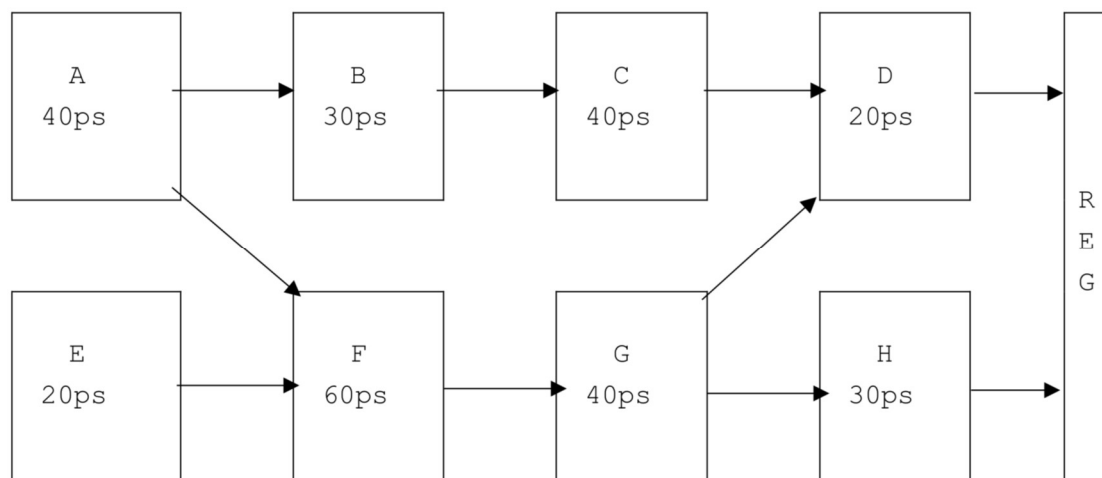


1. 判断以下说法是否正确

- () (1) 流水线的深度越深，总吞吐率越大，因此流水线应当越深越好。
- () (2) 流水线的吞吐率取决于最慢的流水级，因此流水线的划分应当尽量均匀。
- () (3) 假设寄存器延迟为 20ps，那么总吞吐率不可能达到或超过 50 GIPS。
- () (4) 数据冒险总是可以只通过转发来解决。
- () (5) 数据冒险总是可以只通过暂停流水线来解决。

2. 一条三级流水线，包括延迟为 50ps, 100ps, 100ps 的三个流水级，每个寄存器的延迟为 10ps。那么这条流水线的总延迟是_____ps，吞吐率是_____GIPS。

3. A~H 为 8 个基本逻辑单元，下图中标出了每个单元的延迟，以及用箭头标出了单元之间的数据依赖关系。寄存器的延迟均为 10ps。



- (1) 计算目前的电路的总延迟
- (2) 通过插入寄存器，可以对这个电路进行流水化改造。现在想将其改造为两级流水线，为了达到尽可能高的吞吐率，问寄存器应插在何处？获得的吞吐率是多少？
- (3) 现在想将其改造为三级流水线，问最优改造所获得的吞吐率是多少？

4. 一个只使用流水线暂停、没有数据前递的 Y86 流水线处理器，为了执行以下的语句，至少需要**停顿**多少个周期？一共需要运行多少个周期？

<code>irmovq \$1, %rax</code> <code>irmovq \$2, %rbx</code> <code>addq %rax, %rcx</code> <code>addq %rbx, %rdx</code> <code>halt</code>	<code>rrmovl %eax, %edx</code> <code>mrmovl (%ecx), %eax</code> <code>addl %edx, %eax</code> <code>halt</code>	<code>irmovl \$0x40, %eax</code> <code>mrmovl (%eax), %ebx</code> <code>subl %ebx, %ecx</code> <code>halt</code>
---	---	---

5. 考虑 Y86 中的控制逻辑。jXX 总是预测分支跳转。

(1) 写出流水线需要处理 ret 的条件 (ret 对应的常量为 IRET)：

(2) 发现 (1) 中的条件后，流水线寄存器应如何设置？(选填 stall, bubble, normal)

	Fetch	Decode	Execute	Memory	Writeback
ret					

(3) 写出流水线需要处理jXX分支错误的条件（jXX对应的常量为IJXX）：

(4) 发现(3)中的条件后，流水线寄存器应如何设置?(选填stall, bubble, normal)

	Fetch	Decode	Execute	Memory	Writeback
JXX错误					

(5) 写出流水线需要处理load/use hazard(加载/使用冒险)的条件：

(6) 发现(5)中的条件后，流水线寄存器应如何设置?(选填stall, bubble, normal)

	Fetch	Decode	Execute	Memory	Writeback
load/use					

(7) 在Y86流水线中，是否存在一组指令序列可能同时满足上面的三个条件?是否可能同时出现上面的两种条件?

(8) 考察以下两组指令序列，写出发现冒险时应该如何设置流水线寄存器。

...					
popq %rsp					
ret					
	Fetch	Decode	Execute	Memory	Writeback
load/use+ret					
	Fetch	Decode	Execute	Memory	Writeback
jXX+ret					

(9) 根据以上各小问，补全以下pipe.hcl中的控制逻辑。

```
bool F_bubble = 0;
bool F_stall =
    E_icode in { IMRMVQ, _____ } &&
    E_dstM in { _____, _____ } ||
    # Stalling at fetch while ret passes through pipeline
    IRET in { _____, _____, _____ };

bool D_bubble =
    # Mispredicted branch
    (E_icode == IJXX && !_____) ||
    # Stalling at fetch while ret passes through pipeline
    # but not condition for a load/use hazard
    !(_____ in { IMRMVQ, IPOPOP } &&
      _____ in { d_srcA, d_srcB }) &&
    _____ in { D_icode, E_icode, M_icode };
bool D_stall =
    # Conditions for a load/use hazard
    E_icode in { IMRMVQ, IPOPOP } &&
    E_dstM in { d_srcA, d_srcB };
```