On the Road to Self-Driving IC Design Tools and Flows

Andrew B. Kahng
Depts. of CSE and ECE
UC San Diego
abk@ucsd.edu
http://vlsicad.ucsd.edu/~abk
Agenda

• Motivations
Design Crimes: Cost, Expertise, Unpredictability

- **Design cost:** not scaling
  - Design, process roadmaps not coupled
  - Figure: Andreas Olofsson, DARPA, ISPD-2018 keynote

- **Quality:** also not scaling
  - Design Capability Gap
  - **Available density:** 2x/node
  - **Realizable density:** 1.6x/node
  - Figure: UCSD / 2013 ITRS
Design is Too Difficult!

- Tools and flows have steadily increased in complexity
  - Modern P&R tool: 10000+ commands/options

- Hard to design with latest tools in latest technologies
  - Even harder to predict design quality, schedule
  - Expert users are required
  - Increased cost and risk not good for industry!

- Still have “CAD” mindset more than “DA” mindset
  - Again: assumes expert users

How do we escape this “local minimum”?
U.S. DARPA IDEA: No-Humans, 24-Hours

IDEA will create a no-human-in-the-loop hardware compiler for translating source code to layouts of System-On-Chips, System-In-Packages, and Printed Circuit Boards in less than 24 hours

- Part of DARPA Electronics Resurgence Initiative
- Traditional focus: ultimate quality
- New focus: ultimate ease of use
- No humans, 24-hour TAT = “equivalent scaling”
- Overarching goal: designer access to silicon

A. Olofsson, DARPA ISPD-2018 keynote
Agenda

- Motivations
- The OpenROAD Project
OpenROAD

DEMOCRATIZING HARDWARE DESIGN
The OpenROAD project attacks the barriers of Cost, Expertise and Uncertainty (i.e., Risk) that block the feasibility of hardware design in advanced technologies.

About OpenROAD
Problem: Hardware design requires too much effort, cost and time.
Challenge: $55 costs and "expertise gap" block system designers' access to advanced technology.
Objective: We want to enable no-humans, 24-hour design and catalyze open-source EDA.

Foundations and Realization of Open, Accessible Design
Prof. Kahng and the OpenROAD team are aiming to develop open-source tools that achieve autonomous, 24-hour layout implementation.
More Details: PowerPoint presentation from 2018 ERI Summit

Latest News and Events
OpenROAD project at the Emerging Technologies for EDA workshop in Hsinchu
Mar. 21th, 2019
Thank you! — UTD-BoxRouter
Mar. 2nd, 2019
Thank you! — BoxRouter
OpenROAD: A New Design Paradigm

24 hours, no humans – no PPA loss

Mindsets
- Achieve **predictability** from the user’s POV
- Use cloud/parallel to recover solution **quality**
- Focus on reducing **time and effort = schedule, cost**

**Machine Learning is CENTRAL to this**

- **Quality**
- **Schedule**
- **Cost**
The OpenROAD Project

- **Initial target:** digital IC flow “RTL to GDS”
  - **Inputs:** .v, .sdc, .lib, .lef
  - .def, .spef in point tools
  - config files, pre-characterizations required
  - **Outputs:** routed .def, timing/power reports

- **Open source**
- **No-human-in-loop**
  - Limited “knobs”, restricted field of use
  - Replace intelligent humans (partition, floorplan, …)

- **V1.0 release:** June 2020

See: [https://theopenroadproject.org/publications/](https://theopenroadproject.org/publications/)
Placement  https://github.com/abk-openroad/RePIAce

• RePIAce features
  • Timing-driven (OpenSTA timer integrated)
  • Mixed-size (macros + cells)
  • Electrostatics analogy in analytic placement

• RePIAce used in:
  • Physical synthesis
  • Floorplanning
  • Clock tree synthesis
  • Traditional standard-cell placement

• BSD-3 License
Static Timing Analysis https://github.com/abk-openroad/OpenSTA

• **OpenSTA**: open-sourced static timing analysis tool
  • Developer: James Cherry (Parallax Software)
  • Tested with ASAP7, GF14, TSMC16, ST28, etc.
  • **GPLv3 license**
**Slack, WNS, TNS** 28nm

**aes_cipher_top** (28nm, 12T, clkp=1000ps)

<table>
<thead>
<tr>
<th>aes_cipher_top</th>
<th>WNS (ps)</th>
<th>TNS (ps)</th>
<th>#viol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signoff STA</td>
<td>-61</td>
<td>-289</td>
<td>7</td>
</tr>
<tr>
<td>OpenSTA (arnoldi)</td>
<td>-57</td>
<td>-314</td>
<td>9</td>
</tr>
</tbody>
</table>
Agenda

• Motivations
• The OpenROAD Project
• Machine Learning for “Self-Driving”
ML in IC Design: Not Like Chess or Cat Photos

• Getting to self-driving IC design: not so obvious
  • Do recent ML successes transfer well?
  • 3-week SP&R&Opt run is NOT like playing chess!

• Design lives in a \{servers, licenses, schedule\} box
• Distributions of outcomes matter \textit{cloud, parallel}
• A “stack of models” is mandatory: Predictions of downstream outcomes are also optimization objectives

• Still uncharted road to self-driving tools and flows
  • How do we overcome “\textbf{small, expensive data}” challenges?
  • \textbf{Standards}: Learning comes from \{design + tool + technology\}, all of which are highly proprietary
    • Need mechanisms for IP-preserving sharing of data and models
Four Stages of Machine Learning

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

Huge space of tool, command, option trajectories through design flow
Stage 2. Orchestration of Search, Optimization

• How to best apply N “robot engineers” within given risk, resource limits?
  • Concurrent search of N flow trajectories
  • Explore, identify good flow options efficiently
  • Constraint: compute and license resources
• Example: “Go with the winners”
  • Launch multiple optimization threads
  • Periodically identify promising thread
  • Clone promising thread and terminate others
• Example: “Adaptive multi-start”
  • Best solutions are central to other good solutions: “big valley”
  • Adaptively choose start points for next iteration
Four Stages of Machine Learning

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

Huge space of tool, command, option trajectories through design flow
Stage 3. Modeling and Prediction

• Prediction of tool- and design-specific outcomes over longer and longer subflows
  • Wiggling of longer and longer ropes
• \( \rightarrow \text{Pruning, termination} \) to avoid wasted resources
  • Simple way to think about it: “Identify Doomed X”
  • Doomed Floorplan, Optimization run, Detailed Route run, …
  • Allocate resources elsewhere
  • Better outcome within given resource budget
• Complementary research goal: heuristics and tools that are inherently more predictable and modelable
  • Ensembles might be modeled/predicted
  • Prediction requirement might be relaxed “get user into a ballpark”? 
Generic Need: Predicting Doomed Runs

- Picture: progressions of #DR violations in commercial router
- Simple strategy: track and project key metrics as time series
- Example method: use Markov decision process (MDP): “GO” vs. “STOP” strategy card to terminate “doomed runs” early
Target 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

…………... many more requests than available students!
Generic Need: Golden From Non-Golden

ML shifts the Accuracy-Cost Tradeoff Curve (for free)!
(Old) Example: ML-based Timer Correlation

Artificial Circuits
Train

Real Designs
Validate

New Designs
Test

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

If error > threshold

ONE-TIME

INCREMENTAL

BEFORE

AFTER

ML Modeling

123 ps

31 ps

~4x reduction

T_2 Path Slack (ns)

T_1 Path Slack (ns)

T_2 Path Slack (ns)

T_1 Path Slack (ns)

DATE14, SLIP15
Lately: Predicting PBA from GBA

- PBA (Path-Based Analysis) is less pessimistic than GBA (Graph-Based Analysis)
- But, can have MUCH more expensive runtime!
- ML task: Predict PBA timing from GBA timing
  - \(\rightarrow\) Improved quality of results in P&R, optimization
  - \(\rightarrow\) Less-expensive timing analysis usable earlier in flow

[Diagram of GBA Mode and PBA Mode]

https://vlsicad.ucsd.edu/Publications/Conferences/361/c361.pdf
Lately: Reduce #Corners in STA and Opt

- Want all the benefits of STA at N corners, but want to pay for analysis at only M << N corners
  - "Missing Corner Prediction" ("matrix completion") saves runtime, licenses
  - "Primary corners" methodology $\rightarrow$ errors caught at signoff cause iteration

---

Model Training

<table>
<thead>
<tr>
<th>C_1</th>
<th>C_2</th>
<th>C_3</th>
<th>...</th>
<th>C_{n-2}</th>
<th>C_{n-1}</th>
<th>C_n</th>
</tr>
</thead>
<tbody>
<tr>
<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>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>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>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>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>

Model Application

<table>
<thead>
<tr>
<th>C_1</th>
<th>C_2</th>
<th>C_3</th>
<th>...</th>
<th>C_{n-2}</th>
<th>C_{n-1}</th>
<th>C_n</th>
</tr>
</thead>
<tbody>
<tr>
<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>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>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>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>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>

https://vlsicad.ucsd.edu/Publications/Conferences/369/c369.pdf
Lately: “ML-LEAK” (leakage recovery predictor)

- Predict leakage recovery if user runs {Tweaker, Tempus ECO, PTSI ECO, homegrown script, …}
  - Beneficial to methodology team when trying out various DOEs
  - Saves time for implementation team: skip leakage recovery if it won’t help
- Blended model of design and instance level predictions gives best results

Actual leakage power recovery: 0.076%.
Our model prediction: 1% leakage power recovery.

Actual (x) vs. predicted (y) percentage change in leakage power after recovery
Agenda

• Motivations
• The OpenROAD Project
• Machine Learning for “Self-Driving”
• Infrastructure for Machine Learning: METRICS
ML in IC Design Requires Infrastructure!

- **Support for ML in IC design**
  - Standards for model encapsulation, model application, and IP preservation when models are shared

- **Standard ML platform for EDA modeling**
  - Design metrics collection, (design-specific) modeling, prediction of tool/flow outcomes
  - This recalls “METRICS” [http://vlsicad.ucsd.edu/GSRC(metrics](http://vlsicad.ucsd.edu/GSRC(metrics)

- **Datasets to support ML**
  - Real designs, Artificial designs and “Eyecharts”
  - **Shared training data** – e.g., analysis correlation, post-route DRV prediction, sizer move trajectories and outcomes, …
  - Challenges and incentives: “Kaggle for ML in IC design”
“METrICS”

The METRICS Initiative

Recent Updates

• Survey (1st draft) for design quality and productivity (the multiple-choice version)
• Reduced survey (2nd draft) for design quality and productivity that is distributed at June 2001 GSRC workshop
• Updated survey (3rd draft) for design quality and productivity that reflects the discussion at June 2001 GSRC workshop
• Workshop notes for METRICS discussion at June 2001 GSRC workshop
• List of prediction/estimator models enabled by METRICS System
• DAC02 Birds-of-a-Feather meeting summary (June 12, 2002)

• METRICS (1999; DAC00, 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
Original METRICS Architecture

- Instrumentation of design tools:
  - Wrapper scripts to extract data from outputs and logfiles,
  - Callable API codes that allow direct interaction from within the design tools
- METRICS server: central data collection (Oracle8i)
- Data mining process: analyzes existing data to improve existing design flow (CUBIST, etc.)
Proposed METRICS 2.0 Architecture
## METRICS 2.0 Evolution Path

<table>
<thead>
<tr>
<th>METRICS2.0 metric</th>
<th>Definition</th>
<th>Flow Stage</th>
<th>Collectable from tool reports?</th>
<th>Derivable from tool run data?</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNTHESIS::INST::COMB</td>
<td>Total #Comb Instances</td>
<td>Synth</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SYNTHESIS::INST::REG</td>
<td>Total #Registers</td>
<td>Synth</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PLACEMENT::INSTANCES::TOTAL</td>
<td>Total #Instances</td>
<td>Place</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PLACEMENT::PIN_DENSITY</td>
<td>Ratio of Total #Pins / Total Area</td>
<td>Place</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>PLACEMENT_OPT::POWER::TOTAL</td>
<td>Total power after preCTS Opt</td>
<td>Place</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>CLOCK::INST</td>
<td>Total #Clock Insts</td>
<td>CTS</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>CLOCK::TIMING::SETUP::TNS</td>
<td>Total TNS of timing paths</td>
<td>CTS</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>CLOCK::TIMING::SKEW::AVG</td>
<td>Average clock skew</td>
<td>CTS</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ROUTING::DRC::TOTAL</td>
<td>Total DRC Count</td>
<td>Routing</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ROUTING::VIA::TOTAL</td>
<td>Total #Vias</td>
<td>Routing</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ROUTING::WIRELENGTH::TOTAL</td>
<td>Total Wirelength</td>
<td>Routing</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

- METRICS 2.0 → learning from recent runs → adaptive flows → reinforcement learning
- METRICS 2.0 + Grid Computing = shared burden of “big data”
- METRICS 2.0 + Federated ML = privacy-preserving models
METRICS 2.0 + Federated Learning

- **Centralized**
  - Have storage and computation need on server
  - Exposure of METRICS to public domain

- **Federated**
  - Light server, distributed, spare cycle-aware training
  - Data remains private
Agenda

• Motivations
• The OpenROAD Project
• Machine Learning for “Self-Driving”
• Infrastructure for Machine Learning: METRICS
• Conclusions
Conclusions

• **Self-driving IC design is a mandatory scaling lever**
  • Restores accessibility of hardware design

• **New mindsets required**
  • Design lives within a {servers, licenses, schedule} box
  • Care about distributions, expectations of outcomes
  • “Stack of models” is mandatory: Predictions of downstream outcomes are also optimization objectives

• **Largely uncharted road to self-driving tools, flows**
  • “Intelligence” and ML focus today, vs. “small, expensive data”
  • IP- and privacy-preserving model sharing, federated learning
  • Not discussed here: open-source tools as part of big picture

• **Active, global engagement essential**
  • Standards (METRICS 2.0, ML model encapsulation, …)
  • Existing modeling methods (e.g., analysis calibration), codes

• Last message: Please engage and help!
THANK YOU!

Research supported by U.S. National Science Foundation, U.S. DARPA, Samsung, Qualcomm, NXP Semiconductors, Mentor Graphics, and the C-DEN Center.

A few links:
- [https://theopenroadproject.org/](https://theopenroadproject.org/)
- Several presentations are linked under News at [https://vlsicad.ucsd.edu/](https://vlsicad.ucsd.edu/)
- [https://vlsicad.ucsd.edu/Publications/Conferences/356/c356.pdf](https://vlsicad.ucsd.edu/Publications/Conferences/356/c356.pdf)
- [https://vlsicad.ucsd.edu/Publications/Conferences/360/c360.pdf](https://vlsicad.ucsd.edu/Publications/Conferences/360/c360.pdf)