Implemented a OOO processor Tomasulo algorithm with an 8-entry re-order buffer. The number of reservation stations is shown below. Contents of the architectural register file (ARF) are shown. We have one execution unit for Add, one for MUL/Div (not pipelined)

1. Issue and Write take one cycle each. So, if data is written to CDB in clock cycle ‘a’, the data is read from the reservation station or register in the next clock cycle ‘a+1’
2. Similarly, if execution completes in cycle ‘a’, it is written to CDB in cycle ‘a+1’
3. Latency: Add : 1 cycle. Mult: 10 cycles. Divide: 40 cycles, Load: 5 cycles

Implement a single-issue RISC-V processor which reads in the instructions shown in the table and runs the instructions out of order. Follow the RISC-V instruction encodings as in the RISC-V Specification. The MUL and DIV are part of the ‘M’ extensions of RISC-V. You can choose the number of stages in the pipeline as per the Tomasulo algorithm (4 stages) or you can implement a 7-stage pipeline.

The Tomasulo structure with RoB is shown below (CDB is an important component of the design). RAT also needs to be a part of the design.

**Three Reservation station (Add/Sub)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | Busy | Destination tag | Source tag1 | Source tag2 | Value of source 1 | Value of source 2 |
| ADD | 1 | ROB3 |  |  | 1--R7 | 2--R8 ISSUED IN 5 |
| SUB | 1 | ROB5 |  |  | 36 | R5--3 |
| ADD | 1 | ROB6 |  |  | 33 | 0 |

**Two Reservation stations (Multiplier/Divide)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | Busy | Destination tag | Source tag1 | Source tag2 | Value of source 1 | Value of source 2 |
| DIV | 1 | ROB3 |  |  | 2 | R4--5 |
| MUL | 1 | ROB4 |  |  | 12 | 3 |

**Three Load/Store buffers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instruction | Busy | Destination tag | Address offset | Source register |
| LW | 1 | ROB0 | 0 | R2--16 ISSUED TO EX IN CYCLE 2 |
|  |  |  |  |  |
|  |  |  |  |  |

**Architectural Register file (ARF)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 | R10 |
| 33 | 0 | 3 | 33 | 3 | 4 | 1 | 2 | 2 | 3 |

**Instructions**

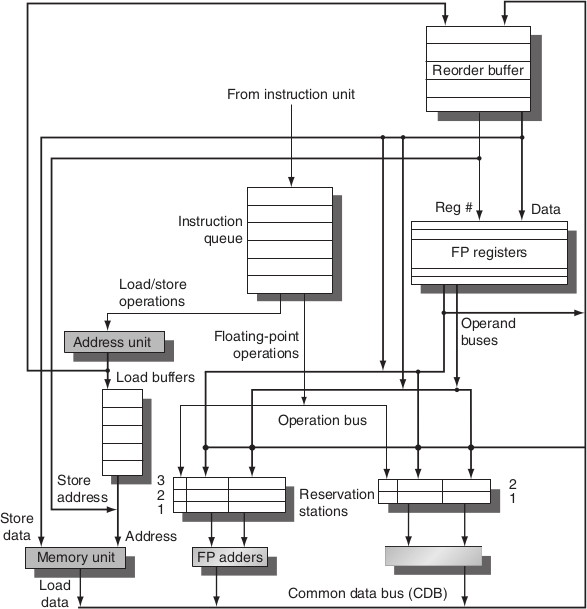
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | IF | ID-  DECODING AND SENDING TO RS | Execute start/Addres s calculation for LD | Execute/ Memory access complete | Write to CDB | Commit |
| LW R3, 0(R2) | 0 | 1 | 2 | 6 | 7 | 8 |
| DIV R2, R3, R4 | 1 | 2 | 14 | 53 | 54 | 55 |
| MUL R1, R5, R6 | 2 | 3 | 4 | 13 | 14 | 56 |
| ADD R3, R7, R8 | 3 | 4 | 5 | 6 | 7 | 57 |
| MUL R1, R1, R3 | 4 | 5 | 54 | 63 | 64 | 65 |
| SUB R4, R1, R5 | 5 | 6 | 65 | 66 | 67 | 68 |
| ADD R1, R4, R2 | 6 | 7 | 68 | 69 | 70 | 71 |

**Register allocation table (RAT)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 | R10 |
| ROB2--ROB4--ROB5 | ROB1 | ROB0--ROB3 | ROB5 |  |  |  |  |  |  |

**ROB**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Instruction type | Destination | Value |
| RoB0 | LW | R3 | 2 COMITTED IN 8 |
| RoB1 | DIV | R2 | 0 COMMITTED IN 55 |
| RoB2 | MUL | R1 | 12 COMMITTED IN 56 |
| RoB3 | ADD | R3 | 3 COMMITTED IN 57 |
| RoB4 | MUL | R1 | 36 COMMITTED IN 65 |
| RoB5 | SUB | R4 | 33 COMMITTED IN 68 |
| RoB6 | ADD | R1 | 33 COMMITTED IN 71 |
| RoB7 |  |  |  |



CA QA, H&P Ch 3.6