

9. (1) 12条指令, 至少需要 $12 - 1 + 2 = 13$ 个周期数.

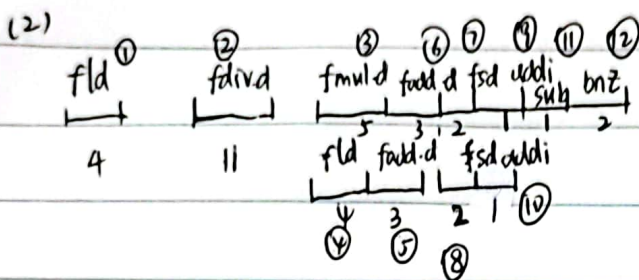
1-2 会有 RAW 冲突, 需加上 $4 - 1 = 3$ 个周期

2-3: WAR 冲突, 加上 $11 - 1 = 10$ 个周期

4-5: WAW, RAW, 加上 $4 - 1 = 3$ 个周期.

6-7: RAW, 加上 $3 - 1 = 2$ 个周期

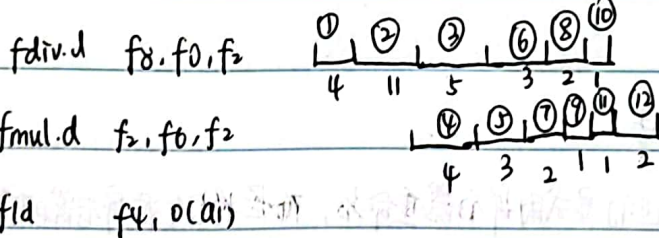
\therefore 共计 $13 + 3 + 10 + 3 + 2 = 31$ 个周期.



\therefore 至少 $4 + 11 + 5 + 3 + 2 + 1 + 1 + 2 = 29$ 个周期

(3) 考虑到 a_1 对 bnz 无影响, 可以在对 a_0 $addi$ 时对 f_4 存诸, 以 B 对 a_1 的 $addi$.

新序列: Loop: fld $f_2, 0(a_0)$



(作为此取数) $fadd.d \ f_{10}, f_8, f_2$

⑥ $fmul.d \ f_4, f_0, f_2$

⑦ $fsd \ f_{10}, 0(a_0)$

⑧ $fsd \ f_4, 0(a_1)$

$addi \ a_0, a_0, 8$

$addi \ a_1, a_1, 8$

$sub \ x_{20}, x_4, a_0$

$bnz \ x > 0, \text{Loop.}$

共计 $4 + 11 + 4 + 3 + 2 + 1 + 1 + 2 = 28$ 个周期, 减少了一个周期

同时未改变逻辑.

10. 重命名的目的为消除 WAW 和 WAR 冲突

指令④和⑤的 f_4 冲突; ④和⑥的 f_4 冲突; ⑥和⑦的 f_8 冲突;

重命名后: `fld T9, 0(a0)`

`fmul.d T10, T11, T10`

`fdiv.d T12, T9, T10`

`fld T20, 0(a1)`

`fadd.d T13, T11, T20`

`fsub.d T30, T12, T13`

`fsd T30, 0(a1)`

11. 显式重命名 优点: 寄存器名称更直观, 可以更精确控制寄存器使用情况

缺点: 需要更多硬件支撑, 寄存器名称数量有限

隐式—— 优点: 节省指令码空间, 节省硬件资源

缺点: 不直观

实现方式: 寄存器分配表; 重命名表

区别: 隐式重命名不需要对程序进行显式的寄存器重命名, 而是将每个源寄存器映射到一个新寄存器