2.5 数据的校验

（1）奇偶校验

（2）海明校验

（3）循环冗余校验

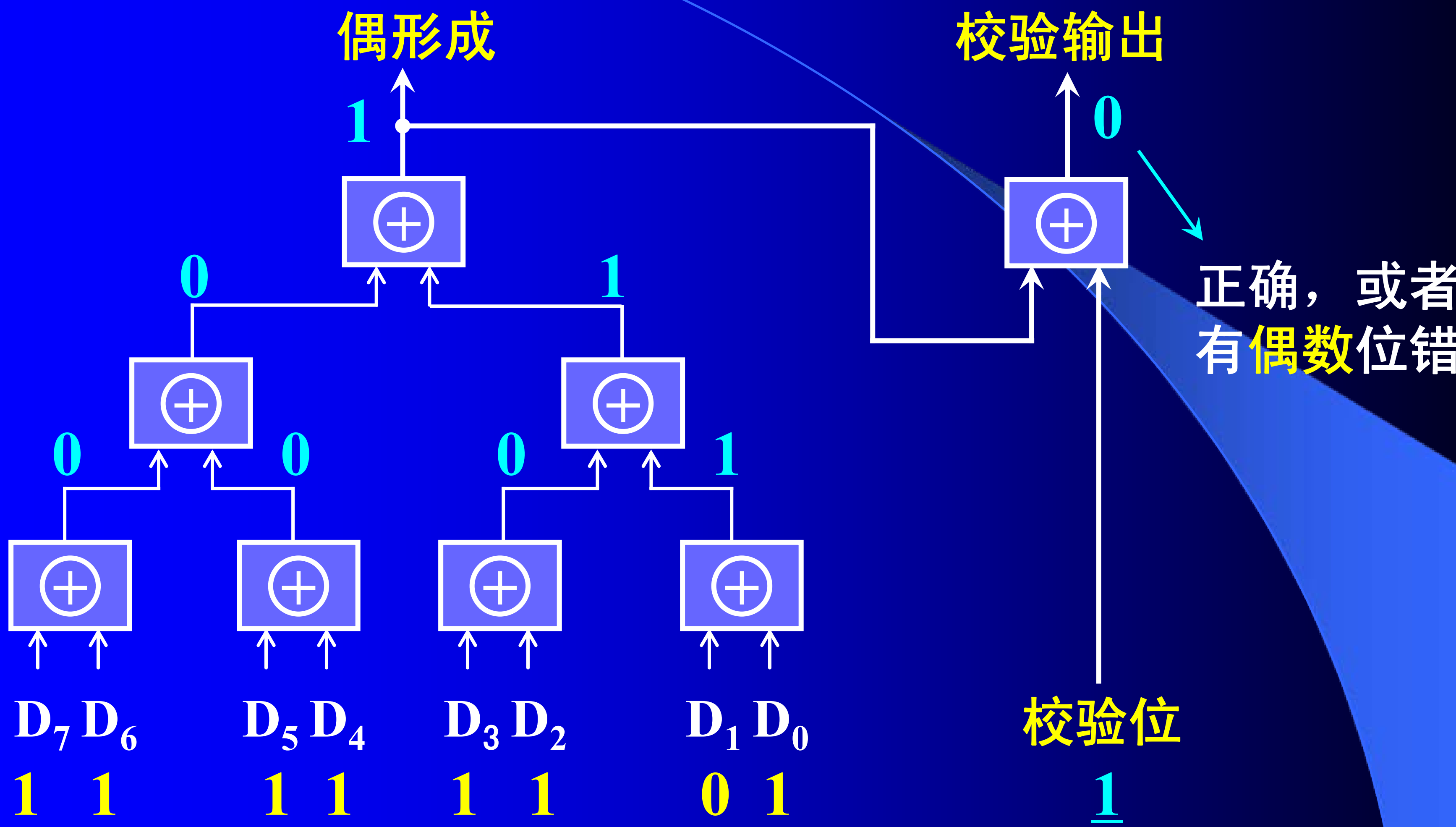
2.5.1 奇偶校验

（1）编码规则

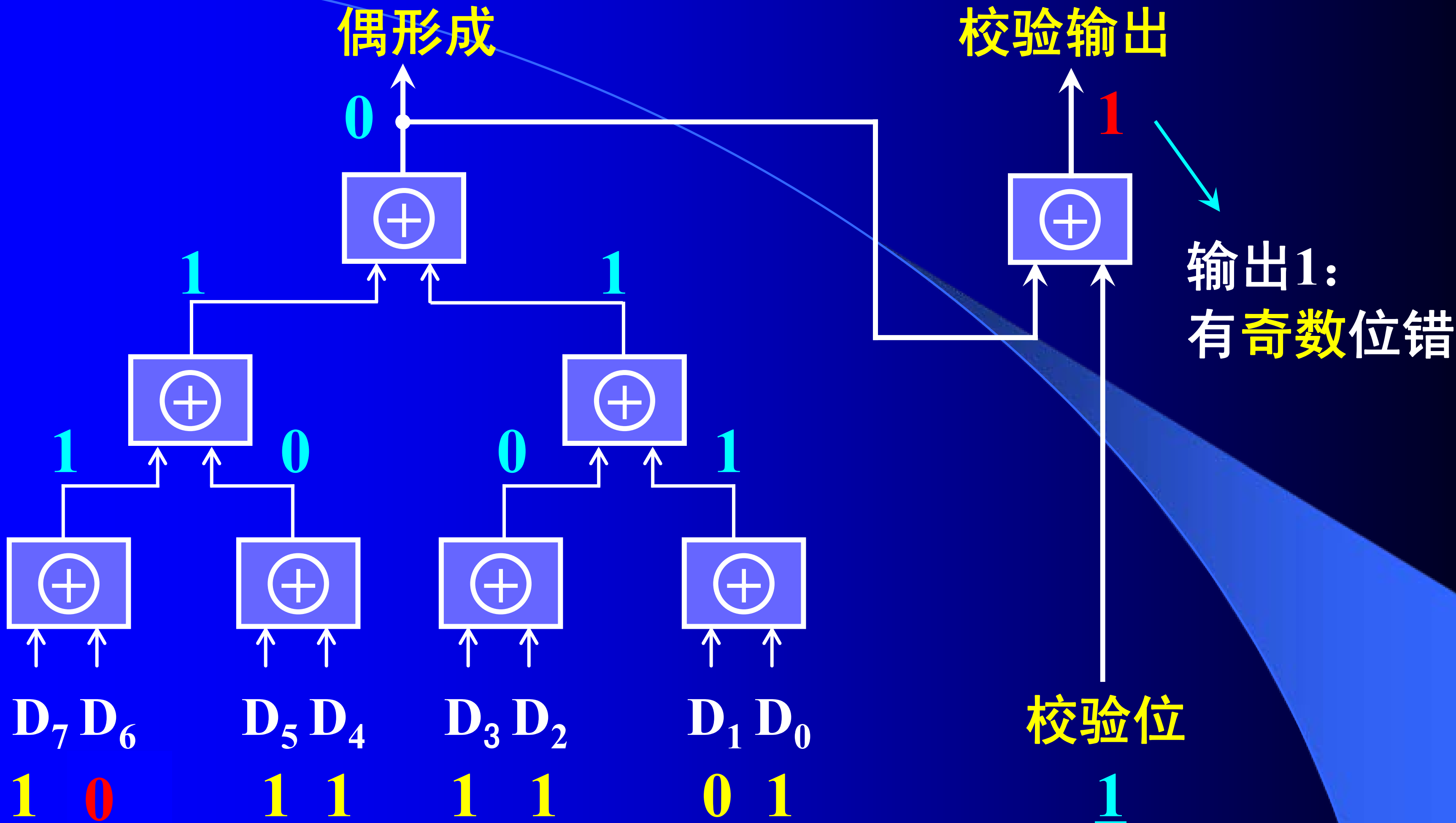
增设1位校验位，从而使1的个数是奇或偶数

待编码信息	1011 0001		
奇校验编码	1011 0001	1	→ 5个“1”
偶校验编码	1011 0001	0	→ 4个“1”
		↓ 校验位	

（2）偶校验电路逻辑



偶校验逻辑电路图



※任何1个数，减去1个偶数，其奇偶性不变

所以奇偶校验均不能发现偶数位错，也无法定位错误

2.5.2 海明校验

√是一种多重分组奇偶校验；

√将代码组织为若干分组，每组进行奇偶校验；

√能够检验是否出错，也能定位出错位；

（1）分成几组？每组包含多少校验位？

[假设]待编码信息K位，分成r组，每组1个校验位

校验码位数:r位;

海明编码总长:N=k+r位;

海明编码时：各组单独进行奇偶校验编码，以确定各组的校验位。

代码检验时：每组能产生1个指误码

→ r位指误码 比如：$G_3G_2G_1G_0$

→ 2^r种可能的指误代码 如0000、0001、0010、….

指误码为全0 ←→ 海明编码无错；

其余(2^r-1)种指误代码：

→分别用于指示(2^r-1)种只有1位错的情况

※各参数应满足：$N = k+r \leqslant 2^r-1$

若$k = 4$, 则 $r \geqslant 3$满足上述定理，可组成7位海明码。

（2）分组方法

有效信息：$A_1A_2A_3A_4$,校验位：$P_1P_2P_3$, 偶校验方式

	1	2	3	4	5	6	7	指误码
	P_1	P_2	A_1	P_3	A_2	A_3	A_4	
第3组				√	√	√	√	G_3
第2组		√	√			√	√	G_2
第1组	√		√		√		√	G_1
数据码	1	0	1	1	0	1	0	$G_3G_2G_1=000$
1位错	1	0	1	1	1	1	0	$G_3G_2G_1=101$

（k=4,r=3）编码的海明距离：d=3；　　第5位错 ← 5

如果有多位数字发生错误呢？需增加分组扩大码距！

（3）编码规则

每组均采用偶校验，填入校验码，组内具有偶数个1

[例如]4比特有效码(待编码数据)1001

(k=4,r=3)海明编码：0011001（黄色的为检验码）

（4）检错与纠错

[例如]读取到数据：0011011

指误码：第3组0011011 → $G_3=1$

第2组0011011 → $G_2=1$

第1组0011110 → $G_1=0$

$G_3G_2G_1=110_2=6_{10}$

据此推断第6位出错，0011011 → 0011001 定位+纠错

增加分组，能提高检错和纠错能力

2.5.3 循环冗余校验

即CRC, Cyclic Redundancy Check;

[校验原理]用待校验数据除以某个约定代码，能除尽则表明数据正确，否则通过循环移位校正出错位。

(1) 编码方法

①将待编码的k位有效数据M(x)左移r位得到全编码多项式$M(x)\cdot x^r$，空出r位，以装填r位余数;

②选取一个r+1位的生成多项式G(x)，对$M(x)\cdot x^r$进行模2除运算，得到商Q(x)和余数R(x)的代码;

③将左移r位的待编码信息，与余数R(x)模2加，可拼接成为包含有效数据在内的CRC编码。

[例]将4位有效信息（1100）编成CRC码，使用生成多项式为代码（1011）

※CRC编码过程：

①先确定$G(x)$和$M(x)\cdot x^r$

待编码数据：1100（k=4），则$M(x)=\underline{1}x^3+\underline{1}x^2+\underline{0}x^1+\underline{0}$

$=x^3+x^2$

生成多项式代码：1011， 则$G(x)=\underline{1}x^3+\underline{0}x^2+\underline{1}x^1+\underline{1}$

$=x^3+x^1+1$

余数的位数r：等于G(x)中最高项的指数，r=3

得到：$M(x)\cdot x^r=(x^3+x^2)\cdot x^3=\underline{1}x^6+\underline{1}x^5+\underline{0}x^4+\underline{0}x^3+\underline{0}x^2+\underline{0}x^1+\underline{0}$

$=x^6+x^5 \leftrightarrow 1100\underline{000}$

②计算R(x)并编码

"模2运算"，可忽略进位、借位

$$\frac{M(x)\cdot x^3}{G(x)} = \frac{1100000}{1011}$$

```
          1110
     ┌────────
1011 │ 1100000
       1011
       ────
        1110
        1011
        ────
         1010
         1011
         ────
           10
```

$Q(x)=1110$

$R(x)=10$

$M(x)\cdot x^3 \oplus R(x) = 1100\underline{000} \oplus 10$

$\oplus$ ↓ 模2加

$= 1100\underline{010}$

1100在当前生成多项式G(x)下的CRC编码为：

1100010

（2）生成多项式G(x)的说明

不同的G(x)，产生不同的余数特性

①G(x)的最高位和最低位必须为1

②CRC码中任何1位出错，根据G(x)得到余数不为全0

③不同数位发生错误，G(x)得到的余数互不相同

④余数继续做模2运算，应能使余数循环出现

CRC-4: x^4+x+1 (ITU G.704)

CRC-8: $x^8+x^5+x^4+1$ (CCITT)

CRC-16: $x^{16}+x^{12}+x^5+1$ (CCITT)

CRC-32: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ (IEEE)

（3）CRC余数的特性分析（最多1位出错时）

有效数据：1100，G(x)：1011　　　$D_{6\sim0}$ / G(x)

	出错位	D_6	D_5	D_4	D_3	D_2	D_1	D_0	余数	
正确	无	1	1	0	0	0	1	0	000	
仅1位出错	D_0	1	1	0	0	0	1	1	001	=0010/1011
	D_1	1	1	0	0	0	0	0	010	=0100/1011
	D_2	1	1	0	0	1	1	0	100	=1000/1011
	D_3	1	1	0	1	0	1	0	011	=0110/1011
	D_4	1	1	1	0	0	1	0	110	=1100/1011
	D_5	1	0	0	0	0	1	0	111	=1110/1011
	D_6	0	1	0	0	0	1	0	101	

1010 / 1011 = 001（再次出现D_0出错时的余数）$T=7=2^3-1$ 数据位数

※CRC余数特性归纳：

【最多有1位数据出错时】

①余数为全0时，数据无错；

②余数非全0时，数据有错，且余数与出错位存在“一一对应”关系，余数001对应最低位出错的模式。

③相邻两个非0余数，对应的出错位也相邻；

④任何一个非0余数，循环执行余数低位补0并重新计算余数，余数会循环出现，对应的出错位也在随之循环左移，循环周期$T=2^r-1$。

[启发]利用这些特性，进行检错和纠错。

（4）CRC的检错和纠错

码制(n = 7，k = 4)、生成多项式G(x)=1011

这里n为CRC编码的总长，k为有效数据的位数

※CRC码中仅有1位数据出错时，余数与出错位的对应关系，只与码制(n, k)和G(x)相关。

[基本原理]

计算编码与G(x)模2运算的初始余数R：

①R为全0，表明数据无错

②R非全0，表明数据有错

有错时执行循环：余数补0计算新余数、CRC数据循环左移，新余数为001时D_0变反，R再次出现时停止。

[例1]若读取到数据1100011

1100011/1011 → 初始余数=<u>001</u>（非全0，且为001）

将D_0（即最低位）变反：$D_{6\sim0}$=110001<u>0</u>

循环终止

$D_{6\sim0}$=1100010　已修复的数据

[例2]若读取到数据1100110

1100110/1011→ 初始余数=100（非全0，不是001）

1	1000/1011→	余数011	1100110	← 循环左移	1001101
2	0110/1011→	余数110	1001101	← 循环左移	0011011
3	1100/1011→	余数111	0011011	← 循环左移	0110110
4	1110/1011→	余数101	0110110	← 循环左移	1101100
5	1010/1011→	余数001	1101100	← 循环左移	1011001

将D_0（最低位）变反：$D_{6\sim0}$=1011000

6	0010/1011→	余数010	1011000	← 循环左移	0110001
7	0100/1011→	余数100	0110001	← 循环左移	1100010

↓ n=7次循环　　↓ 循环出现　　↓ 已修复的数据

※纠错方法的特点

最多1位错、(n=7, k=4)码制、G(x)=1011

①始终将余数**001**作为出错位D_0的定位依据；

②每一步循环中，余数补0产生新余数，数据需循环左移1位，以保持余数和出错位的对应关系；

③当余数001**第1次**出现时，将当前D_0变反；

④初始余数再次出现时，结束循环，刚好是2^r-1步；

⑤**无须定位出错位**(与海明不同)，仅通过1次码位变反(固定为D_0) 和2^r-1次循环操作，即可校正出错位。

一般不会试图去纠正多位数据错误，代价太高！