9. Loop: fld f_2,0(a_0)
fdiv.d f_8,f_0,f_2
fmul.d f_2,f_6,f_2
fld f_4,0(a_1)
fadd.d f_4,f_0,f_4
fadd.d f_{10},f_8,f_2
fsd f_{10},0(a_0)
fsd f_4,0(a_1)
addi a_0,a_0,8
addi a_1,a_1,8
sub x_{20},x_4,a_0
bnz x_{20},Loop

11)
~~4+1+1+3+2+1+1+1+2=15~~
4+11+3+2+1+1+1+2=25(个)

12)
~~4+1+1+3+2=20~~
4+1+1+3+2+1+2=23(个)

~~13)~~
13) 4+1+1+3+2+1+2=21(个)

新顺序:
fld f_2,0(a_0)
fld f_4,0(a_1)
fdiv.d f_8,f_0,f_2
fmul.d f_2,f_0,f_2
fadd.d f_4,f_0,f_4
~~addi a_0,a_0,8~~
~~addi a_1,a_1,8~~
~~sub x_{20},x_4,a_0~~
fsd f_4,0(a_1)
addi a_1,a_1,8
fadd.d f_{10},f_8,f_2
fsd f_{10},0(a_0)
addi a_0,a_0,8
sub x_{20},x_4,a_0
bnz x_{20},Loop

10.

```
Loop: fld f4,0(a0)          Loop: fld T9,0(G0)
      fmul f2,f0,f2               fmul T10,R1,R0f2
      fdiv.d f8,f4,f2       →     fdiv.d T11,T10,T10
      fld f4,0(a1)                fld T12,R0(a1)
      fadd.d f6,f0,f4             fadd.d T13,f0,T12
      fsub.d f8,f8,f6             fsub.d T14,T11,T13
      fsd f8,0(a1)                fsd T14,0(a1)
```

11. 区别：显式重命名要确保物理寄存器堆具有比真实寄存器数目比ISA定义的寄存器数目更多。通过空闲列表和重命名列表来使程序正常运作。

隐式重命名中物理寄存器的数量与ISA规定保持一致，但其中仅存放最终已经写回的指令结果。处于推测状态的指令值由一些其他的结构保存，而且整个结构需要一个额外的表项来记录寄存器的最新值储存在哪。

优缺点：显式重命名不需要在重排序缓冲区创建大量的存储临时值空位，节省空间，但是由于在面对真实数据冲突(RAW)或结构冲突时依然会发生停顿，因此吞吐率不如隐式重命名。

隐式重命名与显式不同，它占用更多空间，但是吞吐率更高。

可能实现方式：显式的一种可能是方式是通过FL和RT列表，FL来选择空闲寄存器而RT来作对应映射关系

隐式的一种经典实现方式就是Tomasulo算法，在每个执行单元前引入一个保留站，在站中一系列寄存器内储存指令信息包括依赖关系，最终通过数据总线来广播执行结果。