

3-9. 1)

数据冲突

Loop:	fld	f2,0(a0)	4	← 每条指令所需 的周期数
	fdiv.d	f8,f0,f2	11	
	fmul.d	f2,f6,f2	5	
	fld	f4,0(a1)	4	
	fadd.d	f4,f0,f4	3	
	fadd.d	f10,f8,f2	3	
	fsd	f10,0(a0)	2	
	fsd	f4,0(a1)	2	
	addi	a0,a0,8	1	
	addi	a1,a1,8	1	
	sub	x20,x4,a0	1	
	bnz	x20,Loop	2	

指令类型	总周期数
内存加载	4
内存存储	2
整型运算	1
分支	2
浮点加法	3
浮点乘法	5
浮点除法	11

下表表示各个周期运算单元的情况：

(蓝色圈表示指令的发射)

周期	1	...	4	5	...	15	16	17	...	20	21	22	23	24	25	26	27	28	29	30	31	
LD1	4	...	1				4	...	1													
FDIV				11	...	1																
FMUL						5	4	...	1													
FADD1										3	2	1										
FADD2										3	2	1										
SW1													2	1								
SW2													2	1								
INT													1	1	1							
B																	2	1				

故一共需要31个周期。

2) 不需要考虑WAR冲突，且一次发射两条指令：因此需要23个周期。

周期	1	...	4	5	6	...	9	10	11	12	...	15	16	17	18	19	20	21	22	23	
LD1	4	...	1	4	...	1															
FDIV			11	10	...	7	6	5	4	...	1										
FMUL			5	4	...	1															
FADD						3	2	1			3	2	1								
SW1													2	1							
SW2													2	1							
INT1													1	1							
INT2													1								
B																2	1				

### 3) 调整指令序列如下：

周期	1	2	3	4	5	6	7	8	9	10	...	15	16	17	18	19	20	21	22	
LD1	4	3	2	1																
LD2	4	3	2	1																
FDIV			11	10	9	8	7	6	...	1										
FMUL			5	4	3	2	1													
FADD1			3	2	1						3	2	1							
SP						2	1					3	1							
INT						1						0	1							
B													2	1						

Loop:	调整顺序后：		
	fld	f2,0(a0)	4
fdiV.d	f8,f0,f2	11	fld f2,0(a0)
fmul.d	f2,f6,f2	5	fld f4,0(a1)
fld	f4,0(a1)	4	fdiV.d f8,f0,f2
fadd.d	f4,f0,f4	3	fmul.d f2,f6,f2
fadd.d	f10,f8,f2	3	fadd.d f4,f0,f4
fsd	f10,0(a0)	2	fsd f4,0(a1)
fsd	f4,0(a1)	2	addi a1,a1,8
addi	a0,a0,8	1	fadd.d f10,f8,f2
addi	a1,a1,8	1	fsd f10,0(a0)
sub	x20,x4,a0	1	addi a0,a0,8
bnz	x20,Loop	2	sub x20,x4,a0
			bnz x20, Loop.

调整后一次迭代需要22个周期。

### 3-10. 重命名后的指令序列：

Loop: fld	T9, 0(a0)
fmul.d	T11, T10, T11
fdiV.d	T12, T9, T11
fld	T13, 0(a1)
fadd.d	T14, T10, T13
fsub.d	T15, T12, T14
fsd	T15, 0(a1)

Loop: fld	f4,0(a0)
fmul.d	f2,f0,f2
fdiV.d	f8,f4,f2
fld	f4,0(a1)
fadd.d	f6,f0,f4
fsub.d	f8,f8,f6
fsd	f8,0(a1)

**区别1：**显示重命名确保物理寄存器位具有的真实寄存器数要比ISA定义的寄存器数要更多。而对于隐式重命名，物理实现的寄存器数量与ISA规定保持一致，但其中仅有存放已最终写回的结果，处于推测状态的指令值则由其他硬件保存。

**优缺点：**对于显示重命名，不需要在重排序缓冲区中创建大量的存储临时值定位。

但是需要实现比ISA规定更多的寄存器数量；而隐式重命名不需要实现更多的物理寄存器，但是需要额外的表项来记录寄存器的值是写回APF还是暂存在重排序缓冲区。

**实现方式：**显示重命名引入两种硬件：空闲列表(FL)以维护物理寄存器的空闲状态信息，以及重命名列表以维护物理寄存器与ISA寄存器之间的映射关系。

对于隐式重命名，则利用重排序缓冲区来保存处于推测状态的指令值，并使用额外的表项记录寄存器最新值是已写回寄存器还是暂存在重排序缓冲区。