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” \( \rightarrow \) can we predict outcome?
- Many chicken-egg loops \( \rightarrow \) 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” \( \rightarrow \) 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 $\rightarrow$ heuristics piled on heuristics
- “Noise” or “Chaos” when EDA tools “try hard”
- Unpredictability $\rightarrow$ added margin and schedule

14nm PULPino: $\Delta$area = 6% from $\Delta$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: $\Delta$area = 6% from $\Delta$freq = 10MHz!
Today’s SOC Design

- # Partitions ↓
- Design Flexibility ↑
- Predictability ↓
- # Iterations ↑
- Turnaround Time ↑
- Margins ↑
- Achieved Design Quality ↓
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

Machine Learning will be a key piece of this ...

✓ Quality
✓ Schedule
✓ Cost
Agenda

• Crises…
• … and a Vision
• Machine Learning in PD
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 Overflows](image1)

![Actual DRVs](image2)

**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…

A. B. Kahng, 180327 ISPD--2018
**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

**Standard Flow**
- Placed & optimized netlist
  - Global route
  - Track assignment
  - Detailed route

**New ML-Based Flow**
- Placed & optimized netlist
  - Predictor-guided cell spreader
  - Global route
  - Track assignment
  - Detailed route

Parameter collection (from placement and GR)

- Pre-stored DRC predictor model
  - DRC Prediction

- **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

Local move

Analytical models
Routing: FLUTE, STST
Cell delay: Liberty LUTs
Wire delay: Elmore, D2M

Delta delays

Learning-based model

- 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?

<table>
<thead>
<tr>
<th>Errors</th>
<th>Training (Total = 1200)</th>
<th>Testing (Total = 3442)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N = 200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total Training Error</td>
<td></td>
</tr>
<tr>
<td>1 STOP</td>
<td>29.66%</td>
<td></td>
</tr>
<tr>
<td>2 consecutive STOPs</td>
<td>10.5%</td>
<td></td>
</tr>
<tr>
<td>3 consecutive STOPs</td>
<td>8.5%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#TYPE 1 Errors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(wrong STOP prediction)</td>
<td></td>
</tr>
<tr>
<td>1 STOP</td>
<td>251</td>
<td></td>
</tr>
<tr>
<td>2 consecutive STOPs</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>3 consecutive STOPs</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#TYPE 2 Errors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(no STOP)</td>
<td></td>
</tr>
<tr>
<td>1 STOP</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>2 consecutive STOPs</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>3 consecutive STOPs</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total Training Error</td>
<td></td>
</tr>
<tr>
<td>1 STOP</td>
<td>35.2%</td>
<td></td>
</tr>
<tr>
<td>2 consecutive STOPs</td>
<td>8.3%</td>
<td></td>
</tr>
<tr>
<td>3 consecutive STOPs</td>
<td>4.2%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#TYPE 1 Errors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(wrong STOP prediction)</td>
<td></td>
</tr>
<tr>
<td>1 STOP</td>
<td>1317</td>
<td></td>
</tr>
<tr>
<td>2 consecutive STOPs</td>
<td>307</td>
<td></td>
</tr>
<tr>
<td>3 consecutive STOPs</td>
<td>154</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#TYPE 2 Errors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(no STOP)</td>
<td></td>
</tr>
<tr>
<td>1 STOP</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2 consecutive STOPs</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3 consecutive STOPs</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Agenda

• Crises…
• … and a Vision
• Machine Learning in PD
• Modeling and Prediction
• Analysis Correlation
ML Shifts the Accuracy-Cost Tradeoff Curve!

Accuracy

100%

(100-x)%

Cost / Runtime

+ML

+ML

+ML
Example 4: ML-based Timer Correlation

DATE-2014 (+ SLIP-2015)

Train

Validate

Test

Artificial Circuits

Real Designs

Outliers (data points)

New Designs

MODELS
(Path slack, setup time, stage, cell, wire delays)

If error > threshold

ONE-TIME

INCREMENTAL

BEFORE

AFTER

ML Modeling

123 ps

31 ps

~4× reduction

T_2 Path Slack (ns)

T_1 Path Slack (ns)

T_2 Path Slack (ns)

T_1 Path Slack (ns)
“SI for Free” with Machine Learning

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

Timing Reports in SI Mode
Timing Reports in Non-SI Mode
Create Training, Validation and Testing Sets
ANN (2 Hidden Layers, 5-Fold Cross-Validation)
SVM (RBF Kernel, 5-Fold Cross-Validation)
HSM (Weighted Predictions from ANN and SVM)

Before After

Actual Path Delay (ps)

Non-SI Path Slack (ns)

ML Modeling

Worst absolute error = 8.2ps
Average absolute error = 1.7ps
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?
  - \(\rightarrow\) Better optimization in P&R&Opt, less expensive STA

![Diagram of GBA and PBA modes](image)

![Graph showing PBA - GBA Slack Gain](image)
## Costs of GBA vs. PBA Pessimism

<table>
<thead>
<tr>
<th>GBA Actual Slack</th>
<th>PBA Actual Slack</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSITIVE</td>
<td>POSITIVE</td>
<td>Power recovery can’t exploit usable slack</td>
</tr>
<tr>
<td>NEGATIVE</td>
<td>POSITIVE</td>
<td>Schedule, Area, Power wasted fixing false timing violations</td>
</tr>
<tr>
<td>NEGATIVE</td>
<td>NEGATIVE</td>
<td>Schedule, Area, Power waste from over-fixing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PBA Actual Slack</th>
<th>PBA Predicted Slack (Model)</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>LOW</td>
<td>Power recovery can’t exploit all of usable slack</td>
</tr>
<tr>
<td>LOW</td>
<td>HIGH</td>
<td>Masking of real violations</td>
</tr>
</tbody>
</table>
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
**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

<table>
<thead>
<tr>
<th>Model Training</th>
<th>C&lt;sub&gt;1&lt;/sub&gt;</th>
<th>C&lt;sub&gt;2&lt;/sub&gt;</th>
<th>C&lt;sub&gt;3&lt;/sub&gt;</th>
<th>...</th>
<th>C&lt;sub&gt;n-2&lt;/sub&gt;</th>
<th>C&lt;sub&gt;n-1&lt;/sub&gt;</th>
<th>C&lt;sub&gt;n&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.434008</td>
<td>0.733149</td>
<td>0.234008</td>
<td>...</td>
<td>0.700667</td>
<td>0.556834</td>
<td>0.575091</td>
</tr>
<tr>
<td></td>
<td>0.373264</td>
<td>0.718715</td>
<td>0.273264</td>
<td>...</td>
<td>0.685265</td>
<td>0.551267</td>
<td>0.528366</td>
</tr>
<tr>
<td></td>
<td>0.350191</td>
<td>0.657966</td>
<td>0.250191</td>
<td>...</td>
<td>0.639694</td>
<td>0.508947</td>
<td>0.495575</td>
</tr>
<tr>
<td></td>
<td>0.394141</td>
<td>0.737795</td>
<td>0.294141</td>
<td>...</td>
<td>0.708014</td>
<td>0.565921</td>
<td>0.535571</td>
</tr>
<tr>
<td></td>
<td>0.375669</td>
<td>0.736253</td>
<td>0.237669</td>
<td>...</td>
<td>0.695926</td>
<td>0.560965</td>
<td>0.518217</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model Application</th>
<th>C&lt;sub&gt;1&lt;/sub&gt;</th>
<th>C&lt;sub&gt;2&lt;/sub&gt;</th>
<th>C&lt;sub&gt;3&lt;/sub&gt;</th>
<th>...</th>
<th>C&lt;sub&gt;n-2&lt;/sub&gt;</th>
<th>C&lt;sub&gt;n-1&lt;/sub&gt;</th>
<th>C&lt;sub&gt;n&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.324854</td>
<td>0.623142</td>
<td>0.124708</td>
<td>...</td>
<td>0.591637</td>
<td>0.456314</td>
<td>0.455871</td>
</tr>
<tr>
<td></td>
<td>0.253674</td>
<td>0.639105</td>
<td>0.193234</td>
<td>...</td>
<td>0.597605</td>
<td>0.451987</td>
<td>0.438606</td>
</tr>
<tr>
<td></td>
<td>0.250441</td>
<td>0.538296</td>
<td>0.170112</td>
<td>...</td>
<td>0.543954</td>
<td>0.428477</td>
<td>0.377785</td>
</tr>
<tr>
<td></td>
<td>0.314232</td>
<td>0.601725</td>
<td>0.194119</td>
<td>...</td>
<td>0.628149</td>
<td>0.465281</td>
<td>0.438718</td>
</tr>
<tr>
<td></td>
<td>0.301009</td>
<td>0.618537</td>
<td>0.157692</td>
<td>...</td>
<td>0.585919</td>
<td>0.460605</td>
<td>0.426170</td>
</tr>
</tbody>
</table>
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

---

**Tool Outcomes (Area, Power, WNS/TNS)**

**Constraints**

**SAMPLER**

Arms to Sample
Samples per Arm
Max Frequency

**Parallel Tool Runs**

**Sampled Frequency vs. Iteration**

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

![Sample Graph]

- Sampled Frequency vs. Iteration
- Area vs. Time
- Iteration vs. Frequency Sampled (GHz)
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

\[
\hat{y} = f(\vec{x})
\]

\( \vec{x} = \text{netlist, constraints, floorplan} \)

\( f = ??? \)

\[
\hat{y} = \text{Slack (w/}, \text{ w/o IR)}
\]
Modeling Techniques and Flow

Parameters from netlist sequential graph

Parameters from floorplan context, constraints

Slack reports from P&R, multiphysics STA

LASSO with L1 regularization

SVM with RBF kernel

ANN with 1 input, 2 hidden, 1 output layer

Boosting with SVM as weak learner

Combine using weights

Save model and exit

Ground Truth
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

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass</td>
<td>Pass</td>
<td>584</td>
</tr>
<tr>
<td>Fail</td>
<td>Fail</td>
<td>31</td>
</tr>
</tbody>
</table>

False positives: 42
False negatives: 31
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 $\rightarrow$ 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 DRVAs 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.
• 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!

[Diagram showing target layout, 1D wires, cut masks, and final layout with dummy fill and extension]
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 (!)
Closing Multiphysics Analysis Loops

[ASPDAC16]

- Tech files, signoff criteria, corners
- AVS
- Slack
- P&R + Optimization
- Timing/Noise
- Sim Results (Dyn.) Activity Factor (Static)
- IR Drop Map
- Timing / Glitches
- Power Trace
- Power Analysis
- Reliability Report
- Functional Sim
- Sim vectors Benchmark RTL
- Temp Map
- Task Mapping/ Migration/ (DVFS)
- MTTF & Aging
Closing Multiphysics Analysis Loops

[ASPDAC16]

A. B. Kahng, 180327 ISPD--2018
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

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


Some other machine learning / data mining papers from ABKGroup

- [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)

\[ N_{\text{dies}} = \pi \left( \frac{r^2}{A} - \frac{2r}{\sqrt{2A}} \right) \]
(r: wafer radius)
Benefits from Margin Reduction at 45nm

- 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

Experiments with industry chip implementation flow
Agenda

• Scaling, Moore’s Law and Crises
• Scaling Prospects
• What’s Left for the Future?
“More Than Moore”: 2.5D/3D Integration

**Conventional Path**

<table>
<thead>
<tr>
<th>2.5D Interposer-based</th>
<th>Interconnect</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Die</td>
<td>IP Die</td>
</tr>
<tr>
<td>Micro Bump</td>
<td></td>
</tr>
<tr>
<td>TSV</td>
<td></td>
</tr>
<tr>
<td>C4 Bump</td>
<td></td>
</tr>
<tr>
<td>Interposer</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2.5D MOCHI (Marvell)</th>
<th>SoC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“Virtual” SoC</td>
</tr>
<tr>
<td>CPU</td>
<td>CPU</td>
</tr>
<tr>
<td>MODEM</td>
<td>MODEM</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>Wi-Fi</td>
</tr>
<tr>
<td>USB</td>
<td>USB</td>
</tr>
<tr>
<td>G. inn</td>
<td>G. inn</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3D TSV-based</th>
<th>Tier3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier2</td>
<td></td>
</tr>
<tr>
<td>Tier1</td>
<td></td>
</tr>
</tbody>
</table>

| 3D Bonding-based | D2W / D2D |

**Futures**

<table>
<thead>
<tr>
<th>3D Monolithic Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D- VLSI gate level</td>
</tr>
<tr>
<td>3D- VLSI MOS level</td>
</tr>
</tbody>
</table>

**3D Transfer Printing**

1. Stamp
2. Grab objects off of donor subs
3. Donor subs
4. Prints objects onto receiver

Source: LETI


A. B. Kahng, 180327 ISPD–2018
New (“Rebooting Computing”) Paradigms

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

\[
\begin{align*}
Z &= X_1 \times X_2 \\
3/8 &= 4/8 \times 6/8
\end{align*}
\]

\[\begin{array}{c}
01101010 (4/8) \\
10111011 (6/8)
\end{array}\]

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

\[Z = X_1 \times X_2 \]

\[\begin{array}{c}
00101010 (3/8)
\end{array}\]

• 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 $\rightarrow$ **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)

![Area vs. Target Frequency - AES Cipher in 28FDSOI](image)

Timing Fail

65%
BUT: Even If Wires Were Perfect (No R, C) ...

Path Delays (JPEG Encoder)

Min. cycle time = 2.8

Min. cycle time = 2.25

Delay (ns)

Path Index
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×

\[0.7 \times 0.7 = 0.49 \implies \text{density doubles at each “technology node”}\]
“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

Nominal Vdd

Static IR drop
Power grid IR gradient
Dynamic IR
HCI/NBTI

Signoff

Power and Thermal Analysis
- System
- Board
- 3DIC
- Chip
- Core
- Block
- Timing Path
- Bitcell, Stage
- Device, Wire

Analysis Uncertainty
- Timing, Noise Analysis
- On-Chip Variation
- Coupling
- Delay
- Dyn IR
- MIS
- Aging
- Perf Models
- PEX Models
- Power Models
- Device and Interconnect Models
- Measurements, Corner Definitions
- Process Technology

source: Wu 08

A. B. Kahng, 180327 ISPD--2018
Corner Explosion Worsens

Corners = \text{Process} \times \text{RCX} \times \text{Temperature} \times \text{Voltage} \times \ldots

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

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

- Lose design turnaround time (TAT) == schedule
  - Non-convergence, “ping-pong” 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
Lateral (Area) Scaling: MOL and Tracks (1)

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

- 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

<table>
<thead>
<tr>
<th>Cells</th>
<th>12T</th>
<th>9T</th>
<th>7.5T</th>
<th>6T</th>
<th>5T/4T/3T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pins</td>
<td>M1</td>
<td>M1</td>
<td>MINT/M1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>Bidirectional</td>
<td>Unidirectional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOL</td>
<td>N/A</td>
<td>Yes: MINT/M0 below M1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VDD/VSS</td>
<td>M1</td>
<td>M2</td>
<td>M1/MINT</td>
<td>Buried/backside P/G</td>
<td></td>
</tr>
</tbody>
</table>

# M2 routing tracks
~9, ~6, 5, 6, 5/4/3
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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPP=x0.78, MP=x0.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node conventional name</th>
<th>2014</th>
<th>2016</th>
<th>2018</th>
<th>2020</th>
<th>2023</th>
<th>2025</th>
</tr>
</thead>
<tbody>
<tr>
<td>N14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contacted poly pitch – [nm]</th>
<th>70</th>
<th>52</th>
<th>42</th>
<th>32</th>
<th>25</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>LELE</td>
<td>LELE,SADP</td>
<td>SADP</td>
<td>SAQP</td>
<td>SAQP</td>
<td>SAQP</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metal pitch – [nm]</th>
<th>52</th>
<th>36</th>
<th>24</th>
<th>16</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LELE, SADP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metal patterning</th>
<th>LELE, SADP</th>
<th>SAQP, EUV</th>
<th>SAQP, EUV</th>
<th>SAQP, DSA, EUV</th>
<th>SAQP, DSA, EUV</th>
<th>SAQP, DSA, EUV</th>
</tr>
</thead>
</table>

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

Tool Outcomes (Area, Power, WNS/TNS)

<table>
<thead>
<tr>
<th>Arms to Sample</th>
<th>Samples per Arm</th>
<th>SAMPLER</th>
<th>Parallel Tool Runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints</td>
<td>Max Frequency</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Sampled Frequency vs. Iteration**

- Unsuccessful samples in 5x40 run
- Successful samples in 5x40 run
- Best from 10 samples x 20 iterations
- Best from 5 samples x 40 iterations

A. B. Kahng, 180327 ISPD–2018
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
• **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

AI / ML, Academia

EDA

IC Designers

Open up tool knobs
Tool / design models; new optimization objectives

Pain points; airlocks to training / validation data;
Design-adaptive tool; flow guidance; tool outcome

Improved tools; trained (“private-label”) models

$$$$$
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

### Parameters
- Random 20% gcells for training
- Remaining 80% gcells for testing

### Initial linear model

<table>
<thead>
<tr>
<th>Predicted</th>
<th>W/o DRC</th>
<th>With DRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>W/o DRC</td>
<td>98260</td>
<td>350</td>
</tr>
<tr>
<td>With DRC</td>
<td>481</td>
<td>111</td>
</tr>
</tbody>
</table>

- True positive rate = \( \frac{tp}{t} \)
- False positive rate = \( \frac{tn}{n} \)

### Non-linear SVM model

<table>
<thead>
<tr>
<th>Predicted</th>
<th>W/o DRC</th>
<th>With DRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>W/o DRC</td>
<td>98571</td>
<td>117</td>
</tr>
<tr>
<td>With DRC</td>
<td>170</td>
<td>344</td>
</tr>
</tbody>
</table>

- True positive rate: 24%
- False positive rate: 0.5%
- True positive rate: 74%
- False positive rate: 0.2%
Improved Learning-Based Predictor

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

(a) Learning-based Prediction
(b) Actual DRC
(c)
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

<table>
<thead>
<tr>
<th>Errors</th>
<th>Testing (Total = 3442 logs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N = 200</td>
<td>Number of runs that need to be stopped</td>
</tr>
<tr>
<td>1 STOP</td>
<td>398</td>
</tr>
<tr>
<td>2 consecutive STOPs</td>
<td>398</td>
</tr>
<tr>
<td>3 consecutive STOPs</td>
<td>398</td>
</tr>
</tbody>
</table>

Test data = M0 runs
For one run, \( \# \text{iterations saved} = 20 - \) (iteration number where MDP says STOP)
Average \( \# \text{iterations saved} = \frac{\text{Sum}(\# \text{iterations saved})}{398} \)
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. \( \# \text{iterations saved} \)
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

<table>
<thead>
<tr>
<th>Errors</th>
<th>Training (Total = 1200)</th>
<th>Testing (Total = 3442)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Training Error</td>
<td>#Errors wrongly predicted to STOP (TYPE 1)</td>
</tr>
<tr>
<td>N = 200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 STOP</td>
<td>29.66%</td>
<td>251</td>
</tr>
<tr>
<td>2 consecutive STOPs</td>
<td>10.5%</td>
<td>27</td>
</tr>
<tr>
<td>3 consecutive STOPs</td>
<td>8.5%</td>
<td>3</td>
</tr>
</tbody>
</table>
Machine Learning Gives Us Scaling!

- High-value opportunities **in and around** EDA
- **Modeling and Prediction**
  - Predict tool outcome $= F(\text{design, constraints, tool config})$
  - How to run tool “optimally” for given design and design goals?
  - Avoid “failed runs” $\rightarrow$ reduce iterations in design flow
  - Dream: one-pass design flow
- Model analysis errors (crude vs. golden analyses)
  - Reduced guardbands and pessimism $\rightarrow$ 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 $\uparrow\uparrow$, schedule $\downarrow\downarrow$
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” \(\rightarrow\) reduce iterations in design flow
  • Dream: one-pass design flow

• Model analysis errors (crude vs. golden analyses)
  • Reduced guardbands and pessimism \(\rightarrow\) 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 \(\rightarrow\) Drive down iterations, margins
  • Cloud-targeted, large-scale optimizations \(\rightarrow\) 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*)