Contents

References

[1] The Sniper Multi-Core Simulator

[2] O. Tange (2011): GNU Parallel - The Command-Line Power Tool

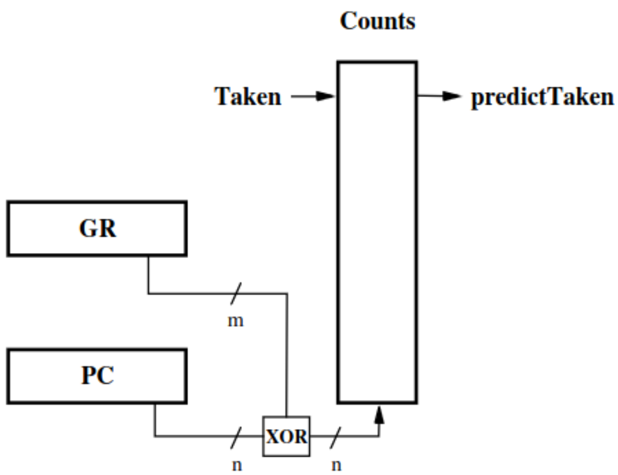
[3] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh and A. Gupta, The SPLASH-2 Programs: Characterization and Methodological Considerations, Proceedings 22nd Annual International Symposium on Computer Architecture, Santa Margherita Ligure, Italy, 1995, pp. 24-36

[4] John L. Hennessy and David A. Patterson. 2017. Computer Architecture, Sixth Edition: A Quantitative Approach. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

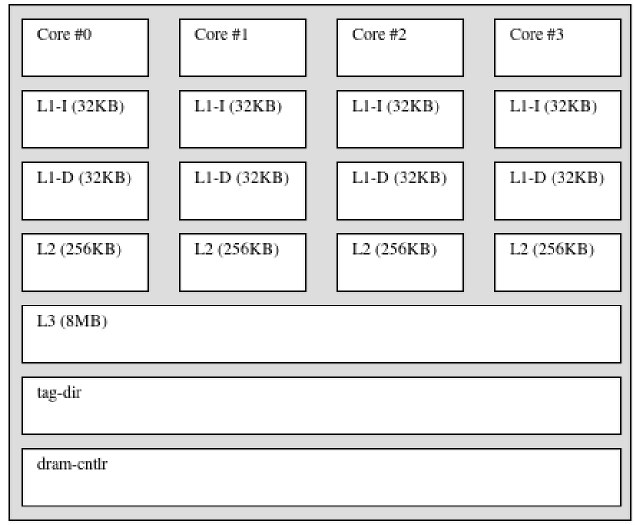
[5] McFarling, Scott, 1993. Combining Branch Predictors, Technical Note TN-36, Western Research Laboratory. Digital Equipment Corp., Palo Alto, CA.

Benchmark	**PHT index bits (gshare)**
`cholesky`	0 (default branch predictor)
`fmm`	4
`lu.cont`	8
`radiosity`	16
`raytrace`	

Table 1: Configuration parameters and values for PHT index bits for the `gshare` branch predictor swept in the experiment. The "default" branch predictor is `pentium`.



(a) `gshare` predictor taken from *McFarling* [5]



(b) Topology for all five benchmark tests where only the number of PHT index bits was varied (all cache sizes remained consistent through every simulation). All benchmarks were run in `Sniper-7.3` with the `gainestown` configuration using the `--viz` and `--roi` options.

Figure 1

1 Intro

Control hazards due to branch misprediction can cause significant performance loss. Techniques that address branching and these penalties include always predicting taken/not-taken, flushing the pipeline, static branch prediction, and dynamic branch prediction. Simple dynamic branch predictors typically keep a branch history table (BHT) that "tracks" previous branch behaviors (taken/not-take) and use that information to inform predictions of the same branch [4].

Correlating branch predictors also take into account branch outcomes of *other* branches to inform the branch being predicted, i.e. a global predictor scheme. In general, such a (m, n)-predictor uses the last m branches (a global history) to choose from 2^m n-bit branch predictors. One such predictor is `gshare` (see Figure 1a); it uses a pattern history table (PHT), a separate BHT for each global pattern. In `gshare`, the index into the PHT is formed by XOR-ing the branch address and recent branch history [4, 5].

This experiment examines the performance of `gshare` as size of the PHT increases, comparing against a default predictor in `Sniper`, which is a `pentium M` style, global branch predictor [1]. Performance and energy consumption improves with increasing PHT size, minimizing the calculated energy-delay product (EDP), which informs how a feature impacts the performance/energy efficient relationship, for the given range of sizes swept.

2 Experimental Setup

Simulations ran for an x86 architecture simulator, Sniper 7.3 [1]. Since this experiment looked to sweep the `gshare` branch predictor performance through PHT sizes, each simulation is configured with the same topology (Figure 1b). The same default configurations were set in `gainestown.cfg`.

Three PHT sizes were swept for simulations using the `gshare` branch predictor, $2^4, 2^8, 2^{16}$. The sweep for "size" 2^0 represents simulations that ran the default branch predictor, `pentium`. `gshare` was implemented separately according to technical specifications [5] in C++, and integrated into `Sniper-7.3` with `gcc-7.4.0`. There was a total of 26 simulations (see Table 1). The different branch predictor configurations were simulated with five `splash2` benchmarks not previously used: `cholesky`, `lu.cont`, `radiosity`, `raytrace`, and `fmm` [3].

The workloads are briefly described as follows:

cholesky : The `cholesky` factors a sparse matrix into the product of a lower triangular matrix and its transpose without globally synchronizing between steps.

fmm : The `fmm` application implements the adaptive Fast Multipole Method to simulate interactions of systems of N-bodies (particles, galaxies, etc.) in 2D with unstructured communication patterns.

lu.cont : The `lu.cont` factors a dense matrix into the product of a lower and upper triangular matrix, exploiting temporal locality on submatrix elements. Blocks are allocated sequentially and locally to processors that own them in order to improve the spatial locality.

radiosity : The `radiosity` application computes the equilibrium distribution of light in a scene using the iterative hierarchical diffuse radiosity method using the light transport interactions and subdivisions in polygons. This application has highly irregular computation structure and data structure accesses.

raytrace : The `raytrace` renders a 3D scene using ray tracing through each pixel in the image plane, reflecting them off objects in unpredictable and multiple ways; therefore, data structure access patterns are also unpredictable.

All the simulations ran concurrently using bash script(s) and GNU `parallel` shell tool [2], and post-processing of the data were handled with python (v2.7) and bash scripts (included separately). Simulations ran on a python virtual environment and in a detached `tmux` session, due to long duration of the experiments. Sniper provided data processing tools used were: `gen_topology.py`, `cpi-stack.py`, and `mcpat.py`.

3 Results & Analysis

3.1 Performance Analysis

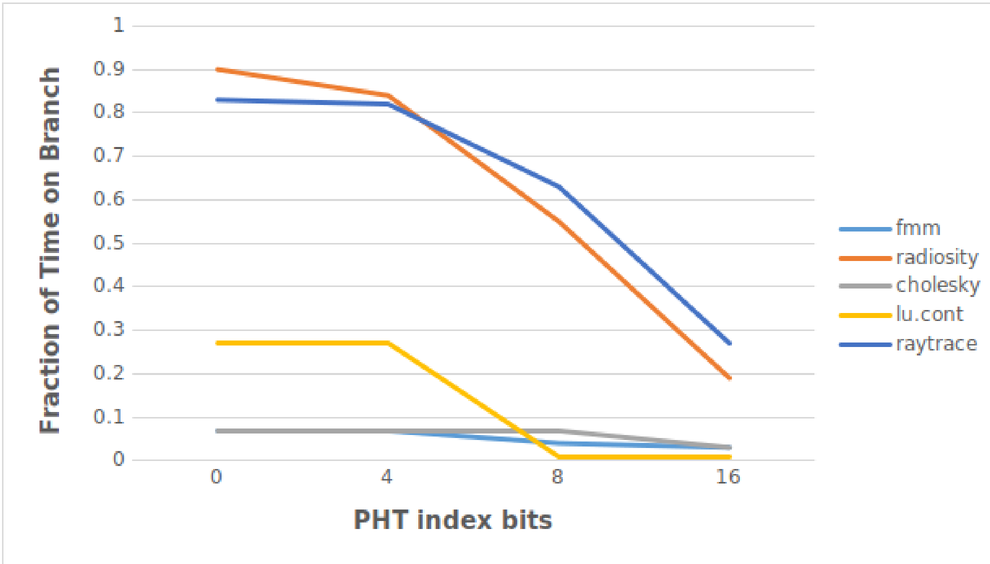


Figure 2: CPI Stack Result - Fraction of time spent on branches. As time decreases, less time is spent handling branches, including mispredictions. (See Figures 9, 13, 17, 21, and 25 for correlating CPI stacks.)

For most workloads, as size of the PHT table increased, time spent on branch instructions noticeably decreases (Figure 2). `fmm` and `cholesky` have less noticeable reductions because those implementations introduced less control hazards. `radiosity` and `raytrace` see more noticeable reduction due to their irregular and unpredictable access patterns more affected by branches; while `lu.cont` has less control hazards to begin with, it even sees about 96% less time spent on branches at the highest PHT size swept compared to the default. The reduced fraction of time spent on branches indicates less time incurring cost of misprediction penalties, likely as branching information became more refined with more branch histories to inform branch decisions.

Reducing time on branching correlates with performance gains seen through increased IPC (Figure 3) and reduced execution time (Figure 4). With less branch misses, more correctly speculated instructions are put on the execution path, leading to more instruction executed and retired in a cycle. Without time spent recovering from control hazards, it also makes sense that the applications also run faster. These suggest that misprediction penalties contribute to a significant portion of performance loss across the five workloads, but are positively impacted by `gshare` with more global history.

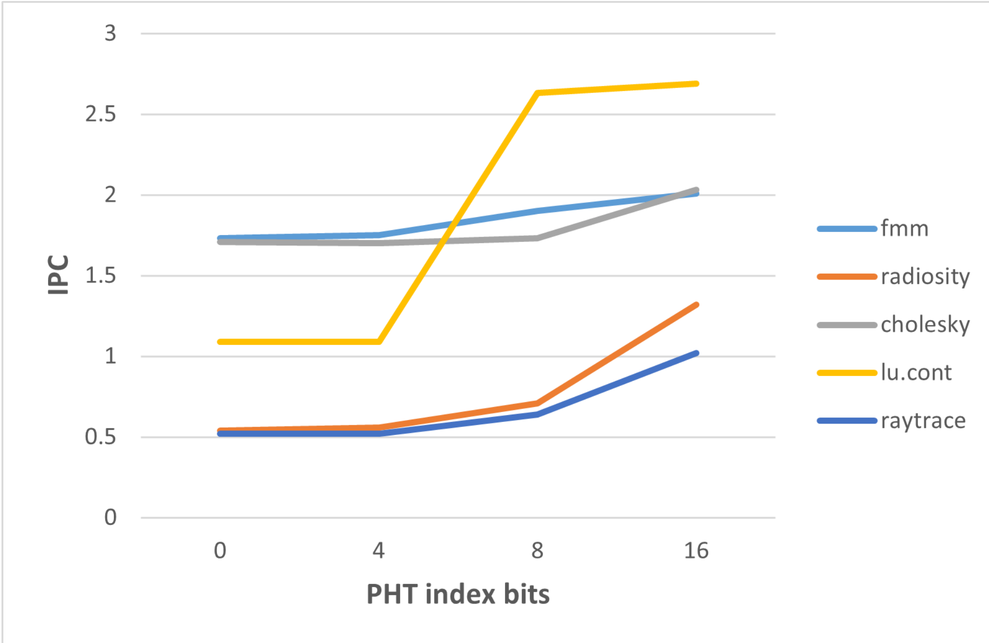


Figure 3: IPC across PHT size sweep for five benchmars. IPC increases for four sweeps and plateaus around 16 PHT index bits for `lu.cont`. (*See attached all.csv file for specific data values.*)

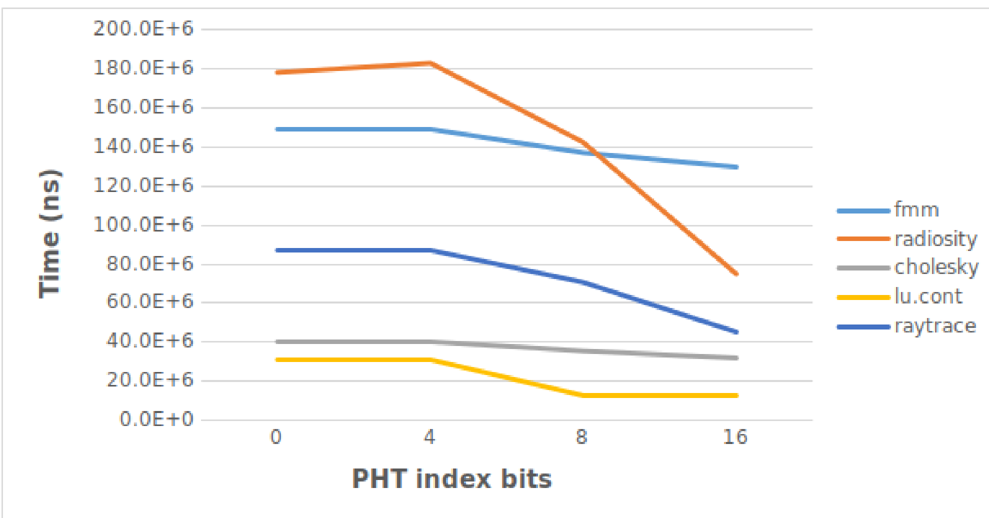


Figure 4: Execution time across PHT size sweep for five benchmarks.

3.2 Energy Consumption

Peak dynamic processor power increases for the sweep across all benchmarks (Figure 5). This is likely due not only to the complexity of the individual applications, but also the implementation of the integrated `gshare`, as well as the total increasing bits in a PHT. However, overall power consumption typically decreases with increasing PHT size across the five workloads (Figures 7, 11, 15, 19, and 23). Along with faster execution time across all sweeps (Figure 4), this gives a decreasing trend of EDP with the sweep, as evident by the results plotted in Figure 6. The lower EDPs show a positive relationship between the performance and energy efficiency tradeoffs due to refining branch histories in `gshare`. Once again, this indicates the significant impact of branch penalties on a system.

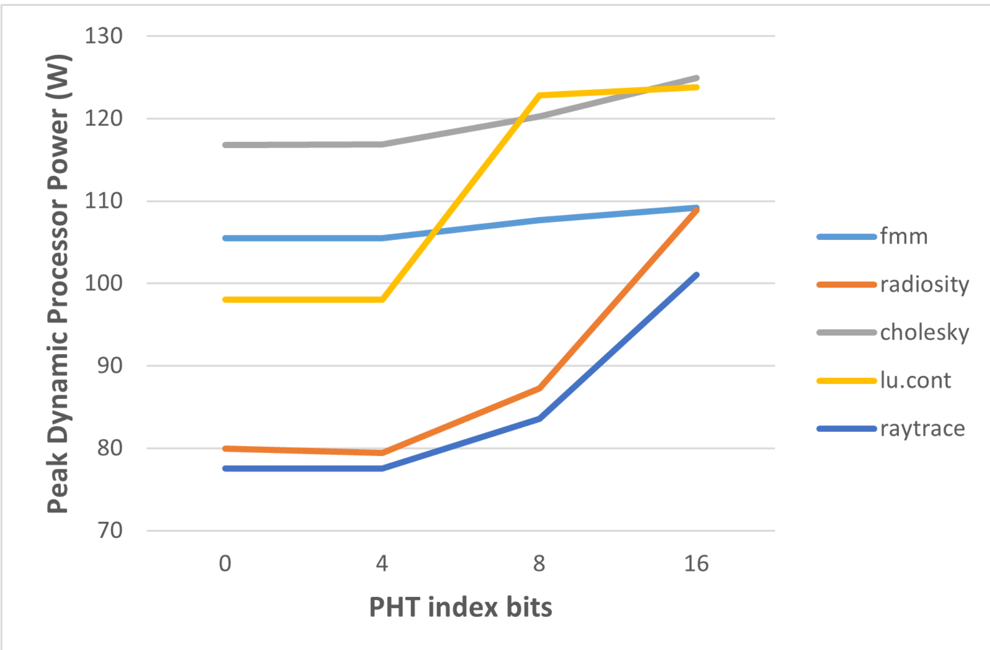


Figure 5: Peak dynamic processor power across PHT size sweep for five benchmarks. Peak dynamic power continues to increase for `radiosity` and `raytrace` at the max for this work, it only slowly increases for `cholesky` and `fmm`, and plateaus for `lu.cont` around 8 to 16 PHT index bits.

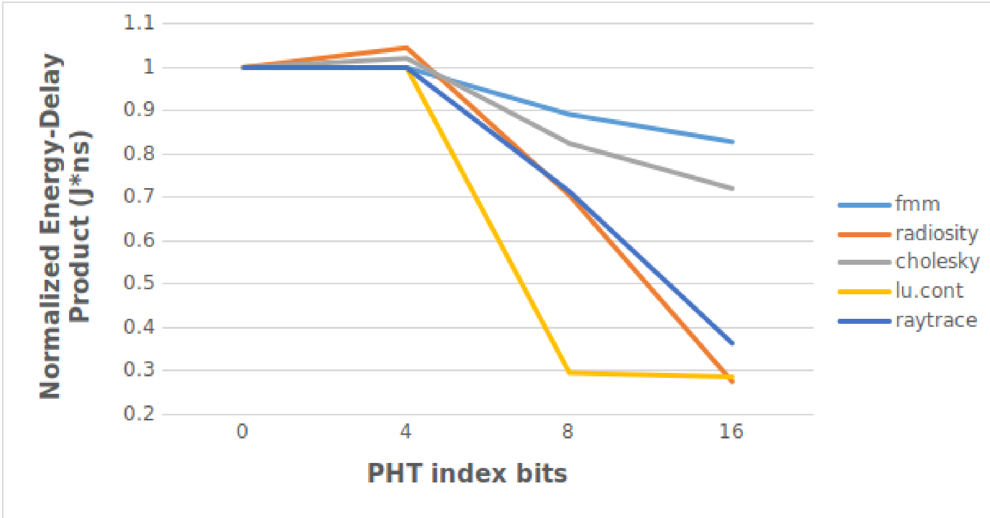


Figure 6: Normalized energy-delay-product (EDP) across PHT size sweep for five benchmarks. EDP is given by the energy consumption (in Joules) times the execution time of the application (in nanoseconds). (See Figures 7, 11, 15, 19, and 23 for correlating Power Results.)

4 Conclusion

The integrated `gshare` predictor performed better than the default, `pentium`; with increasing PHT sizes, `gshare` reduces time spent on branching, which increases overal IPC and execution time. Not only that, but incurring less misprediction penalty leads to improved energy consumption, with an overall lowering of EDP. This shows the importance of branch predictor choice and design in overall performance. These results also suggest several obvious direction to expand this work. Only three sizes are swept for `gshare`, which shows a trend for improved performance and energy consumption; more sizes should be swept to see if the trend continues or if an optimal range for PHT size exists, after which point performance and/or energy degrades. Along with PHT size, other elements that could be examined are the number of bits in the global history register and the counter bits, as those could also refine `gshare` performance to a point. In addition, other factors of interest include comparing `gshare` against other correlating predictors, dynamic predictors, and static ones because `gshare` is already known to work well and is used as a baseline for other predictors [4], but not necessarily the optimal predictor for all types of workloads.

5 Appendix: Raw Post Processed Data

5.1 cholesky

5.1.1 Power Results

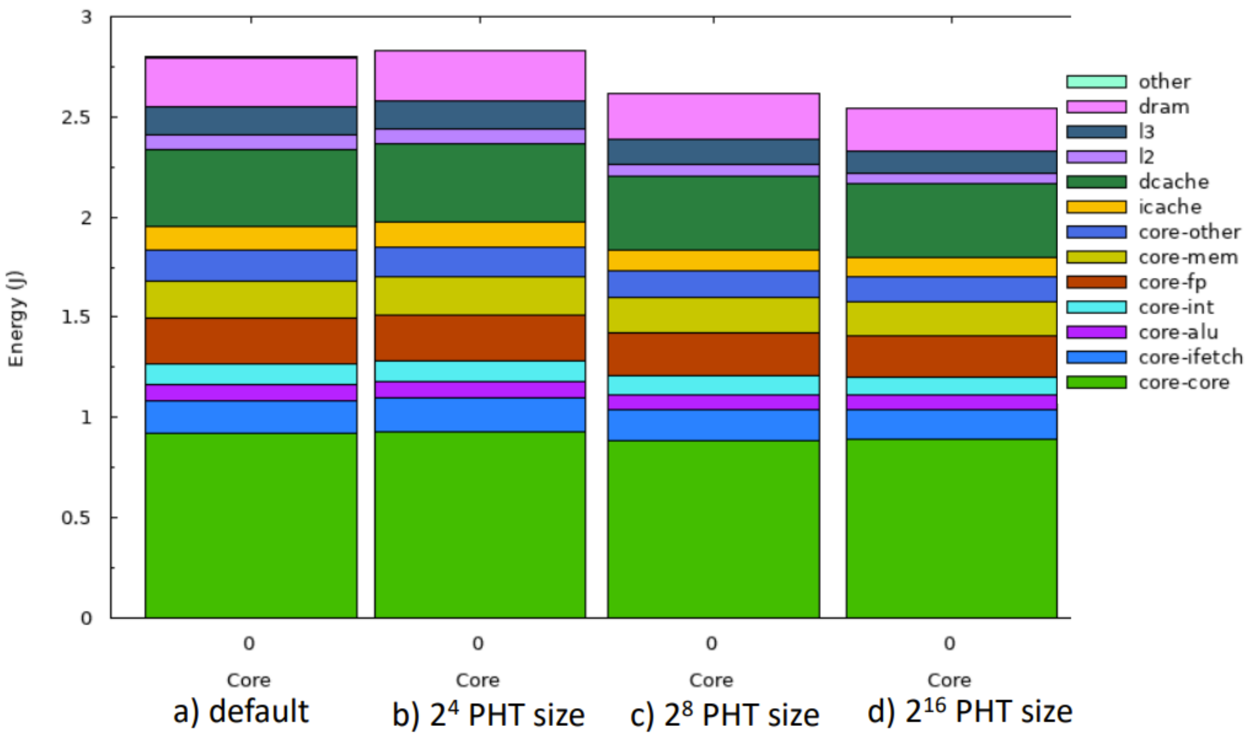


Figure 7: Processor power for various PHT sizes.

	Power	Energy	Energy %
core-core	22.91 W	0.92 J	32.89%
core-ifetch	4.04 W	0.16 J	5.79%
core-alu	2.06 W	0.08 J	2.95%
core-int	2.60 W	0.10 J	3.73%
core-fp	5.68 W	0.23 J	8.15%
core-mem	4.57 W	0.18 J	6.55%
core-other	3.79 W	0.15 J	5.45%
icache	2.90 W	0.12 J	4.16%
dcache	9.71 W	0.39 J	13.93%
l2	1.71 W	0.07 J	2.46%
l3	3.48 W	0.14 J	4.99%
dram	6.19 W	0.25 J	8.89%
other	0.03 W	1.31 mJ	0.05%
core	45.65 W	1.83 J	65.52%
cache	17.79 W	0.71 J	25.54%
total	69.67 W	2.80 J	100.00%

(a) default

	Power	Energy	Energy %
core-core	22.92 W	0.93 J	32.86%
core-ifetch	4.06 W	0.16 J	5.82%
core-alu	2.05 W	0.08 J	2.94%
core-int	2.61 W	0.11 J	3.74%
core-fp	5.65 W	0.23 J	8.10%
core-mem	4.59 W	0.19 J	6.59%
core-other	3.79 W	0.15 J	5.44%
icache	2.96 W	0.12 J	4.24%
dcache	9.75 W	0.40 J	13.97%
l2	1.71 W	0.07 J	2.45%
l3	3.48 W	0.14 J	4.99%
dram	6.15 W	0.25 J	8.82%
other	0.03 W	1.33 mJ	0.05%
core	45.68 W	1.85 J	65.49%
cache	17.89 W	0.73 J	25.65%
total	69.76 W	2.83 J	100.00%

(b) 2^4 PHT size

	Power	Energy	Energy %
core-core	25.07 W	0.89 J	33.92%
core-ifetch	4.26 W	0.15 J	5.76%
core-alu	2.18 W	0.08 J	2.95%
core-int	2.71 W	0.10 J	3.67%
core-fp	6.04 W	0.21 J	8.18%
core-mem	4.89 W	0.17 J	6.61%
core-other	3.79 W	0.13 J	5.13%
icache	2.88 W	0.10 J	3.89%
dcache	10.42 W	0.37 J	14.09%
l2	1.72 W	0.06 J	2.32%
l3	3.49 W	0.12 J	4.72%
dram	6.45 W	0.23 J	8.72%
other	0.03 W	1.16 mJ	0.04%
core	48.94 W	1.73 J	66.20%
cache	18.50 W	0.66 J	25.03%
total	73.93 W	2.62 J	100.00%

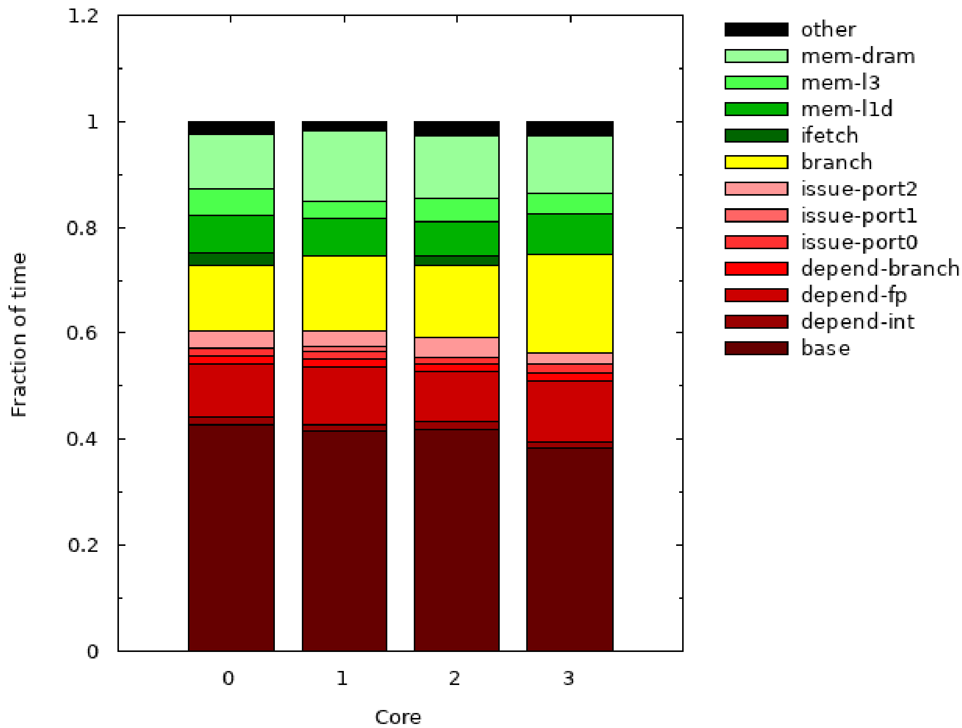
(c) 2^8 PHT size

	Power	Energy	Energy %
core-core	27.97 W	0.89 J	34.91%
core-ifetch	4.66 W	0.15 J	5.81%
core-alu	2.29 W	0.07 J	2.86%
core-int	2.90 W	0.09 J	3.62%
core-fp	6.39 W	0.20 J	7.98%
core-mem	5.44 W	0.17 J	6.79%
core-other	3.79 W	0.12 J	4.74%
icache	3.15 W	0.10 J	3.93%
dcache	11.54 W	0.37 J	14.40%
l2	1.72 W	0.05 J	2.15%
l3	3.51 W	0.11 J	4.38%
dram	6.71 W	0.21 J	8.38%
other	0.03 W	1.04 mJ	0.04%
core	53.45 W	1.70 J	66.72%
cache	19.92 W	0.63 J	24.86%
total	80.11 W	2.55 J	100.00%

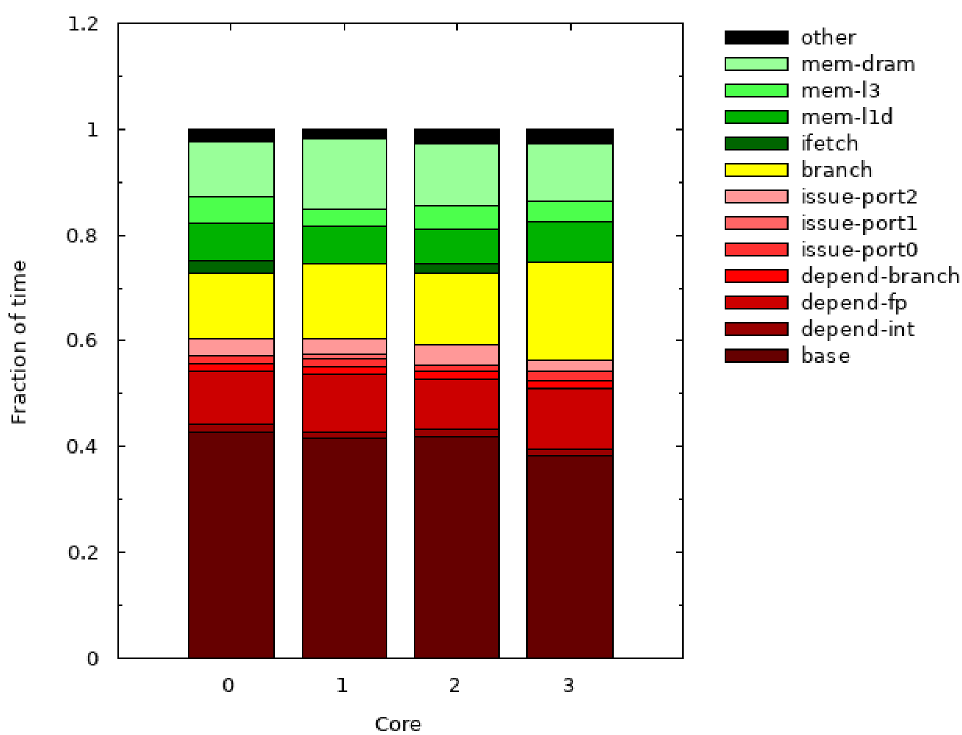
(d) 2^{16} PHT size

Figure 8: Specific values for each components' power consumption.

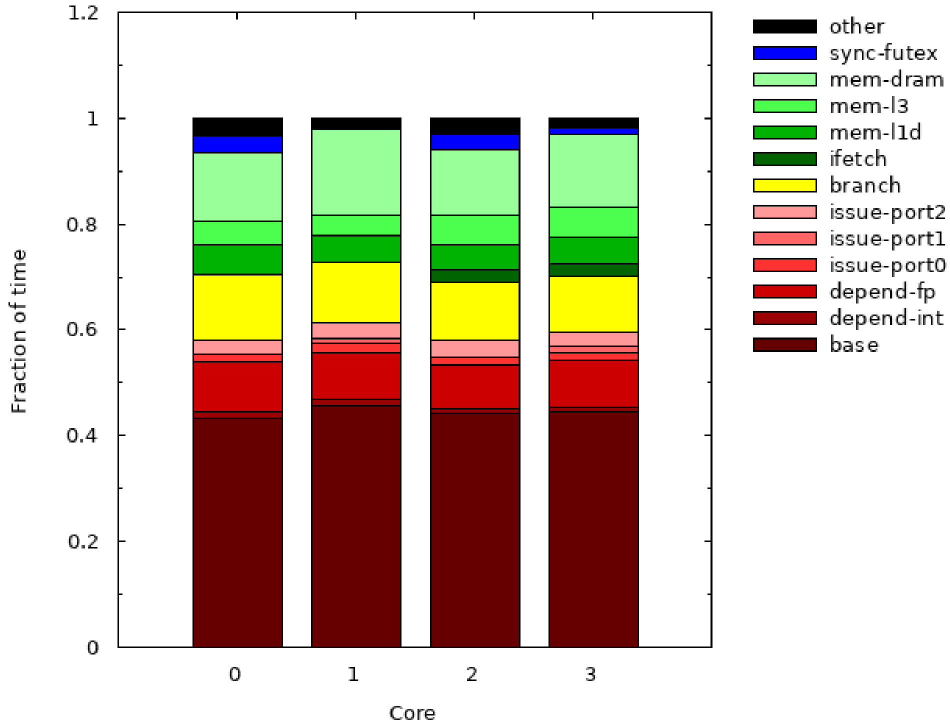
5.1.2 CPI Stacks



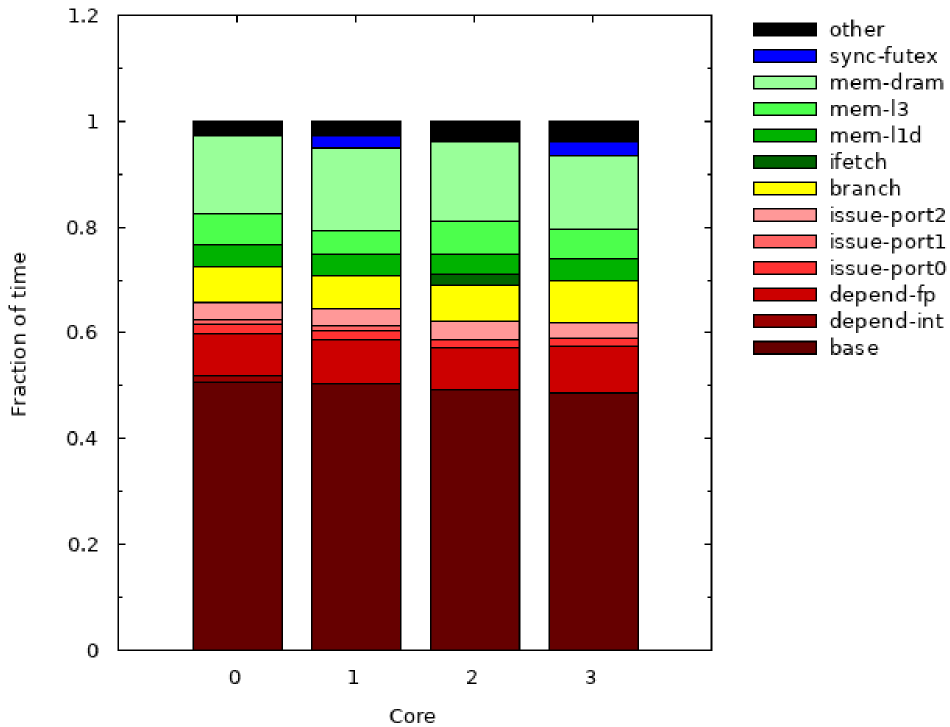
(a) default



(b) 2^4 PHT size



(c) 2^8 PHT size



(d) 2^{16} PHT size

Figure 9: CPI stacks for various PHT sizes.

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.01	0.01	0.01	0.01
depend-fp	0.06	0.06	0.05	0.08
depend-branch	0.01	0.01	0.01	0.01
issue-port0	0.01	0.01	0.01	0.01
issue-port1	0.00	0.01	0.00	0.00
issue-port2	0.02	0.02	0.02	0.01
branch	0.07	0.09	0.08	0.12
ifetch	0.01	0.00	0.01	0.00
mem-l1d	0.04	0.04	0.04	0.05
mem-l3	0.03	0.02	0.03	0.03
mem-dram	0.06	0.08	0.07	0.07
other	0.01	0.01	0.02	0.02
total	0.58	0.60	0.60	0.65

(a) default

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.01	0.01	0.01	0.01
depend-fp	0.06	0.06	0.05	0.08
depend-branch	0.01	0.01	0.01	0.01
issue-port0	0.01	0.01	0.01	0.01
issue-port2	0.02	0.02	0.02	0.01
branch	0.07	0.07	0.08	0.11
ifetch	0.01	0.00	0.00	0.00
mem-l1d	0.04	0.04	0.04	0.05
mem-l3	0.03	0.02	0.03	0.03
mem-dram	0.06	0.07	0.06	0.07
sync-futex	0.01	0.00	0.03	0.02
other	0.01	0.02	0.02	0.02
total	0.59	0.57	0.60	0.66

(b) 2^4 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.01	0.01	0.01	0.01
depend-fp	0.05	0.05	0.05	0.05
issue-port0	0.01	0.01	0.01	0.01
issue-port1	0.00	0.01	0.00	0.01
issue-port2	0.02	0.02	0.02	0.01
branch	0.07	0.06	0.06	0.06
ifetch	0.00	0.00	0.01	0.01
mem-l1d	0.03	0.03	0.03	0.03
mem-l3	0.03	0.02	0.03	0.03
mem-dram	0.08	0.09	0.07	0.08
sync-futex	0.02	0.00	0.02	0.01
other	0.02	0.01	0.02	0.01
total	0.58	0.55	0.57	0.56

(c) 2^8 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.01	0.00	0.00	0.00
depend-fp	0.04	0.04	0.04	0.05
issue-port0	0.01	0.01	0.01	0.01
issue-port1	0.01	0.00	0.00	0.00
issue-port2	0.01	0.01	0.02	0.02
branch	0.03	0.03	0.03	0.04
ifetch	0.00	0.00	0.01	0.00
mem-l1d	0.02	0.02	0.02	0.02
mem-l3	0.03	0.02	0.03	0.03
mem-dram	0.07	0.08	0.08	0.07
sync-futex	0.00	0.01	0.00	0.01
other	0.01	0.01	0.02	0.02
total	0.49	0.50	0.51	0.51

(d) 2^{16} PHT size

Figure 10: Specific values for each components' CPI stack.

5.2 fmm

5.2.1 Power Results

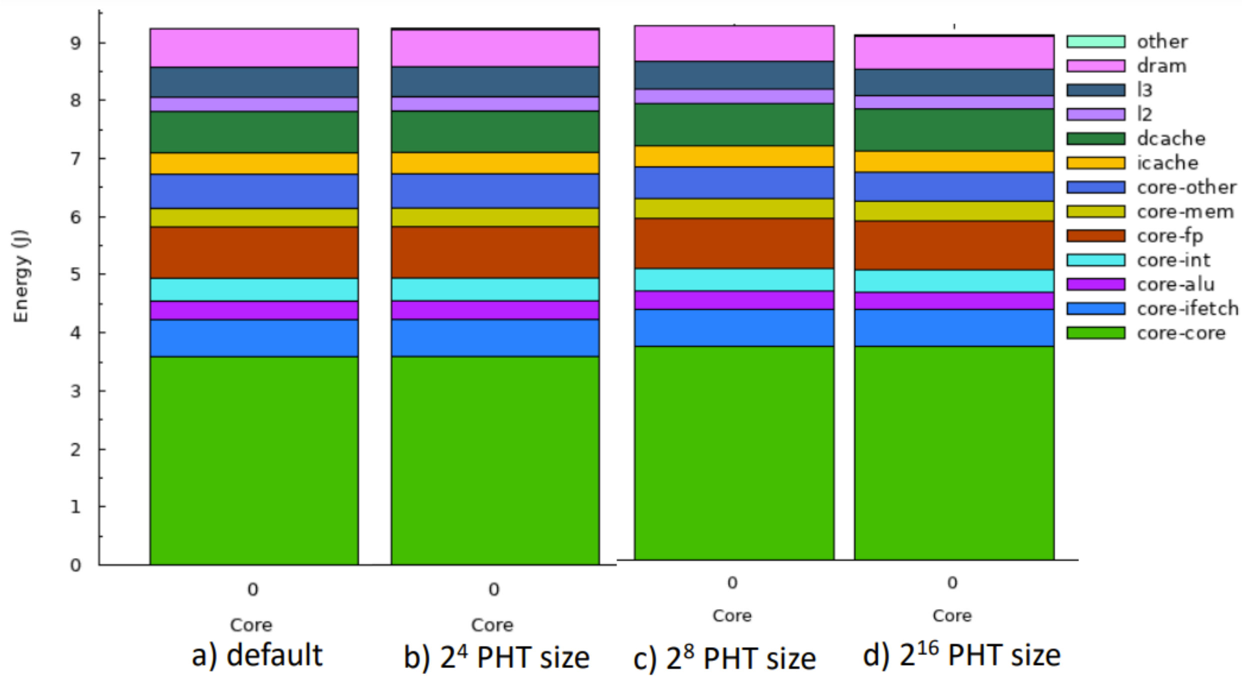


Figure 11: Processor power for various PHT sizes.

	Power	Energy	Energy %
core-core	24.12 W	3.60 J	38.99%
core-ifetch	4.22 W	0.63 J	6.83%
core-alu	2.18 W	0.33 J	3.52%
core-int	2.62 W	0.39 J	4.23%
core-fp	5.84 W	0.87 J	9.44%
core-mem	2.26 W	0.34 J	3.66%
core-other	3.79 W	0.57 J	6.13%
icache	2.54 W	0.38 J	4.11%
dcache	4.78 W	0.71 J	7.73%
l2	1.68 W	0.25 J	2.71%
l3	3.41 W	0.51 J	5.51%
dram	4.38 W	0.65 J	7.08%
other	0.03 W	4.88 mJ	0.05%
core	45.04 W	6.72 J	72.81%
cache	12.41 W	1.85 J	20.06%
total	61.87 W	9.23 J	100.00%

(a) default

	Power	Energy	Energy %
core-core	24.13 W	3.60 J	38.99%
core-ifetch	4.22 W	0.63 J	6.83%
core-alu	2.18 W	0.33 J	3.52%
core-int	2.62 W	0.39 J	4.23%
core-fp	5.84 W	0.87 J	9.44%
core-mem	2.26 W	0.34 J	3.66%
core-other	3.79 W	0.57 J	6.13%
icache	2.54 W	0.38 J	4.11%
dcache	4.78 W	0.71 J	7.73%
l2	1.68 W	0.25 J	2.71%
l3	3.41 W	0.51 J	5.51%
dram	4.38 W	0.65 J	7.08%
other	0.03 W	4.88 mJ	0.05%
core	45.05 W	6.72 J	72.81%
cache	12.41 W	1.85 J	20.06%
total	61.88 W	9.23 J	100.00%

(b) 2^4 PHT size

	Power	Energy	Energy %
core-core	26.20 W	3.59 J	40.08%
core-ifetch	4.51 W	0.62 J	6.90%
core-alu	2.27 W	0.31 J	3.47%
core-int	2.75 W	0.38 J	4.20%
core-fp	6.10 W	0.84 J	9.33%
core-mem	2.43 W	0.33 J	3.73%
core-other	3.79 W	0.52 J	5.81%
icache	2.70 W	0.37 J	4.12%
dcache	5.10 W	0.70 J	7.81%
l2	1.68 W	0.23 J	2.57%
l3	3.41 W	0.47 J	5.22%
dram	4.39 W	0.60 J	6.72%
other	0.03 W	4.48 mJ	0.05%
core	48.05 W	6.58 J	73.51%
cache	12.89 W	1.77 J	19.72%
total	65.36 W	8.96 J	100.00%

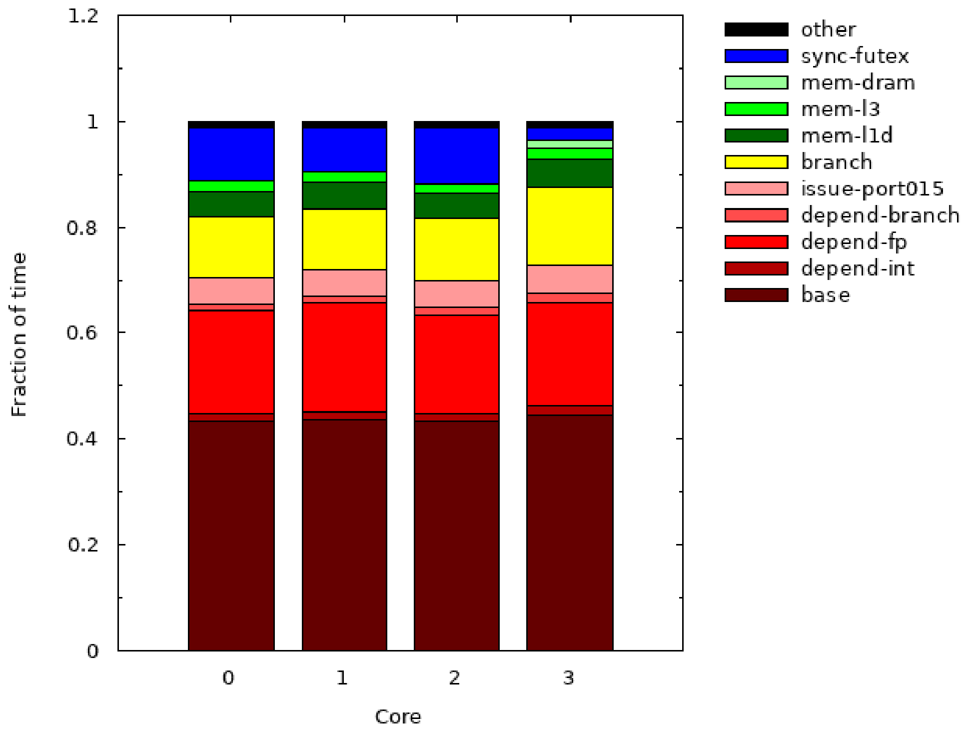
(c) 2^8 PHT size

	Power	Energy	Energy %
core-core	27.61 W	3.58 J	40.76%
core-ifetch	4.70 W	0.61 J	6.94%
core-alu	2.33 W	0.30 J	3.44%
core-int	2.84 W	0.37 J	4.19%
core-fp	6.27 W	0.81 J	9.26%
core-mem	2.55 W	0.33 J	3.76%
core-other	3.79 W	0.49 J	5.60%
icache	2.80 W	0.36 J	4.13%
dcache	5.32 W	0.69 J	7.85%
l2	1.68 W	0.22 J	2.48%
l3	3.41 W	0.44 J	5.04%
dram	4.40 W	0.57 J	6.49%
other	0.03 W	4.24 mJ	0.05%
core	50.09 W	6.50 J	73.96%
cache	13.21 W	1.71 J	19.50%
total	67.74 W	8.79 J	100.00%

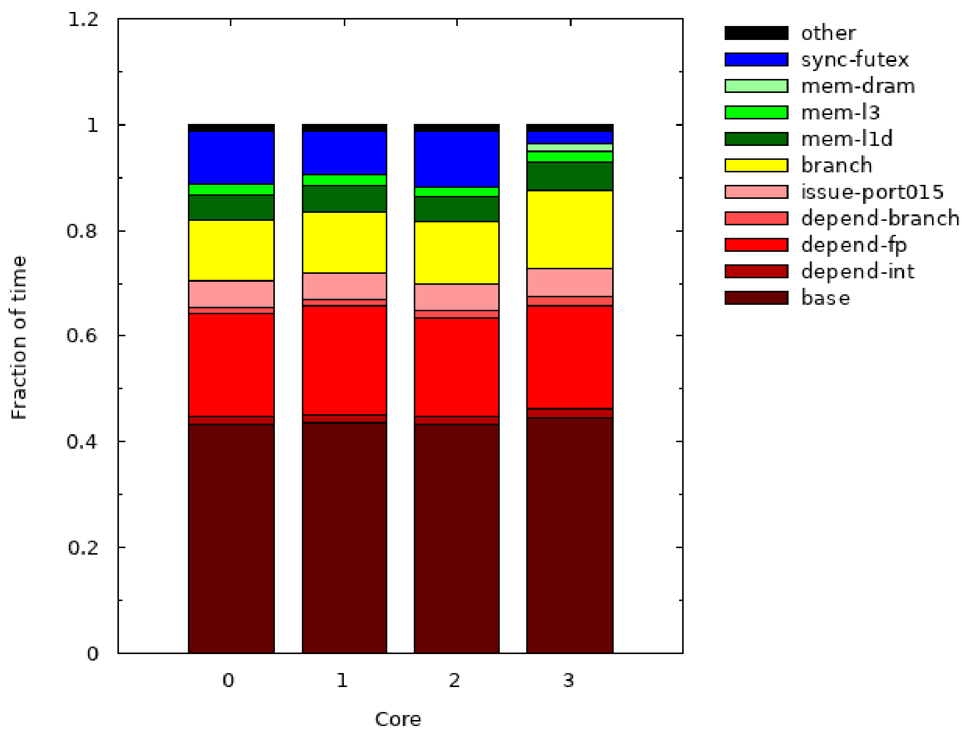
(d) 2^{16} PHT size

Figure 12: Specific values for each components' power consumption.

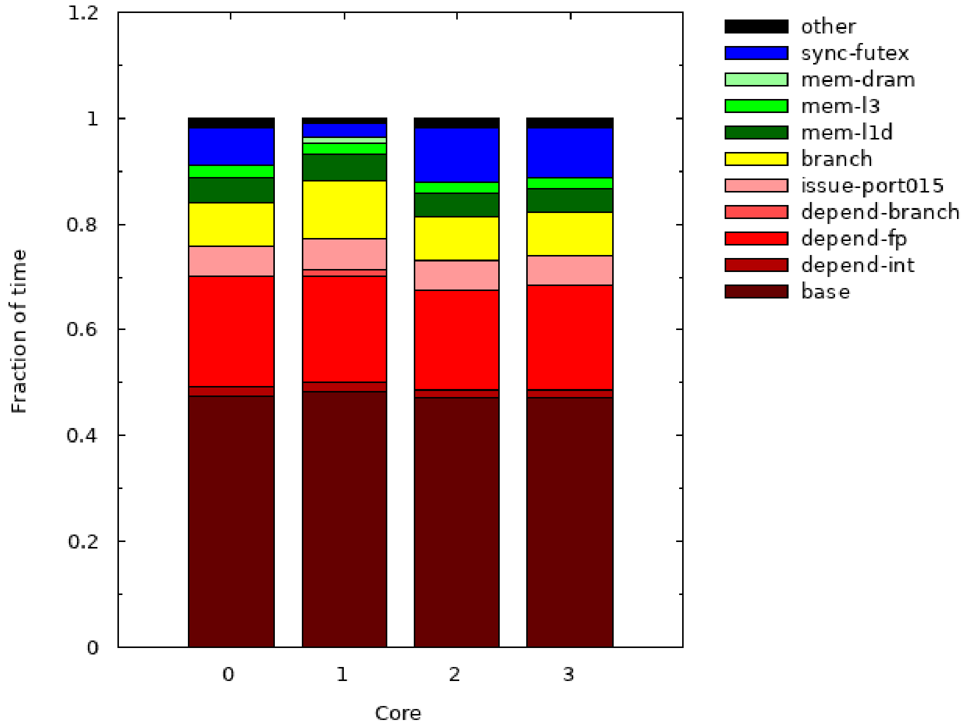
5.2.2 CPI Stacks



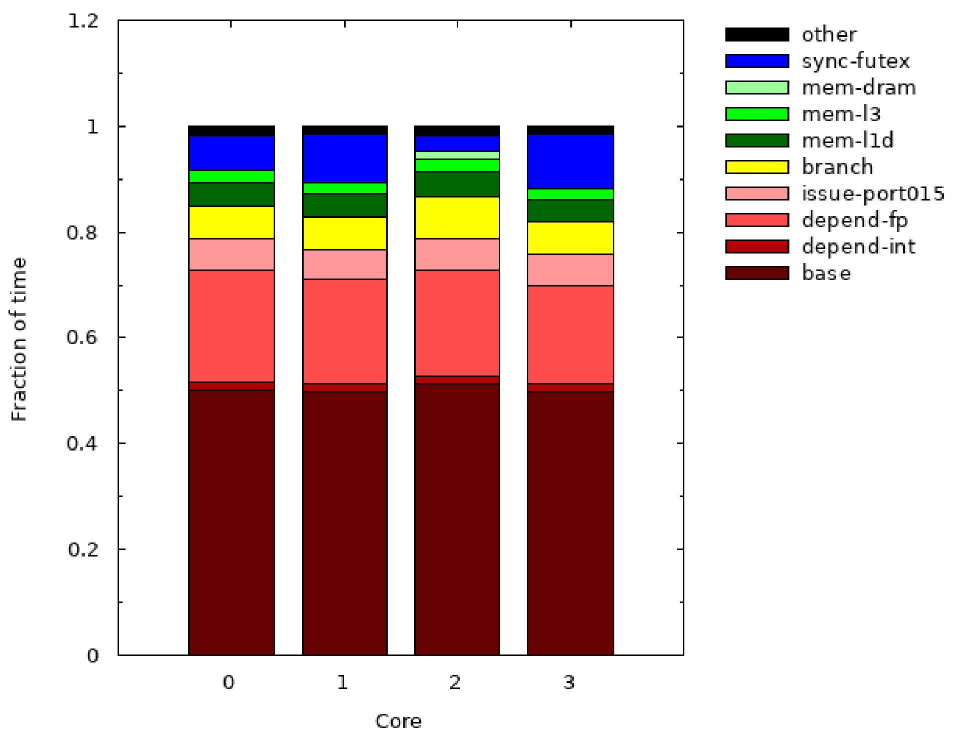
(a) default



(b) 2^4 PHT size



(c) 2^8 PHT size



(d) 2^{16} PHT size

Figure 13: CPI stacks for various PHT sizes.

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.01	0.01	0.01	0.01
depend-fp	0.11	0.12	0.11	0.11
depend-branch	0.01	0.01	0.01	0.01
issue-port015	0.03	0.03	0.03	0.03
branch	0.07	0.07	0.07	0.08
mem-l1d	0.03	0.03	0.03	0.03
mem-l3	0.01	0.01	0.01	0.01
mem-dram	0.00	0.00	0.00	0.01
sync-futex	0.06	0.05	0.06	0.01
other	0.01	0.01	0.01	0.01
total	0.58	0.57	0.58	0.56

(a) default

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.01	0.01	0.01	0.01
depend-fp	0.12	0.11	0.11	0.11
depend-branch	0.01	0.01	0.01	0.01
issue-port015	0.03	0.03	0.03	0.03
branch	0.07	0.08	0.07	0.07
mem-l1d	0.03	0.03	0.03	0.03
mem-l3	0.01	0.01	0.01	0.01
mem-dram	0.00	0.01	0.00	0.00
sync-futex	0.05	0.01	0.06	0.06
other	0.01	0.01	0.01	0.01
total	0.57	0.56	0.58	0.58

(b) 2^4 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.01	0.01	0.01	0.01
depend-fp	0.11	0.10	0.10	0.10
depend-branch	0.00	0.01	0.00	0.00
issue-port015	0.03	0.03	0.03	0.03
branch	0.04	0.06	0.04	0.04
mem-l1d	0.03	0.03	0.02	0.02
mem-l3	0.01	0.01	0.01	0.01
mem-dram	0.00	0.01	0.00	0.00
sync-futex	0.04	0.01	0.06	0.05
other	0.01	0.00	0.01	0.01
total	0.53	0.52	0.53	0.53

(c) 2^8 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.01	0.01	0.01	0.01
depend-fp	0.11	0.10	0.10	0.09
issue-port015	0.03	0.03	0.03	0.03
branch	0.03	0.03	0.04	0.03
mem-l1d	0.02	0.02	0.02	0.02
mem-l3	0.01	0.01	0.01	0.01
mem-dram	0.00	0.00	0.01	0.00
sync-futex	0.03	0.05	0.01	0.05
other	0.01	0.01	0.01	0.01
total	0.50	0.50	0.49	0.50

(d) 2^{16} PHT size

Figure 14: Specific values for each components' CPI stack.

5.3 lu.cont

5.3.1 Power Results

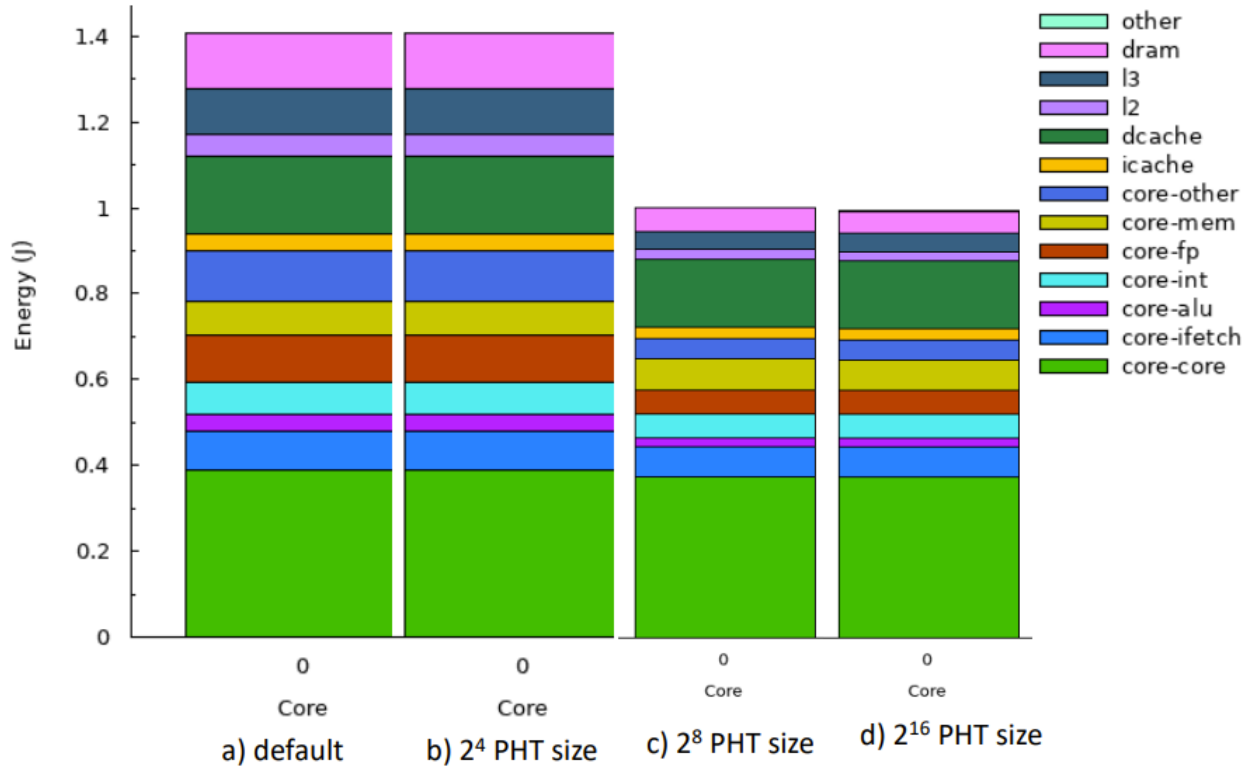


Figure 15: Processor power for various PHT sizes.

	Power	Energy	Energy %
core-core	12.70 W	0.39 J	27.60%
core-ifetch	2.95 W	0.09 J	6.42%
core-alu	1.37 W	0.04 J	2.98%
core-int	2.45 W	0.08 J	5.33%
core-fp	3.59 W	0.11 J	7.79%
core-mem	2.58 W	0.08 J	5.60%
core-other	3.79 W	0.12 J	8.24%
icache	1.34 W	0.04 J	2.92%
dcache	5.88 W	0.18 J	12.78%
l2	1.67 W	0.05 J	3.63%
l3	3.40 W	0.10 J	7.40%
dram	4.25 W	0.13 J	9.24%
other	0.03 W	1.00 mJ	0.07%
core	29.44 W	0.90 J	63.97%
cache	12.30 W	0.38 J	26.72%
total	46.03 W	1.41 J	100.00%

(a) default

	Power	Energy	Energy %
core-core	12.70 W	0.39 J	27.60%
core-ifetch	2.95 W	0.09 J	6.42%
core-alu	1.37 W	0.04 J	2.98%
core-int	2.46 W	0.08 J	5.33%
core-fp	3.59 W	0.11 J	7.79%
core-mem	2.58 W	0.08 J	5.60%
core-other	3.79 W	0.12 J	8.24%
icache	1.34 W	0.04 J	2.92%
dcache	5.88 W	0.18 J	12.78%
l2	1.67 W	0.05 J	3.63%
l3	3.40 W	0.10 J	7.39%
dram	4.25 W	0.13 J	9.24%
other	0.03 W	1.00 mJ	0.07%
core	29.45 W	0.90 J	63.97%
cache	12.30 W	0.38 J	26.72%
total	46.03 W	1.41 J	100.00%

(b) 2^4 PHT size

	Power	Energy	Energy %
core-core	29.83 W	0.38 J	37.39%
core-ifetch	5.74 W	0.07 J	7.20%
core-alu	1.66 W	0.02 J	2.09%
core-int	4.36 W	0.06 J	5.46%
core-fp	4.46 W	0.06 J	5.60%
core-mem	5.80 W	0.07 J	7.27%
core-other	3.79 W	0.05 J	4.76%
icache	2.06 W	0.03 J	2.58%
dcache	12.65 W	0.16 J	15.86%
l2	1.68 W	0.02 J	2.10%
l3	3.45 W	0.04 J	4.32%
dram	4.25 W	0.05 J	5.33%
other	0.03 W	0.41 mJ	0.04%
core	55.65 W	0.70 J	69.76%
cache	19.84 W	0.25 J	24.86%
total	79.77 W	1.01 J	100.00%

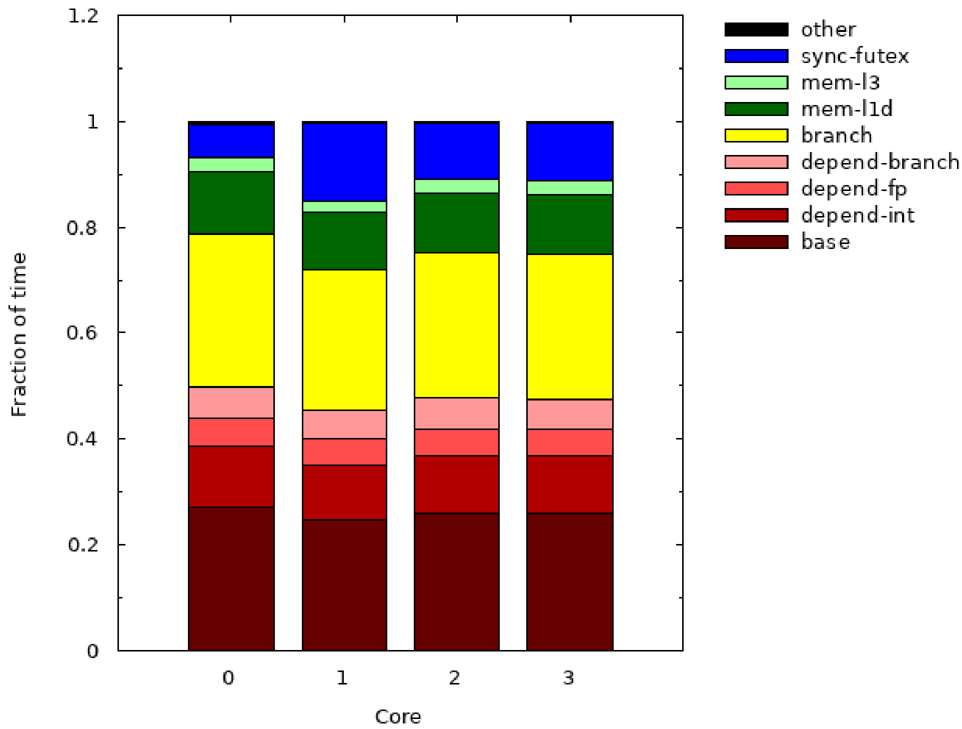
(c) 2^8 PHT size

	Power	Energy	Energy %
core-core	30.48 W	0.38 J	37.60%
core-ifetch	5.85 W	0.07 J	7.22%
core-alu	1.68 W	0.02 J	2.07%
core-int	4.43 W	0.05 J	5.47%
core-fp	4.50 W	0.06 J	5.55%
core-mem	5.92 W	0.07 J	7.31%
core-other	3.79 W	0.05 J	4.68%
icache	2.08 W	0.03 J	2.57%
dcache	12.91 W	0.16 J	15.93%
l2	1.68 W	0.02 J	2.07%
l3	3.45 W	0.04 J	4.26%
dram	4.25 W	0.05 J	5.25%
other	0.03 W	0.40 mJ	0.04%
core	56.65 W	0.70 J	69.89%
cache	20.12 W	0.25 J	24.82%
total	81.06 W	1.00 J	100.00%

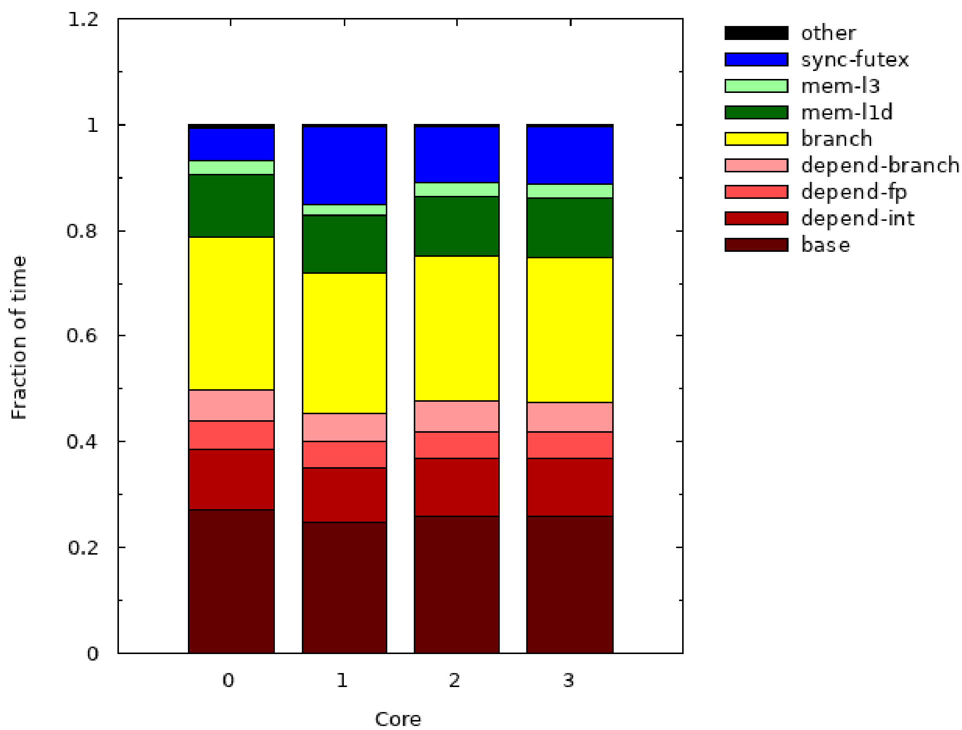
(d) 2^{16} PHT size

Figure 16: Specific values for each components' power consumption.

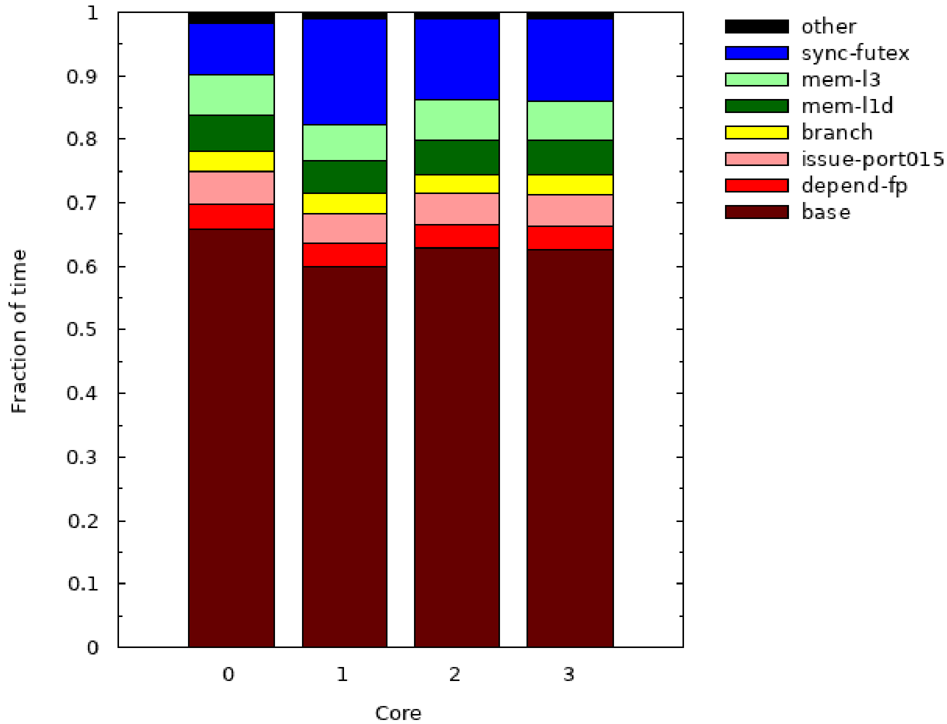
5.3.2 CPI Stacks



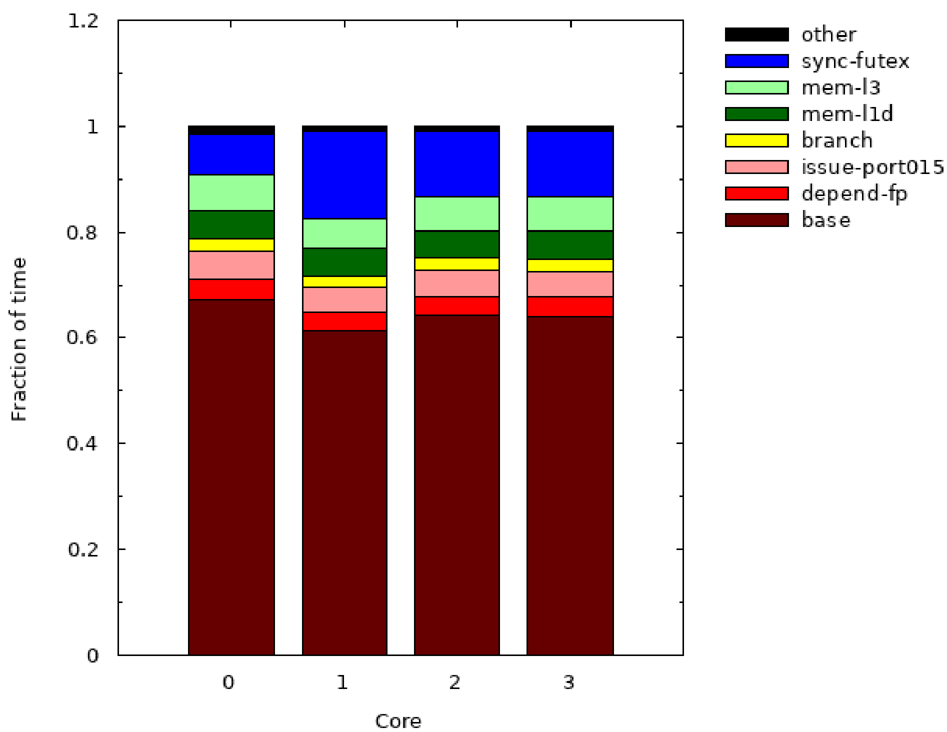
(a) default



(b) 2^4 PHT size



(c) 2^8 PHT size



(d) 2^{16} PHT size

Figure 17: CPI stacks for various PHT sizes.

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.10	0.10	0.10	0.10
depend-fp	0.05	0.05	0.05	0.05
depend-branch	0.05	0.05	0.05	0.05
branch	0.27	0.27	0.27	0.26
mem-l1d	0.11	0.11	0.11	0.11
mem-l3	0.02	0.02	0.03	0.03
sync-futex	0.06	0.15	0.10	0.10
other	0.01	0.00	0.00	0.00
total	0.92	1.01	0.96	0.97

(a) default

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.10	0.10	0.10	0.10
depend-fp	0.05	0.05	0.05	0.05
depend-branch	0.05	0.05	0.05	0.05
branch	0.27	0.27	0.27	0.27
mem-l1d	0.11	0.11	0.11	0.11
mem-l3	0.02	0.02	0.03	0.03
sync-futex	0.06	0.15	0.10	0.10
other	0.01	0.00	0.00	0.00
total	0.92	1.01	0.96	0.96

(b) 2^4 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-fp	0.01	0.02	0.01	0.01
issue-port015	0.02	0.02	0.02	0.02
branch	0.01	0.01	0.01	0.01
mem-l1d	0.02	0.02	0.02	0.02
mem-l3	0.02	0.02	0.02	0.03
sync-futex	0.03	0.07	0.05	0.05
other	0.01	0.00	0.00	0.00
total	0.38	0.42	0.40	0.40

(c) 2^8 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-fp	0.01	0.01	0.01	0.01
issue-port015	0.02	0.02	0.02	0.02
branch	0.01	0.01	0.01	0.01
mem-l1d	0.02	0.02	0.02	0.02
mem-l3	0.02	0.02	0.03	0.02
sync-futex	0.03	0.07	0.05	0.05
other	0.01	0.00	0.00	0.00
total	0.37	0.41	0.39	0.39

(d) 2^{16} PHT size

Figure 18: Specific values for each components' CPI stack.

5.4 radiosity

5.4.1 Power Results

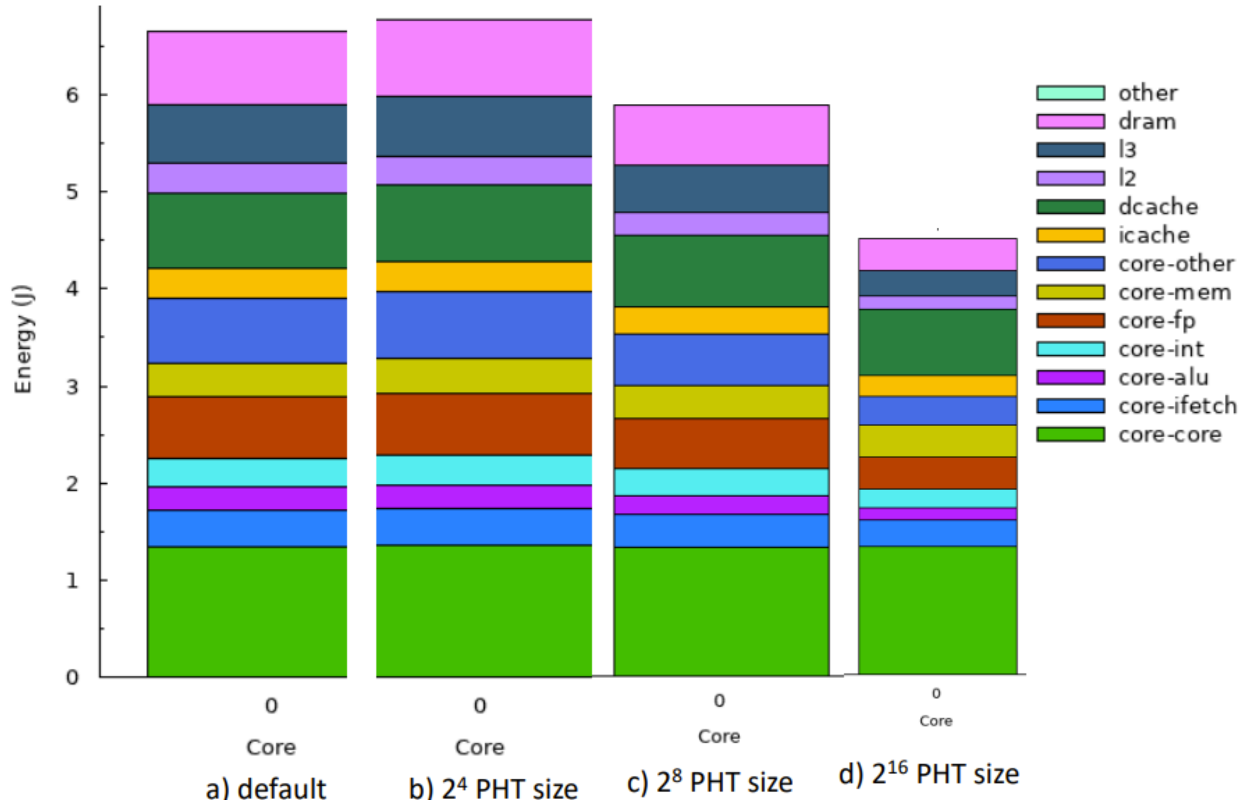


Figure 19: Processor power for various PHT sizes.

	Power	Energy	Energy %
core-core	7.53 W	1.34 J	20.14%
core-ifetch	2.08 W	0.37 J	5.56%
core-alu	1.35 W	0.24 J	3.60%
core-int	1.72 W	0.31 J	4.61%
core-fp	3.53 W	0.63 J	9.44%
core-mem	1.94 W	0.35 J	5.18%
core-other	3.79 W	0.68 J	10.15%
icache	1.70 W	0.30 J	4.56%
dcache	4.36 W	0.78 J	11.67%
l2	1.67 W	0.30 J	4.47%
l3	3.38 W	0.60 J	9.05%
dram	4.29 W	0.76 J	11.48%
other	0.03 W	5.83 mJ	0.09%
core	21.94 W	3.91 J	58.69%
cache	11.12 W	1.98 J	29.74%
total	37.38 W	6.66 J	100.00%

(a) default

	Power	Energy	Energy %
core-core	7.39 W	1.35 J	19.93%
core-ifetch	2.06 W	0.38 J	5.55%
core-alu	1.34 W	0.25 J	3.62%
core-int	1.71 W	0.31 J	4.62%
core-fp	3.52 W	0.64 J	9.49%
core-mem	1.90 W	0.35 J	5.13%
core-other	3.79 W	0.69 J	10.24%
icache	1.68 W	0.31 J	4.54%
dcache	4.29 W	0.79 J	11.57%
l2	1.67 W	0.31 J	4.51%
l3	3.38 W	0.62 J	9.13%
dram	4.29 W	0.79 J	11.58%
other	0.03 W	5.99 mJ	0.09%
core	21.71 W	3.97 J	58.58%
cache	11.03 W	2.02 J	29.75%
total	37.06 W	6.78 J	100.00%

(b) 2^4 PHT size

	Power	Energy	Energy %
core-core	9.32 W	1.33 J	22.57%
core-ifetch	2.36 W	0.34 J	5.72%
core-alu	1.39 W	0.20 J	3.37%
core-int	1.88 W	0.27 J	4.56%
core-fp	3.67 W	0.52 J	8.89%
core-mem	2.36 W	0.34 J	5.71%
core-other	3.79 W	0.54 J	9.19%
icache	1.93 W	0.27 J	4.66%
dcache	5.20 W	0.74 J	12.59%
l2	1.67 W	0.24 J	4.05%
l3	3.39 W	0.48 J	8.20%
dram	4.30 W	0.61 J	10.42%
other	0.03 W	4.66 mJ	0.08%
core	24.78 W	3.53 J	60.01%
cache	12.18 W	1.73 J	29.49%
total	41.30 W	5.88 J	100.00%

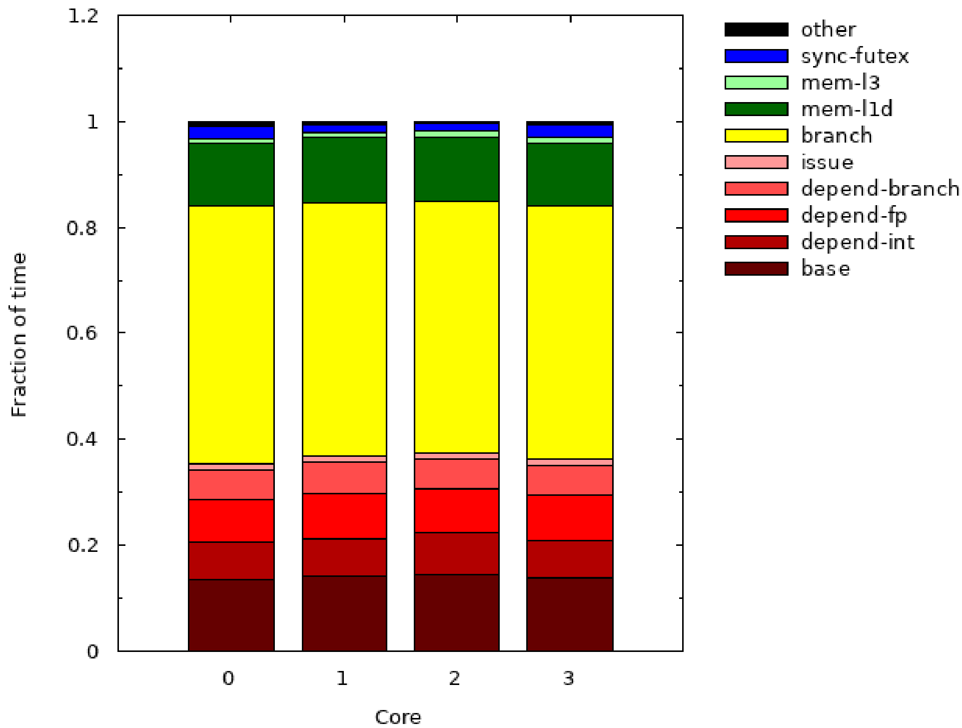
(c) 2^8 PHT size

	Power	Energy	Energy %
core-core	17.07 W	1.28 J	29.27%
core-ifetch	3.59 W	0.27 J	6.15%
core-alu	1.60 W	0.12 J	2.74%
core-int	2.57 W	0.19 J	4.41%
core-fp	4.31 W	0.32 J	7.38%
core-mem	4.20 W	0.31 J	7.19%
core-other	3.79 W	0.28 J	6.51%
icache	2.90 W	0.22 J	4.98%
dcache	8.86 W	0.66 J	15.19%
l2	1.68 W	0.13 J	2.87%
l3	3.40 W	0.25 J	5.82%
dram	4.34 W	0.33 J	7.44%
other	0.03 W	2.45 mJ	0.06%
core	37.12 W	2.78 J	63.64%
cache	16.84 W	1.26 J	28.86%
total	58.33 W	4.37 J	100.00%

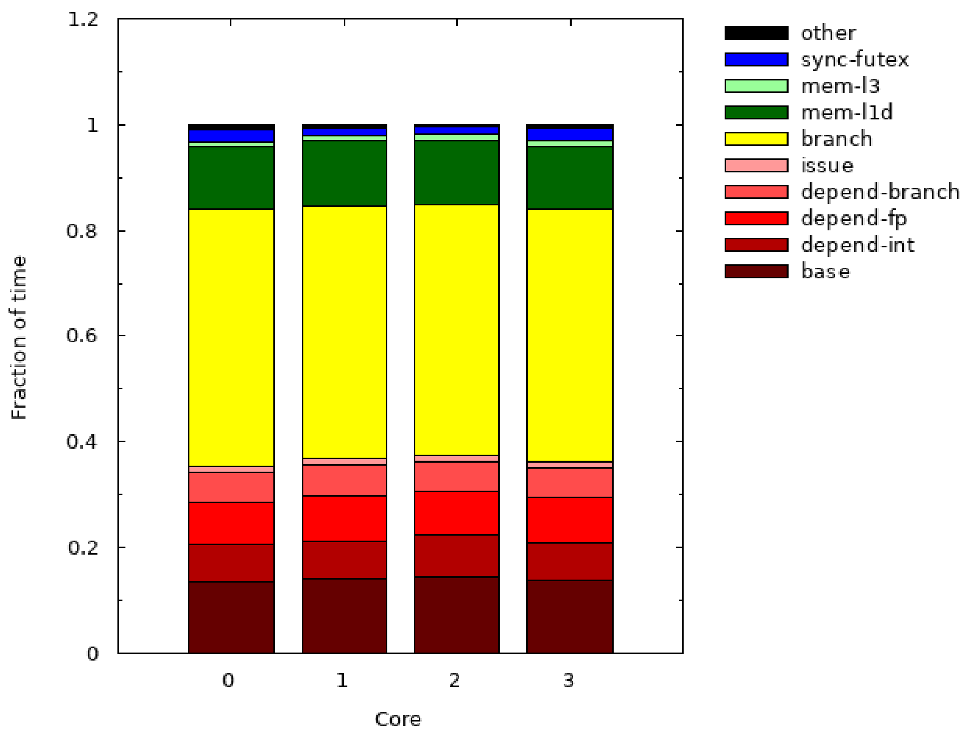
(d) 2^{16} PHT size

Figure 20: Specific values for each components' power consumption.

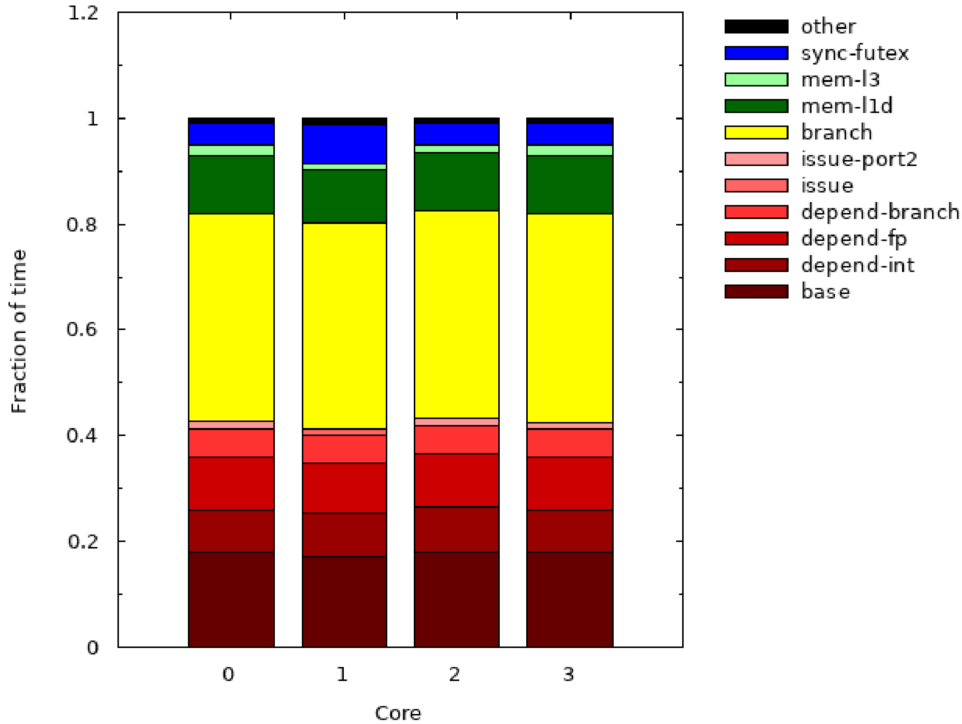
5.4.2 CPI Stacks



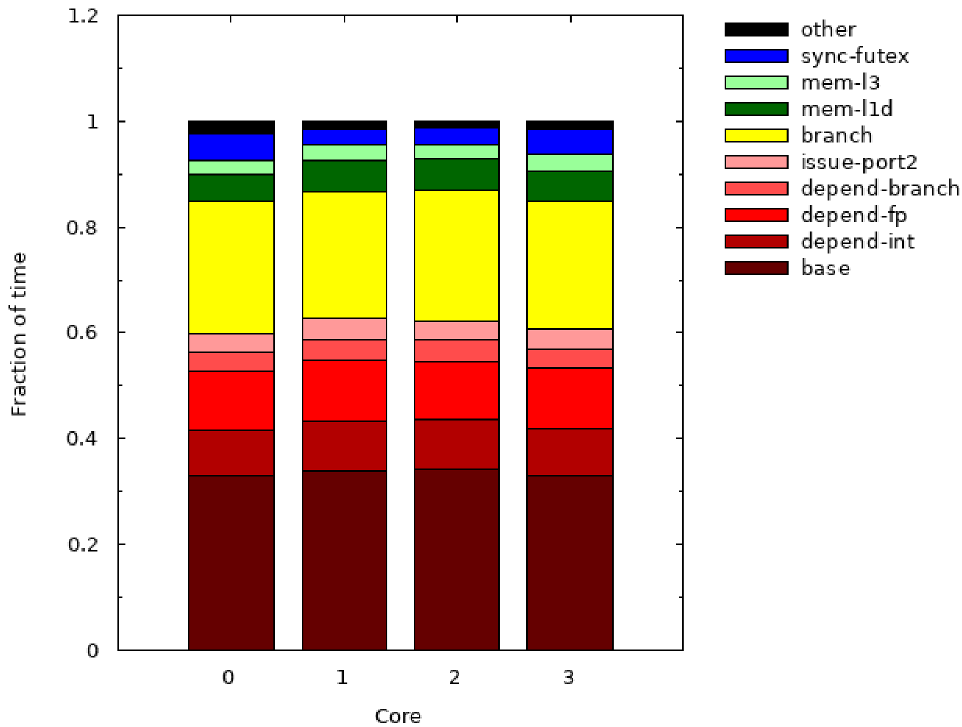
(a) default



(b) 2^4 PHT size



(c) 2^8 PHT size



(d) 2^{16} PHT size

Figure 21: CPI stacks for various PHT sizes.

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.13	0.13	0.13	0.13
depend-fp	0.15	0.15	0.14	0.15
depend-branch	0.10	0.10	0.10	0.10
issue	0.02	0.02	0.02	0.02
branch	0.90	0.85	0.81	0.85
mem-l1d	0.22	0.21	0.21	0.21
mem-l3	0.02	0.02	0.02	0.02
sync-futex	0.04	0.03	0.03	0.04
other	0.02	0.01	0.01	0.01
total	1.85	1.77	1.72	1.79

(a) default

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.13	0.13	0.14	0.13
depend-fp	0.15	0.15	0.14	0.15
depend-branch	0.10	0.10	0.10	0.10
issue	0.02	0.02	0.02	0.02
branch	0.84	0.83	0.87	0.84
mem-l1d	0.21	0.21	0.21	0.21
mem-l3	0.03	0.02	0.00	0.03
sync-futex	0.06	0.06	0.11	0.06
other	0.01	0.01	0.03	0.01
total	1.80	1.78	1.87	1.80

(b) 2^4 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.11	0.12	0.11	0.11
depend-fp	0.14	0.14	0.14	0.14
depend-branch	0.07	0.08	0.07	0.07
issue	0.00	0.02	0.00	0.00
issue-port2	0.02	0.00	0.02	0.02
branch	0.55	0.57	0.54	0.55
mem-l1d	0.15	0.15	0.15	0.15
mem-l3	0.03	0.02	0.02	0.03
sync-futex	0.06	0.11	0.06	0.06
other	0.01	0.02	0.01	0.01
total	1.40	1.46	1.38	1.40

(c) 2^8 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.07	0.07	0.07	0.07
depend-fp	0.08	0.09	0.08	0.09
depend-branch	0.03	0.03	0.03	0.03
issue-port2	0.03	0.03	0.03	0.03
branch	0.19	0.18	0.18	0.18
mem-l1d	0.04	0.04	0.04	0.04
mem-l3	0.02	0.02	0.02	0.02
sync-futex	0.04	0.02	0.02	0.04
other	0.02	0.01	0.01	0.01
total	0.76	0.73	0.73	0.76

(d) 2^{16} PHT size

Figure 22: Specific values for each components' CPI stack.

5.5 raytrace

5.5.1 Power Results

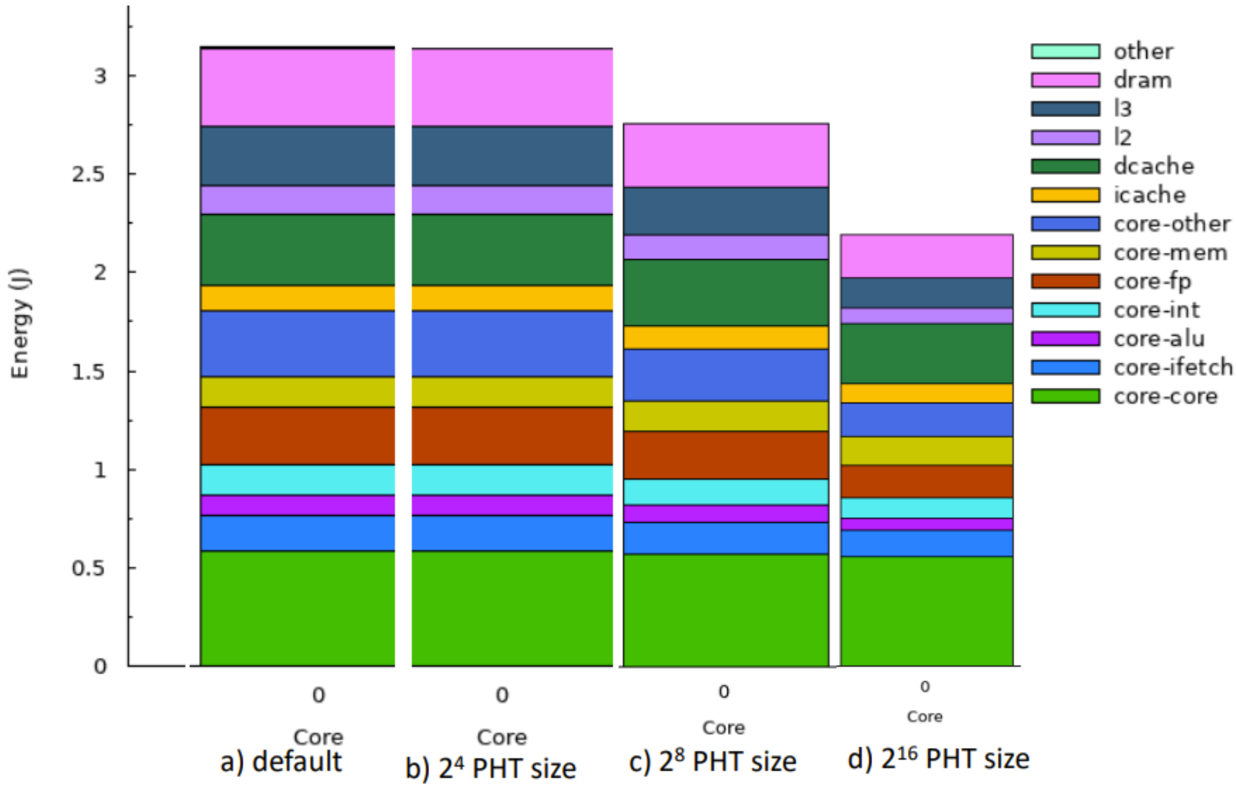


Figure 23: Processor power for various PHT sizes.

	Power	Energy	Energy %
core-core	6.74 W	0.59 J	18.71%
core-ifetch	1.99 W	0.17 J	5.51%
core-alu	1.27 W	0.11 J	3.53%
core-int	1.72 W	0.15 J	4.78%
core-fp	3.37 W	0.29 J	9.34%
core-mem	1.79 W	0.16 J	4.98%
core-other	3.79 W	0.33 J	10.53%
icache	1.54 W	0.13 J	4.27%
dcache	4.11 W	0.36 J	11.41%
l2	1.69 W	0.15 J	4.69%
l3	3.42 W	0.30 J	9.48%
dram	4.57 W	0.40 J	12.68%
other	0.03 W	2.85 mJ	0.09%
core	20.68 W	1.80 J	57.38%
cache	10.76 W	0.94 J	29.85%
total	36.03 W	3.14 J	100.00%

(a) default

	Power	Energy	Energy %
core-core	6.74 W	0.59 J	18.71%
core-ifetch	1.99 W	0.17 J	5.51%
core-alu	1.27 W	0.11 J	3.53%
core-int	1.72 W	0.15 J	4.78%
core-fp	3.37 W	0.29 J	9.34%
core-mem	1.79 W	0.16 J	4.98%
core-other	3.79 W	0.33 J	10.53%
icache	1.54 W	0.13 J	4.27%
dcache	4.11 W	0.36 J	11.41%
l2	1.69 W	0.15 J	4.69%
l3	3.42 W	0.30 J	9.48%
dram	4.57 W	0.40 J	12.68%
other	0.03 W	2.85 mJ	0.09%
core	20.68 W	1.80 J	57.38%
cache	10.76 W	0.94 J	29.85%
total	36.04 W	3.14 J	100.00%

(b) 2^4 PHT size

	Power	Energy	Energy %
core-core	8.18 W	0.58 J	20.83%
core-ifetch	2.22 W	0.16 J	5.65%
core-alu	1.30 W	0.09 J	3.31%
core-int	1.86 W	0.13 J	4.75%
core-fp	3.46 W	0.24 J	8.81%
core-mem	2.14 W	0.15 J	5.46%
core-other	3.79 W	0.27 J	9.66%
icache	1.70 W	0.12 J	4.34%
dcache	4.82 W	0.34 J	12.27%
l2	1.69 W	0.12 J	4.31%
l3	3.43 W	0.24 J	8.72%
dram	4.65 W	0.33 J	11.82%
other	0.03 W	2.31 mJ	0.08%
core	22.97 W	1.62 J	58.46%
cache	11.64 W	0.82 J	29.64%
total	39.29 W	2.77 J	100.00%

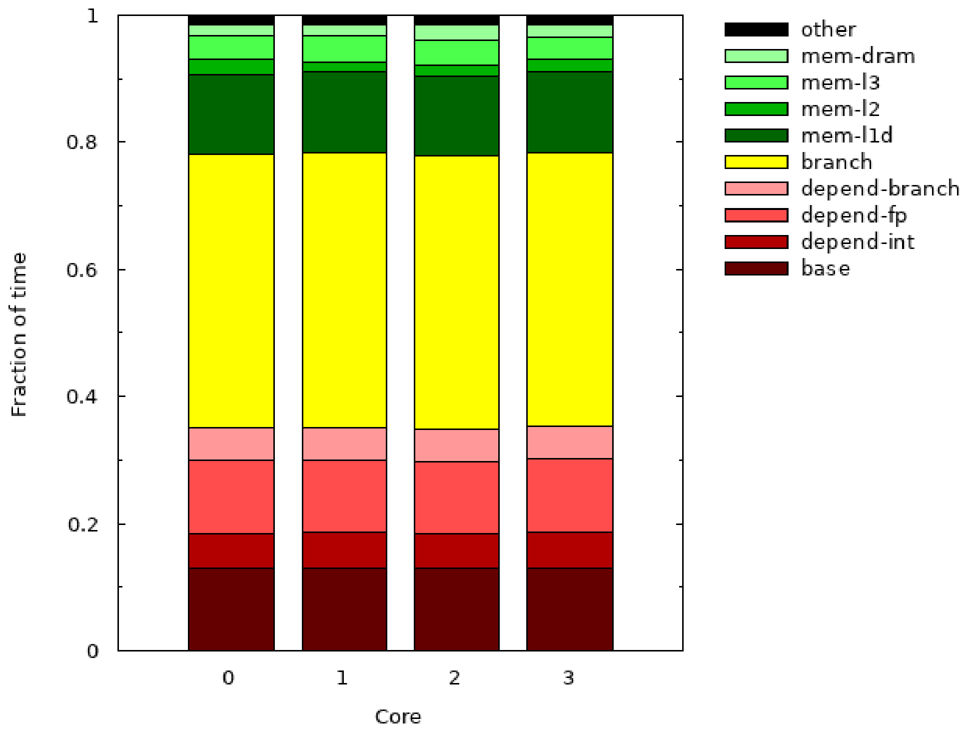
(c) 2^8 PHT size

	Power	Energy	Energy %
core-core	12.45 W	0.56 J	25.45%
core-ifetch	2.92 W	0.13 J	5.96%
core-alu	1.37 W	0.06 J	2.81%
core-int	2.29 W	0.10 J	4.68%
core-fp	3.74 W	0.17 J	7.64%
core-mem	3.18 W	0.14 J	6.51%
core-other	3.79 W	0.17 J	7.76%
icache	2.19 W	0.10 J	4.48%
dcache	6.92 W	0.31 J	14.14%
l2	1.71 W	0.08 J	3.50%
l3	3.46 W	0.16 J	7.06%
dram	4.87 W	0.22 J	9.95%
other	0.03 W	1.48 mJ	0.07%
core	29.75 W	1.34 J	60.80%
cache	14.28 W	0.64 J	29.18%
total	48.92 W	2.21 J	100.00%

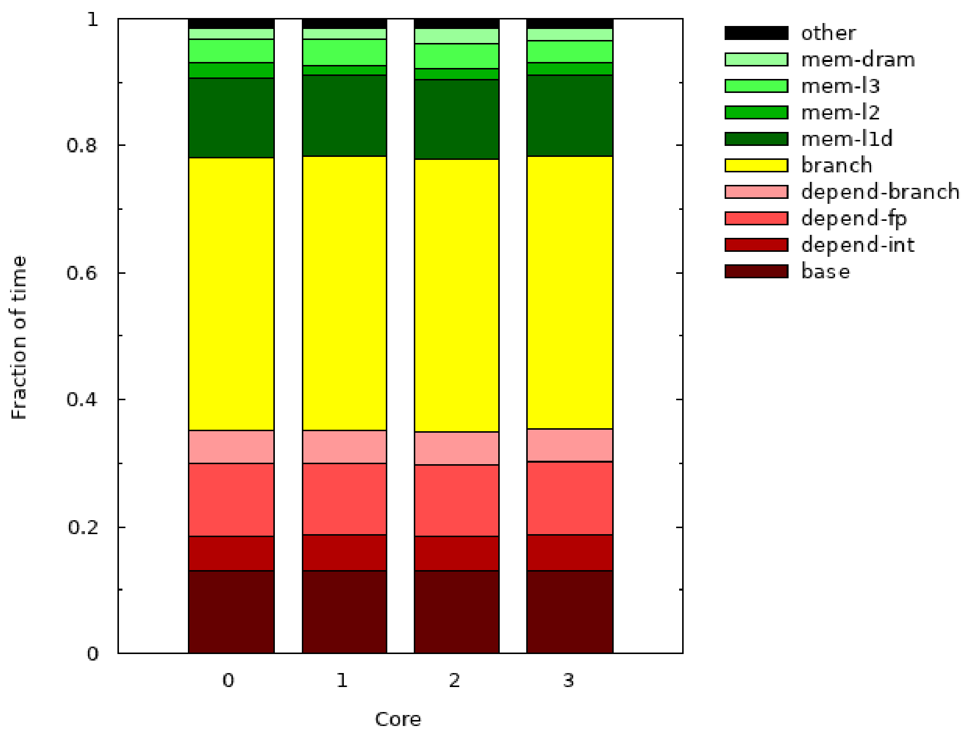
(d) 2^{16} PHT size

Figure 24: Specific values for each components' power consumption.

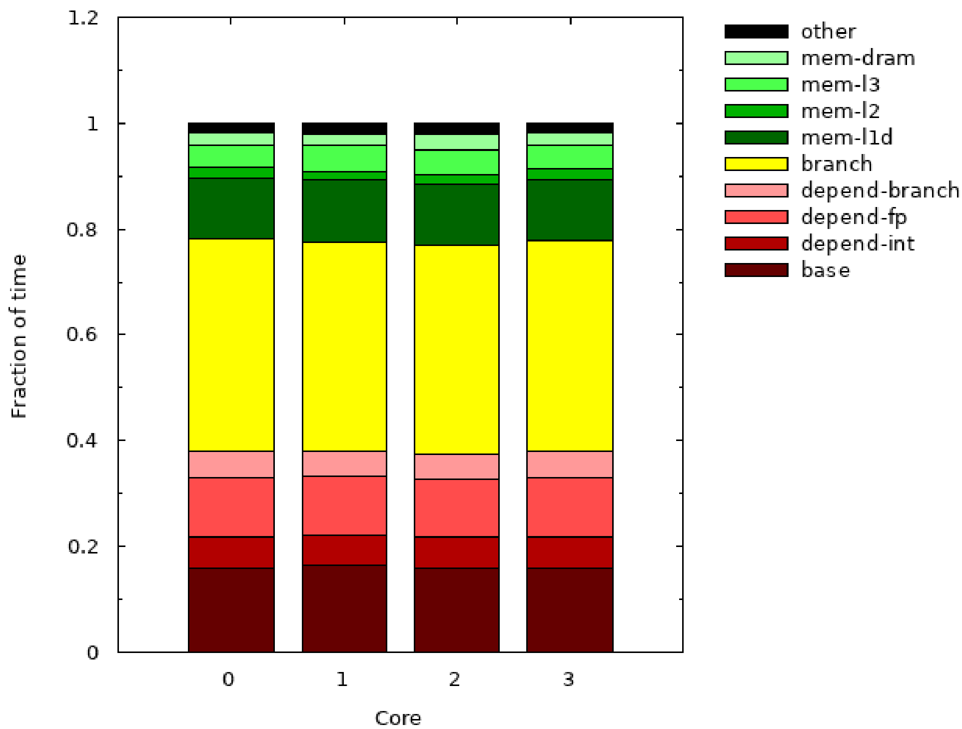
5.5.2 CPI Stacks



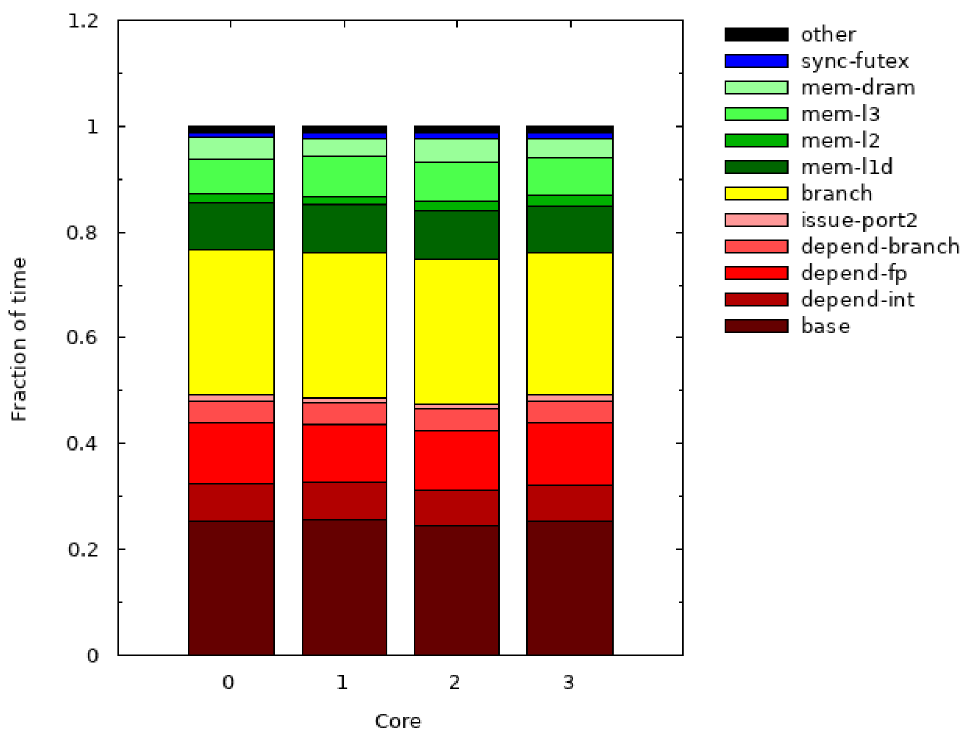
(a) default



(b) 2^4 PHT size



(c) 2^8 PHT size



(d) 2^{16} PHT size

Figure 25: CPI stacks for various PHT sizes.

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.11	0.11	0.11	0.11
depend-fp	0.22	0.22	0.22	0.22
depend-branch	0.10	0.10	0.10	0.10
branch	0.83	0.83	0.83	0.82
mem-l1d	0.24	0.24	0.24	0.25
mem-l2	0.05	0.03	0.04	0.04
mem-l3	0.07	0.08	0.07	0.06
mem-dram	0.04	0.03	0.05	0.04
other	0.03	0.03	0.03	0.03
total	1.94	1.91	1.93	1.92

(a) default

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.11	0.11	0.11	0.11
depend-fp	0.22	0.21	0.22	0.22
depend-branch	0.10	0.10	0.10	0.10
branch	0.82	0.83	0.83	0.83
mem-l1d	0.25	0.24	0.24	0.24
mem-l2	0.04	0.03	0.04	0.05
mem-l3	0.06	0.07	0.08	0.07
mem-dram	0.04	0.03	0.05	0.04
other	0.03	0.03	0.03	0.03
total	1.92	1.91	1.93	1.93

(b) 2^4 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.09	0.09	0.09	0.09
depend-fp	0.18	0.17	0.17	0.18
depend-branch	0.08	0.07	0.08	0.08
branch	0.63	0.60	0.62	0.63
mem-l1d	0.18	0.18	0.18	0.18
mem-l2	0.03	0.02	0.03	0.03
mem-l3	0.06	0.08	0.08	0.07
mem-dram	0.04	0.03	0.05	0.04
other	0.03	0.03	0.03	0.03
total	1.56	1.52	1.56	1.57

(c) 2^8 PHT size

CPI	Core 0	Core 1	Core 2	Core 3
base	0.25	0.25	0.25	0.25
depend-int	0.07	0.07	0.07	0.07
depend-fp	0.11	0.11	0.11	0.12
depend-branch	0.04	0.04	0.04	0.04
issue-port2	0.01	0.01	0.01	0.01
branch	0.27	0.27	0.28	0.27
mem-l1d	0.09	0.09	0.09	0.09
mem-l2	0.02	0.01	0.02	0.02
mem-l3	0.06	0.07	0.08	0.07
mem-dram	0.04	0.03	0.05	0.04
sync-futex	0.01	0.01	0.01	0.01
other	0.01	0.01	0.01	0.01
total	0.98	0.98	1.02	0.99

(d) 2^{16} PHT size

Figure 26: Specific values for each components' CPI stack.