

	31
PP1	1
PP2	1
PP3	1
PP4	1
PP5	1
PP6	1
PP7	1
PP8	1

	31
PP1	这一部分的4: 2压缩
PP2	
PP3	
PP4	

	31
PPC1_1	
PPC1_2	

	31
PP5	
PP6	
PP7	
PP8	

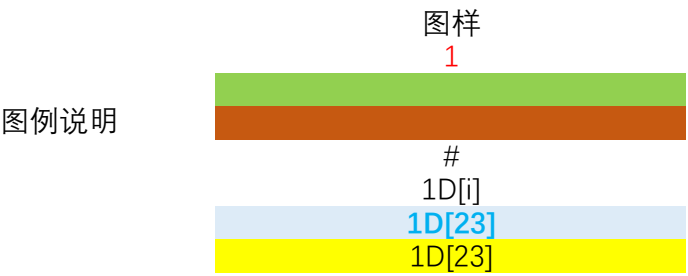
PPC1_3	
PPC1_4	

	31
PPC1_1	
PPC1_2	
PPC1_3	
PPC1_4	

	31
PPC2_1	
PPC2_2	

结果

符号位由乘数和被乘数异或得到



最高位符号位由辵

30	29	28	27	26
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	2
1	1	2	3	4

30	29	28	27	26
完全可以使使用23位的一个4：2压缩替代,因为其输入输出关系是一样的				

30	29	28	27	26
1D[23]	1D[23]	1D[23]	1D[23]	1D[23]
1C[23]	1C[23]	1C[23]	1C[23]	1C[23]

30	29	28	27	26
1	1	1	1	1
1	1	1	1	1
1	1	1	2	3
1	2	3	4	5

1D[30]	1D[29]	1D[28]	1D[27]	1D[26]
1C[29]	1C[28]	1C[27]	1C[26]	1C[25]
注意这里的1C[i]或者1D[i]与上面几行里的数值不相同，这里只是表达位置关系，二者				

30	29	28	27	26
1D[23]	1D[23]	1D[23]	1D[23]	1D[23]
1C[23]	1C[23]	1C[23]	1C[23]	1C[23]
1D[30]	1D[29]	1D[28]	1D[27]	1D[26]
1C[29]	1C[28]	1C[27]	1C[26]	1C[25]

---

30	29	28	27	26
2D[30]	2D[29]	2D[28]	2D[27]	2D[26]
2C[29]	2C[28]	2C[27]	2C[26]	2C[25]

说明

符号扩展

使用4-2压缩

使用3-2压缩

补零

第一级压缩器的在第*i*位权重位置的和输出

使用精简版4：2(只输出本位权重，实际上是异或)

由4：2压缩后的数据扩展得到

前入的两个数据的高位异或得到

25	24	23	22	21	20	19	18	17
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	3
1	1	1	1	1	2	3	4	5
1	2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12	13

第一级处理得到PPC1\_1 PPC2\_1

25	24	23	22	21	20	19	18	17
		1	1	1	1	1	1	1
		1	1	1	1	1	1	2
		1	1	1	1	2	3	4
		1	1	2	3	4	5	6

第一级处理得到PPC1\_1 PPC2\_2

25	24	23	22	21	20	19	18	17
1D[23]	1D[23]	1D[23]	1D[22]	1D[21]	1D[20]	1D[19]	1D[18]	1D[17]
1C[23]	1C[23]	1C[22]	1C[21]	1C[20]	1C[19]	1C[18]	1C[17]	1C[16]

第一级处理PPC1\_3 PPC2\_3

25	24	23	22	21	20	19	18	17
1	1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13	14

第一次处理PPC1\_3 PPC2\_4

25	24	23	22	21	20	19	18	17
1D[25]	1D[24]	1D[23]	1D[22]	1D[21]	1D[20]	1D[19]	1D[18]	1D[17]
1C[24]	1C[23]	1C[22]	1C[21]	1C[20]	1C[19]	1C[18]	1C[17]	1C[16]
数据不同								

第二级处理得到PPC2\_1 PPC2\_2

25	24	23	22	21	20	19	18	17
1D[23]	1D[23]	1D[23]	1D[22]	1D[21]	1D[20]	1D[19]	1D[18]	1D[17]
1C[23]	1C[23]	1C[22]	1C[21]	1C[20]	1C[19]	1C[18]	1C[17]	1C[16]
1D[25]	1D[24]	1D[23]	1D[22]	1D[21]	1D[20]	1D[19]	1D[18]	1D[17]
1C[24]	1C[23]	1C[22]	1C[21]	1C[20]	1C[19]	1C[18]	1C[17]	1C[16]

25	24	23	22	21	20	19	18	17
2D[25]	2D[24]	2D[23]	2D[22]	2D[21]	2D[20]	2D[19]	2D[18]	2D[17]
2C[24]	2C[23]	2C[22]	2C[21]	2C[20]	2C[19]	2C[18]	2C[17]	2C[16]

16	15	14	13	12	11
1	1	2	3	4	5
2	3	4	5	6	7
4	5	6	7	8	9
6	7	8	9	10	11
8	9	10	11	12	13
10	11	12	13	14	15
12	13	14	15	16	
14	15	16			

2\_1

16	15	14	13	12	11
1	2	3	4	5	6
3	4	5	6	7	8
5	6	7	8	9	10
7	8	9	10	11	12

1过程

16	15	14	13	12	11
1D[16]	1D[15]	1D[14]	1D[13]	1D[12]	1D[11]
1C[15]	1C[14]	1C[13]	1C[12]	1C[11]	1C[10]

4

16	15	14	13	12	11
9	10	11	12	13	14
11	12	13	14	15	16
13	14	15	16	17	
15	16	17			

过程

		1Cin[i] = 1Co[i-1]			
1D[16]	1D[15]	1D[14]	1D[13]	1D[12]	14
1C[15]	1C[14]	1C[13]	1C[12]	#	16

2\_2

16	15	14	13	12	11
1D[16]	1D[15]	1D[14]	1D[13]	1D[12]	1D[11]
1C[15]	1C[14]	1C[13]	1C[12]	1C[11]	1C[10]
1D[16]	1D[15]	1D[14]	1D[13]	1D[12]	14
1C[15]	1C[14]	1C[13]	1C[12]	#	16

2过程

					2Cin[i] = 2Co[i-1]
16	15	14	13	12	11
2D[16]	2D[15]	2D[14]	2D[13]	2D[12]	2D[11]
2C[15]	2C[14]	2C[13]	2C[12]	2C[11]	2C[10]



10	9	8	7	6	5	4
6	7	8	9	10	11	12
8	9	10	11	12	13	14
10	11	12	13	14	15	16
12	13	14	15	16		
14	15	16				
16						

---

10	9	8	7	6	5	4
7	8	9	10	11	12	13
9	10	11	12	13	14	15
11	12	13	14	15	16	17
13	14	15	16	17		

---

1Cin[i] = 1Co[i-1] .Cin[9] = 1Co[8].Cin[8] = 1Co[7].Cin[7] = 1Co[6] 1Cin[6] = 0						
10	9	8	7	6	5	4
1D[10]	1D[9]	1D[8]	1D[7]	1D[6]	1D[5]	1D[4]
1C[9]	1C[8]	1C[7]	1C[6]	1C[5]	1C[4]	#

---

10	9	8	7	6	5	4
15	16	17				
17						

---

15	16	17				
17						

---

10	9	8	7	6	5	4
1D[10]	1D[9]	1D[8]	1D[7]	1D[6]	1D[5]	1D[4]
1C[9]	1C[8]	1C[7]	1C[6]	1C[5]	1C[4]	#
15	16	17				
17						

---

2cin[10] = 0						
10	9	8	7	6	5	4
2D[10]	2D[9]	2D[8]	1D[7]	1D[6]	1D[5]	1D[4]
2C[9]	2C[8]	#	1C[6]	1C[5]	1C[4]	#

3	2	1	0
13	14	15	16
15	16		

3	2	1	0
14	15	16	17
16	17		

3	2	1	0
14	15	16	17
16	17		

3	2	1	0


3	2	1	0
14	15	16	17
16	17		

3	2	1	0
14	15	16	17
16	17	#	#