

单次循环需要 13 个 Cycles

| 3)               | 1  | 2  | 3  | 4   | 5  | 6   | 7   | 8   | 9  | 10    | 11  | 12 | 13 |
|------------------|----|----|----|-----|----|-----|-----|-----|----|-------|-----|----|----|
| lw a4, 0(a3)     | IF | ID | EX | MEM | WB | WB  |     |     |    |       |     |    |    |
| addw a1, a4, a1  |    | IF | ID | S   | EX | MEM | WB  | WB  |    |       |     |    |    |
| addiw a2, a2, -1 |    |    | IF | S   | ID | EX  | MEM | WB  | WB |       |     |    |    |
| addiw a3, a3, 4  |    |    |    | IF  | ID | EX  | MEM | WB  | WB |       |     |    |    |
| bneq a2, Loop    |    |    |    |     | IF | ID  | EX  | MEM | WB | (a3)P | AIF |    |    |

单次循环需要 12 个 Cycles

| 4.18                 | 开始执行的周期数 | 执行完成的周期数      |
|----------------------|----------|---------------|
| 9. (1) fld f2, 0(a0) | 1        | 4 (a0)0.5 64  |
| fdiv f8, f0, f2      | 5        | 15 (a0)0.5 64 |
| fmul f2, f6, f2      | 16       | 20            |
| fld f4, 0(a1)        | 17       | 20 (a0)0.5 64 |
| fadd f4, f0, f4      | 21       | 23 (a0)0.5 64 |
| fadd f10, f8, f2     | 22       | 24            |
| fsd f10, 0(a0)       | 25       | 26            |
| fsd f4, 0(a1)        | 26       | 27            |
| addi a0, a0, 8       | 27       | 27            |
| addi a1, a1, 8       | 28       | 28            |
| sub x20, x4, a0      | 29       | 29            |
| sub x20, x4, a0      | 29       | 29            |
| bnez x20, Loop       | 30       | 31            |

∴ 一次迭代需要 31 个周期 (不含取指和译码)

只有 RAW 才停顿？

Ex Finish EX

|     |                |    |    |                |
|-----|----------------|----|----|----------------|
| (2) | fld f2,0(a0)   | 1  | 4  |                |
|     | fdiv f8,f0,f2  | 5  | 15 |                |
|     | fmul f2,f6,f2  | 5  | 9  | (第二条指令完成后才 WB) |
|     | fld f4,0(a1)   | 6  | 9  |                |
|     | fadd f4,f0,f4  | 10 | 12 |                |
|     | fadd f10,f8,f2 | 10 | 12 |                |
|     | fsd f10,0(a0)  | 13 | 14 |                |
|     | fsd f4,0(a1)   | 13 | 14 |                |
|     | addi a0,a0,8   | 14 | 14 |                |
|     | addi a1,a1,8   | 14 | 14 |                |
|     | sub x20,x4,x0  | 15 | 15 |                |
|     | bnez x20, Loop | 16 | 17 |                |

..一次迭代需要 17 个周期执行完成

|     |                |    |    |  |
|-----|----------------|----|----|--|
| (3) | fld f2,0(a0)   | 1  | 4  |  |
|     | fld f4,0(a1)   | 1  | 4  |  |
|     | fdiv f8,f0,f2  | 5  | 15 |  |
|     | fmul f2,f6,f2  | 5  | 9  |  |
|     | fadd f4,f0,f4  | 6  | 8  |  |
|     | fadd f10,f8,f2 | 6  | 8  |  |
|     | fsd f4,0(a1)   | 9  | 10 |  |
|     | fsd f10,0(a0)  | 9  | 10 |  |
|     | addi a0,a0,8   | 10 | 10 |  |
|     | addi a1,a1,8   | 10 | 10 |  |
|     | sub x20,x4,x0  | 11 | 11 |  |
|     | bnez x20, Loop | 12 | 13 |  |

..一次迭代需 13 个周期

10. Loop: fld T9, 0(a0)

fmul T10, T0, T2

fdiv T11, T9, T10

fld T12, 0(a1)

fadd T13, T9, T12

fsub T14, T11, T13

fsd T14, 0(a1)

转换表 f0 f1 f2 f3 f4 f5 f6 f7 f8

T0 T1 T10 T3 T12 T5 T13 T7 T14

11. "显式重命名": 以 BOOM 处理器为例, map-table 记录逻辑寄存器与物理寄存器的映射关系; free-list 记录物理记存器的空闲状态, busy-table 记录寄存器是否可读。

当一条指令发起重命名请求时: "由 map-table 获取源操作数; 由 free-list 分配目的寄存器; 由 busy-table 判断源操作数是否可读, 如果可读则发射指令

(2) 隐式重命名: ROB 保存正在执行、尚未提交的指令的结果; ARF 保存已经提交的指令中即将写入寄存器中的值。