Honours Project Report

Vedant Pahariya — Priyanshi Jain

July – December 2025

Contents

1 Ara

1.1 Literature Review

In the following sections, we study about the Ara vector coprocessor and its research papers.

1.2 About Vector Processor

1.2.1 Why we need it?

To tackle the Von Neumann Bottleneck (VNB), referred as the slowdown caused by limited bandwidth between memory and CPU, which restricts how fast instructions and data can be fetched, especially in data-intensive tasks.

By using the vector processors, we can reduce the number of instructions needed to perform operations on large datasets, thereby improving performance and efficiency.

> A scalar processor fetches one instruction, processes one data point.
> A vector processor fetches one instruction, but processes many data points.

1.2.2 What is a vector processor?

A vector processor is a CPU that executes one instruction on an entire array (vector) of data simultaneously, making it ideal for data-parallel tasks like scientific computing and AI, where it improves both speed and energy efficiency by reducing instruction overhead and maximizing hardware usage.

Note: Vector processors do not directly increase memory bandwidth or change shared path. They just reduce pressure on memory bandwidth by doing more work per instruction and organizing memory accesses more efficiently.

1.3 SIMT vs SIMD

1.3.1 Understanding Thread

A thread is just a sequence of instructions that the CPU runs like a mini program. If you run the same code twice at the same time with different inputs, it means two threads.

So when your program runs in parallel, it spawns multiple threads — each has its own instructions, registers, and data, and the hardware runs them in parallel (or sometimes interleaved). Single Instruction, Multiple Threads (SIMT) is a thread level implementation. SIMT = many threads, all run the same instruction at the same time but on different data.

SIMT and SIMD can (and do) coexist in the same system. In fact, that's how most modern GPUs work.

SIMD is hardware-level, strict parallelism. All lanes do the same thing at the same time.

SIMT is thread-level. Threads run in parallel but can behave differently, though better performance comes if they stay in sync.

Why do all GPUs Use SIMT - Isn't SIMT Slow With Branching?

Refer this Youtube Video and its Doc

1.4 Structure of Ara

1.4.1 Basic Latex Template

Introduce about the Title here.

Reference: https://www.youtube.com/watch?v=ic1UMeuCBA8
GeeksforGeeks

- **1**:
- **2**:

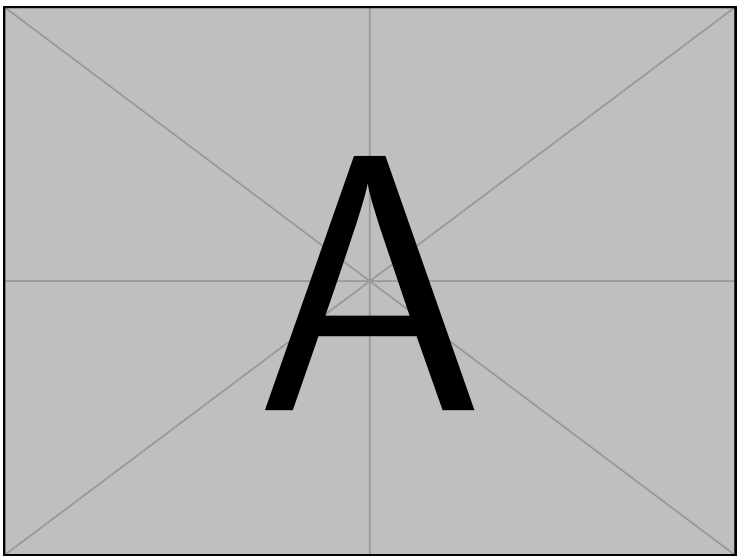

Figure 1: Sample Image

Verilog is a Case Sensitive language.
The term "module" refers to the text enclosed by the keyword pair **module** . . . **endmodule**.
Module is the fundamental descriptive unit in Verilog language.
Keyword "module" is followed by the name of the design (ABC here) and parenthesis - enclosed list of ports.