Week 9. 9.10.11

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？
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。

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) 4 11 3 2 1 1 1 2

I_1 ××× I_2 ×××××××××× I_6 ×× I_7 × I_9 I_{10} I_{11} I_{12} ×

I_3 ×××× I_4 ××× I_5 ×× I_8 ×

⇒ Total_cycle = 4+11+3+2+1+1+1+2 = 25

(2) 4 11 3 2 1 1 2

I_1 ××× I_2 ×××××××××× I_6 ×× I_7 × I_9 I_{11} I_{12} ×

I_3 ×××× I_4 ××× I_5 ×× 00 I_8 × I_{10}

Total = 4+11+5+3+2+1+1+2 = 29

(3) ⇒ I_1 ××× I_2 ×××××××××× I_6 ×× I_7 × I_9 I_{11} I_{12} ×

⇒ I_4 ××× I_3 ×××× I_5 ×× I_8 × I_{10}

排序：

考虑如下的指令序列：

```
Loop:  1  fld     f2,0(a0)
       2  fdiv.d  f8,f0,f2
       3  fmul.d  f2,f6,f2
       4  fld     f4,0(a1)
       5  fadd.d  f4,f0,f4
       6  fadd.d  f10,f8,f2
       7  fsd     f10,0(a0)
       8  fsd     f4,0(a1)
       9  addi    a0,a0,8
       10 addi    a1,a1,8
       11 sub     x20,x4,a0
       12 bnz     x20,Loop
```

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

Total = 4+11+3+2+1+1+2 = 29

10.

```
fld     T9, 0(a0)
fmul.d  T10, f0, f2
fdiv.d  T11, f8, T10
fld     T12, 0(a1)
fadd.d  T13, f0, T12
fsub.d  T11, T11, T13
fsd     T11, 0(a1)
```

使 T9、T12 将序始 f4 区分开

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。

显式：通过在编译时或程序员手动重命名寄存器来消除数据冲突和 reg 依赖。实现：通过编译器对 code 静态分析和优化，识别出其中的数据依赖并重新分配寄存器来消除依赖。

优点是硬件复杂度低，可并行性高；缺点是对编译器要求较高，寄存器资源使用效率低，而且可能会存在动态路径中依赖无法优化

隐式：在运行时通过处理器硬件来消除冲突依赖：如使用 RRT 和 RAT。

实现方式：用硬件结构在 ID 阶段将逻辑 reg 映射到物理 reg 时根据表中信息动态调度和执行指令以消除冲突依赖。

优点是：编译器要求低，动态处理更彻底，提高可并行性；缺点是：硬件设计更复杂，增加功耗和面积。