

一种改进的基 4-Booth 编码流水线大数乘法器设计

周 怡, 李树国

(清华大学 微电子学研究所, 北京 100084)

摘 要: 大数乘法器是密码算法芯片的引擎, 它直接决定着密码芯片的性能. 由此提出了一种改进的基 4-Booth 编码方法来缩短 Booth 编码的延时, 并提出了一种三级流水线大数乘法器结构来完成 256 位大数乘法器的设计. 基于 SMIC0.18 μm 工艺, 对乘法器设计进行了综合, 乘法器的关键路径延时 3.77 ns, 它优于同类乘法器.

关键词: Booth 编码; wallace 压缩; 乘法器; 公钥密码运算

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)01-0060-04

Design of Pipeline Large Integer Multiplier Based on an Implementation of Radix-4 Modified Booth Encoding

ZHOU Yi, LI Shu-guo

(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

Abstract: A large integer multiplier is the engine of cryptography chip and determines the performance of cryptography chip. In this paper, we propose a new implementation of radix-4 modified Booth encoding, which has a shorter delay than methods proposed in previous works. Upon this encoding method, we also propose a new structure of 256-bit three stage pipeline multiplier. After synthesizing based on SMIC 0.18 μm CMOS process, the critical path delay of the multiplier is 3.77 ns, which is superior to that of other multipliers.

Key words: Booth encoding; Wallace tree; multiplier; public-key cryptographic

1 引言

大数乘法器广泛地应用于信息安全密码算法的运算, 它的性能直接决定着公钥密码算法 RSA 和 ECC 芯片的性能. 公钥密码算法都是建立在大整数运算基础上, 模约减、模乘、模幂运算被大量使用^[1], 而模乘、模幂运算的引擎就是大数乘法器. 乘法运算可以分为三个步骤: 部分积生成、部分积压缩至两行向量、最后将两行向量再相加. 在部分积生成中, 通常采用基 4-Booth 编码^[2]. 基 4-Booth 编码可使乘法器的部分积数量减少一半, 但部分积数量的减少也带来了编码延时的增加. 因此, 研究缩短 Booth 编码延时对乘法器整体性能的提升具有重要意义.

在改进的基 4-Booth 编码算法^[3-4]中, 乘数 B 按三位一组进行划分, 相互重叠一位, 编码形成五种部

分积选项. 在文献[3]的编码方法中引入编码值 $X_i, 2X_i, M_i$ 来区分表示部分积绝对值为一倍的被乘数、两倍的被乘数和负部分积. 结合编码值以及被乘数 A, 可以表示出部分积每一比特位的逻辑表达式:

$$p_i[j] = (A[j] \cdot X_i + A[j-1] \cdot 2X_i) \oplus M_i.$$

文献[4]中提到的编码方案是引入四个编码值, 来得到部分积的每一比特, 部分积每一比特位的逻辑表达式如下所示:

$$p_i[j] = (A[j]PL_i + (\sim A[j])M_i) \cdot X_i + (A[j-1]PL_i + (\sim A[j-1])M_i) \cdot 2X_i.$$

以上两种编码方法都是在原始的 Booth 编码基础上, 对编码重新进行改进. 依照上述两种方法, 经 ASIC 实现后, 第一种方法关键路径包含一个非门、三个与门、两个或门以及一个异或门; 第二种方法关键路径包括一个与非门、三个与门、两个或门. 这两

收稿日期: 2013-03-15; 修回日期: 2013-05-11

基金项目: 国家“八六三”计划(2012AA012402); 国家自然科学基金(61073173); 清华大学自主研发计划(2011Z05116)

种方法仍然存在着较长的路径延时,满足不了高性能乘法器的设计要求.为此,本文提出一种新的编码方法来降低 Booth 编码的延时.

2 一种新的基 4-Booth 编码实现方法

在分析文献[3-4]两种 Booth 编码方案的不足后,提出一种新的 Booth 编码方案.如表 1 所示,乘数 B 每相邻的三位共有八种组合方式,不同的组合形式分别代表部分积选择是 $0, \pm A, \pm 2A$ 中的一种,其中 A 代表被乘数.表中 pp 代表部分积(partial product).用编码值 X_i 等于 1 来表示绝对值是被乘数自身的组合方式,其余组合方式时令 X_i 值为 0;用编码值 M_i 等于 1 来表示部分积为负数的组合方式;同时,我们引入另一个编码值 $Modify_i$,它的值为 1 来指示绝对值非零的六种组合方式,我们称它为修正位.以上编码值中的 i 表示生成第 i 个部分积所需要的编码值序号,对 256 位乘法器来说,其范围是 $0 \sim 128$.

表 1 本文所提的基 4-Booth 编码方案

$B[2i+1]$	$B[2i]$	$B[2i-1]$	pp	X_i	M_i	$Modify_i$
0	0	0	0	0	0	0
0	0	1	+A	1	0	1
0	1	0	+A	1	0	1
0	1	1	+2A	0	0	1
1	0	0	-2A	0	1	1
1	0	1	-A	1	1	1
1	1	0	-A	1	1	1
1	1	1	0	0	1	0

经过分析和实验,得出了上述的表格中的逻辑关系.这时,编码值 $X_i, M_i, Modify_i$ 的逻辑表达式可概括为式(1)、式(2)和式(3),其中 $B[2i]$ 代表乘数的第 $2i$ 比特位.

$$X_i = B[2i-1] \oplus B[2i] \quad (1)$$

$$M_i = B[2i+1] \quad (2)$$

$$Modify_i = B[2i+1] \oplus B[2i-1] + B[2i] \oplus B[2i-1] \quad (3)$$

部分积每一比特位的生成如下述逻辑表达式所示:

$$p_i[j] = (A[j] \oplus M_i + (\sim X_i)) \cdot (A[j-1] \oplus M_i + X_i) \cdot (X_i + Modify_i) \quad (4)$$

经过验证,该表达式可正确生成所有的部分积.

根据以上编码逻辑关系以及部分积生成表达式,可以得到门级电路如图 1 所示.

在本文的 Booth 编码方案电路实现中,其关键

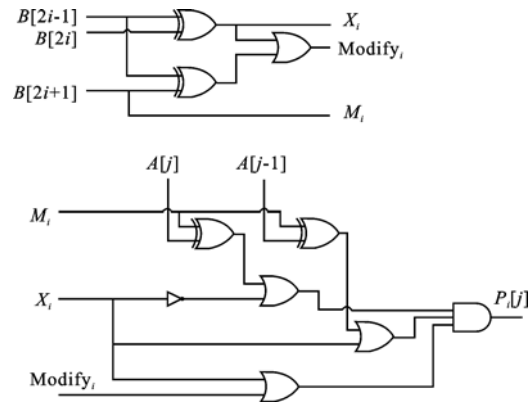


图 1 本文编解码方案的电路逻辑

路径包含一个异或门、两个或门、一个与门共四个门;而文献[3]中方案的关键路径包括一个非门、三个与门、两个或门以及一个异或门共七个门,文献[4]的方案包括一个与非门、三个与门、两个或门共六个门.

3 256 位三级流水线乘法器结构

大数乘法器的设计核心问题就是降低大数乘法器的关键路径时延.在分析了 Booth 编码、部分积阵列压缩的方案后,提出了一种三级流水线乘法器结构来实现 256 位大数乘法运算.其中:第一级包括 Booth 编码以及采用 4:2 压缩器^[5]和 3:2 压缩器的 wallace tree 混合压缩结构^[6],此级的输出部分积数量由 129 行压缩至 16 行;第二级继续混合使用 4:2 和 3:2 压缩器进行阵列压缩,此级输出部分积数量由 16 行压缩至 2 行;第三级使用超前进位加法器对两行部分积相加得最终一行部分积,即最终的乘法结果.其三级流水线乘法器结构如图 2 所示.

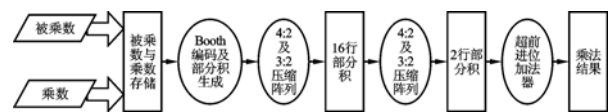


图 2 256 位三级流水线大数乘法器整体结构

在这样三级流水线结构中,第一级成为关键路径,其中 Booth 编码延时较大,在这级设计中,采用了本文提出的基 4-Booth 编码方案来减少 Booth 编码延时.Booth 编码生成部分积阵列如图 3 所示.图 3 中被乘数及乘数分别用 multiplicand A 和 multiplier B 表示,encoder 表示编码逻辑,即对乘数每相邻三位编码得到编码值 $X_i, M_i, Modify_i$; $Cell_{i,j}$ 代表生成第 i 个部分积的第 j 比特位的生成逻辑; p_i 代表第 i 个部分积,每个部分积是 256 位(不含符号位),共 129 个部分积.由于部分积中存在负的部分积,符号位扩展过程中会产生大量的符号位,占据多

余的硬件资源并增加了功耗. 为了实现减少资源消耗, 使用了符号扩展位消除技术^[7].

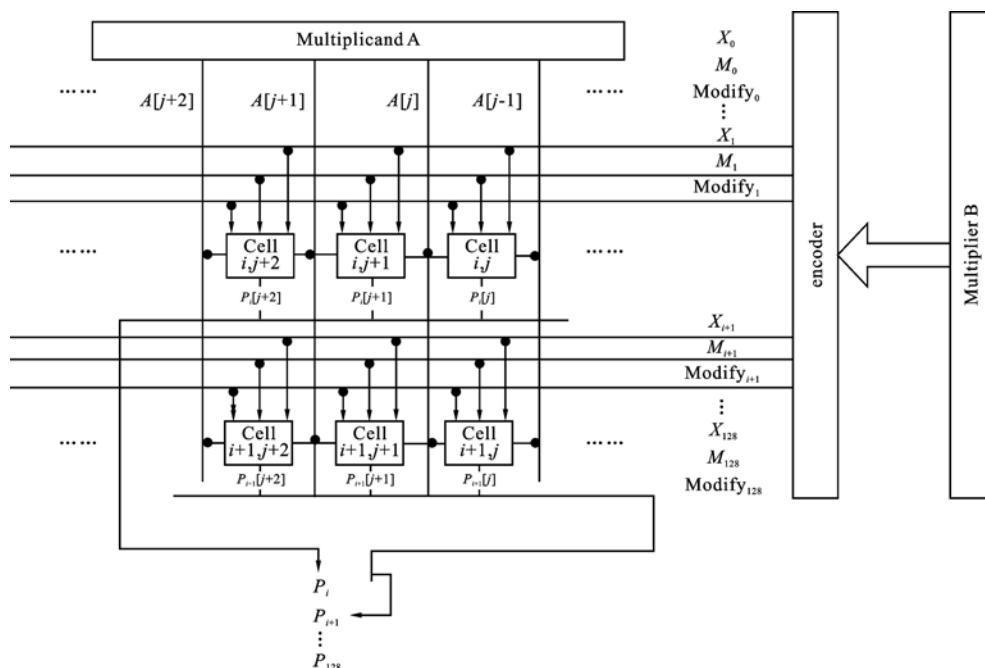


图 3 编码生成部分积结构

采用的 4:2 压缩器对 4 行每行位长为 n 位部分积进行压缩, 需要使用 n 个这样的 4:2 压缩器并行压缩组成. 其实现结构如图 4 所示. 压缩所得的两行向量 $S[n-1:0]$ 和 $C[n-1:0]$ 位长都是 n 位, 考虑到最终输出的 $Cout[n-1]$ 位, 可以用两个向量 $Sum = \{Cout[n-1], S[n-1:0]\}$ 以及 $Carry = \{C[n-1:0], 1'b0\}$ 来表示 4:2 压缩的结果, 即两个位宽 $n+1$ 的向量. 为了方便引用此模块作等效替换, 将图 4 所示的功能模块用图 5 等效地表示, 模块命名为“Module4_2”.

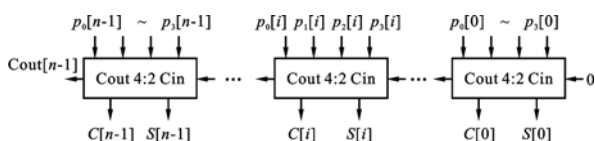


图 4 4 行压缩成 2 行的模块结构

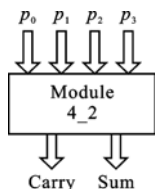


图 5 4 行压缩成 2 行模块的等效表示

同理可以构造 n 位长的 3 行压成 2 行模块, 命名为“Module3_2”. 对 Booth 编码生成的 129 行的部分积阵列进行树形压缩, 其压缩结构如图 6 所示.

第一层次压缩部分中, 129 行部分积分成 120

行和 9 行, 其中的 120 行分成 30 组“Module 4_2”压缩, 剩余的 9 行采用 3 组“Module 3_2”压缩; 如图 6 共分为 7 层压缩, 这 7 层压缩分在两级流水线中进行, 这样做是为了避免流水线的第一级形成更长关键路径, 以保持流水线的每级延时均衡. 以上方法采用 Cadence 仿真工具进行功能验证, 其结果正确.

4 实现结果及其比较

基于 SMIC0.18 μm 工艺库, 通过 Synopsys 综合工具 Design Compiler 对于本文的编码方案和 256 位大数乘法器的设计进行综合, 其编码时延 0.71 ns, 面积 2.08 mm^2 . 同时, 为了进行对比, 也对文献[3-4]的编码方案进行了实现. 三种编码方案分别应用于 256 位大数乘法器, 编码器的延时和面积见表 2, 从中可以看出本文所提的编码方案具有最短的时延. 基于本文所提出的编码方案及三级流水线乘法器完成的 256 位乘法器, 其性能得到一定程度的改进, 其乘法器时延 3.77 ns, 面积 6.25 mm^2 , 详细比较如表 3 所示.

表 2 三种不同 Booth 编码时延和面积

实现方案	延时/ns	面积/ mm^2
文献[3]	1.05	1.76
文献[4]	0.95	2.50
我们的方案	0.71	2.08

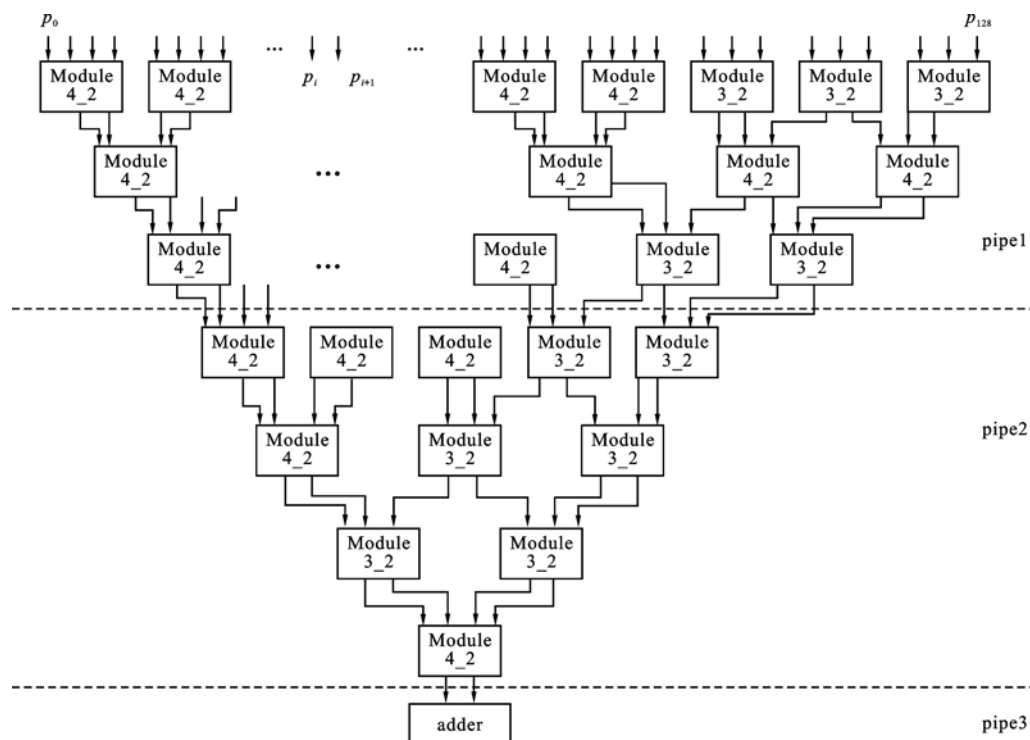


图 6 wallace tree 压缩阵列结构

表 3 不同方案下 256 位三级流水线大数乘法器性能比较

方案	Designware	文献 [8]	本文方案
工艺	SMIC0.18 μm	SMIC0.18 μm	SMIC0.18 μm
组合逻辑/ mm^2	5.06	5.59	5.86
时序逻辑/ mm^2	2.09	0.28	0.39
总体逻辑/ mm^2	7.16	5.86	6.25
延时/ns	4.46	4.67	3.77

5 结束语

经过综合和实验,文中提出的改进的基 4-Booth 编码方案与其他方案相比具有更短的延时性能,将此 Booth 编码方法用于 256 位三级流水线大数乘法器设计获得了满意的效果. 相比较于文献[8]中基 64-Booth 编码大数乘法器,本方法设计的大数乘法器关键路径延时缩短了 19.3%,硬件资源增加了 6.8%. 所以,这个设计方案在安全公钥密码算法中具有一定的实用价值.

参考文献:

- [1] 桑波. 大整数乘法算法的研究与快速实现[D]. 广州: 华南理工大学, 2012.
- [2] Booth A D. A signed binary multiplication technique [J]. Quarterly Journal of Mechanical and Applied Math, 1951, 4(2): 236-240.
- [3] Gensuke Goto, Tomio Sato, Masao Nakajima, et al. A 54 _ 54-b regularly structured tree multiplier [J]. IEEE Journal of Solid-State Circuits, 1992, 27 (9): 1229-1236.
- [4] Gensuke Goto, Ryoichi Ohe, Shin Mitarai, et al. A 4.1ns compact 54 _ 54b multiplier utilizing sign select booth encoders [J]. IEEE Journal of Solid-State Circuits, 1997, 32(11): 1676-1682.
- [5] Vojin G, Oklobdzija, David Villeger. Improving multiplier design by using improved column compression tree and optimized final adder in CMOS technology [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1995, 3(2): 292-301.
- [6] Wallace C S. A suggestion for a fast multiplier [J]. IEEE Transactions on Computers, 1964, 13 (2): 14-17.
- [7] Milos D. Ercegovac, Tomas Lang. Digital Arithmetic [M]. California, USA: Morgan Kaufmann Publishers, 2004.
- [8] Xiaodong Yan, Shuguo Li. Montgomery multiplier based on secondary booth encoded algorithm [C]// ASIC, 2007. ASICON'07. 7th International Conference on. China: Guangxi, IEEE, 2007: 197-200.

(下转第 67 页)

5 结束语

提出一种基于改进 SIFT 特征匹配的快速图像拼接算法,即在 SIFT 提取不变特征向量过程中加入多尺度 Harris 检测算子,使特征点更能体现图像中物体轮廓,减少特征点数量,提高运算速度.运用 PERANSANC 算法计算变换矩阵,可大量过滤不符合条件的模型参数,减少运行时间. KLT 跟踪算法可准确定位角点位置,得到精确的变换矩阵.实验结果表明本文所提出的算法可取得理想的图像拼接效果.

参考文献:

- [1] Harris C G, Stephens M. A combined corner and edge detector[C]// proceedings of the 4th Alvey Vision Conference. Manchester,1988:147-151.
- [2] Lowe D G. Distinctive image features from scale-in-

variant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

- [3] 邵巍,朱圣英.一种快速多尺度特征点匹配算法[J].中国图象图形学报,2009,12(14):2572-2576.
- [4] Brown M, Lowe D. Invariant features from interest point groups[C]// Proceedings of the 13th British Machine Vision Conference. Cardiff,2002:253-262.
- [5] 陈付幸.基于预检验的快速随机抽样一致性算法[J].软件学报,2005,16(8):431-443.
- [6] 尚明姝.基于局域熵的全自动图像拼接算法[J].微电子学与计算机,2013,30(1):65-68.
- [7] 邓荣峰,李熙莹.基于 SIFT 特征匹配的稳健图像拼接算法[J].计算机应用,2009,29(6):19-221.

作者简介:

尚明姝 女,(1977—),硕士,实验师.研究方向为图像处理与计算机视觉.

(上接第 63 页)

作者简介:



周 怡 男,(1987—),硕士.研究方向为信息安全算法的大规模数字集成电路设计与实现.



李树国 男,(1963—),教授,博士生导师.研究方向为信息安全算法的大规模数字集成电路设计与实现.