ASSIGNMENT 6

PREKSHA PATANGE

MC23BT029

Abstract

This report explores the influence of varying cache configurations on overall processor performance. In particular, it focuses on a two-level cache hierarchy comprising separate level-1 instruction (L1i) and data (L1d) caches. Multiple configurations of L1 cache sizes are implemented and evaluated. The system's performance is assessed across benchmark programs using the Instructions Per Cycle (IPC) metric, which serves as a key indicator of processor efficiency. Through this analysis, we aim to identify optimal cache configurations that enhance execution performance.

1 Experimental Setup

Two caches were introduced in the pipeline:

- **L1i-cache** (Instruction Cache): Between the IF stage and main memory.
- **L1d-cache** (Data Cache): Between the MA stage and main memory.

Cache Parameters:

- Cache sizes: 16B, 128B, 512B, 1kB
- Line size: 4B
- Associativity: 2-way set associative
- Write Policy: Write Through
- Latency: 1 to 4 cycles (increasing with size)

2 Analysis 1: Varying L1i-cache Size

The size of the L1i-cache was varied while the L1d-cache was fixed at 1kB.

Table 1: IPC vs L1i-cache Size (L1d-cache fixed at 1kB)

Benchmark	**16B (1-cycle)**	**128B (2-cycle)**	**512B (3-cycle)**	**1kB (4-cycle)**
Descending	0.025002	0.11878	0.10448	0.09750
Prime	0.03243	0.04538	0.04347	0.04172
Even or Odd	0.02238	0.02197	0.02158	0.02120
Fibonacci	0.02280	0.05368	0.04797	0.04604
Palindrome	0.03026	0.07174	0.06388	0.06012

3 Analysis 2: Varying L1d-cache Size

The size of the L1d-cache was varied while the L1i-cache was fixed at 1kB.

Table 2: IPC vs L1d-cache Size (L1i-cache fixed at 1kB)

Benchmark	16B (1-cycle)	128B (2-cycle)	512B (3-cycle)	1kB (4-cycle)
Descending	0.09750	0.10386	0.10058	0.09750
Prime	0.04172	0.04184	0.04178	0.04172
Even or Odd	0.02120	0.02135	0.02127	0.02120
Fibonacci	0.04604	0.04785	0.04693	0.046044
Palindrome	0.060122	0.60270	0.060196	0.06012

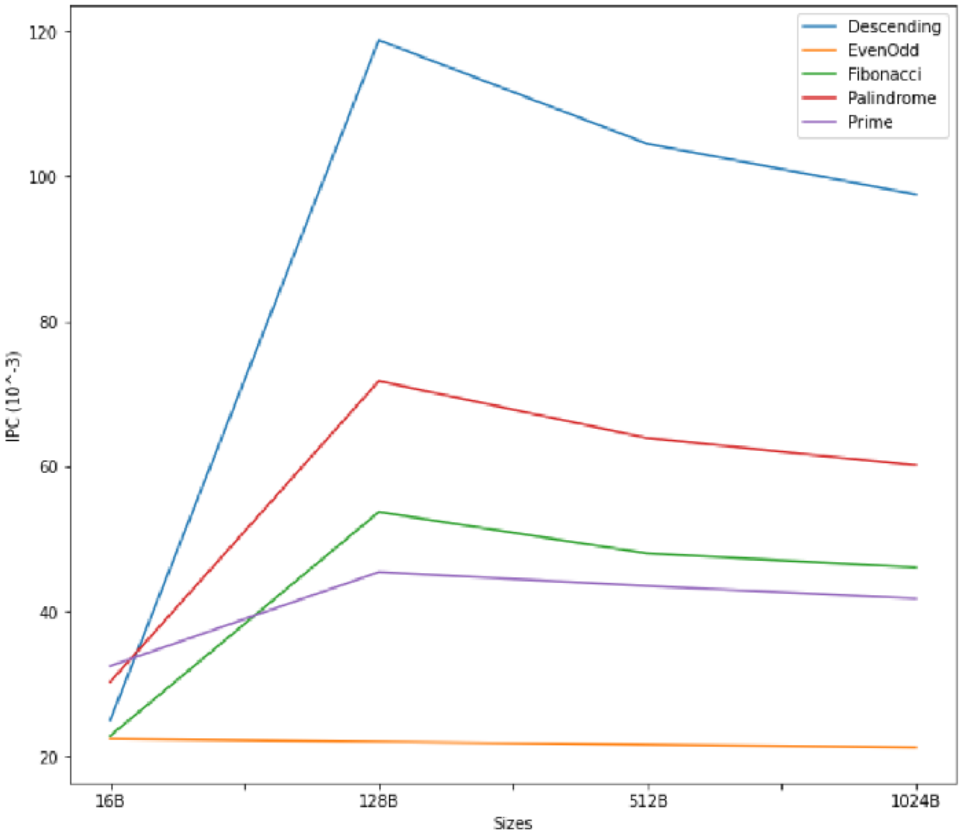

Figure 1: Combined IPC values for all benchmarks for L1d(1KB) cache configurations

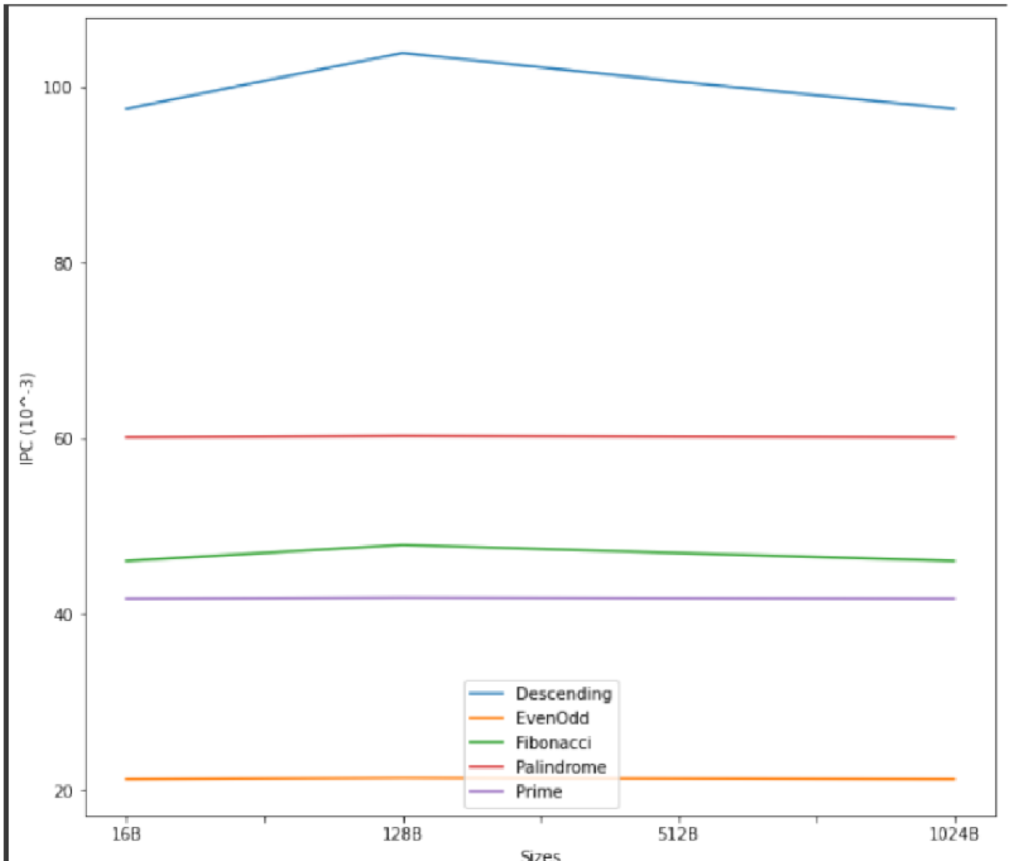

Figure 2: Combined IPC values for all benchmarks for L1i(1KB) cache configurations

4 Observations

From the results, we observe:

4.1 Impact of Varying L1i-cache Size (L1d-cache fixed at 1kB)

- IPC significantly increases when the L1i-cache size is increased from 16B to 128B, especially for benchmarks like *Descending*, *Fibonacci*, and *Palindrome*.
- Benchmarks such as *Even or Odd* show minimal changes in IPC, implying a small instruction working set or limited dependency on instruction cache size.
- *Descending* shows the highest sensitivity, with IPC jumping from 0.025 to 0.118 between 16B and 128B, demonstrating a strong dependence on instruction cache size.
- *Fibonacci* and *Palindrome* also benefit from larger instruction caches, though to a slightly lesser degree.

4.2 Impact of Varying L1d-cache Size (L1i-cache fixed at 1kB)

- The IPC remains nearly constant across varying L1d-cache sizes for most benchmarks, suggesting that performance is not highly sensitive to L1d-cache size.
- Benchmarks like *Prime*, *Even or Odd*, and *Fibonacci* display negligible variation in IPC, likely due to small data footprints or minimal memory intensity.
- Increasing L1i-cache size up to 128B significantly improves performance for instruction-heavy benchmarks.
- L1d-cache size has a lesser effect on performance, as the data access patterns in the tested programs are not memory-intensive.

5 Toy Benchmark

For Toy benchmark we have used greatest program which finds greatest element from given element.

For L1d=1024B:

- L1i=16B to 128B , IPC = 0.022680 to 0.022312

For L1i=1024B:

- L1d=16B to 128B , IPC = 0.0223 to 0.0218