Lab 4: Out-of-Order PARCv2 Processor

Bryan SebaRaj

Professor Abhishek Bhattacharjee

EENG 420 - Computer Architecture

May 7, 2025

Abstract

Out-of-order execution is a technique used in computer architecture design to improve the performance a processor by allowing instructions to be executed in a different order than they appear in the program. This lab focused on implementing an out-of-order PARCv2 processor with a reorder buffer and speculation. Splitting the implementation into two parts, I first implemented a reorder buffer to allow for out-of-order execution. I then extended the reorder buffer to support speculation, allowing the processor to execute instructions before it is known whether they will be needed. The processor was tested using a set of benchmarks, including vector add, complex multiply, masked filter, and binary search. The cycle counts and IPC for each benchmark were recorded and compared across the two implementations and a baseline pipelined processor with bypassing and no reorder buffer. The results showed that the out-of-order PARCv2 processors with a reorder buffer and speculation outperformed the baseline pipelined processor with bypassing in most cases, demonstrating the benefits of out-of-order execution and speculation.

Design

Objective 1: Out-of-Order PARCv2 Processor with Reorder Buffer

Given the existing I2O2 PARCv2 processor without a reorder buffer, the I2O1 PARCv2 processor was constructed by adding a reorder buffer, enabling bypassing out of the reorder buffer in the scoreboard, and modifying the datapath to enable values to be written from the reorder buffer instead of from the writeback stage. Specificaly, the reorder buffer was implemented as a

Objective 2: Out-of-Order PARCv2 Processor with Speculation

No extensions were implemented.

Testing Methodology

Evaluation

The pipelined in-order processor with bypassing, pipelined out-of-order processor with the reorder buffer, and pipelined out-of-order processor with speculation were tested using the following benchmarks: vector add, complex multiply, masked filter, and binary search. The cycle counts and IPC for each benchmark are shown below in the format, cycle count / IPC.

Benchmark	pv2byp	pv2ooo	pv2spec
binary search	1749/0.731275		
complex multiply	15,312/0.121735		
masked filter	13,832/0.32526		
vector add	473/0.961945		

Discussion

Trivial analysis of the performance of the three processors with the provided benchmarks elucidates the significant benefits of out-of-order execution when compared to naive, in-order bypassing processor.

Figures

Figure 1. Datapath of out-of-order, singule-issue processor.

Functional Unit[8]	Instruction Type[7:6]	Pending[5]	W[4]	X3[3]	X2[2]	X1[1]	X0[0]

Figure 2. Diagram of reorder buffer for part 1.

Functional Unit[8]	Instruction Type[7:6]	Pending[5]	W[4]	X3[3]	X2[2]	X1[1]	X0[0]

Figure 3. Diagram of reorder buffer for part 3.