L1 Cache Block Size Simulation - Report

Note: I runned Gem5 on ***Google Colab***, since Google Colab has all linux features such as bash scripts, gcc and etc.

1. Before start to simulate Gem5, I install below packages.

```
!sudo apt install build-essential git m4 scons zlib1g zlib1g-dev
libprotobuf-dev protobuf-compiler libprotoc-dev
libgoogle-perftools-dev python-dev python
```

2. After installing missing packages above, I clone below git repository.

```
!git clone https://gem5.googlesource.com/public/gem5
```

3. Then, I build my first Gem5 with below command-line. This part takes the longest time in the homework.

```
%cd gem5
!scons build/X86/gem5.opt -j9
```

1- Which files in the simulator you mainly changed?

Note: In first time, I got normal output from ***"matmul.c"*** but there was a problem about importing. Therefore, I got a static output from ***"matmul.c"*** as ***"matmul.x86"***. Then, I gave to Gem5.

```
!gcc -static  -o matmul.x86 matmul.c
```

- I mainly change the ***"se.py"*** in folder which is "`configs/example/se.py`".

```
!build/X86/gem5.opt configs/example/se.py
--cmd=tests/test-progs/hello/bin/x86/linux/matmul.x86
--cpu-type=TimingSimpleCPU --caches --l1d_size=64kB --l1i_size=64kB
--cacheline_size=...
```

2- What is the main code you added into the simulator (not all the lines, just your main change is enough)

- Firstly, I move the ***"matmul.x86"*** file to the folder which is "`tests/test-progs/hello/bin/x86/linux/matmul.x86`" and I change below path in below command-line in ***"cmd=file_path".***
- Secondly, I change the below command-line to change L1 cache size in ***"--l1d=... --l1i=...".***
- Finally, I change the below command-line to change cache block size in ***"cacheline_size=...".***

```
!build/X86/gem5.opt configs/example/se.py
--cmd=tests/test-progs/hello/bin/x86/linux/matmul.x86
--cpu-type=TimingSimpleCPU --caches --l1d_size=64kB --l1i_size=64kB
--cacheline_size=...
```

3- Tables showing the L1 hit rate for different block sizes for both data and instruction caches

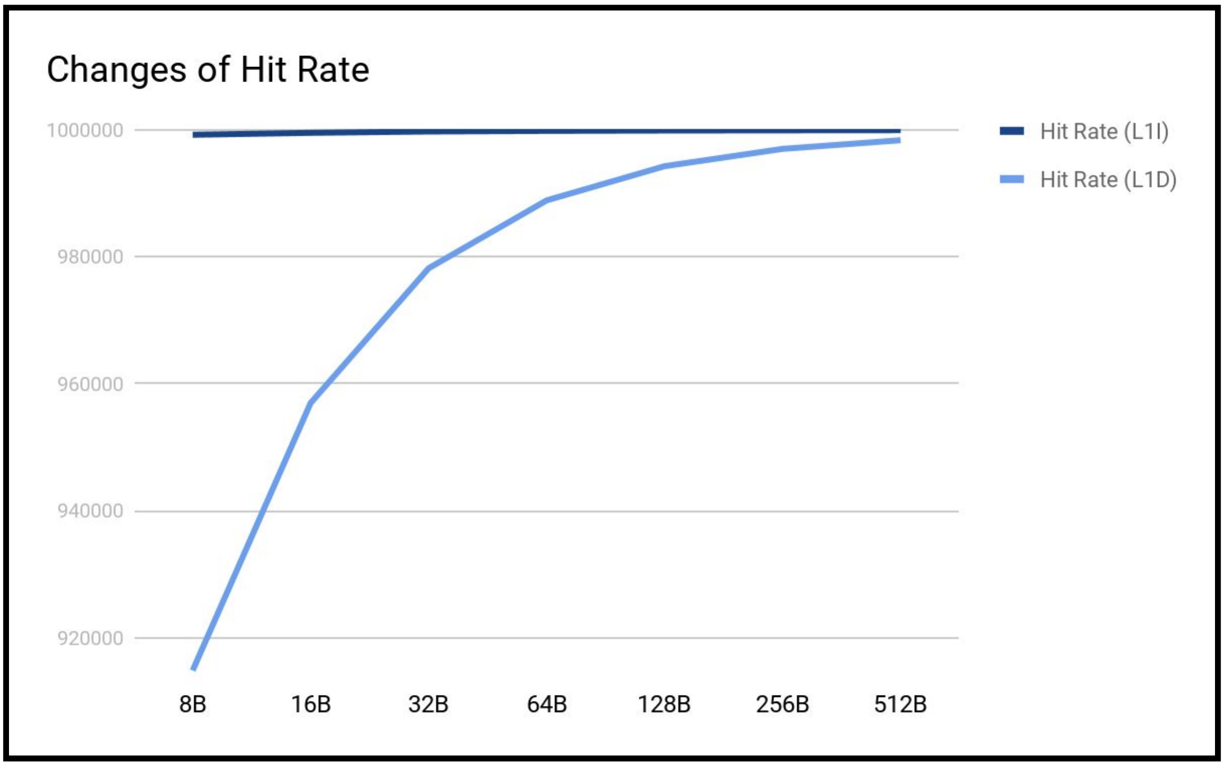
Note: I found hit rate by $(1 - \mathit{system.cpu.icache.overall\_miss\_rate} :: \mathit{total})$ for L1I and $(1 - \mathit{system.cpu.dcache.overall\_miss\_rate} :: \mathit{total})$ for L1D from **"stats.txt"** in **"m5out"** folder.

Note: When I run the code *for 1B and 2B*, I got error like "`fatal: Block size must be at least 4 and a power of 2`" and *for 4B*, I got again error like below:

- "`BaseCache::satisfyRequest(PacketPtr, CacheBlk*, bool, bool): Assertion `pkt->getOffset(blkSize) + pkt->getSize() <= blkSize' failed.`"

L1 Cache Size	Hit Rate (L1I)	Hit Rate (L1D)
8B	0.999231	0.914889
16B	0.999535	0.957005
32B	0.999747	0.978211
64B	0.999842	0.988898
128B	0.999898	0.994283
256B	0.999930	0.997010
512B	0.999956	0.99837

4- Graphs showing the changes in the hit rate according to the block size for both data and instruction caches.


Changes of Hit Rate
8B
16B
32B
64B
128B
256B
512B
Hit Rate (L1I)
Hit Rate (L1D)