Wires:

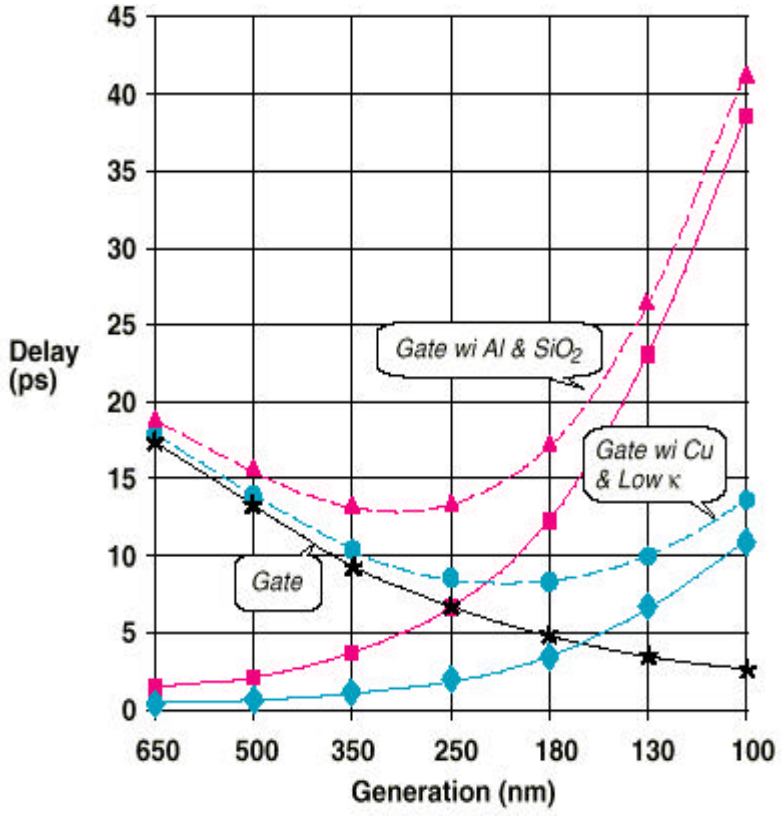
A User's Guide

Mark Horowitz, Ron Ho, Ken Mai
Computer Systems Laboratory
Stanford University
horowitz,ronho,demon@stanford.edu

The Buzz is VLSI Wires are Bad

- A lot of talk about VLSI wires being a problem:
  - Delay
  - Noise coupling
- What are the characteristics of electrical wires?
  - What is the baseline that other technologies should improve upon
- And what does it really mean?
  - To CAD developers
  - To architects

A very popular figure



Outline

- Predicting the future
- Scaling technologies
  - Devices
  - Wires
- Designer responses
  - Wire engineering
  - Repeaters
- Implications
  - CAD Tools
  - Architectures
- Conclusions

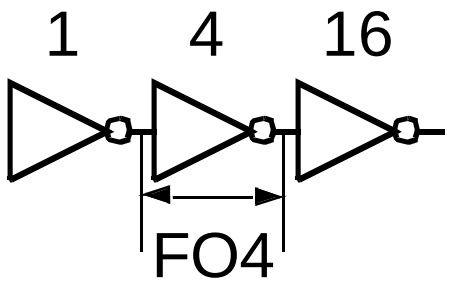
Predicting the Future (without making a fool of yourself)

- Is very difficult
  - The only guarantee is:

    The future will happen, and you will be wrong
- Two approaches
  - Think about limitations
    - SIA 1994 Roadmap
      - Limited oxide thickness, small clock frequency growth, etc.
      - Industry hit points above the curve
  - Project from current trends
    - SIA 1997 Roadmap
      - Allow miracles to occur, continue trends
      - Project clock rates higher than physically possible
- So use a range of technology scalings
  - Better chance of covering the correct answer

Logic Gate Speed

- Performance of wires is measured relative to logic gates
- How does the speed of gates depend on technology?
- Use a Fanout of 4 inverter metric



  - Measure the delay of an inverter with $C_{out}/C_{in} = 4$

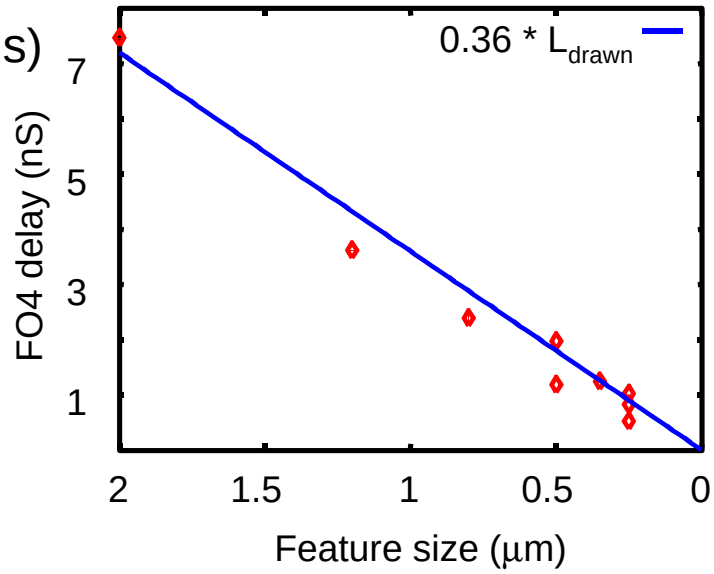
- Divide speed of a circuit by speed of FO4 inverter
  - Metric tracks pretty well, over process, temp, and voltage

FO4 Inverter Delay Under Scaling

- Device performance will scale
  - FO4 delay has been linear with tech

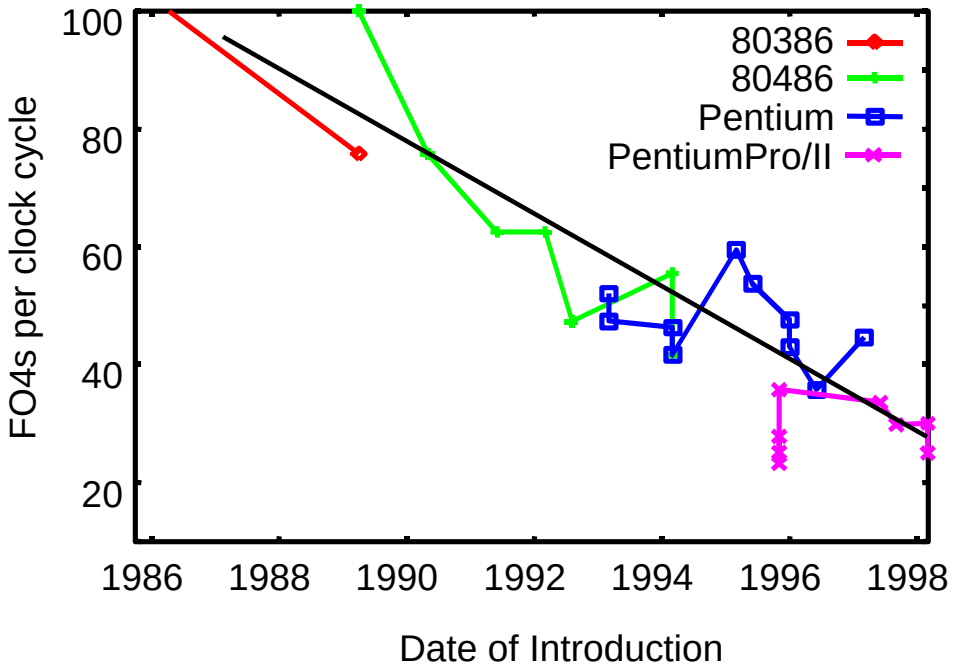
    Approximately 0.36 nS/μm*L_{drawn} at TT

    (0.5nS/μm under worst-case conditions)
- Easy to predict gate performance
  - We can measure them
    - Labs have built 0.04μm devices
  - Key issue is voltage scaling
    - Need to scale Vdd for power
    - Hard since Vth does not scale



Gates Per Clock

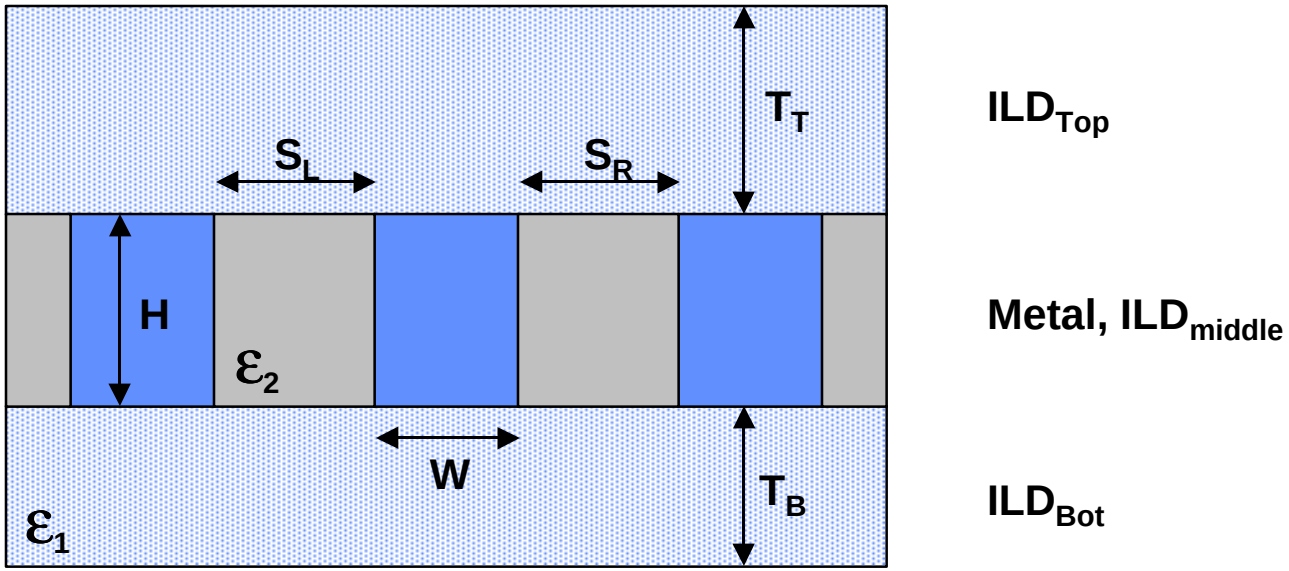
- How can we predict clock cycle times under scaling?
- Look at the number of FO4 inverter delays per cycle



- Current machines are in the mid twenties
- Overheads for clocks less than 16 FO4 become large
- Generating a clock less than 8 FO4 is very hard
- SIA roadmap has clocks less than 8 FO4
- We will use 16 FO4 for clock

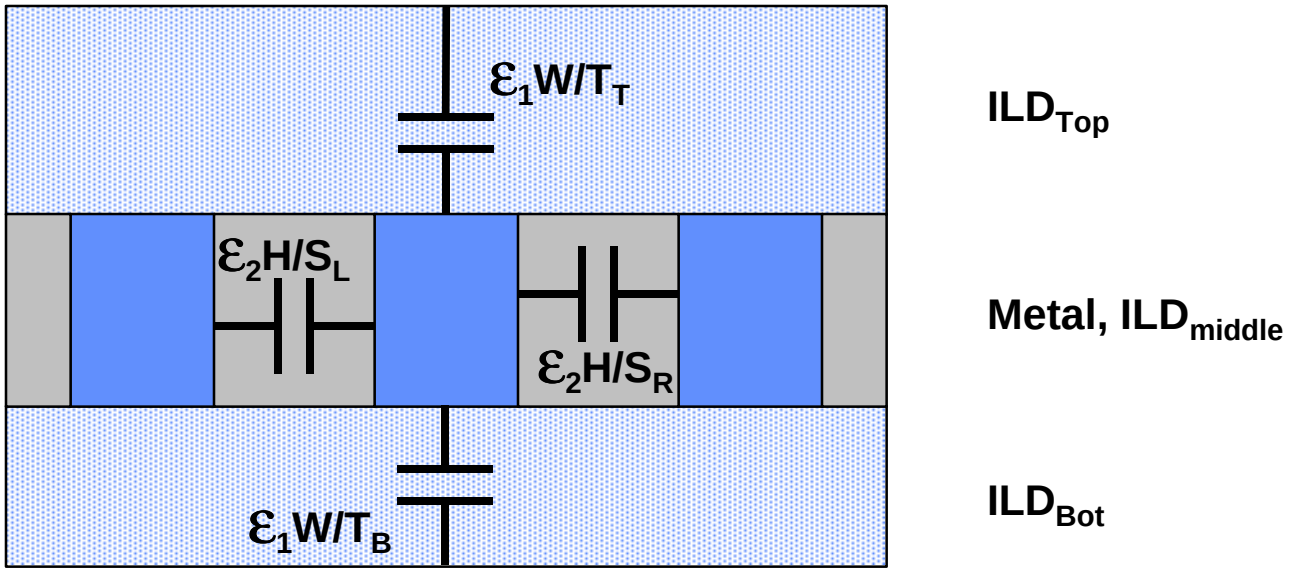
Wire Scaling

- More uncertainty than transistor scaling
  - Many options with complex trade-offs
- For each metal layer
  - Need to set H, T_T, T_B, ε_1, ε_2, conductivity of the metal



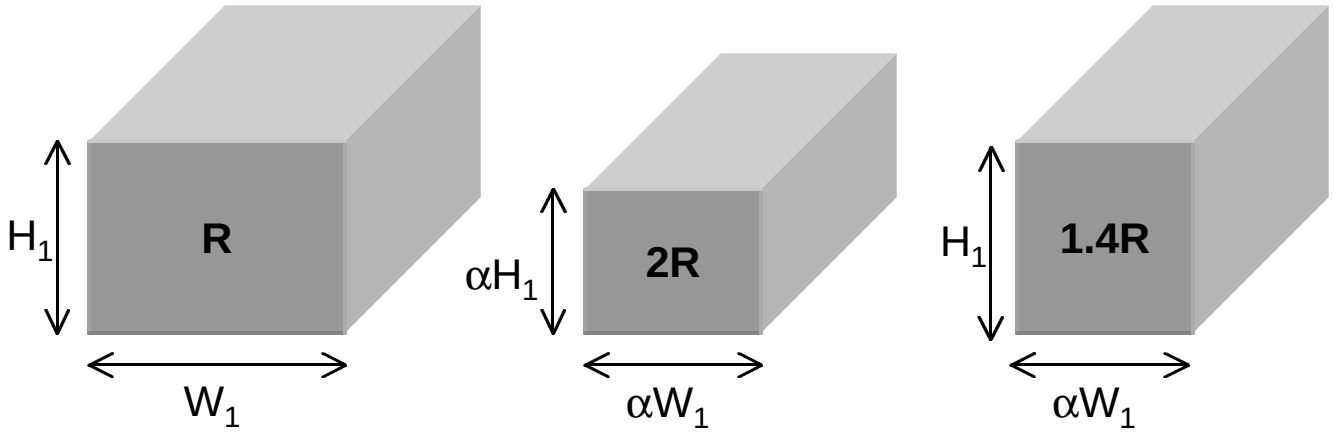
Wire Capacitance

- Capacitance per micron is roughly constant
  - Simple approx. = fringe (0.07fF/μm today) + 4 parallel plates
  - Depends only on the ratio of the parameters
  - Coupling becomes a large issue with increased εH/S ratio



Wire Resistance

- Resistance is simpler
  - $R/\mu m = \rho/wh$
  - Scales up as the technology shrinks
  - Main reason that wire height has not scaled much



- Tradeoffs between height, width and wire pitch

Wire Layers

Not all wiring layers have the same characteristics

- Today have three types of levels
  - M1
    - Finest pitch, highest resistance, local interconnection in a cell
  - M2-4?
    - Normal routing level, most of the wires
  - M5+
    - Thick coarse metal, used for global wires
    - When scaling forces thinner metal, create new top layer

Noise Issues

- Two main noise sources for wires
  - Capacitance coupling
  - Inductive coupling
- Capacitance coupling is mostly a nearest neighbor issue
  - High aspect ratio wires make this worse
  - Real push for low-κ dielectric between wires
- Inductive coupling
  - Is much more complex to analyze
    - Depends on where the return currents flow
  - Reduce these problem by design constraints
    - Gnd returns in buses, power and gnd planes (e.g. 21264)

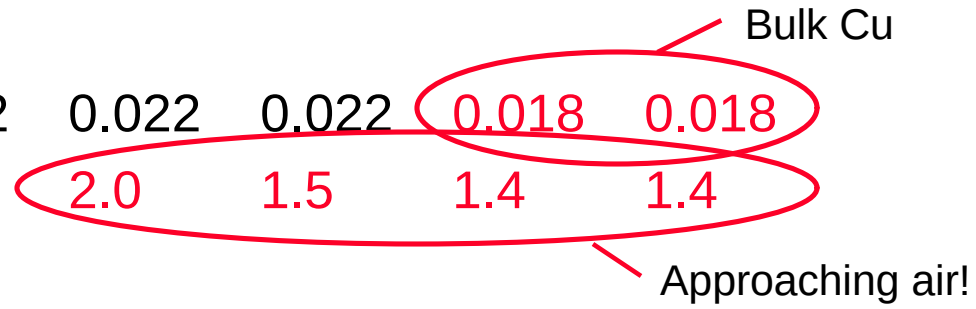
Delay Issues

Interested in how the delay of two different wires scale

- Fixed length (in microns) **----- Global wires**
  - The delay of this wire scales as α^{-1} to α^{-2} **α is 0.7**
  - D_{wire}/D_{gate} scales as α^{-2} to α^{--3}
- Scaled length (fixed # of gate pitches) **----- Local wires**
  - Same connection ability as before
  - Length of wire scales as α
  - The delay of this wire scales as 1 to α
  - D_{wire}/D_{gate} scales as 1 to α^{-1}
- All this assumes minimum width wires

SIA Wire Projections

Tech	0.25	0.18	0.13	0.10	0.07	0.05
Width	0.50	0.36	0.26	0.20	0.14	0.10
Spacing	0.50	0.36	0.26	0.20	0.14	0.10
Thickness	0.90	0.66	0.55	0.48	0.38	0.30
Aspect Ratio	1.8	1.826	2.118	2.4	2.705	3
ILD	0.975	0.75	0.54	0.48	0.405	0.315
Rho	0.022	0.022	0.022	0.022	0.018	0.018
Er	3.9	2.7	2.0	1.5	1.4	1.4
ps/mm^2	6.6	10	16	21	31	58

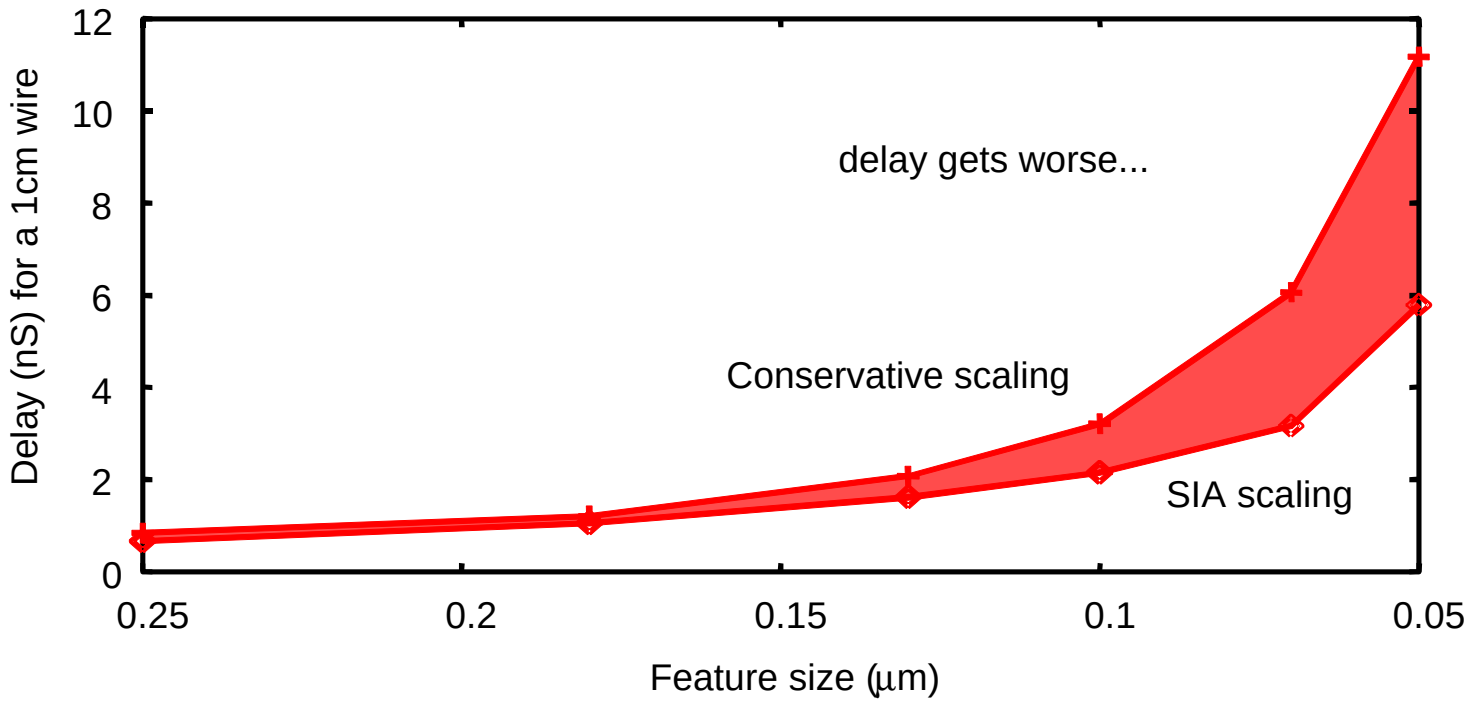


Alternative Wire Projections

Tech	0.25	0.18	0.13	0.10	0.07	0.05
Width	0.50	0.36	0.26	0.20	0.14	0.10
Spacing	0.50	0.36	0.26	0.20	0.14	0.10
Thickness	0.90	0.65	0.52	0.40	0.28	0.20
Aspect Ratio	1.8	1.8	2	2	2	2
ILD	0.975	0.75	0.54	0.48	0.405	0.315
Rho	0.033	0.022	0.022	0.022	0.022	0.022
Er	3.9	3.5	3.2	2.9	2.6	2.3
ps/mm^2	8.4	12	20	32	60	111

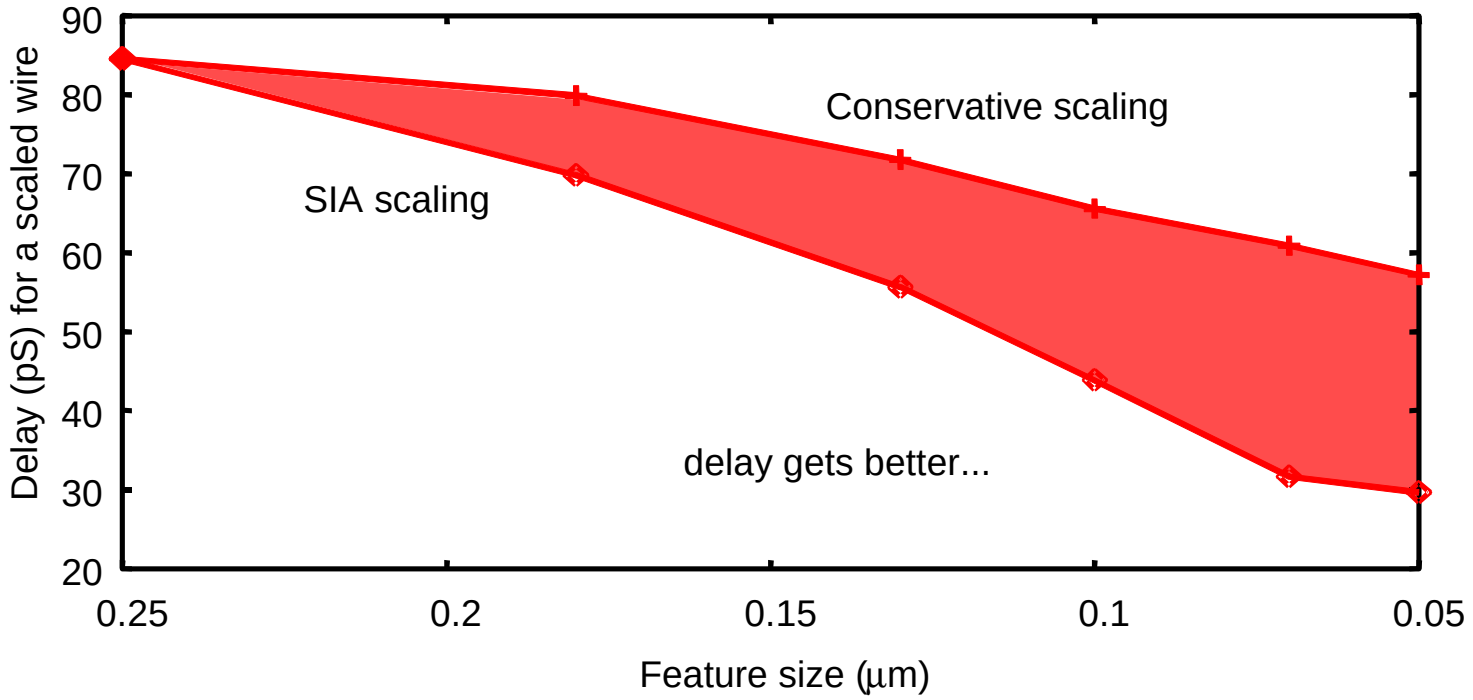
Delay of one kind of wire

- What is the delay of a fixed length (global) wire?
  - 1cm in this case



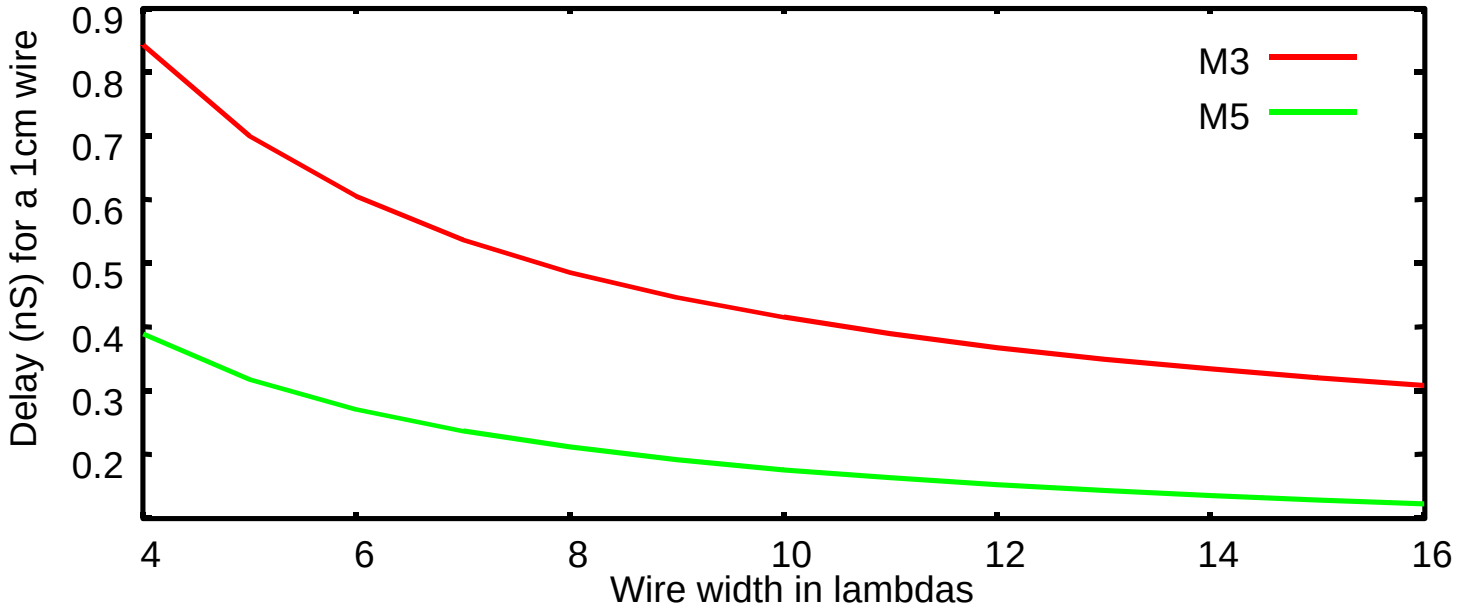
Delay of the other kind

- What is the delay of a scaled length (local) wire?
  - Length of a 50K gate block in this case



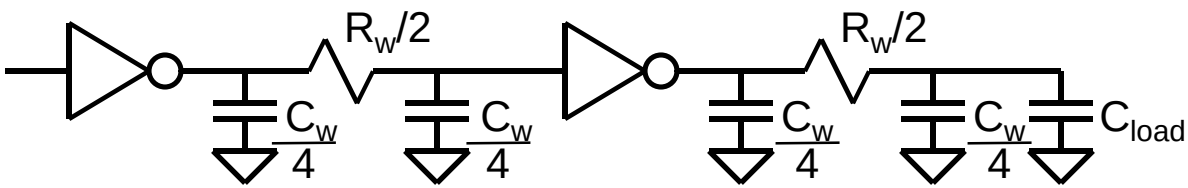
Designer Responses

- Wire engineering -- use wider wires or thicker wires
- Making wires wider will improve performance
  - Resistance = k/W; Capacitance = C_0 + b*W
  - RC = kC_0/W + kb; b << C_0 so increasing W helps quite a bit



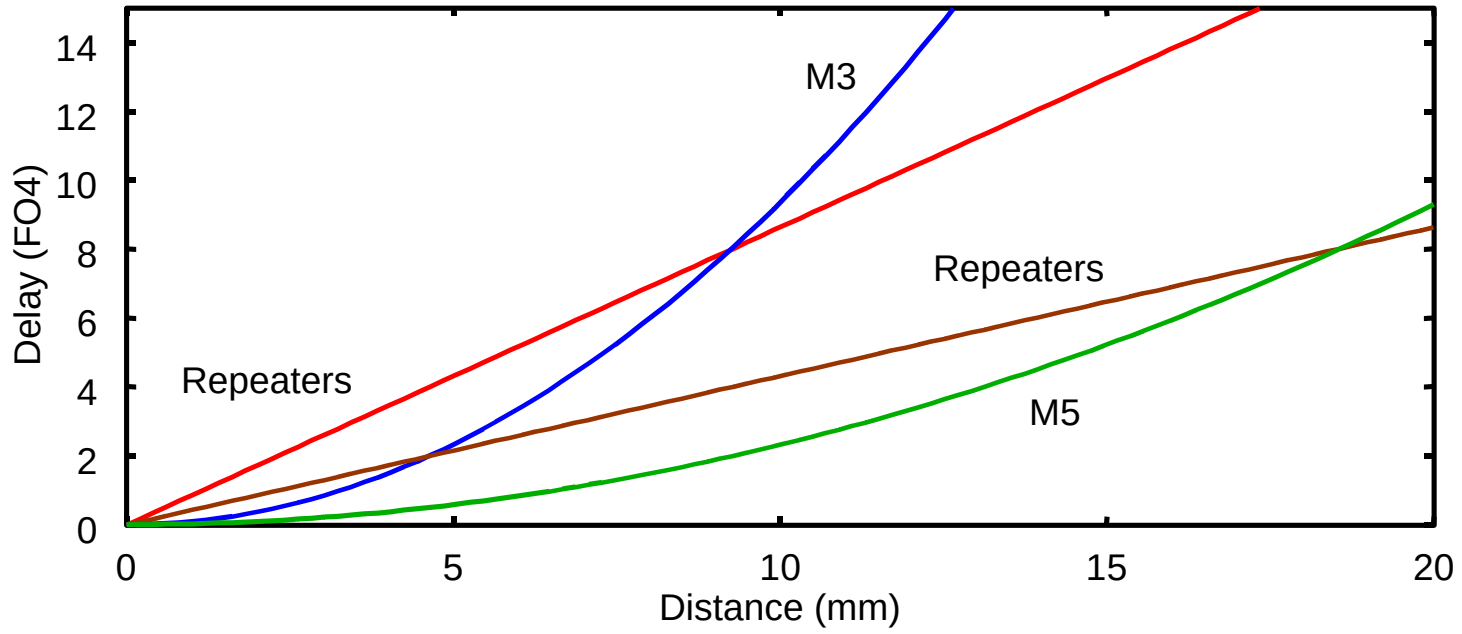
Designer Responses

- Circuit solution -- use repeaters
  - Break the wire into segments
  - Delay becomes linear with length
  - Signal velocity = k (FO4 R_w C_w)$^{1/2}$



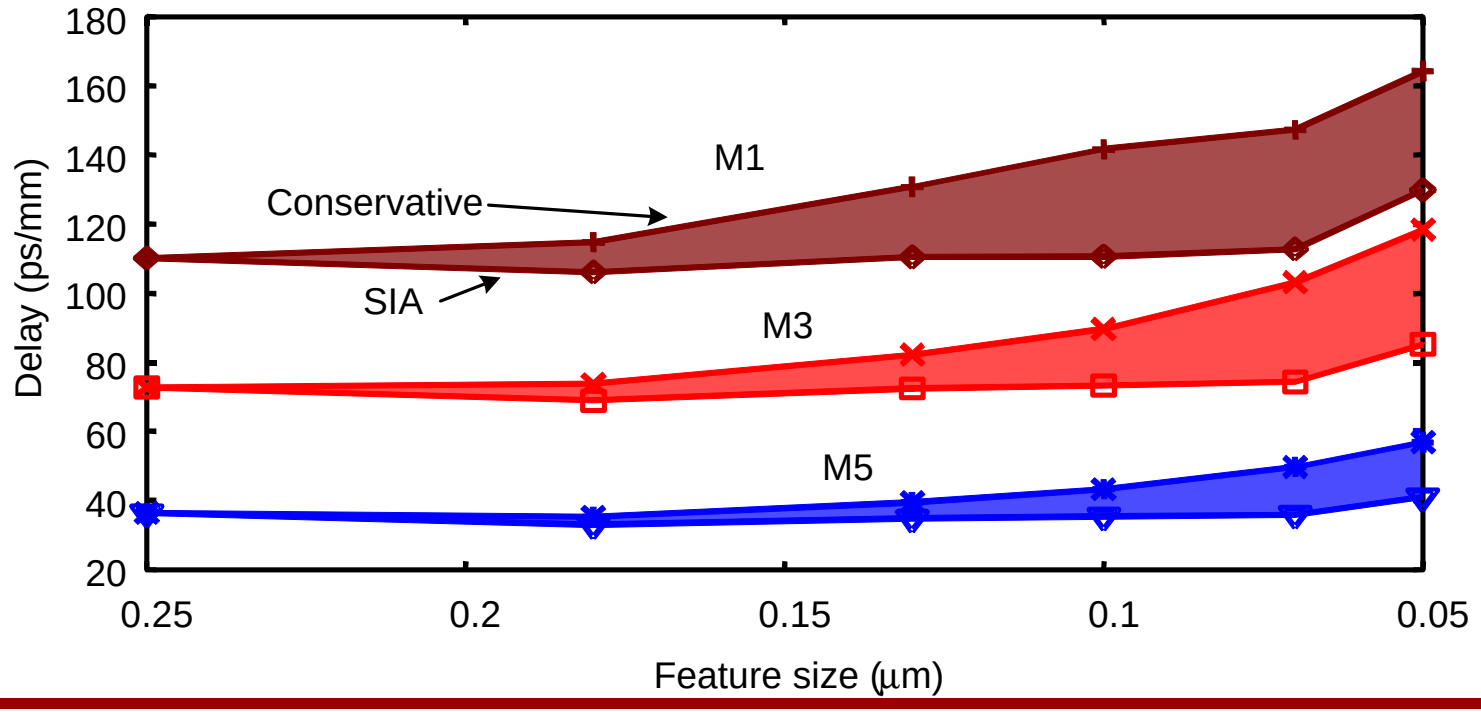
When to Add Repeaters

- Repeaters have overhead and can introduce logic complexity
  - Add when they reduce the delay
  - When the wire RC is greater than 8 FO4
- Plot for 0.25µm minimum width wires:



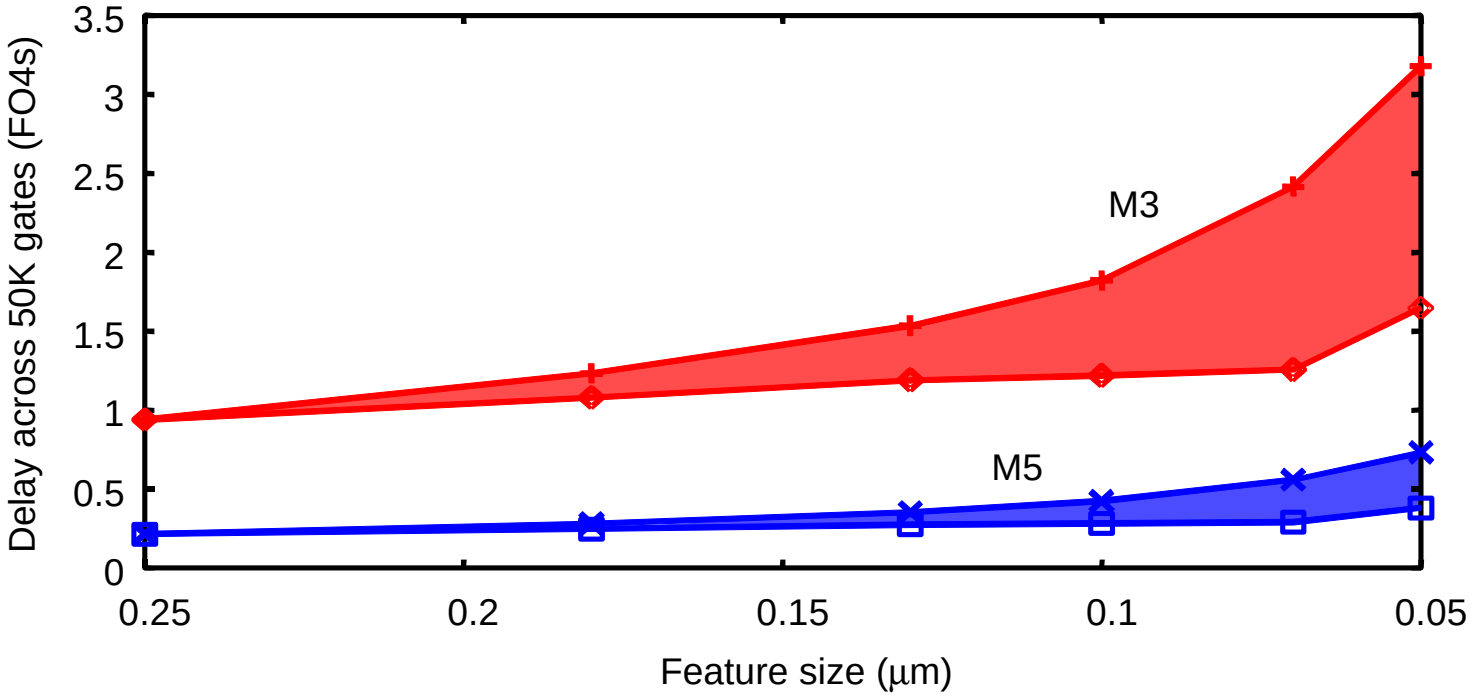
Signal Velocity for Repeated wires

- Under SIA scaling, pretty constant over many generations
- Under conservative scaling, slow change at sub-0.1µm techs
  - Makes wire delay increase slowly



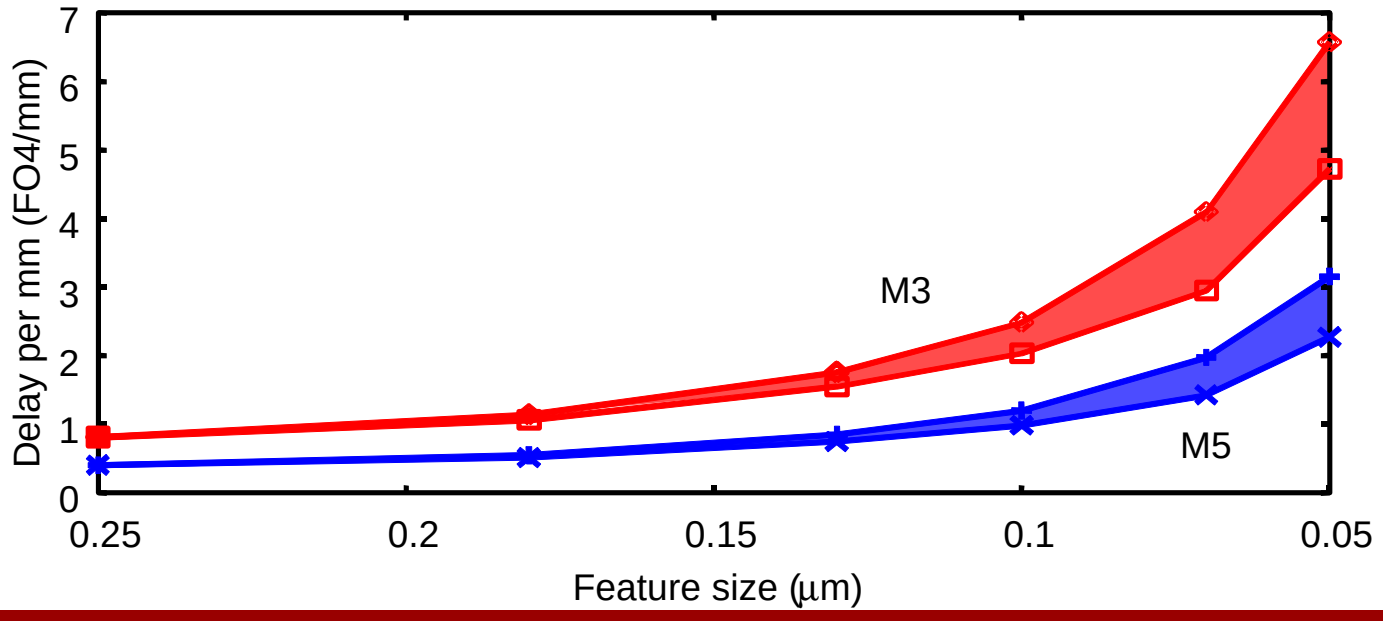
Local Wire Delay versus Gate Delay

- Local wire delay compared to gate delay is not that bad
- Bandwidth across a 50K gate block scales with technology
  - $1.6/L_{drawn}$ Tb/s (6.4 Tb/s in 0.25μm; ~ 30 Tb/s in 0.05μm)



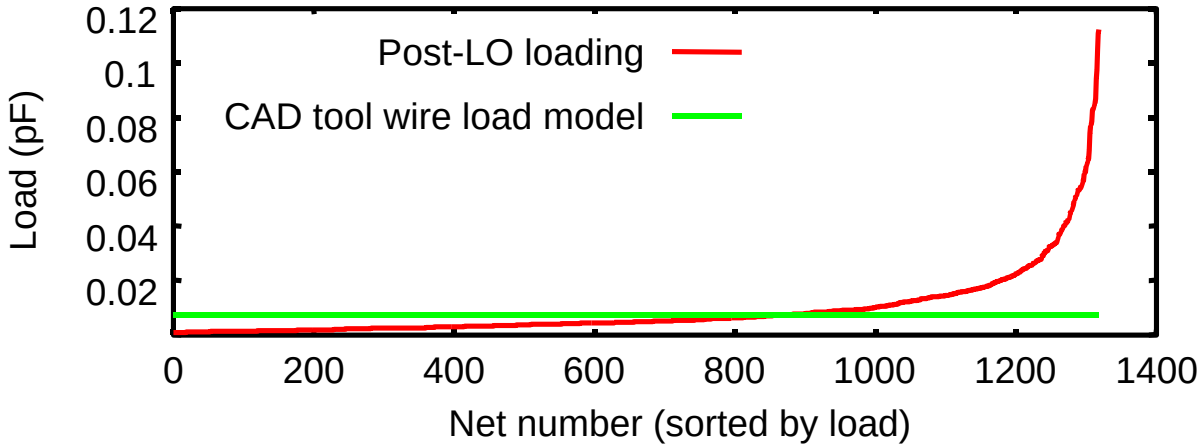
Global Wire Delay versus Gate Delay

- Delay of global wires compared to gates is becoming an issue
  - With repeaters, delay is roughly constant
  - But measured in FO4s it is still increasing
- With repeaters chip bandwidth scales as α^{-2}
  - 21 Tb/s in 0.25µm and doubling per generation



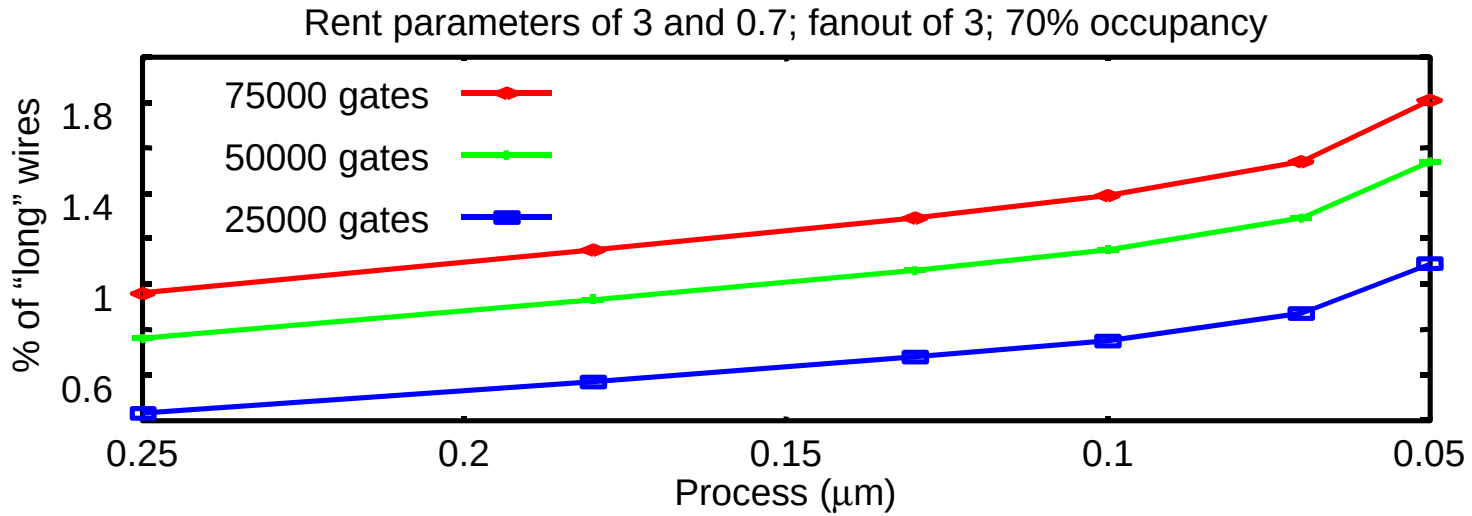
CAD Tool Implications

- Long wires are a problem for CAD tools
  - Actual loads differ from what CAD tools predict
  - Must be dealt with manually in CAD flows



- At first glance, the future looks promising
  - CAD tools work on chunks that have a fixed number of gates
  - We saw that local wires don't get much worse under scaling
    - Ratio of wire delay to gate changes slowly

CAD Tool Implications



- But ... big problem is chips are getting more complicated
  - More gates and more of those chunks for the CAD tools
  - More and more of the manual exceptions to deal with
- Something must give
  - Either design teams must grow, or
  - Tools need to handle more long wires (exceptional cases)

Chip Design

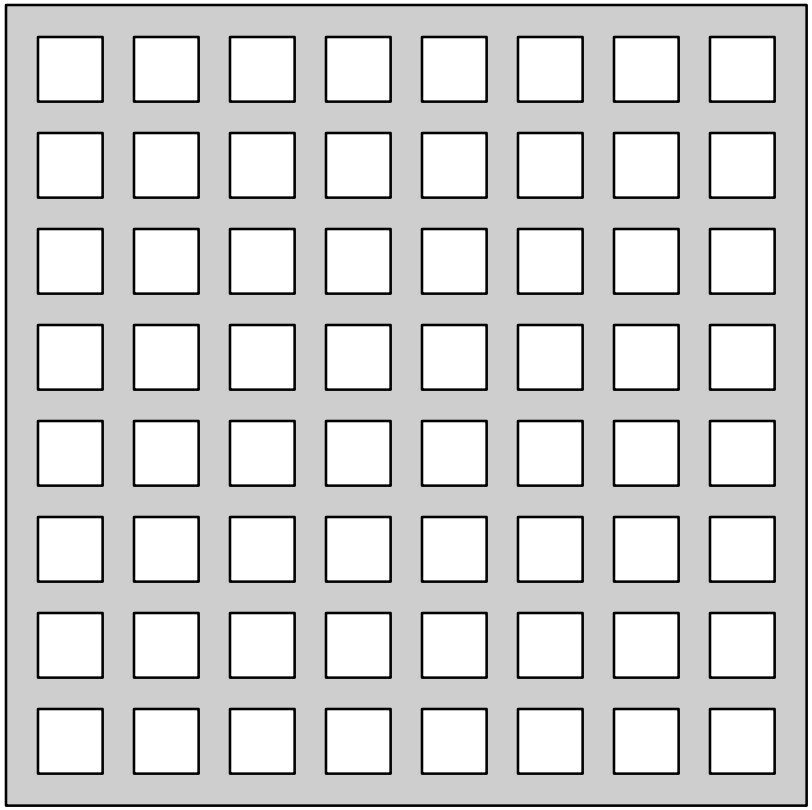
- Old model
  - Off-chip wires are expensive
  - On-chip wires are free
- New model
  - Off-chip wires are VERY expensive
  - Long on-chip wires are like old off-chip wires
  - Only local wires are free
- Architectures with large global communications are difficult
  - Model that has driven processor architecture
    - Superscalar
    - Out of order execution
    - Speculation
  - Need to think about partitioned machines

Approaching a Discontinuity

Current GP architectures are not sustainable:

- Still based on the free communication model
  - Maintaining a global shared resource model
  - Large, complex communication needed
- Poor modularity
  - Large design teams required
  - Huge design and verification costs

The Answer: Modular Computers



The Question:

How do you make a useful modular computer?

(Useful => Efficient -- cost, power)

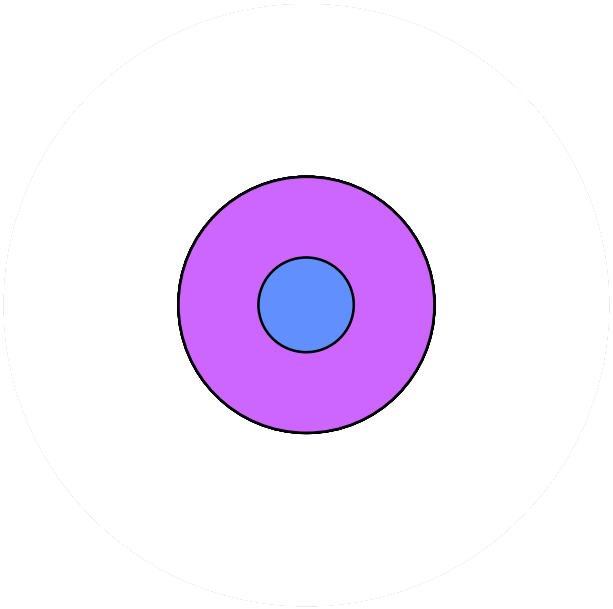
Slow Process

Need to change the way people think at all levels

- Hardware design
  - Function centric, write functions in Verilog
    - Wires are implicit, through variables
- Programs are even worse
  - All communication is through variables
  - Memories are great communication boxes
    - Any part of the program can read your output
- Algorithm design
  - Today focus is on efficient computation
  - Need to focus on efficient communication

Conclusion

- Wire are not that bad
  - Delay of local wires scale, Tb/s of bandwidth
- The real issue is complexity
  - With scaling the # of gates on a chip is growing exponentially



Old view: a chip looks small to a wire

How far I can go in 1 cycle

A chip's edge

New view: a chip looks really big to a wire

Conclusions

- CAD Tools need to deal with more wires
  - Need to handle a higher percentage of long wire cases
    - Need to be more wire aware
- The number of gates on a chip exceeds the number of gates that can communicate in a clock cycle
- Chip designers need to think differently
  - Communication on chip is no longer free
  - Back to the future -- it looks like board/box design
    - But the wires are (relatively) faster