

9. 将指令序列从上到下依次记为 1~12 号

(1) 指令	开始周期~结束周期	(2) 指令	开始周期~结束周期
① fld	1-4	① fld	1-4
② fdiv.d	5-15	② fdiv.d	5-15
③ fmul.d	6-10	③ fmul.d	5-9
④ fld	7-10	④ fld	6-9
⑤ fadd.d	11-13	⑤ fadd.d	10-12
⑥ fadd.d	16-18	⑥ fadd.d	16-18
⑦ fsd	19-20	⑦ fsd	19-20
⑧ fsd	20-21	⑧ fsd	19-20
⑨ addi	22-22	⑨ addi	20-20
⑩ addi	22-22	⑩ addi	20-20
⑪ sub	23-23	⑪ sub	21-21
⑫ bnz	24-25	⑫ bnz	22-23

单发射考虑分支和数据冲突

双发射考虑真数据冲突

一次迭代需 25 周期

一次迭代需 23 周期

(3) 调整为:	① fld f2, 0(a0)	1-4
	② fdivd f8, f0, f2	5-15
	③ fmul.d f2, f6, f2	5-9
	④ fld f4, 0(a1)	6-9
	⑤ fadd.d f4, f0, f4	10-12
	⑥ fadd.d fsd f4, 0(a1)	13-14
	⑦ fadd.d f10, f8, f2	16-18
	⑧ fsd f10, 0(a0)	19-20
	⑨ addi a0, a0, 8	19-19
	⑩ addi a1, a1, 8	20-20
	⑪ sub x20, x4, a0	20-20
	⑫ bnz x20, Loop	21-22

一次迭代需 22 周期



10. 重命名: Loop: fld T9, 0(a0)

fmul.d T10, f0, f2

fdiv.d T11, T9, T10

fld T12, 0(a1)

fadd.d T13, f0, T12

fsub.d T14, T11, T13

fsd T15, 0(a1)

11. 显式重命名: 物理上的寄存器堆中具有的真实寄存器数目比ISA定义的寄存器数目更多。对每条需写回的指令重新分配一个目的寄存器。其中包括已经提交和尚未提交处于“推测”状态的寄存器。

隐式重命名: 物理寄存器数和ISA定义的寄存器数相同, ISA寄存器只保存已经提交的指令的值而不包括处于推测状态的值(这部分由ROB保存)。

显式重命名在逻辑实现上会比较简单, 设计复杂度不高, 可获得性能也较好。缺点是需较多的物理寄存器。

而隐式重命名需要的物理寄存器数少, 实现上虽较复杂, 但更优。

其缺点在于每个操作数在其生命周期需保存在ROB-SARF两位置。

读取数据的复杂度较高, 功耗更高。

显式实现: 映射表记录逻辑寄存器与物理寄存器间的映射关系。用空闲列表记录物理寄存器的空闲状态, 用忙碌表记录寄存器是否可读。当一条指令发起重命名请求, 通过索引映射表获取其源操作数逻辑寄存器对应的物理寄存器, 由空闲列表分配一个空闲的物理寄存器作为目的寄存器, 最后以忙碌表判断其是否可读, 是则发生

隐式实现: 用ROB保存正在执行尚未提交的指令效果, 用ARF保存已提交指令中即将写入寄存器中的值。建立映射表来记录操作数在ROB中的位置, 指令被写入ROB即完成重命名。

