

基于 Booth 算法的 32 位流水线型乘法器设计

翟召岳, 韩志刚

(同济大学 电子科学与技术系, 上海 201804)

摘 要: 为了减少乘法指令在保留站中的等待时间, 设计了一款 32 位流水线型乘法器, 该乘法器将应用于作者设计的一款超标量处理器中. 该乘法器应用了改进型的 booth 编码算法, 对部分积生成电路进行了优化, 并采用了 4-2 压缩器与 3-2 压缩器相结合的 Wallace 树型结构对部分积进行压缩, 最后再根据各级的延迟, 在电路中插入了流水线寄存器, 使其运算速度得到了提高. 该乘法器使用 GSMC 0.18 μm 工艺进行综合. 经过仿真验证, 该乘法器大大减少了在保留站中等待执行的乘法指令的完成时间, 使每个时钟周期都有一条新的乘法指令被发送至乘法器进行运算.

关键词: Booth 算法; Wallace 树; 压缩器; 流水线

中图分类号: TP332.2

文献标识码: A

文章编号: 1000-7180(2014)03-0146-04

32 Bit Pipeline Multiplier Design Based on Booth Encoder

ZHAI Zhao-yue, HAN Zhi-gang

(Department of Electronic Science and Technology, Tongji University, Shanghai 201804, China)

Abstract: In order to reduce the waiting time of the multiplication instructions in reservation, a 32-bit pipelined multiplier is designed in this paper. It will be applied to the design of a superscalar processor. In this design, a modified booth encoder is used and partial product generator circuit is optimized. The partial product is compressed by the combination with 4-2 compressor and 3-2 compressor in Wallace tree structure. At last, pipeline registers are inserted according to the delay of the several levels, which will accelerate computing speed. The multiplier is synthesized in GSMC 0.18 μm process. According to simulation and verification, this multiplier reduce the complement time of the IR waiting to execute in reservation stations. A new multiplication is sent to multiplier to compute in every cycle.

Key words: Booth algorithm; Wallace tree; compressor; pipeline

1 引言

在处理器设计中, 乘法器的性能将对整个处理器的性能产生制约. 由于乘法器执行一条指令所需时间较长, 因此会使较多的指令暂存于保留站中等待操作数. 本文设计了一款乘法器, 采用改进型的 Booth 算法减少部分积个数, 使用 4-2 与 3-2 压缩器相结合的 Wallace 树型结构加速部分积的压缩, 最后根据各部分的延迟, 加入流水线寄存器. 该乘法器的结构为: 数据输入 $\Rightarrow$ Booth 编码 $\Rightarrow$ 部分积生成电路 $\Rightarrow$ 压缩器 $\Rightarrow$ 求和.

2 Booth 编码

传统型 booth 编码采用基 2 的形式:

$$P = A \times B = A \times \left(\sum (-x_i + x_{i+1}) \times 2^i \right),$$
 对于 n 位的乘数 B , 产生 n 个部分积.

改进型 booth 编码采用基 4 的形式:

$$P = A \times B = A \times \left(\sum (-2x_{2i+1} + x_{2i} + x_{2i-1}) \times 4^i \right),$$
 对于 n 位的乘数 B , 产生 $n/2$ 个部分积, 因此与传统型 booth 编码相比, 运算效率将更加高效.

其他编码方式如基 8 booth 编码, 对于 n 位的乘数 B , 产生 $n/3$ 个部分积, 部分积数目较 booth 基

4 编码产生的部分积数目少,但对于被乘数 A 的操作中,出现了 $\pm 3A$,无法使部分积通过移位等简单方式实现,因此这将使部分积的生成电路变得较为复杂.综合考虑,本文中选择了基 4 booth 编码的方式来生成部分积.

3 部分积生成电路

采用基 4 booth 编码的编码方式及其对应控制信号如表 1 所示.

表 1 改进型基 4 booth 编码及其控制信号

$b_{2n+1}b_{2n}b_{2n-1}$	PP_n	Neg	B_{1_h}	B_{2_l}
000	$+0 * A2^{2n}$	0	0	1
001	$+1 * A2^{2n}$	0	1	0
010	$+1 * A2^{2n}$	0	1	0
011	$+2 * A2^{2n}$	0	0	1
100	$-2 * A2^{2n}$	1	0	1
101	$-1 * A2^{2n}$	1	1	0
110	$-1 * A2^{2n}$	1	1	0
111	$-0 * A2^{2n}$	1	0	1

其中 $+1$ 与 $+2$ 分别表示被乘数的 1 倍与 2 倍, -1 与 -2 分别表示被乘数 1 倍并求反加 1、被乘数 2 倍并求反加 1.

在文献[1]中,编码实现电路及部分积生成电路采用图 1 的形式.

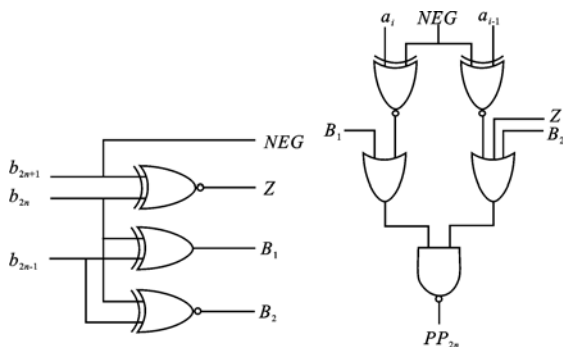


图 1 文献[1]编码单元实现方式与部分积生成电路单元

在本文中,使用的编码方式与部分积生成电路如图 2 所示.

与文献[1]中的实现方式相比较可知,本文中使用的编码方式(图 2)中简化了一个异或非门,同时,在部分积生成电路单元中,改由两个与门、一个或门和一个异或门来实现.与文献[1]中相比,总体上简化了一个异或非门.由于编码单元与部分积生成电路单元使用次数较多,因此,这一改进将会优化电路的版图面积.

此外,对于部分积为 $-2 * A * 2^{2n}$ 或 $-1 * A$

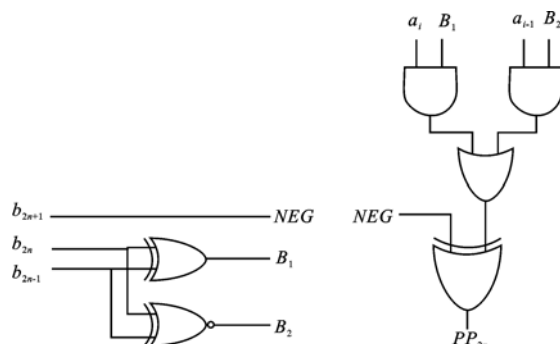


图 2 本文编码单元实现方式与使用的部分积生成电路单元 $* 2^{2n}$ 的形式,需要对被乘数所有位数取反加 1,对于其中的加 1 这一操作,如果在被乘数取反之后立马进行,则会增加部分积生成过程中的延迟,影响运算速度.因此,本文中加 1 的这一操作保留至下一个部分积的 0 位上去,如图 3 所示,如果 pp0 需要取反加 1,则 $pp0 = \sim A$; $pp1[0] = 1'b1$;其中 $pp1[0]$ 对应于 $pp0[0]$. 即如果部分积生成中需要取反加 1,则 $c=1$,不需要则 $c=0$. 通过以上方法,在部分积相加过程中,加 1 操作将会完成,以此可以减小部分积生成过程中的延迟.

11111111XXXXXX	pp0	yXXXXXXXX	pp0
111111XXXXXX c	pp1	yXXXXXXXX c	pp1
1111XXXXXX c	pp2	yXXXXXXXX c	pp2
1XXXXXX c	pp3	yXXXXXXXX c	pp3
c		10101011 c	pp4

图 3 部分积符号扩展示意图

4 符号扩展问题

乘法器中符号扩展分为乘数的符号扩展和部分积的符号扩展两部分.

(1)乘数的符号扩展:为将带符号数间的乘法与不带符号数间的乘法区别开来,需要对乘数的符号位进行扩展.假设被乘数为 $A[31:0]$,乘数为 $B[31:0]$,如果为带符号数的乘法,则需要对乘数 B 的符号位进行扩展;扩展两位则可以满足基 4 booth 算法对位数的要求,即: $B[33] = B[32] = B[31]$; 如果为不带符号数的乘法,则 $B[33] = B[32] = 1'b0$.

(2)部分积的符号扩展:由于需要将乘数也进行符号扩展,即在被乘数的高位补两位符号位,因此部分积中的数据即为补码形式.扩展的符号位在部分积的加法运算中要消耗相当大的硬件资源,因此,将对部分积的符号位再次进行扩展,以 8×8 产生的四个部分积为例,如图 3 所示.

即将部分积 pp0 从[7:0]扩展到[15:0],[15:8]全部为 1, pp1、pp2、pp3 与 pp0 类同. 如果将此扩展的部分积直接相加, 也将消耗大量资源, 由于扩展的值全部为固定值 1, 因此可以将所有的 1 相加, 将结果作为部分积 pp4, 改动后由图 3 左变为图 3 右. 对于 32×32 的乘法器来说, 部分积的符号扩展部分的总和 pp17 为

pp17 = $(2^34 - 1) - (2^0 - 1) + (2^34 - 1) - (2^2 - 1) + (2^34 - 1) - (2^4 - 1) + \dots + (2^34 - 1) - (2^{32} - 1)$, 由于运算结果为 64 位, 则取低 32 位即可, 即:

pp17 =
32'b10101010101010101010101010101011.

由于 pp17 与 pp16 是对齐的, 因此需要对 pp16 中传给 pp17 的 c 位做特殊处理, 由于 pp17 为固定值, 因此 pp17 的高 29 位仍为固定值, 低 3 位根据 c 的值选择是 3'b100 还是 3'b011. 由于符号扩展全部为 1, 即表示各个部分积为负值, 因此如果生成的部分积为正值, 则令图 3 中的 $y=1$ 即可, 如果生成的部分积为负值, 则 $y=0$.

因此对于 32×32 的乘法器, 符号扩展后增加了两个部分积, 使部分积的总数变为 18 个, 由于 c 和 y 的加入, 除部分积 pp0、pp16 为 33 位、pp17 为 32 位之外, 其他部分积都为 34 位. 对于 pp16 与 pp17 的处理可以节省版图面积.

5 部分积相加

在本文中, 部分积相加部分主要考虑部分积的个数以及使用的压缩器的延迟. 在部分积的相加过程中, 通常使用 3-2 压缩器与 4-2 压缩器. 其中 3-2 压缩器即带进位的一位全加器, 4-2 压缩器可以使用两个 3-2 压缩器实现, 如图 4 左所示. 对于图 4 左的结构, 可以根据其逻辑表达式进行化简, 使得:

$$\text{Carry} = (A \cdot B \cdot C \cdot D) \& \text{Cin} + (\sim(A \cdot B \cdot C \cdot D)) \& D$$

$$\text{Cout} = (A \cdot B) \& C + (\sim(A \cdot B)) \& A$$

改进后的 4-2 压缩器如图 4 右所示.

由于该乘法器中部分积总共有 18 个, 因此如果像文献[2]中完全采用 4-2 压缩器对部分积进行相加运算的话, 需要先对其中的 16 个部分积进行相加运算, 通过 3 级 4-2 压缩器得到 1 个伪和、1 个伪进位的结果, 再将这两个结果与剩余的 2 个部分积通过 4-2 压缩器进行相加, 得到最终的 cout 与 sum, 将 cout 与 sum 通过超前进位加法器进行相加, 则可以得到最终的结果. 采用这种方法在部分积相加的过

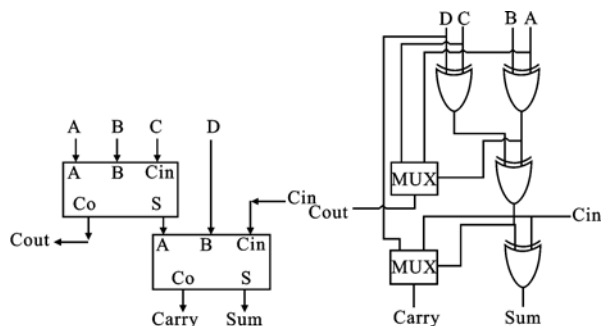


图 4 由 3-2 压缩器构成的 4-2 压缩器与改进后的 4-2 压缩器

程中需要经过 4 级 4-2 压缩器相加.

在本文中, 考虑到有 18 个部分积, 4-2 压缩器不能刚好完全压缩, 因此第一级采用 3-2 压缩器, 刚好使 18 个部分积完全压缩, 并产生 12 个中间结果, 再通过一级 4-2 压缩器与一级 3-2 压缩器后得到 4 个中间结果, 再通过最后一级 4-2 压缩器后, 即得到两个中间结果, 最后通过超前进位加法器, 即可得到最后的运算结果, 部分积相加电路结构采用 Wallace 树型结构如图 5 所示. 对应位少于 3 个数, 采用 1 位全加器或直接连线来实现, 以此减小部分积压缩电路的面积.

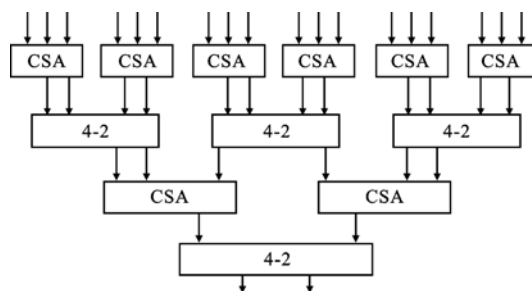


图 5 部分积相加电路结构图

6 流水线结构的实现

由于本文的乘法器是所设计的超标量处理器中的一部分, 为满足运算结果的需求时间与保留站个数的限制, 在该乘法器中加入流水线结构. 流水线插入级数的数目根据电路的延迟与以上两个条件来决定. 在 GSMC 0.18 μm 工艺下, 使用 Synopsys DC 进行综合, 在不添加最后一级加法器的情况下, 该乘法器 worst delay 为 6.02 ns, best delay 为 2.53 ns, 加入非流水线结构的超前进位加法器后, 得到的 worst delay 为 23.83 ns, best delay 为 9.950 ns, 可见最后一级加法器的延迟对整个乘法器的性能有很大影响. 根据之前测得的延迟数据, 插入三级流水线寄存器, 构成四级流水线结构, 这样将大大减少在保留站中等待的指令的完成时间, 综合后的总面积为

107 702.18 μm^2 ,其中组合电路面积为89 340.45 μm^2 .与文献[3]相比,考虑到后续的布局布线,该结果中组合电路的面积仍较为理想.使用Modelsim对乘法器进行仿真,结果如图6所示,可以看出在乘数与被乘数输入四个周期之后,即可得到输出结果.如果在保留站中存有多条乘法指令需要执行,采用流水线结构之后,则可以每个时钟周期执行一条新的指令,这将大大提高保留站中的指令完成速度,对处理器的性能提升有很大帮助.



图6 采用四级流水线结构的乘法器仿真结果

该乘法器使用Xilinx ISE10.1的Spartan 3E的FPGA模型进行了综合验证,在不加流水线结构的情况下,该乘法器的最大延迟为25.288 ns,其中布线延迟占到近55%.与文献[4]相比较,性能大幅提高,在加入流水线结构之后该乘法器的时钟频率可以最高达200.08 MHz.考虑到FPGA的lut单元内部的门在编译过程中不能完全利用,对布局布线将会产生影响,因此如果将该乘法器在ASIC上实现,则性能又将会获得提升.

7 结束语

本文对乘法器各级电路结构进行了优化,与其他文献中设计方法相比,减少了乘法器各级的延迟与面积.根据综合结果中的延迟信息,在对应电路结构中插入了流水线寄存器,使该乘法器加速了保留站中等待指令的完成速度,其延迟、面积与功耗满足

了处理器中的设计要求.在本文的设计中,最后一级加法器的延迟仍然较大,对该加法器的优化将是本课题未来的研究重点.

参考文献:

- [1] 李军强,李东生,李奕磊,等. 32×32 高速乘法器的设计与实现[J]. 微电子学与计算机, 2009, 26(12): 23-26.
- [2] 常静波,郭立. 一种3级流水线Wallace树压缩器的硬件设计[J]. 微电子学与计算机, 2005, 22(1): 160-162.
- [3] 许琪,原巍,沈绪榜,等. 一种新的树型乘法器的设计[J]. 西安电子科技大学学报:自然科学版, 2002, 29(5): 580-583.
- [4] 栾玉霞,李存志. 32×32 乘法器的一种设计[J]. 西安电子科技大学学报:自然科学版, 2004, 31(1): 16-20.
- [5] Yeh W C, Jen C W. High-speed booth encoded parallel multiplier design[J]. IEEE Transactions on Computers, 2000, 49(7): 692-701.
- [6] David A Patterson, John L Hennessy. 计算机体系结构—量化研究方法[M]. 3版. 北京: 电子工业出版社, 2004.
- [7] 陈海民,李峥,谢铁顿,等. 基于Radix-4Booth编码的乘法器优化设计[J]. 计算机工程, 2012, 38(1): 233-235.

作者简介:

翟召岳 男, (1988—), 硕士研究生. 研究方向为计算机体系结构、专用集成电路设计.

韩志刚 男, (1972—), 副教授. 研究方向为电力电子及电源管理集成电路设计、模拟集成电路设计、数模混合集成电路设计.

(上接第145页)

- [6] Farrahi M R, Mohamed C. A multi-scale framework for adaptive binarization of degraded [J]. Pattern Recognition, 2010(10): 1016-1028.
- [7] Kamel M, Zhao A G. Extraction of binary character/graphics images from grayscale document images [J]. Graphics Models and Image Processing, 1993, 55(3): 203-217.
- [8] Yang Y B, Yan H. An adaptive logical method for binarization of degraded document images [J]. Pattern Recognition, 2000, 33(5): 787-807.
- [9] Zhao M S. Image thresholding technique based on fuzzy partition and entropy maximization[D]. University of Sydney, Doctor of Philosophy. 2004.
- [10] Ntirogiannis K, Gatos B, Pratikakis I. A modified adaptive logical level binarization technique for historical document images [C]//Proc of 10th International Conference on Document Analysis and Recognition. Barcelona: IEEE, 2009: 1171-1175.
- [11] 谢志民. 中国女书作品的文体特点及其分类[J]. 世界文学评论, 2007, 65(1): 250-254.

作者简介:

孙阳光 男, (1978—), 博士, 讲师. 研究方向为图像处理、机器视觉、目标检测与识别.

蔡志华 男, (1988—), 硕士研究生. 研究方向为图像处理和目标检测.