**Memory Hierarchies Exploration using Gem5.**

Sanjay Subedi

University of the Cumberlands

Computer Architecture and Design (MSCS-531-M50)

Dr. Cooper

Oct 3, 2025

**Environment Setup:**

* ARM ISA is being used to run Gem5.
* All the dependencies required, such as Python , Scons, and Clang compiler, were installed.
* The Gem5 was cloned using the code ***git clone*** [***https://github.com/gem5/gem5***](https://github.com/gem5/gem5)
* The following command was used to enter the gem5 directory

*cd gem5*

**Benchmark:**

* A "Hello World" program was used as the benchmark. The binary is located at tests/test-progs/hello/bin/x86/linux/hello.

Simulation of Cache Performance:

**Default Cache Configuration:**

The following file(s) were ran to see which is better to use as default cache configuration:

Using se.py:

**./build/X86/gem5.opt configs/deprecated/example/se.py --cmd=tests/test-progs/hello/bin/x86/linux/hello**

**A screen shot of a computer screen

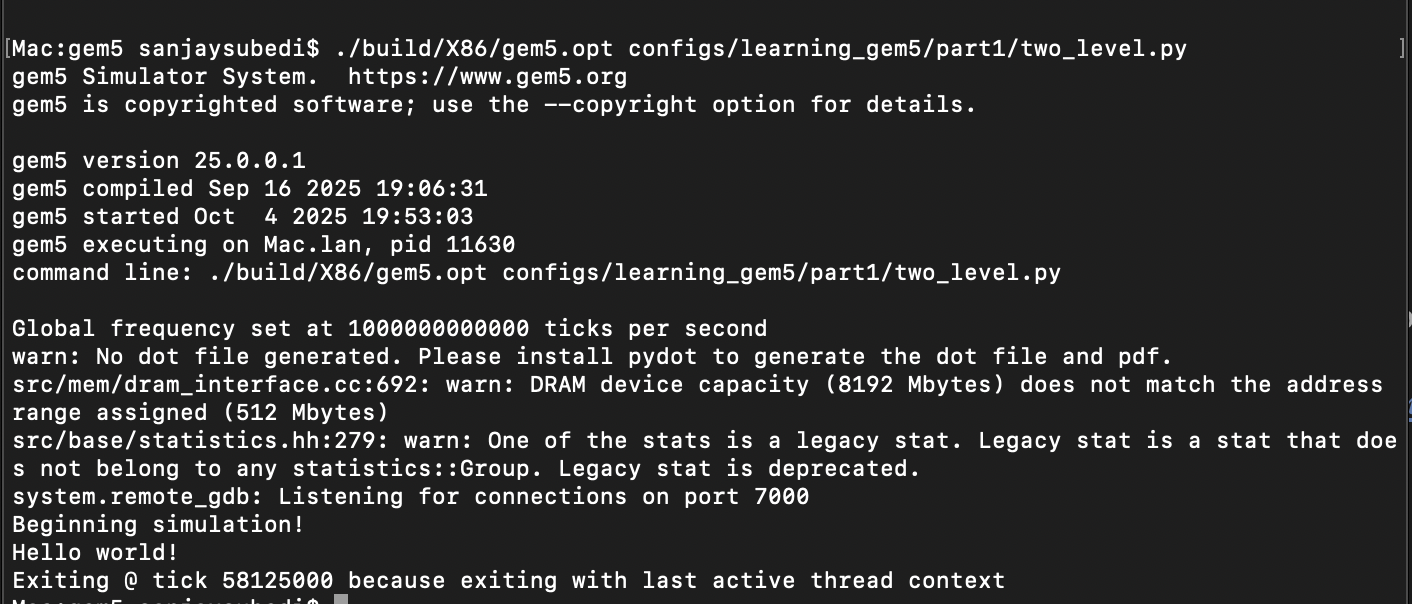
AI-generated content may be incorrect.**

The exiting tick was 5943000.

**Using two\_level.py:**

To run the simulation of default cache configuration, the two\_level.py was run using the code :

*./build/X86/gem5.opt configs/learning\_gem5/part1/two\_level.py*

**

After running the two\_level.py, for detail information the code : *less m5out/stats.txt* was used to see the files.

A screenshot of a computer screen

AI-generated content may be incorrect.

On the default cache configuration, the number of ticks was 58125000.

Dcache hits were 1944, misses were 136.

Icache hit rate were 7836 and misses were 235.

L2 cache hit rate was 6 and misses were 229.

Miss rate = 2.9%

Hit rate = 97.1 %

A screenshot of a computer program

AI-generated content may be incorrect.

**Using se.py , different cache configs were done:**

***Cache configuration 1:***

The size being used was: 64 kb for Icache, Dcache

Associativity: 4 – way and block size

The following code as used to pass the sizes and run the simulation:

*build/X86/gem5.opt configs/deprecated/example/se.py --cpu-type=DerivO3CPU --caches --l1d\_size=64kB --l1i\_size=64kB --l1d\_assoc=4 --l1i\_assoc=4 --cmd=tests/test-progs/hello/bin/x86/linux/hello*

A screenshot of a computer program

AI-generated content may be incorrect.

*cat m5out/stats.txt* was used to see the results of simulation results.

D-Cache (Data Cache)

* Hit Rate: 92.4%
* Miss Rate: 7.6%
* AMAT: 4719 ticks

I-Cache (Instruction Cache)

* Hit Rate:  81.6%
* Miss Rate: 18.4%
* AMAT: 9818 ticks

Cache configuration 2:

The size being used was: 32 kb for Icache, Dcache .

Associativity: 4– way

The following command was used to run cache config 2.

*build/X86/gem5.opt configs/example/se.py --cpu-type=DerivO3CPU --caches --l1d\_size=32kB --l1i\_size=32kB --l1d\_assoc=4 --l1i\_assoc=4 --cmd=tests/test-progs/hello/bin/x86/linux/hello*

**

D-Cache (Data Cache)

* Hit Rate: 92.4%
* Miss Rate: 7.6%
* AMAT: 5005 ticks

I-Cache (Instruction Cache)

* Hit Rate:  81.6%
* Miss Rate: 18.4%
* AMAT: 10943 ticks

**Cache configuration 3:**

The size was kept the same as config II, but the Associativity for both caches was changed to 8-

way.

The following code was used to run the config:

*build/X86/gem5.opt configs/deprecated/example/se.py --cpu-type=DerivO3CPU --caches --l1d\_size=32kB --l1i\_size=32kB --l1d\_assoc=8 --l1i\_assoc=8 --cmd=tests/test-progs/hello/bin/x86/linux/hello*

A screen shot of a computer screen

AI-generated content may be incorrect.

D-Cache (Data Cache)

* Hit Rate: 92.4%
* Miss Rate: 7.6%
* AMAT: 4716 ticks

I-Cache (Instruction Cache)

* Hit Rate:  81.6%
* Miss Rate: 18.4%
* AMAT: 9819 ticks

**Results:**

Increasing Cache Size: A Larger cache retains a larger amount of data, which reduces the need for frequent evictions. Hence, it yields the most significant improvement.

Increasing block size: This offered the lowest change.

Increasing Associativity: Increasing associativity decreased the ticks, and decreasing the associativity increased the ticks.

**Troubleshooting:**

When using two\_level.py, it was very complicated to change the associativity of the L1 and L2 caches. So se.py was used to tinker with the cache configurations for ease of understanding.