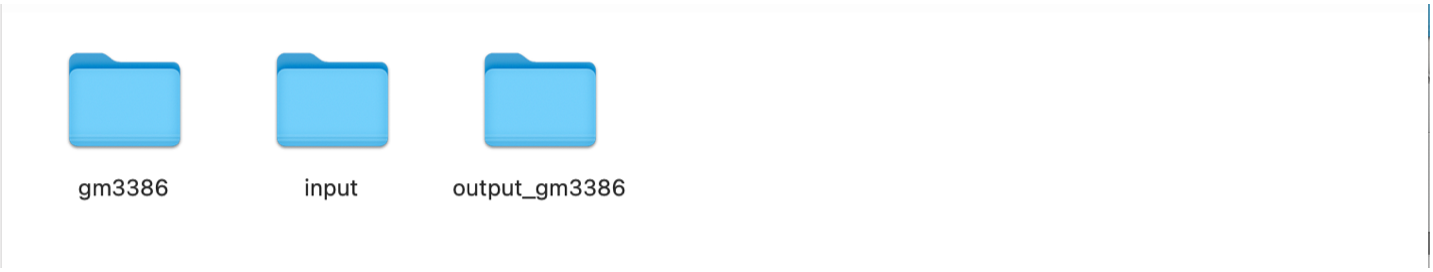
CSA Project phase 2
Giorgi Merabishvili
gm3386

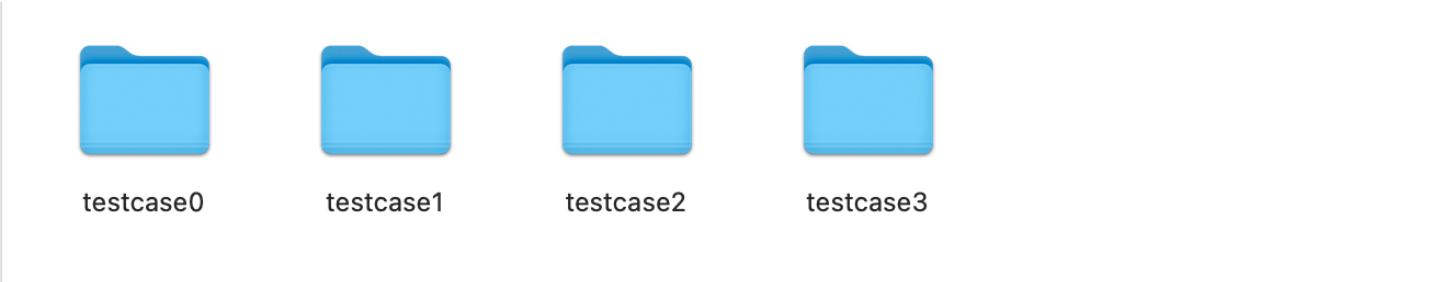
Complete code is inside the folder gm3386 named as “main.py”.

The language used for the project is python.

In order to run the code, the following structure of files should be followed, and no other external libraries need to be installed.



Inside gm3386 the main.py is present and inside Input, the testcases are present as shown below



terminal ss of running the code for different test cases.

```
SA_Project/gm3386/main.py
IO Directory: /Users/giorgi/Desktop/CSA_Project/input/testcase3
IO Directory: /Users/giorgi/Desktop/CSA_Project/input/testcase2
IO Directory: /Users/giorgi/Desktop/CSA_Project/input/testcase0
IO Directory: /Users/giorgi/Desktop/CSA_Project/input/testcase1
(base) 10-18-188-213:CSA_Project giorgi$
```

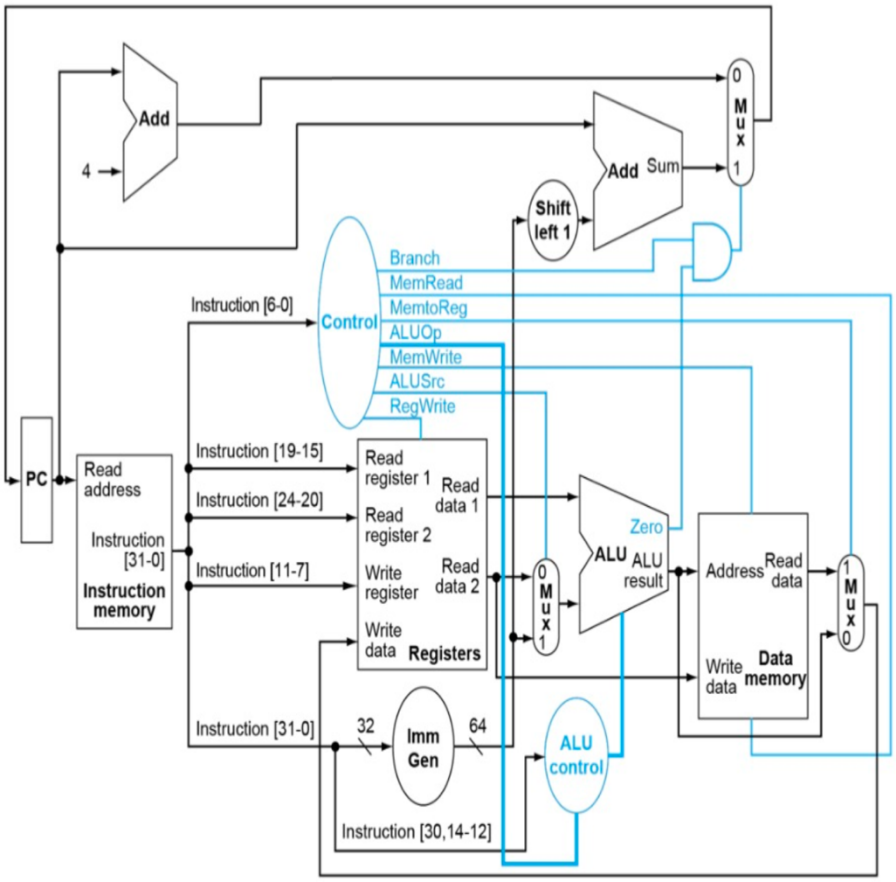
Running the program:

go to project folder in terminal and Run the command:

⇒ python net_id/main.py

main.py file is in gm3386 folder

1)Draw the schematic for a single-stage processor and fill in your code to run the simulator.

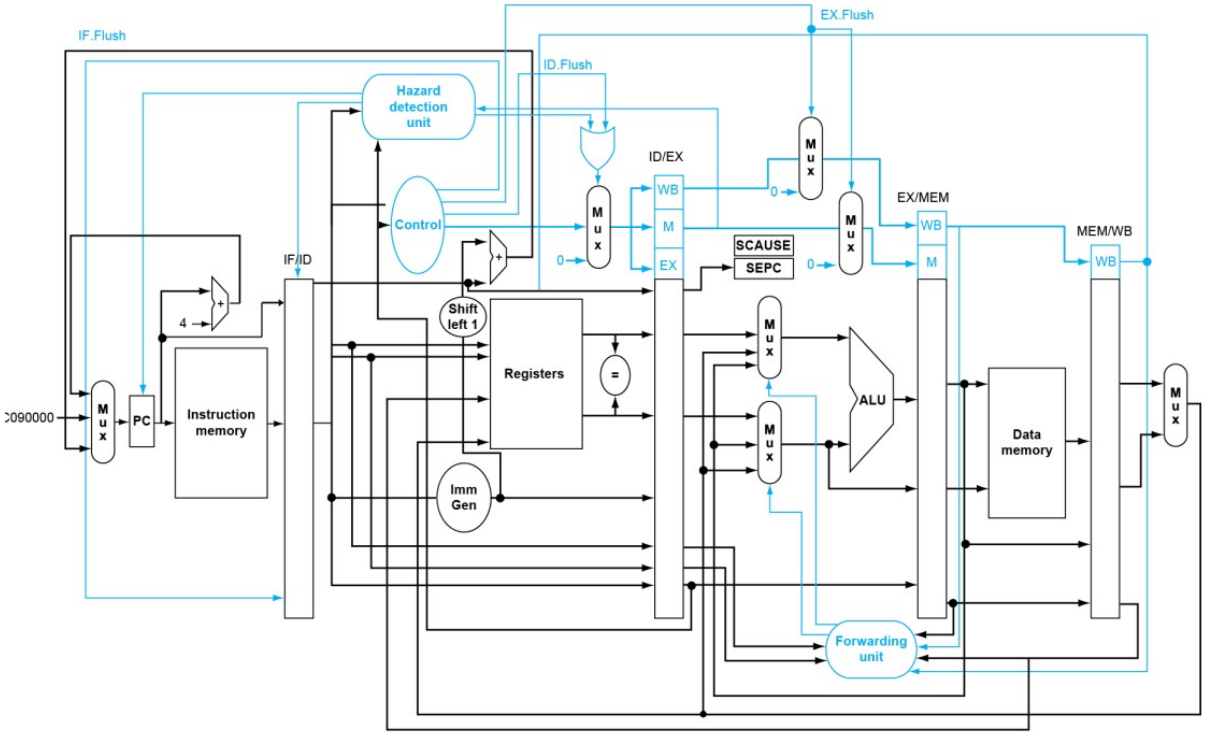


*Source: Class Slides

- The diagram depicts a simplified view of the datapath of a MIPS processor, detailing the critical components involved in instruction processing. The Arithmetic Logic Unit (ALU) is

at the heart of the architecture, responsible for performing arithmetic and logical operations. Instructions are read from the instruction memory, directing the control unit to generate appropriate signals for directing processor operations. The Program Counter (PC) ensures that instructions are executed in the correct order by storing the address of the current instruction and preparing the address for the next one. Multiplexers (MUX) are critical components of conditional decision-making and data flow control. Registers are used to store immediate values and intermediate results, whereas data memory is used for data handling read and write operations. To facilitate decoding and execution, the instruction's various components, such as the operation code, source register identifiers, and immediate values, are routed to their respective units.

2) Draw the schematic for a five stage processor and fill in your code in the provided file to run the simulator.



*Source: Class Slides

- The diagram depicts a MIPS processor with advanced pipeline management, including hazard detection and resolution mechanisms. The hazard detection unit detects potential instruction execution conflicts that could disrupt the computational flow. To mitigate these risks, a forwarding unit is used, which allows data to bypass certain pipeline stages while maintaining instruction processing continuity. Flush controls, labeled IF.Flush, ID.Flush, and EX.Flush, are used to clear pipeline segments during

branch mispredictions or instruction cancellations, ensuring program execution accuracy. SCAUSE and SEPC are specialized program counters that indicate the processor's ability to handle exceptions and interrupts, which is critical for responding to unexpected events. These components work together to optimize instruction throughput while also ensuring the integrity of operations within the processor's architecture.

The final code is available in gm3386/main.py

3) Measure and report average CPI, Total execution cycles, and Instructions per cycle by adding performance monitors to your code. Make sure you output these values to a file.

Testcase 0:

Single Stage Core Performance Metrics

Total execution cycles = 6

Average CPI = 1.2

Instructions per cycle = 0.833333

Five Stage Core Performance Metrics:

Number of cycles taken: 10

Cycles per instruction: 2.0

Instructions per cycle: 0.5

```
Single Stage Core Performance Metrics----------------------------
Number of cycles taken: 6
Cycles per instruction: 1.2
Instructions per cycle: 0.833333

Five Stage Core Performance Metrics----------------------------
Number of cycles taken: 10
Cycles per instruction: 2.0
Instructions per cycle: 0.5
```

Testcase 1:

Single Stage Core Performance Metrics

Total execution cycles = 40

Average CPI = 1.02564

Instructions per cycle = 0.975001

Five Stage Core Performance Metrics

Number of cycles taken: 46

Cycles per instruction: 1.17949

Instructions per cycle: 0.847824

```
Single Stage Core Performance Metrics----------------------------
Number of cycles taken: 40
Cycles per instruction: 1.02564
Instructions per cycle: 0.975001

Five Stage Core Performance Metrics-----------------------------
Number of cycles taken: 46
Cycles per instruction: 1.17949
Instructions per cycle: 0.847824
```

Testcase 2:

Single Stage Core Performance Metrics

Total execution cycles = 7

Average CPI = 1.75

Instructions per cycle = 0.571429

Five Stage Core Performance Metrics

Number of cycles taken: 10

Cycles per instruction: 2.5

Instructions per cycle: 0.4

```
Single Stage Core Performance Metrics-----------------------------
Number of cycles taken: 7
Cycles per instruction: 1.75
Instructions per cycle: 0.571429

Five Stage Core Performance Metrics-----------------------------
Number of cycles taken: 10
Cycles per instruction: 2.5
Instructions per cycle: 0.4
```

Testcase 3:

Single Stage Core Performance Metrics

Number of cycles taken: 28

Cycles per instruction: 2.54545

Instructions per cycle: 0.392858

Five Stage Core Performance Metrics

Number of cycles taken: 38

Cycles per instruction: 3.45455

Instructions per cycle: 0.289473

```
Single Stage Core Performance Metrics-----------------------------
Number of cycles taken: 28
Cycles per instruction: 2.54545
Instructions per cycle: 0.392858

Five Stage Core Performance Metrics-----------------------------
Number of cycles taken: 38
Cycles per instruction: 3.45455
Instructions per cycle: 0.289473
```

4) What optimizations or features can be added to improve performance? (Extra credit)

To improve processor performance, the following optimizations or features can be added:

- Resource constraints cause structural hazards when multiple processes must be carried out sequentially rather than concurrently. Memory or register file port restrictions during multi-cycle processes could be the cause of these. This can be fixed by increasing the processor's resources, speeding up the circuitry, or improving the pipelining procedure.

- Processor performance can be improved by increasing the number of pipeline steps or the depth of pipelining. As a result, each stage's logic processing will be reduced, improving clock rate and throughput.

- Branch misprediction latency is a major factor affecting processor performance decrease. This issue can be solved by replacing the branch instructions with arithmetic or lookup operations, which avoids adding control flow hazards.

- Obtaining nearly perfect balance among the five stages in order to alleviate time constraints caused by the slowest phase.

- Reducing the amount of overhead required between stages for the storage and retrieval of critical data.

5) Compare the results from both the single stage and the five stage pipelined processor implementations and explain why one is better than the other.

Metric	Single Stage Processor	Multi Stage Processor
Execution time	Higher Execution Time	Lower Execution Time
Throughput	Low Throughput	High Throughput
Hazards	No hazards will occur	Control, data or structural hazards may occur
Cycles	Low total no of cycles	High total no of cycles
CPI and IPC	Low CPI, High IPC	High CPI, Low IPC
Execution	One instruction per cycle is executed	Multiple instruction are executed parallelly in a single cycle

Advantages of a single-stage processor over a five-stage pipelined processor:

- Because there is no resource conflict or inaccurate memory access with a single-stage processor, there are no structural, control, or data hazards.

- There is no need for data storage and access between stages.

Advantages of a five-stage pipelined processor over a single-stage processor:

- The presence of five pipeline stages reduces the amount of logic processed, resulting in a faster clock rate and higher processor throughput.

- The cycle time of a single-stage processor is determined by the total time required to process each instruction sequentially. If m instructions take n seconds each, the total processing time is (m * n) seconds. In a five-stage processor, however, the cycle time is influenced by the slowest stage, roughly (m * n)/5, because there are 5 stages. As a result, even with the slowest stage being (m * n)/5 seconds, it is faster than the single-stage processor.

For example, in testcase 1:

Assuming that the execution time of a single-stage processor is n seconds, the execution time of a five-stage processor with 5 stages is n/5 seconds.

Metric	SingleStage Processor	MultiStage Processor
Execution time	40*n seconds	46 * n seconds
Instruction(approx)	40	31
Cycles	40	46
IPC	0.9750009	0.847824
CPI	1.02564	1.17949