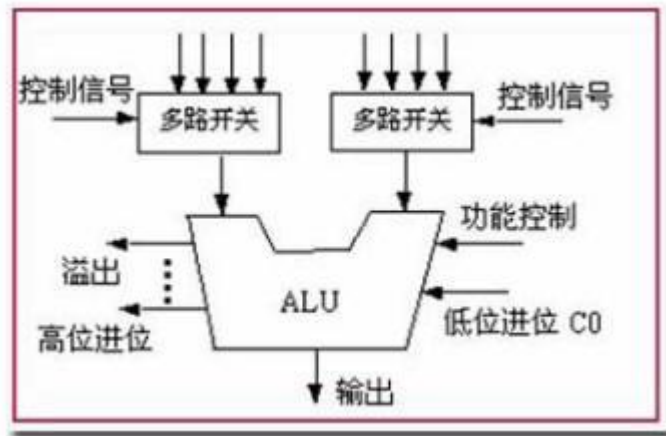


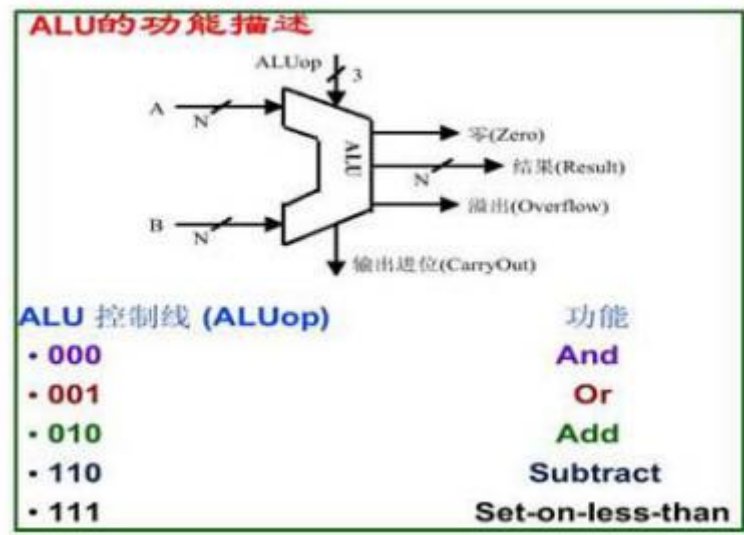
第八节 算术逻辑单元 (ALU)

ALU (Arithmetic Logic Unit)：用来执行各种算术和逻辑运算。

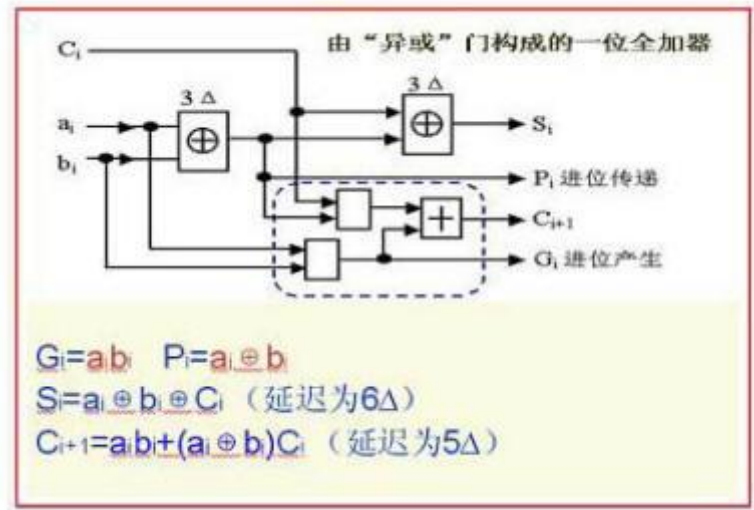
ALU 的核心是加法器，以下围绕加法器介绍



ALU 功能描述



全加器逻辑图



行波进位 ALU

一个 n 位 ALU 可以由 n 个一位 ALU 串行构成。这种 ALU 称为行波进位 ALU (Ripple-Carry ALU)。

全加逻辑方程: ($i=0, 1, \dots, n$)

$$S_i = a_i \oplus b_i \oplus C_i$$

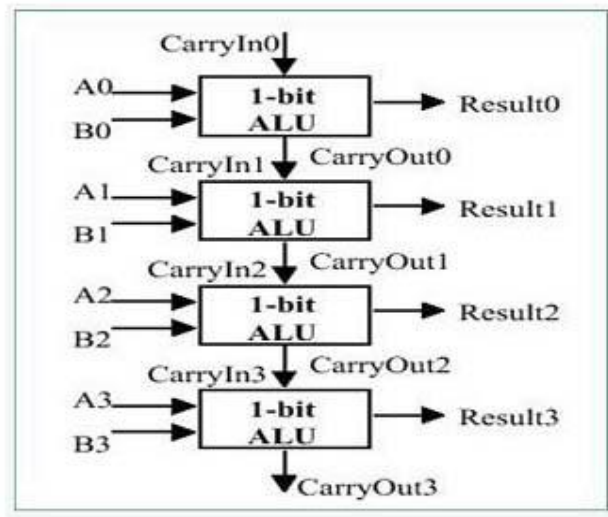
$$C_{i+1} = a_i b_i + (a_i \oplus b_i) C_i \quad (\text{延迟为 } 5\Delta)$$

上述进位逻辑与下列逻辑等价。

$$C_{i+1} = a_i b_i + (a_i + b_i) C_i \quad (\text{延迟为 } 3\Delta)$$

下面是一个 4 位行波进位 ALU。

4 位行波进位 ALU



和的产生延迟与 n 成正比。

先行进位 ALU

(1) 为什么用先行进位方式？

行波进位是串行逐级传递的，整个和的生成受到行波进位的影响。因此，现代计算机采用一种先行进位 (Carry look ahead) 方式。

(2) 如何产生先行进位？

定义两个辅助函数： $G_i = a_i b_i \cdots$ 进位生成 $P_i = a_i \oplus b_i \cdots$ 进位传递

通常把实现上述逻辑的电路称为进位生成/传递部件

全加逻辑方程： $S_i = P_i \oplus C_i$ $C_{i+1} = G_i + P_i C_i$ ($i=0, 1, \dots, n$) 设 $n=4$, 则:

$$C_1 = G_0 + P_0 C_0$$

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

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

$$C_4 = G_3 + P_3C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$$

由上式可知:各进位之间无等待,相互独立并同时产生。

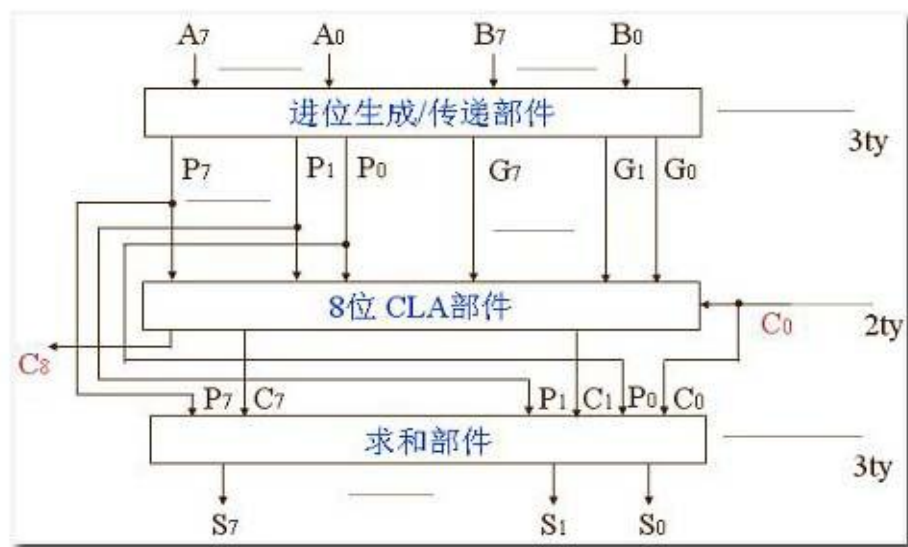
通常把实现上述逻辑的电路称为 4 位 CLA 部件

由此,根据 $S_i = P_i \oplus C_i$, 可并行求出各位和。

通常把实现 $S_i = P_i \oplus C_i$ 的电路称为求和部件

CLA 加法器由“进位生成/传递部件”、“CLA 部件”和“求和部件”构成。

8 位全先行进位加法器



和的总延迟: $3+2+3=8ty$; 进位 C_8 的延迟: $3+2=5ty$

局部先行进位加法器

局部先行进位加法器 (Partial Carry Lookahead Adder)

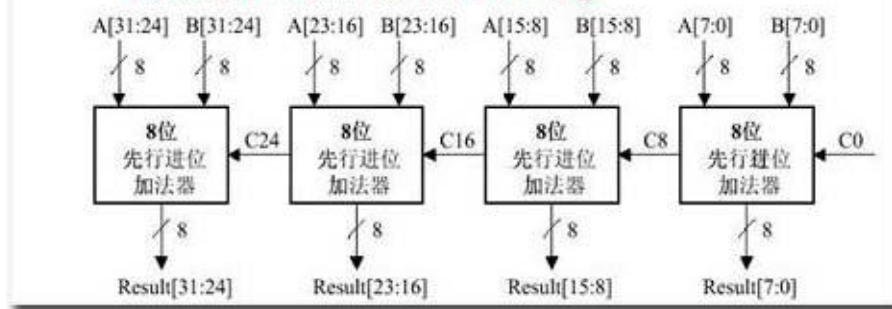
“实现全先行进位加法器的成本太高

- 想象 **Cin31** 的逻辑方程的长度

“一般性经验:

- 连接一些 **N** 位先行进位加法器，形成一个大加法器
- 例如: 连接 **4** 个 **8** 位进位先行加法器，形成 **1** 个 **32** 位局部先行进位加法器

所有和数产生的延迟为: $5+2+2+5=14t_y$



多级先行进位加法器

(3) 多级先行进位加法器

单级(局部)先行进位加法器的进位生成方式:

“组内并行、组间串行”

所以，单级先行进位加法器虽然比行波加法器延迟时间短，但高位组进位依赖低位组进位，故仍有较长的时间延迟。

通过引入组进位生成/传递函数来实现“组内并行、组间也并行”的进位生成方式。 设 $n=4$, 则:

$$C1=G0+P0C0$$

$$C2=G1+P1C1=G1+P1G0+P1P0C0$$

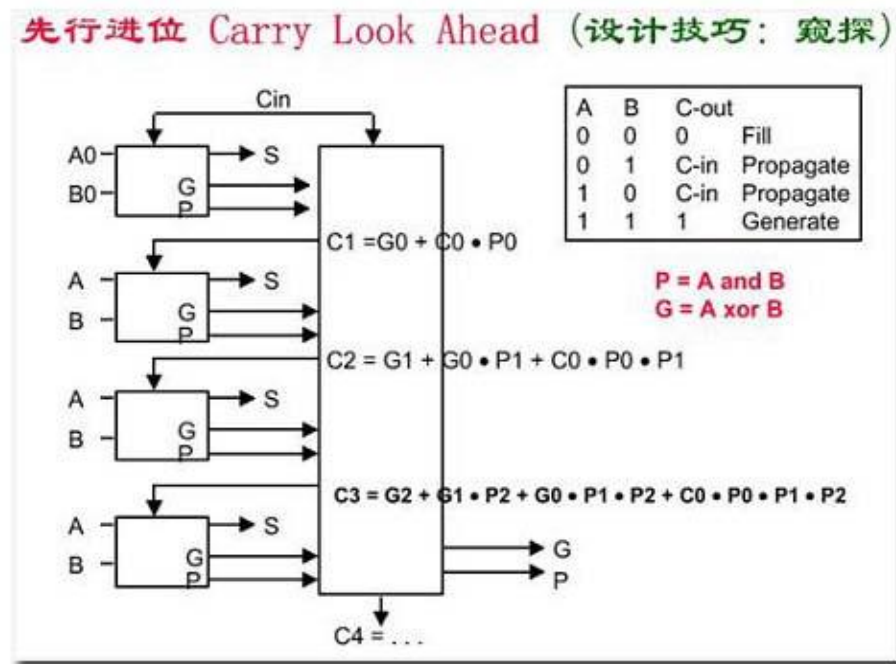
$$C3=G2+P2C2=G2+P2G1+P2P1G0+P2P1P0C0$$

$$G3^*=G3+P3C3=G3+P3G2+P3P2G1+P3P2P1G0$$

$$P3^* = P3P2P1P0$$

把实现上述逻辑的电路称为 4 位 BCLA 部件。

两级先行进位加法器



先行进位 ALU

(4) 快速先行进位 ALU

SN74181 是早期国际流行的四位 ALU 芯片，是中规模集成电路。它在原有先行进位加法器的基础上再附加部分线路，实现了按位逻辑运算，因此具有基本的算术运算和逻辑运算功能。

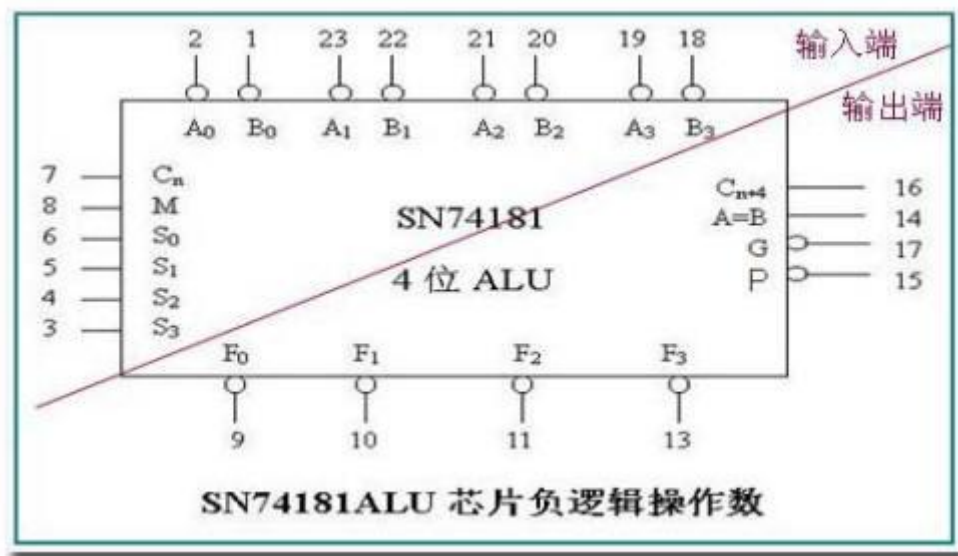
SN74182 是 4 位 BCLA (成组先行进位) 芯片。

用 4 个 SN74181 芯片可直接构成 16 位单级先行进位 ALU

用 4 个 SN74181 芯片与 1 个 SN74182 芯片可构成 16 位快速两级先行进位 ALU

用 16 个 SN74181 芯片与 5 个 SN74182 芯片可构成 64 位快速三级先行进位 ALU

一个典型的 4 位中规模 ALU 芯片-SN74181



输入端： A_i 和 B_i 分别为第一和第二操作数， C_n 为低位进位， M 为功能选择线， S_i 为操作选择线。

输出端： F_i 为运算结果， C_{n+4} 、 P 和 G 为进位，“ $A=B$ ”为相等标志

SN74181 正逻辑功能表

S ₃	S ₂	S ₁	S ₀	M=H 逻辑运算	M=L 算术运算	
					$\overline{C_n}=1$	$\overline{C_n}=0$
L	L	L	L	\overline{A}	A	A+1
L	L	L	H	$\overline{A+B}$	A+B	(A+B)加1
L	L	H	L	$\overline{A \cdot B}$	A+B	(A+B)加1
L	L	H	H	"0"	减1	"0"
L	H	L	L	$\overline{A \cdot \overline{B}}$	A加(A· \overline{B})	A加(A· \overline{B})加1
L	H	L	H	\overline{B}	(A· \overline{B})加(A+B)	(A· \overline{B})加(A+B)加1
L	H	H	L	$A \oplus B$	A减B减1	A减B
L	H	H	H	$A \cdot \overline{B}$	(A· \overline{B})减1	A· \overline{B}
H	L	L	L	$\overline{A+B}$	A加(A·B)	A加(A·B)加1
H	L	L	H	$\overline{A \oplus B}$	A加B	A加B加1
H	L	H	L	B	(A·B)加(A+B)	(A·B)加(A+B)加1
H	L	H	H	$A \cdot B$	(A·B)减1	A·B
H	H	L	L	"1"	A加A	A加A加1
H	H	L	H	$A \cdot \overline{B}$	A加(A+B)	A加(A+B)加1
H	H	H	L	A+B	A加(A+B)	A加(A+B)加1
H	H	H	H	A	A减1	A

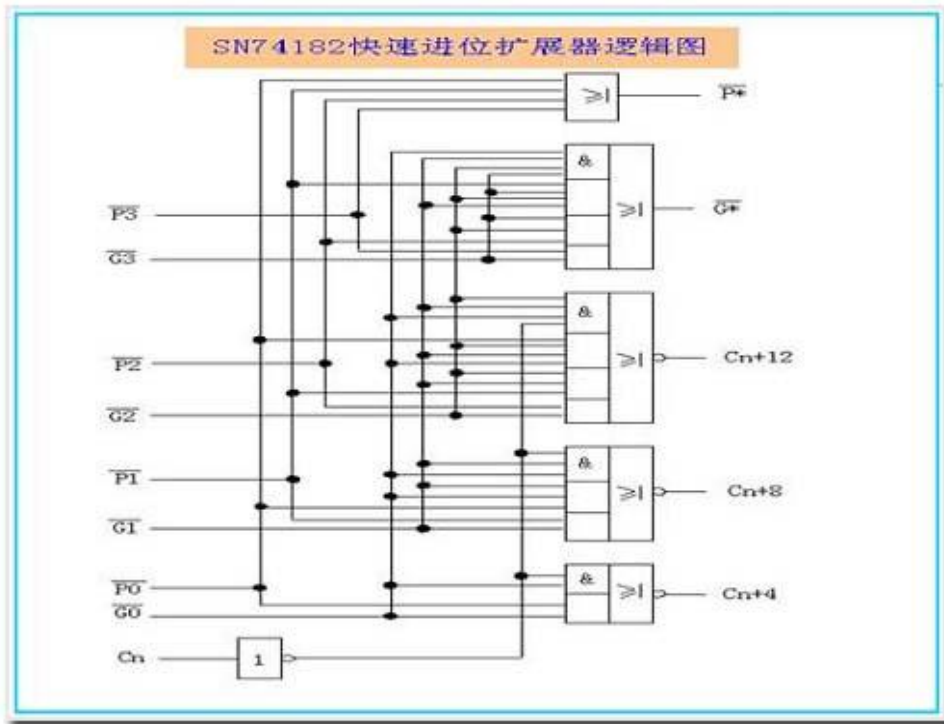
SN74182 芯片的引脚



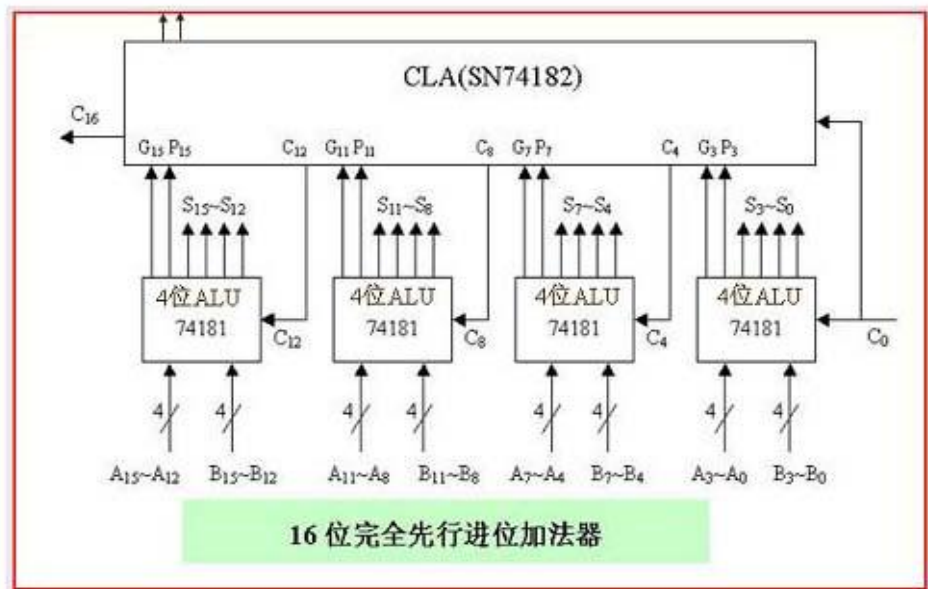
输入端：P_i 和 G_i 分别为第 i 组的组内进位传递函数和进位生成函数，C_n 为低位进位。

输出端：C_{n+4}、C_{n+8}、C_{n+12} 为相应组的组内进位，P^{*}和 G^{*}分别为整个大组的组进位传递函数和进位生成函数。

SN74182 芯片的逻辑电路图



SN74181 和 SN74182 组成 16 位先行进位加法器



数据通路(运算器)

数据通路（运算器）的概念

运算器的基本功能是进行数据运算。ALU 是其核心部件。数据通路是指计算机的数据信息从一个部件传输到另一个部件所经过的路径，连同路径上的设备。如：寄存器、暂存器、多路选择器、移位器、加工部件等。

一个简单数据通路实例

一个四位数据通路芯片-AM2901A

现代计算机所用数据通路(流水线/超标量/...)

一个简单数据通路芯片-Am2901A

