1 Expected Results and Discussion

The following section presents the results that will be included in the next revision of the paper. Convert any of the `.tbl` files to latex tables using the following command:

```
python3 latex_gen.py PAPER/<table>.tbl
```

This command will translate the given `.tbl` file to a latex table. Please refer to the tables and figures in the paper. We discuss how to compare the obtained results to the expected results for each table in the section below.

1.1 Benchmark and Hardware Summary (`bmarks.tbl`, `hwblocks.tbl`, `hwboard.tbl`)

These tables summarize the benchmark characteristics. Each line can be independently confirmed by investigated the `.py` file in the `prog` directory and inspecting the `dsinfo`, `dssim`, and `dsprog` functions. We do not expect any significant deviations.

1.2 LGraph Configuration Results (`circuit-lgraph.tbl`)

Claim: The compiler is able to make use of all the blocks on the device, and does not use more crossbar blocks than necessary (with the exception of `bont`).

Discussion: This table reports a breakdown of the connections and blocks used in each graph. The locally obtained counts should match the counts reported in the paper. Notably, the `bont` benchmark should be the only benchmark that uses 5 crossbar `xbar` blocks.

1.3 LScale Configuration Results (`circuit-lscale.tbl`)

Claim: The scaled configurations are complex and contain multiple distinct scaling factors. For some benchmarks, injected variables are used to scale the system.

Discussion: This table reports the scaling transform statistics from the `lscale` compilation process. The `mdpe`, `mape` and `time constant` may be differ because there is non-determinism in the configuration generation process, but should remain roughly the same. The other columns should remain the same, though it's acceptable if there are slight differences as the other statistics mainly demonstrate that the scaling transform has many distinct values. Generating this table from the dataset (`oscilloscope_data.zip`) should produce exactly the same results.

1.4 Compile Times (`compile-time.tbl`)

Claim: Our compiler is fast. Performance does not degrade significantly with increasing benchmark size.

Discussion: This table reports the time required to compile each benchmark. The results in this table should be comparable, but probably not exactly the same, as the run times reported in the paper (Table XXX). The reported runtimes were obtained by compiling the benchmarks on a Macbook Pro late-2013 laptop.

Q: why is the compilation process longer than the sum of the reported times?

A: One reason is the execution script generates many candidate configurations for the same benchmark – it is therefore executing the compilation process many times. The scaling process also takes longer because the compiler is directed to search for the best parameter setting. This procedure performs a binary search over parameters, where each step in the binary search executes the scaling procedure.

1.5 Quality, Energy and Runtime (`quality-energy-runtime.tbl`)

Claim: We are able to execute simulations using relatively little time and energy and with acceptable quality (2-5% error).

Discussion: This table reports the quantitative waveform quality, energy and runtime results. The energy and runtime results are computed from the configuration and the quality result is computed from the oscilloscope waveform. Note that the previously discussed runtime results are validated by the waveform analysis procedure – if the runtime wasn't accurate, the collected waveform wouldn't align to the reference waveform. The `quality-energy-runtime.tbl` table should match the results reported in Figure XXX. We note that it isn't possible to generate this table without running the experiments on the analog device.

Conversely, the `energy-runtime.tbl` can be computed from just the configuration and can therefore be generated without the oscilloscope waveforms. The average energy and runtime numbers reported in this table should be comparable to the numbers reported in the `quality-energy-runtime.tbl` table and the variance should be relatively low.

1.6 Quality Results (`delta-max-fit-*.pdf`)

Claim: For each benchmark, we are able to produce configurations that yield visually indistinguishable or visually close simulations.

Discussion: These figures are visual comparisons between the collected analog waveform and the reference waveform. Dynamical system outputs are often visually interpreted, or interpreted in accordance to some domain specific criteria (e.g. it is able to track and input, or eliminate a perturbation). These simulations therefore do not need to exactly match the reference simulation.

Q: why are there variations in quality across configurations?

A: The variations in quality occur due to a couple of reasons. First, the configuration may have been generated without manufacturing variation compensation (`ne` and `ng` calibrated block tags), or the configuration may be using blocks calibrated with an error-minimizing objective function (`de`). Second, the block instances used by the configuration may have some unwanted behavior that we have been unable to characterize – this is out of scope of this paper. Third, some of the block instances are usable but have higher error. Handling this is out of scope for this paper as we do not not attempt to rank configurations based on block characteristics or prioritize lower error blocks.

1.7 Without Manufacturing Compensation (`naive-max-fit-*.pdf`)

Claim: Manufacturing variation compensation is necessary to produce good results.

Discussion: Compare the `naive-max-fit-*.pdf`files to the `delta-max-fit-*.pdf`files. The former almost always produces poorer quality results than the latter. This is because the `delta-max-fit-*.pdf`configurations compensate for manufacturing variations across individual blocks. The `naive-max-fit-*.pdf`configurations make the assumption all blocks instances adhere to the ADS block specification.

1.8 With Limited Parametric Behavior (`delta-max-fit-*.pdf`)

Claim: Parametric behavior is necessary to produce good results.

Discussion: Compare the `extended` files to the `standard` files. The former almost always produces better quality results than the latter. This is because the `standard` configurations are restricted to a very small subset of parametric modes for each block (only modes that negate outputs) while the `extended` configurations are allowed to use a much larger range of modes. If each block was restricted to use only one mode, we would not be able to synthesize any configurations.