中山大學 软件工程学院

SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

计算机组成原理实验

授课老师：吴炜滨

大纲

- 快速加法器设计实验1
  - 8位可控加减法器
  - 4位先行进位电路

实验报告

- 本次实验报告提交时间
  - 周二教学班：11月8号凌晨0点前（第11周周二）
  - 周四教学班：11月10号凌晨0点前（第11周周四）

实验目的

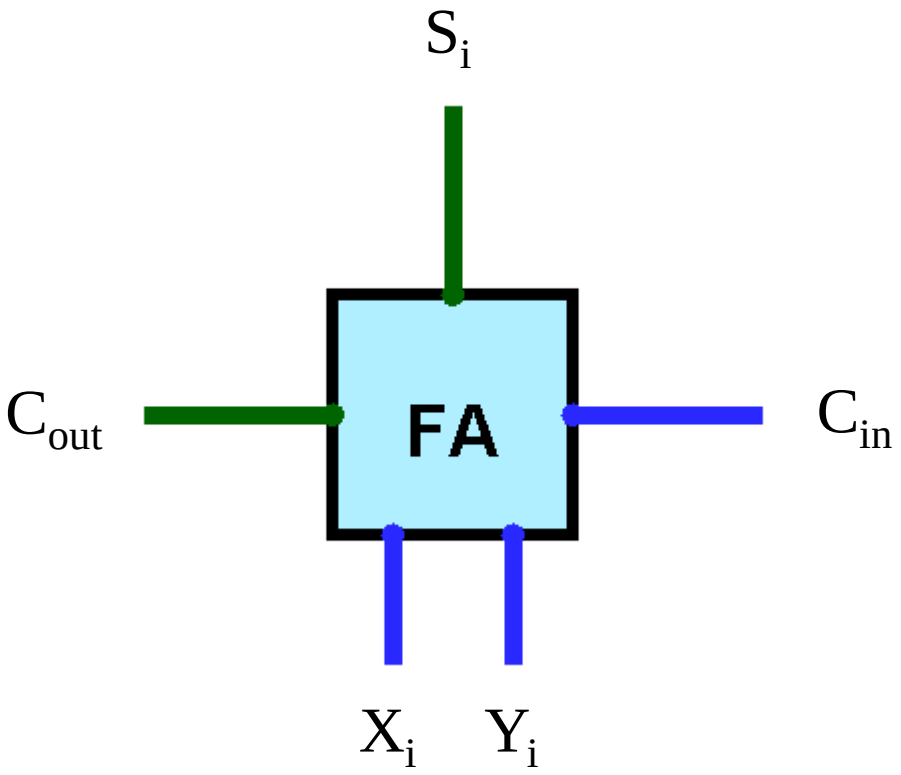
- 验证串行加法器逻辑实现
  - 能设计8位可控加减法电路
- 掌握快速加法器逻辑实现
  - 能设计4位先行进位电路

实验电路

- 实验涉及子电路
  - 8位可控加减法器（子电路1，需完成）
  - 4位先行进位74182（子电路2，需完成）

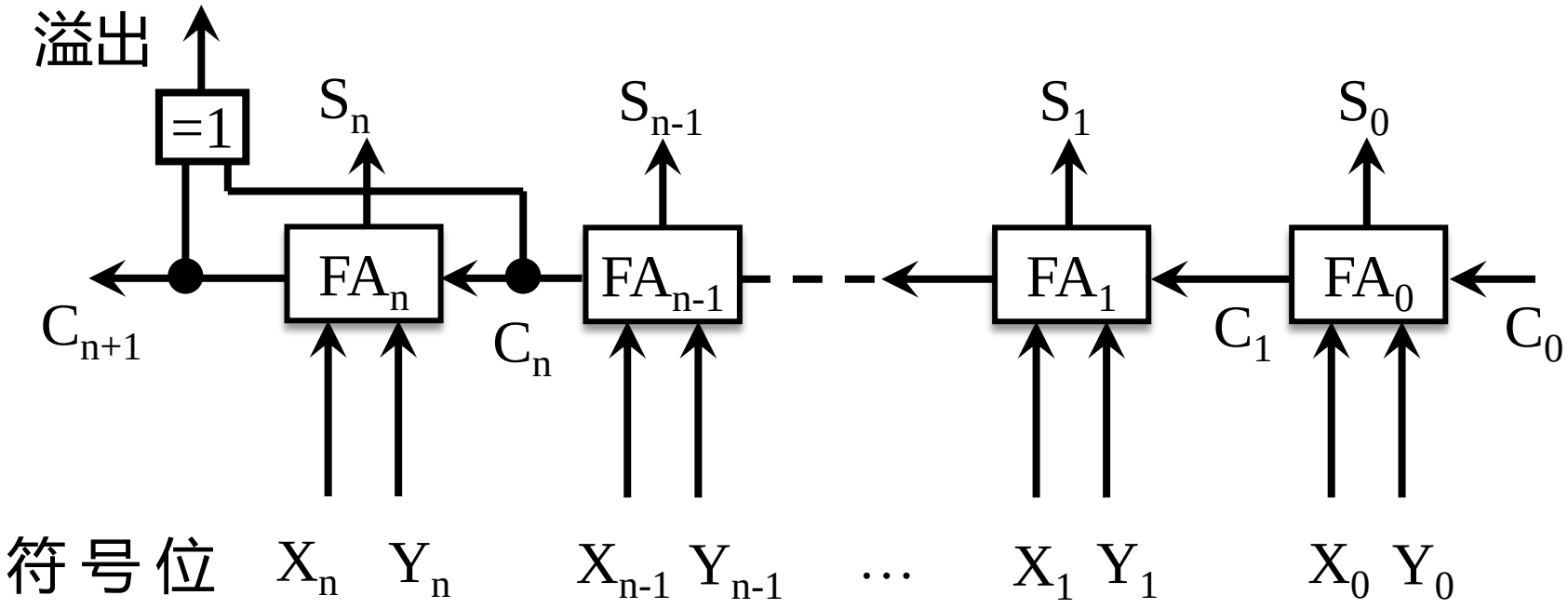
子电路1： 8位可控加减法器

- 一位全加器
  - S_i：和数
  - C_{out}：高位进位输出
  - C_{in}：低位进位输入
  - X_i，Y_i：操作数
  - $S_i = X_i \oplus Y_i \oplus C_{in}$
  - $C_{out} = X_i Y_i + (X_i \oplus Y_i) C_{in}$



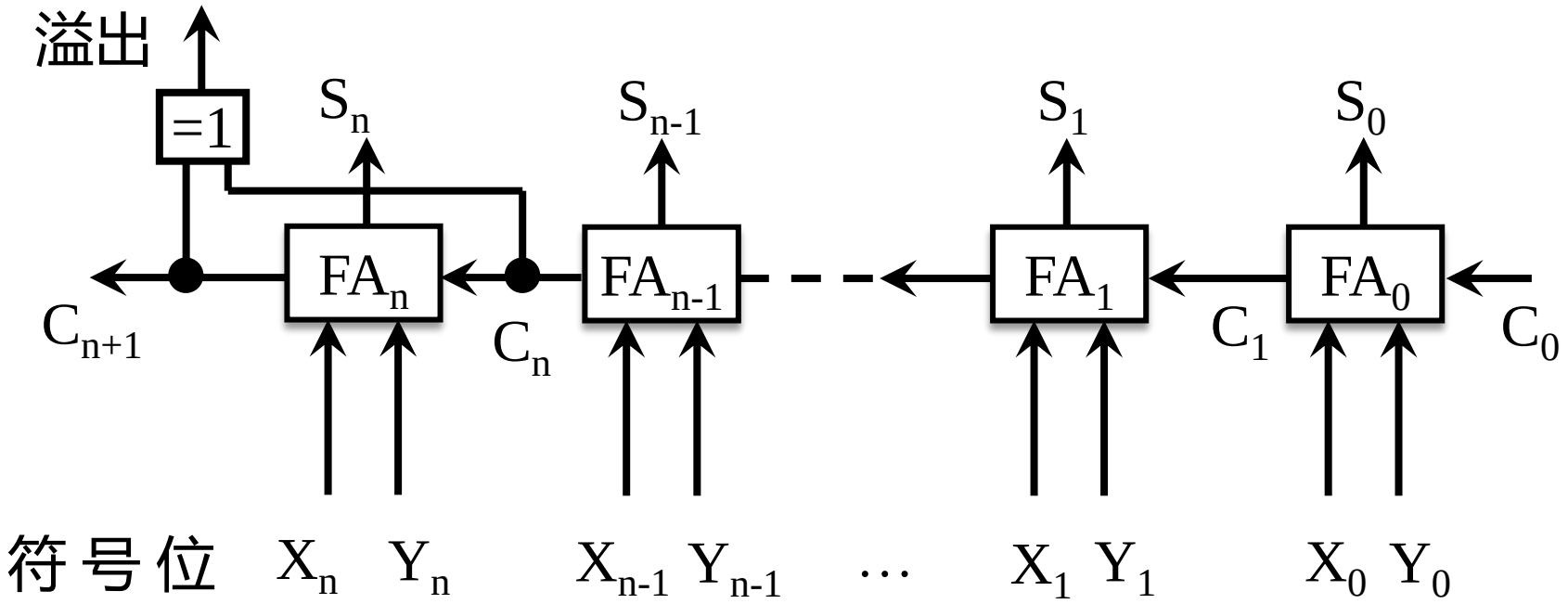
子电路1： 8位可控加减法器

- n+1位加法器
  - 最高位为符号位
  - 溢出判断
    - 数值最高位的进位 ⊕ 符号位的进位



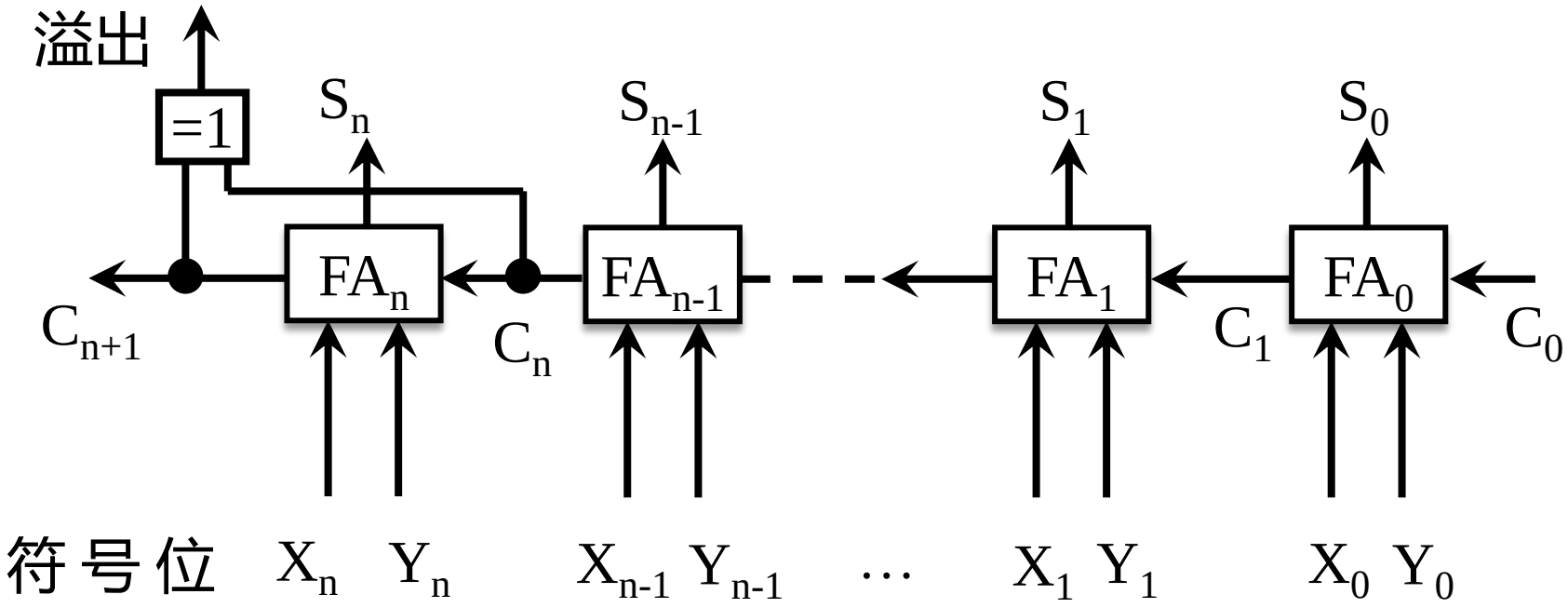
子电路1： 8位可控加减法器

- n+1位加法器
  - 输入：操作数X、Y
  - 作加法
    - $C_0 = 0$，$X = X_n \ldots X_0$，$Y' = Y_n \ldots Y_0$



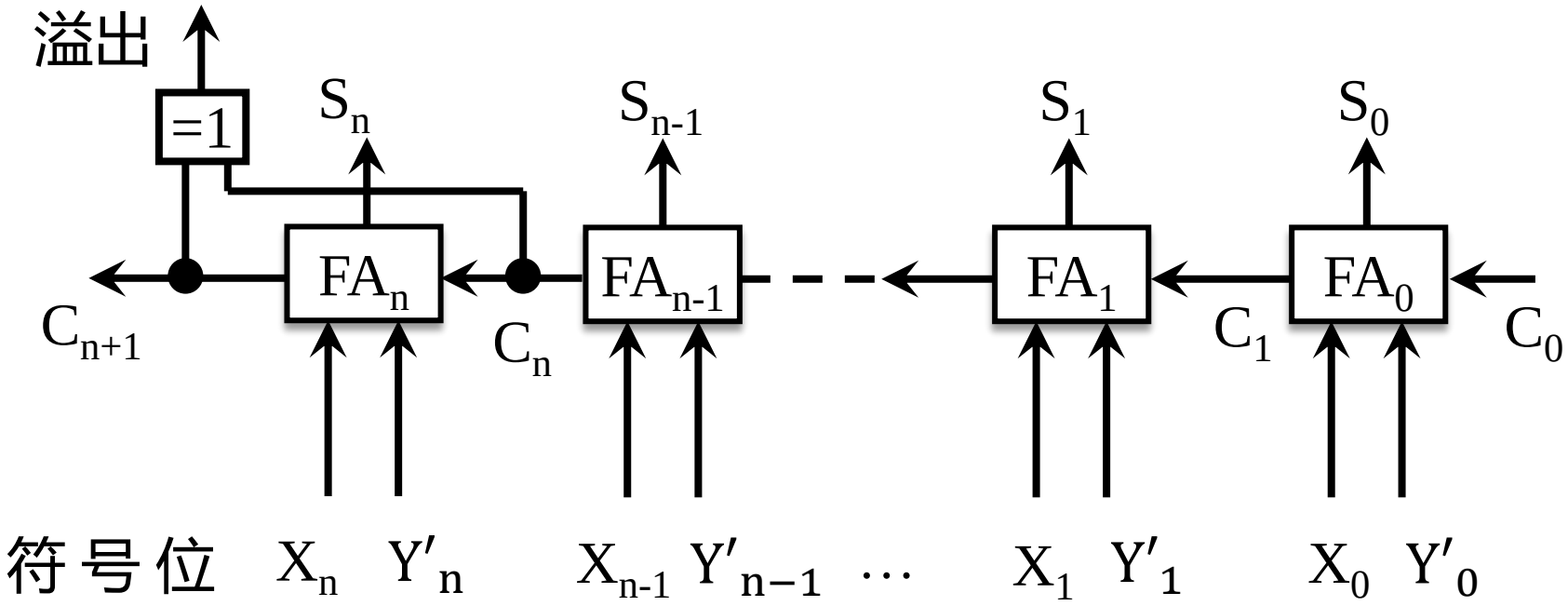
子电路1： 8位可控加减法器

- n+1位加法器
  - 输入：操作数X、Y
  - 作减法
    - $C_0 = 1$，$X = X_n \ldots X_0$，$Y' = \overline{Y_n} \ldots \overline{Y_0}$



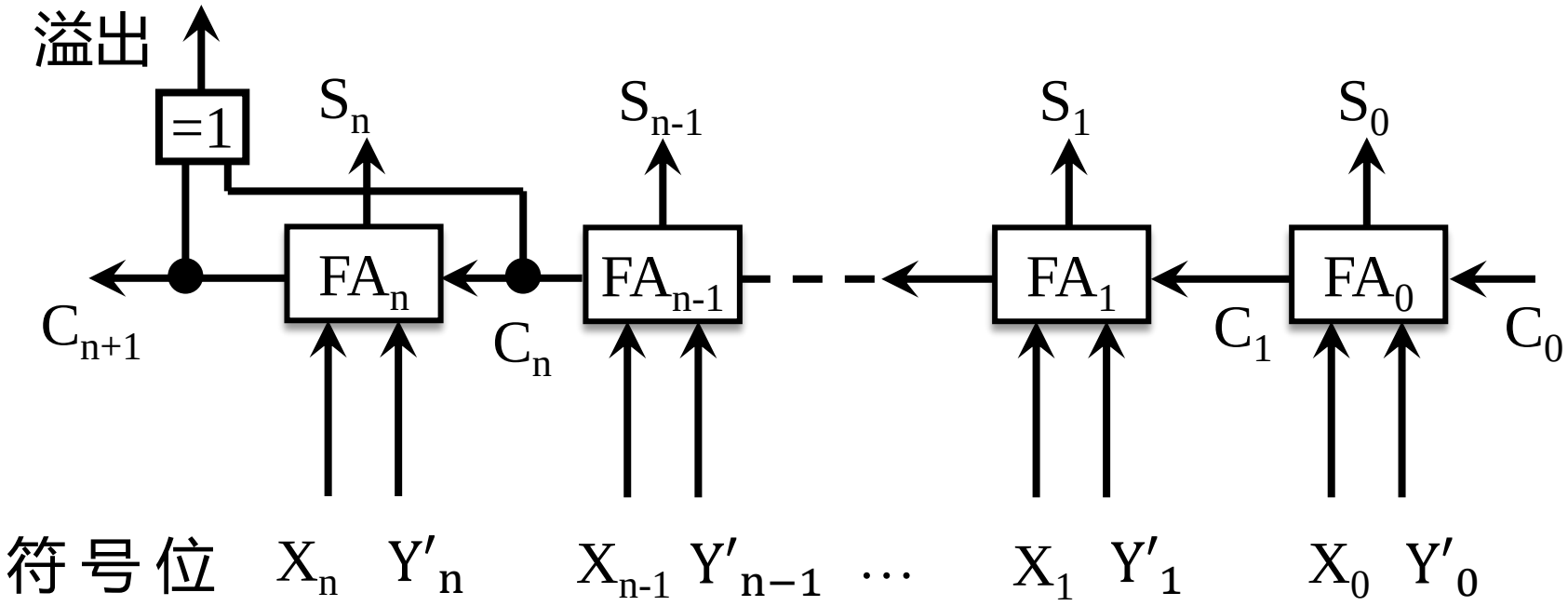
子电路1： 8位可控加减法器

- 可控加减法器
  - 引入减法标志Sub
  - Sub=0：作加法， Sub=1：作减法
  - 输入：操作数X、Y，减法标志：Sub



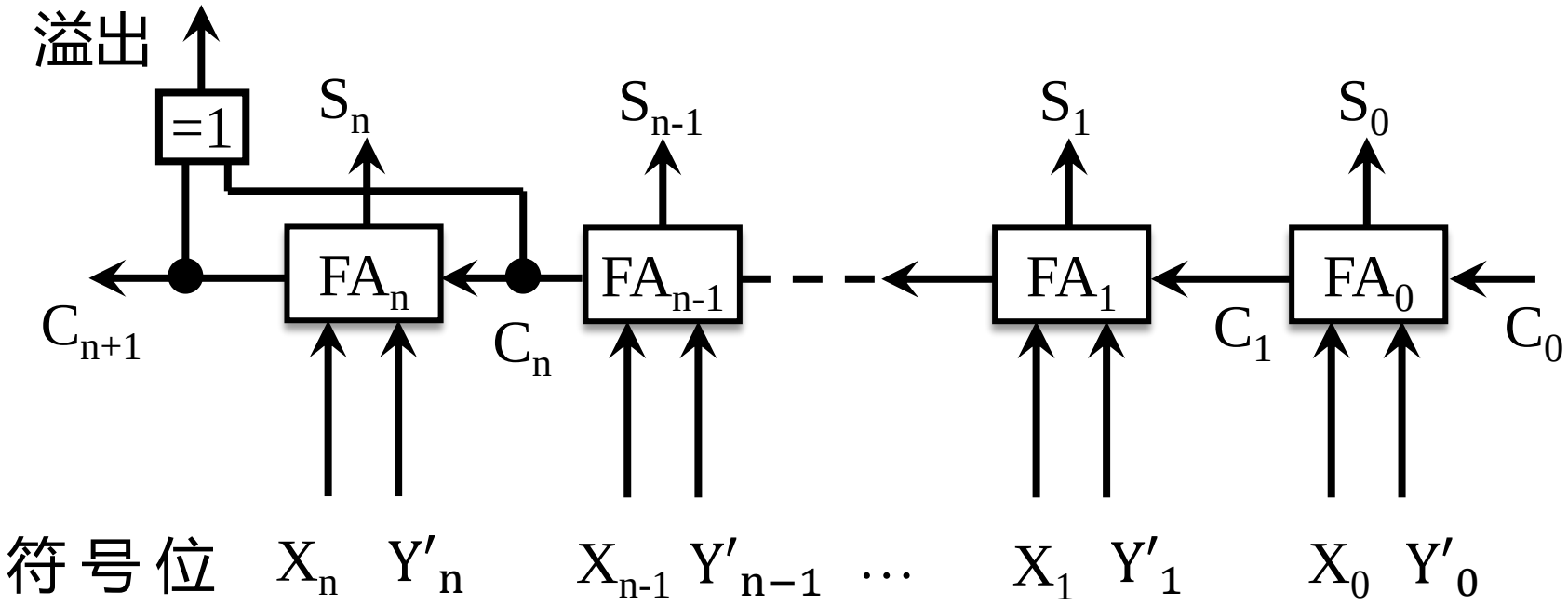
子电路1： 8位可控加减法器

- 可控加减法器
  - 输入：操作数X、Y，减法标志：Sub
  - 作加法， Sub=0
    - $C_0 = \text{Sub}$，$X = X_n \ldots X_0$，$Y'_i = Y_i \oplus \text{Sub}$



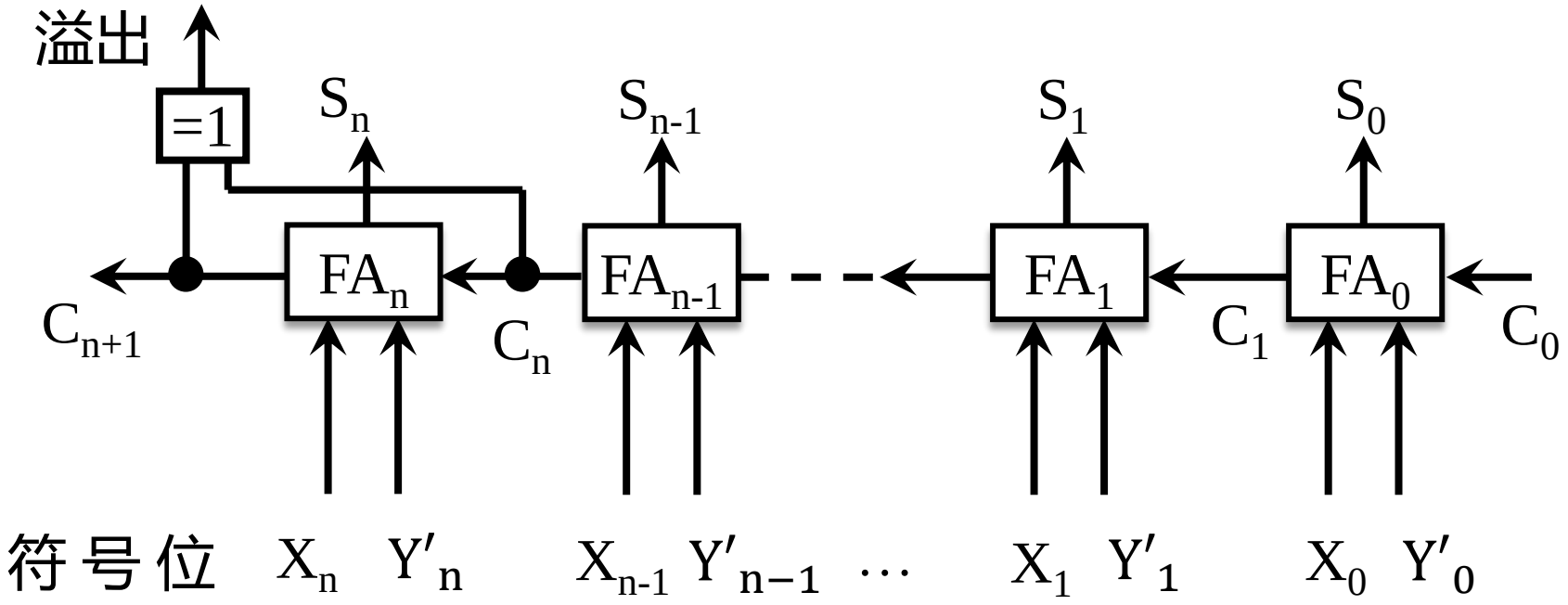
子电路1： 8位可控加减法器

- 可控加减法器
  - 输入：操作数X、Y，减法标志：Sub
  - 作减法， Sub=1
    - $C_0 = \text{Sub}$，$X = X_n \ldots X_0$，$Y'_i = Y_i \oplus \text{Sub}$



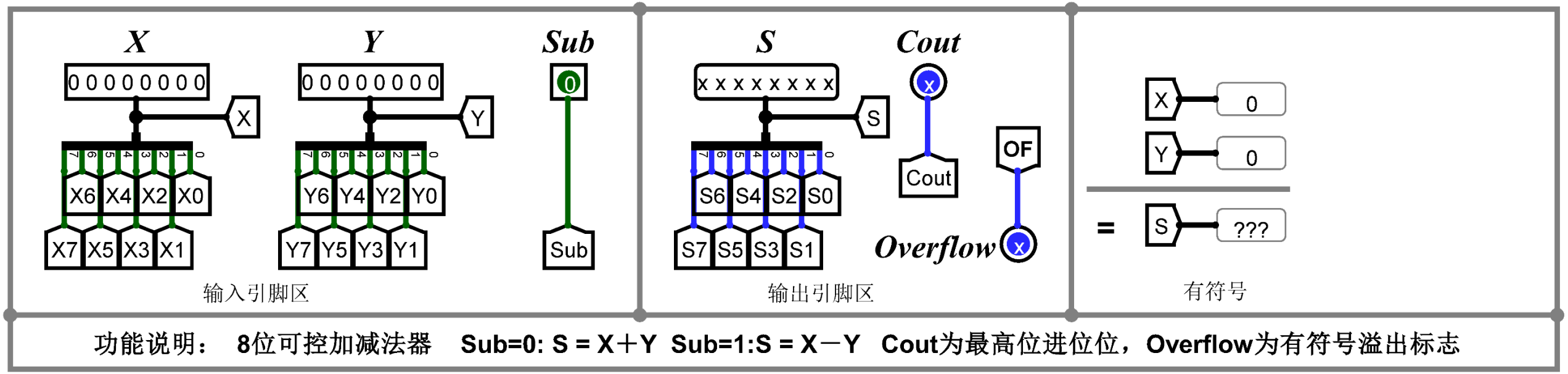
子电路1： 8位可控加减法器

- 可控加减法器
  - 输入：操作数X、Y，减法标志：Sub
  - $C_0 = \text{Sub}$，$X = X_n \ldots X_0$，$Y'_i = Y_i \oplus \text{Sub}$
  - 溢出：$C_{n+1} \oplus C_n$



子电路1： 8位可控加减法器

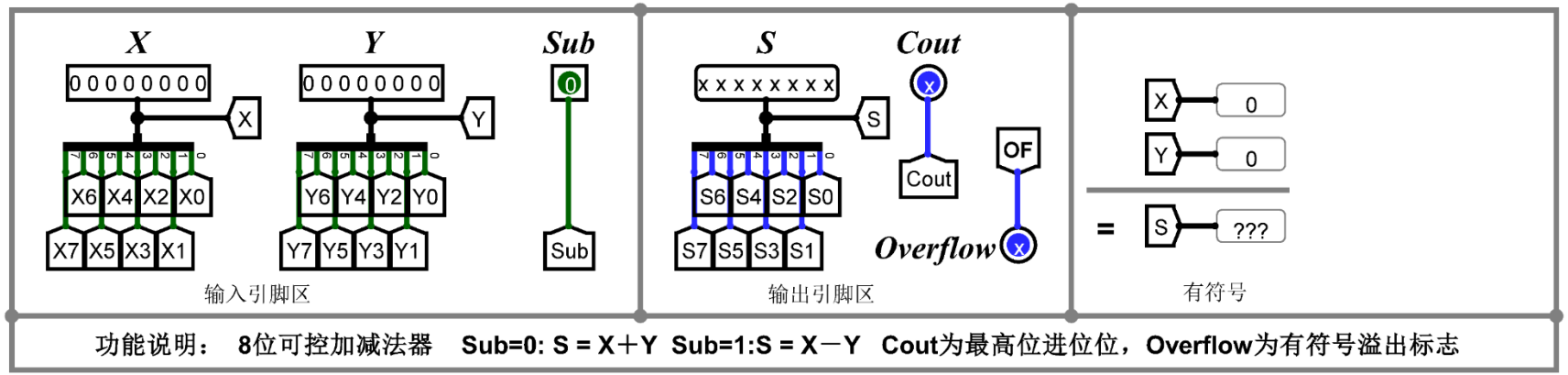
- 8位可控加减法器
  - 输入：8位操作数X、Y，减法标志：Sub
  - 输出： 8位和数S，最高位进位：Cout，溢出标志：Overflow
  - 请勿增改删引脚及子电路封装，使用隧道标签实现相应逻辑



子电路1： 8位可控加减法器

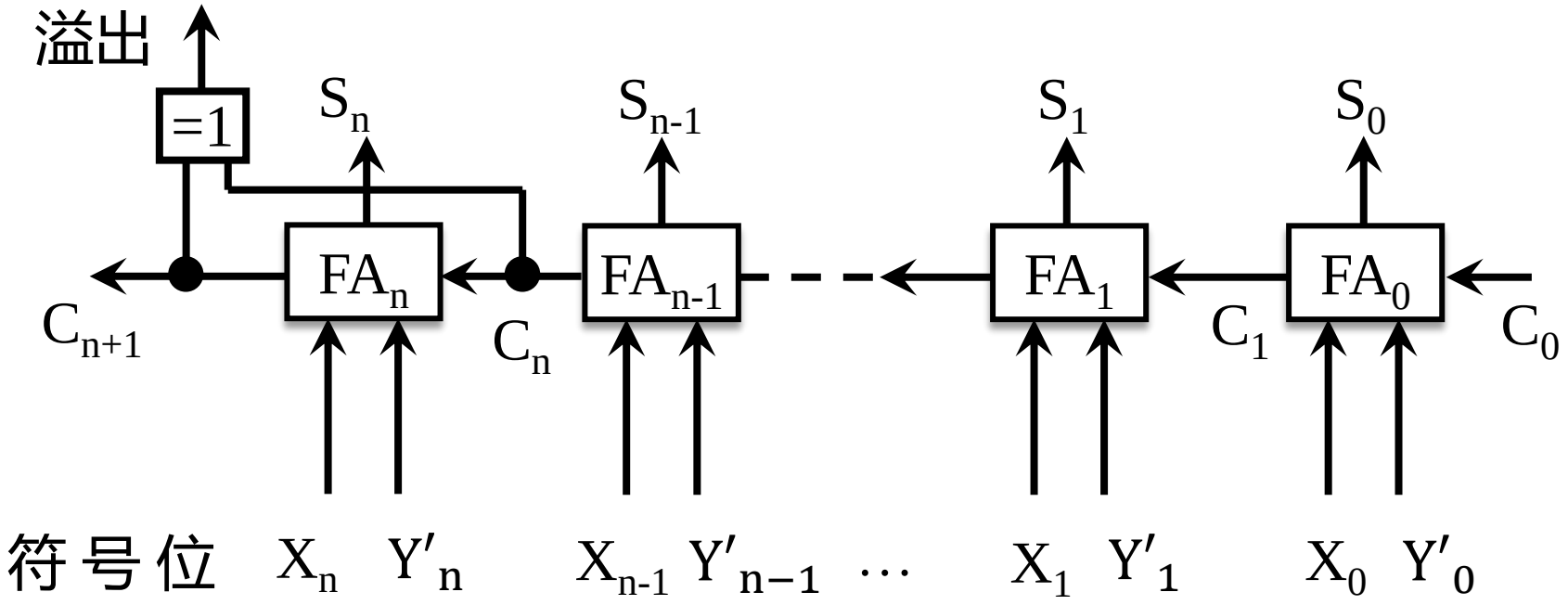
测试要求

- 作加法运算（共两张截图）
  - 当出现溢出和没有溢出时，观察得到的输出结果（8位和数S，最高位进位：Cout，溢出标志：Overflow）是否正确，各截图一张
- 作减法运算（共两张截图）
  - 当出现溢出和没有溢出时，观察得到的输出结果（8位和数S，最高位进位：Cout，溢出标志：Overflow）是否正确，各截图一张



子电路2：4位先行进位74182

- 串行进位链逻辑表达式分析
  - 进位链：传送进位的电路
  - $S_i = X_i \oplus Y_i \oplus C_i$
  - $C_{i+1} = X_i Y_i + (X_i \oplus Y_i) C_i$



子电路2：4位先行进位74182

- 串行进位链逻辑表达式分析
  - $S_i = X_i \oplus Y_i \oplus C_i$
  - $C_{i+1} = X_iY_i + (X_i \oplus Y_i)C_i$
    - $G_i = X_iY_i$
      - 进位生成函数Generate
    - $P_i = X_i \oplus Y_i$
      - 进位传递函数Propagate
    - P_i、G_i：只与操作数有关
  - $C_{i+1} = G_i + P_iC_i$

子电路2：4位先行进位74182

- 串行进位链逻辑表达式分析
  - $S_i = X_i \oplus Y_i \oplus C_i$
  - $C_{i+1} = G_i + P_i C_i$
    - P_i、G_i：只与操作数有关
- 高位运算依赖于低位进位
  - 计算不能并行
- 能否提前得到各位的进位输入？

子电路2：4位先行进位74182

- 串行进位链逻辑表达式分析
  - $C_1 = G_0 + P_0C_0$
  - $C_2 = G_1 + P_1C_1$

    $= G_1 + P_1(G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$
  - $C_3 = G_2 + P_2C_2$

    $= G_2 + P_2(G_1 + P_1G_0 + P_1P_0C_0) = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$
  - $C_4 = G_3 + P_3C_3$

    $= G_3 + P_3(G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0)$

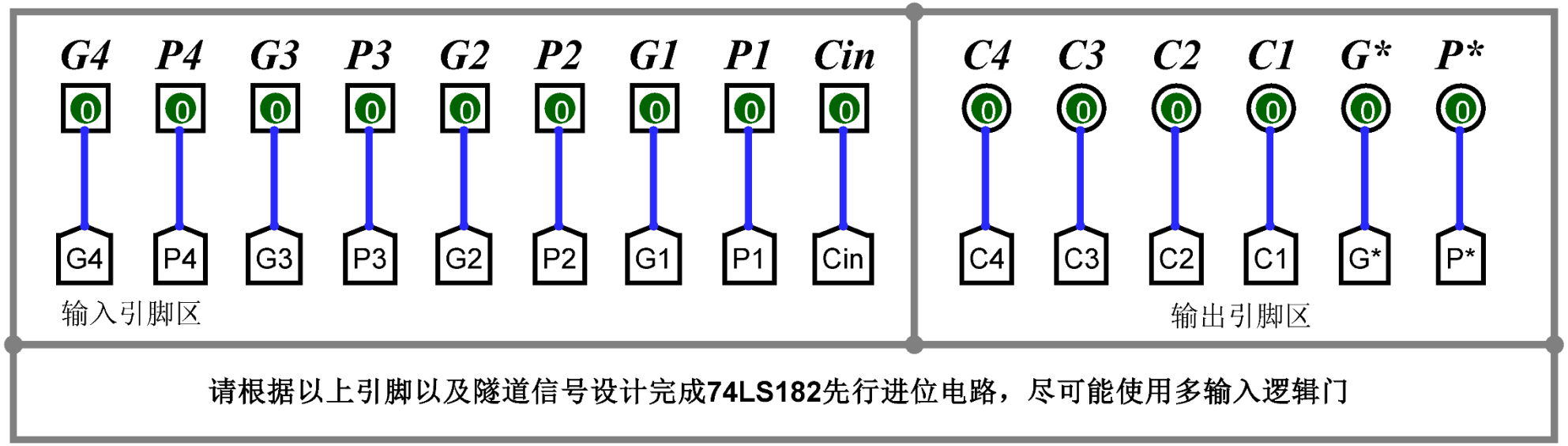
    $= G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$

子电路2：4位先行进位74182

- 串行进位链逻辑表达式分析
  - $C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$
  - $G^* = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$
    - 成组进位生成函数
    - 只与操作数有关
  - $P^* = P_3P_2P_1P_0$
    - 成组进位传递函数
    - 只与操作数有关
  - $C_4 = G^* + P^*C_0$
  - $C_1 = G_0 + P_0C_0$　上下两式形式一样

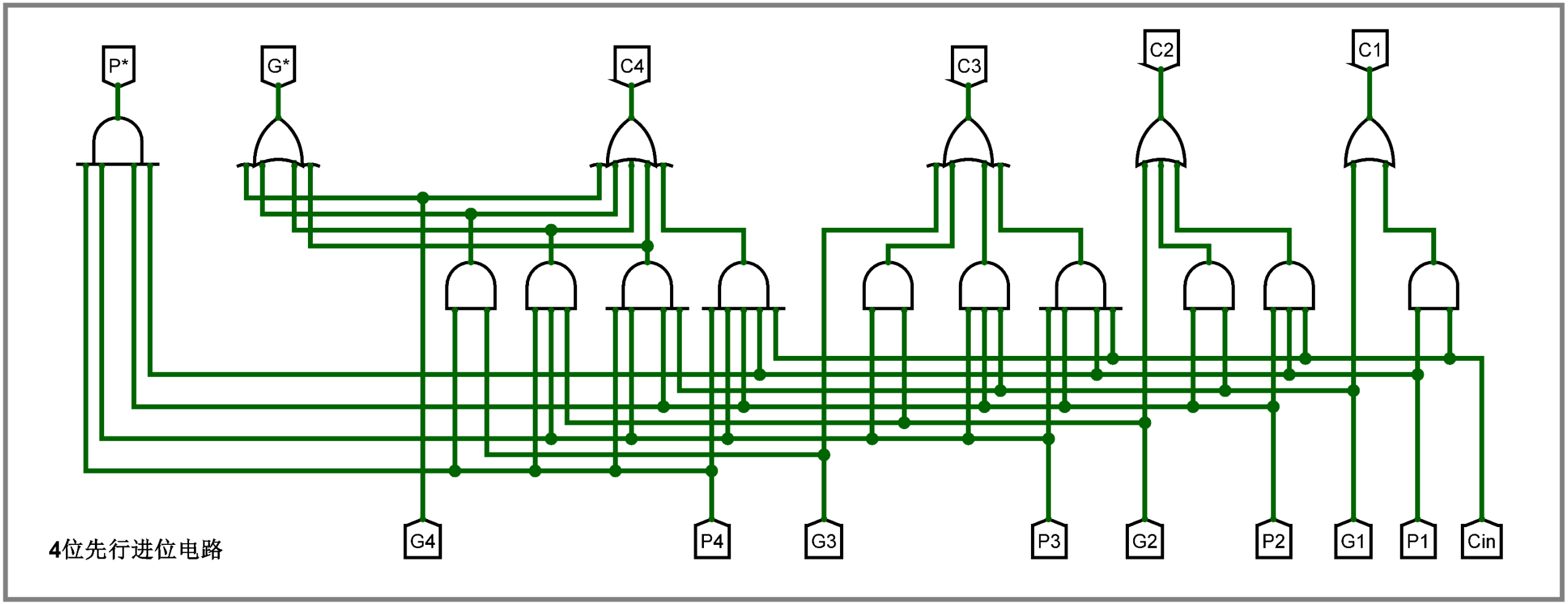
子电路2：4位先行进位74182

- 4位先行进位74182
  - 输入：进位生成函数$G_4 - G_1$，进位传递函数$P_4 - P_1$，最低位进位C_{in}
  - 输出： 进位输出$C_4 - C_1$，成组进位传递函数P*，成组进位生成函数G*
  - 请勿增改删引脚及子电路封装，使用隧道标签实现相应逻辑



子电路2：4位先行进位74182

- 4位先行进位74182
  - $C_0 = C_{in}$
  - $C_1 = G_1 + P_1 C_{in}$
  - $C_2 = G_2 + P_2 G_1 + P_2 P_1 C_{in}$
  - $C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_{in}$
  - $C_4 = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 C_{in}$
  - $G^* = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1$
  - $P^* = P_4 P_3 P_2 P_1$

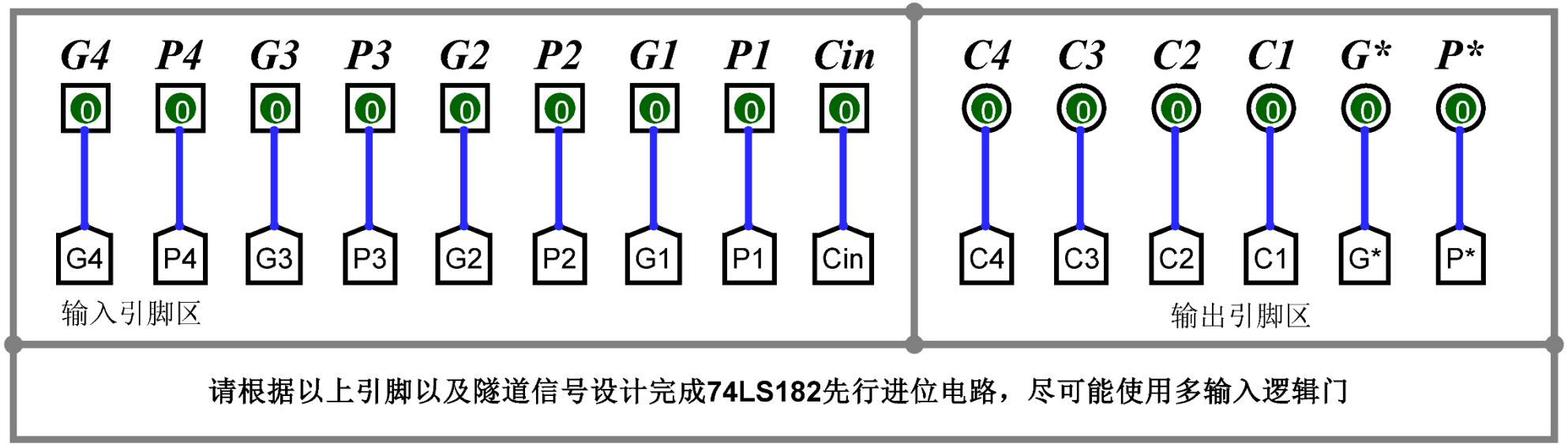


4位先行进位电路

子电路2：4位先行进位74182

测试要求

- 手动输入：进位生成函数$G_4 - G_1$，进位传递函数$P_4 - P_1$，最低位进位C_{in}
- 观察输出（进位输出$C_4 - C_1$，成组进位传递函数P*，成组进位生成函数G*）是否正确
- 截图一张



谢谢！