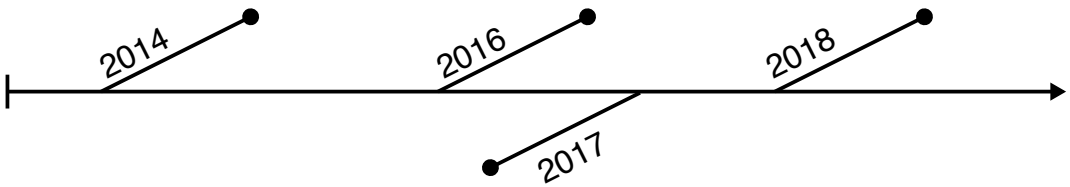
A New Era of Silicon Prototyping in Computer Architecture Research

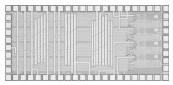
Christopher Torng

Computer Systems Laboratory
School of Electrical and Computer Engineering
Cornell University

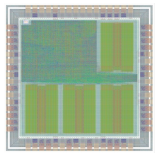
Recent History of Prototypes at Cornell University



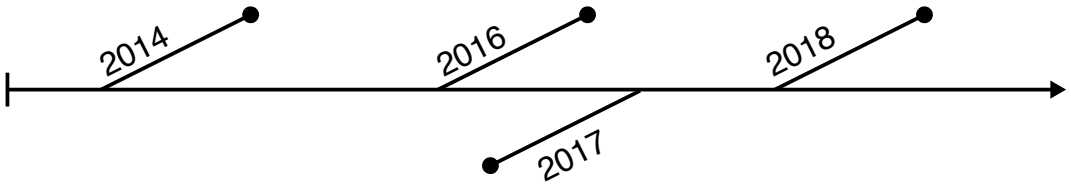
Recent History of Prototypes at Cornell University



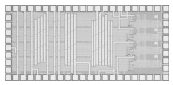
DCS (2014)
TSMC 65nm
1mm x 2.2mm



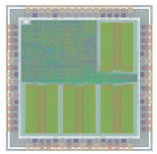
BRGTC1 (2016)
IBM 130nm
2mm x 2mm



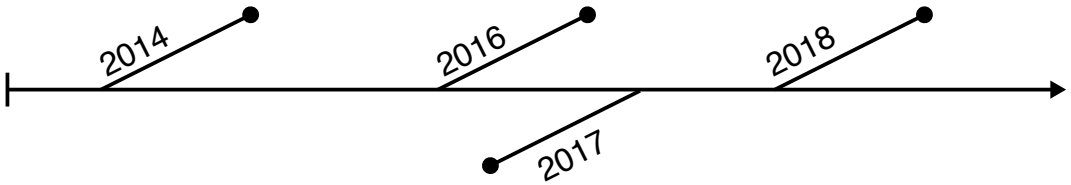
Recent History of Prototypes at Cornell University

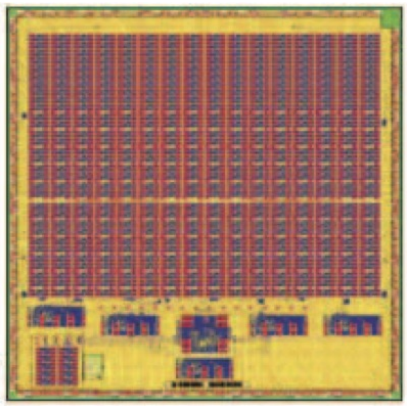


DCS (2014)
TSMC 65nm
1mm x 2.2mm



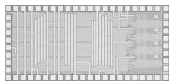
BRGTC1 (2016)
IBM 130nm
2mm x 2mm



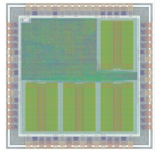


Celerity (2017)
TSMC 16nm FinFET
5mm x 5mm

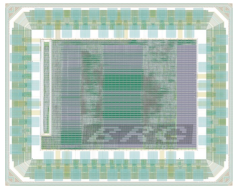
Recent History of Prototypes at Cornell University



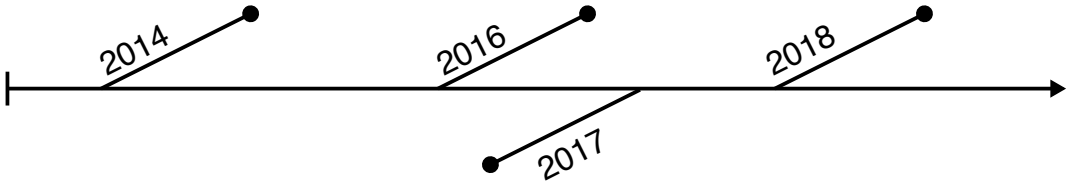
DCS (2014)
TSMC 65nm
1mm x 2.2mm

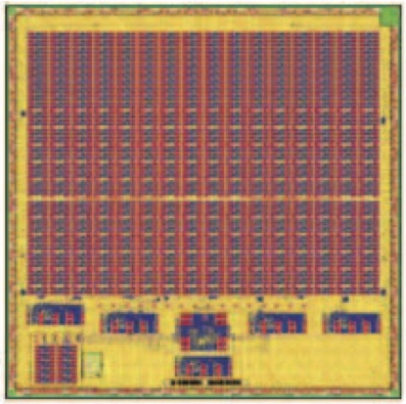


BRGTC1 (2016)
IBM 130nm
2mm x 2mm



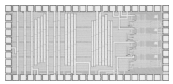
BRGTC2 (2018)
TSMC 28nm
1mm x 1.25mm



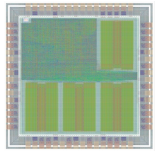


Celerity (2017)
TSMC 16nm FinFET
5mm x 5mm

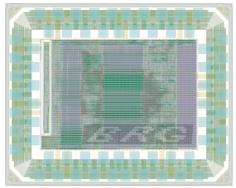
Recent History of Prototypes at Cornell University



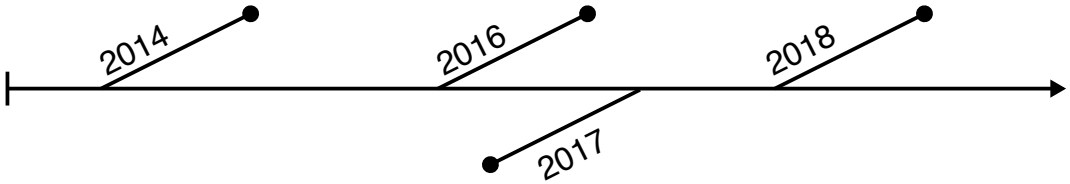
DCS (2014)
TSMC 65nm
1mm x 2.2mm

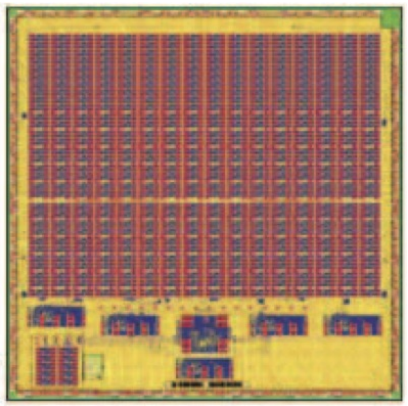


BRGTC1 (2016)
IBM 130nm
2mm x 2mm

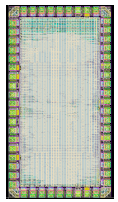


BRGTC2 (2018)
TSMC 28nm
1mm x 1.25mm



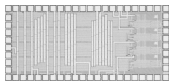


Celerity (2017)
TSMC 16nm FinFET
5mm x 5mm

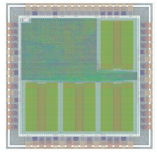


PCOSYNC (2018)
IBM 180nm
2mm x 1mm

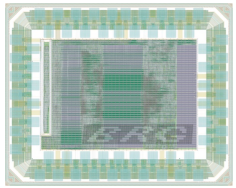
Recent History of Prototypes at Cornell University



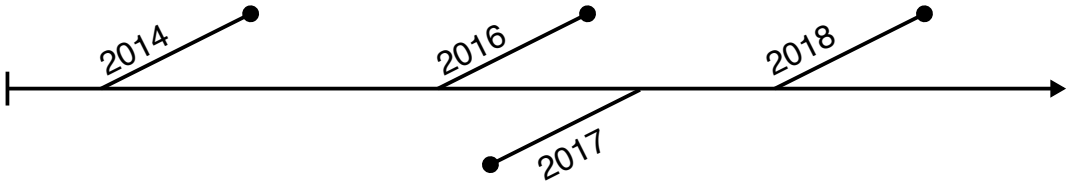
DCS (2014)
TSMC 65nm
1mm x 2.2mm

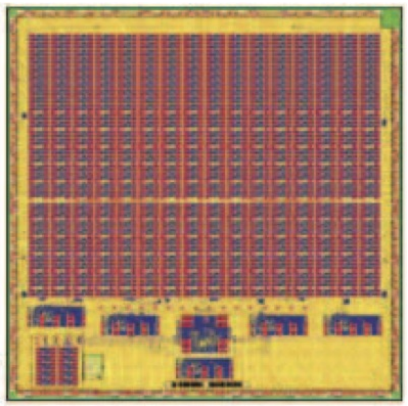


BRGTC1 (2016)
IBM 130nm
2mm x 2mm

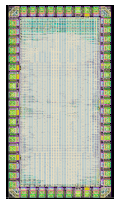


BRGTC2 (2018)
TSMC 28nm
1mm x 1.25mm





Celerity (2017)
TSMC 16nm FinFET
5mm x 5mm



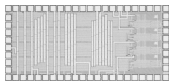
PCOSYNC (2018)
IBM 180nm
2mm x 1mm

Why Prototype?

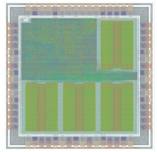
Research Ideas

- Smart Sharing Architectures
- Interconnection Networks for Manycores
- Python-Based Hardware Modeling
- High-Level Synthesis
- Synthesizable Analog IP
- Scalable Baseband Synchronization
- Integrated Voltage Regulation

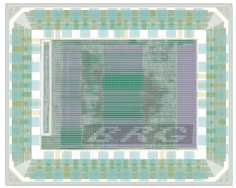
Recent History of Prototypes at Cornell University



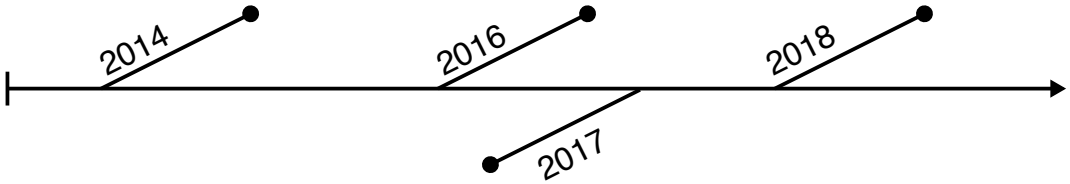
DCS (2014)
TSMC 65nm
1mm x 2.2mm

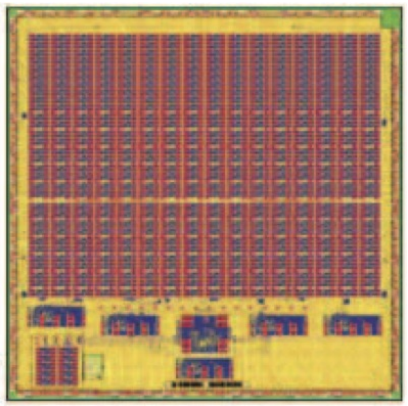


BRGTC1 (2016)
IBM 130nm
2mm x 2mm

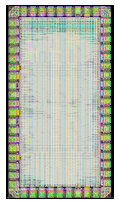


BRGTC2 (2018)
TSMC 28nm
1mm x 1.25mm





Celerity (2017)
TSMC 16nm FinFET
5mm x 5mm



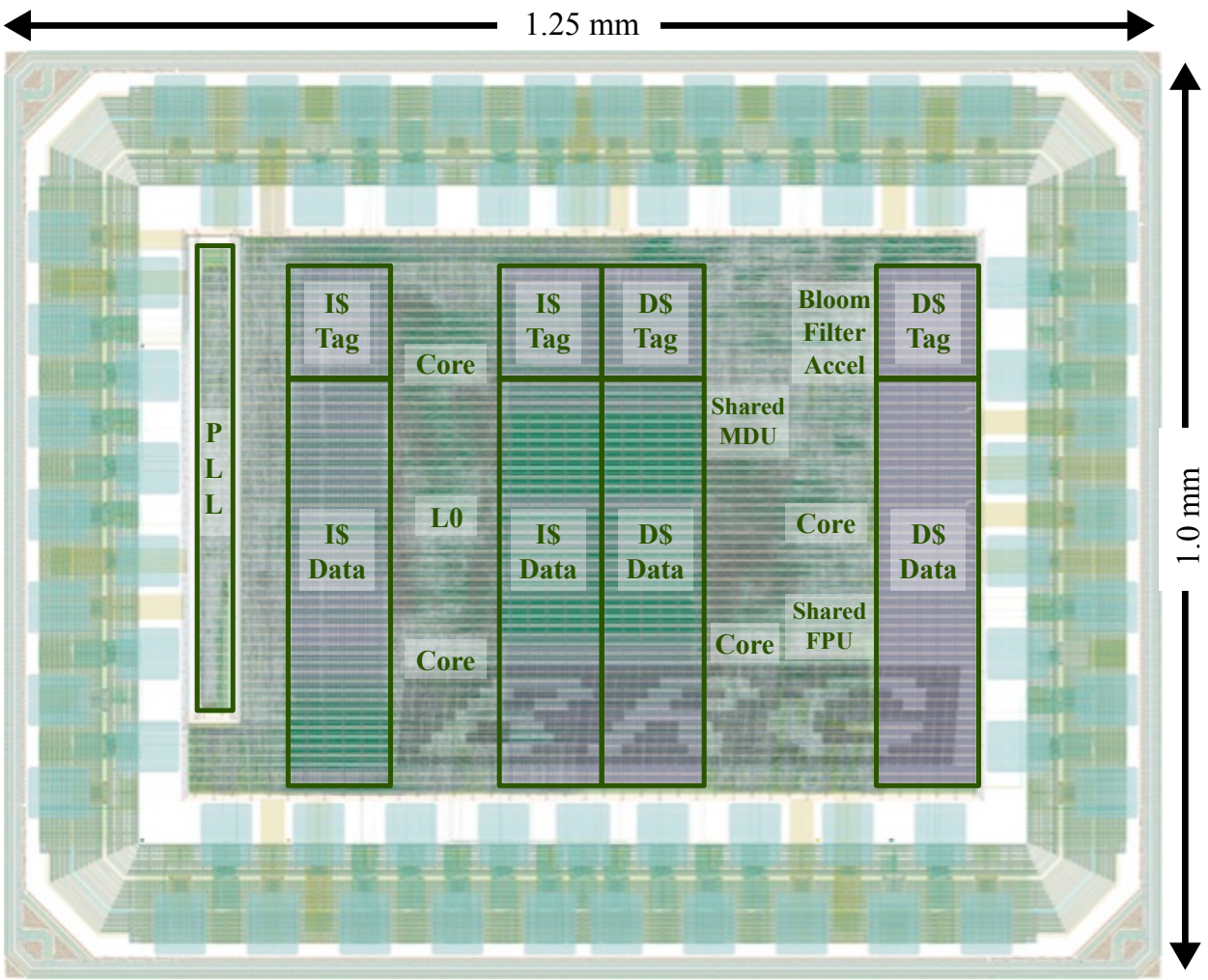
PCOSYNC (2018)
IBM 180nm
2mm x 1mm

Why Prototype?

Chip-Based Startups

- Graphcore
- Nervana
- Cerebras
- Wave Computing
- Horizon Robotics
- Cambricon
- DeePhi
- Esperanto
- SambaNova
- Eyeriss
- Tenstorrent
- Mythic
- ThinkForce
- Groq
- Lightmatter

BRGTC2 — Batten Research Group Test Chip 2



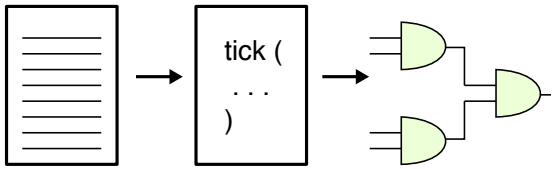
Chip Overview

- TSMC 28 nm
- 1 mm × 1.25 mm
- 6.7M-transistor
- Quad-core in-order RISC-V RV32IMAF
- Shared L1 caches (32kB) Shared LLFUs
- Designed and tested in PyMTL (Python-based hardware modeling)
- Fully synthesizable PLL
- Smart sharing mechanisms
- Hardware bloom filter xcel
- Runs work-stealing runtime

Key Changes Driving A New Era

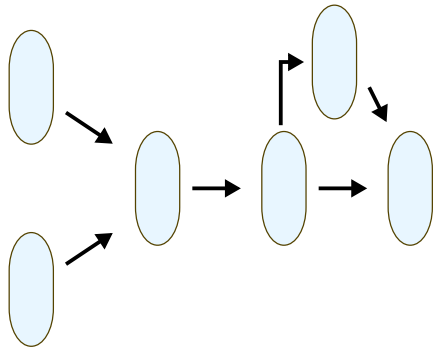
Ecosystems for Open Builders

Problem: Closed tools & IP makes dev tough
Changes: Open-source ecosystem with RISC-V



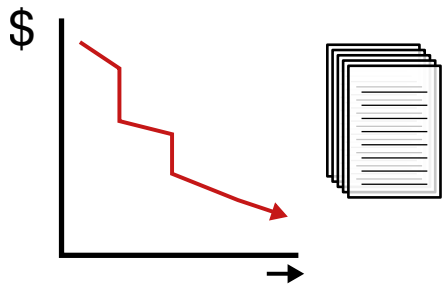
Productive Tools for Small Teams

Problem: Small teams with a limited workforce
Changes: Productive & open tool development



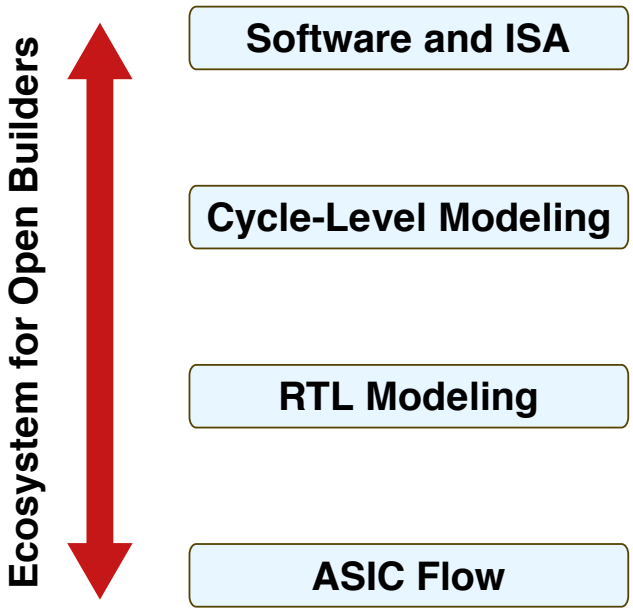
Significantly Cheaper Costs

Problem: Building chips is expensive
Changes: MPW tiny chips in advanced nodes



Ecosystems for Open Builders

Problem: A closed-source chip-building ecosystem (tools & IP) makes chip development tough



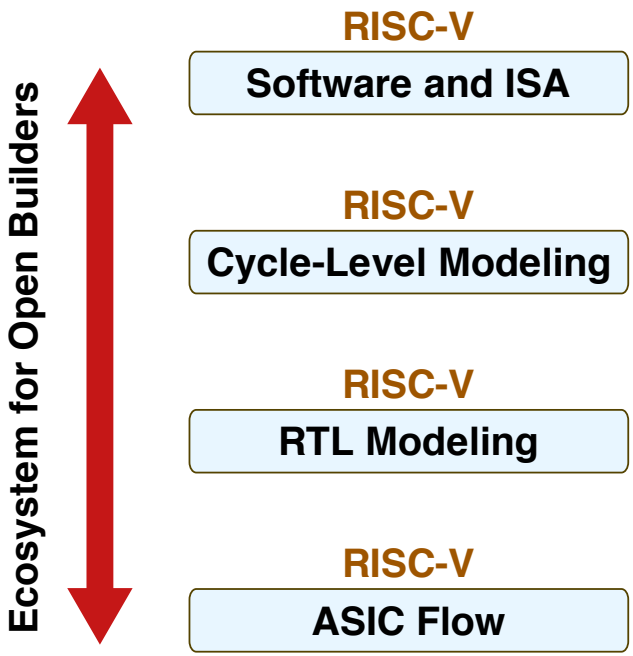
Problems with Closed-Source Infrastructure

- Difficult to replicate results (including your own)
- Anything closed-source propagates up and down the stack
  - E.g., modified MIPS ISA
  - Spill-over to other stages of the design flow
- Heavy impact on things I care about
  - Sharing results and artifacts
  - Portability
  - Maintenance
- Reinventing the wheel

How important is a full ecosystem?

Ecosystems for Open Builders

Key Change: The open-source ecosystem revolving around RISC-V is growing

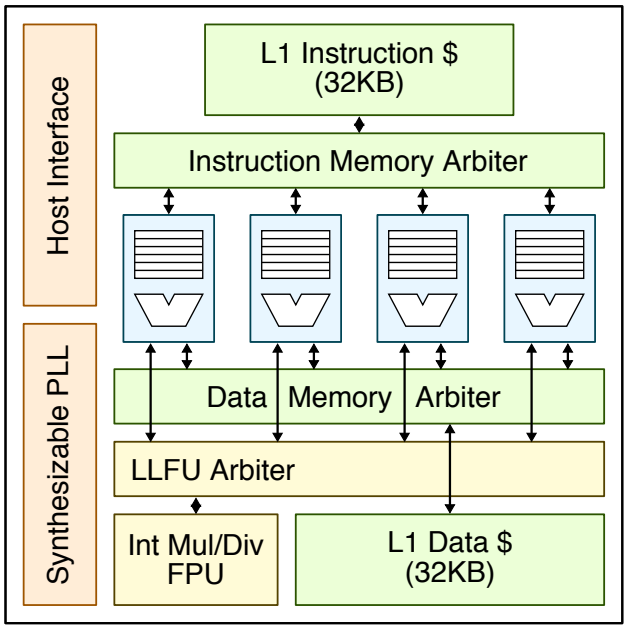


The RISC-V Ecosystem

- Software toolchain and ISA
  - Linux, compiler toolchain, modular ISA
- Cycle-level modeling
  - gem5 system-level simulator supports RISC-V multicore
  - We can now model complex RISC-V systems
- RTL modeling
  - Open implementations and supporting infrastructure (e.g., Rocket, Boom, PULP, Diplomacy, FIRRTL, FireSim)
- ASIC flows
  - Reference flows available from community for inspiration

Ecosystems for Open Builders

How has the RISC-V ecosystem helped in the design of BRGTC2?



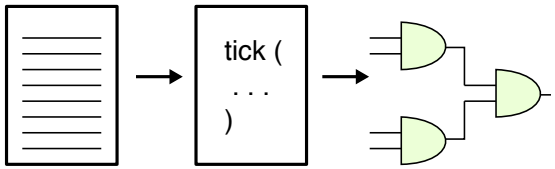
BRGTC2 in the RISC-V Ecosystem

- Software toolchain and ISA
  - Not booting Linux...
  - Upstream GCC support
  - Incremental design w/ RV32 modularity
- Cycle-level modeling
  - Multicore gem5 simulations of our system
  - *Decisions*: L0 buffers, how many resources to share, impact of resource latencies, programs fitting in the cache
- RTL modeling
  - This was our own...
- ASIC flows
  - Reference methodologies available from other projects (e.g., Celerity)

Key Changes Driving A New Era

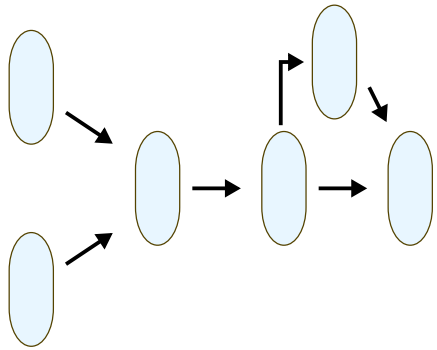
Ecosystems for Open Builders

Problem: Closed tools & IP makes dev tough
Changes: Open-source ecosystem with RISC-V



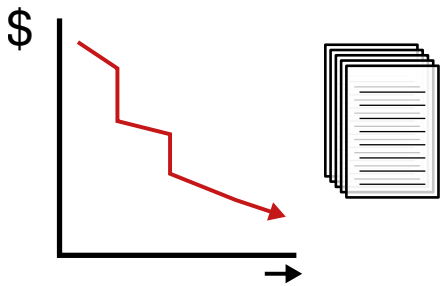
Productive Tools for Small Teams

Problem: Small teams with a limited workforce
Changes: Productive & open tool development



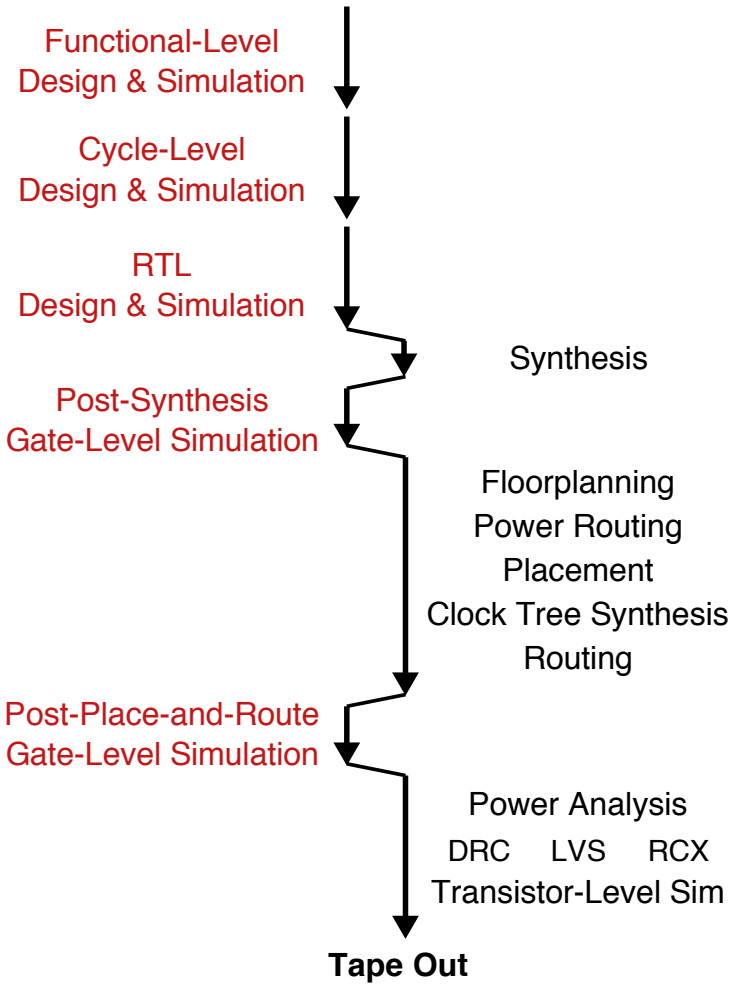
Significantly Cheaper Costs

Problem: Building chips is expensive
Changes: MPW tiny chips in advanced nodes



Productive Tools for Small Teams

Problem: Small teams have a limited workforce and yet must handle challenging projects

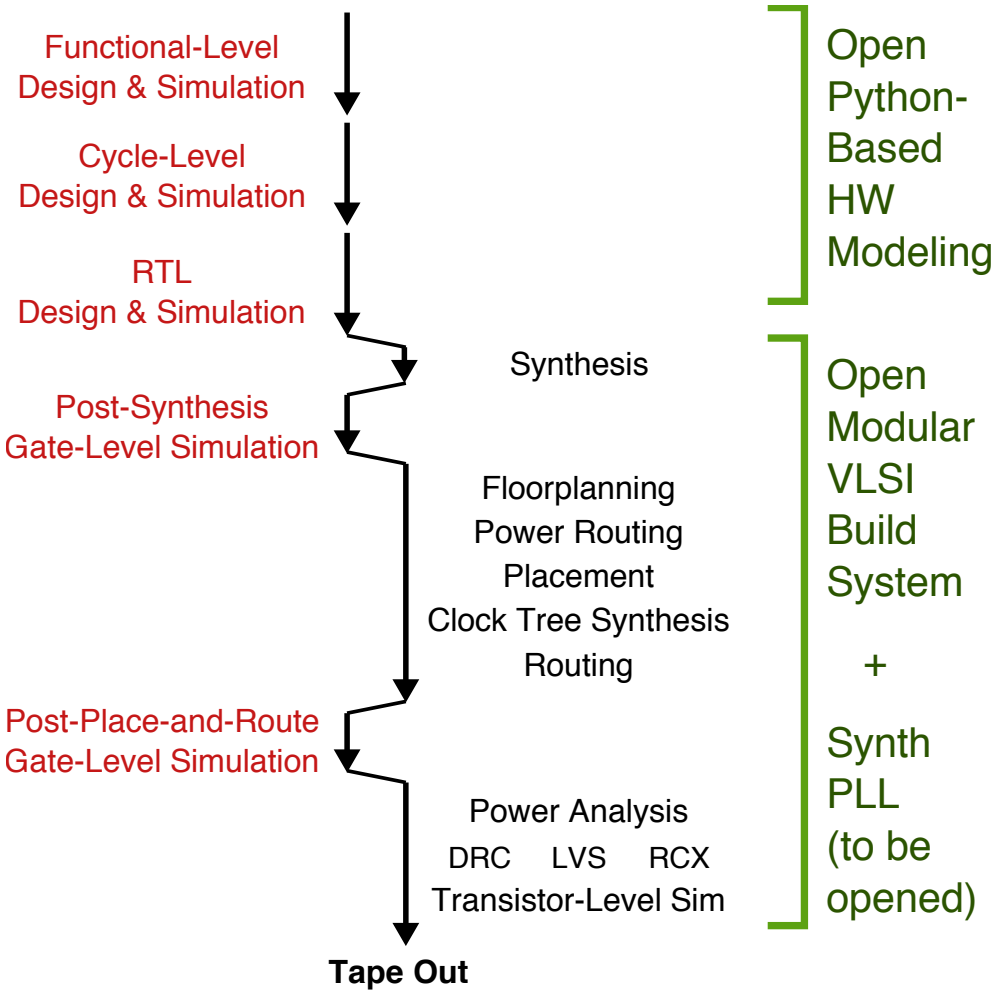


An Enormous Challenge for Small Teams

- Small teams exist in both academia as well as in industry
- Time to first tapeout can be anywhere up to a few years
- What do big companies do?
  - Throw money and engineers at the problem
- Generally stuck with tools that "work"
  - If you have enough engineers
  - E.g., System Verilog

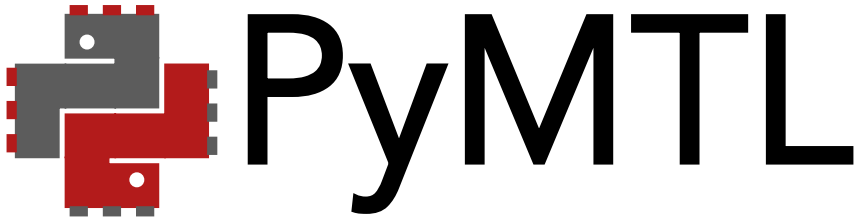
Productive Tools for Small Teams

Key Change: Productive open-source tools progressing and maturing quickly



Focusing on BRGTC2

- PyMTL Hardware Modeling Framework
  - Python-based hardware design and test
  - Beta version of PyMTL v2
  - https://github.com/cornell-brg/pymtl
- The Open Modular VLSI Build System
  - Two chips taped out (180nm/28nm)
  - Reference ASIC flow available
  - https://github.com/cornell-brg/alloy-asic
- Fully Synthesizable PLL
  - To be open-sourced soon
  - All-digital PLL used in BRGTC2/Celerity
  - Avoid mixed-signal design



PyMTL: A Unified Framework for Vertically Integrated Computer Architecture Research

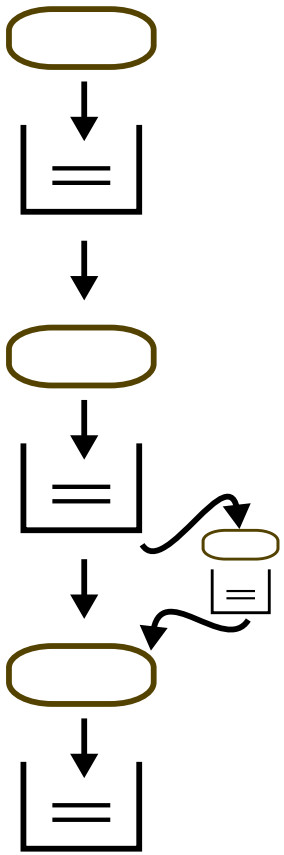
Derek Lockhart, Gary Zibrat, Christopher Batten
47th ACM/IEEE Int'l Symp. on Microarchitecture (MICRO)
Cambridge, UK, Dec. 2014

Mamba: Closing the Performance Gap in Productive Hardware Development Frameworks

Shunning Jiang, Berkin Ilbeyi, Christopher Batten
55th ACM/IEEE Design Automation Conf. (DAC)
San Francisco, CA, June 2018

Open Modular VLSI Build System – At A High Level

https://github.com/cornell-brg/alloy-asic



Problem: Rigid, static ASIC flows

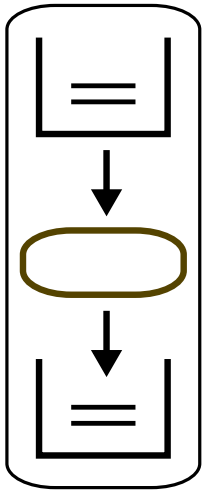
Typical ASIC Flows

- Flows are automated for exact sequences of steps
  - Want to add/remove a step? Modify the build system. Copies..
  - Once the flow is set up, you don't want to touch it anymore
- Adding new steps between existing steps is troublesome
  - Steps downstream magically reach upstream — hardcoding
  - In general, the overhead to add new steps is high
- Difficult to support different configurations of the flow
  - E.g., chip flow vs. block flow
  - How to add new steps before or after
  - Each new chip ends up with a dedicated non-reusable flow

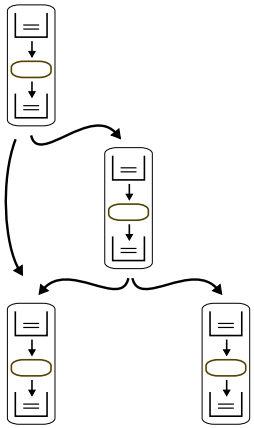
Open Modular VLSI Build System – At A High Level

https://github.com/cornell-brg/alloy-asic

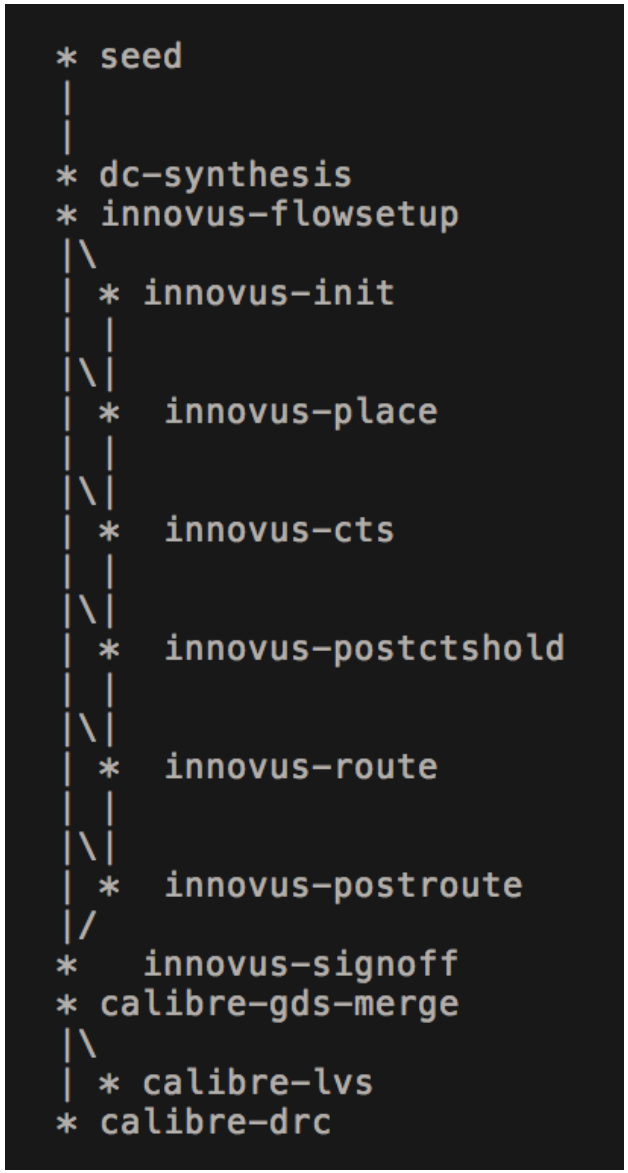
Better ASIC Flows – Modularize the ASIC flow!



- Use the build system to mix, match, and assemble steps together
  - Create modular steps that know how to run/clean themselves
  - The build system can also check prerequisites and outputs before and after execution to make sure each step can run
- Assemble the ASIC flow as a graph
  - Can target *architecture* papers by assembling a minimal graph
  - Can target *VLSI* papers by assembling a medium graph w/ more steps (e.g., need dedicated floorplan)
  - Can target a *chip* by assembling a full-featured tapeout graph



Simple Front-End-Only ASIC Flow

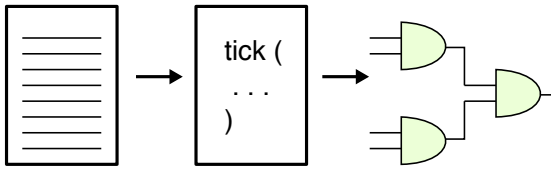


BRGTC2 ASIC Flow

Key Changes Driving A New Era

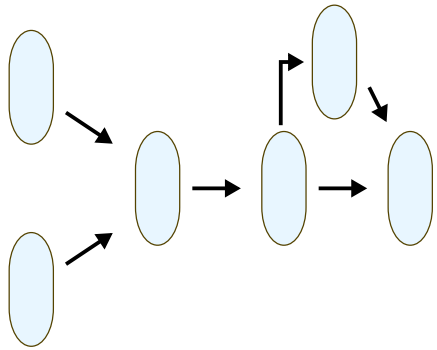
Ecosystems for Open Builders

Problem: Closed tools & IP makes dev tough
Changes: Open-source ecosystem with RISC-V



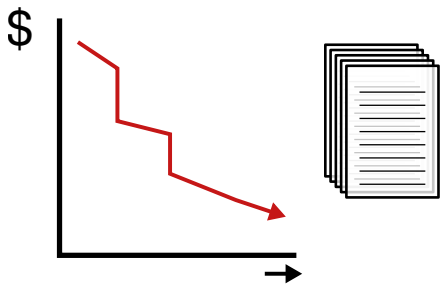
Productive Tools for Small Teams

Problem: Small teams with a limited workforce
Changes: Productive & open tool development



Significantly Cheaper Costs

Problem: Building chips is expensive
Changes: MPW tiny chips in advanced nodes



Significantly Cheaper Costs

Problem: Building chips is expensive

Key Change: Multi-project wafer services offer advanced node runs with small minimum sizes

Snapshot from Muse Semiconductor

MUSE SEMICONDUCTOR

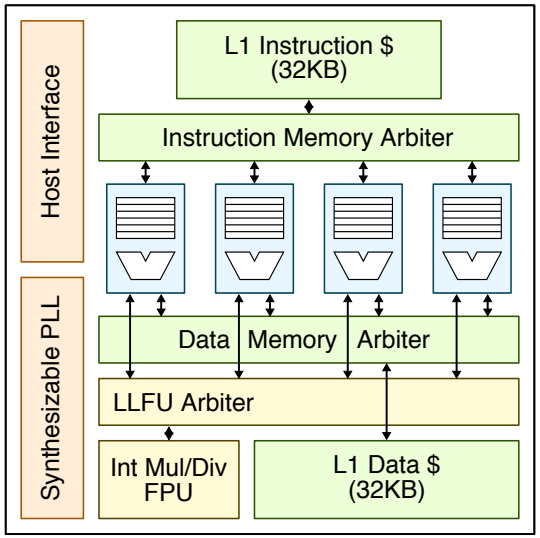
It's not just an MPW.

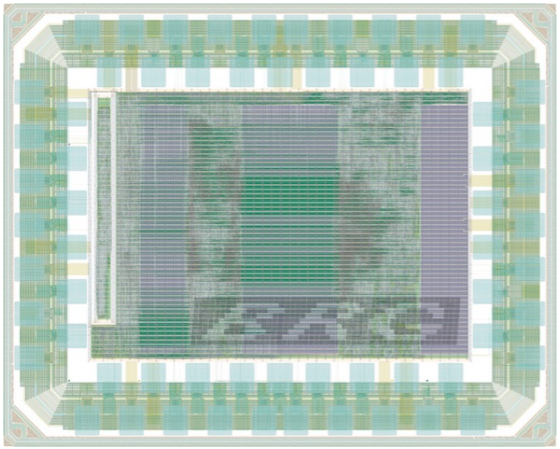
August 2018 Shared Block Tapeout Opportunities

Tech (nm)	Flavor	Min Area (mm2)	Price ($/mm2)	Trial	Final	Tapeout	Est. Ship
180	MS RF G	5	1000	8/22/18	8/29/18	9/5/18	10/17/18
180	MS RF G	5	1000	10/24/18	10/31/18	11/7/18	12/19/18
65	MS RF GP	1	4700	9/24/18	10/1/18	10/8/18	12/17/18
65	MS RF GP	1	4700	10/24/18	10/31/18	11/7/18	1/16/19
65	MS RF GP	1	4700	11/21/18	11/28/18	12/5/18	2/13/19
65	MS RF LP	1	4700	9/24/18	10/1/18	10/8/18	12/17/18
65	MS RF LP	1	4700	10/24/18	10/31/18	11/7/18	1/16/19
40	MS RF G	1	7250	10/17/18	10/24/18	10/31/18	1/20/19
28	HPC RF	1	14000	10/31/18	11/7/18	11/14/18	2/3/19

Send us an e-mail to reserve area or request information.

BRGTC2 Timeline and Costs





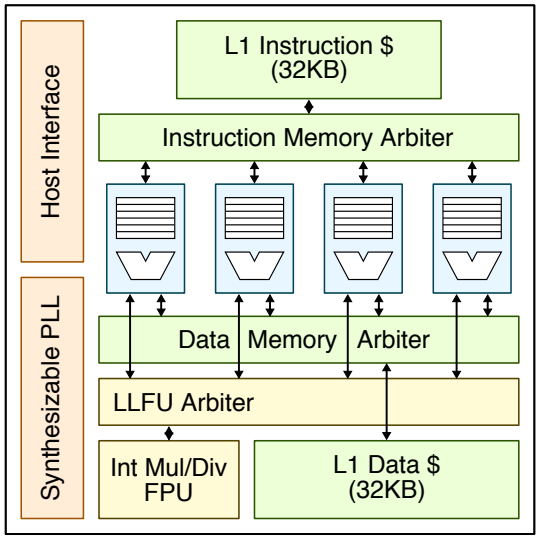
Time breakdown

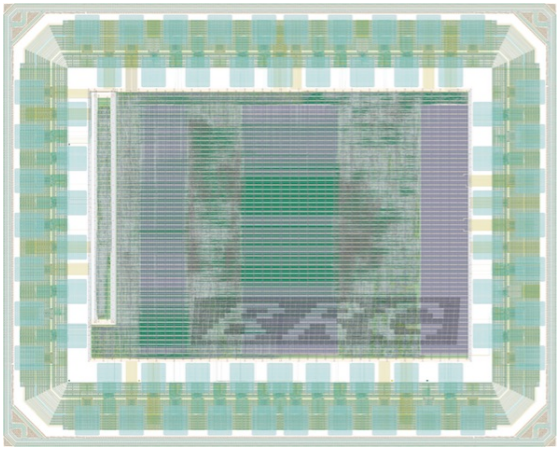
- One month for one student to pass DRC/LVS for dummy logic with staggered IO pads and no SRAMs
- One-month period with seven graduate students using PyMTL for design, test, and composition

Seven graduate students working across:

- Applications development
- Porting an in-house work-stealing runtime to RISC-V target
- Cycle-level design-space exploration with gem5
- RTL development and testing of each component including SRAMs
- Composition testing at RTL and gate level
- SPICE-level modeling of the synthesizable PLL
- IO floorplanning
- Physical design and post-PnR performance tuning

BRGTC2 Timeline and Costs

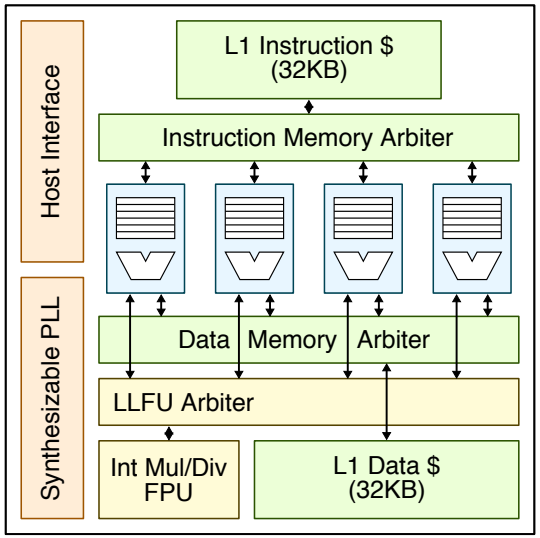


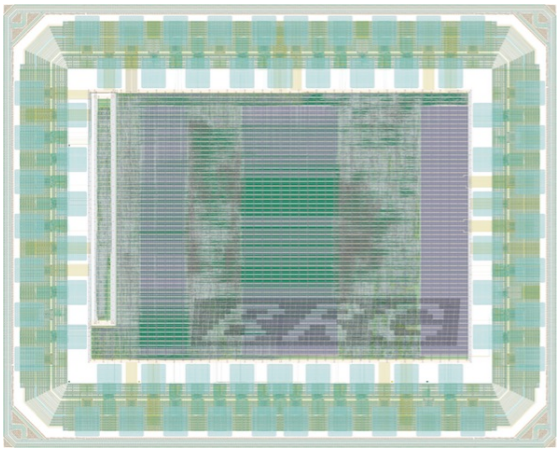


Cost breakdown

- 1×1.25 mm die size and one hundred parts for about $18K under the MOSIS Tiny2 program
- Packaging costs (about $2K for twenty parts)
- Board costs (less than $1K for PCB and assembly)
- Graduate student salaries
- Physical IP costs
- EDA tool licenses

A New Era of Silicon Prototyping in Computer Architecture Research





Key Takeaways

- Building silicon prototypes is traditionally challenging and costly
- Challenges have significantly reduced
  - Ecosystems for open builders (based on RISC-V)
  - Productive tools for small teams (e.g., PyMTL, ASIC flows)
- Costs have significantly reduced
  - MPW services support small minimum sizes in advanced nodes
- It is now feasible and attractive to consider RISC-V silicon prototypes for supporting future research

Acknowledgements

- NSF CRI Award #1512937
- NSF SHF Award #1527065
- DARPA POSH Award #FA8650-18-2-7852
- Donations from Intel, Xilinx, Synopsys, Cadence, and ARM
- Thanks: U.C. Berkeley, RISC-V Foundation, Shreesha Srinath

Backup Slides