

9: 1) f1d cycle1 发射 C5完成, C8回;

fdiv.d C7 发射 C8完成 C17回 (依赖)

fmul.d C8 发射 C13完成 C18回

f1d C9 发射 C13完成 C18回

fadd.d 依赖 f1d, C15 发射 C18完成 C19回

fadd.d 依赖 f1d, C20 发射 C23完成 C24回

fsub.d 依赖 f1d, C25 发射 C27完成

fsub.d 依赖 f1d, C26 发射 C28完成

addi 指令的无依赖 C27, C28 发射, C29, C30 回

sub 依赖 a0, C30 发射, C32 回

bnez 依赖 x20, C33 发射, C35 完成

→ 共 35 个周期

2) f1d 同 1) [pipe 1]

fdiv.d C7 进入 pipe 2, C6 开始执行, C16 完成, C17 回

fmul.d C8 进入 pipe 1, C7 完成, C18 回

f1d C9 进入 pipe 2, C6 完成, C17 回

fadd.d 依赖 f1d, C20 进入 pipe 1, C7 执行, C24 完成

fadd.d 依赖 f1d, C21 进入 pipe 2, C8 执行, C25 完成

fsub.d 依赖 f1d, C26 进入 pipe 1, C9 执行, C31 完成

fsub.d 依赖 f1d, C27 进入 pipe 2, C10 执行, C32 完成

addi 无依赖, C29 进入, C30 执行, C31 完成

sub 依赖 a0, C30 执行, C31 完成 C32 回

bnez C33 执行, C35 完成

→ 共 34 个周期

3) 主序为 f2 → f8 → f10 → a0 → x20 → bnez 线

让 fsub 指令与 addi 执行同时进入, 减少 WAR 影响

让 sub 指令先于 addi a1 进入, 缩短周期

新顺序为

f1d	f2, 0(a0)
fdiv.d	f8, f0, f2
fmul.d	f2, f0, f2
fadd.d	f2, f0, f2
addi	a1, a1, 8
fadd.d	f10, f8, f2
fsub.d	f10, 0(a0)
addi	a0, a0, 8
fsub.d	f2, f0, f2
sub	x20, x1, a0
bnez	x20, loop

共约 3 个周期

共 23 个

10. f1d f4, 0(a0) f1d T9, 0(a0)

fmul.d f2, f0, f2 fmul.d T10, T0, T2

fdiv.d f8, f4, f2 fdiv.d T11, T9, T10

f1d f4, 0(a1) f1d T12, 0(a1)

fadd.d f6, f0, f4 fadd.d T13, T0, T12

fsub.d f8, f8, f6 fsub.d T14, T11, T13

fsub.d f8, 0(a1) fsub.d T14, T11, T13

addi a1, a1, 8

bnez x20, loop

(若 f0 ~ f8 对应 T0 ~ T8)

11. 显式重命名使用 map table 记录一个对应关系, 用多个物理上实际存在的寄存器, 将数据都记在物理寄存器中。

隐式重命名建立映射表, 记录大量数据, 无需额外数目的实际寄存器, 相比之下它的寄存器需求少, 但对数据读取要求很高, 因此功耗很大