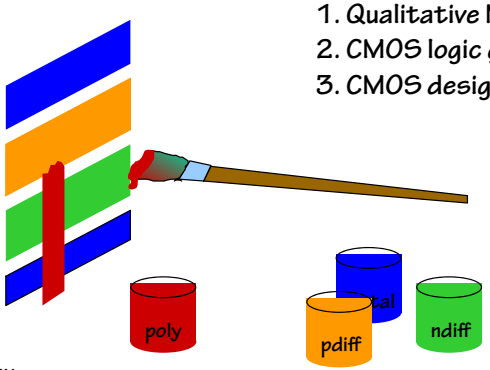
6.004 Computation Structures
Spring 2009

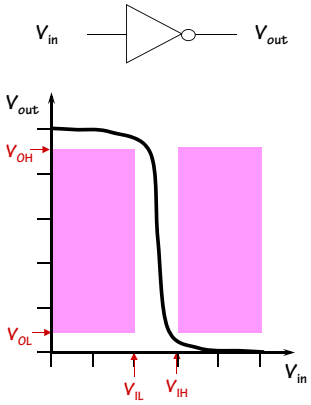
CMOS Technology



1. Qualitative MOSFET model
2. CMOS logic gates
3. CMOS design issues

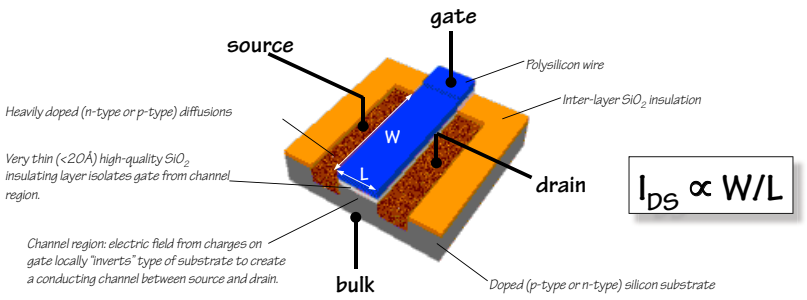
NEXT WEEK:
- TUE: no lecture
- THU: Lab 1 due!
- FRI: QUIZ 1!!!

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?)

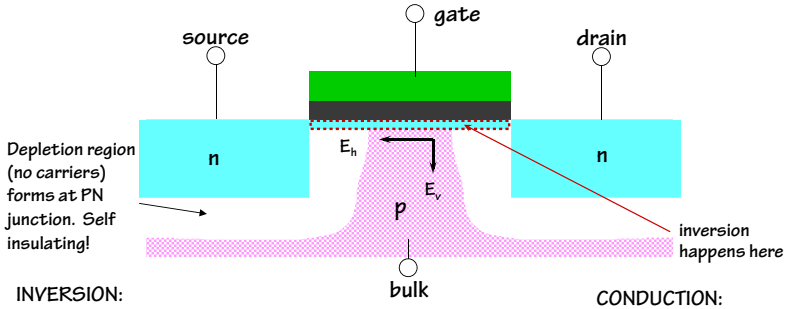
MOSFETS: Gain & non-linearity



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.

FETs as switches

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

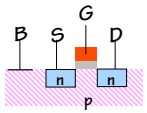


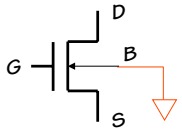
INVERSION:
A sufficiently strong vertical field will attract enough electrons to the surface to create a conducting n-type channel between the source and drain. The gate voltage when the channel first forms is called the *threshold voltage* -- the mosfet switch goes from "off" to "on".

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

FETs come in two flavors

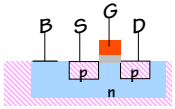
NFET: n-type source/drain diffusions in a p-type substrate. Positive threshold voltage; inversion forms n-type channel

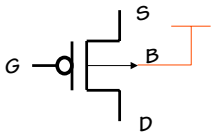




Connect B to GND to keep PN reverse-biased (Vp < Vn); keeps D and S insulated from B

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





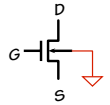
Connect B to VDD to keep PN reverse-biased

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 then we can model the behavior of the mosfets as simple switches:

Rule #1: only use NFETs in pulldown circuits (paths from output node to GND)
Rule #2: only use PFETs in pullup circuits (paths from output node to V_{DD})

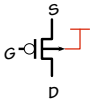


NFET Operating regions:

"off": $V_G < V_{TH,NFET}$

"on": $V_G > V_{TH,NFET}$

$\sim V_{DD}/5$



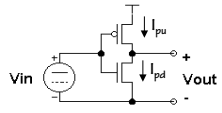
PFET Operating regions:

"off": $V_G > V_{DD} + V_{TH,PFET}$

"on": $V_G < V_{DD} + V_{TH,PFET}$

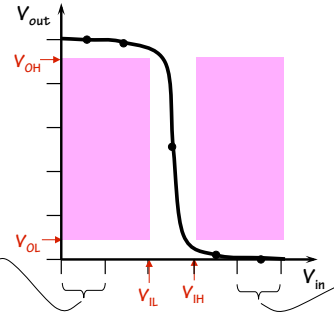
$\sim -V_{TH,NFET}$

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 VIN 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

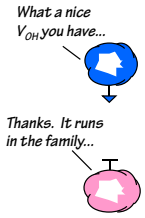
Now you know what the "C" in CMOS stands for!

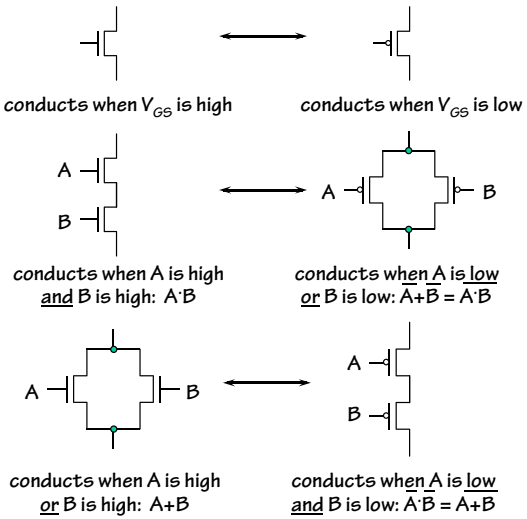
We want *complementary* pullup and pulldown logic, i.e., the pulldown should be "on" when the pullup is "off" and vice versa.

pullup	pulldown	F(A1,…,An)
on	off	driven "1"
off	on	driven "0"
on	on	driven "X"
off	off	no connection

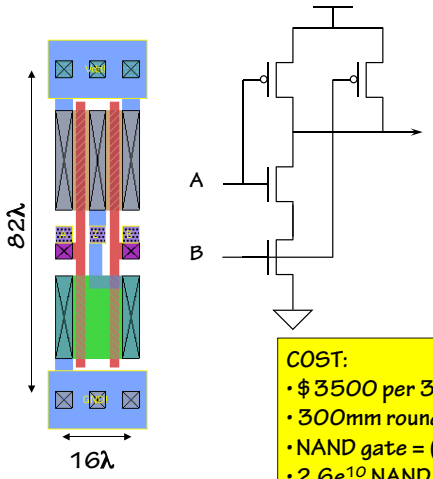
Since there's plenty of capacitance on the output node, when the output becomes disconnected it "remembers" its previous voltage -- at least for a while. The "memory" is the load capacitor's charge. Leakage currents will cause eventual decay of the charge (that's why DRAMs need to be refreshed!).

CMOS complements





A pop quiz!



What function does this gate compute?

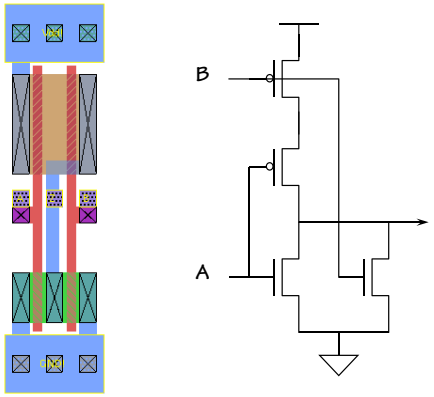
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

NAND

Current technology: λ = 45nm

COST:
- $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$

Here's another…



What function does this gate compute?

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

NOR

General CMOS gate recipe

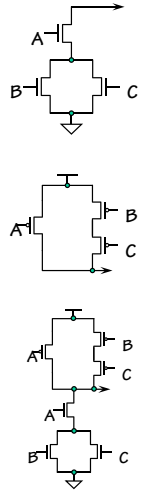
Step 1. Figure out pulldown network that does what you want, *e.g.*,

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

(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.

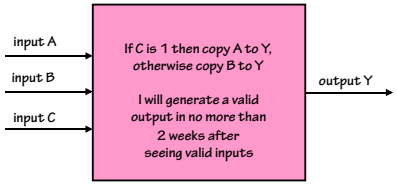


So, whats the big deal?

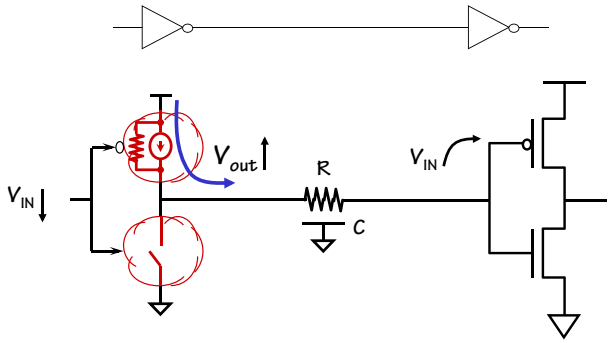
A Quick Review

- 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



Big Issue 1: Wires



Today (i.e., 100nm):

$\tau_{RC} \approx 50$ps/mm

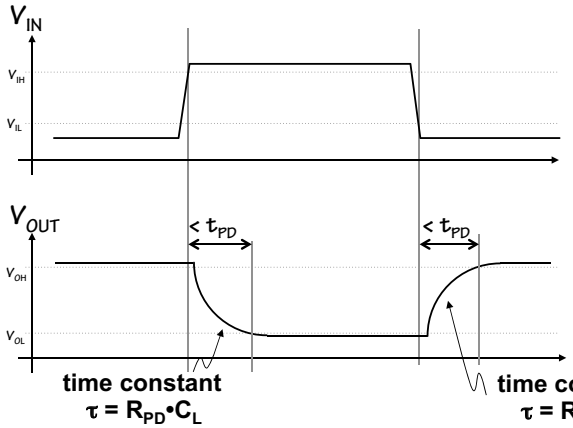
Implies > 1 ns to traverse a 20mm x 20mm chip

This is a long time in a 2GHz processor

Due to unavoidable delays…

Propagation delay (t_{PD}):

An UPPER BOUND on the delay from valid inputs to valid outputs.



GOAL:

minimize propagation delay!

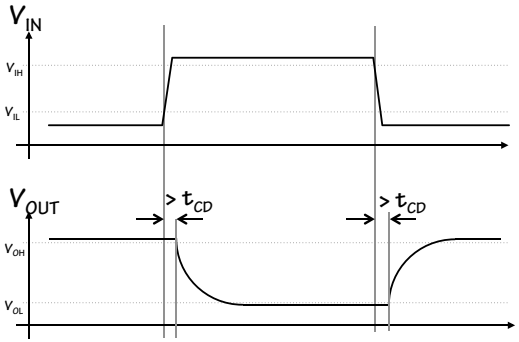
ISSUE:

keep Capacitances low and transistors fast

Contamination Delay

an optional, additional timing spec

INVALID inputs take time to propagate, too…



Do we really need t_{CD}?

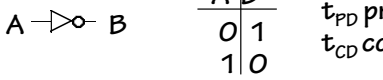
Usually not… it'll be important when we design circuits with registers (coming soon!)

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

CONTAMINATION DELAY, t_{CD}

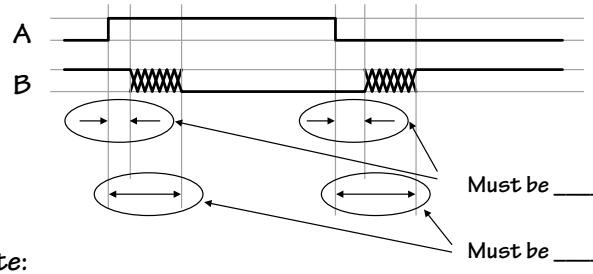
A LOWER BOUND on the delay from any invalid input to an invalid output

The Combinational Contract



A	B
0	1
1	0

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



Must be $> t_{CD}$

Must be $< t_{PD}$

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

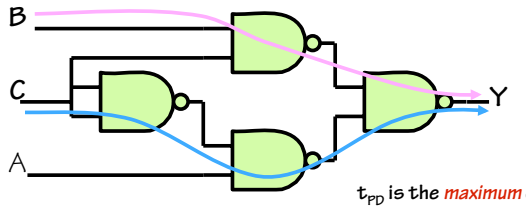
Acyclic Combinational Circuits

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

$t_{PD} =$ 12 nS

$t_{CD} =$ 2 nS

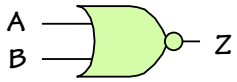
t_{CD} is the *minimum* cumulative contamination delay over all paths from inputs to outputs



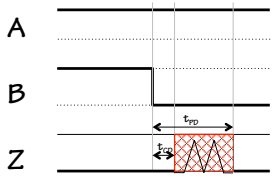
t_{PD} is the *maximum* cumulative propagation delay over all paths from inputs to outputs

Oh yeah… one last issue

NOR:



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



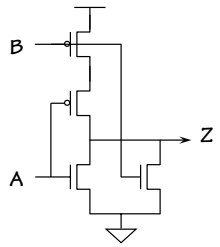
Recall the rules for *combinational devices*:

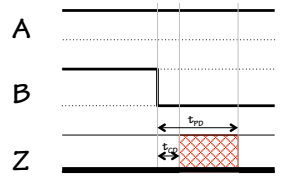
Output guaranteed to be valid when all inputs have been valid for at least t_{PD}, and, outputs may become invalid no earlier than t_{CD} after an input changes!

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

What happens in this case?

CMOS NOR:





Input A alone is sufficient to determine the output

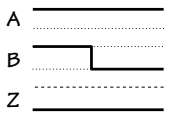
***LENIENT* Combinational Device:**
Output guaranteed to be valid when any combination of inputs sufficient to determine output value has been valid for at least t_{PD}.
Tolerates transitions -- and invalid levels -- on irrelevant inputs!

NOR:

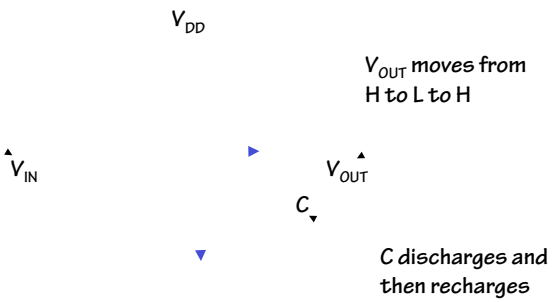
A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

Lenient NOR:

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



Big Issue 2: Power



Energy dissipated = $C V_{DD}^2$ per cycle
Power consumed = $f n C V_{DD}^2$ per chip

where f = frequency of charge/discharge
n = number of gates /chip

Unfortunately…

Modern chips (UltraSparc III, Power4, Itanium 2) dissipate from 80W to 150W with a Vdd ≈ 1.2V (Power supply current is ≈ 100 Amps)

Cooling challenge is like making the filament of a 100W incandescent lamp cool to the touch!

Worse yet…
- Little room left to reduce Vdd
- nC and f continue to grow

Hey: could we Somehow recycle the charge?

MUST computation consume energy?

(a tiny digression…)

How energy-efficient can we make a gate? It seems that switching the input to a NAND gate will always dissipate some energy…

http://www.research.ibm.com/journal/rd/441/landauerii.pdf

Landauer's Principle (1961): *discarding information* is what costs energy!

http://www.research.ibm.com/journal/rd/176/ibmrd1706G.pdf

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

NAND GATE: 2 bits → 1 bit *(information Loss!)*

Bennett (1973): Use *reversible* logic gates, not NAND, and *there's no lower bound to energy use!*

The fundamental physical limits of computation, Bennett & Landauer, Scientific American. Vol. 253, pp. 48-56. July 1985

A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

FEYNMAN GATE: 2 bits → 2 bits *(information Preserving!)*

Bennett, Fredkin, Feynman, others: Computer systems constructed from info-preserving elements.

Theory: NO lower bound on energy use!

Practice: Research frontier (qubits, etc.)

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