Exploring Memory Technology Simulators

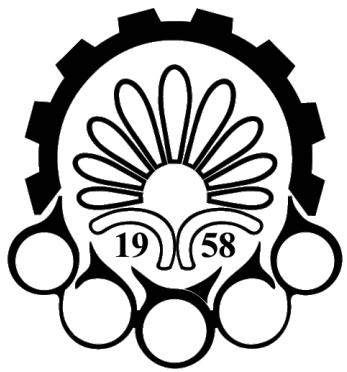
Reza Adinepour

Computer Engineering Department, Tehran Ploytechnic

adinepour@aut.ac.ir

June 27, 2024

Memory Technologies - Spring 2024



Amirkabir University of Technology
(Tehran Polytechnic)

Agenda

DRAMSIM

SimpleScaler

References

DRAMSIM

1. DRAMSim use for simulate Dynamic RAMs.
   - 1.1 DRAM modeling it's very important because the technology is trying to provide CPU and DRAM integrated in one chip.
   - 1.2 This provides high density:
     - 1.2.1 High density
     - 1.2.2 Optimal performance
     - 1.2.3 Lower power consumption
2. DRAMSim is provide in three version:
   - 2.1 DRAMSim 1
   - 2.2 DRAMSim 2
   - 2.3 DRANSim 3

     In this talk, we discuss about the last version of DRAMSim
3. DRAMSim developed in C++ and write in modularly.

DRAMSIM (Cont.)

1. DRAMSim can be connected to GEM5
2. DRAMSim can simulate following protocol:
   - 2.1 DDR3
   - 2.2 DDR4
   - 2.3 LPDDR3
   - 2.4 LPDDR4
   - 2.5 GDDR5
   - 2.6 GDDR6
   - 2.7 HBM
   - 2.8 HMC
   - 2.9 STT-MRAM

The structure of main block of DRAMSim is shown in figure 1.

DRAMSIM (Cont.)

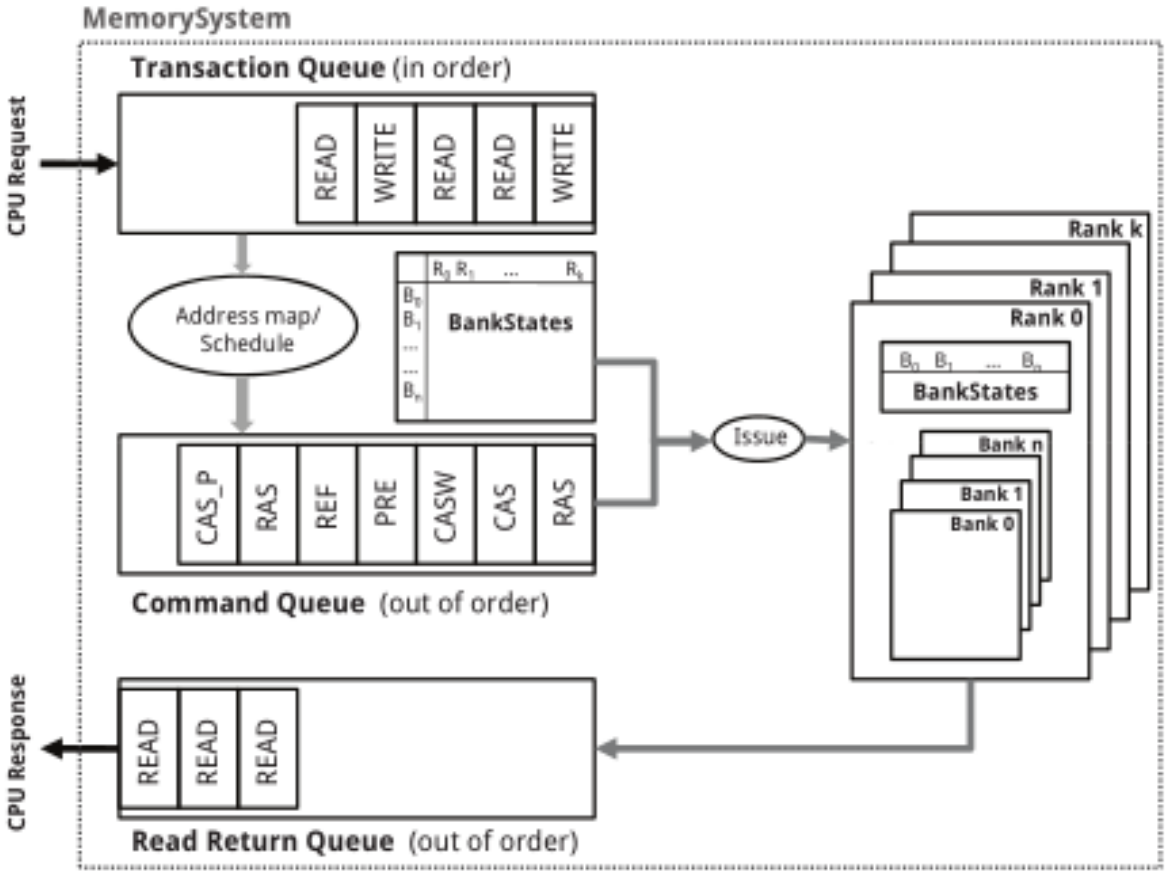


Figure: Main block of DRAMSim

DRAMSIM (Cont.)

Advantages:

1. The possibility of simulating new DRAM technologies like DDR4 and GDDR6
2. High flexibility in configuration
3. Synchronize with system simulators

Disadvantages:

1. Dependence on the model and configuration
2. Don't report power consumption and area

DRAMSIM (Cont.)

How install and build DRAMSim?

We should clone repository in first step:

Clone repository

```
$ git clone https://github.com/umd-memsys/DRAMsim3
$ DRAMsim3cd
```

now we should build it:

Build

```
$ mkdir build
$ cd buildcd
$ cmake ..
$ make -j4
$ -DTHERMAL=1.. cmake
```

DRAMSIM (Cont.)

If the simulation builds successfully, you can see **Built target** on your terminal like figure 2

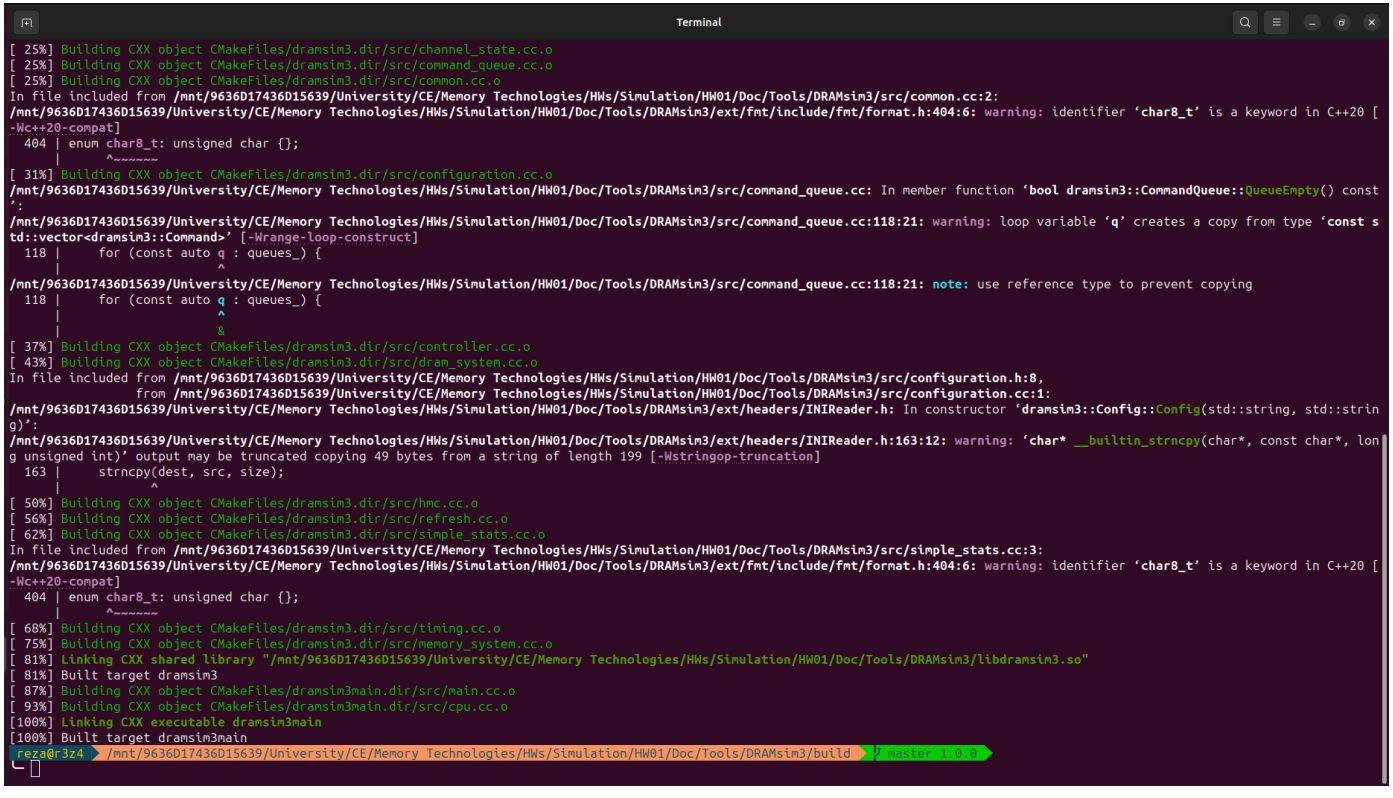


Figure: DRAMSim built target

DRAMSIM (Cont.)

How can run sample simulation?

1. in first, create a folder for save output file of simulation:

Create output directory

```
$ mkdir output
```

2. then, with this command, run simulation for `sample_trace.txt` config file:

Run simulation

```
$ ./build/dramsim3main configs/DDR4_8Gb_x8_3200.ini
-c 100000 -t tests/example_trace.txt -o output/
```

every various configurations files, located in `configs/` directory. for this simulation we use `DDR4_8Gb` config file.

DRAMSIM (Cont.)

after simulation is finished, you can see output in `output/` directory in `dramsim3.txt` file like bellow:

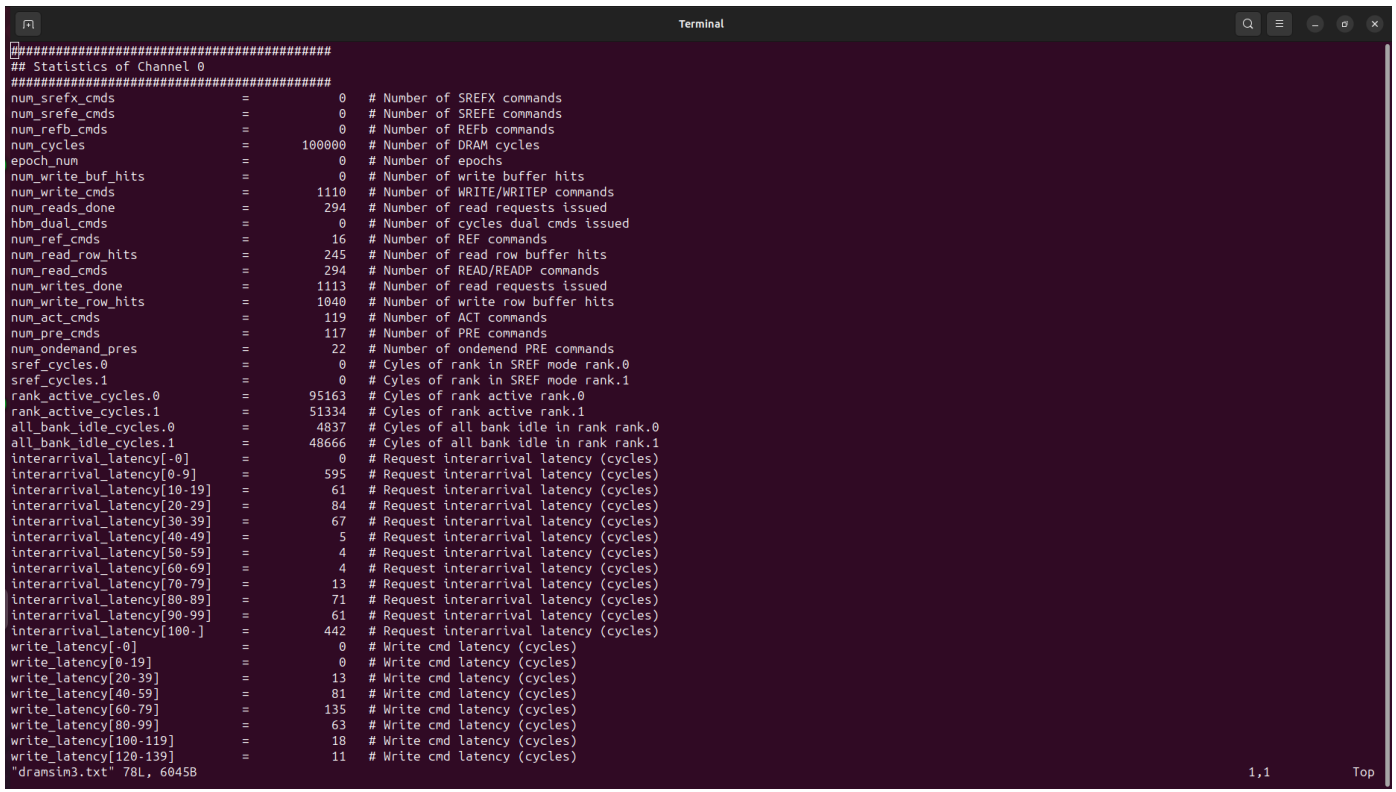


Figure: Output report

DRAMSIM (Cont.)

we can plot read latancy, interarrival latancy, write latancy and ... with some python scripts located in `script/` directory.

plot

```
$ python3 scripts/plot_stats.py output/dramsim3.json
```

the output of simulation:

DRAMSIM (Cont.)

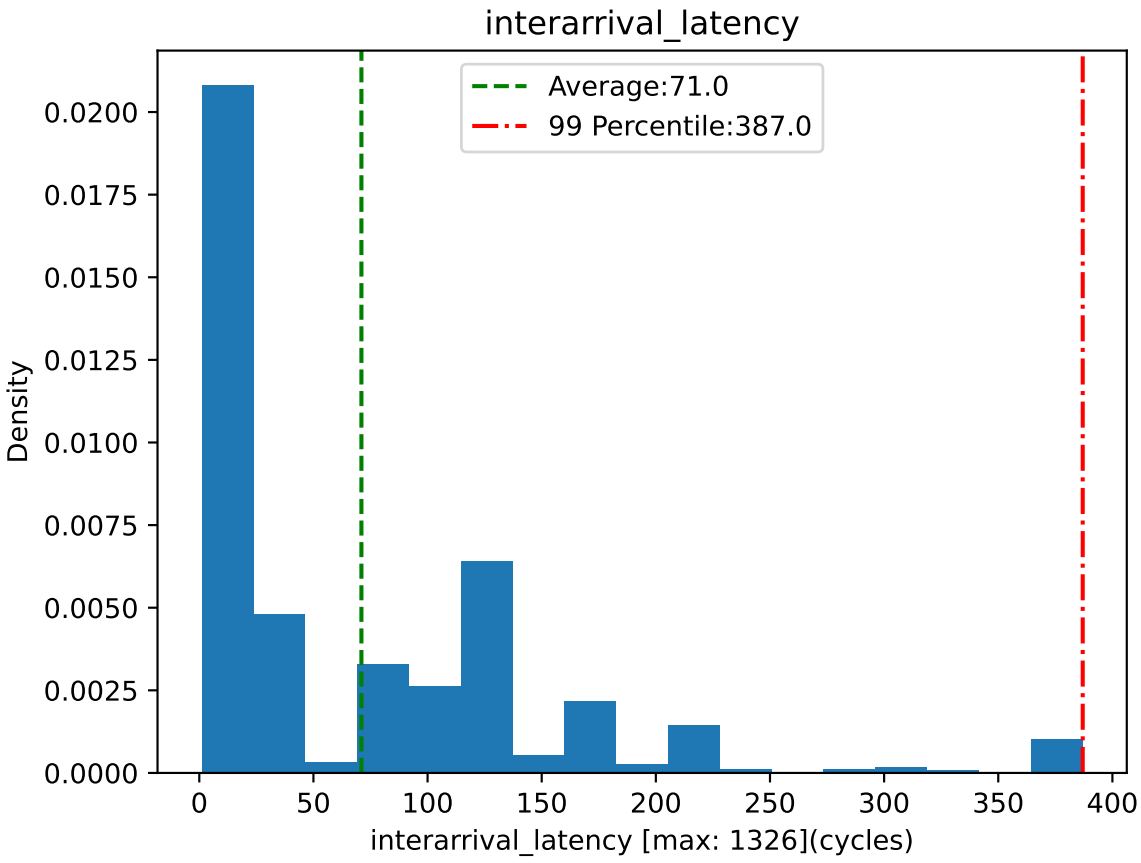


Figure: Interarrival latancy

DRAMSIM (Cont.)

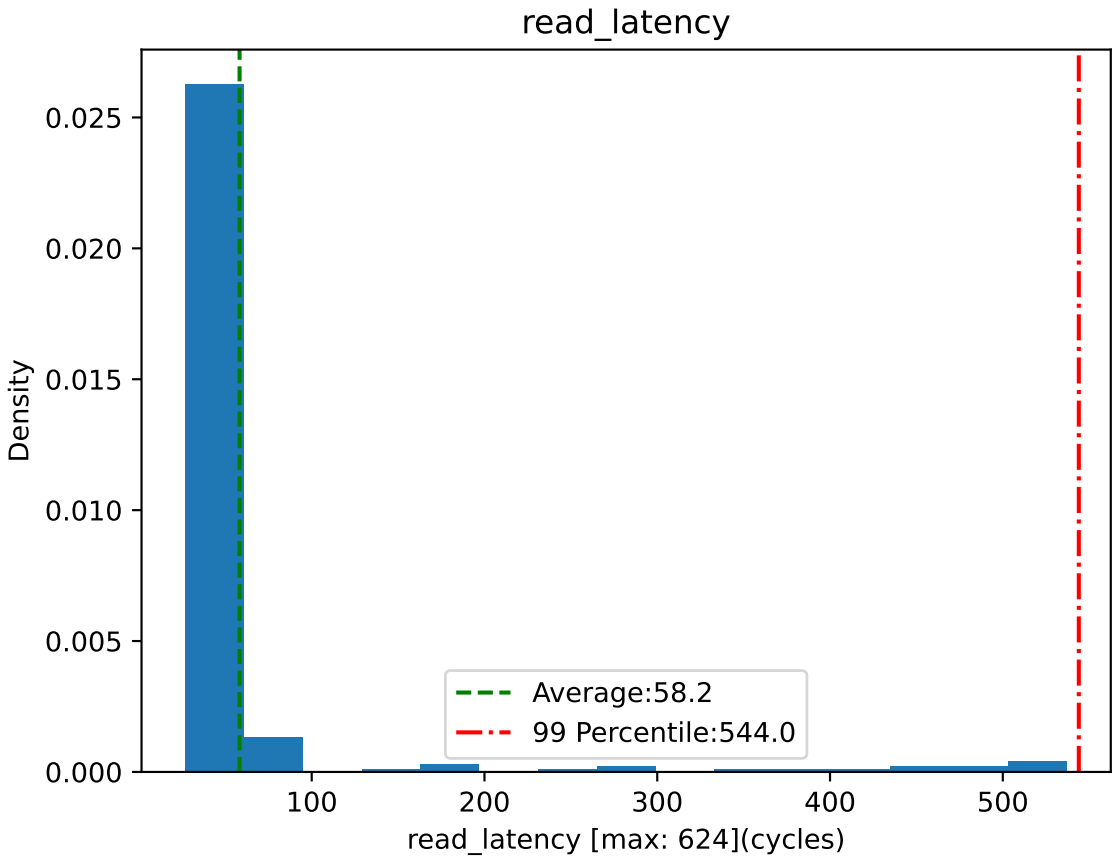


Figure: Read latancy

DRAMSIM (Cont.)

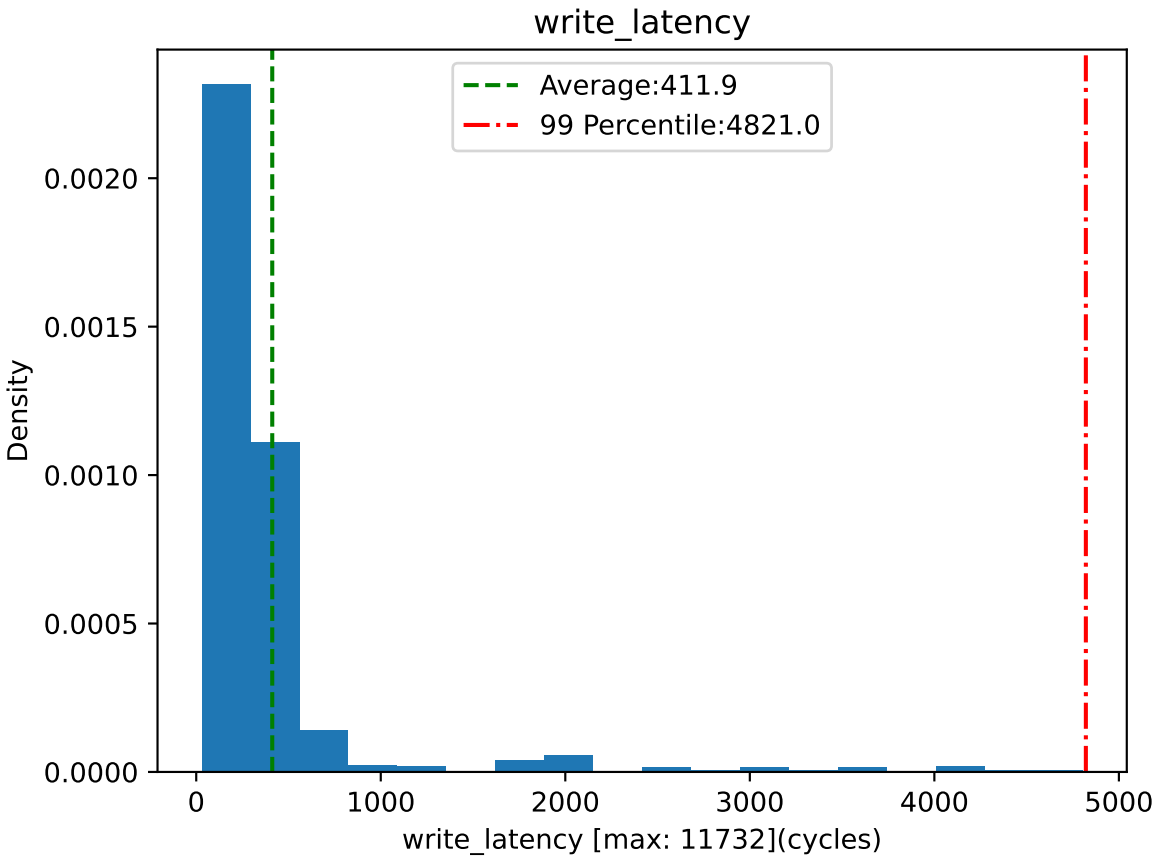


Figure: Write latancy

SimpleScaler

1. This simulator was the doctoral thesis of Mr. Austin Todd from University of Wisconsin, which was written in C language
2. This simulator is not just for memories. like Gem5, it is a system simulator.
3. By default, this simulator is capable of simulating Alpha and PISA ISA. but other ISAs can also be added to it.
4. With SimpleScaler we can simulate this Micro Architecture:
   - 4.1 **Sim-fast:** simulate without considering cache, pipeline and any type of micro architecture
   - 4.2 **Sim-safe:** simulate with considering access to memories
   - 4.3 **Sim-profile:** report number of simulations and dynamic instructions
   - 4.4 **Sim-cache:** simulate a system with access to cache
   - 4.5 **Sim-bpred:** report total branch prediction of program
   - 4.6 **Sim-outorder:** All the previous features are collected in this

SimpleScaler (Cont.)

The structure of SimpleScaler is shown in bellow:

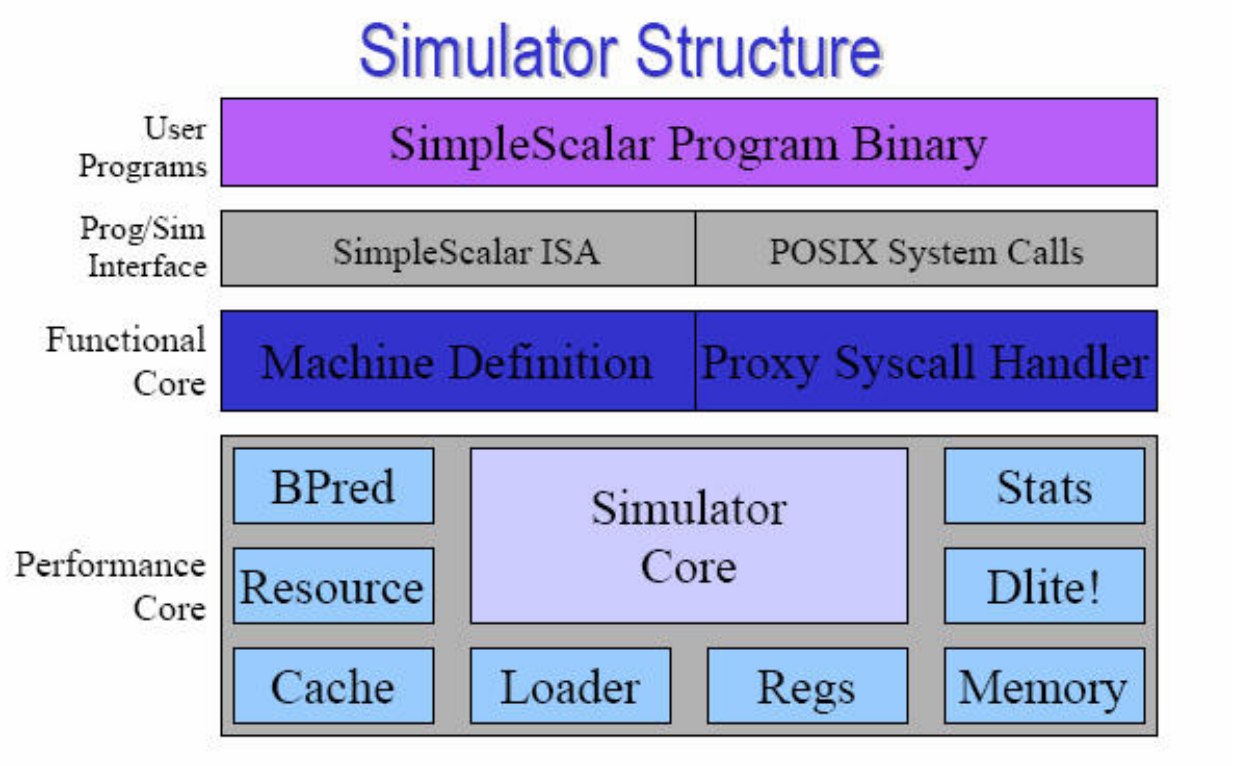


Figure: Structure of SimpleScaler

SimpleScaler (Cont.)

Advantages:

1. Open source
2. System level computer with more detail
3. Support for different architectures

Disadvantages:

1. No direct access to memory
2. Not support a new memory technologies
3. Don't report analysis with detail like `stats` file in GEM5

SimpleScaler (Cont.)

How install and build SimpleScaler?

We should clone repository in first step:

Clone repository

```
$ git clone
https://github.com/stevekuznetsov/simple-scalar.git
$ simple-scalar
```

before build, we need install dependencies:

Install dependencies

```
$ sudo apt-get update
$ sudo apt-get update install build-essential
$ sudo apt-get update install flex bison
$ sudo apt-get update install libx11-dev
```

SimpleScaler (Cont.)

Now we can build simulator:

Build

```
$ make config-alpha
```

If the build is successful, your terminal output will look like this:

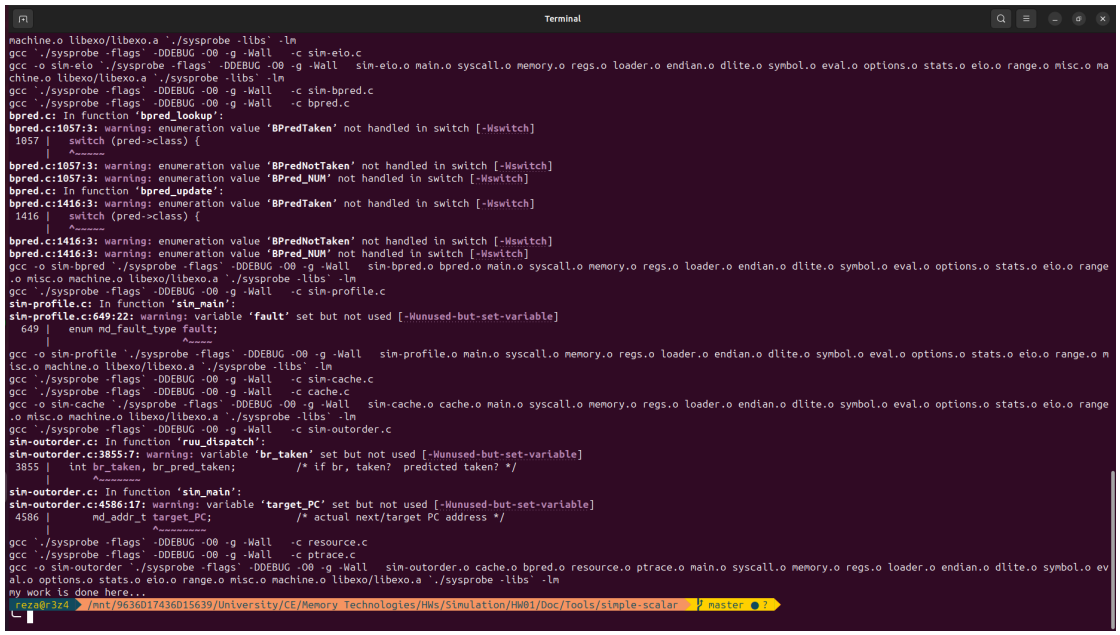


Figure: Build successful

SimpleScaler (Cont.)

Run simulation:

The default program's `.exe` file is located in the `tests/bin/` path. also the source code of program located in `tests/src/` directory. in this simulation we use `test-math` program. this program calculates sine, tangent and several other mathematical operations for various inputs.

Run simulation with this command:

Build

```
$ ./sim-safe tests/bin/test-math
```

SimpleScaler (Cont.)

The output report of simulation as bellow:

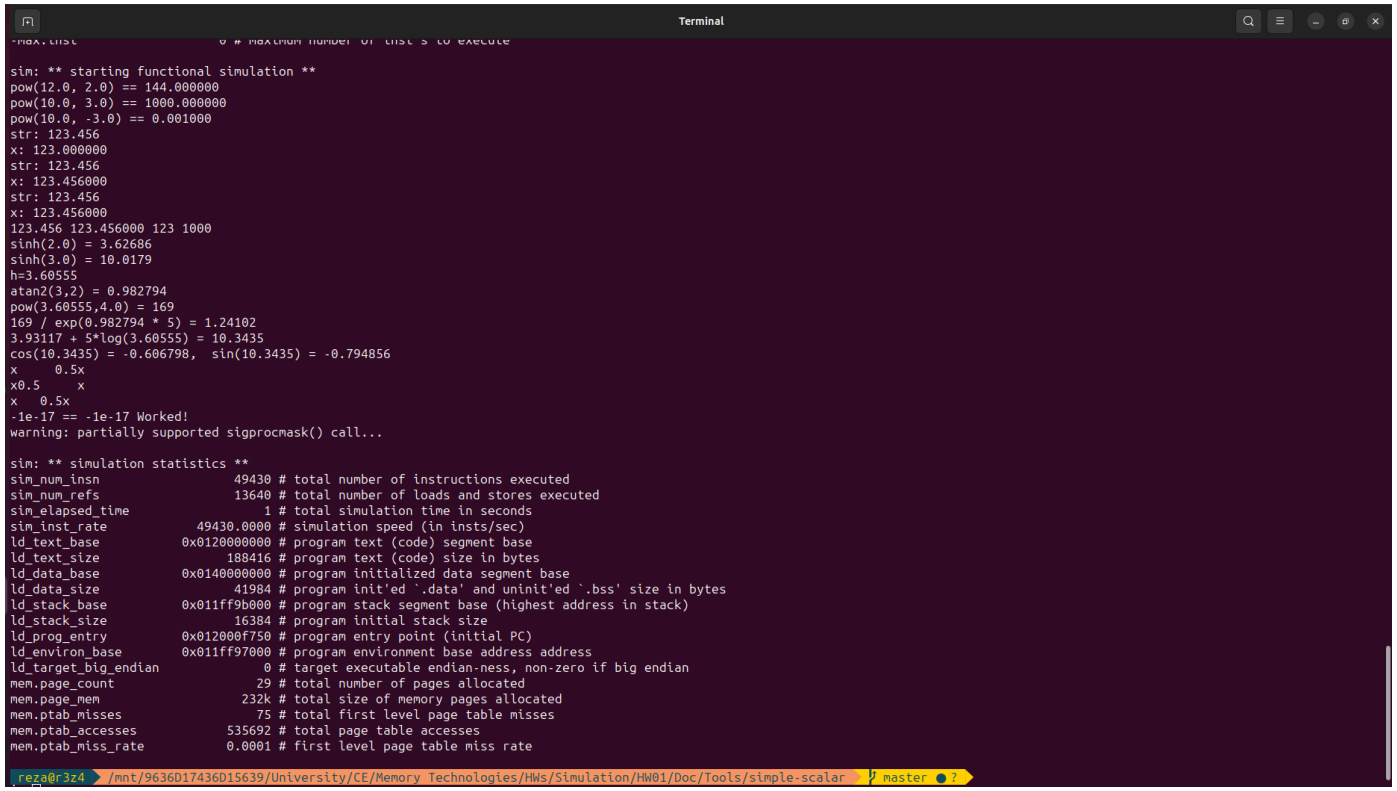


Figure: Report of `test-math` program

References

Back to start


S. Senni, *Exploration of non-volatile magnetic memory for processor architecture*, 2015.


N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, "Cacti 6.0: A tool to understand large caches," *University of Utah and Hewlett Packard Laboratories, Tech. Rep*, vol. 147, 2009.


P. Rosenfeld, E. Cooper-Balis, and B. Jacob, "Dramsim2: A cycle accurate memory system simulator," *IEEE computer architecture letters*, vol. 10, no. 1, pp. 16–19, 2011.


S. Li, Z. Yang, D. Reddy, A. Srivastava, and B. Jacob, "Dramsim3: A cycle-accurate, thermal-capable dram simulator," *IEEE Computer Architecture Letters*, vol. 19, no. 2, pp. 106–109, 2020.


D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel,

The End

Questions? Comments?

You can find this slides here:

github.com/M-Sc-AUT/M.Sc-Computer-Architecture/Memory Technologies