ReadMe: Project 2 CSE 420

* The objective of this project is to get familiar with Gem5 and implement different configurations of caches on different benchmarks.
* We used Gem5 to simulate ARM processor.
* **Benchmarks used:** used benchmarks from the MiBench Suite, networking benchmarks like dijkstras and other benchmarks like FFT, qsort, basicmath.
* How to change the cache configurations while running gem5?  
  We used command line to make changes to the cache architecture configuration.  
    
  Eg: ./build/ARM/gem5.opt ./configs/example/se.py --cpu-type=TimingSimpleCPU --caches --l1d\_size=1kB --l1d\_assoc=1 --l2\_size=16kB --l2\_assoc=16 --l1i\_size=2kB --cacheline\_size=16 --l1i\_assoc=1 --l2cache --num-l2caches=1 -c./benchmarks/dijkstra/dijkstra\_small -o./benchmarks/dijkstra/input.dat

1. Impact of Cache Size on Execution Time and Miss Rate  
   Although increasing cache size can result in potentially longer hit time andhigher cost and power, it also allows the more data to fit in the cache i.e.more spatial and temporal locality to explore. So, miss rate goes down andperformance increases.
2. Impact of Cache Associativity on Execution Time and Miss Rate  
   Increasing associativity obviously reduces conflict misses but at the cost of increased hit time and power consumption goes up. It may increase even miss penalty.
3. Impact of Block Size on Miss Rate  
   Larger blocks help exploiting spatial locality, reducing miss rate. They reduce the compulsory miss but also increase the miss penalty. If the block size becomes significant fraction of the cache size (often in smaller caches), the number of blocks that can be held in the cache reduces and hence, the capacity or conflict misses increase with the increasing block size.  
   Choosing the right block size is always a trade-off that depends on the size of the cache vs. miss penalty and usually the block size is kept at 32 or 64 bytes.
4. Performance vs. Energy Cost Trade-off