姓名：李雅帆　　学号：2213041　　班级：信安班

4.16

(1) 350 ps　　1250 ps

(2) 1750 ps　　1250 ps

(3) 拆为 ID．　拆为后的处理器的时钟周期为 700 ps

(4) 35%

(5) 65%

4.19

(1) 在不处理数据冒险的流水线中执行代码，寄存器 $s5 的最终结果分别为：

$s0 的值为 7；　$s2 的值为 49；$s3 的值为 42；$s4 的值为 54；　$s1 的值为 22

4.22

(1) 流水线图

IF → ID → EX → MEM → WB

指令和数据共用一个存储器，如果同一个时钟周期内同时取指令和访问数据，将产生结构冒险，因此，流水线可能会出现阻塞情况。

(2) 可以。

通过对指令进行重新排列，可以使访存指令之后不立即需要使用其结果的指令能够得到执行，从而减少阻塞或需要插入的 NOP 指令数量。

(3) 不可以。

结构冒险通常需要通过硬件来处理。通常需要使用流水线互锁、旁路技术或者其他硬件策略来避免对指令执行的影响。

(4) 阻塞的发生是因为指令和取指令发生了冲突，即 MEM 和 IF 级有重合。因为只有 Load 和 Store 指令会用到 MEM，那么这种的结构冒险只由这两句指令引起。

即每出现一次 Load 指令或 Store 指令，就会引起一次阻塞。

Load (15%) + Store (11%) = 26%

4.26.

(1) EX to 1st only：
```
add $s0, $s1, $s2
sub $s3, $s0, $s4
add $s5, $s6, $s7
```
MEM to 1st only：
```
lw $s0, 8($s1)
add $s2, $s0, $s3
add $s4, $s5, $s6
```

EX to 2nd Only: add $s0, $s1, $s2
sub $s3, $s4, $s5
add $s6, $s0, $s7

MEM to 2nd only: lw $s0, 8($s1)
add $s3, $s2, $s4
add $s5, $s0, $s6

EX to 1st and EX to 2nd: add $s0, $s1, $s2
sub $s3, $s0, $s1
add $s4, $s0, $s2

(2) ① 2条；一、二行之间
② 2条；一、二行之间
③ 1条；二、三行之间
④ 1条：二、三行之间
⑤ 2条，二、三行之间

(3) add $s0, $s1, $s2
add $s3, $s4, $s0
add $s3, $s5, $s0
1个 nop

(4) CPI： 5%×3 + 20%×3 + 5%×2 + 10%×2 + 10%×3 + 50%×1 = 1.85
阻塞占比：$\frac{1.85-1}{1.85} \times 100\% \approx 45.6\%$

(5) 只有 MEM to 1st Only 需要1条 nop
CPI： 1×20% + 1 = 1.2
阻塞占比：$\frac{1.2-1}{1.2} \approx 16.7\%$

(6) EX/MEM转发 CPI为 1+1×(20% + 10% + 5%+10%) = 1.45
MEM/WB转发 CPI为 1+1×(5% + 20% + 10%) = 1.35
MEM/WB转发更好.

(7) EX/MEM $\frac{1.85 \times 120}{1.45 \times 110} \times 100\% \approx 128\%$
MEM/WB $\frac{1.85 \times 120}{1.35 \times 130} \times 100\% \approx 137\%$
全旁路 $\frac{1.85 \times 120}{1.2 \times 130} \times 100\% \approx 142\%$

(8) $\frac{1.2 \times 130}{1 \times 230} \times 100\% \approx 68\%$

(9) 因为 MEM to 2nd 中间只需要1个 nop，相当于用 MEM to 1st 填充

4.27

(1)
```
add  $s3, $s1, $s0
nop
nop
lw   $s2, 4($s3)
lw   $s1, 0($s4)
nop
or   $s2, $s3, $s2
nop
nop
sw   $s2, 0($s3)
```

(2)
```
add  $s3, $s1, $s0
lw   $s1, 0($s4)
nop
lw   $s2, 4($s3)
nop
nop
or   $s2, $s3, $s2
nop
nop
sw   $s2, 0($s3)
```

(3) 如果处理器没有实现冒险检测单元，但有旁路机制，那么在执行这段代码时会发生数据相关性的问题，例如 WAR和 WAW，这可能导致错误的结果或者程序挂起等问题。在这种情况下，处理器将不会检测数据相关性，并且不会采取任何措施来解决相关性导致的问题。这可能导致指令乱序执行，造成程序执错误。

(4) 时钟周期1：译码指令

时钟周期2：执行第一条指令

时钟周期3：执行第二条指令

时钟周期4：执行第三条指令

时钟周期5：执行第四条指令

时钟周期6：执行第五条指令

时钟周期7：写回结果

(5) 输入：ID/EX的寄存器和 EX/MEM的寄存器

输出：无

原因：在缺少旁路单元之后，由于R型指令和lw指令需要的Rd寄存器不同，所以需要冒险检测单元进行检查；是否需要寄存器进入EX和E也是需要分别的

6) 时钟周期1：无

时钟周期2：WAR类型的冒险，等待1个周期

时钟周期3：无

时钟周期4：WAW类型的冒险，等待1个周期

时钟周期5：无