Microprocessors Report (Applied Project)

Team: micros

Pola Mawaad Aziz Saleeb **43-17505**
Mahmoud Abdrabo **43-13112**
Mohamed Ahmed Helmy **43-9565**
Omar Abdelhamid Ismael Youssef Altobgy **43-12268**

Assumptions :

- **Our algorithm works only on FP instructions.**

- **Loads' and Stores' offset are always ready (Since we only accept FP instructions, so we can't operate on the offset which has to be a general-purpose register).**

- **Add station contains 3 entries.**
- **Multiplier station contains 2 entries.**
- **Load buffer contains 3 entries.**
- **Store buffer contains 3 entries.**

- **You may change how many cycles an instruction type takes to execute (Will show where to change it at the beginning of the demo).**

Project programming language :

- **Java**

The project is divided into 7 main classes :

1- Computer (aka. The master class that runs our program).
2- Instruction (aka. The instruction table).
3- RegisterFile (aka. The register file).
4- Addstation (aka. The adder reservation station).
5- Mulstation (aka. The multiplier reservation station).
6- LoadBuffer (aka. The load buffer).
7- StoreBuffer (aka. The store buffer).

Program flow :

- **Each cycle we check if an instruction can be issued (There is an empty slot ready for that instruction in its suitable station). If so, then we issue and add it to its station.**

- **When an instruction gets added to a station, we check for any dependencies by looping over the register file.**

- **If there are no dependencies, the instruction is ready to be executed in the next cycle.**

- **When it's time to start executing, we first check if there is no instruction that uses the same component and has started executing at this cycle.**

- **When an instruction finishes executing, it's ready to write at the next cycle.**

- **When it's time to start writing, we first check if there is no instruction that is writing at this cycle.**

- **When an instruction finishes writing in the register file, we check if there is any instruction that is waiting for the result and update its value.**

- **If an instruction got any of its values updated, we check if it's ready to execute or not.**

- **The program ends when all instructions have finished writing or if it's a store instruction then it finishes when it finishes executing.**