Digital abstraction: MOS abstraction as switch – CMOS Inverter-

Andreas G. Andreou
Pedro Julian

Electrical and Computer Engineering
Johns Hopkins University

http://andreoulab.net
The inverter is probably the most important basic logic cell in circuit design. This chapter introduces the logical concepts of the inverter, its layout implementation, the link between the transistor size and the static and analog characteristics. The manual design of the inverter is detailed.

The performances of the inverter are analyzed in terms of static transfer function, switching speed, MOS options influence, and power consumption.

1. Logic symbol

Two logic symbols are often used to represent the inverter: the "old style" inverter (left of figure 4-1), and the IEEE symbol (right of figure 4-1). In DSCH, we preferably use traditional symbol layout. As the logic truth table of figure 4-1 shows, the cell inverts the logic value of the input \( \text{In} \) into an output \( \text{Out} \).

<table>
<thead>
<tr>
<th>\text{In}</th>
<th>\text{Out}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

In the truth table, the symbol 0 represents 0.0V while 1 represents the logic supply, which is 1.2V in 0.12\( \mu \)m. The symbol \( X \) means "undefined". This state is equivalent to an undefined voltage, just like with a floating input node without any input connection. The undefined state appears in gray in the simulations and chronograms.

\[ \text{Out} = \text{NOT} (\text{In}) \]
\[ \text{Out} = \sim (\text{In}) \]
The digital abstraction (I)

1. Making bits concrete
2. What makes a good bit
3. Getting bits under contract

from Lecture 2 in MIT 6.004 Computation Structures http://6004.csail.mit.edu/
Concrete encoding of information

We can build upon almost any physical phenomenon...

Wait! Those last ones might have potential...

- lanterns
- dominos
- engraved stone tablets
- Billiard balls
- E. Coli
- polarization of a photon

from Lecture 2 in MIT 6.004 Computation Structures http://6004.csail.mit.edu/
The digital abstraction (II)

Keep in mind that the world is not digital, we would simply like to engineer it to behave that way. Furthermore, we must use real physical phenomena to implement digital designs!

from Lecture 2 in MIT 6.004 Computation Structures http://6004.csail.mit.edu/
The digital abstraction (III)

- Divide voltage into discrete regions
  - Logic 0
  - Logic 1
  - X - between 0 and 1
  - Out of range
    - may damage devices

- Each logic gate *restores* the signal
  - Output noise < input noise
  - Noise is not cumulative
    - In fact it is attenuated
  - Noise margin
    - How much noise won't change output
MOS transistor encoding bits

Rather than worrying about the precise voltages on the terminals of the transistor, guarantee that voltages will fall within two regions, one represents a logic ‘0’ and the other a ‘1’.

Need to compute the output only for inputs in the allowable range
- Much simpler than before
- Model transistor as being either conducting, or off

Need to ensure that the output is always in the allowable voltage range
- Need to make sure you produce valid digital outputs to the next stage
- Also want to have level restore

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
Input-Output voltage ranges and noise

$V_{IL}$: Maximum input voltage recognized as logic 0

$V_{IH}$: Minimum input voltage recognized as logic 1

$V_{OL}$: Maximum output voltage corresponding to logic 0

$V_{OH}$: Minimum output voltage corresponding to logic 1

- Consider Inverter
  - Suppose Output = $V_{OH}$
  - Suppose this inverter output connected to another inverter input
  - If $V_{OH} < V_{IH}$: PROBLEM
    - $V_{OH}$ better be > $V_{IH}$
  - $V_{OH} - V_{IH} = \text{NMH (noise margin high)}$
  - $\text{NML (Noise margin Low)}$: similar

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
Simplest Model for MOS transistor switch

\[ \text{NMOS transistors} \quad \begin{array}{c} G \quad \text{D} \\ S \end{array} = \quad \begin{array}{c} D \quad \text{G=1 (Vdd)} \\ S \end{array} \quad \begin{array}{c} G \quad \text{D} \\ S \end{array} = \quad \begin{array}{c} D \quad \text{G=0 (Gnd)} \\ S \end{array} \quad \begin{array}{c} \text{PMOS transistors} \end{array} \]

- Let Logic value 1 be Vdd, value 0 be Gnd
- NMOS devices are switches
  - G is 1 -> the drain D and source S are connected
  - G is 0 -> the drain D and source S are not connected
- PMOS devices are switches
  - G is 0 -> the drain D and source S are connected
  - G is 1 -> the drain D and source S are not connected

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
Terminology

Note that the source and drain terminals are really the same, but by convention the source terminal is the one with the lower voltage on it. Thus, the maximum voltage between the gate and the \{source, drain\} is the voltage between the source and the gate. (This fact will be important later.)

The voltage on the gate controls the connection between the source and the drain. When the gate is high, the source and drain are connected together. When it is low, the terminals are disconnected.

**CAUTION**: do NOT use the words “open” and “closed” to describe switches. Is open an open electrical circuit (no flow), or an open fluid valve (flow)? You get opposite results, depending on which analogy you use.

This description is for nMOS transistors. For pMOS everything is reversed. The source is the higher voltage terminal, and the transistor is on when the gate is much lower than the source. More on pMOS later

adapted from Introduction to VLSI  Stanford University by Subhasish Mitra
Transistor Examples

The state of these transistors:

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
Switch networks (I)

- Since transistors can be modeled as switches
  - Look at what we can make out of switches
  - Draw an abstract switch as
    - Control (gate) terminal is on top
- Can build switch networks
  - Are not logic gates themselves!!!
  - Are a collection of switches that still have two non-control terminals
    - Define function of a switch network as the inputs conditions that connect the two non-control terminals of the network
- Structure of switch network sets its logic functions:
  - ‘OR’ functions are constructed by parallel switches
  - ‘AND’ function are constructed by series switches

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
Switch networks (II)

- The function of a switch network is true when the two terminals of the network are connected together. Since for parallel switches the terminals are connected if either switch is on, the function is OR. For series switches the network is conducting only if both switches are on, hence an AND.

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
General switch networks

- More complex connections are possible
- Composition rules are simple. Use a recursive definition:
  - Parallel combination of switch networks yields an OR of the component switch networks’ functions
  - Series combination of switch networks yields an AND of the component switch networks’ functions.

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
Building functions out of switches

- Have switches, but we want to build logic gate

- Logic gate abstraction:
  - Unidirectional
    - Inputs drive output, but output does not drive inputs
  - Digital gate
    - Reduces input noise
  - Operation completely described by a boolean operator

- Not a difficult task if you follow a few simple rules
  - Connect switches so output is always driven to either Vdd, Gnd
    - Output noise should be small, if power supply noise is small
  - GND and Vdd are never connected to one another
    - Don’t want to short the power supply out.
Basic idea: Build two switch networks: one good for 1s and one good for 0s.

- Requirement 1: output is always driven by one of the two branches.
- Requirement 2: output is never driven by both at the same time.
- Make the switch functions complementary.

adapted from Introduction to VLSI, Stanford University by Subhasish Mitra.
The genesis of the inverter!

- Meets the rules
  - Out is always driven (Either pMOS or nMOS is always active)
  - Vdd and Gnd are never shorted
    - At least with valid inputs

adapted from Introduction to VLSI  Stanford University by Subhasish Mitra

More gates next lecture!
NMOS is good for “transmitting” 0s: because current from Drain to Source pulls low

- Operates on one of two curves
  - on
    - Looks like a current source initially (high $V_{ds}$)
    - Looks like a resistor later (low $V_{ds}$)
  - off
    - Open circuit always

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
PMOS is good for “transmitting” 1s: because current from Drain to Source pulls high

- Same behavior as NMOS
  - Open circuit when off
  - Current source or resistor when on

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
CMOS pass transistor structures

- Concept of switch is really versatile
  - But if we don’t know the value being switched, neither NMOS nor PMOS alone implements it
- Solution: use nMOS and pMOS in parallel
  - Drive gates with complementary signals
  - Completely bidirectional
- How can we take advantage of this?

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
Transmission gate muxes

- Arbitrary number of inputs can be muxed together
  - As long as selects are mutually exclusive
- Can be cascaded in series to make more complicated networks
- CAREFUL: Not restoring as drawn (no gain)
  - Inverter often used to buffer output
- More on using transmission gates later

adapted from Introduction to VLSI Stanford University by Subhasish Mitra
## Logic Levels Definitions

<table>
<thead>
<tr>
<th>Logical Value</th>
<th>Voltage</th>
<th>Name</th>
<th>Symbol in DSCH</th>
<th>Symbol in Microwind</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0V</td>
<td>VSS</td>
<td><img src="image" alt="Symbol" /></td>
<td><img src="image" alt="Symbol" /></td>
<td>(Green in logic simulation)</td>
</tr>
<tr>
<td>1</td>
<td>1.2V in cmos 0.12μm</td>
<td>VDD</td>
<td><img src="image" alt="Symbol" /></td>
<td><img src="image" alt="Symbol" /></td>
<td>(Red in analog simulation)</td>
</tr>
<tr>
<td>X</td>
<td>Undefined</td>
<td>X</td>
<td>(Gray in simulation)</td>
<td>(Gray in simulation)</td>
<td></td>
</tr>
</tbody>
</table>
MOS transistor

At 0 Volt, no electrical connection between drain and source.

At high voltage, electrons are attracted below the gate, creating a channel.

NMOS off

NMOS on
NMOS and PMOS digital “models”

<table>
<thead>
<tr>
<th>Gate</th>
<th>Source</th>
<th>Drain</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gate</th>
<th>Source</th>
<th>Drain</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
</tbody>
</table>
what does this all mean?

- **nMOS device**
  - Drain floating
  - Channel off
  - Channel on
  - Pass 0
    - 0 Ron 0
  - Pass 1
    - 1 Ron 1

- **pMOS device**
  - Drain connected to source
  - Channel on
    - 0 Ron 0
  - Channel off
    - 1 Ron 1
microwind

\[ \lambda = \frac{L_{\text{min}}}{2} \]

0.12 \text{ um CMOS process}
and back to layout

Figure 2-14: Bird's view of the n-channel and p-channel MOS device layout (allMosDevices.MSK)
cross-section and zoom

Figure 2-15: Vertical cross-section of an n-channel and p-channel MOS devices in 0.12μm technology
MOS transistor layout

Optimal poly thickness: $2\lambda$

Minimum $4\lambda$

N+ diffusion

Minimum extra poly: $3\lambda$

Polysilicon gate

nMOS with contacts

Gate contact from poly to metal

Minimum $4\lambda$ between two metal interconnects

Source contact from N+ diffusion to metal

Drain contact from N+ diffusion to metal
<table>
<thead>
<tr>
<th>Layer name</th>
<th>Code</th>
<th>Description</th>
<th>Color in Microwind</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polysilicon</td>
<td>Poly</td>
<td>Gate of the n-channel and p-channel MOS devices</td>
<td>Red</td>
</tr>
<tr>
<td>N+ diffusion</td>
<td>Diffn</td>
<td>Delimits the active part of the n-channel device. Also used to polarize the N-well</td>
<td>Dark green</td>
</tr>
<tr>
<td>P+ diffusion</td>
<td>Diffn</td>
<td>Delimits the active part of the p-channel device. Also used to polarize the bulk</td>
<td>Maroon</td>
</tr>
<tr>
<td>Contact</td>
<td>Contact</td>
<td>Makes the connection between diffusions and metal for routing. The contact plug is fabricated by drilling a hole in the oxide and filling the hole with metal.</td>
<td>White cross</td>
</tr>
<tr>
<td>First level of metal</td>
<td>Metal1</td>
<td>Used to rout devices together, in order to create the logic or analog function</td>
<td>Blue</td>
</tr>
<tr>
<td>N well</td>
<td>Nwell</td>
<td>Low doped diffusion used to invert the doping of the substrate. All p-channel MOS are located within N well areas.</td>
<td>Dotted green</td>
</tr>
</tbody>
</table>
dynamic behavior: another level of abstraction

Demo: MOSN.msk
and the perfect switch!
simulation of the perfect switch

Demo: TGATE.msk
CMOS Inverter

- NFET’s pull down, PFET’s pull up
- Pull up and pull down NOT at the same time
- Output always connected to VDD or GND

<table>
<thead>
<tr>
<th>X</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(b)  
(c)  
(d)  
\[ D = \overline{X} \]
multiple contacts: why?

Fig. 2-46. A strong current through a single contact could damage the metal structure.
multiple contacts

Fig. 2-47. A single contact cannot handle more than 1mA. A series of contacts is preferred. (MosLayout.MSK)
multiple contacts

Fig. 2-48. A series of contacts also reduced the serial access resistance (MosContacts.MSK)
Computer Aided Design Tools

MICROWIND Tool Design Flow

MICROWIND 3

Schematic Modeling
Analog & digital Library models

Digital Simulation
Verilog Extraction
SPICE Extraction

SPICE Simulator
(3rd Party)

LTSpice
WindSpice

MICROWIND Tool

3rd Party Tools

Contact:
Sales: sales@microwind.net
Support: support@microwind.net

http://www.microwind.net