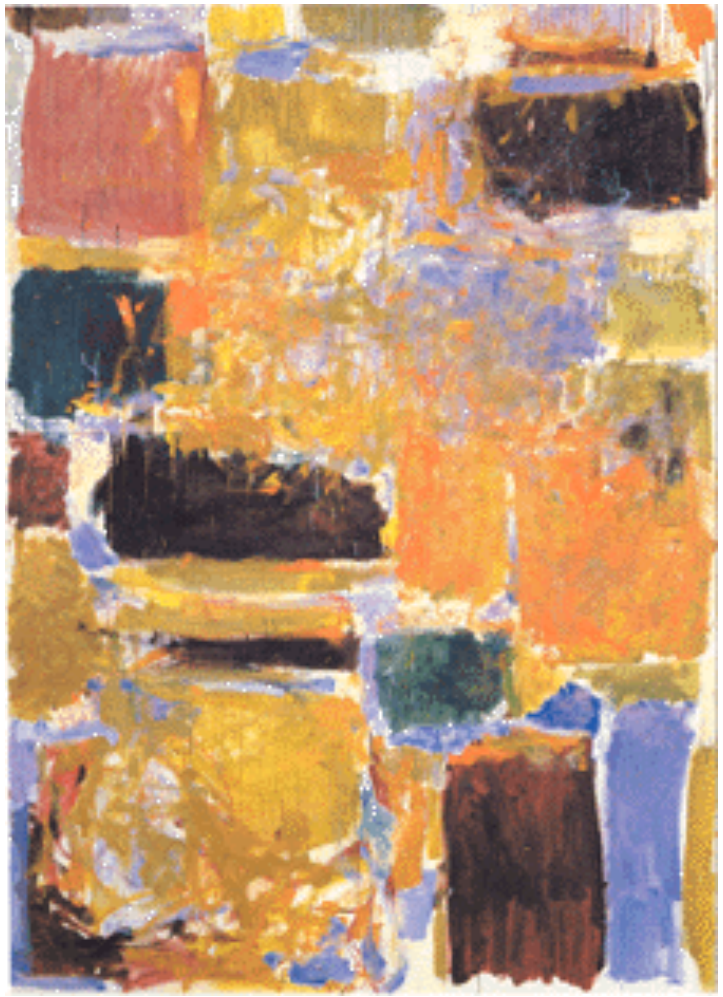


EECS 151/251A
Fall 2017
Digital Design and Integrated Circuits

Instructors:
Weaver and Wawrzynek

Lecture 8

Administration

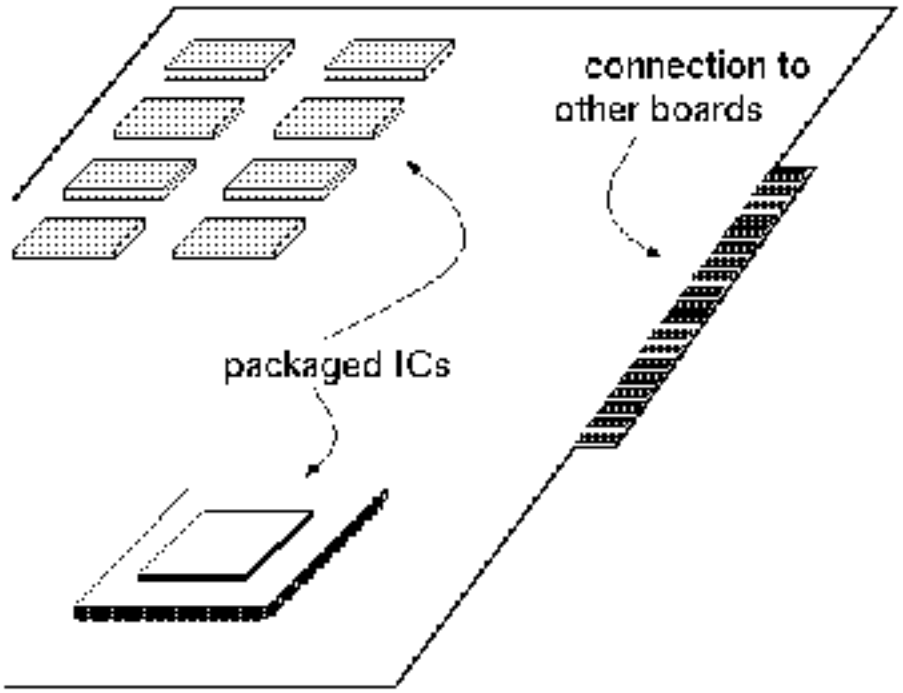
- Exam in One Week
  - Take here in class and with an extra 30 minutes (5:30-7:30).
  - Covers topics: beginning through 2/6.

Overview of Physical Implementations

The stuff out of which we make systems.

- Integrated Circuits (ICs)
  - Combinational logic circuits, memory elements, analog interfaces.
- Printed Circuits (PC) boards
  - substrate for ICs and interconnection, distribution of CLK, Vdd, and GND signals, heat dissipation.
- Power Supplies
  - Converts line AC voltage to regulated DC low voltage levels.
- Chassis (rack, card case, ...)
  - holds boards, power supply, fans, provides physical interface to user or other systems.
- Connectors and Cables.

Printed Circuit Boards

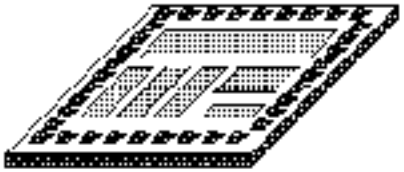


- fiberglass or ceramic
- 1-25 conductive layers
- ~1-20in on a side
- IC packages are soldered down.

Multichip Modules (MCMs)

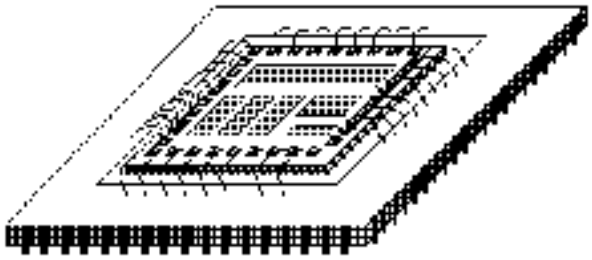
- Multiple chips directly connected to a substrate. (silicon, ceramic, plastic, fiberglass) without chip packages.

Integrated Circuits



- Primarily Crystalline Silicon
- 1mm - 25mm on a side
- 100 - 20B transistors
- (25 - 250M “logic gates”)
- 3 - 10 conductive layers
- 2018 state-of-the-art feature size 7nm = 0.007 x 10^{-6} m
- “CMOS” most common - complementary metal oxide semiconductor

Chip in Package



- Package provides:
  - spreading of chip-level signal paths to board-level
  - heat dissipation.
- Ceramic or plastic with gold

From Gates to Circuits

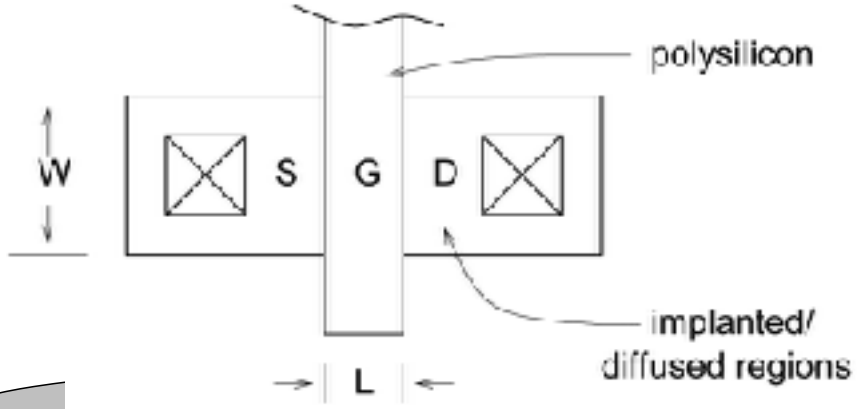
- Digital abstraction
- CMOS abstraction
- Switch logic
- Transient properties

CMOS abstraction

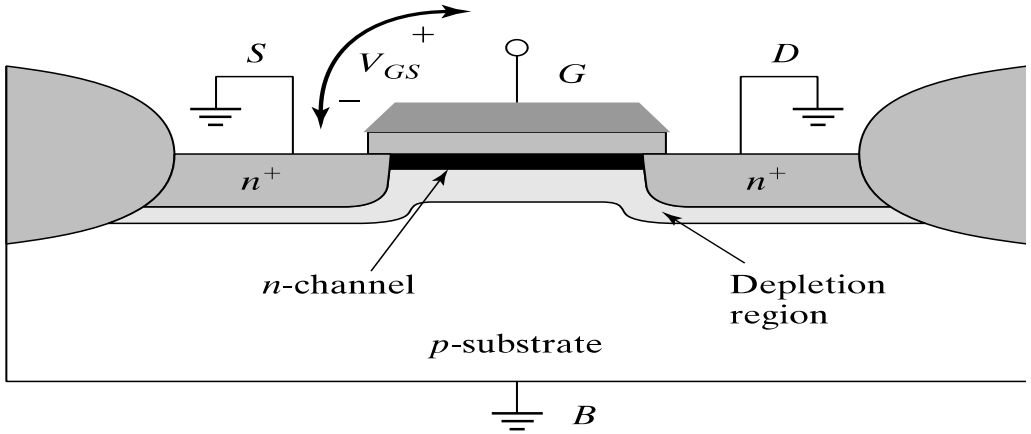
CMOS Devices

- MOSFET (Metal Oxide Semiconductor Field Effect Transistor).

Top View

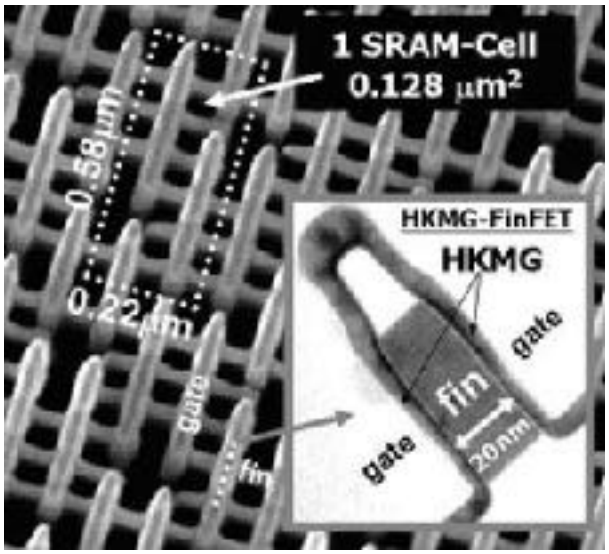


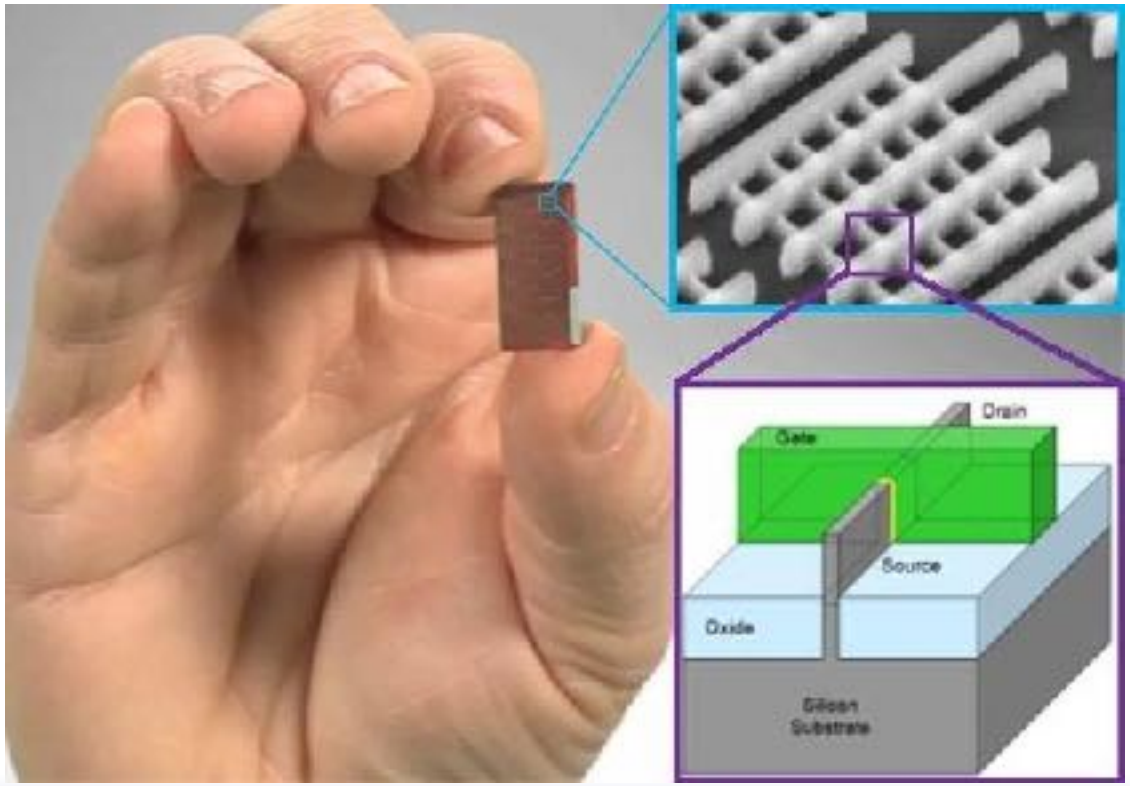
Cross-section View



The gate acts like a capacitor. A high voltage on the gate attracts charge into the channel. If a voltage exists between the source and drain a current will flow. In its simplest approximation, the device acts like a switch.

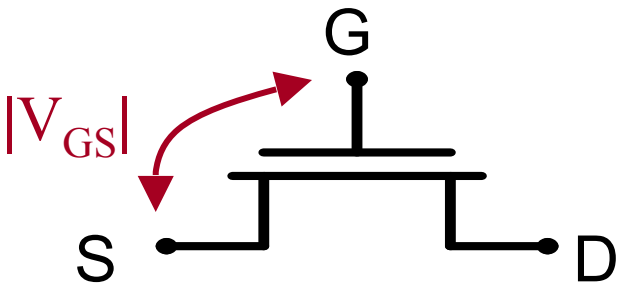
CMOS Transistors – State-of-the-Art

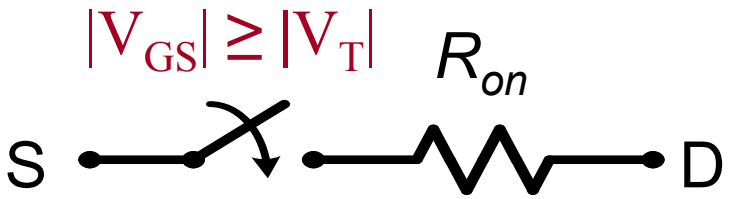




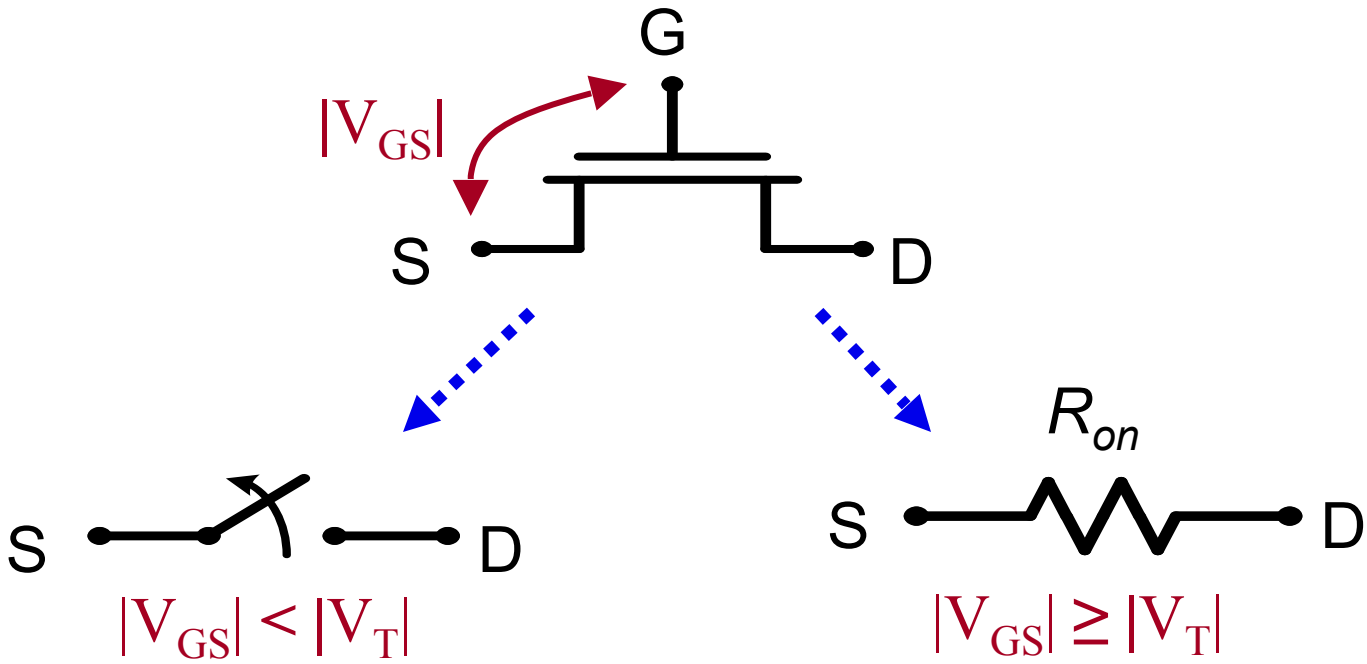
MOS Transistor as a Switch

MOS Transistor ⟷ A Switch!





ON/OFF Switch Model of MOS Transistor

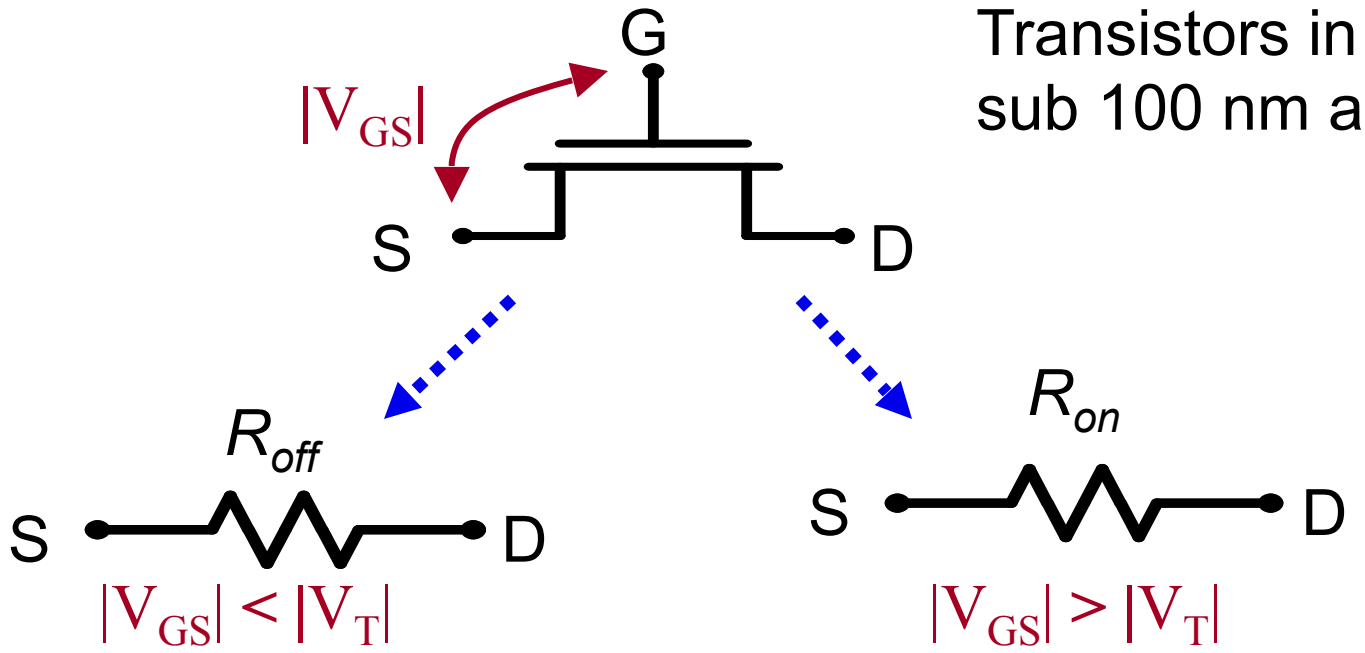


Drain versus Source - Definition

MOS transistors are symmetrical devices
(Source and drain are interchangeable)

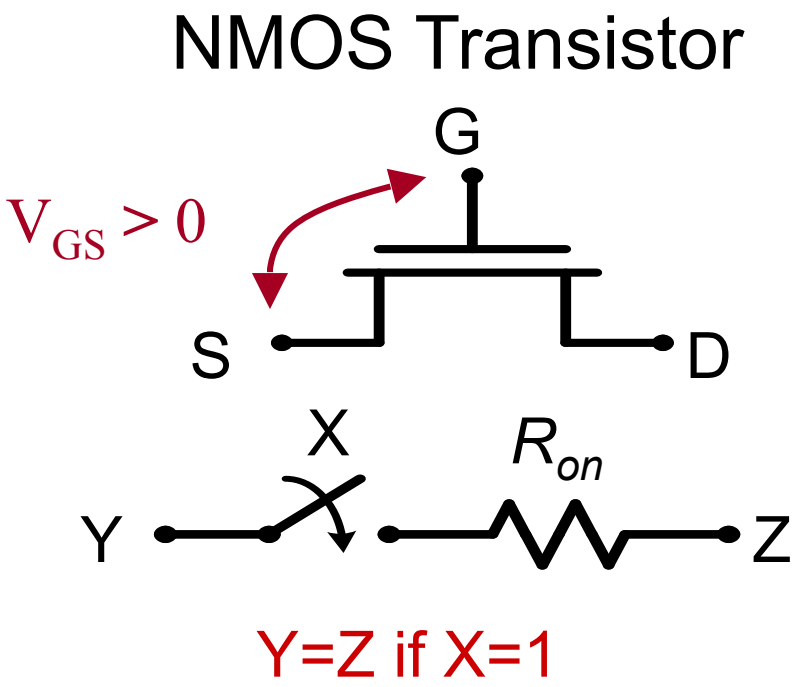
Source is the node w/ the lowest voltage

A More Realistic Switch

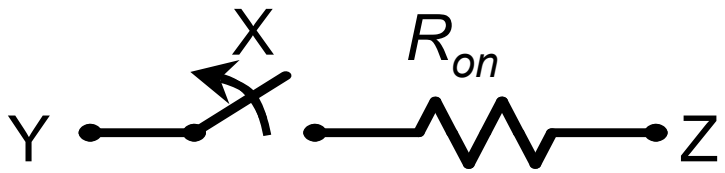


Transistors in the sub 100 nm age

A Logic Perspective

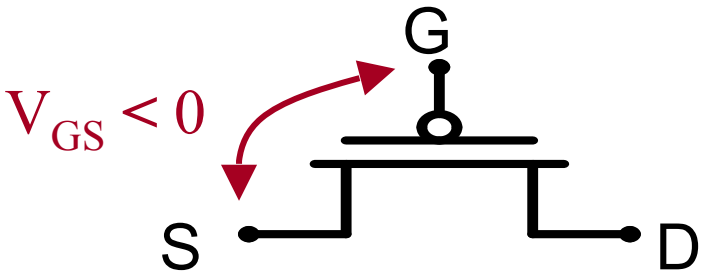


A Complementary Switch



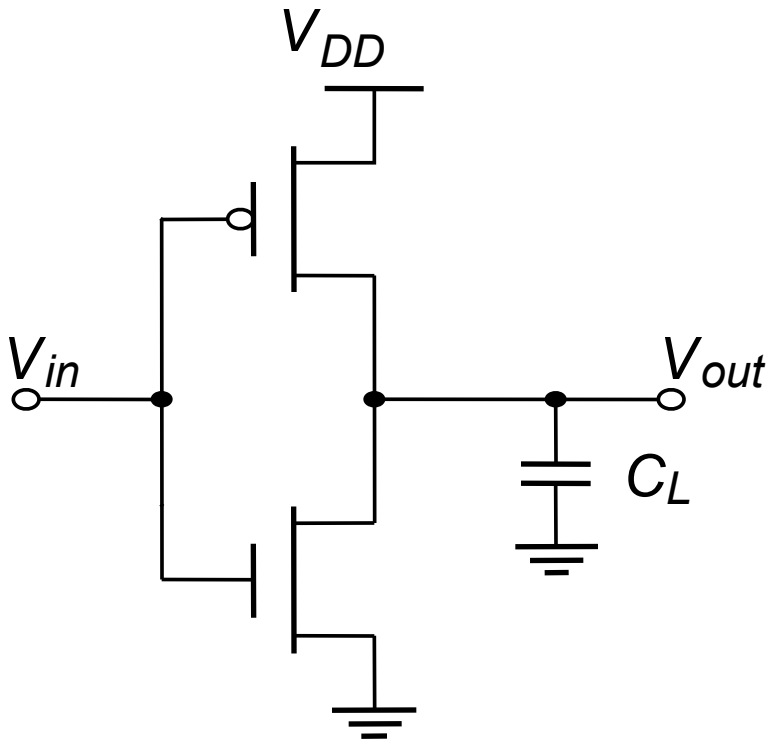
Y=Z if X=0

PMOS Transistor

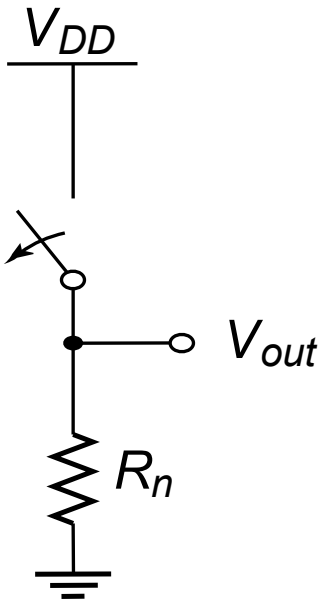


Source is the node w/ the highest voltage!

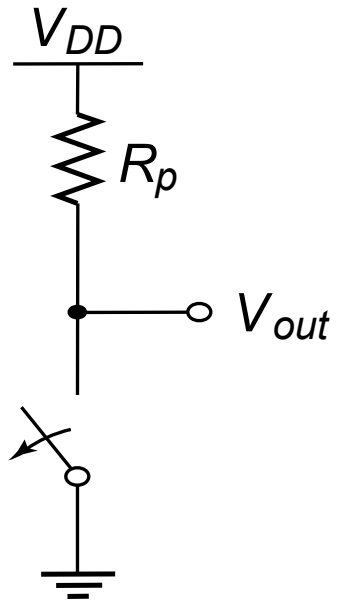
The CMOS Inverter: A First Glance



The Switch Inverter
First-Order DC Analysis



$V_{in} = V_{DD}$



$V_{in} = 0$

$V_{OL} = 0$

$V_{OH} = V_{DD}$

$V_M = f(R_n, R_p)$

Switch logic

Static Logic Gate

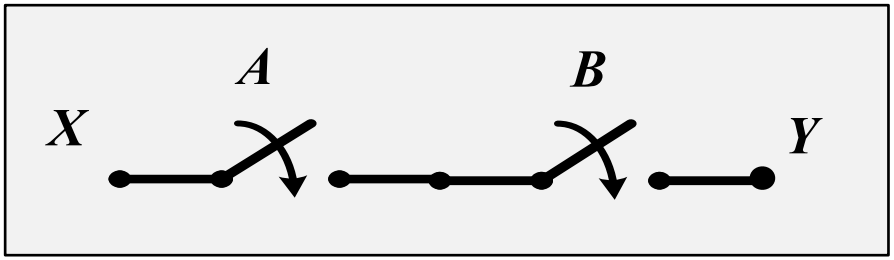
- At every point in time (except during the switching transients) each gate output is connected to either V_{DD} or V_{SS} via a low resistive path.

- The output of the gate assumes at all times the value of the Boolean function implemented by the circuit (ignoring, once again, the transient effects during switching periods).

Example: CMOS Inverter

Building logic from switches

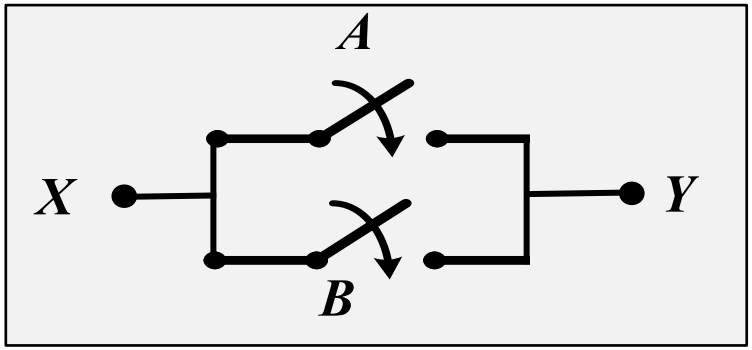
Series



AND

$Y = X$ if A AND B

Parallel



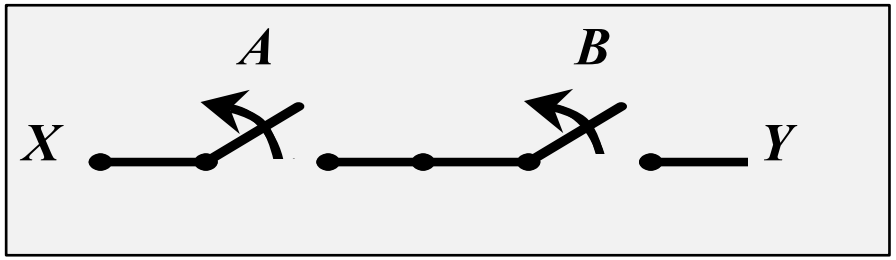
OR

$Y = X$ if A OR B

(output undefined if condition not true)

Logic using inverting switches

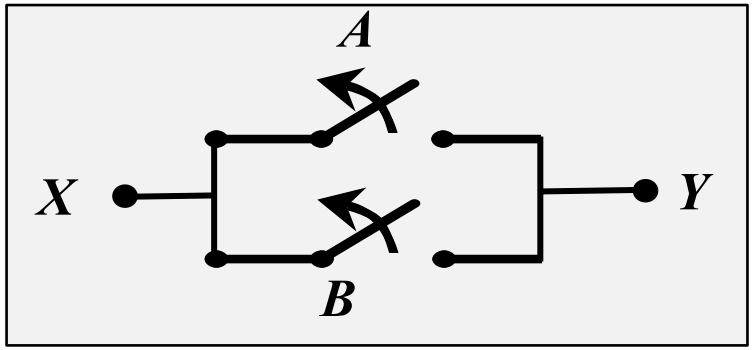
Series



NOR

$$Y = X \text{ if } \overline{A} \text{ AND } \overline{B}$$
$$= \overline{A + B}$$

Parallel

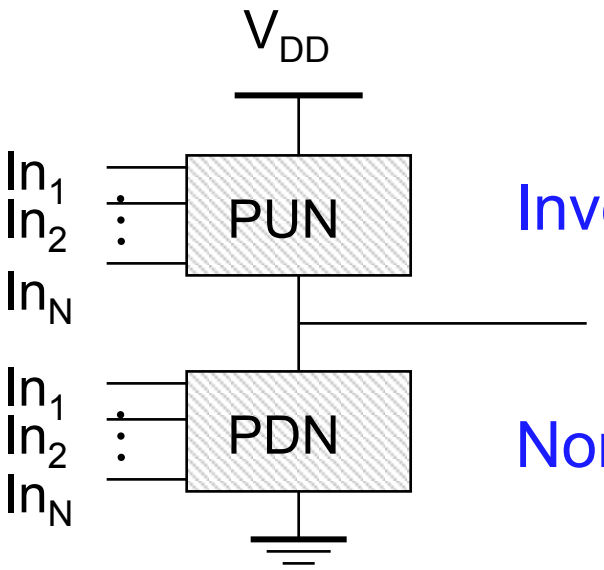


NAND

$$Y = X \text{ if } \overline{A} \text{ OR } \overline{B}$$
$$= \overline{AB}$$

(output undefined if condition not true)

Static Complementary CMOS



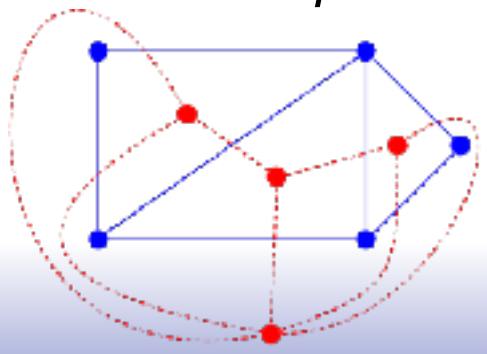
Inverting switches

$F(In_1, In_2, \ldots In_N)$

Non-Inverting switches

PUN and PDN are dual logic networks
PUN and PDN functions are complementary

Dual Graphs

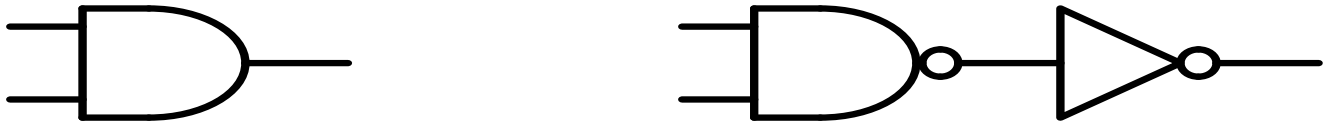


Complementary CMOS Logic Style

- PUN is the **dual** to PDN
  (can be shown using DeMorgan's Theorems)

$$\overline{A + B} = \bar{A}\bar{B}$$
$$\overline{AB} = \bar{A} + \bar{B}$$

- Static CMOS gates are always inverting

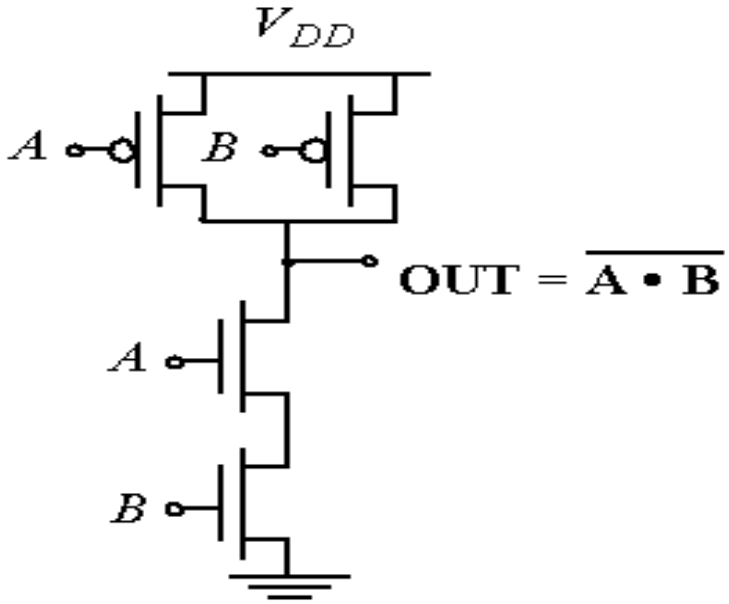


AND = NAND + INV

Example Gate: NAND

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

Truth Table of a 2 input NAND gate

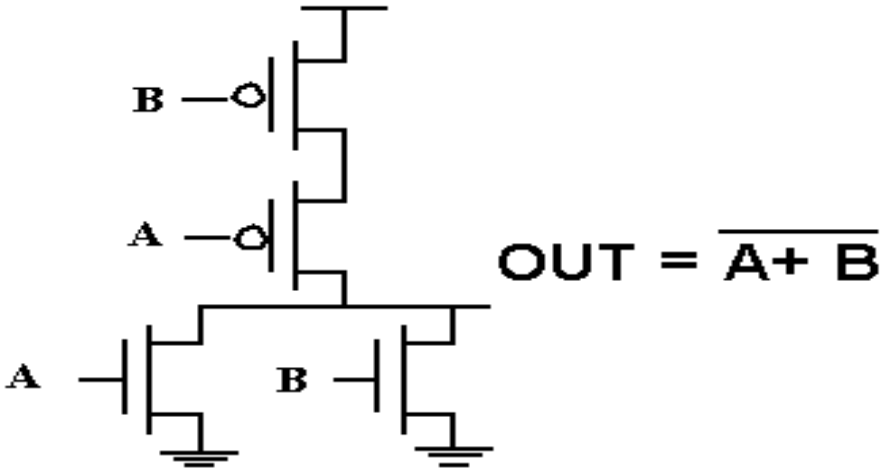


- PDN: G = AB $\Rightarrow$ Conduction to GND
- PUN: F = $\overline{A}$ + $\overline{B}$ = $\overline{AB}$ $\Rightarrow$ Conduction to V_{DD}
- $\overline{G(In_1,In_2,In_3,\ldots)} \equiv F(\overline{In_1},\overline{In_2},\overline{In_3},\ldots)$

Example Gate: NOR

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

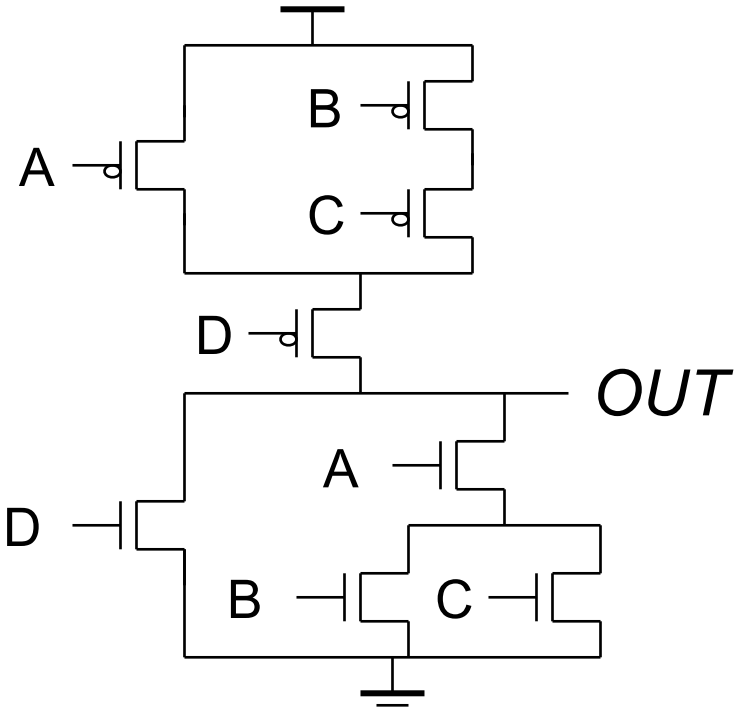
Truth Table of a 2 input NOR gate



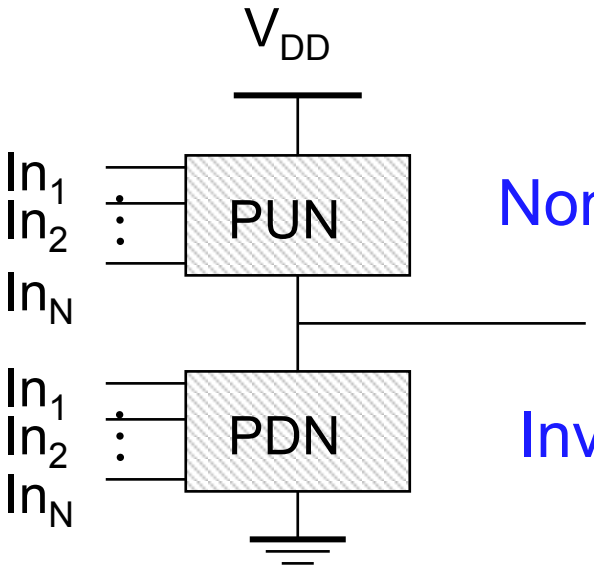
Complex CMOS Gate

$$OUT = \overline{D + A \bullet (B + C)}$$

$$OUT = \overline{D \bullet A + B \bullet C}$$



Non-inverting logic



Non-Inverting switches

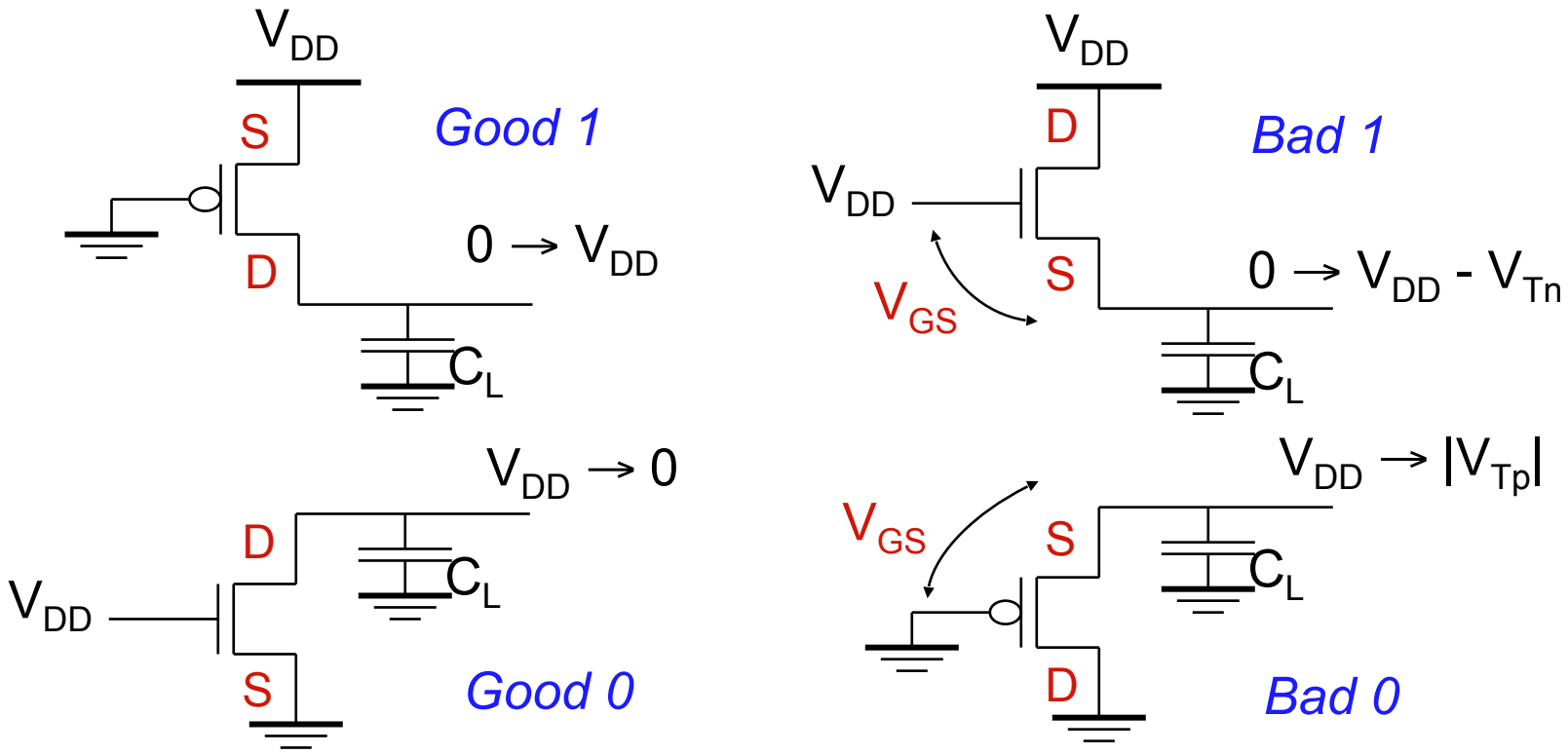
$F(In_1,In_2,\ldots In_N)$

Inverting switches

Why is this a bad idea?

PUN and PDN are dual logic networks
PUN and PDN functions are complementary

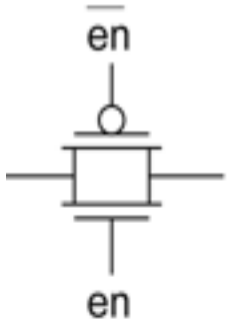
Switch Limitations



Tough luck …

Transmission Gate

- Transmission gates are the way to build “switches” in CMOS.
- In general, both transistor types are needed:
  - nFET to pass zeros.
  - pFET to pass ones.
- The transmission gate is bi-directional (unlike logic gates).

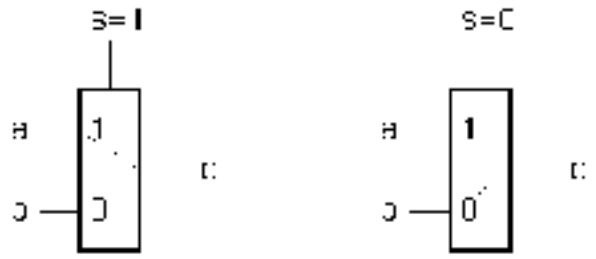


- Does not directly connect to Vdd and GND, but can be combined with logic gates or buffers to simplify many logic structures.

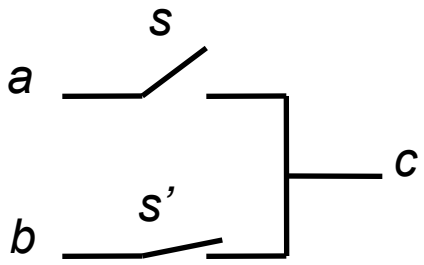
Transmission-gate Multiplexor

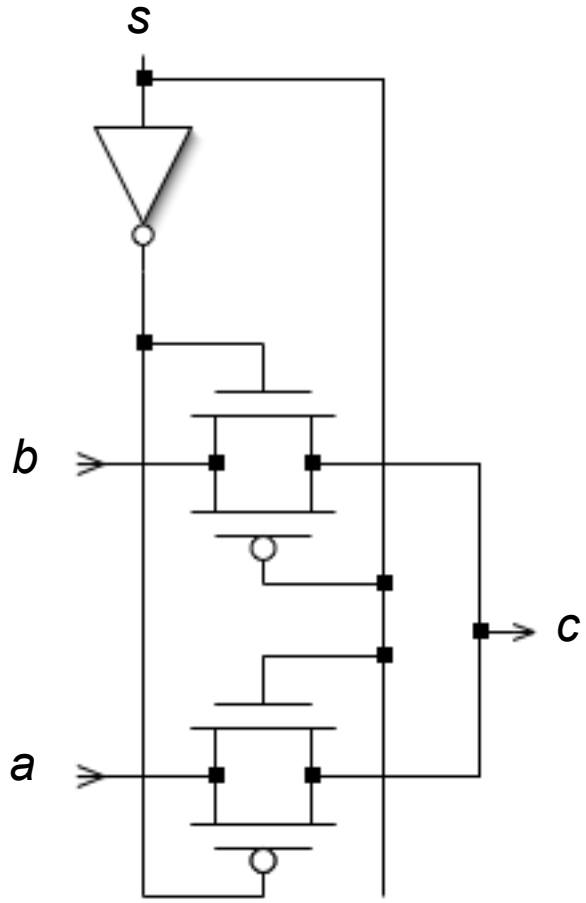
2-to-1 multiplexor:

$c = sa + s'b$



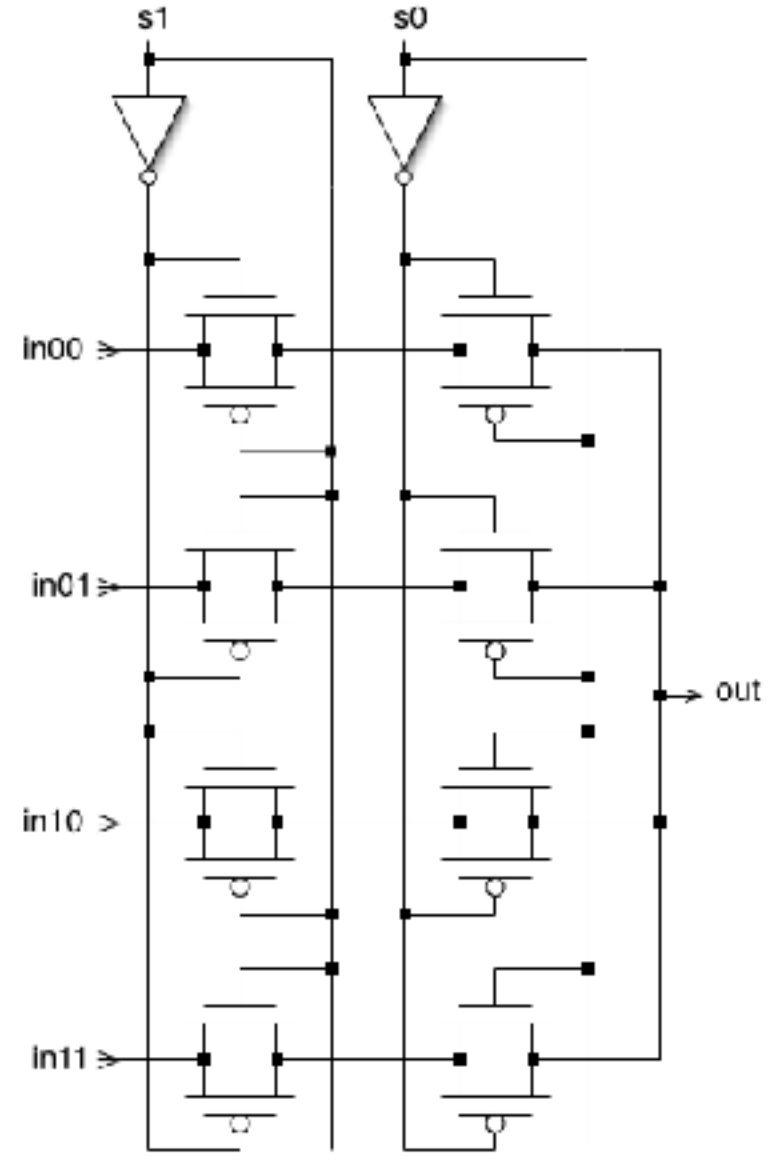
Switches simplify the implementation:





Compare the cost to logic gate implementation.

4-to-1 Transmission-gate Mux

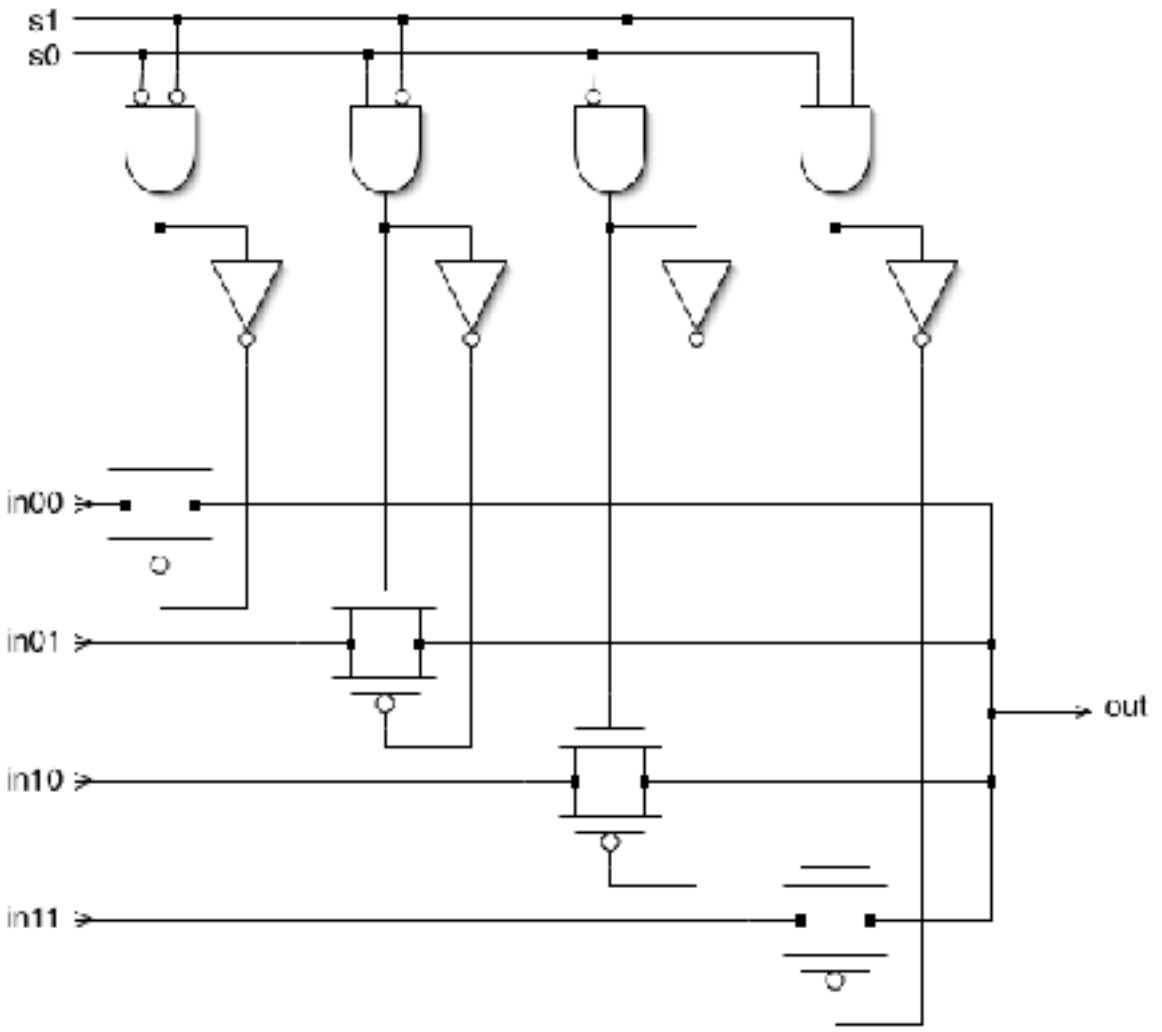


- The series connection of pass-transistors in each branch effectively forms the AND of s1 and s0 (or their complement).
- Compare cost to logic gate implementation

Any better solutions?

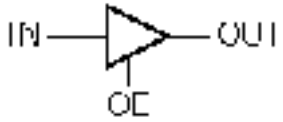
Alternative 4-to-1 Multiplexor

- This version has less delay from in to out.
- In both versions, care must be taken to avoid turning on multiple paths simultaneously (shorting together the inputs).



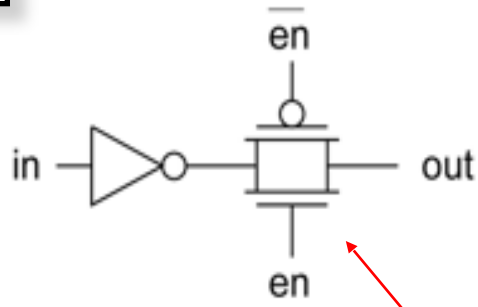
Tri-state Buffers

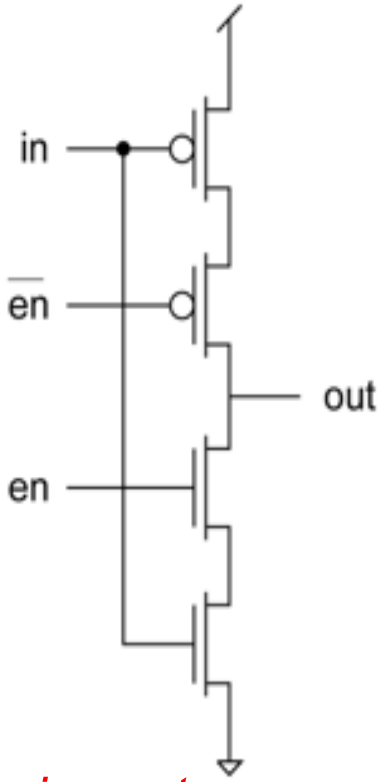
Tri-state Buffer:



OE	IN	OUT
0	0	Z
0	1	Z
1	0	0
1	1	1

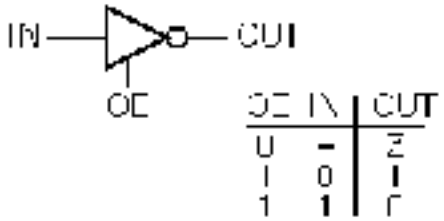
"high impedance" (output disconnected)





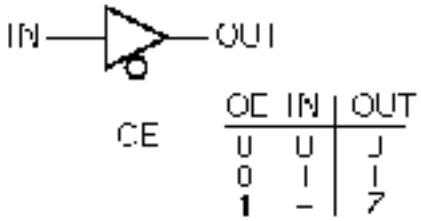
transmission gate useful in implementation

Variations:



OE	IN	OUT
0	–	Z
1	0	1
1	1	0

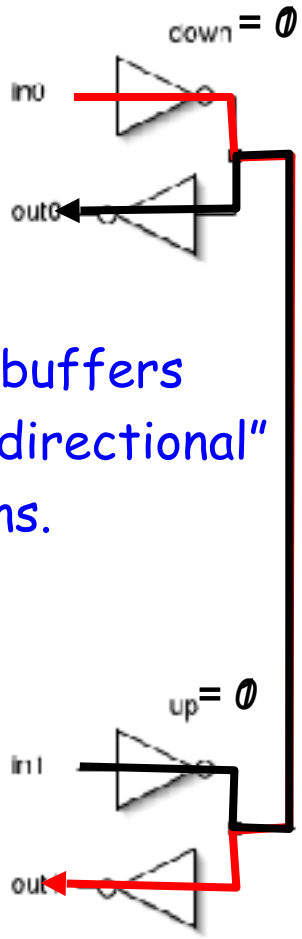
Inverting buffer



CE	IN	OUT
0	0	1
0	1	1
1	–	Z

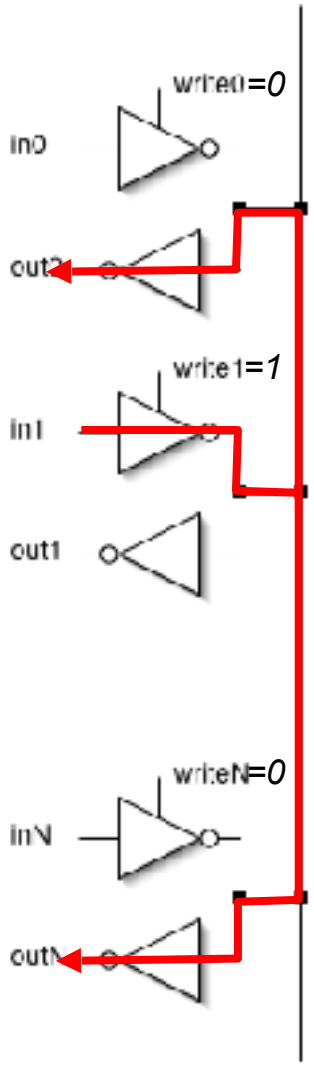
Inverted enable

Tri-state Buffers



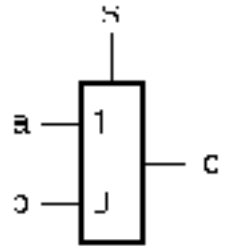
Tri-state buffers enable "bidirectional" connections.

Tri-state buffers are used when multiple circuits all connect to a common wire. Only one circuit at a time is allowed to drive the bus. All others "disconnect" their outputs, but can "listen".

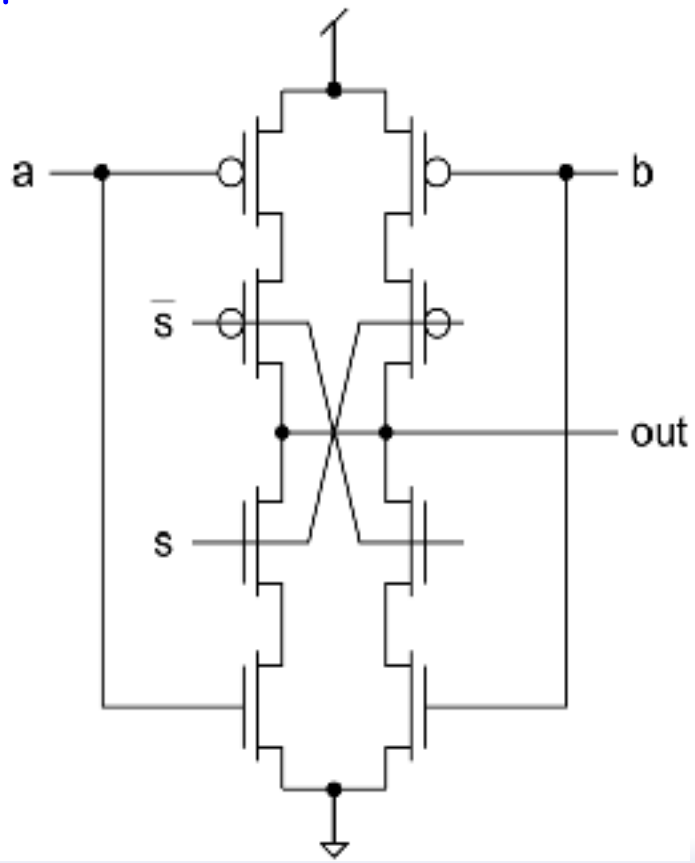


Tri-state Based Multiplexor

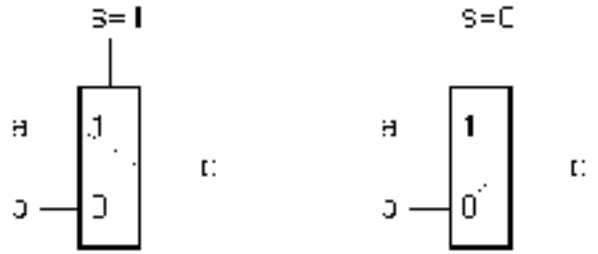
Multiplexor



Transistor Circuit for inverting multiplexor:

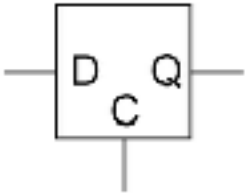


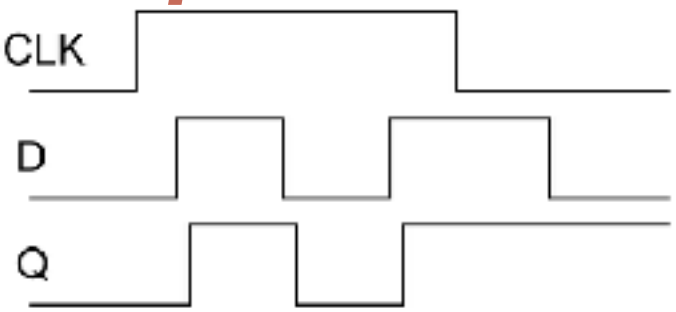
If s=1 then c=a else c=b



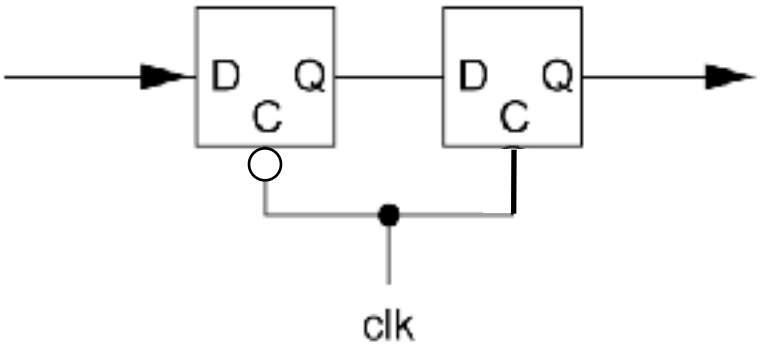
Latches and Flip-flops

Positive Level-sensitive ***latch***:

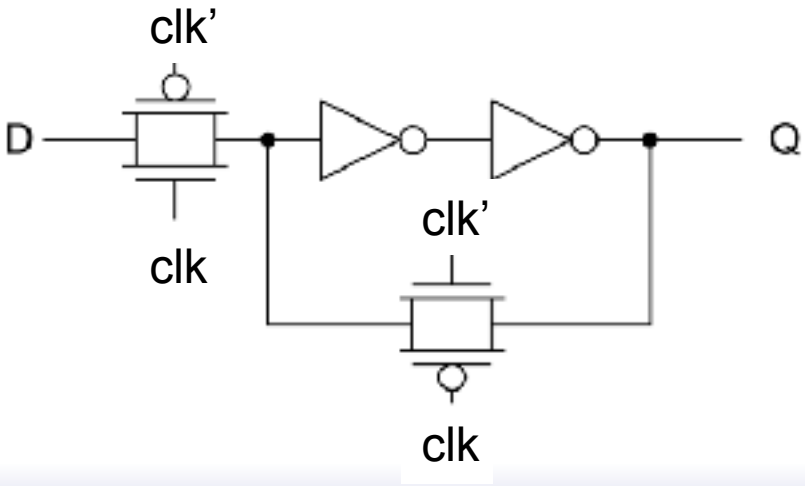




Positive Edge-triggered **flip-flop** built from two level-sensitive latches:



Latch Implementation:



Summary: Complimentary CMOS Properties

- Full rail-to-rail swing
- Symmetrical VTC
- No (…) static power dissipation
- Direct path current during switching

Digital abstraction

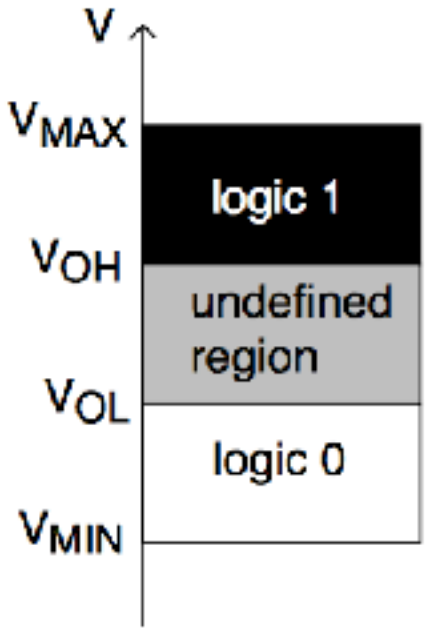
Noise and Digital Systems

- Circuit needs to works despite “analog” noise
  - Digital gates can and must reject noise
  - This is actually how digital systems are defined

- Digital system is one where:
  - Discrete values mapped to analog levels and back
  - Elements (gates) can reject noise
    - For “small” amounts of noise, output noise is less than input noise
  - Thus, for sufficiently “small” noise, the system acts as if it was noiseless
  - This is called regeneration

Bridging the digital and the analog worlds

- How to represent 0's and 1's in a world that is analog?

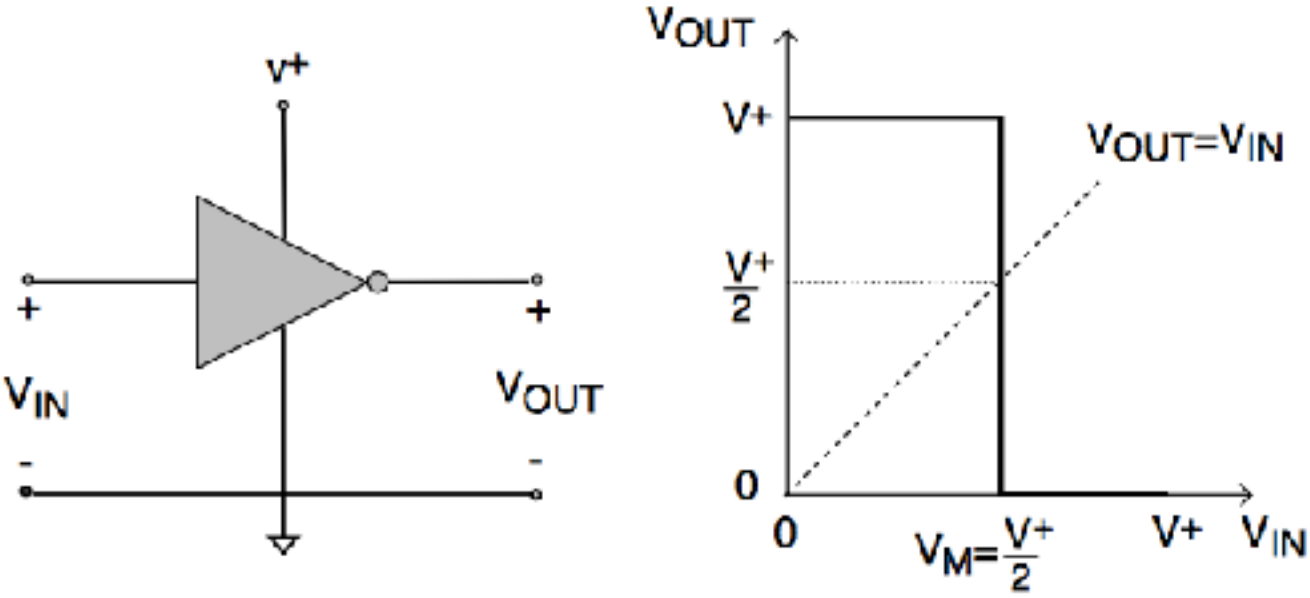


The Static Definition

- ***Logic 0:*** $V_{MIN} \le V \le V_{OL}$
- ***Logic 1:*** $V_{OH} \le V \le V_{MAX}$
- ***Undefined logic value:*** $V_{OL} \le V \le V_{OH}$

Ideal Inverter

Circuit representation and ideal transfer function:



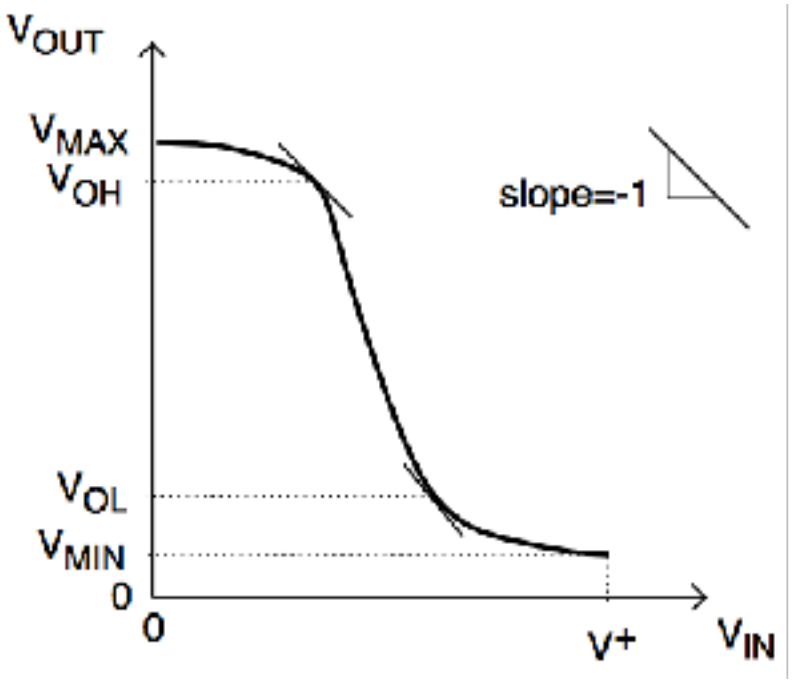
Define ***switching point*** or ***logic threshold*** :

- $V_M \equiv$ input voltage for which $V_{OUT} = V_{IN}$
  - For $0 \le V_{IN} < V_M \quad \Rightarrow V_{OUT} = V^+$
  - For $V_M < V_{IN} \le V^+ \quad \Rightarrow V_{OUT} = 0$

Ideal inverter returns well defined logical outputs (0 or V^+) even in the presence of considerable noise in V_{IN} (from voltage spikes, crosstalk, etc.)

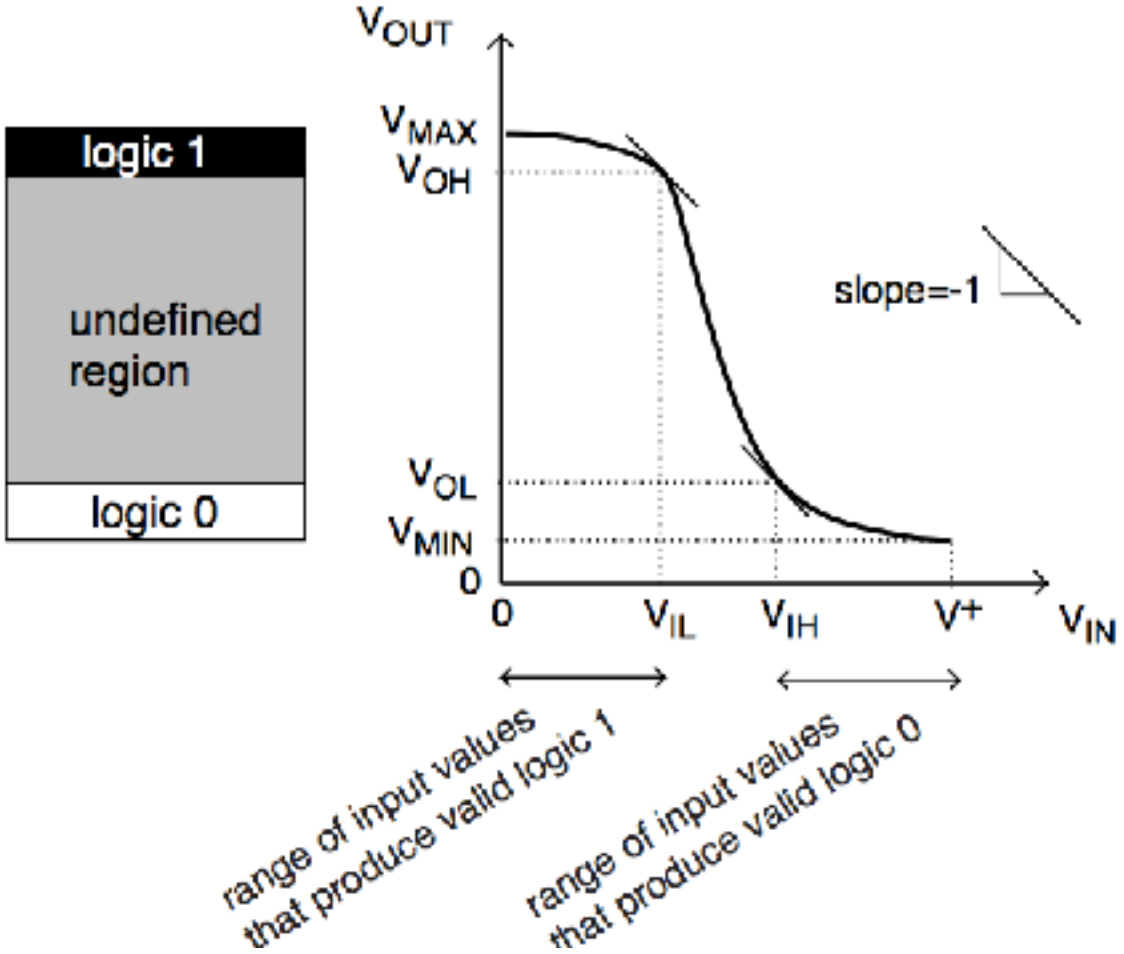
$\Rightarrow$ signal is ***regenerated***!

"Real Inverter"



- Logic 0:
  - $V_{MIN} \equiv$ output voltage for which $V_{IN} = V^+$
  - $V_{OL} \equiv$ smallest output voltage where slope = -1
- Logic 1:
  - $V_{OH} \equiv$ largest output voltage where slope = -1
  - $V_{MAX} \equiv$ output voltage for which $V_{IN} = 0$

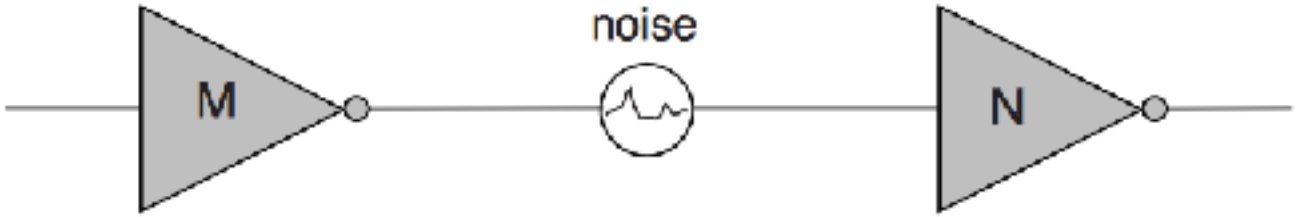
Valid Input Ranges

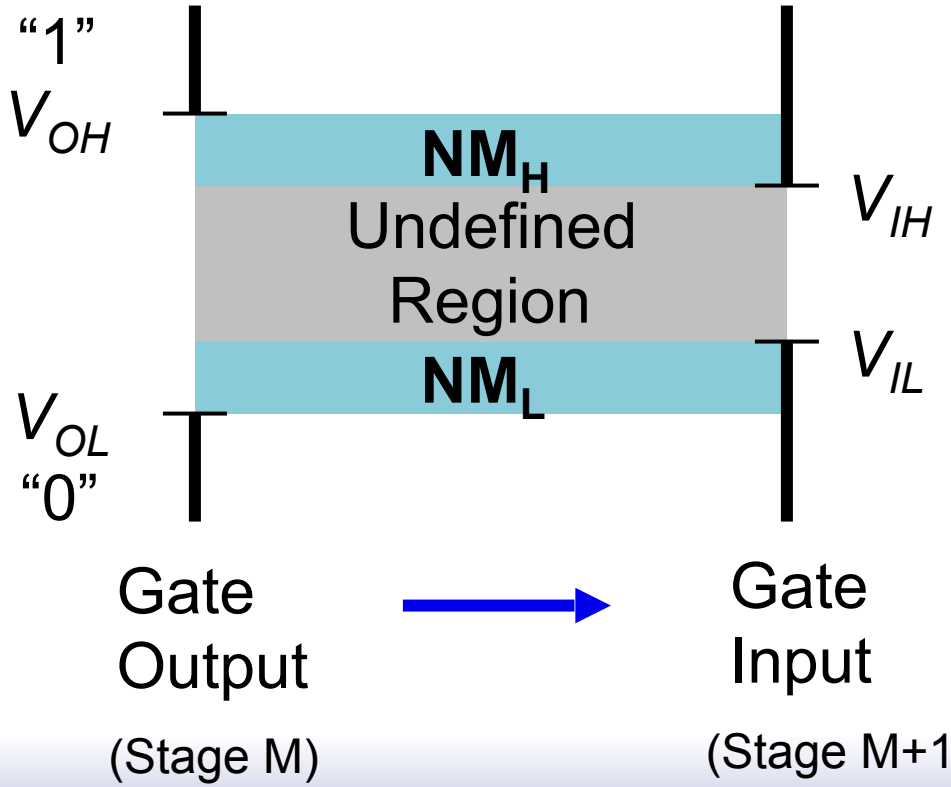


If range of output values V_{OL} to V_{OH} is ***wider*** than the range of input values V_{IL} to V_{IH}, then the inverter exhibits some noise immunity. (|Voltage gain| > 1)

Quantify this through ***noise margins.***

Definition of Noise Margins

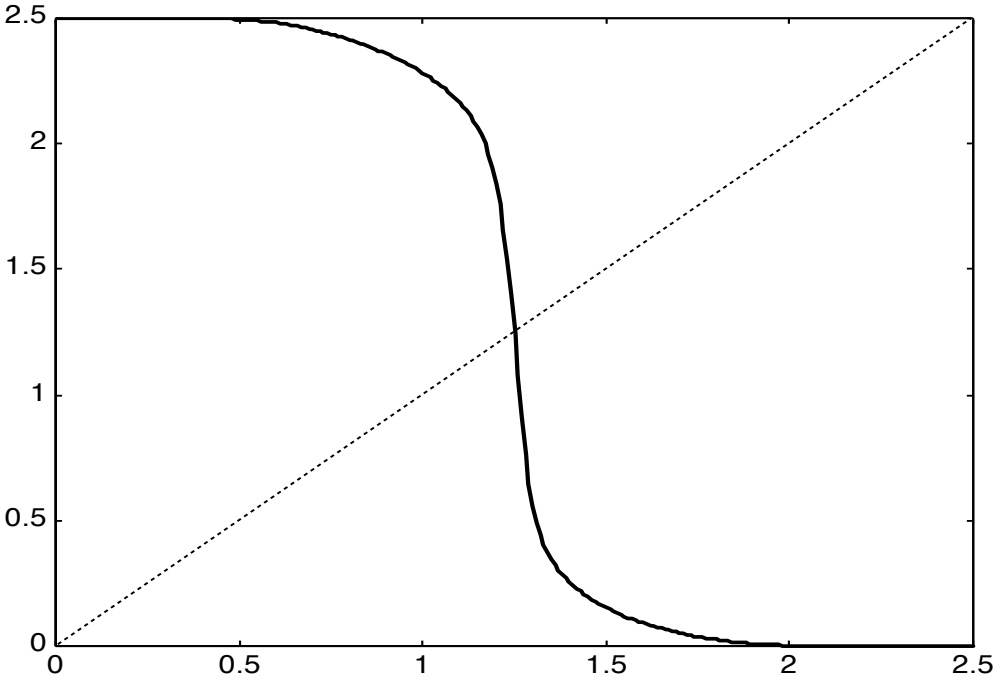




Noise margin high:
$NM_H = V_{OH} - V_{IH}$

Noise margin low:
$NM_L = V_{IL} - V_{OL}$

Simulated Inverter VTC (Spice)



- V_{OH} =
- V_{OL} =
- V_{IL} =
- V_{IH} =
- N_{MH} =
- N_{ML} =
- V_M =