

高性能并行全冗余十进制乘法器的设计

张柳, 崔晓平, 董文雯

(南京航空航天大学电子信息工程学院, 江苏南京 211106)

摘要: 商业计算、金融分析等领域对高精度计算的需求对硬件十进制运算提出了越来越高的要求. 已有的全冗余十进制乘法器由于全冗余加法器的结构复杂, 已经给其性能的提升造成了瓶颈. 本文优化设计了基于超载十进制数集 (Overloaded Decimal Digit Set, ODDS) 的全冗余 ODDS 加法器以降低其复杂度, 并设计了一种新的基于该加法器的十进制压缩树模块. 本文在部分积产生模块采用有符号的基-10 编码和冗余的二进制 (Binary Coded Decimal, BCD) 编码快速产生十进制部分积. 在最终积产生模块采用优化的编码转换电路快速产生 BCD-8421 乘积. 实验结果显示所设计的并行全冗余十进制乘法器速度较快、面积较小.

关键词: 乘法器; 十进制运算; BCD 编码; 冗余编码; 全冗余加法器; 编码转换

中图分类号: TN492 **文献标识码:** A **文章编号:** 0372-2112 (2018)06-1519-05

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2018.06.036

High-Performance Parallel Fully Redundant Decimal Multiplier

ZHANG Liu, CUI Xiao-ping, DONG Wen-wen

(College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 211106, China)

Abstract: High-performance decimal hardware arithmetic is now a high demand due to the requirement for accurate computation in fields like commercial computing and financial analysis. The performance of fully redundant decimal multiplier is limited because the circuit for fully redundant adder is complex. A modified fully redundant adder based on overloaded decimal digit set (ODDS) and a new decimal reduction tree based on fully redundant ODDS adders are proposed. The signed-digit radix-10 recoding and redundant binary coded decimal (BCD) codes are used for fast partial product generation. A recoding conversion circuit is proposed to generate BCD-8421 product fast. Comparison shows that the delay and area of the proposed decimal multiplier are small.

Key words: multiplier; decimal arithmetic; BCD coding; redundant coding; fully redundant adder; recoding conversion

1 引言

二进制与十进制的转换存在误差和舍入, 因此不能满足商业计算、金融分析等领域对十进制算术运算的精度要求. 2008 年发行的 IEEE 754 标准的修订版^[1] 加入了十进制浮点 (Decimal Floating-Point, DFP) 的格式. IBM 的 Z 系列微处理器^[2]、富士通 Sparc64 X 微处理器^[3] 等已经包括了完全符合新标准的硬件 DFP 运算单元. 十进制乘法器作为 DFP 乘加运算单元的重要组成部分, 其性能直接影响 DFP 运算单元的性能^[4-6].

十进制乘法运算包括十进制部分积产生 (Partial Product Generation, PPG)、部分积压缩 (Partial Product Reduction, PPR) 和最终积产生三个模块. 十进制被乘数

X 和乘数 Y 采用 BCD 编码表示, 十进制乘法器的被乘数倍数的取值范围为 $\{0X, \dots, 9X\}$, 因此产生难度较大. 有符号的 (Signed-Digit, SD) 基-10 编码、基-5 编码和双 BCD 编码可以缩小被乘数倍数的范围^[7-14]. 基-10 编码将被乘数倍数的范围从 $\{0X, \dots, 9X\}$ 缩小为 $\{-5X, \dots, 0X, \dots, 5X\}$ ^[9,10,12], 对于 $d \times d$ -digit (1-digit 表示 1 个十进制位, 用 4-bit 二进制数表示) 十进制乘法运算, 通过基-10 编码产生 $(d+1)$ 行部分积 (Partial Product, PP).

非冗余和冗余十进制数格式^[7-9,12] 被广泛应用于十进制乘法器的设计中. 基于非冗余 SD 基-10 编码的 PPG 电路存在一个明显的弊端, 即难倍数 $3X$ 无法采用无进位的方式产生, 需将 X 与 $2X$ 通过十进制加法器相

收稿日期: 2017-04-25; 修回日期: 2017-07-06; 责任编辑: 覃怀银

基金项目: 国家自然科学基金 (No. 61404087); 航空科学基金重点实验室类 (No. 20152052025); 南京航空航天大学研究生创新基地 (实验室) 开放基金 (No. kfjj20160407); 中央高校基本科研业务费专项资金 (No. NS2015045)

加获得,增加了 PPG 电路的延时. 冗余 BCD 编码的应用解决了难倍数的问题. 文献[12]采用冗余三(Excess-3, XS-3)编码([-3,12])和 ODDS 编码([0,15])产生十进制 PP. XS-3 编码因其自补性被用于 PP 的产生,通过预计算所有的 +3 项,可以将 XS-3 编码的 PP 直接转换成 ODDS 部分积,转换后的 ODDS 部分积可以通过二进制压缩树进行压缩并对压缩过程中产生的十进制进位(又称为字进位)进行修正.

文献[11]提出了一种基于全冗余 ODDS 加法器的十进制压缩树结构. 全冗余十进制乘法器(Fully Redundant Decimal Multiplier, FRDM)的 PPR 模块采用全冗余十进制加法. 全冗余 ODDS 加法器压缩比为 2:1,相比已有的十进制 3:2 压缩器^[8,9],其压缩效率更高,因其计算规则遵循十进制数的进位规则,无需设计修正方块对字进位进行计数与修正,因此电路结构规整. 不足的是目前全冗余 ODDS 加法器的电路结构复杂,难于实现.

本文设计的 FRDM 采用文献[7]的 PPG 电路产生 $(d+1)$ 行 ODDS 部分积,采用优化设计的全冗余 ODDS 加法器和 ODDS-BCD 编码转换电路提高 FRDM 的性能. 实验结果显示,与目前国内外性能最佳的十进制乘法器相比较,所提出的 FRDM 的综合性能优越.

2 全冗余 ODDS 加法器

全冗余 ODDS 加法器的结构如图 1 所示^[11]. 设 a_i ($a_i^3 a_i^2 a_i^1 a_i^0$) 和 b_i ($b_i^3 b_i^2 b_i^1 b_i^0$) 是全冗余 ODDS 加法器的两个第 i 字 ODDS 操作数. s_i ($s_i = a_i + b_i$) 是第 i 字十进制加法运算的和. 为了快速产生向高位的进位,可以将两个操作数 a_i 和 b_i 拆分为左拆分项 l_i 和右拆分项 r_i ,使得 $\|s_i\| = \|l_i\| + \|r_i\|$ ($\|u\|$ 代表操作数 u 的算术值). l_i 可用于产生字进位 t_{i+1} 以及内部和 z_i ,使得 $\|l_i\| = \|z_i\| + 10t_{i+1}$. 当 l_i 和 r_i 所包含的位数分别为 5 和 3 时,产生字进位的速度较快^[11].

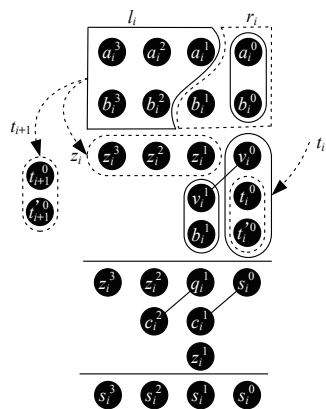


图1 全冗余ODDS加法器整体结构图

由图 1 可知,左拆分项满足 $0 \leq \|l_i\| \leq 26$, 因此所产生的字进位 t_{i+1} 满足 $0 \leq t_{i+1} \leq 2$, 将 t_{i+1} 表示为 $t_{i+1} = t_{i+1}^0 + t_{i+1}^1$. 字进位 t_{i+1} 被传递至第 $i+1$ 字,用于第 $i+1$ 字十进制全冗余加法的计算. 内部和 z_i 、右拆分项 r_i 和第 $i-1$ 字的进位构成了第 i 字的十进制和 s_i ($s_i^3 s_i^2 s_i^1 s_i^0$), t_{i+1} 和 z_i 的逻辑图如图 2 所示^[11].

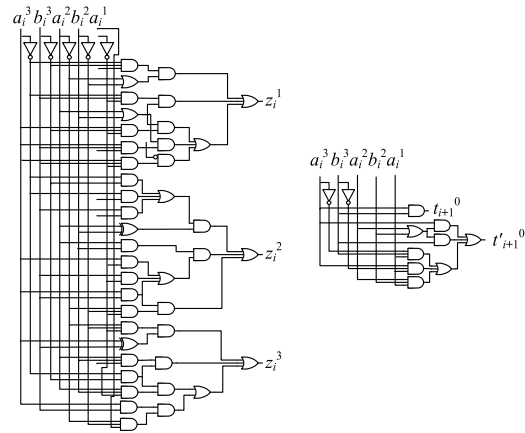


图2 t_{i+1} 和 z_i 的逻辑图

文献[11]的 FRDM 经双 BCD 编码产生 32 行 BCD-8421 PP,由 PPR 模块产生一行 32-digit 的 ODDS 乘积,通过 32-digit 的 ODDS-BCD 编码转换电路获得 BCD-8421 乘积. 设 s_i 为第 i 字的 ODDS 乘积, P_i 为第 i 字的 BCD-8421 乘积. 当 $s_i > 9$ 时,向第 $i+1$ 字传递进位 c_i^{out} , P_i 的值由该位 s_i 的值和第 $i-1$ 字进位输入 c_i^{in} 决定. 根据文献[11],ODDS-BCD 编码转换公式为:

$$P_i^0 = c_i^{in} \oplus s_i^0 \tag{1}$$

$$P_i^1 = c_i^{in} [(s_i^1 \oplus s_i^0) \overline{s_i^3} + s_i^1 + s_i^0 \overline{s_i^3} s_i^2 + s_i^3 s_i^1 s_i^0] + c_i^{in} (s_i^3 s_i^2 \overline{s_i^1} + \overline{s_i^3} s_i^1) \tag{2}$$

$$P_i^2 = c_i^{in} \{ (s_i^3 s_i^2 s_i^1 + s_i^3 s_i^2 s_i^0) + [(s_i^3 \overline{s_i^2}) (s_i^1 s_i^0)] + s_i^2 \overline{s_i^1} s_i^0 \} + c_i^{in} (s_i^3 s_i^2 + s_i^2 s_i^1) \tag{3}$$

$$P_i^3 = c_i^{in} (\overline{s_i^3} s_i^2) (s_i^1 s_i^0) + c_i^{in} (s_i^3 \overline{s_i^2} \overline{s_i^1}) + [(s_i^3 \overline{s_i^2}) (\overline{s_i^1} s_i^0)] \tag{4}$$

$$c_i^{out} = c_i^{in} \overline{s_i^3} s_i^0 + s_i^3 s_i^2 + s_i^3 s_i^1 \tag{5}$$

为了提高编码转换电路的速度,运用超前进位单元快速产生进位. 进位产生信号 g_i 和进位传递信号 p_i 由式(6)获得^[11]:

$$g_i = s_i^3 (s_i^2 + s_i^1), p_i = s_i^3 s_i^0 \tag{6}$$

3 全冗余十进制乘法器的优化设计

采用 SD 基-10 编码, XS-3 编码和 ODDS 编码快速产生 17 行 ODDS 部分积,将优化设计的全冗余 ODDS 加法器应用于 FRDM 的 PPR 模块中,将优化设计的 32-digit ODDS-BCD 编码转换电路用于 BCD-8421 乘积的产生. 图 3 给出了基于全冗余 ODDS 加法器的 17:1 十进

制 PPR 树的结构. 全冗余 ODDS 加法器的压缩比为 2:1, 最高列共包括 5 级全冗余 ODDS 加法器的压缩, 5 级压缩比分别为 17:9、9:5、5:3、3:2 和 2:1.

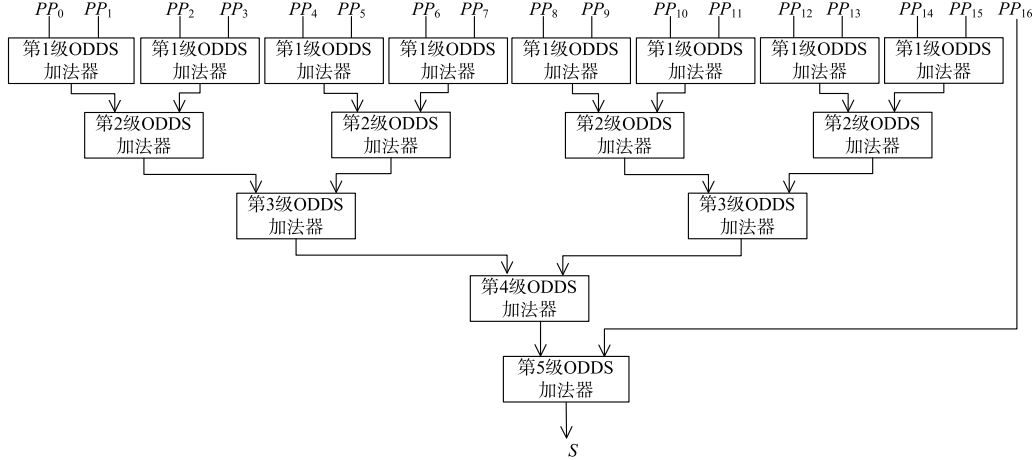


图3 基于全冗余ODDS加法器的17:1十进制PPR树结构图

将十进制加法的两个操作数 a_i 和 b_i 进行如图 1 所示的拆分, 对 t_{i+1} 、 z_i 的电路结构进行优化设计. 由于全冗余 ODDS 加法器的电路结构十分复杂, 所以优化设计的首要任务是减少电路的复杂度. 优化设计的全冗余 ODDS 加法器的字进位 t_{i+1} 以及内部和 z_i 的电路结构如图 4 所示. 通过快速产生进位信号 t'_{i+1} 来提高全冗余 ODDS 加法器的速度. 通过 2:1 数据选择器, 2 输入与非门、2 输入或非门和 2 输入异或门减小电路的延时和面积^[15], 同时复用相同的异或门、与(与非)门和或(或非)门以减小电路面积. 优化后的逻辑表达式为:

$$t_{i+1}^0 = a_i^3 b_i^3 \quad (7)$$

$$t'_{i+1}^0 = (a_i^3 + b_i^3 + a_i^2 + b_i^2) (a_i^3 \oplus b_i^3) a_i^1 (a_i^2 b_i^2) a_i^1 \quad (8)$$

$$z_i^3 = (a_i^3 \oplus b_i^3) [a_i^2 + b_i^2 a_i^1 + (a_i^2 b_i^2) a_i^1] + a_i^1 [(a_i^3 + b_i^3) (a_i^2 b_i^2)] + a_i^1 [(a_i^2 + b_i^2) (a_i^3 b_i^3)] \quad (9)$$

$$z_i^2 = a_i^3 + b_i^3 (a_i^2 \oplus b_i^2) + (a_i^3 + b_i^3) [(a_i^2 b_i^2) a_i^1] + a_i^3 b_i^3 [a_i^1 (a_i^2 \oplus b_i^2)] + a_i^3 b_i^3 (a_i^2 b_i^2 + a_i^2 + b_i^2 a_i^1) \quad (10)$$

$$z_i^1 = a_i^1 [a_i^2 + b_i^2 (a_i^3 b_i^3) + (a_i^2 + b_i^2) (a_i^3 \oplus b_i^3)] + a_i^1 [a_i^3 + b_i^3 a_i^2 b_i^2 + (a_i^2 + b_i^2) (a_i^3 b_i^3)] \quad (11)$$

对 ODDS-BCD 编码转换电路进行优化设计以减小电路的复杂度, 优化设计的逻辑表达式为:

$$P_i^0 = c_i^{\text{in}} \oplus s_i^0 \quad (12)$$

$$P_i^1 = c_i^{\text{in}} [s_i^3 s_i^1 + s_i^3 (s_i^2 s_i^1)] + c_i^{\text{in}} \{s_i^3 (s_i^1 \oplus s_i^0) + s_i^3 [(s_i^1 + s_i^0) (s_i^2 + s_i^1 + s_i^0)]\} \quad (13)$$

$$P_i^2 = c_i^{\text{in}} [s_i^2 + (s_i^3 s_i^1)] + c_i^{\text{in}} \{s_i^2 (s_i^3 s_i^1 s_i^0) + s_i^2 [(s_i^3 + s_i^1) (s_i^3 + s_i^0) (s_i^1 + s_i^0)]\} \quad (14)$$

$$P_i^3 = c_i^{\text{in}} (s_i^3 s_i^2 s_i^1) + c_i^{\text{in}} [s_i^2 (s_i^3 s_i^1 + s_i^0) + s_i^2 (s_i^3 s_i^1 s_i^0)] \quad (15)$$

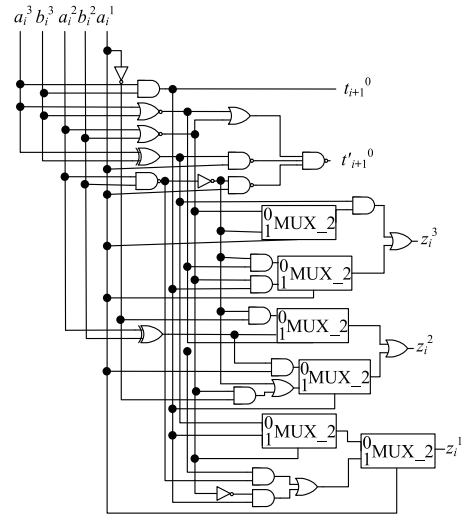


图4 优化设计的 t_{i+1} 和 z_i 产生电路

ODDS-BCD 编码转换时, 当第 i 字的 ODDS 部分积大于 9 时, 产生一个传递至第 $i+1$ 字的进位. 快速产生进位信号对编码转换的性能至关重要, 为了提高进位信号的产生速度, 采用基于二进制 Kogge-Stone (KS) 树^[10,16,17] 的前缀运算单元, 并将进位产生信号 g_i 和进位传递信号 p_i 分别用 $\overline{g_i}$ 和 $\overline{p_i}$ 取代, $\overline{g_i}$ 由一个或与或非门产生, $\overline{p_i}$ 由一个 2 输入与非门产生. 32-digit ODDS-BCD 编码转换电路需进行 5 级前缀运算, 其中奇数级采用或与或非和 2 输入或非门, 偶数级采用或与或非门和 2 输入与非门.

4 实验结果

所设计的 16×16 -digit 全冗余十进制乘法器采用 SD 基-10 编码, 冗余 XS-3 编码和 ODDS 编码快速产生 17 行 ODDS 部分积阵列, 优化设计了全冗余 ODDS 加法器的电路结构并将其运用于 17:1 十进制压缩模块的

设计中,对 32-digit ODDS-BCD 编码转换电路的结构进行优化设计以进一步提升所设计的 FRDM 的性能。

用 Verilog HDL 硬件描述语言对优化设计的 16×16 -digit FRDM 和文献[7~9,11]的十进制乘法器进行描述和功能验证.在 Nangate Open Cell 45nm CMOS 标准工艺库下,在 1.25V 电压和室温条件下,通过 Synopsys 公司综合工具 Design Compiler 进行综合,获得延时和面积.采用 Synopsys 公司综合工具 Power Compiler 获得功耗.将优化设计的全冗余 ODDS 加法器以及 32-digit ODDS-BCD 编码转换电路的延时、面积和功耗与文献[11]中相应的设计进行比较,比较结果如表 1 所示.

表 1 全冗余 ODDS 加法器和 32-digit ODDS-BCD 编码转换电路的延时、面积、功耗

	分类	延时(ns)	面积(μm^2)	功耗(μW)
全冗余 ODDS 加法器	文献[11]	0.54	116.51	13.83
	本文	0.50	86.18	9.52
32-digit ODDS-BCD 编码转换	文献[11]	0.79	2110.71	258.13
	本文	0.67	1674.20	167.97

由表 1 可知,本文所设计的全冗余 ODDS 加法器与文献[11]所设计的全冗余 ODDS 加法器相比,延时减小了 7.4%,面积减小了 26.0%,功耗减小了 31.2%.本文的 32-digit ODDS-BCD 编码转换电路与文献[11]相比,延时减小 15.2%,面积减小 20.7%,功耗降低了 34.9%.优化设计的全冗余 ODDS 加法器和 32-digit ODDS-BCD 编码转换电路的性能得到有效的提高.

表 2 给出了 16×16 -digit 非冗余十进制乘法器^[9]、我们原先的冗余十进制乘法器^[7,8]、FRDM^[11]和本文的 FRDM 的延时、面积、功耗.与文献[9]的非冗余十进制乘法器结构相比较,本文提出的 16×16 -digit 十进制乘法器的延时降低了 17.6%,面积减小了 9.2%,功耗降低了 9.2%.与文献[8]的冗余十进制乘法器相比,延时降低了 6.2%,面积增大了 1.9%,功耗增大了 1.4%.与文献[11]的 FRDM 相比较,以延时增加 0.6%为代价,面积减少了 32.1%,功耗降低了 29.9%.与我们原先的高性能冗余十进制乘法器[7]相比,延时略有降低,面积增加了 3.89%.

表 2 16×16 -digit 十进制乘法器的延时、面积、功耗

	延时(ns)	面积(μm^2)	功耗(mW)
文献[9]	3.86	48326.1	5.56
文献[11]	3.16	64617.5	7.20
文献[8]	3.39	43053.5	4.98
文献[7]	3.21	42217.9	4.84
本文	3.18	43858.6	5.05

5 结束语

本文所设计的 16×16 -digit 并行全冗余十进制乘法器可以快速有效地完成十进制乘法运算.文章优化了全冗余 ODDS 加法器的结构并将其运用于 17:1 十进制 PPR 模块的设计中.在 ODDS-BCD 编码转换电路的优化设计中,采用基于 KS 树的并行前缀结构快速产生编码转换电路的十进制进位,得到 BCD-8421 最终乘积.从实验结果看,本文所设计的 FRDM 与目前最具有代表性的高性能十进制乘法器相比较,其综合性能优越.

参考文献

- [1] IEEE Std 754 (TM)-2008. IEEE Standard for Floating-Point Arithmetic[S].
- [2] CARLOUGH S, COLLURA A, MUELLER S, et al. The IBM zEnterprise-196 decimal floating point accelerator [A]. Proceedings of 20th IEEE Symposium on Computer Arithmetic[C]. Tübingen: IEEE, 2011. 139-146.
- [3] YOSHIDA T, MARUYAMA T, AKIZUKI Y, et al. Sparc64 X: Fujitsu's new-generation 16-core processor for unix servers[J]. IEEE Micro, 2013, 33(6): 16-24.
- [4] WAHBA A A, FAHMY H A H. Area efficient and fast combined binary/decimal floating point fused multiply add unit[J]. IEEE Transactions on Computers, 2017, 66(2): 226-239.
- [5] HAN L, ZHANG H, KO S B. Decimal floating-point fused multiply-add with redundant internal encodings [J]. IET Computers & Digital Techniques, 2016, 10(4): 147-156.
- [6] GORGIN S, JABERIPUR G. Sign-magnitude encoding for efficient VLSI realization of decimal multiplication [J]. IEEE Transactions on Very Large Scale Integer (VLSI) Systems, 2017, 25(1): 75-85.
- [7] CUI X P, DONG W W, LIU W Q, et al. High performance parallel decimal multipliers using hybrid BCD codes [J]. IEEE Transactions on Computers, 2017, PP(99): 1-1.
- [8] CUI X P, LIU W Q, DONG W W, LOMBARDI F. A parallel decimal multiplier using hybrid binary coded decimal (BCD) codes [A]. Proceedings of the 23rd Symposium on Computer Arithmetic [C]. Santa Clara, CA: IEEE, 2016. 150-155.
- [9] VAZQUEZ A, ANTELO E, MONTUSCHI P. Improved design of high-performance parallel decimal multipliers [J]. IEEE Transactions on Computers, 2010, 59(5): 679-693.
- [10] JABERIPUR G, KAIVANI A. Improving the speed of parallel decimal multiplication [J]. IEEE Transactions on Computers, 2009, 58(11): 1539-1552.
- [11] GORGIN S, JABERIPUR G. A fully redundant decimal

- adder and its application in parallel decimal multipliers [J]. *Microelectronics Journal*, 2009, 40 (10): 1471 - 1481.
- [12] VAZQUEZ A, ANTELO E, BRUGUERA J. Fast radix-10 multiplication using redundant BCD codes [J]. *IEEE Transactions on Computers*, 2014, 63(8): 1902 - 1914.
- [13] KENNEY R D, SCHULTE M J. High-speed multioperand decimal adders [J]. *IEEE Transactions on Computers*, 2005, 54(8): 953 - 963.
- [14] ERLE M A, SCHWARZ E M, SCHULTE M J. Decimal multiplication with efficient partial product generation [A]. *Proceedings of 17th IEEE Symposium on Computer Arithmetic* [C]. Cape Cod, MA: IEEE, 2005. 21 - 28.
- [15] YEH W, Jen C. High-speed Booth encoded parallel multiplier design [J]. *IEEE Transactions on Computers*, 2000, 49(7): 692 - 701.
- [16] KOGGE P M and STONE H S. A parallel algorithm for the efficient solution of a general class of recurrence equations [J]. *IEEE Transactions on Computers*, 1973, C-22 (8): 786 - 793.
- [17] DIMITRAKOPOULOS G and NIKOLOS D. High-speed parallel-prefix VLSI Ling adders [J]. *IEEE Transactions on Computers*, 2005, 54(2): 225 - 231.

作者简介



张 柳 女, 1992 年 7 月出生于江苏南通. 现为南京航空航天大学研究生. 主要研究方向为电路与系统.

E-mail: zhangliunuaa@gmail.com



崔晓平 (通信作者) 女, 1962 年 1 月出生于安徽巢湖. 现为南京航空航天大学副教授、硕士生导师. 主要研究方向为数字集成电路设计和计算机算术运算系统.

E-mail: wnhexp@nuaa.edu.cn