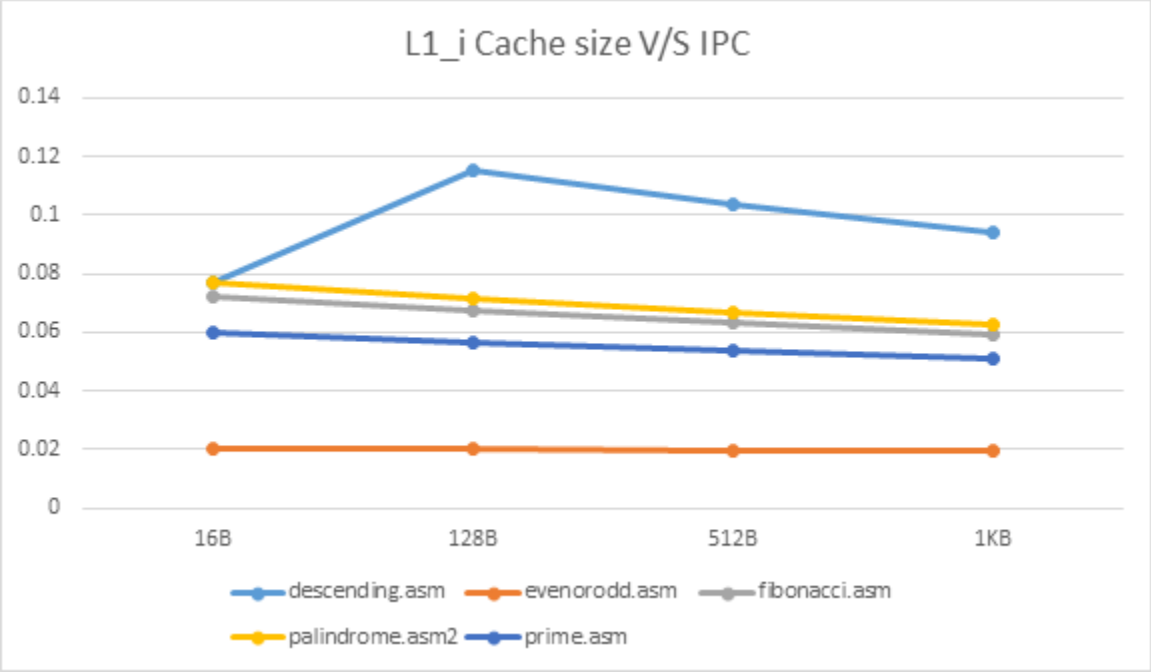
ASSIGNMENT 6 REPORT

TEJAL LADAGE 210010026
ASHWIN WAGHMARE 210010060

L1_d Cache size = 1KB

IPC					
L1_i cache size	Descending.asm	Evenorodd.asm	Fibonacci.asm	Palindrome.asm	Prime.asm
16B	0.07732924544370635	0.02046783625730994	0.07249190938511327	0.07701421800947868	0.060240963855421686
128B	0.11566878980891719	0.02005730659025788	0.0675920337960169	0.07150715071507151	0.056818181818181816
512B	0.10367663850194109	0.019662921348314606	0.06331260599208592	0.06673511293634497	0.053763440860215055
1KB	0.09393751293192634	0.01928374655647383	0.05954279638490165	0.06256015399422522	0.05102040816326531

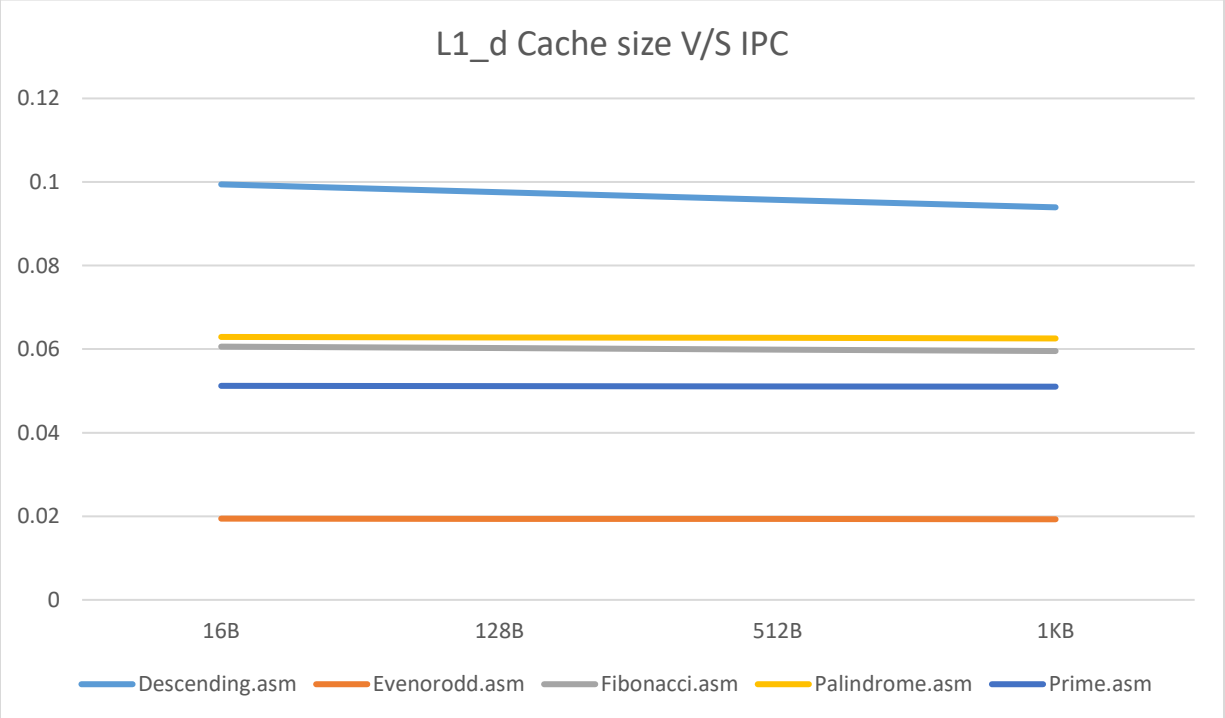


Analysis

It is observed that all the benchmarks achieve their highest IPC when the size of L1i cache is 128B. In the initial size of L1i cache (at 16B), IPC does not appreciably increase indicating that an increase in size of the cache is equally weighed out by a two□fold increase in the latency. But, as the cache size is increased to 128B, there are many cache hits and this outweighs the effect of latency. But, as we further increase cache size, since the earlier cache size was already sufficient to hold the instructions with a good hit rate, this increase has no appreciable effect rather the increase in latency now has an upper hand leading to a decrease in IPC. The effect of varying cache size is more profound on descending.asm as it has very much greater number of instructions when compared to other benchmarks. Also, evenorodd.asm shows a different linear decreasing trend as the number of dynamic instructions are very less for cache size to have any effect at all on its IPC. However, even then, latency dominates at higher cache sizes.

L1_i Cache size = 1KB

IPC					
L1_d cache size	Descending.asm	Evenorodd.asm	Fibonacci.asm	Palindrome.asm	Prime.asm
16B	0.0994305738063951	0.019444444444444445	0.06060606060606061	0.06292352371732816	0.05121638924455826
128B	0.09752953813104188	0.019390581717451522	0.06024744486282948	0.06280193236714976	0.05115089514066496
512B	0.09569983136593592	0.019337016574585635	0.059893048128342244	0.0626808100289296	0.05108556832694764
1KB	0.09393751293192634	0.01928374655647383	0.05954279638490165	0.06256015399422522	0.05102040816326531



Analysis

It is observed that as the number of data memory accesses is quite low, varying the cache size does not have a significant effect on all other benchmarks except for descending.asm, which again shows a similar trend as in the previous case and can explained exactly in the same terms.