

存算一体芯片中的符号位加法器电路设计

艾力

(杭州智芯科微电子科技有限公司 310000)

摘要：本文基于 TSMC22N 工艺设计了一种符号位加法器电路，应用于存算一体芯片中。该加法器电路为 4bit 加法器树，其输入为多个 4bit 二进制数，输出为所有输入进行二进制相加的结果。本设计将 4bit 加法器树拆分成一个单 bit 和一个 3bit 加法器树，利用单 bit 树可进行符号位处理，利用 3bit 树进行无符号位的运算，并将这两个树的结果送到累加器进行移位累加和符号位处理，输出最终的乘加运算结果。通过时分复用，该加法器电路实现了二进制有符号位的 8bit 乘加运算，并节省了芯片面积，仿真结果也验证了功能正确。

关键词：符号位处理；加法器树；存算一体；累加器；

前言

传统的“冯·诺依曼”架构存在“存储墙”和“功耗墙”^[1]，限制了算力的发展，而存内计算^[2]直接将计算在内存中实现，打破了“存储墙”和“功耗墙”的限制，可提高算力和能效比。

存内计算用到的基本电路单元为存储器，乘法器，加法器，累加器。其中加法器和累加器是计算的关键电路，占了绝大部分功耗。为了节省面积，我们采用了时分复用^[3]的方法，将 8bit 的 weight 数据分成高 4bit 和低 4bit，8bit 的输入数据 XIN 按位依次与 weight 的高 4bit 和低 4bit 进行相乘，得到结果输送到 4bit 的加法器树，需要 16 个周期才能完成。如果进行 8bit 有符号位的二进制乘加运算^[4]，则还需要进行符号位的处理，低 4bit 是无符号位的，高 4bit 是有符号位的，因此加法器树需要兼容有符号位和无符号位的运算。本文的符号位加法器电路就是基于以上应用而提出的。

1. 二进制乘加运算原理分析

两个 8bit 有符号位二进制数的乘加运算公式如下：

$$y = \sum_{i=0}^{255} x_i < 7:0 > * w_i < 7:0 > \quad (1)$$

$x_i < 7 >$ 和 $w_i < 7 >$ 为符号位， i 为个数，上式中有 256 个 x_i 和 w_i 相乘以后再相加，其最终的结果 y 为一个 24bit 的二进制数。将 w_i 分成高 4bit 和低 4bit，并展开①式，得：

$$y = \sum_{i=0}^{255} (2^0 x_i < 0 > * w_i < 3:0 > + 2^4 x_i < 0 > * w_i < 7:4 > + 2^1 x_i < 1 > * w_i < 3:0 > + 2^5 x_i < 1 > * w_i < 7:4 > + 2^2 x_i < 2 > * w_i < 3:0 > + 2^6 x_i < 2 > * w_i < 7:4 > + 2^3 x_i < 3 > * w_i < 3:0 > + 2^7 x_i < 3 > * w_i < 7:4 > + 2^4 x_i < 4 > * w_i < 3:0 > + 2^8 x_i < 4 > * w_i < 7:4 >$$

$$+ 2^5 x_i < 5 > * w_i < 3:0 > + 2^9 x_i < 5 > * w_i < 7:4 > + 2^6 x_i < 6 > * w_i < 3:0 > + 2^{10} x_i < 6 > * w_i < 7:4 > + (-1)^{x_i < 7 >} 2^7 x_i < 7 > * w_i < 3:0 > + (-1)^{x_i < 7 >} 2^{11} x_i < 7 > * w_i < 7:4 > \quad (2)$$

从上式可以看出，是先进行乘法运算以后再求和。由于这种算法的实现比较复杂，非常耗资源和面积，对上式进行变形得公式如下：

$$y = 2^0 \sum_{i=0}^{255} (x_i < 0 > * w_i < 3:0 >) + 2^4 \sum_{i=0}^{255} (x_i < 0 > * w_i < 7:4 >) + 2^1 \sum_{i=0}^{255} (x_i < 1 > * w_i < 3:0 >) + 2^5 \sum_{i=0}^{255} (x_i < 1 > * w_i < 7:4 >) + 2^2 \sum_{i=0}^{255} (x_i < 2 > * w_i < 3:0 >) + 2^6 \sum_{i=0}^{255} (x_i < 2 > * w_i < 7:4 >) + 2^4 \sum_{i=0}^{255} (x_i < 4 > * w_i < 3:0 >) + 2^8 \sum_{i=0}^{255} (x_i < 4 > * w_i < 7:4 >) + 2^5 \sum_{i=0}^{255} (x_i < 5 > * w_i < 3:0 >) + 2^9 \sum_{i=0}^{255} (x_i < 5 > * w_i < 7:4 >) + 2^6 \sum_{i=0}^{255} (x_i < 6 > * w_i < 3:0 >) + 2^{10} \sum_{i=0}^{255} (x_i < 6 > * w_i < 7:4 >) + 2^7 \sum_{i=0}^{255} ((-1)^{x_i < 7 >} x_i < 7 > * w_i < 3:0 >) + 2^{11} \sum_{i=0}^{255} ((-1)^{x_i < 7 >} x_i < 7 > * w_i < 7:4 >) \quad (3)$$

从上式可以看出共有 16 项 4bit 的加法，无符号位和有符号位的加法运算各有 7 项，还有两项存在负数运算，需要对其相加的结果取补码。

2. 加法器电路设计

传统的 4bit 加法器树^[5]不能实现无符号位和有符号位的兼容运算。因此，本文提出将 4bit 的加法器树拆分成单 bit 和 3bit 两个加法器树，单 bit 树进行 4bit 里面最高位的运算，3bit 树进行剩下的低 3bit 的运算，两种树同时运行，实现 4bit 加法器树的功能。

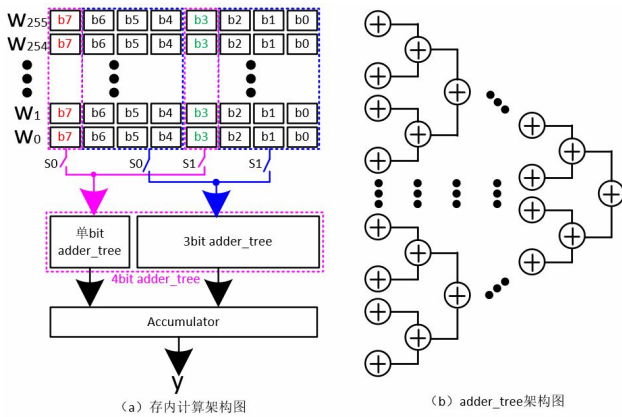


图1 存内计算 4bit adder_tree 架构示意图

如上图 1 (a) 所示, 共有 256 行 8bit weight 数据进行乘加运算, b7 为符号位, b0~b6 为数值位, S0 和 S1 为 MUX 开关控制信号。将 weight 数据分为高 4bit 和低 4bit, 每次运算只用其中的一种。4bit adder_tree 由无符号位的单 bit 和 3bit adder_tree 组成。b7 和 b3 与输入 XIN 数据相乘的结果通过 MUX 开关输入到单 bit adder_tree, b4~b6 和 b2~b0 与输入 XIN 数据相乘的结果通过 MUX 开关输入到 3bit adder_tree。单 bit adder_tree 将 256 输入相加以后输出 1 个 9bit 的二进制结果到累加器。同理, 3bit adder_tree 输出 1 个 12bit 的二进制结果到累加器。

如果单 bit 或者 3bit adder_tree 进行的是符号位的运算, 即计算公式③中含有 $(-1)^{x_i < 7} x_i < 7$ 这一项, 则其输出结果在 Accumulator 中需要取补码。如果单 bit 或者 3bit adder_tree 进行的是非符号位的运算, 则其输出结果不需要取补码。

图 1 提出的 adder_tree 需要 16 个周期才能完成一次运算。Adder_tree 16 次的输出结果需要在累加器进行移位以后再累加才能输出最终的结果。

如图 1 (b) 所示, adder_tree 的架构为两两相加后送入下一级, 直到最后一级两两相加才结束。单 bit 和 3bit 的加法器树架构是一样的, 只不过他们的输入数据一个是单 bit, 另一个是 3bit, 输出结果一个是 9bit, 另一个是 12bit。

3 仿真结果与分析

输入 XIN 全部为“0101 0101”, 即十进制 85, weight 数据共 256 个, 其中 128 个为“0101 0101”, 即十进制 85, 另外 128 个为“1010 1010”, 即十进制-86。其仿真波形如图 2 所示。

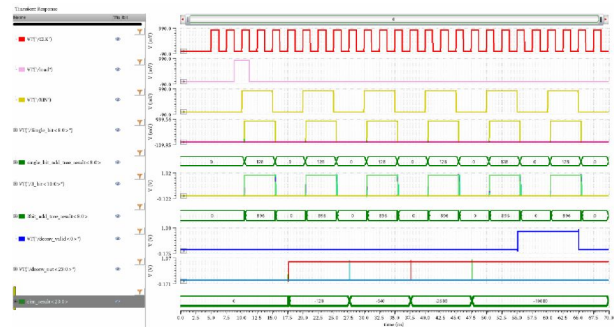


图 2 存算一体电路仿真波形

根据输入的 XIN 和 weight 数据, 可以理论推算出来运算结果: $128 * 85 * 85 + 128 * 85 * (-86) = -10880$ 。上图 2 的仿真结果与理论推导一致。单 bit adder_tree 和 3bit adder_tree 的输出结果也符合预期。

4 结论

本文基于存算一体芯片的应用, 设计了一种加法器树电路, 实现了 8bit 二进制有符号位的乘加运算。Spectre 仿真结果表明, 单 bit 和 3bit adder_tree 都可以进行分时复用, 大大节省了芯片面积, 符号位的处理放在累加器中处理, 减少加法器树的复杂度, 且仿真结果表明 CIM 功能正确。

参考文献:

- [1]周立昕. 基于 SRAM 结构的高效能存算一体技术研究 [D]. 桂林电子科技大学, 2022.
- [2]陆春帆, 刘爽, 周洲. 一种新型存算一体电路设计与研究 [J]. 科学技术创新, 2022, 36:91-94.
- [3]张钟宣, 张海清, 艾力, 徐康健. 向量乘加器的 SRAM 存内计算装置和电子设备. CN115658011B [P]. 2023.
- [4]王兴华, 王天, 王乾, 李潇然. 基于带符号位的浮点运算的多位宽 3D RRAM 设计 [J]. 北京理工大学学报, 2022, 12(42):1299-1304.
- [5]Bonan Yan. A 1.041-Mb/mm² 27.38-TOPS/W Signed-INT8 Dynamic-Logic-Based ADC-less SRAM Compute-In-Memory Macro in 28nm with Reconfigurable Bitwise Operation for AI and Embedded Applications [C]//COMPUTE-IN-MEMORY AND SRAM, ISSCC, 2022.

作者简介: 艾力, 男 1989 04, 湖北咸宁, 汉, 本科, 中级工程师, 集成电路设计。