Computer Architecture Assignment 6

Srihari K G 210030035
Aryan Gulhane 210010006

October 27, 2023

1 Simulation using Cache system

Abstract: In this assignment, we simulate the pipelined processor with data and instructions read from caches and main memory with corresponding latenices. We present the throughput in terms of Instructions per cycle (IPC) vs cache size for the benchmark programs.

2 IPC vs L1i cache size for fixed L1D size and latency

2.1 Even or Odd

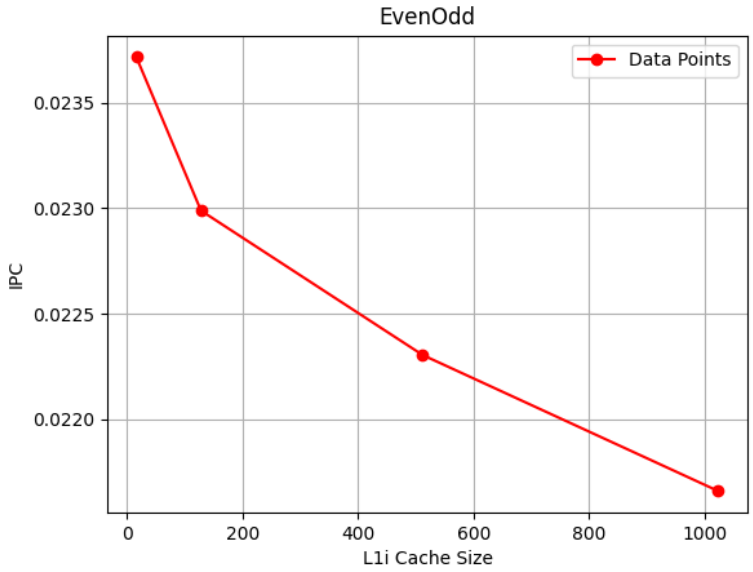

Figure 1: Even or Odd

2.2 Prime

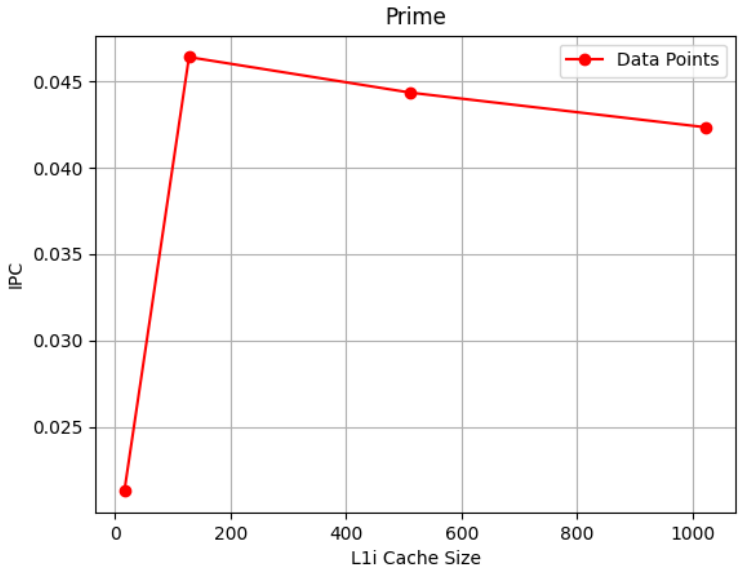

Figure 2: Prime

2.3 Palindrome

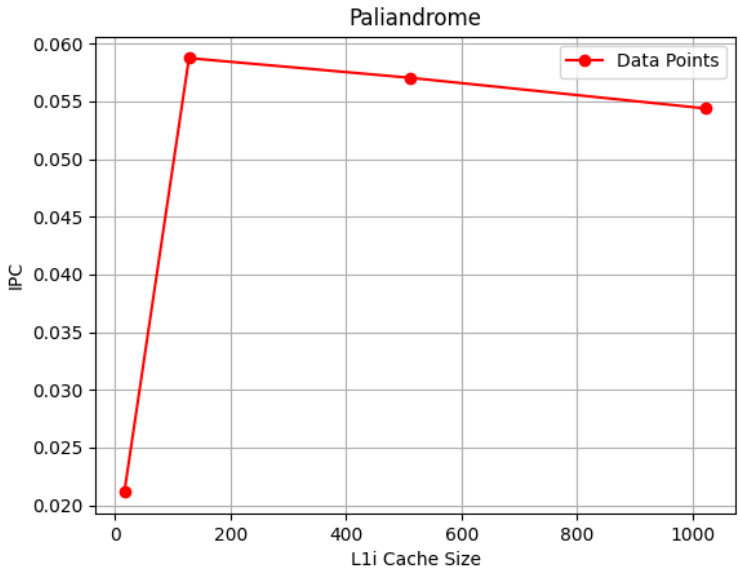

Figure 3: Palindrome

2.4 Fibonacci

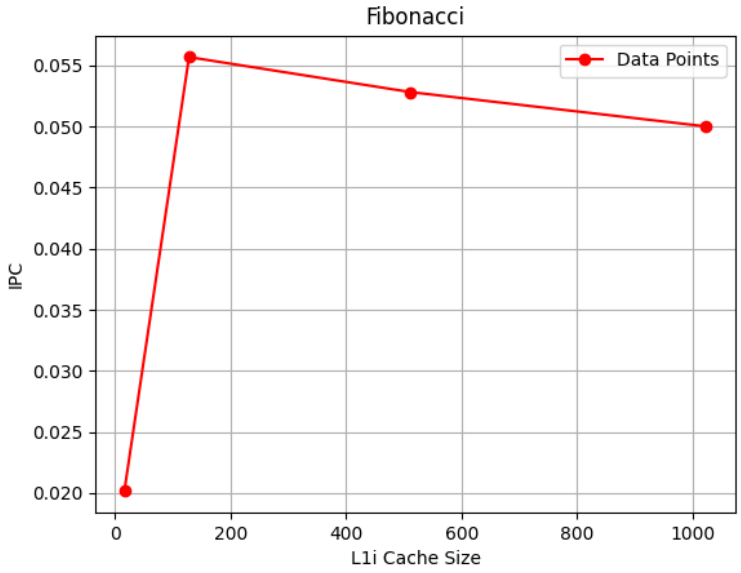

Figure 4: Fibonacci

2.5 Descending

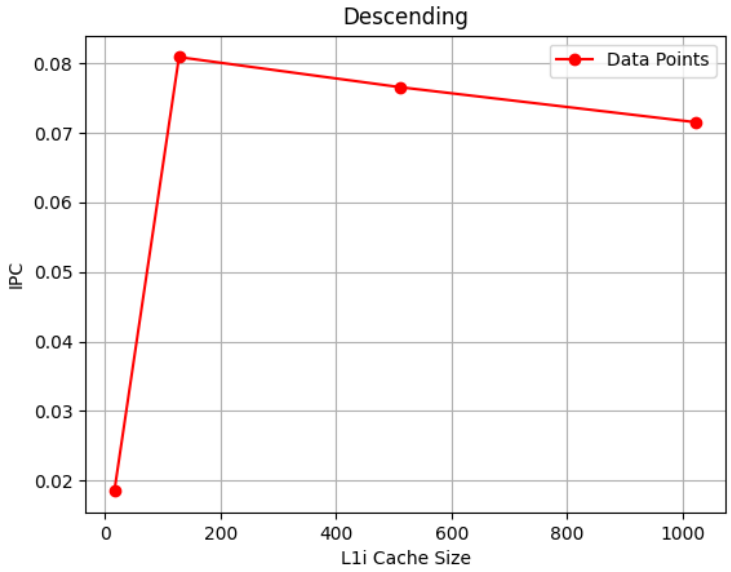

Figure 5: Descending

3 IPC vs L1D cache size for fixed L1i size and latency

3.1 Even or Odd

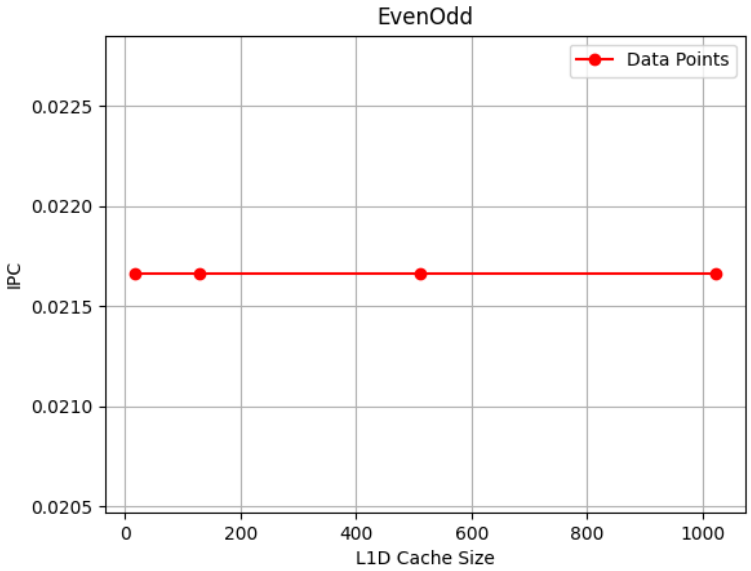

Figure 6: Even or Odd

3.2 Prime

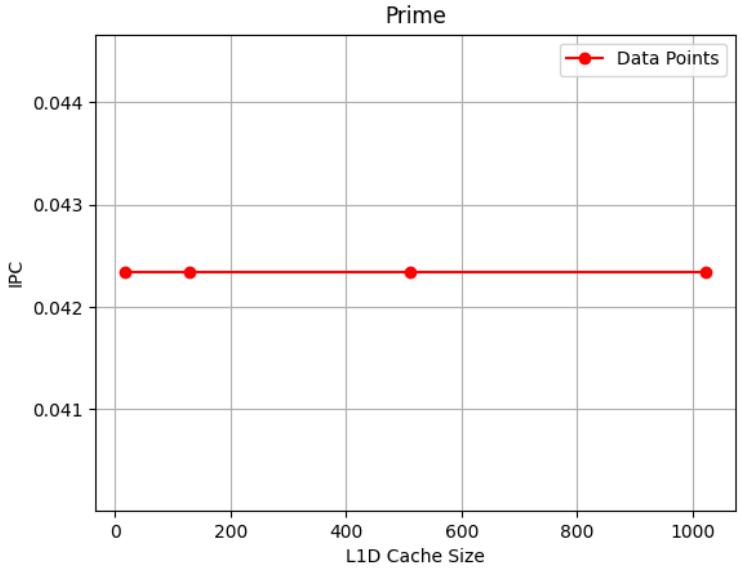

Figure 7: Prime

3.3 Palindrome

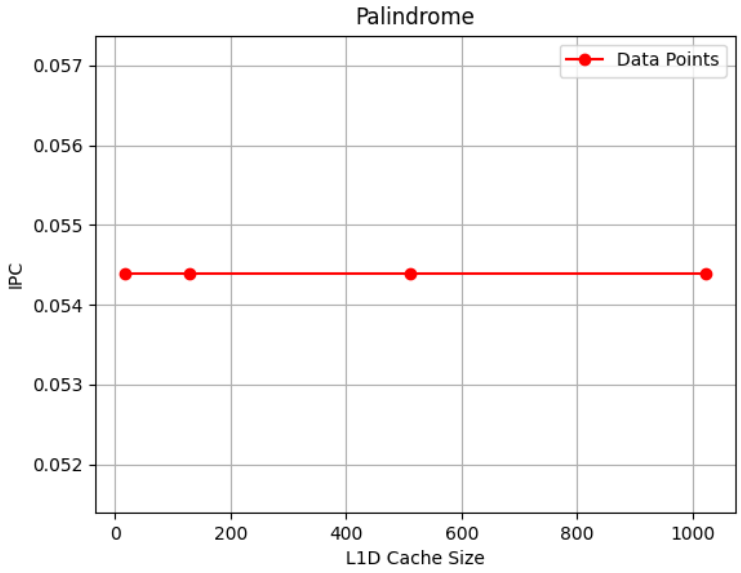

Figure 8: Palindrome

3.4 Fibonacci

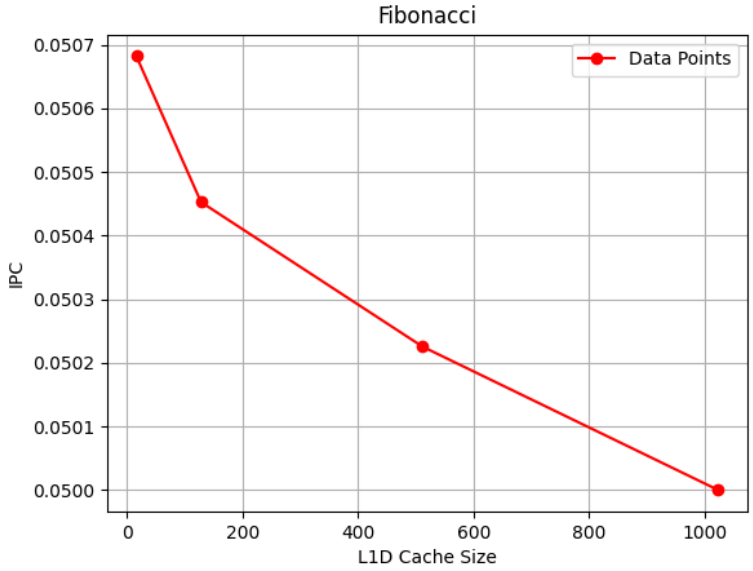

Figure 9: Fibonacci

3.5 Descending

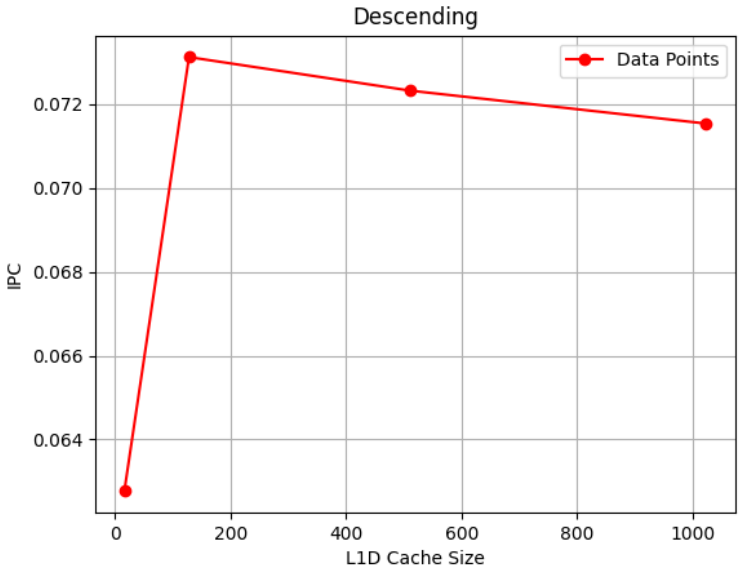

Figure 10: Descending

4 Cache size analysis

1. For fixed L1i and varying L1D we see that for evenodd , prime and palindrome benchmarks throughput remains the same because there are very few memory read/write involved.
2. However in descending and palindrome , extensively memories are accessed so there is varied throughput
3. In even odd program , since there is only 1 memory read instruction so irrespective of cache size we can fit the instructions and the data so increasing cache size simply decreases the throughput.
4. In descending we see the advantage of temporal locality and since the numbers are very less all these numbers can fit in the cache , when the cache size is increased , latency is increased so IPC decreases.
5. We also observe that except for even odd all benchmarks have higher throughput when L1i cache size is 128B.

5 Benchmark program for L1i Cache

This assembly code shows the performance improvement when L1i cache size is increased from 16B to 128B keeping L1D as 1 kilobytes.

```
    .data
a:
    10
    .text
main:
    sub %x19, %x19, %x19
    addi %x19, 10, %x19
loop:
    addi %x23, 1, %x23
    addi %x23, 2, %x23
    addi %x23, 3, %x23
    addi %x23, 4, %x23
    addi %x23, 3, %x23
    addi %x23, 4, %x23
    addi %x23, 5, %x23
    addi %x23, 6, %x23
    subi %x19, 1, %x19
    bgt %x19, %x0, loop
end
```

Comparison

Parameters	L1i Cache Size	
	16B	128B
No. of Instructions	103	103
No. of Cycles	4678	911
IPC	0.022	0.113

Table 1: Performance Metrics

6 Benchmark program for L1D Cache

This assembly code shows the performance improvement when L1D cache size is increased from 16B to 128B keeping L1i as 1 kilobytes.

```
    .data
a:
    1
    2
    3
    4
    5
    6
    7
    .text
main:
    sub %x12, %x12, %x12
    sub %x3, %x3, %x3
    addi %x3, 10, %x3
lol:
    load %x12, $a, %x12
    load %x12, $a, %x12
    load %x12, $a, %x12
    load %x12, $a, %x12
    load %x12, $a, %x12
    load %x12, $a, %x12
    load %x12, $a, %x12
    sub %x12, %x12, %x12
    subi %x3, 1, %x3
    bgt %x3, %x0, lol
    end
```

Comparison

Parameters	L1D Cache Size	
	16B	128B
No. of Instructions	104	104
No. of Cycles	3645	1341
IPC	0.028	0.07

Table 2: Performance Metrics