

条件推测性十进制加法器的优化设计

崔晓平* 王书敏 刘伟强 董文雯
(南京航空航天大学电子信息工程学院 南京 210016)

摘要: 随着商业计算和金融分析等高精度计算应用领域的高速发展, 提供硬件支持十进制算术运算变得越来越重要, 新的 IEEE 754-2008 浮点运算标准也添加了十进制算术运算规范。该文采用目前最佳的条件推测性算法设计十进制加法电路, 给出了基于并行前缀/进位选择结构的条件推测性十进制加法器的设计过程, 并通过并行前缀单元对十进制进位选择加法器进行优化设计。采用 Verilog HDL 对 32 bit, 64 bit 和 128 bit 十进制加法器进行描述并在 ModelSim 平台上进行了仿真验证, 在 Nangate Open Cell 45nm 标准工艺库下, 通过 Synopsys 公司综合工具 Design Compiler 进行了综合。与现有的条件推测性十进制加法器相比较, 综合结果显示该文所提出的十进制加法器可以提升 12.3% 的速度性能。

关键词: 十进制加法; 条件推测十进制加法; 并行前缀; 进位选择加法器

中图分类号: TN431.2

文献标识码: A

文章编号: 1009-5896(2016)10-2689-06

DOI: 10.11999/JEIT151416

Design of Optimized Conditional Speculative Decimal Adders

CUI Xiaoping WANG Shumin LIU Weiqiang DONG Wenwen

(College of Electronic and Information Engineering, Nanjing University of
Aeronautics & Astronautics, Nanjing 210016, China)

Abstract: There are increasing interests in hardware support for decimal arithmetic due to the demand of high accuracy computation in commercial computing, financial analysis, and other applications. New specifications for decimal floating-point arithmetic have been added to the revised IEEE 754-2008 standard. In this paper, the algorithm and architecture of decimal addition is studied comprehensively. A decimal adder is designed by using the parallel-prefix/carry-select architecture. The parallel-prefix unit is used to optimize the decimal carry select adder. The decimal adder has been realized by Verilog HDL and simulated with ModelSim. The synthesis results of this design by Design Compiler is also given and analyzed under Nangate Open Cell 45nm library. The results show that the delay performance of the proposed circuit can be improved by up to 12.3%.

Key words: Decimal addition; Conditional speculative decimal addition; Parallel prefix; Carry select adder

1 引言

提供硬件支持十进制浮点(Decimal Floating Point, DFP)算术运算正在成为一个热门的研究方向, 2008年发行的 IEEE 754 标准的修订版本(IEEE 754-2008)^[1]包括 DFP 算术运算的最新规范。越来越多的处理器制造商倾向于在自己的处理器芯片中集成专用的十进制浮点运算单元, IBM 面向工作站和服务器的 Power 6^[2]微处理器以及 Z10 大型机^[3]的处理器中已经包括了完全符合 IEEE754-2008 标准的十进制浮点运算硬件单元。在处理器中提供专用的十进制运算单元将成为趋势。

十进制算术运算中最基础的十进制加法一直是

研究的热点, 目前的十进制加法运算基本上采用 8421 二-十进制编码(Binary Coded Decimal, BCD)^[1-9], 采用 8421-BCD 码设计十进制加法器的优势在于可以利用二进制加法器中成熟且性能优越的电路结构来设计十进制加法器, 使其电路结构更为简单与规整。不论在二进制加法^[10-18]还是十进制加法中, 影响加法电路运算速度的主要因素在于低位向高位传播的进位链。二进制加法和 BCD 十进制加法的不同点是: (1) 二进制加法的进位规则是逢二进一, 其进位的产生与传递比较简单, 而十进制加法运算需要计算十进制数之间的进位, 其进位规则是逢十进一。(2) 4 bit 编码的 8421-BCD 共有 16 种状态, 其中 6 种编码(1010, 1011, 1100, 1101, 1110, 1111)是误码, 因此, 当采用二进制运算方法对 4 bit 8421-BCD 进行相加运算时, 需要对二进制运算结果进行修正。

为了提高 8421-BCD 码十进制加法的性能, 研究人员提出了多种算法与结构, 其中最经典的两种方法是直接十进制加法^[5]和推测性十进制加法^[7]。直接十进制加法是一种无需进行修正的十进制加法算法, 该算法推导出直接产生十进制和与十进制进位的方法, 在 IBM S/360 Model 195 机型的处理器中使用该算法完成浮点运算^[4]。推测性十进制加法采用预先修正, 二进制求和并再修正的算法, 这种采用预先修正的十进制加法, 称之为推测性十进制加法。文献[7-9]根据此思路提出了条件推测性十进制加法, 即有条件地对操作数进行预先加 6 修正。

条件推测性十进制加法器主要包括+6 预处理模块、二进制并行前缀加法器模块和十进制进位选择加法器模块。二进制并行前缀加法器(Parallel Prefix Adder, PPA)可以看成是超前进位加法器的一种改进结构, 其常见的结构包括 Kogge-Stone(KS)树^[18]、Brent-Kung(BK)树^[14]、Sklansky(SK)树^[15]、Han-Carlson(HC)树^[16]等基本树形结构和并行前缀/进位选择混合加法器(Hybrid Parallel-Prefix/Carry-Select Adder, PPF/CSA)结构。并行前缀/进位选择混合加法器被广泛应用于宽位加法器的设计中^[12,17,18]。

文献[7-9]使用 QT (Quaternary Tree, QT)树形结构^[12]产生进位信号, 该结构与基于 SK 的 PPF/CSL 加法器结构相同, 进位选择加法器模块的长度为 4, 适用于设计 4 位一组的 BCD 十进制加法器。SK 树形结构随着操作数位数的增大, 其最大扇出数呈线性增长, 导致延迟时间增大。KS 并行前缀加法器具有最短的延时, 且结构规整并具有相同的扇出因子, 但不足之处是复杂度随着操作数位数增加, 因此导致面积和功耗的增大, 采用基于 KS 的 PPF/CSA 加法器结构可以得到高速的十进制加法器。本文重点研究基于 KS 的 PPF/CSL 十进制定点加法器的算法与相关结构, 并在第 3 节针对条件推测性十进制加法器给出优化设计方法以降低电路的复杂度。

本文结构如下: 第 2 节介绍了基于 8421-BCD 码的十进制加法; 第 3 节给出了新的基于 KS 结构的条件推测性十进制加法器的设计; 第 4 节给出了仿真结果并与现有的二进制和十进制加法器进行了对比分析。

2 基于 8421-BCD 码的十进制加法概述

在设计 N bit($N=4d$)十进制加法器时, 采用 8421-BCD 码对两个位宽为 d 的十进制被加数 A 和加数 B 进行编码, 具体形式为

$$A = \sum_{i=0}^{d-1} A_i \cdot 10^i,$$

$$A_i = \sum_{j=0}^3 a_i[j] \cdot 2^j, \quad 0 \leq i \leq (N/4-1), 0 \leq j \leq 3 \quad (1)$$

其中, $A_i = \{a_i[3], a_i[2], a_i[1], a_i[0]\} \in [0, 9]$ 是十进制操作数 A 的第 i 个十进制位, $a_i[j] \in [0, 1]$ 是 A_i 的 8421-BCD 编码的第 j 位。操作数 B 的编码方式与 A 同理。

基于 8421-BCD 码的十进制加法的基本算法是: 首先对十进制被加数和加数按二进制加法进行运算, 再对运算结果进行纠错。产生错误的原因是十进制数相加的进位原则是“逢十进一”, 而 4 bit 二进制数相加采用“逢十六进一”的进位原则, 两者相差 6。因此, 按二进制数运算规则得到的 8421-BCD 码运算结果需要修正。修正的方法是当和数大于 9 或产生进位时, 需要对该位的和加 6 修正。上述算法的最大缺陷是修正时的进位链会导致延时增加。研究人员提出了几种改进方法, 主要有直接十进制加法^[5], 推测性十进制加法以及条件推测性加法^[7-9]。

1 位直接十进制加法的输入为 8421-BCD 码的十进制被加数 A_i , 加数 B_i 以及一个 1 bit 的十进制进位输入信号 c_i , 直接产生十进制和 S_i , 以及一个 1 bit 的十进制进位输出 c_{i+1} , c_{i+1} 的位权是 S_i 的 10 倍, 其表达式为

$$(c_{i+1}, S_i) = A_i + B_i + c_i \quad (2)$$

推测性十进制加法对操作数 A 的每一个十进制位 A_i 先加 6, 然后对 $A_i + 6, B_i$ 按照二进制的方法进行求和, 如果该十进制位的进位输出为 0, 则说明加 6 操作是多余的, 进行减 6 修正, 其结构如图 1 所示。

文献[7-9]依据此思路提出有条件的推测性加法算法, 该算法没有对操作数 A 的所有十进制位 A_i 加 6, 而是根据一定条件判断是否需要某个十进制位进行加 6 预操作, 称之为条件推测性十进制加法, 条件推测性十进制加法结构如图 2 所示。

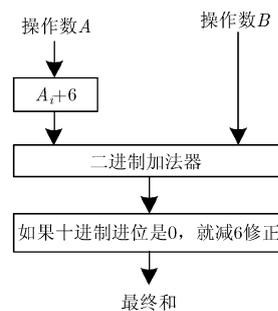


图1 推测性十进制加法结构图

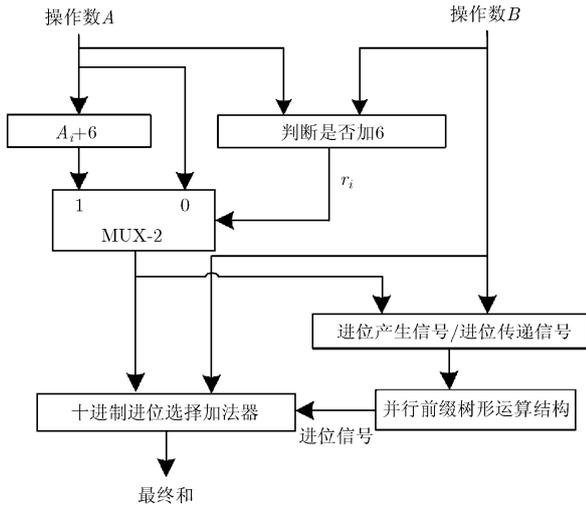


图 2 条件推测性十进制加法结构图

对于 1 位十进制数 A_i ，定义其 8421-BCD 编码的高 3 bit 为 A_i^U 。同时定义 $c_i[1], c_i[2], c_i[3]$ 分别为 8421-BCD 编码的第 0 位、第 1 位和第 2 位的进位输出信号。 c_i 为相应位的十进制进位输入信号， $(S_i^*)^U$ 为和的高 3 bit，那么产生十进制进位的条件是

$$S_i^* = A_i + B_i + c_i \geq 10 \quad (3)$$

则 $S_i^* - 1 \leq (S_i^*)^U \leq S_i^*$ ，其中 $(S_i^*)^U$ 为偶数。因此产生十进制进位的条件可以表示为

$$(S_i^*)^U = A_i^U + B_i^U + c_i[1] \cdot 2 \geq 10 \quad (4)$$

由此可以得到：

$$A_i^U + B_i^U \geq 10 - c_i[1] \cdot 2, \quad c_i[1] \in [0, 1] \quad (5)$$

所以产生十进制进位的必要非充分条件是 $A_i^U + B_i^U \geq 8$ 。若定义 8421-BCD 码的每一位的进位产生信号是 $g_i[j](g_i[j] = a_i[j] \cdot b_i[j])$ ，每一位的进位传递信号是 $p_i[j](p_i[j] = a_i[j] + b_i[j])$ 。则由条件 $A_i^U + B_i^U \geq 8$ ，可以得到判断是否对操作数 A_i 加 6 的公式为

$$r_i = p_i[3] + g_i[2] + p_i[2] \cdot g_i[1] \quad (6)$$

如图 2 所示，当 $A_i^U + B_i^U \geq 8$ ，也就是+6 控制模块的输出 $r_i = 1$ 时，对操作数 A_i 进行加 6 预操作，否则不加 6。当 $A_i^U + B_i^U = 8$ 且 $c_i[1] = 0$ 时，预加 6 操作是错误的，十进制和需要减 6 修正，修正工作在十进制进位选择加法器模块中完成。定义 A_i^* 为经过加 6 预处理电路之后的操作数， $A_i^* + B_i$ 的半加和为 $h_i^*[j] = a_i^*[j] \oplus b[j]$ 。由 $A_i^U + B_i^U = 8, c_i[1] = 0$ 得到需要减 6 修正的条件是 $h_i^*[3] = h_i^*[2] = h_i^*[1] = 1, c_i[1] = 0$ 。

图 2 的 $A_i + 6$ 操作模块的电路图如图 3 所示。进位产生/进位传递信号模块产生进位产生信号

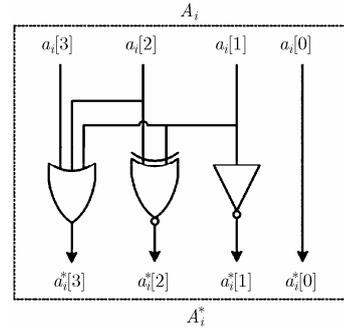


图 3 加 6 操作电路图

$g_i^*[j](g_i^*[j] = a_i^*[j] \cdot b_i[j])$ 和进位传递信号 $p_i^*[j](p_i^*[j] = a_i^*[j] + b_i[j])$ 。二进制并行前缀树形运算模块采用二进制运算方法产生十进制进位信号。

3 建议的基于 KS 结构的条件推测性十进制加法器的设计

在设计条件推测性十进制加法器时，完成加 6 预操作之后十进制进位和二进制相应位的进位信号一致，因此在二进制加法器设计中广泛采用的并行前缀/进位选择结构可以用于十进制加法器的设计。文献[7-9]采用基于 SK 的 QT 树形结构产生进位信号。典型的 16 bit SK 结构如图 4 所示，SK 树的逻辑级数最小，为 $\log_2 n$ ，运算结点只有 $(n \log_2 n) / 2$ 个。但是 SK 树形结构随着操作数位数的增大，其最大扇出数呈线性增长，导致延迟时间增大。为了获得高性能的十进制加法器，采用基于 KS 的并行前缀/进位选择加法器结构设计 32 bit, 64 bit 和 128 bit 十进制加法器，并对 4 bit 十进制进位选择加法器进行优化设计。典型的 16 bit KS 结构如图 5 所示。

为了减少加法器的复杂度，本文将利用并行前缀单元对文献[7]中的十进制进位选择单元进行改进。令 $w_i = h_i^*[3] \cdot h_i^*[2] \cdot h_i^*[1]$ ， c_i 为相应的十进制进位输出信号，当 $w_i = 0$ 时，运算结果无需修正；当 $w_i = 1$ 时，不管 $c_i[1]$ 等于 0 或者 1， $s_i[1] = s_i[2] = 0$ ，

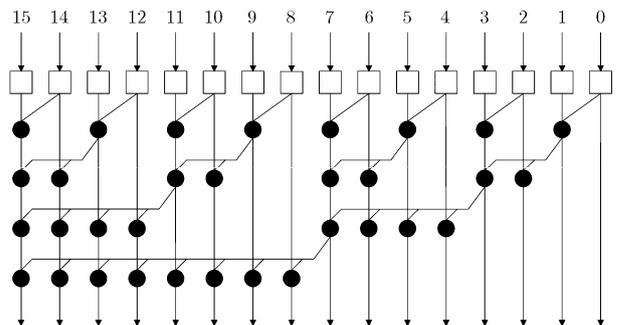


图 4 典型 16 bit Sklansky 前缀结构

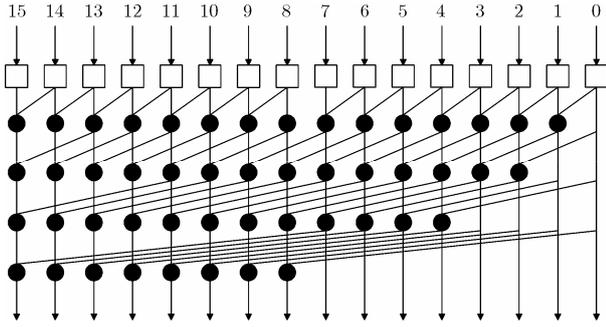


图 5 典型 16 bit Kogge-Stone 前缀结构

$s_i[0]$ 和 $s_i[3]$ 与 w_i 无关。定义 $g_i^*[1:0]$ 是 0 位和 1 位的方块进位产生信号, $p_i^*[1:0]$ 是 0 位和 1 位的方块进位传递信号; 定义 $g_i^*[2:0]$ 是 0 位、1 位和 2 位的方块进位产生信号, $p_i^*[2:0]$ 是 0 位、1 位和 2 位的方块进位传递信号。则

$$\left. \begin{aligned} g_i^*[1:0] &= g_i^*[1] + p_i^*[1] \cdot g_i^*[0] \\ p_i^*[1:0] &= p_i^*[1] \cdot p_i^*[0] \\ g_i^*[2:0] &= g_i^*[2] + p_i^*[2] \cdot g_i^*[1] + p_i^*[2] \cdot p_i^*[1] \cdot g_i^*[0] \\ p_i^*[2:0] &= p_i^*[2] \cdot p_i^*[1] \cdot p_i^*[0] \end{aligned} \right\} (7)$$

根据二进制进位信号 $c_i[k+1] = g_i^*[k:0] \cdot \bar{c}_i + (g_i^*[k:0] + p_i^*[k:0]) \cdot c_i$, ($1 \leq k \leq 2$, 当 $k=0$ 时 $c_i[1] = g_i^*[0] \cdot \bar{c}_i + (g_i^*[0] + p_i^*[0]) \cdot c_i$), 可以得到十进制进位选择加法器的运算逻辑为

$$\left. \begin{aligned} s_i[0] &= h_i^*[0] \cdot \bar{c}_i + \overline{h_i^*[0]} \cdot c_i \\ s_i[1] &= (h_i^*[1] \oplus (g_i^*[0] + w_i)) \cdot \bar{c}_i \\ &\quad + (h_i^*[1] \oplus (p_i^*[0] + w_i)) \cdot c_i \\ s_i[2] &= (h_i^*[2] \oplus (g_i^*[1:0] + w_i)) \cdot \bar{c}_i \\ &\quad + (h_i^*[2] \oplus (g_i^*[1:0] + p_i^*[1:0] + w_i)) \cdot c_i \\ s_i[3] &= (h_i^*[3] \oplus g_i^*[2:0]) \cdot \bar{c}_i \\ &\quad + (h_i^*[3] \oplus (g_i^*[2:0] + p_i^*[2:0])) \cdot c_i \end{aligned} \right\} (8)$$

由式(7)和式(8)得到改进的十进制进位选择加法器如图 6 所示。

32 bit 基于 KS 结构的并行前缀/进位选择加法器由 8 bit KS 结构的并行前缀加法器扩展得到, 产生的 7 个进位输出信号作为十进制 4 bit 进位选择加法器单元进位选择信号。改进的 32 bit 基于 KS 结构的并行前缀/进位选择十进制加法器如图 7 所示。

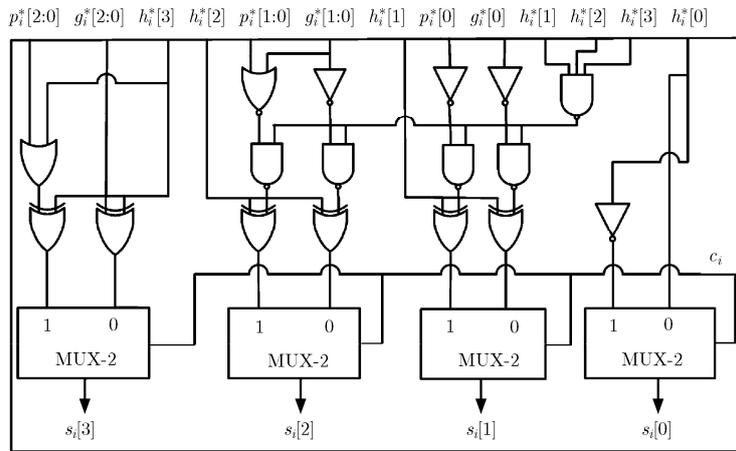


图 6 改进的十进制进位选择加法器单元

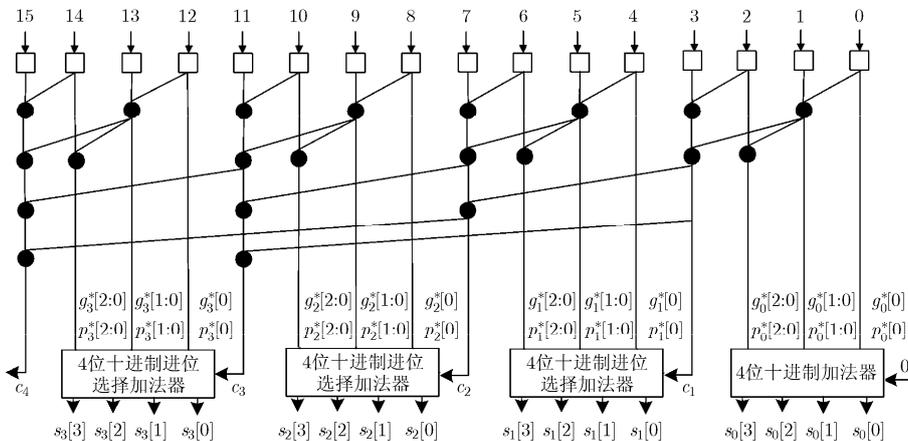


图 7 改进的 16 bit 基于 KS 并行前缀/进位选择结构条件推测性十进制加法器

图 7 中的十进制进位选择加法器模块如图 6 所示。由图 6 和图 7 可以看到，充分利用并行前缀中已经存在的方块进位产生信号和方块进位传递信号来简化十进制进位选择加法电路，避免在进位选择加法器中重复计算，可以减小电路的复杂度。

4 实验结果

64 bit 基于 KS 结构的并行前缀/进位选择加法器由 16 bit KS 结构的并行前缀加法器扩展得到，产生的 15 个进位输出信号作为十进制 4 bit 进位选择加法器单元进位选择信号，128 bit 基于 KS 结构的并行前缀/进位选择加法器由 32 bit KS 结构的并行前缀加法器扩展得到，产生的 31 个进位输出信号作为十进制 4 bit 进位选择加法器单元进位选择信号。使用 Verilog HDL 硬件描述语言分别对 32 bit, 64 bit 和 128 bit 基于并行前缀/进位选择结构的条件推测性十进制加法器进行描述。在 NanGate Open Cell 45nm CMOS 标准工艺库下，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 Power Compiler 获取功耗。最终得到本文提出的 32 bit, 64 bit, 128 bit 十进制加法器，基于 KS 结构的二进制并行前缀/进位选择加法器^[17]和文献[9]中的电路结构的延迟、面积、功耗参数结果如表 1 所示。延时对比和延时-功耗积对比如图 8 和图 9 所示。

从图 8 和图 9 对比结果可知，与文献[9]所采用

的加法器结构相比较，在不增加面积和功耗的情况下，本文提出的 32 bit, 64 bit 和 128 bit 十进制加法器的延迟分别降低 9.5%，9.6%和 12.3%，随着位宽的增加，速度提高的效果更加明显。其延时-功耗积分别减少了 14.5%，13.0%和 13.8%，其性能得到有效的改善。

与基于 KS 结构的 PPF/CSL 的二进制加法器相比较，本文提出的 32 bit, 64 bit 和 128 bit 十进制加法器的延时-功耗积分别增加了 25.9%，28.0%和 6.5%。从综合结果来看，十进制加法器的速度低于二进制加法器。需要说明的是，十进制加法器和二进制加法器的综合结果的比较仅具有参考意义，目前十进制算术运算只是应用于商业和金融等高精度计算领域，它并不能取代二进制算术运算。

5 结束语

条件推测性十进制加法器可以有效地完成十进制加法器运算，本文采用基于 KS 结构的 PPF/CSL 加法器构成条件推测性十进制加法器，并对 4 bit 进位选择单元进行优化设计，利用并行前缀中的方块进位产生信号和方块进位传递信号来简化十进制进位选择加法的电路。从实验结果看出，本文提出的 32 bit, 64 bit 和 128 bit 十进制加法器相比较于文献[9]中的电路结构延时-功耗积分别降低了 14.5%，13.0%和 13.8%。本文提出的条件推测性十进制加法器的性能得到了有效的提升。

表 1 二进制、十进制加法器综合结果比较

位宽	分类	延时(ns)	面积(μm ²)	功耗(μW)	延时-功耗积(PJ)
32 bit	二进制 ^[17]	0.48	1097	362	0.174
	文献[9]	0.73	879	350	0.256
	本文	0.66	842	332	0.219
64 bit	二进制 ^[17]	0.55	2201	701	0.386
	文献[9]	0.83	1791	684	0.568
	本文	0.75	1744	659	0.494
128 bit	二进制 ^[17]	0.60	4473	1782	1.069
	文献[9]	0.97	3647	1363	1.322
	本文	0.85	3613	1340	1.139

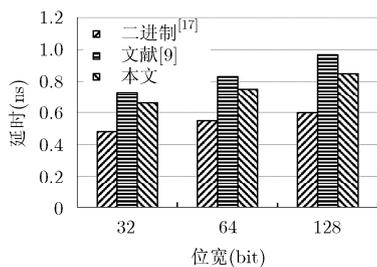


图 8 改进的十进制加法器与文献[9]和文献[17]的延时对比

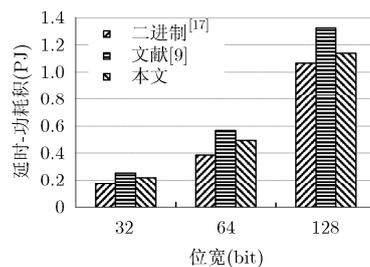


图 9 改进的十进制加法器与文献[9]和文献[17]的延时-功耗积对比

参考文献

- [1] IEEE Std 754(TM)-2008. IEEE standard for floating-point arithmetic[S]. IEEE CS, 2008. doi: 10.1109/ieeestd.2008.4610935.
- [2] EISEN L, WARD J W, TAST H W, *et al.* IBM POWER6 accelerators: VMX and DFU[J]. *IBM Journal of Research and Development*, 2007, 51(6): 663–684. doi: 10.1147/rd.516.0663.
- [3] SCHWARZ E M, KAPERNICK J S, and COWLISHAW M F. Decimal floating-point support on the IBM system z10 processor[J]. *IBM Journal of Research and Development*, 2009, 53(1): 4:1–4:10. doi: 10.1147/JRD.2009.5388585.
- [4] WANG L K, ERLE M A, TSEN C, *et al.* A survey of hardware designs for decimal arithmetic[J]. *IBM Journal of Research and Development*, 2010, 54(2): 8:1–8:15. doi: 10.1147/JRD.2010.2040930.
- [5] SCHMOOKLER M and WEINBERGER A. High speed decimal addition[J]. *IEEE Transactions on Computers*, 1971, 20(8): 862–866. doi: 10.1109/T-C.1971.223362.
- [6] LIU Han, ZHANG Hao, and SEOK-BUM Ko. Area and power efficient decimal carry-free adder[J]. *Electronics Letters*, 2015, 51(23): 1852–1854. doi: 10.1049/el.2015.0786.
- [7] VAZQUEZ A and ANTELO E. Conditional speculative decimal addition[C]. Proceedings of Seventh Conference on Real Numbers and Computers, Nancy, France, 2006: 47–57.
- [8] VAZQUEZ A, ANTELO E, and MONTUSCHI P. Improved design of high-performance parallel decimal multipliers[J]. *IEEE Transactions on Computers*, 2010, 59(5): 679–693. doi: 10.1109/TC.2009.167.
- [9] VAZQUEZ A, ANTELO E, and BRUGUERA J. Fast radix-10 multiplication using redundant BCD codes[J]. *IEEE Transactions on Computers*, 2014, 63(8): 1902–1914. doi: 10.1109/TC.2014.2315626.
- [10] KORNERUP P. Reviewing high-radix signed-digit adders[J]. *IEEE Transactions on Computers*, 2015, 64(5): 1502–1505. doi: 10.1109/TC.2014.2329678.
- [11] MOHANTY B K. Area-delay-power efficient carry-select adder[J]. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2014, 61(6): 418–422. doi: 10.1109/TCSII.2014.2319695.
- [12] MATHEW S K, ANDERS M, KRISHNAMURTHY R K, *et al.* A 4 GHz 130 nm address generation unit with 32-bit sparse-tree adder core[J]. *IEEE Journal of Solid-State Circuits*, 2003, 38(5): 689–695. doi: 10.1109/JSSC.2003.810056.
- [13] KOGGE P M and STONE H S. A parallel algorithm for efficient solution of a general class of recurrence equations[J]. *IEEE Transactions on Computers*, 1973, 22(8): 786–793. doi: 10.1109/TC.1973.5009159.
- [14] BRENT R P and KUNG H T. A regular layout for parallel adders[J]. *IEEE Transactions on Computers*, 1982, 31(3): 260–264. doi: 10.1109/TC.1982.1675982.
- [15] SKLANSKY J. Conditional-sum addition logic[J]. *IRE Transactions on Electronic Computers*, 1960, EC-9(2): 226–231. doi: 10.1109/TEC.1960.5219822.
- [16] HAN TACKDON and CARLSON D A. Fast area-efficient VLSI adders[C]. IEEE 8th Symposium on Computer Arithmetic, 1987: 49–56. doi: 10.1109/ARITH.1987.6158699.
- [17] DIMITRAKOPOULOS G and NIKOLOS D. High-speed parallel-prefix VLSI Ling adders[J]. *IEEE Transactions on Computers*, 2005, 54(2): 225–231. doi: 10.1109/TC.2005.26.
- [18] HE Yajuan and CHANG C H. A power-delay efficient hybrid carry-lookahead/carry-select based redundant binary to two's complement converter[J]. *IEEE Transactions on Circuits & Systems I: Regular Papers*, 2008, 55(1): 336–346. doi: 10.1109/TCSI.2007.913610.
- 崔晓平: 女, 1962年生, 副教授, 硕士生导师, 研究方向为数字集成电路设计和计算机算术运算系统.
- 王书敏: 男, 1990年生, 硕士生, 研究方向为数字系统设计与计算机应用.
- 刘伟强: 男, 1983年生, 副教授, 硕士生导师, 研究方向为数字集成电路设计和加密硬件.
- 董文雯: 女, 1993年生, 硕士生, 研究方向为数字系统设计与计算机应用.