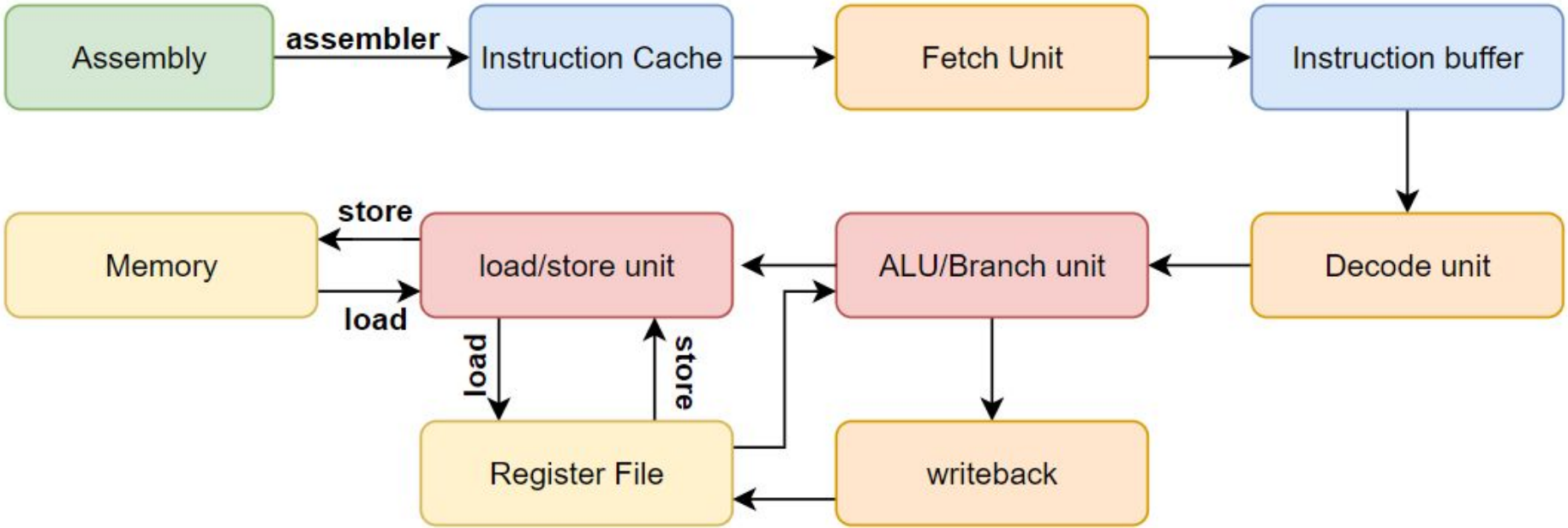
# Simple non-pipelined scalar processor simulator

Leo Lai
um21226@bristol.ac.uk

# Architecture diagram

** currently load/store instructions pass through the ALU first but this will change so that the decode unit will put instructions directly into the LSU's reservation station. **



# Current ISA

**Instruction set: (inspired by MIPS)**
ADD
ADDI (add immediate)
SUB
MUL
DIV
CMP
LD (R1 ← MEM[R2 + offset])
LDI (load immediate)
ST (MEM[R2 + offset] ← R1)
BEQ (branch to immediate/label)
BNE (branch to immediate/label)
BLT (branch to immediate/label)
BGT (branch to immediate/label)
J (relative branching)
B (branching directly to immediate/label)
HALT

**Isa features:**
Fetch 1 instruction
Decode 1 instruction
Execute 1 instruction
Writeback 1 result to registers

**Execution units:**
ALU/Branch
LSU

**Benchmark kernels:**
Bubble sort, cycles: 4352
Factorial, cycles: 152
Greatest common divisor, cycles: 140

# Experiments I plan to do (performance on benchmark kernels)

1. Performance with superscaling vs without
2. Performance With vs without reservation stations
3. Reservation stations with and without bypass
4. Comparing performance with and without branch prediction
5. Comparing the performance of different types of branch prediction
6. Performance when increasing the number of execution units (ALU/LSU/branch units)