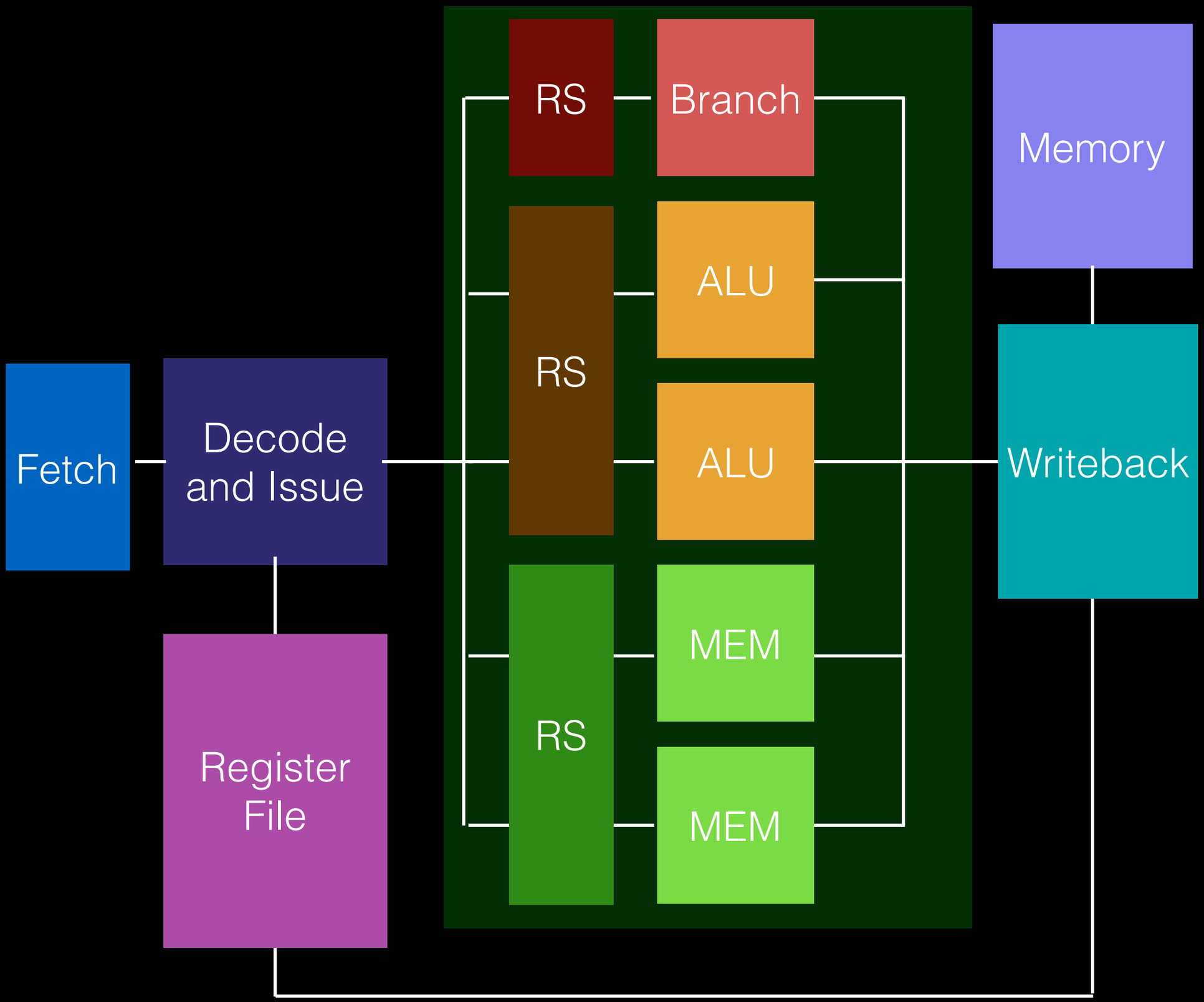
# Superscalar Processor

Iman Malik

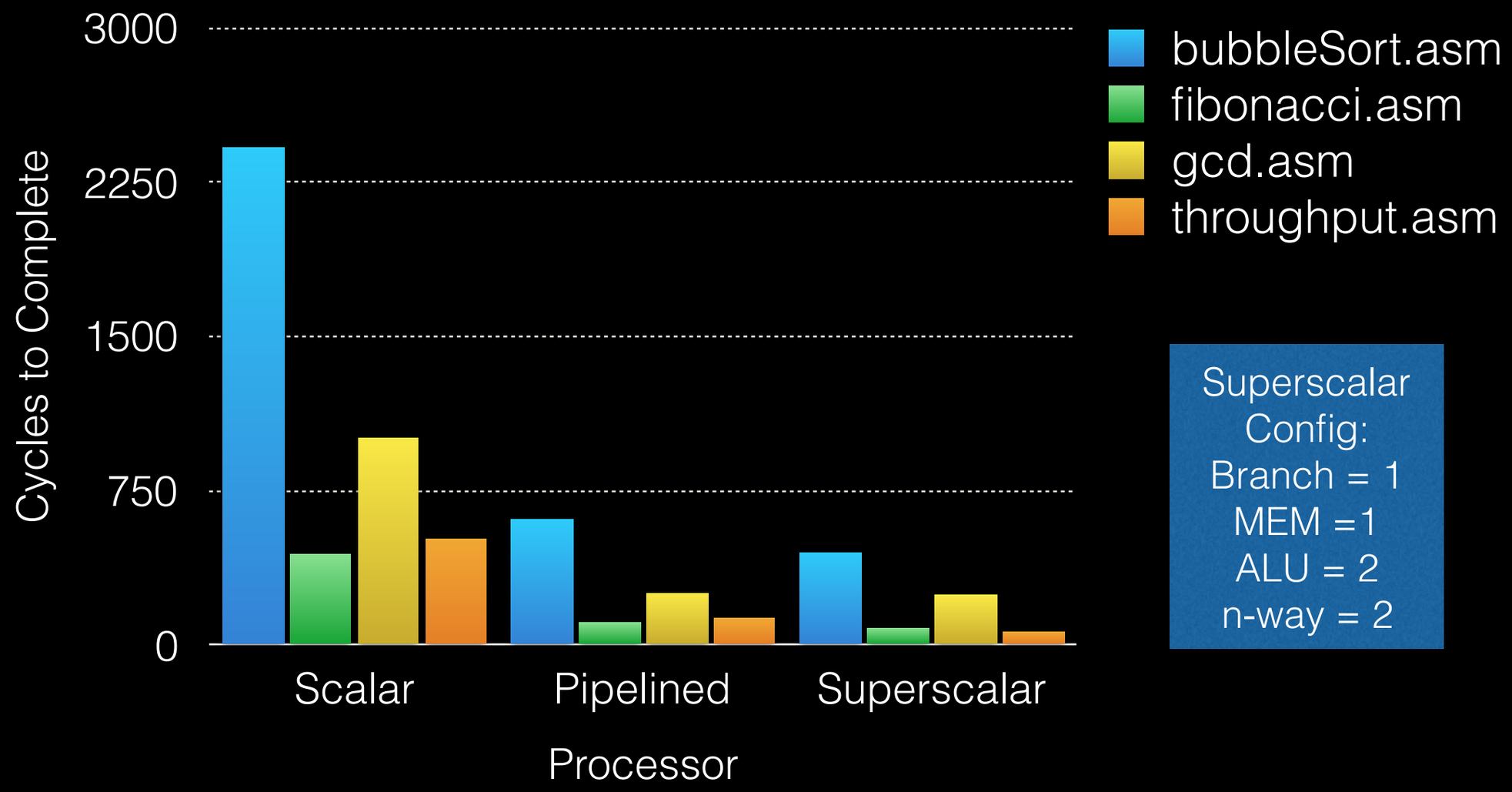
# Processor Architecture



# Processor Features

- 4-way superscalar out-of-order execution
- Pipelined Execution with 4 stages
    - Fetch, Decode, Execute and Writeback
- Multiple ALUs and Memory units and single Branch unit
- Reservation station for each unit type (Tomasulo's)
- Static Branch Prediction
    - Always taken
- Blocking issue with Branch instructions

# Hypothesis 1:
## Performance of processor increases from scalar to pipelined to superscalar

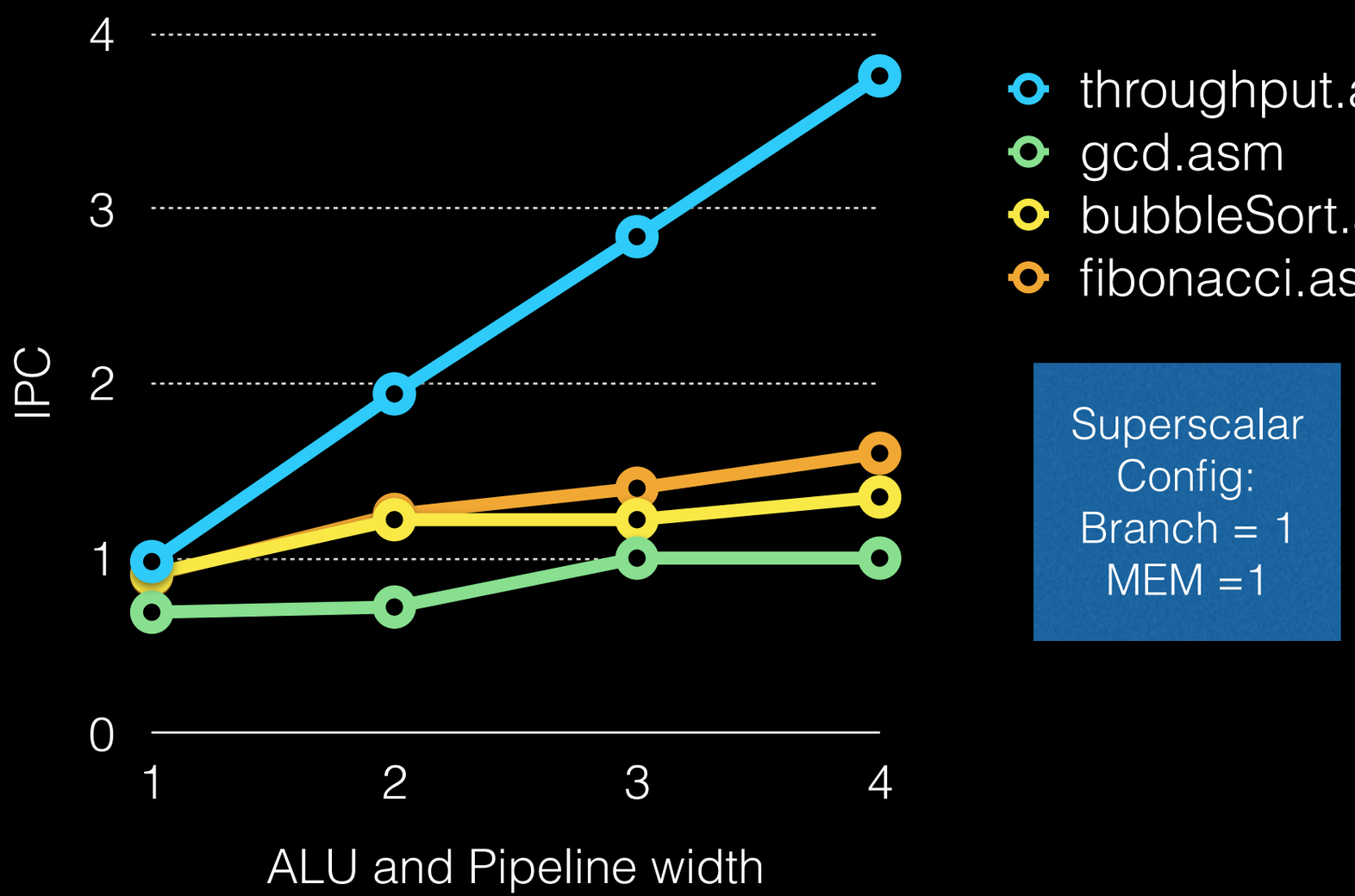



Result:

From left to right, the performance of the processor improves as less cycles were taken to complete the programs due to the increase of instructions issued per cycle.

# Hypothesis 2:

Performance when there is an increase in pipeline width and number of ALUs.

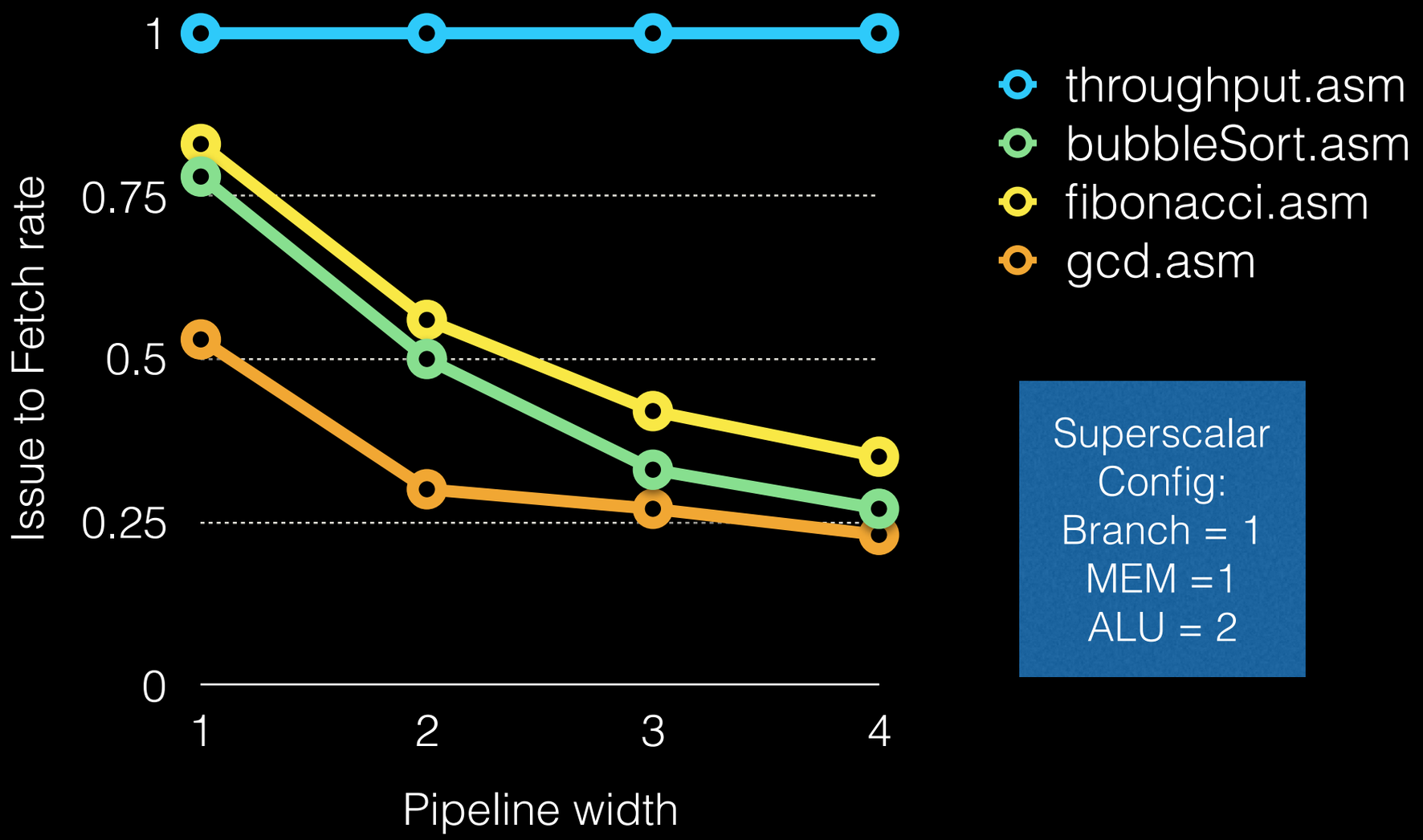



## Result:

The superscalar processor has almost achieved N instructions per cycle. The rest of the programs only showed small increases in their IPC rates due to the fact that they contain a large number of loads and stores.

# Hypothesis 3:

Issue to Fetch ratio should decrease as pipeline width increases when executing programs with loops.




## Result:

Due to there being more instructions being discarded and retrieved in the fetch stage, the issue to fetch ratio decreases with a larger pipeline. Throughput.asm has a ratio of 1 due to there being no branches. With more branches present, instructions must be discarded if a branch is taken which is why the rest of the programs have ratios below 1 with a pipeline width of 1.