SWITCH LEVEL MODELING

Introduction

- The MOS transistor is the basic element around which a VLSI is built.
- Designers familiar with logic gates and their configurations at the circuit level may choose to do their designs using MOS transistors.
- Verilog has the provision to do the design description at the switch level using such MOS transistors, which is the theme of the present chapter.
- Switch level modeling forms the basic level of modeling digital circuits.
- The switches are available as primitives in Verilog

MOS Switches

- The MOS switches are:

  cmos pmos nmos rcmos rpmos rnmos

- These gates model unidirectional switches, that is, data flows from input to output and the data flow can be turned off by appropriately setting the control input(s).
- The pmos(p-type MOS transistor), nmos (n-type MOS transistor), rnmos ('r' stands for resistive) and rpmos switches have one output, one input and one control input.
- The basic syntax for an instantiation is:

```
gate_type  [ instance_name ] ( OutputA , InputB , ControlC );
```

a. BASIC SWITCH PRIMITIVES

nmos switch primitives

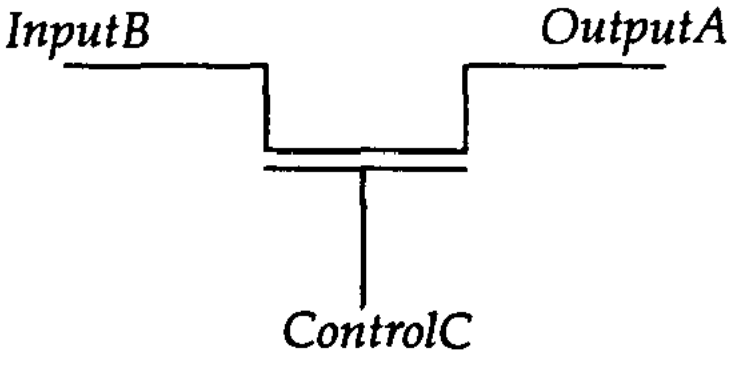
nmos (out, in, control);

pmos switch primitives

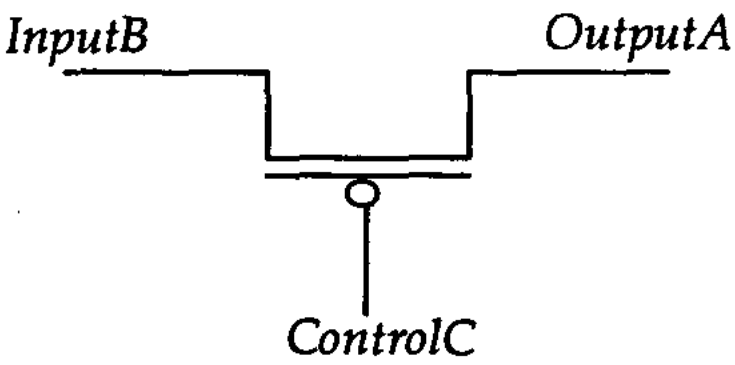
pmos (out, in, control);

pmos rpmos		Control			
		0	1	x	z
Data	0	0	z	0/z	0/ z
	1	1	z	1/z	1/ z
	x	x	z	x	x
	z	z	z	z	z

nmos rnmos		Control			
		0	1	x	z
Data	0	z	0	0/ z	0/ z
	1	z	1	1/ z	1/ z
	x	z	x	x	x
	z	z	z	z	z



nmos switch



pmos switch

```
pmos P1 (BigBus, SmallBus, GateControl);
rnmos RN1 (ControlBit, ReadyBit, Hold);
```

The first instance instantiates a pmosswitch with instance name PI. The input to the switch is SmallBus and the output is BigBus and the control signal is GateControl

b. Resistive switches

- **nmos and pmos** represent switches of low impedance in the on-state.
- **rnmos and rpmos** represent the resistive counterparts of these respectively.

```
rnmos (output1, input1, control1);
rpmos (output2, input2, control2);
```

- It inserts a definite resistance between the input and the output signals but retains the signal value
- The rpmos and rnmos switches function as unidirectional switches; the signal flow is from the input to the output side.

Strength level

- Output-side strength levels for different input-side strength values of rnmos, rpmos, and rcmos switches

Input strength	Output strength
Supply – drive	Pull – drive
Strong – drive	Pull – drive
Pull – drive	Weak – drive
Weak – drive	Medium – capacitive
Large – capacitive	Medium – capacitive
Medium – capacitive	Small – capacitive
Small – capacitive	Small – capacitive
High impedance	High impedance

c. CMOS SWITCH

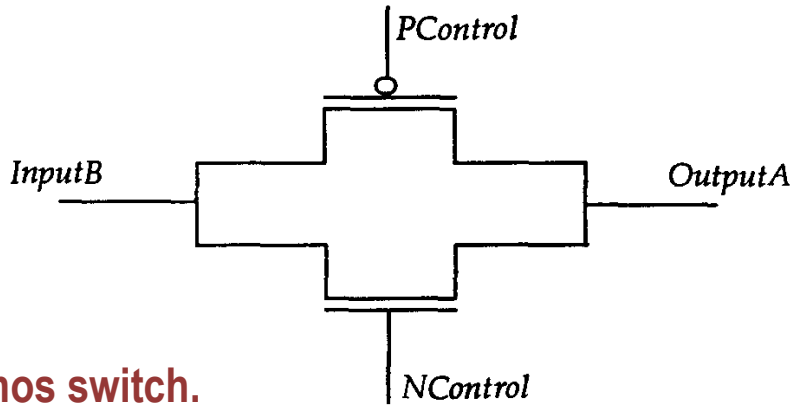
- A CMOS switch is formed by connecting a PMOS and an NMOS switch in parallel – the input leads are connected together on the one side and the output leads are connected together on the other side.

- The CMOS switch is instantiated as shown below.

cmos csw (out, in, N_control, P_control);

```
(r)cmos  [ instance_name ]
                    ( OutputA , InputB , NControl , PControl );
```

- The cmos and rcmos (resistive version of cmos) switches have one data output, one data input and two control inputs.
- The first terminal is the output, the second is the input, the third is the n channel control input and the fourth terminal is the p-channel control input.
- A cmos (rcmos) switch behaves exactly like a combination of a pmos (rpmos) and an nmos (rnmos) switch with common outputs and common inputs;



(r)cmos switch.

2. pullup and pulldown

- A MOS transistor functions as a resistive element when in the active state. Realization of resistance in this form takes less silicon area in the IC as compared to a resistance realized directly.
- pullup and pulldown represent such resistive elements.

pullup (x);

- Here net x is pulled up to the supply1 through a resistance.

pulldown(y);

- pulls y down to the supply0 level through a resistance.
- The pullup and pulldown primitives can be used as loads for switches or to connect the unused input ports to VCC or GND, respectively.

3. Bidirectional Switches

- The bidirectional switches are:

  tran rtran tranif0 rtranif0 tranif1 rtranif1

- These switches are bidirectional, that is, data flows both ways and there is no delay when data propagates through the switches.
- The last four switches can be **turned off** by setting a **control signal** appropriately.
- The tran and rtran switches **cannot be turned off.**
- The syntax for instantiating a tran or a rtran (resistive version of tran) switch is:

```
(r)tran  [ instance_name ] ( SignalA , SignalB );
```

- The terminal list has only two terminals and data flows unconditionally both ways, that is, from SignalA to SignalB and vice versa.
- The syntax for instantiating the other bidirectional switches is:

```
gate_type  [ instance_name ] ( SignalA , SignalB , ControlC );
```

- The first two terminals are the bidirectional terminals, that is, data flows from SignalA to SignalB and vice versa.
- The third terminal is the control signal. If ControlC is 1 for tranif0 and rtranif0, and 0 for tranifl and rtranifl, the bidirectional data flow is disabled.
- For the resistive switches(rtran, rtranif0 and rtranifl), the strength of the signal reduces when it passes through the switch.

Primitives Supported by Verilog

COMBINATIONAL LOGIC GATES			SWITCH-LEVEL PRIMITIVES		
Multi-input gates	Multi-output gates	Three-state gates	MOS transistor switches	MOS Pull gates	MOS Bi-directional switches
and	buf	bufif0	nmos	pullup	tran
nand	not	bufif1	pmos	pulldown	tranif0
or		notif0	cmos		tranif1
nor		notif0	rnmos		rtran
xor			rpmos		rtranif0
xnor			rcmos		rtranif1

notif1

Power and Ground

- The power (Vdd, logic 1) and Ground (Vss, logic 0) sources are needed when transistor level circuits are designed.
- Power and ground sources are defined with keywords supply1 and supply0.
- Sources of type supply1 are equivalent to Vdd in circuits and place a logical 1 on a net.
- Sources of the type supply0 are equivalent to ground or Vss and place a logical 0 on a net.
- Both supply1 and supply0 place logical 1 and 0 continuously on nets throughout the simulation.

Example:

- **supply1, supply0**

```
supply1 vdd;
supply0 gnd;

assign a = vdd; //Connect a to vdd
assign b = gnd; //Connect b to gnd
```

Signal Strengths

- Verilog's logic system has logic values and logic strengths
- The strength of a signal refers to the ability to act as a driver determining the resultant logic value on a net
  - gives additional information that determines the result of contending drivers on a net when multiple drives are present, or when the nets are modeled as charge storage capacitors
- The signals in gate-level models are "strong" by default

STRENGTHS

Strength Name	Strength Level	Element Modelled	Declaration Abbreviation
Supply Drive	7	Power supply connections.	supply
Strong Drive	6	Default gate & assign output strength.	strong
Pull Drive	5	Gate & assign output strength.	pull
Large Capacitor	4	Size of trireg net capacitor.	large
Weak Capacitor	3	Gate & assign output strength.	weak
Medium Capacitor	2	Size of trireg net capacitor.	medium
Small Capacitor	1	Size of trireg net capacitor.	small
High Impedence	0	Not Applicable.	highz

- **Output-side strength levels for different input-side strength values of rnmos, rpmos, and rcmos switches**

Input strength	Output strength
Supply – drive	Pull – drive
Strong – drive	Pull – drive
Pull – drive	Weak – drive
Weak – drive	Medium – capacitive
Large – capacitive	Medium – capacitive
Medium – capacitive	Small – capacitive
Small – capacitive	Small – capacitive
High impedance	High impedance

TIME DELAYS WITH SWITCH PRIMITIVES

1. MOS and CMOS switches

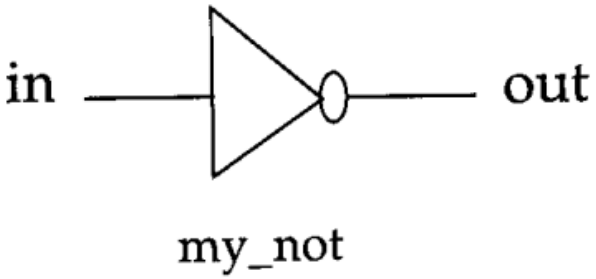
Switch Element	Delay Specification	Examples
pmos, nmos, rpmos, rnmos	Zero (no delay)	pmos p1(out, data, control);
	One (same delay on all transitions)	pmos #(1) p1(out, data, control);
	Two (rise, fall)	nmos #(1, 2) p2(out, data, control);
	Three (rise, fall, turnoff)	nmos #(1, 3, 2) p2(out, data, control);
cmos, rcmos	Zero, one, two, or three delays (same as above)	cmos #(5) c2(out, data, nctrl, pctrl);
		cmos #(1,2) c1(out, data, nctrl, pctrl);

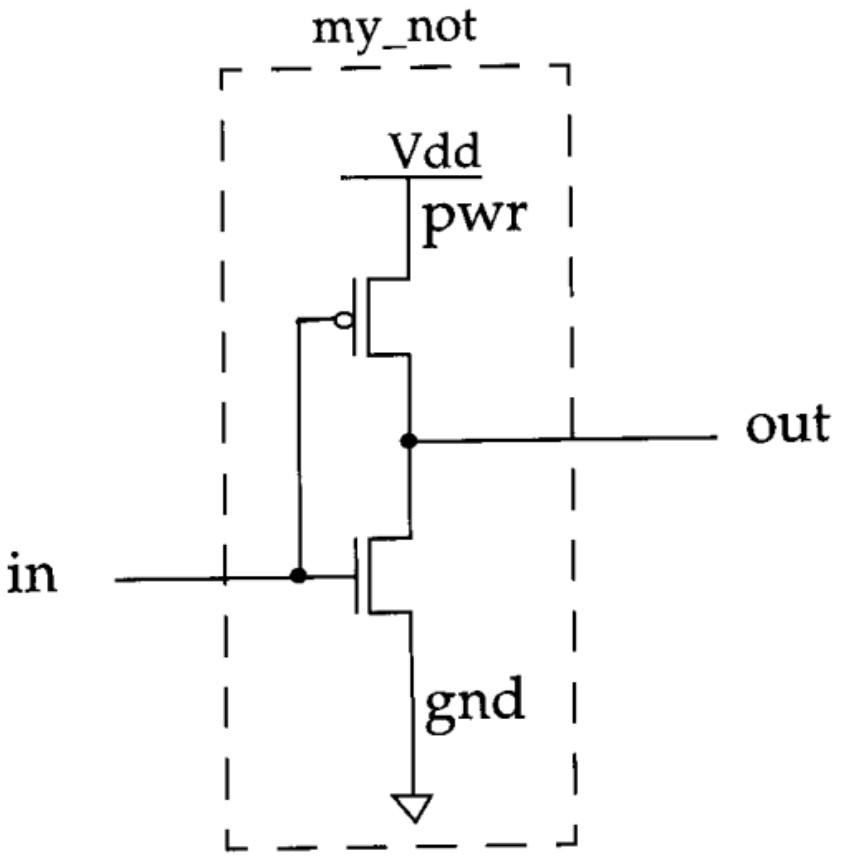
2. Bidirectional pass switches

Switch Element	Delay Specification	Examples
tran, rtran	No delay specification allowed	

tranif1, rtranif1 tranif0, rtranif0	Zero (no delay)
	
	One (both turn-on and turn-off)
	
	Two (turn-on, turn-off)
		rtranif0 rt1(inout1, inout2, control);
		
		tranif0 #(3) T(inout1, inout2, control);
		
		tranif1 #(1,2) t1(inout1, inout2, control);

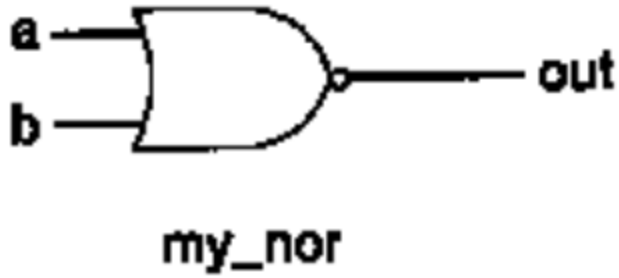
1. Verilog Model for CMOS Inverter

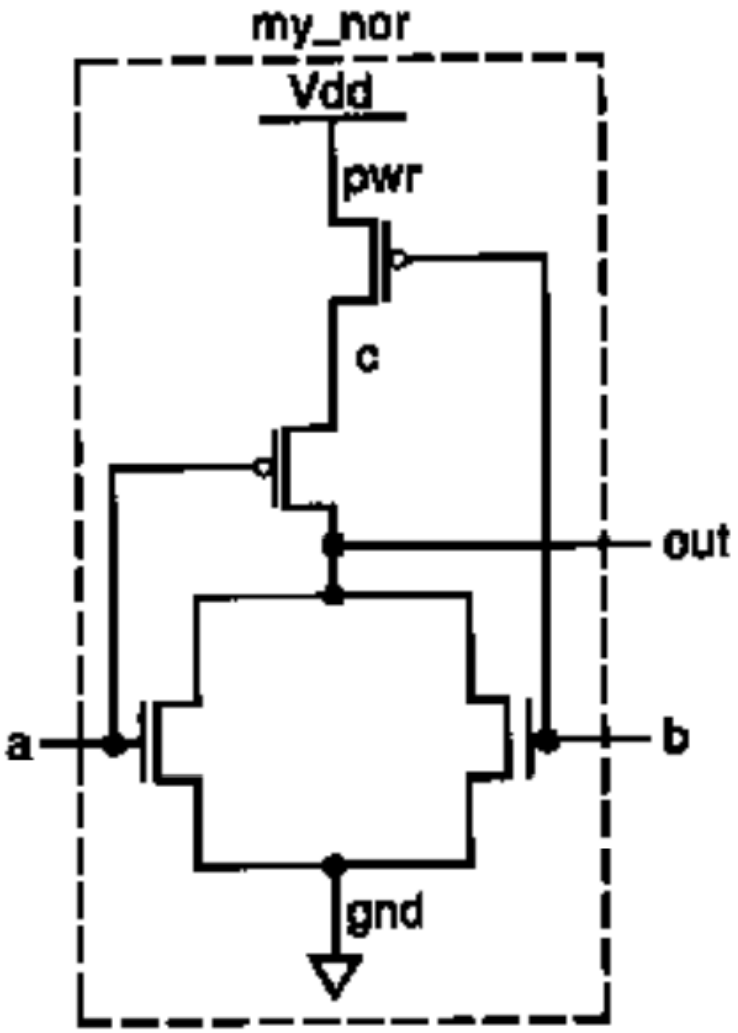




```
//Define an inverter using MOS switches
module my_not(out, in);

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);

endmodule
```

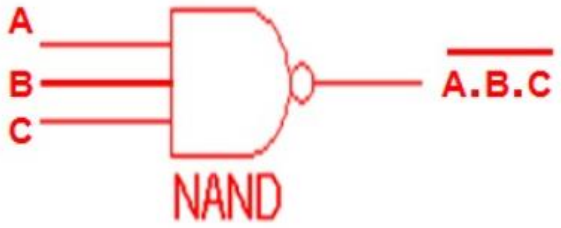
2. CMOS NOR gate



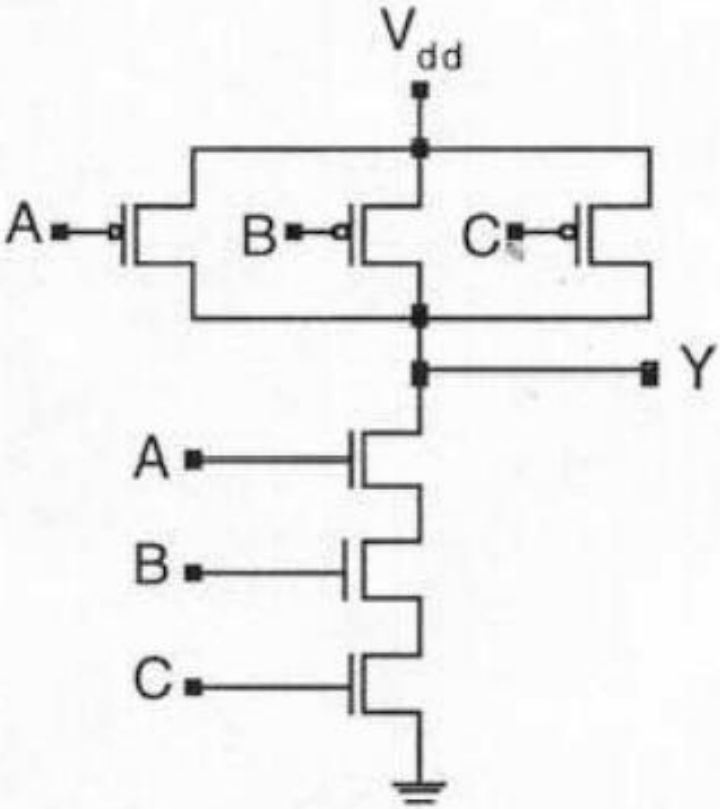


```
//Define our own nor gate, my_nor
module my_nor(out, a, b);

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
```

3. 3-input nand Gate



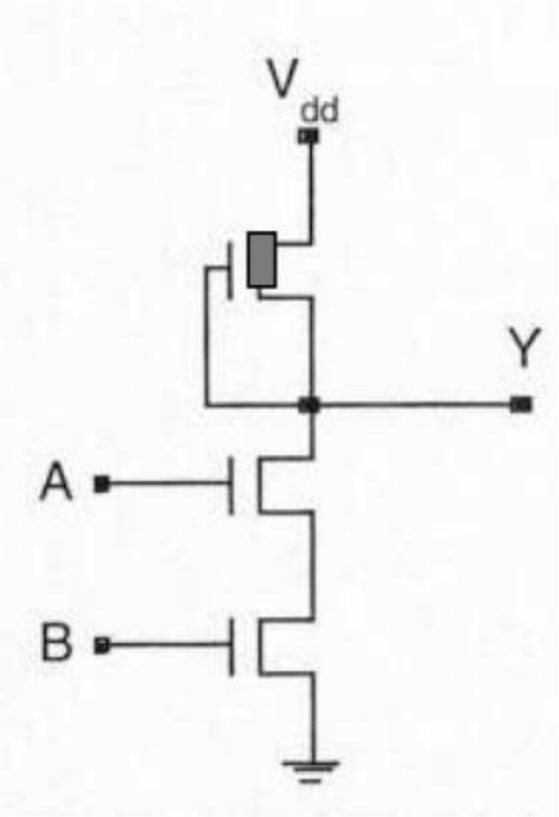

```
module nand_3 (Y, A, B, C);
  output        Y;
  input         A, B, C;
  supply0       GND;
  supply1       PWR;
  wire          w1, w2;

  pmos          (Y, PWR, A);
  pmos          (Y, PWR, B);
  pmos          (Y, PWR, C);
  nmos          (Y, w1, A);
  nmos          (w1, w2, B);
  nmos          (w2, GND, C);
endmodule
```



4. 2-input nand Gate using nmos

- nmos pull-down logic + depletion load pull-up device

```
module nmos_nand_2 (Y, A, B);
  output        Y;
  input         A, B;
  supply0       GND;
  tri           w1;

  pullup        (Y);
  nmos          (Y, w1, A);
  nmos          (w1, GND, B);
endmodule
```



THANK YOU