

改进 OR1200 CPU 流水线的设计^①

曹凯宁, 沈兴浩, 姬梦飞, 常玉春

(吉林大学 电子科学与工程学院, 长春 130012)
通讯作者: 常玉春, E-mail: changyc@jlu.edu.cn

摘要: 流水线是制造高性能 CPU 的关键技术, 目前被广泛研究的 OR1200 是一款带有四级流水线的免费开源 CPU. 为了提高流水线的效率, 针对 OR1200 没有设计访存流水段, 流水线会暂停等待加载存储类指令这个问题, 在 LSU 操作即访存操作模块, 为 OR1200 增加了访存流水段, 设计了冒险检测和旁路单元, 因此 CPU 在访存阶段不需要暂停, 从而使 OR1200 变为真正的五级流水线 CPU; 另一方面, 当需要用加载指令加载数据的时候, 会导致加载类数据冒险问题, 为了解决此类冒险, 设计了数据有效信号 Tag, 用来控制流水线暂停, 对乘法计算、访存阶段以及其他不能在执行阶段得到结果的运算作流水线暂停判断, 以等待数据的获取. 通过实验仿真证明, Tag 信号暂停流水线一个时钟后会把数据反馈回去, 成功解决了必须暂停数据相关问题的暂停判断问题.

关键词: OR1200; 流水线; 数据冒险; 数据旁路

引用格式: 曹凯宁, 沈兴浩, 姬梦飞, 常玉春. 改进 OR1200 CPU 流水线的设计. 计算机系统应用, 2017, 26(12): 268-271. <http://www.c-s-a.org.cn/1003-3254/5938.html>

Improved Design of Pipeline CPU Based on OR1200

CAO Kai-Ning, SHEN Xing-Hao, JI Meng-Fei, CHANG Yu-Chun

(College of Electronic Science and Engineering, Jilin University, Changchun 130012, China)

Abstract: The pipeline is the key technology of manufacturing high-performance CPU. The OR1200, which has been widely studied currently, is a 4-stage pipeline CPU with a free open source. Without MEM stage which should be designed in OR1200, the pipeline will be stalled to wait for load or store instruction. In this research, we design a MEM stage for OR1200 in LSU. Hazard detection and data forwarding units have been included for efficient implementation of the pipeline. On the other hand, when a data requested by a load instruction has not yet become available, it leads to load-use hazards. To resolve this hazard problem, we design a data valid signal Tag to control stalling of pipeline. The pipeline is stalled by the Tag signal for one stage and then continues with the forwarding of data, as the simulation result shows.

Key words: OR1200; pipeline; data hazard; data forwarding

随着计算机系统和微电子技术的不断发展, 将整个电子应用系统集成在单一的芯片上, 构成功能强大的完整片上系统 (SoC), 是 IC 产业发展的必然趋势, 也是技术发展的必然要求. 而作为 SoC 运算核心和控制核心的 CPU 核, 其性能直接影响系统的整体性能, 因此设计实现高性能的 CPU 核已成为 SoC 最重要的环节.

目前, 国内虽然已经有了设计 CPU 的能力, 但大

多采用了国际主流 CPU 系列的兼容指令集与架构, 比如“龙芯”和海思公司分别取得了 MIPS 与 ARM 的授权, 以及去年海光获得了 x86 的授权. 其中的一个重要原因是这些主流架构都拥有完整或比较完整的商业生态圈, 避免了因指令集不兼容而导致 CPU 没有市场的尴尬局面. 近年来, 采用 ARM 授权方式的国内 CPU 厂商可以借助围绕 ARM 架构的生态环境获得较好的发

^① 基金项目: 吉林省省级经济结构战略调整引导资金专项项目 (2015Y041); 吉林省重点科技攻关项目 (20160204042GX)

收稿时间: 2016-12-06; 修改时间: 2016-12-19; 采用时间: 2017-01-16

展, 这该模式是以牺牲了巨大的商业利益和产品安全性为代价. 因此, 这种方式很大程度上限制了我国自主 CPU 的设计与发展. 通过对 CPU 设计发展进行充分的横向 (各研究机构、厂商) 及纵向 (时间上) 研究, 我们从头设计了拥有全新的指令集架构、完善的编译器/软件开发环境的高性能、通用性 JearCore 开源 CPU. Jear Core 基于在学术界广泛流行的 OpenRISC 架构, 利用其先进的、开放性的底层架构体系, 所具备的完整的开发、编译工具链, 从底层做起, 重新改写流水线, 彻底优化现有代码, 增添辅助模块, 不断完善和发展这种应用前景巨大的 32 位、高性能 RISC(精简指令集计算机)^[1,2] CPU 设计, 并无偿提供给企业进行产业化.

本文凭借 JearCore CPU 设计理念, 以 Xilinx 公司的 FPGA 为目标器件, 用 Verilog 语言描述了 32 位的流水线 CPU. CPU 流水线技术是一种将指令分解为多步, 并让不同指令的各步操作重叠, 从而实现几条指令并行处理, 以加速程序运行过程的技术. 每个子过程对应一个时钟, 称为流水级或者流水段^[3]. 流水线的引入提高了各个流水段的利用率, 同时也产生了数据冒险的情形. 此次我们研究的 OR1200 是由 OpenCores 组织^[4]开发的一个 32 位标量 RISC. 虽然 OR1200 手册^[5]声称 OR1200 是五级流水线, 但是由于它并没有设计访存流水段, 因而被认为是四级流水线的 CPU. 针对以上问题, 我们为 OR1200 设计的访存流水段, 并且针对增加流水段而引起的数据冒险进行了分析和解决.

1 OR1200 的流水线以及改进

五级流水线^[6]的 CPU 一般分为五步进行, 分别是“取指”、“译码”、“执行”、“访存”、“写回”. OR1200 将“执行”阶段的计算得到的地址送给总线, 而控制访存阶段的控制信号依然用的是执行阶段的控制信号, 因而造成了在访存阶段必暂停的问题^[7,8], 如图 1 所示. 图中 ex_insn 为执行阶段的指令, 其中“d4031000”为存储指令 l.sw 0x0(r3), r2 的机器码, “84830000”为加载指令 l.lwz r4, 0x0(r3) 的机器码. OR1200 执行访存指令的时候通过 lsu_stall 判断暂停, 当总线 dcpu_ack_i 为 1 的时候访存结束继续执行下一条指令.



图 1 OR1200 访存操作仿真图

通过对 OR1200 改进的时候增加了访存阶段的控制信号, 使得 CPU 在访存阶段没有出现 cache miss 情形下不需要暂停^[9], 如图 2 所示. 图中虚线表示流水线之间的寄存器. 我们将 OR1200 访存操作从执行流水段中分离出来, 来解决 OR1200 遇到访存操作暂停的问题. 这样一方面提高了系统主频; 另一方面提高了硬件模块的利用率和吞吐量.

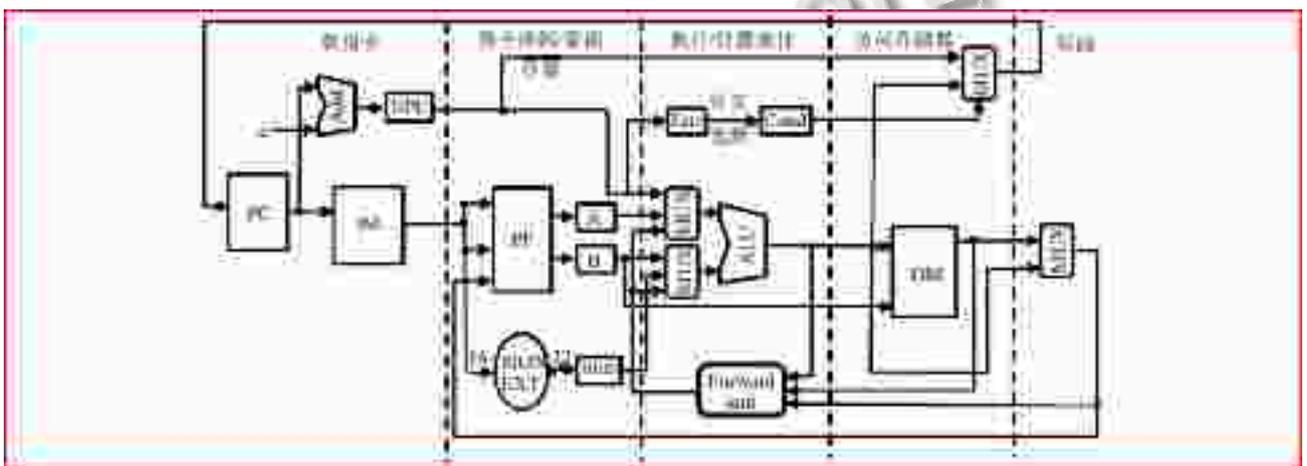


图 2 改进后 CPU 流水线结构图

向改进后的 CPU 中输入图 1 中相同的指令, 得到的仿真结果如图 3 所示. 可知在执行存储指令 l.sw

0x0(r3), r2 和加载指令 l.lwz r4, 0x0(r3) 流水线均未暂停. lsudata 为加载指令的加载结果.



图3 改进后 CPU 访存操作的仿真图

2 数据相关以及旁路技术

引入流水线提高 CPU 效率的同时,我们会遇到流水线冒险的问题,称之为流水线相关。流水线相关分为:结构相关、数据相关和控制相关。由于 OR1200 采用的是内部哈佛结构即分开的指令数据缓存,所以不存在结构相关。OR1200 在处理分支跳转的时候,通过暂停等待跳转信息的到来,所以 OR1200 已经解决了控制相关的问题。在增加访存流水段以后,OR1200 对于数据相关的处理电路已经不再适用了,因此我们需要重新考虑。

数据相关主要包括三种: RAW(read after write)、WAW(write after read) 和 WAR(write after write)。对于五级流水线 CPU 而言,只有在流水线回写阶段才会写寄存器,因此不存在 WAW 相关。又因为只能在流水线译码阶段读寄存器、回写阶段写寄存器,不存在 WAR 相关,所以五级流水线 CPU 一般只存在 RAW 相关。RAW 相关有以下三种情况:

(1) 相邻指令间存在数据相关,举例如下:

```
l.ori r1, r0, 0x1
l.ori r2, r1, 0x0
```

(2) 相隔 1 条指令的指令间存在数据相关,举例如下:

```
l.ori r1, r0, 0x1
l.ori r3, r0, 0x0
l.ori r2, r1, 0x0
```

(3) 相隔 2 条指令的指令间存在数据相关,举例如下:

```
l.ori r1, r0, 0x1
l.ori r3, r0, 0x0
l.ori r4, r0, 0x0
l.ori r2, r1, 0x0
```

为了解决数据相关^[9,10]的问题,我们引入了旁路技术^[11]来解决这个问题,如图 4 所示。我们将旁路技术分为两个部分设计,一是操作数选通器中的设计,二是在写入寄存器堆之前使寄存器堆完成对数据的前推。

图中第一条指令的结果需要在 CC5 这个时钟写回到寄存器堆。而接下来的 4 条指令都需求这个结果。因此如果此时直接选取寄存器堆里的指令结果一定是错误的,所以在设计的操作数选通器使用旁路技术将

结果前推。这样第二条指令可以直接从 ALU 计算得出的结果中得到所需的操作数,而不需要等待 CPU 将结果写回寄存器。而第三条指令则可以从 MEM 阶段的前推数据中获取。第四条指令可以从 WB 阶段的前推数据中获取。将指令计算的结果通过数据前推给操作数选通器,这样便解决了 RAW 的问题。

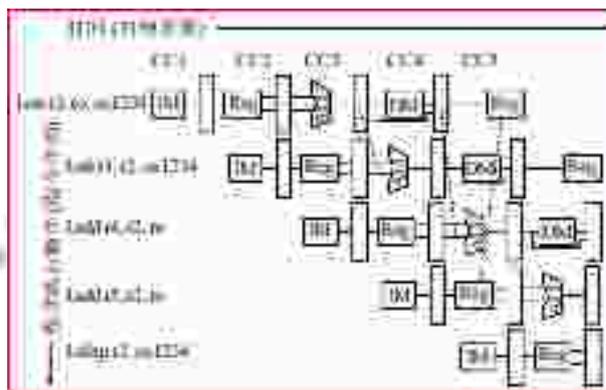


图4 旁路技术对数据相关问题的解决

3 数据相关暂停机制设计

在讨论数据相关问题的時候,还得考虑一个问题,就是如果第二条指令需要第一条指令的结果,但是第一条指令是一条加载指令,并不能在执行阶段给出结果,必须在访存阶段才会得到结果。而此时运行在译码阶段的指令并不能通过将执行阶段的结果前推来得到对应的操作数。如图 5 所示。l.lwz r4, 0x0(r3) 的结果在 CC4 才能得出,而在 CC3 就需要这个结果,所以此时需要暂停流水线^[12]。

因此设计了数据有效信号 tag 来解决这个问题。如图 6 所示。在 CC3 这个时钟,指令 l.sfeq r2, r4 的 a 操作数地址和指令 l.lwz r4, 0x0(r3) 写寄存器地址相同。指令 l.sfeq 需求指令 l.lwz 的结果,而此时数据是无效的,所以数据 Tag 为 0,这时就会发出流水线暂停的请求,暂停流水线。在 CC4 这个时钟的时候,指令 l.lwz 获取到了结果,这时 Tag 变为 1,同时将结果送给指令 l.sfeq,流水线解除暂停。

为此设计了汇编指令对此进行仿真,汇编指令如下:

```
l.movhi r2, 0x8483
l.ori r2, r2, 0x8281
l.ori r3, r0, 0x50
l.sw 0x0(r3), r2
l.lwz r4, 0x0(r3)
l.sfeq r4, r2
```

将汇编指令通过 OR1200 模拟器^[13]进行仿真,得

到的结果如图 7 所示, l.sw 将数据“r2=84838281”存储在地址为 0x50 的内存地址上, 然后通过加载指令 l.lwz 加载相同内存地址的数据保存到 r4 中, 之后由 l.sfeq 比较 r2 和 r4 的结果是否相等, 比较的结果为相等, 因此 flag 信号变成 1.

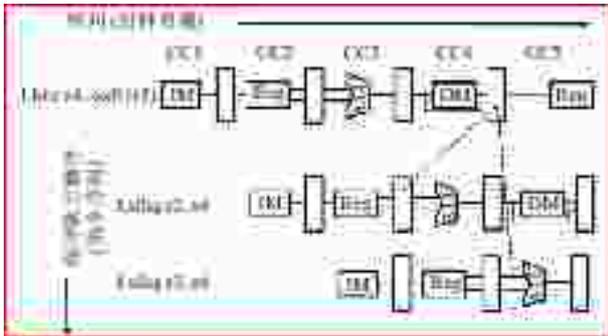


图 5 使用加载指令存在的数据相关

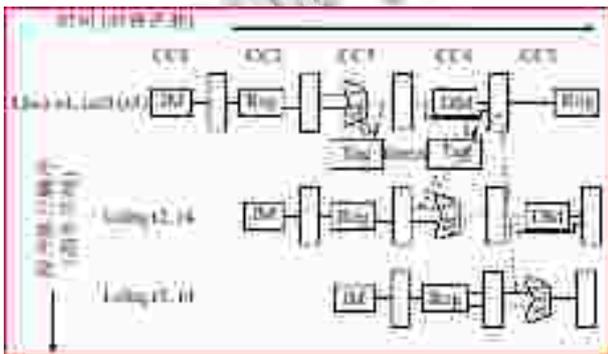


图 6 数据有效信号 Tag 在暂停判断中的作用



图 7 OR1200 模拟器仿真结果

将以上指令通过 modelsim 仿真得到图 8. 由图可知, 指令 l.sfeq r4, r2 在解码阶段操作数 a 的地址为 0x4 与上一条指令 l.lwz r4, 0x0(3) 执行阶段的地址相同, 而此时 ex_wdatatag 为 0, 显示此时指令 l.lwz r4, 0x0(3) 并未在执行阶段得到结果. 所以这时候 if_freeze 和 id_freeze 变成 1, 流水线的取指和解码工作暂停一个时钟. 当指令 l.lwz r4, 0x0(3) 进行到访存阶段, 此时 mem_wdatatag 信号已经为 1, 表示指令得到了结果. reg1 将此时 mem_wdata 的结果作为输入传递给操作数寄存器 a. 再经过一个时钟比较指令 l.sfeq r4, r2 到达访存阶段的时候 flag 被置为 1.



图 8 数据相关暂停机制的 modelsim 仿真

4 结语

流水线是现代 CPU 中普遍采用的一种技术, 它只需要增加很少的硬件就能使 CPU 的速度提高很多. 本文通过修改 OR1200 的流水线解决了 OR1200 通过暂停等待访存结束的问题, 针对流水线数据相关的问题重新设计了旁路判断电路. 遇到加载类指令数据相关的时候, 我们通过设计的写寄存器数据有效信号 Tag 来进行写数据有效性判断, Tag 变为 1 的同时将结果送给指令 l.sfeq, 流水线暂停会被解除. 最终通过汇编指令对数据相关暂停机制仿真, 仿真结果符合设计要求.

参考文献

- 1 Bhavsar DK. An algorithm for row-column self-repair of RAMs and its implementation in the Alpha 21264. Proc. of the 1999 IEEE International Test Conference. Washington, DC, USA. 1999. 311-318.
- 2 Smith JE, Weiss S. Power PC 601 and Alpha 21064: A tale of two RISCs. Computer, 1994, 27(6): 46-58. [doi: 10.1109/2.294853]
- 3 Hennessy JL, Patterson DA. 计算机系统结构: 量化研究方法. 郑纬民译. 北京: 电子工业出版社, 2004.
- 4 OpenCores.org. Home page. <http://www.opencores.org>.
- 5 OR1200 OpenRISC Processor. http://opencores.org/or1k/OR1200_OpenRISC_Processor.
- 6 王绍坤. 基于 FPGA 的五级流水线 CPU. 计算机系统应用, 2015, 24(3): 18-23.
- 7 金鹏, 杨刚, 胡耀. 基于 OR1200 的批处理协处理器设计. 微电子学与计算机, 2014, (10): 39-42.
- 8 姚永斌. 超标量处理器设计. 北京: 清华大学出版社, 2014.
- 9 Patterson DA, Hennessy JL. Computer organization and design: The hardware/software interface. Amsterdam, Boston, USA: Morgan Kaufmann, 2005.
- 10 Gautham P, Parthasarathy R, Balasubramanian K. Low-power pipelined MIPS processor design. Proc. of the 12th International Symposium on Integrated Circuits. Suntec, Singapore. 2009. 462-465.
- 11 张德学, 张小军, 郭华. FPGA 现代数字系统设计及应用. 北京: 清华大学出版社, 2015.
- 12 甄建勇. 深入理解 OpenRISC 体系结构. 北京: 机械工业出版社, 2015.
- 13 张望亭. 基于 OpenRISC1200 32 位 CPU 的 miniSoC 系统设计和软硬件验证[硕士学位论文]. 成都: 电子科技大学, 2013.