



EECS 151/251A
Spring 2023
Digital Design and
Integrated Circuits

Instructors:
Wawrzynek

Lecture 10: CMOS2

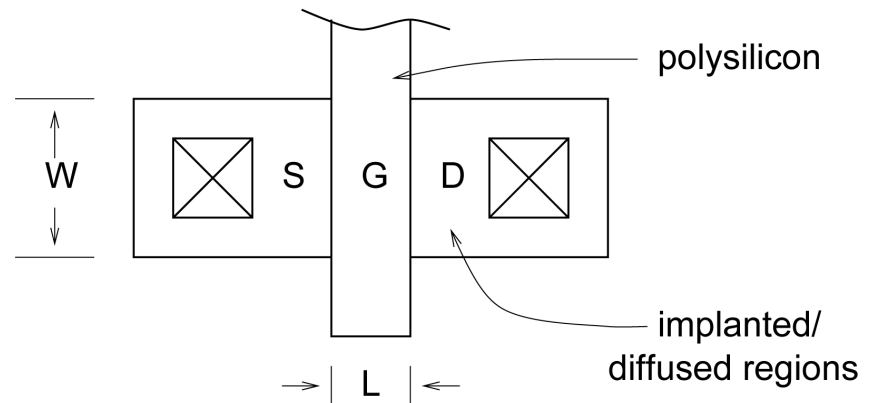
Announcements



CMOS Transistors

Transistor Strength and Symmetry

1. Transistor “strength” proportional to W/L . In digital circuits, L is almost always minimal allowed by process.

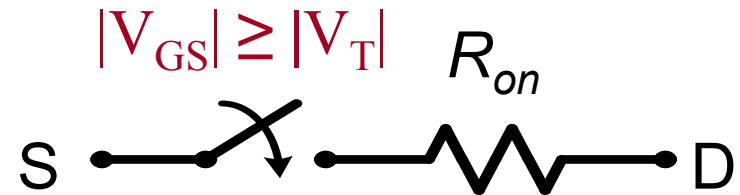
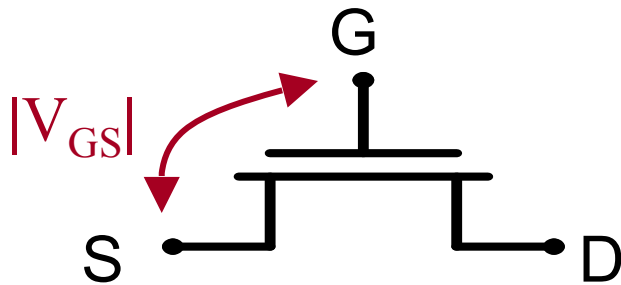


2. MOS transistors are symmetrical devices (Source and drain are interchangeable). But usually designed to be used in one direction.

For nFET, source is the node w/ the lowest voltage. For pFET source is node with highest voltage.

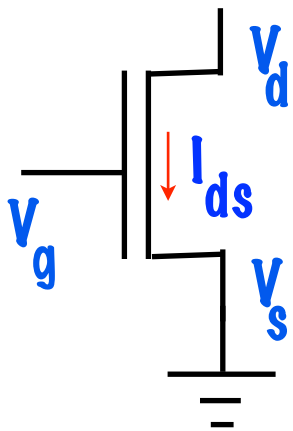
MOS Transistor as a Resistive Switch

MOS Transistor \leftrightarrow A Switch!

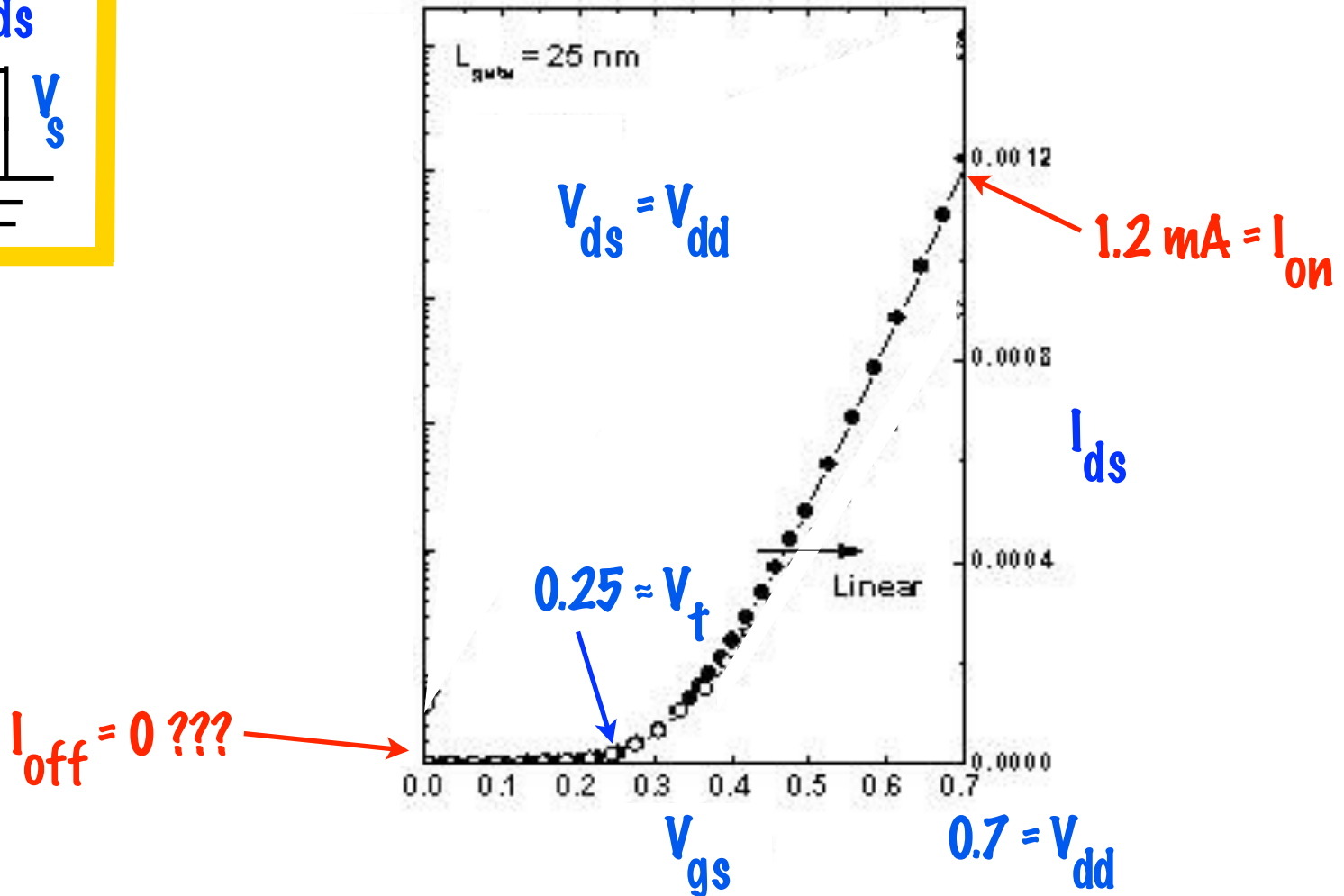


*Let's look beneath the abstraction:
origins of V_T and R_{on}*

MOSFET Threshold Voltage

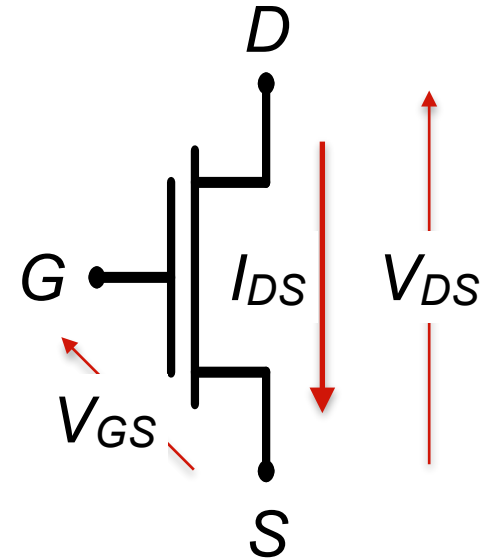
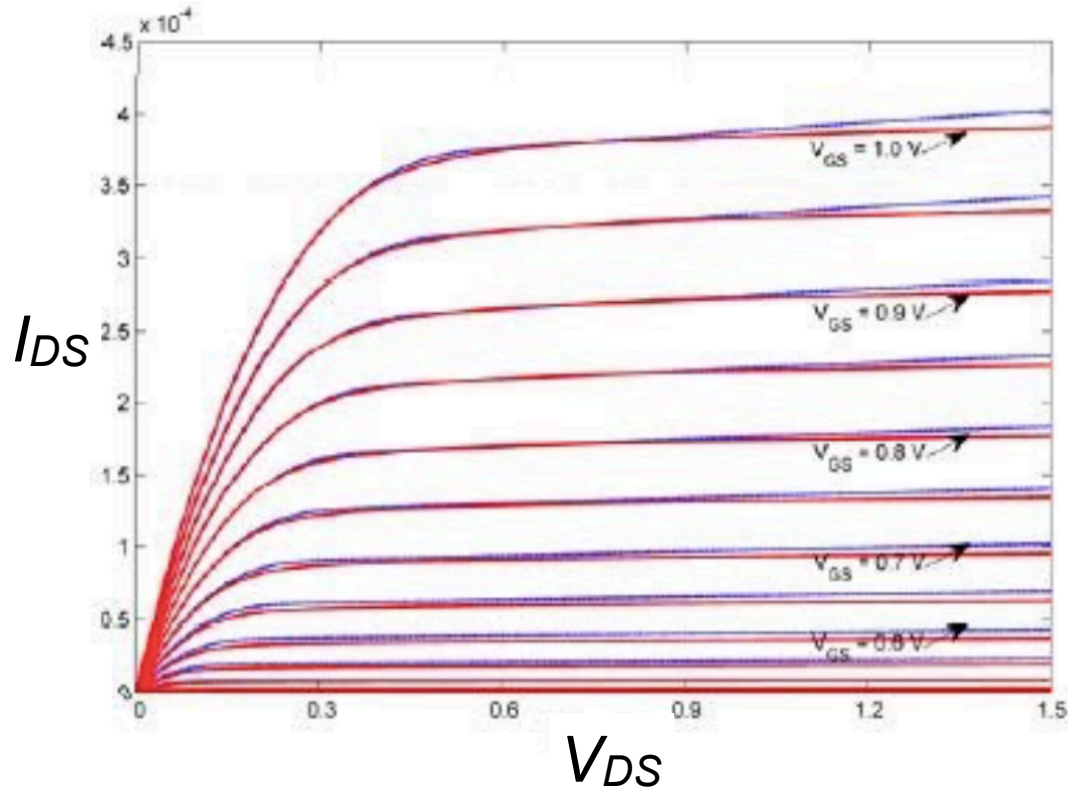


Transistor "turns on" when V_{gs} is $> V_t$.



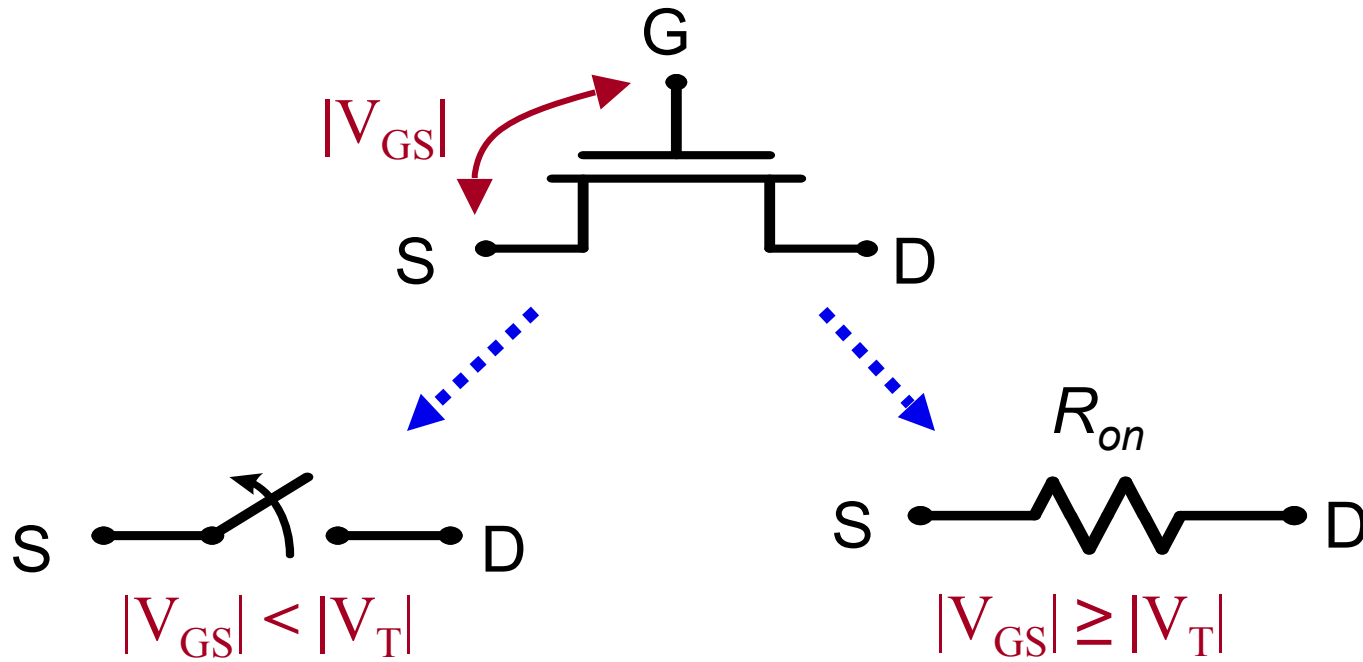
Transistor “resistance”

- Nonlinear I/V characteristic:



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

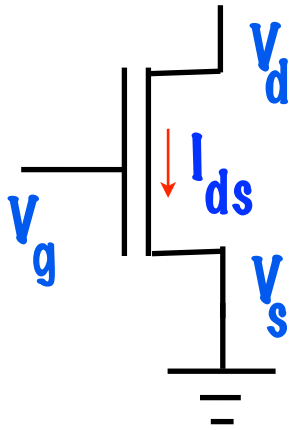
ON/OFF Switch Model of MOS Transistor



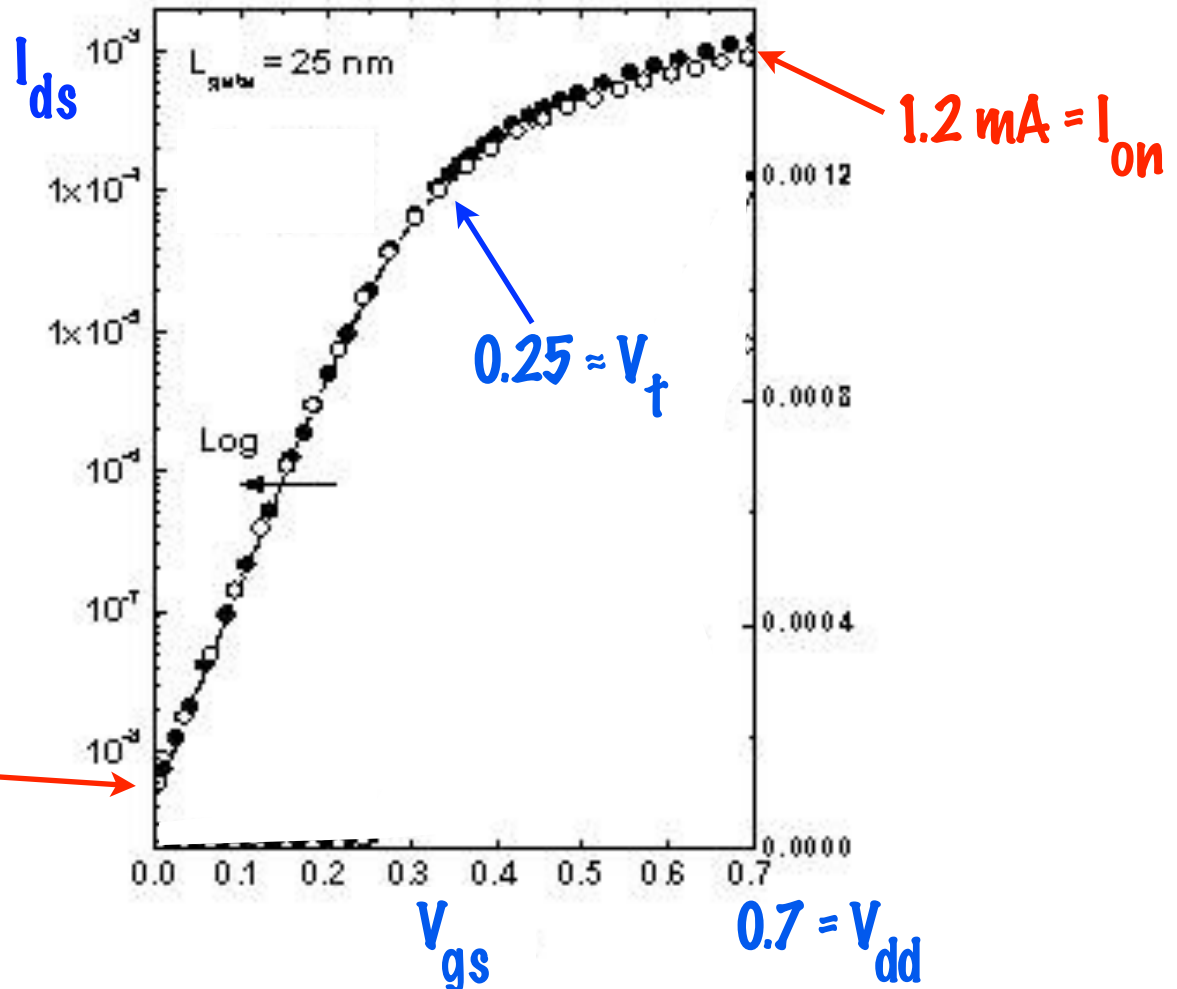
Plot on a "Log" Scale to See "Off" Current

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} .

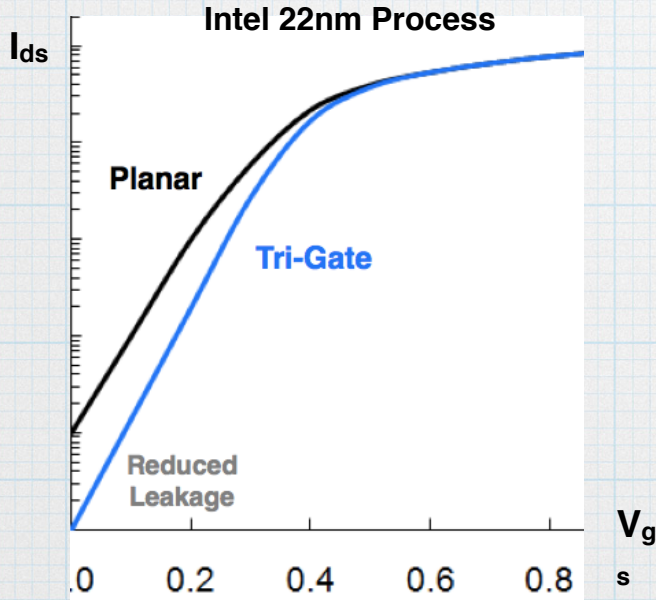
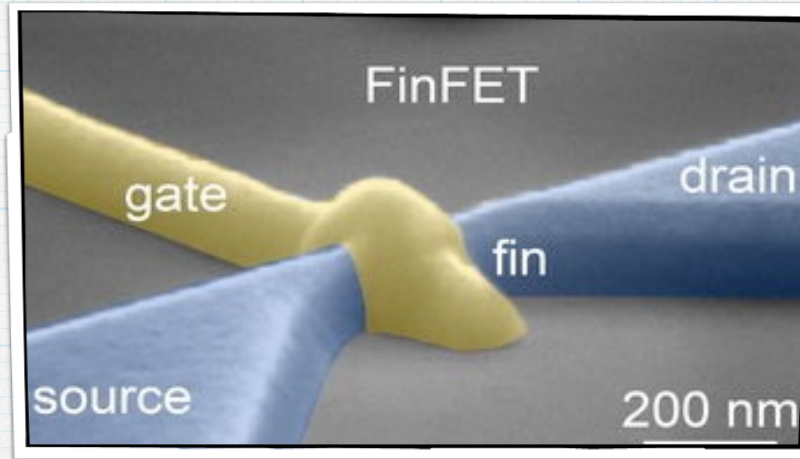


$$V_{ds} = V_{dd}$$

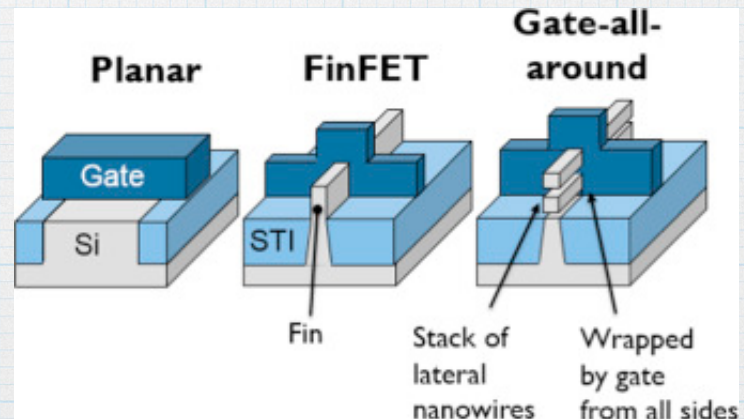


$$I_{off} \approx 10 \text{ nA}$$

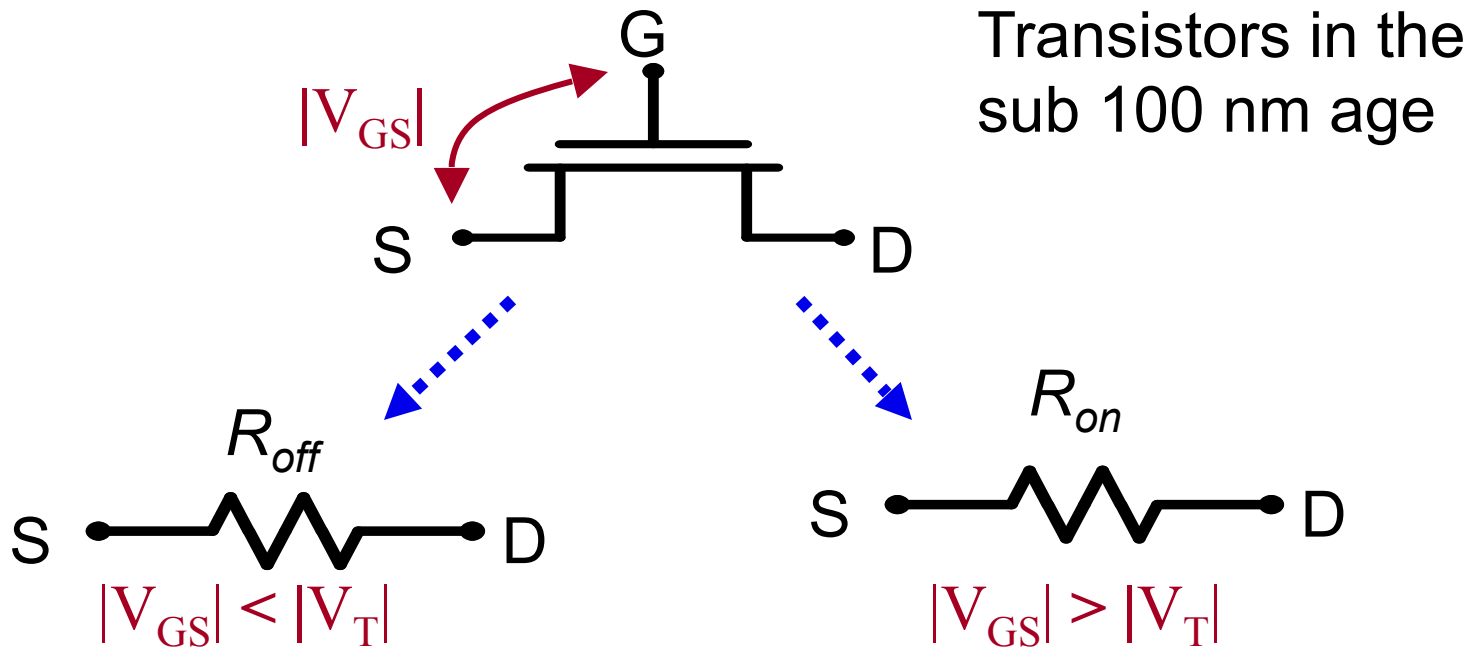
Latest Modern Process



Transistor channel is a raised fin.
Gate controls channel from sides and top.

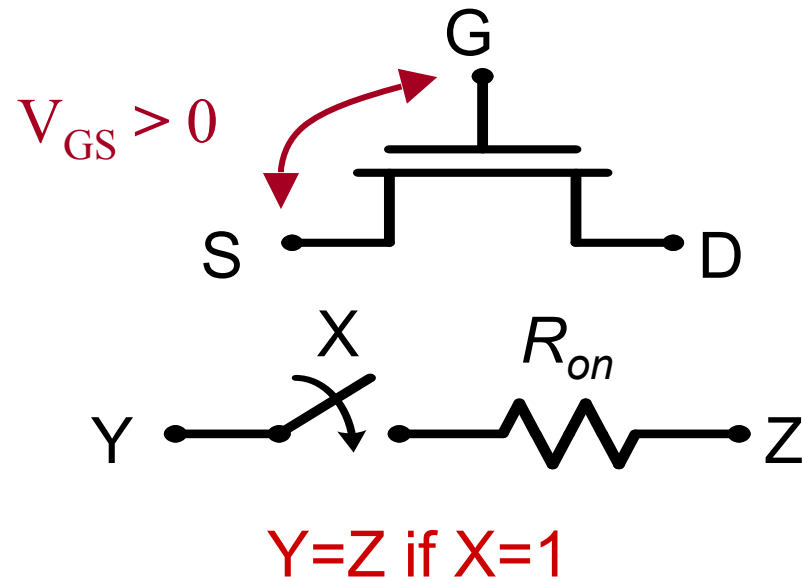


A More Realistic Switch

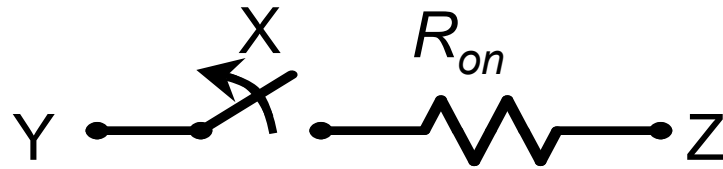


A Logic Perspective

NMOS Transistor

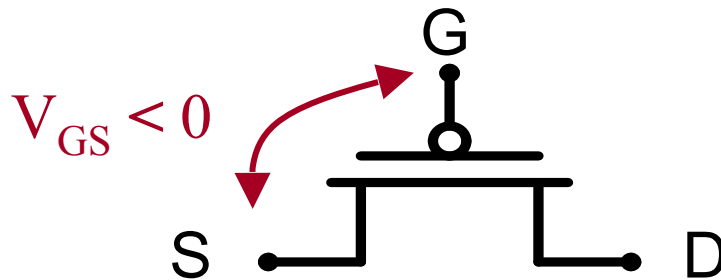


A Complementary Switch



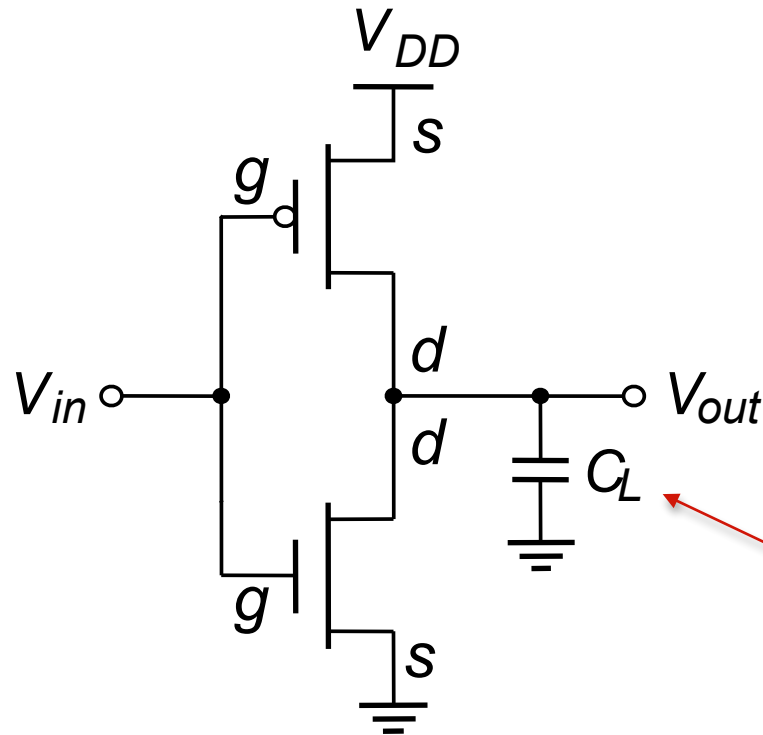
$Y=Z$ if $X=0$

PMOS Transistor



Remember, source is the node w/ the highest voltage.

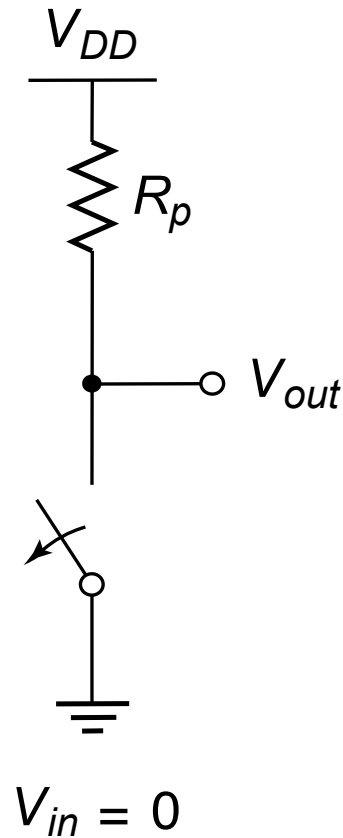
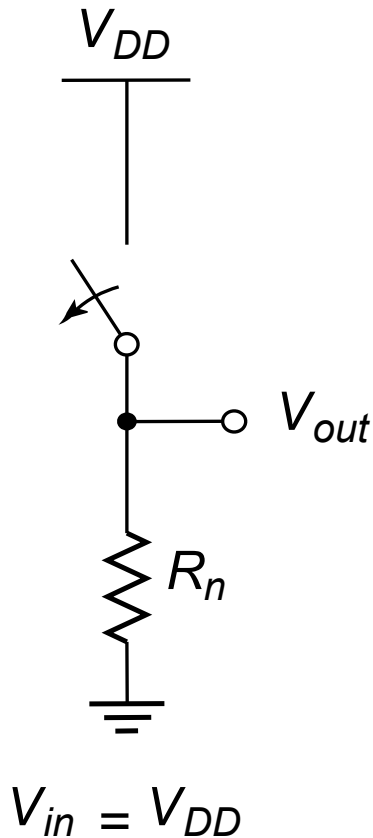
The CMOS Inverter: A First Glance



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*



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

*First-order means we will ignore Capacitance.

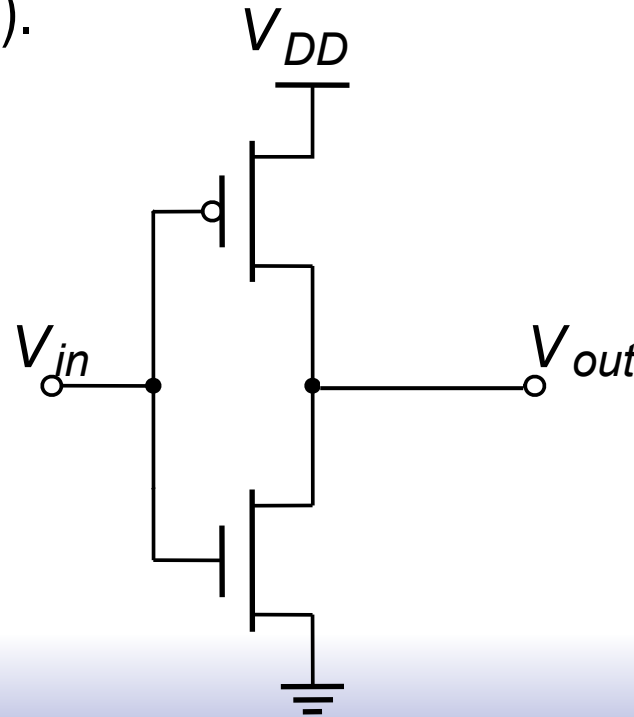


Switch logic

Static Logic Gate

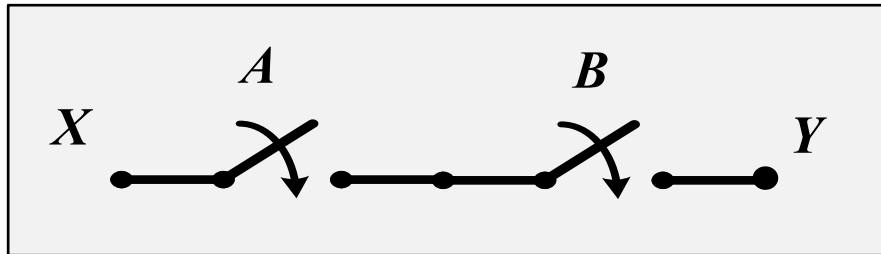
- At every point in time (except during the switching transients) each gate output is connected to either V_{DD} or V_{GND} via a low resistive path.
- The output of the gate **assumes at all times the value of the Boolean function** implemented by the circuit (ignoring, once again, the transient effects during switching periods).

Example: CMOS Inverter



Building logic from switches (NMOS)

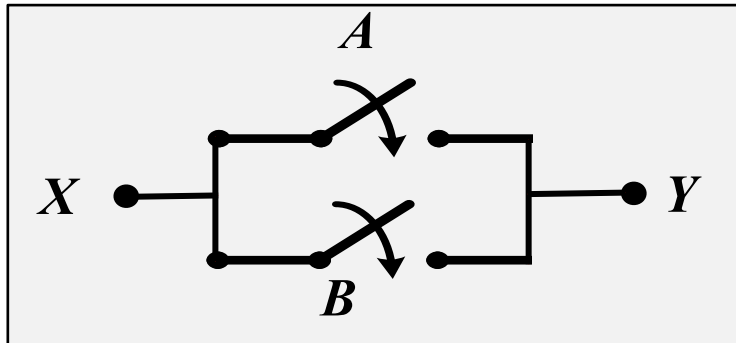
Series



AND

$Y = X$ if A AND B

Parallel



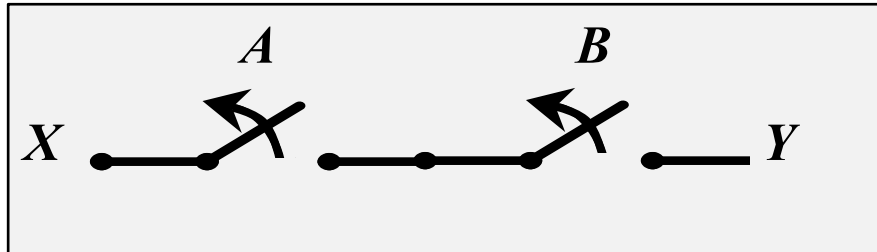
OR

$Y = X$ if A OR B

(output undefined if condition not true)

Logic using inverting switches (PMOS)

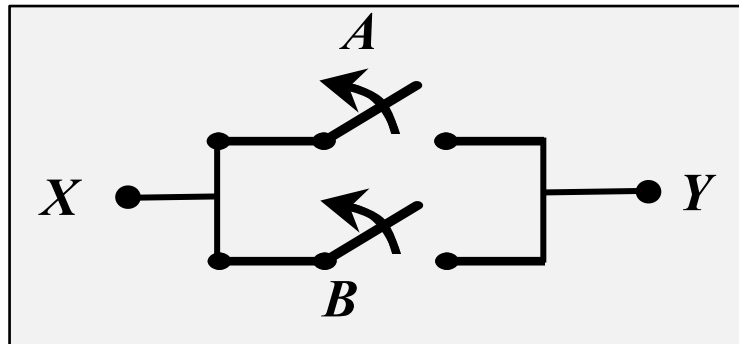
Series



NOR

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

Parallel



NAND

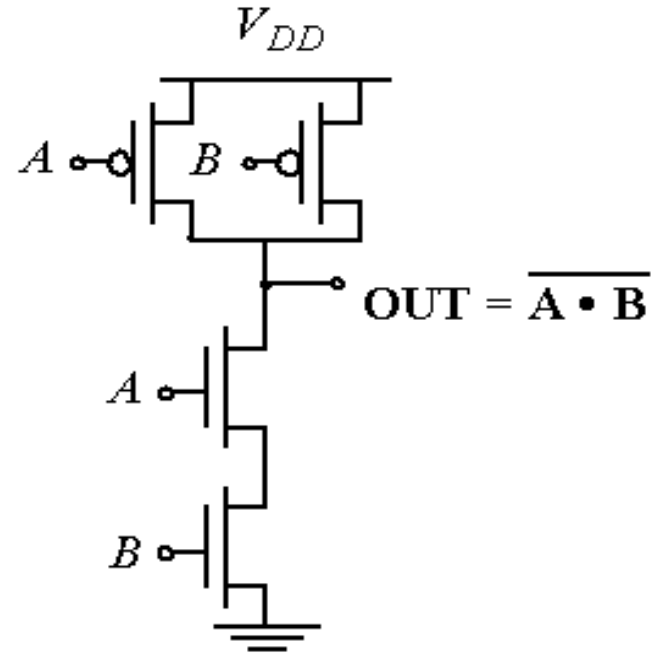
$$Y = X \text{ if } \bar{A} \text{ OR } \bar{B} \\ = \overline{AB}$$

(output undefined if condition not true)

Example Gate: NAND

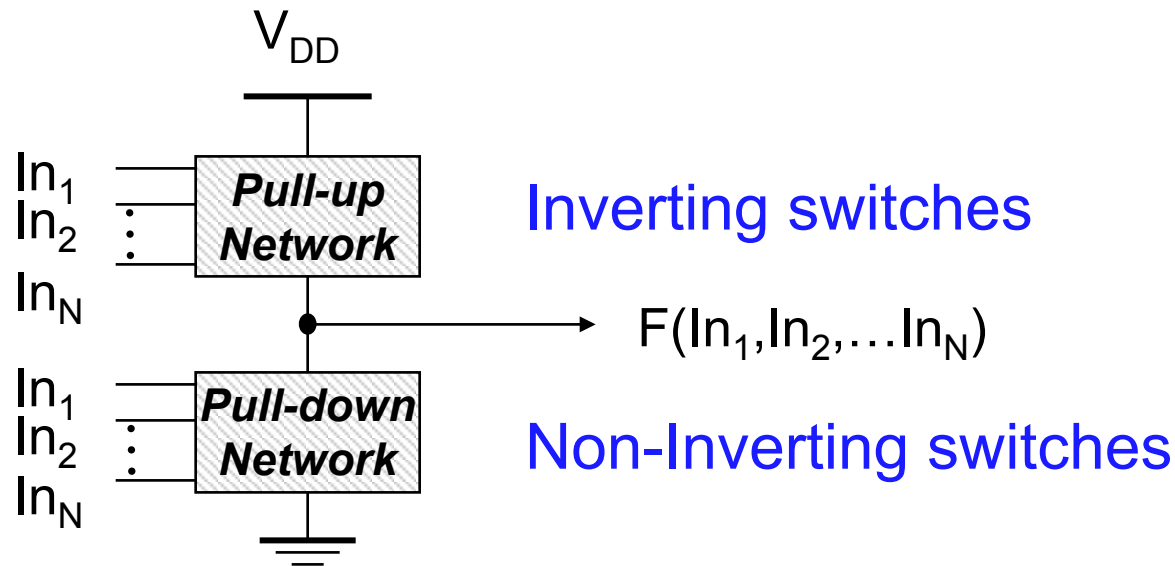
A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

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}

Static Complementary CMOS



PUN and PDN are dual logic networks:

series connections in the PUN are parallel connections in the PDN
parallel connections in the PUN are series connections in the PDN

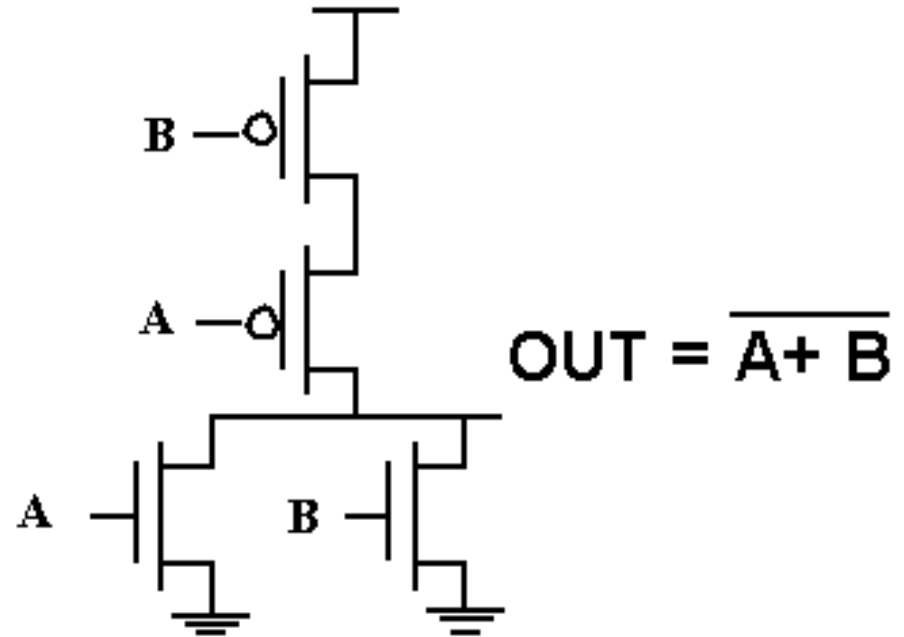
PUN and PDN functions are complements:

guarantees they are mutually exclusive, under all input values,
one or the other is conductive, but never both!

Example Gate: NOR

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

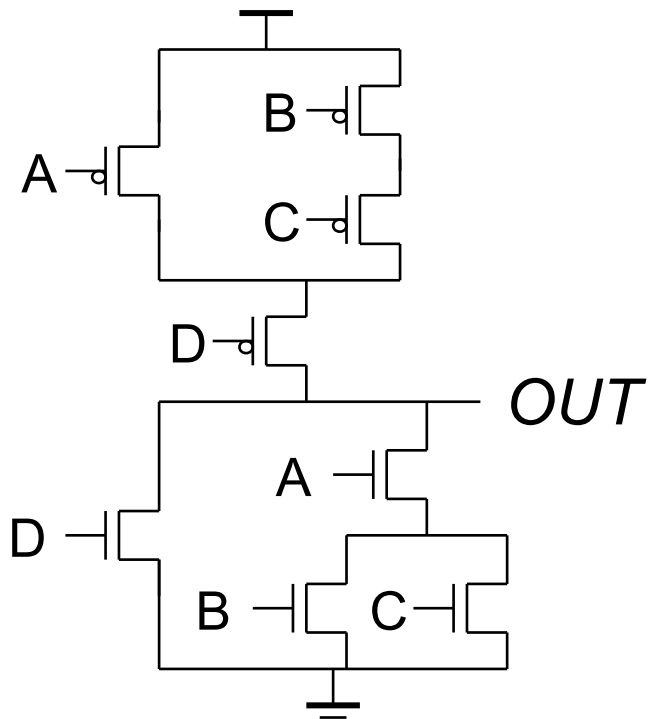
Truth Table of a 2 input NOR gate



Complex CMOS Gate

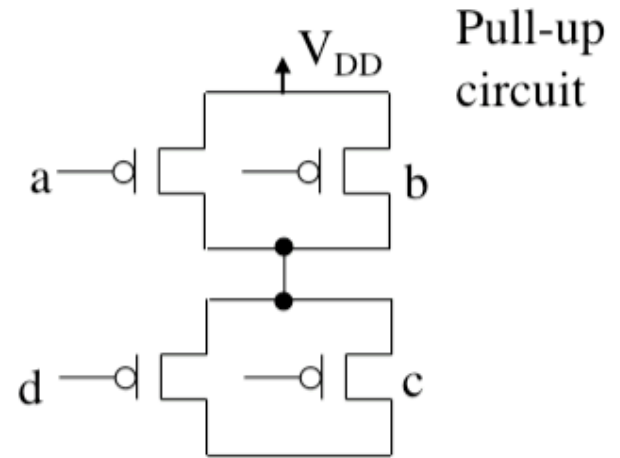
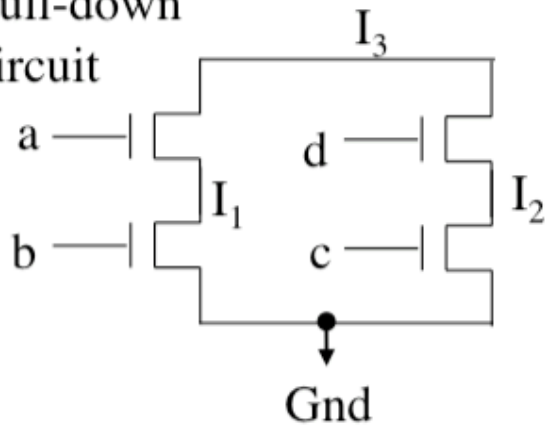
$$\overline{OUT} = D + A \cdot (B + C)$$

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

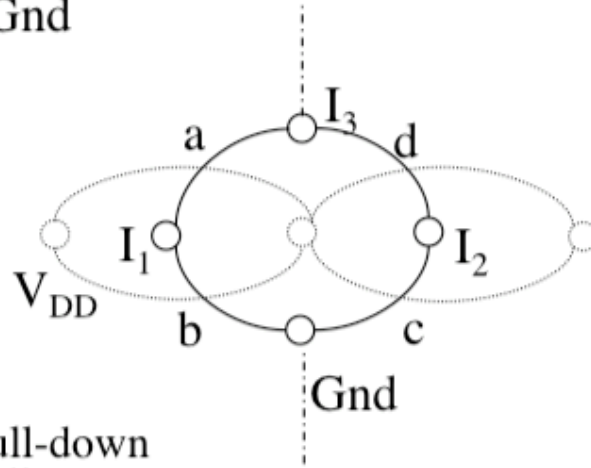


Graph Models for Duals

Pull-down circuit

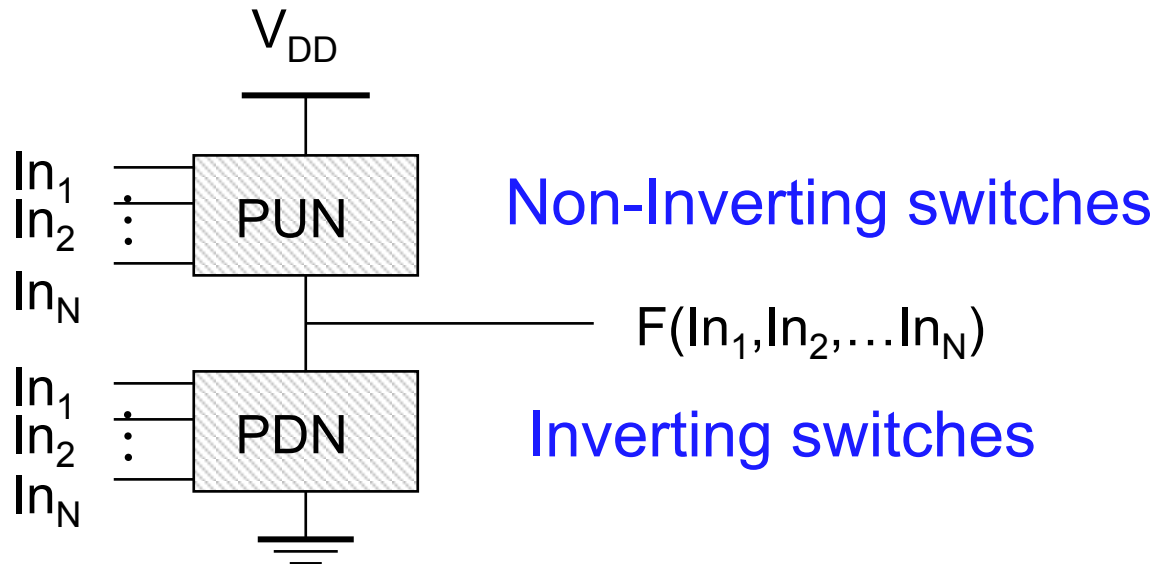


Pull-up circuit



— Pull-down
 Pull-up

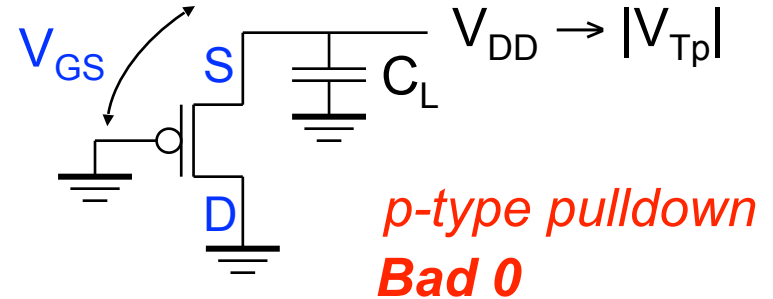
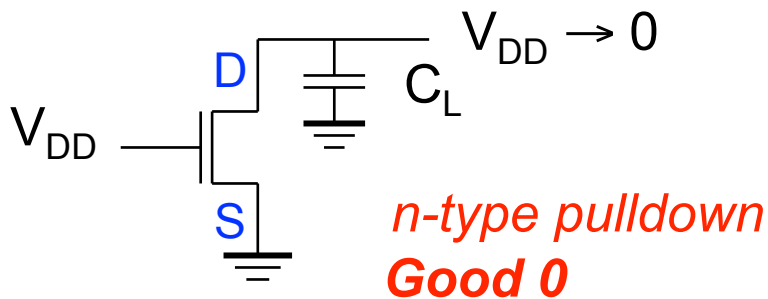
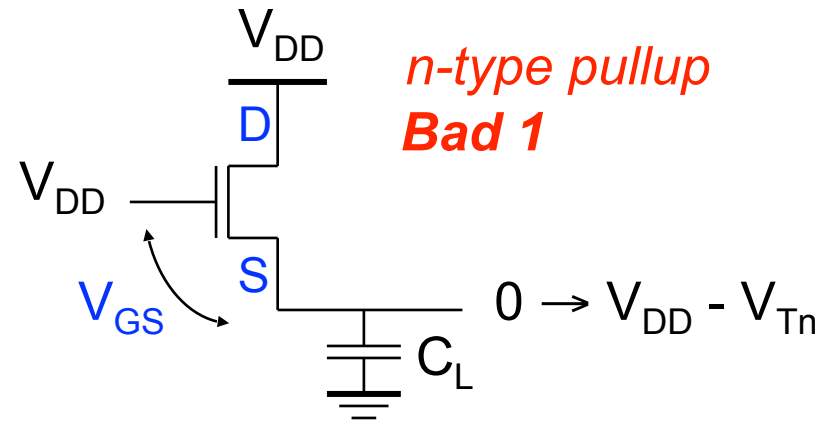
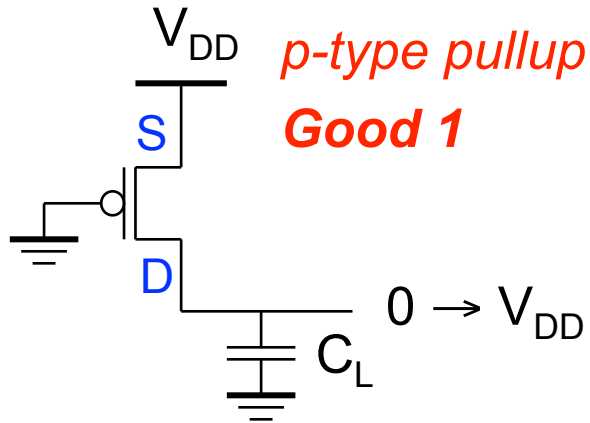
Non-inverting logic



*Why is this
a bad idea?*

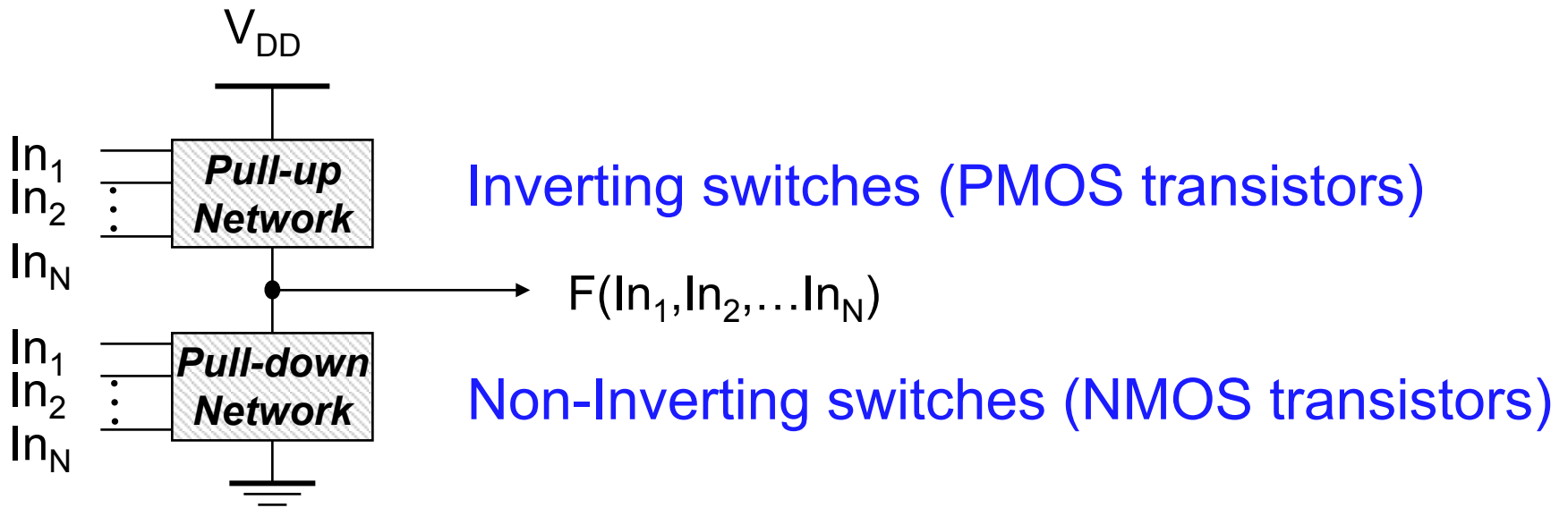
PUN and PDN are **dual** logic networks
PUN and PDN functions are **complementary**

Switch Limitations

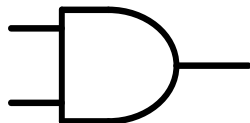


Tough luck ...

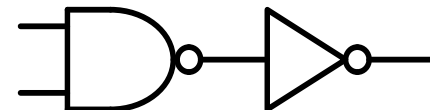
“Static” CMOS gates



- ❑ Static CMOS gates are always inverting

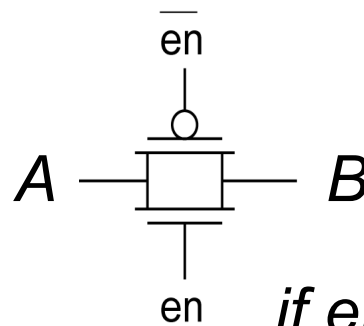


AND = NAND + INV



Transmission Gate

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



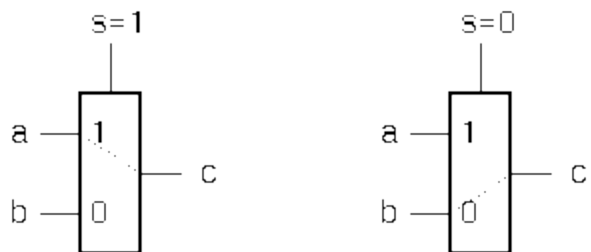
if $en == 1$ then A connects to B

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

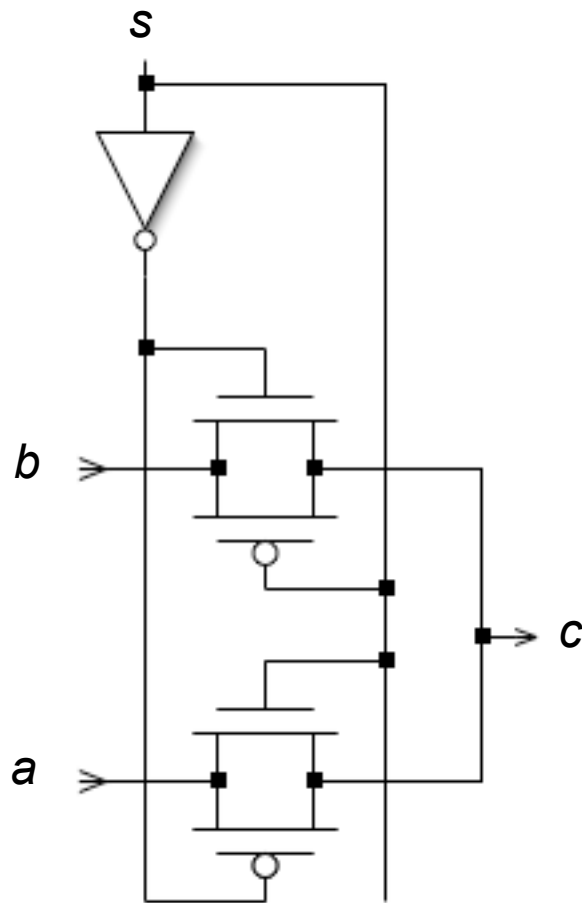
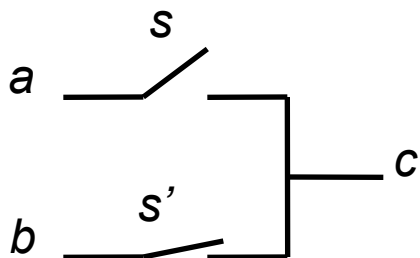
Transmission-gate Multiplexor

2-to-1 multiplexor:

$$c = sa + s'b$$



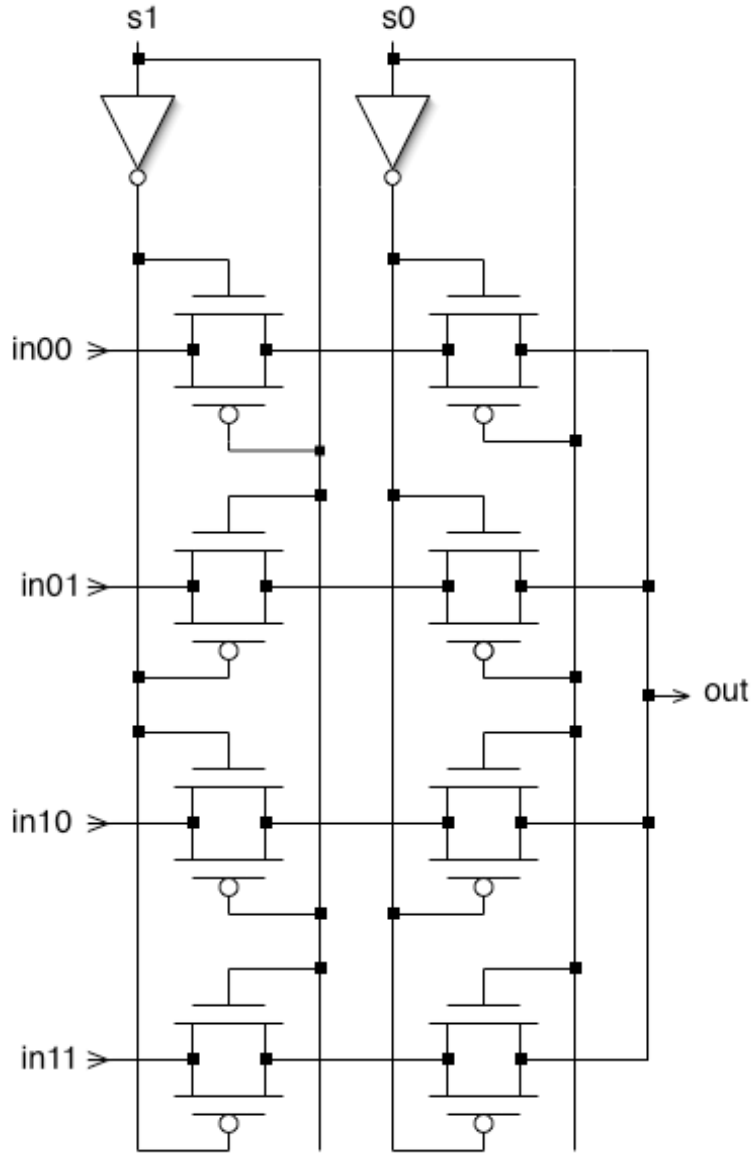
Switches simplify the implementation:



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

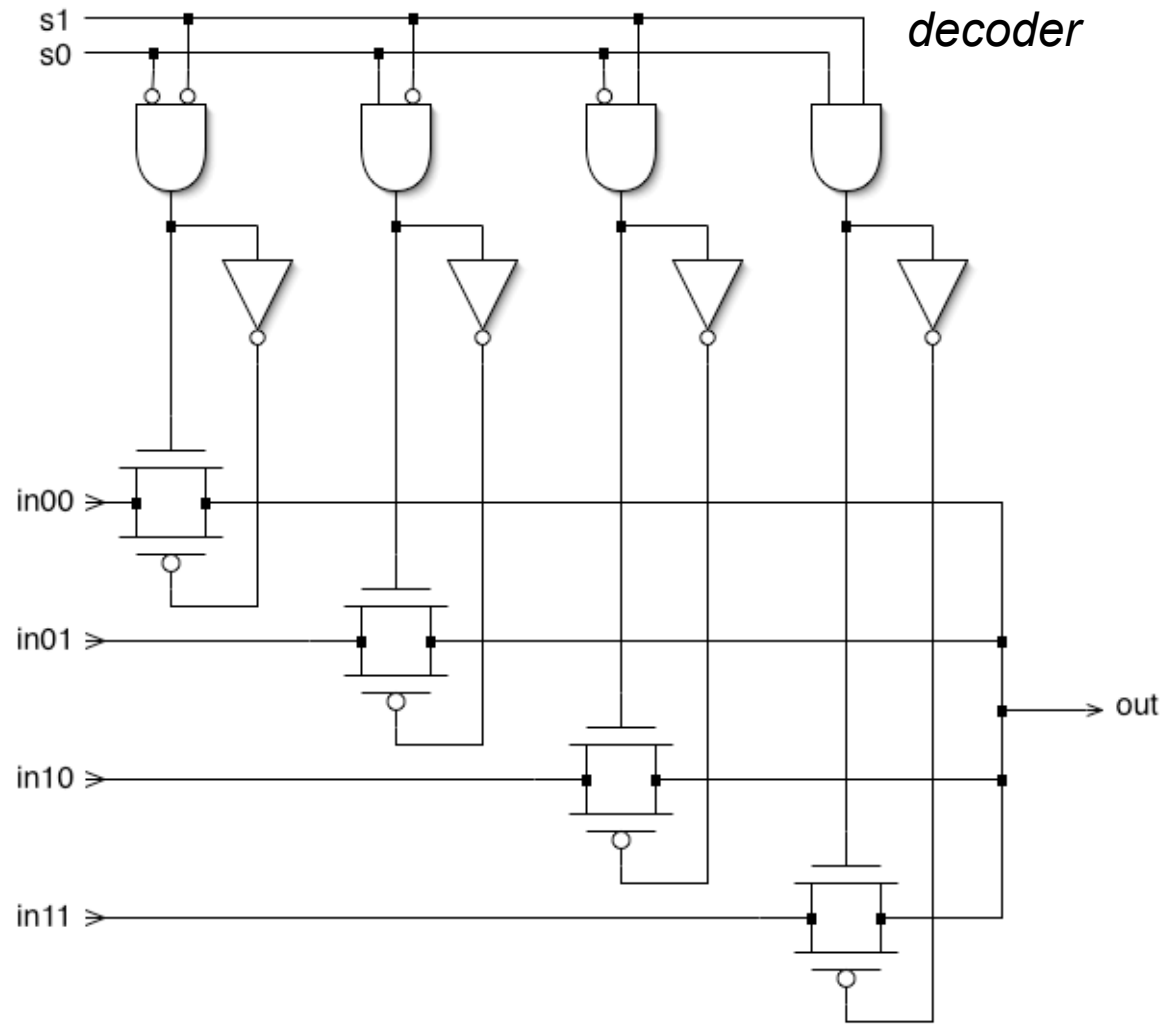


- The series connection of pass-transistors in each branch effectively forms the AND of s_1 and s_0 (or their complement).
- Compare cost to logic gate implementation

Any alternate solutions?

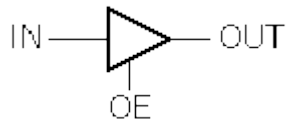
Alternative 4-to-1 Multiplexor

- ❑ 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

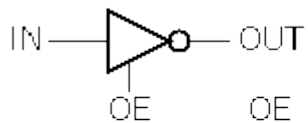
Tri-state Buffer:



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

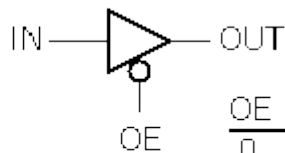
“high impedance” (output disconnected)

Variations:



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

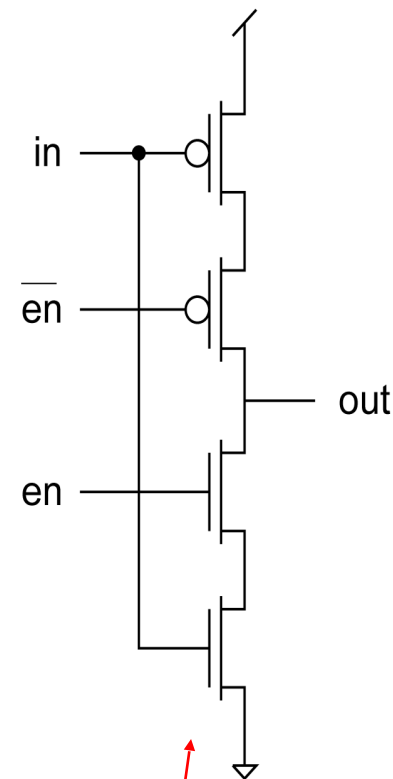
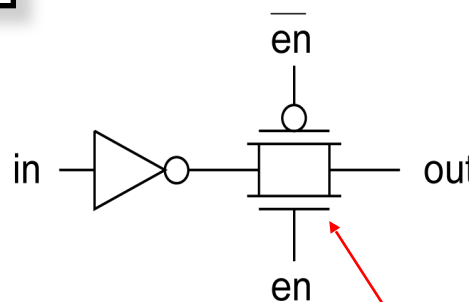
Inverting buffer



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

Inverted enable

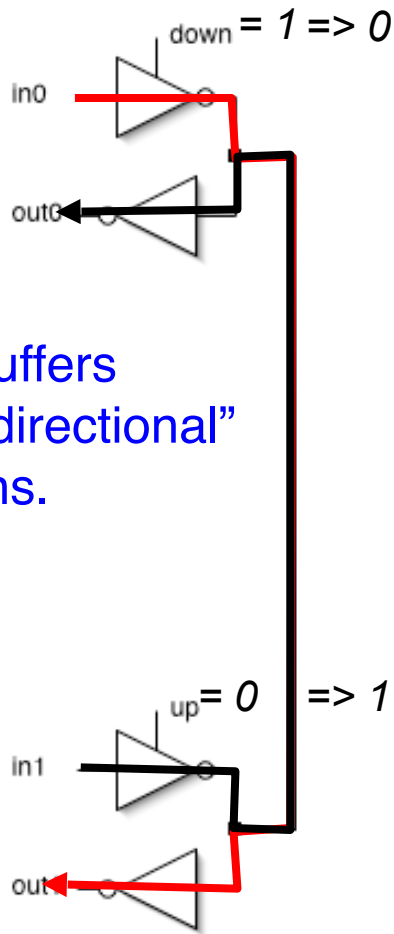
CMOS Implementation



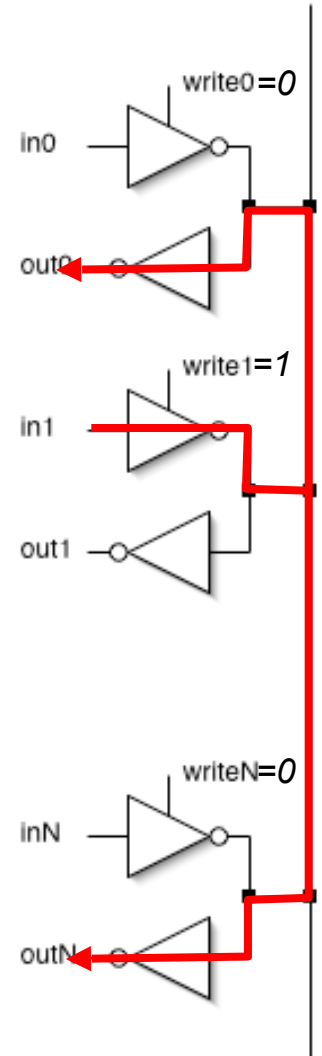
transmission gate provide the isolation:
usually designed this way

Tri-state Buffers

Tri-state buffers enable “bidirectional” connections.

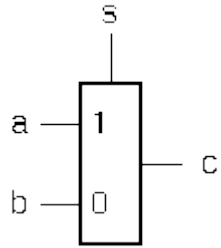


Tri-state buffers are used when multiple circuits all connect to a common node or wire. Only one circuit at a time is allowed to drive the bus. All others “disconnect” their outputs, but can “listen”.

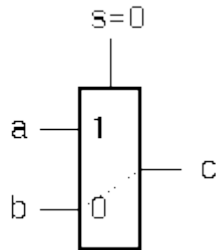
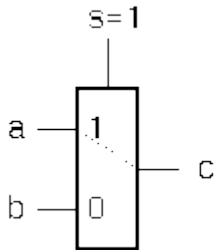


Tri-state Based Multiplexor

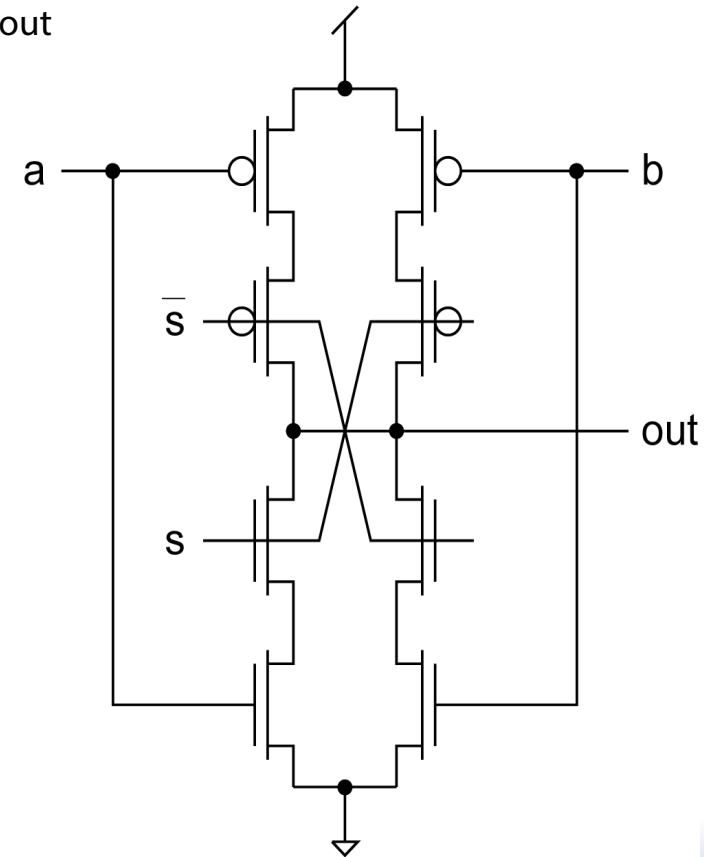
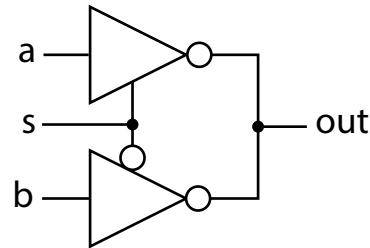
Multiplexor:



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

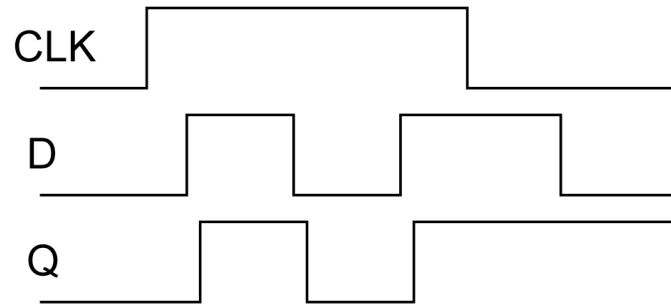
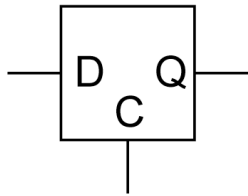


Transistor Circuit for inverting-multiplexor:

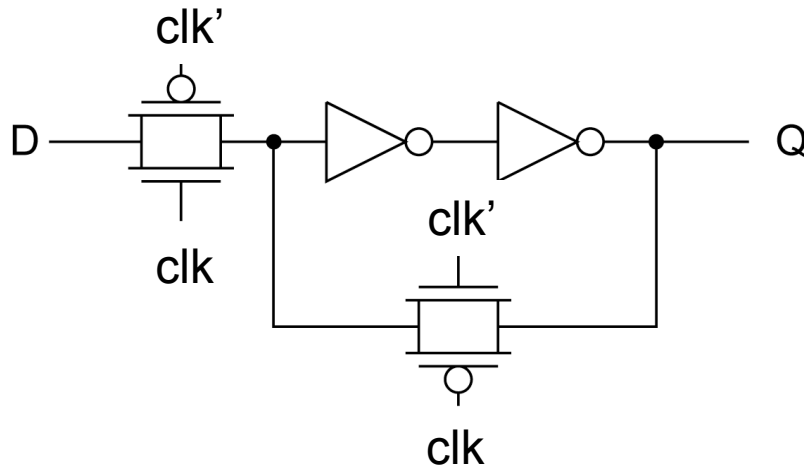


Latches and Flip-flops

Positive Level-sensitive *latch*:

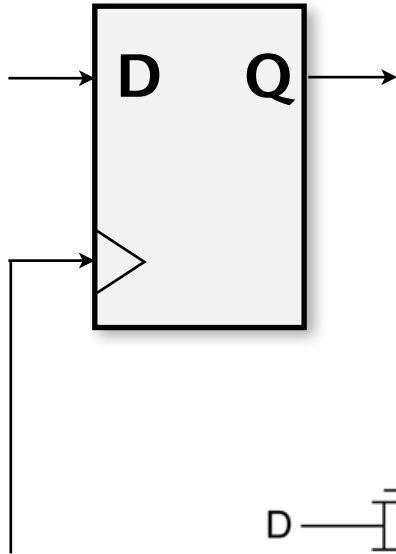


Latch Implementation:

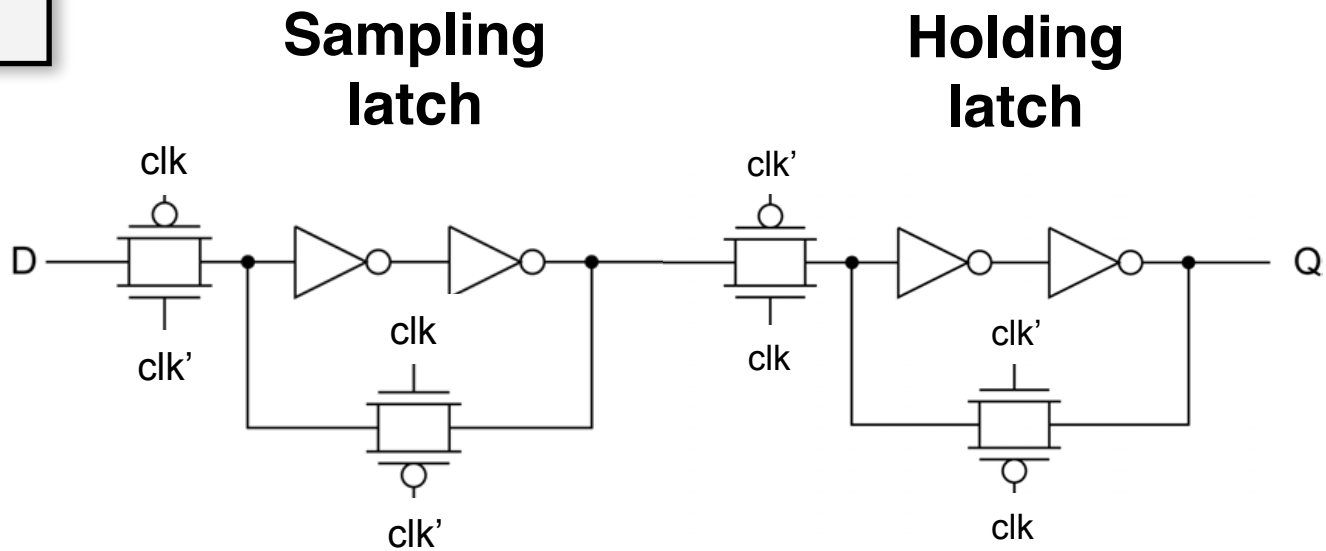


When the clock is high, the latch is “transparent”

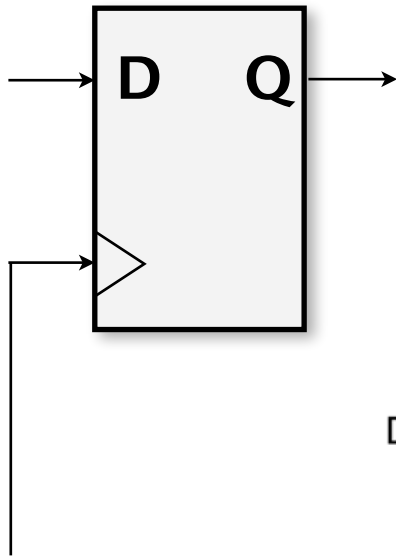
Positive edge-triggered flip-flop



A flip-flop “samples” right before the edge, and then “holds” value.

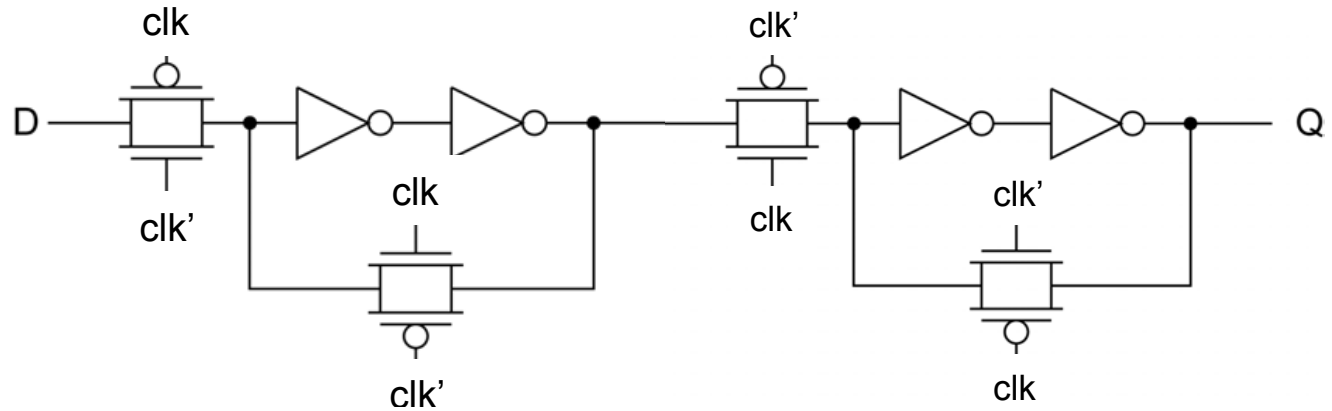


Sensing: When clock is low

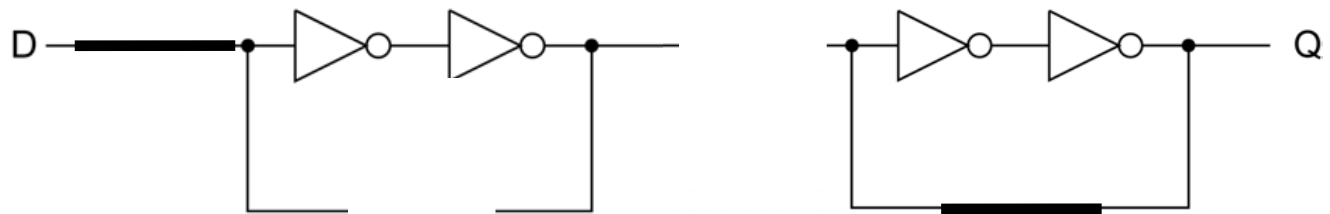


A flip-flop “samples” just before the edge, and then “holds” value.

Sampling latch



$clk = 0$
 $clk' = 1$

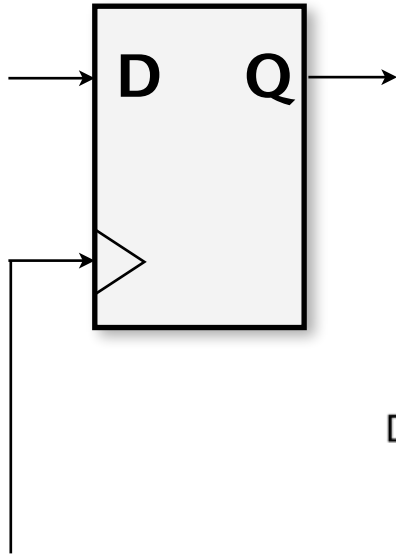


Ready to capture new value on clock edge

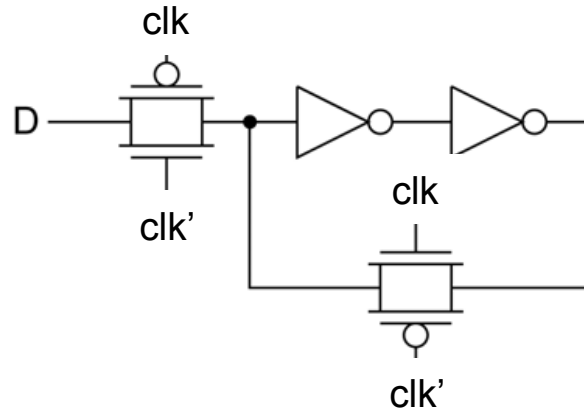
Is outputting last previous captured value.

Capture: When clock goes high

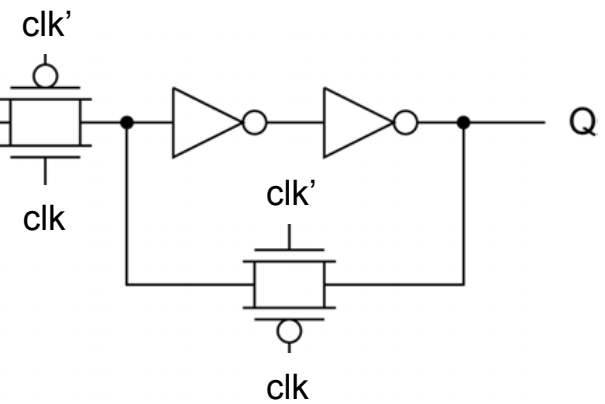
A flip-flop “samples” right before the edge, and then “holds” value.



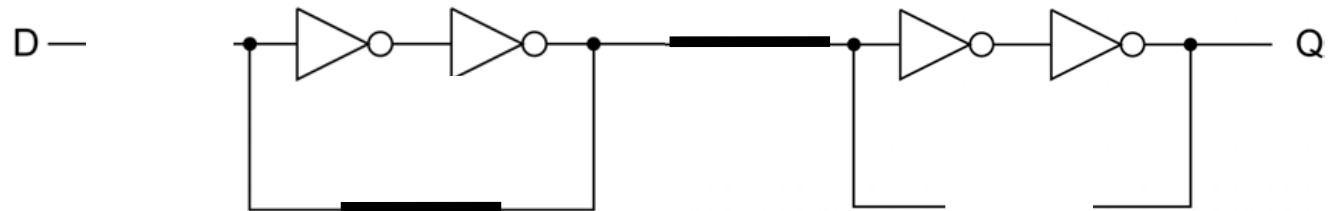
Sampling latch



Holding latch



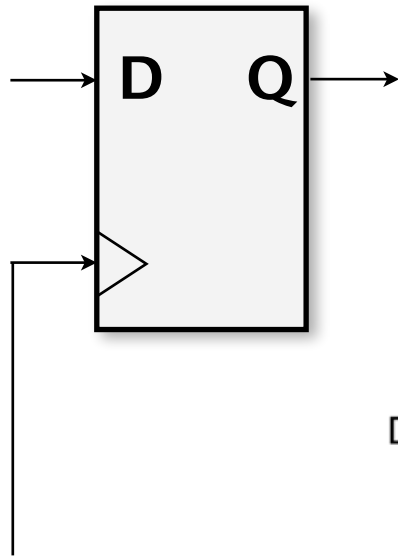
$clk = 1$
 $clk' = 0$



Remembers value just captured.

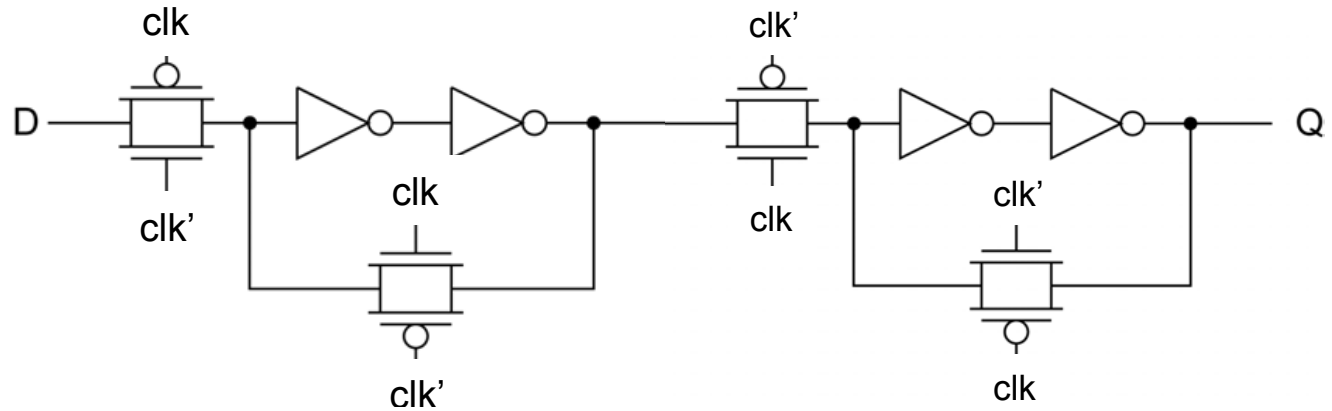
Outputs value just captured.

Return to sensing: When clock is low

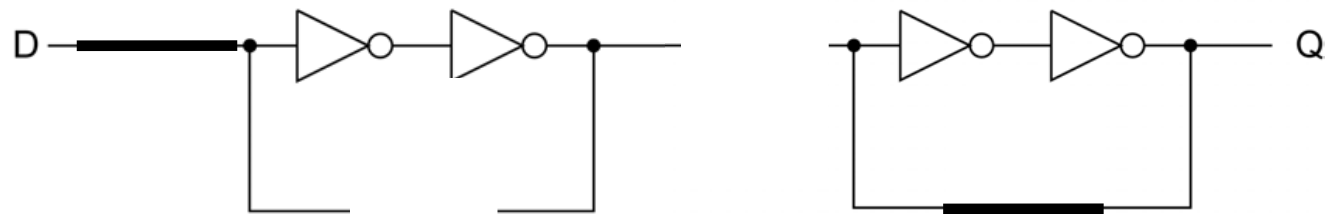


A flip-flop “samples” just before the edge, and then “holds” value.

Sampling latch



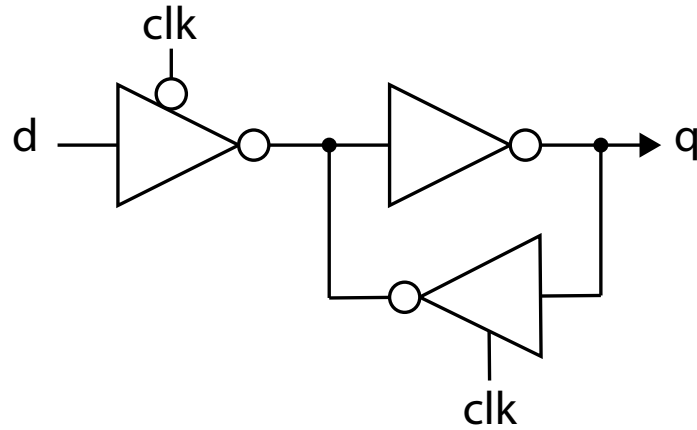
$clk = 0$
 $clk' = 1$



Ready to capture new value on clock edge

Continue to output captured value.

Tri-state-Inverter Latch



*Negative
Level-sensitive
latch:*

- ❑ Commonly used in standard cell flip-flops.
- ❑ More transistors than pass-transistor version, but more robust.
- ❑ Lays out well with modern layout rules.