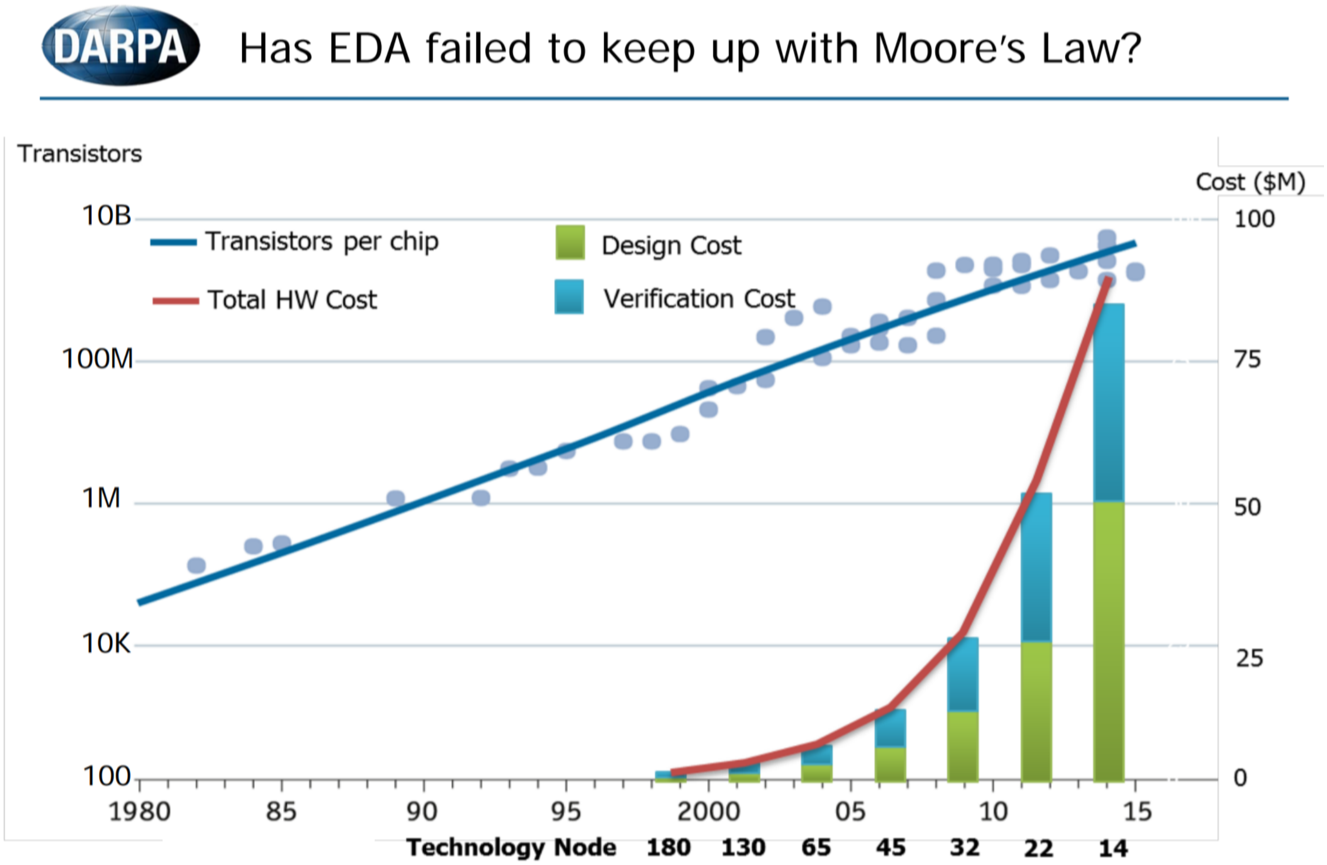
DAC 2019

Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project

Tutu Ajayi, Vidya A. Chhabria, Mateus Fogaca, Soheil Hashemi, Abdelrahman Hosny, Andrew B. Kahng, Minsoo Kim, Jeongsup Lee, Uday Mallappa, Marina Neseem, Geraldo Pradipta, **Sherief Reda**, Mehdi Saligane, Sachin S. Sapatnekar, Carl Sechen, Mohamed Shalan, William Swartz, Lutong Wang, Zhehong Wang, Mingyu Woo and Bangqi Xu

Design Crises: Cost, Expertise, Unpredictability

Design cost: not scaling! Design, process roadmaps not coupled



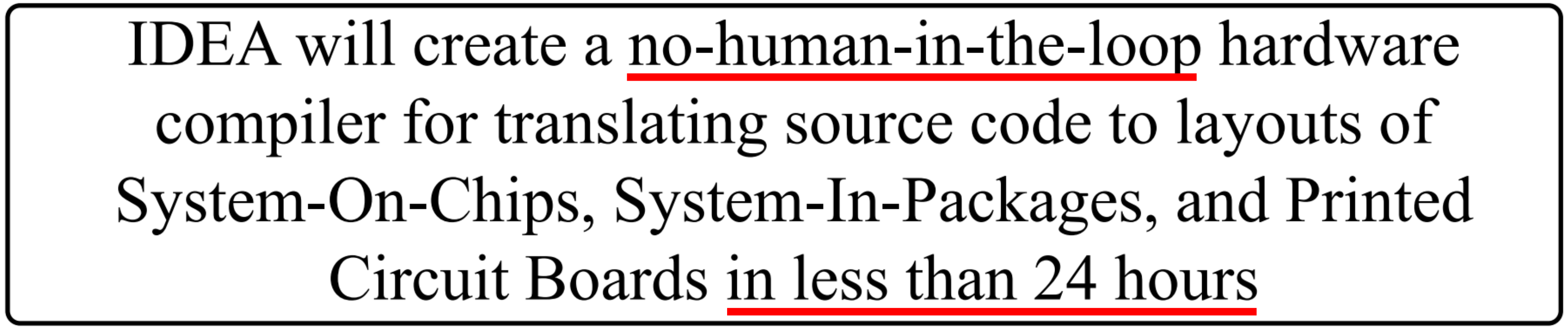
[Figure: Andreas Olofsson, DARPA, ISPD-2018 keynote]

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 quality, schedule
  - Expert users 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” ?

IDEA: No Human in the Loop, 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

A. Olofsson, DARPA
ISPD-2018 keynote

- 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

theopenroadproject.org

ANDREW KAHNG*

UC San Diego

CHUNG-KUAN CHENG

UC San Diego

LAWRENCE SAUL

UC San Diego

SHERIEF REDA*

Brown University

DAVID BLAAUW

U. Michigan

DENNIS SYLVESTER*

U. Michigan

RONALD DRESLINSKI

U. Michigan

SACHIN SAPATNEKAR*

U. Minnesota

CARL SECHEN*

UT Dallas

BILL SWARTZ

UT Dallas

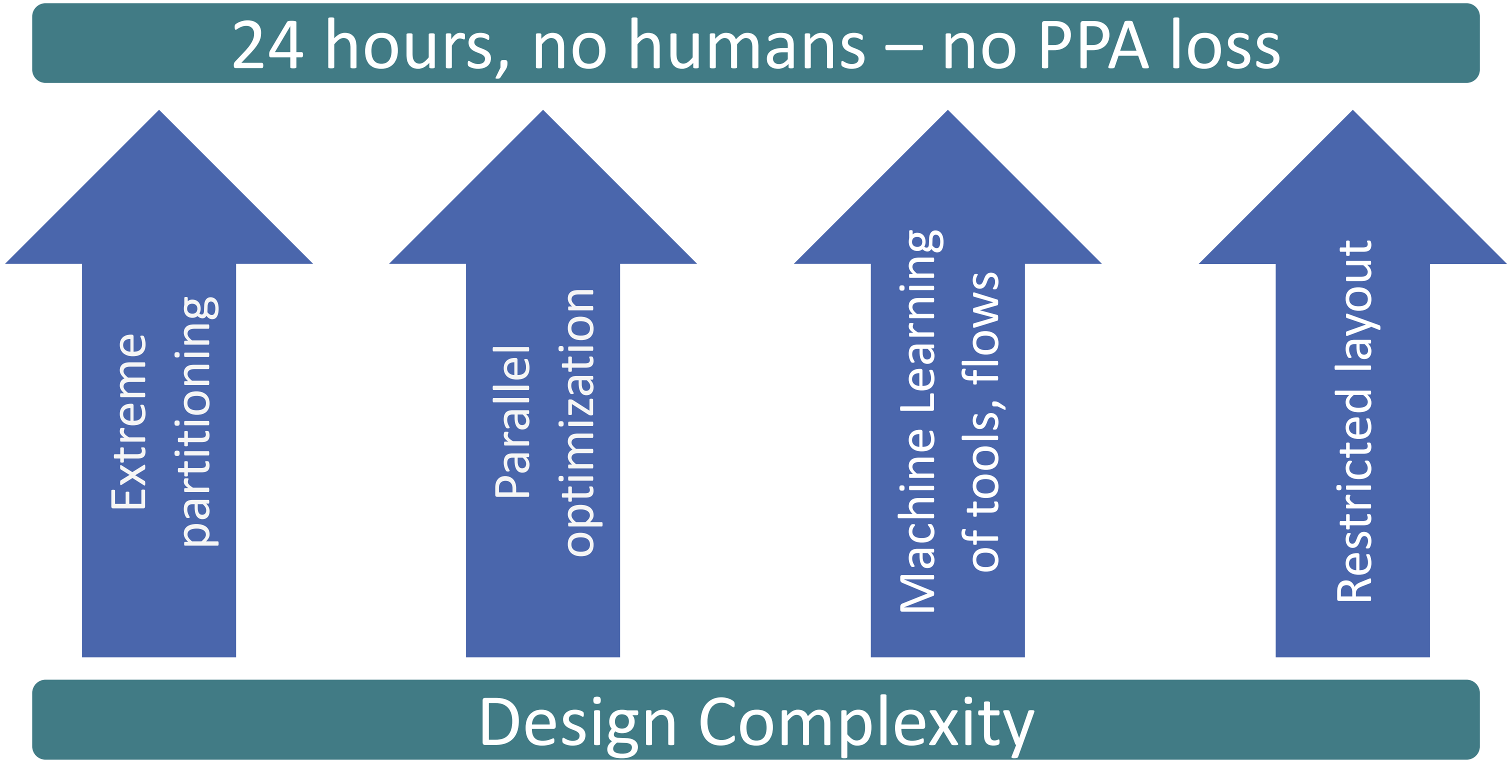
MATTEO COLTELLA*

Arm

PAUL PENZES*

Qualcomm

OpenROAD: A New Design Paradigm



Machine Learning is CENTRAL to this

Mindsets

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

✓ Quality
✓ Schedule
✓ Cost

Open Source

- Clarity, better science
  - Leading edge becomes visible and well-defined
  - Advances are verifiable: no more “irreproducible results” !
- Avoid reinventing wheels
  - Field advances more rapidly
- Strong correlation with maturity, health of field
  - Look at AI / ML, computer vision …
  - Strategic advantages to researchers and companies alike
- Do for EDA what Linux has done for software
  - Follow on the heels of other open source hardware initiatives

Also: “Paying It Forward”

- EDA began with open source: SPICE, SUPREM, MAGIC, ESPRESSO, FASTCAP, MIS/SIS, …
- Many open-source tools continue to be developed today

Tool	Provider	License	I/O Format
yosys	Clifford Wolf	ISC (BSD-2)	Industry
Parquet	UMich		Academic
NTUPlace 2.0	NTU		Academic
NTHU GR	NTHU		Academic
Qrouter	Tim Edwards	GPL 2.0	Industry
OpenTimer2.0	UIUC	MIT	Industry
Rsyn	UFRGS	Apache 2.0	Industry

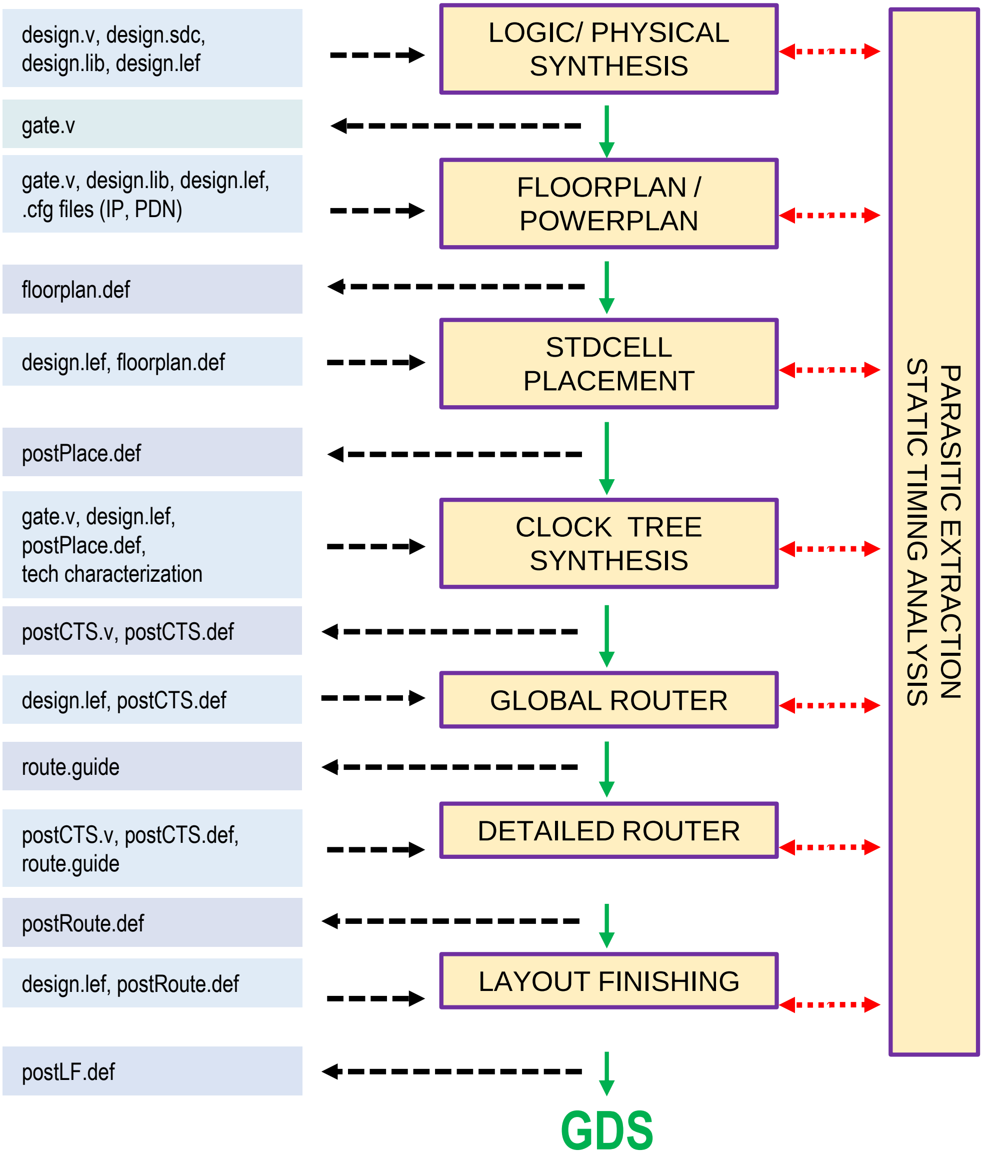
We want to see critical mass, culture of FOSS in our domain

Initial Target: RTL-to-GDS Layout Generation

- Inputs: .v, .sdc, .lib, .lef
  - .def, .spef in point tools
  - config files required
  - pre-characterizations required

- Outputs: post-route .def, timing/power estimates

- Alpha release: July 2019
- V1.0 release: June 2020



design.v, design.sdc, design.lib, design.lef

LOGIC/ PHYSICAL SYNTHESIS

gate.v

gate.v, design.lib, design.lef, .cfg files (IP, PDN)

FLOORPLAN / POWERPLAN

floorplan.def

design.lef, floorplan.def

STDCELL PLACEMENT

postPlace.def

gate.v, design.lef, postPlace.def, tech characterization

CLOCK TREE SYNTHESIS

postCTS.v, postCTS.def

design.lef, postCTS.def

GLOBAL ROUTER

route.guide

postCTS.v, postCTS.def, route.guide

DETAILED ROUTER

postRoute.def

design.lef, postRoute.def

LAYOUT FINISHING

postLF.def

GDS

PARASITIC EXTRACTION STATIC TIMING ANALYSIS

Testcases and libraries

Processor Designs

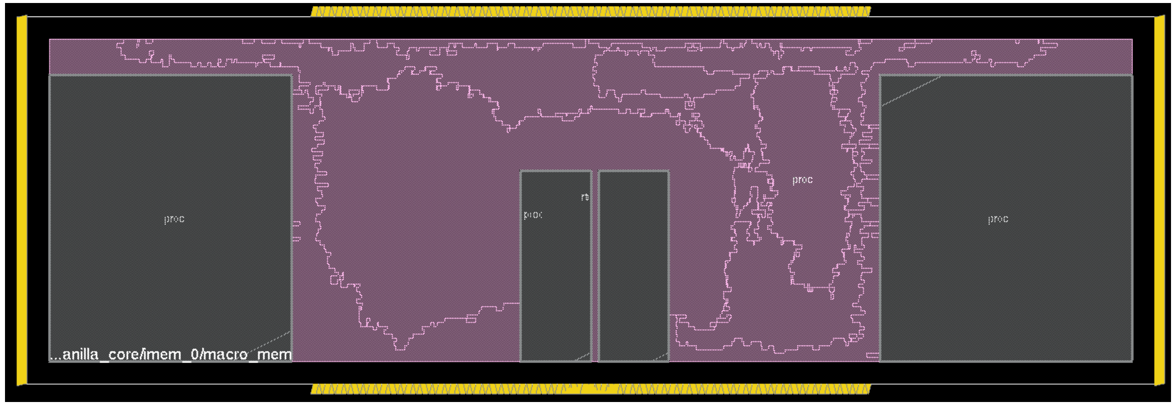
- Ariane – RISC-V based core from ETH
- Vanilla Bean – RISC-V based core from Michigan
- Black Parrot – Processor core from Washington
- Coyote – RISC-V based core from Michigan
- Sparc Core – Sparc based core from Princeton
- Cortex-A53 – Commercial core from Arm

Accelerator Designs

- aes
- gcd
- jpeg
- dynamic_node

Test Case Characteristics

- Number of cells: 200 – 250,000
- Total cell area: 1,000 – 1,301,194 um^2
- Design Utilization: 19% - 65%
- Memory/Macros: 0 – 26
- Library: TSMC 65 nm LP

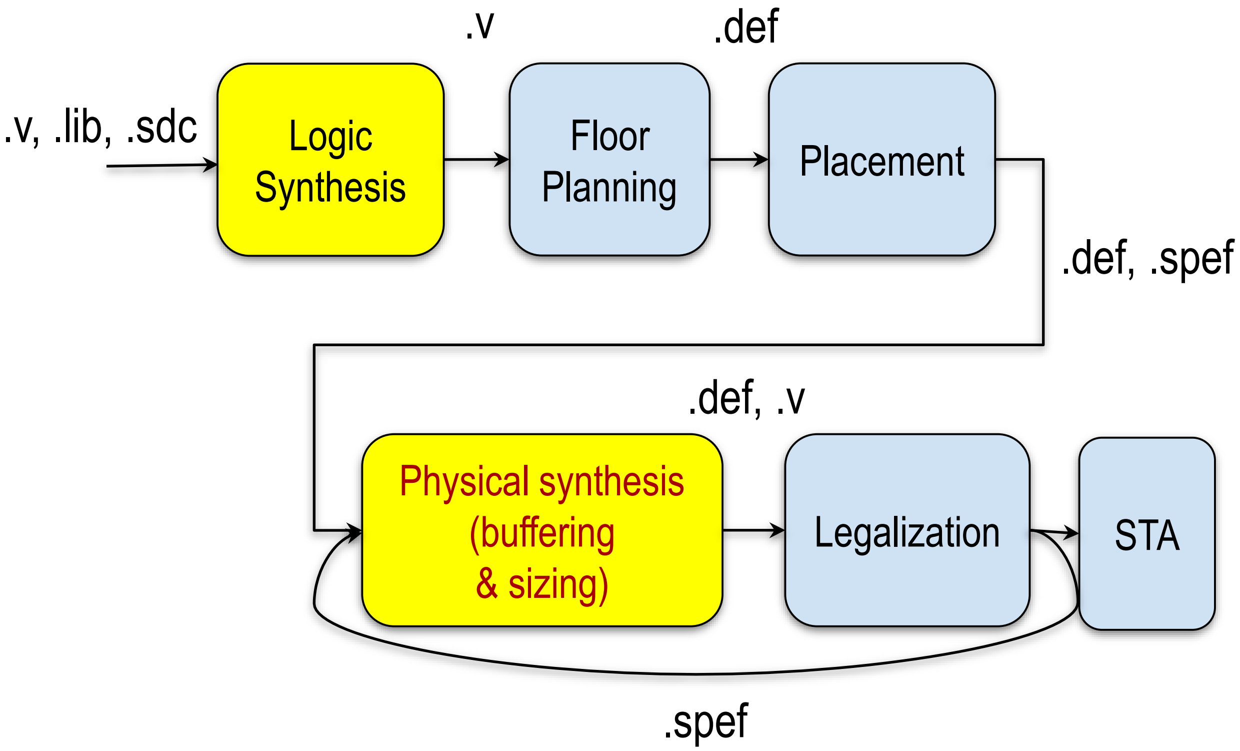


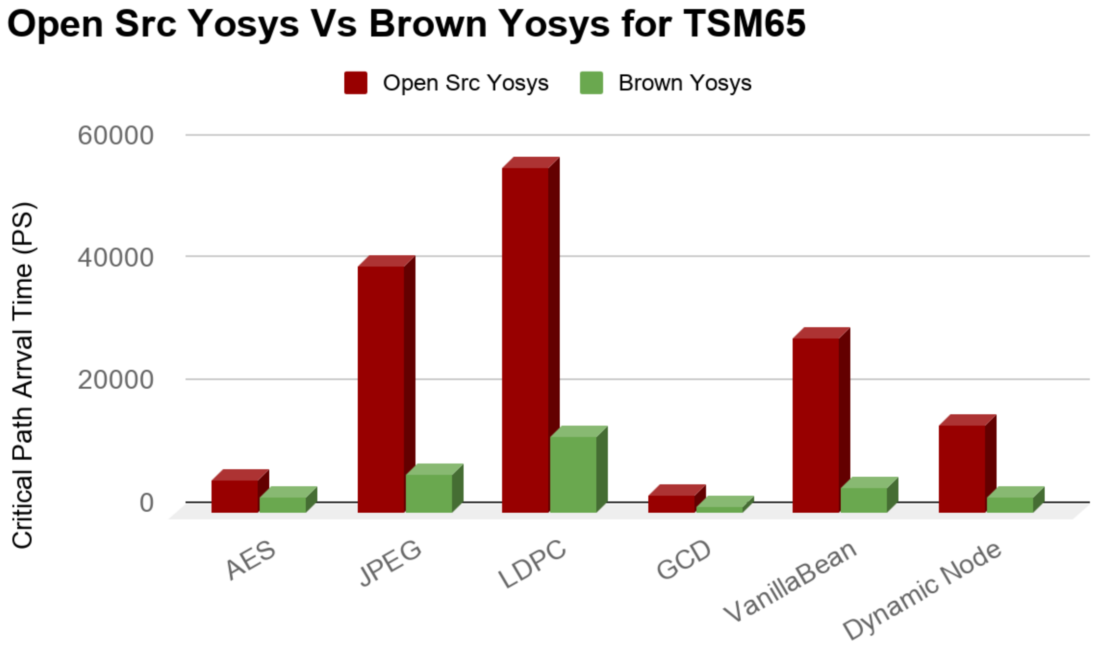
Vanilla Bean

1. Logic and Physical Synthesis

- Builds on open-source yosys-abc and adds large support for timing-driven synthesis: (1) physical synthesis, (2) SDC support, and (3) flow optimization

1. Physical synthesis (buffering, gate sizing, cloning, logic restructuring, etc)





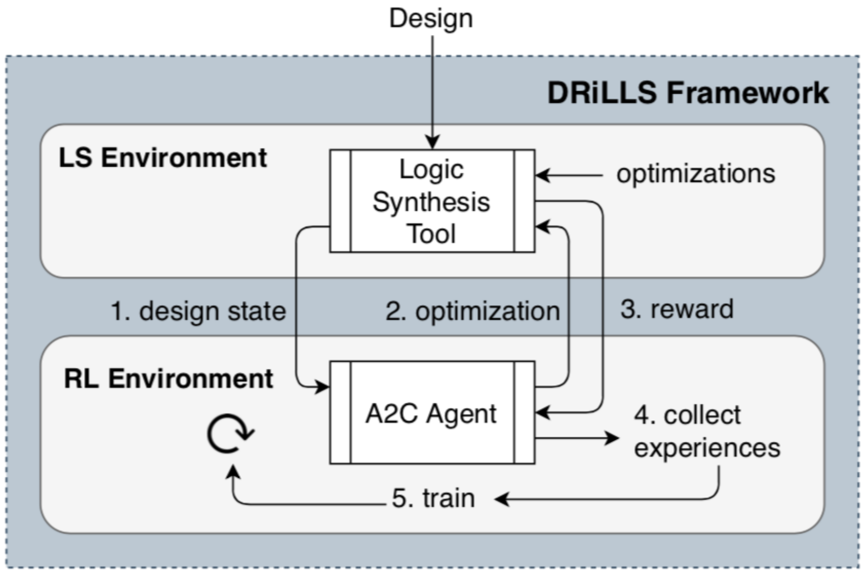
1. Logic and Physical Synthesis

2. SDC constraints support

- Integrated an open source SDC Parser in Yosys/ABC
- Add support for some SDC commands in ABC: create_clock, set_load, set_max_fanout, set_max_transition, etc.
- Roadmap for SDC command implementation (e.g., set_driving_cell, set_input_delay, set_output_delay, set_min_delay,…)

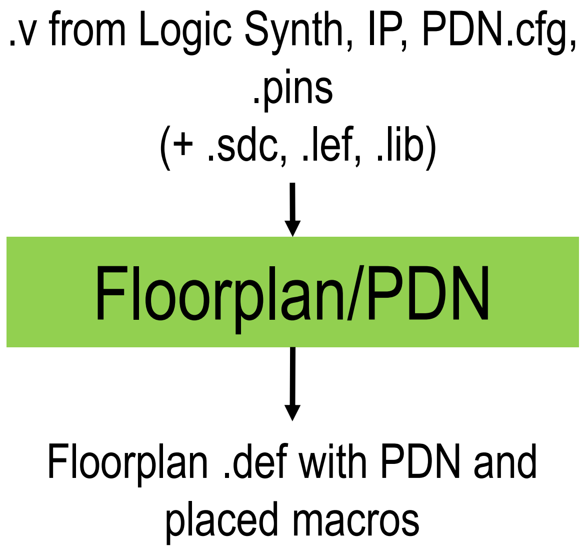
3. Automated synthesis scripting

- Synthesis scripts have 10s-100s of commands crafted by experts
- Approach: Devised a reinforcement learning controller for synthesis.
- Results: ~14% improvement in area and meets timing constraints.



2. Floorplan / Powerplan

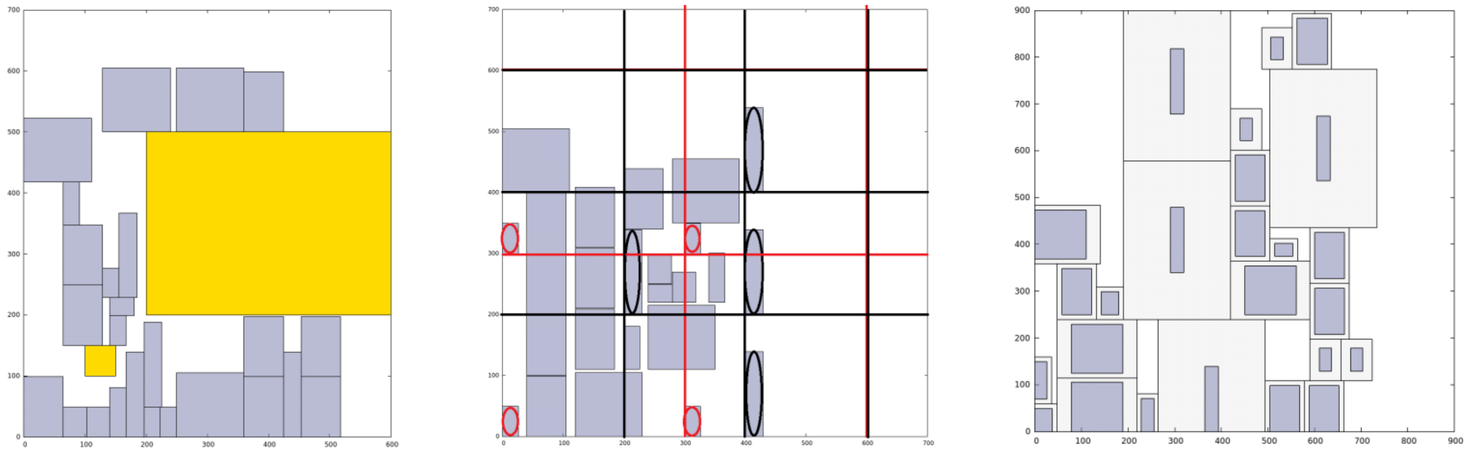
- **Macro packing** evolved from UMichigan Parquet B*-tree-based simulated annealing
  - Halos, channels, etc. rules captured in "config" files
- PDN generation along with endcap, tapcell layout also part of tool
- Verilog to DEF brings synthesis outputs
  - Link: https://github.com/abk-openroad/OpenROAD-Utilities



.v from Logic Synth, IP, PDN.cfg, .pins
(+ .sdc, .lef, .lib)

Floorplan/PDN

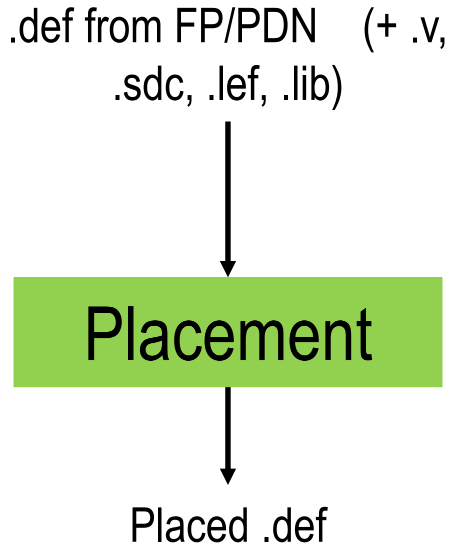
Floorplan .def with PDN and placed macros



TCAD '03: Fixed-outline floorplanning: Enabling hierarchical design

3. Placement

- RePlAce features
  - Timing-driven (OpenSTA timer integrated)
  - Mixed-size (macros + cells)
  - Electrostatics analogy in analytic placement
- RePlAce used in:
  - Physical synthesis
  - Floorplanning
  - Clock tree synthesis
  - Traditional standard-cell placement
- BSD-3 License

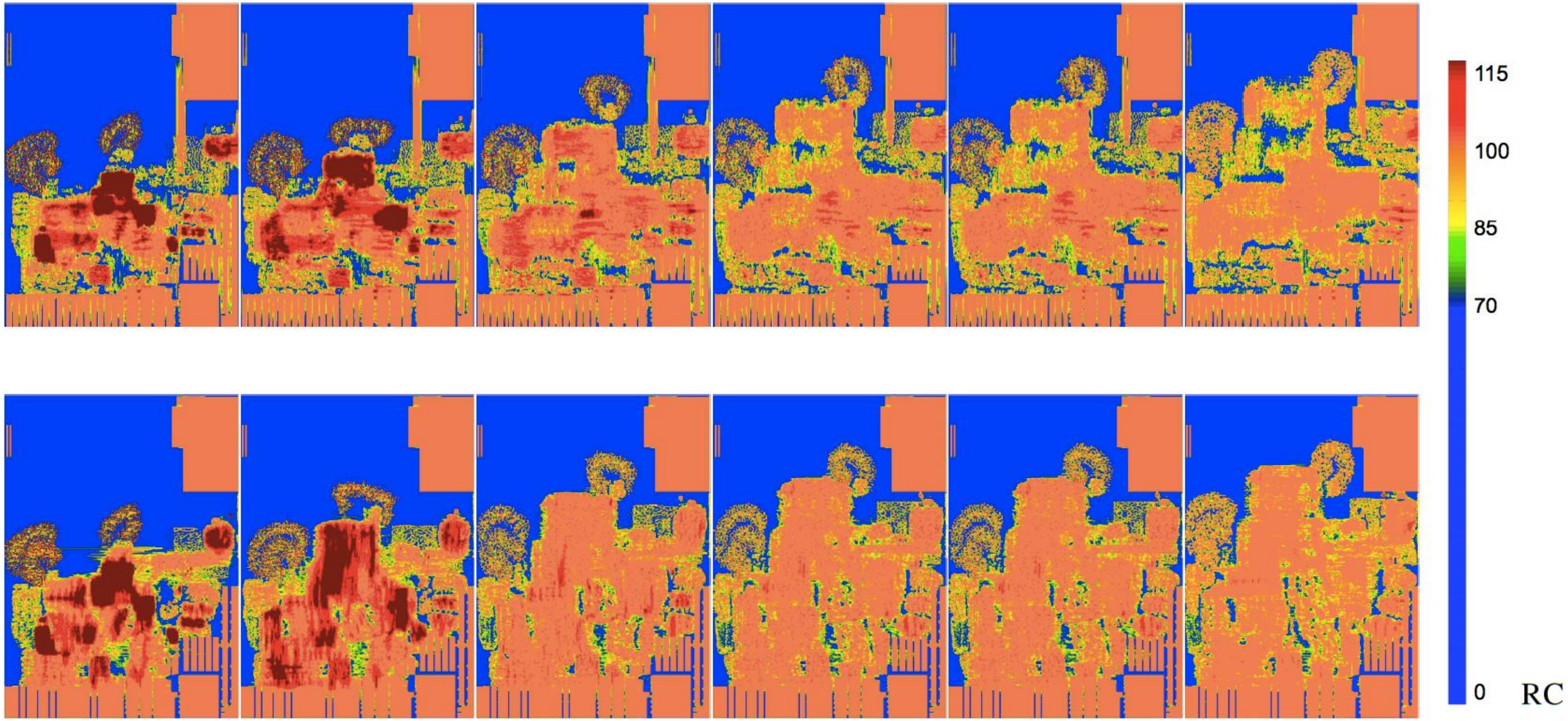


https://github.com/abk-openroad/RePlAce

TCAD'18: RePlAce: Advancing Solution Quality and Routability Validation in Global Placement

3. Placement

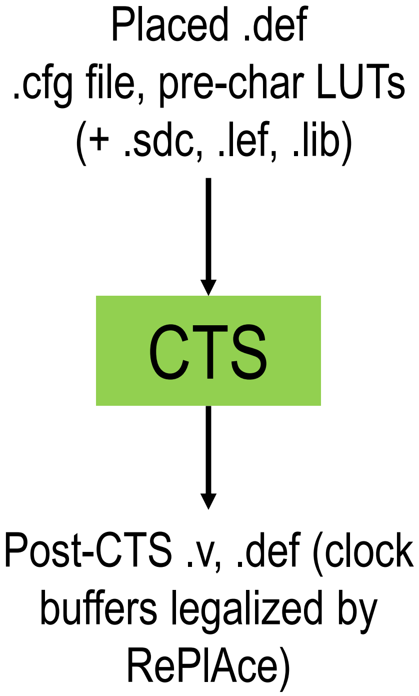
Global routing during routability-driven global placement



Routability-driven loop

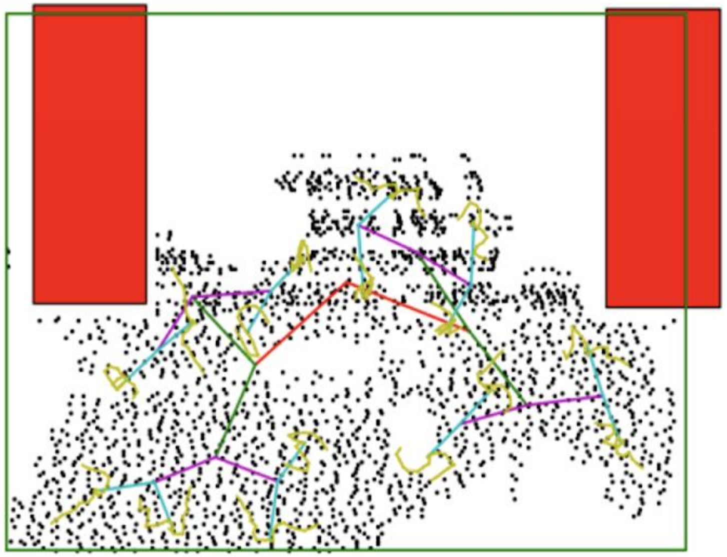
4. Clock Tree Synthesis

- Evolved from GH-Tree [Han et al., TCAD18] Generalized H-Tree approach
- All commercial tool dependencies in original implementation removed
- One-time technology, library characterization uses OpenSTA
- Sink clustering using capacitated k-means (ILP formulation removed)
- Heuristic to handle segment length constraints and buffer placement


Placed .def
.cfg file, pre-char LUTs
(+ .sdc, .lef, .lib)

CTS

Post-CTS .v, .def (clock buffers legalized by RePlAce)

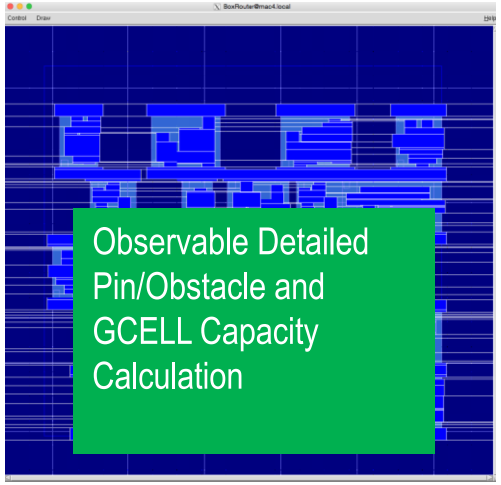


Clock tree for UM VanillaBean, 16FFC

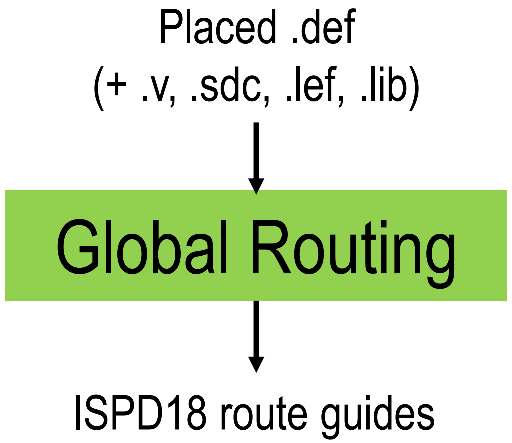
TCAD'18: Optimal Generalized H-Tree Topology and Buffering for High-Performance and Low-Power Clock Distribution

5. Global Routing

- Adaption of BoxRouter 2.0 Cho et al. UT Austin
- LP-based 3D Global Router (glpk)
- GCELL model based
- LEF/DEF input format
- Detailed obstacle and pin support
- Generates ISPD guide format



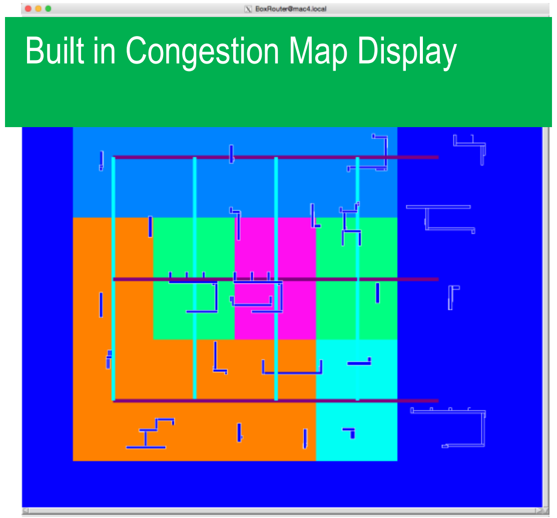
Observable Detailed Pin/Obstacle and GCELL Capacity Calculation



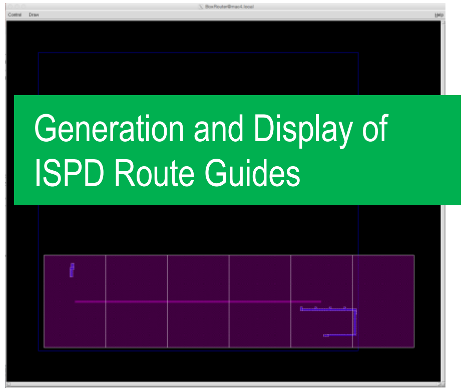
Placed .def
(+ .v, .sdc, .lef, .lib)

Global Routing

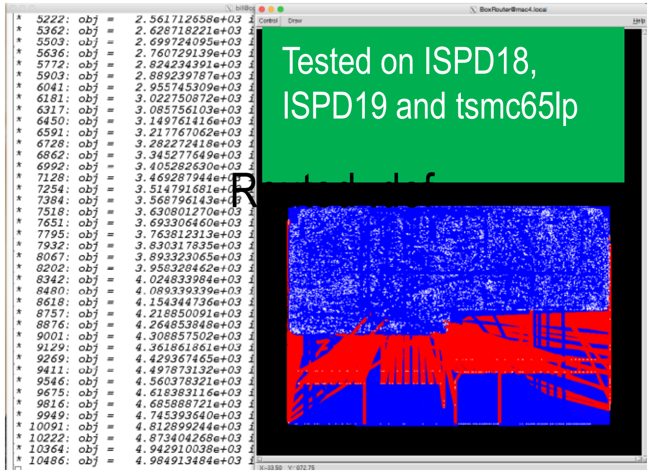
ISPD18 route guides



Built in Congestion Map Display



Generation and Display of ISPD Route Guides



```
*   5222:  obj  =   2.561712658e+03  i
*   5362:  obj  =   2.628718221e+03  i
*   5503:  obj  =   2.699724095e+03  i
*   5636:  obj  =   2.760729139e+03  i
*   5772:  obj  =   2.824234391e+03  i
*   5903:  obj  =   2.889239787e+03  i
*   6041:  obj  =   2.955745309e+03  i
*   6181:  obj  =   3.022750872e+03  i
*   6317:  obj  =   3.085756103e+03  i
*   6450:  obj  =   3.149761416e+03  i
*   6591:  obj  =   3.217767062e+03  i
*   6728:  obj  =   3.282272418e+03  i
*   6862:  obj  =   3.345277649e+03  i
*   6992:  obj  =   3.405282630e+03  i
*   7128:  obj  =   3.469287944e+03
*   7254:  obj  =   3.514791681e+0
*   7384:  obj  =   3.568796143e+03
*   7518:  obj  =   3.630801270e+03  i
*   7651:  obj  =   3.693306460e+03  i
*   7795:  obj  =   3.763812313e+03  i
*   7932:  obj  =   3.830317835e+03  i
*   8067:  obj  =   3.893323065e+03  i
*   8202:  obj  =   3.958328462e+03  i
*   8342:  obj  =   4.024833984e+03  i
*   8480:  obj  =   4.089339339e+03  i
*   8618:  obj  =   4.154344736e+03  i
*   8757:  obj  =   4.218850091e+03  i
*   8876:  obj  =   4.264853848e+03  i
*   9001:  obj  =   4.308857502e+03  i
*   9129:  obj  =   4.361861861e+03  i
*   9269:  obj  =   4.429367465e+03  i
*   9411:  obj  =   4.497873132e+03  i
*   9546:  obj  =   4.560378321e+03  i
*   9675:  obj  =   4.618383116e+03  i
*   9816:  obj  =   4.685888721e+03  i
*   9949:  obj  =   4.745393640e+03  i
*  10091:  obj  =   4.812899244e+03  i
*  10222:  obj  =   4.873404268e+03  i
*  10364:  obj  =   4.942910038e+03  i
*  10486:  obj  =   4.984913484e+03  i
```

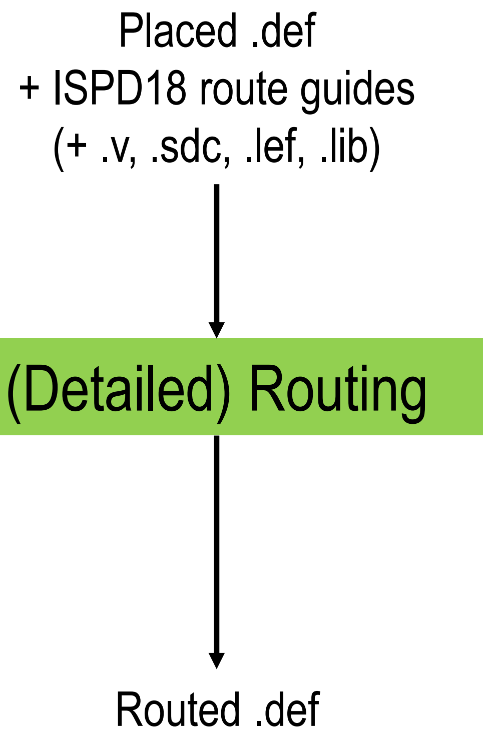
Tested on ISPD18, ISPD19 and tsmc65lp

R

ICCAD’07: BoxRouter 2.0: Architecture and implementation of a hybrid and robust global router

6. Detailed Routing

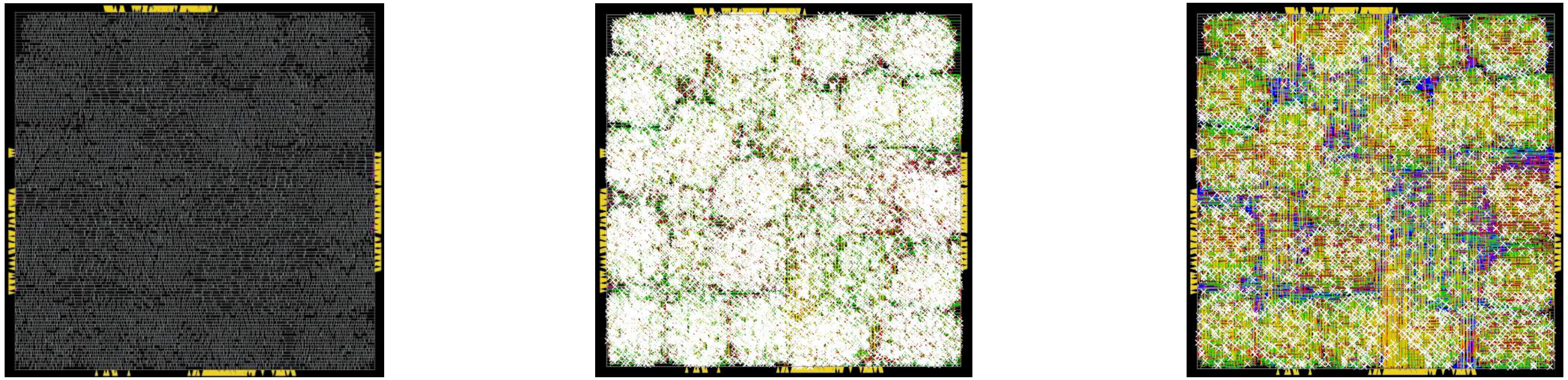
- Evolved from ISPD18 Detailed Routing contest winner [Kahng et al., ICCAD18]
  - Greedy approach (no ILP formulation) for track assignment
  - LVS-clean initial detailed route at foundry nodes
  - Violation removal accomplished with iterative search and repair (S&R)
- Many foundational components added beyond "academic contest winner"
  - DRC / quick DRC engines, pin access analysis, etc.
- Example (~April 2019) with TSMC65 foundry node
  - 10K-instance design block with ~50% utilization
  - Shown in figure: two S&R iterations reduce #violations from 50487 to 3487



Placed .def
+ ISPD18 route guides
(+ .v, .sdc, .lef, .lib)

(Detailed) Routing

Routed .def

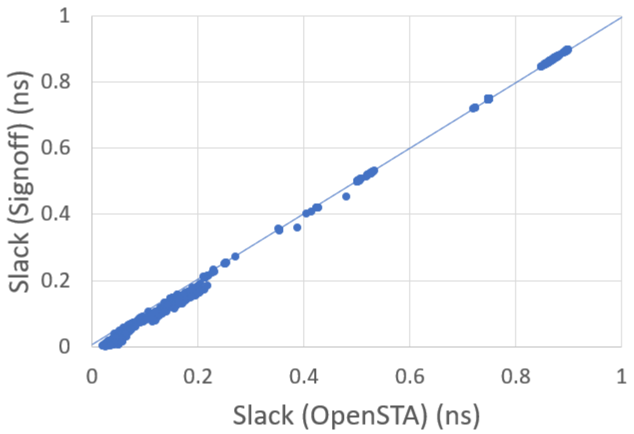


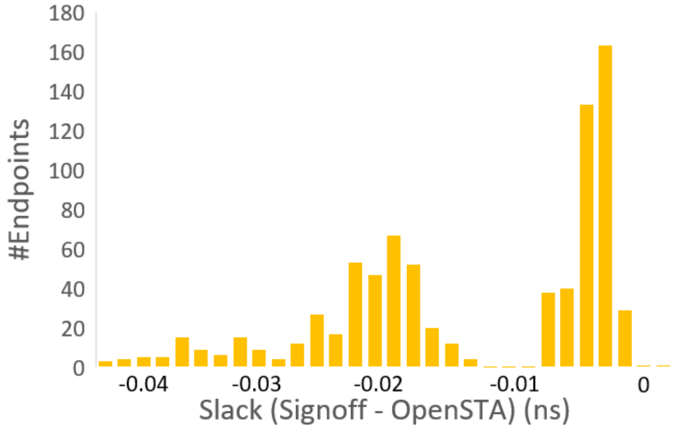
ICCAD'18: TritonRoute: An Initial Detailed Router for Advanced VLSI Technologies

Analysis tools: 1. Static Timing Analysis

- OpenSTA: open-sourced static timing analysis tool
  - Developer: James Cherry (Parallax Software)
  - Tested with ASAP7, GF14, TSMC16, ST28, etc.
  - Integrated into several aspects of the tool
  - GPLv3 license

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



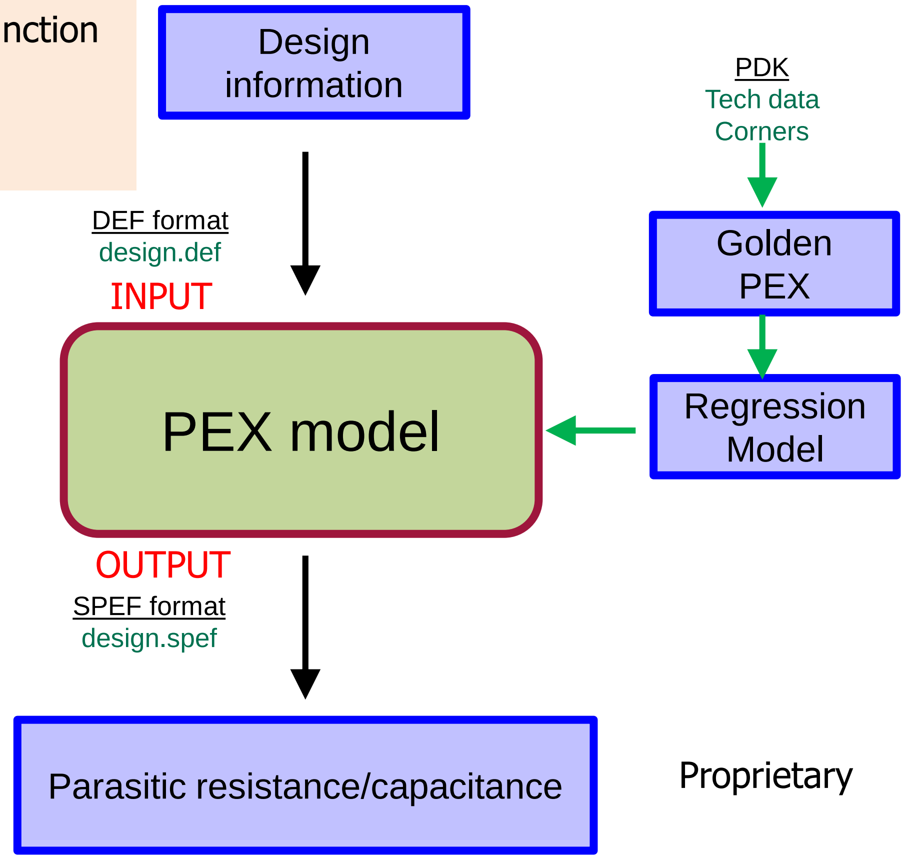


aes_cipher_top	WNS (ps)	TNS (ps)	#viol.
Signoff STA	-61	-289	7
OpenSTA (arnoldi)	-57	-314	9

https://github.com/abk-openroad/OpenSTA

Analysis tools: 2. Parasitic Extraction

PEX is a callable function
<u>Calling function</u>
STA, FP, PL,
CTS, PSI



Design information

<u>DEF format</u>
design.def
INPUT

<u>PDK</u>
Tech data
Corners

Golden PEX

Regression Model

PEX model

OUTPUT
<u>SPEF format</u>
design.spef

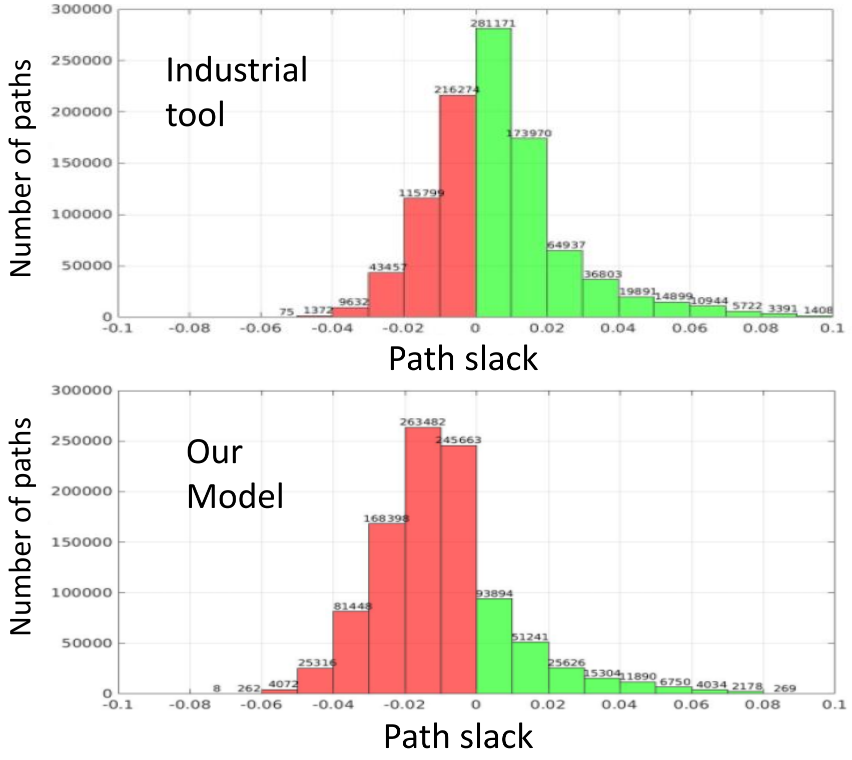
Parasitic resistance/capacitance

Proprietary

Results of comparing regression model SPEF with industry-tool SPEF on the AES design (CMP28 PDK)

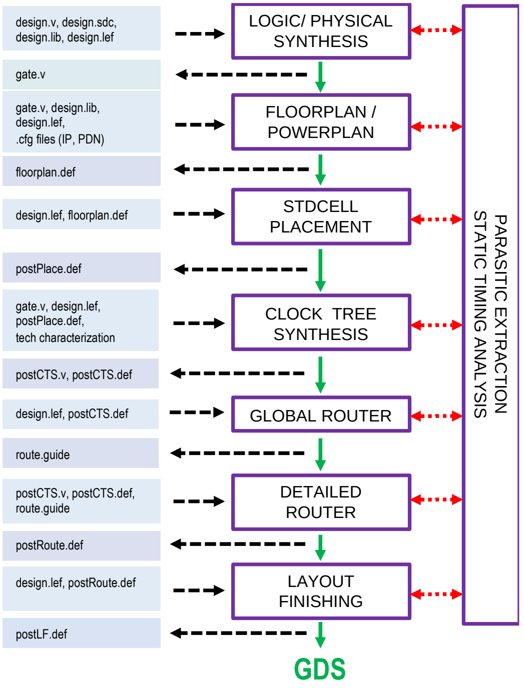
- Total R, C: Mean error = -3% (<u>pessimistic</u>)
- Cell delay [all timing arcs] (CMP28): Mean error = -4% (<u>pessimistic</u>)

Path slack distribution



Conclusions

- OpenROAD = digital layout, no-humans, 24-hours, open source
- Open source (FOSS ecosystem, etc.) restores productivity and attractiveness of academic innovation
- OpenRoad's first year focus is developing the basic design flow using 65 nm
- Second year will focus on database development, support of more advanced nodes, machine-learning driven optimizations, runtime improvements
- Alpha release in late July 2019 (not available till then)
  - https://github.com/The-OpenROAD-Project
  - https://flow.theopenroadproject.org

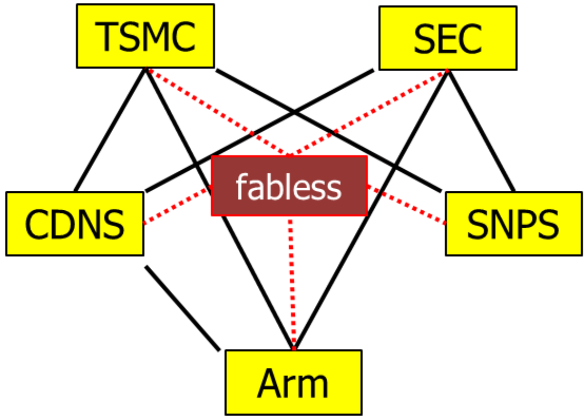

design.v, design.sdc, design.lib, design.lef
LOGIC/ PHYSICAL SYNTHESIS
gate.v
gate.v, design.lib, design.lef, .cfg files (IP, PDN)
FLOORPLAN / POWERPLAN
floorplan.def
design.lef, floorplan.def
STDCELL PLACEMENT
postPlace.def
gate.v, design.lef, postPlace.def, tech characterization
CLOCK TREE SYNTHESIS
postCTS.v, postCTS.def
design.lef, postCTS.def
GLOBAL ROUTER
route.guide
postCTS.v, postCTS.def, route.guide
DETAILED ROUTER
postRoute.def
design.lef, postRoute.def
LAYOUT FINISHING
postLF.def
GDS
PARASITIC EXTRACTION
STATIC TIMING ANALYSIS

THANK YOU!

Open-Source Birds-of-a-Feather DAC meeting Wed 7pm N256
2nd WOSET co-located with ICCAD 2019

Why Open Source is Critical 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

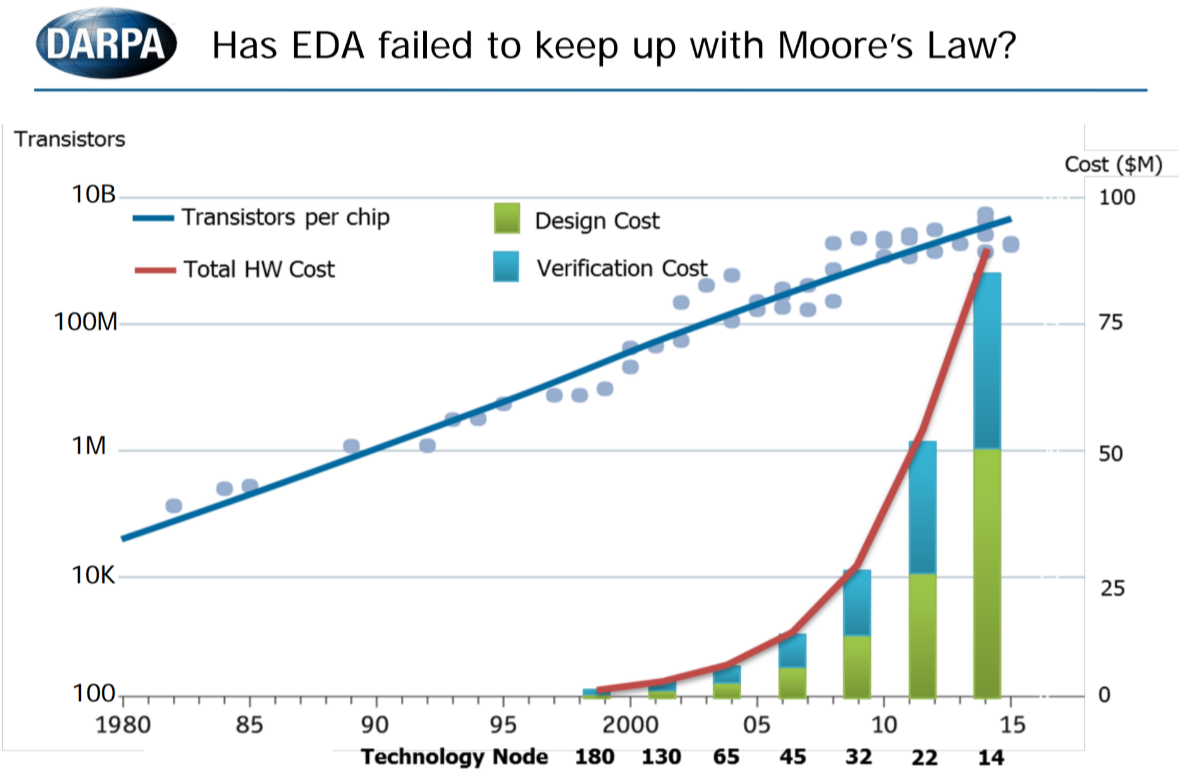


- Foundry: protect IP leakage, roadmap better than current DTCO mechanisms
- Fabless: protect key SOC methodology, design innovation that otherwise leaks to competitors
- EDA: more useful research, in more usable forms
- Academia: see previous slides (and, engage … or not)

Design Crises: Cost, Expertise, Unpredictability

Design cost: not scaling

- Design, process roadmaps not coupled

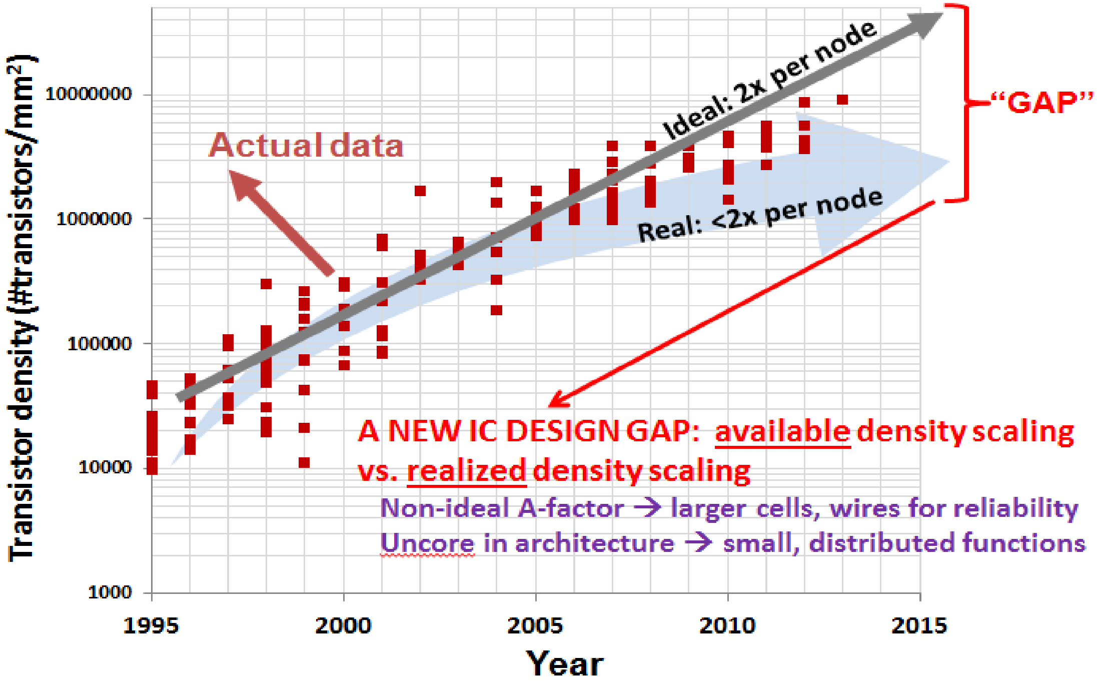

Has EDA failed to keep up with Moore's Law?

[Figure: Andreas Olofsson, DARPA, ISPD-2018 keynote]

Quality: also not scaling

Available density: 2x/node

Realizable density: 1.6x/node



A NEW IC DESIGN GAP: available density scaling vs. realized density scaling

Non-ideal A-factor → larger cells, wires for reliability

Uncore in architecture → small, distributed functions

Figure: UCSD / 2013 ITRS