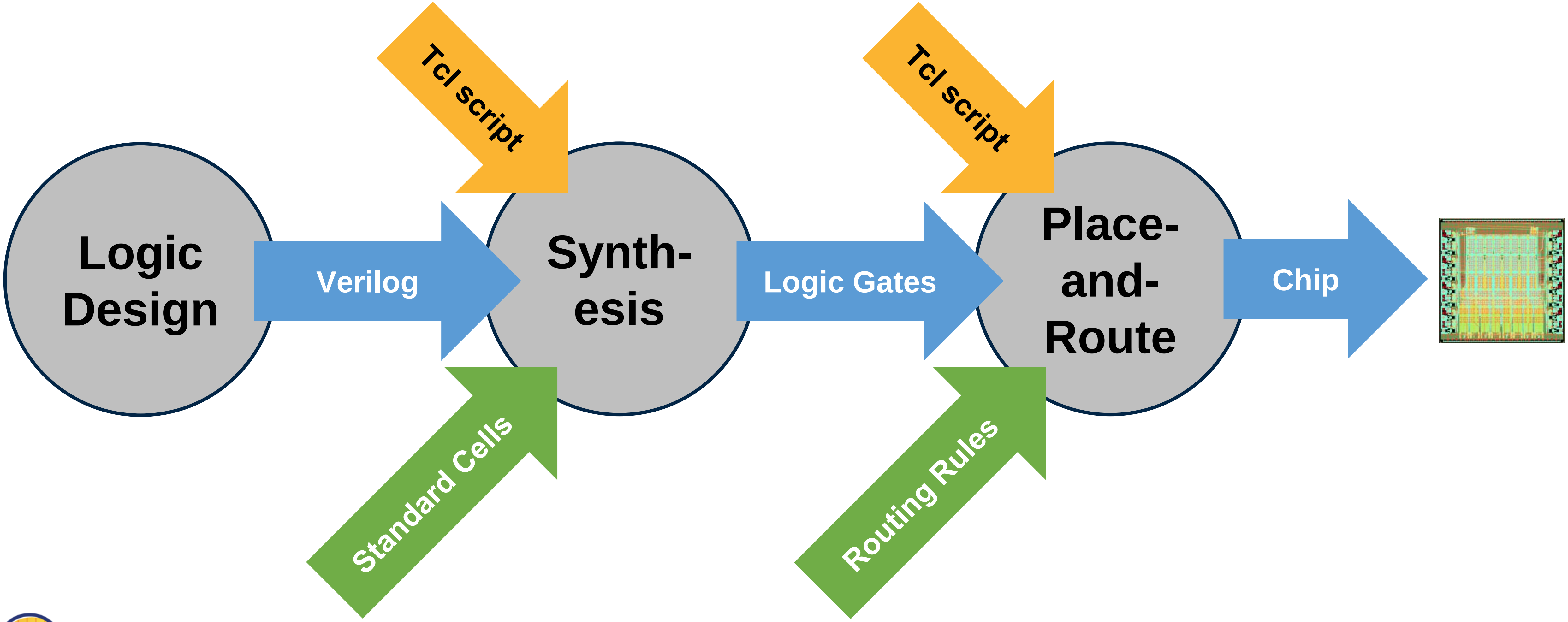
Hammer VLSI Flow

Palmer Dabbelt, Edward Wang, John Wright, Colin Schmidt,
Harrison Liew, Daniel Grubb, and many others,
Borivoje Nikolić, Krste Asanović, Jonathan Bachrach

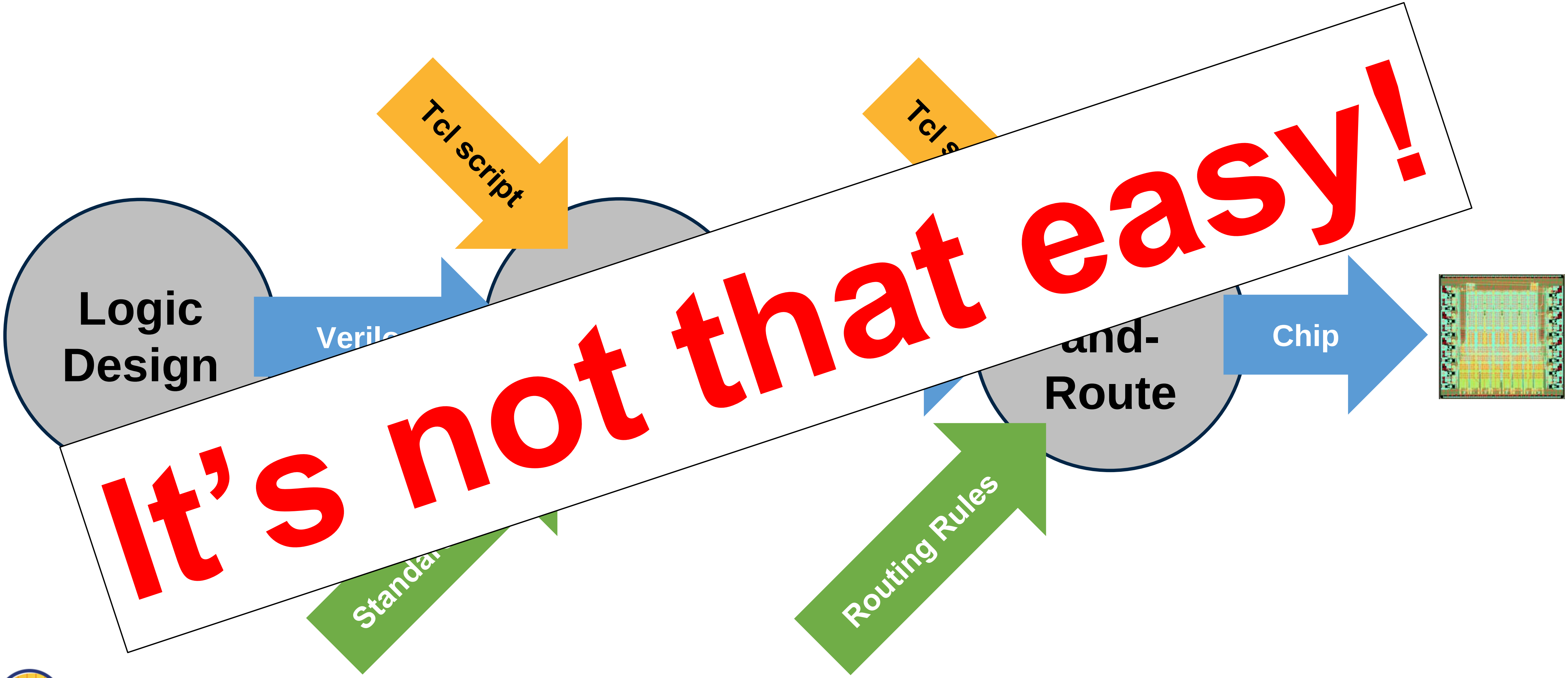
Berkeley Architecture Research

CHIPYARD

Motivation: “Advertised” VLSI Flows

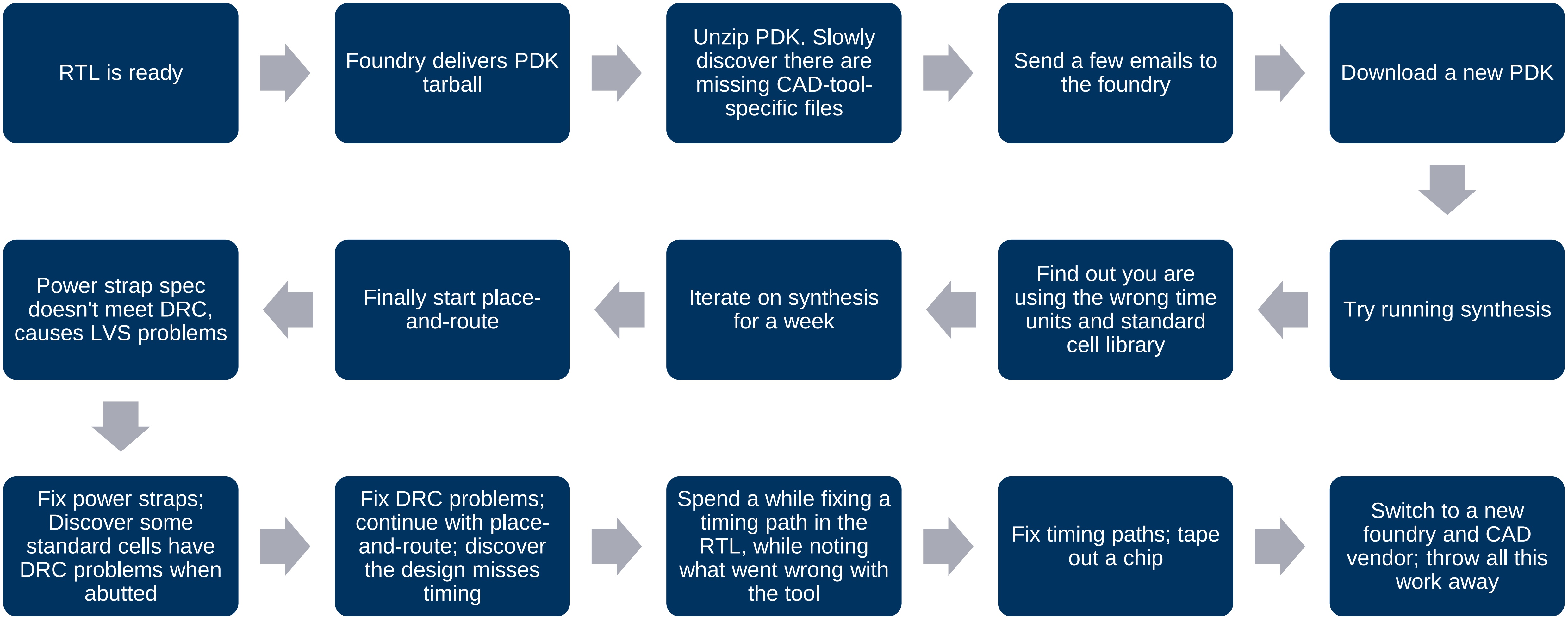


Motivation: “Advertised” VLSI Flows



It’s not that easy!

Motivation: Real VLSI Flows



RTL is ready

Foundry delivers PDK tarball

Unzip PDK. Slowly discover there are missing CAD-tool-specific files

Send a few emails to the foundry

Download a new PDK

Try running synthesis

Find out you are using the wrong time units and standard cell library

Iterate on synthesis for a week

Finally start place-and-route

Power strap spec doesn't meet DRC, causes LVS problems

Fix power straps; Discover some standard cells have DRC problems when abutted

Fix DRC problems; continue with place-and-route; discover the design misses timing

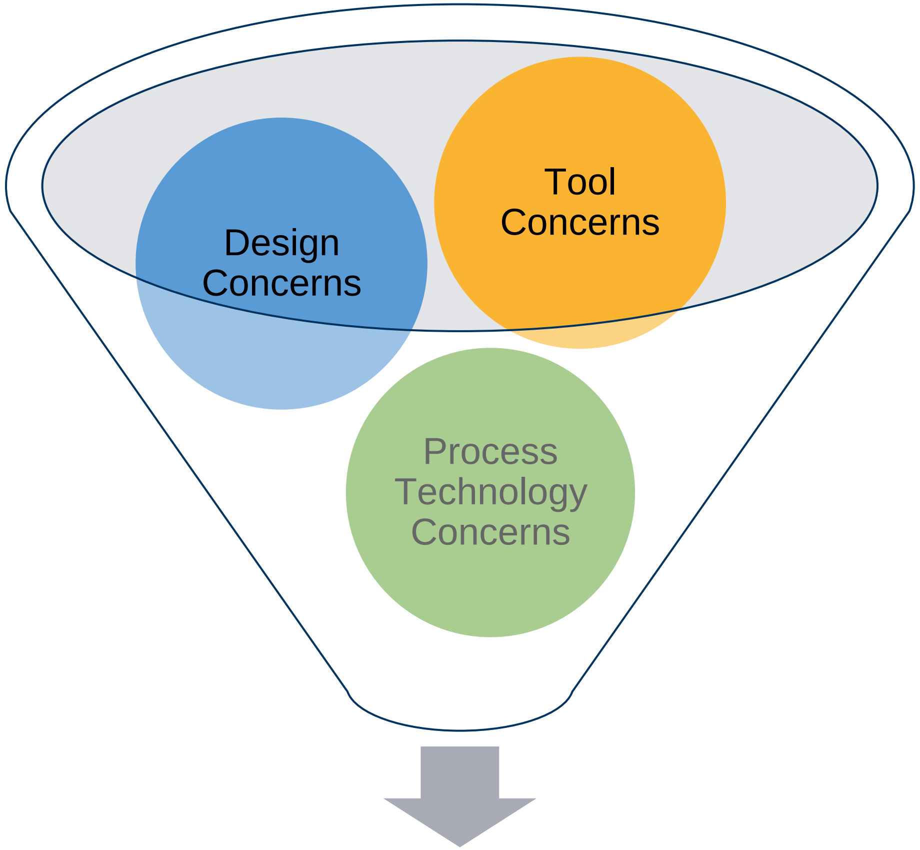
Spend a while fixing a timing path in the RTL, while noting what went wrong with the tool

Fix timing paths; tape out a chip

Switch to a new foundry and CAD vendor; throw all this work away

Motivation: Real VLSI Flows

- **Problem**: VLSI flows must be **rebuilt** for each project
- Overhead compounded by
  - Changing CAD tools
    - Commands / features change
    - File formats / library locations
  - New process technology
    - SRAMs (compiled/pre-generated?)
    - DRC rules
  - Different design
    - Floorplanning / power / clock

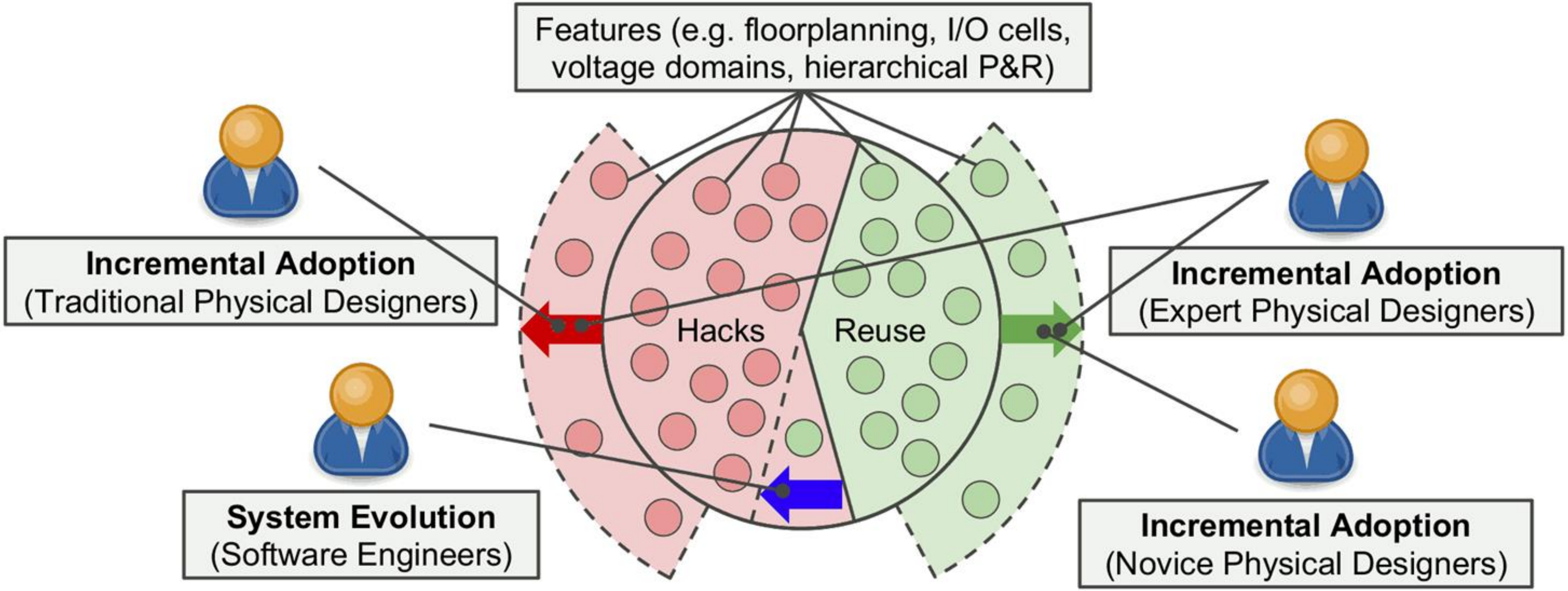


Non-reusable Tcl script

Solution: Hammer

- **Goal**: generate modular, reusable VLSI flows
- **Philosophy**:
  - Incremental Adoption
    - Users: reuse what you can, hack what you can’t
  - System Evolution
    - Devs: generalize hacks for future users
  - Modularity + Abstraction = Clarity
    - Separate concerns, standardize data exchange

Hammer Design Philosophy


Features (e.g. floorplanning, I/O cells, voltage domains, hierarchical P&R)
Incremental Adoption (Traditional Physical Designers)
Hacks
Reuse
Incremental Adoption (Expert Physical Designers)
System Evolution (Software Engineers)
Incremental Adoption (Novice Physical Designers)

Using **modularity** and **abstractions**, HAMMER is designed to support **incremental adoption** and **system evolution**.

Hammer Design Philosophy

- **Separation of Concerns**
  - 3 input categories:
    1. Design-specific
    2. Tool/Vendor-specific
    3. Technology-specific
- **Hammer IR**
  - Standard YAML/JSON input & data exchange format
  - Metaprogramming: modifiable attributes
- Modular tech & tool **plugins**
  - Default settings, flow steps, helper methods
  - Interchangeable + extensible = reusable!

Design:

- Floorplan
- Clocks
- Hierarchy

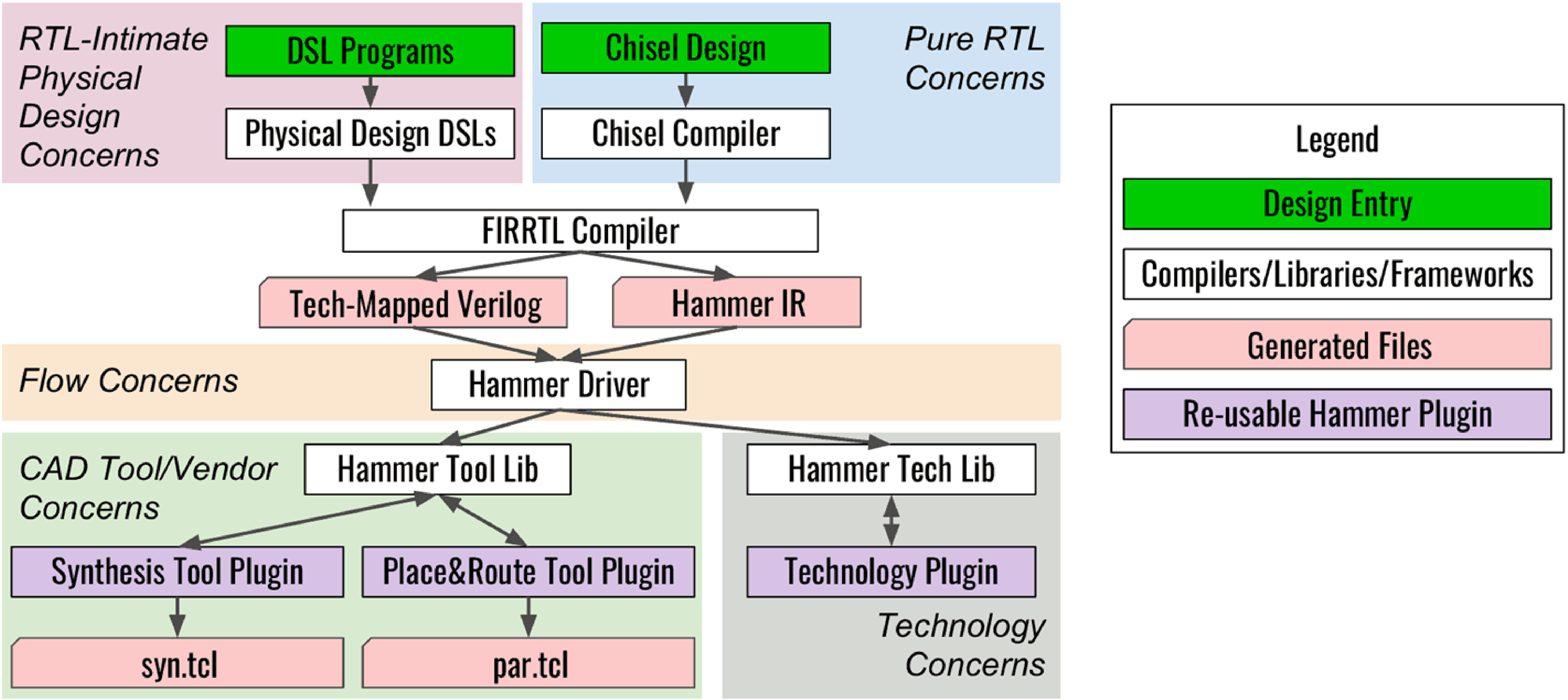
Tool:

- In/out files
- Tcl code
- Tech. file formats

Tech.:

- SRAMs
- Std. cells
- Stack-up
- Power straps

Software Architecture



History of Hammer

Palmer Dabbelt's PLSI

- Make-based
- *Stringly*-typed
- JSON dictionaries
- Not reused

Edward Wang's MS Thesis

- Re-architected
- *Strongly*-typed
- Clear abstractions
- EAGLEs (TSMC 16)
- Tapeout class (ST 28)
- Published: ISQED '20

UCB-BAR project + Chipyard integration

- Many more devs
- Documentation
- New plugins & features
  - RTL/gate-level simulation
  - DRC/LVS
  - Power, EM/IR analysis
  - SRAM, PCB collateral
  - APIs: power straps, bumps, pins, floorplan

Projects/Classes	Technology
Hydra, BEAGLE, PRIDE	Intel 22
ARGO (RTML)	GF 12
151/251A, 241B	ASAP 7
290C (tapeout), HDC	TSMC 28
Oski Bear	Skywater 130

Current State + Future of Hammer

- Hammer is for everyone!
  - Tape out in many process techs
  - Arch. DSE with open-source techs
  - Reusability regardless of design
- Current key features:
  - Bottom-up hierarchical
  - SRAM library compilation
  - Many APIs: e.g., power straps
  - Floorplan visualization
  - User & tech hooks (i.e., hacks)

- What we’re planning:
  - Metrics parsing, constraints feedback
  - OpenROAD plugins
  - LEC, characterization plugins
  - Aspect-oriented Chisel floorplanning
  - Abutment/partition-based hierarchical
  - True multi-clock/power domain gen.
  - Cloud compute, CI, IR validity checks

Hammer users are also developers!
→ driven by projects + tapeouts

Learn More

- Github: https://github.com/ucb-bar/hammer/
- Documentation: https://hammer-vlsi.readthedocs.io/
- Chipyard-specific documentation: https://chipyard.readthedocs.io/en/dev/VLSI/index.html
- User mailing list: hammer-users@googlegroups.com
- Plugin access requests: hammer-plugins-access@lists.berkeley.edu
  - Cadence, Synopsys, and Mentor