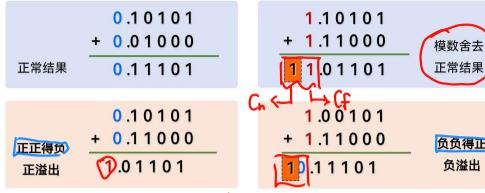


从补码计算合理性证明见 PPT
计算机识别运算结果是否溢出方法



$$\begin{aligned} X+Y \text{ 补} &= [X] \text{ 补} + [Y] \text{ 补} \\ [X-Y] \text{ 补} &= [X] \text{ 补} - [Y] \text{ 补} = [X] \text{ 补} + [-Y] \text{ 补} \\ [-Y] \text{ 补} &= ! [Y] \text{ 补} + 1 \end{aligned}$$

溢出逻辑：正正得负，负负得正

① 单符号位运算溢出：设两数符号位 f_0, f_1 和数据符号位 f_s
溢出检测 OVERFLOW = $f_0 f_1 f_s + \bar{f}_0 \bar{f}_1 f_s$

② 单符号位运算溢出：2. 符号位进位 CF，最高位进位 C_n

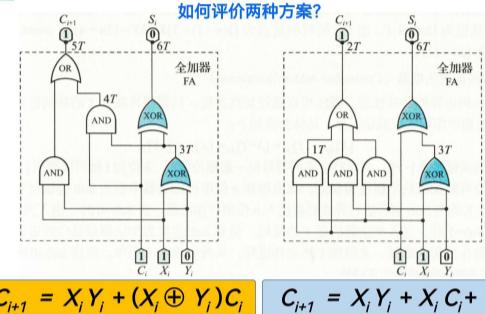
溢出检测 OVERFLOW = $C_f \oplus C_n$

③ 双符号位运算溢出：符号位最高位 f_1 ，次高位 f_2
溢出检测 OVERFLOW = $f_1 \oplus f_2$

最短位，短位是正确操作

④ 无符号数溢出检测：产生进位/借位信号即溢出 $VOF \leftarrow Sub \oplus Cout$

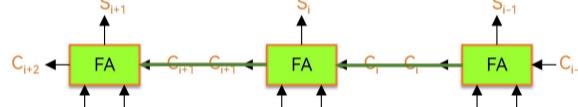
电路实现：相同权重进位相加，进位从低向高传递；首先考虑进位链



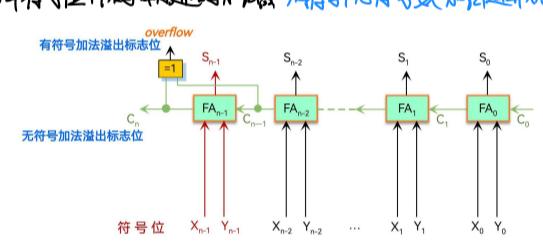
n位加法器

串行进位加法器电路

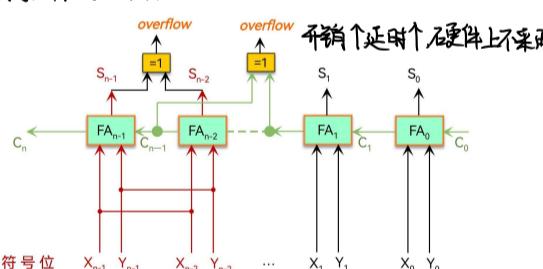
① 包含n个全加器 ② n个后全加器串联 ③ 低进位输出连接到高进位输入



(1) 单符号位补码串行进位加法器 有符号/无符号数加法运算规则相同，判断溢出方式不同



(2) 双符号位补码加法器电路

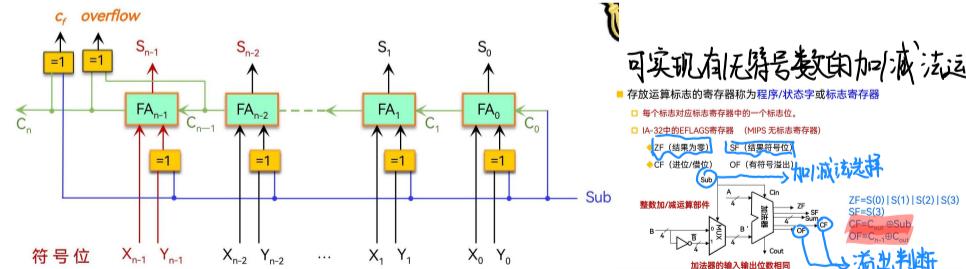


加法器功能完善 \Rightarrow 对 Y 输入进行选择 (引入 sub 作运算控制位)

※ 减法器电路实现 $[X] \text{ 补} - [Y] \text{ 补} = [X] \text{ 补} + [-Y] \text{ 补}$ 关键，方法：Y补同符号位逐位取反未位加一 $[-Y] \text{ 补} = [-[Y] \text{ 补}] \text{ 补}$

$sub = \begin{cases} 0 & \text{0 时作加法, } [Y] \text{ 补送入加法器} \\ 1 & \text{1 时作减法, } [-Y] \text{ 补送入加法器 (对 Y 逐位取反后未位加一)} \end{cases}$

※ 减法器实现 $[X] \text{ 补} - [Y] \text{ 补} = [X] \text{ 补} + [-Y] \text{ 补}$ 方案：Y补同符号位逐位取反未位加一



快速加法器 \Rightarrow 打破进位依赖 (进位穿路/进位选择/超前(先行)进位)

$$S_i = X_i \oplus Y_i \oplus C_i$$

$$C_{i+1} = X_i Y_i + (X_i \oplus Y_i) C_i \Rightarrow C_{i+1} = G_i + P_i C_i$$

G_i 进位生成 P_i 进位传递 \Rightarrow 传递到和生成部分

* 高位进位信息产生

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 C_0 G_0 + P_0 C_0 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 C_1 G_1 + P_1 G_0 + P_1 P_0 C_0 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

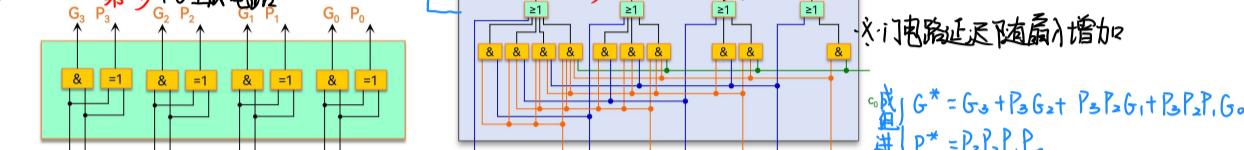
.....

$$C_n = G_{n-1} + P_{n-1} G_{n-2} + P_{n-1} P_{n-2} G_{n-3} + \dots + P_{n-1} P_{n-2} \dots P_0 C_0$$

进位输出仅与 P, G 与最低级进位输出 C_0 有关，经过一级与逻辑和一级或逻辑

位数长，进位链电路复杂度越高；常四组一组分组运算

成组进位生成类似

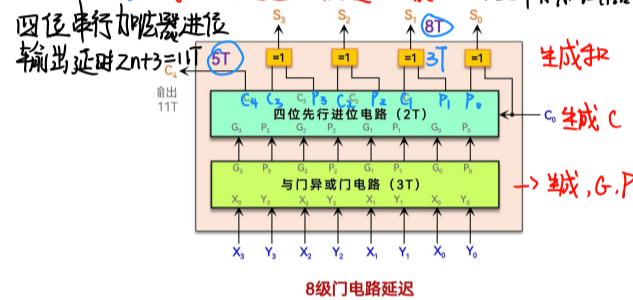


* 门电路延迟随扇入增加

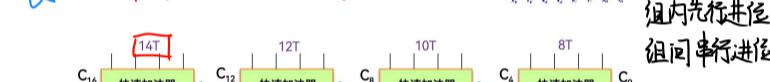
$$G^* = G_0 + P_0 G_1 + P_1 P_0 G_0 + P_0 P_1 G_0$$

$$P^* = P_0 P_1 P_0$$

四位先行进位快速加法器 (四位串行加法器和输出延时 $2n+4 = 12T$)



十六位加法器 \Rightarrow 四个四位先行进位加法器



16位串行加法器 $(2n+4)T = 16T$ 上半快加器节约 22T

成组进位电路

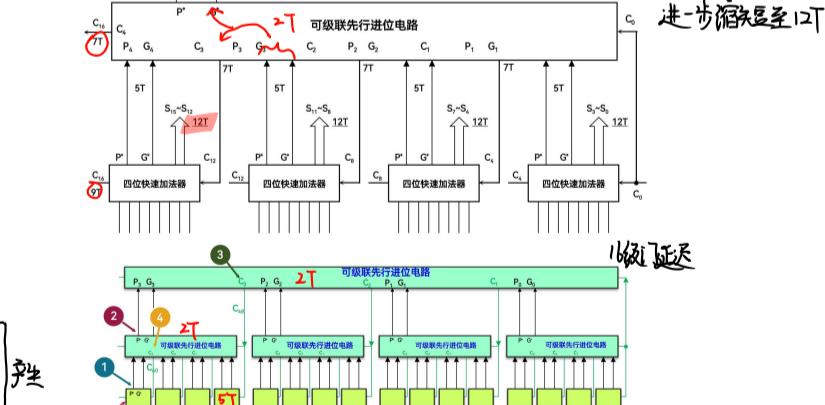
$$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

$$G^* = G_0 + P_0 G_1 + P_1 P_0 G_0 + P_0 P_1 G_0$$

$$P^* = P_0 P_1 P_0$$

$$C_4 = G^* + P^* C_0 \rightarrow 将每一组进位输出再按先行进位公式展开$$

组与组之间传递



① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T \rightarrow ④ 生成 C_{60} 需 2T \rightarrow ⑤ 求得进位需 2T \rightarrow ⑥ 求得 3T

进一步缩短至 12T

16位延时

① 生成 P, G, 第一级 P*, G* 需 5T \rightarrow ② 生成第二级 P*, G* 需 2T

\rightarrow ③ 生成 C_{3/48} 需 2T $\rightarrow</$