Machine Learning Applications in Physical Design: Recent Results and Directions

Andrew B. Kahng
CSE and ECE Departments
UC San Diego

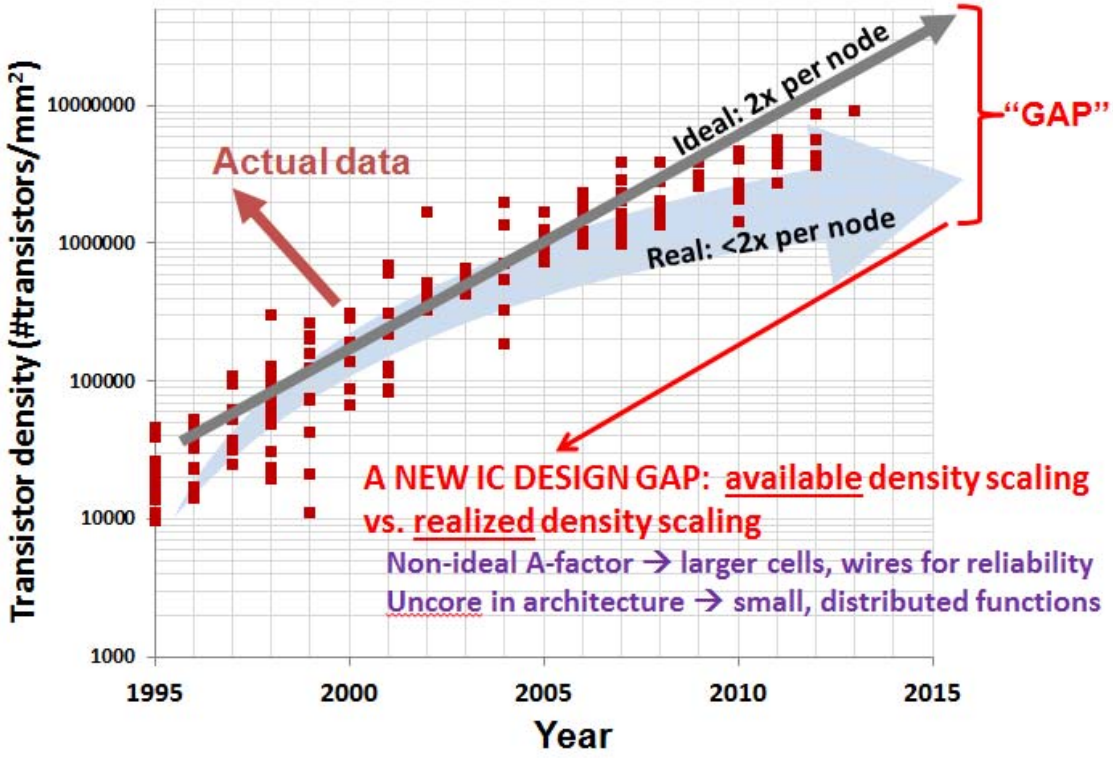
http://vlsicad.ucsd.edu

Agenda

- **Crises…**

IC Industry Crises: Cost, Quality of Design

- **Can't afford to design chips (tools, people, time, risk)**
- **Return on investment for new technology is poor**
  - **$$M to move to new node (28nm → 14nm →10nm → 7nm → …)**
  - **Benefit from new node: ~20% power, speed, area (less, today)**
- **Design Capability Gap**
  - **Available density grows at 2x/node**
  - **Realizable density grows at 1.6x/node**
  - **UCSD / 2013 ITRS**



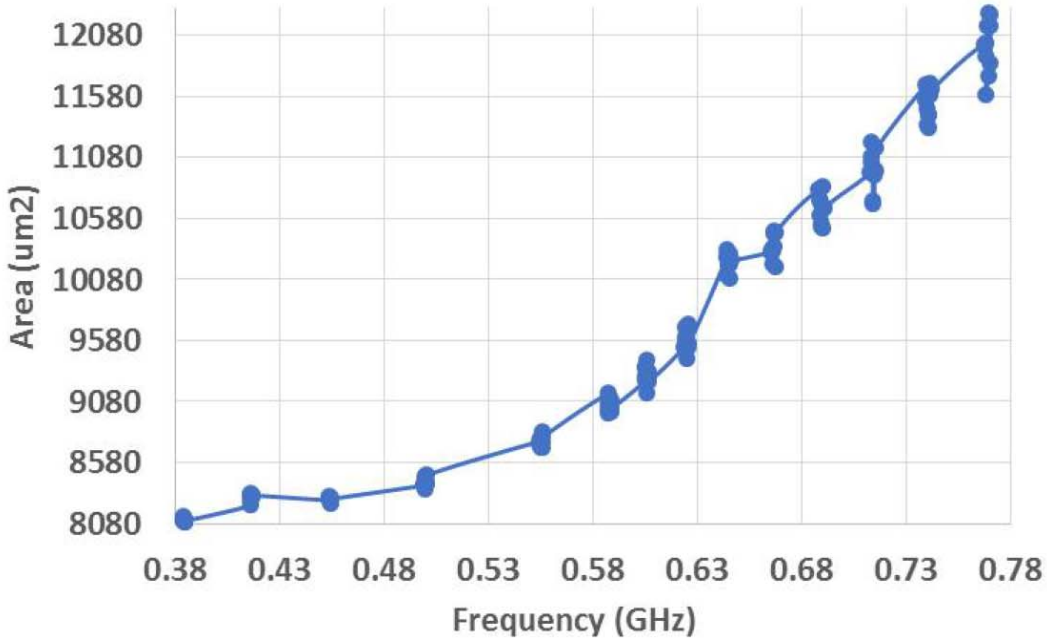
IC Design Crises: Unpredictability, Schedule

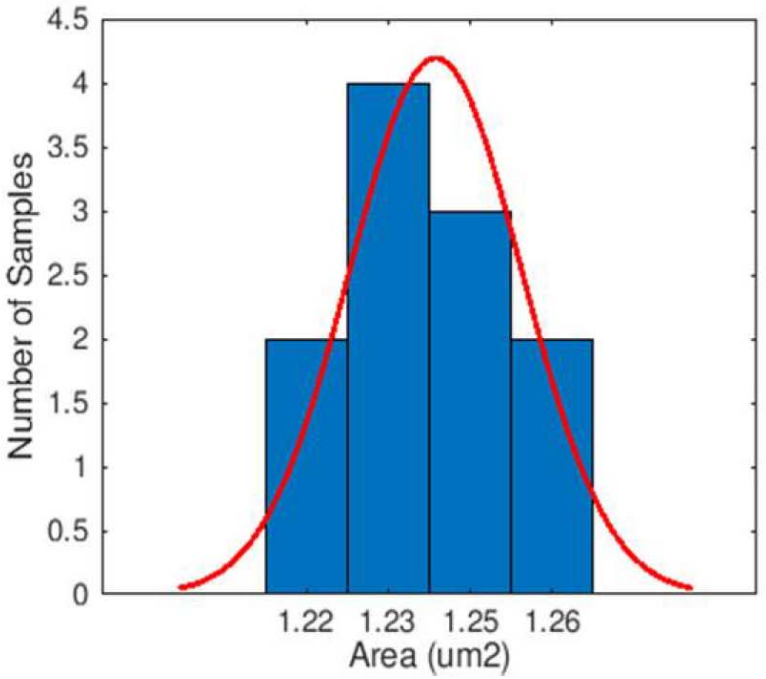
- **Many steps in long "design flow"** → **can we predict outcome?**
- **Many chicken-egg loops** → **convergence point? how to initialize?**
- **Nearly all problems are NP-hard**
  - Min-cut hypergraph bisection, Quadratic assignment, Multicommodity flow, Max-weight independent set, Multi-vehicle TSP, k-colorability, …
- **Huge "n"→ metaheuristics piled on metaheuristics**
- **Suboptimality is expensive**
  - 10% of {power, speed, area} is half of benefit from new node
- **Iteration is expensive**
  - Moore's Law: 1 week = 1 percent
- **Conservatism ("margin") is expensive**
  - But: "oops" (didn't fit, didn't route, too slow) is unacceptable

Unpredictability of Design

- Intractable optimizations → heuristics piled on heuristics
- "Noise" or "Chaos" when EDA tools "try hard"
- Unpredictability → added margin and schedule

  14nm PULPino: ∆area = 6% from ∆freq = 10MHz !





Challenges: Schedule, Quality, Cost

"The Last Semiconductor Scaling Levers"

- **Quality**
  - Improved design tools and methods
  - Reduced margins
- **Schedule**
  - 1 week = 1%
- **Cost**
  - IC design is expensive (engineers, tools, spins, …)

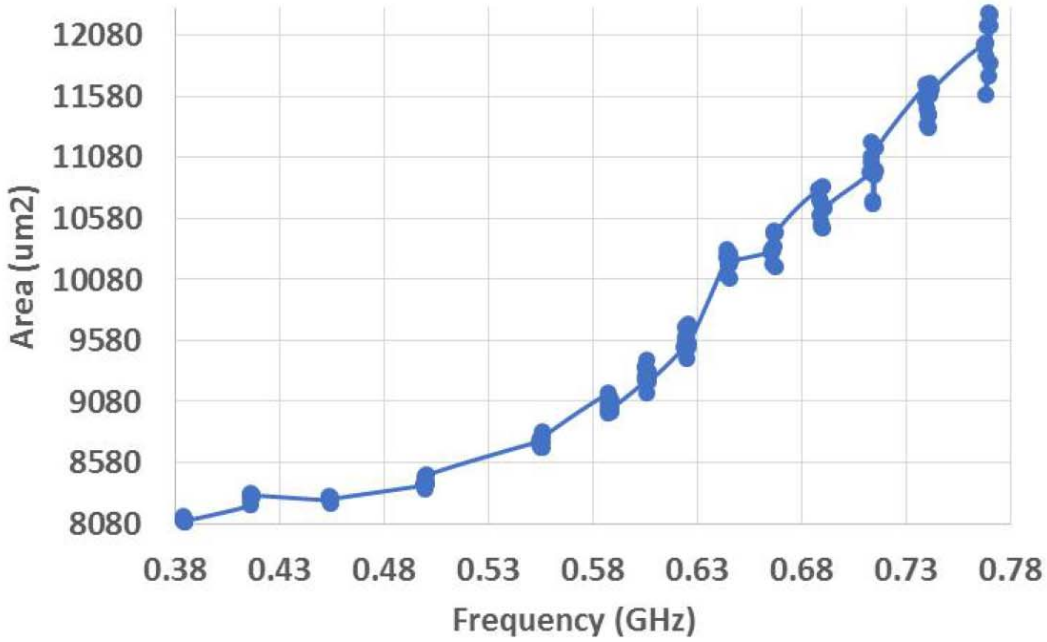
Agenda

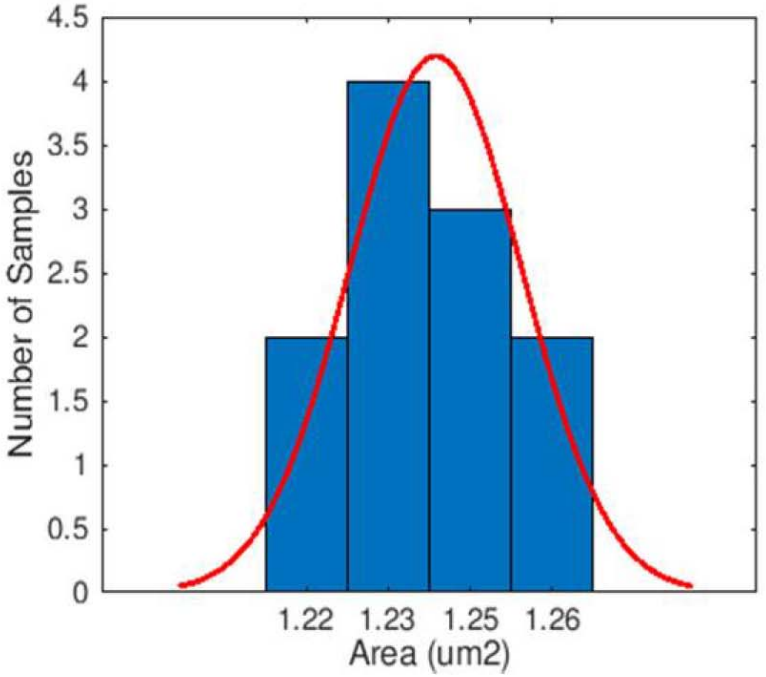
- **Crises…**
- **… and a Vision**

Unpredictability of Design

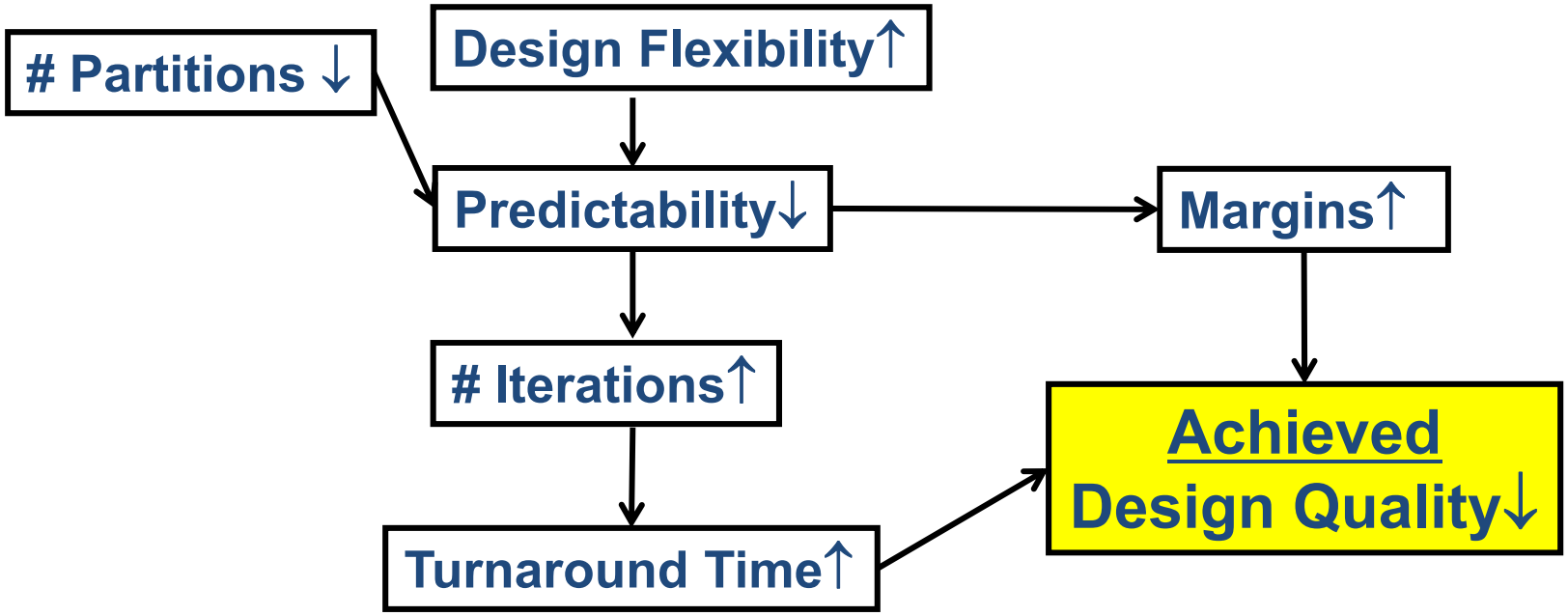
- Intractable optimizations → heuristics piled on heuristics
- "Noise" or "Chaos" when EDA tools "try hard"
- Unpredictability → added margin and schedule

  14nm PULPino: Δarea = 6% from Δfreq = 10MHz !

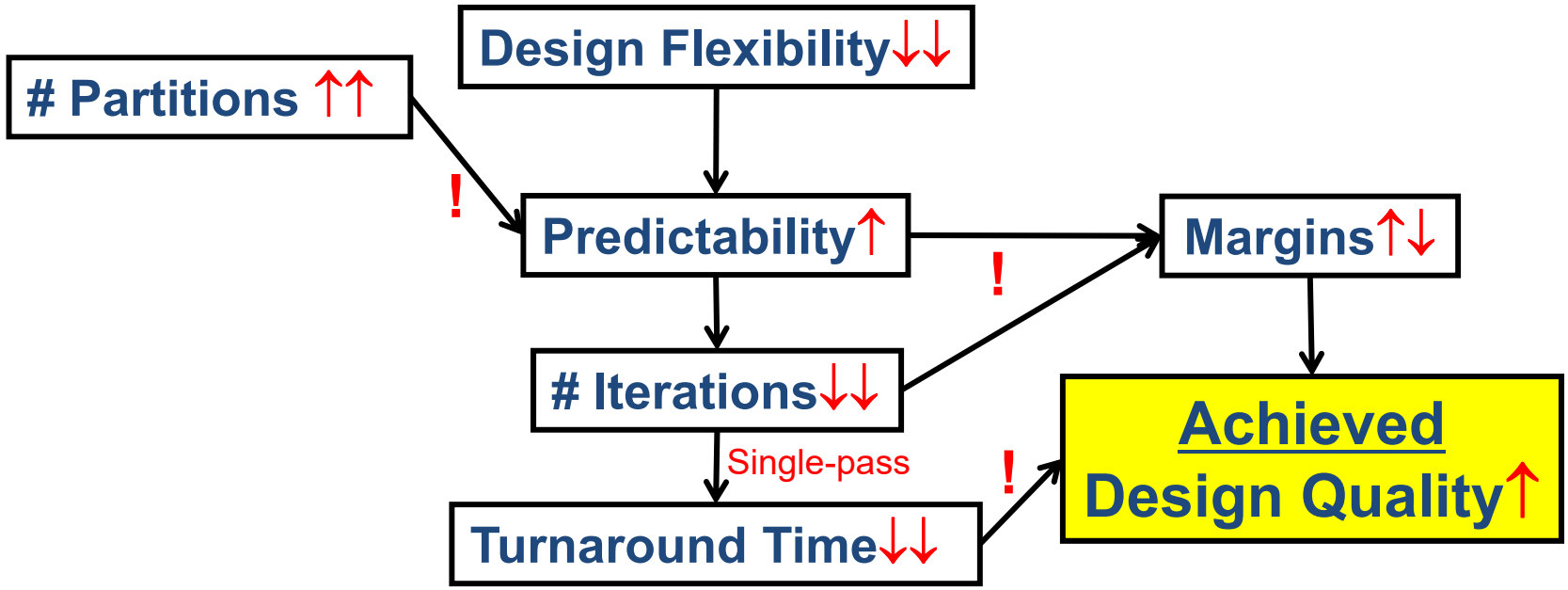




Today's SOC Design



Vision for Future SOC Design



Mindsets

- **Tools should not return unexpected results**
- **Achieve predictability from the user's POV**
- Use cloud/parallel to recover solution quality
- Focus on reducing design time, design effort

✓ **Quality**
✓ **Schedule**
✓ **Cost**

Machine Learning will be a key piece of this …

Agenda

- Crises…
- … and a Vision
- Machine Learning in PD

Machine Learning in Physical Design

Problem types solved with Machine Learning

- Classification
- Regression
- Dimensionality reduction
- Structured prediction
- Anomaly detection

Past ML applications in EDA literature

- Yield modeling (anomaly detection, classification)
- Lithography hotspot detection (classification)
- Identification of datapath-regularity (classification)
- Noise and process-variation modeling (regression)
- Performance modeling for analog circuits (regression)
- Design- and implementation-space exploration (regression)

ML in PD: modeling, prediction, correlation, …

Near-Term Opportunities

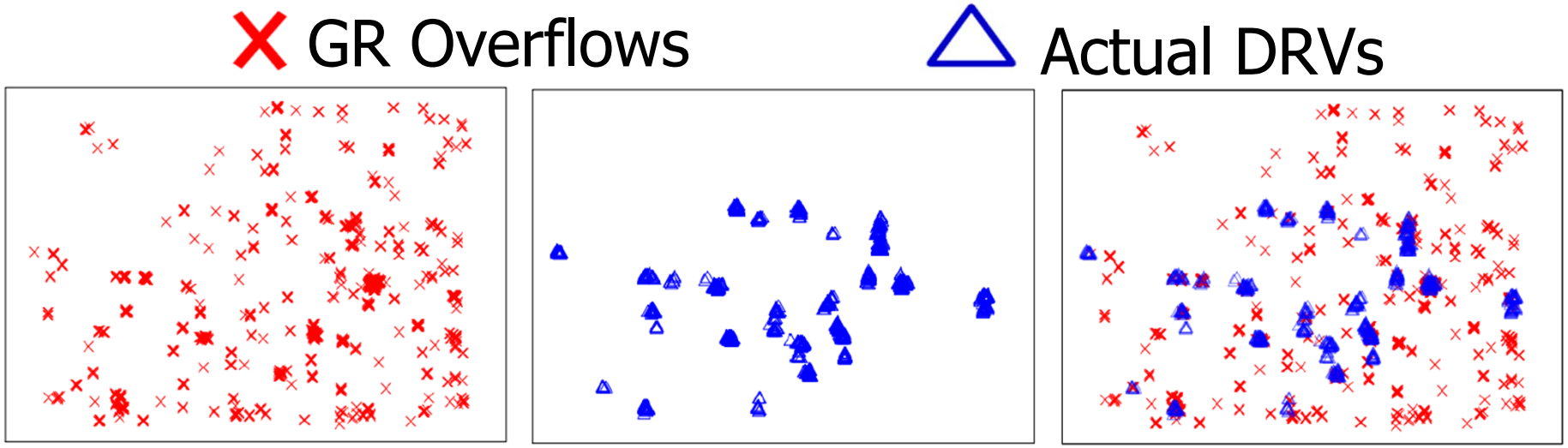
- **Modeling and Prediction**
  - Predict tool outcome = F(design, constraints, tool config)
    - How to run tool "optimally" for given design and design goals?
    - Avoid "failed runs" → reduce iterations in design flow
    - Dream: one-pass design flow
- **Analysis Correlation**
  - Model analysis errors (crude vs. golden analyses)
    - Reduced guardbands and pessimism → better design quality
- **Optimization (ML models = objective functions!)**
  - ML models = objective functions for higher-level optimization
  - Better use of resources (tools, schedule, engineers) + better tools
  - Project-level prediction, adaptive scheduling
- **Later: "Taxonomy and Roadmap"**

Agenda

- Crises…
- … and a Vision
- Machine Learning in PD
- Modeling and Prediction

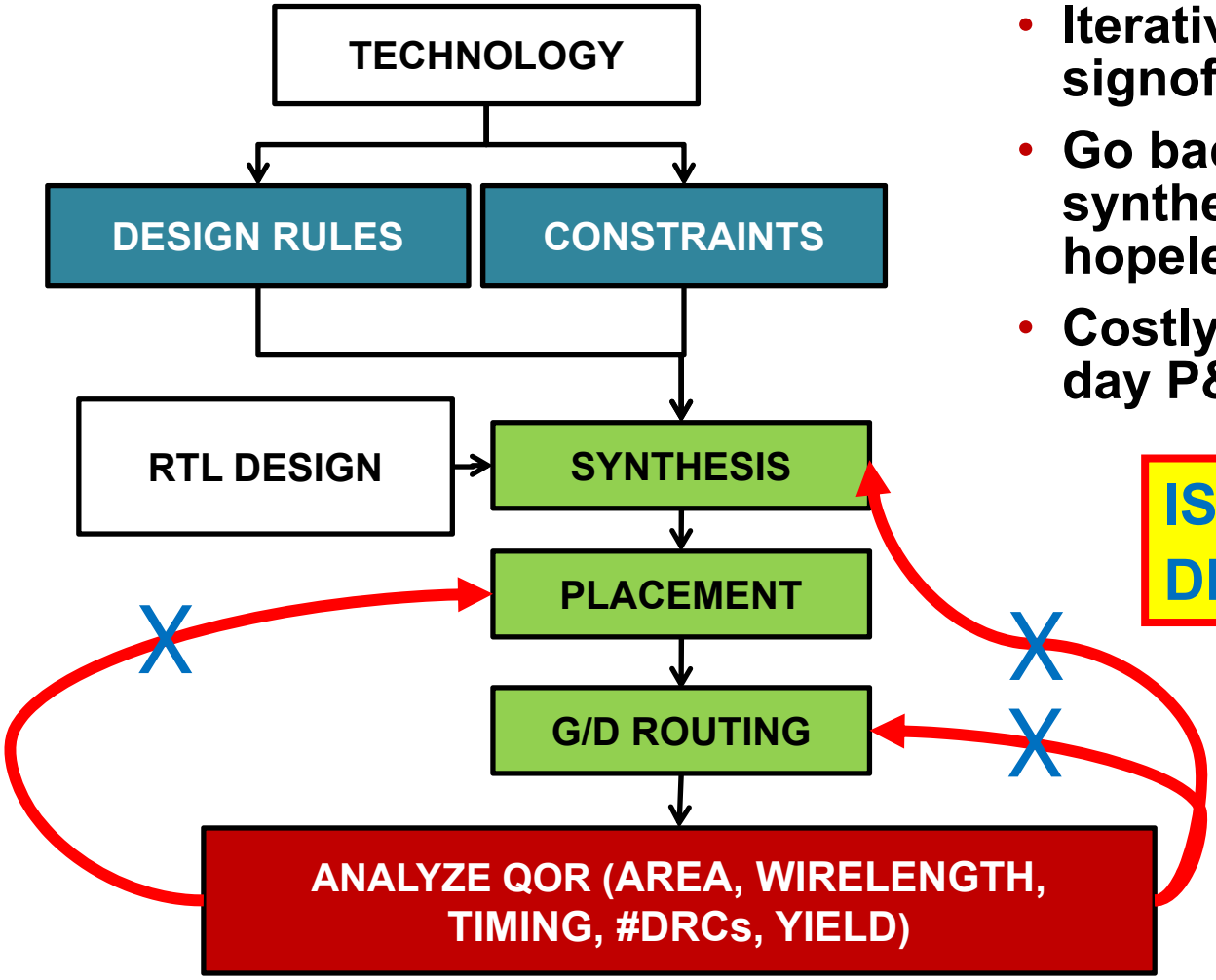
Example 1: Interface Between Global-Detailed Route

- 7nm P&R: global route (GR) congestion map does not correlate well with post-route (actual) DRC violations (DRVs)
- Many false-positive overflows in GR congestion map
- False positives do not correspond to actual DRVs



GR-based prediction can mislead routability optimizations!!!

Too Many Expensive Iterations



Conventional closure

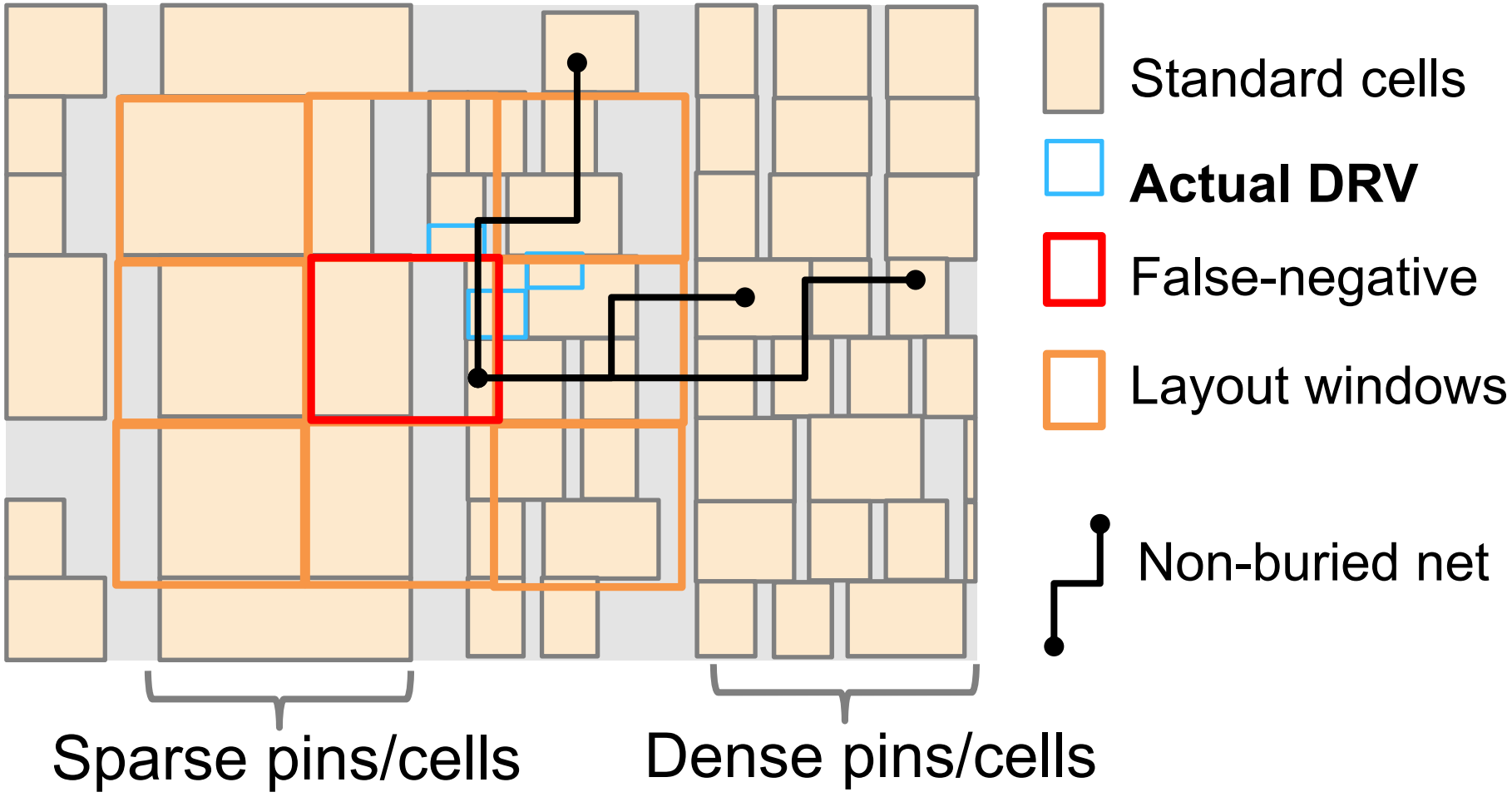
- Iteratively fix design before signoff
- Go back to placement or synthesis or FP if QOR is hopeless
- Costly iterations and TAT (7-day P&R runs…)

ISPD17: ML-based DRV predictor

Iteration with space padding,
NDR modifications,
density screens ...

Insight From Layout Studies

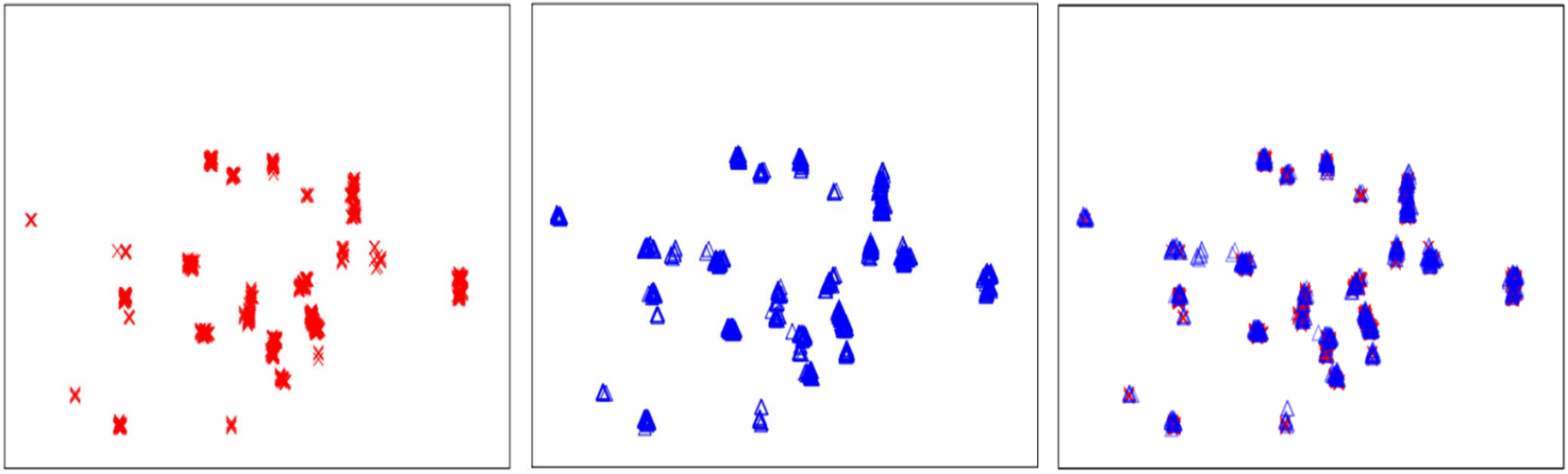
- Initial prediction from GR overflows and cell/pin density map
- Red DRV-hotspot likely a False Negative due to low cell-pin density
- Larger windows , buried nets (, NDRs, FFs, etc.) added to model inputs



Improved Learning-Based Predictor

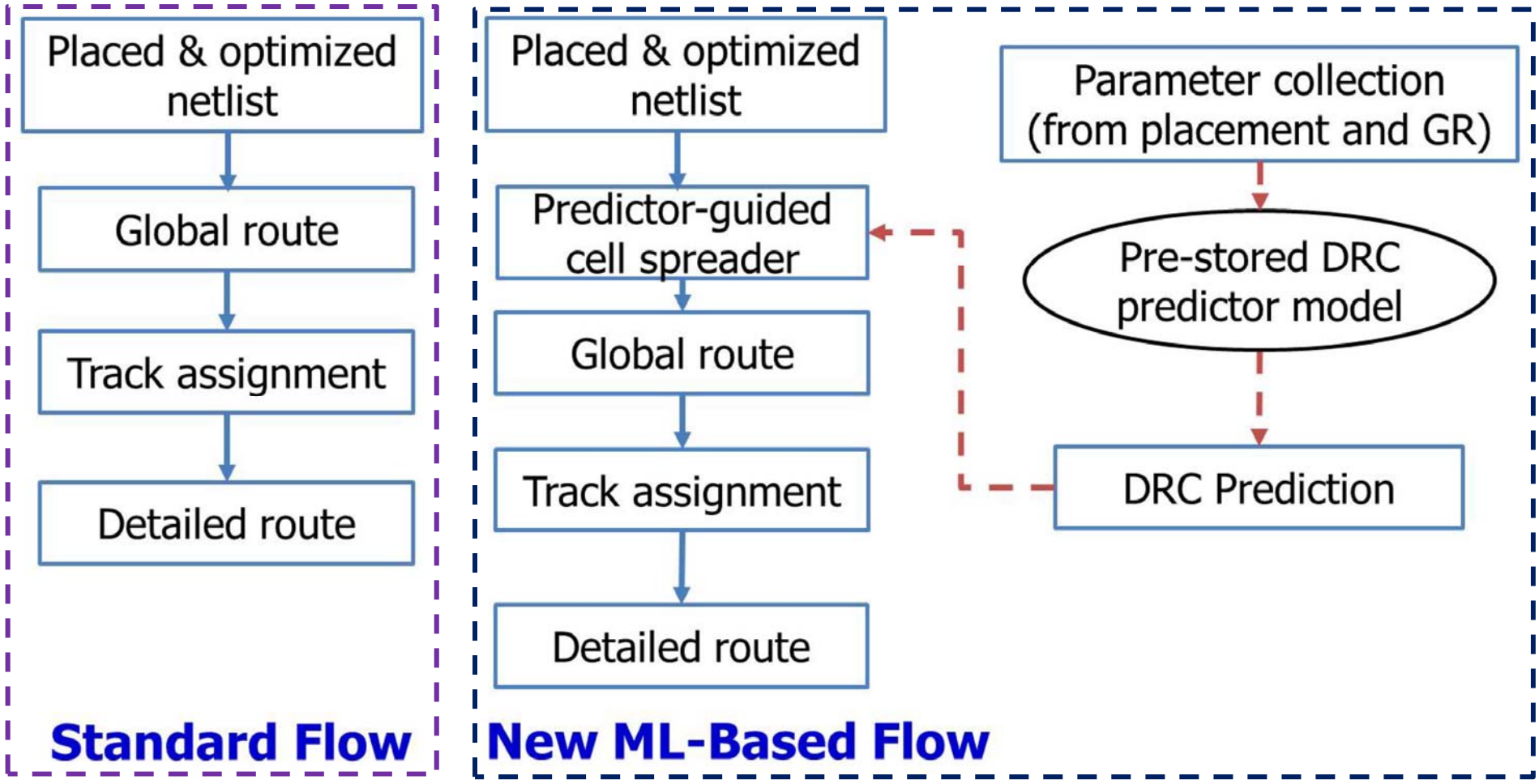
- Captures all true-positive clusters
- Maintains low false-positive rate

Learning-based Prediction **Actual DRVs**



(a) **(b)** **(c)**

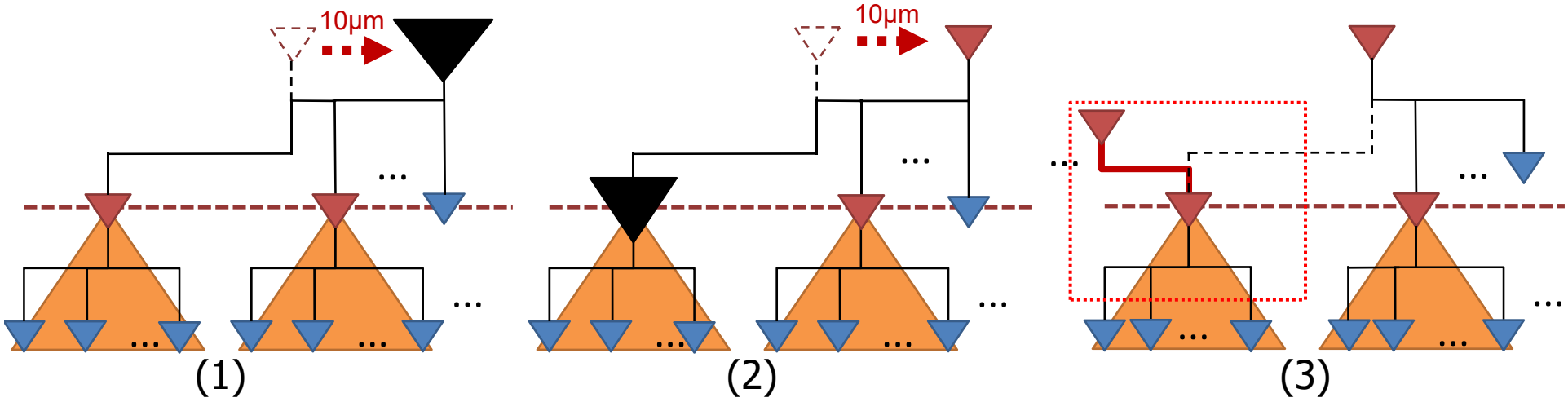
ISPD17: Model-Guided Routability Opt



- **New: True-Positive rate = 74%, False-Positive rate = 0.2%**
- Previous: True-Positive rate = 24%, False-Positive rate = 0.5%

Example 2: Local CTS Optimization Moves

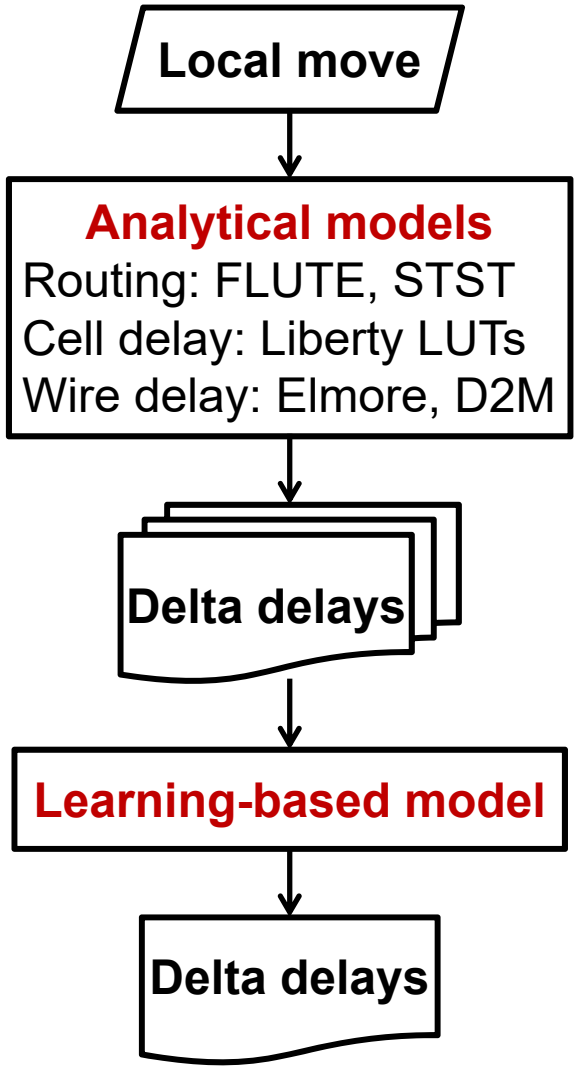
- Iterative local moves to minimize skew variation across corners
  1. Displacement {N, S, E, W, NE, NW, SE, SW} by 10μm x one-step sizing
  2. Displacement by 10μm x one-step sizing on child buffer
  3. Reassign to a new driver (i) at the same level, (ii) within bounding box of 50μm x 50μm

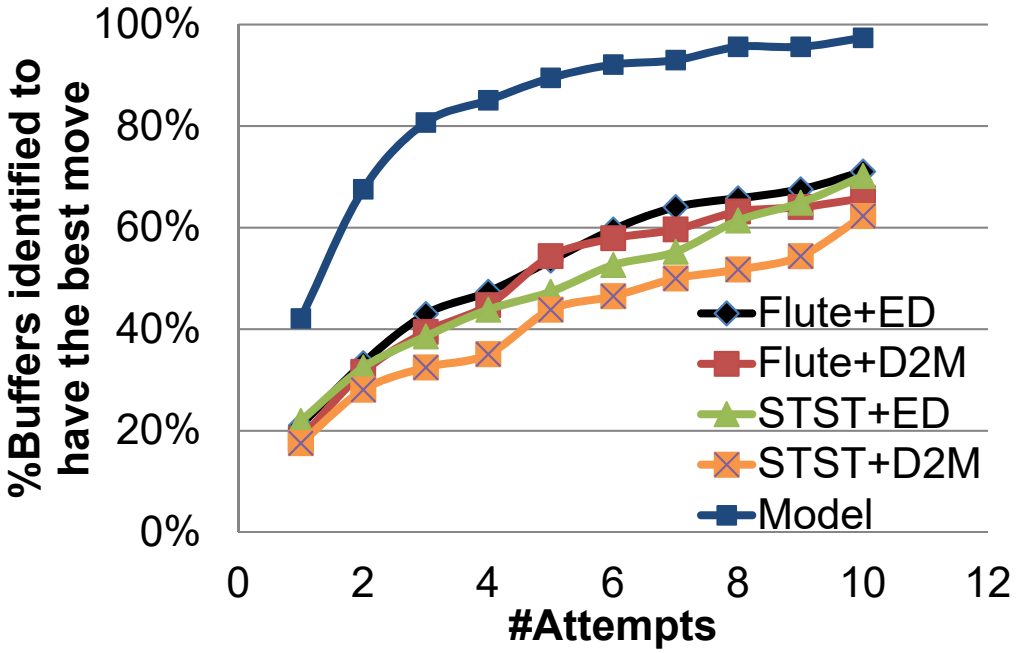


- Each move is expensive (legalization, ECO routing, RC extraction, STA)
- Each buffer has many candidate moves
- **DAC-15: learning-based model**

DAC15: CTS Outcome Prediction

- Predict driver-to-fanout latency change due to local moves

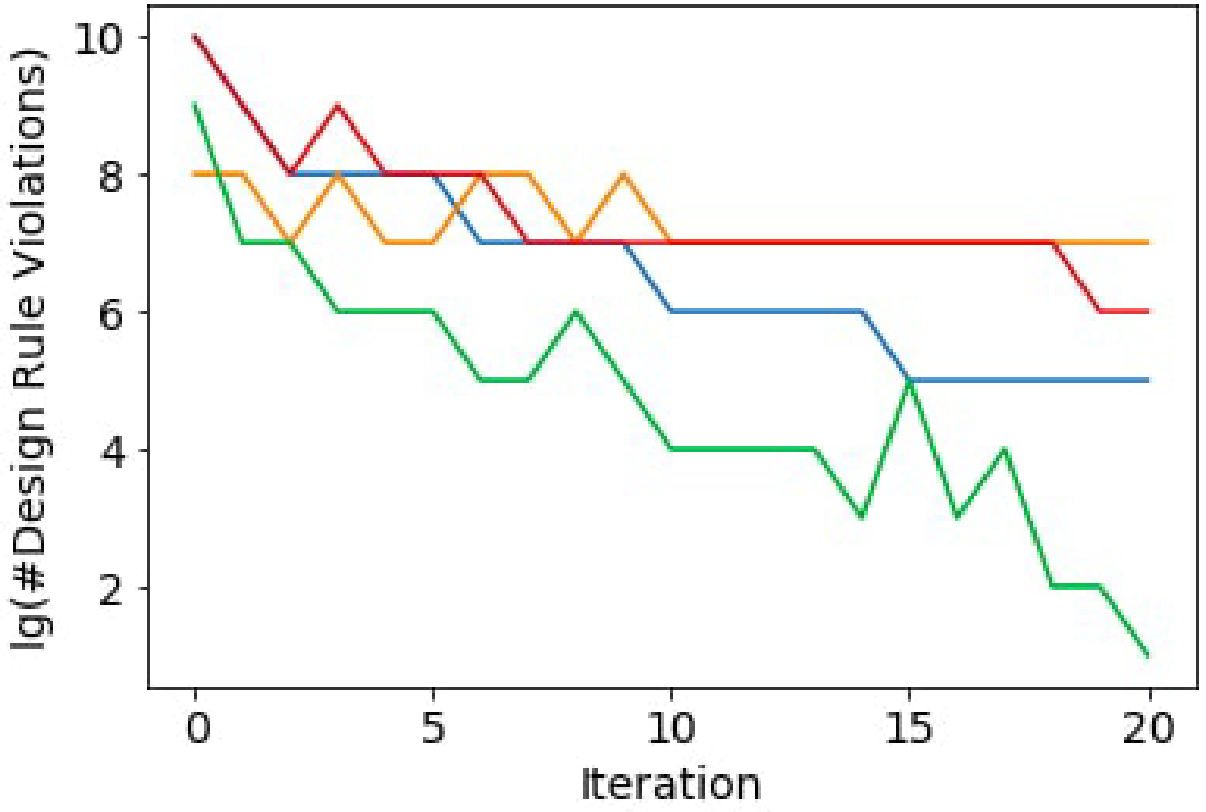




- Each attempt is a local move
- 114 buffers
- 45 candidate moves for each buffer
- Learning-based model identifies best moves for more buffers with less #attempts

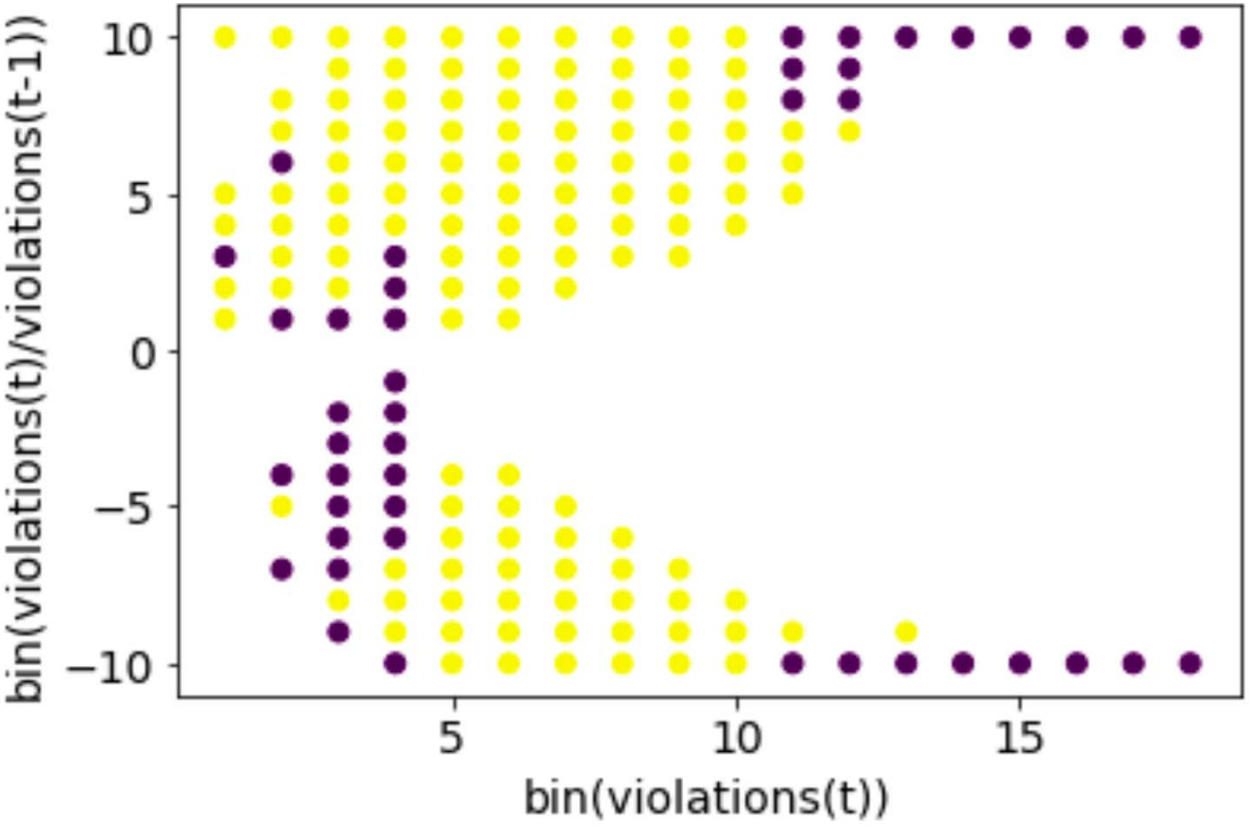
Example 3: Prediction of Doomed Runs?

- Some P&R runs end up with too many post-route DRVs
- Approach: track and project metrics as time series
- Markov decision process (MDP): terminate “doomed runs” early
- Shown: 4 example progressions of #DRVs (commercial router)
  - Stopping red, yellow runs early would save resources and schedule !

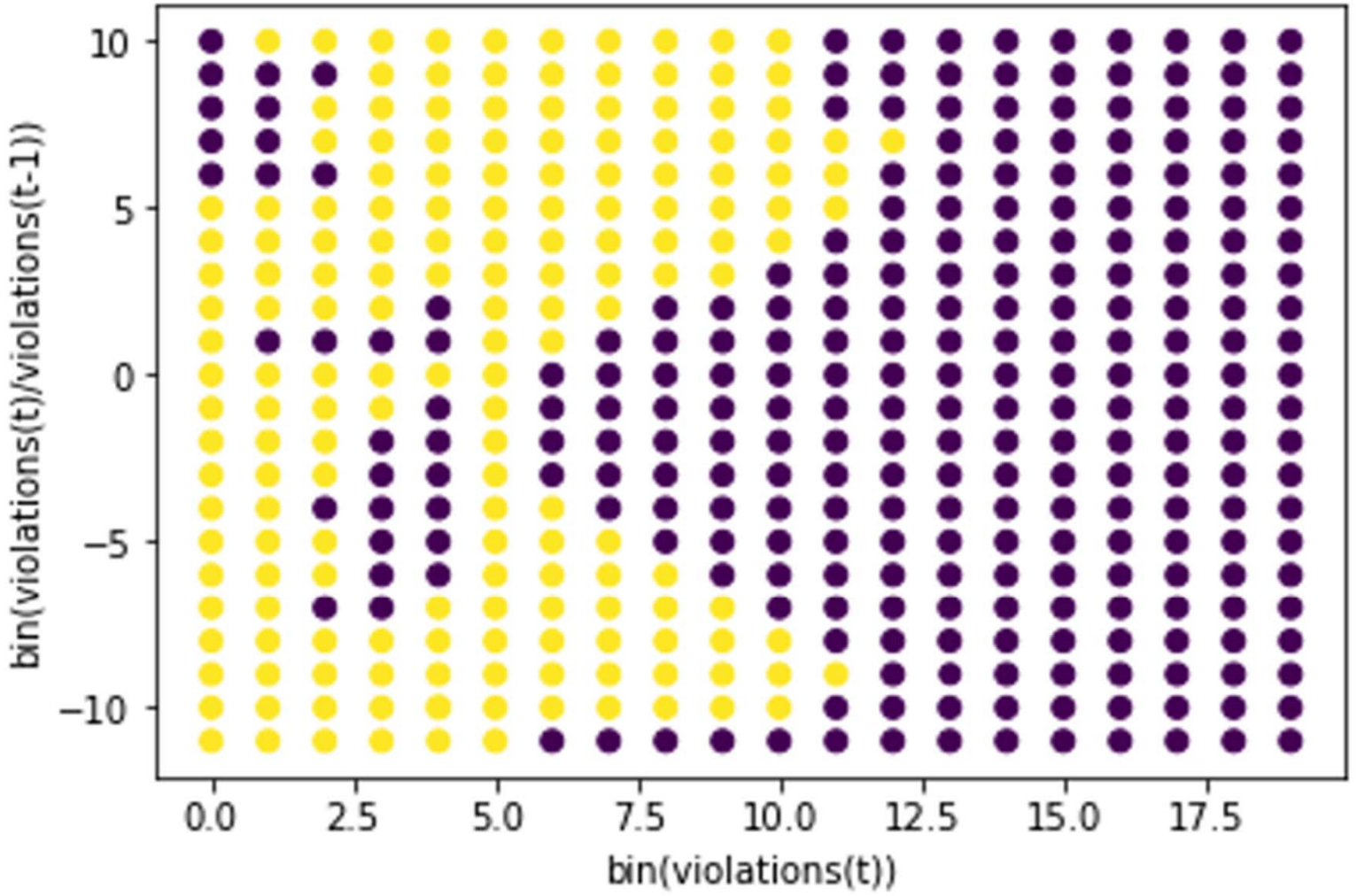


Markov Decision Process = "Strategy Card"

- **State space** from Fibonacci binning
- **Actions** – ***GO*** or ***STOP***
- **Rewards** at each state – e.g., small negative reward for *non-stop* state, large positive reward for *stop* with low #DRVs, etc.
- Automatically trained MDP "strategy card": **Yellow = GO, Purple = STOP**



Strategy Card “Completion”



Promising Initial Studies

- **TYPE 1 Prediction Error:** MDP STOPs a run that will eventually succeed
- **TYPE 2 Prediction Error:** MDP predicts GO at each iteration, but run fails
- **Training data: 1200 logfiles from PROBE experiments**
- **Testing data: 3442 logfiles from ARM Cortex M0 floorplan experiments**
- **Substantial #iterations saved for doomed runs (398 / 3442 cases)**

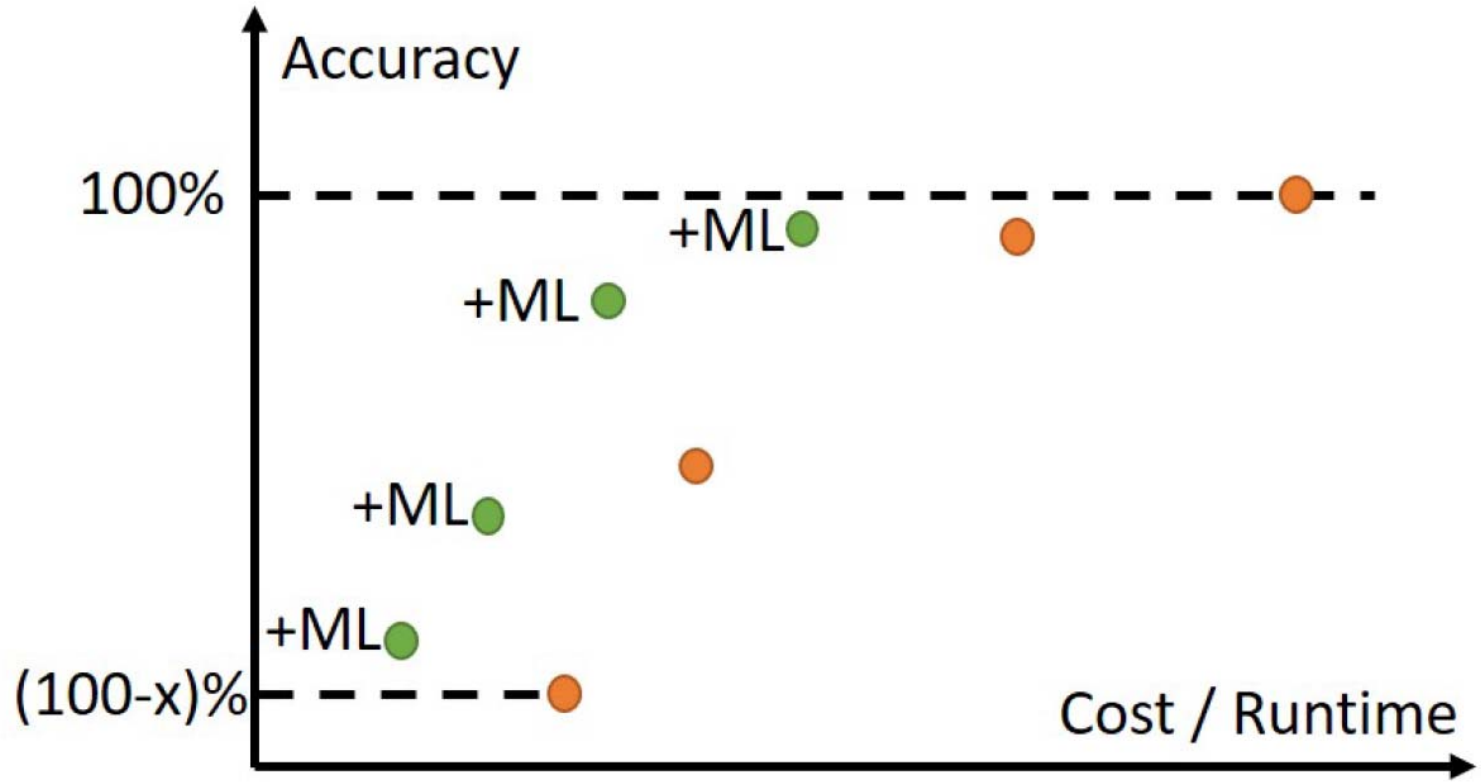
Latest P&R tools have increased #iterations → larger benefit in future ?

Errors	**Training (Total = 1200)**			**Testing (Total = 3442)**		
N = 200	Total Training Error	#TYPE 1 Errors (wrong STOP prediction)	#TYPE 2 Errors (no STOP)	Total Training Error	#TYPE 1 Errors (wrong STOP prediction)	#TYPE 2 Errors (no STOP)
1 STOP	29.66%	251	99	35.2%	1317	3
2 consecutive STOPs	10.5%	27	99	8.3%	307	3
3 consecutive STOPs	**8.5%**	**3**	**99**	**4.2%**	**154**	**3**

Agenda

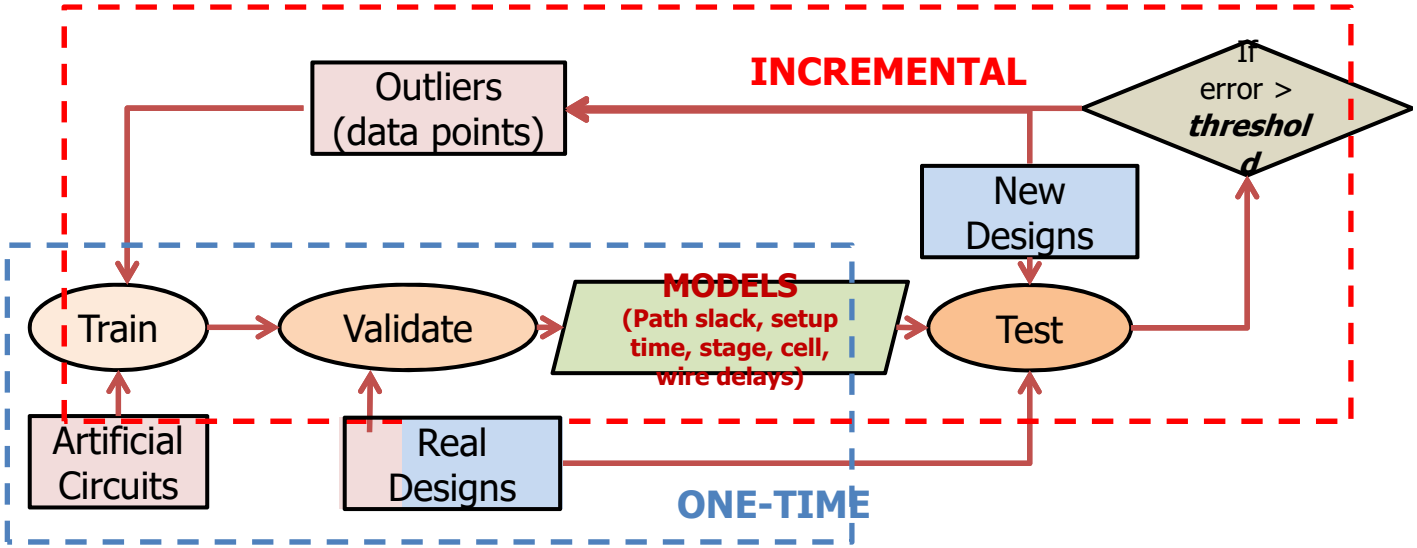
- Crises…
- … and a Vision
- Machine Learning in PD
- Modeling and Prediction
- Analysis Correlation

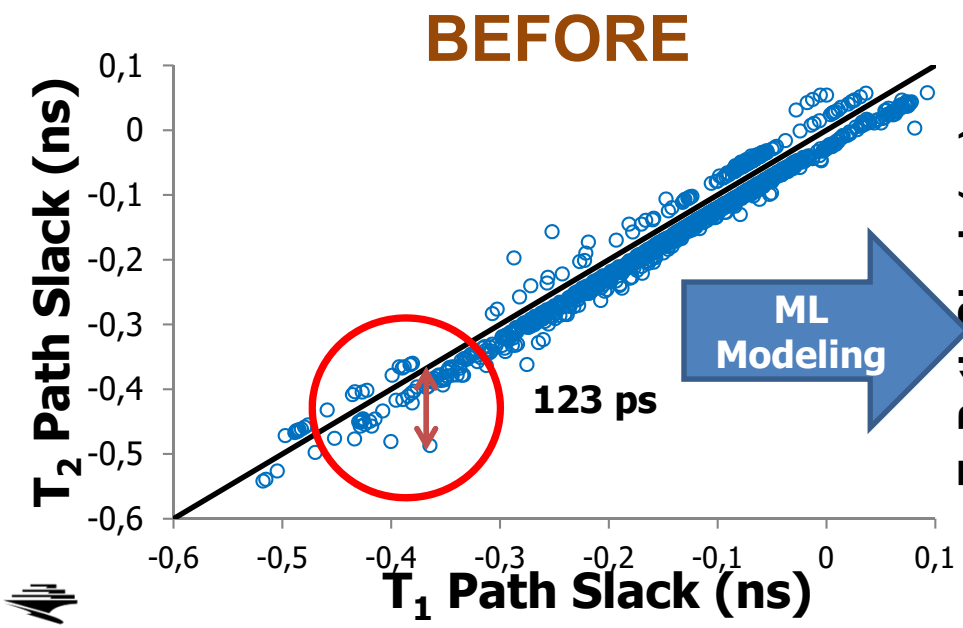
ML Shifts the Accuracy-Cost Tradeoff Curve!

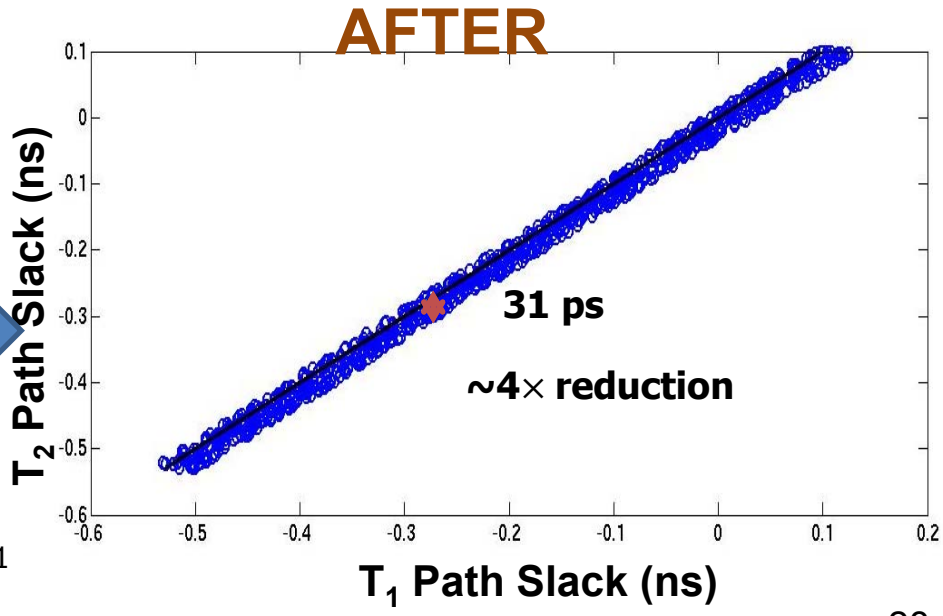


Example 4: ML-based Timer Correlation

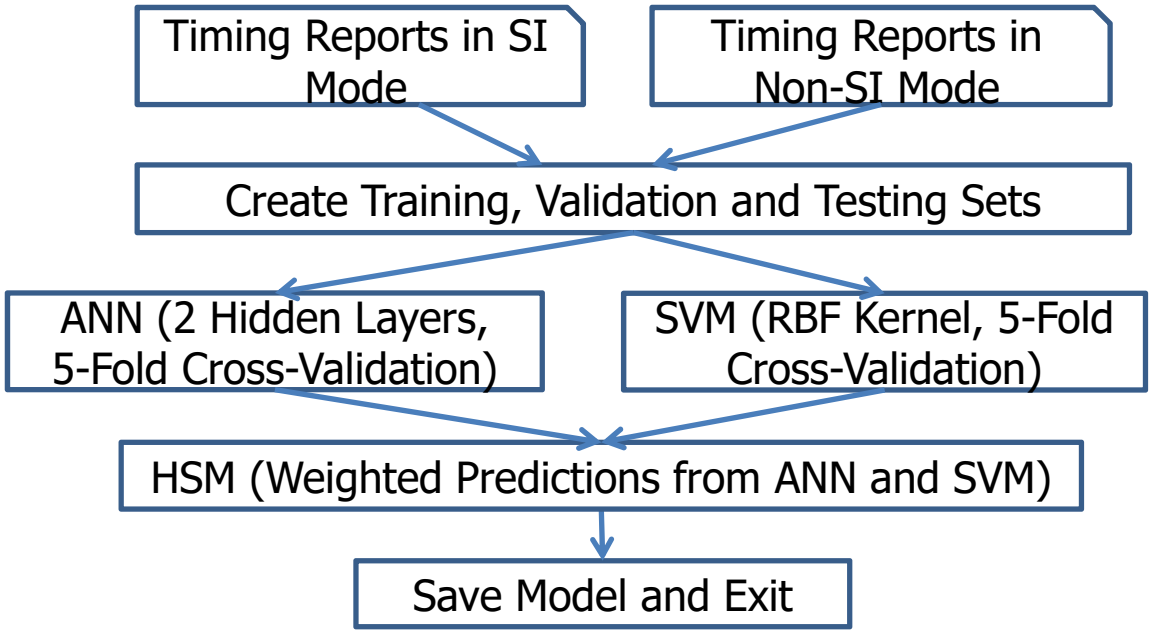
DATE-2014
(+ SLIP-2015)



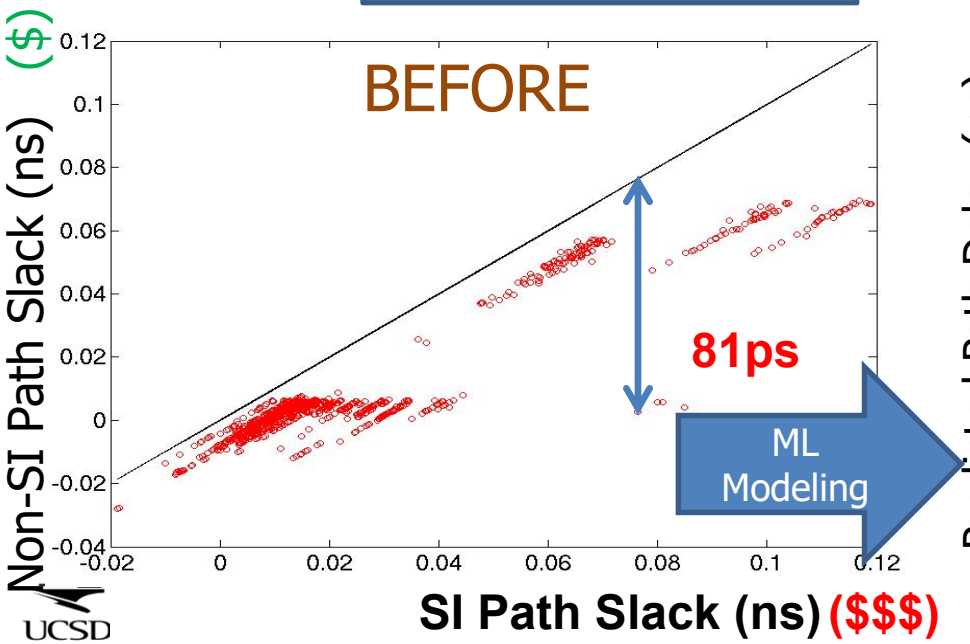


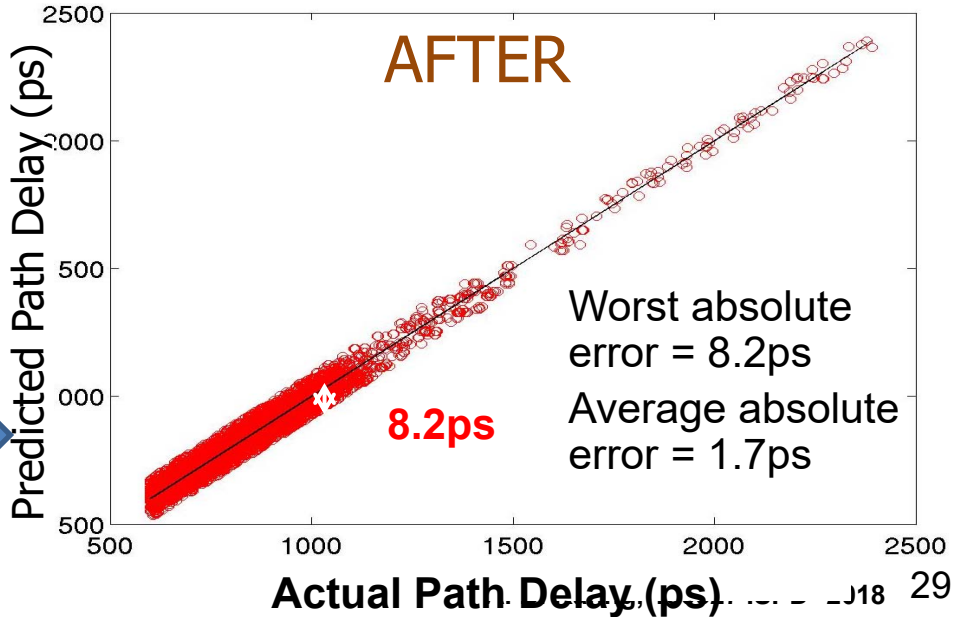


"SI for Free" with Machine Learning



- Machine learning of incremental transition time, delay due to SI
- Accurate SI-aware path delays, slacks

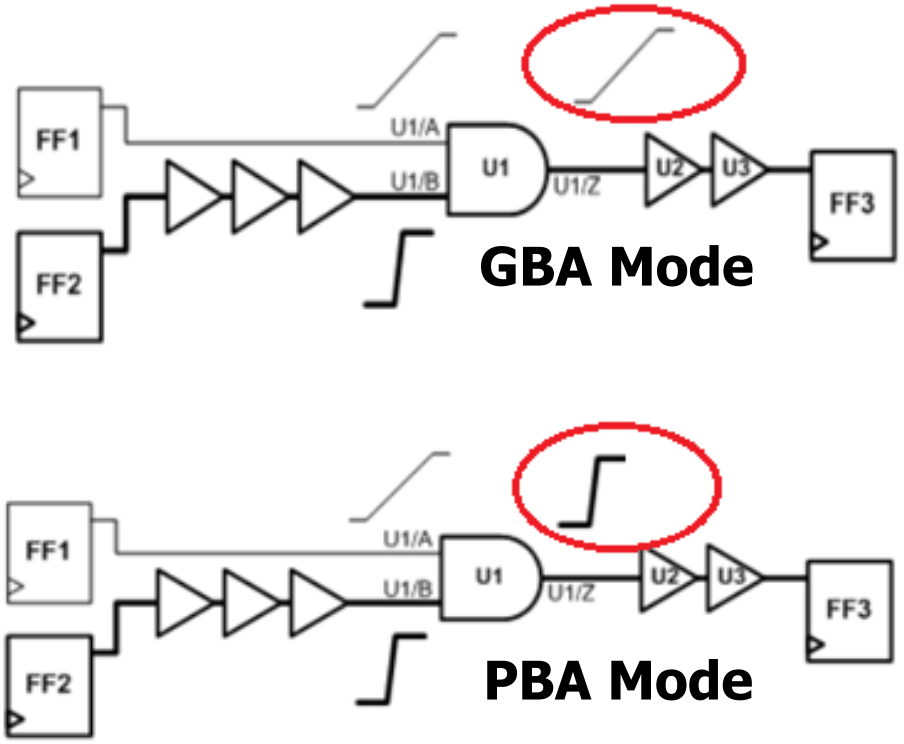


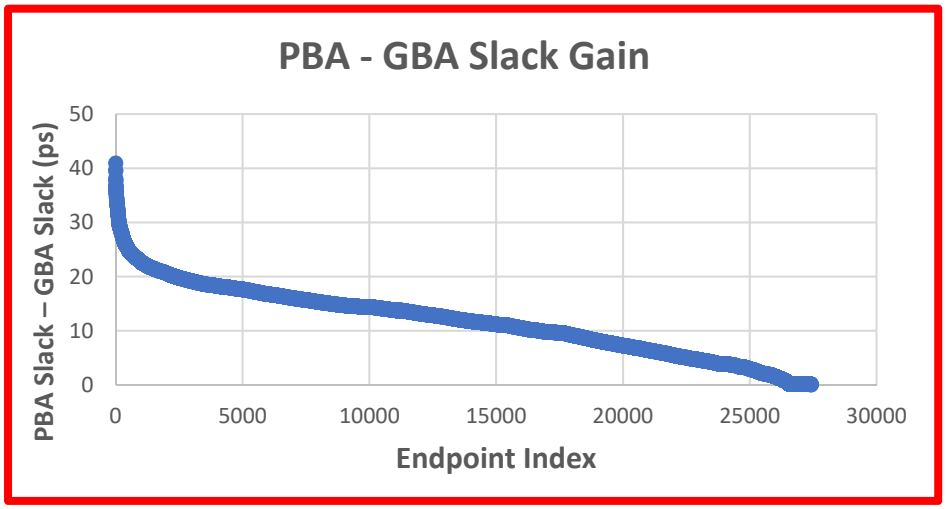


UCSD

Example 5: Predicting PBA from GBA?

- PBA (Path-Based Analysis) is less pessimistic than GBA (Graph-Based Analysis)
- But, more expensive runtime !
- **Question: Can we predict PBA timing from GBA timing?**
  - **→ Better optimization in P&R&Opt, less expensive STA**





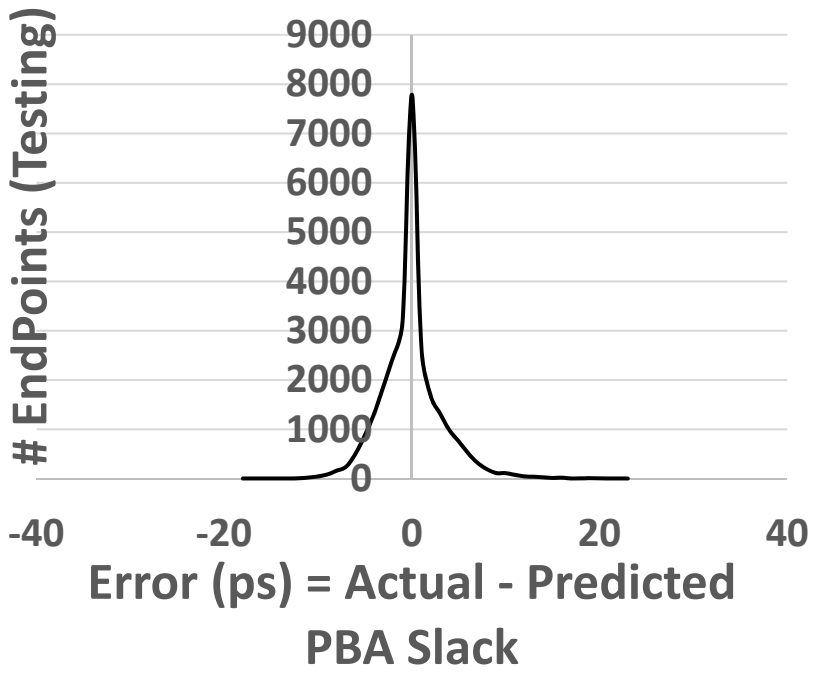
Costs of GBA vs. PBA Pessimism

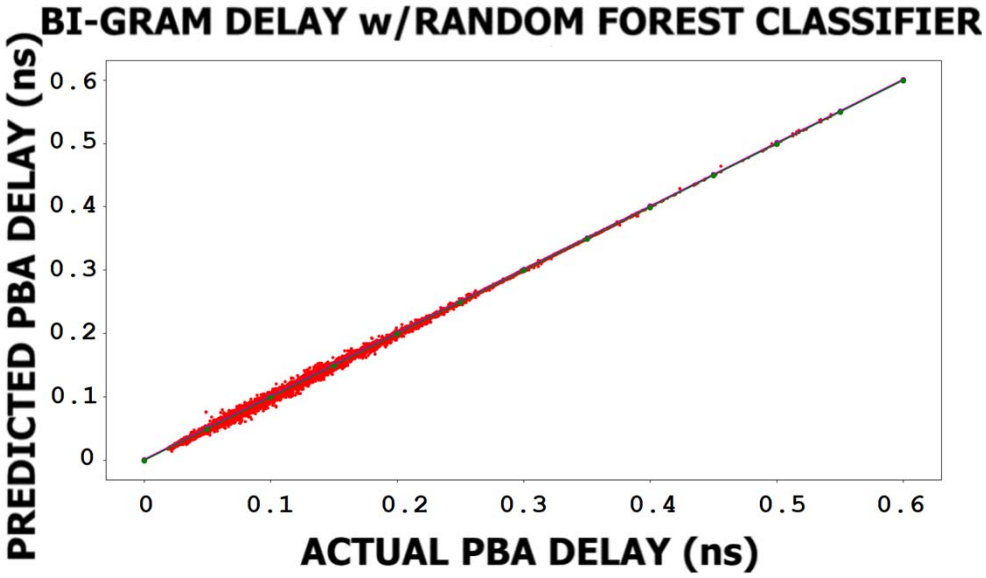
GBA Actual Slack	PBA Actual Slack	Impact
POSITIVE	**POSITIVE**	Power recovery can't exploit usable slack
NEGATIVE	**POSITIVE**	Schedule, Area, Power wasted fixing **false timing violations**
NEGATIVE	**NEGATIVE**	Schedule, Area, Power waste from **over-fixing**

PBA Actual Slack	PBA Predicted Slack (Model)	Impact
HIGH	**LOW**	Power recovery can't exploit all of usable slack
LOW	**HIGH**	Masking of real violations

Promising Initial Studies

- Early model with MARS (multiple adaptive regression splines): 90% of predicted PBA slacks within 5ps
- Also: random forest classifier for 2-stage “bi-grams”
- Testcase: netcard, 28nm FDSOI





Bi-gram =2-stage unit in timing path

Example 6: Reduce Corners in STA, Opt !

- Want benefits of STA at N corners, using just M << N corners
  - "Missing Corner Prediction" ("matrix completion") saves runtime, licenses
  - Avoids optimistic timing that is caught at detailed signoff, causing iteration

Model Training

C_1	C_2	C_3	...	C_{n-2}	C_{n-1}	C_n
0.434008	0.733149	0.234008	...	0.700667	0.556834	0.575091
0.373264	0.718715	0.273264	...	0.685265	0.551267	0.528366
0.350191	0.657966	0.250191	...	0.639694	0.508947	0.495575
0.394141	0.737795	0.294141	...	0.708014	0.565921	0.535571
0.375669	0.736253	0.237669	...	0.695926	0.560965	0.518217

Model Application

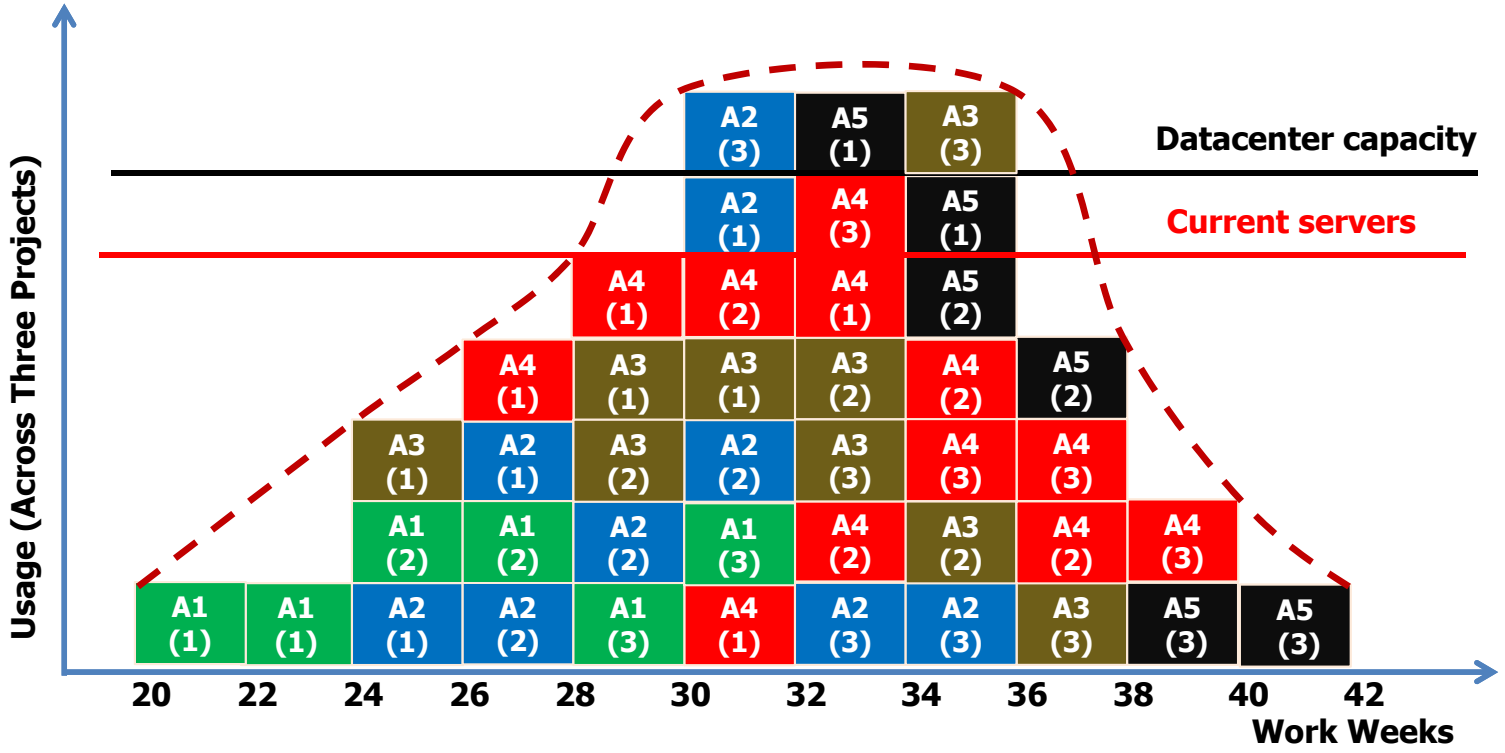
C_1	C_2	C_3	...	C_{n-2}	C_{n-1}	C_n
0.324854	0.623142	0.124708	...	0.591637	0.456314	0.455871
0.253674	0.639105	0.193234	...	0.597605	0.451987	0.438606
0.250441	0.538296	0.170112	...	0.543954	0.428477	0.377785
0.314232	0.601725	0.194119	...	0.628149	0.465281	0.438718
0.301009	0.618537	0.157692	...	0.585919	0.460605	0.426170

Agenda

- Crises…
- … and a Vision
- Machine Learning in PD
- Modeling and Prediction
- Analysis Correlation
- Optimization

Example 7: Design Cost Optimization

- Predictive models == Optimization objectives
- Enables schedule, resource optimizations up to enterprise level



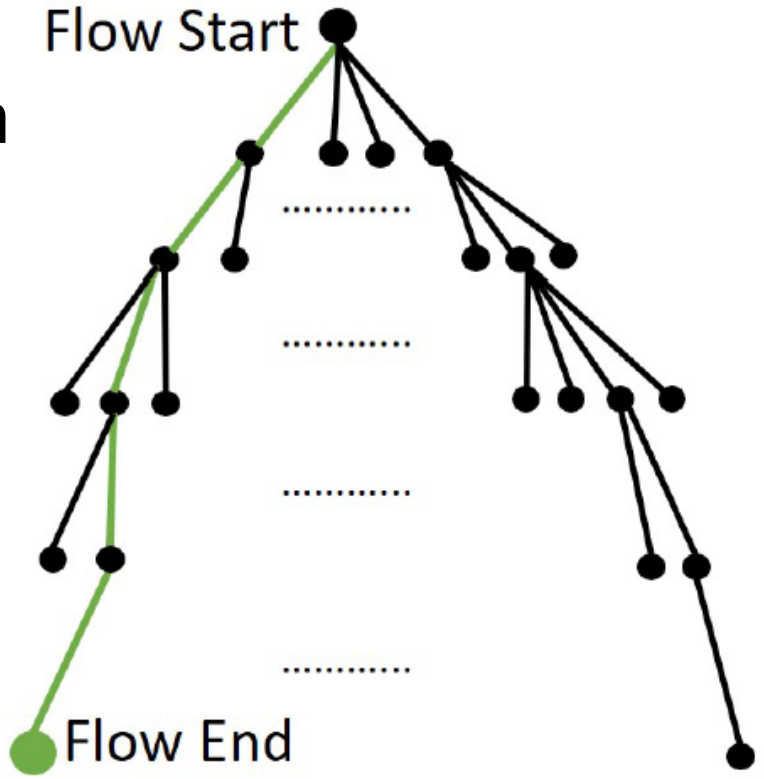
- TODAES 2017: Schedule Cost Minimization, Resource Cost Minimization ILPs
  - "How do I pack 12 tapeouts into my design center during Q4?

Agenda

- Crises…
- … and a Vision
- Machine Learning in PD
- Modeling and Prediction
- Analysis Correlation
- Optimization
- A Roadmap

Four Stages of ML Insertion in IC Design

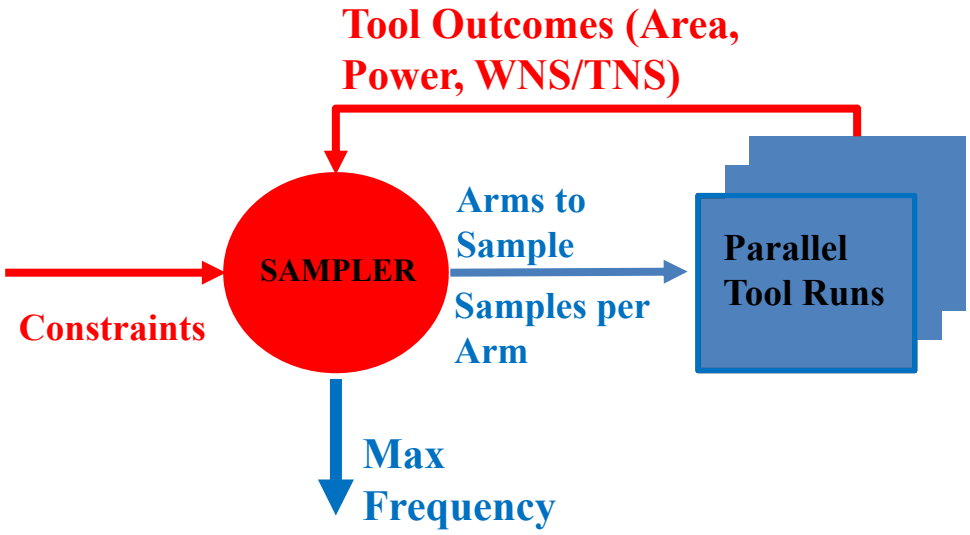
1. Mechanization and Automation
2. Orchestration of Search and Optimization
3. Pruning via Predictors and Models
4. Reinforcement Learning and Intelligence

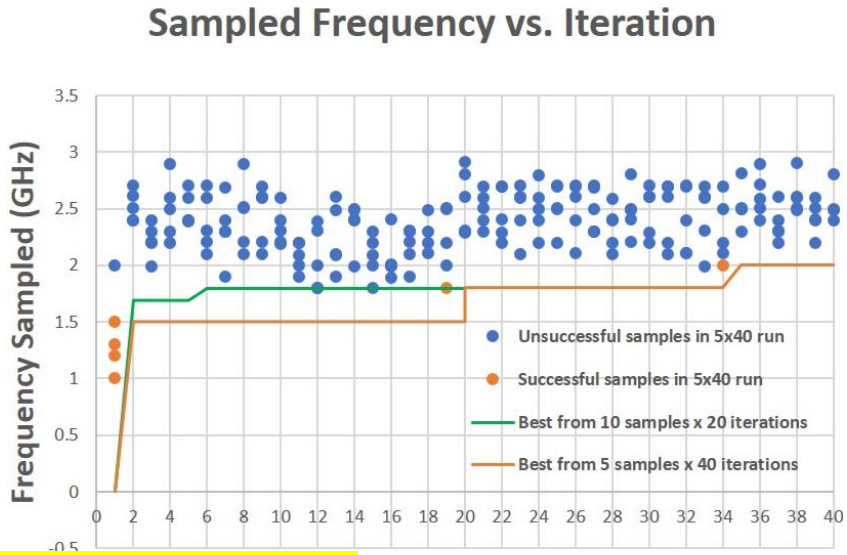


Huge space of tool, command, option trajectories through design flow

1. Mechanization and Automation

- **Create "robot IC design engineers"**
  - Observe and learn from humans
  - Search for command sequences in design tools
- **Multi-Armed Bandit Problem**: Given slot machine with N arms, maximize reward obtained using T pulls
  - Well-studied in context of Reinforcement Learning
- **IC Design: "arm"** = target frequency; **"pull"** = run flow

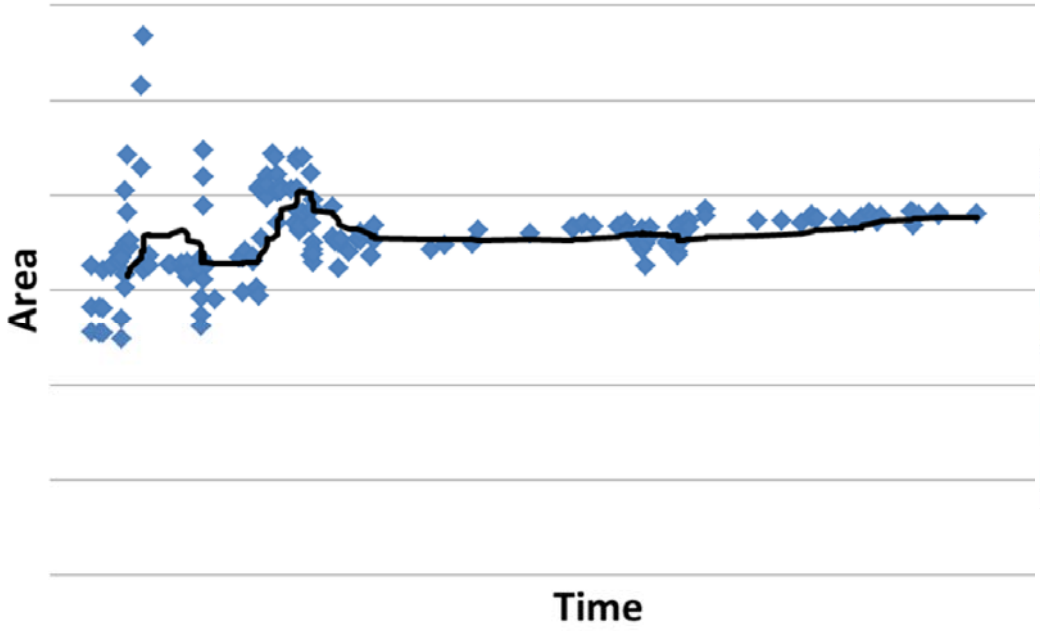


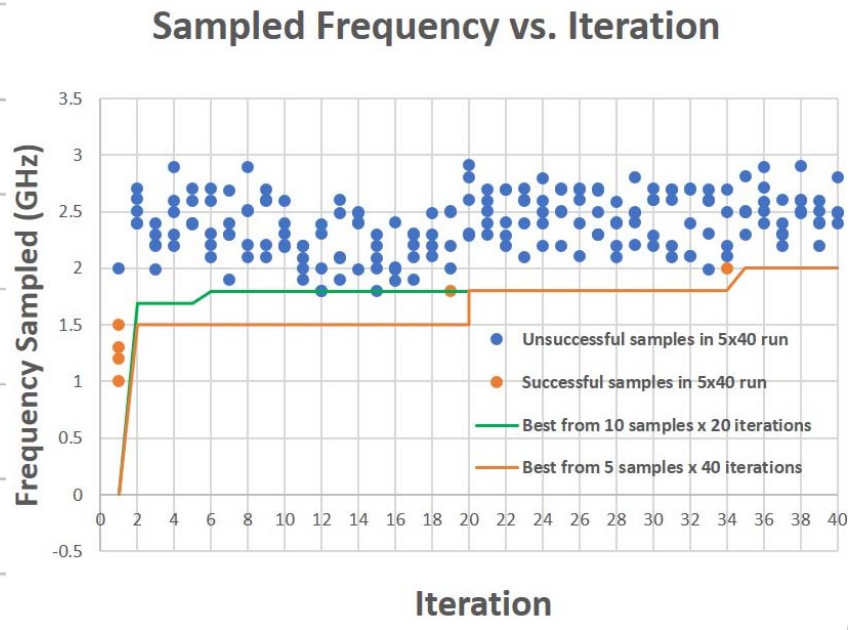


DAC-18 session: "The Road to No-Human-in-the-Loop IC Design" (UCSD, Qualcomm, Synopsys)

1. Mechanization and Automation

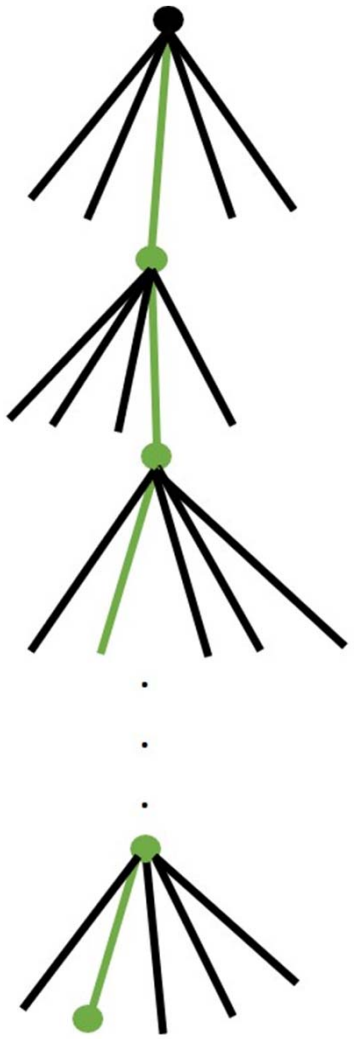
- **Create "robot IC design engineers"**
  - Observe and learn from humans
  - Search for command sequences in design tools
- **Multi-Armed Bandit Problem**: Given slot machine with N arms, maximize reward obtained using T pulls
  - Well-studied in context of Reinforcement Learning
- **IC Design: "arm"** = target frequency; **"pull"** = run flow





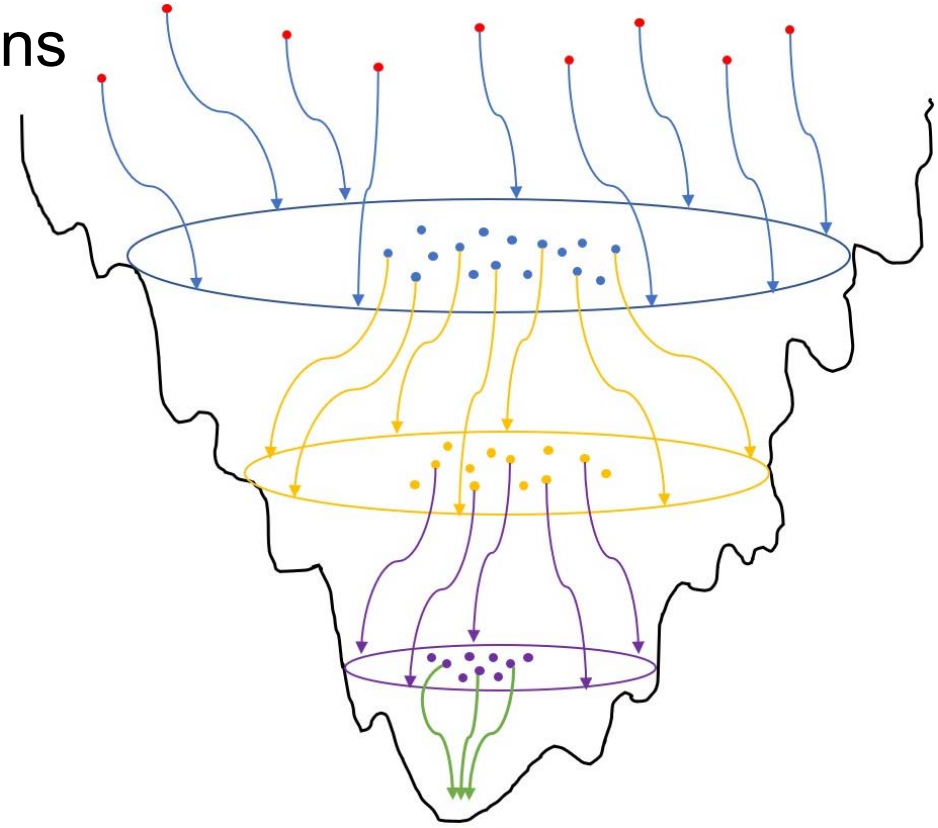
2. Orchestration of Search and Optimization

- **How to optimally orchestrate N robot engineers?**
  - Concurrent search of N flow trajectories
  - Explore, identify good flow options efficiently
  - Constraint: compute and license resources
- Goal: best QOR within resource, risk limits
- Example strategy: “Go with the winners”
  - Launch multiple optimization threads
  - Periodically identify promising thread
  - Clone promising thread and terminate others



Another Example: "Adaptive Multi-Start"

- Optimization cost landscapes often have "big valley" structures
  - Best local minima are central to all other local minima
- Adaptive Multi-Start (AMS)
  - Identify promising configurations in current iteration
  - Adaptively choose better start points for next optimization iteration



3. Pruning via Predictors and Models

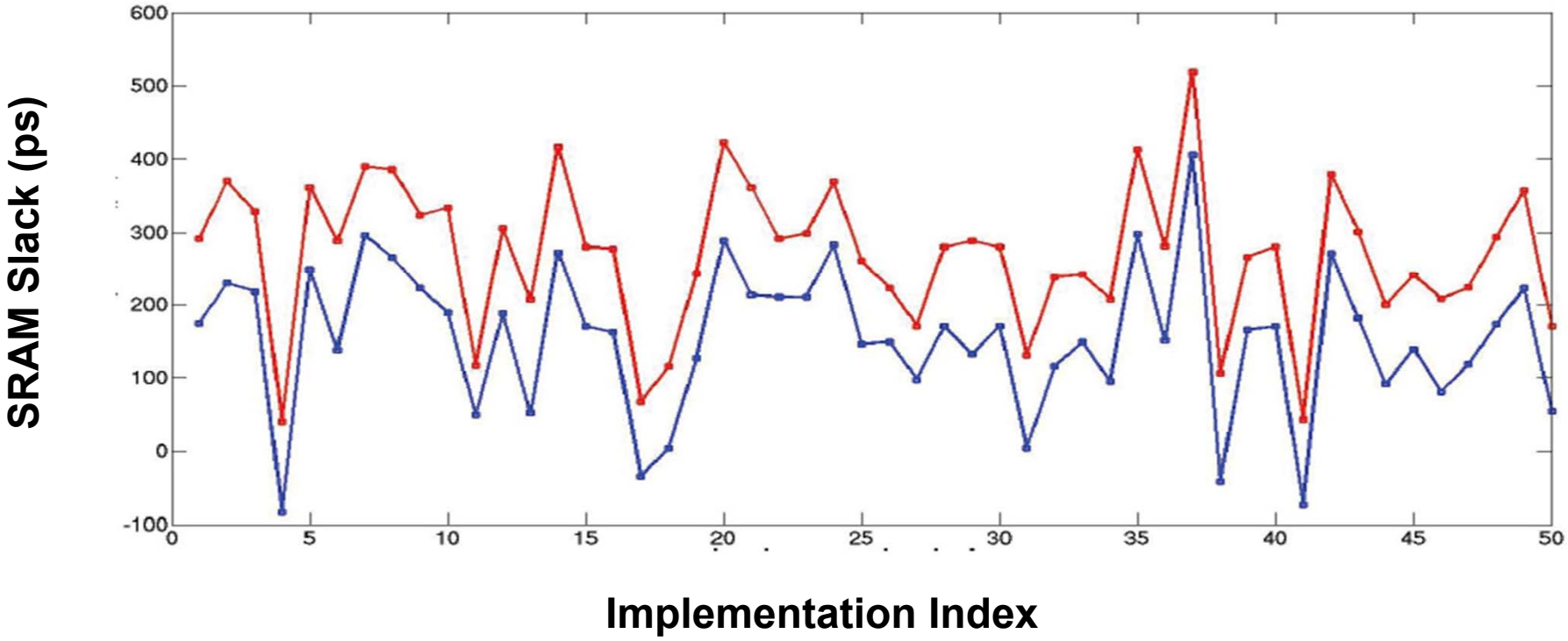
- Prediction of tool- and design-specific outcomes over longer and longer subflows
  - Wiggling of longer and longer ropes

Example 8: Prediction of SRAM Timing Failure

- Multiphysics effects (IR drop, thermal, etc.) affect timing closure
- Floorplanning with SRAMs is complicated
  - P&R blockages
  - Unpredictable post-P&R timing
- Goal: Early prediction of post-P&R slack (“doomed floorplans”) to save schedule
- But estimating post-P&R timing at floorplan stage is challenging:
  - Wire delay estimate has no spatial embedding information
  - Gate delay estimate has no buffering information

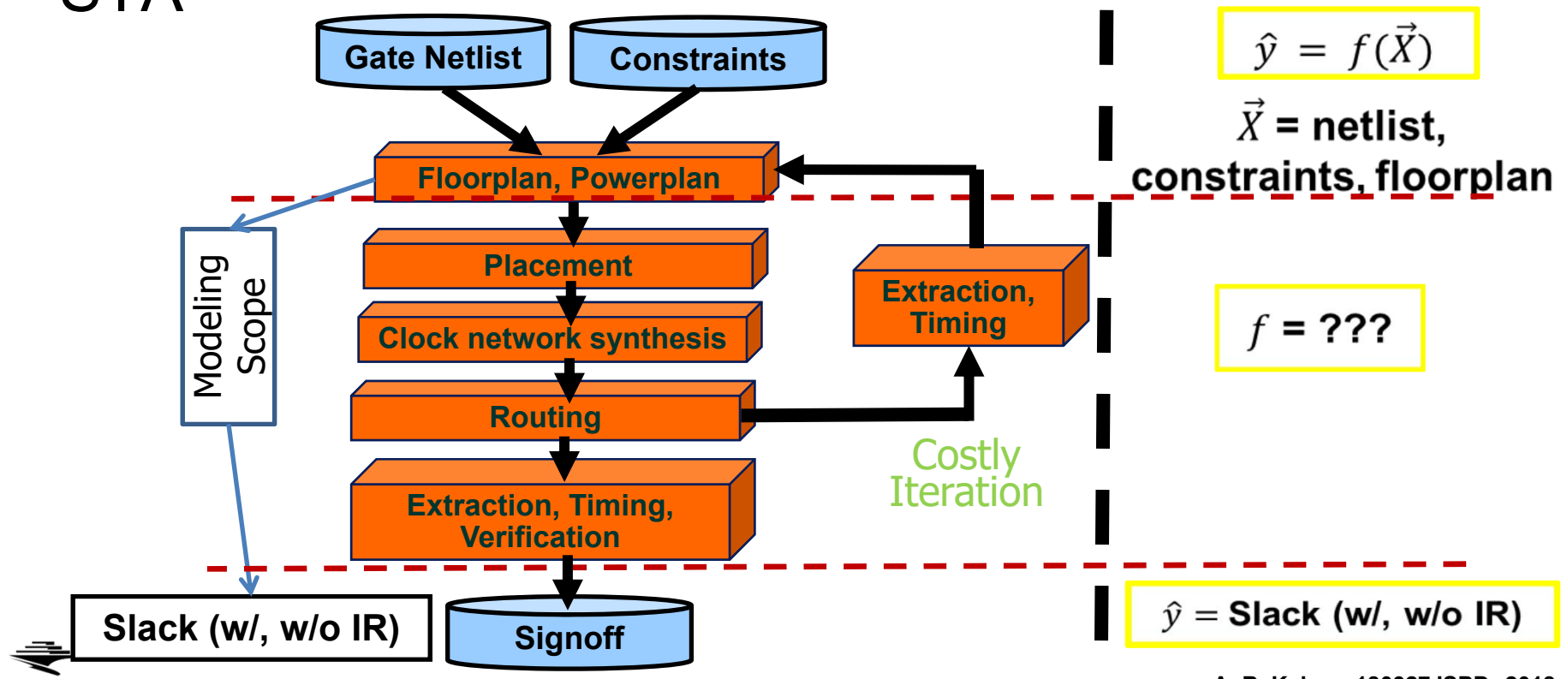
Multiphysics Analysis is Difficult to Predict

- IR drop, thermal, reliability, crosstalk, etc.
- **ASP-DAC 2016 (UCSD, Samsung):** Can we predict "risk map" for embedded memories at floorplan stage?

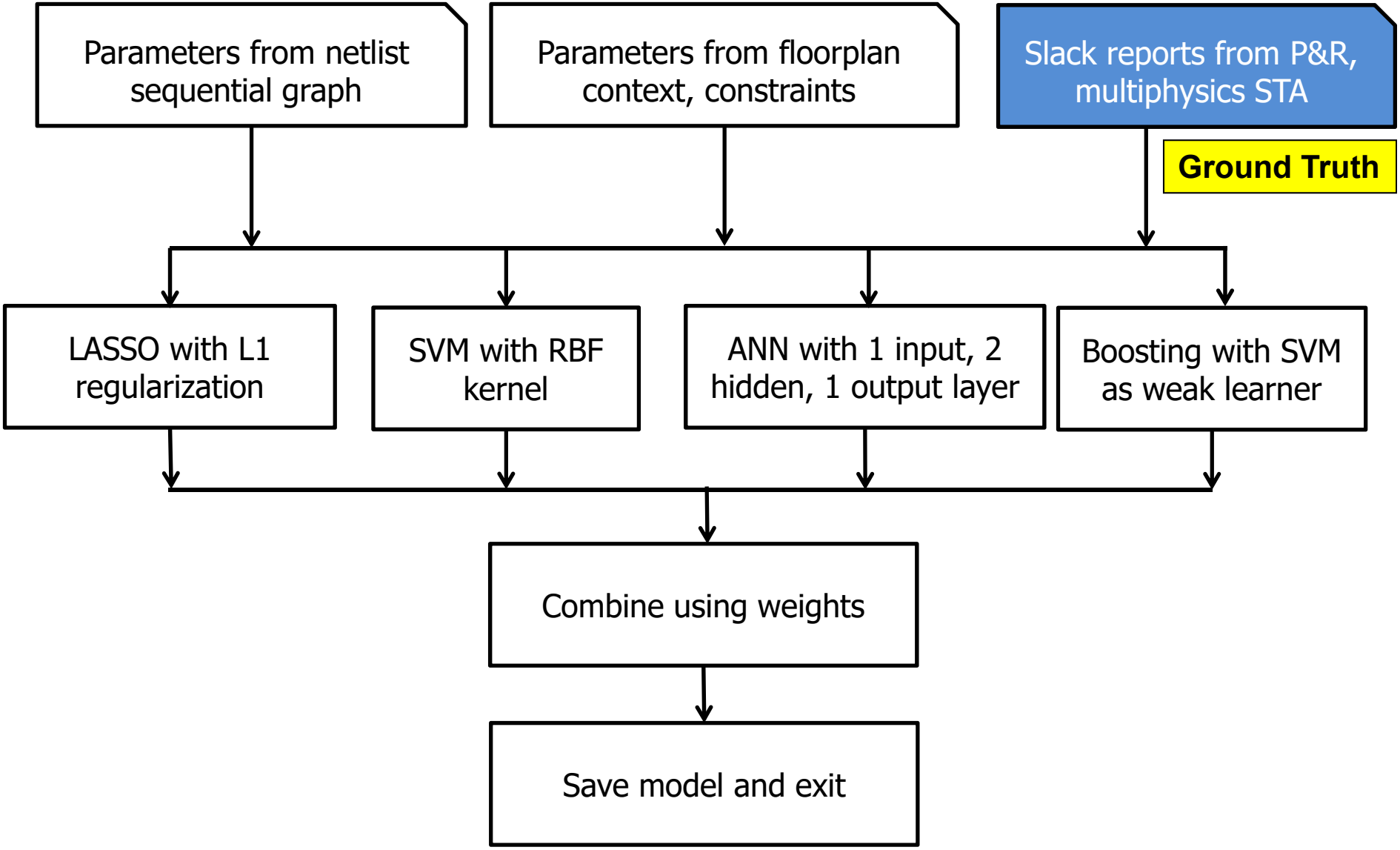


Floorplan Pathfinding with Machine Learning

- Filter bad floorplans (e.g., embedded memory placements, power plans) comprehending downstream PD flow
- Model ***f*** estimates combined effects of netlist, constraints, placement, CTS, routing, optimization, STA

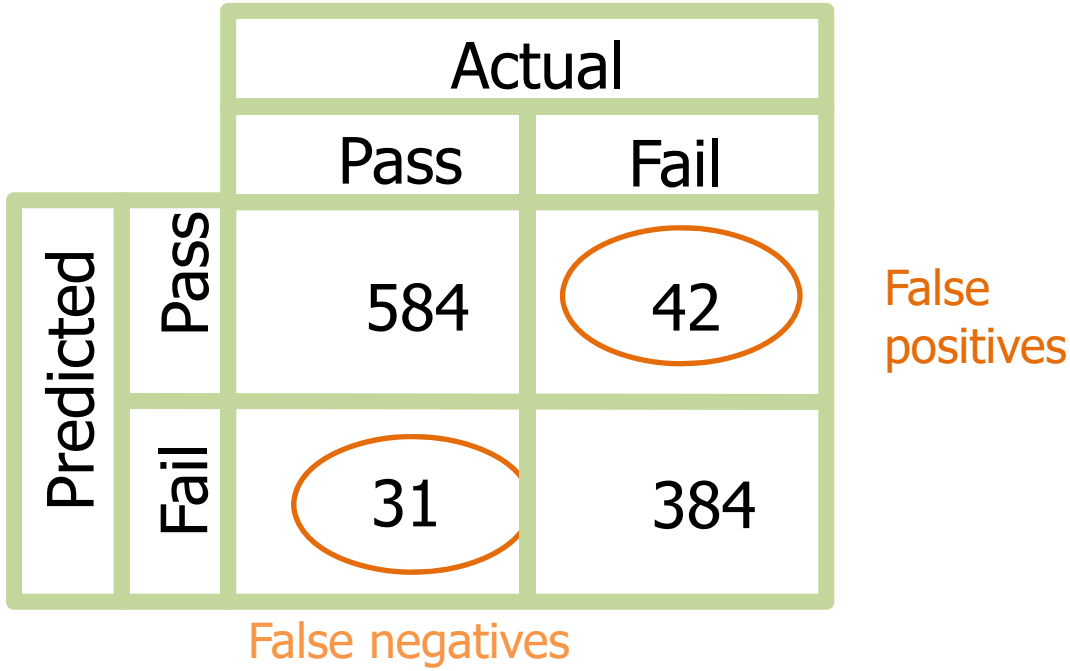


Modeling Techniques and Flow



Floorplan Pathfinding Model

- False negatives = 3%
  - Pessimistic predictions → floorplan change that is actually not required
- False positives = 4%
  - Model incorrectly deems a floorplan to be good



3. Pruning via Predictors and Models

- Prediction of tool- and design-specific outcomes over longer and longer subflows
  - Wiggling of longer and longer ropes
- Prune, terminate → avoid wasted design resources
  - Better outcome within given resource budget
- **Implicit: improved *predictability* and *modelability* of heuristics and tools**

4. Reinforcement Learning and "Intelligence"

Many challenges on the road ahead…

- Latency and unpredictability of IC design tools/flows
  - Can't "play the IC design game" 100M times in 3 days
- "Small data" challenge with a big-data problem
  - Data points are expensive
  - Huge implementation space
  - Tool versions, design versions, technology all changing (pictures of cats and trees don't change)
- Model parameters come from domain experts today
- Open: bridging real (top-secret!) and artificial (fake!)
  - My group: many years of "eye chart" papers

Todo List: "Last Mile" Robots

- **Automation of manual DRC violation fixing**
  - P&R tools cannot handle latest rule decks, unavoidable lack of routing resource in high-utilization block, etc.
- **Automation of manual timing closure**
  - After routing and optimization, several thousand violations of maxtrans, setup, hold constraints exist
  - Engineer fixes 200-300 DRVs by hand, per day
- **Placement of memory instances in a P&R block**
- **Package layout automation**
  - How to assess post-routed quality (e.g., bump inductances) of SOC floorplan and die-package pin map?
  - Required for: pin map, power delivery optimization
  - Requires: automation/estimation of manual package routing

Todo List: Improving Analysis Correlation

- **Prediction of the worst PBA path**
- **Prediction of the worst PBA slack per endpoint, from GBA analysis**
- **Prediction of timing at "missing corners"**
  - Predict other impacts (e.g., transition times, ..) of an ECO as well
- **Closing of multi-physics analysis loops**
  - **Early priorities: vectorless dynamic IR drop, power-temperature loops**
- Continued improvement of timing correlation and estimation !
  - Faster and better always helpful !

Todo List: Predictive Models of Tools, Designs

- Predict convergence point for P&R, non-uniform PDN
- Estimate PPA response of block to floorplan context
- Estimate useful skew impact on post-route WNS,TNS
- "Auto-magic" determination of netlist constraints for given performance and power targets
  - Key opportunity: exactly ONE netlist is passed into place-and-route – how to generate this best netlist?
- Predict best "target sequence" of constraints through layout optimization phases
- Predict "most-optimizable" cells during design closure
- Predict divergence (detouring , timing/slew violations) between trial/global route and final detailed route
- Predict "doomed runs" at all steps of design flow

Todo List: And More…

- **Infrastructure for machine learning in IC design**
  - Standards for model encapsulation, model application, and IP preservation when models are shared
- **Standard ML platform for EDA modeling**
  - Enablement of design metrics collection, tool/flow model generation, design-adaptive tool/flow configuration, prediction of tool/flow outcomes
  - This recalls “METRICS” http://vlsicad.ucsd.edu/GSRC/metrics
- **Modelable algorithms and tools**
  - Smoother, less chaotic outcomes than present methods
- **Datasets to support ML**
  - Artificial circuits and “eyecharts”
  - Shared training data – e.g., timer correlation, post-route DRV prediction, optimal sizing

Agenda

- Crises…
- … and a Vision
- Machine Learning in PD
- Modeling and Prediction
- Analysis Correlation
- Optimization
- A Roadmap
- Conclusion

Conclusion

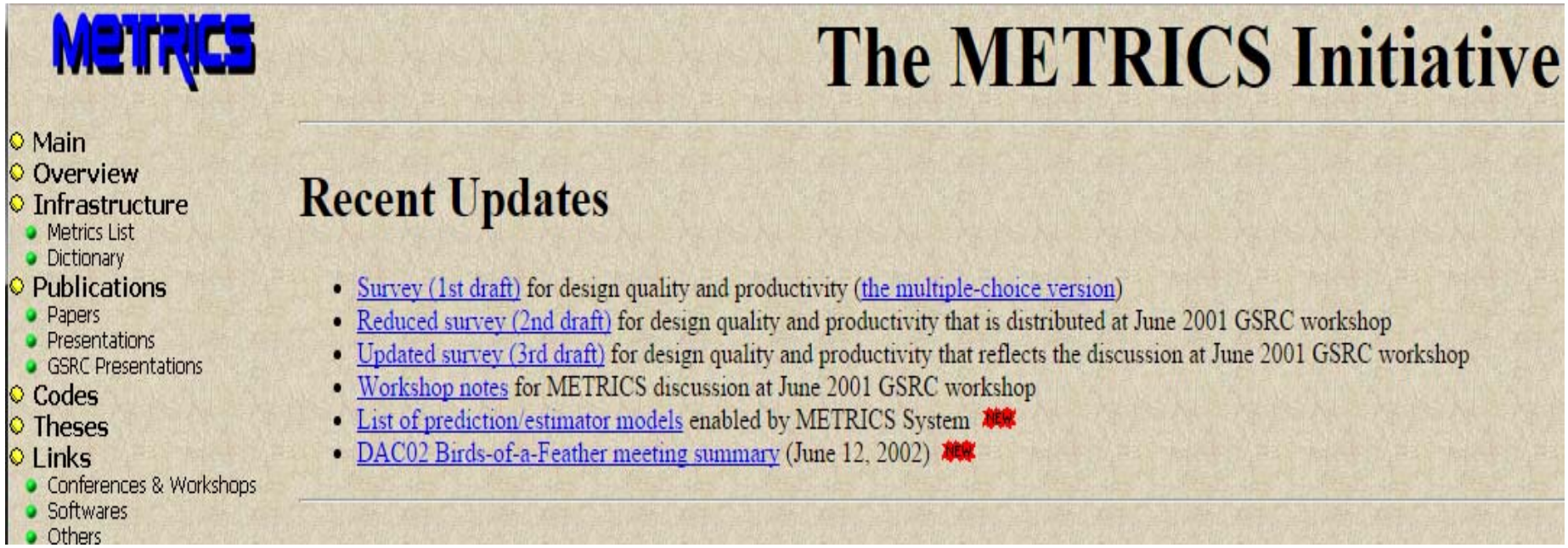
- Many high-value opportunities for ML in physical design
  - Analysis correlation → less margin, improved design QOR, faster convergence
  - Predictive modeling of tools/flows and designs → fewer loops, less wasted effort, less pessimism, better design optimization, better resource management
- Roadmap
  - Robots
  - Orchestration of robots
  - Pruning via predictors and models
  - Intelligence + many specific “todos”
- Other facets: enablement, standards, openness,…
- **I hope that many of you will join this quest !!!**

THANK YOU !

Support from NSF, Qualcomm, Samsung, NXP, Mentor Graphics and the C-DEN center is gratefully acknowledged.

(This is "METRICS" !)

[ISQED01]

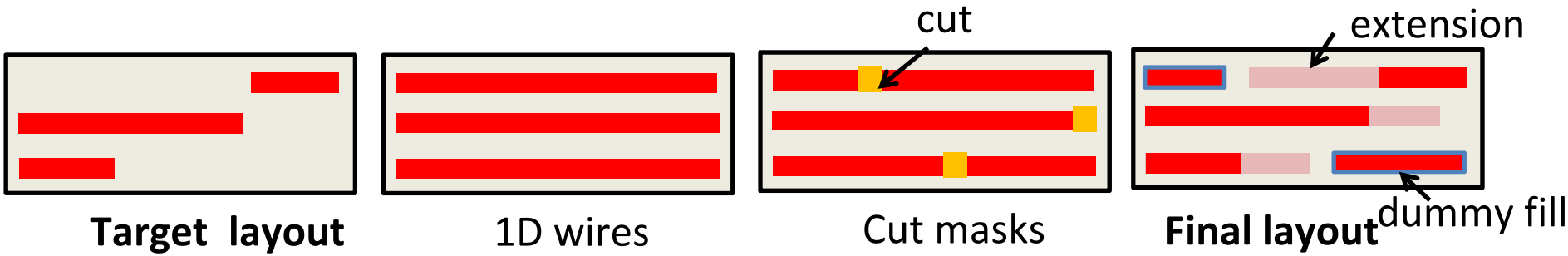


- METRICS (1999; ISQED01): "Measure to Improve"
  - Goal #1: Predict outcome
  - Goal #2: Find sweet spot (field of use) of tool, flow
  - Goal #3: Dial in design-specific tool, flow knobs

http://vlsicad.ucsd.edu/GSRC/metrics

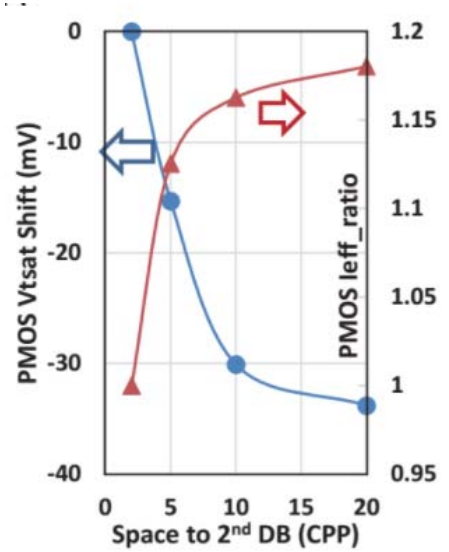
Patterning and Margins for Wires ("BEOL")

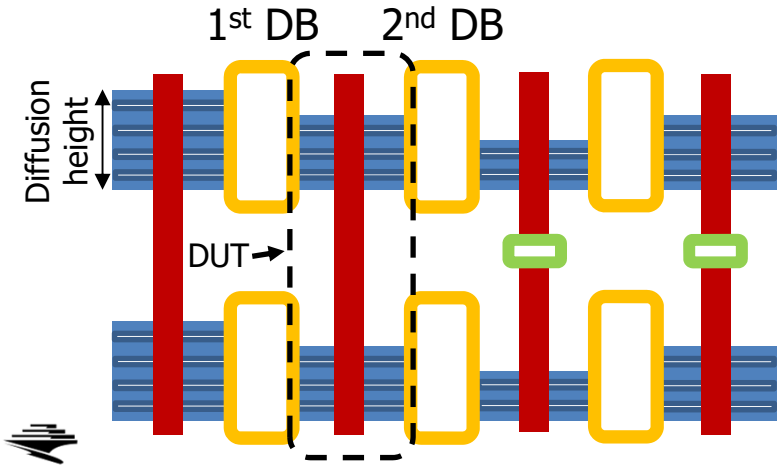
- Self-aligned multiple patterning + Cutmask
- Make a "sea of wires"
- Make "cuts"
- Cut shapes and locations determine **dummy wires** and **end-of-line extensions** of wire segments
- **Final layout ≠ Target layout**
  - → Timing and power not the same as originally designed !
  - → Need more margin !

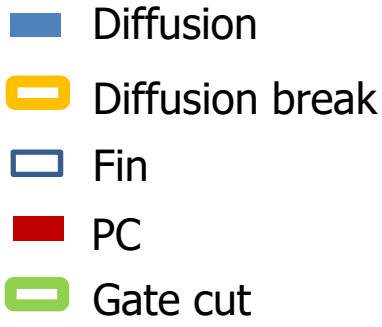


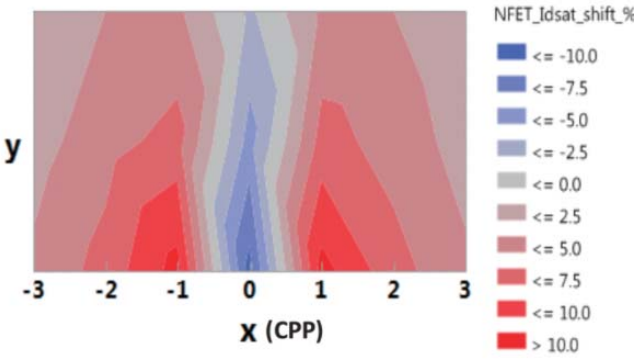
Patterning and Margins for Gates ("FEOL")

- **Neighbor diffusion effect (NDE)**
  - Diffusion step = neighboring diffusion area height change
  - Transistor drive strength and leakage prop. to horizontal fin spacing
- **2nd Diffusion Break (DB)**
  - Vt shift as a function of spacing to the 2nd diffusion break
- **Gate Cut (GC)**
  - Idsat shifts as a function of gate-cut distance to DUT
- **Worst corner has to consider NDE + 2nd DB + GC**
  → **More margin added besides PVT (!)**







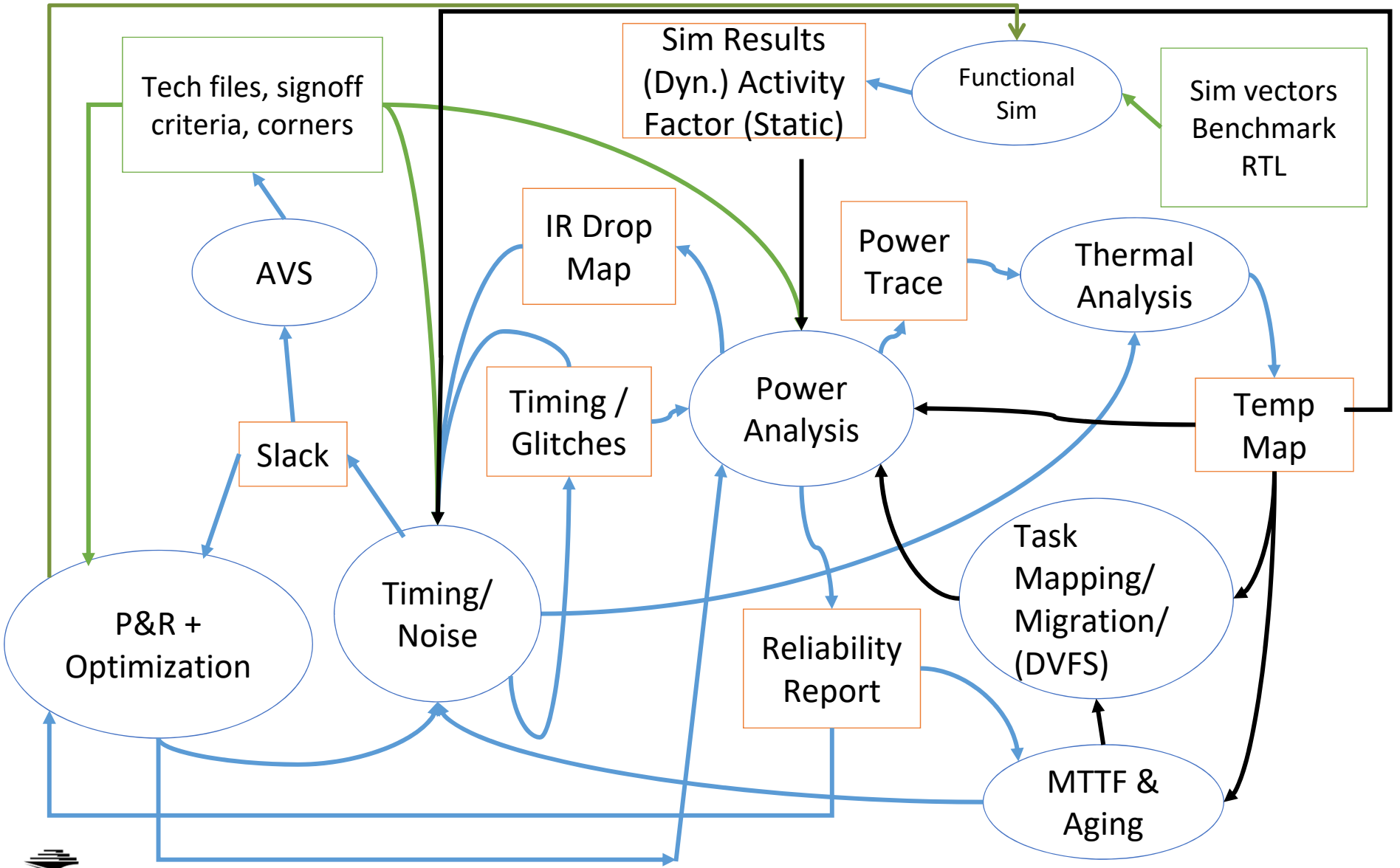


Gate Cut (GC) Effect

UCSD

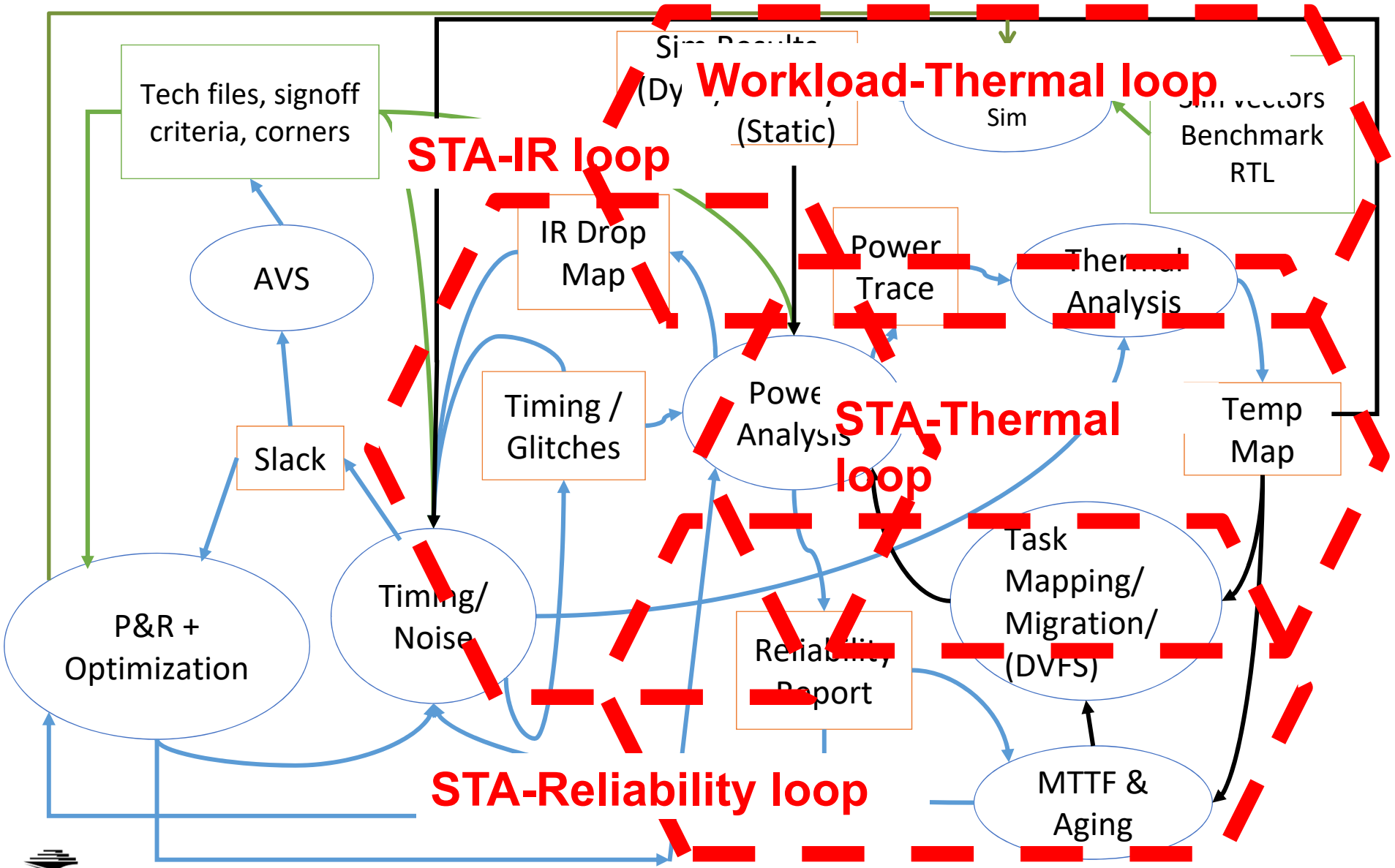
Closing Multiphysics Analysis Loops

[ASPDAC16]



Closing Multiphysics Analysis Loops

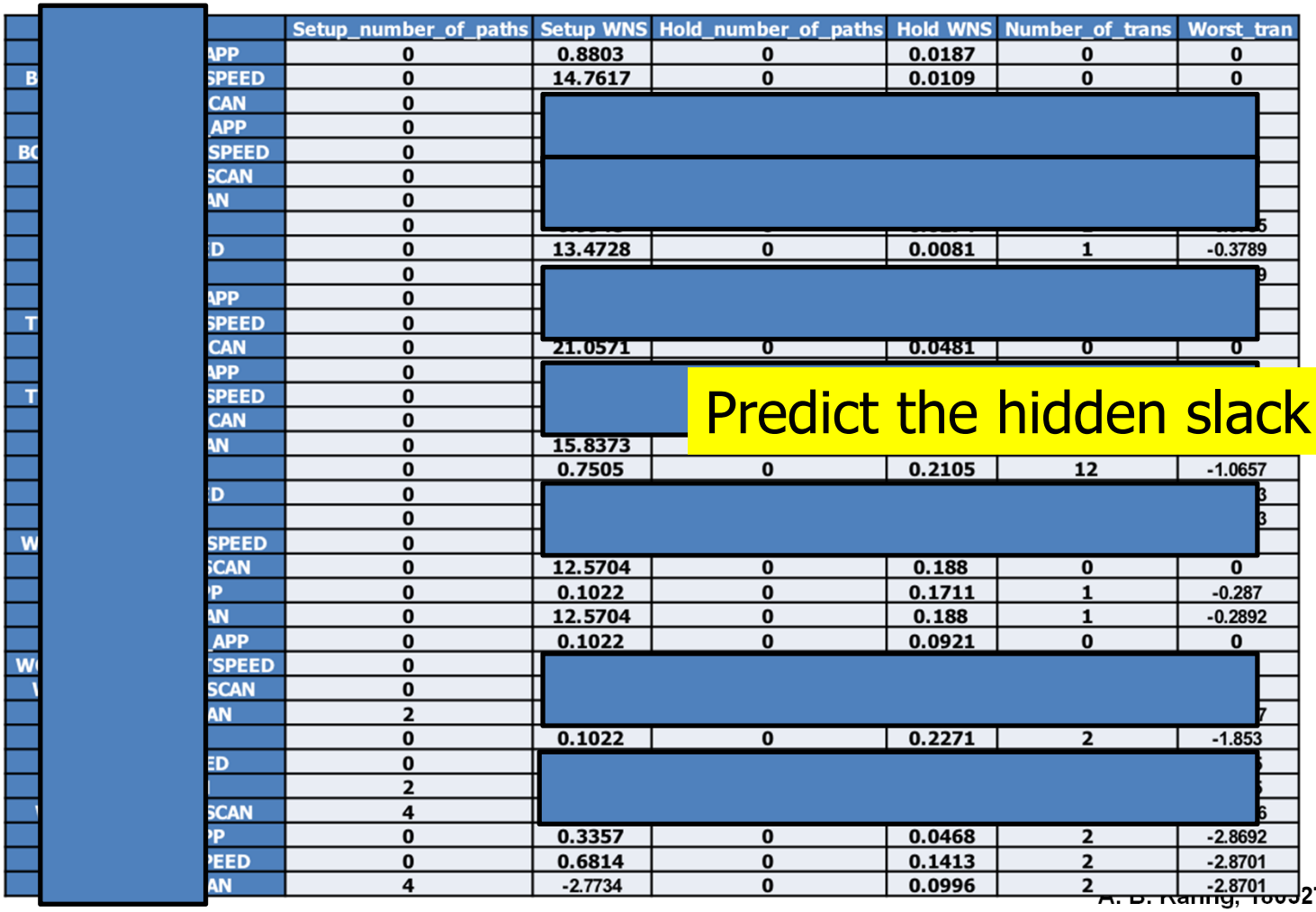
[ASPDAC16]



BACKUP

Many Operating Conditions ("Corners")

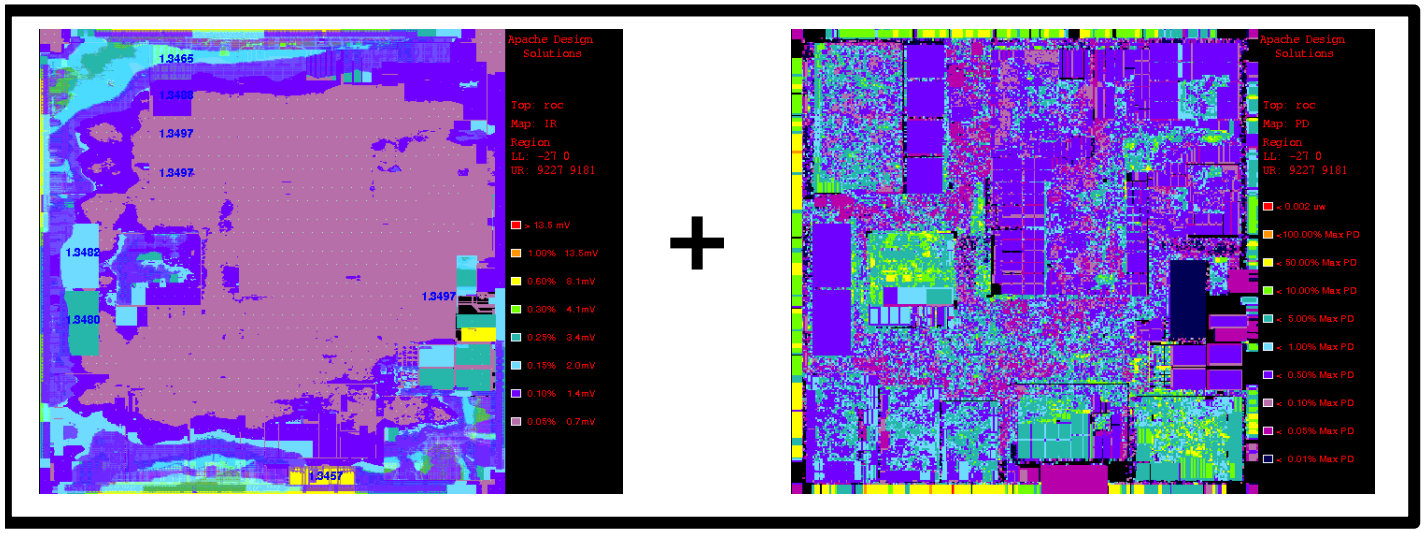
- Chip must work at many (500+) operating conditions (corners)
- Each corner = another run of the timing tool
- **GOAL:** Run as few timing corners as possible; predict the rest



	Setup_number_of_paths	Setup WNS	Hold_number_of_paths	Hold WNS	Number_of_trans	Worst_tran
	0	0.8803	0	0.0187	0	0
	0	14.7617	0	0.0109	0	0
	0					
	0					
	0					
	0					
	0					
	0					
	0	13.4728	0	0.0081	1	-0.3789
	0					
	0					
	0					
	0	21.0571	0	0.0481	0	0
	0					
	0					
	0					
	0	15.8373				
	0	0.7505	0	0.2105	12	-1.0657
	0					
	0					
	0					
	0	12.5704	0	0.188	0	0
	0	0.1022	0	0.1711	1	-0.287
	0	12.5704	0	0.188	1	-0.2892
	0	0.1022	0	0.0921	0	0
	0					
	0					
	2					
	0	0.1022	0	0.2271	2	-1.853
	0					
	2					
	4					
	0	0.3357	0	0.0468	2	-2.8692
	0	0.6814	0	0.1413	2	-2.8701
	4	-2.7734	0	0.0996	2	-2.8701

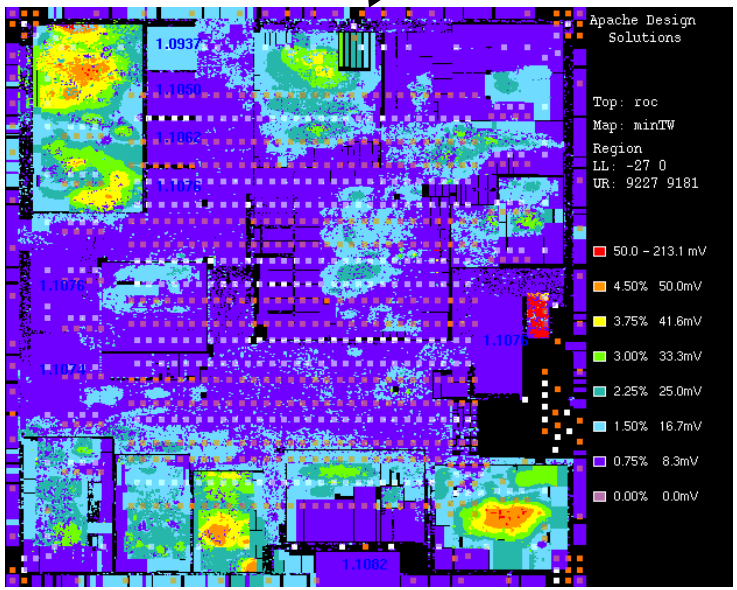
Predict the hidden slack values!

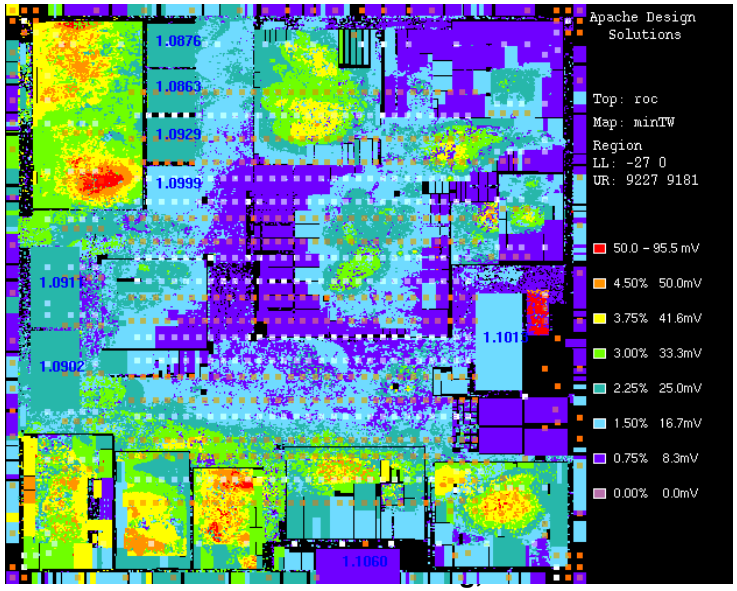
And a Dream … [predicting dynamic voltage drop]



Inexpensive
Static analysis
+ Current map

Expensive
Dynamic analyses





Some References

Highlighted in the talk from ABKGroup

- [RISKMAP] W.-T. J. Chan, K. Y. Chung, A. B. Kahng, N. D. MacDonald and S. Nath, "Learning-Based Prediction of Embedded Memory Timing Failures During Initial Floorplan Design", (.pdf), *Proc. ASPDAC*, 2016.
- [GT1GT2]] S. S. Han, A. B. Kahng, S. Nath and A. Vydyanathan, "A Deep Learning Methodology to Proliferate Golden Signoff Timing", (.pdf), *Proc. DATE*, 2014.
- [GT1GT2] A. B. Kahng, M. Luo and S. Nath, "SI for Free: Machine Learning of Interconnect Coupling Delay and Transition Effects", (.pdf), *Proc. SLIP*, 2015.
- [#ML/ROPT] W.-T. J. Chan, Y. Du, A. B. Kahng, S. Nath and K. Samadi, "BEOL Stack-Aware Routability Prediction from Placement Using Data Mining Techniques", (.pdf), *Proc. ICCD*, 2016.
- [#ML/ROPT] W.-T. J. Chan, P.-H. Ho, A. B. Kahng and P. Saxena, "Routability Optimization for Industrial Designs at Sub-14nm Process Nodes Using Machine Learning", (.pdf), *Proc. ISPD*, 2017.
- [CTS] K. Han, A. B. Kahng, J. Lee, J. Li and S. Nath, "A Global-Local Optimization Framework for Simultaneous Multi-Mode Multi-Corner Skew Variation Reduction",(.pdf), *Proc. DAC*, 2015.

Some other machine learning / data mining papers from ABKGroup

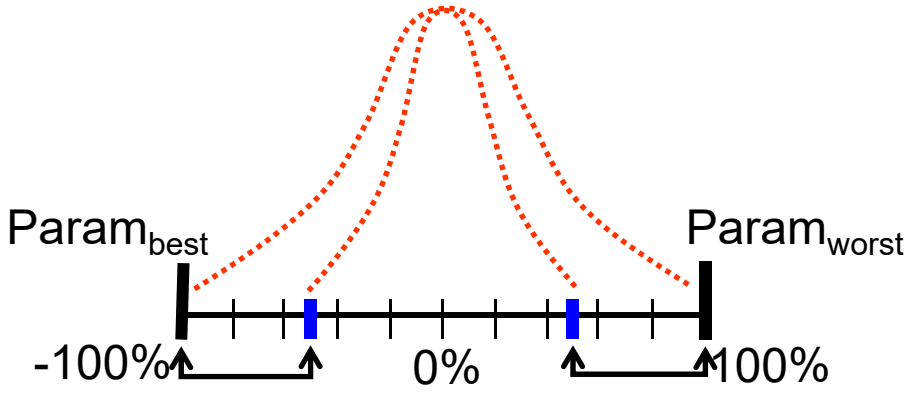
- [3DPE] W.-T. J. Chan, Y. Du, A. B. Kahng, S. Nath and K. Samadi, "3D-IC Benefit Estimation and Implementation Guidance from 2D-IC Implementation", (.pdf), *Proc. DAC*, 2015.
- [HS] A. B. Kahng, C.-H. Park and X. Xu, "Fast Dual-Graph Based Hotspot Detection" (.pdf), *Proc. BACUS, 2006.*
- [INT] A. B. Kahng, S. Kang, H. Lee, S. Nath and J. Wadhwani, "Learning-Based Approximation of Interconnect Delay and Slew in Signoff Timing Tools", (.pdf), *Proc. SLIP*, 2013.
- [METRICS] S. Fenstermaker, D. George, A. B. Kahng, S. Mantik and B. Thielges, "METRICS: A System Architecture for Design Process Optimization", (.pdf), *Proc. DAC*, 2000.
- [METRICS] A. B. Kahng and S. Mantik, "A System for Automatic Recording and Prediction of Design Quality Metrics", (.pdf), *Proc. ISQED,* 2001.
- [HSM] A. B. Kahng, B. Lin and S. Nath, "Enhanced Metamodeling Techniques for High-Dimensional IC Design Estimation Problems", (.pdf), *Proc. Design, Automation and Test in Europe*, 2013, pp. 1861-1866.
- [HHSM] A. B. Kahng, B. Lin and S. Nath, "High-Dimensional Metamodeling for Prediction of Clock Tree Synthesis Outcomes", (.pdf), *Proc. ACM/IEEE International Workshop on System-Level Interconnect Prediction*, 2013.
- [METRICS] GSRC/METRICS: http://vlsicad.ucsd.edu/GSRC/metrics/

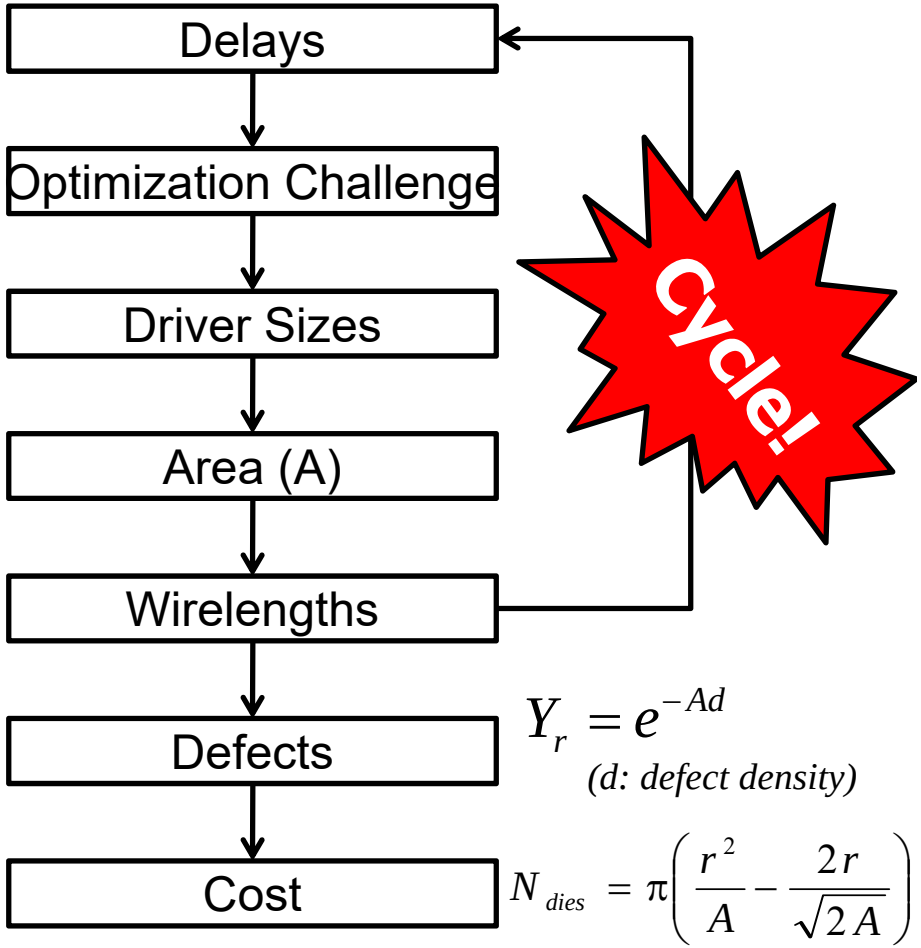
See also: Center for Design-Enabled Nanofabrication, http://cden.ucsd.edu

Cycles of Margin Implications [ISQED08]

50% decrease of margin?

Or 100% increase?





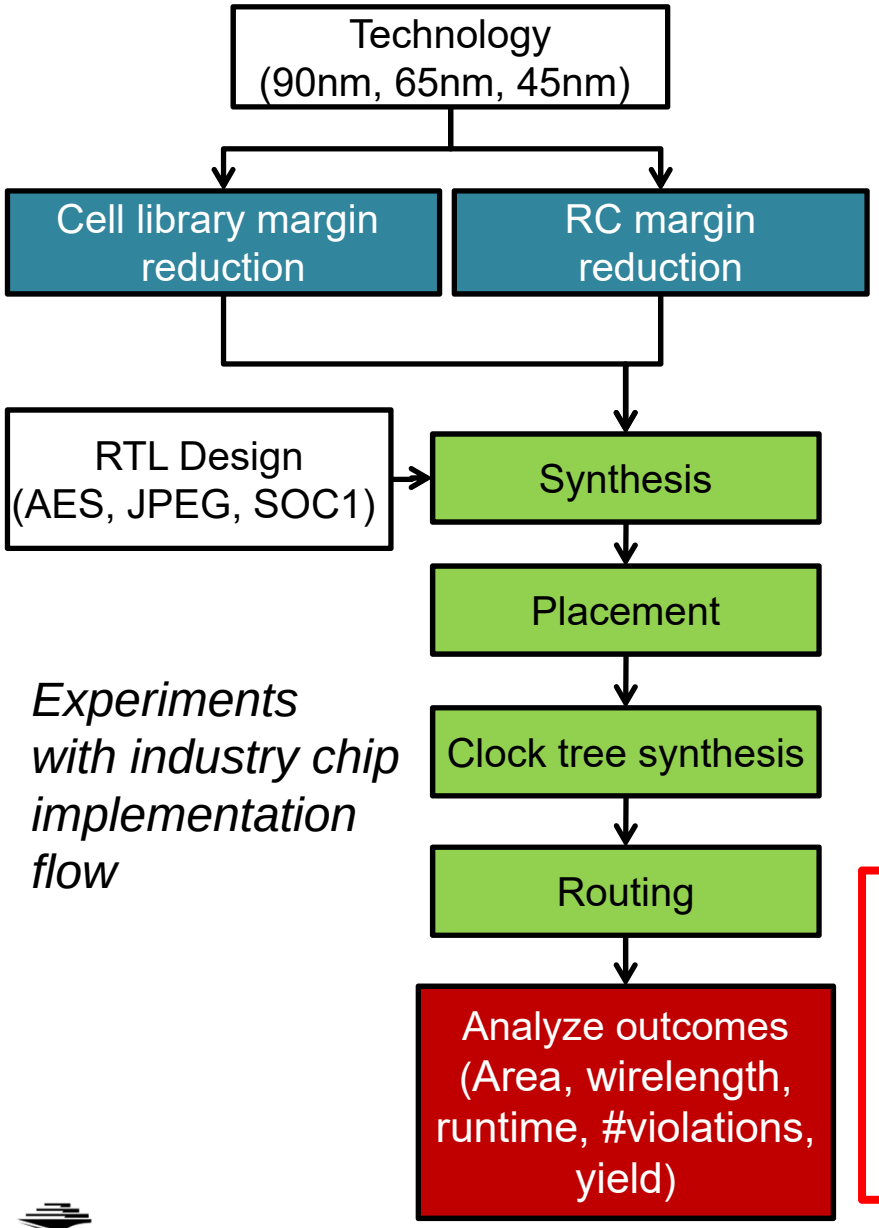
$$Y_r = e^{-Ad}$$

(d: defect density)

$$N_{dies} = \pi\left(\frac{r^2}{A} - \frac{2r}{\sqrt{2A}}\right)$$

(r: wafer radius)

Benefits from Margin Reduction at 45nm



Experiments with industry chip implementation flow

- 40% margin reduction
  - Area: **13%** reduction
  - Dynamic power: **13%** reduction
  - Leakage power: **19%** reduction
  - Wirelength: **12%** reduction
  - Tool runtime (S,P&R): **28%** reduction
  - **#Timing viols.:100% reduction → saves iterations and schedule**
  - #Good dies per wafer (w/o process enhancement): **4%** increase

- **More margin = more cost**
- **Less margin = less cost**
- **Cost reduction ↔ must cure unpredictability of design tools**

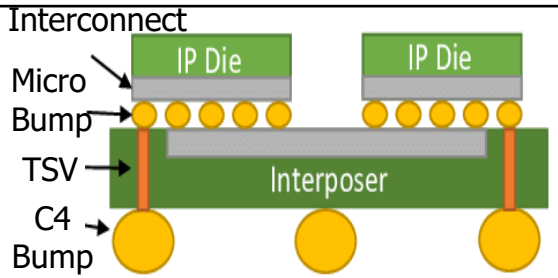
Agenda

- Scaling, Moore's Law and Crises
- Scaling Prospects
- What's Left for the Future?

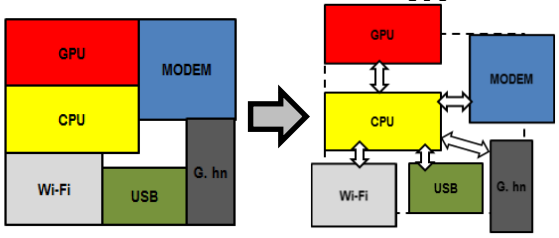
"More Than Moore": 2.5D/3D Integration

Conventional Path

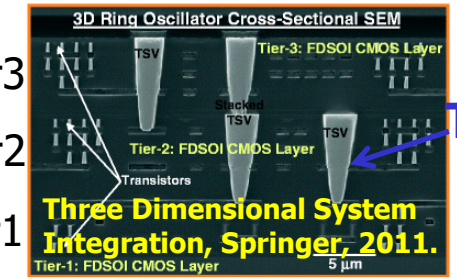
2.5D Interposer-based



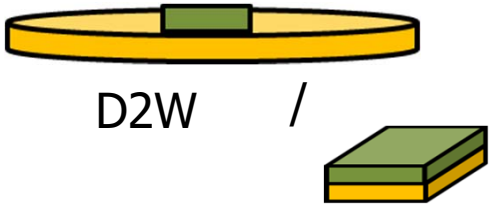
2.5D MOCHI (Marvell)



3D TSV-based

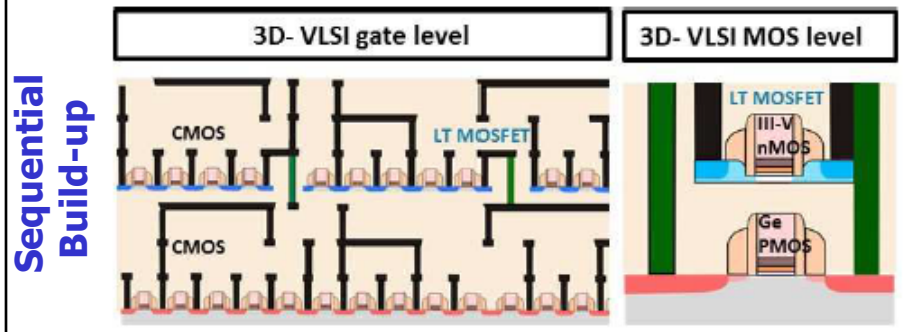


3D Bonding-based



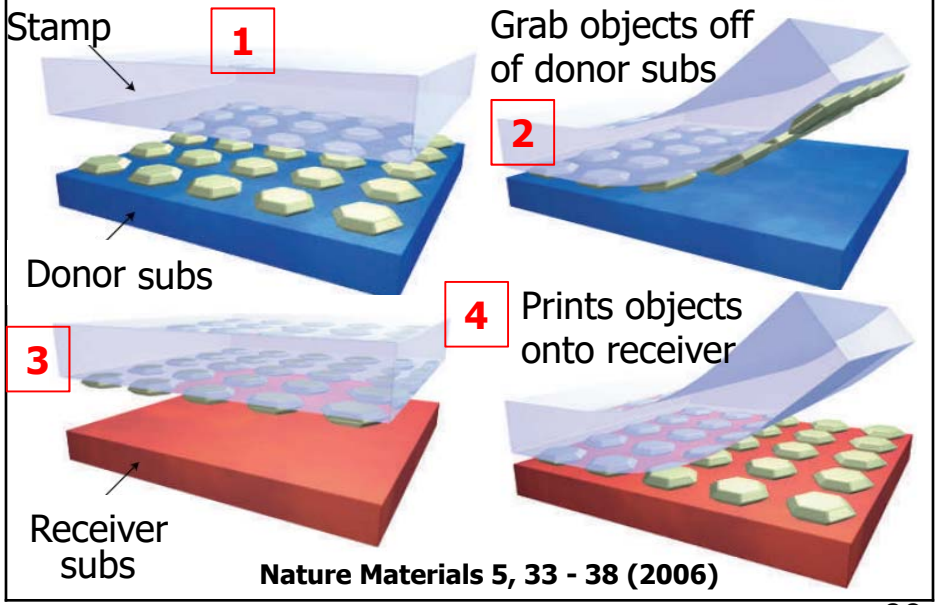
Futures

3D Monolithic Integration



Source: LETI

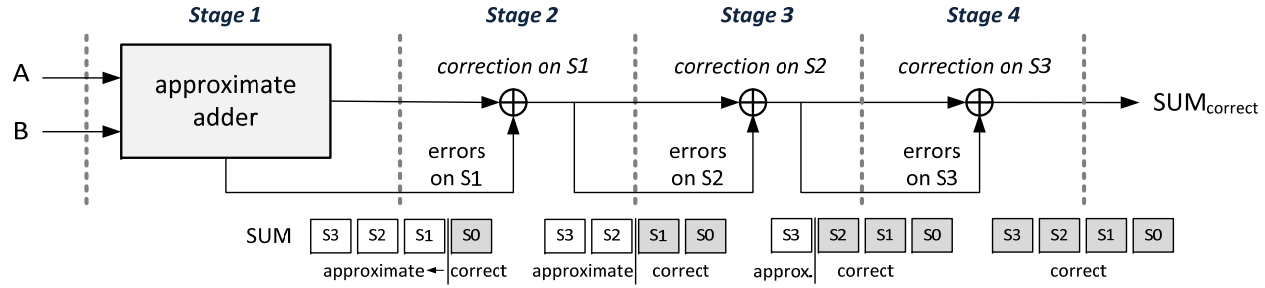
3D Transfer Printing



Nature Materials 5, 33 - 38 (2006)

New ("Rebooting Computing") Paradigms

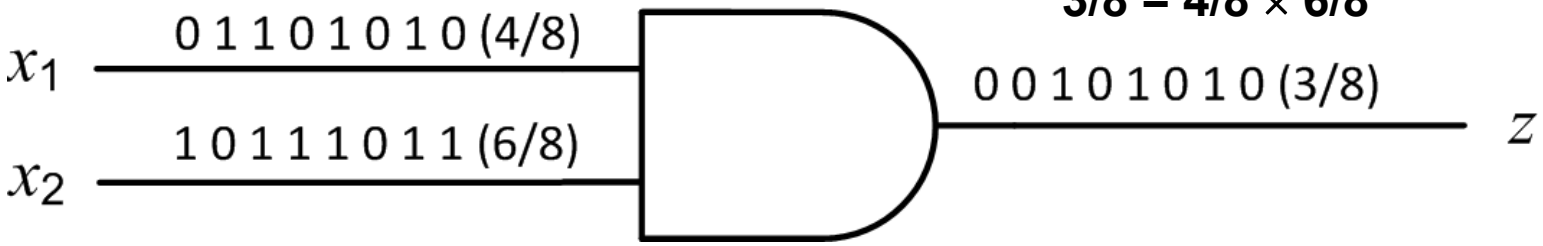
- **Approximate Computing**
  - E.g., cut carry chain in adder to trade off throughput, accuracy



- **Stochastic Computing**
  - Represent numbers by pseudo-random bitstreams
  - Tolerant to delay-induced error compared to parallel number representation

$$Z = X_1 \times X_2$$

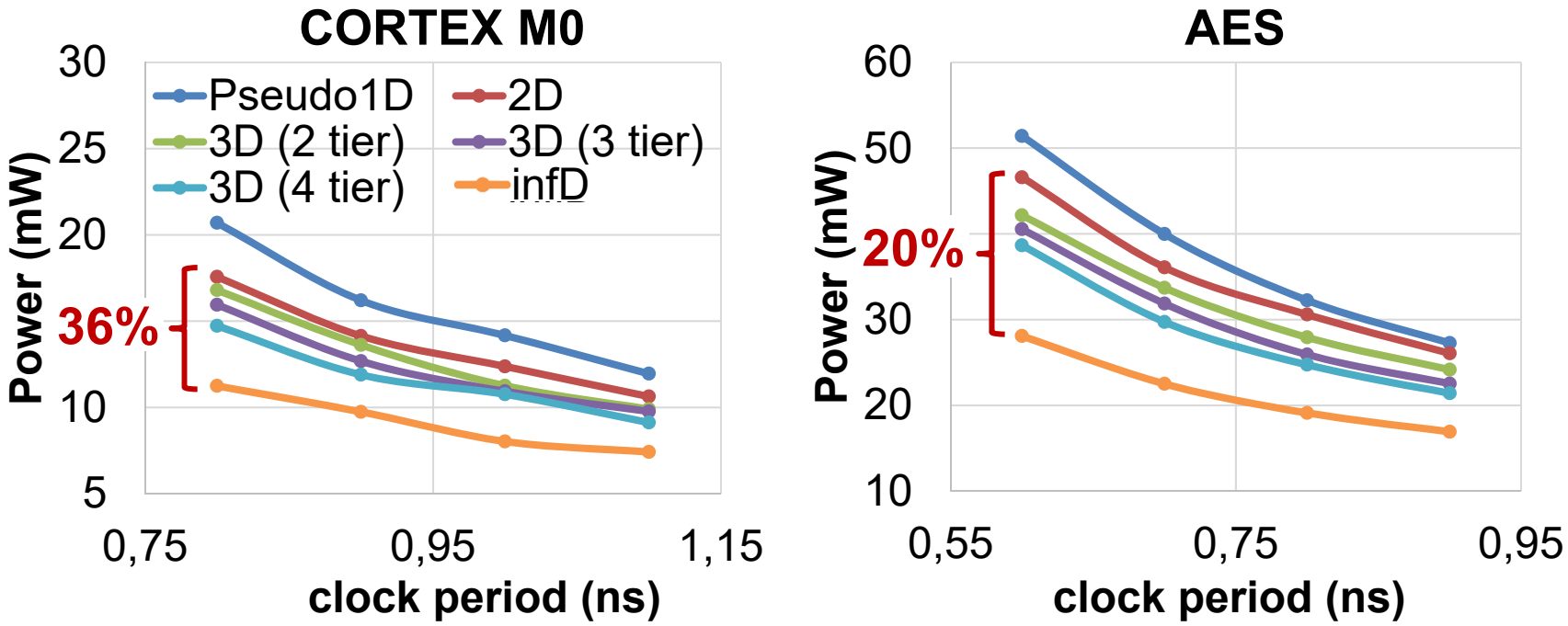
3/8 = 4/8 × 6/8



- **Neuromorphic Computing …**

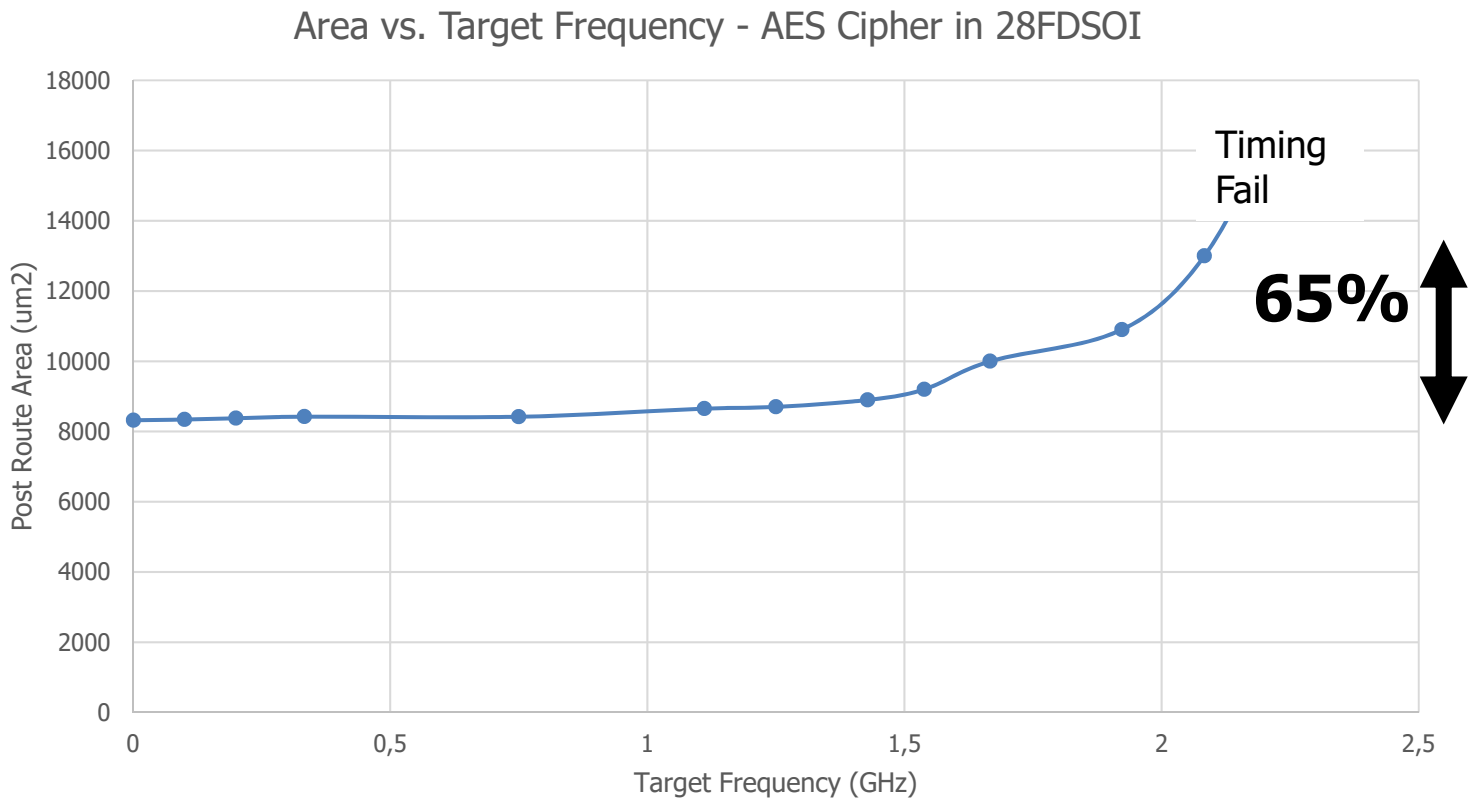
BUT: Even If We Had Infinite Dimensions...

- **Idea: Infinite dimension gives us a bound on 3DIC benefits**
- **Infinite dimension:** netlist optimization with zero wire parasitics
- Gap between infinite dimension and 2D → **maximum power benefit from 3DIC = 36% for CORTEX M0, 20% for AES**

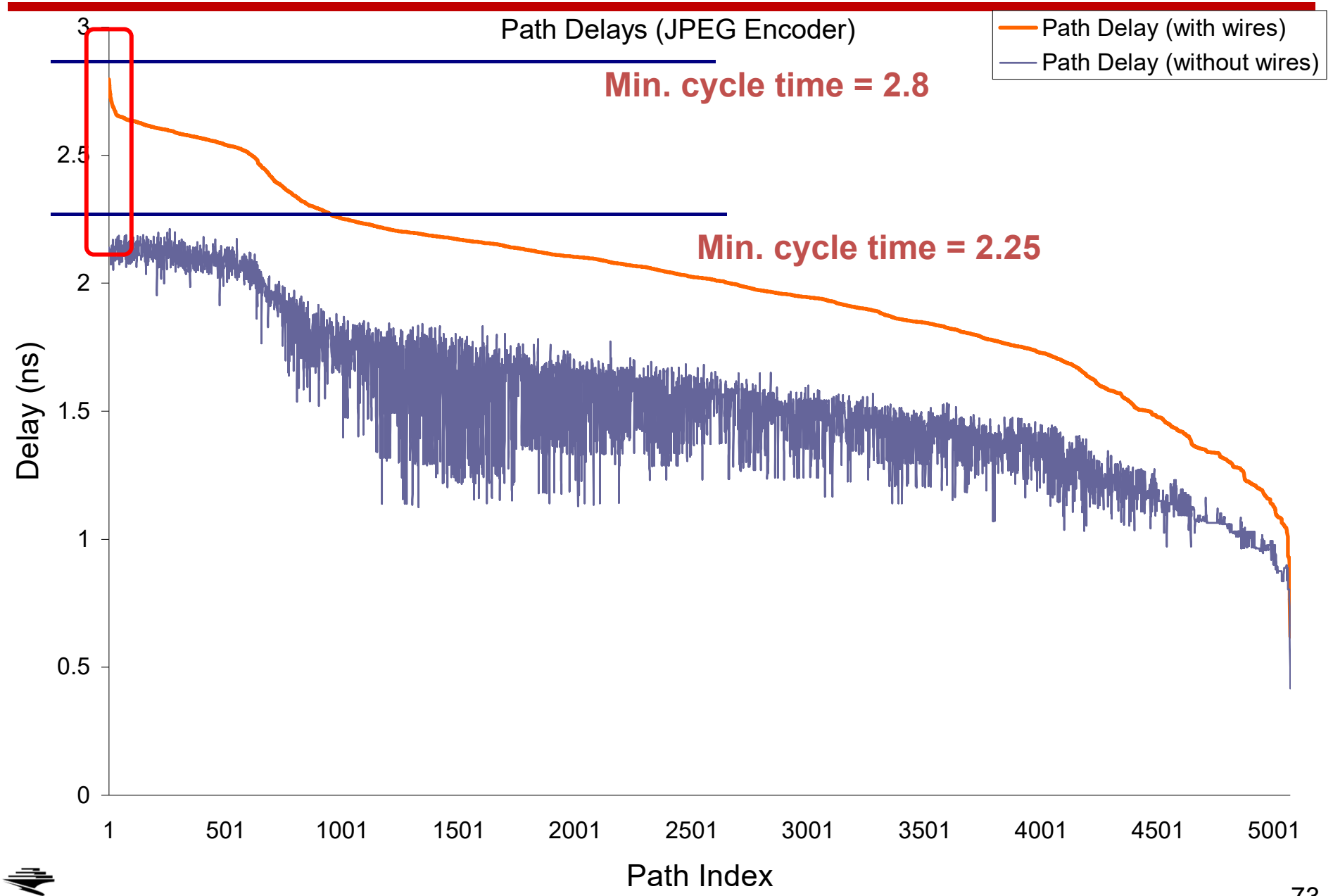


BUT: Even If Frequency Didn't Matter At All…

- Up to ~65% area difference (usually ~30%) between minimum clock period constraint (2.08GHz) and relaxed clock period constraint (28FDSOI, AES)



BUT: Even If Wires Were Perfect (No R, C) ...



Agenda

- Scaling, Moore's Law and Crises
- Scaling Prospects
- What's Left for the Future?
- The Last Semiconductor Scaling Levers

Takeaways

- **Quality, Schedule, Cost are "the last levers for semiconductor scaling"**
  - Accessibility of hardware / semiconductor design
  - Continue semiconductor value trajectory (for a while longer)
- **Foundation #1: machine learning in, around EDA**
  - Pervasive ML → Drive down iterations, margins
  - Cloud-targeted, large-scale optimizations → drive down TAT
- ***Foundation #2: open-source EDA***
  - *Will a "Linux of EDA" be possible this time around?*
- ***Foundation #3: partitioning and cloud EDA***
  - *Also part of schedule reduction*
- **Design Capability Gap is a crisis for the industry**
  - **Need all hands on deck!**

Quality, Schedule, and Cost: Design Technology and the Last Semiconductor Scaling Levers

Andrew B. Kahng
CSE and ECE Departments
UC San Diego

http://vlsicad.ucsd.edu

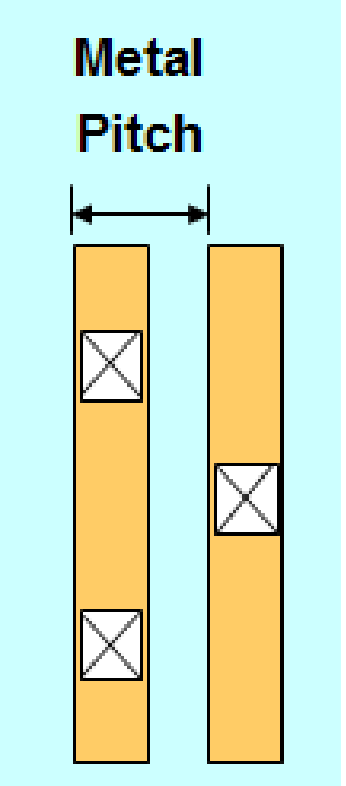
Agenda

- **Scaling, Moore's Law, and Crises**

What is “Scaling”?

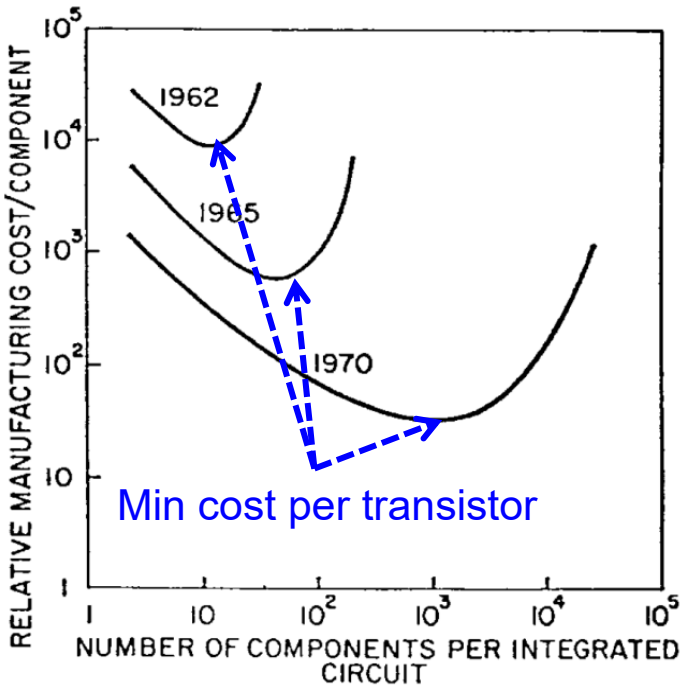
- ITRS = International Technology Roadmap for Semiconductors (http://www.itrs2.net/)
- Key metric of (density) progress: half-pitch (F
- Contacted Poly pitch scales by 0.7×
- Mx pitch scales by 0.7×

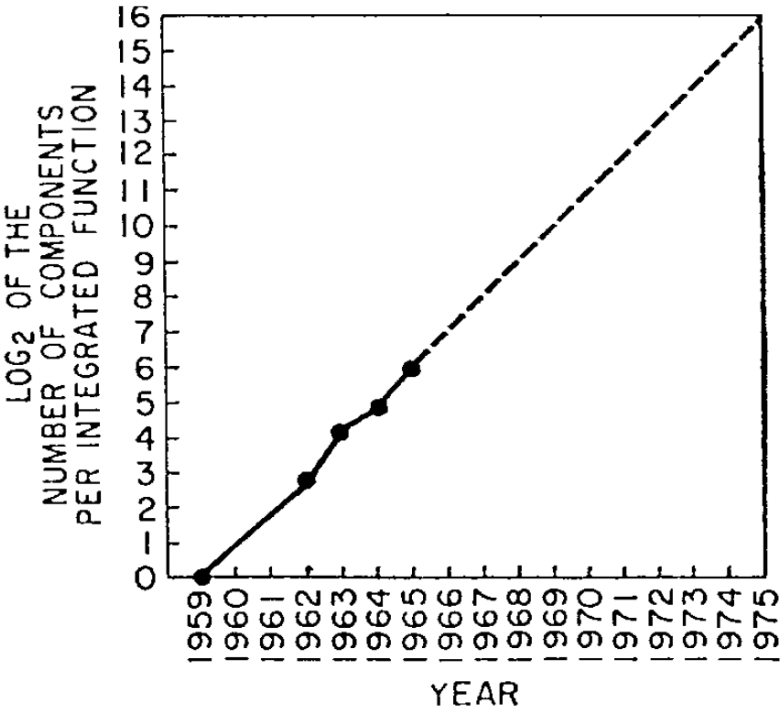
0.7 x 0.7 = 0.49 ⇒ density doubles at each “technology node”



"Moore's Law" = Scaling of Cost and Value

- **Moore, 1965**: "The complexity for minimum component costs has increased at a rate of roughly a factor of two per year"

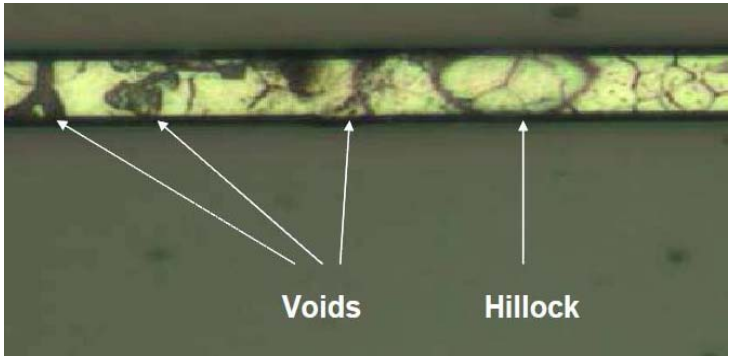




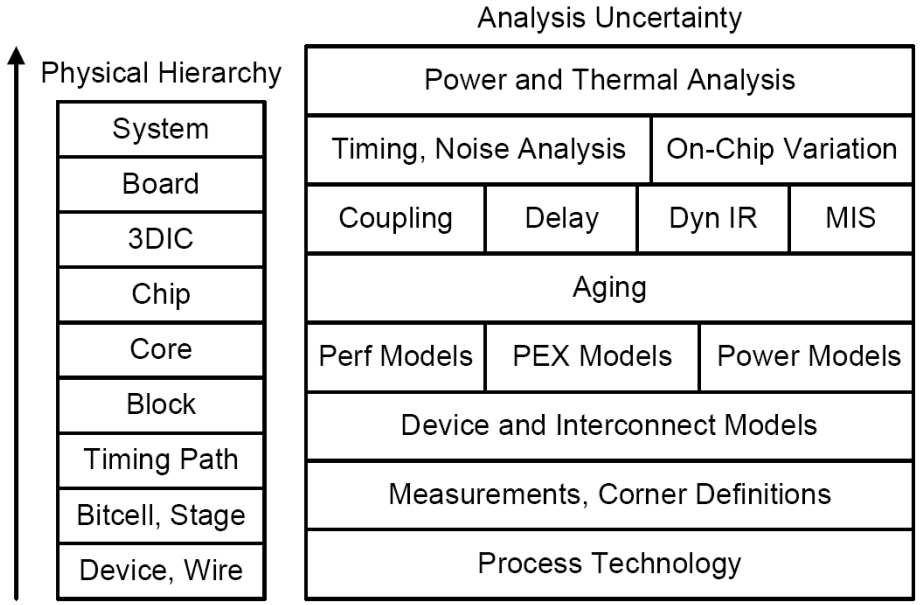
- **Moore's Law is a law of cost reduction** **(1% = 1 week)**
- **Proxy for cost reduction: "scaling of value"**
- **Proxies for value:** "bits", "hertz", "density" (= utility, integration)

Today: Bigger Stacks of Margin ("Corners")

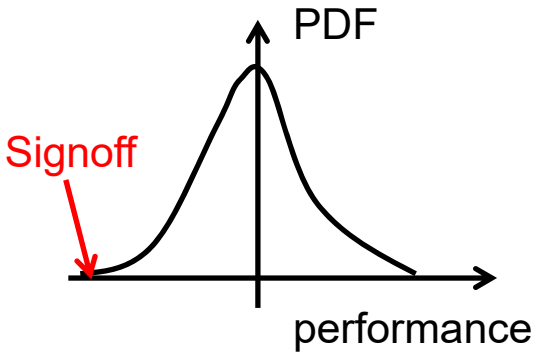
Design margin = stacks of layers of conservatism



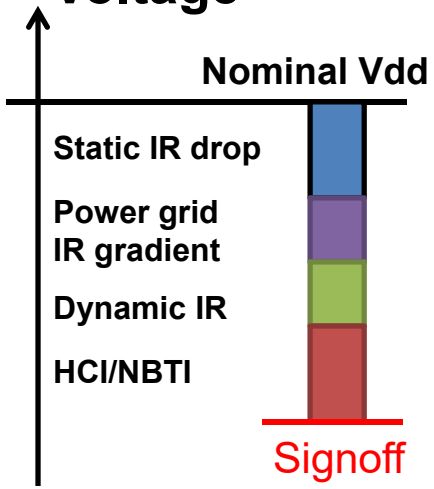
Reliability



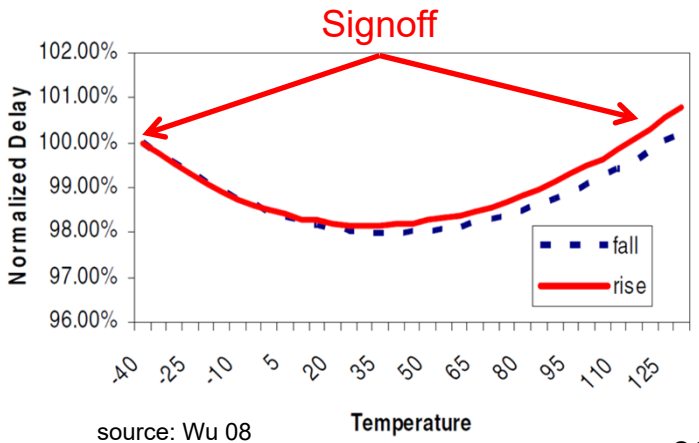
Process



Voltage



Temperature



source: Wu 08

Corner Explosion Worsens

Corners = Process x RCX x Temperature x Voltage x ...

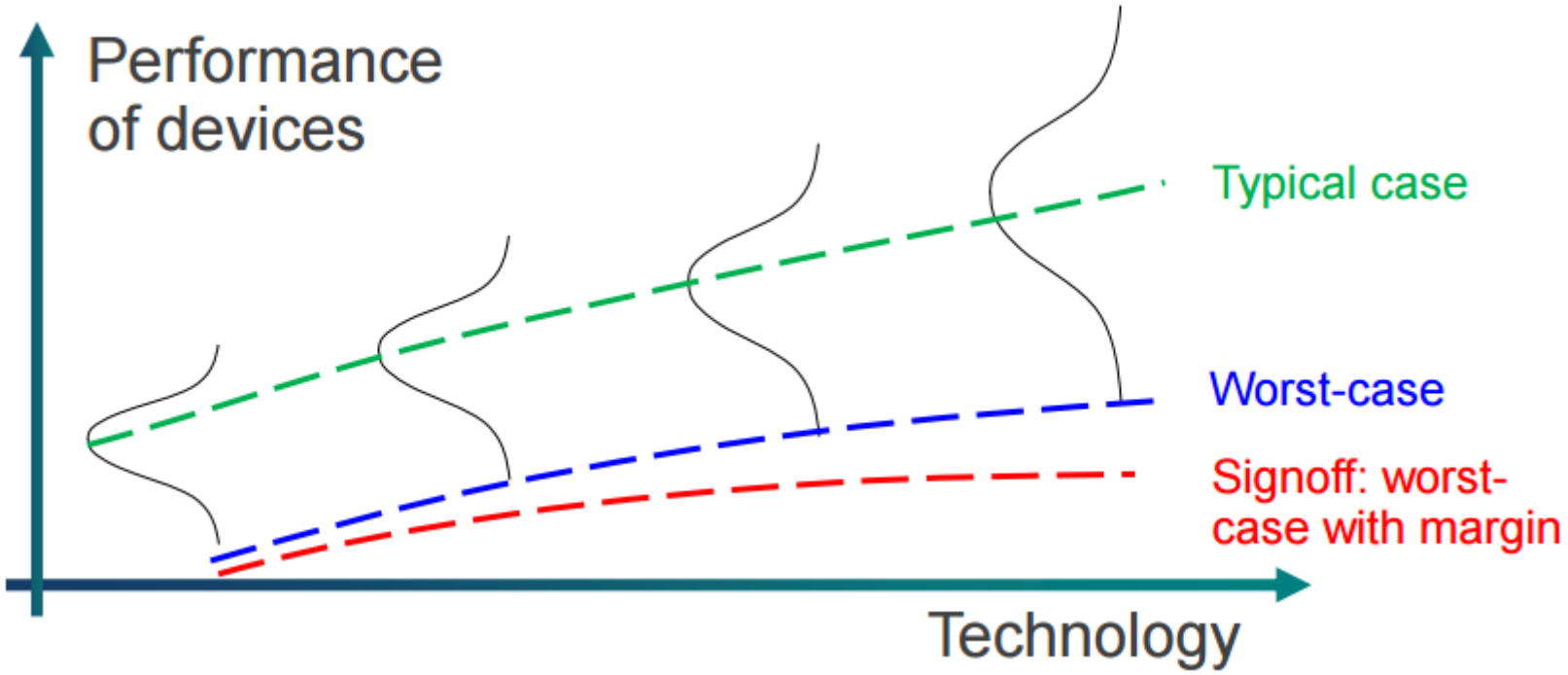
Process	RCX	Temperature	Voltage
FF, FFG, FS, SF, TT, SSG, SS, …	C-worst, Cc-worst, C-best, Cc-best, RC-worst, RC-best, …	-40°C, 0°C, 80°C, 125°C, …	0.7V, 0.8V, 0.9V, 1.0V, 1.1V, …

- Each corner is a new "objective function" and a new set of constraints!

- Lose design turnaround time (TAT) == schedule
  - Non-convergence, "ping-ponging" in timing closure

Consequences

- **Diminishing ROI from next node**
- **Typical**: Moore's Law-ish scaling
- **Worst-case**: Scales, but worse return on investment
- **Signoff with excessive margin: gain is wiped out**

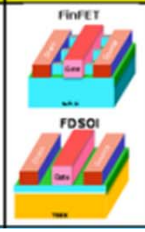
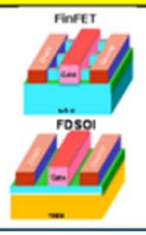
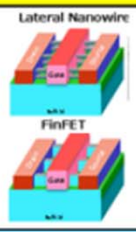
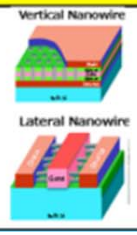
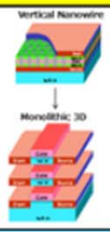
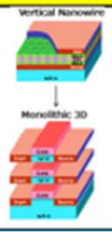
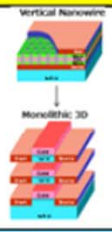


Agenda

- Scaling, Moore's Law and Crises
- Scaling Prospects…
  - Difficult and costly, with limits ahead !

Scaling Will Continue (!)

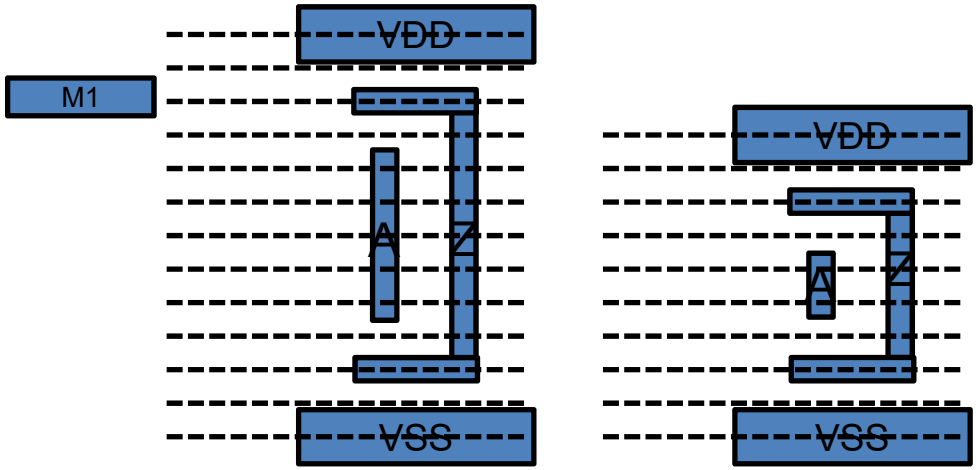
- Lateral scaling in semiconductor manufacturing and device architecture is still predicted to occur
  - Extremely challenging after 5nm/3nm node (i.e., N5/N3)
  - Monolithic 3D will drive scaling afterwards
- Beyond this roadmap, new scaling levers are needed

YEAR OF PRODUCTION	2015	2017	2019	2021	2024	2027	2030
Logic device technology naming	P70M56	P54M36	P42M24	P32M20	P24M12G1	P24M12G2	P24M12G3
Logic industry "Node Range" Labeling (nm)	"16/14"	"11/10"	"8/7"	"6/5"	"4/3"	"3/2.5"	"2/1.5"
Logic device structure options	finFET FDSOI	finFET FDSOI	finFET LGAA	finFET LGAA VGAA	VGAA, M3D	VGAA, M3D	VGAA, M3D
							

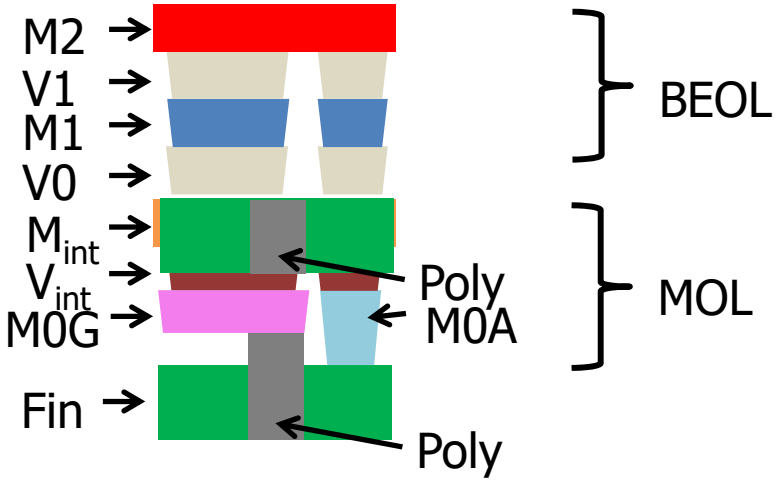
Source: IRDS

Lateral (Area) Scaling: MOL and Tracks (1)

- Old technology node layer stack
  - OD / Poly – V0 – M1 – V1 – M2



Inverter (old) Inverter (old)

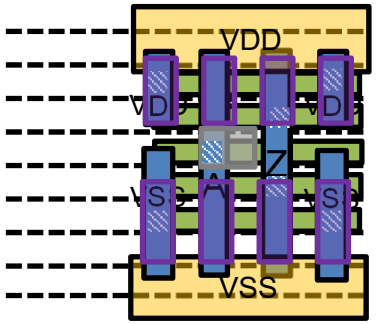


- Advanced node layer stack
  - OD – M0A – VINT – MINT – V0 – M1 – V1 – M2
  - Poly – M0G – VINT – MINT – V0 – M1 – V1 – M2

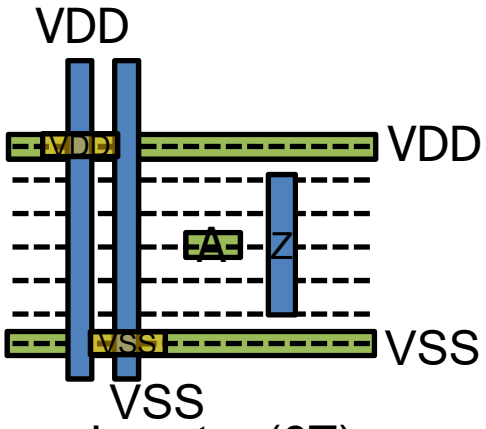
Lateral (Area) Scaling: MOL and Tracks (2)

- N10/N7/N5 technology nodes

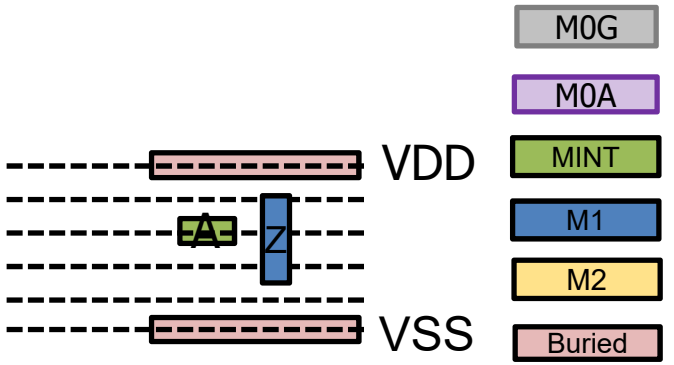
Cells	12T	9T	7.5T	6T	5T/4T/3T
Pins	M1		M1	MINT/M1	
M1	Bidirectional		Unidirectional		
MOL	N/A		Yes: MINT/M0 below M1		
VDD/VSS	M1		M2	M1/MINT	Buried/backside P/G
# M2 routing tracks	~9	~6	5	6	5/4/3



Inverter (7.5T)



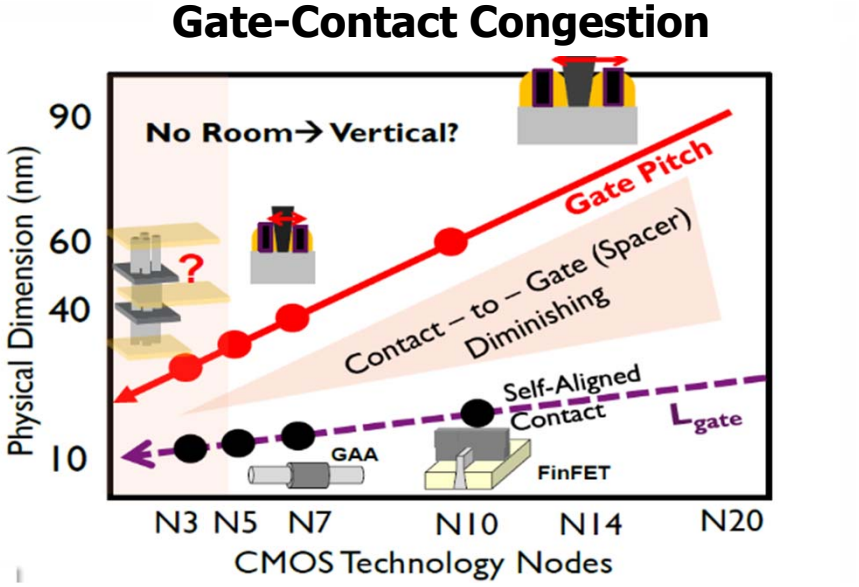
Inverter (6T)



Inverter (5T)

Area Scaling Teardown (CPP x MP)

- 0.5x target area scaling to continue Moore's Law
- Combines Contacted Poly Pitch (CPP) scaling and Metal Pitch (MP) scaling
- → Need new design technology and device technologies



0.5x area scaling = CPP scaling x metal pitch scaling

	2014	2016	2018	2020	2023	2025
Node conventional name	N14	N10	N7	N5	N3	N1.5
Ground rules CPP=x0.78, MP=x0.65						
Contacted poly pitch – [nm]	70	52	42	32	25	25
	LELE	LELE,SADP	SADP	SAQP	SAQP	SAQP
Metal pitch – [nm]	52	36	24	16	10	10
Metal patterning	LELELE, SADP	SAQP, EUV	SAQP, EUV	SAOP, DSA, EUV	SAOP, DSA, EUV	SAOP, DSA, EUV

[source] M. Badaroglu, "More Moore scaling: opportunities and inflection points"

Scaling is Doable, but ...

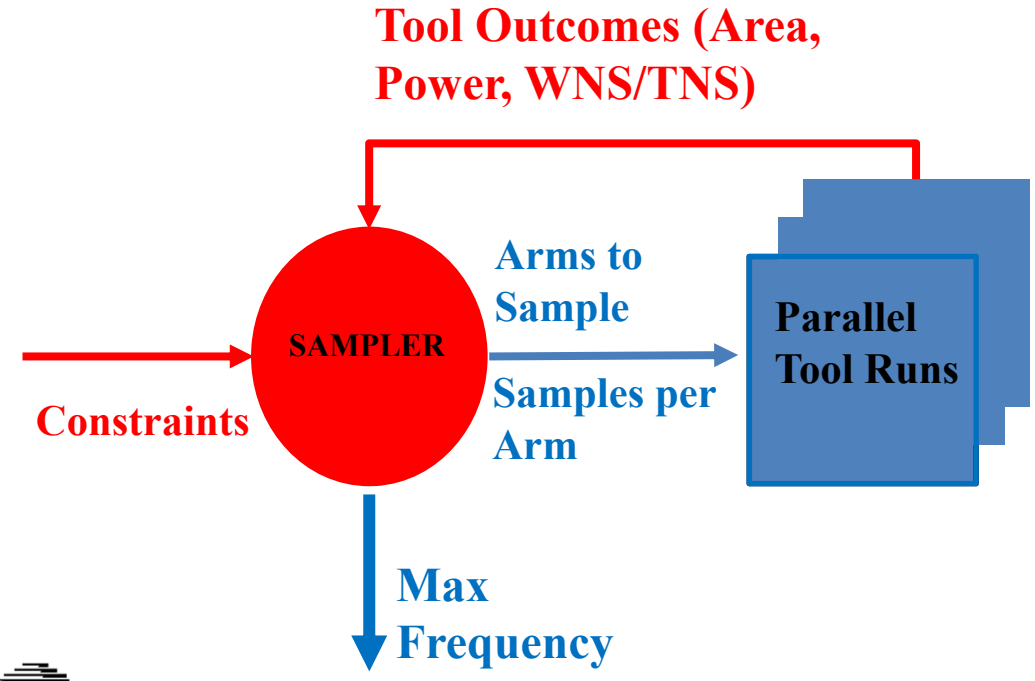
... it's getting tough ☺

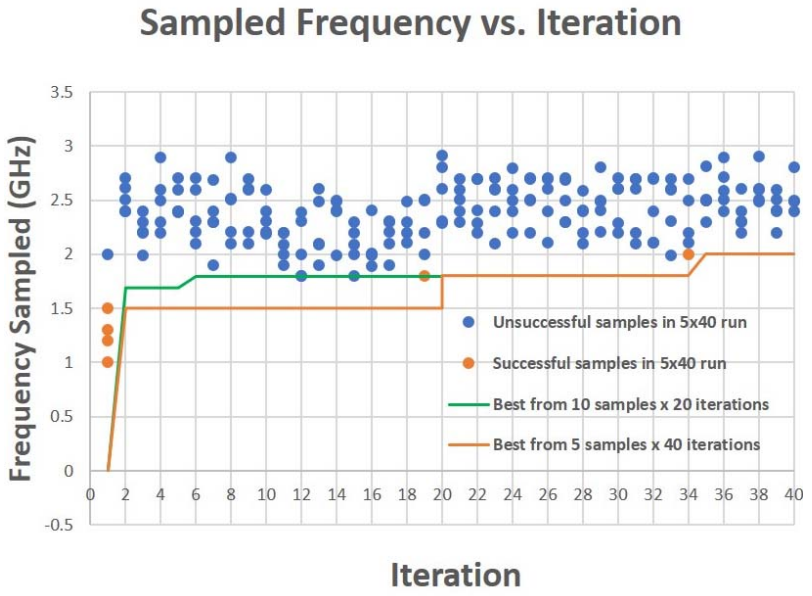
Machine Learning Gives Us Scaling !

- **High-value opportunities in and around EDA**
- **Modeling and Prediction**
  - Predict tool outcome = F(design, constraints, tool config)
    - How to run tool "optimally" for given design and design goals?
    - Avoid "failed runs" → reduce iterations in design flow
    - Dream: one-pass design flow
  - Model analysis errors (crude vs. golden analyses)
    - Reduced guardbands and pessimism → better design quality
- **Optimization (ML models = objective functions!)**
  - Better use of resources (tools, schedule, engineers) + better tools
  - Project-level prediction, adaptive scheduling
- **Today: the major focus for IC industry**
  - **U.S. DARPA IDEA program: automation↑↑, schedule↓↓**
  - **24-hour TAT, "no-human-in-the-loop"**

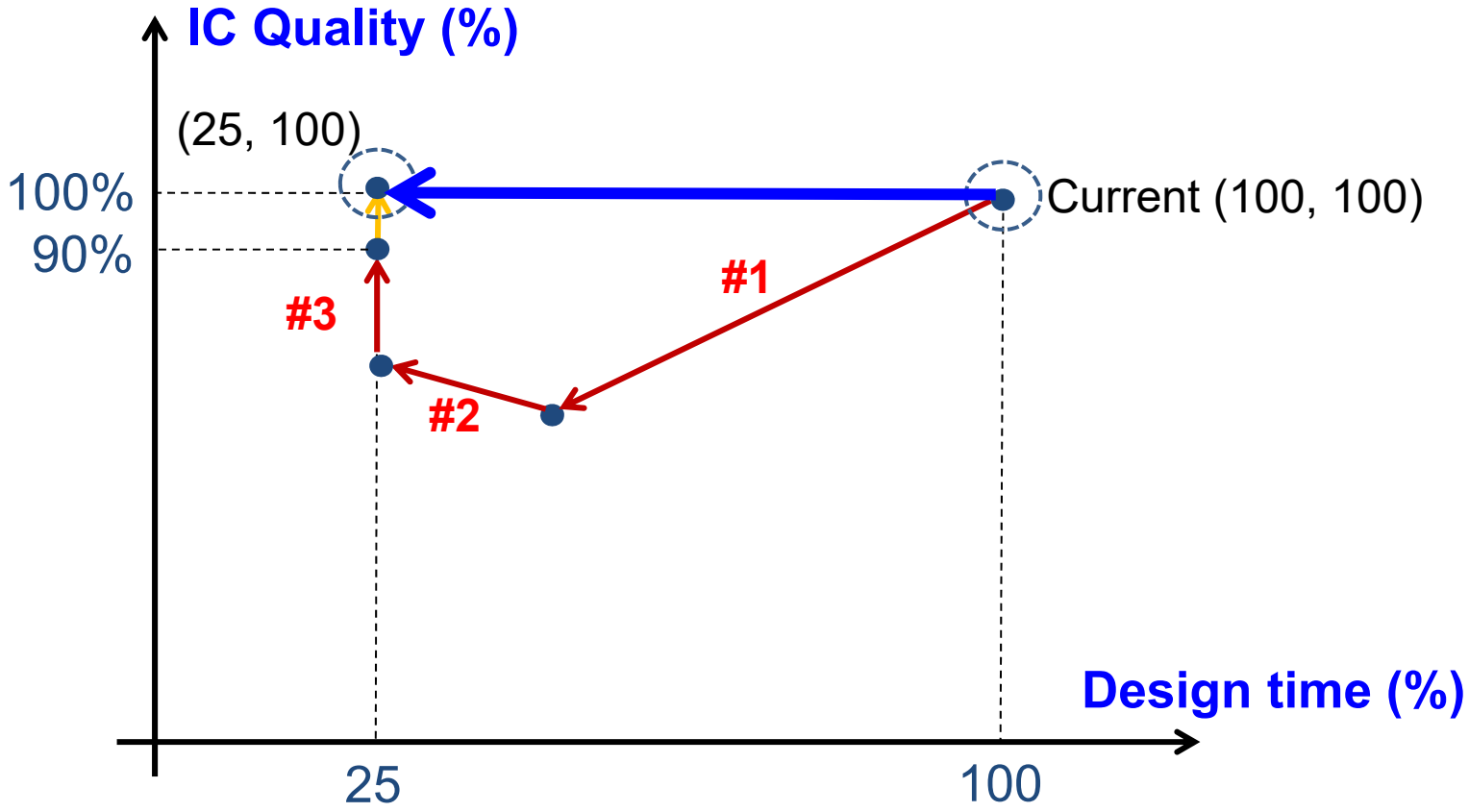
What About … "No Human In The Loop"?

- **Multi Armed Bandit Problem**: Given a slot machine with N arms, maximize total reward obtained using T pulls (iterations)
  - Well-studied in context of Reinforcement Learning
- **IC Design:** "arm" = target frequency; "pull" = run of flow
  - UCSD scripts available upon request





Same Quality in Less Time = Scaling

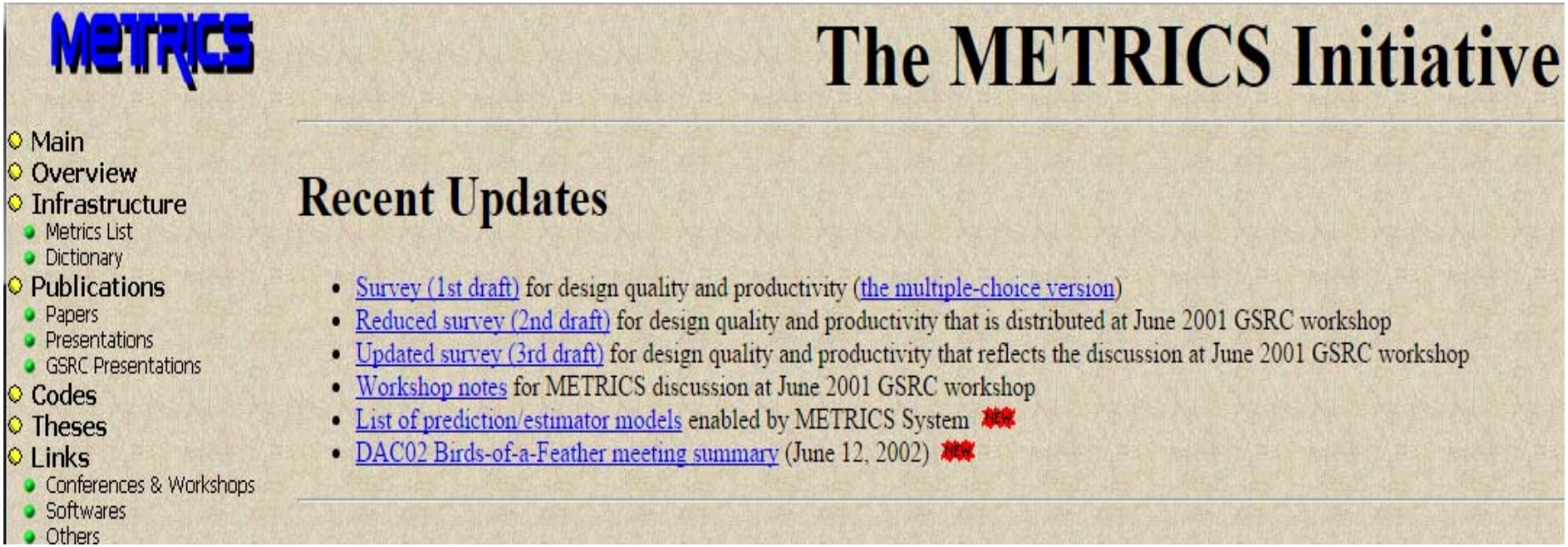


#1. tool/flow models; design-adaptive, learning-based, one-pass flows
#2. analysis correlation, prediction; reduced margins/corners; correct by construction
#3. cloud-based design to recover global optimization; SP&R improvements

Machine Learning (Data + Intelligence) is essential for this

(This is "METRICS" !)

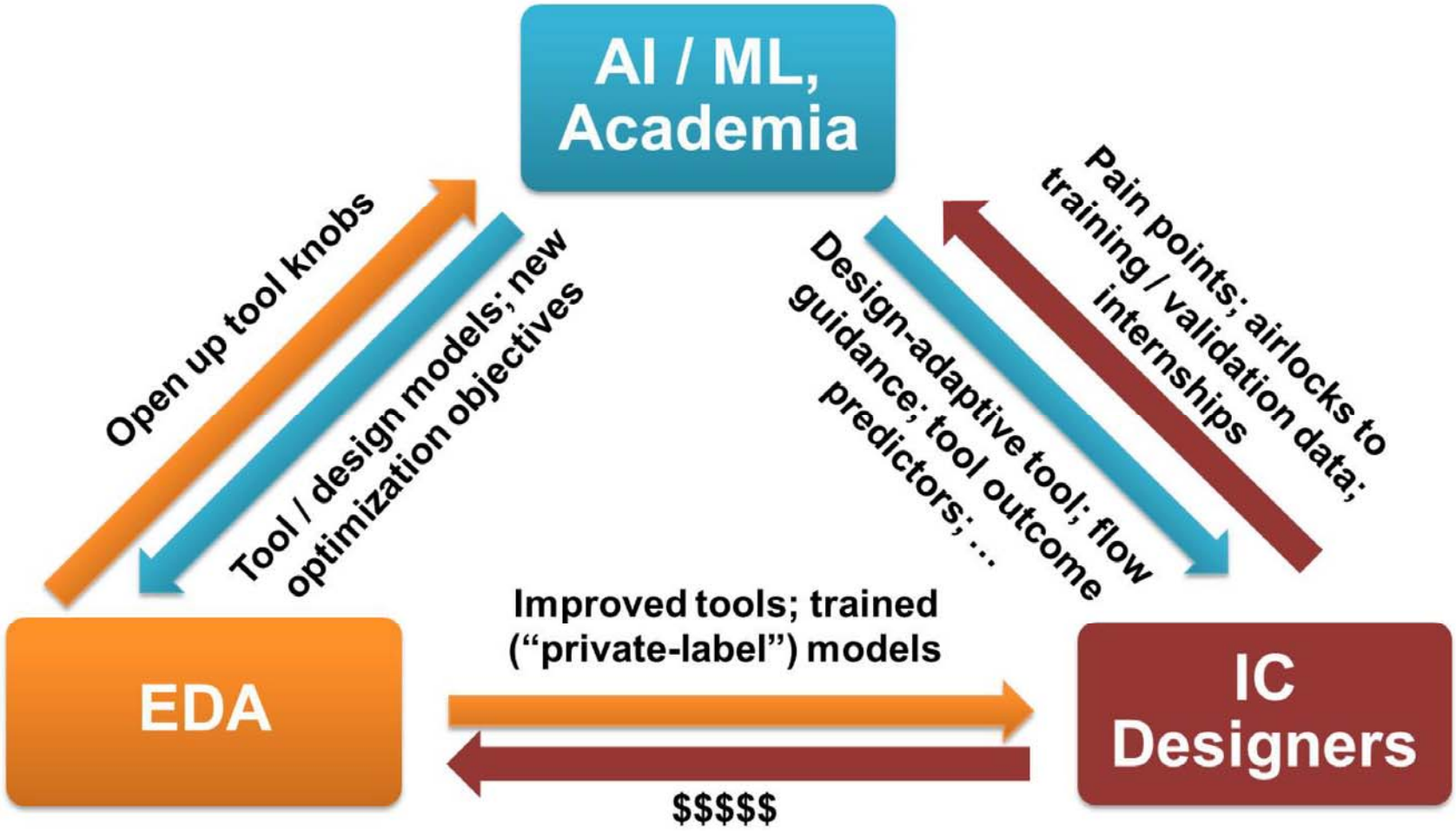
[ISQED01]



- METRICS (1999; ISQED01): "Measure to Improve"
  - Goal #1: Predict outcome
  - Goal #2: Find sweet spot (field of use) of tool, flow
  - Goal #3: Dial in design-specific tool, flow knobs

http://vlsicad.ucsd.edu/GSRC/metrics

A Future Ecosystem



Agenda

- Scaling, Moore's Law and Crises
- Scaling Prospects
- What's Left for the Future?
- The Last Semiconductor Scaling Levers
- Going Forward: Foundation #1 = ML in/around EDA
- Going Forward: Foundation #2

Attacking the Design Capability Gap

- Not enough R&D attention on EDA challenges
  - ~10,000 worldwide EDA, internal CAD, academic research headcount
- Long latency of technology transfer
  - Latest CAD research technologies unavailable to chip designers
  - 5-7 years from ASP-DAC proceedings to production IC design flow
- → Opportunity for another form of “scaling”

Is It Time for "Linux of EDA"?

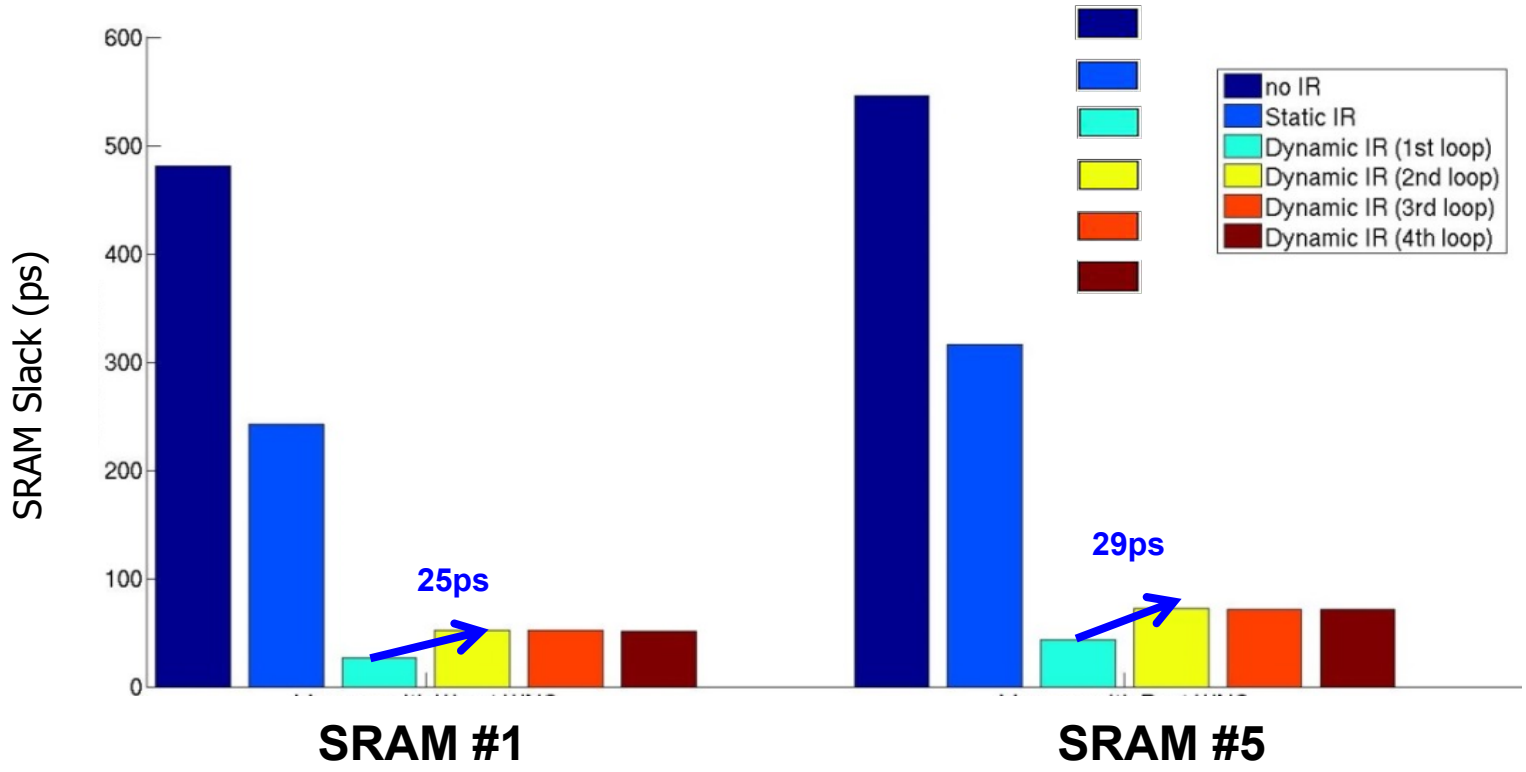
- **Free open-source software (FOSS) has sparked rapid innovation in many fields**
  - **Common standards, platforms avoid wasted energy**
  - **Recent U.S. DARPA "IDEA" program solicitation: IC design that is "no human in the loop" and "24-hour TAT"**
- **Older efforts**
  - **MARCO GSRC Bookshelf**
  - **Berkeley tools (SPICE, MIS/SIS/ABC, …)**
  - **UCLA/UCSD/UM tools (Capo, MLPart, …)**
  - **OpenAccess and OAGears**
- **Many recent efforts worldwide**
  - **OpenTimer, Yosys, RSyn, Ophidian, Open Design Flow, CloudV.io, …**
  - **Will "critical mass" be possible this time around?**

Agenda

- Scaling, Moore's Law and Crises
- Scaling Prospects
- What's Left for the Future?
- The Last Semiconductor Scaling Levers
- Going Forward: Foundation #1 = ML in/around EDA
- Going Forward: Foundation #2 = "Linux of EDA"
- Going Forward: Foundation #3 = partitioning, cloud
- Takeaways

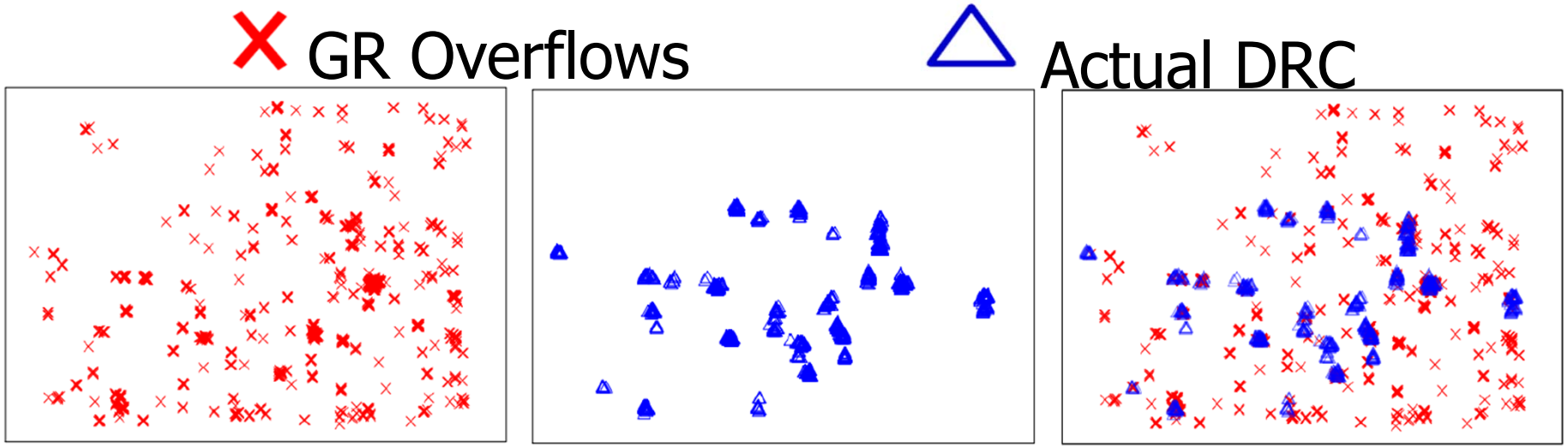
Multiphysics Analysis is Difficult to Predict

- IR drop, thermal, reliability, crosstalk, etc.
- Example: Can we predict “risk map” for embedded memories at floorplan stage ?



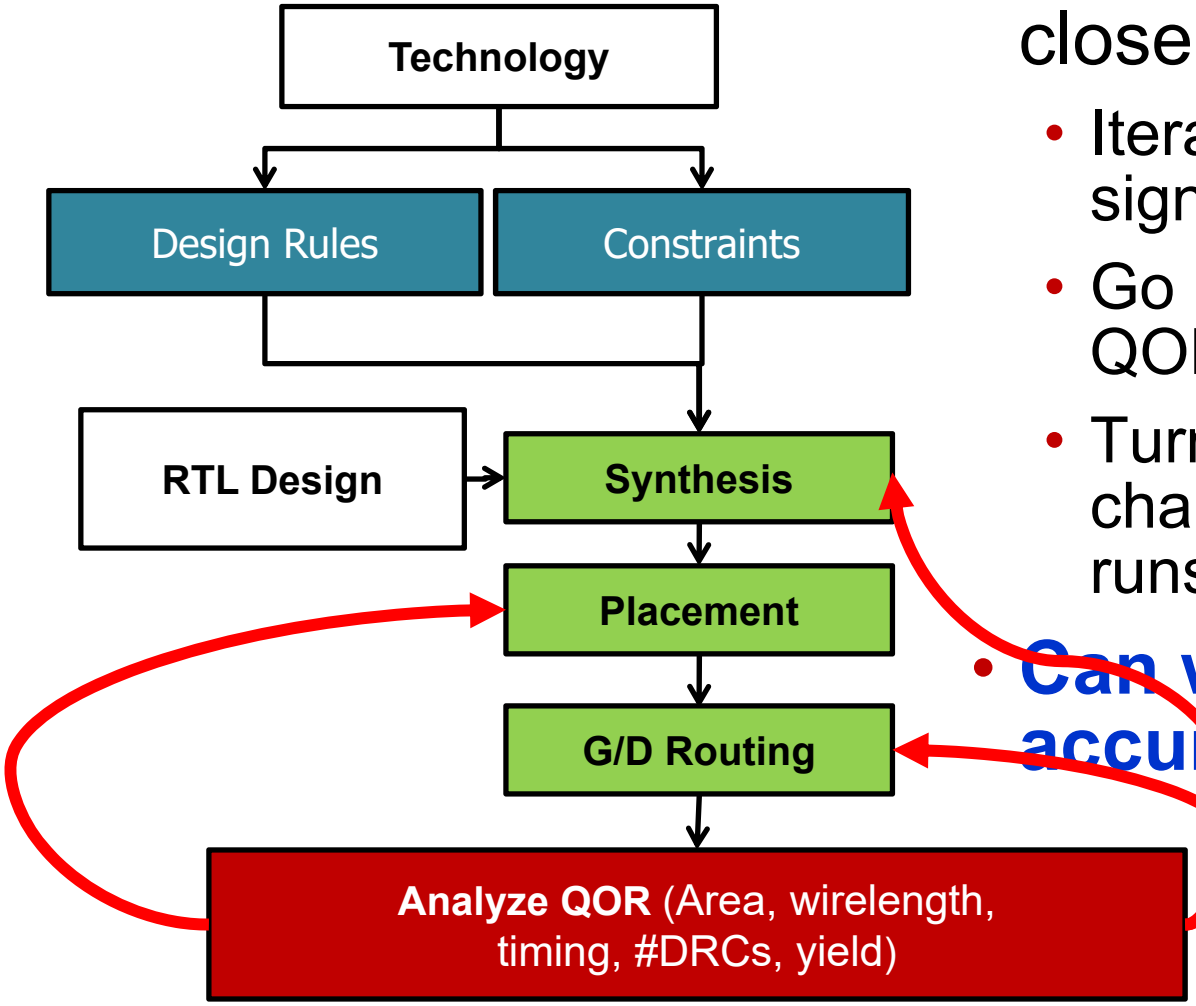
Key Challenge: Global-Detailed Route Correlation

- 7nm P&R: global route (GR) congestion map does not correlate well with post-route (actual) DRC violations
- Many false-positive overflows in GR congestion map
- False-positive → do not correspond to actual DRC violations



GR-based prediction can mislead routability optimizations!!!

If We Know DRC Hotspots before Routing…

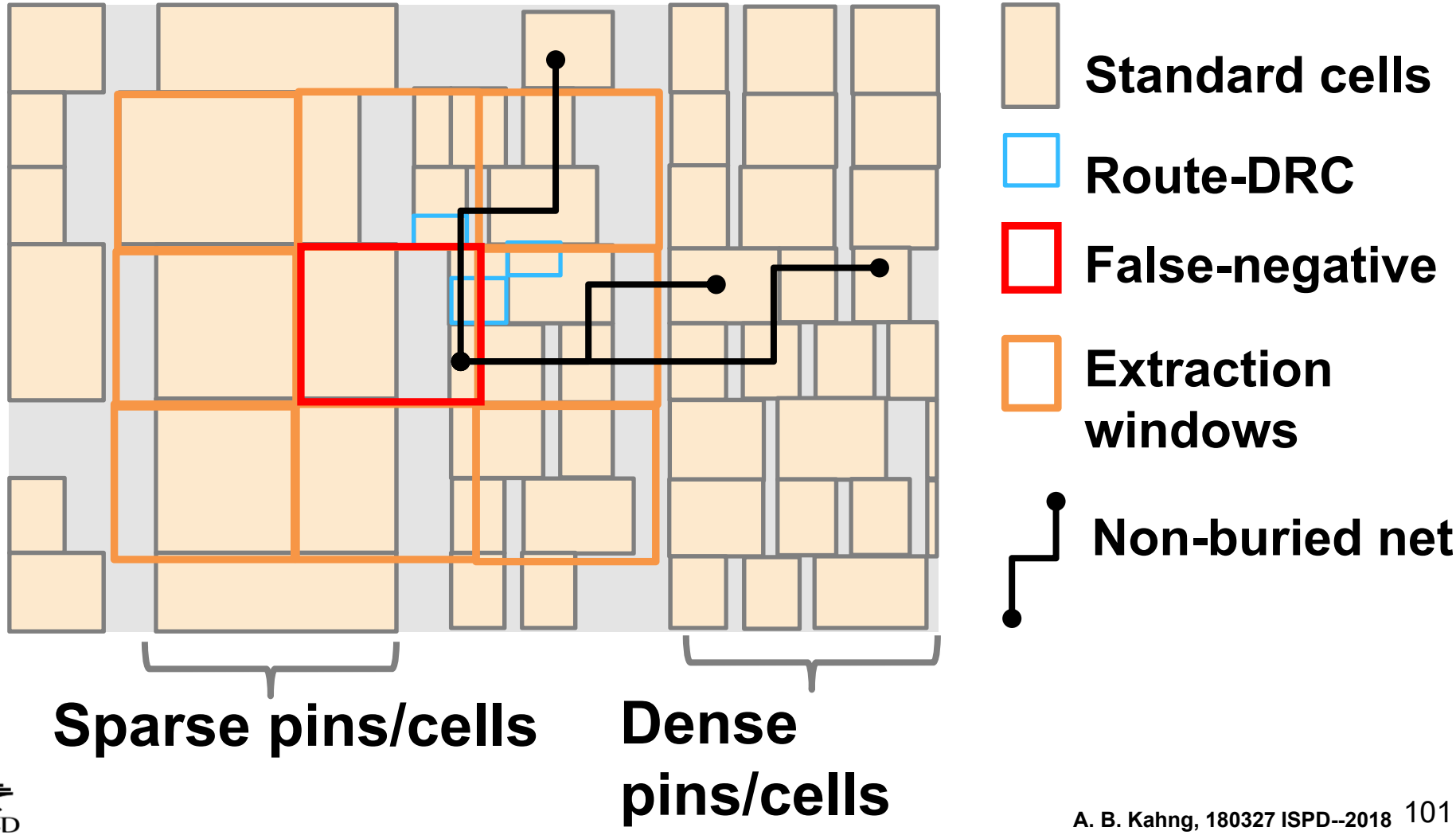


- Conventional way to close designs
  - Iteratively fix design before signoff
  - Go back to placement if QOR is hopeless
  - Turnaround time is VERY challenging (7-day P&R runs…)
- **Can we do better with accurate prediction?**

Iteration with space padding, NDR modifications, density screens ...

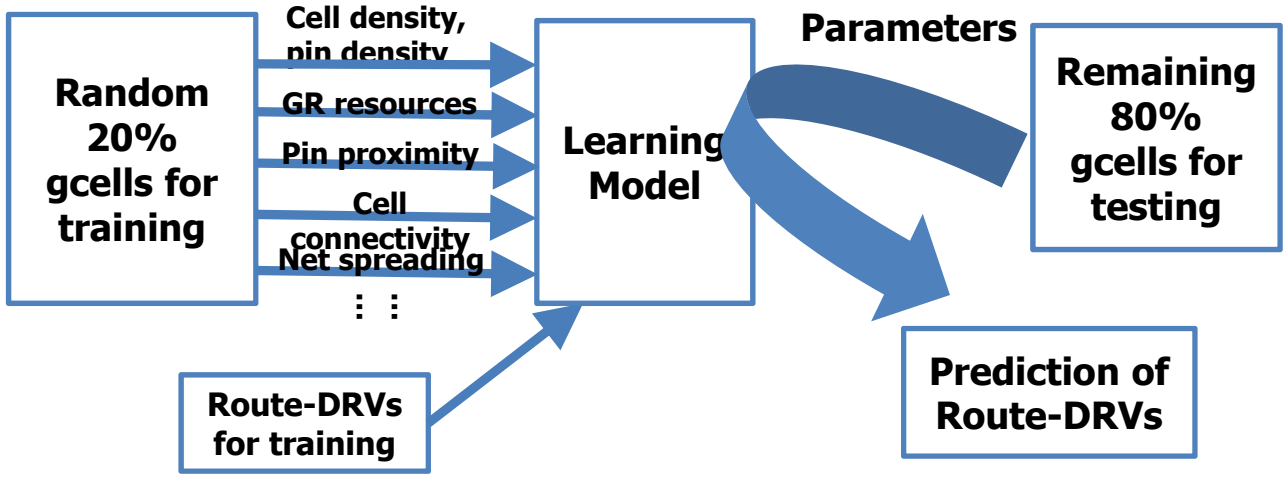
Layout Study

- Initially predict with GR overflows and cell/pin density map
- Red DRC-hotspot likely be rejected due to low cell-pin density
- Larger windows and buried nets metrics to guide prediction



DRV Prediction with Machine Learning

- Predictor is used to guide routability optimization
- SVM with weighting to compensate biased training data



True positive rate = tp / t
False positive rate = tn / n

Initial linear model

Actual / Predicted	W/o DRC	With DRC
W/o DRC	98260	350
With DRC	481	111

True positive rate: **24%**
False positive rate: **0.5%**

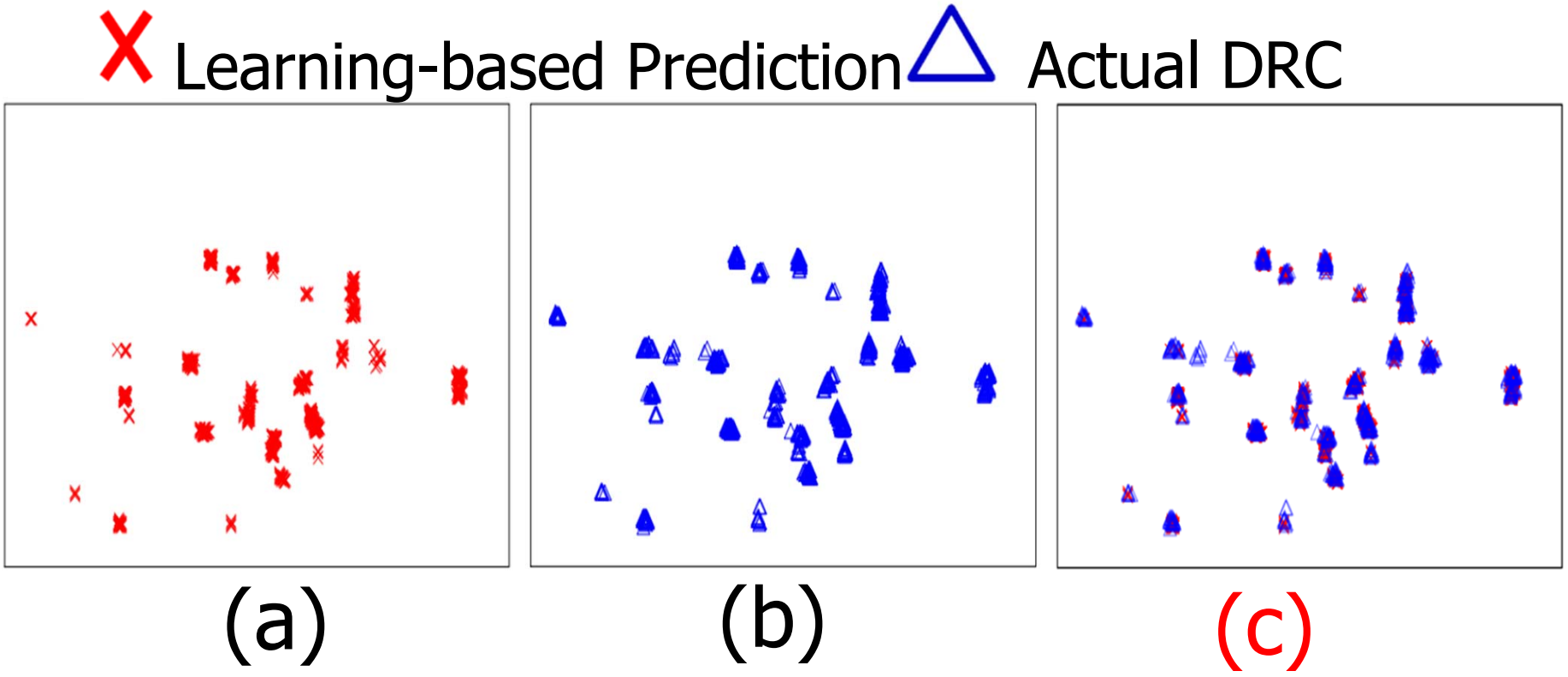
Non-linear SVM model

Actual / Predicted	W/o DRC	With DRC
W/o DRC	98571	117
With DRC	170	344

True positive rate: **74%**
False positive rate: **0.2%**

Improved Learning-Based Predictor

- Captures all true-positive clusters
- Maintains low false-positive rate



Machine Learning Gives Us Scaling !

- **High-value opportunities in and around EDA**
- **Modeling and Prediction**
  - Predict tool outcome = F(design, constraints, tool config)
    - How to run tool "optimally" for given design and design goals?
    - Avoid "failed runs" → reduce iterations in design flow
    - Dream: one-pass design flow
  - Model analysis errors (crude vs. golden analyses)
    - Reduced guardbands and pessimism → better design quality
- **Optimization (ML models = objective functions!)**
  - Better use of resources (tools, schedule, engineers) + better tools
  - Project-level prediction, adaptive scheduling (=separate talk)
- **Today: the major focus for IC industry**
  - **U.S. DARPA IDEA program: automation↑↑, schedule↓↓**

Agenda

- Crises…
- … and a Vision
- Machine Learning

PREDICTION

Agenda

- Scaling, Moore's Law and Crises
- Scaling Prospects
- What's Left for the Future?
- The Last Semiconductor Scaling Levers
- Going Forward: Foundation #1

Savings due to MDP

Errors	Testing (Total = 3442 logs)		
N = 200	Number of runs that need to be stopped	Number of runs stopped correctly out of these	Average number of iterations saved
1 STOP	398	394	18.9644
2 consecutive STOPs	398	391	17.9309
3 consecutive STOPs	398	380	16.9736

Test data = M0 runs

For one run, #iterations saved = 20 – (iteration number where MDP says STOP)

Average #iterations saved = Sum(#iterations saved)/398

In almost every one of these 398 cases, the run starts with a huge number of violations, and the MDP stops it almost immediately. Hence, large avg. #iterations saved

Doomed Runs – Updated Error Criteria

- Prediction is wrong if:
  - DR ends with less than N violations and we predict STOP at 3 consecutive iterations (less stringent)
    (where N is the number of violations which a human designer finds it hard to resolve - usually N ~100-200)
  - DR ends with more than N violations and we predict GO at each iteration (already relaxed, but predictor does not have information about N)
- Training data: 1200 logfiles from PROBE experiments
- Testing data: 3745 logfiles from ARM Cortex M0 floorplan experiments

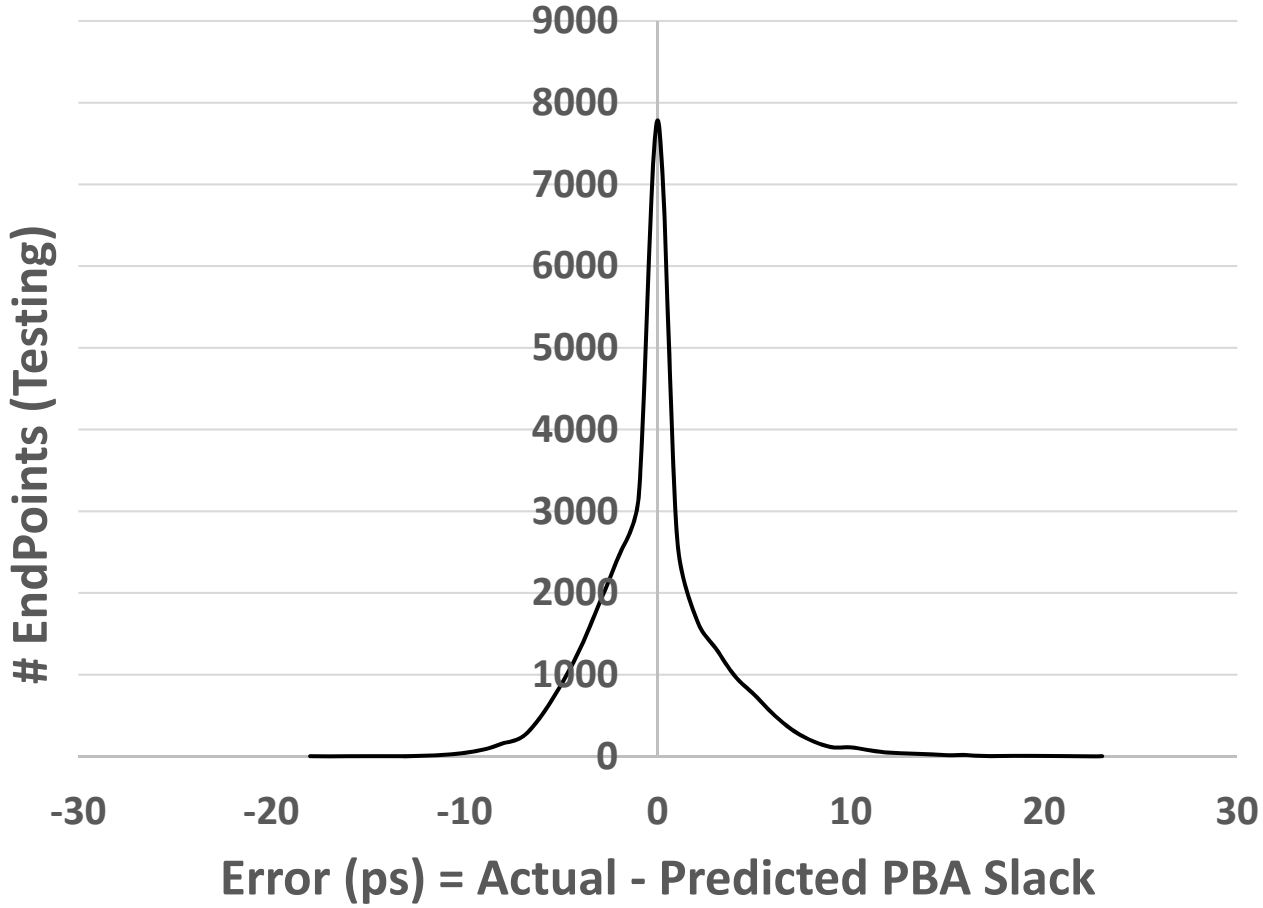
Errors	**Training (Total = 1200)**			**Testing (Total = 3442)**		
N = 200	Total Training Error	#Errors wrongly predicted to STOP (TYPE 1)	#Errors with no STOP (TYPE 2)	Total Training Error	#Errors wrongly predicted to STOP	#Errors with no STOP
1 STOP	29.66%	251	99	35.2%	1317	3
2 consecutive STOPs	10.5%	27	99	8.3%	307	3
3 consecutive STOPs	8.5%	3	99	4.2%	154	3

Machine Learning Gives Us Scaling !

- **High-value opportunities in and around EDA**
- **Modeling and Prediction**
  - Predict tool outcome = F(design, constraints, tool config)
    - How to run tool "optimally" for given design and design goals?
    - Avoid "failed runs" → reduce iterations in design flow
    - Dream: one-pass design flow
  - Model analysis errors (crude vs. golden analyses)
    - Reduced guardbands and pessimism → better design quality
- **Optimization (ML models = objective functions!)**
  - Better use of resources (tools, schedule, engineers) + better tools
  - Project-level prediction, adaptive scheduling (=separate talk)
- **Today: the major focus for IC industry**
  - **U.S. DARPA IDEA program: automation↑↑, schedule↓↓**

Example Early Result

- Early model with MARS (multiple adaptive regression splines): 90% of predicted PBA slacks within 5ps
- Testcase: netcard, 28nm FDSOI



Machine Learning Gives Us Scaling !

- **High-value opportunities in and around EDA**
- **Modeling and Prediction**
  - Predict tool outcome = F(design, constraints, tool config)
    - How to run tool "optimally" for given design and design goals?
    - Avoid "failed runs" → reduce iterations in design flow
    - Dream: one-pass design flow
  - Model analysis errors (crude vs. golden analyses)
    - Reduced guardbands and pessimism → better design quality
- **Optimization (ML models = objective functions!)**
  - Better use of resources (tools, schedule, engineers) + better tools
  - Project-level prediction, adaptive scheduling
- **Today: the major focus for IC industry**
  - **U.S. DARPA IDEA program: automation↑↑, schedule↓↓**

Takeaways

- **Quality, Schedule, Cost are "the last levers for semiconductor scaling"**
  - Accessibility of hardware / semiconductor design
  - Continue semiconductor value trajectory (for a while longer)
- **Foundation #1: machine learning in, around EDA**
  - Pervasive ML → Drive down iterations, margins
  - Cloud-targeted, large-scale optimizations → drive down TAT
- **Foundation #2: open-source EDA**
  - Will a "Linux of EDA" be possible this time around?
- **Foundation #3: partitioning and cloud EDA**
  - Also part of schedule reduction
- **Design Capability Gap is a crisis for the industry**
  - **Need all hands on deck!**

Conclusions and Futures (2)

- **ML+EDA: challenges of technology**
  - "Small data" problem alongside "big data" problem
  - Huge implementation space, difficult parameter identification
  - Complicated by tool versions, design versions, technology changes **(pictures of cats and trees don't change every year)**
  - Possibly helpful: EDA folks know what's in their tools!
- **ML in EDA: industry challenges**
  - EDA {doesn't like to, doesn't know how to} model itself
  - Dependence on customers and customer data to understand what is needed
  - **Open: Will customers or EDA vendors (or foundries) drive ML into design enablements and production flows?**
- **METRICS** … revisited? *(measure, record, model, predict, improve)*