry this is 8 plus 2, so, that means, this is 2. So, here I will get 8 here I will get 2. So, total 10 bit shift means a x into 2 to the power 10, I will get as the output of. So, this is the structured for this is the architecture for x into 2 to the power n.

So, thank you for today's class. Again, we will see other circuit implementation technique in the next class.

Thank you.