Assignment 1

Familiarization with GEM-5

Submitted By : Manish Kumar(2023MCS2497)

Submitted On : **16th August, 2024**

DEPARTMENT OF COMPUTER SCIENCE & TECHNOLOGY
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi, 110016

Academic Year: 2024-25

Introduction

Gem-5 is a widely-used open-source computer architecture simulator that allows us to **simulate and analyze** various computer systems, ranging from simple processors to complex, multi-core systems. It is highly configurable, supporting multiple instruction set architectures (ISAs) like X86, ARM, RISC-V, and others.

***Key Features of gem-5*:**

- **Flexible and Modular:**

gem-5 is designed to be flexible and modular, enabling users to model different system components, such as CPUs, caches, memory controllers, and networks, in a customizable way.

- **Multiple ISAs:**

It supports a variety of ISAs, making it suitable for simulating systems with different processor architectures.

- **Full-System Simulation:**

gem-5 can simulate entire systems, including the operating system, which is useful for studying system-level interactions as for **FS mode**. It also includes **SE mode**, which involves simulating a simplified environment where system calls made by the program are intercepted and handled directly by gem5. This mode is useful for evaluating programs and system models without the overhead of simulating an entire OS, allowing for faster simulations at the cost of reduced fidelity.

- **Detailed and Fast Simulation Modes:**

gem-5 provides different simulation modes, ranging from detailed cycle-level simulation (for accuracy) to faster functional simulation (for speed).

In gem5, **SimObjects** are fundamental building blocks used to define and model the various components of a simulated system. They provide an abstraction layer for hardware components and system elements, making it easier to configure, connect, and simulate complex systems.

Problem Statement:

- **Implementation of Matrix Multiplication**: Perform matrix multiplication on relatively small input sizes.

- **Simulation Using gem5**: Run the implemented program on the gem5 simulator, varying the following hardware parameters:
  - **CPU Models**: Test the program across different CPU models available in gem5.
  - **CPU Frequencies**: Evaluate performance by varying CPU frequencies from 600 MHz to 3.3 GHz in steps of 200 MHz.
  - **Memory Configurations**: Assess the impact of different memory models on the performance of matrix multiplication.

- **Performance Analysis**: Describe and analyze the changes in performance (e.g., execution time, CPI) resulting from the different CPU models, frequencies, and memory configurations.

Simulation Parameters:

- **CPU Models**:. AtomicSimpleCPU, DerivO3CPU, TimingSimpleCPU

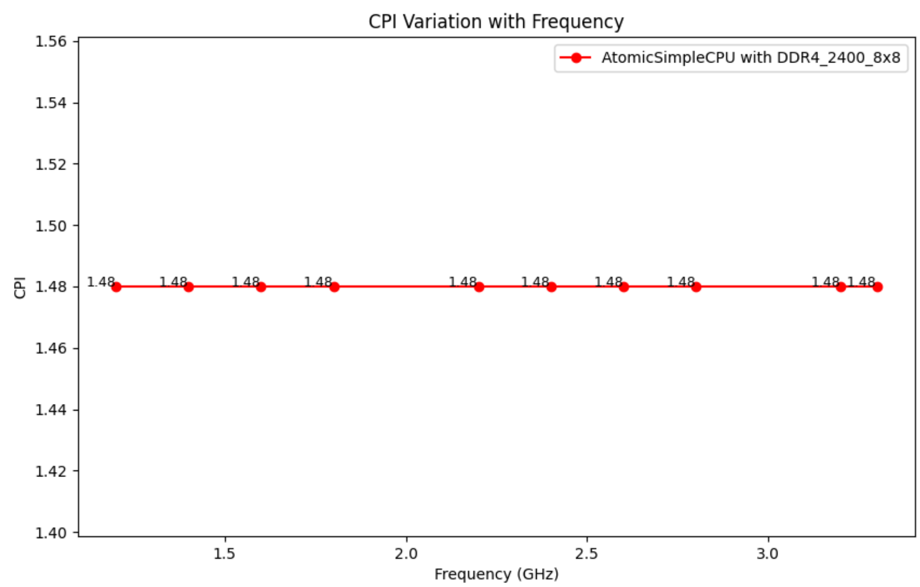
- **CPU Frequencies**: 600 MHz to 3.3 GHz (step-size 200 MHz)

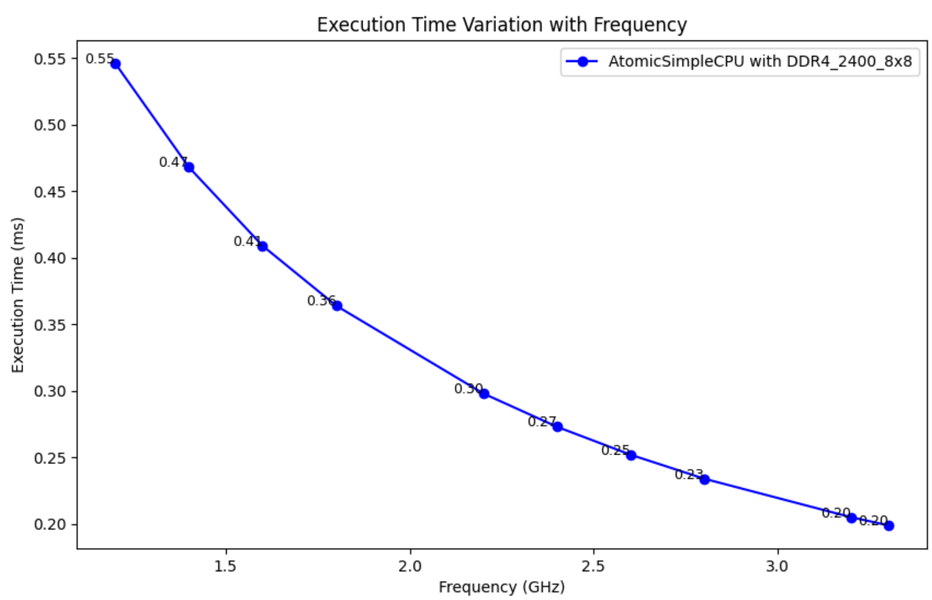
- **Memory Configurations**: DDR3_1600_8x8, DDR3_2133_8x8, DDR4_2400_8x8, DDR5_4400_4x8, LPDDR5_5500_1x16_8B_BL32

Evaluation:

- ***AtomicSimpleCPU*** **:**

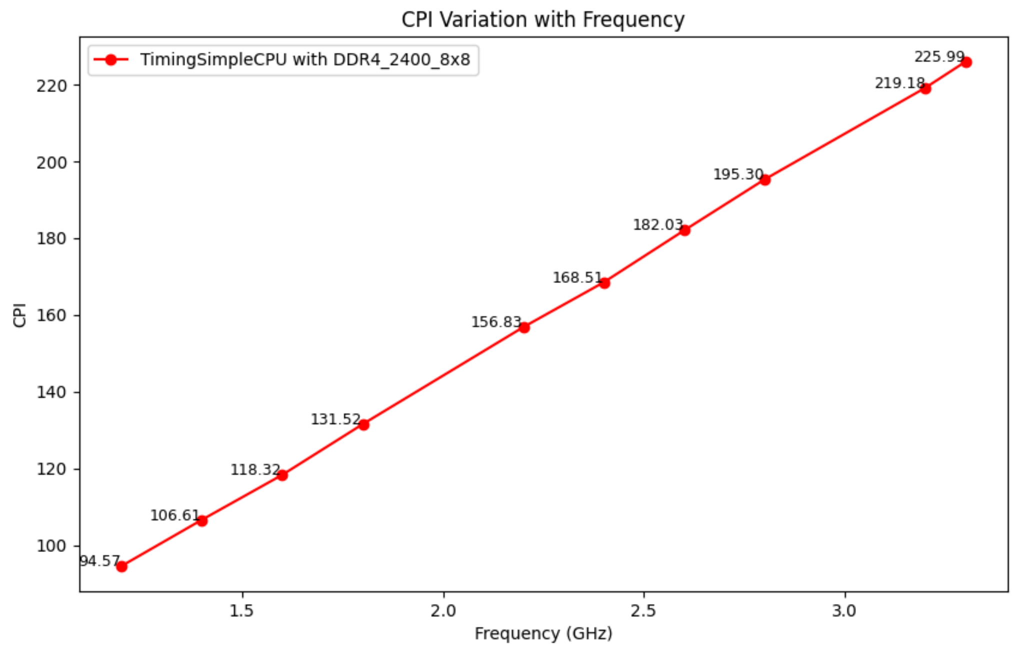
  It is a **simplified, single-cycle, non-pipelined CPU model**. It uses atomic memory accesses, meaning it assumes memory operations complete instantly without modeling detailed timing or contention. Results in very high simulation speeds.

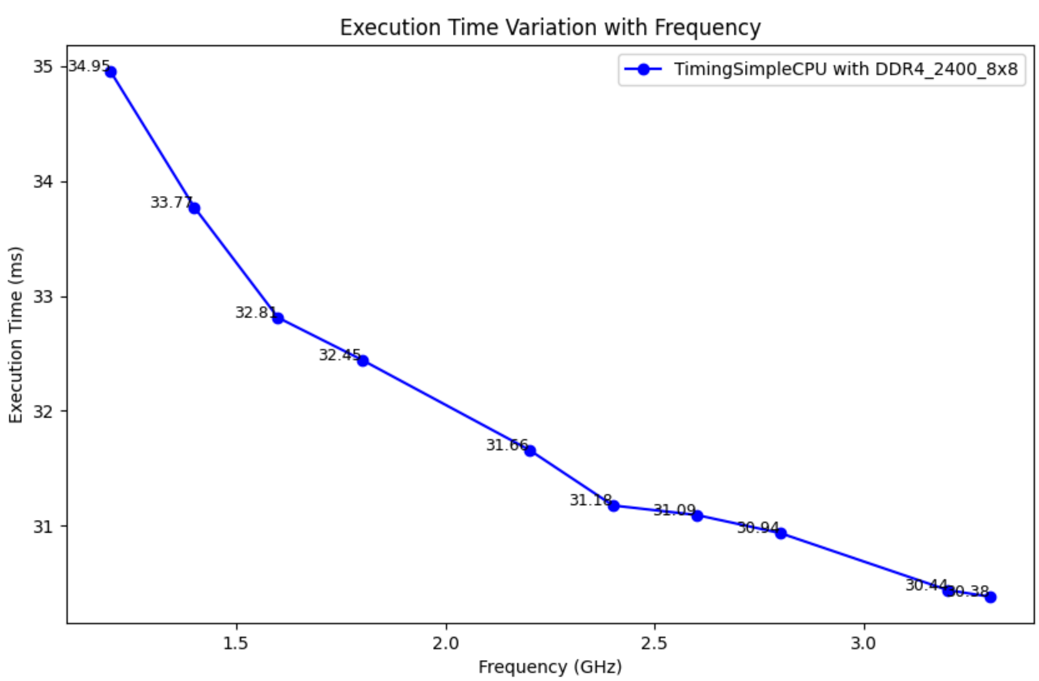




- ***TimingSimpleCPU*** **:**

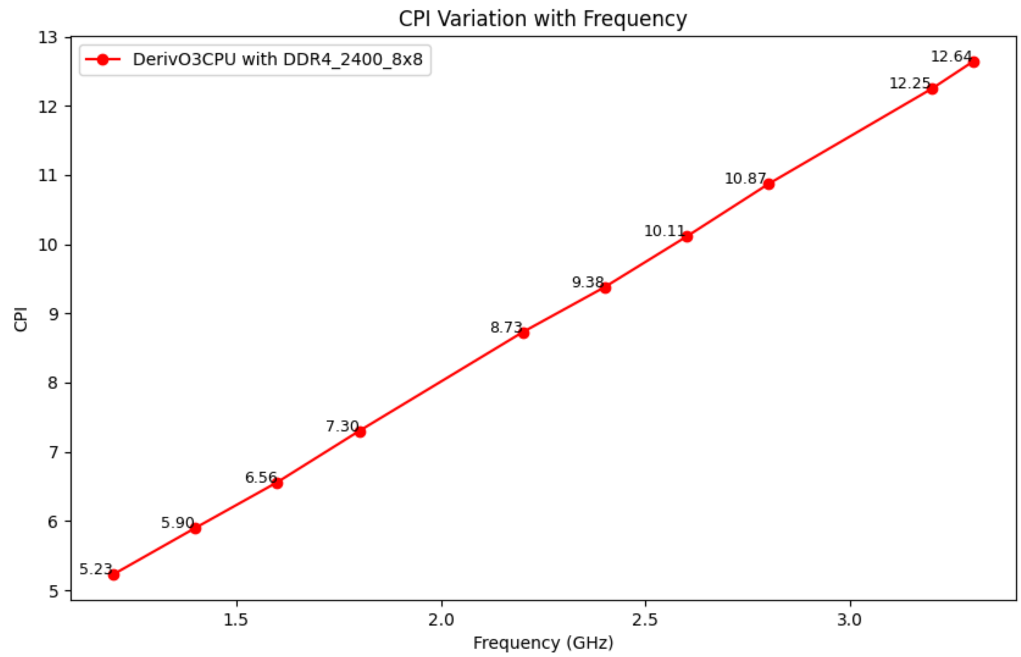
It is also a **simplified, in-order non- pipelined CPU model**. It uses detailed timing memory accesses, meaning it accounts for memory latency and contention. It is slower than *AtomicSimpleCPU*, but is more accurate.

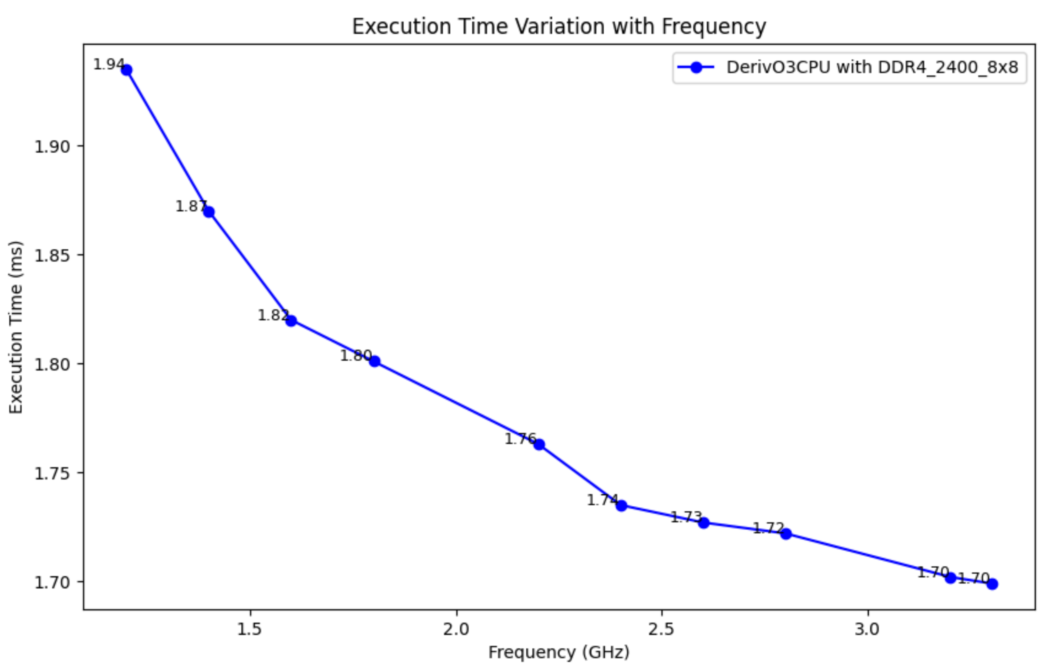


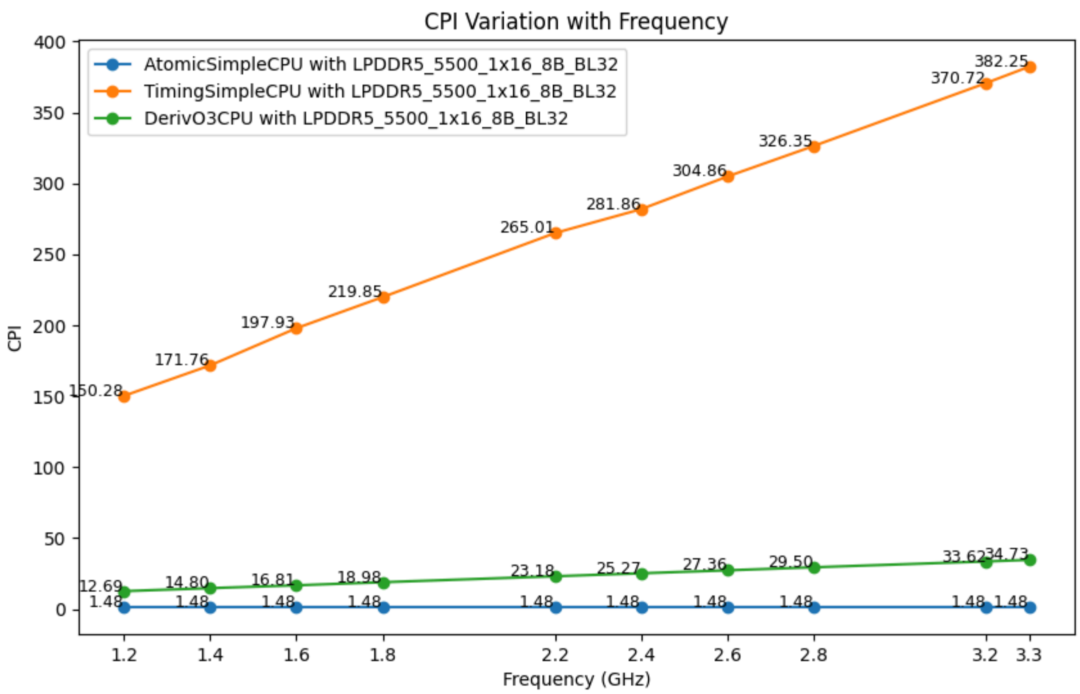


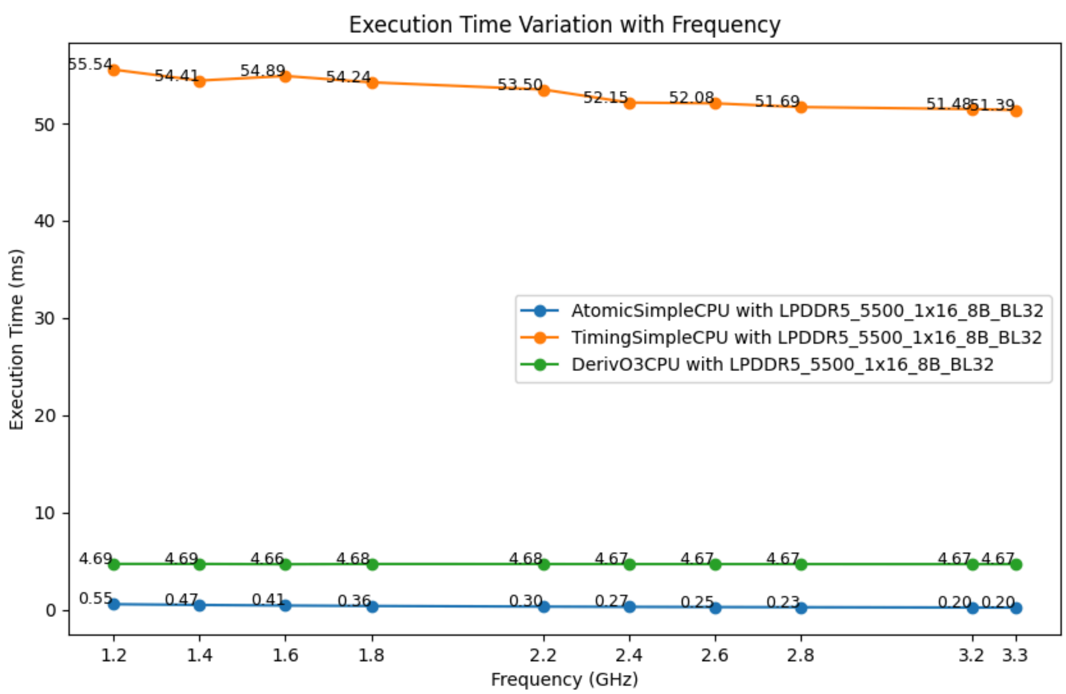
- ***<u>DerivO3CPU</u>*** **:**

  It is a **detailed, out-of-order pipelined CPU model**. It can execute multiple instructions simultaneously by breaking down the instruction execution into stages. It is significantly slower than *TimingSimpleCPU* and *AtomicSimpleCPU* due to its detailed nature, but provides high accuracy in modeling CPU behavior.

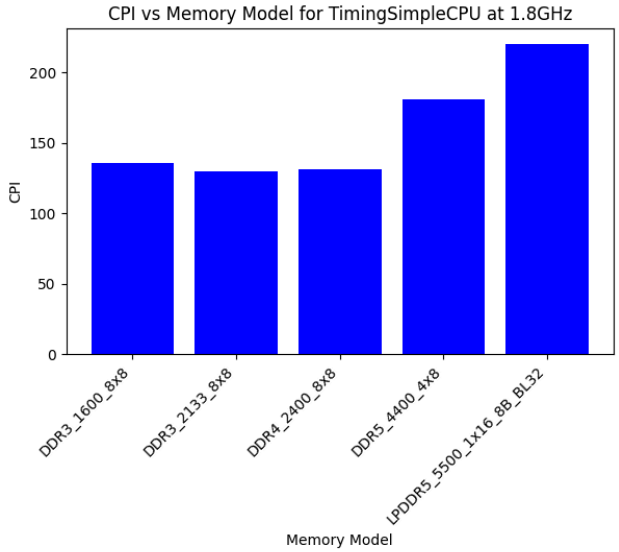


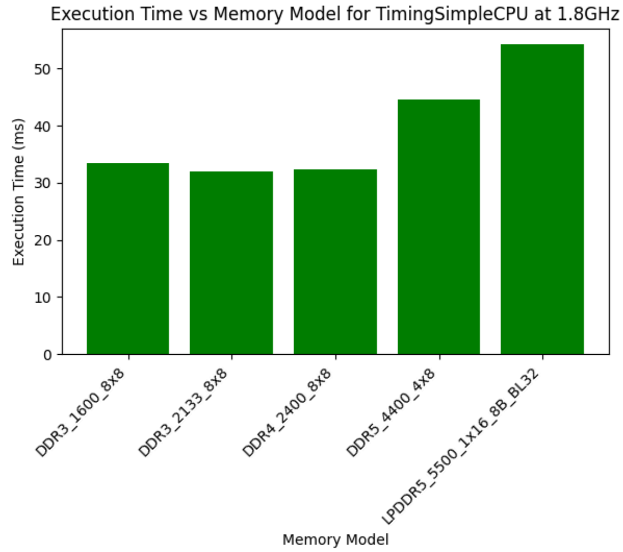






Trend of CPI and execution time *for Matrix Multiplication program across various CPU models with a fixed memory model.*





Above is the required CPI & executions time for ***TimingSimpleCPU model clocked at 1.8Ghz****.*

Out of the multiple available parameters for evaluation, the following is a partial list of parameters used to assess various CPU and memory models clocked at different rates :

```
PS C:\Users\manis\OneDrive\Desktop\HPC_sem3\lab1> python .\bench_marking.py
Randomly Selected CPI and Execution Time values:

Configuration Name                                  CPI                 Execution Time (ms)
==========================================================================================
DerivO3CPU_600MHz_LPDDR5_5500_1x16_8B_BL32          6.38                4.721
AtomicSimpleCPU_2.2GHz_DDR4_2400_8x8                1.48                0.298
DerivO3CPU_1.2GHz_DDR3_1600_8x8                     5.41                2.000
AtomicSimpleCPU_2.8GHz_DDR4_2400_8x8                1.48                0.234
TimingSimpleCPU_3.3GHz_LPDDR5_5500_1x16_8B_BL32     382.25              51.389
TimingSimpleCPU_3.2GHz_DDR5_4400_4x8                308.07              42.784
AtomicSimpleCPU_1.2GHz_DDR4_2400_8x8                1.48                0.546
TimingSimpleCPU_1.6GHz_DDR3_2133_8x8                116.51              32.309
DerivO3CPU_1.8GHz_DDR3_2133_8x8                     7.23                1.784
DerivO3CPU_3.2GHz_LPDDR5_5500_1x16_8B_BL32          33.62               4.669
PS C:\Users\manis\OneDrive\Desktop\HPC_sem3\lab1>
```

Comparative Analysis of Simulation Models:

CPU Model Impact on CPI:

- ***AtomicSimpleCPU:*** *Consistently shows a CPI of* ***1.476753*** *across different frequencies and memory models. This indicates that the AtomicSimpleCPU's CPI is not affected by changes in frequency or memory model in this simulation setup.*
- ***DerivO3CPU:*** *Exhibits a significant range in CPI values from* ***8.596721*** *to* ***10.583175****. The CPI decreases as the frequency increases and varies based on the memory model.*
- ***TimingSimpleCPU:*** *Shows a very high CPI, ranging from* ***92.74279*** *to* ***109.6874****. The CPI values are significantly higher compared to AtomicSimpleCPU and DerivO3CPU, and the effect of memory models is less pronounced compared to CPU frequency but not as significantly.*

Frequency Impact:

- *For* ***DerivO3CPU****, higher frequencies generally result in lower CPI values, suggesting better performance with increased frequency.*
- *For* ***TimingSimpleCPU****, the trend is less clear, but higher frequencies do not consistently lead to lower CPI. This could indicate that the performance of TimingSimpleCPU might be less sensitive to frequency changes compared to DerivO3CPU.*

Memory Model Impact:

- *For* ***AtomicSimpleCPU****, the CPI remains constant across different memory models, suggesting that memory bandwidth is not a limiting factor for this CPU model.*
- *For* ***DerivO3CPU****, there is some variation in CPI with different memory models, but the effect is less pronounced compared to changes in frequency.*
- *For* ***TimingSimpleCPU****, the CPI values vary across memory models, but the impact of memory models seems less significant compared to the impact of frequency.*