**开发文档**

本文档为2019小学期PPCA的MIPS的开发文档，主要完成的是五级流水结构，共分为5个版本。在分支预测的预测上仍可做改进。

**Version1.1：串行（函数式）**

建立一个Instruction类，内部存有所需的各项参数，利用IF、ID、EX、MA、WB函数，顺序执行这些函数，来执行一条指令。在该程序中，IF为取出指令（得到32位的指令），ID为解析指令（解析是何指令，所需的rs1，rs2，imm以及这些寄存器对应的值），EX进行计算，MA对内存进行读写，WB改变寄存器和pc的值。

**Version1.2：串行（各个阶段作为类）**

从Instruction类中派生出IF、ID、EX、MA、WB类，每个类有run（）和pass（）函数。每个类执行的操作同1.1。

**Version2.1：并行（通过lock标记进行stall）**

为了模拟并行，使指令倒序运行，但由于会产生data hazard和control hazard，此版本中利用lock标记进行stall。

在Version1.2上在寄存器类中加lock标记和st状态标记。并且修改了各个阶段进行的操作，修改了各阶段的pass指令，如果st==pause，则向下传递一个NOP（此处用LUI， 0， 0代替）。

* IF阶段，取出指令并判断它是什么指令，取出rs1，rs2，rd，imm，并使rd的lock增加。如果rs1或rs2的lock不为0，则修改st标记为pause；如果该指令涉及pc操作，则pc的lock增加。IF阶段还增加了judge指令，来进行特判（rd和rs相同而且已经被加锁的情况）。
* ID阶段，取出Val\_rs1，Val\_rs2。
* EX阶段和MA阶段同Version1.2。
* WB阶段，如果该指令修改寄存器，则使rd寄存器的lock减少；如果该指令修改pc，则使pc寄存器的lock减少。

在运行时，每个指令只有在st==NormallyRun时才运行，同时，只有在pc的lock为0时才执行IF指令。IF阶段在传指令之前还需要进行判断指令函数（进行特判）。

**Version2.2：并行（stall+forwarding）**

在Version2.1上更好地处理data hazard。主要修改的阶段在ID阶段，分为三种情况。

①不存在data hazard时，直接在ID阶段读取寄存器的值。

②特殊情况：存在datahazard时的特殊情况。若为I指令紧接\*需要I指令读入寄存器值的指令\*。当\*需要I指令读入寄存器值的指令\*进行到ID阶段时：第一秒需要进行一次暂停，暂停一个ID阶段，在ID阶段向后传一个NOP，暂停一个IF阶段。第二秒在ID阶段直接读取WB阶段得到的Val\_Ma（因为这一秒时MA阶段已将值传给WB）。

③一般情况：若存在data hazard且不为特殊情况，它所需要的值一定在EX阶段，只需读取EX阶段得到的Val\_Ex即可。

**Version2.3：并行（forwarding+predict）**

在Version2.3上添加了分支预测。添加了predict类，参数为pcUpdate（默认值为-1，表示不需更新），和过去的信息。Predict类有ID\_predict和EX\_Judge函数，进行预测和判断预测是否正确。同时需要修改原先的MA、WB阶段。

①对于跳转指令JAL和JALR，在ID阶段可以直接计算出应该让pc跳转到什么位置。

②对于分支指令。在EX阶段，判断是否预测正确，如果预测不正确，则修改pc值，并将原指令撤销。在ID阶段，若pcupdate == -1 （即预测正确和默认情况）则正常运行ID\_predict\*，否则（代表预测错误）初始化并传递NOP进行暂停。在IF阶段，若pc需要更新，则更新pc。

\*ID\_predict函数只对B类指令有操作。

\*分支预测成功率：

* array\_test1：13/22 59.0909%
* array\_test2：16/26 61.5385%
* basicopt1：119511/155139 77.0348%
* bulgarian：51835/71493 72.5036%
* expr：88/111 79.2793%
* gcd：87/120 72.5%
* hanoi：12838/17457 73.5407%
* lvalue2：3/6 50%
* magic：43898/67869 64.6805%
* multiarray：135/162 83.3333%
* pi：33195398/39956380 83.0791%
* qsort：150016/200045 74.9911%
* queens：55769/77116 72.3183%
* statement\_test：131/202 64.8515%
* superloop：379752/435027 87.2939%
* tak：45477/60639 74.9963%

感想与总结：

* 通过模拟CPU五级流水，学习到了机器语言、汇编语言、汇编指令集和硬件上是怎么进行操作的。在实践中大大地提高了对计算机运行和程序运行的理解。
* 锻炼了我的git能力，虽然我仍然不怎么会在命令行中直接使用git。
* 体会到了面向对象编程的重要性和对于一个项目，从简单到复杂慢慢修改、慢慢学习的过程。在学习一件事物时，动手做一些简化版本能够帮助我们了解到它的内涵和规律，从而在把它完善时能够更加轻松。
* 体验极佳。