MICRO PROCESSORS PROJECT REPORT

SUBMITTED BY:Samar Ahmed Shokry 43-176

Omar Salah Ads 43-12699

Ahmed Ossama 43-3823

Team:LATE

Main CODE Components:

Classes:

Reservation Station:

```
String Op; //Operation type "ADD","SUB","LW","SW","MULT","DIV"
Boolean Busy;
float Vj;
float Vk;
float res;
int Qj;
int Qk;
int Address;
int cyclesleft;
     // ADD/SUB 3 cycles
    // MULT/DIV 4 cycles
    // Store/Load 2 cycles
```

Instruction Unit:

```
    String Ins; //Operation Name "ADD","SUB","LW","SW","MULT","DIV"

    int des;
    int op1;
    int op2;
```

Register File:

```
int Qi;
float content;
```

Bus:

```
int Tag;

 float value;
```

Simulation Class:

```
    Bus CDB;
    ReservationStation[] Station;
    //Array of 12 Stations for all stations
    // 0-2 ADD/SUB
    // 3-5 MULT/DIV
    // 6-8 LW
    // 9-11 SW
    int[] GeneralRegisters; //Array of 10 General Purpose Registers

    float[] MemoryData; //Array of 100 Memory Data values
    Boolean Writing;    //State of writeback stage
    ArrayList<InstructionUnit> instructions; //Array of fed instructions
    RegisterFile[] RegisterFile; //Array of 5 RegisterFiles
```

Code Logic:

The code logic consists of a while loop with 3 main stages:

Issue Stage:

1 instruction is fetched from the InstructionUnit array to be saved into a station from the Station array,by checking its op1 and op2 field and comparing it to the registerfile array if the register op1 or op2 was not ready meaning Qi was -1 meaning some station will be writing into this registerfile then the Qj/Qk field will be updated to be the value of the register Qi.If Qi was found to be -1 meaning the content of the register is fresh and ready to use then the Vj/Vk fields are updated with the content of the register.Also if the instruction was LW or SW the same check is done for the des field but only updated in Qj or Vj according to the above logic and the address field is calculated by adding the op1 value to the generalpurpose at index op2 to be address in memory that will be later stored to or loaded from(All stations have the same components for simplicity reasons however only correct fields are to be updated).After the above logic the registerfile index of des field in instruction will be updated to have Qi equal to station number and the instruction state will be busy (busy=true) and now the instruction is ready to be executed.

Execute Stage:

A loop will be conducted on the station array checking if the station is busy and its Vj and Vk fields are occupied (not equal -1) meaning the instruction is ready to execute so it will start executing then .however this check is altered for the LW and SW because of having only Vi or Qi occupied at any time. For ALU operation actual logic of operation will be calculated And The value of the execution will be saved into a temporary storage integer named res to keep it saved and ready to be written back in the next cycle (just for simulation).The instruction will only execute if its cyclesleft count is greater than zero and when the cyclesleft reaches -1 then the instruction is ready to be written back.Keep in mind the address for LW and SW was already calculated in the issue stage.If either the Vj or Vk field was -1 then the instruction wont be able to start execution and will have to wait.

WiteBack Stage:

A loop will be conducted on the station array checking if the instruction cyclesleft reached zero meaning its time to write back however it must first check that no other instruction is writing back also so here comes the writing Boolean attribute that will be declared before the write back stage which will be set to true when the instruction is writing meaning this unit is busy now and no other instruction should writeback now.If the writing was false meaning the instruction can write its res then the result is written to the registerfile content field and the the Qi will return to -1 meaning the content of the register file is the newest value and can be used.In case of SW the value in the Vj will be written to memorydata array in index of address field of the station and for LW the memorydata at index of address field will be loaded and written in the content field of register at index of Vj.Finally the station will be reseted indicating that this station is idle and ready to be used once again.

Main Loop:

The three stages will be run simultaneously in a while loop however you will see in the code that the issue stage comes at last to prevent an instruction from being issued and start executing in same cycle.Also in the code sample run you will find statement saying executing in cycle zero however the station wont be executing its just indicating the station is ready to writeback to prevent the station from finishing execution and writing back in the same cycle.

Sample Run:

Sample Input:

```
("LW",4,1,2);

("SUB",2,4,2);

("MULT",1,1,3);

("ADD",1,5,3);
```

Sample output:

```
Cycle: 1 ISSUED LW
Station:6 busy:true Vj:4.0 Qj:-1 Vk:-1.0 Qk:-1 A:10 Cycles Left:2

Cycle: 2 In Execution Now Station 6 CycleNumber: 2
Cycle: 2 ISSUED SUB
Station:0 busy:true Vj:-1.0 Qj:7 Vk:7.0 Qk:-1 A:0 Cycles Left:3

Cycle: 3 In Execution Now Station 6 CycleNumber: 1
Cycle: 3 ISSUED MULT
Station:3 busy:true Vj:6.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:4

Cycle: 4 In Execution Now Station 3 CycleNumber: 4
Station:3 busy:true Vj:6.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:4
Cycle: 4 In Execution Now Station 6 CycleNumber: 0
Cycle: 4 Writing Now Station:
Station:6 busy:true Vj:4.0 Qj:-1 Vk:-1.0 Qk:-1 A:10 Cycles Left:0
Station: 6 Reseted
Station:0 busy:true Vj:41.0 Qj:-1 Vk:7.0 Qk:-1 A:0 Cycles Left:3
Cycle: 4 ISSUED ADD
Station:1 busy:true Vj:10.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:3

Cycle: 5 In Execution Now Station 0 CycleNumber: 3
Station:0 busy:true Vj:41.0 Qj:-1 Vk:7.0 Qk:-1 A:0 Cycles Left:3
Cycle: 5 In Execution Now Station 1 CycleNumber: 3
Station:1 busy:true Vj:10.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:3
Cycle: 5 In Execution Now Station 3 CycleNumber: 3
Station:3 busy:true Vj:6.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:3

Cycle: 6 In Execution Now Station 0 CycleNumber: 2
Station:0 busy:true Vj:41.0 Qj:-1 Vk:7.0 Qk:-1 A:0 Cycles Left:2
Cycle: 6 In Execution Now Station 1 CycleNumber: 2
Station:1 busy:true Vj:10.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:2
Cycle: 6 In Execution Now Station 3 CycleNumber: 2
Station:3 busy:true Vj:6.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:2

Cycle: 7 In Execution Now Station 0 CycleNumber: 1
Station:0 busy:true Vj:41.0 Qj:-1 Vk:7.0 Qk:-1 A:0 Cycles Left:1
Cycle: 7 In Execution Now Station 1 CycleNumber: 1
Station:1 busy:true Vj:10.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:1
Cycle: 7 In Execution Now Station 3 CycleNumber: 1
Station:3 busy:true Vj:6.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:1

Cycle: 8 In Execution Now Station 0 CycleNumber: 0
Station:0 busy:true Vj:41.0 Qj:-1 Vk:7.0 Qk:-1 A:0 Cycles Left:0
Cycle: 8 In Execution Now Station 1 CycleNumber: 0
Station:1 busy:true Vj:10.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:0
Cycle: 8 In Execution Now Station 3 CycleNumber: 0
Station:3 busy:true Vj:6.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:0
Cycle: 8 Writing Now Station:
Station:0 busy:true Vj:41.0 Qj:-1 Vk:7.0 Qk:-1 A:0 Cycles Left:0
Station: 0 Reseted
```

Cycle: 9 Writing Now Station:
Station:1 busy:true Vj:10.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:0
Station: 1 Reseted
Cycle: 10 Writing Now Station:
Station:3 busy:true Vj:6.0 Qj:-1 Vk:8.0 Qk:-1 A:0 Cycles Left:0
Station: 3 Reseted