第 24 卷第 5 期　　半 导 体 学 报　　Vol. 24, No. 5
2003 年 5 月　　CHINESE JOURNAL OF SEMICONDUCTORS　　May, 2003

一种改进的近似平方算法的 VLSI 实现

李 侠　章倩苓

(复旦大学专用集成电路与系统国家重点实验室, 上海　200433)

摘要: 提出了一种适用于 Viterbi 算法的改进的近似平方算法——二阶近似算法. 该算法最大相对误差(maximum relative error, MRE)和平均相对误差(average relative error, ARE)都非常低, 与最新报道相比, MRE 和 ARE 分别减小了 20% 和 70% 左右. 同时, 在 0.6μm CMOS 工艺条件下, 实现了基于该算法的 7-bit 平方器, 其延时和晶体管数与最新报道相当.

关键词: Viterbi 算法; 近似平方算法; PLA; MRE; ARE

EEACC: 2570D; 1265B

中图分类号: TN402　　**文献标识码**: A　　**文章编号**: 0253-4177(2003)05-0539-05

1 引言

Viterbi 算法是信号处理中一类非常重要的算法, 主要应用于卷积码解码[1,5], 而平方运算是 Viterbi 算法中最基本的运算之一[5].

目前, 实现平方算法有两种方案: 精确算法实现[2]和近似算法实现[1]. 精确算法实现一般采用查找表法或加法器-选择器法[2], 它们的主要缺点是电路规模很大, 并且最大延时较大; 近似算法实现由 Eshraghi 等人首先提出[1], 与精确算法实现相比, 它的电路规模与最大延时均大大减小, 但相对误差较大. 当实现 10-bit 平方器时, 最大相对误差(MRE)甚至高达 24.95%. 最新报道[4]中, Sheu 等人对 Eshraghi 的近似算法进行改进, 进一步减小了电路规模与延时, 同时也降低了相对误差, 但仍然比较大. 本文提出了一种改进的近似平方算法——二阶近似算法, 与 Sheu 算法相比, 它的 MRE 和平均相对误差(ARE)显著减小. 由于最新报道中, Sheu 等人给出了 0.6μm CMOS 工艺条件下 7-bit 平方器的实现结果, 为了便于比较, 本文也在 0.6μm CMOS 工艺条件下, 实现了基于二阶近似算法的 7-bit 平方器, 其延时和晶体管数与 Sheu 等人实现结果相当. 因此, 本文算法特别适合于对平方算法精度要求较高、对电路规模与最大延时要求很严的 Viterbi 算法的实现.

2 算法原理

2.1 Eshraghi 算法

平方算法基本原理是给出 A, 求得 A^2, 其中 A 为 n 比特, A^2 为 $2n$ 比特. 下面先给出 Eshraghi 等人提出的近似平方算法.

设有 n-bit 的二进制数 A, 表示为 $A = a_n a_{n-1} \times a_{n-2} \cdots a_1$, 为了进行近似平方运算, 将 A 改写为:

$$A = a_n \underbrace{00\cdots0}_{n-1\text{zero}} + a_{n-1}a_{n-2}\cdots a_1 \tag{1}$$

根据 $(x+y)^2 = x^2 + 2xy + y^2$, 可得

$$A^2 = \left(a_n \underbrace{00\cdots0}_{n-1\text{zero}}\right)^2 + 2\left(a_{n-1}a_{n-2}\cdots a_1\right)\left(a_n \underbrace{00\cdots0}_{n-1\text{zero}}\right) + \left(a_{n-1}a_{n-2}\cdots a_1\right)^2 \tag{2}$$

忽略最后一项, 并提出前两项的公共项得近似值 D 为

$$D = \left(a_n \underbrace{00\cdots0}_{n-1\text{zero}}\right)\left(a_n \underbrace{00\cdots0}_{n-1\text{zero}} + a_{n-1}a_{n-2}\cdots a_1 0\right) \tag{3}$$

李　侠　男, 1978 年出生, 博士研究生, 主要研究方向为高性能 VLSI 设计.

章倩苓　女, 1936 年出生, 教授, 博士生导师, 主要研究方向为 VLSI 系统集成等.

2002-06-27 收到, 2002-08-28 定稿

参考文献[1]中的图4给出了该近似平方电路实现的框图,主要包括Controller、桶型移位电路及modifier三个模块.该算法实现电路规模及最大延时均比较优化,但MRE和ARE较大.为了减小相对误差,本文提出了二阶近似算法.

2.2 二阶近似算法

令 $A = a_n a_{n-1} a_{n-2} \cdots a_1$,将 A 改写如下:

$$A = a_n a_{n-1} \underbrace{00\cdots0}_{n-2\text{zero}} + a_{n-2} a_{n-3} \cdots a_1 \tag{4}$$

根据 $(x+y)^2 = x^2 + 2xy + y^2$ 可得

$$\begin{aligned} A^2 = {} & (a_n a_{n-1} \underbrace{0\cdots0}_{n-2\text{zero}})^2 \\ & + 2(a_{n-2} a_{n-3} \cdots a_1)(a_n a_{n-1} \underbrace{0\cdots0}_{n-2\text{zero}}) \\ & + (a_{n-2} a_{n-3} \cdots a_1)^2 \end{aligned} \tag{5}$$

忽略最后一项,并提出前两项的公共项得近似值 D 为

$$D = (a_n a_{n-1} \underbrace{0\cdots0}_{n-2\text{zero}})(a_n a_{n-1} \underbrace{0\cdots0}_{n-2\text{zero}} + a_{n-2} a_{n-3} \cdots a_1 0) \tag{6}$$

其中 a_n 为输入 A 中最高非零位.如果直接实现(6)式,当 a_n、a_{n-1} 均为1时,需进行加法运算,这增大了实现的复杂度.因此对(6)式进行修正,得 D 的近似值DA为:

$$\mathrm{DA} = \begin{cases} (\underbrace{10\cdots0}_{n-1\text{ zero}})\{1, a_{n-2}, a_{n-1} \& \overline{a_{n-2}} \mid a_{n-3}, a_{n-4}, \cdots, a_1, 0\} \\ \qquad \text{when} \quad a_{n-1} = 0 \\ (\underbrace{10\cdots0}_{n\text{ zero}})\{1, a_{n-2}, a_{n-1} \& \overline{a_{n-2}} \mid a_{n-3}, a_{n-4}, \cdots, a_1, 0\} \\ \qquad \text{when} \quad a_{n-1} = 1 \end{cases} \tag{7}$$

其中 $\{1, a_{n-2}, a_{n-1} \& \overline{a_{n-2}} \mid a_{n-3}, a_{n-4}, \cdots, a_1, 0\}$ 表示将这些比特拼接起来,$a_{n-1} \& \overline{a_{n-2}} \mid a_{n-3}$ 表示 a_{n-2} 先取反,然后与 a_{n-1} 相与,再与 a_{n-3} 相或.同时,根据分析 A^2 的值可知,A^2 的二进制表达式中,最低位与 A 的最低位相同;A^2 次低位的值为0.因此进一步对结果DA做如下修正:

$$\begin{aligned} \mathrm{DA}[1] &= 0 \\ \mathrm{DA}[0] &= A[0] \end{aligned} \tag{8}$$

其中 DA[1]、DA[0]分别表示DA的次低位和最低位;$A[0]$ 表示 A 的最低位,即 a_1.可以看到,(7)、(8)式实现的近似平方算法与Eshraghi算法相比,在二阶近似的基础上进行了修正,因此称之为"二阶近似算法",它避免了(6)式中的加法运算,大大简化了实现复杂度.定义相对误差为[4]:

$$E = (A^2 - \mathrm{DA})/A^2 \times 100\% \tag{9}$$

其中 A^2 为精确值.定义MRE和ARE分别为:

$$\mathrm{MRE} = \max\{E_i\} \quad i = 0, 1, \cdots, 2^n - 1 \tag{10}$$

$$\mathrm{ARE} = \left(\sum_{i=0}^{2^n-1} E_i\right)/2^n \tag{11}$$

表1中列出了 $n = 4 \sim 10$ 时Eshraghi算法、Sheu算法以及本文二阶近似算法的MRE和ARE.可以看出,二阶近似算法ARE比Sheu算法减小了70%左右($n = 10$ 时达到70.9%),MRE在 $n > 5$ 时比Sheu算法减小了20%左右($n = 10$ 时达到25.3%),并且,随着 n 增大,MRE的减小更加明显,因此本文二阶近似算法的相对误差显著降低.

表1 相对误差比较

Table 1 Comparison of relative error

n	Eshraghi算法		Sheu算法		本文二阶近似算法	
	MRE/%	ARE/%	MRE/%	ARE/%	MRE/%	ARE/%
4	21.78	7.71	9.47	1.04	11.11	0.30
5	23.41	9.15	13.17	1.96	12.80	0.40
6	24.21	10.07	15.87	2.77	13.22	0.80
7	24.61	10.62	17.27	3.36	13.42	1.02
8	24.80	10.95	17.66	3.77	13.51	1.14
9	24.90	11.13	18.01	4.04	13.56	1.20
10	24.95	11.24	18.19	4.21	13.58	1.23

注:n 为输入比特宽度

3 VLSI实现结构

为了评估二阶近似算法平方器实现的性能,本文实现了 $n = 7$ 时基于该算法的近似平方电路.图1为7-bit近似平方器实现的结构框图.

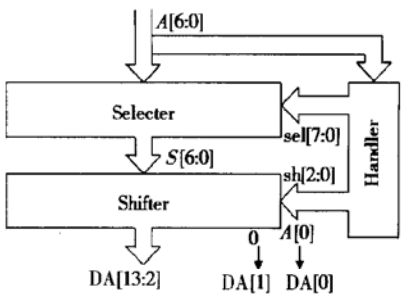

图1 7-bit近似平方电路实现框图

Fig. 1 Implementation dagram of a 7-bit approximate squaring algorithm

图1中,$A[6:0]$ 为7-bit输入数据,亦可表示为 $a_7 a_6 \cdots a_1$,$\mathrm{DA}[13:0]$ 为14-bit结果.其中,se-

lecter 模块获得(7)式后半部分的值, shifter 模块实现(7)式前半部分的移位功能, 而 handler 则产生 selecter、shifter 的控制信号. 下面分别说明各个子模块的实现.

3.1　预处理模块 selecter

selecter 模块获得(7)式后半部分的值, 其功能可用表 2 真值表描述.

表 2　Selecter 模块的真值表

Table 2　Truth table for selector module

a_7	a_6	a_5	a_4	a_3	a_2	a_1	$S[6:0]$						
0	0	0	0	0	0	X	0	0	0	0	0	0	0
0	0	0	0	0	1	X	0	0	0	0	0	1	0
0	0	0	0	1	X	X	0	0	0	0	1	a_1	$a_2\&\overline{a_1}$
0	0	0	1	X	X	X	0	0	0	1	a_2	$a_3\&\overline{a_2}\mid a_1$	0
0	0	1	X	X	X	X	0	0	1	a_3	$a_4\&\overline{a_3}\mid a_2$	a_1	0
0	1	X	X	X	X	X	0	1	a_4	$a_5\&\overline{a_4}\mid a_3$	a_2	a_1	0
1	X	X	X	X	X	X	1	a_5	$a_6\&\overline{a_5}\mid a_4$	a_3	a_2	a_1	0

$S[6:0]$ 即对应图 1 中 selecter 模块的输出. 由表 2 可见, $S[6]$ 即为 a_7, $S[5]$ 的前 6 种情形即为 a_6 值, $S[4:0]$ 可以类推, 因此 $S[6:0]$ 可由 $a_7, a_6, \cdots, a_1$ 以及它们的逻辑组合(如 $a_6\&\overline{a_5}\mid a_4$, $a_5\&\overline{a_4}\mid a_3$ 等)经多路选择器选择得到, 而多路选择器的控制信号 sel[7:0] 由 handler 模块产生.

3.2　桶型移位器 shifter

shifter 模块实现(7)式前半部分的功能. 根据(7)式, 可得桶型移位器移位位数 sh[2:0] 如表 3 所示, 而 sh[2:0] 由 handler 模块产生.

表 3　Shifter 移位位数真值表

Talbe 3　Truth table for shift amount

a_7	a_6	a_5	a_4	a_3	a_2	a_1	sh[2:0]		
0	0	0	0	0	0	X	0	0	0
0	0	0	0	0	1	X	0	a_1	$\overline{a_1}$
0	0	0	0	1	X	X	0	1	a_2
0	0	0	1	X	X	X	a_3	$\overline{a_3}$	$\overline{a_3}$
0	0	1	X	X	X	X	1	0	a_4
0	1	X	X	X	X	X	1	a_5	$\overline{a_5}$
1	X	X	X	X	X	X	1	1	a_6

桶型移位器的电路如图 2 所示. 为了减少晶体管数, 二选一多路选择器采用图 2 中所示的两管结构, 同时, 在 shifter 的输出端增加两级非门, 以保证输出端的驱动能力以及波形的规整性. 由于最低两位按照(8)式进行修正, 因此移位器中最后一行可节省两个二选一. 移位器的输出构成最终结果的高 12 位, 即 DA[13:2].

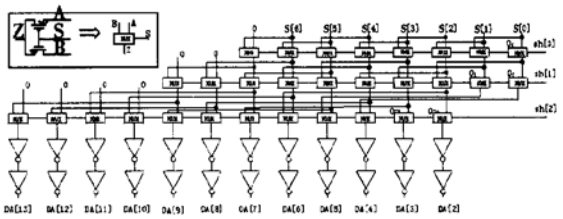

图 2　桶型移位器电路

Fig. 2　Barrel shifter

3.3　控制模块 handler

控制模块产生 selecter、shifter 模块所需的控制信号 sel[7:0] 和 sh[2:0]. sel[7:0] 和 sh[2:0] 的值分别根据表 2 和表 3 的真值表得到, 它们均是 $a_7, a_6, \cdots, a_1$ 逻辑表达式. 考虑到使用 PLA 实现随机逻辑可显著减少电路的规模与延时[6,8], 因此本文使用 PLA 实现 handler 模块.

Handler 模块的电路图如图 3 所示. PLA 实现基于如下原理: 任意组合逻辑均可以化为与或形式, 然后使用 MOS 管分别构成与阵列和或阵列. 图 3 中的上半部分为与阵列, 下半部分为或阵列, 输入和输出均使用 p 型 MOS 管作为负载, 同时, 在输入输出端使用非门以获得正确的逻辑功能及足够的驱动能力. 在 handler 电路中, 产生 sel[7:0] 时, 经过适当的布尔表达式化简, 可以得到多个 "与"项公共项, 使用 PLA 实现时, 利用这些公共项, 可大大减少晶体管数.

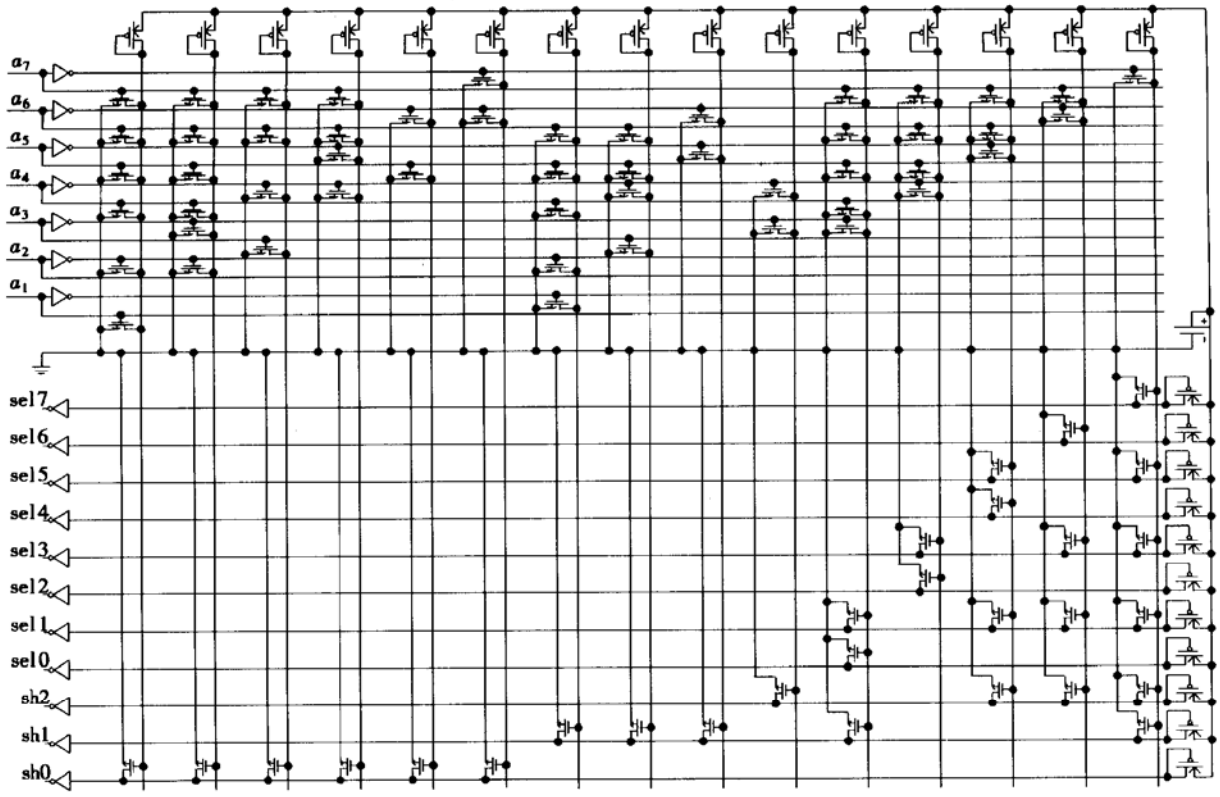

图 3 基于 PLA 的 handler 电路

Fig. 3 Handler circuit based on PLA

4 实现结果

由于在最新报道中, Sheu 等人给出了 0.6μm CMOS 工艺条件下 7-bit 平方器的实现结果, 为了便于比较, 本文也在 0.6μm CMOS 工艺条件下实现了基于二阶近似算法的 7-bit 平方器, 使用 Workview Office 中的 Viewdraw 工具输入晶体管级电路图, Wspice 生成 spice 网表, 共使用了 203 个晶体管. 表 4 中列出了几种实现方法的性能比较.

表 4 7-bit 平方器的性能比较

Table 4 Performance comparison of implementation of 7-bit squaring function

	精确实现	Eshraghi	Sheu	本文实现
MRE	/	24.61	17.27	13.42
ARE	/	10.62	3.36	1.02
晶体管数	784	275	184	203

使用 Hspice 进行仿真, 得到上升、下降延时分别为 1.6ns 和 1.1ns, 而 Sheu 方法在 0.6μm CMOS 工艺条件下上升、下降延时分别为 3.1ns 和 2.5ns[4]. 考虑到 Sheu 的结果包含了 I/O pad 的延时, 而 0.6μm CMOS 工艺条件下输入 pad 的上升、下降延时一般为 0.23ns、0.25ns, 输出 pad 的上升、下降延时一般为 1.00ns、1.00ns[7], 因此本文实现的 7-bit 平方器延时与 Sheu 实现的结果相当.

可见, 本文实现的基于二阶近似算法的 7-bit 平方器, 其 MRE 和 ARE 与最新报道相比均显著降低, 而实现电路的延时和晶体管数均与最新报道的相当.

5 结论

本文给出了一种适用于 Viterbi 算法的改进的近似平方算法——二阶近似算法, 与最新报道[4]相比, 它的最大相对误差和平均相对误差均有显著降低. 同时, 在 0.6μm CMOS 工艺库实现了基于该算法的 7-bit 平方电路, 它的延时与晶体管数均与最新报道相当. 因此, 本文算法非常适合于对平方算法精度要求很高、对电路规模与最大延时要求很严的 Viterbi 算法的实现.

参考文献

[1] Eshraghi A, Fiez T S, et al. Design of a new squaring function for the viterbi algorithm. IEEE J Solid-State Circuits, 1994, 29(9): 1102

[2] Shammanna M, Whitaker S, et al. Cellular logic array for computation of squares. In: 3rd NASA Symp VLSI Design, 1991, 2.4.1

[3] Hiasat A A, Abdel-Aty-Zohdy H S. Combinational logic approach for implenenting an improved approxiamate squaring function. IEEE J Solid-State Circuits, 1999, 34(2): 1102

[4] Sheu M H, Lin S H. Fast compensative design approach for the approximate squaring function. IEEE J Solid-State Circuits, 2002, 37(1): 95

[5] Liu Fuquan. The technique and application of error control coding. Harbin: Publishing House of the Institute of Shipping Engineering, 1992[刘富全. 纠错编码技术及应用. 哈尔滨: 哈尔滨船舶工程学院出版社, 1992]

[6] Segars S. ARM7TDMI power consumption. IEEE Micro, 1997, 17(4): 12

[7] ASIC & System State Key Laboratory, Fudan University. CSMC06 CMOS standard cell library databook. Wuxi CSMC-HJ Co Ltd, August 2000: 131

[8] Zhang Xing, Wei Liqiong, Wang Yangyuan. Development of 1.5μm fully depleted CMOS/SIMOX gate array. Chinese Journal of Semiconductors, 1996, 17(6): 452(in Chinese)[张兴, 魏丽琼, 王阳元. 1.5μm 全耗尽 CMOS/SIMOX 门阵列的研制. 半导体学报, 1996, 17(6): 452]

VLSI Implementation of Modified Approximate Squaring Algorithm

Li Xia and Zhang Qianling

(*State Key Laboratory of ASIC & System, Fudan University, Shanghai* 200433, *China*)

Abstract: A modified approximate squaring algorithm, named as quadratic approximate algorithm, suitable for Viterbi algorithm is presented. The maximum relative error (MRE) and average relative error(ARE) of quadratic approximate algorithm are significantly improved by about 20% and 70%, respectively, comparing with the latest existing approach. Moreover, a 7-bit squaring function based on quadratic approximate algorithm is implemented using 0.6μm CMOS technology, and the timing delay and transistor counts are equivalent with the latest existing approach.

Key words: Viterbi algorithm; approximate squaring algorithm; PLA; MRE; ARE

EEACC: 2570D; 1265B

Article ID: 0253-4177(2003)05-0539-05

Li Xia male, was born in 1978, PhD candidate. His research interests include high-performance VLSI design.

Zhang Qianling female, was born in 1936, professor. Her research interests include the integration of VLSI system.

Received 27 June 2002, revised manuscript received 28 August 2002