Final Project: Abstract

Superscalar Fixed-Length Vector Processor

November 7, 2023

Dylan-Matthew Garza
Elijah Sargeant
Dr. Sawalha
ECE-5580: Computer Architecture
Due: 4 November
Western Michigan University
Department of Electrical and Computer Engineering

Contents

1 Proposal

1.1 Project Objective

The goal of this project is to comprehensively understand vector processors and they achieve efficiency through handling data-parallel workloads. Vector processors are a very important paradigm in regards to SIMD (Single Instruction Multiple Data) methodologies since they achieve parallelism by having parallel data operations. In this implementation, the the vector processor will be superscalar with out of order execution, in-order commit. In this implementation, the vector processor will support fixed-length vector sizes of 32-bit.

1.2 Methodology

1.2.1 Simulator Features

A superscalar vector processor will require several components such as:

- Multiple Functional Units for each lane on vector instructions
- Register Renaming Using RAT and RRAT
- Vector Instruction Queue: Instruction queue to hold vector instructions
- Vector Pipeline Stages:
  - Instruction Fetch
  - Decode and Rename
  - Dispatch
  - Vector Execute
  - Writeback/Memory Access
  - Commit
- Flexible Chaining of operation results to other Functional units and Load/Store Queue
- 4-Lane Vector Functional Units
- 4-Sets of Vector Funcional Units

1.2.2 Input and Output

Input

- Trace file based on vector instructions
- Number of execution lanes to functional units

Output

- Number of vector instructions executed per cycle (IPC)
- Average IPC
- Maximum IPC

1.3 Evaluation Plan

The project will evaluate the efficiency of the implemented superscalar fixed-length pipeline. By gathering information about the IPC, we can analyze how the number of lanes affect the IPC of a fixed-length system.

2 Project Members

- Dylan-Matthew Garza:
  Responsible for writing the simulator and researching the related topics.
- Elijah Sargeant:
  Arbitrating the progress of the project as a whole and ensure correct implementation.

3 Block Diagram

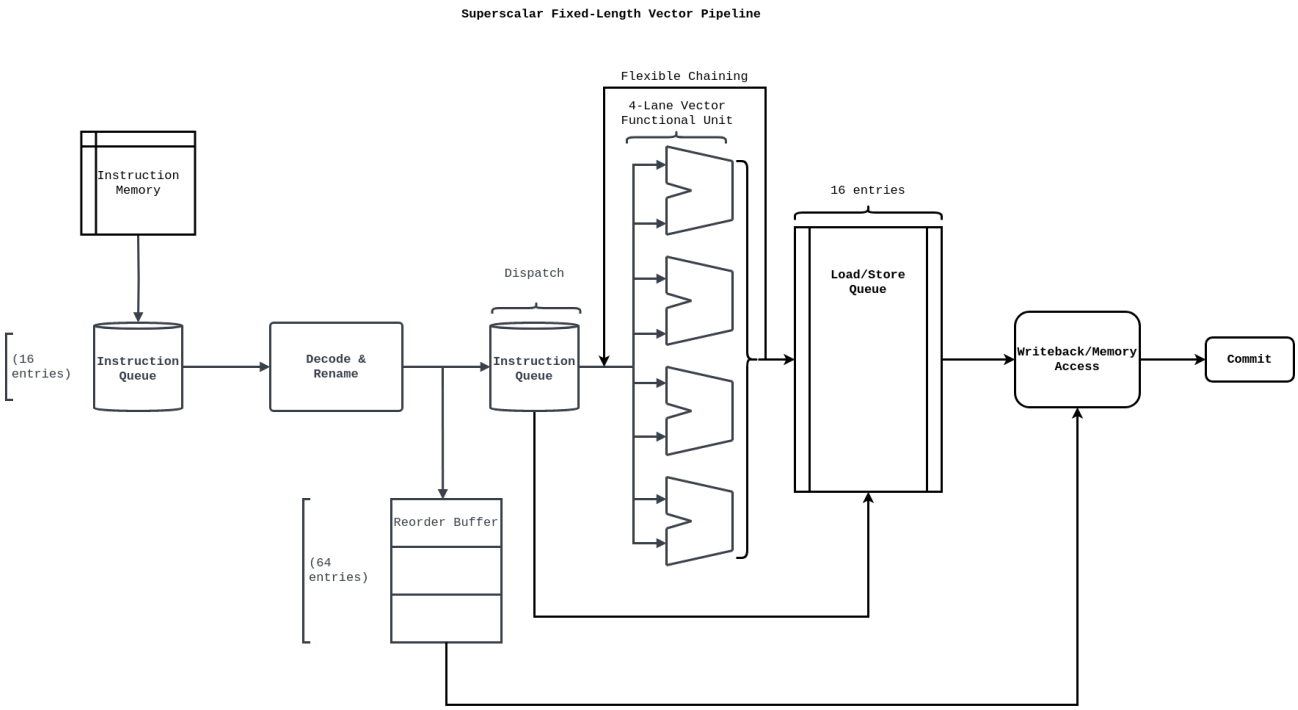

Figure 1: Proposed Pipeline features to Simulate

In this implementation, the exectution of vector instructions will not be limited to the number of vector functional units, only by the number of lanes to each functional unit.

4 References

[1] R. Espasa, M. Valero and J. E. Smith,
"Out-of-order vector architectures," Proceedings of 30th Annual
International Symposium on Microarchitecture, Research Triangle
Park, NC, USA, 1997, pp. 160-170, doi: 10.1109/MICRO.1997.645807.

[2] K. Patsidis, C. Nicopoulos, G. C. Sirakoulis and G. Dimitrakopoulos,
"RISC-V2: A Scalable RISC-V Vector Processor," 2020 IEEE
International Symposium on Circuits and Systems (ISCAS), Seville,
Spain, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9181071.

[3] S. S. Chen and A. J. Schiffleger, ‘‘Flexible chaining in
vector processor with selective use of vector registers as
operand and result registers ,’’ Apr. 28, 1987

[4] Di Mascio, Stefano & Menicucci, Alessandra & Gill, Eberhard
& Monteleone, Claudio. (2023). Extending the NOEL-V Platform with
a RISC-V Vector Processor for Space Applications. Journal of Aerospace
Information Systems. 20. 1-10. 10.2514/1.I011097.