4541.831:
Integrated Circuit Design Projects

Jaeha Kim

Mixed-Signal IC and System Group (MICS)

Seoul National University

jaeha@snu.ac.kr

Course Basics

- Instructor: Jaeha Kim
  - Office: Bldg. 104-1, Room 314 or Bldg. 301, Room 507
  - Office Hours: Mon & Wed 1pm~2pm
  - Email: jaeha@snu.ac.kr, Phone: 880-1671
- Teaching assistants
  - Sangho Youn (PhD), Hanseok Kim (MS)
  - Email: shyoun@mics.snu.ac.kr, hskim@mics.snu.ac.kr
- Course website:
  - TBD on etl.snu.ac.kr
  - TA email alias: ta@mics.snu.ac.kr

About Your Instructor

- BS from SNU
- MS and PhD from Stanford (Advisor: Mark Horowitz)
- Worked at True Circuits, SNU-ISRC, Rambus, …
  - Most recently with Stanford as Acting Assistant Prof.
  - With SNU since 2010
- Main research interests include:
  - Low-power mixed-signal systems
    - such as high-speed I/Os, PLLs, CDRs, ADCs, DACs, …
  - And their design methodologies
    - key to achieve optimal balance between competing metrics

Course Goals

- To introduce *systematic* ways of designing analog circuits by leveraging the circuit intent/abstraction
  - Our belief is that we can make analog design as easy and effective as digital design!
- Systematic analog design consists of:
  - Expressing what we want and what we don't want – the circuit intent and specifications
  - Measuring these quality metrics in automatic ways to enable quick design iterations
  - Leveraging design equations to build understanding/insights on the circuits and guide the optimizers to find the best solution
  - Use of models to scale the complexity of the circuit/system

Course Scopes

- This course is not about circuit topologies or architectures
  - e.g. I won’t cover all the different op-amp topologies such as two-stage, folded-cascode, telescopic, …
  - Those topics should be covered by another course
- Instead, this course will focus on how to arrive at the best implementation of a given circuit idea
  - Without running extensive SPICE simulations while sweeping parameters exhaustively
  - These methodologies should serve as the foundations for making your circuit research fun!

Prerequisites

- Some experiences in circuit design and simulation
  - Carried out a circuit project with SPICE
  - Tried to size transistors either with design equations or with SPICE simulations
  - Have been frustrated with endless simulations that seemed to head nowhere
- Some understanding on the linear system theory and probability theory
  - Linear systems: impulse responses and transfer functions
  - Probability density functions (PDF), auto-correlation functions, power-spectral densities (PSD), …

About Teaching Materials

- This course is a fresh new attempt to make analog circuit design systematic and therefore the lecture notes will serve as the principle reference material
- If you want to get some general ideas on where we are heading, I recommend reading:
  - □ J. Kim, M. Jeeradit, B. Lim, M. A. Horowitz, "Leveraging Designer's Intent: A Path Toward Simpler Analog CAD Tools", Custom Integrated Circuits Conference (CICC), 2009.
  - □ Or, my other recent publications on analog design methodology

Course Syllabus

- Introduction to analog abstractions (week 2)
  - □ The circuit intent for analog – linear system
- Understanding IC process technology (week 3)
  - □ Use of automated simulation scripts
  - □ Process characterization
- Test-driven methodology (week 4-5)
  - □ “Agile” for analog circuit design
  - □ Understanding circuit specs and how to measure them
- Circuit simulation strategies (week 6)
  - □ SPICE review
  - □ Introduction to periodically time-varying analyses (SpectreRF)

Course Syllabus

- Circuit simulation strategies (cont'd: week 7~9)
  - Introduction to Verilog-A analog behavioral language
  - Variable domain transformation
  - PTV/ISF simulation for clocked comparators
  - Variability/mismatch simulations
- Writing design equations (week 9)
  - Intent-based design equations
  - Checking the soundness
  - Introduction to Gm/Id methodology

Course Syllabus

- Introduction to circuit optimization (10 week)
  - Optimization basics
  - Common pitfalls with circuit optimizers
  - Ways to use circuit optimizers effectively
- Working with analog models (11 week)
  - Using analog models to speed up the simulation and check your design equations (intent)
  - Top-down vs. bottom-up strategies for writing models
  - Strategies for verifying analog models
- System-level considerations (12 week)
  - Propagating system-level specs to circuit-level specs
  - Mixed-signal system designs

Course Grading

- Your course grade will be determined based on:
  - Homework: 40%
  - Final Project: 40%
  - Participation: 20%
- Each homework is actually a small piece of the project
  - HW#1: get to know the process with simulation scripts
  - HW#2: choose a circuit and define its purpose and ways to evaluate its quality
  - HW#3: compose simulation testbenches that measure the quality metrics
  - HW#4: write design equations for the quality metrics
  - HW#5: try out circuit optimizers for finding the best sizes

Project

- Choose your own circuit and find its best implementation
  - That is, find the optimal transistor sizes or configurations that give the best performances under the specs
  - You could select a circuit from the literature and try to size the transistors to meet its specs
  - You can work solo or as a group of two
- *Important:* your work will be evaluated based on how you got the answer rather than how good your answer is
  - I can't check your answer directly so you need to convince me that what you have done finds the optimum
  - Clever new ways/methodologies to achieve this is a big plus
  - This is exactly what you do at companies – "design reviews"

Project

- To see if you have indeed arrived at the best answers using good methodologies rather than resorting to brute-force simulations or by accident, I may introduce some changes at the last minute, for example:
  - Updating the transistor models
  - Migrating the design to a different process technology
  - Changing the target specs
- All of these last-minute changes are in fact very REAL
  - Only way to make these changes "painless" is to automate the process as much as possible

Participation

- In this course, participation takes *20%* of the grading
  - The idea is that each homework/project is actually an experimentation of a new design flow and it deserves an active discussion
  - For homeworks with open problems, this substitutes distributing the solutions
  - Students may be asked to present their homework and others to comment on their findings
  - In every class, students who actively participated in discussions will collect points
  - Note that simply attending the class but not participating in discussions may not earn you points

CAD Tools Used in the Class

- In this course, you will be introduced to a number of CAD tools for analog circuit design
- Some are commercial:
  - □ Cadence Virtuoso Schematic Editor
  - □ Synopsys HSPICE or Cadence Spectre
- While others are in-house built:
  - □ Mulan/Simba simulation scripts
  - □ Eva equation tools
  - □ Polka optimizer
- For technology models, we will provide the PTM models but you may use the process of your choice as well

Honor Code

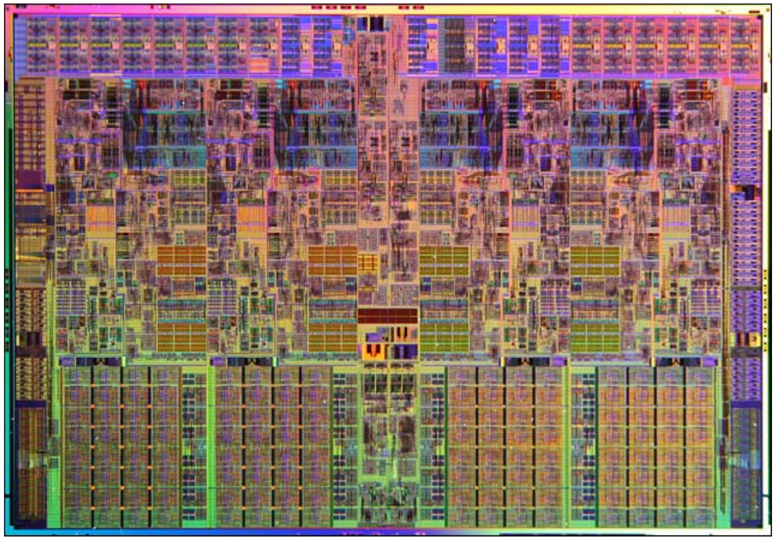
- I am less worried about the issue for this class, but please remember you are bound by the honor code:
  - □ Do not plagiarize others' work (e.g. homework, project, exam,…)
  - □ If you need help, feel free to ask the instructor and TAs
- But if you are found cheating it is *very serious*
  - □ Besides getting zero score for the work
  - □ There will be a formal hearing
  - □ You can be thrown out of SNU
- So save yourself and us a huge hassle and be honest
- Always remember that your goal in this class is to <u>learn</u> how to design and build circuits
  - □ No way to cheat that process

Lecture 1. Analog Bottleneck

Jaeha Kim
Mixed-Signal IC and System Group (MICS) / ISDL
Seoul National University
jaeha@ieee.org

The Analog Bottleneck

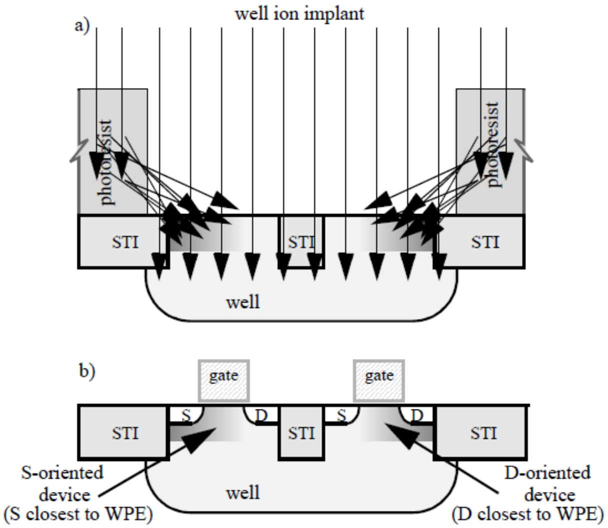
- The main driving force behind the complexity scaling of CMOS systems has been digital, not analog
  - Near a billion of transistors in modern processors
- Analog circuit hasn't scaled its complexity much
  - 10~1000s of transistors
  - Technology scaling makes its design difficult and forces even simpler analog
  - Trend is to replace pesky analog parts with digital for easy process migration and design management (Big D, Little A)



Intel 4-core Nehalem processor (820M)

Challenge 1: Device Behavior Complexity

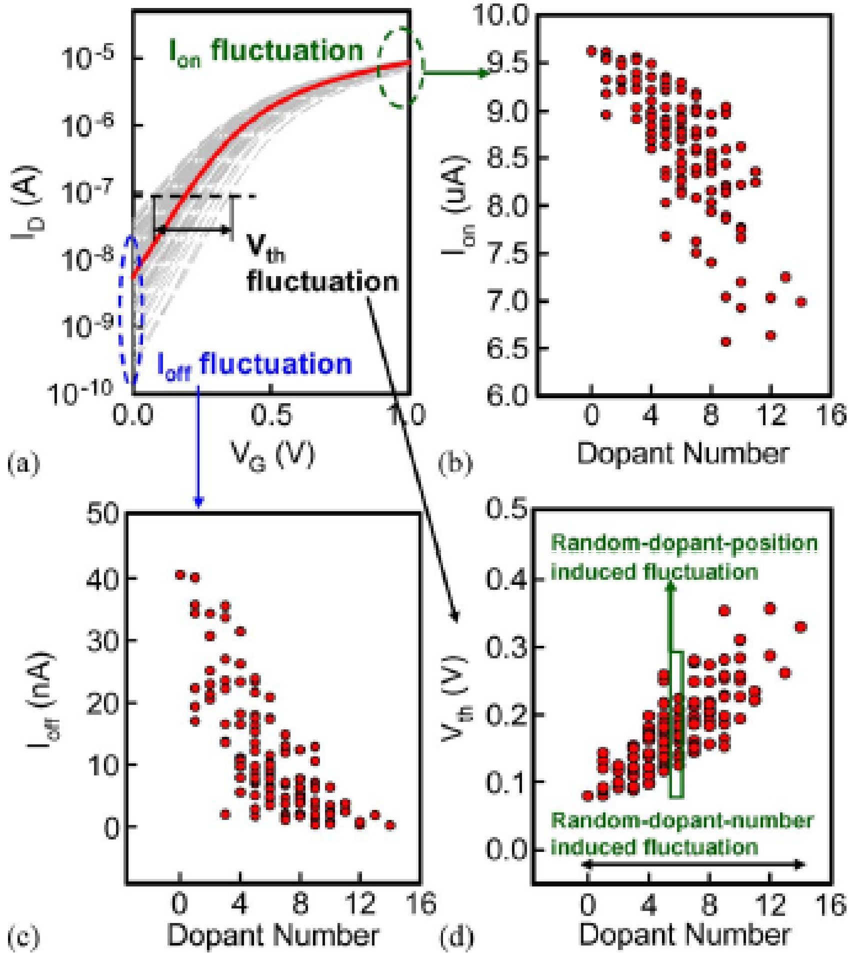
- With technology scaling, the list of “strange effects” keeps piling up – making circuit designs more difficult
- Deviations from classical, long-channel model:
  - Carrier velocity saturation
  - Drain-induced barrier lowering (DIBL)
  - Subthreshold leakage
  - Gate direct tunneling leakage
- Proximity effects:
  - Well proximity effect (WPE)
  - Shallow trench isolation stress effect
- Reliability degradation:
  - Negative bias temperature instability (NBTI)



P. Drennan, et al, “Implications of Proximity Effects for Analog Design”, CICC ‘06

Challenge 2: Uncertainty and Variability

- Designers have to cope with uncertainties in device parameters
  - □ PVT variations (die-to-die)
  - □ Systematic and random mismatches (within-die)
- Calling for circuit designs that
  - □ Are insensitive to parameter variations
  - □ Can be calibrated to the best condition

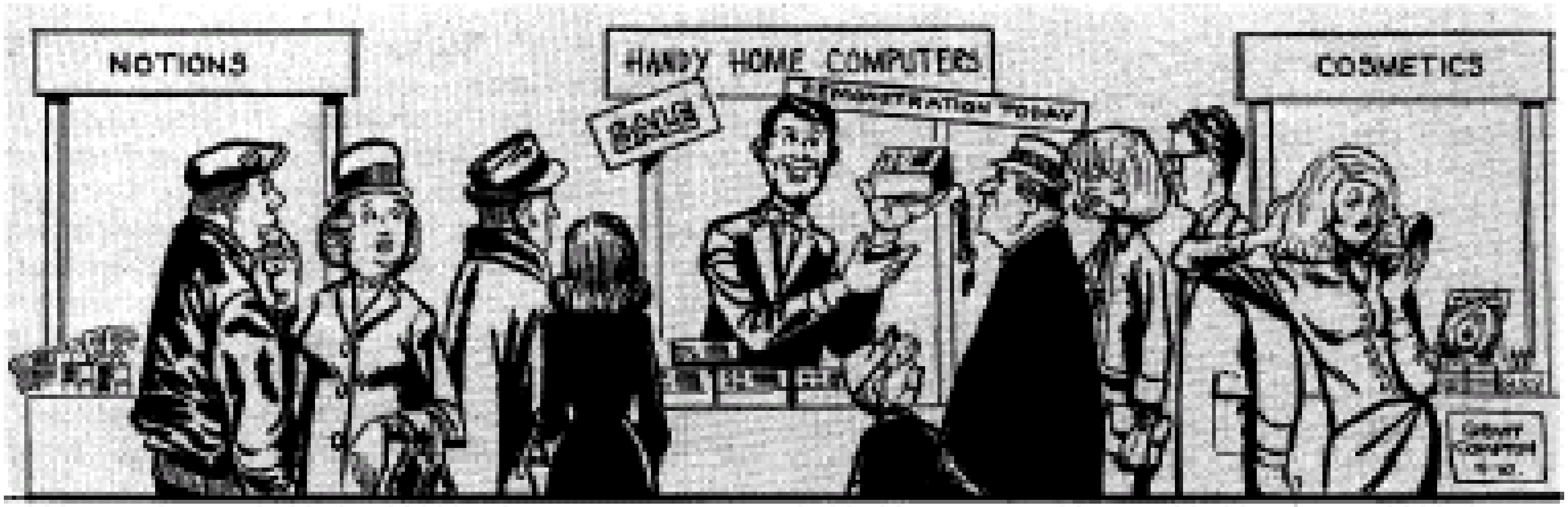


Y. Li, "Random-Dopant-Induced Variability in Nano-CMOS Devices and Digital Circuits," Trans. Electron Devices, 08/2009.

Signs of the End of CMOS Scaling

- In his original paper in 1965, G. Moore predicted issues that can limit the scaling
  - Power dissipation
  - Design cost
  - Running out of things to do with all those transistors

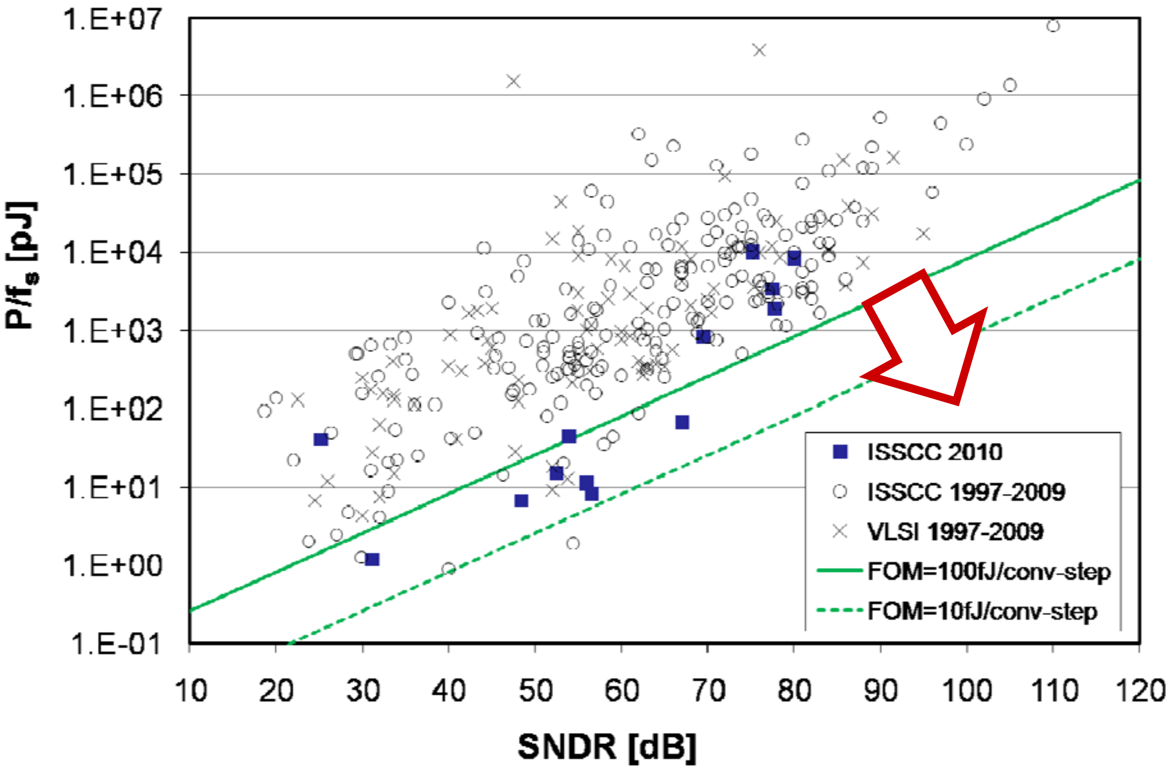
Electronics, Volume 38, Number 8, April 19, 1965



ftp://download.intel.com/research/silicon/moorespaper.pdf

Challenge 3: Push for Lower Power

- Power limits what you can integrate on the chip

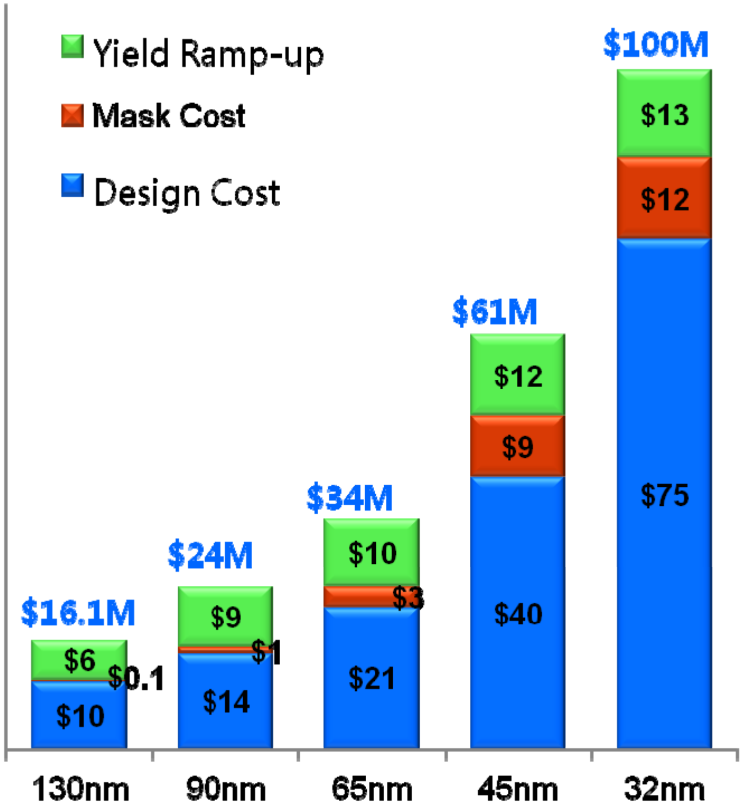


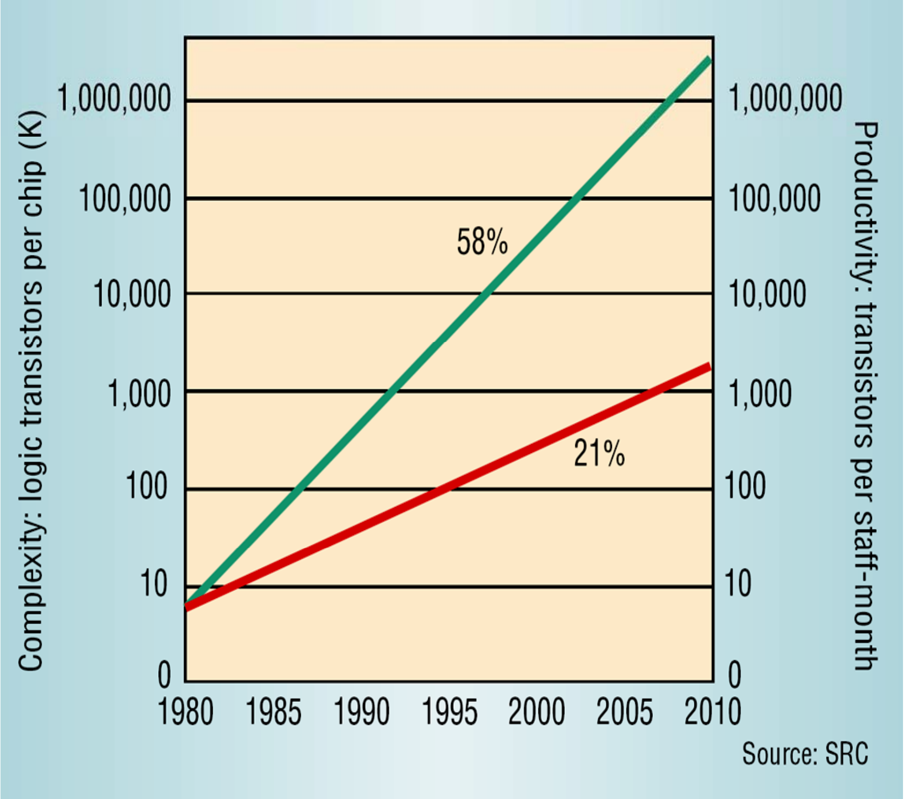
- Designers need to consider multiple objectives: esp. performance vs. power
- And understand the trade-offs and identify ways to reduce power

B. Murmann, "ADC Performance Survey 1997-2010," [Online]. Available: http://www.stanford.edu/~murmann/adcsurvey.html

Challenge 4: High Design Costs

- Rising design costs due to ever-increasing design and verification costs can justify very few markets today





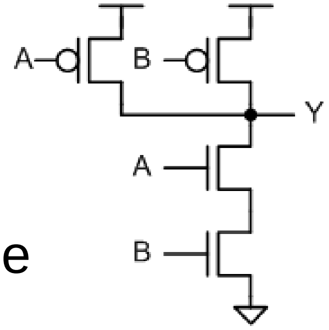
[Source: Y. Huh, SiliconMitus, Inc.]

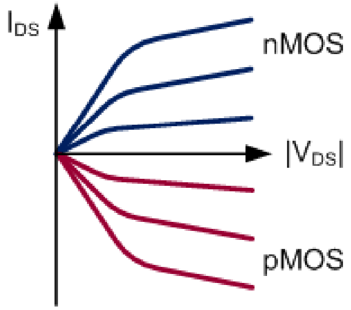
Analog Circuit Design – A Dark Art?

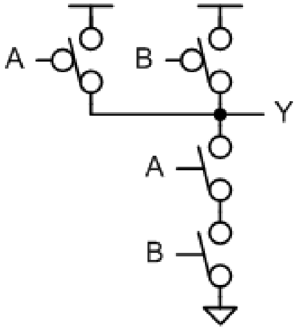
- Wait – all of these challenges apply to digital as well
  - Digital systems are made of the same transistors
  - Facing similar power and cost constraints
  - In fact, more difficult – must deal with higher complexity!
- The real problem for analog is lack of design process
  - It's not clear how good analog circuits are built
  - Paper and pencil approach?
  - Exhaustive SPICE simulation?
- I believe the root cause is:
  - Designers don't think much about *what they really want*
  - EDA industries try to provide tools that can do everything (e.g. fast & accurate SPICE that can simulate any circuits)

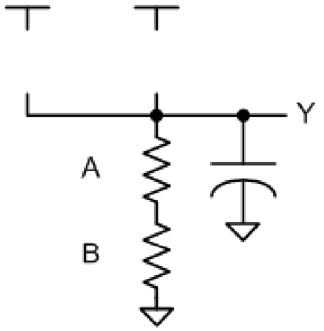
Knowing What You Want Goes a Long Way

- How do you design a good NAND gate in CMOS?
  - How do you know if this circuit will operate correctly in presence of complex device behaviors?
  - Run many SPICE sims?
- Knowing the intent lets you divide and conquer the problem
  1. Check if the circuit has the correct functionality in the ideal world
  2. Use first order models to estimate its delay and power
  3. Use SPICE to check its noise margins



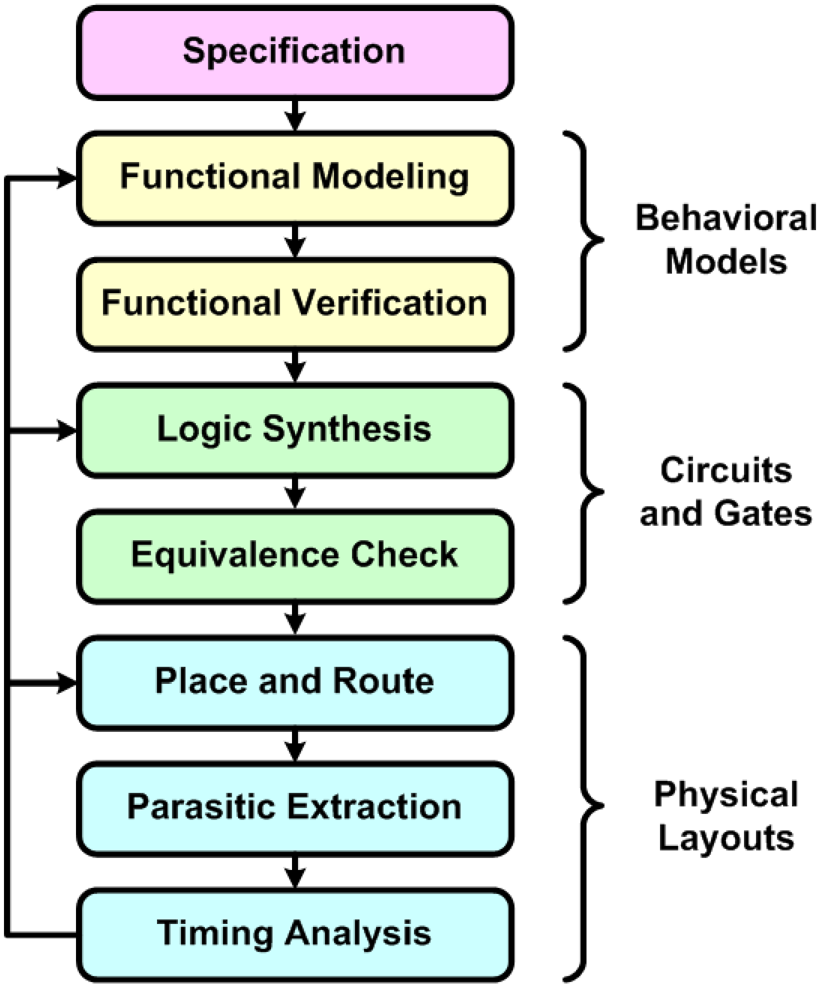






Closer Look at Digital Design Flow

- Digital has an established design flow with well-defined milestones
  - Do models have correct functionalities?
  - Do circuits have same functionalities with models?
  - Are delay and power within acceptable bounds?
- Each question can be answered with help of automation tools
  - Not necessarily with almighty SPICE



How to Make Analog Design Like Digital?

- This is a long-sought question: how do we make analog circuit designs as easy as digital designs?
- Some say: let’s make analog circuits with digital gates
  - e.g. all-digital PLL or CDR
  - Does it solve the problem? -- Can I design/validate the PLL based on the digital flow?
- Others say: let’s pretend analog circuits behave like digital
  - Discretize the analog signals with many quantization steps
  - After discretization, the analog circuits can be analyzed with digital tools (called “hybrid verification”)
  - What is the problem with this approach?

How to Make Analog Design Like Digital?

- Note that digital tools leverage “abstractions” effectively
  - □ Digital abstraction: Boolean (value), synchronous (time)
  - □ Tools that simulate abstraction models, convert between abstraction layers, check equivalence, measure coverage, …

So, analog tools must leverage abstractions – but what is the proper abstraction for analog?

  - □ No notion on analog abstraction; SPICE treats any circuit as a general nonlinear system
  - □ Designers scream for *faster* SPICE; although it’s not the solution