UNIVERSITY OF BRISTOL

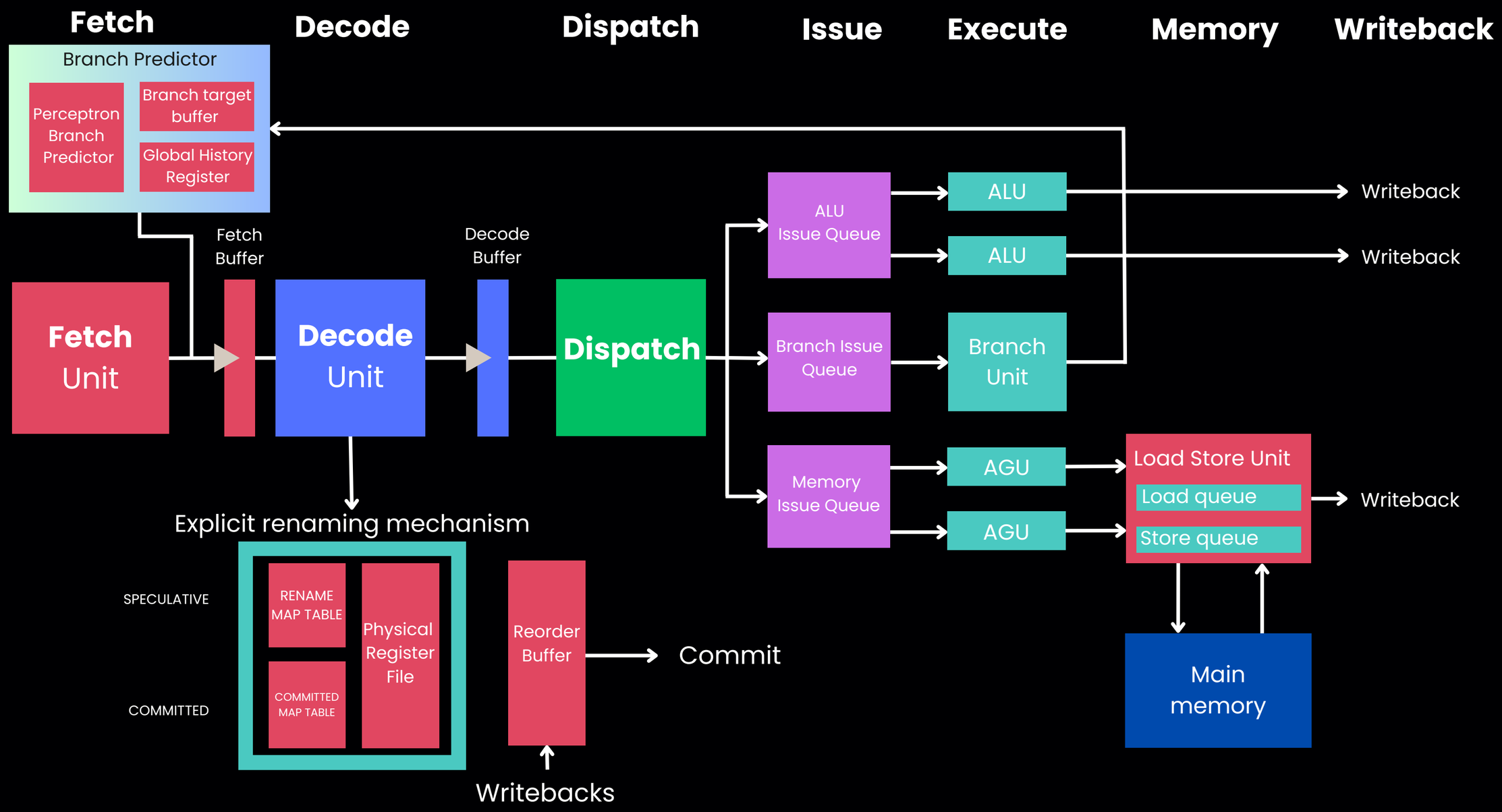
ADVANCED COMPUTER ARCHITECTURE

CLARENCE LEE SHENG

DESIGNING A SUPERSCALAR PROCESSOR SIMULATOR

FINAL ASSIGNMENT

DESIGN OF THE ARCHITECTURE



SIMULATOR FEATURES

7 stage pipelining execution

- Pipelined execution comprising of Fetch, Decode/Rename, Dispatch, Issue, Execute, Memory and Writeback stages

N-way Superscalar

- Configurable number of ALUs and Address generation Units
- Configurable number of Fetch/Decode/Dispatch/Issue/Commit Instructions per cycle
- Configurable size of fetch and decode buffers

Out of order execution and explicit renaming

- Register renaming with 128-entry Reorder Buffer
- Explicit register renaming with unified physical register file (n registers in physical register file > architectural register file)
- RAT snapshotting on branch instructions
- Forwarding enabled with Bypass network and central data bus
- Broadcasting enabled to issue queues/reservation stations

SIMULATOR FEATURES

Fetch and Decode Buffers with 3 Issue Queues

- Fetch and Decode Buffers of size 10 each, decode stalls when no free registers or reorder buffer is full
- Separate issue queues for ALU, Memory and Branch Units with 32 entries each

Branch Prediction (Configurable)

- Includes branch target buffer and global history register
- Static: Always taken, not taken
- Dynamic: 1-Bit, 2-Bit saturating counter with Branch Target Buffer
- Machine learning: Perceptron branch predictor using global history register with 8 entries

Registers and Memory

- 32 Architectural Registers (configurable)
- 4096 Bytes Main Memory (configurable)
- 128 Physical Registers (configurable)

BENCHMARKS

Vector Addition:

- **A**dd two vectors of size 20

Matrix Multiplication:

- Multiply two 8 x 8 matrices

Odd counts:

- Increment a sum at every odd number

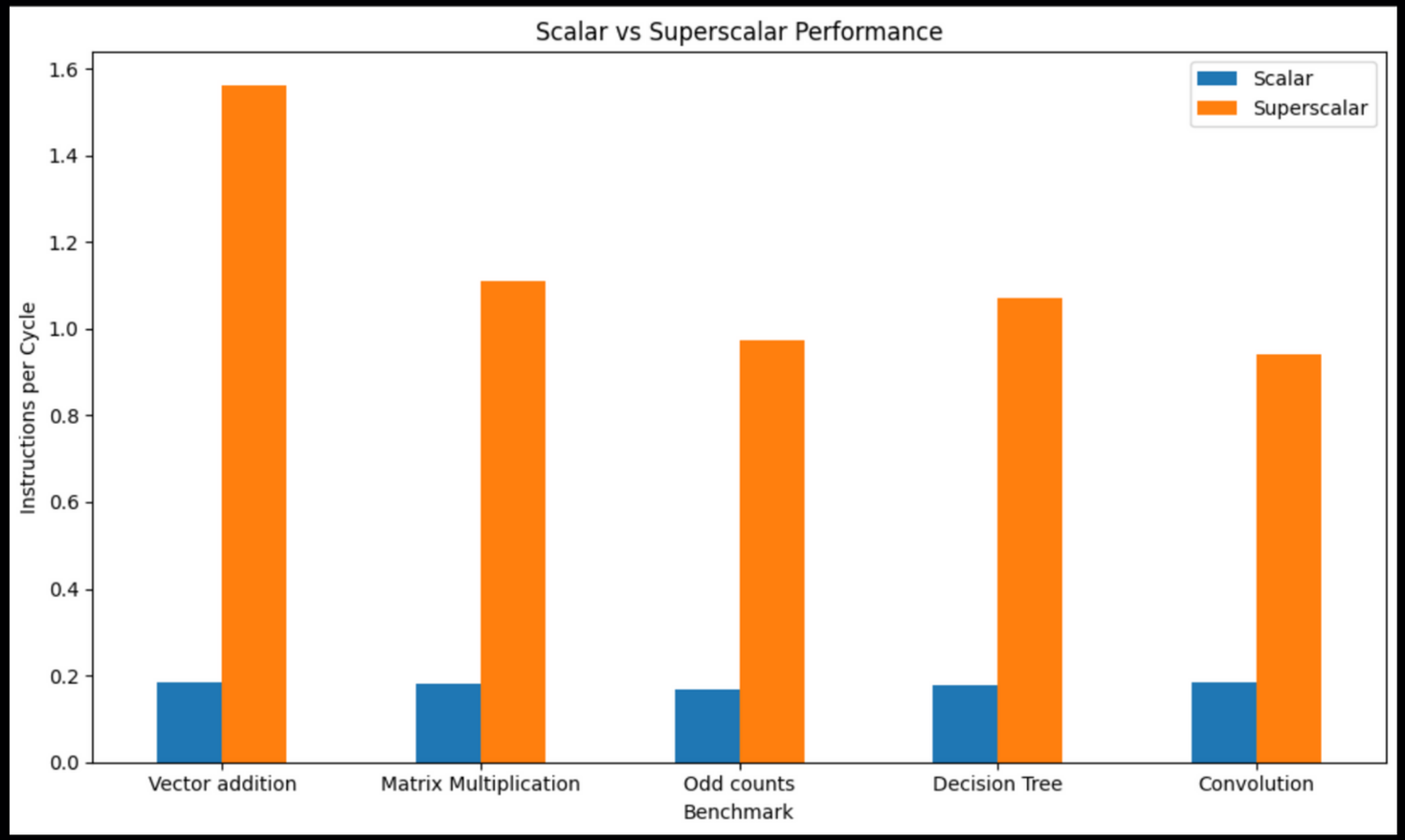
Convolution Layer:

- Performs a convolution operation with a 2x2 kernel and stride 1 on an 8 x 8 matrix

Decision Tree Classifier:

- Performs binary classification on 300 data points with 3 features each on a trained decision tree classifier of max depth 3

SCALAR VS SUPERSCALAR



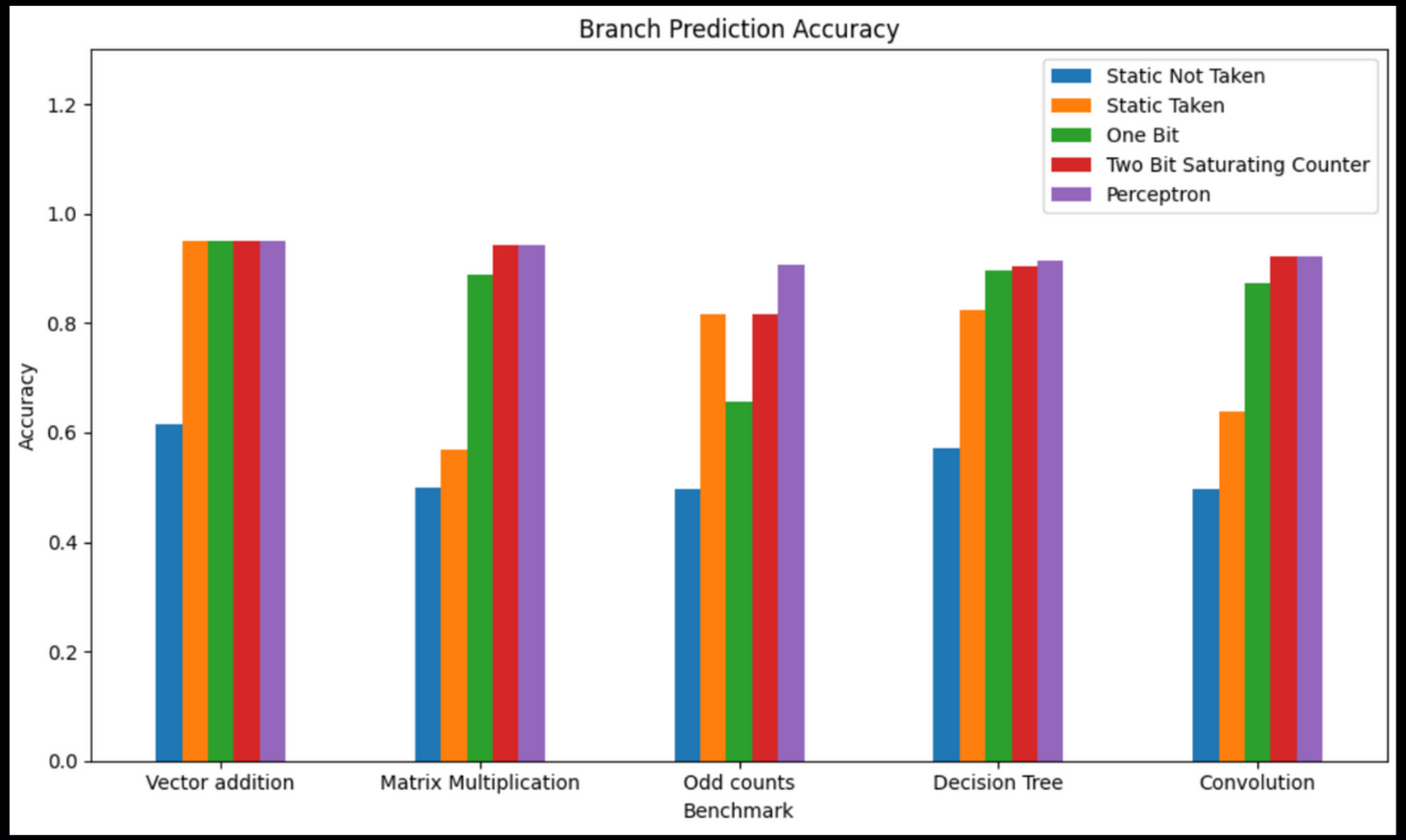
Hypothesis:
Superscalar and out of order performance will improve the instructions per cycle performance of processor

Experiment:
Measured the instructions per cycle for each benchmark for scalar and 5-way superscalar performance

Results:
Across all benchmarks, performance of superscalar increased by at least 5 times as compared to a scalar, increasing by around 8 times for vector addition.

Note: Scalar used a 5 stage while Superscalar used a 7 stage pipeline

BRANCH PREDICTORS



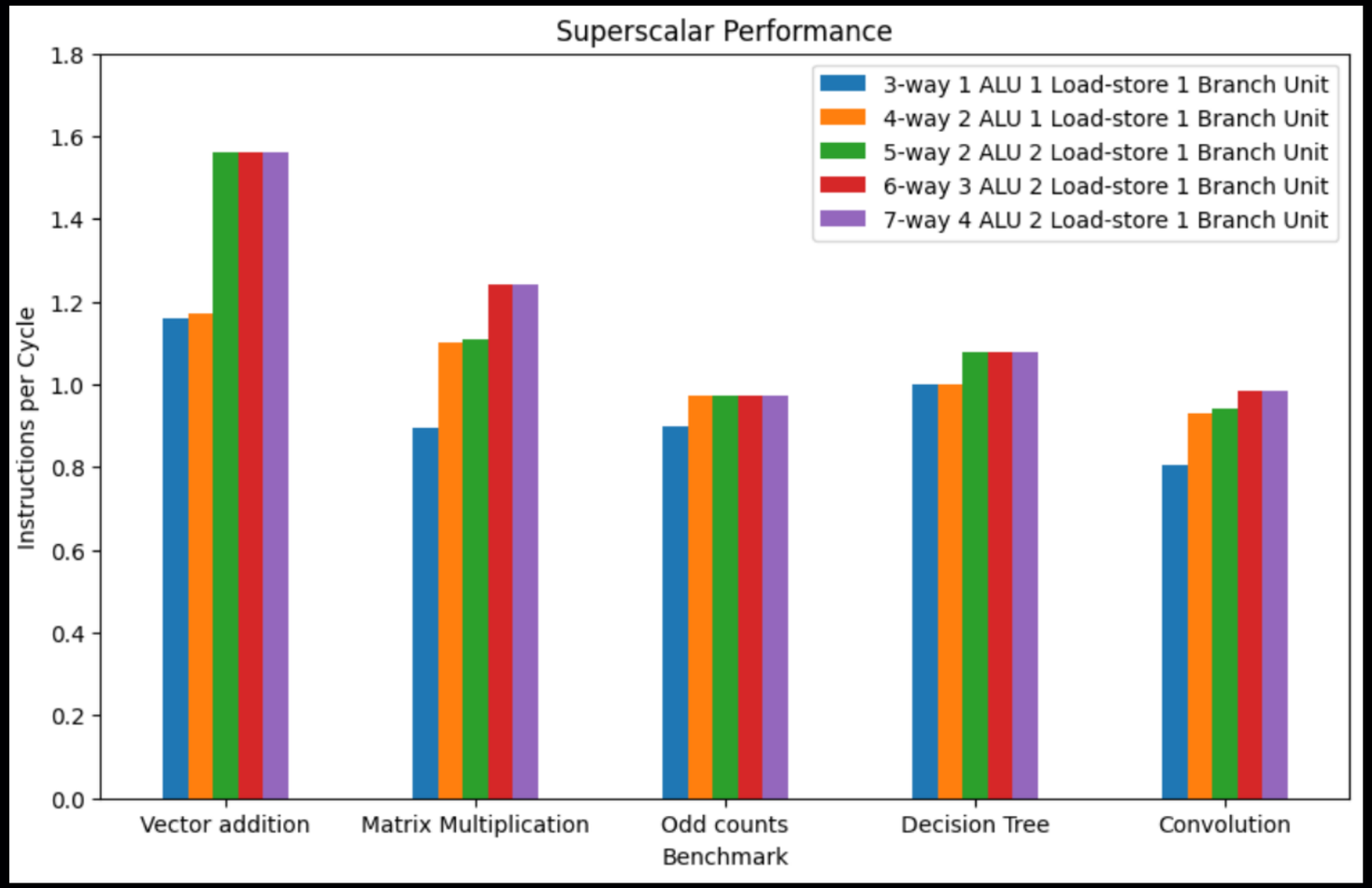
Hypothesis:
Using dynamic branch predictors will improve branch prediction results, Using a perceptron based branch predictor should lead to improved branch prediction results

Experiment:
Branch prediction accuracy was tested across 5 different benchmarks

Results:
Dynamic branch predictors typically perform better than static branch predictors with **2 Bit Saturating counter** and **Perceptron** based branch prediction providing the best results. **Perceptron** branch predictor consistently produced the **top branch prediction accuracy.**

Perceptron branch accuracy over two bit saturating counter becomes evident in benchmarks with more **unpredictable branch outcomes** such as odd counts and decision tree classifier

SUPERSCALAR



Hypothesis:
Increasing the number of execution units should increase performance

Experiment:
N-way superscalar performance was tested across different benchmarks

Results:
There is a general trend of improvement when adding more execution units. However, across all benchmarks, the improvement reduces when adding more ALU execution units.

It is likely that the benchmark tests are limited by the branching of the superscalar units, hence in the future, further improvements can be achieved by allowing multiple branch units