Chapter 2b: Switch-Level Modeling

Prof. Ming-Bo Lin

Department of Electronic Engineering
National Taiwan University of Science and Technology
Syllabus

- Objectives
- Switch primitives
- Delay specifications
- Signal strengths
Objectives

After completing this chapter, you will be able to

- Describe what is the structural modeling
- Describe how to instantiate switch primitives
- Describe how to model a design in switch primitives
- Describe how to specify delays in switches
- Describe other features of switch primitives
Syllabus

- Objectives
- **Switch primitives**
  - MOS switches
  - CMOS switch
  - Bidirectional switches
- Delay specifications
- Signal strengths
Switch Primitives

- Ideal switches – without a prefixed letter “r”
- Resistive switches – with a prefixed letter “r”

- MOS switches
  - nmos
  - pmos
  - cmos
- Bidirectional switches
  - tran
  - tranif0
  - tranif1
- Power and ground nets
  - supply1
  - supply0

- Resistive switches
  - rnmnos
  - rpmos
  - rcmos

- Resistive bidirectional switches
  - rtran
  - rtranif0
  - rtranif1

- Pullup and pulldown
  - pullup
  - pulldown
Syllabus

❖ Objectives

❖ Switch primitives
  ▪ MOS switches
  ▪ CMOS switch
  ▪ Bidirectional switches

❖ Delay specifications

❖ Signal strengths
The nmos and pmos Switches

- To instantiate switch elements
  
  \[
  \text{switch\_name} [\text{instance\_name}] (\text{output}, \text{input}, \text{control});
  \]

  - The instance\_name is optional

<table>
<thead>
<tr>
<th>nmos</th>
<th>control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>z</td>
</tr>
<tr>
<td>1</td>
<td>z</td>
</tr>
<tr>
<td>x</td>
<td>z</td>
</tr>
<tr>
<td>z</td>
<td>z</td>
</tr>
<tr>
<td>in</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>z</td>
<td>z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pmos</th>
<th>control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>z</td>
<td>z</td>
</tr>
</tbody>
</table>

(a) nMOS switch

(b) pMOS switch
Example 1 --- The CMOS Inverter

module mynot (input x, output f);
// internal declaration
supply1 vdd;
supply0 gnd;
// NOT gate body
    pmos p1 (f, vdd, x);
    nmos n1 (f, gnd, x);
endmodule
Example 2 --- CMOS NAND Gates

module my_nand (input x, y, output f);
supply1 vdd;
supply0 gnd;
wire a;
// NAND gate body
  pmos p1 (f, vdd, x);
  pmos p2 (f, vdd, y);
  nmos n1 (f, a, x);
  nmos n2 (a, gnd, y);
endmodule

(a) Circuit

(b) Logic symbol
Example 4 --- A Pseudo nMOS Gate

module my_pseudo_nor(input x, y, output f);
supply0 gnd;
  // Pseudo nMOS gate body
  nmos nx (f, gnd, x);
nmos ny (f, gnd, y);
pullup (f);
endmodule
Syllabus

- Objectives
- Switch primitives
  - MOS switches
  - CMOS switch
  - Bidirectional switches
- Delay specifications
- Signal strengths
CMOS Switch

- To instantiate CMOS switches:
  ```
  cmos [instance_name]
  (output, input, ncontrol, pcontrol);
  ```

- The `instance_name` is optional.

```
(a) Symbol
```

```
(b) Truth table
```

<table>
<thead>
<tr>
<th>control</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>n p</td>
<td>0 1 x z</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1 x z</td>
</tr>
<tr>
<td>0 1</td>
<td>z z z z</td>
</tr>
<tr>
<td>0 x</td>
<td>L H x z</td>
</tr>
<tr>
<td>0 z</td>
<td>L H x z</td>
</tr>
<tr>
<td>1 0</td>
<td>0 1 x z</td>
</tr>
<tr>
<td>1 1</td>
<td>0 1 x z</td>
</tr>
<tr>
<td>1 x</td>
<td>0 1 x z</td>
</tr>
<tr>
<td>1 z</td>
<td>0 1 x z</td>
</tr>
<tr>
<td>x 0</td>
<td>0 1 x z</td>
</tr>
<tr>
<td>x 1</td>
<td>L H x z</td>
</tr>
<tr>
<td>x x</td>
<td>L H x z</td>
</tr>
<tr>
<td>x z</td>
<td>L H x z</td>
</tr>
<tr>
<td>z 0</td>
<td>0 1 x z</td>
</tr>
<tr>
<td>z 1</td>
<td>L H x z</td>
</tr>
<tr>
<td>z x</td>
<td>L H x z</td>
</tr>
<tr>
<td>z z</td>
<td>L H x z</td>
</tr>
</tbody>
</table>

Digital System Designs and Practices Using Verilog HDL and FPGAs @ 2008~2010, John Wiley
An Example --- A 2-to-1 Multiplexer

```verilog
module my_mux (out, s, i0, i1);
output out;
input    s, i0, i1;
//internal wire
wire sbar; //complement of s
not (sbar, s);
//instantiate cmos switches
ccmos (out, i0, sbar, s);
ccmos (out, i1, s, sbar);
endmodule
```
Syllabus

- Objectives
- Switch primitives
  - MOS switches
  - CMOS switch
  - Bidirectional switches
- Delay specifications
- Signal strengths
Bidirectional Switches

- instance_name is optional

To instantiate bidirectional switches:

- tran [instance_name] (in, out);
- tranif0 [instance_name] (in, out, control);
- tranif1 [instance_name] (in, out, control);

- instance_name is optional
Chapter 2: Structural Modeling

Syllabus

- Objectives
- Switch primitives
- Delay specifications
  - MOS/CMOS switches
  - Bidirectional switches
- Signal strengths
Delay Specifications --- MOS/CMOS Switches

- **Specify no delay**
  
  ```
  mos_sw [instance_name](output, input, ...);
  cmos [instance_name](output, input, ...);
  ```

- **Specify propagation delay only**
  
  ```
  mos_sw #(prop_delay)[instance_name](output, input, ...);
  cmos #(prop_delay)[instance_name](output, input, ...);
  ```

- **Specify both rise and fall times**
  
  ```
  mos_sw #(t_rise, t_fall)[instance_name](output, input, ...);
  cmos #(t_rise, t_fall)[instance_name](output, input, ...);
  ```

- **Specify rise, fall, and turn-off times**
  
  ```
  mos_sw #(t_rise, t_fall, t_off)[instance_name](output, input, ...);
  cmos #(t_rise, t_fall, t_off)[instance_name](output, input, ...);
  ```
Syllabus

- Objectives
- Switch primitives
- Delay specifications
  - MOS/CMOS switches
  - Bidirectional switches
- Signal strengths
Delay Specifications --- Bidirectional Switches

- **Specify no delay**
  
  \[\text{bdsw name [instance name](in, out, control);}\]

- **Specify a turn-on and turn-off delay**
  
  \[\text{bdsw name #(t_on_off)[instance name](in, out, control);}\]

- **Specify separately turn-on and turn-off delays**
  
  \[\text{bdsw name #(t_on, t_off)[instance name](in, out, control);}\]
Syllabus

- Objectives
- Switch primitives
- Delay specifications
- Signal strengths
  - Signal strengths
  - trireg examples
## Signal Strengths

- Can be weakened or attenuated by the resistance of the wires

<table>
<thead>
<tr>
<th>Strength</th>
<th>Strength0</th>
<th>Strength1</th>
<th>Type</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>supply</td>
<td>supply0</td>
<td>supply1</td>
<td>driving</td>
<td>strongest</td>
</tr>
<tr>
<td>strong</td>
<td>strong0</td>
<td>strong1</td>
<td>driving</td>
<td></td>
</tr>
<tr>
<td>pull</td>
<td>pull0</td>
<td>pull1</td>
<td>driving</td>
<td></td>
</tr>
<tr>
<td>large</td>
<td>large0</td>
<td>large1</td>
<td>storage</td>
<td></td>
</tr>
<tr>
<td>weak</td>
<td>weak0</td>
<td>weak1</td>
<td>driving</td>
<td></td>
</tr>
<tr>
<td>medium</td>
<td>medium0</td>
<td>medium1</td>
<td>storage</td>
<td></td>
</tr>
<tr>
<td>small</td>
<td>small0</td>
<td>small1</td>
<td>storage</td>
<td></td>
</tr>
<tr>
<td>highZ</td>
<td>highz0</td>
<td>highz1</td>
<td>high Z</td>
<td>weakest</td>
</tr>
</tbody>
</table>
Single Strength Reduction

Input | Resistive switches | Output
--- | --- | ---
supply | | pull
strong | | weak
pull | | medium
large | | small
weak | | high-z
medium | | small
small | | high-z
high-z | |
Syllabus

- Objectives
- Switch primitives
- Delay specifications
- Signal strengths
  - Signal strengths
  - trireg nets
trireg Nets

- Driven state
- Capacitive state
An Example of trireg Net

❖ At simulation time 0
  - a, b, and c = 1
  - x = 0
  - y -> 0
  - z -> driven state and = strong0

❖ At simulation time 10
  - b = 0
  - y -> a high-impedance
  - z -> capacitive state and = medium0
An Example of Charge Sharing Problem

- At simulation time 0
  - \( a = 0 \)
  - \( b = c = 1 \)
  - \( x, y, \) and \( z = \text{strong1} \)

- At simulation time 10
  - \( b = 0 \)
  - \( y \rightarrow \text{capacitive state and} = \text{large1} \)
  - \( z \rightarrow \text{driven state and} = \text{large1} \)
An Example of Charge Sharing Problem

- At simulation time 20
  - \( c = 0 \)
  - \( z \rightarrow \) capacitive state and = small1

- At simulation time 30
  - \( c = 1 \) again,
  - \( y \) and \( z \) share the charge

- At simulation time 40
  - \( c = 0 \)
  - \( z \rightarrow \) capacitive state and = small1