

MIPS 五级流水实验报告

张嘉恒

2014 级 ACM 班

上海交通大学

2016 年 9 月 10 日

目录

1	简介	3
2	整个工程的具体介绍和实现效果	3
2.1	工程的指导思想	3
2.2	工程内部文件的具体实现细节	3
2.2.1	module top	3
2.2.2	module Alu, Ctr, AluCtr	4
2.2.3	module HazardCtr	4
2.2.4	module ForwardCtr	4
2.2.5	module data_memory	4
2.2.6	module inst_memory	4
2.2.7	module register	4
2.2.8	module signext	4
2.3	工程实现的效果图	4
2.4	实现上的几个优点	5
2.5	实现上的不足之处	5
3	遇到的一些挑战以及解决方式	6
3.1	关于模拟编程软件	6
3.2	关于变量命名	6
3.3	关于一些细节改动	6

目录	2
3.4 关于调试与测试	6
4 未来发展和完善的方向	7
4.1 跳转预测	7
4.2 缓存	7
4.3 精确中断	7
4.4 上板子和操作系统	7
5 学习感悟	7
6 参考文献	8

1 简介

我所设计的 MIPS 五级流水非常简单，经过测试可以实现绝大部分的功能，可以保证李亚明测试数据的正确性。我的整个工程主要参考了电子信息与电气工程学院的电子技术实验的实验用书。这本书从实验 1 到实验 6 详细讲述了五级流水线的实现过程，并且提供了部分代码，使用起来比较方便。实验所使用的语言为 verilog HDL。所使用的编程仿真软件为 ISE Design suite。整个工程由顶层文件 top.v 统领，控制整个五级流水的连线和流程。其他的各个阶段由不同的文件控制完成，最后达到流水的效果。整个工程首先实现了比较简单的单周期，然后实现五级流水，为了避免 control hazard 和 data hazard，专门又设计了 hazardcontrol 模块以及 forwarding 模块，最后的效果还是不错的。

2 整个工程的具体介绍和实现效果

2.1 工程的指导思想

1. 整个工程文件存在一个顶层文件 top.v，实现其余部件的逻辑链接。
2. 五级流水的每一个阶段都很关键，先将其分离出来完成，再实现流水操作。
3. 本着由简到繁的原则，先实现单周期，后实现流水线，最后解决冒险。

2.2 工程内部文件的具体实现细节

2.2.1 module top

这一模块为顶层文件，实现了所有模块的有序拼接，控制着整个 MIPS 五级流水过程，是工程的核心。其中包括对各个阶段所需要的变量的定义，所有部件之间的连线和逻辑关系。还有所有变量的初始化过程，和各个模块的实例化。还有当冒险跳转等问题发生时的解决方案。

2.2.2 module Alu, Ctr, AluCtr

这三个模块主要控制运算，根据命令中的 *opcode* 对流水进行控制，实现不同的运算功能。

2.2.3 module HazardCtr

这个模块是用来解决一部分冒险的。如果是与跳转有关的冒险，则要使得已经上线的指令全部下线。出现这种情况时，传递几个 *stall* 参数给 *top*，让相关指令下线。如果是与 *lw* 有关的冒险，则传递参数给 *top* 让其等待数据。

2.2.4 module ForwardCtr

这个模块是通过预取来解决数据冒险的，同样是发现此类情况就传递参数给 *top* 文件，直接通过寄存器拿到结果进行运算。

2.2.5 module data__memory

这个模块主要实现了对内存数据的读写操作。

2.2.6 module inst__memory

这个模块主要是读入文件取指令。

2.2.7 module register

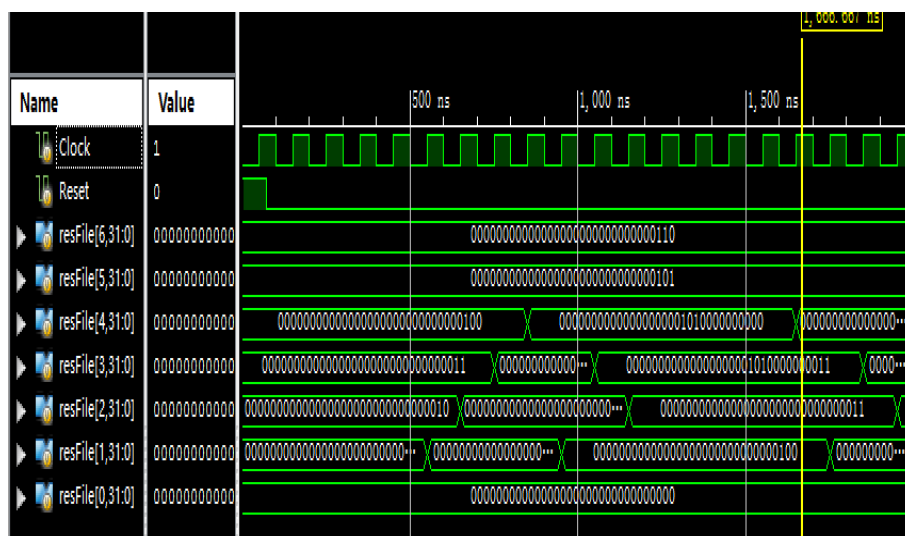
这个模块主要实现了对寄存器的读写操作。

2.2.8 module signext

这个模块实现了符号扩展。

2.3 工程实现的效果图

比如运行李亚明数据的第一个循环后的波形图。其中寄存器的初始值与编号相同。其中寄存器中的值都用 2 进制数表示。



2.4 实现上的几个优点

1. 由于实现上是根据电院的教材一步步学习的，所以每一个部分都是一个实验，可以单独测试，由易到繁，检查起来比较方便。

2. 各个独立模块的实现逻辑关系都非常简单，还添加了简要注释，非常便于他人理解，而且一旦需要修改都可以很快解决。

3, 所有的文件只由 top 统领, 当流水线想要添加其他功能时, 只需要添加功能模块, 设置参数, 然后在 top 文件处修改即可, 不会影响到其他模块。有良好的扩展性。

2.5 实现上的不足之处

1.top 文件过于庞大，涉及多个变量的命名，并且还有各种连线，传参，初始化，很容易写错，非常不好调试，最好可以将其分离。

2. 对于冒险的处理有更好的方法节约时间，比如代码重排和跳转预测，但是感觉实现起来非常麻烦，所以只采用了简单粗暴的方法，效率不高。

3 遇到的一些挑战以及解决方式

3.1 关于模拟编程软件

由于我没有学过信息学竞赛，还是从数学班转过来的学生，所以对电脑操作的熟悉程度远不及其他同学。光是安装虚拟机和编程软件就耗费了很长时间，还好有一个电院的同学帮助我，并且他们的教材讲得也比较清楚，最终掌握了软件的使用方法。

3.2 关于变量命名

由于我的文件结构，决定了 top 文件是一个巨大的程序文件，需要将各种参数和寄存器在多个阶段传递，于是变量非常多很容易分不清。一开始我就感觉无从下手，几次都出错。最后参考了一个学长的命名规范，学会使用长命名，变量的名字代表变量的含义，调试了许久才可以通过。

3.3 关于一些细节改动

刚开始跟着教材走，后来才发现教材比较简单，测试时出现了很多问题，比如设计中没有考虑所有的 mips 指令，有很多指令没有对应的编码。而且我使用的 aluop 位宽比较小，这就导致了我在测试时不断修改加指令，还要修改位宽，浪费了不少时间。以后还是要想清楚完善了再动手，否则修改起来很麻烦。

3.4 关于调试与测试

这回的调试代码的过程非常有趣，完全是依靠波形图。自己设计 mips 指令，然后扔进去跑，结果由波形图显示，一旦与预期不符合，就要让各种变量在波形图中显示，查看哪里有可能出错。我曾经因为输入指令格式的问题调试了半天才好，不过后来就慢慢习惯了看图调试，感觉挺有意思的。

4 未来发展和完善的方向

4.1 跳转预测

在梁老师的课堂上，曾经提到了许多关于跳转预测的话题，但是由于时间紧迫，并没有实现，希望以后可以完成。

4.2 缓存

缓存对于中心处理器的运算速度有非常大的提升，可是我完全没有考虑过缓存，如果加入缓存甚至包括空间和时间稳定性，性能应该会更好吧。

4.3 精确中断

如何实现精确中断也是应该考虑的问题。

4.4 上板子和操作系统

我们这门课的终极目的是要玩成操作系统，并且需要在板子上实践，之前听电院的同学说过，上板子不是一件容易的事情，所以可能还要花费一些时间。此外如何在自己写的非常基础的流水上实现操作系统恐怕是更难的一件事吧。

5 学习感悟

经过这次的实验练习，使我对上学期学习的体系结构又有了新的认识。看起来简单易懂的知识，在实践中却并不容易。尤其这是一种新的语言，而且 verilog 与其说是编程语言不如当作是电路，而电路知识恰恰是我们没有接触过的。一开始有很多不适应，但是后来就慢慢习惯了。在具体实践中才发现有很多原来没有想到会出现的问题，并且最终解决了它，这是最让人高兴的。完成之后才感受到梁老师的良苦用心：纸上得来终觉浅，绝知此事要躬行！

总之，通过这次实现 MIPS 五级流水线，提高了我的解决问题的能力并且拓宽了我的编程知识，这对我未来的学习是有极大帮助的。最后感谢助教的辛勤付出和良苦用心。谢谢你了！

6 参考文献

- [1]J.L. Hennessy, D.A. Patterson, and K. Asanovi'c. Computer Architecture: A Quantitative Approach. Computer Architecture: A Quantitative Approach. Morgan Kaufmann/Elsevier, 2012.
- [2]Yamin Li. Computer Principles and Design in Verilog HDL. Tsinghua University Press, 2011.
- [3]D.A. Patterson and J.L. Hennessy. Computer Organization and Design: The Hardware/Software Interface. The Morgan Kaufmann Series in Computer Architecture and Design. Elsevier Science, 2013.
- [4] 雷思磊. 自己动手写 CPU. 电子工业出版社, 2014.
- [5] 电院电子技术实验指导用书