

# 32 位定/浮点乘法器设计

于敦山 沈绪榜

(西安微电子技术研究所, 西安 710065)

**摘要:** 针对 Wallace 树连接线复杂度高, 版图实现比较困难的缺点, 提出了一种新的加法器阵列结构。这种结构在规则性和连接复杂度方面优于 ZM 树和 OS 树。同时提出一种新的 CLA 加法器结构以提高乘法器的性能。乘法器采用  $1.5\mu\text{m}$  CMOS 工艺实现, 完成一次定点与浮点乘法操作的时间分别是 56ns 和 76ns。

**关键词:** 乘法器; 浮点; 定点

**EEACC:** 1265B; 2570F; 5230

**中图分类号:** TN 702    **文献标识码:** A    **文章编号:** 0253-4177(2001)01-0091-05

## 1 引言

高性能乘法器是现代微处理器中的重要部件, 是实时图像处理和数字信号处理的核心。同时乘法器也是微处理器数据处理的关键路径, 乘法器完成一次乘法操作的周期基本上决定了微处理器的主频。因此人们对提高乘法器的性能进行了大量的研究。

为了便于乘法器的实现, 提高其版图的规则性和紧凑性, 人们对乘法器中完成部分积求和的加法器阵列进行了深入的研究。重复阵列(Iterative Array, 简称 IA)和 Wallace 树形结构<sup>[1]</sup>是加法器阵列最为典型的两种结构。IA 结构规整, 易于产生版图但速度最慢; Wallace 树结构速度最快, 但由于复杂的连接使其实现非常困难<sup>[2]</sup>。这两种结构都不适于乘法器的实现。为了解决这个问题, 人们提出了两种连接关系较为简单的树形结构: ZM 树<sup>[3]</sup>和 OS 树<sup>[4]</sup>。ZM 树(由 Zuras 和 McAllister 两人提出, 因此简称为 ZM 树)和 OS 树(Overturned-Stairs Trees)都是将 IA 树分为几段, 每段称之为子树。子树内部连接采用 IA 结构, 而子树间采用树形连接, 以此来降低连接复杂度。这两种结构都是在规则性与速度上进行折衷, 其规则性与速度也是在 IA 和

Wallace 树形结构之间。这两种结构有一个相同的缺点就是子树间的连接线数目虽少但很长, 这一方面增加了布线的复杂度, 另外还增加了延时。对于深亚微米工艺, 长连接线对提高乘法器性能是非常不利的。为此本文提出了一种连线关系较 ZM 和 OS 更为简单的树形结构。同时为了提高乘法速度, 乘法器设计还采用了一种新的 CLA 加法器结构。乘法器采用  $1.5\mu\text{m}$  CMOS 工艺实现。在最坏情况下, 完成一次定点与浮点乘法操作的时间分别是 56ns 和 76ns。

## 2 乘法器设计

### 2.1 树形加法器结构

由于 4-2 加法器构成的乘法器在总体性能上优于 3-2 加法器构成的乘法器<sup>[2]</sup>, 所以现在的乘法器在实现时均采用了 4-2 加法器作为树形加法器的基本单元。对于由 4-2 加法器构成的加法器阵列,  $n$  个 4-2 加法器能够完成  $2n+2$  个部分积的求和。也就是说, 无论采用哪种结构, 对于固定数目的部分积, 所需的加法器单元的个数是相同的, 关键在于采用什么样的结构使其连接关系更为简单。一个完成  $(20, 2)$  的 IA、一阶 ZM 树和一阶 OS 树的拓朴图分

于敦山 男, 1970 年出生, 博士生, 现在从事 CMOS 大规模集成电路的研究和开发。

沈绪榜 男, 中国科学院院士, 现在从事 VLSI 与微计算机设计与研究工作。

1999-09-28 收到, 1999-12-10 定稿

©2001 中国电子学会

别如图 1(a)、(c)、(d) 所示。图中符号 ‘○’ 表示一个 4-2 加法器，‘•’ 表示两个部分积，‘↓’ 表示两条连接线。图 1(c) 中的 ZM 树由高度分别为 2、2、3 的子树构成，子树内部采用 IA 结构。子树间由连接单元 ‘●’ 连接构成整个树形结构。单元 ‘●’ 也是一个 4-2 加法器。由连接单元连接在一起的两个子树我们称之为相邻子树。树的连接关系复杂度通常是用子树间的 TLW (Trans-Layer Wires, 跨跃单元的连接线数目) 来衡量<sup>[4]</sup>。参考文献[4] 采用 3-2 加法器构成整个树形结构，一阶 ZM 树和 OS 树的 TLW 分别是 2 和 3。TLW 基本上能够反映连接关系的复杂度。但由 4-2 加法器构成的一阶 ZM 树和 OS 树的 TLW 都是 2。因此，对于由 4-2 加法器构成的树，子树间的 TLW 已不能正确地说明连接线的复杂度。

在实现乘法器时，为了得到一个矩形的版图需要将加法单元按矩形排列，并通过线移位器使部分积之间权值相同的位对齐。这样 TLW 每跨过一个 ‘•’ 单元，就需要右移 4 位。(因为每个 ‘•’ 单元代表两个部分积，而采用二阶 Booth 方法产生的相邻的两个部分积对应各位的权值差 2<sup>2</sup>)。TLW 跨跃的 ‘•’ 单元数目越多，TLW 的长度就越长，连接关系就越复杂。因此连接复杂度既与 TLW 的数目有关，还与 TLW 的长度，也就是跨跃的 ‘•’ 单元的个数有关。由于 ‘○’ 单元数和 ‘•’ 单元数存在线性关系，因此 TLW 的长度也可以由跨过的 ‘○’ 单元数来表示。根据以上讨论，我们提出用整个树的 TLWT (Trans-Layer Wires of Tree) 来衡量连接复杂度。TLWT 的计算方法为：

$$TLWT = \sum_{i=1}^{n-1} TLW_i \times BC_i$$

式中  $n$  是树中子树的个数； $TLW_i$  是相邻子树之间连接线数目。以 4-2 加法器构成的树的  $TLW_i$  都为 2； $BC_i$  是相邻子树中 ‘○’ 单元数最小值。当子树内包含 ‘●’ 时也计之为 ‘○’ 单元。以图 1(d) 中的 OS 树的 TLWT 计算为例，各个 ‘↓’ 跨过的 ‘○’ 数如线旁标注。其 TLWT 为  $2(1+2+3)=12$ 。同样，图 1(c) 中 ZM 树的 TLWT 为 10。而一个完成(32, 2) 加法的 Wallace 如图 1(e) 所示，其 TLWT 是 34。从 TLWT 的数值来看，各种树的连线复杂性是显而易见的。ZM、OS 树较 Wallace 树在布线复杂度上已大大简化。

各种树结构的连线复杂度可以用图 2 中的二维图形表示。图中 X 轴表示树的宽度，Y 轴表示树的

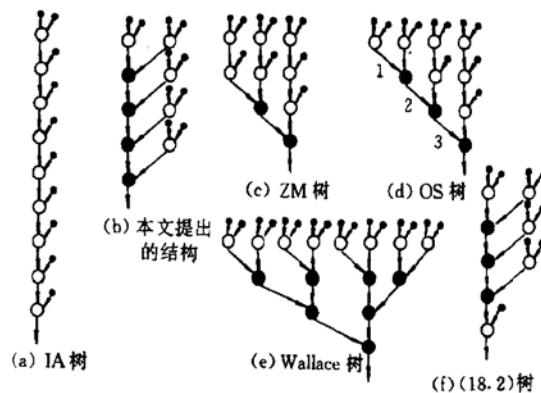


图 1 乘法器中 4-2 加法器的连接关系

FIG. 1 Connect Relation of 4-2 Adder in Multiplexer

复杂度。树越宽(表示子树越多)，布线复杂度就越高。树最宽、连接关系最复杂的是 Wallace 树。而 IA 结构可以看作是树宽为 1 的树，其连接关系也最简单。Wallace 树和 IA 树是树结构的两个极限，而 ZM 和 OS 树是这两种极限的中间结构。

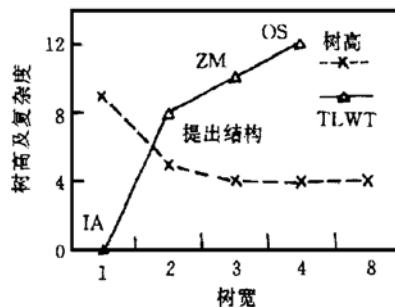


图 2 树复杂度与树宽的关系

FIG. 2 Complex-Level of Tree vs Its Width

要完成(20, 2) 加法，加法器树的宽度可以是(1, 2, 3, 4)。其中树宽分别为 1, 3, 4 结构的典型代表是 IA, 1 阶 ZM 树和 1 阶 OS 树。我们提出一种树宽是 2 的树，结构如图 1(b) 所示，其 TLWT 为 8，低于 ZM 和 OS 树。由于 32×32 的乘法器的部分积只有 18 个，因此可以将图 1(b) 再简化为图 1(f) 所示结构，其 TLWT 是 6，但树高为 5。也就是说这种树的树高高于一阶 ZM 树和 OS 树，但其规则性也优于 ZM 树和 OS 树。

## 2.2 CSA 加法器

高速加法器的结构有很多种，但在乘法器设计中一般都采用 CSA 加法器。通常的 CSA 加法器由两套 CLA 加法器分别计算低位进位为 0' 和为 1' 的加法结果，再由多路器根据实际进位选择最终的

结果。为了减小芯片面积, 参考文献[5]提出一种以 $(G, P)$ 为传递信号的 CLA 加法器。与通常的以进位信号 $C$ 为传递信号的 CLA 加法器相比, 这种加法

$$\begin{aligned} G_4 &= g_4 + p_4(g_3 + p_3(g_2 + p_2(g_1 + p_1G_0))) \\ G_3 &= g_3 + p_3(g_2 + p_2(g_1 + p_1G_0)) \\ G_2 &= g_2 + p_2(g_1 + p_1G_0) \\ G_1 &= g_1 + p_1G_0 \end{aligned}$$

其中  $g_i = P_{a_i} \cdot P_{b_i}$ ,  $p_i = P_{a_i} \oplus P_{b_i}$ ,  $(G_0, P_0)$  是前一组向本组的传递信号,  $(G_4, P_4)$  是向下一组的传递信

号。这种结构在关键路径上等效逻辑结构如图 3(a)所示:

$$\begin{aligned} P_4 &= p_4p_3p_2p_1P_0 \\ P_3 &= p_3p_2p_1P_0 \\ P_2 &= p_2p_1P_0 \\ P_1 &= p_1P_0 \end{aligned}$$

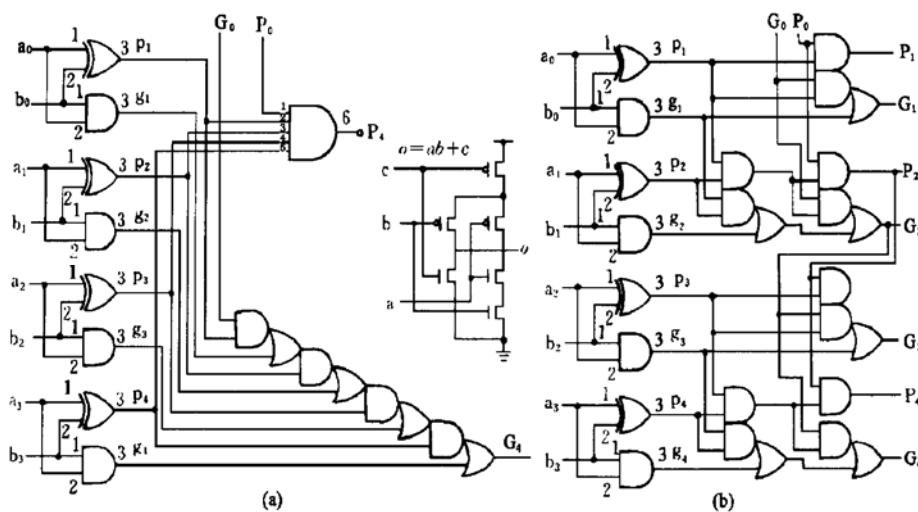


图 3 以 $(G, P)$ 为传递信号的 4 位 CLA 加法器 (a) 参考文献[5]中 CLA 关键路径的等效结构, (b) 本文提出的 CLA 结构。

FIG. 3 4 Bit CLA Adder with Transfer Signal ( $G, P$ )

为了进一步提高这种加法器的速度, 我们提出一种先行进位加法器与树形加法器相结合的结构, 也就在 4 位组内进行 2-2 分块的电路, 逻辑表达式为:

$$\begin{aligned} G_{4:3} &= g_4 + p_4g_3 & P_{4:3} &= p_4p_3 \\ G_{2:1} &= g_2 + p_2g_1 & P_{2:1} &= p_2p_1 \\ G_1 &= g_1 + p_1G_0 & P_1 &= p_1P_0 \\ G_2 &= G_{2:1} + P_{2:1}G_0 & P_2 &= P_{2:1}P_0 \\ G_3 &= g_3 + p_3G_2 & P_3 &= p_3P_2 \\ G_4 &= G_{4:3} + P_{4:3}G_2 & P_4 &= P_{4:3}P_2 \end{aligned}$$

其电路如图 3(b) 所示。在进位链上, 参考文献[5]中的 CLA 有四级串连的 AND-OR 等效结构, 而提出的结构只有两级。AND-OR 结构可由一个传输管构成的二选一电路代替<sup>[6]</sup>, 这样在进位链上只有两级串联的传输管延时。鉴于生产工艺上的原因, 在实际设计中我们没有采用这种电路, 而是采用传统的

CMOS 门电路。

我们对参考文献[5]中的加法器与提出的加法器进行了 SPICE 模拟。采用的 MOS 管尺寸为: NMOS:  $L_n = 1.5\mu\text{m}$ ,  $W_n = 3.5\mu\text{m}$ ; PMOS:  $L_p = 1.75\mu\text{m}$ ,  $W_p = 7\mu\text{m}$ 。同时我们也对  $W_p = 3.5\mu\text{m}$  的情况进行了模拟。模拟时激励信号是经反相器加到加法器电路以模拟实际工作情况下的输入信号。模拟结果如图 4 所示(所有输出负载电容为  $0.2\text{pF}$ )。由模拟结果可知, 提出结构与参考文献[5]中的加法器相比在性能上可提高 30%。

我们分别对采用参考文献[5]中的 CLA 和本文提出的 CLA 构成的 64 位 CSA 加法器进行了模拟, 结果如图 5 所示。图中  $P_a$  和  $P_b$  是乘法阵列产生的伪和与进位信号;  $P'$  和  $P$  是分别采用参考文献[5]及本文提出的 CLA 加法器时 CSA 各位得出时间。相对于  $P'$ ,  $P$  的计算速度提高了 21.5%。由于

$P_a, P_b$  各位产生时间不同使我们提出的先行进位电路构成的 64 位加法器没有达到提高 30% 的水平.

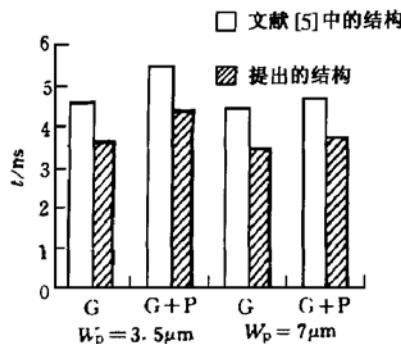


图 4 参考文献[5]中的结构与提出结构的性能比较

FIG. 4 Performance Comparison of Two Structures

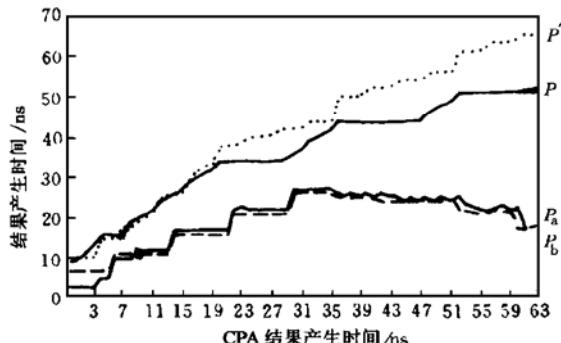


图 5 加法器阵列结果产生与 CPA 结果产生时间

FIG. 5 Forming Time of CPA's Results vs One of Array Result in Adder

### 3 乘法器性能分析

乘法器采用  $1.5\mu m$  双层铝线 CMOS 工艺设计以便于在国内生产. 其版图如图 6 所示, 图中左下角方块是指数处理部分, 右边是乘法阵列, 上面是 64 位 CSA 加法器、规格化及结果输出部分. 由版图可以看出, 树形加法器阵列的结构规整, 除了全局性的总线信号外, 其它信号均是局部信号. 乘法器的规模为 47700 管, 面积为  $5.9mm \times 3.5mm$ .

乘法器版图设计完成后, 用 Cadence 系统中的 Dracula 工具对版图进行参数提取并进行后仿真. 乘法器的测试图形很多, 我们选用最坏情况下的测试图形对乘法器进行性能测试. 对定点数乘法, 最坏情况下的输入数据<sup>[6]</sup>为:

$h0000\ 0001 \times hFFFF\ FFFF$  ——关键图形, 将所有输出置为 1

这个输入图形使 Booth 译码器只产生一个(-1)的部分积. 在这种情况下, 只有一个部分积为全“1”数据而其它部分积为全“0”. 由于阵列中含有异或门使得阵列的每一个节点都处于活动状态, 并且最后的加法器 CPA 的输入为  $(-1) + 1$ , 使得这 64 位加法器的每一位都有进位生. 因此这组输入数据是最坏的情况.

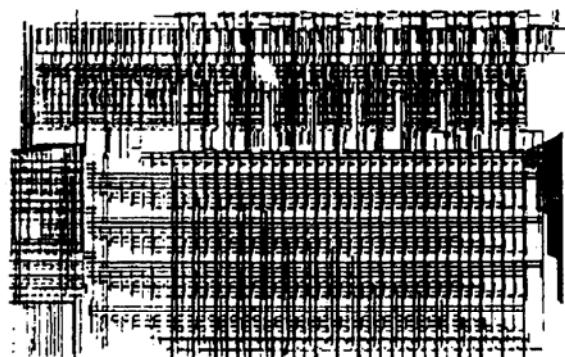


图 6 乘法器的版图

FIG. 6 Diagram of Layout for Multiplexer

对于浮点运算, 当一个操作数是反规格化数并且 48 位乘积的高 24 位都是符号位时的输入数据是关键图形. 我们选择的测试数据为:

$h4FFF\ FFFF \times 0000\ 0001$  ——关键图形. 规格化数  $\times$  反规格化数, 结果是规格化数.

通过模拟, 在最坏延时情况下, 完成上述定点与浮点乘法的时间分别为 56ns 和 76ns. 浮点乘法比定点乘法慢主要是由于浮点乘法需要进行操作数类型判断、尾数规格化及舍入和指数再调整操作.

### 4 结论

本文提出的树形结构在规则性和连接线复杂度方面优于现有 ZM 树和 OS 树结构, 使乘法器的版图设计更易于实现. 提出的先行加法器电路可以使乘法器中的 64 位加法器性能提高 21%. 乘法器的性能可进一步通过进行优化逻辑, 如采用 DPL 或 CPL 传输管逻辑来提高. 本文设计的定浮点乘法器也可用于其它微处理器设计中.

## 参考文献

- [1] C. S. Wallace, IEEE Trans. Electron. Comput., 1964, **EC-13**(2): 14—17.
- [2] Norio Ohkubo and Makoto Suzuki, IEEE J. Solid-State Circuits, 1995, **30**(5): 251—256.
- [3] D. Zuras and W. H. McAllister, IEEE J. Solid-State Circuits,

1986, **SC-21**(5): 814—819.

- [4] Z-J Mou and F. Jutand, 1990 IEEE International Conference on Computer Design: VLSI in Computers and Processors, IEEE Comput. Soc. Press, 1990, 251—254.
- [5] A. Tyagi, IEEE Trans. Comput., 1993, **42**(10): 1163—1170.
- [6] J. Mori and Masato Magamatsu, IEEE J. Solid-State Circuits, 1991, **26**(4): 600—606.

## Design of a 32-Bit CMOS Fix Floating Point Multiplier

YU Dun-shan and SHEN Xu-bang

(Xi'an Institute of Microelectronic Technology, Xi'an 710065, China)

**Abstract:** Wallace tree multipliers are very difficult to implement due to their complex routing requirement. A novel tree structure is presented, which requires simpler wiring than ZM trees and OS trees, and a novel CLA adder with 30% faster than the conventional one is proposed too to enhance the speed performance. The multiplier is fabricated with 1.5 $\mu$ m CMOS technology and can perform a 32-bit floating point multiplication (based on the proposed IEEE P754 standard format) and a 32-bit fixed point multiplication in 56ns and 76ns, respectively.

**Key words:** multiplier; floating point; fixed point

**EEACC:** 1265B; 2570F; 5230

**Article ID:** 0253-4177(2001)01-0091-05

YU Dun-shan was born on April 18, 1970. He received the M. S. and Ph. D. degrees in Xi'an Microelectronics Institute in 1994 and 1999 respectively. Now he is engaged in the research and development of CMOS LSI's.

SHEN Xu-bang, academician of CAS, is engaged in the research and development of microcomputer and CMOS LSI's.

Received 28 September 1999, revised manuscript received 10 December 1999

©2001 The Chinese Institute of Electronics