

基于RV32I的3级流水线CPU的设计与实现

计2104班：暴旭阳 指导教师：康磊 论文类型：毕业论文

摘要：本文基于RISC-V指令集中的RV32I，在Logisim软件中设计并实现了一款3级流水线的CPU。首先，文章介绍了设计CPU要用到的关键技术，并给出了CPU的基本流水线数据通路。其次，文章详细介绍了CPU中各个功能模块的端口功能与电路逻辑，以及它们的内部电路与封装。然后，文章针对流水线中存在的冒险问题提出了解决方案，并给出了CPU的总体电路图。最后，通过样例程序在Logisim中的运行结果与RARS中结果进行比较，并将该CPU与其他CPU进行性能对比，得出了一定的结论，并提出了相关的改进方案。

关键词：CPU；RISC-V；RV32I；流水线

1 研究背景

RISC-V指令集架构是一种新兴的开源指令集架构，在近年来受到了学术界和工业界的广泛关注。RISC-V相比传统的指令集架构，其结构更为简洁，并且RISC-V开源免费，降低了CPU的研发成本和门槛。RV32I作为RISC-V指令集的基础整数指令集部分，为构建简单且高效的CPU提供了一个坚实的基础，在学术研究和教学实践中具有重要的价值。

在计算机体系结构中，流水线技术是一种被广泛应用的性能提升的方法。流水线级数的提高可以增加CPU的运行效率，但与此同时也会带来功耗和面积的增加、硬件复杂化、时序优化难度加大以及可能出现的冒险问题等一系列挑战。因此，研究和实践基于一定级数流水线的CPU设计实现，对于深入理解流水线技术的原理、优势以及面临的困难等方面具有重要的意义。

本课题既结合了当前热门的RISC-V架构，又具有相对适中的设计复杂度，适合作为本科阶段的课程设计项目和毕业论文选题。通过参与这样的项目，学生和研究人员可以学习CPU的各个组成部分，也能对流水线技术的实际应用有更加深刻的认识，为后续在计算机体系结构及相关领域的深入学习和研究打下坚实的基础。

综上所述，基于RV32I的3级流水线CPU的设计与实现这一选题具有一定的理论研究价值和实践应用意义，符合当前计算机技术发展的趋势，也符合教育科研的实际需求，值得深入地开展研究和探索工作。

2 发展现状

2.1 国外发展现状

2019年12月15日，西部数据发布了SweRV，它是一个开源的32位顺序CPU内核，

具有9级流水线以及双重任务机制的2路超标量处理器。2020年12月，来自东京工业大学的几位研究人员研发了一款针对FPGA优化的RV32I软处理器，其可以作为系统环境并适用于嵌入式系统。2021年，来自英国的研究人员设计了RISC-V处理器的轻量级开源实现，可以运行在学生使用的FPGA开发板上。2024年，在物联网领域发布了两款基于RV32I与RV32M的五级流水线处理器内核。

2.2 国内发展现状

2018年11月初，中国开放指令生态(RISC-V)联盟成立。2020年6月2日，5名国科大本科生发布了可以运行Linux操作系统的64位RISC-V处理器SoC芯片。2021年6月22日，“香山”开源高性能RISC-V处理器在首届RISC-V中国峰会上亮相。2022年，中国科学院支持众多单位，共同研发RISC-V通用工具软件、核心基础软件、跨层优化框架等。2023年12月16日举行的openEuler峰会上，华为海思推出了支持openEuler系统且基于RISC-V的A²MCU解决方案。2024年，来自杭州大学的研究员对RISC-V五级流水线的架构进行了升级，减少了依赖关系冲突对处理器的负面影响。同年，中国工程院提出了“RV++”生态正在形成。在人工智能、物联网和智联汽车等领域，中国正在发挥自己的优势推进“RV++”生态发展，以尽快形成x86、ARM和RISC-V三分天下的格局。

3 相关分析

3.1 CPU 系统架构分析

本设计使用3级流水线的架构，并在设计功能模块时遵循自顶向下的设计方式。图3.1展示了CPU的基本流水线数据通路。

在取值阶段（黄色部分），主要运行单元有程序计数器、指令存储器、寄存器文件、控制器与分支预测单元等；在执行阶段（绿色部分），主要运行单元是运算单元（ALU）、旁路转发单元等；在结果写入阶段（蓝色部分），通过数据存储器与寄存器文件实现；在每个流水线阶段之间，都存在一个级间缓冲器，用来保存上一条指令的信号；最后，通过用户输入单元和输出单元作为CPU的用户I/O接口（图3.1中未出现）。

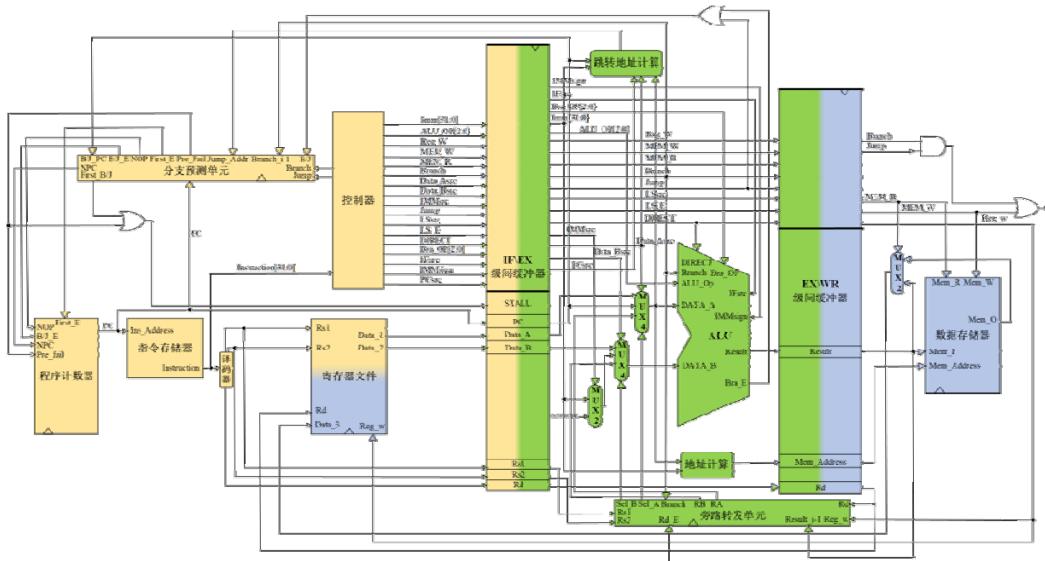


图3.1 CPU的流水线数据通路

3.2 流水线冒险分析

3.2.1 结构冒险

结构冒险是指在流水线中，当多条指令同时访问一个存储器时，由于存储器的输出端口不够用而导致的流水线执行冲突的现象。为了解决这一问题，本设计将内存具体分割为数据存储器与指令存储器。

3.2.2 数据冒险

当一条指令需要用到前面指令的执行结果，而这些指令均在流水线中重叠执行，就可能引起数据冒险。本设计引入了旁路转发技术，用于将发生数据冒险的两条指令的相关数据通过旁路及时地送至运算单元，以确保指令执行的准确性，同时消除了数据冒险需要阻塞流水线的问题，使流水线不发生停顿，提高了CPU的运行效率。

旁路转发单元在CPU运行中，每开始执行完一条指令，就会将其目的寄存器的编号以及其运算结果等信号通过寄存器临时存储起来。执行下一条指令时，便将该指令的两个源寄存器与目的寄存器进行比对，若源寄存器等于目的寄存器，则说明存在数据冒险，需要将目的寄存器存储的值转发给运算单元。

3.2.3 控制冒险

分支转移指令将产生控制冒险，因为在取指令时是否发生跳转还未确定，所以流水线CPU并不知道到底取哪条指令。本设计采用动态分支预测技术，可以有效的减少阻塞次数，提高CPU的整体性能。

分支预测单元在流水线取指阶段每接收到一条分支指令，就会将其与分支历史表中存储的指令进行比对。若当前分支指令与分支历史表中的指令的指令地址相等，则证明该指令至少跳转过一次，并预测该指令会发生跳转，运行跳转目标地址的指

令。在下一个周期，若运算单元发出了跳转信号，则说明预测成功，否则预测失败，立即清除当前流水线，并输出给程序计数器正确的指令地址，等待下一个周期指令的执行。

3.3 结果与分析

本文通过3个RISC-V样例程序在Logisim中CPU的模拟运行结果与RARS中的运行结果进行比对后，数据存储器的值二者均一致，除去寄存器组中与指令地址有关的寄存器的值不一致之外，其余寄存器的值也相同。由于Logism中指令首地址与RARS中的不一致，因此可以证明Logisim中CPU的模拟运行结果是正确的，验证了CPU设计的正确性与可行性。

最后，本文还将该CPU与无分支预测的CPU和非流水CPU进行了性能对比，运行样例程序1分别减少了16.8%和65.4%的运行时间，证明了本设计可以显著提升CPU的运行效率。

4 研究结论与对策建议

4.1 研究结论

本文基于RV32I指令集，针对3级流水线CPU开展了深入研究与实践探索，最终得出结论：本文所设计的CPU是合理的，不仅可以成功运行RISC-V程序，还为用户界面提供了灵活的输入输出功能，这为学生和研究人员在相关领域深入学习和研究打下了良好的基础。

同时，本设计引入动态分支预测功能，通过分支历史表的记录可以使CPU的性能得到一定的提高。最后，相比非流水的CPU，本设计能大幅降低程序的运行时间。

4.2 对策建议

作者由于初次接触有关流水线CPU的设计，故在许多方面还存在有待改进的地方，具体问题如下：

(1) 本设计中涉及到时序电路的部分大多采用了寄存器与计数器完成，而这些硬件的实际成本较高，可考虑使用锁存器或触发器等时序电路部件完成。

(2) 分支预测单元中的分支历史表每遇到一条新的分支指令时，其只能替换掉分支历史表中固定位置的一条指令。当程序中分支指令的数量较多时，预测成功率将大打折扣。因此可以采用动态替换的方式，将分支历史表中预测成功率最低的一条记录替换成新的分支指令，而那些预测成功率高的记录将始终存在于分支历史表。

(3) CPU中的用户界面虽然具有I/O功能，但其输入输出的功能仅局限于本文提到的3个样例程序，相关研究人员可对其进行优化改进。