



Federation: Open Source SoC Design Methodology

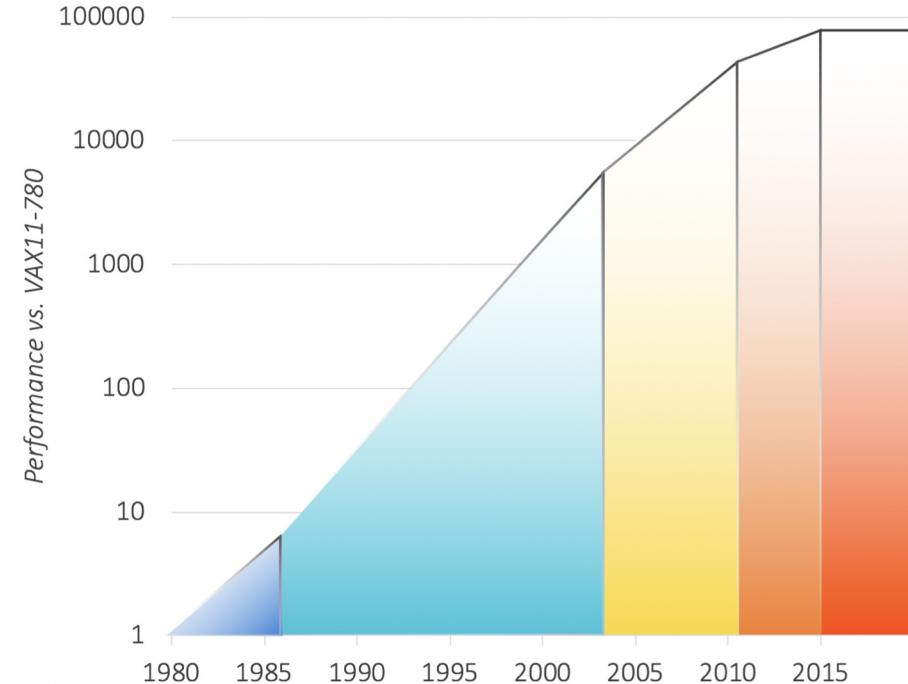
Jack Koenig, Staff Engineer
Aug 3, 2019

Hardware Trends: Compute Needs are Changing



e.g., machine learning

Source: Medium, Entering the world of Machine Learning



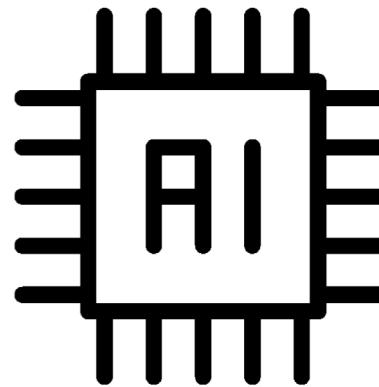
Based on [SPECIntCPU](#). Source: John Hennessy and David Patterson,
Computer Architecture: A Quantitative Approach, 6/e. 2018

But, CPUs are not getting faster!

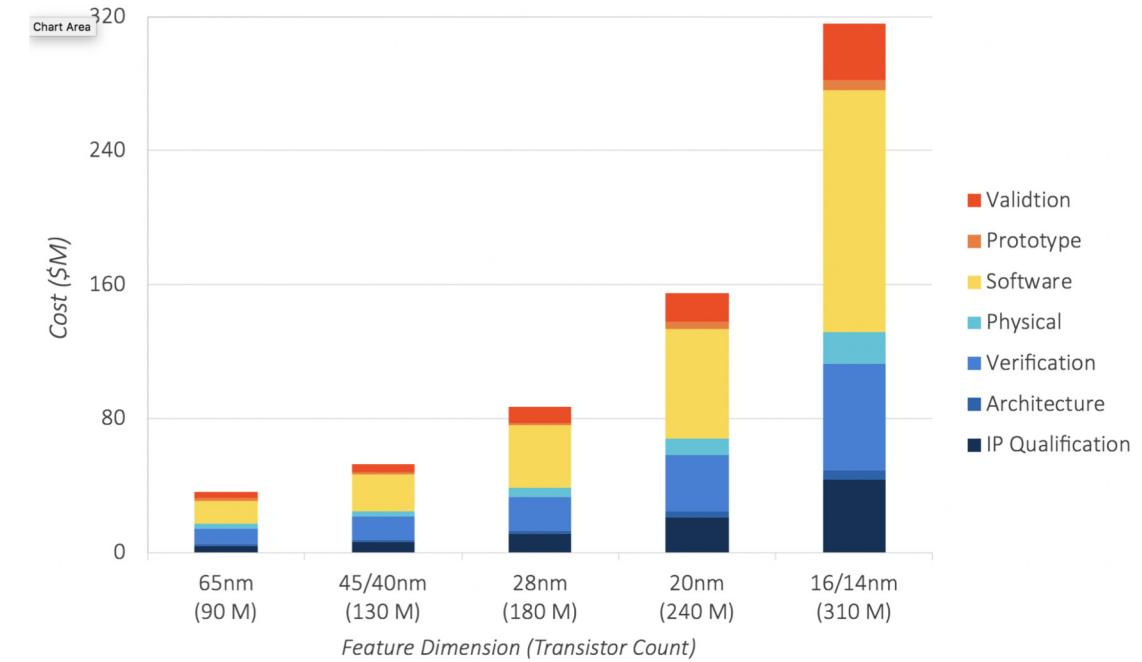
Hardware Trends: Custom Hardware to the Rescue!



GPUs



Custom Chips
(e.g., Google TPU)



But, custom chip development costs are too high!

*How did  Instagram turn into a
\$1B acquisition with only 13 employees?**



Why do silicon projects need experts from at least 14+ disciplines just to get started?



Architect



Logic



RTL



Analog



Verification



Simulation



Emulation



Synthesis



Place
& Route



Layout



ECO



Foundry



Package



Test



<Your Name Here>
Tech Stack

Readily-Available
Reusable
Technology

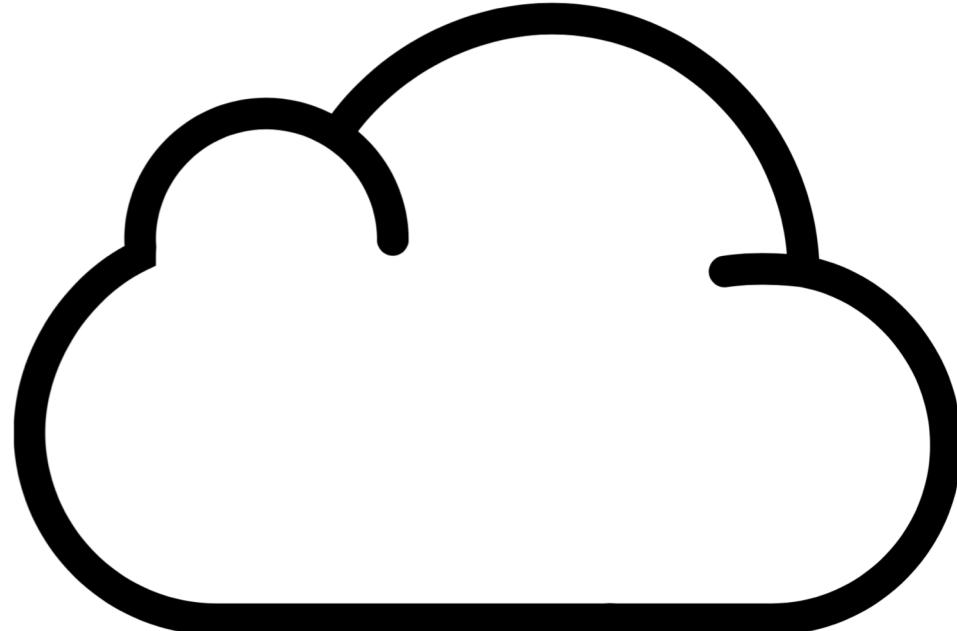
from techstacks.io



Infrastructure

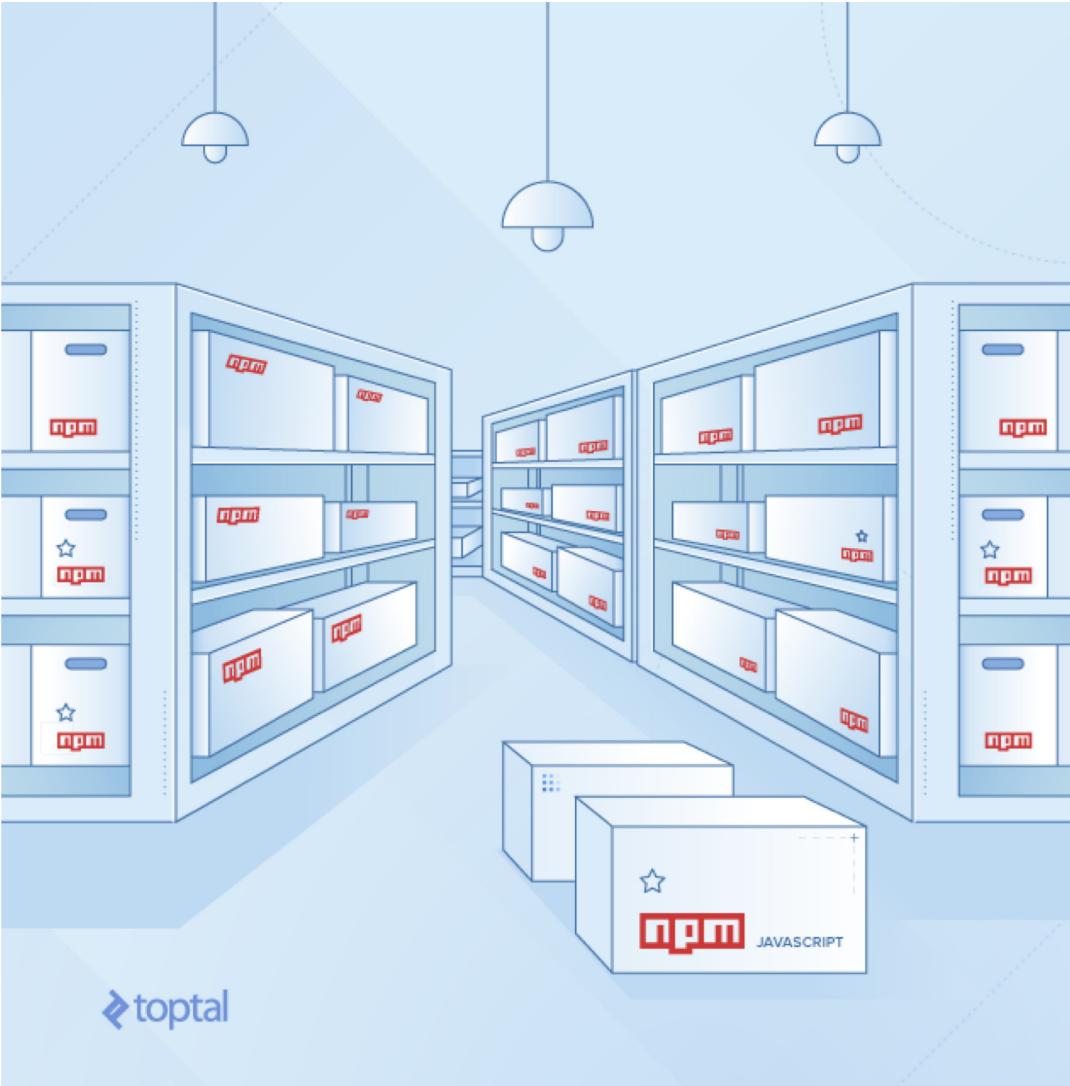


We Should Copy “Innovations” from the Software Industry



open-source standards
abstraction and code reuse
composability and APIs
productivity tooling
commodity infrastructure

npm Enables Modular Web Development with Reusable Packages



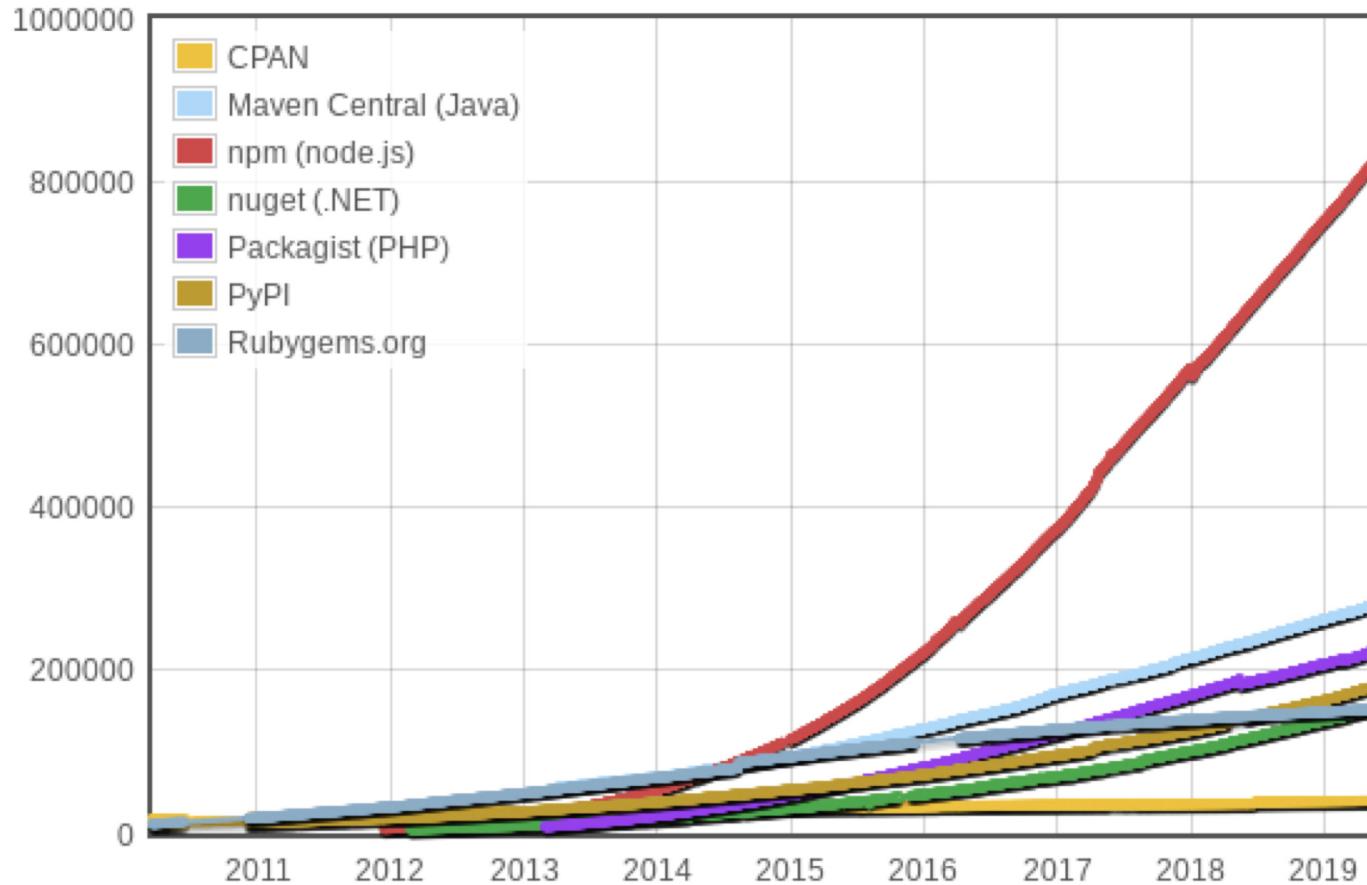
package.json

Schema: <http://json.schemastore.org/package>

```
1  {
2    "name": "sense-o-some-or-other-unique-package-name",
3    "description": "this is the description for some or other package",
4    "version": "1.0.0",
5    "responsibleAuthority": "Vogon HQ-and I am the value of a custom property",
6    "author": [
7      {
8        "name": "Marvin the depressed robot",
9        "web": "http://www.galacticpublishing.com"
10      }
11    ],
12    "collaborators": [
13      "Unnamed Developer <unnameddeveloper@galacticpublishing.com>",
14      "Another Developer <unnameddeveloper@galacticpublishing.com>"
15    ],
16    "license": "MIT",
17    "dependencies": {
18      "lodash": "4.0.0"
19    },
20    "devDependencies": {
21      "eslint": "3.7.0"
22    },
23    "optionalDependencies": {
24      "vogs-sphere": "***"
25    },
26    "scripts": {
27      "start": "grunt build && grunt env:local && node app.js"
28    }
29  }
```

npm Enables Javascript as Most Popular Programming Language

Module Counts



- 830.000 of packages
- 10 million users
- 30 billion packages downloads per month
- 90% of Web built of NPM packages
- NPM used everywhere: client, server, mobile, IoT
- Open source libraries availability is the major driving force for Language adoption [Leo2013] proven by NPM
- 70+ programming languages can be transpiled into JS, WebAssembly and published on NPM

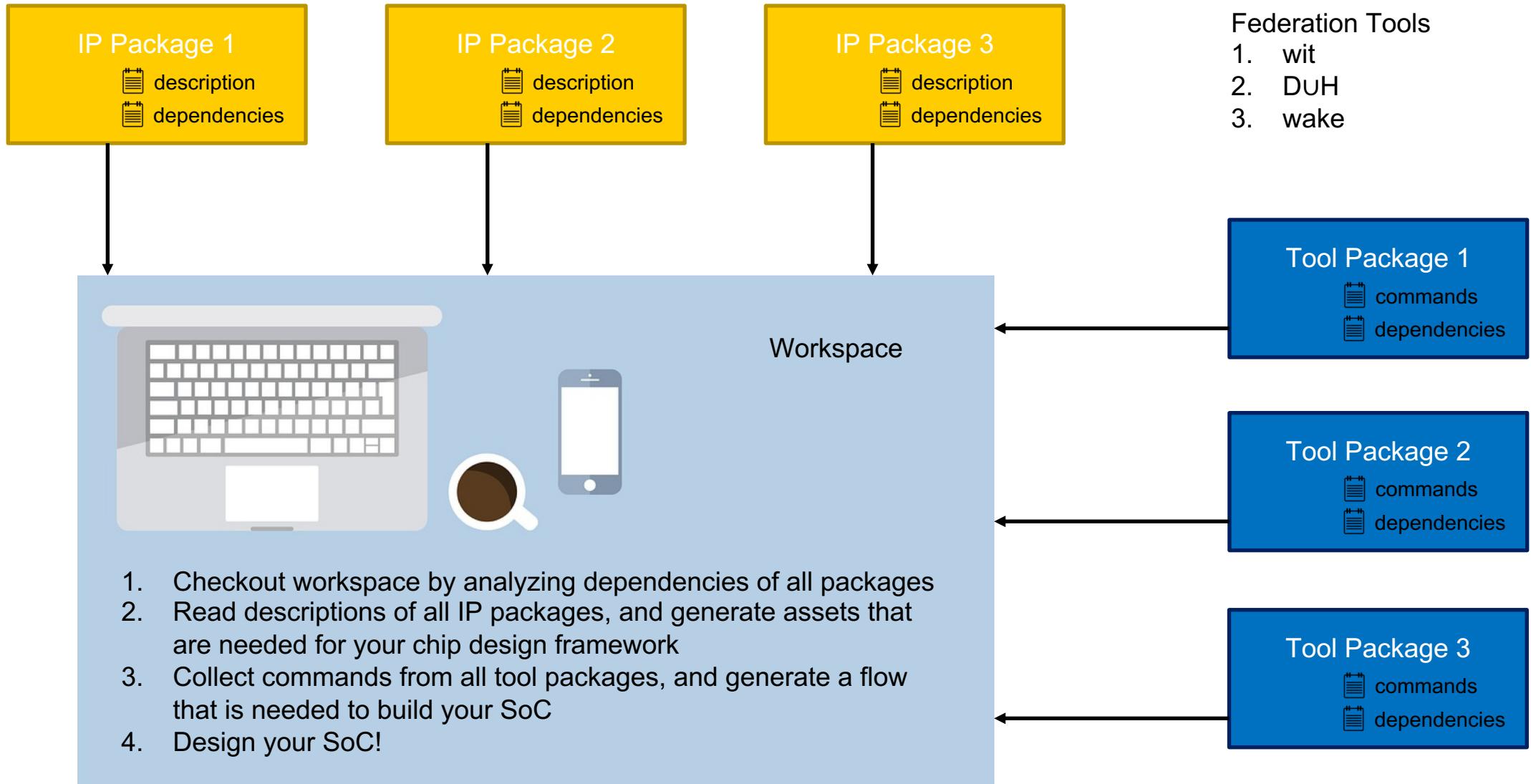
Federation:

*the action of forming states or organizations
into a single group with centralized control,
within which smaller divisions have
some degree of internal autonomy*

Federation:

*a suite of open-source tools
that SiFive is using to orchestrate
modular SoC design workflows*

Federation Tools Enable a Modular SoC Design Methodology

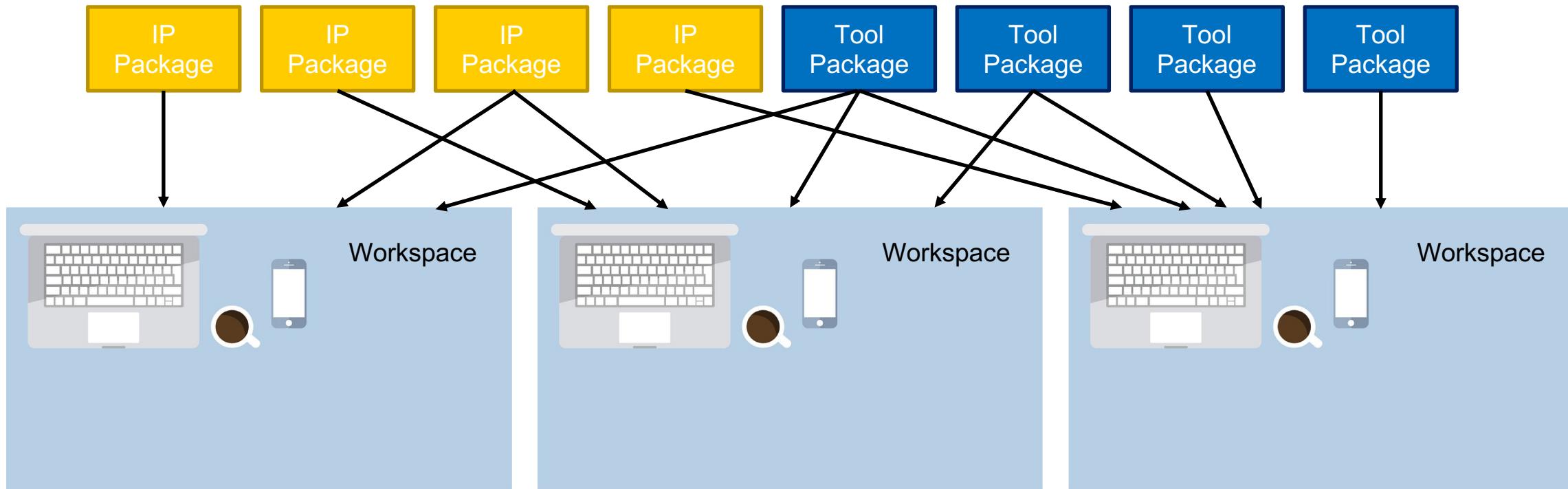




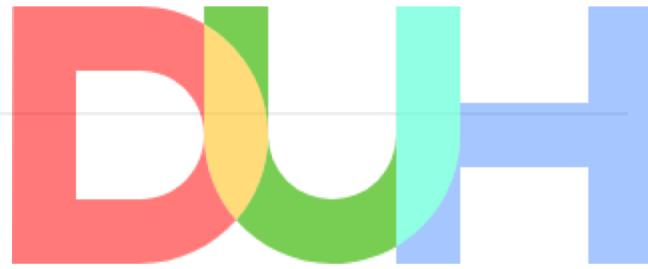
Federation Tools: **Wit**, the Workspace Integration Tool

- <https://github.com/sifive/wit>
- wit is a tool for managing dependencies between git repos
- (Think of it like git submodules++)
- Supplement to (not a replacement for) git-based workflows
- Provides a flexible, yet reproducible and deterministic, dependency-resolution algorithm
 - generates a flattened directory structure in which each package may exist only once
- Enables multiple developers to simultaneously develop and deploy features that touch multiple repositories

Benefits of Small, Modular, and Reusable Packages

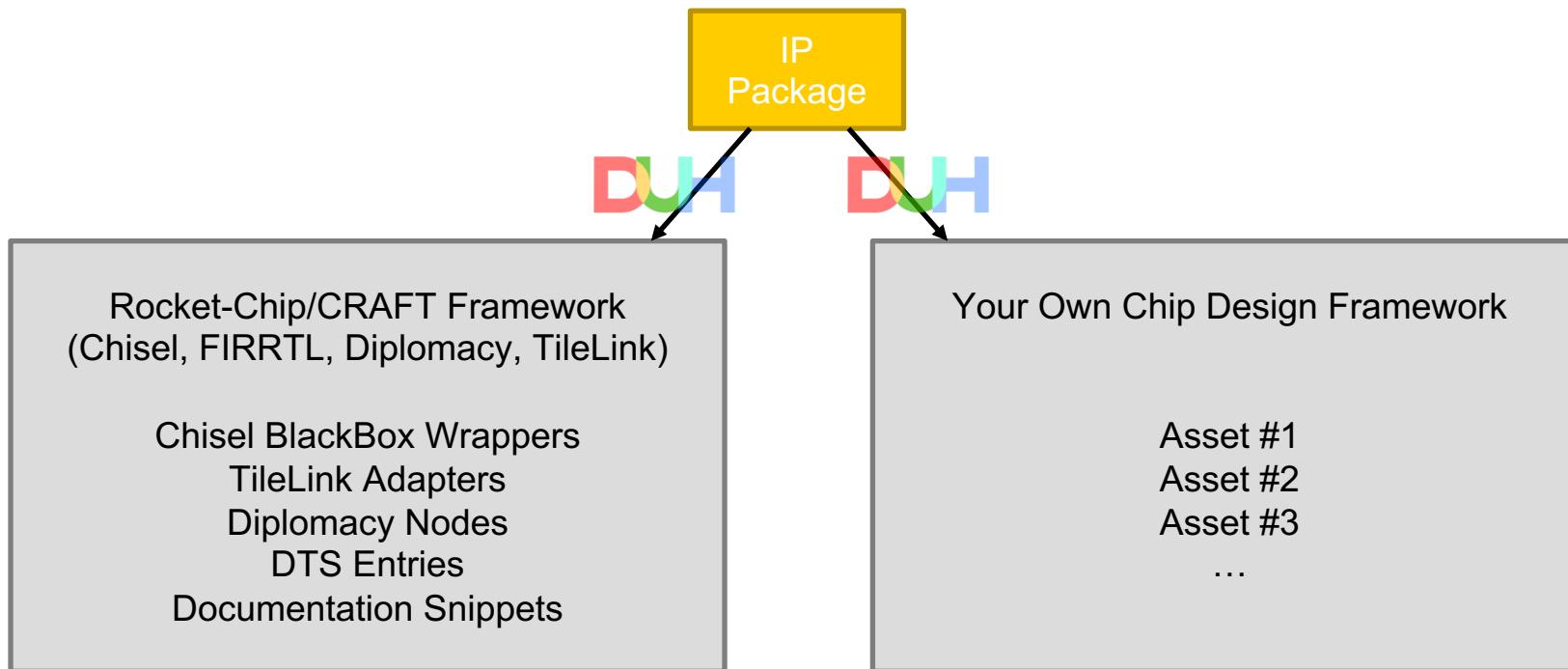


1. Enforces an interface between components; makes it easier to reuse parts in different contexts and designs
2. Encourages engineers to write unit tests for packages; makes it easier to filter bugs early
3. Localizes regression test failures; makes it easier to root-cause them
4. Enables fine-grained access control; makes it easier to open-source or give differential access to components



- <https://github.com/sifive/duh> pronounced [dûx]
- **A manifest document format capturing integration intent**
 - Components, Designs, Configurations
 - Ports, Bus Interfaces, Parameter Constraints, Registers,...
 - Open-Source, JSON5, Extensible, and Generator-Agnostic
- **Packaging tools for reusable hardware components and designs**
 - **Import:** Verilog, SystemRDL, IP-XACT,...
 - **Authoring:** Document creation, validation
 - **Export:** Scala, Chisel, Verilog, IP-XACT, TLM, Documentation
 - **GUI:** Query, Search, Configuration, Integration, Analysis

Benefits of a Data Format that Describes IP Generators



Decouples the details of the chip design framework from the underlying IP generator; as a result:

1. Acts as a single source of truth, enforcing the DRY principle
2. Automatically generates assets that are needed for your chip design framework
3. Lowers the barrier to adopt your chip design framework
4. Makes it easier to use the IP package in other chip design frameworks



Federation Tools: **Wake**, orchestrating your build flow

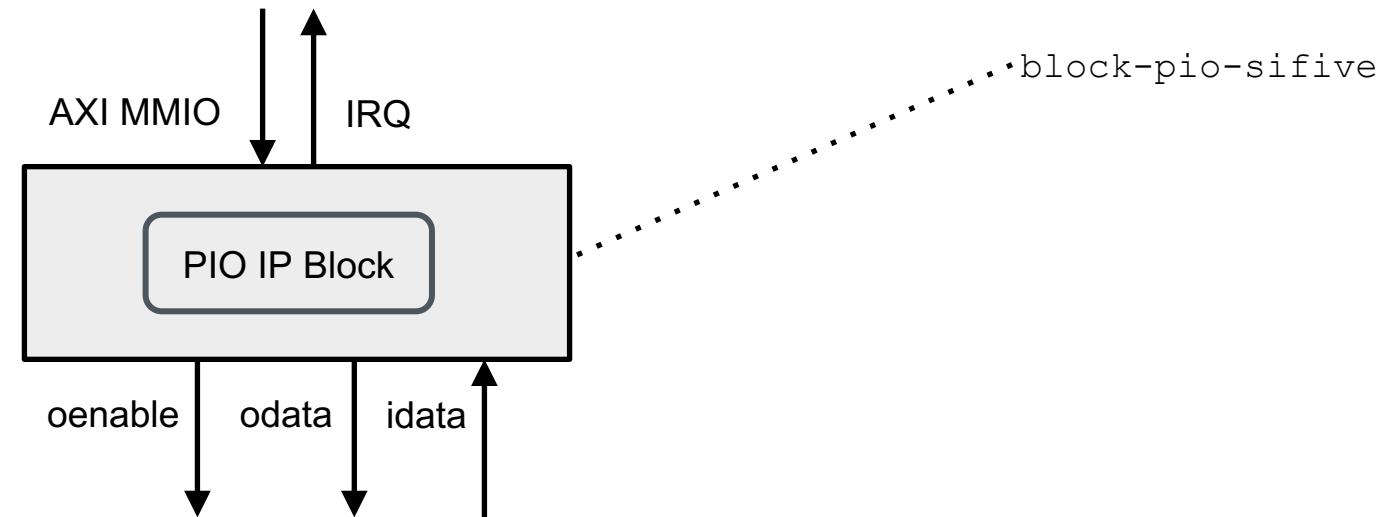
- <https://github.com/sifive/wake>
- Wake is a tool and a language to make build systems composable
- Defines APIs to distinguish tooling choices from design flow steps
- Plugs packages' build steps together with zero source changes



Federation Tools: Wake, orchestrating your build flow

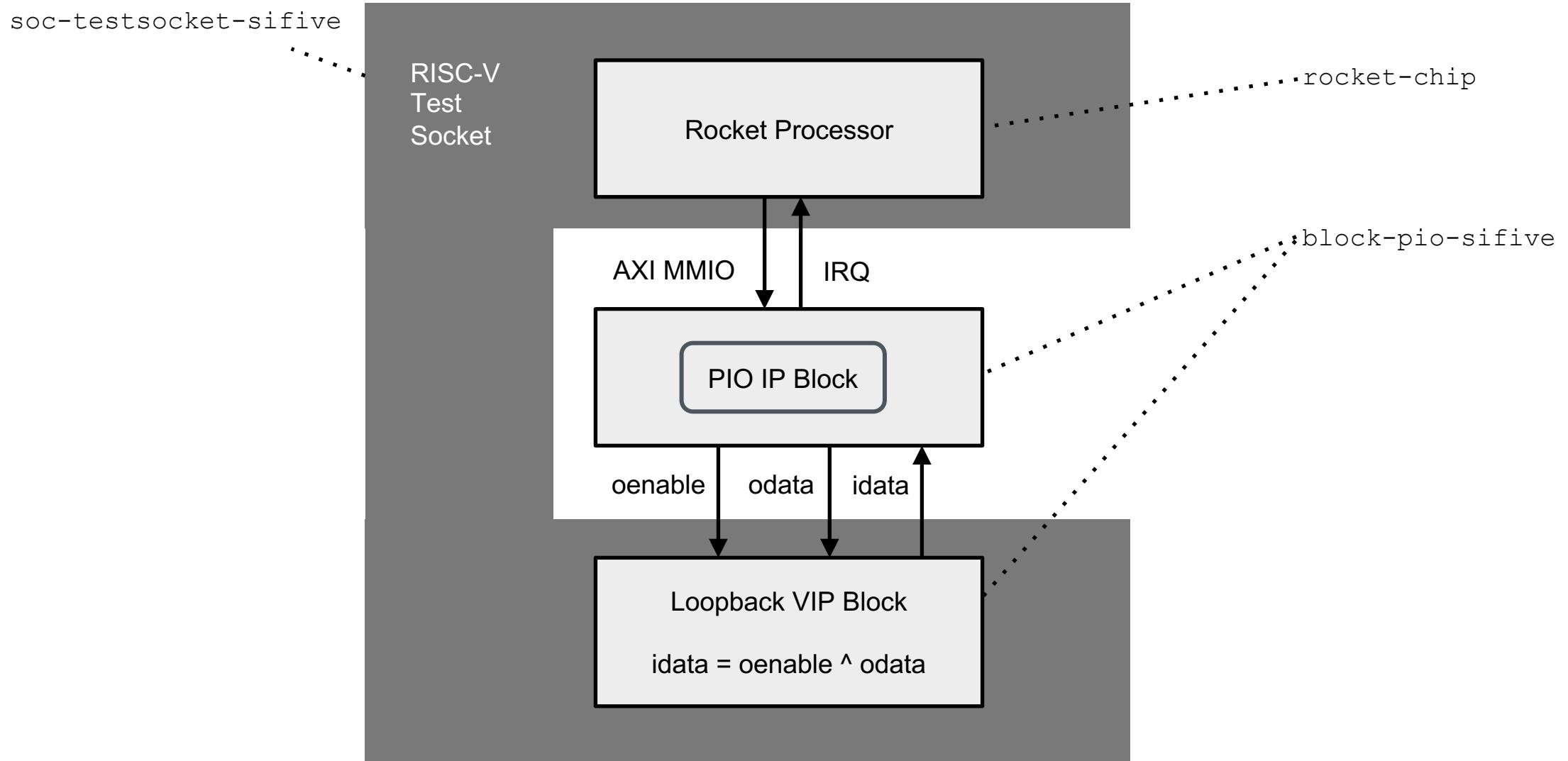
- **Dependent job execution:**
 - Which jobs to run next can depend on the results of previous jobs. All jobs may be dependent.
- **Dependency analysis:**
 - Fails builds that will not be reproducible due to underspecified inputs dependencies.
 - Prunes unused input dependencies so the job will not be re-run unless it must.
 - You almost never need to tell wake what files a job builds; it knows.
- **Build introspection:**
 - Wake keeps a database to record what it did.
 - Query that database at any time to find out exactly how a file in your workspace got there.
- **Intrinsically-parallel language**
 - While your build orchestration files describe a sequence of compilation steps, the wake language automatically extracts parallelism.
 - Only true data dependencies cause wake to sequence jobs.
- **Shared build caching**
 - If wake can prove it's safe, it will just copy the prebuilt files and save you time.
 - PRs whose regression tests pass immediately, increasing productivity.
 - (Work in Progress)

Case Study 1: Programmed-IO IP Package



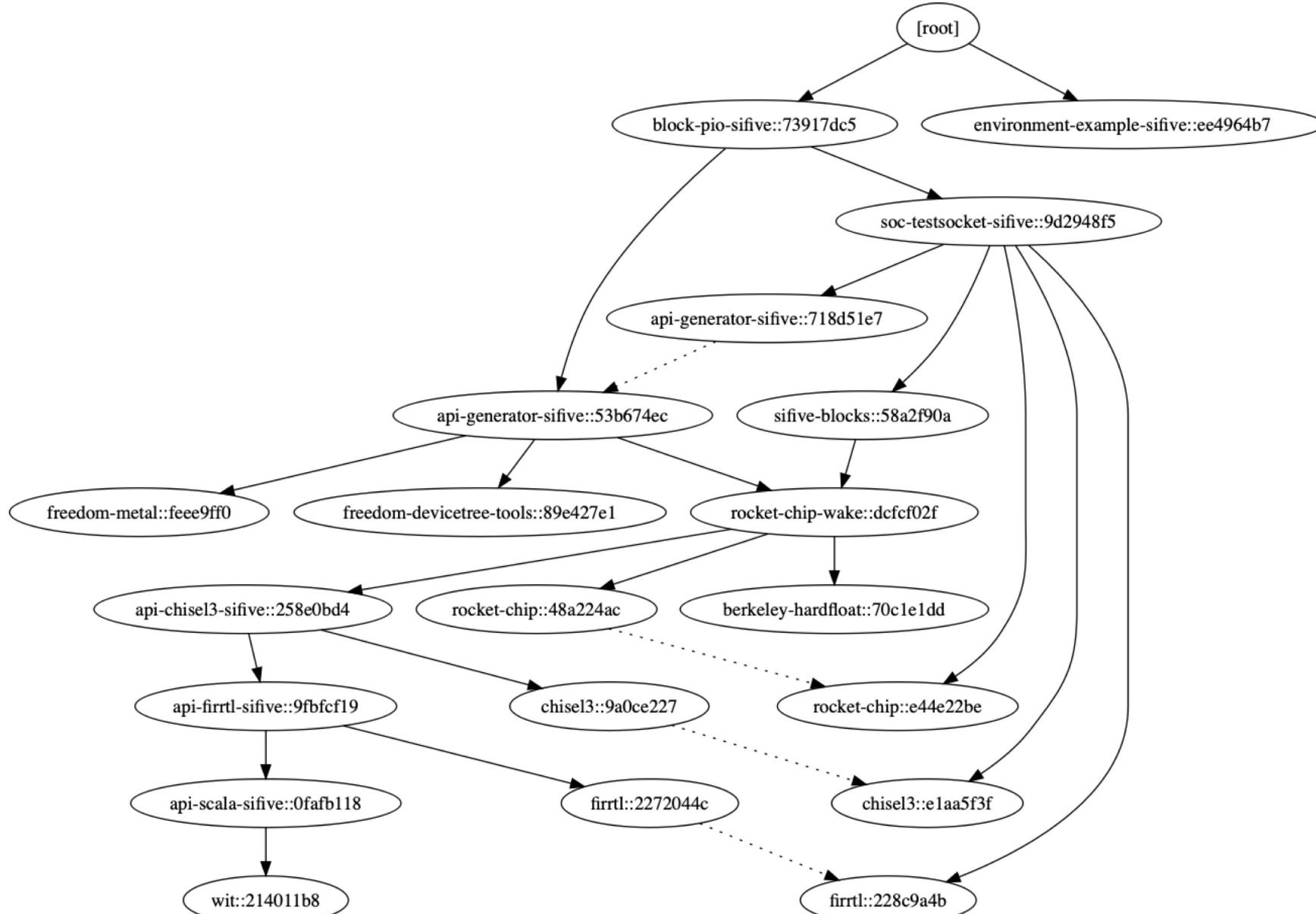
- IP containing IO-programmable register map
- DUH generation of wrapper containing Scala classes, Chisel blackbox, Diplomacy Nodes, Rocket-Chip traits

Case Study 1: Programmed-IO IP Package Integration

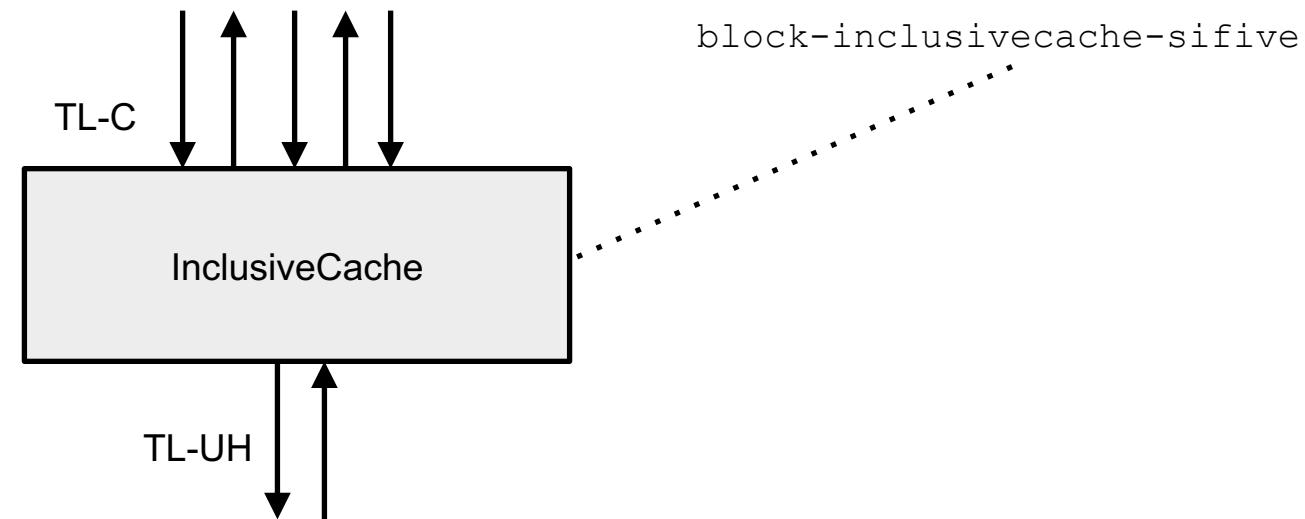




block-pio-sifive Package Dependency Graph

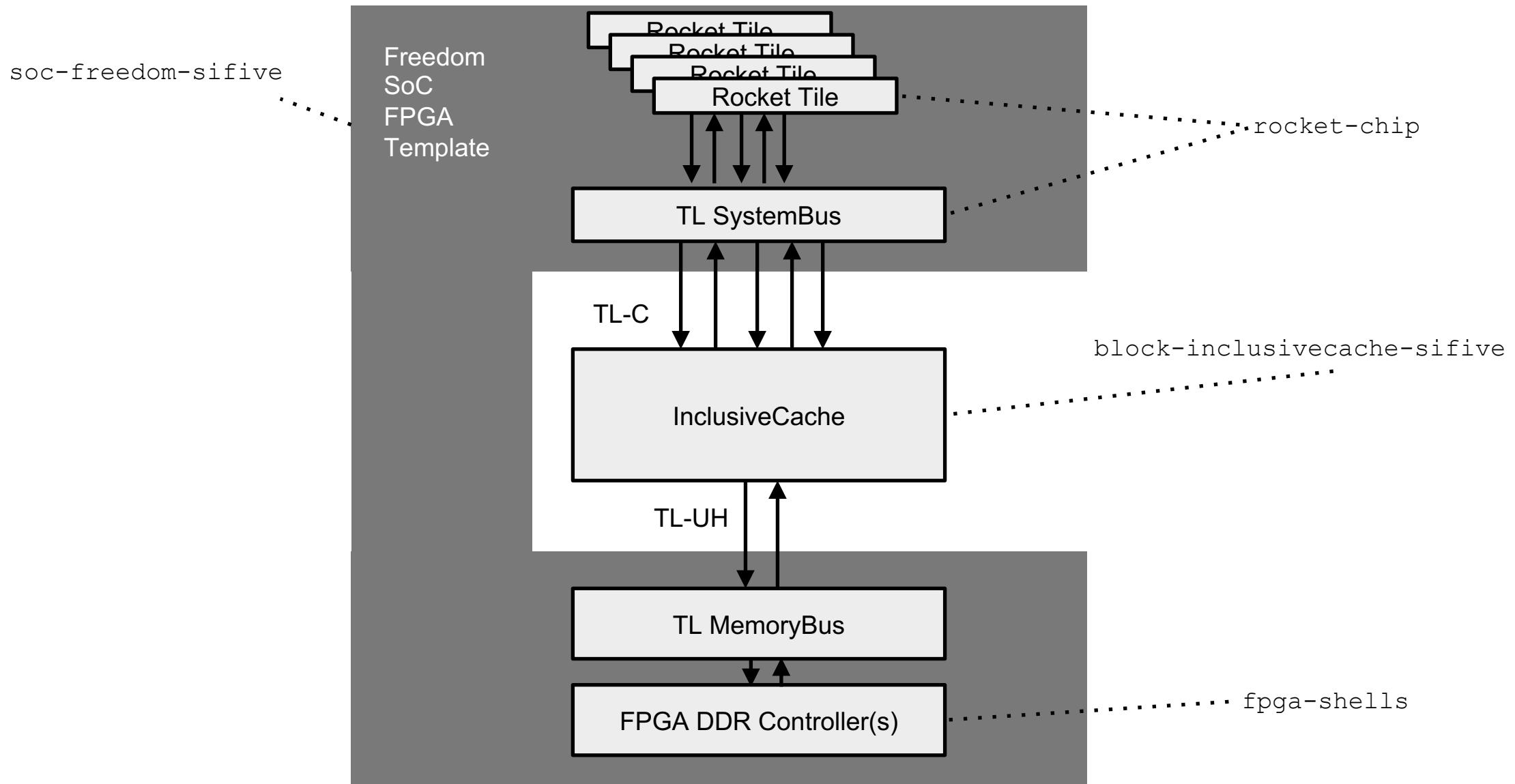


Case Study 2: Cache Coherent SoC Template



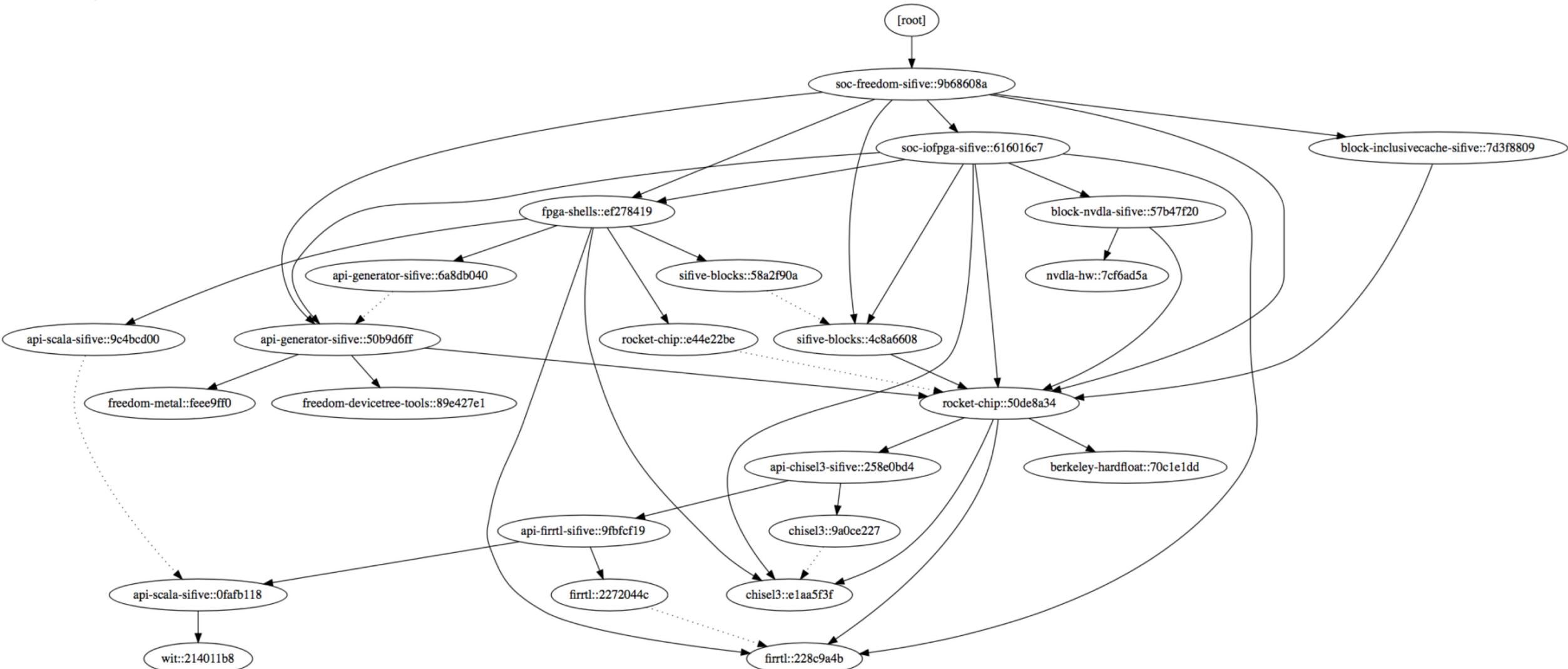
- Coherent, last-level inclusive cache (eg. L2 cache for rocket-chip!)
- Full directory bits stored with metadata tag
- TileLink adapter; drop-in replacement for `tilelink.BroadcastHub`
- SW-controlled flush interface
- Easily Configure size, ways, banking and sub-banking factors, external bandwidth, buffering

Case Study 2: Cache Coherent System Template Integration





soc-freedom-sifive Package Dependency Graph





Open Source Package Progress (SiFive packages)

- Example projects
 - <https://github.com/sifive/block-pio-sifive>
 - <https://github.com/sifive/soc-testsocket-sifive>
 - <https://github.com/sifive/soc-iofpga-sifive>
 - <https://github.com/sifive/soc-freedom-sifive>
- CRAFT framework APIs
 - <https://github.com/sifive/api-generator-sifive>
 - <https://github.com/sifive/api-scala-sifive>
 - <https://github.com/sifive/api-chisel3-sifive>
 - <https://github.com/sifive/api-firrtl-sifive>
 - <https://github.com/sifive/environment-example-sifive>
- Packaged blocks
 - <https://github.com/sifive/block-inclusivecache-sifive>
 - <https://github.com/sifive/block-pio-sifive>
 - <https://github.com/sifive/block-nvdla-sifive>
- Federation tools
 - <https://github.com/sifive/wit>
 - <https://github.com/sifive/duh>
 - <https://github.com/sifive/wake>





Open Source Roadmap (Other Related Projects)

- Rest of SiFive's Freedom repository will be refactored into packages in Q3 of 2019
 - <https://github.com/sifive/freedom> => <https://github.com/sifive/soc-freedom-sifive>
- rocket-chip has been moved to the CHIPS Alliance Github org and is currently being split into packages
 - <https://github.com/freechipsproject/rocket-chip> => <https://github.com/chipsalliance/rocket-chip>
- CHIPS Alliance: a subsidiary of the Linux Foundation <https://github.com/chipsalliance/> <https://chipsalliance.org/>
 - Rocket-Chip
 - Chisel / FIRRTL
 - TileLink + OmniXtend
 - Verilator
 - SweRVE
 - FuseSoC
 - CocoTB





Questions?



01. Design

02. Review

03. Build

E5 Series

Bedford Falls

Review

Modes & ISA

On-Chip Memory

Ports

Security

Debug

Interrupts

Branch Prediction

Modes & ISA

Privilege Modes

- Machine Mode ?
- User Mode

ISA Extensions

- Multiply (M Extension) ?

Multiply Performance

4 CYCLE

PIPELINED

- Atomics (A Extension) ?

- Floating Point (F & D Extensions) ?

Bedford Falls Core Complex

E5 SERIES CORE RV64IMAFDC

Machine Mode • User Mode
Multiply (Pipelined) • Atomics • FP
16 Local Interrupts • 2 Perf Counters

28 BTB Entries • 512 BHT Entries
6 Ret Addr Stack

Physical Memory Protection
8 Regions

No ECC Support

Instr Cache
16KiB • 2-way

Data TIM
64KiB

JTAG Debug

4 HW Breakpoints
DMA

PLIC

7 Priority Levels
255 Global Int.

Front Port
32-bit AXI4

System Port
32-bit AXI4

Peripheral Port
32-bit AXI4

SiFive Core Designer

RISC-V Core IP Customization and Push-Button Configuration

<https://www.sifive.com/core-designer>

01. Design

02. Review

03. Prototype

Freedom Unleashed SoC

[Review](#)

IP Library Go to Category ▾

[Filter](#)

Rambus Phase Locked Loop PLL 40-2400MHz [ODS](#)

Mastering

AI ACCELERATOR

NVIDIA NVidia (Deep Learning Accelerator) NVIDIA Deep Learning Accelerator [Open Source](#)

CONFIGURABLE LOGIC

flexLogix Flex Logix EFLX-2.5K Embedded FPGA Embedded FPGA for reprogrammable hardware blocks [ODS](#)

HIGH SPEED INTERFACE

mobiveil Mobiveil GPEX PCI-Express 4.0 Controller PCI Express Gen4 Controller [ODS](#)

SECURITY

Rambus CryptoManager RT630 A complete solution for chip and system security [ODS](#)

Mastering Subsystems

Gigabit Ethernet Media Access Control

Front Bus 64-bit 300 MHz
System Bus 64-bit 300 MHz

Peripherals

Pulse Width Modulation Peripheral

UART Peripheral

I2C Peripheral

SPI Peripheral

QSPI Interface Peripheral

GPIO Controller

Corigine USB 3.1 Controller

SiFive DMA Engine

SiFive MaskROM

Foundational Blocks

Crystal Oscillator

Phase Locked Loop

SiFive Clock/Reset Control

Memory Bus 32-bit 600 MHz

DDR3/3L4 Controller

Memory Subsystem

DDR3/3L4 Controller

Chip Details

Platform: Freedom Unleashed

Process: TSMC 28nm

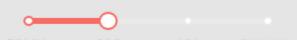
Base Design: Application Processor

Chip Settings

CPU Clock: **600 MHz**



Peripheral Bus: **200 MHz**



Mastering Bus Clock Ratio



Memory Port Width (bits)



CPU Clock speed and Bus Clock Ratio determine Front and System Bus speeds.

Bus widths are determined by the choice of Core IP.

Memory Bus speed is determined by IP in the Memory Subsystem.

SiFive Chip Designer

Coming soon



Design



Infrastructure



EDA



IP



Fab



Package/Test