**流水线寄存器**

为了实现让多条指令同时执行，要将五个模块的执行过程进行组合，从时间的维度进行重叠，从而加速 CPU 指令执行的吞吐量，提高 CPU 执行指令的效率，因此增加流水线寄存器。

流水线寄存器用来控制整个数据通路中不同流水阶段里信号、数据的传递。五级流水分为 IF、ID、EX、MEM 和 WB 五个阶段，我们在段与段之间分别设置Reg IF/ID、Reg ID/EX、Reg EX/MEM、Reg MEM/WB 这四个流水线寄存器，用来连接流水的五个阶段。

流水线寄存器作用：① 将各段的工作隔开，使得它们不会互相干扰 ② 保存相应的处理结果 ③ 向后传递后面将要用到的数据或者控制信息，并共享给其他指令。

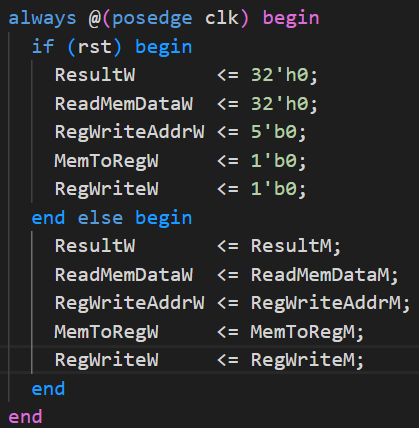
**将数据向后传递**

在后面阶段所需要的数据必须经由流水线寄存器进行传递。比如 LW 指令需要将数据存储器中的数据写入目标寄存器中，目标寄存器最终写入的数据是由指令在第四阶段 MEM 才获取到的，而在第五阶段 WB 才写回。因此，rd（目标寄存器）必须经由全部四个流水线寄存器传递。

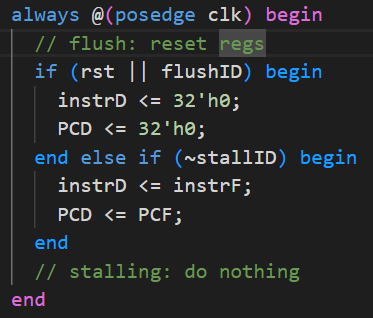
**将控制信号向后传递**

控制信号也需要进行传递。控制信号的生成与单周期 CPU 中的生成过程一致：在取指令之后，ID 模块对指令译码并生成相应的控制信号。但是这些控制信号很多时候一直到指令执行的第五阶段才会用到，因此控制信号可以随着其他数据一同在流水线寄存器中进行前递。

以Reg MEM/WB为例，关键代码如下：



针对数据相关、控制信号冲突等问题，Hazards Unit控制各个流水寄存器的流通。当reset或flush信号有效时，则进行复位操作。当需要stall整个流水线时，不进行操作。否则，将寄存器中的内容按既定规则进行更新。关键代码如下：



顶层模块是整个 CPU 的控制模块，通过连接各个子模块来达到运行 CPU 的目的。在顶层模块中将所有需要使用到的模块进行实例化，并利用wire导线进行连接，在顶端模块内部定义用于连接各个模块输入输出的内部端口。这样就能让全部模块连接起来，成为完整的CPU电路。以第四阶段MEM为例，关键代码如下：

