

一种改进的 CSA 低功耗阵列乘法器的实现

徐东明, 卢 斌

(西安邮电大学 通信与信息工程学院, 陕西 西安 710061)

摘 要: 以实现电能采集中所需求的低功耗、小面积的乘法器为目标, 设计了一种 16×16 位高性能阵列改进乘法器. 系统采用 Booth-4 编码器产生部分乘积项, 通过对部分积重组后并采用改进的 CSA 阵列完成压缩, 直接得出乘法结果. 这消除了传统并行乘法器的进位加法器部分, 节省了大量的晶体管, 从而有效降低了系统的整体功耗. 设计采用 $0.6 \mu\text{m}$ SMIC 工艺布线, 利用 H-spice 工具仿真验证, 结果表明当工作在 2.0 V 单输入电压, 150 MHz 输入频率时, 乘法器系统功耗为 8.98 mW , 延迟为 8.76 ns .

关键词: 乘法器; 低功耗; 改进的 CSA 阵列; 关键路径; Booth-4 算法

中图分类号: TP332.2

文献标识码: A

文章编号: 1000-7180(2016)09-0019-05

A Low Power Multiplier with Modified Carry Save Array

XU Dong-ming, LU Bin

(School of Communication and Information Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710061, China)

Abstract: The low power and small area of the multiplier is widely in the electrical energy acquisition, A High quality of 16×16 multiplier with Carry Save Array is designed. The modified Booth-4 algorithm is adopted to generate partial product and the design of modified CSA is introduced to compress the partial product, eliminating the propagate adder at the final stage of the conventional multipliers. Due to removal of a few transistors in the array architecture, the proposed multiplier with Carry Save Array reduces the power dissipation and the area. In the design, At the voltage of 2.0 V , the H-spice is used to carry out the results for $0.6 \mu\text{m}$ SMIC technology which reveals the proposed design has a measured power dissipation of 8.98 mW and that multiplication time of multiplier is 8.76 ns at a frequency of 150 MHz .

Key words: multiplier; low power consumption; critical path; dual logic; booth-4 algorithm

1 引言

在专用集成电路、数字滤波以及信号处理领域, 乘法器始终是系统不可或缺的运算单元, 它影响着整个系统的运算效率和稳定性. 随着近些年来科技产品正趋于智能化、微型化、便于携带等特点, 功耗已经成为乘法器设计中不可忽略的因素之一^[1]. 因此, 乘法器的低功耗设计越来越受到人们的普遍重视.

20 世纪 50 年代, 随着 Booth 算法^[2]的问世, 在

很大程度上提高了乘法器的性能, 随后, 各种针对 Booth 算法的改善也相继实现, 如 Booth-2, Booth-4, Booth-8, Booth-16 以及基于这些算法的改进. 综合考虑, Booth-4 整体性能最佳. 虽然 Booth 乘法器在速度上得到很大的提升, 其运算能力有所增强, 但是与传统的乘法器相比, 功耗随之也有所增长, 总之, Booth 乘法器是以牺牲功耗的代价获得速度上的提升, 然而如今的科技产品对功耗的要求越来越高, 因为功耗的增加不仅会减少移动设备的工作时间, 同时产生的热量会间接地缩短芯片的使用寿命

甚至损坏。

本次设计基于 CSA 全加器构建一种低功耗,低面积的阵列乘法器。采用改进的 Booth-4 编码算法产生部分积,根据文中的 CSA 阵列完成对部分积进行压缩直接得出结果,从而省略了传统乘法器最后的进位加法器。此外,基于高速、低功耗乘法器的设计,文中多处对电路进行了完善。

2 阵列乘法器的设计

2.1 乘法器设计结构

设计的低功耗系统的结构如图 1。主要包括(1)部分积的产生:根据改进的 Booth-4 算法设计出的编码器可以减少一半的部分乘积项;高位的符号扩展以及当对被乘数取反时低位的加 1 操作;(2)部分积的压缩和乘积结果的生成:对传统的 CSA 阵列重组,消除了传统意义上的最后的全加器,可以节省大量的晶体管。

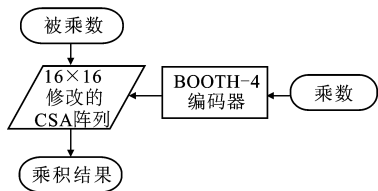


图 1 乘法器设计结构

2.2 部分积的产生

在高速的乘法器设计中,文献[3]提出几种在 Booth 算法上的改进并且进行了性能的比较,因此不同的改进所产生的系统性能是有差异的,在此系统设计中同样采用 Booth-4 算法,与传统的移位乘法器相比,不仅降低了部分乘积项的移位延时,同时使部分乘积项减少一半。Booth-4 算法的原理如下:设乘数为 Y ,

$$Y = -Y_{N-1}2^{N-1} + \sum_{i=0}^{N-2} Y_i 2^i$$

$$\sum_{i=0}^{2^{N-1}-1} (Y_{2i} + Y_{2i-1} - 2Y_{2i+1}) 2^{2i}$$

$$= \sum_{i=0}^{2^{N-1}-1} E_i 2^{2i} \quad (1)$$

E_i 的值取 $\{-2, -1, 0, 1, 2\}$ 。由式(1)可知 Booth-4 编码一次考虑相邻三位,处理两位,由于表中不同的值对乘数的操作不同,为了设计出编码器,根据文中的编码器分析设计如图 2 所示。

由上述编码单元组成的编码器根据被乘数和乘数的输入则会产生所需要的部分积。

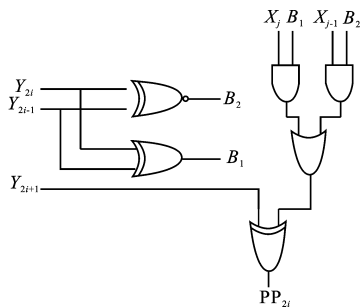


图 2 Booth-4 编码门级电路

2.3 部分积的压缩以及乘积结果的产生

在低功耗系统的设计中,不同阶段的功耗分布不同,其中部分乘积项的压缩工耗几乎占了整个系统的一半。因而必须优化部分乘积项的压缩,降低这个阶段的功耗就显得尤其重要。本部分不仅采用低功耗的全加器,而且对压缩方式也进行了改善。结果表明乘法器的功耗大幅度改善[4]。

在乘法器设计中通常用于减少部分积的方式主要通过 Dadda Tree[5]、Wallace Tree[6]或 CSA(Carry Save Adder)[7],部分积压缩产生的进位和本位值需要一个进位加法器处理进而得出结果。通常意义上虽然 Wallace Tree、Dadda Tree 提高了乘法器速度,但是功耗比较高,而 CSA 本身比较规则[8],形成的阵列版图非常紧凑,所需要的晶体管数目少,功耗低。因此在这个阶段采用 CSA 阵列并对其改进以此完成对部分积的压缩。

在本设计当中,为了实现 CSA 阵列的最佳性能,在以下两个方面进行改进:(1)逻辑级和晶体管级:作为 CSA 阵列的基本全加器单元,其时序的延迟、功耗的高低直接决定了系统性能的好坏,因此设计中在关键和非关键路径上采用两种不同结构的全加器,这样可以根据实际情况的不同对延时和功耗适当进行调整。(2)RTL 级:对传统的 CSA 阵列[9]进行重组,采用新的走线方式,不仅减少了系统中全加器的数量同时消除了传统意义上乘法器必不可少进位加法器。通过这次优化对乘法器的整体性能有很大的提升。

2.3.1 CSA 阵列的重组

乘法器传统的 CSA 阵列组成如下:在阵列的第一行一般使用全加器或者半加器。如果部分积的第一行为全加器,然后第三个输入 C_m 将会被赋予“0”,对半加器不需要 C_m 。每一排的全加器的进位作为第二排的斜对角线全加器的输入,产生的乘法器就叫做进位保留加法器,因为全加器产生的进位并没有

直接参与第二排正下方全加器运算而是保留作为斜对角线的全加器的输入,以此类推,在最后阶段,需要一个 CPA(Carry Propagate Adder)对 CSA 阵列产生的进位值和本位值进行加法运算得出结果。

以 8 位 * 8 位的乘法器为例,经过编码器产生的 4 个部分乘积项如图 3(a)部分积重组前。其中 P08,P09,P10,P18,P19,P28,P29,P38,P39 为每个部分积高位的符号扩展,C₀,C₁,C₂,C₃ 为每个部分积低位的加 1 操作数。为了使用更少的全加器运算单元,对 4 个部分积进行阵列的重组。部分积的重组方式如图 3(b)部分积重组后。

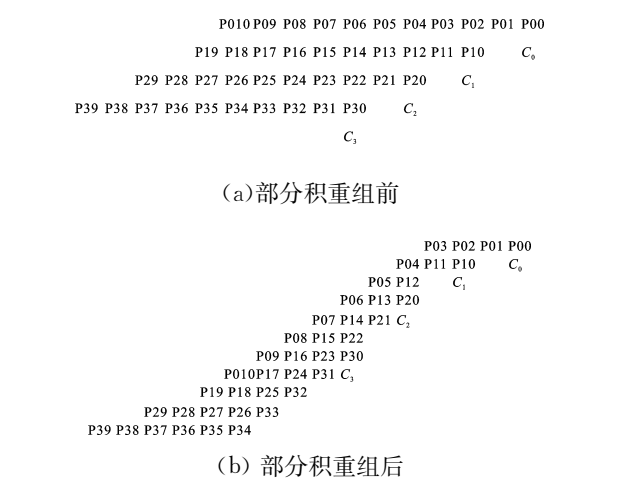


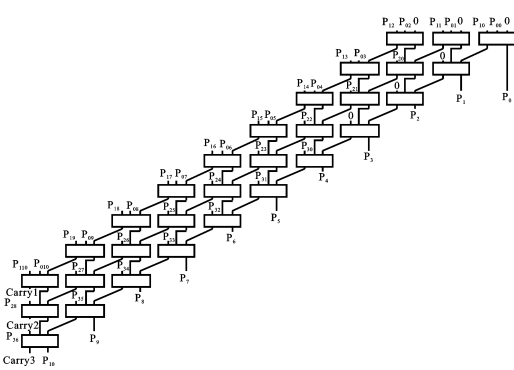
图 3 部分积的重组

具体其电路分为两个部分:并行重组部分和串行重组部分。

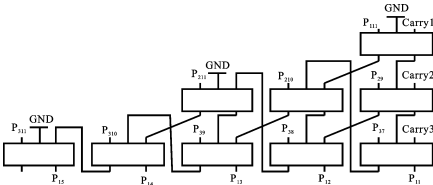
如图 4(a)所示,对于阵列的并行重组部分,每一行的第一个全加器相对右上斜对角线的全加器会有一个全加器单元的延时,但是会节省一个全加器单元。也就是说在并行重组的同时,以一定的时延换取全加器数量的减少,并且时延的多少与减小的全加器数量成正相关。

而在阵列的串行重组部分如图 4(b)所示,将每一列的最后阶段的进位作为下一列第一个全加器的输入。也就是图当中第一列(从右侧开始)最后阶段的进位作为第二列第一个全加器的输入,第二列最后阶段的进位作为第三列第一个全加器的输入,以此类推,第 5 列的全加器进位则作为最高位 MSB (Most Significant Bit)。同理,同并行重组部分一样,是以时延为代价换取全加器单元的减少。因此,在时延可以容忍的范围内,这种阵列改进方法可以大大降低乘法器的功耗。

此次设计的乘法器与传统的阵列乘法器、Wallace Tree 以及 Dadda Tree 的最大不同之处还在于,消除了最后阶段的加法器。与文献当中的进位保留



(a)并行重组的级联方式



(b)串行重组的级联方式

图 4 部分级重组级联方式

并行乘法器相比,无论在延时、面积还是功耗方面都有很大的提高。除此之外,与传统的阵列相比,设计出来的版图规则,利于布局布线,因此在面积方面也会有一定的改善。

对于 16 位 * 16 位的运算,按照上述的改进方式,会减少相当可观的晶体管,这无疑会使传统 CSA 阵列的功耗进一步降低。

2.3.2 CSA 级联方式

文献[10]提出不同的路径产生的性能是不同的,为了使 CSA 阵列进行压缩部分积所耗的时延最小,我们对 CSA 运算单元进行时延分析,如图 5 所示。

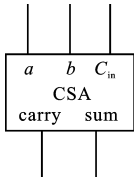


图 5 CSA 加法器单元

在 CSA 加法器当中,进行时延分析发现三个输入信号与输出信号之间的时延是不同的,设 T_{ij} 为输入信号 i 到输出信号 j 内部逻辑的延迟时间。则:

$$\begin{aligned} T_{a \rightarrow \text{carry}}; T_{b \rightarrow \text{carry}}; T_{C_{in} \rightarrow \text{carry}} &= 2:2:1 \\ T_{a \rightarrow \text{sum}}; T_{b \rightarrow \text{sum}}; T_{C_{in} \rightarrow \text{sum}} &= 2:2:1 \\ T_{a \rightarrow \text{carry}}; T_{a \rightarrow \text{sum}} &= 3:2 \end{aligned}$$

根据上述公式对传统的连接方式(如图 6(a)),

进行改进,如图 6(b)为调整后的连接方式,此种连接顺序更加理想,Carry 端和 Sum 端的延时相同且比较小,因此在压缩部分积时 CSA 阵列采取图 6 的连线方式以减少从 CSA 运算单元的关键路径到与门的延迟,提高速度. 因此,在设计当中,无论是关键路径还是非关键路径采用这种级联的方式作为降低 CSA 阵列时延的一种改进.

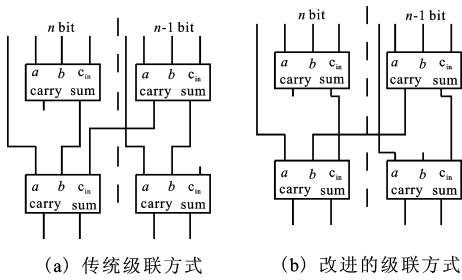


图 6 全加器的级联方式

2.3.3 关键路径上的全加器改进

在改进的阵列乘法器的关键路径(由阵列每行最后一个全加器单元组成),假设其一个 CSA 单元的输入分别为 A, B, C ,其中 A 为产生的部分积作为全加器的输入位, B 为上一层全加器的输出 Sum 位, C 为上一层前一列的进位,输出为 Carry, Sum. 则根据全加器的原理可得:

$$\text{Carry} = AB + BC + AC \tag{2}$$

$$\text{Sum} = A \oplus B \oplus C = (A + B + C) \overline{\text{Carry}} + ABC \tag{3}$$

定理:在改进的 CSA 阵列的关键路径上存在 $BC = ABC$.

证明:

(1) 当 X, Y 为 4 位时:

设 $X = X_3X_2X_1X_0, Y = Y_3Y_2Y_1Y_0$ 时,当 $X = 1111, Y = 1111$ 时, $Z = X * Y = 11100001$,当 $X = 1111, Y = 0111$ 时, $Z = X * Y = 01101001$.

(2) 当 X, Y 为 8 位时,

设 $X = X_7X_6X_5X_4X_3X_2X_1X_0, Y = Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$ 时,当 $X = 11111111, Y = 11111111$ 时, $Z = 1111111000000001$,当 $X = 11111111, Y = 01111111$ 时, $Z = 111111010000001$.
.....

(3) 当把二进制转化为相应的十进制:设 X, Y 分别为 N 位,则根据数学计算可得:

$$(2^N - 1) * (2^{M-1} - 1) = 2^{N+M-1} - 2^N - 2^M + 1 < 2^{N+M-1}$$

由此当最高位 $A = 0$ 时, $\text{Carry} = 0$,根据式(2),

可得当 $A = 0$ 时, $\text{Carry} = AB + BC + AC = BC = 0$,我们可以写成 $BC = ABC$,很显然,当 $A = 1$ 时, $ABC = BC$.

因此可以对式(5)、(6)进行化简如下:

$$\begin{aligned} \text{Carry} &= AB + BC + AC \\ &= AB + ABC + AC = A(B + C) \end{aligned} \tag{4}$$

$$\begin{aligned} \text{Sum} &= A \oplus B \oplus C = (A + B + C) \overline{\text{Carry}} + ABC \\ &= (A + B + C) \overline{\text{Carry}} + BC \end{aligned} \tag{5}$$

通过式(4)、(5)与式(2)、(3)的对比发现,经过化简后的全加器逻辑表达式更简单,对全加器的内部电路重新进行设计,如图 7 所示,不仅可以减少一部分晶体管,降低乘法器的功耗,而且也可以减少在关键路径上的延时,从而改善系统整体的运算时间,提高速度.

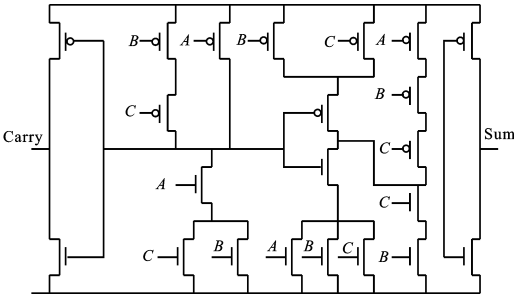


图 7 关键路径上的全加器电路

3 仿真结果和讨论

首先利用 Multisim 对其正确性进行验证,如图 8 所示.



图 8 采用改进的 CSA 阵列设计的乘法器仿真结果

通过仿真表明乘法器的功能设计正确,然后对其性能进行验证,通过 H-spice 工具对由逻辑全加器构建的 CSA 阵列进行电路仿真,仿真环境在标准电压 1.1 V,温度 25℃,乘法器的系统时钟为 150 MHz,对不同位数的 CSA 阵列的性能仿真,结果如表 1.

表 1 不同位数的 CSA 阵列性能

N 位	时延 /ns	平均功耗 /μW	功耗延时积 /(μW * ns)
4	0.442	106.7	49.9
8	0.452	204.9	101.8
16	0.468	423.7	184.7

在同样的条件下与其他全加器构建的改进的 CSA 阵列相比,其性能更加优越,节省的晶体管数

目非常客观,时延也有一定的改善.同时乘法器使用 0.6 μm SMIC 工艺,在系统时钟为 150 MHz,电压 2.0 V,系统功耗 8.98 mW,延迟为 8.76 ns.

4 结束语

本次设计提出一种改进的 CSA 阵列设计高速低功耗的乘法器的方案,该系统改进了用于产生部分积的 Booth-4 编码器的基本单元和用于完成部分积压缩的 CSA 阵列,消除了传统意义上的最终的加法器直接得出乘积结果.其中 CSA 运算单元采用新型的不同逻辑的结构混合设计,可以适当的调整关键路径和非关键路径延时和功耗的关系.由于 CSA 阵列的延时性会随着乘法器的位数而非线性增加,因此本设计仅适合位数低于 16 位的低功耗乘法器的设计.通过仿真结果表明,这种由新型全加器构建经过改善后的 CSA 阵列减少了相当可观的

参考文献:

[1] Kalaiselvi K, Mangalam H. Area efficient high speed and low power MAC unit[J]. International Journal of Computer Applications, 2013, 67(23):40-44.

[2] 王晓东, 富坤, 耿恒山, 等. 在 8 位微程序控制的模型计算机中 Booth 算法的实现[J]. 河北科技大学学报, 2013, 33(5):443-447.

[3] 徐东明. 实现快速乘法的几种改进贝斯算法[J]. 西安邮电大学学报, 2006, 11(1):61-65.

[4] 翟召岳, 韩志刚. 基于 Booth 算法的 32 位流水线型乘

法器设计[J]. 微电子学与计算机, 2014, 31(3): 146-149.

[5] 李路路, 何春, 宗竹林, 等. 一种基于 Dadda 树的乘法器设计[J]. 微电子学与计算机, 2011, 28(5): 176-179.

[6] 杜慧敏, 马超. 一种快速浮点乘法单元的设计与实现[J]. 西安邮电大学学报, 2013, 18(1):62-66.

[7] Cilaro A, De Caro D, Petra N, et al. High speed speculative multipliers based on speculative carry-save tree[J]. Circuits & Systems I Regular Papers IEEE Transactions on, 2014, 61(12):3426-3435.

[8] Seo Y H, Kim D W. A new VLSI architecture of parallel multiplier-accumulator based on radix-2 modified Boothalgorithm[J]. IEEE Transactions on Very Large Scale Integration Systems (VLSI), 2010, 18(2): 201-208.

[9] 朱世宇, 夏汝华, 甘科, 等. 基于 FPGA 的阵列乘法器的设计与实现[J]. 自动化与仪器仪表, 2011,30(4): 60-61.

[10] Ku T, Agrawal S, Giomi J C. Methods and apparatus for data path cluster optimization; US,8935651[P]. 2015-01-13.

作者简介:

徐东明 男,(1963-),教授,硕士生导师.研究方向为专用集成电路.

卢 斌(通讯作者) 男,(1990-),硕士研究生.研究方向为专用集成电路. E-mail:949649117@qq. com.

(上接第 18 页)

[8] Chang M H, Chiu Y T, Lai S L, et al. A 1kb 9T sub-threshold SRAM with bit-interleaving scheme in 65nm CMOS[C]//Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design. Italy,Rome,IEEE Press, 2011:291-296.

[9] Yabuuchi M. A 45nm 0.6V cross-point 8T SRAM with negative biased read/write assist[C]// VLSI Circuits, 2009 Symposium on. Japan, Kyoto, 2009: 158-159.

[10] Wang Jiajing, Nalam S, Calhoun B H. Analyzing static and dynamic write margin for nanometer SRAMs[M].

UK,Cardiff, Low Power Electronics and Design, International Symposium on ACM Press, 2008:129-134.

作者简介:

刘冰燕 男,(1992-),硕士研究生.研究方向为低功耗 SRAM 设计、低功耗数字电路设计.

E-mail:liubingyan@ince. ac. cn.

蔡江铮 男,(1989 -),博士研究生.研究方向为近亚阈值 SRAM 设计、超低功耗 SRAM 设计.

黑 勇 男,(1974 -),研究员,博士生导师.研究方向为大规模集成电路设计、低功耗数字集成电路设计.