Digital Electronic Circuits
Prof. Goutam Saha
Department E & EC Engineering
Indian Institute of Technology, Kharagpur

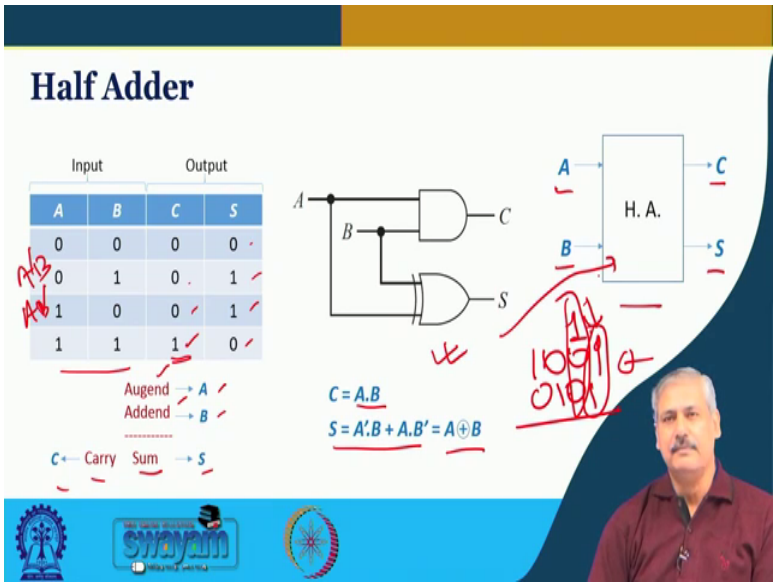
Lecture – 23
Arithmetic Building Blocks – I

Hello everybody. In the last class we had discussed number systems and arithmetic operation: binary addition, binary subtraction, tools complement arithmetic. So, we had understood the theory behind the arithmetic operation addition, subtraction operation using binary numbers. Now, in this particular class we shall look at circuits by which these arithmetic operations can be done. So, it will be extended to next class as well.

(Refer Slide Time: 00:50)

So, we shall discuss half adder, full adder, ripple carry adder and adder subtractor unit in this particular class.

(Refer Slide Time: 00:58)



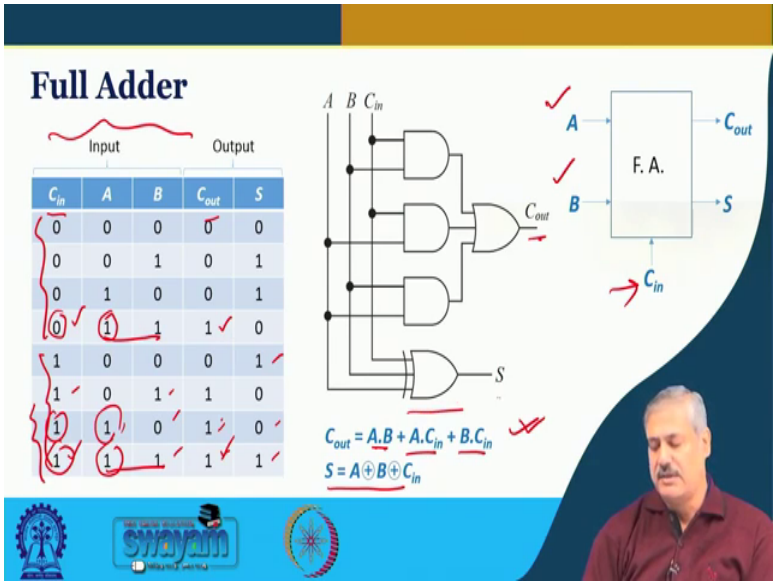
So, first we discussed half adder. Why it is called half? It is because it is does not consider any carry from the previous stage. So, if you are considering addition of 2 numbers is 1 0 0 1 and with another number say 0 1 0 1. So, in the first stage when you are adding this number units place carry is not required from the previous stage. So, there comes the usefulness of the half adder and when in the next stage. So, this is 0 and this is 0 and a carry is generated from the previous stage. So, basically there are 3 inputs.

So, first stage there could be 2 inputs. So, that is what is half adder half addition. So, half adder has got 2 inputs A and B right and it generates data outputs which is one is sum another is carry. So, this is augend, this is addend sum and carry ok. And we designate by A B and A and C the way waves in this particular block and what would be the corresponding truth table. So, 4 possibilities at the input 0 0 0 1 1 0 and 1 1 and the inputs then corresponding output 0 added with the 0, carry is 0, sum is 0. 0 added with 1 carry is 0 sum is 1 1 0 sum is 1 carry is 0 and when both of them are one carry is 1 and sum is 0.

So, how we write it in the form of a logic equation to realize the circuit. So, the carries C is you can see it is simply A and B ok. This is the mean term associated A and B both are having value 1 and sum is obtained sum it is obtained with A prime B and AB prime. So, this is your A prime B and this is your AB prime you sum them up and that is nothing,

but XOR operation. So, this is the circuit which is there is side this particular block ok. This is half addition.

(Refer Slide Time: 03:12)



Now, when you talk about full addition, as I have mentioned before we are also considering the carry inputs. So, then this particular block will be having 3 inputs. So, the basic data input A and B that particular addition process and carry that is coming if it is there from the previous stage right. So, this is the carry in now your separating it from carry out. So, this is input. So, then there are with this 3 inputs there are 8 different possibilities in the truth table.

So, we feel investigate the truth table for carry in 0 we have what we had see in case of the half adder, the truth table simply follows the half adder. And, when carry is equal to 1 then we are looking at you carry is n is 1 and 0. 0 then the output is 1 this is 1 and one of the input is 1, then I carry is united and these sum it is 0. And, when all of them are 1 then carry is 1 and the output is also 1.

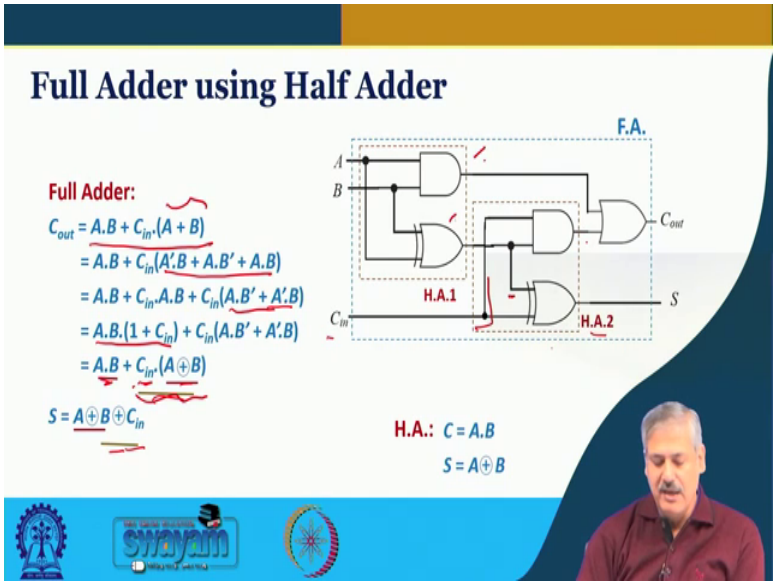
So, how we write a ah equation for this Boolean equation for this. So, we can use Karnaugh map for minimization or you can take the minimum in mean terms and then we do algebraic manipulation. And, by that we can find that carry out is AB AC in BC in ok, if you combine this product terms we get the carry out right. So, just you can see for example, when A and B both are 1. So, A and B both are 1 here and here right at that

time irrespective of carry n is 0 or 1 right the output carry out is 1, you can see when A and C in this particular term A is 1 and C in is 1.

So, A is 1 here and C C in is 1 here C in is not 1 here. So, this is not 1 here. So, C in is here and C in is here this is this are the 2 cases right and both the cases in this 2 cases irrespective value of B you see the output carry output is 1 right. So, accordingly you can find out otherwise you can minimize also using standard process and you can find the carry out is like this. And, what about the sum bit you can again see in this particular truth table when 1 or 3 1 are present in the input combination in this input combination 1 or 3 1 are present, then the sum bit is sum output bit is 1.

So, that is that is nothing, but a 3 input XOR operation right. So, that is A XOR B and they and XORed with in the C in that is what gives you the sum output. So, then the circuit is simple. So, basically this is the 3 AND gates generating AB AC in BC in terms and that are ORed to get C out and this 3 input XORed with that is generating the sum bit.

(Refer Slide Time: 06:39)



So, full adder generation using half adder. So, if you have got half adder circuit made by someone and several such units are available and you want to get full adder what you can do ok. So, we can see that 2 half adders and 1 2 input OR gate is sufficient to get a full adder out of it ok, out of those basic unit. So, how we can get that we can see over here.

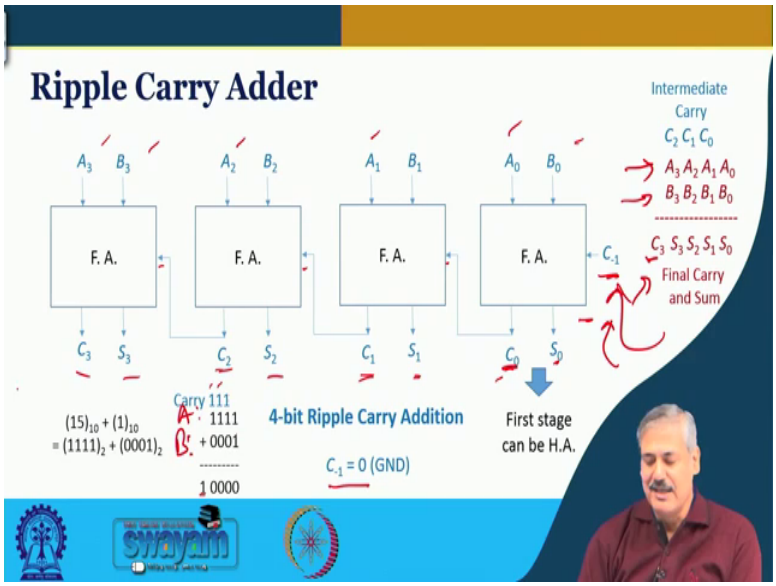
So, carry out we have this is the basic equation we can write it in this form also C in has been taken common and out. So, C in A plus B right. So, this is OR operation right. So, this OR operation includes all the mean terms accept A prime B prime right. So, in A and B both are 0 that time only the or output is 0 otherwise or output is 1 for all other combinations; so, all other all 3 combination ok. Then what you do? You take AB prime and A prime B out over here.

Why? Because, we need the XOR relationship from the half adder to be used in this full adder development, and this C in AB is taken here. So, if you take to common then A B ANDed with 1 or C in that is AB only right. So, we can see A and B are given as input we can generate this term from one half adder this color you can see, this colour right.

So, A B over here and A XOR B we can generate by this right. And, then this the rest of the terms that that is A XOR B which is already generated to be ANDed which C in here and for the sum generation this A XOR B is also ANDed with XOR with C in. So, this is this is can come from the another half adder block. So, this A XOR B these A XOR B output from first half adder that is going to another half adder right, this is half adder the XOR input and C in to get the some final sum output and the other input where A XOR B is ANDed with C in.

So, this C in you can see that is getting ANDed with C in to generate this particular term this particular term right. So, this 2 terms are now available A B and C in ANDed with A XOR B over here and we just need to OR them. So, this is the way we can get it.

(Refer Slide Time: 09:24)



Now, we have seen 1 bit addition right. So, if you want to multiple bit same bit addition for example, say 4 bit addition how we go about. So, in that particular case we need multiple such blocks of full adder. So, inside the full adder we know the circuit that is there. So, that has been seen in the previous slide.

So, this is the full adder slide ok. So, this is the full adder block and full adder block we are using to develop 4 bit addition ok. So, what you can see over here is this, this each of the full adder this is the first stage which I said it could have been half adder also ok. And, then the corresponding inputs are number A with A 3 to A naught as different bits and number B B 3 to B naught 4 bit number a different bits.

So, A 3 A 2 A 1 A 3 A 2 A 1 A naught is supplied to one of the input of the each of the full adder and B 3 to B naught to the other ok. And, carry of the first stage is feed us carry of the second stage, carry of second stage is feed us the carry of third stage and carry of third stage is set as sent set as carry of the fourth stage. So, this is way we normally do the addition and this is the final carry right. So, then this whole thing together it will give you sum bit sum bit these are intermediate carry ok.

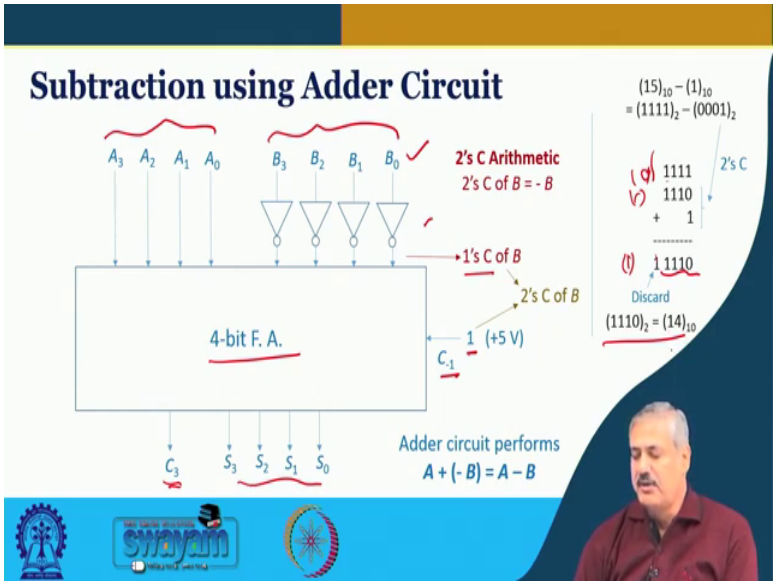
This C naught C 1 C t C 2 or intermediate carry ; so, the final output will be S naught S 1 S 2 S 3 and C 3 ok, the way we have seen it here right. And, if we are using full adder instead of half adder for the first stage we shall make D C minus 1 first carry input 0 right. And, if it is a block which is to be cascaded with another block there is block

proceeding to it than that block carry output which is C 3 will be feed us input here. So, in a previous block carry input will be feed as input to this particular block ok, if that is to be used in cascade there are more than 4 bits to be added and this is the 4 bit unit right.

And if you just look at one example how the addition take place say 1 1 1 1 is your A and B is your 0 0 0 0 0 1 3 0 1. You have taken this specific example you can see right this is the very simple addition, but each of this cases 1 carry is generated. So, 1 plus 1 this is 0. So, carry is there. So, carry this 1 carry 1 and 1 this is 0 another carry is there these 1 and 1 again carry is there. So, this is a final carry. So, the carry which is generated in the first stage C naught right.

So, these again goes as generate C 1 C 1 generate C 2 generate C 3. So, as if it the carry ripples through each of this adder that is why it is also called ripple carry adder addition ok. This is the maximum you know distant it travels; so, as to say maximum path of.

(Refer Slide Time: 12:56)



Now, if we remember how we conducted 2's complement arithmetic. So, subtraction we got by addition ok. So, how did you do that? We got negative number of the original number by taking 2's complement of it. And what was 2's complement? It was in version that is 1 compliment 1's compliment then plus 1.

So, remembering remember that concept of 2's complement arithmetic and we can see how in the hardware we can implement subtraction 2's compliment by 2's complement

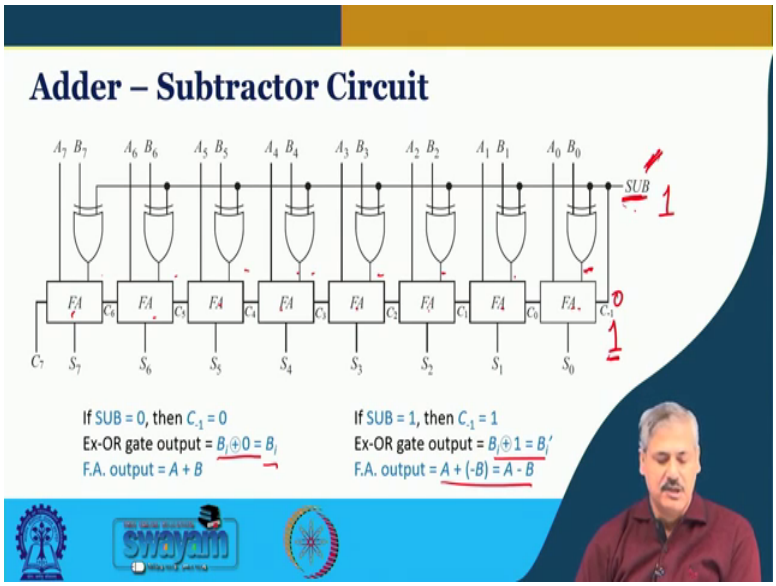
method by using a adder circuit. So, now what we have got this is the 4 bit full adder. So, within it they are 4 individual 1 bit adder and we have already seen before how a 1 bit adder is main using basic logic gates right. So, this 4 bit adder has got this 2 input ANDed (Refer Time: 14:02) and this is the some output and this is the final carry right.

So, the subtrend the 1 that we want to subtract, we pass it through a bank of naught gate inverter. So, at this point we get 1's complements right and then we need to add 1. Now, you can see the C minus 1 we made it the first stage full adder which could have been half adder just for the addition purpose unless it is cascaded also. But, here we can see when we want to use this as a subtractor right then we need to add 1. So, that one can come can we feed here. So, automatically the you do not need to have any additional thing to do. So, instead of this carry input 1 right which is doing the 2's compliment of the number B..

It is used to get the 2's compliment of the number B ok. Then we add then we see the result and we interpret the result based on the carry and other things the way we have seen it before right. And so, how I mean we let us see the example. So, 15 minus 1 right. So, 15 is your 1 1 1 1 we met that the here we are using the this 4 1's. So, the sign bit over here it implied the is the 0 over here and 1 is your 0 0 0 1 right. So, when you takes 1's complement of it. So, this is 1 1 1 0 right and then you add 1. So, what you get over here? 1 1 1 0.

So, this is a carry that is getting generated we shall discard it and if you include considered the you know the in this case the sign bit then this would be 1 and they finally, they will be carry generated right. So, these are implied thing the sign bit over here is implied right ok, as long it is within the range we have no issue. So, this is this 1 1 1 0 is your final answer. So, if you convert it decimal it is 14 right.

(Refer Slide Time: 16:35)



Now, we look at how we can get within the same arrangement, same framework both addition and subtraction. So, for that what we are doing? We are we have placed in example of 8 bit adder subtactor ok. So, 8 bit adder subtractor so, we have got 1 8 full adders right. So, this is 1 full adder 1 2 3 4 5 6 7 8, we can see 8 full adders there there and, the numbers A naught to A 7 and B naught to B 7 right.

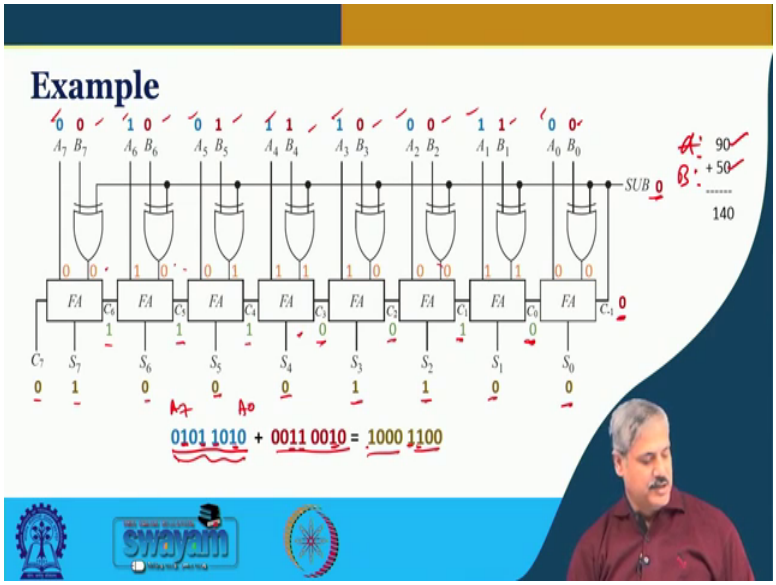
So, this is way the numbers have been placed right. Now, you see one thing that this B number B naught B 7 is passing though a XOR gate and one of the input to the XOR gate is connected to a external input. So, this external input we are representing as sub standing for subtraction right. So, when you put it 0 right; that means, no subtraction is taken place. So, it will be normal addition right. So, when this 0 XOR gate we know, this e XOR relationship any input XOR with 0 it is the input itself right.

So, at this point at the output of the XOR gate right what will come, it will come the number B itself right; B 7 to B naught that will come and then this is 0. So, this will come as a 0 right. So, it will be 8 normal addition standard addition. Now, instead if we have got sum is equal to 1; that means, we are interested in doing subtraction. So, subtraction is when it is 1 of course, this carry so, called carry here it will become 1 and B i XORed with 1 is B i prime.

So, you will get inverted inversion of B over here complement of B 1's complement of B and 1's complement of B and added with 1 we get 2's complement right. So, then these

2's complement get added and then the normal output the we have seen before or we understand how to complement arithmetic takes place. So, it will be A plus minus B that is A minus B. So, this is a single adder subtractor unit which is performing both the job of addition as well as subtraction depending on what you give here as input. If input is here is 0 this is addition, input here is 1 it is subtracted right.

(Refer Slide Time: 19:26)



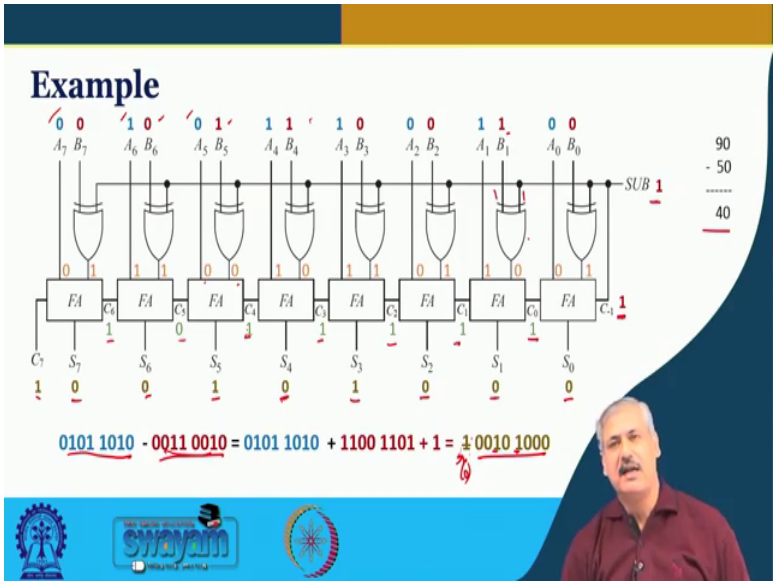
Now, we shall look at any 2 examples how it happens in the hardware. So, we look at a addition example to start with. So, the numbers that you want to add is 90 and 50 ok. So, in decimal of course, and then the corresponding binary equivalent you have to get. So, this is your 90 this is your A and this is your B right. So, this is your 64 right this one represent, this is 16 64 and 16 80, this is your 8 right 64 plus 16 80 and 80 and 88 this one is representing 2.

So, 90 right so, this particular number is 90. So, this we will put as A 7 to A naught. So, A 7 to A naught this how this is blue one that you can see is 7 0 1 0 1 1 0 1 0 that is how it is placed, is it and then 50 right. So, 50 for 50 what we are doing we have to convert first it will binary. So, this is your 32 plus 16 that is 48 and this is your 2 that is 50 right. So, this is in a different color. So, this is your B B 7 to B naught. So, 0 0 1 1 you can see 0 0 1 1 and 0 0 1 0. So, this is what is placed here right and since it is addition operation. So, this input over will be 0, this carry input here 0 and what will be the corresponding XOR output.

So, this is 0 the same number will come over here 0 0 1 1 and 0 0 1 0 and for the other input of full adder will be the same as what is the original number is of course, because there is no such case of inversion. Now, if you perform the addition you can see what is happening intermediate carry and the finally, carry that will be generated and the sum bits. So, 0 0 0 over here so, this intermediate carry output is 0 and sum is also 0. So, this is 0, this is 1 and this is 1 right. So, sum is 0 this carry is 1 so, this 1 0 0 for the full adder. So, this is 1 and carry is 0. So, 0 0 1 that gets added. So, sum is 1 carry is 0; so, 0 1 1 for this particular full adder for this particular full adder. So, sum is 0 and carry is 1.

So, this carry 1 1 this to some bits 2 bits ANDed or added bits so, 1 and 1. So, this is 0 and carry is 1 this 1 0 1 this 3 getting added. So, this 0 this is 0 and 1 0 0 this is finally, you 1 and the carry is 0. So, this is the number that will get 1 0 0 0 1 0 0 0 1 1 0 0; what is the value right. So, this is your 64 after that it is 128 this is 8 36 and this is 4 40 on 40 clear ok. So, we shall look at one subtraction example right.

(Refer Slide Time: 23:07)



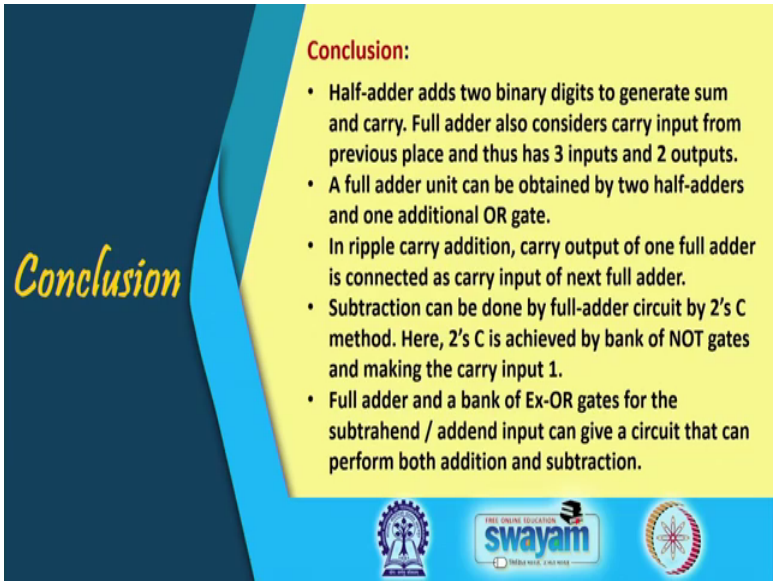
So, this same 90 and 50 is the original A and B which start with so, that it becomes easier to understand. So, this is your 90 and this your 50 right. So, A is placed as 90 this is the blue colored one that you can see right and 50 is the brown colored one. So, that you can see over here that is where how the 50 is placed. Now, the subtraction is input is 1 because you want to do subtraction. So, the carry in over here for the first adder is 1 ok.

So, this because of this 1 there will be a inversion taking place for the XOR gate because, 1 and 0 XORed is 1 right 1 and 1 for the second one. So, this is 1 and this is 1 XORed is 0. So, this is the way for the rest of the things right. So, this is the how the full adder input will be now looking like for each of these full adder. Now, if your perform the addition so, this is 1 and 1 and this is 0 right. So, you just add them up. So, this is 0 and carry is generated as 1 ok. So, this is 1 0 1 this 3 are added so, this sum is 0 carry is 1.

So, 1 1 sum is 0 carry is 1 3 1 sum is 1 carry is 1 1 0 1. So, this is 0 carry is 1. So, 1 0 0 0 1 0 0 0 sum is 1 carry is 0 then this 2 are 1 and 1. So, sum is 0 carry is 1 and this is are 2 1 1 sum is 0 carry is 1 is it fine. So, this is the number. So, this 1 that is you are getting here we discard it and this is what you get and the answer we know that it is negative. So, basically this is representing that the number is sorry number this 1 is getting discarded ok.

So, if you have a the concept of sign bit so, this 1 will add will that sign bit. So, this will become 0 and there will be a 1 here, that 1 will get discarded and this value will be 0 sign bit will be 0 right; if we include the implied sign bit here right. So, this is the answer. So, answer here what we see, this is 32 right and this is 8 32 plus 8 this 40 clear. So, this is how we do addition and subtraction and from the basic circuit that we have seen.

(Refer Slide Time: 25:59)



So, to conclude half adder adds two binary digits to generate sum and carry. Full order also considered carry input, full adder also considered carry input from previous place

and thus has 3 input and 2 outputs ok. A full adder unit can be obtained by two half adders and one additional OR gate. In ripple carry addition, carry output of one full adder is connected as carry input of the next full adder. And, subtraction can be done by full adder circuit by 2's complement method. Here, 2 complement is achieved by a bank of NOT gate and making the carry input 1. And finally, a full adder and the bank of Ex-OR gates for the subtend or addend input can give a circuit that can perform both addition and subtraction ok. So, this is what you have seen in this particular class.

Thank you.