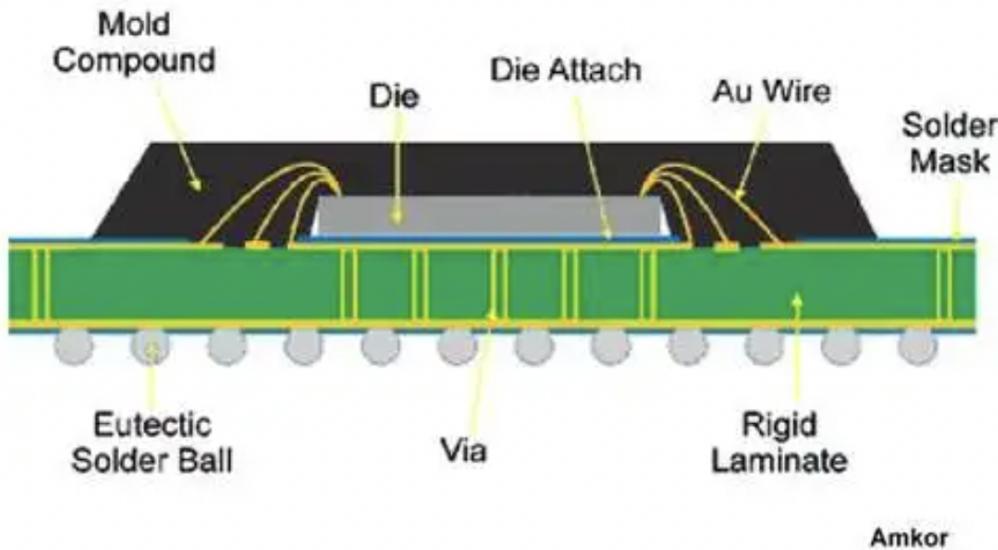


Post-Tapeout Bringup & Research Landscape

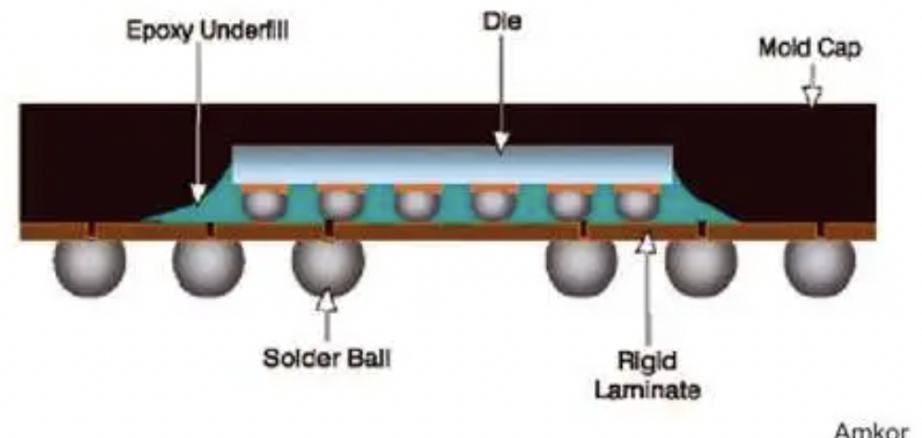
98-154/18-224/18-624: Intro to Open-Source Chip Design

Post-Manufacturing: Packaging

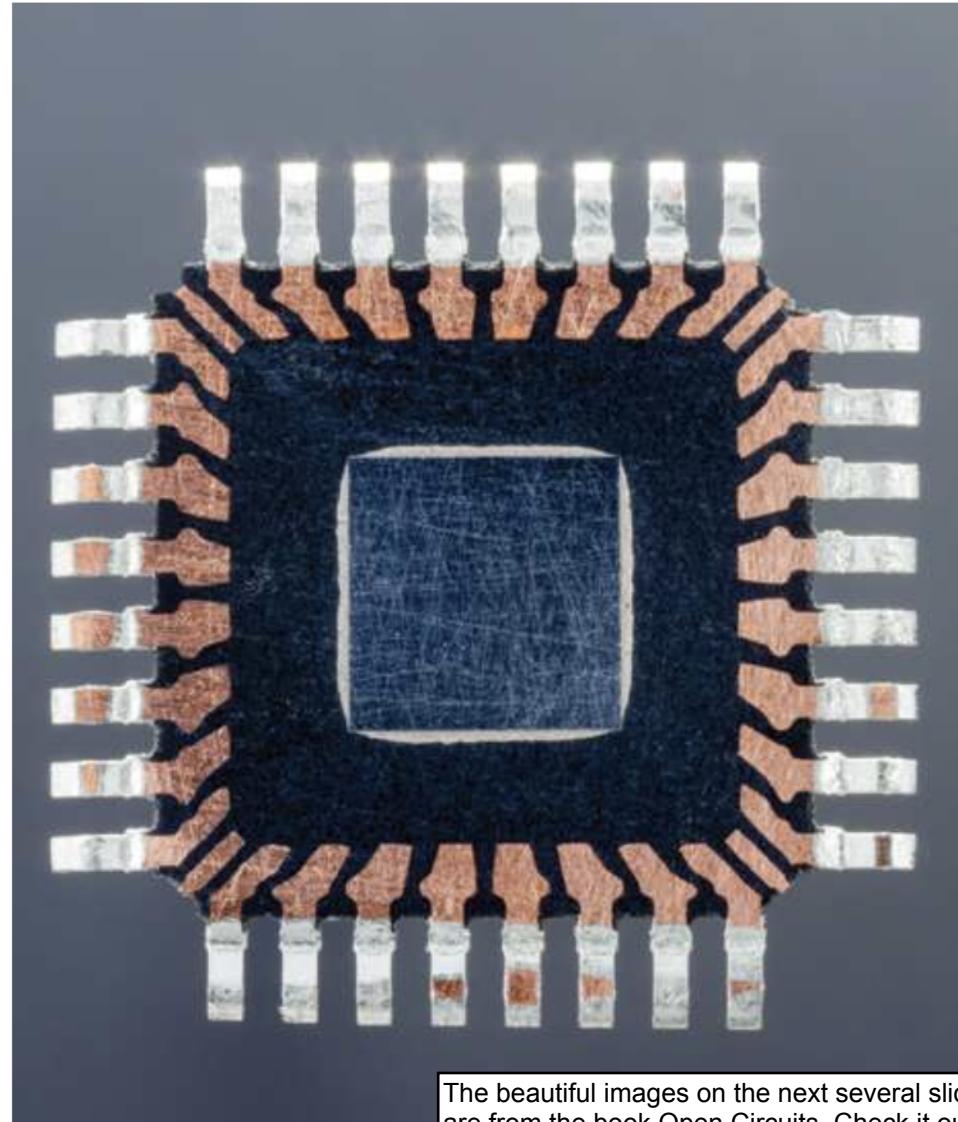
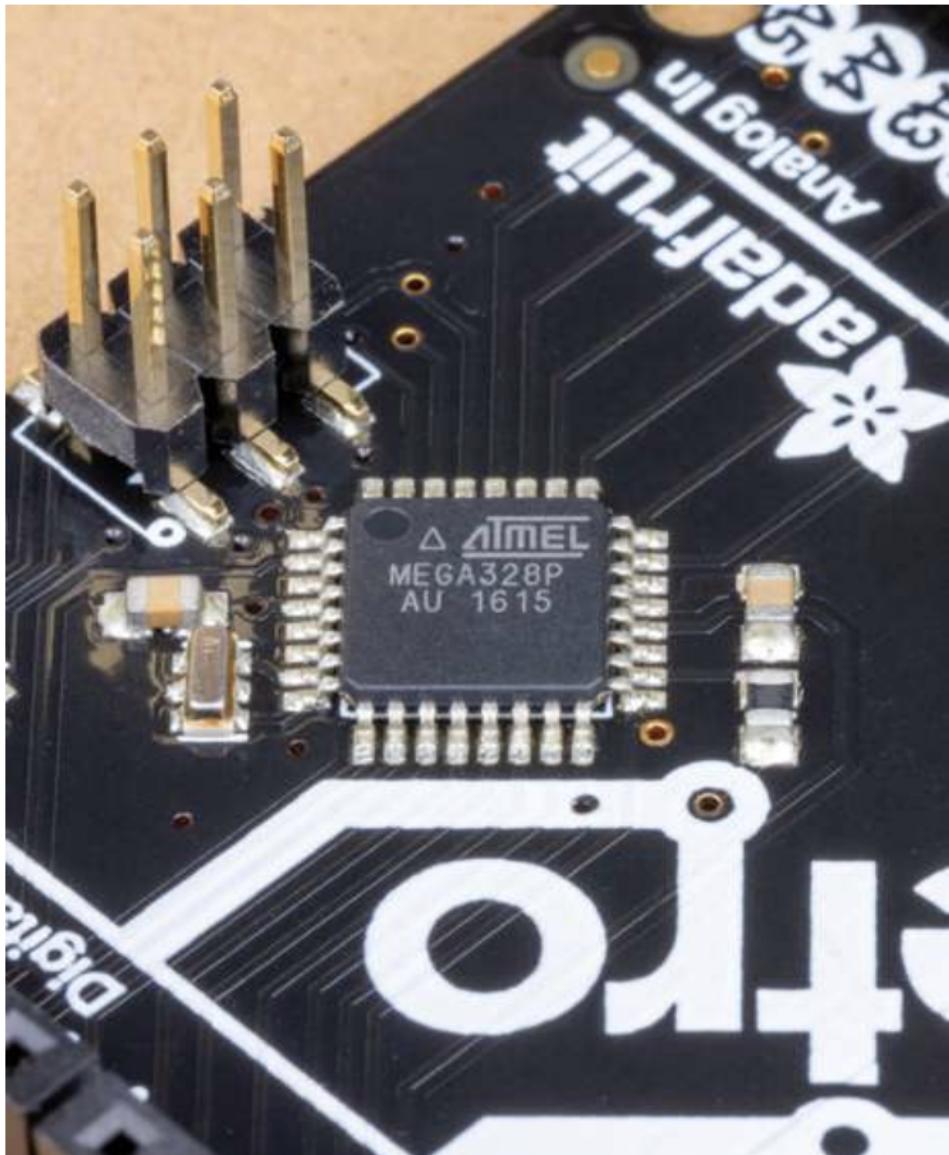
Typical Multi-tier Wire Bond (PBGA) Package



Flip Chip PBGA (FC-PBGA)

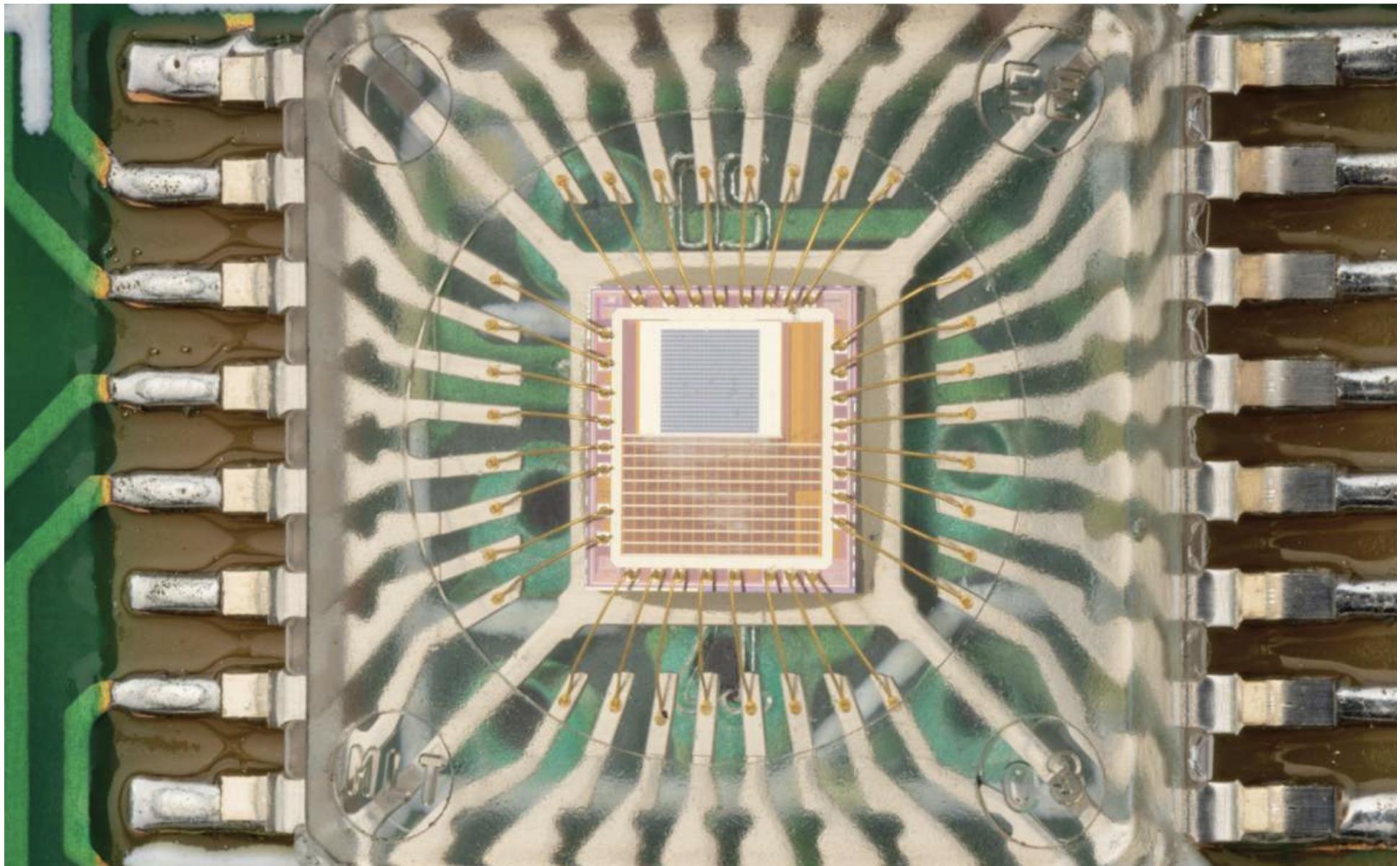


Post-Manufacturing: Packaging



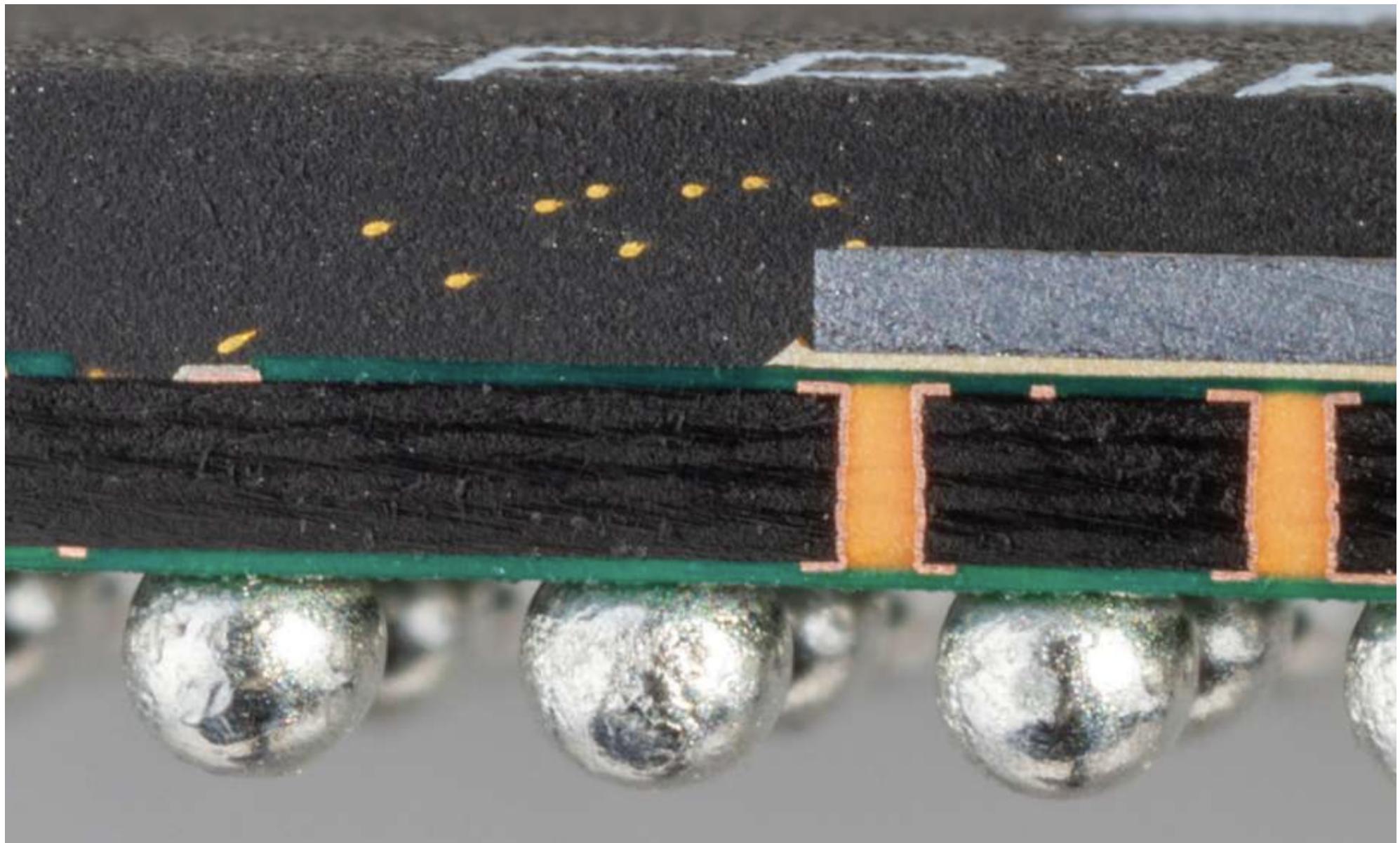
The beautiful images on the next several slides are from the book Open Circuits. Check it out if you're interested in how electronics work!

Post-Manufacturing: Packaging



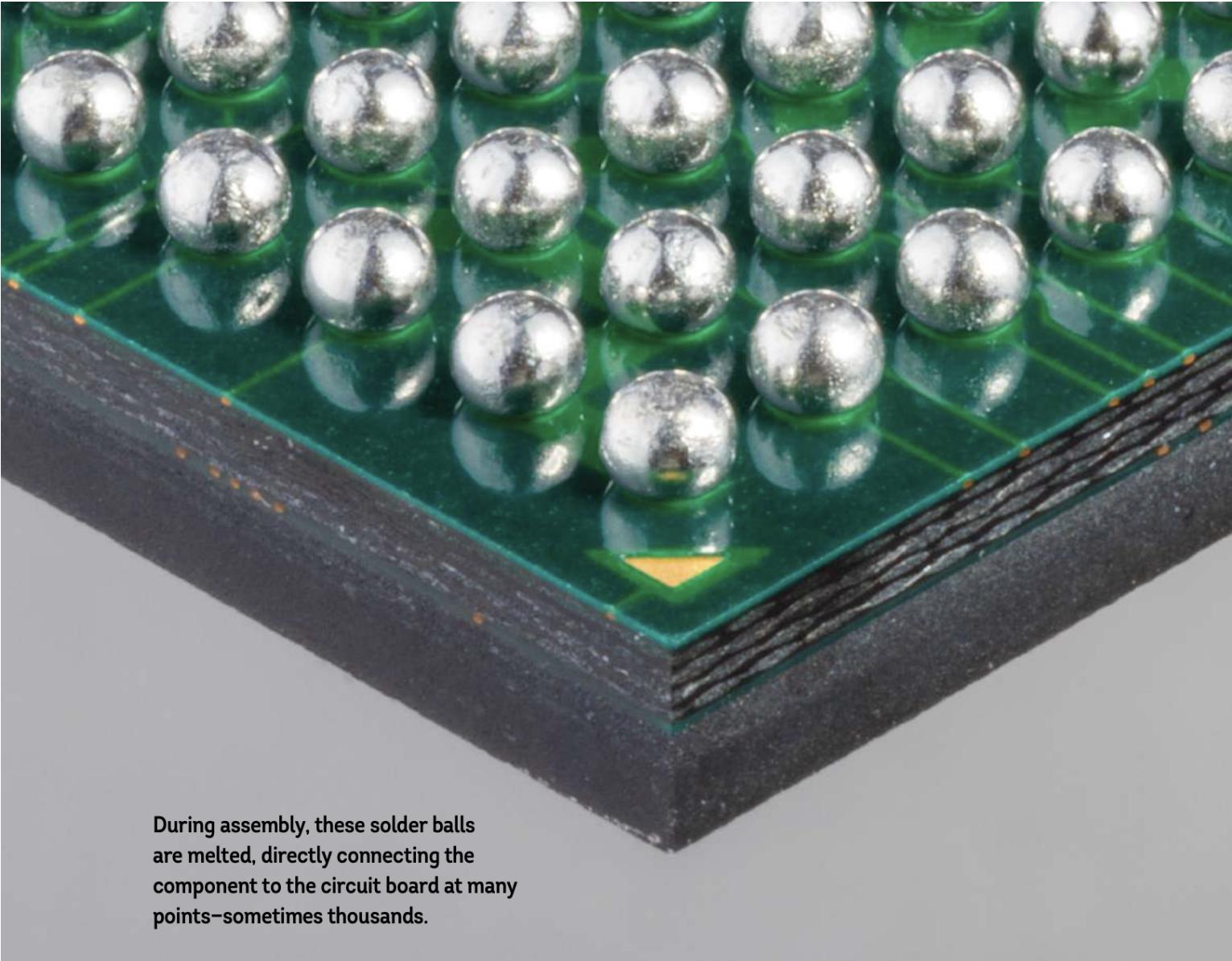
98-154/18-224/18-624: Intro to Open-Source Chip Design

Post-Manufacturing: Packaging



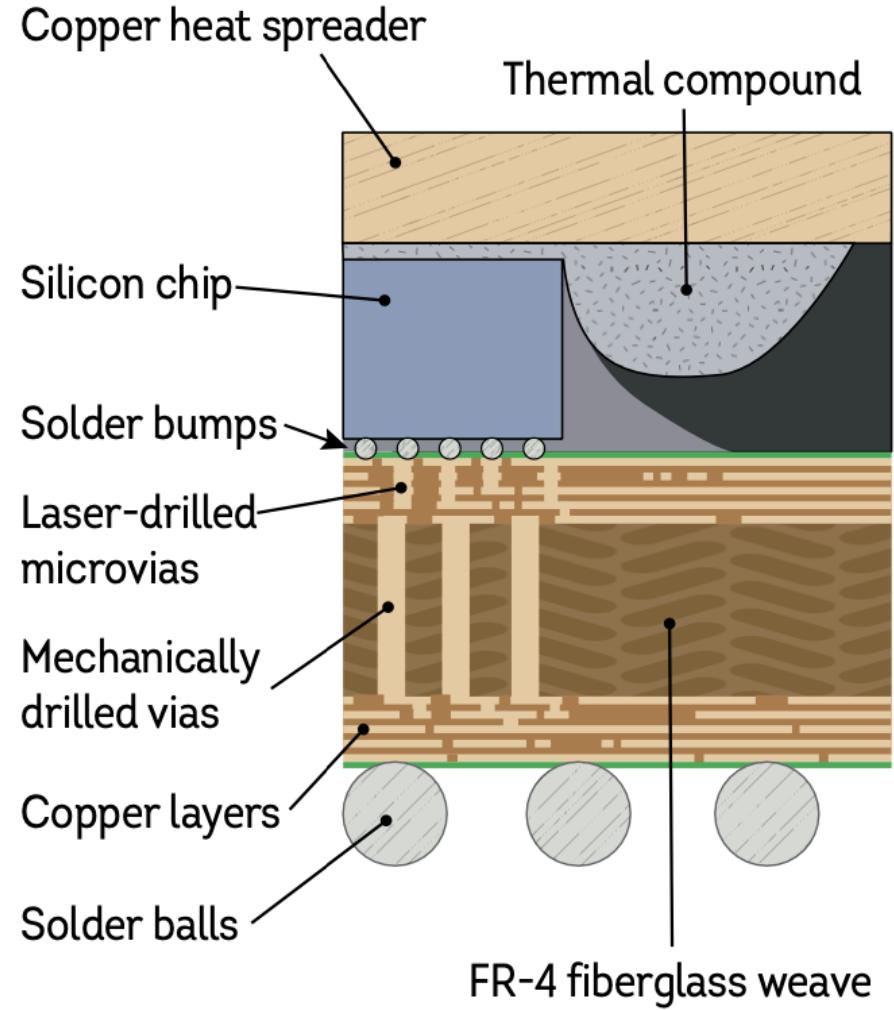
98-154/18-224/18-624: Intro to Open-Source Chip Design

Post-Manufacturing: Packaging

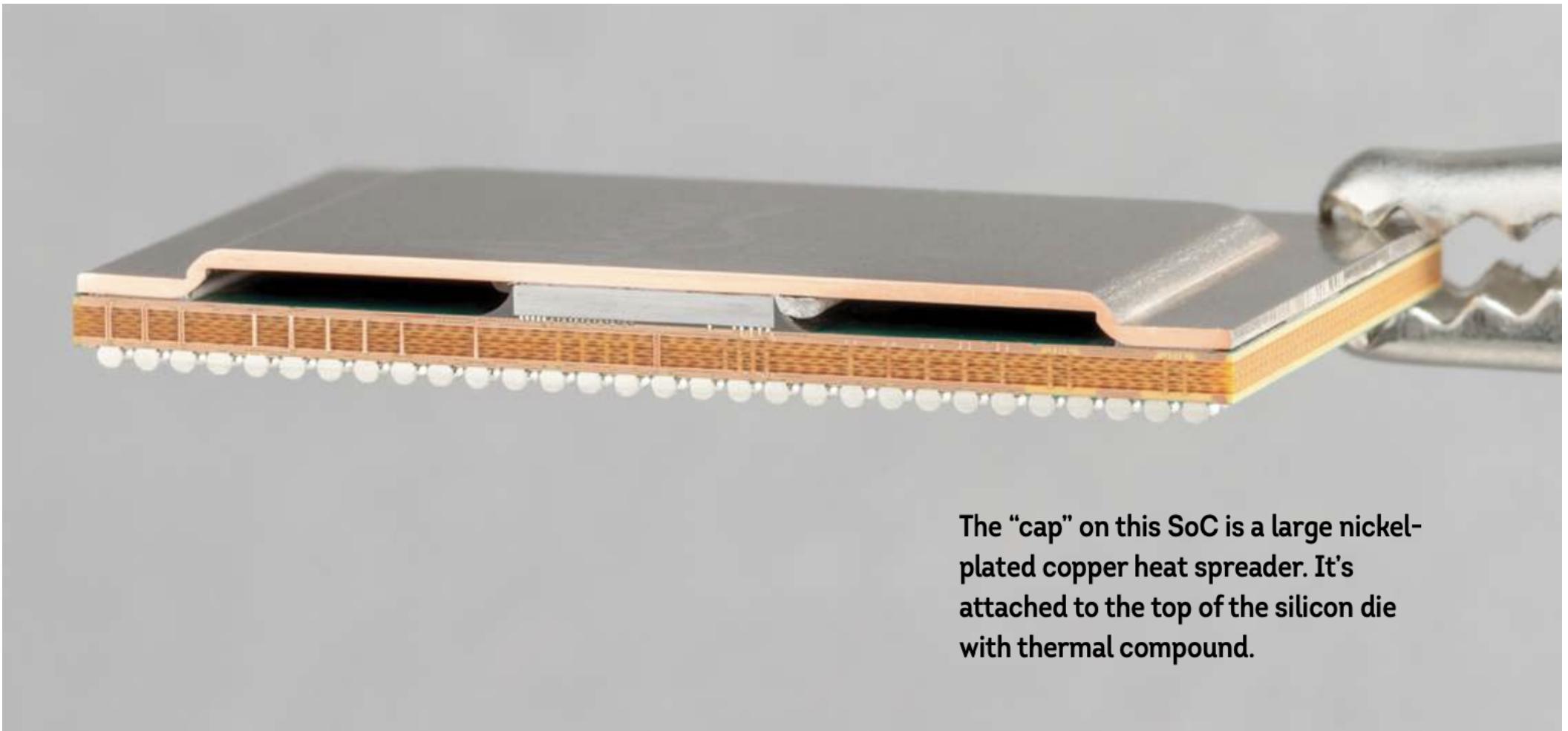


During assembly, these solder balls are melted, directly connecting the component to the circuit board at many points—sometimes thousands.

Post-Manufacturing: Packaging

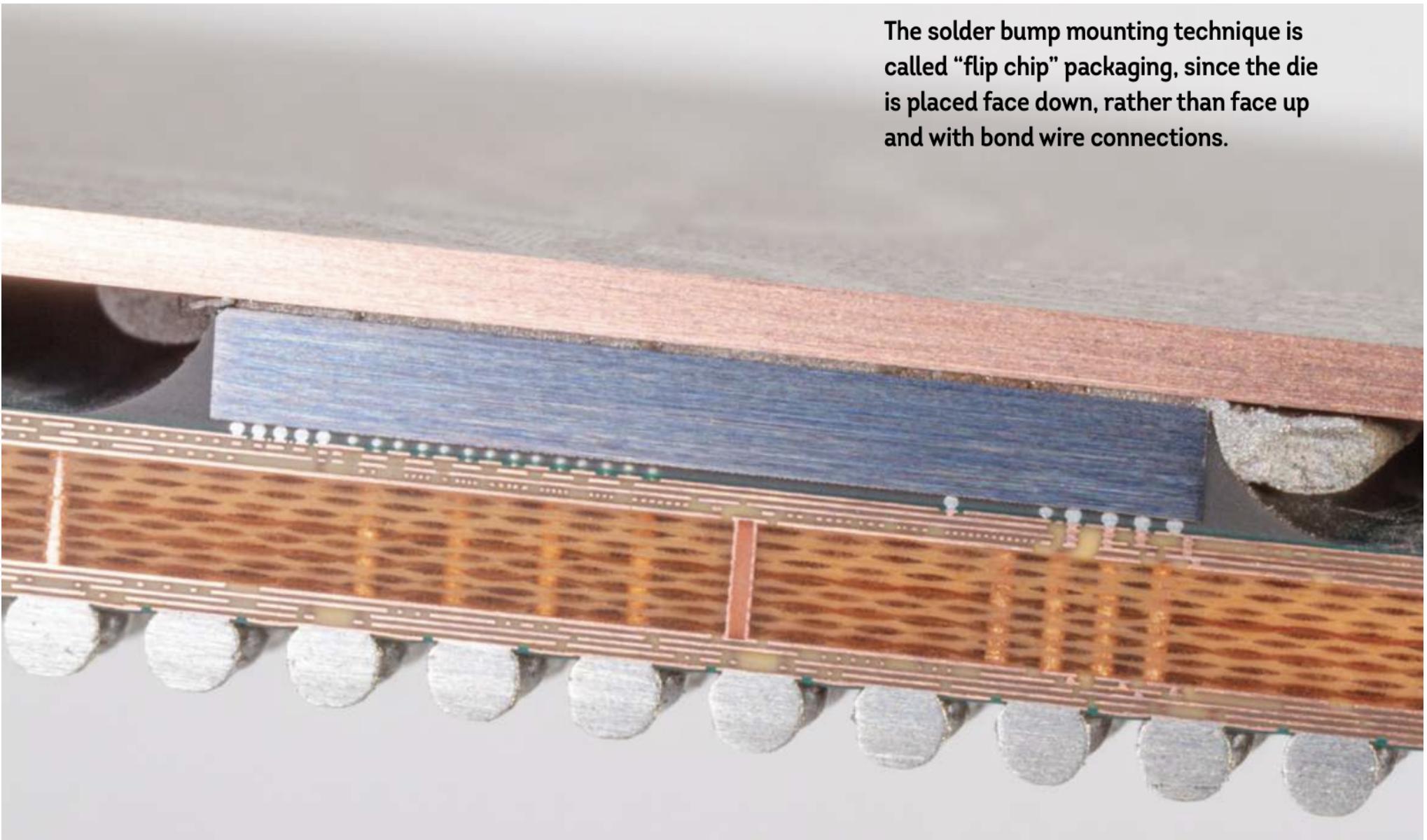


Post-Manufacturing: Packaging



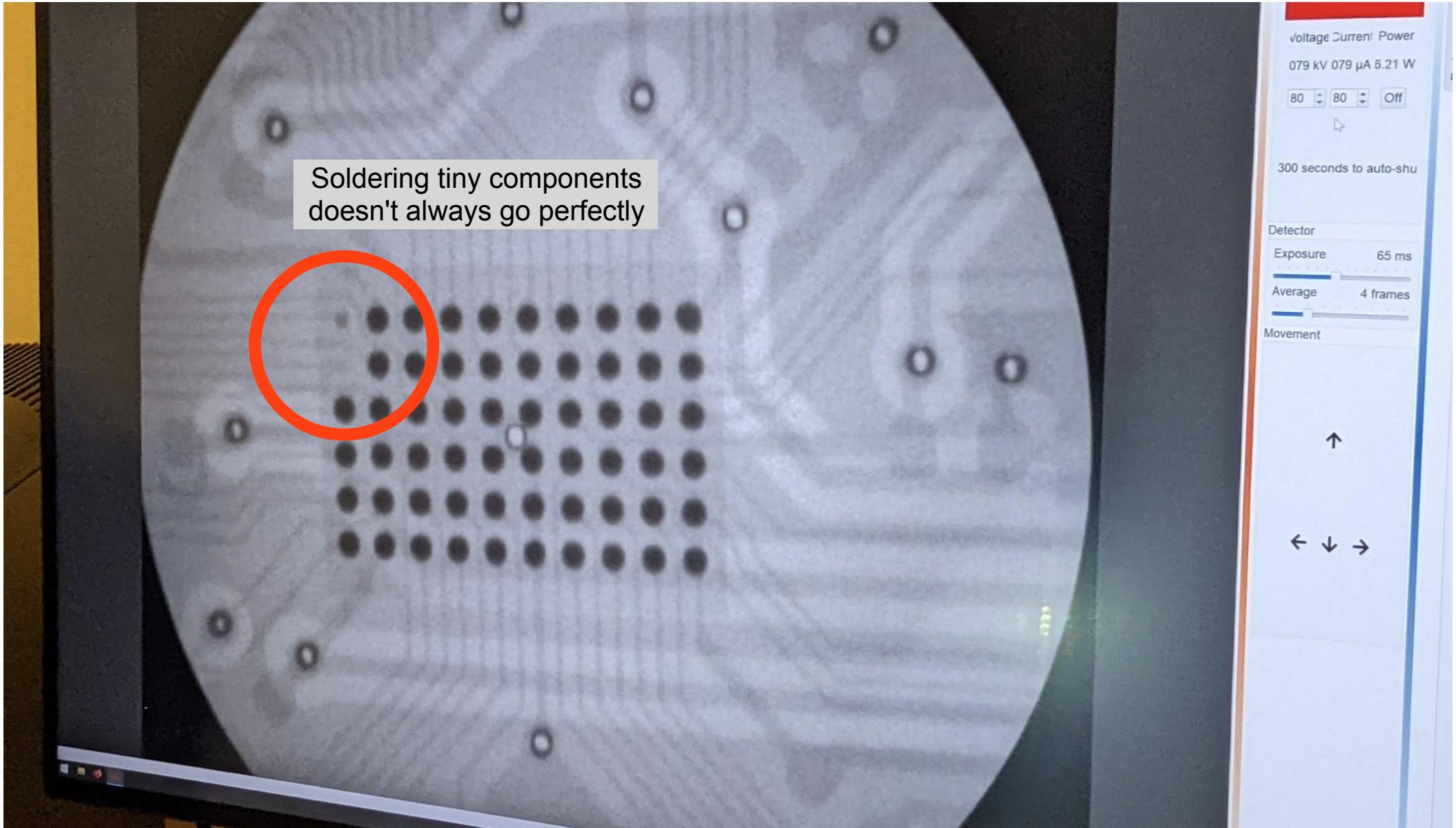
The “cap” on this SoC is a large nickel-plated copper heat spreader. It’s attached to the top of the silicon die with thermal compound.

Post-Manufacturing: Packaging

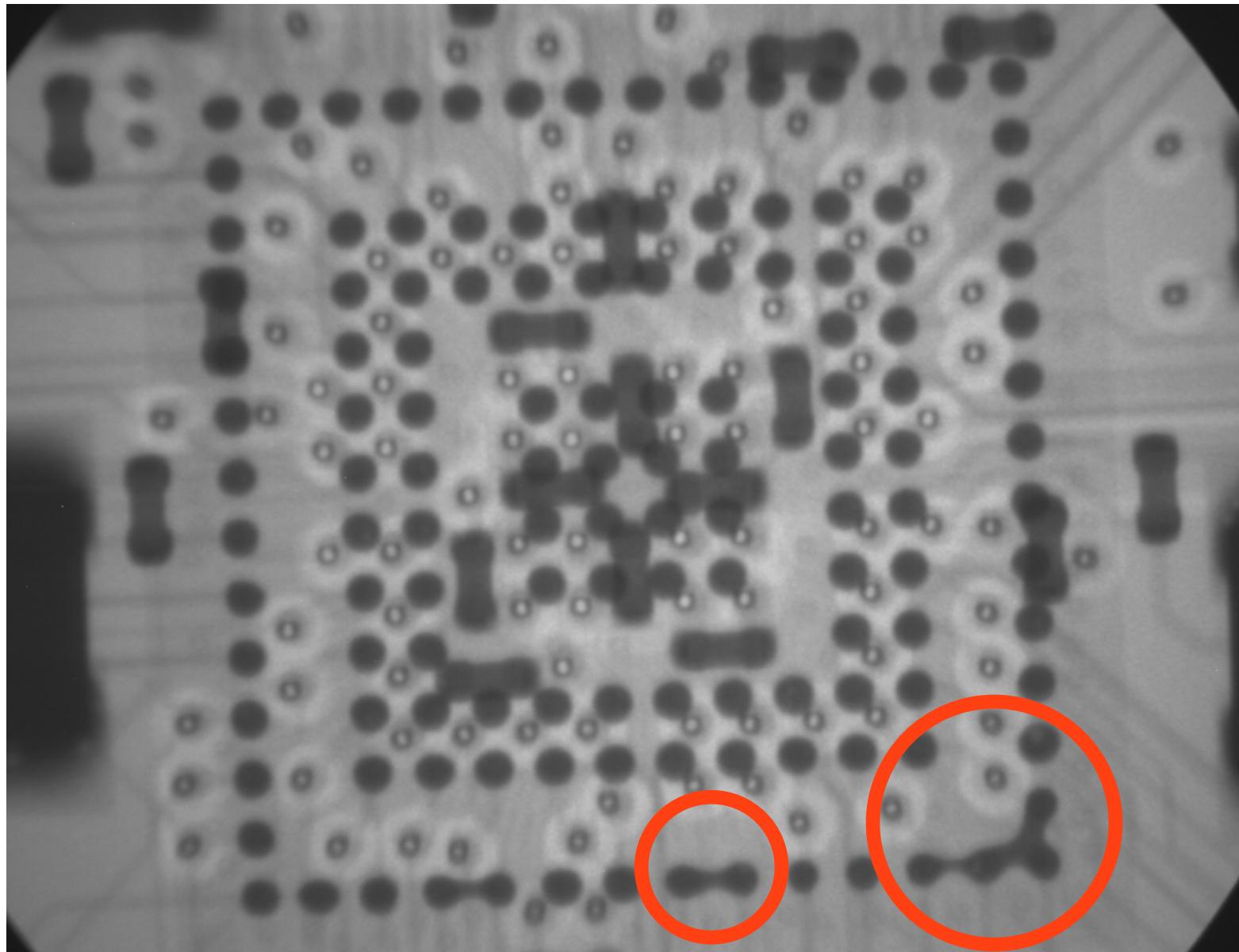


The solder bump mounting technique is called “flip chip” packaging, since the die is placed face down, rather than face up and with bond wire connections.

Post-Manufacturing: Assembly

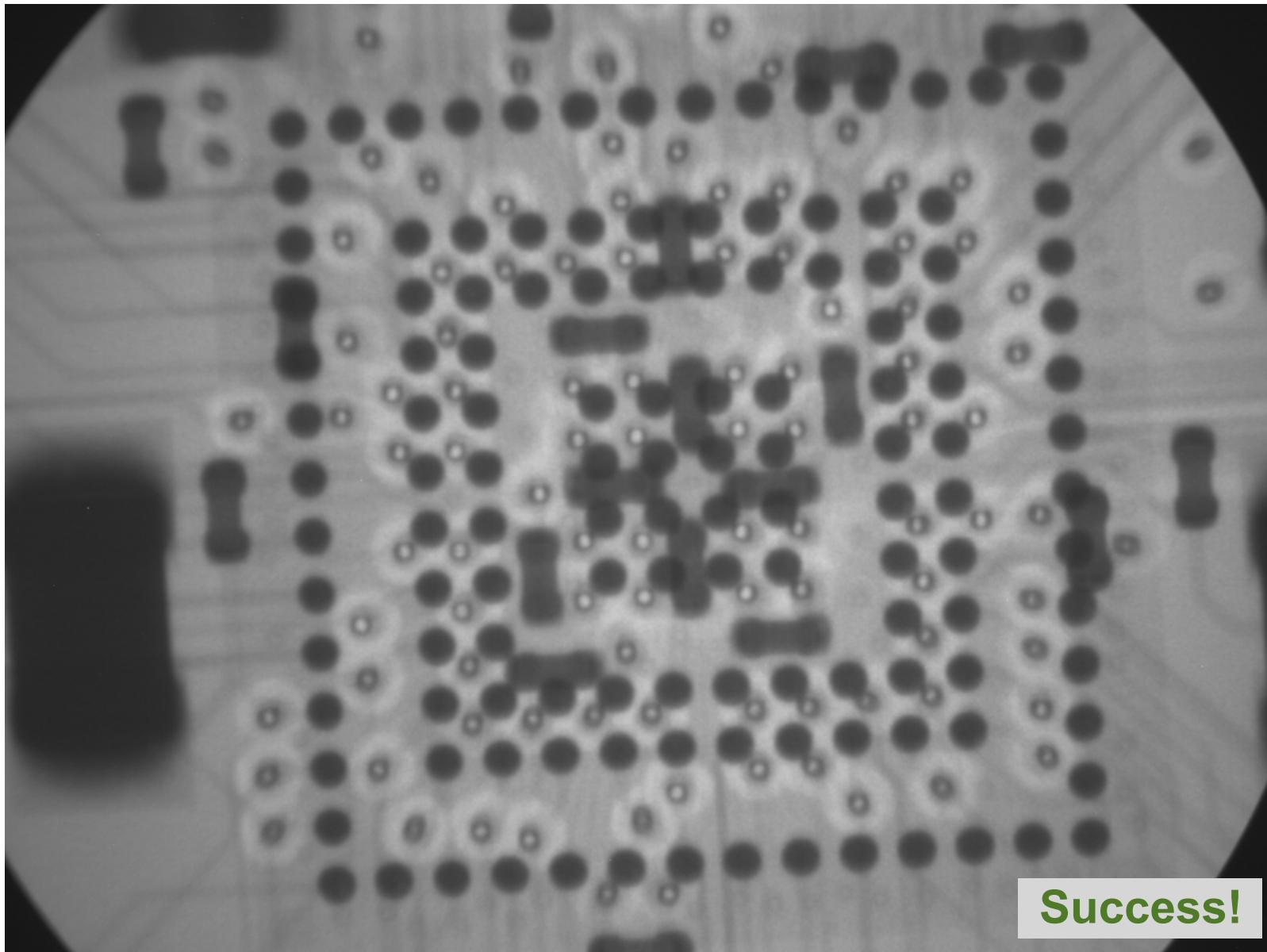


Post-Manufacturing: Assembly



98-154/18-224/18-624: Intro to Open-Source Chip Design

Post-Manufacturing: Assembly



JTAG & Scan Chains

- Manufacturing a chip has long lead-time, can't re-manufacture if you find a bug
- Chip is tiny, can't probe individual wires inside the physical chip
- Design For Test = add testing infrastructure into design, to help with bug-finding and also for individual-chip validation

Boundary Scan

Scanning the I/Os of a chip

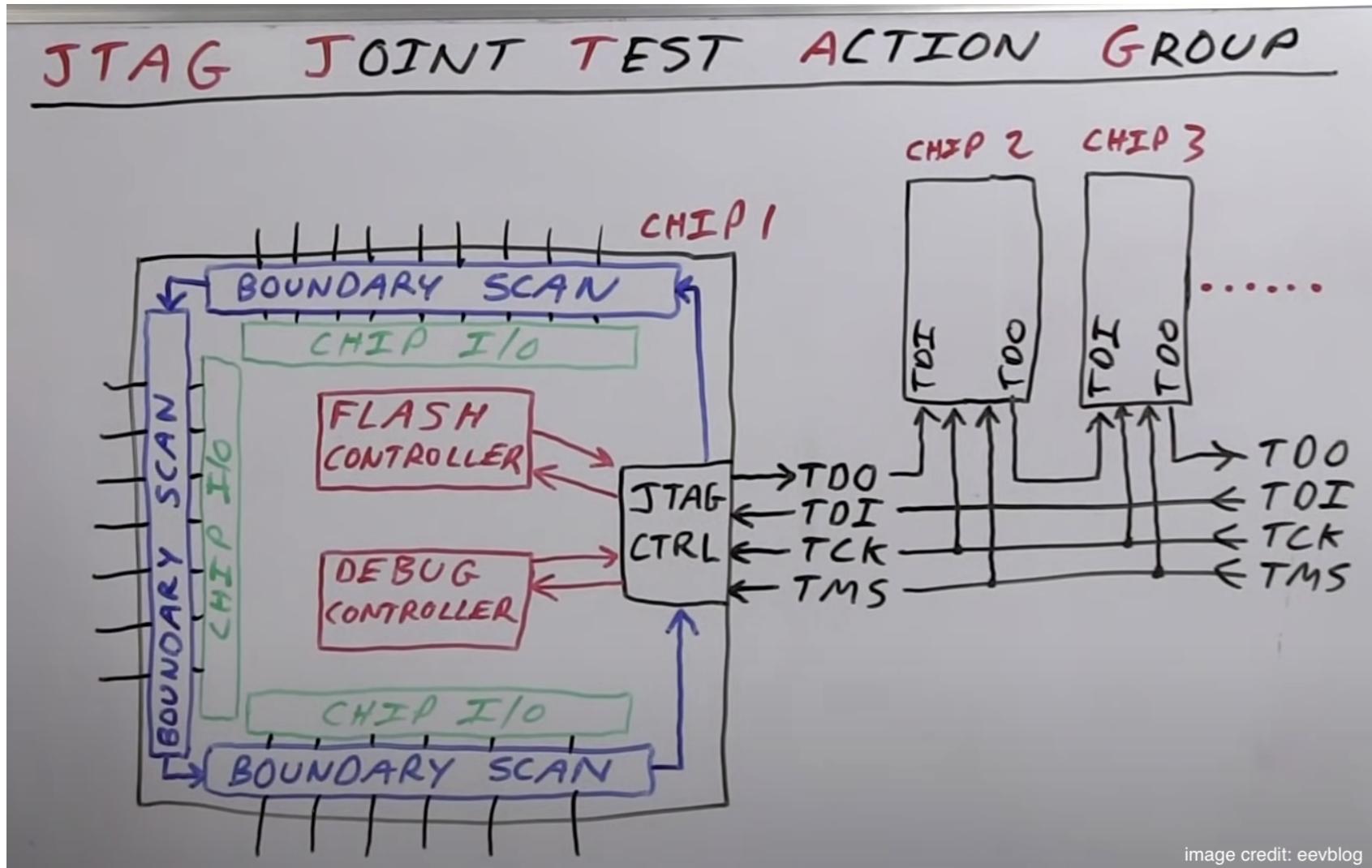
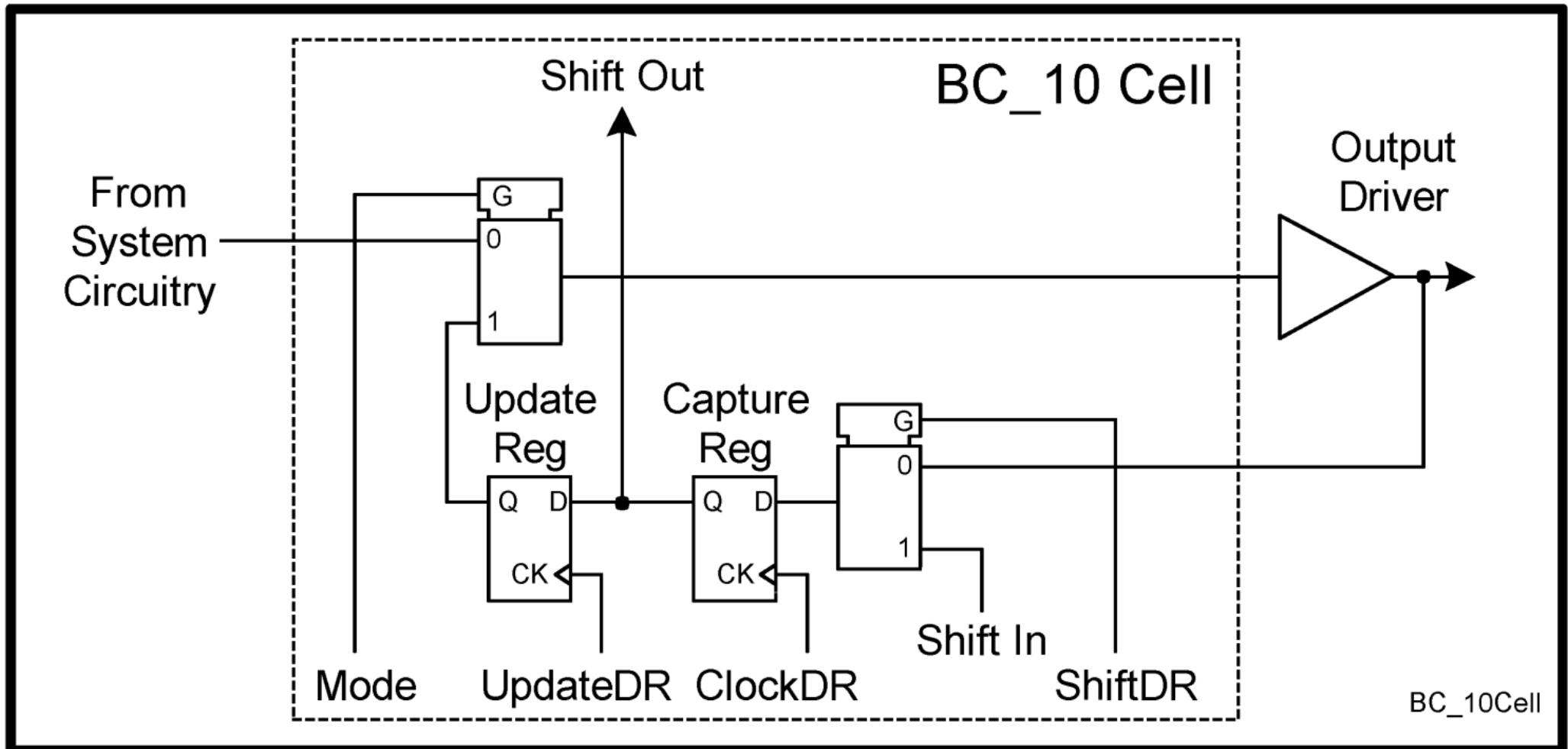


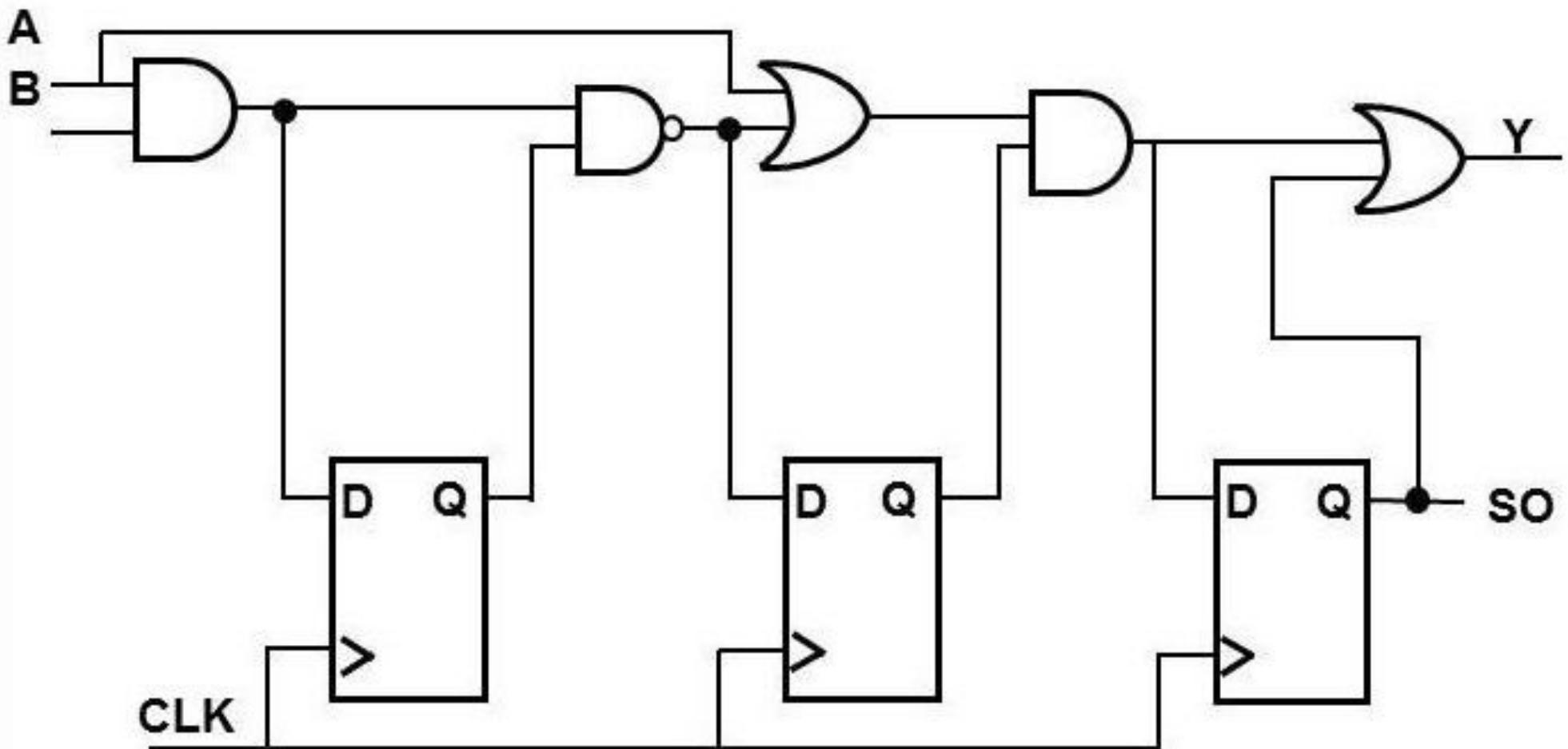
image credit: eevblog

Boundary Scan Output Cell



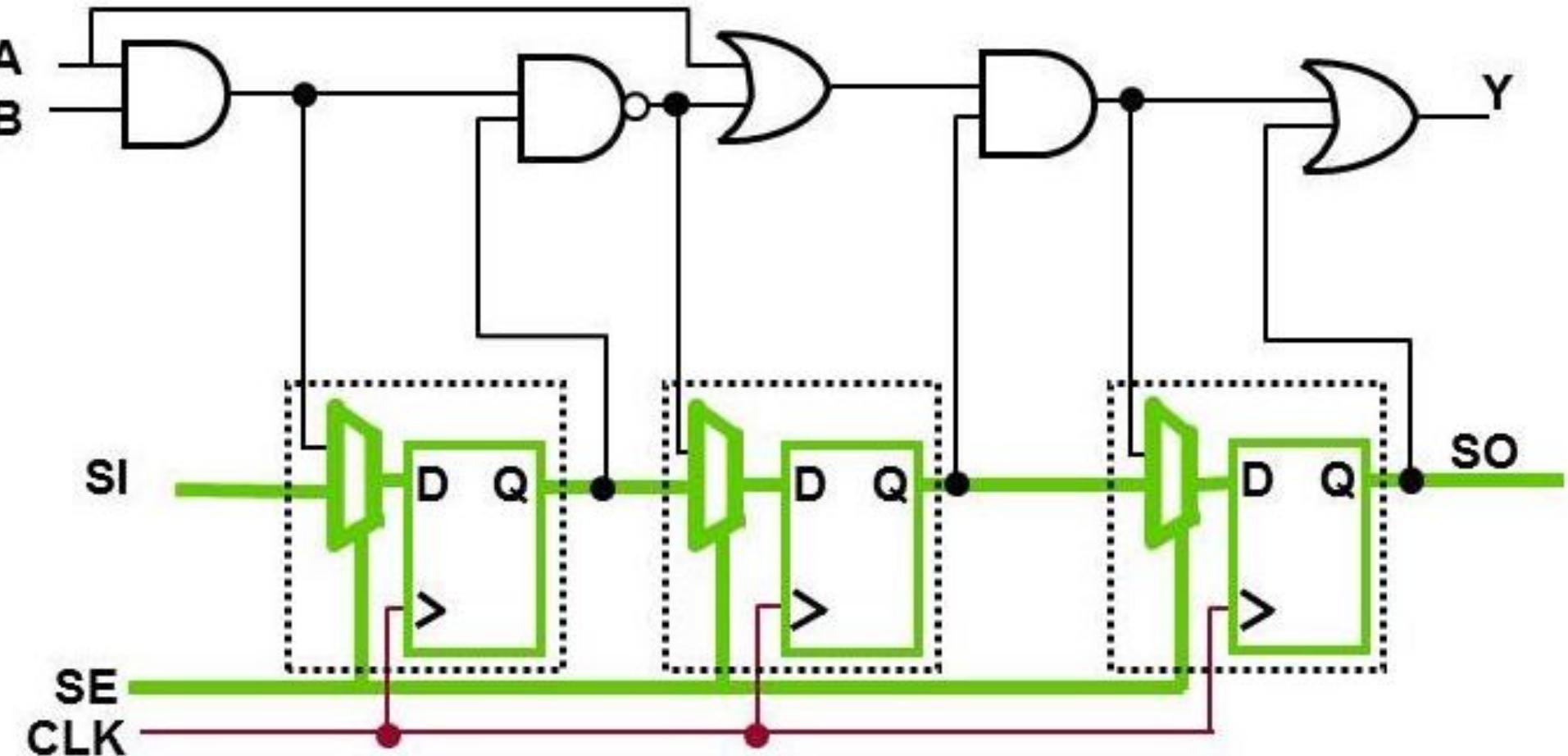
Scan Test Insertion

Scanning internal registers of a chip



Scan Test Insertion

Scanning internal registers of a chip



Scan Test Waveforms

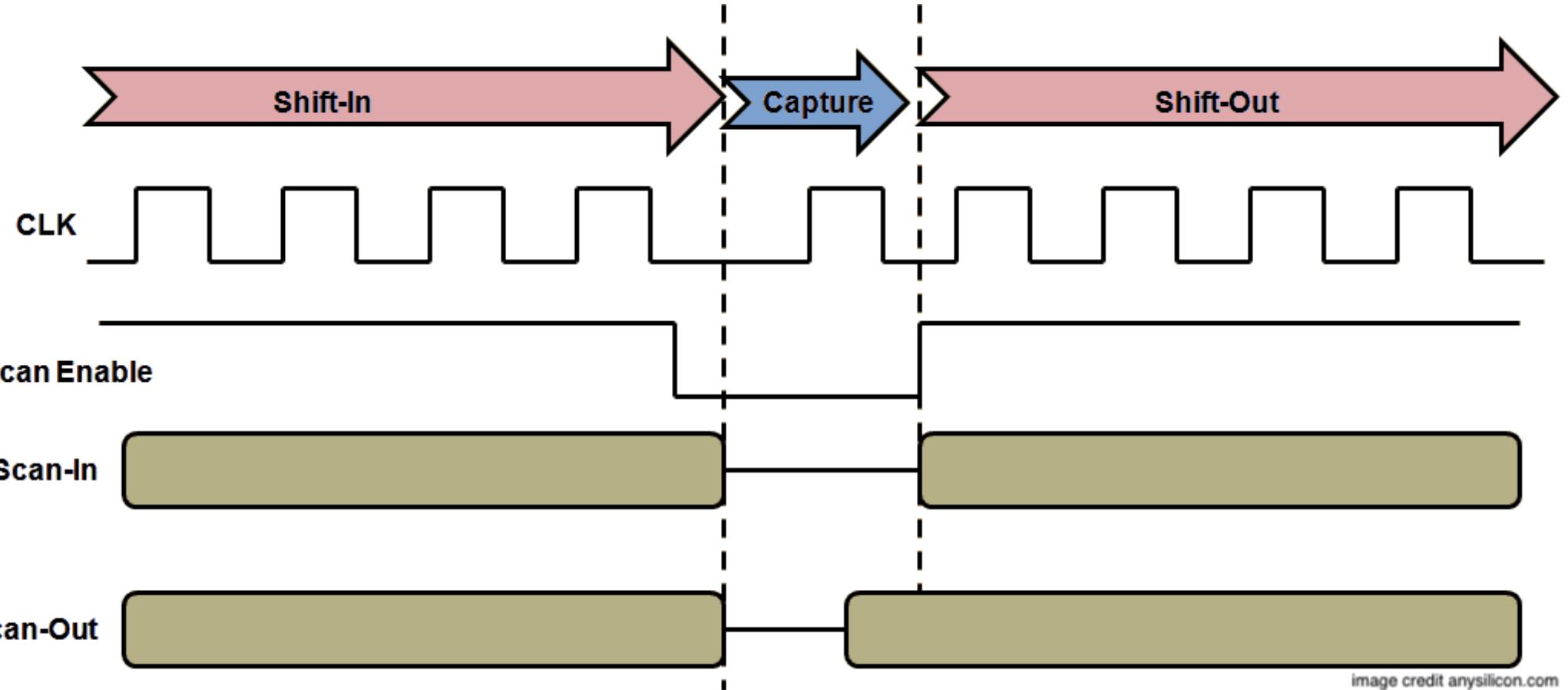
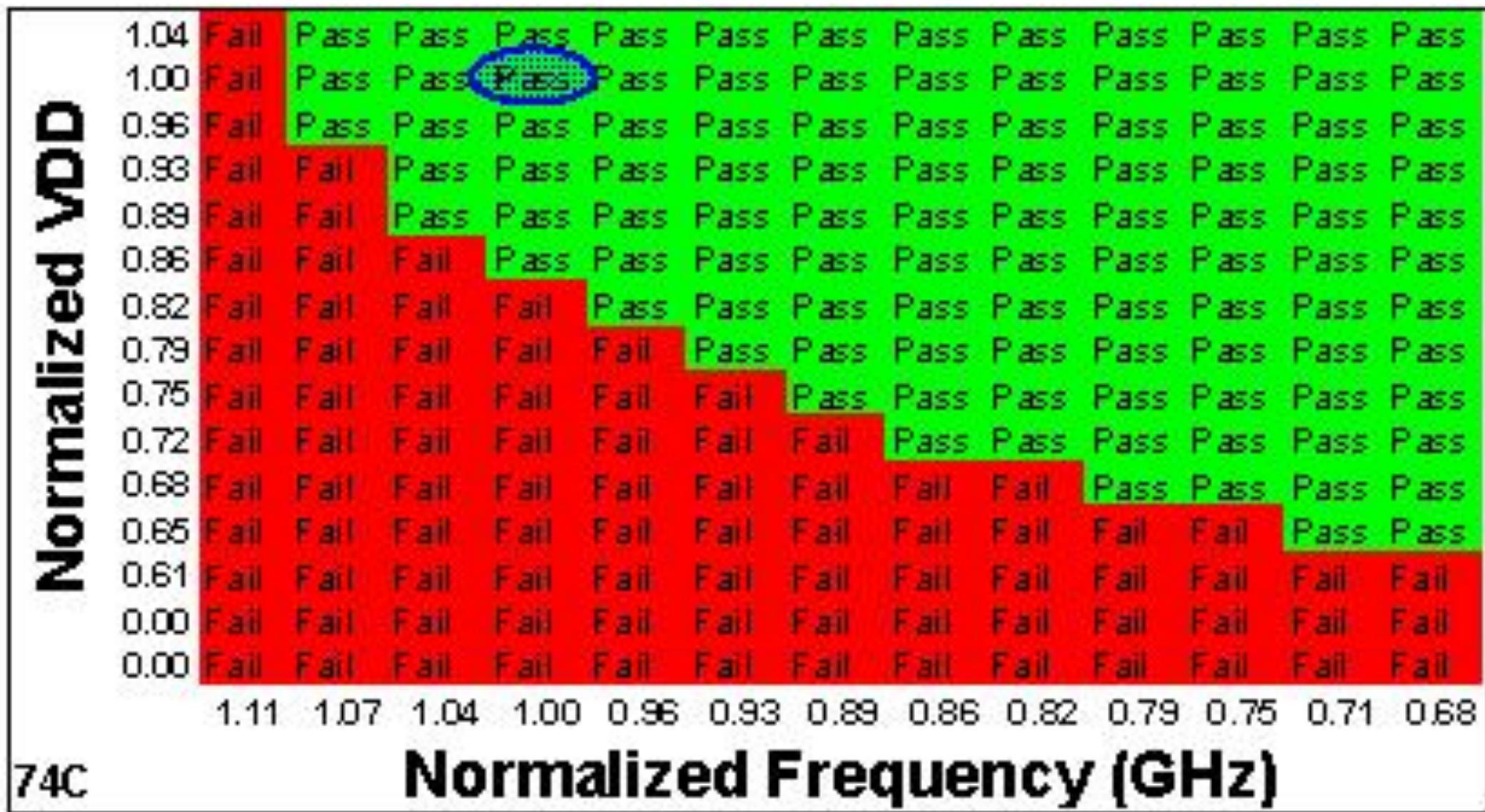


image credit anysilicon.com

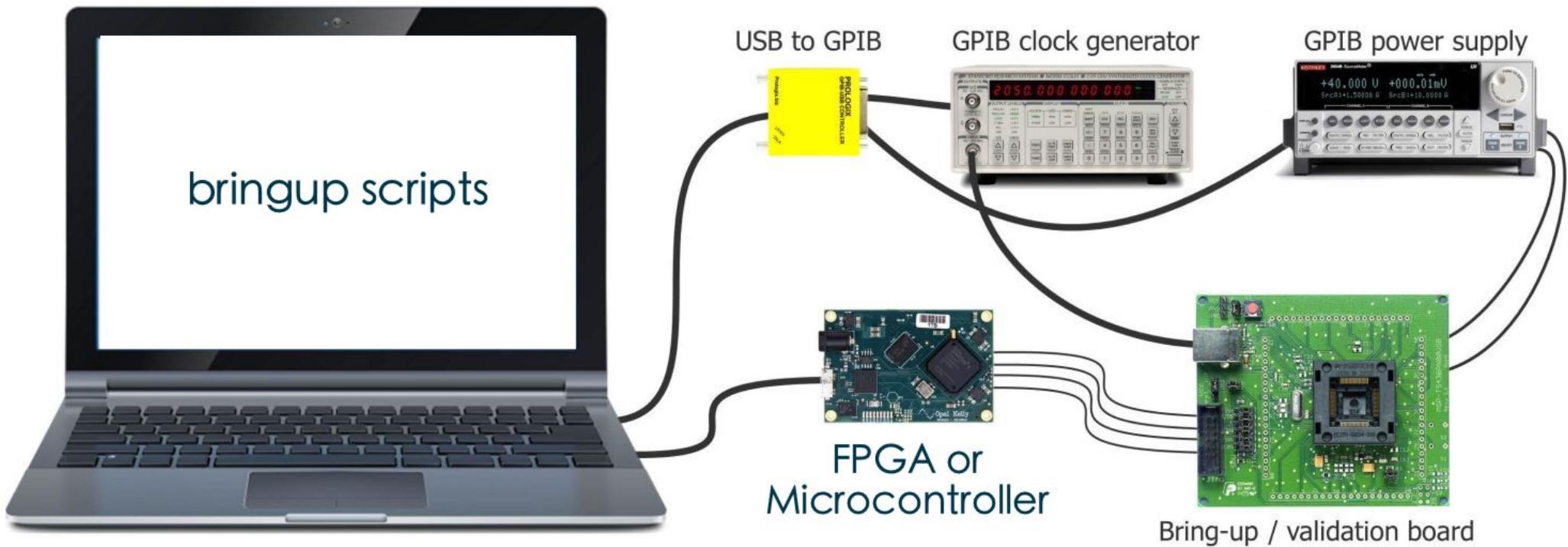
Bringup Steps

- Initial bringup
 - Interfacing with the chip, enabling debug interfaces
- Silicon validation
 - Verify functionality via test-vectors, DFT, etc.
- Characterization
 - Find limits of voltage and clock-speed
- Integration
 - Integrate with software/firmware/associated system

Characterization: Shmoo Plot



Bringup Lab Setup



Open-Source Bringup Lab

- Sigrok: “a portable, cross-platform, Free/Libre/Open-Source signal analysis software suite”
 - Integrates with logic analyzers, scopes, power supplies, function generators, multimeters, etc
 - [Sigrok Overview](#)
- ADALM2000 / Digilent Analog Discovery
- Saleae Logic

Research Literature

- Lots of work coming out of the academic world on building open-source platforms for open-source EDA/CAD (electronic design automation / computer-aided design) tools
- Surprisingly, a lot of these platforms are quite well-built (not the standard “clobbered together in time for a paper deadline”)
- WOSET (Workshop on Open-Source EDA Technology), colocated with IEEE/ACM ICCAD
 - Focused on *practical* work in open-source tooling
 - Papers tend to be quite readable and not very dense

WOSET 2020

CVC: Circuit Validity Checker: An open source netlist reliability verification system

- Automated analog-level validity checker (for digital designs), without user having to manually specify error conditions
- Checks for errors such as MOSFET pinouts, diode biasing, leaks, current-flow paths, expected values, power nets, etc.
- (paper)

WOSET 2020

OpenPhySyn: An Open-Source Physical Synthesis Optimization Toolkit

- Automated tooling for physical synthesis, specifically resolving DRC and timing issues using efficiently (the better such a tool is, the less manual work needs to be done by an end-user)
- Buffer-insertion, gate-sizing, pin-swapping, gate-cloning, logic-transformations
- (paper)

WOSET 2020

PyVSC: SystemVerilog-Style Constraints, and Coverage in Python

- Constrained-random verification and coverage analysis using Python - these are some of the features you'd normally lose when using Python for verification as opposed to SystemVerilog
- [\(paper\)](#)

WOSET 2020

OpenFPGA: Towards Automated Prototyping for Versatile FPGAs

- Framework for automated design-space exploration and prototyping of FPGAs with different architectures
- Important for research into FPGA architectures, security, etc.
- (paper)*

WOSET 2020

Edalize it. Don't criticise it

- Generalized framework for building projects that work with different EDA tools
- Designed to make it easy to switch between tools (i.e. switching from Quartus to Yosys) by translating to each tool's language and commands
- (paper)*

WOSET 2021

CFU Playground: Build your own ML Processor using Open Source

- Framework for experimenting with acceleration of complex algorithms on an FPGA
- Provides a CPU and easy interface / toolchain for connecting a custom design and evaluating its performance against a software implementation
- (paper)

WOSET 2021

CFU Playground: Build your own ML Processor using Open Source

- Framework for experimenting with acceleration of complex algorithms on an FPGA
- Provides a CPU and easy interface / toolchain for connecting a custom design and evaluating its performance against a software implementation
- (paper)

WOSET 2021

FABulous: an Open-Everything Framework for Embedded FPGAs

- Framework for building small, flexible embedded FPGAs into larger designs
 - Special features such as portability across process nodes, low power reconfiguration, and dynamic partial reconfiguration
- Works with open-source toolchains for programming the generated FPGAs
- (paper)

WOSET 2021

OpenCache: An Open-Source OpenRAM Based Cache Generator

- OpenRAM is a framework for generating SRAM automatically
- OpenCache is a framework which accounts for OpenRAM's strengths and weaknesses and uses it to generate an optimal cache controller and generate the relevant RAM blocks
- (paper)

WOSET 2022

Accelerate Silicon Design with Jupyter Notebooks

- The ML world has come up with Jupyter and Conda as a great way for teaching and prototyping complex software processes
- Extends this to hardware design - uses devops-style tools to provision environments and notebooks to build interactive experiments and tutorials
- (paper)

What next?

- If you found this stuff interesting and want to learn more:
 - Buy or borrow a cheap FPGA board
 - Build small personal projects (great chance to try new languages and tools, and build up a GitHub portfolio)
 - Interact with the community (IRC channels, Discord servers, Gitter chats, GitHub issues/PRs)
 - Submit to future Open-MPW tapeouts