

9.(1) 周期数、流水线状态

1	fld f <sub>2</sub> , 0(a <sub>0</sub> )	
2	S (RAW)	
3	S	
4	S	
5	fdiv.d f <sub>8</sub> , f <sub>0</sub> , f <sub>2</sub>	
6	fmul.d f <sub>2</sub> , f <sub>6</sub> , f <sub>2</sub>	
7	fld f <sub>4</sub> , 0(a <sub>1</sub> )	
8	S (RAW) f <sub>4</sub> 写完 才能被读取	
9	S	
10	S	
11	fadd.d f <sub>4</sub> , f <sub>0</sub> , f <sub>4</sub>	
12	S (RAW) f <sub>8</sub> 写完 才能被读取	
13	S	
14	S	
15	S	
16	fadd.d f <sub>10</sub> , f <sub>8</sub> , f <sub>2</sub>	
17	S (RAW) f <sub>10</sub> 写完后 才能被读取	
18	S	
19	fsd f <sub>10</sub> , 0(a <sub>0</sub> )	
20	fsd f <sub>4</sub> , 0(a <sub>1</sub> )	
21	addi a <sub>0</sub> , a <sub>0</sub> , 8	
22	addi a <sub>1</sub> , a <sub>1</sub> , 8	
23	sub X <sub>20</sub> , X <sub>4</sub> , a <sub>0</sub>	
24	bnez X <sub>20</sub> , Loop	!! 总共需要 25 个周期
25	S (分支需 2 个周期)	
26		

(2) 地址	流水线1 状态	流水线2 状态
1	fld f2,0(a0)	空闲
2	s (RAW)	空闲
3	s	空闲
4	s	空闲
5	fdiv.d f8, f0, f2	fmul.d f2, f6, f2
6	fld f4, 0(a1)	s (RAW) f4 需等待 fld f4, 0(a1)
7	s	s
8	s	s
9	s	s
10	fadd.d f4, f0, f4	s fadd.d f8 等待 fdiv.d
11	s	s
12	s	s
13	s	s
14	s	s
15	s	s
16	fadd.d f10, f8, f2	s fsd 等待 fadd f10
17	s (RAW)	s
18	s	s
19	-fsd f10, 0(a0)	fsd f4, 0(a1)
20	addi a0, a0, 8	addi a1, a1, 8
21	s ab x20, x4, a0	bnz x20, Loop
22		s 分支需 2 个周期

∴ 共需要 22 个周期

(3)

调整后指令序列：

fld f2, o(a0)  
 fld f4, o(a1)  
 fdiv.d f8, f0, f2.  
 fadd.d f4, f0, f4  
 fmul.d f2, f6, f2.  
 fsd f4, o(a1)  
 addi a1, a1, 8  
 fadd.d f10, f8, f2.  
 fsd f10, o(a0)  
 addi a0, a0, 8  
 sub x20, x4, a0  
 bnez x20, loop

周期数

	流水线 1	流水线 2
1	fld f2, o(a0)	fld f4, o(a1)
2	S	S
3	S	S
4	S	S
5	fdiv.d f8, f0, f2	fadd.d f4, f0, f4
6	fmul.d f2, f6, f2	S
7	S	S
8	fsd f4, o(a1)	addi a1, a1, 8
9	S	S
10	S	S
11	S	S
12	S	S
13	S	S
14	S	S
15	S	S
16	fadd.d f10, f8, f2	S
17	S	S
18	fsd f10, o(a0)	addi a0, a0, 8
19	S	S
20	sub x20, x4, a0	bnez x20, loop
21	S	S

共需要 21 个周期，减少了 1 个周期。

10. Loop : fild T9, o(a0)  
 fmul.d T10, f0, f2  
 fdiv.d T11, T9, T10  
 fild T12, o(a1)  
 fadd.d T13, f0, T12  
 fsub.d T14, T11, T13  
 fsd T14, o(a1).

f0	f1	f2	f3	f4	f5	f6	f7	f8
T10				T9		T13		T11
					T12			T14

### 11. 显示寄存器重命名:

其基本结构是：map-table 记录逻辑寄存器与物理寄存器之间的映射关系；free-list 记录物理寄存器的空闲状态；busy-table 记录寄存器是否可读。显示重命名方案中 ROB 不记录指令的结果，即将提交的数据和处于推测状态的数据都保存到物理寄存器，因此物理寄存器数目要高于逻辑寄存器数目。当一条指令发起重命名请求时，通过 map-table 获取其源操作数逻辑寄存器对应的物理寄存器，由 free-list 分配一个空闲的物理寄存器作为指令的目的寄存器，最后通过 busy-table 判断指令的原操作数寄存器是否可读，如果可读，指令将被发射。

### 隐式寄存器重命名：

此种方案中，ROB 保存正在执行、尚未提交的指令的结果；ARF 保存已经提交的指令中即将写入寄存器中的值。隐式重命名方案中 ARF 只保存已经提交的指令的值，处于推测状态的指令的值由 ROB 保存，因此需要的物理寄存器数量与逻辑寄存器数量相同，隐式

重命名操作还需要建立一个映射表，记录操作数在 ROB 中的位置。由于流水线中后续指令经提交的指可能有相同的目的寄存器，映射表需要增加一个表项，记录对应寄存器的最新值保存在 ROB 还是 ARF 中，这一设计为实现数据前馈、消除 RAW 冲突创造了条件。隐式重命名操作不需要 free-list 来记录物理寄存器状态，指令被写进 ROB 即完成重命名。相比于显式重命名，隐式重命名需要的物理寄存器数目更少，但每个操作数在其生命周期内需要保存在 ROB 和 ARF 两个位置，读取数据复杂度较高、功耗更高。