

VLIW 处理器的设计与实现

唐 骞, 杨小雪

(西南交通大学 信息科学与技术学院, 四川 成都 610031)

摘要: 介绍了基于 FPGA 实现 VLIW 微处理器的基本方法, 对 VLIW 微处理器具体划分为 5 个主要功能模块。依据 FPGA 的设计思想, 采用自顶向下和文本与原理图相结合的流水线方式的设计方法, 进行 VLIW 微处理器的 5 个模块功能设计, 从而最终实现 VLIW 微处理器的功能, 并进行了板级功能验证。

关键词: VLIW; 微处理器; FPGA; 流水线

中图分类号: TP33

文献标识码: A

文章编号: 1674-7720(2010)11-0027-03

Design and implementation of VLIW microprocessor

TANG Qian, YANG Xiao Xue

(School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: This paper introduces the realization of the VLIW microprocessor based on FPGA and divides VLIW microprocessor into five basic modules. Based on FPGA design ideas, using "top and down" and schematic to design the VLIW microprocessor modules. The characteristic is making use of the pipeline of design and verifying the function on the development board.

Key words: VLIW; microprocessor; FPGA; pipeline

超长指令字 VLIW (Very Long Instruction Word) 微处理器架构采用了先进的清晰并行指令设计^[1]。VLIW 微处理器的最大优点是简化了处理器的结构, 删除了处理器内部许多复杂的控制电路^[2], 它能从应用程序中提取高度并行的指令数据, 并把这些机器指令均匀地分配给芯片中的众多执行单元^[3]。本设计是针对 VLIW 微处理器的基本功能设计实现的, 是针对 64 位指令字和 192 位数据进行操作处理, 主要功能是将指令和数据分别划分到 3 个并行操作单元中, 在执行单元中根据 3 个并行指令控制, 对 3 个并行操作单元的数据进行并行处理, 同时对操作处理数据进行存储管理。由于篇幅所限, 只是将取指令单元、译码单元、执行单元、写回单元和寄存器堆这 5 个单元的设计方法与设计原理作简要介绍, 没有给出各个模块设计仿真波形图。

1 VLIW 微处理器

1.1 VLIW 微处理的基本结构

VLIW 微处理器的基本架构如图 1 所示, 采用 4 级流水线的形式进行架构组织, 分别是取指令 (IF)、译码 (DE)、执行 (EXE)、写回 (WR)^[4]。VLIW 微处理器从外部储

存器中, 取出指令和数据, 指令通过译码操作, 要求处理器进行执行动作, 处理寄存器中的数据或者是取指令单元传输来的数据, 并通过写回单元, 将处理的数据存储到寄存器中。取指令单元是从存储器中取出相对应的指令和数据, 并将相应的信息传输给译码单元和寄存器堆单元; 译码单元的作用是将取指令单元的指令信息翻译为执行单元能够识别的操作, 将相应的信息传输到执行单元中; 执行单元是执行对应的指令信息和处理相关的数据, 同时也对寄存器堆送来的数据和指令进行相应的操作; 写回单元是将执行单元的数据存储到寄存器堆中; 寄存器堆是暂时存放取指令单元和写回单元的数据

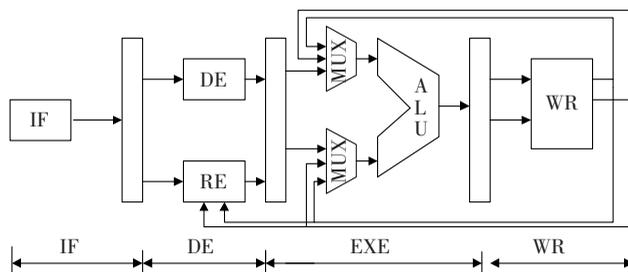


图 1 VLIW 微处理器的基本架构

或是指令。

1.2 VLIW 微处理器操作方式

VLIW 处理器指令操作方式是实行 3 种并行的操作,指令格式设置为:最高 4 位作为保留位,其余各 20 位依次划给操作 1、操作 2、操作 3,在每个 20 位中又分别留有 4 位的保留位,最后对 16 位的指令进行划分为操作码、源 1 地址、源 2 地址、目标地址。3 种操作方式是并行执行,相互独立,互不干扰,并且操作执行都是 64 位的数据;每一种操作方式都可以实现 16 种操作运算:空操作、加、减、乘、装载、移动、读、比较、或、与非、或非、取反、左移、右移、循环左移、循环右移等操作。任何一种操作都可以对 16 个寄存器中的数据进行操作处理,且寄存器中所存储的数据是 64 位。整个操作过程是将源 1 与源 2 寄存器中的数据进行某种操作运算,将其操作处理结果存入目标寄存器中,其操作处理方式是采用寄存器寻址方式。

2 VLIW 微处理器的设计与实现

根据 VLIW 微处理器系统架构和基本原理,将整个 VLIW 微处理划分为 5 个主要部分^[5],并对如图 2 所示的 5 个部分进行功能设计与实现。具体原理:取指令模块把外部存储器的数据和指令传送给译码模块,并将部分指令信息传送给寄存器堆,实现取指令的功能;译码模块将取指模块的数据和指令进行相对应的译码,实现某一种具体的操作,并将译码结果传送到执行模块中;执行模块将译码模块或者寄存器堆模块的数据和指令实现其操作的具体功能,并将操作结果传送到写回模块中;写回模块将执行模块的信息写入寄存器中,并输出相关操作处理的数据。

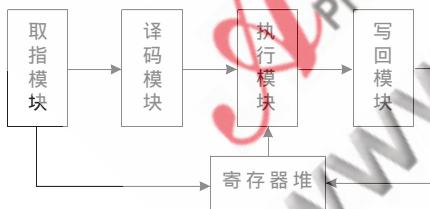


图 2 VLIW 处理器模块划分

2.1 取指令模块的设计

取指令模块的功能是从外部指令/数据的 Catch 取得 VLIW 指令和数据,将取得的信息传输给译码模块,让指令进行译码,同时也将取得的信息传输给寄存器堆模块,让执行模块执行操作进入内部寄存器而从寄存器堆取回数据。取指令模块设计实现如图 3 所示。将指令和数据通过选择器分配到 3 个操作并行单元中,然后细分操作执行的方式。具体划分为顶层模块和 3 种并行操作模块。顶层模块主要实现对指令和数据进行分流,分成操作码和操作地址的形式。由于都是并行操作,只需

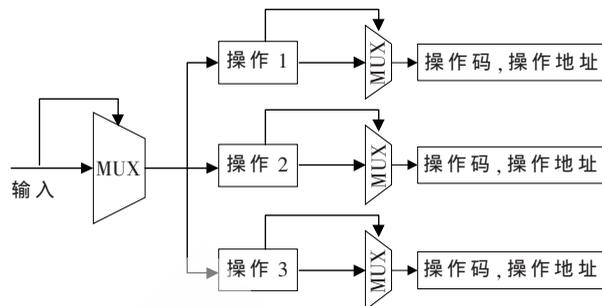


图 3 取指模块划分原理图

要写出操作 1 的功能,其余的操作功能一致,只是输入的操作指令和操作地址不一样。

2.2 译码操作模块的设计

译码操作模块的功能是将取指令模块传输来的信息进行译码操作,将所进行译码操作传输给执行模块。由于在进行译码操作时,操作 1、操作 2、操作 3 的指令与数据没有相互冲突,而且每一种操作的功能都是对 16 种操作进行相对应的译码,不同在于输入的各个操作的数据或指令的不相同,因此,在实现译码模块时,只需要实现一个操作译码的功能就可以。在改变输入的情况下就实现操作 2、操作 3 的 2 种操作译码的功能,从而可实现译码模块的整个功能。译码模块其划分原理如图 4 所示。

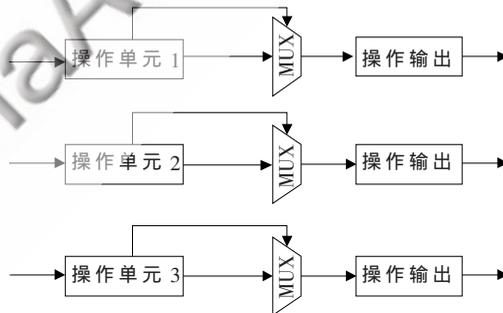


图 4 译码模块划分原理图

2.3 执行模块的设计

执行模块是在 VLIW 微处理器所有模块中最复杂的模块,其主要功能是执行 VLIW 指令操作,并将执行的操作结果送到写回模块中。在执行模块中采用寄存器旁路的特点,寄存器旁路用于处理指令或者数据之间的相互冲突问题。

将执行模块划分为:顶层模块、操作 1 模块、操作 2 模块、操作 3 模块、功能执行模块、异常处理模块。顶层模块实现 3 种操作的数据与指令分流和中间变量的处理;操作 1 模块实现寄存器旁路的功能和数据的处理;操作 2 模块和操作 3 模块实现的功能与操作 1 模块相同,不同点在于中间数据冲突处理不一样;功能执行模块实现 16 种操作的具体功能;异常处理模块实现数据

或指令的异常处理的具体功能。将各个模块连接就实现执行单元的整个功能。

2.4 寄存器堆的实现

寄存器堆的结构如图 5 所示。输入端的数据、指令是来自 2 个单元,一个是取指令单元数据和指令,另一个是写回单元的数据;输出的指令、数据要传输到执行单元中进行执行操作。因为 3 种操作是并行执行的,每一种操作功能都是一样,因此在某个输入信号下,经由某种操作单元时,在选择器 MUX 下,选择某种具体操作运算。在功能单元 FU 中,对取出的存储器数据进行相应的操作运算处理,并将操作处理的结果输出送到执行单元中。图中虚线表示将写回单元输送来的操作数据直接存储到存储器中。实现时,要注意处理数据的相关问题。

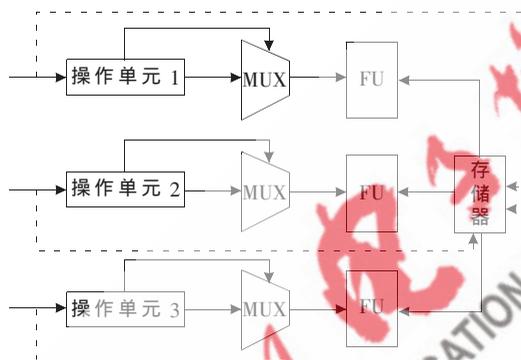


图 5 寄存器堆模块划分原理设计

2.5 写回单元的设计

写回单元是 VLIW 微处理器的最后一个单元,它的功能是将执行单元的操作结果写回到寄存器堆中,并在读有效的情况下,输出相应的操作处理数据。该模块设计相对比较简单,要实现写回寄存器堆的数据和该数据的目标寄存器地址,以及读出处理单元的操作处理数据。

2.6 综合仿真与测试

将上述所设计的模块,运用原理图的设计方法,按图 2 VLIW 微处理模块划分结构进行相应的连接,并选择相应的器件进行编译测试,其仿真测试功能波形如图 6 所示。图中的波形图添加了中间变量寄存器的波形图,以便能够更好地对比输出波形,分析处理器功能。VLIW 微处理器首先将要处理的数据装入寄存器中,利用寄存器寻址的方式,对数据进行相应的 16 种功能操作运算,然后利用读的功能读出操作处理的数据结果。

在图 6 中,添加的中间变量是寄存器地址(memor-

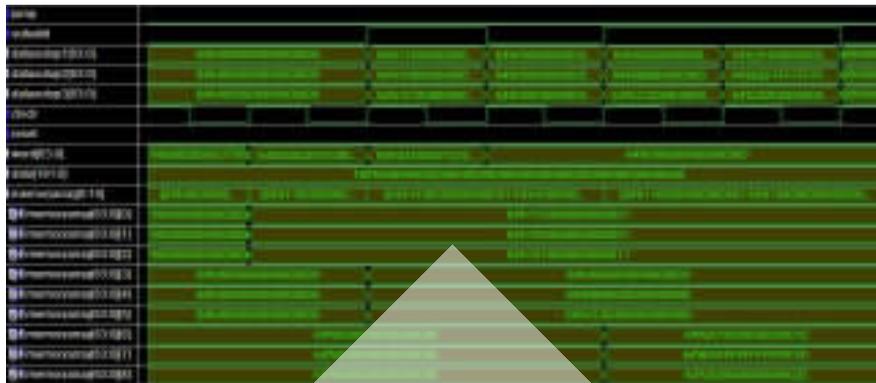


图 6 综合功能仿真

yarray),基本的输入有时钟信号(clock)、复位信号(reset)、指令(word)、数据(data)。复位信号为低电平时,电路处于有效的工作状态,输出相关的操作数据。运用装载指令对寄存器依次装入数据,在下一个时钟脉冲时,对寄存器的数据进行读出操作,输出数据有效信号输出高电平。要看到操作处理结果就必须用读的操作指令,操作执行的结果是存储到寄存器中。

基于 FPGA 实现微处理器的设计是一个热点,在主流设计中主要模块的划分一般大致相同,主要区别在于主要模块下的小模块划分是完全不相同的,并且实现与设计方式也不相同。在 VLIW 处理器设计中,细分各个小模块,运用硬件描述语言实现各个基本模块的功能,从而最终实现整个微处理器的逻辑功能,并在开发板进行相应的逻辑分析与功耗分析,为实际 DSP 并行处理器架构提供一定的参考基础。下阶段的主要方向是进行各个小部件的进一步优化设计,增强操作功能与操作指令的实现设计,进而设计实现浮点 VLIW 微处理器。

参考文献

- [1] 石教英. 计算机体系结构[M]. 杭州: 浙江大学出版社, 2004.
- [2] 尹朝庆. 计算机系统结构教程[M]. 北京: 清华大学出版社, 2005.
- [3] 李学干. 计算机系统的体系结构[M]. 北京: 清华大学出版社, 2006.
- [4] LEE W F. VLIW microprocessor hardware design for ASIC and FPGA [M]. The McGraw Companies, 2008.
- [5] FISHER J A, FARABOSCHI P, YOUNG C. Embedded computing: a VLIW approach to architecture [M]. Compilers and Tools, Denise E. M. Penrose, 2005.

(收稿日期: 2010-01-12)

作者简介:

唐骞,男,1978 年生,在读研究生,主要研究方向:数字集成电路、FPGA 设计。