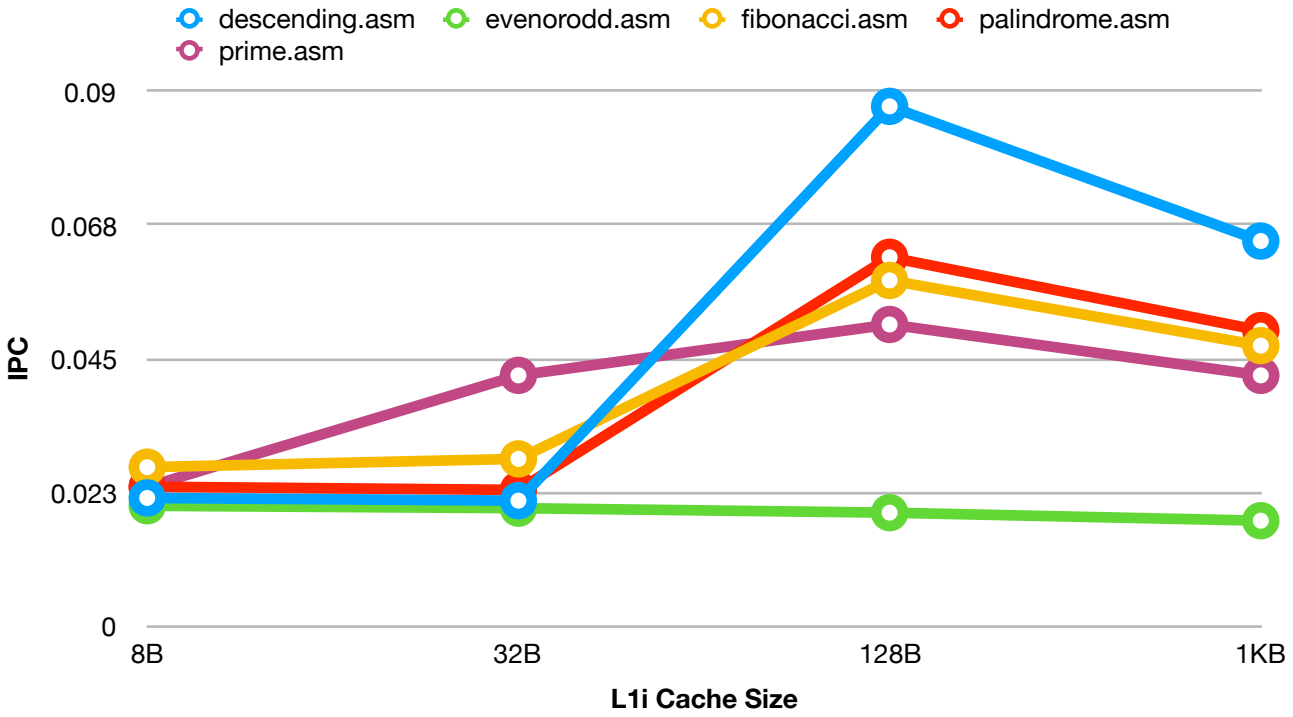
Assignment 6 Report

L1d Cache Size = 1KB

	IPC				
L1i Cache Size	descending.asm	evenorodd.asm	fibonacci.asm	palindrome.asm	prime.asm
8B	0.0215502919257607	0.0202312138728324	0.0267366913344474	0.0234487734487734	0.0234192037470726
32B	0.0210956739928442	0.0198300283286119	0.0281336347651344	0.0229115262601339	0.0421940928270042
128B	0.087492773174022	0.0190735694822888	0.0581818181818182	0.0620821394460363	0.050761421319797
1KB	0.0648108493932905	0.0177215189873418	0.0471976401179941	0.0497322111706197	0.0421940928270042

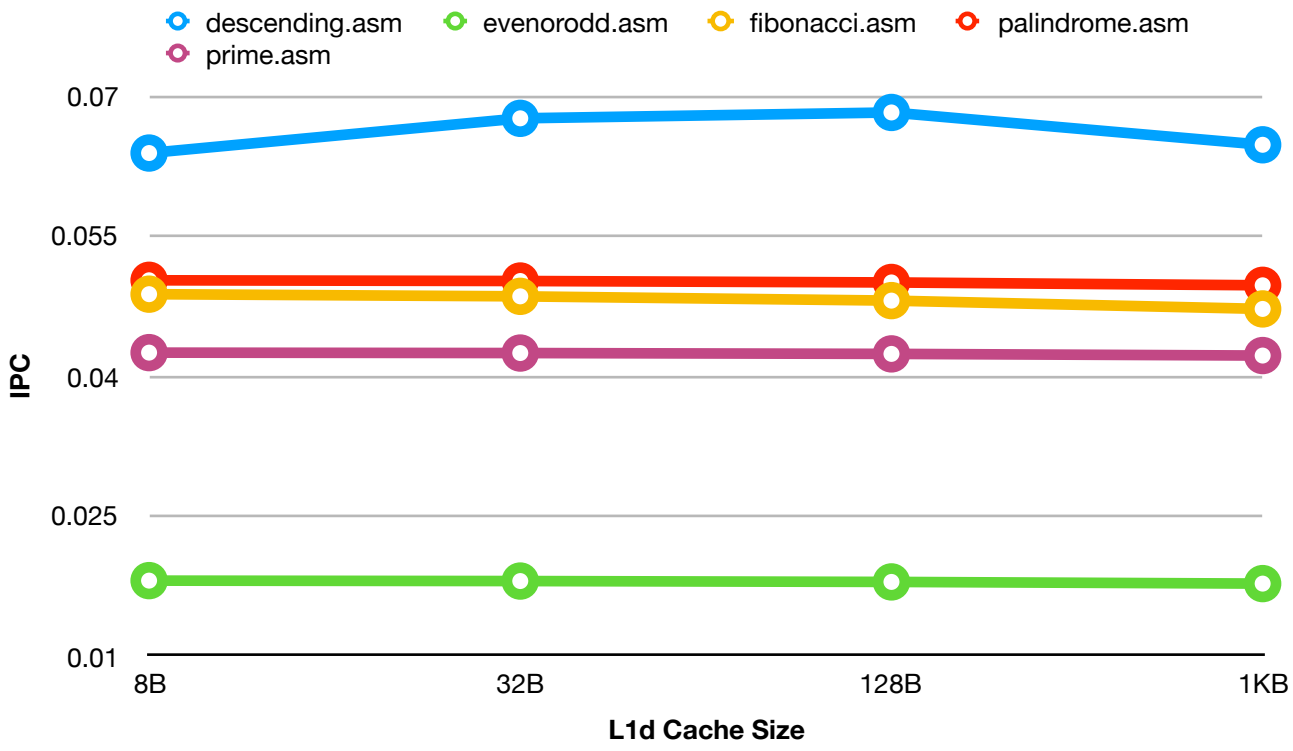


Analysis

It is observed that all the benchmarks achieve their highest IPC when the size of L1i cache is 128B. In the initial sizes of L1i cache (at 8B and 32B), 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.

L1i Cache Size = 1KB

	IPC				
L1d Cache Size	descending.asm	evenorodd.asm	fibonacci.asm	palindrome.asm	prime.asm
8B	0.0639256547451422	0.0180412371134021	0.0487804878048781	0.0502706883217324	0.0425079702444208
32B	0.0676501266577261	0.0179948586118252	0.0485478977026441	0.0501930501930502	0.0424628450106157
128B	0.0682809445029328	0.0179028132992327	0.0480893087161872	0.0500384911470362	0.0423728813559322
1KB	0.0648108493932905	0.0177215189873418	0.0471976401179941	0.0497322111706197	0.0421940928270042



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.