Assignment 6

Computer Architecture Lab CS311

Chidaksh Ravuru
200010046

Pavan Kumar Patil
200030041

---

1 Task

Add caches to the simulated memory system.

• Add one cache between the IF stage and the main memory. Let us call this the "level 1 instruction cache" or the L1i-cache.

• Add one cache between the MA stage and the main memory. Let us call this the "level 1 data cache" or the L1d-cache.

Perform the following analyses:

1. First fix the size of the L1d-cache at 1kB. Vary the size of the L1i-cache from 16B to 1kB (remember to change latency accordingly), and study the performance (instructions per cycle). Plot your results for the different benchmarks.

2. Now fix the size of the L1i-cache at 1 kB. Vary the size of the L1d-cache from 16B to 1kB, and plot your results for the different benchmarks.

3. In your report, correlate the nature of the benchmark to the observed plot.

4. Create a toy-benchmark that shows significant performance improvement when the L1i-cache is increased from 16B to 128B. Assume favorable L1dcache size.

5. Create a toy-benchmark that shows significant performance improvement when the L1d-cache is increased from 16B to 128B. Assume favorable L1i-cache size.

2 Cache Implementation

2.1 Effect of size of L1i-cache (for a fixed size (1kB) of L1d-cache) on IPC:

The IPC Values that we get for having L1d = 1kB are as follows:

L1i-cache size	Latency	IPC				
		prime.asm	fibonacci.asm	palindrome.asm	even-odd.asm	descending.asm
16B	1 cycle	0.0237	0.0229	0.0231	0.0224	0.0229
128B	2 cycles	0.0502	0.0586	0.0695	0.0215	0.0884
512B	3 cycles	0.0457	0.0531	0.0611	0.0208	0.078
1024B	4 cycles	0.0421	0.0485	0.0545	0.0201	0.0697

Table 1: IPC Values when L1d = 1kB

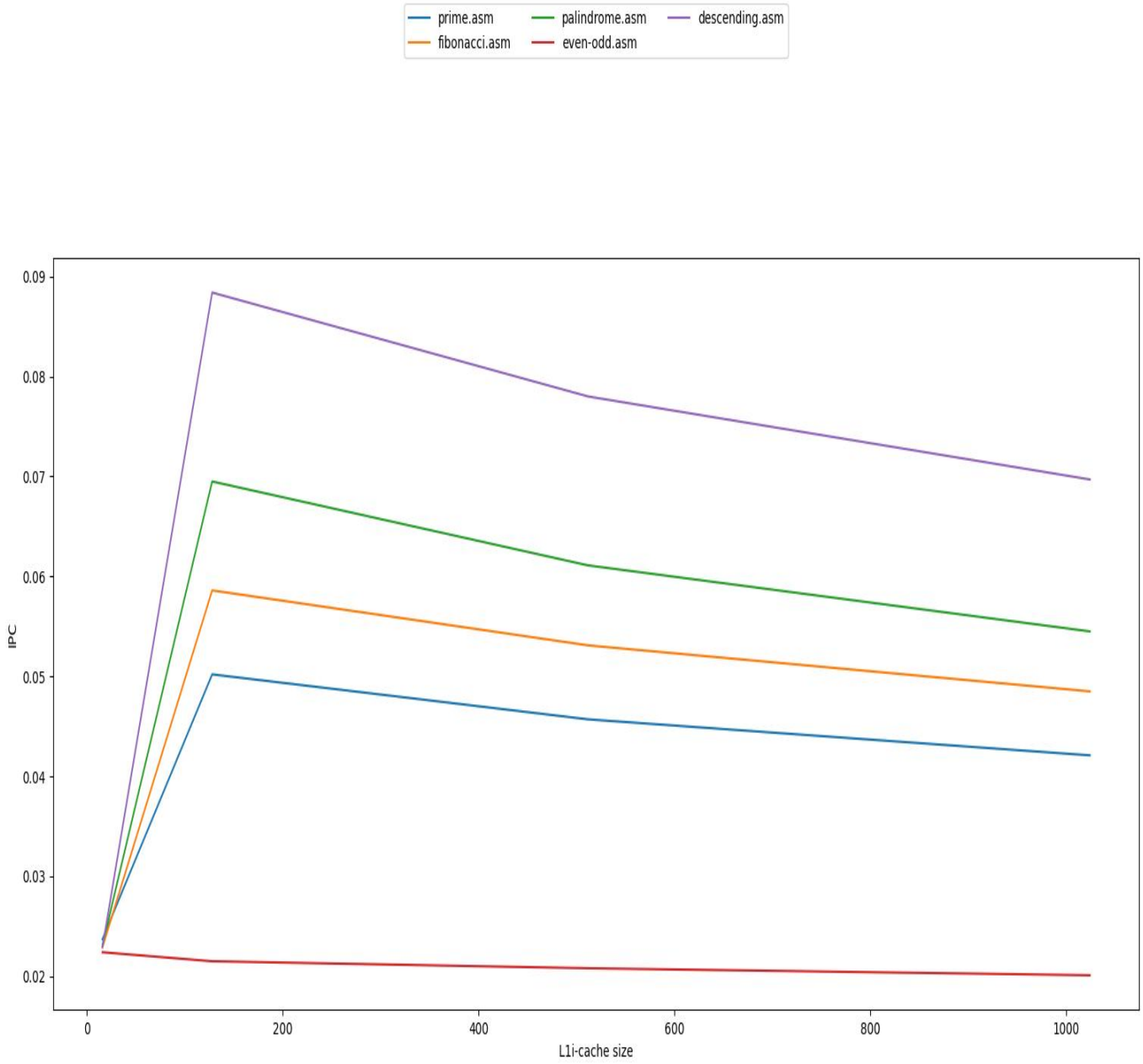

Figure 1: IPC values when L1d = 1kB

2.2 Effect of size of L1d-cache (for a fixed size (1kB) of L1i-cache) on IPC:

The IPC Values that we get for having L1d = 1kB are as follows:

L1d-cache size	Latency	IPC				
		prime.asm	fibonacci.asm	palindrome.asm	even-odd.asm	descending.asm
16B	1 cycle	0.0421	0.0491	0.0545	0.0201	0.0624
128B	2 cycles	0.0421	0.0488	0.0545	0.0201	0.0707
512B	3 cycles	0.0421	0.0486	0.0545	0.0201	0.0702
1024B	4 cycles	0.0421	0.0485	0.0545	0.0201	0.0697

Table 2: IPC Values when L1i = 1kB

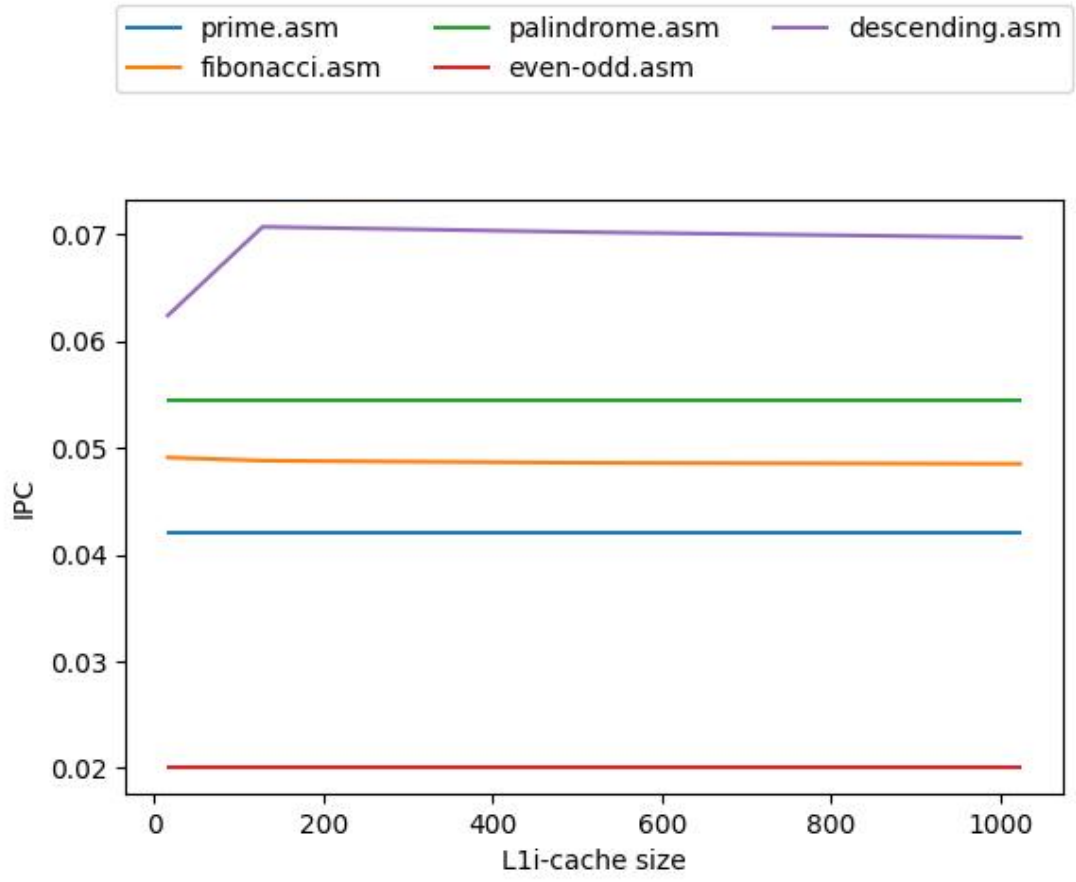

Figure 2: IPC values when L1i = 1kB

2.3 Correlate the nature of the benchmark to the observed plot

- From Figure 1 we can infer how IPC value changes as the L1i cache size increases from 16B to 1024B (1kB) and the L1d value is fixed at 1kB. We observe that as cache size increases, then latency also increases and number of accesses to memory and instructions remain the same, thus the IPC falls down after rising initially. While in all other cases we observe the IPC improves and peaks around 128B and then decreases, as beyond 128B we don't need more cache to store the instructions as and increase in cache size results only increase in latency which decreases the IPC value.
- From Figure 2 we can infer how IPC value changes as the L1d cache size increases from 16B to 1024B (1kB) and the L1i value is fixed at 1kB. We see that the IPC value doesn't change much as our programs don't have much memory accesses in them and we are starting from 16B cache size. So the given programs aren't having much improvement in the performance. IPC values of febonacci.asm and descending.asm programs change slightly.

2.4 toy-benchmark showing significant performance improvement when the L1i-cache is increased from 4B to 128B.

We have chose benchmark program as descending.asm, with L1d cache= 1kB.
For **descending.asm:**

At L1i=16B:
IPC value is: 0.0229
Number of Cycles taken: 15913
Number of Instructions: 365

At L1i=128B:
IPC value is: 0.0884
Number of Cycles taken: 4125
Number of Instructions: 365

We see that the IPC increased from 0.0229 to 0.0884 when L1i-cache is changed from 16B to 128B for descending.asm.We can clearly see that number of cycles decreased drastically.

2.5 Toy-benchmark showing significant performance improvement when the L1d-cache is increased from 32B to 128B.

We have chose benchmark program as descending.asm, with L1i cache= 1kB.
For **descending.asm:**

At L1d=16B:
IPC value is: 0.0624
Number of Cycles taken: 5845
Number of Instructions: 365

At L1d=128B:
IPC value is: 0.0707
Number of Cycles taken: 5123
Number of Instructions: 365

We see that the IPC increased from 0.0624 to 0.0707 when L1i-cache is changed from 16B to 128B for descending.asm. We can clearly see that number of cycles decreased drastically.