Switch Level Modeling
Switch modeling elements

- MOS switches
  - nmos
  - pmos
- nmos (out, data, control)
- pmos (out, data, control)
Switch modeling elements

- CMOS switches
  - nmos (out, data, ncontrol, ncontrol)
Switch modeling elements

- Bidirectional switches
  - Conduct in both directions
  - `tran, tranif0, tranif1`

```plaintext
tran t1(inout1, inout2); //instance name t1 is optional
tranif0 (inout1, inout2, control); //instance name is not specified
tranif1 (inout1, inout2, control); //instance name is not specified
```
Power and ground

- `supply1, supply0`

```plaintext
supply1 vdd;
supply0 gnd;

assign a = vdd;  //Connect a to vdd
assign b = gnd;  //Connect b to gnd
```
CMOS inverter
/Define an inverter using MOS switches
module my_not(out, in);

output out;
input in;

//declare power and ground
supply1 pwr;
supply0 gnd;

//instantiate nmos and pmos switches
pmos (out, pwr, in);
nmos (out, gnd, in);

dendmodule
CMOS NOR gate
CMOS NOR gate

//Define our own nor gate, my_nor
module my_nor(out, a, b);

output out;
input a, b;

//internal wires
wire c;

//set up power and ground lines
supply1 pwr;  //pwr is connected to Vdd (power supply)
supply0 gnd;  //gnd is connected to Vss(ground)

//instantiate pmos switches
pmos (c, pwr, b);
pmos (out, c, a);

//instantiate nmos switches
nmos (out, gnd, a);
nmos (out, gnd, b);

endmodule
// stimulus to test the gate
module stimulus;
reg A, B;
wire OUT;

// instantiate the my_nor module
my_nor n1(OUT, A, B);

// apply stimulus
initial
begin
  // test all possible combinations
  A = 1'b0; B = 1'b0;
  #5 A = 1'b0; B = 1'b1;
  #5 A = 1'b1; B = 1'b0;
  #5 A = 1'b1; B = 1'b1;
end

// check results
initial
$monitor($time, " OUT = %b, A = %b, B = %b", OUT, A, B);
endmodule
Exercise

1. Draw the circuit diagram for an xor gate, using nmos and pmos switches. Write the Verilog description for the circuit. Apply stimulus and test the design.

2. Draw the circuit diagram for and and or gates, using nmos and pmos switches. Write the Verilog description for the circuits. Apply stimulus and test the design.

3. Design the 1-bit full-adder shown below using the xor, and, and or gates built in Exercise 1 and Exercise 2 above. Apply stimulus and test the design.

![Circuit Diagram](image)