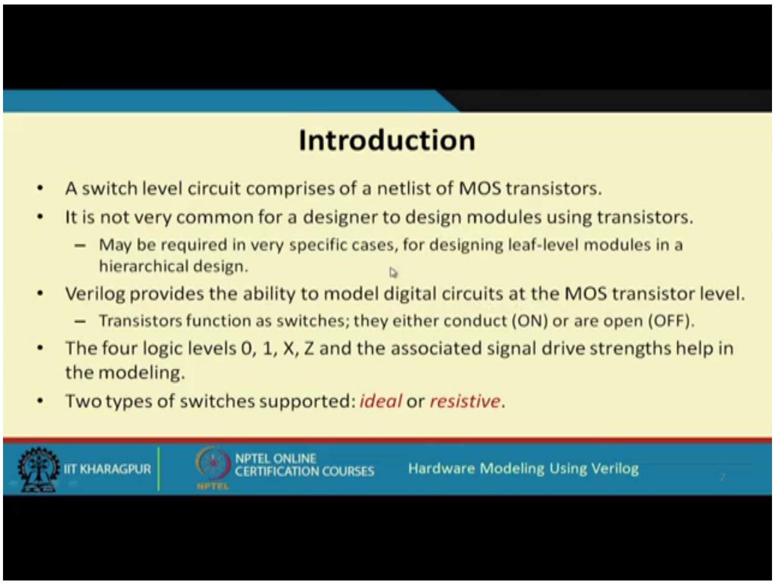
Hardware Modeling using Verilog
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 35
Switch Level Modeling (Part 1)

So, in this lecture we shall be moving away slightly from what we have been discussing and we shall be looking at something called switch level modelling using verilog. Now, this is something which I thought that as a designer someone should know that how we can model some circuits not only at the level of gates and functional blocks, but also at a lower level when our building blocks are transistors and switches. So, we shall see; what are the facilities that are provided by the language verilog and what kind o f modelling you can do or carry out using that ok.

(Refer Slide Time: 01:08)



Introduction

- A switch level circuit comprises of a netlist of MOS transistors.
- It is not very common for a designer to design modules using transistors.
  - May be required in very specific cases, for designing leaf-level modules in a hierarchical design.
- Verilog provides the ability to model digital circuits at the MOS transistor level.
  - Transistors function as switches; they either conduct (ON) or are open (OFF).
- The four logic levels 0, 1, X, Z and the associated signal drive strengths help in the modeling.
- Two types of switches supported: *ideal* or *resistive*.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

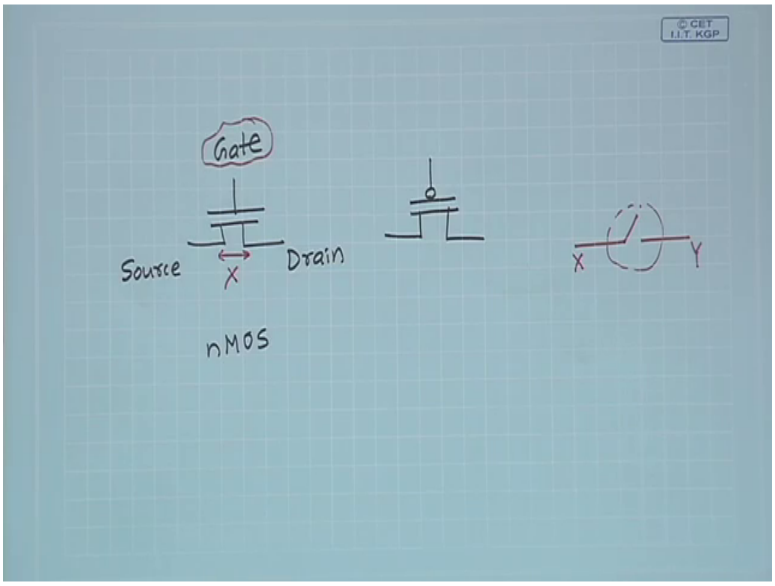
Let see. So, so when you talk about switch level circuit basically we are talking about a circuit that consists of MOS transistors, now for those of you who are familiar with MOS transistors means you will be knowing that there are 2 kinds of MOS transistors, to the then MOS and p MOS depending on the kind of impurities you put in the source and the renditions right ok.

Now, the point to notice that as a designer when you are using verilog to design a digital circuit it is not very common for you to design a model using transistors, but there may be some very specific cases, where a low level module which is sometimes called leaf level module you can be using MOS transistors to model them as a hierarchical design, just I am give an example suppose you want that well verilog provides gate level primitives and, or, nand, nor, x or all right, but you want some your own design using MOS transistors to implement some gate.

So, you can write a very low level module using MOS transistor let say to implement an exclusive or gate and that x or gate you can use, you can instantiate it in higher level designs to create more complex designs. So, it is only under these conditions you can possibly use switch level modelling normally you do not use switch level models in a real design.

Now, as I had said verilog provide some facilities for modelling at the MOS level where the transistors are regarded as a switch you see in MOS level a transistor is represented like this.

(Refer Slide Time: 03:13)



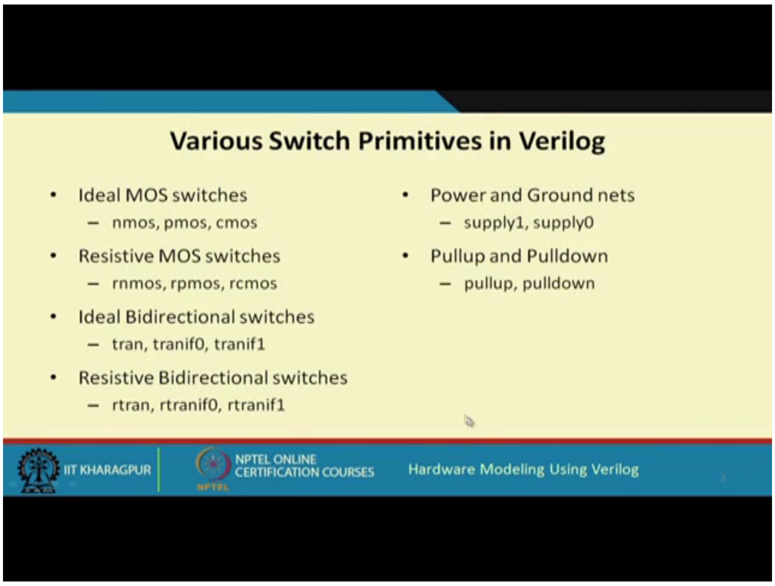
This is called the gate and the other 2 terminals one of them is source, other is drain, this is an example of an n MOS transistor the symbol and p MOS transistor is same where there is a bubble here, same kind.

Now, we are regarding a transistor as a switch, let us see what is a switch a switch schematically represent like this, there are 2 terminals x and y they may be connected if I close the switch they will be not be connected if I open the switch. Here also depending on what voltage I am applying to gate either, the source and drain may be conducting they may be connected or they may not be connected.

So, you can regard a transistor as a switch that is what is meant by switch level modelling right. So, earlier we have seen that in verilog there are 4 logic values which are supported 0 1 x and z and earlier we also talked about signal drive strength, drive strengths. Now, here when you are just working with MOS transistors and the basic primitives we shall be seeing we shall be appreciating that why the signal drive strength are required in the modelling and regarding the switches there are 2 types of switches which are supported 1 is called ideal or this called resistive.

Now, in ideal switch means when you close a switch there will be 0 resistance and a resistive switch means when you close the switch there will be a low resistance, but not 0 a finite low value of resistance. So, if there is a low resistance what will happen is that if I if we apply a signal at one side on the other side this strength of the signal will be reducing a little bit, this is how the concept of signal strength comes in ok.

(Refer Slide Time: 05:44)



Various Switch Primitives in Verilog

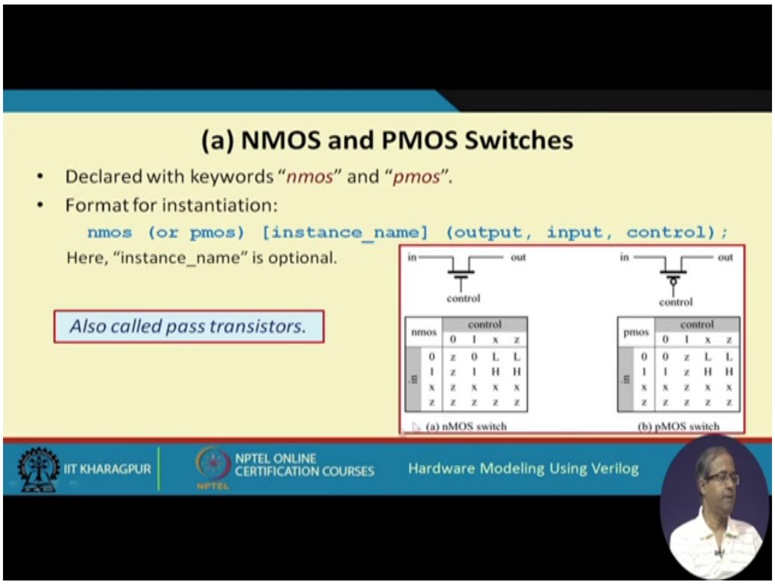
- Ideal MOS switches
  - nmos, pmos, cmos
- Resistive MOS switches
  - rnmos, rpmos, rcmos
- Ideal Bidirectional switches
  - tran, tranif0, tranif1
- Resistive Bidirectional switches
  - rtran, rtranif0, rtranif1
- Power and Ground nets
  - supply1, supply0
- Pullup and Pulldown
  - pullup, pulldown

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

When a signal is passing through a resistive switch the strength of the signal decreases fine, now the various switch level primitives which are supported by verilog are as follows, the first are the ideal MOS switches they are represented by this primitives n MOS p MOS and c MOS we shall see them. There are resistive versions of these also just an r before this names very similar, but there will be a non 0 resistance here and this switches are normally assume to be conducting current in one direction, but there is another kind of a switch which is called bidirectional switch which is assume to conduct in both directions ok.

So, you can have bidirectional switches also these are denoted by tran, tran, tran if 0 tran if 1 we will see what these are and similarly there are resistive versions of these switches with r and there are some keywords to indicate supply voltages power supply 1 and supply 0 directly you can mention them and there is something all pull up and pull down this also we shall see.

(Refer Slide Time: 07:03)



(a) NMOS and PMOS Switches

- Declared with keywords "*nmos*" and "*pmos*".
- Format for instantiation:

```
nmos (or pmos) [instance_name] (output, input, control);
```

Here, "instance_name" is optional.

Also called pass transistors.

in
out
control
in
out
control

nmos		control			
		0	1	x	z
in	0	z	0	L	L
	1	z	1	H	H
	x	z	x	x	x
	z	z	z	z	z

(a) nMOS switch

pmos		control			
		0	1	x	z
in	0	0	z	L	L
	1	1	z	H	H
	x	x	z	x	x
	z	z	z	z	z

(b) pMOS switch

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

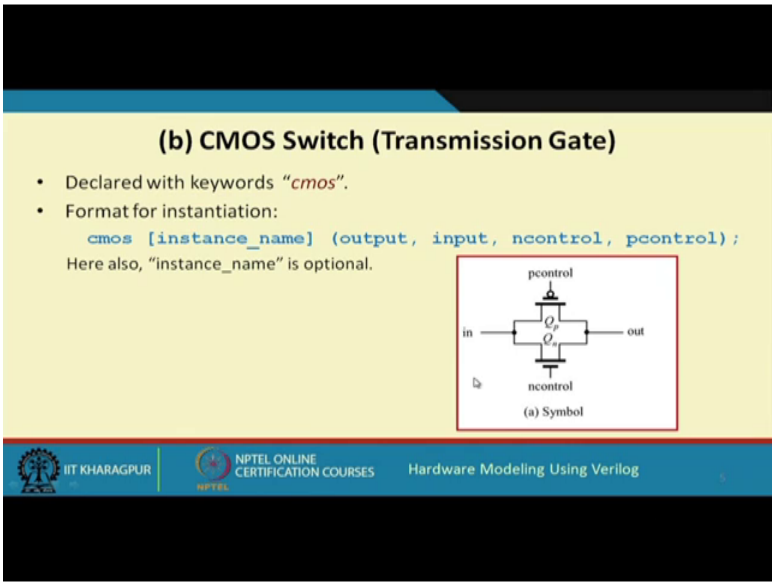
Now, let us come to n MOS and p MOS switches first, this n MOS and p MOS switches they are declared with the key key words n MOS and p MOS. Now as I had shown you earlier this is a schematic of an n MOS transistor acting in the switch and this is a p MOS transistor there are 2 input and output terminals and there is a control, gate is the control here also it is similar.

Now, the wait works is that if the control for an n MOS switch if the control is at high if the control is 1 then the gate is conducting. So, if the input is 0 this is input then the output is 0 this is output if the input is 1, the output is 1, but if the control is 0 then the switch is off. So, the output will be in the high impedance state it has not connected to anything ok.

Just ignore these 2 columns for the time being, if the control is either undefined or in the high impendence state well the verilog language specifies that if we apply a 0 the output voltage will also be at the low level ground level. If we apply a 1 the output will be at a high level because there is no voltage drop, but if you apply I means x and z in the input then the output will indeterminate x and z, similarly for a p MOS it is just a reverse if the control is set to 0 then the switch is conducting if the control is one then the switch is off ok.

We need only this much, this this part of the table and the way you can instantiate is you write either n MOS or p MOS, well instance name is optional you can give a name and first the output then the input then the control.

(Refer Slide Time: 09:11)



(b) CMOS Switch (Transmission Gate)

- Declared with keywords "*cmos*".
- Format for instantiation:

```
cmos [instance_name] (output, input, ncontrol, pcontrol);
```

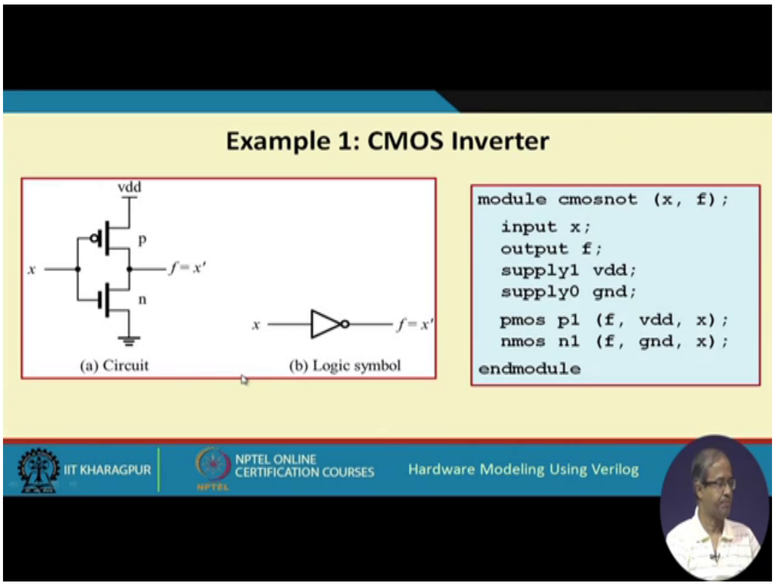
Here also, "instance_name" is optional.

pcontrol
in
Q_p
Q_n
out
ncontrol
(a) Symbol
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

This is the order these switches are sometimes also called pass, well there is a c MOS version of CMOS stands for complimentary MOS, complimentary MOS.

So, in a CMOS switch there is an n MOS switch and a p MOS switch which are connected in parallel and there are 2 control signals n control and p control. So, when we instantiate it, it is output, input n control and p control. Normally this n control and p control are complements of each other say if I apply one here and 0 here then it will be conducting if I apply 0 here and one here both of them are off so it will be off. This is a better switch as compared to a single transistor switch because, when you use single n type or p type transistor as a switch there is a voltage degradation which happens, but if you 2 back to back switches that voltage degradation is avoided.

(Refer Slide Time: 10:22)



Example 1: CMOS Inverter

vdd
p
n
x
f = x'
(a) Circuit
x
f = x'
(b) Logic symbol
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

```
module cmosnot (x, f);
  input x;
  output f;
  supply1 vdd;
  supply0 gnd;
  pmos p1 (f, vdd, x);
  nmos n1 (f, gnd, x);
endmodule
```

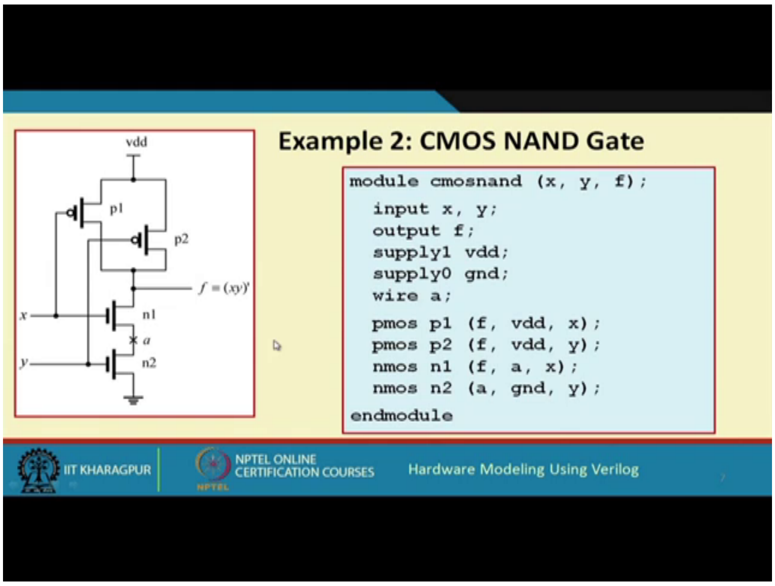
So, I am not going into details of this here, but just remember this that, the c MOS switch is a better kind of a switch let us take some example, designs a c MOS inverter for those of you who have studied MOS level gates will just identify the circuit. There is an n MOS transistor, there is a p MOS transistor whose gates are connected this is the input and the sources or the drain whatever you call this is output other side is connected to power supply vdd and ground this is the symbolic notation.

Now, the way to works is very simple if the input x is 0, if it is 0 then the p type switch is on, if it is and the n type switch is off. If p type is on that vdd will be connected to the output, output will be 1 if it is 0 output is 1 and if the input is 1 then the n type will be conducting this switch is on.

So, the output will be connected to ground output will be 0 so it is not gate. So, you can directly specify this net list in verilog like this, let us call it c MOS not x is the input f is the output so input x output f. So, here you are using vdd and ground terminals also, that is why you define 2 variables one you called as vdd another called g n d of type supply one and supply 0. There is 1 p MOS transistor less quality p 1 f an vdd are the 2 terminals this is f and this is vdd and the gate is x.

Similarly, there is another transistor n MOS just call it n 1 f and ground are the 2 terminals and the input is x. So, this is the complete description.

(Refer Slide Time: 12:11)



Example 2: CMOS NAND Gate

vdd
p1
p2
$f = (xy)'$
x
n1
a
y
n2

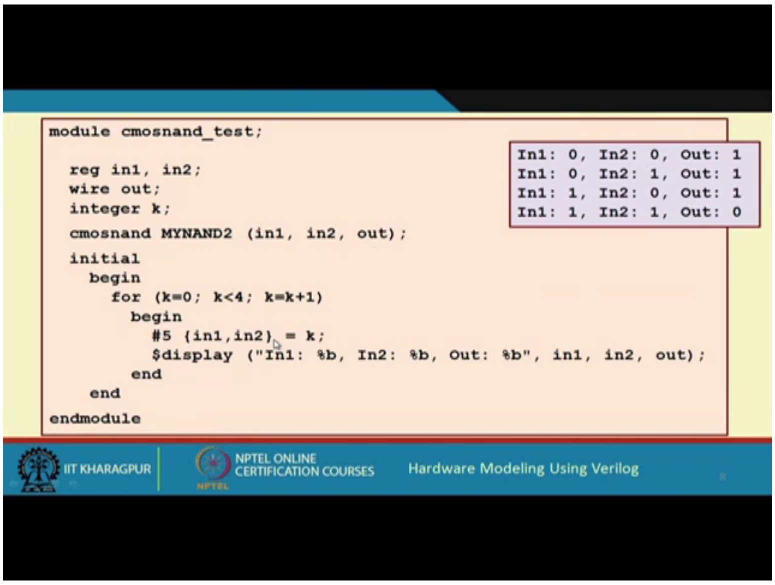
```
module cmosnand (x, y, f);
  input x, y;
  output f;
  supply1 vdd;
  supply0 gnd;
  wire a;
  pmos p1 (f, vdd, x);
  pmos p2 (f, vdd, y);
  nmos n1 (f, a, x);
  nmos n2 (a, gnd, y);
endmodule
```

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Hardware Modeling Using Verilog

Similarly, you can have a nand gate 2 input nand gate. So, 2 input nand gate there are 2 n type transistors here and 2 p type transistors here, for a nand gate you recall when both the inputs are 1 1 output is 0. So, when both x and y are 1 and 1, both the switches n 1 and n 2 will be conducting on. So, the output will be connected to ground it will be 0, but if if any one of them is 0. So, either p 1 or p 2 will be conducting. So, there will be a path from vdd to the output. So, the output will be 1.

So, if any one of the input is 0 output is 1 that is nand. So, description see you see is I am not here trying to teach you how to design circuits using MOS transistor. So, what I am saying is that given a MOS level circuit how to modulate in verilog, you say this circuit can be model in verilog like this x y and f, inputs are x y output f, again vdd ground are the 2 supplies and intermediate there is one line a I declared as wire. P MOS just this p 1 f vdd x f is this, this is f, this is vdd and this is x. For p 2 f vdd, y f vdd y for n 1, they said f a x, f a x and for n 2 ground a or a ground whatever order you specify and y a ground y ok.

(Refer Slide Time: 13:55)



```
module cmosnand_test;

  reg in1, in2;
  wire out;
  integer k;

  cmosnand MYNAND2 (in1, in2, out);

  initial
    begin
      for (k=0; k<4; k=k+1)
        begin
          #5 {in1,in2} = k;
          $display ("In1: %b, In2: %b, Out: %b", in1, in2, out);
        end
    end
endmodule
```

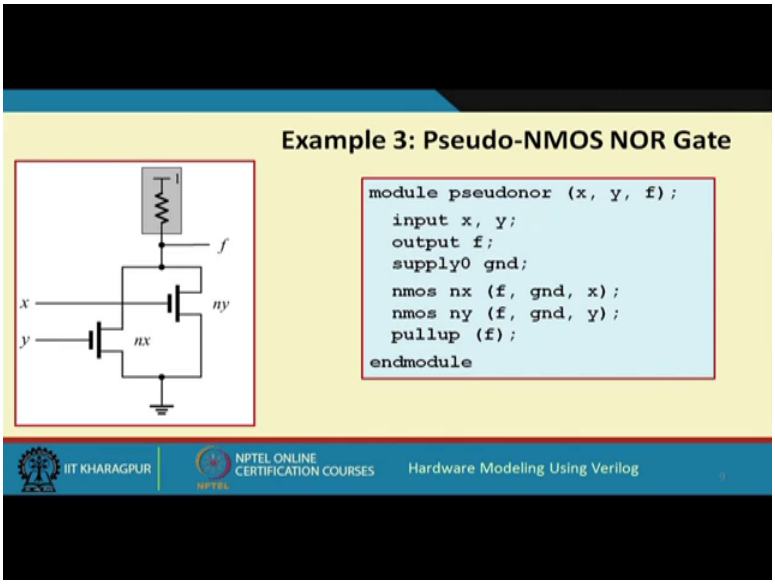
```
In1: 0, In2: 0, Out: 1
In1: 0, In2: 1, Out: 1
In1: 1, In2: 0, Out: 1
In1: 1, In2: 1, Out: 0
```

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Hardware Modeling Using Verilog

Now, this n MOS nand if you write a simple test bench to see whether it works or not. So, so you can simulate and see that it works really works. So, we have instantiated this c MOS nand in a test bench like this we called it my nand 2, n 1, n 2 out the inputs were declared as ragout as wire and defined an integer k. So, why just in order to apply all possible inputs you see this is a 2 input circuits. So, what you did, in this initial block we give a for loop which runs from k equal to 0 up to 3 k less than 4 0 1 2 3 if there are 2 inputs. So, the combinations will be 0 0, 0 1, 1 0 and one one which means 0, 1, 2, 3 and counting in decimal that k.

So, in k I am counting decimal 0 up to 3 and that value of k I am assigning to the inputs using concatenation operation say for example, if k is 3 it means 1 1, in 1 will be 1 in 2 will be 1. So, this will be happening automatically and we are displaying the values of in 1 in 2 and out.

(Refer Slide Time: 15:24)



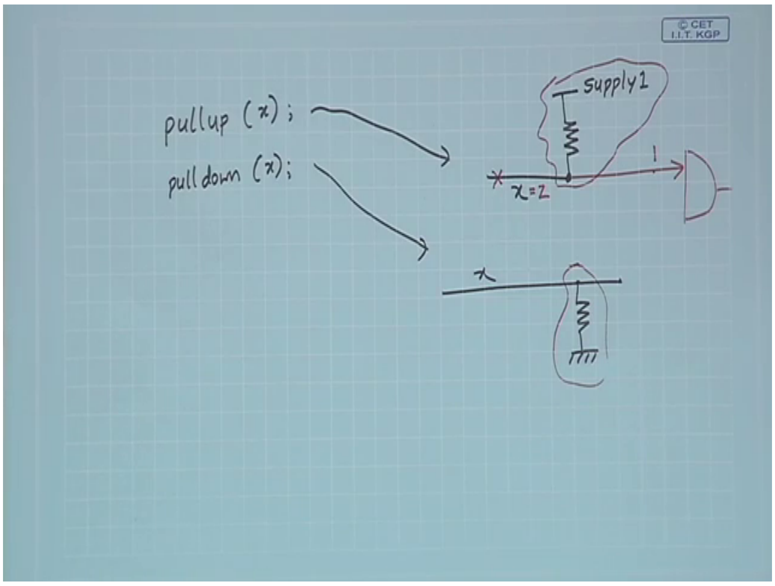
Example 3: Pseudo-NMOS NOR Gate

```
module pseudonor (x, y, f);
  input x, y;
  output f;
  supply0 gnd;
  nmos nx (f, gnd, x);
  nmos ny (f, gnd, y);
  pullup (f);
endmodule
```

1
f
x
ny
y
nx
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, if you just simulate it we get this result 0 0 output is 1, 0 1 output is 1, 1 0 1, 1 1 0 this is nand fine.

(Refer Slide Time: 15:41)



Let us take another example this is called pseudo NMOS nor gate, but before we explain this let us tell you about the primitive that are available pull up and pull down. See you can with a signal let say x you can either write pull up x or with a signal y say x you can write pull down x. what does this mean pull up x means there is a signal line x pull up means, from here you are connecting a resistance to supply one; that means, positive supply voltage this is called pull up. So, when you say pull up this whole thing will be included by default there.

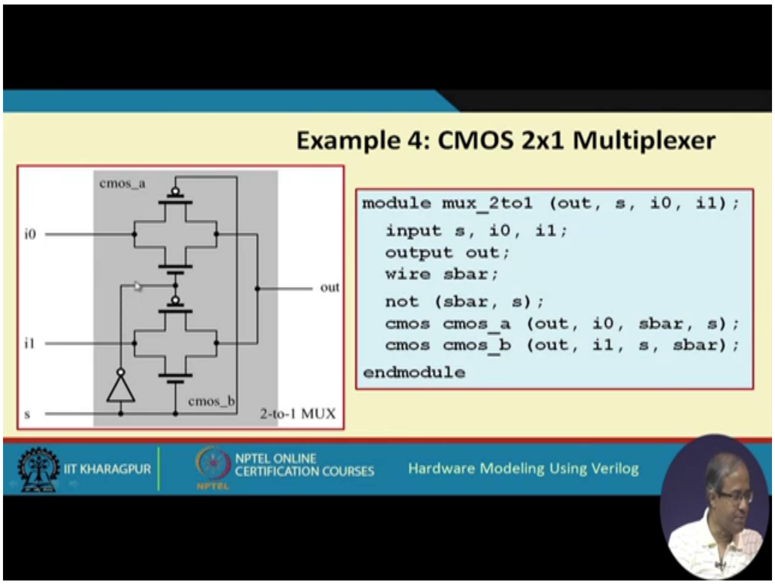
So, the meaning of pull up is that suppose this output your feeding to the input of some, let us see say and because of some reason this x is in the high impedance state let say x equal to z, then because of the pull up this input will still be at 1 because it is connected to the supply with the resistance. So, it will not be in the high impedance. So, input will not be z.

Similarly, pull down means this x is the signal you connect a resistance to ground, this means pull down. So, in circuits we often use this kind of pull up and pull down configurations to implement several things, like here we are showing an alternate way of of designing gates, this is a this is a nor gate and this is called pseudo n MOS. Pseudo means here you have a resistance in the pull up and down you have 2 n MOS transistors you see why it is nor in a nor gate how does it work. So, if the inputs are 0 0 then only the output is 1 otherwise the output is 0, let us say here if the inputs are 0 and 0 both the transistors are off.

So, f is not connected to ground so f will be connected through this pull up to one. So, f will be high or 1, but if any one of them is 0 that transistor will be on and f will be connected to ground. Now, see f is connected on one hand to the ground through an on switch, on the other side it is connected to a pull up resistance to a supply voltage. So, will the output be ground or high you say the the idea is pull up or pull down has a lower signal strength because of the resistance. So, when 2 such signals are tied together the signal which is stronger it will dominate like here on one side you are trying to connect to ground and other side you are connecting to one, but ground is a stronger signal here.

So, ground will dominate because this is ideally 0 resistance so f will be 0 right. So, declaration is very simple x y input f is output and here we need to declare only ground supplies g n d there is 1 n MOS n x, another n MOS n y f ground x, f ground x, f ground y and there is a pull up at point f, at point f is a pull up this is the module description. So, this module in a same way if you do a simulation that same test bench we have just included my nor instead of the other one pseudo nor. So, if you simulate it remaining part is same you see the output is coming like this 0 0 is 1 otherwise output is 0 this is nor.

(Refer Slide Time: 20:01)



Example 4: CMOS 2x1 Multiplexer

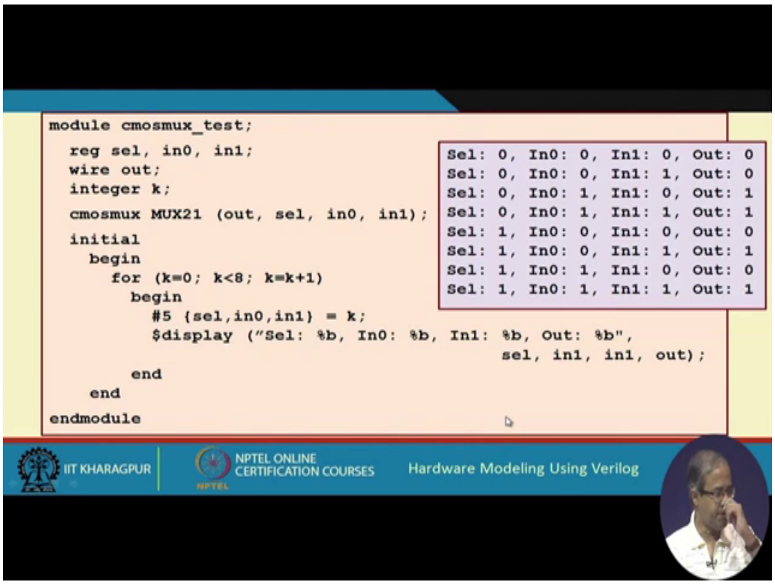
```
module mux_2to1 (out, s, i0, i1);
  input s, i0, i1;
  output out;
  wire sbar;
  not (sbar, s);
  cmos cmos_a (out, i0, sbar, s);
  cmos cmos_b (out, i1, s, sbar);
endmodule
```

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Hardware Modeling Using Verilog

So, this is 1 more example let us take here we use a c MOS switch, just see this diagram there are 2 c MOS switches 1 c MOS switch is here another c MOS switch is here and there is a we are implementing a 2 to 1 multiplexer and the 2 inputs are connected to this c MOS switches and the select line s is connected to the n MOS transistor of one of the switches and to the p MOS transistor of the other switch and there is a not gate not of s is connected to the p transistor of this switch and n transistor of this switch. What does this mean? If s is 0 if s is 0 then this will be 1 not this will be on this will also be on. So, this switch is on, but this switch is off because this is on I 0 will go to the output and if s equal to 1 then this will be on and also this will be on, but this will be off.

So, I one will go to out. So, representing in verilog is very simple this input s I 0 I 1 output is out and there is an intermediate line output of this not gate I call it s bar. So, I instantiate this not s is the input s bar is the output and 2 c MOS gates I have just instantiated one I called c MOS a, other I called c MOS b. So, the inputs are output the I 0 and control is I means s bar in means n type and s in p type. So, s bar n type and p type and for c MOS b out this is I one and here it is s in p type I means in n type and s bar in p type s and s bar.

(Refer Slide Time: 22:18)



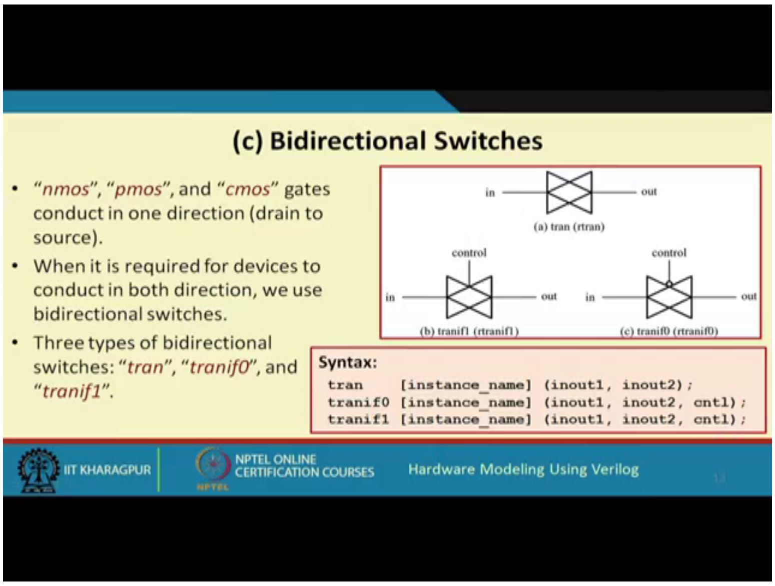
```
module cmosmux_test;
  reg sel, in0, in1;
  wire out;
  integer k;
  cmosmux MUX21 (out, sel, in0, in1);
  initial
    begin
      for (k=0; k<8; k=k+1)
        begin
          #5 {sel,in0,in1} = k;
          $display ("Sel: %b, In0: %b, In1: %b, Out: %b",
                                        sel, in1, in1, out);
        end
    end
endmodule
```

```
Sel: 0, In0: 0, In1: 0, Out: 0
Sel: 0, In0: 0, In1: 1, Out: 0
Sel: 0, In0: 1, In1: 0, Out: 1
Sel: 0, In0: 1, In1: 1, Out: 1
Sel: 1, In0: 0, In1: 0, Out: 0
Sel: 1, In0: 0, In1: 1, Out: 1
Sel: 1, In0: 1, In1: 0, Out: 0
Sel: 1, In0: 1, In1: 1, Out: 1
```

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, this again see here there are 3 inputs total right. So, we have written test bench similarly, but we have applied 8 pattern 0 up to 7 and this case assigned to select n 0, n 1 like this. So, we have printed the values of cell in one there should be n 0 actually n 0 n 1 and out. So, you see f select is 0 then n 0 is getting selected if select is 1 then n 1 is selected, 0 1 0 is getting selected right. So, it work as a multiplexer.

(Refer Slide Time: 23:00)



(c) Bidirectional Switches

- "*nmos*", "*pmos*", and "*cmos*" gates conduct in one direction (drain to source).
- When it is required for devices to conduct in both direction, we use bidirectional switches.
- Three types of bidirectional switches: "*tran*", "*tranif0*", and "*tranif1*".

Syntax:

```
tran    [instance_name] (inout1, inout2);
tranif0 [instance_name] (inout1, inout2, cntl);
tranif1 [instance_name] (inout1, inout2, cntl);
```

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, and lastly in this lecture we shall be talking about something called bidirectional switches. So, of course, will not be giving example just to tell you what it is, see in a normal switch p MOS n MOS or c MOS well it is assumed that current flows in only one direction, but this is only for the purpose of simulation in a real switch current actually frozen both directions. So, if in a design you need to have a switch where you need current to flow in both directions then you should use something called bidirectional switchs switch are called tran.

So, there are 3 kinds of bidirectional switches tran tran if 0 and tran if 1. So, the syntax of the tran switch is tran well instance name as usual is optional means because it is bidirectional I call them in out in out 1 in out 2, there are 2 terminals and this is always conducting, but tran if 0 and tran if one means there is a control signal. Tran if 0 says if the control signal is 0 then it will conduct and tran if one says if the control signal is 1 if 1 then it will conduct and if it is control is other way around then output will be tri stated this will be off, this is how the bidirectional switches work.

So, with this we come to the end of this lecture where we talked about some of the low level MOS switch primitives which also you can use in verilog modelling, but again I am telling you these are mostly used for simulation purposes, most of the synthesis tools will not be supporting this kind of MOS level primitives. So, in the next lecture we shall be looking some more examples on this kind of switch level modellings and a few other things.

Thank you.