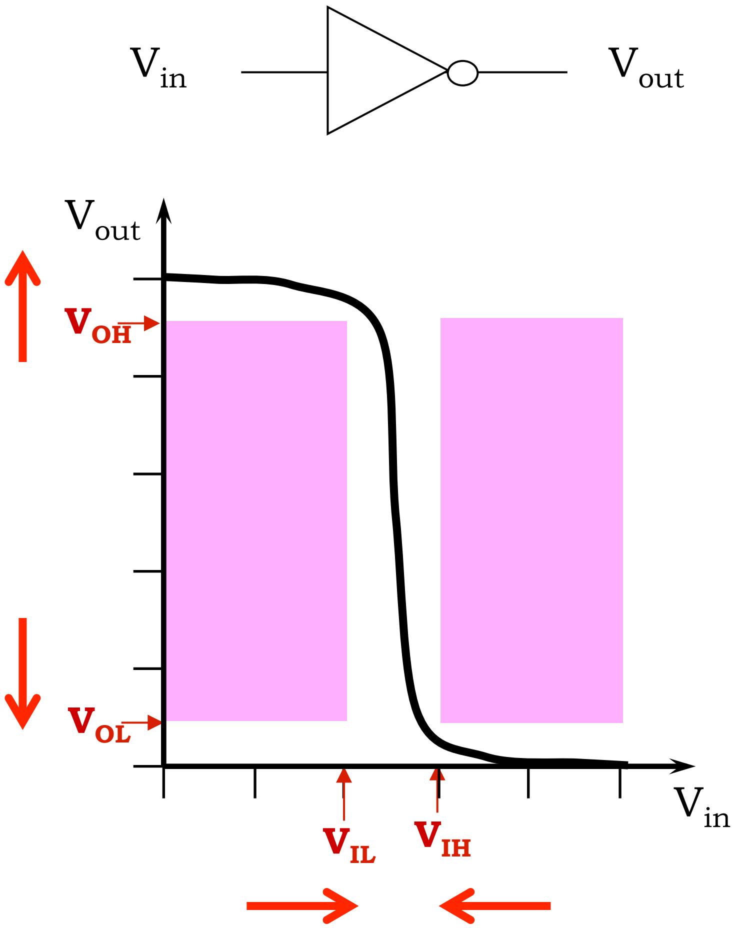
3. CMOS Technology

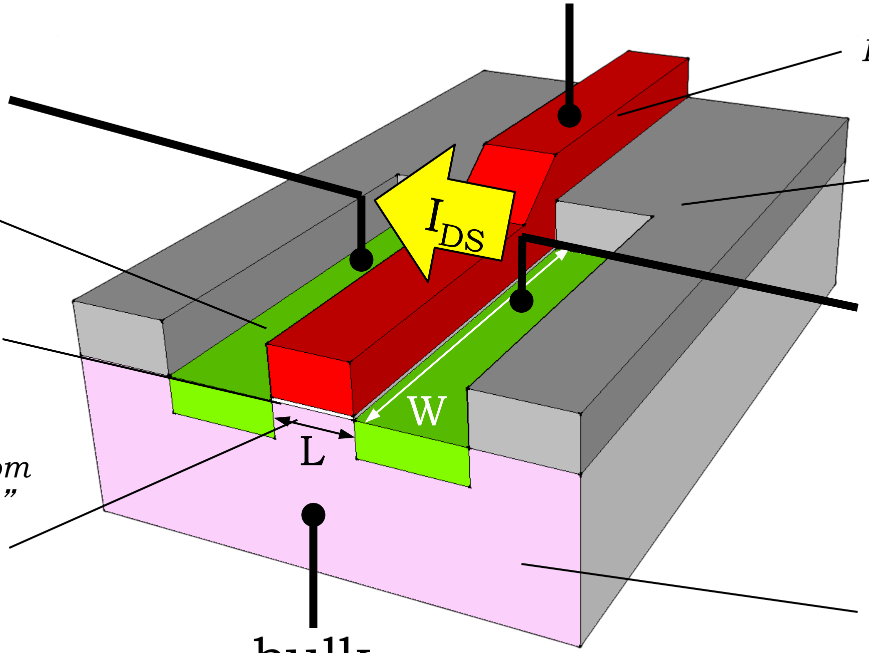
6.004x Computation Structures
Part 1 – Digital Circuits

Combinational Device Wish List



- ✓ Design our system to tolerate some amount of error
  ⇒ Add positive noise margins
  ⇒ VTC: gain>1 & nonlinearity
- ✓ Lots of gain ⇒ big noise margin
- ✓ Cheap, small
- ✓ Changing voltages will require us to dissipate power, but if no voltages are changing, we'd like zero power dissipation
- ✓ Want to build devices with useful functionality (what sort of operations do we want to perform?)

N-Channel MOSFET: Physical View



Polysilicon wire

Heavily doped n-type diffusions

Inter-layer SiO_2 insulation

Very thin (<20Å) high-quality SiO_2 insulating layer isolates gate from channel region.

$$\boxed{I_{DS} \propto W/L}$$

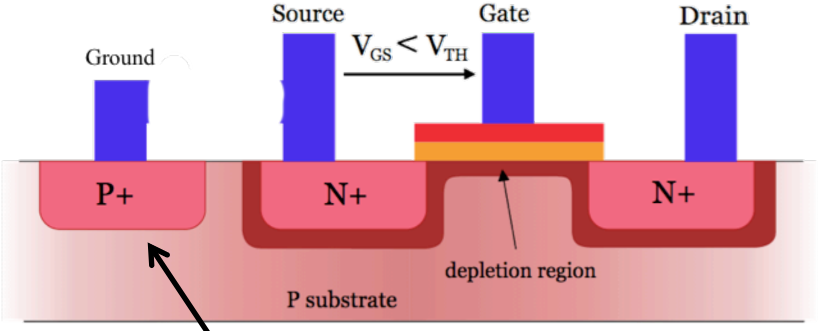
Channel region: electric field from charges on gate locally "inverts" type of substrate to create a conducting channel between source and drain.

Doped p-type silicon substrate

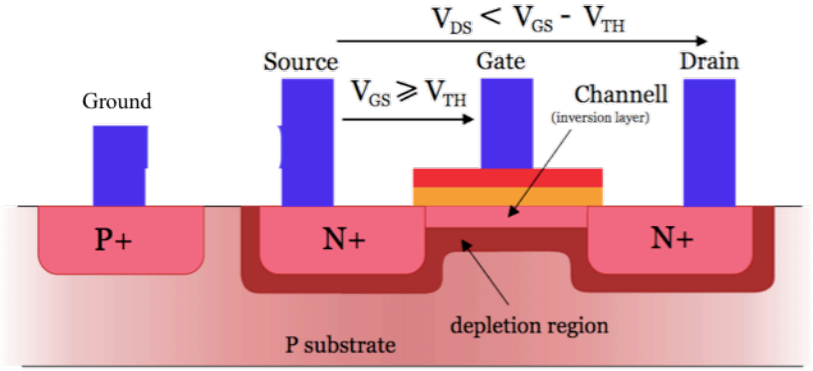
MOSFETs (metal-oxide-semiconductor field-effect transistors) are four-terminal voltage-controlled switches. Current flows between the diffusion terminals if the voltage on the gate terminal is large enough to create a conducting "channel", otherwise the mosfet is off and the diffusion terminals are not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

N-Channel MOSFET: Electrical View

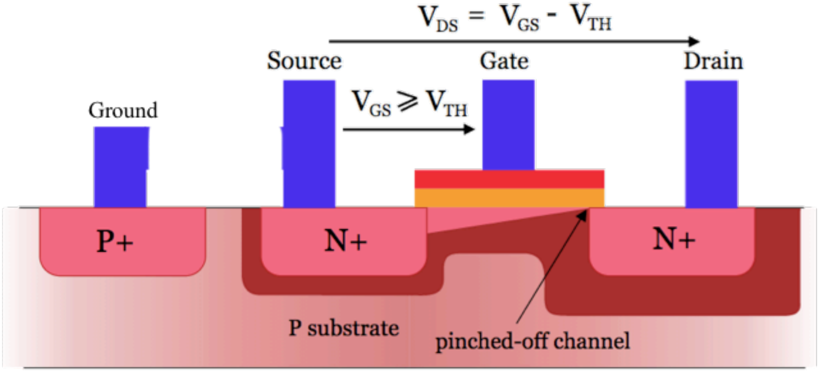
The four terminals of a Field Effect Transistor (gate, source, drain and bulk) connect to conductors that generate a set of electric fields in the channel region which depend on the relative voltages of each terminal.



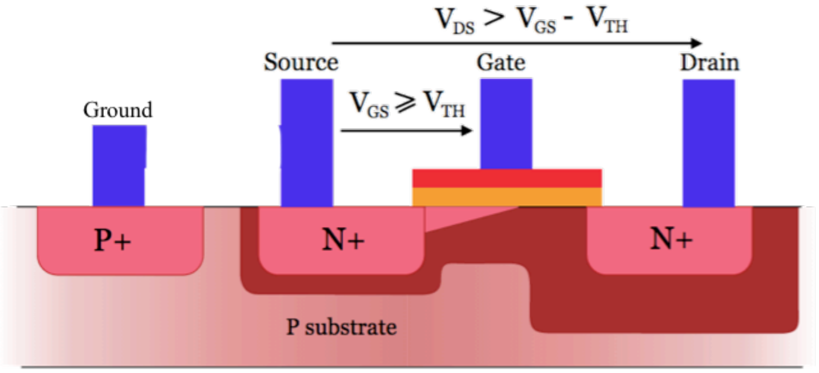
Want $V_P \leq V_N$



Linear operating region (ohmic mode)



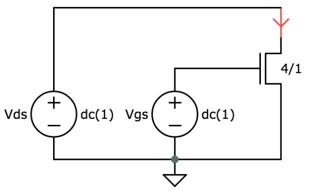
Saturation mode at point of pinch-off

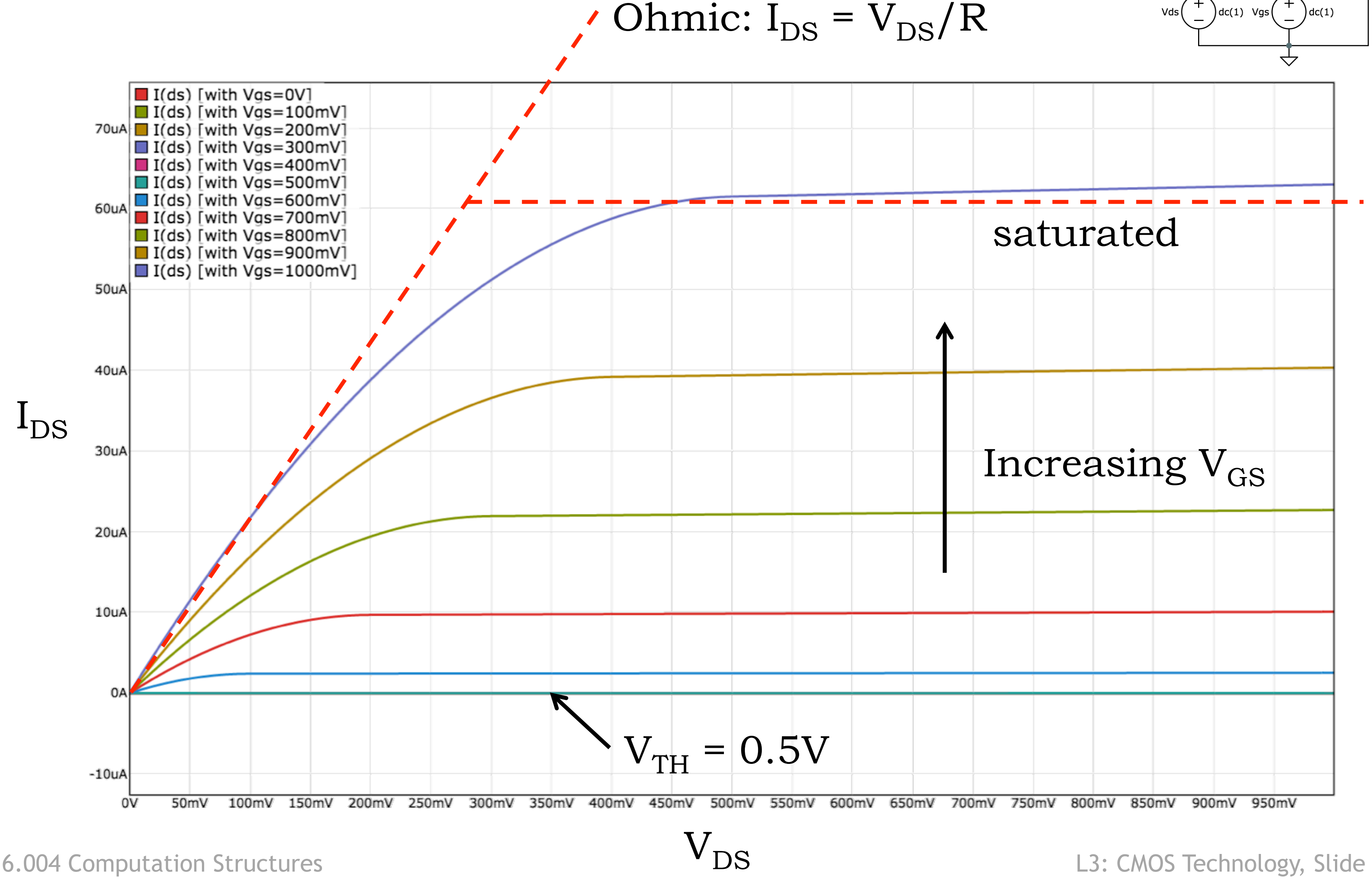


Saturation mode

Olivier Deleage and Peter Scott (CC BY-SA 3.0)

N-channel MOSFET I_{DS} vs. V_{DS}





Ohmic: $I_{DS} = V_{DS}/R$

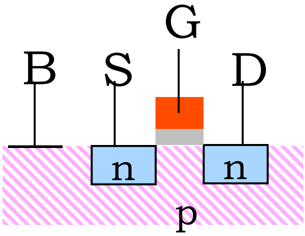
saturated

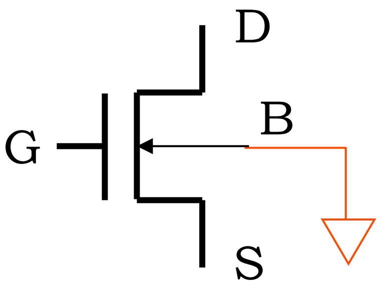
Increasing V_{GS}

$V_{TH} = 0.5V$

FETs Come in Two Flavors

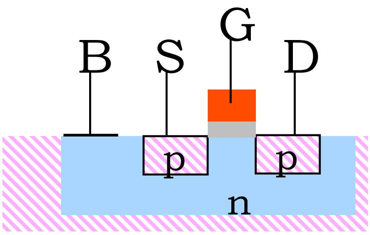
NFET: n-type source/drain diffusions in a p-type substrate. Positive threshold voltage; inversion forms n-type channel

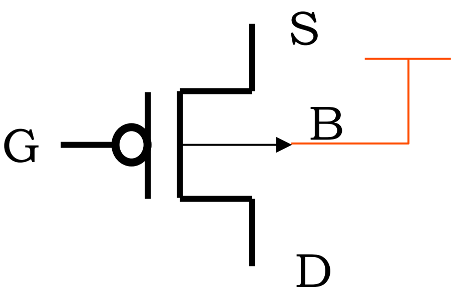




Connect B to GND to keep PN reverse-biased (Vp ≤ Vn); keeps D and S insulated from B

PFET: p-type source/drain diffusions in a n-type substrate. Negative threshold voltage; inversion forms p-type channel.





Connect B to VDD to keep PN reverse-biased

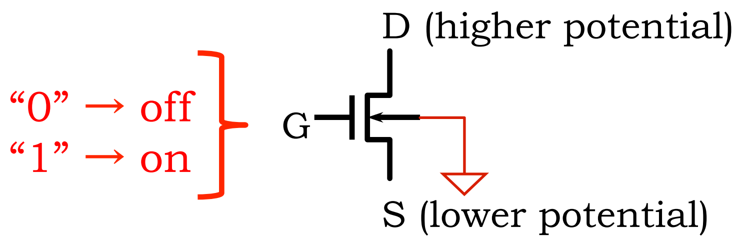
The use of both NFETs and PFETs – complimentary transistor types – is a key to CMOS (complementary MOS) logic families.

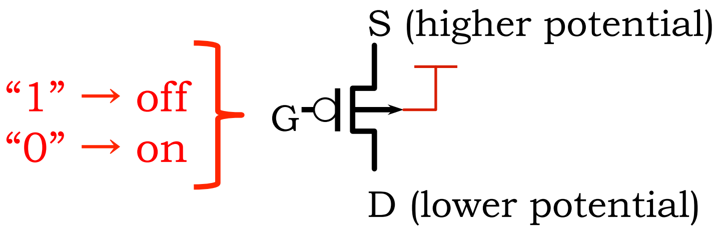
CMOS Recipe

If we follow two rules when constructing CMOS circuits, we can model the behavior of the mosfets as simple *voltage-controlled switches*:

Rule #1: only use NFETs in pulldown circuits

Rule #2: only use PFETs in pullup circuits

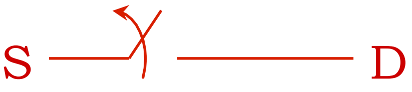




NFET Operating regions:

"off":

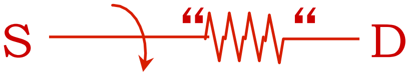
$V_{GS} < V_{TH,NFET}$



$V_{GS}\Uparrow \Rightarrow$ "R"$\Downarrow$

"on":

$V_{GS} > V_{TH,NFET}$

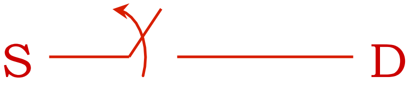


NFET threshold = ~0.5V

PFET Operating regions:

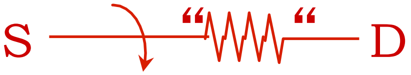
"off":

$V_{GS} > V_{TH,PFET}$



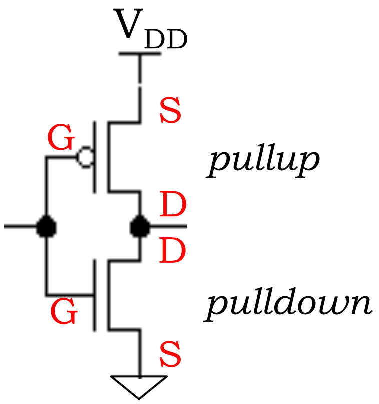
"on":

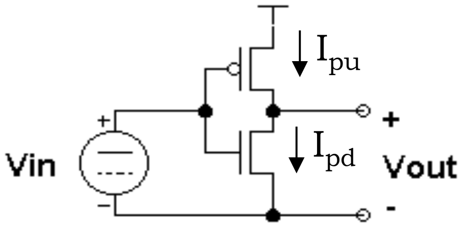
$V_{GS} < V_{TH,PFET}$



PFET threshold = ~ −0.5V

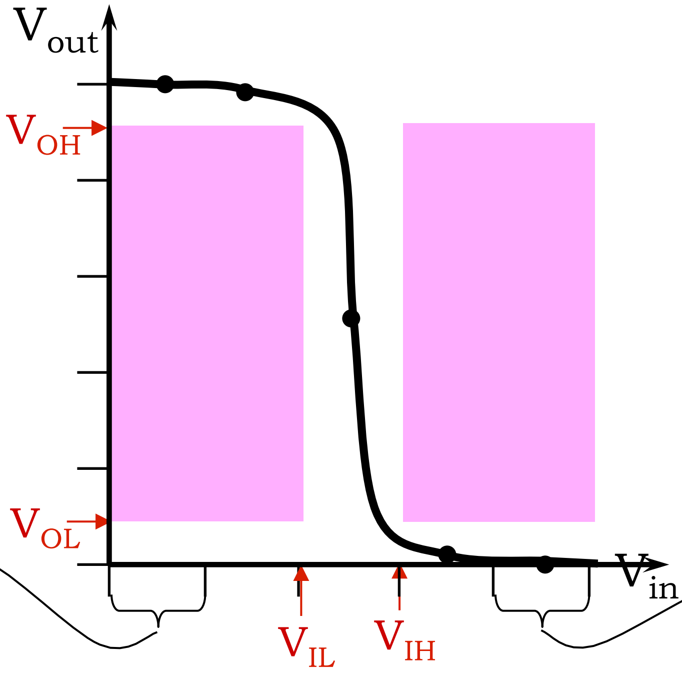
CMOS Inverter VTC





Steady state reached when V_{out} reaches value where $I_{pu} = I_{pd}$.

When V_{IN} is low, the nfet is off and the pfet is on, so current flows into the output node and V_{OUT} eventually reaches V_{DD} (> V_{OH}) at which point no more current will flow.



When V_{IN} is high, the pfet is off and the nfet is on, so current flows out of the output node and V_{OUT} eventually reaches GND (< V_{OL}) at which point no more current will flow.

Pfet "on"
nfet "off"

Pfet "off"
nfet "on"

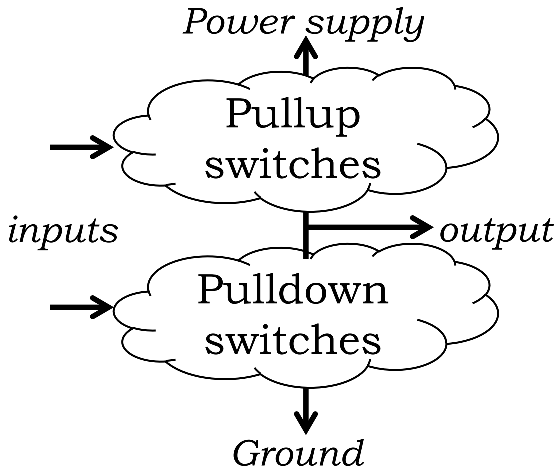
When V_{IN} is in the middle, both the pfet and nfet are "on" and the shape of the VTC depends on the details of the devices' characteristics. CMOS gates have very high gain in this region (small changes in V_{IN} produce large changes in V_{OUT}) and the VTC is almost a step function.

Beyond Inverters:
Complementary pullups and pulldowns

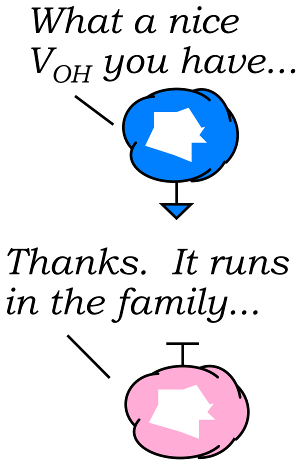
Now you know what the "C" in CMOS stands for!

We want *complementary* pullup and pulldown logic, i.e., the pulldown should be "on" when the pullup is "off" and vice versa.

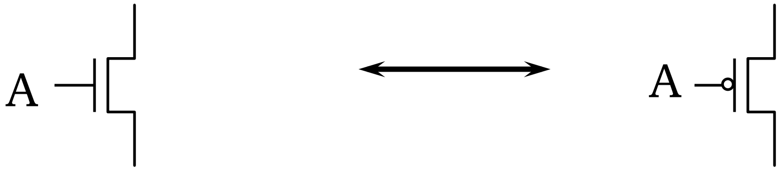
pullup	pulldown	F(inputs)
on	off	driven "1"
off	on	driven "0"
on	on	driven "X"
off	off	no connection



Since there's plenty of capacitance on the output node, when the output becomes disconnected it "remembers" its previous voltage -- at least for a while. The "memory" is the load capacitor's charge. Leakage currents will cause eventual decay of the charge (that's why DRAMs need to be refreshed!).

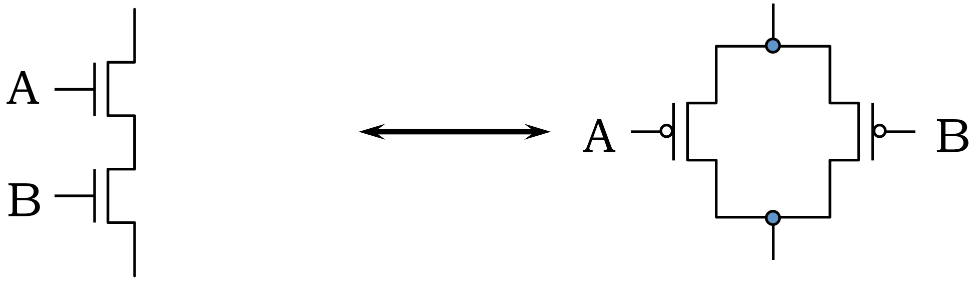

What a nice V_{OH} you have...
Thanks. It runs in the family...

CMOS Complements



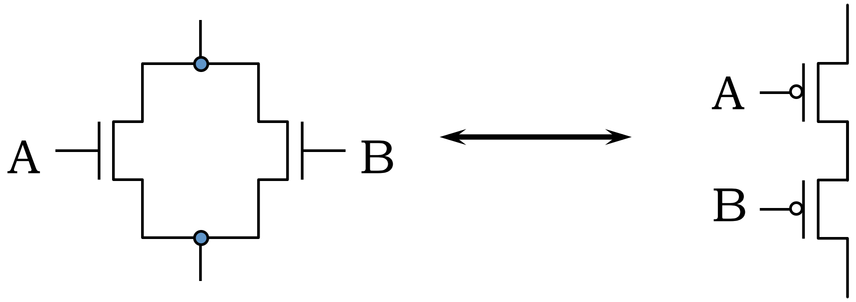
conducts when A is high

conducts when A is low: $\overline{A}$



conducts when A is high
and B is high: $A\cdot B$

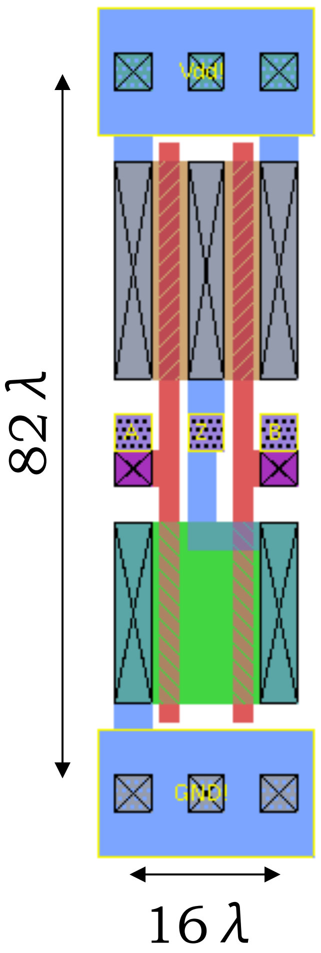
conducts when A is low
or B is low: $\overline{A}+\overline{B} = \overline{A\cdot B}$



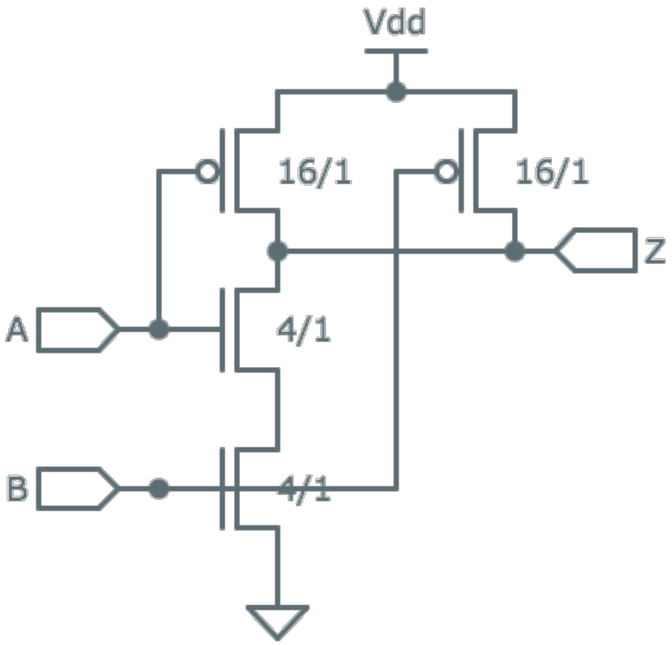
conducts when A is high
or B is high: $A+B$

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

A Pop Quiz!



Current technology: λ = 14nm



What function does this gate compute?

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

NAND

COST for an older 45nm process:

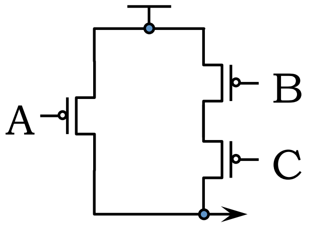
- \$3500 per 300mm wafer
- 300mm round wafer = $\pi(150e^{-3})^2 = .07m^2$
- NAND gate = $(82)(16)(45e^{-9})^2=2.66e^{-12}m^2$
- $2.6e^{10}$ NAND gates/wafer (= 100 billion FETS!)
- marginal cost of NAND gate: 132n\$

General CMOS Gate Recipe

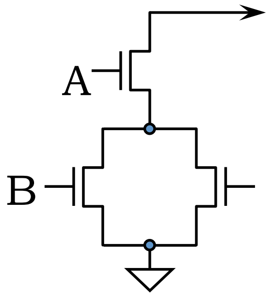
Step 1. Figure out the pullup network that does what you want, *e.g.*,

$$F = \overline{A} + \overline{B} \cdot \overline{C}$$

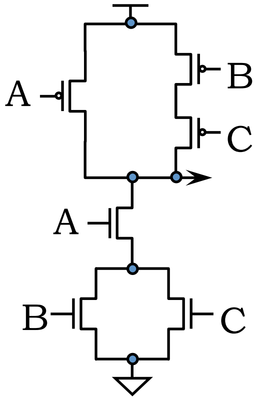
(Determine what combination of inputs generates a high output)



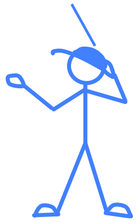
Step 2. Walk the hierarchy replacing nfets with pfets, series subnets with parallel subnets, and parallel subnets with series subnets



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



Does this recipe work for all logic functions?



CMOS Gates Are Naturally Inverting

In a CMOS gate, rising inputs (0→1) lead to falling outputs

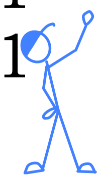
- NFETs go from "off" to "on"
  → pulldown paths connected
  → output may be connected to ground

- PFETs go from "on" to "off"
  → pullup paths disconnected
  → output may be disconnected from V_{DD}

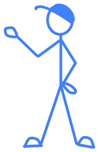
For CMOS gate:
All inputs 0
→ nfets off, pfets on
→ output must be 1
All inputs 1
→ nfets on, pfets off
→ output must be 0

Corollary: you can't build positive logic, e.g., AND, with one CMOS gate

A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1



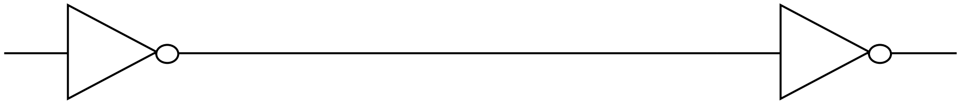
A=1, B rising…



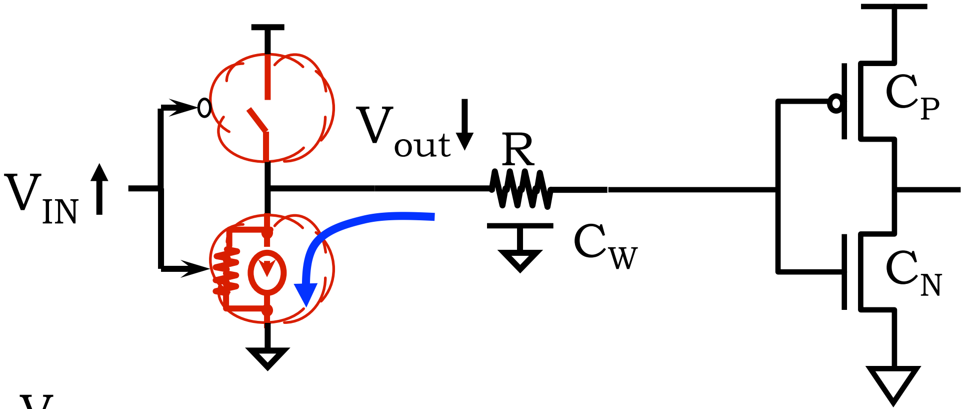
Oops, output is also rising!

CMOS Timing Specifications

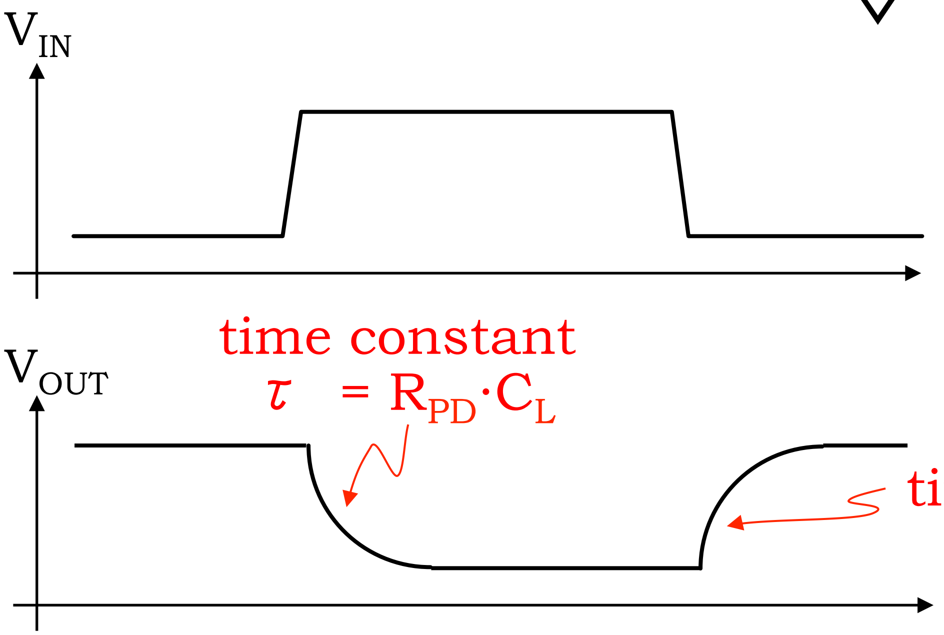
Circuit:



Electrical model:



Waveforms:

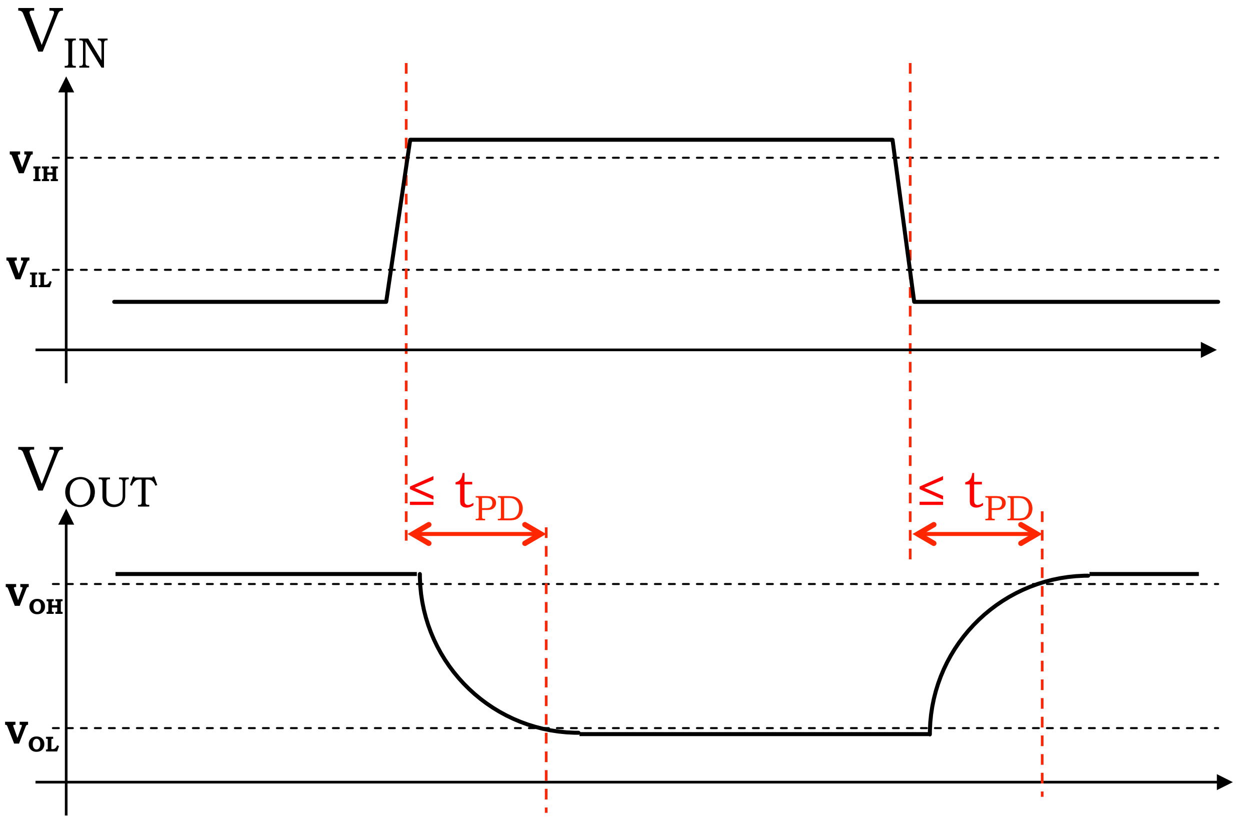


time constant
$\tau = R_{PD} \cdot C_L$

time constant
$\tau = R_{PU} \cdot C_L$

Propagation Delay

Propagation delay (t_{PD}): An UPPER BOUND on the delay from valid inputs to valid outputs.

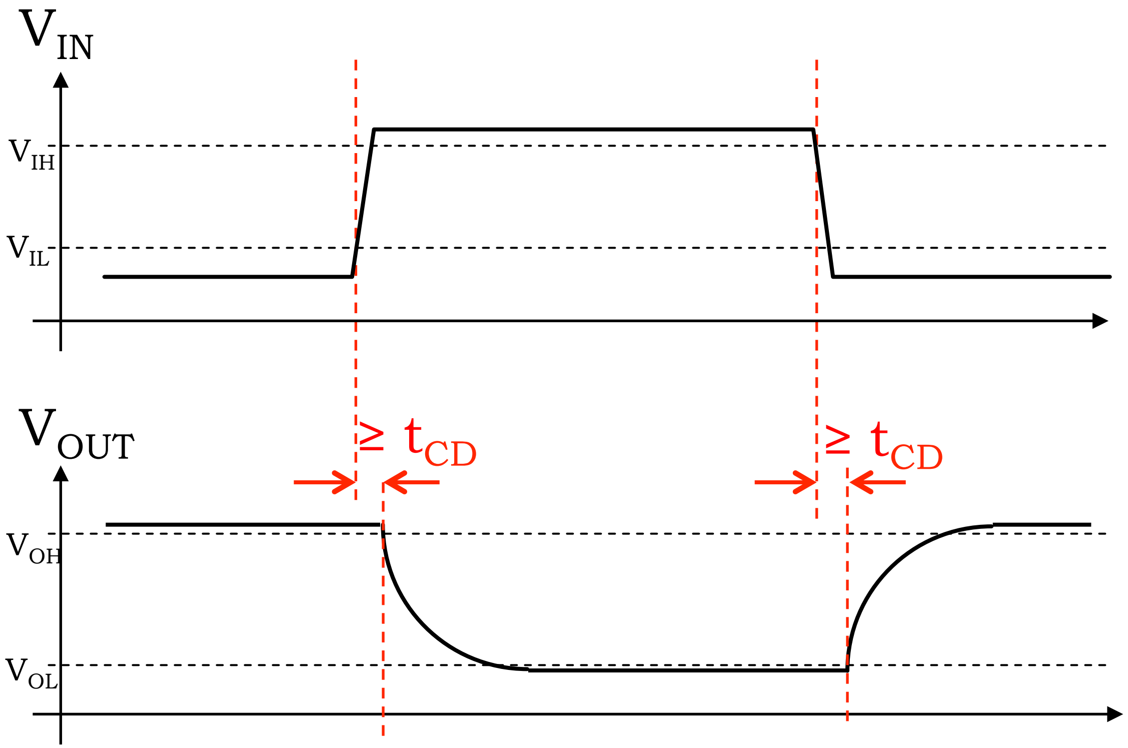


GOAL:
minimize propagation delay!

ISSUE:
keep capacitances low and transistors fast

Contamination Delay

Contamination delay (t_{CD}): A LOWER BOUND on the delay from any invalid input to an invalid output

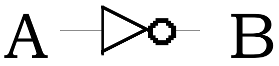


Do we really need t_{CD}?

Usually not… it'll be important when we design circuits with registers (coming soon!)

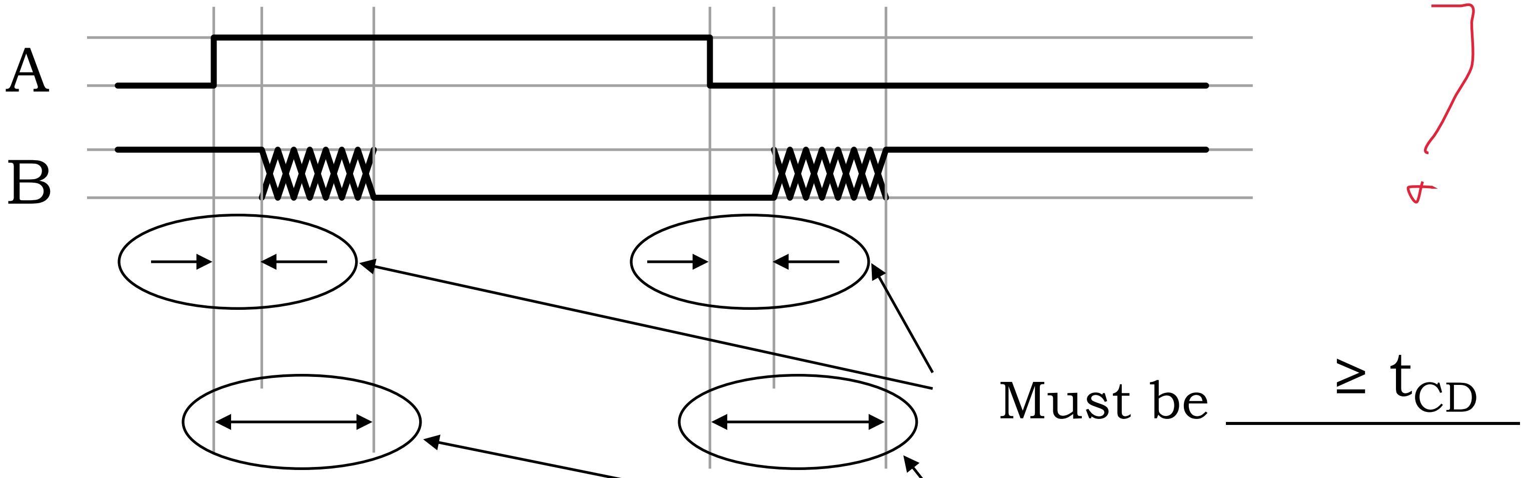
If t_{CD} is not specified, safe to assume it's 0.

The Combinational Contract



A	B
0	1
1	0

t_{PD} propagation delay
t_{CD} contamination delay



Must be $\geq t_{CD}$

Must be $\leq t_{PD}$

Notes:

1. *No Promises* during
2. Default (conservative) spec: $t_{CD} = 0$

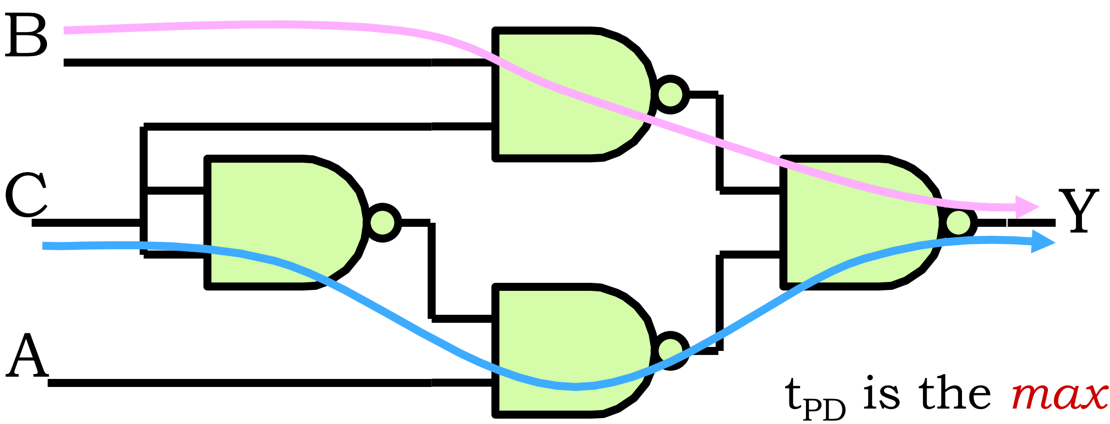
Acyclic Combinational Circuits

If NAND gates have a t_{PD} = 4nS and t_{CD} = 1nS

t_{PD} = 12 nS

t_{CD} = 2 nS

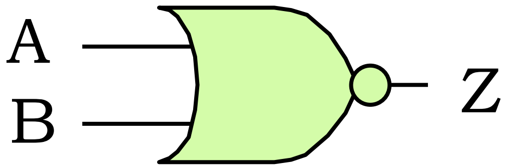
t_{CD} is the *minimum* cumulative contamination delay over all paths from inputs to outputs



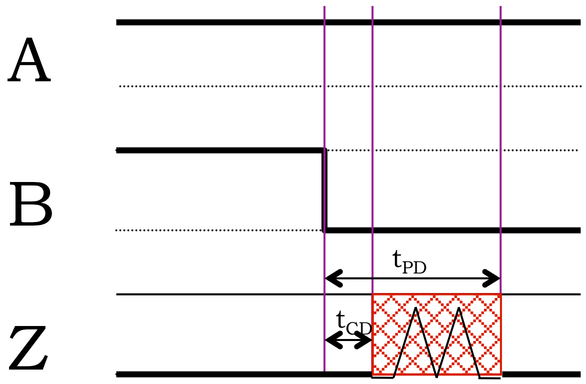
t_{PD} is the *maximum* cumulative propagation delay over all paths from inputs to outputs

One Last Timing Issue…

NOR:



A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

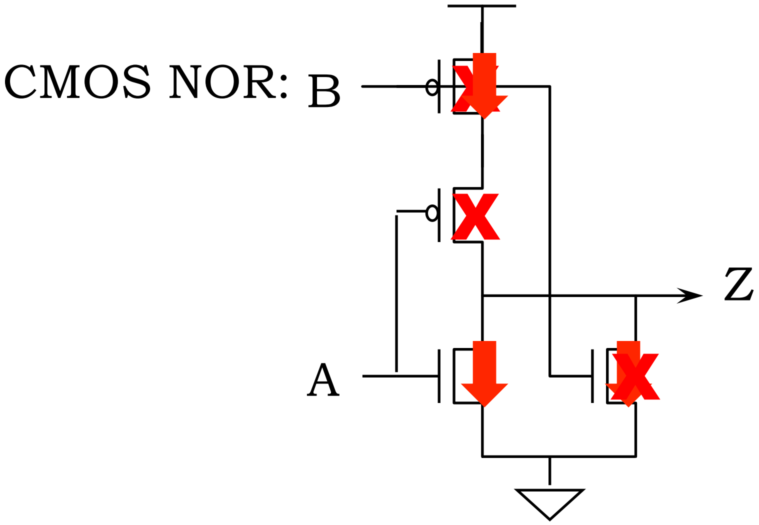


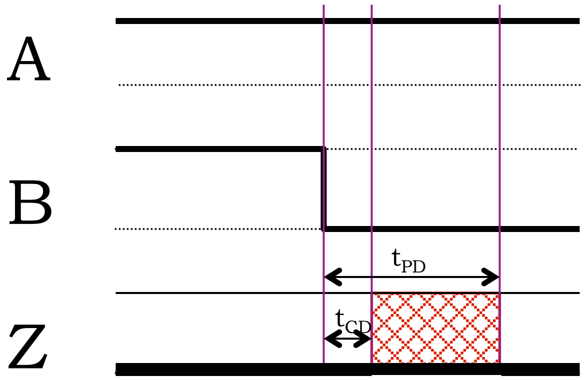
Recall the rules for *combinational devices*:

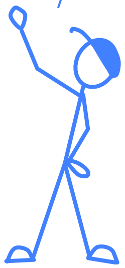
Output guaranteed to be valid when *all* inputs have been valid for at least t_{PD}, and, outputs may become invalid no earlier than t_{CD} after an input changes!

Many gate implementations—e.g., CMOS— adhere to even tighter restrictions.

What Happens In This Case?







Input A=1 is sufficient to determine the output

LENIENT Combinational Device:

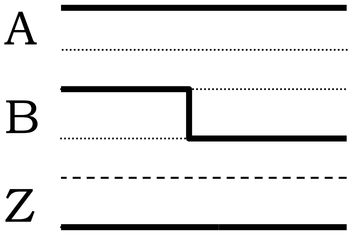
Output guaranteed to be valid when any combination of inputs sufficient to determine the output value has been valid for at least t_{PD}. *Tolerates transitions -- and invalid levels -- on irrelevant inputs!*

NOR:

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

Lenient NOR:

A	B	Z
0	0	1
X	1	0
1	X	0



Summary

- CMOS
  - Only use NFETs in pulldowns, PFETs in pullups → mosfets behave as voltage-controlled switches
  - Series/parallel Pullup and pulldown switch circuits are complementary
  - CMOS gates are naturally inverting (rising input transition can only cause falling output transition, and vice versa).
  - "Perfect" VTC (high gain, $V_{OH} = V_{DD}$, V_{OL} = GND) means large noise margins and no static power dissipation.
- Timing specs
  - t_{PD}: upper bound on time from valid inputs to valid outputs
  - t_{CD}: lower bound on time from invalid inputs to invalid outputs
  - If not specified, assume $t_{CD} = 0$
  - Lenient gates: output unaffected by some input transitions
- Next time: logic simplification, other canonical forms