

1.

不可以，因为上一条指令的ALU结果是在EX阶段结束后才有的，而beq的比较发生在ID阶段中，是和上一条指令的EX阶段同时进行的，必须要在beq指令的ID阶段开始前转发才行，所以必须要加一个阻塞（否则在转发过来前，比较已经完成了）。

此外，假如beq前是lw这种指令（load-use冒险），将lw指令的EX阶段ALU结果转发是没有意义的，因为lw的结果是在MEM阶段结束后才产生的，此时需要阻塞两个周期且把MEM阶段结束后的结果转发到beq指令的ID阶段。

2. a)

0x00与0x08之间存在数据冒险，不过只需转发即可，0x08与0x0c之间存在控制冒险，需要阻塞三个周期并转发，0x0c与0x10之间存在load-use冒险，需要阻塞一个周期并转发，0x14与0x08之间存在load-use冒险，由于中间间隔了指令，所以只需要转发，0x14与0x0c之间存在数据冒险，但是只需要转发即可，0x18的j指令后续存在控制冒险，需要阻塞一个周期。

综上，总共执行的指令数为 $2 + 1000 \times (5 + 5) + 3 + 2 = 10007$ ，执行时间为 $10007 + 4 = 10011$ 个时钟周期数。

2. b)

调换之后，0x0c与0x10之间不需要阻塞，只需转发即可。

综上，总共执行指令数为 $2 + 1000 \times (5 + 4) + 3 + 2 = 9007$ ，执行时间为 $9007 + 4 = 9011$ 个时钟周期。

2. c)

第一次执行到beq时，转发单元将EX/MEM中的ALU计算结果转发到ID阶段中进行beq判断。此后执行到beq时，由于0x14的lw指令处于WB阶段，因为寄存器的先写后读特性，无需转发。

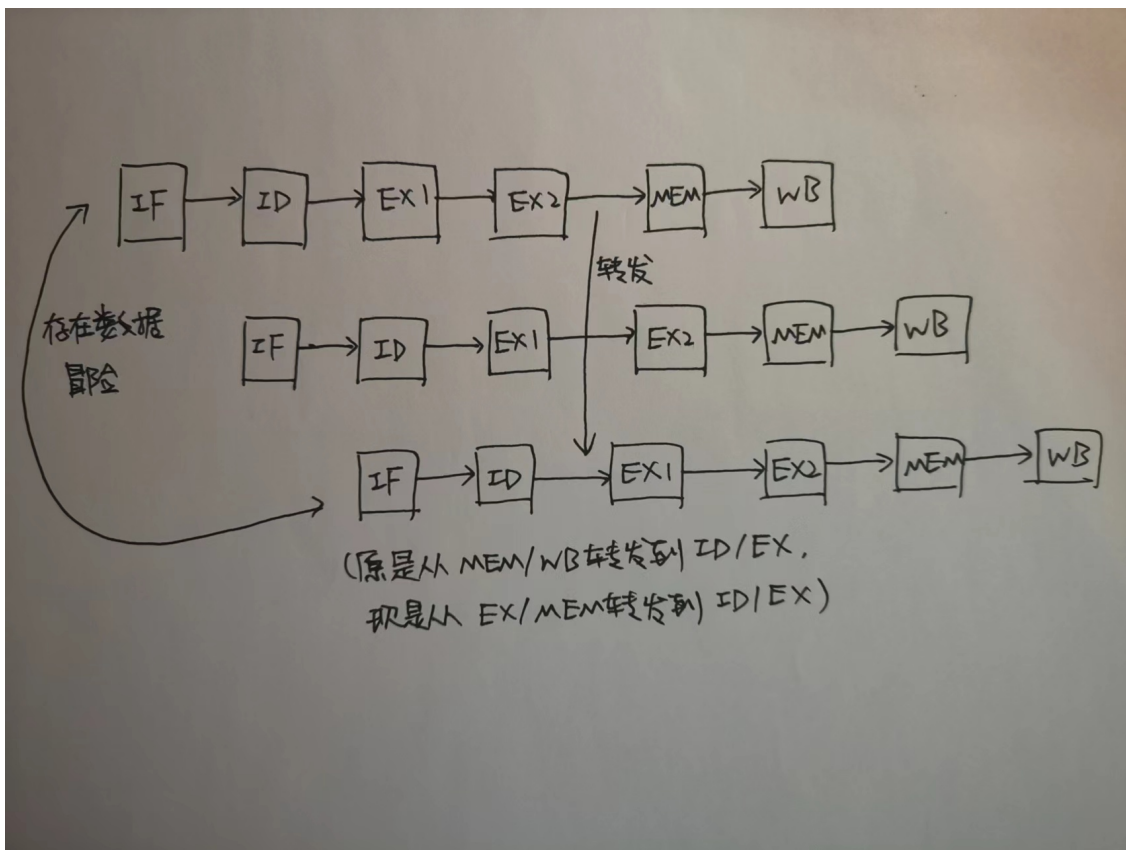
采用ID阶段进行提前分支判断的方式，则0x08与0x0c之间需要阻塞一个周期，0x0c与0x10之间需要阻塞一个周期，0x18后需要阻塞一个周期。

综上，总共执行指令数为 $2 + 1000 \times (5 + 3) + 1 + 2 = 8005$ ，执行时间为 $8005 + 4 = 8009$ 个时钟周期。

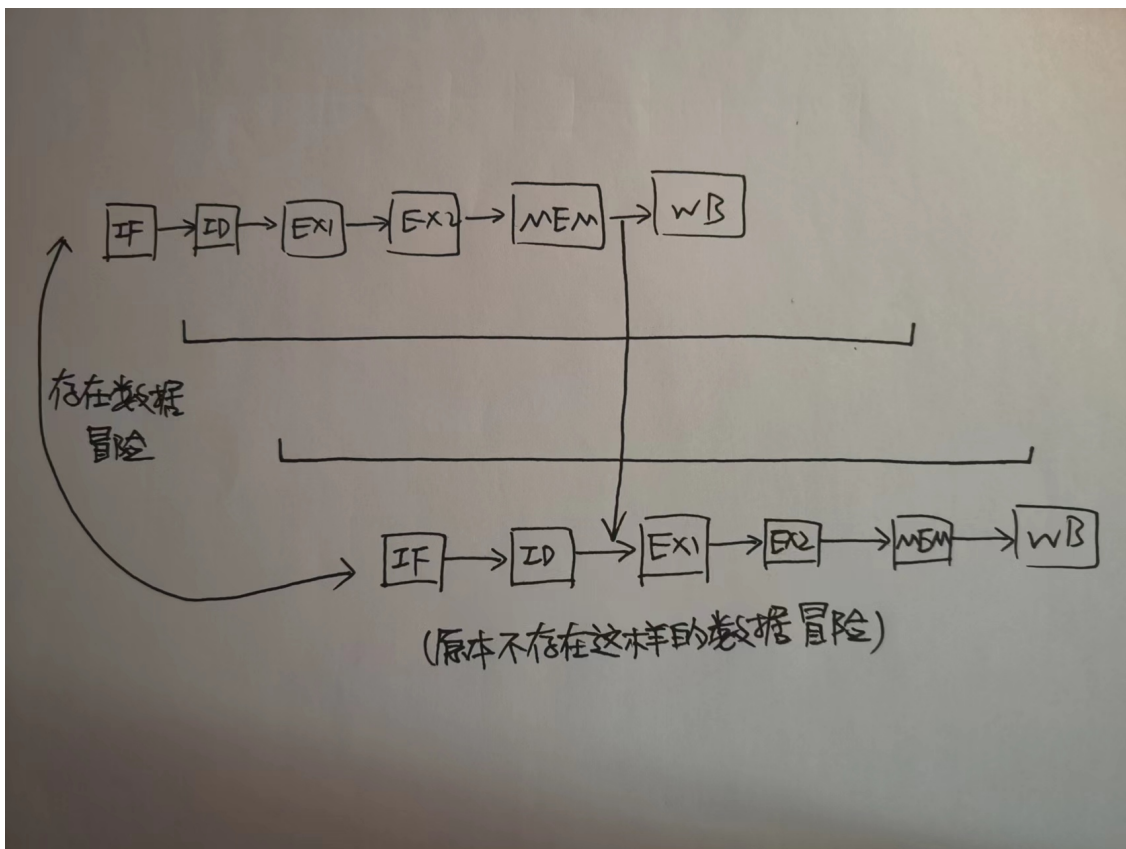
3. a)

需要修改。修改如下：

- 发生当前指令和前前条指令的数据冒险时，原本是从MEM/WB转发到ALU输入端，而现在变为从EX/MEM转发到ALU输入端：



- 发生当前指令和前前条指令的数据冒险时，原本不需要转发，而现在需要从MEM/WB转发到ALU输入端：



3. b)

冒险检测单元需要修改，因为一般的数据冒险（两个紧邻的指令，后者计算用到了前者的结果，比如两个add）之间也需要阻塞一个周期（这个或许可以算一种新的冒险，因为原本不需要阻塞），而load-use冒险之间现在需要阻塞两个周期。

3. c)

如果后续指令使用前面指令的计算结果，那就阻塞足够长周期数，直到计算完成，EX结果更新后，再继续正常的流水线。（可以针对MEM和EX各增加一个额外信号，当阶段正在进行时信号为0，阶段完成后再置为1。当其他指令需要该指令结果而当前信号为0时stall，直到信号为1后再进行转发）