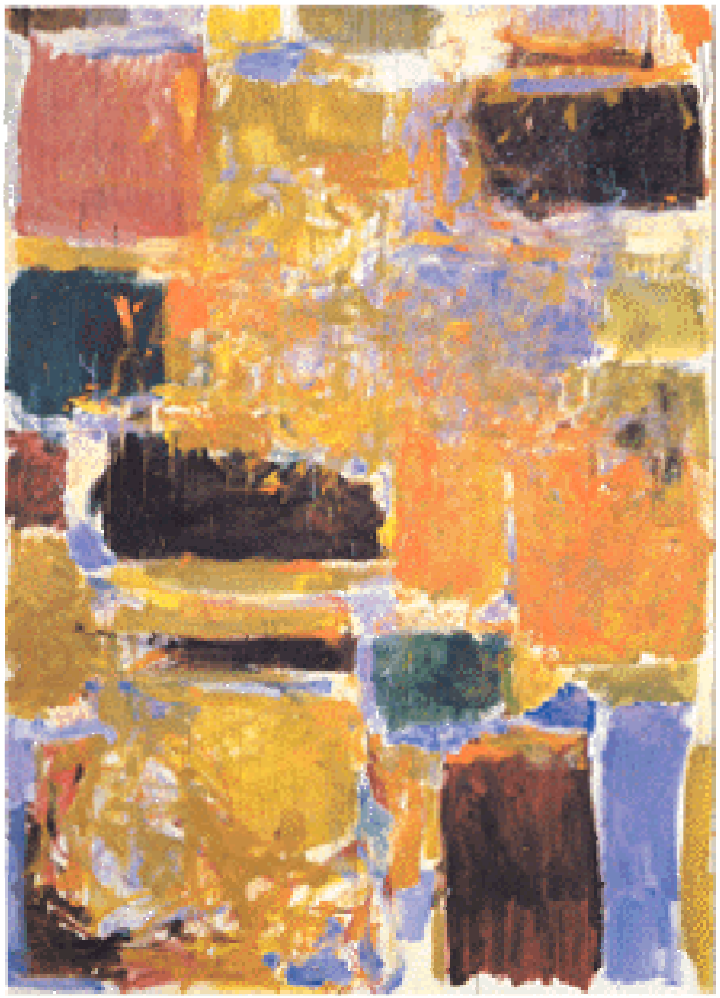
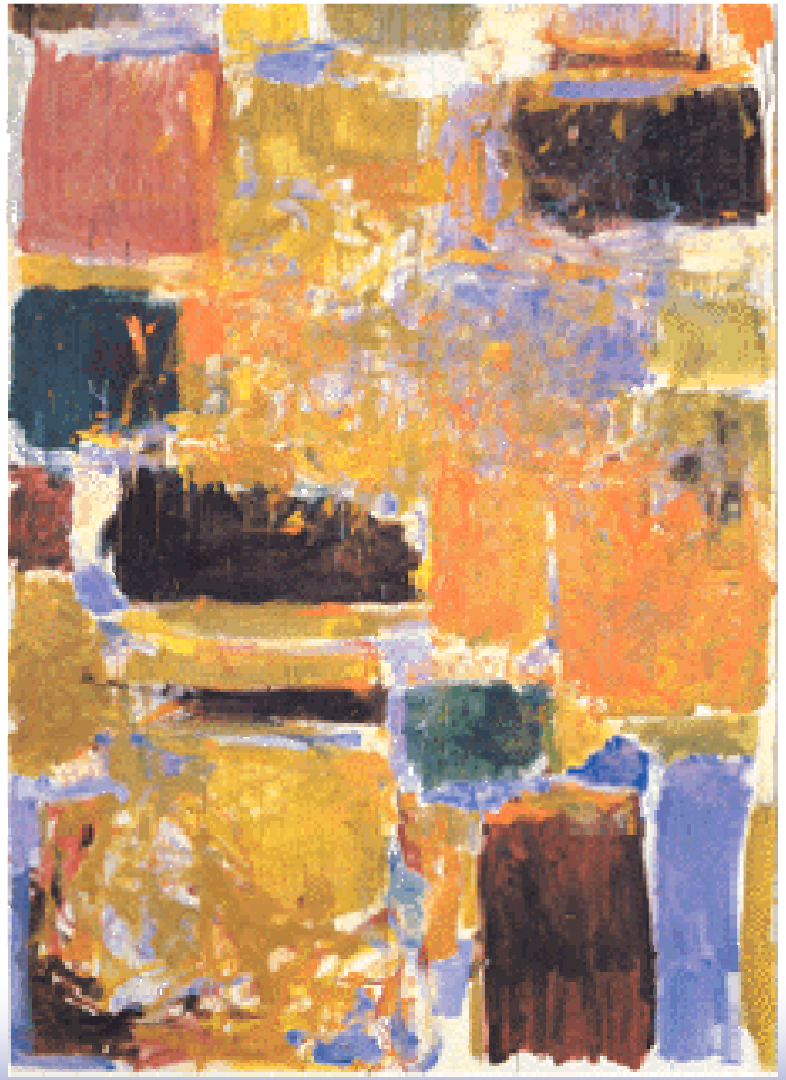


CS150 - EE141/241A
Fall 2014
Digital Design and Integrated Circuits

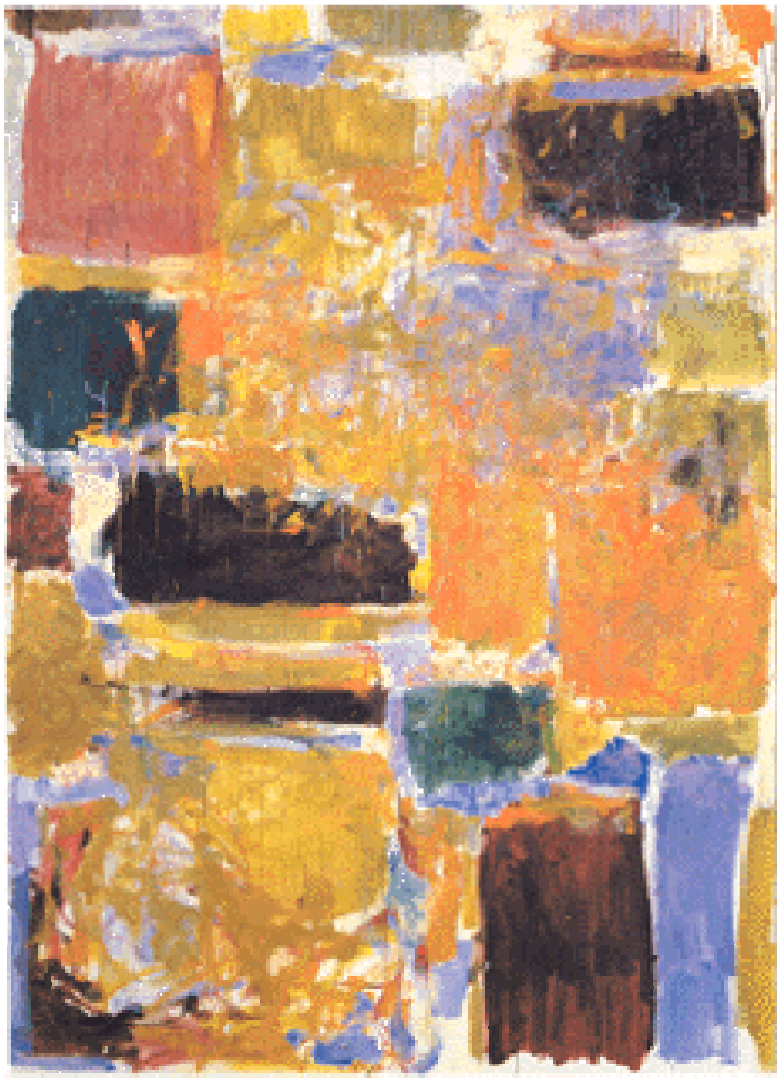
Instructors:
John Wawrzynek and Vladimir Stojanovic

Lecture 8

Outline



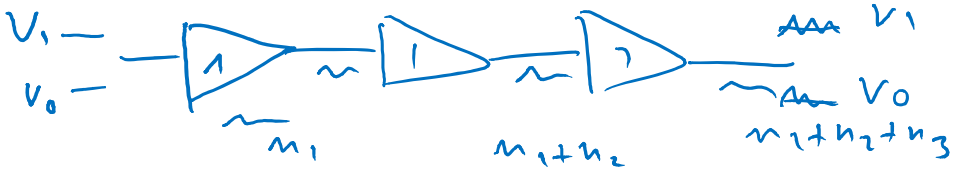
- Digital abstraction
- CMOS abstraction
- Switch logic



Digital abstraction

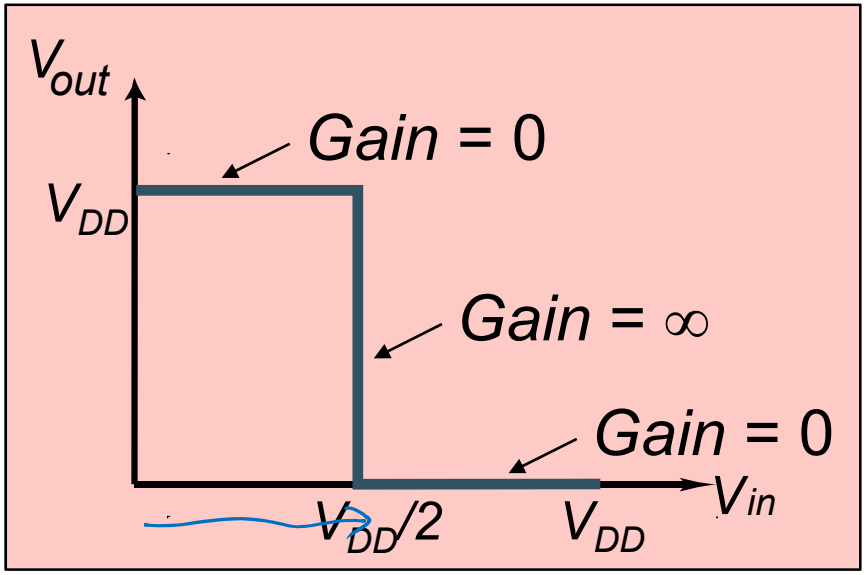
Noise and Digital Systems

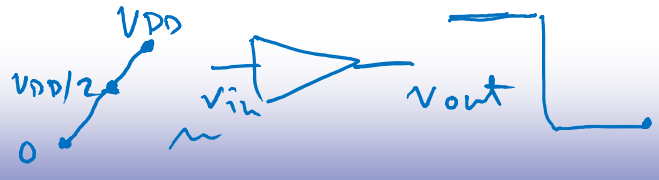
- Circuit needs to works despite "analog" noise
  - Digital gates can reject noise
  - This is actually how digital systems are defined
- Digital system is one where:
  - Discrete values mapped to analog levels and back
  - All the elements (gates) can reject noise
    - For "small" amounts of noise, output noise is less than input noise
  - Thus, for sufficiently "small" noise, the system acts as if it was noiseless



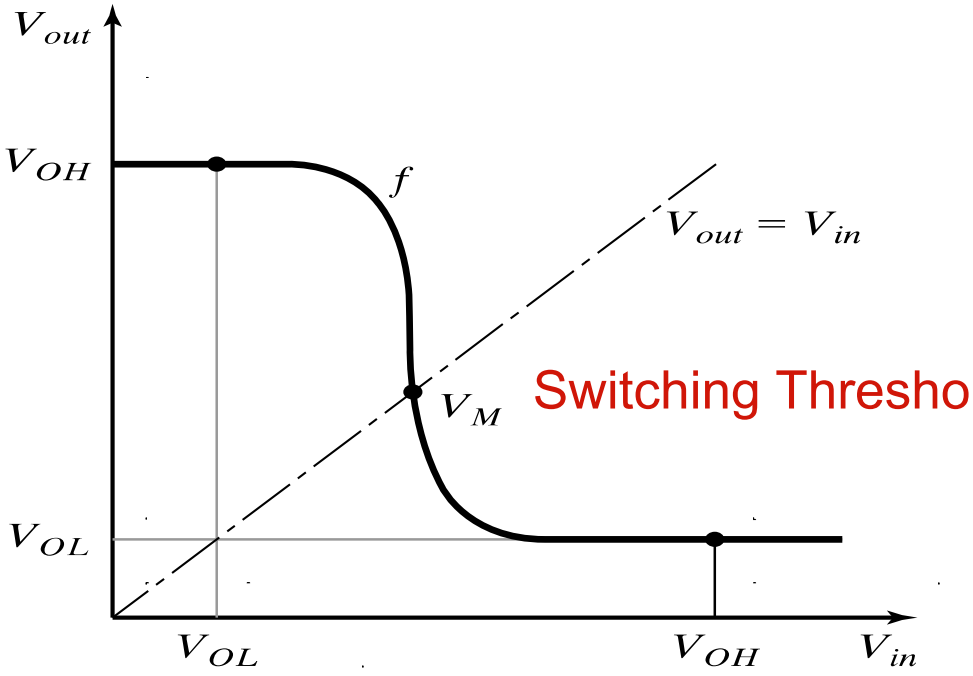
Noise Rejection

- To see if a gate rejects noise
  - Look at its DC voltage transfer characteristic (VTC)
  - See what happens when input is not exactly 1 or 0
- Ideal digital gate:
  - Noise needs to be larger than $V_{DD}/2$ to have any effect on gate output





The Voltage Transfer Characteristic

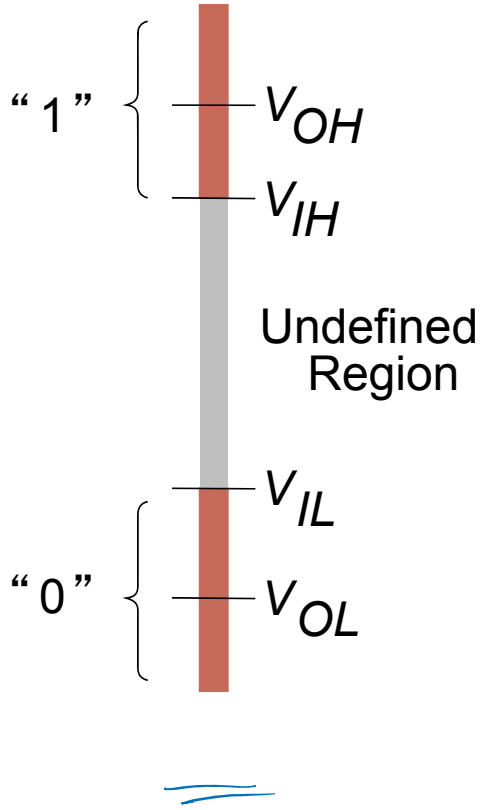


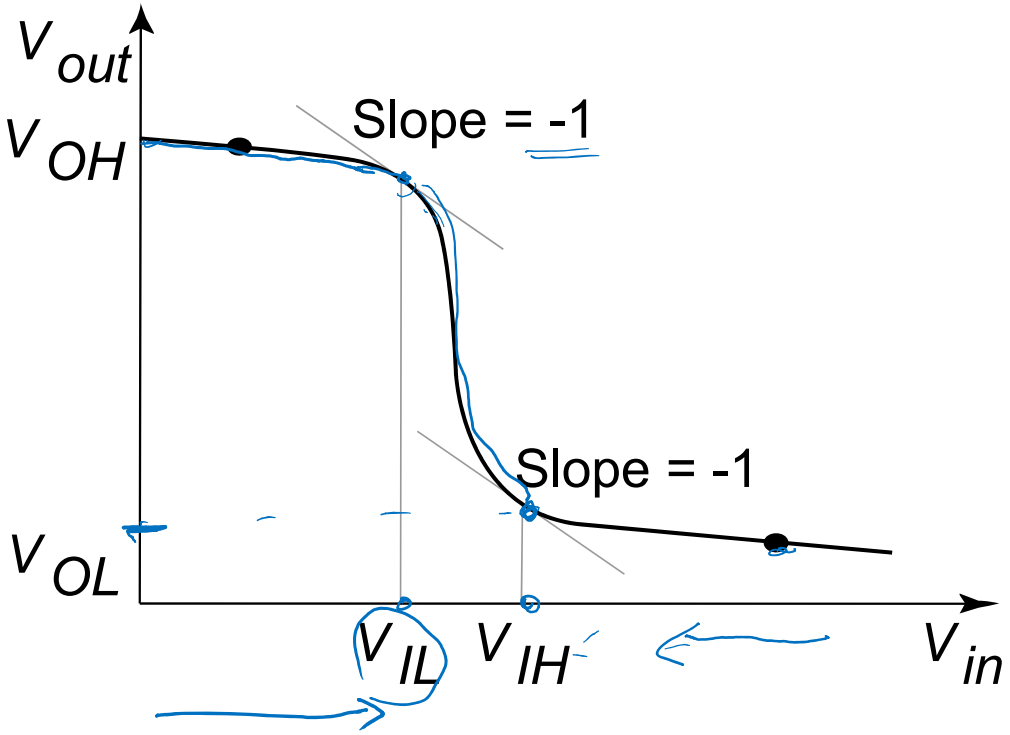
VOH = *f*(VOL)
VOL = *f*(VOH)
VM = *f*(VM)

Switching Threshold

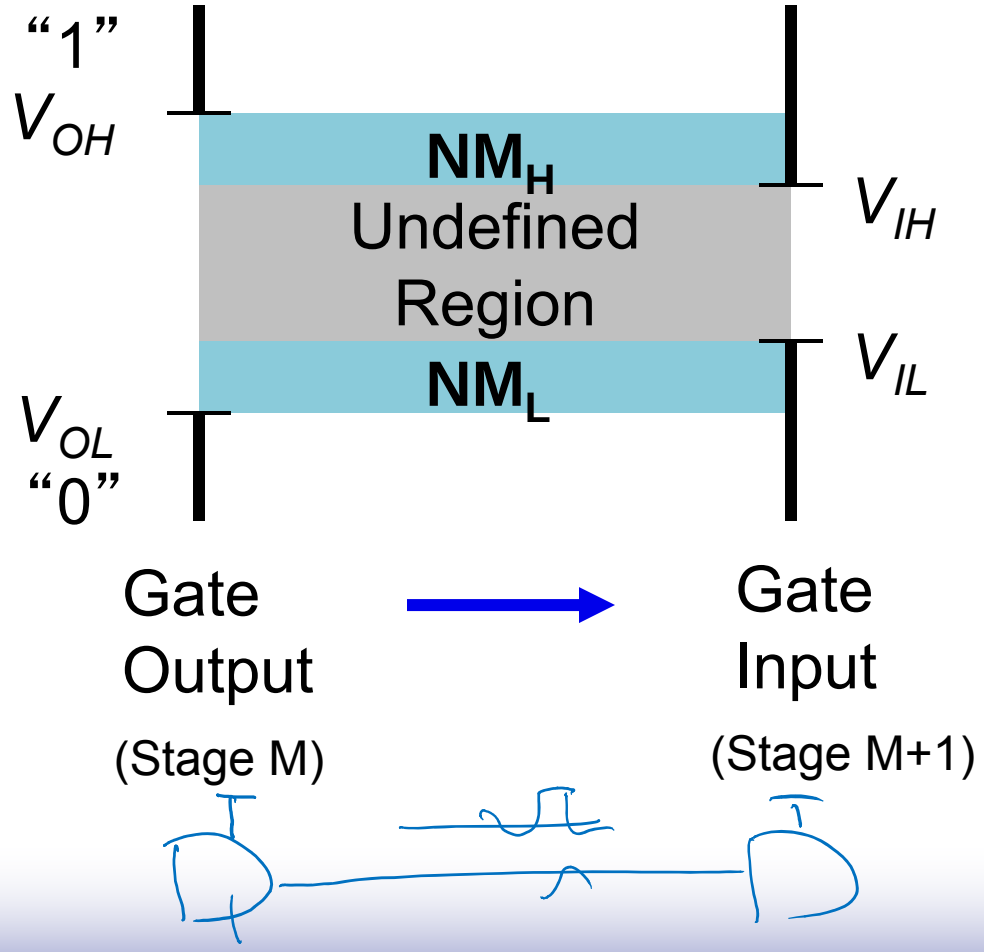
Nominal Voltage Levels

Voltage Mapping



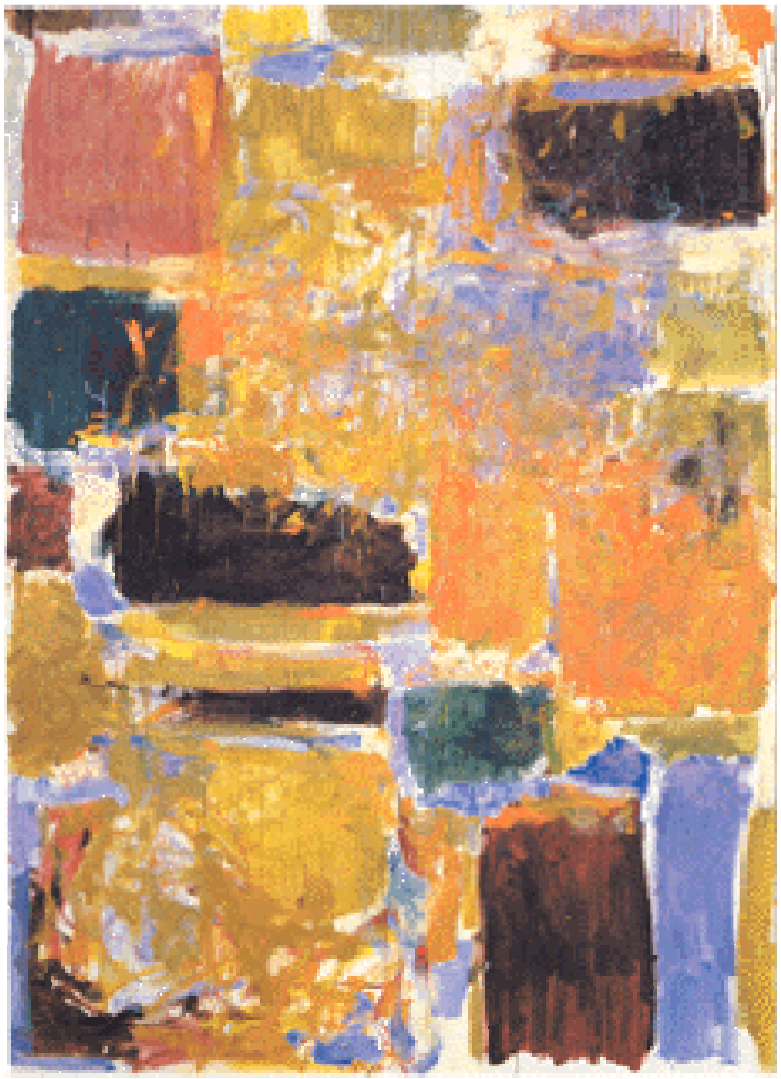


Definition of Noise Margins



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

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

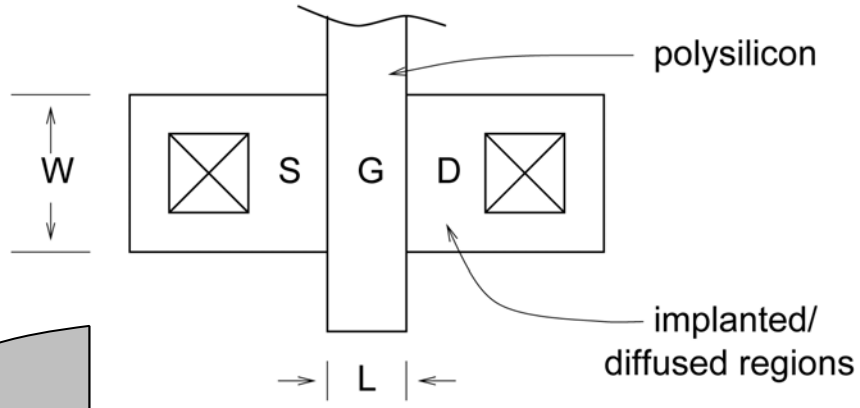


CMOS abstraction

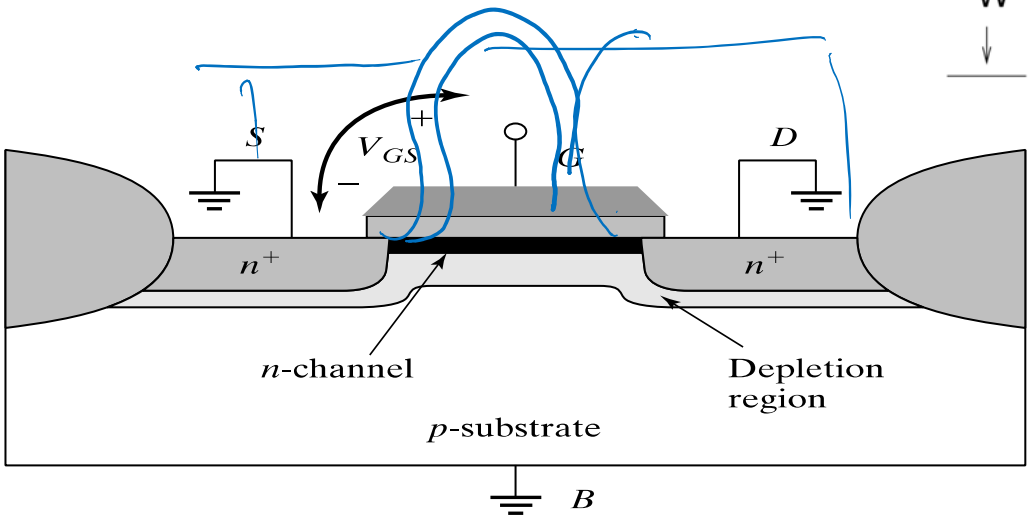
CMOS Devices

- MOSFET (Metal Oxide Semiconductor Field Effect Transistor).

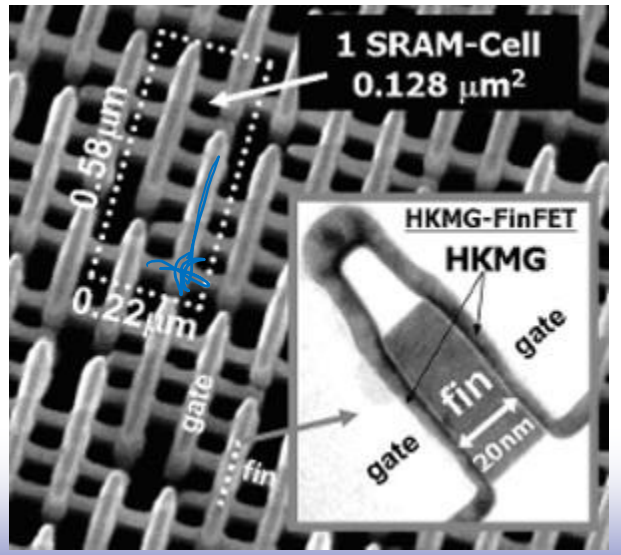
Top View



Cross-section View



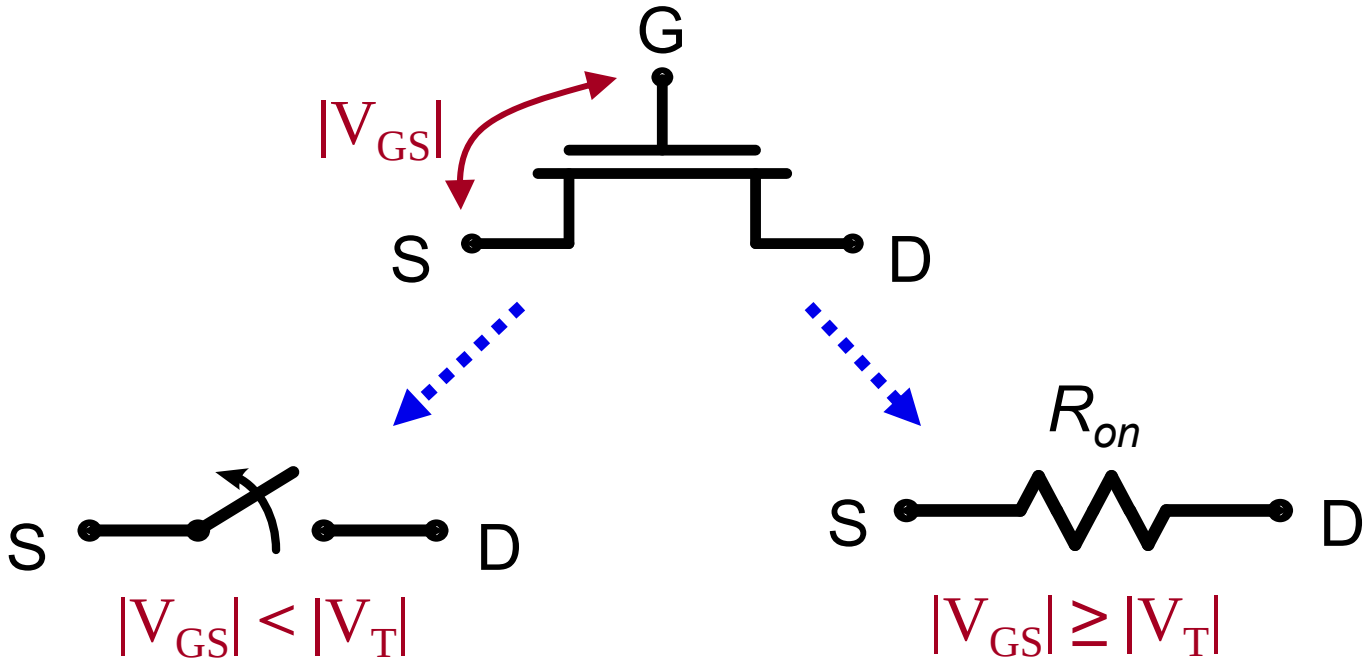
The gate acts like a capacitor. A high voltage on the gate attracts charge into the channel. If a voltage exists between the source and drain a current will flow. In its simplest approximation, the device acts like a switch.



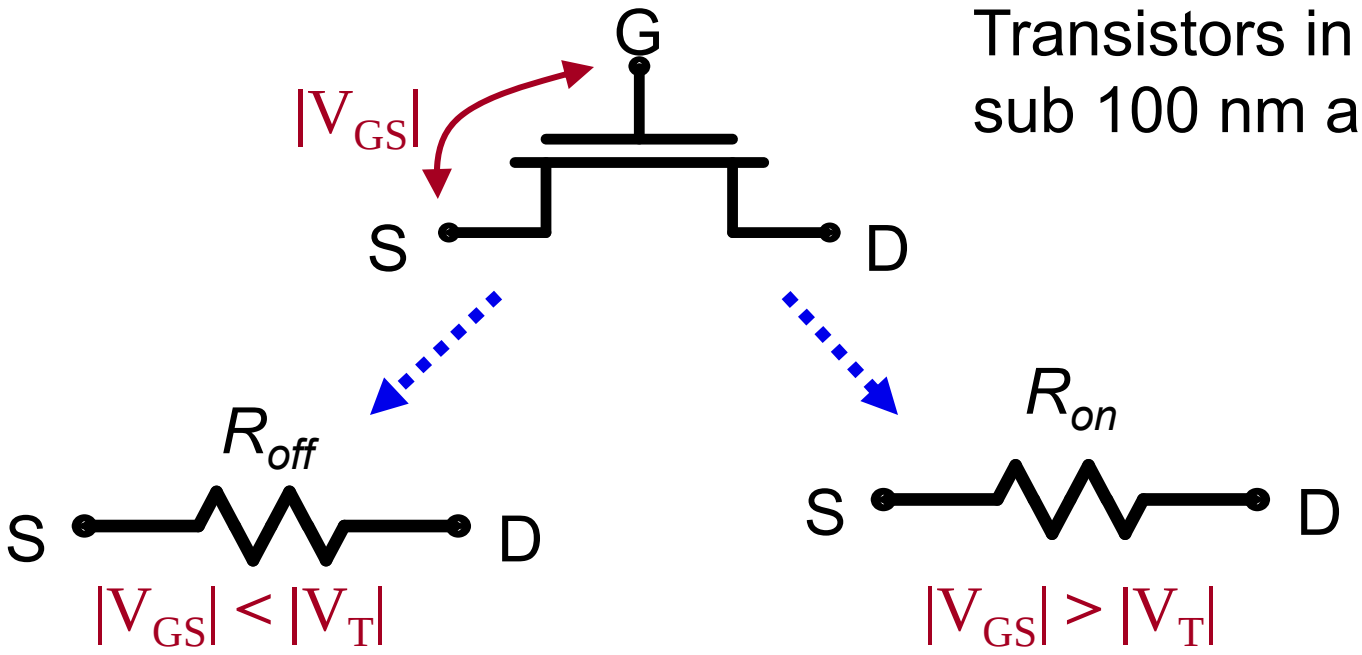
MOS Transistor as a Switch

An MOS Transistor ⟷ A Switch!

ON/OFF Switch Model of MOS Transistor

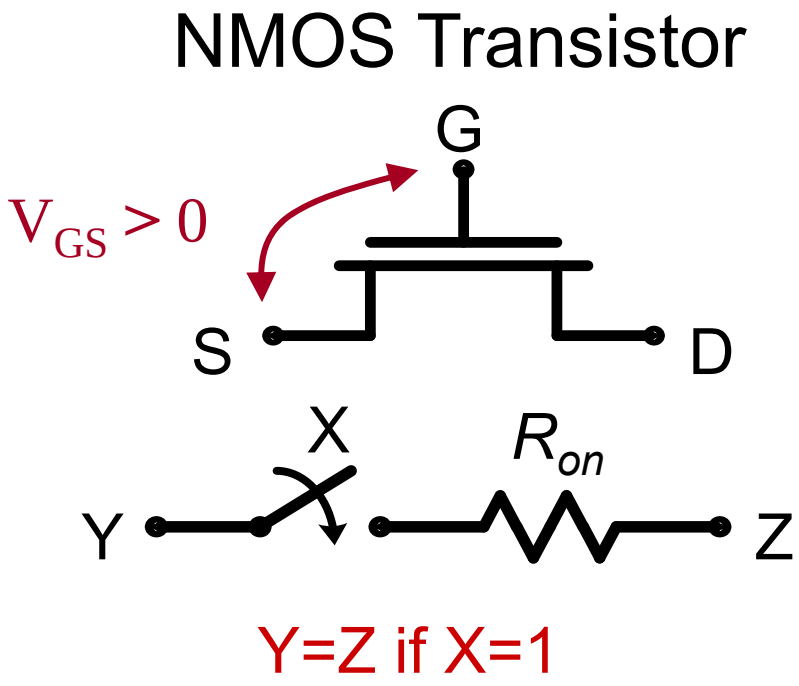


A More Realistic Switch

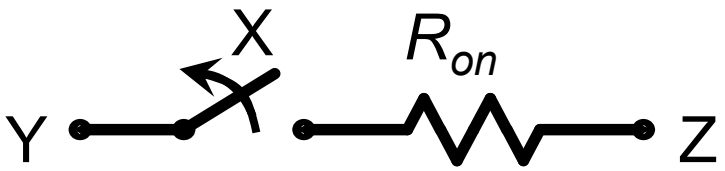


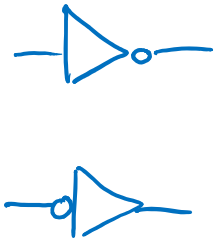
Transistors in the sub 100 nm age

A Logic Perspective



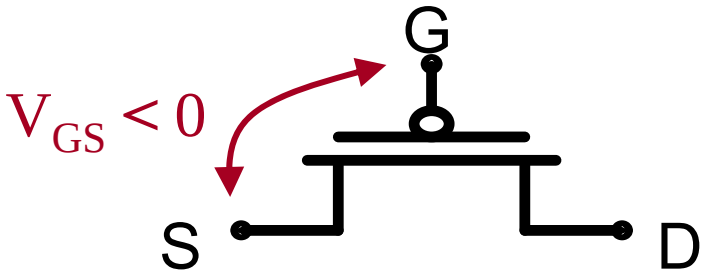
A Complementary Switch



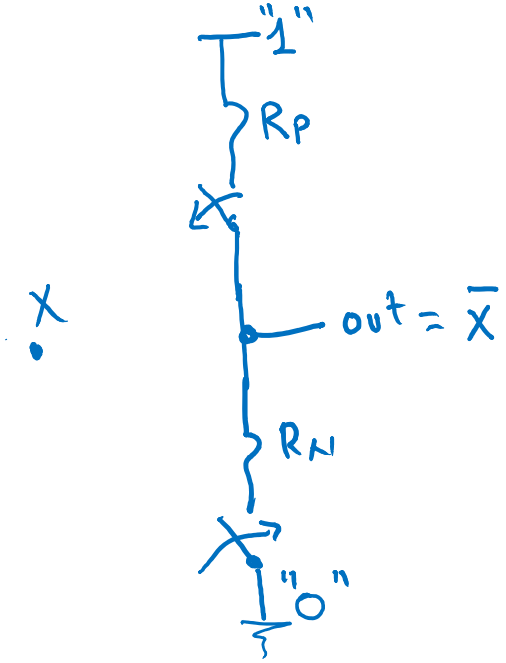


Y=Z if X=0

PMOS Transistor



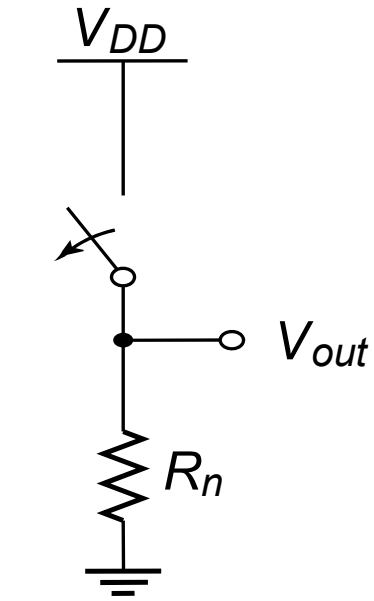
The Switch Inverter



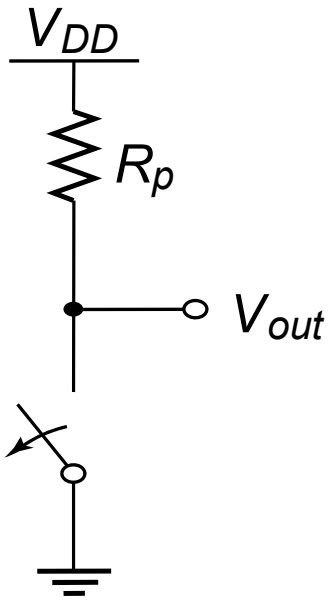
X = 1 out = 0

X = 0 out = 1

The Switch Inverter
First-Order DC Analysis



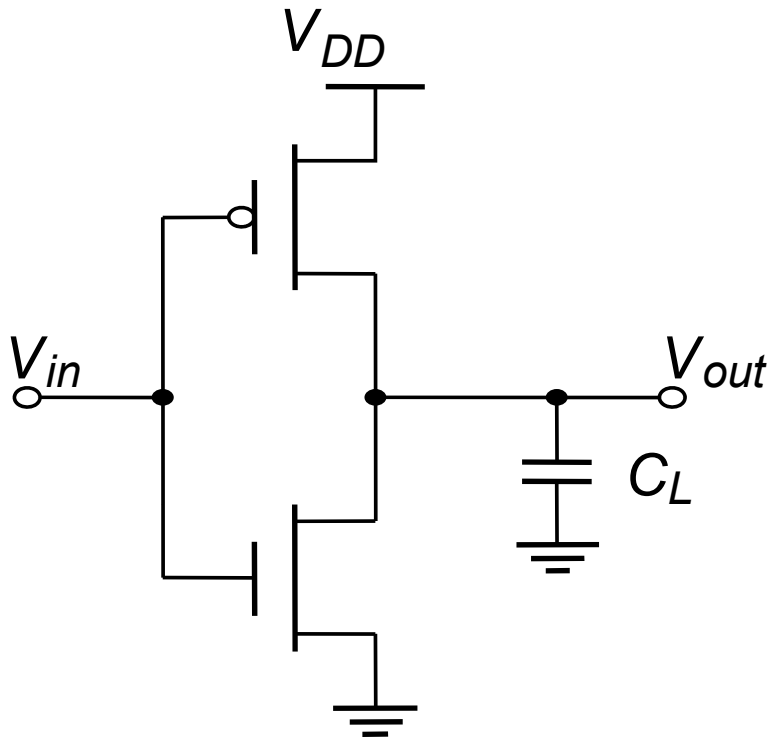
$V_{in} = V_{DD}$



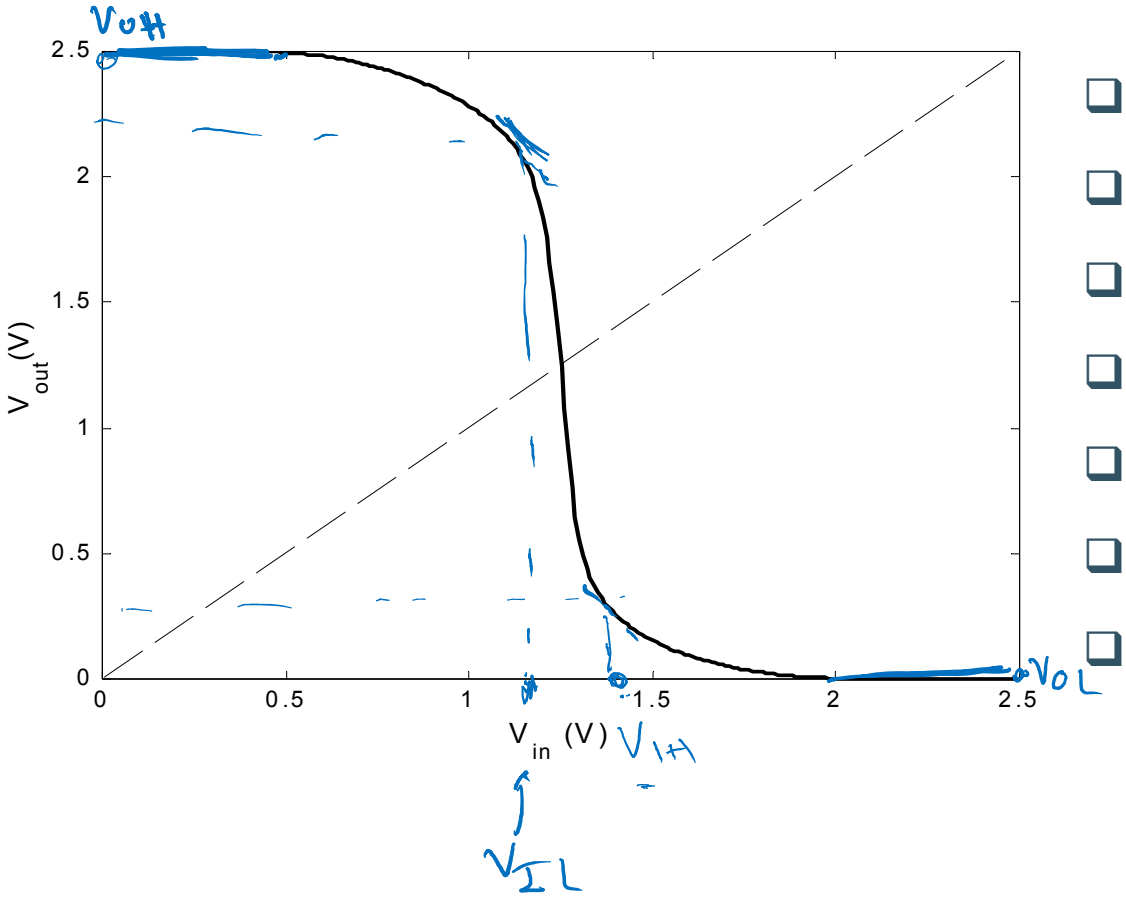
$V_{in} = 0$

$V_{OL} = 0$
$V_{OH} = V_{DD}$
$V_M = f(R_n, R_p)$

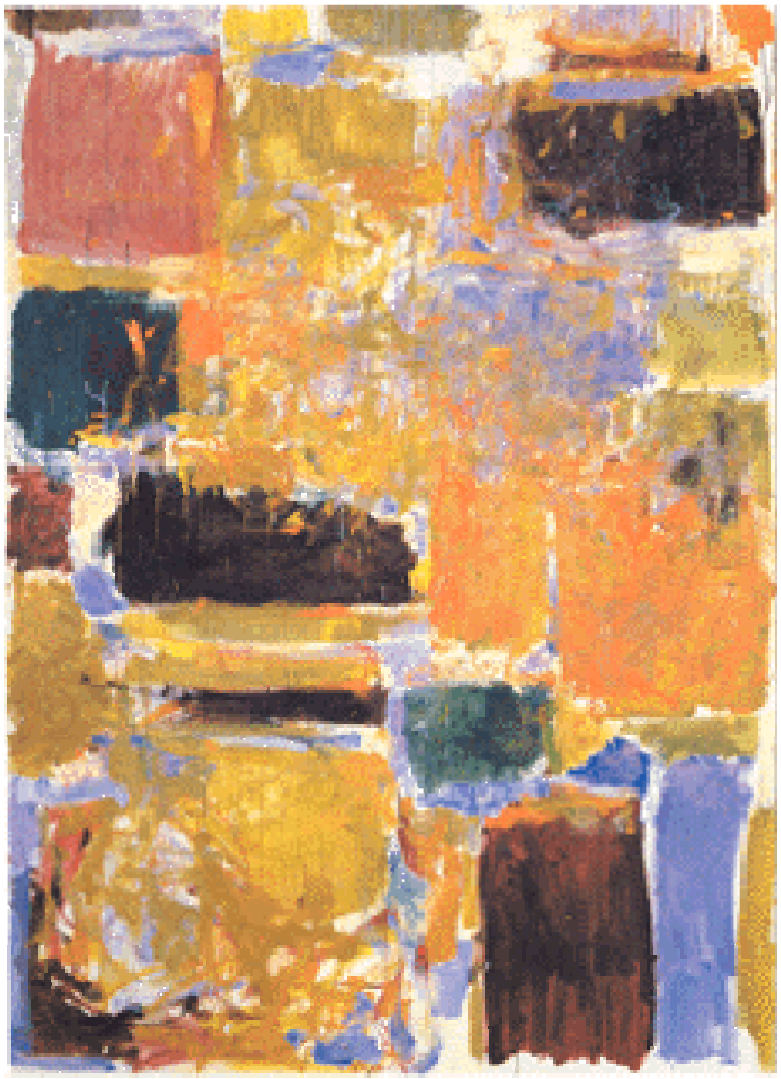
The CMOS Inverter: A First Glance



Simulated Inverter VTC (Spice)



- V_{OH} = 2.5 V
- V_{OL} = 0 V
- V_{IL} = 1.1 V
- V_{IH} = 1.4 V
- N_{MH} = $V_{OH} - V_{IH}$ = 1.1 V
- N_{ML} = $V_{IL} - V_{OL}$ = 1.1 V
- V_M = 1.25 V



Switch logic

Static Logic Gate

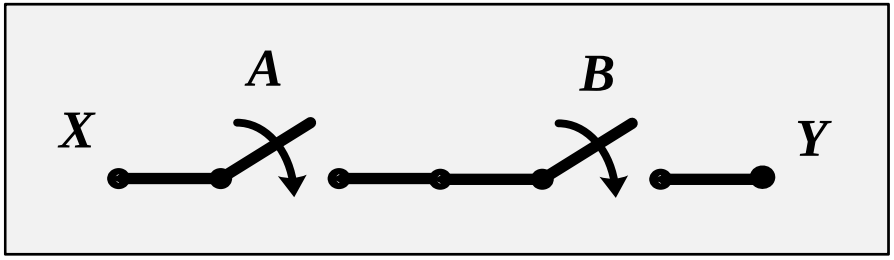
- At every point in time (except during the switching transients) each gate output is connected to either V_{DD} or V_{SS} via a low resistive path.

- The outputs 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

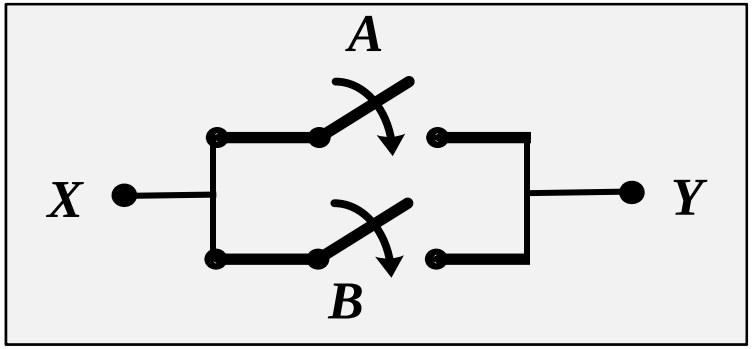
Series



AND

$Y = X$ if A AND B

Parallel



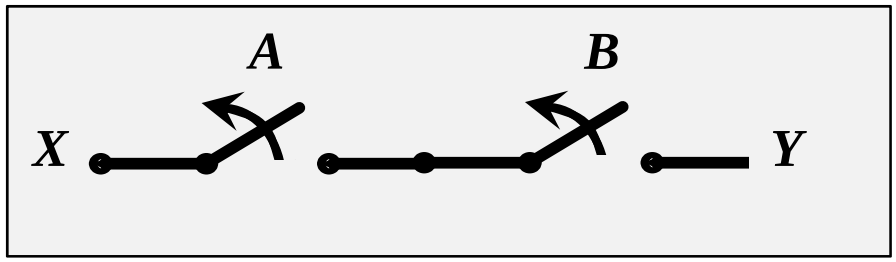
OR

$Y = X$ if A OR B

(output undefined if condition not true)

Logic using inverting switches

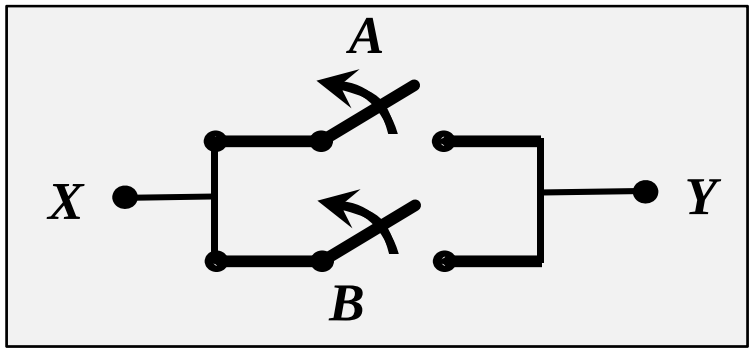
Series



NOR

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

Parallel

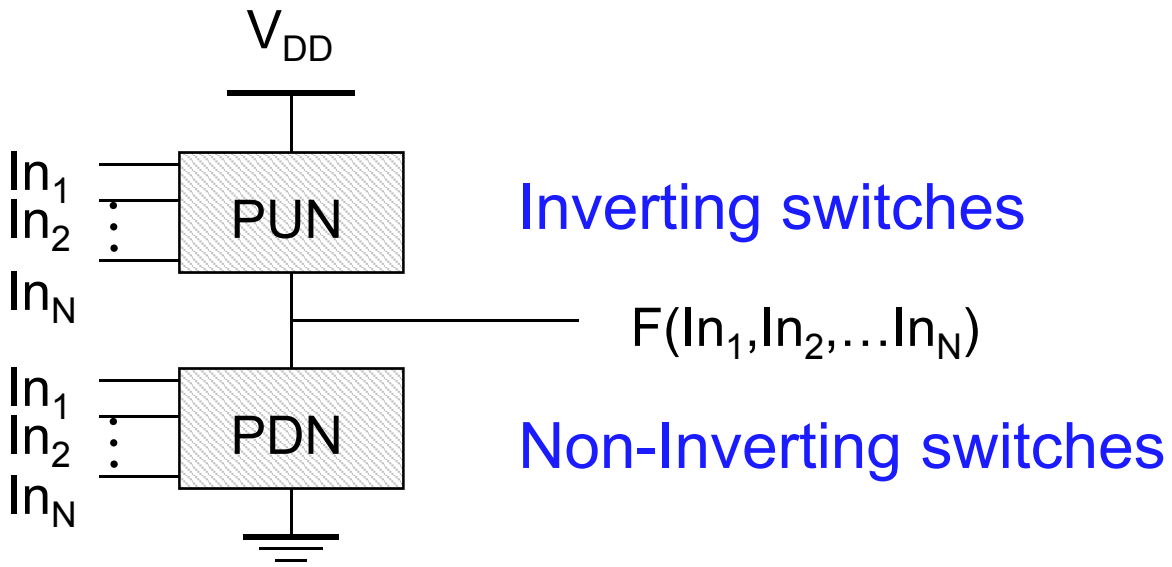


NAND

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

(output undefined if condition not true)

Static Complementary CMOS



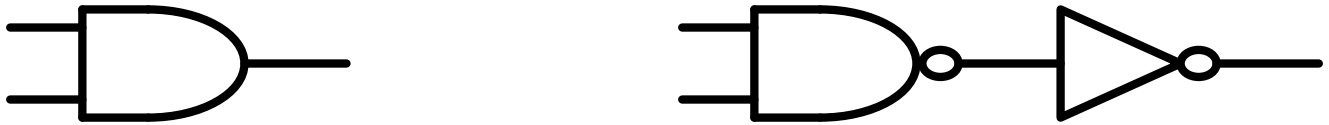
PUN and PDN are dual logic networks
PUN and PDN functions are complementary

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

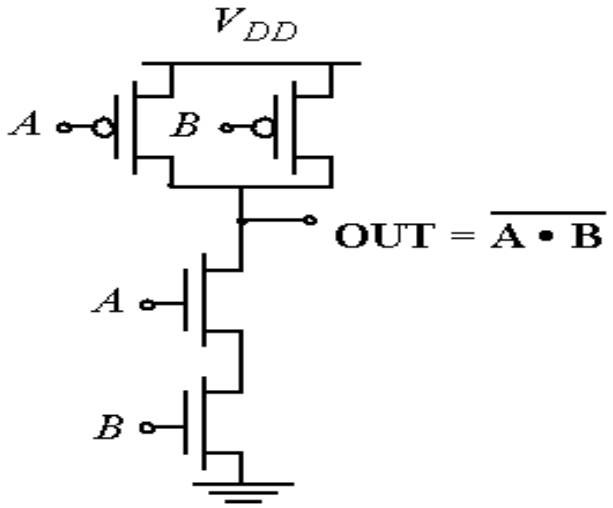


AND = NAND + INV

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

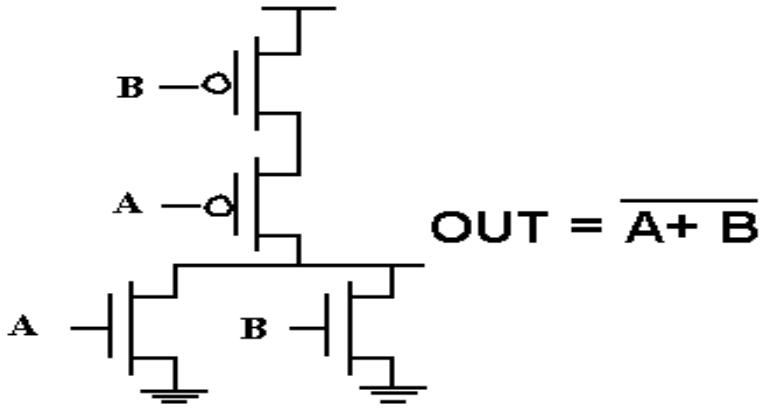


- PDN: G = AB $\Rightarrow$ Conduction to GND
- PUN: F = $\overline{A}$ + $\overline{B}$ = $\overline{AB}$ $\Rightarrow$ Conduction to V_{DD}
- $\overline{G(In_1,In_2,In_3,\ldots)} \equiv F(\overline{In_1},\overline{In_2},\overline{In_3},\ldots)$

Example Gate: NOR

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

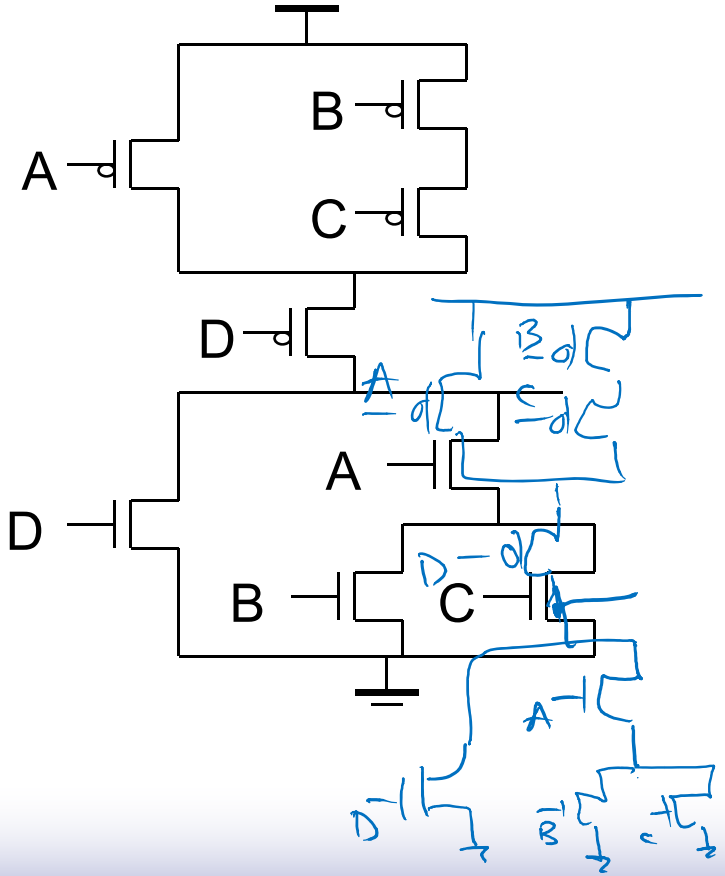
Truth Table of a 2 input NOR gate



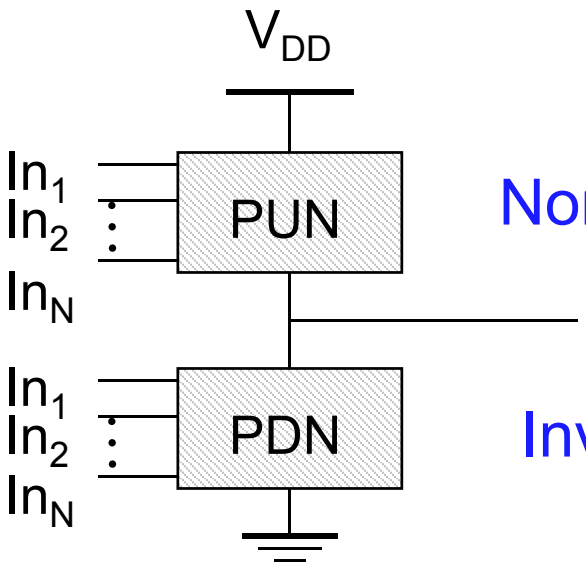
Complex CMOS Gate

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

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



Non-inverting logic



Non-Inverting switches

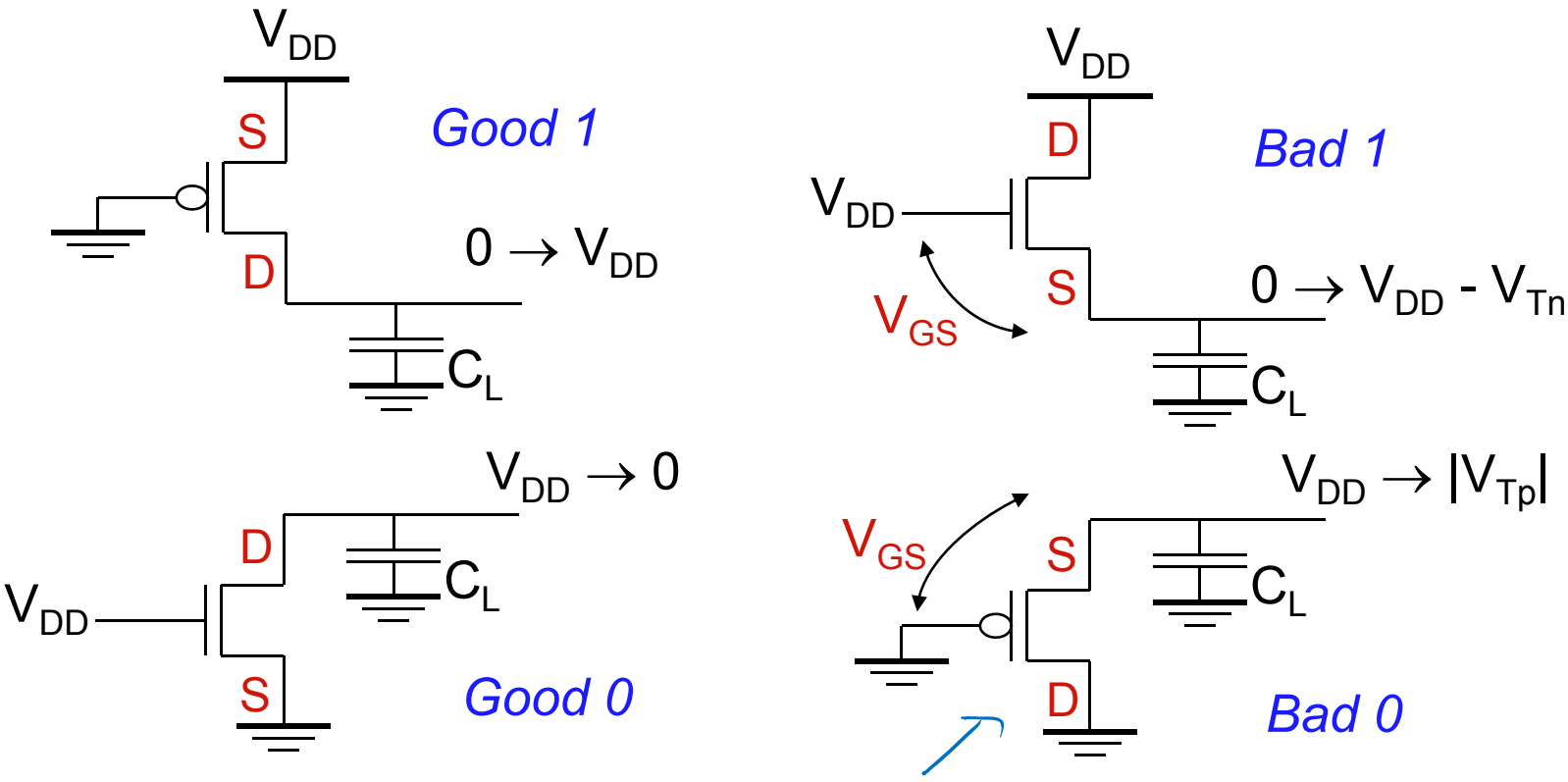
$F(In_1,In_2,\ldots In_N)$

Inverting switches

Why is this a bad idea?

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

Switch Limitations

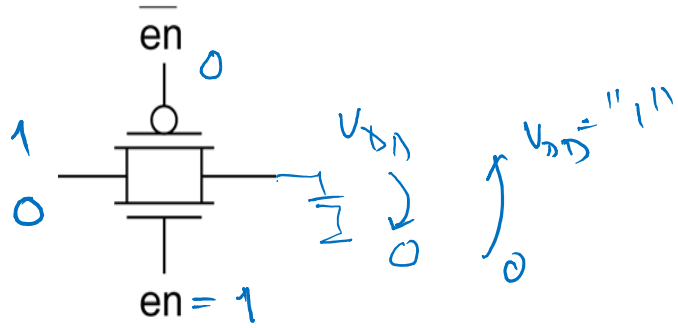


$V_{OL} = |V_{Tp}|$

$V_{OH} = V_{DD} - V_{Tn}$

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).

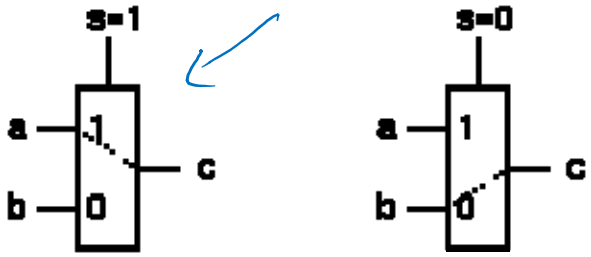


- Does not directly connect to Vdd and GND, but can be combined with logic gates or buffers to simplify many logic structures.

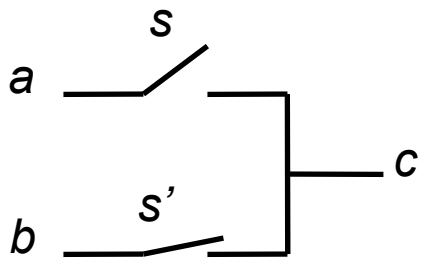
Transmission-gate Multiplexor

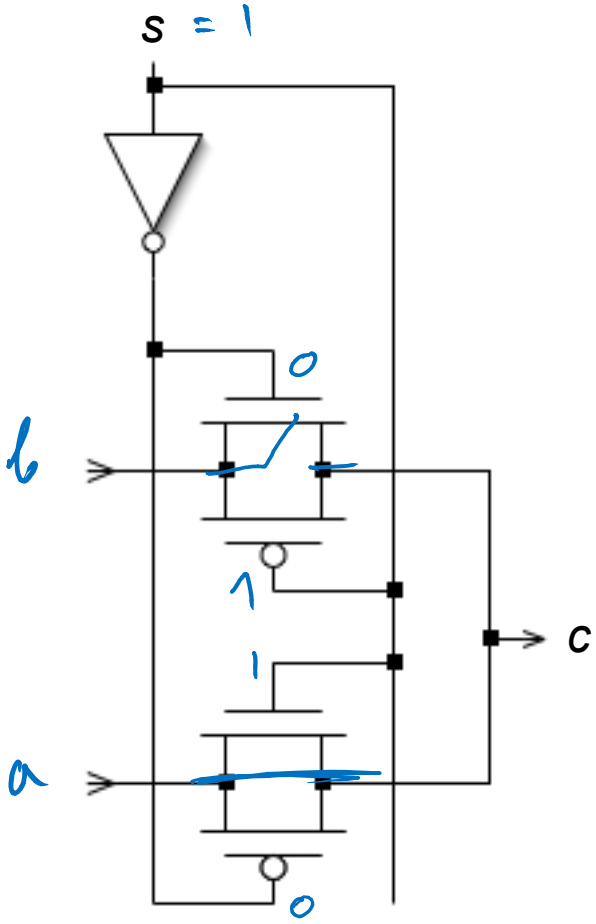
2-to-1 multiplexor:

$C = sa + s'b$



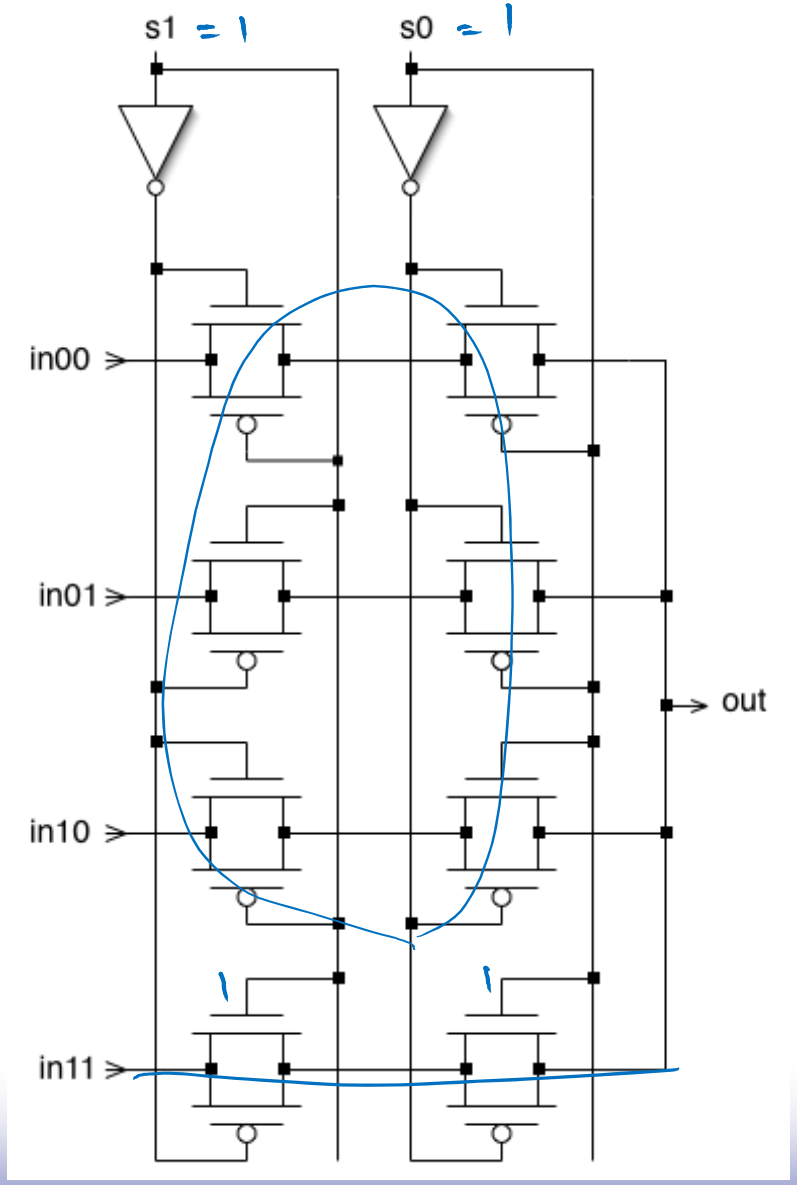
Switches simplify the implementation:





Compare the cost to logic gate implementation.

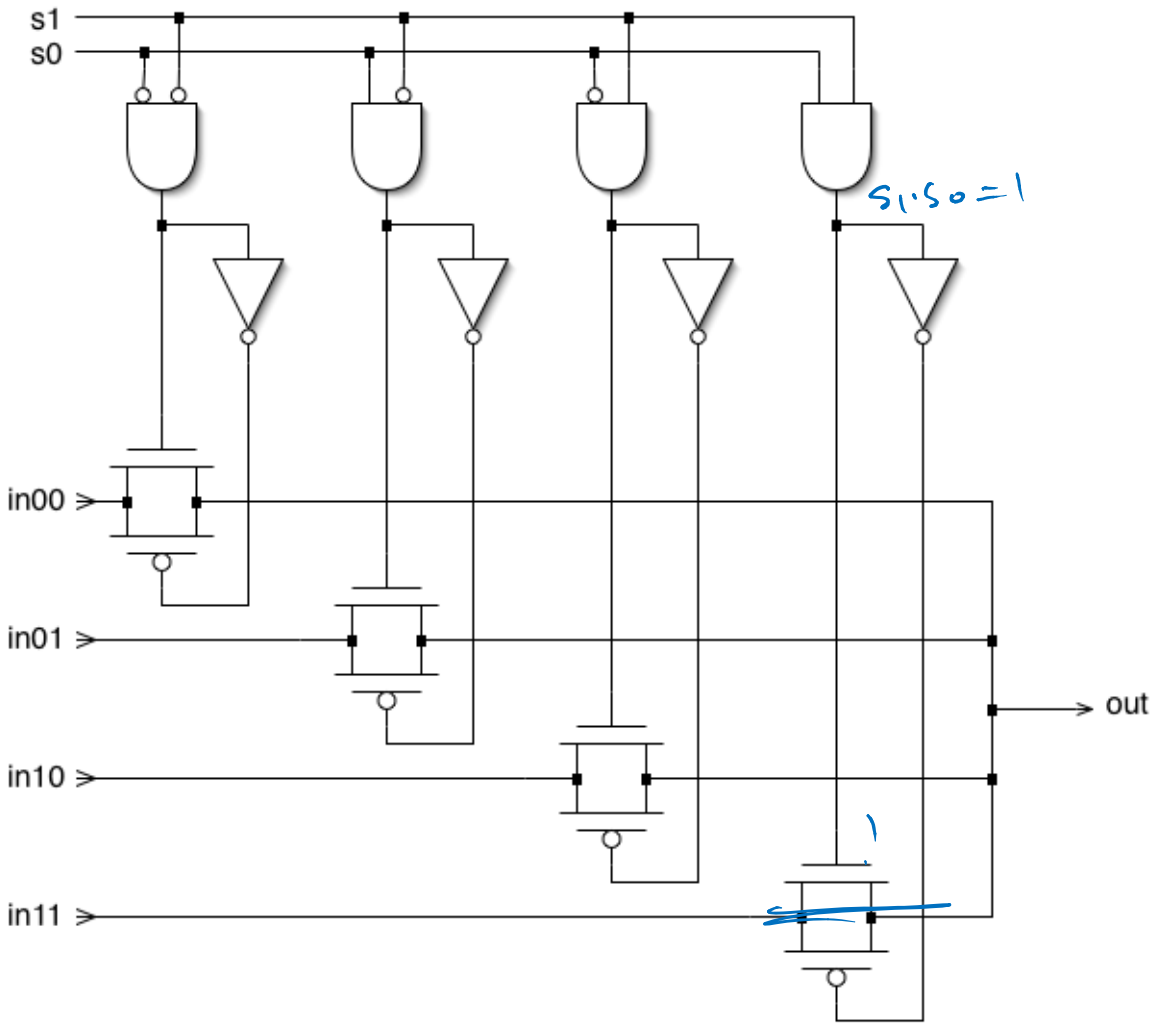
4-to-1 Transmission-gate Mux



- 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

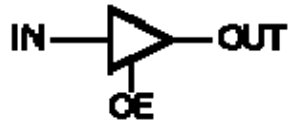
Alternative 4-to-1 Multiplexor

- 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

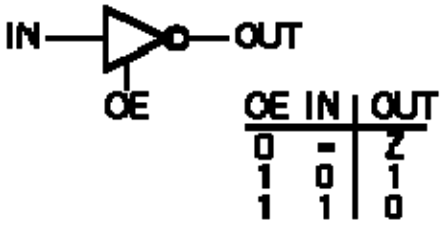
Tri-state Buffer:



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

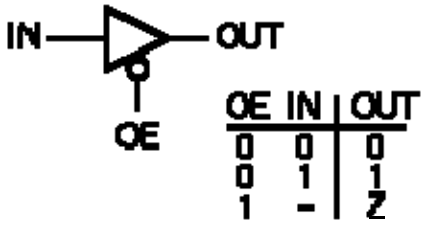
"high impedance"
(output disconnected)

Variations:



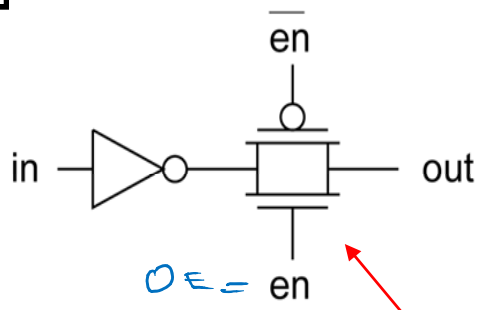
OE	IN	OUT
0	-	Z
1	0	1
1	1	0

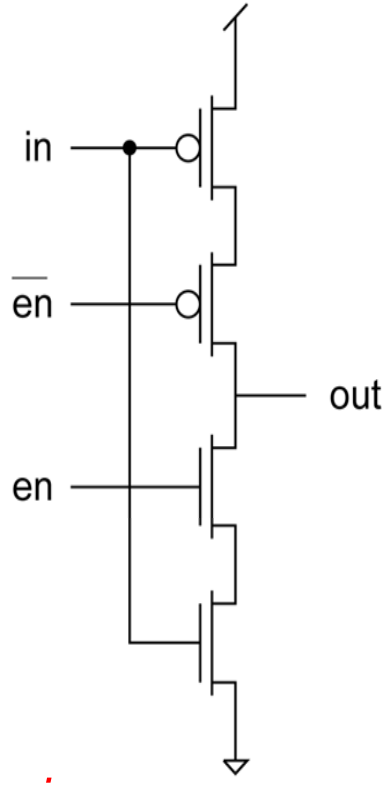
Inverting buffer



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

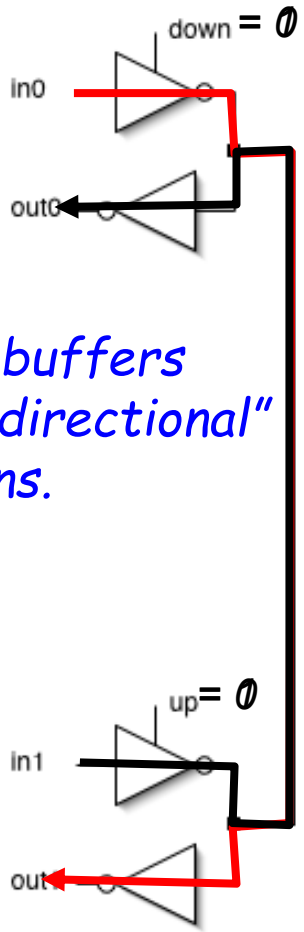
Inverted enable





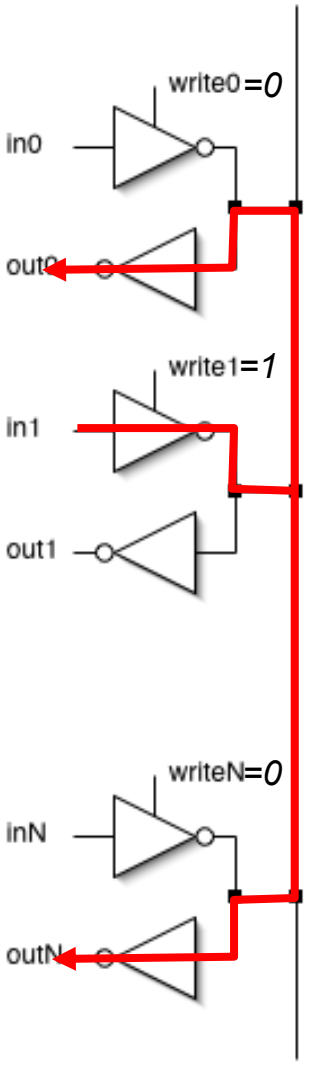
transmission gate useful in implementation

Tri-state Buffers



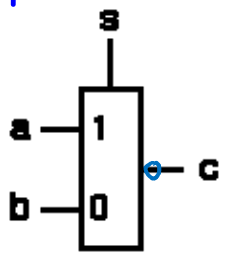
Tri-state buffers enable "bidirectional" connections.

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".

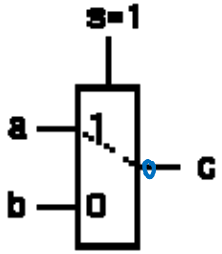


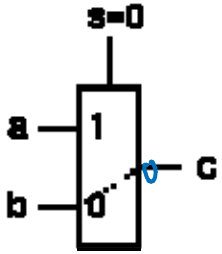
Tri-state Based Multiplexor

Multiplexor

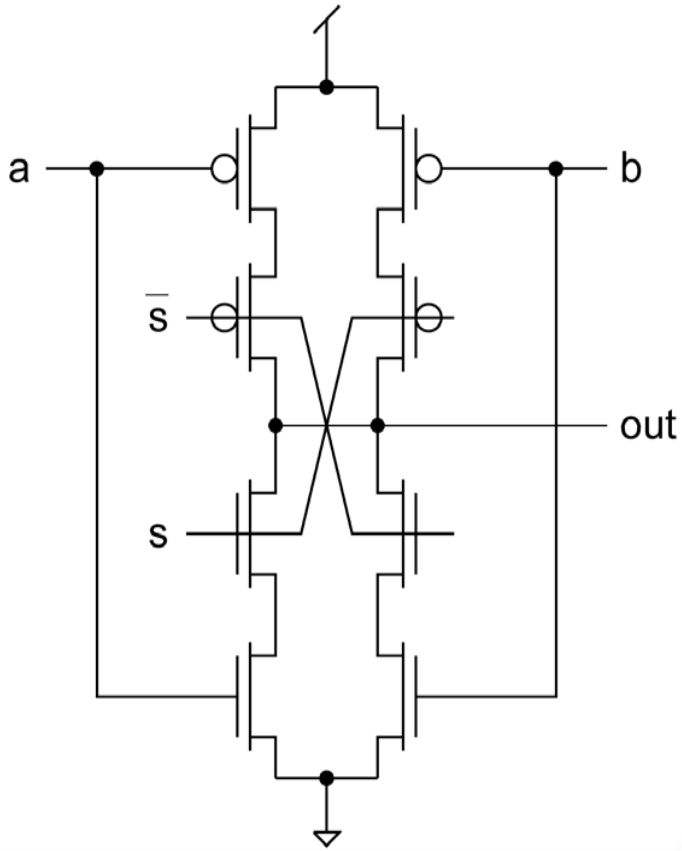


If s=1 then c=a



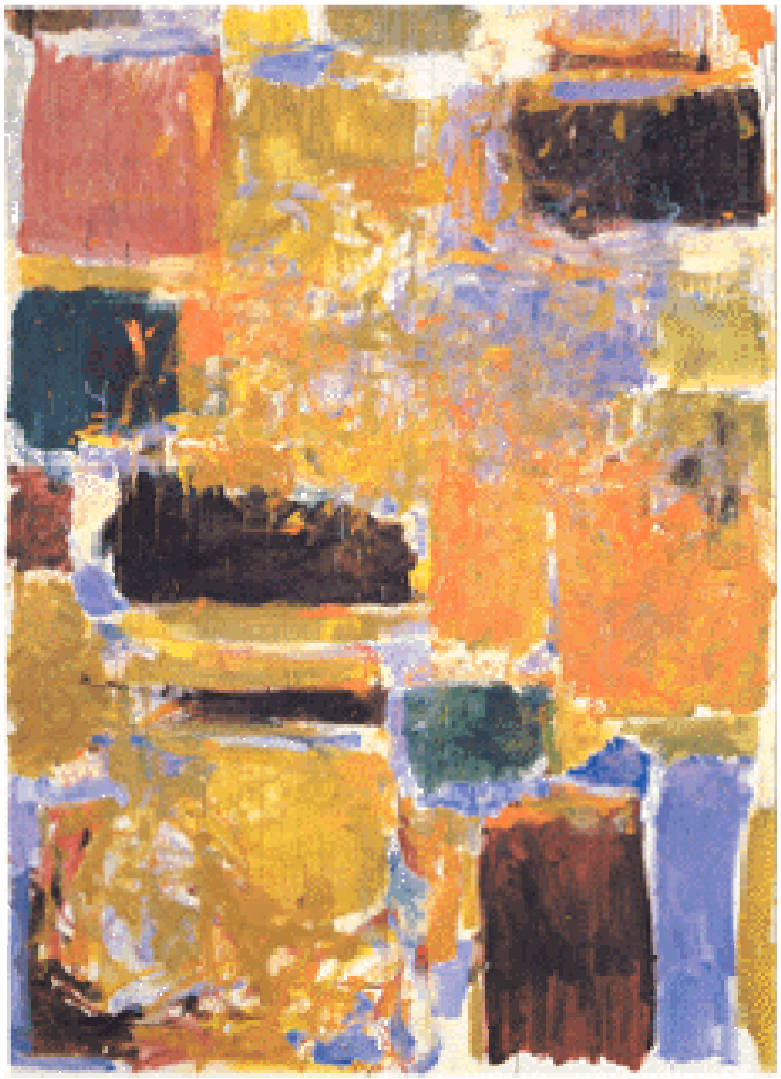


Transistor Circuit for inverting multiplexor:



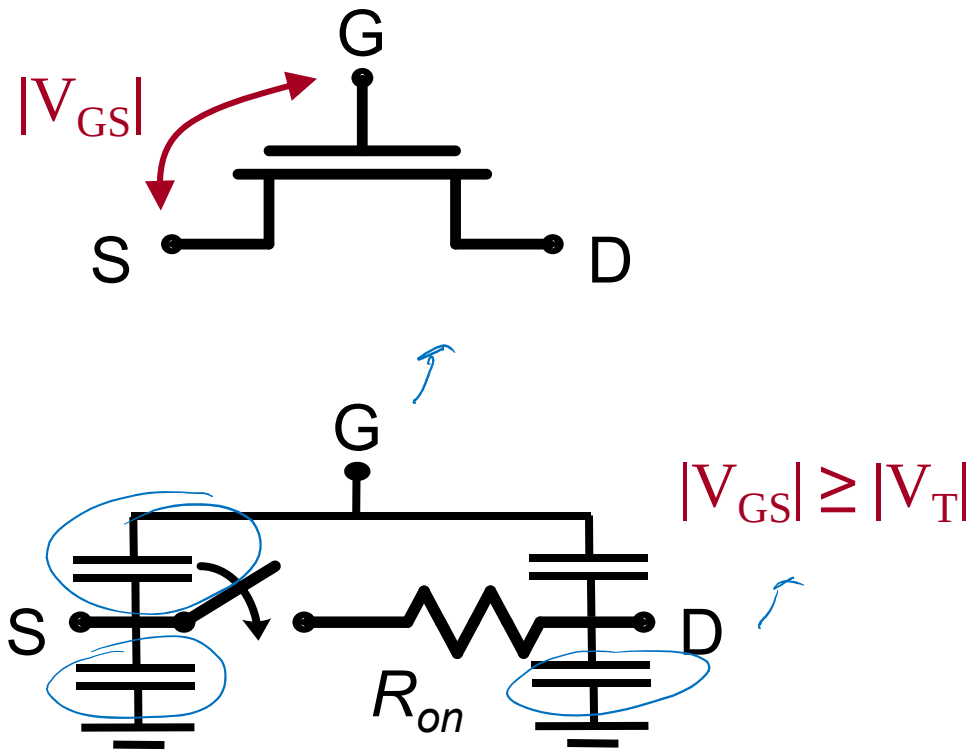
Summary: Complimentary CMOS Properties

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

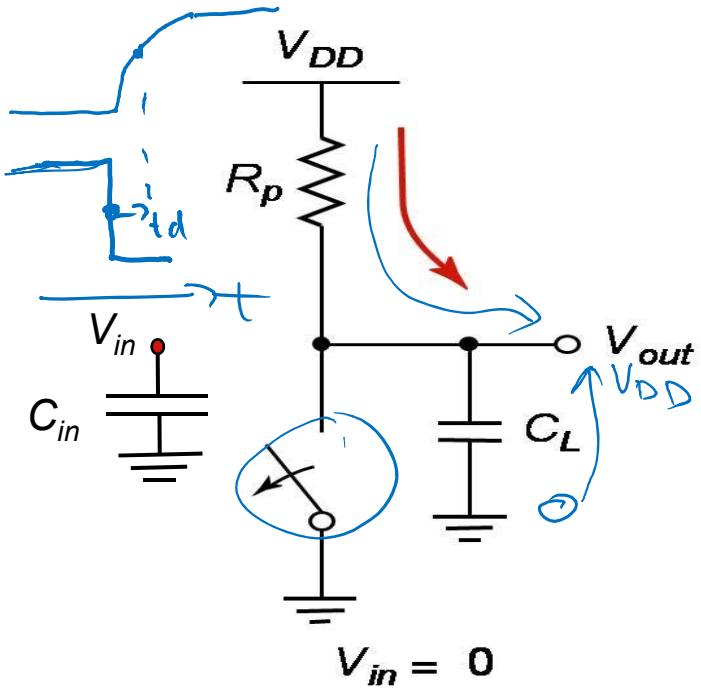


Transient properties

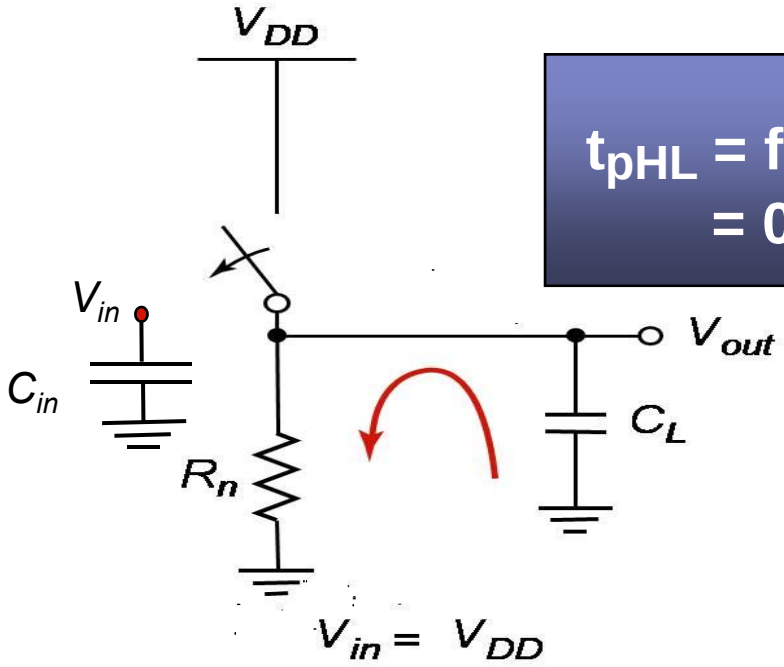
The Switch Revisited



The Switch Inverter: Transient Response



(a) Low-to-high

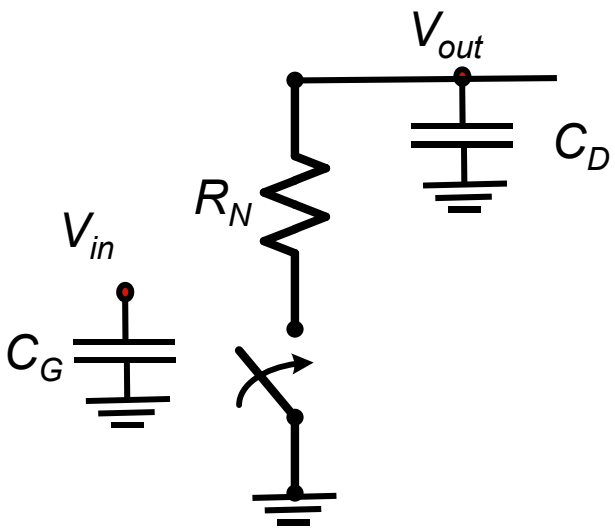


(b) High-to-low

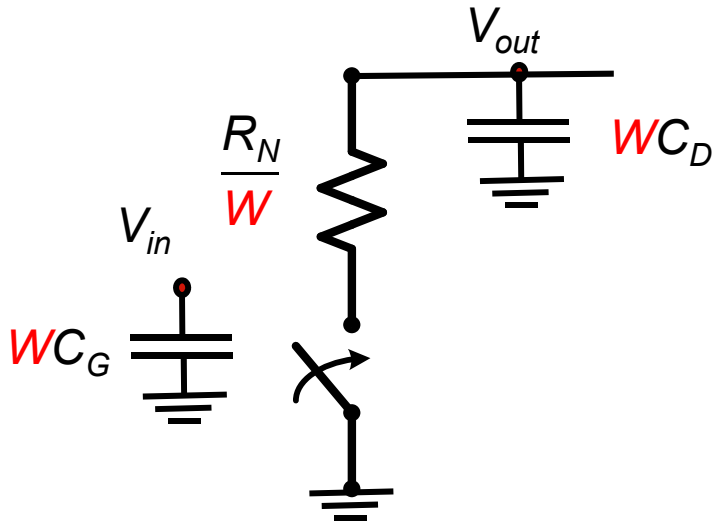
$t_{pHL} = f(R_{on}C_L)$
$= 0.69\ R_n C_L$

Switch Parasitic Model

The pull-down switch (NMOS)



Minimum-size switch

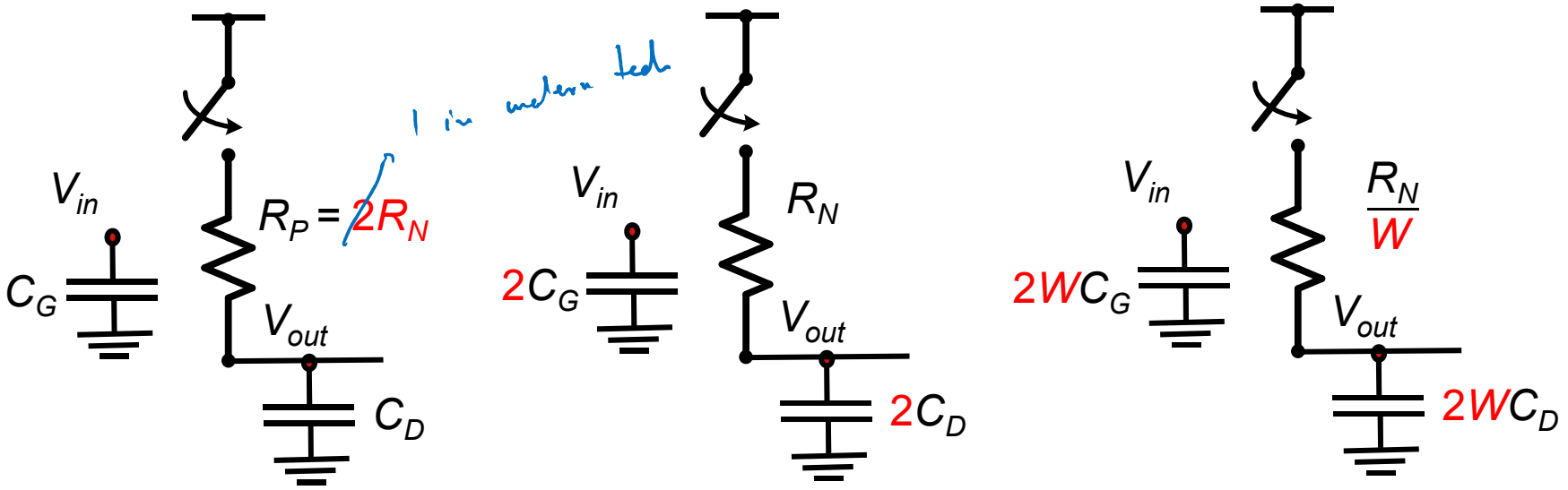


Sizing the transistor (factor W)

↑ Changing the width

Switch Parasitic Model

The pull-up switch (PMOS)

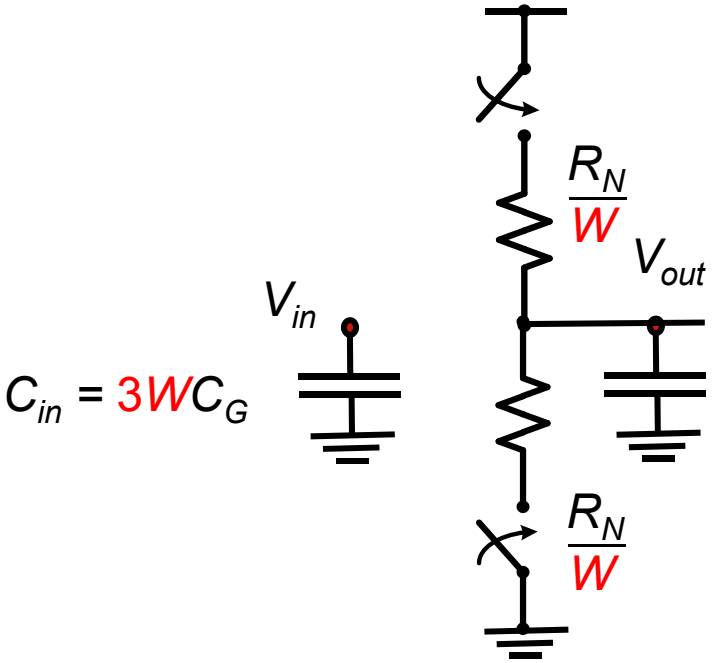


Minimum-size switch

Sized for symmetry

General sizing

Inverter Parasitic Model



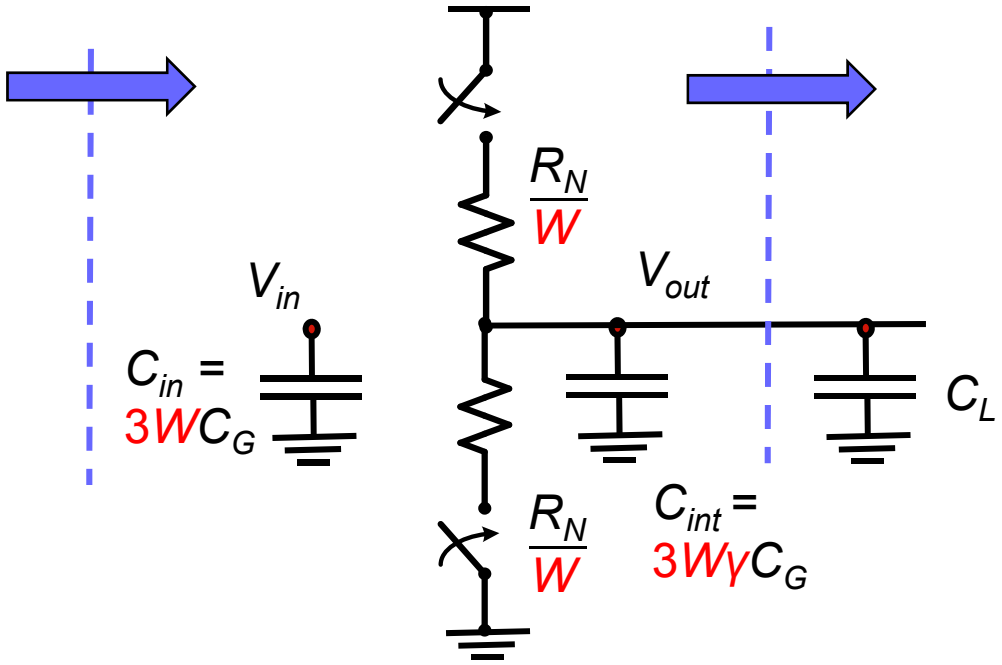
Drain and gate capacitance of transistor are **directly** related by process (γ≈1)

$$C_D = \gamma C_G$$

$$t_p = 0.69\left(\frac{R_N}{W}\right)(3W\gamma C_G) = 0.69(3\gamma)R_N C_G$$

Intrinsic delay of inverter
independent of size

Inverter with Load Capacitance



$$t_p = 0.69\left(\frac{R_N}{W}\right)(C_{\text{int}} + C_L)$$

$$= 0.69\left(\frac{R_N}{W}\right)(3W\gamma C_G + C_L)$$

$$= 0.69(3C_G R_N)(\gamma + \frac{C_L}{C_{in}})$$

$$= t_{inv}(\gamma + \frac{C_L}{C_{in}}) = t_0(\gamma + f)$$

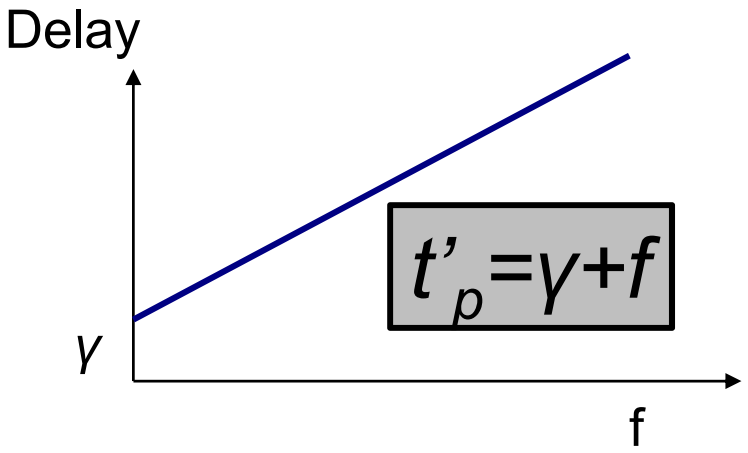
f = **fanout** = ratio between load and input capacitance of gate

Inverter Delay Model

$$t_p = t_{inv}(\gamma + f)$$

t_{inv} technology constant

- Can be dropped from expression
- Delay unit-less variable (expressed in unit delays)



Question: how does transistor sizing (W) impact delay?