

全流水线结构双精度浮点乘加单元的设计

蔡 敏, 闵言灿

(华南理工大学 电信学院, 广东 广州 510641)

摘 要: 提出了一种支持非规格化数的全流水线结构双精度浮点乘加单元(Multiply-Add-Fused Unit, MAF, $A \times C + B$)。该乘加单元并行处理了主加法和舍入操作, 解决了进位保存形式的乘法结果带来的一位误差, 改进了规格化移位以便于流水线的划分。整个乘加单元划分为三级流水线, 在 $0.13\mu\text{m}$ CMOS 标准单元库中, 综合结果支持 333MHz 的时钟频率。

关键词: 乘加单元; 流水线; 非规格化数

中图分类号: TN4

文献标识码: A

文章编号: 1000-7180(2010)01-0053-04

Design of a Fully Pipelined Double-Precision Multiply-Add-Fused Unit

CAI Min¹, MIN Yan-can²

(School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China)

Abstract: This paper presents a fully pipelined double precision MAF(Multiply Add Fused unit, $A \times C + B$), which accepts denormalized numbers. The unit is based on combination of the rounding with final addition, moreover, a method to deal with the one bit error from the carry saved format of multiply result is proposed, finally, the rounding architecture is improved for getting better pipeline structure. The overall MAF has a latency of 3 cycles a throughput of 1 cycle, and a frequency of 333MHz in $0.13\mu\text{m}$ CMOS technology.

Key words: Multiply-Add; pipeline; denormalized numbers

1 引言

浮点运算单元(Floating Point Unit, FPU), 是 CPU 性能的关键单元, 在各种浮点运算中, 使用最为广泛的是浮点加(减)法和乘法运算(MAF)。MAF 的使用, 不仅通过硬件共享节省了芯片面积, 缩短了先乘后加操作的延时, 而且减少了一次舍入, 提高了结果精度。

2 乘加单元的整体结构

作为 FPU 中的一个关键单元, MAF 的算法已被深入地研究^[1-3], 但这些算法中却少有提及非规格化数的处理, 文中对文献[1-3]中的算法做了部分

改进, 设计出一个支持非规格化数的 MAF, 其结构如图 1 所示。

图 1 中虚线边框单元表示融入了非规格化数处理的结构, 由图 1 可见, 非规格化数的处理基本是融入到主数据通路中进行的, 这样可以在不增加关键路径的延时的前提下, 并行处理非规格化数。

3 乘加结构详细描述

按课题要求, 将 MAF 划分为三级流水线, 在图 1 中用点划线划分。下面分别介绍。

3.1 第一级流水线

本级流水线主要实现 A、C 尾数的乘法和 B 尾数的移位对齐操作。IEEE754 标准^[4]中双精度规格

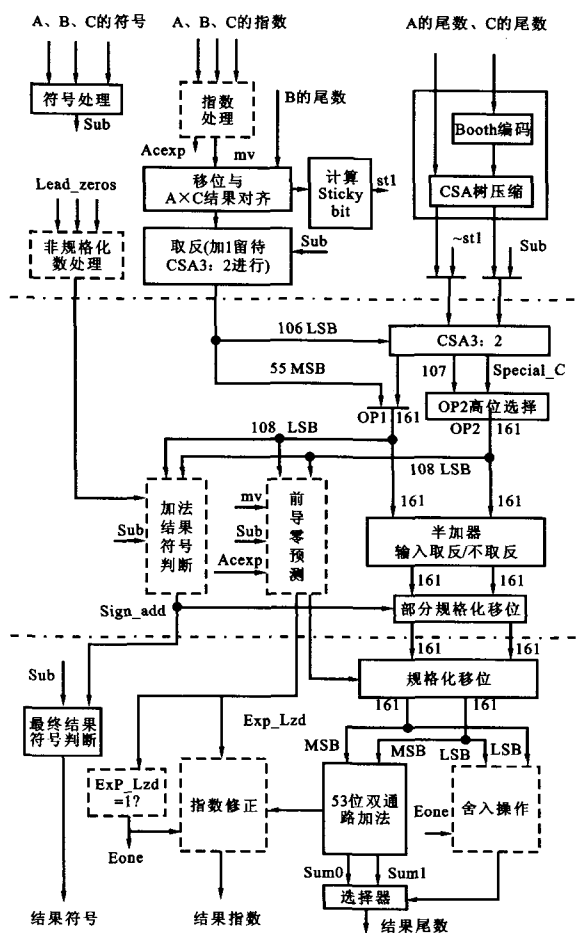


图1 乘加结构框图

化数和非规格化数分别表示为: $(-1)^{\text{sign}} * (1 + \text{mant}) * 2^{\text{exp} - 1023}$ 和 $(-1)^{\text{sign}} * (0 + \text{mant}) * 2^{-1022}$, 其中 sign 表示符号, mant 表示尾数, exp 表示指数. 对于非规格化数, 若修改其指数, 令其为 1, 这样非规格化数表示为 $(-1)^{\text{sign}} * (0 + \text{mant}) * 2^{1 - 1023}$, 于是非规格化数就能以规格化数的形式参与运算. 接下来, 并行进行符号指数处理、B 移位对齐、Booth 乘法和非规格化数预处理四个操作, 其中前三个操作和经典算法一致, 关于这里只介绍非规格化数的预处理操作:

对于非规格化数, 必须定位尾数第一个 1 出现的位置. 并且, 需要确定移位对齐后的 ACmant 和 Bmant 首个 1 的位置 (这里 ACmant 表示 $A * C$ 的尾数, Bmant 表示 B 的尾数). 令 LZA、LZB、LZC 分别表示 A、B、C 尾数中首个 1 前面 0 的个数, LZ_AC、LZ_B 分别表示移位对齐后 ACmant 与 Bmant 首个 1 前面 0 的个数, 令 mv 代表移位对齐时 Bmant 的移位个数. 这样 $\text{LZ_AC} = 55 + \text{LZA} + \text{LZC}(\text{LZ_AC}0)$ 或 $\text{LZ_AC} = 56 + \text{LZA} + \text{LZC}(\text{LZ_AC}1)$, $\text{LZ_B} =$

$\text{LZB} + \text{mv}$. 这两个数值将用于第二级流水线.

3.2 第二级流水线

该级流水线主要为将规格化移位移至主加法之前进行做必要的准备.

首先, 通过一个 107 位的 CSA3:2 (CarrySaved Adder), 将 A、C 尾数 Booth 算法得到的两进位保存结果和 B 尾数移位结果压缩为两操作数. 注意, 此时需要消除一个误差, 具体来说, 53 位的 Amant 和 Cmant 的乘积最多为 106 位, 而以进位保存形式存在的两个结果相加可能溢出. 为解决这一问题, 需分类讨论:

若 A、C 都是规格化数, 则 ACmant 在区间 $[1, 4)$ 内, ACmant 两进位保存结果形如 $1X. ? + 1X. ?$ 、 $11. ? + 01. ?$ 、 $01. ? + 11. ?$ 时, 则会溢出 (其他情形不会溢出, 如 $10. ? + 01. ?$ 时, 如果小数部分向高两位进位, 则会使最终结果小于 1); 若 A、C 中有一个为非规格化数, 则 ACmant 在区间 $[0, 2)$ 内, ACmant 两进位保存结果为 $1X. ? + XX. ?$ 或 $XX. ? + 1X. ?$ 时, 则会溢出, 其他情形均不会溢出, 这里 “X” 表示任意一位数、“?” 表示任意多位数. 用 Special_C 标志溢出情形, 并根据它计算主数据通道中的两操作数 (记为 OP1 和 OP2).

到此, 已获得两 161 位的 OP1、OP2, 直接将两操作数相加, 便可得精确结果. 显然, 161 位的加法代价太大, 为缩短主加法长度, 并行进行如下三个操作:

3.2.1 前导零预测

前导零预测使得规格化移位前置主加法之前成为可能. 该模块检测出 OP1 和 OP2 之和的前导零个数, 并计算出规格化左移位数 (记为 Numlsft) 以及规格化时指数的中间结果 (记为 Exp_Lzd).

前导零预测的方法有多种^[5-6]. 实际上, 这里只有 OP1 和 OP2 的低 108 位才需预测前导零, 因为在 B 移位对齐过程中, 若首位 1 的位置出现在前 53 位, 那么前导零个数直接由 B 移位结果得到. 另外, 考虑到非规格化数, 规格化移位时不能移掉所有的零. 详细描述如下 (参见表 1):

(1) $\text{mv} = 0$ 时, 不需要规格化左移, 中间指数即 B 的指数.

(2) $0 < \text{mv} < 54$. 此时, B 移位结果小数点比 ACmant 至少高出三位. 若 B 为非规格化数, 考虑到 $\text{Exp_Lzd} > 0$, 于是, 规格化左移位数等于 Bmant 右移位数, $\text{Exp_Lzd} = \text{Bexp} = 1$; 若 B 为规格化数, 且做等效减法时, 不会向 B 移位结果最高位借位,

$Numlsft = mv, Exp_Lzd = Bexp$;若 B 为规格化数,且做等效加法,此时有可能向 B 移位结果最高位进位,规格化左移时需保留一位,即 $Numlsft = mv - 1, Exp_Lzd = Bexp + 1$,其中 $Bexp$ 代表 B 的指数.

(3) $mv \geq 54$. 这里,需要判断 $OP1$ 和 $OP2$ 低 108 位相加结果的前导零个数(记为 LZ),这里参考 [5] 中提供的方法,先得到一个等效序列,然后解码获得 LZ .

$Bmant$ 在右移 54 位之后,能确保不向高 53 位进位,若不考虑结果为非规格化数的情形,则 $Numlsft = 53 + LZ$;考虑到中间指数 Exp_Lzd 至少为 1,即 $ACexp - LZ + 3 \geq 1$ (加 3 是因为 $ACmant$ 小数点前面有四位,而实际结果小数点之前只有一位).

于是, $LZ \leq ACexp + 2$ 时, $Numlsft = LZ + 53, Exp_Lzd = ACexp - LZ + 3$; $LZ > ACexp + 2$ 时,结果为非规格化数, $Numlsft = ACexp + 55, Exp_Lzd = 1$,其中 $ACexp$ 代表 $A * C$ 指数.

表 1 Numlsft 及 Exp_Lzd 的计算

情形	Numlsft	Exp_Lzd
$mv = 0$	0	$Bexp$
$0 < mv < 54$ 且 B 非规格化	mv	$Bexp$
$0 < mv < 54$ 且做等效减	mv	$Bexp$
$0 < mv < 54$ 且做等效加	$mv - 1$	$Bexp + 1$
$mv \geq 54$ 且 $LZ \leq ACexp + 2$	$53 + LZ$	$ACexp - LZ + 3$
$mv \geq 54$ 且 $LZ > ACexp + 2$	$55 + ACexp$	1

前导零预测可能产生一位的误差,文献[7]中提出了解决方法,但是处理方法相当复杂,在下一级流水线中巧妙处理了此问题.

3.2.2 加法结果符号判断

按照 IEEE754 标准,浮点数使用原码表示正数和负数,而在浮点运算的中间结果都是用补码表示,所以若想在 前导零预测时,并行处理取反加一的操作,就必须预先知道加法结果的正负.实际上,符号判断只有在等效减法操作中,且在移位后的 $Bmant$ 和 $ACmant$ 最高位相同时(即 $LZ_B = LZ_AC0$ 或 $LZ_B = LZ_AC1$ 时),才需要比较;而在等效加法时,结果符号肯定为零.具体判断如下:

在等效减法时,若 $LZ_B > LZ_AC1$,则加法结

果符号为 0;若 $LZ_B < LZ_AC0$,则加法结果符号为 1;其他情形,则通过比较 $\sim OP1$ 和 $OP2$ 的低位可得结果符号.

3.2.3 双通道半加及部分规格化移位

由于加法结果符号判断及前导零预测的延时较大,这样在 $CSA3:2$ 与规格化移位之间便有一个时间间隙.这里将规格化左移的前 53 位移位于此处进行,以填充剩余的时间空隙.最后根据加法结果符号对结果加以选择.

考虑到加法结果可能为负值,这里采用两个半加器分别处理输入取反和不取反的结果,这里需要处理两个加 1 操作,一个置于半加器进位字的最低位,另一个留待下一级流水线中处理.

3.3 第三级流水线

该级流水线主要完成规格化移位及主加法和舍入操作.

3.3.1 规格化移位

由于上级流水线已完成 53 位的规格化移位,这里只用处理 $Numlsft$ 可能大于 53 的情形,即表 1 的后两种情形.这时最大可能的移位数为 $161 - 53 = 108$.在移位过程中,若加法结果为负值,则需在移位的“和字”后面全补 1,这是为了在处理上一级流水线留下的加 1 操作时,进位可以一直传递到移位前的最低位.

3.3.2 加法和舍入

加法和舍入模块的基本算法思想:规格化移位后的高位送入加法器,低位送入舍入模块,加法和舍入是并行运行的,加法产生两个结果,一个不考虑低位进位的结果,一个考虑低位进位的结果,最后根据舍入模块加以选择.

加法和舍入模块与文献[2]、文献[3]、文献[8]中的对应模块类似,考虑到非规格化数的处理,这里做了局部修改,得到如下结构,参见图 2.

该结构将输入的 161 位操作数分为三部分处理:高 51 位,中间 4 位和低 106 位,其中高位和低位的处理方式基本与参考文献中的一致,这里只介绍与非规格化数处理以及舍入操作密切相关的中间 4 位加法.

中间 4 位加法中融入了舍入操作,考虑到低位可能的进位,四位加法需同时计算两个结果.在舍入过程中,还需处理左移和不左移两种情形.具体操作如图 3 所示.

是否需要左移由高 51 位加法不考虑低位进位与考虑低位进位的结果的最高位(分别记为 $sum0_$

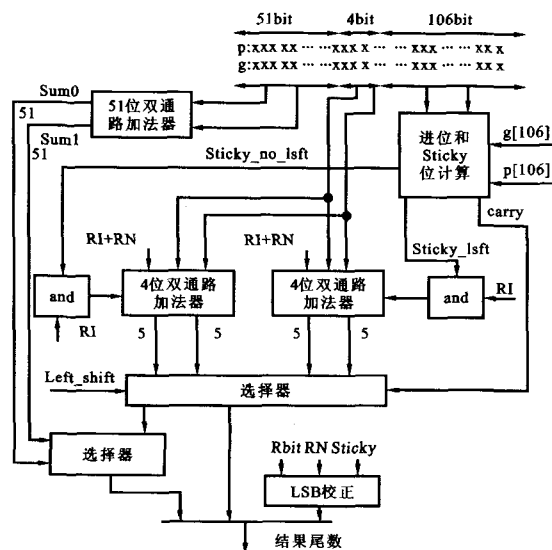


图2 加法和舍入

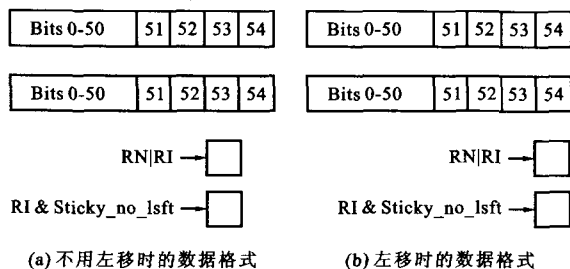


图3 四位加法数据格式

msb、sum1_msb)以及假设需要左移时中间四位向高51位的进位信号 c_lsft 确定;同时考虑到非规格数,当中间指数为1(标志位 $Eone=1$)时,根据前级流水线规定,结果为非规格化数,此时不能移位,这样最终的左移信号表达式为:

$Left_shift = \sim Eone \& \sim (sum0_msb \mid (sum1_msb \& c_lsft))$ 至于上一级流水线中前导零预测可能产生的一位误差,其处理方式和文献[3]中“加法和舍入模块”一致,这里不再详叙。

4 测试验证和设计结果

FPU的验证一直是设计的难点,1994年Intel公司就为奔腾处理器中浮点处理部分的瑕疵付出了沉重的代价.本设计的验证工作主要分两步进行。

我们首先使用了200万组随机测试向量对模块进行了测试,双精度MAF的输入可能有2192种,200万也只是冰山一角.测试向量由IEEE754标准工作组推荐的Softfloat软件生成,但由于该软件只支持单独的加减乘运算,我们对其进行了改进,使之支持乘加运算.乘加运算的规范结果,通过采集POWER PC系列芯片的乘加运算结果得到。

接下来,使用直接生成的测试向量进行测试,该测试向量由国内某研究所提供,专门用于测试浮点运算,其中包含了各种边角情况.用这组测试向量来测试MAF的针对性和对特殊情况的覆盖率,都会比随机测试要强一些。

最后使用Synopsys VCS Coverage Metrics进行覆盖率评估.由于MAF结构的功能固定,这里只需评价其代码覆盖率.测试结果显示,随机测试的覆盖率偏低,而对于直接生成的测试向量,由于考虑到了各种边角情形,覆盖率接近100%,测试结果符合项目要求.模块在测试验证后使用Synopsys DC进行综合.在 $0.13\mu m$ CMOS标准单元库中,在最坏工艺条件下,整个乘加单元的延时为8.5ns,划分为三级流水线后,支持333MHz的工作频率。

5 结束语

文中给出了一种基于全流水线结构的支持非规格化数的双精度浮点乘加单元结构:在数据通路中融入了非规格化数的并行处理;为减少关键路径延时,将规格化移位前置主加法之前进行,主加法由传统算法的161位缩减至53位;为填充第二级流水线的时间空隙,规格化移位分为两步进行,前53位的移位至于第二级流水中;解决了A、C尾数乘积进位保存形式结果带来了一位误差;使用较简便的方法实现了主加法和舍入操作的并行进行。

本设计在基本不增加关键路径延时的前提下,处理非规格化数,为非规格化浮点运算的设计提供了参考。

总的来说,FPU的前端设计空间并不大,文中给出的是一种较为成熟的算法结构,要想进一步提高性能,应该从后端入手,如采用全定制后端设计和使用最新的工艺等。

参考文献:

- [1] Jessani M N, Putrino M. Comparison of single and dual-pass multiply-add fused floating-point units[J]. IEEE Trans. Computers, 1998, 47(9):927-937.
- [2] Tomas L, Javier D B. Floating-point multiply-add fused with reduced latency[J]. IEEE Trans. Computers, 2004, 53(8):988-1003.
- [3] Gongqiong Li, Zhaolin Li. Design of a fully pipelined single precision multiply add fused unit[C]// IEEE 20th International Conference on VLSI Design. Beijing: Tsinghua University, 2007.

(下转第60页)

- [7] 李倩,王永县,朱友芹. 人工神经网络混合剪枝算法[J]. 清华大学学报, 2005, 45 (6): 831 - 834.
- [8] Islam M, Murase K. A new algorithm to design compact two-hidden-layer artificial neural networks[J]. Neural Networks, 2001(14): 1265 - 1278.
- [9] Huang G B, Saratchandran P, Sundararajan N. A generalized growing and pruning RBF (CGAP-RBF) neural network for function approximation[J]. IEEE Trans. on Neural Networks, 2005(16): 57 - 67.
- [10] Ahmed S, Abdullah - Al - Mamun K, Islam M. A novel algorithm for designing three layered artificial neural networks[J]. International Journal of Soft Computing, 2007, 2(3): 450 - 458.
- [11] Narasimhaa P L, Delashmitb W H, Manrya M T, et al. An integrated growing - pruning method for feedforward network training[J]. Neuro computing, 2008(71): 2831 - 2847.

作者简介:

肖秀春 男, (1976 -), 博士研究生, 讲师. 研究方向为神经网络、图像处理、信息安全等.

姜孝华 男, (1963 -), 博士, 教授, 博士生导师. 研究方向为神经网络、信号检测与处理技术.

张雨浓 男, (1973 -), 博士, 教授, 博士生导师. 研究方向为神经网络、机器人、高斯过程.

(上接第 56 页)

- [4] IEEE Standard. Binary floating - point arithmetic[S]. ANSI/IEEE Standard 754, 1985.
- [5] Schmookler M S, Nowka K J. Leading zero anticipation and detection - a comparison of methods[C]// Proc. IEEE 15th symp. USA: Austin Computer Arithmetic, 2001: 7 - 12.
- [6] 梅小露. 浮点乘加部件中三操作数前导 1 预测算法的设计[J]. 微电子学与计算机, 2005, 22(12): 16 - 20.
- [7] Javier B, Tomas L. Leading-one prediction with concurrent position correction[J]. IEEE Trans. Computers, 1999, 48(10): 1083 - 1097.
- [8] Yu R K, Zyner G B. 167 MHz radix 4 floating point multiplier[C]// Proc. IEEE 12th Symp. Computer Arithmetic. UK, Bath, 1995: 149 - 154.

作者简介:

蔡 敏 男, (1955 -), 博士, 教授, 博士生导师. 研究方向为集成电路设计.

闵言灿 男, 硕士. 研究方向为集成电路设计.