CS2160 Computer Organisation

Report 7

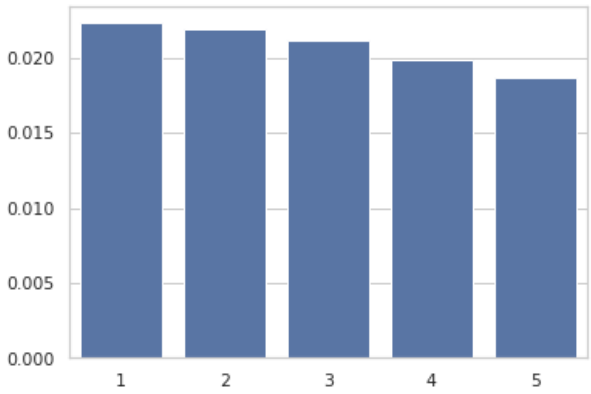
Team Members
Ishwar Govind 111901024
Abraham Mathew Illickan 111901003

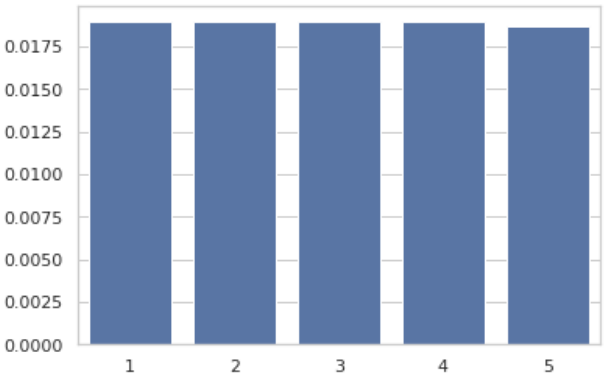
For each program we have plotted the I.P.C. (Instructions per Cycle) vs Cache size/latency. First, we have fixed the L1d Cache (1KB (256 Lines) and 12 cycles-latency) and varied the size of L1i cache. Then we have fixed the L1i Cache (1KB (256 Lines) and 12 cycles-latency) and varied the size of L1d cache.

All the .asm files and. stats are located in src/objectfiles.

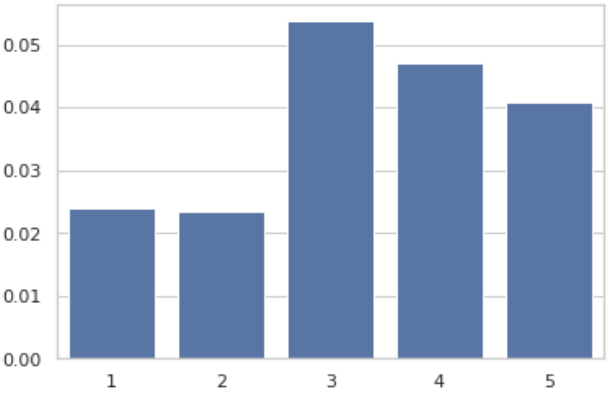
The left graph is for varying L1i with fixed L1d (1kB,12 cycles) and right graph is varying L1d with fixed (1kB,12 cycles) L1i.
The {1,2,3,4,5} in x-axis is
{(4B,1 cycle), (8B,2 cycles), (32B,4 cycles), (128B,8 cycles), (1kB,12 cycles)}

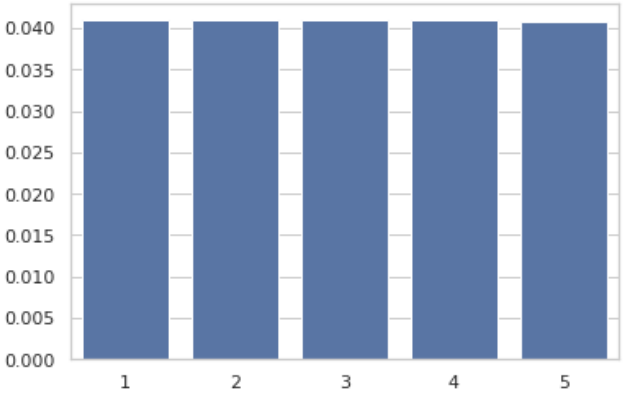
Even-Odd



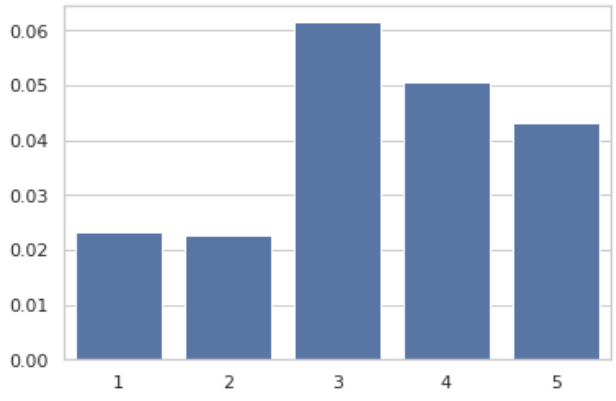


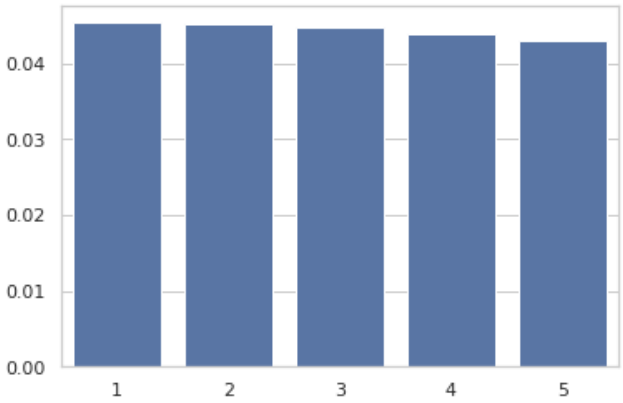
Palindrome



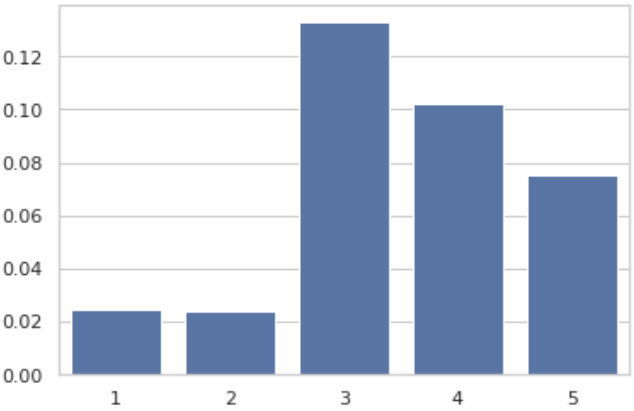


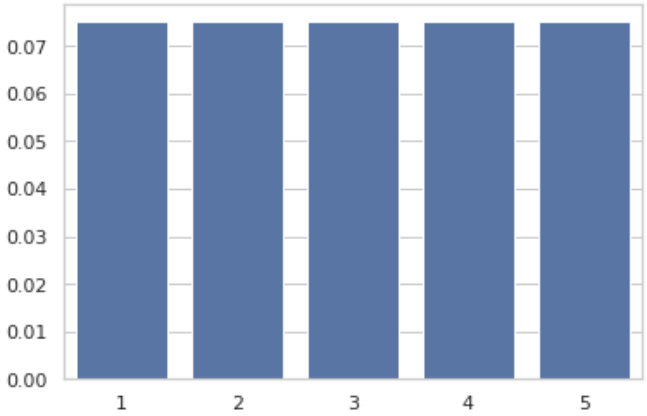
Fibonacci



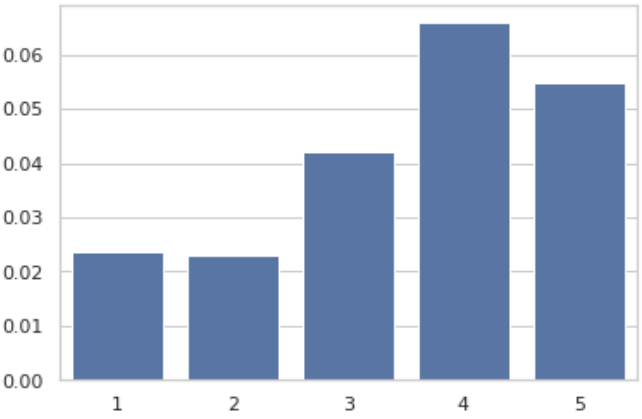


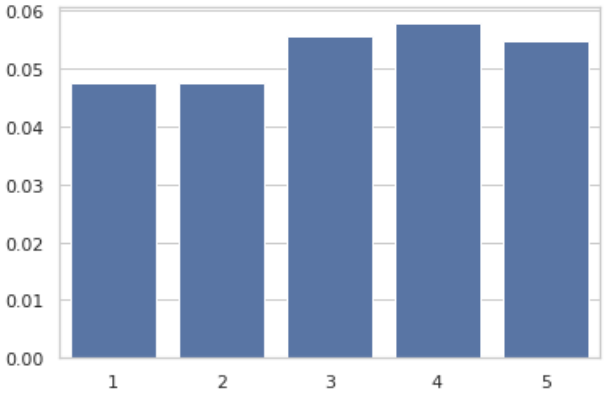
Prime





Descending





Observation

Even-odd

Instructions are not repeating in even-odd. So, there are no cache hits. IPC decreases because of increased latency.

There is only 1 memory access in even-odd program. So it doesn't make any difference as L1d cache.

Palindrome

There is a loop of length 8 in this program. It shows maximum performance when L1i cache is 8 lines long. Performance decreases after this because of increased latency

Again, there is only 1 memory access in this program and it is running in parallel with the instruction fetch of some other instruction. So it doesn't make any difference as L1d cache changes.

Fibonacci

As is the case in Palindrome, there is a loop of length 8 in this program. It shows maximum performance when L1i cache is 8 lines long. Performance decreases after this because of increased latency

Again, there is only 1 memory access in this program but here it is not running in parallel with the instruction fetch of some other instruction. So it actually affects the performance and IPC decreases as L1d cache size/latency increases.

Prime

As in the previous 2 cases, there is a loop of length 8 in this program. It shows maximum performance when L1i cache is 8 lines long. Performance decreases after this because of increased latency

Again, there is only 1 memory access in this program and it is running in parallel with the instruction fetch of some other instruction. So it doesn't make any difference as L1d cache changes.

Descending

In this case there is a loop that uses 17 different instructions. It shows maximum performance when L1i cache is 32 lines long. After this it drops off due to increased latency.

There are multiple load and store operations in this program. They access 9 different locations. So the performance peaks at 32 lines cache(smallest cache larger than 9 lines).