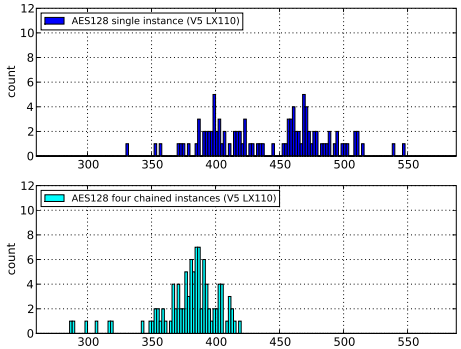
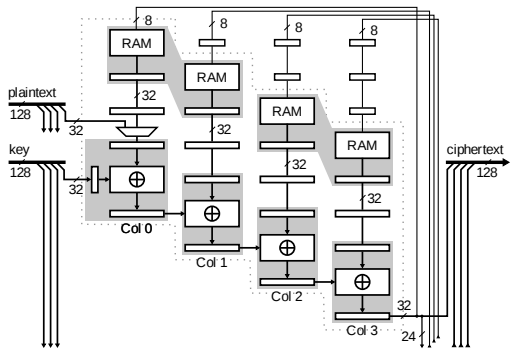
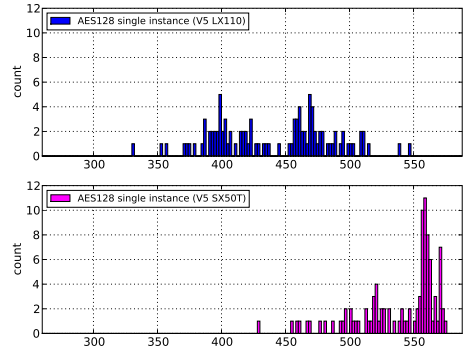
Reproducing and benchmarking FPGA designs

Saar Drimer
http://www.cl.cam.ac.uk/~sd410







UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Is it possible to create a standard for hardware benchmarking?

Very hard. Why?

- Technology: there are too many variables (architecture, software, implementation options)
- Academic: no emphasis on reproducibility; methodologies are hard to maintain; and general acceptance of meaningless comparisons and incomplete implementation data

Here's what I think is *the best we can do*.

Comparing/benchmarking FPGA designs considered harmful

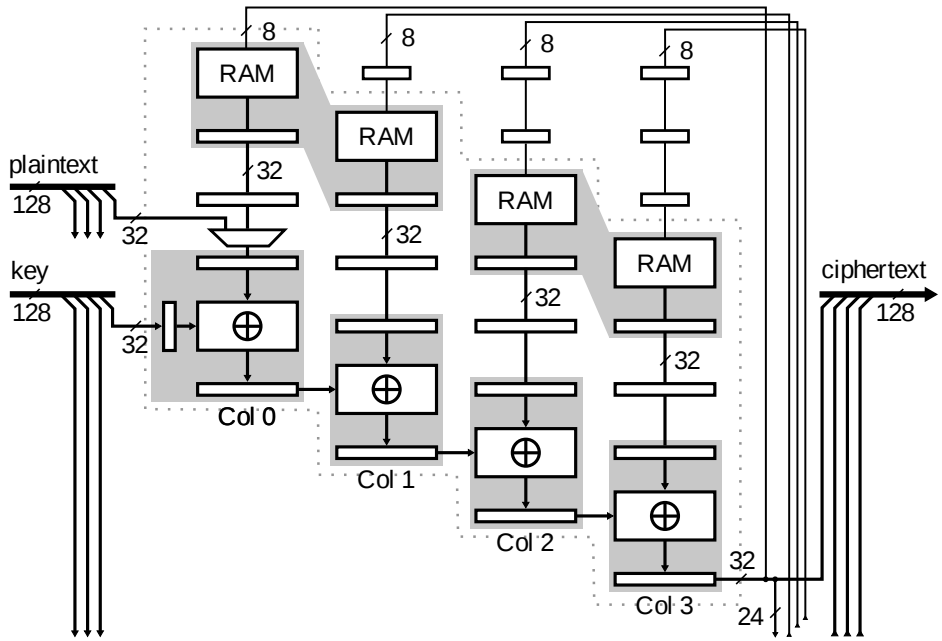
We often see...

- inappropriate use of metrics ("throughput/slice");
- implementations for different architectures compared;
- missing implementation and platform details;
- meaningless resource conversions (BRAM to logic);
- comparison to ASIC implementations; and
- comparison to industry offerings.

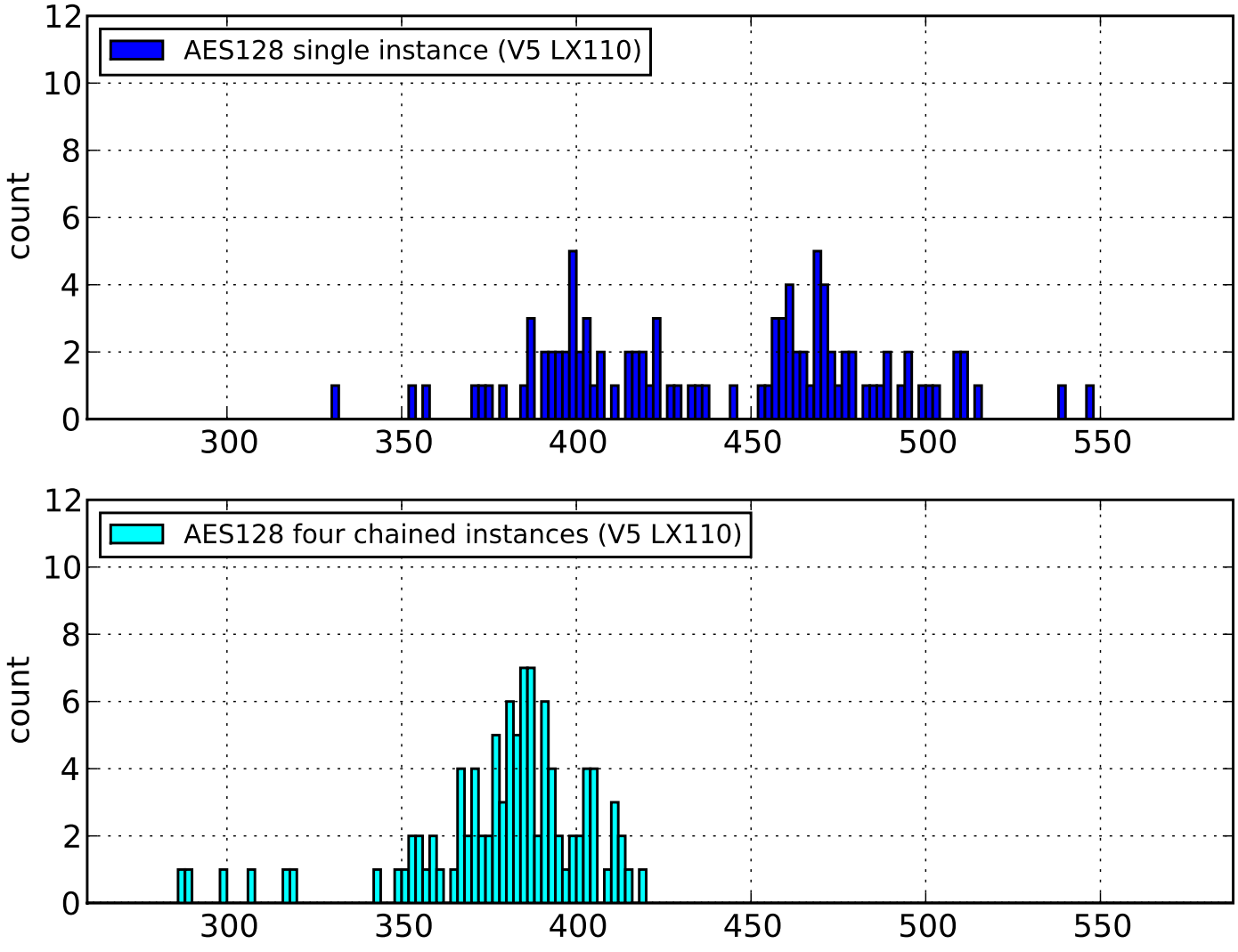
We almost never see... source code.

Should we get excited over a 10% performance improvement?

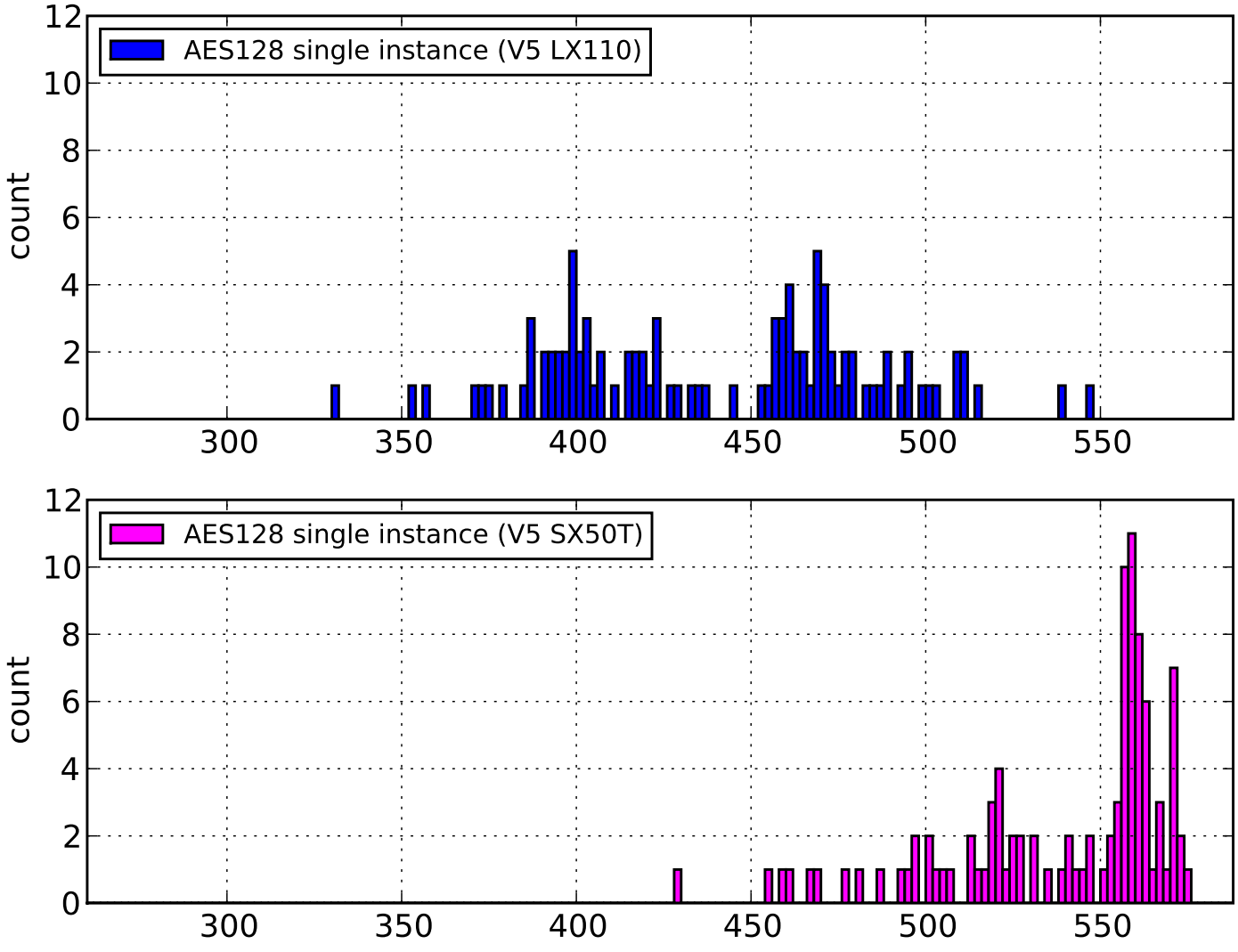
Rarely is context given for modules, as the are implemented unconstrained, leading to the most favorable results; these may be hard to achieve in practice. **Example case, an AES implementation**:



Variability can be huge, over 30% of the entire range; constrained designs perform noticeably worse



Implementation on different devices from the same family; which result is “correct” to report?



My main point is. . .

Reproducible research

leads to

better benchmarking

I avoid "fair" because it is subjective; we know that not many comparison methodologies are accepted by everyone. We need a methodology that is accepted as *reasonable* by most – so it is not only used by its creators – and that is maintainable.

Let's encourage reproducible results by grading reproducibility during manuscript reviews

5. **Fully reproducible:** complete source code, simulation testbenches, and compilation instructions are available with submission.
4. **Fully reproducible later**: as above, but authors commit to having material available at publication time.
3. **Limited reproducibility**: (partial) source code is available but requires significant effort to reproduce reported results.
2. **Reproducible by redesign**: description of the design is complete and simple enough to reproduce without source code.
1. **Unreproducible plus**: description lacks sufficient information for reproducing results, but implementation report files are available. **Or**, some implementation conditions are missing.
0. **Unreproducible**: description lacks sufficient information for reproducing results given any amount of effort.

Encouraging reproducible research will...

- (roughly) compensate for low originality scores;
- allow independent reproduction of results for verification;
- allow researchers to better deflect criticism of their designs;
- allow finding mistakes and omissions;
- promote sound coding and code maintenance practices; and
- **allow reimplementation so comparisons make sense.**

Reproducible results can have a greater contribution to the field than unreproducible, yet more original, results

* Of course this evaluation criteria should only be used where it makes sense (such as with all implementation papers).

Establishing an academic-industry committee to fix some variables seems unavoidable – community effort

The long-term committee will define the following:

- benchmarking platforms from multiple vendors (FPGA type, speed grade, size, etc.);
- benchmarking software tools and settings (PAR seeds, efforts, optimization goals);
- concise information on how to compare designs well, as a guide to implementers and manuscript reviewers;
- target goals (throughput, power, area, etc.) to avoid unconstrained "best achievable case"; and
- a simple, architecture-agnostic HDL wrapper for modules under test (this is the only code to be made available by the committee. . . the rest is information).

Two comments on the committee

Since no benchmarking methodology "fits" all designs, adherence to the committee guidelines should be optional. The availability of source code gives implementers a way to re-implement and compare designs in a meaningful way outside the defined methodology.

The committee will have permanent roles – EDA vendor and FPGA vendor representatives, and researchers – but temporary members, so methodologies are kept up-to-date with technology and new hardware generations.

* A long document describing the content of this presentation, together with experimental results will (hopefully) be available in a short while.