DUTCTL: A Flexible Open-Source Framework for Rapid Bring-Up, Characterization, and Remote Operation of Custom-Silicon RISC-V SoCs

Thomas Benz[1], Paul Scheffler[1], Jennifer Holborn[1], Luca Benini[1,2]
[1]Integrated Systems Laboratory, ETH Zurich
[2]Department of Electrical, Electronic, and Information Engineering, University of Bologna

open source

1 Motivation

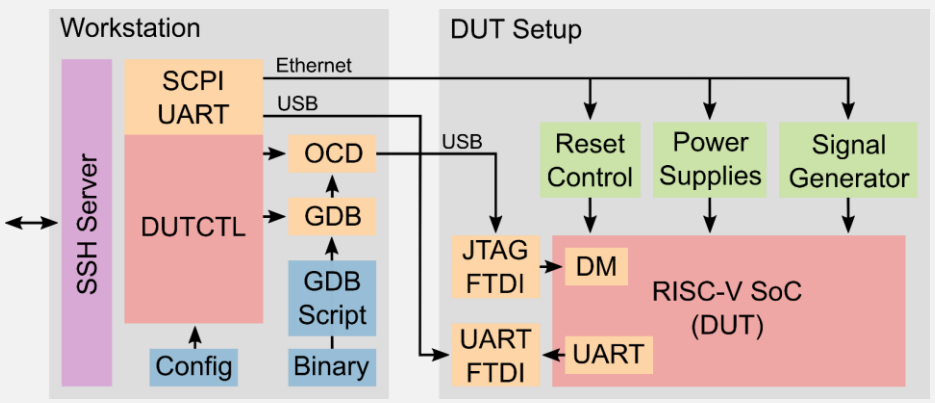
- Industrial silicon testing and characterization uses **automated test equipment (ATE)**[1]
  - **Efficient** for large runs, but **expensive**

- **Prototype SoC bring-up** does not need ATEs
  - **High SW controllability** through RISC-V debug module[2]
  - **Standard IOs**: use COTS adapters and lab equipment
- However, ATEs provide **coordination and automation**
  - **Coordinated** clocking, reset, power supplies, IO
  - **Automated** test flows and parameter sweeps

➢ ***DUTCTL* automates rapid and ATE-less bringup, characterization, and remote operation of RISC-V SoCs**

2 DUTCTL Architecture

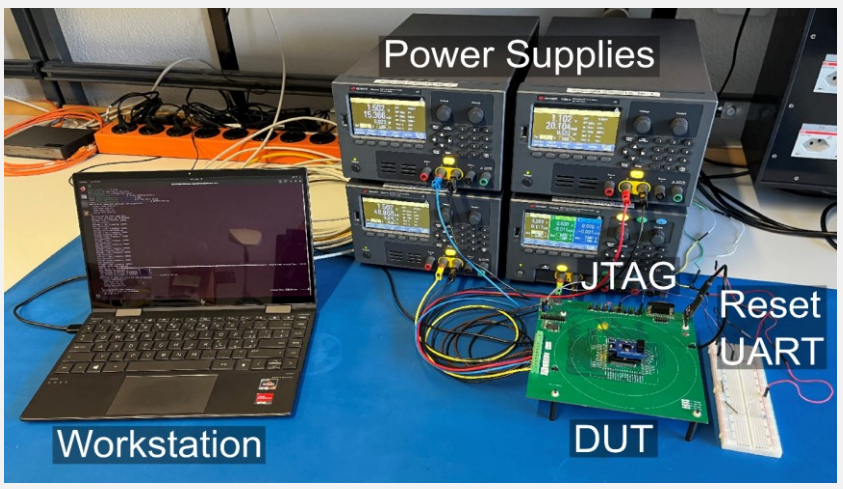
- A **DUTCTL setup** comprises
  - **DUTCTL software** running on (remote) workstation
  - **IO adapters** (JTAG debugger, UART) over USB
  - **Lab instruments** (supplies, signal gens) over Ethernet (**extensible Python library** issuing SCPI[3] commands)



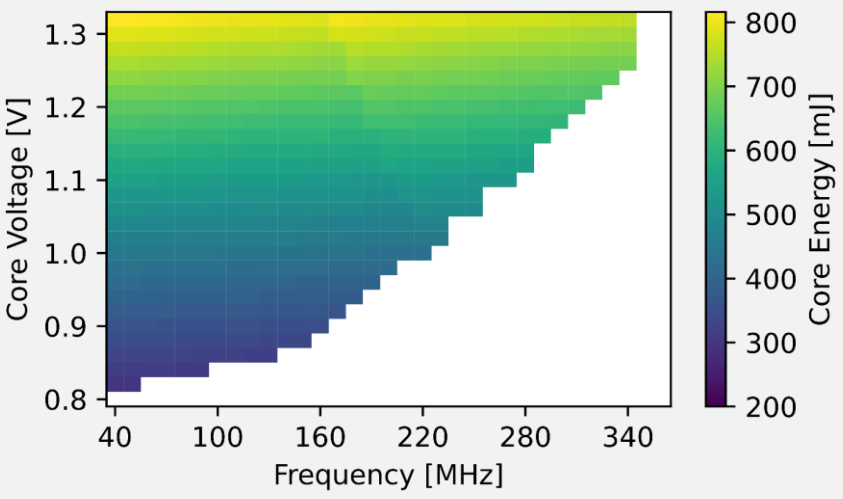
- A **DUTCTL session** automates
  1. **Instrument configuration** (YAML file)
  2. **DUT reset** over instrument GPIOs
  3. **Launching a scripted GDB[4] session**
- During session, the **DUT can control DUTCTL** over UART
  - Record internal, request supply **measurements**
  - **Control instruments** for **parametric tests**
- Sequences of sessions enable **sweeps** and full **test flows**

3 Example: Characterizing an RV64 SoC

- We use DUTCTL to automate the **bring-up** and **characterization o** an open-source 64-bit RISC-V SoC:



- We set up a 2D (frequency, voltage) sweep, sequencing **832 sessions** to create an **enhanced Schmoo[5] plot**:



4 Conclusion

- DUTCTL is an **open-source framework** for **ATE-less bringup and characterization of RISC-V SoCs**
- It is **modular**, **extensible**, and **configurable**
- It can be used **remotely**, **controlled from the DUT**, and **sequenced** for **sweeps** and full **test flows**

source code

References

1. M. Bushnell and V. Agrawal, “Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits”.
2. RISC-V International, “RISC-V External Debug Support Version 0.13.2”.
3. National Instruments, “Documentation About the SCPI Specification”.
4. Free Software Foundation, “GDB: The GNU Project Debugger”, https://www.sourceware.org/gdb/.
5. C. P. Womack, “Schmoo Plot Analysis of Coincident-Current Memory Systems” in *EC-14* (1965).