# Lecture 3

# It is a Digital World

Adapted from: Mark Horowitz
Stanford University
horowitz@stanford.edu

w/ material from Don Stark and Subhasish Mitra

# Overview

Reading

W&H 2.5.1-2.5.2 – Voltage transfer and noise margins

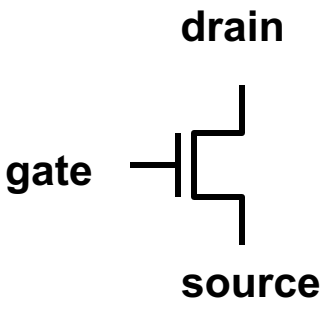
+ W&H Chapter 2 – Transistor models

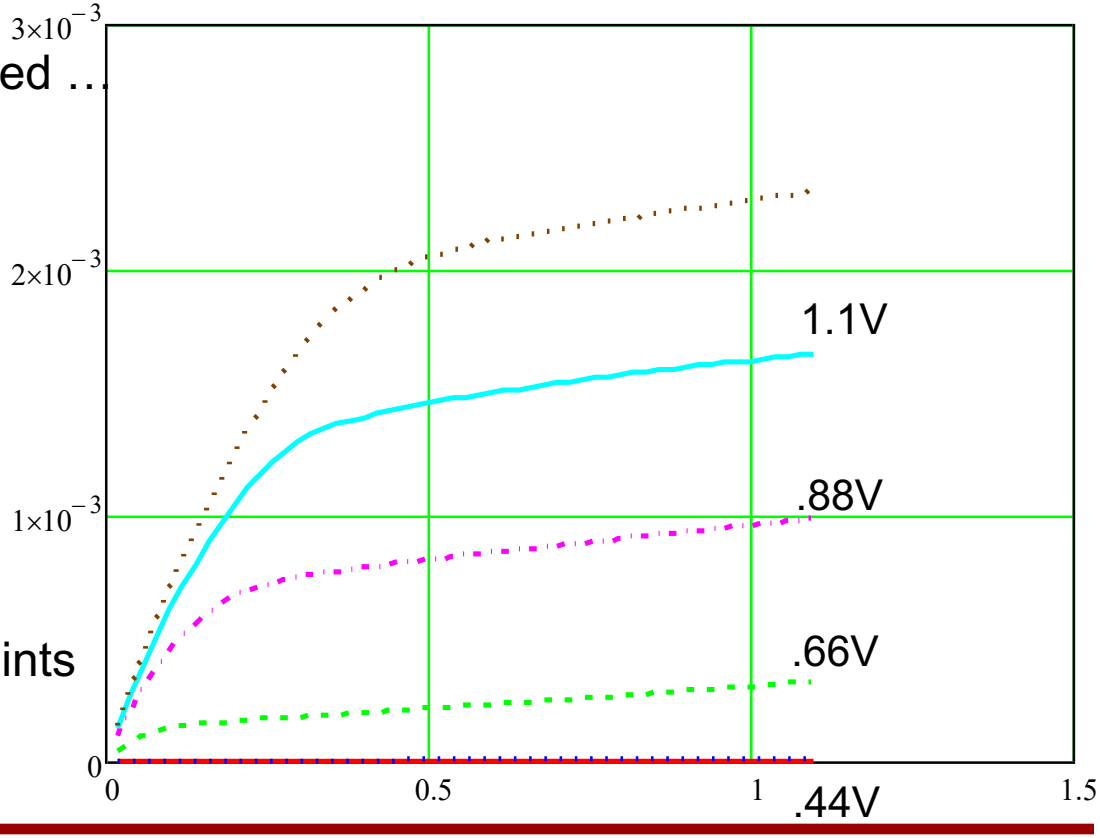
Introduction

In this lecture we look relook at transistors and wires in more detail to better understand why our buffer won't work, and to figure out the parameters that we will need to understand the power and delay of CMOS gates. While a transistor is a complicated non-linear element, for most of the time we can use a simple abstraction for it, modeling it as a switched resistor. And it turns out that we only need to understand the resistance and the capacitance of our transistors and wires to understand both power and delay.

# Real Transistors

- The voltage on the gate controls the current that flows between the source and drain. The transistor model is often displayed by drawing its current-voltage curve.
- But this is pretty complicated …




- Simplify by adding constraints

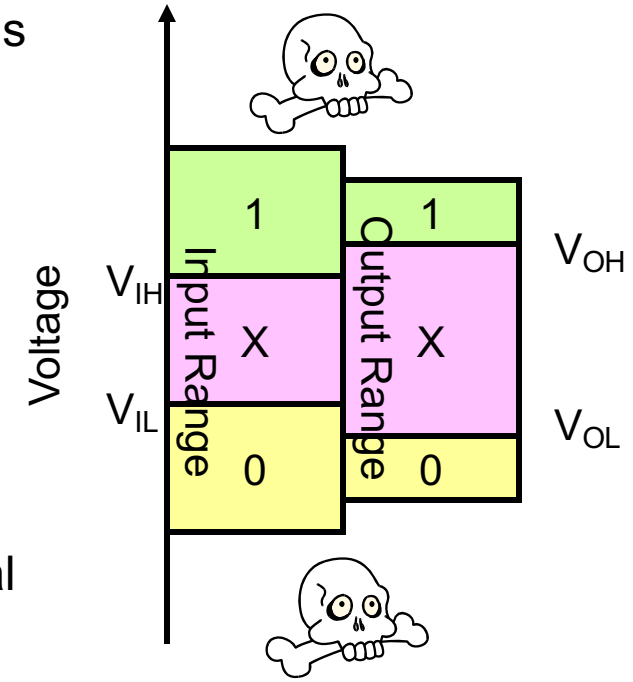
# Simplify using Digital Constraint

Rather than worrying about the precise voltages on the terminals of the transistor, guarantee that voltages will fall within two regions, one represents a logic ‘0’ and the other a ‘1’.

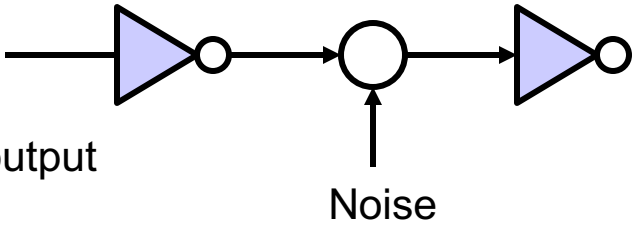
- Need to compute the output only for inputs in the allowable range
  - Much simpler than before
  - Model transistor as being either conducting, or off
- Need to ensure that the output is always in the allowable voltage range
  - Need to make sure to produce valid digital outputs to the next stage
  - Also want to have level restore. Allowable voltage range for output range should be smaller than allowable input range

# The Digital Abstraction

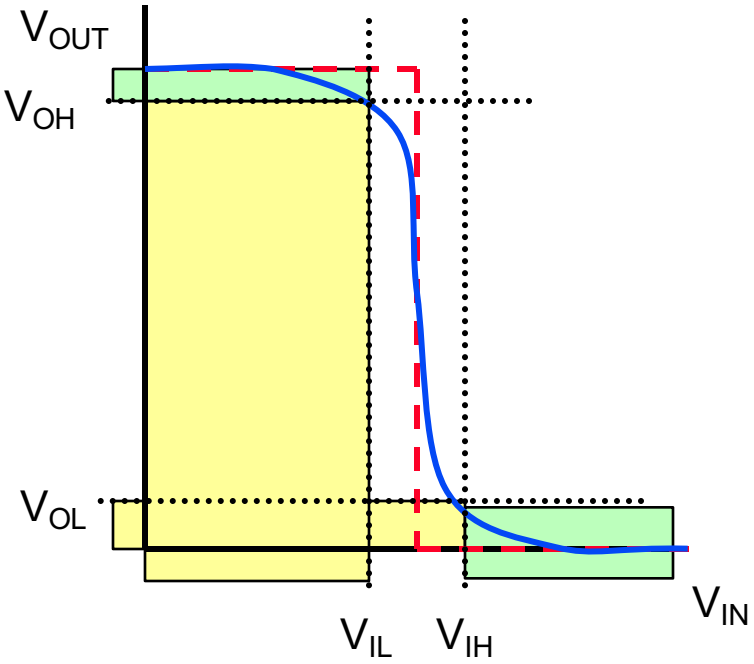
- Divide voltage into discrete regions
  - Logic 0
  - Logic 1
  - X - between 0 and 1
  - Out of range
    - may damage devices




- Each logic gate *restores* the signal
  - Output noise < input noise
  - Noise is not cumulative
    - In fact it is attenuated
  - Noise margin
    - How much noise won't change output



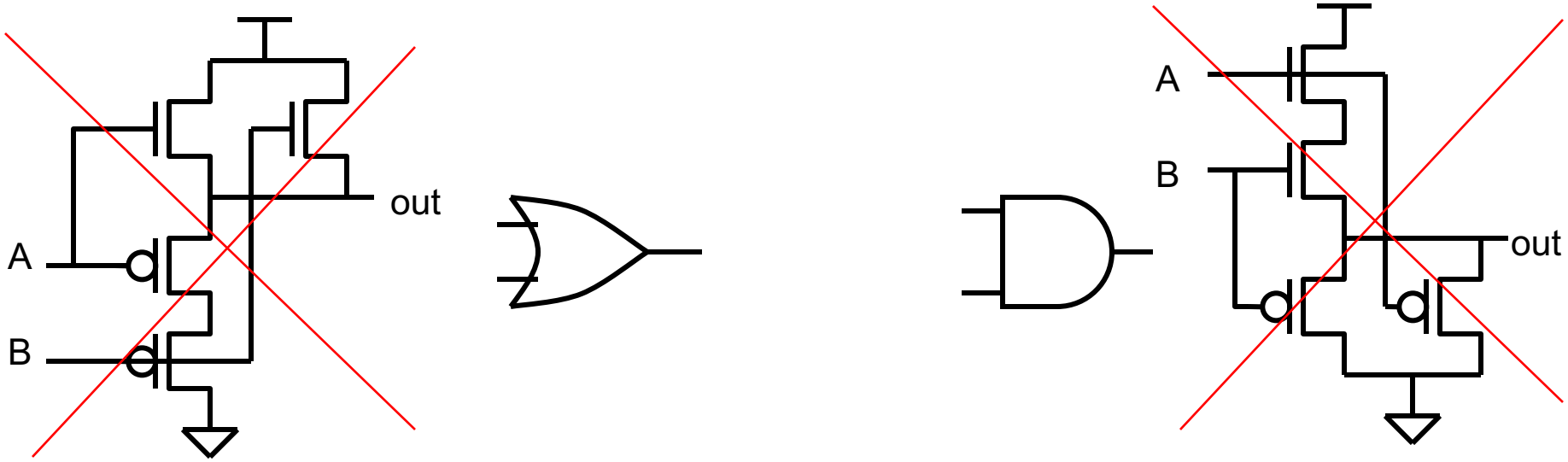
# CMOS Inverting Gates: Nice Digital Circuits




The dashed red line is an idea gate, since the output noise would be zero. CMOS gates are very close to ideal

- A great digital gate
  - Gain = 0 near voltage near Vdd or Gnd
    - Noise is attenuated
  - Relatively sharp transition between 1 and 0 output
    - Allowable input range is pretty large without getting wrong answer!

# How About AND & OR?

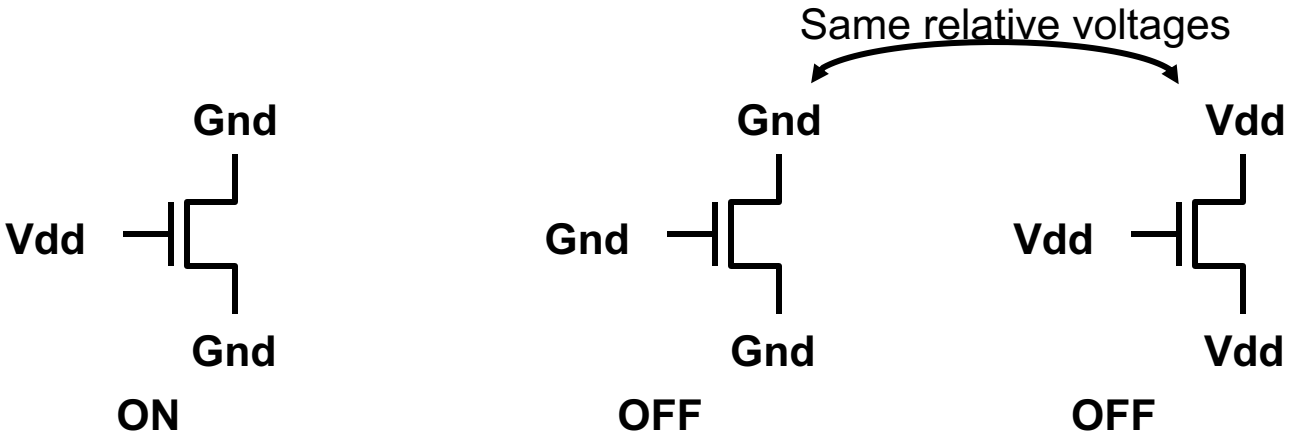



- These circuits don't work
  - Transistors turn themselves off
  - nMOS like to conduct Gnd (0)
  - pMOS like to conduct Vdd (1)

# Basic Problem With Simple Model

- Voltage is a relative measurement
  - 1V does not mean anything
  - It must be 1V between A and B
  - PLEASE PLEASE remember this!!!

- In previous model nMOS conducts when Gate is at Vdd
  - But what are the other terminals at?



# The Truth

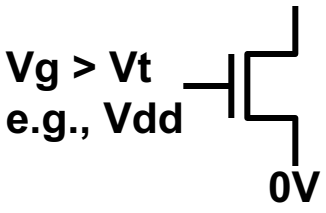
- In inverting logic gates
  - There is always voltage from gate to source
  - Positive Vdd for nMOS
  - Negative Vdd for pMOS

- For nMOS instead of conducting when gate is a 1 (Vdd)
  - At Vdd relative to what, you should ask
  - Truth is that Gate to Source must be Vdd
- For pMOS instead of conducting when gate is a 0 (Gnd)
  - At Gnd relative to what, you should ask
  - Truth is that Gate to Source must be –Vdd

- Check previous logic gates, this is always true!

# More Accurate nMOS Model

## nMOS pulling low

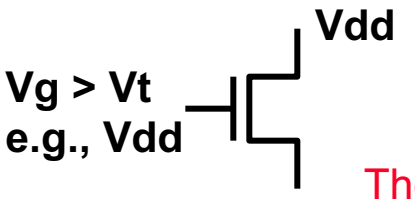



The voltage on this terminal will be 0V

Fixed Vgs (always on)

current characteristics on slide18

## nMOS pulling high

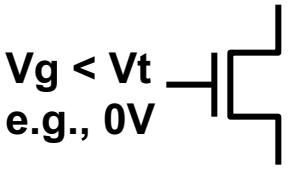



The voltage on this terminal will be Vg-Vt

Variable Vgs (Ids will be 0 when Vgs = Vt)

current characteristics on slide 20

## nMOS off



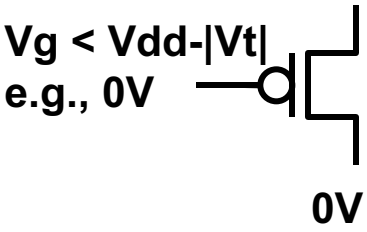

Since all voltages are positive, there is no way for this transistor to turn on

nMOS can pass min. 0 when it is ON

nMOS can pass max. Vg-Vt when it is ON

# More Accurate pMOS Model

## pMOS pulling low

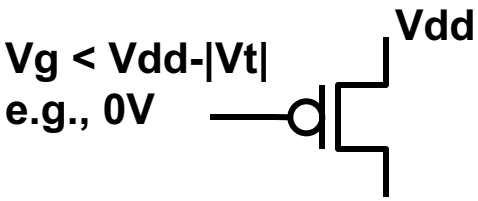



The voltage on this terminal will be Vg+|Vt|

Vt of a PMOS is negative
Variable Vgs

current characteristics on slide 21

## pMOS pulling high

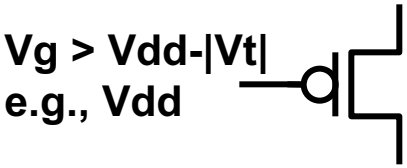



The voltage on this terminal will be Vdd

Fixed Vgs (always on)

current characteristics on slide 19

## pMOS off




For max voltage of Vdd, there is no way for this transistor to turn on

pMOS can pass min. Vg+|Vt| when it is ON

pMOS can pass max. Vdd when it is ON

# Two Options

- Either remember more complex model
  - And the voltage drops through transistors
- Or only route Gnd through nMOS
  - And Vdd through pMOS
- In this class (and most CMOS design today)
  - We will take the easier approach
  - Assume nMOS only routes Gnd, and pMOS Vdd
- What happens if you want to route a signal (could be either)
  - Need a real switch!
  - Put both (nMOS and pMOS) in parallel

# - Why Buf/AND/OR Don't Work

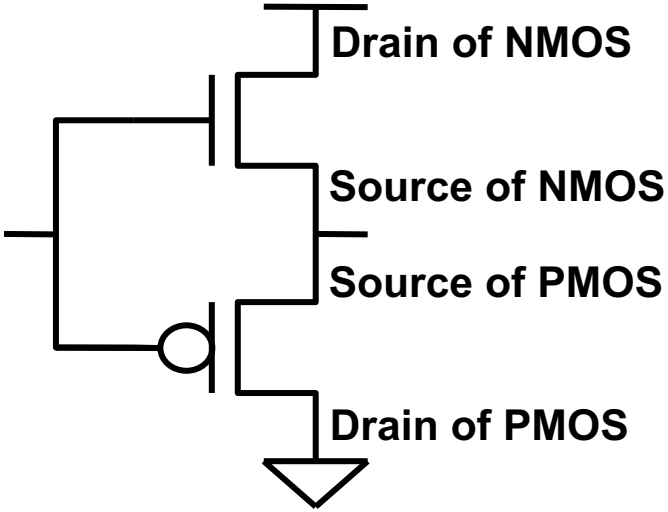
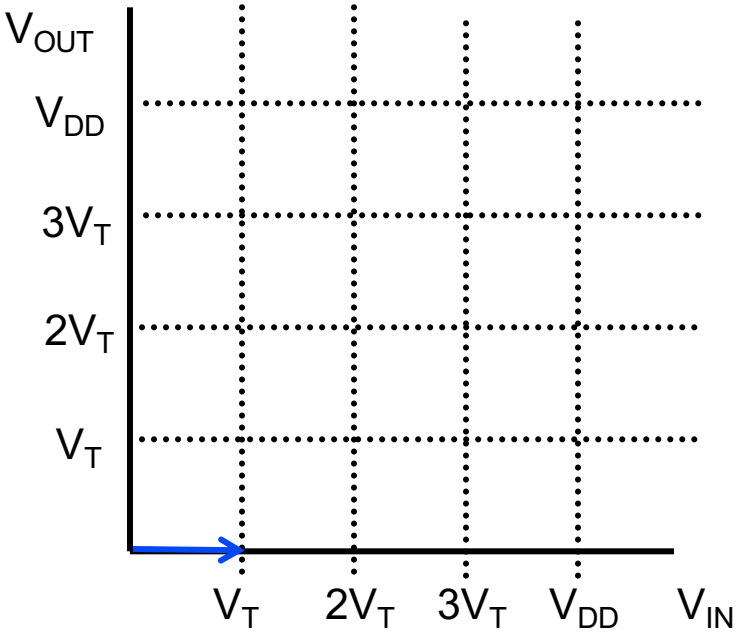



**Figure assumes $V_{TN}=-V_{TP}=0.25V_{DD}$**

- Let's write down source & drain for each transistor

# - Why Buf/AND/OR Don't Work



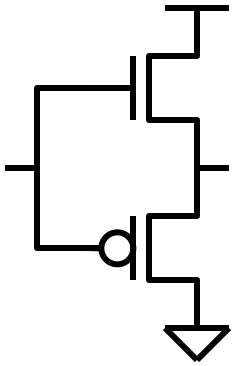
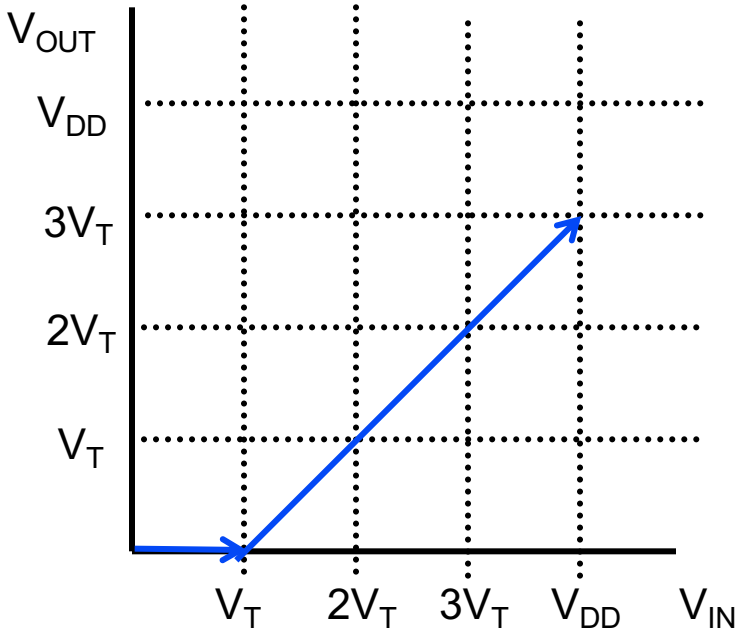



**Figure assumes $V_{TN}$=-$V_{TP}$=0.25$V_{DD}$**

- Suppose IN = 0V → Can NMOS turn on?
  - Depends on PREVIOUS value at output (previous output state)
  - Suppose previous output state was 0 (pls. try other cases yourself)
  - Nobody turns on UNTIL Vin = Vtn

# - Why Buf/AND/OR Don't Work



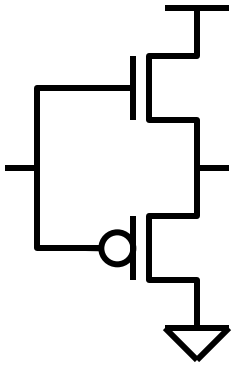
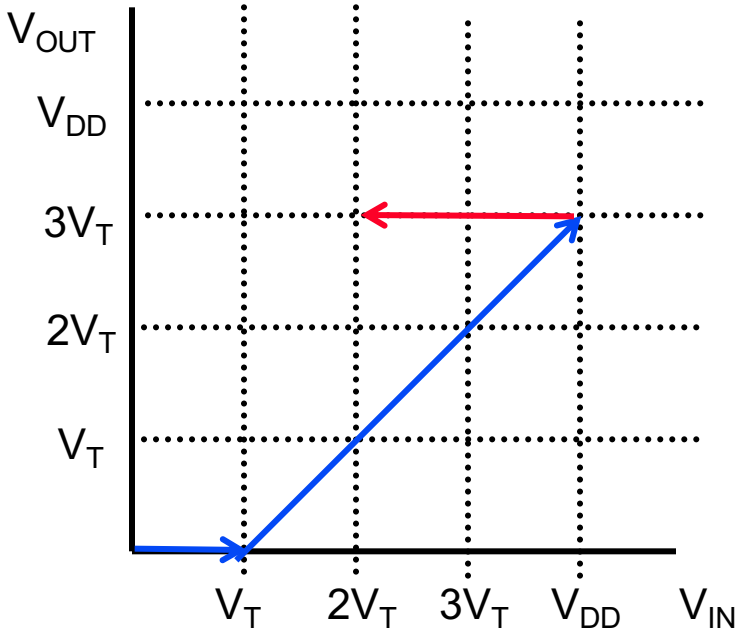



**Figure assumes $V_{TN}$=-$V_{TP}$=0.25$V_{DD}$**

- Suppose Vin goes up now
  - Vout continues to go up
    - NMOS pulls up to Vg - Vt
  - When Vin = Vdd (=4Vt), Vout = 3Vt (= Vdd – Vt)

# - Why Buf/AND/OR Don't Work



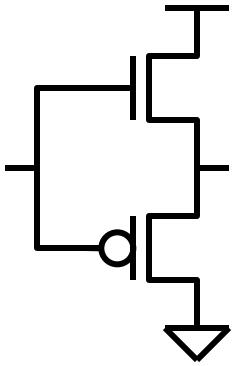
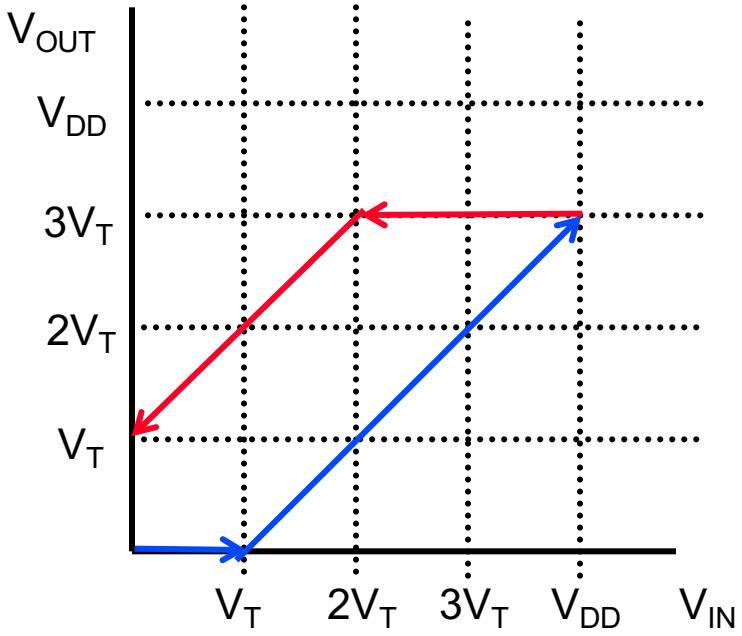



**Figure assumes $V_{TN}$=-$V_{TP}$=0.25$V_{DD}$**

- Suppose Vin goes down now
  - NMOS turns off, PMOS is already off (all this time)
  - Output continues to stay at 3Vt
  - How long does PMOS stay off? – until Vin = 2Vt

# - Why Buf/AND/OR Don't Work



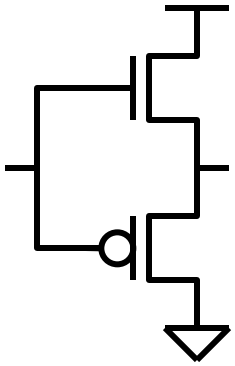
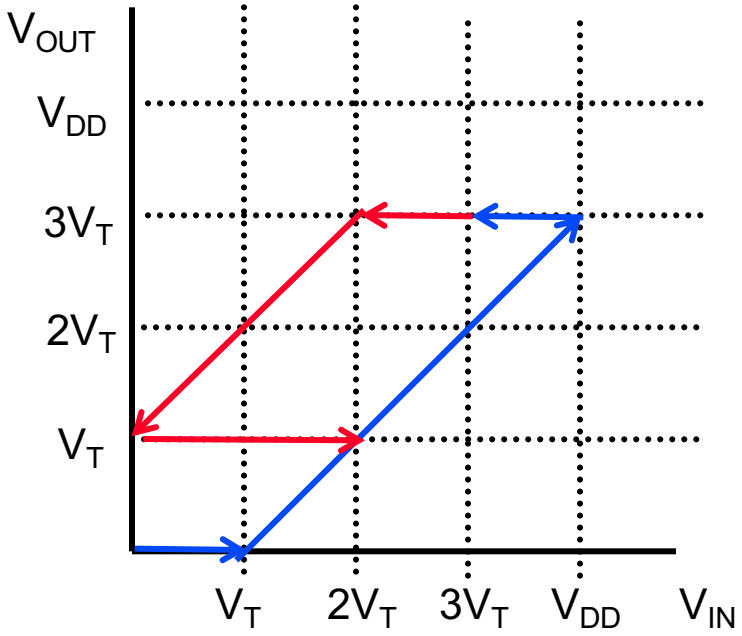



**Figure assumes $V_{TN}$=-$V_{TP}$=0.25$V_{DD}$**

- Vin < 2Vt
  - PMOS turns on – output continues to go down
    - PMOS pulls down to Vg + |Vtp|
  - How long does PMOS stay on? – until Vin = 0

# - Why Buf/AND/OR Don't Work



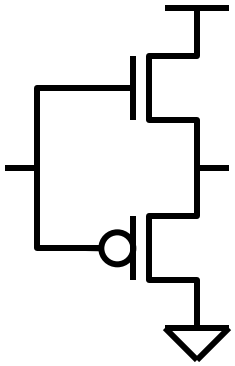
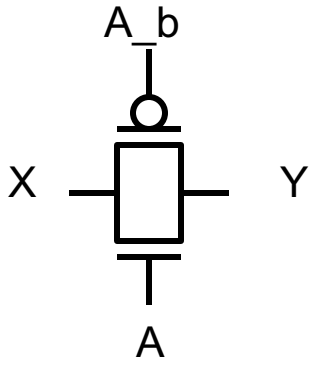



**Figure assumes $V_{TN}$=-$V_{TP}$=0.25$V_{DD}$**

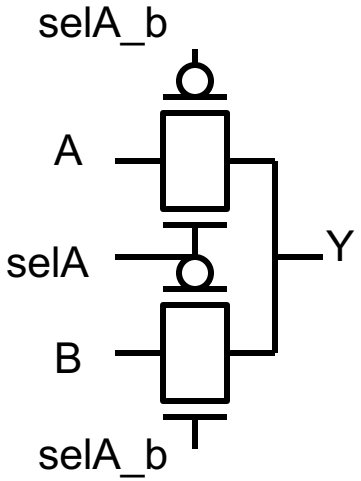
- Suppose Vin starts increasing again
  - PMOS is off, NMOS is off
  - When does NMOS turn on? Vin = 2Vt
  - What happens after then? Same as before

# + Beyond Gates: Pass Transistor Structures




- Concept of switch is really versatile
  - But if we don't know the value being switched, neither NMOS nor PMOS alone implements it
- Solution: use nMOS and pMOS in parallel
  - Drive gates with complementary signals
  - Completely bidirectional
- How can we take advantage of this?

# + Transmission Gate Muxes

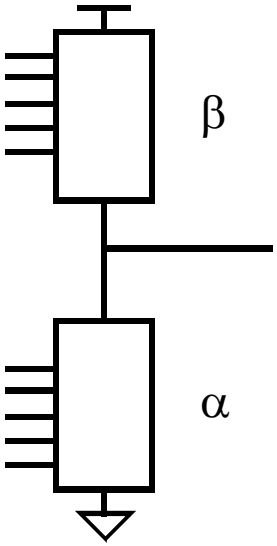



- Arbitrary number of inputs can be muxed together
  - As long as selects are mutually exclusive
- Can be cascaded in series to make more complicated networks
- CAREFUL: Not restoring as drawn (no gain)
  - Inverter often used to buffer output

# Review: Gates Design

- To build a logic gate f'(x1, …, xn),

The pullup network connects the output to Vdd when f is false.




pMOS only, since only passes 1

The pulldown network connects the output to Gnd when f is true.

nMOS only, since only passes 0

- Pulldown
  - $\alpha(x1, \ldots, xn) = f(x1, \ldots, xn)$
- Pullup
  - $\beta(x1', \ldots, xn') = f'(x1, \ldots, xn)$ (pMOS invert inputs)

# Review: DeMorgan and Duals

- The pullup and pulldown switch networks are complements
- By DeMorgan:
  - f'(x1, …, xn) = DUAL { f }(x1', …, xn')
  - pMOS invert inputs
- $\alpha$(x1, …, xn) is dual of $\beta$(x1, …, xn)
- Good news / Bad news
  - Good news – it is easy to create dual network
  - Bad news – either pullup or pulldown has series devices
- We will explain why series devices are bad soon …

# Power and Delay: A Gate's Metrics

- When we use a gate we care about its logic function
  - That is the desired output
- It also consumes "resources" that we care about
  - Delay
    - The output becomes valid some time after inputs settle
  - Power
    - The gate also consumes some energy from the power supply
  - Area
    - This is often less important in today's chips
    - The wires needed to connect the gates takes the most space
- These "charges" are not difficult to estimate
  - But you will need to estimate them to be a good designer

# R and C is All You Need

- Delay can be estimated by simple RC models
- Power can be estimated (mostly) from C alone
- But to do either, we need to have a R, C model of gates
  - And the wires
- But first for those CS types in the class
  - And those EEs who were asleep in their circuit classes

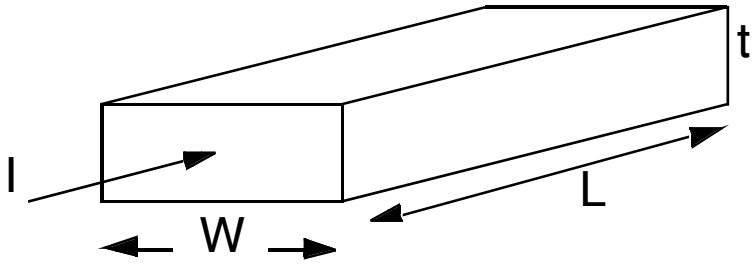
# Resistance? Capacitance?

- Resistance
  - Relates current to voltage (V = IR)
  - Measures how easy it is for current to flow
  - (Actually weird – relates a force qV/L to a velocity)

- Capacitance
  - Relates charge to voltage (Q = CV)
  - Exists between any two conductors
  - Causes delay in circuits ($\tau$ = RC) and data storage (memory)
  - Causes energy consumption
    - But does not consume any energy

# Resistance

- Resistance of a conductor
  - Resistivity $\rho$ * Length/Area
  - Designer does not control $\rho$, t
  - Generally deal with $\rho$/t
  - Called ohm/square (Rsq)

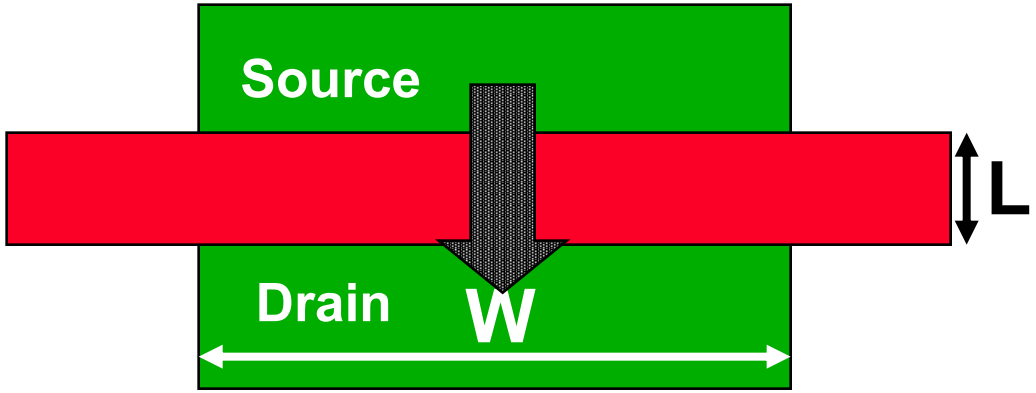


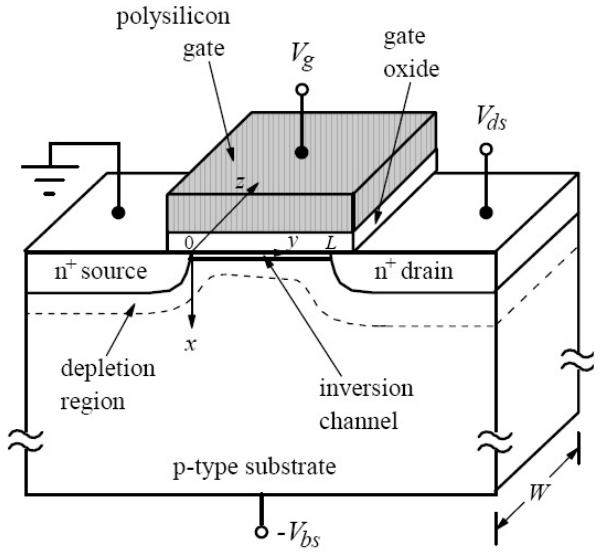

$$R = \frac{\rho L}{tW} = \frac{\rho}{t}\frac{L}{W}$$

- That's why LONG wires have higher resistances

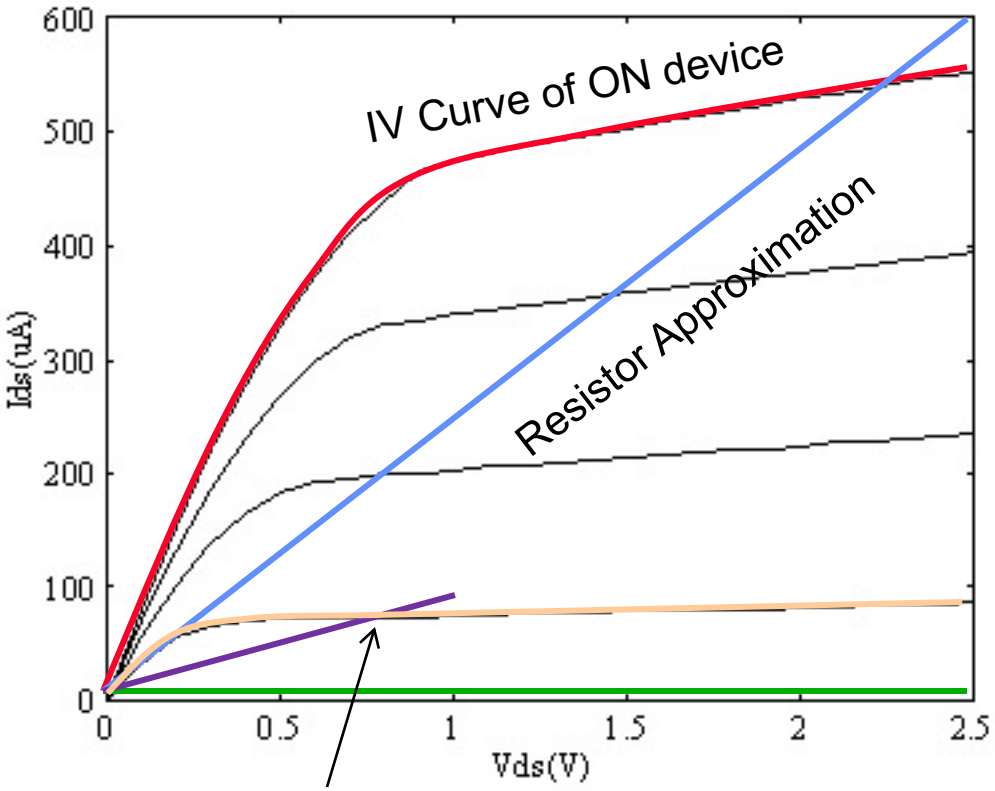
# Transistor Resistance

- For transistors
  - Designer chooses
    - W and L
  - Wider transistor
    - More current
    - Lower R
  - R is a surrogate for
    - 1/Ids
    - R depends on Vdd
  - R increases as Vdd decreases



# + Our Switched Resistor Model




The orange and purple line show the ids and effective resistance if Vdd was only 1V for this transistor. The resistance is much higher.

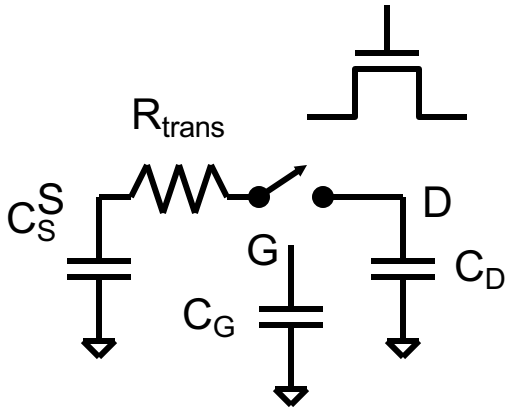
With digital input on gate, device is either ON or OFF

Approximate ON device with resistor (blue line)

$R_{trans}$ = L/**W** x **Constant dependent on technology**

Since L is generally min, just give W

**You can think in terms of current as well (proportional to W)**



# Resistance: nMOS vs. pMOS

- For same geometry, i.e., same L & W
  - PMOS and NMOS Ids values differ: hole vs. electron mobility
  - We will go by: electron mobility = 2x hole mobility
  - i.e., nMOS Ids = 2x pMOS Ids

- Talk about the resistance / square for a transistor
  - For a transistor of length L and width W
  - $R_{trans,\ p} = R_{sq,\ p}$ x L/W
  - $R_{trans,\ n} = R_{sq,\ n}$ x L/W

# Rules of Thumb for Resistance

| Transistor Resistance | Resistance /sq | |
|---|---|---|
| | 1μ | 45nm |
| nMOS | 18kΩ | 12kΩ |
| pMOS | 36kΩ | 26kΩ |

Note that the resistance/sq does not change much with scaling. It used to be constant, but now is scaling down because of transistor magic (strain, new materials) to make devices faster.

| Wire Resistance | Resistance/sq | | Resistance per mm | |
|---|---|---|---|---|
| | 1μ | 45nm | 1μ | 45nm |
| Metal 1 | 0.05Ω | 0.3Ω | 25Ω | 3KΩ |
| Top Metal | 0.03Ω | 0.03Ω | 15Ω | 60Ω |

The min width wire in the 1μ tech is 2μ.

The min width wire in 45nm is 0.1μ for m1 and 0.5μ for top metal

# Why Worry About Wire Resistance?

- Transistor resistance (per sq) are much larger
  - By many orders of magnitude
- Historically one could ignore wire resistance
  - And for short wires one still can
- Problem is that wire length in sq is growing with scaling
  - At least for the long wires
  - Transistors need to be large to drive these wires
    - Large transistors, is large W
    - Implies a very small number of squares
  - So wire resistance can dominate!

# Capacitance and Delay

- Capacitors store charge

  Q = CV <- charge is proportional to the voltage on a node
- This equation can be put in a more useful form

$$i = \frac{dQ}{dt} \Rightarrow i = C\frac{dV}{dt} \Rightarrow \frac{C\Delta V}{i} = \Delta t$$

- So to change the value of node (from 0 to 1 for example), the transistor or gate that is driving that node must charge (up, in our example) the capacitance associated with that node. The larger the capacitance, the larger the required charge, and the longer it will take to switch the node.
- Define $R_{trans}$ so that the current (i) is approximately $V/R_{trans}$

$$\Delta t = \frac{C\Delta V}{i} = \frac{C\Delta V}{V/R} = R_{trans}C$$

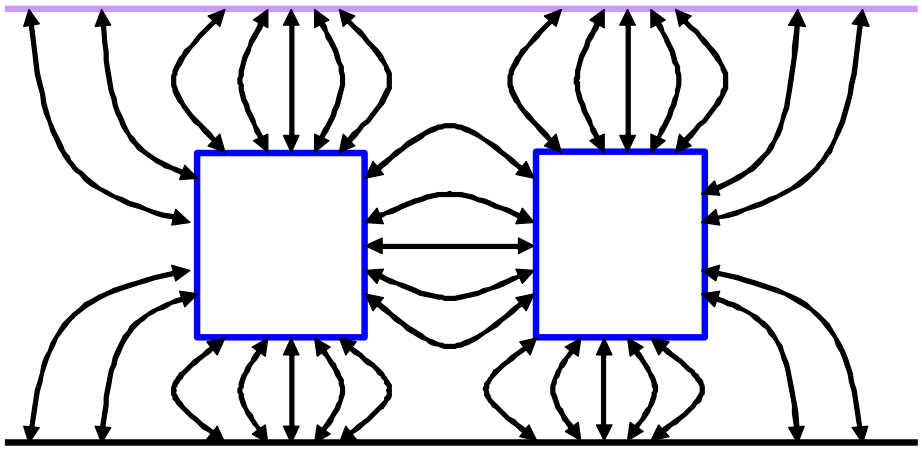
This is pretty high level, we are not worrying about small constant prefactors

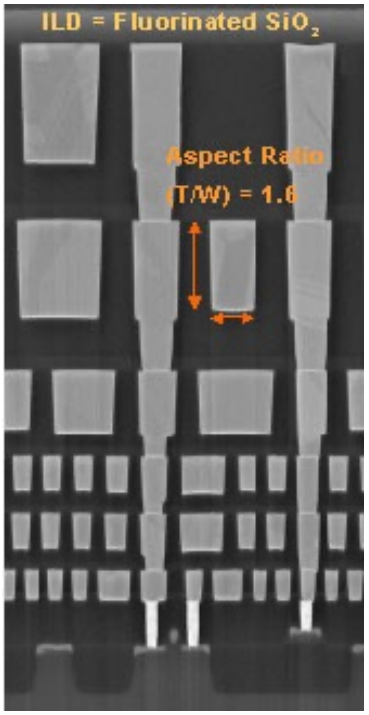
# Load Capacitance

- $C_{load}$ comes from three factors:
  1. Gate capacitance of driven transistors.
  2. Diffusion capacitance of source/drain connected to the wire.
  3. Wire capacitance

- Today, a 1$\mu$ technology is the really cheap technology that students use, and advanced processes are running at 0.014$\mu$.
- I use metrics that historically haven't changed much with technology scaling (Rsq, and Cap/$\mu$), but you should always find the correct numbers for the technology that you will use before starting a design. And, since you don't want to extract the $C_{load}$ numbers by hand, make sure that the CAD tools have the right numbers too.

# Real Wires






- Are not parallel plate capacitors.
  - Closest conductor is the neighboring wires
  - But capacitance still will be proportional to length

- Capacitance to neighboring wires is called coupling capacitance
  - Can inject noise (neighbor switches when you are quiet)
  - Can increase delay (neighbor is switching in opposite direction)

# Coupling Capacitance

- What happens if some of the capacitance is to another wire?
  - Need to think of the whole circuit
  - Remember capacitors have two terminals
- The equations remain the same
  - But you need to think about the voltage across the coupling cap
- You will do coupling example in HW 2

# Rules of Thumb for Capacitance

| Transistor Cap | Capacitance per μ of W | |
|---|---|---|
| | 1μ | 45nm |
| Cg - gate | 2.0 fF | 1.2fF |
| Cd - ndiff | 2.0 fF | 1.2fF |
| Cd - pdiff | 2.0 fF | 1.2fF |

Cinv = Input cap of a min sized inverter

4λ nMOS 8λ pMOS

| Wire Cap | Capacitance per μ | | Length when C=Cinv | |
|---|---|---|---|---|
| | 1μ | 45nm | 1μ | 45nm |
| Poly wiring | 0.2fF | 0.2fF | 60μ | 3μ |
| Metal 1 | 0.3fF | 0.3fF | 40μ | 2μ |

# What To Do With These R's and C's?

- Want to use information about Rs and Cs to optimize our chip
  - But how?
- Most important question when doing optimization …
  - What is your objective
- In VLSI we have 4 objectives:
  - Design time, which is related to probability of error
  - Power
  - Performance
  - Area
- Next lecture we will talk about these metrics
  - And how they relate to R, C