ORDER: Open-Source Rooted Design Experts with Repute

Integration Meeting July 2019

Princeton University and

University of Washington

PRINCETON UNIVERSITY

UNIVERSITY of WASHINGTON

SoC Design Advisor Team

David Wentzlaff

Professor at Princeton, EE
Tilera Co-founder & Lead Architect

MIT PhD & MS, UIUC BS

Research in Manycore, Cloud Systems, and Biodegradable Computing

Runs OpenPiton Project

Michael B. Taylor

Professor at University of Washington, CSE & ECE
MIT PhD & MS

Earliest research on Dark Silicon, Tiled Multicore, ASIC Clouds
RISC-V manycore

OpenCelerity and BaseJump open source projects

C.-J. Richard Shi

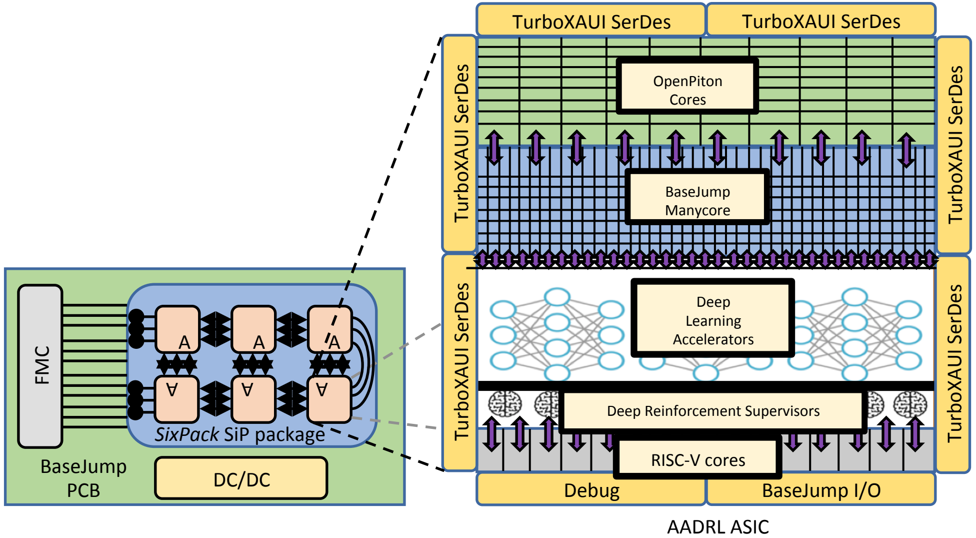
Professor at University of Washington, ECE

Founder of Orora Design Technologies, Inc, a pioneer in automated analog design and verification

PI on multiple DoD/DARPA projects (NZERO, RHBD, 3DIC NeoCAD)

ORDER Project Overview

- Fulfill role of SoC Design Advisors to help DARPA create a pushbutton CAD technology that accelerates the rate of progress by making chip design easier
- Independent team of hardware design experts with a strong track record in open source and open collaboration, and design in advanced process nodes
- Provide requirements, feedback, designs, and real world design experience to TA-1 toolflow teams
- Build and Fabricate Chips, Boards, and SiP Package using IDEA tools



Building Open Source SoCs with IDEA Tools

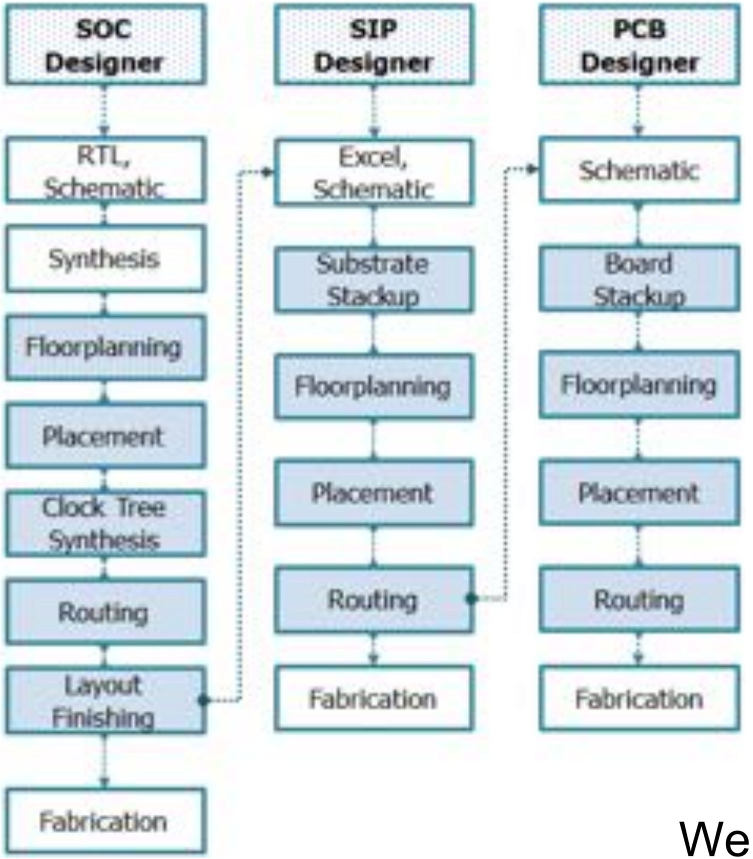
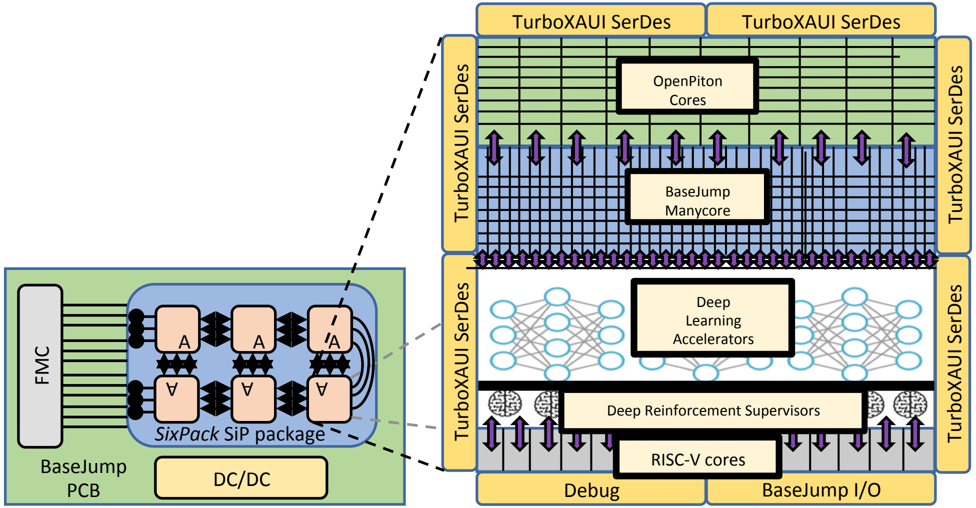


Image Credit: DARPA IDEA BAA

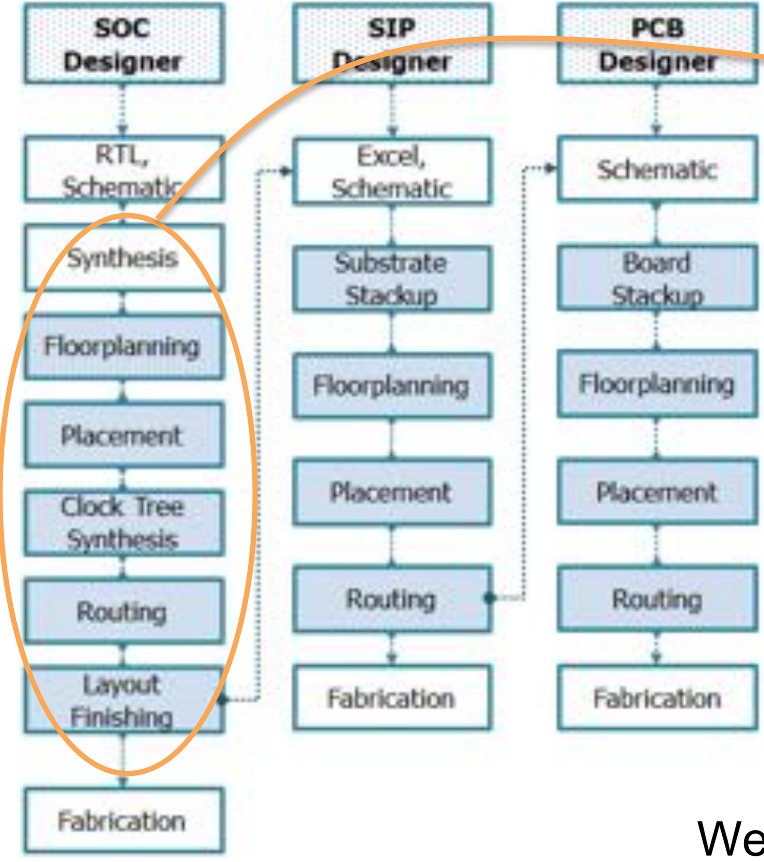


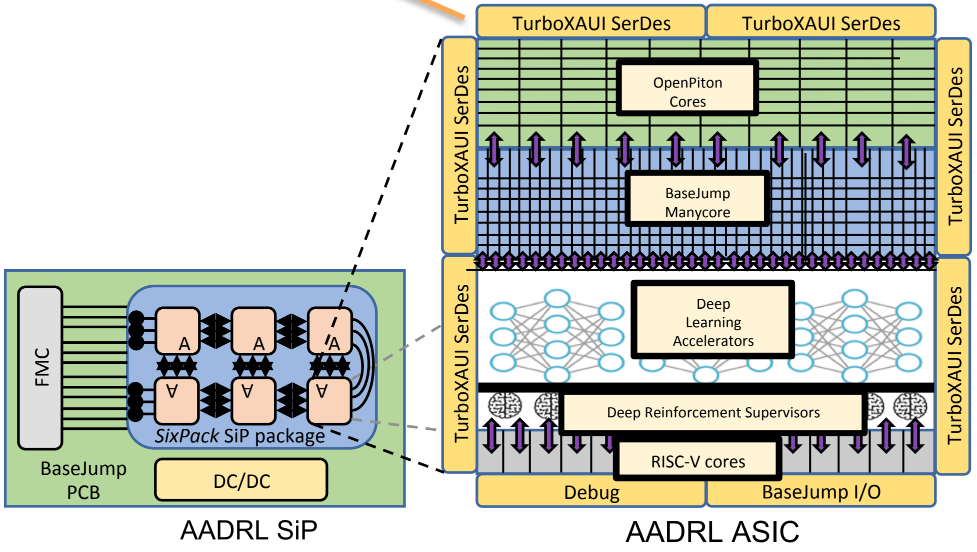
AADRL SiP

AADRL ASIC

We are excited to use your tools!

Building Open Source SoCs with IDEA Tools

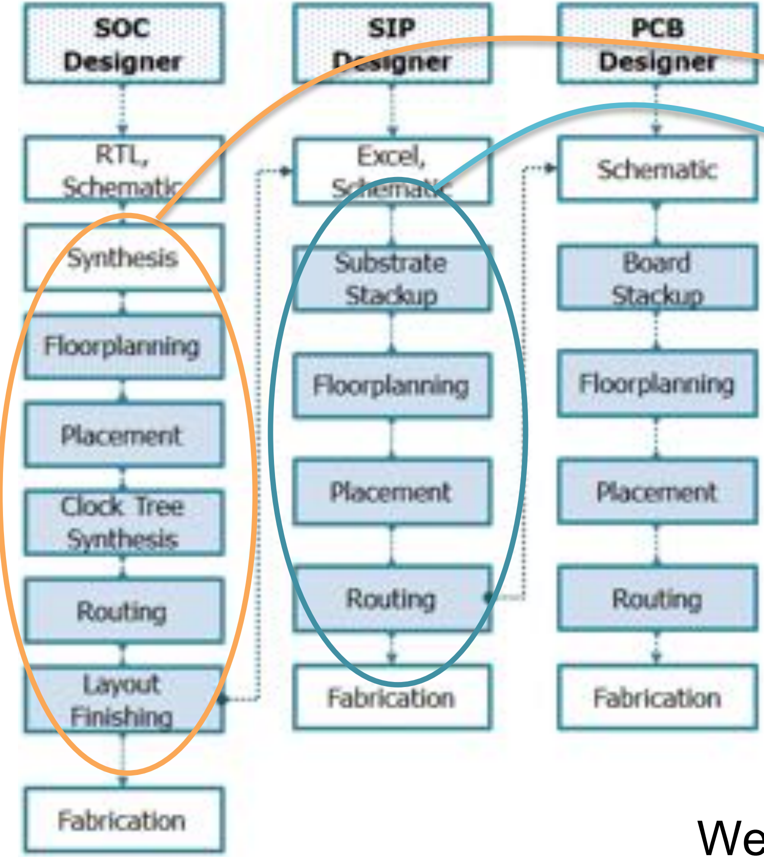




We are excited to use your tools!

Image Credit: DARPA IDEA BAA

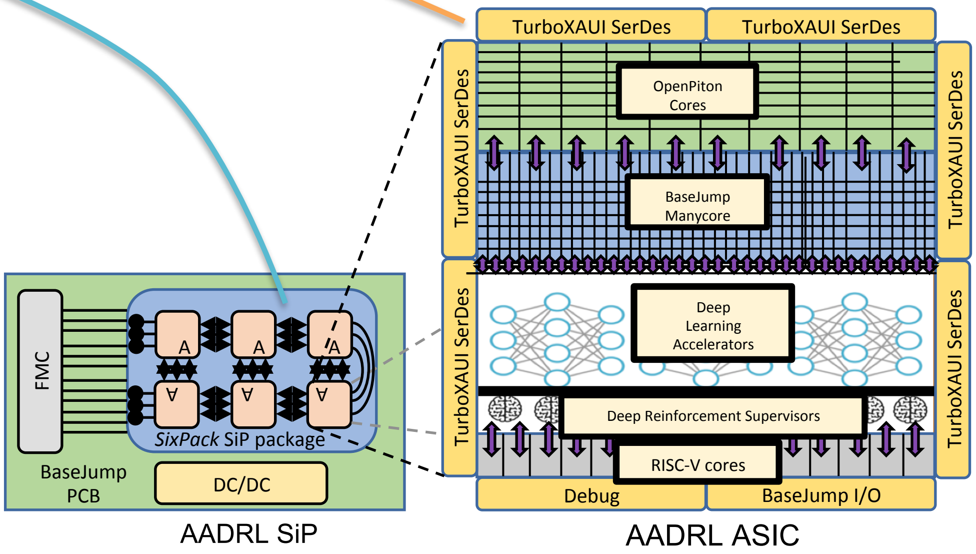
Building Open Source SoCs with IDEA Tools



SOC Designer
RTL, Schematic
Synthesis
Floorplanning
Placement
Clock Tree Synthesis
Routing
Layout Finishing
Fabrication

SIP Designer
Excel, Schematic
Substrate Stackup
Floorplanning
Placement
Routing
Fabrication

PCB Designer
Schematic
Board Stackup
Floorplanning
Placement
Routing
Fabrication



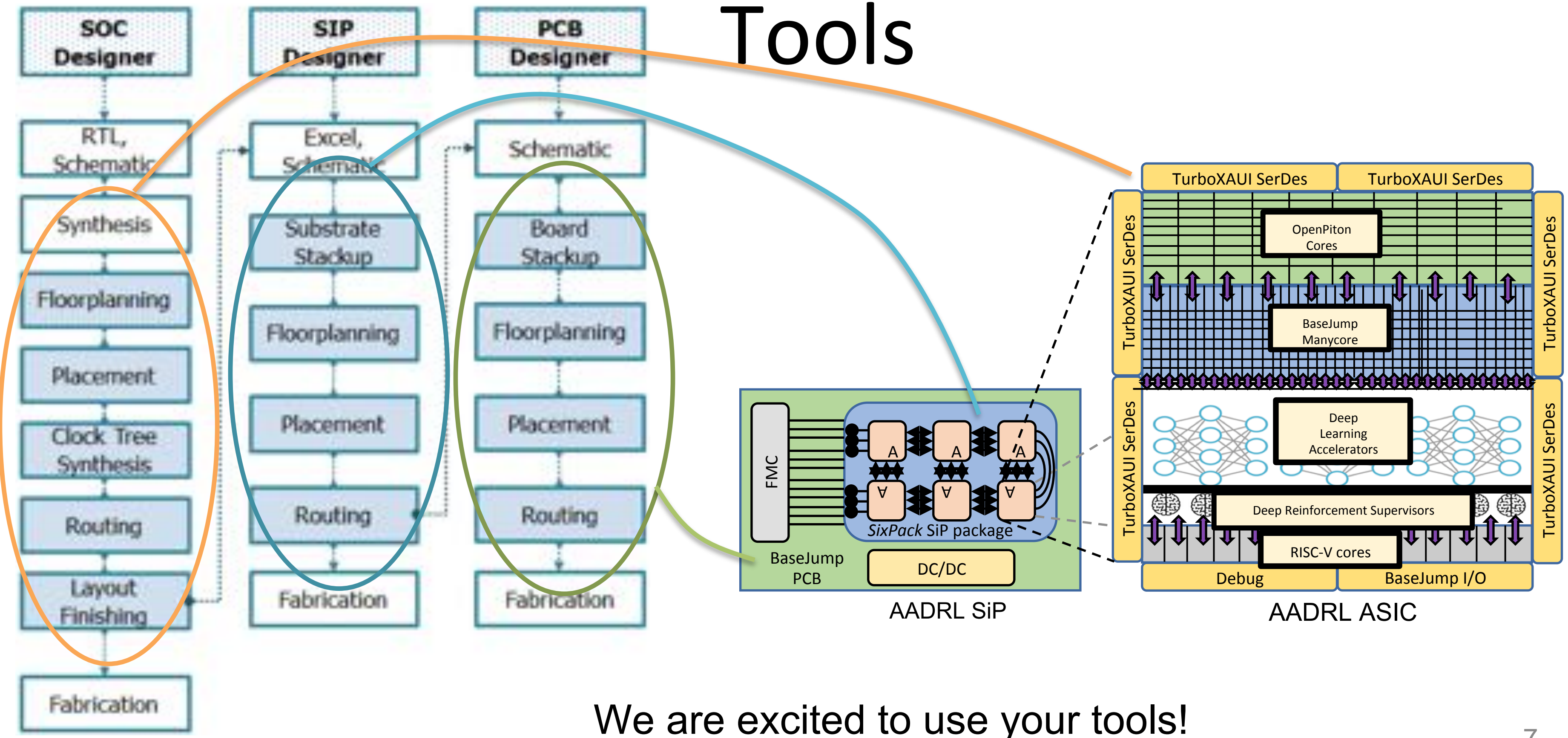
AADRL SiP

AADRL ASIC

We are excited to use your tools!

Image Credit: DARPA IDEA BAA

Building Open Source SoCs with IDEA Tools



We are excited to use your tools!

Image Credit: DARPA IDEA BAA

Outline

- Benchmarks
  - OpenPiton Design Benchmark
    - Demo
  - BSG Pipeclean Testsuite
  - UW Idea Analog Test Cases
- Tools and Formats
  - IDEA Dimensionless Format (IDF)
  - SV to Verilog
    - Demo

Enabling IDEA Through Convenient Benchmarks

World's first release of hundreds of high-quality, open-source, self-contained (Pickled into single file) hardware designs.

OpenPiton Design Benchmark

https://github.com/PrincetonUniversity/OPDB

UW IDEA Analog Test Cases

https://github.com/uwidea/UW-IDEA_AnalogTestCases

BSG Pipeclean

https://github.com/bespoke-silicon-group/bsg_pipeclean_suite

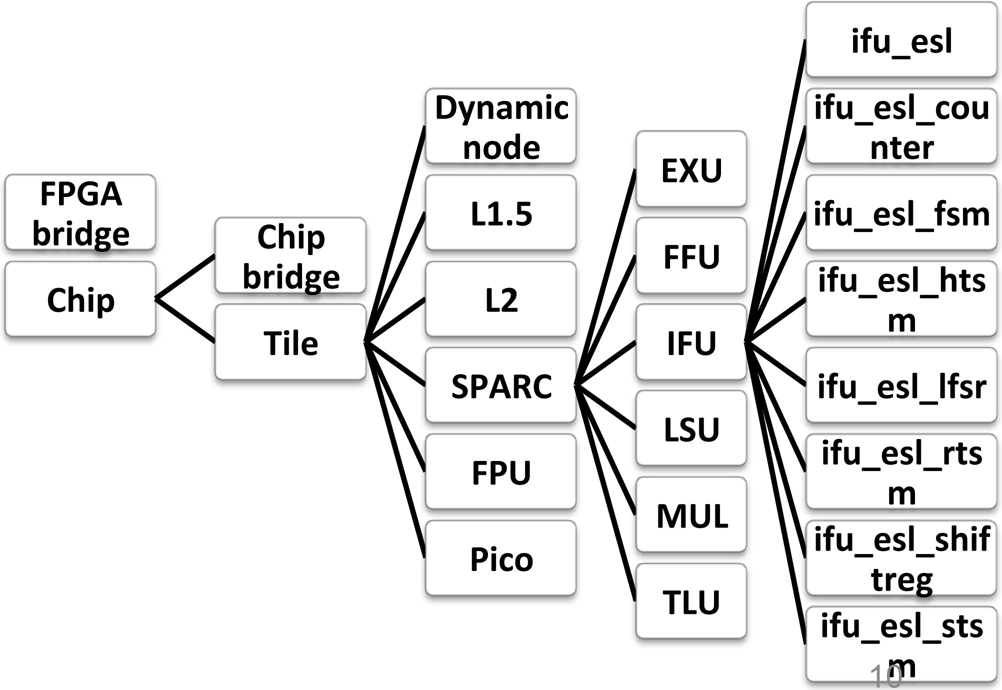
- Creation of "Pickling" flow to generate hundreds of easy to ingest HW designs.
- Release of high quality, complex, open-source HW designs supporting IDEA tool developers.
  - OpenPiton Design Benchmark
  - IDEA Analog Test Cases
  - BSG Pipeclean Suite

Challenges Addressed:

- HW designs require users to use complex build flows/infrastructure. Our "Pickled" designs enable CAD tool designers to only look at one file.
- CAD tool designers are in dire need for high-quality, non-trivial, open-source designs. We provide hundreds.

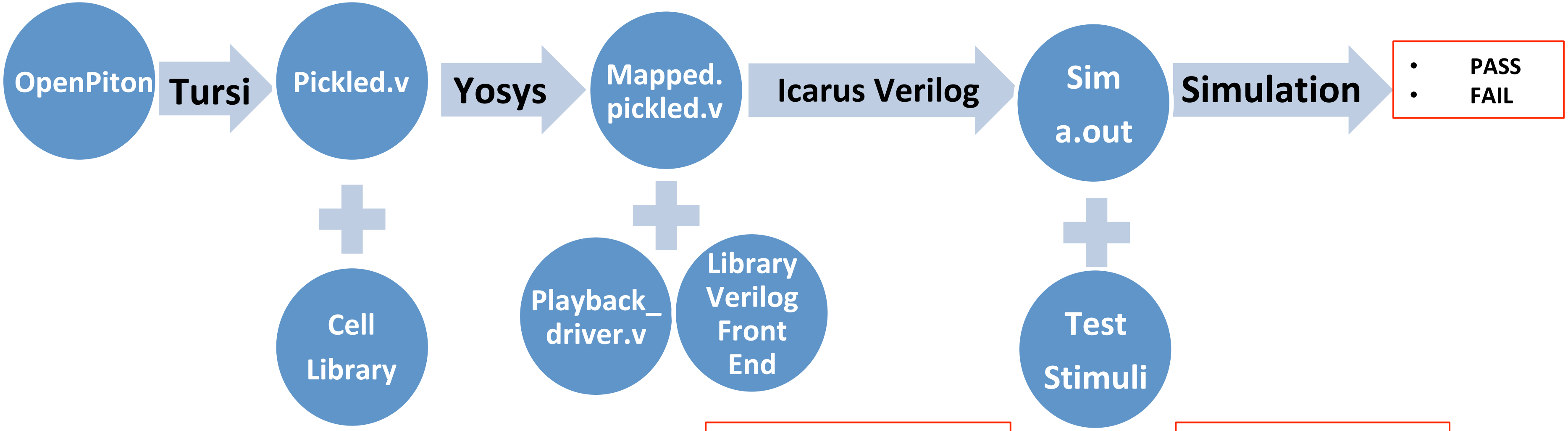
OpenPiton Design Benchmark

- Based on the OpenPiton open-source research processor
- 24 different modules
  - Variety of sizes, functionalities, core counts
- 636 “pickled” Verilog designs
  - Instantiation of modules with different configuration parameters
- Includes floorplan and .sdc files for multiple configurations
- Built “pickling” tool (Tursi) using FuseSoC



https://github.com/PrincetonUniversity/OpenPiton
https://github.com/PrincetonUniversity/OPDB

OpenPiton Design Benchmark Open Source Tools Demo



OpenPiton → Tursi → Pickled.v → Yosys → Mapped. pickled.v → Icarus Verilog → Sim a.out → Simulation →

- PASS
- FAIL

Pickled.v + Cell Library

Mapped. pickled.v + Playback_ driver.v + Library Verilog Front End

Sim a.out + Test Stimuli

1. Create single "pickled" file of dynamic node module using Tursi (FuseSoC + Icarus Verilog)

2. Synthesize "pickled" file with a given library using Yosys

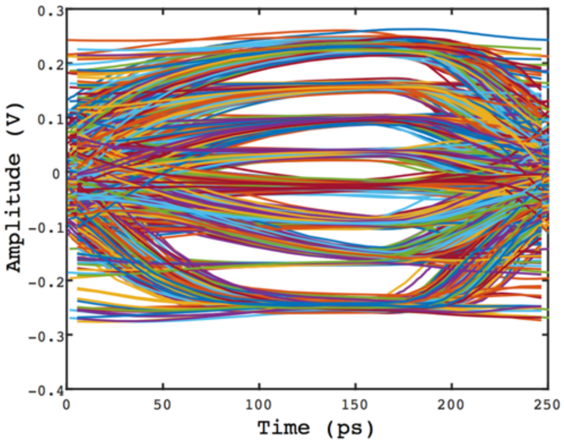
3. Compile playback driver module and synthesized pickle file into an executable using Icarus Verilog

4. Simulation with test stimuli

IDEA Analog Test Cases

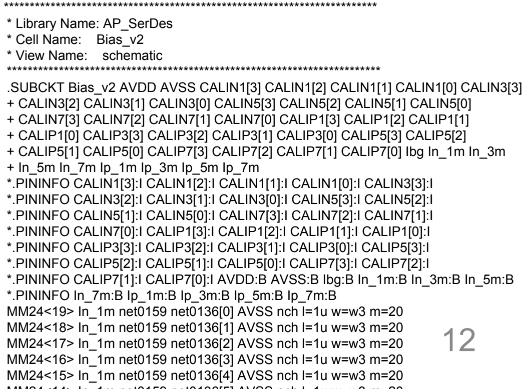
https://github.com/uwidea/UW-IDEA_AnalogTestCases

- Collating pre-existing, developed for other projects, and developed for ORDER analog test cases into test suite
- Hierarchical schematics
- CDL files describe circuit (SPICE Netlist)
- Documentation and simulation results to enable others to match results
- Working to extend to larger test suite
- Exercising IDEA tools from: Purdue, Sandia, UMinn, UT-Austin



Eye Diagram for TurboXAUI Transmitter

SPICE Netlist

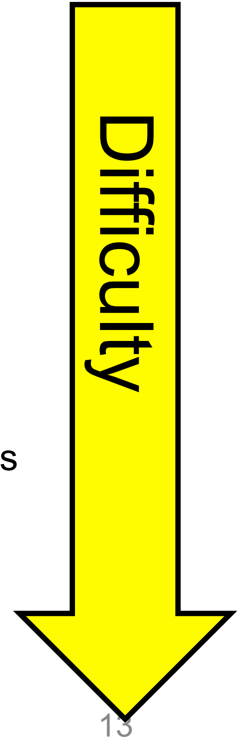


```
************************************************************************
* Library Name: AP_SerDes
* Cell Name:    Bias_v2
* View Name:    schematic
************************************************************************
.SUBCKT Bias_v2 AVDD AVSS CALIN1[3] CALIN1[2] CALIN1[1] CALIN1[0] CALIN3[3]
+ CALIN3[2] CALIN3[1] CALIN3[0] CALIN5[3] CALIN5[2] CALIN5[1] CALIN5[0]
+ CALIN7[3] CALIN7[2] CALIN7[1] CALIN7[0] CALIP1[3] CALIP1[2] CALIP1[1]
+ CALIP1[0] CALIP3[3] CALIP3[2] CALIP3[1] CALIP3[0] CALIP5[3] CALIP5[2]
+ CALIP5[1] CALIP5[0] CALIP7[3] CALIP7[2] CALIP7[1] CALIP7[0] Ibg In_1m In_3m
+ In_5m In_7m Ip_1m Ip_3m Ip_5m Ip_7m
*.PININFO CALIN1[3]:I CALIN1[2]:I CALIN1[1]:I CALIN1[0]:I CALIN3[3]:I
*.PININFO CALIN3[2]:I CALIN3[1]:I CALIN3[0]:I CALIN5[3]:I CALIN5[2]:I
*.PININFO CALIN5[1]:I CALIN5[0]:I CALIN7[3]:I CALIN7[2]:I CALIN7[1]:I
*.PININFO CALIN7[0]:I CALIP1[3]:I CALIP1[2]:I CALIP1[1]:I CALIP1[0]:I
*.PININFO CALIP3[3]:I CALIP3[2]:I CALIP3[1]:I CALIP3[0]:I CALIP5[3]:I
*.PININFO CALIP5[2]:I CALIP5[1]:I CALIP5[0]:I CALIP7[3]:I CALIP7[2]:I
*.PININFO CALIP7[1]:I CALIP7[0]:I AVDD:B AVSS:B Ibg:B In_1m:B In_3m:B In_5m:B
*.PININFO In_7m:B Ip_1m:B Ip_3m:B Ip_5m:B Ip_7m:B
MM24<19> In_1m net0159 net0136[0] AVSS nch l=1u w=w3 m=20
MM24<18> In_1m net0159 net0136[1] AVSS nch l=1u w=w3 m=20
MM24<17> In_1m net0159 net0136[2] AVSS nch l=1u w=w3 m=20
MM24<16> In_1m net0159 net0136[3] AVSS nch l=1u w=w3 m=20
MM24<15> In_1m net0159 net0136[4] AVSS nch l=1u w=w3 m=20
MM24<14> In_1m net0159 net0136[5] AVSS nch l=1u w=w3 m=20
```

BSG Pipeclean Benchmark Suite

https://github.com/bespoke-silicon-group/bsg_pipeclean_suite

`small_comb`:	8 bit multiplier
`medium_comb`:	32-bit multiplier
`large_comb`:	128-bit multiplier
`black_parrot_be_only_2019_03_11`:	Back end (be) of early version of black parrot
`black_parrot_fe_only_2019_03_11`:	Front end (fe) of early version of black parrot
`black_parrot_2019_03_28`:	Entire Black Parrot RISC-V core with reduced size crossbars *Less challenging routing problem*
`black_parrot_2019_03_11`:	Entire Black Parrot RISC-V core with full crossbars *More challenging routing problem*



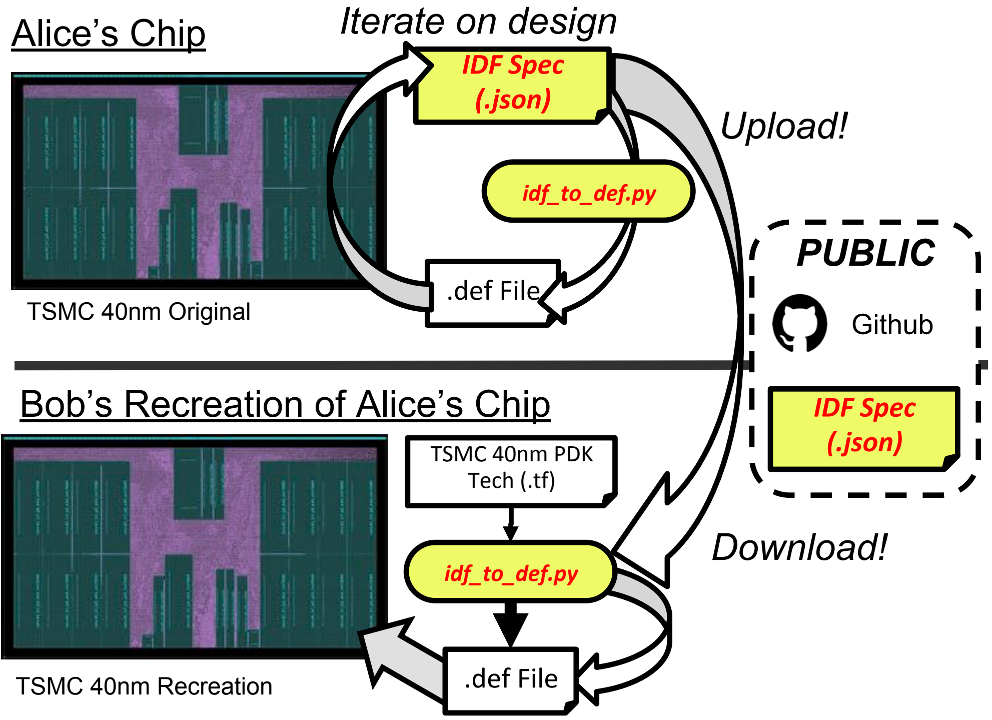
All of these designs work with commercial tools; use IDF dimensionless format

Outline

- Benchmarks
  - OpenPiton Design Benchmark
    - Demo
  - BSG Pipeclean Testsuite
  - UW Idea Analog Test Cases
- Tools and Formats
  - IDEA Dimensionless Format (IDF)
  - SV to Verilog
    - Demo

Refining the IDF (IDEA Dimensionless Format)

New technology allowing open-source collaborators to share physical design descriptions of chips “in the clear” without violating NDAs.



Alice’s Chip

Iterate on design

IDF Spec (.json)

idf_to_def.py

.def File

TSMC 40nm Original

Upload!

PUBLIC

Github

IDF Spec (.json)

Download!

Bob’s Recreation of Alice’s Chip

TSMC 40nm PDK Tech (.tf)

idf_to_def.py

.def File

TSMC 40nm Recreation

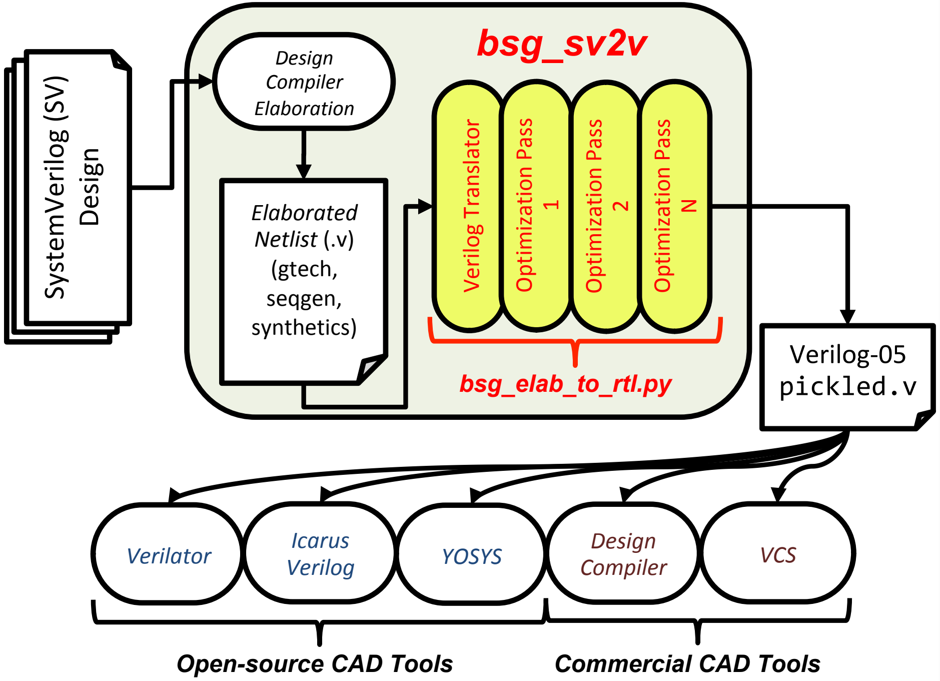
- IDEA Dimensionless Format (IDF)
  - Simple, coherent coarse floorplan spec allowing open-sourced physical design constraints.
  - Dimensionless = All units are PDK agnostic revealing no sensitive info (NDA safe)
  - IDF-to-DEF converter for turnkey integration into existing CAD tool infrastructure and EDA vendor compatibility.
    - 30.7s to go from IDF file to floorplan in IC Compiler for TSMC 40nm.
- Challenge: IDF excels at moving between similar PDKS but moving between dissimilar PDKs (process node, SRAM compiler) will require the intelligence of a tool like OpenROAD to adjust.

https://github.com/bespoke-silicon-group/bsg_idf_tools

IDF specified in collaboration with UW (Taylor), Princeton, Michigan, Andreas, and Greater IDEA Team from San Diego Meeting

Providing A Pathway for SystemVerilog to Open Source

First open source conversion infrastructure to allow academics to export modern HW designs in SystemVerilog for use with open-source tools.



- SystemVerilog to Verilog RTL Converter (bsg_sv2v)
  - Converts cutting edge hardware designs written in SystemVerilog (SV) to a single Verilog 2005 compliant RTL file for maximum compatibility with currently available open-source CAD tools.
  - Extensible framework for post-converted optimizations.
    - Reg-Redux: For a CPU frontend, improved VCS sim speed by 2.3x and LoC by 2.8x over un-optimized pickled netlist.
- Challenge: Tools may differ in their interpretation of SystemVerilog. We use Design Compiler as our parser since all academics have it, and it is the industry standard, and post-process the output to regenerate Verilog.

https://github.com/bespoke-silicon-group/bsg_sv2v

UW (Taylor)

ORDER: Open-Source Rooted Design Experts with Repute

- OpenPiton Design Benchmark - https://github.com/PrincetonUniversity/OPDB
- BSG Pipeclean Suite - https://github.com/bespoke-silicon-group/bsg_pipeclean_suite
- UW IDEA Analog Test Cases - https://github.com/uwidea/UW-IDEA_AnalogTestCases
- IDF Tools - https://github.com/bespoke-silicon-group/bsg_idf_tools
- System Verilog to Verilog Flow https://github.com/bespoke-silicon-group/bsg_sv2v
- OpenPiton GitHub - https://github.com/PrincetonUniversity/openpiton
- Open Celerity - http://opencelerity.org
- BaseJump - http://bjump.org

PRINCETON UNIVERSITY

UNIVERSITY of WASHINGTON

The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.