

### 第三章 3、9、10、11 题

(1)可能发生数据依赖，条件是 **a2** 和 **a3** 指向的虚拟地址页映射到同一个物理地址页，且 **a0** 和 **a1** 的偏移量相同。因为存储指令 **sd** 会将数据写入物理地址页，而加载指令 **ld** 会从物理地址页读取数据，如果两条指令访问同一物理地址页的同一偏移量，那么在存储指令执行后，该物理地址页中的数据就已经被修改，而在加载指令执行时，该物理地址页的数据就会被读取并写入到 **a3** 寄存器中。

(2)不可能发生数据依赖。因为存储指令和加载指令访问的是不同的偏移量，它们不会相互影响。

(3)不可能发生数据依赖。因为虚拟地址页的大小为 **4KB**，即一个页表项对应的物理地址范围为 **4KB**，而 **ld** 指令中的偏移量是 **4096**，说明加载指令访问的是不同的页表项，因此不会和存储指令产生数据依赖。

9.(1)45 (2)18

(3)调整指令的排列顺序，可以进一步减少一次迭代所需的周期数。一种可能的调整方式如下：

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

这种调整方式可以将一次迭代所需的总周期数进一步减少到 **15**。

10.重命名后的指令序列如下：

```
Loop: fld f4,0(a0)
      fmul.d f2,f0,T9
      fdiv.d T10,f4,f2
      fld f4,0(a1)
      fadd.d f6,f0,f4
      fsub.d T11,T10,f6
      fsd T11,0(a1)
```

其中，**T9** 用于重命名 **f2**，**T10** 用于重命名 **f8**，**T11** 用于重命名 **f6**。其余寄存器保持不变。

11.显式重命名和隐式重命名都是指在处理器中使用重命名技术来解决数据依赖问题的方法，它们的区别在于重命名的实现方式和对程序员的可见性。

显式重命名是指程序员可以直接使用重命名后的寄存器，即程序员可以看到重命名的过程。显式重命名的优点是实现简单，可以通过编译器来实现，不需要对硬件进行修改。缺点是需要修改程序，增加了编译器的复杂度，而且程序员需要了解重命名的细节，编写代码时需要考虑重命名的影响。

隐式重命名是指程序员看不到重命名的过程，程序员使用的是原始的寄存器名称，而重命名的过程由处理器自动完成。隐式重命名的优点是程序员不需要了解重命名的细节，编写代码

时可以像使用原始寄存器一样使用重命名后的寄存器，不需要修改程序。缺点是实现比较复杂，需要对硬件进行修改，而且需要增加一些额外的硬件来支持重命名。

可能的实现方式包括：

基于编译器的实现：编译器可以在编译时对程序进行重命名，生成重命名后的代码。这种实现方式简单，但需要修改程序，增加了编译器的复杂度。

基于硬件的实现：处理器可以使用额外的硬件来支持重命名，例如重命名表、重命名寄存器等。这种实现方式比较复杂，但不需要修改程序，可以在运行时动态地进行重命名。

基于混合的实现：处理器可以同时使用编译器和硬件来支持重命名，编译器可以在编译时对程序进行重命名，同时处理器也可以使用硬件来支持重命名。这种实现方式综合了前两种实现方式的优点，但也增加了实现的复杂度。