

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
ld	4
sd	2
sub/ add	1
bnz	2
add	3
mul	5
div	11

sub / add

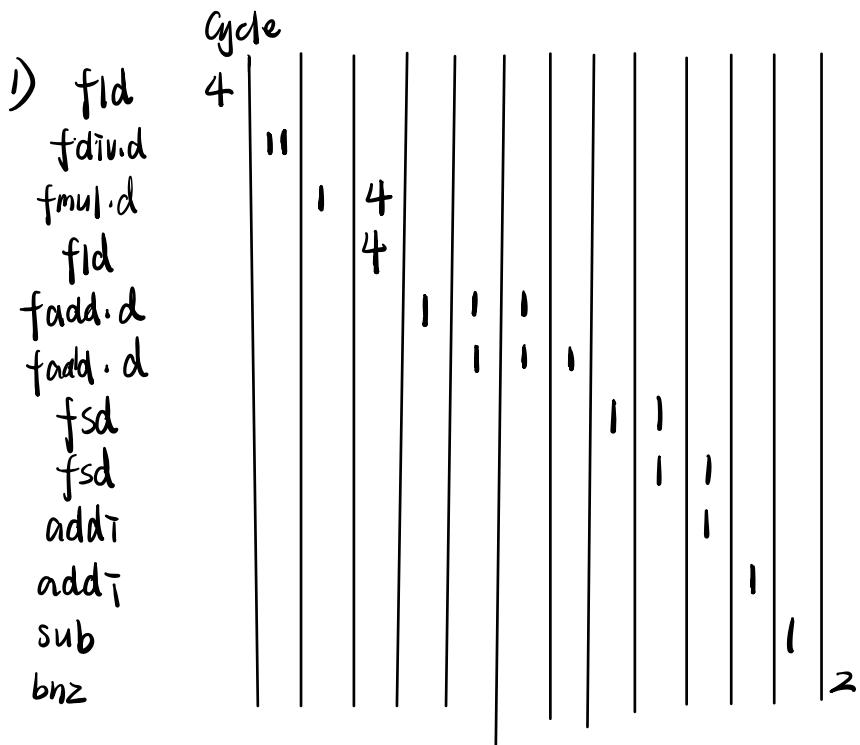
考虑如下的指令序列：

```

Loop:    fld      f2,0(a0)
          fdiv.d   f8,f0,f2 | WAR冲突
          fmul.d   f2,f6,f2
          fld       f4,0(a1)
          fadd.d   f4,f0,f4
          fadd.d   f10,f8,f2
          fsd      f10,0(a0) ← ⋆Attention: 将 a0 寄存器值写入为地址对齐的寄存器中
          fsd      f4,0(a1) ← 值写入
          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？
- 2) 假设一条双发射流水线，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。



则一次迭代需要 3 个周期。

2) 双发射顺序流水线 \Rightarrow 一次发射两条指令
仅 RAW 冲突导致流水线停顿.

fld	4					
fdiv.d		1	4	6		
fmul.d		1	4			
fld			4			
fadd.d				3		
fadd.d				3		
fsd					1	1
fsd					1	1
addi					1	
addi					1	
sub						1
bnz						2

3)	Cycle												
f1d f2, 0(a0)	4												
f1d f4, 0(a1)	4												
fdiv.d f8, f0, f2		3	2	b									
fadd.d f4, f0, f4		3											
fmul.d f2, f6, f2			2										
fsd f4, 0(a1)			2										
addi a1, a1, 8				1									
fadd.d f10, f8, f2					1	2							
fsd f10, 0(a0)							1	1					
addi a0, a0, 8							1						
sub x20, x4, a0								1					
bnz x20, Loop									2				

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。

执行开始前，处理器默认 $f0 \sim f8$ 对应 $T_0 \sim T_8$

Loop:	fld	$T_9, 0(a0)$
	fmul.d	T_{10}, T_0, T_2
	fdiv.d	T_{11}, T_9, T_{10}
	fld	$T_{12}, 0(a1)$
	fadd.d	T_{13}, T_0, T_{12}
	fsub.d	T_{14}, T_{11}, T_{13}
	fsd	$T_{14}, 0(a1)$

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。

① 显式重命名：引入空槽（维护物理寄存器的空闲状态信息）与重命名表（维护物理寄存器与 ISA 寄存器之间映射关系）。在显式重命名情况下，重排序缓冲区本身并不存储指令计算结果，而将需提交及处于推测状态的数据均保存在物理寄存器内，由表为维护映射关系。
优点：无需在重排序缓冲区中创建大量的存储临时值空位，因此在现代高性能处理器设计中被大量使用。

实现方式：提供更多的物理寄存器

② 隐式重命名：物理实现的寄存器数量与 ISA 规定保持一致，但其中只存放已最终写回结果，处于推测状态指令值由一些其他结构保存，例如存放在重排序缓冲区中，该机构负责与缓存系统记录寄存器最新值已写回 APF 还是暂存在重排序缓冲区中，为此重排序缓冲区一般需支持前读。
实现方式：将推测值暂存于重排序缓冲区等其他结构。