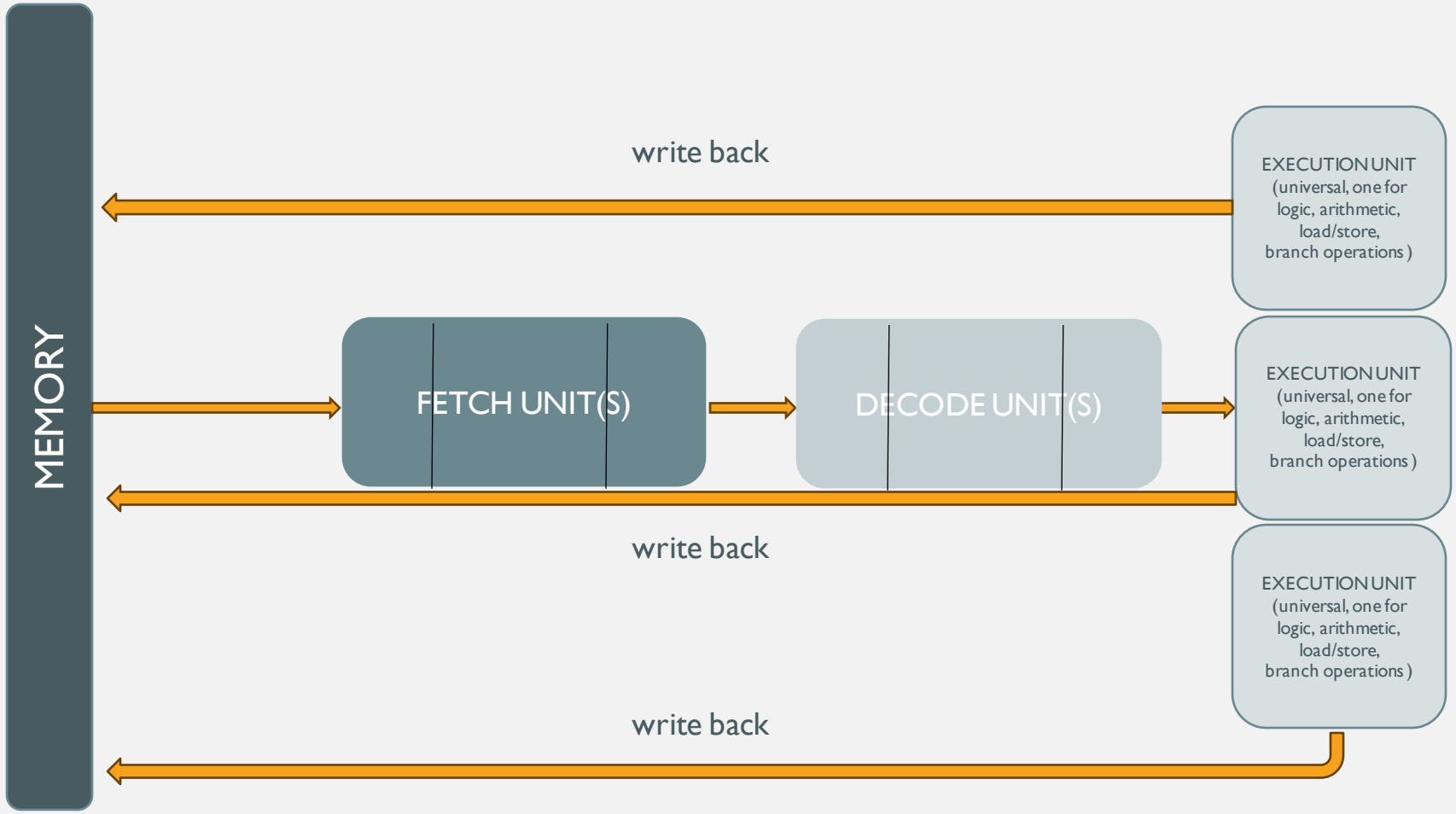
ARCHITECTURE



FEATURES

- 3 modes: the simulation can act as either a simple, scalar, serial CPU, a pipelined CPU and a 3-way superscalar CPU
- 3 stage pipeline:
  - Fetch
  - Decode
  - Execute
- Registers and Memory: 32 registers, memory size is 1024, program instructions are loaded to the front, from memory location at index 0, values from the back, from index 1023
- Dependencies and Hazards: handled via flushing and stalling by inserting NOP instructions. The superscalar processor is unable to handle control dependencies

BENCHMARKS

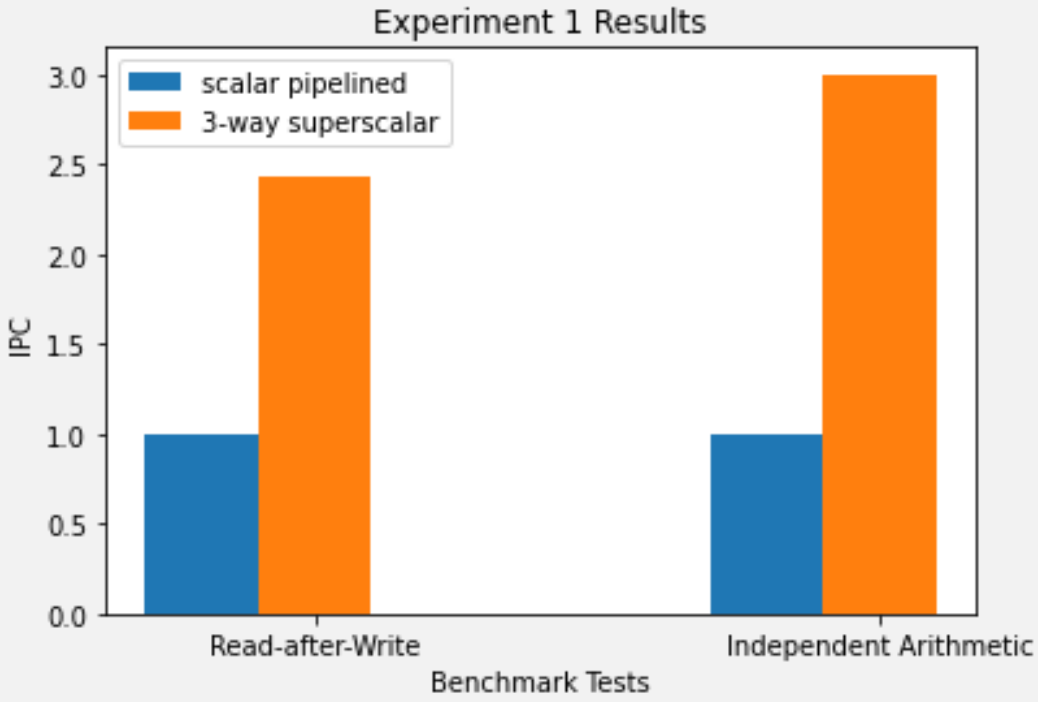
Benchmark	Description
Bubble sort	Iterative bubble sort on 5 numbers in place
Fibonacci	Calculates the Fibonacci sequence up to 10 and stores it in memory
Independent Arithmetic Operations	Loads values from memory and performs additions without dependencies between them
Data Dependent Operations	Loads values from memory and performs additions with read-after-write dependencies between them, stores some of the values in memory, also with dependencies

EXPERIMENT 1

Hypothesis: A 3-way superscalar CPU executes up to 3 times more instructions per cycle than a scalar pipelined CPU.

Experiment: I measure the number of instructions executed per cycle when executing a benchmark test with arithmetic instructions with frequent Read-after-Write and Write-after-Write data dependencies, as well as one without, meaning containing only independent arithmetic and load/store instructions. This shows how the two perform in two contrasting conditions. The reason the experiment is limited to data dependencies is the incompleteness of the superscalar CPU implementation, which cannot handle control dependencies.

Result: The superscalar processor at its best can execute 3 instruction per cycle, when a scalar one can execute only one, therefore affirming the hypothesis.



EXPERIMENT II

Hypothesis: A pipelined 3 stage CPU increases the number of instructions executed per cycle up to 3-fold compared to a non-pipelined one.

Experiment: I measure the number of instructions executed per cycle when executing a number of benchmark tests: bubble sort, calculating the fibonacci sequence, arithmetic operations with dependencies as well as independent ones. These benchmark tests exercise reading from and writing to memory and registers, arithmetic operations, and conditional instructions, giving a more comprehensive view of the CPUs' performance.

Result: a serial processor always executes one instruction every 3 cycles, as the sequence of the instructions it operates on does not get disrupted by stalling. Since it handles one at a time, stalling is not necessary. While the pipelined CPU might not execute any instructions in some cycles, it can execute up to one per cycle, both affirming the hypothesis and averaging to more than the serial CPU.