Design Project 1: Caches

Connor Smith

I. Introduction

In a perfect world, the entire memory space would lie within a single cycle's reach of a CPU. Since constraints on things such as cost, manufacturing, and heat do not allow for this, designers insert caches into the memory hierarchy to provide a relatively small amount of fast memory. In order to accomplish these goals, the designer must select values for a large number of design parameters, each of which will have effects on each other as well as on overall metrics of performance, power, area, etc. A directed methodology is necessary to narrow the design space in order to produce the caches that yield the best results.

II. Design Methodology

For the purposes of this project, the caches for each configuration are composed of an instruction-cache and data-cache of the same capacity and associativity. Separate instruction and data caches are very common in the L1 level, and it is not unusual for both to be the same capacity and have the same associativity. For L2, both unified and split caches are common, although the main reason for not using a unified design was for simplicity in running simulations and computing results. Cache capacities ranged from 4KB to 2MB, while associativity varied from 1-way (direct-mapped) to 8-way, and block size was either 32B or 64B. The number of banks in each cache was kept at 1, and a constant 32nm technology node was assumed. For each configuration, version 5.3 of HP Lab's CACTI application was used to generate the relevant cache data. The SimpleScalar simulator was then used to run a comprehensive set of 43 SPEC benchmarks. For the sake of simplicity, the cache access time was used as the cycle time for the simulations, which may not be less than 0.2ns (corresponding to a 5GHz clock speed), and a simple FIFO cache replacement policy was used.

The design decisions in this project were chosen to create a cache suited for the desktop computing market. As such, the primary metric of interest is raw performance, measured in execution time, with mild constraints on total power dissipation and area. The initial configuration models a typical desktop Intel i5/i7 processor with a Haswell microarchitecture. The details for this configuration are shown in Table 1.

TABLE 1. HASWELL CACHE CONFIGURATION

L1 Capacity	32KB
L1 Associativity	8
L2 Capacity	128KB
L2 Associativity	8
Block Size	64

This configuration was then modified in each of the five design parameters shown above. Since the potential design space is so massive, optimizations were made in an exploratory fashion by varying one parameter at a time. If the new design yielded better results, the next iteration would modify the same parameter again. Once the results were no longer beneficial, the next design parameter was chosen. The L1 capacity and associativity were varied first, since these are likely to have a significant effect on the L2 design as well as main memory performance.

Several metrics were used to justify whether the new configuration was better or worse than the previous. The most important of these metrics for the chosen market is execution time. The total execution time for each benchmark was computed by multiplying the number of cycles run by the cycle time of the processor. These results were then aggregated into an arithmetic mean to illustrate the "average" case, as well as a median to resist the effects of outliers, such as those due to the 181.mcf.spec_ref benchmark. The minimum and maximum execution times were also recorded so that outliers were not completely removed from consideration. Note that instructions per cycle (IPC) or cycles per instruction (CPI) were not used, since comparison of these values requires that clock speed not change, a requirement that is not met across the designs for this project. Total power dissipation is also a concern in desktop computing, so "good" designs were limited to a power dissipation of within 2x the Haswell configuration. This power is computed by summing the static and dynamic power for both L1 and L2, as shown below:

$$P_{tot}\ (W) = P_{static} + P_{dynamic} = 2 * (P_{leak,L1} + P_{leak,L2} + P_{dyn,L1} + P_{dyn,L2})$$

The last metric of concern is total area, which was also limited to within 2x the total area of the Haswell configuration. The area is computed by summing the areas of the L1 and L2 caches:

$$A_{tot}\ (mm^2) = A_{L1} + A_{L2}$$

III. RESULTS

The results for the Haswell configuration serve as a baseline for the rest of the design. Figure 1 shows the execution time for each benchmark. Note the large execution times for benchmarks 179-183; these are likely to skew the mean, which is why the median is also used. Table 2 shows the relevant metrics for this configuration.

FIGURE 1. HASWELL BENCHMARK EXECUTION TIMES

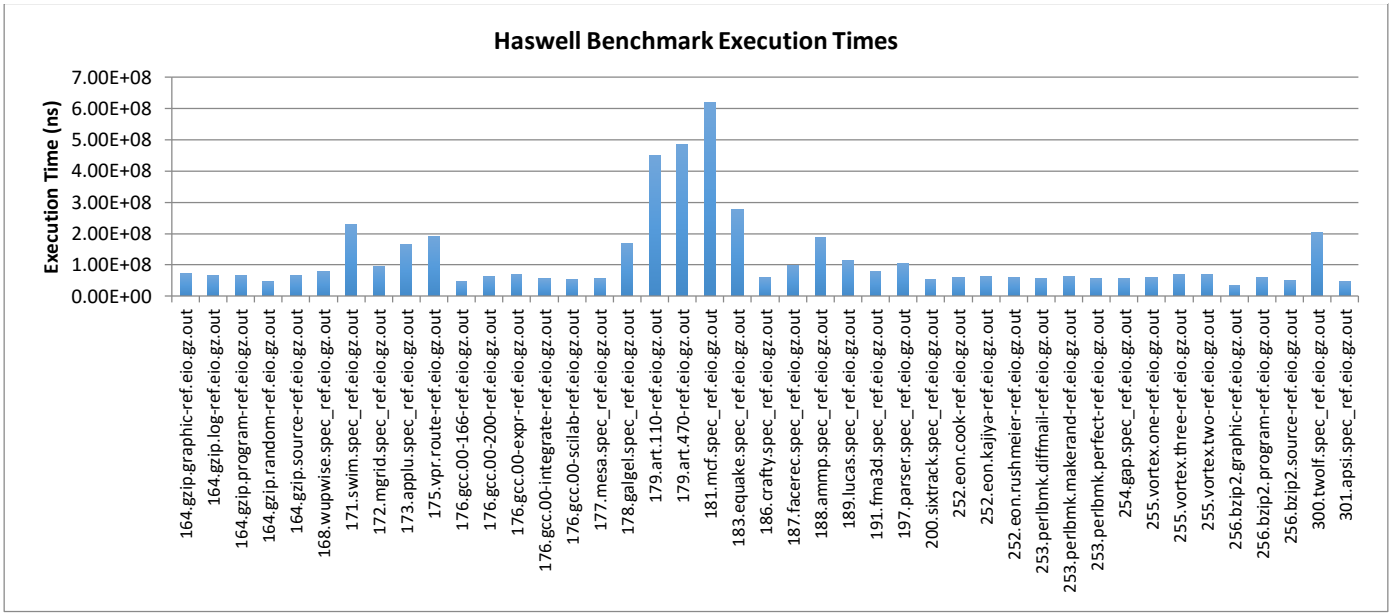


TABLE 2. HASWELL RESULTS

Mean time (ns)	1.19E+08
Median time (ns)	6.53E+07
Maximum time (ns)	6.17E+08
Minimum time (ns)	3.57E+07
Power dissipation (W)	3.345480672
Area (mm^2)	3.229457102

TABLE 3. LOWER L1 ASSOCIATIVITY RESULTS

	4-way	2-way	1-way
Mean time (ns)	1.00E+08	9.06E+07	8.39E+07
Median time (ns)	4.80E+07	4.01E+07	3.45E+07
Maximum time (ns)	5.85E+08	5.66E+08	5.55E+08
Minimum time (ns)	2.33E+07	1.71E+07	1.23E+07
Power dissipation (W)	2.453271341	2.067508994	2.127816421
Area (mm^2)	2.669086571	2.3882799	2.290956509

The first change made to the design was lowering the L1 cache associativity from 8-way to 4-way. This change resulted in benefits across the board, including a 1.2x speedup using only 73% of the power and 83% of the area of the original configuration. The L1 associativity was further decreased to 2-way and then direct-mapped, shown in Tables 4 and 5. The results continued to be beneficial; with a direct-mapped L1 cache, a 1.4x speedup was achieved using only 64% of the power and 71% of the area of the Haswell configuration. The results are shown in Table 3,

and a graph of mean and median execution times is given in Figure 2. As expected, a direct-mapped L1 cache yields better performance as well as dissipating less power and taking up less area.

FIGURE 2. LOWER L1 ASSOCIATIVITY EXECUTION TIMES

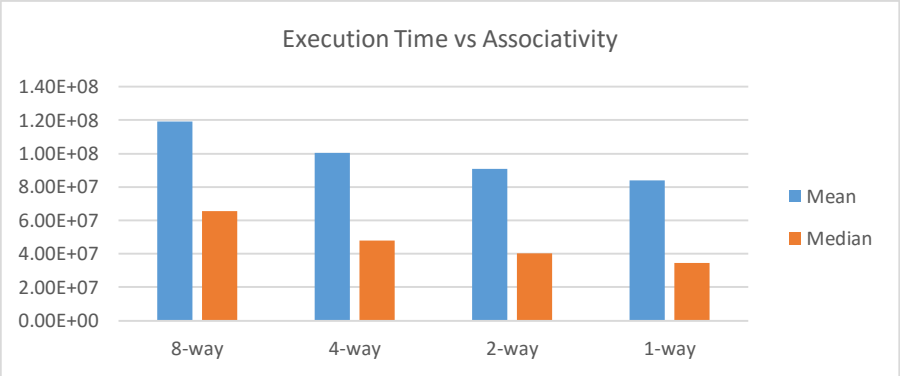


TABLE 4. SMALLER L1 CAPACITY RESULTS

	32KB	16KB	8KB	4KB
Mean time (ns)	8.39E+07	8.08E+07	8.12E+07	8.20E+07
Median time (ns)	3.45E+07	3.15E+07	3.10E+07	3.18E+07
Maximum time (ns)	5.55E+08	5.51E+08	5.53E+08	5.56E+08
Minimum time (ns)	1.23E+07	1.04E+07	9.98E+06	1.35E+07
Power dissipation (W)	2.127816421	1.880132262	1.855850188	1.844680032
Area (mm^2)	2.290956509	2.200989105	2.162909354	2.143699949

The next optimization attempt was made by shrinking the L1 cache. Table 4 shows the results of having a 16KB, 8KB, and 4KB L1 cache. Shrinking down to 8KB resulted in slightly better execution times, power dissipation, and area. However, the 4KB L1 cache actually performed slightly worse than its counterparts. This is likely due to an increase in miss rate because the cache is smaller. While the 4KB cache does have a slightly smaller power and area footprint, the reduction in performance is not justified, so the 8KB configuration is the best. Having reduced associativity and capacity, the last parameter change for the L1 cache was to reduce the block size from 64B to 32B. The results, shown in Table 5, were disappointing. Decreasing the block size had no positive effects on execution time, and nearly negligible benefits on power dissipation and area. It appears the benchmarks benefit from having a larger block size, so this parameter remained at 64B for future configurations.

TABLE 5. REDUCED BLOCK SIZE RESULTS

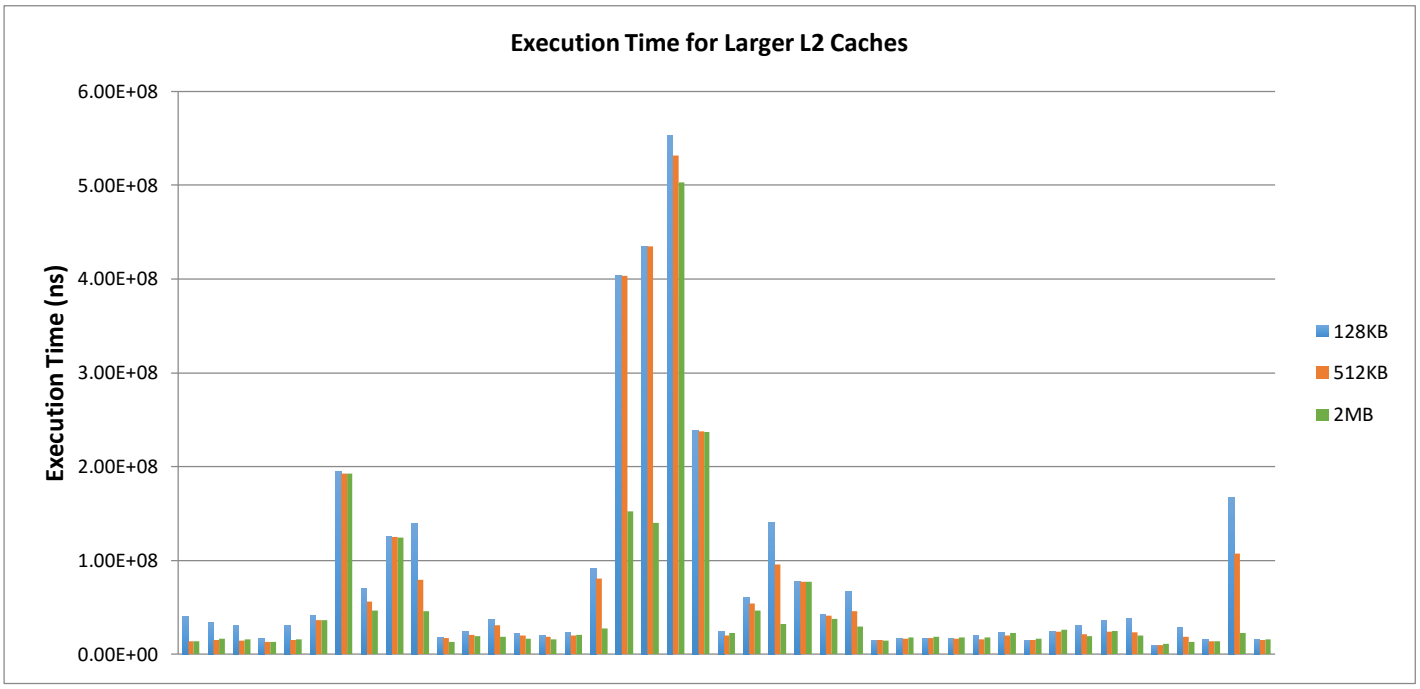
	64B blocks	32B blocks
Mean time (ns)	8.12E+07	1.13E+08
Median time (ns)	3.10E+07	6.05E+07
Maximum time (ns)	5.53E+08	5.69E+08
Minimum time (ns)	9.98E+06	1.54E+07
Power dissipation (W)	1.855850188	1.823030816
Area (mm^2)	2.162909354	2.03889281

With optimal parameters for L1 chosen, the L2 cache was next. The capacity of the L2 cache was varied from the original 128KB up to 2MB, with results shown in Table 6. Increasing the capacity of the L2 cache had a dramatic impact on the execution time. For capacities 512KB and larger, there was nearly a 1.5x speedup in median execution time. Note that the mean execution time did not decrease as substantially until the capacity reached 2MB; this is because of the outliers mentioned previously, and can best be seen in Figure 3. The largest decrease in execution time for most of the benchmarks can be seen in the transition from a 128KB cache to 512KB, but the longest-running benchmarks are hardly affected until the L2 cache is 2MB. Diminishing returns limited the effectiveness of making the L2 cache too large, however. There was hardly a benefit in median time when exceeding 512KB, and the power dissipation continues to increase, nearing the original Haswell value of 3.345W. Total area explodes as well, growing 3x alone in the transition from 512KB to 2MB. These tradeoffs are simply not worth it, so the 512KB configuration is best so far.

TABLE 6. LARGER L2 CAPACITY RESULTS

	128KB	256KB	512KB	1MB	2MB
Mean time (ns)	8.12E+07	7.61E+07	7.18E+07	6.05E+07	5.18E+07
Median time (ns)	3.10E+07	2.33E+07	2.10E+07	2.12E+07	2.03E+07
Maximum time (ns)	5.53E+08	5.42E+08	5.32E+08	5.22E+08	5.03E+08
Minimum time (ns)	9.98E+06	9.97E+06	9.96E+06	1.04E+07	1.09E+07
Power dissipation (W)	1.855850188	1.965696723	2.597437844	2.421634384	2.744118
Area (mm^2)	2.162909354	2.687168164	4.018394706	6.073273434	12.03310412

FIGURE 3. LARGER L2 EXECUTION TIMES

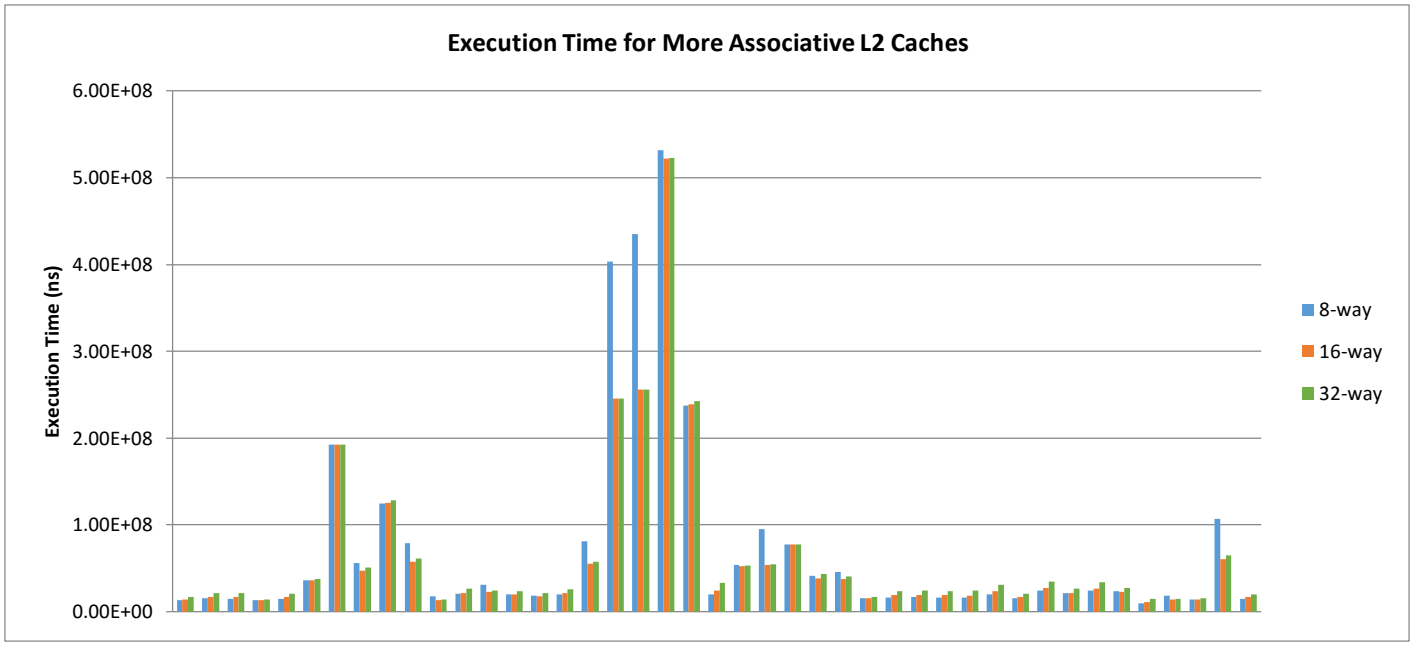


The final chosen design parameter to alter was L2 associativity. The associativity was increased from 8-way to 16-way and 32-way, with the results shown in Table 7. While increasing the associativity did slightly decrease the mean execution times, the median time actually increased, which implies that only the long-running outliers saw a noticeable benefit. This is confirmed by viewing the individual benchmark results in Figure 4. In addition, the power dissipation grew by a factor of 4.5 and area increased by a factor of 2.2, which is unacceptable for such a small performance boost in only some of the benchmarks. Thus the L2 cache remained 8-way associative.

TABLE 7. LARGER L2 ASSOCIATIVITY RESULTS

	8-way	16-way	32-way
Mean time (ns)	7.18E+07	6.06E+07	6.39E+07
Median time (ns)	2.10E+07	2.28E+07	2.63E+07
Maximum time (ns)	5.32E+08	5.22E+08	5.23E+08
Minimum time (ns)	9.96E+06	1.14E+07	1.38E+07
Power dissipation (W)	2.597437844	6.307958234	11.61194676
Area (mm^2)	4.018394706	6.436712734	8.952232063

FIGURE 4. MORE ASSOCIATIVE L2 EXECUTION TIMES



As one final possible optimization, the block size was reduced again from 64B to 32B to see if it would have a positive effect after increasing the L2 cache's capacity. Table 8 and Figure 5 show the results of this final configuration. There was a slight performance loss overall based off the mean and median execution times, but it is important to note that the maximum and minimum times did not change very significantly. In addition, the smaller block size resulted in only 35% of the power dissipation and 70% of the area of the 64B block size configuration. Which configuration is "best" will likely be determined by the exact market for the CPU as well as CPU design parameters not directly related to the cache. Table 8 and Figure 5 also show the results for the starting Haswell configuration. Both final configurations show a significant improvement in all areas.

TABLE 8. REDUCED BLOCK SIZE RESULTS

	Haswell	64B	32B
Mean time (ns)	1.19E+08	7.18E+07	8.49E+07
Median time (ns)	6.53E+07	2.10E+07	2.35E+07
Maximum time (ns)	6.17E+08	5.32E+08	5.33E+08
Minimum time (ns)	3.57E+07	9.96E+06	9.22E+06
Power dissipation (W)	3.345480672	2.597437844	0.923735507
Area (mm^2)	3.229457102	4.018394706	2.848673903

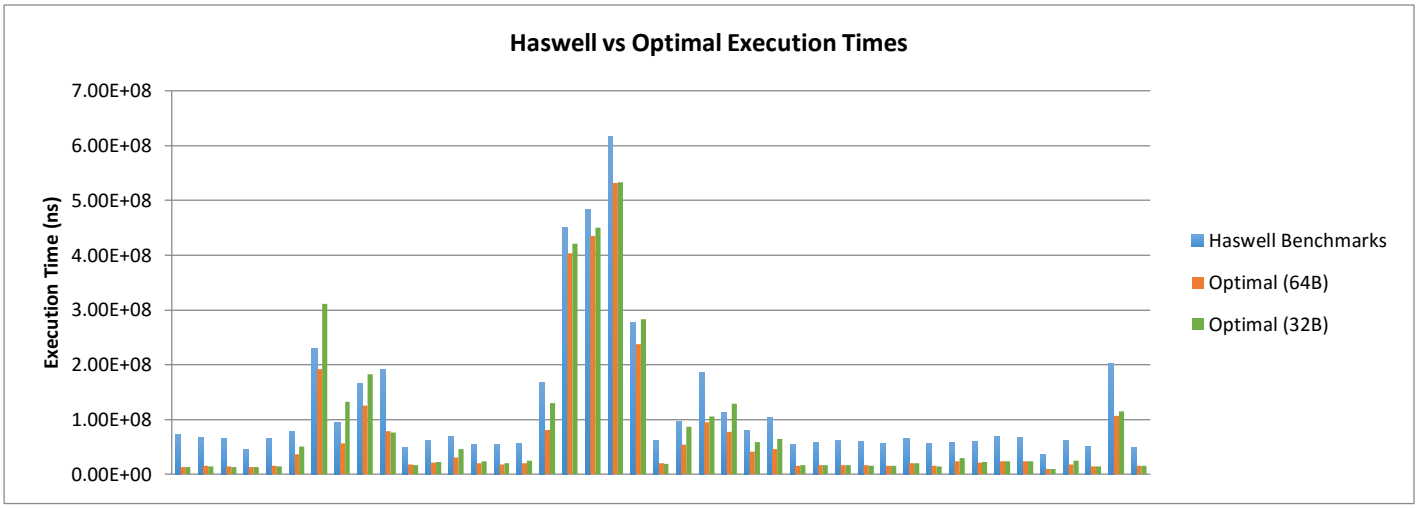


TABLE 9. HASWELL VS FINAL CONFIGURATIONS

	Haswell	Optimal
L1 Size (each)	32KB	8KB
L1 Associativity	8	1
L2 Size (each)	128KB	512KB
L2 Associativity	8	8
Block Size	64	32/64
L1 Access time (ns)	0.9790	0.2313

IV. CONCLUSION

Table 9 summarizes the changes in configuration made between the starting configuration and the two optimal solutions. The best configurations for performance, power, and size were the result of moderate values; as usual, extremes are of little use. It is interesting that the "best" configurations differ quite significantly from Intel's Haswell design. In the best configuration, the L1 cache is smaller and has a lower associativity. Course texts usually state that smaller, direct-mapped caches are better for L1, which matches the results found in this project. On the other hand, the L2 cache in the best configurations is larger than what the Haswell architecture uses. The block size in the optimal configurations could be either 32B or 64B. These differences are due in part to simplifications and assumptions made by CACTI and SimpleScalar that do not reflect the full set of constraints found in industry design. It is also important to recall that the L1 access time was assumed to be one cycle, which is likely not the case in modern processors. If anything, this shows that oversimplifying during the design process can yield quite different results than reality.