James Bickerstaff & Jefferson Boothe

**ECE 2162 P1 Report**

**Section 1**

* The code is written in python, and must be executed on a system in which it is installed (Windows OS, Linux OS, WSL, etc)
* The name of the main file that will be run is “tomasulo.py”, and all supporting code is located within the same repository
* To execute the code, run the following command:

python3 tomasulo.py \*instruction file name\* \*configuration file name\*

* The output of the program will be sent to the terminal window, including the ARF values, memory values, and timing information

**Section 2**

Initializing FUs to the following spec:

Using Integer Adder with 2 reservation stations, 1 cycle in EXE, and 1 FU

Using Floating Point Adder with 3 reservation stations, 3 cycles in EXE, and 1 FU

Using Floating Point Multiplier with 2 reservation stations, 20 cycles in EXE, and 1 FU

Using Load/Store Unit with 250 (infinite) reservation stations, 1 cycle in EXE, 4 cycles in MEM, and 1 FU

Using an instruction buffer of size 12

Using a ROB of size 128

Using 1 CDB with buffers of size 1 per FU in the event of multiple instructions trying to WB

**Tier 1: Straight-line cases where no dependencies exist among instructions**

Test 1

Instructions and initial values:

|  |  |
| --- | --- |
| LD F2,4(R0) | R1 = 0, R2 = 20, R3 = 16, R4 = 95 |
| ADD R1,R2,R3 | F1 = 0, F2 = 0, F3 = 6, F4 = 6, F5 = 1, F6 = 2 |
| SUB.D F3,F4,F5 | F7 = 8, F8 = 22, F9 = 39 |
| MULT.D F6,F7,F8 | MEM[4] = 19, MEM[8] = 0 |
| SD F9,8(R0) |  |
| ADDI R4,R4,5 |  |

Output:

A picture containing calendar

Description automatically generated

**Tier 2: Straight-line code where there are dependencies (false and true):**

Test 1

Instructions and initial values:

|  |  |
| --- | --- |
| SUB.D F1,F2,F3 | F1 = 0, F2 = 20.5, F3 = 7.5, F4 = 0, F5 = 5 |
| ADD.D F1,F2,F3 | F6 = 0, F7 = 0, F8 = 12 |
| MULT.D F4,F1,F5 | MEM[4] = 19, MEM[8] = 0 |
| ADD.D F6,F4,F5 |  |
| LD F7,4(R0) |  |
| ADD.D F7,F7,F8 |  |
| SD F7,8(R0) |  |

Output:

A picture containing text

Description automatically generated

**Section 3**

James Bickerstaff

Major portions of code written: Branch Predictor, Load Store Unit, int/fp ARF, RAT

Test benchmarks written: case1\_noDeps, case2\_deps, LDchain, LDSD chain, SD chain, 4\_loop\_iterations

Debugging Effort: worked to debug the components written and related integration issues

Participation Effort: contributed significantly and completed/tested components written

Jefferson Boothe

Major portions of code written: ROB, int/fp Adder, fp multiplier, CDB, instruction buffer

Test benchmarks written: case3\_forward, case4\_hazards, FloatOps, IntOps, random\_instrs

Debugging Effort: worked to debug the components written and related integration issues

Participation Effort: contributed significantly and completed/tested components written