1、

(1) a.流水线处理器：500ps（MEM），非流水线处理器：300 + 400 + 350 + 500 + 100 = 1650ps

b.流水线处理器：200ps（IF），非流水线处理器：200 + 150 + 120 + 190 + 140 = 800ps

(2) a.流水线处理器：500 × 5 = 2500ps，非流水线处理器：300 + 400 + 350 + 500 + 100 = 1650ps

b.流水线处理器：200 × 5 = 1000ps，非流水线处理器：200 + 150 + 120 + 190 + 140 = 800ps

(3) a.选择 MEM 级，划分后得到 MEM1 和 MEM2 级的延迟分别为 250ps，处理器的时钟周期为 400ps（ID）

b.选择 IF 级，划分后得到 IF1 和 IF2 级的延迟分别为 100ps，处理器的时钟期为 190ps（MEM）

(4) a. 15% + 10% = 25%

b. 30% + 15% = 45%

(5) a. 50% + 15% = 65%

b. 30% + 30% = 60%

2、

(1) a. I1 和 I3：RAW $1，I1 和 I2：WAR $6，I2 和 I3： RAW $6

b. I1 和 I2、I3：RAW $5，I1、I2 和 I3：WAR $5，I1 和 I3： WAW $5

(2)

	指令序列		指令序列
a	**lw $1, 40($6)** **add $6, $2, $2** **nop** **nop** **sw $6, 50($1)**	**b**	**lw $5, -16($5)** **nop** **nop** **sw $5, -16($5)** **add $5, $5, $5**

(3)

	指令序列		指令序列
a	**lw $1, 40($6)** **add $6, $2, $2** **sw $6, 50($1)**	**b**	**lw $5, -16($5)** **nop** **sw $5, -16($5)** **add $5, $5, $5**

(4)

a.无转发：(7 + 2) × 300 = 2700ps，充分的转发：7 × 400ps = 2800ps，加速比：2700/2800 = 0.96

b.无转发：(7 + 2) × 200 = 1800ps，充分的转发：(7 + 1) × 250 = 2000ps，加速比：1800/2000 = 0.90

(5)

	指令序列		指令序列
a	**lw $1, 40($6)** **add $6, $2, $2** **nop** **sw $6, 50($1)**	**b**	**lw $5, -16($5)** **nop** **nop** **sw $5, -16($5)** **add $5, $5, $5**

(6)

a.无转发：(7 + 2) × 300 = 2700ps，仅有 ALU 至 ALU 的转发：(7 + 1) × 360ps = 2880ps，加速比：2700/2880 = 0.94

b.无转发：(7 + 2) × 200 = 1800ps，仅有 ALU 至 ALU 的转发：(7 + 2) × 220 = 1980ps，加速比：1800/1980 = 0.91

3、

(1)

a.需要在 ALU 前的 MUX 增加一个输入端 0；需要在 WB 中增加一个比较数据存储器值与 0 的比较器；需要在 PC 前的 MUX 控制信号中加入比较器的结果，并且此时 MUX 的控制信号必须一直阻塞到 WB 之后才能进行选择。

b.需要为寄存器堆加入一个新的读寄存器地址写入端和读出端；需要在 EX 中 ALU 之后加入一个 MUX 来选择要写入数据存储器的值是 Rd（对于 swi 指令）还是 Rt（对于 sw 指令）。

(2)

a.需要为 ALU 前的 MUX 增加 1 位控制信号位，使其可以选择出新添加的输入端 0；需要为 PC 前的 MUX 控制信号加入比较器的结果，来决定是否需要修改 PC。

b.需要为新加入的 MUX 增加控制信号。

(3)

a.会。bezi 指令会引入一个新的控制冒险，它的 PC 直到 WB 才能确定是否需要修改，而且无法通过缩短分支延迟的方法提前判断出分支是否执行，因而会使已有冒险导致的阻塞更加严重。

b.不会。swi 指令不改变任何寄存器的值，不会引起数据冒险，同时它也不是分支指令，不会引起控制冒险。

4、

(1)

	指令序列		指令序列
a	**lw $1, 40($6)** **nop** **nop** **add $2, $3, $1** **add $1, $6, $4** **nop** **sw $2, 20($4)** **and $1, $1, $4**	**b**	**add $1, $5, $3** **nop** **nop** **sw $1, 0($2)** **lw $1, 4($2)** **nop** **nop** **add $5, $5, $1** **sw $1, 0($2)**

(2)

	指令序列		指令序列
a	**I1: lw $7, 40($6)** **I3: add $1, $6, $4** **nop** **I2: add $2, $3, $7** **I5: and $1, $1, $4** **nop** **I4: sw $2, 20($4)**	**b**	**I1: add $7, $5, $3** **I3: lw $1, 4($2)** **nop** **I2: sw $7, 0($2)** **I4: add $5, $5, $1** **I5: sw $1, 0($2)**

(3)

a.I2 获得的$1 的值是 I1 前面的那一条指令中$1 的值，此时$1 的值还没有被 lw 指令更新。

b.I4 获得的$1 的值是 I3 前面的那一条指令（即 I2）中$1 的值，此时$1 的值还没有被 lw 指令更新。

(4)

输入信号：冒险检测单元需要在 EX 中检查 R 型指令和 lw 指令的 Rd 寄存器，在 MEM 中检查目标寄存器号，因此需要添加 ID/EX 流水线寄存器的 Rd 和 EX/MEM 的输出寄存器作为输入信号，图 4-60 中已经有 ID/EX 的 Rt 这个输入信号了，不需要再次添加。

输出信号：不需要额外的输出信号。

5、

I1: 1）GRF 的 A3 和 WD 的输入接反了

I2：1）没有立即数扩展功能；2）ALU 输入源中不包含立即数 3）GRF 的 A3 和 WD 的输入接反了

I3: 1）没有立即数扩展功能 2）PC 无法进行跳转

I4: 1）没有立即数扩展功能 2）PC 无法进行跳转

I5: 1）没有立即数扩展功能；2）ALU 输入源中不包含立即数 3）GRF 的 A3 和 WD 的输入接反了