# EECS 151/251A
## Spring 2019
## Digital Design and Integrated Circuits

Instructors:
Wawrzynek

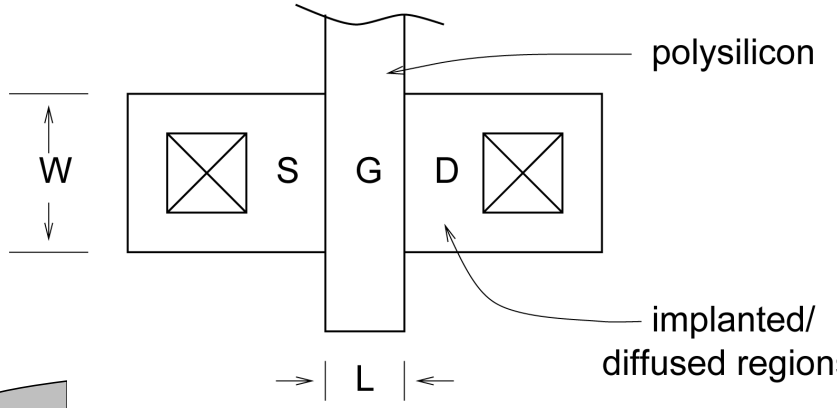
## Lecture 9

# CMOS abstraction

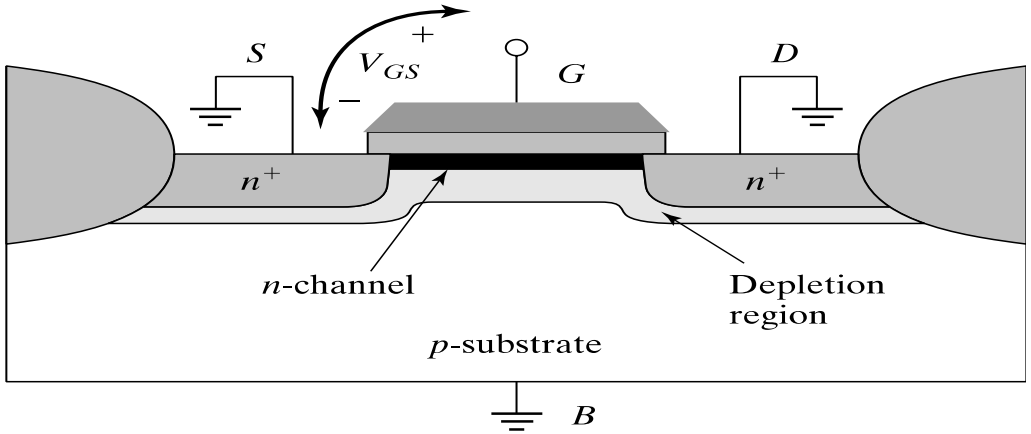
# CMOS Devices

- MOSFET (Metal Oxide Semiconductor Field Effect Transistor).

*Top View*

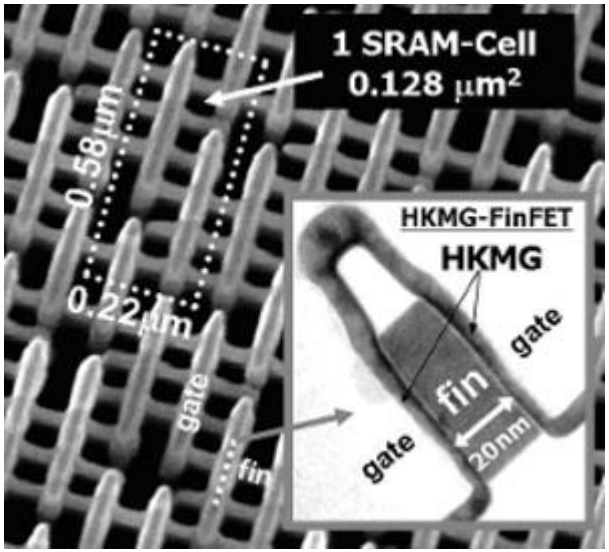


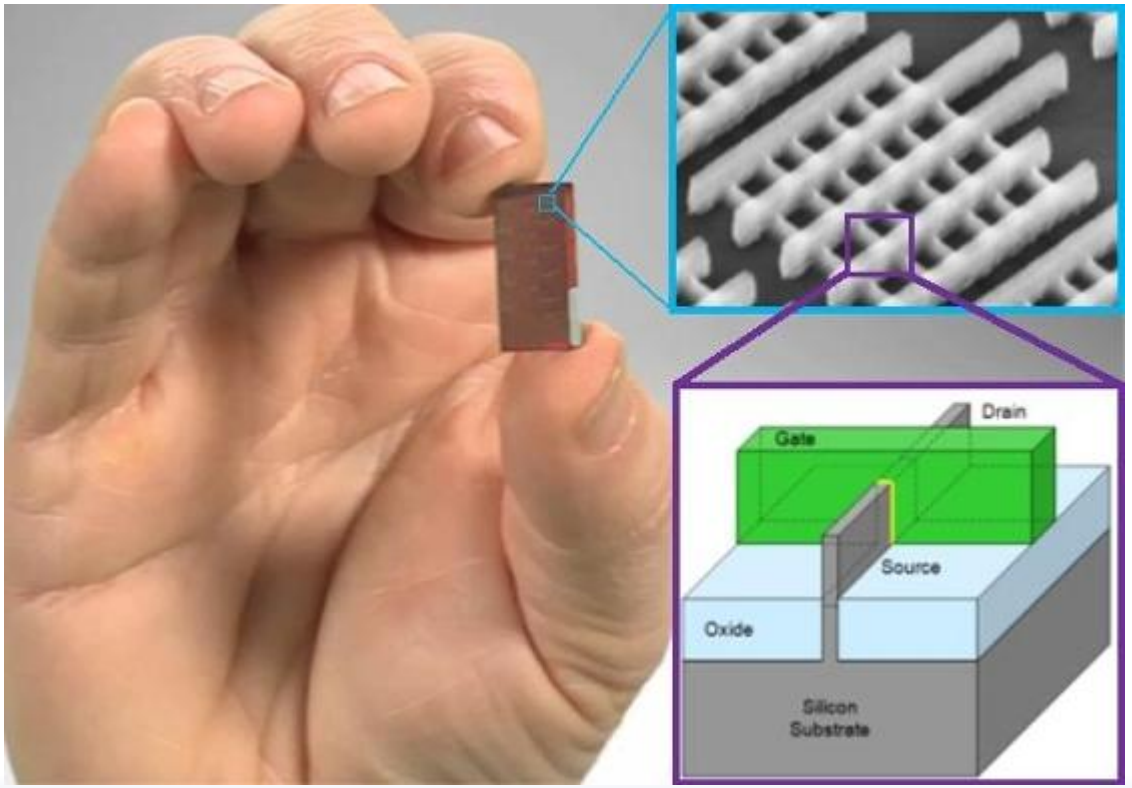

*Cross-section View*



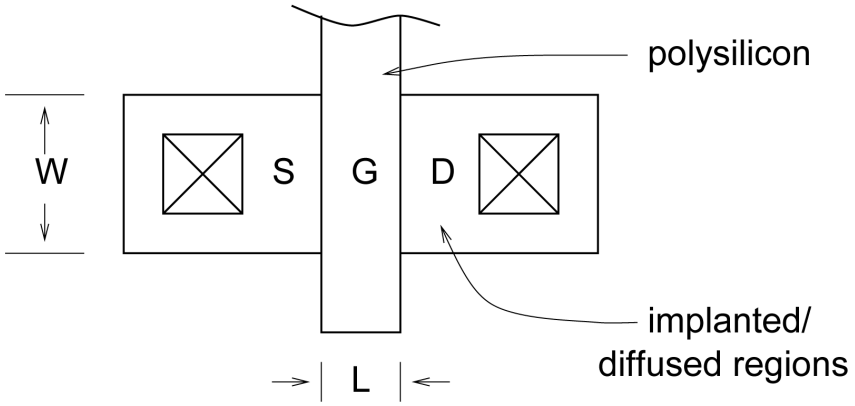

*The gate acts like a capacitor. A high voltage on the gate attracts charge into the channel. If a voltage exists between the source and drain a current will flow. In its simplest approximation, the device acts like a switch.*

# CMOS Transistors – State-of-the-Art



# Drain versus Source - Definition

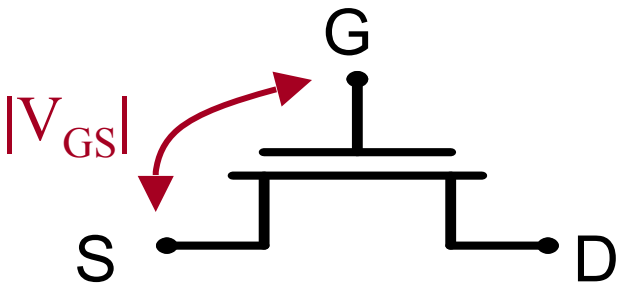


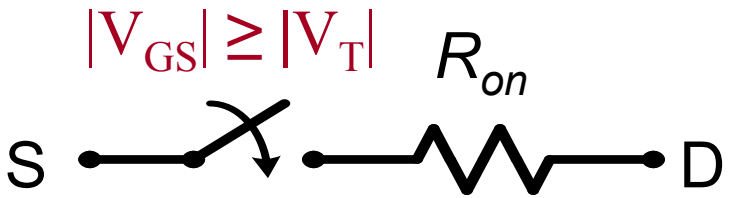

*MOS transistors are symmetrical devices (Source and drain are interchangeable)*

*Source is the node w/ the lowest voltage*

# MOS Transistor as a Resistive Switch

MOS Transistor ⟷ A Switch!

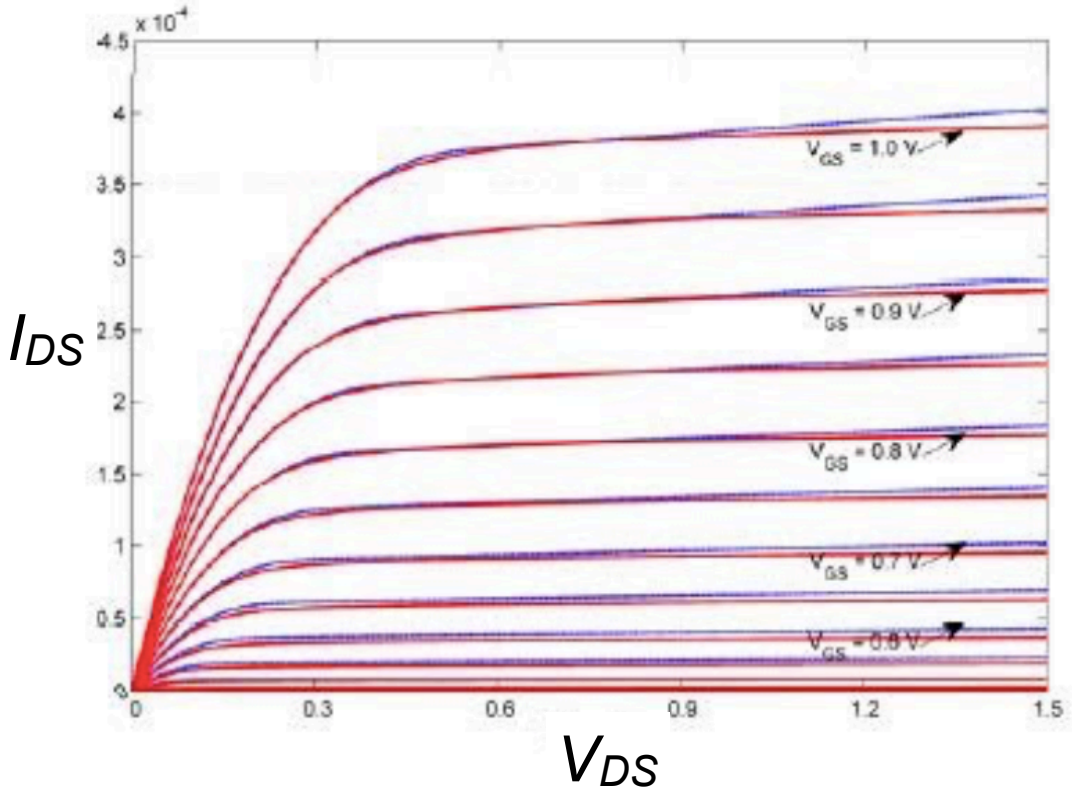


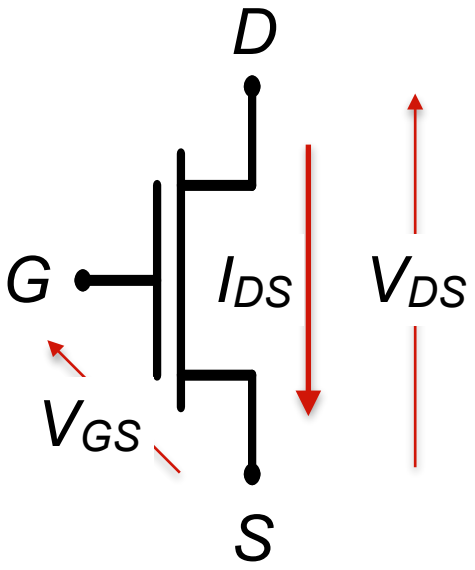




*Let's look beneath the abstraction:*
*origins of $R_{on}$ and $V_T$*

# Transistor "resistance"

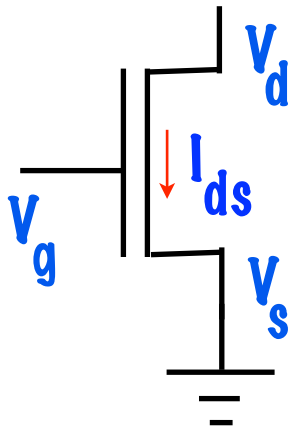
- Actually, nonlinear I/V characteristic:



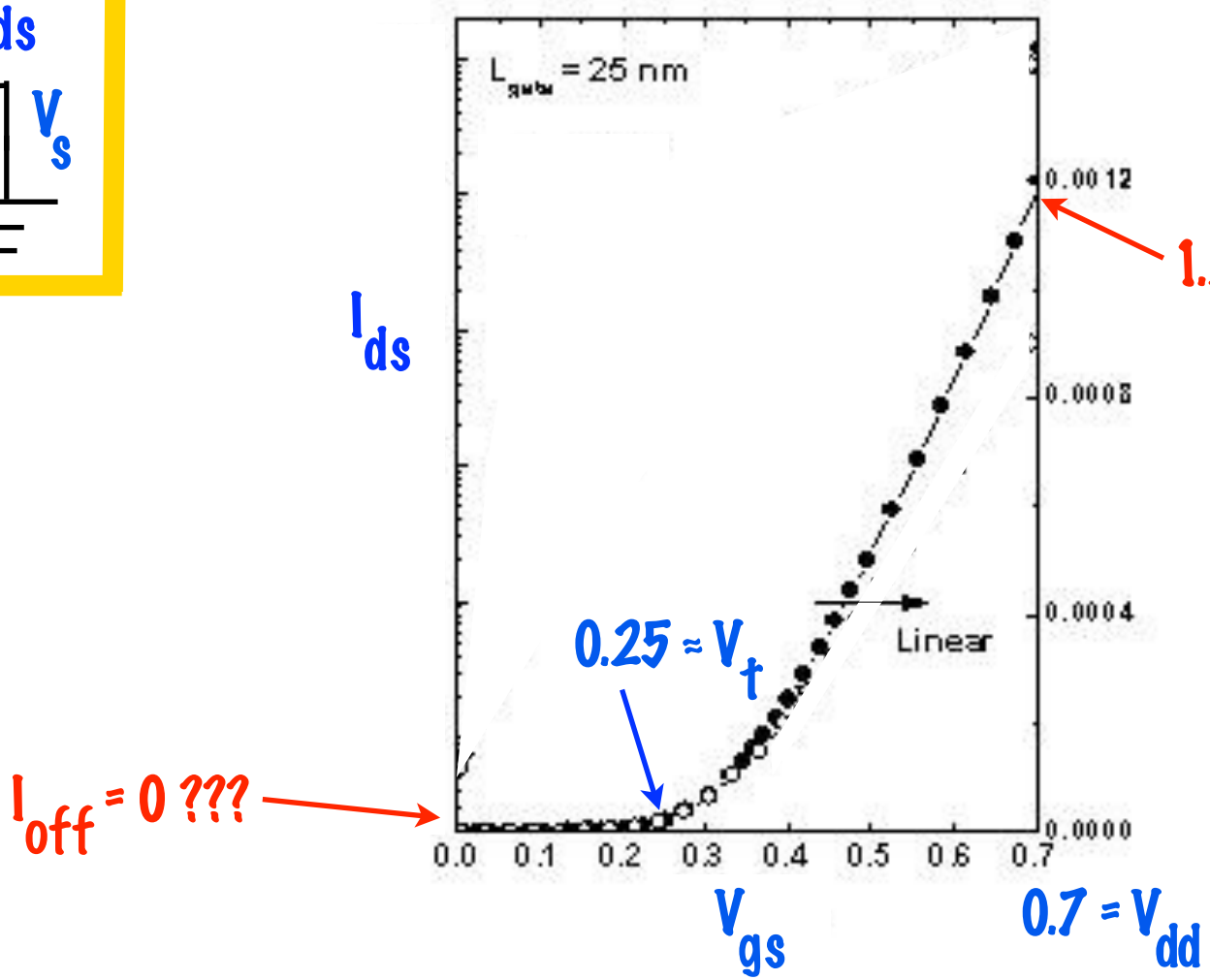




- Linearizing makes all delay and power calculations simple (usually just 1st order ODEs):

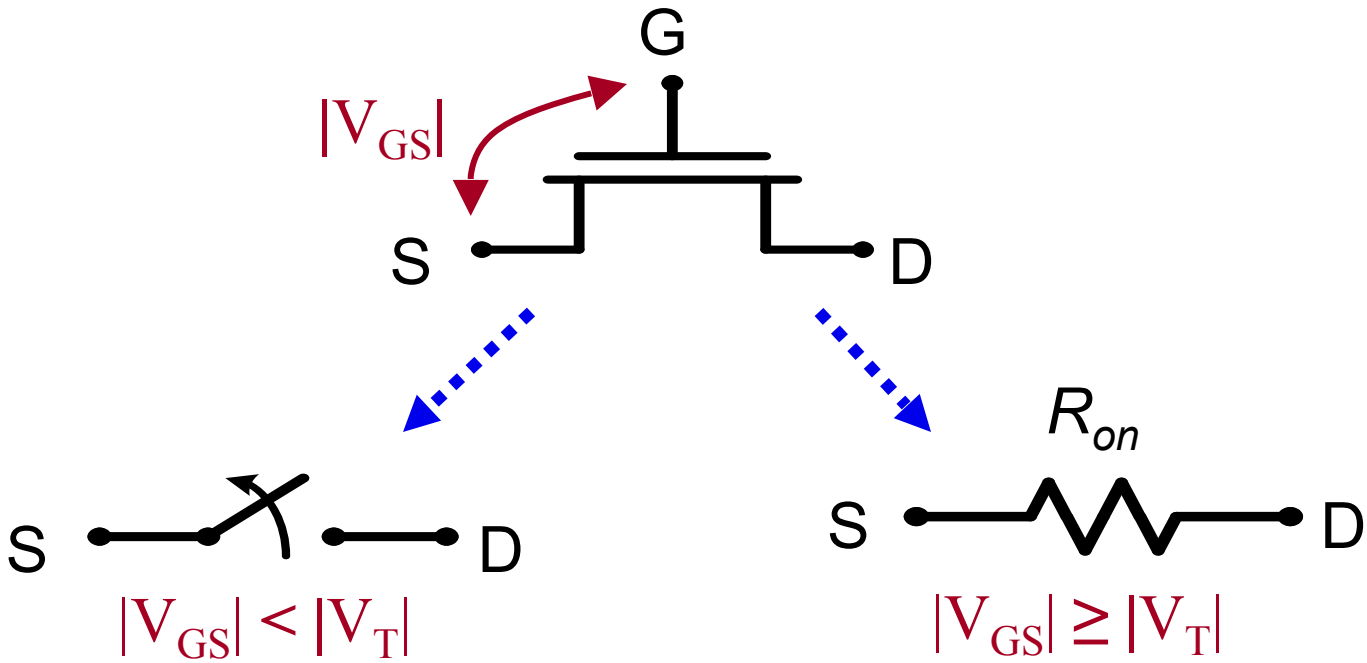
# MOSFET Threshold Voltage



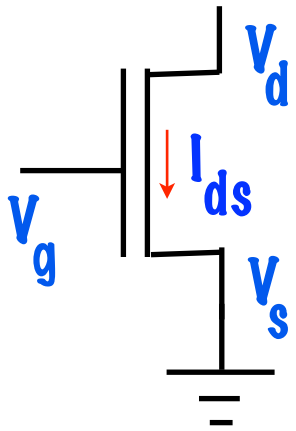

Transistor "turns on" when $V_{gs}$ is > $V_t$.



# ON/OFF Switch Model of MOS Transistor



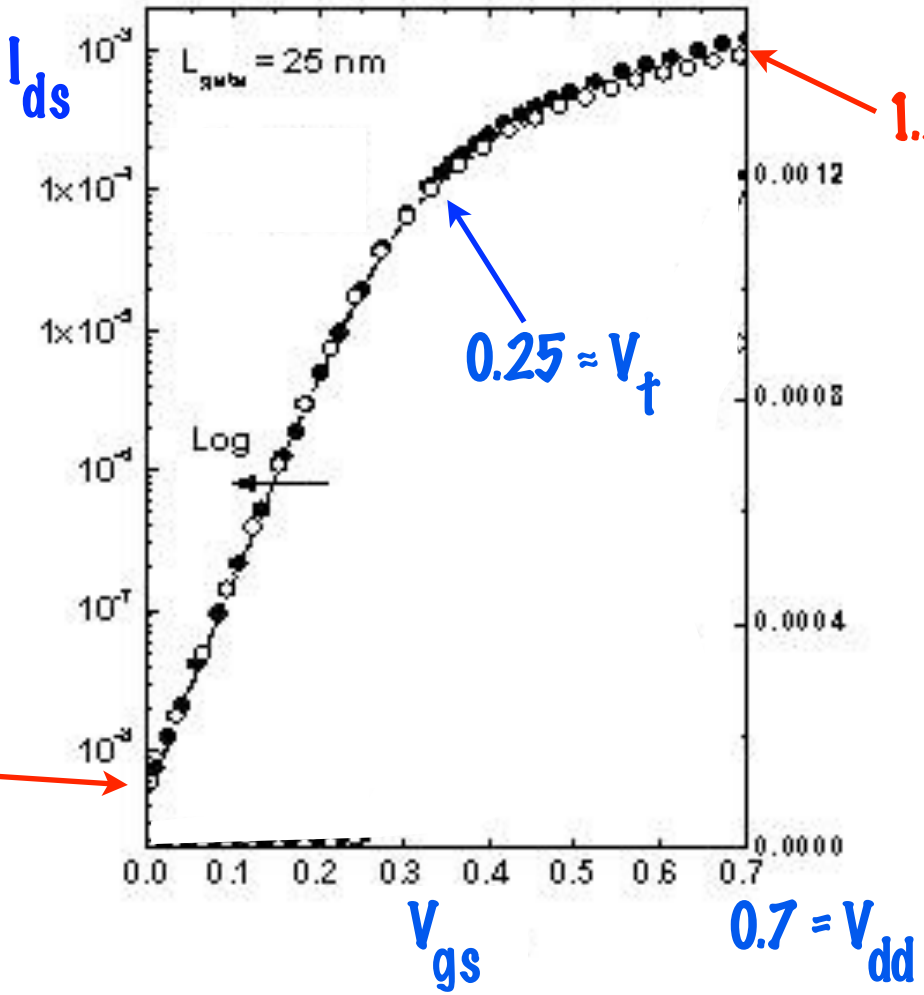
# Plot on a "Log" Scale to See "Off" Current



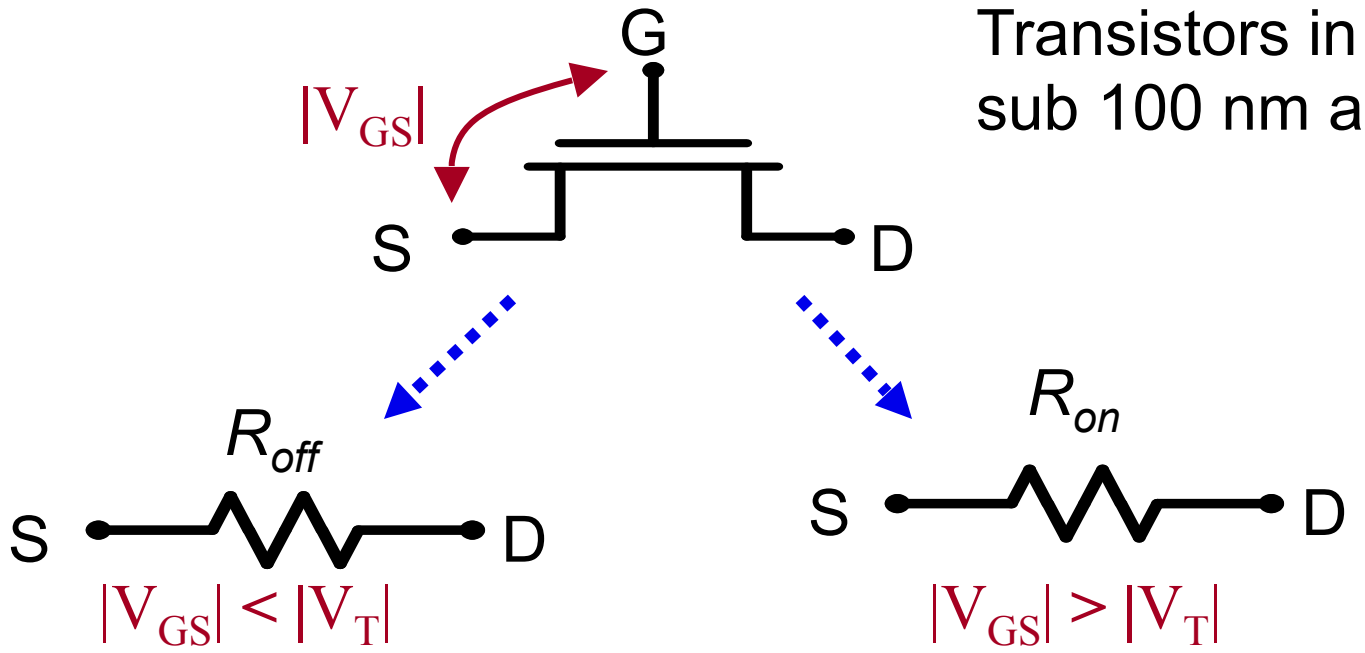

Process engineers can:

increase $I_{on}$ by lowering $V_t$ - but that raises $I_{off}$

decrease $I_{off}$ by raising $V_t$ - but that lowers $I_{on}$.

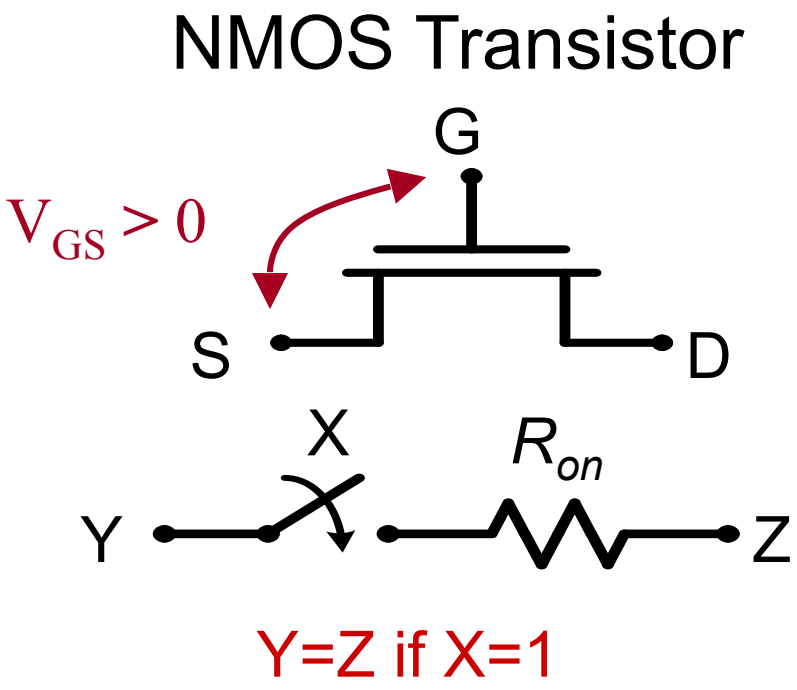


# A More Realistic Switch

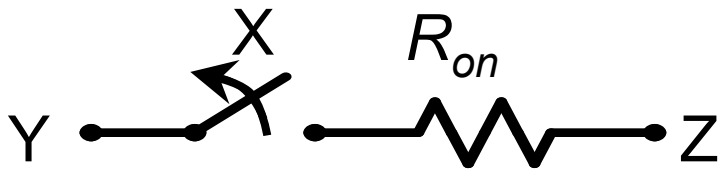



Transistors in the sub 100 nm age

# A Logic Perspective

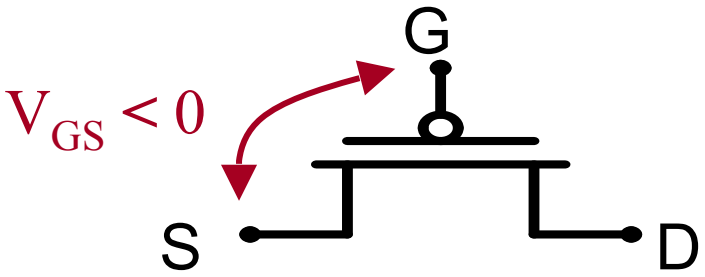


# A Complementary Switch



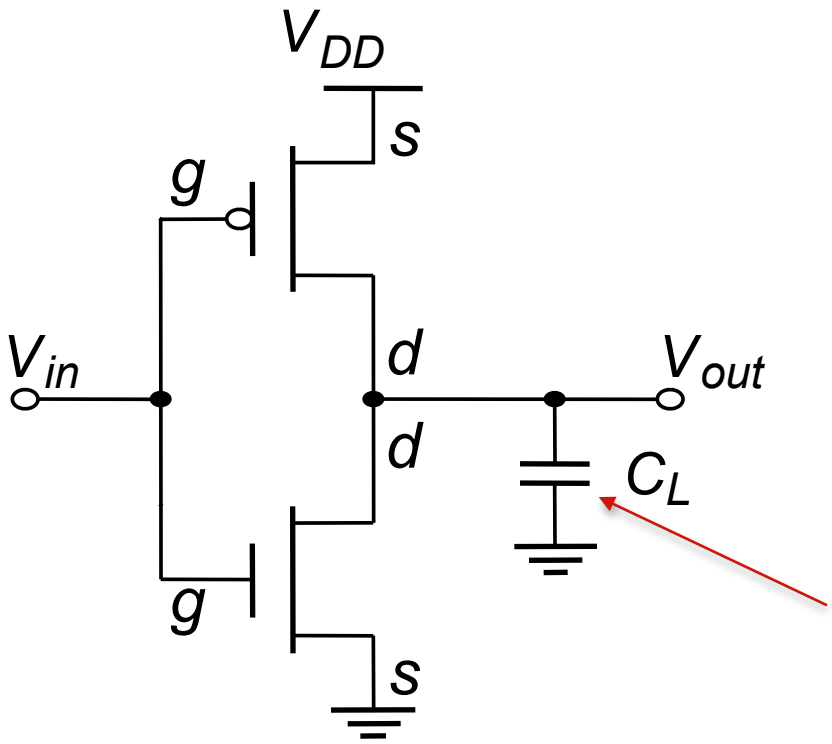

Y=Z if X=0

PMOS Transistor



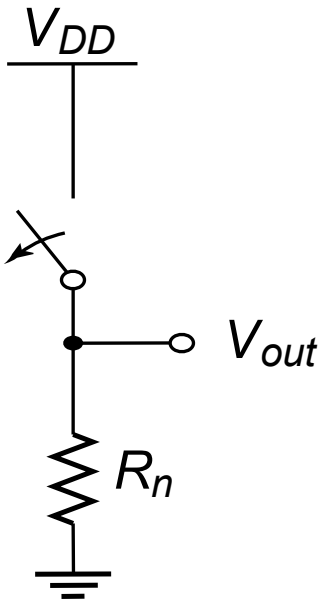

*Source is the node w/ the highest voltage!*

# The CMOS Inverter: A First Glance

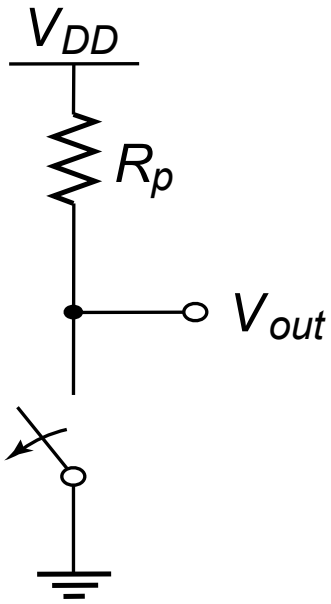



*Represents the sum of all the capacitance at the output of the inverter and everything to which it connects: (drains, interconnections gate capacitance of next gate(s))*

# The Switch Inverter
## First-Order DC Analysis*




$V_{in} = V_{DD}$




$V_{in} = 0$

$V_{OL} = 0$
$V_{OH} = V_{DD}$

*First-order means we will ignore Capacitance.

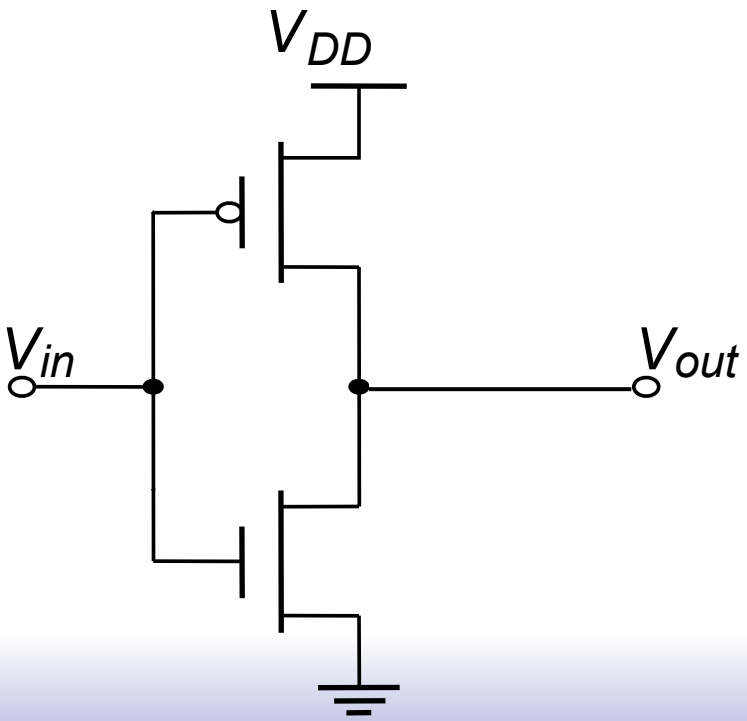
# Switch logic

# Static Logic Gate

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

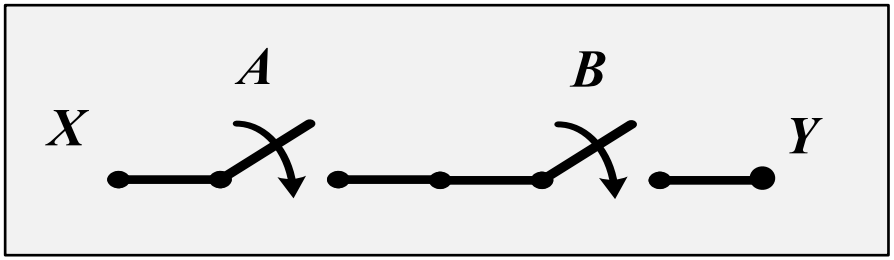
- The output of the gate assumes at all times the value of the Boolean function implemented by the circuit (ignoring, once again, the transient effects during switching periods).

Example: CMOS Inverter



# Building logic from switches

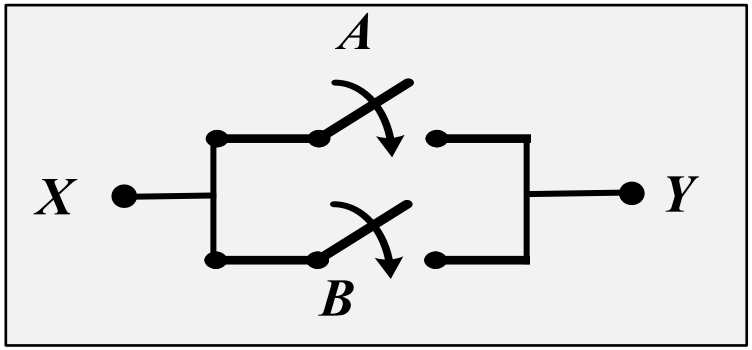
**Series**




**AND**

$Y = X$ if $A$ AND $B$

**Parallel**



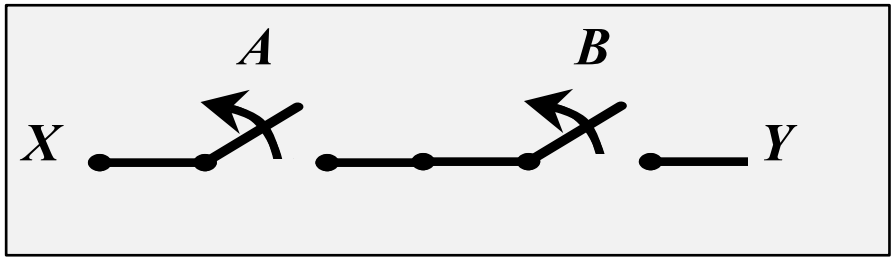

**OR**

$Y = X$ if $A$ OR $B$

(output undefined if condition not true)

# Logic using inverting switches

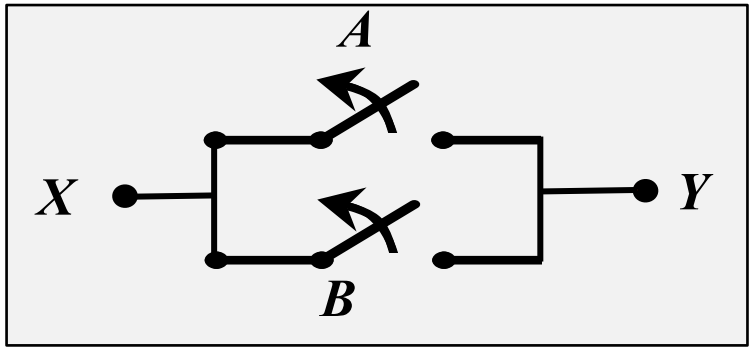
**Series**




**NOR**

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

**Parallel**

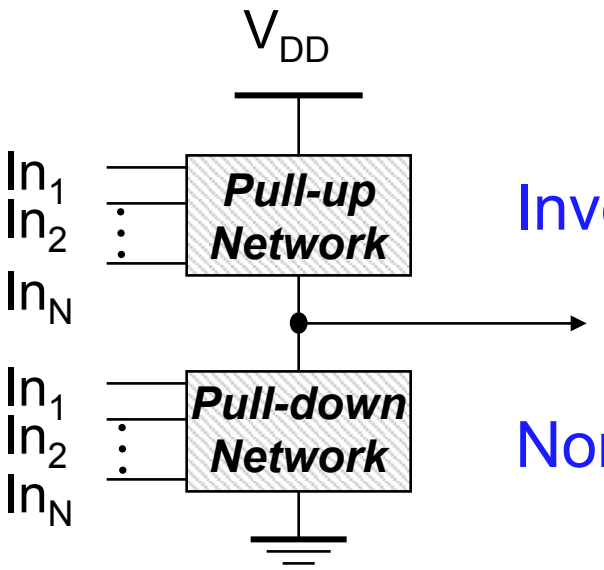



**NAND**

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

(output undefined if condition not true)

# Static Complementary CMOS



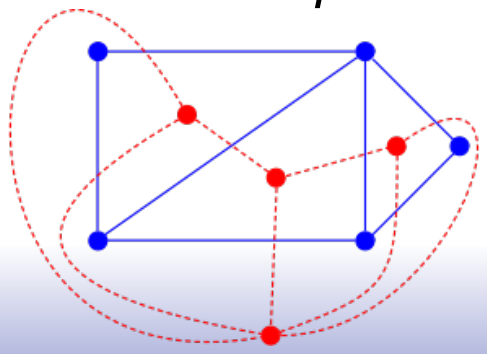

Inverting switches

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

Non-Inverting switches

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

*Dual Graphs*

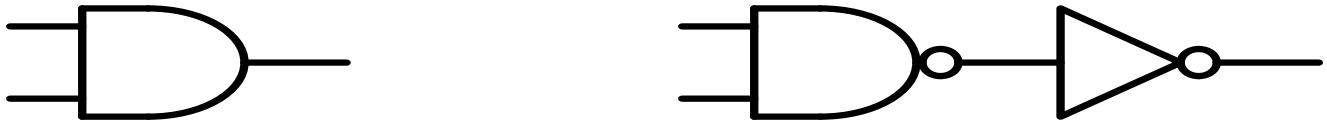


# Complementary CMOS Logic Style

- 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

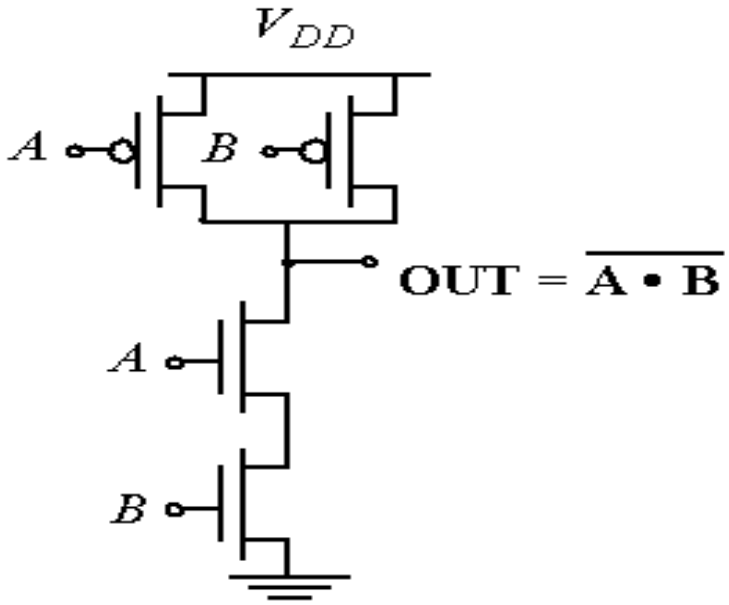


**AND = NAND + INV**

# 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

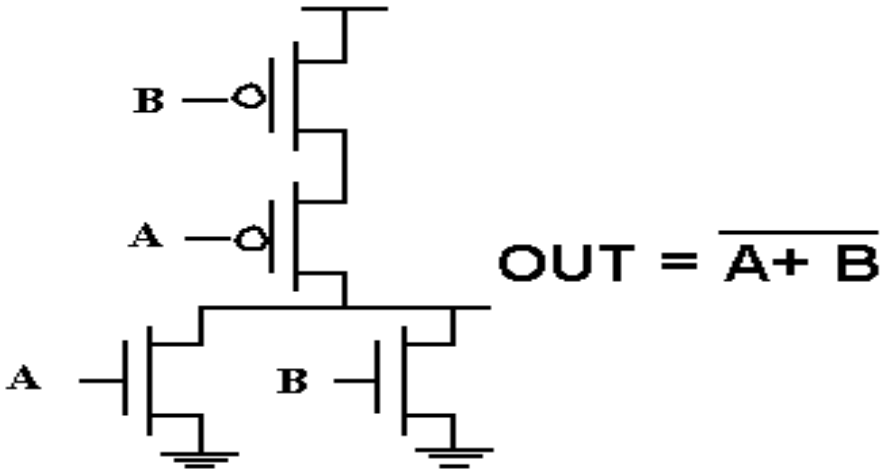



- PDN: G = AB $\Rightarrow$ Conduction to GND
- PUN: F = $\overline{A}$ + $\overline{B}$ = $\overline{AB}$ $\Rightarrow$ Conduction to $V_{DD}$
- $\overline{G(In_1,In_2,In_3,\ldots)} \equiv F(\overline{In_1},\overline{In_2},\overline{In_3},\ldots)$

# Example Gate: NOR

| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

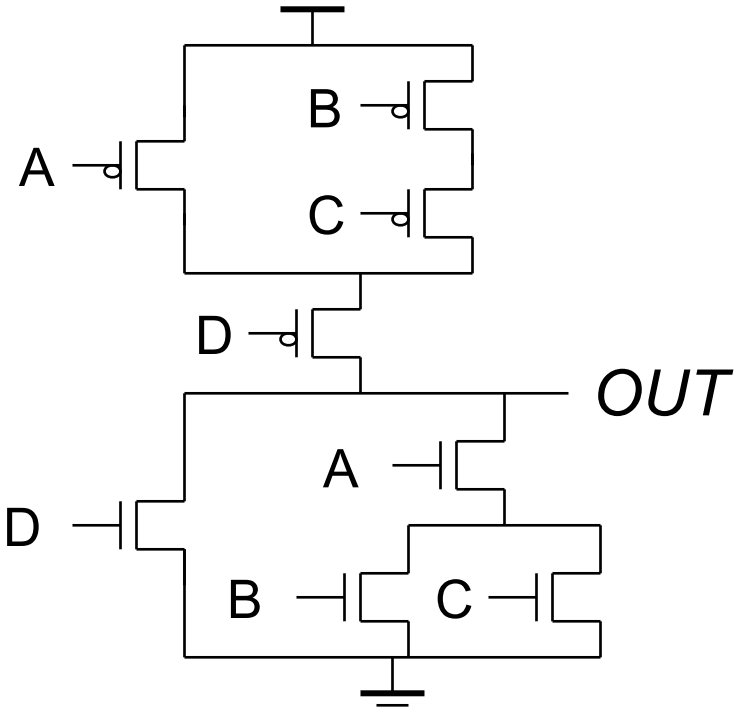
Truth Table of a 2 input NOR gate



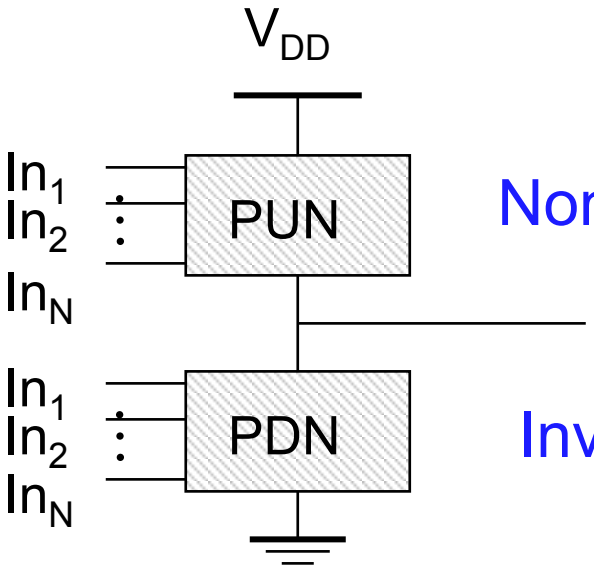
# Complex CMOS Gate

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

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



# Non-inverting logic




Non-Inverting switches

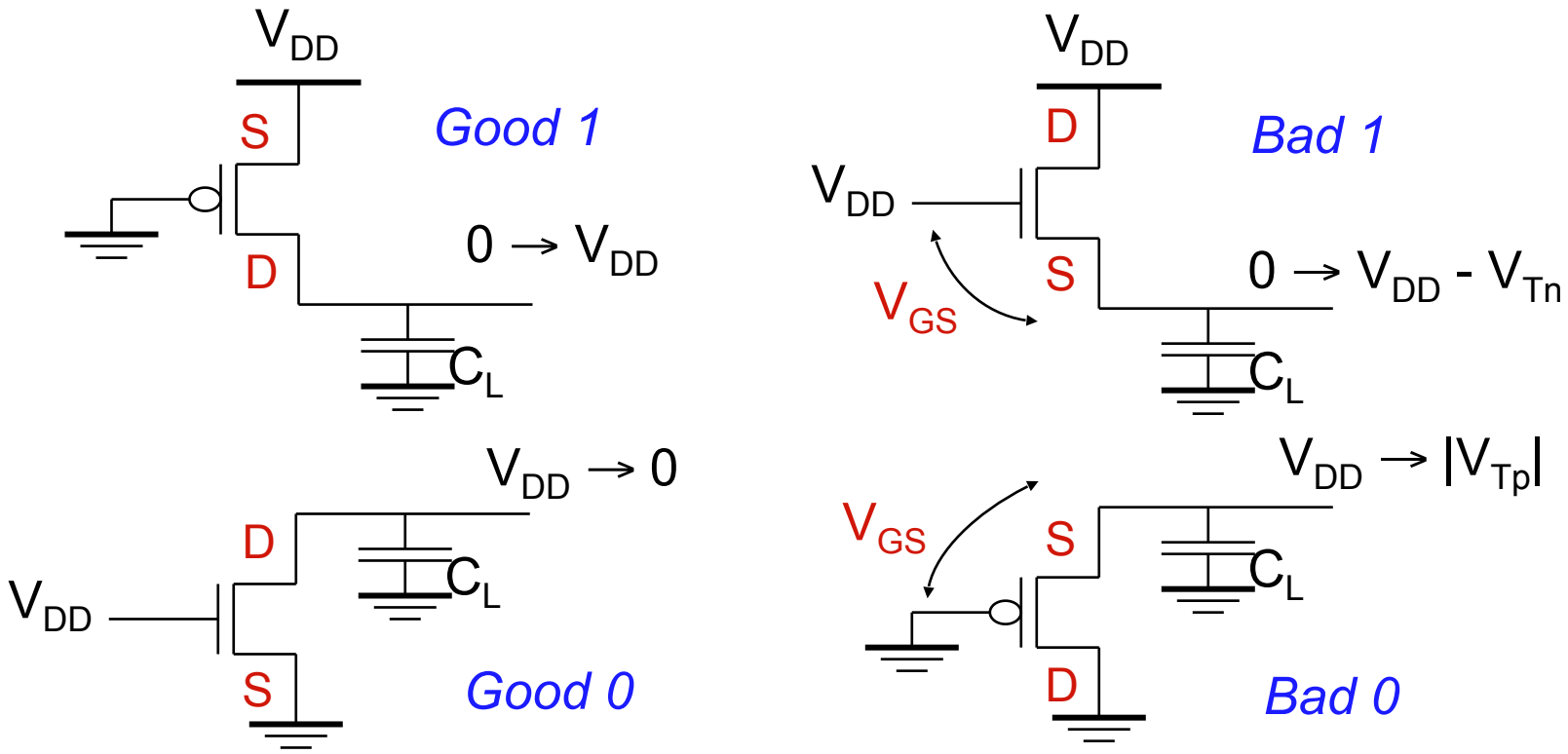
$F(In_1,In_2,\ldots In_N)$

Inverting switches

*Why is this a bad idea?*

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

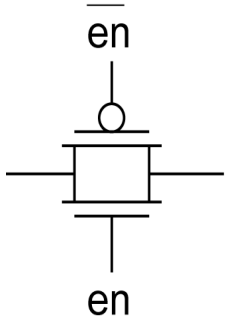
# Switch Limitations




*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).

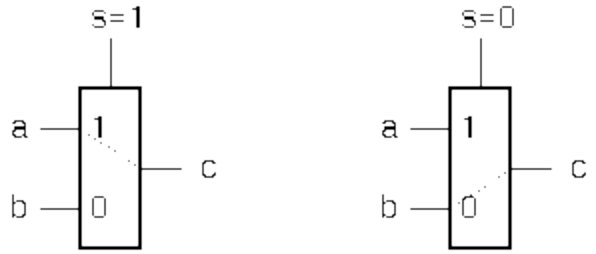


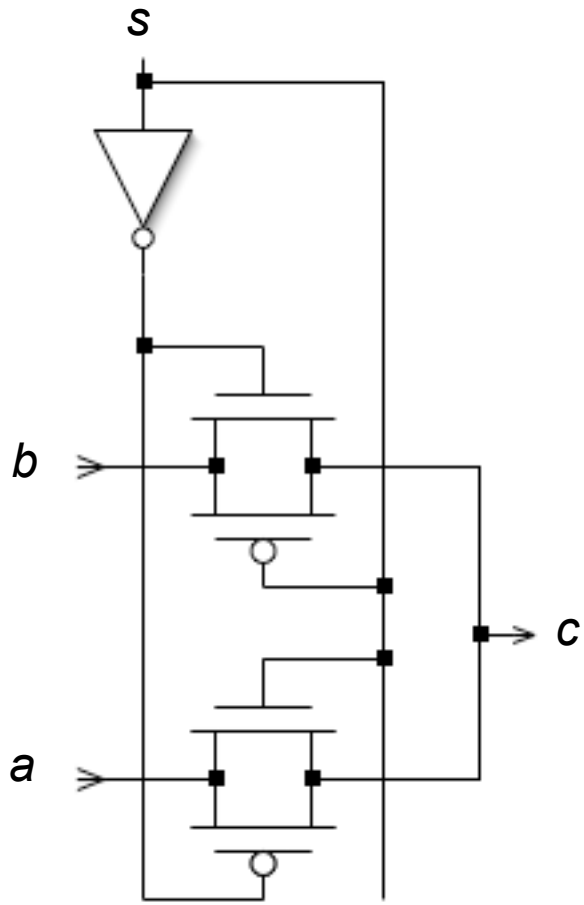

- Does not directly connect to Vdd and GND, but can be combined with logic gates or buffers to simplify many logic structures.

# Transmission-gate Multiplexor

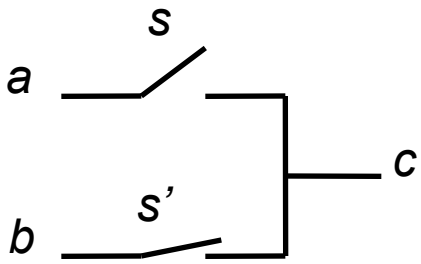
2-to-1 multiplexor:

$c = sa + s'b$



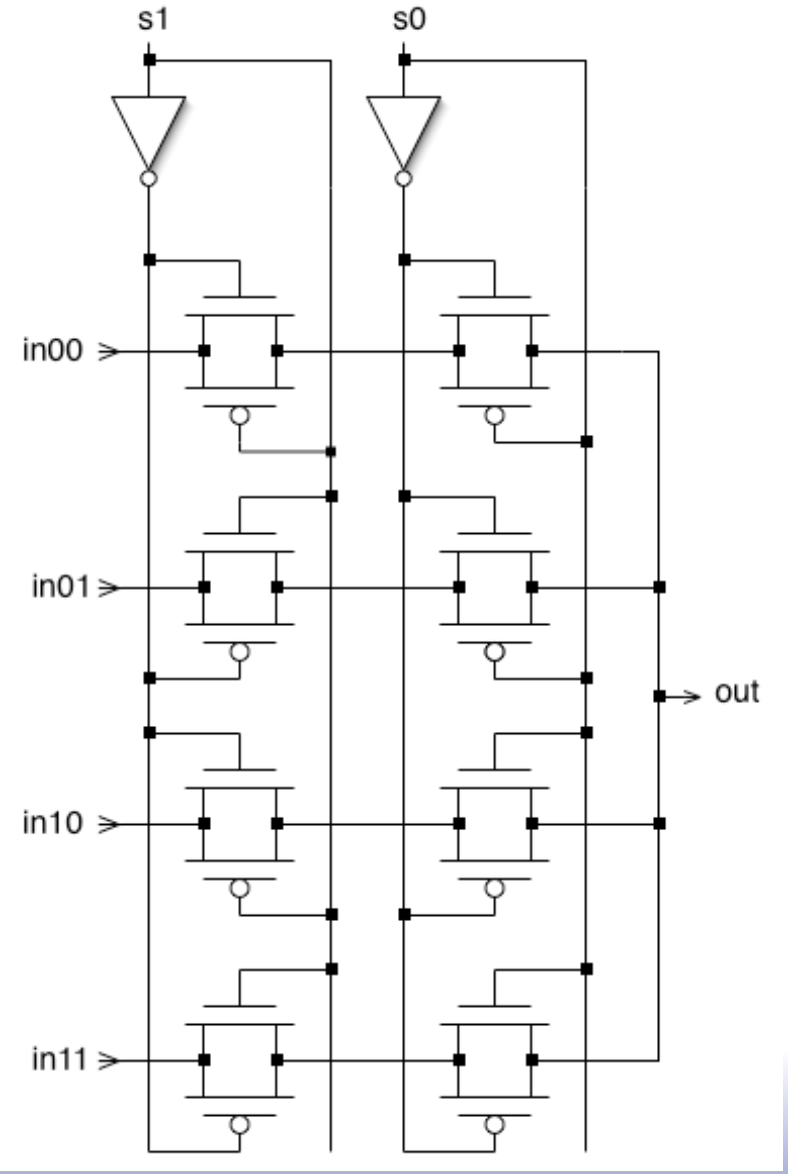

Switches simplify the implementation:




Compare the cost to logic gate implementation.

Care must be taken to not string together many pass-transistor stages. Occasionally, need to "rebuffer" with static gate.

# 4-to-1 Transmission-gate Mux

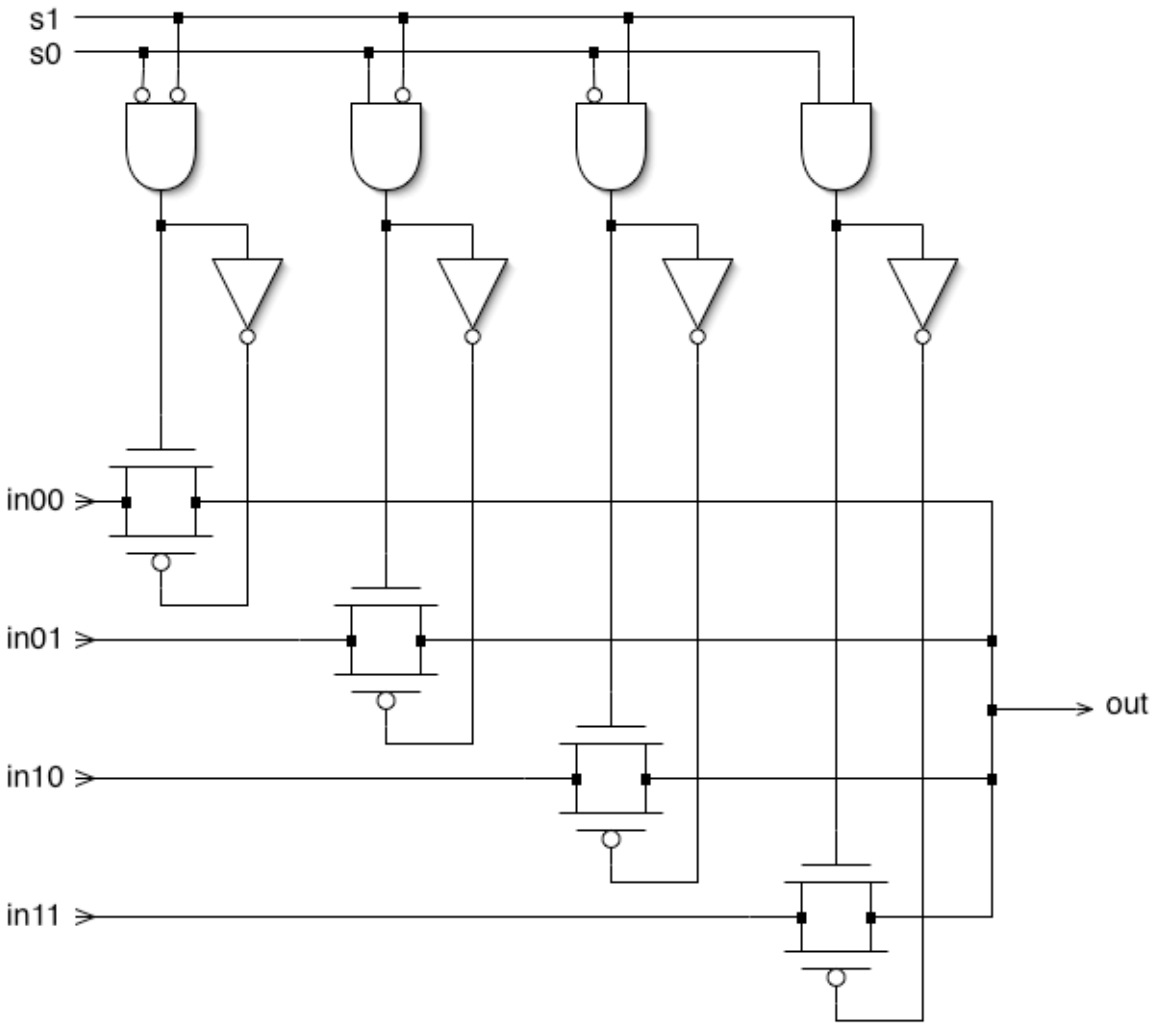



- The series connection of pass-transistors in each branch effectively forms the AND of s1 and s0 (or their complement).
- Compare cost to logic gate implementation

*Any better solutions?*

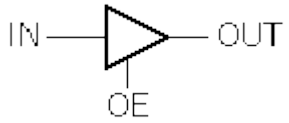
# Alternative 4-to-1 Multiplexor

- 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

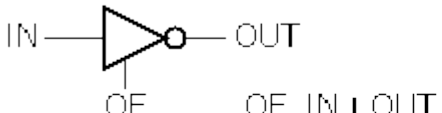
*Tri-state Buffer:*




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

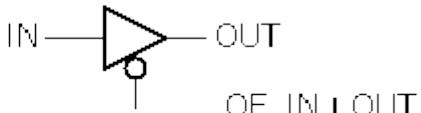
*"high impedance" (output disconnected)*

## Variations:



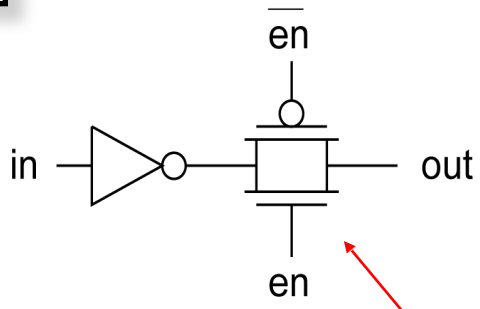

| OE | IN | OUT |
|---|---|---|
| 0 | – | Z |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

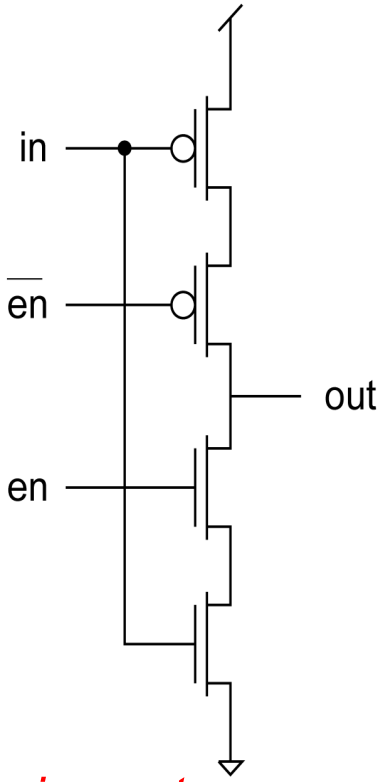
*Inverting buffer*




| OE | IN | OUT |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | – | Z |

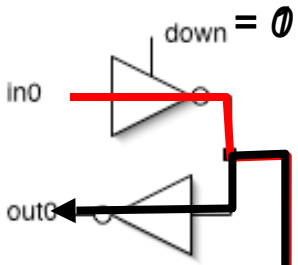
*Inverted enable*



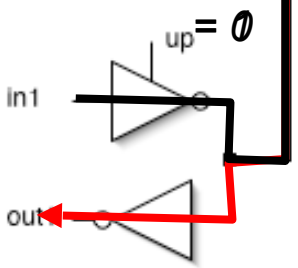




*transmission gate useful in implementation*

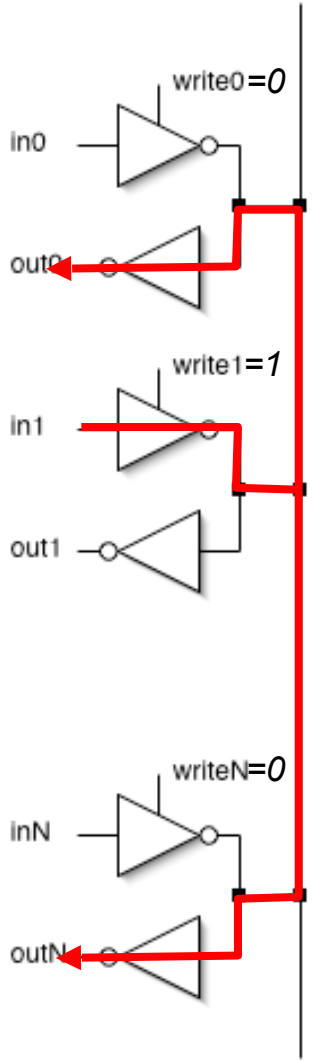
# Tri-state Buffers




Tri-state buffers enable "bidirectional" connections.

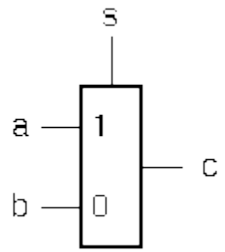



Tri-state buffers are used when multiple circuits all connect to a common wire. Only one circuit at a time is allowed to drive the bus. All others "disconnect" their outputs, but can "listen".

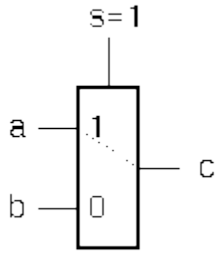


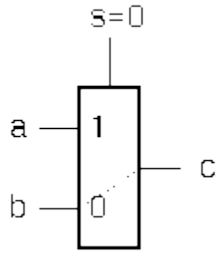
# Tri-state Based Multiplexor

Multiplexor

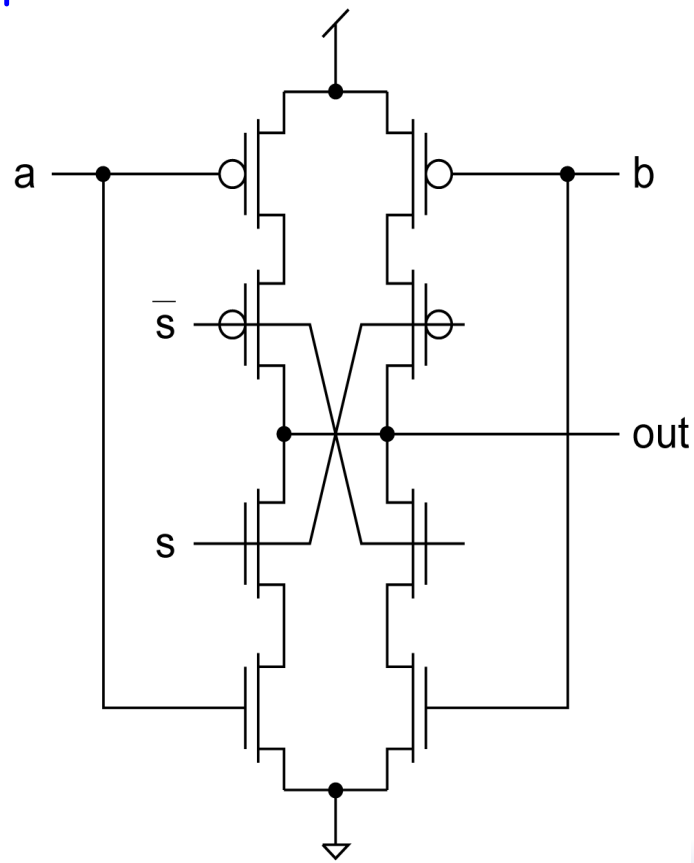



If s=1 then c=a else c=b



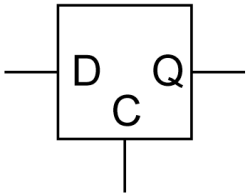



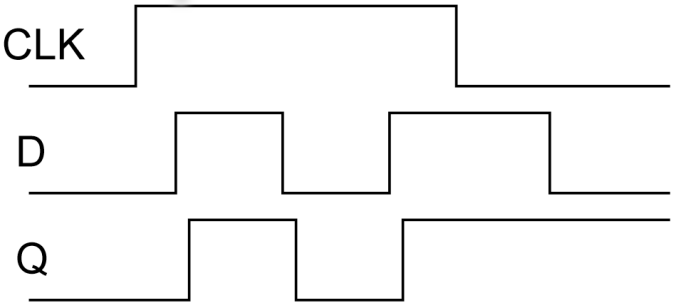

Transistor Circuit for inverting multiplexor:



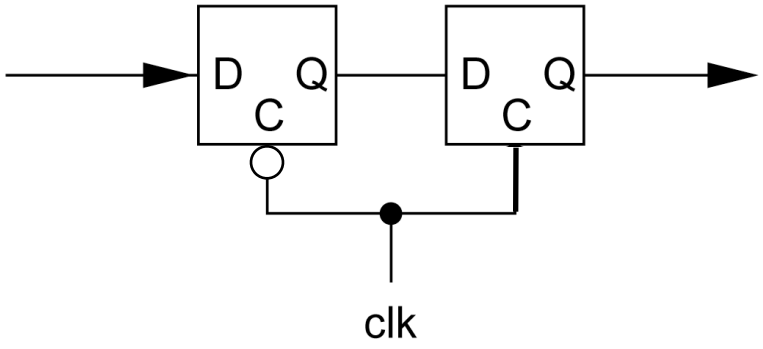
# Latches and Flip-flops

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

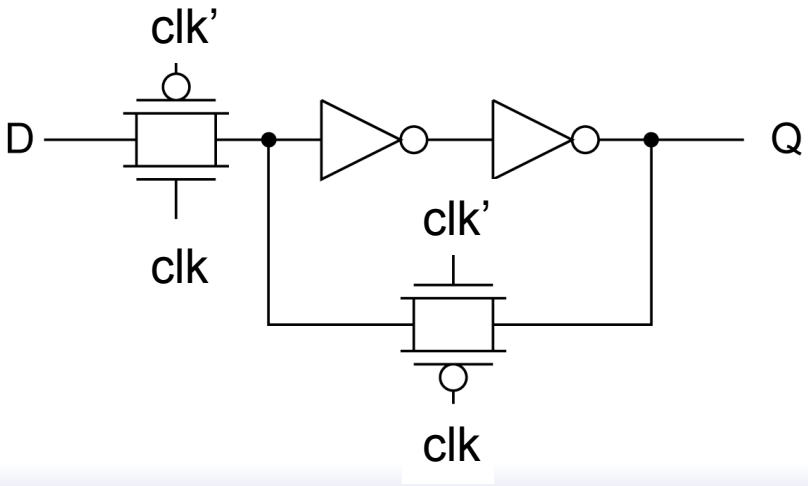



Positive Edge-triggered **flip-flop** built from two level-sensitive latches:




**Latch** Implementation:



# Summary: Complimentary CMOS Properties

- Full rail-to-rail swing
- Besides leakage (due to $I_{off}$), no static power dissipation
- Direct path current during switching