

9 (1) 在题设下可以得到下图周期变化

指令序列:	Issue	EX	WR
fld f2,0(a0)	1	$1+4=5$	6
fdiv.d f8,f0,f2	2	$6+1=7$	18
fmul.d f2,f6,f2	7	$7+5=12$	19
fld f4,0(a1)	8	$8+4=12$	20
fadd.d f4,f0,f4	9	$20+3=23$	24
fadd.d f10,f8,f2	21	$21+3=24$	25
fsd f10,0(a0)	22	$25+2=27$	28
fsd f4,0(a1)	26	$26+2=28$	29
addi a0,a0,8	27	$27+1=28$	30
addi a1,a1,8	28	$28+1=29$	31
sub x20,x4,a0	29	$30+1=31$	32
bnz x20,Loop	31	$32+2=34$	35

∴ 一次迭代总共需要 35 个周期

(2) 改为双发射得到下图，一次迭代需 35 个周期

指令序列:	Issue	EX	WR
fld ① f2,0(a0) 4	1	$1+4=5$	6
fdiv.d ② f8,f0,f2 11	1	$6+1=7$	18
fmul.d ③ f2,f6,f2 5	7	$7+5=12$	19
fld ① f4,0(a1) 4	7	$7+4=11$	20
fadd.d ② f4,f0,f4 3	8	$20+3=23$	24
fadd.d ④ f10,f8,f2 3	8	$19+3=22$	25
fsd ⑤ f10,0(a0) 2	21	$25+2=27$	28
fsd ③ f4,0(a1) 2	21	$24+2=26$	29
addi ⑥ a0,a0,8 1	26	$26+1=27$	30
addi ④ a1,a1,8 1	26	$26+1=27$	31
sub ⑦ x20,x4,a0 1	27	$30+1=31$	32
bnz ⑧ x20,Loop 2	27	$32+2=34$	35

(3) 按前后相关性将指令进行排序可以得到  
2条互不相关的指令串(蓝色与红色), 指令后面  
是从发射到执行需要的周期数。

同时也能将12条指令分为3大部分  
其中第3部分无法再进行改善

指令序列:		Issue	EX	WR
① {	fld ① f2,0(a0) 4	1	$1+4=5$	6
	fdiv.d ② f8,f0,f2 11	1	$6+11=17$	18
	fmul.d ③ f2,f6,f2 5	7	$7+5=12$	19
	fld ① f4,0(a1) 4	7	$7+4=11$	20
	fadd.d ② f4,f0,f4 3	8	$20+3=23$	24
	fadd.d ④ f10,f8,f2 3	8	$19+3=22$	25
② {	fsd ⑤ f10,0(a0) 2	21	$25+2=27$	28
	fsd ③ f4,0(a1) 2	21	$24+2=26$	29
③ {	addi ① a0,a0,8 1	26	$26+1=27$	30
	addi ④ a1,a1,8 1	26	$26+1=27$	31
	sub ⑦ x20,x4,a0 1	27	$30+1=31$	32
	bnz ⑧ x20,Loop 2	27	$32+2=34$	35

针对第1部分可以将ld指令放在一起  
同时将EX周期数较大的div mul指令放在  
一起并在经过比较后把mul指令提前

而第2部分可以把两条指令顺序调换  
以符合①中先计算出f4后计算出f10的顺序

调换后的周期变化如下：

指令序列:	Issue	EX	WR
fld f2,0(a0)	1	$1+4=5$	6
<del>fld f2,0(a0)</del>	1	$1+4=5$	7
<del>fmul.d f2,f6,f2</del>	2	$6+4=10$	11
<del>fdiv.d f8,f0,f2</del>	2	$6+11=17$	18
fadd.d f4,f0,f4	7	$7+3=10$	19
fadd.d f10,f8,f2	7	$18+3=21$	22
fsd <del>f4,0(a1)</del>	19	$19+2=21$	23
fsd <del>f10,0(a0)</del>	19	$22+2=24$	25
addi a0,a0,8	23	$23+1=24$	26
addi a1,a1,8	23	$23+1=24$	27
sub x20,x4,a0	24	$26+1=27$	28
bnz x20,Loop	24	$28+2=30$	31

共 31 个周期

10 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)

WAR  
 WAW

指令序列: Loop: fld T9,0(a0)

fmul.d T10,T11,T10

fdiv.d T12,T13,T10

fld T14,0(a1)

fadd.d T15,T11,T14

fsub.b T16,T12,T15

fsd T16,0(a1)

11 显式重命名安排比指令集要求的寄存器 (ISA 寄存器) 数量更多的物理寄存器。在重命名的同时每个指令的寄存器指示符指向所使用的物理寄存器。隐式重命名物理寄存器数量和 ISA 寄存器数量相同, ISA 寄存器只保存已提交指令的值而不包括处于“推测”状态的值。

相比于显式, 隐式需要物理寄存器更少, 但每个操作数在其生命周期中需保存在 ROB、ARF 两位置。读取数据的复杂度较高, 功耗较高。

对 BOM 处理器: 显式:

- 索引 nap-table 获取操作数逻辑寄存器
- free-list 分配一个空闲的物理寄存器作为目的寄存器
- busy-table 判断将操作数寄存器是否可读
- 若可读则发射

隐式: { ROB 保存正执行尚未提交的结果  
ARF 保存已提交、即将写入的值  
建立一张链表记录操作数在 ROB 中位置。  
↓  
增加表项记录对应寄存器最新值保存在 ROB 还是 ARF 中