

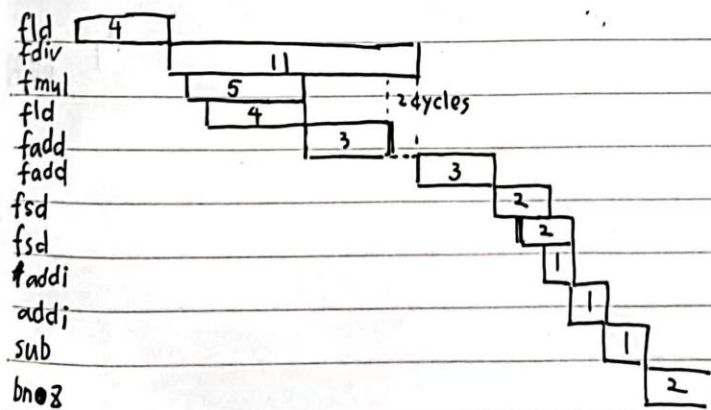
7. 顺序流水线, 忽略取指、译码, 处理器从发射到执行完成不同指令

所需总周期数如表.

考虑如下指令序列:

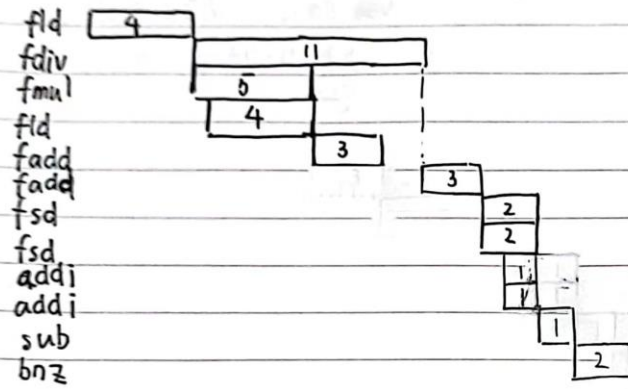
```
Loop: fld f2, 0(a0)
      fdiv.d f8, f0, f2
      fmul.d f2, f0, f2
      fld f4, 0(a1)
      fadd.d f4, f0, f4
      fadd.d f10, f8, f2
      fsd f10, 0(a0)
      fsd f4, 0(a1)
      addi a0, a0, 8
      addi a1, a1, 8
      sub x20, x9, a0
      bnez x20, Loop
```

1) 假设一条单发射流水线, 在没有数据冲突或分支指令时, 每周期均发射一条指令 (运算单元充足), 冲突或分支则暂停发射, 上述代码一次迭代需要多少周期执行完成?



总周期 = $4 + 11 + 3 + 2 + 1 + 1 + 1 + 2 = 25$ cycles

2) 假设一条双发射顺序流水线, 取指和译码带宽足够, 运算单元充足且数据在两级流水线间传递是无延迟的, 因此只有真数据冲突会导致流水线停顿, 则上述代码段一次迭代的周期是多少?



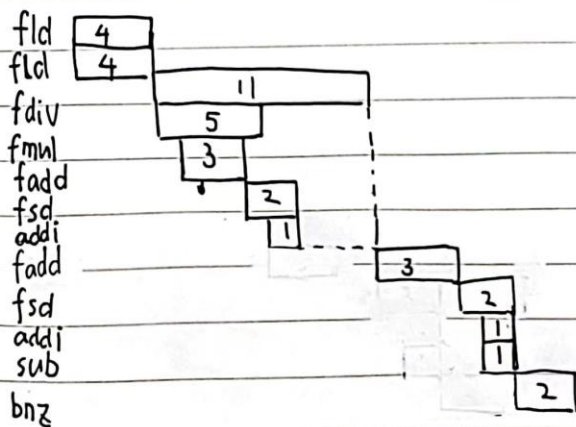
所需周期数 = 4 + 11 + 3 + 2 + 1 + 2 = 23 cycles

3) 调整指令的顺序排列, 使其在上世双发射流水线中完成一次迭代的周期数最少, 请给出调整后的指令序列和一次迭代的周期数。

调整后的指令:

```

Loop: fld f2, 0(a0)
      fld f4, 0(a1)
      fdiv.d f3, f0, f2
      fmul.d f2, f0, f2
      fadd.cl f4, f0, f4
      fsd f4, 0(a1)
      addi a0, a0, 8
      fadd.d f10, f3, f2
      fsd f10, 0(a0)
      addi a1, a1, 8
      sub x20, x4, a0
      bnez x20, Loop
    
```



调整后如上例所示, 需要迭代一次的周期数为22。

10. 考虑如下代码片段

```
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 $T_0 \sim T_63$ 的临时寄存器, 且 T_9 开始的寄存器可用于重命名, 写出重命名的指令序列。

```
Loop: fld T9, 0(a0)
```

```
      fmul.d T10, f0, f2
```

```
      fdiv.d T11, T9, T10
```

```
      fld T12, 0(a1)
```

```
      fadd 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。

显式重命名: 使用比 ISA 规定的寄存器数多得多的物理寄存器, 通过明确的重命名来更改寄存器名称来解决寄存器冲突。

隐式重命名: 在乱序处理器核心中, 引入重排序缓冲区 (ROB) 实现乱序执行和顺序写回。隐式重命名是指为每个指令分配一个 ROB 条目, 为目标寄存器创建一个新物理名称。

显式重命名的实现思路和逻辑比较简单, 容易设计且可有效解决数据依赖, 但需要额外的硬件开销 (物理寄存器等)。

隐式重命名不需要单独的物理寄存器, 可简化硬件, 但会增加 ROB 的压力。

显式重命名一般是通过重命名表的映射实现, 而隐式重命名中则需要 ISA Register File (ARF) 来存储已提交寄存器的状态, 在提交时通过 ROB 向 ARF 传输数据实现重命名。