Lecture 8

Transistor Models

Computer Systems Laboratory
Stanford University
horowitz@stanford.edu

Introduction

- Readings (for next lecture on wires)
  - Arora Capacitance extraction from layout
    This is just background reading (read quickly)
  - Ho The Future of Wires
    This covers most of the material in the next lecture (and then some)

- Today's topics
  - Review of transistor models (quick review of EE313)
    - From the simple to the complex
    - How to "calibrate" a technology
    - How to use models to think about technologies and circuits
  - Examination of transistor variations
    - Local variations, or mismatch between pairs
    - Run-to-run variations

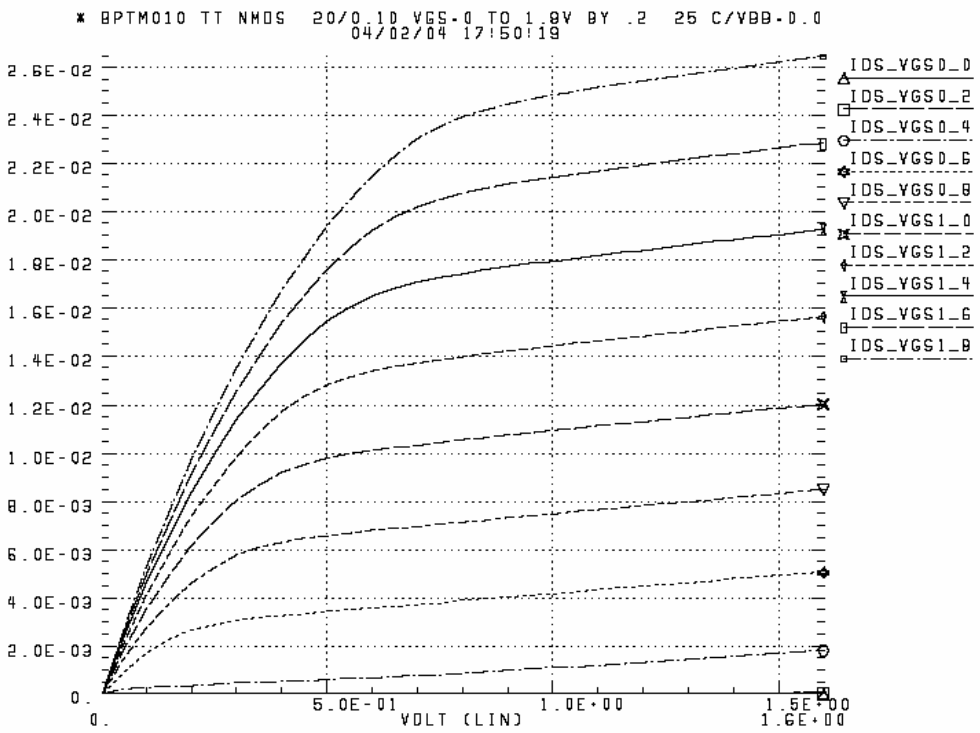
MOS Device Behavior

- Assume you know MOS device issues from EE313
  - We'll look at some I-V curves, review some important issues
  - Read Hodges & Jackson (EE313 text) if you need to

- For I-V curves we need to understand
  - Basic shapes of the I-V curves
  - Threshold voltage
  - Mobility effects and velocity saturation
  - Subthreshold conduction
  - Scaling
  - Variations in these parameters

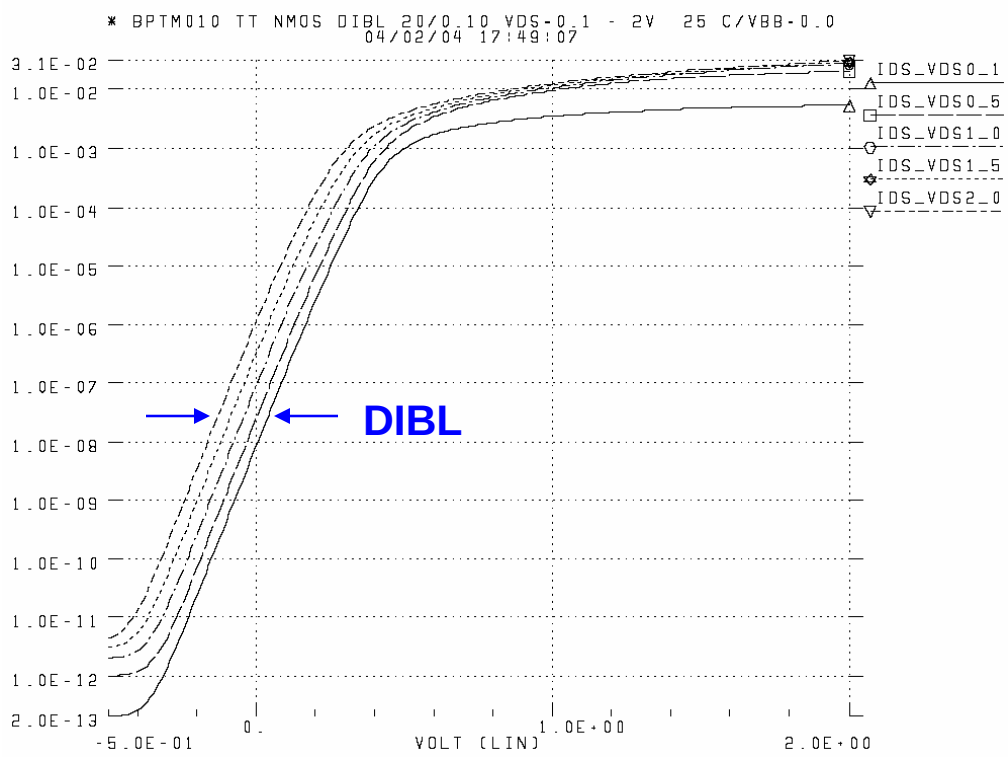
EE313 Review: Basic I-V Curves: I_{ds} versus V_{ds}

- Plot has two regions
  - Linear (low V_{ds})
  - Saturated (high V_{ds})
- Linear region
  - Looks like a resistor
- Saturated region
  - "Constant" current
  - $g_{ds} = 1/r_0$



EE313 Review: Basic I-V Curves: I_{ds} versus V_{gs}

- Two typical plots
  - Linear I_{ds}
    - For $V_{gs}>V_{th}$
    - Lots of current
    - Can get g_m
  - Log I_{ds}
    - For $V_{gs}<V_{th}$
    - Leakage current
    - Can get V_t, DIBL
- Measuring V_{th}
  - Extrapolate linearly
  - Beware of DIBL



E313 Review: Mobility

- Mobility (cm^2/Vsec) relates carrier drift velocity to lateral E-field
- Falls quickly as temperature rises

$$\mu = \mu_0 \cdot \left(\frac{T}{T_0}\right)^{-1.5}$$

  - As temp rises from 27° to 130°, current falls 0.65x
  - Circuit runs 1.6x slower

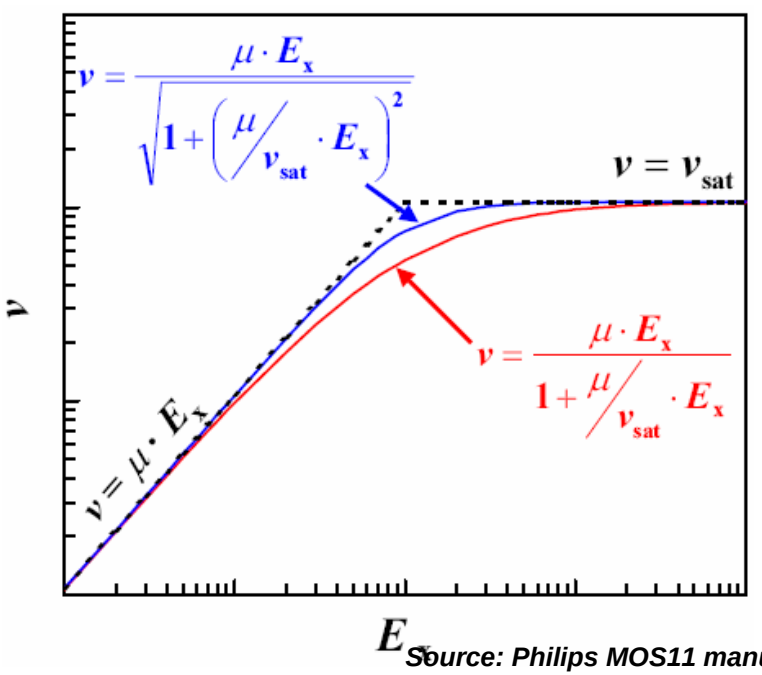
- Also decreases as vertical field increases (here, T_{ox} in nm)

$$\mu_n(V_{gs}, V_{th}, T_{ox}) = \frac{540}{1+\left(\frac{V_{gs}+V_{th}}{0.54T_{ox}}\right)^{1.85}}$$

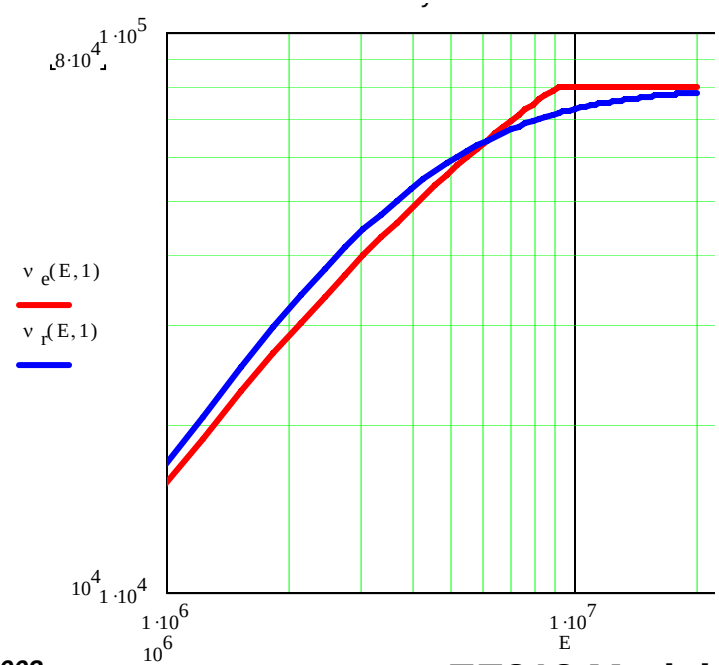
  - Why (V_{gs}+V_{th})? That's a strange term…
  - B/c E-field proportional to Q_b+0.5Q_{inv} = $C_{ox}V_{th}$+0.5C_{ox}(V_{gs}-V_{th}); see Chen

EE313 Review: Velocity Saturation

- Carrier velocity and E-field relationship is not always linear
  - Saturates out; max velocity around 8×10^6 cm/s



Source: Philips MOS11 manual, 2003



EE313 Model is red

- Critical E-field (velocity is ½ down) is about 4V/μm

EE313 Review: Velocity Saturated Current

- Drain current is worse when carrier velocity saturates

$$i_{dsat} = W v_{sat} C_{ox} \frac{(V_{gs} - V_{th})^2}{V_{gs} - V_{th} + \frac{2 v_{sat} L}{\mu_{eff}}}$$

$E_{crit} \cdot L$

- Look at both limits: $(V_{gs} - V_{th})$? $(E_{crit} \cdot L)$
  - When not saturated
  - When saturated

EE313 Review: Subthreshold Conduction

- The threshold voltage V_{th} is not a magical place
  - It's just where the channel charge is roughly equal to the doping
  - Device still has channel charge when $V_{gs} < V_{th}$

- What happens in subthreshold?
  - Gate voltage directly controls Φ_s, not channel charge
  - Channel charge exponentially related to Φ_s
  - Looks like a BJT

- Current is exponential with V_{gs}: $i_{ds} = I_s \cdot e^{\frac{V_{gs}-V_{th}}{\alpha V_t}}$
  - $V_t = kT/q = 26mV$ @ room temperature
  - I_s depends on definition of V_{th}, around 0.3μA/μm
  - α comes from cap voltage divider (C_{ox} and C_{depl}), around 1.3-1.5

Predicting Scaled MOS Device Performance

- Shockley quadratic model estimates scaling effects poorly
  - A better model (up until 90nm):

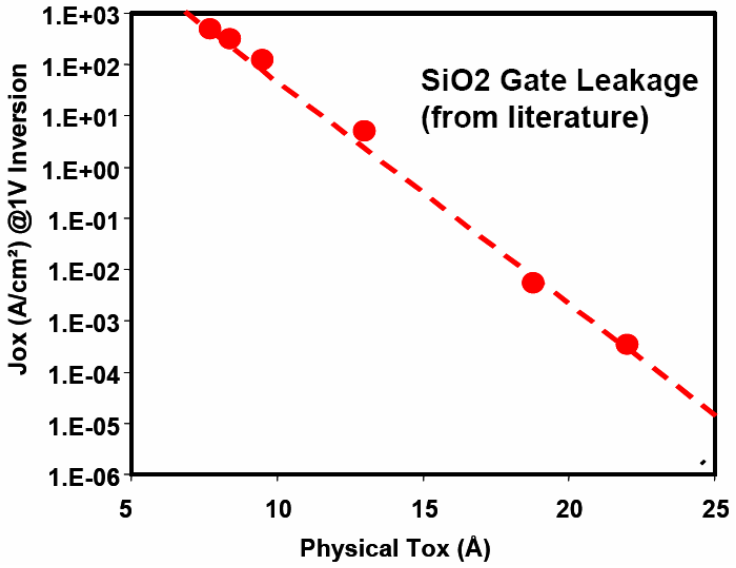
$$I_{dsat} = K \cdot W \cdot L_{eff}^{-0.5} \cdot T_{ox}^{-0.8}(V_{gs} - V_{th})^{1.25}$$

- Scaling example: Assume L, T_{ox}, and V_{gs} all scale by α
  - Current (per micron) will remain constant (0.5-0.8 mA/μm)
    - Current of the scaled transistor scales down by α
  - Voltage scales down by α
  - Capacitance scales down by α
  - So delay scales down, too: $\Delta t = CV/i = \alpha\Delta t$

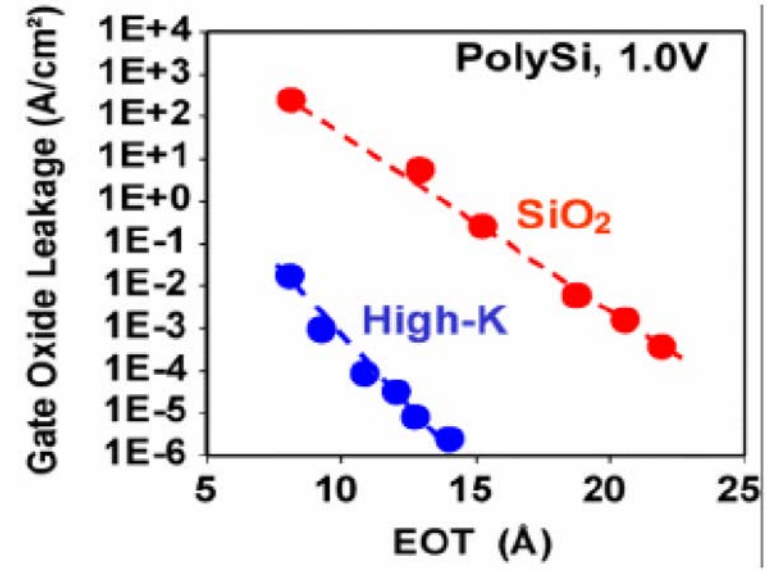
- Sub 90nm, this model breaks
  - Vth is not scaling, so Vdd does not scale …

Other Currents to Consider – I_g

- Also can look at I_g, gate tunneling current
  - Increasing as oxide thicknesses continue to shrink
  - T_{ox} 2nm today (130nm process); research lines at 0.8nm (30nm)
  - This is limiting gate oxide scaling in modern devices



Source: Marcyk, Intel, 2002

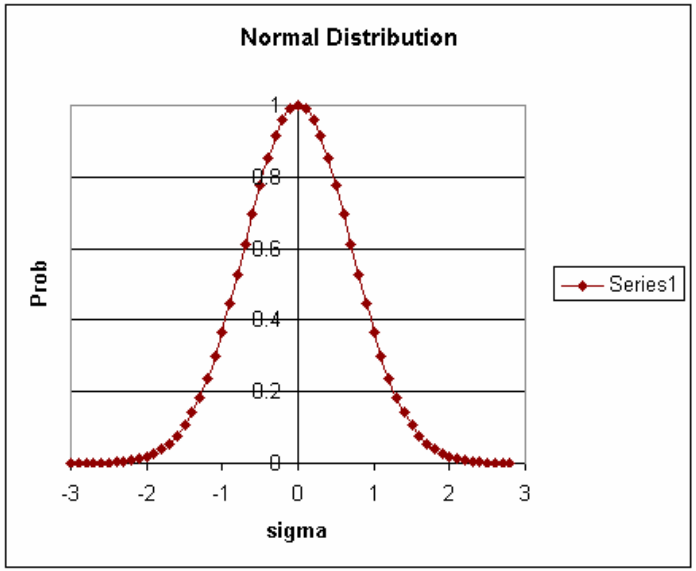


B. Doyle et al, *Intel Technology Journal*, vol. 6, issue 2, p. 42 (2002).

- Often not well modeled in SPICE; talk to your process engineers

Remember Parameter Variations

- No two transistors are exactly the same
  - They vary from wafer to wafer and from die to die
- Parameters of a fabrication run generally normally distributed
- Extract data from real wafers
  - 3-σ (or 4/5/6-σ) parameters
  - Use it in design



Parameter Variations

Variations come from many sources

1. Die to die variations
   - All devices in the die are correlated
   - Processing for this die/wafer varies from die to die and run to run

2. Across die variations
   - Two transistors on die have different parameters
   - Caused by many layout proximity effects
   - Across die processing variations

3. Random variations
   - Random dopant fluctuations, line edge roughness

1 used to dominate, but with scaling 2 and 3 are comparable issues

EE371 Corners

- We write our corners with a 3-letter code
  - nMOS and pMOS can each be **S**low, **T**ypical, **F**ast
  - V_{dd} can be low (**S**low devices), **T**ypical, or high (**F**ast devices)
  - Temp can be cold (**F**ast devices), **T**ypical, or hot (**S**low devices)

- Example: TTSS corner
  - Typical nMOS
  - Typical pMOS
  - Slow voltage = Low V_{dd}
    - Say, 10% below nominal
  - Slow temperature = Hot
    - Say, 100^{o} C → junction temperature

Which Corners Matter?

- Really depends on the circuits you are simulating
  - And what you want your die yield to be

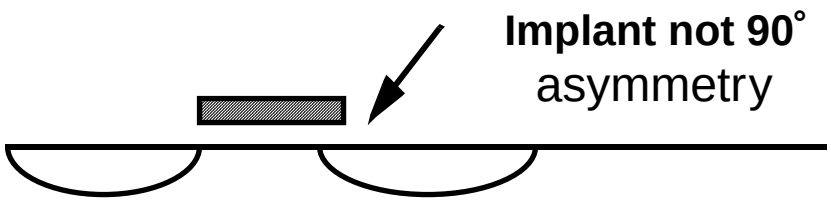
- Some important corners
  - TTSS: Must hit the timing specification here
    - Since this might be how it is used in a system
    - Will mean 50% performance yield loss (1/2 distribution will fail)
  - SSSS: Sometimes need to hit the timing spec here, too
    - Also worry about signals collapsing from slow risetimes
  - FFFF: See how much power your circuit burns
    - Also worry about narrow pulses disappearing
  - SFSS: Does pMOS-ratioed logic work? Race conditions
  - FSSS: Does nMOS-ratioed logic work? Race conditions
  - And so on…

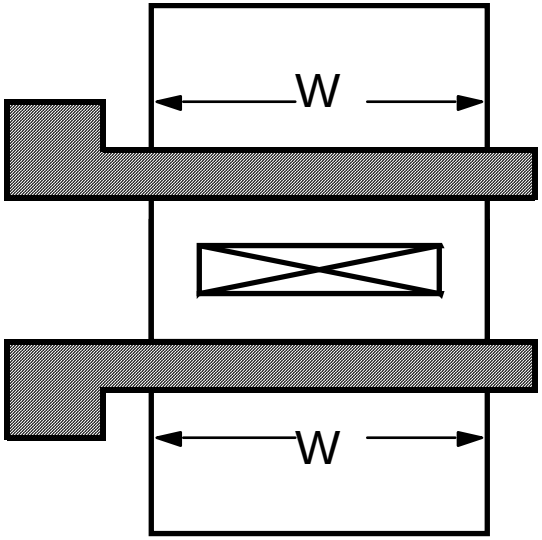
A Caution About Matching

- If your circuit depends on matching
  - Either in an analog component (like a sense amplifier)
  - Or a digital component like matched delays
- Simulation is much more difficult
  - Need to simulate the difference in the matched elements
  - Corner files don't do this, since they modify all transistors the same
- Need to do Monte Carlo simulations
  - This is where you do many simulations
  - Computer chooses random parameters for the transistors
    - You need to provide these models
  - Then you need to compute Mean / Sigma of circuit

Providing Matching Statistics

- If you want two transistors to match you need to be very careful
  - Almost anything will make them different
- In SPICE all transistors match perfectly
  - You need to add mismatch explicitly
  - Process corners do not help here
- Orientation matters

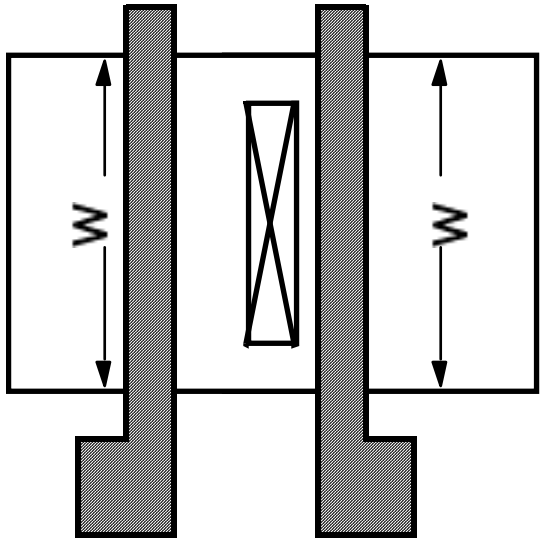




These transistors will not match

More on Matching

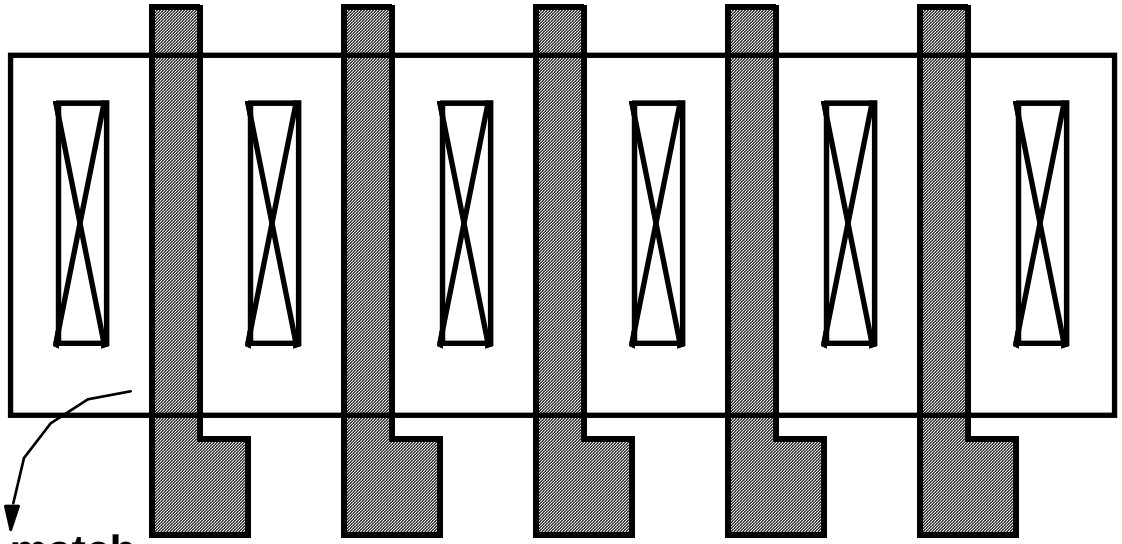
- Poly alignment is important



- Here, diffusion resistance and diffusion cap will not match
- Make currents flow in the same direction in matched devices
- Easy if all the transistors are folded

Even More Matching

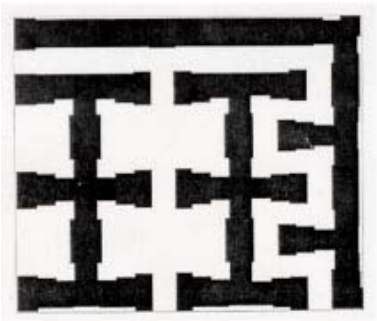
- Poly width control depends on local environment

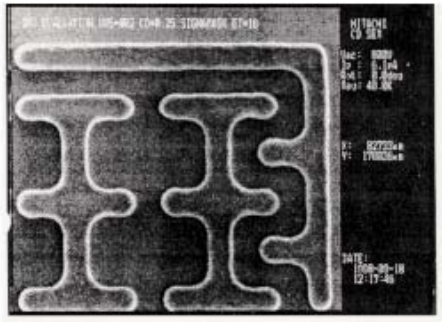


- Poly density affects etch rates, so end devices will be different
- To match transistors, add dummy devices
  - SRAMs often use entire dummy rows and dummy columns
- Modern technology need many dummies!

Welcome To Modern Technology

- Feature size is below the wavelength of lithography light
  - Hard to get sharp edges, so preprocess to add serifs





OPC = optical proximity correction

RET = resolution enhancement tech

- Variation is getting larger → foundries imposing rules
  - All transistors must be vertical
  - Poly edges must be far from diffusion

- Moving toward regular arrays of transistors
  - Looks similar to old gate array designs

Statistical Matching

- The errors we have been talking about are systematic
  - You can (in theory) make them zero
  - And you generally can figure out what happened
- But fundamentally even if you do everything right
  - There will still be some random mismatches between transistors
  - These are caused by random doping variations in the device
  - And small random variations in the etching process
- These effects can be models by adding an uncertainty to
  - V_{th}
  - K, or β, the current prefactor in the current equation

Statistical Matching

- Read Pelgrom's paper (and Lovett's paper)
  - It is the classic paper in this area

- His equations are still being used today
  - Data indicates that the matching depends on the area of the device
  - V_{th} standard deviation (T_{ox} in μm)

$$\sigma(V_{th}) = \frac{0.6V \cdot T_{ox}}{\sqrt{L_{eff} W_{eff}}}$$

  - K (or β) mismatch is addition to variation from V_{th}

$$\sigma(\beta) = \frac{2\%}{\sqrt{L_{eff} W_{eff}}}$$

The $64 Question

How does one analyze circuits?

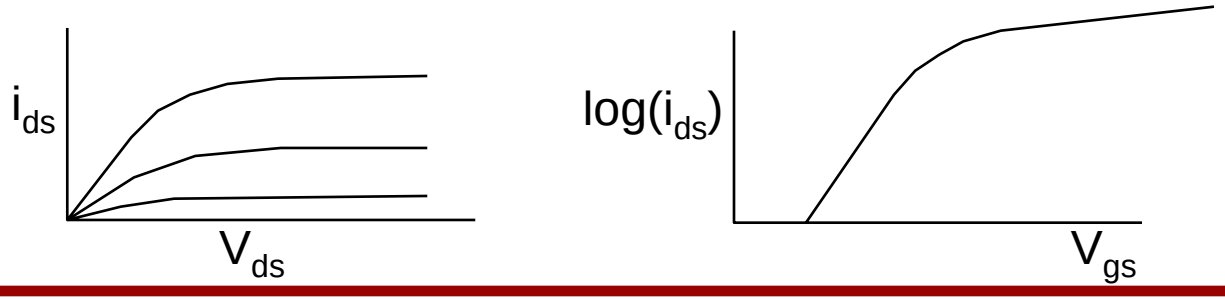
1. "Use your intuition and your pencil and paper analysis"
   - These are things that you understand
   - SPICE is prone to Garbage In / Very Pretty Garbage Out
   - You need to understand the circuit to check SPICE, and not vice versa

2. "Use SPICE"
   - VLSI circuitry has enormous complexity and ugly nonlinearity
   - Very difficult to do accurate hand analysis
   - Competitive market pushes sophisticated circuitry, which needs SPICE
   - Relying on hand analysis means you get steamrolled by your competitors

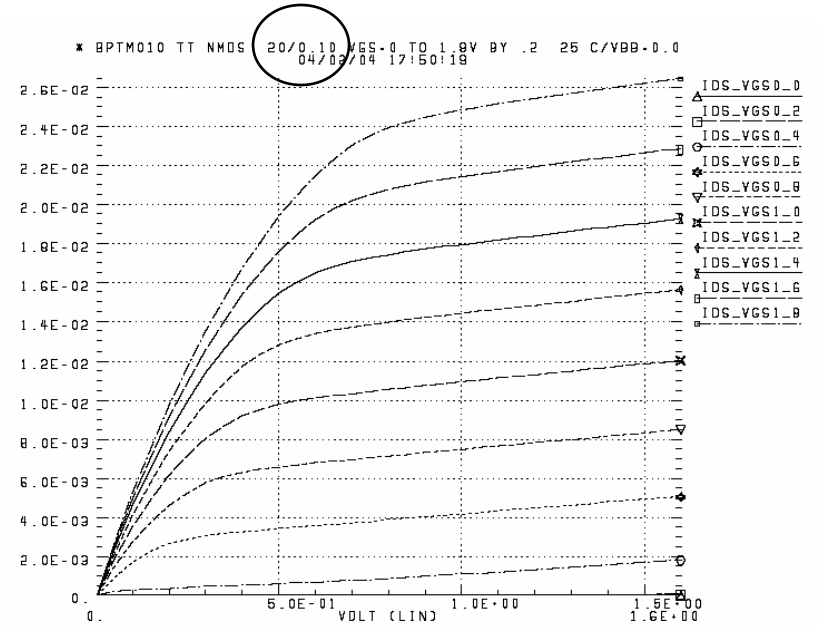
- Kernels of truth in both schools of thought
  - So you end up doing both

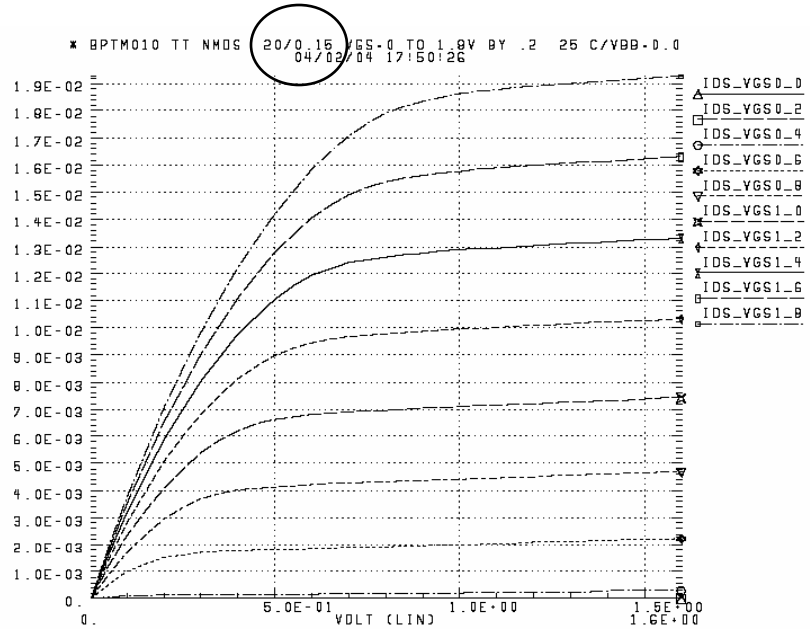
Calibrating a Technology

- What do you do when you get a new technology?
  - Run some simple simulations to get a feel for the transistor behavior
  - Generate some rules-of-thumb for reasoning about the circuits
- First look at the basic I-V curves
  - Examine a couple of different channel lengths
  - Do the curves look reasonable?
- What do they say about
  - Velocity saturation and output conduction?
  - V_{th}, V_{bb} sensitivity, and subthreshold conduction?
  - DIBL and V_{th} effects from W and L?



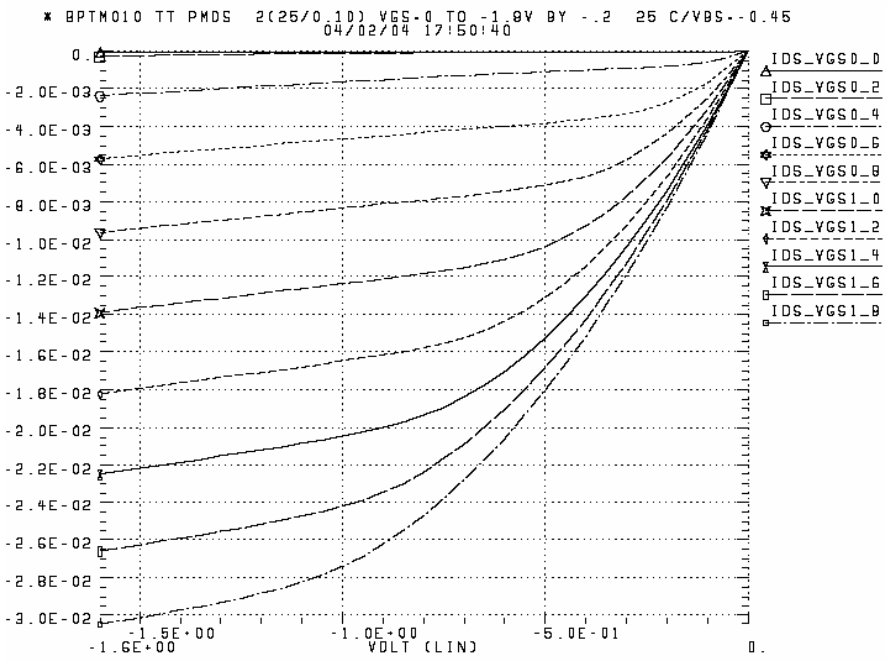
I_{ds} vs. V_{ds} (nMOS)

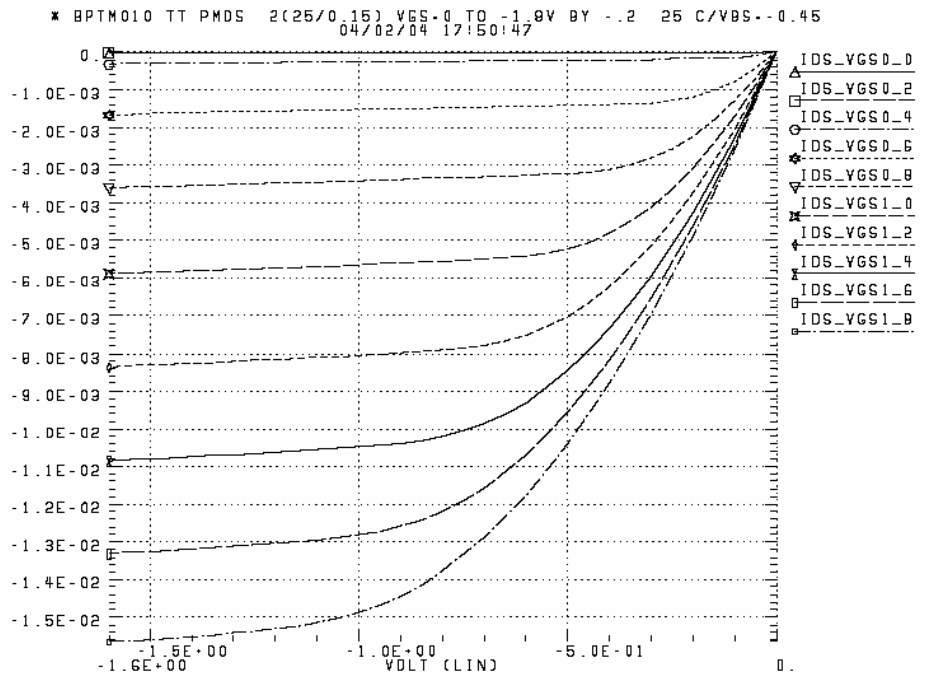




- Different channel length nMOS devices
  - Difference in output slope
  - Linear g_m in longer channel device

I_{ds} vs. V_{ds} (pMOS)

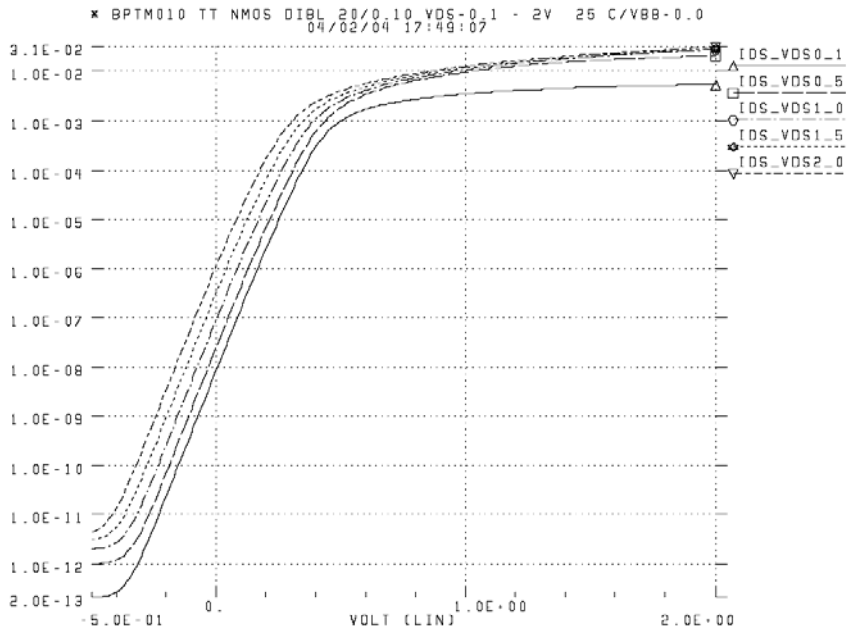




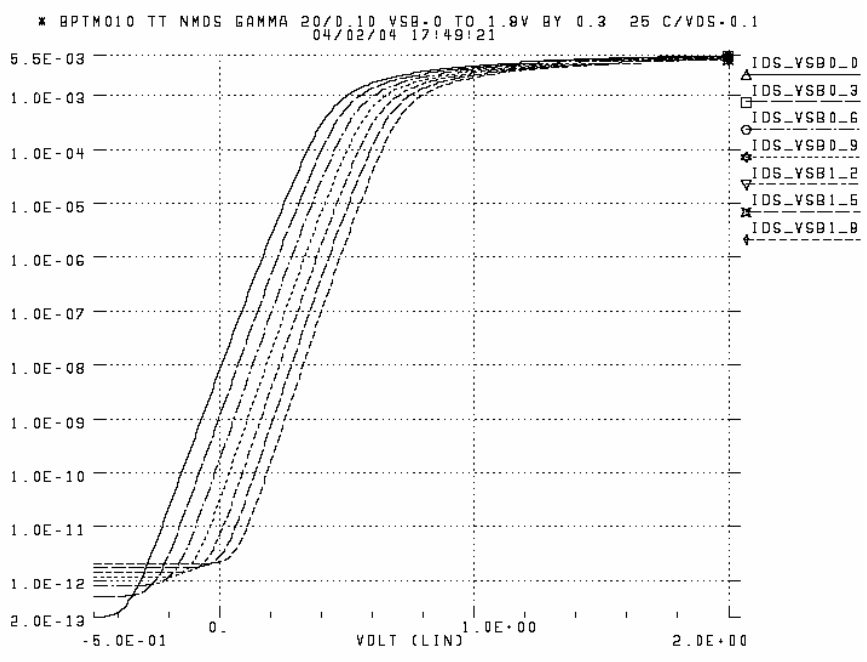
- Different channel length pMOS devices
  - Difference in saturation voltage from nMOS
  - Linear g_m in longer channel device, change in output slope

I_{ds} vs. V_{gs} (nMOS)

Sweep V_{ds}



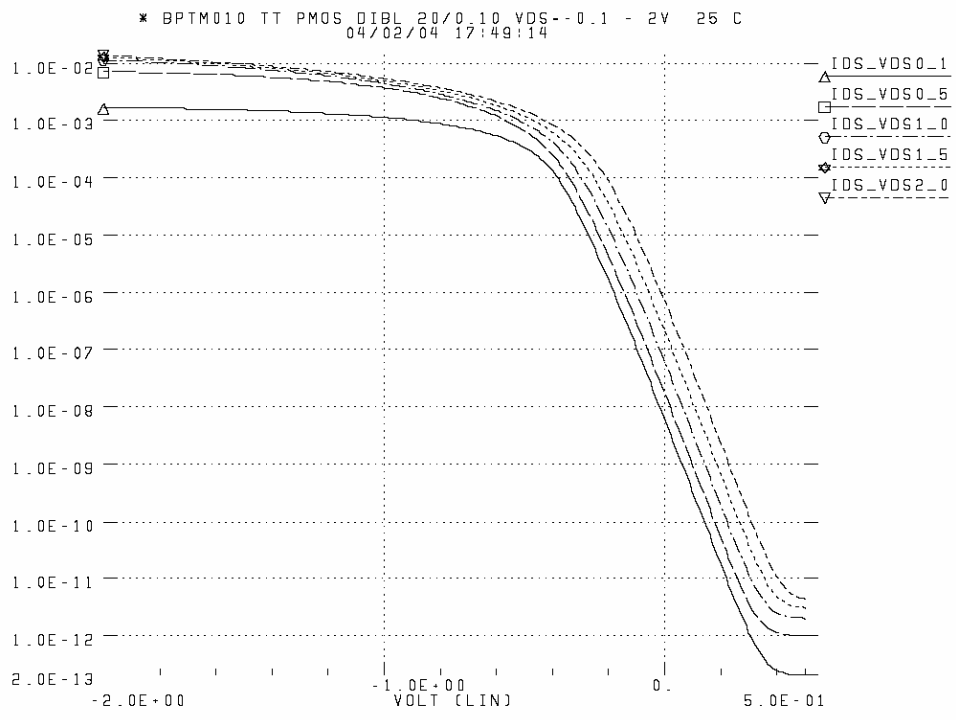
Sweep V_{bs}



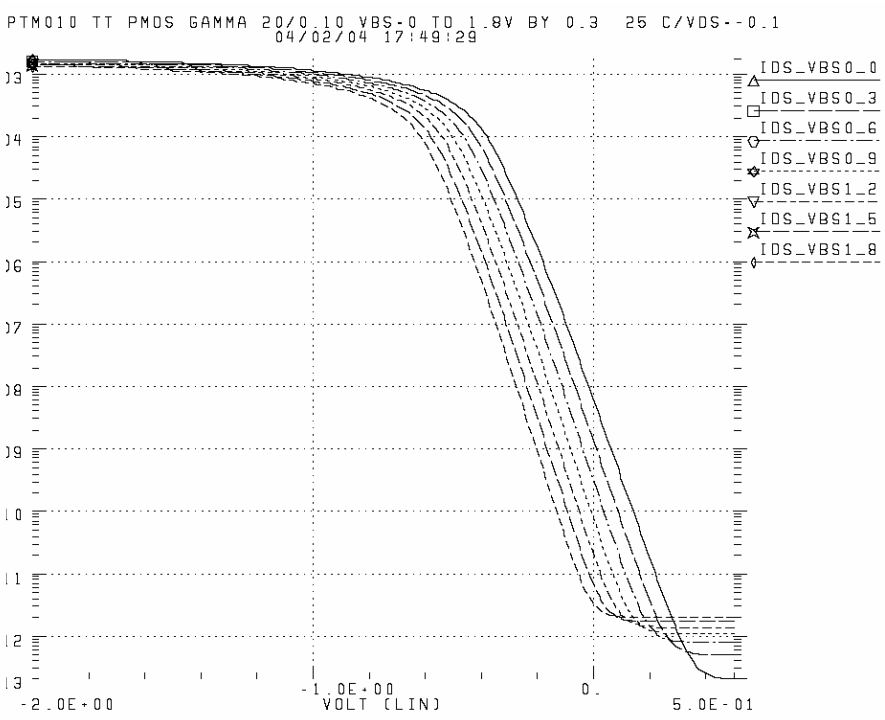
- V_{ds} plot → DIBL (drain-induced barrier lowering) $V_t = V_t - \eta V_{ds}$
- V_{bs} plot → γ (body effect) $V_t = V_t + \gamma\left(\sqrt{\phi_s - V_{bs}} - \sqrt{\phi_s}\right)$

I_{ds} vs. V_{gs} (pMOS)

Sweep V_{ds}

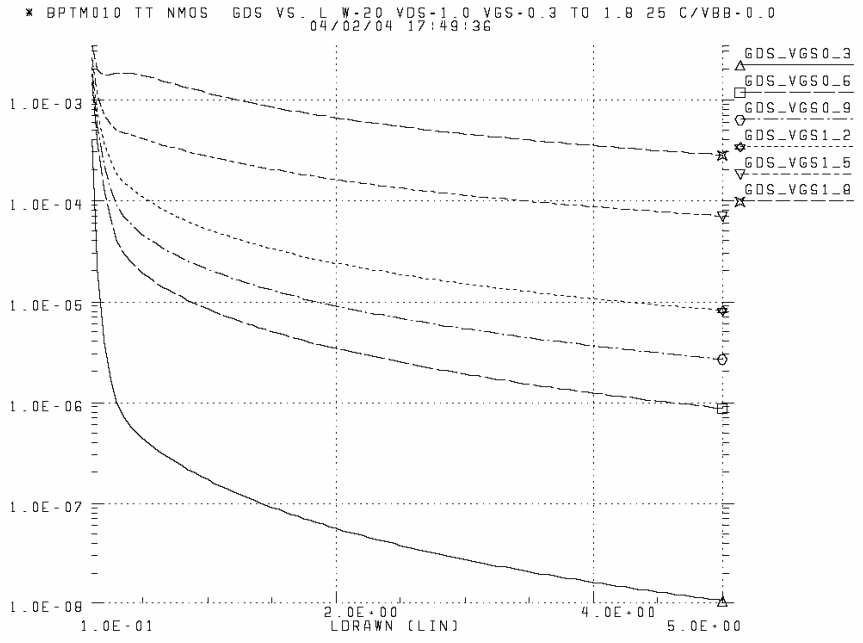


Sweep V_{bs}

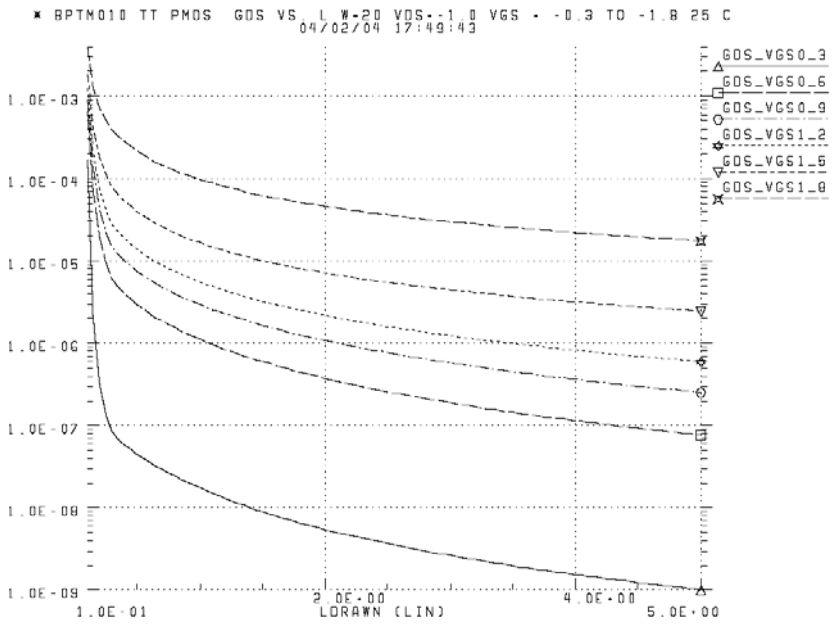


g_{ds} vs. L

nMOS



pMOS

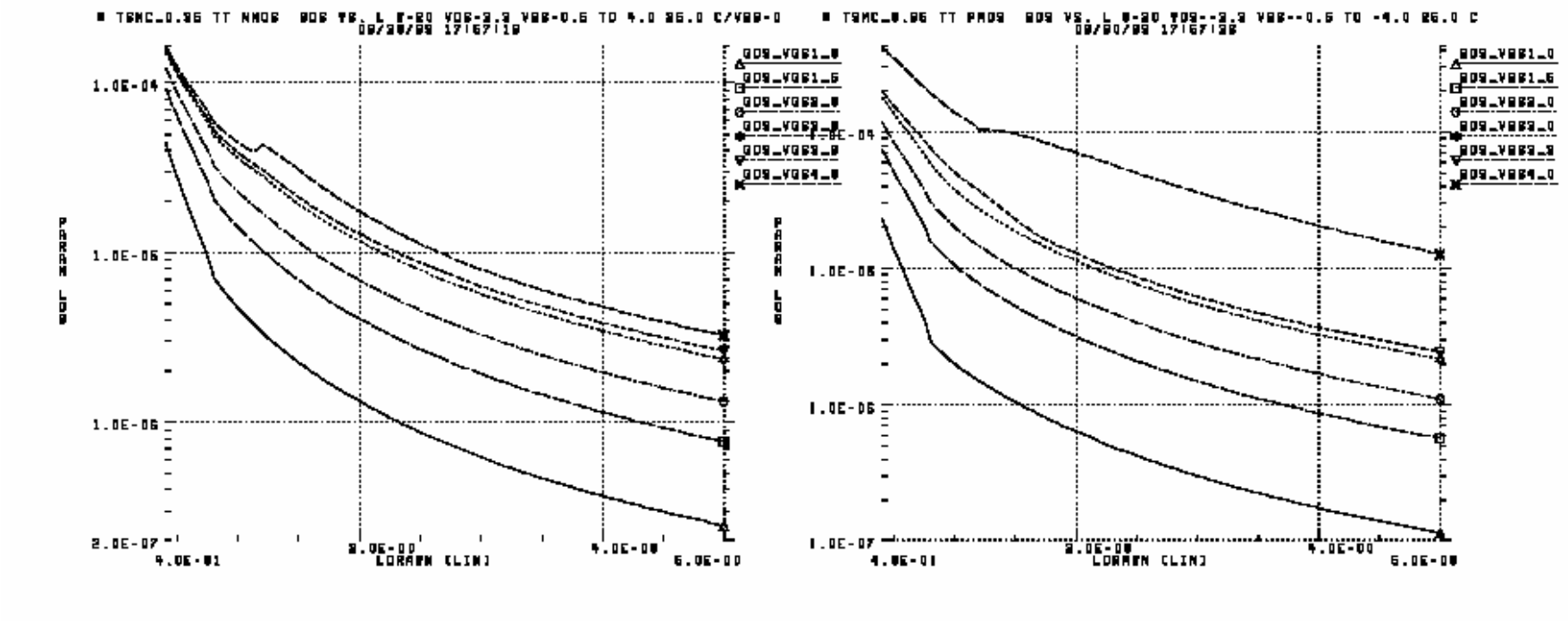


- Scale on sim run was wrong – Max L should be probably 1μ

Beware of Model Binning

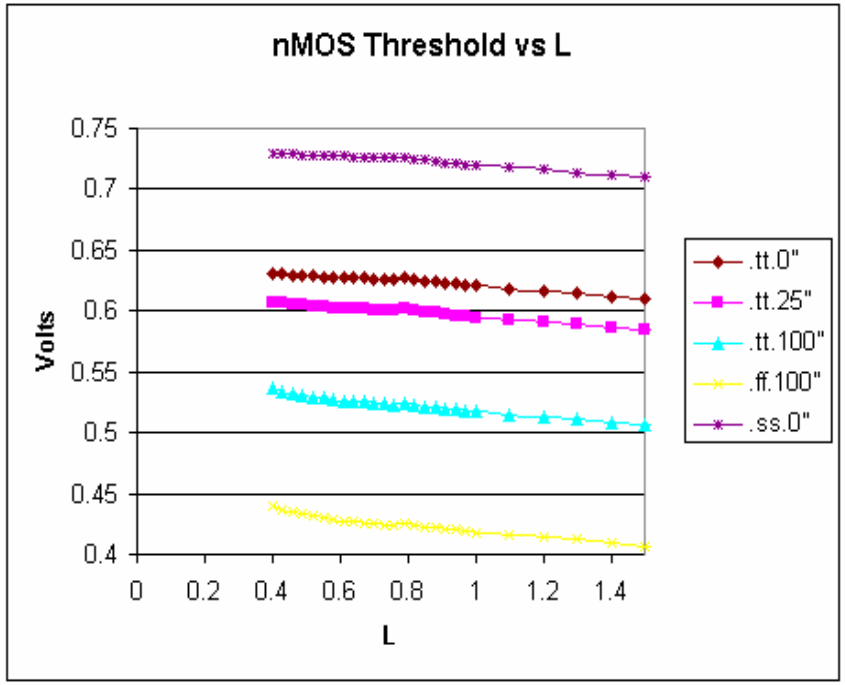
nMOS

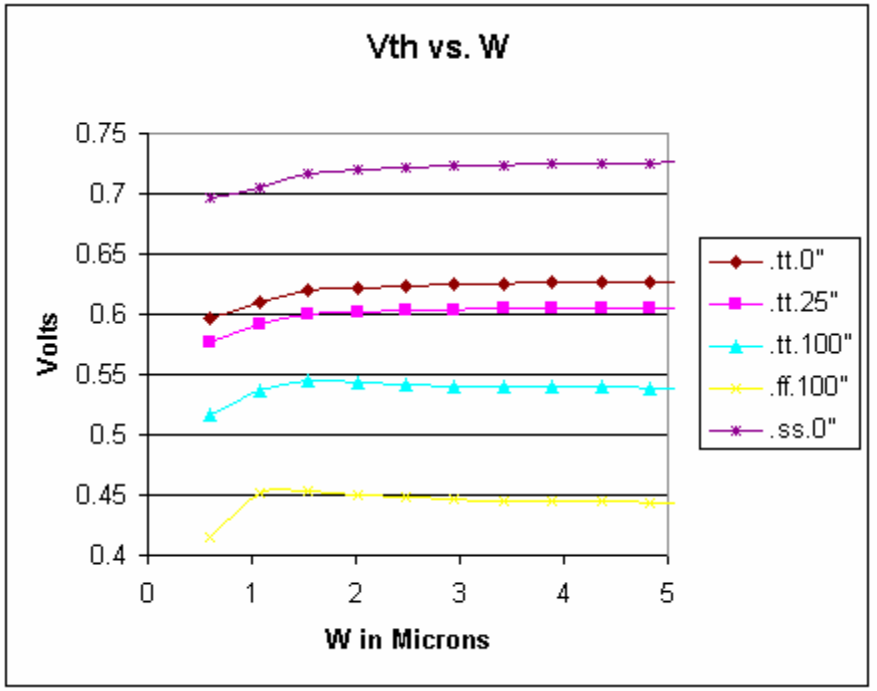
pMOS



- Plot of g_{ds} versus L for a 350nm technology
- Odd (un-natural) kinks as we move from size “bin” to size “bin”

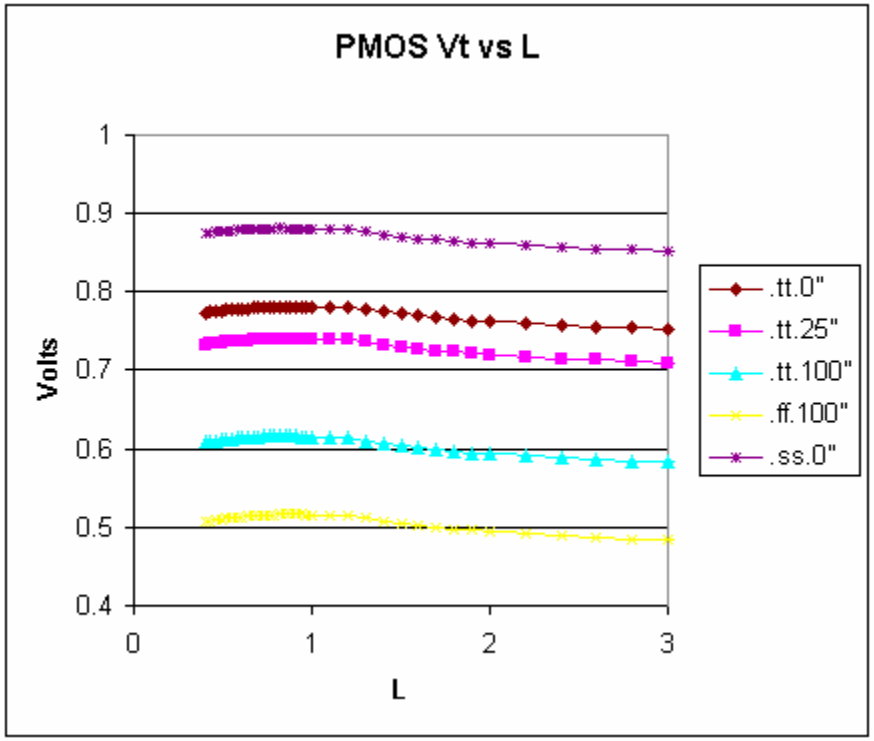
Threshold Voltage nMOS (0.35μ)

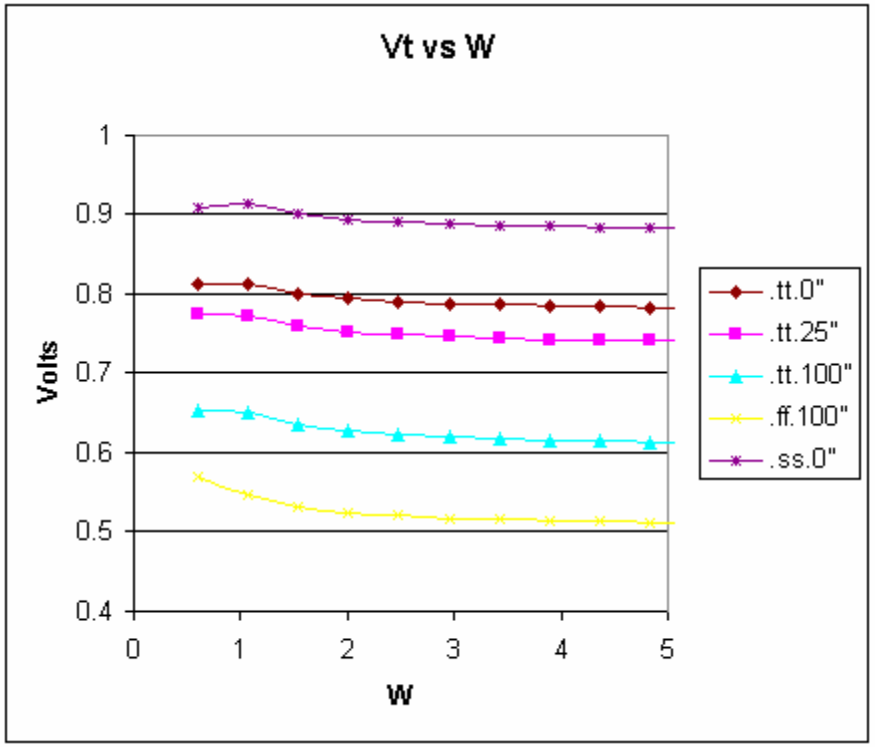




- $V_{th}(w)$ depends on type of isolation and dopant segregation
  - In nMOS, Boron segregates into oxide, lowering V_{th} for small W
  - With LOCOS, V_{th} rises as W falls due to prop. excess Si to deplete
  - With trench isolation, V_{th} falls as W falls due to prop. greater C_{gate}

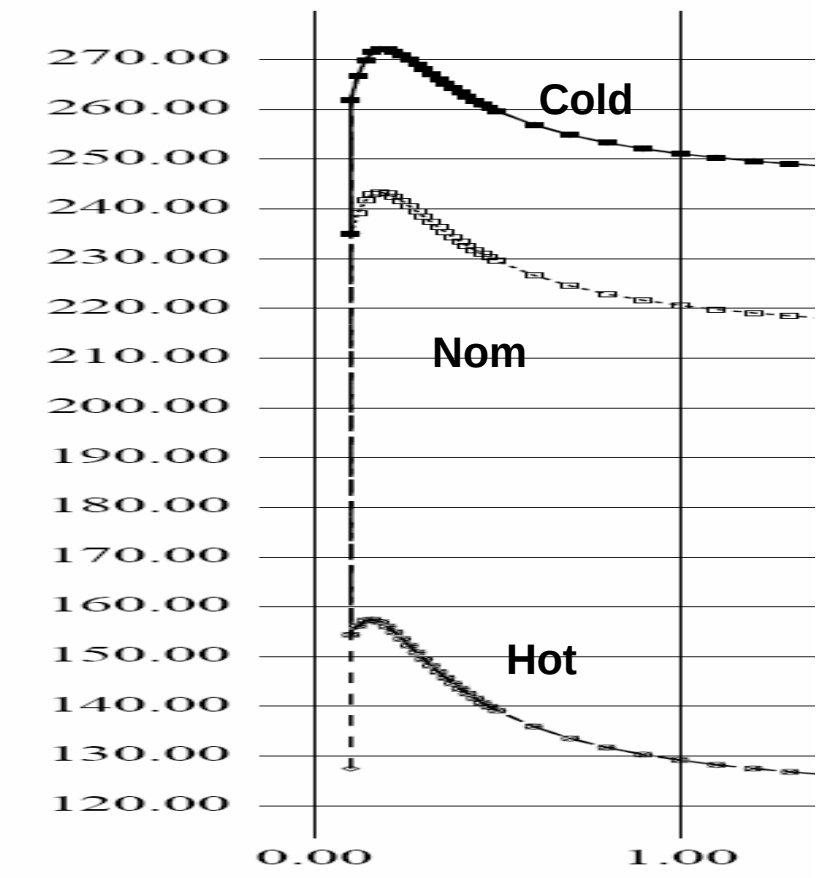
Threshold Voltage pMOS 0.35μ

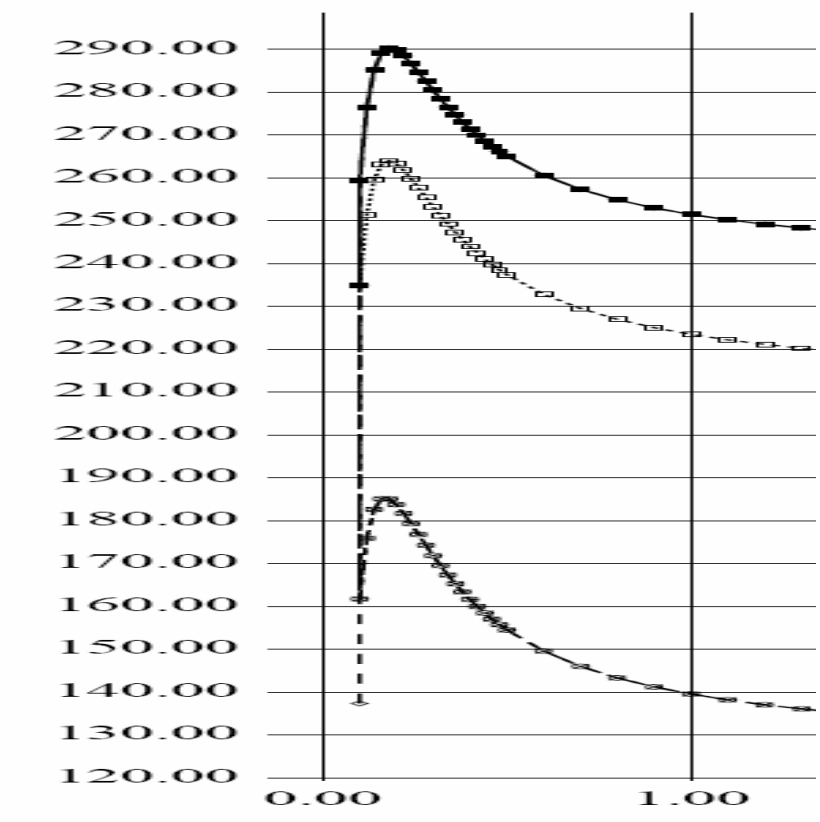




- V_{th}(w) still depends on type of isolation and dopant segregation
  - In pMOS, P/As pile up in Silicon, increasing V_{th} for small W

Threshold Voltage in Newer Processes





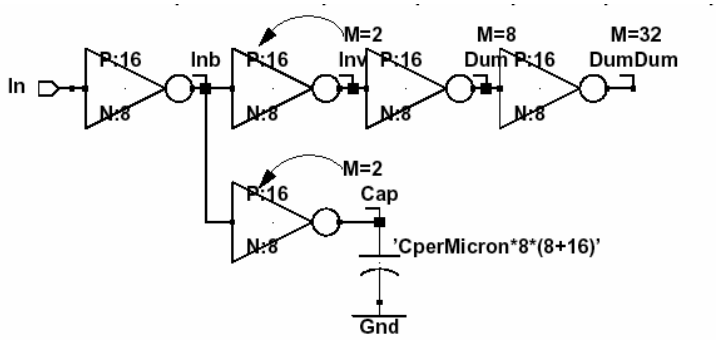
- Reverse short-channel effect

Calibrating a Technology – Next Steps

- Now we have a feeling of how the transistors behave
  - Believe the process/device model (more or less)
  - Or at least understand its limitations
- Move on to thinking about circuit-level issues
  - Timing
  - Parasitics
- We know how to think about digital circuit delays
  - RC trees and logical effort
  - So now calibrate technology for effective R and C values

C_g Calibration for Delay

- Gate capacitance is nonlinear and bias dependant
  - But we can curve-fit a single number (fF/μm) that works for delay
  - Will depend on input slope, output slope, temp, V…



- Find C so delay of 2nd gate (4x) gate is the same in both paths
  - Can change pre/post gate to change input/output slope
  - Fanout of 4 at each stage

C_g Calibration for Power

- If we measured current from V_{dd} at the drive gate we include
  - Current into the load inverter gate (good)
  - Short circuit current due to the drive gate (bad)
  - Current into the drive gate's parasitic diodes and gate overlap (bad)

- Instead, measure the current going into M=8 gate
  - Add a 0V voltage source between driver and gate
  - Average current through the source will be zero (rising and falling)
  - Measure the one-way current (to charge capacitor, for example)
  - C= Q/Vdd and Q = integral of current
  - This should give you the correct answer

- Note that C_g for delay and C_g for power are different

Parasitic Capacitance Calibration

- Effective capacitance of transistor parasitics
  - Can be fF/μm or fF/μm^2 (edge or area)

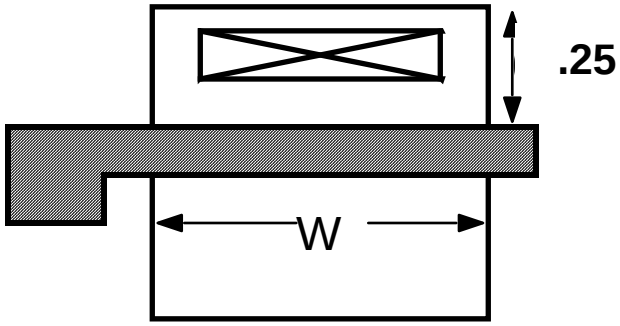
- Complicated because may depend on gate W
  - Gate overlap, diffusion edge under gate
  - Avoid optimization of using very small W to reduce parasitics
    - You end up adding Source or Drain series parasitic resistances

- To extract cap of gate overlap, diffusion edge, and diffusion area
  - Replace M=8 inverter with diode (transistor with grounded gate)
  - Changing gate width, PS, and AS can allow you to estimate caps
  - E.g., setting AS=0, PS=0 gives gate overlap + junction under gate

- Note: diffusion cap for rising and falling transitions are different

Using MOS Capacitances

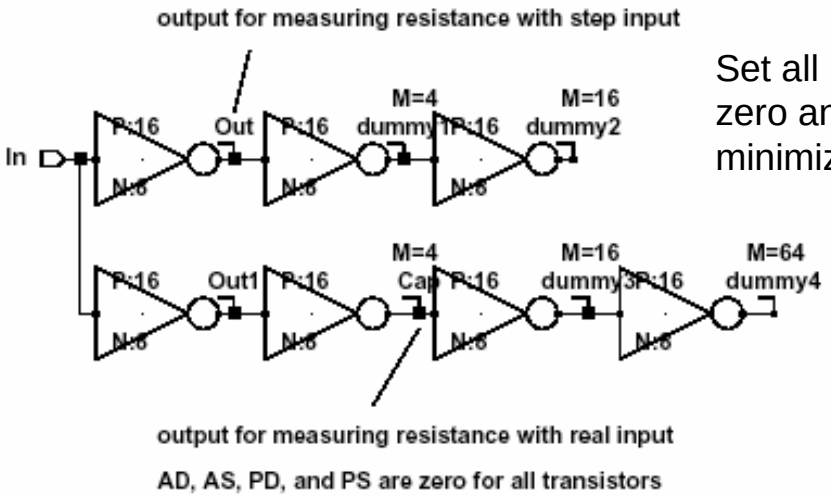
- A 0.1μm technology has a 2.5nm gate oxide
  - C_{ox} = 14 fF/μm^2 = 1.4fF/μm width
  - Gate overlap cap ~ 0.35 fF/μm (per edge)
  - Diffusion cap
    - 1.5 fF/μm^2 bottom plate
    - 0.2 fF/μm sidewall



- Total
  - C_{gate} = 1.4 W
  - $C_{overlap}$ = 0.7 W
  - C_{bot} = 0.4W
  - C_{side} = 0.4W + small constant
    - Counts both gate and non-gate sides of the diffusion

R_{tran} Calibration

- Resistance of a transistor measured in $\Omega\mu m$
  - Know gate effective cap, so R = GateDelay/C_{eff}
  - Will vary with input slope, temp, V



Set all parasitics you can to zero and use large fanouts to minimize other parasitics

- We can also check how R's add (two transistors in series)
  - Replace inverter with enabled tristate inverter. Beware parasitic cap
- Better method: measure delay vs fanout; R_{eff} comes from slope
  - Just change the fanout of all the gates in the chain

Now What?

- Use your simple RC models to reason about circuit
  - Look at different trade-offs
  - Try to determine what is important
  - If you need more information, do some sims to build new model
  - Come up with ’good’ first pass design

- Simulate it
  - First look at a few of the corners that might be interesting
  - Do the results make sense?
    - If they don’t match your model, something is wrong!
    - If not, check the schematics, SPICE files, and your models
  - Check it over many corners

Simulation Issues

- Complexity gives rise to a conflict in simulating ICs

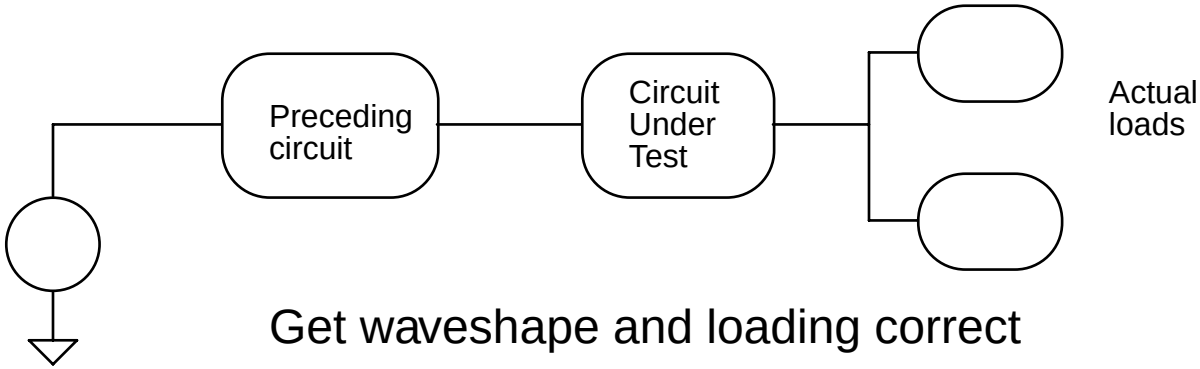
1. "Simulation is cheap, silicon is VERY expensive"
   - Don't scrimp when you construct a SPICE deck
   - Simulate the real circuit under real conditions (temp, power, clock)
   - Include the real input waveform and real output load devices

2. "SPICE decks that are too complex have too confusing results"
   - Very easy to make mistakes in entry
   - You may be simulating the wrong thing
   - Big decks have lots of interacting small mistakes → hard to debug
   - Simulations run very slowly

Start Simple and Add Complexity

- Incremental simulation is a design compromise
  - Start with an understandable and predictable simulation deck
  - Add more complexity
  - Check at each step that the results make sense
  - End up with complete simulation file

- Make sure to eventually add all the effects you need to model



Get waveshape and loading correct