中国科学技术大学
University of Science and Technology of China

数字集成电路设计

第十一章　数据通路子系统

白雪飞

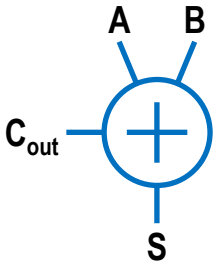
中国科学技术大学微电子学院

提纲

- 加法和减法
- 基本运算
- 并行前缀运算
- 乘法

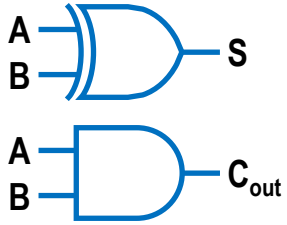
加法和减法

半加器



$S = A \oplus B, \quad C_{out} = A \cdot B$

半加器 (Half Adder, HA)

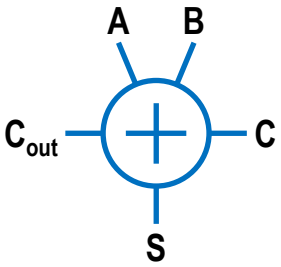


半加器电路

半加器真值表

A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

全加器



全加器 (Full Adder, FA)

$$G = A \cdot B$$

$$P = A \oplus B$$

$$K = \overline{A} \cdot \overline{B} = \overline{A + B}$$

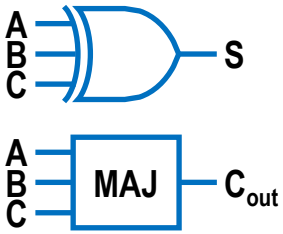
全加器真值表

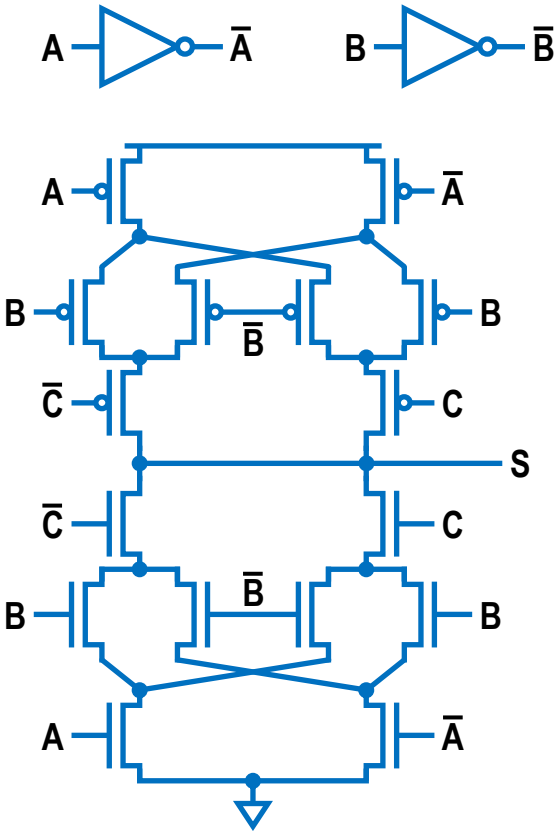
A	B	C	G	P	K	C_{out}	S
0	0	0	0	0	1	0	0
		1				0	1
0	1	0	0	1	0	0	1
		1				1	0
1	0	0	0	1	0	0	1
		1				1	0
1	1	0	1	0	0	1	0
		1				1	1

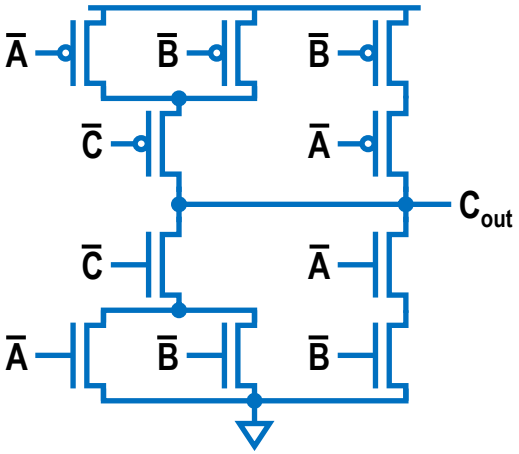
$$\begin{aligned} S &= A \oplus B \oplus C \\ &= P \oplus C \\ &= ABC + (A + B + C)\,\overline{C_{out}} \end{aligned}$$

$$\begin{aligned} C_{out} &= AB + AC + BC \\ &= AB + C(A + B) \\ &= \overline{\overline{A}\,\overline{B} + \overline{C}\left(\overline{A} + \overline{B}\right)} \\ &= \mathrm{MAJ}(A, B, C) \end{aligned}$$

全加器电路设计





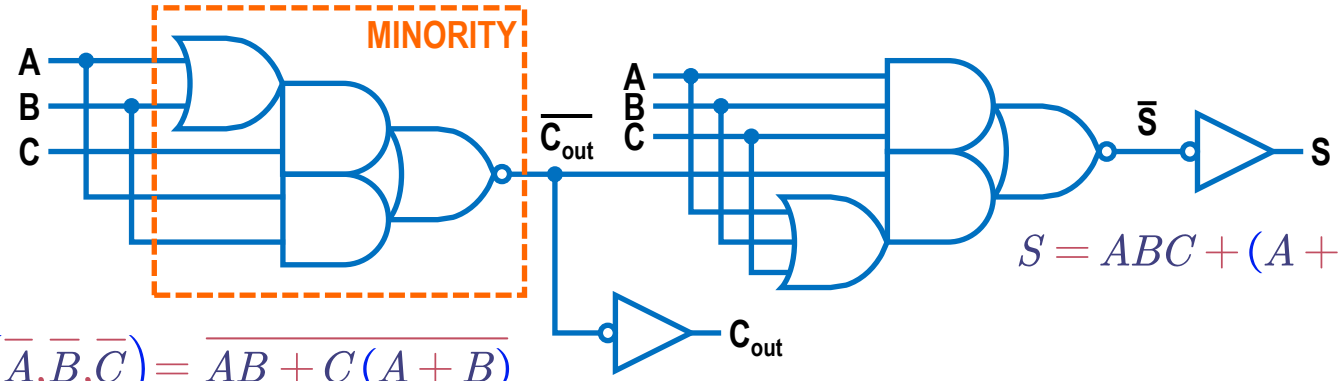


$$S = A \oplus B \oplus C$$
$$= A\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}\,\overline{B}C + ABC$$

$$C_{\text{out}} = \text{MAJ}(A,B,C)$$
$$= \overline{\overline{A}\,\overline{B} + \overline{C}\left(\overline{A} + \overline{B}\right)}$$

全加器电路设计

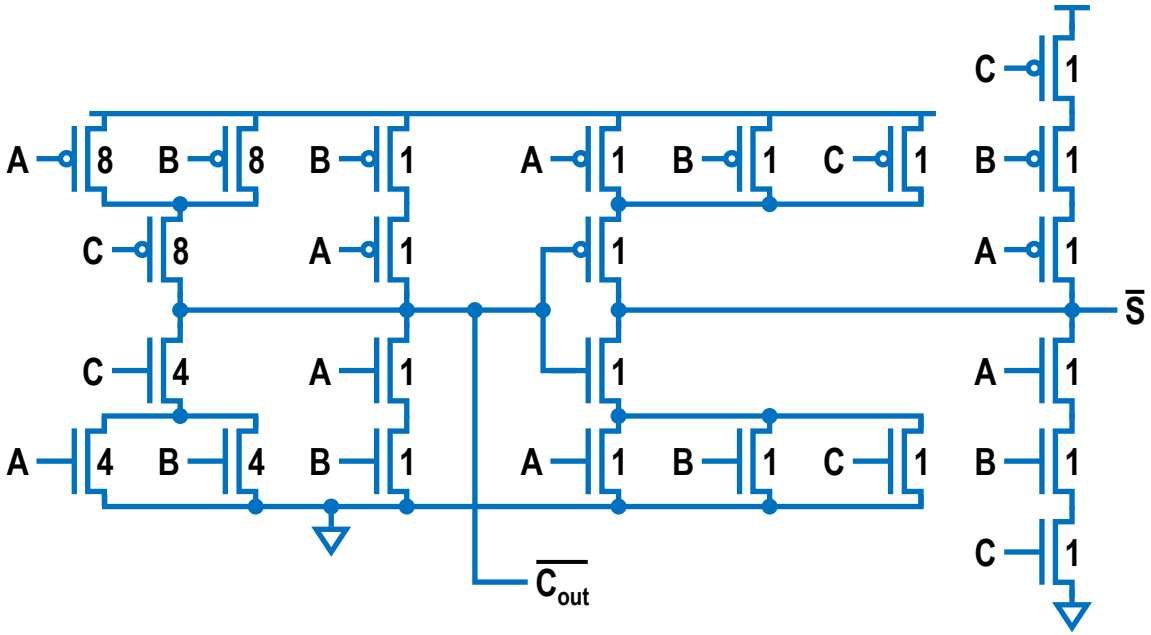
用于行波进位的全加器



$$\overline{C_{out}} = \mathrm{MAJ}\left(\overline{A},\overline{B},\overline{C}\right) = \overline{AB + C(A+B)}$$

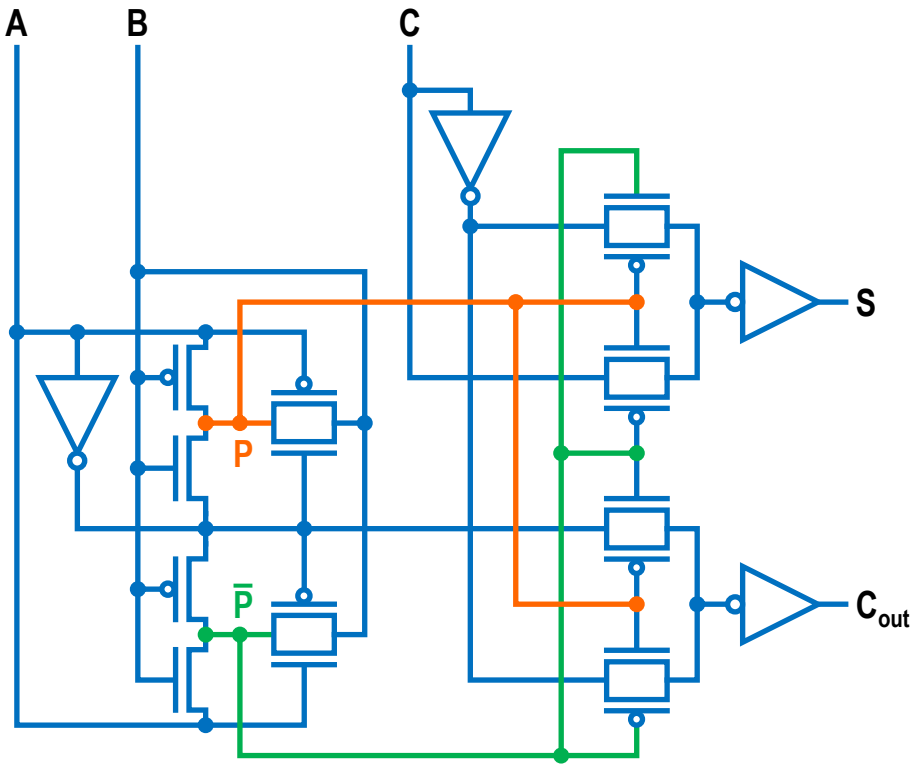
$$S = ABC + (A+B+C)\,\overline{C_{out}}$$

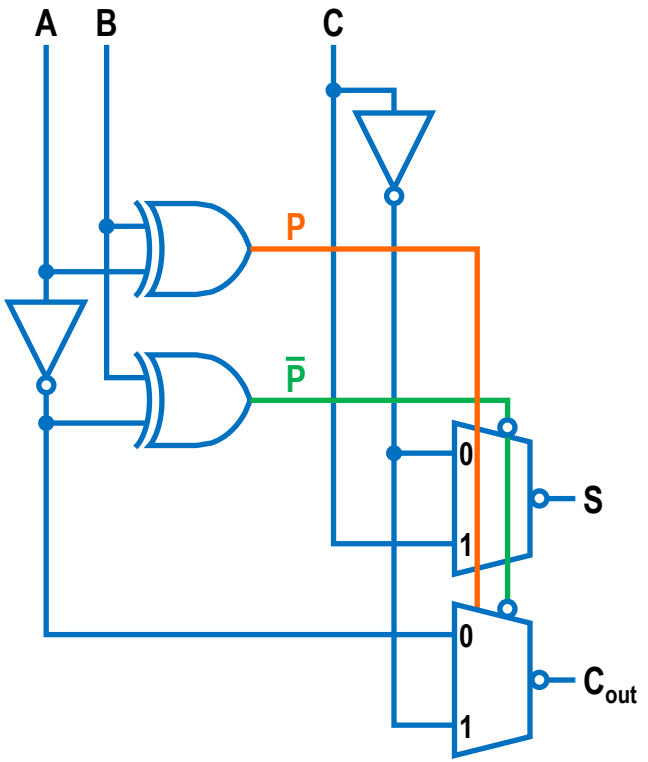
用于行波进位的全加器



镜像加法器 (Mirror Adder)

传输门全加器

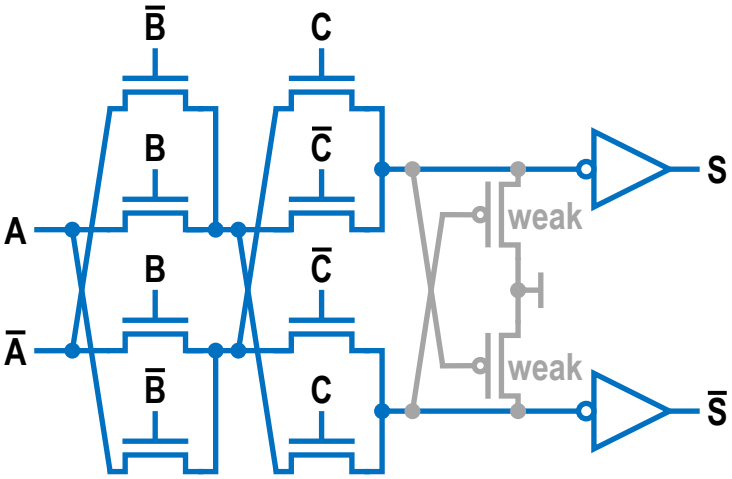




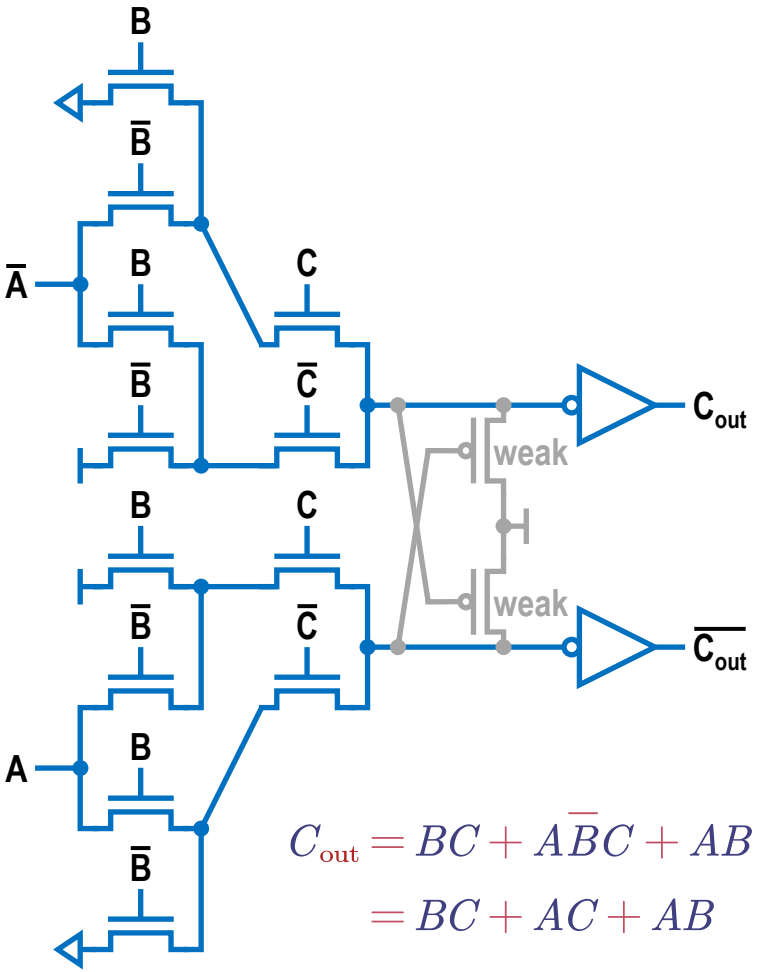
$$P = A \oplus B; \quad S = P \oplus C; \quad C_{\text{out}} = AB + PC = \overline{P}A + PC$$

传输门全加器

互补传输管逻辑全加器



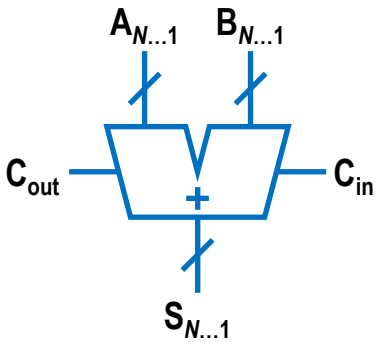
$$S = ABC + A\bar{B}\,\bar{C} + \bar{A}\,\bar{B}C + \bar{A}B\bar{C}$$

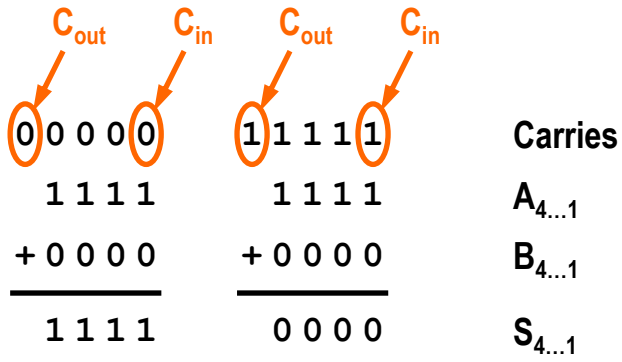


$$C_{\text{out}} = BC + A\bar{B}C + AB\bar{C}$$
$$= BC + AC + AB$$

互补传输管逻辑(CPL)全加器

进位传播加法器

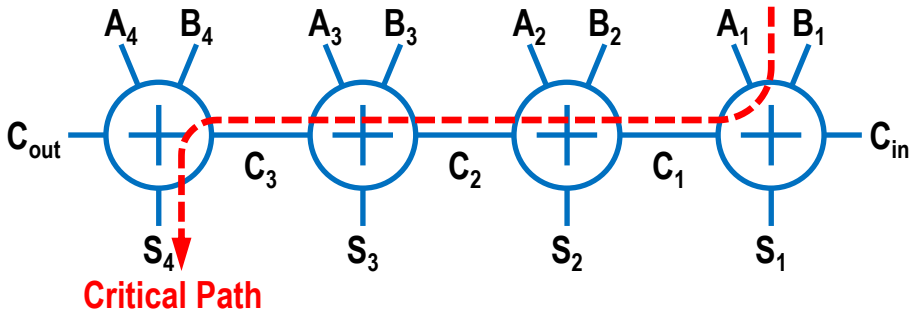


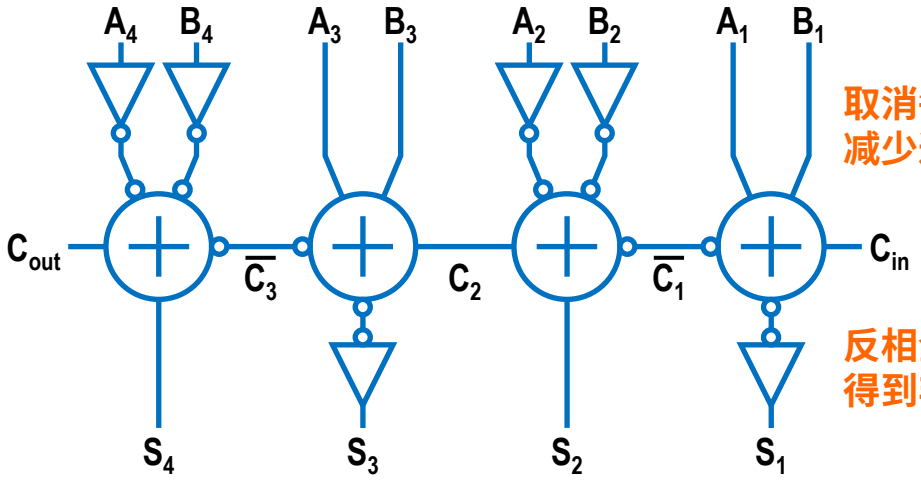


进位传播加法器 (Carry-Propagate Adder, CPA)

每一位的进位输入都可能影响所有后续高位的进位输入

行波进位加法器



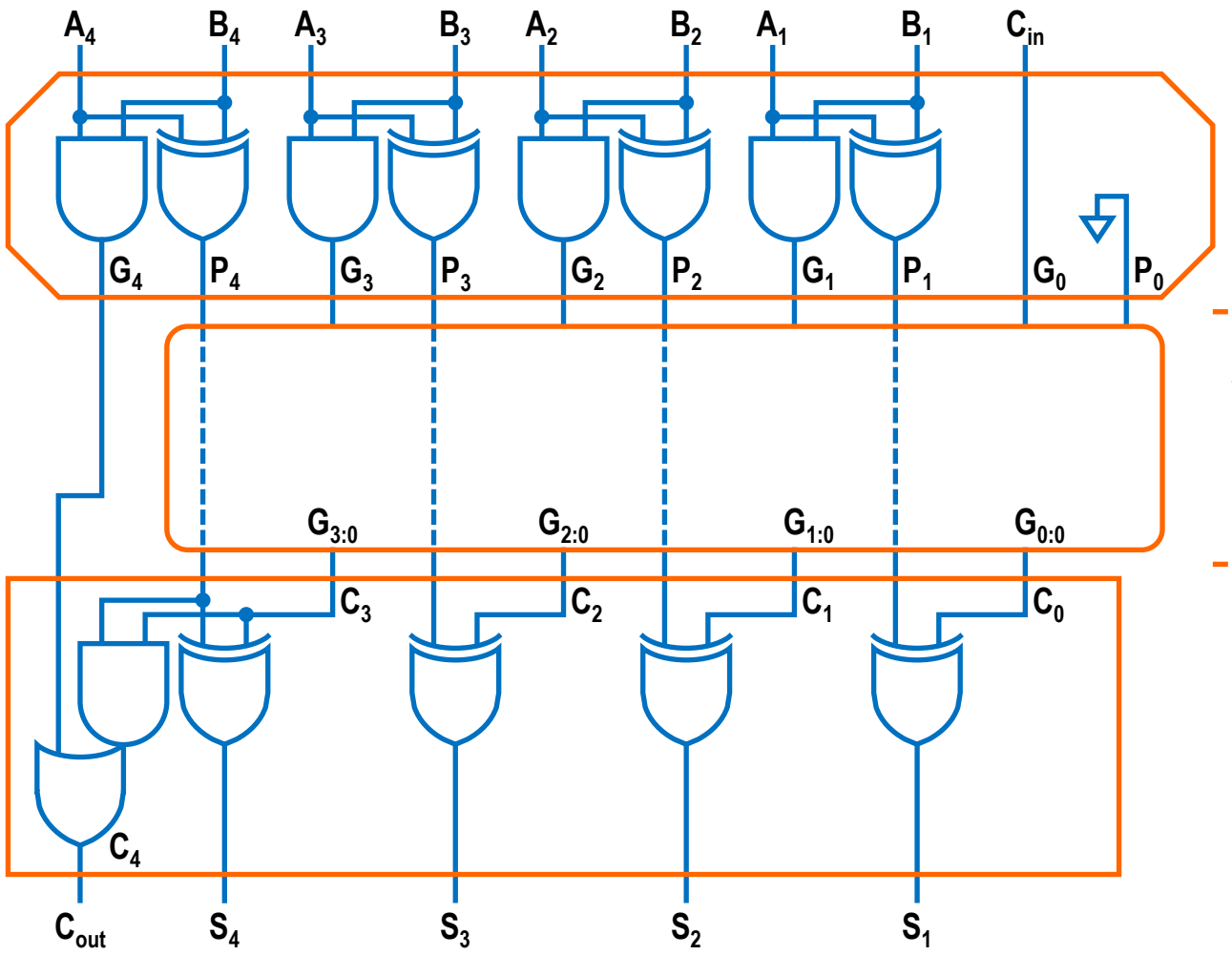


取消每级输出端反相器
减少进位传播路径延时

反相全加器输入反相时
得到非反相的和与进位

行波进位加法器 (Ripple-Carry Adder, RCA)

进位产生和传播



1. Bitwise PG Logic

$$G_{0:0} = C_{\text{in}}$$

$$P_{0:0} = 0$$

$$G_{i:i} \equiv G_i = A_i \cdot B_i$$

$$P_{i:i} \equiv P_i = A_i \oplus B_i$$

2. Group PG Logic

$$G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j}$$

$$P_{i:j} = P_{i:k} \cdot P_{k-1:j}$$

3. Sum Logic

$$C_0 = C_{\text{in}}$$

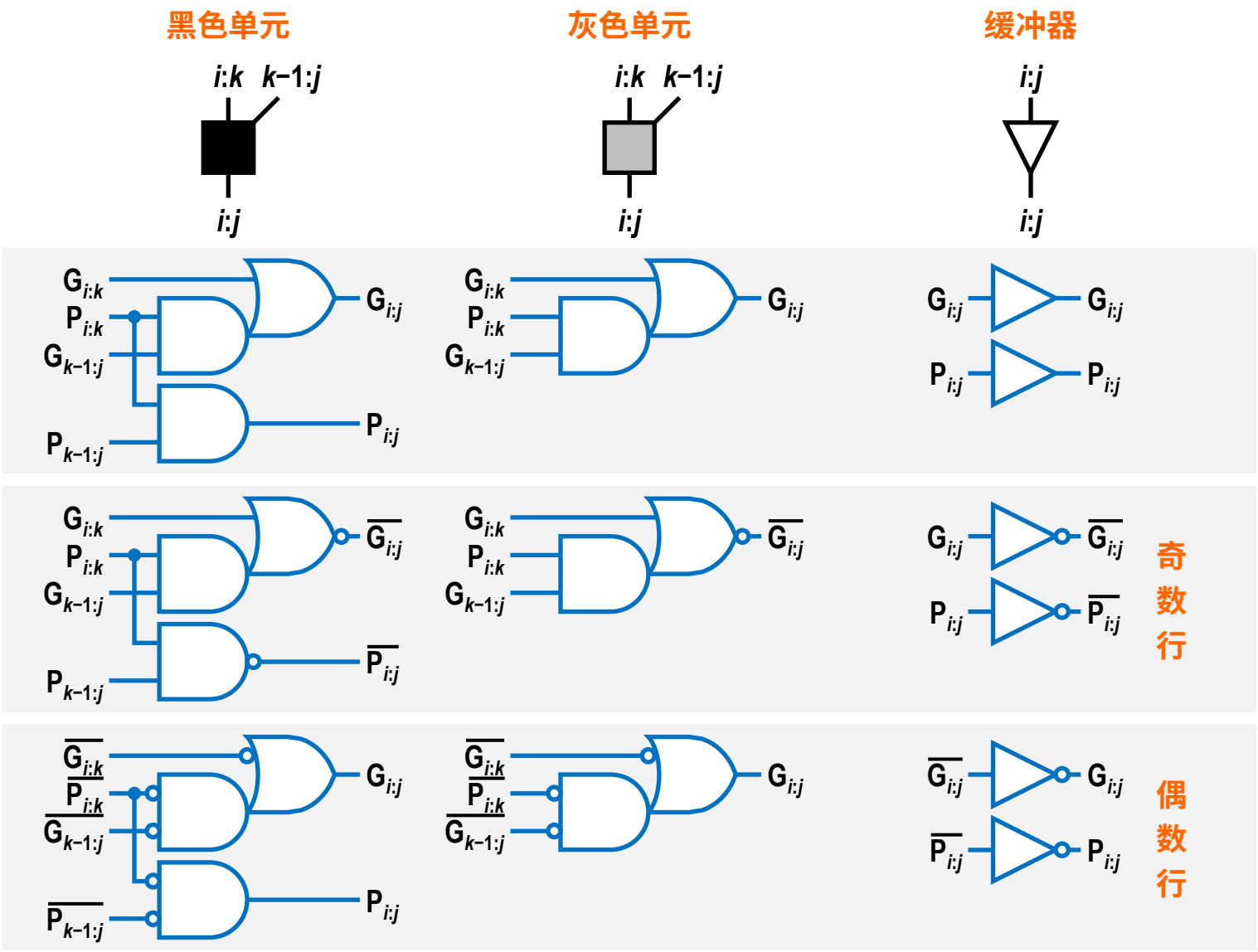
$$C_{\text{out}} = C_N$$

$$C_{i-1} = G_{i-1:0}$$

$$S_i = P_i \oplus G_{i-1:0}$$

带进位产生和传播逻辑的加法器

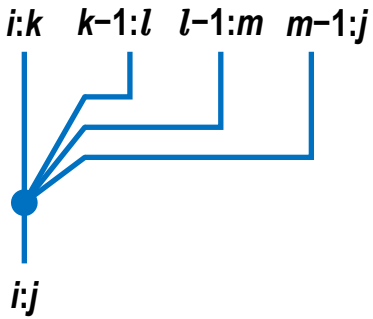
组PG单元

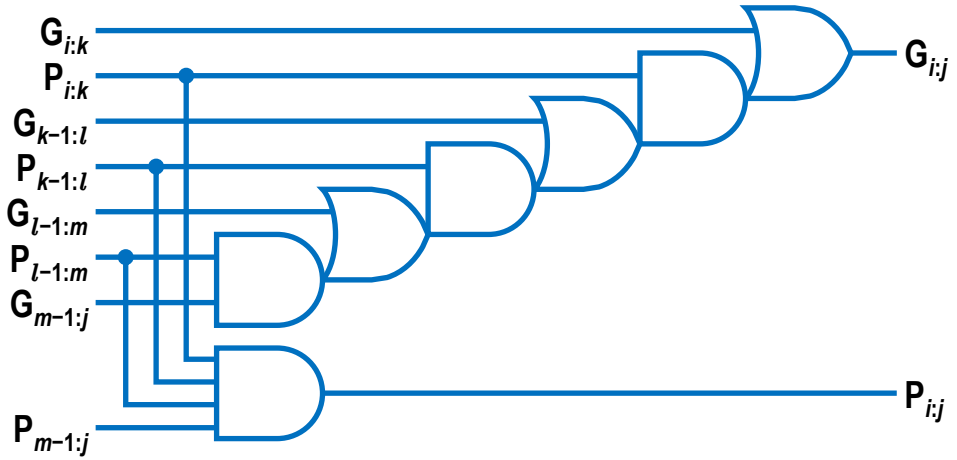


二阶组PG单元

四阶组PG逻辑

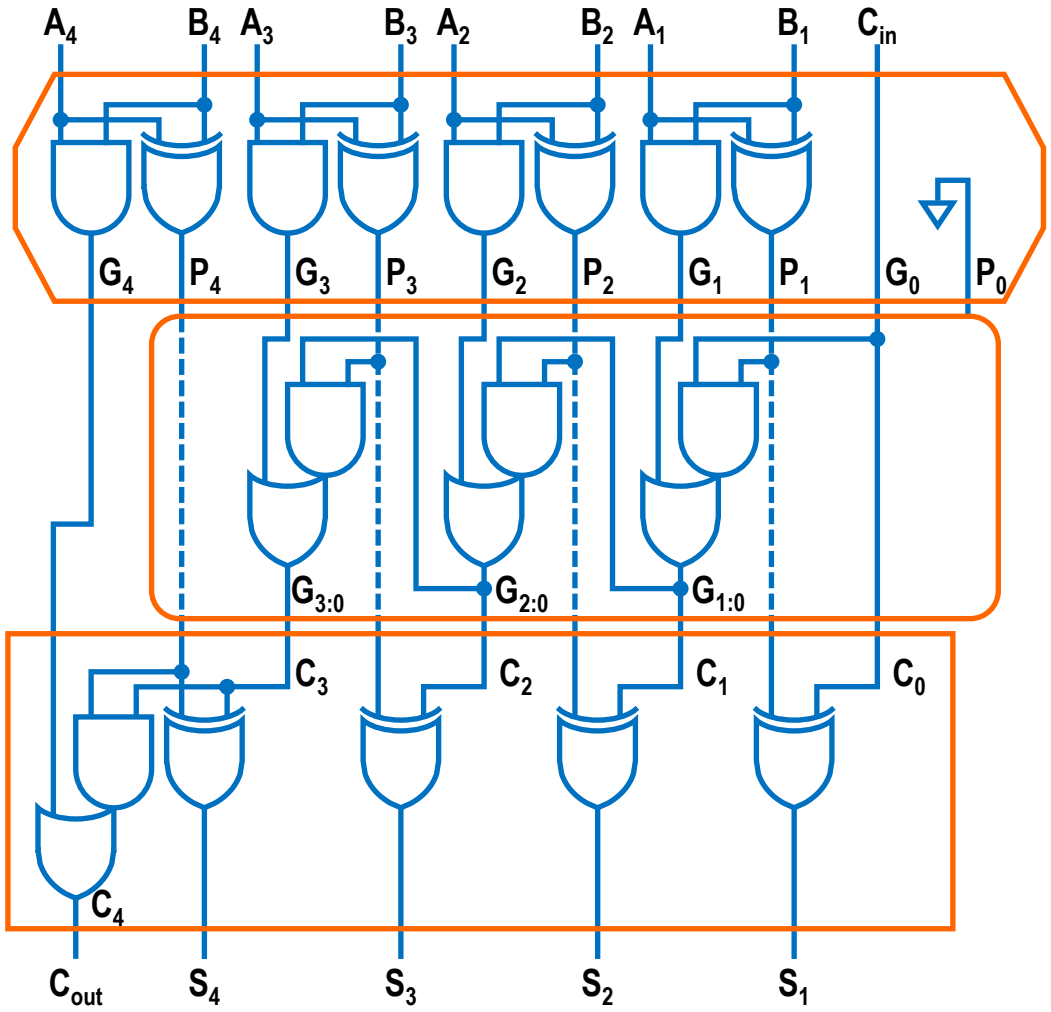
$$
\left.
\begin{aligned}
G_{i:j} &= G_{i:k} + P_{i:k} \cdot G_{k-1:l} + P_{i:k} \cdot P_{k-1:l} \cdot G_{l-1:m} + P_{i:k} \cdot P_{k-1:l} \cdot P_{l-1:m} \cdot G_{m-1:j} \\
&= G_{i:k} + P_{i:k}\left(G_{k-1:l} + P_{k-1:l}\left(G_{l-1:m} + P_{l-1:m} G_{m-1:j}\right)\right) \\
P_{i:j} &= P_{i:k} \cdot P_{k-1:l} \cdot P_{l-1:m} \cdot P_{m-1:j}
\end{aligned}
\right\} (i \geqslant k > l > m > j)
$$



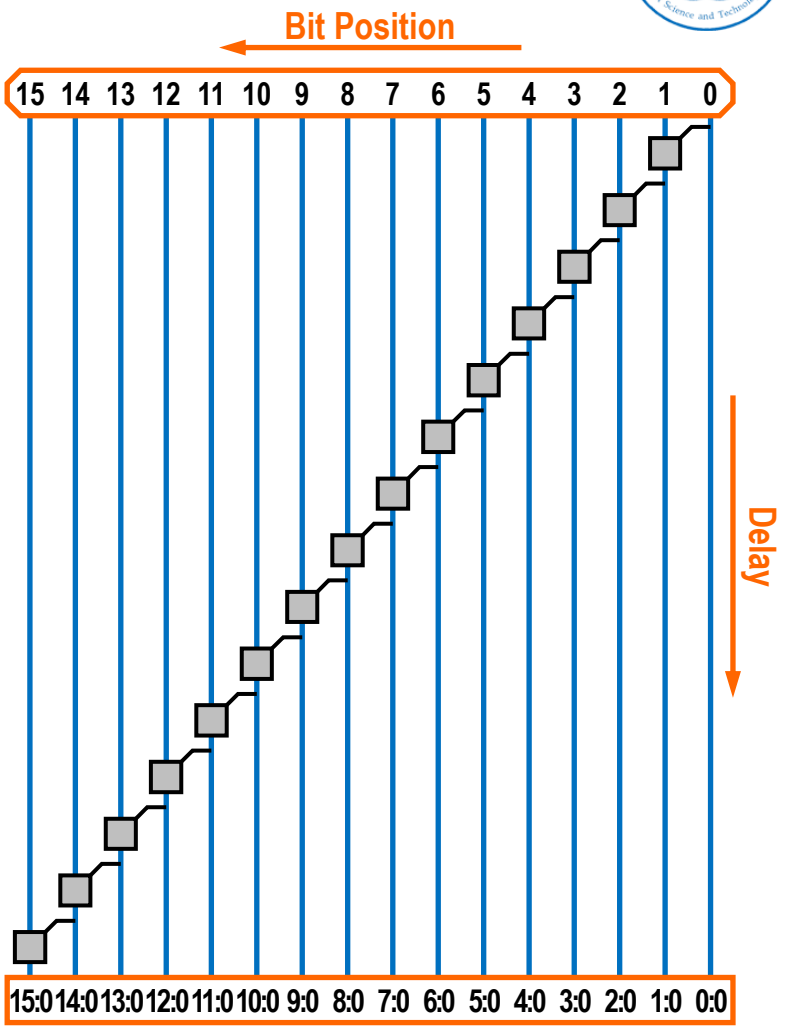


四阶组PG逻辑及电路单元

PG行波进位加法器



采用PG逻辑的四位RCA

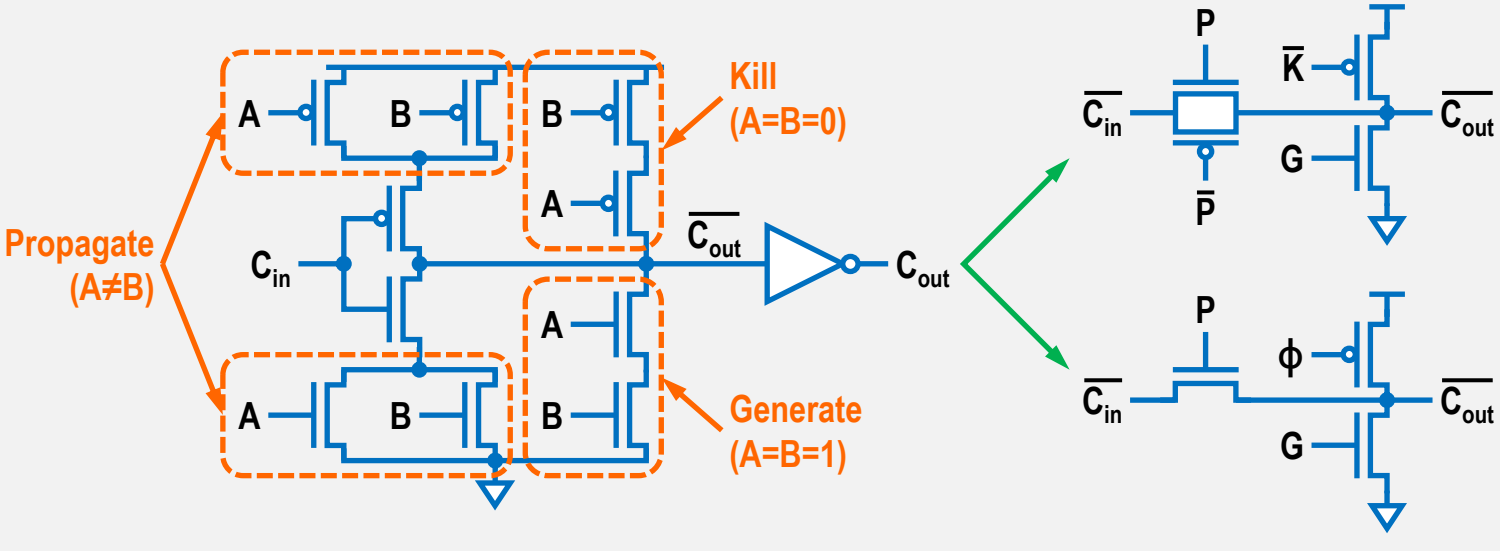


16位RCA的组PG网络

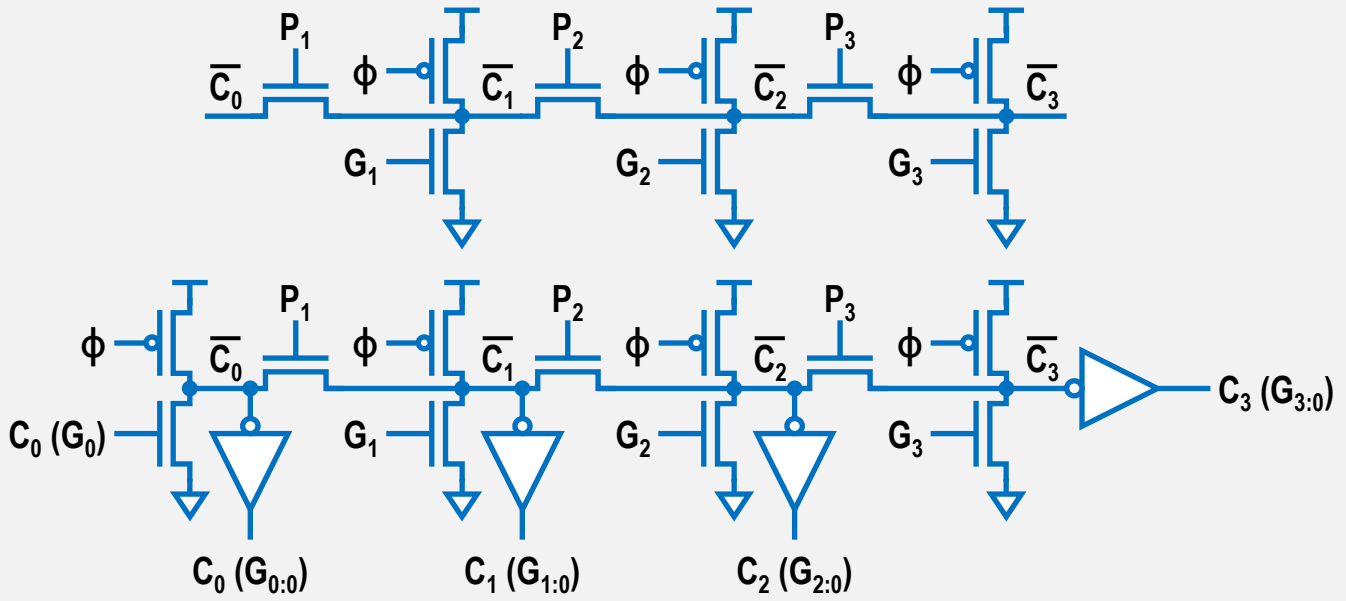
$$G_{i:0} = G_i + P_i \cdot G_{i-1:0}$$

$$t_{\text{ripple}} = t_{pg} + (N-1)t_{\text{AO}} + t_{\text{xor}}$$

曼彻斯特进位链加法器



进位链的设计

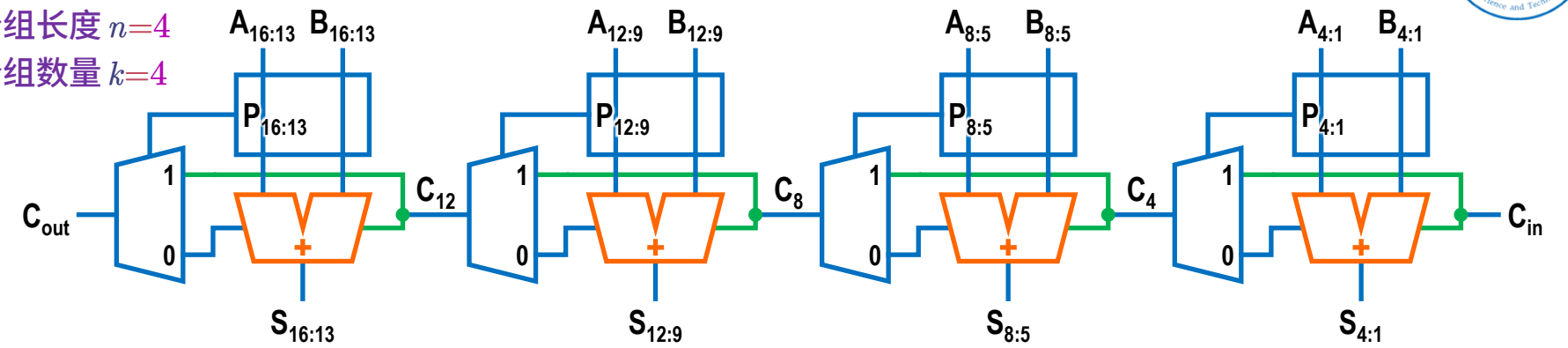


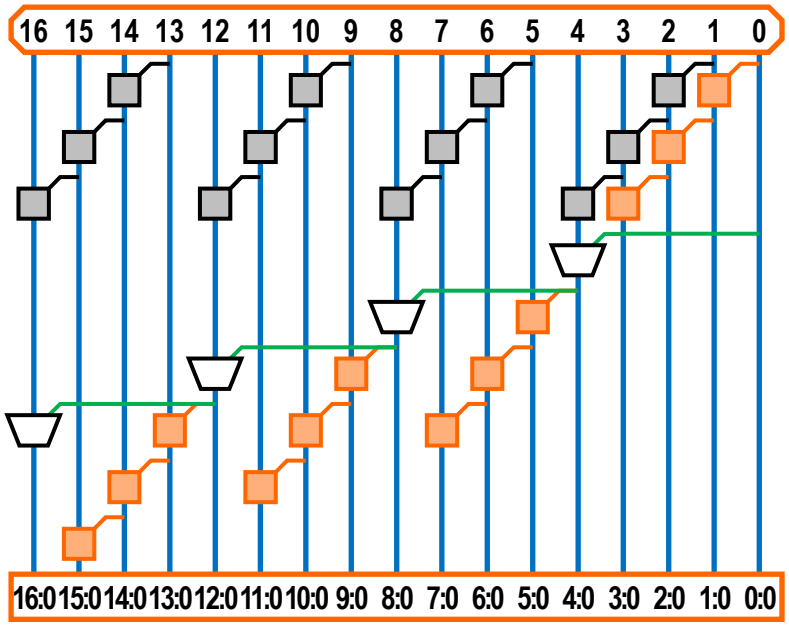
曼彻斯特进位链
(Manchester Carry Chain)

进位跳跃加法器

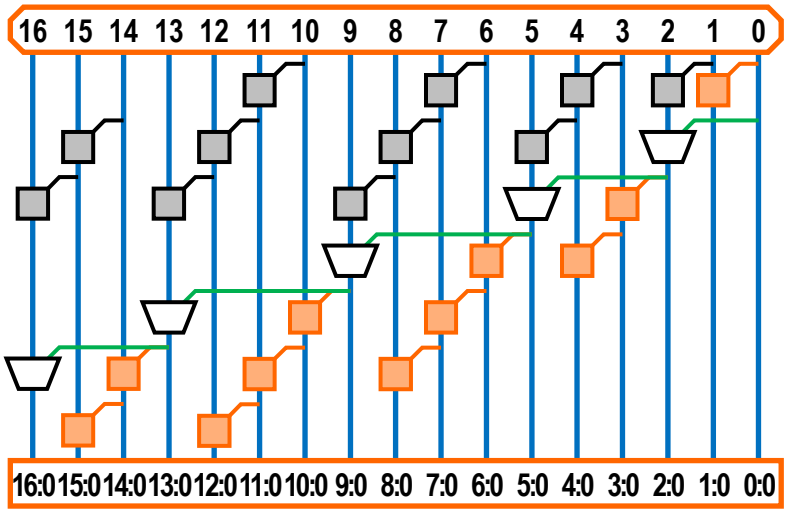
分组长度 $n=4$

分组数量 $k=4$





$$t_{\text{skip}} = t_{pg} + 2(n-1)t_{\text{AO}} + (k-1)t_{\text{mux}} + t_{\text{xor}}$$

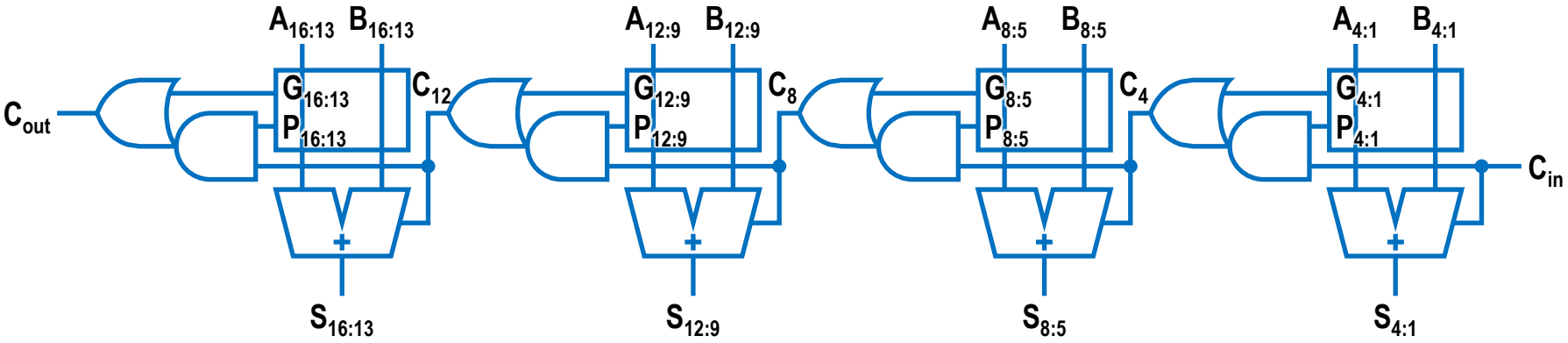


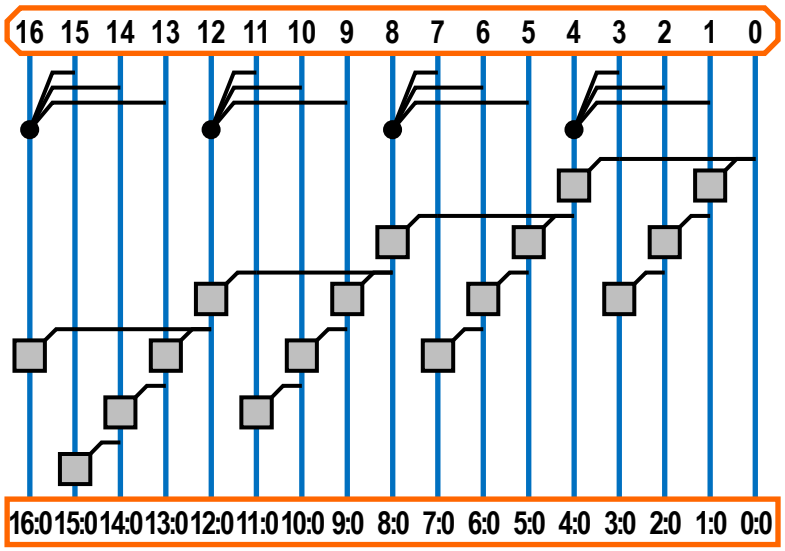
可变组长CSKA，分组长度[2,3,4,4,3]

通过在关键路径始端和末端处采用较短的组，中间部分采用较长的组，可以缩短关键路径

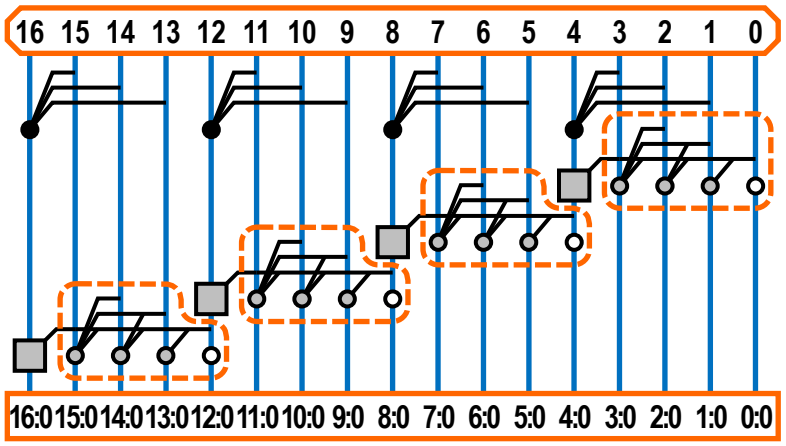
进位跳跃加法器 (Carry-Skip Adder, CSKA)、进位旁路加法器 (Carry-Bypass Adder, CBA)

超前进位加法器





$$t_{\text{cla}} = t_{pg} + t_{pg(n)} + [(n-1)+(k-1)]t_{\text{AO}} + t_{\text{xor}}$$

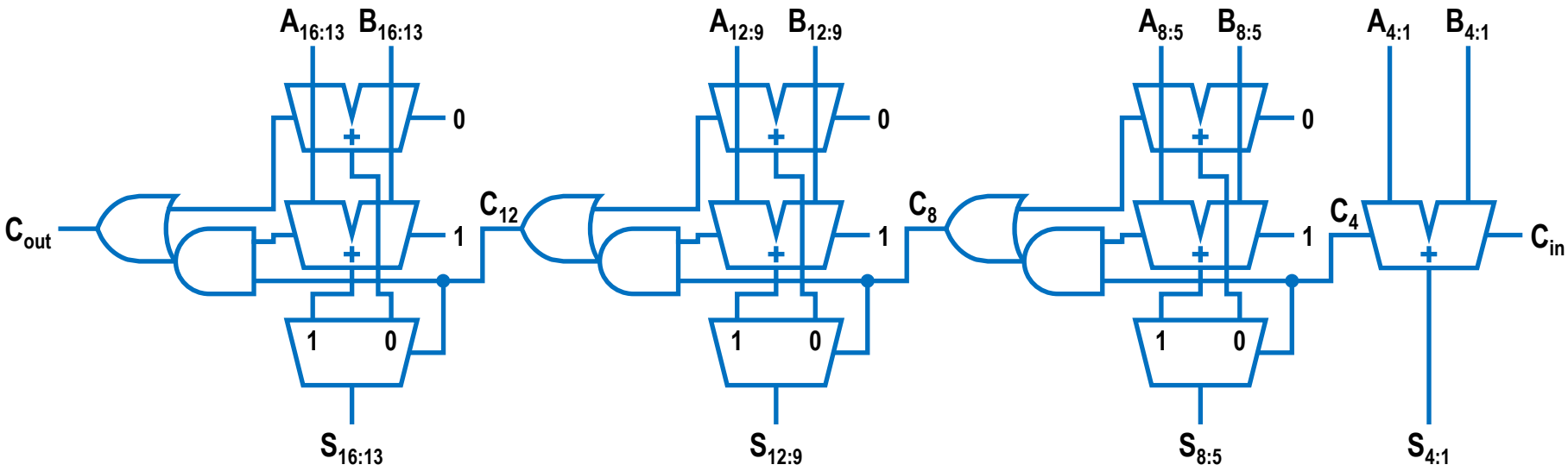


改进的CLA

采用高阶单元并行计算多个进位来减少延时，例如，曼彻斯特进位链或并行运算的静态门

超前进位加法器 (Carry-Lookahead Adder, CLA)

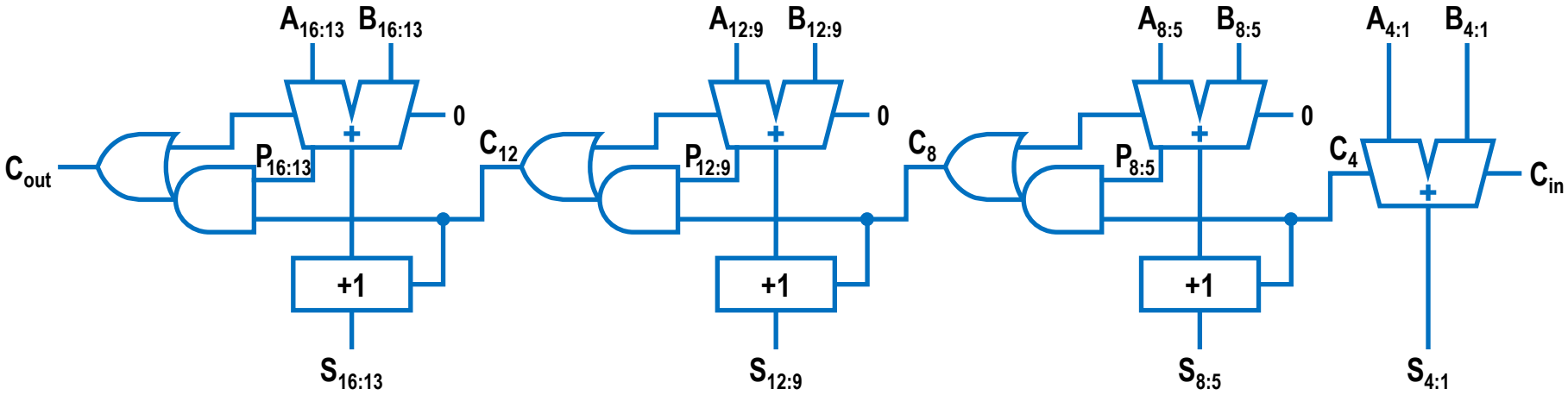
进位选择加法器

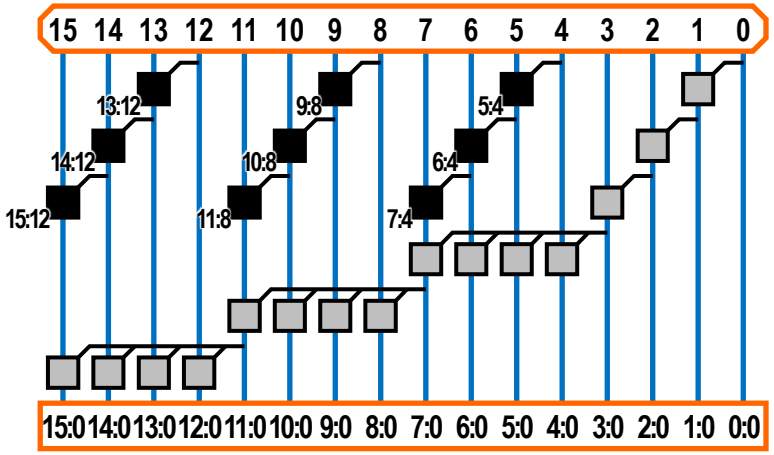


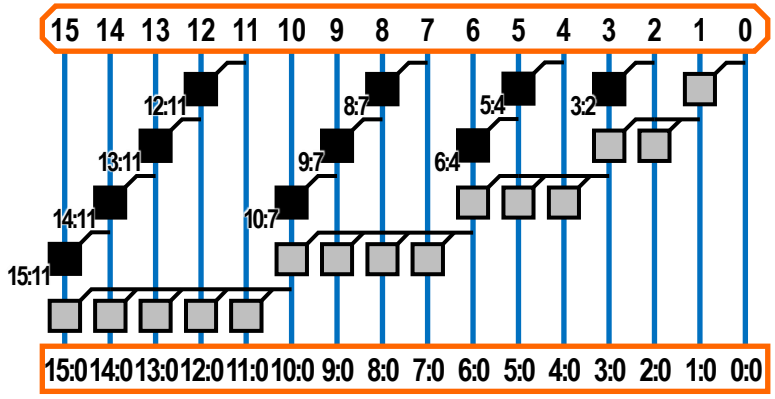
$$t_{\text{select}} = t_{pg} + [n + (k-2)]t_{\text{AO}} + t_{\text{mux}}$$

进位选择加法器 (Carry-Select Adder, CSLA)

进位增量加法器





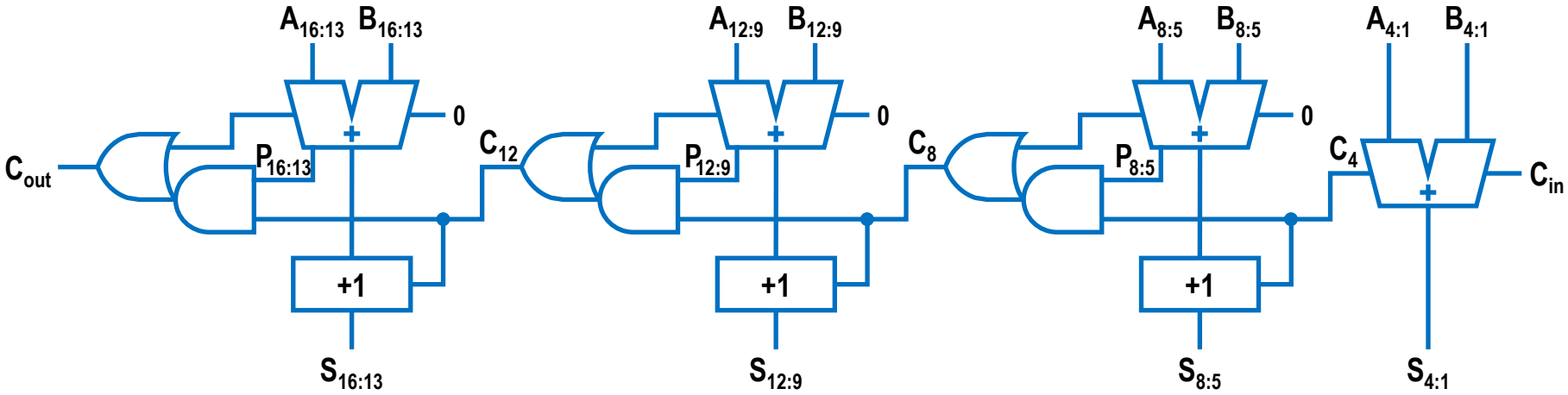


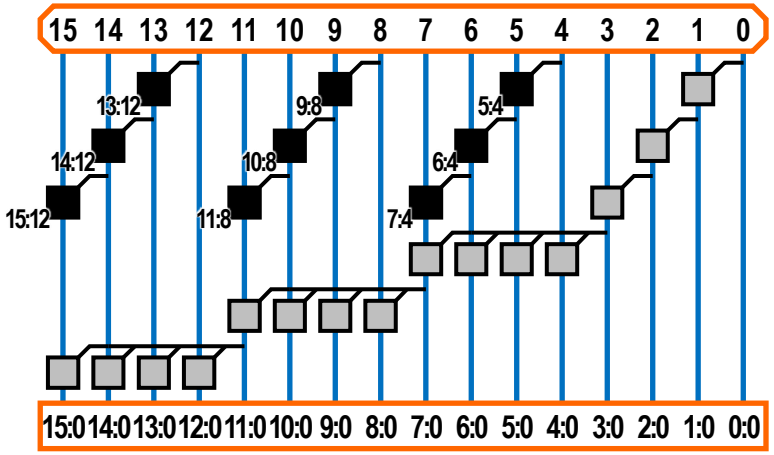
$$t_{\text{increment}} = t_{pg} + [(n-1)+(k-1)]t_{\text{AO}} + t_{\text{xor}}$$

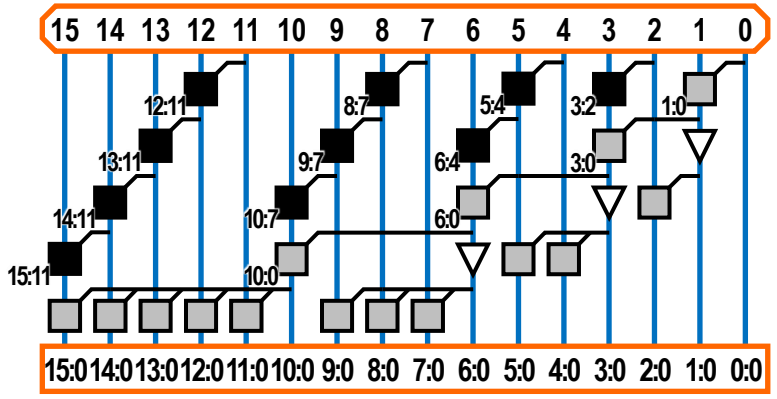
$$t_{\text{increment}} \approx t_{pg} + \sqrt{2N}\,t_{\text{AO}} + t_{\text{xor}}$$

进位增量加法器 (Carry-Increment Adder, CIA)

进位增量加法器







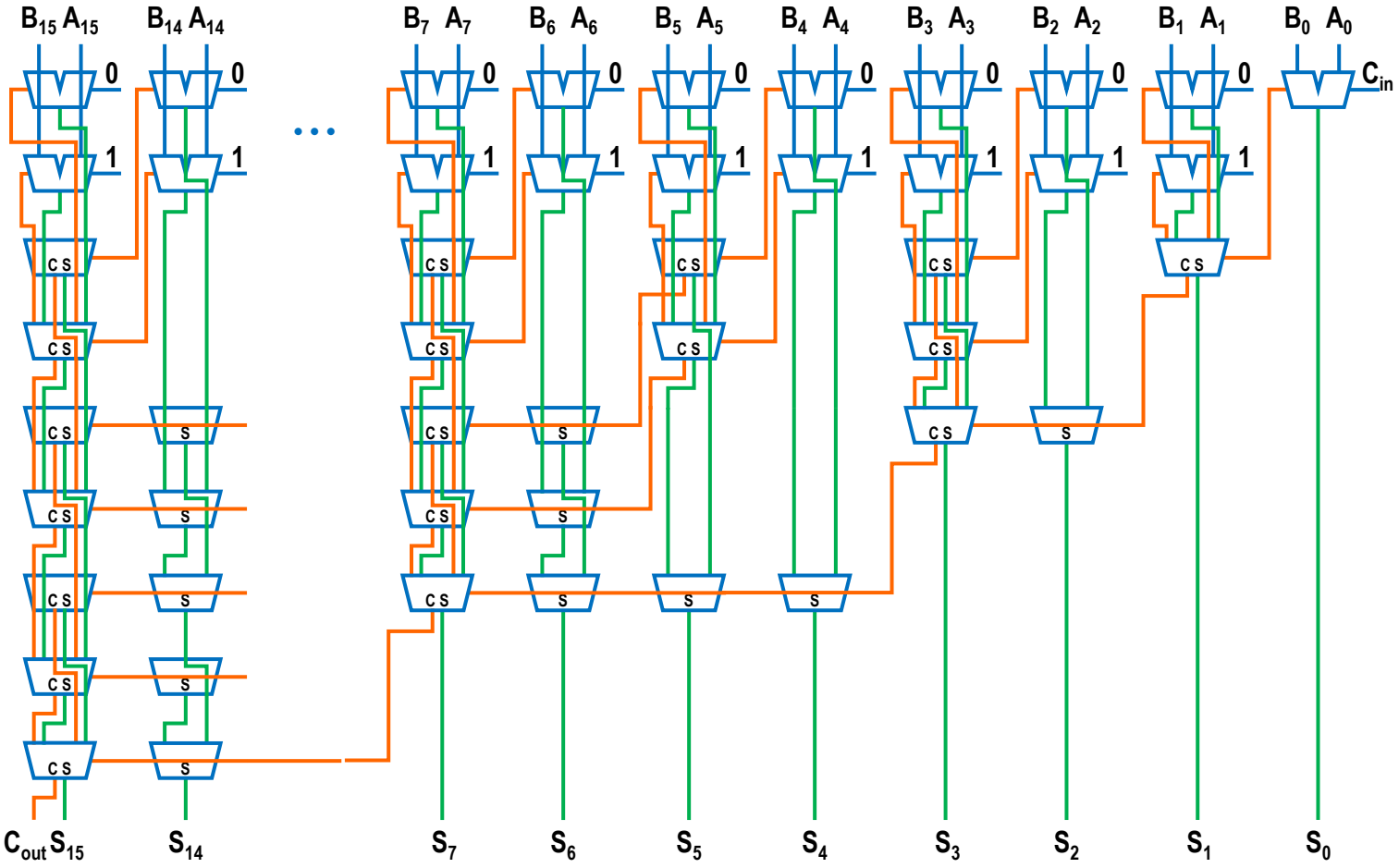
$$t_{\text{increment}} = t_{pg} + [(n-1)+(k-1)]t_{\text{AO}} + t_{\text{xor}}$$

$$t_{\text{increment}} \approx t_{pg} + \sqrt{2N}\,t_{\text{AO}} + t_{\text{xor}}$$

进位增量加法器 (Carry-Increment Adder, CIA)

条件和加法器

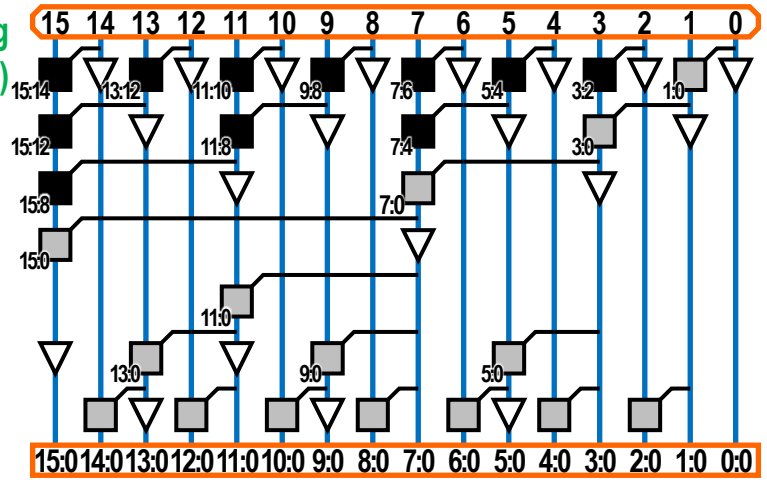
Block Width	Block Carry In
1	0
	1
2	0
	1
4	0
	1
8	0
	1
16	C_{in}



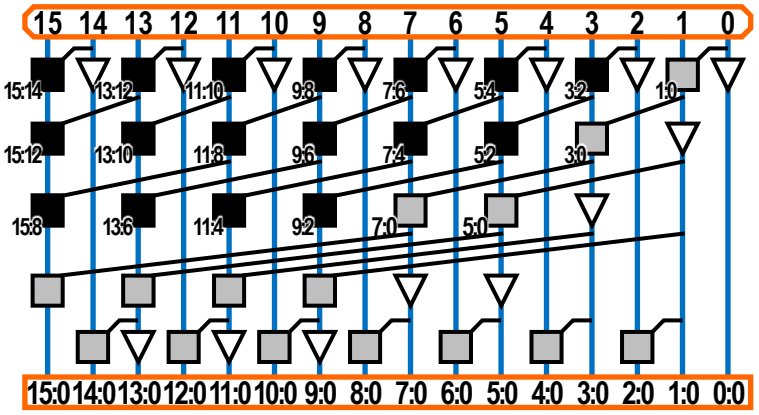
条件和加法器 (Conditional-Sum Adder, COSA)

并行前缀加法器 (Parallel-Prefix Adder, PPA)

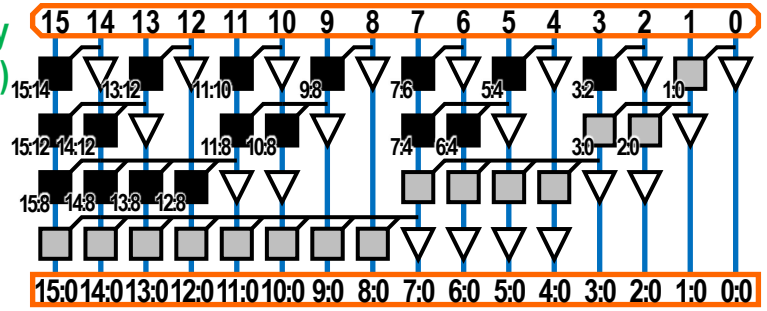
Brent-Kung
(PPA-BK)



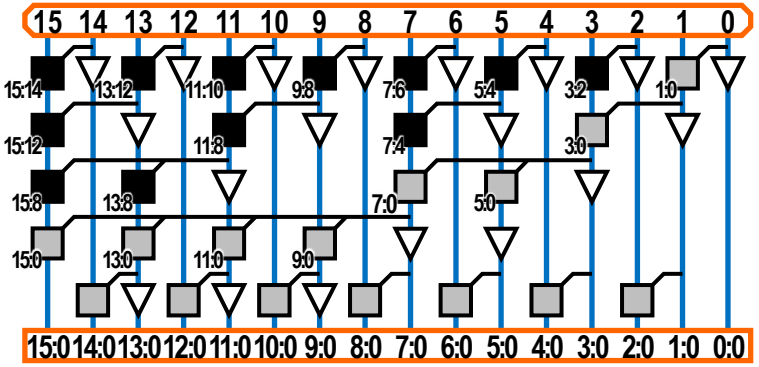
Han-Carlson
(PPA-HC)
(BK+KS)



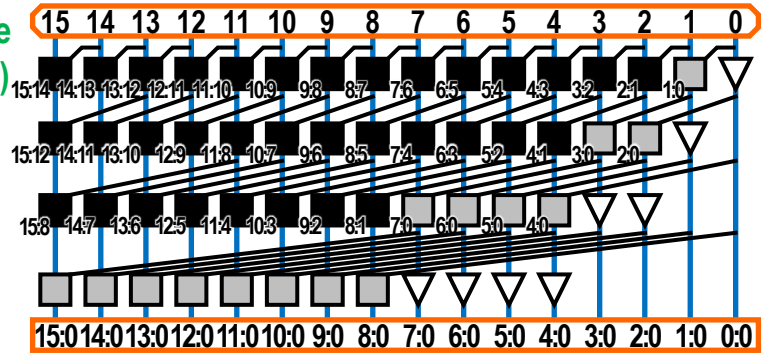
Sklansky
(PPA-SK)



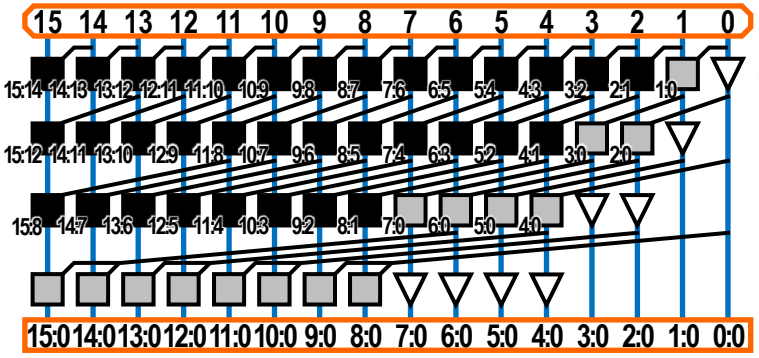
Ladner-Fischer
(PPA-LF)
(BK+SK)



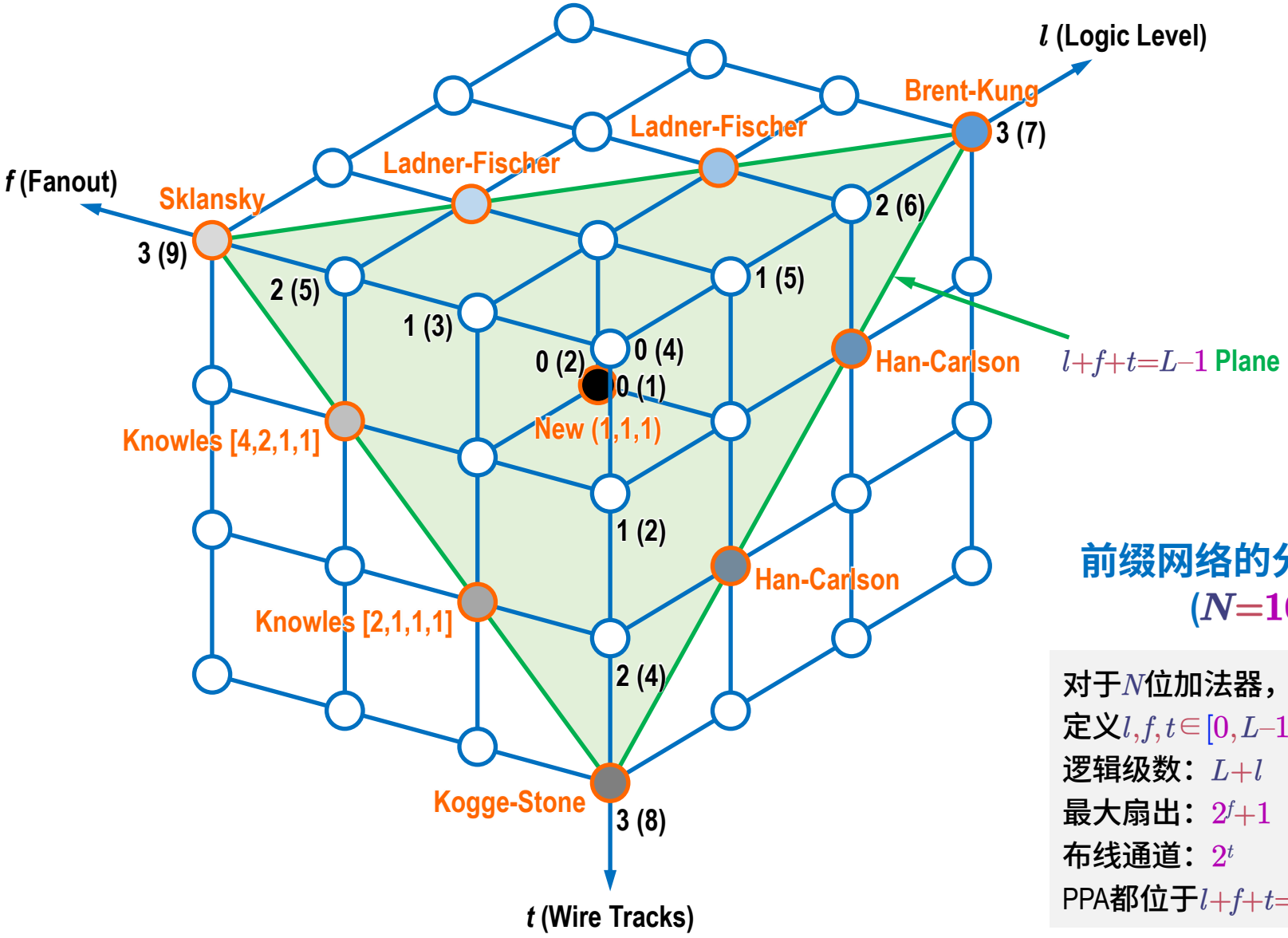
Kogge-Stone
(PPA-KS)



Knowles
(PPA-KN)
[2,1,1,1]
(SK+KS)



并行前缀加法器的分类



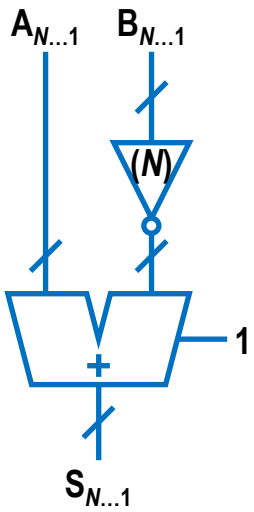
前缀网络的分类方法 (N=16)

对于N位加法器，取$L=\log_2 N$，
定义$l,f,t\in[0,L-1]$为：
逻辑级数：$L+l$
最大扇出：2^f+1
布线通道：2^t
PPA都位于$l+f+t=L-1$平面上

进位传播加法器小结

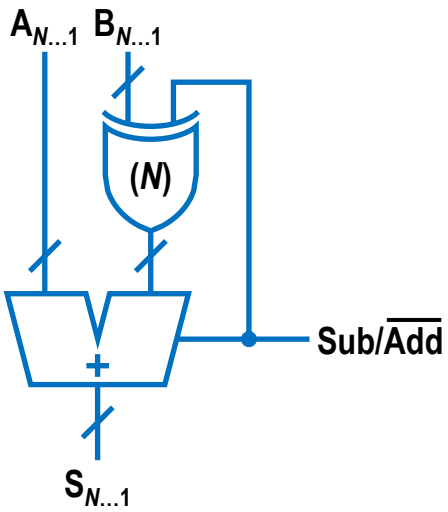
加法器结构	分类	逻辑级数	最大扇出数	布线通道数	单元数
RCA		N–1	1	1	N
CSKA (n=4)		$N/4$+5	2	1	1.25N
CIA (n=4)		$N/4$+2	4	1	2N
CIA (Variable Group)		$\sqrt{2N}$	$\sqrt{2N}$	1	2N
Brent-Kung	(L–1, 0, 0)	2$\log_2 N$–1	2	1	2N
Sklansky	(0, L–1, 0)	$\log_2 N$	$N/2$+1	1	0.5$N\log_2 N$
Kogge-Stone	(0, 0, L–1)	$\log_2 N$	2	$N/2$	$N\log_2 N$
Han-Carlson	(1, 0, L–2)	$\log_2 N$+1	2	$N/4$	0.5$N\log_2 N$
Ladner Fischer (l=1)	(1, L–2, 0)	$\log_2 N$+1	$N/4$+1	1	0.25$N\log_2 N$
Knowles [2,1,⋯,1]	(0, 1, L–2)	$\log_2 N$	3	$N/4$	$N\log_2 N$

减法器



$$S = A - B = A + \overline{B} + 1$$

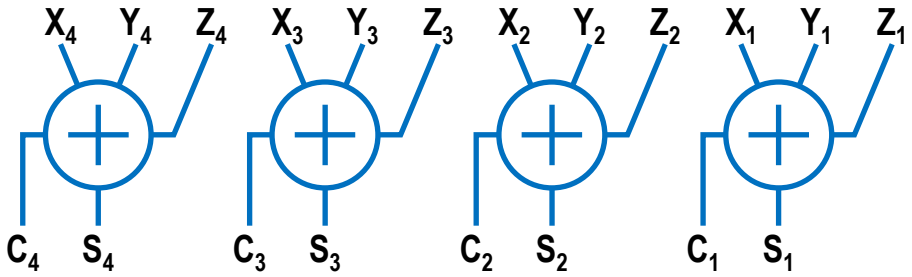
减法器

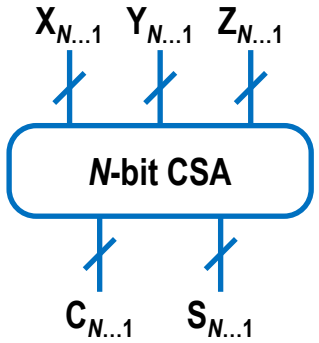


$$S = A \mp B$$

加法/减法器

进位保留加法器

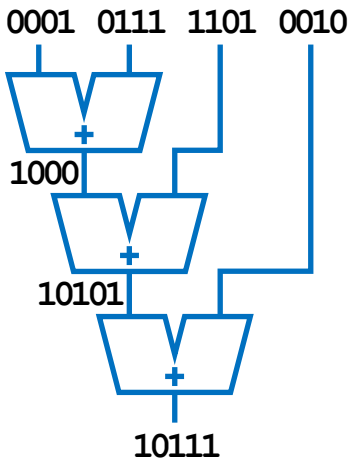


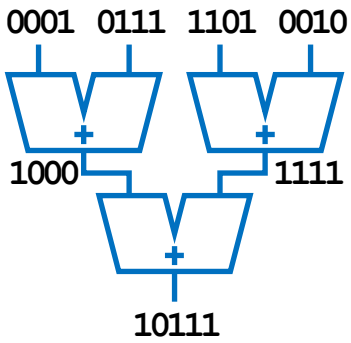


$$X + Y + Z = S + 2C$$

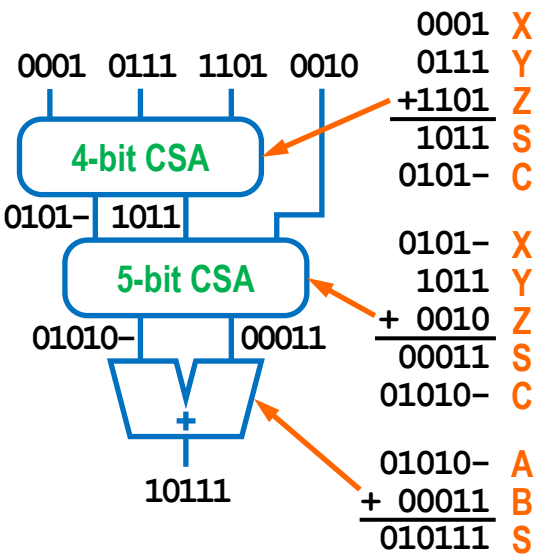
[3:2]进位保留加法器 ([3:2] Carry-Save Adder, CSA)

多输入加法器

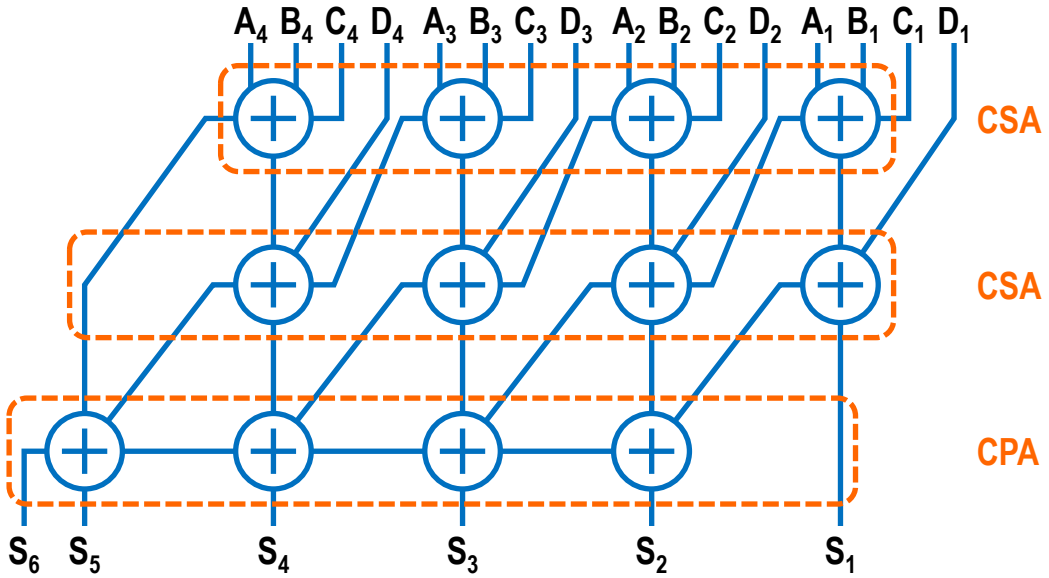




基于CPA的多输入加法



基于CSA的多输入加法



多输入加法器

标志前缀加法器 (Flagged Prefix Adder)

- **模2^n–1加法**

  $(A+B) \bmod (2^n-1)$:

  ① $A+B$

  ② $\text{inc} = G_{n:1} + P_{n:1}$

- **绝对差**

  $|A-B|$:

  ① $A+\overline{B}$

  ② $\text{inc} = G_{n:1}$, $\text{inv} = \overline{G_{n:1}}$

  ③ $S_i = (P_i \oplus \text{inv}) \oplus G'_{i-1:1}$

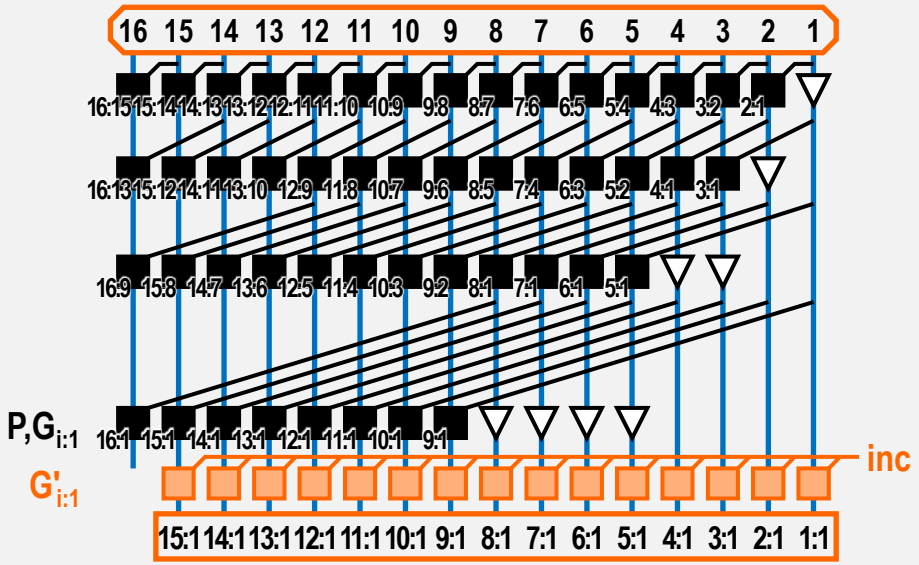
- **原码加法**

  Signs Agree:

  $S_{n-1:1} = A_{n-1:1} + B_{n-1:1}$, $S_n = A_n$

  Signs Differ:

  $S_{n-1:1} = |A_{n-1:1} - B_{n-1:1}|$, $S_n = A_n \oplus \overline{G_{n-1:1}}$

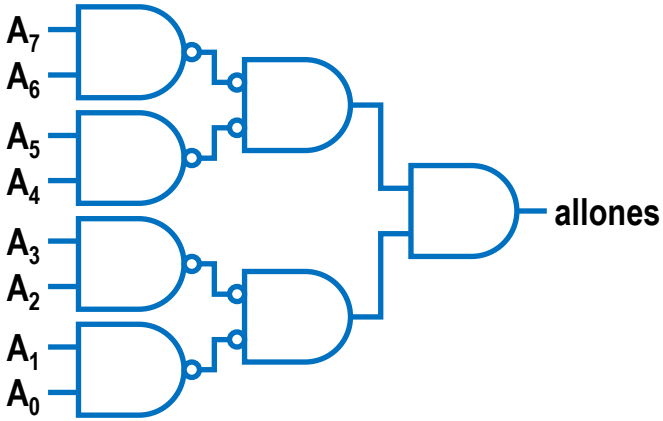


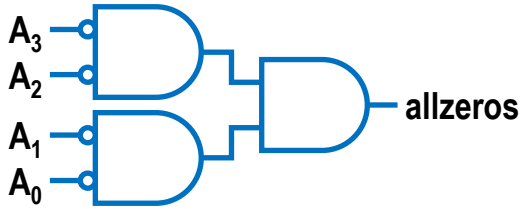
$G'_{i:1} = G_{i:1} + P_{i:1} \cdot \text{inc}$, $S_i = P_i \oplus G'_{i-1:1}$

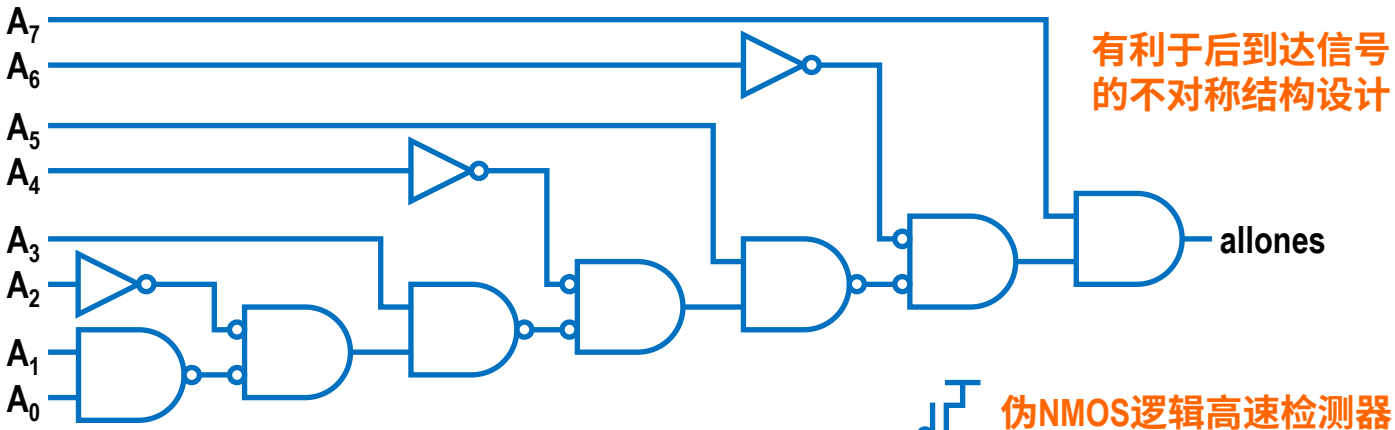
标志前缀Kogge-Stone加法器

基本运算

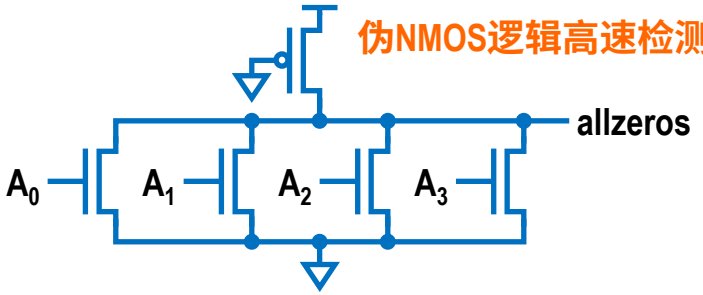
1/0检测器







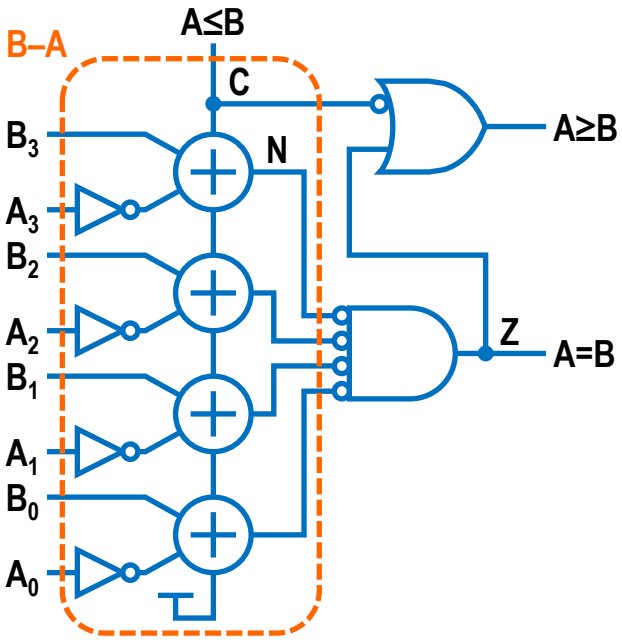
有利于后到达信号
的不对称结构设计



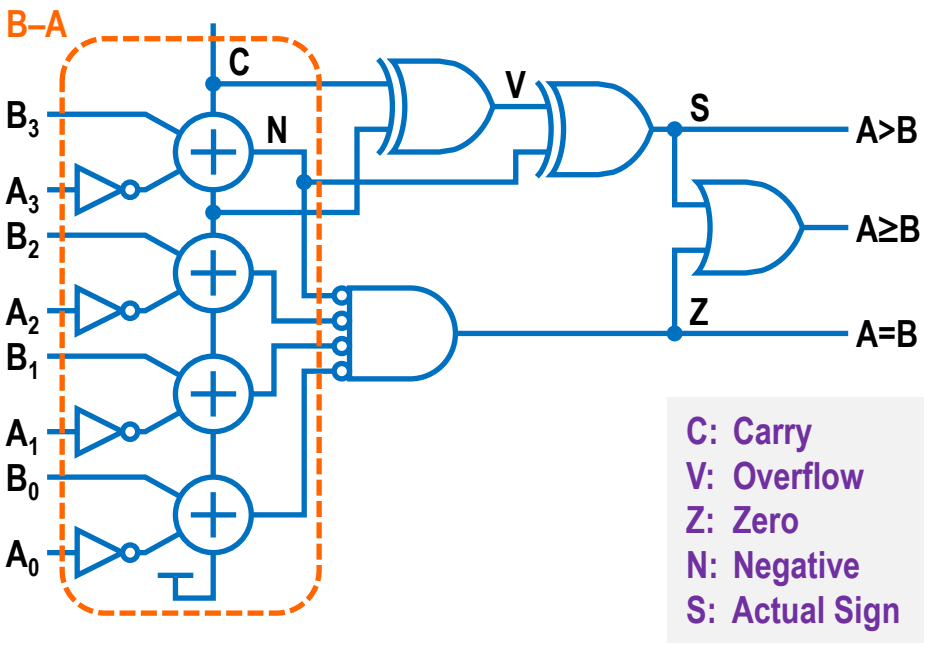
伪NMOS逻辑高速检测器

1/0检测器

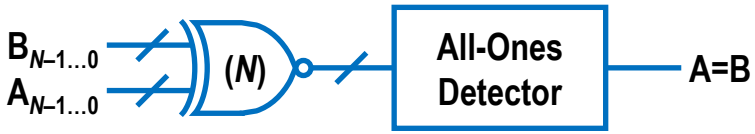
比较器



无符号数值比较器

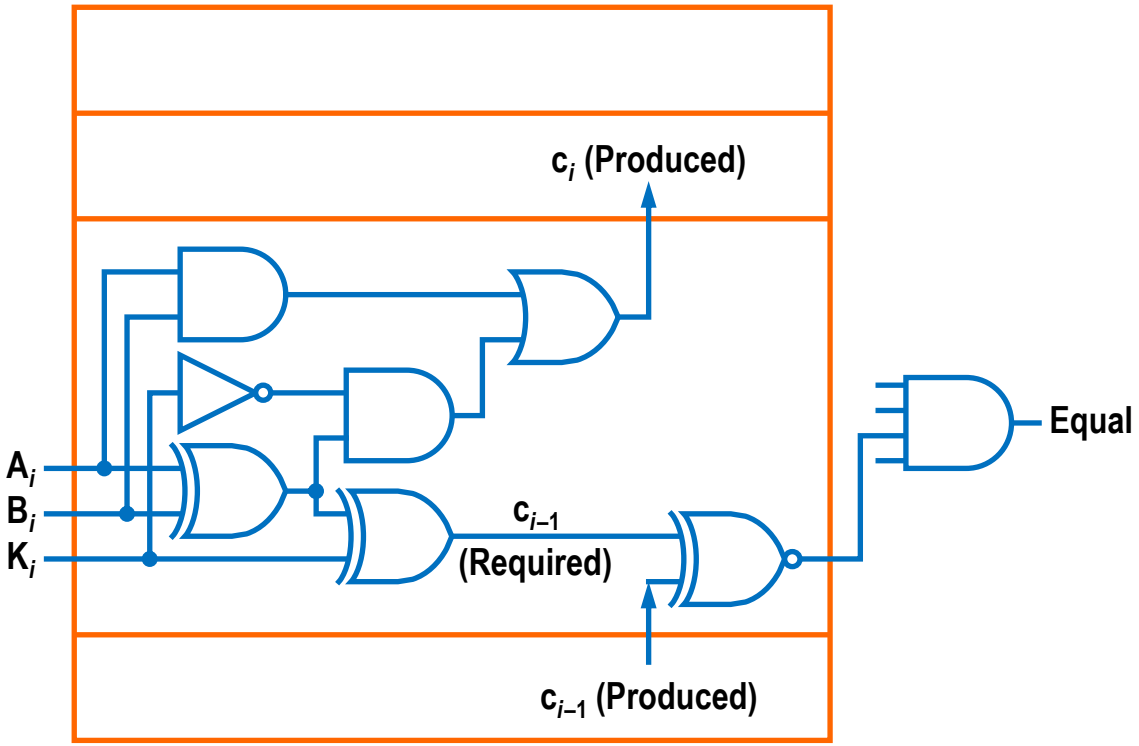


有符号数值比较器



相等比较器

K=A+B比较器



$$c_{i-1}(\text{Required}) = A_i \oplus B_i \oplus K_i$$

$$c_{i-1}(\text{Produced}) = (A_{i-1} \oplus B_{i-1}) \cdot \overline{K_{i-1}} + A_{i-1} \cdot B_{i-1}$$

K=A+B比较器

计数器

计数器的种类

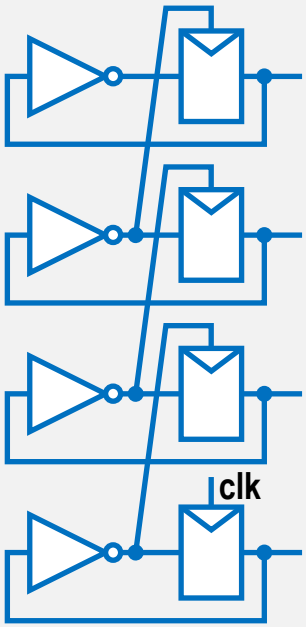
- 二进制计数器 (Binary Counter)
  - N位二进制计数器按二进制顺序经过2^N个输出值
  - 简单设计的最小周期时间随N的增加而增加，快速设计在固定时间内完成操作
- 线性反馈移位寄存器 (Linear-Feedback Shift Register, LFSR)
  - N位线性反馈移位寄存器按随机顺序经过至多2^N-1个输出值
  - 具有很短的最小周期时间，可用作伪随机数发生器或极高速计数器

计数器的特点

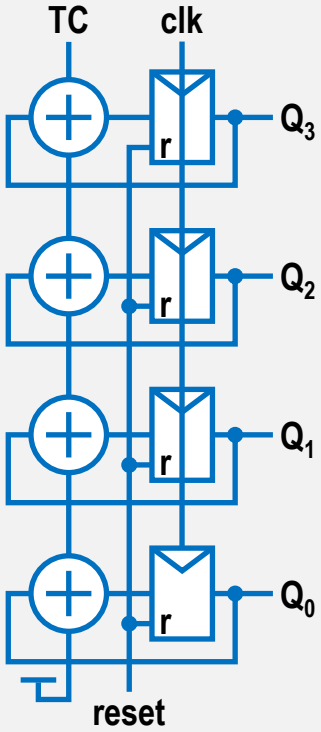
- 可复位 (Resettable)　当复位信号有效时，计数值复位至0
- 可装载 (Loadable)　当装载信号有效时，可将数值装载入计数器
- 使能 (Enable)　当使能信号有效时，计数器按照时钟周期计数
- 可逆 (Reversible)　根据递增/递减信号，计数器增1或减1
- 终点计数 (Terminal Count)　当计数器溢出时，终点计数输出信号有效

二进制计数器

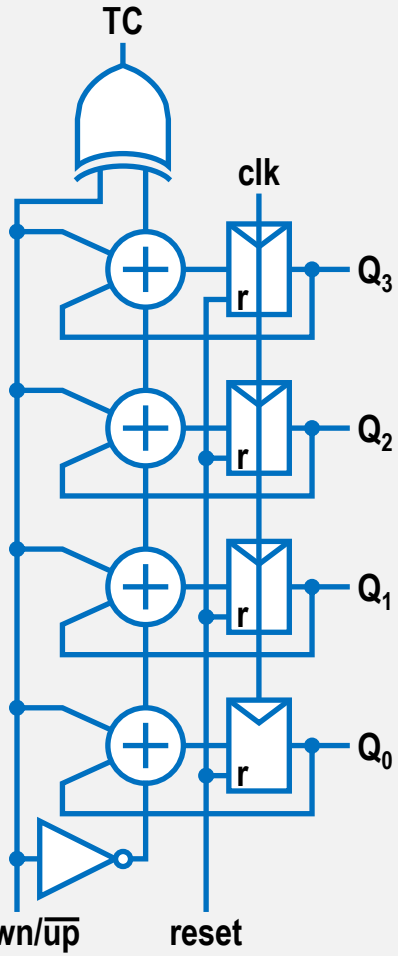
异步行波进位计数器



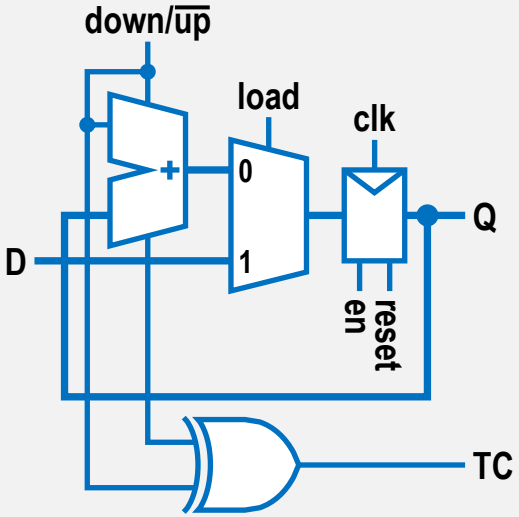
同步递增计数器
同步增量器



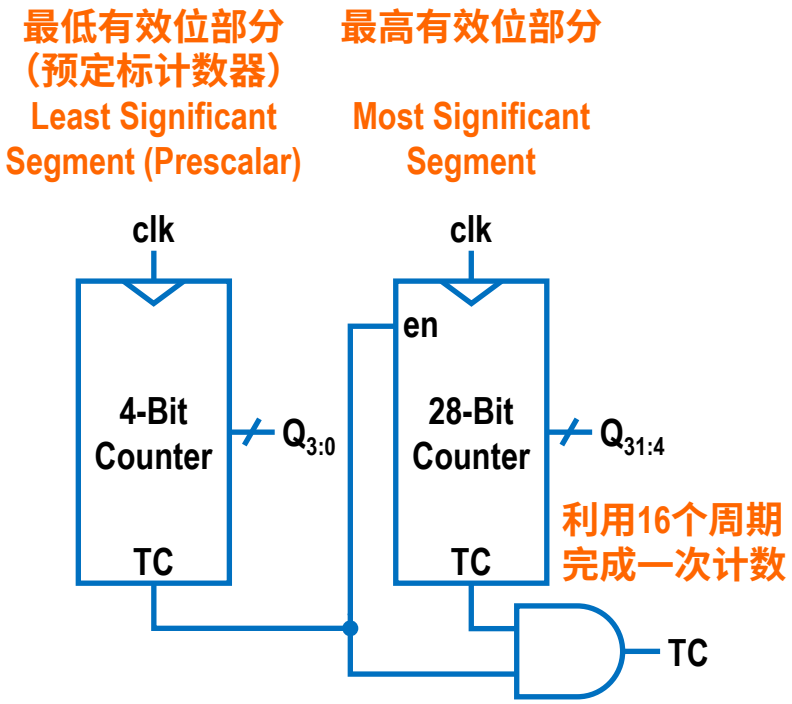
同步双向计数器



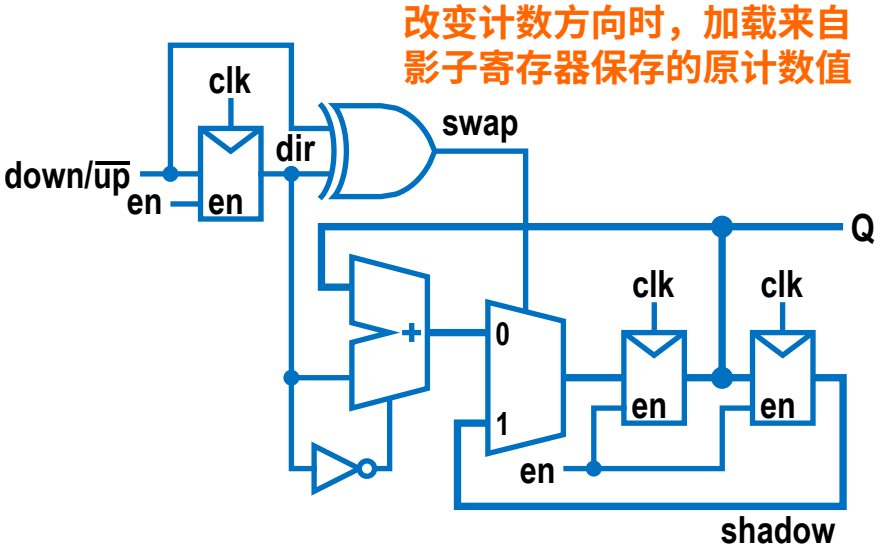
具有复位、装载、使能控制的同步双向计数器



高速二进制计数器

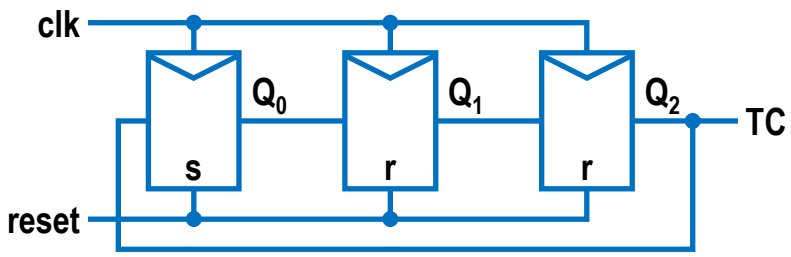


高速二进制计数器

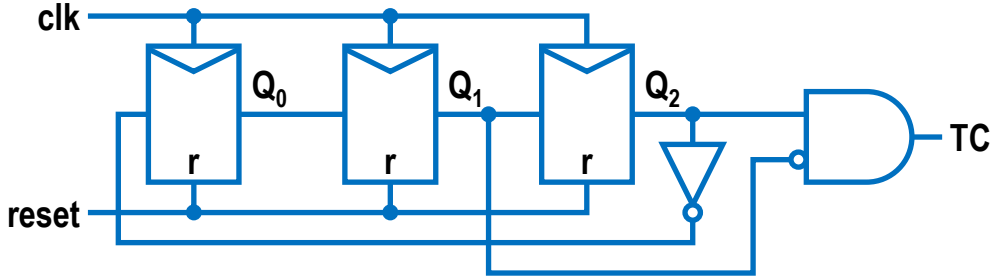


高速二进制双向计数器
（最高有效位部分）

环形计数器和Johnson计数器



环形计数器 (Ring Counter)

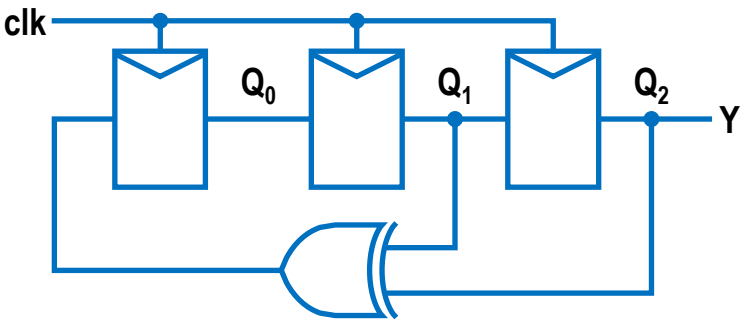


Johnson计数器、 Mobius计数器

Johnson计数器序列

Cycle	Q_0	Q_1	Q_2	TC
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	0	1	1	0
5	0	0	1	1
6	0	0	0	0
Repeats Forever				

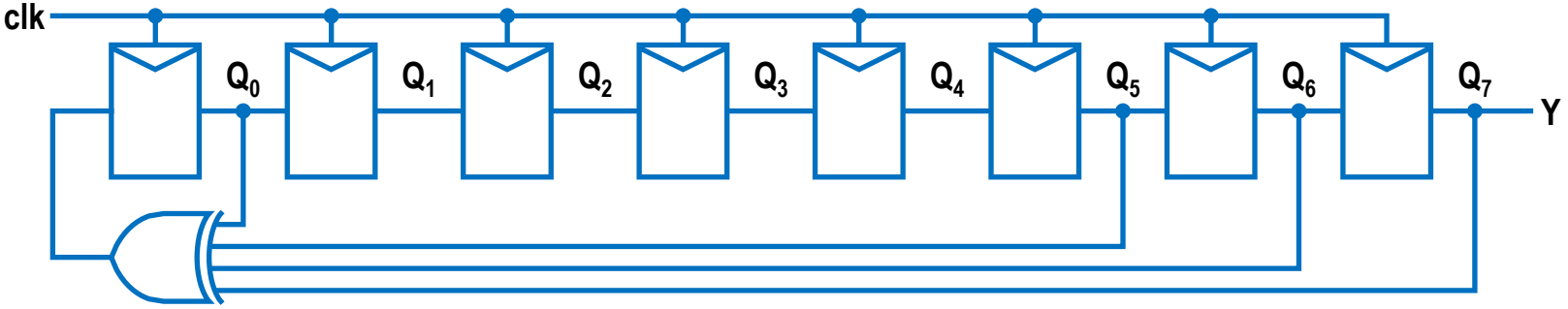
线性反馈移位寄存器



$1 + x^2 + x^3$

LFSR序列

Cycle	Q_0	Q_1	Q_2/Y
0	1	1	1
1	0	1	1
2	0	0	1
3	1	0	0
4	0	1	0
5	1	0	1
6	1	1	0
7	1	1	1
Repeats Forever			

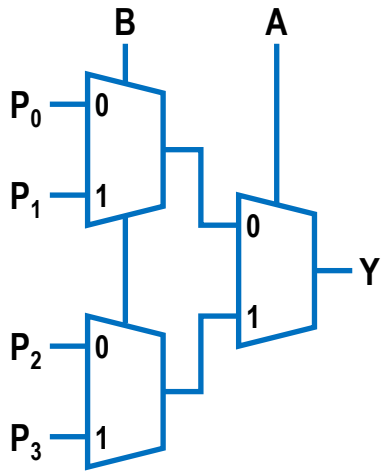


$1 + x + x^6 + x^7 + x^8$

线性反馈移位寄存器

(Linear-Feedback Shift Register, LFSR)

布尔逻辑运算

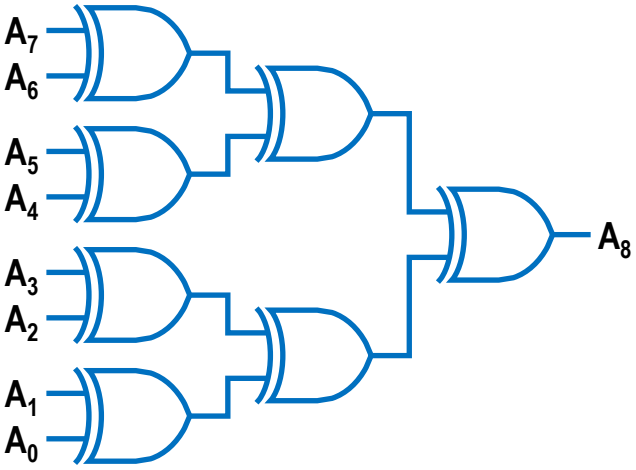


布尔逻辑单元

布尔逻辑运算的实现方式

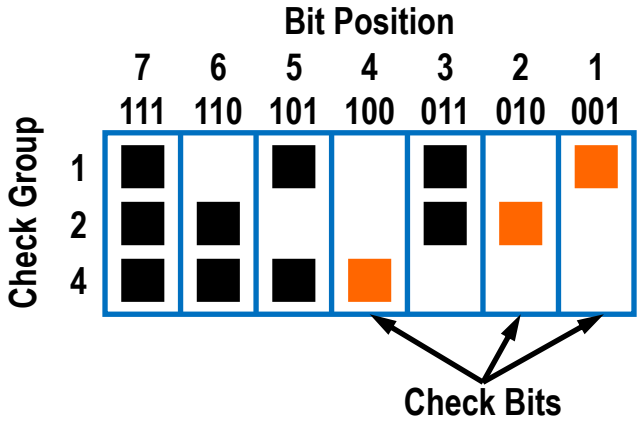
运算	P_0	P_1	P_3	P_4
AND(A, B)	0	0	0	1
OR(A, B)	0	1	1	1
XOR(A, B)	0	1	1	0
NAND(A, B)	1	1	1	0
NOR(A, B)	1	0	0	0

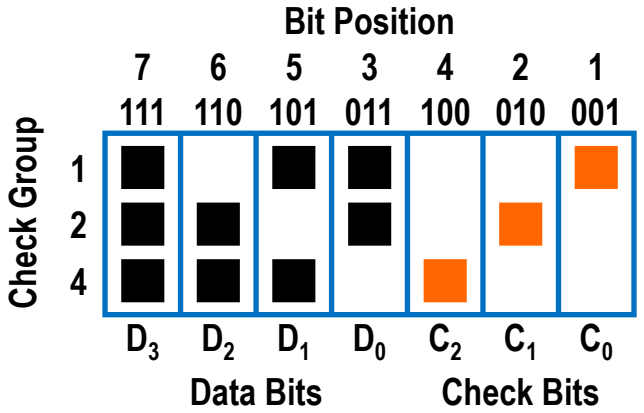
检错码和纠错码



$$A_n = \text{PARITY} = \bigoplus_{i=0}^{n-1} A_i$$

八位奇偶校验生成器





$$C_0 = D_3 \oplus D_1 \oplus D_0$$

$$C_1 = D_3 \oplus D_2 \oplus D_0$$

$$C_2 = D_3 \oplus D_2 \oplus D_1$$

奇偶校验矩阵

格雷码 (Gray Code)

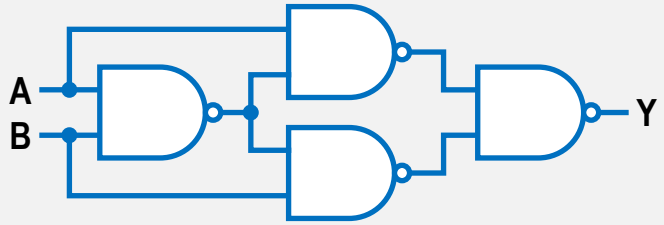
二进制反射格雷码
(Binary-Reflected Gray Code)

Number	Binay	Gray Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

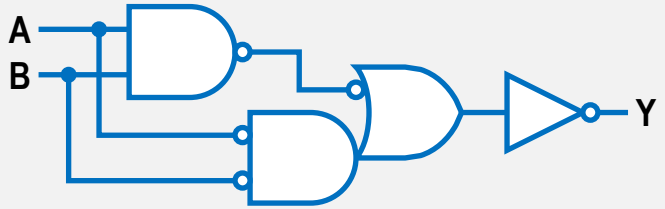
$$G_{N-1} = B_{N-1}$$
$$G_i = B_{i+1} \oplus B_i$$

$$B_{N-1} = G_{N-1}$$
$$B_i = B_{i+1} \oplus G_i$$

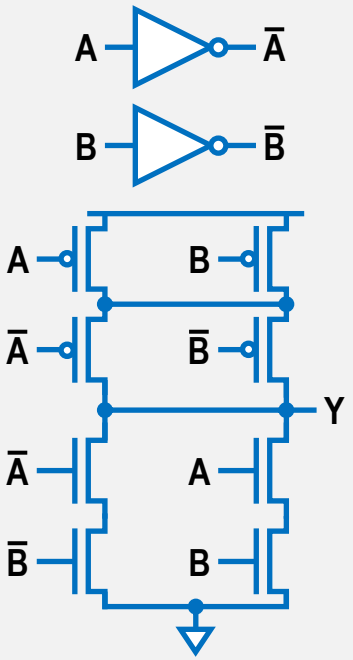
异或门/同或门

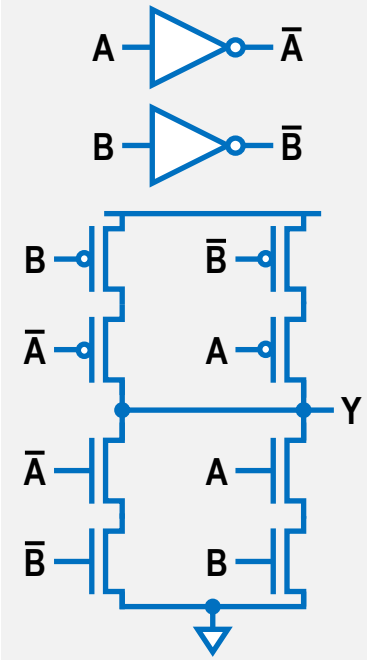


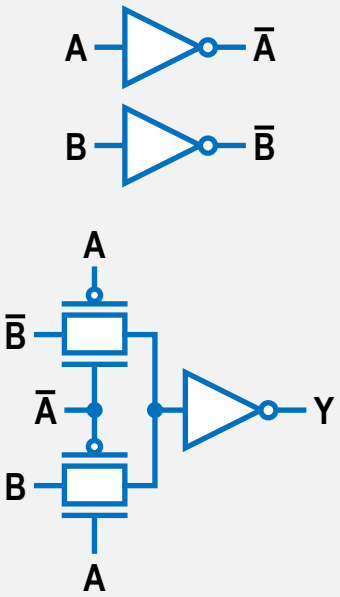
$$Y = A \cdot \overline{AB} + B \cdot \overline{AB} = A\overline{B} + \overline{A}B$$

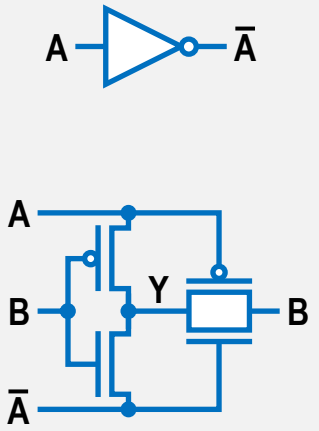


$$Y = \overline{AB + \overline{A}\,\overline{B}} = \overline{A \odot B}$$

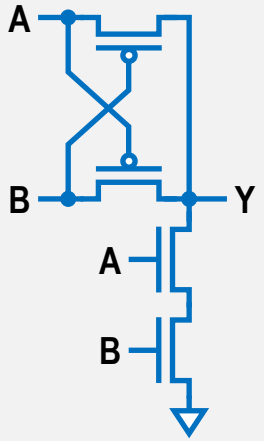








很简洁但
电平不可恢复

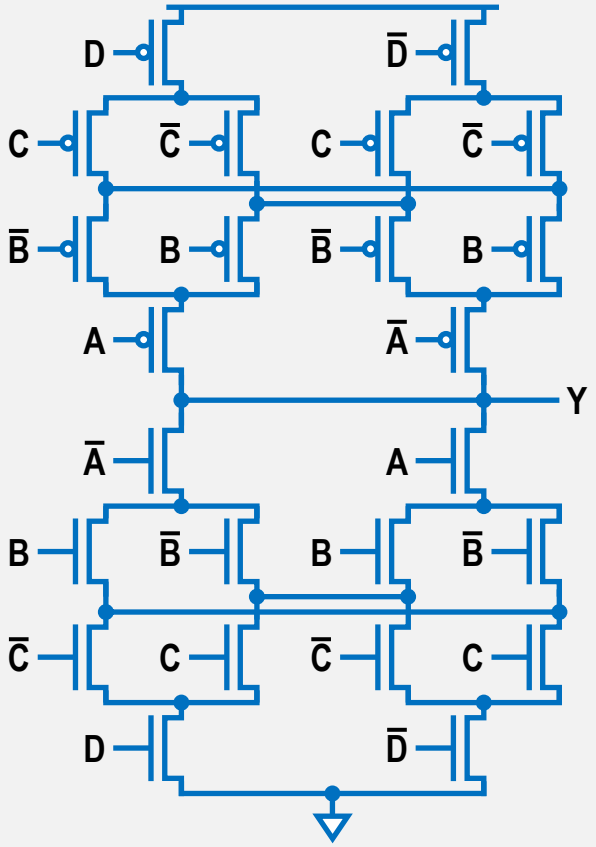


简洁但不能
达到轨至轨摆幅

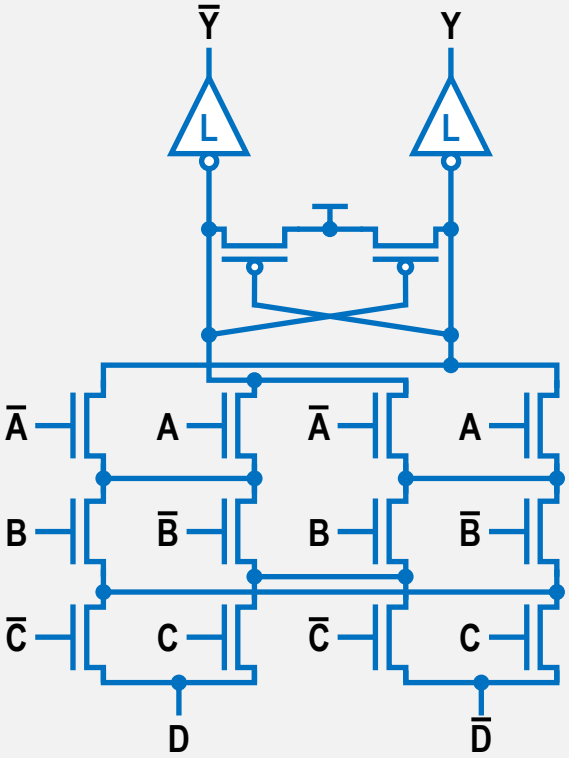
静态二输入异或门设计

多输入异或门

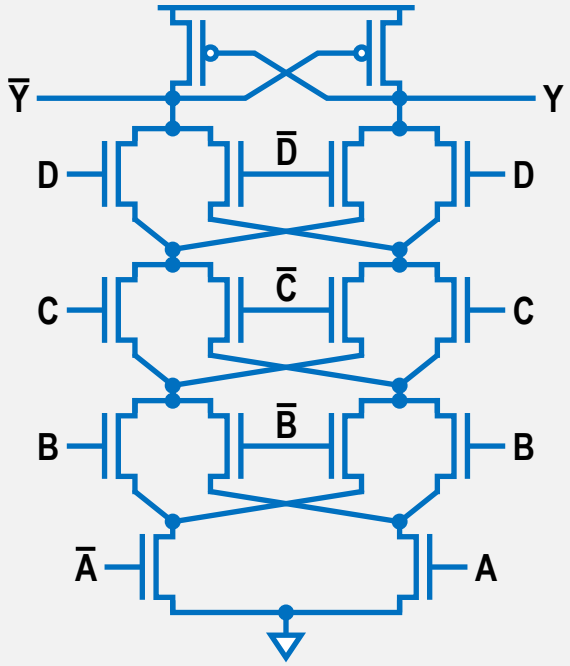
静态CMOS逻辑



互补传输管逻辑 (CPL)



级联电压开关逻辑 (CVSL)



四输入异或门设计

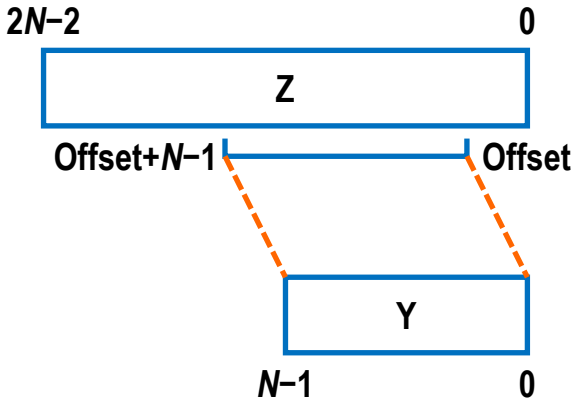
移位器

- **固定位数移位**
  - 使用固定连线即可完成
- **可变位数移位**
  - 移位类型
    - 循环移位：一端空出的位置由另一端移出的数字填充
    - 逻辑移位：左移或右移数字，空位由零填充
    - 算术移位：算术右移时，空位用符号位填充
  - 移位器结构
    - 阵列移位器、对数移位器
    - 漏斗移位器、桶式移位器

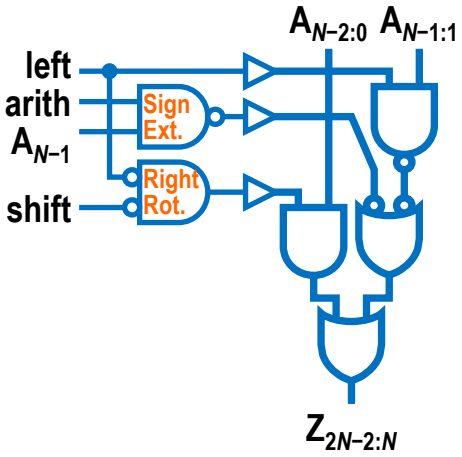
漏斗移位器

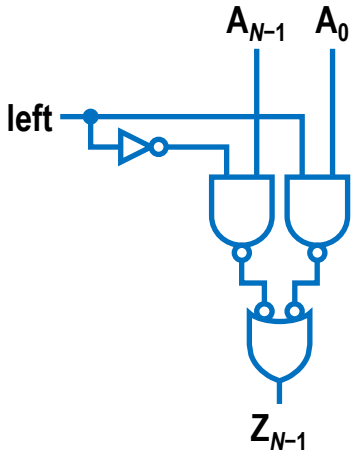
漏斗移位器源数据生成器

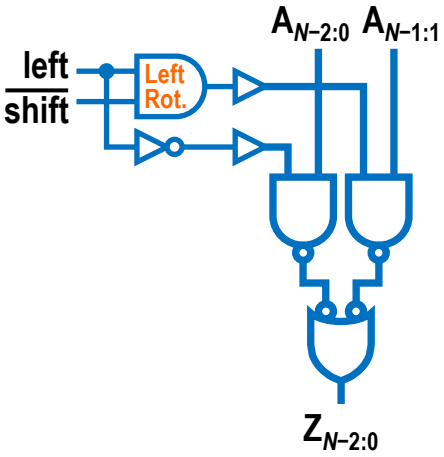
移位类型	$Z_{2N-2:N}$	Z_{N-1}	$Z_{N-2:0}$	Offset
循环右移	$A_{N-2:0}$	A_{N-1}	$A_{N-2:0}$	k
逻辑右移	0	A_{N-1}	$A_{N-2:0}$	k
算术右移	符号	A_{N-1}	$A_{N-2:0}$	k
循环左移	$A_{N-1:1}$	A_0	$A_{N-1:1}$	$N-1-k$
逻辑/算术左移	$A_{N-1:1}$	A_0	0	$N-1-k$



漏斗移位器的操作

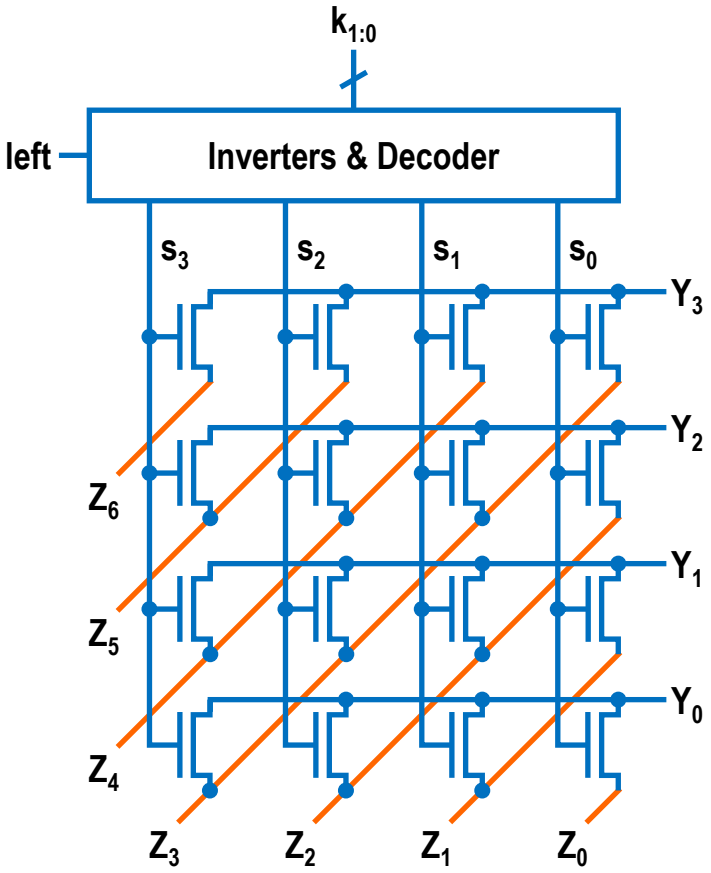




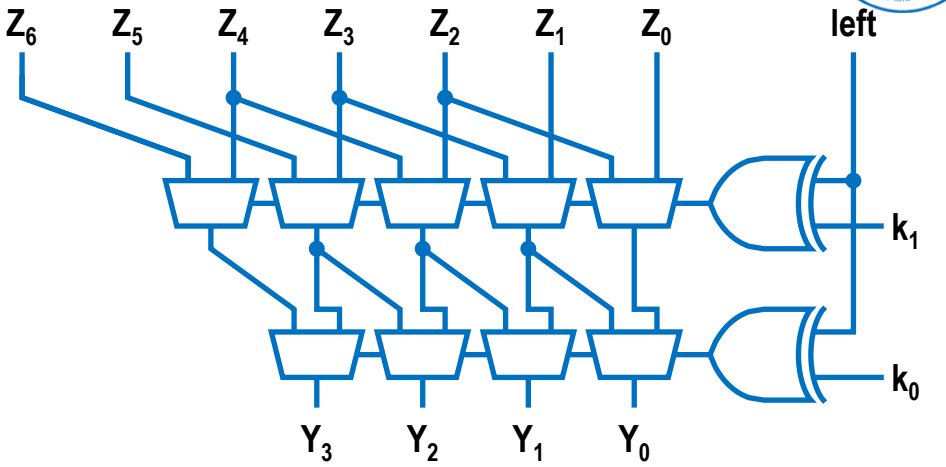


优化的源数据生成器逻辑

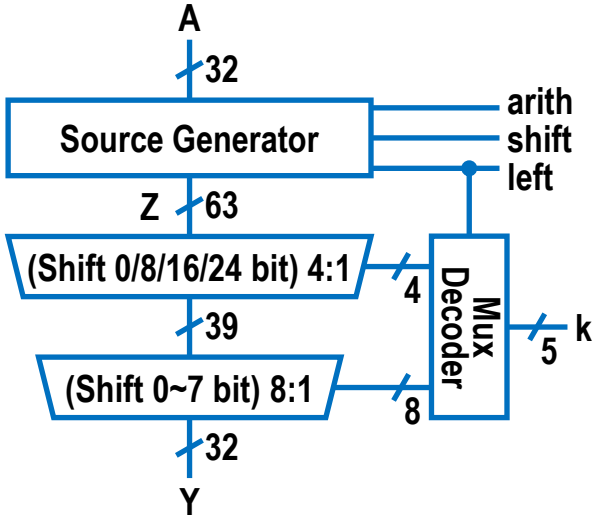
漏斗移位器



四位阵列漏斗移位器

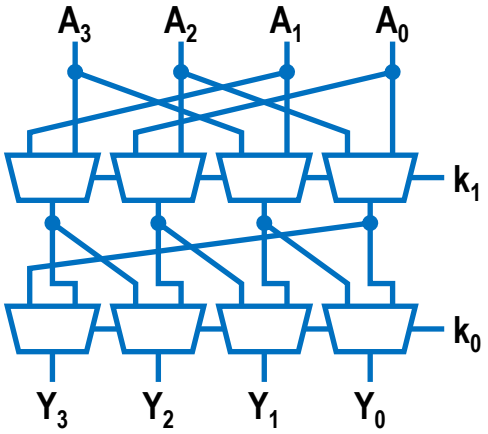


四位对数漏斗移位器

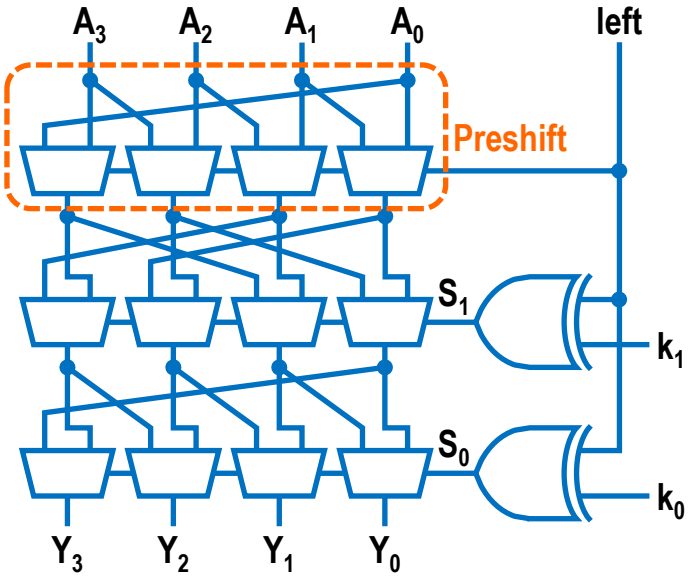


32位对数漏斗移位器

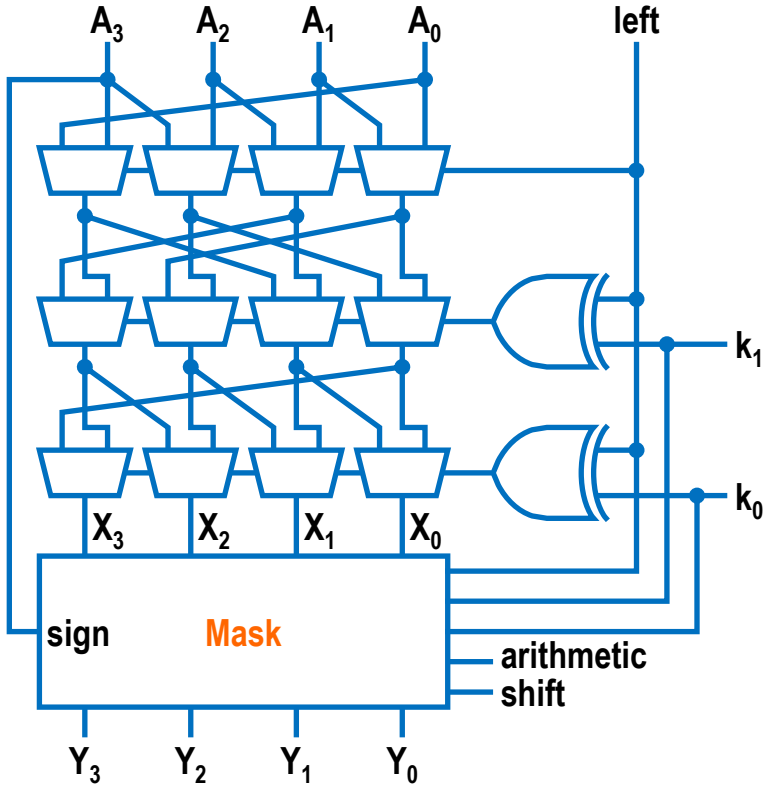
桶形移位器



循环右移桶形移位器

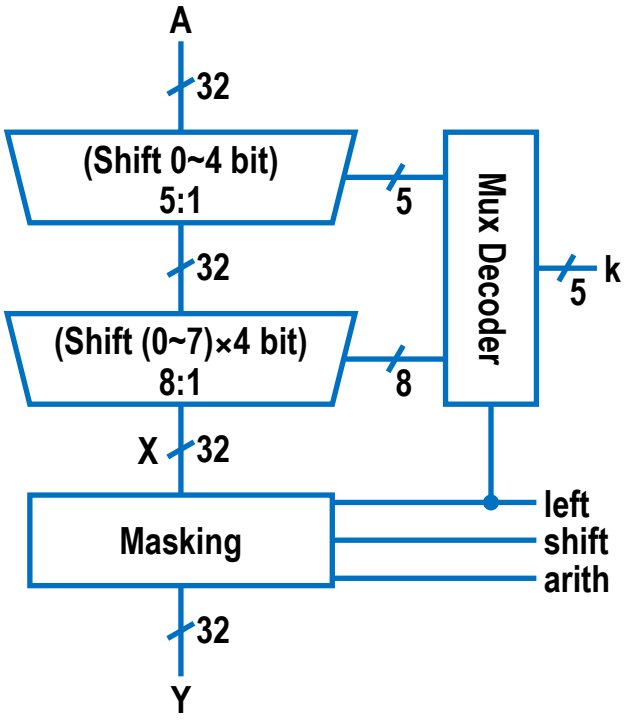


循环左移右移桶形移位器

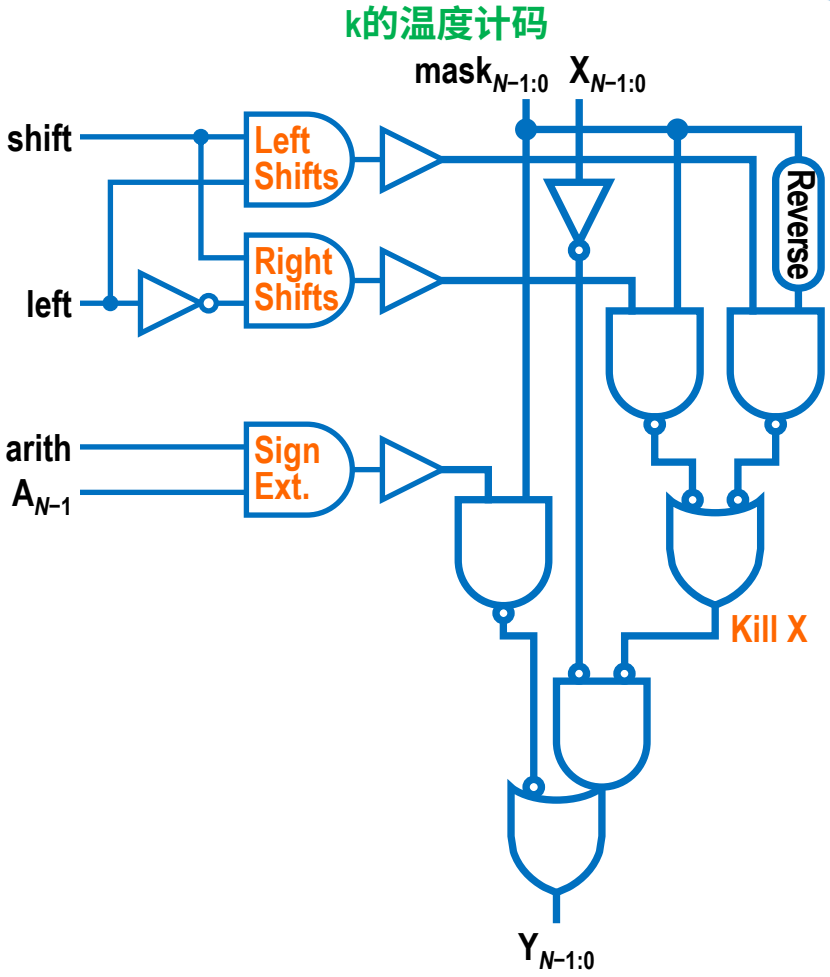


移位和循环桶形移位器

桶形移位器



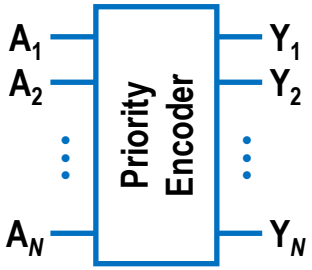
32位对数桶形移位器



桶形移位器的屏蔽逻辑

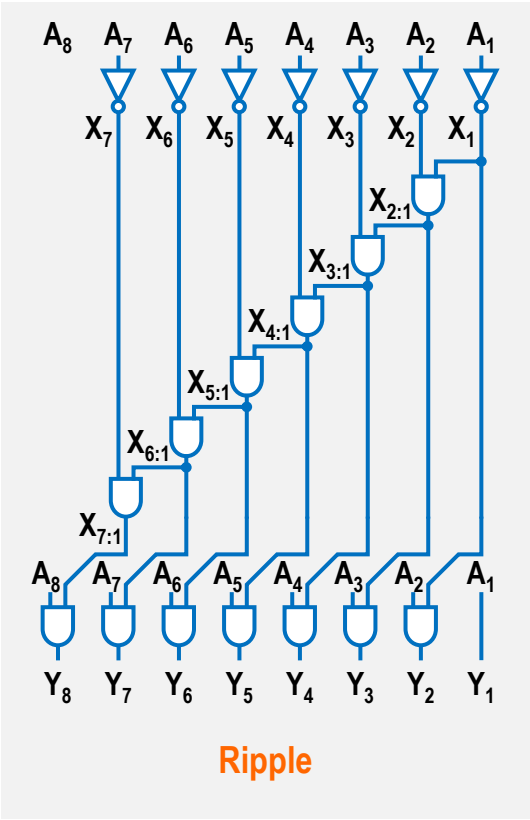
并行前缀运算

优先级编码器



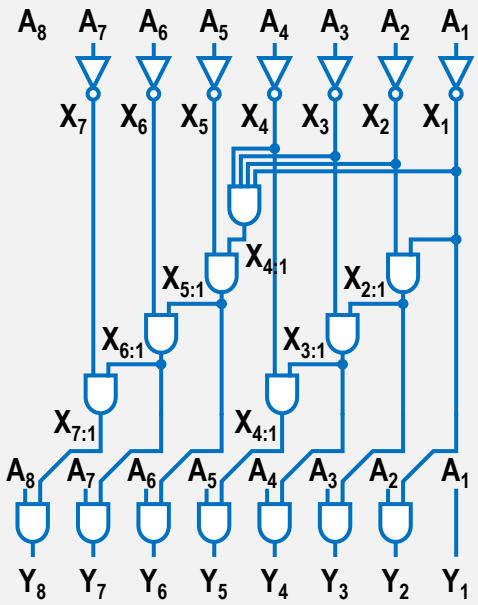
$$
\begin{cases}
Y_1 = A_1 \\
Y_2 = A_2 \cdot \overline{A_1} \\
Y_3 = A_3 \cdot \overline{A_2} \cdot \overline{A_1} \\
\vdots \\
Y_N = A_N \cdot \overline{A_{N-1}} \cdot \cdots \cdot \overline{A_1}
\end{cases}
$$

$$
\begin{cases}
X_{i:i} = \overline{A_i} & \text{Bitwise Precomputation} \\
X_{i:j} = X_{i:k} \cdot X_{k-1:j} & \text{Group Logic} \\
Y_i = A_i \cdot X_{i-1:1} & \text{Output Logic}
\end{cases}
$$

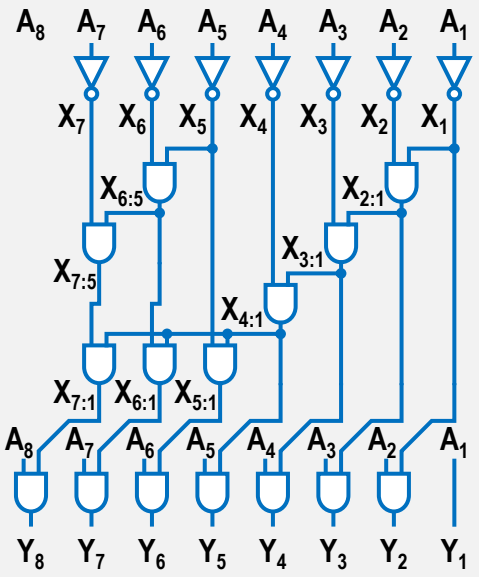


优先级编码器及电路实现

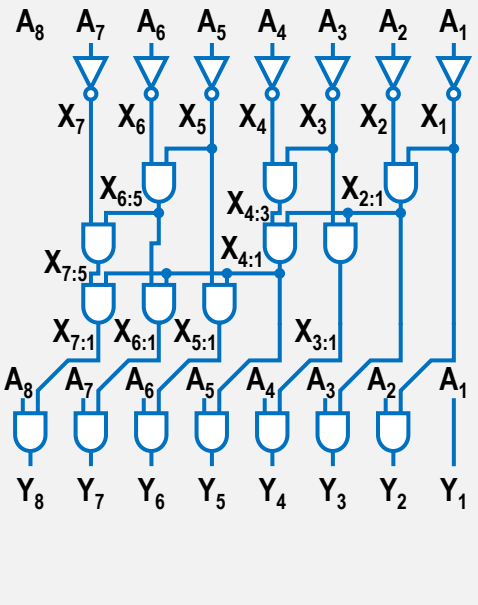
优先级编码器



Lookahead



Increment



Sklansky

$$
\begin{cases}
X_{i:i} = \overline{A_i} & \text{Bitwise Precomputation} \\
X_{i:j} = X_{i:k} \cdot X_{k-1:j} & \text{Group Logic} \\
Y_i = A_i \cdot X_{i-1:1} & \text{Output Logic}
\end{cases}
$$

优先级编码器电路实现

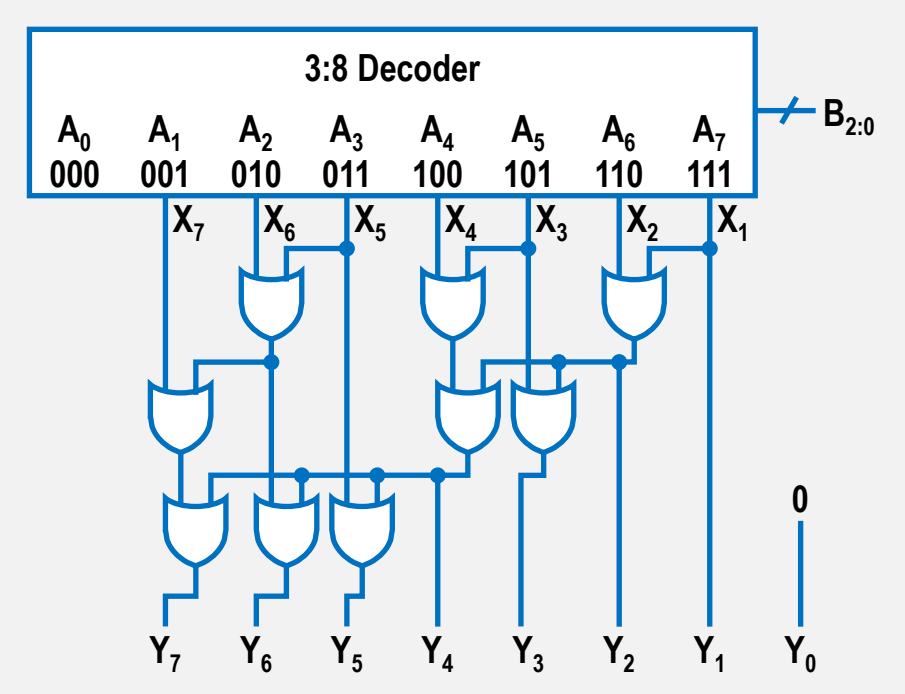
二进制码至温度计码译码器

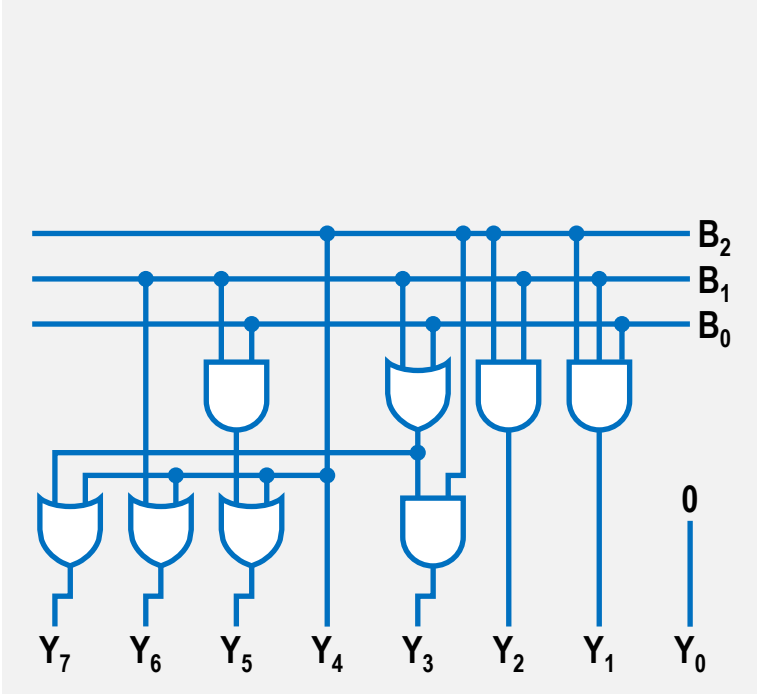
二进制码至温度计码译码真值表

B (Binary)	A (One-Hot)	Y (Thermometer)
000	00000001	00000000
001	00000010	10000000
010	00000100	11000000
011	00001000	11100000
100	00010000	11110000
101	00100000	11111000
110	01000000	11111100
111	10000000	11111110

$$
\begin{cases}
X_{i:i} = A_{N-i} & \text{Bitwise Precomputation} \\
X_{i:j} = X_{i:k} + X_{k-1:j} & \text{Group Logic} \\
Y_i = X_{i:0} & \text{Output Logic}
\end{cases}
$$

二进制码至温度计码译码器





$$
\begin{cases}
X_{i:i} = A_{N-i} & \text{Bitwise Precomputation} \\
X_{i:j} = X_{i:k} + X_{k-1:j} & \text{Group Logic} \\
Y_i = X_{i:0} & \text{Output Logic}
\end{cases}
$$

二进制码至温度计码译码器电路实现

乘　法

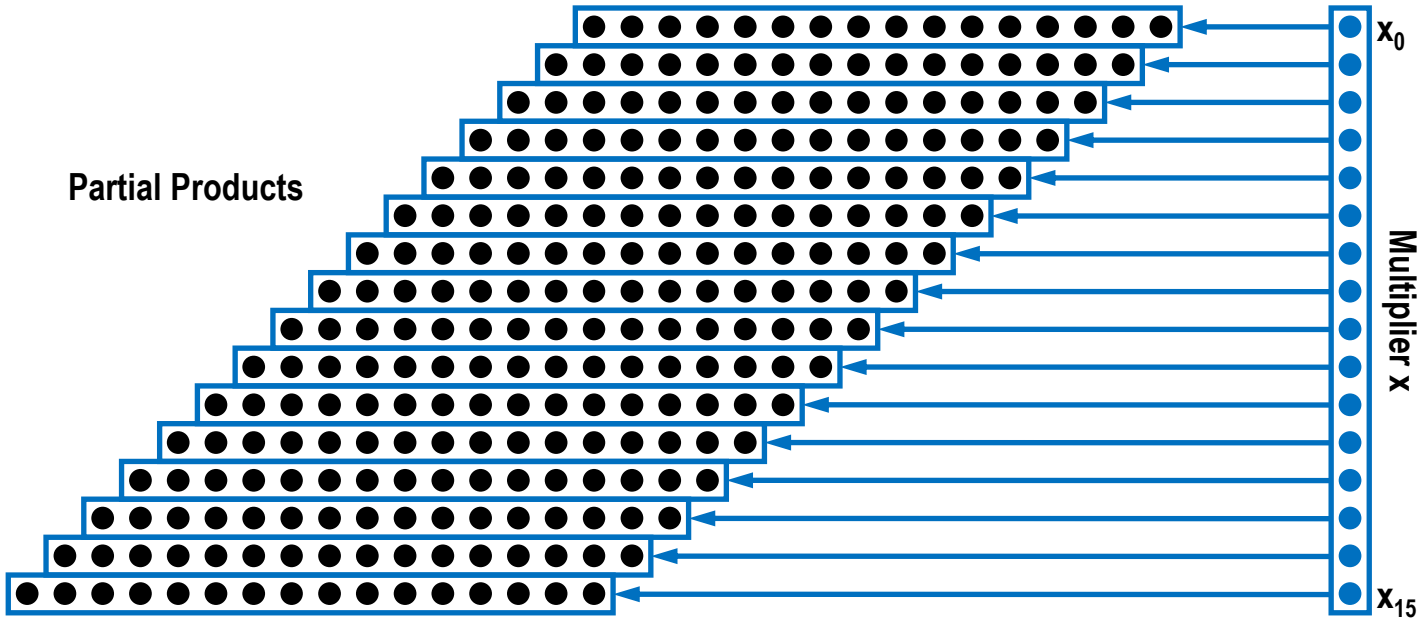
无符号数乘法

$$P = \left(\sum_{j=0}^{M-1} y_j 2^j\right)\left(\sum_{i=0}^{N-1} x_i 2^i\right) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} x_i y_j 2^{i+j}$$

						y_5	y_4	y_3	y_2	y_1	y_0	Multiplicand
						x_5	x_4	x_3	x_2	x_1	x_0	Multiplier
						x_0y_5	x_0y_4	x_0y_3	x_0y_2	x_0y_1	x_0y_0	
					x_1y_5	x_1y_4	x_1y_3	x_1y_2	x_1y_1	x_1y_0		
				x_2y_5	x_2y_4	x_2y_3	x_2y_2	x_2y_1	x_2y_0			Partial Products
			x_3y_5	x_3y_4	x_3y_3	x_3y_2	x_3y_1	x_3y_0				
		x_4y_5	x_4y_4	x_4y_3	x_4y_2	x_4y_1	x_4y_0					
	x_5y_5	x_5y_4	x_5y_3	x_5y_2	x_5y_1	x_5y_0						
p_{11}	p_{10}	p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0	Product

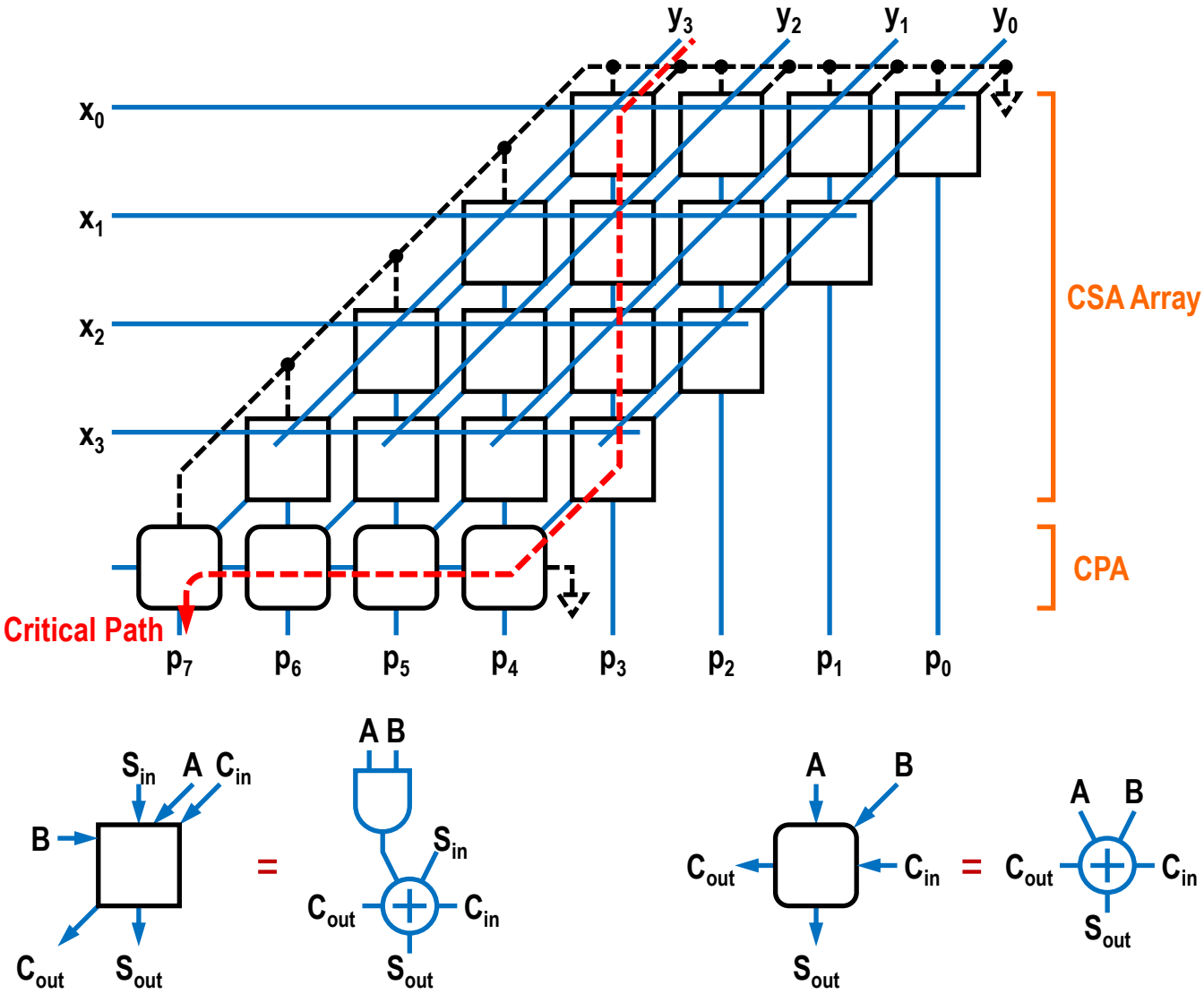
无符号数乘法和部分积

乘法的点图



点图 (Dot Diagram)

无符号阵列乘法



阵列乘法器

补码阵列乘法

$$P=\left(-y_{M-1}2^{M-1}+\sum_{j=0}^{M-2}y_j2^j\right)\left(-x_{N-1}2^{N-1}+\sum_{i=0}^{N-2}x_i2^i\right)$$

$$=\sum_{i=0}^{N-2}\sum_{j=0}^{M-2}x_iy_j2^{i+j}+x_{N-1}y_{M-1}2^{M+N-2}-\left(\sum_{i=0}^{N-2}x_iy_{M-1}2^{i+M-1}+\sum_{j=0}^{M-2}x_{N-1}y_j2^{j+N-1}\right)$$

$\sum_{i=0}^{N-2}\sum_{j=0}^{M-2}x_iy_j2^{i+j}$

$+\ x_{N-1}y_{M-1}2^{M+N-2}$

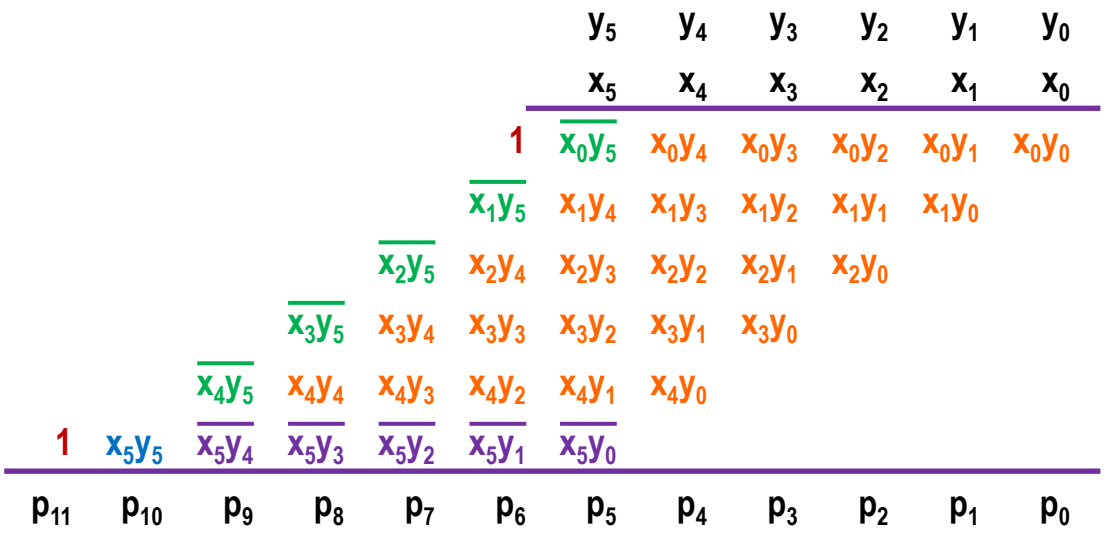
$-\sum_{i=0}^{N-2}x_iy_{M-1}2^{i+M-1}$

$-\sum_{j=0}^{M-2}x_{N-1}y_j2^{j+N-1}$

						y_5	y_4	y_3	y_2	y_1	y_0
						x_5	x_4	x_3	x_2	x_1	x_0
							x_0y_4	x_0y_3	x_0y_2	x_0y_1	x_0y_0
						x_1y_4	x_1y_3	x_1y_2	x_1y_1	x_1y_0	
					x_2y_4	x_2y_3	x_2y_2	x_2y_1	x_2y_0		
				x_3y_4	x_3y_3	x_3y_2	x_3y_1	x_3y_0			
			x_4y_4	x_4y_3	x_4y_2	x_4y_1	x_4y_0				
	x_5y_5										
1	1	$\overline{x_4y_5}$	$\overline{x_3y_5}$	$\overline{x_2y_5}$	$\overline{x_1y_5}$	$\overline{x_0y_5}$	1	1	1	1	1
											1
1	1	$\overline{x_5y_4}$	$\overline{x_5y_3}$	$\overline{x_5y_2}$	$\overline{x_5y_1}$	$\overline{x_5y_0}$	1	1	1	1	1
											1
p_{11}	p_{10}	p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

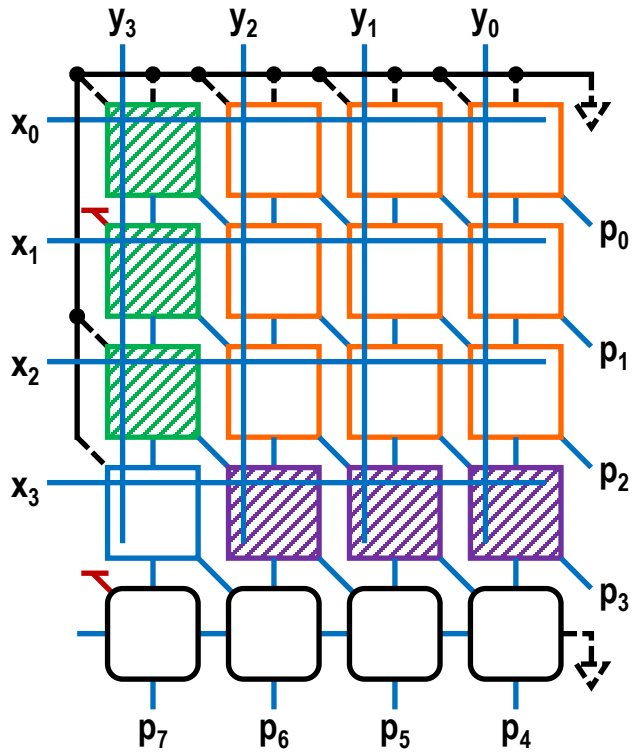
补码阵列乘法器及部分积

改进型Baugh-Wooley乘法器



补码乘法器简化后的部分积

(6 bit × 6 bit)



改进型Baugh-Wooley补码乘法器

(4 bit × 4 bit)

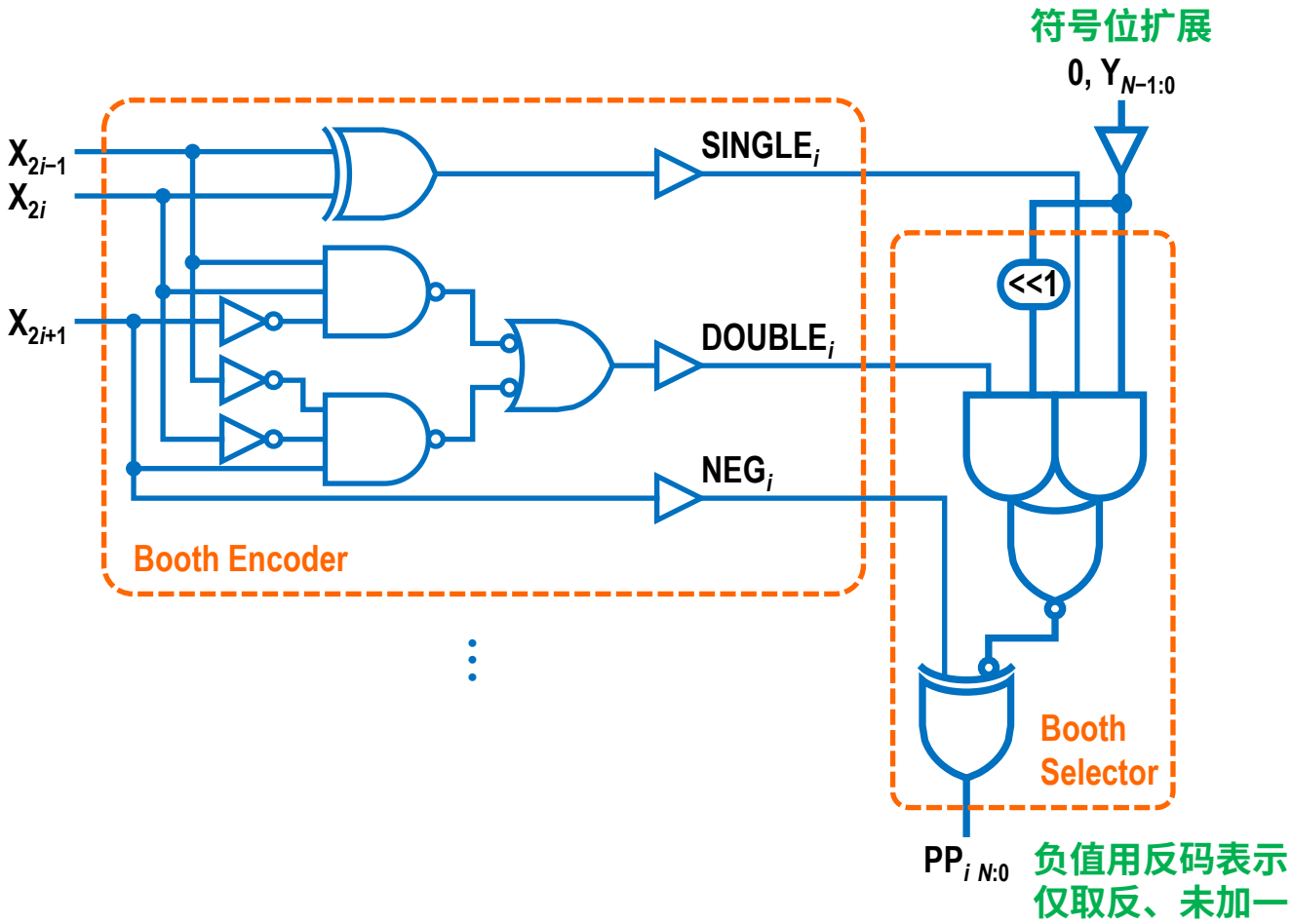
Booth编码

基4改进Booth编码值

Inputs			Partial Product	Booth Selects		
x_{2i+1}	x_{2i}	x_{2i-1}	PP_i	$SINGLE_i$	$DOUBLE_i$	NEG_i
0	0	0	0	0	0	0
0	0	1	Y	1	0	0
0	1	0	Y	1	0	0
0	1	1	2Y	0	1	0
1	0	0	−2Y	0	1	1
1	0	1	−Y	1	0	1
1	1	0	−Y	1	0	1
1	1	1	−0	0	0	1

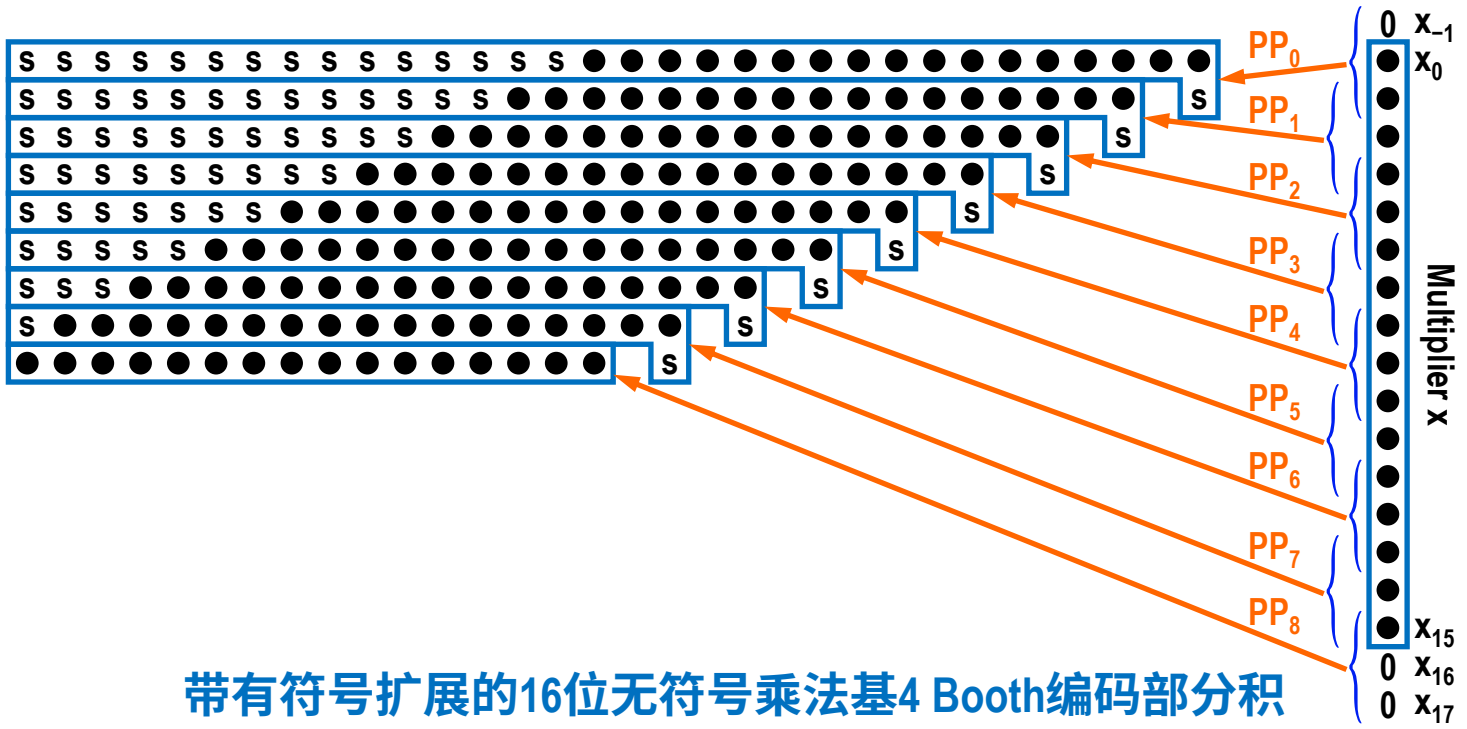
$$X = \sum_{i=0}^{n/2} \underbrace{(x_{2i-1} + x_{2i} - 2x_{2i+1})}_{\{-2,-1,0,+1,+2\}} 2^{2i}; \quad x_{-1} = 0$$

Booth编码器

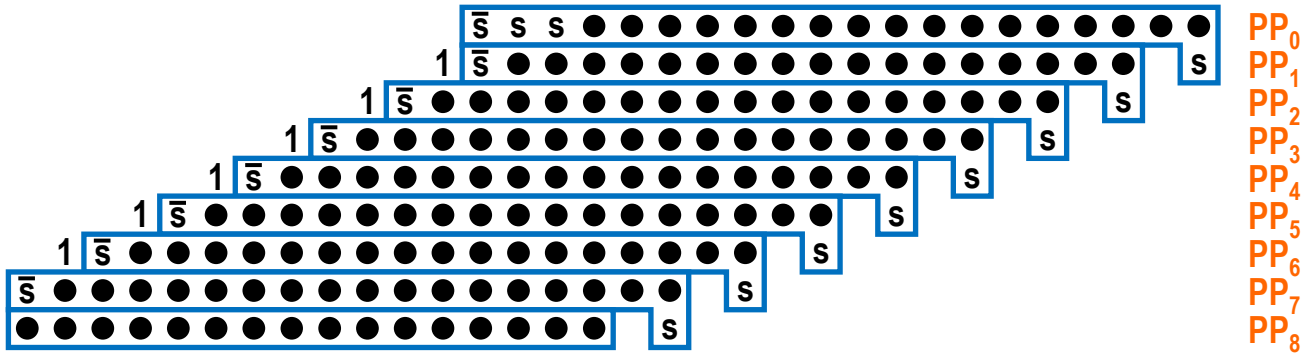


基4 Booth编码器和选择器

Booth编码部分积

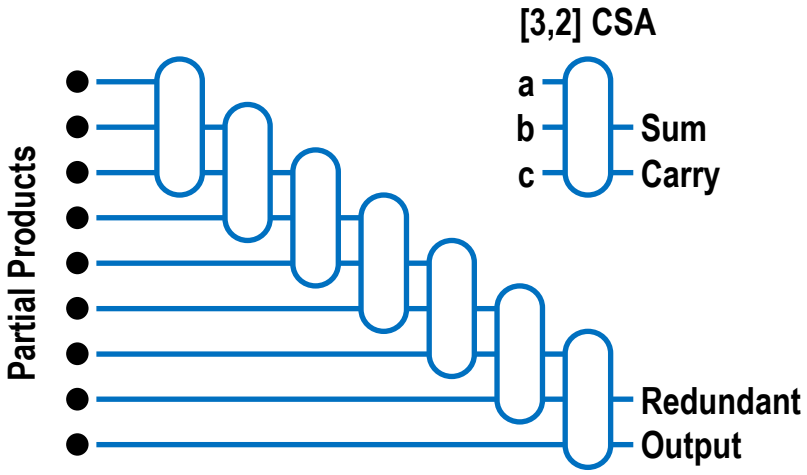


带有符号扩展的16位无符号乘法基4 Booth编码部分积

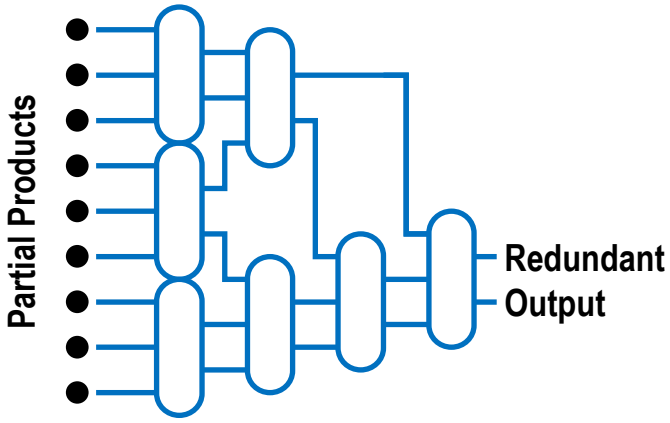


简化符号扩展的16位无符号乘法基4 Booth编码部分积

列加法



阵列乘法器点图

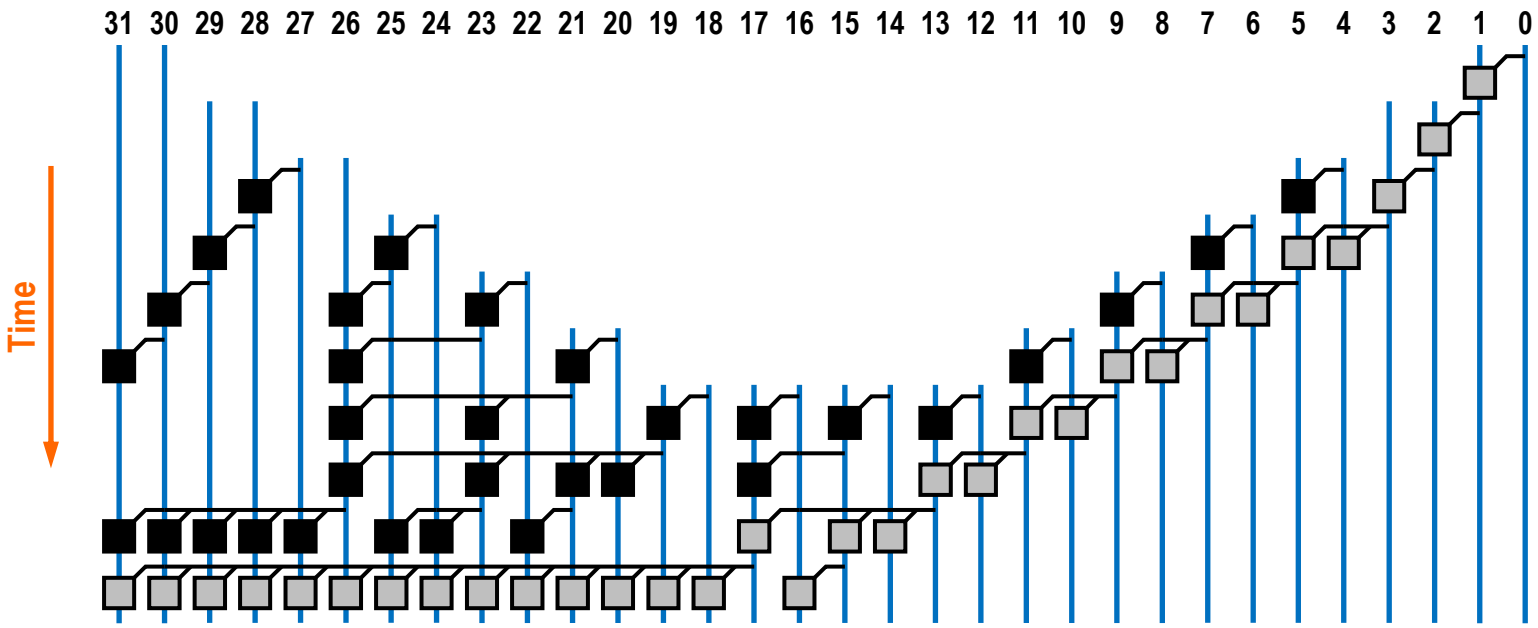


Wallace树乘法器点图

N个输入要求的[3,2] CSA级数为

$$\left\lceil \log_{3/2}\left(N/2\right)\right\rceil$$

最终加法



利用输入到达时间不一致的CPA前缀网络

本章结束