封装成单周期 CPU

宏观 PC

我们的需求是要让我们的 CPU 从外部看上去是一个单周期的 CPU（具体的原因在前一章有阐述）。但是实际上，我们的 CPU 是一个五级流水的并行 CPU。所以本质上我们要实现一套机制，来让我们的 CPU 满足这个需求。

为了检验同学们的实现效果，我们引入宏观 PC 这一概念。宏观 PC 表示整个 CPU “宏观”运行指令所对应的PC地址。所谓“宏观”指令，表示该指令之前的所有指令序列对 CPU 的更新已完成，该指令及其之后的指令序列对 CPU 的更新未完成。具体实现上，宏观 PC 通常上是以某一个流水级 PC 作为界限，作为输出端口输出出来。这个流水级一般是你 CP0 所在的流水级。

精确异常

对于异常，我们能明确指出是哪条指令导致了异常，并称这条指令为**异常受害指令**。精确异常的特性是，在异常受害指令**前面的所有指令都执行完毕**，而**受害指令及其后续指令都像从来没有开始**（准确说是当异常处理结束后重新执行这些指令，与未发生异常时执行这些指令的效果一样）。这样的处理思路使得从使用者的角度来看， CPU 执行异常处理是顺序执行的，从而隐藏了流水线设计的细节。

清空流水线

为了达到精确异常的效果，我们需要在异常发生的时候清空流水线，以避免宏观 PC 之后的指令被执行。清空流水线一方面是要清空宏观 PC 之后的指令所在的流水线寄存器，即插入 `nop`。

在了解了这点以后，我们可以总结一下我们对于流水线寄存器的控制。流水线寄存器需要接受多种控制信号，如复位信号，阻塞信号，刷新信号，请求信号。所以有可能同时会有多个信号控制同一个寄存器，那么寄存器该展现怎样的行为呢？这是一个需要考虑的事情。

例如在 D 级处于被阻塞状态时发生 `Req`，那么 D 级流水线寄存器就应该立刻被清空，而不是保持原值；正在 `Req` 的时候发生`reset`，那么 CPU 应该立刻复位，而不是进行异常处理。此处处理不当可能造成评测时缺少中断的情况。实现上，可按如下优先级：

信号	**优先级**
reset	最高，复位大于一切。
Req	次高，中断请求比内部阻塞重要。

信号	优先级
flush / stall	最低，流水线信号，外部人员看不到。

接下来我们考虑，哪些寄存器中的位段需要优先级。只有两个，一个是 PC 寄存器，原因之前论述过了；另一个是 CP0 `Cause` 寄存器的 `BD` 位。它在 flush 的时候需要保持原来的信息，因为在外部去看的话，会发现宏观 PC 是相同的，但是延迟槽标记是不同的，这显然是不正确的。如果在延迟槽指令被阻塞时产生中断，并且 `nop` 没有流水延迟槽标记，那么 `EPC` 就会被设置错误的值，无法通过评测。

清空流水线的另一个方面就是避免异常受害指令和其之后的指令产生影响（比如写寄存器，写 DM），这一点将在下一小节“确定 CP0 的位置”中讨论。

确定 CP0 的位置

CP0 需要放置在某个具体的流水级上，我们认为宏观 PC 所在的流水级就是 CP0 的流水级。为了满足宏观 PC 的性质，CP0 所处的位置不能够太靠前，比如设置在 F 级，那么异常会发生在之后的流水级，那么就检测不到这个异常了（如果宏观 PC 就是异常受害指令的话）。但是也不能太靠后，比如在 W 级，因为我们需要清除受害指令之后的指令造成的影响，但是此时 `store` 类指令已经修改了外设，清除影响较为困难。

因此，请根据所学挑选你的 CP0 在流水线上的位置，需要强调，没有标准答案或者最优答案。

流水异常码

异常信号 ExcCode 应该流水到 CP0 所在的流水级，而不能直接提交到 CP0。

这是因为 CPU 实际上是并行的，直接提交到 CP0 可能会导致后面的指令发生异常的时间比前面指令发生异常的时间要早。例如 `sw` 后接 `j` 指令。如果这两个都是异常指令，那么 `j` 在 D 级产生异常，`sw` 在 M 级产生异常（假设这个异常是超范围了）时不流水，就将先处理 `j` 异常，显然不符合我们的要求，因为 `sw` 异常被忽略了（`sw` 继续往后流，前面的流水级开始流异常处理程序，等异常返回之后，就会直接到跳转目标指令了，`sw` 的异常没有得到处理）。我们将异常信号流水以后，就可以先处理 `sw` 异常，然后运行到 `j`，再处理 `j` 异常。

写入 EPC

发生异常的一个重要行为是将中断指令的 PC 写入 EPC，就像函数跳转之前，要将返回地址写入 `$ra`。更严谨的说，对于异常情况只要考虑异常指令是不是延迟槽指令，如果是延迟槽指令，那么存的是异常指令的 PC - 4，如果不是，那么就存异常指令的 PC。

这样造成的结果就是，返回的时候将重新执行异常指令（如果异常处理程序不对 EPC 进行修改的话）。这里的 PC 指的都是宏观 PC。

思考题

倘若中断信号流入的时候，在检测宏观 PC 的一级如果是一条空泡（你的 CPU 该级所有信息均为空）指令，此时会发生什么问题？在此例基础上请思考：在 P7 中，清空流水线产生的空泡指令应该保留原指令的哪些信息？

思考题

为什么 `jalr` 指令的两个寄存器不能相同，例如 `jalr $31, $31`？