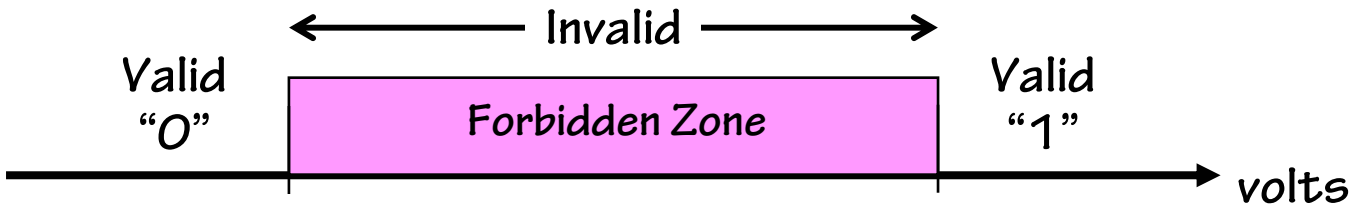
Using Voltages "Digitally"

- Key idea: don't allow "0" to be mistaken for a "1" or vice versa
- Use the same "uniform representation convention", for *every* component and wire in our digital system
- To implement devices with high reliability, we outlaw "close calls" via a representation convention which forbids a range of voltages between "0" and "1".



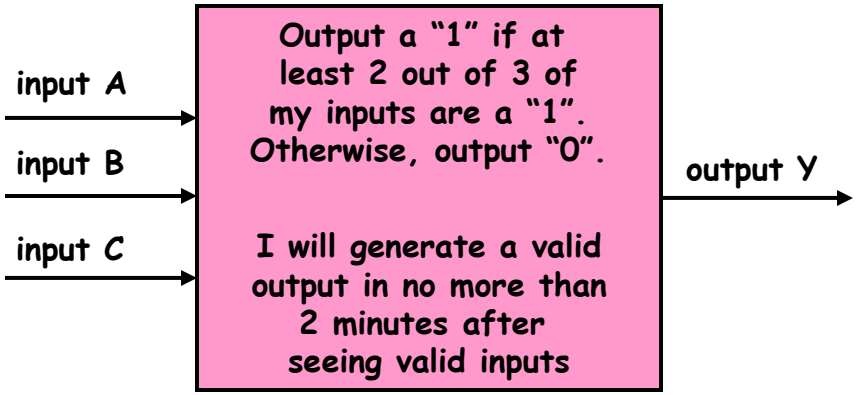
CONSEQUENCE:

Notion of "VALID" and "INVALID" logic levels

A Digital Processing Element

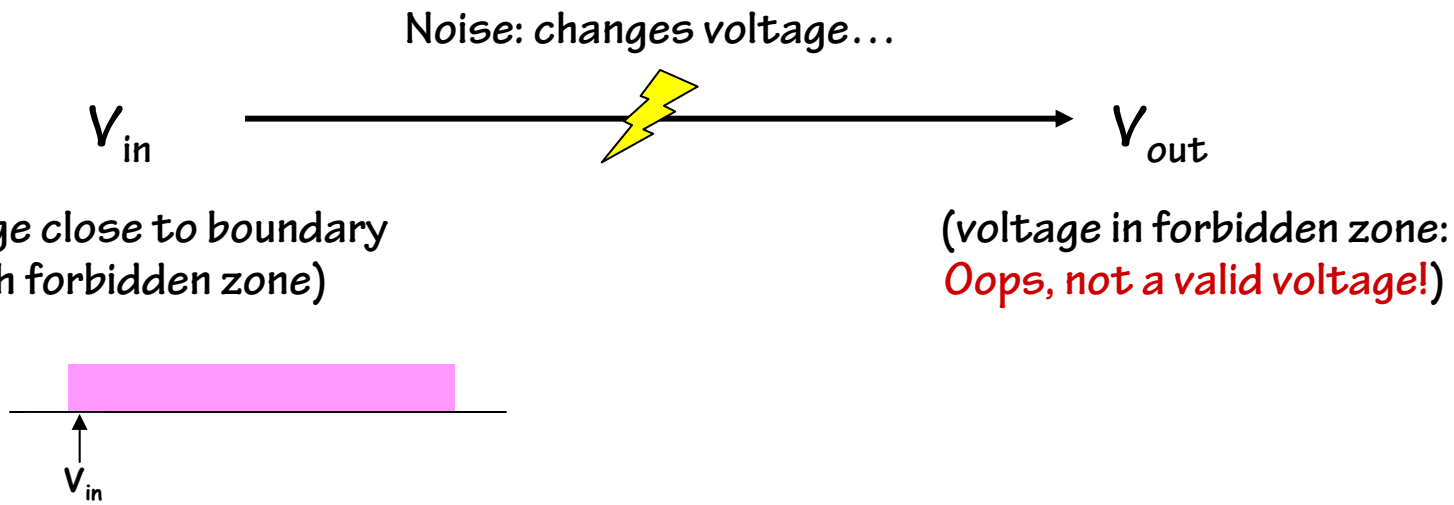
- A *combinational device* is a circuit element that has
  - one or more digital *inputs*
  - one or more digital *outputs*
  - a *functional specification* that details the value of each output for every possible combination of valid input values
  - a *timing specification* consisting (at minimum) of an upper bound t_{pd} on the required time for the device to compute the specified output values from an arbitrary set of stable, valid input values

Static discipline



Wires: theory vs. practice

Does a wire obey the static discipline?



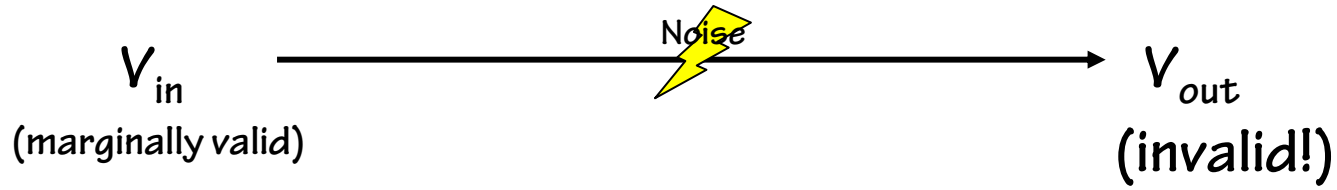
Questions to ask ourselves:

In digital systems, where does noise come from?

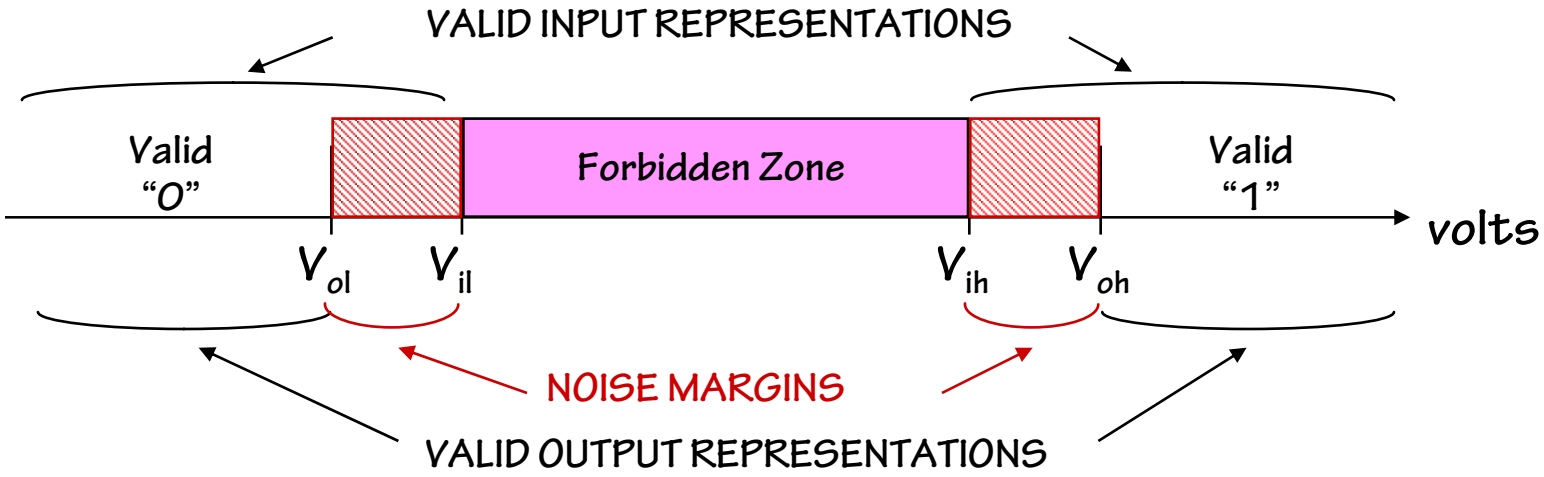
How big an effect are we talking about?

Needed: Noise Margins!

Does a wire obey the static discipline?

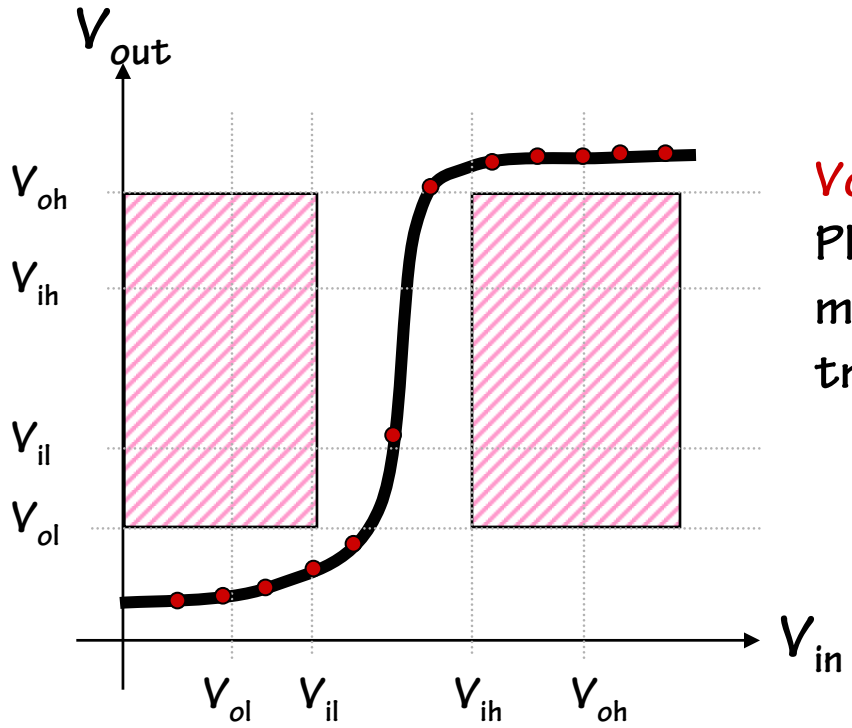


No! A combinational device must restore marginally valid signals. It must accept marginal inputs and provide unquestionable outputs (i.e., to leave room for noise).



A Buffer

A simple BUFFER:



Voltage Transfer Characteristic (VTC):
Plot of V_{out} vs. V_{in} where each measurement is taken after any transients have died out.

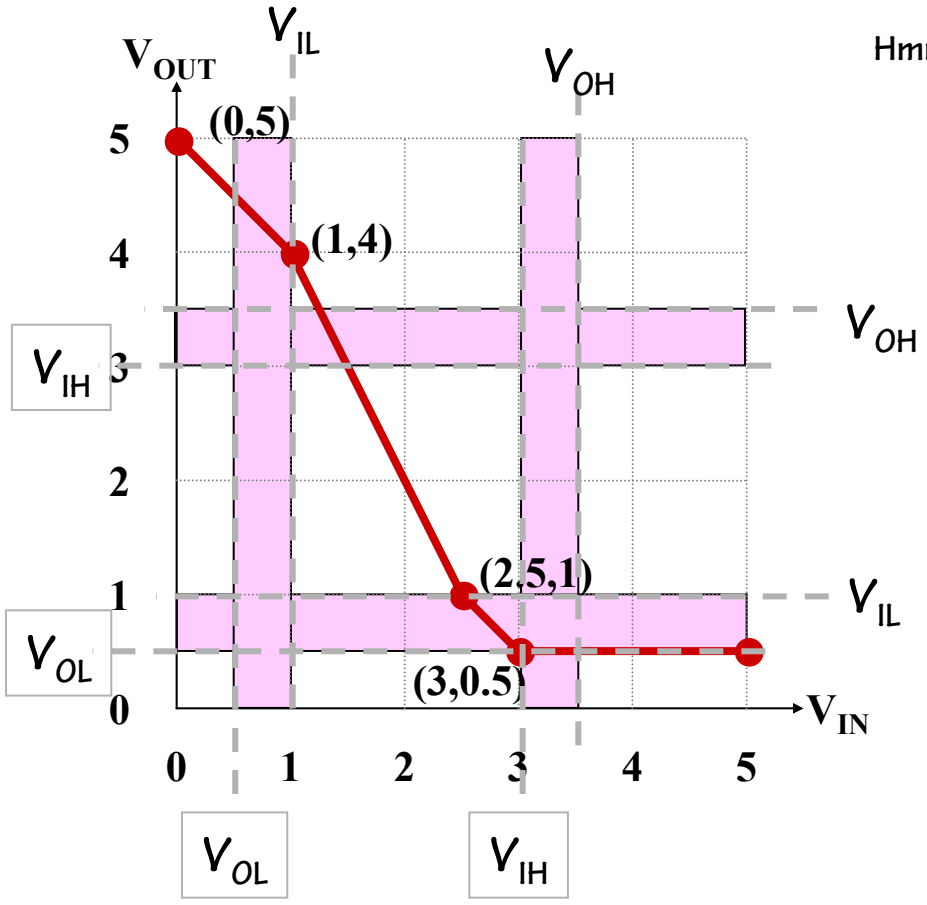
Note: VTC does not tell you anything about how fast a device is— it measures static behavior not dynamic behavior

Static Discipline requires that we avoid the shaded regions (aka "forbidden zones"), which correspond to *valid* inputs but *invalid* outputs. Net result:

combinational devices must have GAIN > 1 and be NONLINEAR.

Can this be a combinational device?

Suppose that you measured the voltage transfer curve of the device shown below. Could we build a logic family using it as a single-input combinational device?



Hmmm, it had better be an *INVERTER*…

The device must be able to actually produce the desired output level. Thus, V_{OL} can be no lower than 0.5 V.

Try V_{OL} = 0.5 V

V_{IH} must be high enough to produce V_{OL}

Try V_{IH} = 3 V

Now, choose noise margins – find an N and set

$V_{OH} = V_{IH} + N$

$V_{IL} = V_{OL} + N$

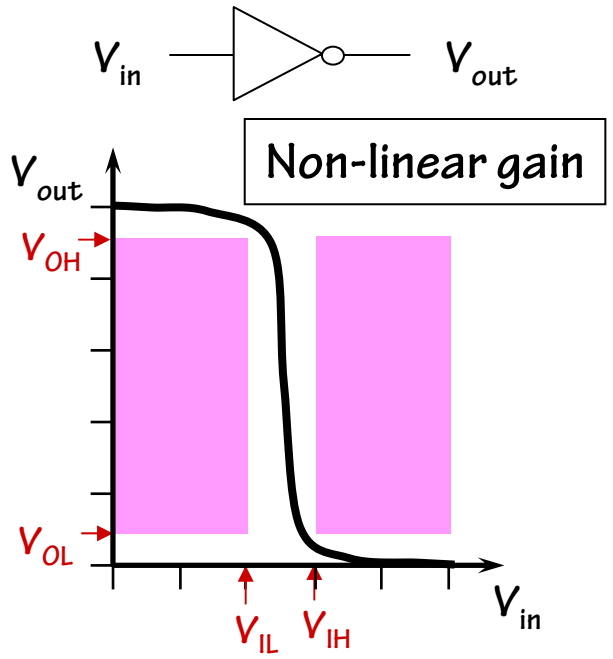
Such that

V_{IH} IN generates V_{OL} or less out; AND

V_{IL} IN generates V_{OH} or more out.

Try N = 0.5 V

Building Bits from Atoms



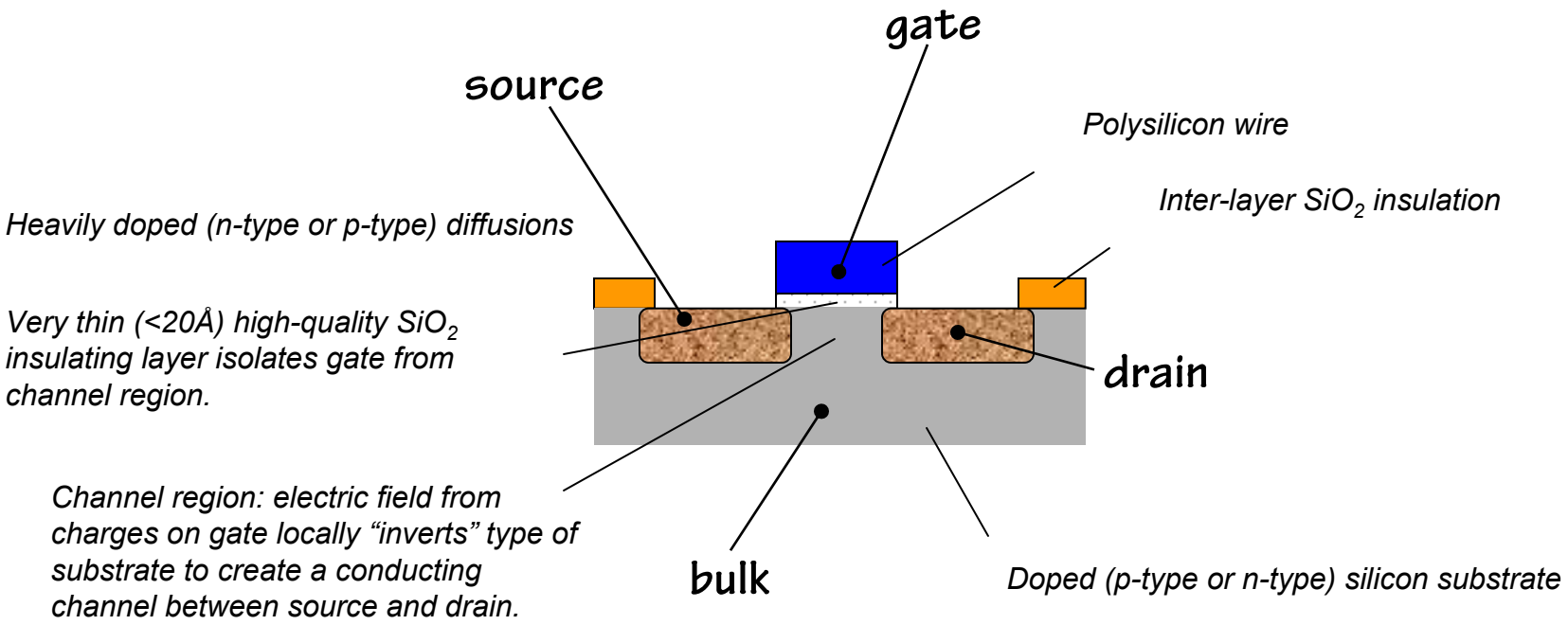
We Need Three Things:

1. Represent and communicate bits
2. Transform bits (Invert, AND, OR,…)
3. Remember bits (storage)

…subject to the fundamentals of physics:

Uncertainty, Noise, c, Thermodynamics,…

MOSFETS: Gain & non-linearity

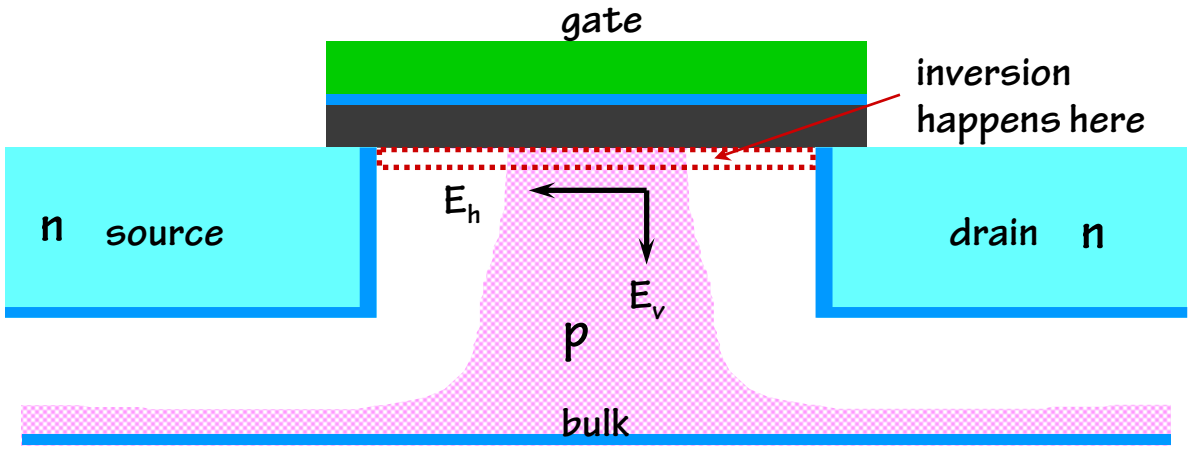


MOSFETs (metal-oxide-semiconductor field-effect transistors) are four-terminal voltage-controlled switches. Current flows between the diffusion terminals if the voltage on the gate terminal is large enough to create a conducting "channel", otherwise the mosfet is off and the diffusion terminals are not connected.

Why are MOS devices King?

FETs as switches

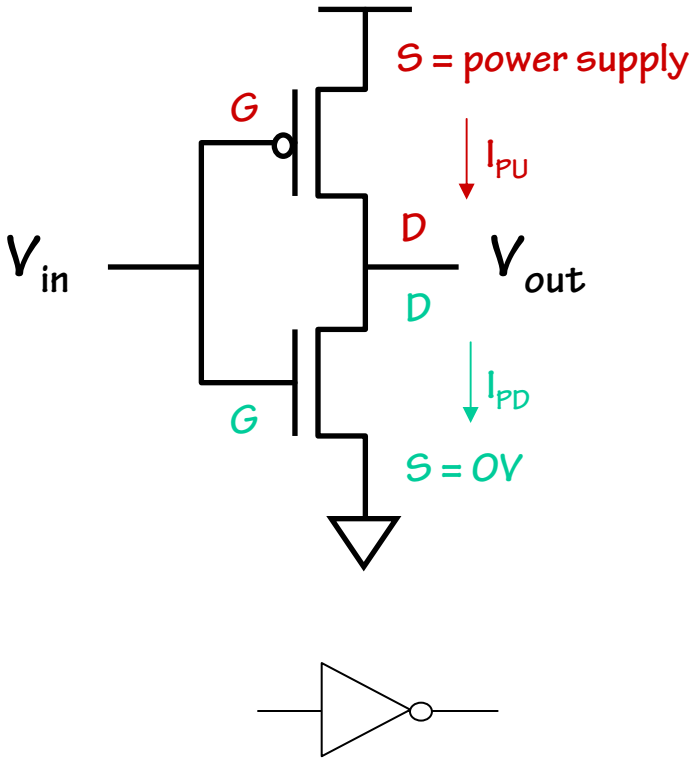
The four terminals of a Field Effect Transistor (gate, source, drain and bulk) connect to conducting surfaces that generate a complicated set of electric fields in the channel region which depend on the relative voltages of each terminal.

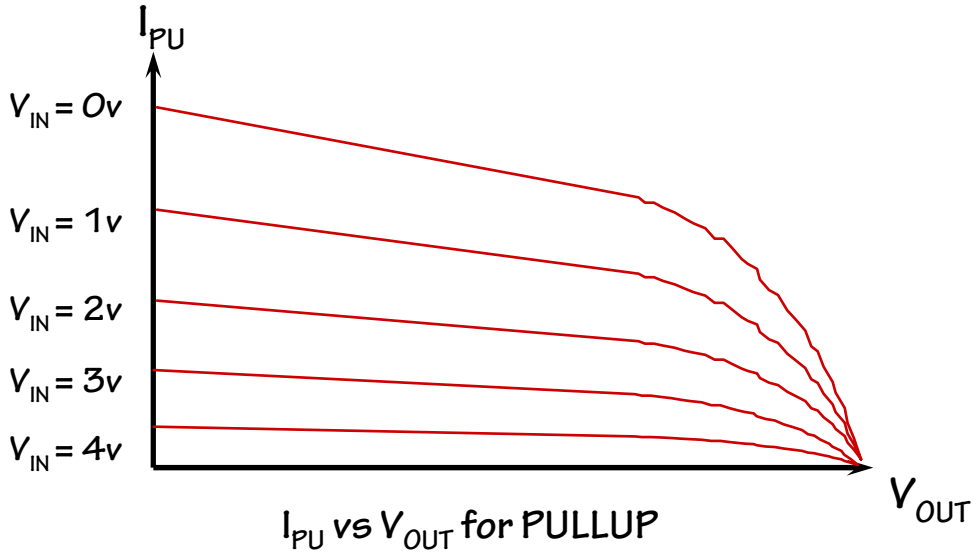


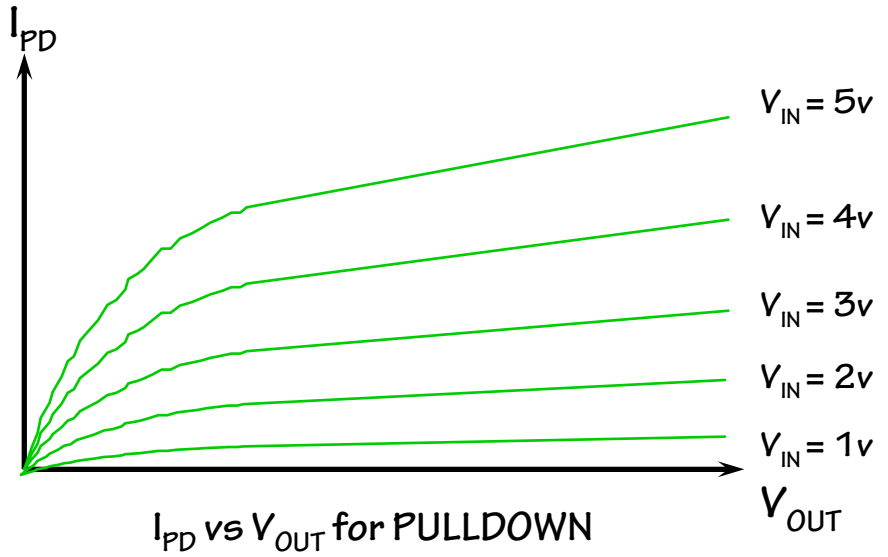
INVERSION:
A sufficiently strong vertical field will attract enough electrons to the surface to create a conducting n-type channel between the source and drain.

CONDUCTION:
If a channel exists, a horizontal field will cause a drift current from the drain to the source.

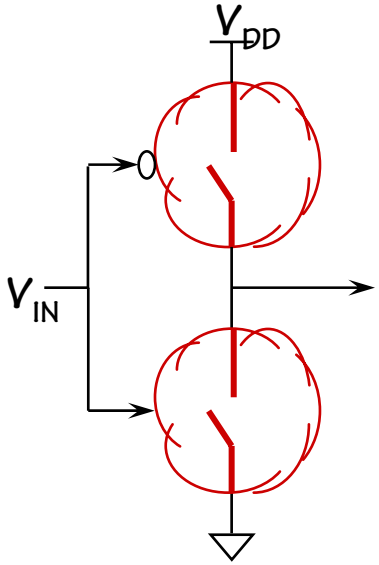
CMOS Inverter





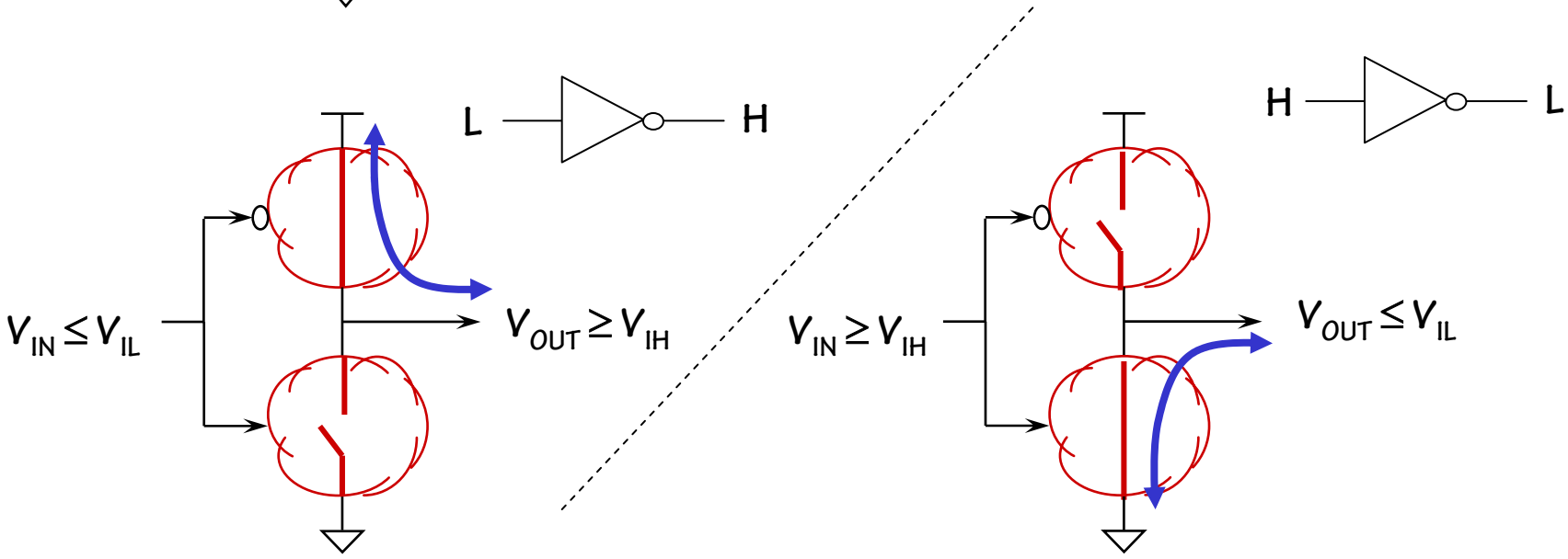


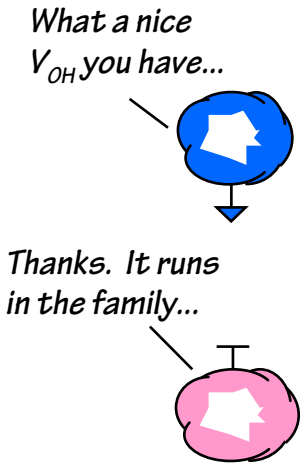
Think Switches



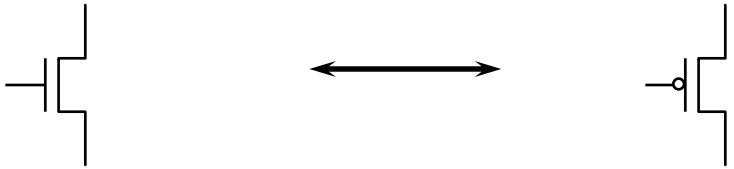
pullup: make this connection when V_{IN} near 0 so that $V_{OUT} = V_{DD}$

pulldown: make this connection when V_{IN} near V_{DD} so that $V_{OUT} = 0$



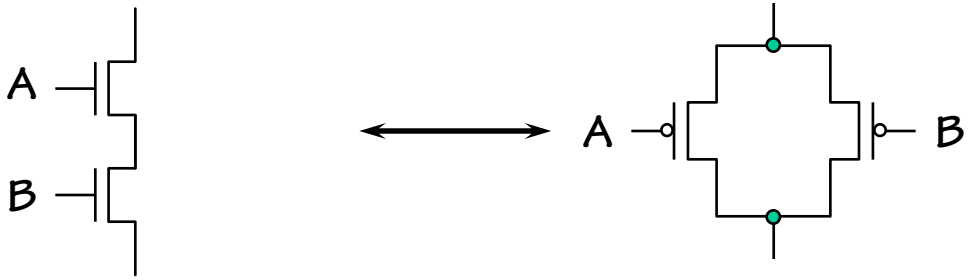


CMOS complements



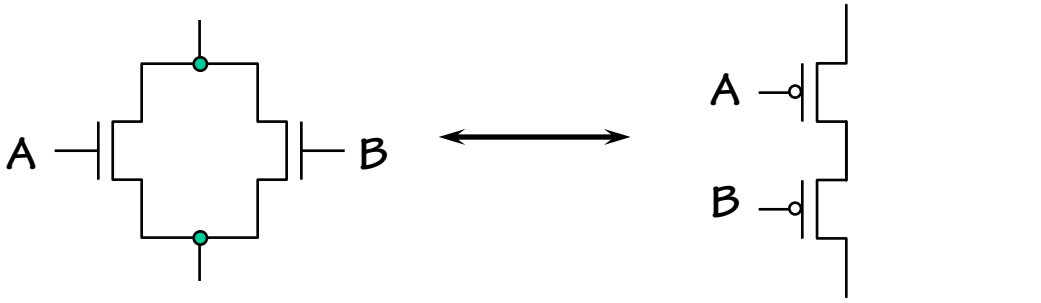
conducts when V_{GS} is high

conducts when V_{GS} is low



conducts when A is high <u>and</u> B is high: $A \cdot B$

conducts when A is <u>low</u> <u>or</u> B is low: $\overline{A}+\overline{B} = \overline{A \cdot B}$

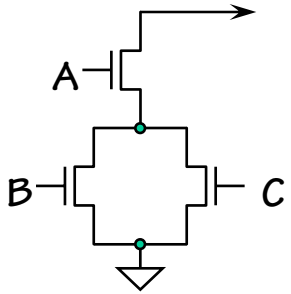


conducts when A is high <u>or</u> B is high: $A+B$

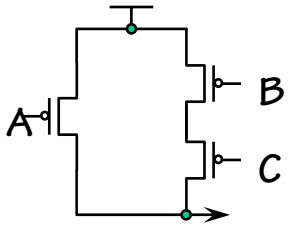
conducts when A is <u>low</u> <u>and</u> B is low: $\overline{A} \cdot \overline{B} = \overline{A+B}$

General CMOS gate recipe

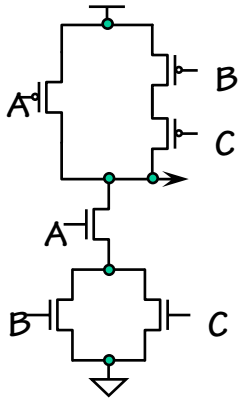
Step 1. Figure out pulldown network that does what you want, *e.g.*, F = A*(B+C) (What combination of inputs generates a low output)



Step 2. Walk the hierarchy replacing nfets with pfets, series subnets with parallel subnets, and parallel subnets with series subnets



Step 3. Combine pfet pullup network from Step 2 with nfet pulldown network from Step 1 to form fully-complementary CMOS gate.



But isn't it hard to wire it all up?