基于 Radix-4 Booth 编码的并行乘法器设计

范文兵, 周健章

(郑州大学 电气与信息工程学院,河南 郑州 450001)

摘　要:速度和面积是评价乘法器单元性能优劣的两个基本指标。针对当前乘法器设计难以平衡版图面积和传输延时的问题,采用 Radix-4 Booth 算法,设计了一种新型的 16 位有符号定点乘法器。在部分积生成过程中,首先改进对乘数的取补码电路,然后优化基数为 4 的改进 Booth 编码器和解码器,此结构采用较少的逻辑门资源,并且易对输入比特进行并行化处理。在 Wallace 压缩电路中,对符号扩展位进行预处理并设计新的压缩器结构,优化整个 Wallace 压缩模块。在第二级压缩过程中提前对高位使用纹波进位加法器结构计算,减小了多 bit 伪和的求和位数。在求和电路中,使用两级超前进位加法器结构,在缩短关键路径传输延时的同时避免增大芯片面积,提高了乘法器的运行速度。新型定点乘法器与已有的乘法器结构相比,减少了 12.0%的面积,降低了 20.5%的延时。

关键词:Radix-4 Booth 编码; 面积; 传输延时; 编码器; 解码器; Wallace 压缩

中图分类号:TN402　　**文献标志码**:A　　**doi**:10.13705/j.issn.1671-6833.2024.04.011

随着 OpenAI 公司的 ChatGPT(chat generative pre-trained transformer)、百度的文心一言以及复旦大学的 MOSS 等各种 AI 模型平台的陆续推出,人工智能话题迅速成为广泛关注的焦点。这些模型平台的出现是基于各种结构神经网络的广泛应用。随着对 AI 模型智能化要求的提高,各种 AI 模型任务的复杂度也随之提高,其中神经网络中的乘加单元运算量也不断增多。目前,AlexNet 网络模型训练的乘法运算量已经达到 9×10^8 次[1]。同时乘法器也是数字信号处理器(DSP)的重要组成部分,其运行速度极大地影响了处理器的运行速度。因此降低乘法器的延时和减小其芯片面积至关重要。

Radix-4 Booth 编码乘法器算法的核心步骤是部分积生成、部分积压缩和最终求和电路。部分积生成和压缩主要影响着乘法器的版图面积,最终求和电路则影响着乘法器的工作速度。在部分积生成方面[2],使用 Radix-4、Radix-8、Radix-16 等高基编码来减小部分积的数量。在部分积压缩阶段,Wallace 压缩方式是利用保留进位加法器构成树状结构对部分积阵列压缩处理,这种压缩方式通过设计不同结构压缩器来提高压缩速率[3],在最终求和部分,使用快速进位加法器求和来提高乘法器的运行速度。姚若河等[4]在部分积生成方面,采用基数更大的 Radix-16、Radix-32 和 Radix-64 编码使得部分积的个数降低,但基数更大的编码算法使得编码器电路复杂度增加,进而需要在解码器电路中引入更多的组合逻辑电路,降低了乘法器的运行速度。Scheunemann 等[5]在 Wallace 压缩阶段,使用 5 级进位保留加法器(carry save adder,CSA),增大了压缩模块的延时。Jain 等[6]在最终求和部分,使用纹波进位加法器(ripple carry adder,RCA)结构,减小了面积,但因为多 bit 伪和的求和电路位数是 32 位,其门传输延时是 65 级,此加法器结构降低了乘法器的工作速度。Moni 等[7]采用 Dadda 树压缩方式,分别将 16 位乘数和被乘数分为高 8 位和低 8 位,采用阵列压缩的方式,降低了压缩阵列的延时,但是增大了此模块的面积。

本文综合考虑了面积和延时两方面因素,提出了一种新型的 16 位定点乘法器。本文提出的 Radix-4 Booth 算法乘法器结构有以下 3 点改进。

(1)提出了一种新型的取补码电路,以硬件资源少的方式实现被乘数补码电路的生成。并通过改进

收稿日期:2024-01-28;**修订日期**:2024-03-26

基金项目:河南省科技攻关项目(192102210086)

作者简介:范文兵(1969—),男,河南周口人,郑州大学教授,博士,博士生导师,主要从事混合集成电路分析与设计、图像处理与射频识别技术的研究,E-mail:iewbfan@zzu.edu.cn。

编码器和解码器电路,用 3 个 flag 信号控制产生部分积,减少了部分积的个数。此模块在减小乘法器面积的同时,降低了部分积生成模块的门传输延时。

(2)在 Wallace 压缩结构中仅使用了两级压缩结构。在压缩电路中,灵活采用改进的 4-2 压缩器和 CSA 结构。并对扩展符号位进行预处理和在第二级 Wallace 压缩中提前进行高位求和运算,缩短了最后求和电路的关键路径。

(3)提出了一种新型的快速求和电路,进一步优化电路关键路径延时,进而提高乘法器的工作速度。

1 乘法器整体设计

为了使部分积阵列易于压缩,在不增加乘法器延时的前提下,本文提出两种乘法器结构方案。方案 1 为纯组合逻辑电路组成的乘法器(multipliers composed of pure combinational logic circuits, MCL),如图 1(a)所示。MCL 乘法器结构最大限度地减小乘法器面积,并通过新型求和电路缩短门电路传输延时,进而提高乘法器的运行速度。方案 2 为带有时序逻辑的乘法器(multiplier with sequential logic, MSL),MSL 的单元结构是两级流水线,用于增加乘法器单元的吞吐量,如图 1(b)所示。$P_1 \sim P_8$ 为编码器和解码器电路生成的 8 个部分积。在部分积生成单元中引入了流水线寄存器的第一个阶段,并在部分积压缩阶段和最终求和阶段(C 和 S)之间增加了另一个流水线阶段。对比 MCL 和 MSL 两种乘法器结构,除了 MSL 增加流水线寄存器以外,其他结构均相同。

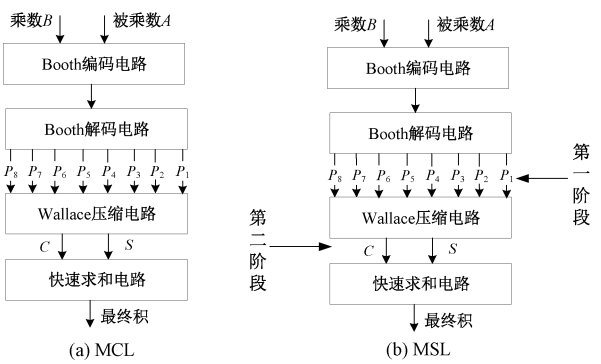

图 1 MCL 和 MSL 乘法器结构图

Figure 1 Schematic diagram of MCL and MSL multiplier

2 部分积生成

2.1 传统的编码电路和解码电路

Radix-4 Booth 编码算法规则是乘数最低位补零,然后依次从低到高检测乘数的 3 位数,以 $A \times B$ 两个 16 位有符号定点数进行举例说明,乘数 B 的 3 位组合如图 2 所示。

图 2 乘数 B 组合示意图

Figure 2 Schematic diagram of multiplier B combination

在图 2 中,$b_{-1} = 0$。相邻检测 3 位之间重叠 1 位,通过编码器电路,将部分积从 16 个减少为 8 个,从而提升了乘法器的运算速度。将 16 位有符号二进制数 B 转化为其对应的十进制如式(1)所示,A 与 B 的乘积可以表示为式(2)。

$$B = -2 \times 2^{14} b_{15} + \sum_{n=0}^{14} 2^{n-1} b_n + b_{-1}。 \tag{1}$$

$$\begin{aligned} A \times B &= A \times \sum_{n=0}^{7} (-2b_{2n+1} + b_{2n} + b_{2n-1}) \times 2^{2n} \\ &= A \times \sum_{n=0}^{7} Y_n \times 2^{2n}。 \end{aligned} \tag{2}$$

根据式(2)可以得到多项式 Y 的真值表和部分积的关系,如表 1 所示。

表 1 多项式 Y 的真值表

Table 1 Truth table of polynomial Y

$b_{2n+1}b_{2n}b_{2n-1}$	Y	部分积	$b_{2n+1}b_{2n}b_{2n-1}$	Y	部分积
000	0	0	100	−2	$-2A$
001	1	A	101	−1	$-A$
010	1	A	110	−1	$-A$
011	2	$2A$	111	0	0

通过表 1 可知,对乘数 B 进行编码可以得到 5 种部分积的结果,当 $Y=0$ 时,则对被乘数 A 每 1 位都置零;当 $Y=1$ 时,则对被乘数 A 保持不变;当 $Y=-1$ 时,则对被乘数 A 每 1 位都取反加 1;当 $Y=2$ 时,则对被乘数 A 左移 1 位,并最低位补零;当 $Y=-2$ 时,则对被乘数 A 左移 1 位后,再取反加 1。

2.2 改进的编码电路和解码电路

2.2.1 改进的取补码电路

本文改进的被乘数取补码电路使用新的方法通过化简卡诺图得到 $-A$ 的第 i 位进位逻辑表达式如式(3)所示。

$$c_i = a_i \oplus (a_{i-1} + a_{i-2} + a_{i-3} + \cdots + a_1 + a_0)。 \tag{3}$$

式中:c_i 为被乘数 A 第 i 位向 $i+1$ 位的进位;⊕表示异或门。

$-A$ 可以用 c_i 表示,其逻辑表达式如式(4)所示。

$$-A[i:0] = c_i c_{i-1} \cdots c_1 c_0。 \tag{4}$$

本文以4位二进制数 X 来求出 $-X$ 为例进行对比说明,如图3所示。图3(a)是传统半加器求解 $-X$ 的电路,图3(b)是采用改进的方法。其中每个半加器使用1个异或门和1个与门,所以传统方法一共使用了4个非门,4个与门,4个异或门。图3(b)采用了2个或门,3个异或门。由于与门和或门在电路中的面积和延时基本相同,所以这种改进的求 $-X$ 的电路节省了4个非门,2个或门和1个异或门。由此将这个方法用到16位求补码电路中,可以节省17个非门,2个或门和1个异或门。

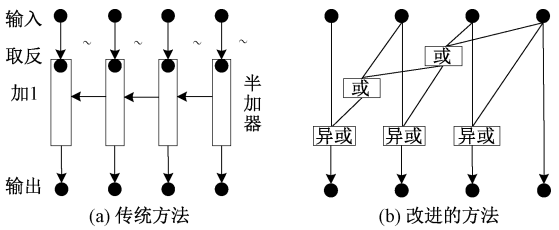

图3 取补码电路

Figure 3 Complementary code taking circuit

2.2.2 改进的编码器电路

改进的编码器电路需要对多项式 Y 真值表进行重新编码,如表2所示。

表2 改进的编码器编码表

Table 2 Improved encoder coding table

Y	部分积	F_{2A}	F_{-A}	F_A
0	0	0	0	0
1	A	0	0	1
1	A	0	0	1
2	$2A$	1	0	1
-2	$-2A$	1	1	0
-1	$-A$	0	1	0
-1	$-A$	0	1	0
0	0	0	0	0

根据表2可以得出新的3个flag(F_A, F_{-A}, F_{2A})信号逻辑表达式如式(5)所示。

$$\begin{cases} F_A = \overline{b_{2n+1}}(b_{2n} + b_{2n-1}); \\ F_{-A} = \overline{\overline{b_{2n+1}} + b_{2n}b_{2n-1}}; \\ F_{2A} = \overline{(b_{2n} \oplus b_{2n-1})}。 \end{cases} \tag{5}$$

根据逻辑表达式5可以得到实现3个flag信号的控制电路如图4所示。

在图4中,改进的编码器电路使用了3个非门、4个与(或)门和1个异或门。与文献[5]和文献[8]相比,编码器电路门电路个数上基本相同。

2.2.3 改进的解码器电路

通过改进的编码器电路得到解码器电路如

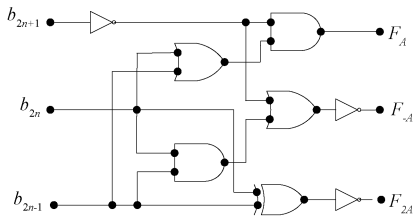

图4 改进的编码器电路

Figure 4 Improved encoder circuit

图5所示。在图5中,由 F_A 和 F_{-A} 来选择生成部分积是保持乘数 A 还是取补码后的 $-A$,接着通过第二个多路选择器的选通端 F_{2A} 选出是否需要左移的数据。

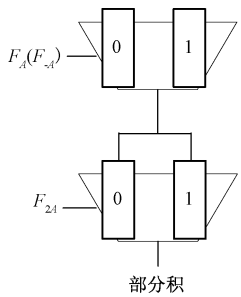

图5 改进的解码器电路

Figure 5 Improved decoder circuit

将本文部分积生成模块的门电路个数和总的门传输延时与文献[5]和文献[8]进行对比,得到表3。

表3 部分积生成模块对比

Table 3 Comparison of partial product generation modules

乘法器结构	门个数	门传输延时
文献[5]	4个多路选择器	6级与(或)门
文献[8]	3个多路选择器	7级与(或)门
本文设计	2个多路选择器	6级与(或)门

在表3中,部分积生成模块中编码器门个数基本相同,解码器中门个数对芯片面积影响更为重要。因此,本文只比较解码器结构中所使用的多路选择器数量。

本文与文献[5]相比,门电路的传输延时相同,但是少了2个多路选择器;与文献[8]相比,少了1个多路选择器和少了1级门传输延时。因为部分积是18位的,所以解码器电路中的1个多路选择器对应18个二输入的选择器,又因为Radix-4 Booth算法生成的8个部分积每一个都需要经过编码和解码模块,所以减少1个多路选择器带来的是编解码电路面积成倍缩小。因此,本文改进的编解码电路不仅减小了此模块的面积开销,而且提高了此模块的运行速度。

详细的电路设计以生成 P_1 为例进行说明,如图

6 所示。在图 6 中,F_A 和 F_{-A} 通一级与或门选出是否需要对被乘数取补码的 17 位中间值 M,当编码电路 F_A 和 F_{-A} 信号同时为 0 时,通过 2 级与(或)门将 $M[16:0]$全部置零,然后将通过 F_{2A} 信号得到的 $P_1[17:0]$也全部置零。当 $F_A=1$、$F_{-A}=0$ 时,$M[16:0]$保持被乘数 A 数据本体。接着通过第二级多路选择器,当 $F_{2A}=1$ 时,代表左移有效,$P_1[16:1]=M[15:0]$;当 $F_{2A}=0$ 时,代表左移无效,$P_1[16:1]=M[16:1]$;因为是有符号数,$P_1[17]$是部分积的符号位,所以无论左移信号 F_{2A} 是否有效,$P_1[17]=M[16]$。对于 $P_1[0]$,当 $F_{2A}=1$ 时,$P_1[0]=0$;当 $F_{2A}=0$ 时,$P_1[0]=M[0]$。

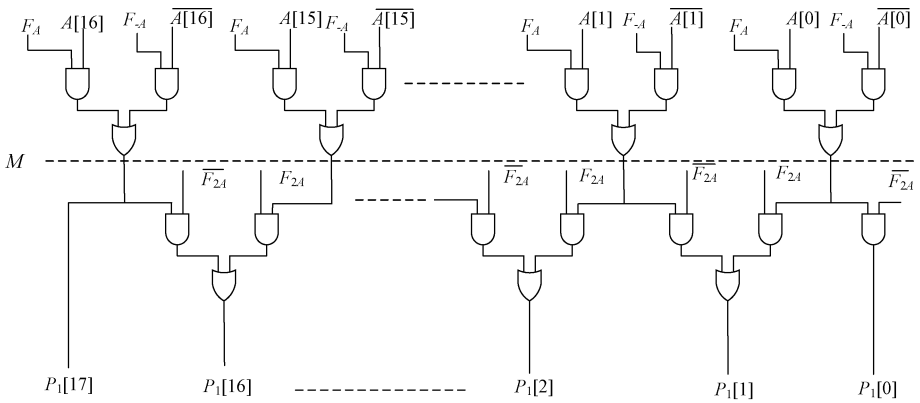

图 6 P_1 部分积生成示意图

Figure 6 Schematic diagram of P_1 partial product generation

3 Wallace 压缩电路

由于 Wallace 压缩模块并行性很高,所以在此模块主要考虑如何减小版图面积,也要避免增大门传输延时。本文主要从两个方面进行改进:一方面通过设计不同的压缩器进而使用混合压缩的结构对生成的 8 个部分积压缩;另一方面要对过多的符号扩展位进行预处理,从而减少使用压缩器个数,达到减小面积的目的。

3.1 传统的 Wallace 压缩模块

3.1.1 传统的压缩器结构

Wallace 压缩模块是由 4-2 压缩器、CSA 和半加器组合而成,4-2 压缩器是由两个 CSA 组合得到的,其结构如图 7 所示。

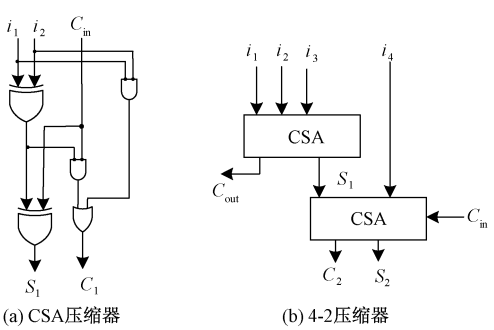

图 7 传统的 CSA 和 4-2 压缩器

Figure 7 Traditional CSA and 4-2 compressor

图 7(a)为 CSA 电路结构图,图 7(b)为 2 个 CSA 级联得到的 4-2 压缩器电路图。CSA 使用了 2 个异或门和 3 个与门,最长门传输延时为 2 级异或门。4-2 压缩器的工作原理是:i_1、i_2、i_3 进入第一级 CSA,得到 C_{out} 和 S_1,然后将 S_1 和 i_4 和 C_{in} 送入第二级 CSA,得到 C_2 和 S_2。这种 4-2 压缩器使用门的数量是 CSA 结构的两倍,最长门传输延时为 4 级异或门。

由图 7 可以得到 CSA 结构的 C_1 和 S_1 的逻辑表达式如式(6)所示。

$$\begin{cases} C_1 = i_1 i_2 + C_{in}(i_1 \oplus i_2); \\ S_1 = i_1 \oplus i_2 \oplus C_{in}。 \end{cases} \tag{6}$$

由 CSA 级联得到的 4-2 压缩器的 C_{out}、C_2 和 S_2 的逻辑表达式如式(7)所示。

$$\begin{cases} C_{out} = i_1 i_2 + i_3(i_1 \oplus i_2); \\ S_1 = i_1 \oplus i_2 \oplus i_3; \\ C_2 = i_4 S_1 + C_{in}(i_4 \oplus S_1); \\ S_2 = i_4 \oplus C_{in} \oplus S_1。 \end{cases} \tag{7}$$

对于 4-2 压缩器有多种设计方法,但是必须满足基本公式(8)。

$$i_1 + i_2 + i_3 + i_4 + C_{in} = S_2 + 2(C_2 + C_{out})。 \tag{8}$$

3.1.2 传统的 Wallace 压缩结构

对比式(7)中 CSA 的逻辑表达式可以发现 CSA 就是进位保留的全加器,生成了两个伪和 C_1 和 S_1。而对比 4-2 压缩器的逻辑表达式可以发现其压缩器的 C_{out} 和 C_{in} 没有代数关系,不会造成级联,适合并行乘法器的使用。但是对于传统的 4-2 压缩器,缺点是关键路径即 C_2 的生成经过了 4 级异或门的传输延时。传统的 Wallace 压缩结构图如图 8 所示,两级压缩电路传输延时最长的路径是第一级 4-2 压

缩器的输入到第二级压缩器输出的 S_{21},最长门传输延时是 8 级异或门。

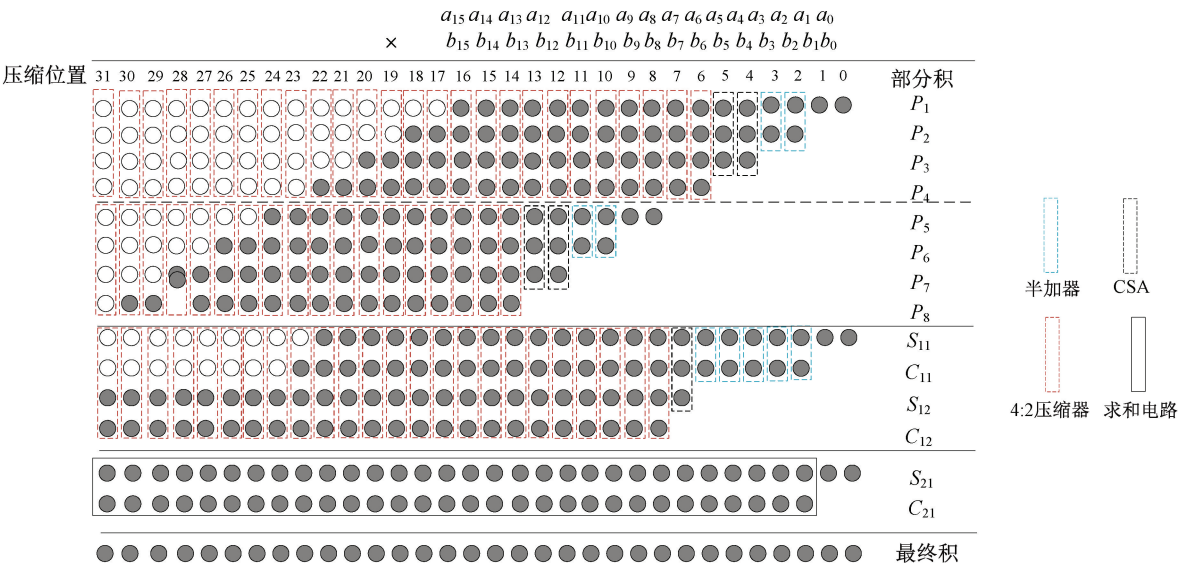

图 8　传统的 Wallace 压缩结构图

Figure 8　Traditional Wallace compressed structure diagram

在图 8 中,白色圆圈代表符号扩展位,$P_1 \sim P_8$ 为编解码电路生成的 8 个部分积,每个部分积的符号扩展位需要扩展到与最终积长度一致。首先 $P_1 \sim P_4$ 采用 4-2 压缩器、CSA 混合压缩生成了两个伪和 S_{11} 和 C_{11},同理 $P_4 \sim P_8$ 生成两个伪和 S_{12} 和 C_{12}。然后再将生成的 S_{11}、C_{11}、S_{12}、C_{12} 进行混合压缩生成第二级伪和 S_{21} 和 C_{21}。从图 8 中可以看出,传统 Wallace 压缩结构一共使用了 68 个 4-2 压缩器,造成 Wallace 压缩结构面积过大。

从压缩模块的面积和门传输延时两个角度来看,Wallace 压缩电路有很大的优化空间。

3.1.3　传统的求和电路

对于多位求和电路,常用结构和算法包括串行进位加法器(RCA)、跳跃进位加法器(CSPA)、选择进位加法器(SCA)、超前进位加法器(CLA)和并行前缀加法器(PPA)5 种。RCA 的速度最慢,面积最小;CSPA 速度相比于 RCA 有了很大提高,而且面积增加不大;SCA 和 CLA 的速度相差不大,都可以达到高速运算的要求;PPA 速度最快,相当于 SCA 的 1.5 倍,但电路结构复杂造成了其面积仍然比较大。

从式(8)中可以看出,4-2 压缩器的 C_{in} 跟 C_{out} 没有级联关系,并且图 8 中可以看出前 2 级的部分积压缩是高度并行的,所以关键路径的延时在于最后两个伪和数组成的求和电路。

传统求和电路使用 RCA 结构 30 位 S_{21} 和 C_{21} 进行求和,单个的 RCA 加法器结构 C_{in} 到 C_{out} 门传输延时是 2 级与(或)门,30 位 RCA 结构级联的门传输延时是 61 级。此加法器结构面积最小,但是门传输延时很大。文献[9]采用一级 CLA 结构,减小了求和电路传输延时,但是一级 32 位 CLA 造成了求和电路面积过大。

3.2　改进的 Wallace 压缩模块

3.2.1　改进的压缩器结构

本文改进的 CSA 结构如图 9(a)所示,改进的 4-2 压缩器设计如图 9(b)所示。在任何工艺中,与非门和或非门比与门和或门内部结构更简单,传输延时更小[10]。改进的 CSA 传输延时从 2 级与门变为 2 级与非门。改进 4-2 压缩器的最长门传输延时为 3 级异或门,比图 7(b)少了 1 级异或门,压缩器的门个数基本相同,同时满足 4-2 压缩器的基本公式。

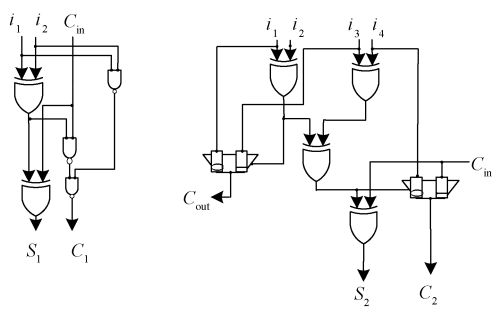

图 9　改进的 CSA 和 4-2 压缩器

Figure 9　Improved CSA and 4-2 compressor

3.2.2　符号位预处理

根据图 8 可以知,传统的 Wallace 压缩结构中每一行的部分积符号位都需要扩展到与乘积结果长度相同,符号扩展位较多,这需要用大量的 4-2 压缩器来对符号扩展位进行压缩,无疑增大了芯片面积。

针对符号扩展位过多的问题,由于 16 位乘法器生成的积位数最多是 32 位,对于符号位相加位数超过 32 位的数可以忽略不计,所以将符号扩展位进行计算得到规律表达如式(9)所示。将式(9)和式(10)中的 $\overline{S_1}S_1S_1$0101010101010 放入对应部分积的符号扩展位,对于 P_1 符号扩展位,根据式(10)可以把 $\overline{S_1}S_1S_1$ 放入 P_1 的 17~19 位。$S_1 \sim S_8$ 是 8 个部分积的符号位,通过对符号扩展位的预处理,将 72 个符号扩展位缩减到 10 位和 6 个 1。每个部分积符号位不再需要扩展到第 31 位,减小了 Wallace 压缩结构所使用的 4-2 压缩器数量。

$$\begin{aligned} Sign &= \sum_{n=1}^{8}\left(S_n \cdot \sum_{k=2n+15}^{31} 2^k\right) \\ &= \sum_{n=1}^{8}\left(1-\overline{S_n}\right)\cdot\left(2^{32}-2^{2n+15}\right) \\ &= 2^{32}\cdot\left(8-\sum_{n=1}^{8}\overline{S_n}\right)+\sum_{n=1}^{8}\left(\overline{S_n}2^{2n+15}\right)-\sum_{n=1}^{8}2^{2n+15} \\ &= \sum_{n=1}^{8}\left(\overline{S_n}2^{2n+15}\right)+(010101010101011)\cdot 2^{17}。 \end{aligned} \tag{9}$$

$$011+00\overline{S_1}=\overline{S_1}S_1S_1。 \tag{10}$$

3.2.3 改进的 Wallace 压缩结构

本文采用改进的 CSA 和 4-2 压缩器混合压缩,优化后的 Wallace 压缩结构如图 10 所示。此混合压缩结构一共使用了 11 个半加器、40 个 4-2 压缩器、11 个 RCA 和 9 个 CSA。另外采用新型 4-2 压缩器结构,使得两级压缩中的门传输延时由 8 级异或门缩短到 6 级异或门。

改进的混合压缩结构在第二级 4-2 压缩中 S_{12} 和 C_{12} 的[7:4]和[31:26]位提前使用 11 个 RCA 计算。优化后 C_{21} 的[31:26]和[8:0]位全部为 0,在 S_{21} 和 C_{21} 的[30:24]位可以使用半加器级联,31 位不必关心进位。通过第二级压缩高位的提前计算,使得伪和相加的两位求和电路相加位数从 32 位变为 16 位,显著提高了乘法电路的运算速度。

3.2.4 改进的求和电路

本设计将图 10 中的两位求和电路分为 4 组,采用两级 CLA 结构。组内采用 CLA 结构,组间采用 CLA 结构,其电路结构示意图如图 11 所示。

在图 11 中,$S_0 \sim S_{15}$ 为 16 位和,C_i 为低位进位,C_o 为两位求和电路的进位,$P_{G1} \sim P_{G4}$ 为组内的进位产生和进位传递信号产生模块;C_3、C_7、C_{11} 为利用组间进位模块产生的组内 CLA 进位输入端。改进的两级 CLA 求和电路只有 8 级门电路传输延时,相对于使用 RCA 结构的 33 级,缩短了 25 级。将改进的 16 位求和结构应用于 Wallace 压缩电路 S_{21} 和 C_{21} 的[25:10]位中,使得伪和相加电路门传输延时缩短为 16 级。改进的求和电路既缩短了使用 RCA 结构带来的过度延时,也避免了使用一级 CLA 结构带来的巨大面积开销。

4 仿真分析

本文设计采用硬件描述语言 Verilog HDL,使用 ModelSim 进行仿真,并在赛灵思 Xilinx 公司的 FPGA-xc3s500e 开发板上实现了 MCL 和 MSL 结构。通过写测试文件,遍历有符号 16 位数据范围(-37268~37267)。将测试文件数据 A 和 B 输入 VIVADO18.3 软件的 IP 核中,遍历所有乘法数据可能,对比两组乘法运算结果,如果一致则 correct 输

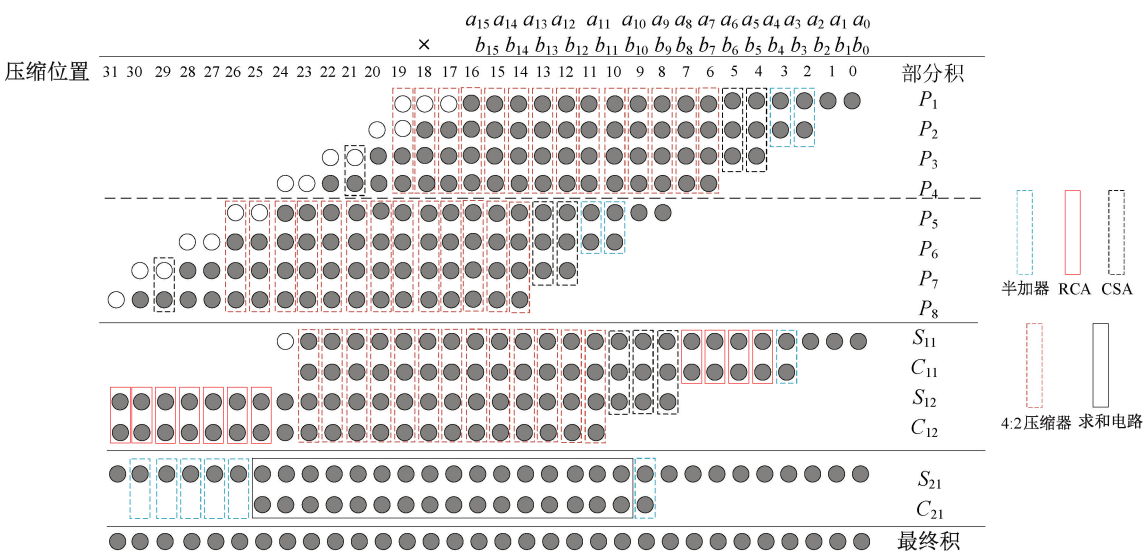

图 10 改进的 Wallace 压缩结构图

Figure 10 Improved Wallace compression structure diagram

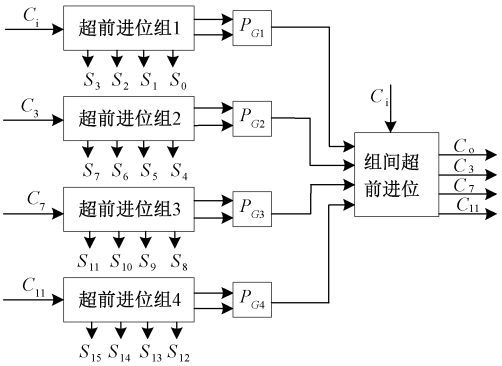

图 11　两级 CLA 结构示意图

Figure 11　Schematic diagram of two-level CLA structure

出为 1,否则为 0。初步验证两种乘法器结构功能的正确性后,在相同的综合策略下用 FPGA 板卡实现,将本文设计的 MSL 结构与文献[11]乘法器所使用的硬件资源、性能以及功耗进行比较,如表 4 所示。

表 4　仿真结果对比

Table 4　Comparison of simulation results

乘法器结构	LUT	FF	IO	电路延时/ns	功耗/mW
文献[11]	687	369	66	28.86	86.18
MSL	580	193	66	17.47	83.40

本文设计的 MSL 比文献[11]少了 107 个 LUT(查找表)和 176 个 FF(寄存器)。延时降低 39.5%,功耗减少 3.2%。因此初步判定,MSL 在面积和性能方面要优于现有的流水线乘法器。

5　结果分析

本文提出的 16 位高效乘法器在 VIVADO18.3 软件中验证功能正确性后,接着使用 TSMC180 nm 工艺标准阈值电压数字标准单元库,用 EDA Synopsys 工具 Design Compiler 对乘法器进行综合得到总电路的面积、电路延时和功耗延迟积 *PDP* 信息。新型乘法器与现有乘法器的对比如表 5 所示。

表 5　TSMC180 nm 综合结果对比

Table 5　Comparison of comprehensive results of TSMC180 nm

结构	面积/μm^2	电路延时/ns	功耗/mW	PDP/(ns·mW)
Synopsys	27 060	5.82	2.86	16.65
文献[4]	22 192	4.79	2.54	12.17
文献[12]	19 250	10.34	2.06	21.30
MCL	19 540	3.81	2.65	10.10
MSL	30 349	1.97	14.14	27.43

由表 5 对比可知,MCL 在面积,延时,功耗方面的数据表现均好于 Synopsys 中已经商用的乘法器 IP。MCL 与文献[4]相比面积减少了 12.0%,速度提高了 20.5%,功耗仅增加 4.3%,*PDP* 降低了 17.0%。MCL 与文献[12]相比速度提高 63.2%,PDP 降低了 52.6%,面积仅增加 1.5%。MCL 结构在面积、电路延时、功耗方面的表现比较均衡,适用于对乘法器单元的 3 个指标有同样高要求的 DSP 处理单元。MSL 采用增加流水线寄存器的方式,提高了乘法器的运行速度,所以性能最好,但同时带来的是面积和功耗的增加。MSL 结构在性能方面表现比较突出,适用于超高速微处理器的乘法计算单元中。

6　结论

本文主要目的是设计出一种高速并且面积小的,可以执行 16 位有符号数乘法的定点乘法器单元。新型乘法器对 Radix-4 编解码模块进行改进,改进后的编解码器电路并行度高、级数少。同时优化压缩模块的符号扩展位和压缩器单元,并提前计算部分数据位,降低了压缩模块电路的复杂度,减少了多位求和电路的位宽。综合结果表明,MCL 结构相比已有的乘法器面积减少了 12.0%,电路延时也降低了 20.5%,可以满足高性能定点 DSP 对乘法器在面积和速度上的要求。MSL 结构增加流水线寄存器来获得更快的运行速度,但代价是额外的面积和功耗。在未来工作中,可将标准单元库扩展,并使用扩展单元对关键路径进行优化,进一步提高乘法器的速度。

参考文献:

[1] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [EB/OL]. (2015-04-10) [2024-01-12]. http://arxiv.org/abs/1409.1556.

[2] VAMSI H S R, REDDY K S, BABU C, et al. Design of reversible logic based 32-bit MAC unit using Radix-16 Booth encoded Wallace tree multiplier[C]//2018 International Conference on Computer Communication and Informatics. Piscataway: IEEE, 2018: 1-6.

[3] RAMAKRISHNA A, BALAJI N, SRIHARI P. An efficient and enhanced memory based FFT processor using Radix 16 Booth with carry skip adder[C]//2016 International Conference on Signal Processing, Communication, Power and Embedded System. Piscataway: IEEE, 2016: 1608-1612.

[4] 姚若河, 徐新才. 基于冗余符号数的定点乘法器的设计[J]. 华南理工大学学报(自然科学版), 2014, 42(3): 27-34.
YAO R H, XU X C. Design of a fixed-point multiplier

based on redundant signed digit[J]. Journal of South China University of Technology (Natural Science Edition), 2014, 42(3): 27-34.

[5] SCHEUNEMANN J C, SIGALES M S, FONSECA M B, et al. Optimizing encoder and decoder blocks for a power-efficient Radix-4 modified Booth multiplier[C]//2021 34th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design. Piscataway: IEEE, 2021: 1-6.

[6] JAIN R, PAHWA K, PANDEY N. Booth-encoded Karatsuba: a novel hardware-efficient multiplier[J]. Advances in Electrical and Electronic Engineering, 2021, 19(3): 272-281.

[7] MONI D J, SOPHIA P E. Design of low power and high speed configurable booth multiplier[C]//2011 3rd International Conference on Electronics Computer Technology. Piscataway: IEEE, 2011: 338-342.

[8] 曾宪恺. 高性能并行乘法器半定制设计方法研究[D]. 杭州: 浙江大学, 2012.
ZENG X K. Research on semi-custom design method of high-performance parallel multiplier[D]. Hangzhou: Zhejiang University, 2012.

[9] PATIL P A, KULKARNI C. Multiply accumulate unit using Radix-4 Booth encoding[C]//2018 Second International Conference on Intelligent Computing and Control Systems. Piscataway: IEEE, 2018: 1076-1080.

[10] RAVULA M R, POTHARAJU A, VIDYADHAR R P. Designing carry look ahead adder to enrich performance using one bit hybrid full adder[C]//2022 International Conference on Electronics and Renewable Systems. Piscataway: IEEE, 2022: 86-89.

[11] RAM G C, SUBBARAO M V, VARMA D R, et al. Delay enhancement of Wallace tree multiplier with binary to excess-1 converter[C]//2023 5th International Conference on Smart Systems and Inventive Technology. Piscataway: IEEE, 2023: 113-117.

[12] RAJU A, SA S K. Design and performance analysis of multipliers using Kogge Stone adder[C]//2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology. Piscataway: IEEE, 2017: 94-99.

Design of Parallel Multiplier Based on Radix-4 Booth Coding

FAN Wenbing, ZHOU Jianzhang

(School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: Speed and area are two basic indexes to evaluate the performance of multiplier unit. Aiming at the problem of balancing layout area and transmission delay in current multiplier design, a new 16-bit signed fixed-point multiplier was designed by using Radix-4 Booth algorithm. In the process of partial product generation, firstly, the complementary code circuit for multiplier was improved, and then the modified Booth encoder and decoder with Radix-4 were optimized. The structure used less logic gate resources and was easy to parallelize the input bits. In the Wallace compression circuit, the symbol extension bits were preprocessed and a new compressor structure was designed to optimize the whole Wallace compression module. In the second stage of compression, the ripple carry adder structure was used to calculate the high bits in advance, which reduced the sum bits of multi-bit pseudo-sums. In the summation circuit, a two-stage carry look-ahead adder structure was used to shorten the transmission delay of the critical path and avoid increasing the chip area, thus improving the running speed of the multiplier. Compared with the existing multiplier structure, the new fixed-point multiplier reduced the area by 12.0% and the delay by 20.5%.

Keywords: Radix-4 Booth coding; area; transmission delay; encoder; decoder; Wallace compression