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

Lecture – 15
Efficient Adder Architecture

Welcome back to the course on architectural design of ICs. So, in the last few classes we have seen the architectural; that means, implementation techniques for some algorithm, how we can implement different architecture considering speed, power, area as the major constraint ok. So, and after that there will be one tutor; that means, some of the tips and tricks about increasing the; that means, how we can increase the speed by and whenever we are doing this things. That means, we are applying this tricks at that time we will not, that means, of focus from that the functionality will remain same.

That means keeping the functionality same, how we can change the algorithm or in the architecture, what we can change so that I can get the benefit in terms of power or in terms of area or in terms of speed ok. So, that tricks we have seen and in your; that means, suppose for your work it is not that all the circuits we are covering in this particular course. So, suppose you are facing some problems in your circuit design. So, at that time please let me know via forum of this particular course.

So, today we will start with another newer topic which is basically where, here what we are trying to do, we are dealing with the basic computer arithmetic operation. So, computer arithmetic operation means, in computers we generally use or we can perform this addition, multiplication, division and all this things then shifting all this things you we can do ok. So, how; that means, why I need to learn this architecture or whenever we are learning this architecture at the time how they have been evolved; that means, from what they have been started.

And now at this point, now where we are so; that means, why there is a; that means, this evolvement is there because as I said that; that means, day by day this, that requirement is basically increasing ok. So, that in terms of speed or that in terms of power or that in terms of area ok, requirement means either I need low power for handled device or; that means, battery power device I need the; that means, the power. That means, should be

such a that power consumption should be in a such a low level so that I can operate the device for a longer time ok.

And the second thing is that I need more speed; that means, I need the computation time should be so much less, so that, that that whenever I will give the input at the time the processing of the that signal processing blocks or this arithmetic operation that should be so much fast that, I can get, I do not have time to wait ok. So, I need it very fast for the computation purpose ok. So, for that reason I need, the critical path should be low because critical path is the measure constraint or that is the bottle neck of your circuit, which defines the overall operating; that means, frequency of your design ok.

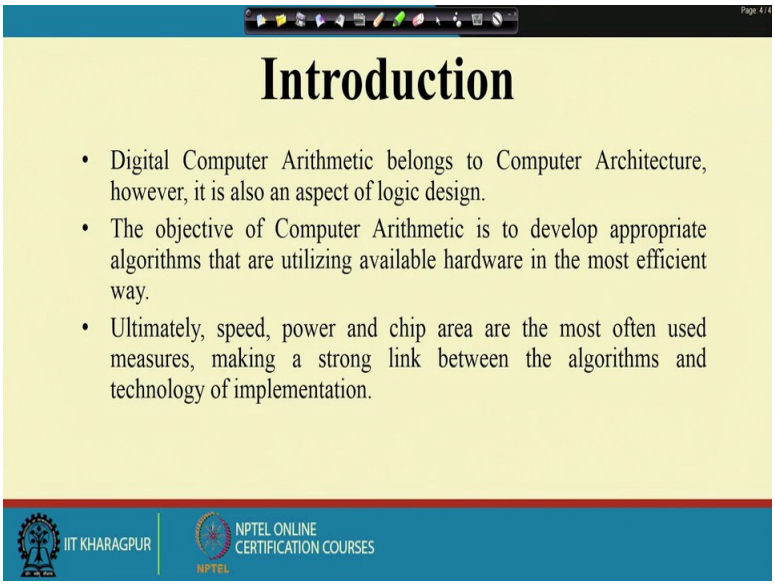
So, that is why we are trying to, we will always try to reduce the critical path in particular circuit ok. So, today we will start with a newer chapters, that is adder architectures. So, adder we generally know that is; that means addition operation basically ok. So, now, why I need addition operation to be consider, we need that basic operation of full adder or half adder architecture we have already seen in the digital electronics course. Then why in this particular course architecture design of ICs again we need to learn this adder architecture ok. Here, in this architectural design of ICs we need to learn this adder architecture, we have learned only in digital electronics we have learned only two type of architecture.

Which are the; that means, stating of adder which is nothing, but that ripple carry adder and carry look add adder, but after that there are; that means, several architecture which are already people have developed or people have if I know; that means, found out for different application ok. So, different, different application means, suppose where somewhere I need more power; that means, the consumption power consumption should be very much low. So, at that time that architecture should be different, some somewhere I need the; that means, the speed of the architecture or because addition operation I need, suppose I need to do multiplication.

So, at the time I have to use addition operation right. So, addition operation is the basic component of a multiplication. So, at the time if I just; that means, improved the quality or the performance of the adder so; obviously, I can improve the the quality or performance of the multiplier too or in any function or in any circuit if it is used then, at that time I can get the benefit of that, ok. So, that is why we will, we will here in this

particular adder architecture, we will start with the basic things, then from the basic how we are being evolved day by day that we will done ok. And nowadays where we are, we will take some few of the works and then we will try to discuss on, in this particular course ok.

(Refer Slide Time: 06:36)



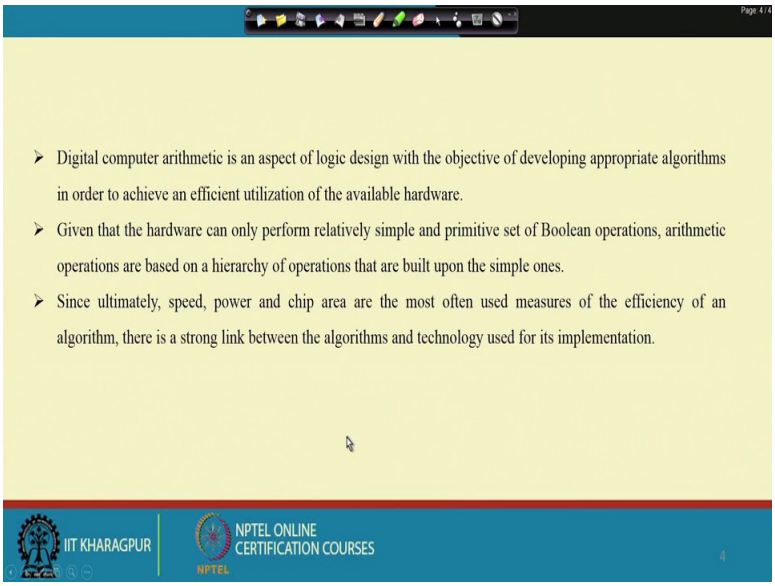
So, then as I said that digital computer arithmetic belongs to computer architecture, how where it is also an aspects of logic design ok. So, and the objective of computer arithmetic is to develop appropriate algorithms, that are utilizing available hardware in the most efficient way ok. And what is my our ultimate target, our ultimate target is that the speed power and the chip area should be in a lower, on the lower side ok, so that is our ultimate target. So, here what we see that ultimately, speed, power a and chip area are the most often used measures making a strong link between the algorithms and technology of implementation ok.

So; that means, whenever we are designing one; that means, computer architecture. So, at the time we have to follow this digital computer arithmetic or we have to design this computer arithmetic circuit. So, where we will basically implement different algorithms, considering this ultimate target of speed, power, area ok.

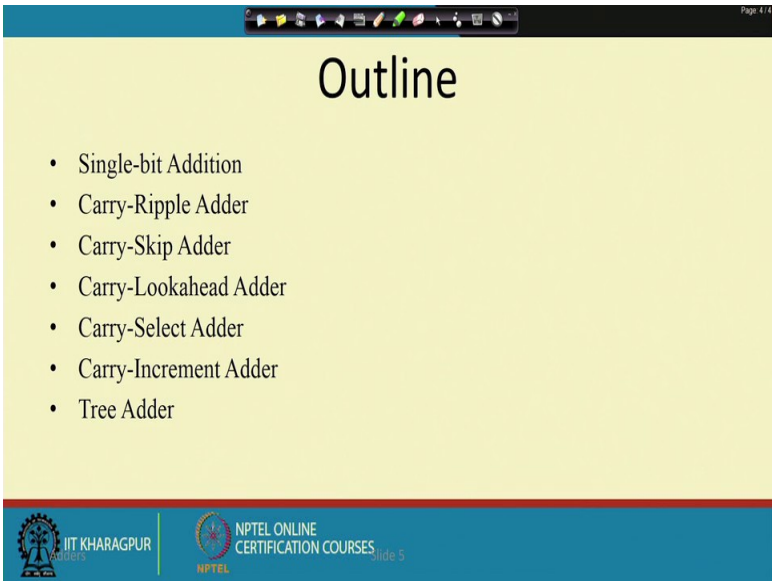
(Refer Slide Time: 07:50)

So, with the basic operation is computer it; that means, architectures are this addition, multiplication, multiply add, division and then evaluation of different function, then multimedia operation. So, these are the basic operation in any this computer architecture.

(Refer Slide Time: 08:08)



(Refer Slide Time: 08:14)



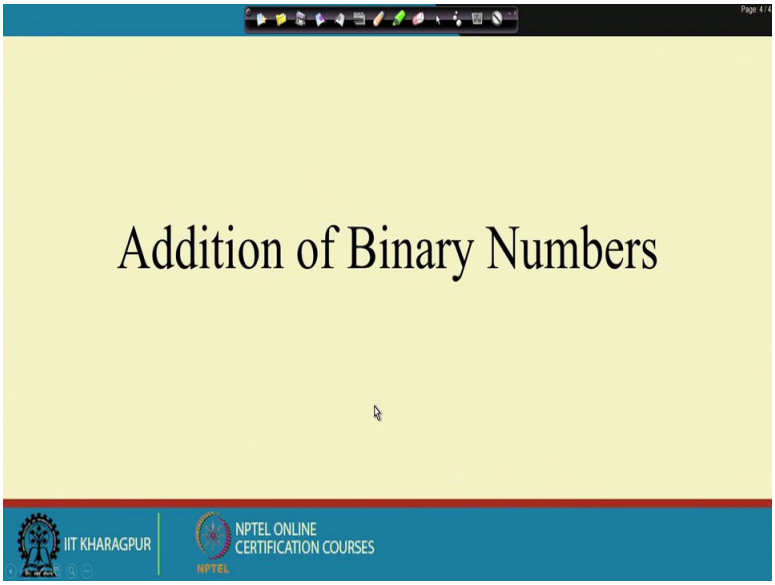
So, if we see; that means, if we start with this addition operation first, as here you see this addition. So, if you see the with this addition operation. So, at the time what will happen there a several; that means, adder architecture. So, here we, I have listed only few of them, it is not that all the; that means, adder architectures are listed in this particular; that means, presentation, but we will see more on the, more of them whenever we will just continuing with this particular lecture series ok.

So, first the thing is that we have started with half adder and full adder which is which are nothing, but they single addition operation, then carry ripple adder. So, carry ripple adder is nothing, but whenever we are considering n number of bits; that means, whenever we are considering multi bits. So, at the time we have to follow this ripple carry adder or carry rippling adder, then there is this carry look ahead adder. So, carry look ahead adder along with this carry look ahead adder, there is another adder which is carry skip adder, then carry select adder, then carry save adder, then carry increment adder, then tree adder then conditional sum adder.

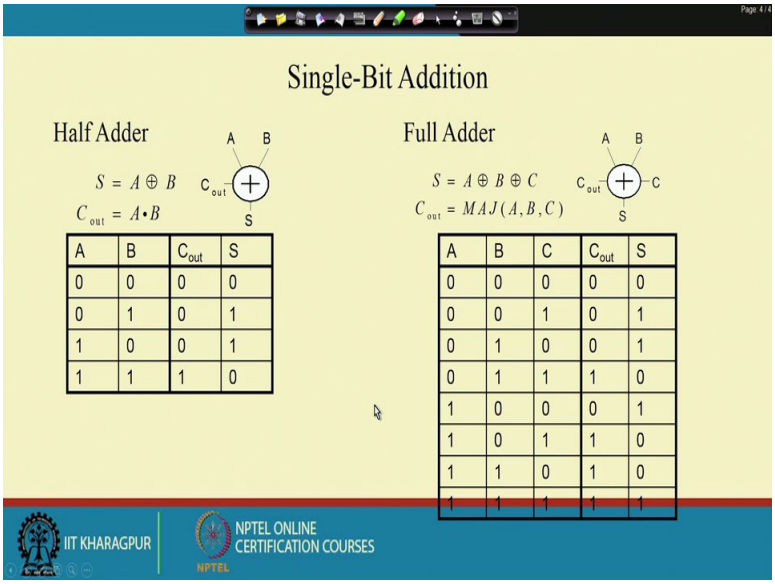
So, there are so much of topology or so much of techniques for implementing this single addition operation; that means, that is nothing, but two add, two addition of two numbers basically ok. So, in we will see in what aspects basically they are different, why they are named as of different name. So, whenever they have named as different; that means, they

have some indication or they have some modification on the particular circuit ok. So, we will look into that, how it is being developed and we will see in this particular course.

(Refer Slide Time: 10:22)



(Refer Slide Time: 10:25)



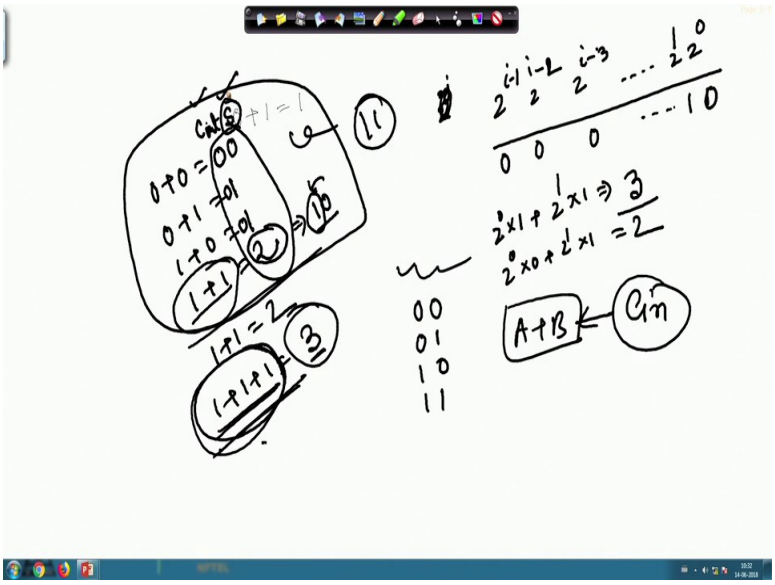
So, addition of binary numbers; so this is the basic we have learned in digital electronics. So, in half adder what you have seen that suppose this is this A, B and, A and B are the two inputs and C out and S are the two output.

Where S indicates the sum bit and C out is the carry output bit. So, as we are considering two bits A and B as input of half adder circuit, so at the time I need to do binary addition

operation ok. So, binary addition operation means whenever I will add 0 along with 0; that means, at the time my sum will be 0 and carry adder will be 1. If I add 0 with 1 at the time sum will be 1, carry out will be 0 whenever 1 and 0 then again sum is 0 1 and carry out is 0 and whenever this both the bit; that means, A and B are 1 at the time sum is 0 and carry out is 1.

So, why I need that ok, so why I, I am getting C out as 1 and sum as 0 at this particular position. So, if I just; that means, take one example of that then at the time it, if you just see that in decimal; that means, addition operation if you do.

(Refer Slide Time: 11:59)



That is 0 plus 1, that is equals to 1 ok, initially what we know that 0 plus 0 that is equals to 0. So, 0 plus 1 that is equals to 1, 1 plus 0 that is equals to 1, then 1 plus 1 that is equals to 2 in decimal addition operation ok.

So, in binary we cannot represent; that means this 2, but we can represent this 2 using two bits, using single bit, I cannot represent this 2. So, these 2 can be represented 1 0 in the binary; that means, coding style ok. So, why 1 0 basically, in binary what we know binary coding style what we know; that means, each of the bit position is sorry 2 to the power i minus 1 then sorry, this is 2 to the power i minus 1 this is 2 to the power 2 then i minus 3 dot dot up to 2 to the power 0, this is 2 to the power 1 ok. So, the bit position is weighted something like this; that means, if I put 0, 0, 0 and 1, 1 over here. So, at the time what will be the corresponding decimal value for this particular binary number.

That is 2 to the power 0 into 1, plus 2 to the power 1 into 1 that is equals to 3 ok. So, for 2 what I need, I need 0 so; that means, 2 to the power 0 into 0 plus 2 to the power 1 into 1 that is 2. So, for that whenever I am getting 0, 1, 1 so this I can represent as 0, 0 this I can represent as 0, 1, this I can represent as 0 1 too, but for 2 I need 1 and 0. So, that is why this bit is basically carry out and this bit is the sum ok. So, that means, whenever I am adding this 1 plus 1 at the time this sum is 0 and the over flow bit which is nothing, but this carry bit, ok.

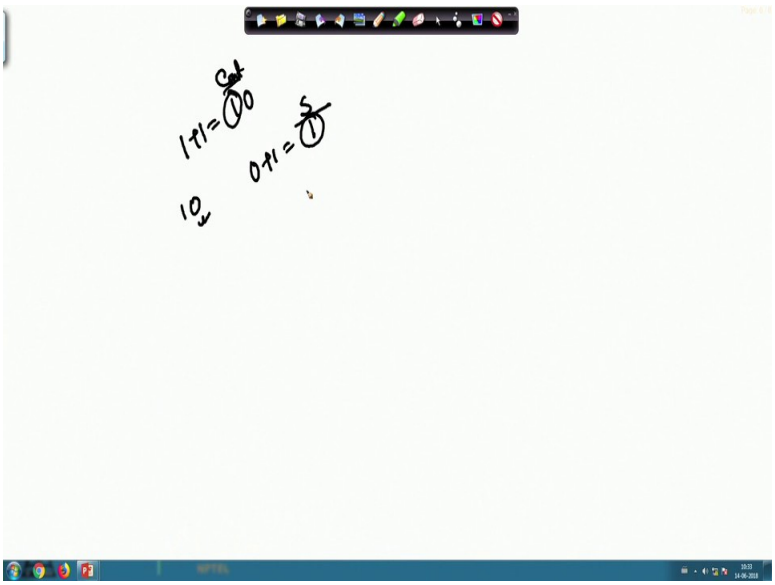
So, this is the; that means, addition operation for a single bit addition half half adder operation ok. Now, if I, that means, consider that I am having the, here what we are seeing; that means, we are seeing that I am adding or I am considering only two particular bit which are A and B, which is basically giving me S sum and C out, now consider one full adder circuit. So, full adder circuit means where I will have, apart from this or except from this A and B input I am having one another C out that means, sorry C in or carry in input bit to the addition operation. That means, so at the time if you just see that this particular table if you see. So, at the time if you see that I am having A and B along with I am having this carry in bit, for here what we are doing we are having 0 0. So, for 0, 0 of A and B I can have the values of carry in 0 and 1.

So, at the time what will be the corresponding sum and C out, if carry is 1 then sum will be 1 other ways carry out will be 0. So, when 1 and 1; that means, for this 0, 1 case or 0, 1 case when carry bit is 0, so at the time this is 1, that means, as in the case of 1, but whenever this for A and B sum of this A and B is 1. Now, carry in is again 1 that means now again I am adding 1 along with 1. So, if am I 1 to or I have to add 1 with 1 so, at the time; obviously, sum will be 0 and carry out will be 1.

So, if I just; that means, if I just go back to the previous slide. So, here you see that means, now 1 for 1 plus 1, I am getting 2. Now, if I just 1 to add what I will get? I will get 3 and if I am considering only these two bit ok, so these two bit if I am considering. So, at that time what are the combination I can get that is 0 1, 1 0 and 1 1, but in this half adder operation I am not getting 1 1 things because I mean I am, I if I add three of one's so at the time I will get 3. So, that means, using 2 bit only I can represent or I can accommodate these three ones which is A and B are two of the that means, input where along with this C in or carry in is the another input which I am considering though I am considering the C out and sum that are of two bit.

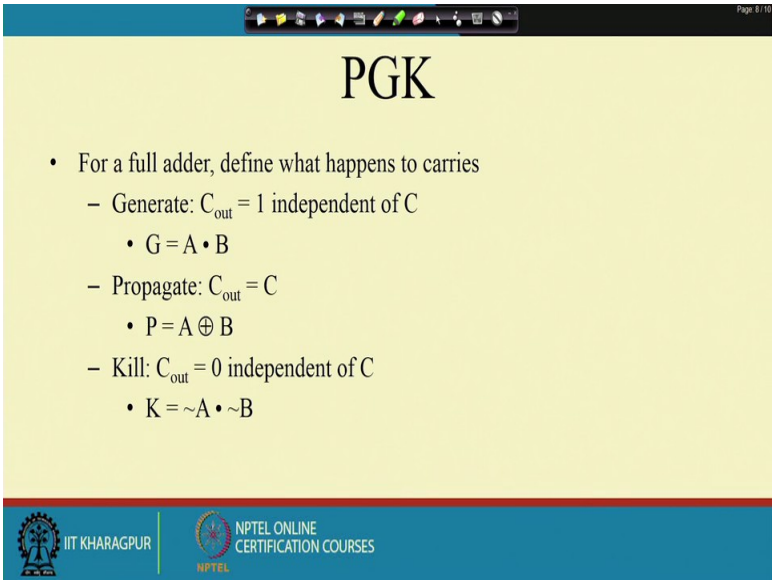
So, again just coming to the, I mean coming to the slide, here you see for this 1, 1, 1 I am getting sum as 1; that means, 1 plus 1 that is 1 and 0, along with 1 and 0 if I want to add 1. So, that will be again sum will be 1 and rest will be 0, why sum that means, what is that?.

(Refer Slide Time: 18:38)



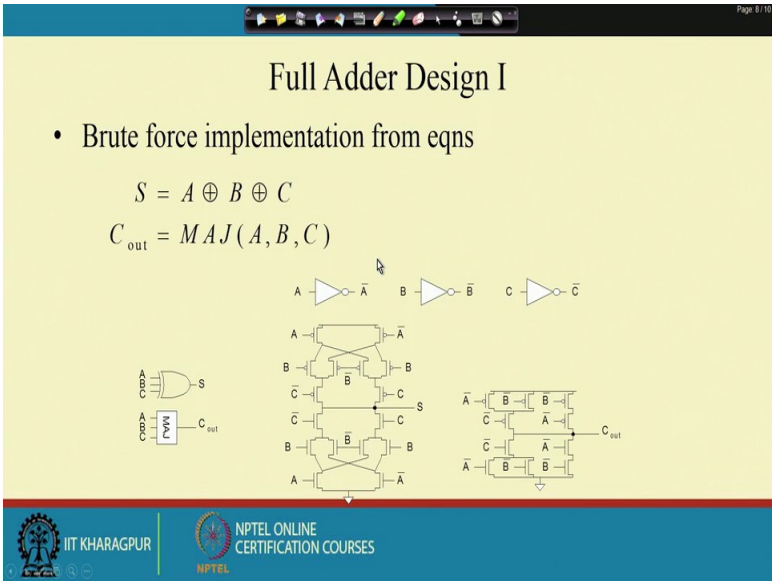
That is 1 plus 1 that is 1 0, now along with this 1 0. So, this is the LSB position, now 0 plus 1 that is equals to 1. So, sum is 1 and this is the carry out bit so that is 1 so here sum is 1, understood.

(Refer Slide Time: 19:16)



So, then if I just go to the, that means, next then for a full adder define what happens to this carry, carry values ok.

(Refer Slide Time: 19:31)

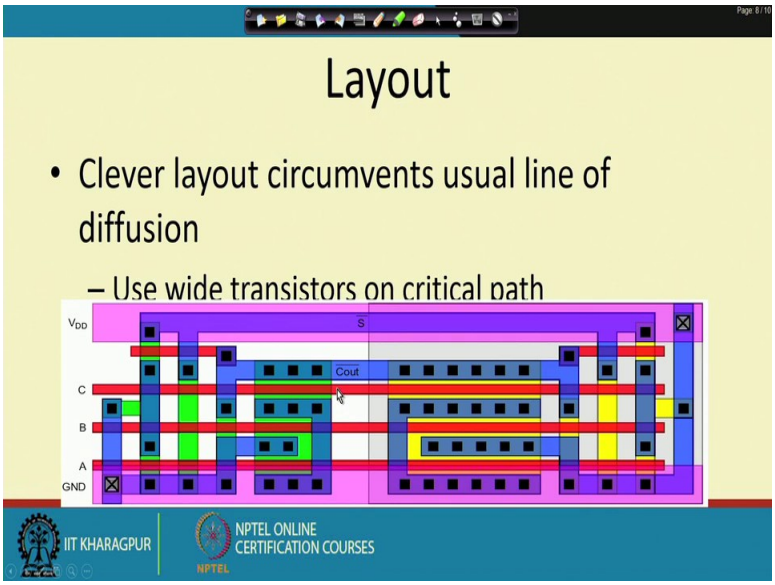


So, actually if I just, I will come to that point later and this is the ; that means, what we know that and if I just go back here. So, here what is the logic for sum that if A XOR sorry if I just that is A XOR B and for C out that is A and B, why A and B? Whenever I am getting 1 and 1 at that time only C out is my 1 ok. So, here also you see; that means, the sum is A x or B x or C ok, for the sum I am getting that.

And for carry ok, for carry the logic if I just put it into a karnaugh map. So, at the time the logic will be A, B plus B, C plus C, A there is another logic for the C out 2 ok. So, that we will come in just after few slides ok. So, how we can implement or how they there is; that means, for this particular c out; that means, I can get these values considering a different logical expression ok, where instead of this A B plus B C plus C A I will get different expression for C out ok. So, why I need to do that that? We will also the discuss at that point ok.

So, then this is the corresponding logical expression. So, this logical expression if I just want to; that means, implement at the time what I need? I need to, that means, implement this function using this gates right. So, now, this gates are basically made by the transistors ok.

(Refer Slide Time: 21:51)



So, this transistors are now, how; that means, it has been connect; that means, how for this A XOR B XOR C what is the logic; that means, this XOR gates are being implemented using this transistor only. And here for A B plus B C plus C A the corresponding transistor level implementation is shown here, ok. So, this is for C out and this is for sum, so here you see how many transistor I require.

So, for this particular; that means, expression I need one sorry, two input XOR gate right, but here how many; that means, transistor I require 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 number of transistors I require for implementing the sum and for this I need 3, 4, 5, 10. So, 10 number of transistor for C out, so total of 16 plus 10 that is equals to 26 number of transistor I require, to implement this particular full adder design ok. But people have ; that means, done the research and they have find out that they how many, how we can because this full adder is nothing, but that is the basic cell, suppose now full adder means just one bit addition operation right.

So, now I want to; that means, perform 16 bit operation. So, 16 of this full adders I need to use. so logic wise I cannot change. So, in the transistor level if I can reduce ok, so this expression is already optimized. So, whenever I am having this; that means, transistor level optimization; that means, for one single bit operation if I can reduce these number on the lower side; that means, if I can; that means, reduce this number, let us say by 4 if I use 16 bit; that means, 16 bit adder design. So, for one bit or addition operation if I can

reduce 4 number of transistor so 16 into 4 means 64 number of transistor I can reduce whenever I will implement 16 bit adder using this particular full adder cell.

So; obviously, I will get area and power consumption; that means, benefit. So, that is why people have tried to design this, they have implanted different architecture, they have different the implemented different this transistor level implementation of this particular logic ok. So, people have tried different different techniques to achieve more on savings on these speed power area ok. So, this lay out, what is layout? Layout is nothing, but the geometrical description of this transistor placement ok. So, whatever transistor; that means, whether NMOS, PMOS and how their source gate drain how; that means that are connected.

So, that is basically informed in this layout form. So, layout is nothing, but this geometrical description of your circuit ok. So, here you see ; that means, there are some of this things, that means, this pink color line pink color line here, then blue, green, red something like this and then black. So, basically this is the part of VLSI design and this p colour line they are the; that means, metal layer lines where it increases this V DD and ground. So, V DD means this is connected to the supply voltage and ground means this is connected to the corresponding that means, 0 voltage or ground voltage ok. And then I am having this A, B, C as the input that are connected over here and then I am having this particular black dots ok.

Which are nothing, but the connection between the layers which are the vias and then I am having different layers of metal which are where somewhere I am having this green, somewhere I am having this like yellow ok. So, these are the different layers of metal which are now connected along with this A, B, C V DD and ground and why and how this is basically connected, basically this is connected based on this particular transistor level implementation. So if, if you see that the source of these and source of this particular transistor they are two of them are connected and then this sorry drain of this drain of this is connected with the source of this C.

So, at the time in these particular layout also we have to connect something like this. So, after doing this, this information will basically send it to the fabricate; that means, from this foundry where this fabrication will be done, they will build this they will fabricate

on the; that means, we can layer or the silicon mask. So, after that you will get the corresponding chip which will work as a full adder ok.

So, and whenever we are doing the layout or we are that means, here we will not consider this layout; that means, this is not the scope of this particular course here we will try to design or try to reduce at the logic level, this transistor level optimization that is also not, we will not be covered in this particular course that is basically comes in analogue circuit design ok. So, here we will particular particularly point on the logical optimization or this algorithmic optimization of different in this arithmetic operation ok.

So, thank you for today's class, we will continue with this particular lecture series on adder architecture in the after classes ok.

Thank you for today.