Vector Processor Simulator

Project Objectives:

The primary goal of this project is to provide students with a comprehensive understanding of vector processors and their efficiency in handling data-parallel workloads. Through the development of a vector processing simulation environment, students will explore the architectural features that allow vector processors to achieve high levels of data throughput for certain types of applications. Additionally, the project aims to investigate the effects of various vector processor parameters on the performance of data-parallel tasks. This will also serve to enhance the students' skills in programming, collaborative work, technical writing, and critical analysis of performance metrics and architectural trade-offs.

Project Details:

PART I:

Students are tasked with creating a simulation environment that models a vector processor capable of executing data-parallel workloads. The simulation should be able to demonstrate the processor's performance across a variety of vector lengths and operation types. The vector processor simulator will have to account for several key architectural components, such as vector registers, vector functional units, and memory management units optimized for vector operations.

Simulator Input: The input to the simulator will consist of a set of synthetic data-parallel workload traces designed to test the processor's performance across different vector operation types (e.g., arithmetic, logical, and memory access operations). Each workload trace will contain a series of vector operations, their associated vector lengths, and data dependencies. The traces will represent diverse application scenarios to ensure a comprehensive evaluation of the vector processor's capabilities.

Simulator Features to Implement:

- Vector Register File: A configurable number of vector registers of variable lengths.
- Vector Functional Units: A set of functional units that can perform vector operations, with support for pipelining to exploit instruction-level parallelism.
- Memory Management: A memory subsystem that includes vector load/store capabilities and can simulate memory alignment constraints and stride access patterns.
- Performance Counters: Mechanisms to track and report various performance metrics, such as vector operation throughput, latency, and efficiency.
- Configurable Parameters: Allow users to configure key parameters such as vector register size, number of functional units, memory bandwidth, and

latency to study their impact on performance.

Simulator Output: The output should include detailed performance statistics for each workload trace, including:

- The number of vector operations completed.
- The average throughput of vector operations (operations per cycle).
- The utilization ratio of vector registers and functional units.
- Memory bandwidth usage and the impact of memory alignment and stride on performance.
- A breakdown of the latency contributions from different stages of the vector processing pipeline.

Students will be expected to document their simulator design, justify the architectural choices made, and provide a comprehensive analysis of the performance results obtained from the simulator. The analysis should include a discussion on how different vector processor configurations affect the performance of data-parallel workloads and identify potential bottlenecks or inefficiencies in the simulated architecture.