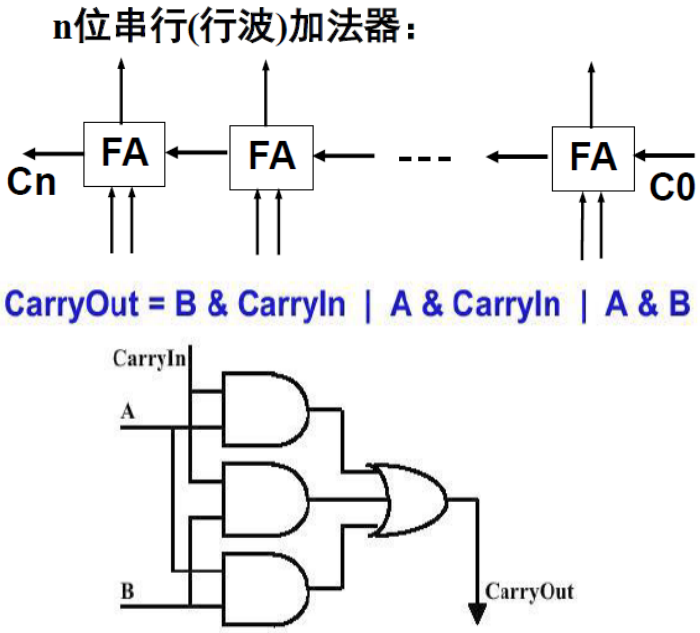
算数逻辑部件（ALU）杂谈

ALU是计算机重要的部件，而ALU的实现是很复杂的一件事。
关于ALU的构造，“加法器”是ALU的重要部分，不仅加减法可以通过补码直接用加法器实现，乘法和除法也只需要ALU有左/右移功能即可直接用算法实现。
因此，本文着重于讲述加法器的原理，希望读者可以在看完本文后可以对ALU的来历更加清晰。

Part0:行波加法器

这一部分其实不想多讲，看行波加法器的原理图，我们可以就可以知道行波加法器的计算过程和我们平时的竖式加法是一样的。



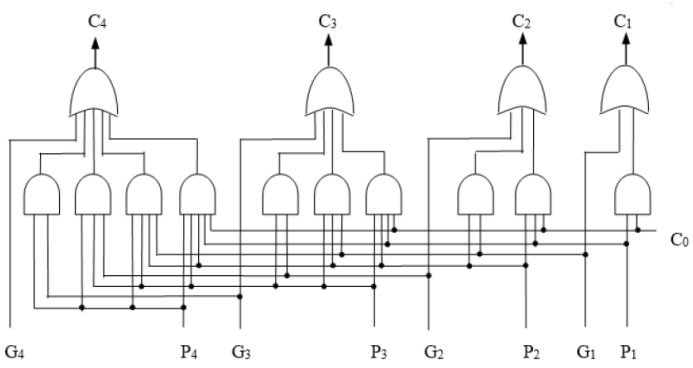
Part1:CLA（并行进位加法器）的构造

先说CLA中的FA比起全加器的改进。本来的一位全加器输入A、B、Cin，输出S、Cout，而这里的FA则输出F、P、G。那F、P、G分别是什么意思呢?
F是这一个FA的结果，一位全加器的输出是三个值的异或（这一点不难理解）；G是生成进位函数，即如果A和B均为1，则进位一定产生（和已经为2）；P是进位传递函数，即如果A和B中至少一个为1，则再接收到Cin后，进位一定会传递（和已经为2）。

考虑完改进FA的构造后，再考虑CLU的原理。虽然CLU的电路图很复杂，但总结起来就一个递推式：

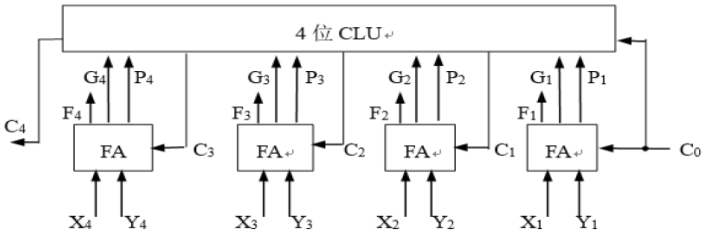
$$C_i = G_{i-1} + P_{i-1} * C_{i-1}。$$

即进位传递器传递进位（要么G已经确保有进位，要么P和C同时存在则也传递进位）。随后就是通过递推式一步步将C_i展开，让四个C可以同时计算出结果，而不是一个一个递推出来。



4位CLU部件

最后考虑CLA的构造。CLA的构造和串行加法器的原理是一样的，都是接受输入$A1-A4$、$B1-B4$和$Cin0$，输出结果$S1-S4$和高位传递进位$Cout$。只是Cin是通过CLU立刻算出来并迅速传递给高位的（G、P的结果均和Cin无关）。因此，各级进位的产生是同时的，回输入至FA中也可以同时输出$S1-S4$。



Part2:高位CLA的构造

首先是组件进位的函数逻辑单元，这个器件是用来判断是否有向上的进位的。因此Gg是确定“一定有进位”，而Pg则是确定能否将最低位的进位Cin向上传递为4位上的进位$Cout$。
那么Gg和Pg的式子就很好懂了。
当$G4, G3*P4, G2*P3*P4, G1*P2*P3*P4$任何一个为真的时候，已经保证有进位，分别来自$G4/G3/G2/G1$并且可以一路传递至最高位。故：

$$Gg = G4 + G3*P4 + G2*P3*P4 + G1*P2*P3*P4。$$

再说Pg。如果最低位的进位可以传递至最高位，那么可以传递的条件是$P1/P2/P3/P4$全部为真，只有这样进位才可以一路传递到最上方，那么非常自然有：

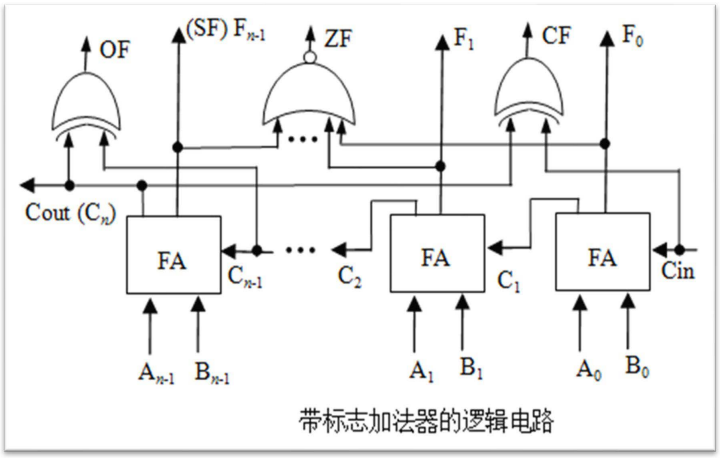
$$Pg = P1 * P2 * P3 * P4。$$

随后我们介绍可级联的4位CLA。这只是在原本的基础上加上了Pg和Gg，即这个CLA是可以“向上传递”进位的，类似于四位的FA。因此只需要在原本的CLA设计上，将四个FA的G、P拿出来接至组件进位的函数逻辑即可成为了可级联的CLA。

最后我们探讨16位CLA。4位的可级联CLA至于16位的CLA，类似于FA至于4位CLA。因此，两者的架构是一样的，而且进位的考虑只需要同样用4位CLU，打个比方的话，这一次是类似于“16进制加法”的FA，所以和之前的逻辑是一样的。

Part3:带符号位加法器中符号位的含义及其由来

符号位的意义不难理解，但符号位为什么是这样的式子呢？本段将详细剖析。



带标志加法器的逻辑电路

首先是SF(Sign Flag)。
SF是带符号整数加减运算结果的符号位，因此直接取结果的最高位作为SF。

然后是零标志ZF(Zero Flag)。
是根据每一位加法器的计算结果相或得到的，所以只有在结果的每一位都为0的时候所有位相或才能得到0的输出。

然后是溢出标志OF(Overflow Flag)。
OF是判断带符号数是否溢出的位。若是两个正数相加越界，则本来其符号位均是0，非符号的最高位一定会溢出，但符号位均是0，即使最高位有进位，符号位也不会有进位。反之，如果是两个负整数相加越界为正，因为符号位均为1且和的符号位为0，则最高位不可能有进位给符号位，而且符号位之和一定有进位。
因此采取：

$$OF = Cn/Cout \oplus Cn - 1。$$

最后是进/借位标志CF(Carry Flag)。
进借位标志其实是无符号数是否上溢（进位）或者下溢（借位）。这里就必须提到说为什么一定要有一个Cin了。因为ALU中的加法器是一个整体，所以Cin并不是输入的进位，毕竟这个加法器不是级联的，最低位是不可能接受下面的进位的。

因此Cin代表加还是减。由于补码是取反加一，因此可以说有Cin的加法器是可以用补码实现减法的。当加法时，$Cin = 0$，因为加法不需要加一；而$Cin = 1$时，则其中一方的输入取反后设$Cin = 1$，这样计算就更为方便。因此也很好理解CF的来历。

$Cin = 0$时是加法，因此有最高位上溢就代表进位；$Cin = 1$时是减法，最高位没有上溢（注意减数一定最高位是1，因为是补码）则代表最高位只可能是0和1，而且没有进位代表结果最高位是1，故结果为补码负数，即减法越界，有借位。故：

$$CF = Cout \oplus Cin$$

最后在以减法举例，说明为什么是如此。
假设是四位的加法，$(A - B) = (A_{补} + (-B)_{补}) = A + (1111 - B + 1) = 10000 + (A - B)$，而10000这个值是进位的最小值。
因此，当$A - B < 0$时答案小于10000，是没有进位的，$A >= B$时有进位)

希望本文有助于读者理解加法器。