

RV32IM 处理器乘法电路的设计与实现

张 凯¹, 李 涛^{1,2}, 秦晨蕊¹, 圣 飞¹

(1 西安邮电大学 电子工程学院, 陕西 西安 710121;

2 西安邮电大学 计算机学院, 陕西 西安 710121)

摘 要: 为了实现 RV32IM 处理器中整数乘法的操作, 对 RISC-V 指令集中整数乘法的“M”标准扩展进行实现. 设计中对于乘法指令的实现, 采用基 4 的 Booth 算法和 Wallace 树型 4-2 压缩器. 将该设计嵌入到 RV32IM 处理器中, 通过仿真和 SMIC 65 nm 高密度标准单元库进行综合表明: 该乘法单元电路功能正确且显著提高了乘法的运算效率, 最大工作频率能够达到 500 MHz.

关键词: 乘法器; Booth 算法; 4-2 压缩器; RV32IM 处理器; RISC-V 指令集

中图分类号: TP302

文献标识码: A

文章编号: 1000-7180(2018)09-0125-04

Design and Implementation of Multiplication Circuit in RV32IM Processor

ZHANG Kai¹, LI Tao^{1,2}, QIN Chen-rui¹, SHENG Fei¹

(1 School of Electronic Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710121, China;

2 School of Computing, Xi'an University of Posts & Telecommunications, Xi'an 710121, China)

Abstract: In order to achieve the integer multiplication operation in the RV32IM processor, The "M" standard extension of the integer multiplication of the RISC-V instruction set is implemented. For achieving the design for multiplication instructions, it uses the Booth algorithm of Base 4 and the Wallace Tree 4-2 Compressor. The design embedded in the RV32IM processor, through the simulation and SMIC 65nm high density standard cell library to show that: The multiplication circuit functions correctly and significantly improved the operation efficiency of the multiplication, the maximum operating frequency can reach 500MHz.

Key words: multiplier; Booth algorithm; 4-2 Compressor; RV32IM processor; RISC-V instruction set

1 引言

RV32IM 处理器是由最简单的移位相加乘法算法实现串行乘法器^[1], 这种算法比较简单, 硬件实现起来也比较容易, 但是缺点是每完成一次乘法运算需要很多拍, 这样大大降低了处理器的性能. 因此, 需要设计高速的乘法器, 而高速乘法器的设计关键是要尽可能的减少进入加法阵列的部分积数目以及采用合适的加法阵列压缩器提高部分积的求和速度, 为了减少求和部分积的数目, 本文采用基 4 的 Booth 编码算法^[2], 使部分积的数目减少为原来算

法的一半; 同时为了采用合适的加法阵列压缩器, 本文对两个 3-2 压缩器组成的 4-2 压缩器^[3]进行了优化, 设计了以改进 4-2 压缩器为核心的新型 Wallace 树压缩架构, 该架构根据部分积的数目, 将压缩器进行合理的排列, 使其组成的新型压缩架构既减少了硬件资源消耗又缩短了关键路径的延迟.

2 乘法算法原理

2.1 Booth 编码算法

Booth 编码算法是乘法器中生成部分积最有效的编码方法, 本设计采用的是基 4 的 Booth 编码

算法,其编码的优点是:对有符号数和无符号数可以用统一的编码方法,只需要做相应的符号位扩展;基 4 的 Booth 编码算法可以使部分积数目大大缩减,是基 2 的 Booth 编码算法部分积的一半,直接减少 Wallace 树型压缩器的级数,从而提高了乘法运算的速度. 基 8 的 Booth 编码算法虽然生产部分积数目更少,理论上速度更快,但是因为其编码中会产生 3 倍的部分积,通过简单的移位操作已经不能解决,复杂程度大大增加,不容易实现,因此采用基 4 的 Booth 编码算法是最可靠、有效的方法. 基 4 的 Booth 编码算法部分积产生规则如下表 1 所示.

表 1 基 4 的 Booth 算法部分积产生规则

y_{i+1}	y_i	y_{i-1}	操作
0	0	0	$+0$
0	0	1	$+ [X]_{\text{补}}$
0	1	0	$+ [X]_{\text{补}}$
0	1	1	$+ 2[X]_{\text{补}}$
10	0		$- 2[X]_{\text{补}}$
10	1		$- [X]_{\text{补}}$
1	10		$- [X]_{\text{补}}$
1	1	1	$+0$

2.2 Wallace 树型 4-2 压缩器

部分积的压缩是作为乘法器的另一个关键问题,Wallace 树可以用简单的硬件快速的把多个数相加归约成两个数相加,从而提高在乘法运算过程中多个部分积相加的速度. Wallace 树型压缩结构的核心就是压缩器,最常见的就是 3-2 压缩器,即进位保留加法器,通常 4-2 压缩器是由两个 3-2 压缩器实现的,其逻辑表达式为:

$$C_{out} = ab + bc + ac \tag{1}$$

$$Sum = a \oplus b \oplus c \oplus d \oplus C_{in} \tag{2}$$

$$Carry = (a \oplus b \oplus c) d + (a \oplus b \oplus c) C_{in} + d C_{in} \tag{3}$$

由两个 3-2 压缩器实现的 4-2 压缩器在电路的关键路径上需要 4 个异或门的延时,并且输入的信号不是同时参与运算的,在累加的过程中会很容易产生电路运算问题的,因此,要对电路做一些优化. 通过对传统的 4-2 压缩器逻辑表达式的研究,对其进行一些变形和化简,可以达到对电路优化的目标,其化简后的逻辑表达式如下:

$$C_{out} = (a \oplus b) c + \overline{(a \oplus b)} a \tag{4}$$

$$Carry = (a \oplus b \oplus c \oplus d) C_{in} + \overline{(a \oplus b \oplus c \oplus d)} d \tag{5}$$

优化后的 4-2 压缩器如下图 1 所示,此电路的

结构不仅使 Sum 的关键路径延时为 3 个异或门延时,同时也使 4 个输入同时参与运算.

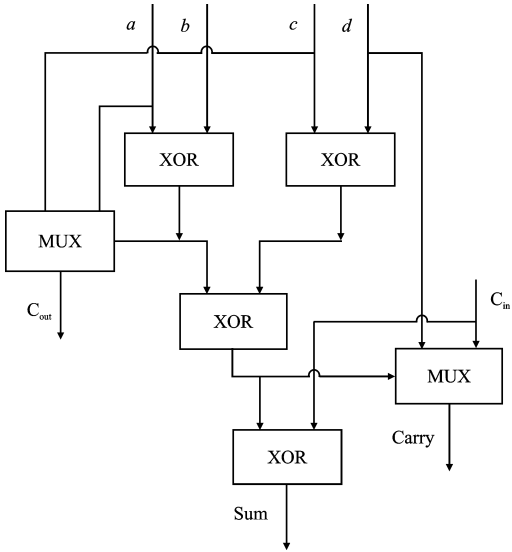


图 1 优化后的 4-2 压缩器逻辑图

3 RV32IM 处理器乘法器设计

在 RV32IM 处理器中,乘法运算作为独立的功能模块,其要与通用寄存器之间进行数据的交换^[4],即要从寄存器中取数据,或者要将乘法运算的结果存入到寄存器中,以便提供给其他功能模块使用. 整数乘法指令如下:

a. 乘法 MUL 指令:执行一个 32 位 * 32 位的乘法,并将运算结果的低 32 位存入到目标寄存器中,指令格式 MUL rdl,rs1,rs2.

b. 乘法 MULH 指令:执行一个有符号 32 位 * 有符号 32 位的乘法,并将运算结果的高 32 位存入到目标寄存器中,指令格式 MULH rdh,rs1,rs2^[5].

c. 乘法 MULHU 指令:执行一个无符号 32 位 * 无符号 32 位的乘法,并将运算结果的高 32 位存入到目标寄存器中,指令格式 MULHU rdh,rs1,rs2.

d. 乘法 MULHSU 指令:执行一个有符号 32 位 * 无符号 32 位的乘法,并将运算结果的高 32 位存入到目标寄存器中,指令格式 MULHSU rdh,rs1,rs2.

在 RV32IM 处理器中,如果同时需要乘法运算结果的高位和低位,那么执行的指令顺序为: MULH[S][U] rdl,rs1,rs2; MUL rdl,rs1,rs2^[6]. RV32IM 处理器中乘法器整体框图如下所示.

该 32 位定点补码乘法器的设计包含控制结果处理模块和乘法运算模块两个模块.

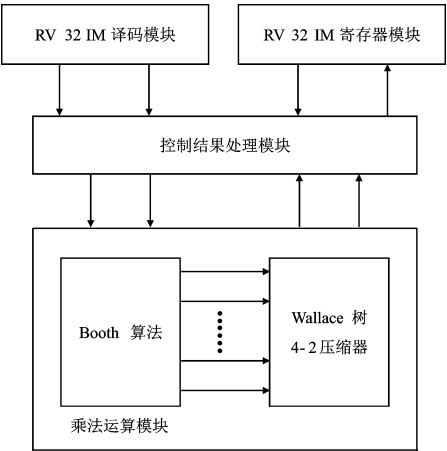


图 2 RV32IM 中乘法器整体框图

3.1 控制结果处理模块

控制结果处理模块其实是一个状态机,处理各个请求的状态跳转,该模块划分为如图 3 所示的 4 个状态,这 4 个状态分别为 s_idle,s_compute,s_setup_output 和 s_done,其中 s_idle 状态主要完成对来自 RV32IM 处理器中译码模块的控制信号和数据的处理;s_compute 状态主要完成对乘法指令的执行,即完成乘法运算操作;s_setup_output 状态首先完成对乘法运算结果高 32 位和低 32 位的选择,然后将选择的 32 位结果存入到 RV32IM 处理器的寄存器中,以便其他模块使用;s_done 结束状态直接跳回到 s_idle 空闲状态,继续检测 RV32IM 处理器中译码模块的控制信号和数据的处理. 控制结果处理模块状态机如下图 3 所示.

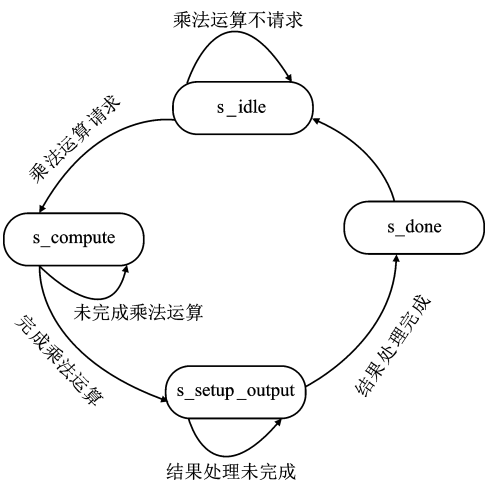


图 3 控制结果处理模块状态机

3.2 乘法运算模块

乘法运算模块包含 Booth 编码算法模块和 Wallace 树型 4-2 压缩器模块两个模块,它们分别完

成了乘法部分积的产生和部分积的压缩功能.

3.2.1 Booth 编码算法模块

Booth 编码算法模块主要完成乘法运算部分积的生成.

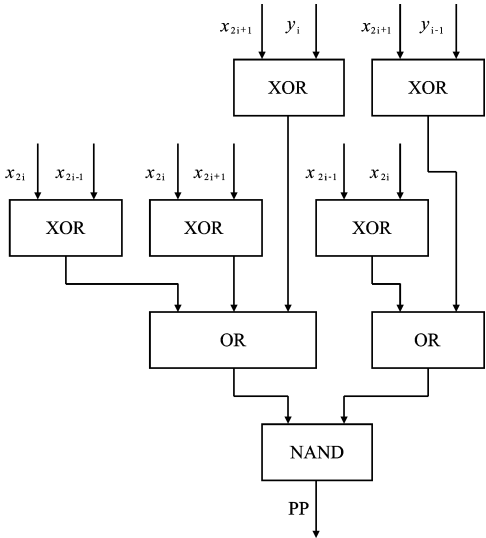


图 4 部分积产生电路框图

图 4 部分积产生电路框图采用基 4 的 Booth 编码算法,可以每次扫描乘数的 3 位,本位、相邻高位、相邻低位,根据乘数的 3 位选择 $+0$ 、 $+ [X]_{补}$ 、 $+2[X]_{补}$ 、 $-2[X]_{补}$ 、 $- [X]_{补}$. 其 $2[X]_{补}$ 是把 $[X]_{补}$ 左移一位, $- [X]_{补}$ 是把 $[X]_{补}$ 按位取反后加 1.

3.2.2 Wallace 树型 4-2 压缩器模块

Wallace 树型 4-2 压缩器模块主要完成对部分积压缩的功能. 通过基 4 的 Booth 编码算法模块,总共产生有 16 个部分积,如果完全采用 3-2 压缩器对 16 个部分积进行压缩,则需要 6 级进位保留加法器才能将其压缩为两个部分积,如果混合采用 3-2 压缩器和 4-2 压缩器进行压缩,则需要 4 级进位保留加法器才能将其压缩为两个部分积. 最后将压缩得到的两个部分积进行相加,就可得到乘法运算的最终结果.

为了节约硬件资源消耗和减少电路的延时,本文设计采用的是 4-2 压缩器进行压缩,只需要 3 级进位保留加法器就能将其压缩为两个部分积. Wallace 树压缩器结构如下图 4 所示.

此结构与完全采用 3-2 压缩器结构和混合结构相比,本设计的 Wallace 树压缩器结构,比这两种结构消耗的 logic 资源要少,并且关键路径的延时也要小.

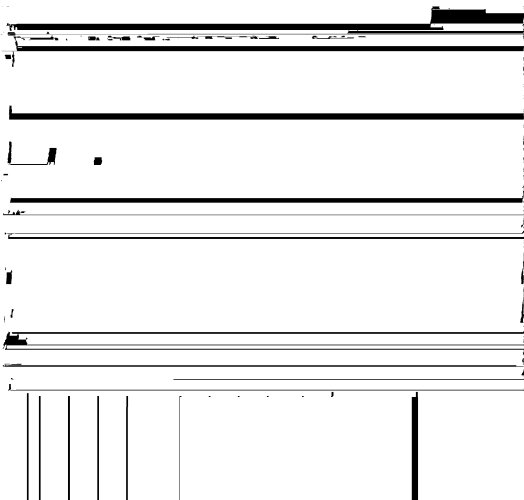


图 5 Wallace 树压缩器结构

4 仿真结果与分析

4.1 仿真结果分析

运用 Modelsim 软件仿真工具对本设计进行逻辑功能验证。首先,通过搭建 SystemVerilog 验证平台随机产生 100 万测试激励,对设计进行验证,经过验证,所得结果正确。然后,将本设计的乘法器嵌入到 RV32IM 处理器中,执行乘法指令 MUL rd2, rs5, rs8; MUL rd3, rs2, rs8; MUL rd4, rs5, rs8; MULH rd6, rs5, rs8;通过 Modelsim 仿真所得功能仿真波形图如图 6 所示。

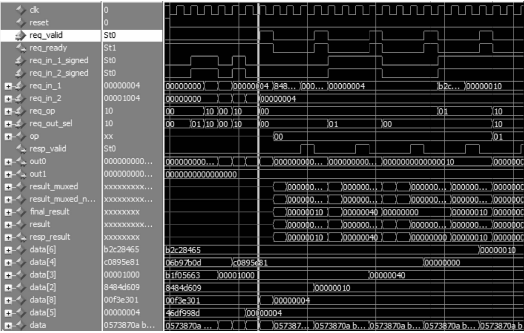


图 6 RV32IM 处理器执行乘法指令仿真波形

4.2 综合结果分析

为了对 RV32IM 处理器原有乘法器和本文设计的乘法器的性能,采用 SMIC 公司 65 nm,1.2 V, SC9 高密度标准单元库,对其进行 DC 综合,综合后的结果如表 2。

综合结果表明,本文设计的定点补码乘法器的时钟频率最高可达 500 MHz,电路面积 0.011 5 mm²,总单元数目为 4 256。相比于 RV32IM 处理器

原有乘法器,本文设计的乘法器性能更高,电路最大延时减少了 8.5%,速度性能改善了 94.3%。

表 2 乘法器综合后性能对比表

参数	原乘法器	本文乘法器
电路最大延时/ns	2.58	2.36
所需时钟周期数	32	2

5 结束语

本文针对 RISC-V 指令集中整数乘法指令的特点,设计了一种应用于 RV32IM 处理器中的定点补码乘法器,完成了功能仿真,通过 Synopsys DC 进行综合表明:相比于原乘法器,该电路最高时钟频率为 500 MHz,速度性能改善了 94.3%,非常适用于实现高效的乘法指令运算。

参考文献:

[1] Yunsup Lee, Andrew Waterman, Henry Cook, et al. An Agile Approach to Building RISC-V Microp-rocessors[J]. IEEE Micro. 2016, 36(2): 8-20.

[2] 石敏,王耿,易清明. 基于改进的 Booth 编码和 Wallace 树的乘法器优化设计[J]. 计算机应用与软件, 2016, (5): 13-16.

[3] 袁寿财,朱长纯. 快速乘法器中高速 4-2 压缩器的设计(英文)[J]. 微电子学与计算机. 2002, 19(4): 53-56.

[4] Aneesh Raveendran, Vinayak Baramu Patil, David Selvakumar, et al. A RISC-V Instruction Set Processor-Micro-architecture Design and Anal-ysis[C]. 2016 International Conference on VLSI Systems, Architectures, Technology and Applica-tions. 2016: 1-7.

[5] Andrew Waterman, Yunsup Lee, Rimas Avizieni-s, et al. The RISC-V instruction set[J]. 2013 IEEE Hot Chips 25 Symposium . 2013: 1-1.

[6] Yunsup Lee, Andrew Waterman, Henry Cook, et al. An Agile Approach to Building RISC-V Microp-rocessors[J]. IEEE Micro. 2016, 36(2): 8-20.

作者简介:

张 凯 男, (1990-), 硕士研究生. 研究方向为电路与系统. E-mail: 295952575@qq. com.

李 涛 男, (1954-), 博士, 教授. 研究方向为计算机体系结构、计算机图形学、大规模集成电路.

秦晨蕊 女, (1993-), 硕士研究生. 研究方向为电路与系统.

圣 飞 男, (1990-), 硕士研究生. 研究方向为电子通信与工程.