1 Hypothesis

Increasing the number of reservation stations will increase IPC because increasing the instruction-level parallelism (ILP) can reduce data hazards and control hazard stalls by queuing up more instructions, and therefore improves performance. The effects on power consumption as the number of reservation stations increase may be application dependent, as useless speculations can raise average power consumption if execution time isn't sufficiently lowered [3] (3.9).

2 Experiment Plan

Benchmark	Reservation Station Entries
splash2-ocean.cont	2
splash2-radix	4
splash2-barnes	8
npb-is	16
npb-ua	32
	48
	64
	98
	128

Table 1: Configuration parameters and values swept in the experiment [5, 1, 2].

The experiment will sweep the above number of reservation stations across the benchmarks listed in Table 1. The `commit_width` configuration value will be at a constant 4. Based on previous experiments, the time estimate to do the 45 simulations is about 6 hours. Since the experiment focuses on a particular ILP and not thread-level parallelism, it will run with 1 core, but may also try 2 and 4 cores if there is time to see any effects on the parallel benchmarks. No performance changes may be seen in the parallel benchmarks, unlike the single threaded benchmarks for similar reasons. A processor that aggressively exploits ILP has up to 64 issues and dispatchs per clock ([4], [3] (3.13)), so a few values beyond 64 reservation stations are included to see if 64 is where this peaked.

Cache associativty and block size do not affect my hypothesis with regards to exploiting the ILP; however, there may be some application dependent benefits (namely, memory access heavy benchmarks) to increasing both in L1 or L2, based on previous experiements sweeping cache sizes and associativity. Likewise, prefetching is not expected to affect the hypothesis because it is already focusing on increasing the number of instructions in flight at once.

References

[1] Bailey, D., Barszcz, E., Barton, J., Browning, D., Carter, R., Dagum, L., Fatoohi, R., Frederickson, P., Lasinski, T., Schreiber, R., Simon, H.,

Venkatakrishnan, V., and Weeratunga, S. The nas parallel benchmarks. *Int. J. High Perform. Comput. Appl. 5*, 3 (sep 1991), 63–73.

[2] Feng, H., Vanderwijngaart, R. F., Biswas, R., and Mavriplis, C. Unstructured adaptive (ua) nas parallel benchmark. version 1.0.

[3] Hennessy, J. L., and Patterson, D. A. *Computer Architecture, Sixth Edition: A Quantitative Approach*, 6th ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2017.

[4] Wall, D. W. Limits of instruction-level parallelism. In *Proceedings of the fourth international conference on Architectural support for programming languages and operating systems* (1991), pp. 176–188.

[5] Woo, S., Ohara, M., Torrie, E., Singh, J., and Gupta, A. The splash-2 programs: characterization and methodological considerations. In *Proceedings 22nd Annual International Symposium on Computer Architecture* (1995), pp. 24–36.