

Impact of Local Interconnects on Timing and Power in a High Performance Microprocessor

Rupesh S. Shelar

Low Power IA Group

Intel Corporation, Austin, TX

Marek Patyra

Enterprise Microprocessor Group

Intel Corporation, Hillsboro, OR

ISPD 2010
San Francisco, CA



Objective

- To convey the severity of the delay/power impact and the challenges it presents to physical design



Agenda

- Introduction
- Impact on Timing
- Impact on Power
- Conclusions

Why Look at Interconnects Closely

- Unlike transistors, they do not perform computation
- They just transfer information from one place to another
- Paying power/timing cost for interconnects yields nothing, unlike that for transistors
- Secondary effects: Cause area growth, delay penalty, yield issues indirectly due to routing congestion

Motivation I: Interconnect Delay

- Interconnects known to contribute significantly to path delays
- For intra-block paths, exact numbers probably not known, as these vary depending on the block-size, design style
- Many academic studies (Keutzer, Horowitz, Cong, Saraswat, Saxena) exist (and 1000s of papers start the introduction section with “interconnect delay scaling...”)
- Most based on combination of some (small) design data and simplistic assumptions about scaling and do not solely focus on data from real design, for example, high performance microprocessor core

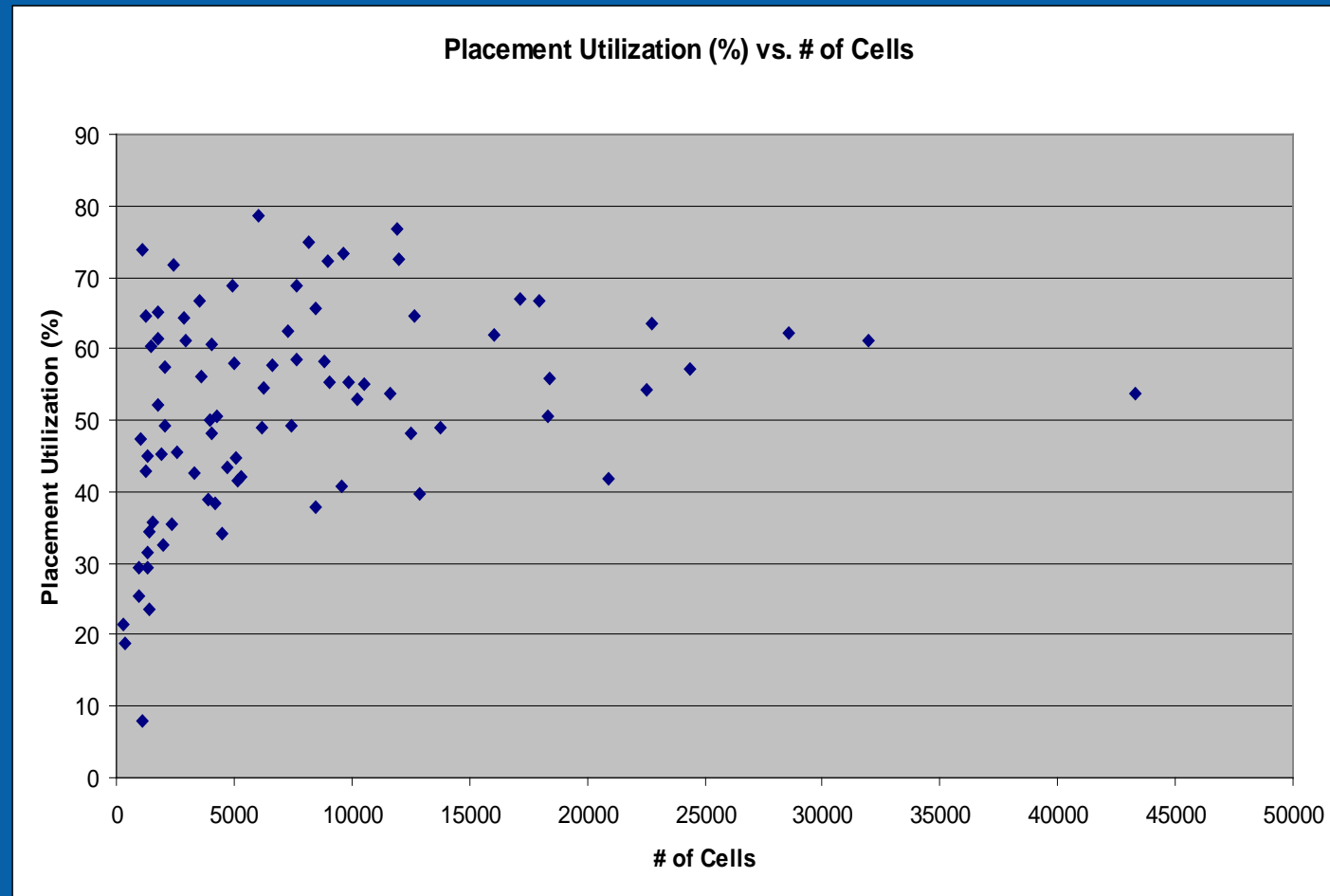
Motivation II: Power in Local Interconnects

- More than 70% of power in datapath and control logic blocks
- 60% of the total power is dynamic/glitch
 - 66% of the total dynamic power in local, *i.e.*, intra-block, interconnects (Source: SLIP'04 paper, based on a microprocessor study)
- Still relatively less attention paid on power dissipation in interconnects

About Data

- Delay/power data from blocks in high performance microprocessor core [Kumar et al., JSSCC 2008] in 45 nm technology
- Blocks implemented using different design Styles
 - RTL-to-Layout Synthesis (RLS), aka random logic synthesis
 - Mostly automatic (using vendor/in-house tools); write RTL, partition, and run tools/flows
 - Design quality determined by algorithms, tools, flows, parameters; supposedly poor utilization, or sparse layouts
 - Structured Data Paths (SDP)
 - Mostly manual; extract regularity using hierarchies, draw schematics, hierarchical placement and routing
 - Routing can be done flat; supposedly high utilization, or dense layouts
- RLS (SDP): 86 (133) blocks; cell count more than 600 (700) K
- Local interconnects:
 - RLS uses, mostly, M2 to M5, mostly minimum width, flat routing
 - SDP uses M2 to M7, different widths, hierarchical routing
- Delay/Power impact due to interconnects inside standard cells is considered as cell-delay/power contribution in this study

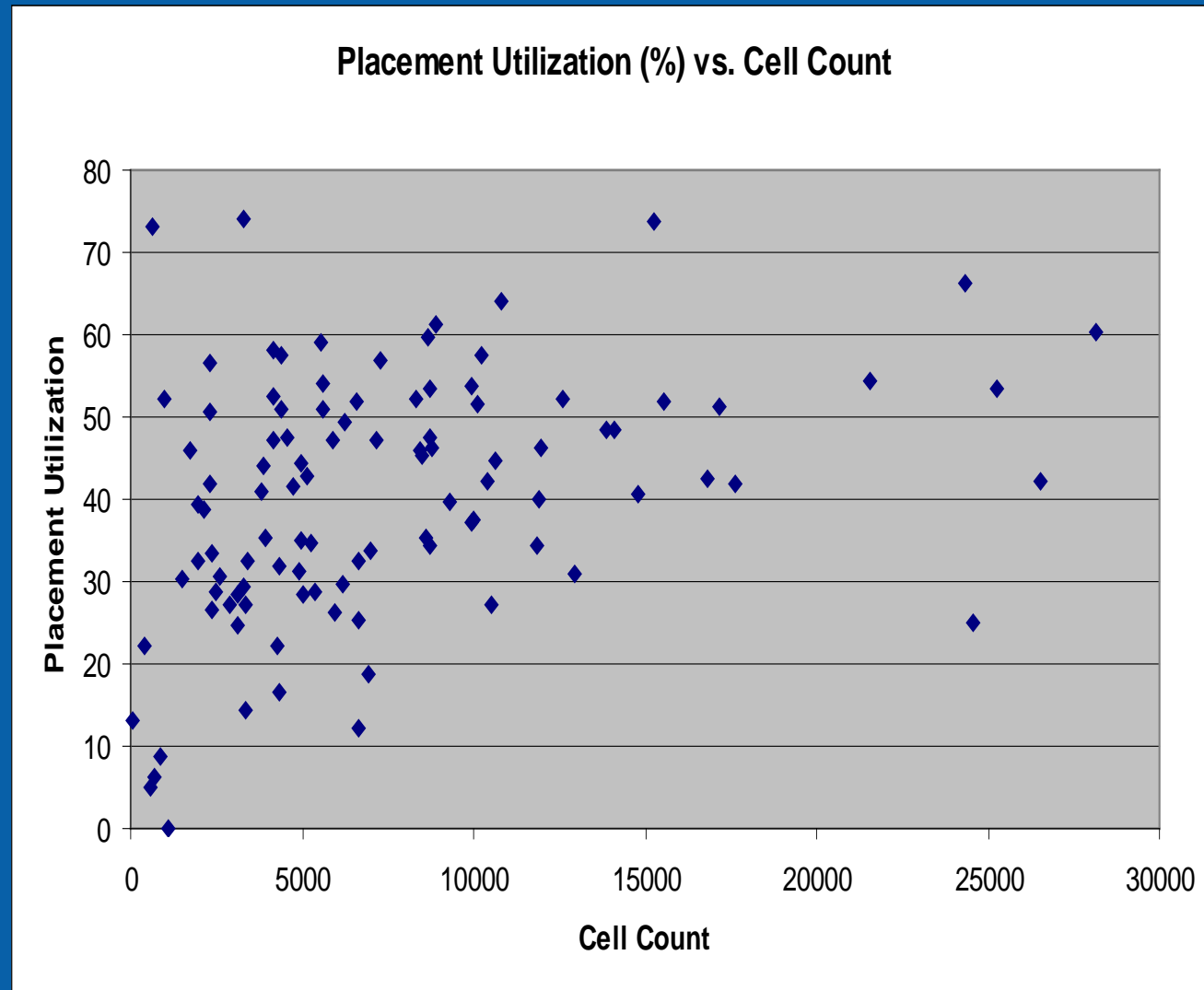
Utilization in RLS



Utilization = std. cell area/block area

- Avg. utilization: 51.69%
- Varies from 7% to 78%
 - Utilization varies significantly for blocks with < 5000 cells, possibly because of floorplan; for blocks with > 15000 cells, varies between 40 to 70%
 - Higher than 70% utilization blocks fairly difficult to converge
- Avg. block size: 7817, varies from 323 to 43298
- Reasons for low utilization:
 - Difficult to route and converge timing due to congestion, if the utilization is higher
 - Synthesis/placement not doing good job?
 - Space for ECOs: even if we assume generous 10% white space, 60% utilization may still be considered low

Utilization in SDP



Utilization = std. cell area/block area

- Utilization varies from 0.07% to 74%
- Avg. Utilization: 40.40%
- Avg. block size: 7542 cells
- The SDP layouts are not denser than RLS; reasons:
 - Routing congestion caused “artificially” by the hierarchies
 - Even with flat routing, it is not clear why, and how much, the congestion/utilization may improve (net ordering problem)
 - Matching bit-widths?
 - ???

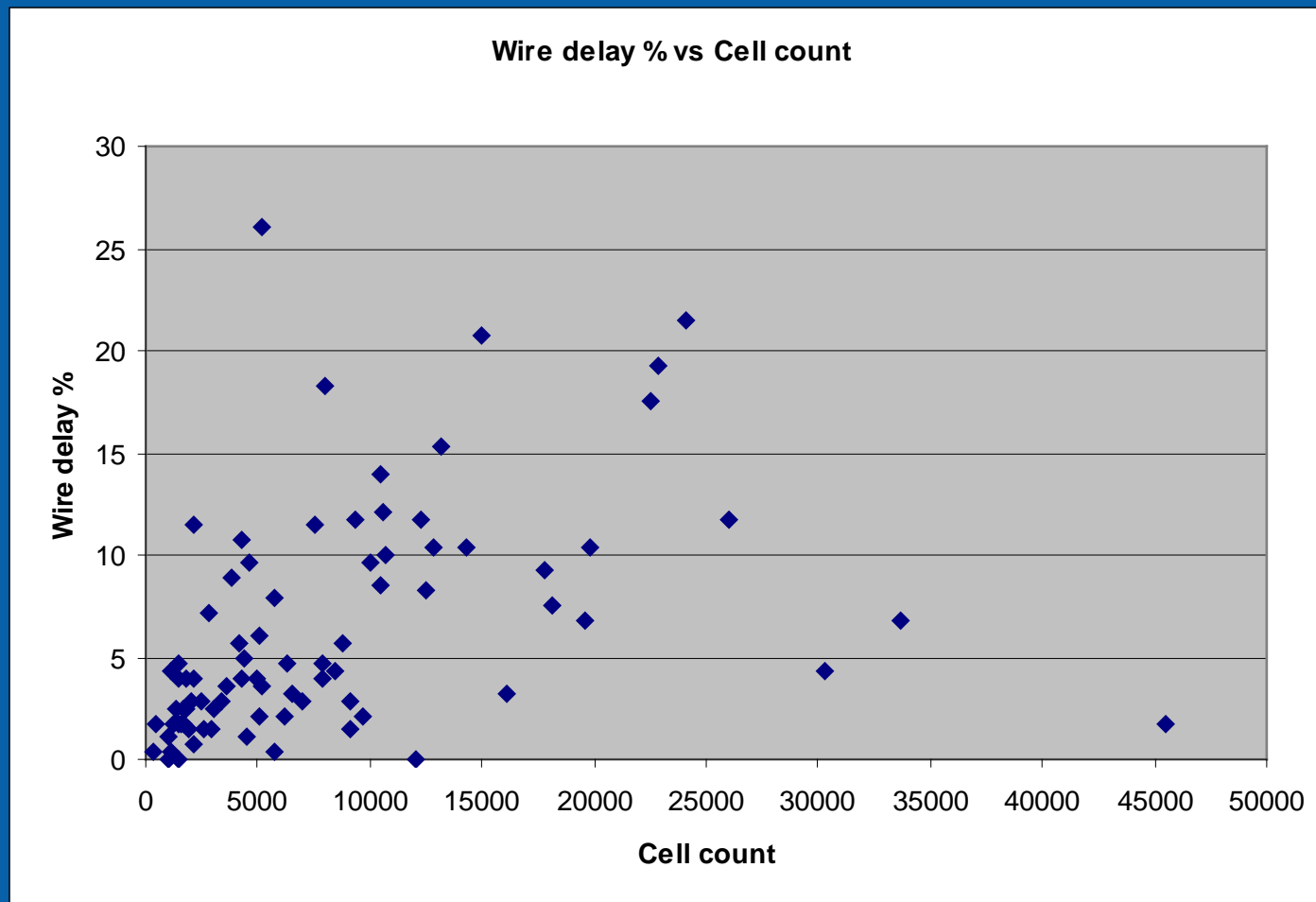
Agenda

- Introduction
- Impact on Timing
- Impact on Power
- Conclusions

Impact of Interconnects on timing

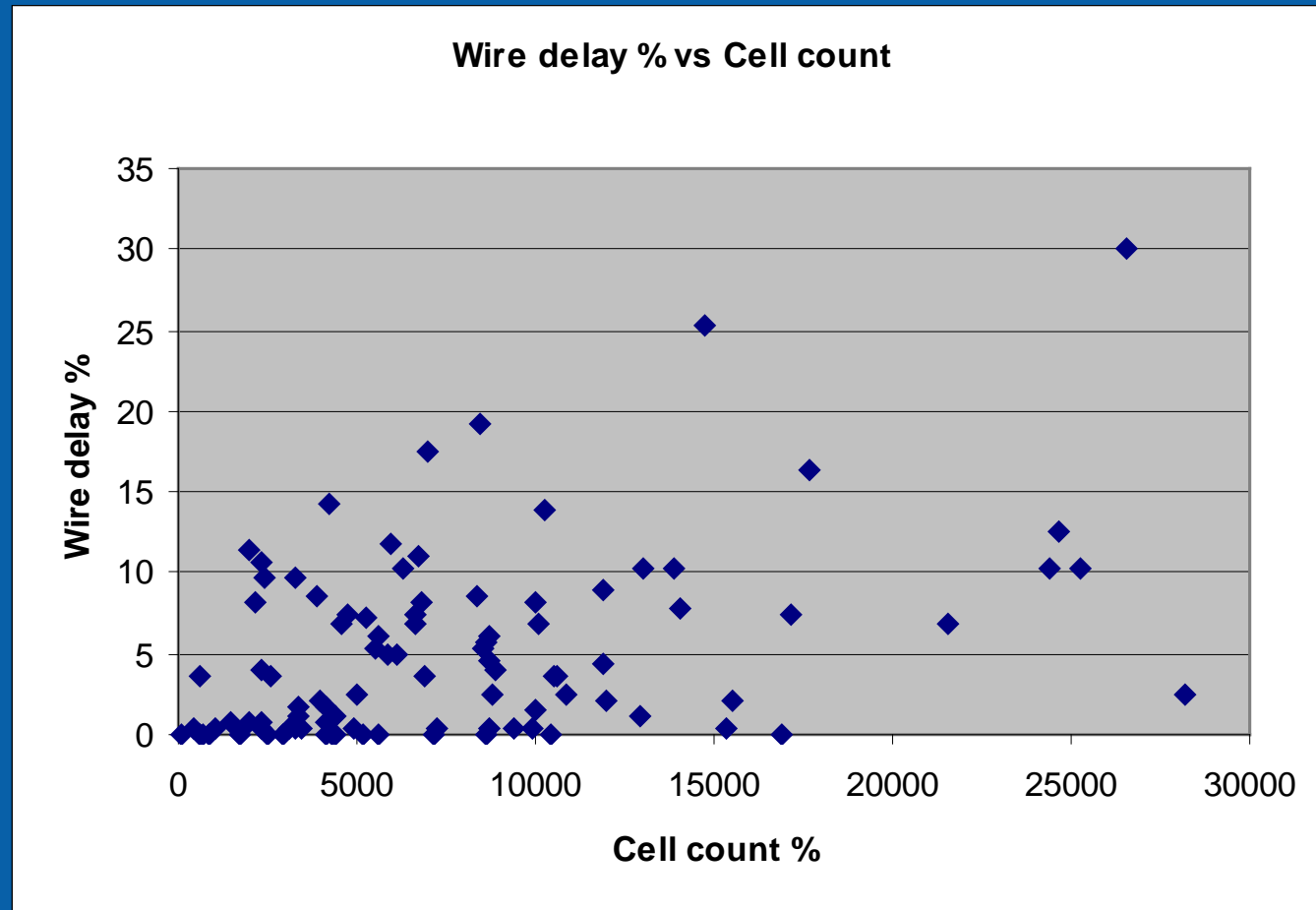
- For max timing, interconnects contribute in terms of
 - Wire delay
 - Slope degradation (slows down receivers)
 - Cell-delay degradation (extra cap to drive)
 - Cumulative effect of above 3 on path delays
 - Delays due to repeaters (inserted for timing/slope/noise)
- Chose 3 metrics on the worst internal paths:
 - Wire delay
 - Interconnect impact (obtained by setting $R=C=0$)
 - Repeater delay
- Why internal paths: should exclude the effect of timing constraints on primary i/os on synthesis flows (RLS)/manual design (SDP)
- Why worst paths: determines operating frequency

Wire Delay on Worst Paths in RLS blocks



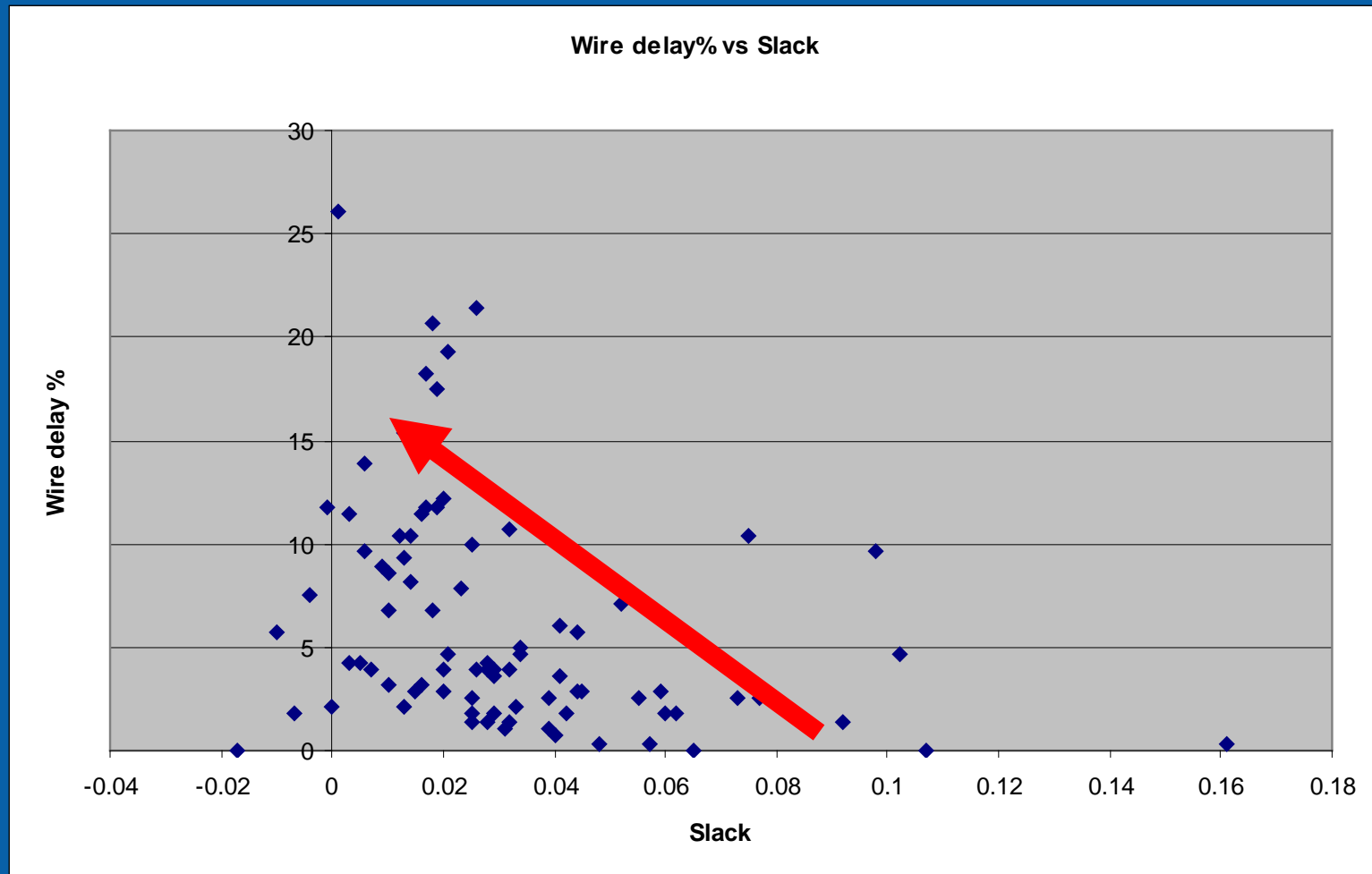
- Varies from 0 to 26% of cycle-time
- Average wire delay: 6%
- Excludes repeater delay and cell-delay/slope-degradation

Wire Delay on Worst Paths in SDP Blocks



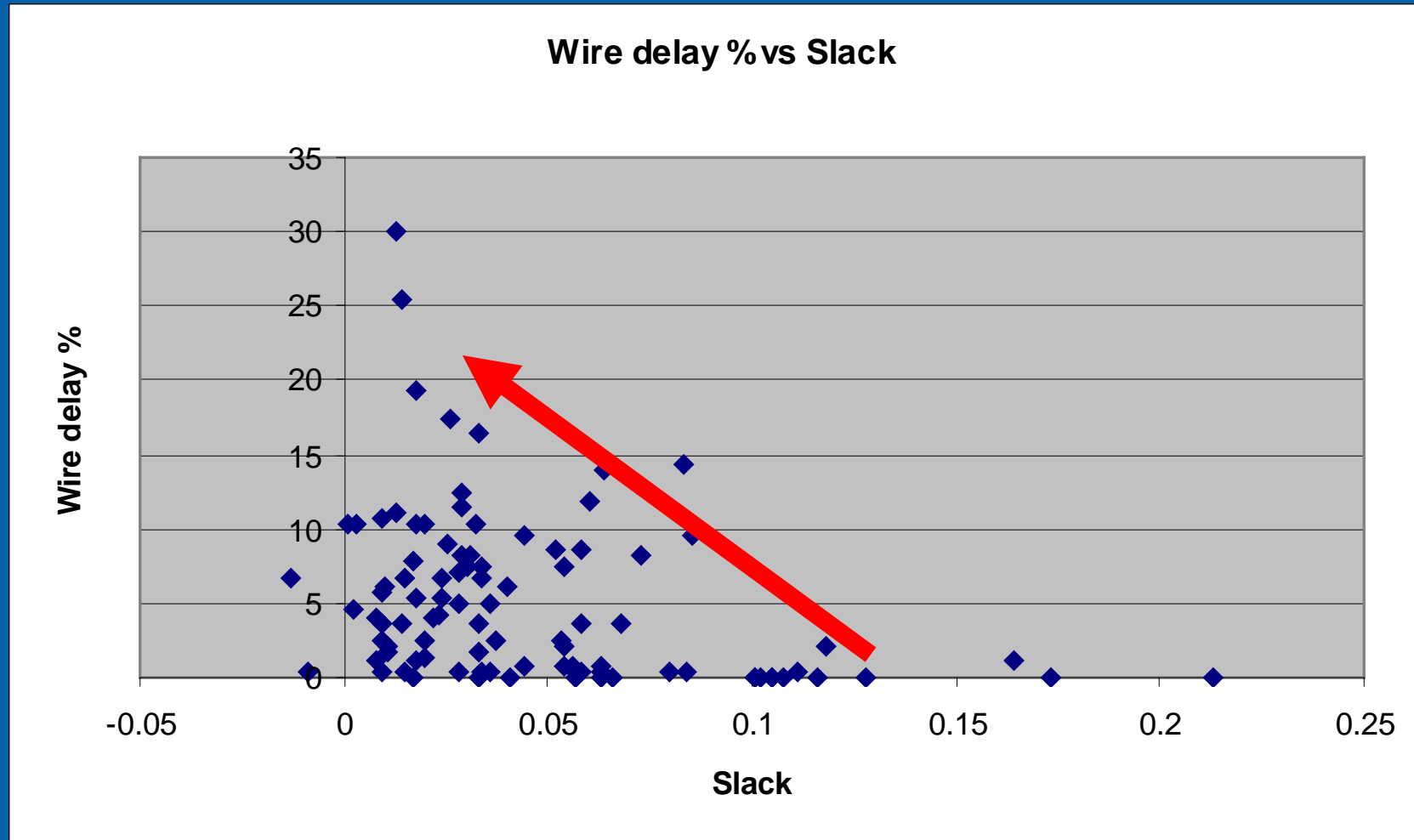
- Varies from 0 to 30%
- Average wire delay: 5%
- Several blocks with 0 wire delay on internal critical path implies careful design
- Excludes repeater delay and cell-delay/slope-degradation

Wire Delay vs. Slack for RLS blocks



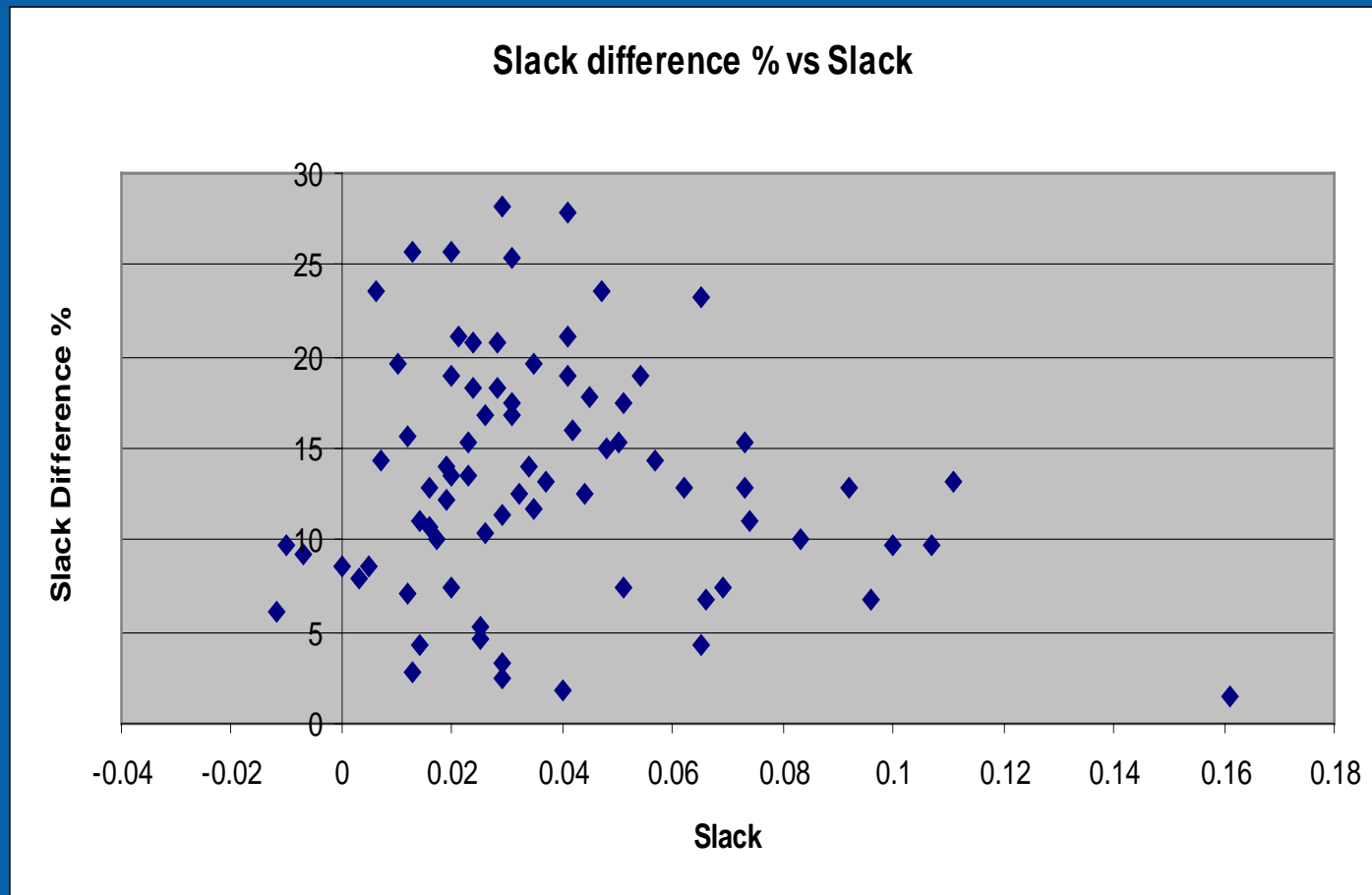
- Wire delay component increases as slack decreases
- Critical paths interconnect dominant ones

Wire Delay vs. Slack for SDP blocks



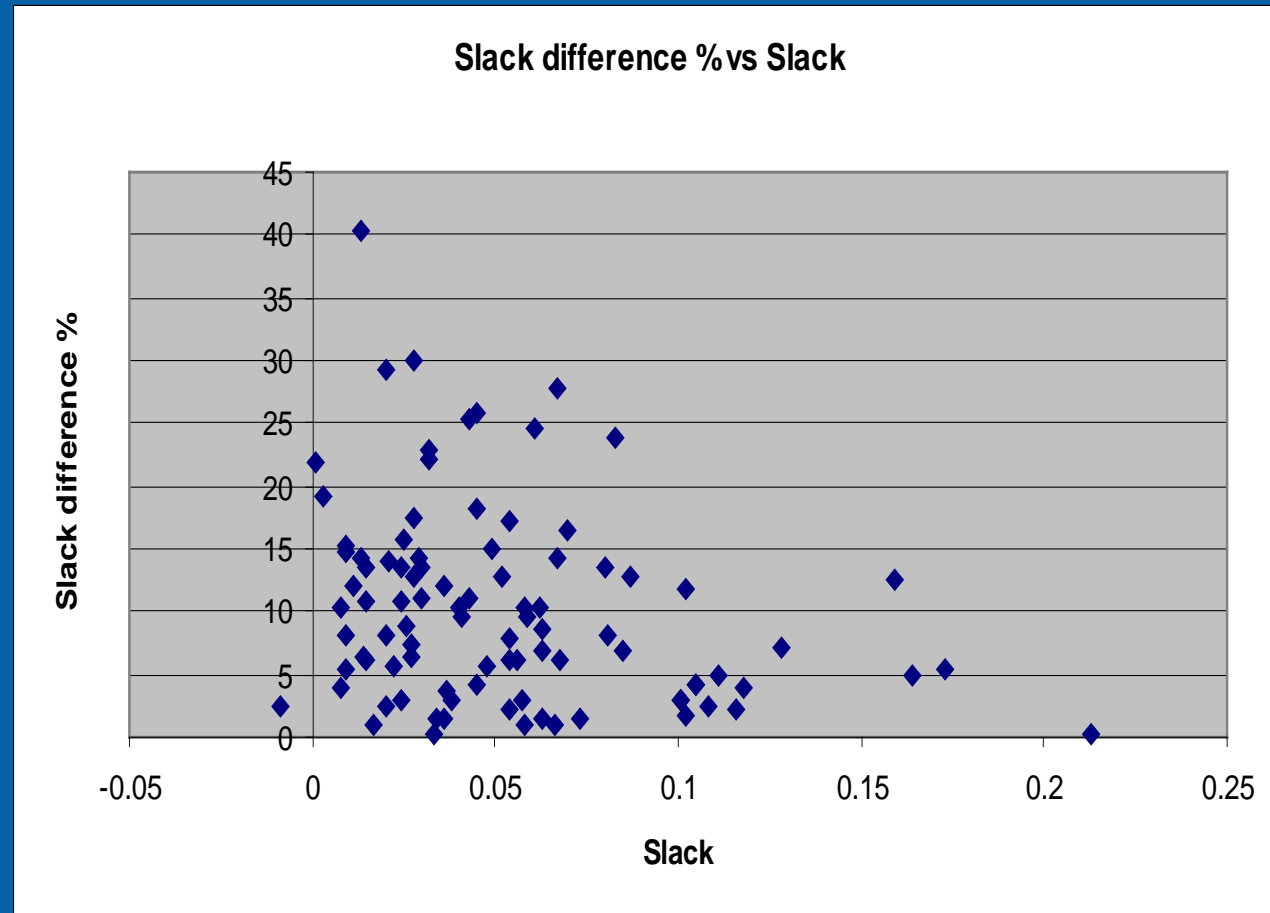
- Wire-delay component increases as slack decreases
- Critical paths interconnect dominant ones

Interconnect Delay Contribution on Internal Paths in RLS blocks



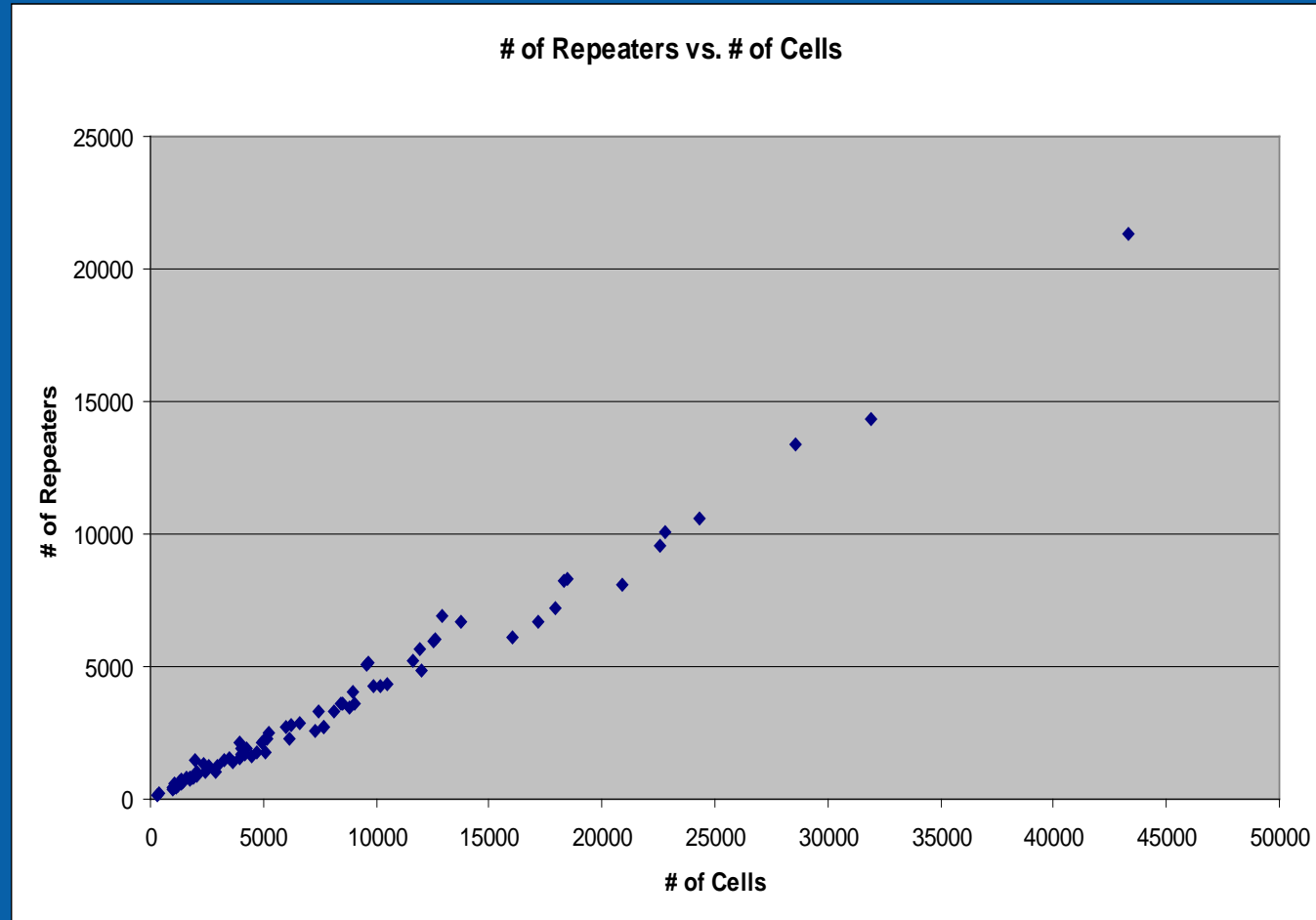
- How much would the timing improve, if $R=C=0$ for local interconnects
- Measured as the slack difference on the worst internal paths by setting $R=C=0$
 - Includes cumulative effect of wire delay, slope, cell-delay degradation
- Varies from 0 to 27%; average 13%
- Average impact slightly more than twice the average wire delay
- Excludes repeaters delay

Interconnect Delay Contribution on Internal Paths in SDP blocks



- How much would the timing improve, if $R=C=0$ for local interconnects
- Slack difference varies from 0 to 40%
- Average slack difference 9%
 - Smaller average implies that for many blocks the worst internal path were cell-delay dominated (consistent with wire delay slide for SDP)
- Average impact close to twice the average wire delay
- Excludes repeater delay

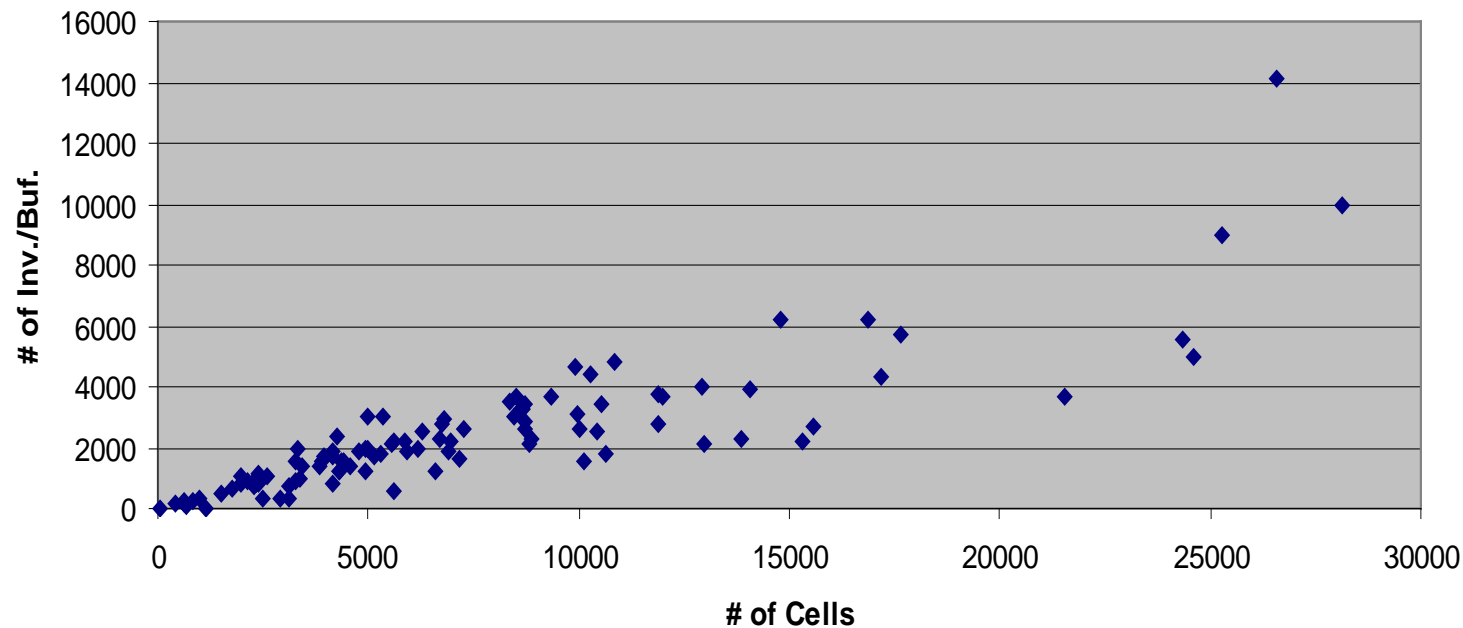
Repeater Count in RLS blocks



- Varies almost linearly with block-size
- Repeater count varies from 183 to 21315
- Out of 641002, 176205 (27.48%) inverters and 106346 (16.59%) buffers
- Inv./buf. contribute to ~44% of cell count
- Synthesis possibly did not do a great job

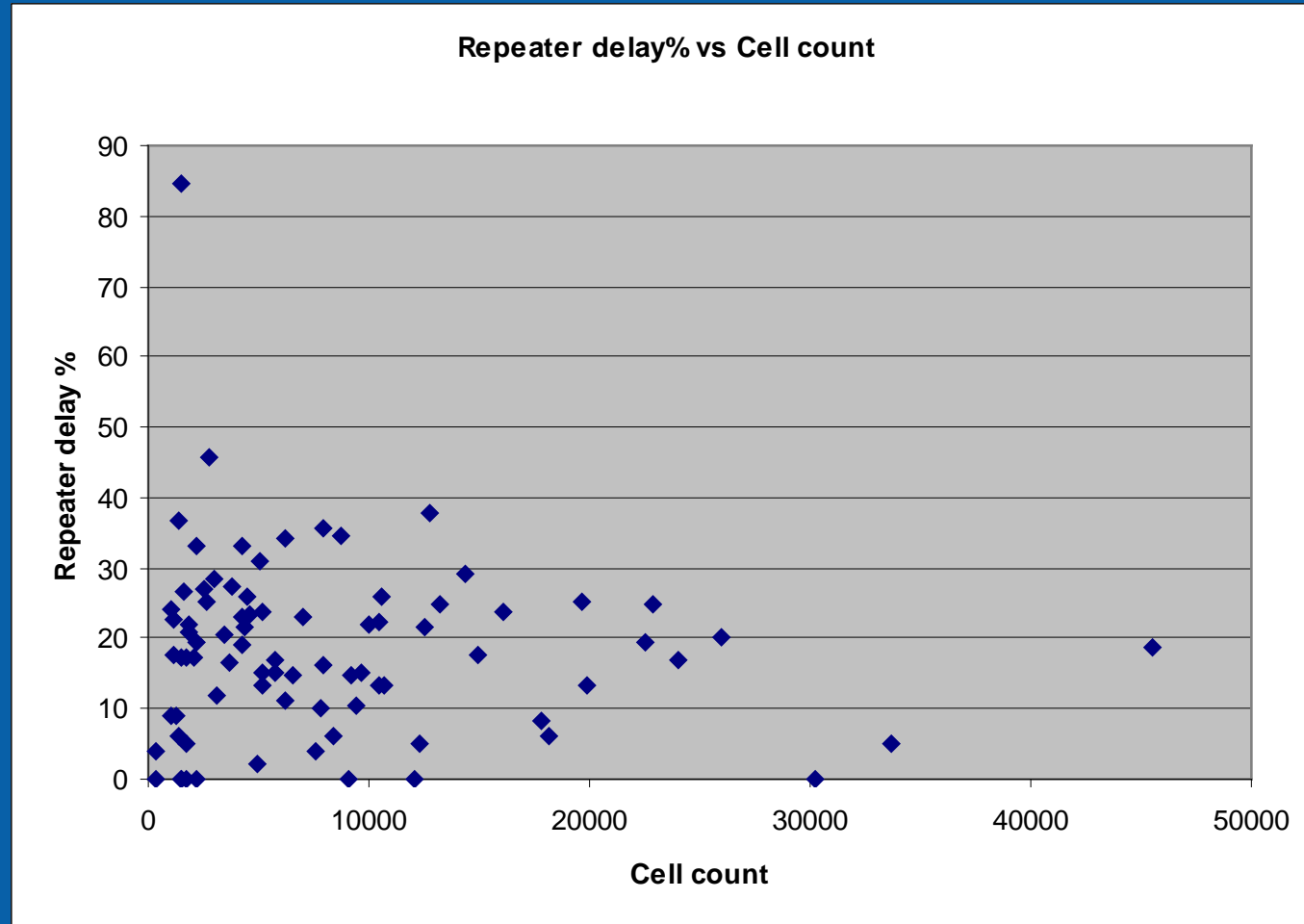
Repeater Count in SDP blocks

of Inv./Buf. vs. # of Cells



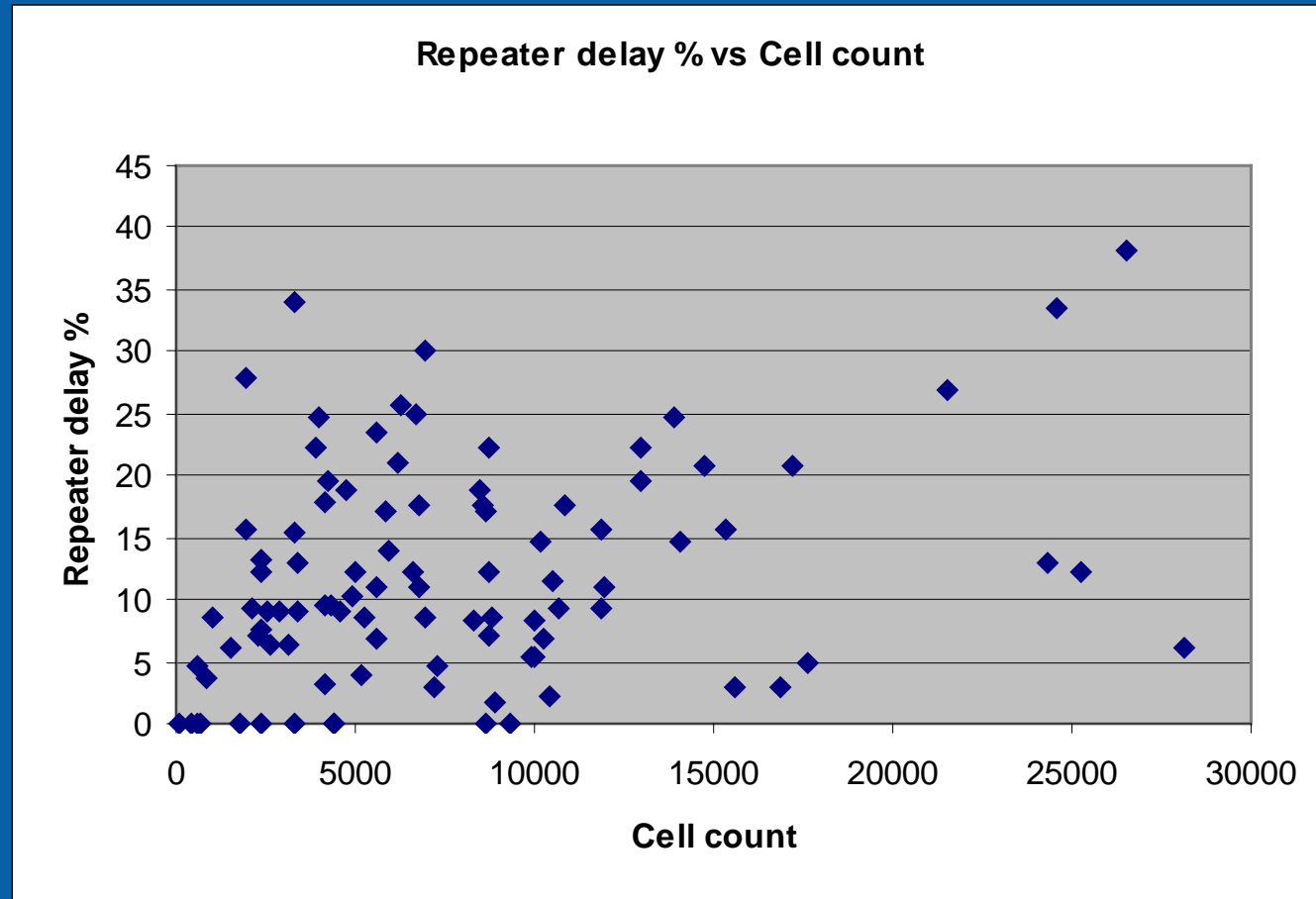
- Increases with cell-count, but spread is larger than that in RLS
 - Depends on how different DEs do schematic, buffer insertion
 - # of buffers not necessarily increasing as linearly with cell count as in RLS; DEs used them sparingly as compared to tools
- Buffer count varies from 0 to 14089
- Out of 770306, 177037 (22.98%) inverters and 68069 (8.83%)
- Inv./buf. contribute to ~31% of cell count; 13% better than RLS

Repeater Delay in RLS blocks



- Varies from 0 to 45%
- Average repeater delay: 19%
- Includes both, inverter and buffer delay

Repeater Delay in SDP blocks

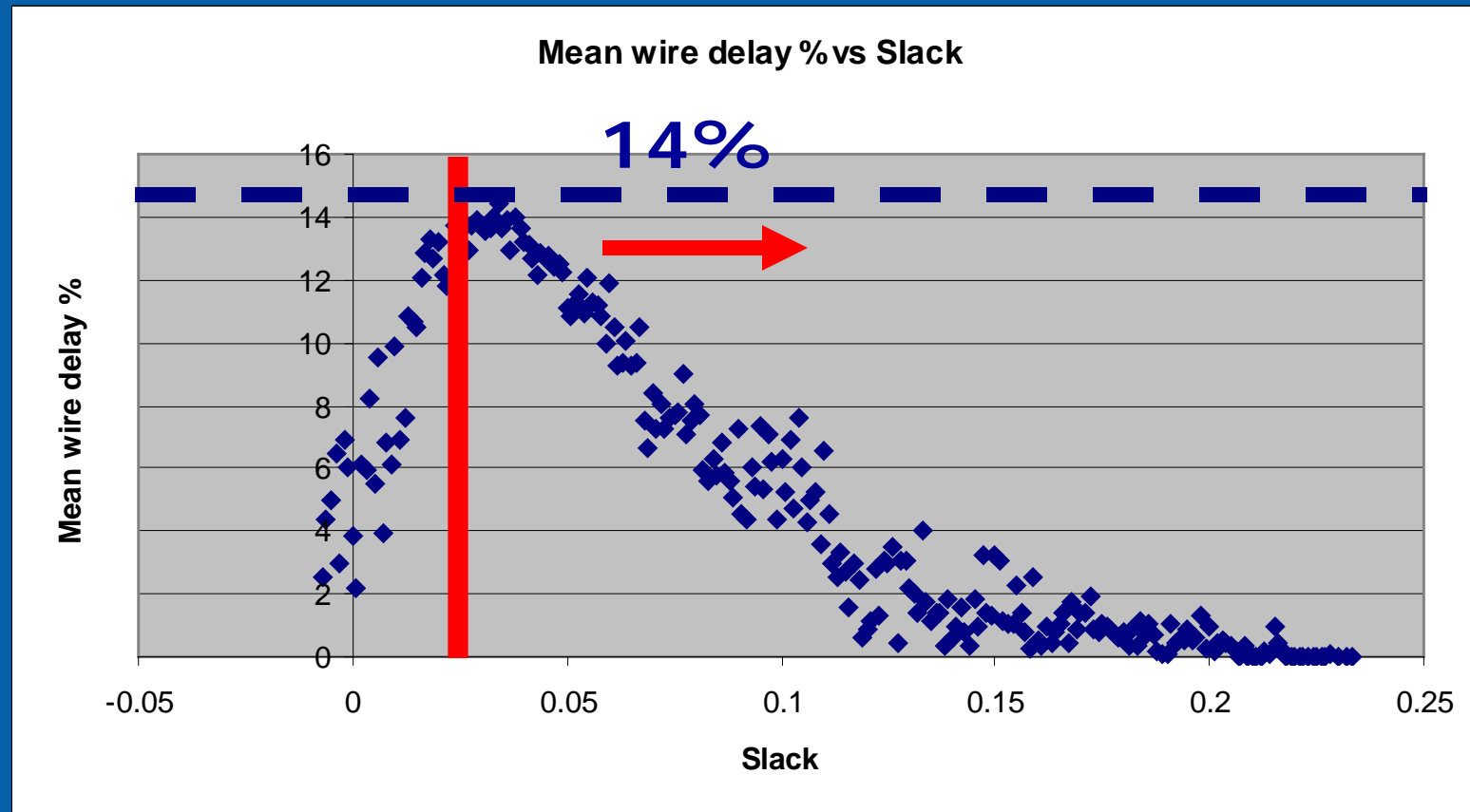


- Varies from 0 to 38%
- Average repeater delay: 11%
- Includes both, inverter and buffer delay

Summary of Observations so far

- Interconnect delay dominance regardless of design style
- Secondary effects, slope-/cell-delay degradation as big as wire delay
- Repeater count more than 40% and linear in the size of blocks
- Repeater delay contributes as much as wires
- SDP design with more manual control better than synthesis

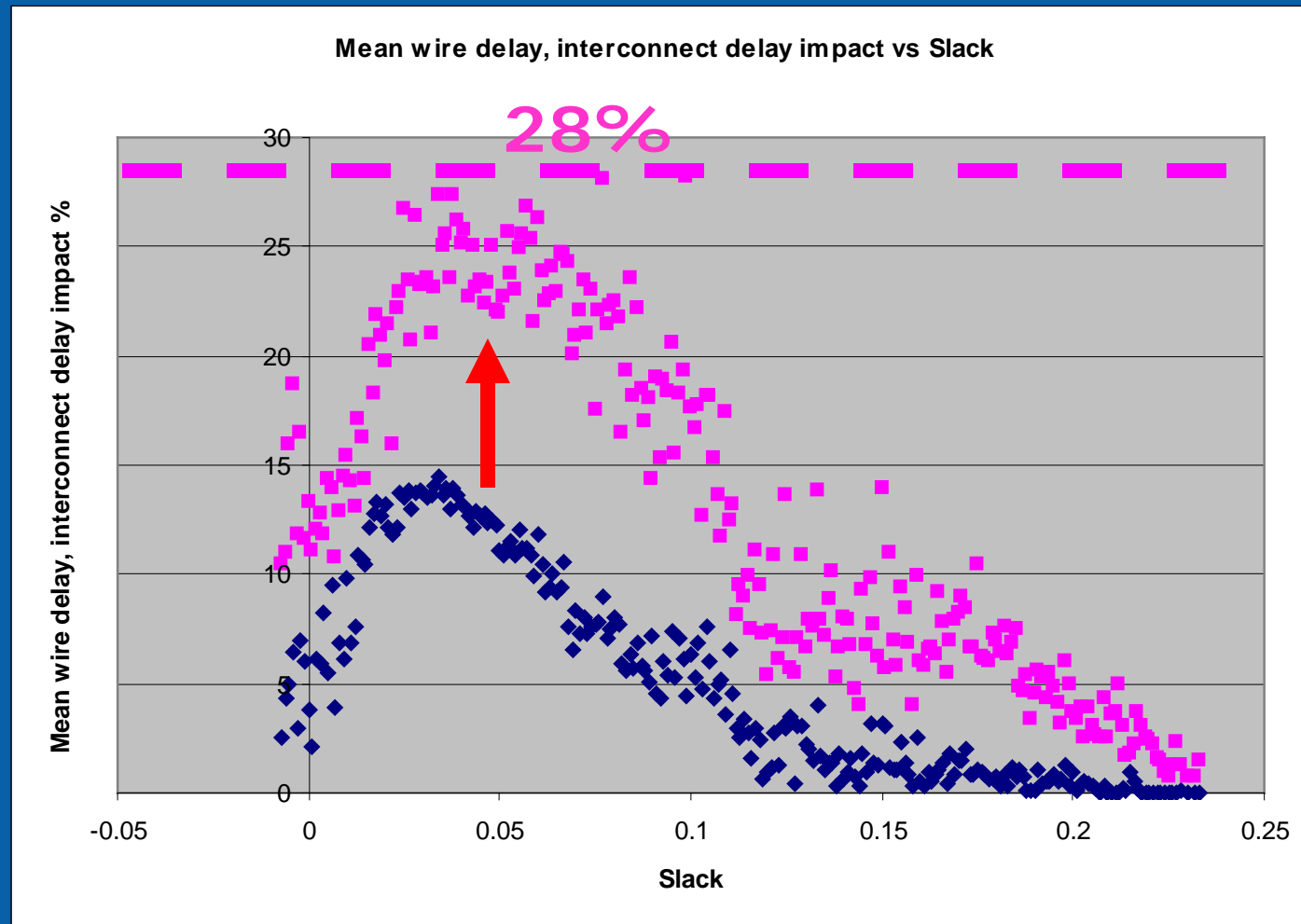
A Closer Look at One Block: Wire Delay



- Wire delay increases as slack decreases
- Timing wall due to sizing/II-insertion because of emphasis on power also
- Interconnect delay impact won't change without power optimization

Mean wire delay vs slack for worst internal paths between unique pair of sequentials in a ~40 K cell block with ~4 K sequentials

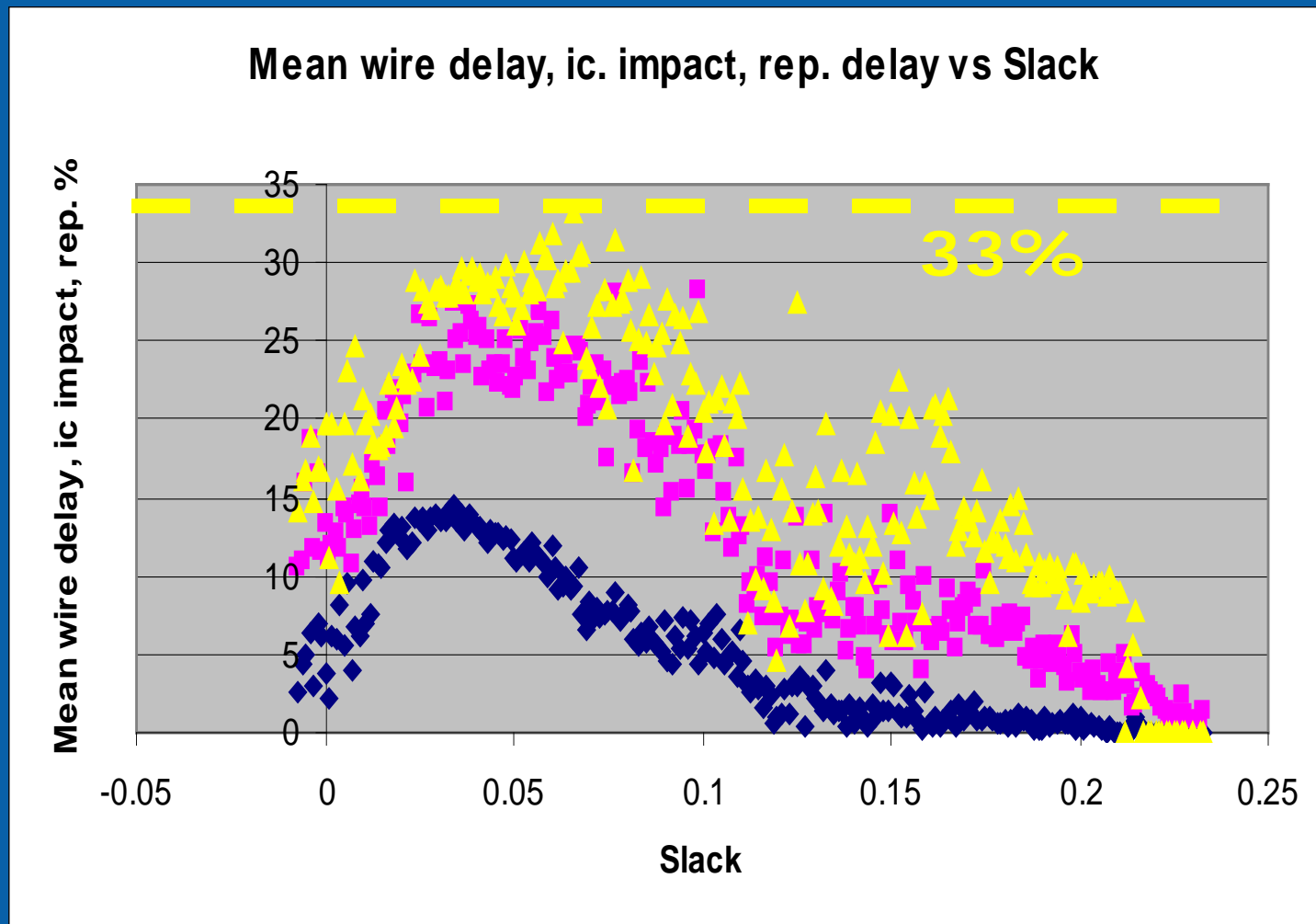
A Closer Look: Slope-/Cell-delay Degradation



- Slope-/cell-delay degradation contribute as much as wire delay
- Secondary effect not second order

Mean wire delay & impact vs slack for worst internal paths between unique pair of sequentials

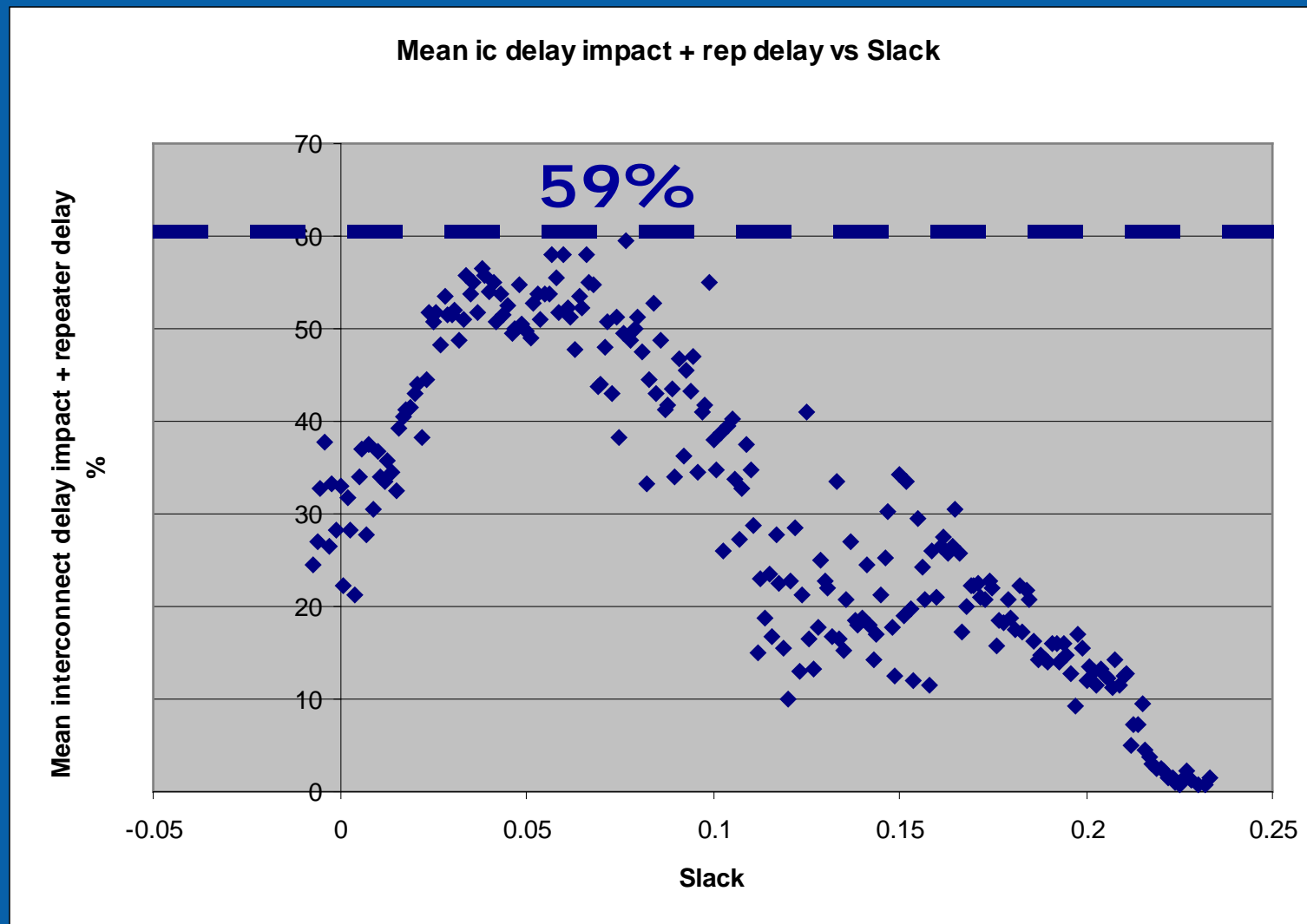
A Closer Look: Repeater Delay



Mean wire delay, interconnect impact, repeater delay vs slack for worst internal paths

- Repeater = inverter or buffer
- On critical path, most inverters/buffers are repeaters
 - Cell library is granular
- Repeater delay same as interconnect delay impact

A Closer Look: Adding all 3



- Average overall impact: 30%
- Similar behavior for smaller block sizes
 - Same quality: repeaters are indicators of synthesis quality
- One has hoped for better!

Overall interconnect delay impact, including repeater delay vs slack for worst internal paths

Implications

- [Bohr 95] “Interconnect Scaling – The Real Limiter to High Performance ULSI”
- Would have been true*, had it not been for the emphasis on power
- Pushing speed
 - Microprocessors? Cores already run at 3.2 GHz
 - Processors in netbooks/smartphones
 - Graphics processors
- Technology scaling:
 - Transistors improve; $R/\mu\text{m}$ increases; $C/\mu\text{m}$ stays the same
 - RC stays the same, assuming ideal length scaling
 - Interconnect component likely continue to increase

Possible Solutions

- From technology side:
 - 3 D?
 - Al → Cu → ? Low k?
 - Not in sight for next few years?
- From CAD
 - Placement, routing, physical synthesis running out of steam: “don’t know what the opportunities are”
 - Logic synthesis/tech. mapping doesn’t help, where it is used: serves the purpose of creating a netlist from RTL
 - “Death of Logic Synthesis” – ISPD’05?
- How about logic synthesis after global routing

Logic Synthesis After Global Routing

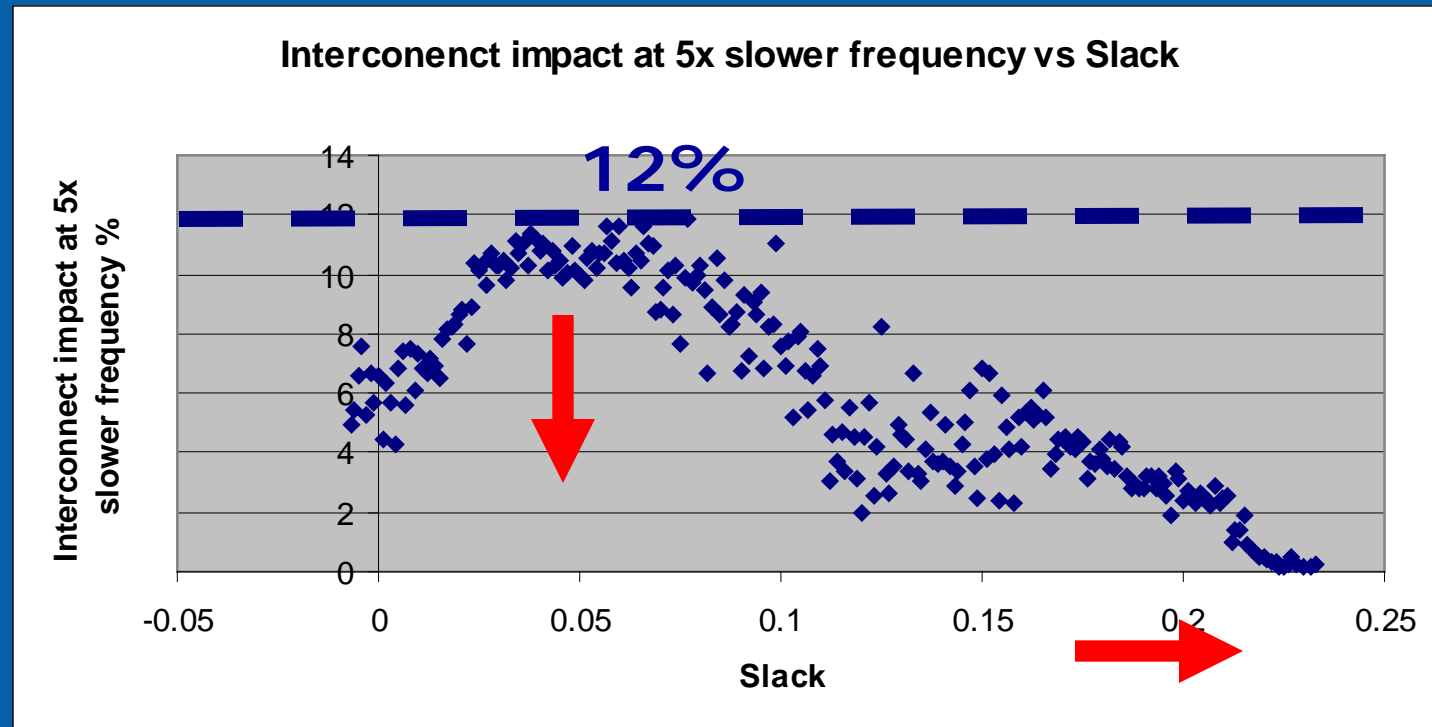
- Why?

- Routing picture known after placement/CTS/global route
- Only then we know the real impact of interconnects on delay
 - Dependence on topology, layers, vias, repeaters, detours, congestion
- Logic synthesis/technology mapping are powerful transformations, but...

- Challenges:

- Using placement/routing information
- Requires more memory/computation: faster/multi-core CPUs with more memory
- Polynomial time algorithms performing simultaneous optimizations
 - An example: simultaneous mapping/placement

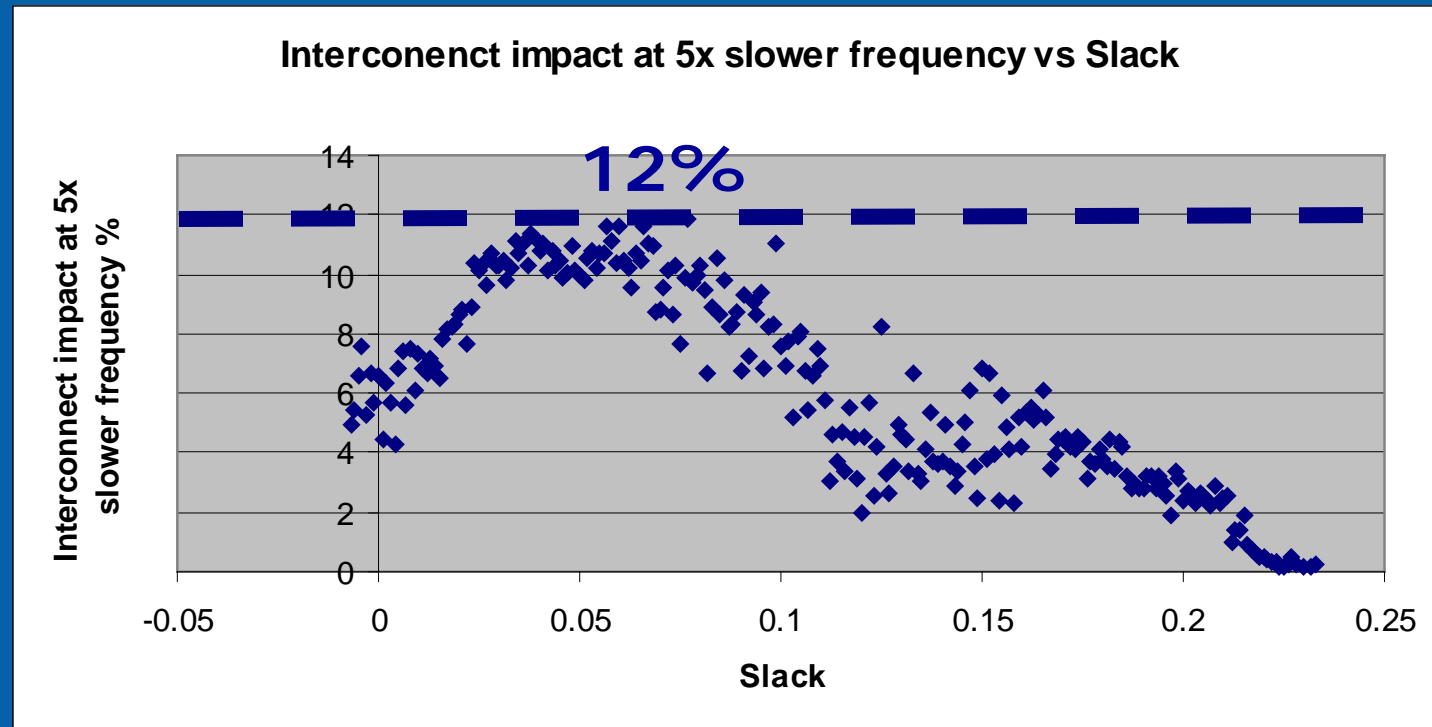
Low Frequency (high 100s of MHz)/Low Power Designs



Projected* interconnect delay impact for 5x slower design (could be much lower)

- Processor running at 5X slower frequency consumes 5x lower dynamic power
 - Interconnect delay impact as percentage of cycle time reduces by same factor
- Additional quadratic power savings due to supply voltage reduction
 - Slower gates, but interconnect component stays roughly the same
 - Overall interconnect impact on delay goes down further
 - Doesn't require as many repeaters
 - Critical paths gate-delay dominated

Low Frequency (high 100s of MHz)/Low Power Designs



Projected* interconnect delay impact for 5x slower design (could be much lower)

- Effect of re-pipelining on delay
 - Less sequentials → Less clock buffers/nets → More routing resources for signals → Better routing → Lower interconnect impact
- Problems for low power/high speed not the same!
- 1 Million cell placement for 600 MHz != 200 K cell placement for 3 GHz
- What if we want to run a processor in both the modes

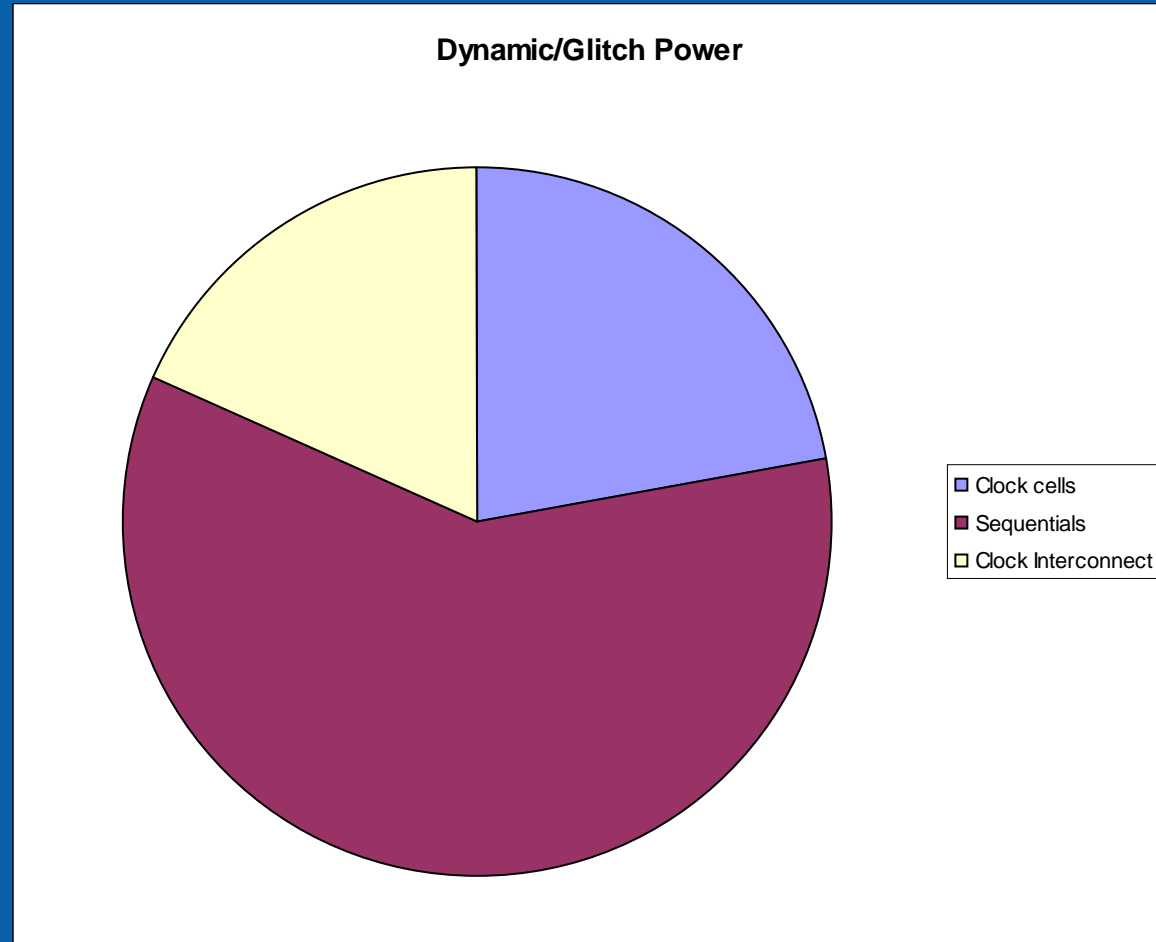
Agenda

- Introduction
- Impact on Timing
- Impact on Power
- Conclusions

Power Dissipation in RLS/SDP blocks

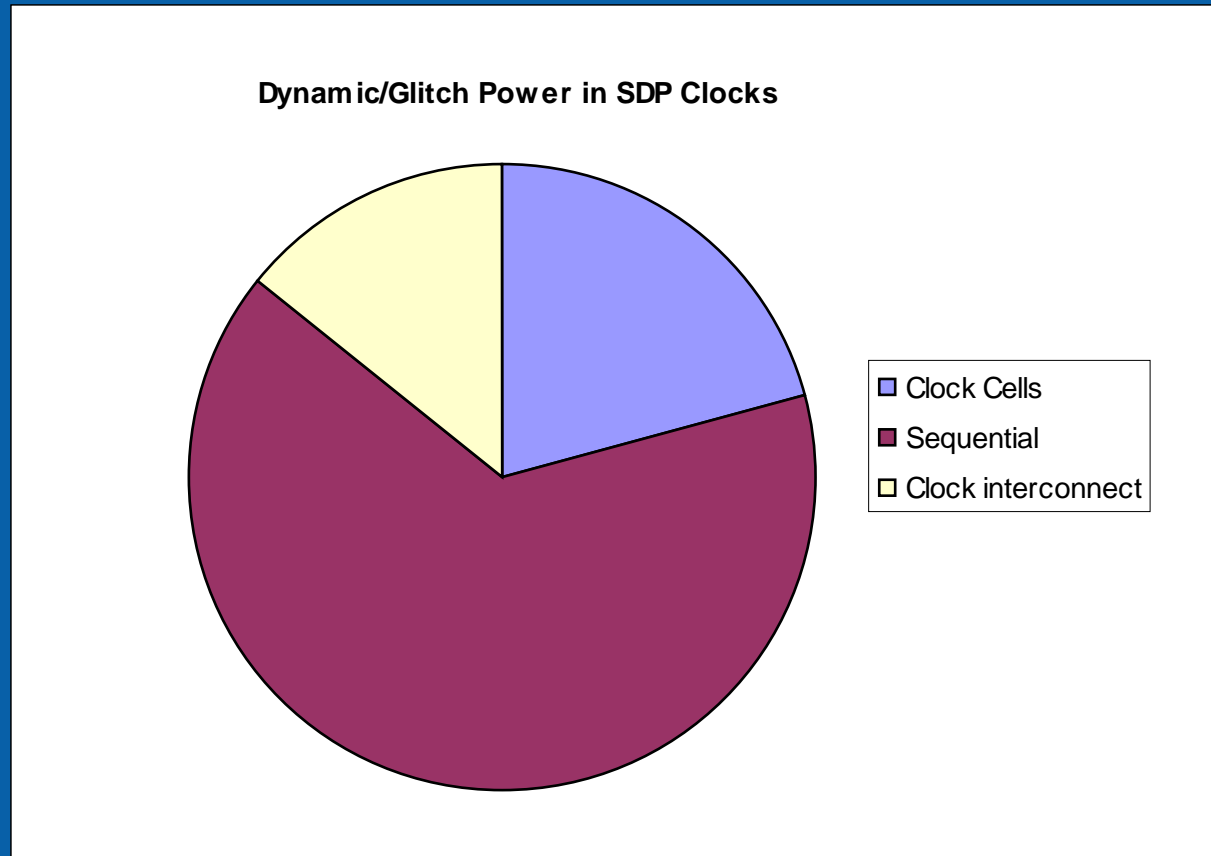
- Typical power dissipation distribution in high speed microprocessors: 60% dynamic; 10% short circuit; 30% leakage
 - High-k metal gate transistors with strain, high percentage of low-leakage/high-vt devices along with power gates has largely contained the leakage
 - High use of clock gating reduces the dynamic power in combinational logic
- RLS and SDP blocks contribute to more than 70% of the total power in the core
- RLS contributes to nearly 1/3rd and SDP 2/3rd

Clock Interconnect Power in RLS blocks



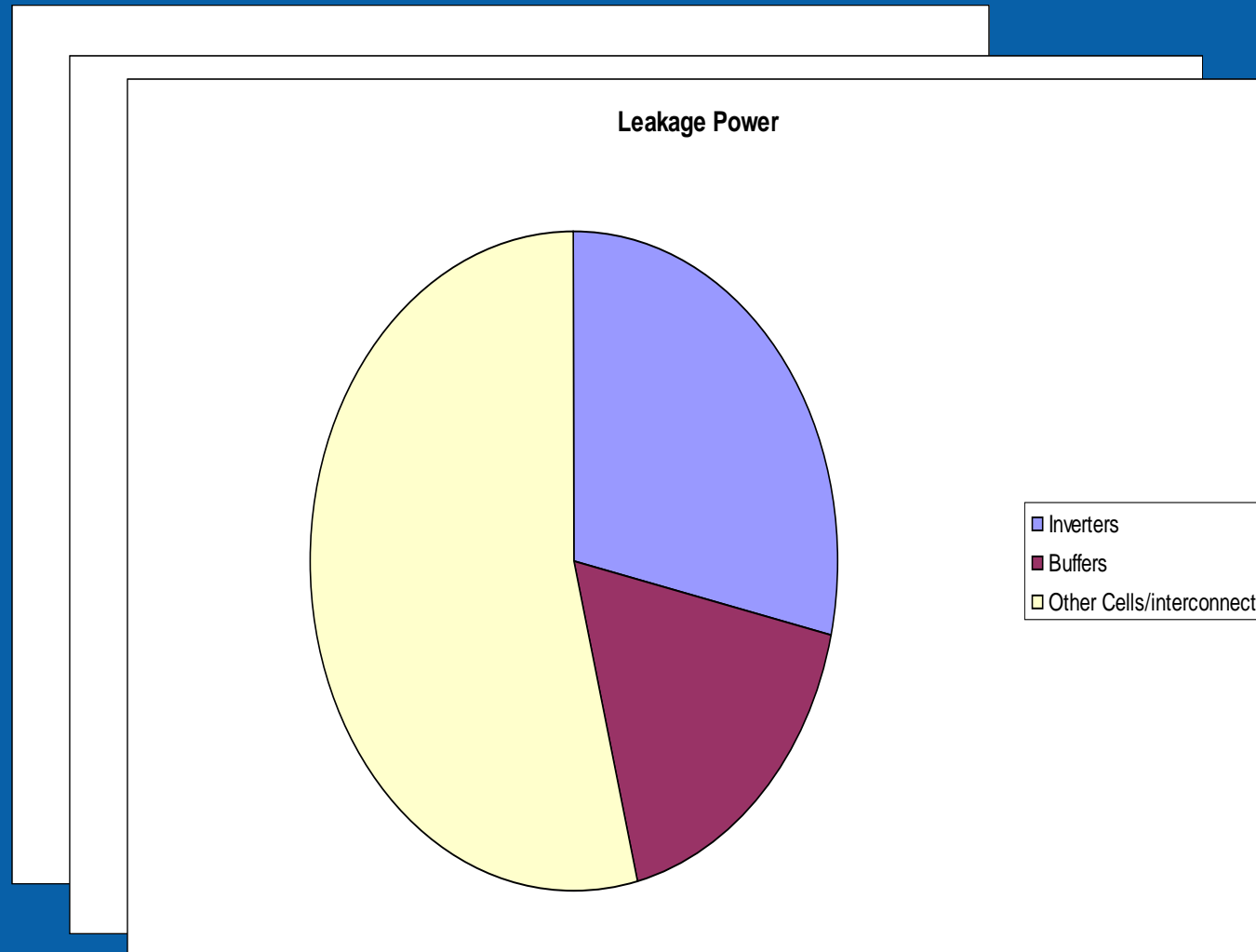
- Interconnects contribute to 18% of dynamic/glitch power in clocks
- Clock tree (including sequentials) contribute to 71% of dynamic power
 - # of sequentials contribute roughly to 1/5th of cell count in RLS
- Out of total dynamic/Glitch power in RLS blocks
 - Clock cells contribute 16%
 - Clock Interconencts contribute 13%
 - Sequentials contribute 42% of dynamic power in RLS

Clock Interconnect Power in SDP blocks



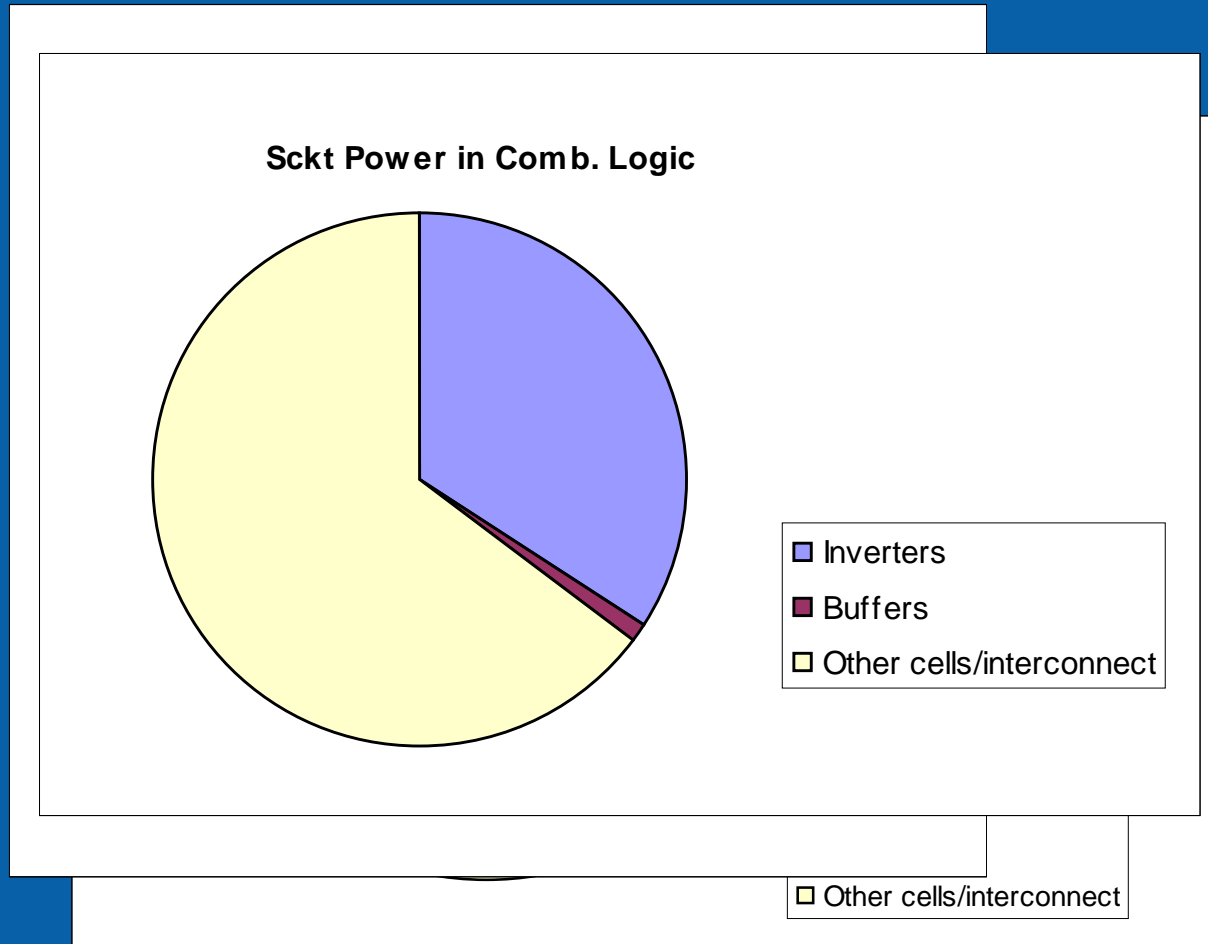
- Interconnects contribute to 14% of dynamic/glitch power in clocks
 - 4% less than RLS, because of (i) choices of upper metal layers with spacing, (ii) more regular placement than RLS, and since (iii) DEs may have duplicated buffers more than necessary
- Clock tree (including sequentials) contribute to 36% of dynamic power
 - Nearly half of the corresponding number in RLS
 - Highly active combinational logic
- Out of total dynamic/Glitch power in SDP blocks
 - Clock cells contribute 7%
 - Clock Interconnects contribute 5%
 - Sequentials contribute 23% of dynamic power in RLS
- 35% of total dynamic/glitch power in SDP local clocks as compared to 71% in RLS
 - Less number of sequentials: roughly 1/8th of SDP cell count as compared to 1/5th in RLS

Repeater Power in RLS blocks



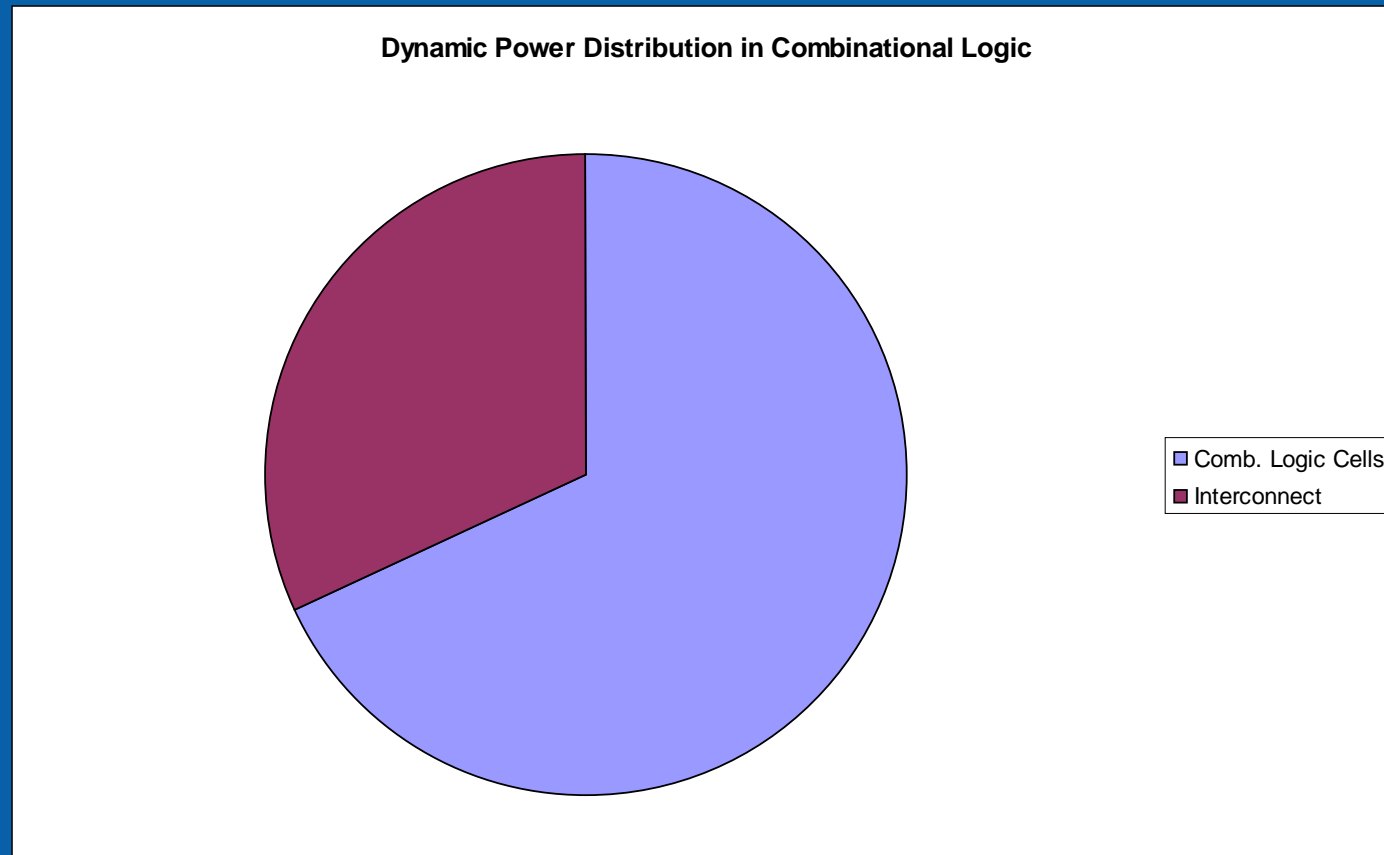
- Dynamic power in combinational logic: 27% of dynamic power in RLS
 - Inv./buf. contribute 30% to that; somewhat low, given 44% of cell count, since activity factors for combinational logic are lower than those in clock tree
- SC power in combinational logic: 50% of SC power in RLS
 - Inv./buf. contribute 65% to that; high since no transistors for stacking
- Lkg power in combinational logic: 71% of leakage in RLS
 - Inv./buf. contribute to 46% to that; can be explained by 44% repeater count

Repeater Power in SDP blocks



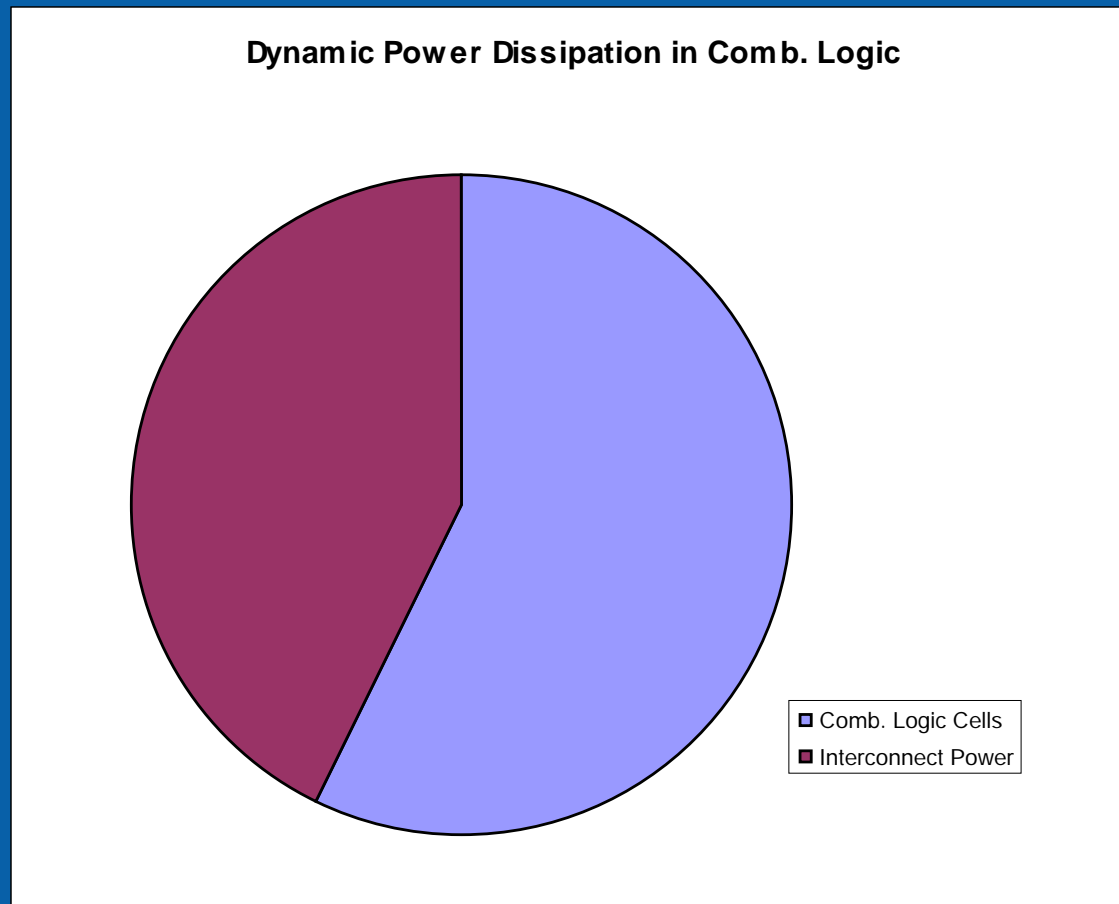
- Dynamic power in combinational logic: 63% of dynamic power in SDP
 - Inv./buf. contribute 20% to that; 32% repeater count, as compared to 44% in RLS
- SC power in combinational logic: 50% of SC power in SDP
 - Inv./buf. contribute 35% to that; lower as compared to RLS, since repeater count is less
- Lkg power in combinational logic: 80% of total leakage in SDP
 - Inv./buf. contribute to 39% to that; can be explained by 32% repeater count

Interconnect Power in Combinational Logic in RLS blocks



- 32% of dynamic/glitch power in combinational logic; 8% of dynamic/glitch power in RLS

Interconnect power in Combinational Logic in SDP blocks



- 47% out of dynamic power in combinational logic; 30% of total dynamic/glitch power in SDP
- 15% higher than corresponding RLS number
- Could be result of better logic distribution (less repeaters), *i.e.*, power in interconnect and combinational logic is balanced, unlike in RLS

Agenda

- Introduction
- Impact on Timing
- Impact on Power
- Conclusions

Conclusions: Impact on Timing

	RLS Avg.	SDP Avg.
Wire-delay %	6	5
Wire-delay + slope-/cell-delay degradation %	13	9
Repeater-delay %	19	11

- Avg. IC delay impact + repeater delay for RLS/SDP; 33%/20% of cycle time
- SDP design (manual) although less dense than RLS (implying as long wires or as sparse wire-density), on an average, still has less interconnect impact on timing
- In case of RLS, interconnect delay impact on timing is more than 30%, on an average, pointing to the limited success of physical design/synthesis research

Conclusions: Interconnect Impact on Repeaters

- Repeater count as a percentage of cell count:
 - RLS: 27% inverters, 17% buffers; total 44%
 - SDP: 22% inverters, 9% buffers; total 32%
- Impact of repeaters on power is not much, because of clock gating and low leakage due to better transistors
- SDP blocks have 12/13% less repeaters than RLS: careful manual design can avoid repeaters
- Repeater percentage in RLS varies linearly with cell count; not so, in SDP
 - Artifact of algorithms/tools/flows in RLS...?
- According to repeater count metric, RLS tools/flows could improve 13%

Conclusions: Impact of Interconnect on Power

- Power in clock interconnects:
 - RLS: clock wires contribute to 18% of dynamic/glitch power in clock tree and 13% of total RLS dynamic/glitch power
 - SDP: clock wires contribute to 7% of dynamic/glitch power in clock tree and 5% of total SDP dynamic/glitch power
- Power in combinational interconnects:
 - RLS: combinational wires contribute 32% of dynamic/glitch power in combinational logic and 8% of total RLS dynamic/glitch power
 - SDP: combinational wires contribute 47% of dynamic/glitch power in combinational logic and 30% of total SDP dynamic/glitch power
- Power in repeaters:
 - RLS: 30% to dynamic/glitch power in comb. logic and 8% to total RLS dynamic/glitch power; 65% to SC in RLS comb. logic and 32% to total RLS SC; 46% to lkg. in RLS comb. logic and 32% to total lkg. in RLS
 - SDP: 20% to dynamic/glitch power in combinational logic and 13% to total SDP dynamic/glitch power; 35% to SC in SDP comb. logic and 25% to total SDP SC; 39% to lkg. in comb. logic and 30% to total lkg. in SDP
- Interconnect power:
 - RLS: 21% of dynamic/glitch in RLS; 30% including repeater dynamic/glitch power
 - SDP: 35% of dynamic/glitch in SDP; 48% including repeater dynamic/glitch power



Acknowledgments

- Noel Menezes, Intel
- Xinning Wang, Intel
- Wei-kai Shih, Intel
- Andy Carle, Intel
- ... many from EMG/TMG, Intel

Q&A