

1)	if	id	EXMEMW
fld f2, 0(a0)	1	2	3456
fdiv.d f8, f0, f2	2	7	8-18
fmul.d f2, f6, f2	7	8	9, 10, 11, 12, 13
fld f4, 0(a1)	8	9	10, 11, 12, 13
fadd.d f4, f0, f4	9	14	15 16 17
fadd.d f10, f8, f2	14	19	20, 21, 22
f.s.d f10, 0(a0)	19	23	24 25 25
f.s.d f4, 0(a1)	23	24	25 26
addi a0, a0, 8	24	25	26
addi a1, a1, 8	25	26	26 27
sub x20, x4, a0	26	27	28
bnz x20, Loop	27	29	30, 31

31
共 31 条

2) fld f2, 0(a0)	1	2	3456
fdiv.d f8, f0, f2	1	7	8-18
fmul.d f2, f6, f2	7	8	9, 10, 11, 12, 13
fld f4, 0(a1)	7	8	9, 10, 11, 12
fadd.d f4, f0, f4	8	13	14, 15, 16
fadd.d f10, f8, f2	8	19	20, 21, 22
f.s.d f10, 0(a0)	19	23	24 25
f.s.d f4, 0(a1)	19	20	21 22
addi a0, a0, 8	23	24	25
addi addi a1, a1, 8	23	24	25
sub x20, x4, a0	24	26	27
bnz x20, Loop	24	28	29, 30

共 30 条



3.	if	id	emw
fld, f2, 0(a0)	1	2	3 4 5 6
fdivd f8, f0, f2	1	7	8 - 18
fmul.d f2, f6, f2	7	8	9 - 13
fld f4, 0(a1)	7	8	9 - 12
fadd.d f6, f0, f4	8	13	14 15 16
fadd.d f10, f8, f2	8	19	20 21 22
addi a0, a0, 8	19	20	21
addi a1, a1, 8	19	20	21
sub x20, x4, a0	20	21	23
fsw f4, 0(a1) 0(a1)	20	21	22 23
fsw f10, 0(a0)	22	23	24 25
bnz x20, Loop	22	24	25 26 共26条

11.

显式重命名每个操作符都有一个明确的名称和标识符, 可以通过修改这些标识符和寄存器流的顺序来重命名。该方案确保物理寄存器数目比 ISA 定义的更多, 优点: 简单, 易于调试和优化, 缺点: 会导致重复计算和额外的数据传输, 降低了系统的效率和性能。

隐式: 物理实现的寄存器数量与 ISA 规定保持一致, 但其中仅存放已最终写回的指令结果。该寄存器堆一般称为 ARF, 优点: 减少重复计算, 提高效率, 缺点: 实现相对复杂, 难以测试和优化。

实现方式: 显式 ~: 引入两种硬件: 1. 空闲列表, 指示当前物理寄存器堆中有哪些是可用的; 2. 重命名列表, 用于维护物理寄存器和 ISA 寄存器之间的关系。

隐式: 引入重排序缓冲区, 且为了保持正确的数据依赖关系, 整个结构需要大量的额外的表项来记录寄存器的最新值是否已写回 ARF 中还是暂存在重排序缓冲区中。



0

	IF	ID	EX	WE
op: fld f4, 0(a0)	1	2	3	4
fmul.d f2, f0, f2	2	3	4	5
fdiv.d f8, f4, f2	3	5	6	7
fld f4, 0(a1)	4	5	6	7
fadd.d f6, f0, f4	5	7	8	9
fsub.d f8, f8, f6	6	10	11	12
fsl f8, 0(a1)	7	13	14	15

f0	f1	f2	f3	f4	f5	f6	f7	f8
T0	T1	T2	T3	T4	T5	T6	T7	T8
		T10		T9				T11
				T102		T13		T14
								T15

⇒ fld T0, 0(a0)

fmul T10, T0, T2

fdiv.d T11, T4, T10

fld T12, 0(a1)

fadd.d T13, T0, T12

fsub.d T14, T11, T13

fsl T15, 0(a1)

