**CS 311 – Computer Architecture Lab**
Assignment–6 Report

**Student 1:** A.N.S Dharmasastha
**Roll-No:** 210150002

**Student 2:** V. Yashwanth
**Roll-No:** 210010057

# 1 L1i-cache

The below table shows the performance (instructions per cycle) of different programs for 1KB fixed size of L1d-cache and varying size of L1i-cache from 16B to 1KB

| File Name | 16B | 128B | 512B | 1KB |
|---|---|---|---|---|
| descending.out | 0.030315 | 0.03102205 | 0.029892 | 0.02968504 |
| evenorodd.out | 0.026315 | 0.02631579 | 0.02631579 | 0.02631579 |
| fibonacci.out | 0.0285252 | 0.028403 | 0.02828175 | 0.02816153 |
| palindrome.out | 0.0262237 | 0.0261437 | 0.0260642 | 0.02598527 |
| prime.out | 0.028421 | 0.02831 | 0.028210 | 0.028106509 |

Table 1: statistics for L1i-cache

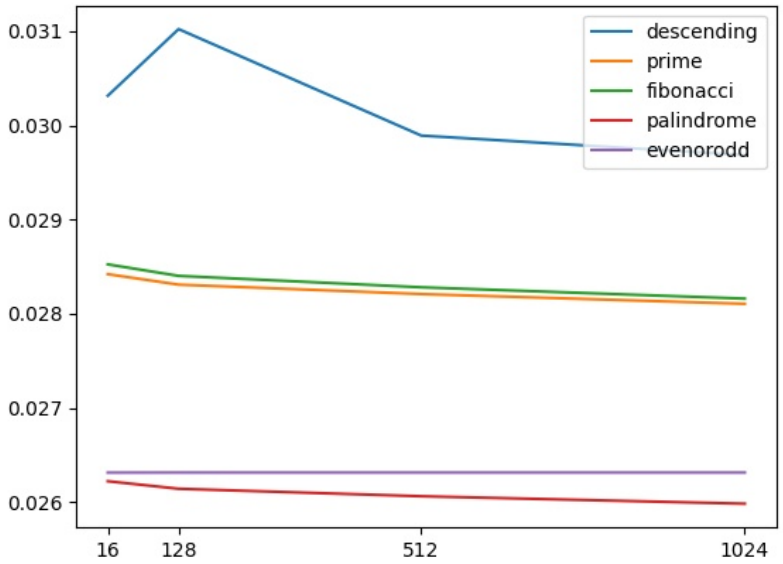


Figure 1: Caption

# 2 L1d-cache

The below table shows the performance (instructions per cycle) of different programs for 1KB fixed size of L1i-cache and varying size of L1d-cache from 16B to 1KB

| File Name | 16B | 128B | 512B | 1KB |
|---|---|---|---|---|
| descending.out | 0.030068591 | 0.029939644 | 0.029811798 | 0.02968503 |
| evenorodd.out | 0.0265252 | 0.026455026 | 0.026385223 | 0.02631579 |
| fibonacci.out | 0.02816153 | 0.02816153 | 0.02816153 | 0.02816153 |
| palindrome.out | 0.02605297 | 0.02603037 | 0.026007803 | 0.025985274 |
| prime.out | 0.02816901 | 0.028148148 | 0.028127313 | 0.028106509 |

Table 2: statistics for L1d-cache

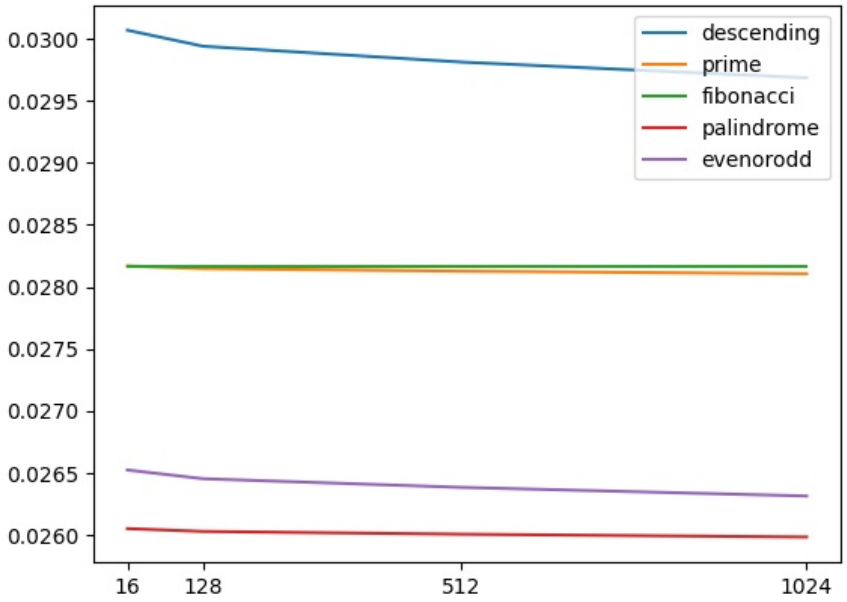


Figure 2: Caption

# 3 Observations

## 3.1

Initially, as Cache size increases, IPC increases because cache is used instead of memory and latency is less than memory. However, as we continue to increase cache size, IPC decreases because the peak required cache size has been reached, and as we continue to increase cache size, latency increases and IPC decreases.

## 3.2

We've picked descending.asm as our benchmark. When the size of L1-i cache is increased from 16B to 32b to 128B, the hit rate increases from 0.7082228 to 0.9811321 to 1. As cache size rises, the hit rate increases as predicted and eventually stabilises after the program has sufficient cache.

## 3.3

We've picked fibonacci.asm as our benchmark. When the size of L1-d cache is increased from 16B to 32B to 128B, the hit rate is constant at 1. Here, as the cache size grows, the hit rate stays constant since there are fewer memory accesses.