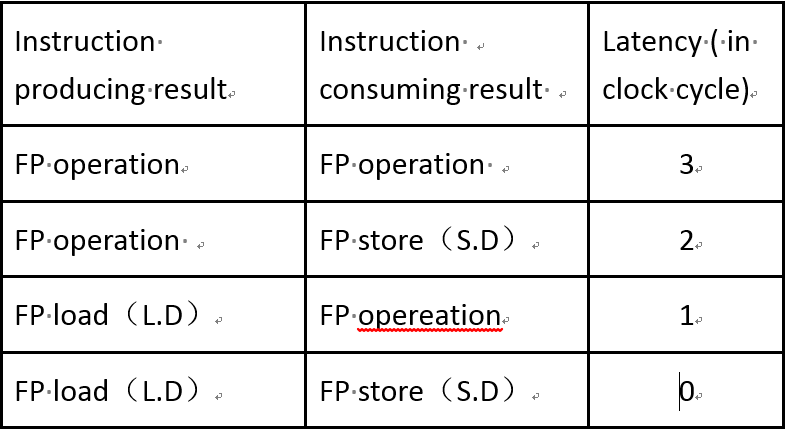
1. (50 points) Run the following code on Scoreboard with function units including two FP multipliers，one FP divider, one FP adder that can perform float add and sub operations, and two integer units that is responsible for memory accessing, integer ALU operations and branch).

(1) Fill the following tables with the scoreboard status at the end of **5th** clock cycle when the first L.D instruction just finish WB stage. The scoreboard status includes instruction status, function units status and FP register status.



L.D F0,0(R1) ; F0🡨MEM[R1+0]

L.D F4,0(R2) ; F4🡨MEM[R2+0]

MUL.D F6,F0,F4 ; F6🡨F0 \* F4

ADD.D F8,F0,F2 ; F8🡨F0 + F2

S.D F6, 0(R3) ; MEM[R3 + 0] 🡨 F6

S.D F8, 0(R4) ; MEM[R4 + 0] 🡨 F8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instructions | 发射（Issue）  clock | 读操作数(RO)  clock | 执行（EXE）  clock | 写结果(WB)  clock |
| L.D F0, 0(R1) | 1 | 2 | 3-4 | 5 |
| L.D F4, 0(R2) | 2 | 3 | 4-5 | 6 |
| MUL.D F6,F0,F4 | 3 | 4-6 |  |  |
| ADD.D F8,F0,F2 |  |  |  |  |
| S.D F6, 0(R3) |  |  |  |  |
| S.D F8, 0(R4) |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Function units | Function Unit Status | | | | | | | | | |
| Busy | Op | Fi | Fj | Fk | Qj | Qk | Rj | Rk | A |
| Integer1 | No |  |  |  |  |  |  |  |  |  |
| Integer2 | Yes | L.D | F4 | R2 |  |  |  | No |  |  |
| FPMult1 | Yes | MUL.D | F6 | F0 | F4 |  | Integer2 | Yes | No |  |
| FPMult2 | No |  |  |  |  |  |  |  |  |  |
| FPDivider | No |  |  |  |  |  |  |  |  |  |
| FPAdder | No |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Register Status | | | | | | | | |
| F0 | F2 | F4 | F6 | F8 | F10 | F12 | **……** | F30 |
| FU | MEM[R1+0] |  | Integer2 | FPMult1 |  |  |  |  |  |

(2) Fill the following table with instruction status at the end of **6th** Clock Cycle as the first line in the table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 发射（Issue） | 读操作数(RO) | 执行(EXE) | 写结果(WB) |
| L.D F0, 0(R1) | 1 | 2 | 3-4 | 5 |
| L.D F4, 0(R2) | 2 | 3 | 4-5 | 6 |
| MUL.D F6,F0,F4 | 3 | 4-6 | 7 |  |
| ADD.D F8,F0,F2 | 6 | 7 |  |  |
| S.D F6, 0(R3) |  |  |  |  |
| S.D F8, 0(R4) |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Function units | Function Unit Status | | | | | | | | | |
| Busy | Op | Fi | Fj | Fk | Qj | Qk | Rj | Rk | A |
| Integer1 | No |  |  |  |  |  |  |  |  |  |
| Integer2 | No |  |  |  |  |  |  |  |  |  |
| FPMult1 | Yes | MUL.D | F6 | F0 | F4 |  |  | Yes | Yes |  |
| FPMult2 | No |  |  |  |  |  |  |  |  |  |
| FPDivider | No |  |  |  |  |  |  |  |  |  |
| FPAdder | Yes | ADD.D | F8 | F0 | F2 |  |  | Yes | Yes |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Register Status | | | | | | | | |
| F0 | F2 | F4 | F6 | F8 | F10 | F12 | **……** | F30 |
| FU | MEM[R1+0] |  | MEM[R2+0] | FPMult1 | FPAdder |  |  |  |  |

2. （50 points）For the following instruction sequence:

L.D F6, 34(R2)

L.D F2, 45(R3)

MUL.D F0, F2, F4

SUB.D F8, F2, F6

DIV.D F10, F0, F6

ADD.D F6, F8, F2

1. Fill in the tables that Tomasulo algorithm used when the first instruction completes and finishes writing result. Assume the memory access unit needs two clock cycles to do execution: one for address calculation and one for memory access.

|  |  |  |  |
| --- | --- | --- | --- |
| Instruction status | | | |
|  | ISSUE | EXECUTE | WRITE RESULT |
| 1 | Finish | Finish | Finish |
| 2 | Finish | Finish |  |
| 3 | Finish |  |  |
| 4 | Finish |  |  |
| 5 |  |  |  |
| 6 |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Reservation stations | | | | | | | |
| NAME | BUSY | Op | Vj | Vk | Qj | Qk | A |
| Load1 | No |  |  |  |  |  |  |
| Load2 | Yes | L.D |  |  |  |  |  |
| Add1 | Yes | SUB.D |  |  | Load2 |  |  |
| Add2 | No |  |  |  |  |  |  |
| Add3 | No |  |  |  |  |  |  |
| Mult1 | Yes | MUL.D | M(36+R2) |  |  | Load2 |  |
| Mult2 | No |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Register status | | | | | | | | | |
| Field | F0 | F2 | F4 | F6 | F8 | F10 | F12 | … | F30 |
| Qi | Mult1 | Load2 |  | M(36+R2) | Add1 |  |  |  |  |

1. Assume the following latencies: load is 1 clock cycle; add is 2 clock cycles; multiply is 10 clock cycles; divide is 40 clock cycles. Fill in the tables when the instruction MUL.D is about to write result ( write result in next cycle).

|  |  |  |  |
| --- | --- | --- | --- |
| Instruction status | | | |
|  | ISSUE  Clock | EXECUTE finish  Clock | WRITE RESULT  Clock |
| 1 | Finish | Finish | Finish |
| 2 | Finish | Finish | Finish |
| 3 | Finish | Finish |  |
| 4 | Finish | Finish | Finish |
| 5 | Finish |  |  |
| 6 | Finish | Finish | Finish |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Reservation stations | | | | | | | |
| NAME | BUSY | Op | Vj | Vk | Qj | Qk | A |
| Load1 | No |  |  |  |  |  |  |
| Load2 | No |  |  |  |  |  |  |
| Add1 | No |  |  |  |  |  |  |
| Add2 | No |  |  |  |  |  |  |
| Add3 | No |  |  |  |  |  |  |
| Mult1 | Yes | MUL.D | M(45+R3) | Value(F4) |  |  |  |
| Mult2 | No | DIV.D |  | M(36+R2) | Mult1 |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Register status | | | | | | | | | |
| Field | F0 | F2 | F4 | F6 | F8 | F10 | F12 | … | F30 |
| Qi | Mult1 | M(45+R3) |  | Value | M(45+R3)-M(36+R2) | Mult2 |  |  |  |