

文章编号: 1671-5896(2017)06-0612-05

五级流水线 RISC-V 处理器软硬件协同仿真验证

李东泽^a, 曹凯宁^a, 曲明^{a, b}, 王富昕^b

(吉林大学 a. 电子科学与工程学院; b. 计算机科学与技术学院, 长春 130012)

摘要: 针对国内 RISC-V (Reduced Instruction Set Computer-Five) 处理器领域的空白以及对处理器性能的优化问题, 将开源 3 级流水线 RISC-V 处理器 VScale 扩展为 5 级流水线处理器。在对比 3 级流水线和 5 级流水线的差异的基础上, 为 5 级流水线设计了冒险检测以及旁路单元, 解决了 5 级流水线的冒险相关问题, 并为该处理器编写外设 (LCD1602、UART) 控制器, 最终在 FPGA (Field-Programmable Gate Array) 开发板上实现了软硬件协同仿真。仿真结果表明, 扩展后的处理器运行正常, 且速度比扩展前的处理器快约 30%。

关键词: RISC-V 处理器; 流水线; 数据冒险; SoC 技术

中图分类号: TP368.1

文献标识码: A

DOI:10.19292/j.cnki.jdxxp.2017.06.006

Hardware and Software Co-Simulation of RISC-V Processor with Five-Stage Pipeline

LI Dongze^a, CAO Kaining^a, QU Ming^{a, b}, WANG Fuxin^b

(a. College of Electronic Science and Engineering; b. College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract: To fill the blank of domestic RISC-V (Reduced Instruction Set Computer-Five) processor field and optimize processor performance, the open source three-stage pipeline RISC-V processor VScale is extended to the five-stage pipeline processor. Firstly, the difference between the three-stage and the five-stage pipeline are compared. To solve the five-level pipeline data-related issues, the five-stage adventure detection and bypass unit for the five-stage pipeline are designed. Peripheral (LCD1602, UART) controllers for the processor are also implemented. The co-simulation of hardware and software are implemented on an FPGA (Field-Programmable Gate Array) development board. Simulation results show that the extended processor is running normally, and running faster than the original processor about 30%.

Key words: RISC-V processor; pipeline; data adventure; system-on-a-chip (SoC) technology

0 引言

随设计与制造技术的发展, 集成电路设计从晶体管的集成发展到逻辑门的集成, 目前已经发展到 IP (Intellectual Property) 的集成, 即 SoC (System-on-a-Chip) 设计技术。现在集成电路技术已将数量更惊人的计算机系统集成到一块芯片上, 在芯片上处理器与缓存几乎随处可见。SoC 有效地降低了电子信息系统的产品开发成本, 缩短了开发周期, 提高了产品的竞争力, 是目前工业界采用的最主要的产品开发方式。

RISC-V (Reduced Instruction Set Computer-Five)^[1] 开源指令集与传统指令集相比, 具有比 x86 和 ARM^[2] (Advanced RISC Machines) 更好的内部结构, 且开源可扩展和快速移植^[3], 使处理器具有了前所未有的创新性与灵活性。近年来, RISC-V 的生态系统也正日趋完善, Kapre 等^[4] 在 FPGA (Field-

收稿日期: 2017-09-19

基金项目: 吉林省发改委省级经济结构战略调整引导资金专项基金资助项目 (2015Y041)

作者简介: 李东泽 (1995—), 男, 长春人, 吉林大学硕士研究生, 主要从事 CPU、数字信号处理等集成电路设计研究, (Tel) 86-18844561092 (E-mail) 1418547095@qq.com; 王富昕 (1975—), 女, 长春人, 吉林大学高级工程师, 主要从事计算机接口与组成研究, (Tel) 86-13039223630 (E-mail) wfx@jlu.edu.cn。

Programmable Gate Array) 中放入了400个定制的 RISC-V 处理器并用片上网络将其连接在一起,实现并行计算,为以后的云计算引擎加速奠定了基础;2017年初,美高森美公司宣布成为首家针对 RISC-V 设计提供全面软件工具链和知识产权内核的可编程逻辑器件供应商,并与 SiFive 合作开发的 RV32IM RISC-V 新内核使客户能利用开放式指令集架构进行设计。2017年中旬,RISC-V 展示了 FPGA 解决方案、安全 IP、调试基础架构等方面的广泛生态系统,部分 ARM 客户已经开始使用 RISC-V 设计定制处理器。这表明 RISC-V 在国外备受关注,但在国内对其研究却非常少。

目前,RISC-V 官网列举了一些 RISC-V 开源软核,如6级流水线处理器 rocket-chip,BOOM,3级流水线处理器 VScale 和没有流水线的处理器 PicoRV32。在摩尔定律不再适用的今天,定制处理器已成为满足成本优化需求的主要途径,而相比于其他流水线,5级流水线处理器也是效率最高的。笔者希望设计一种成本更低,效率更高的处理器,并无偿提供给各大企业以实现产业化。CPU 流水线技术是目前 CPU 加速运行的核心,它将一条指令分解为多步,让其并行执行,增加了对时钟的利用率,但也会带来数据冒险等问题。笔者以3级流水线 VScale 为原型,对比3级流水线与5级流水线的冒险差异,将其扩展为5级流水线,解决随之带来的数据相关问题。同时利用 GNU(GNU's Not Unix) 工具链^[5],最终通过编译器和调试器对搭建的系统在 Xilinx 公司的 VIRTEX-5 系列开发板上资源进行了进一步验证。

1 VScale 流水线以及改进

5级流水线一般分为取值(IF: Instruction Fetch)、解码(ID: Instruction Decode)、执行(EX: Execute)、访存(MEM: Memory)和写回(WB: Write Back)^[6],其划分可使每个阶段所需时钟大致相等。若一个阶段时钟周期过长,则整体的时钟周期不得不向该阶段时钟周期看齐,拖慢处理器的速度。原3级流水线处理器 VScale(见图1a)解码执行周期(DX: Decode and Execute)和访存周期过长,将 VScale 扩展,并解决随之带来的数据相关问题,最后形成一个新的5级流水线处理器(见图1b)。从而可极大地提高系统主频和硬件的利用率^[7-8]。

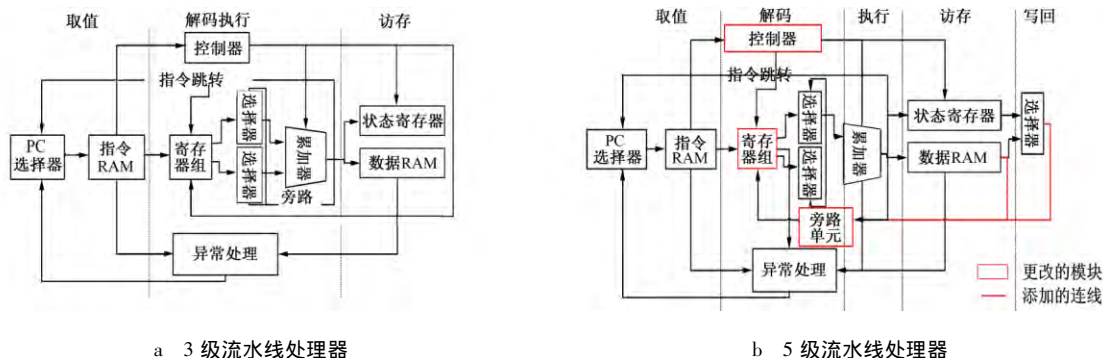


图1 原 CPU 结构与扩展后的 CPU 结构图

Fig.1 The original CPU structure and the expansion of the CPU structure

1.1 数据相关及其旁路

在5级流水线中,数据冒险与3级流水线中的数据冒险情况有所区别。在3级流水线中,只有相邻的指令之间存在冒险,而该冒险通过一条旁路即可消除(见图2)。

而5级流水线更为复杂(见图3),需3条旁路才可妥善解决,这3条旁路分别处理如下的数据相关情况。

相邻指令间存在的数据相关(图3标号(1)):

```
li a5 ,1;
addi a5 a5 2;
```

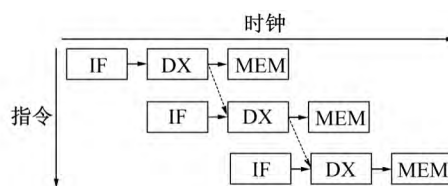


图2 3级流水线旁路

Fig.2 Three-stage pipeline bypass

隔一条指令的指令间存在的数据相关(图3标号(2)):

```
li a5 ,1;
nop;
addi a5 ,a5 ,2;
```

隔两条指令的指令间存在的数据相关(图3标号(3)):

```
li a5 ,1;
nop
nop
addi a5 ,a5 ,2;
```

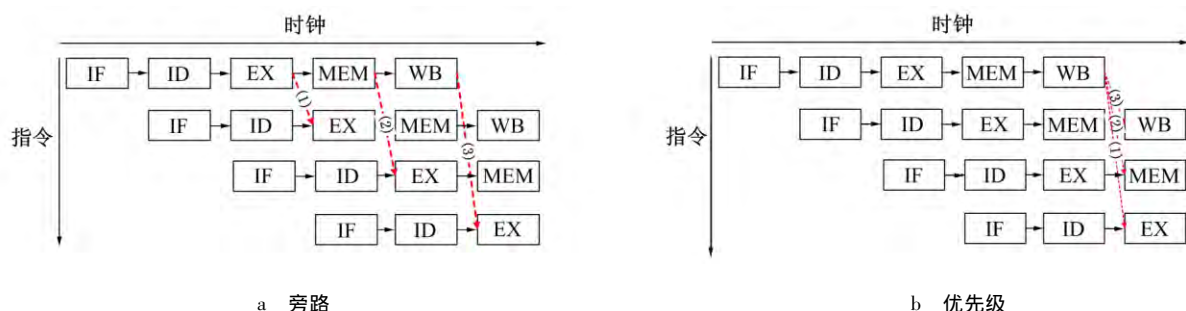


图3 5级流水线旁路及其优先级

Fig.3 Five-stage pipeline bypass and its priority

若在同一时刻,以上3种数据相关问题同时发生,则同时产生了3个旁路(见图3),这说明旁路要有优先级的区分。如图3b所示,应确保旁路1优先级最高,旁路2优先级次之,旁路3优先级最低,指令举例:

```
li a5 ,1;
li a5 ,2;
li a5 ,3;
addi a5 ,a5 ,2;
```

为解决隔两条指令存在的数据相关问题,需改写寄存器组,并建立额外的数据通路模块,并保证优先级次序,部分代码如下^[9]:

```
Assign rs1_data_bypassed = bypass_rs1? bypass_data_MEM: (bypass_rs1_wb? bypass_data_WB: rs1_next_data);
Assign rs2_data_bypassed = bypass_rs2? bypass_data_MEM: (bypass_rs2_wb? bypass_data_WB: rs2_next_data);
```

针对改写完的处理器,用如下两个汇编程序仿真:

```
1) ori a0 ,zero ,8;      #a0 = 0x1000
   ori a1 ,a0 ,4;        #a1 = 0x1100
   ori a2 ,a0 ,2;        #a2 = 0x1010
   ori a3 ,a0 ,1;        #a3 = 0x1001
```

该程序用于仿真3条旁路的工作情况,指令的注释给出了预期执行结果。仿真结果如图4所示,可以看到,a1、a2、a3寄存器内存储的值符合预期值。

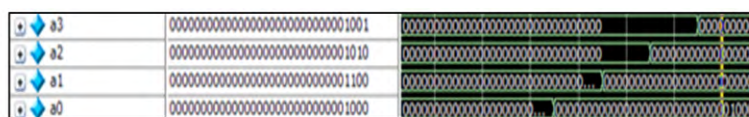


图4 旁路仿真结果

Fig.4 Bypass simulation results

```
2) li a5 ,1;      a5 = 0x0001
   li a5 ,2;      #a5 = 0x0010
```

```
li a5 3;          #a5 = 0x0011
addi a5 a5 2;     #a5 = 0x0101
```

该程序用于仿真 3 条指令相关同时发生的情况,指令的注释给出了预期执行结果,仿真结果给出了连续 4 个时刻 a5 寄存器内部的数值(见图 5)。可以看到,a5 寄存器的变化符合预期结果。

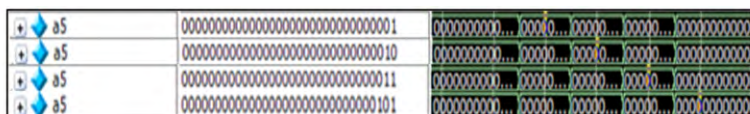


图 5 旁路优先次序仿真结果

Fig. 5 Bypass priority simulation results

1.2 Load 相关

在流水线中还存在一类相关问题,即在指令 load 加载数据马上用到该数据的情况。由于 load 指令只在访存阶段才能访问内存得到数据,所以需采取额外的措施解决。3 级流水线在处理 load 相关时只需暂停一个时钟周期,而 5 级流水线更为复杂,这里分为两种情况进行讨论。

1) 相邻指令 load 相关。若发生相邻指令 load 相关,处理器也需要暂停一个周期,以等待 load 指令运行至访存阶段得到数据。这时,数据在 MEM 阶段,而下一条指令在 ID 处等待。于是利用图 3 设计的旁路,将该数据送至执行阶段参与下一条指令的运行。

如程序

```
load a5 0(zero);
mv a0 a5;
```

执行过程如图 6 所示。

2) 相隔一条、两条指令 load 相关。当相隔一条指令的 load 相关发生时,在访存阶段得到数据,所以只需跟相邻两条指令的相关一样处理。同理,相隔两条指令的 load 相关类似,无需特别处理。

以上内容通过修改处理器 Vscale 的流水线,将 3 级流水线扩展为 5 级流水线并解决相关问题。下面将为该处理器增加外设并用软件控制,以验证处理器的稳定性。

2 硬件平台

处理器设计的硬件系统如图 7 所示,验证平台在 ISE14.7 中搭建,在 Xilinx Virtex5 系列 FPGA 开发板中进行验证。处理器外接数据 RAM 和指令 RAM,以避免结构冒险,由于该 FPGA 容量较大,这两个 RAM 都用 FPGA 生成,以提高运行速度。连接 4 个外设控制器,分别为 LCD1602 控制器、串口输出控制器、字符输入装置和控制按钮。其中,串口作为该处理器的输出终端,以 LCD1602 作为该处理器的显示装置,以字符输入装置作为简单的键盘进行输入,控制按钮用于控制设备的工作状态。

3 软件模块

软件部分包括 lcd1602 控制程序、存储显示内容的栈程序以及串口终端输出程序。主要完成以下任务: 1) 初始化处理器寄存器组; 2) 跳转至主函数运行; 3) 设置串口波特率及相关配置; 4) 监听用户输

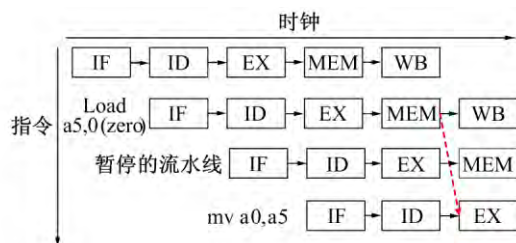


图 6 load 相关执行示意

Fig. 6 Load related to the implementation of the instructions

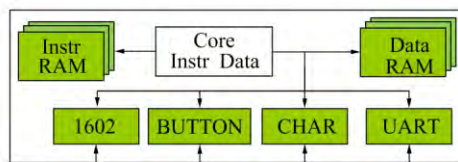


图 7 硬件结构级外部控制器接口

Fig. 7 Hardware structure level external controller interface

入并显示在 LCD1602 屏幕上^[10]; 5) 串口终端打印信息。

4 协同仿真结果

进一步实现系统在 FPGA 开发板上的仿真验证: 程序烧录完成后, 再次上电。首先在 1602 输入“RISC-V is good”(见图 8), 同时观察串口助手内容, 并打印操作过程以及最后的输出内容, 完成基于 RISC-V 处理器的软硬件协同仿真与设计。



图 8 LCD1602 显示结果及串口输出

Fig. 8 LCD1602 display results and serial output

5 结 语

笔者扩展 RISC-V3 级流水线处理器到 5 级流水线, 解决其数据相关问题。通过搭建基于 IP 核的硬件系统基本模型以及编写配套软件进行交叉编译和仿真, 最终在 Xilinx 公司的 VIRTEX5 开发板上实现了硬件系统与软件平台的交互, 验证了该处理器的稳定性, 最终结果符合设计要求。

参考文献:

- [1] WATERMAN A, LEE Y, PATTERSON D A, et al. The RISC-V Instruction Set Manual, Volume I: Base User-Level ISA [J]. Eecs Department, 2011, 7(9): 475.
- [2] FURBER S. ARM System-on-Chip Architecture [M]. Boston, Massachusetts, USA: Addison-Wesley Longman Publishing Co Inc, 2000.
- [3] 雷思磊. RISC-V 架构的开源处理器及 SoC 研究综述 [J]. 单片机与嵌入式系统应用, 2017, 17(2): 56-60.
LEI Silei. Research on RISC-V Architecture Open Source Processor and SoC Research [J]. Microcontrollers & Embedded Systems, 2017, 17(2): 56-60.
- [4] KAPRE N, GRAY J. Hoplite: A Deflection-Routed Directional Torus NoC for FPGAs [J]. Acm Transactions on Reconfigurable Technology & Systems, 2017, 10(2): 1-24.
- [5] 潘建, 董金祥. 基于 GNU 工具链的嵌入式操作系统开发 [J]. 计算机工程与应用, 2004, 40(26): 112-116.
PAN Jian, DONG Jinxiang. The Development of Embedded Operating System Based on GNU Toolkit [J]. Computer Engineering and Applications, 2004, 40(26): 112-116.
- [6] PATTERSON D A, HENNESSY J L. Computer Organization and Design, the Hardware/Software Interface [M]. San Francisco, CA, USA: Morgan Kaufmann, 2005.
- [7] GAUTHAM P, PARTHASARATHY R, BALASUBRAMANIAN K. Low-Power Pipelined Mips Processor Design [C] // International Symposium on Integrated Circuits. Piscataway, NJ, USA: IEEE, 2009: 462-465.
- [8] KATZ R. Reduced Instruction Set Computers [C] // Solid-State Circuits Conference. [S.l.]: IEEE, 1986: 218-219.
- [9] 雷思磊. 自己动手写 CPU [M]. 北京: 电子工业出版社, 2014.
LEI Silei. Write Your Own CPU [M]. Beijing: Electronic Industry Press, 2014.
- [10] 陈慧南. 数据结构——C 语言描述 [M]. 3 版. 西安: 西安电子科技大学出版社, 2015.
CHEN Huinan. Data Structure-C Language Description [M]. 3rd ed. Xi'an: Xidian University of Electronic Science and Technology Press, 2015.

(责任编辑: 张洁)