第二章 计算机中数的表示及基本逻辑部件

2.8 节拍分配器

2.9 总线

2.10 加法器

2.11 进位链

主要知识点

- 掌握定点数、浮点数、原码、反码、补码和移码的表示
- 掌握数据的校验方法及原理
- 掌握常用的逻辑部件的工作原理
- 掌握加法器进位链的工作原理

2. 1数据信息在机器中的表示

计算机中的数据简称为机器数，一个完整的机器数一般应含有三个方面：符号、数值和小数点。

2.1.1数的定点和浮点表示

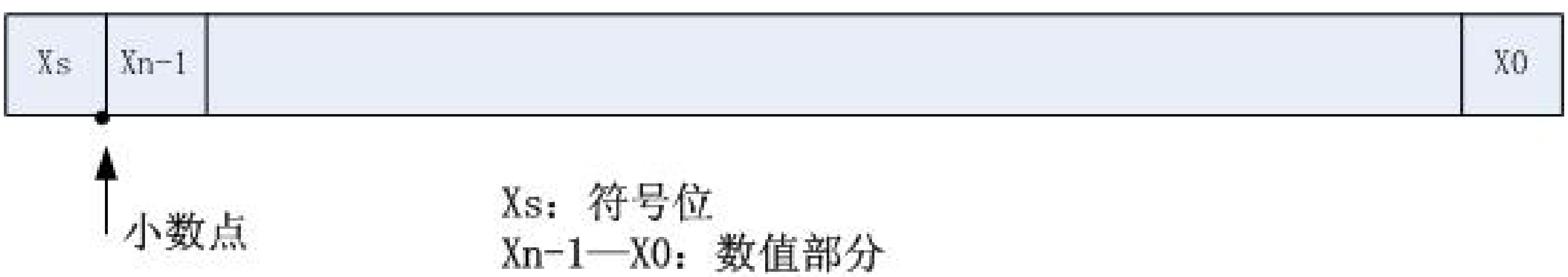
在计算机中按机器数的小数点位置是否固定，把数分成定点表示和浮点表示两种。

1.定点表示法

定点表示法约定机器中所有数据的小数点位置固定不变，一般采用两种简单的约定。

（1）定点小数

小数点放在最高数位之前，符号位之后。



2. 1数据信息在机器中的表示

2.1.1数的定点和浮点表示

假定机器字长为n位，其中一位符号位，其余n-1位是有效数值位，则定点小数能表示的数值范围为：

$$-(1-2^{-(n-1)}) \leqslant X \leqslant 1-2^{-(n-1)}$$

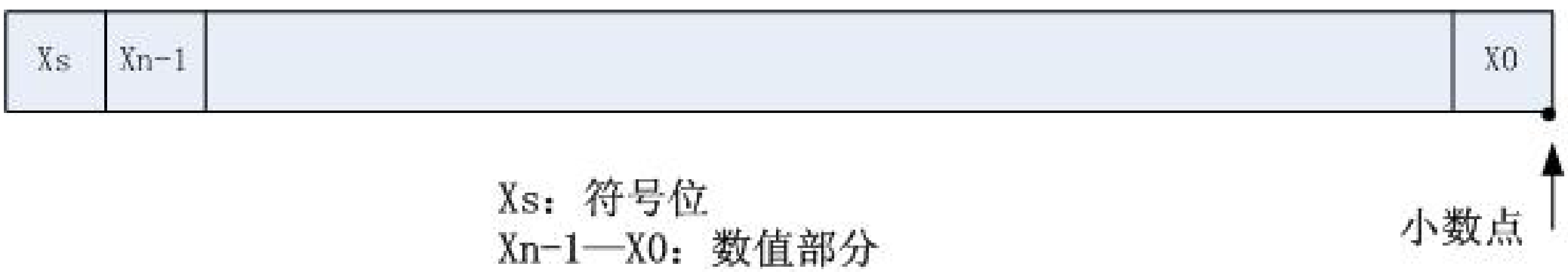
即：$-\underbrace{0.1111111\cdots\cdots1}_{n-1} \sim \underbrace{0.1111111\cdots\cdots1}_{n-1}$

2. 1数据信息在机器中的表示

2.1.1数的定点和浮点表示

（2）定点整数

小数点的位置在数的最低位之后，即参与运算的数是纯整数。



假定机器字长为n位，其中一位符号位，其余n-1位是有效数值位，则定点整数能表示的数值范围为：

$$-(2^{n-1}-1) \leqslant X \leqslant 2^{n-1}-1$$

即：$-\underbrace{1111111\cdots\cdots1}_{n-1} \sim \underbrace{1111111\cdots\cdots1}_{n-1}$

2.数的浮点表示

在浮点数的表示中，由于小数点的位置是变动的，所以，要用一种方法来表示小数点的位置 。

例:

$435=0.435\times 10^{3}$

$110101.11010=0.11010111010\times 2^{110}$

2. 1数据信息在机器中的表示

2.1.1数的定点和浮点表示

一般地，任何一个数都可以写成：

$$X = \pm S \times b^{\pm N}$$

b — 进位基数

±N — 阶

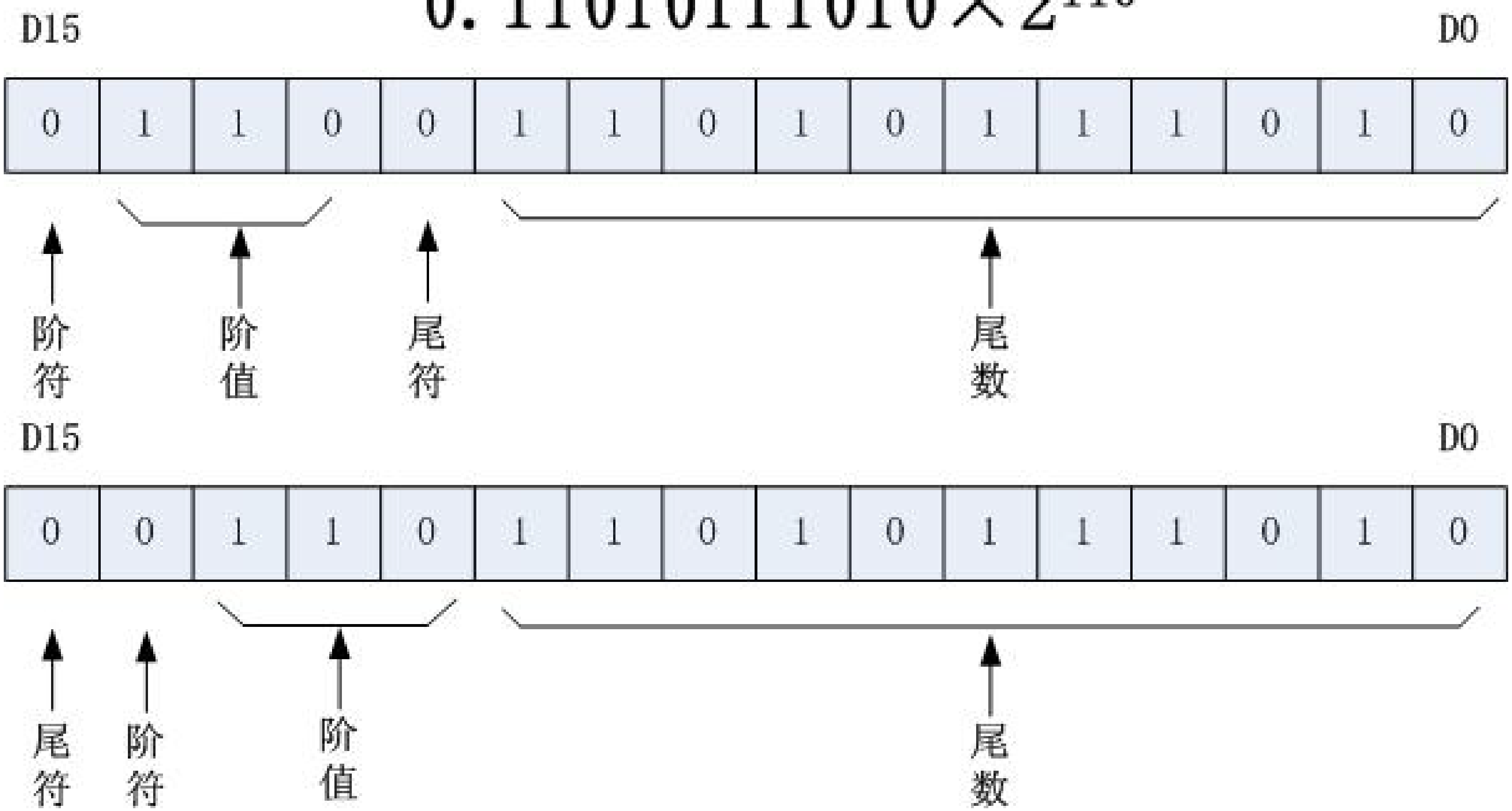
±S — 尾数

2.1数据信息在机器中的表示

2.1.1数的定点和浮点表示

在计算机中浮点数表示为：

$$0.11010111010\times 2^{110}$$



2. 1数据信息在机器中的表示

2.1.1数的定点和浮点表示

- 尾数是一个纯小数，尾数的位数越多，精度越高。整个浮点数的正负号由尾符决定。
- 阶值是一个纯整数，阶符表示数的实际小数点位置的方向，若阶符为正，则实际小数点在假象小数点的右边，若阶符为负，则实际小数点在假象小数点的左边，其具体位置由阶值确定。阶值的位数越多，数的表示范围就越大。

2. 1数据信息在机器中的表示

2.1.1数的定点和浮点表示

根据IEEE 754国际标准，常用的浮点数有两种格式：

(1) 单精度浮点数(32位)，阶码8位(内含1位符号位)，尾数24位(内含1位符号位)。

(2) 双精度浮点数(64位)，阶码11位(内含1位符号位)，尾数53位(内含1位符号位)。

3.浮点数规格化和溢出

浮点数分为规格化浮点数和非规格化浮点数，在计算机中常用的是规格化浮点数。

规格化的目的:提高数的表示精度。

所谓的规格化尾数：就是尾数的最高有效位和符号位相反（补码表示的尾数），即尾数值不为0时，其绝对值应大于或等于$(0.5)_D$。

$$1/2 \leqslant |S| < 1$$

2. 1数据信息在机器中的表示
2.1.1数的定点和浮点表示

例：正尾数 0.0XXXXXXXX 非规格化数

0.1XXXXXXXX 规格化数

负尾数 1.1XXXXXXXX 非规格化数

1.0XXXXXXXX 规格化数

注：以上的尾数是以补码表示。

2. 1数据信息在机器中的表示
2.1.1数的定点和浮点表示

浮点数的溢出有尾数溢出和阶值溢出。

尾数发生溢出时，可以用尾数右移一位，而阶值加一，就消除尾数溢出。

阶值发生溢出时，又分为“下溢”和“上溢”，当阶码的值超出机器中能表示的最小值时，称为“下溢”，一般把该浮点数作为0处理，称为机器零。当阶码的值超出机器中能表示的最大值时，称为“上溢”。

2.1.2机器数的编码表示

1.带符号数的表示

一般情况下，通常用二进制数的最高位表示数的符号，把一个数及符号在机器中的表示加以数值化，这样的数称为机器数，而机器数所表示的数称为该机器数的真值。

（1）原码

机器数的最高位为符号位，0表示正数，1表示负数，数值跟随其后，并以绝对值形式给出。这是与真值最接近的一种表示形式。

原码的定义：

$$[X]_{原}=\begin{cases} X & 0 \leqslant X \leqslant 2^{n-1}-1 \\ 2^{n-1}+|X| & -(2^{n-1}-1) \leqslant X \leqslant 0 \end{cases}$$

即 $[X]_{原}$=符号位+｜X｜。

2. 1数据信息在机器中的表示

2.1.2机器数的编码表示

例：设字长 n=8

X=+1011000，$[X]_{原}$=01011000；

X=-1011000，$[X]_{原}$=11011000。

2.1数据信息在机器中的表示

2.1.2机器数的编码表示

根据定义，当X=-1011000时，

$[X]_{原} = 2^{n-1} + |X| = 2^{8-1} + |-1011000|$

$= 10000000 + |-1011000| = 11011000$

数值零的真值有+0和-0两种表示形式，

$[X]_{原}$也有两种表示形式：

$[+0]_{原} = 00000000$

$[-0]_{原} = 10000000$。

（2）反码

反码的定义：

$$[X]_{反}=\begin{cases} X & 0 \leqslant X \leqslant 2^{n-1}-1 \\ (2^{n}-1)-|X| & -(2^{n-1}-1) \leqslant X \leqslant 0 \end{cases}$$

例：设字长 n=8

X=+1011000，$[X]_{反}$=01011000；

$X=-1011000$,

$[X]_{反}=(2^n-1)-|X|=(2^8-1)-|-1011000|$
$=(100000000-1)-1011000=11111111-1011000$
$=10100111$

反码零有两种表示形式:

$[+0]_{反}=00000000$

$[-0]_{反}=11111111$

（3）补码

补码的定义：

$$[X]_{补}=\begin{cases} X & 0\leqslant X\leqslant 2^{n-1}-1 \\ 2^{n}-|X| & -2^{n-1}\leqslant X<0 \end{cases}$$

例：设字长 n=8

X=+1011000，$[X]_{补}$=01011000；

2. 1数据信息在机器中的表示

2.1.2机器数的编码表示

$X=-1011000$,

$[X]_{补}=2^n-|X|=2^8-|-1011000|$

$=100000000-1011000=10101000$

补码零有唯一的表示形式:

$[+0]_{补}=[-0]_{补}=00000000$

（4）移码

移码的定义：

$$[X]_{移}= 2^{n-1}+X \qquad -2^{n-1}\leqslant X<2^{n-1}$$

例：设字长 n=8， X=+1011000

$[X]_{移}=2^{n-1}+X=10000000+1011000$

$=11011000$

2. 1数据信息在机器中的表示
2.1.2机器数的编码表示

$X=-1011000$,

$[X]_{移}=2^{n-1}+X=10000000+(-1011000)$

$=10000000-1011000=00101000$

移码零有唯一的表示形式:

$[+0]_{移}=[-0]_{移}=10000000$

移码的正数时，符号用“1”表示

移码的负数时，符号用“0”表示

2. 1数据信息在机器中的表示
2.1.2机器数的编码表示

(5)浮点数常用编码

尾数可用原码、补码，常用补码。

阶值可用原码、补码、移码，常用移码。

2.BCD码

用四位二进制表示十进制。

0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

3.字符编码

美国信息交换标准代码ASCII使用的最普遍，一般用8位二进制表示一个字符。

2.1.3数据校验方法

计算机系统中的数据，在读写、存取和传送的过程中可能产生错误。为减少和避免这类错误，一方面是精心设计各种电路，提高计算机硬件的可靠性；另一方面是在数据编码上找出路，即采用某种编码法，通过少量的附加电路，使之能发现某些错误，甚至能确定出错位置，进而实现自动改错的能力。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

数据校验码是一种常用的带有发现某些错误或自动改错能力的数据编码方法。它的实现原理，是加进一些冗余码，使合法数据编码出现某些错误时，就成为非法编码。这样，就可以通过检测编码的合法性来达到发现错误的目的。合理地安排非法编码数量和编码规则，就可以提高发现错误的能力，或自动改正错误的目的。这里用到一个码距的概念。

1.码距

码距—是根据任意两个合法码之间至少有几个二进制位不相同而确定的，不相同的位数称为此编码的码距。

例：四位二进制数有16种不同的编码，在这16种编码中的任意两个编码之间，至少有一位不相同，其码距为1。

0000	0001	0010	0011	0100	0101	0110	0111
1000	1001	1010	1011	1100	1101	1110	1111

码距为1的编码是不能发现错误的。

一般来说，合理地增大码距，就能提高发现错误的能力，但编码所使用的二进制位数变多，增加了数据存储的容量或数据传送的数量。

0001	0010	0100	0111
1000	1011	1101	1110

2. 1数据信息在机器中的表示

2.1.3数据校验方法

在确定与使用数据校验码的时候，通常要考虑在不过多增加硬件开销的情况下，尽可能发现或改正更多的错误。

码距大于1的编码才能发现错误或校正错误。

2 .奇偶校验

奇偶校验码是一种开销最小，能发现数据代码中一位出错情况的编码，常用于存储器读写检查，或ASCII字符传送过程中的检查。它的实现原理，是使码距由1增加到2。若编码中有一个二进制位的值出错了，由1变成0，或由0变成1，这个码都将成为非法编码。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

实现的具体方法，通常是为一个字节补充一个二进制位，称为校验位，用设置校验位的值为0或1，使字节的8位和该校验位含有1值的个数为奇数或偶数。在使用奇数个1的方案进行校验时，称为奇校验，反之，则称为偶校验。

例：以四位二进制编码为例。

0000	0001	0010	0011	0100	0101	0110	0111
1000	1001	1010	1011	1100	1101	1110	1111

奇校验

00001	00010	00100	00111	01000	01011	01101	01110
10000	10011	10101	10110	11001	11010	11100	11111

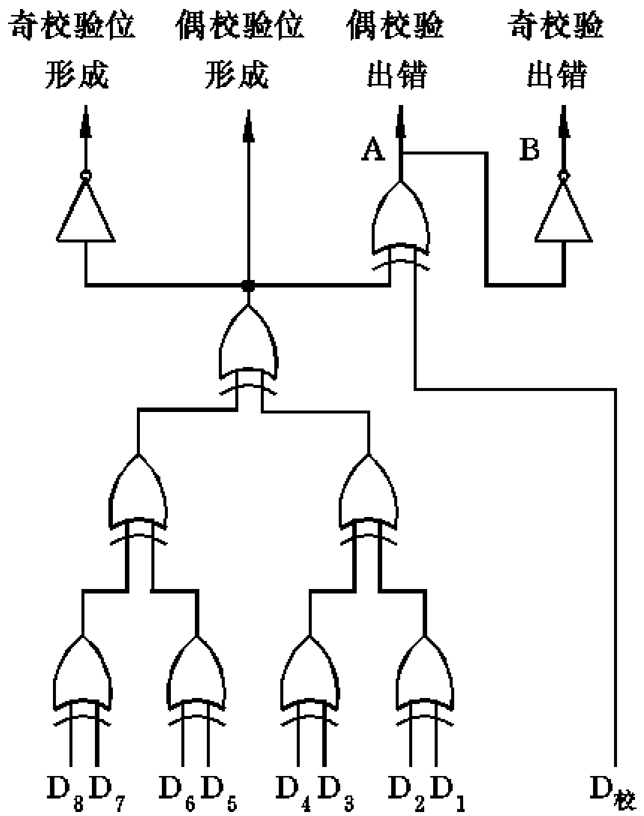
偶校验

00000	00011	00101	00110	01001	01010	01100	01111
10001	10010	10100	10111	11000	11011	11101	11110

2. 1数据信息在机器中的表示

2.1.3数据校验方法

奇偶校验位的实现逻辑电路如下：



3.交叉校验

当一次传送一个数据块时，不但对每一个字节设有奇偶校验位（横向校验），而且全部字节的同一位也设置一个奇偶校验位（纵向校验），可以对数据块的横向和纵向代码同时校验。

2.1数据信息在机器中的表示

2.1.3数据校验方法

例：有四个字节组成的数据块，纵横向均采用奇校验。

	D7	D6	D5	D4	D3	D2	D1	D0	横向校验位
数据1	1	0	0	1	1	0	0	1	1
数据2	0	0	1	0	1	1	0	0	0
数据3	1	1	1	0	0	1	1	1	1
数据4	0	1	1	1	1	1	1	0	1
纵向校验位	1	1	0	1	0	0	1	1	0

4.循环冗余校验（CRC）码

二进制信息位流沿一条线逐位在部件之间或计算机之间传送称为串行传送。CRC(cyclic redundancy check)码可以发现并纠正信息存储或传送过程中连续出现的多位错误，因此在磁介质存储和计算机之间通信方面得到广泛应用。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

CRC码一般是指k位信息码之后拼接r位校验码。

应用CRC码的关键是如何从k位信息位简便地得到r位校验位(编码)，以及如何从k+r位信息码判断是否出错。

下面仅就CRC码应用中的问题做简单介绍。

（1） CRC码的编码方法

在CRC码的编码中，要用到模2运算。

模2运算是指以按位模2相加为基础的四则运算，运算时不考虑进位和借位。

① 模2加减：即按位加，可用异或逻辑实现。模2加与模2减的结果相同，即0±0=0，0±1=1，1±0=1，1±1=0。两个相同的数据的模2和为0。

② 模2乘——按模2加求部分积之和。

例：1010乘以101。

$$
\begin{array}{r}
1010 \\
\times \quad 101 \\
\hline
1010 \\
0000 \\
1010 \\
\hline
100010
\end{array}
$$

2. 1数据信息在机器中的表示

2.1.3数据校验方法

③ 模2除——按模2减求部分余数。每求一位商应使部分余数减少一位。上商的原则是：当部分余数的首位为1时，商取1；当部分余数的首位为0时，商取0。当部分的余数的位数小于除数的位数时，该余数即为最后余数。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

例：1110除以101。

```
          11
    ┌──────
101 │ 1110
      101
    ──────
       100
       101
    ──────
        01
```

2. 1数据信息在机器中的表示
2.1.3数据校验方法

CRC码的编码方法：

设被校验数据M(x)是由n位二进制组成，

$M(x)=C_{n-1}x^{n-1}+C_{n-2}x^{n-2}+\ldots+C_ix^i+\ldots+C_1x+C_0$

式中C_i为0或1。

将M(x)信息位左移k位后，被一个约定的生成多项式G(x)相除，G(x)必须是k+1位二进制数组成，相除后得到k位余数就是校验位。将k位余数拼接到n位的M(x)后面,即形成n+k位长的循环冗余校验码，称为（n+k,n)码，因此所得CRC码可被G(x)表示的数码除尽。

2. 1数据信息在机器中的表示
2.1.3数据校验方法

例：对4位有效信息M(x)=1100,求循环校验编码，选择生成多项式G(x)=1011。

$M(x)=x^3+x^2=1100$ (n=4)

$M(x)\cdot x^3=x^6+x^5=1100000$ (左移k=3位)

$G(x)=x^3+x+1=1011$ (k+1=4位)

$M(x)\cdot x^3/G(x)=1100000/1011$(模2除)

结果：余数=010，商= 1110

2. 1数据信息在机器中的表示
2.1.3数据校验方法

将余数拼接到M(x)上，即得到CRC校验码。

M(x)的CRC校验码：

CRC校验码=1100000+010=1100010

(模2加)

将编好的循环校验码称为(7，4)码，即n=4,k=3。

（2）CRC的纠错原理

将收到的循环校验码用约定的生成多项式G(x)去除，如果循环校验码无误则余数应为0，如有某一位出错，则余数不为0，不同位数出错余数不同。如下表给出的关系可作为(7，4)码的判别依据。

(7，4)循环码的出错模式(生成多项式G(x)=1011)

	D_6	D_5	D_4	D_3	D_2	D_1	D_0	余数	出错位
正 确	1	1	0	0	0	1	0	0 0 0	无
错 误	1	1	0	0	0	1	1	0 0 1	D_0
	1	1	0	0	0	0	0	0 1 0	D_1
	1	1	0	0	1	1	0	1 0 0	D_2
	1	1	0	1	0	1	0	0 1 1	D_3
	1	1	1	0	0	1	0	1 1 0	D_4
	1	0	0	0	0	1	0	1 1 1	D_5
	0	1	0	0	0	1	0	1 0 1	D_6

2. 1数据信息在机器中的表示

2.1.3数据校验方法

更换不同的待测码字可以证明：余数与出错位的对应关系是不变的，只与码制和生成多项式有关。

对于其他码制或选用其他生成多项式，出错模式将发生变化。

（3）生成多项式

并不是任何一个(k+1)位多项式都可以作为生成多项式的。从检错及纠错的要求出发，生成多项式应能满足下列要求：

① 任何一位发生错误都应使余数不为0。

② 不同位发生错误应当使余数不同。

③ 对余数继续作模2除，应使余数循环。

常用多项式

名称	G(x)多项式	G(x)二进制码
CRC-5-USB	$x^5 + x^2 + 1$ (用途：USB 信令包)	100101
CRC-7	$x^7 + x^3 + 1$ (用途：通信系统)	10001001
CRC-8-ATM	$x^8 + x^2 + x + 1$ (用途：ATM HEC)	100000111
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$ (用途：通信系统)	1100000001111
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$ (X25, V.41, Bluetooth, PPP, IrDA)	10001000000100001
CRC-16-IBM	$x^{16} + x^{15} + x^2 + 1$	11000000000000101
CRC-32-IEEE 802.3	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	100000100110000010001110110110111

5 .海明校验码

　　这是由Richard Hamming于1950年提出的、目前还被广泛采用。

实现原理:

　　在数据中加入几个校验位，并把数据的每一个二进制位分配在几个奇偶校验组中。当某一位出错后，就会引起有关的几个校验组的值发生变化，这不但可以发现出错，还能指出是哪一位出错，为自动纠错提供了依据。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

假设校验位的个数为r，则它能表示2^r个信息(2^r 编码)，用其中的一个信息（编码）指出“没有错误”，其余的2^r-1个信息（编码）指出错误发生在哪一位。然而错误也可能发生在校验位，因此只有$k=2^r-1-r$个信息能用于纠正被传送数据的位数，也就是说要满足关系： $2^r \geq k+r+1$。

例：假如校验位有4位，则$k=2^4-1-4=11$,即被校验的数据位最长为11位。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

如要能检测与自动校正一位错，并发现两位错，此时校验位的位数r和数据位的位数k应满足下述关系： $2^{r-1} \geq k+r$

数据位k与校验位r的对应关系如表：

k 值（数据位）	最 小 的 r 值（校验位）
1~4	4
5~11	5
12~26	6
27~57	7
58~120	8

（1）海明码编码规则

若海明码的最高位号为m，最低位号为1， 即$H_mH_{m-1}\ldots H_2H_1$，则此海明码的编码规律通常是：

① 校验位与数据位之和为m，每个校验位Pi在海明码中被分在位号2^{i-1}的位置，其余各位为数据位，并按从低向高逐位依次排列的关系分配各数据位。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

② 海明码的每一位码H_i(包括数据位和校验位本身)由多个校验位校验，其关系是被校验的每一位位号要等于校验它的各校验位的位号之和。

例：设校验位为4位，$P_4P_3P_2P_1$，被校验的数据位是11位，$D_{10}D_9D_8D_7D_6D_5D_4D_3D_2D_1D_0$,生成海明码。

生成的海明码：

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

海明码位号	数据位/校验位	参与校验的校验位位号	被校验位的海明码位号=参与校验位位号之和
H_1	P_1	$1(P_1)$	1=1
H_2	P_2	$2(P_2)$	2=2
H_3	D_0	$1(P_1)$，$2(P_2)$	3=1+2
H_4	P_3	$4(P_3)$	4=4
H_5	D_1	$1(P_1)$，$4(P_3)$	5=1+4
H_6	D_2	$2(P_2)$，$4(P_3)$	6=2+4
H_7	D_3	$1(P_1)$，$2(P_2)$，$4(P_3)$	7=1+2+4
H_8	P_4	$8(P_4)$	8=8
H_9	D_4	$1(P_1)$，$8(P_4)$	9=1+8
H_{10}	D_5	$2(P_2)$，$8(P_4)$	10=2+8
H_{11}	D_6	$1(P_1)$，$2(P_2)$，$8(P_4)$	11=1+2+8
H_{12}	D_7	$4(P_3)$，$8(P_4)$	12=4+8
H_{13}	D_8	$1(P_1)$，$4(P_3)$，$8(P_4)$	13=1+4+8
H_{14}	D_9	$2(P_2)$，$4(P_3)$，$8(P_4)$	14=2+4+8
H_{15}	D_{10}	$1(P_1)$，$2(P_2)$，$4(P_3)$，$8(P_4)$	15=1+2+4+8

③ 校验位$P_4P_3P_2P_1$值的确定。

校验位P_i的取值就是该校验位所校验的数据位的异或。

$$P1=D0\oplus D1\oplus D3\oplus D4\oplus D6\oplus D8\oplus D10$$
$$P2=D0\oplus D2\oplus D3\oplus D5\oplus D6\oplus D9\oplus D10$$
$$P3=D1\oplus D2\oplus D3\oplus D7\oplus D8\oplus D9\oplus D10$$
$$P4=D4\oplus D5\oplus D6\oplus D7\oplus D8\oplus D9\oplus D10$$

这时$P_4P_3P_2P_1$的取值是采用偶校验，当采用奇校验时,$P_4P_3P_2P_1$取偶校验值得反。

（2）海明码校验和纠错

$$S1=D0\oplus D1\oplus D3\oplus D4\oplus D6\oplus D8\oplus D10\oplus P1$$
$$S2=D0\oplus D2\oplus D3\oplus D5\oplus D6\oplus D9\oplus D10\oplus P2$$
$$S3=D1\oplus D2\oplus D3\oplus D7\oplus D8\oplus D9\oplus D10\oplus P3$$
$$S4=D4\oplus D5\oplus D6\oplus D7\oplus D8\oplus D9\oplus D10\oplus P4$$

采用偶校验时，$S_4S_3S_2S_1$的值为0000，则传送正确。否则传送错误。

采用奇校验时，$S_4S_3S_2S_1$的值为1111，则传送正确。否则传送错误。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

当传送发生错误时（以偶校验为例），假设$S_4S_3S_2S_1$ =1100，说明是P_4P_3同时校验的那位数据位出错。P_4P_3同时校验的数据位是D_7出错，所以，只要将该位取反就可纠错。

例：请写出被校验数据10110100110的海明码，采用4位校验位和偶校验。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

已知： $D_{10}D_9D_8D_7D_6D_5D_4D_3D_2D_1D_0=10110100110$

$$P_1=D_0\oplus D_1\oplus D_3\oplus D_4\oplus D_6\oplus D_8\oplus D_{10}$$
$$P_2=D_0\oplus D_2\oplus D_3\oplus D_5\oplus D_6\oplus D_9\oplus D_{10}$$
$$P_3=D_1\oplus D_2\oplus D_3\oplus D_7\oplus D_8\oplus D_9\oplus D_{10}$$
$$P_4=D_4\oplus D_5\oplus D_6\oplus D_7\oplus D_8\oplus D_9\oplus D_{10}$$

$$P1=0\oplus 1\oplus 0\oplus 0\oplus 0\oplus 1\oplus 1=1$$
$$P2=0\oplus 1\oplus 0\oplus 1\oplus 0\oplus 0\oplus 1=1$$
$$P3=1\oplus 1\oplus 0\oplus 1\oplus 1\oplus 0\oplus 1=1$$
$$P4=0\oplus 1\oplus 0\oplus 1\oplus 1\oplus 0\oplus 1=0$$

已知： $D_{10}D_9D_8D_7D_6D_5D_4D_3D_2D_1D_0=10110100110$

$P1=0\oplus1\oplus0\oplus0\oplus0\oplus1\oplus1=1$

$P2=0\oplus1\oplus0\oplus1\oplus0\oplus0\oplus1=1$

$P3=1\oplus1\oplus0\oplus1\oplus1\oplus0\oplus1=1$

$P4=0\oplus1\oplus0\oplus1\oplus1\oplus0\oplus1=0$

生成海明码：

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
1	0	1	1	0	1	0	0	0	1	1	1	0	1	1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

如在传送中未产生错误，则：

$$S_1=0\oplus1\oplus0\oplus0\oplus0\oplus1\oplus1\oplus1\ (P_1)\ =0$$
$$S_2=0\oplus1\oplus0\oplus1\oplus0\oplus0\oplus1\oplus1\ (P_2)\ =0$$
$$S_3=1\oplus1\oplus0\oplus1\oplus1\oplus0\oplus1\oplus1\ (P_3)\ =0$$
$$S_4=0\oplus1\oplus0\oplus1\oplus1\oplus0\oplus1\oplus0\ (P_4)\ =0$$

如产生错误，D_0出错

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
1	0	1	1	0	1	0	0	0	1	1	1	1	1	1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

2. 1数据信息在机器中的表示

2.1.3数据校验方法

$$P_1=D_0 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_8 \oplus D_{10}$$
$$P_2=D_0 \oplus D_2 \oplus D_3 \oplus D_5 \oplus D_6 \oplus D_9 \oplus D_{10}$$
$$P_3=D_1 \oplus D_2 \oplus D_3 \oplus D_7 \oplus D_8 \oplus D_9 \oplus D_{10}$$
$$P_4=D_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8 \oplus D_9 \oplus D_{10}$$

$$P1=0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1=1$$
$$P2=0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1=1$$
$$P3=1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1=1$$
$$P4=0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1=0$$

2.1数据信息在机器中的表示

2.1.3数据校验方法

$S_1=1\oplus1\oplus0\oplus0\oplus0\oplus1\oplus1\oplus1\ (P_1)=1$

$S_2=1\oplus1\oplus0\oplus1\oplus0\oplus0\oplus1\oplus1\ (P_2)=1$

$S_3=1\oplus1\oplus0\oplus1\oplus1\oplus0\oplus1\oplus1\ (P_3)=0$

$S_4=0\oplus1\oplus0\oplus1\oplus1\oplus0\oplus1\oplus0\ (P_4)=0$

$S_4S_3S_2S_1$=0011；3号位（H3）出错

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

如产生错误，D_{10}出错

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
1	0	1	1	0	1	0	0	0	1	1	1	0	1	1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
0	0	1	1	0	1	0	0	0	1	1	1	0	1	1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

2. 1数据信息在机器中的表示

2.1.3数据校验方法

$$P_1=D_0\oplus D_1\oplus D_3\oplus D_4\oplus D_6\oplus D_8\oplus D_{10}$$
$$P_2=D_0\oplus D_2\oplus D_3\oplus D_5\oplus D_6\oplus D_9\oplus D_{10}$$
$$P_3=D_1\oplus D_2\oplus D_3\oplus D_7\oplus D_8\oplus D_9\oplus D_{10}$$
$$P_4=D_4\oplus D_5\oplus D_6\oplus D_7\oplus D_8\oplus D_9\oplus D_{10}$$

$$P_1=0\oplus 1\oplus 0\oplus 0\oplus 0\oplus 1\oplus 1=1$$
$$P_2=0\oplus 1\oplus 0\oplus 1\oplus 0\oplus 0\oplus 1=1$$
$$P_3=1\oplus 1\oplus 0\oplus 1\oplus 1\oplus 0\oplus 1=1$$
$$P_4=0\oplus 1\oplus 0\oplus 1\oplus 1\oplus 0\oplus 1=0$$

2.1数据信息在机器中的表示

2.1.3数据校验方法

$$S_1=0\oplus 1\oplus 0\oplus 0\oplus 0\oplus 1\oplus 0\oplus 1\,(P_1)=1$$

$$S_2=0\oplus 1\oplus 0\oplus 1\oplus 0\oplus 0\oplus 0\oplus 1\,(P_2)=1$$

$$S_3=1\oplus 1\oplus 0\oplus 1\oplus 1\oplus 0\oplus 0\oplus 1\,(P_3)=1$$

$$S_4=0\oplus 1\oplus 0\oplus 1\oplus 1\oplus 0\oplus 0\oplus 0\,(P_4)=1$$

$S_4S_3S_2S_1$=1111；15号位（H15）出错

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

2. 1数据信息在机器中的表示

2.1.3数据校验方法

如产生错误，P_4出错

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
1	0	1	1	0	1	0	0	0	1	1	1	0	1	1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
1	0	1	1	0	1	0	1	0	1	1	1	0	1	1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

1907-2007
百年同济
TONGJI UNIVERSITY

2. 1数据信息在机器中的表示

2.1.3数据校验方法

$$S_1=0\oplus1\oplus0\oplus0\oplus0\oplus1\oplus1\oplus1\ (P_1)=0$$

$$S_2=0\oplus1\oplus0\oplus1\oplus0\oplus0\oplus1\oplus1\ (P_2)=0$$

$$S_3=1\oplus1\oplus0\oplus1\oplus1\oplus0\oplus1\oplus1\ (P_3)=0$$

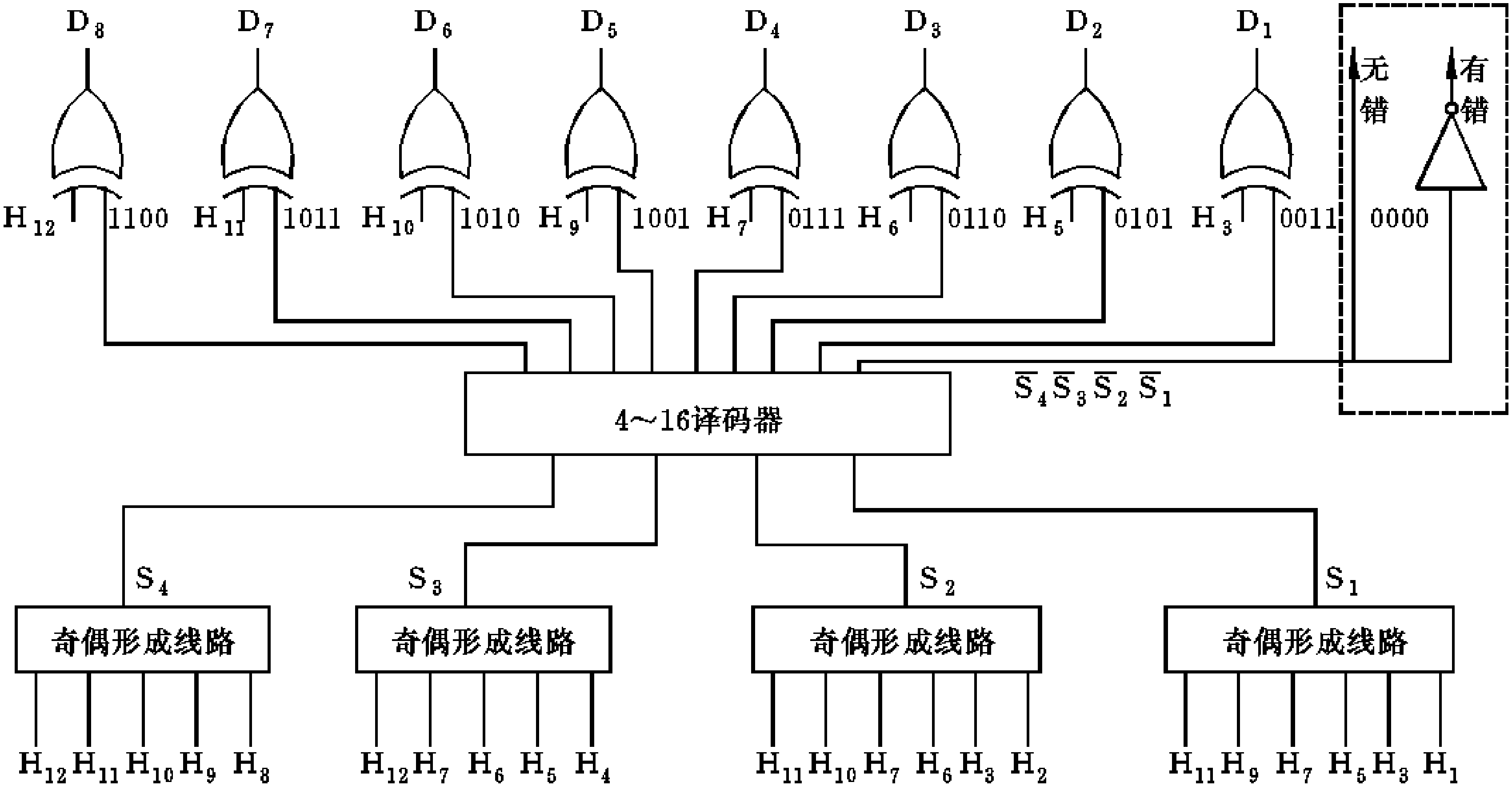
$$S_4=0\oplus1\oplus0\oplus1\oplus1\oplus0\oplus1\oplus1\ (P_4)=1$$

$S_4S_3S_2S_1$=1000；8号位（H8）出错

H15	H14	H13	H12	H11	H10	H9	H8	H7	H6	H5	H4	H3	H2	H1
D10	D9	D8	D7	D6	D5	D4	P4	D3	D2	D1	P3	D0	P2	P1

2. 1数据信息在机器中的表示

2.1.3数据校验方法



2. 1数据信息在机器中的表示

2.1.3数据校验方法

H_{12}	H_{11}	H_{10}	H_9	H_8	H_7	H_6	H_5	H_4	H_3	H_2	H_1
D_7	D_6	D_5	D_4	P_4	D_3	D_2	D_1	P_3	D_0	P_2	P_1

$$S_1=D_0 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus P_1=H_3 \oplus H_5 \oplus H_7 \oplus H_9 \oplus H_{11} \oplus H_1$$

$$S_2=D_0 \oplus D_2 \oplus D_3 \oplus D_5 \oplus D_6 \oplus P_2=H_3 \oplus H_6 \oplus H_7 \oplus H_{10} \oplus H_{11} \oplus H_2$$

$$S_3=D_1 \oplus D_2 \oplus D_3 \oplus D_7 \oplus P_3=H_5 \oplus H_6 \oplus H_7 \oplus H_{12} \oplus H_4$$

$$S_4=D_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus P_4=H_9 \oplus H_{10} \oplus H_{11} \oplus H_{12} \oplus H_8$$

2.1数据信息在机器中的表示

2.1.3数据校验方法

假如要进一步判别是1位错还是2位错，则再增加一个校验位P_5。

H_{13}	H_{12}	H_{11}	H_{10}	H_9	H_8	H_7	H_6	H_5	H_4	H_3	H_2	H_1
P_5	D_7	D_6	D_5	D_4	P_4	D_3	D_2	D_1	P_3	D_0	P_2	P_1

$$P_5=D_0\oplus D_1\oplus D_2\oplus D_3\oplus D_4\oplus D_5\oplus D_6\oplus D_7\oplus P_4\oplus P_3\oplus P_2\oplus P_1$$

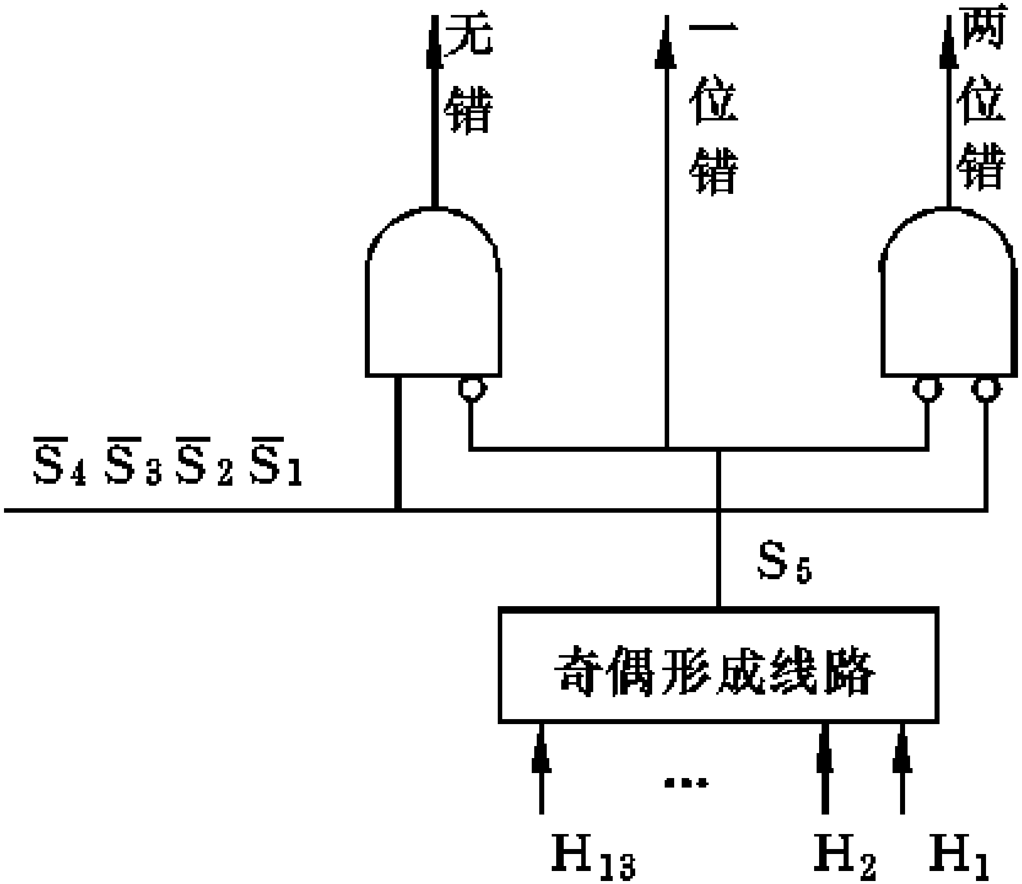
$$S_5=D_0\oplus D_1\oplus D_2\oplus D_3\oplus D_4\oplus D_5\oplus D_6\oplus D_7\oplus P_4\oplus P_3\oplus P_2\oplus P_1\oplus P_5$$

此时增加了一个奇偶形成线路S_5。如为一位错，仍按前图来纠正数据位；如为两位错（S_5为0），能检测出，则无法纠正错误。

2. 1数据信息在机器中的表示

2.1.3数据校验方法

无错=1: $S_4S_3S_2S_1=0000$ $S_5=0$
一位错=1: $S_4S_3S_2S_1$=非全零 $S_5=1$
两位错=1: $S_4S_3S_2S_1$=非全零 $S_5=0$



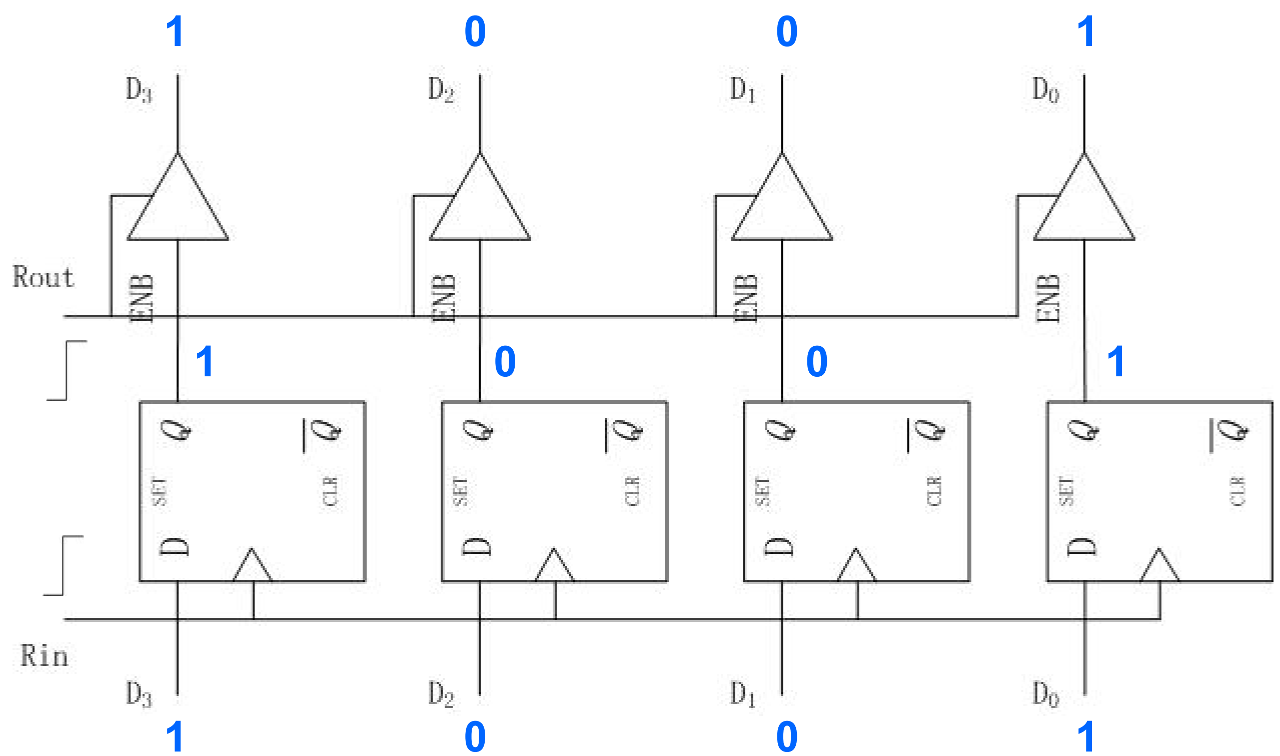
习 题（1）

P67

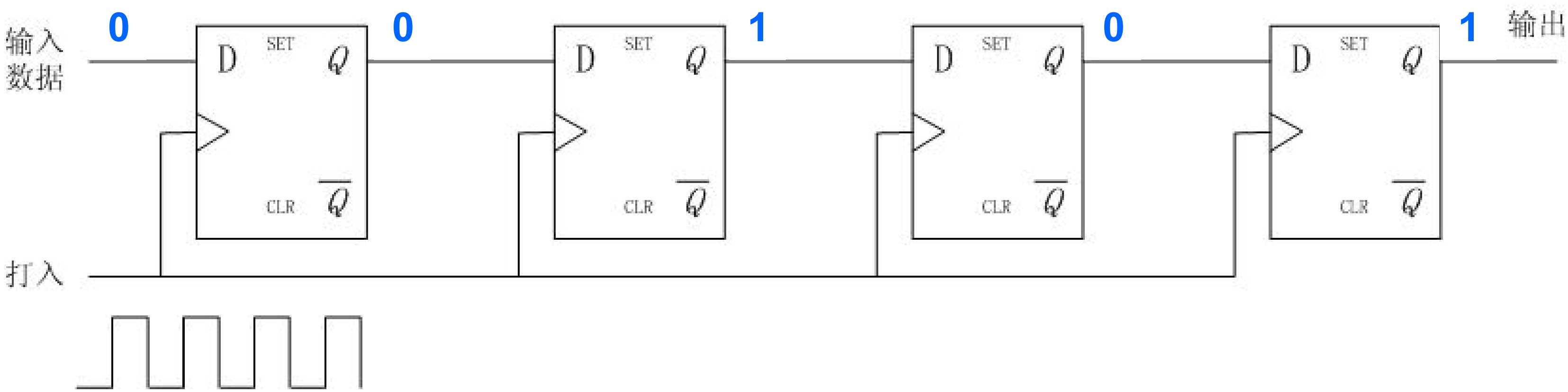
习题：

9，10，11，12，25，26，27，28，29，30

1.D触发器构成寄存器

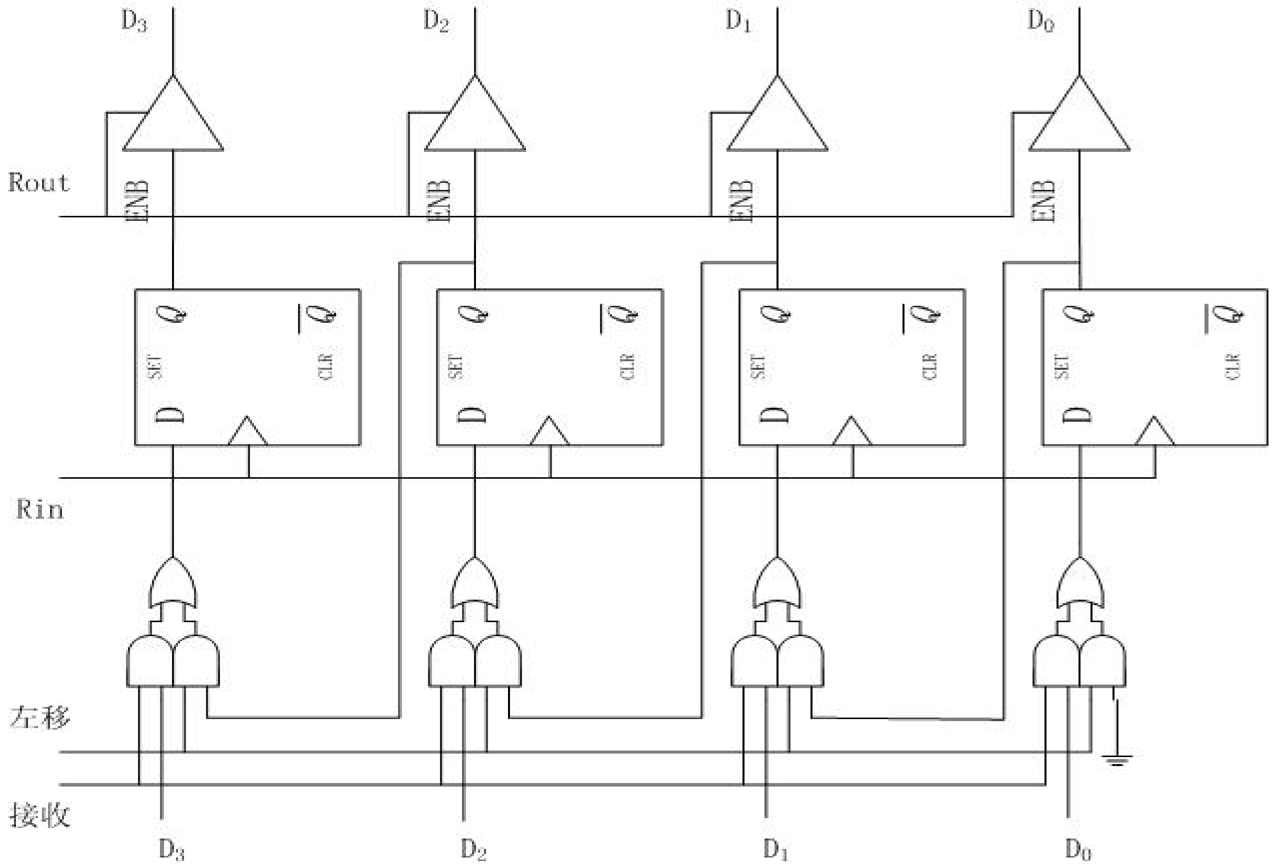

1
0
0
1
D3
D2
D1
D0
Rout
ENB
ENB
ENB
ENB
1
0
0
1
Q
Q
SET
CLR
D
Rin
D3
D2
D1
D0
1
0
0
1

2.串行移位寄存器

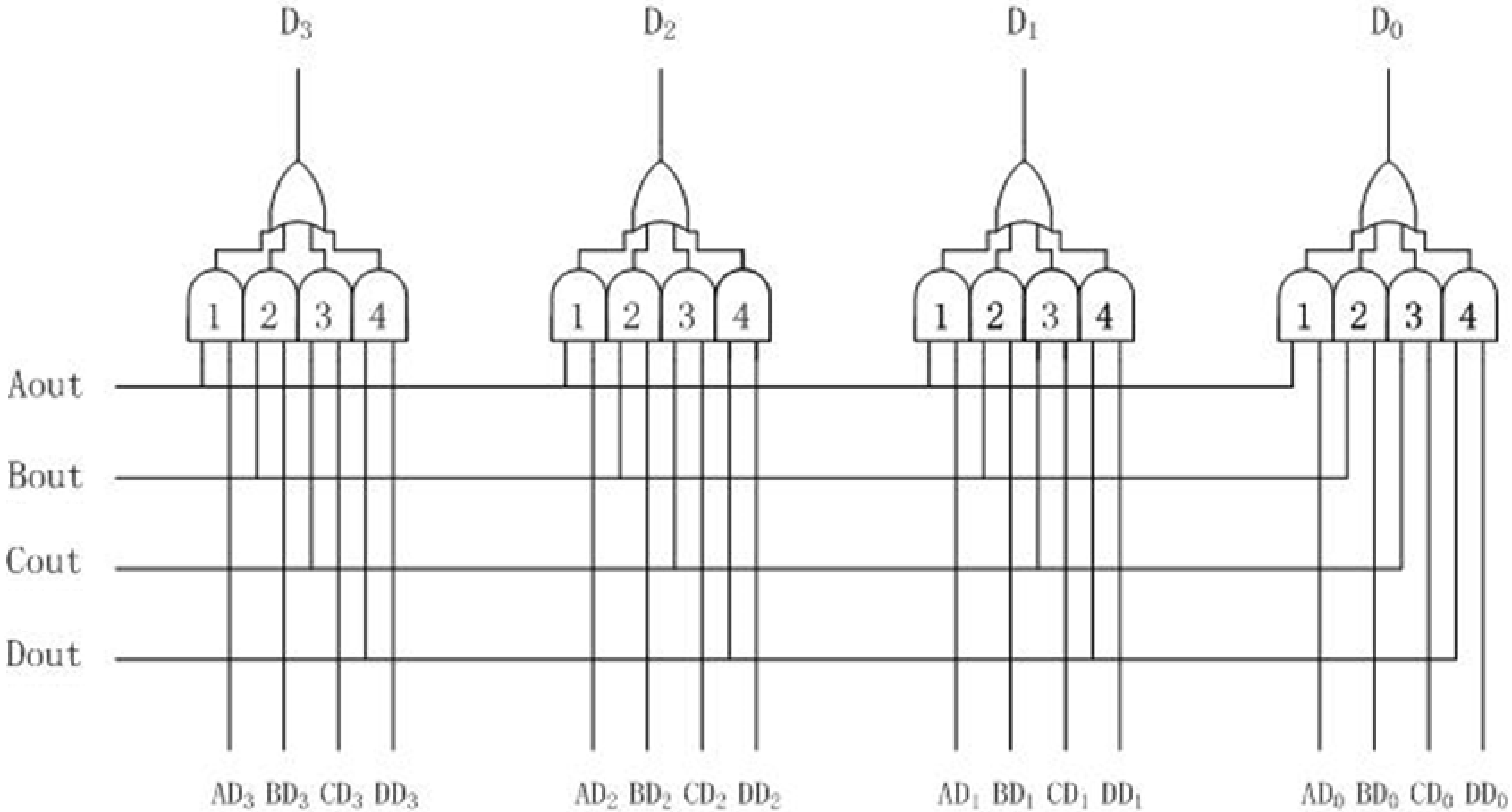


移入 0101

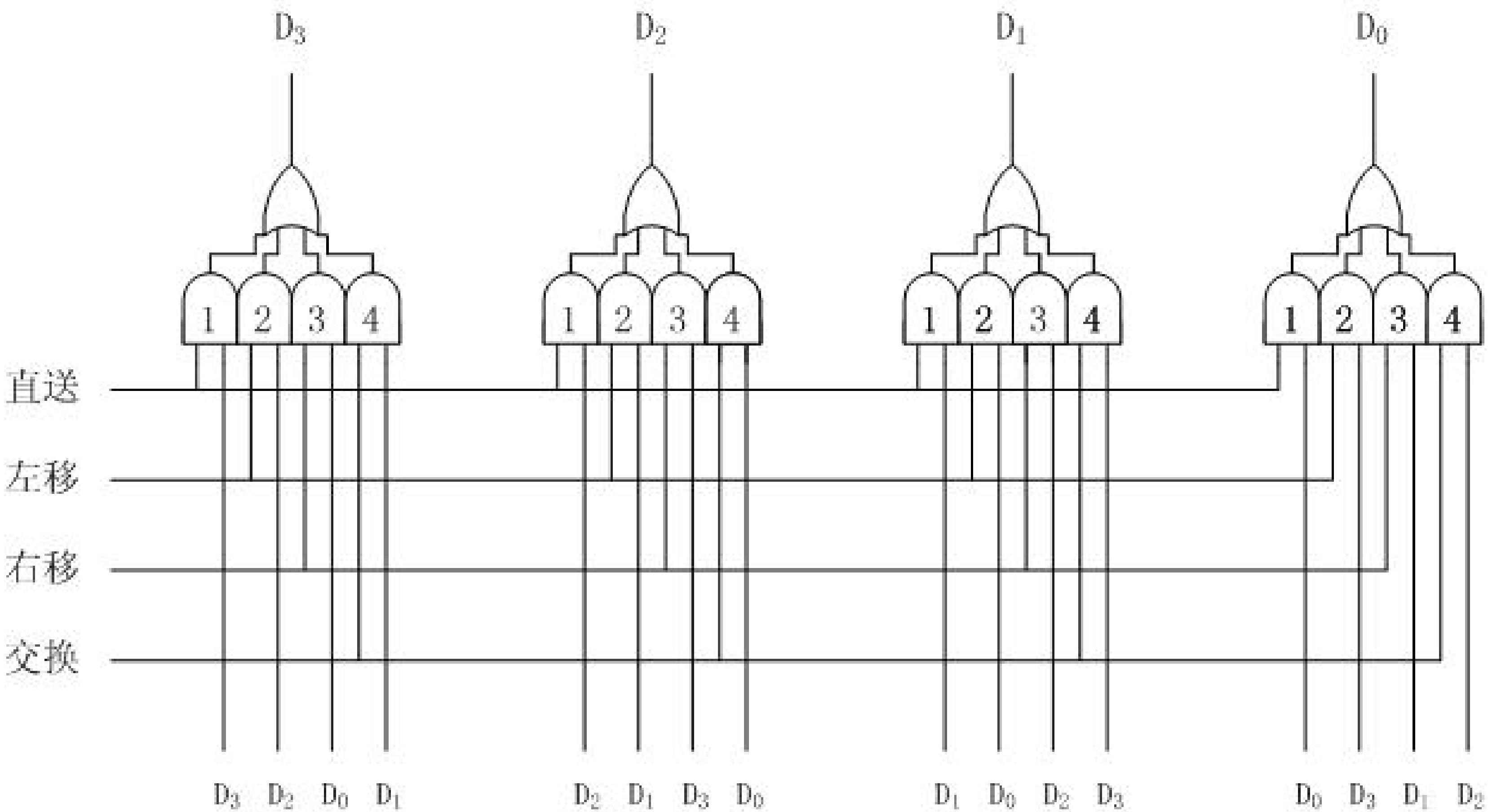
3.并行移位寄存器



2.3 多路选择器


D3
D2
D1
D0
1
2
3
4
Aout
Bout
Cout
Dout
AD3 BD3 CD3 DD3
AD2 BD2 CD2 DD2
AD1 BD1 CD1 DD1
AD0 BD0 CD0 DD0

2.4 移位器

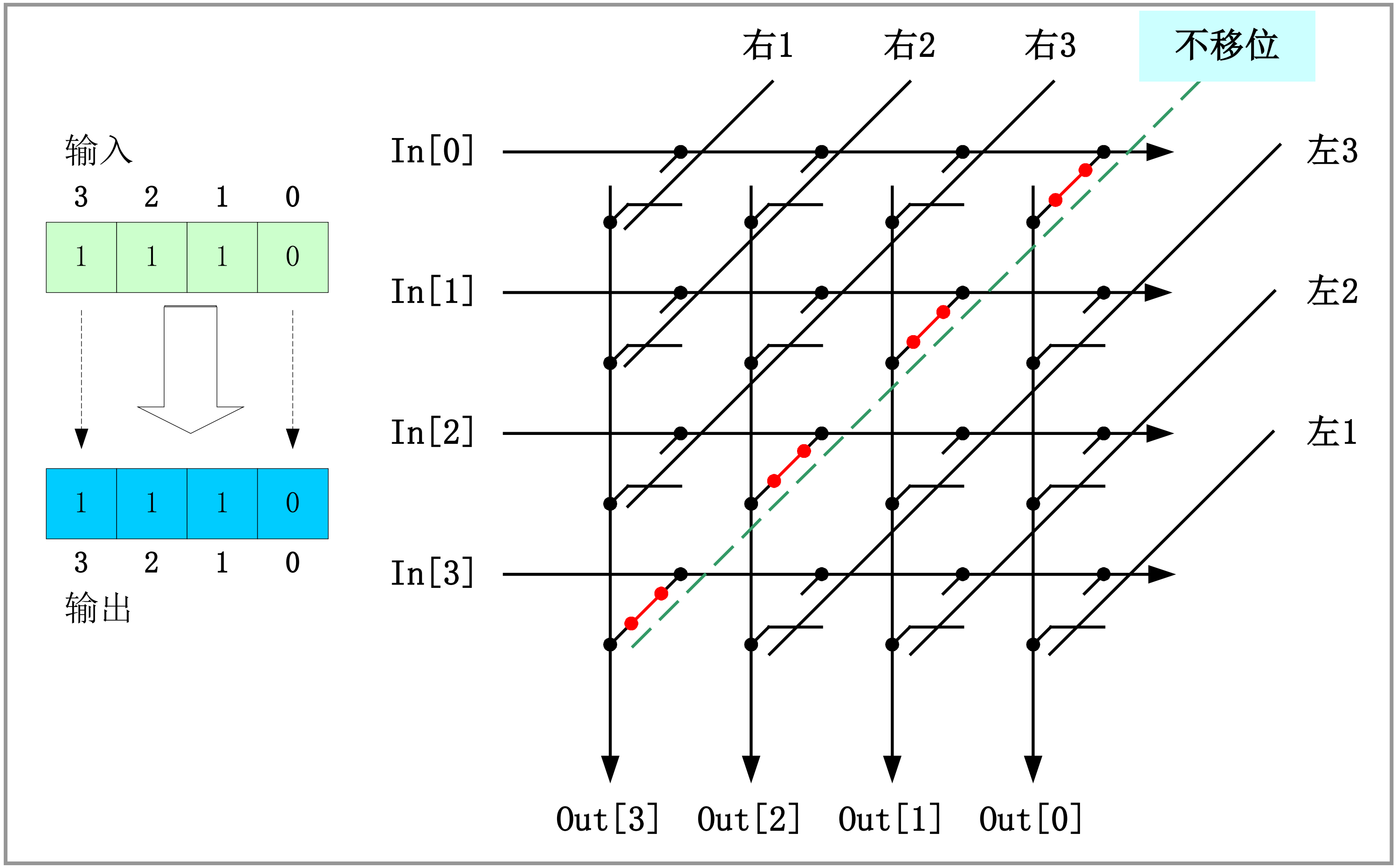

D3
D2
D1
D0
1
2
3
4
直送
左移
右移
交换
D3 D2 D0 D1
D2 D1 D3 D0
D1 D0 D2 D3
D0 D3 D1 D2

2.4 移位器

桶形移位器

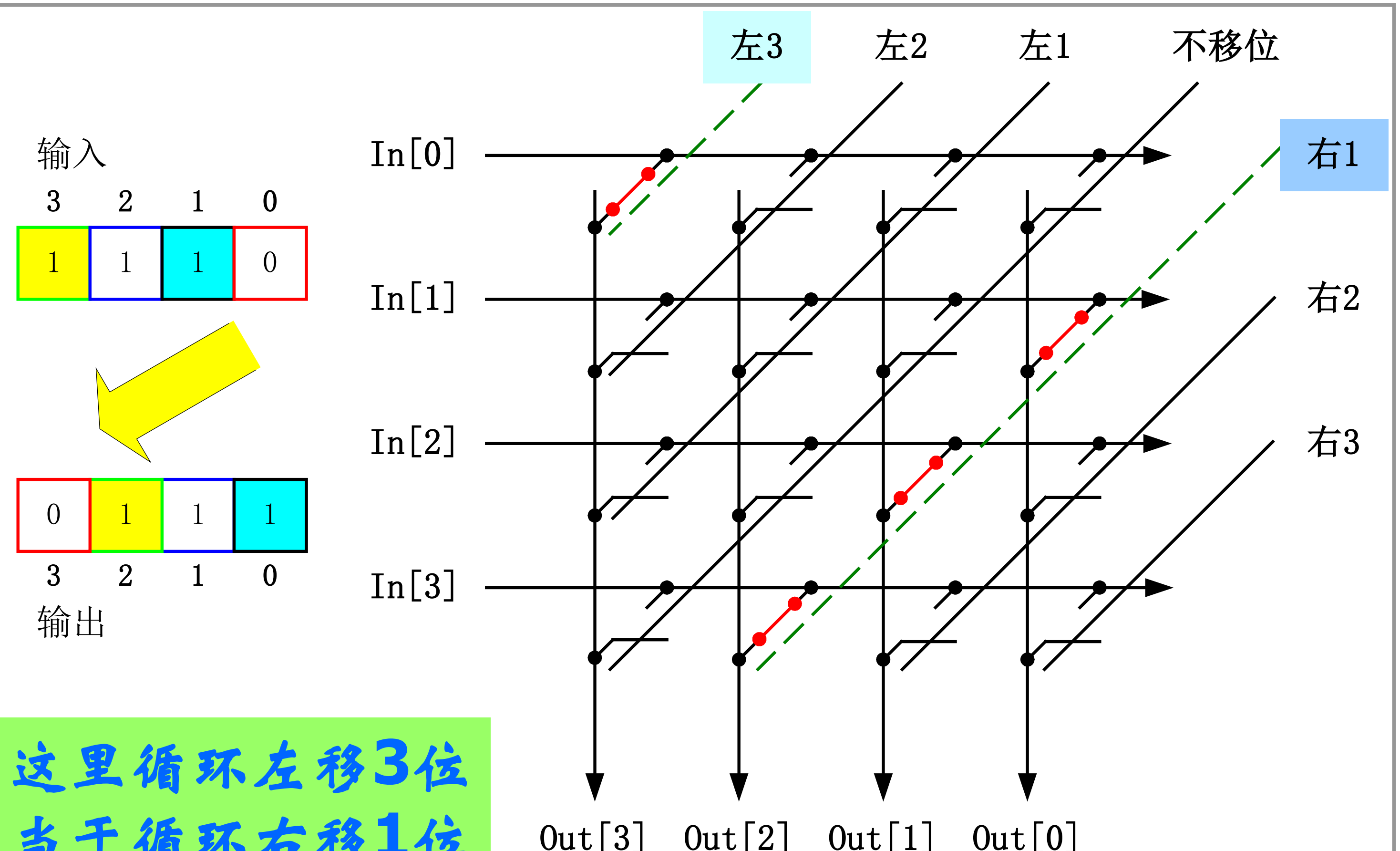
- 通常的移位器都是一个时钟脉冲左移或者右移1位。
- 桶型移位器采用了开关矩阵电路，可以做到用1个时钟脉冲移位任意位。
- 参看下面的开关矩阵工作示意图。

2.4 移位器



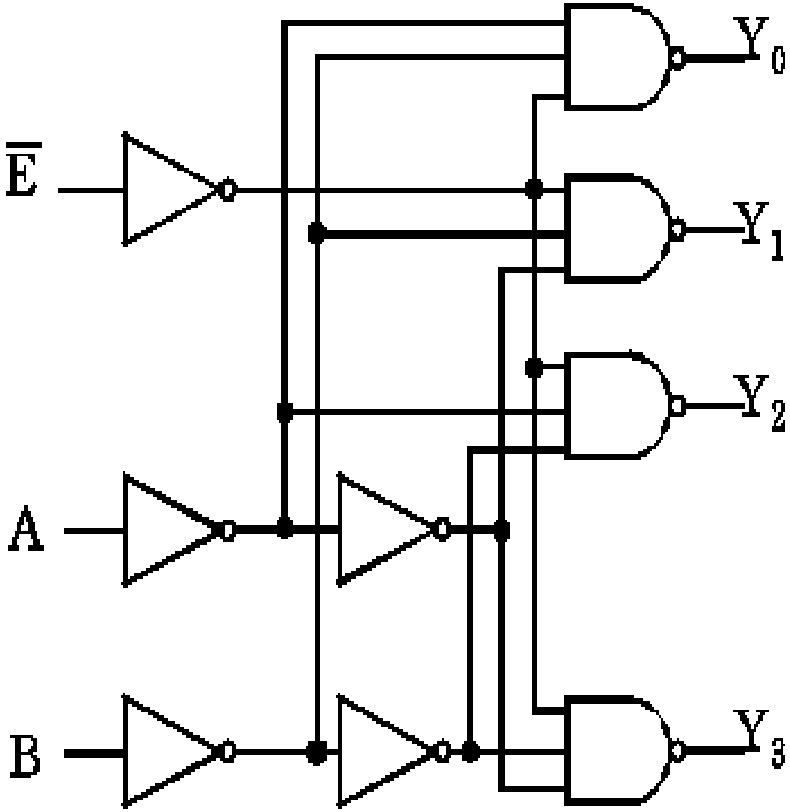
输入
3 2 1 0
1 1 1 0
1 1 1 0
3 2 1 0
输出
右1
右2
右3
不移位
In[0]
In[1]
In[2]
In[3]
左3
左2
左1
Out[3]
Out[2]
Out[1]
Out[0]

2.4 移位器



在这里循环左移3位
相当于循环右移1位

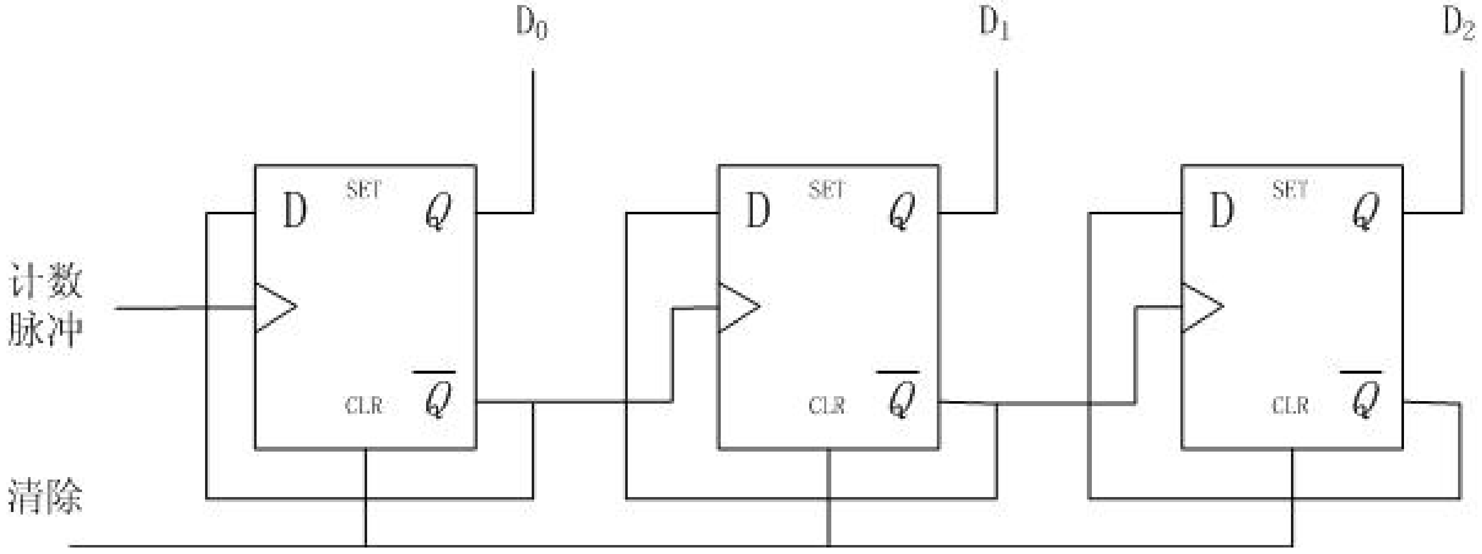
2.5 译码器



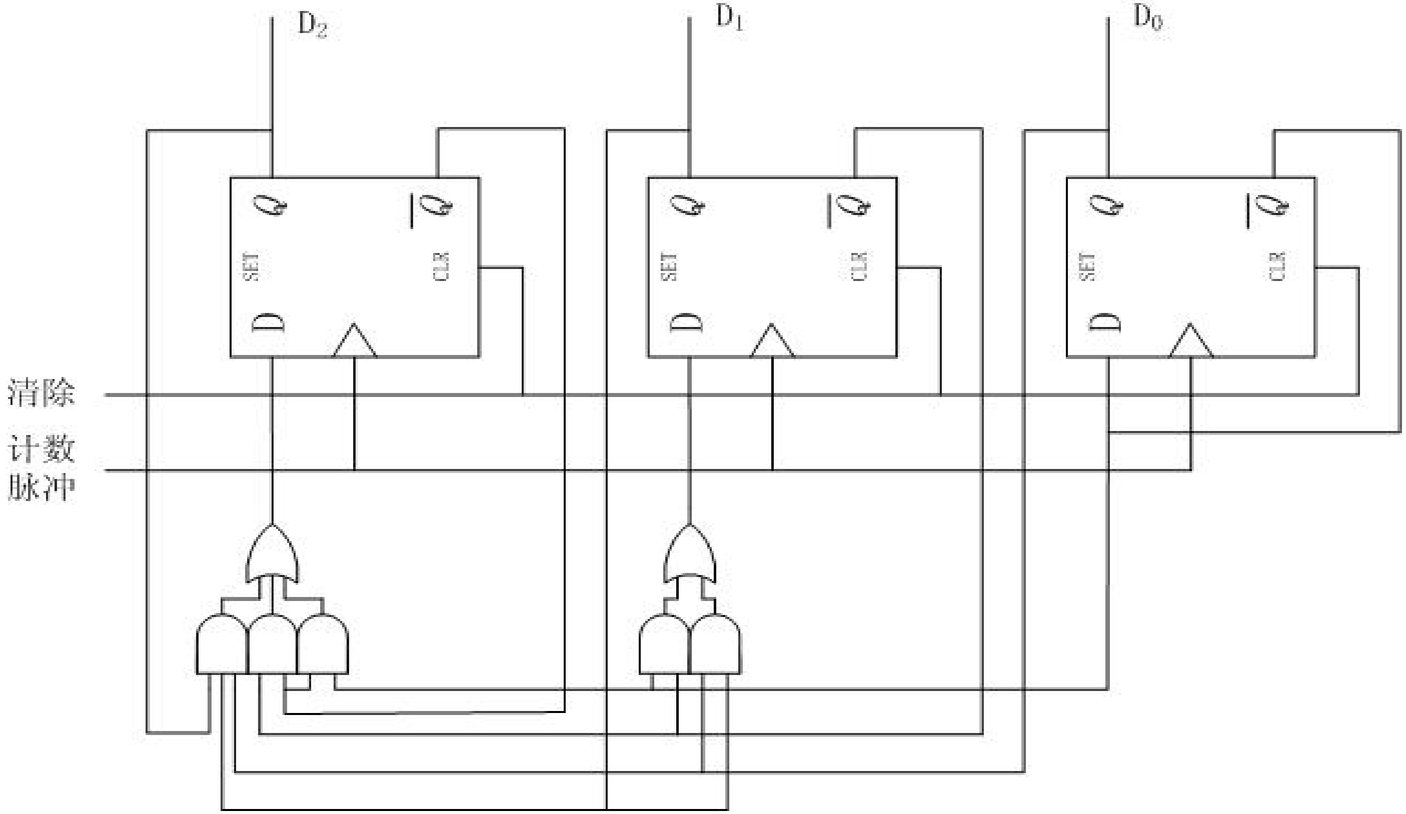
功能表

E	B	A	Y_0	Y_1	Y_2	Y_3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	X	X	1	1	1	1

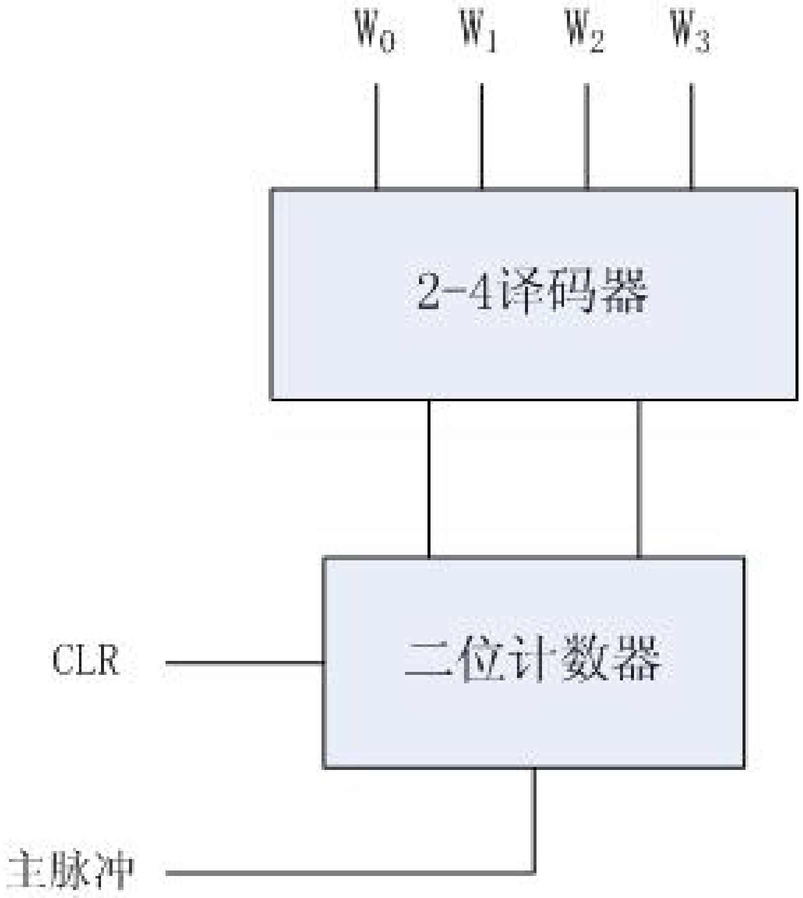
1.二进制异步计数器



2.二进制同步计数器



2.7 节拍分配器



2.8 总线

所谓的总线，就是连接计算机各部件的一族公共的信号线，即：计算机中各部件之间传送数据的公共通道。

总线一般有地址线、数据线和控制线组成。

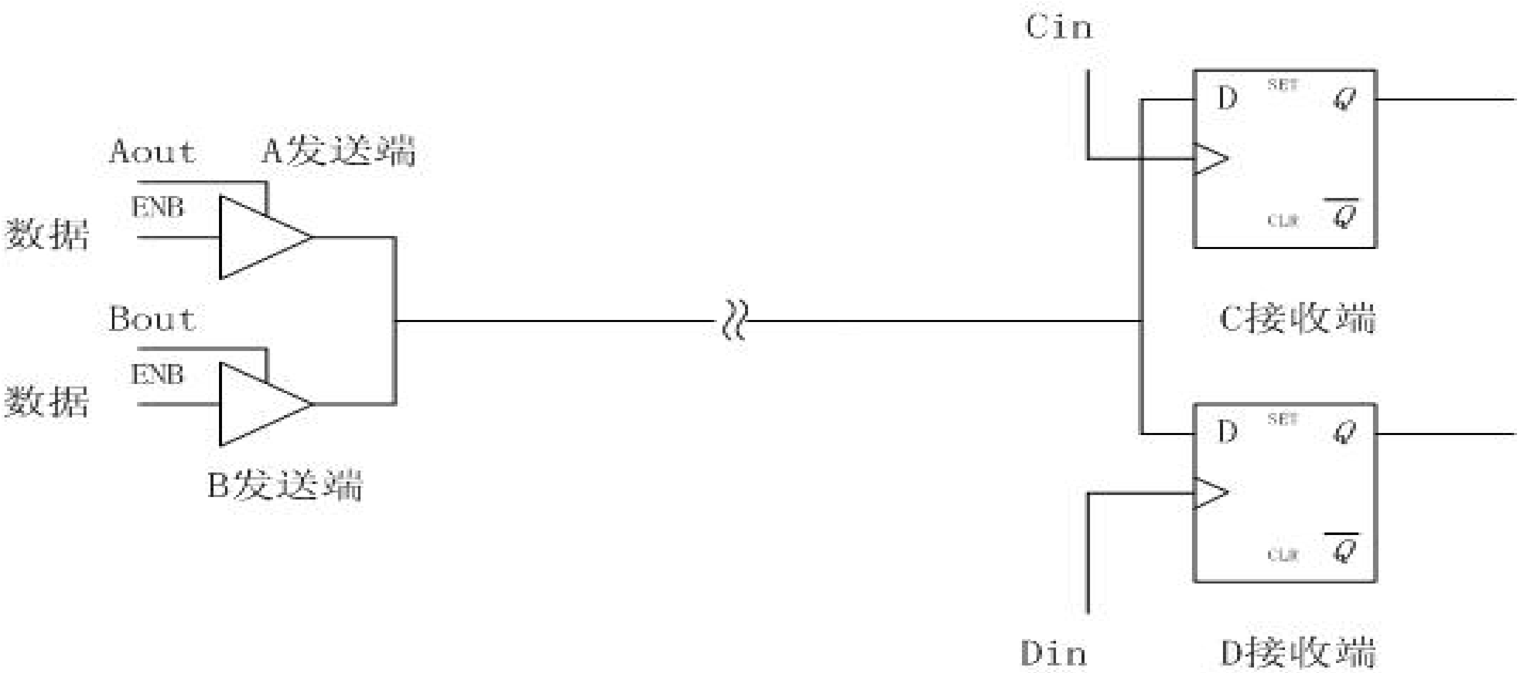
1.实现总线连接的线路

（1）集电极开路与非门（OC门）

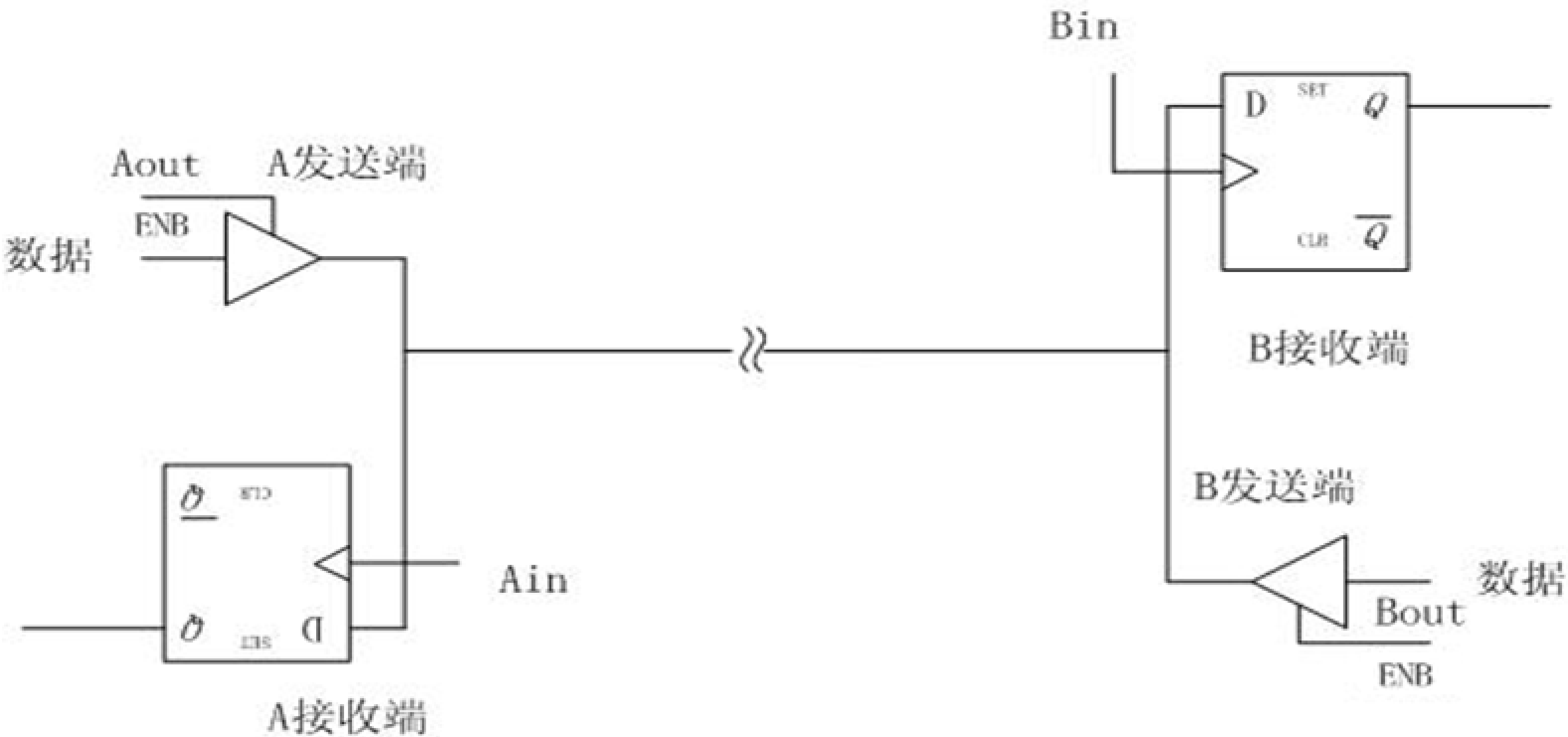
（2）三态门

2.总线的实现方法

（1）单向总线



（2）双向总线



1.单元加法电路

（1）半加器组成

只考虑本位的被加数和加数的相加，不考虑较低位来的进位。

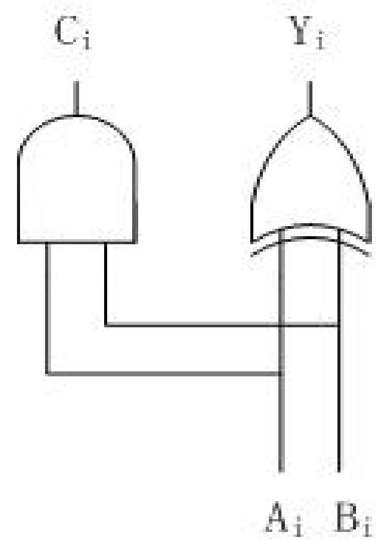
$Y_i = A_i \oplus B_i$

$C_i = A_i B_i$

Y_i—本位和

C_i—本位进位

$A_i B_i$—本位被加数和加数



2.9 二进制加法器

(2) 全加器组成

不但考虑本位的被加数和加数的相加，还考虑较低位来的进位。

$S_i = A_i \oplus B_i \oplus C_{i-1}$

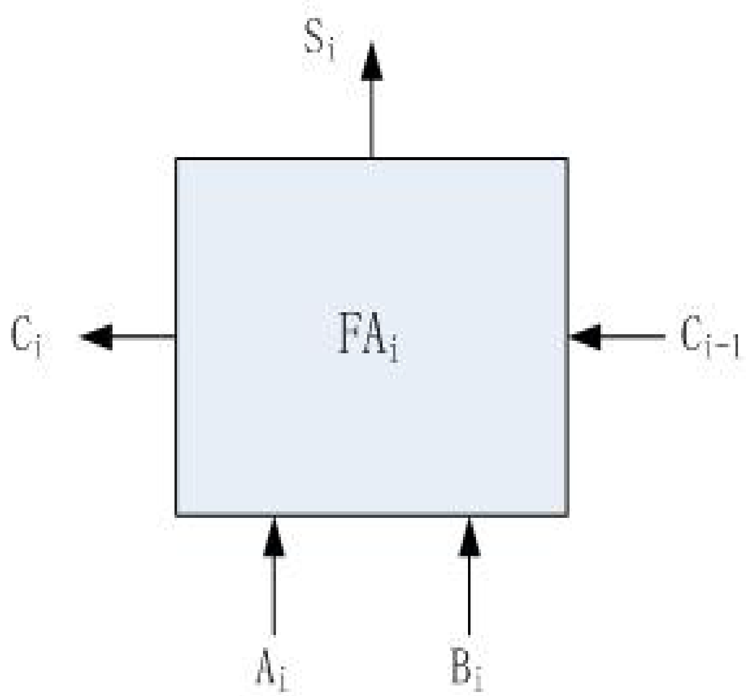
$C_i = (A_i \oplus B_i) C_{i-1} + A_i B_i$

S_i—本位和

C_i—本位进位

$A_i B_i$—本位被加数和加数

C_{i-1} —低位进位

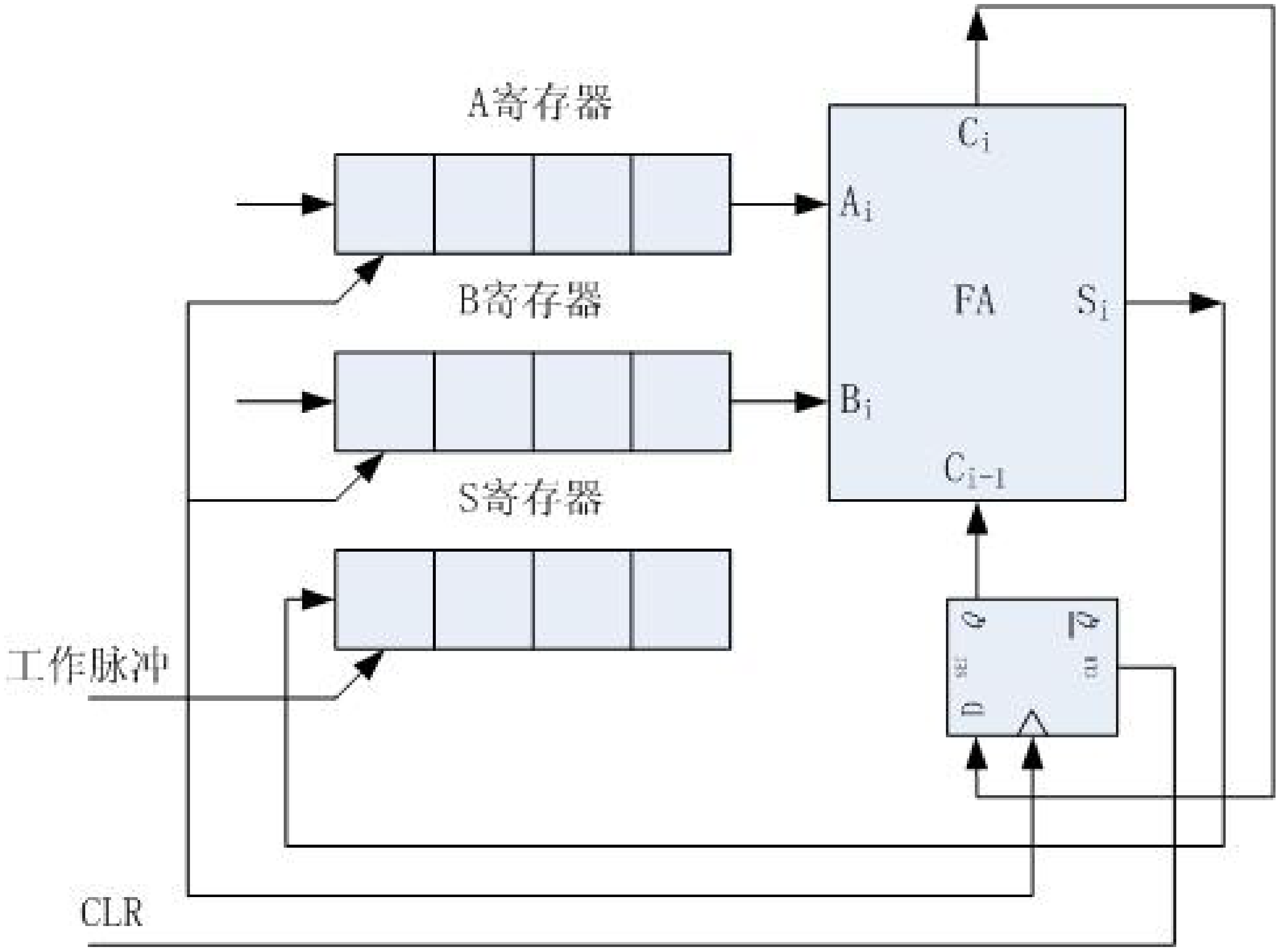


2. n位加法器的组成

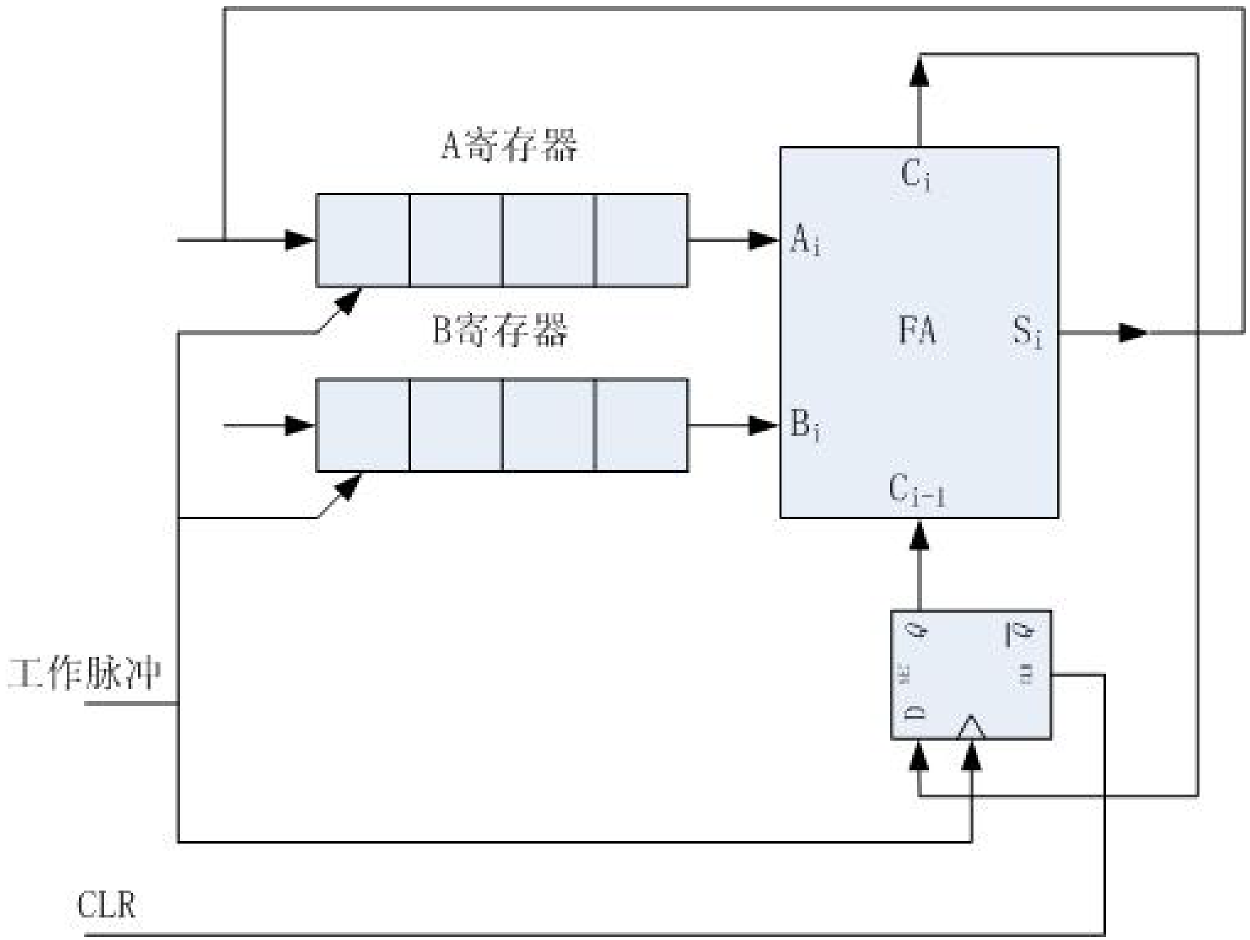
（1）串行加法器

所谓的串行加法器，是指两个二进制数相加的过程是从低位至高位逐位进行，同人工进行两个多位数相加的过程一样。

2.9 二进制加法器

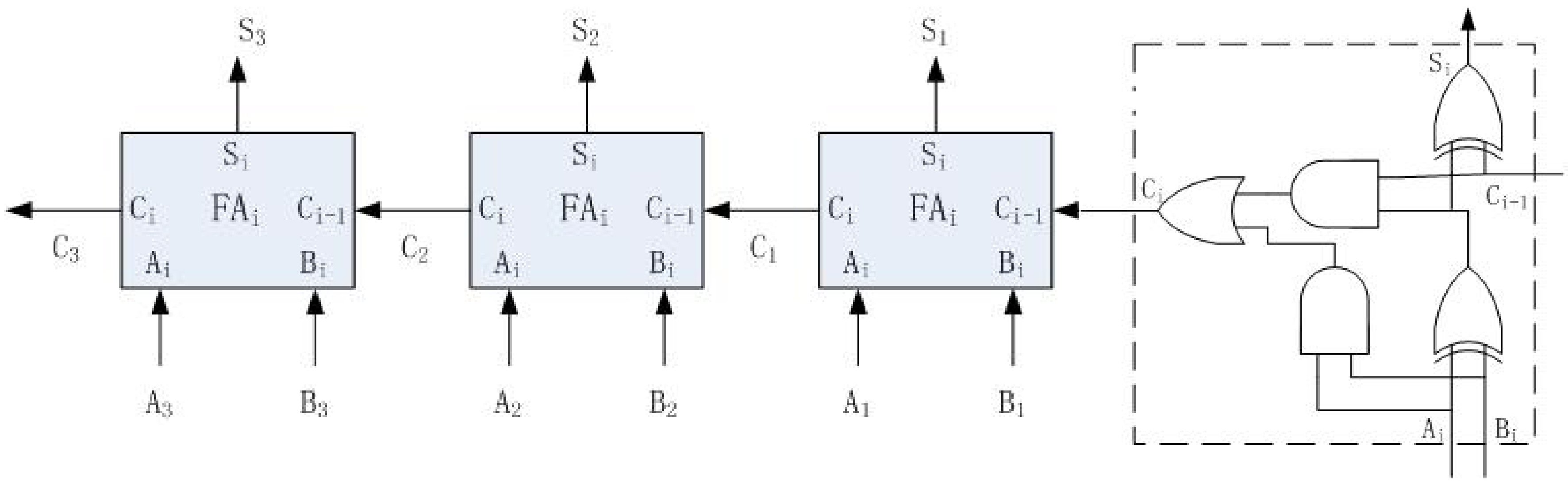


2.9 二进制加法器

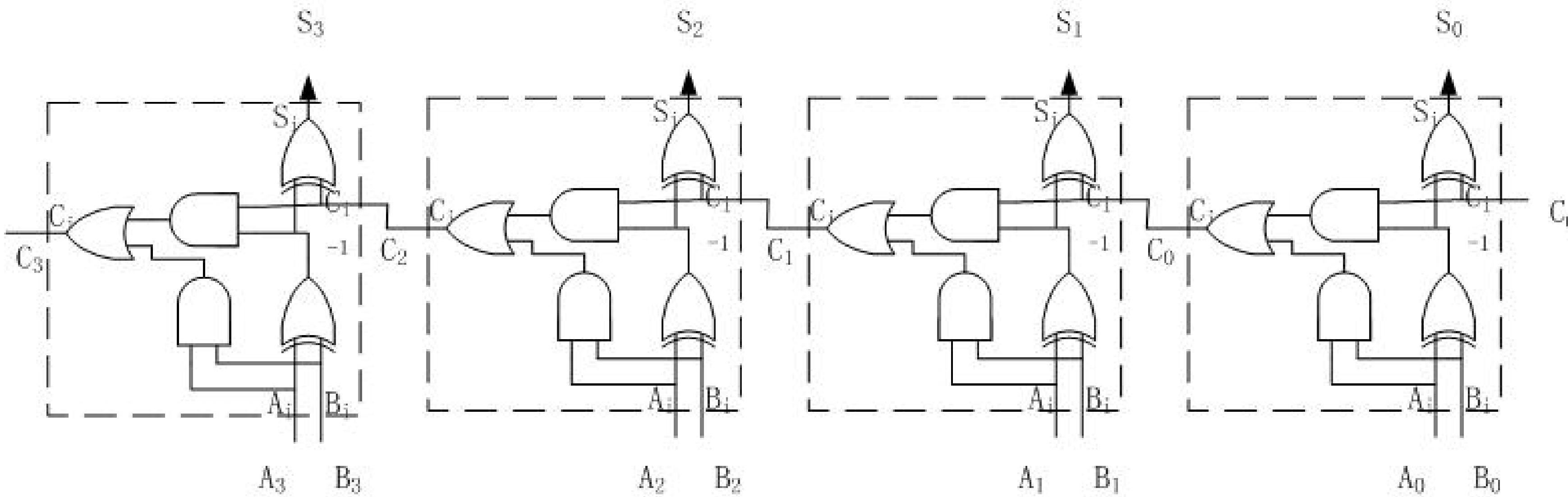


2.9 二进制加法器

(2) 并行加法器



2.9 二进制加法器



$S_3=A_3\oplus B_3\oplus C_2$　　$S_2=A_2\oplus B_2\oplus C_1$　　$S_1=A_1\oplus B_1\oplus C_0$　　$S_0=A_0\oplus B_0\oplus C_m$

$C_3=A_3B_3+(A_3\oplus B_3)C_2$　　$C_2=A_2B_2+(A_2\oplus B_2)C_1$　　$C_1=A_1B_1+(A_1\oplus B_1)C_0$　　$C_0=A_0B_0+(A_0\oplus B_0)C_m$

百年同济 TONGJI UNIVERSITY

2.10 进位链

由n个全加器对两个加数的各位分别进行运算，低位全加器的进位输出Cm接相邻高位全加器的进位输入Ci。各位加法器只对本位的输入进行运算，高位的进位输入必须等待低位运算结束后逐级传输，所以运算速度受位数影响。

2.10 进位链

1.同时进位

$$S_0=A_0 \oplus B_0 \oplus C_m \;;\quad C_0=A_0B_0+(A_0 \oplus B_0)C_m$$

$$S_1=A_1 \oplus B_1 \oplus C_0 \;;\quad C_1=A_1B_1+(A_1 \oplus B_1)C_0$$

$$S_2=A_2 \oplus B_2 \oplus C_1 \;;\quad C_2=A_2B_2+(A_2 \oplus B_2)C_1$$

$$S_3=A_3 \oplus B_3 \oplus C_2 \;;\quad C_3=A_3B_3+(A_3 \oplus B_3)C_2$$

设：传递函数 $P_i=A_i \oplus B_i$

进位产生函数 $G_i=A_iB_i$

2.10 进位链

$$C_0=A_0B_0+(A_0\oplus B_0)C_m;$$
$$C_1=A_1B_1+(A_1\oplus B_1)C_0;$$
$$C_2=A_2B_2+(A_2\oplus B_2)C_1;$$
$$C_3=A_3B_3+(A_3\oplus B_3)C_2;$$

$$P_i=A_i\oplus B_i$$
$$G_i=A_iB_i$$

$$C_0=G_0+P_0C_m$$
$$C_1=G_1+P_1C_0$$
$$C_2=G_2+P_2C_1$$
$$C_3=G_3+P_3C_2$$

2.10 进位链

$C_0=G_0+P_0C_m$
$C_1=G_1+P_1C_0$
$C_2=G_2+P_2C_1$
$C_3=G_3+P_3C_2$

$C_0=G_0+P_0C_m$

$C_1=G_1+P_1C_0=G_1+P_1(G_0+P_0C_m)=G_1+P_1G_0+P_1P_0C_m$

$C_2=G_2+P_2C_1=G_2+P_2(G_1+P_1G_0+P_1P_0C_m)$
$= G_2+P_2G_1+P_2P_1G_0+P_2P_1P_0C_m$

$C_3=G_3+P_3C_2=G_3+P_3(G_2+P_2G_1+P_2P_1G_0+P_2P_1P_0C_m)$
$=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3P_2P_1P_0C_m$

2.10 进位链

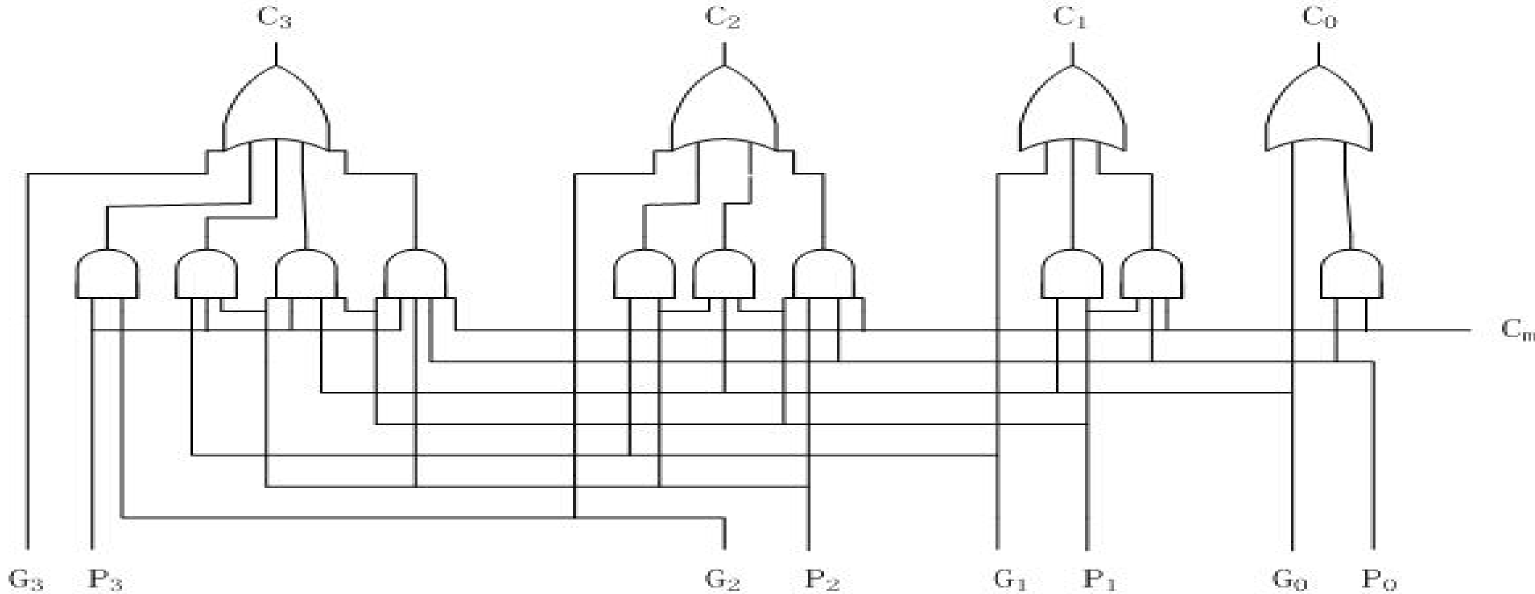
$$C_0=G_0+P_0C_m$$

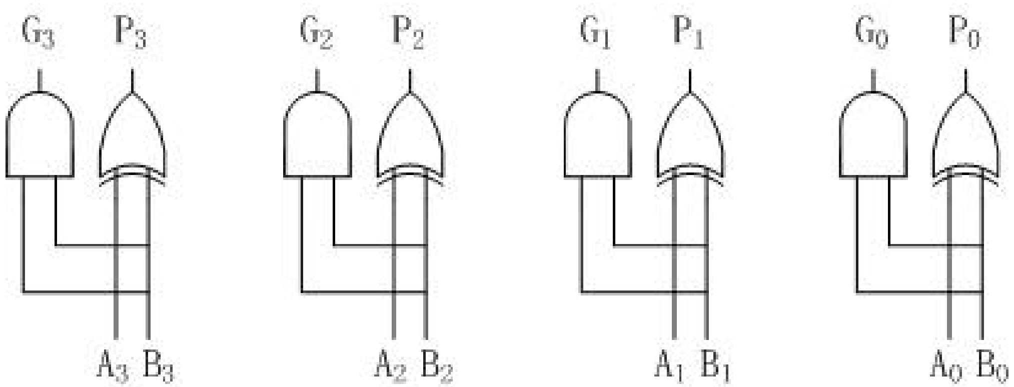
$$C_1=G_1+P_1G_0+P_1P_0C_m$$

$$C_2=G_2+P_2G_1+P_2P_1G_0+P_2P_1P_0C_m$$

$$C_3=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3P_2P_1P_0C_m$$

2.10 进位链





传递函数

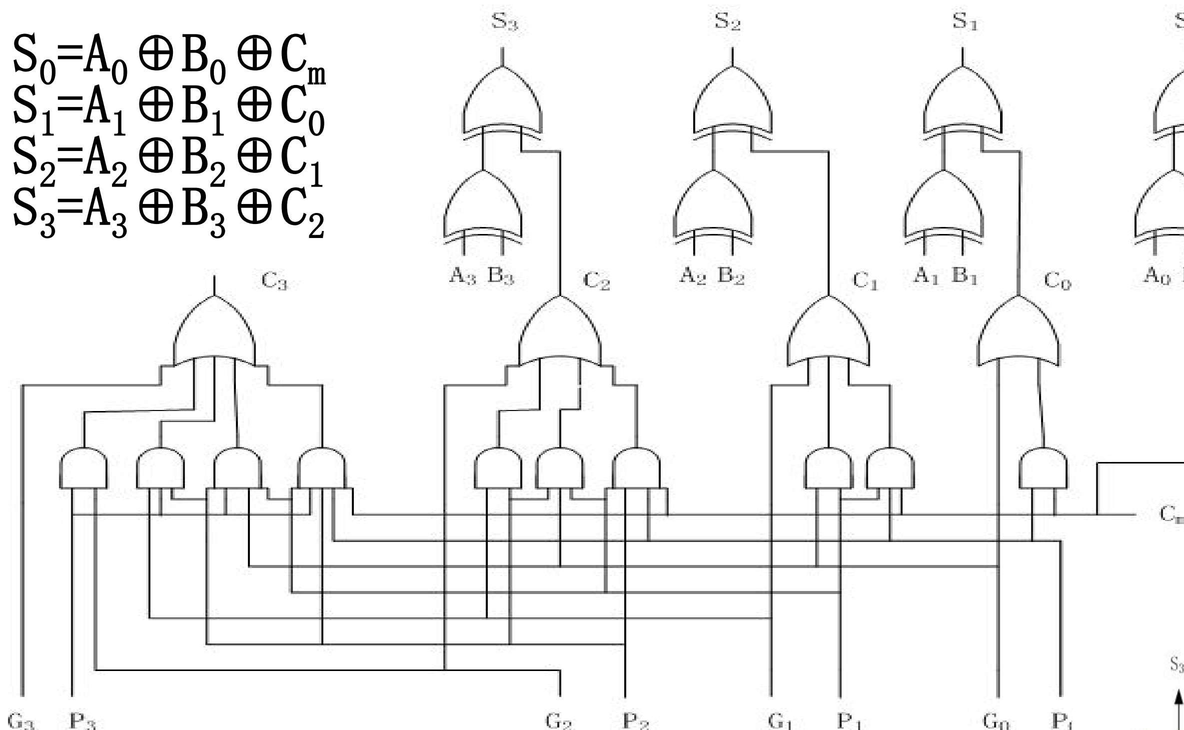
$P_i = A_i \oplus B_i$

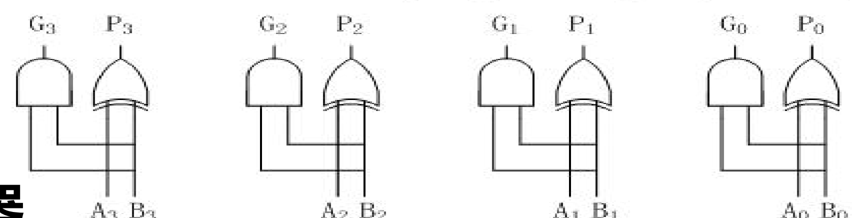
进位产生函数

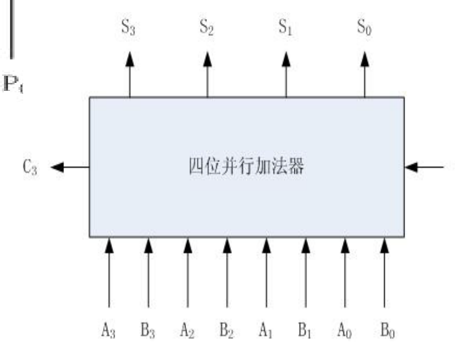
$G_i = A_i B_i$

2.10 进位链

$S_0=A_0 \oplus B_0 \oplus C_m$
$S_1=A_1 \oplus B_1 \oplus C_0$
$S_2=A_2 \oplus B_2 \oplus C_1$
$S_3=A_3 \oplus B_3 \oplus C_2$



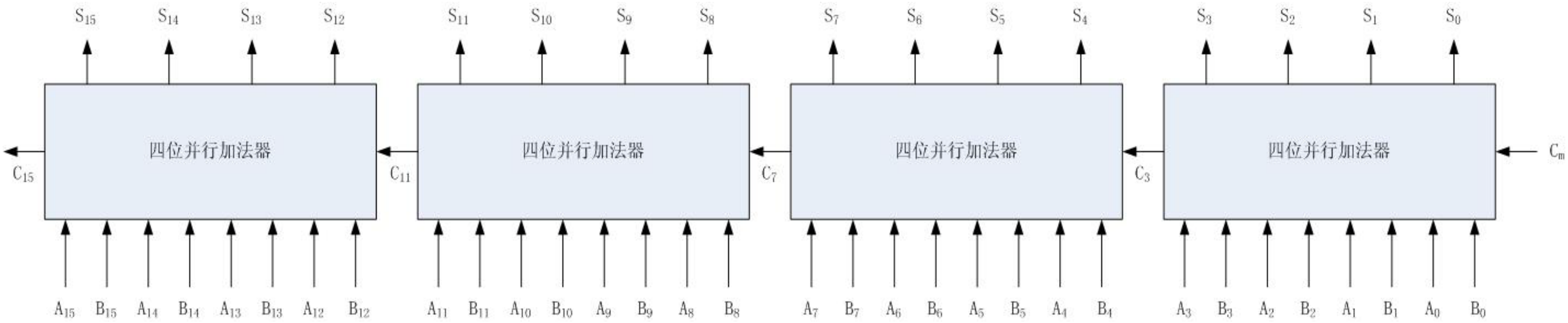




四位并行进位加法器

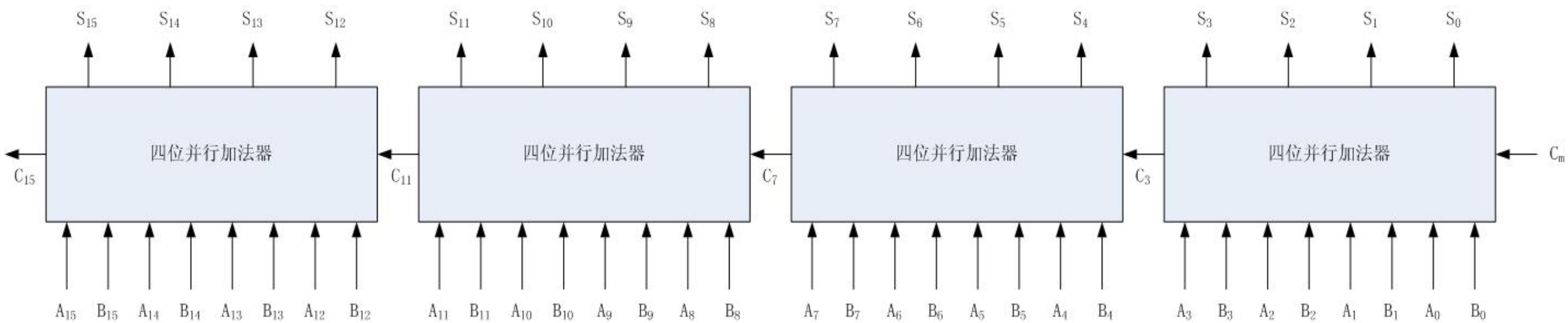
2.单级分组跳跃进位

构成一个十六位的加法器



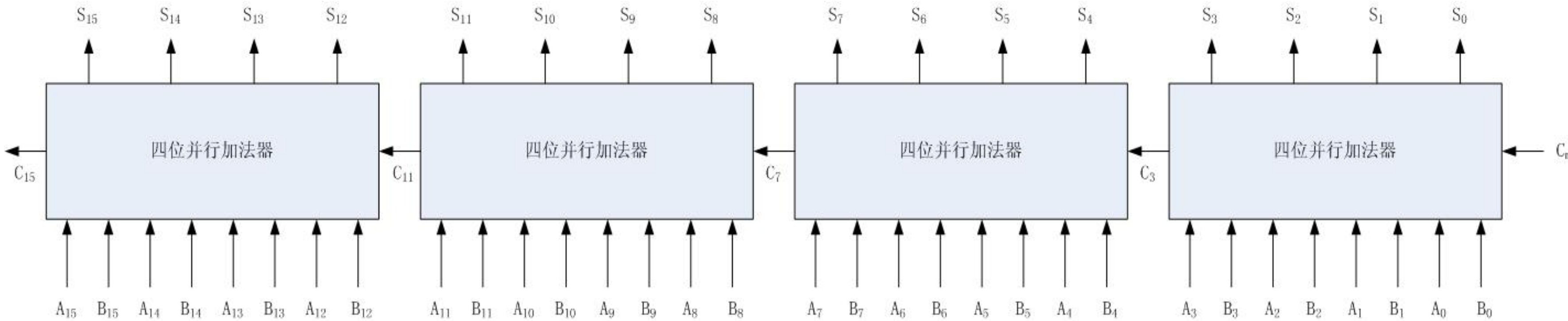
2. 10 进位链

3.多级分组跳跃进位



$$C_3=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3P_2P_1P_0C_m$$
$$C_7=G_7+P_7G_6+P_7P_6G_5+P_7P_6P_5G_4+P_7P_6P_5P_4C_3$$
$$C_{11}=G_{11}+P_{11}G_{10}+P_{11}P_{10}G_9+P_{11}P_{10}P_9G_8+P_{11}P_{10}P_9P_8C_7$$
$$C_{15}=G_{15}+P_{15}G_{14}+P_{15}P_{14}G_{13}+P_{15}P_{14}P_{13}G_{12}+P_{15}P_{14}P_{13}P_{12}C_{11}$$

3.多级分组跳跃进位



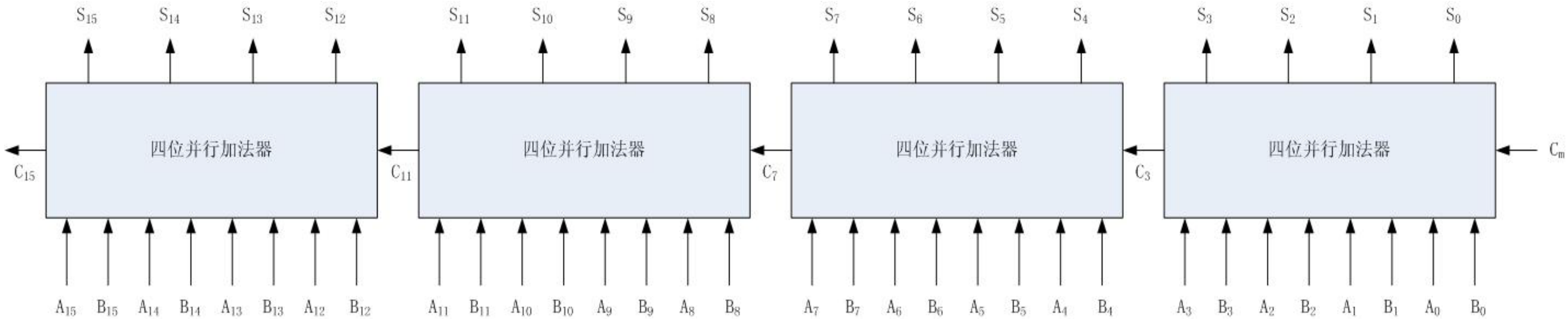
$$C_3=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3P_2P_1P_0C_m$$

$$C_7=G_7+P_7G_6+P_7P_6G_5+P_7P_6P_5G_4+P_7P_6P_5P_4C_3$$

$$C_{11}=G_{11}+P_{11}G_{10}+P_{11}P_{10}G_9+P_{11}P_{10}P_9G_8+P_{11}P_{10}P_9P_8C_7$$

$$C_{15}=G_{15}+P_{15}G_{14}+P_{15}P_{14}G_{13}+P_{15}P_{14}P_{13}G_{12}+P_{15}P_{14}P_{13}P_{12}C_{11}$$

3.多级分组跳跃进位



$C_3=D_0+T_0C_m$　　$D_0=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0$　　$T_0=P_3P_2P_1P_0$

$C_7=D_1+T_1C_3$　　$D_1=G_7+P_7G_6+P_7P_6G_5+P_7P_6P_5G_4$　　$T_1=P_7P_6P_5P_4$

$C_{11}=D_2+T_2C_7$　　$D_2=G_{11}+P_{11}G_{10}+P_{11}P_{10}G_9+P_{11}P_{10}P_9G_8$　　$T_2=P_{11}P_{10}P_9P_8$

$C_{15}=D_3+T_3C_{11}$　　$D_3=G_{15}+P_{15}G_{14}+P_{15}P_{14}G_{13}+P_{15}P_{14}P_{13}G_{12}$　　$T_3=P_{15}P_{14}P_{13}P_{12}$

2.10 进位链

$C_3=D_0+T_0C_m$

$C_7=D_1+T_1C_3$

$C_{11}=D_2+T_2C_7$

$C_{15}=D_3+T_3C_{11}$

$C_3=D_0+T_0C_m$

$C_7=D_1+T_1D_0+T_1T_0C_m$

$C_{11}=D_2+T_2D_1+T_2T_1D_0+T_2T_1T_0C_m$

$C_{15}=D_3+T_3D_2+T_3T_2D_1+T_3T_2T_1D_0+T_3T_2T_1T_0C_m$

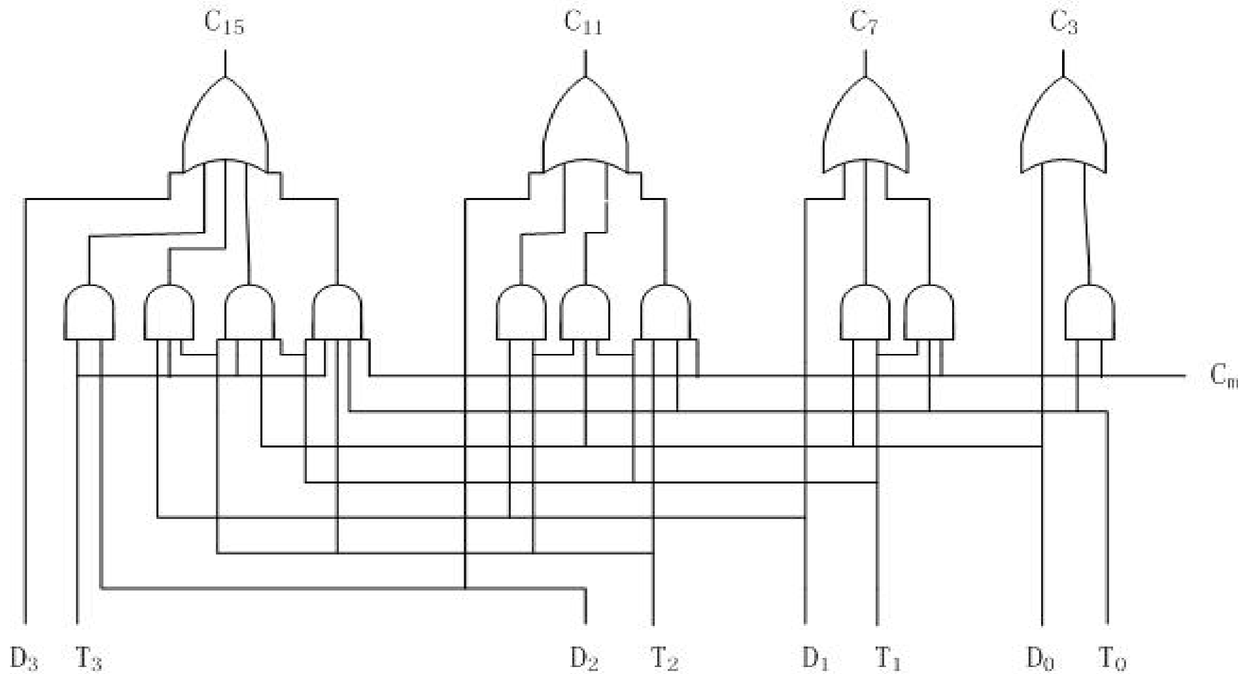
2.10 进位链

$C_3=D_0+T_0C_m$

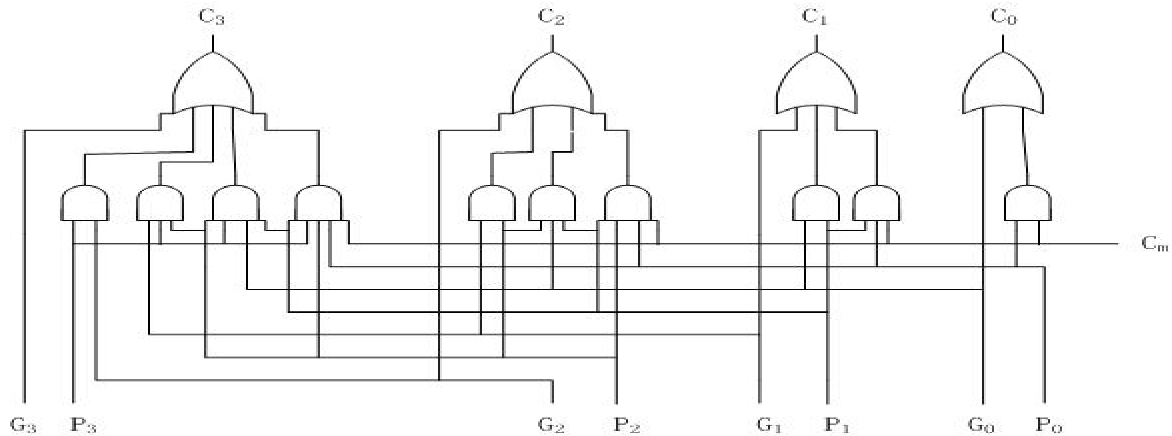
$C_7=D_1+T_1D_0+T_1T_0C_m$

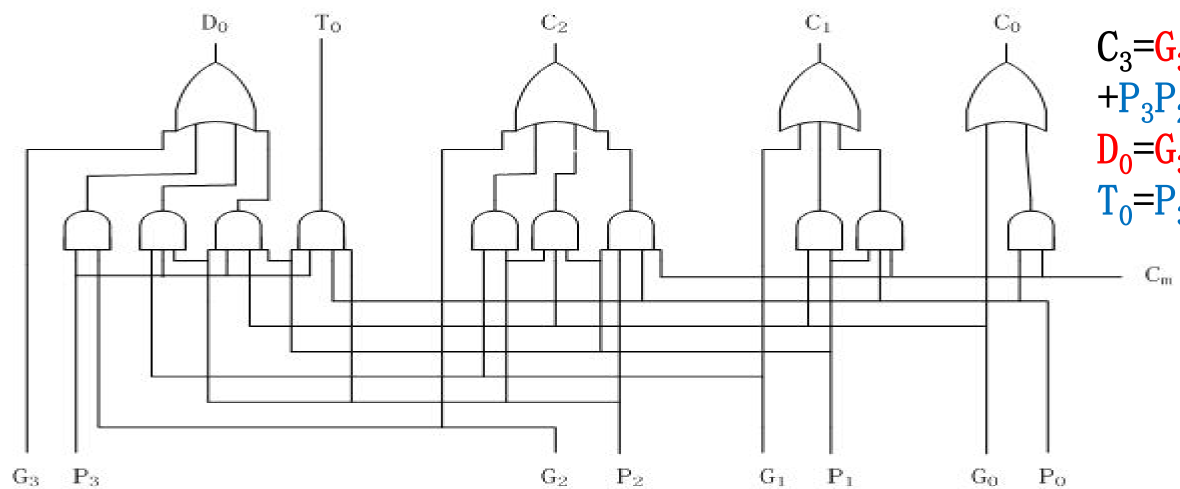
$C_{11}=D_2+T_2D_1+T_2T_1D_0+T_2T_1T_0C_m$

$C_{15}=D_3+T_3D_2+T_3T_2D_1+T_3T_2T_1D_0+T_3T_2T_1T_0C_m$



2.10 进位链



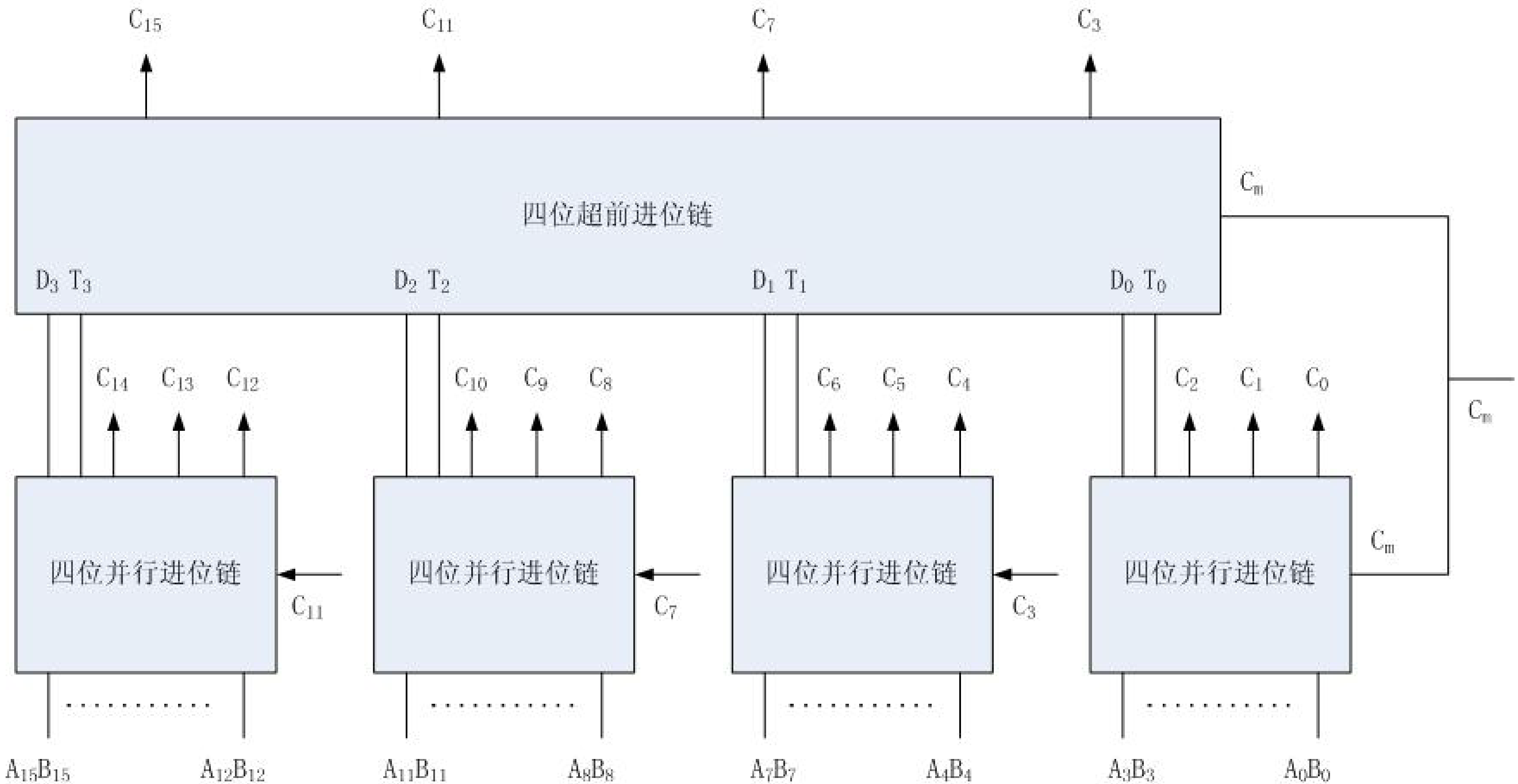


$C_3=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0$
$+P_3P_2P_1P_0C_m$

$D_0=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0$

$T_0=P_3P_2P_1P_0$

2.10 进位链



C_{15}
C_{11}
C_7
C_3
四位超前进位链
C_m
D_3 T_3
D_2 T_2
D_1 T_1
D_0 T_0
C_{14} C_{13} C_{12}
C_{10} C_9 C_8
C_6 C_5 C_4
C_2 C_1 C_0
C_m
四位并行进位链
C_{11}
四位并行进位链
C_7
四位并行进位链
C_3
四位并行进位链
C_m
$A_{15}B_{15}$
$A_{12}B_{12}$
$A_{11}B_{11}$
A_8B_8
A_7B_7
A_4B_4
A_3B_3
A_0B_0

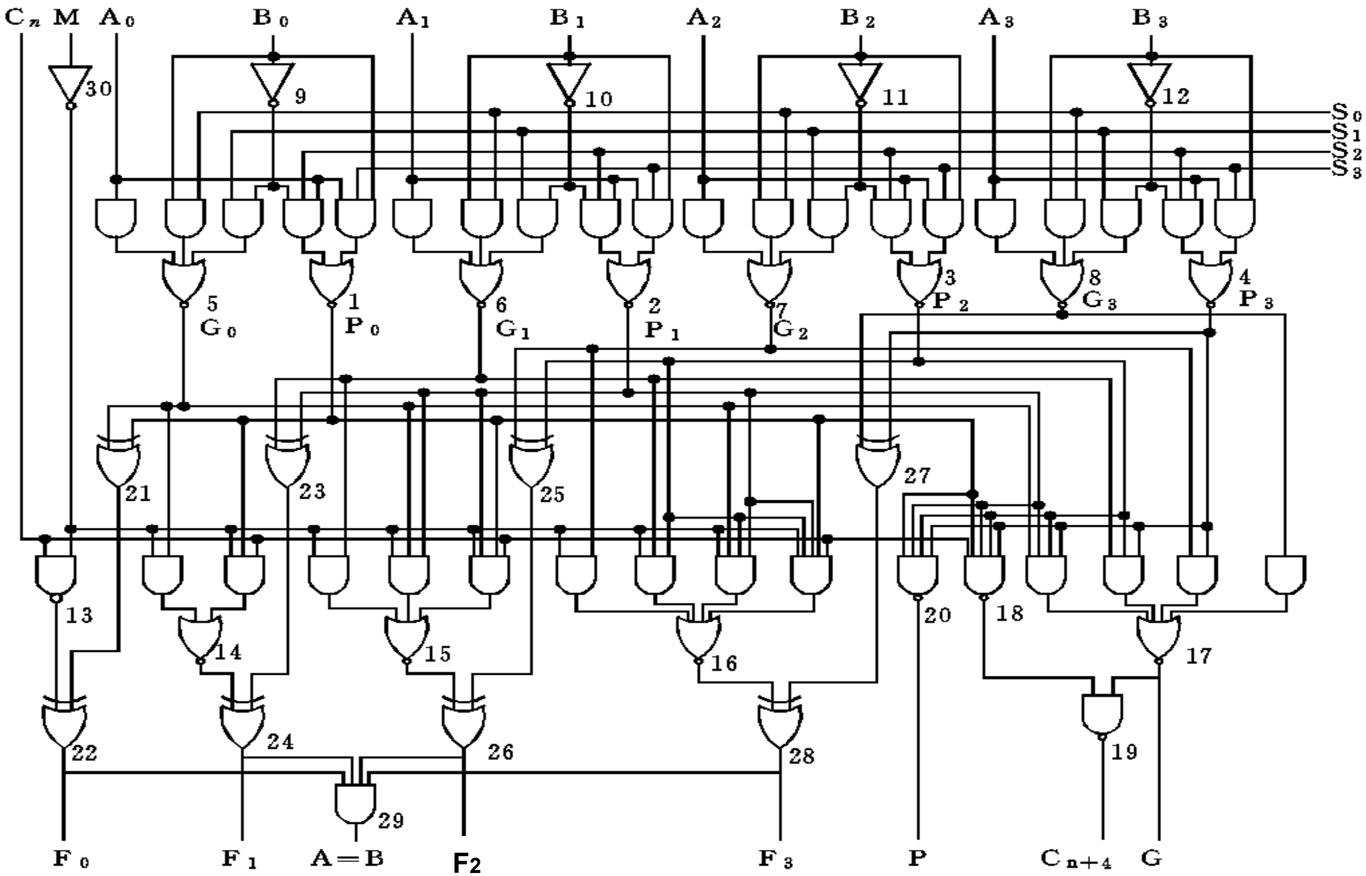
2.11 SN74181

1.四位ALU SN74181

SN74181型ALU逻辑图及其在正逻辑下的功能表，在功能表中，“加”表示算术加，“+”表示逻辑加。它能执行16种算术运算和16种逻辑运算，M是状态控制端，当M=H，执行逻辑运算；M=L，执行算术运算，S_0-S_3是运算选择控制端，它决定电路执行哪种算术运算或哪种逻辑运算。A_3～A_0，B_3～B_0是参加运算的两个数，C_n是ALU的最低位进位输入，F_3～F_0是运算结果，注：脚3表示最高位。

百年同济
1907-2007
TONGJI UNIVERSITY

2.11 SN74181



(a) 逻辑图

2.11 SN74181

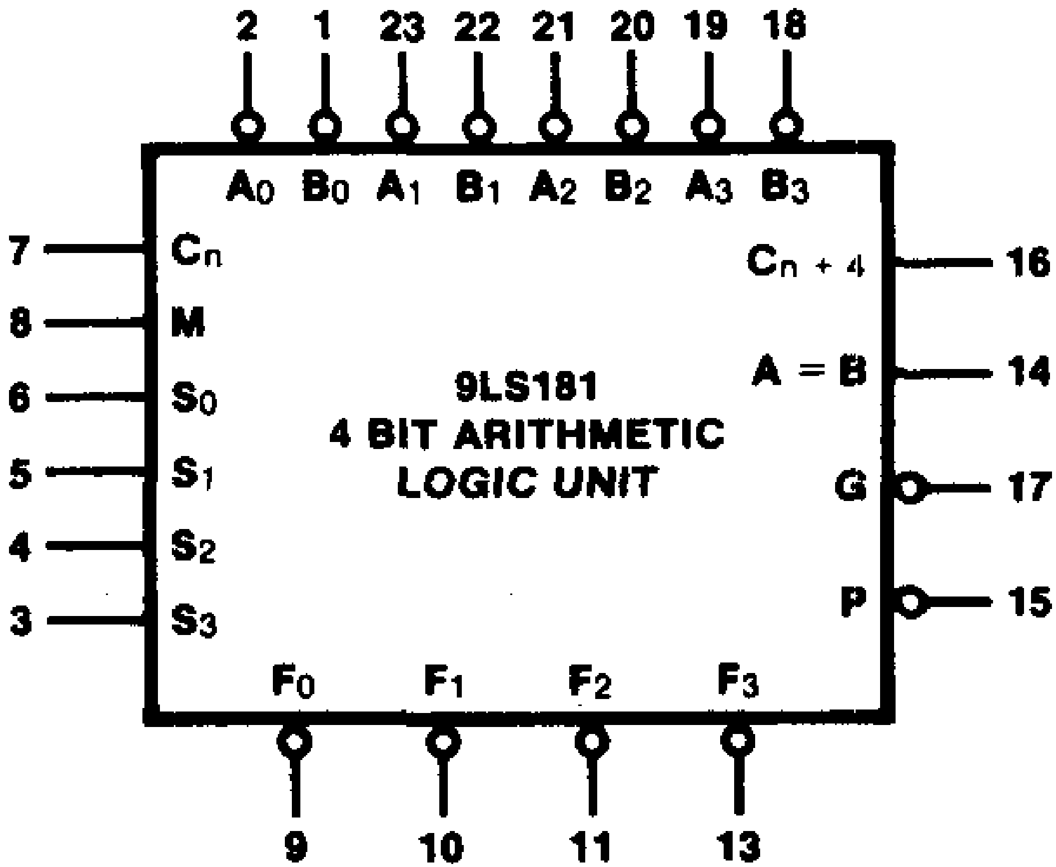
S_3	S_2	S_1	S_0	正 逻 辑		
				M=H 逻辑运算	M=L 算术运算	
					C_n=1	C_n=0
L	L	L	L	A	A	A+1
L	L	L	H	$\overline{A+B}$	A+B	(A+B)加1
L	L	H	L	$\overline{A}\bullet B$	A+B	(A+B)加1
L	L	H	H	0	减1	“0”
L	H	L	L	$\overline{A\bullet B}$	A加(A·B)	A加(A·B)加1
L	H	L	H	B	(A·B)加(A+B)	(A·B)加(A+B)加1
L	H	H	L	$A\oplus B$	A减B减1	A减B
L	H	H	H	A·B	(A·B)减1	A·

2.11 SN74181

S_3	S_2	S_1	S_0	正 逻 辑		
				M=H 逻辑运算	M=L 算术运算	
					C_n=1	C_n=0
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·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+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

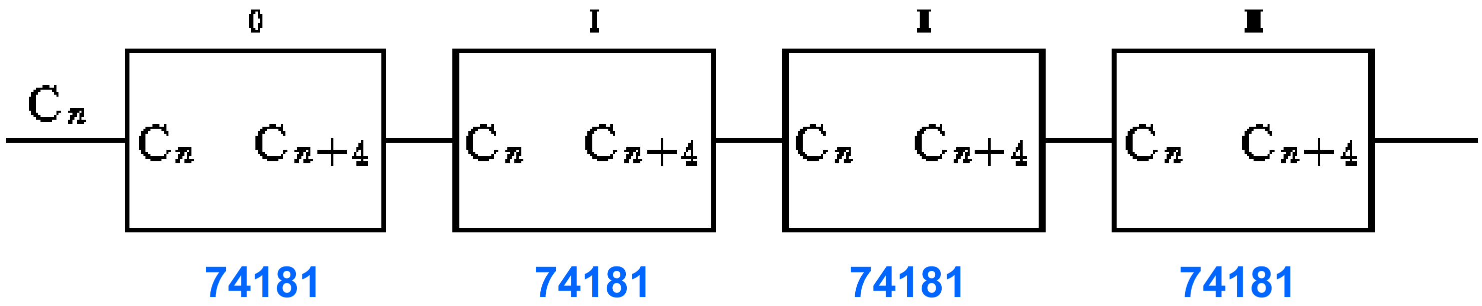
百年同济
TONGJI UNIVERSITY

2.11 SN74181

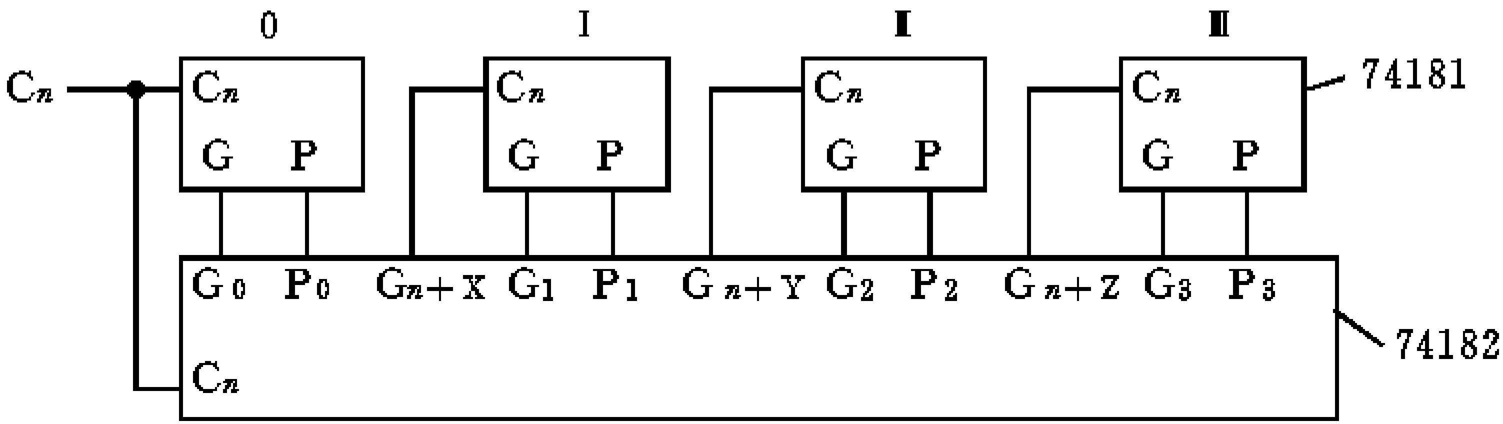


2.11 SN74181

2.构成16 位 ALU



2.11 SN74181


0
I
II
III
Cn
Cn
G P
Cn
G P
Cn
G P
Cn
G P
74181
G0 P0 Gn+X G1 P1 Gn+Y G2 P2 Gn+Z G3 P3
Cn
74182

习 题（2）

P30
习题：3，6，7