南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

第 2 章 数据信息的表示与运算单元

主讲 张功萱

第2.7节

数据校验码

1. 数据传输过程出错怎么办？
2. 海明码的校验依据是什么？
3. CRC校验码及特点是什么？

- 数据在计算机系统内形成、存取和传送过程中，可能会因为某种原因而产生错误。为减少和避免这类错误，一方面是从电路、电源、布线等方面采取多方面措施提高机器的抗干扰能力，尽可能提高计算机硬件本身的可靠性，另一方面是在数据编码上采取检错纠错的措施。

- 数据校验码：具有检测某些错误或带有自动纠正错误能力的数据编码方式。
- 常用的数据校验码有奇偶校验码、海明校验码、循环校验码等。

数据校验码的实现原理

- 在正常编码中加入一些冗余位，即在正常编码组中加入一些非法编码，当合法数据编码出现某些错误时，就成为非法编码，因此就可以通过检测编码是否合法来达到自动发现、定位乃至改正错误的目的。
- 在数据校验码的设计中，需要根据编码的码距合理地安排非法编码的数量和编码规则。

2.8.1　码距与数据校验码

- **编码的距离**：通常把一组编码中任何两个编码之间代码不同的位数称为这两个编码的距离，也称**海明距离**。
- **码距**：指在一组编码中任何两个编码之间最小的距离。
- 例如 编码0011与0001，仅有一位不同，称其海明距离为1。

- 例：一组编码采用四位二进制编码表示16种状态，因为从0000到1111这16种编码都用到了，因此这组编码的码距为1。
- 因为这组编码的码距为1，在这组编码中任何一个状态的四位码中的一位或几位出错，都会变成另一个合法编码，所以这组编码没有查错和纠错能力。

- 例：一组编码采用四位二进制表示8个状态，例如只将其中的8种编码0000、0011、0101、0110、1001、1010、1100、1111用作合法编码，而将另外8种编码作为非法编码，此时这组编码的码距为2。
- 在这组编码中，要从一个合法编码改为另一个合法编码需要修改2位。如果在数据传输过程中，任何一个合法编码有一位发生了错误，就会出现非法编码。
- 例如编码0000的任意一位发生错误形成的编码都不是合法编码，因此系统只要检查编码的合法性，就可以发现错误。

- 校验码通常是在正常编码的基础上按特别规定增加一些附加的校验位形成的，即通过增大编码的码距来实现检查和纠正错误的目的。
- 合理地增加校验位、增大码距，就能提高校验码发现错误的能力。

检查错误的位数与码距的关系

- 要检查1位错误，编码的码距需要1＋1＝2。
- 要检查e位错，编码的码距需要e＋1。

- 因为对于码距为e＋1的编码，一个码字在e位以内出错，是无法将一个合法编码变为另外一个合法编码的。

纠正错误的位数与码距的关系

- 如果编码出错的位置能够确定，则将出错位的内容取反，就能够自动纠正错误。
- **要纠正t位错，编码的码距需要2t＋1。**
- 这是因为当码距达到2t＋1时，即使合法编码中有t位出错，它与原合法编码的编码距离还是比与其他任何合法码字的编码距离要小，这样就可以惟一地确定它的合法编码，即可以自动纠正错误。

- 例：考虑下面只有四个合法编码的编码组
- 0000000000，0000011111，
  1111100000，1111111111。
- 这个编码组的码距为5，说明它着能纠两位错。
- 如果在数据传输过程中，接受方接收到一个编码0000000111，就能够知道原来的正确编码应该是0000011111（必须假定不会出现两位以上的错误）

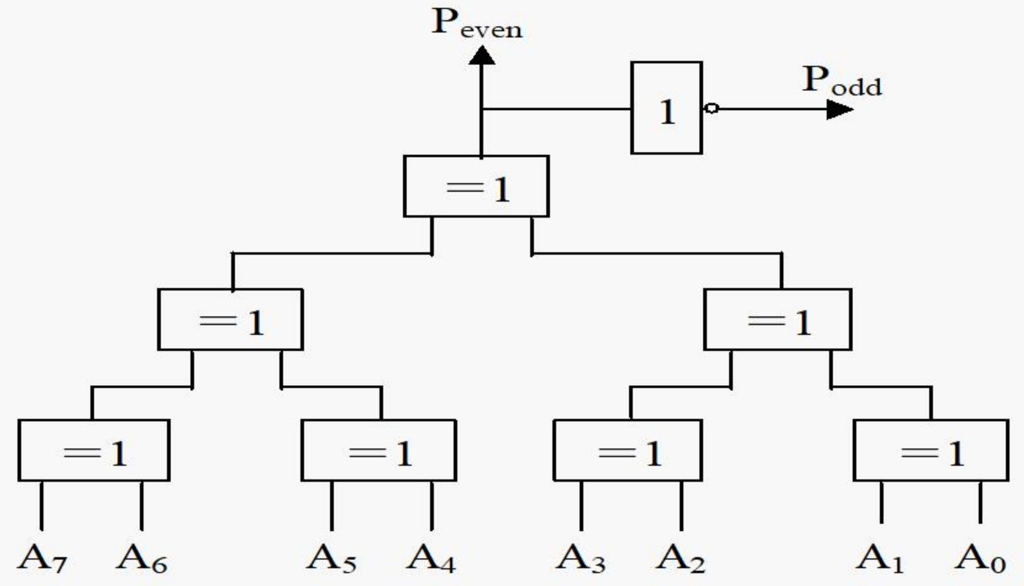
码距与校验码的检错和纠错能力的关系

- 校验位越多，码距越大，编码的检错和纠错能力越强。
- 记码距为d，码距与校验码的检错和纠错能力的关系是：
- $d \geq e+1$　　可检验e个错。
- $d \geq 2t+1$　　可纠正t个错。
- $d \geq e+t+1$　且$e>t$，可检e个错并能纠正t个错。

2.8.2 奇偶校验码

- 奇偶校验码是一种最简单、最常用的校验码，广泛用于主存的读写校验或ASCII码字符传送过程中的检查。
- 1. 奇偶校验码的编码方法
- 在n位有效信息位上增加一个二进制位作为校验位P，构成n＋1位的奇偶校验码。
- 奇校验Odd：校验位P的取值（0或1）使n＋1位的奇偶校验码中“1”的个数为奇数
- 偶校验Even：校验位P的取值（0或1）使n＋1位的奇偶校验码中“1”的个数为偶数。
- 校验位的位置在有效信息位的最高位之前或者在最低位之后。

- 例：设$A_6A_5A_4A_3A_2A_1A_0$为7位有效信息，加一个校验位P，构成8位的奇偶校验码为：
- $A_6A_5A_4A_3A_2A_1A_0P$ 或 $PA_6A_5A_4A_3A_2A_1A_0$
- 若采用偶校验，则：
- $P_{even} = A_6 \oplus A_5 \oplus A_4 \oplus A_3 \oplus A_2 \oplus A_1 \oplus A_0$
- 若采用奇校验，则
- $P_{odd} = \overline{P_{even}}$

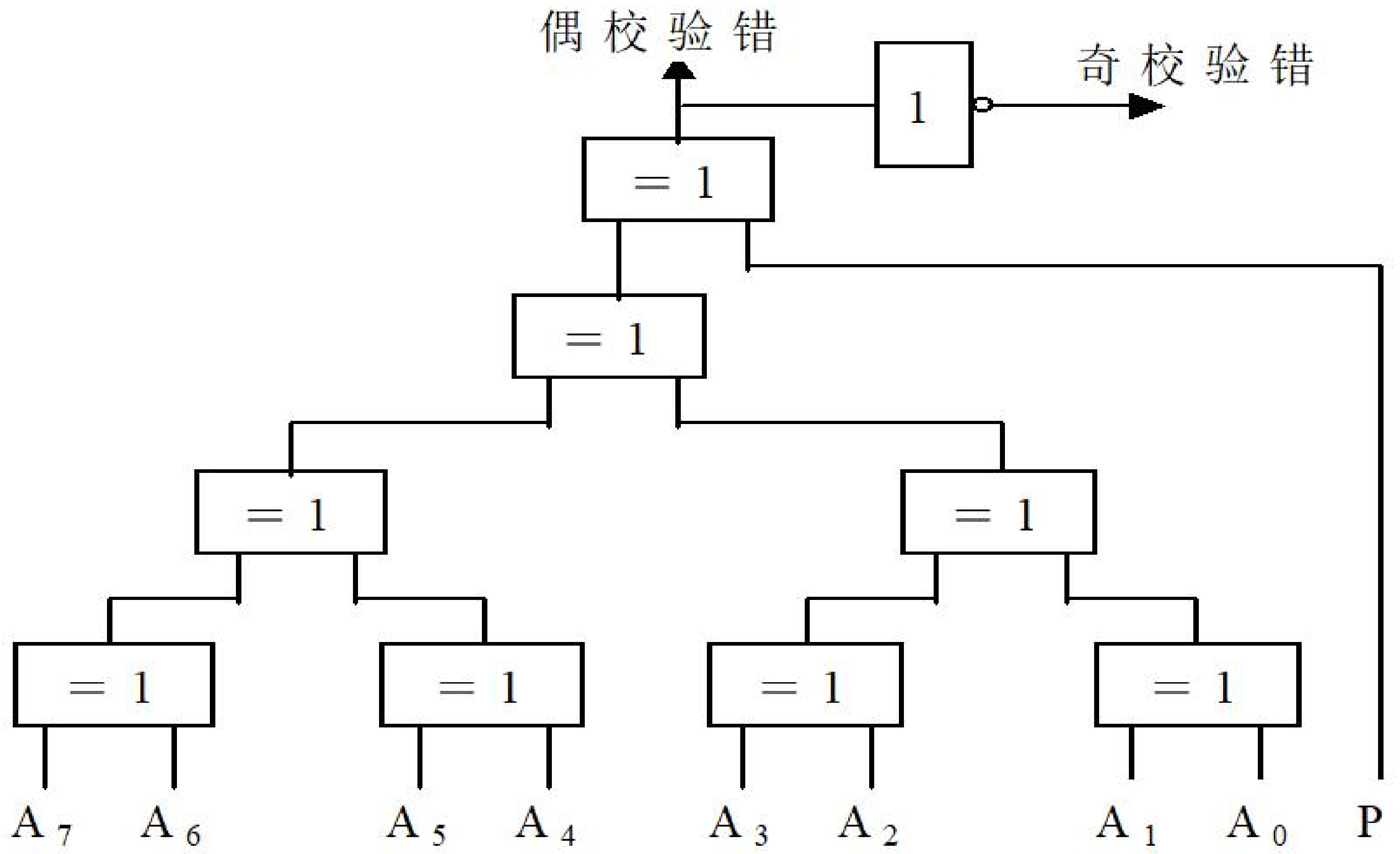


校验位 P 的形成电路

- 例：求7位信息码1100111的奇校验码和偶校验码（设校验位在最低位）。
- 解：
- (1) 1100111的奇校验码
- 因为1100111中“1”的个数为奇，所以奇校验位P＝0，1100111的奇校验码为11001110。
- (2) 1100111的偶校验码
- 因为1100111中“1”的个数为奇，所以偶校验位P＝1，1100111的偶校验码为11001111。

2. 奇偶校验码的校验

- 若接到一奇校验码中“1”的个数为偶数，或接到一偶校验码中“1”的个数为奇数，则表示有一位出错。
- 以上面的七位有效信息的奇偶校验码为例：
- 偶校验错：
- $E = A6 \oplus A5 \oplus A4 \oplus A3 \oplus A2 \oplus A1 \oplus A0 \oplus P_{even}$
- 奇校验错：
- $E = \overline{A6 \oplus A5 \oplus A4 \oplus A3 \oplus A2 \oplus A1 \oplus A0 \oplus P_{odd}}$
- E＝0，表示无错； E＝1，表示校验出错



奇偶校验电路

3. 奇偶校验码的校错能力

- 奇偶校验码只能发现奇数位个错误，而无法发现偶数位个错误，而且即使发现奇数位个错误也无法确定出错的位置，因而无法自动纠正错误。
- 由于现代计算机可靠性比较高，出错概率很低，而出错中只有一位出错的概率最高，因此用奇偶校验检测一位出错，能够满足一般可靠性要求。在CPU与主存的信息传送过程中，奇偶校验被广泛应用。

2.8.3 海明校验码

- 海明校验码的实质

  在奇偶校验的基础上，增加校验位的位数，构成多组奇偶校验，以便发现错误并自动纠正错误。

- 海明校验码校验位数的选择

- 设有效信息位的位数为n，校验位数为k，则能够检测一位出错并能自动纠正一位错误的海明校验码应满足下面的关系：

$$2^k \geq n + k + 1$$

由此式可计算出具有检1纠1错能力的海明校验码中n与k的关系。

1. 有效信息位与校验位的关系

k（最小）	n
2	1
3	2～4
4	5～11
5	12～26
6	27～57
7	58～120

2. 海明校验码的编码方法

- (1) n位有效信息选择k个校验位，构成n＋k位的海明校验码。若校验码位号从左向右（或从右向左）按从1到n＋k排列，则校验位的位号分别为2^i，i＝0，1，2…k－1，有效信息位按原排列次序安排在其它位号中。

- 例：一个字符的ASCH码为$A_6A_5A_4A_3A_2A_1A_0$，根据n与k的关系选择k＝4。构成7＋4＝11位的海明校验码。

位号	1	2	3	4	5	6	7	8	9	10	11
编码	P_1	P_2	A_6	P_4	A_5	A_4	A_3	P_8	A_2	A_1	A_0

P_1、P_2、P_4、P_8分别为四个校验位，其下标是它们在海明码中的位号，即2^0、2^1、2^2、2^3的位置。

- (2) k个校验位构成k组奇偶校验，每个有效信息位都被2个或2个以上的校验位校验。被校验的位号等于校验它的校验位位号之和。
  - 在上例中，A_6的位号为3，3＝1＋2，所以A_6被P_1、P_2校验。A_3位号为7，7＝1＋2＋4，所以A_3被P_1、P_2、P_4校验。以此方法可知每个信息位分别被哪些校验位校验。
  - A_6：P_1、P_2　　A_5：P_1、P_4
  - A_4：P_2、P_4　　A_3：P_1、P_2、P_4
  - A_2：P_1、P_8　　A_1：P_2、P_8
  - A_0：P_1、P_2、P_8

- 由此可得形成k个校验位的校验组
- P1：A6、A5、A3、A2、A0　　（第一组）
- P2：A6、A4、A3、A1、A0　　（第二组）
- P4：A5、A4、A3　　（第三组）
- P8：A2、A1、A0　　（第四组）
- (3)统计参与每组奇偶校验的位号，按奇偶校验原理，由已知的有效信息按奇校验或偶校验求出各个校验位，进而形成海明校验码。

- 以7位ASCII码的海明校验码为例
- 按偶校验求出各个校验位的方法是：
- $P_{1even} = A_6 \oplus A_5 \oplus A_3 \oplus A_2 \oplus A_0$
- $P_{2even} = A_6 \oplus A_4 \oplus A_3 \oplus A_1 \oplus A_0$
- $P_{4even} = A_5 \oplus A_4 \oplus A_3$
- $P_{8even} = A_2 \oplus A_1 \oplus A_0$
- 按奇校验求出各个校验位的方法是：
- $P_{1odd} = \overline{P_1}_{even}$
- $P_{2odd} = \overline{P_2}_{even}$
- $P_{4odd} = \overline{P_4}_{even}$
- $P_{8odd} = \overline{P_8}_{even}$

- 例：编制ASCII字符“M”的海明校验码。
- 解：“M”的ASCII码为$A_6A_5A_4A_3A_2A_1A_0=1001101$
- $P_{1even}=A_6\oplus A_5\oplus A_3\oplus A_2\oplus A_0=1\oplus 0\oplus 1\oplus 1\oplus 1=0$
- $P_{2even}=A_6\oplus A_4\oplus A_3\oplus A_1\oplus A_0=1\oplus 0\oplus 1\oplus 0\oplus 1=1$
- $P_{4even}=A_5\oplus A_4\oplus A_3=0\oplus 0\oplus 1=1$
- $P_{8even}=A_2\oplus A_1\oplus A_0=1\oplus 0\oplus 1=0$
- 得到ASCII码字符“M”的海明校验码为：
- 01110010101(偶校验)
- 10100011101(奇校验)
- 海明校验码产生后，将信息位和校验位一起存入内存。

3. 海明校验码的校验

- 校验时，K个校验位进行K组奇偶校验，校验结果形成K位的“指误字” EkEk-1…E2E1。
- 若某组校验结果正确，指误字相应位为0；若校验结果错误，指误字相应位为1。若校验结果：
- EkEk-1…E2E1＝全0，则表示无错；
- EkEk-1…E2E1≠全0，则表示有错，
- 指误字代码所对应的十进制值就是出错位的位号。将该位取反，错误码即得到自动纠正。

- 指误字指示出错的前提是代码中只存在一个错。若有多个错，可能查不出来。
- 只有在只存在一个错的前提下，海明校验码才能实现检一纠一错。

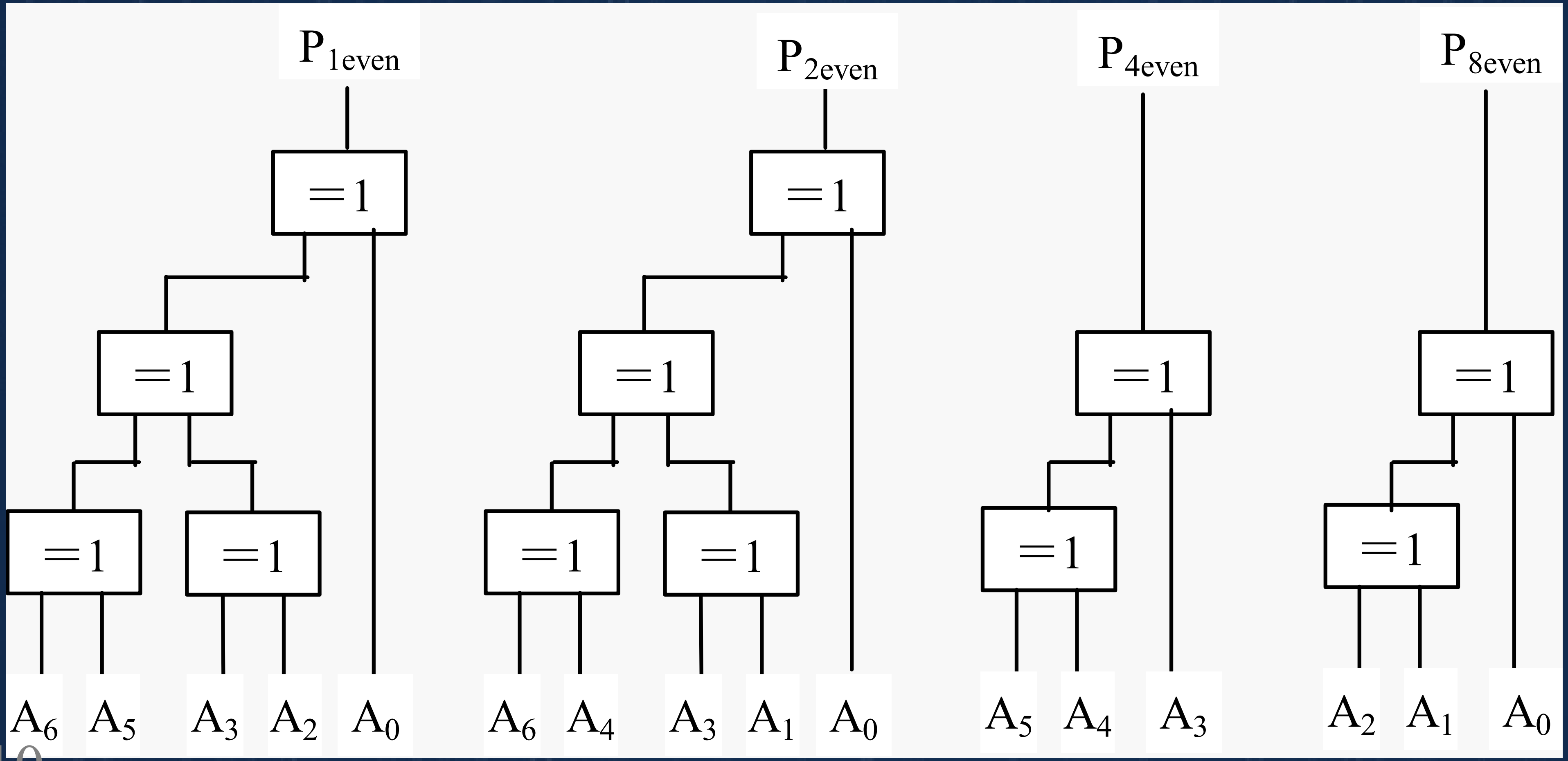
- 例：字符“M”的偶校验码为01110010101，其指误字为
- $E_1 = P_1 \oplus A_6 \oplus A_5 \oplus A_3 \oplus A_2 \oplus A_0$
- $E_2 = P_2 \oplus A_6 \oplus A_4 \oplus A_3 \oplus A_1 \oplus A_0$
- $E_3 = P_4 \oplus A_5 \oplus A_4 \oplus A_3$
- $E_4 = P_8 \oplus A_2 \oplus A_1 \oplus A_0$

- 若接收正确，即接收到的代码是01110010101，则
- $E1 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0$
- $E2 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0$
- $E3 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$
- $E4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$
- 若接收出错，即接收到的代码是01110110101，则
- $E1 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0$
- $E2 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 1$
- $E3 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$
- $E4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$

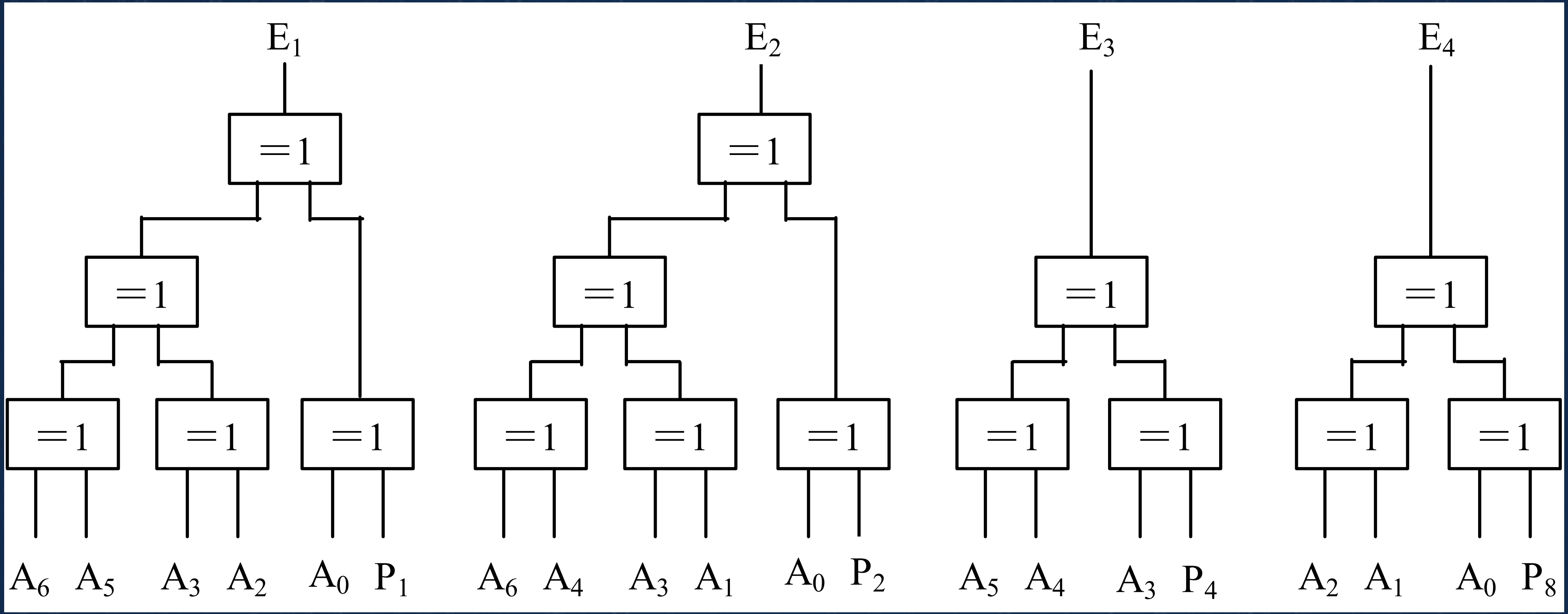
- 得到的指误字为 $E_4E_3E_2E_1$＝0110，表示第6位上的数码出错。将第6位上的数码A_4取反，即可得到正确结果。
- 上述海明校验码可以检查出一位错误并予以纠正，但如果有两位信息出错，就无法正确检测并自动纠错。

- 例：设正确的“M”的偶校验海明校验码为 01110010101，如果信息传送时第3位、第6位同时出错，接受到的校验码为 01010110101，校验时，得到指误字为 $E_4E_3E_2E_1$＝0101，指出第5位代码出错，结果越纠越错。

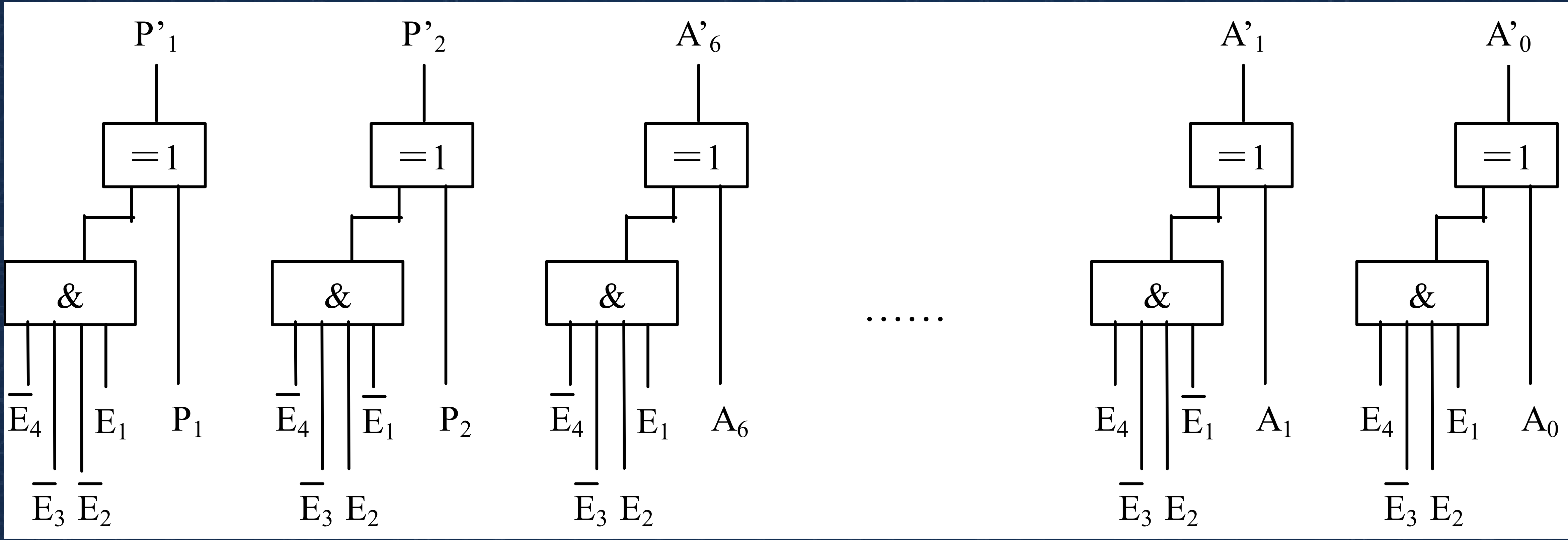
有效信息位n＝7，校验位k＝4的11位检一纠一错海明校验码的形成电路



11位检一纠一错海明校验码的偶校验电路



11位检一纠一错海明校验码的自动纠错电路



4. 扩展的海明校验码

- 为了满足检二纠一错的要求，可将检一纠一错海明校验码再进行奇偶校验。
- 扩展海明校验码：在检一纠一错海明校验码上增加一个奇偶校验位P_0。
- 按偶校验规则，P_0的取值是使n＋k＋1位的校验码中的“1”的个数为偶数。即：
- $P_0 = P_1 \oplus P_2 \oplus A_6 \oplus P_4 \oplus A_5 \oplus A_4 \oplus A_3 \oplus P_8 \oplus A_2 \oplus A_1 \oplus A_8$

- 校验时，首先对包括P_0整个n＋k＋1位海明校验码进行校验，校验结果为E_0。
- $E_0 = P_0 \oplus P_1 \oplus P_2 \oplus A_6 \oplus P_4 \oplus A_5 \oplus A_4 \oplus A_3 \oplus P_8 \oplus A_2 \oplus A_1 \oplus A_8$

- 若校验结果：
- $E_0 = 0$，奇偶校验正确；
- $E_0 = 1$，奇偶校验出错。

- 然后再按检一纠一错海明校验码对不包括P_0的各组进行校验，得到指误字$E_k E_{k-1} \ldots E_2 E_1$。

- 若校验结果：
- $E_0=0$，$E_kE_{k-1}\ldots E_1=00\ldots0$　表示无错。
- $E_0=1$，$E_kE_{k-1}\ldots E_1\neq00\ldots0$　表示有一位出错，可根据$E_kE_{k-1}\ldots E_1$的值确定出错位号并自动纠正。
- $E_0=0$，$E_kE_{k-1}\ldots E_1\neq00\ldots0$　表示有两位出错，此时无法确定出错位置，也无法纠错。
- $E_0=1$，$E_kE_{k-1}\ldots E_1=00\ldots0$　表示P_0出错，将P_0取反，即可自动纠正错误。

- 在上述校验中，每一个数据位都至少出现在2个P_i值的形成关系中，当任一位数码出错时，必将引起2到3个P_i值发生变化。任何偶数个数出错，E_0肯定为0，因此可以区分两位或一位出错。

- 从上面讨论中，可以看出校验位位数越多，检错纠错的能力就越强。

例子：

8421码		采用偶校验		海明校验码	
0000		00000		0000000	
0001		00011		1101001	
0010	码距为1	00101	码距为2	0101010	码距为3
0011	不能检错	00110	可检1位	1000011	可检1位
0100	和纠错	01001	错	1000100	纠1位错
0101		01010		0101101	
0110		01100		1100110	
0111		01111		0001111	
1000		10001		1110000	
1001		10010		0011001	

- 校验位越多，码距d越大，编码的检错和纠错能力越强。码距与纠错能力的关系是：
- $d \geqslant e+1$　　可检验e个错。
- $d \geqslant 2t+1$　　可纠正t个错。
- $d \geqslant e+t+1$　　且$e>t$，可检e个错且能纠正t个错。
- 扩展海明码的码距为4，可检2个错且能纠正1个错。

2.8.4 循环冗余校验码

- 循环冗余校验码(Cyclic Redundancy Check)简称为CRC码。是一种具有很强检错纠错能力的校验码。
- CRC码广泛用于磁盘、磁带等辅助存储器的校验，在计算机网络和通信中亦被广泛采用。
- 循环冗余校验码通过除法运算来建立有效信息和校验位之间的约定关系。

1. CRC码的编码思想

- 对于多项式M(x)和G(x)有

$$\frac{M(x)}{G(x)}=Q(x)+\frac{R(x)}{G(x)} \longrightarrow M(x)-R(x)=Q(x)\times G(x)$$

- 如果对于一个M(x)，事先按某一G(x)求得Q(x)和R(x)。
- 传送时，把M(x)－R(x)作为编好的校验码传送，当收到该校验码后仍用原约定的多项式G(x)去除。
- 若能整除，即余数为0，则表示该校验码正确；
- 若不能整除，即余数不为0，表明有错，并可根据余数值确定出错位号，进行自动纠正。

2. 模2运算

- 由于减余数操作可能涉及到借位运算，难于用简单的拼装方法实现编码。为回避借位，将运算改为模2运算。
- 循环冗余校验码是一种基于模2运算建立编码规律的校验码，
- 模2运算：不考虑进位和借位的二进制运算。

- (1) 模2加减：即按位加，用异或规则实现。即：
- 0±0＝0　0±1＝1±0＝1　1±1＝0
- (2) 模2乘：按模2加求部分积之和，不进位。
- (3) 模2除：按模2减求部分余数，不借位。若部分余数(首次为被除数)最高位为1，则上商为1；若部分余数最高位为0，则上商为0。每求一位商应使部分余数减少一位，即去掉部分余数的最高位，再继续求下一位商。当部分余数的位数小于除数位数时，该余数就是最后的余数。

- 例：按模2加减规则，计算1100＋0110；1010－0111；1010＋1010。

1100	1010	1010
＋ 0110	－ 0111	＋ 1010
1010	1101	0000

- 可见，在模2加的条件下，R(x)＋R(x)＝0。

- 例：按模2乘规则，计算1010×1011；1101×1001。

$$\begin{array}{r} 1010 \\ \times\qquad 1011 \\ \hline 1010 \\ 1010 \\ 0000 \\ 1010 \\ \hline 1001110 \end{array} \quad \cdots \text{乘积}$$

$$\begin{array}{r} 1101 \\ \times\qquad 1001 \\ \hline 1101 \\ 0000 \\ 0000 \\ 1101 \\ \hline 1100101 \end{array} \quad \cdots \text{乘积}$$

- 例：按模2除规则，计算1000000÷1001；1011001÷1101 。

1001 … 商Q

1001) 1000000
1001
0010
0000
0100
0000
1000
1001
001 … 余数R

1100 … 商Q

1101) 1011001
1101
1100
1101
0010
0000
0101
0000
101 … 余数R

3. CRC码的编码方法

- (1) 把待编的n位有效信息表示为多项式M(x)，
- $M(x)=C_{n-1}x^{n-1}+C_{n-2}x^{n-2}+\ldots+C_1x^1+C_0$
- $C_i=0$ 或 1 对应M中第i位的信息
- 例： $M(x)=1001=x^3+1$

- 选择一个k+1位约定多项式G(x)作为约定除数（又称生成多项式）
- $G(x)=G_kx^k+G_{k-1}x^{k-1}+\ldots+G_1x^1+G_0$
- 例： $G(x)=1011=x^3+x^1+1$

- (2) 把M(x)左移k位，得到$M(x) \cdot X^k$。然后按模2除法，用$M(x) \cdot X^k$除以G(x)得到k位余数R(x)。即：

$$\frac{M(x) \cdot X^k}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

- (3)将$M(x) \cdot X^k$与余数R(x)作模2相加，即形成**循环冗余校验码**。
- $M(x) \cdot X^k + R(x) = Q(x) \cdot G(x) + R(x) + R(x)$
  $= Q(x) \cdot G(x)$ （模2加）

在模2加的条件下，$R(x) + R(x) = 0$。

- 由于$M(x) \cdot X^k + R(x)$可被$G(x)$整除，所以可作为循环冗余码。
- 因为$M(x)$左移k位后空出k位，所以与$R(x)$的模2加可由简单的拼装来实现。

- 例：将4位有效信息1101编成7位CRC码。其中生成多项式为4位多项式，$G(x)=x^3+x^1+1$。
- 解： $M(x)=1101=x^3+x^2+1$，M(x)左移3位，得：
- $M(x)\bullet X^3=1101000$， $G(x)=x^3+x^1+1=1011$
- 将$M(x)\bullet X^3$ 模2除以 G(x)得到余数：
- $R(x)=001$
- ∵ $M(x)\bullet X^3+R(x)=1101000+001=1101001$
- ∴ M(x)的7位CRC码为1101001。
- 在该编码中，由于n=4，n+k=7，故称(7、4)码。

4. CRC码的校验

- 把接收到的CRC码用原约定的生成多项式G(x)作模2除，若除得余数为0，表示没有错误；若除得余数不为0，表示有一位出错。根据余数值可确定出错的位置。

上例中(7，4)码的出错模式(G(x)＝1011)

	A7	A6	A5	A4	A3	A2	A1	余数	出错位
正确码	1	1	0	1	0	0	1	000	无
错误码	1	1	0	1	0	0	0	001	1
	1	1	0	1	0	1	1	010	2
	1	1	0	1	1	0	1	100	3
	1	1	0	0	0	0	1	011	4
	1	1	1	1	0	0	1	110	5
	1	0	0	1	0	0	1	111	6
	0	1	0	1	0	0	1	101	7

- 以表中第一行错误码为例，把余数001补0再除以G(x)＝1011，第二次余数为010，再补0除以1011，得余数为100，按此继续除下去，得余数依次为011，110，111，101，然后又回到001，反复循环，这就是循环码的来历。

$$
\begin{array}{r|l}
 & 0 \\
\hline
1011 & 0010 \\
 & 0000 \\
\hline
 & \ \ 010
\end{array}
\qquad
\begin{array}{r|l}
 & 0 \\
\hline
1011 & 0100 \\
 & 0000 \\
\hline
 & \ \ 100
\end{array}
\qquad
\begin{array}{r|l}
 & 0 \\
\hline
1011 & 1000 \\
 & 1011 \\
\hline
 & \ \ 011
\end{array}
$$

- 根据循环码的特性，当求出的余数不为0时，即表示出错。
- 纠错时，可以一边对余数补0继续作模2除，同时使被检测的CRC码循环左移。每作一次除法，让被检测的CRC码循环左移一次。当出现余数101时，表示原来出错的位已移到A7的位置，通过异或门（$1 \oplus A7 = \overline{A7}$），把A7纠错后在下次移位时送回A1。故移满一个循环（对(7，4)码共移7次），就可以得到一个纠错后的CRC码。
- 当位数增加时，循环冗余校验能有效地降低硬件成本，故得到广泛应用。

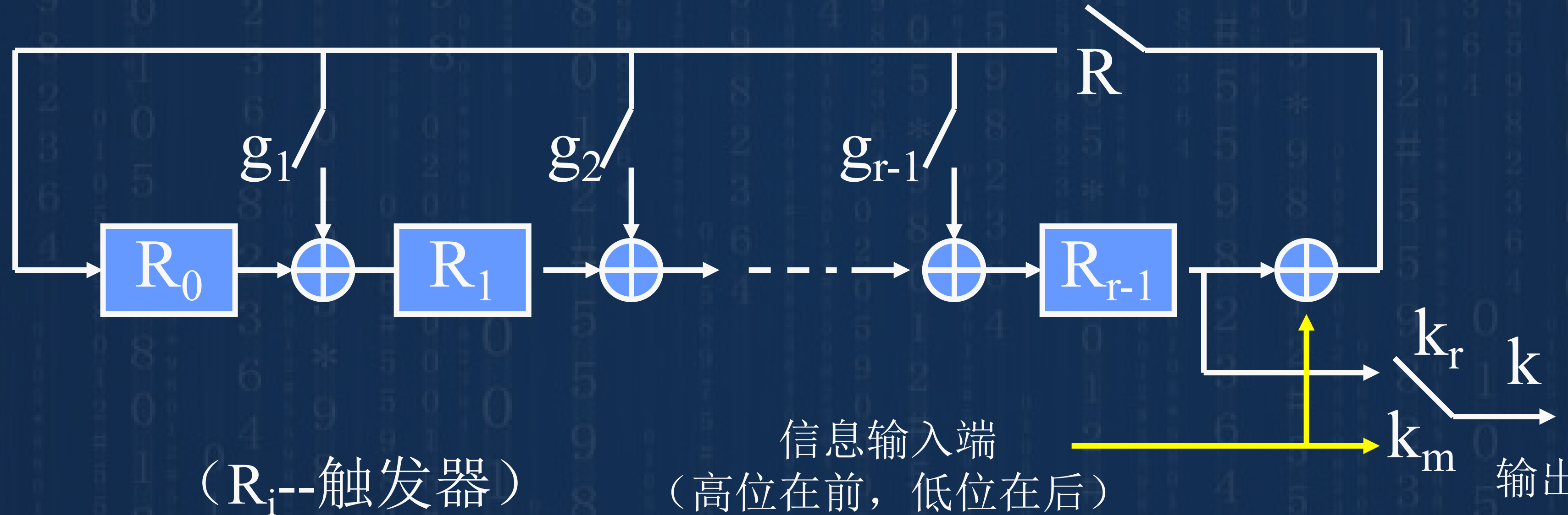
最高位出错码

- 当G(x)→k+1多项式给定，且M(x)的信息位数确定
- 则CRC最高位的出错码也确定。
- 如 G(1011)
- M(1100)→CRC(1100010)
- 错误码（0100010）的余数是
- 101

CRC的逻辑电路实现-编码电路

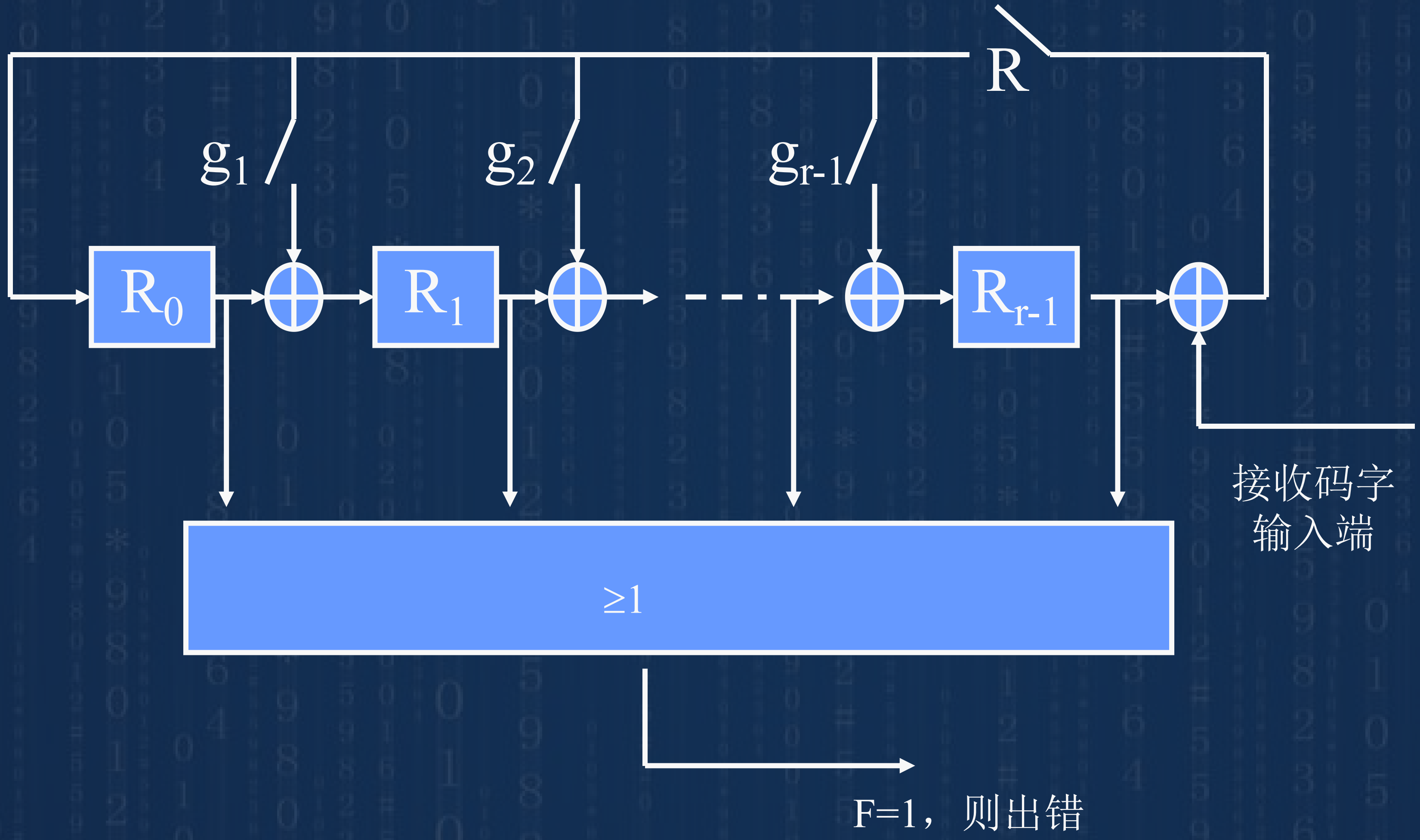
- $G(x)=1+g_1x^1+g_2x^2+\ldots+g_{r-1}x^{r-1}+x^r$

  $g_i \in \{1,2,\ldots,r-1\}$



（R_i--触发器）

CRC的逻辑电路实现-译码电路



5. 循环冗余校验的生成多项式

- 在循环冗余校验中，并非任何一个k＋1位的多项式都可作为生成多项式使用。生成多项式应满足下列要求：
- (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)对余数作模2除法，应能使余数循环。
- 选择不同的生成多项式，CRC码的码距不同，因而检错、校错能力也不同。
- 生成多项式不同，CRC码的出错模式也不同。

常用的生成多项式

CRC码长	有效信息位	码距	G(x) 多项式	G(x)二进制
7	4	3	x^3+x+1	1011
7	4	3	x^3+x^2+1	1101
7	3	4	$x^4+x^3+x^2+1$	11101
7	3	4	x^4+x^2+x+1	10111
15	11	3	x^4+x+1	10011
15	7	5	$x^8+x^7+x^6+x^4+1$	111010001
15	5	7	$x^{10}+x^8+x^5+x^4+x^2+x+1$	10100110111
31	26	3	x^5+x^2+1	100101
31	21	5	$x^{10}+x^9+x^8+x^6+x^5+x^3+1$	11101101001
63	57	3	x^6+x+1	1000011
63	51	5	$x^{12}+x^{10}+x^5+x^4+x^2+1$	1010000110101

- 在数据通信与网络中，通常n相当大，由一千甚至数千个二进制数据位构成一帧，为检测信息传输的正确与否，广泛采用CRC码进行校验。
- 这时所使用的生成多项式的次数比较高，常用的k＝16和k＝32的生成多项式有：
- $CRC-16=x^{16}+x^{15}+x^{2}+1$
- $CRC-CCITT=x^{16}+x^{12}+x^{5}+1$
- $CRC-32=x^{32}+x^{26}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$

- 在网络通信中，通常使用CRC码检测错误，而不是纠正错误。通信发送方按约定的生成多项式形成有效信息的CRC码进行发送，接收方接收到信息后，用约定的生成多项式进行模2除，如果只要得到的余数不为0，就认为检测到差错，于是通知发送方没有正确接收到信息。

本章小结

- 1. 数据表示与数据类型、数据结构的区别，
- 与指令的关系。
- 2.四种码制的编码及特点（补码、移码）。
- 3.定浮点数的表示（IEEE754标准）。
- 4. 定点数/浮点数四则运算单元及组成。
- 5. 数据的长度与存储。
- 6. 数据校验码（海明码、CRC码）。