第 2 章　运算器组成实验

2.1　可控加减法电路设计实验

2.1.1　实验目的

掌握 1 位全加器的实现逻辑，掌握多位可控加减法电路的实现逻辑，熟悉 Logisim 平台基本功能，能在 Logisim 中实现多位可控加减法电路。

2.1.2　背景知识

1. 1 位全加器

多位定点加法可由 1 位加法器通过进位链串联实现，所以实现多位加法器的基础部件是 1 位全加器，其输入为两个相加数 X_i、Y_i，低位进位输入 C_{i-1}，输出为和数 S_i，高位进位输出 C_i，对应逻辑表达式如下：

$$S_i = X_i \oplus Y_i \oplus C_{i-1}$$

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

具体逻辑电路如图 2.1 所示，假设所有逻辑门的延迟都是 T，则 1 位加法器的总时间延迟为 $3T$。

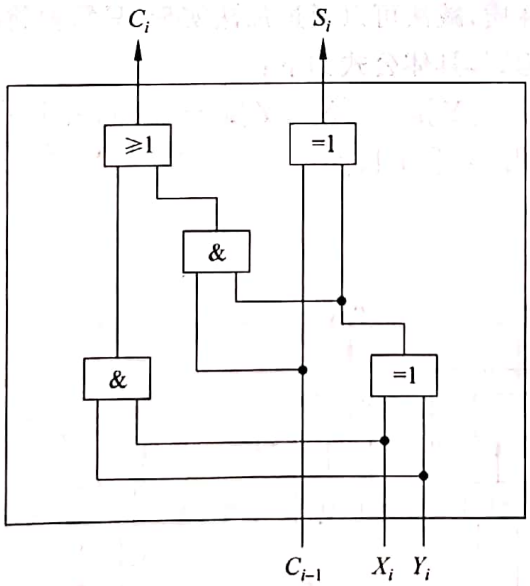

图 2.1　一位全加器逻辑框图

2. 多位串行加法器

将 n 个 1 位全加器的进位链串联即可得到 n 位加法器，如图 2.2 所示，由于补码符号位也可以参与运算，所以此电路既可以用于有符号数的运算，也可以用于无符号数的运算，但

二者在溢出检测上是有区别的。以有符号补码加法为例，当两个正数相加结果为负数时，发生正溢出；当两个负数相加结果为正数时，发生负溢出。根据补码符号位定义，假设相加数符号位分别为 f_0，f_1，和数符号位为 f_s，则溢出检测信号 $Overflow$ 逻辑表达式为：

$$Overflow = \bar{f}_0\,\bar{f}_1 f_s + f_0 f_1\,\bar{f}_s$$

另外如果已知符号位进位为 C_F 和最高位数值位进位为 C_D，也可以利用如下逻辑表达式进行溢出检测：

$$Overflow = C_F \oplus C_D$$

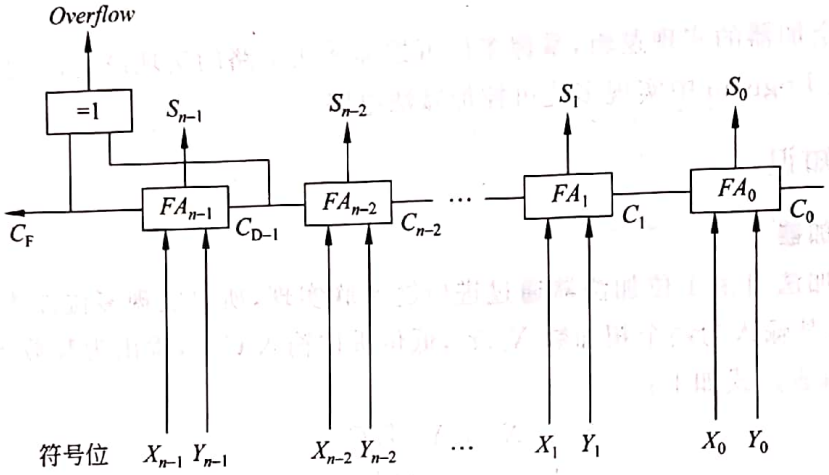

图 2.2　多位串行加法器逻辑框图

3. 可控加减法电路

由于补码运算的特殊性质，减法可以通过加法实现，只需要将减数 Y 的补码再次求补后送入加法器即可实现减法运算，具体公式如下：

$$[X]_{补} - [Y]_{补} = [X-Y]_{补} = [X]_{补} + [-Y]_{补}$$

$$[-Y]_{补} = [[Y]_{补}]_{补}$$

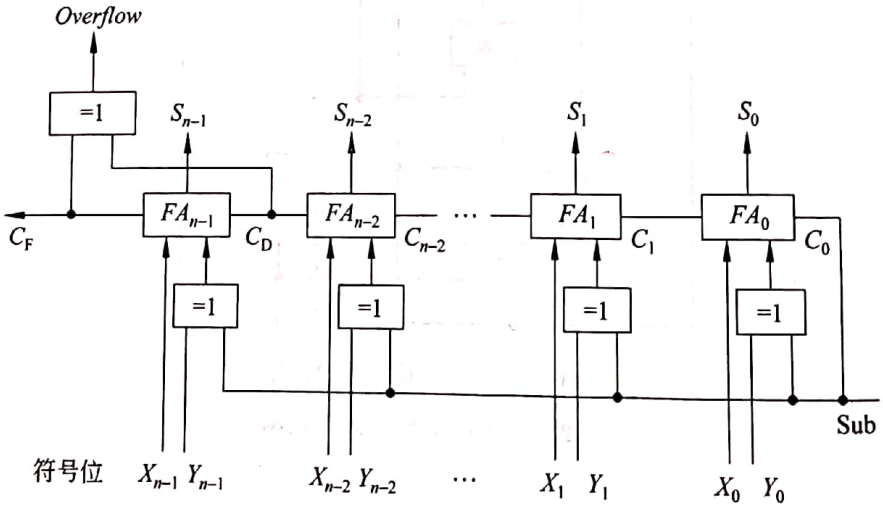

图 2.3　多位可控加减法电路逻辑框图

图 2.3 是多位可控加减法电路逻辑框图，该电路在串行加法器中引入 Sub 控制信号，输

入数 Y 的所有位 Y_i 均与 Sub 信号进行异或后送入多位串行加法器，当 Sub=0 时，送入加法器的是 Y 本身，当 Sub=1 时，送入加法器的是 Y 的反码，另外 Sub 连接到加法器的最低位的进位输入，实现了对 Y 操作数的逐位取反，末位加 1 的求补过程，从而完成减法操作，当 Sub=0 时，低位进位为零，不影响加法结果的正确性。对于减法的溢出检测，最直观的判断依据是正数减负数结果为负数，负数减正数结果为正数。

2.1.3　实验内容

在 Logisim 模拟器中打开 alu. circ 文件，在对应的子电路中利用已经封装好的全加器设计 8 位串行可控加减法电路，其电路引脚定义如图 2.4 所示，用户可以直接在电路中使用对应的隧道标签，其中 X 和 Y 为两输入数，Sub 为加减控制信号，S 为运算结果输出，C_{out} 为进位输出，$Overflow$ 为有符号运算溢出标志。

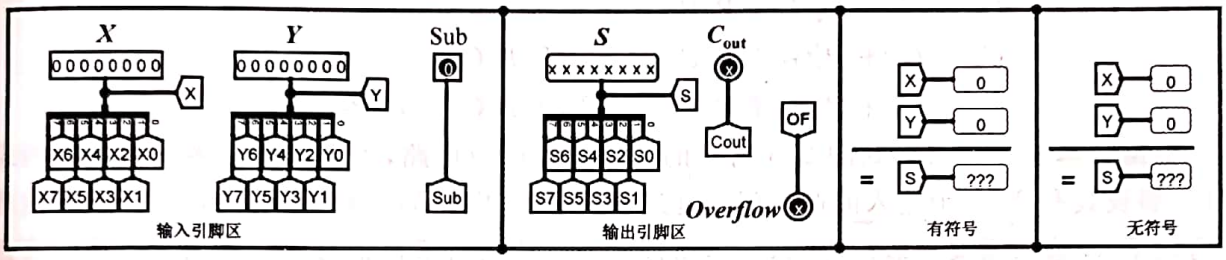

图 2.4　8 位串行可控加减法电路引脚定义

2.1.4　实验思考

(1) 假设所有门电路延迟均为 T，为什么 8 位串行可控加减法器的时间延迟是 $18T$？

(2) 在 CPU 中可控加减法电路中的 Sub 信号何时会产生？由谁产生？依据是什么？

2.2　4 位快速加法器设计实验

2.2.1　实验目的

掌握快速加法器中先行进位的原理，能利用相关知识设计 4 位先行进位电路，并利用设计的 4 位先行进位电路构造 4 位快速加法器，能分析对应电路的时间延迟。

2.2.2　背景知识

多位串行加法电路中和数与进位位的逻辑表达式如下：

$$S_i = X_i \oplus Y_i \oplus C_{i-1}$$

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

高位的运算依赖于低位运算的进位输入 C_{i-1}，所有全加器不能并行运行，多位加法器整体延迟与位数成线性关系，当位宽较大时性能较差。串行加法器性能较差的原因是由于进位链依赖，如果能打破这种依赖关系，提前得到所有全加器所需的进位信号，则所有全加器可以并行运算，从而提升加法运算性能。

假设进位生成函数 $G_i=X_iY_i$，进位传递函数 $P_i=X_i\oplus Y_i$，则以上公式变换成如下公式：

$$S_i = P_i \oplus C_{i-1}$$

$$C_i = G_i + P_iC_{i-1}$$

利用进位链公式进行重复迭代，可得如下公式：

$$C_n = G_n + P_nG_{n-1} + P_nP_{n-1}G_{n-2} + P_nP_{n-1}P_{n-2}G_{n-3}\cdots + P_nP_{n-1}P_{n-2}\cdots P_1C_0$$

上述公式表明，高位进位输出可以直接由已知的变量 G_i、P_i 和 C_0 经过电路运算得到，根据此公式可以利用额外电路计算所有进位信号，提前产生各位加法运算需要的所有进位输入，再利用 $S_i=P_i\oplus C_{i-1}$ 即可得到最终的和数，这就是先行进位的基本原理，注意这里进位信号的产生也是需要时间延迟的，n 越大则延迟越大，为了简化电路，通常按照 4 位一组进行先行进位，根据公式可得到如下逻辑表达式：

$$C_1 = G_1 + P_1C_0$$

$$C_2 = G_2 + P_2G_1 + P_2P_1C_0$$

$$C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1C_0$$

$$C_4 = G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1 + P_4P_3P_2P_1C_0$$

根据逻辑表达式可得到图 2.5 所示的 4 位先行进位电路，图中采用了多输入系数的逻辑门，假设具有 2～5 个输入的逻辑门的延迟相同，则该电路的逻辑总延迟为 $2T$。考虑生成所有的 P、G 输入需要一级门电路延迟，所以最终生成所有进位信号的延迟为 $3T$，当所有进位信号产生后，再增加一级异或门延迟即可生成所有位的和数，完成多位加法运算。因此，这里 4 位全加器的延迟为 $4T$，相比 4 位串行加法器有较大的性能提升，4 位快速加法器逻辑框图如图 2.6 所示。

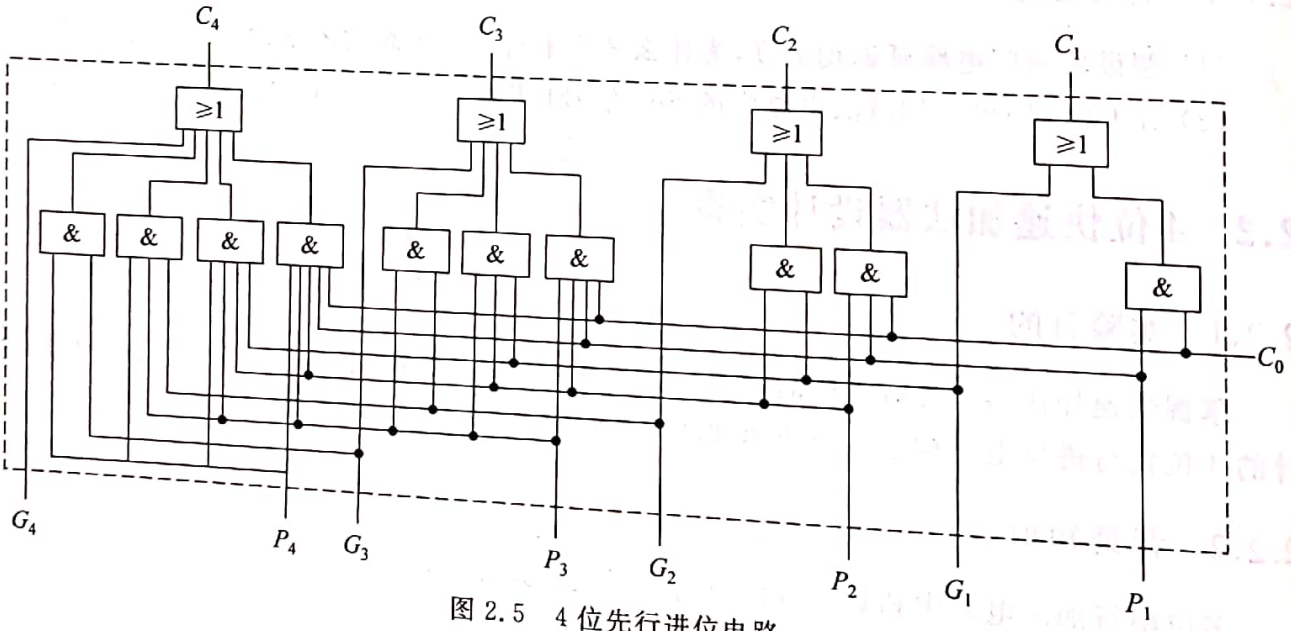

图 2.5　4 位先行进位电路

有了 4 位快速加法器，如何构建更多位宽的加法器电路，例如 16 位加法器，最简单的方法是将 4 个快速加法器进位链进行串联，但这种方法只能实现 4 位一组组内并行加法，组间仍然是串行运算，性能较差，如图 2.7 所示。

有没有方法进一步提升性能呢？4 位快速加法器进位输出与进位输入之间的关系如下：

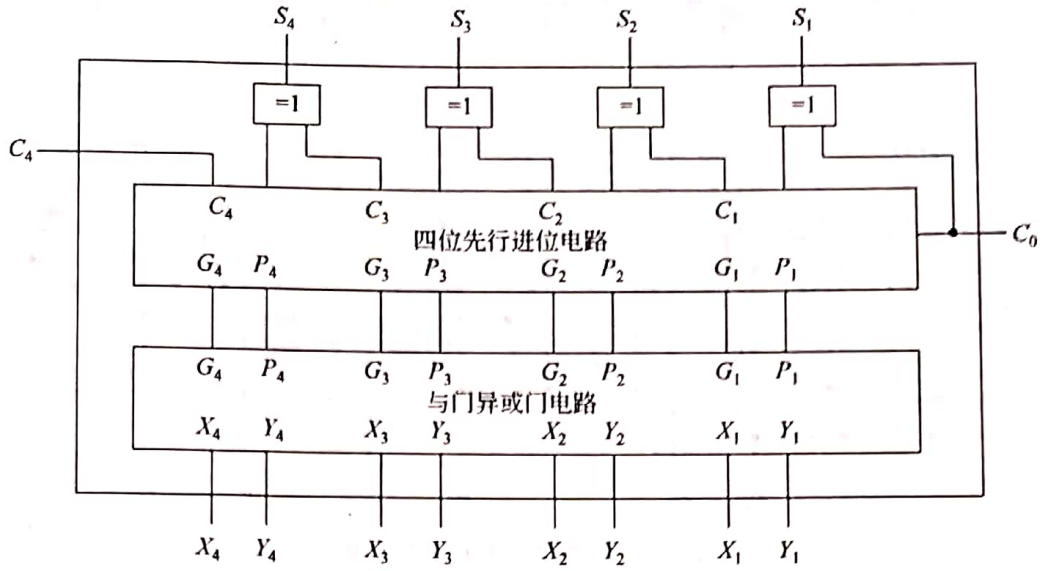

图 2.6　4 位快速加法器

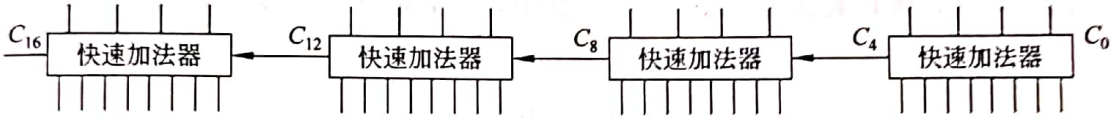

图 2.7　16 位组内并行、组间串行加法器

$$C_4 = G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1 + P_4P_3P_2P_1C_0$$

令 $G_4^* = G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1$，又称成组进位生成函数，令 $P_4^* = P_4P_3P_2P_1$，又称成组进位传递函数。公式变换如下

$$C_4 = G_4^* + P_4^* C_0$$

此公式与先行进位中的 $C_1 = G_1 + P_1C_0$ 公式形式完全相同，这也意味着只要提前生成了对应成组进位生成函数与成组进位传递函数，就可以复用 4 位先行进位电路，生成 4 位一组的组间先行进位信号。对 4 位先行进位电路进行改造，即可得到可以用于级联的先行进位电路，具体封装如图 2.8 所示，很多教科书上将该电路称为 CLA74182 电路，输入为 4 对 P、G 信号进位输入以及低位进位 C。输出为 4 个进位输出以及成组进位生成函数 P^* 和成组进位传递函数 G^*，具体电路实现如图 2.9 所示。

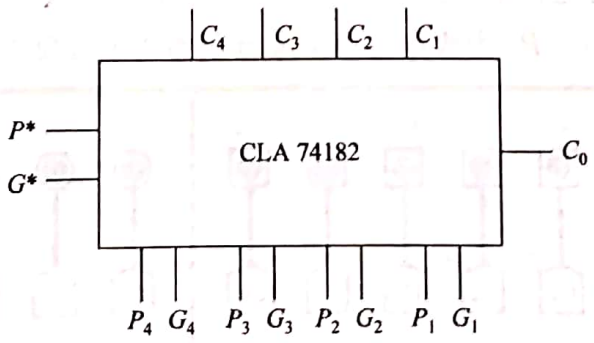

图 2.8　4 位先行进位电路封装

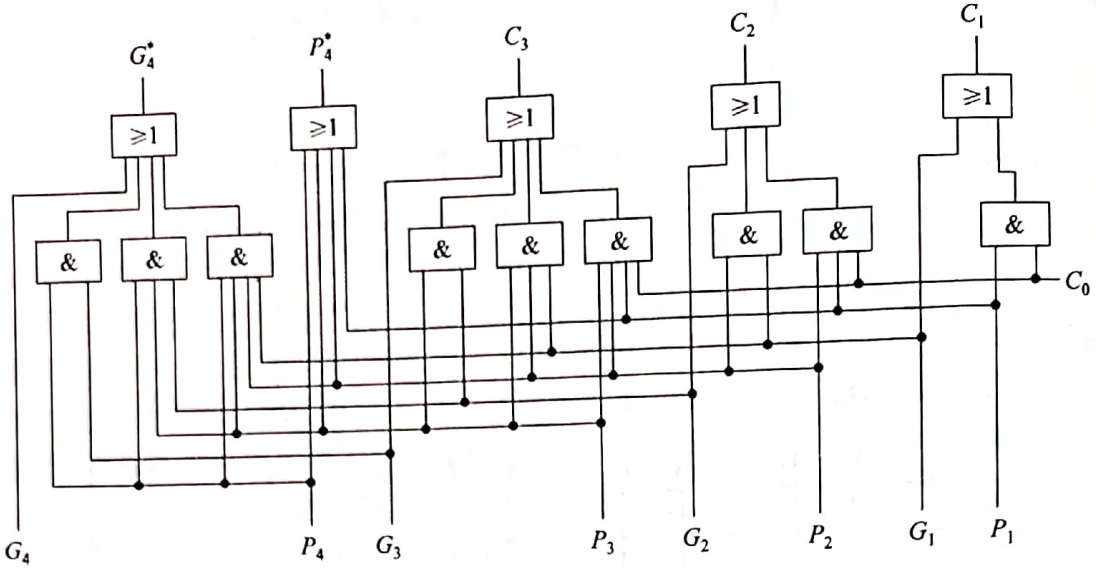

图 2.9　可级联的先行进位电路

另外，很多组成原理教科书上介绍了 SN74181 运算器芯片，该芯片内部核心电路是一个快速加法器，其具体封装如图 2.10 所示。其中，P、G 输出就是成组传递函数和成组生成函数。

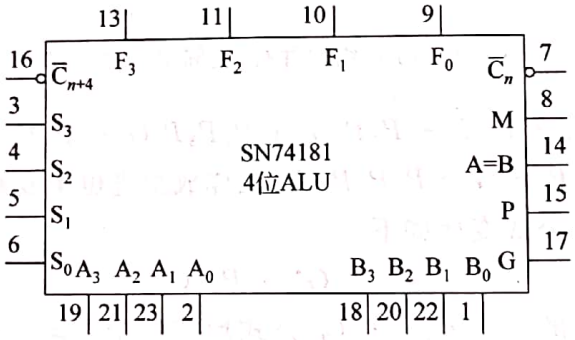

图 2.10　4 位快速加法器电路封装

2.2.3　实验内容

(1) 在 Logisim 中打开 alu.circ 文件，按照图 2.11 定义的输入输出引脚，在对应子电路中实现可级联的 4 位先行进位电路。其中 G_i、P_i 为进位生成函数和传递函数，C_{in} 为进位输入，$C_1 \sim C_4$ 为进位输出，G^*、P^* 为成组进位生成函数和成组进位传递函数。

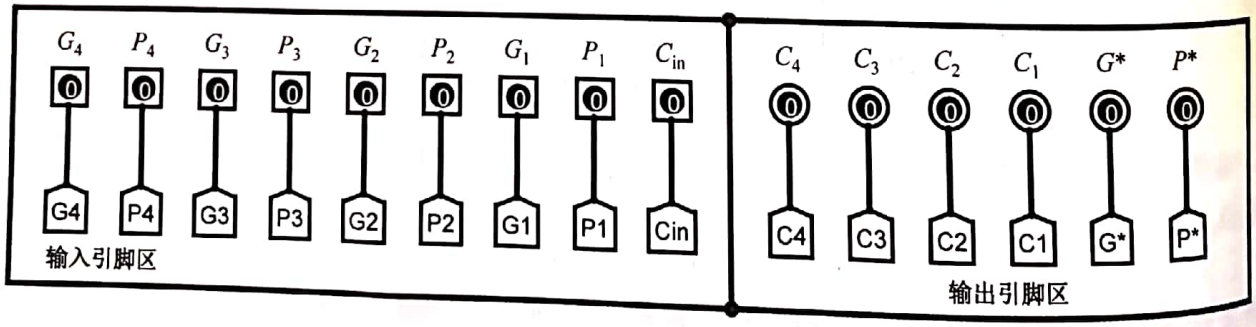

图 2.11　4 位先行进位电路引脚定义

（2）利用前一步设计好的4位先行进位电路构造4位快速加法器，其引脚定义如图2.12所示，其中 X、Y 为四位相加数，C_{in} 为进位输入，S 为和数输出，C_{out} 为进位输出，G^*、P^* 为4位成组进位生成函数和成组进位传递函数。

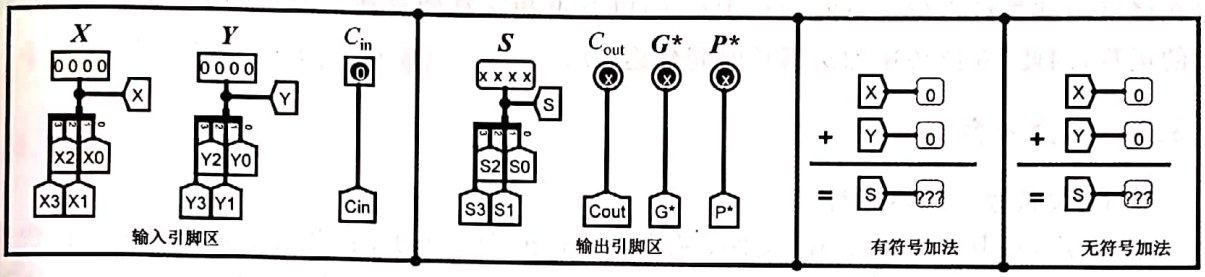

图 2.12　4位快速加法器引脚定义

2.2.4　实验思考

4位快速加法器与4位串行加法器相比，其性能提升了多少？

2.3　多位快速加法器设计实验

2.3.1　实验目的

理解成组进位产生函数，成组进位传递函数的概念，熟悉 Logisim 平台子电路的概念，能利用前述实验封装好的4位先行进位子电路以及4位快速加法器子电路构建16位、32位、64位快速加法器，并能利用相关知识分析对应电路的时间延迟，理解电路并行的概念。

2.3.2　背景知识

有了4位快速加法器和4位可级联的先行进位电路，即可构建组内并行、组间并行的16位快速加法器，如图2.13所示，该电路由4片4位快速加法器和1片可级联先行进位电路构成，4片快速加法器输出4对成组的 P^*、G^* 作为上层先行进位电路的输入，由上层先行进位电路产生下层快速加法器需要的组间进位信号 C_4、C_8、C_{12}。电路工作时，关键路径是底层

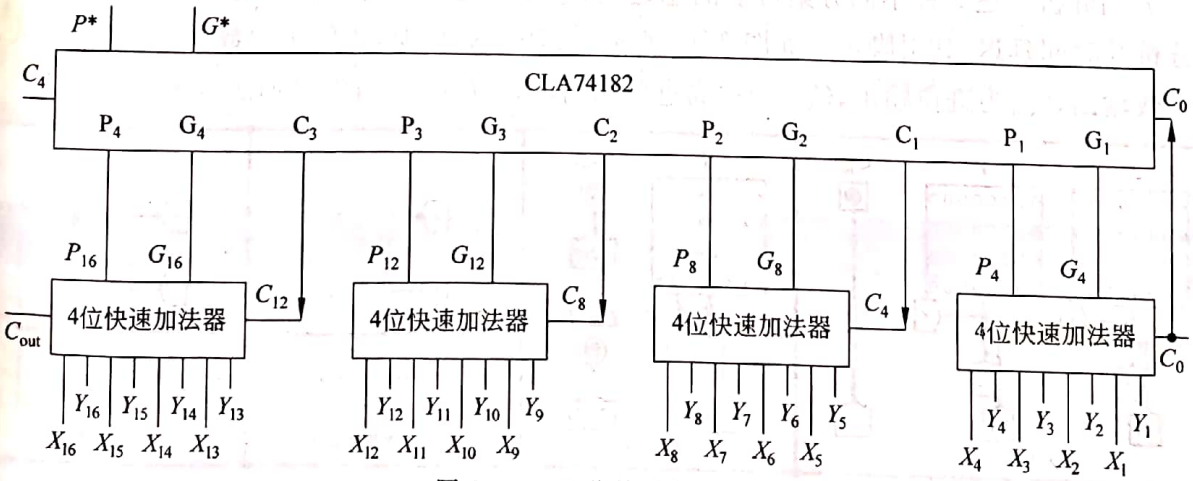

图 2.13　16位快速加法器