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

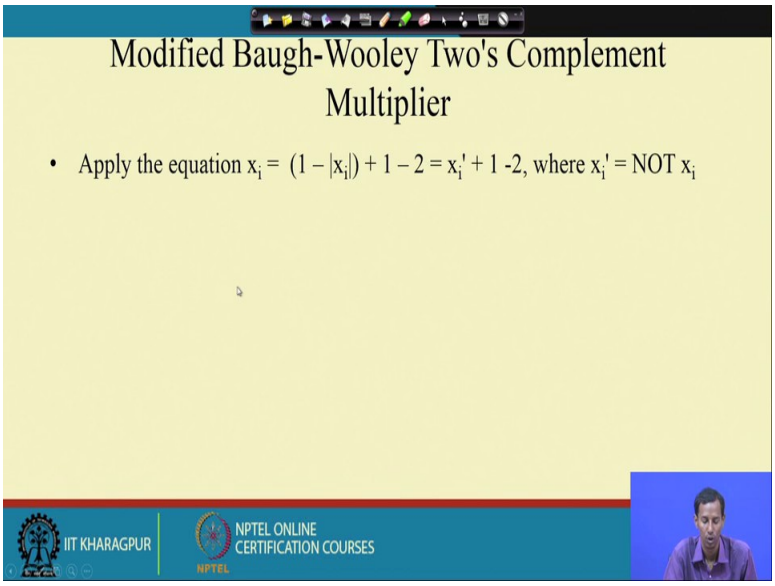
Lecture – 34
Multiplier Architecture (Contd.)

Hello everyone, welcome back to the course again on Architectural Design of ICs. So, in the last class we have we have seen that this sign multiplication method ok. So, how we have extended the sign using the Boolean algebra; we have converted that we know that so that z bar equals to 1 minus z and then z equals to 1 minute z bar. So, based on that 2 particular equation, so now we have come down to that how we can represent this as we are considering this sign multiplication, so at the time because of the MSB of that both the multiplier and multiplicand there width of negative weight. So, based on that we will get the partial product MSB that will also be in the negative, so that will represent negative numbers ok.

So, whenever we will add all these partial products, so at the time what I have to do I have to extend the sign. So, how I can extend the sign so based on this 2 Boolean algebra, so how we have that means, come down to the equation something like this suppose for minus a 4 x 0 bar ok. So, that we have come down to the equation that is equals to, so minus a4 and then a 4 x 0 bar and that is added with a 4 ok. So, that equation we have seen and we have continued to that with that particular concept or with that. That means, idea and then we have come down to the architecture which are being which are using lesser number of multiples whenever we are doing this sign multiplication method, which is known as this Baugh Woolley multiplier.

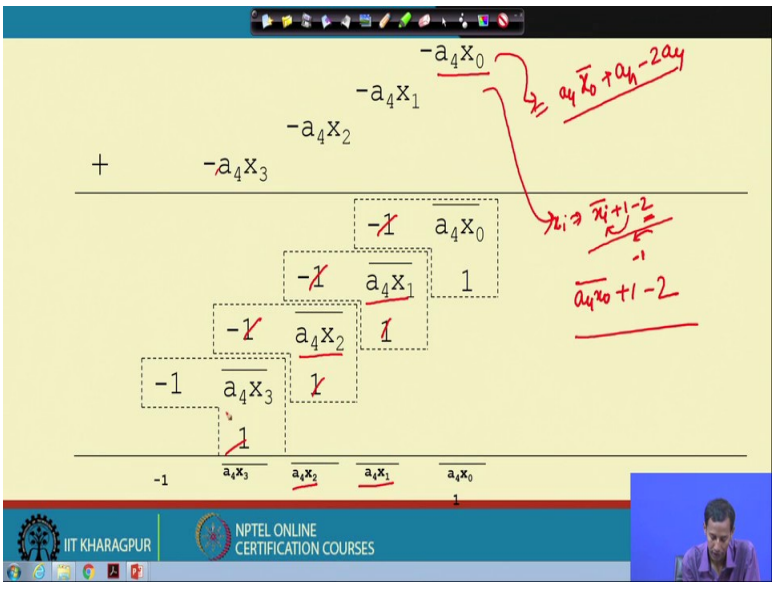
But still we that means, we have seen actually or we have analysed or we have observed that because, of that particular notation what we are doing we are increasing the maximum depth in corresponds to the unsigned multiplication ok. So, as we are increasing the depth that means depth of addition operation in chain. So, as I am increasing the depth that means I am increasing the delay too because, the number of stages for addition operation that will increase. So, that obviously it will increase the delay, so how can I reduce that then can I reduce that. So that we will see today's class.

(Refer Slide Time: 03:03)



So, in the last that means, in whenever we are seeing or we are reviewing that means sign addition operation. So, at the time we have seen this equation that xi equals to xi can be represented as xi bar plus 1 minus 2; where xi dash equals to that means, that is the xi bar or NOT of xi. So, this equation can be same as this xi ok. So, this is the equation which we follows in two's complement multiplier. So, if we follow this particular example or if we follow this; that means, this equation on that whatever for Baugh Woolley; that means, Baugh Woolley multiplier we are getting.

(Refer Slide Time: 03:58)



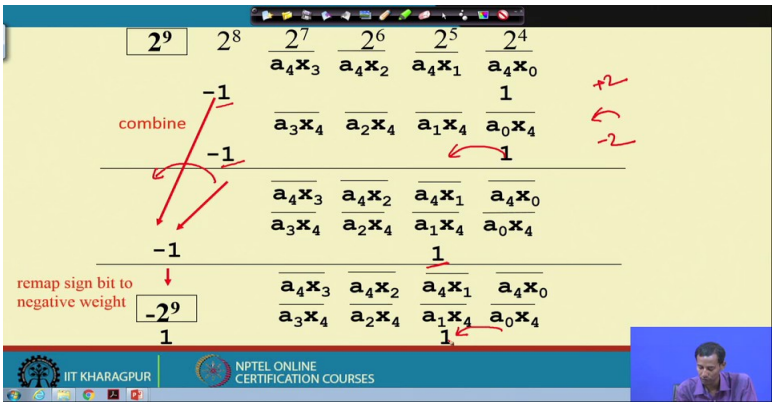
So, at the time for the this particular position what the value we are getting that is minus a 4 x 0 here minus a 4 x 1 this is minus a 4 x 2 this is a 4 x 3. So, now as I said that so this is now this particular. So this is if I consider this is a 4 x 0 if this is a 4 x 0. So, then this particular equation can be written as a 4 x 0 bar 1 over here minus 1 over here; that means, this equation is nothing but if I consider these as a xi, so xi equals to xi bar plus 1 minus 2. So, minus 2 means it is it comes to the that minus 1 to the next MSB and 1 means it is at the same position along with x I is a 4 x 0, so it will be a 4 x 0 bar ok

So that means, now instead of that initially what we are doing we are doing this as a 4 x 0 bar plus a 4 minus 2 of a 4, this equation has been represented in this manner. But now this equation has been represented using this that there is a4 x 0 bar plus 1 minus 2 ok. So, if we do for this all these values if we do the same, so at the time what I will get at this particular position so this will be a 4 x 0 bar 1 over here. So, 1 will cancel over here 1 1 here 1 1 here and minus 1 at this particular position.

So, what will be left at this particular position this is a 4 x 1 this is a 4 x 2 bar this is a 4 x 3 bar and then for the next this a 3 x 4 a 2 x 4 a 1 x 4 and a 0 x 4 on the same fashion, again I can write this a 0 x 4 as a 0 x 4 bar plus 1 minus 2 this a 1 x 4 minus of a 1 x 4 that can be represented as a 1 x 4 bar a 1 x 4 bar plus 1 minus 2 ok.

So, in this manner if I just represented all these negative terms, so what I will get I will get, so again 1 over here minus 1 minus 1 cancel minus plus 1 cancels or minus 1 plus 1 cancel minus 1 plus 1 cancel, so minus 1 at this particular position. So, a 3 a 4 a 3 x 4 bar a 2 x 4 whatever values I left so that we come to this particular position. So, then finally what I have to do finally, I have to add all these ok.

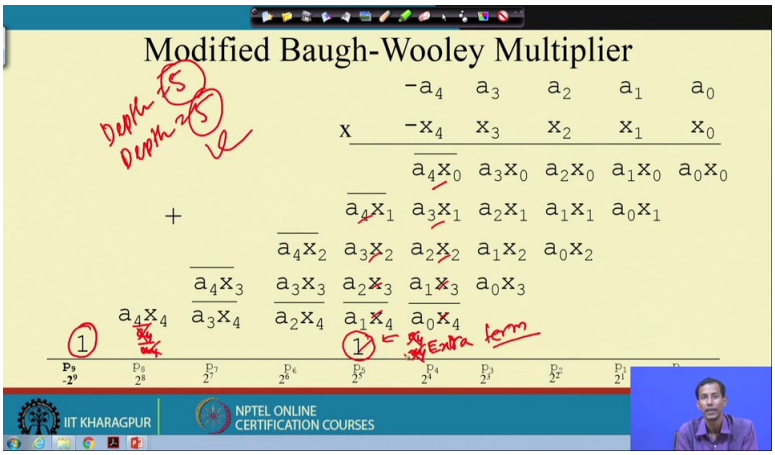
(Refer Slide Time: 07:20)



So, now what is happening, so this position was a 4 x 4; a 4 x 0 bar 1 a 4 x 1 1, so this 1 and at this particular position there was minus 1 minus 1. Now I have to combine this 1 and 1 means plus 1 sorry plus 2 ok. So, plus 2 means that 1 will be shifted to the next corresponding MSB; so now this 1 that will come to because of this 1 plus 1 that will come to this particular position with 1 ok.

So, then 1 minus 1 and minus 1 that is minus 2, so then again this minus 1 will come to the 2 to the power 9 position; so that means, 2 to the power 9 is the minus 1 with 1 and instead initially I was getting 1 at 2 to the power 4 position, but now it comes to this fifth position because 1 and 1 I am doing so that means 1 will come over here. So, what will be the final result that means, final circuit then I can represented that this a; at this particular position there is no extra term.

(Refer Slide Time: 08:53)



Initially what for that means, that Baugh Woolley multiplier what was there was a 4 and x 4 at this particular position, but in the modified Baugh Woolley multiplier it is not there and then at this particular position. Now this two 1 they are because, of that this one is coming at this particular position this 2 to the power 5 position and here initially what was there. That means, sorry for I think for this particular position it was a4 bar and x4 bar, but it is also not required. So, 1 is coming over here at 2 to the power minus 2 to the power 9 position ok. So that means, now what is my maximum depth that is 1 2 3 4 5 over here 1 2 3 4 5 over here; so that means, now the depth is maximum 5 ok.

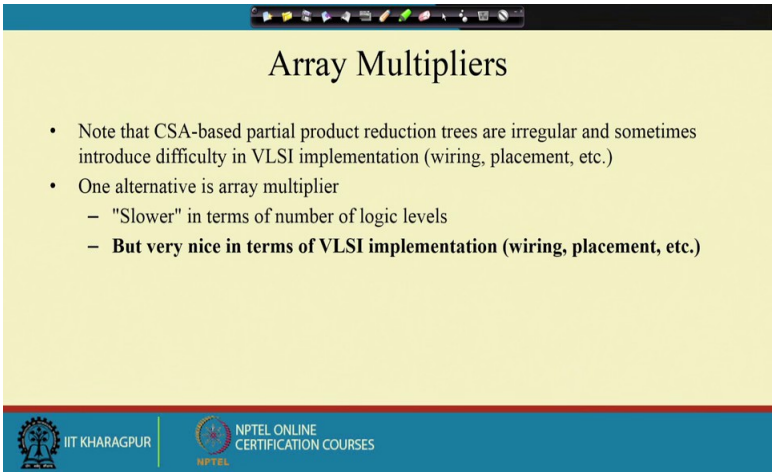
So that means, I can I can do this sign multiplication, for unsigned multiplication the maximum depth will be also of 5 for signed multiplication this for this modified Baugh

Woolley multiplier the depth will be maximum depth will be of 5 ok. So that means, now delay wise this circuit is improvised, so this is the modified Baugh Woolley multiplier. Now, you can do anything that means now you can use any of the tree; that means, structure for addition of this each of the that means operands or each of this multiple or you can use. That means, it for each of the stage you can use carry propagate adder or any; that means, first adder to add that means, to sum up all these partial products that also you can do ok.

So, now you can that means, do or you can for the final products to get the final products or to obtain the final product now you can add these partial products in your way. You can do this tree, tree method to add these partial products or you can use any first adder to sum up this partial product or you can follow the serial manner also to do that ok, now it is your choice. But the corresponding multiples generation that is basically done in this manner in modified Baugh Woolley multiplier ok. So, then these so that means, this is the basic fundamentals of sign multiplication which are basically this Baugh Woolley multiplier and then we have seen this modified Baugh Woolley multiplier ok.

Then there is another type of multiplication or multiplier architecture which is known as this array multiplication. So, then a name by itself it says that; that means so array means there will be kind of it will be kind of 2D architecture ok; so 2 dimensional architecture that means 1 in vertical 1 in horizontal ok. So, how I can do or how I can that compute this particular multiplication method or I can implement 1 multiplier using this array architecture that we will see now.

(Refer Slide Time: 12:55)



So, in array multiplier it is basically this carry save adder based partial product reduction tree, as we have seen that there that is this if you follow this Wallace tree or this dadda tree method they are structure is very much irregular ok. So, that is why we get some the issues in VLSI implementation.

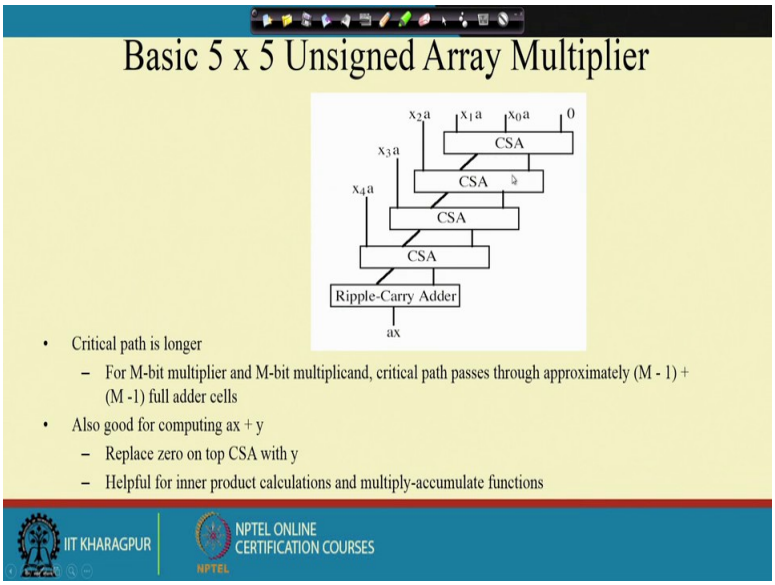
What are the issues are come with this that means, wiring of each of the cells and then placement of those cells. So, that becomes that means, a major issue whenever we consider one huge number of; that means, the word length for the inputs are more at the time it becomes serious issue for the VLSI implementation. For the lower number of bit width it is not that much that means problematic, but if you consider 32 bit or 64 bit multiplications at the time it becomes problematic or the complexity is increased for that Wallace tree or dadda tree method ok.

So, then what is the; that means, this array multiplication is that means the alternative tree reduction method to do that ok. So, what is that? That means, the problem of this irregularity in tree reduction or the tree multiplication that can be solved if we use this array based multiplication method. But it is basically slower in terms of number of logic levels and it gives me as because the structure is very much; that means, regular the flow of the data that is very much easy or that is very much like regular, so that whenever we will follow these VLSI implementation or whenever we do this VLSI implementation. So, at that time the problems or the complexity becomes lesser in compared to the tree based multiplication method.

But what is the problem? Here the thing problem here is that it is slow in operation in compared to this tree reduction method, tree reduction method whenever we have compared at the time what we have seen the tree reduction. That means tree based multiplier they are very much faster in nature ok. So, used this tree reduction that means tree based multiplier whenever you need speed as a major constraint ok.

So, then this is the that means basic 5 cross 5 unsigned array multiplier, so in array multiplier what is there; that means, the corresponding terms are basically coming through the carry save adder then sum, then it produce sum and carry.

(Refer Slide Time: 15:48)

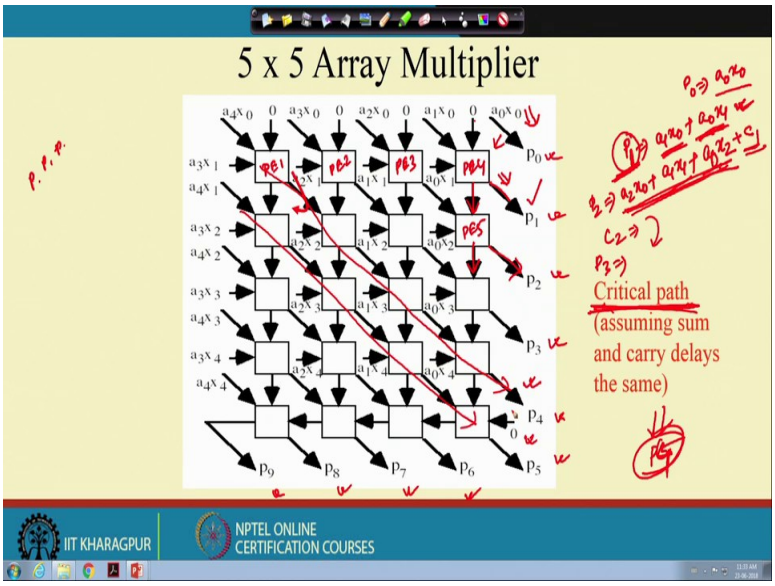


So, then each of the levels it considered because 2 of the inputs are basically coming to the next carry save level adder too. So, then 0 1 a sorry so this is for 0 1 a 1 that means, x 1 a x 2 a x 3 a x 4 a. So, all are basically that means added using this carry save adder and finally the bits has been added in carry ripple adder to get the corresponding results.

So, here the as you have you see the corresponding critical path at it passes through several that means face. That means, the structure is what one by one addition operation we are doing we are not doing that in parallel, so that is why the critical path is more. So, for M bit multiplier and M bit multiplicand the critical passes through approximately M minus 1 plus M minus 1 full adder cell ok. So that means, for M bit and M bit multiplicand and multiplier both are of M bit, so at the time it has to pass 2 M minus 2 number of full adder cell ok. But it can be good solution for computing a x plus y ok. So, how to do that that we will see in that means, later of this that means later of this slide ok.

So, as we have discussed that but the structure is very much regular ok; so that means, one by one you just put this corresponding input and then you will get the final results ok. So, this is the basic or this is; that means the idea of this array multiplication and from this look in to this picture I can easily say that the critical path is more, but the structure is very much regular.

(Refer Slide Time: 18:09)



So, if we just follow this particular that array multiplier architecture, so at that time what is happening so that means, each of this particular cell belongs to the, if I say this is the processing element if I name this as processing elements. So, this is processing element 1 this is processing element 2 then 3 then 4 something like this, ok. So, at this particular from these particular positions I am getting the corresponding outputs of the results ok. So, at the very beginning where from we are starting basically we are starting from this particular end ok, so what we know this is unsigned multiplication. So, what we know that p 0 p 0 is a 0 x 0 so a 0 x 0 is directly coming to this p 0, then what is my p 1; what is my p 1 that is a 1 x 0 plus a 0 x 1 correct.

So, in this next you see what I am this is p 4. So, in p 4 it consumes 3 input and produce 2 output ok; so that means, in this case so if a 1 x 0 and a 0 x 1 they are coming and they are producing this p 1 ok. So, then in p 2 in p 2 what is the case, in case of p 2 it will be what it will be it will be of a 2 x 0 plus a 1 x 1 plus a 1 x 2. And apart from that what will be another thing, which I have to add that is the carry from p 2 while I am adding this p sorry this p 1 while I am computing this p 1 at that time the sum bit is basically the p 1, but the carry which is produced because of this particular addition operation, so that will come over here ok.

So, that is why here you see that sum bit that is p 1, but the carry bit is basically coming to the next and in this case this is the this is considering this a 2 x 0, so a 2 x 0 and a 1 x

1 so this will produce this carry over here and sum will come to this ok. So, if this is p 5 so sum will come to this then again that will be added with a 1 x 2 and the c 1 from the previous case. So, c 1 is coming over here so and a 0 so this will be a 0 x 2. So, a 0 x 2 is coming over here, so finally it is producing this p 2 and the corresponding carry c 2 is going to the next level for calculating this p 3.

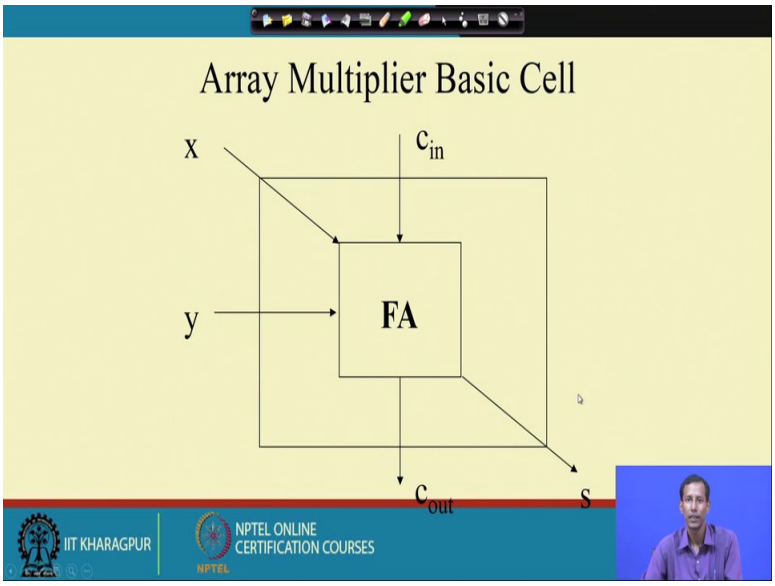
So, then again the p 3 will be considered this and again that will be c 2 will come down to this particular position, then again in each stage the corresponding carry is forwarded to the next level ok. So, then what is the that means corresponding critical path in this case which 1 will be the maximum; that means, the path which is considering from the input to the output ok. So, where I will get the maximum of that; that means, where I will get the maximum critical or the maximum depth I will get if I consider this corresponding that operation of this unsigned multiplication or multiplier architecture, so at the time the middle position which is nothing but this I think p 5 no p 0 p 1 p 2 p p 5. So, for p 5 position I will get the depth as maximum 1 ok, so p 5 or p 4 I think p 5 or p 4 ok.

So, at the time if I consider that means for p 5 oh sorry for p 4 this is the path it is following ok. So, and for p 5 this is the path it is following; that means, for p 4 and p 5 both are basically consuming 4 processing elements in this particular direction ok. So, then if I consider that each of the processing element is nothing but 1 full adder cell considering these 3 input bit and it produce this 2 output bit ok. So, then if I consider this; so; that means, here this particular architecture.

So, this will be the corresponding this p 4 and not only this p 4 again as this p 4 is basically again coming to this; that means, this is coming over here then at this particular position is also it is traversing to produce this p 6 p 7 p 8 and p 9 ok. So that means the total path which is basically the total critical path over here which is nothing but, then these to this; this is the maximum path which I am getting for computation of p 9 ok.

So, now that means if we consider this array multiplier architecture, the structure is very much regular but the delay is more because the carry is rippling through the corresponding array ok.

(Refer Slide Time: 25:44)



So, that is why this array multiplier array multiplier architecture this will gives you one better that means implementation on the implementation aspects it will be better, but for the other constraint it will be not that much advantageous ok. So, then we have seen this that Baugh Woolley multiplier twos complement Baugh Woolley multiplier is something like this.

So, this is the equations which we have got in Baugh Woolley two's complement multiplier, so now my question is that is this particular this operation or this table can it be represented using this array multiplier architecture. If you it can be so at that time how it can be represented or implemented via this array architecture that we will see in the next class. So, this is it.

Thank you for today.