

PAM 代码简述

现在要实现的是一个近似乘法器，实现方式如下：

对于指数和符号部分，不做修改使用传统方法进行计算，即：

1. 对于符号部分，使用异或算法
2. 指数部分直接相加
3. 优化的是 mantissa 部分的计算

假设两个 mantissa 输入是 x 和 y（包含整数位的 1）

M 指的是 mantissa result

对于 Level $n = 0$: $\text{delta } M = (0.5 + 1/2^{(2n+2)}) + 1.5(x-1) + 1.5(y-1)$

对于 Level $n(n > 0)$: $\text{delta } M = P_{n,x} + P_{n,y}$

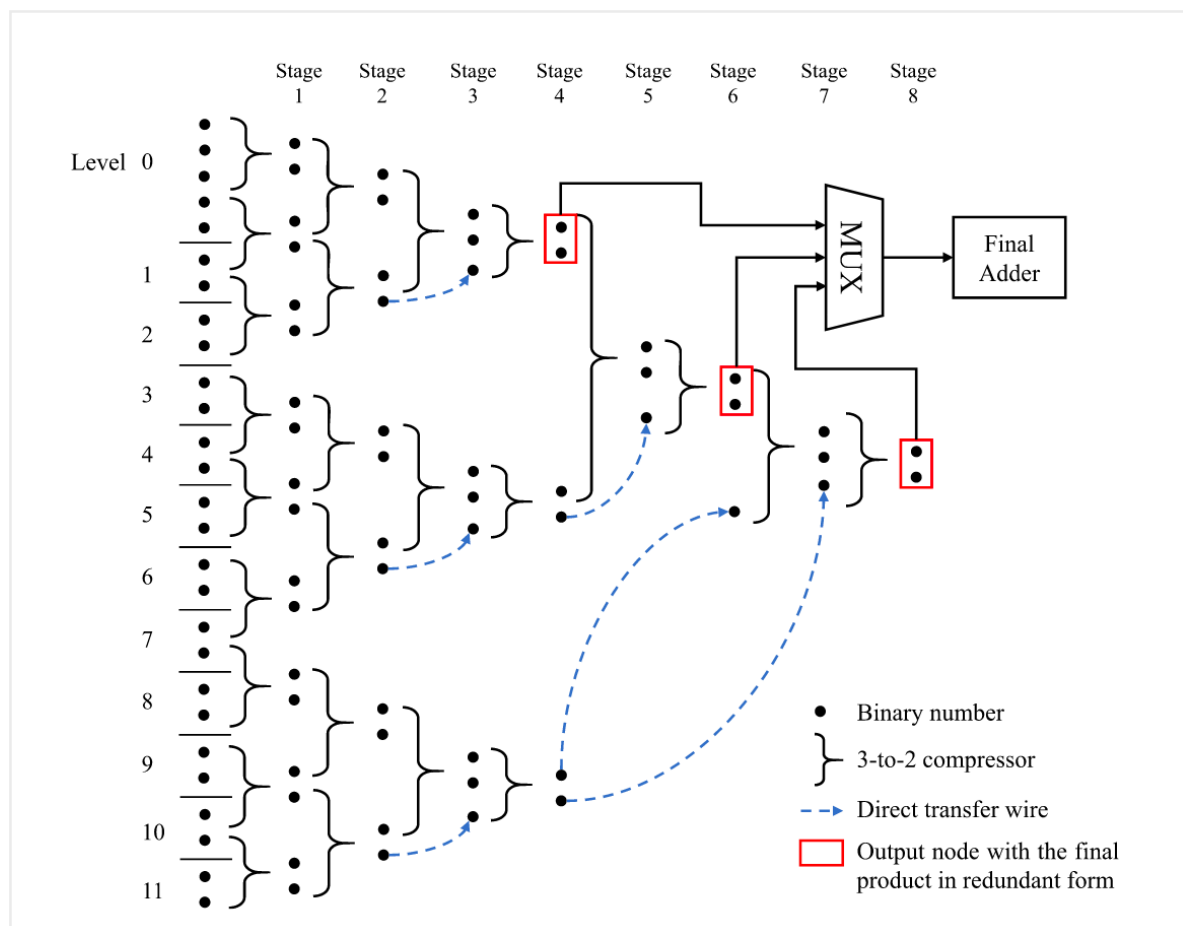
对于 $P_{n,x}$ $P_{n,y}$ 的解释：n 表示 Level - 近似级别

P 的公式：

$$p_{x,n} = \{x[n] \odot y[n : M]\}_{2n+1} + \{\overline{x[n]}\}_{M+n+1}$$
$$p_{y,n} = \{y[n] \odot x[n + 1 : M]\}_{2n+2} + \{\overline{y[n]}\}_{M+n+1}$$

根据所设定的近似级别 n 计算出 0 - n 的 delta M,

这篇文章提出了一种降低能耗计算最终和的办法，并不是单纯的累加，如图所示



这个方法利用了一个树状结构来对计算进行压缩，图中每一个小点代表加法中的一项，

其中level 0中的5项是 $x + y + 0.5x + 0.5y + (0.5 + 1/2^{(2n+2)})$ 【 $(0.5 + 1/2^{(2n+2)})$ 这个部分就可以转化为一个十分位是1，在 $2n+2$ 位上是1，其余位数都是0的一个小数】

对于其余的level，两个小点就是 $p_{n,x}$ 和 $p_{n,y}$

有了这样一个树状结构之后，近似乘法器使用 3-2 compressor 对这些项进行压缩【按照图中的形式】，经历多个 stage 的 compress 以后利用一个多路选择器可以实现 3 个主 level 的计算 Level 2，level6，Level 11

而在 0-2，2-6，6-11 之间的 level，则使用 and 门进行控制，将特定的 P 设为 0，以达到对应的近似级别

最后，将 M 和之前得到的指数部分进行综合，如果 F 大于 2，需要进位，指数部分加一，M 部分需要进行右移