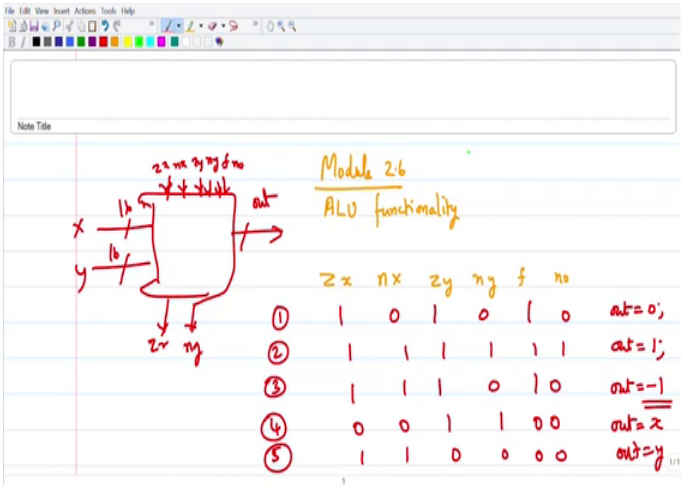
Foundations to Computer Systems Design
Professor V.Kamakoti
Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Module 2.6
HACK ALU Functionality

(Refer Slide Time: 00:23)



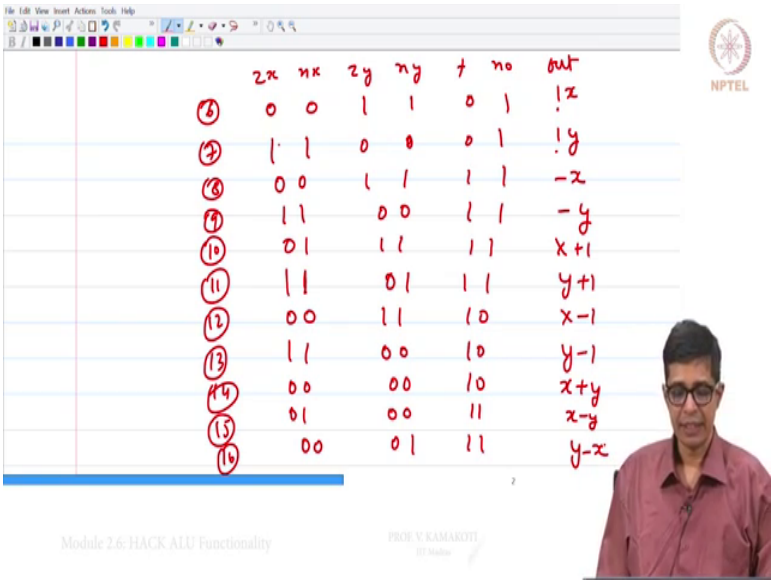
Welcome to module 2.6 and we will be talking about ALU Functionality as we mentioned last time there are multiple value functions that need to be done and that will be a combination of six control signals namely Z x, n x, Z y, n y f and n o just to recap Z x will make x as 0 Z y will make y as 0 n x will make whatever x it is inverted n y will make whatever y it will invert it all this will happen when they are set to 1. For example when Z x is set to 1 it is a single bit if it is set to 1 then it will make x 0, n x if it is set to 1 it will invert whatever is there on x.

Similarly Z y, n y, f if it is set to 1 it will add if it is set to 0 it will do an (())(1:15) and similarly n o if it is set to 1 will invert the final output, output of this f ok so this is the definition we have already seen this definition in the thing. So using this we need to basically do lot of functions atleast close to twenty of them. Let us see one by one on this, so the first function is I am making 1 0 1 0 1 0 so the output will be we will generate a 0 as an output this you can go and check if I do 1 0,if I make it as the second one is if I make all as 1 1 1 1 1 1 the out will be 1. So this are the ways by which I could generate some constant like 0 1 etc.

So the next one is 1 1 1 1 0 1 0 and that will give me minus 1 this minus 1 already we explain it will be in 2's compliment format and this is a 16-bit architecture right so minus 1 will be in the 16-bit format whatever is for minus 1. So what will be minus 1? It will be 1111 1111 1111 1111 this will be minus 1 in 2's compliment format. Now the fourth function that we can implement is 0 0 1 1 0 0 the out will be actually x so this is the ALU defend like this right, so x and y each of 16-bits they are same thing to an ALU with this six signals Z x, n x Z y, n y f , n o and you get an out and ofcourse you get two flags whether it is a zero or not greater than.

This are all control signals that are coming into the system (this are control). So when I put 0 0 1 1 0 0 please note that I will be just passing x outside just passing of x. Similarly the fifth function 1 1 0 0 0 0 out will be equal to y so we have seen six functions here now will see some more functions.

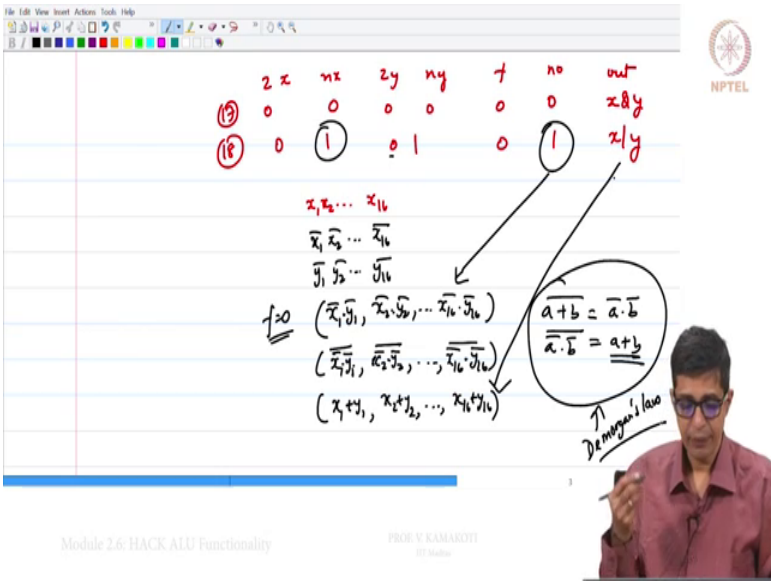
(Refer Slide Time: 4:26)



So let me just write it again. Z x, n x, Z y, n y f, n o the next function the output will be not of x, 00 11 11 this is minus x minus y so this is this 6th, 7th, 8th and 9th functionalities this is the 10th functionality 01 11 11 that is x plus 1, this will be y plus 1 automatically the x minus 1 this will be y minus 1 this will be x plus y (this is) x minus y this will be y minus x.

So we have seen sixteen different functionalities by just varying just two x and y and (())(6:05) Z x, n x, Z y, f n o out and thing get to all this very interesting things right. Now we will just see two more functions.

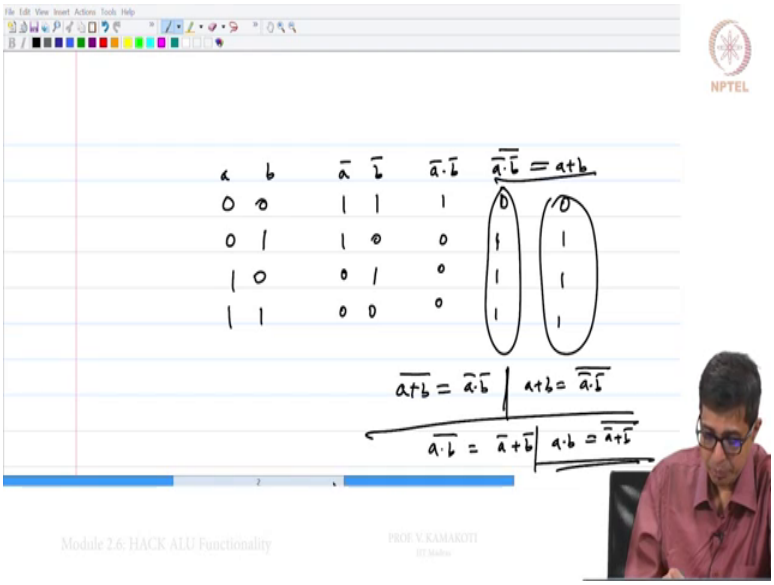
(Refer Slide Time: 6:28)



So let all be zeros for then I put Z x, n x, Z y n y, f n o output we have seen till sixteen functions here there is last one 17th last but one this will be x and y and (eighteen) 01 01 01 will be x or y so how this x or y? Just will see so x is a 16-bit number n x will make it as so if I give x1, x2 till x16 since n x is 1 this will become x1 bar, x2 bar till x16 bar.

Since y is 1 it will become y1 bar, y2 bar to y16 bar and since f is 0 you get x1 bar dot y n bar, x2 bar dot y2 bar, x16 bar dot y16 bar. Again since n o is 1 you put this will become x1 bar dot y1 bar the whole bar, x2 bar dot y2 bar the whole bar like x16 bar dot y16 bar the whole bar. So a plus b the whole bar is equal to a bar dot b bar that means a bar dot b bar the whole bar is equal to a plus b so this will give me x1 plus y1, x2 plus y2, x16 plus y16 right because nothing but(()) (8:39). So this law this is called Demorgan's Law and we will show it with a simple example how Demorgan's Law work.

(Refer Slide Time: 8:56)



A, b, a bar, b bar, a bar dot b bar, a bar dot b bar the whole bar and a plus b, so 00 01 10 11 (()) (9:11) 11 10 01 00, a bar dot b bar 1 0 0 0 this double bar (1) 0 1 1 1 a plus b is also 0 1 1 1 now note these two are equal. So this is called a plus b the whole bar is equal to a bar dot b bar or a plus b is equal to a Bardot b bar the whole bar. Just put a is equal to a bar here, a bar the whole bar is equal to a bar plus b bar. In other words a dot b is equal to a bar plus b bar the whole bar ok. So this is how we do multiple theorems here so we have done something like eighteen different functionalities and this is how this works and this is part of your project P2 and I hope all of you have done P1 and now we have also finished P2 this would be the ultimate goal of this particular module 2.6, thank you!