三级流水线 RISC-V 处理器设计与验证

折如义[1],李炳辉[2],姜佩贺[2]
(1.河套学院 理学院,内蒙古 巴彦淖尔 015000;2.烟台大学 光电信息科学技术学院,山东 烟台 264005)

摘 要:RISC-V 作为一种开源精简指令集架构,自发布以来便得到了大量关注。设计了一种三级流水线的 RISC-V 处理器。其中,采用静态预测 BTFN 技术处理流水线执行中的分支情况,采用前向旁路传播技术解决数据冒险问题,同时,采用资源共享的办法,复用寄存器堆、加法器、选择器等模块,使设计面积得到一定的优化。在 VCS 和 Verdi 等 EDA 工具中,使用RV32I 整数运算指令集对处理器进行了仿真测试,结果表明,所设计的处理器功能正确,达到预定目标。

关键词:RISC-V 指令集;流水线;处理器
中图分类号:TN4 **文献标识码**:A DOI:10.16157/j.issn.0258-7998.200028

中文引用格式:折如义,李炳辉,姜佩贺. 三级流水线 RISC-V 处理器设计与验证[J].电子技术应用,2020,46(5):44-49.
英文引用格式:She Ruyi,Li Binghui,Jiang Peihe. Design and verification of RISC-V processor with three-stage pipeline[J]. Application of Electronic Technique,2020,46(5):44-49.

Design and verification of RISC-V processor with three-stage pipeline

She Ruyi[1],Li Binghui[2],Jiang Peihe[2]
(1.Department of Science,Hetao College,Bayannur 015000,China;
2.School of Opto-Electronic Information Science and Technology,Yantai University,Yantai 264005,China)

Abstract: As an open source reduced instruction set architecture, RISC-V gained a lot of attention since its release. A three-stage pipelines RISC-V processor is designed. Back taken forward not taken(BTFN) is used to handle branch situation in pipeline execution. Bypass and forward technology is used to solve data risk. At the same time, the method of resource sharing is adopted, and the modules such as general register heap, adder and selector are reused to optimize the design area. In the EDA tools, simulation is carried out using the RV32I integer arithmetic instruction set. The result shows that the designed processor works correctly and achieves the predetermined goal.

Key words: RISC-V instruction set;pipeline;processor

0 引言

集成电路产业是国家战略性产业,是推动信息产业发展的源泉和动力,而我国集成电路产业发展严重滞后[1]。在各行各业需求量与日俱增的处理器领域,ARM 处理器在嵌入式领域占主导地,Intel x86 架构处理器在桌面和服务器领域占据着垄断地位[2]。RISC-V 指令集是加州大学伯克利分校于 2014 年设计并发布的一款开源指令集架构[3],具有免费开放、短小精悍、性能优越三大特征,可以被任何学术机构或商业组织自由使用,能够满足从微控制器到超级计算机等各种应用的需求[4]。RISC-V 的出现可能改变由 ARM 和 Intel x86 主导处理器架构的竞争格局[5]。

流水线是处理器设计最重要的环节之一,严重影响着处理器的运算速度和运算模块的张度。早期的经典流水线是五级流水[6],分别为取指、译码、执行、访存和写回,流水线的长短不仅仅影响吞吐率而且影响面积开销。现代的高性能处理器相比最早期的处理器往往具有更深级别的流水线。流水线的级数越多,流水线被切得越细,每一级流水线内容纳的硬件逻辑越小,进而吞吐率性能更佳,这是流水线深度加深的正面意义[7]。但由于级数加深,会消耗更多的寄存器,带来更多的面积开销,同时对于分支预测失败只能采取冲刷流水线的方法解决,浪费了处理器性能。因此,流水线的深度要根据不同的应用场景选择,本设计采用三级流水线结构,以在兼顾处理器功能的前提下实现低功耗的设计目标。

基于以上背景,本研究在分析了 RISC-V 指令系统的基础上,使用 Verilog 语言分别设计了 RISC-V 处理器的取值单元、译码单元和执行单元,最终实现了一款基于 RISC-V 指令集的 32 位三级流水处理器,并使用 RV32I整数运算指令集对处理器进行了仿真验证,达到预定目标。

1 RISC-V 指令系统

RISC-V 指令集架构是一个基础的整数指令集架构,

在设计之初就秉承“精简”的设计风格，目前的“RISC-V 架构文档”分为“指令集文档”和“特权架构文档”。“指令集文档”的篇幅为 100 多页，而“特权架构文档”的篇幅也仅为 100 页左右，与动辄上千页的 X86 架构或者 ARM 架构的架构文档相比，可以说是极其短小精悍。RISC-V 架构还有着其他架构不具备的模块化、极简、可扩展的技术特点，从而可以通过一套统一的架构满足各种不同的应用需求。

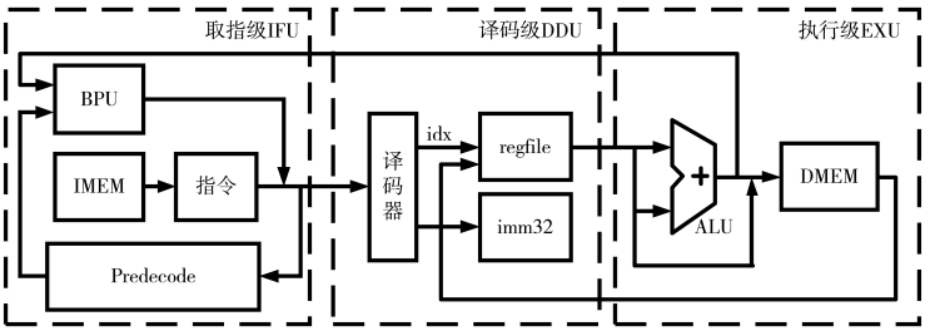

图 2 整体设计框图

1.1 指令格式

按照指令的格式，RISC-V 指令可以分为四种：R 型指令为算数操作指令，I 型指令为逻辑操作指令和加载指令，S 型指令为存储类指令，U 型指令为加载高位立即数指令，如图 1 所示，图中的 opcode 为指令操作码，funct7 为 7 位的功能码，funct3 为 3 位的功能码，rs1 和 rs2 表示两个 5 bit 源寄存器，rd 表示 5 bit 的目的寄存器，imm 表示不同长度的立即数。

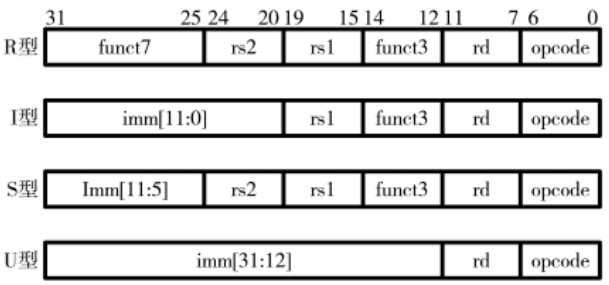

图 1 RISC-V 指令格式

1.2 基础指令

RISC-V 指令集基本指令只有 47 条，有 31 个通用寄存器 x1~x31，用于保存整数值。寄存器 x0 是硬件连线的常数 0，没有硬件连线的子程序返回地址连接寄存器，但是在调用中，标准软件调用约定使用寄存器 x1 来保存返回地址。

RISC-V 指令集手册中的基本指令，包括逻辑操作指令、移位操作指令、简单算术操作指令、加载和存储指令、存储器模型 FENCE 指令和状态寄存器指令。其中寄存器模型 FENCE 指令用于顺序化其他 RISC-V 线程、外部设备和协处理器 I/O 设备访问，状态寄存器用于访问那些可能需要特权访问的系统功能。

2 RISC-V 处理器设计

本设计是具有三级流水线的处理器，和处理器经典的五级流水相比，是将执行、访存和写回在一个流水周期内完成，最终分为三级，分别称为取值级(Instruction Fetch Unit，IFU)、译码级(Decode & Dispatch Unit，DDU)和执行级(Execute Unit，EXU)，整体框图如图 2 所示。IFU 主要包括指令计数器模块、预译码模块，主要作用是在取指后进行预译码并将指令地址盒指令传输到译码级；DDU 主要包含译码模块、寄存器 regfile 模块和相应的数据通路。主要作用是将指令译码，使处理器明晰本条指令的运算类型、源寄存器、源操作数、目标寄存器等信息。EXU 主要包括 ALU、数据存储器访问模块和写回模块，主要功能是根据译码结果执行相应的运算或操作，并将运算或操作结果存储到存储器或写回到寄存器堆。

2.1 取值单元设计

取指是流水线执行的第一步，在 RISC-V 架构中，除常规的取指单元外，还需加入预译码模块，这样可以更快地译码出指令类型和操作数，简化硬件设计，对特殊指令(如条件分支)起到预测功能，避免流水线性能浪费。

分支预测可以在方向(跳转\顺序)和地址两方面进行预测。在本设计中，方向预测选用的是静态 BTFN(Back Taken，Forward Not Taken)预测，即对于向后的跳转预测为跳，向前的跳转预测不跳。地址预测选用返回地址堆栈技术(Return Address Stack，RAS)，即使用有限的硬件堆栈存储函数调用返回的地址。

设计的取指单元顶层框图如图 3 所示。

2.2 译码单元设计

在译码单元的设计中，不仅需要考虑取指后的指令的译码，还需要根据预译码部件的结果做现场记录，并完成写回部件到第二级流水线寄存器的相关操作。

此外，流水线在传播过程中，不同指令可能在同一时刻对同一寄存器产生读写信号，指令先后顺序的不同可能造成读写数据发生错误，即产生数据冒险。数据冒险主要分为 3 种：先读后写(Write After Read，WAR)、先写后写(Write After Write，WAW)和先写后读(Read After Write，RAW)。对于前两种可以通过寄存器重命名来避免，但第 3 种 RAW 必须要等到先序指令执行完成，再进行后续操作，这将造成流水线停顿。本设计采用数据旁路传播技术来避免该问题。在设计中加入一个二选一选择器，对目的寄存器和源操作数寄存器相同的相邻指令采用旁路传播技术，如图 4 所示，当电路检测出数据相关后，写回信号有效且写回寄存器不为 x0，把数据从流水线寄存器直接读取到运算所需的 ALU 输入端，从而流水

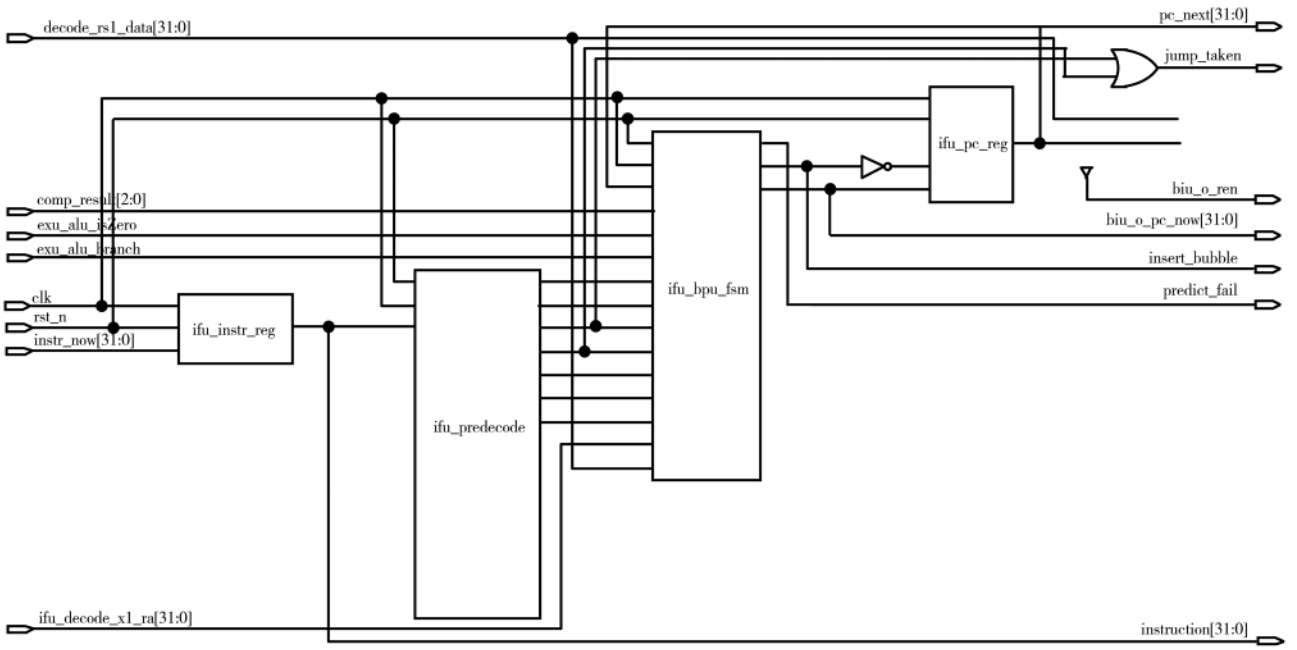

图 3 取值单元顶层框图

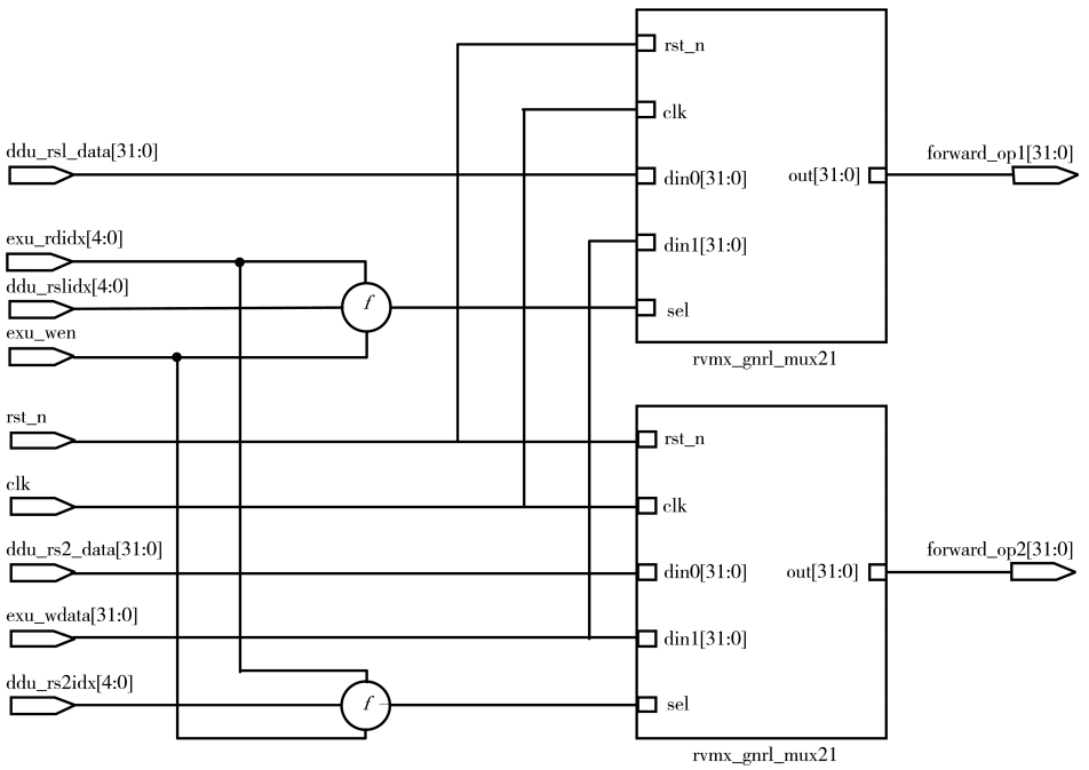

图 4 旁路传播单元框图

线可以正常顺序执行此类数据相关指令。设计完成的译码器单元如图 5 所示。

2.3 执行单元设计

在指令译码之后，执行部件便根据指令的具体操作类型发射给具体的运算单元进行执行，主要包括基本算术逻辑运算、移位运算、比较运算等。此外，执行单元中算数逻辑运算单元 ALU 需根据流水线状态解决可能出现的数据冒险。存储器访问单元，即访问存储器可以直接调用执行阶段的所有计算结果。

由于本处理器设计为三级流水线结构，因此在指令执行结束之后，会在同一时钟周期访问数据存储器，并且根据运算结果写回到对应的部件。写回是流水线最后一步，将指令的运算结果写回到通用寄存器堆。设计的执行单元顶层框图如图 6 所示。

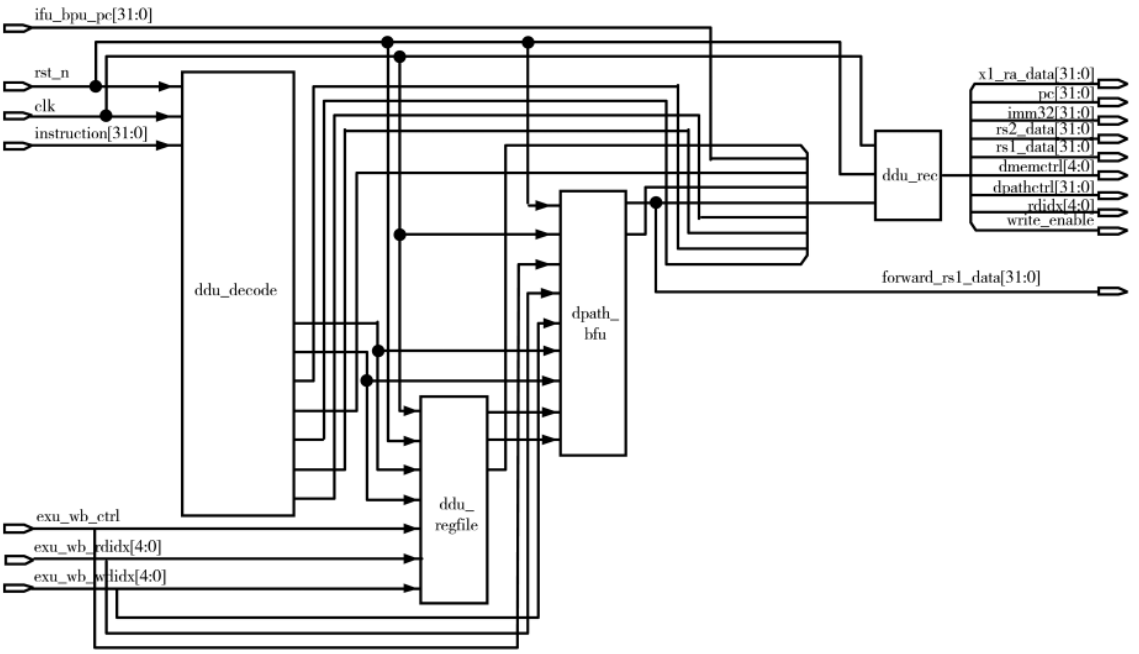

图 5 译码器单元顶层框图

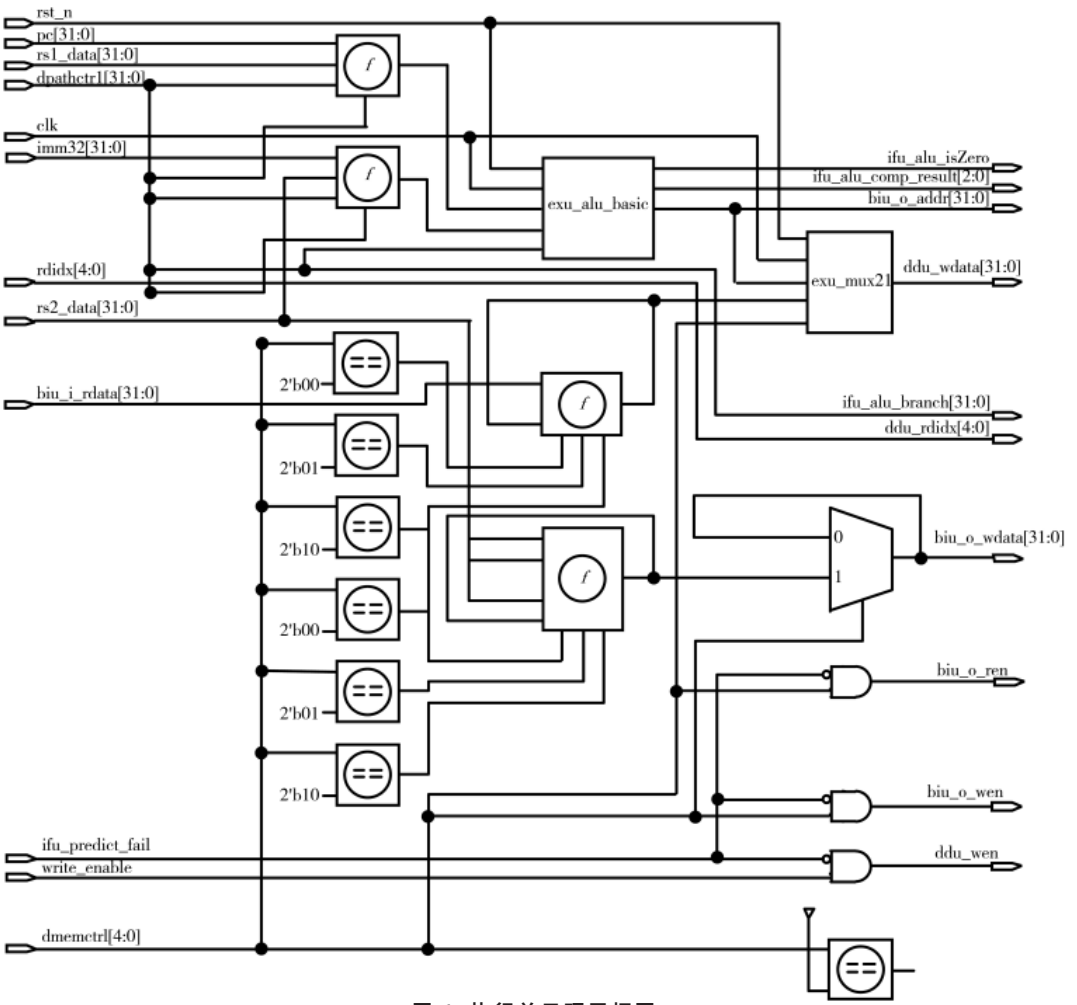

图 6 执行单元顶层框图

3 RISC-V 处理器功能验证

RISC-V 工具链由开源社区维护，可通过 RISC-V 基金会网站下载安装。在 Linux 环境下载 GNU 工具链，搭建交叉编译环境。将汇编语言程序借助编译工具链，生

成 Verilog 中 readmemh 函数可读入的二进制文件送入指令存储器，借助 VCS 和 Verdi 仿真并记录处理器运算波形。针对逻辑操作指令、移位指令、算数指令、跳转指令和存储指令分别进行了仿真验证。

3.1 逻辑操作指令

RISC-V 的 32 位基础指令集定义整数运算指令中的逻辑操作指令有 8 条：LUI、AUIPC、AND、ANDI、OR、ORI、XOR 和 XORI；当 PC 为 174 时，流水线仿真波形如图 7 所示。执行级算数逻辑单元 ALU 将根据送入的操作数进行运算。此时译码后输出的 ALU 控制信号为 XOR，ALU 的操作数执行 XOR 运算，并在执行级结束后输出执行结果。

图 7 XOR 指令验证波形

3.2 算数指令

RISC-V 的 32 位基础指令集定义整数运算指令中的简单算术操作指令有 7 条：ADD、ADDI、SUB、SLT、SLTU、SLTI 和 SLTIU，这些指令的基本实现都在执行级 EX 中通过相应的代码完成。当 PC 为 174 时，流水线仿真波形如图 8 所示，此时数据通路运算信号为 ADD。源操作数分别为 65536 和 -32 时，执行 ADD 运算，可以看到最后运算结果输出为 65504。

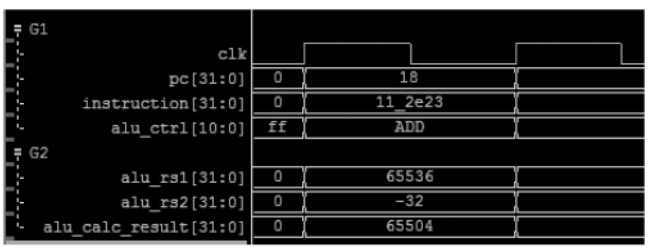

图 8 ADD 指令验证波形

3.3 移位指令

RISC-V 的 32 位基础指令集定义整数运算指令中的移位操作指令有 6 条：SLL、SLLI、SRL、SRLI、SRA 和 SRAI，这些指令的基本实现都在执行级 EX 中通过相应的代码完成。当 PC 为 174 时，流水线仿真波形如图 9 所示，可以根据汇编指令译码结果查看目的寄存器并添加至波形。译码后 ALU 需要执行的运算类型为 SRA，rd 寄存器位 x15，将源操作数运算结束后结果写回目的寄存器 x15。

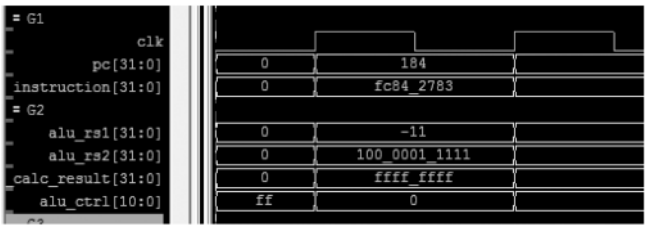

图 9 SRA 指令验证波形

3.4 跳转指令

控制转移指令 RISC-V 的 32 位基础指令集定义控制转移指令中的无条件跳转和条件分支指令有 8 条：JAL、JALR、BEQ、BNE、BLT、BLTU、BGE 和 BGEU，这两条指令的基本实现都在译码级 DDU 中通过相应的代码完成；如图 10 所示，对指令 d8000ef 按照 RV32I 编码规则译码后，为无条件跳转指令 JAL，并且跳转的立即数数值为 216。在图形界面查看波形，可以看到预译码指示信号 predec_jal 和实际跳转指示信号 jump_taken 均发生翻转，立即数生成器输出为 216，说明对无条件跳转的预测和执行都正确。

3.5 存储指令

RISC-V 的 32 位基础指令集定义的加载和存储指令有 8 条：SB、SH、SW、LB、LBU、LH、LHU 和 LW，这些指令的基本实现都在执行级 EXU 中访存单元相应的代码完成。对指令 2812c23 按照 RV32I 编码规则译码后，为存储指令 SW。在图形界面查看波形，可以看到译码为 STORE，如图 11 所示。数据存储器接收到 STORE 信号，状态标志 dmem_i_store 发生翻转，说明处理器对加载和存储指令的行为执行正确。

4 结论

为促进 RISC-V 处理器的应用和我国集成电路产业的发展，面向兼顾处理器性能和低功耗的设计目标，设计

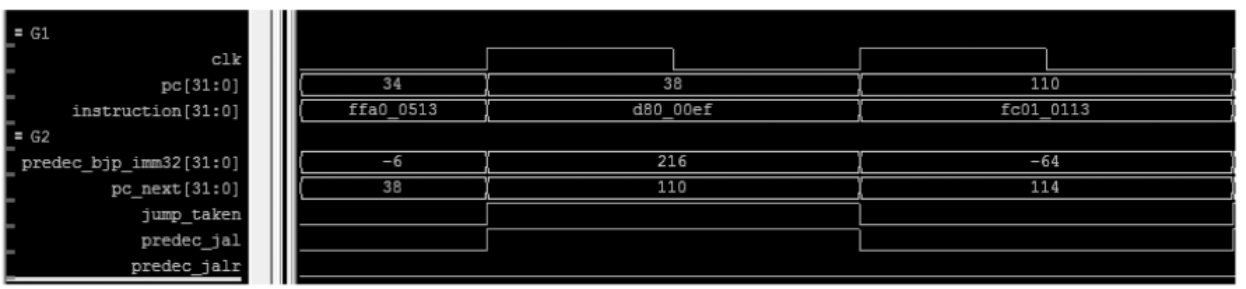

图 10 JAL 指令验证波形

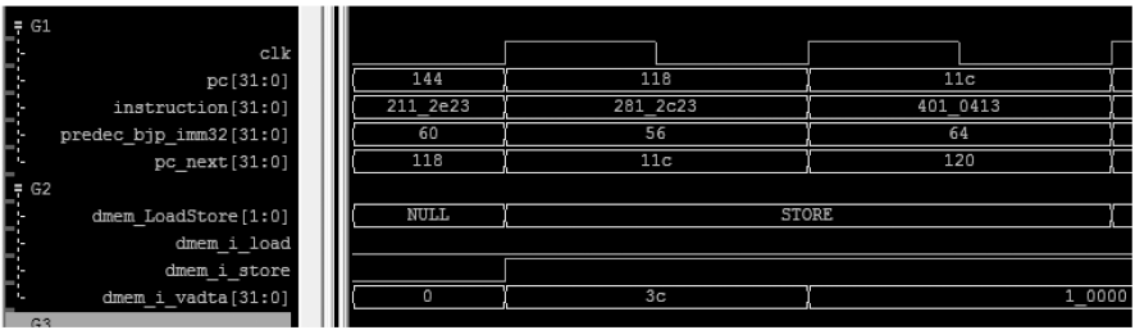

图 11 STORE 指令验证波形

了一种三级流水线的 RISC-V 处理器。使用 RV32I 整数运算指令集对处理器进行了仿真测试，结果表明，所设计的处理器功能正确，达到预定目标。

下一步将通过 SystemVerilog 语言结合 UVM 平台模块的工业级验证方法学，将各个子模块充分测试，实现完备的处理器设计与验证，并在 FPGA 硬件板卡中进行实现应用。

参考文献

[1] 蔡文伟.集成电路设计产业发展现状及对策分析[J].科技经济导刊，2019，27(30)：200-201.

[2] 张晔嘉.国内信息系统自主可控生态环境分析[J].电子质量，2019(7)：58-61.

[3] WATERMAN A，LEE Y，PATTERSON D A，et al.The RISC-V instruction set manual，volume I：User-level ISA[R].CS Division，EECE Department，University of California，Berkeley，2014.

[4] WATERMAN A S.Design of the RISC-V instruction set architecture[R].UC Berkeley，2016.

[5] 詹剑锋.论中国如何发展自主可控和开放的科技产业[J].中国科学院院刊，2019，34(6)：657-666.

[6] 李东泽，曹凯宁，曲明，等.五级流水线 RISC-V 处理器软硬件协同仿真验证[J].吉林大学学报(信息科学版)，2017，35(6)：612-616.

[7] HENNESSY J L，PATTERSON D A.Computer architecture：a quantitative approach[M].Elsevier，2011.

(收稿日期：2020-01-11)

作者简介：

折如义(1965-)，男，本科，副教授，主要研究方向：计算机系统。

李炳辉(1999-)，男，本科，主要研究方向：嵌入式系统。

姜佩贺(1988-)，通信作者，男，博士，讲师，主要研究方向：集成电路设计、芯片安全。

(上接第 43 页)

3 结论

基于 FPGA 的自定义指令集 CPU 可以处理较为复杂的算法，并且计算速度较快。在 FPGA 中构建这种指令集计算部分算法，可以分摊整个系统的算法计算工作量。在一颗 FPGA 芯片中可以构建多个这样的自定义 CPU，达到并行计算、分布式计算的目的。这种 CPU 可以灵活地配置和修改，满足一些特殊指令集的处理。

在实验中，伺服控制算法中的 FOC 算法其计算流程耗时只需要 7.48 μs。这已经比部分 Cortex-M 架构的 ARM 芯片的计算速度要快。

参考文献

[1] 钟瑜，吴明钦.一种高性能并行计算架构的 FPGA 实现[J].电讯技术，2019，59(7)：829-835.

[2] Webb，Warren.Customizable CPU platform combines DSP，FPGA[J].EDN，2008，53(6).

[3] 刘景文.基于 CPLD/FPGA 的 CPU 设计[J].数字技术与应用，2014(2)：162-164.

[4] 尼萨，周维译.计算机系统要素：从零开始构建现代计算机[M].北京：电子工业出版社，2007.

[5] 王培麟.FPGA 架构的 8 位 CISC CPU 设计[J].煤炭技术，2010，29(10)：212-214.

[6] 夏宇闻.Verilog 数字系统设计教程[M].北京：北京航空大学出版社，2008.

[7] 潘松，黄继业.EDA 技术与 VHDL[M].北京：清华大学出版社，2002.

(收稿日期：2020-01-06)

作者简介：

李俊(1975-)，男，硕士，高级工程师，主要研究方向：电源、变频器、伺服系统等开发研究工作。

任连新(1997-)，男，硕士研究生，主要研究方向：电力电子及运动控制技术。

廖振雄(1987-)，男，硕士，工程师，主要研究方向：伺服系统开发研究工作。

版权声明