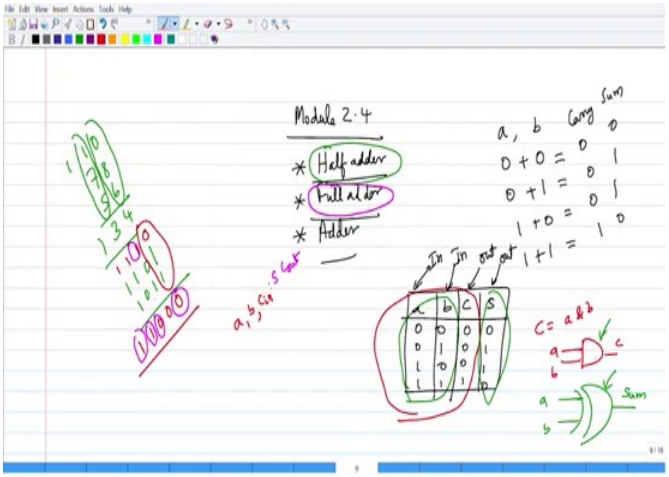
Foundations to Computer Systems Design
Professor V.Kamakoti
Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Module 2.4
Binary Adder Circuits

(Refer Slide Time: 00:17)



So welcome to module 2.4, so in this module 2.4 we are going to talk about certain logic circuits so we will talk about three circuits one is called the half adder then a full adder and then adder ok. So we will see these three circuits. Now these three will be using it as a part of hardware simulation also. Now what is an half adder? I am adding two bits and what is the answer I could get? So there are two bits a, b when I add 0 and 0 the answer is 0, when I add 0 and 1 the answer is 1, 1 and 0 1 so 0 plus 0, 0 plus 1 but when I add 1 and 1 what you get? I get 1 and 1 is 2, 2 is 0, 1 so I get a 0 and another 1 as a carry.

So this four bits are called sum bits and the remaining things are carry this ok, it is also quite useful for our reading if I put the carry bits first and the sum bit next so this will give 0 0,0 1,0 1,1 0 right so(so) basically this gives us this interesting table a b carry sum, 0 0, 0 1, 1 0, 1 1, 0 0, 0 1, 0 1, 1 0 right. Now if you look at how a b and c are working this two are inputs and this two are outputs if you just view this part of the truth table alone a, b and c you will understand that c is equal to a and b in such a AND gate this is exactly the truth table of an AND gate.

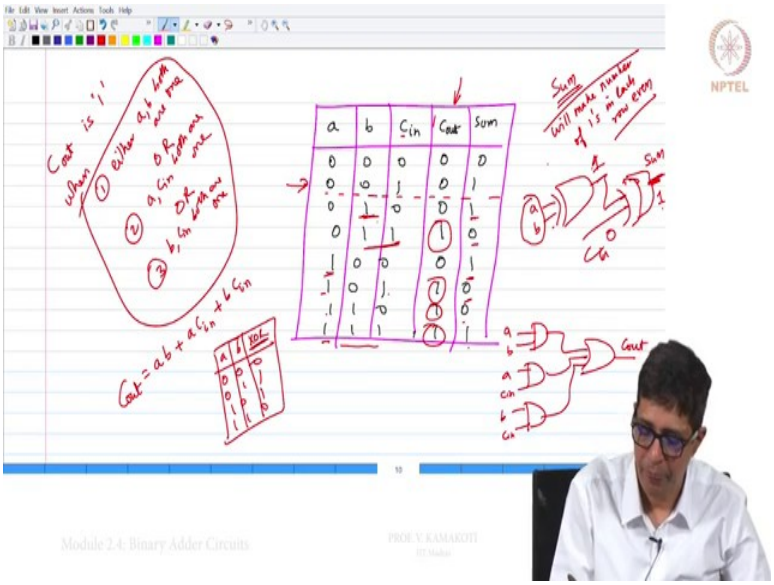
AND gate was introduced in the previous module right this is a truth table. Now if you look at a, b and s (a, b and s) alone you will now understand that this is a EX-OR gate so a b EX-OR we have the EX-OR also in the previous module so this is the sum right. So what this so this circuit is nothing but what we call as a half adder, so what does an half adder do? It takes two bits and it adds and gives the carry bit and the sum bit. The carry bit plus sum bit put together will give you the result of this addition right.

Now what is the circuit involved a and b are the inputs bits c and that is the output bits C is a AND b so use an AND gate to generate the carry and use an EX-OR gate to generate the sum. Now for all people who have just getting to understand computing please understand that this how logic gates namely AND gate, EX-OR gate that we introduced earlier is now used for doing arithmetic computation so this is how the logic allow maps on to arithmetic or arithmetic maps on to logic that is a (better) correct way of put. So I am doing an arithmetic and basically trying to add 2 bits and this 2 bits I am mapping it on to 2 logic gates here right so this is how an half adder is constructed.

Now when we do addition so when what did we do? 78 plus 56, 8 plus 6 14 I take 1 carry above here 7 plus 1 8 plus 5 13 I take another carry and bring it here. So what is after all in addition? This is an addition with 0 as a carry this is an addition with 1 as a carry. So normally when we add we do not add a two digits here actually we try and add three digits at every time right because carry is also one digit. So this is in the case of binary in the case of decimal the same thing is true in the case of binary. So if an looking at adding 1 1 0 0 plus 1 0 sorry I will make it little more interesting 1 1 0 1 plus 1 1 1 1 so initially this 0 is the carry now I will do the computation with red, 0 is the carry so 0 plus 1 plus 1 is I get something 0 1, 1 plus 1 is 2 that is something 0 1, 1 plus 1 is 2 the sum bit is 1 plus 1 plus 1 is 3 so 1 and this the carry.

So that means every time I add a, b and the carry in and what it generate out is every time I add 0 1 1 I generate a sum bit and the carry bit the 0 1 1 generated a 0 and a 1, this 1 0 1 now generated a 0 which is a sum bit and the carry bit which is 1, is 1 1 0 generated a sum bit which is 0 and the carry bit that is 1, 1 1 1 actually generated a sum bit and a carry bit which is 1. So I give three inputs and I get two outputs which is the sum and the carry out right. So this is basically the full adder circuit. So now so how do you design the full adder circuit?

(Refer Slide Time: 6:47)



Let us go so will rather to complete this is called the truth table so I have a, b, c in c out and sum so I will put 0 0 0, 0 0 1, 0 1 0, 0 1 1,1 0 0, 1 0 1, 1 1 0, 1 1 1, 0 plus 0 plus 0 is 0, 0 plus 0 plus 1 is 1, 0 plus 1 plus 0 is 1, 0 plus 1 plus 1 is 2, 1 plus 0 plus 0 is 1, sorry (1 plus 1) 0 plus 1 is 2, 1 plus 1 plus this 2, 1 plus 1 plus 1 is 3 ok. So when we look at this entire thing we will find out that this one of the interesting thing is your carry is 1 when either a and b your carry is 1 (carry is 1) that is carry means C out, C out is 1 when either a, b both are 1 or a, c in both are 1 or b, c in both are 1. One of this three condition have to be satisfied.

So let us see when the carry is 1, the carry is 1 when b and c are both are 1 or a and c are both are 1 or a and b are both are 1, since the (())(9:06) it can be a and b or b and c or here this condition is also satisfied. So there is the thing. So I can write C out as a, b plus a C in plus b C in ok. So now if you just (use) this is under a and b or a C in or b C in so the circuit for this would be a C in ok so you can, to construct this circuit what the half adder and the full adder you can use the half adder and the EX-OR gate that we have done as part of project 1 again I repeat please complete project 1 before doing this then only you will appreciate what we have done and you can map things so please spend that one hour (())(10:21) and do that project 1.

Otherwise this will not be that effective right so (this is) after all we are asking you this one 20 hours totally for this course and you learn lot of computer science so I think it is worth it so kindly do that project 1 again I repeat it is very important so once you have done project 1 you

will appreciate how to generate the circuit this is going to be very straight forward we have two outputs and you know you have done so many Mux, De-Mux etc so just use AND gate and X-OR gate which you have done already as a part of project 1 use it to generate an half adder.

Similarly when you are doing a full adder so this is very easy right you just basically use three AND gates again and we have already done and AND was done using NAND so that AND gate which is in turn realize that NAND as you see and then you can use this three way or OR so multi way gates we have seen this is a three way OR so we can do this to realize this circuit. Now interestingly we will also find out that sum will make number of 1's in each row even, right? So let us take row 1 how many 1's are there 0 1's are there? Sum also remains to be 0 to keep the whole thing 0.

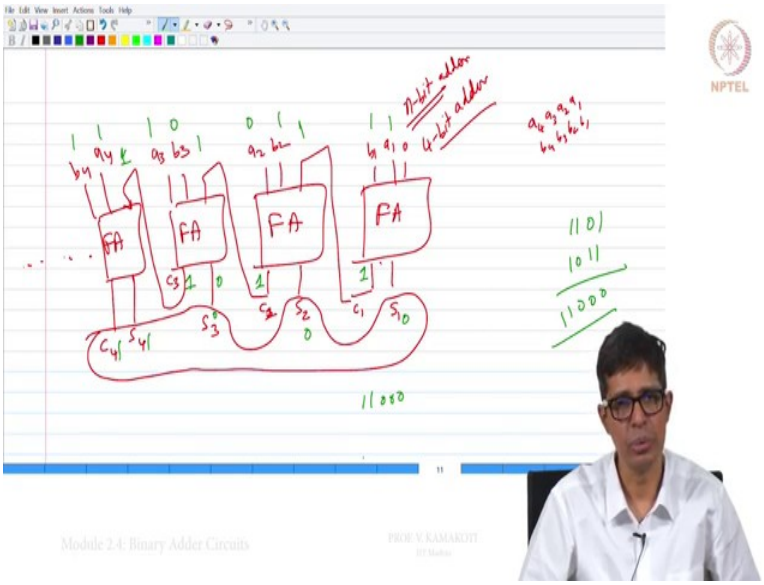
In their second thing how many 1's are there in the second row that I am seeing here? How many 1's are there? That is 1 1 here so the sum so don't consider C out just among the inputs there is only 1 1 so sum becomes 1 to make the total thing to similarly in third row we have 1 1 so the sum also becomes 1 to make it two, in the fourth row we have two 1's so the sum becomes 0 to make it even. The fifth one that is 1 1 so it is and in seventh one there are already two (eight) sorry sixth and seventh there are already two so remains at 0 and the seventh one there are three 1's so this remains 1 right, so how do you realize this?

You basically realize it is a EX-OR b, a EX-OR b will again try to make it even right if you take the EX-OR gate a, b this is a EX-OR of b so 0 0 is 0, 0 1 is 1, 1 0 is 1, 1 1 is 0 so the output of this EX-OR will try to see that the number of 1's in every row is here right. So I will say a EX-OR b and this I will take it to EX-OR c and this is sum. So when this will be 1? This will be 1 if this is 0 and this is 1 or this is 1 and this is 0 if this is 0 and this is 1 that is case 1. If this is 0 and this is 1 that means a and b have even number of 1's and c is having an extra 1 C in ok C in is having extra 1 so total numbers odd numbers of 1's and hence the answer sum will make it even.

And similarly if this is 1 the other reason is if this is 1 and this is 0 there you will find out if this is 1 then a and b has odd number of 1's that is one of them is1 and C is 0 so that means out of a, b, c there are odd number of 1's and so the sum will become sum is extra 1 which will make it even ok. So your sum with is nothing but a EX-OR b that output fed to C to get this. Now using the EX-OR gate, AND gate and OR gate of our previous lecture you would be in a position very

easily to construct a full adder and what will the full adder do it will take three bits and generate a sum bit and a carry bit.

(Refer Slide Time: 15:10)



Now with this how do we construct a n-bit adder or four bit adder? So we will talk of n-bit adder but for a special case let us say we want four bit adder. So take full adder so what are the four bits to the adder? To four bits let me take a4, a3, a2, a1 and b4, b3, b2, b1 so this is full adder let me say full adder I will make four full adders right. So four full adders and here I will put a1, b1 and 0 as input so I will get a sum bit and a carry bit let me call it C1 which goes here and a2, b2 I will get a sum bit and another carry bit which goes here then I put a3, b3 I will get another sum bit another carry bit goes here then I put a4, b4 so I get another sum bit and a carry bit ok.

Now c4 this are the five bits this is the answer. So automatically now I have constructed a substitute which will add four two 4-bits numbers right I can keep extending it the same thing (())(16:48) and please note that this exactly does however addition happen. So we gave some example of an addition right in our previous thing 1 1 0 1 plus 1 0 1 1so let us say I want to do 1 1 1 sorry 1 1 0 1 so the a is 1 1 0 1 and b was some 1 0 1 1 (1 0 1 1) 1 1 0 1 plus 1 0 1 1 so 1 and1 I add 1 and 1 so I get 0 this goes as 1, 0 1 1, 1 and 1 this is 1 ok 0 1 1 will give me again 0 1 goes here 10 1 will give me again 0 1 goes here 1 1 1 will give me 1 1 so the answer is 0, 1 1 0 0 0 this is what happens here 1 plus 1 is 0, 1 plus 1 is 0(1 plus 1 is 0).

So this is how you construct a full adder from (four) sorry construct an n-bit adder from n full adders we have constructed a 4-bit adder with four full adders in the project 2 that will be doing (())(18:30) you may construct a 16-bit adder also ok, so in this module we have seen three circuits namely full adder, half adder, full adder and then adder and the this will be asked to do as a project right. So will continue with an ALU will be constructing an ALU in the next module, thank you.