流水线CPU报告

1. 流水线的意义、

之前完成的单周期CPU指的是处理器执行每一条指令都要在一个时钟周期内完成，而只有完成一条指令才能进入下一条指令的执行过程，所以时钟周期受最长执行时间的那条指令的限制。而流水线的思想就是为了加速指令整体的执行序列，将整个逻辑按功能分阶段分成几个小的逻辑，而这几个逻辑能够并行执行，通过增加数据的吞吐量来提高执行效率。

1. 流水线的设计

流水线的设计有多种想法，不过当然按照最经典的划分想法将执行一条指令的过程分成五个阶段：取指、译码、执行、访存和写回。

事实上在上学期单周期CPU的设计中我们也是将指令按照这五个步骤来进行的，只是在单周期中并不存在多个指令在不同阶段的并行执行，而要将单周期CPU向多周期CPU进行转变也就是要能让多条指令的不同阶段能够做到并行执行，首先就是要将整个执行逻辑拆分开来，打断各个子模块之间的“路径”，但是其之间的信号流通没有改变，因而要在每两个子模块间添加一组流水线寄存器，通过将上一条指令的前阶段得出的东西存入流水线寄存器从而使得下一条指令不用等待上一条指令完全完成就可以进入并且还不会干扰上一条指令的执行。

下面对每个阶段和每个阶段之间的流水线寄存器进行详细描述：

1. 取指阶段（IF\_stage）

该阶段的功能就是产生新的pc值和读取指令存储器，所以包含了单周期CPU中pc\_reg和instr\_memory两个子模块，因为instr\_memory之后要作为外部存储器（大概是ip核？），所以在这个阶段不加入。

1. IF\_ID流水线寄存器

在取指和译码阶段之间需要传递的信号是pc和读出的32位指令，因为在执行阶段计算下一个pc的值和跳转指令的目标地址时需要用到当前模块的pc所以pc需要逐级传递否则有可能被下一条指令更新丢失，保证pc和所在阶段执行的指令一致。

1. 译码阶段（ID\_stage）

译码阶段要做的就是将传入的指令根据指令集进行译码然后读取寄存器堆，所以包含instr\_decode和registers两个模块，同时还要收到写回阶段传回来的rd信号，可以在此阶段将要写回寄存器的数据写回。

1. ID\_EX流水线寄存器

在译码和执行阶段需要传递的信号首先还是需要有pc因为pc一直需要保存到最后写回阶段写回寄存器中，然后是译码阶段指令中包含的数据，可能是立即数也可能是寄存器堆读出来的数据。同样也需要传递rd信号直到写回阶段，但是可以不经过执行阶段。

1. 执行阶段（EX\_stage）

该阶段要做的事情就是alu的算术运算以及pc新的值的计算（包含得到跳转的地址值），可以说alu的结果一般是算术运算指令的结果、新的pc地址等，而得到什么结果取决于不同的数据来源也就是上个阶段传递过来的信号和控制器传递的控制信号。因而需要包含的是alu模块，branch\_judge模块，加法器模块（用于计算pc+4/imme），mux选择器模块（alu数据来源的选择，选择pc顺序执行或者pc跳转，jalr选择（即选择jalr还是跳转/顺序））。

1. EX\_MEM流水线寄存器

在执行阶段和访存阶段之间需要保存和传递的信号是ALU\_result,pc\_jump,pc\_order,Rd\_data2,imme,其中ALU\_result会在访存阶段作为访存对应的地址使用，在写回阶段作为运算结果写入写回寄存器，pc\_jump和pc\_order则是在写回阶段作为pc新值的待选写回寄存器，Rd\_data2则是在访存阶段作为写入存储器的数据使用，imme同样也在写回阶段使用，rd也需要继续传输。

1. 访存阶段（MEM\_stage）

对于该阶段由于实验要求指令存储器和数据存储器都是和cpu分开的，因而只需要根据输出地址和数据给数据存储器，或者接收数据存储器读出的数据即可，直接在顶层的数据通路进行体现即可。

1. MEM\_WB流水线寄存器

而从访存阶段到写回阶段所传递的需要使用的信号有ALU\_result,pc\_jump,pc\_order,rd，imme,loaddata,其中前面四种都是之前传递到写回阶段的信号，rd是直接传输到下一个周期的译码阶段直接找寄存器，而最后的loaddata就是在访存阶段读出来的数据，是新的需要传递的数据。

1. 写回阶段（WB\_stage）

写回阶段主要的任务就是通过四个选择器选择需要的数据写回寄存器堆（第一个是alu结果和读出的数据选择，然后是将其作为输入和传入的顺序运行pc进行选择而进行jalr选择，还有输入是顺序运行的pc和立即数跳转的lui选择，最后是将lui选择和jalr选择结果根据是否为u-type指令而选择出最后输出的一个数据）。

1. 最后则是将流水线按顺序以及其中的流水线寄存器连接起来传递信号即可。

完成之上的步骤只是完成了基础的流水线架构，但是还有类比单周期CPU的控制器部分，也就是产生控制信号并填入数据通路中。

1. 控制器的设计

控制器实际上并不需要重新设计，控制信号的产生和单周期的CPU产生的逻辑是相同的，因而只需要在数据通路中将控制器产生的对应的控制信号填入即可，需要注意的只是需要扩展流水线寄存器将这些控制信号存入流水线寄存器从而保存控制信号的现场，否则会导致控制信号的错位。

1. ID/EX流水线寄存器的扩展

译码阶段是第一次产生跳转等控制信号的阶段，然后要将这些信号通过ID/EX流水线寄存器进行保存。因而要对ID/EX流水线寄存器进行控制信号的扩展。

1. EX/MEM流水线寄存器的扩展

执行阶段使用了ALUsrc,ALUctl以及beq等b系控制信号之后就不再使用所以可以不用保存到EX/MEM流水线寄存器，而jal，jalr，RegWrite等信号需要继续保留至写回阶段，因而这些信号需要继续保存。

1. MEM/WB流水线寄存器的扩展

到该部分减少的控制信号就是访存阶段用到的MemRead和Write以及RW\_type信号，其他信号仍然需要保存。