Computer Architecture

210010017 & 210010036

October 29, 2023

Cache Configuration:

Cache size	16B	128B	512B	1kB
Latency	1 cycle	2 cycles	3 cycles	4 cycles
Line Size	4B			
Associativity	2			
Write Policy	Write Through			

Varying Instruction Cache:

First we fix the size of the L1d-cache at 1kB. Then, we vary the size of the L1i-cache from 16B to 1kB (remember to change latency accordingly), and study the performance (instructions per cycle). Plots are included later.

L1d=1024 B

Program	Without Cache	L1i=16 B	L1i=128 B	L1i=512 B	L1i=1024 B
Descending	0.024931798	0.022937221	0.088484846	0.07804148	0.069776334
EvenorOdd	0.024390243	0.02238806	0.021582734	0.020833334	0.020134227
Fibonacci	0.02482679	0.02291565	0.05864005	0.053137366	0.04850361
Palindrome	0.024626208	0.023159636	0.06947891	0.06113537	0.054580897
Prime	0.024630541	0.02375961	0.050221566	0.04576043	0.042027194

Varying Data Cache:

Now we fix the size of the L1i-cache at 1kB. Then, we vary the size of the L1d-cache from 16B to 1kB (remember to change latency accordingly), and study the performance (instructions per cycle). Plots are included later.

L1i=1024 B

Program	Without Cache	L1d=16 B	L1d=128 B	L1d=512 B	L1d=1024 B
Descending	0.024931798	0.0649099	0.07075015	0.07025986	0.069776334
EvenorOdd	0.024390243	0.020134227	0.020134227	0.020134227	0.020134227
Fibonacci	0.02482679	0.04903495	0.04885655	0.04867944	0.04850361
Palindrome	0.024626208	0.054580897	0.054580897	0.054580897	0.054580897
Prime	0.024630541	0.042027194	0.042027194	0.042027194	0.042027194

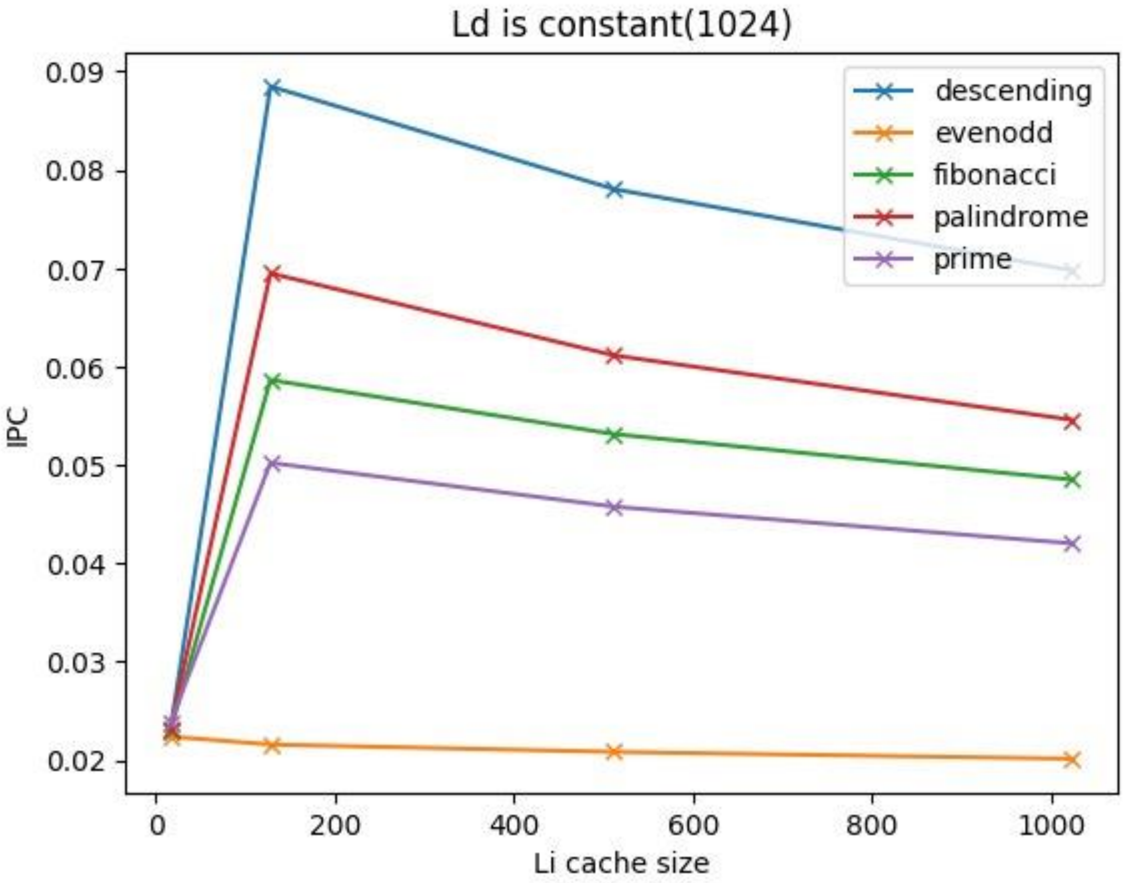

Figure 1

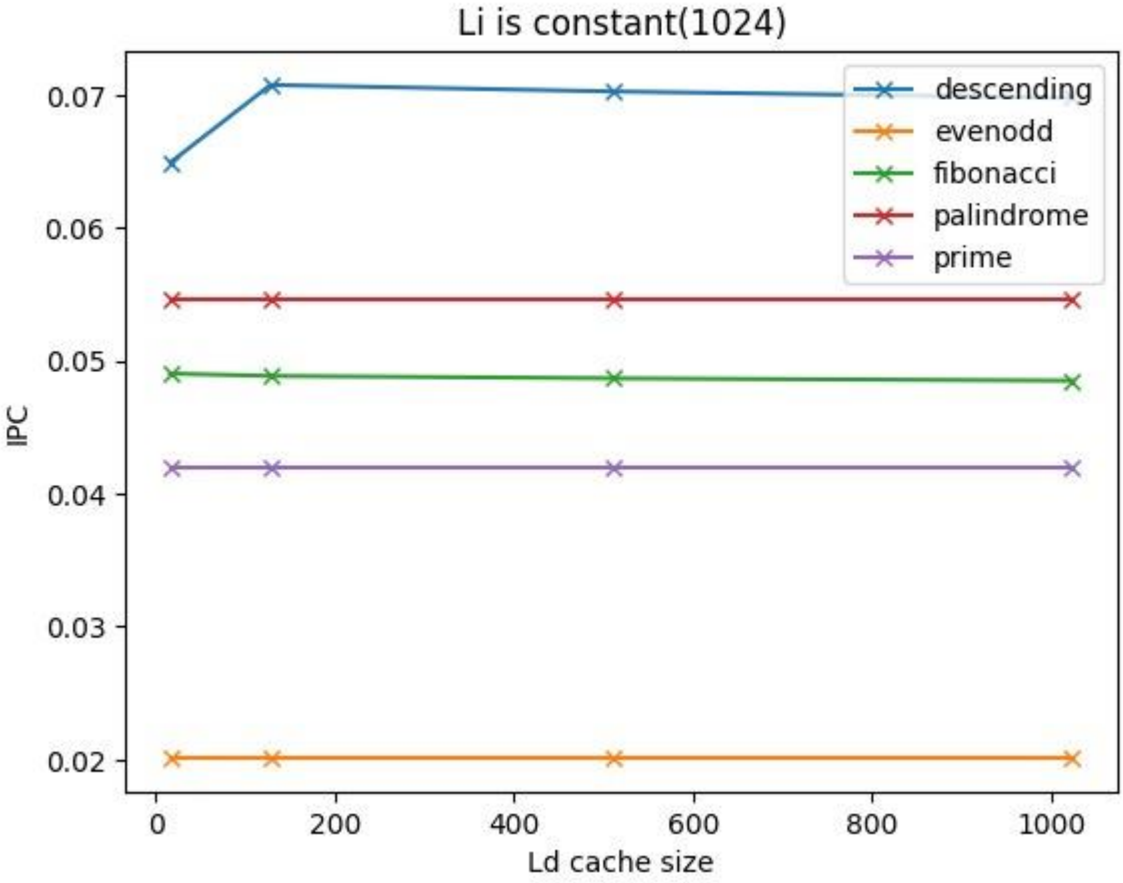

Figure 2

Observations:

L1i Cache Impact:

- Introduction of an L1i cache initially results in a decrease in Instructions Per Cycle (IPC).
- This initial decrease is followed by a sharp increase, except in the case of ‘evenorodd.asm’.
- However, as the size of the L1i cache continues to increase, the IPC later steadily decreases.

L1d Cache Impact:

- The introduction of an L1d cache has a similar pattern: it sharply increases IPC initially.
- With further increases in the size of the L1d cache, the IPC begins to taper off.

Optimal Cache Sizes:

- It can be concluded that, with a significant amount of data, there exists an optimal size for both L1i and L1d caches that maximizes IPC.
- These findings highlight the importance of finding the right balance in cache sizes to achieve optimal system performance.

Impact of Cache Size on Latency and Hit Rate:

- When the L1d cache size is held constant at 1kB and the L1i cache size is varied, a graph (figure 1) shows that latency and hit rate increase with an increase in L1i cache size.
- There is an optimal L1i cache size where IPC reaches its peak performance.

Impact of Instruction Cache Size:

- With a fixed L1i cache size of 1kB and varying L1d cache size, another graph (figure 2) shows that latency increases with L1d cache size up to a certain point, after which it remains constant.
- This suggests that increasing the instruction cache size does not necessarily lead to better performance beyond a certain cache size.

Benchmarking with Descending Assembly Code:

- A specific assembly code benchmark, designed as a toy benchmark, demonstrates a significant increase in IPC and a decrease in latency when both L1d-cache and L1i-cache sizes are increased from 16B to 128B.
- This suggests that the choice of cache size can have a notable impact on performance, and optimizing it can yield substantial improvements in specific workloads.