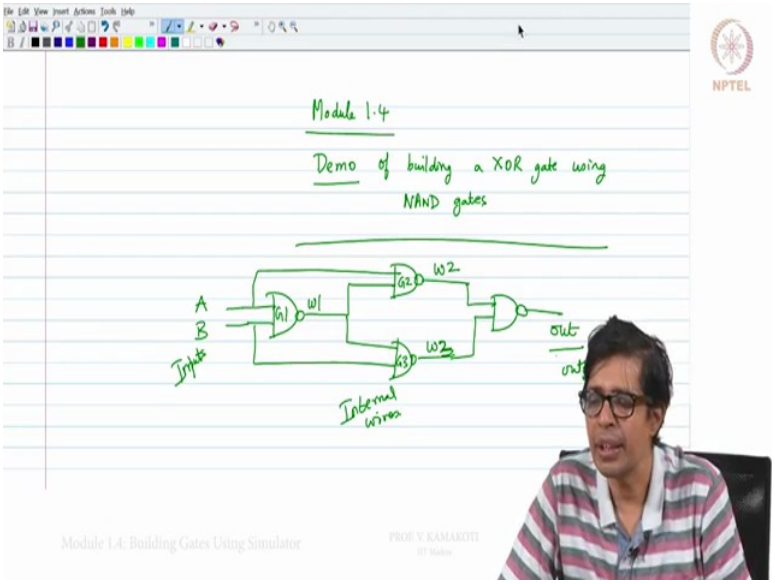
Foundations to Computer System Design
Professor V. Kamakoti
Department of Computer Science and Engineering
Indian Institute of Technology Madras
Module 1.4
Building Gates Using Simulator

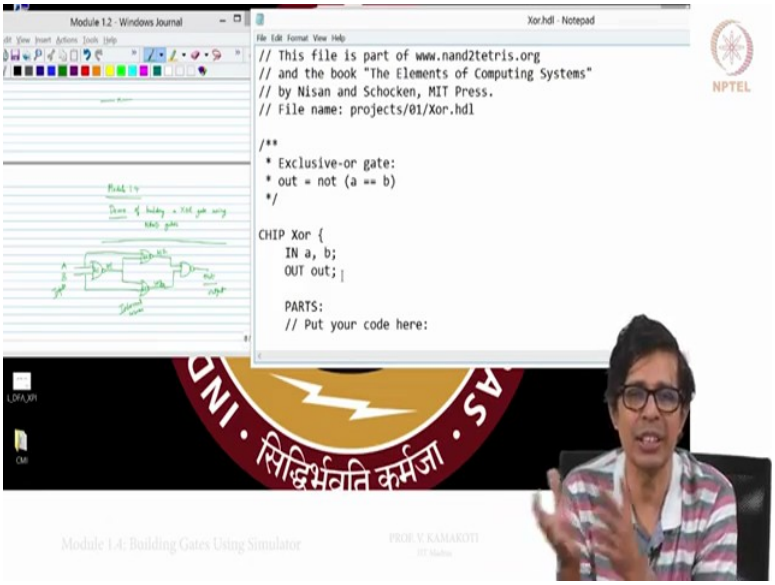
(Refer Slide Time: 0:18)



So in this module we will now go and do a demo of building a XOR gate using NAND gates. Now we know that this is how the… Let me assume that A and B are the inputs and Out is the output. Now let me say this is wire that is a wire one that is coming out of this particular NAND gate G1, this is G2, this is G3 there is a wire that is coming out of this NAND gate G1, this is w1 and there are wires coming out of this NAND gate G2 and G3 let me call it a w2 and w3 and Out. So this particular circuit now has 2 inputs A and B, 1 output, so 2 inputs A and B, 1 output out and 3 internal wires namely w1 and w2 and w3 okay. So let us just basically… Let us keep this in the background and now let us go and do this particular work building of an XOR gate.

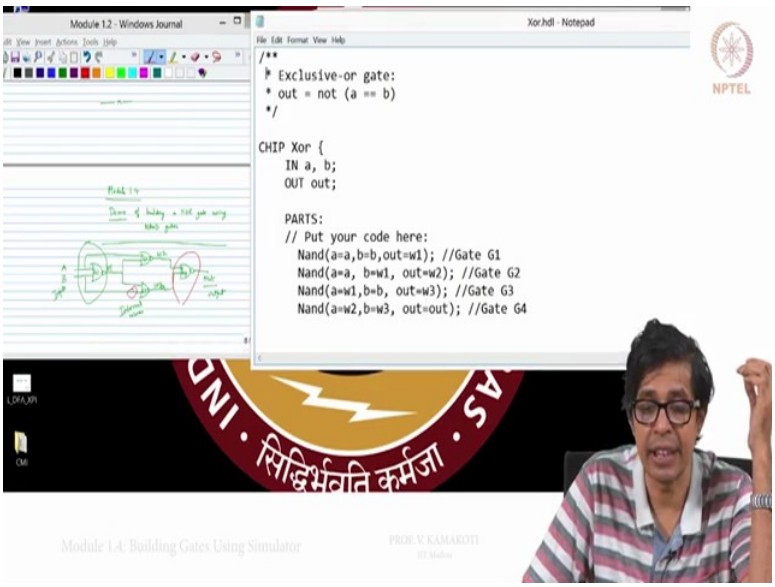
So let us go back to where we have installed, this is basically the tools and let us click on this hardware simulator, the hardware stimulator comes up and we will minimise this and keep this here side-by-side and now let us look at… Let us go back to this…here let us go to the NAND to (())(3:04) go to projects 01, let us open XOR project 01 let us open XOR.hdl let us use it and open it.

(Refer Slide Time: 3:37)



So let me just explain you this file there are other assignments that you have to do but I will just do this here, right? So now we have to build an XOR chip, so the inputs are A, B the output out is out, so this is how you describe your circuit to the system, right? So this is the way of describing the hardware. You are not actually going to build hardware per se but you are going to make a model of the hardware which you simulate, so this is equivalent to the hardware but this is a simulation of that hardware, right?

(Refer Slide Time: 4:25)



So now basically you have to build a XOR gate here and the inputs are A, B and the output is Out and now you have to describe how this code is going to work. So what you do here, first thing is A is a, B is b, and out of this is w1 this is nothing but your gate G1 that you see on the

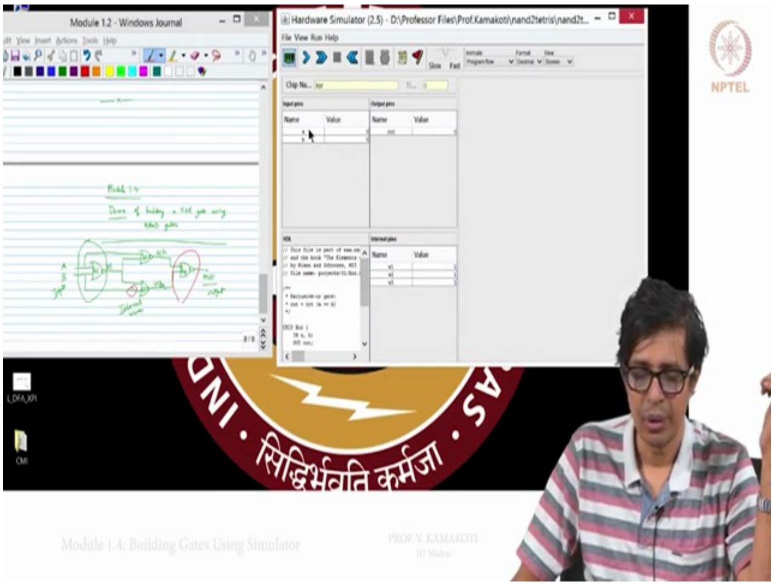
other part of your side gate G1. Now we will do the next NAND gate for which your a is a and your b is w1 and your out is w2.

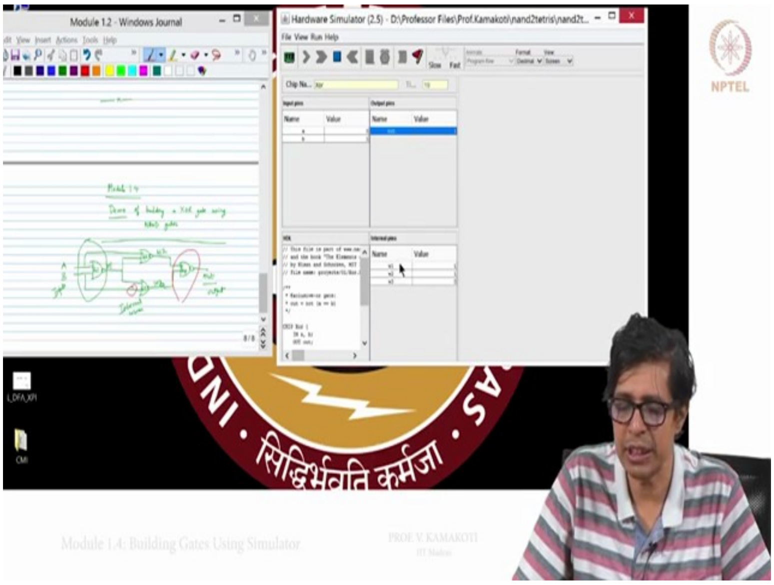
This is gate G2 NAND again a is b actually for w3 as you see here your a is b or a is w1, b is b and your out is w3 this is your gate G3, right? and the last gate NAND a is w2, b is w3, out equal to out. This is gate G4, so you have defined your chip your hardware like this wherein there is an XOR with x a and b as input as I am showing you here a and b as input and out as output. This is capital IN this is capital OUT, whatever is following this capital IN are the inputs a, b, what is following the capital output is the output designated buy out, this is the name of your chip called XOR, okay.

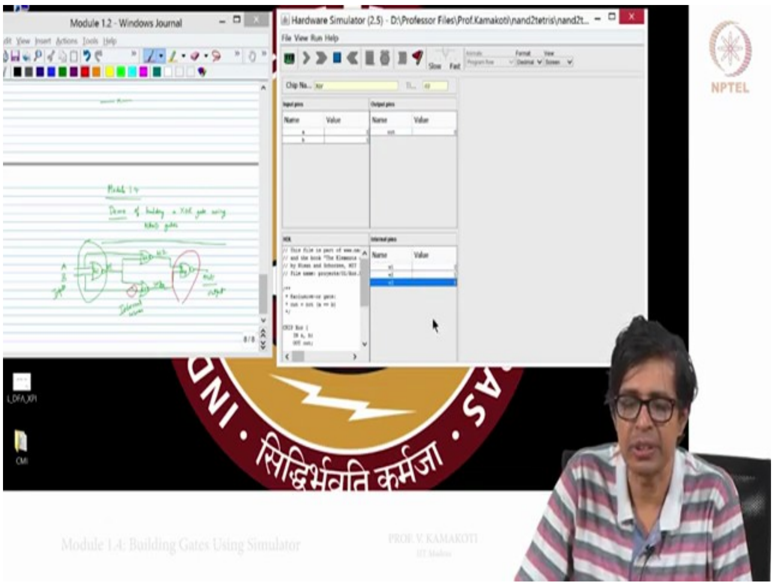
Now what you are doing here NAND a equal to a, b equal to b, out equal to w1 gate G1, a equal to a, so now we have to put 4 NAND gates here. The syntax that we follow here is that this NAND is prebuilt or you know how to make a NAND gate we have shown in the course. Now knowing how to build a NAND gate, how will I go and use that to build an XOR gate that is what we are going to see here. Now this is the syntax, for every NAND gate there are 2 inputs a, b and there is an output out.

How are those 2 inputs connected is what we are saying, for the gate G1 that 2 inputs a and b are this a and b that you see here and out is w1. As you see on the other side for the G1. The gate G2 takes a and w1 as an input and gives w2 as an output. Gate 3 takes w1 and b as an input and gives w3 as an output and the gate 4 which is G4 here, this is G4, the gate G4 basically as I am showing here this is G4, the gate G4 basically takes w2 and w3 as input and give out as output, so this is the thing. Now I go I save this file so this is XOR.hdl you have save this file and now I will quit this.

(Refer Slide Time: 9:34)



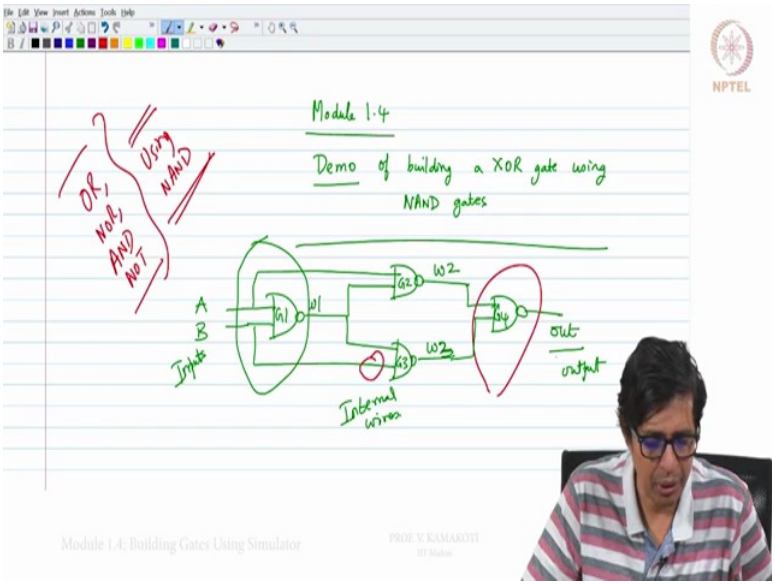


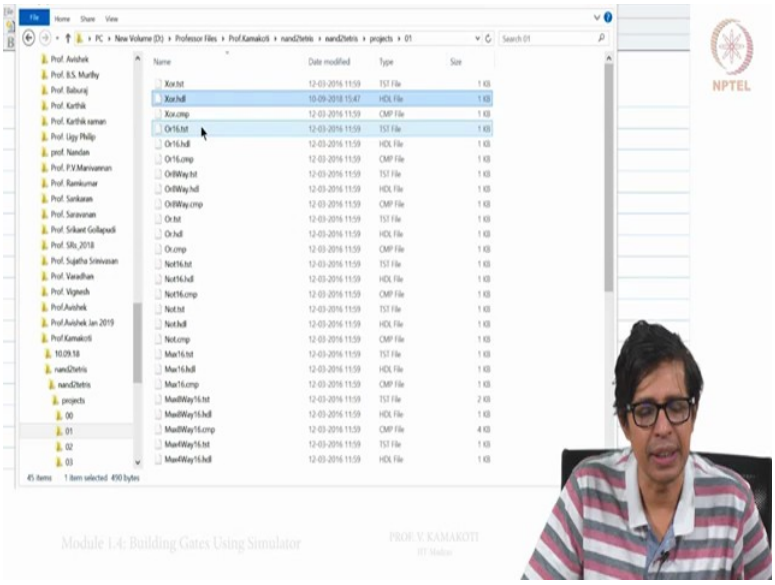


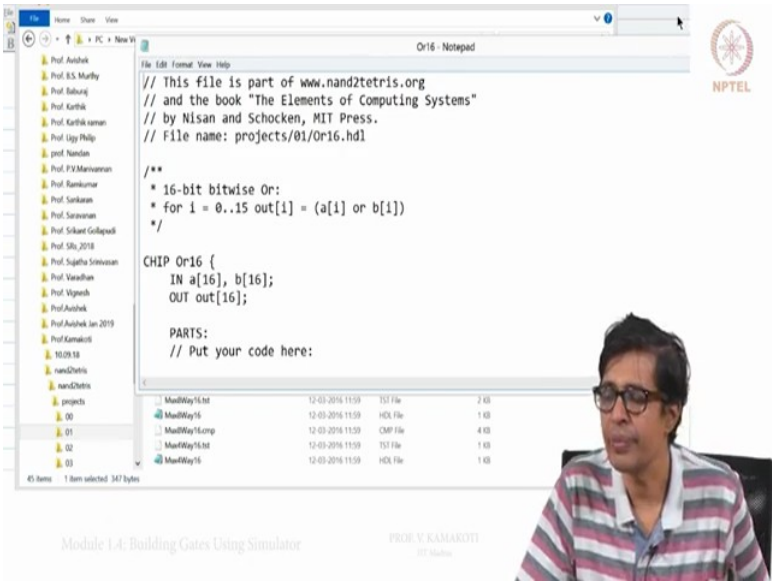
Now I go back to the hardware simulator and I first loaded that file so I say load chip and this is in project XOR I am loading it. Now you see that your entire chip has got loaded, you see this part down here your entire chip has got loaded. Now you can give the input and see, so I am giving a and b as input I am running the program and I am getting 0 as output here, now I can give a is 0 and b is 1 the next input enter, now I can run the program now you see 1 is the output.

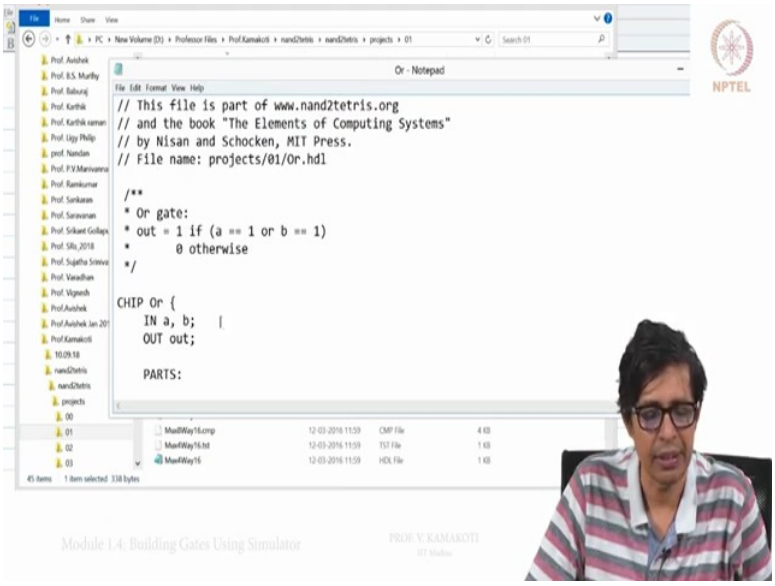
Not only you see that you also see what is the value of w1, w2 and w3 when you apply 0 and 1 as inputs. Now I am making 1 0, 1 and this should become 0 press the enter key and now I run the program I get 1, so 1 0 is 1. Now I make 1 1 and I run the program I get 0. Not only that I also get the values of the 0, 1, 1 so this whatever you are seeing here is a hardware simulator. Generally when you design hardware you cannot actually go to a fabrication unit and make that NAND gate assemble this and bring this is not possible, right. In a lab you cannot do this you do not have a fabrication unit associated with a lab so you can just do it on your system itself, so whatever you are seeing here the hardware simulator that time just moving the cursor upon will actually mimic how the hardware will actually work, so in this hardware we have today mimic the XOR gate.

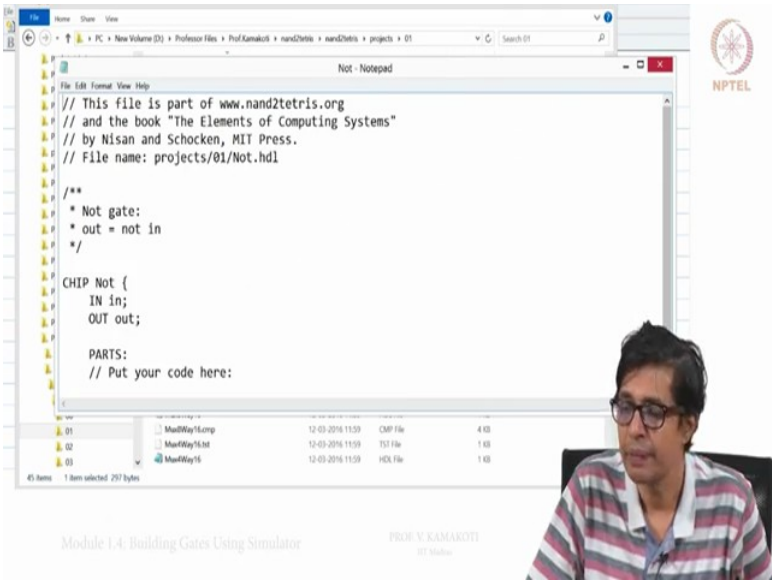
(Refer Slide Time: 12:09)

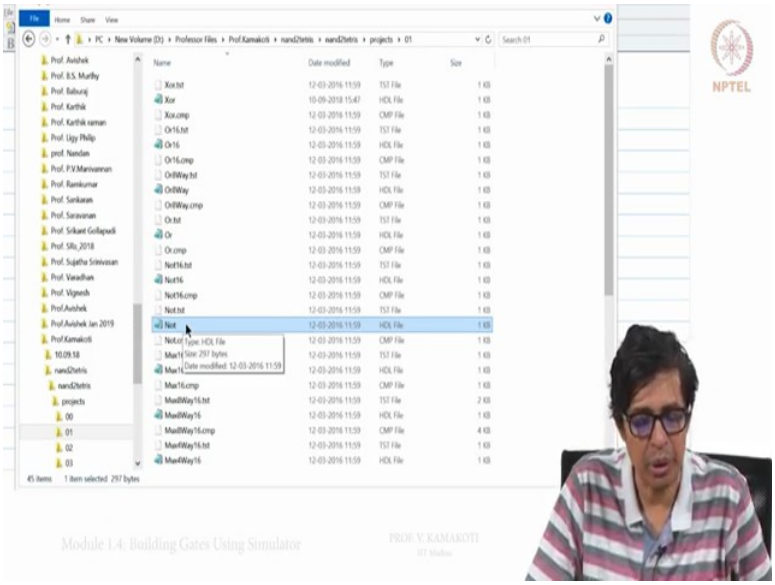



Module 1.4: Building Gates Using Simulator
PROF. V. KAMAKOTI


Or16 - Notepad
File Edit Format View Help
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Or16.hdl
/**
* 16-bit bitwise Or:
* for i = 0..15 out[i] = (a[i] or b[i])
*/
CHIP Or16 {
IN a[16], b[16];
OUT out[16];
PARTS:
// Put your code here:
Module 1.4: Building Gates Using Simulator
PROF. V. KAMAKOTI


Or - Notepad
File Edit Format View Help
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Or.hdl
/**
* Or gate:
* out = 1 if (a == 1 or b == 1)
* 0 otherwise
*/
CHIP Or {
IN a, b;
OUT out;
PARTS:
Module 1.4: Building Gates Using Simulator
PROF. V. KAMAKOTI





Now as a part of this I give it as an exercise or all of you to basically do OR gate, NOR gate and not gate itself using NAND. We have already seen how it works and that I give as an assignment for you to do this to that write the code you go to the directory the same thing as you have seen here so you can write the code here. I just take you back to the directory here, so you can see this Or16.sdl, right? You can see this OR, this is OR gate, you can see the OR we have done the XOR now you can see the OR. Similarly, you can see the Not, right? You can see the OR you can see the Not you can also see the AND, right. Probably there is no NOR but you please create a NOR file like this he is, you can create using notepad and do the NOR also, so this is left as a first exercise for you. Thank you.