30 Days of RTL Coding

-By T Tushar Shenoy

Day 9

Problem Statement: Implementing Logic Gates using Switch Level of Modelling.

Theory:

The switch level of modelling provides a level of abstraction between the logic and analog-transistor levels of abstraction. It describes the interconnection of transmission gates, which are abstractions of individual MOS and CMOS transistors.

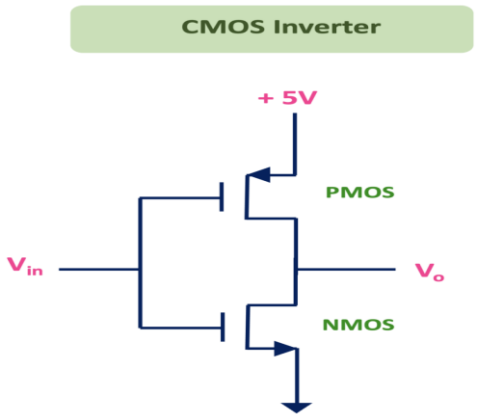
The switch level transistors are modelled as being either on or off, conducting or not conducting. The values carried by the interconnections are abstracted from the whole range of analog voltages or currents to a small number of discrete values. These values are referred to as signal ***strengths***.

Verilog also provides support for transistor level modelling. However, designers rarely use these days as the complexity of circuits has required them to move to higher levels of abstractions rather than use switch level modelling. Usually, the transistor level modelling is referred to model in hardware structures using transistor models with analog input and output signal values.

On the other hand, gate-level modelling refers to modelling hardware structures wing gate models with digital input and output signal values between these two modelling schemes are referred to as switch level modelling.

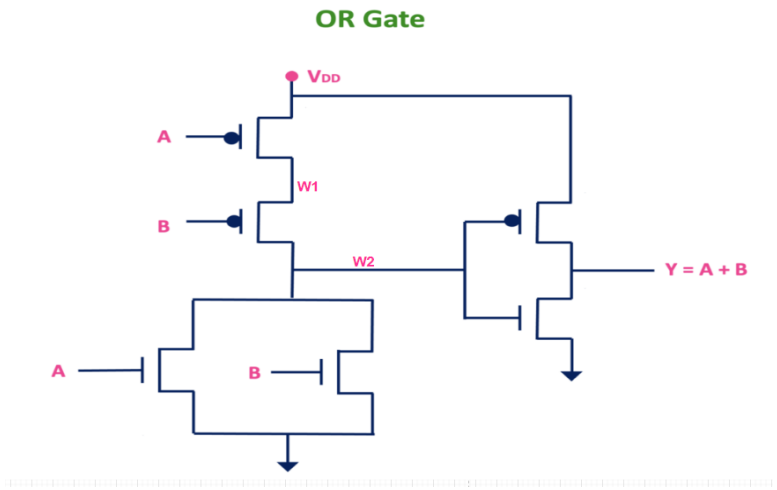
The transistors only exhibit digital behaviour and their input at the transistor level, and output signal values are only limited to digital values. Verilog uses a 4 value logic value system, so Verilog switch input and output signals can take any of the four ***0, 1, Z,*** and ***X*** logic values.

CMOS Inverter Circuit:



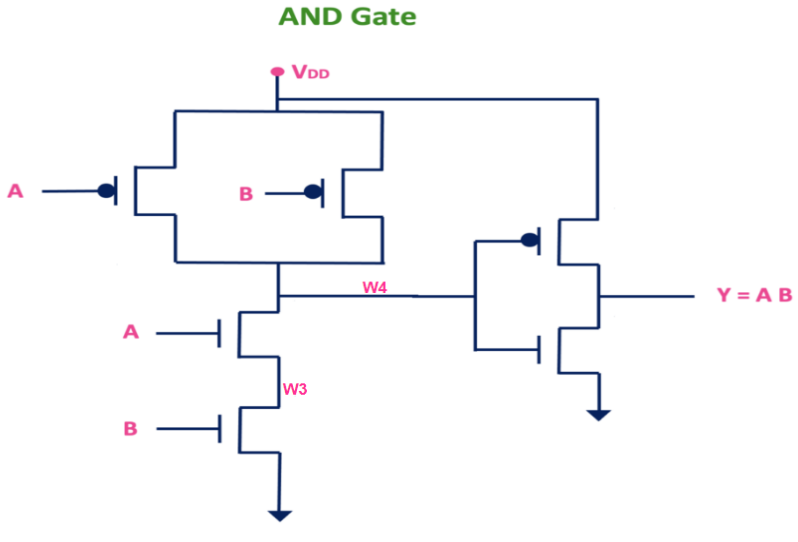
CMOS OR Gate:

To implement the OR gate, just add the inverter at the output of the NOR gate. The CMOS OR gate is shown below.

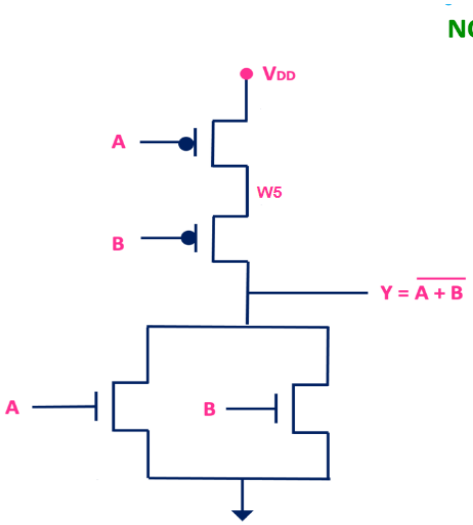


CMOS AND Gate:

Similarly, by connecting the inverter at the output of the NAND gate, we can implement AND gate. The CMOS AND gate is shown below.



CMOS NOR Gate:



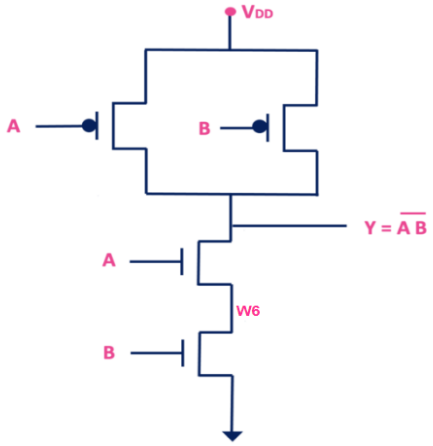
Truth Table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

For two input NOR gate, if A and B are the inputs then its output Y = (A+B)'.In the NMOS network, whenever there is an OR operation between the two variables then two NMOS transistors will get connected in parallel. And the output will be complement of it.The PMOS network will be the dual of the NMOS network. Therefore, in the PMOS network, the two PMOS transistors will get connected in series.

CMOS NAND Gate:

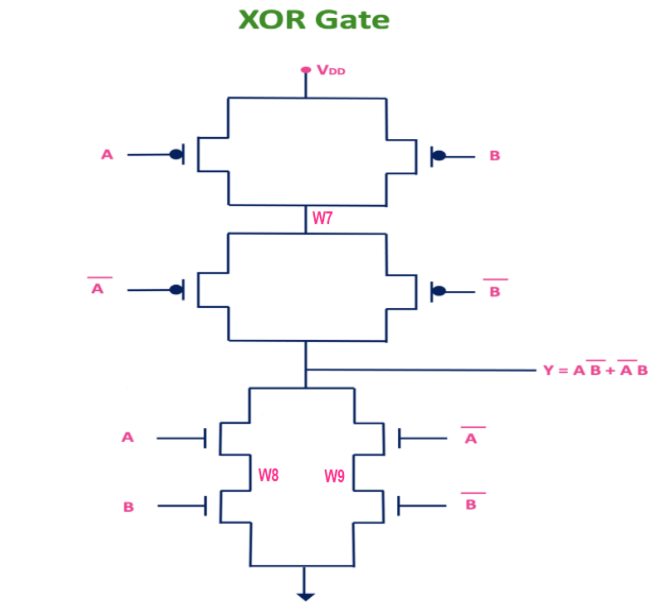
For two input NAND gate, if A and B are the inputs then its output Y = (A.B)'.In NMOS network when we have AND operation between the two variables, then two NMOS transistors will get connected in series. And the output will be complement of it. The PMOS network is dual of the NMOS network. In the NMOS network, if two transistors are connected in series then in the PMOS network, the two PMOS transistors will get connected in parallel.



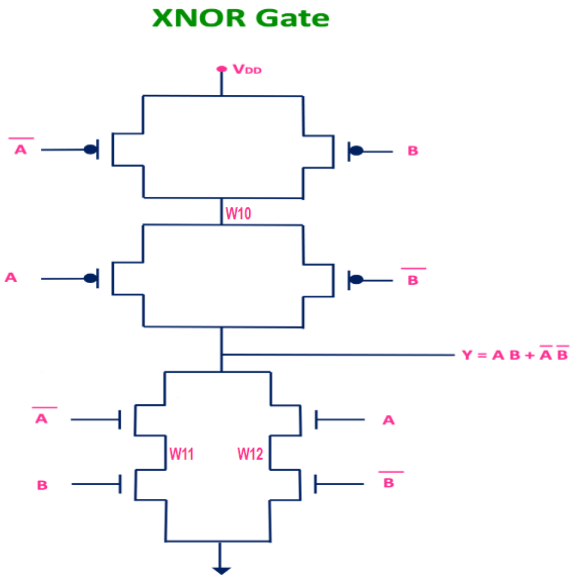
Truth Table

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

CMOS XOR Gate:



CMOS XNOR Gate:



Verilog Code:

```
module cmoslogicgates(A,B,yor,yand,ynor,ynand,yxor,yxnor);
input A,B;
output yor,yand,ynor,ynand,yxor,yxnor;
supply1 Vdd;
supply0 Gnd;

//The Placement of wires has been shown in the Circuit
//OR Gate
wire w1,w2;
    pmos(w1,Vdd,A);
    pmos(w2,w1,B);

    nmos(w2,Gnd,A);
    nmos(w2,Gnd,B);

    pmos(yor,Vdd,w2);
    nmos(yor,Gnd,w2);

//AND Gate
wire w3,w4;
    pmos(w4,Vdd,A);
    pmos(w4,Vdd,B);
```

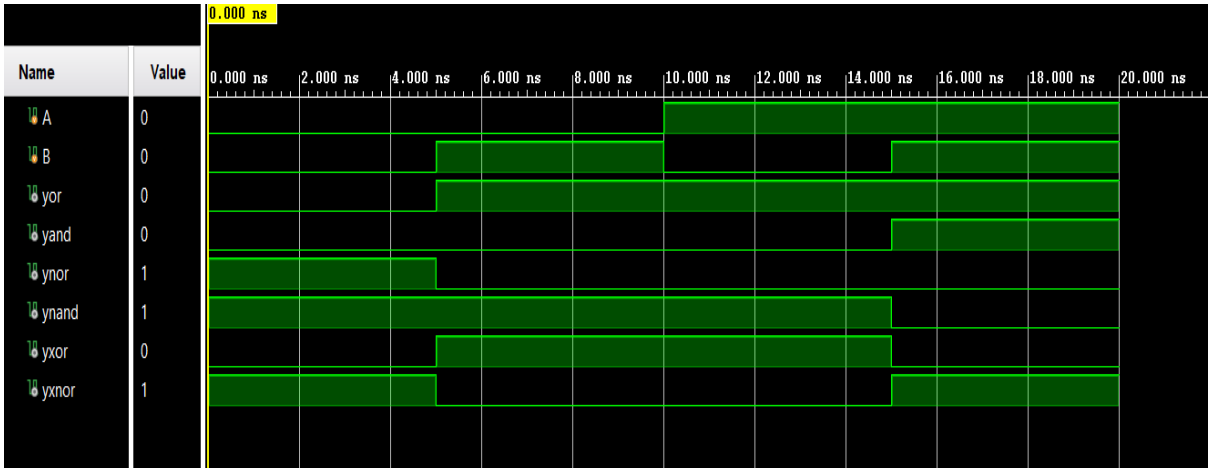
```
    nmos(w4,w3,A);
    nmos(w3,Gnd,B);

    pmos(yand,Vdd,w4);
    nmos(yand,Gnd,w4);

//NOR Gate
wire w5;
    pmos(w5,Vdd,A);
    pmos(ynor,w5,B);

    nmos(ynor,Gnd,A);
    nmos(ynor,Gnd,B);

//NAND Gate
wire w6;
    pmos(ynand,Vdd,A);
    pmos(ynand,Vdd,B);
    nmos(ynand,w6,A);
    nmos(w6,Gnd,B);

//XOR Gate
wire w7,w8,w9;
    pmos(w7,Vdd,A);
    pmos(w7,Vdd,B);
```

```
    pmos(yxor,w7,~A);
    pmos(yxor,w7,~B);

    nmos(yxor,w8,A);
    nmos(yxor,w9,~A);
    nmos(w8,Gnd,B);
    nmos(w9,Gnd,~B);


//XNOR Gate
wire w10,w11,w12;
    pmos(w10,Vdd,~A);
    pmos(w10,Vdd,B);

    pmos(yxnor,w10,A);
    pmos(yxnor,w10,~B);

    nmos(yxnor,w11,~A);
    nmos(yxnor,w12,A);
    nmos(w11,Gnd,B);
    nmos(w12,Gnd,~B);

endmodule
```

Testbench Code:

```
module cmoslogicgates_tb();
reg A,B;
wire yor,yand,ynor,ynand,yxor,yxnor;

cmoslogicgates
dut(.A(A),.B(B),.yor(yor),.yand(yand),.ynor(ynor),.ynand(ynand),.yxor(yxor),
.yxnor(yxnor));
 initial begin
       A=0;B=0;
    #5 A=0;B=1;
    #5 A=1;B=0;
    #5 A=1;B=1;
    #5 $finish;
 end
 endmodule
```

Simulation Output:



GitHub Repository URL: -
https://github.com/tusharshenoy/RTL-Day-9-CMOS-Logic-Gates