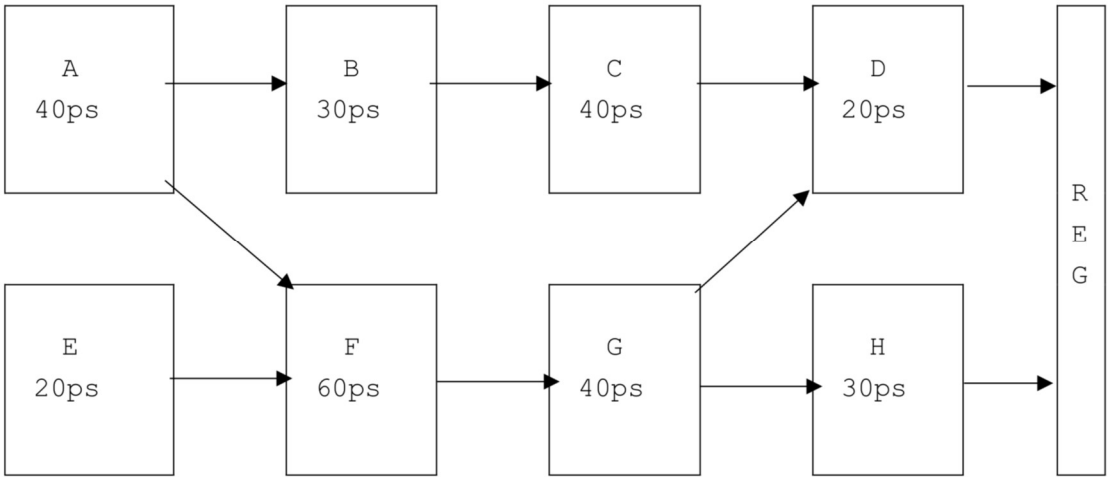
1. 判断以下说法是否正确

( ) (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?

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

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。

| ...<br>popq %rsp<br>ret | | | je L1<br>addq %rdx, %rax<br>.L1:<br>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 { IMRMOVQ, _______} &&
    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 { IMRMOVQ, IPOPQ } &&
       ______ in { d_srcA, d_srcB }) &&
    ______ in { D_icode, E_icode, M_icode };
bool D_stall =
    # Conditions for a load/use hazard
    E_icode in { IMRMOVQ, IPOPQ } &&
     E_dstM in { d_srcA, d_srcB };
```

| 得分 |
| --- |
| |

第四题（20 分）

请分析32位的Y86 ISA中新加入的一组条件返回指令：cretXX，其格式如下。

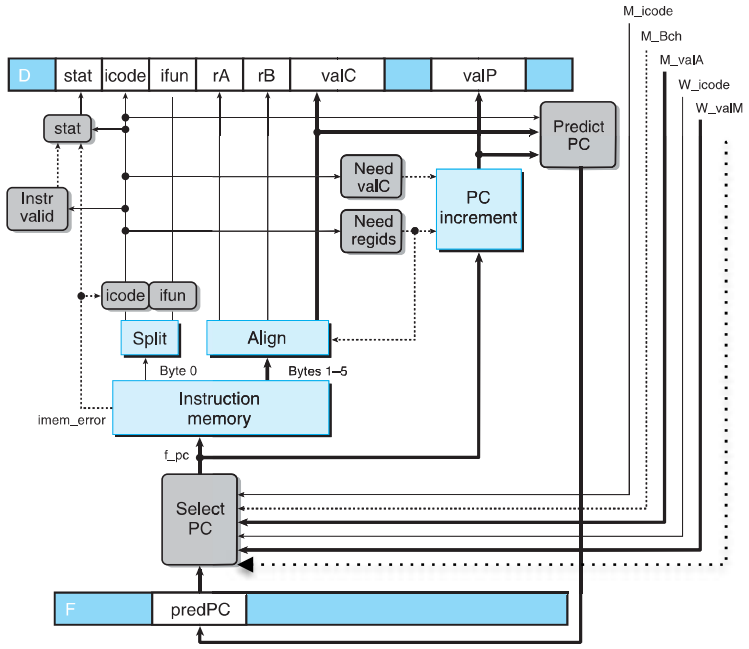
| cretXX | 9 | fun |
| --- | --- | --- |

类似cmovXX，该组指令只有当条件码(Cnd)满足时，才执行函数返回；如果条件不满足，则顺序执行。

1．若在教材所描述的SEQ处理器上执行这条指令，请按下表补全每个阶段的操作。需说明的信号可能会包括：icode, ifun, rA, rB, valA, valB, valC, valE, valP, Cnd; the register file R[], data memory M[], Program counter PC, condition codes CC。其中对存储器的引用必须标明字节数。如果在某一阶段没有任何操作，请填写none指明。

| Stage | cretXX Offset |
| --- | --- |
| Fetch | |
| Decode | |
| Execute | |
| Memory | |
| Write back | |
| PC update | |

2.为了执行cretXX指令，我们需要改进教材所描述的PIPE处理器，在W（Write Back）阶段引入流水线寄存器_____，并将其连接到PC选择器（Select PC）以便有条件地更新PC。假设改进后的处理器总是预测函数返回条件不满足，则如果返回条件满足时，一共会错误取指____条指令。




3.在2中改进的PIPE处理器上执行cretXX指令时，发生预测错误时的判断条件和各级流水线寄存器的控制信号应如何设置？

| Condition | Trigger |
|---|---|
| Mispredicted cret | ( = ICRETXX && )\|\|<br>( = ICRETXX && ) |

| Condition | F | D | E | M | W |
|---|---|---|---|---|---|
| Mispredicted cret | | | | normal | normal |

4. PIPE 处理器上处理器上执行如下代码片段，

```
0x000:    xorl %eax, %eax
0x002:    popl %esp
0x004:    cretne
```

（1）是否会发生 load-use 和 misprediction cret 组合的 hazard 情况？

答：

（2）如果此时“popl %esp”在流水线的 Execute 阶段，请问此时，各级流水线寄存器的控制信号应如何设置？

| Condition | F | D | E | M | W |
|---|---|---|---|---|---|
| Combination | | | | normal | normal |