

引用格式:姚上上,沈立. 基于混合压缩结构的新型浮点乘法器设计[J]. 微电子学与计算机, 2021, 38(9): 74-78. [YAO S S, SHEN L. Design of a new floating point multiplier based on hybrid compression structure [J]. Microelectronics & Computer, 2021, 38(9): 74-78.]

基于混合压缩结构的新型浮点乘法器设计

姚上上, 沈 立

(国防科技大学 电子科学与技术系, 湖南 长沙 410073)

摘 要: 为了进一步提高浮点乘法器的性能, 缩短浮点乘法器关键路径延时, 提出了一种基于新型 4-2 压缩器和 5-2 压缩器的混合压缩结构. 在 Xilinx 的 xc7a35tcsg324 开发板上, 基于该结构实现了 IEEE754 标准的 32 位浮点乘法器. 相较于现有的压缩方式, 提出的新型压缩结构相较于现有的压缩方式, 所使用的 LUT 资源减少了 45, 关键路径延时减少了 0.004 ns. 与传统浮点乘法器相比, 关键路径延时由 6.022 ns 缩短至 4.673 ns, 提升了浮点乘法器的运算性能.

关键词: 4-2 压缩器; 5-2 压缩器; Wallace 树; 浮点乘法器

中图分类号: TN492

文献标识码: A

文章编号: 1000-7180(2021)09-0074-05

Design of a new floating point multiplier based on hybrid compression structure

YAO Shangshang, SHEN Li

(Department of Electronic Science and technology, University of Defense Science and technology,
Changsha 410073, China)

Abstract: In order to further improve the performance of Floating-Point Multiplier and shorten the critical path delay of Floating-Point Multiplier, a hybrid compression structure based on new 4-2 compressor and 5-2 compressor is proposed. On the xc7a35tcsg324 development board of xilinx, the 32-bit Floating-Point Multiplier of IEEE754 standard is implemented based on this structure. Compared with the existing compression methods, the LUT resource and critical path delay are reduced by 45 and 0.004ns respectively. Compared with the traditional Floating-Point Multiplier, the critical path delay is reduced from 6.022ns to 4.673ns, which improves the performance of the Floating-Point Multiplier.

Key words: 4-2 compressor; 5-2 compressor; Wallace tree; floating point multiply

1 引言

在现代处理器中, 浮点乘法器是计算单元的核心部件, 在浮点乘法器的设计过程中, 制约性能的瓶颈在于部分积的产生以及部分积的压缩过程^[1]. 使用 Booth 算法可以将部分积数量减少为一半或者更少, 而使用 Wallace 树的方法可以将生成的部分积并行压缩. 根据文献[2]得出的综合结果, Booth 编

码部分与 Wallace Tree 在乘法器的关键路径延迟时间中大约占了 38.08%. 因此对于 Booth 编码与 Wallace Tree 结构的优化成了提升乘法器性能的关键. 文献[3]简化了每个部分积的符号扩展位, 使部分积的排列方式更加紧凑规则, 文献[4]采用了 7-2 压缩器结构的 Wallace Tree 以及 64 位 Brent Kung 树结构的超前进位加法器, 提高了乘法器的运行速度, 文献[5]提出了一种截断式的加法器进位链, 但

是在对部分积进行压缩处理的关键路径上仍然存在着较大的延时,在文献[6]中,提出了一种新型的 Booth 选择器以及 4-2 压缩器,有效的缩短了关键路径的延时. 本文将 4-2 压缩器以及 5-2 压缩器进一步改进,并重新组合,提出了一种基于 32 bit 浮点乘法运算的新型 Wallace Tree 结构,同时对部分积产生电路进行了优化,使浮点乘法器的整体性能与文献[5]相比得到了显著提升.

2 浮点乘法器

现在广泛所采用的均为 IEEE754 标准^[7]的 32 bit 二进制单精度浮点数. 32 bit 的浮点数分为三个部分,其中最高位 31 位为浮点数的符号位(Sign),0 表示浮点数为正数,1 表示浮点数为负数;指数位(Exponent),为浮点数的第 30~23 位,其中包含了 127 的偏置,原指数为 E-127;尾数位(Mantissa)为第 22~0 位,表示了浮点数的小数部分,其中包含了整数部分 1. 所以原浮点数的表示方法为:

$$F = (-1)^S \times 1.M \times 2^{E-127}$$

(1)

本文基于基 4Booth 算法以及 Wallace 树,实现了 32 bit 浮点数的计算过程,流水线级数为四级. 大致的框架如图 1 所示.

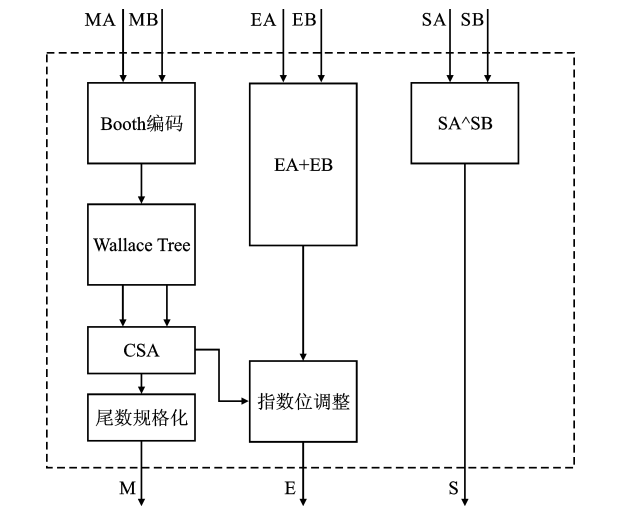


图 1 浮点乘法器结构

(1)尾数计算,首先将乘数与被乘数的尾数经过 Booth 选择器进行编码,产生 13 个部分积,再由 WallaceTree 将 13 个部分积压缩成两个部分和,然后用 CSA 进行求和,再经过规范化处理之后即可得出 23 位尾数部分. 尾数部分的处理为本文重点讨论的部分.

(2)指数计算,主要为乘数与被乘数的指数部分求和,根据溢出情况以及尾数的舍入结果进行调整,

最后加上 127 的偏置值得到最终的指数结果.

(3)符号计算,符号位计算只需将乘数与被乘数的符号位进行异或操作即可.

3 基 4 Booth 编码设计

一般来说,高性能的乘法器都会采用 Booth 编码的方案来加快部分和的生成. 基 2 的 Booth 编码方案与基 4 的 Booth 编码方案相比,因为编码之后产生的部分积的个数与乘数的位数相同,因此随着乘数位数的上升,会产生数目较多的部分和. 而在基 8 的 Booth 算法中,由于会出现三倍的部分和的问题,使设计更为复杂,所以本设计所采用的编码方案为基 4 的 Booth 编码方式. 基 4 的 Booth 算法的计算公式可以使用如下的数学模型表示:

$$P = A \times B$$
$$= A \times \sum_{i=0}^{23} b_i \cdot 2^i$$
$$= A \times \sum_{i=1}^{12} ((-2b_{[2i+1]} + b_{[2i]} + b_{[2i-1]}) \times 2^{2i})$$

(2)

在编码的过程中,首先在尾数的最高位之前补符号位 0,末尾添 0,为了使长度为奇数在符号位前再补 0,根据 Booth 编码对应的编码规则产生部分积. 其编码方式如表 1 所示,Booth 编码卡诺图如图 2 所示.

表 1 基 4Booth 编码

$b_{[n+1]}$	$b_{[n]}$	$b_{[n-1]}$	部分积
0	0	0	+0
0	0	1	+A
0	1	0	+A
0	1	1	+2A
1	0	0	-2A
1	0	1	-A
1	1	0	-A
1	1	1	-0

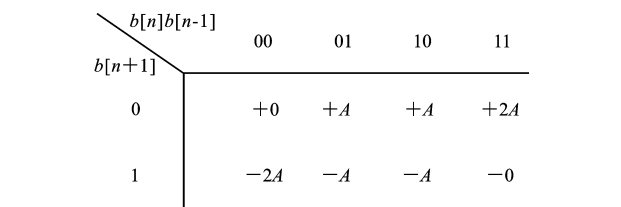


图 2 Booth 编码卡诺图

根据真值表,以及卡诺图,化简,变形的方 法,得出部分积的表达式(3):

$$\begin{aligned}
P = & (\overline{b_{[n+1]}} \cdot (b_{[n]} \parallel b_{[n-1]})) \cdot A \\
& + (\overline{b_{[n+1]}} \cdot (b_{[n]} \cdot b_{[n-1]})) \cdot 2A \\
& + (b_{[n+1]} \cdot (b_{[n]} \oplus b_{[n-1]})) \cdot (-A) \\
& + (b_{[n+1]} \cdot \overline{(b_{[n]} \parallel b_{[n-1]})}) \cdot (-2A) \quad (3)
\end{aligned}$$

式中, $2A$ 为 A 左移一位, $-A$ 为 A 的值取反后加 1, $-2A$ 为 A 的值取 2 倍后取反加 1. 该表达式可以根据不同的乘数位 b 的值, 自动生成部分积 P , 由于没有额外产生 0 的表达式, 电路结构更加简洁, 所耗的逻辑资源更少.

4 4-2 压缩器设计

文献[4]所使用的 4-2 压缩器由两个 3-2 压缩器级联组成, 如图 3 所示, 该压缩器的工作原理是将三个输入信号 X_1, X_2, X_3 送入第一级 3-2 压缩器中, 将产生的 Sum, X_4, C_{in} 送入第二级 3-2 压缩器中, 最终得到 Sum 和 Carry 信号, 由于在一级 3-2 压缩器中存在 2 级异或门的延迟, 所以整个 4-2 压缩器的关键路径延迟为 4 级异或门, 信号延时较大.

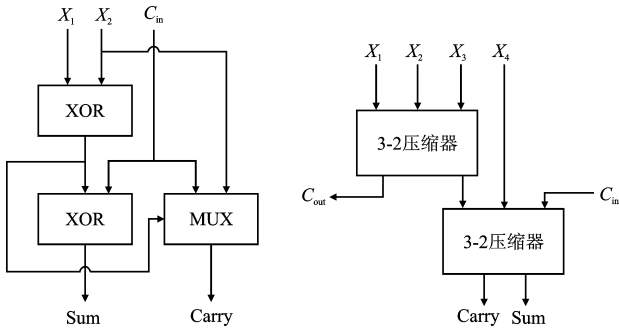


图 3 传统的 3-2 压缩器(左)和 4-2 压缩器(右)

针对于此问题, 本文提出了一种新型的 4-2 压缩器, 如图 4 所示, 此电路的关键路径延迟为 3 级异或门, 相比于文献[4], 进位电路的结构进一步简化, 逻辑门数更少, 同时满足 4-2 压缩器的基本公式(4):

$$\begin{aligned}
X_1 + X_2 + X_3 + X_4 + C_{in} = \\
\text{Sum} + 2 \times (\text{Carry} + C_{out}) \quad (4)
\end{aligned}$$

5 5-2 压缩器设计

传统的 5-2 压缩器, 有五个输入 X_1, X_2, X_3, X_4, X_5 , 两个输出 Sum, Carry. 两个进位输入信号 C_{in1} 和 C_{in2} , 两个进位输出信号 C_{out1} 和 C_{out2} . 由于 3-2 压缩器存在两级异或门延迟, 所以在传统的 5-2 压缩器的关键路径上存在 6 级异或门的延迟. 在文献[8]中, 提出的 5-2 压缩器可以很好的改善 C_{in1} 到 C_{out2} 的延迟问题, 但是在产生 C_{out1} 的信号时, 需要使

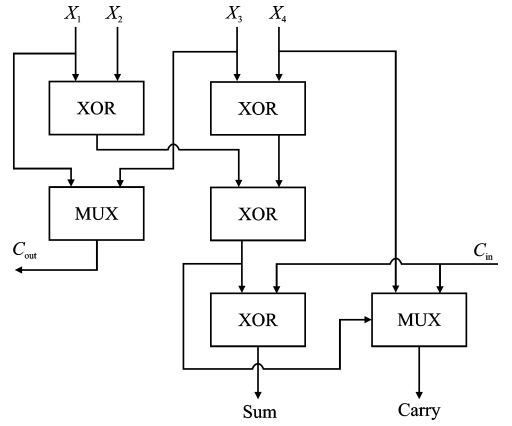


图 4 优化后的 4-2 压缩器

用额外的逻辑门电路, 本文提出的 5-2 压缩器逻辑表达式如式(5)~式(8)所示, 在产生 C_{out1} 和 C_{out2} 的信号时, 利用了产生 Carry 和 Sum 所需的 $X_1 \oplus X_2$ 信号, 有效的减少了进位输出电路所需的逻辑门数.

$$\text{Sum} = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus X_5 \oplus C_{in1} \oplus C_{in2} \quad (5)$$

$$C_{out1} = (X_1 \oplus X_2) \cdot X_3 + \overline{(X_1 \oplus X_2)} \cdot X_1 \quad (6)$$

$$C_{out2} = \overline{(X_4 \oplus X_5)} \cdot X_4 + (X_4 \oplus X_5) \cdot (X_1 \oplus X_2) \quad (7)$$

$$\begin{aligned}
\text{Carry} = & (C_{in1} \oplus C_{in2}) \cdot (X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus X_5) \\
& + \overline{(C_{in1} \oplus C_{in2})} \cdot C_{in1} \quad (8)
\end{aligned}$$

本文提出的新型 5-2 压缩器的关键路径仅存在 4 级异或门的延迟, 两个进位输出信号 C_{out1} 和 C_{out2} 可以实现同时输出, 减少了下一级异或门的等待时间, 其电路结构如图 5 所示. 与传统的 3 级 3-2 压缩器结构相比, 节约了两级异或门的延迟.

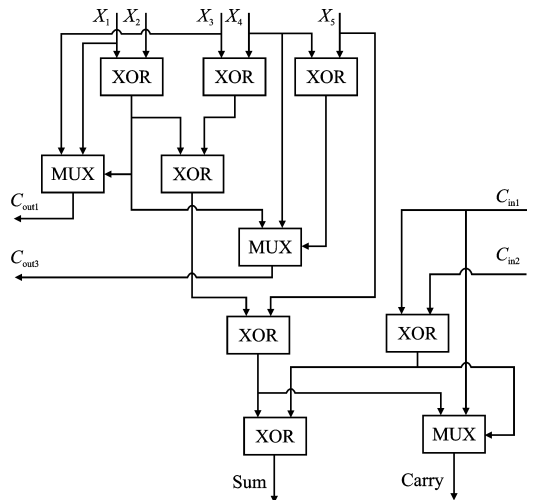


图 5 优化后的 5-2 压缩器

6 Wallace Tree 结构

Wallace Tree 主要是对 Booth 编码完成后的部分积进行压缩,该乘法器总共产生了 13 个部分积,在文献[5]中,使用了 6 个 4-2 压缩器进行压缩,使用了 24 个异或门,12 个数据选择器。

本文设计的 Wallace Tree 结构如图 6 所示,使用了新型 4-2 压缩器以及 5-2 压缩器之后异或门的个数为 23 个,二选一数据选择器的个数为 11 个,压缩级数为 3 级,第一级使用 3 个 4-2 压缩器,对 Booth 选择电路产生的前 12 个部分积进行压缩运算,第二级使用两个 3-2 压缩器对生成的 6 个部分和进一步压缩,最后一级使用本文提出的 5-2 压缩器对上一级生成的 4 个部分和以及第 13 个部分积进行压缩。最终产生的两行部分积使用 32 位 CSA 进行求和计算。该结构中,每一级压缩器可以相互独立的运行,很大程度上增加了运算的并行性,提高了运算速度。

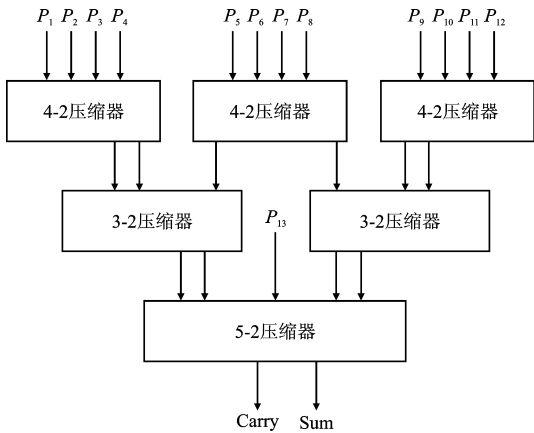


图 6 Wallace Tree 结构

7 仿真结果与分析

本设计采用 Verilog HDL 语言进行描述,使用 Modelsim10.2 进行仿真,基于 Vivado 2020.1 进行综合,并在 Xillinx 的 xc7a35tcsq324 开发板上实现了该浮点乘法器,在相同的综合策略(PerOptimized_high)下与传统的浮点乘法器所使用的 LUT 资源,关键路径延时等参数进行比较。表 2 列出了本文提出的 4-2 压缩器,与文献[4]中的 4-2 压缩器的性能对比,表 3 列出了本文所使用的新型 WallaceTree 与文献[5]的对比。表 4 列出了本文所提出的四级流水线浮点乘法器与传统浮点乘法器的性能对比。通过对比分析,本文所设计的 4-2 压缩器与文献[4]所

使用的结构相比,功耗略低,关键路径延时缩短约 19.6%,在 Wallace Tree 的性能方面,本设计所使用的二选一数据选择器,异或门的数量,比文献[5]所使用的资源更少,在关键路径延迟相差不大的情况下,使用的 LUT 资源由 1 652 减少至 1 607,下降了 2.7%。与传统的浮点乘法器相比,关键路径的延时由 6.022 ns 下降至 4.673 ns,缩短约 22.4%。最高运行频率提升约 28.9%。

表 2 4-2 压缩器性能对比

	主要功耗/W	关键路径延时/ns
新型 4-2 压缩器	0.215	1.899
文献[4]中 4-2 压缩器	0.216	2.362

表 3 Wallace Tree 性能对比

	MUXs	XORs	Slice LUTs	关键路径延时/ns
新型 WallaceTree	11	23	1 607	4.673
文献[5]中 WallaceTree	12	24	1 652	4.677
传统 Wallace Tree	12	24	1 559	6.022

表 4 浮点乘法器性能对比

	主要功耗/W	最高运行频率/MHz
新型浮点乘法器	0.329	214
传统浮点乘法器	0.333	166

8 结束语

本文针对浮点运算过程中尾数相乘部分进行了优化,在传统的 4-2 压缩器和 5-2 压缩器的基础上进行了改进,并根据 32 位浮点运算过程中产生的 13 个部分积设计了基于混合压缩结构的 Wallace Tree,使浮点乘法处理过程的关键延时相比于传统浮点乘法器缩短 22.4%,所使用的 LUT 资源相较于文献[5]减少了 2.7%,相比于同类型的其他文献浮点乘法运算的效率更高。

参考文献:

[1] 夏伟,肖鹏.一种高效双精度浮点乘法器[J].计算机测量与控制,2013,21(4):1017-1020. DOI: 10.3969/j.issn.1671-4598.2013.04.060.
XIA W, XIAO P. An efficient double-precision floating-point multiplier[J]. Computer Measurement & Control, 2013, 21(4): 1017-1020. DOI: 10.3969/j.issn.1671-4598.2013.04.060.

- [2] 吴美琪, 赵宏亮, 刘兴辉, 等. 一种基于改进基 4 Booth 算法和 Wallace 树结构的乘法器设计[J]. 电子设计工程, 2019, 27(16): 145-150. DOI: 10.3969/j.issn.1674-6236.2019.16.032.
- WU M Q, ZHAO H L, LIU X H, et al. A design of multiplier based on modified Booth-4 algorithm and Wallace tree[J]. Electronic Design Engineering, 2019, 27(16): 145-150. DOI: 10.3969/j.issn.1674-6236.2019.16.032.
- [3] 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, 33(5): 13-16. DOI: 10.3969/j.issn.1000-386x.2016.05.004.
- SHI M, WANG G, YI Q M. An optimised design of multiplier based on improved booth encoding and Wallace tree[J]. Computer Applications and Software, 2016, 33(5): 13-16. DOI: 10.3969/j.issn.1000-386x.2016.05.004.
- [4] 朱鑫标, 施隆照. 一种高压压缩 Wallace 树的快速乘法器设计[J]. 微电子学与计算机, 2013, 30(2): 46-49. DOI: 10.19304/j.cnki.issn1000-7180.2013.02.012.
- ZHU XB, SHI LZ. A design of multiplier with high-compress Wallace tree [J]. Microelectronics & Computer, 2013, 30(2): 46-49. DOI: 10.19304/j.cnki.issn1000-7180.2013.02.012.
- [5] 刘容, 赵洪深, 李晓今. 基于改进型选择进位加法器的 32 位浮点乘法器设计[J]. 现代电子技术, 2013, 36(16): 133-136. DOI: 10.3969/j.issn.1004-373X.2013.16.040.
- LIU R, ZHAO H S, LI X J. Design of 32-bit floating-point multiplier based on improved carry-select adder [J]. Modern Electronics Technique, 2013, 36(16): 133-136. DOI: 10.3969/j.issn.1004-373X.2013.16.040.
- [6] 王佳乐, 胡越黎. 基于新型 booth 选择器和压缩器的乘法器设计[J]. 微电子学与计算机, 2020, 37(3): 5-8. DOI: 10.19304/j.cnki.issn1000-7180.2020.03.002.
- WANG J L, HU Y L. Design and implementation multiplier based on new booth selector and compressor[J]. Microelectronics & Computer, 2020, 37(3): 5-8. DOI: 10.19304/j.cnki.issn1000-7180.2020.03.002.
- [7] IEEE. IEEEStd754TM-2008. IEEE standard for floating-point arithmetic[S]. New York: IEEE, 2008.
- [8] 仲亚, 叶瑶瑶. 基于新型压缩器的乘法器设计[J]. 微电子学与计算机, 2019, 36(3): 28-31, 37. DOI: 10.19304/j.cnki.issn1000-7180.2019.03.006.
- ZHONG Y, YE Y Y. Design of multiplier based on new compressor [J]. Microelectronics & Computer, 2019, 36(3): 28-31. DOI: 10.19304/j.cnki.issn1000-7180.2019.03.006.

作者简介:

姚上上 男, (1996-), 硕士研究生. 研究方向为计算机体系结构. E-mail: 754169868@qq.com.

沈立 男, (1976-), 博士, 教授, 研究方向为计算机体系结构.