

基于优化电路的高性能乘法器设计

应琦钢, 郑丹丹, 何乐年

(浙江大学 超大规模集成电路研究所, 浙江 杭州 310027)

摘 要: 为了提高二进制乘法器的速度并降低其功耗, 在乘法器的部分积产生模块采用了改进的基 4Booth 编码和部分积产生电路并在部分积压缩模块应用了 7:3 压缩器电路, 设计并实现了一种高性能的 33×28 二进制乘法器. 在 TSMC 90 nm 工艺和 0.9 V 工作电压下, 仿真结果与 Synopsys 公司 module compiler 生成的乘法器相比, 部分积产生电路速度提高 34%, 7:3 压缩器和其他压缩器的结合使用减少了约一级全加器的延时, 整体乘法器速度提高约 17.7%.

关键词: Booth 编码; 部分积压缩; 7:3 压缩器

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2011)04-0052-05

High Quality Multiplier Design Based on Improved Circuits

YING Qi-gang, ZHENG Dan-dan, HE Le-nian

(Department of VLSI, Zhejiang University, Hangzhou 310027, China)

Abstract: In order to improve the speed and power consumption of the binary multiplier, this paper present a high-quality 33×28 binary multiplier with modified radix-4 Booth encoding and improved partial product circuits in partial product generator and a 7:3 compressor in partial product compression module. Compared with the multiplier created by Synopsys' module compiler, the speed of proposed partial product generator has improved 34%, the time of compressor has reduced about one 3:2 compressor's delay by combining 7:3 compressors with other compressors and the total delay of proposed multiplier has improved 17.7% at 0.9 V on a TSMC 90 nm process technology.

Key words: Booth encoder; partial products generate; 7:3 compressor

1 引言

对于嵌入式 CPU 来说, 乘法器是一个必不可少的功能部件, 其性能对 CPU 的影响很大. 如何提高乘法器的速度以及降低其功耗早已成为人们研究的热点方向^[1-3].

Booth 编码算法在乘法器中的应用, 减半了部分积个数, 却增加了部分积产生的延时^[1-2]. 7:3 压缩器虽然经过了数十年的研究^[3-5], 却还未广泛应用于乘法器的设计中. 本文基于 Ki-seon Cho^[2]文献中的 Booth 编码算法, 提出一种改进的部分积产生电路, 关键路径分析时考虑了驱动电路的延时影响. 在部分积压缩方面引入全定制设计的 7:3 压缩器, 设

计实现了一种 $33 \times 28-b$ 的并行乘法器结构.

2 乘法器结构

如图 1 所示, 基于 Booth 编码算法的 33×28 二进制乘法器主要包括三部分^[1-2]: (1) Booth 编码及部分积的产生模块; (2) 部分积压缩模块; (3) 最终加法器.

28 位乘数首先进行编码, 产生 14 组的选择控制信号. 再由这些控制信号对被乘数乘以 $\pm 2, \pm 1, 0$ 的操作, 以实现对被乘数的 4 进制运算. 14 组控制信号控制被乘数产生 14 组部分积. 然后利用全加器或其他压缩器对部分积矩阵进行压缩, 直至只剩下两个部分积矩阵. 将其输入至最终加法器, 最后输出

收稿日期: 2010-06-10; 修回日期: 2010-08-21

基金项目: 国家自然科学基金项目(60720106003); 浙江大学基本科研业务费专项项目(KYJD09012)

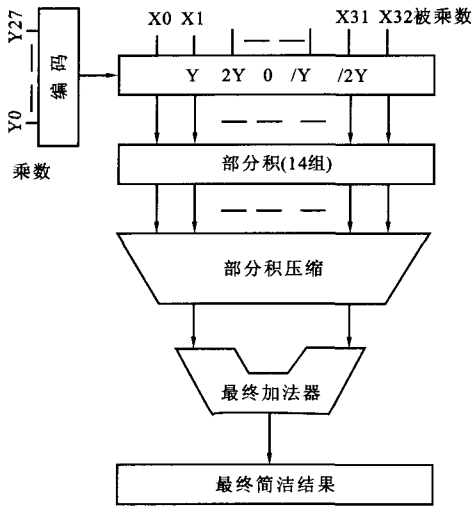


图 1 33×28 乘法器结构总框图

CMOS 电路中 P 管和 N 管同时导通时的瞬间功耗,这两个属于动态功耗。 P_{static} 为静态功耗,由漏电流引起.由式(2)可知,通过减少乘法器内部单元的 MOS 管尺寸可以减小内部负载、 I_{peak} ,在不采用特殊工艺和不改变总体算法的情况下,减少电路的面积可以简单且有效地减少乘法器的功耗.

乘法器的面积主要由部分积产生单元和压缩器单元决定,一组部分积只需要 1 个 Booth 编码单元,编码单元的总面积相比于部分积产生单元可以忽略,所以应重点优化部分积产生电路.

基于 Ki-seon Cho 提出的改进算法^[2],本文提出一种新的部分积产生电路,通过减短关键路径和全定制设计来提高速度,同时采用电路分级的方式来优化面积和功耗. Booth 编码表达式及改进算法^[2]分别如式(3)、(4)所示,其中 Y 为 3 位乘数, X 为 2 位被乘数.

Booth_{Out} =

$$\begin{cases} Y_{2i+1} \oplus X_j & Y_{2j} \oplus Y_{2i-1} = 1 \\ Y_{2i+1} \oplus X_{j-1} & Y_{2i} \oplus Y_{2i-1} = 0 \text{ 且 } Y_{2i+1} \oplus Y_{2i} = 1 \\ 0 & \text{else} \end{cases} \quad (3)$$

$$\begin{cases} F = Y_{2i+1} \\ N = Y_{2i+1} \oplus Y_{2i} \\ S = Y_{2i} \oplus Y_{2i-1} \end{cases} \quad (4)$$

N, S 分别是式(4)的两个条件, F 是式(3)的公共变量. 根据式(4)提出优化的部分积产生电路如图 2 所示.

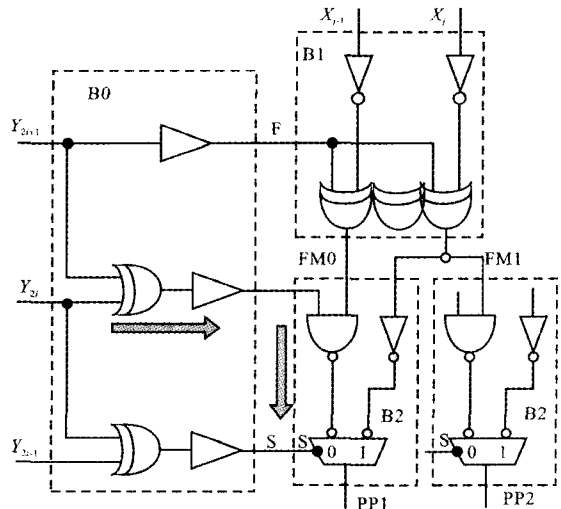


图 2 Booth 编码及部分积产生门级原理图

$B0$ 是 Booth 编码电路,产生 F, N, S 三个控制信号用于产生部分积; $B1$ 是第一级部分积产生电路; $B2$ 是第二级部分积产生电路. 一个编码电路产

乘法结果.

3 乘法器的设计

3.1 部分积产生模块的设计

3.1.1 基 4Booth 编码原理

设乘数为 Y ,基 4Booth 编码的算法原理^[6] 如式(1):

$$Y = \sum_{i=0}^N Y_i * 2^i = \sum_{j=0}^{(N-1)/2} Y_j * 4^j \quad (1)$$

式中, $Y_i \in \{0, 1\}$, $Y_j \in \{-2, -1, 0, 1, 2\}$. 可见,相比于二进制形式,采用四进制形式时部分积的个数至少可以减少为原来的一半. 改进后的基 4Booth 编码一次检查临近 3 位,根据结果对被乘数进行操作,而且编码需要乘数是奇数位,才能使 $(N-1)/2$ 为整数,如果是偶数,则需要在最后一位加一个 0^[7].

3.1.2 改进的 Booth 编码与部分积产生电路设计

虽然 Booth 编码会增加部分积产生模块的延时^[2],但可以减半部分积的个数,明显提高部分积压缩阶段的速度. 另外还有助于减少乘法器面积、简化版图^[7]等. 如果要想进一步提高乘法器的速度,则需要设计更高速的 Booth 编码和部分积产生电路.

乘法器的功率消耗^[4] 可归纳如下:

$$\begin{cases} P_{total} = P_{dynamic} + P_{static} = P_{cap} + P_{sc} + P_{static} \\ P_{cap} = a_{0 \rightarrow 1} f_{clk} C_L V_{DD}^2 \\ P_{sc} = a_{0 \rightarrow 1} f_{clk} I_{peak} \left(\frac{t_r + t_f}{2} \right) V_{DD} \\ P_{static} = I_{static} V_{DD} \end{cases} \quad (2)$$

式中, $a_{0 \rightarrow 1}$ 为单位时间的翻转概率, f_{clk} 为时钟频率. P_{cap} 为对负载电容充放电引起的功耗, P_{sc} 是

生一组部分积的控制信号,而每一个部分积都需要一个部分积产生电路.将部分积产生电路分为两级的主要目的是为了重复利用 FM0 和 FM1 信号以节省面积和功耗.

两级部分积产生电路的应用使得部分积产生单元的等效 MOS 管数为 24 个,而 Wen-Chang Yeh 的电路使用了 38 个 MOS 管, Ki-seon Cho 的电路使用了 34 个 MOS 管, Synopsys 公司 module compiler 产生的乘法器该电路使用了 28 个 MOS 管,所以理论上可以减少至少 14% 的面积,这样可以有效减少乘法器的面积和功耗.同时考虑控制信号驱动电路的影响后,该电路结构的关键路径如图 1 中箭头所示,相比于 Wen-Chang Yeh 和 Ki-seon Cho 提出的电路中的关键路径,约减少了 1.5 个反向器的延时,理论可以提高约 10% 的速度,而相比于 module compiler 产生的乘法器中的电路改进得更多,约减少了 2 个反向器 2 个传输门的延时,约有 30% 的改进.

3.2 部分积压缩模块的设计

虽然目前绝大部分乘法器都是利用全加器或 4:2 压缩器设计部分积压缩模块,但是人们一直在研究其他类型的压缩器^[3-5],特别是 7:3 压缩器,试图结合已有的压缩器来优化乘法器的性能.

本文将 Mayur Mehta^[6]等人提出的 7:3 压缩器应用于实际的乘法器设计,结合 4:2 压缩器、全加器等不同压缩器,以进一步提高乘法器的速度.

4:2 压缩器已经广泛应用于目前的乘法器设计中^[7],根据理论和实际 HSPICE 仿真,4:2 压缩器的速度约为全加器的 1.8 倍左右.

33×28-b 乘法器最长路径上有 14 个部分积串联,分析不同压缩器的组合,可以发现最快的组合级联方式如图 3 所示,相比于全加器级联的方式可以减少约 1 个全加器的延时.

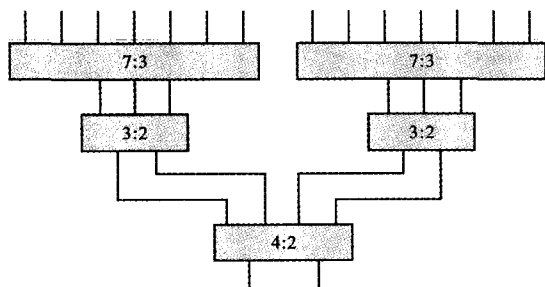


图 3 33×28 乘法器部分积压缩方式

3.3 33×28 乘法器的设计

本文采用图 2 中的电路作为 Booth 编码时的部

分积产生电路、7:3 压缩器电路^[6]并使用图 3 的压缩方式设计部分积压缩模块.另外,采用快速超前进位加法器设计最终加法器,与 Synopsys 公司 module compiler 生成的乘法器中使用的最终加法器一样,设计了一种 33×28-b 的乘法器.

4 仿真实验与分析

在 TSMC 90 nm 工艺下,对 3.3 小节设计的 33×28-b 乘法器进行 HSPICE 仿真实验,工作电压为 0.9 V.

4.1 部分积产生电路实验结果分析

为了比较不同编码和部分积产生电路的性能,本文使用文章中提到的不同实现电路在相同工艺条件下设计了相同功能的部分积产生模块.表 1 是以上模块的仿真数据,其中改进电路即指使用图 2 中的电路设计的部分积产生模块,原有电路为 module compiler 生成的乘法器中的电路设计的部分积产生模块, Yeh^[1]和 Cho^[2]是指使用文献[1-2]中的电路设计的部分积产生模块.

表 1 部分积产生电路数据对比

	原有电路	改进电路	Cho ^[2]	Yeh ^[1]
延时/ps	348	230	297	267
功耗/mW	0.351	0.191	0.293	0.307
面积 ^① /μm ²	82.79	57.935	99.693	101.088

注:①面积是通过相加所有 MOS 管的宽度再乘以 MOS 管长度所得.

从表 1 可以看出,改进电路相比于原有电路,在速度、面积和功耗三方面分别优化了 34%、30% 和 45.6%,相比于文献[1-2]电路尽管在速度上只有 10% 左右的优化,但在面积方面有接近 50% 的优化.因为实际所使用 MOS 管的宽长比更优化,所以在面积上的改进更优于 3.1.2 小节中的理论值,而在速度上的改进基本符合理论值.

可见, Synopsys 公司 module compiler 软件生成乘法器的编码和部分积产生电路面积相对适中,但是延时和功耗比较大.文献[1-2]中的编码和部分积产生电路是牺牲面积为代价来提高速度的.而本文提出部分积产生电路应用了两级部分积产生电路,优化了面积和功耗,并通过合理缩短关键路径长度和全定制设计减小了延时.

4.2 7:3 压缩器的实验结果分析

4.2.1 7:3 压缩器与全加器实验结果对比

使用 4 个全加器通过三级级联的方式可以实现

7 到 3 的压缩,本小节对使用全加器实现的 7:3 压缩器和专门的 7:3 压缩器进行了 HSPICE 仿真,表 2 是三者的仿真数据.其中 M Mehta 指文献[4]里的电路,改进电路是对 M Mehta^[4]全定制设计后的电路.

表 2 不同 7:3 压缩器的数据对比

	原有电路 ^①	改进电路	M Mehta ^[4]
延时/ps	367	285	328
功耗/mW	1.16E-01	7.91E-02	5.89E-02
面积 ^② /μm ²	17.84	12.57	8.98

注:①由 4 个全加器组成的 7:3 压缩器电路;②面积是通过相加所有 MOS 管的宽度再乘以 MOS 管长度所得.

从表 2 可知,使用专门的 7:3 压缩器电路在速度、功耗和面积各方面都优于由全加器组成的乘法器至少 10%.所以当部分积个数为 7 时,完全可以使用 7:3 压缩器代替三级全加器级联的电路.

改进电路比 M Mehta 提出的 7:3 压缩器速度有约 13%的优势,但在面积和功耗上劣于后者.但是,7:3 压缩器使用在关键路径上,所以其面积和功耗在整个部分积压缩模块上只占很小部分,而其速度却直接决定着整个模块的速度,所以速度是 7:3 压缩器的最高优先级.另外,不同 7:3 压缩器优势上的差异可以满足不同的设计要求,扩大了 7:3 压缩器的适用范围.

4.2.2 不同压缩器压缩效果对比

如图 4 所示,用不同的压缩器和不同的级联方式实现 10 个部分积的压缩.仿真结果如表 3 所示.

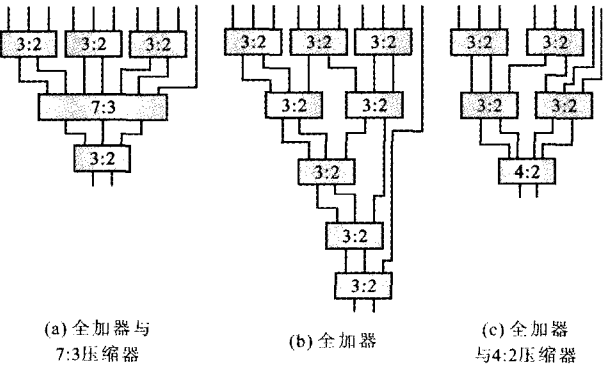


图 4 不同压缩器的压缩组合
表 3 不同压缩组合仿真数据对比

	(a)	(b)	(c)
延时/ps	503	640	550
功耗/mW	0.115	0.145	0.119
面积/μm ²	30.404	35.672	26.852

由表 3 可知,虽然在面积上(a)用 1 级 7:3 压缩器比(c)用 2 级 4:2 压缩器组合要大约 12%,但在速度上(c)比(a)要快约 9%,而功耗上(a)和(c)基本相同,所以可以根据实际设计要求选择不同的压缩器组合.可以看出,无论是速度、功耗还是面积,(a)和(c)这两种组合方式比(b)只用全加器实现都有明显优势.

从以上分析可知,在速度上 1 级 7:3 压缩器的改进程度相当于 2 级 4:2 压缩器的效果,甚至更好,这进一步扩大了 7:3 压缩器的适用场合,为乘法器部分积压缩模块的设计提供了新的选择.

4.3 部分积产生及压缩部分的仿真结果对比

为了测试本文提出的编码与部分积产生电路和 7:3 压缩器电路对整体乘法器的影响,分别对本文设计的 33×28-b 乘法器和用 Synopsys 公司的 module compiler 软件生成的相同功能的乘法器进行仿真比较.仿真波形图如图 5 所示.

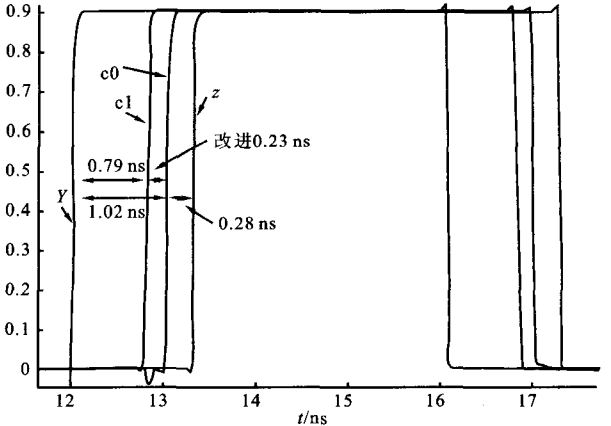


图 5 乘法器仿真结果

图 5 中 Y 为乘法器的输入乘数,Z 为 module compiler 乘法器的输出结果,c0 为 module compiler 乘法器部分积压缩完成后的信号,c1 为改进后乘法器部分积压缩完成后的信号.

因为只修改了部分积产生和部分积累加两个模块,最终加法器没有修改,所以最终加法器部分的延时不会改变.从图中可以看出,经过两部分电路的修改,关键路径上延时减少了约 0.23 ns,其中部分积产生模块约改了 0.9 ns,部分积压缩模块改进了约 0.14 ns,相当于一级全加器的延时,基本符合 3.2 小节中的理论值,整体乘法器关键路径延时为 1.3 ns,速度改进了约 17.7%.

5 结束语

文中提出了优化的部分积产生电路,应用了两

级电路,在速度、面积和功耗三方面都优于 Synopsys 公司 module compiler 软件生成乘法器的编码和部分积产生电路和文献[1-2]中的相关电路,基本上适用于所有基 4Booth 乘法器。同时采用了尚未普遍使用的 7:3 压缩器,结合 4:2 压缩器和全加器等压缩器,达到了较好的效果,并为乘法器部分积压模块的设计引入了新的选择元素。仿真结果表明,本文提出的 Booth 编码电路、部分积产生电路和 7:3 压缩器电路在提高乘法器速度的同时,也优化了面积和功耗,提升了乘法器的总体性能。

参考文献:

- [1] Yeh W C, Jen C W. High-speed booth encoded parallel multiplier design[J]. IEEE Transactions on Computers, 2000, 49(7): 692-701.
- [2] Cho K S, Park J O, Hong J S, et al. 54x54-bit radix-4 multiplier based on modified booth algorithm[C]// Proceedings of the 13th ACM Great Lakes symposium on VLSI, Washington, DC, USA; ACM, 2003(4): 233-236.
- [3] Swartzlander E E. Parallel Counters[J]. IEEE Transactions on Computers, 1973, 22(11): 1021-1024.
- [4] Mehta Mayur, Parmar Vijay. High-speed multiplier design using multi-input counter and compressor circuits[C]// Proceedings 10th Symposium on Computer Arithmetic. Grenoble, France: IEEE, 1991(6): 43-50.
- [5] Swartzlander E E. A review of large parallel counter designs[C]// Proceedings IEEE Computer society Annual Symposium on VLSI, 2004. Lafayette, Louisiana: IEEE, 2004(2): 89-98.
- [6] Rabaey J M, Chandrakasan A. Digital integrated circuit [M]. Beijing: Publishing House of Electronics Industry, 2004.
- [7] Huang Zhijun. High-level optimization techniques for low-power multiplier design [D]. California: University of California, 2003.

作者简介:

应琦钢 男, (1985-), 硕士研究生. 研究方向为超大规模集成电路设计.

郑丹丹 女, (1981-), 博士后. 研究方向为超大规模集成电路设计.

何乐年 男, (1962-), 博士生导师. 研究方向为超大规模集成电路设计.

(上接第 51 页)

难以准确表达实际地形的变化情况. 若对初始特征点进行去噪, 使得算法对噪声敏感度降低, 这样能更好地表达出等高线图的实际信息.

参考文献:

- [1] Park H, Lee J H. B-spline curve fitting based on adaptive curve refinement using dominant points[J]. Computer-Aided Design, 2007, 39(6): 439-451.
- [2] 周红梅, 王燕铭, 刘志刚, 等. 基于最少控制点的非均匀有理 B 样条曲线拟合[J]. 机械设计与制造, 2008(6): 73-77.
- [3] 狄效儒, 刘涛. 一种改进的 Bezier 曲线细分算法[J]. 微电子学与计算机, 2009(12): 128-131.
- [4] 邓飞, 周亚同. 基于曲面插值的剖面三维块状地质建模[J]. 微电子学与计算机, 2009(9): 173-176.
- [5] 施法中. 计算机辅助几何设计与非均匀有理 B 样条 [M]. 北京: 高等教育出版社, 2001.
- [6] Hyungjun Park, Lee JooHaeng. B-spline curve fitting based on adaptive curve Refinement using dominant point [J]. Computer-Aided Design, 2007(39): 439-451.
- [7] Li W, Xu S, Zhao G, et al. Adaptive knot placement in b-spline curve approximation[J]. Computer Aided Design, 2005, 37(8): 791-797.
- [8] Yong Xuea, Min Sunc, Ainai Mac. On the reconstruction of three-dimensional complex geological objects using Delaunay triangulation[J]. Future Generation Computer Systems special issue: Geocomputation archive, 2004, 20(7): 1227-1234.

作者简介:

原庆红 女, (1984-), 硕士研究生. 研究方向为虚拟现实、虚拟战场环境仿真中构建三维地形.