# 8\*8 Wallace树乘法器

Wallace树乘法器的运算原理：从数据最密集的地方开始，不断的反复使用全加器、半加器来覆盖“树”。通过全加器将树的深度不断缩减，最终缩减为一个深度为2的树。最后一级则采用简单的2输入加法器组成。（用四级全加器、半加器来将wallace树压缩，最后进行相加。）

全加器是一个3输入2输出的器件，因此全加器又称作3-2压缩器。

## 部分积

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  | x7 | x6 | x5 | x4 | x3 | x2 | x1 | x0 |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
| a |  |  |  |  |  |  |  | x7y0  a[28] | x6y0  a[21] | x5y0  a[15] | x4y0  a[10] | x3y0  a[6] | x2y0  a[3] | x1y0  a[1] | x0y0  a[0] |
| a |  |  |  |  |  |  | x7y1  a[36] | x6y1  a[29] | x5y1  a[22] | x4y1  a[16] | x3y1  a[11] | x2y1  a[7] | x1y1  a[4] | x0y1  a[2] |  |
| a |  |  |  |  |  | x7y2  a[43] | x6y2  a[37] | x5y2  a[30] | x4y2  a[23] | x3y2  a[17] | x2y2  a[12] | x1y2  a[8] | x0y2  a[5] |  |  |
| b |  |  |  |  | x7y3  a[49] | x6y3  a[44] | x5y3  a[38] | x4y3  a[31] | x3y3  a[24] | x2y3  a[18] | x1y3  a[13] | x0y3  a[9] |  |  |  |
| b |  |  |  | x7y4  a[54] | x6y4  a[50] | x5y4  a[45] | x4y4  a[39] | x3y4  a[32] | x2y4  a[25] | x1y4  a[19] | x0y4  a[14] |  |  |  |  |
| b |  |  | x7y5  a[58] | x6y5  a[55] | x5y5  a[51] | x4y5  a[46] | x3y5  a[40] | x2y5  a[33] | x1y5  a[26] | x0y5  a[20] |  |  |  |  |  |
| c |  | x7y6  a[61] | x6y6  a[59] | x5y6  a[56] | x4y6  a[52] | x3y6  a[47] | x2y6  a[41] | x1y6  a[34] | x0y6  a[27] |  |  |  |  |  |  |
| c | x7y7  a[63] | x6y7  a[62] | x5y7  a[60] | x4y7  a[57] | x3y7  a[53] | x2y7  a[48] | x1y7  a[42] | x0y7  a[35] |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

当部分积生成后，必须将它们相加以获得最终的结果。

而乘法器中压缩部分积的结构就被称为拓扑结构。有规整拓扑结构，也有不规整拓扑结构。常用的简单阵列、双阵列、二进制树等结构就属于规整的拓扑结构，而我们要使用的Wallace树就属于非规整的拓扑结构，可以有效减少延时。

在整个化简的过程中，我们仅采用3位全加器和2位半加器进行“圈划”。为了保证准确性，我们采用了相对保守的圈划方式，后续对成本优化的时候会再度分析。

如图所示，半加器用一个包含2个位的红圈表示，全加器用一个包含3个位的蓝圈表示。生成的结果位和进位用两位存储，例如b0[1:0]，其中b0[0]表示结果位，b0[1]表示进位。

## 第一级(使用16个加法器：12个全加器+4个半加器)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
| a |  |  |  |  |  |  |  | a[28] | a[21] | a[15] | a[10] | a[6] | a[3] | a[1] | a[0] |
| a |  |  |  |  |  |  | a[36] | a[29] | a[22] | a[16] | a[11] | a[7] | a[4] | a[2] |  |
| a |  |  |  |  |  | a[43] | a[37] | a[30] | a[23] | a[17] | a[12] | a[8] | a[5] |  |  |
| b |  |  |  |  | a[49] | a[44] | a[38] | a[31] | a[24] | a[18] | a[13] | a[9] |  |  |  |
| b |  |  |  | a[54] | a[50] | a[45] | a[39] | a[32] | a[25] | a[19] | a[14] |  |  |  |  |
| b |  |  | a[58] | a[55] | a[51] | a[46] | a[40] | a[33] | a[26] | a[20] |  |  |  |  |  |
| c |  | a[61] | a[59] | a[56] | a[52] | a[47] | a[41] | a[34] | a[27] |  |  |  |  |  |  |
| c | a[63] | a[62] | a[60] | a[57] | a[53] | a[48] | a[42] | a[35] |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 第二级(使用16个加法器：13个全加器+3个半加器)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
| a |  |  |  |  |  | a[43] | b11[0] | b9[0] | b7[0] | b5[0] | b3[0] | b2[0] | b1[0] | b0[0] | a[0] |
| a |  |  |  |  |  | b13[0] | b12[0] | b10[0] | b8[0] | b6[0] | b4[0] | b1[1] | b0[1] |  |  |
| a |  |  | b15[1] | b15[0] | b14[0] | b11[1] | b9[1] | b7[1] | b5[1] | b3[1] | b2[1] | a[9] |  |  |  |
| b |  |  | a[58] | b14[1] | b13[1] | b12[1] | b10[1] | b8[1] | b6[1] | b4[1] |  |  |  |  |  |
| b |  | a[61] | a[59] | a[56] | a[52] | a[47] | a[41] | a[34] | a[27] |  |  |  |  |  |  |
| b | a[63] | a[62] | a[60] | a[57] | a[53] | a[48] | a[42] | a[35] |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 第三级(使用10个加法器：6个全加器+4个半加器)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
| a |  |  |  |  | b14[0] | c10[0] | c8[0] | c6[0] | c4[0] | c3[0] | c2[0] | c1[0] | c0[0] | b0[0] | a[0] |
| a |  |  | b15[1] | b15[0] | c12[0] | c11[0] | c9[0] | c7[0] | c5[0] | c2[1] | c1[1] | c0[1] |  |  |  |
| a | a[63] | c15[0] | c14[0] | c13[0] | c10[1] | c8[1] | c6[1] | c4[1] | c3[1] | b4[1] |  |  |  |  |  |
| b | c15[1] | c14[1] | c13[1] | c12[1] | c11[1] | c9[1] | c7[1] | c5[1] |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 第四级(使用11个加法器：7个全加器+4个半加器)

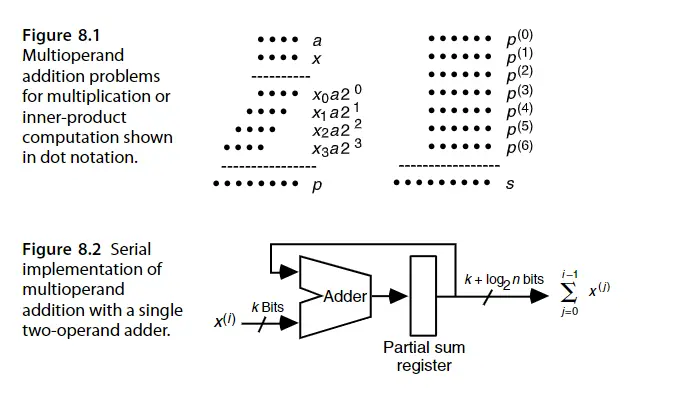
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
|  |  | d9[1] | d9[0] | d8[0] | d7[0] | d6[0] | d5[0] | d4[0] | d3[0] | d2[0] | d1[0] | d0[0] | c0[0] | b0[0] | a[0] |
|  | a[63] | c15[0] | d8[1] | d7[1] | d6[1] | d5[1] | d4[1] | d3[1] | d2[1] | d1[1] | d0[1] |  |  |  |  |
|  | c15[1] | c14[1] | c13[1] | c12[1] | c11[1] | c9[1] | c7[1] | c5[1] |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

第四次压缩后深度为2，直接相加即可得到最终的结果。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
| e10[1] | e10[0] | e9[0] | e8[0] | e7[0] | e6[0] | e5[0] | e4[0] | e3[0] | e2[0] | e1[0] | e0[0] | d0[0] | c0[0] | b0[0] | a[0] |
|  | e9[1] | e8[1] | e7[1] | e6[1] | e5[1] | e4[1] | e3[1] | e2[1] | e1[1] | e0[1] |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

共计使用了53个加法器，包括38个全加器，15个半加器。

# 加速比的理论证明



在计算n个k bits数之和的加法器，一个n个k bits输入的wallace树将它的输入压缩到2个k+log2 n-1 bits的输出。由于树中每层CSA（进位保存加法器）都将操作数数量减少1.5倍，n输入的wallace树的最小树高度函数h（n）可以用下递归式描述：

IMG_256

基于上式，我们可以精准地计算出n操作数所对应的wallace树的精确高度。如果我们忽略式中的向上取整符⌈x ⌉，将其写为h（n）=1+h(2n/3)，就能得到它的下界：

h(n)≥1+log1.5(2n/3)

当且仅当n=2,3时，上式等号成理。

另一种研究树高的思路是：n高wallace树至多能将n(h)个输入数压缩到2个输出。这个n(h)满足下递归式：

IMG_257

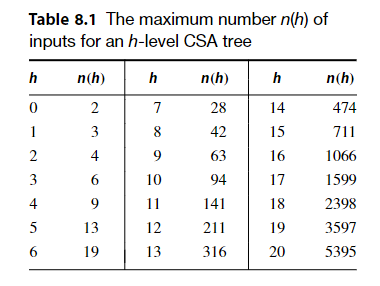
忽略其中的向下取整符⌊x ⌋，有上界：

n(h)≤2(3/2)h

我们也可以获得它的下界：

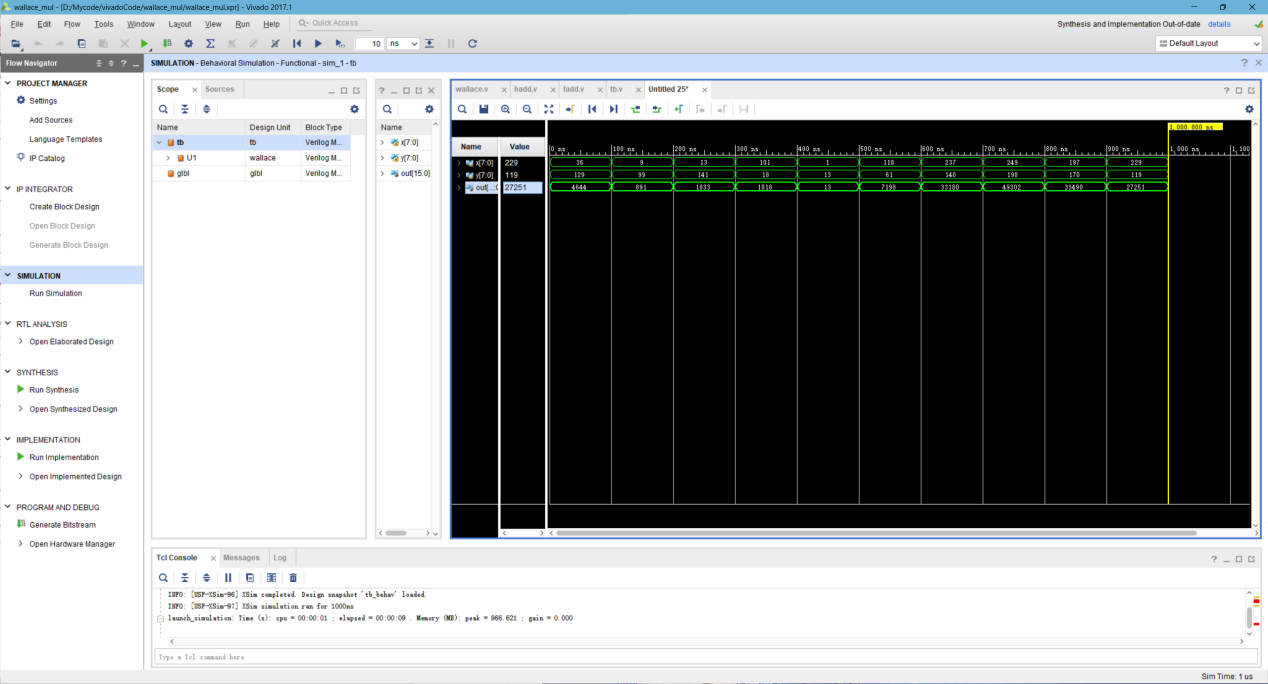
n(h)>2(3/2)h-1

下表给出了0≤h≤20时的n(h)：



wallace树的思路是一有机会就减少操作数的数量。就是说，walalce树在每层都尽可能地利用所有可缩减操作数的击会。

# 仿真验证



# 成本优化

## 第一级(使用19个加法器：15个全加器+4个半加器)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
|  | a[63] | a[61] | a[58] | a[54] | a[49] | a[43] | a[36] | a[28] | a[21] | a[15] | a[10] | a[6] | a[3] | a[1] | a[0] |
|  |  | a[62] | a[59] | a[55] | a[50] | a[44] | a[37] | a[29] | a[22] | a[16] | a[11] | a[7] | a[4] | a[2] |  |
|  |  |  | a[60] | a[56] | a[51] | a[45] | a[38] | a[30] | a[23] | a[17] | a[12] | a[8] | a[5] |  |  |
|  |  |  |  | a[57] | a[52] | a[46] | a[39] | a[31] | a[24] | a[18] | a[13] | a[9] |  |  |  |
|  |  |  |  |  | a[53] | a[47] | a[40] | a[32] | a[25] | a[19] | a[14] |  |  |  |  |
|  |  |  |  |  |  | a[48] | a[41] | a[33] | a[26] | a[20] |  |  |  |  |  |
|  |  |  |  |  |  |  | a[42] | a[34] | a[27] |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | a[35] |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 第二级(使用14个加法器：10个全加器+4个半加器)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  | x7 | x6 | x5 | x4 | x3 | x2 | x1 | x0 |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
|  | a[63] | a[61] | b18[0] | a[54] | b15[0] | b13[0] | a[36] | b8[0] | a[21] | b4[0] | b2[0] | a[6] | a[3] | a[1] | a[0] |
|  |  | a[62] | b17[1] | b17[0] | b16[0] | b14[0] | b11[0] | b9[0] | b6[0] | b5[0] | b3[0] | b1[0] | b0[0] | a[2] |  |
|  |  | b18[1] |  | b15[1] | b13[1] | b11[1] | b12[0] | b10[0] | b7[0] | b2[1] | b1[1] | b0[1] |  |  |  |
|  |  |  |  | b16[1] | b14[1] | b12[1] | b8[1] | b6[1] | b4[1] | b3[1] |  |  |  |  |  |
|  |  |  |  |  |  |  | b9[1] | b7[1] | b5[1] |  |  |  |  |  |  |
|  |  |  |  |  |  |  | b10[1] |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 第三级(7个加法器：6个全加器+1个半加器)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  | x7 | x6 | x5 | x4 | x3 | x2 | x1 | x0 |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
|  | a[63] | c13[0] | c12[0] | a[54] | b15[0] | b13[0] | c7[0] | c5[0] | c3[0] | b4[0] | c1[0] | a[6] | a[3] | a[1] | a[0] |
|  | c13[1] | c12[1] | c11[1] | c11[0] | c10[0] | c9[0] | c8[0] | c6[0] | c4[0] | c2[0] | c0[1] | c0[0] | b0[0] | a[2] |  |
|  |  |  |  | c10[1] | c9[1] | c7[1] | c5[1] | c3[1] | c2[1] | c1[1] |  |  |  |  |  |
|  |  |  |  |  |  | c8[1] | c6[1] | c4[1] |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 第四级(8个加法器：3个全加器+5个半加器)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  | x7 | x6 | x5 | x4 | x3 | x2 | x1 | x0 |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
|  | a[63] | c13[0] | c12[0] | d6[0] | d5[0] | b13[0] | c7[0] | c5[0] | d1[0] | b4[0] | c1[0] | a[6] | a[3] | a[1] | a[0] |
|  | c13[1] | c12[1] | c11[1] | d5[1] | d4[1] | d4[0] | d3[0] | d2[0] | d0[1] | d0[0] | c0[1] | c0[0] | b0[0] | a[2] |  |
|  |  |  | d6[1] |  |  | d3[1] | d2[1] | d1[1] |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

第四次压缩后深度为2，直接相加即可得到最终的结果

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 被乘数 |  |  |  |  |  |  |  | x7 | x6 | x5 | x4 | x3 | x2 | x1 | x0 |
| 乘数 |  |  |  |  |  |  |  | y7 | y6 | y5 | y4 | y3 | y2 | y1 | y0 |
| e7[1] | e7[0] | e6[0] | e5[0] | e4[0] | e3[0] | e2[0] | e1[0] | c5[0] | d1[0] | b4[0] | c1[0] | x3y0  a[6] | x2y0  a[3] | x1y0  a[1] | x0y0  a[0] |
|  | e6[1] | e5[1] | e4[1] | e3[1] | e2[1] | e1[1] | e0[1] | e0[0] | d0[1] | d0[0] | c0[1] | c0[0] | b0[0] | x0y1  a[2] |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

优化后共计使用了48个加法器，包括34个全加器，14个半加器，较大程度上进行了优化。