Open-Source EDA for RTL-to-GDS Implementation: Updates and Futures

Andrew B. Kahng
UCSD Depts. of CSE and ECE
abk@ucsd.edu
https://vlsicad.ucsd.edu

VSDOpen Conference 2019
October 19, 2019

Slides of this talk: See first link under “News” at https://vlsicad.ucsd.edu/
Agenda

• What is Open-Source EDA, and Why Now?
What is Open Source?

Industry standard definition:

- Code released under an open source license

- Very good primers from Tim Ansell (Google):
  - j.mp/eri19-foss101
  - j.mp/eri19-foss102

- Freely usable, freely modifiable, and shareable
Why Open-Source EDA?

- **Accelerate research and innovation**
  - EDA research: avoid reinventing wheels
  - VLSI design: bring ideas into reality **at zero cost**

- **Anyone can use it, for any purpose**
  - To learn
  - To tape out
  - To research new algorithmic ideas

- **Anyone can contribute**
  - Worldwide collective effort is both powerful and needed!
  - Many ways to help
    - Making documentation better is a contribution
    - Contribute a testcase!
  - If you see room for improvement:
    - Work on it and create a pull request to be merged
    - If you don’t code, give feedback!
Open-Source EDA: Why Now?

- Crisis of DESIGN: it’s difficult, expensive and risky!
- This blocks innovation

Has EDA failed to keep up with Moore’s Law?
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

A. Olofsson, DARPA ISPD-2018 keynote

- Part of DARPA Electronics Resurgence Initiative
- Traditional focus: ultimate performance, power, area
- IDEA focus: ultimate ease of use and runtime

*Restore designer access to silicon*
The OpenROAD Project

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

A. Olofsson, DARPA
ISPD-2018 keynote

- Traditional focus: ultimate performance, power, area
- OUR focus: ultimate ease of use and runtime

Initial focus: Digital IC “RTL-to-GDS” flow
Agenda

• What is Open-Source EDA, and Why Now?
• The OpenROAD Project
theopenroadproject.org

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.

READ MORE

Open Source Tools
SHOW ON GITHUB

User Guide
GETTING STARTED

Community
JOIN THE DISCUSSION

OpenROAD 11 Sep
Final submissions of the “ICCAD 2019 LEF/DEF Based Global Routing Contest” are now being evaluated! See the contest description paper by #UCSD and #MentorGraphics here https://t.co/7mW3mw9uPj. Results and open-source bonuses announced at #ICCAD!

#contest #VLSI | Twitter

About OpenROAD

Problem: Hardware design requires too much effort, cost and time.

Challenge: $$$ costs and "expertise gap" block system designers' access to advanced technology.

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.

Latest News and Events

ERI Summit 2019: OpenROAD presentation video posted

September 11, 2019

OpenROAD Alpha tools in the IEEE CEDA DATC’s RDF-2019 flow!
Initial Focus: Digital RTL-to-GDS Flow

- Alpha milestone: July 2019
- DRC-clean in TSMC65LP with Arm cells, IPs
  - Important foundation and proof point
  - Breakthrough for academic detailed routing
- v1.0 release: July 2020
  - Target: DRC-clean in a commercial FinFET node (16/14nm)

See: [https://theopenroadproject.org/publications/](https://theopenroadproject.org/publications/)
[https://github.com/The-OpenROAD-Project](https://github.com/The-OpenROAD-Project)
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>

https://github.com/The-OpenROAD-Project/OpenSTA
RePIAce

• RePIAce: Nesterov-based global placement tool
  • https://github.com/The-OpenROAD-Project/RePIAce
  • Open-sourced July 2018
  • 69 ★s and 228 commits since then
  • 42/50 issues solved

“Coyote” (RocketChip) in TSMC16FFC, Arm 7.5T cells
July 2019 “alpha”

https://github.com/The-OpenROAD-Project/alpha-release
July 2019 “alpha”  
https://github.com/The-OpenROAD-Project/alpha-release

Verification Complete: 0 Viols.

---

A. B. Kahng, 191019 VSDOpen Conference 2019
The OpenROAD Project

OpenROAD seeks to develop and foster an autonomous, 24-hour, open-source layout generation flow (RTL-to-GDS).

https://thopenroadproject.org/

Repositories 55

Pinned repositories

- **alpha-release**
  - Builds, flow and designs for the alpha release
  - Verilog: 41, Python: 12

- **docs**
  - OpenROAD Documentation
  - Python: 4

Find a repository...

Type: All  Language: All

**OpenDB**

Database and Tool Framework for EDA

database  eda

Top languages

- C++
- BSD-3-Clause
- C

Updated 2 hours ago
Agenda

• What is Open-Source EDA, and Why Now?
• The OpenROAD Project
• Critical Mass + Critical Quality = A Bright Future
• **Lesson 1:** Must never lose sight of culture change and worldwide engagement!

• **Lesson 2:** Must achieve critical mass, critical quality
Feedback From the Community

- DAC-2019 Birds-of-a-Feather Meeting on “Open-Source Academic EDA Software”

- Clear feedback especially from potential users (designers):
  - Get a full flow working
  - Show us tools integrated on a shared back-end database
  - Software quality is essential

  = “Table Stakes”
Building for the Future

• **Must** build open-source EDA on strong foundations

• Architecture of integrated RTL-to-GDS application
• Shared database and “incremental substrate”
• Product engineering
• Software engineering
• Devops (build, continuous integration + deployment, QA/test)
• ...

= aspects of Critical Mass, Critical Quality
Integrated Architecture for SP&R

• How is modern synthesis-place-route architected?

• 40 years of EDA industry learning curve:
  • 1980s: file-based integration tool chain (“Alpha”, July 2019)
  • 1990s: “common timing engine”
  • 2000s: tightly coupled algorithms on a shared incremental substrate tight integration (v1.0, July 2020) → implies first-ever shared DB layer in permissive open source!
  • 2010s: “hyperoptimization” on same architecture
Incremental Shared Netlist Architecture

- Script
- Shell Interpreter
- Synthesis
- Placement
- CTS
- Post-Place Opt (Sizing)
- Routing
- Timer
- Delay Calc
- Shared Netlist or Abstract Network Adapter
- Shared Physical or Data Model Adapter
- Timer API
- Netlist changes callback
- Delays for a stage
- Routed Metal

FLOW
Database: OpenDB Release Oct 5, 2019

- Commercially developed back-end database and EDA framework
- [https://github.com/The-OpenROAD-Project/OpenDB](https://github.com/The-OpenROAD-Project/OpenDB)
- A HUGE milestone for open-source EDA (more coming soon)
Usability: Tcl Interfacing

• Tcl i/f being added to all OpenROAD tools
• Example: RePIAce Tcl commands list
  • [https://github.com/The-OpenROAD-Project/RePIAce/blob/master/doc/TclCommands.md](https://github.com/The-OpenROAD-Project/RePIAce/blob/master/doc/TclCommands.md)

Usage with TCL Interpreter

RePIAce has internal TCL Interpreter. The following line will create replace_external objects.

```
replace_external rep
```

After having a replace_external object, a user can type any TCL commands after one spacing from the object name(e.g. rep).

```
rep [tcl_command]
```

File I/O Commands

• `import_lef [file_name]`: *.lef location (Multiple lef files supported. Technology LEF must be ahead of other LEFs.)
• `import_def [file_name]`: *.def location (Required due to FloorPlan information)
• `export_def [file_name]`: Output DEF location
• `set_output [directory_location]`: Specify the location of output results. Default: ./output

Flow Control

• `init_replace`: Initialize RePIAce's structure based on LEF and DEF.
INFO] Nesterov: 110 OverFlow: 0.634284 ScaledHpwl: 106748
INFO] Nesterov: 120 OverFlow: 0.632794 ScaledHpwl: 106791
INFO] Nesterov: 130 OverFlow: 0.630423 ScaledHpwl: 106849
INFO] Nesterov: 140 OverFlow: 0.627051 ScaledHpwl: 106973
INFO] Nesterov: 150 OverFlow: 0.622321 ScaledHpwl: 107141
INFO] Nesterov: 160 OverFlow: 0.615691 ScaledHpwl: 107416
INFO] Nesterov: 170 OverFlow: 0.605333 ScaledHpwl: 107717
INFO] Nesterov: 180 OverFlow: 0.591968 ScaledHpwl: 108084
INFO] Nesterov: 190 OverFlow: 0.575242 ScaledHpwl: 108575
INFO] Nesterov: 200 OverFlow: 0.554654 ScaledHpwl: 109024
INFO] Nesterov: 210 OverFlow: 0.524712 ScaledHpwl: 109445
INFO] Timing: WNS = -5.87949e-10
INFO] Timing: TNS = -1.19034e-07
INFO] Nesterov: 220 OverFlow: 0.494359 ScaledHpwl: 111395
INFO] Nesterov: 230 OverFlow: 0.461807 ScaledHpwl: 111482
INFO] Nesterov: 240 OverFlow: 0.422991 ScaledHpwl: 111586
INFO] Nesterov: 250 OverFlow: 0.391335 ScaledHpwl: 112098
INFO] Nesterov: 260 OverFlow: 0.360562 ScaledHpwl: 112488
INFO] Nesterov: 270 OverFlow: 0.326674 ScaledHpwl: 112939
INFO] Timing: WNS = -7.20778e-10
INFO] Timing: TNS = -1.30904e-07
INFO] Nesterov: 280 OverFlow: 0.288795 ScaledHpwl: 113279
INFO] Nesterov: 290 OverFlow: 0.255209 ScaledHpwl: 113564
INFO] Nesterov: 300 OverFlow: 0.226158 ScaledHpwl: 113971
INFO] Nesterov: 310 OverFlow: 0.197947 ScaledHpwl: 114497
INFO] Nesterov: 320 OverFlow: 0.16818 ScaledHpwl: 114903
INFO] Timing: WNS = -8.57328e-10
INFO] Timing: TNS = -1.49771e-07
INFO] Nesterov: 330 OverFlow: 0.14172 ScaledHpwl: 115343
INFO] Nesterov: 340 OverFlow: 0.121336 ScaledHpwl: 115772
INFO] Nesterov: 350 OverFlow: 0.105172 ScaledHpwl: 116267
INFO] Timing: TNS = -1.52562e-07
nesterovPlace HPWL: 22092.115234375
final WNS: -8.698358699987807e-10
final TNS: -1.5256213714565092e-7
mgwoo at dfm in ~/01_placement/RePlAce/test/demo3 on master [!]$
Power of Community, Agile: Git, Jenkins CI

- **Anyone can contribute**
- Worldwide collective effort is both powerful and needed!
  - Many ways to help
  - Making documentation better
  - Contribute a test
  - If you see something that needs work
  - Want to add something

https://github.com/The-OpenROAD-Project/
Open Source: You Can Create a GitHub Issue

In the image, a screenshot of a GitHub repository is shown. The repository is The-OpenROAD-Project/RePIAce. The screenshot highlights the issues section with 8 open issues. One of the issues is titled "Segfaults on examples" and has been opened 3 days ago by fosshardware. Another issue is titled "Segfaults in the MuxTest*.v series of tests" and was opened on Aug 11 by charboe. The screenshot also shows a button for "New issue" which allows users to create new issues for the repository.
Open Source: You Can Create a GitHub Issue

A. B. Kahng, 191019 VSDOpen Conference 2019
You Can Create a Pull Request
OpenROAD Git Flow

• The ‘develop’ branch maintains day-to-day development activities
  • Runs unit tests and tool regression tests
  • All contributions (PRs) are issued to ‘develop’
• Changes merged to the ‘openroad’ branch are tested against the end-to-end flow
  • A repo owner initiates this merge to ensure that tools still run within OpenROAD flow
• The ‘master’ branch holds a stable release of the codebase
OpenROAD Continuous Integration (CI)

Shared Data Model Adapter

- Timer
- Delay Calc
- Network Adaptor
- Synthesis
- Placement

Flow Unified Build

```
git submodule add https://<git-server>/library-b
```

Timer

```
git push
```

Jenkins Continuous Integration Server

Build Servers

- Build
- Unit Tests

Build

- Uniform Build
- Integration Tests
- QoR Tests

Release

- develop
- unified-build
- master
OpenROAD Continuous Integration (CI)

Parallel Tests

Saving Developer Time
OpenROAD Architecture, Software Leadership

- Tom Spyrou: Well known EDA system architect, OpenAccess database, PrimeTime
  - Full-time in San Diego (office is next door to mine)
  - OpenROAD chief architect / technical project manager

- James Cherry: developer of Pearl and Parallax static timing engines
  - OpenSTA, first Resizer, floorplan .def initialization (.v to .def), readers/writers

- + More (see our People page)
  - EDA veterans giving back to the field, mentoring the next generation
Many balls in the air (PKG, PCB, AI/ML, 7nm, …)
Please contribute!

Building for the Future

- Must build open-source EDA on strong foundations
- Architecture of integrated RTL-to-GDS application
- Shared database and “incremental substrate”
- Product engineering
- Software engineering
- Devops (build, continuous integration + deployment, QA/test)
- ...

= Critical Mass, Critical Quality
@Students, especially: Do you want to learn about EDA development while helping OpenROAD?
Agenda

• What is Open-Source EDA, and Why Now?
• The OpenROAD Project
• Critical Mass + Critical Quality = A Bright Future
• (Brief ad: Come to a full-day tutorial in Bangalore!)
Open-Source EDA and Machine Learning for IC Design: A Live Update

ABSTRACT: Open-source EDA is rapidly enabling new waves of innovation on many fronts. For academic researchers, it speeds up the lifecycle of scientific progress and makes research results relevant to modern industry practice. For EDA professionals and the industry ecosystem, open-source EDA is a complement and booster to commercial EDA. For the IC design community, recent releases under permissive licenses make it possible for design engineers as well as hobbyists to take ideas to manufacturing-ready layout at essentially zero cost. This full-day tutorial will review the very latest progress in open-source EDA, focusing on the digital RTL-to-GDS flow. In particular, the tutorial will make a deep dive into The OpenROAD Project https://theopenroadproject.org/, which brings new open-source tools and machine learning into a “no human in the loop” RTL-to-GDS flow. OpenROAD’s key components will be reviewed, along with how the overall system is architected. Here, we will provide live / hands-on demos on using the OpenROAD flow and integrating it into existing design flows. We will then turn to industry-compatible open-source infrastructure, and then demonstrate how OpenROAD’s continuous integration (CI) system works in order to welcome new contributions and give a low-overhead start for design tool and methodology innovation. Importantly, open-source EDA and the goal of “self-driving”, no-humans IC design puts a spotlight on machine learning that reduces design effort and schedule. We will show examples of important machine learning use cases and proof points in the RTL-to-GDS flow, along with demos of an open-source, freely portable integrated metrics collection system. The target audience of this tutorial includes EDA researchers and developers, graduate students and professors, and IC design methodologists, physical design engineers, and managers. Whether you are a veteran in EDA / IC design or just starting a career, the tutorial will give a new perspective and a roadmap to use and contribute to open-source EDA.
### Full-Day Tutorial Jan 4 2020 in Bangalore

<table>
<thead>
<tr>
<th>Section Title</th>
<th>Duration (hrs)</th>
<th>Contents</th>
<th>By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>0.5</td>
<td>Why open source, who benefits, why not until now ...</td>
<td>ABK</td>
</tr>
<tr>
<td>OpenROAD today</td>
<td>0.5</td>
<td>Including “FAQs” - is it really free? DARPA? ...</td>
<td>ABK</td>
</tr>
<tr>
<td>Key components + DEMOS</td>
<td>1.5</td>
<td>OpenSTA timer, RePIAce and OpenDP placers, OpenDB database and EDA framework, PEX, ...</td>
<td>Both</td>
</tr>
<tr>
<td>“Something we can all build on”</td>
<td>1.5</td>
<td>Concrete examples: how to use in research, how to use in design; how to contribute Highlight: build/CI infrax</td>
<td>AH</td>
</tr>
<tr>
<td>Key use context: Machine learning for EDA and IC</td>
<td>2.0</td>
<td>Toward “no-humans”; metrics + examples of high-value, low-hanging fruit for machine learning. METRICS 2.0 infrastructure and usage demos</td>
<td>Both</td>
</tr>
<tr>
<td>design + DEMOS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Questions? + Flexible Time</td>
<td>0.5</td>
<td></td>
<td>Both</td>
</tr>
</tbody>
</table>
Full-Day Tutorial Jan 4 2020 in Bangalore

Andrew B. Kahng is Professor of CSE and ECE and holder of the endowed chair in high-performance computing at UC San Diego. He was visiting scientist at Cadence (1995-97) and founder/CTO at Blaze DFM (2004-06). He is coauthor of 3 books and over 500 journal and conference papers, holds 34 issued U.S. patents, and is a fellow of ACM and IEEE. He has served as general chair of DAC, ISPD and other conferences, and from 2000-2016 as international chair/co-chair of the ITRS Design and System Drivers working groups. He is currently PI of “OpenROAD” https://theopenroadproject.org/, a $12.4M U.S. DARPA project targeting open-source, autonomous (“no human in the loop”) tools for IC implementation.

Abdelrahman Hosny is a second-year Ph.D. student in the Computer Science department at Brown University. He mixes software industry experience with his research to reimagine the Electronic Design Automation (EDA) landscape. His Ph.D. research tries to make silicon compilation catch up with advance in software compilation: free, open source and easy to use. Towards that goal, he investigates machine learning techniques (specifically, reinforcement learning) for optimizing EDA flows with no human in the loop. Abdelrahman received a Master’s degree in Computer Science and Engineering from the University of Connecticut, and a Bachelor’s degree in Computer Science from Assiut University.
Acknowledgments and Links

- Many thanks and all credits are due to the students and collaborators in the OpenROAD project: [https://theopenroadproject.org/our-team/](https://theopenroadproject.org/our-team/)

- Project links:
  - [https://theopenroadproject.org/](https://theopenroadproject.org/)
  - [https://github.com/The-OpenROAD-Project/](https://github.com/The-OpenROAD-Project/)
  - Twitter: [https://twitter.com/OpenROAD_EDA](https://twitter.com/OpenROAD_EDA)
  - Newsletter: [https://theopenroadproject.org/newsletter/](https://theopenroadproject.org/newsletter/)

- Our forthcoming tutorial on January 4, 2020 in Bangalore:

- Many presentations linked under News at [https://vlsicad.ucsd.edu/](https://vlsicad.ucsd.edu/)

- Background on Machine Learning “in and around IC design tools”:
  - [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)
  - And a collection of links is [HERE](https://vlsicad.ucsd.edu/Publications/Conferences/374/c374.pdf).

- “Looking into the Mirror of Open Source”: [https://vlsicad.ucsd.edu/Publications/Conferences/374/c374.pdf](https://vlsicad.ucsd.edu/Publications/Conferences/374/c374.pdf)

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

• What is Open-Source EDA, and Why Now?
• The OpenROAD Project
• Critical Mass + Critical Quality = A Bright Future
• (Brief ad: Come to a full-day tutorial in Bangalore!)
• Conclusions
Conclusions

• **Open-source EDA offers a bright future for all of us**
  • Accelerate research and innovation in both EDA and IC design
  • Can be used for any purpose
  • Anyone can contribute → many ways to help

• **OpenROAD is all in on open-source EDA**
  • Making visible progress toward Critical Mass, Critical Quality
    • OpenSTA, RTL-to-GDS tool chain, DRC-clean layout, OpenDB, Jenkins/CI, …

• **Looking forward**
  • Running hard, and want to run harder
  • Opening doors to fundamental research to enable no-humans, 24-hour, “self-driving” tools and flows
  • Machine learning for EDA/IC design, “metrics” infrastructure, …

Open-Source EDA is **Rewarding**.
Let’s do this together!
THANK YOU !

abk-openroad@eng.ucsd.edu

Twitter: @OpenROAD_EDA
Backup
IEEE Council on EDA’s “Robust Design Flow”

**OpenROAD tools fill in many gaps to complete the first full reference academic flow**

- “RDF-2019” will be announced at ICCAD-2019 next month
- [https://vlsicad.ucsd.edu/Publications/Conferences/373/c373.pdf](https://vlsicad.ucsd.edu/Publications/Conferences/373/c373.pdf)

---

**OpenROAD**

- **Yosys+ABC**
- **TritonFP**: v2def (Resizer) + ioPlacer + RePIAce + TritonMacroPlace + pdn + tapcell
- **RePIAce** + Resizer + OpenDP
- **OpenDP**
- **OpenSTA** + SPEF from placed DEF
- **Resizer, TritonSizer**
- **TritonCTS** + OpenDP
- **FastRoute4-lefdef**
- **TritonRoute**
- **OpenSTA** + SPEF from routed DEF
- **Magic**

---

DATC RDF-2019: Towards a Complete Academic Reference Design Flow

**Invited Paper**

Jianli Chen¹, Iris Hui-Ru Jiang¹, Jinwook Jung¹, Andrew B. Kahng³, Victor N. Kravets³, Yih-Lang Li³, Shih-Ting Lin³, and Mingyu Woon³

¹National Taiwan University, Taipei, Taiwan
²IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
³UC San Diego, La Jolla, CA, USA

Abstract: We describe a new RDF-2019 release of the IEEE CEDA DATC Robust Design Flow (RDF). RDF-2019 enhances the DATC-RDF to span the entire RTL-to-GDS II implementation functions as initialization of floorplan DEF, IO placement, macro placement, clock tree synthesis, and generation of gate-level netlist.
Well-Known Naysayer Arguments …

• Open source brings a new world where research is reproducible, and progress is made directly at leading edge
  • Benchmarking is encouraged!
  • Tcl scripts for open-source tools can be shared!

• “It’s really, really difficult”? (+ $B’s of R&D, 1000’s of eng-years)
  • Agree! Contributors, advisers, reviewers, cooperation/coordination all needed.
  • And open-source EDA has a totally different target than commercial EDA

• “Students shouldn’t spend time polishing code for open-sourcing”?
  • They saved months if not years because of open source (and so did their advisors…) → Pay it forward!

• “Student developers can’t possibly … {see foundry PDK, accept a commercial testcase with bug report, make golden verifiers, …}”?
  • An engaged community and design ecosystem will find workarounds

• “Not everyone believes in open source”?  
  • Open-source EDA will be a movement and ecosystem (not a law)
OpenROAD’s Foundation Technologies

24 hours, no humans – no PPA loss

Mindsets

• Use cloud/parallel to achieve predictability, recover solution quality
• Relentless focus on time, effort reduction
Open-Source EDA Trajectory

- Census of Open-Source Academic EDA Software (Digital IC)
- (Not all tools have proper open-source licenses)
(DAC-2019 Paper)
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 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 “small, expensive data” challenges?
  • 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
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]

• 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
## 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
Why Open Source has Strategic Value Now

- **Extreme consolidation**: 2 fabs, 2 EDA companies, at most 3 standing in GPU, FPGA, Mobile SOC, …
- **Design technology is higher-value**
- **Private-label EDA is inevitable**
  - One of few levers for differentiation
- **Foundry**: protect IP leakage, roadmap better than current DTCO mechanisms
- **Fabless**: protect key SOC methodology, design innovation that otherwise leaks to competitors
- **EDA**: more useful research, in more usable forms
- **Academia**: see previous slides (and, engage … or not)