

9/

考虑一个顺序流水线，忽略前端的取指和译码，处理器从发射到执行完成不同指令所需要的总周期数如下表所示。

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

考虑如下的指令序列：

```

Loop:    fld      f2,0(a0)
          fdiv.d   f8,f0,f2
          fmul.d   f2,f6,f2
          fld      f4,0(a1)
          fadd.d   f4,f0,f4
          fadd.d   f10,f8,f2
          fsd     f10,0(a0)
          fsd     f4,0(a1)
          addi    a0,a0,8
          addi    a1,a1,8
          sub    x20,x4,a0
          bnz    x20,Loop

```

- 1) 假设一条单发射顺序流水线，在没有数据冲突或分支指令时，每个周期均会新发射一条指令（假设运算单元是充足的）。检测到数据冲突或分支指令时则会暂停发射，直到冲突指令执行完毕才会发射新的指令。则上述代码段的一次迭代需要多少个周期执行完成？

发射到执行完成所处的周期

Loop:	fld	f2,0(a0)	1 → 4
	fdiv.d	f8,f0,f2	5 → 15
	fmul.d	f2,f6,f2	6 → 10
	fld	f4,0(a1)	7 → 10
	fadd.d	f4,f0,f4	11 → 13
	fadd.d	f10,f8,f2	16 → 18
	fsd	f10,0(a0)	19 → 20
	fsd	f4,0(a1)	20 → 21
	addi	a0,a0,8	21 → 21
	addi	a1,a1,8	22 → 22
	sub	x20,x4,a0	23 → 23
	bnz	x20,Loop	24 → 28

共25个周期

- 2) 假设一条双发射顺序流水线，取指和译码的带宽足够、运算单元充足，且数据在两条流水线之间的传递是无延迟的。因此只有真数据冲突才会导致流水线停顿。则上述代码段的一次迭代需要多少个周期执行完成？

- 3) 调整指令的排列顺序，使得其在上述双发射流水线中完成一次迭代需要的周期数量减少。给出调整后的指令序列及一次迭代所需要的周期数。

(2)

Loop:		发射到执行所处周期数
	fld f2,0(a0)	1→4
	fdiv.d f8,f0,f2	5→15
	fmul.d f2,f6,f2	5→9
	fld f4,0(a1)	6→9
	fadd.d f4,f0,f4	10→12
	fadd.d f10,f8,f2	16→18
	fsd f10,0(a0)	14→20
	fsd f4,0(a1)	14→20
	addi a0,a0,8	10→20
	addi a1,a1,8	10→20
	sub x20,x4,a0	21→21
	bnz x20,Loop	22→23

共23个周期

(3) 调整后的指令序列：发射到执行所处周期数

fld f2,0(a0)	1→4
fld f4,0(a1)	1→4
fadd.d f4,f0,f4	5→8
fdiv.d f8,f0,f2	5→15
fmul.d f2,f6,f2	6→10
fsd f4,0(a1)	9→11
addi a1,a1,8	9→9
fadd.d f10,f8,f2	16→18
fsd f10,0(a0)	14→20
addi a0,a0,8	14→14
sub x20,x4,a0	20→20
bnz x20,Loop	21→22

共22个周期

10. 考虑如下的代码片段：

```
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)
```

现将其进行简单的寄存器重命名，假定有 T0~T63 的临时寄存器池，且 T9 开始的寄存器可用于重命名。写出重命名后的指令序列。

```
fld      T9,0(a0)
fmul.d  T10,f0,f2
fdiv.d  T11,T9,f2
fld      T12,0(a1)
fadd.d  T13,f0,T12
fsub.d  T14,T11,T13
fsd      T14,0(a1)
```

11

查阅资料，简述显式重命名和隐式重命名的区别、优缺点以及可能的实现方式

解区别：

显式重命名：物理寄存器堆具有的真实寄存器数比ISA定义的寄存器数目更多。

隐式重命名：该方案中物理实现的寄存器数量与ISA规定保持一致，但其中仅存放已经最终写回的指令结果。处于推测状态的指令值由一些其他结构保存。

优缺点：显式重命名读取数据速度相对较低。

隐式重命名需要更多的物理寄存器，但每个操作数在其生命周期中需要保存在ROB和RF两个位置，读取数据的速度较高，功耗较高。

可能实现的方式：

显式重命名方案一般需要引入两种硬件：①空闲表，用于维护物理寄存器的空闲状态信息，它指示了当前物理寄存器中有那些寄存器是可用的。

②重命名列表，用于维护物理寄存器和ISA寄存器之间的映射关系。

实现方式：当指令译码后，处理器查找FL并选择一个空闲的物理寄存器，将其和该指令要写入的目的ISA寄存器进行绑定并记录在RT中。同时指令的源操作数也需要查找RF以确定是否需要从某个被映射的物理寄存器中取出值。当指令执行阶段结束后，结果被写入对应的物理寄存器。

隐式重命名为了保持正确的数据依赖关系，整个结构需要一个额外的未项来记录寄存器的最新值是已写回ARF中还是暂存在重排序缓冲区中。为此重排序缓冲区一般需要支持前馈，以将处于推测状态的最新值转发给其他指令作为源操作数。