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

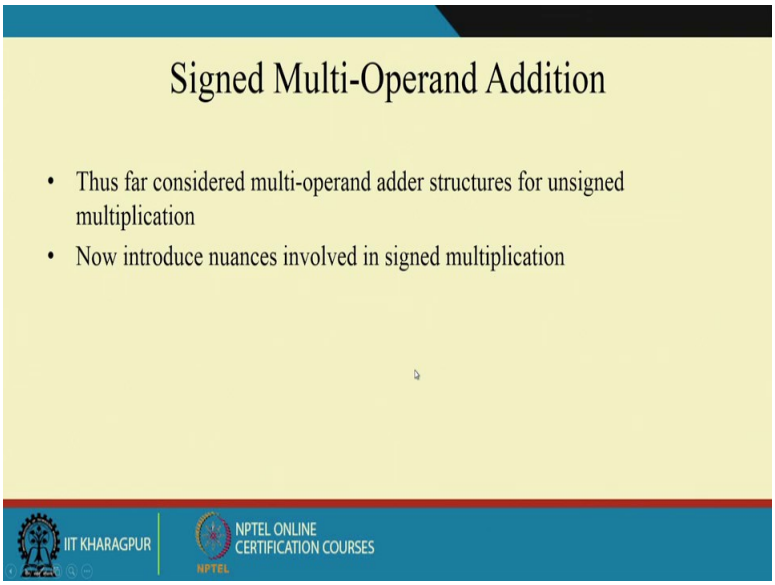
Lecture - 33
Multiplier Architecture (Contd.)

Hello everyone, welcome back to the course on Architectural Design of ICs. So, in the last few classes we are seeing that different multiplier architectures, ok. So, different multiplier architecture for different purposes that means, considering the speed or high speed operation or low power consumption or low area consumption. So, that till now we have seen this tree multiplication method, then booths multiplication method how we can that means combination of booths and then tree multiplication that we have also seen.

So, and then we have seen that if we compare all the multiplier architecture. So, at that time we have seen that if we combine or the hybrid architecture that means, the multiplier architecture which are based on two different techniques. That means, suppose for booths multiplier I am getting the area has the that means, advantage whereas, speed not that much. In compared to the Wallace tree multiplier architecture, where area is not that much efficient, but speed I am getting the advantages, ok. So, combining these two I can get one hybrid architecture where I can get the benefit of area as well as the speed, ok. So, that we have seen.

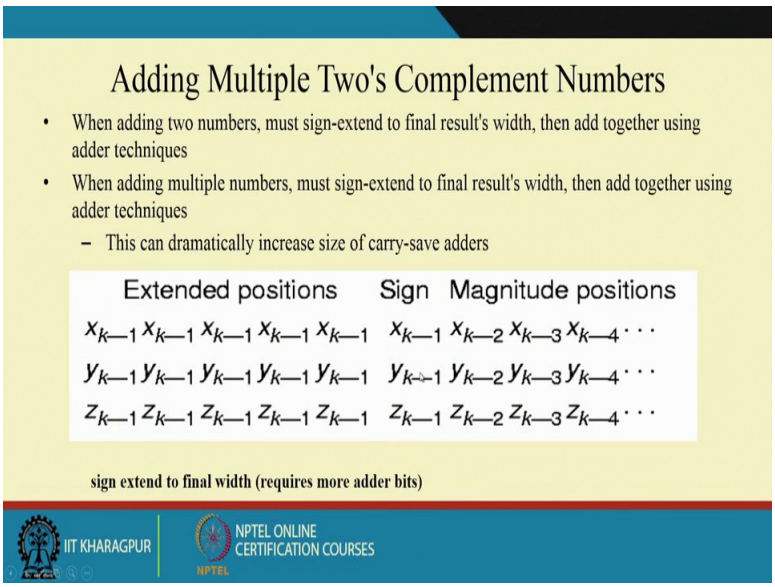
So, till now what we have learned or what architecture we have seen all those are unsigned multiplication, ok. So, now, we will see this how to do sign multiplication, ok. So, to before to start with sign multiplication we will just start with that means, just one review or quick review about this signed addition method because multiplication is nothing, but the addition operation. So, we will go through one quick review of the sign addition operation then we will start this sign multiplication method, ok.

(Refer Slide Time: 02:23)



So, this multi operand signed addition. So, thus far we have considered only this unsigned multiplication the adder structure which are for used for unsigned multiplication, now we will that means try to review this what happened whenever we do or what changes happened whenever we do sign multiplication, ok.

(Refer Slide Time: 02:42)



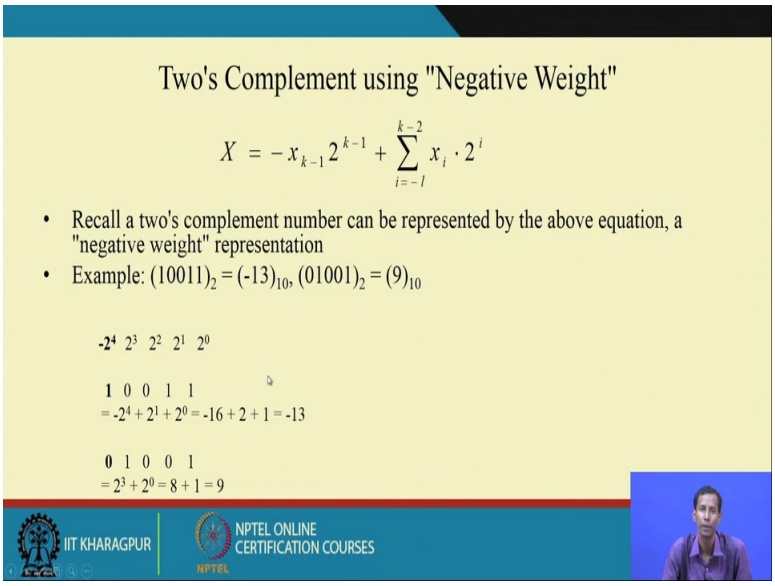
So, these things I we have already seen that in case of unsigned multiplication that we do not have to bother about the sign, ok, but in case of the sign multiplication what we have seen in booths multiplier. So, in those case with what we have to do we have to extend

the sign, because otherwise the corresponding in multiplication what we are doing, we are doing that means, for the first partial products it is same as the or that means, the bitwise ANDing operation with the that means, LSB of the multiplier bit, ok.

Then the corresponding next bit or the next bit position for the next bit position what will be the corresponding multiplicand value that will be one bit shifted and then it will be added. But at the time whenever we are doing the shifting, at the time means what? We are increasing the length in the MSB side from the partial product one, ok. So, in unsigned multiplication we do not have to bother about what is the sign which is for partial product one, but in case of signed multiplication we have to be bother about that, ok. So, that is why here you see the in this particular example suppose this is the magnitude position and this is the sign then this sign has been extended to this to fill all this that means, vacant position which we have not considered in case of unsigned multiplication, ok.

So, when we are adding 2 numbers it must be sign extended to the final results width, ok, and then add together using the adder technique to avoid the that means, or to get the proper results. When adding this multiple numbers must sign extended to final results width and then add together using adder technique this can dramatically increase the size of carry save adder that means, as initially for unsigned multiplication what was there this position was vacant. So, I am is increasing that means, filling with this extended bit sign bit position, so that means, I need more number of adder this position I should add I have to add, ok. So, addition of those require adder operation whether that is full adder cell or half adder cell. So, I need adder so that means, this will increase the size of the carry save adder or full adder cell, if I say.

(Refer Slide Time: 05:20)



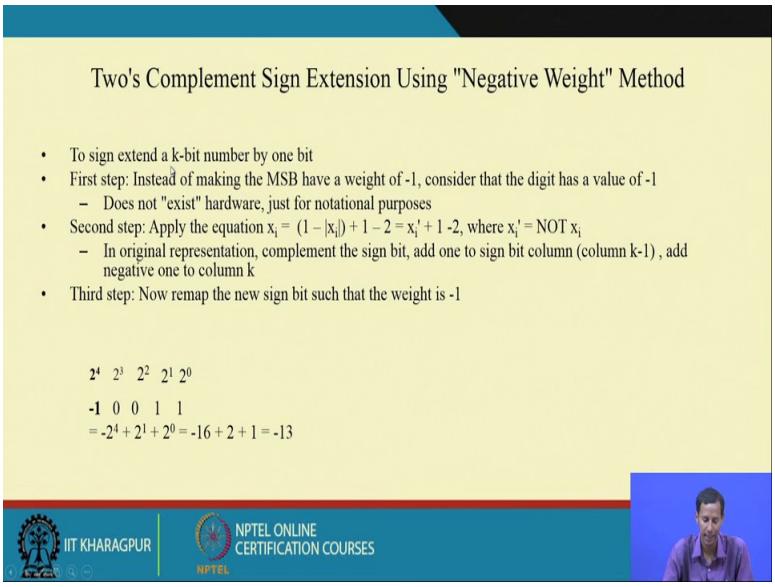
So, the two's complement basically that has been represent or that has one negative width method, ok. So, what is negative width? In two's complement format whatever is the MSB that has been considered as the that means, the weight for that that is minus, ok. So that means if I suppose this 1011, if I consider these in two's complement format. So, at that time this number will represent minus 13. Why minus 13? Because this is 5 bit, so the positions of the bit or the weight of the bit will be 2 to the power 0, 2 to the power 1, 2 to the power 2, 2 to the power 3 and the MSB position will be minus of 2 to the power 4, ok.

So, now as I am getting one at this particular position. So, this is minus 16 and then the corresponding 1 1 for 1 1 this is 3, so 16 plus 3 that is equals to minus 13, ok. So, now, for the same case if I am getting 01001 combination that is also in two's complement format if I represent, so because of this 0 this represent that this is one positive number, ok. So, positive number means to this position is 0 means this is no negative values are there, ok. So, that is why this is this position is 1. So, 2 to the power 3 that is 8 and this position is 2 to the power 0 1 so that means, that value of that this particular number is 9, ok.

So that means, in two's complement, but if we do not use this two's complement number representation. So, at that time what will be this value, this will be 16 plus 3 plus 1 the value will be 19 in case, the signed representation, ok. So, that is the difference between

signed representation and unsigned representation.

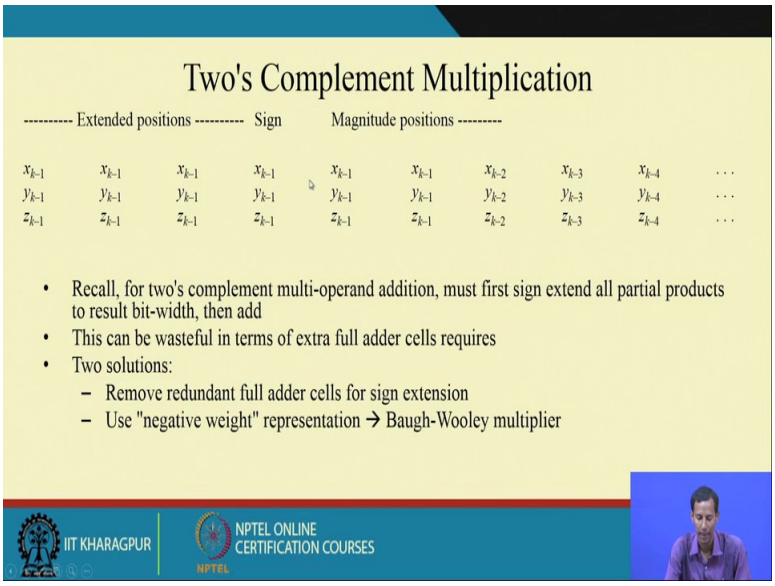
(Refer Slide Time: 07:27)



Then two's complement sign extension method how we can do to extend to sign extend a k-bit number by one bit how to do that that is the first step is that instead of making the MSB have a weight of minus one consider that the digit has a value of minus 1 which does not exist in hardware because this is just for the notational purpose, ok.

And then in the second step apply this equation where this basically this x i dash to represent the inversion of x i bit. In original representation complement the sign bit and add one to the sign bit column and add negative 1 to the column k. So, actually this is the step and in third step now remap the new sign bits such that the weight is minus 1. So, this is the basically steps are involved to how to extend the sign bit, ok. We will see whenever we will take the or consider the example. So, at the time it will be much more clear to you.

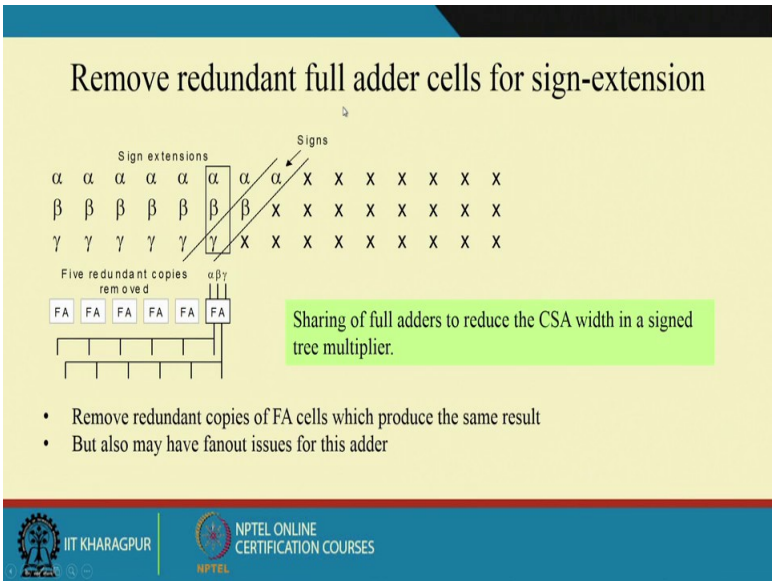
(Refer Slide Time: 08:45)



So, if we consider the signed tree multiplication, so at that time how we can do the signed tree multiplication. So, at that time, so this is the signed position and this is the magnitude position now this extended whatever is this value sign that has been extended to the corresponding width of the final result or width of the product, ok. So, then this what it says recall for two's complement multi operand operation must first sign extend all partial product to result bit width and then add, ok.

So, whatever what we have done? We have extended the sign to the final results all the partial product has been extended up to what up to the results final results bit width. And this can be wasteful in terms of extra full adder cell requirement. Then two solution can be that means, proposed or two solution can be introduced to solve this problem, remove redundant full adder cell for sign extension or use negative weight representation which is basically follows this Baugh-Wooley multiplier, ok. So, what is the architecture for this Baugh-Wooley multiplier that we will see now.

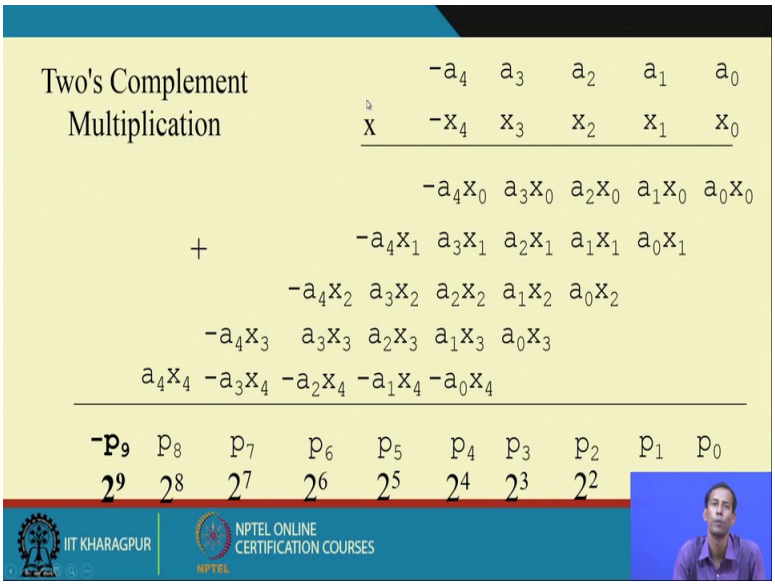
(Refer Slide Time: 10:07)



So, how to remove this redundant full adder cell? How to remove this? These are the signs, and then 5 redundant copies can be removed. Actually we have seen how we can reduce this particular terms that we have seen in the last class we have already discussed that. That is if we are I am representing this particular number in ones complement then the sign will be for the first partial product it will be extended by 3 position that is s bar s and s, and if I have to if I am representing this number in two's complement format then I do not have to add the sign to the LSB position, but if I do not have, if I have not done that at that time I have to add the sign bit to the LSB position.

And for the rest of the case what it will be 1 and s bar. And at the last partial products it will be just as it is. So, before to the last it will be just the s bar only not 1 and s bar. So, that we have already seen in the last class we have already discussed this how to remove this, ok. So, then apart from that there is another method how we can do.

(Refer Slide Time: 11:28)

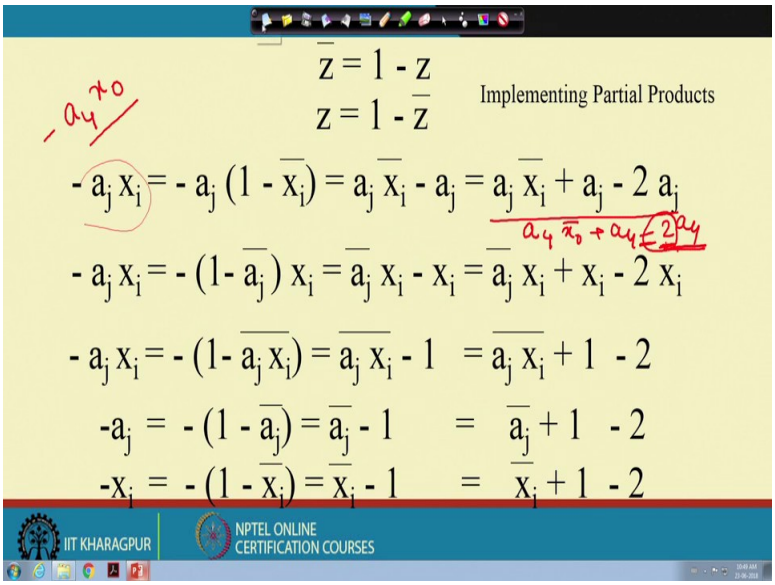


So, suppose this is the two's complement multiplication format, suppose this is the a and this is the b. So, the corresponding MSB of a, this is represent in two's complement as these 2 multiplier and multiplicand both are in represented in two's complement format. So, at that time the MSB of that that represent minus 4 sorry minus a 4 and the multiplier that means, MSB that represent minus x 4.

So, now bitwise ANDing operation if we do then for these I will get minus a 4 x 0, then for this minus a 4 x 1 for this position a 4 x 2 this is minus a 4 x 3. And for this final that means, this for the fifth one all this will be minus except from this because minus it is plus and rest of the rest of the that means, multiples they are in negative, ok.

So, now in basically in hardware whenever we will design the hardware, so at the time how I can represent this as minus, ok. So, what is the technique to represent this particular bit, ok.

(Refer Slide Time: 12:48)

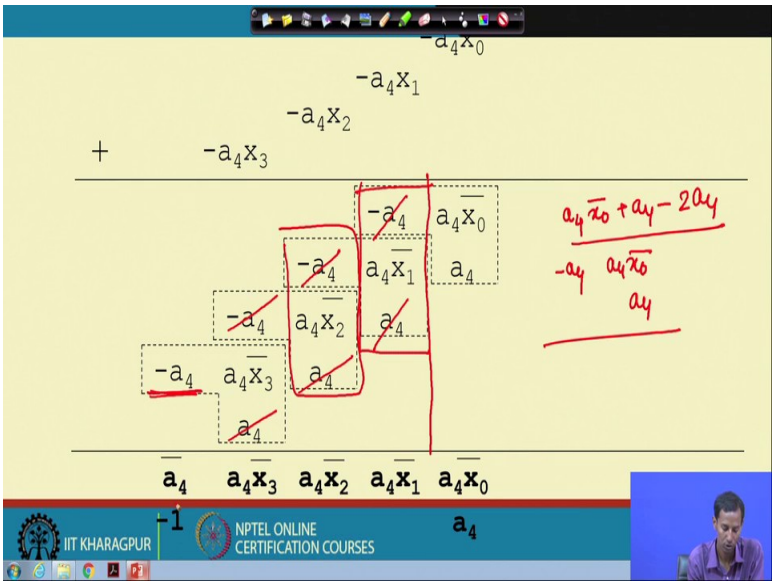


So, there is actually if we follow this Boolean algebra. So, at that time I can write this z bar equals to 1 minus z or z equals to 1 minus z bar, ok. So, now for this particular case, for this particular case whenever I am getting minus a 4 x 0 so that can be written as if I find that, minus a j x i. So, that can be written as a j. So, x i can be written as 1 minus x i bar, ok. So, then it will be what? Plus a j x i bar minus a j. Then again this a j can be minus a j can be written as a j x i bar plus a j minus 2 a j. So, this minus a j x i this can be represented as this one, ok.

So, why I have to or why I have come down to this particular equation? There is one intention to do that because if you remember. So, at the very beginning whenever we are talking about the sign addition operation at that time we have seen that in the second step of extending the sign we have seen that we can represent the number negative number in k in terms of x i bar plus 1 minus 2, ok.

So, if we just see this particular example. So, we have discussed that this particular equation x i bar plus 1 minus 2, ok. So, to if you if I follow this particular equation, if I follow this particular equation this is also nothing, but this is also same as that one. If I take common of a j to everyone so at the time how it will be x i bar plus 1 minus 2, so that means I can represent this number as same as whatever equation I have represented there, ok. So, in this manner I can represent, ok.

(Refer Slide Time: 15:08)



Now, for that particular position suppose for minus a 4 x 0 minus a 4 x 0 that can be represented as how? How it will be represented? So, minus a 4 then a 4 x 0 bar and a 4. Why is so? Because here you see minus 2 a j, so this is minus a j x i, ok. So, this particular bit, ok

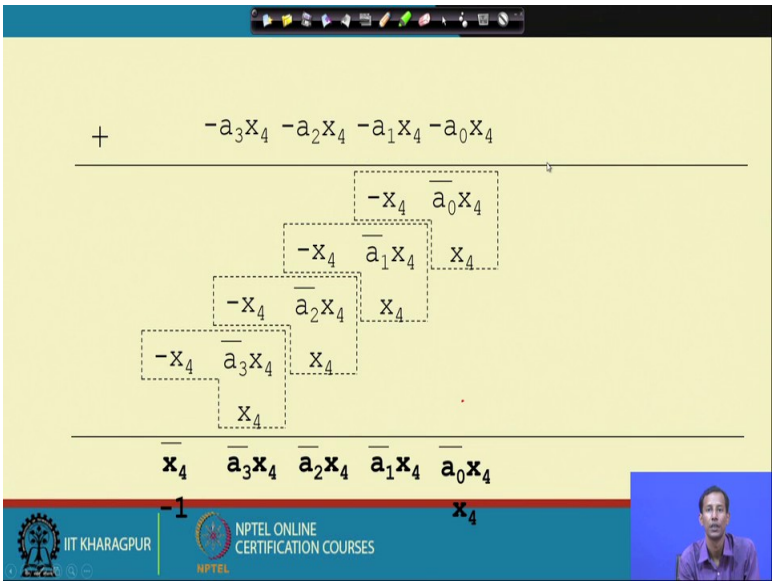
Now, suppose I have to that means, if this is if I consider for the next that means, for these if I consider this as a 4, minus a 4 suppose this is x 0. So, this can be right now if I represent this particular equation in terms of this j and i, so at the time what it will be it will be a 4 x 0 bar plus a 4 minus 2 a 4. So, minus 2 a 4 means what? It is it will come 2 of a 4 means what it is just left shift by 1, ok. And what is the corresponding weight? That is minus and this value is minus so that means, it will be the weight for that will be in minus, ok. So, that is why in this particular minus a 4 x 0 that can be represented as whatever we have seen in the minus a 4 x 0 that will be a 4 x 0 bar plus a 4 minus 2 a 4, right.

So, this can be written as so minus 2 a 4 2 a 4 means, so this is the position where I will get a 4, so this is a 4 x 0 bar and plus a 4 means, so this is the position. So, it is just like the same this equation can be represented as this one because this is this has the 2 bit shift weight. So, this minus a 4 over here means this is minus 2 a 4 plus a 4 x 0 plus a 4. So, this equation can be represented as this one. So, same for these a 4 x 1, so a 4 x 1 for a 4 x 1 what it will be, minus a 4 over here and a 4 x 1 bar over here and a 4.

Then again for this particular case a 4 x 2 minus a 4, a 4 x 2 bar a 4. For this 1 minus a 4 x 3 minus a 4 a 4 x 3 bar a 4 correct. Now, if I just add the corresponding position wise addition operation. So, at this particular case what I am getting at this particular case a 4 x 0 bar a 4, then this at this point at this particular case. So, minus a 4 plus a 4 it will be balanced, ok. So, only what I will get I will get only a 4 x 1 bar over here.

So, at this particular case what will happen? So, a 4 bar sorry minus a 4 and plus a 4 will cancel each other. So, I will get a 4 x 2 bar. So, same as this particular position, so it will be a 4 x 3 bar. And, for this minus a 4, ok. So, this is minus a 4, so minus a 4 can be written as a 4 bar minus 1, ok. So, because of that particular equation, z bar equals to 1 minus z or z equals to 1 minus z bar, ok. So, j that means, a 4 bar means that in that is nothing, but a 4 bar means a 4 bar means if I just consider this minus of a 4. So, that can be represented as a 4 bar minus 1.

(Refer Slide Time: 19:51)

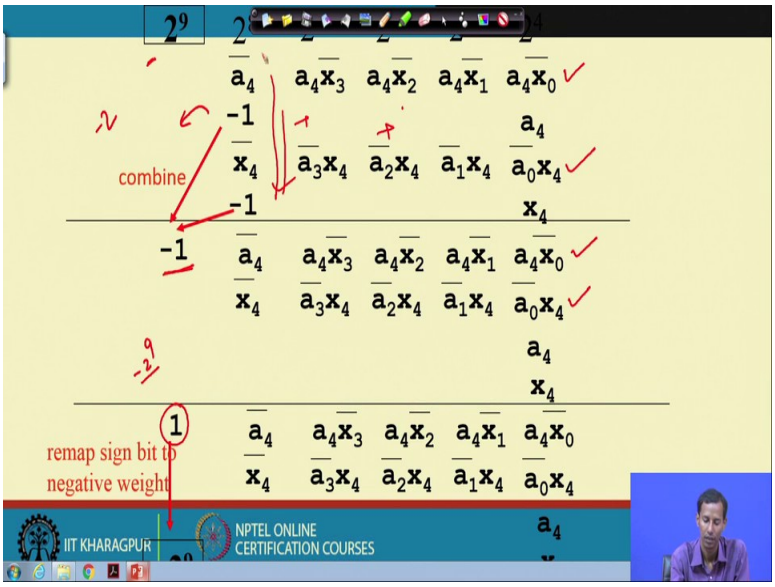


So, then if I just consider the that means, what I left the last position that means, this the last partial products which I am getting, so this one. So, this one is left. So, how I can represent this? So, this can be represented as, so minus a 4 x 4. So, the as same as x 4 minus x 4 at this particular position a 4 bar x 4 and x 4. So, here x 4 is common a 0 is basically changing, right. So, that is why keeping in the previous case what was what we are doing, that a 4 was common in each of the multiples whereas, the values other values x 0 is basically changing. Here x 4 is common and a values are changing. So, that is why

this x 4 is basically coming over here. So, minus x 4 a 0 bar x 4 x 4, ok.

So, on the same way here in this particular equation I can write as or I will come down to this equation that is a 0 bar x 4 x 4, a 1 bar x 4, a 2 bar x 4, a 3 bar x 4 x 4 bar minus 1, ok. So, now, they one that particular equation now, these values will be replaced if I replace this values on that sign multiplication whatever multiples we are getting. So, now, after that means, replacing these values what will be the final equation, ok.

(Refer Slide Time: 21:42)

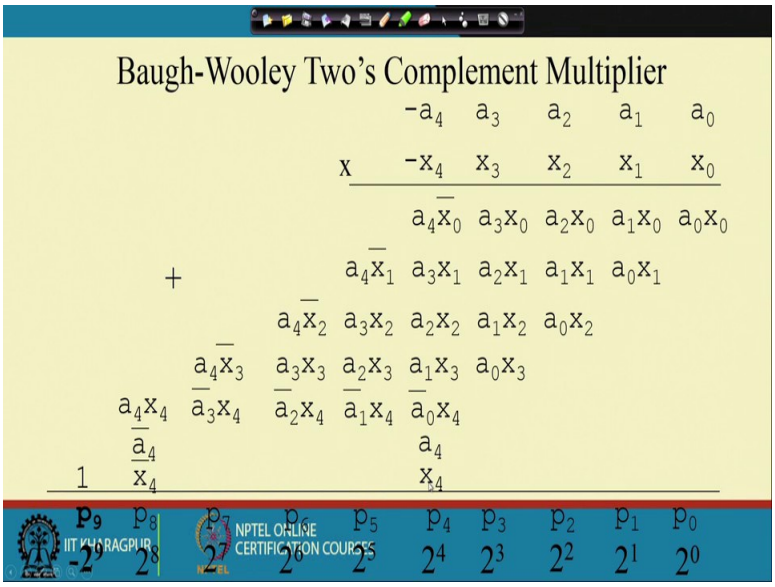


So, if I consider so at that time this 2 to the power 8 position that is minus a 4 minus 1 this is for whenever I that means, a 4 is common, ok. So, minus a 4 minus 1, a 4 x 3 bar, a 4 x 2 bar, a 4 x 1 bar and a 4 x 0 bar along with a 4 at this particular position. And for the minus x 4 where for that means, x 4 is common for a value is changing. So, at the time for that particular equation whatever I will get that is minus x 4, a 3 bar x 4, a 2 bar x 4, a 1 bar x 4 a 0 bar x 4 along with x 4 at this particular position and minus 1 at this particular position.

Now, you just combine this all these that means, equation. So, at that time what I will get at this 2 particular, at this particular position, ok. So, a 4 x 0 bar and a 0 x 4, so a 4 x 0 bar and a 0 bar x 4, ok and then a 4 and x 4 a 4 and x 4, so this position is a 4 x 1 bar, a 1 x 4, so this is this. So, this will be added, this will be added and for what will happen for this particular case.

So, a 4 bar and x 4 bar will come here. So, minus 1 and minus 1 that will give me minus 2. So, again minus 2 means what? Minus 2 means it will come to the or in the that particular width will shift to the next MSB side, ok. So, then this minus 1 will come at this particular 2 to the power 9 position, ok. So that means, minus 1 means this is nothing, but with negative weight. So, this if this bit is 1, so this will be minus 2 to the power 9, and then rest of the that means, equations are something like this, ok.

(Refer Slide Time: 23:52)



So, now this can be now, this particular equation now that can be replaced as this one that means, at this particular position what is the value I am getting? That is a bar a 0 bar x 4, where this is a 4 and x 4 are added at this particular position and here you see a 4 x x 4, a 4 bar and x 4 bar along with one at the 2 to the power 9 position which is minus 2 to the power 9, here you see that is minus 2 to the power 9. As I am getting 1 over here so this position will give me minus 2 to the power 9, ok. So, this is the Baugh-Wooley two's complement multiplier or the that means, operation or the principle of operation of this Baugh-Wooley two's complement multiplier architecture, ok.

So, now my point is that can I reduce then this so that means, for this for the how this is basically different from the normal this unsigned multiplication if you see that in the middle position if I do not consider the sign extension. So, at that time I will get only the multiples up to this that means, from a 4 to this that means the 5 over here ok. But to extra that means, multiples are coming which are a 4 and x 4 because of considering the

two’s complement number representation that means, I am increasing the depth here. So, initially the depth was 5 here, but here now it becomes 7. So that means, if I am increasing the depth maximum of all the that means, possible combination so as I am getting the maximum depth over in this particular position that means, I am increasing the delay for this particular multiplier.

But the thing is that as I have consider this two’s complement that means, multiplication method or I have a multiplier which will give me or which will produce the results are in which are in two’s complement number representation. So, at that time I need to do this there is no other way out to remove or to that means, not to use this a 4 and x 4. So, I have to use this to get the proper result, ok. So, then my point or my observation or my focus will be how to reduce this length, ok, whether I can do that, if I can do that then how I can do that, ok. So, that we will discuss on the from the next class, ok. So, for today this is it.

Thank you.