GitHub Repository:
https://github.com/aravind-aca/Cache-Prefetching-and-Memory-Latency-Characterization

ACA PROBLEM STATEMENT

Question-1:

- Configure and execute the SimpleScalar 3.0 simulator for the PISA ISA to analyze the baseline performance of its out-of-order pipeline.
- Build the simulator, run a sample benchmark, and record essential performance metrics such as CPI, IPC, cache miss rates and branch prediction accuracy.
- Establish a baseline configuration to serve as a reference for evaluating advanced cache and pipeline enhancements in subsequent phases.

Question-2:

- Extend the baseline SimpleScalar simulator by integrating cache prefetching mechanisms to examine their effect on processor performance.
- Implement and compare prefetching schemes including Next-Line, One-Block Lookahead, and Stride Prefetching within the cache subsystem.
- Analyze and interpret changes in CPI, IPC and cache miss rates to identify the most efficient prefetching approach that achieves performance improvement with minimal resource overhead.

Question-3:

- Develop and employ a memory latency measurement model (memlat.c) to characterize latency across varying memory sizes, access strides, and concurrency levels.
- Perform systematic experiments including backward/forward scans, index-based and stride-based accesses, and concurrent thread tests to expose hierarchical memory behavior (L1 → L2 → L3 → DRAM).
- Compare latency trends, identify cache transition points, and calibrate simulated memory parameters using the measured data to achieve realistic timing in SimpleScalar.
- Interpret how calibrated latency values influence overall pipeline and prefetch performance, bridging the gap between simulation and real-hardware behavior.

Question 4:

- Extend the calibrated simulator environment to perform detailed CPU performance benchmarking and Power / Energy modelling using the latency data obtained from Phase 3.
- Execute targeted micro-benchmarks covering integer, floating-point, memory-copy, and mixed workloads to evaluate CPU throughput and analyze the effect of memory hierarchy delays on execution performance.
- Develop and apply a Power / Energy estimation model that maps measured latency (ns) to corresponding energy cost (pJ) for each memory level — L1, L2, L3, and DRAM.
- Correlate latency, performance, and power results to derive an analytical understanding of how increasing memory access latency impacts CPU efficiency and total energy consumption.
- Generate a comprehensive set of calibrated results and energy summaries to serve as the quantitative foundation for Phase 5 visualization and dashboard analysis.

Question 5:

- Design and implement an **interactive CPU performance and power visualization dashboard** that consolidates data from all previous phases (1 → 4) of the project.
- Build a **web-based analytical interface** using Flask + Plotly Dash + Pandas to integrate IPC, CPI, latency and energy metrics into a unified view.
- Process Phase-4 results through automated scripts to generate structured datasets (memlat.csv, power_summary.csv, summary_report.txt).
- Develop interactive charts and tables to compare cache performance, latency transitions and energy scaling across memory levels.
- Validate the dashboard for real-time data refresh and analytical accuracy, demonstrating end-to-end integration from simulation to visualization.