Principles of Paranoid Design

Tim Edwards

SVP Analog

efabless

efabless.com

Open Circuit Design

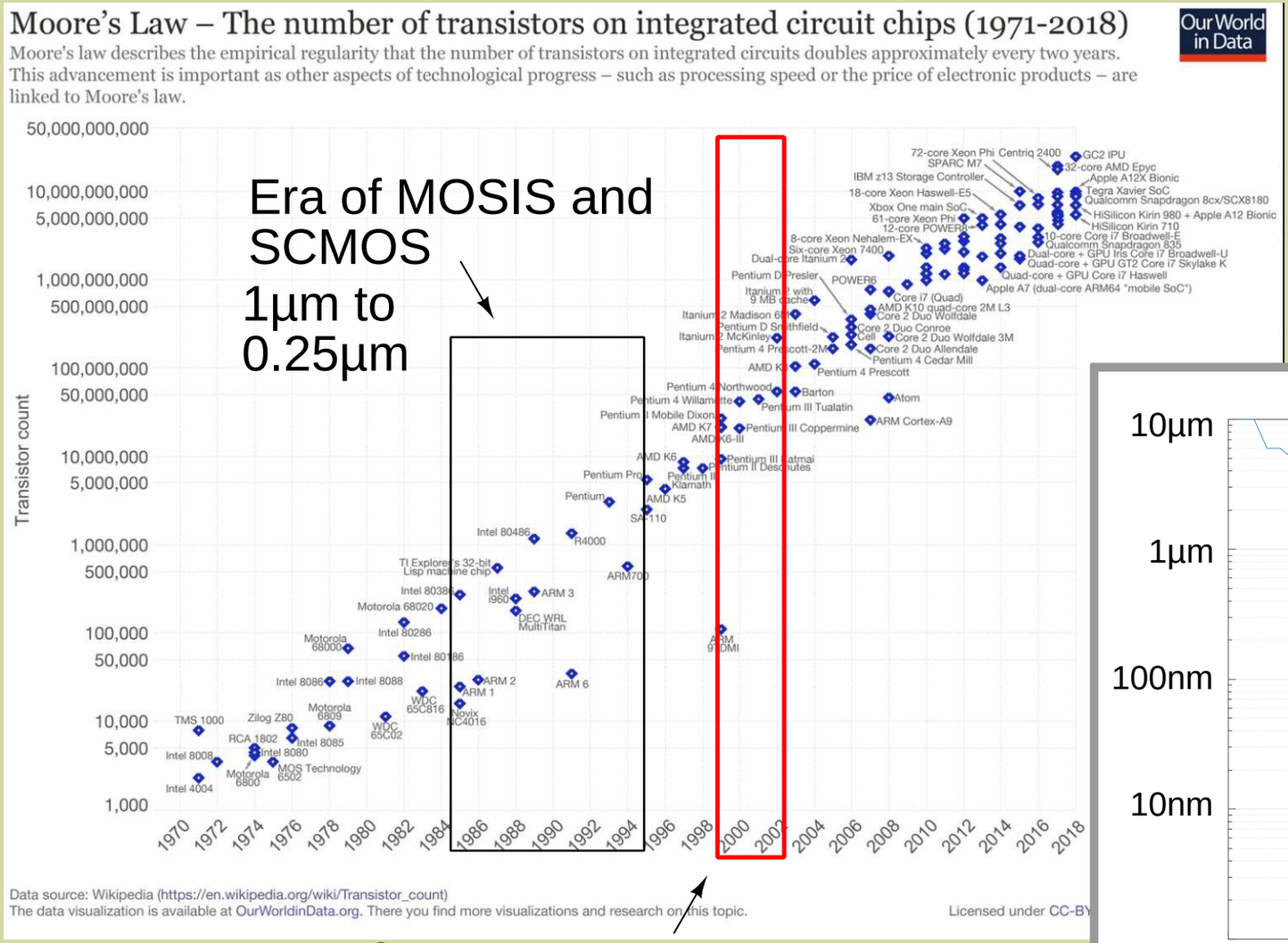
opencircuitdesign.com

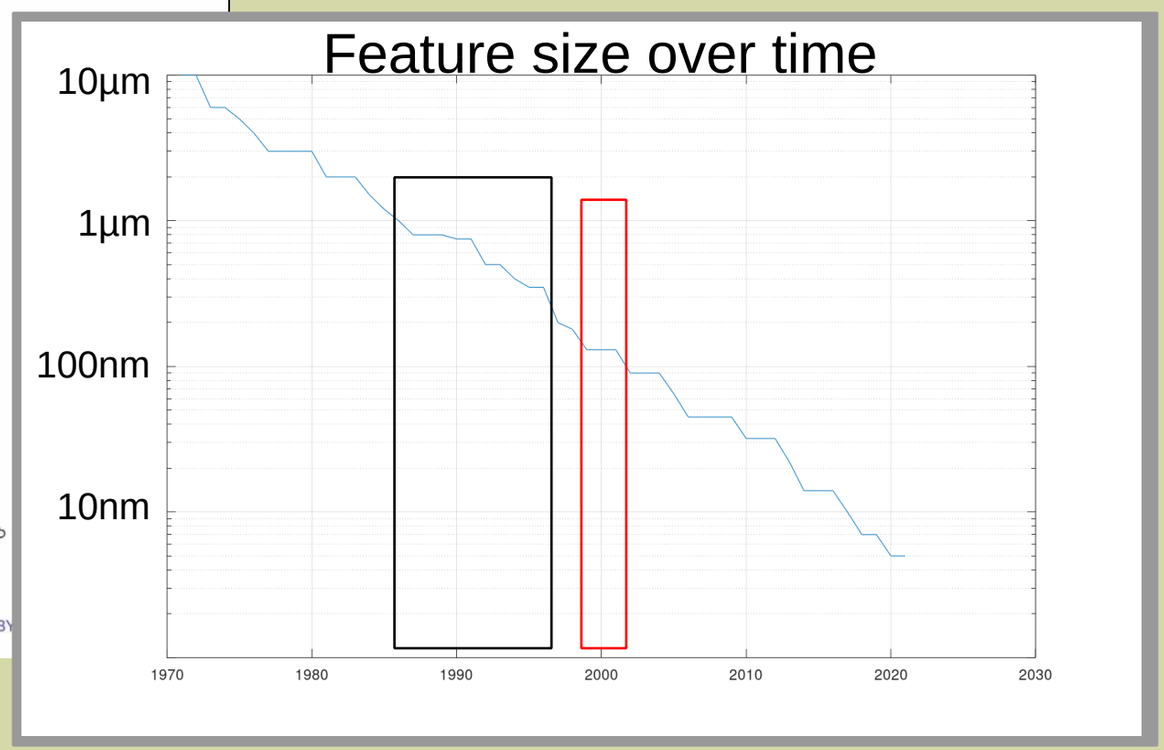
“The paranoid user is a successful user”

OSDA

April 17, 2023

Moore's Law and the Drive for Performance



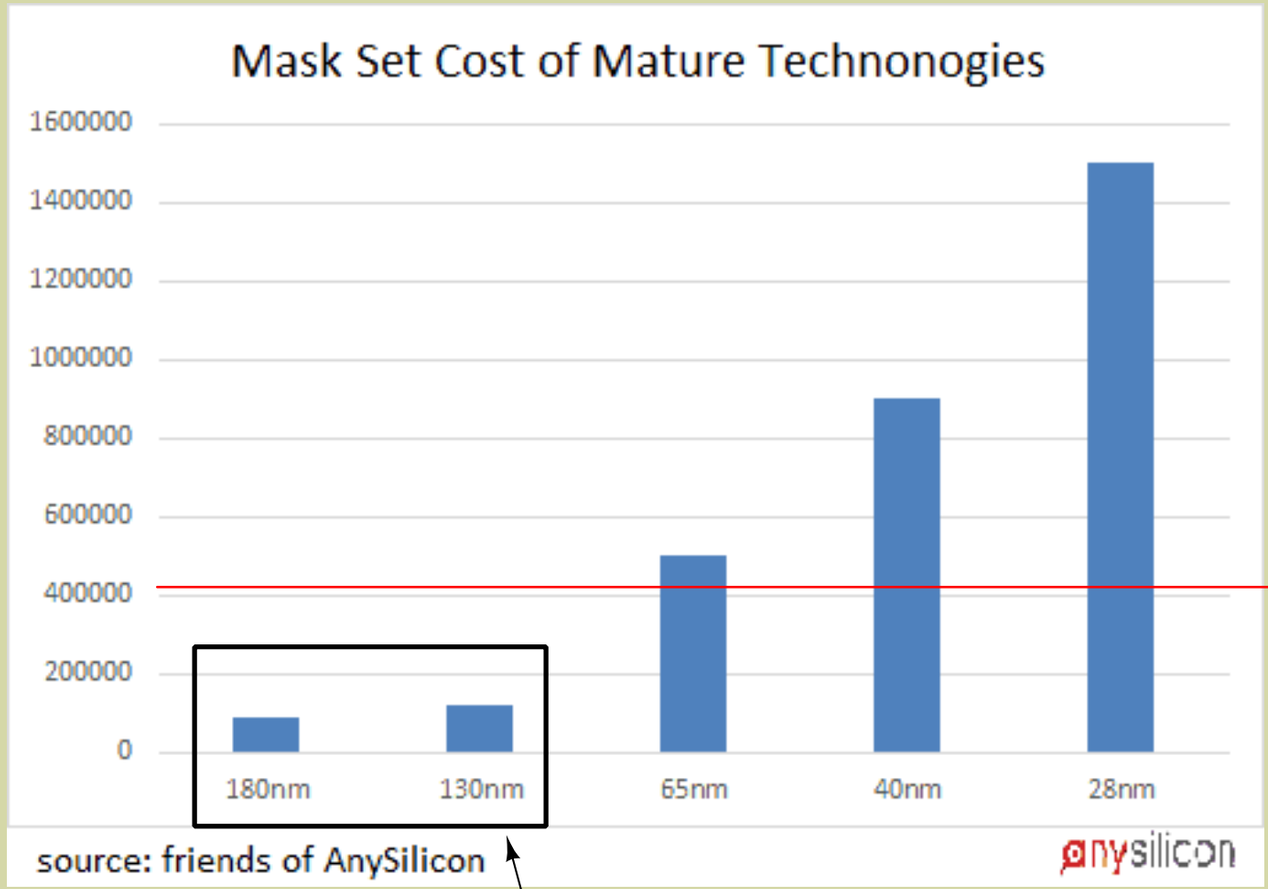


Current open-source processes
180nm, 130nm
First appeared ca. 2000–2002

OSDA

April 17, 2023

Cost of Fabrication (Mask Sets)



Median cost of a house in the U.S. ($425,000)

Current open-source processes
180nm, 130nm
First appeared ca. 2000–2002

Cost of Learning (Price per mm^2)

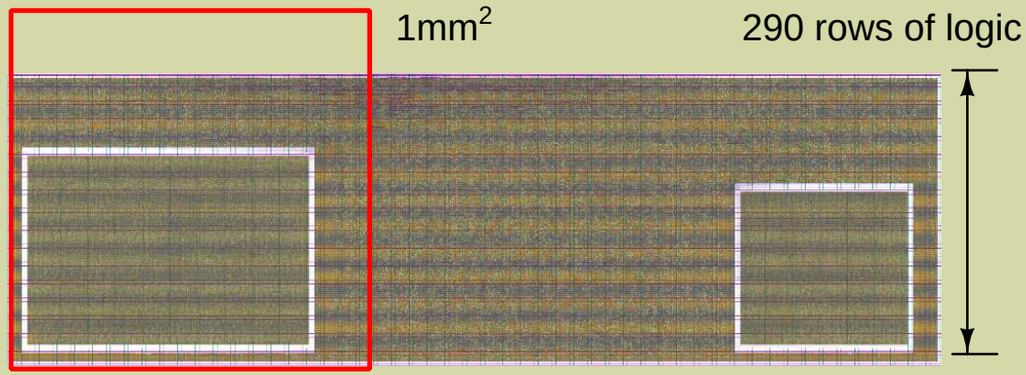
Source: Europractice (combination of GF and X−Fab prices for MPW runs)

(Prices in Euros are approximately the same as USD in 2023)

Feature size	Cost/mm^2 (US$)
12 nm	30,000
22 nm	20,000
45 nm	10,000
55 nm	6,000
130 nm	2,300
180 nm	2,000
350 nm	1,200

This is also the cost of failure if your project doesn't work.

For comparison, the RISC−V processor used in the Efabless Caravel chip is about 2mm^2 at 180 nm.



Reliance on Precision Data

Standard design practices assume a PDK with essentially perfect data.

Foundry data for established nodes in commercial PDKs is generally reliable.

Commercial PDKs developed over many fabrication iterations

Traditional Design Methodologies

Margining

Process corner data required. . . varies from PDK to PDK

Ideally:

FET corners: fast−fast, slow−slow, slow−fast, fast−slow, typical (N, P devices)
BJT corners: fast, typ, slow
Capacitance corners: high, typical, low
Resistance corners: high, typical, low
+ All corners with mismatch variation

Corner data needed for SPICE device models, technology LEF, parasitic extraction, liberty format files.

Monte Carlo Simulation

Better than corners because all parameter variation can be handled at once regardless of device type.

Requires hundreds of simulations to produce statistically meaningful results.

Trust of PDKs and Tools

Reliability of foundry data for open source PDKs is unknown.

Open source PDKs play catch-up

- Some data formats cannot be open sourced
- Some data (e.g., parasitics) are collected from multiple sources or regenerated from scratch.
- Some sources are completely unreliable as they aren’t used by the commercial tool flow and so aren’t checked or kept up to date.
- Open source PDKs are new and there are few measurements from real manufactured silicon.

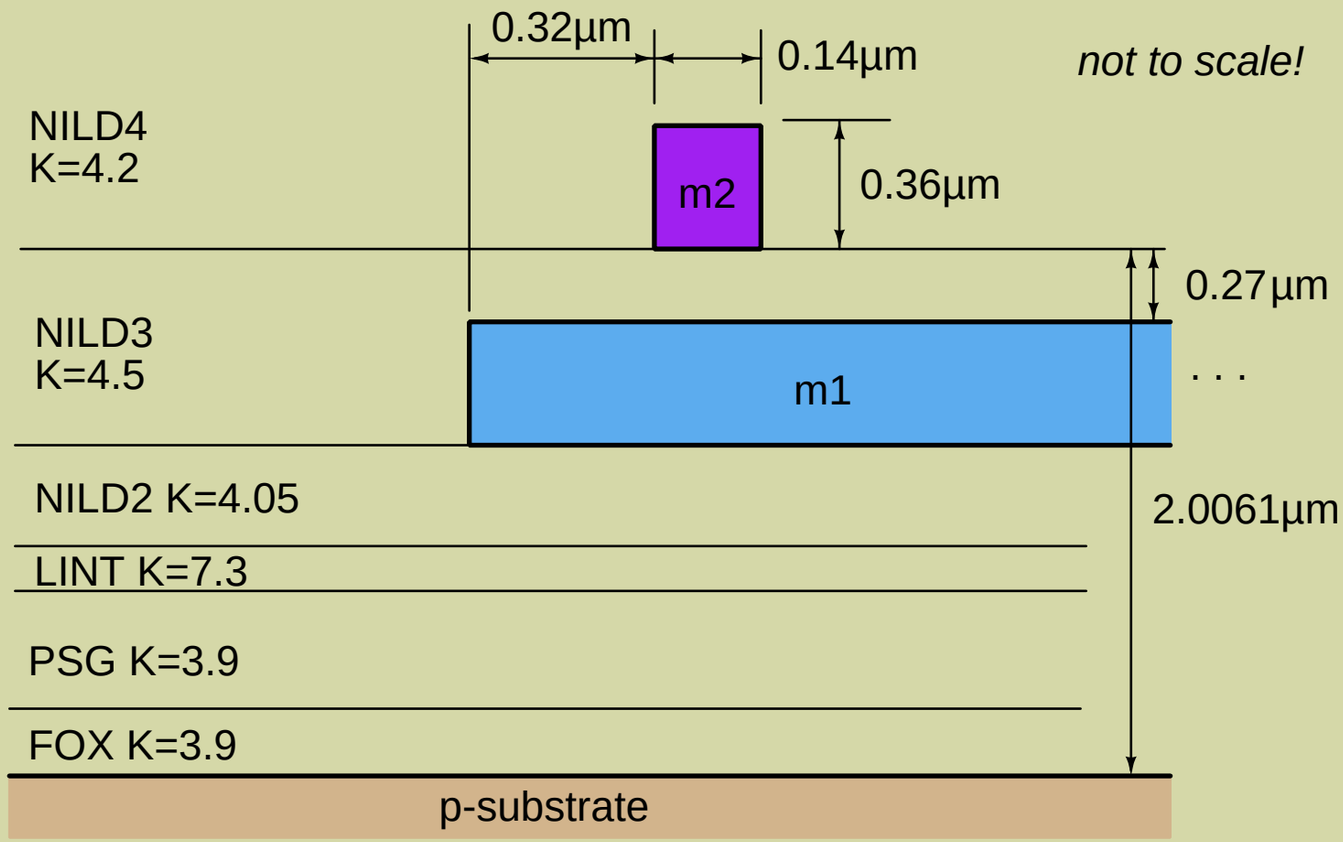
Until we have many more measurements of open−source silicon, need to rely on assorted tools and methods in the public domain.

PDK Cross-Validation

FasterCap

Open source field equation solver (2D and 3D)

https://www.fastfieldsolvers.com

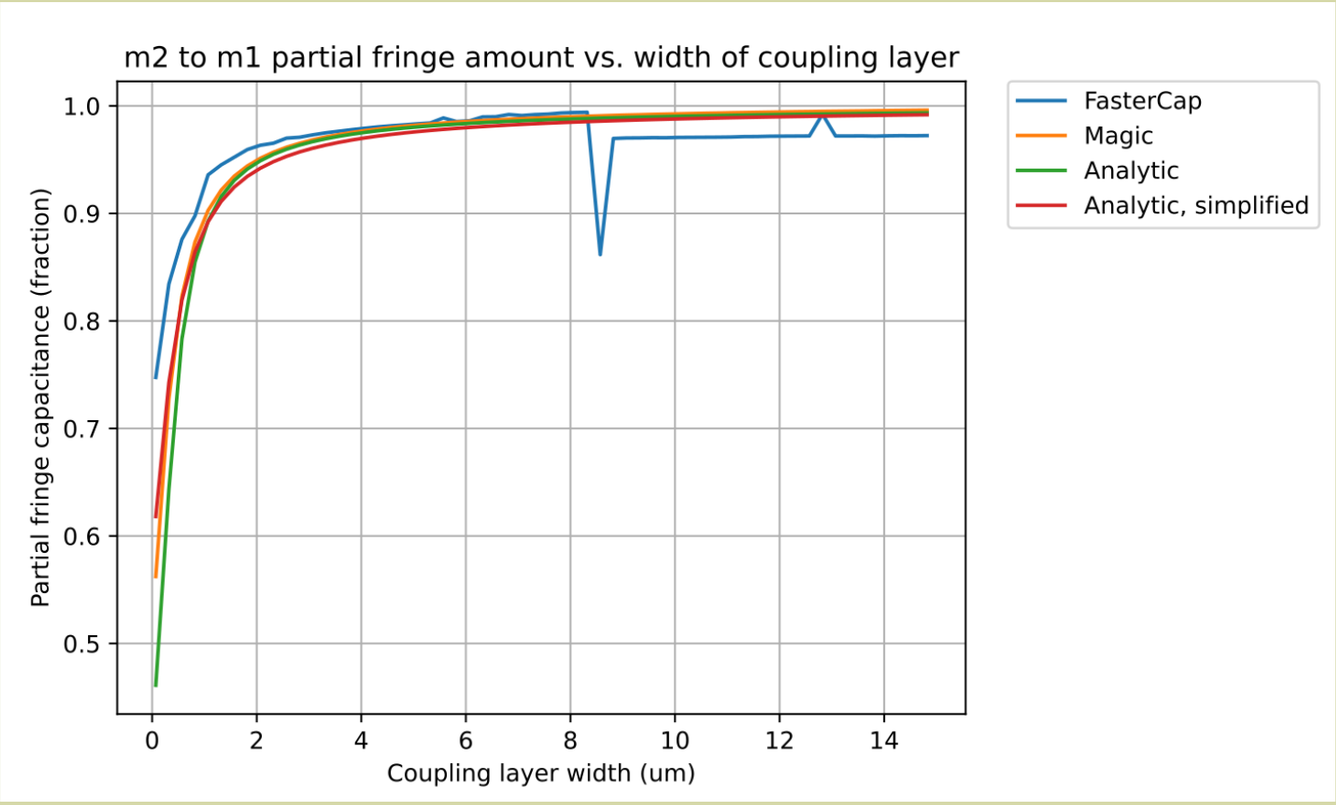


April 17, 2023

PDK Cross-Validation

FasterCap

Capiche Project: https://github.com/RTimothyEdwards/capiche



FasterCap analysis of metal2 to metal1 overlap fringe in sky130A

April 17, 2023

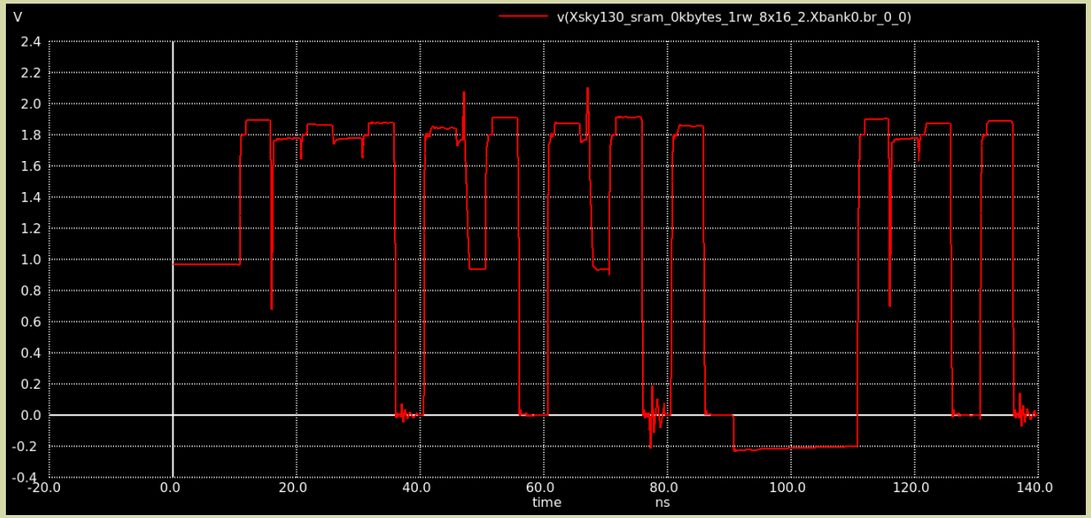
PDK Cross-Validation

Magic

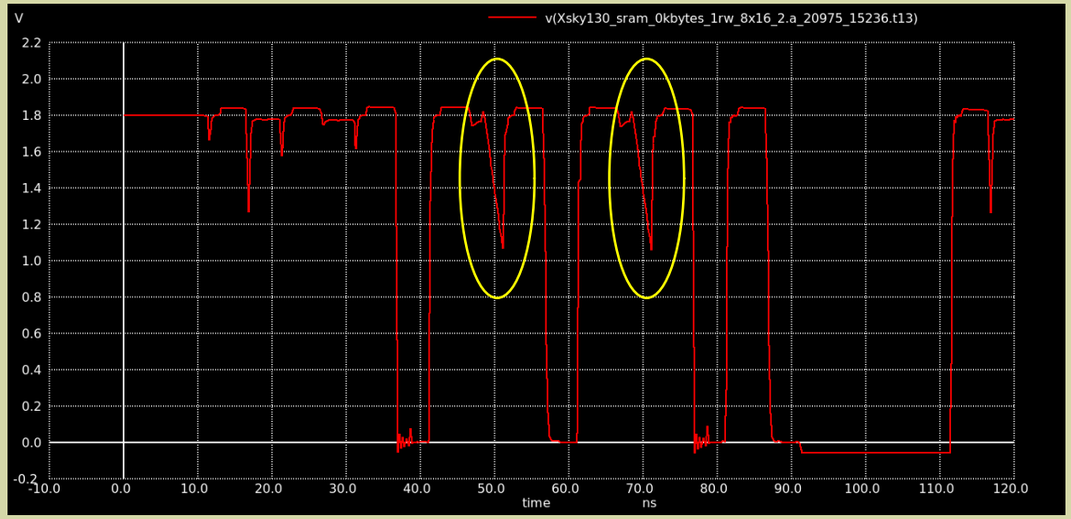
https://github.com/RTimothyEdwards/magic

http://opencircuitdesign.com/magic

Full R-C parasitic extraction



From schematic netlist



From full R−C extracted netlist.

Simulation of OpenRAM macro showing 1st column bit. The full R−C simulation shows failure of the net to fully precharge at 10MHz clock frequency.

PDK Cross-Validation

ngspice

https://ngspice.sourceforge.io

Recent updates for verilog-A integration and plug-in models (OSDI/OpenVAF)

Xyce

https://xyce.sandia.gov

Recent updates for general compatibility with other SPICE variants

Design in Open Source PDKs

Established nodes give an opportunity to explore novel ideas, architectures, and methods.

Performance should not be a critical goal on a process with an open source PDK that has not been independently matched to measurements from silicon!

Need to concentrate on the novel part and not worry about a project being rendered useless by a piece of non−functional infrastructure.

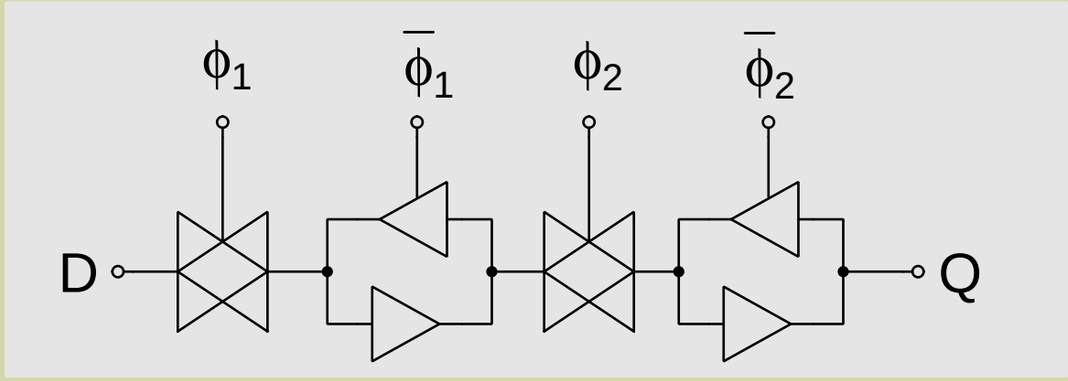
Designers need to be using robust methods.

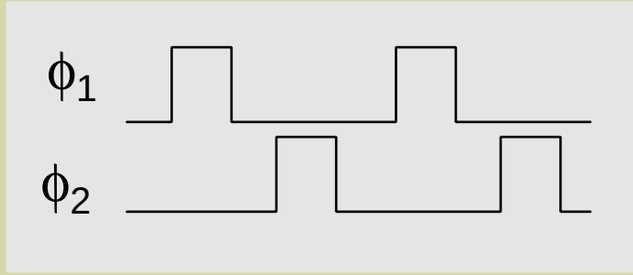
Often, these methods come from long ago, and are not taught, advertised, or promoted due to the incessant push for performance.

Robust (Paranoid) Design Methodologies

2-phase clocking

Avoid all hold violations by clocking on two independent phases



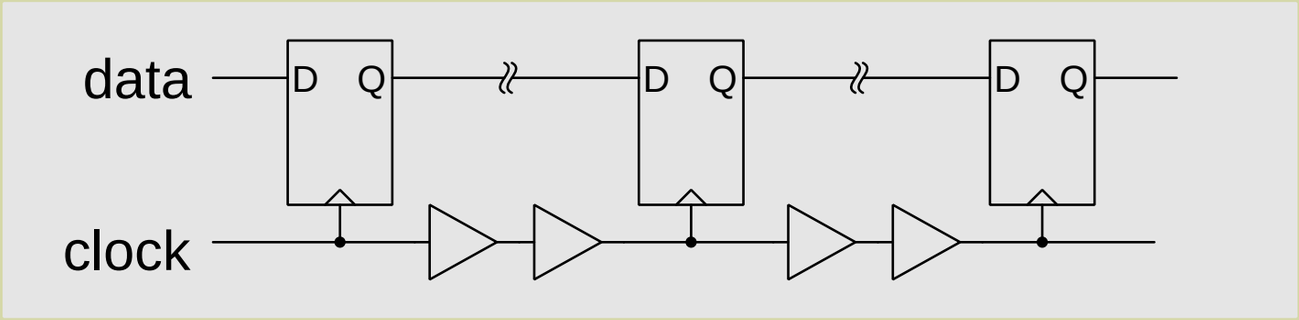


Advantage: All setup violations corrected by slowing the clock frequency, all hold violations corrected by increasing the time between clock phases.

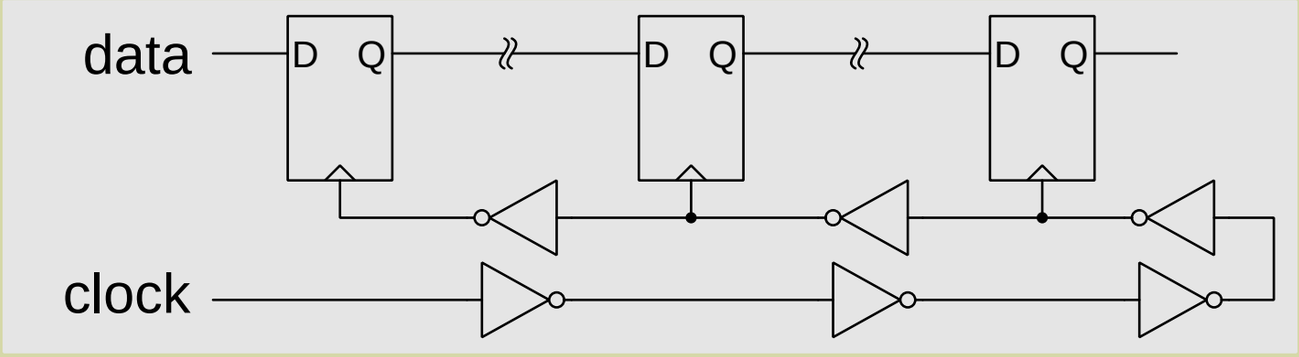
Disadvantage: Two clock trees. Routing tools likely not to know how to route two parallel clock nets efficiently, leading to lower performance.

Robust Design Methodologies

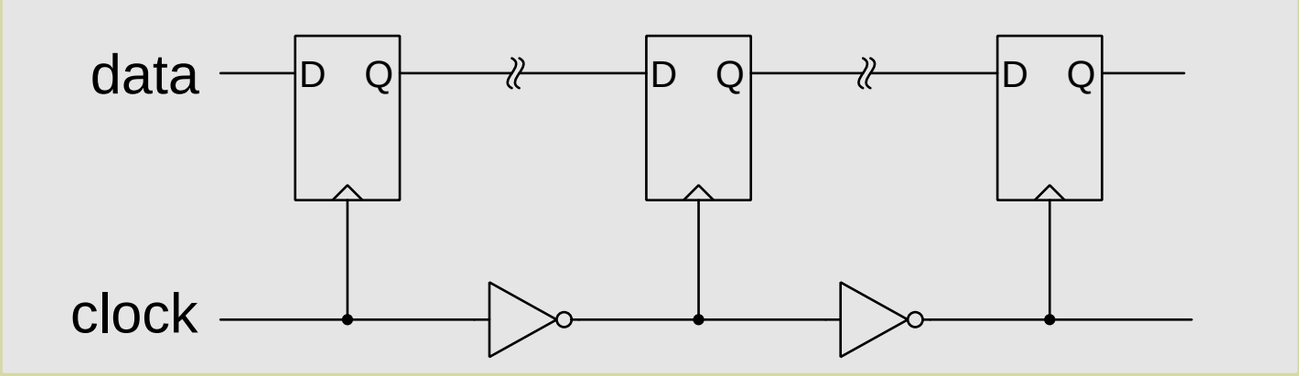
Negative-edge data capture



unreliable without precision delay data



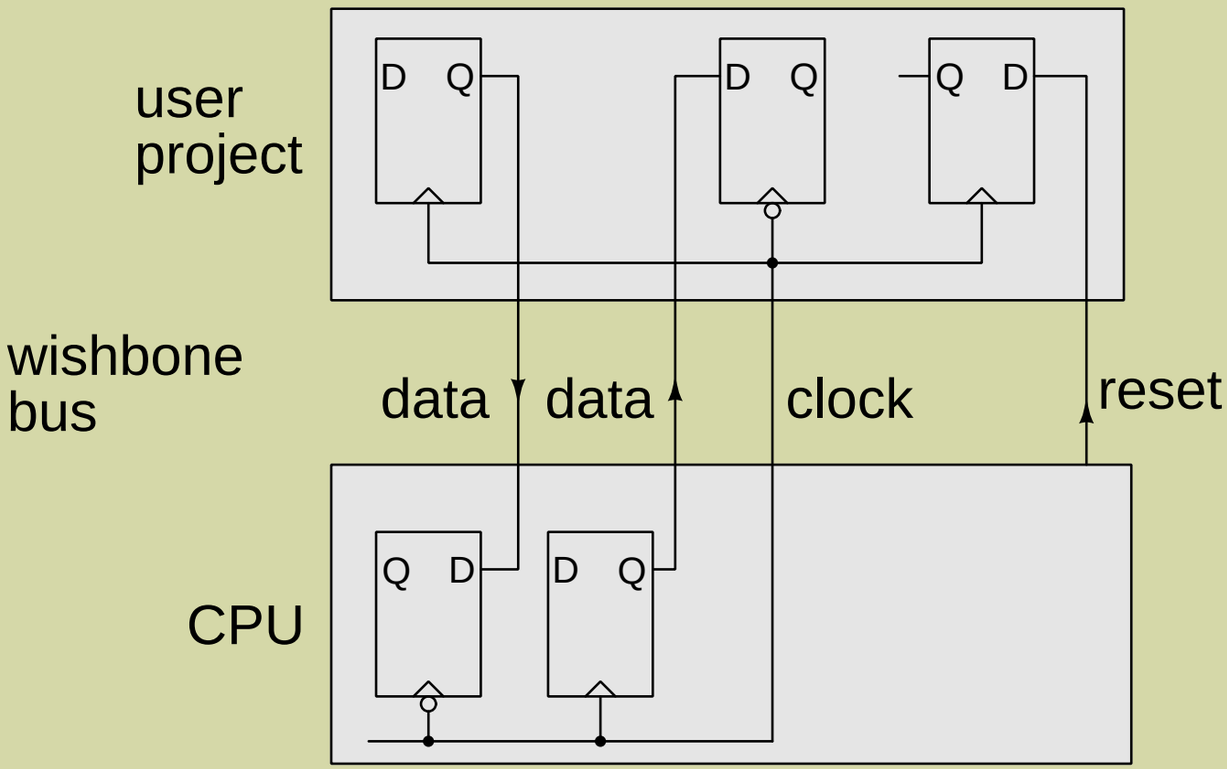
reliable, but with extra wiring, and slower



twice as slow but always robust

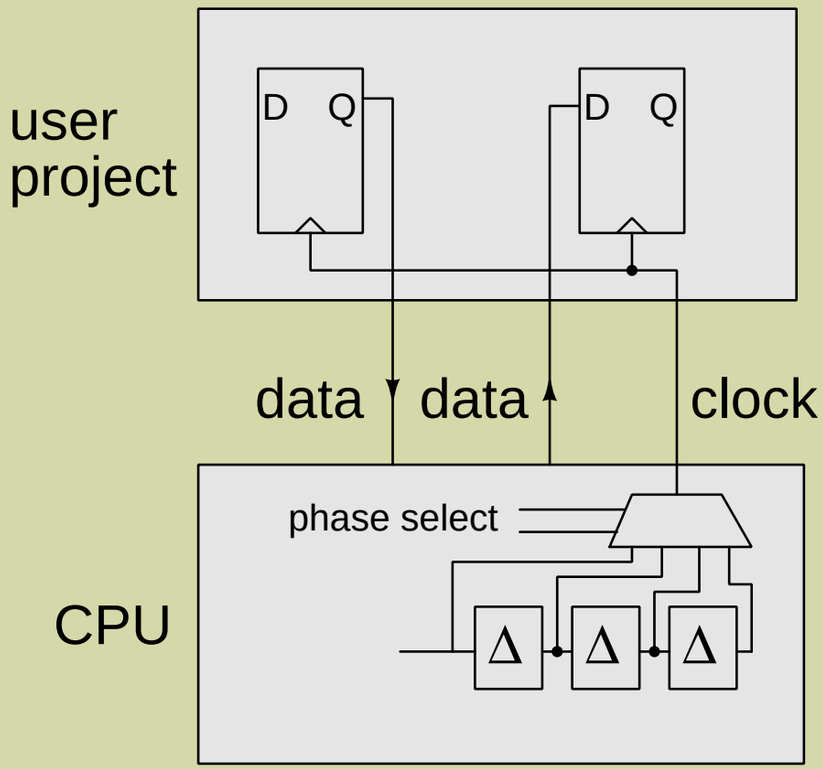
Robust Design Methodologies

Phase-programmable clock/data skew



negative−edge data capture

source: Tobias Strauch
open−source−silicon Slack workspace



programmable clock phase

source: Dinesh Annayya
Riscduino project

Principles of Paranoid Design

Thanks for listening!

Q&A