

一种改进的基于 Kogge-Stone 结构的并行前缀加法器

赵翠华, 娄 晃, 张洵颖, 沈绪榜
(西安微电子技术研究所, 陕西 西安 710054)

摘 要: 基于并行前缀算法的 Kogge-Stone 结构, 通过改进其结构层次上的逻辑电路, 提出一种改进的并行前缀加法器. 与传统电路相比, 该加法器不仅可以减小面积、功耗和延时, 而且随着位宽的加大其优势更加明显, 是适用于宽位的并行前缀加法器.

关键词: 并行前缀算法; Kogge-Stone 结构; 并行前缀加法器

中图分类号: TP332.2+1

文献标识码: A

文章编号: 1000-7180(2011)02-0047-04

Advanced Parallel Prefix Adder Based on Kogge-Stone Architecture

ZHAO Cui-hua, LOU Mian, ZHANG Xun-ying, SHEN Xu-bang

(Xi'an Microelectronics Technology Institute, Xi'an 710054, China)

Abstract: Based on Kogge-Stone architecture of parallel prefix algorithm, an advanced parallel prefix adder by improving the logic circuit of its architecture levels is proposed. Compared with the traditional one, this adder not only can save the area, power dissipation and delay, but also have more preponderant with the operand bits increase, is suitable for parallel prefix adder of multi-bit operands.

Key words: prallel prefix algorithm; Kogge-Stone architecture; parallel prefix adder

1 引言

加法器是处理器数据路径的核心, 是实现乘法器和除法器的基础, 它直接影响处理器的性能以及系统整体性能. 随着处理器运算速度的不断提高, 对加法器速度的要求也越来越高. 虽然到目前为止, 已经提出了很多成熟的加法器的算法和结构, 但高速加法器的设计仍然是研究的热点, 其中, 并行前缀加法器 (Parallel Prefix Adder) 的设计更是热点中的热点. 近几年有很多关于并行前缀加法器的研究^[1-5], 并行前缀加法器是二进制加法器中一类通用的加法器, 它通过改变直通非循 (Directed Acyclic Graph) 的拓扑结构, 可涵盖由行波进位加法器到较快的超前进位加法器等很多种类的加法器^[6]. 并行前缀加法器由许多相同的前缀运算单元组成, 结构规整, 非常适用于 VLSI 实现.

并行前缀算法有很多具体实现结构, 具有代表

性的并行前缀结构有 Sklansky 结构^[1]、Brent-Kung 结构^[2]和 Kogge-Stone 结构^[3]等. 本文将对并行前缀算法的结构进行分析和研究, 并基于 Kogge-Stone 结构, 通过改进其结构层次上的逻辑电路, 提出一种改进的并行前缀加法器.

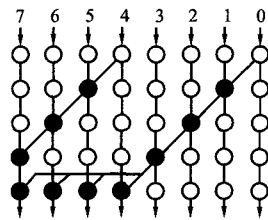
2 并行前缀算法

并行前缀算法可以分为 2 类: 组前缀算法和树前缀算法^[7]. 组前缀算法是将操作数划分为不同位数的组, 使前缀算法有最大的并行级数, 图 1(a) 所示为一个 2 组 1 级并行的组前缀算法结构图. 树前缀算法是将操作数划分为固定位数的组, 使组前缀算法有最大的并行级数, 其典型的结构如图 1(b)~(d) 所示. 组前缀和树前缀算法关系密切, 在组前缀结构上应用最大数目的并行级数, 就可以获得树前缀算法. 实际没有必要区分组前缀和树前缀算法, 只需将其都看作并行前缀算法.

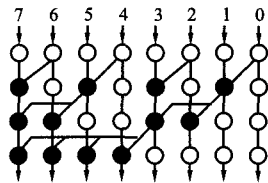
并行前缀加法器由预处理阶段、前缀级进位链阶段和后处理阶段组成. 预处理阶段主要是计算初始各位的进位产生信号和进位传播信号; 前缀级进位链阶段主要是计算最终各位的进位产生信号以及进位传播信号; 后处理阶段求最终的和及进位输出.

表 1 三种结构的并行前缀加法器比较

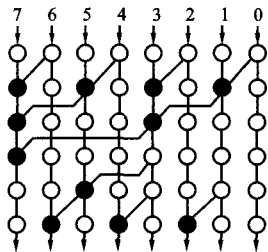
并行前缀结构	计算时间 t	计算面积 A	面积时间乘积 $A \cdot t$	互连线面积 A_{tracks}	最大扇出
Sklansky	$\log n$	$n \log n / 2$	$n \log^2 n / 2$	$\log n$	$n / 2$
Brent-Kung	$2 \log n - 2$	$2n - \log n - 2$	$4n \log n$	$2 \log n - 1$	$\log n$
Kogge-Stone	$\log n$	$n \log n - n - 1$	$n \log^2 n$	$n - 1$	2



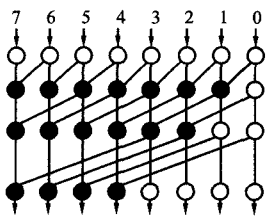
(a) 组前缀算法: 2组1级并行



(b) Sklansky结构



(c) Brent-Kung 结构



(d) Kogge-Stone 结构

图 1 并行前缀算法的结构

可以看出, 与传统的串行结构相比, 这三种结构都以面积换取了时间. 在这三种结构中, Sklansky 结构具有最短的计算时间和最小的互连面积, 但是随着操作数位数的增大, 其最大扇出线性增长, 大的扇出导致运算速度变慢, 也使晶体管尺寸优化变得困难. Brent-Kung 结构的面积-时间乘积是最小的, 但是它的计算时间比较长. Kogge-Stone 结构和 Sklansky 结

其中, 并行前缀进位链阶段是最重要的. 根据文献[7], Sklansky 结构、Brent-Kung 结构和 Kogge-Stone 结构的并行前缀加法器的比较见表 1, 其中 n 为操作数位数, $\log n$ 指 $\log_2 n$.

构的计算时间相同, 都是最短的. 虽然在计算过程中需要复制进位树结构来产生中间进位, 带来了面积和功耗上较大的代价, 但是它的扇出, 特别是关键路径上的扇出, 基本上是一个常数 2, 这使得该结构的互连结构规则, 容易实现与扩展, 延迟也较小. 从电路特性上来分析, 随着工艺尺寸的不断缩小, Kogge-Stone 结构具有一定的优势[8]. 在速度最重要的处理器中, Kogge-Stone 结构是一个很好的选择.

由于动态电路的高速性, 很多加法器是利用动态电路设计的, 但是静态电路具有其独特的优势: 它噪声容限大, 稳定性高, 易实现; 没有时钟所带来的复杂性、不可靠性, 以及功耗问题; 具有可重用性, 可根据工艺参数而改变比例. 通过算法的改进, 在静态电路速度满足需求的情况下, 选用静态电路是当前电路设计的趋势, 故本文将采用静态电路来设计加法器.

3 基于 Kogge-Stone 结构的并行前缀加法器设计

对于 N 位加法器, 定义 $A = a_{N-1}a_{N-2}\cdots a_1a_0$ 为被加数, $B = b_{N-1}b_{N-2}\cdots b_1b_0$ 为加数, C_{in} 为低位进位输入, $S = s_{N-1}s_{N-2}\cdots s_1s_0$ 为求得的和, C_{out} 为高位进位输出, g 为进位产生信号, p 为进位传播信号. 其具体设计按预处理、前缀级进位链和后处理三个阶段来分别说明.

3.1 传统的 Kogge-Stone 结构的并行前缀加法器设计

预处理阶段: 此阶段作为并行前缀算法的第 0 级, 计算 g 和 p 所采用的逻辑公式为

$$g_0 = C_{\text{in}}$$

$$g_i = a_{i-1} \cdot b_{i-1}, i = 1 \sim N$$

$$p_i = a_i \oplus b_i, i = 0 \sim N-1$$

前缀级进位链阶段: 首先计算前缀级数, 计算公式为 $n = \lceil \log_2 N \rceil$, 其中 n 即为求得的前缀级数. 然后

根据第0级的结果对前缀级数进行计算(1~n级).

$$p_i = p_i \cdot p_{i+2^{n-1}}, i = 0 \sim N - 2^n$$

$$g_i = g_i, i = 0 \sim 2^{n-1} - 1$$

$$g_i = p_{i-2^{n-1}} \cdot g_{i-2^{n-1}} + g_i, i = 2^{n-1} - N$$

后处理阶段:

$$s_i = a_i \oplus b_i \oplus g_i, i = 0 \sim N - 1$$

$$C_{out} = g_N + g_o \cdot p_0$$

3.2 改进的 Kogge-Stone 结构的并行前缀加法器设计

预处理阶段:此阶段作为并行前缀算法的第0级,计算 g 和 p 所采用的逻辑公式为

$$\overline{g_0} = \overline{C_{in}}$$

$$\overline{g_i} = \overline{a_{i-1} \cdot b_{i-1}}, i = 1 \sim N$$

$$\overline{p_i} = \overline{a_i + b_i}, i = 0 \sim N - 1$$

前缀级进位链阶段:首先计算前缀级数,计算公式为 $n = \lceil \log_2 N \rceil$,其中 n 即为求得的前缀级数.然后根据第0级的结果对前缀级数进行计算(1~n级),相邻两级之间采用反相逻辑,根据计算级数的奇偶不同,其逻辑公式有所不同.

奇数级:

$$p_i = \overline{p_i + p_{i+2^{n-1}}}, i = 0 \sim N - 2^n$$

$$g_i = \overline{g_i}, i = 0 \sim 2^{n-1} - 1$$

$$g_i = \overline{(p_{i-2^{n-1}} + g_{i-2^{n-1}}) \cdot g_i}, i = 2^{n-1} - N$$

偶数级:

$$p_i = p_i \cdot p_{i+2^{n-1}}, i = 0 \sim N - 2^n$$

$$\overline{g_i} = \overline{g_i}, i = 0 \sim 2^{n-1} - 1$$

$$g_i = p_{i-2^{n-1}} \cdot g_{i-2^{n-1}} + g_i, i = 2^{n-1} - N$$

后处理阶段:按前缀级数 n 为奇偶来分别计算.

n 为奇数:

$$s_i = a_i \oplus b_i \oplus g_i, i = 0 \sim N - 1$$

$$C_{out} = g_N + g_o \cdot p_0$$

n 为偶数:

$$s_i = \overline{a_i \oplus b_i \oplus g_i}, i = 0 \sim N - 1$$

$$C_{out} = \overline{g_N \cdot (g_o + p_0)}$$

3.3 结构层次上的改进

与传统的 Kogge-Stone 结构的并行前缀加法器相比,此加法器在结构层次上做了以下改进:

(1) 在预处理阶段用或逻辑代替传统的异或逻辑来计算进位产生传播信号 p . 由于所得到的进位传播信号 p 是用来计算进位信号 g 的,对和 S 的计算不产生影响,且在计算进位输出 C_{out} 时,其逻辑和异或逻辑所得的结果是相同的. 用 CMOS 工艺来实现时,用或非门电路来代替异或电路,可以减少延时和面积.

(2) 在预处理阶段用反相逻辑代替传统的正相逻辑来计算进位产生信号 g 和进位传播信号 p . 用 CMOS 工艺来实现时,与门是由与非门和非门组成的,或门是由或非门和非门组成的,此处直接利用与非门和或非门电路来实现既减少了延时,又减少了面积.

(3) 在前缀级进位链阶段相邻两级之间采用反相逻辑来实现. 这样不仅可以避免电路产生误动作,增强稳定性,而且还可以加快加法器运算的速度. 在进位链计算 p 和 g 的过程中要使用与非、或非、与或非和或与或非电路,这比使用传统逻辑要用到的与、与或门电路的延时要小的多. 另外,各级计算完成的 g 用反相器来延时,可以加大信号的驱动能力.

3.4 具体实现

采用 VHDL 硬件描述语言在 Modelsim SE PLUS 6.1c 软件环境下,根据结构层次上改进的 Kogge-Stone 结构实现了 32/64/128 位并行前缀加法器. 对于 32 位并行前缀加法器,由于前缀级进位链阶段是最重要的,故此处仅对前缀级进位链阶段进行说明. 具体求 p 和 g 的 Kogge-Stone 结构分别如图 2、3 所示,其中图中第一行的结点表示预处理阶段所计算的 p 和 g (p, g 为反相),下面的灰色结点表示反相器,黑色结点表示具体的逻辑电路.

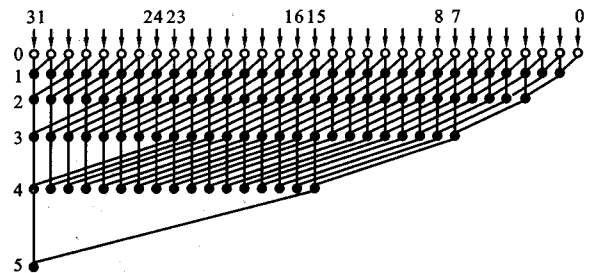


图2 32位加法器求 p 的 Kogge-Stone 树

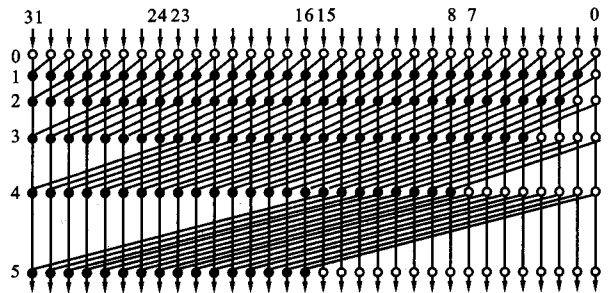


图3 32位加法器求 g 的 Kogge-Stone 树

其具体逻辑公式如下:

第1级:

$$p_i = \overline{p_i + p_{i+1}}, i = 0 \sim 30$$

$$g_i = \overline{g_i}, i = 0$$

$$g_i = \overline{(p_{i-1} + g_{i-1}) \cdot g_i}, i = 1 \sim 32$$

第 2 级:

$$\overline{p_i} = \overline{p_i \cdot p_{i+2}}, i = 0 \sim 28$$

$$\overline{g_i} = \overline{g_i}, i = 0 \sim 1$$

$$\overline{g_i} = \overline{p_{i-2} \cdot g_{i-2} + g_i}, i = 2 \sim 32$$

第 3 级:

$$\overline{p_i} = \overline{\overline{p_i} + \overline{p_{i+4}}}, i = 0 \sim 24$$

$$\overline{g_i} = \overline{g_i}, i = 0 \sim 3$$

$$\overline{g_i} = \overline{(\overline{p_{i-4}} + \overline{g_{i-4}}) \cdot g_i}, i = 4 \sim 32$$

第 4 级:

$$\overline{p_i} = \overline{p_i \cdot p_{i+8}}, i = 0 \sim 16$$

$$\overline{g_i} = \overline{g_i}, i = 0 \sim 7$$

$$\overline{g_i} = \overline{p_{i-8} \cdot g_{i-8} + g_i}, i = 8 \sim 32$$

第 5 级:

$$\overline{p_i} = \overline{p_i \cdot p_{i+16}}, i = 0$$

$$\overline{g_i} = \overline{g_i}, i = 0 \sim 15$$

$$\overline{g_i} = \overline{p_{i-16} \cdot g_{i-16} + g_i}, i = 16 \sim 32$$

64/128 位并行前缀加法器的设计与 32 位并行前缀加法器的设计相同. 假定使用传统 CMOS 工艺来完成 32/64/128 位并行前缀加法器设计, 则非门需 2 个晶体管; 与非门、或非门需要 4 个晶体管; 与门、或与非门和与或非门需要 6 个晶体管; 与或门需要 8 个晶体管; 异或门需要 10 个晶体管^[9]. 由此可以得到传统电路与改进电路的晶体管数目, 见表 2.

表 2 传统电路与改进电路的比较

方法	类别	位宽/bit	晶体管数目/个
传统电路	总晶体管数目	32	2 850
		64	6 512
		128	14 718
	关键路径晶体管数目	32	60
		64	68
		128	76
改进电路	总晶体管数目	32	2 184
		64	4 946
		128	11 100
	关键路径晶体管数目	32	44
		64	50
		128	56

由表 2 可以看出, 改进电路的总晶体管数目和关键路径晶体管数目明显少于传统电路, 因此, 改进电路在面积、功耗和延迟方面优于传统电路. 随着位

宽的加大, 其总晶体管数目的差距越来越大, 改进电路的优势越来越明显, 适合于宽位的并行前缀加法器的实现.

4 结束语

Kogge-Stone 结构的并行前缀加法器的扇出基本上是一个常数 2, 结构规整, 容易实现与扩展. 本文基于 Kogge-Stone 结构, 对其结构层次上的逻辑电路进行改进, 可进一步减小面积、功耗和延时, 且随着位宽的加大, 改进电路的优势将更明显, 适合于宽位的并行前缀加法器.

参考文献:

- [1] Sklansky J. Conditional sum addition logic[J]. IRE Trans Electron Computers, 1960, EC-9(6):226-231.
- [2] Brent R P, Kung H T. A regular layout for parallel adders[J]. IEEE Transactions Computers, 1982, 31(3):260-264.
- [3] Kogge P M, Stone H S. A parallel algorithm for efficient solution of a general class of recurrence equations[J]. IEEE Trans Computers, 1973, 22(8):786-793.
- [4] Matthew M Ziegler, Mircea R Stan. Unified design space for regular parallel prefix adders[J]. Design Automation and Test in Europe Conference and Exhibition, 2004(2):1386-1387.
- [5] Zhu Haikun, Cheng Chungkuan, Ronald Graham. Constructing zero-deficiency parallel prefix adder of minimum Depth[J]. ASP-DAC, 2005(2):883-888.
- [6] 王骞, 马铁夫. 一种稀疏树加法器及结构设计[J]. 电子器件, 2005(2):312-314.
- [7] Reto Zimmermann. Binary Adder Architecture for Cell-Based VLSI and their Synthesis[D]. Zurich: Swiss Federal Institute of Technology, 1997.
- [8] 勒战鹏. 高速浮点加法运算单元的研究与实现[D]. 西安: 西北工业大学, 2006.
- [9] 李玉山, 来新泉. 电子系统集成设计导论[M]. 西安: 西安电子科技大学出版社, 2008.

作者简介:

赵翠华 女, (1986—), 硕士研究生. 研究方向为嵌入式计算机系统结构.

姜 冕 男, (1987—), 硕士研究生. 研究方向为 IC 设计技术.

张洵颖 男, (1981—), 高级工程师. 研究方向为嵌入式计算机系统结构.

沈绪榜 男, (1933—), 教授, 博士生导师, 中国科学院院士. 研究方向为计算机系统结构.