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

授课老师：吴炜滨

# 大纲

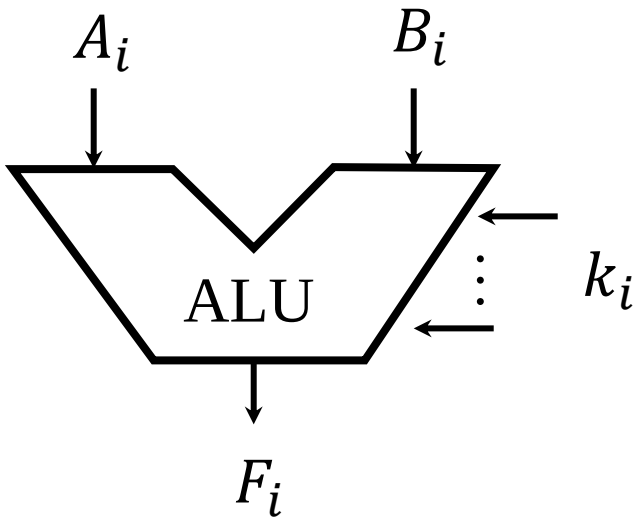
- 算术逻辑单元
  - ALU电路
  - 快速加法器

# 大纲

- 算术逻辑单元
  - ALU电路

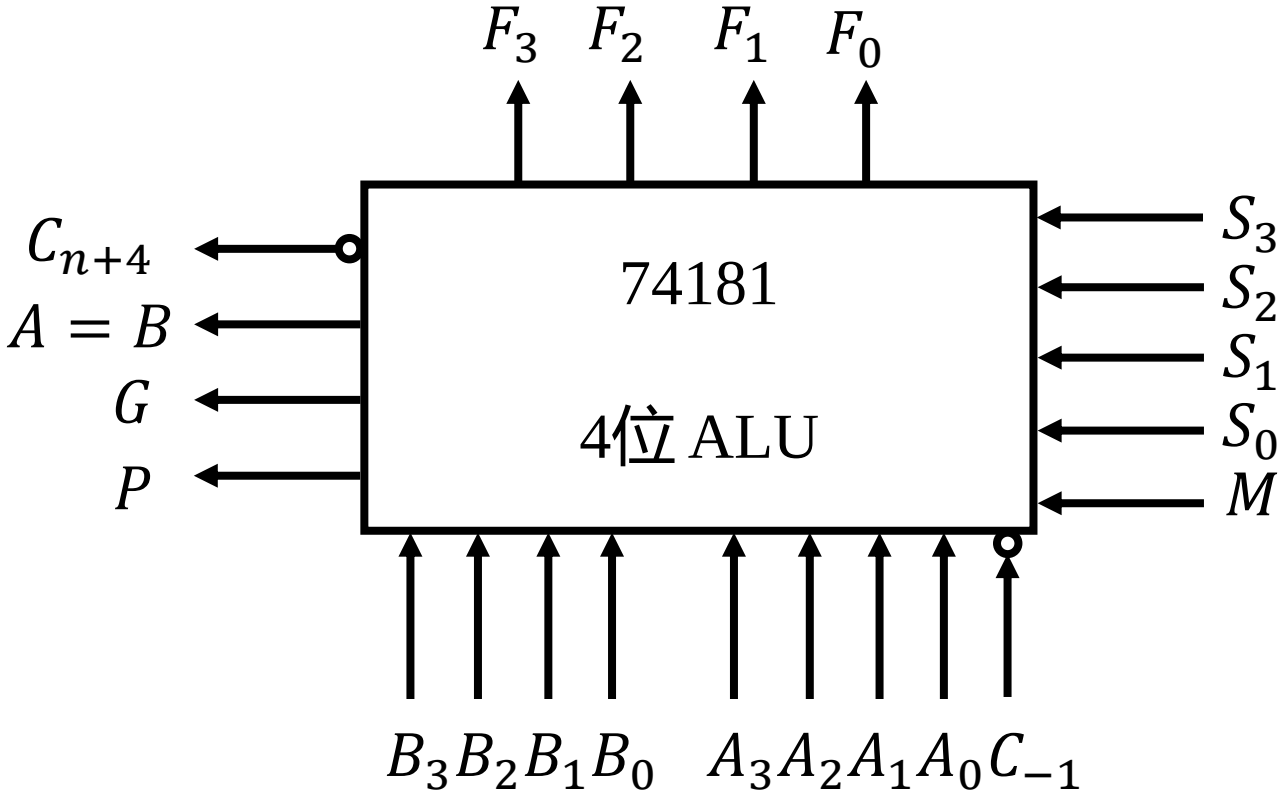
# ALU电路

- ALU：算术逻辑单元
  - 集成了算术运算和逻辑运算的运算电路
  - $A_i$，$B_i$：输入的操作数
  - $F_i$：运算结果
  - $k_i$不同取值决定做哪一种算术或逻辑运算
  - 组合逻辑电路
    - 没有记忆功能，输入消失，输出端的运算结果也会消失
    - 为了保存结果，使运算稳定，输入输出端都连接寄存器



# ALU电路

- 四位ALU 74181外特性
  - $M=0$：算术运算
  - $M=1$：逻辑运算
  - $S_3 \sim S_0$：不同取值，可做不同运算
  - $A_i$，$B_i$：输入的操作数
  - $F_i$：运算结果
  - $C_{-1}$：最低位的外来进位
  - $C_{n+4}$：向高位的进位
  - $G$、$P$：供先行进位使用（成组进位生成、传递函数）
  - $A=B$：判断两操作数是否相等

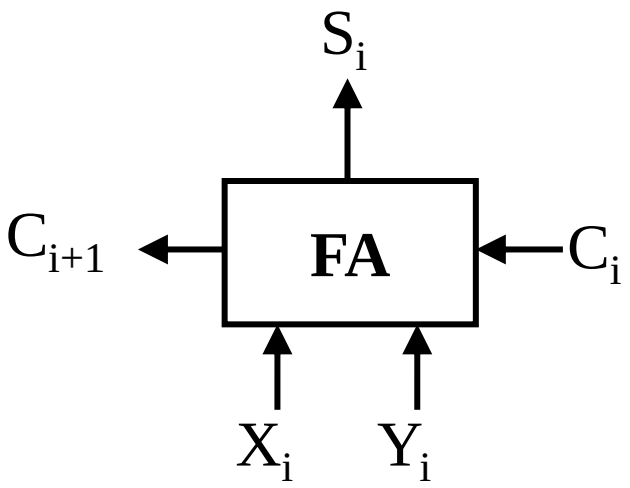


# 大纲

- 算术逻辑单元
  - 快速加法器
    - 全加器
    - 串行加法器
    - 并行进位链
    - 多位快速加法器

# 全加器

- 加法器
  - 算术运算电路的核心
  - 所有算术运算都基于加法器实现
- 一位全加器




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

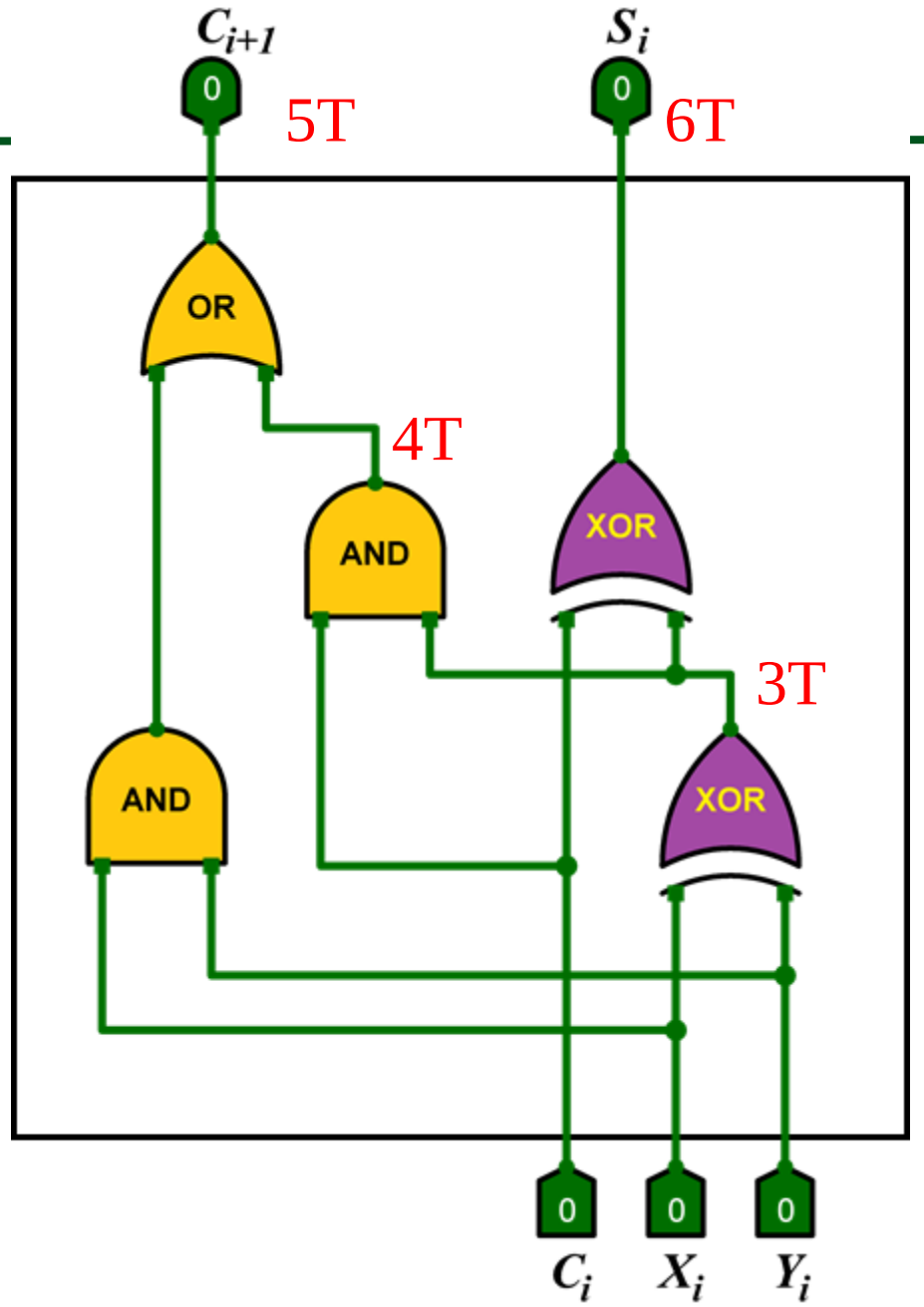
$$C_{i+1} = X_i Y_i + (X_i \oplus Y_i) C_i$$

# 全加器

- 一位全加器
  - 假设与门、或门的传播时间延迟是T
  - 异或门的传播时间延迟是3T
  - 和数$S_i$的时间延迟为
    - 6T
  - 进位输出$C_{i+1}$的时间延迟为
    - 5T

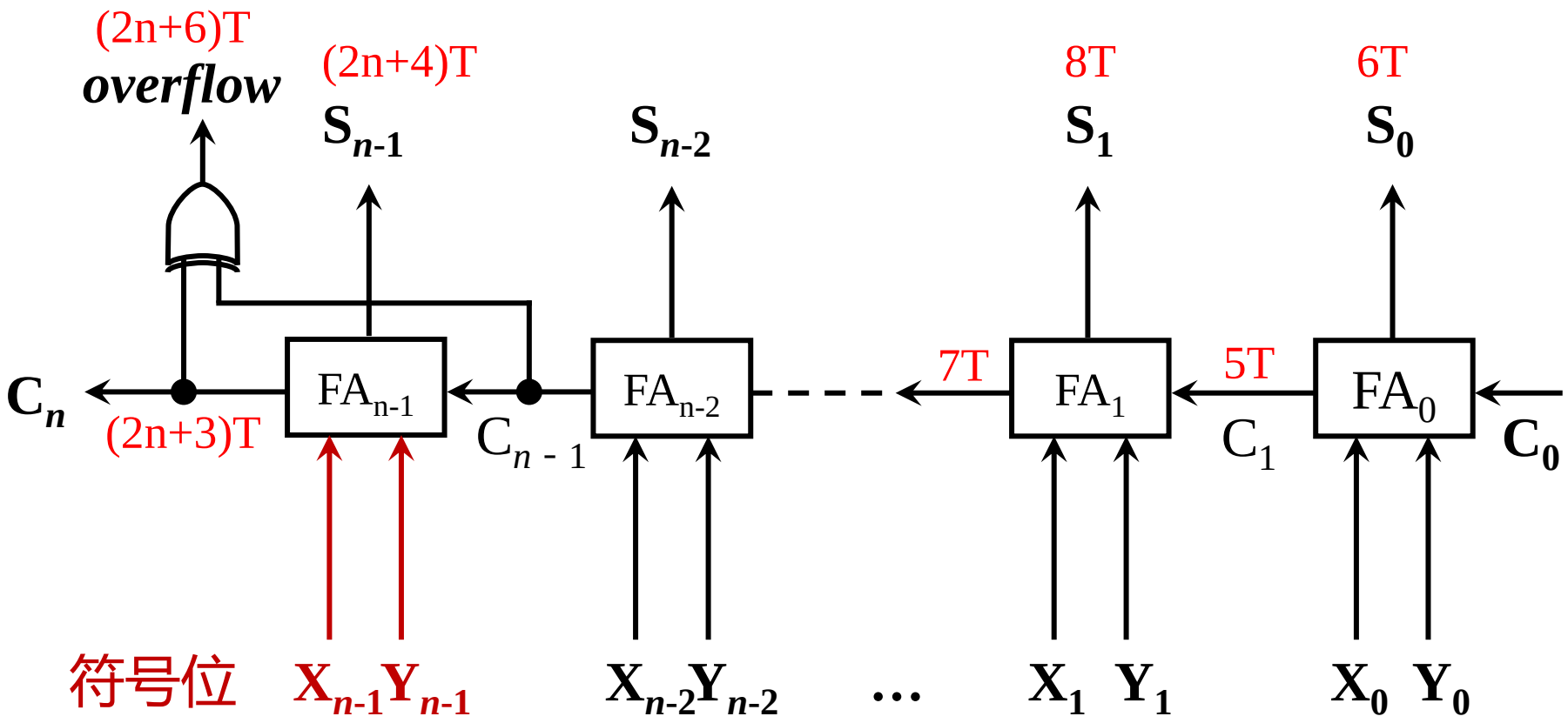
$$S_i = X_i \oplus Y_i \oplus C_i$$

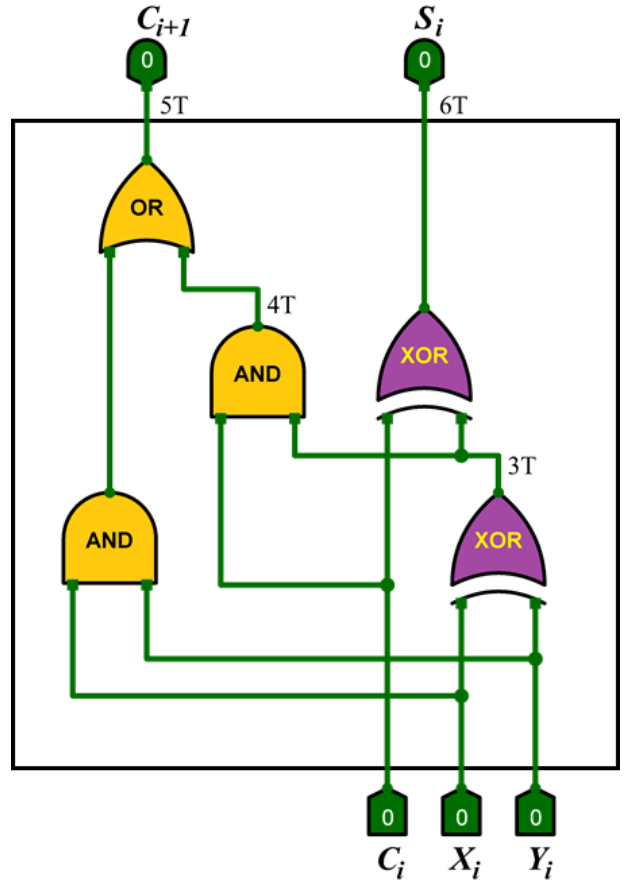
$$C_{i+1} = X_i Y_i + (X_i \oplus Y_i) C_i$$



# 串行加法器

- $n$位串行加法器
  - 假设与门、或门的传播时间延迟是T
  - 异或门的传播时间延迟是3T



# 串行加法器

- $n$位串行加法器
  - 高位全加器必须等待低位进位输入后才能开始运算，影响加法器速度
  - 能否提前产生各位的进位输入?
    - 使得各位的加法运算能并行起来，即可提高多位加法器运算速度
- 进位链
  - 传送进位的电路
    - 串行进位链：进位信号串行传递
    - 并行进位链（先行进位，跳跃进位）：进位信号同时产生

# 并行进位链

- 串行进位链
  - $C_{i+1} = X_i Y_i + (X_i \oplus Y_i) C_i$
    - $G_i = X_i Y_i$
      - 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
  - $C_{i+1} = G_i + P_i C_i$

# 并行进位链

- 串行进位链
  - $C_1 = G_0 + P_0C_0$
  - $C_2 = G_1 + P_1C_1$

    ……
  - $C_n = G_{n-1} + P_{n-1}C_{n-1}$

  - 高位进位依赖于低位进位
  - 计算不能并行，能否提前得到各位的进位输入?

# 并行进位链

- 改进串行进位链
  - $C_1 = G_0 + P_0C_0$
  - $C_2 = G_1 + P_1C_1$

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

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

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

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

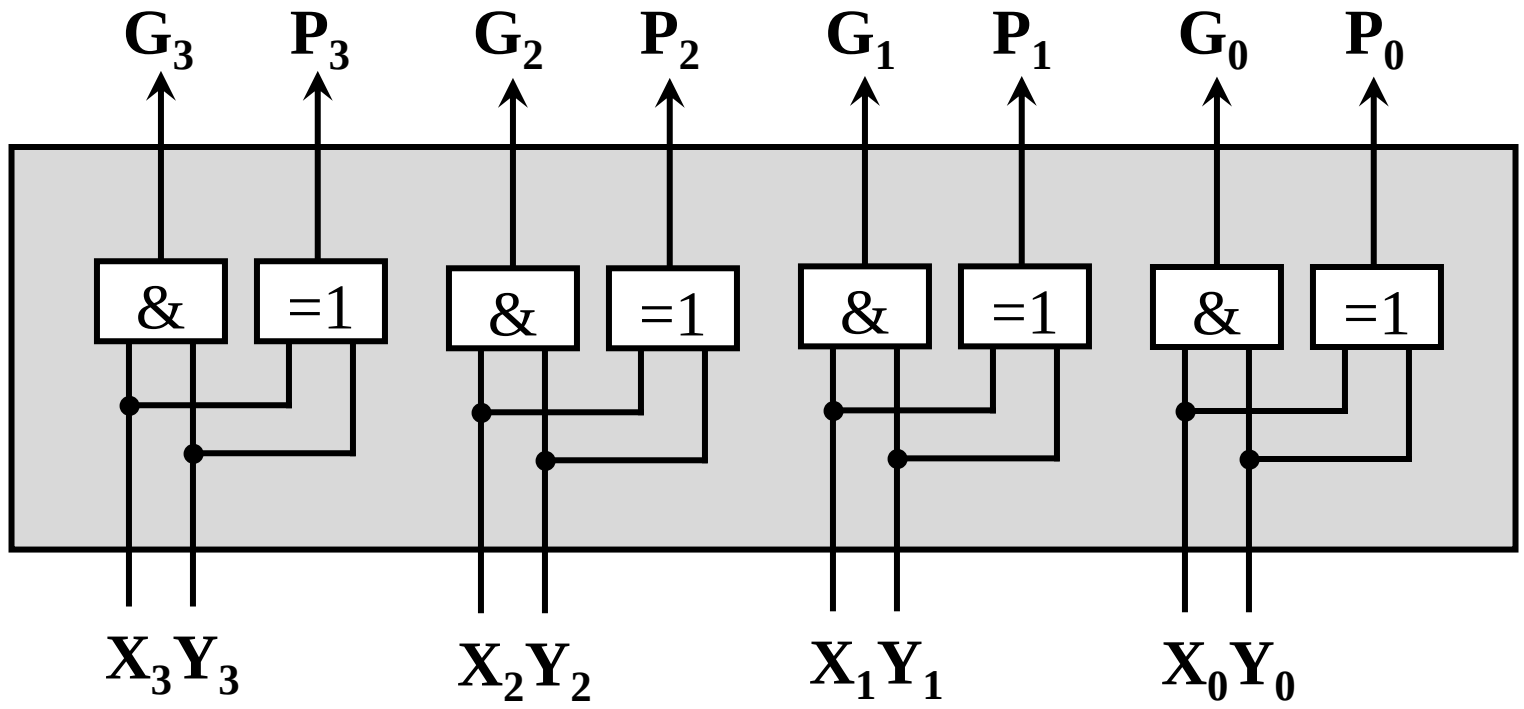
# 并行进位链

- 并行进位链
  - $C_n = G_{n-1} + P_{n-1}G_{n-2} + P_{n-1}P_{n-2}G_{n-3} + \ \ldots \ + P_{n-1}P_{n-2} \ldots P_1P_0C_0$
    - 进位输出$C_n$仅与最低位进位输入$C_0$有关
    - $G_i = X_iY_i$，$P_i = X_i \oplus Y_i$
      - $P_i$、$G_i$：只与操作数有关，可提前产生，之后各进位可并行产生
    - 所求进位位数$n$越高，该并行进位链电路复杂度越高
    - 通常按照4位一组进行分组运算
      - 组内先并行产生$P_3 - P_0$、$G_3 - G_0$，再并行产生各进位$C_4 - C_1$

# 并行进位链

- 与门异或门电路：生成所有$P_i$、$G_i$
  - 假设与门、或门的传播时间延迟是T
  - 异或门的传播时间延迟是3T

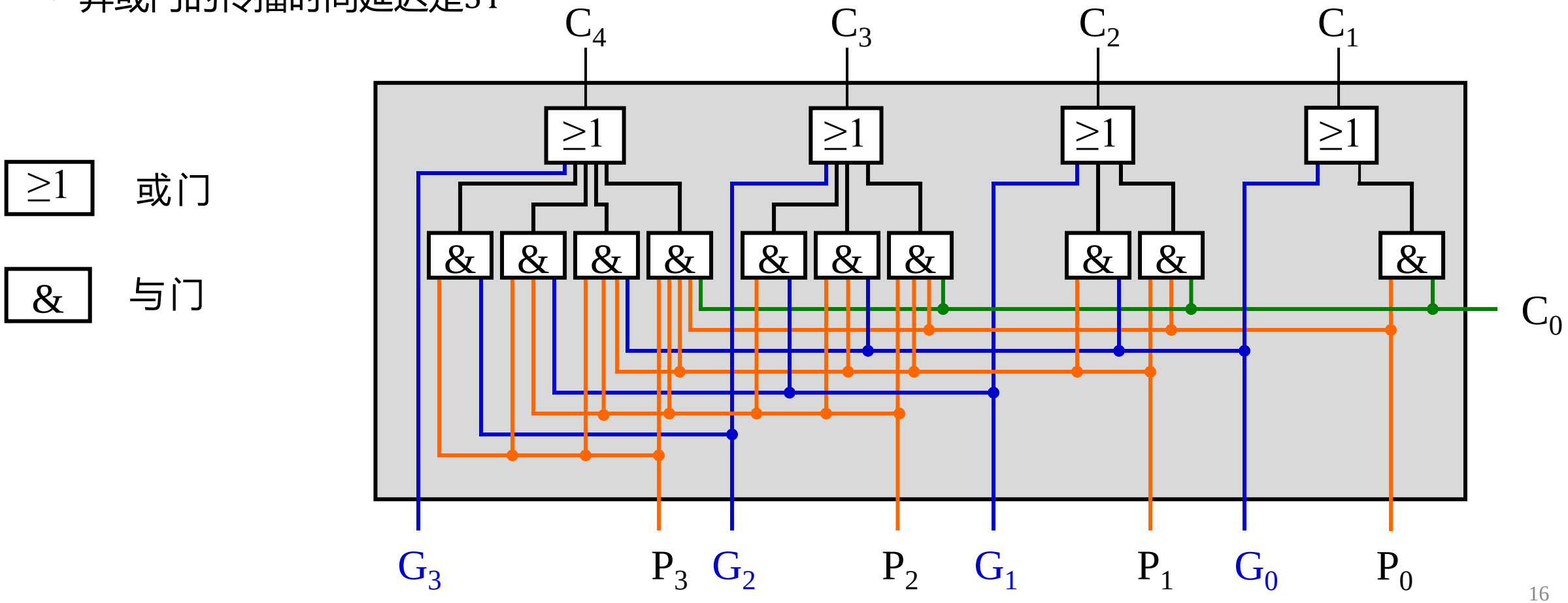
**总时间延迟3T**



# 并行进位链

- 先行进位电路：产生各进位
  - 假设与门、或门的传播时间延迟是T
  - 异或门的传播时间延迟是3T

**总时间延迟2T**



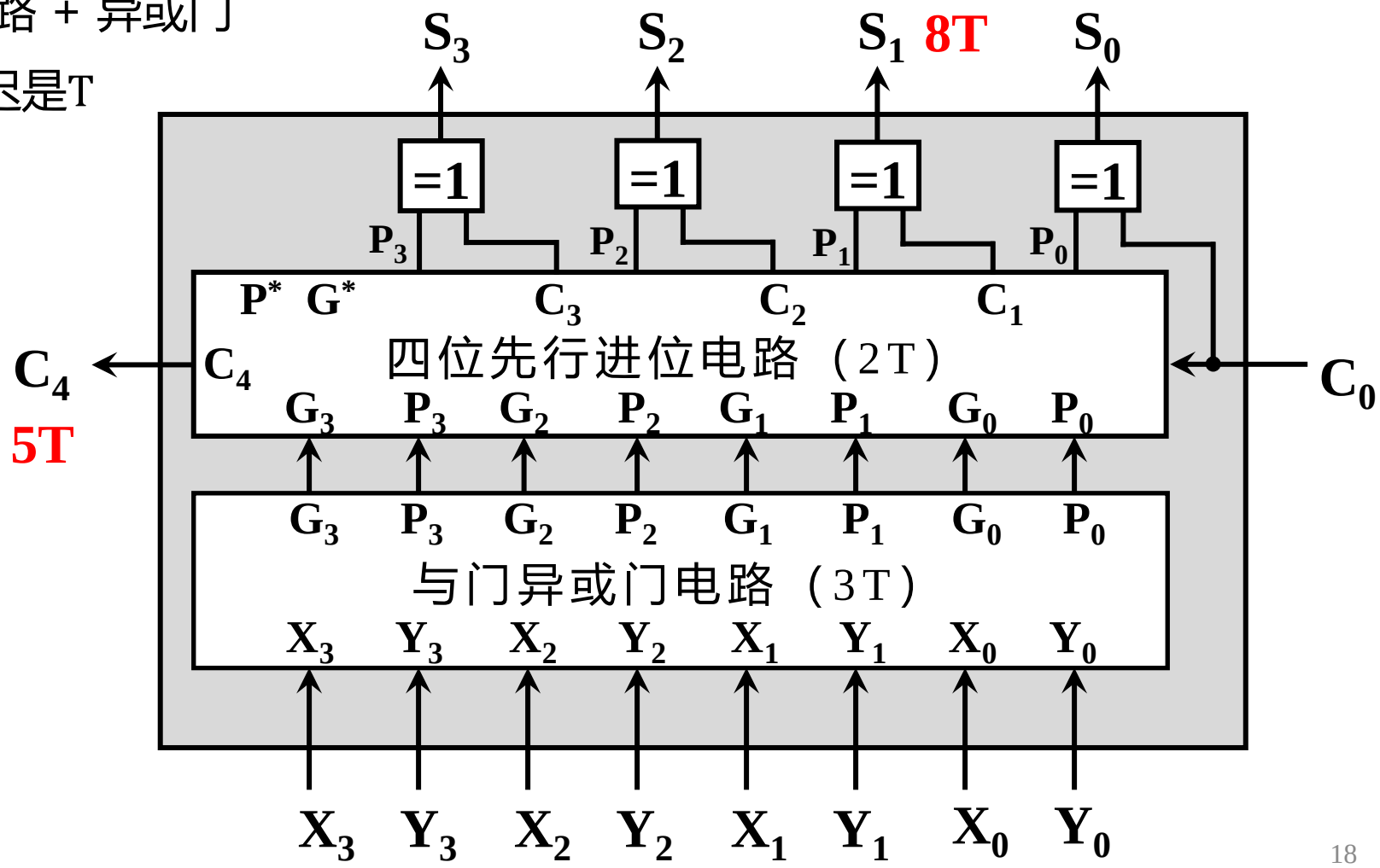
# 4位快速加法器

- 和数?
  - $S_3 = X_3 \oplus Y_3 \oplus C_3 = P_3 \oplus C_3$
  - $S_2 = X_2 \oplus Y_2 \oplus C_2 = P_2 \oplus C_2$
  - $S_1 = X_1 \oplus Y_1 \oplus C_1 = P_1 \oplus C_1$
  - $S_0 = X_0 \oplus Y_0 \oplus C_0 = P_0 \oplus C_0$
  - 进位信号得到后，求和只需一级异或门（经过3T）即可完成

# 4位快速加法器

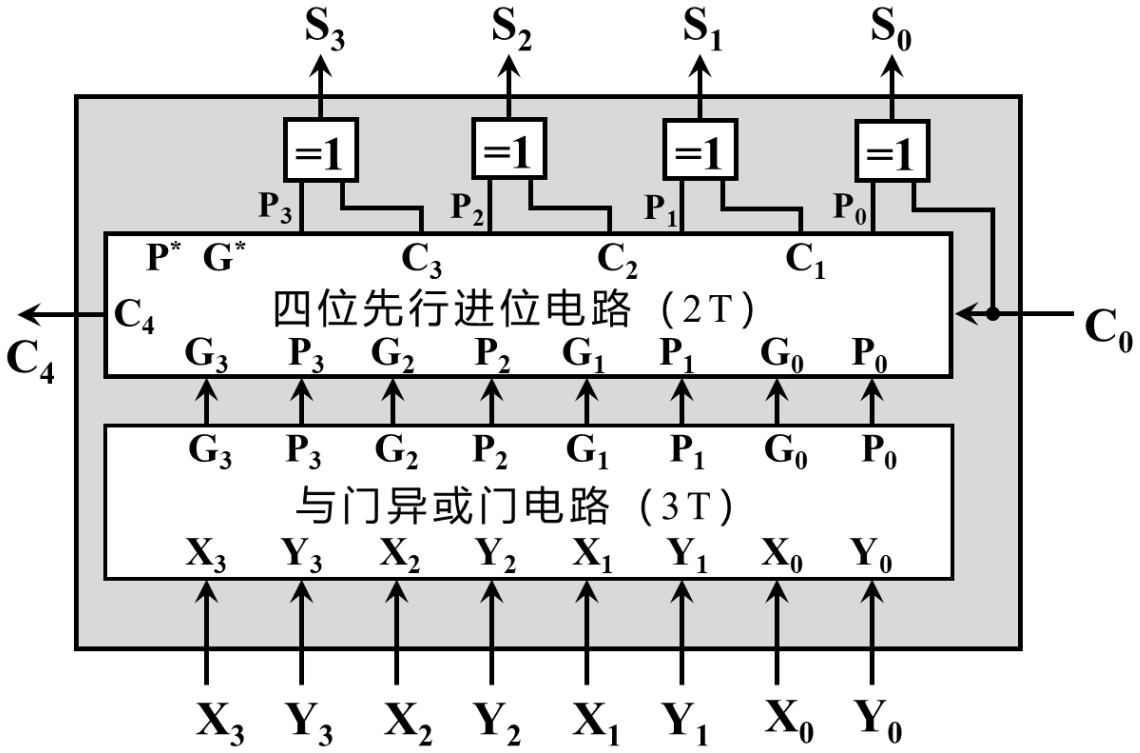
- 组 成
  - 与门异或门电路 + 先行进位电路 + 异或门
  - 假设与门、或门的传播时间延迟是T
  - 异或门的传播时间延迟是3T

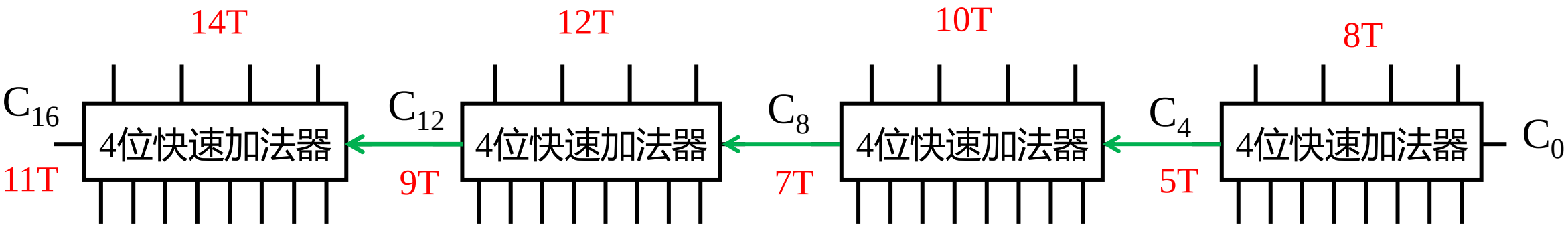
=1 异或门



# 16位快速加法器

- 16位加法器
  - 组内先行进位
  - 组间串行进位
  - 可否组间并行?
    - 能否并行产生$C_{16}$，$C_{12}$，$C_8$，$C_4$?



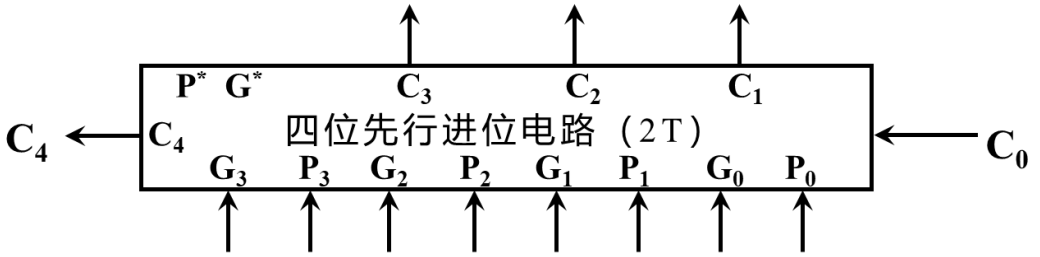
# 16位快速加法器

## 并行进位链

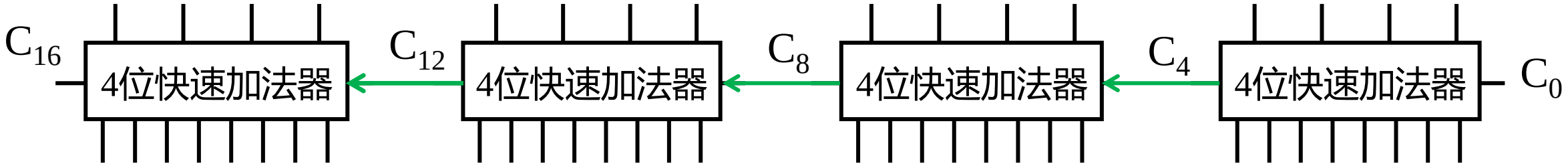
- $C_4 = \boxed{G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0} + \boxed{P_3P_2P_1P_0}C_0$
- $G^* = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$
  - 成组进位生成函数
- $P^* = P_3P_2P_1P_0$
  - 成组进位传递函数
- $G^*$、 $P^*$：只与操作数有关
- $C_4 = G^* + P^*C_0$
  - 与$C_1 = G_0 + P_0C_0$形式相同
  - $C_4$可由$G^*$，$P^*$，$C_0$直接得到

# 16位快速加法器

- 进位链逻辑表达式分析
  - $C_1 = G_0 + P_0C_0$
  - $C_2 = G_1 + P_1C_1$
  - $C_3 = G_2 + P_2C_2$
  - $C_4 = G_3 + P_3C_3$




- 构建4位快速加法器：组内先行进位
  - 并行得到各小组内$G$，$P$（只与操作数有关）
  - 利用先行进位电路并行得到各小组内进位$C_4$，$C_3$，$C_2$，$C_1$



# 16位快速加法器

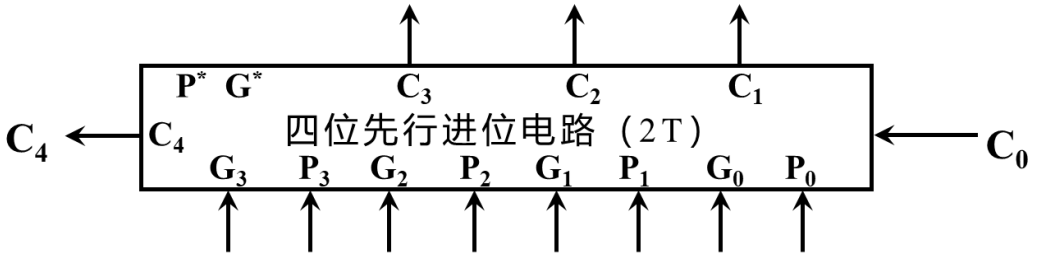
- 进位链逻辑表达式分析
  - $C_1 = G_0 + P_0C_0$
  - $C_2 = G_1 + P_1C_1$
  - $C_3 = G_2 + P_2C_2$
  - $C_4 = G_3 + P_3C_3$

$$C_4 = G_0^* + P_0^*C_0$$

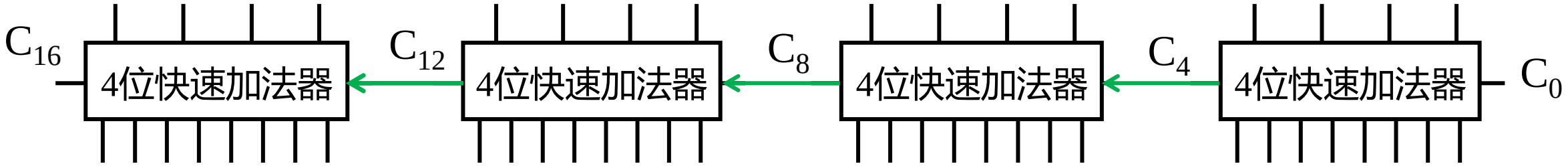
$$C_8 = G_1^* + P_1^*C_4$$

$$C_{12} = G_2^* + P_2^*C_8$$

$$C_{16} = G_3^* + P_3^*C_{12}$$



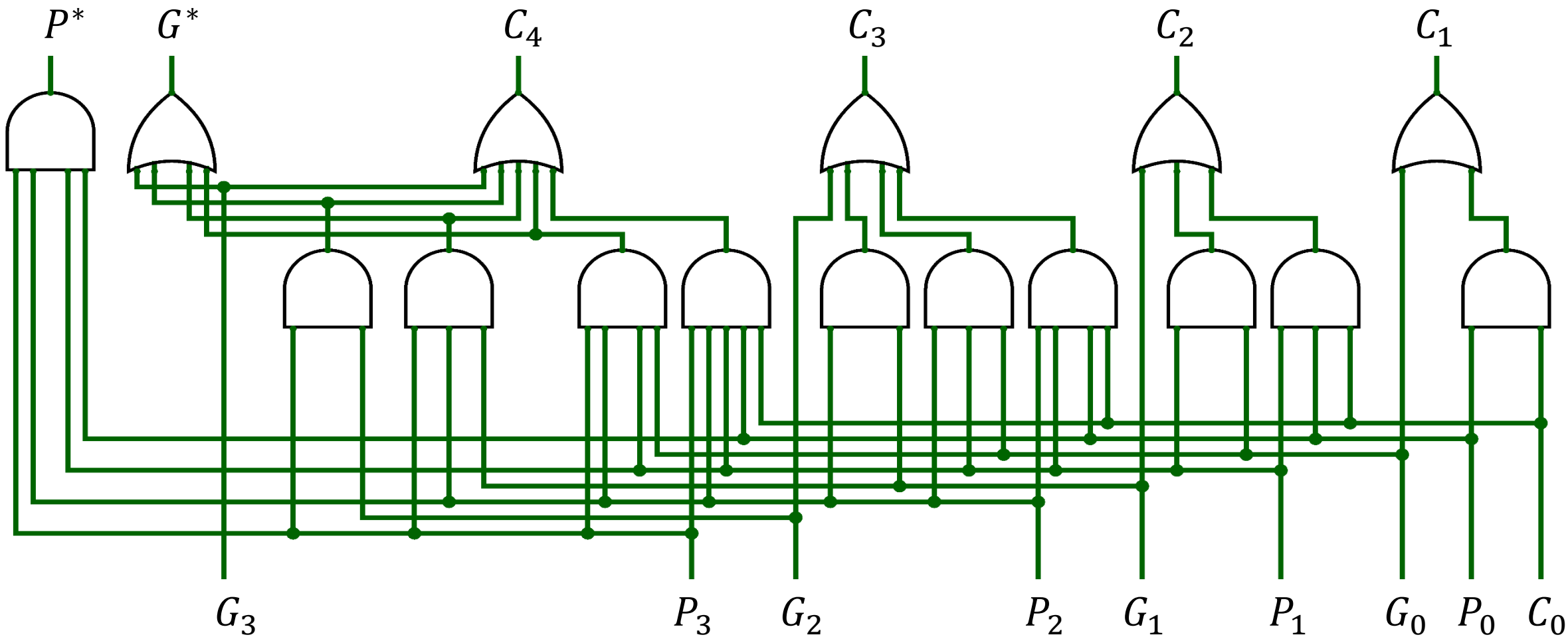

- 构建16位快速加法器：组间先行进位
  - 并行得到各小组内$G^*$，$P^*$（只与操作数有关）
  - 利用先行进位电路并行得到各小组间进位$C_{16}$，$C_{12}$，$C_8$，$C_4$



# 16位快速加法器

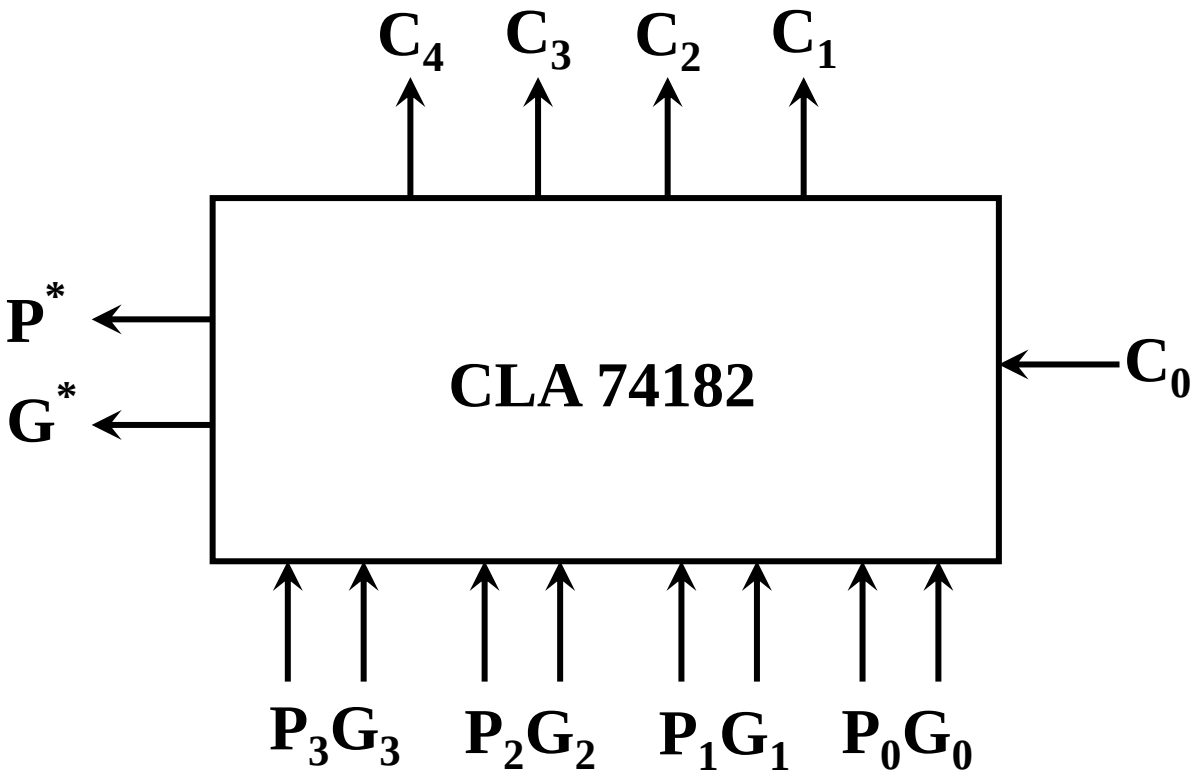
- 可级联的先行进位电路
  - 假设与门、或门的传播时间延迟是T

**总时间延迟2T**



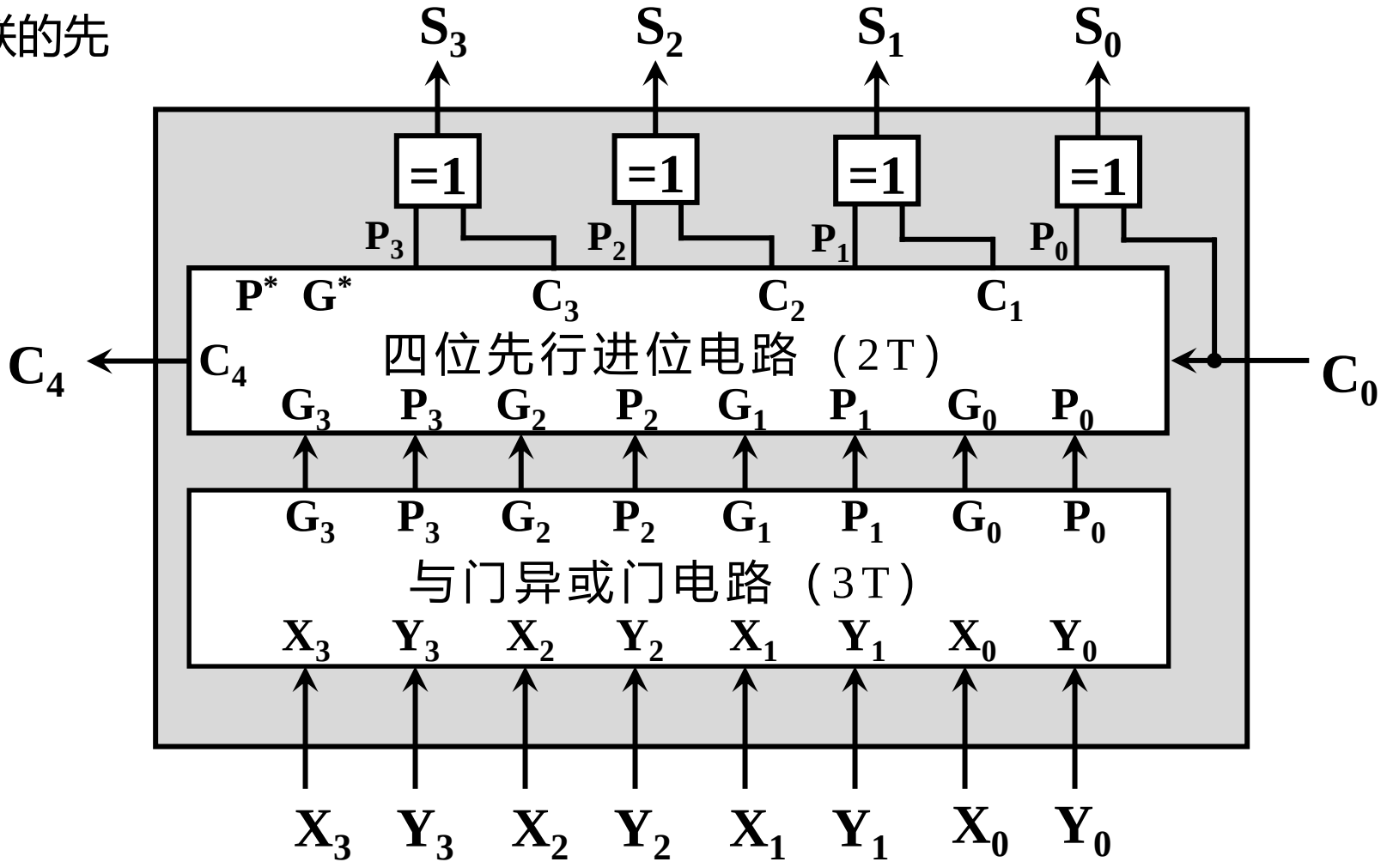
# 16位快速加法器

- 可级联的先行进位电路：先行进位芯片 CLA74182
  - 输入
    - 组内进位生成函数$G_3 - G_0$
    - 组内进位传递函数$P_3 - P_0$
    - 最低位进位输入$C_0$
  - 输出
    - 先行进位输出 $C_4 - C_1$
    - 成组进位传递函数P*
    - 成组进位生成函数G*
  - 2级门电路延迟



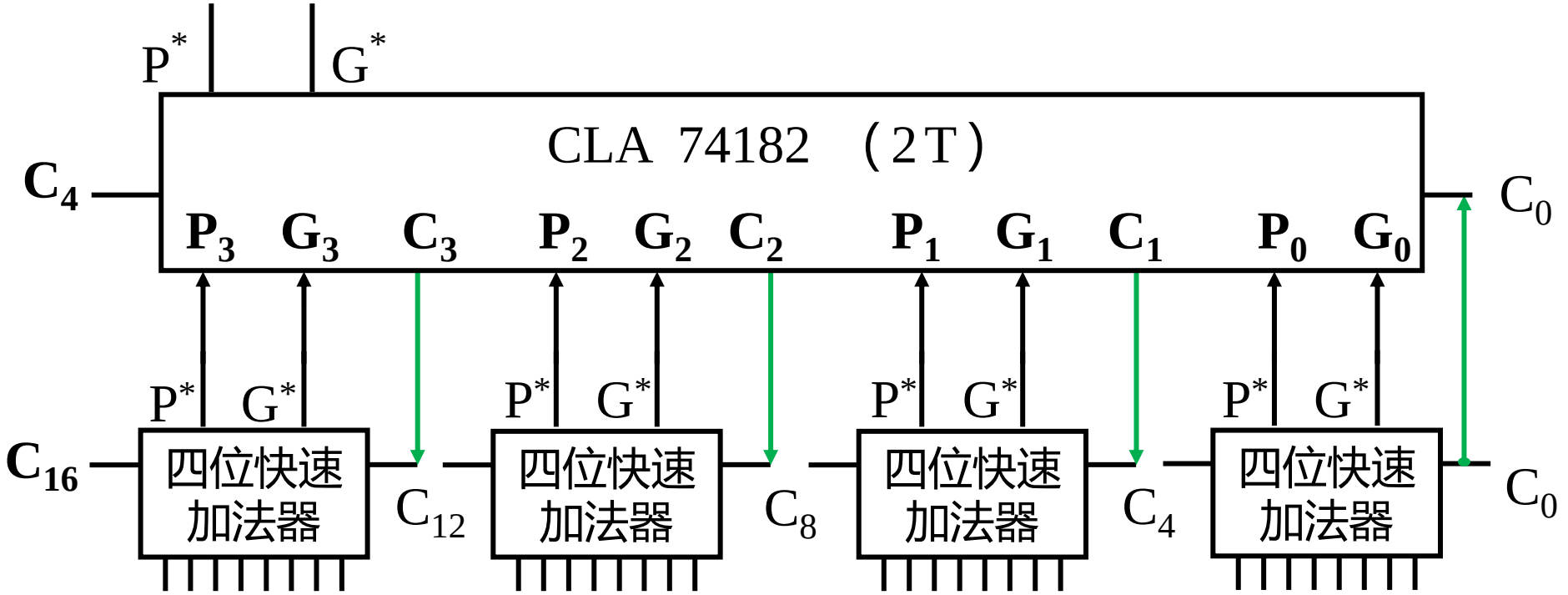
# 16位快速加法器

- 可级联的4位快速加法器
  - 与门异或门电路 + 可级联的先行进位电路 + 异或门



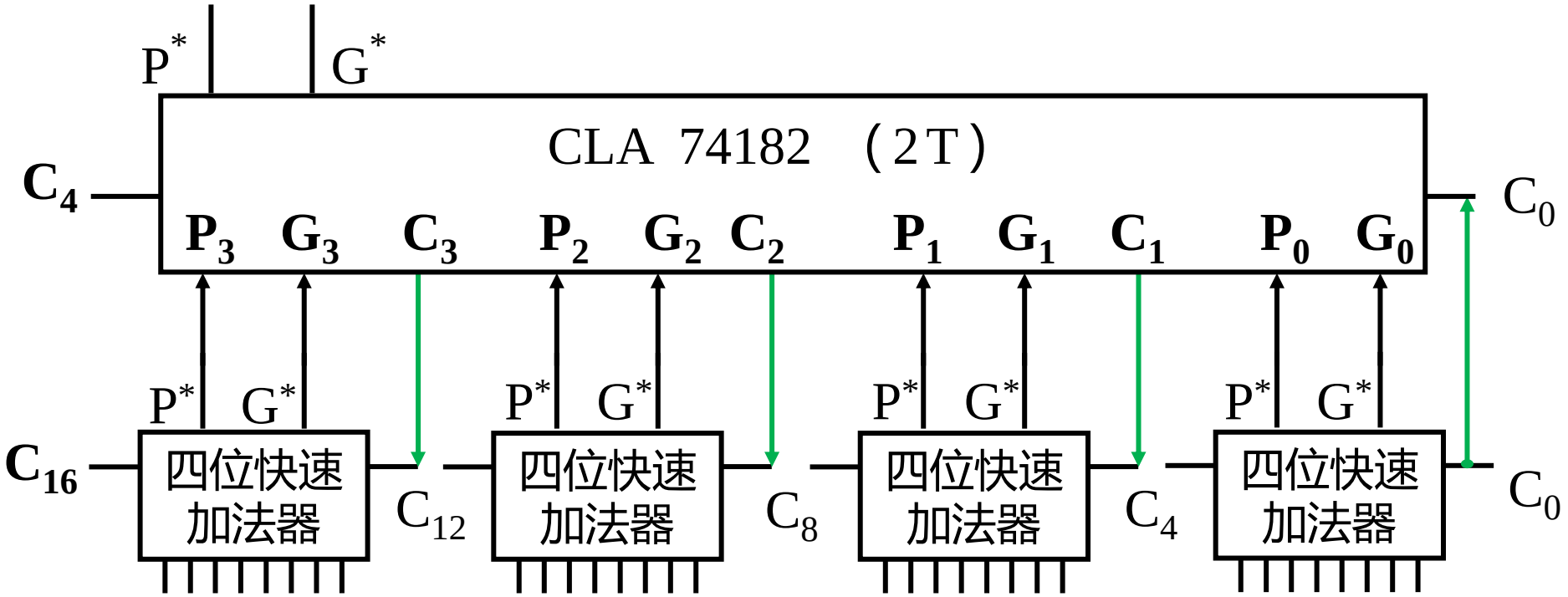
# 16位快速加法器

- 16位快速加法器
  - 组内先行进位，组间先行进位
    - 4个4位快速加法器 + 先行进位芯片 CLA74182



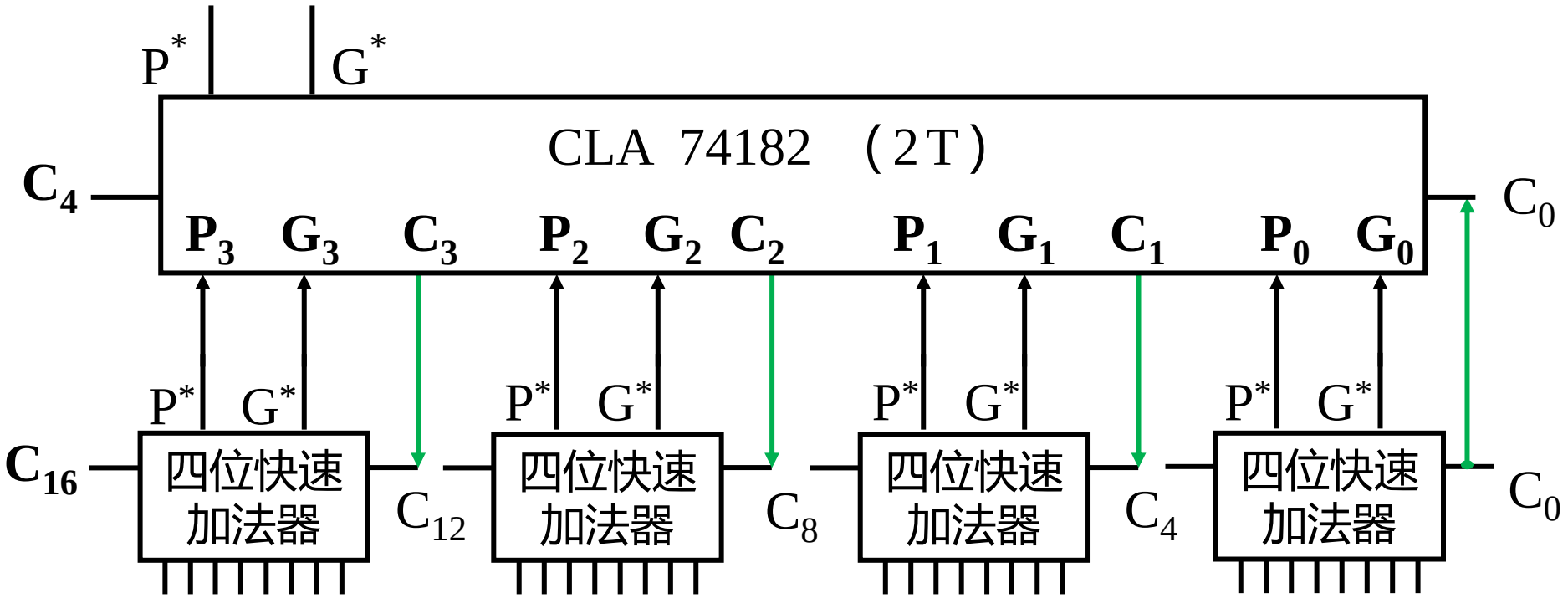
# 16位快速加法器

- 时间延迟分析（设与/或门的传播时延是T，异或门的传播时延是3T）
  - 经过1T：产生组内各$G$；经过3T：产生组内各$P$
    - 经过6T：产生$S_0$
  - 经过4T：产生各组$P^*$。经过5T：产生各组$G^*$，第一个4位快速加法器的各进位$C_4 - C_1$



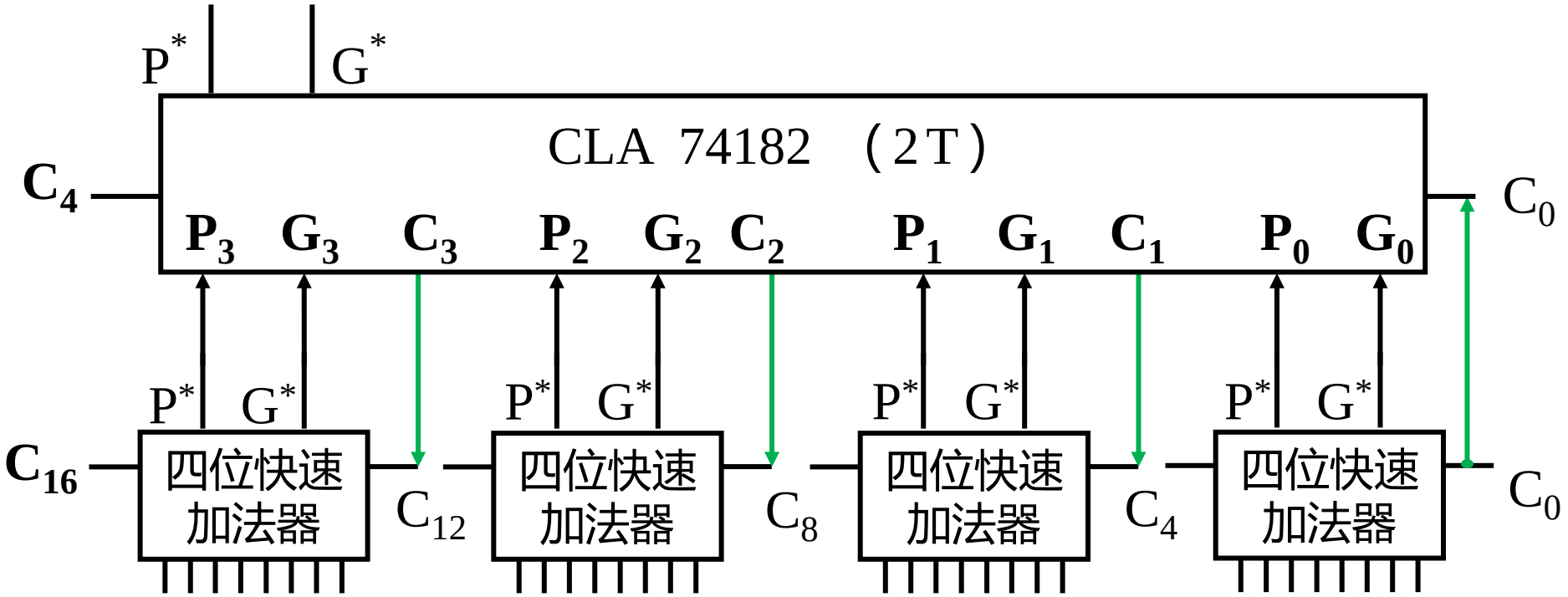
# 16位快速加法器

- 时间延迟分析（设与/或门的传播时延是T，异或门的传播时延是3T）
  - 经过4T：产生各$P^*$。经过5T：产生各$G^*$，第一个4位快速加法器的各进位$C_4 - C_1$
    - 经过6T：产生图中CLA 74182的进位输出$C_1$，即$C_4$（因为 $C_4 = G_0^* + P_0^* C_0$）
    - 经过8T：产生$S_3 - S_1$



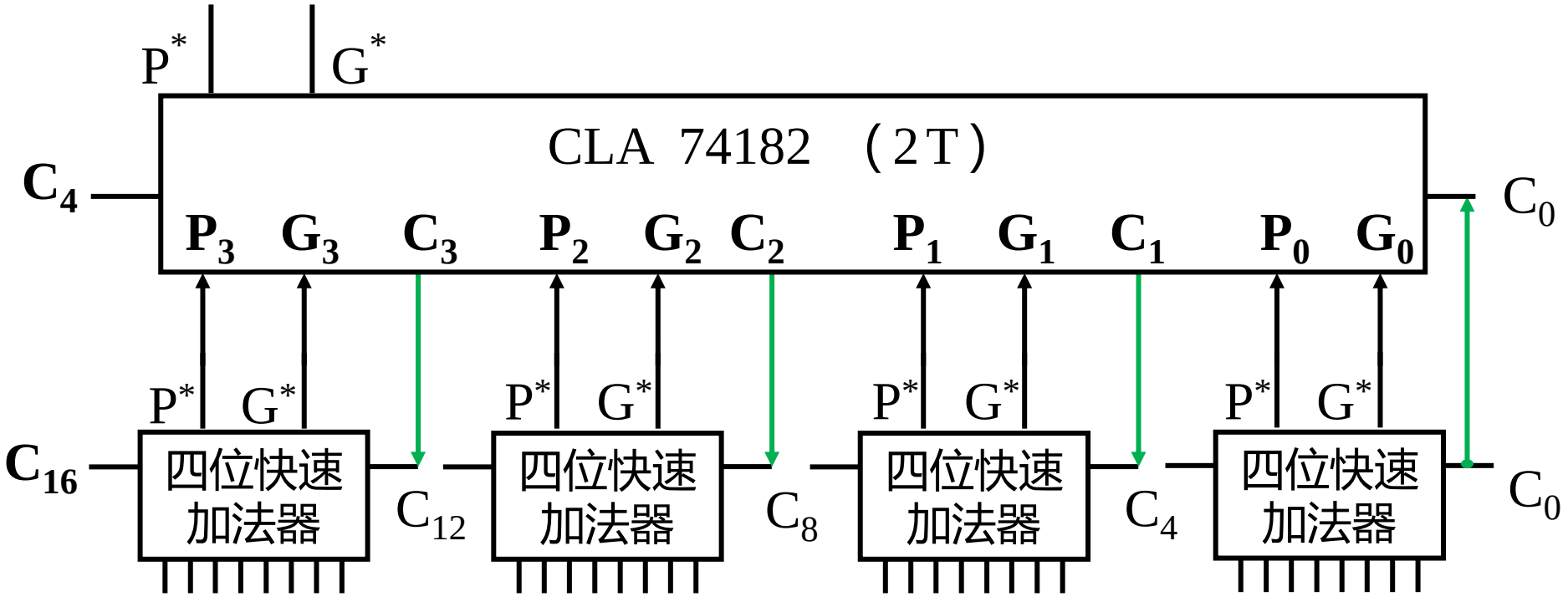
# 16位快速加法器

- 时间延迟分析（设与/或门的传播时延是T，异或门的传播时延是3T）
  - 经过4T：产生各$P^*$。经过5T：产生各$G^*$，第一个4位快速加法器的各进位$C_4 - C_1$
  - 经过7T：产生$C_{16}$，$C_{12}$，$C_8$（即图中CLA 74182的进位输出$C_4 - C_2$）



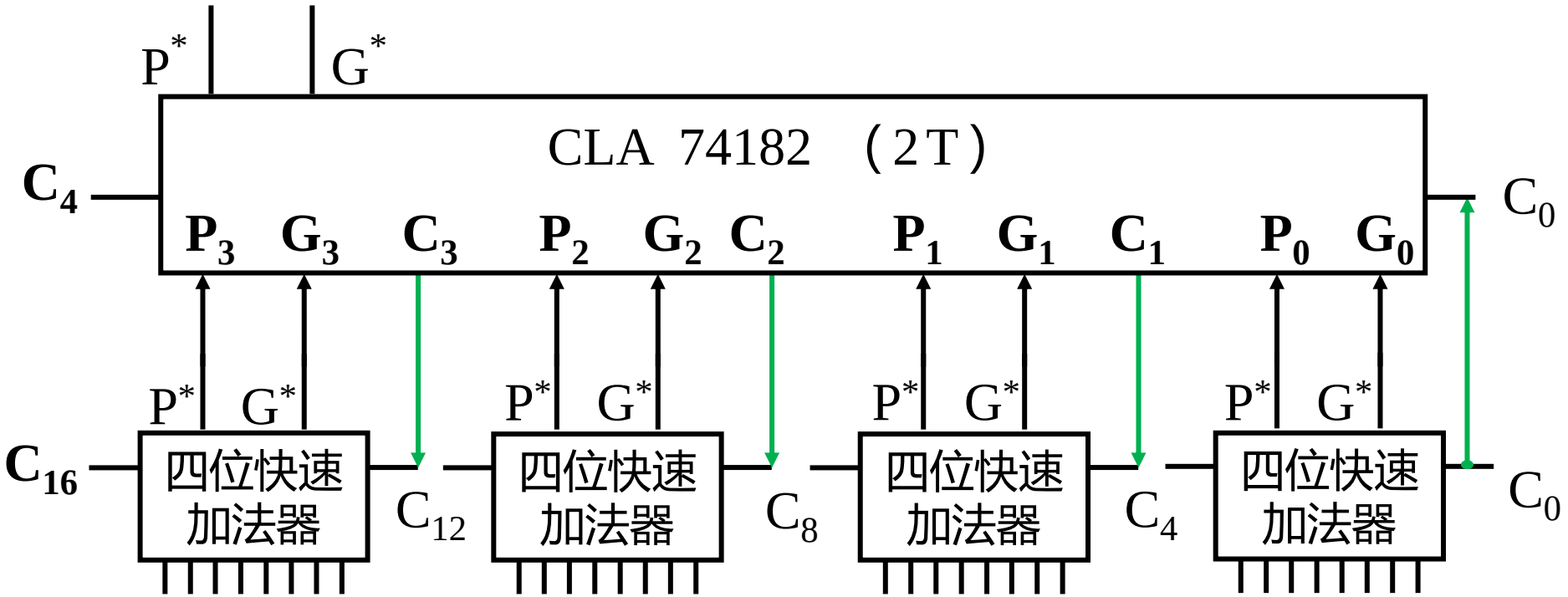
# 16位快速加法器

- 时间延迟分析（设与/或门的传播时延是T，异或门的传播时延是3T）
  - 经过7T：产生$C_{16}$，$C_{12}$，$C_8$，$C_4$（即图中CLA 74182的进位输出$C_4 - C_1$）
    - 经过10T：产生$S_{12}$，$S_8$，$S_4$
  - 经过9T：产生$C_{16} - C_{13}$，$C_{12} - C_9$，$C_8 - C_5$（即图中各四位快速加法器的进位输出）



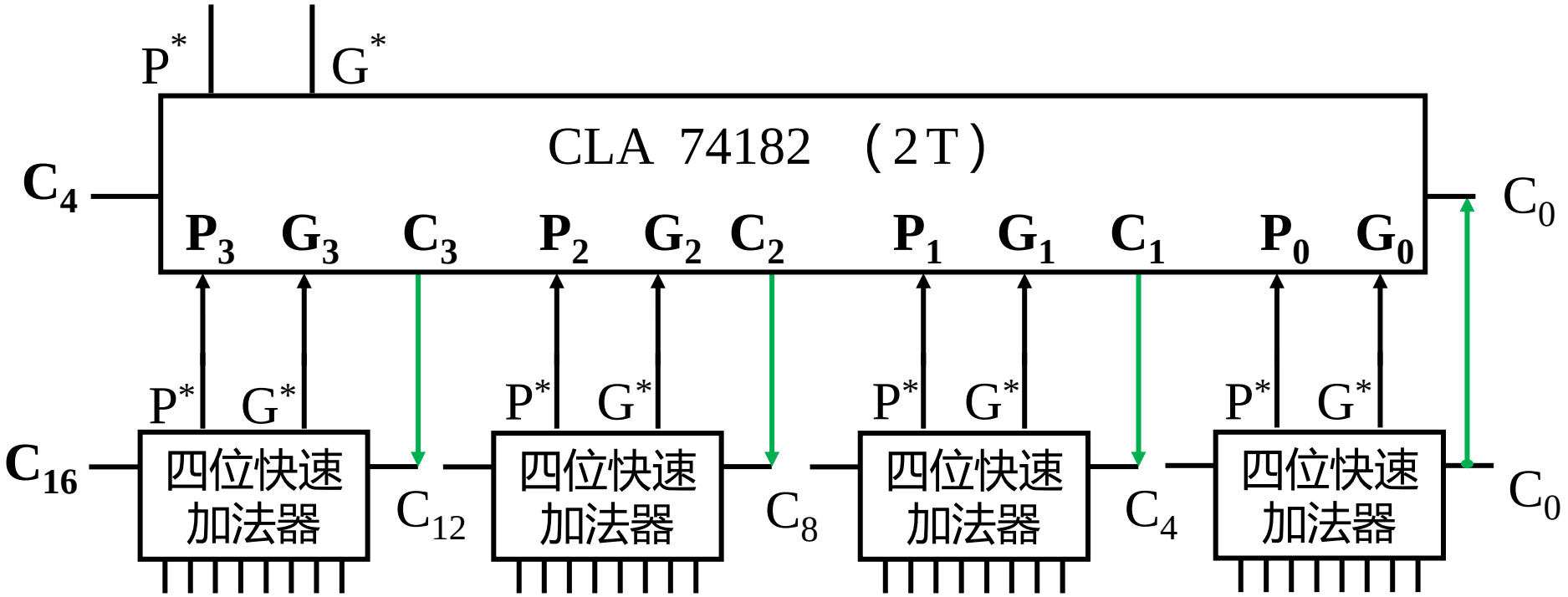
# 16位快速加法器

- 时间延迟分析（设与/或门的传播时延是T，异或门的传播时延是3T）
  - 经过9T：产生$C_{16}-C_{13}$，$C_{12}-C_9$，$C_8-C_5$（即图中各四位快速加法器的进位输出）
    - 经过12T：产生剩余全部和数$S_{15}-S_{13}$，$S_{11}-S_9$，$S_7-S_5$



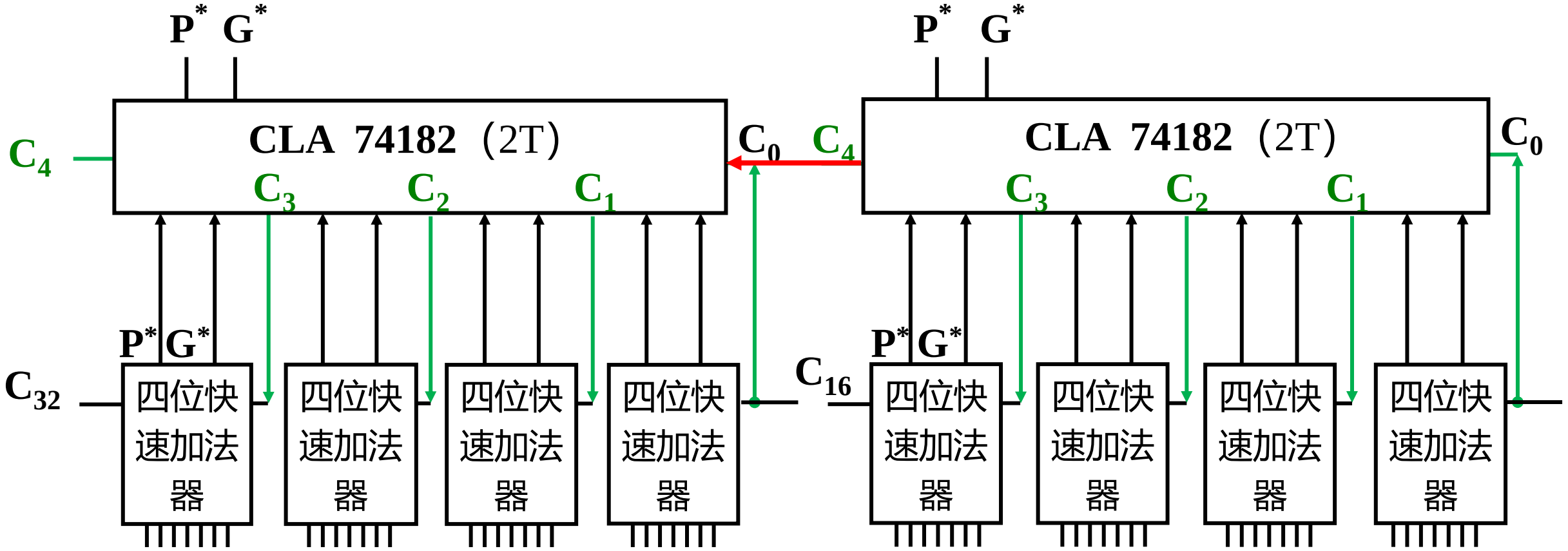
# 16位快速加法器

- 时间延迟分析（设与/或门的传播时延是T，异或门的传播时延是3T）
  - 产生全部和数所需时间： 12T
    - 组内先行进位、组间串行进位的16位加法器产生全部和数所需时间：14T
    - 相比之下，产生全部和数所需时间少了2T



# 32位快速加法器

- 组成
  - 由两个16位快速加法器串联得到




# 谢谢！