Mohamed A. Mohamed 900182267

Nada El-Shenawy 900182782

Computer Architecture

Project 2 Report

Project Description

We have implemented a program that shows the performance of an out-of-order 16-bit RISC processor that uses Tomasulo’s algorithm without speculation.

Simulator stages

* Issue

Before issuing an instruction, we check if the destination register is ready, in order to avoid WAW hazards. We also check if there is an available Reservation Station for this instruction’s functional unit. If rd is ready and a reservation station is available, we issue in the current clock cycle. If not, we issue after the write back cycle of the WAW hazard, or when the first reservation station becomes available.

* Execute

Before starting execution, we need to make sure that both source registers are ready. If not, we wait for all of them to be ready, then we start executing. Instruction execution ends after the number of cycles specified by the user for this functional unit.

* Write back

Instructions are written back in the following cycle after execution ends.

Notes

* Register x0 cannot be overwritten and always holds the value zero.
* Register x0 is not taken into consideration for any of the hazards, as we already know its value.
* The first instruction after a taken branch is issued but not executed.
* The instruction after a JALR or RET instruction is issued in the clock cycle the JALR/RET instruction has completed execution.

Bonus features

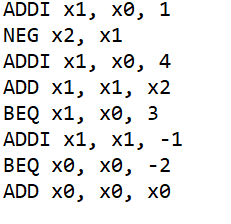
Our simulator supports a variable hardware organization. The user specifies the number of reservation stations for each functional unit. Additionally, the user will specify the number of cycles needed by each functional unit type.

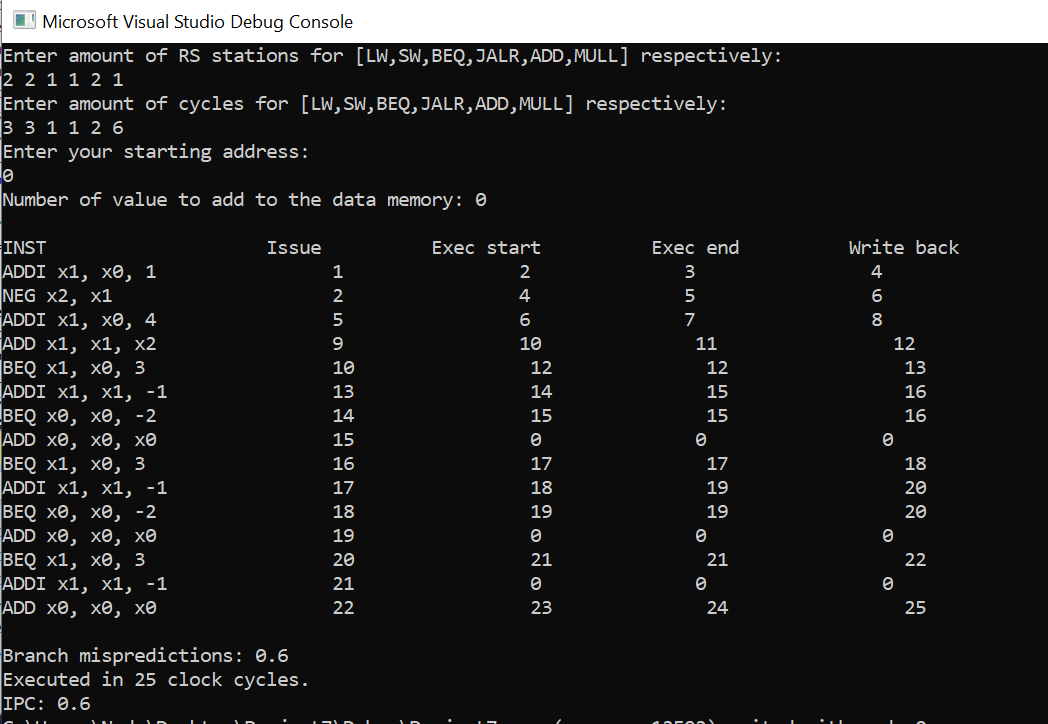
Simulator Test Cases

Test 1 –

Test 2 – Loop:

Assembly code:





Test 3 – Dependencies: