

## 本节主题



# 加法器的优化

北京大学·慕课  
计算机组成  
制作人：陆俊林

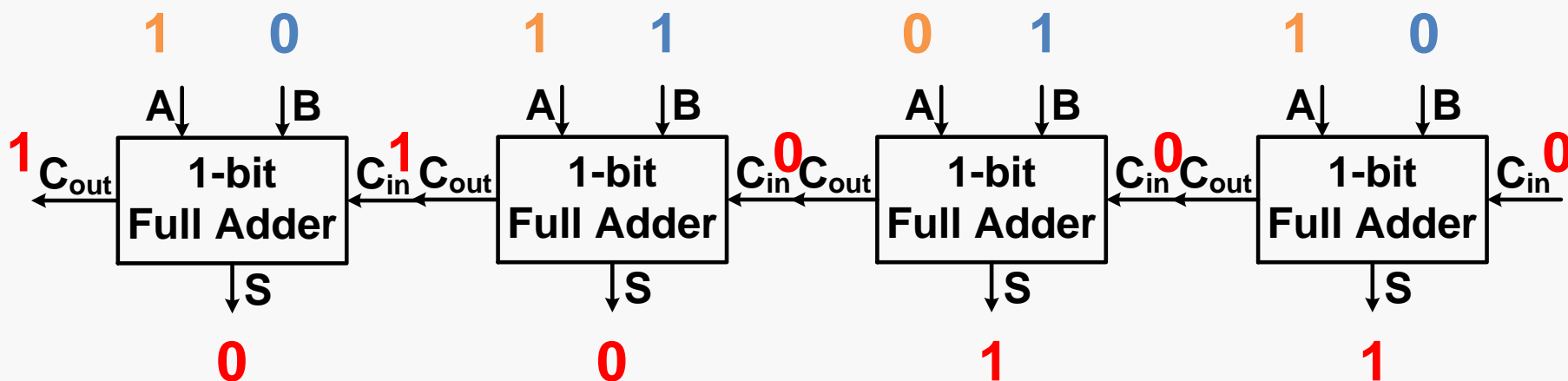


# 4-bit加法器示例

$$\begin{array}{r} \text{被加数 A} \\ 1101 \\ + \text{加数 B} \\ 0110 \\ \hline \text{进位 C} \quad 1 \quad 0011 \\ \text{和 S} \end{array}$$

两个4-bit二进制数相加

由四个全加器构成的4-bit加法器



# 行波进位加法器 ( Ripple-Carry Adder , RCA )



## 结构特点

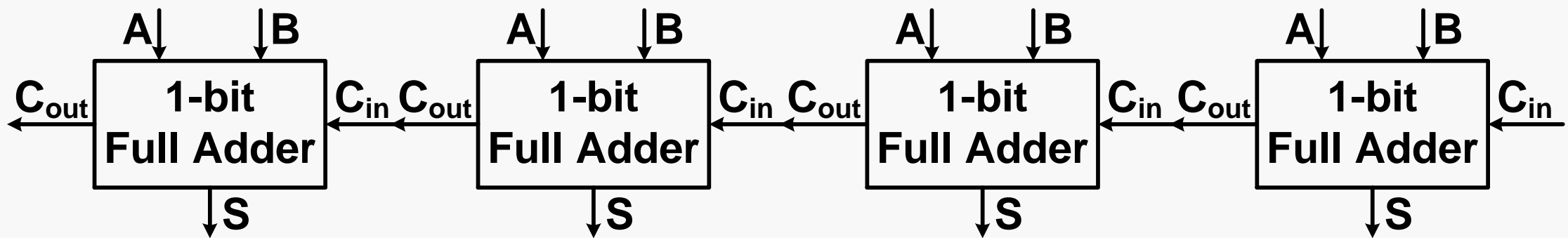
- 低位全加器的 $C_{out}$ 连接到高一位全加器 $C_{in}$

## 优点

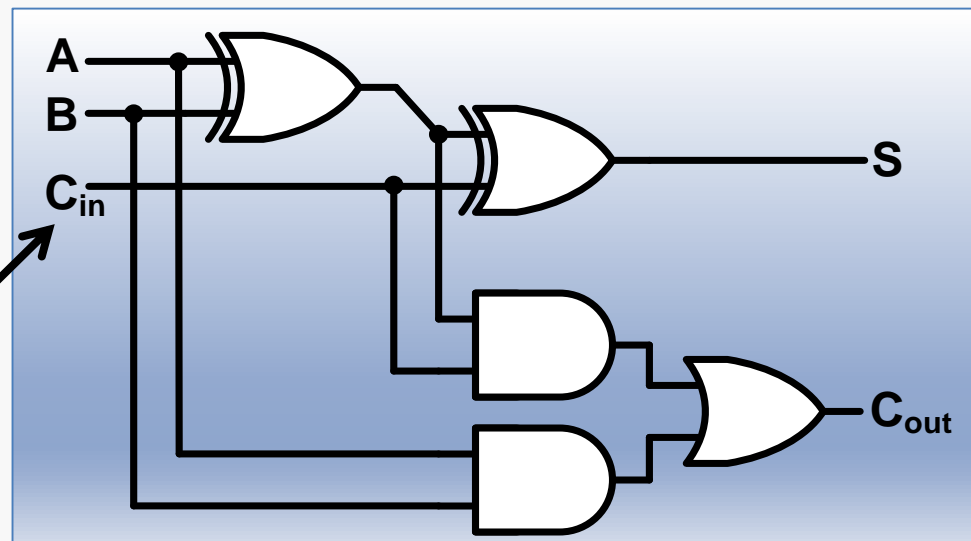
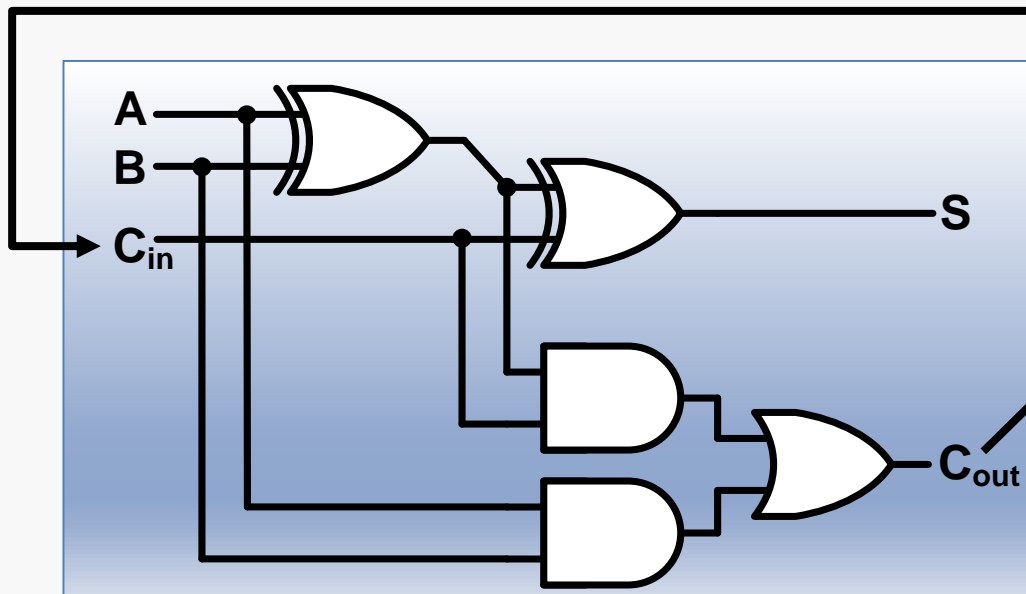
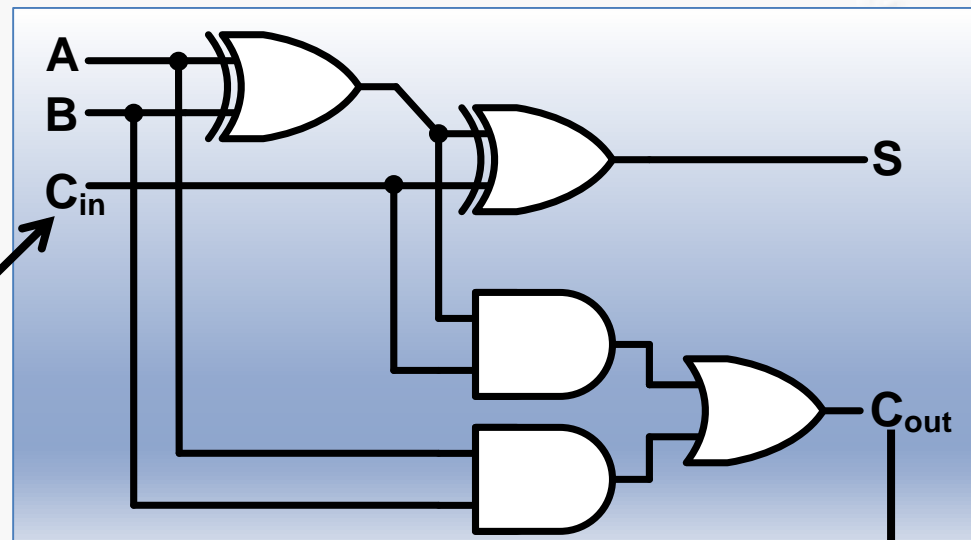
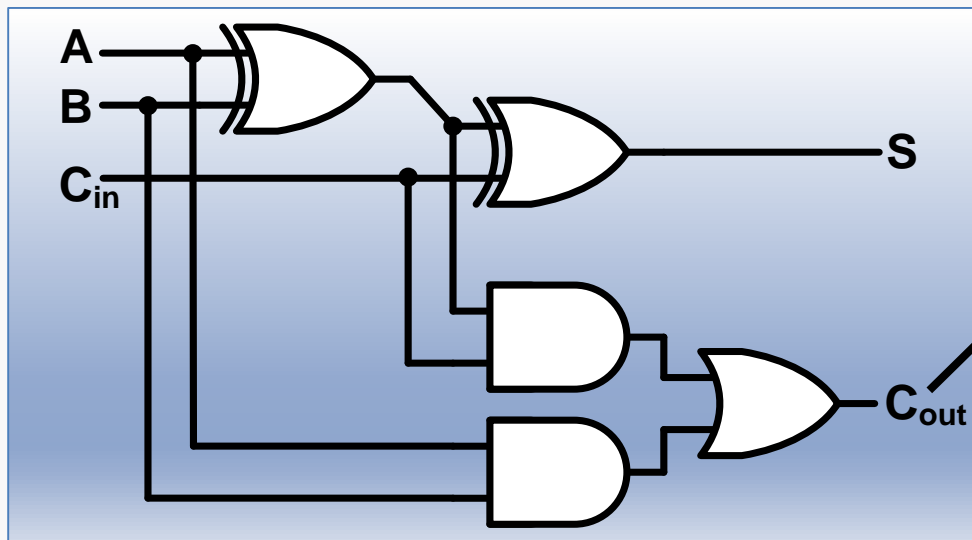
- 电路布局简单，设计方便

## 缺点

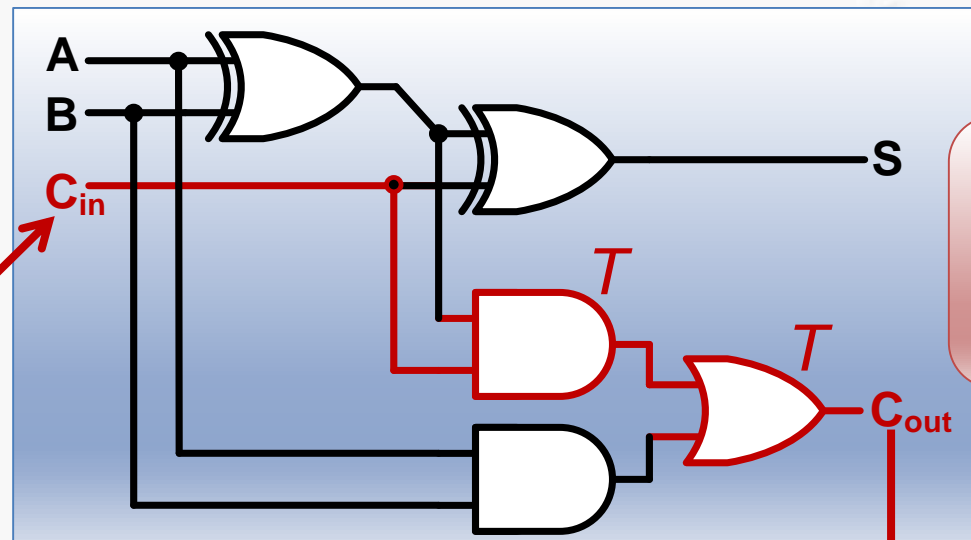
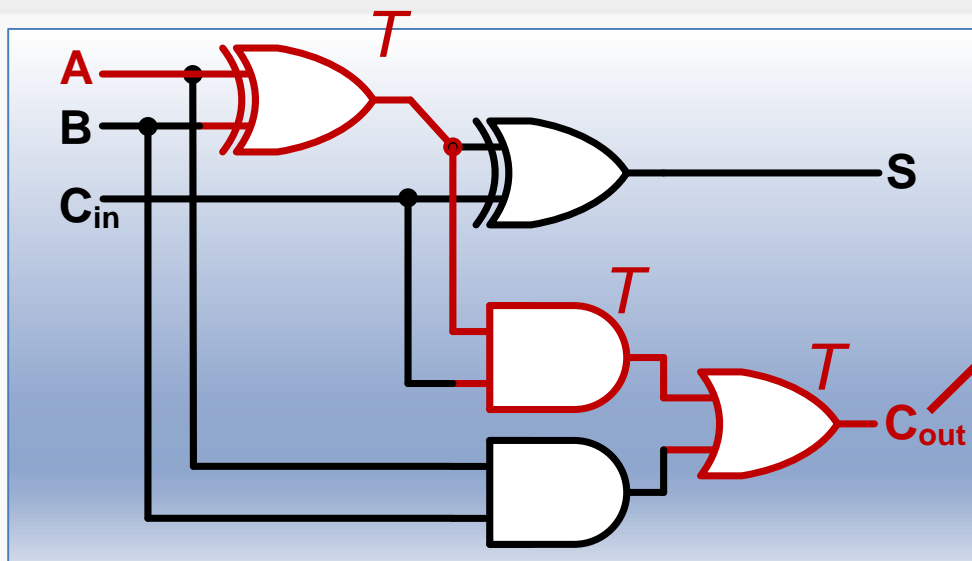
- 高位的运算必须等待低位的运算完成，延迟时间长



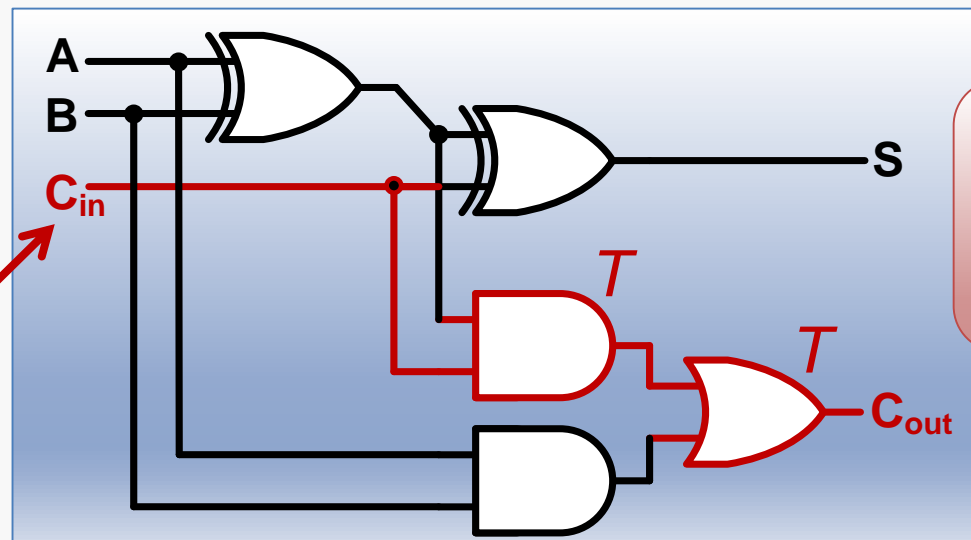
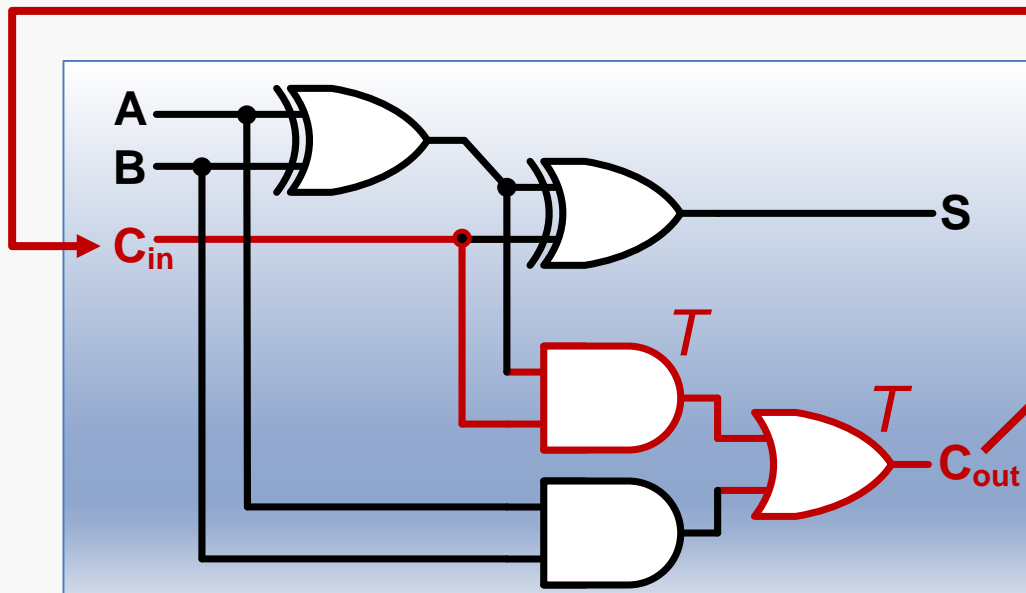
# 4-bit RCA的门电路实现



# 4-bit RCA的关键路径（延迟最长的路径）



总延迟时间  
 $(T+T) \times n + T$   
 $= (2n+1)T$



总延迟时间  
 $(T+T) \times 4 + T$   
 $= 9T$

## 32-bit RCA的性能分析

## 总延迟时间：

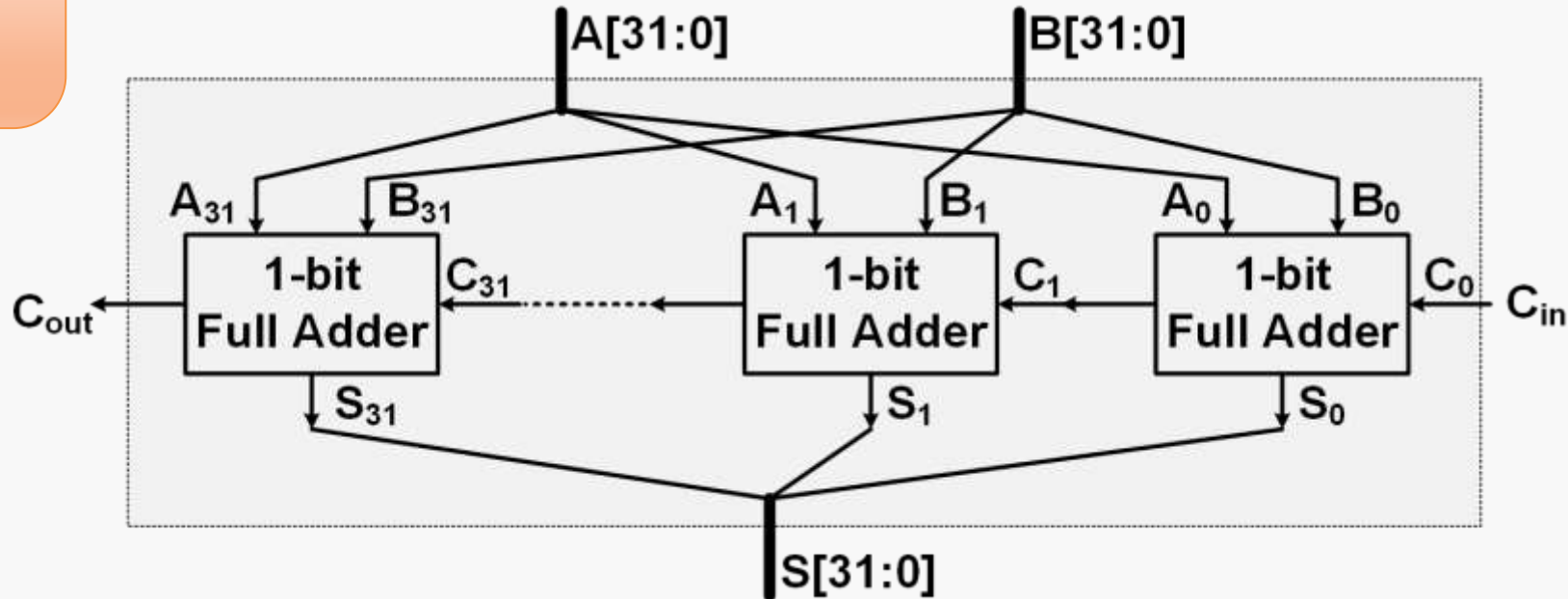
$$(2n+1)T = (2 \times 32 + 1) \times T = 65T$$

## 参考値

水果智能手机5s的A7 SoC  
采用28nm制造工艺  
主频1.3GHz ( 0.66ns )



	延迟时间	时钟频率
4-bit RCA	0.18ns	5.56GHz
32-bit RCA	1.3ns	769MHz
注：参照28nm制造工艺，门延迟 $T$ 设为0.02ns		



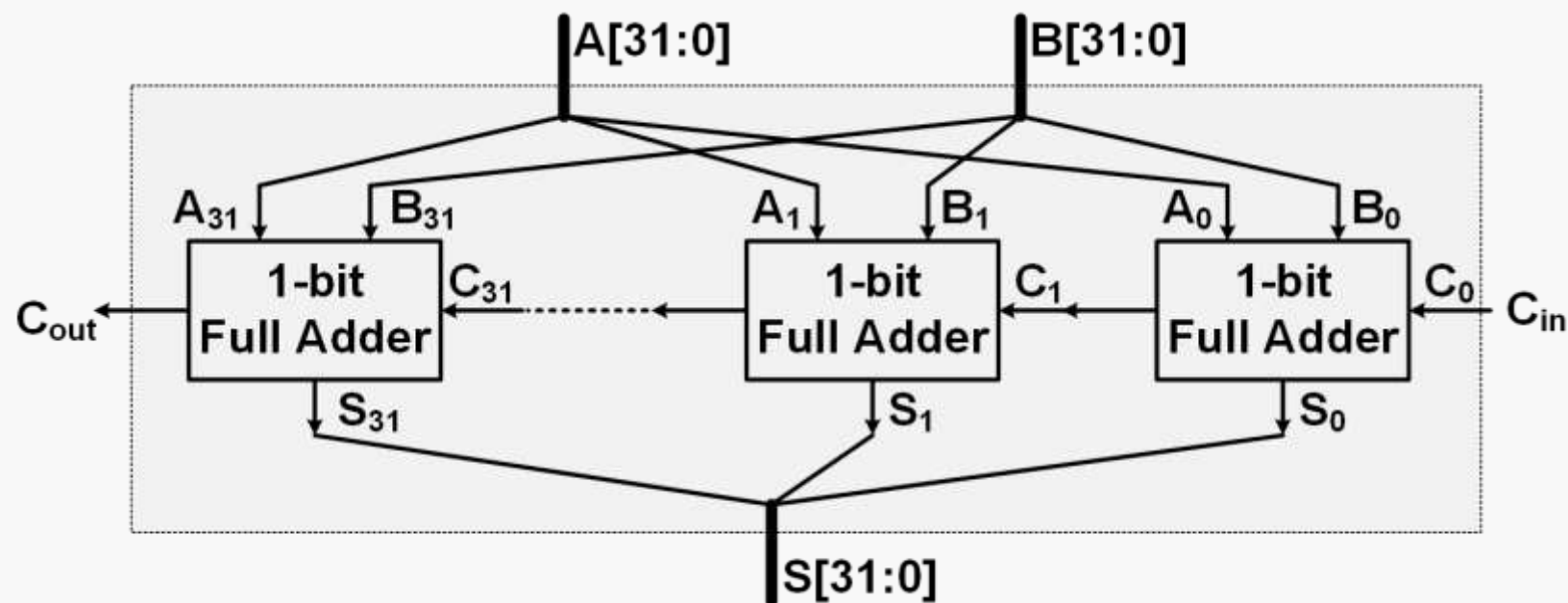
# 加法器的优化思路

## ❶ 主要问题

- 。高位的运算必须等待低位的“进位输出信号”

## ❷ 优化思路

- 。能否提前计算出“进位输出信号”？



# 进位输出信号的分析

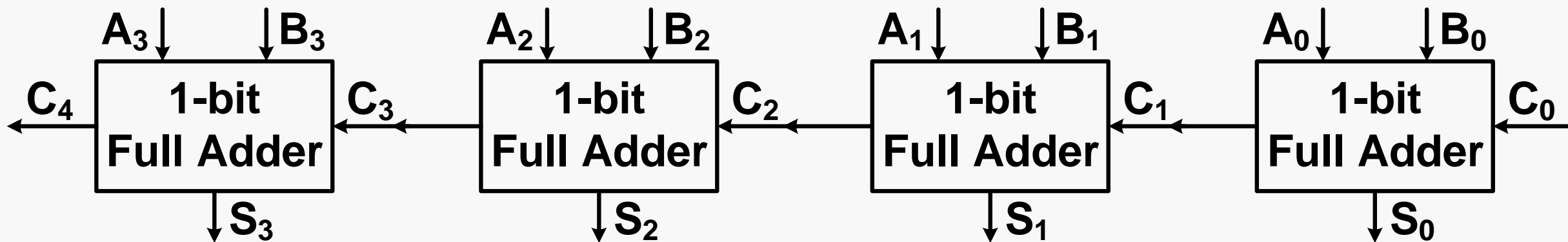


$$\begin{aligned}C_{i+1} &= (A_i \cdot B_i) + (A_i \cdot C_i) + (B_i \cdot C_i) \\&= (A_i \cdot B_i) + (A_i + B_i) \cdot C_i\end{aligned}$$

设：

- 生成 (Generate) 信号： $G_i = A_i \cdot B_i$
- 传播 (Propagate) 信号： $P_i = A_i + B_i$

则： $C_{i+1} = G_i + P_i \cdot C_i$





# 如何提前计算“进位输出信号”



$$\textcircled{a} C_1 = G_0 + P_0 \cdot C_0$$

$$\textcircled{a} C_2 = G_1 + P_1 \cdot C_1$$

$$= G_1 + P_1 \cdot (G_0 + P_0 \cdot C_0)$$

$$= \mathbf{G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0}$$

$$\textcircled{a} C_3 = G_2 + P_2 \cdot C_2$$

$$= G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0)$$

$$= \mathbf{G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0}$$

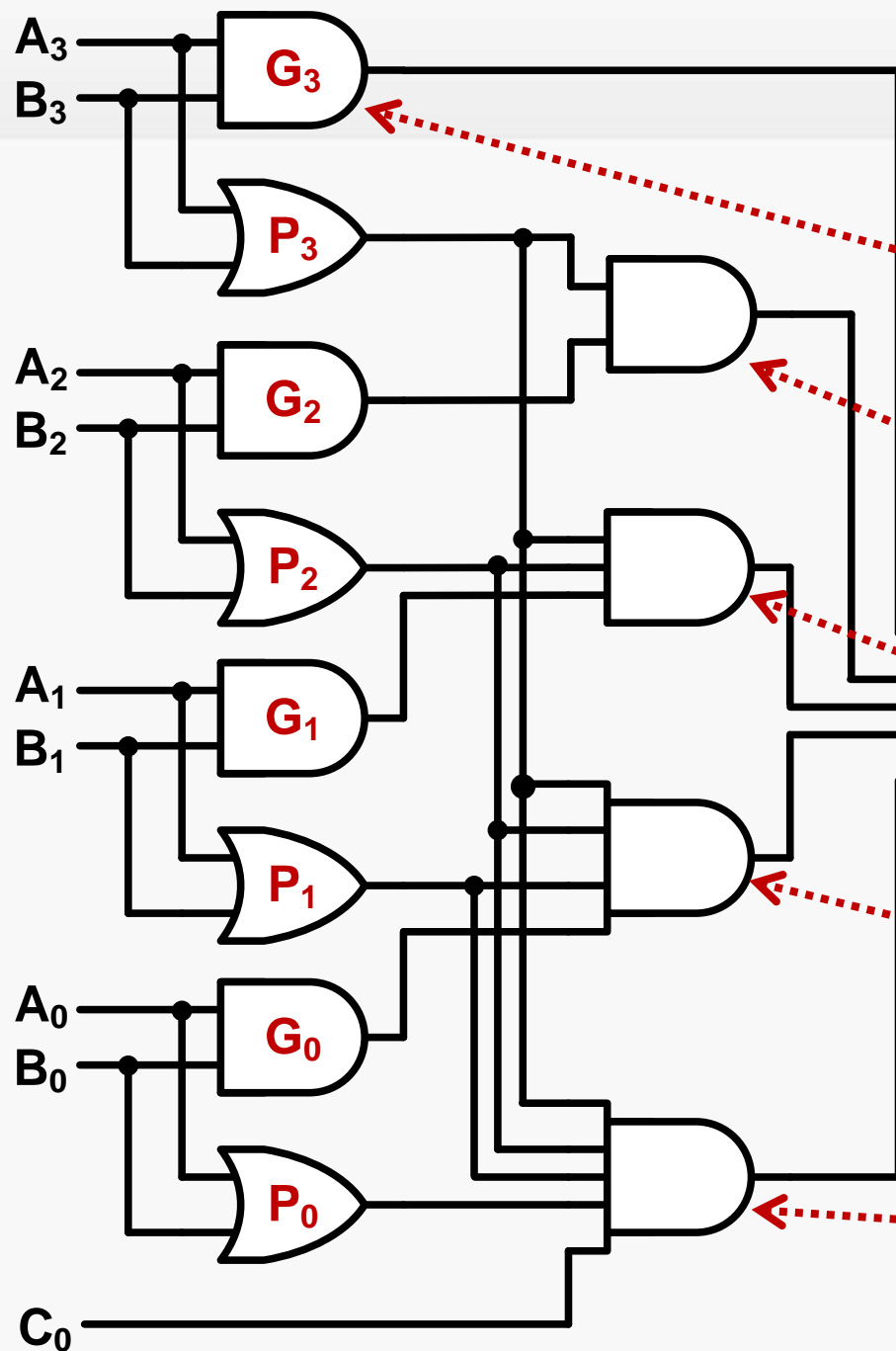
$$\textcircled{a} C_4 = G_3 + P_3 \cdot C_3$$

$$= G_3 + P_3 \cdot (G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0)$$

$$= \mathbf{G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_0}$$

$$\mathbf{C_{i+1} = G_i + P_i \cdot C_i}$$

## 提前计算 $C_4$ 的电路实现



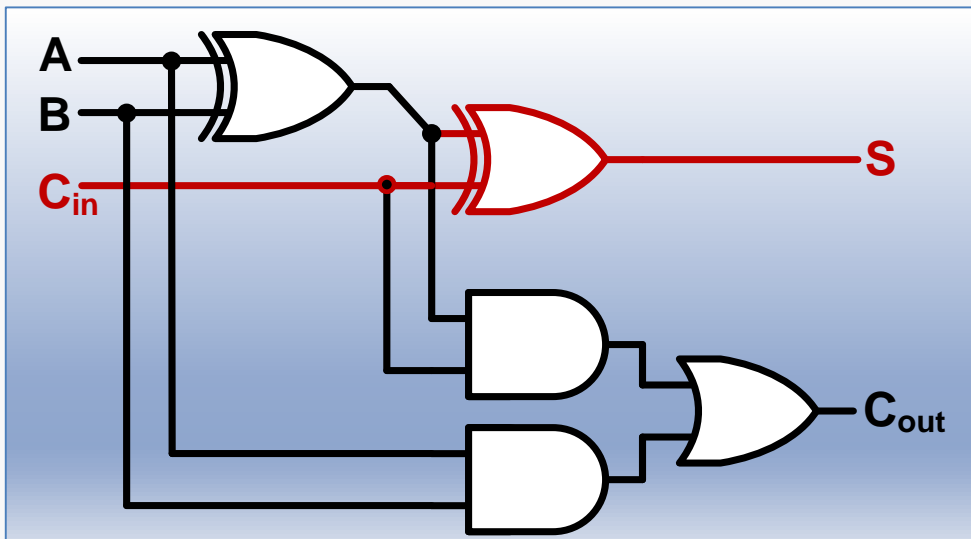
$C_4 =$

$$\begin{aligned} C_4 = & G_3 \\ & + P_3 \cdot G_2 \\ & + P_3 \cdot P_2 \cdot G_1 \\ & + P_3 \cdot P_2 \cdot P_1 \cdot G_0 \\ & + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_0 \end{aligned}$$

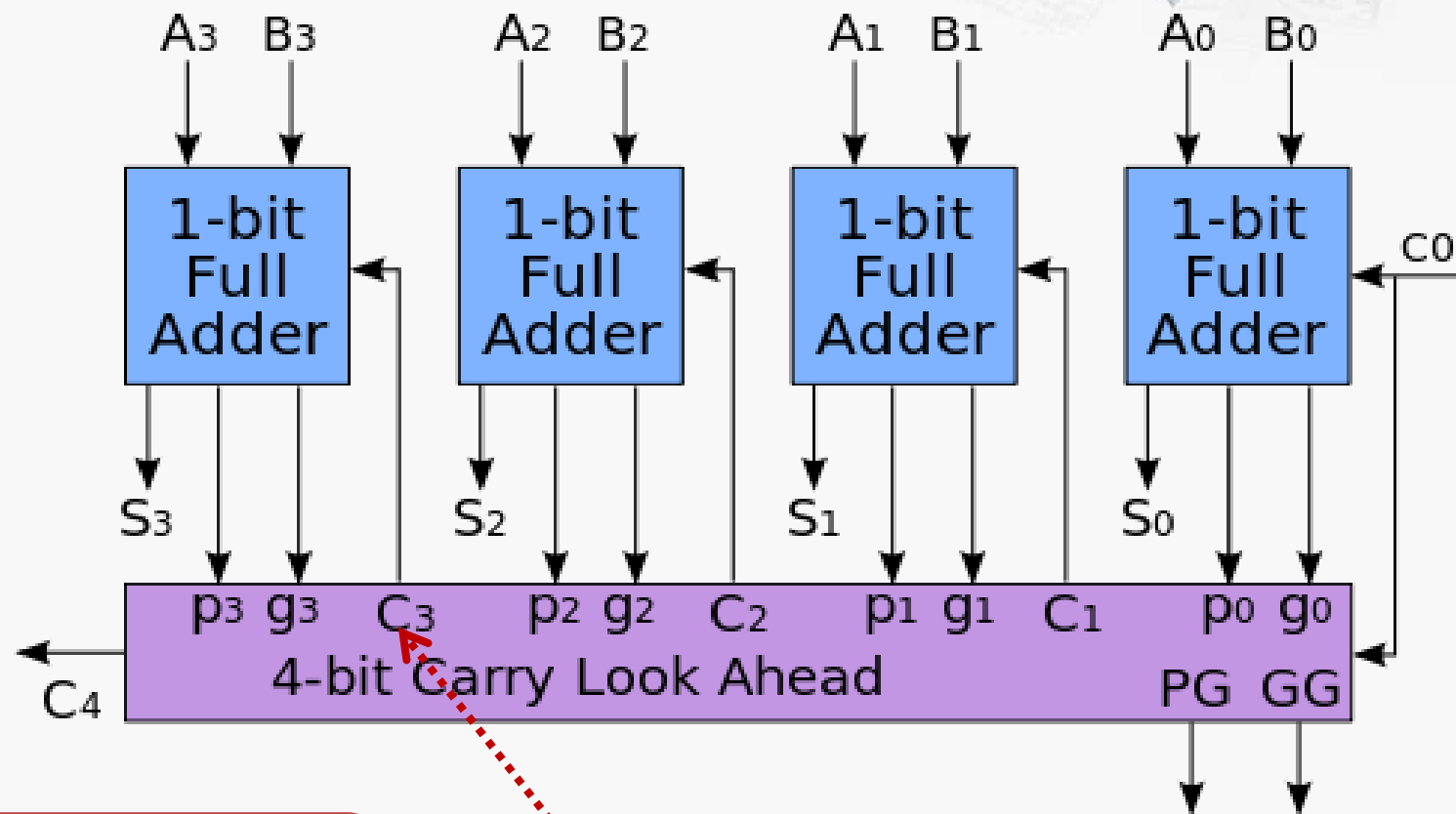
优点：计算 $C_{i+1}$ 的延迟时间固定为三级门延迟，与加法器的位数无关

缺点：如果进一步拓宽加法器的位数，则电路变得非常复杂

# 超前进位加法器 ( Carry-Lookahead Adder , CLA )



最后一级全加器  
还需要1级门延迟



总延迟时间  
为4级门延迟

计算 $C_3$ 需要3级门延迟

参考值：4-bit行波进  
位加法器的总延迟时  
间为9级门延迟

# 32-bit加法器的实现

- 如果采用行波进位
  - 总延迟时间为65级门延迟
- 如果采用完全的超前进位
  - 理想的总延迟时间为4级门延迟
  - 实际上电路过于复杂，难以实现
- 通常的实现方法
  - 采用多个小规模超前进位加法器拼接而成
  - 例如，用4个8-bit的超前进位加法器连接成32-bit加法器

	延迟时间	时钟频率
32-bit RCA	1.3ns	769MHz
单个CLA	0.08ns	/
4级CLA	0.26ns	3.84GHz

注：参照28nm制造工艺，门延迟设为0.02ns

$$C_{31} = G_{30} + P_{30} \cdot G_{29} + P_{30} \cdot P_{29} \cdot G_{28} + \dots + P_{30} \cdot P_{29} \cdot P_{28} \cdot \dots \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_0$$

需要32输入的与门和或门？！

## 本节小结



# 加法器的优化

北京大学·慕课  
计算机组成  
制作人：陆俊林

