Report

Steps

1. Requiremnt Phase, reading the project requirements and discuss it with team mates.
2. Design Phase

- Deciding on Tech/Tools to use.
- Desinging the archeticture of the software.

3. Planning Phase

- Studying the tomasulo algo
- splitting the tasks to the team

4. Develpoment Phase

- Implementing a client server archteitcutre Using React & Flask.
- Implementing a fancy cli using logging techinques and cli argument parsing and configuration using yaml

5. Testing Phase

- The testing phase was includeded in the dev phase where you followed a behivoural driven develpomenet where we first write test scinarios and then write code to implement them
- Finally an end to end system testing was applied where insursing every component is integrating correctly and the whole logic is OK.

Configurations

Type	Size
`ADD/SUB Reservation Station`	`3`
`MUL/DIV Reservatoin Station`	`2`
`Load Buffer`	`3`
`Store Buffer`	`3`

How to run

GUI

```
> git clone https://github.com/aboueleyes/tomasulo
> cd tomasulo
> cd server && pip install -r requirements.txt && python3.10 server.py && cd -
> cd client && npm i && npm start
```

CLI

```
> git clone https://github.com/aboueleyes/tomasulo
> cd tomasulo
> cd server && pip install -r requirements.txt && python3.10 main.py
```

ShowCase

CLI

GUI

Approach

Our Approach for the algo is to simulate what actullay happens in a hardware by simaulating every component by class and include its logic in the class.

A singalton Design Pattern were used to further simulate the hardware.

Code Structure

```bash= . ├── client │ ├── package.json │ ├── package-lock.json │ ├── public │ │ ├── favicon.ico │ │ ├── index.html │ │ ├── logo192.png │ │ ├── logo512.png │ │ ├── manifest.json │ │ └── robots.txt │ ├── README.md │ └── src │ ├── App.css │ ├── App.js │ ├── App.test.js │ ├── components │ │ ├── BasicTable.js │ │ ├── FileEditor.js │ │ ├── Navbar.js │ │ ├── SideTabs.js │ │ └── Tabs.js │ ├── index.css │ ├── index.js │ ├── logo.svg │ ├── pages │ │ ├── Setup.js │ │ └── Simulation.js │ ├── reportWebVitals.js │ ├── setupTests.js │ └── utils │ └── validateFormData.js ├── LICENSE ├── README.md └── server ├── config.yml ├── main.py ├── requirements.txt ├── sample-instructions.txt ├── server.py └── src ├── buffer.py ├── components.py ├── instruction.py ├── instructions_parser.py ├── reservation.py └── tomasulo.py
```

Test Cases

```asm
L.D F1 0
L.D F2 1
MUL.D F1 F2 F1
MUL.D F1 F1 F1
MUL.D F3 F1 F1
```

Type	Latency
L.D	1
MUL.D	1

```
L.D F1 0
L.D F2 1
MUL.D F2 F2 F2
MUL.D F1 F2 F1
MUL.D F1 F1 F1
MUL.D F3 F1 F1
ADD.D F3 F2 F2
```

Type	Latency
L.D	1
MUL.D	2
ADD.D	1

```
L.D F6 90
L.D F2 80
MUL.D F0 F2 F4
SUB.D F8 F2 F6
DIV.D F10 F0 F6
ADD.D F6 F8 F2
```

Type	Latency
L.D	2
MUL.D	10
ADD.D	2
SUB.D	2
DIV.D	40

```
L.D F1 10
MUL.D F1 F1 F1
S.D F1 15
SUB.D F1 F2 F2
```

Type	Latency
L.D	1
MUL.D	4
SUB.D	2
S.D	1