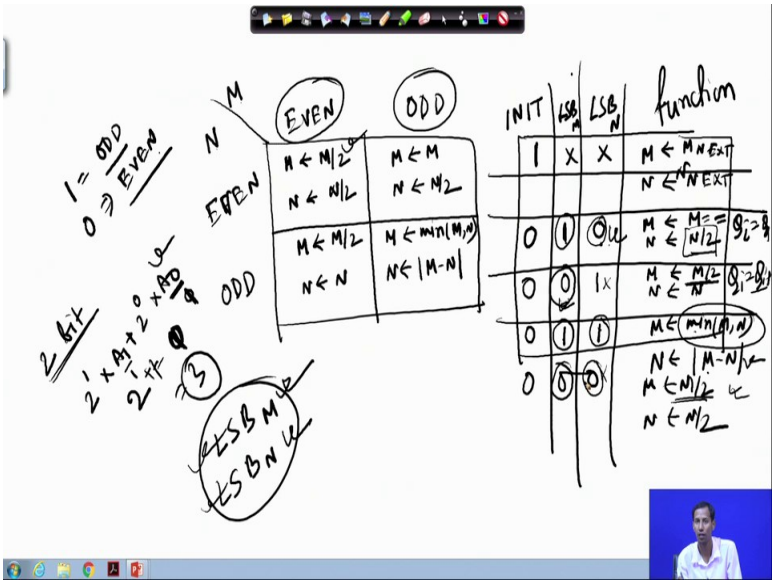
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture – 10
Algorithm to Efficient Architecture Mapping (Cont.)

Welcome back to the course on Architectural Design of IC's. So, in the last class we have seen we are basically trying to find out one efficient architecture for computing the GCD function of 2 numbers ok. So, we have seen that based on this modified Euler's algorithm, we can find out the GCD function and there are another method also find out the GCD function. But here, we will try to find out the GCD function using modified Euler's algorithm and then how efficiently we can these design the circuit in order to so that I can get the benefit in terms of power as well as speed as well as the area of consumption that we have seen in the last class.

And we will just continue from that. So, what we have done? We have done the that means, operation table; then from operation table we are just trying to find out the expression or what is the that means, what will be the architecture for that or what will be the logical expression to calculate the GCD of 2 numbers ok. Let us consider continue with the that means, previous class. So, in the last class where was I? I was drawing this particular function right.

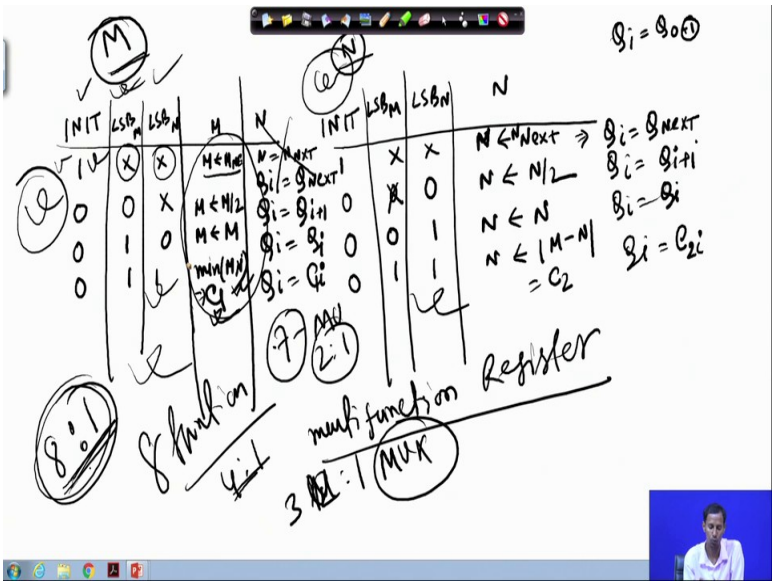
(Refer Slide Time: 01:39)



So, that means this particular function means the functionality, how it works ok. So, based on the LSB of M and LSB of N, I can find out whether the numbers are even or odd and then, based on these particular things I can the, what is a function I need to implement so that we have just get. So, here the interesting things you just see whenever this value is 0, whatever value is for the LSB the of N; that means, whether they even or odd, it does not matter. Whenever the value of M is 0; so at that time, it does the same thing; that means, it does the M by 2 and here also N by 2. But for 1 for odd case it has different functionalities; that means, for N whenever this is N that means, this LSB of it depends on the changes in the LSB of N.

So, when this is odd and this is N; so at that time M equals to M; but when this 1 and LSB of N equals to also 1. So, at that time this is not M this is another else that means, M equals to minimum of M and N and the same things happens for LSB of N 2 that means, you if you see when the whenever this is 0 and this is 0, it does the same thing that means, it is N by 2. So, whenever this is 1 at that time this changes that means, from 1 of M this is minimum of mod of M minus; for 0 that is, so that means now if I just that means, try to reduce or try to rewrite this particular table. How can I rewrite? How I can re-write this same table?

(Refer Slide Time: 03:42)



As something like this. This is LSB of M; this is LSB of N and this is the corresponding sorry M and N. So, if this is 1 whatever is the value M equals to M next and N N equals

to N next ok. This is 0 and whatever value is for this for LSB equals to M M equals to M by 2 ok. I am not putting this value and this is for sorry you do not consider this N. Now, this is only I am calculating for M. So, this is for 0; so at that time if this is 1; this is 0. So, at that time M equals to what? This is M for 0 1 and 1 this is minimum of M and N ok. The same table if I just draw for N, this is for M. Now this is I am drawing for N. So, this is LSB of M this is LSB of N and this is the N. So, for 1 cross cross, this is M N equals to N next for 0 0 sorry what LSB cross for N it is I am considering.

So, N equals to N by 2, for this is 0 for whatever value is for 0 and 1 N equals to N for 0 1 and 1 N equals to mod of M minus N ok. So, this is the that means, corresponding functions which I require for M and N. So, that means how can I implement this particular logic now? How can I implement or what circuit I required to implement this particular logic? So, that means I need 1 this multifunction register which can gives me this particular circuit. So, what I said that M equals to M next means, so Qi that will be loaded as Q NEXT ok. So, M by 2 means Qi will be loaded to Qi plus 1; that means, divide by 2.

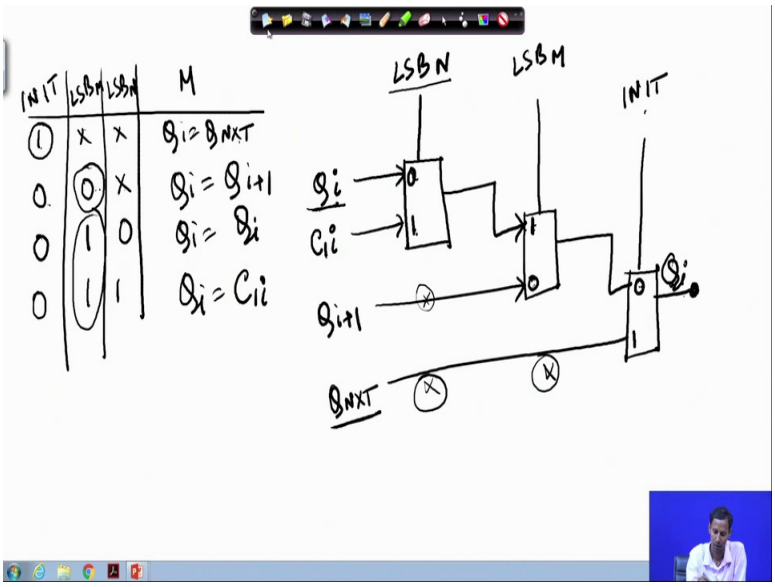
So that means, Qi plus 1 means, so at that time Qi if this is that is Q 0, i value is 0. So, instead of that that will be loaded the 1 ok. Why? Because I have to divide by 2 means I have to discard the LSB position. So, 1 position at that time I have to load right. So, that is why that is Qi plus 1. So, whenever M equals to M means so that is Qi equals to Qi and this is minimum of N if I consider this as a circuit C 1. So, that means Qi we will C i. The same thing will happen for this, here also Qi equals to Q NEXT. This is for N register right. So, this is Qi equals to Qi plus 1 sorry this is i this is 1. So, Qi equals to Qi here, if I consider this as a C 2. So, at that time Qi we will be C 2 i. So, this is C 1 i; this is C 2 i ok. So, now, I have to implement this particular logic using this, I have to implement using the hardware. So, what will be the hardware for this ok?

So, you see there are 3 control signal; but it is not that each of the corresponding that means the value of this. So, 3 control signal means I have to I need to implement for. Here I am having how many operations? I am having only 4 operations; but if I have considering 3 in as that means, this control signal. So, at that time up to 8 functions, I can implement.

But here I don't have the as this is the do not care option that means, if only this has the highest priority in this particular case so that means at that time the other two other these values I it is do not use or it is not at all used for any other function; where the function will change. So, that means if I need if I am having this 3 input control signal. So, at that time I need 8 is to 1 multiplexer. So, but I am having only 4 function to implement. So, then why to implement or why to go for this is 8 is to 1 ok? So, only I can have this 4 is to not 4 is to 1.

So; that means, I need 1 multiplexer; where I will use this 3 control signal, but I have I will implement only this particular function. So, 4 of this functions means only. So, 8 is to 1 multiplexer means what I need 7 multiplexer 7 2 is to 1 multiplexer for implementing this 8 is to 1 multiplexer right. But here, I am I need to implement only 4 functions. So that means, I using only 3 multiplexers 2 is to 1 multiplexer, I can implement this function. So, how can I do that? If I just go back and again if I just I will just draw the corresponding.

(Refer Slide Time: 10:47)



This is INIT this is LSB of M; this is LSB of N and this is M. So, for 1 cross 1, Qi equals Q NEXT for 0 0 do not care Qi equals to Qi plus 1 for 0 1 0 Qi equals Qi and for 0 1 1 Qi equals to C 1 i. So, now if you just see that means, it has the highest priority after that this 0 has the highest priority and after that it basically checks the corresponding bit position of LSB M. So, now, how can I implement the function? So, if I am having the 1

2 is to 1 multiplexer considering this; so that means, now this is for LSB N for 0 and for 1 this is 1.

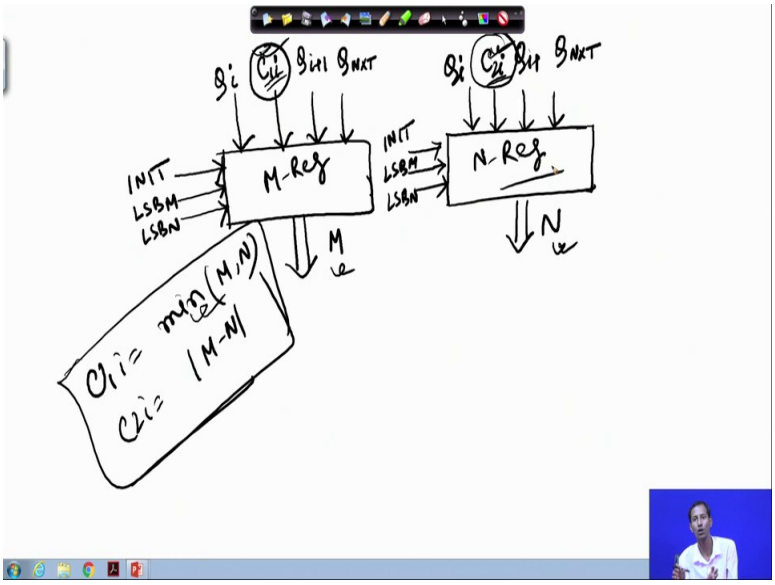
So, for LSB N equals to 0; so, what happens? This is Qi and this is C 1 i; then, this will come to what this will come to another multiplexer, where the other input is this will depend on what? This LSB of N. The another input of this is Qi plus 1 and then, if I just have another multiplexer 2 is to 1 multiplexer where this signal is the INIT signal and this is Q NEXT that you see this is for this is for 1; this is for 0 and this is for 1; this is for 0.

So, here you see how we are implementing this particular table in using this particular circuit. So, you see whenever LSB N equals to 0 or 1 and M equals to LSB M equals to 1. So, at that time INIT value equals to 0. So, if you follow from this. So, 0 1 0 Qi equals to Q i. So, Qi will be this. So, Qi means this is nothing but this Qi what I am considering. So, Qi equals to Qi, for 0 1 1 Qi equals to C 1 i. So, Qi equals to C 1 i ok.

So then, for this particular case for 0 0 x for 0 0 x means if this bit is 0 0. So, 0 0 and then, here it is do not care. So that means, whatever combination is for LSB; I do not have to check for the bit position of LSB N. So, that is why this multiplexer is I have omitted that. So, for 0 0, this Qi equals to Qi plus 1. So, that will come here for initial. So, if this is 1; then this will be Q NEXT. So, this is 1, it will choose directly Q NEXT it do not have to check for the values of LSB M and LSB N.

So, that means now this corresponding function can be implemented by using this kind of circuit. So, though I am having 8 sorry 3 control signal. But here, I require only 3, 2 is to 1 multiplexer instead of 7, 2 is to 1 multiplexer which I required in general as the number of functions are smaller ok.

(Refer Slide Time: 15:29)

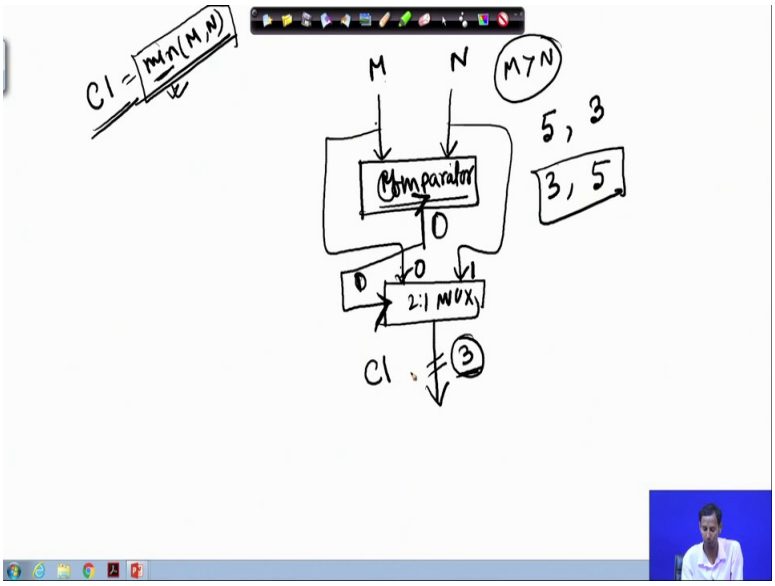


So, if I just draw the block diagram of that. So, at that time I can I will have this 4 of this input and I will have this 3 as the control signal which will produce the corresponding M and the same thing for N also it will be the same only this input values will be the that means, different this is for N. So, so this is INIT, this is LSB of M this is LSB of N. So, here what is that the first is Qi. Sorry, yeah Qi. Then this is C 1 i and Qi plus 1 and then Q NEXT ok.

If I just start from this so that means, the corresponding bit position you have to select sorry at that time what will happen? This will be whatever it may. So, here again it will be Qi; this will be C 2 i; this will be Qi plus 1 and this will be Q NEXT ok. So, this is INIT LSB M this is LSB N ok.

So, this will produce M and this will produce N. So, I have just done that means, implemented the corresponding operation table using this. Now what I left is that I need 2 circuit for C 1 i and for C 2 i where the, what is C 1 i? That is minimum of M and N and C 2 i is mod of M minus N; that means, I have the circuit for implementing which you imp which will implement this two particular circuit I need 2 circuits.

(Refer Slide Time: 18:11)



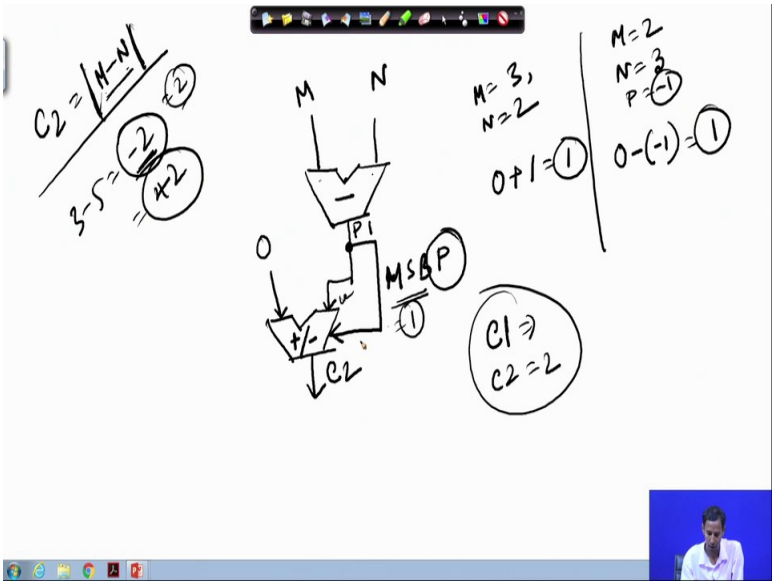
So, how can I design that? So, then you first start with this the C 1 which is the minimum of M and N. So, how can I find that? How can I find that? So, if I just if I just that means, I need 1 comparator ok. So, at that time comparator means whether that is that will that will give me that so greater than suppose this is M this is N ok. So, minimum of N need to choose right; so that means, I require suppose this M greater than equals to N comparator I am choosing over here ok.

So, after that what is what I need is that I am having this as a 2 is to 1 multiplexer and the output of this, this will select the corresponding values of M or N this will choose which value I require. That means, if I find that M is greater than; whenever this M will value will be greater than of; so at that time this will 1. So, at that time this will choose what this will this has to choose this N value. That means what? If this logic is true M is greater than N. So, at that time what will happen? This will obviously, I need to find out minimum of M and N.

So that means, M is greater than N. So, that means the minimum one is; so that means at this point this is 1; this is 0. So, now let us consider these particular circuits suppose M value is 5 and N value is 3. So, if this 3 5 and 3 compare so, this will 5 obviously, is greater than 3. So, this will come to 1; so that means, this is come to 1. So, here I am having 5 and this is 3. So, this multiplexer will choose 3.

So, I have got the minimum of M and N. The just consider the reverse case suppose this is 3 this is 5. So, at that time what will happen; at this particular comparator what will happen? 3 is not greater than 5; that means, at that time instead of 1 this will be false. So, it will come 0. So, 0 means whenever this is 0. So, this will choose M. So, that means at that time 3 will come at the output. So, that means whatever is the value of M and, by using this circuit I can easily implement this minimum of M and N ok, understood. So, that that means this is the C 1 function I have implemented.

(Refer Slide Time: 21:55)



Now, what is the C 2? That is mod of M minus N. Mod of M minus; I have to find out the circuit for that. So, for that what I have to do? That means, what is I require? M minus N first; that means I require what? I require 1 sub tractor which is giving me M minus; after that what I require? I require mod of M minus N; that means, whatever is the value of this M minus N whether that value is greater than if M value is smaller than the N.

So, at that time what will happen? I will get negative number. That means, the result will produce if I just do 3 minus 5; that means, this is this will give me minus 2. But here, mod means it will not produce the minus it will just plus 2 right. So, what will be the logic for implementing that? Now here I am finding out M minus N, but if M is greater at that time again it will by default it will produce plus 2.

So, here what I am doing M minus N and then again, if I am having another particular circuit where it is 0 and this is coming this is add or sub tractor; that means, circuit here this is fully sub tractor here add or sub tract this function is add or subtract. So, add or subtract means it needs 1 control signal. So, this control signal will come from this particular point. Why this particular point? Because what I said that whenever this is negative; So, this is this is the MSB of this particular if this is let us consider P; MSB of P P bit. So, MSB of P signal. How it basically works? This is at suppose so 0 means addition here addition subtraction logic is 0 means addition 1 means subtraction ok.

So, let us consider take 1 example and then you consider this things M equals to let us consider 3 and N equals to 2. So, 3 minus 2 equals to 1 positive 1. So, this MSB of P is 0. So, this will be 0 plus 1 which is 1. If you consider M equals to 2 and N equals to 3; so at that time, what will happen? The P equals to minus 1. So, minus 1 means the MSB of P that is equals to 1.

So, 1 means it will be 0 minus of whatever results I am producing. So, 0 minus of minus 1; so that is equals to 1. So, that means now I can develop the that means, using this particular circuit I can implement the function of mod M minus N. So, what we did? We have implemented the C 1 circuit, we have implemented the C 2 circuit C 1 circuit and C 2 circuit we have implement. So, output of this is C 2 and this is C 1; so, the corresponding bits that will be connected where to this particular positions.

So, C 2 will come here, the output of C 2 that will come here and the output of C 1 circuit that will come over here. So, this is not the this is for this is M register and this is N register. So, this is that means, the basic blocks we are just building to find out the ultimate circuit ok. So, this is the blocks; initially what we are trying to do? Initially, block wise we are trying to implement and then finally, we will integrate all of this to calculate or to achieve the final circuit of GCD of 2 numbers ok.

So, for today's class, this is it. In the next class we will try we will continue from this particular point and we will try to build the final circuit of GCD of 2 numbers.

Thank you for today's class.