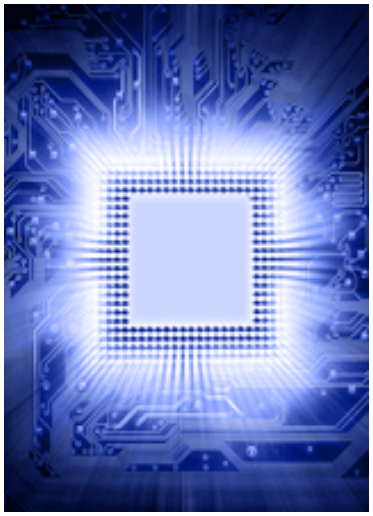
第二章 数制和编码



南京大学人工智能学院

2018-2019 春季

本章学习导引

- 简单地看，数字系统就是处理二进制数码0和1的电路，其中：
  - “二进制数码”就是现实世界中的对象在数字系统中的表示；（数字化的结果）
  - “处理”的方式和二进制数码表示的对象种类有关系。（数字设计的需求）
- 本章学习的目的：初步了解数字系统中
  - 数值量可以如何表示和处理；
  - 非数值对象可以如何表示。

编码

- 用于表示一个数或其他事件的一组二进制码的集合，称为一种**编码**。
  - {对象集，码字集，编码方案}
  - 用于存储、传输、控制、执行等处理；
  - 和具体应用密切相关，无通用处理逻辑。
- 一个含义确切的特定的二进位组合称为**码字**。
  - 码字之间可以有算术关系，也可以没有。
- 编码举例：
  - BCD编码
  - 字符编码：ASCII、GB2312、GB18030、Unicode等
- 特殊的编码需求：检错码、纠错码等。

3、十进制数的二进制编码

- 方便显示和打印。
- 和日常习惯一致。
- 用n位二进制表示
  - 加权码：
    - 8421，2421码等
  - 自反码：
    - 2421，余3码。

Decimal digit	BCD (8421)	2421	Excess-3	Biquinary	1-out-of-10
0	0000	0000	0011	0100001	1000000000
1	0001	0001	0100	0100010	0100000000
2	0010	0010	0101	0100100	0010000000
3	0011	0011	0110	0101000	0001000000
4	0100	0100	0111	0110000	0000100000
5	0101	1011	1000	1000001	0000010000
6	0110	1100	1001	1000010	0000001000
7	0111	1101	1010	1000100	0000000100
8	1000	1110	1011	1001000	0000000010
9	1001	1111	1100	1010000	0000000001
Unused code words					
	1010	0101	0000	0000000	0000000000
	1011	0110	0001	0000001	0000000011
	1100	0111	0010	0000010	0000000101
	1101	1000	1101	0000011	0000000110
	1110	1001	1110	0000101	0000000111
	1111	1010	1111	...	...

- BCD码(Binary-Coded Decimal)=自然二-十进制码=8421码

3、十进制数的二进制编码

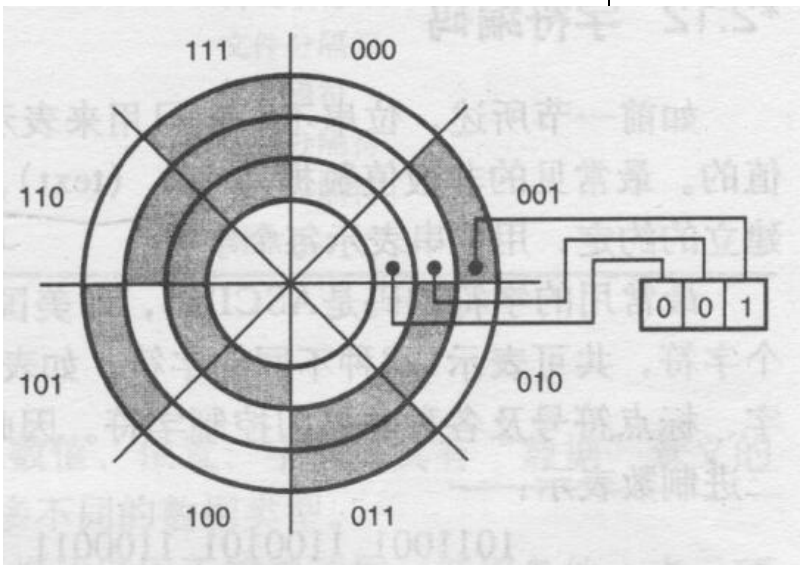
- BCD码运算
  - 类似于4位无符号数加法
  - 超过1001，需要校正，**加6修正**
  - 有进位，**加6修正**

```
    0011  1000     38的BCD码
+)  0100  1001     49的BCD码
-------------------------
    1000  0001     81的BCD码
```

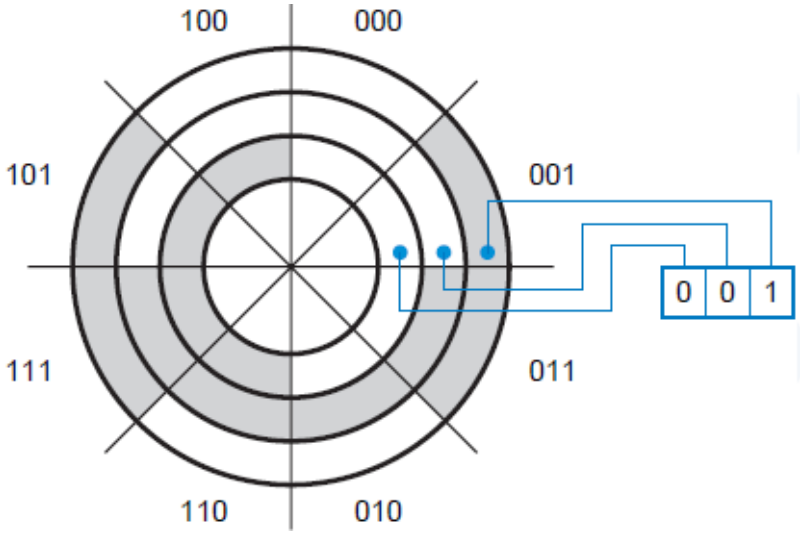
```
    5     0101
+   9   + 1001
-------------
   14     1110
        + 0110   — correction
-------------
10+4       1 0100

    8     1000
+   8   + 1000
-------------
  −16      1 0000
        + 0110   — correction
-------------
10+6       1 0110

    9     1001
+   9   + 1001
-------------
   18     1 0010
        + 0110   — correction
-------------
10+8       1 1000
```

3、格雷码（Gray Code）

- 机械编码盘：
  - 根据盘的旋转位置，触电产生3位二进制编码
  - 相邻编码有多位不同。



- 通过设计数字编码使得每对连续的码字之间只有一个数位变化，这样的编码叫做格雷码。



3、格雷码（Gray Code）

- 产生格雷码的两种方法：
  - 方法1（基于反射码的原理）：
    - 1位格雷码有2个码字：0 1
    - N+1位格雷码中的前2^N个码字是N位格雷码顺序排列，且前面加0。
    - N+1位格雷码中的后2^N个码字是N位格雷码逆序排列，且前面加1
  - 方法二：
    - N位二进制数字从右向左，从0到n-1编号；
    - 如果第i位和第i+1位相同，则对应格雷码的第i位为0，否则为1。

Gray Code(格雷码)

- 利用Gray码的反射特性

0	00	000	0000	1100
1	01	001	0001	1101
		011	0011	1111
	11	010	0010	1110
	10		0110	1010
		110	0111	1011
		111	0101	1001
		101	0100	1000
		100		

Gray Code(格雷码)

- 利用二进制转换到Gray码

  其公式是：$G_n=B_n$

  $$G_i=B_{i+1}\oplus B_i$$

  ⊕称为异或运算(模2加法，即不考虑进位的二进制加法)，运算规则为 0⊕0=0；0⊕1=1；1⊕0=1；1⊕1=0。

  例：二进制数为 1 0 1 1 0 1 0 0

  Gray码 1 1 1 0 1 1 1 0

3、格雷码（Gray Code）

Decimal	Binary	Gray	Decimal	Binary	Gray
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

4、字符编码

- 数字系统中，用位串来表示字符集中的字符。
- ASCII码，7位二进制码，共计128个代码，含：
  - 5
  - 1
  - 3
  - 3

$b_3b_2b_1b_0$	Row (hex)	$b_6b_5b_4$ (column) 000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
0000	0	NUL	DLE	SP	0	@	P	`	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{
1100	C	FF	FS	,	<	L	\	l	|
1101	D	CR	GS	-	=	M	]	m	}
1110	E	SO	RS	.	>	N	^	n	~
1111	F	SI	US	/	?	O		o	DEL

NJU：1001110 1001010 1010101

5、动作、条件和状态的编码

- 数字系统中，将位串用于控制动作、标识条件、表示硬件的状态等。
  - 编码方法有很多种：二进制编码、1-1编码。
  - 处理逻辑：状态的判定、动作的解码等。
  - 使用 b 位二进制编码来表示 n 个不同状态

$$2^b \geq n \Leftrightarrow b = \lceil \log_2 n \rceil$$

表2-12 交通灯控制器的状态

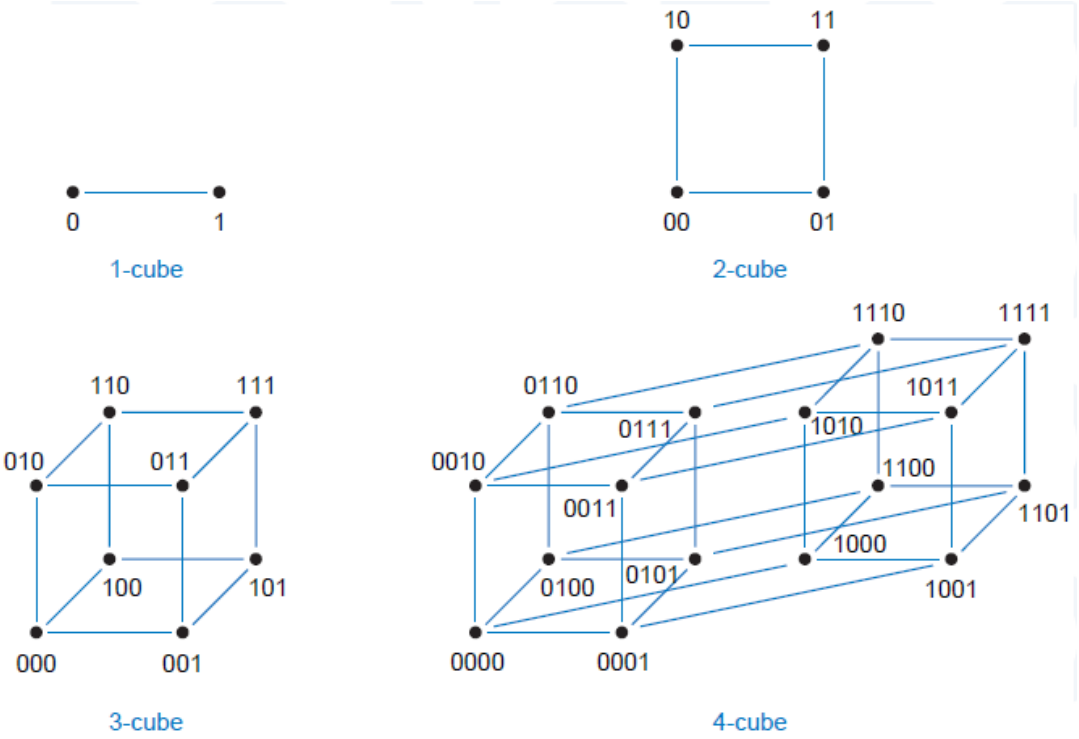
状态	交通灯						码字
	南北方向绿灯	南北方向黄灯	南北方向红灯	东西方向绿灯	东西方向黄灯	东西方向红灯	
南北方向通行	开	关	关	关	关	开	000
南北方向等待	关	开	关	关	关	开	001
南北方向推迟	关	关	开	关	关	开	010
东西方向通行	关	关	开	开	关	关	100
东西方向等待	关	关	开	关	开	关	101
东西方向推迟	关	关	开	关	关	开	110

6、检错码和纠错码

- 具有检测和纠正数据错误的编码特性。
- 数字系统中的差错是指数据损坏，从正确变成其他值。
  - 原因可能很多种：噪声干扰、粒子幅射、介质缺损等，码元可能出错**0→1或1 → 0**。
  - 可能是永久的、可能是暂时的。
- 差错的描述模型：
  - 独立差错模型，单一物理故障只影响单一的数据位。
- 差错类型：
  - 单错：概率大
  - 多错：概率小

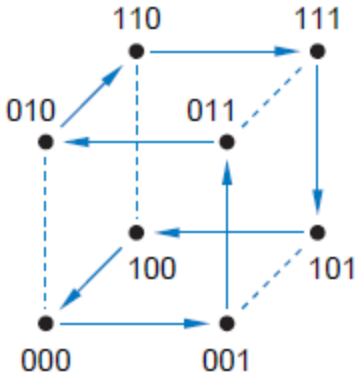
准备知识

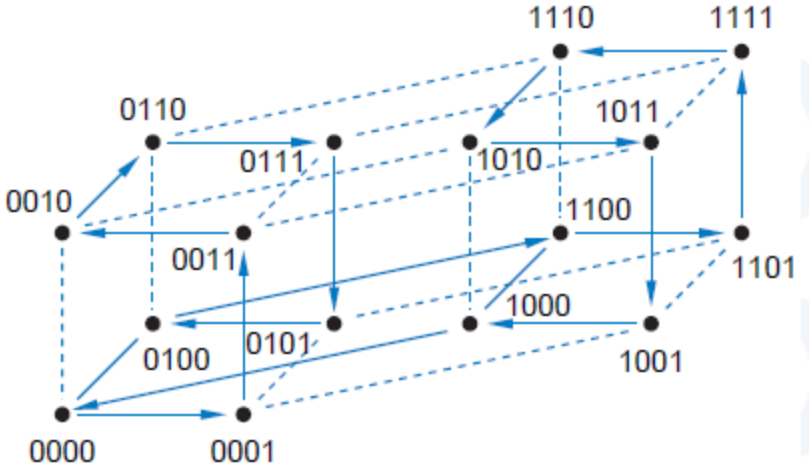
- n维体
  - 把一组编码中的码字看成是一个空间的点，所有码字对应的点构成了n维体（n-cube）



准备知识

- n维体
  - n维体容易将某些编码和逻辑最小化问题形象化。
  - 如：n位格雷码等效于沿着n维体的边寻找一个遍历（仅允许经过一次）所有定点的路径。





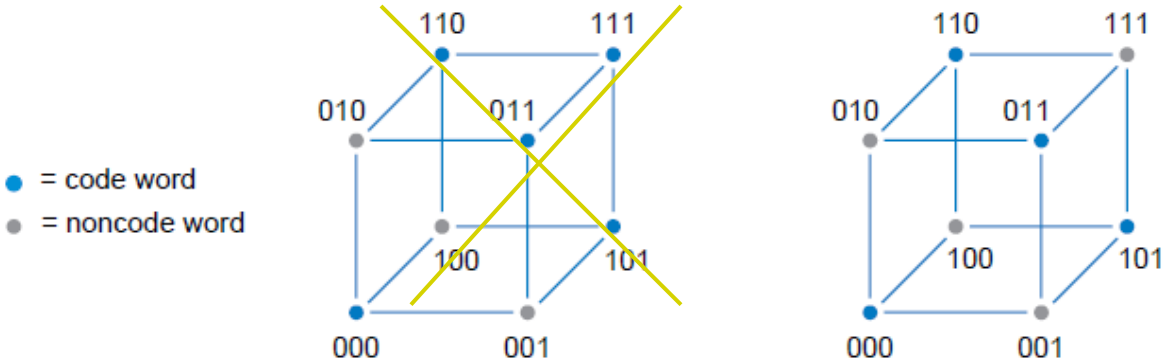
准备知识

- 距离或汉明距离（Hamming distance）
  - 两个位串逐位比较，不同位的数目叫做这两个位串的距离。
  - 就是n维体上，两点之间的最短距离。

- n维体的m维子集（m-subcube）
  - 子集中的顶点有n-m位是完全相同的。
  - 子集可以表示成：xx0。最末一位都相同。
  - 是否属于子集只要判非x位是否匹配即可。

检错码

- 特性：当码字被损坏或改变时，会产生一个不属于编码字集的位串，即非编码字。
- 有错判定：是否合法的编码字。



- 编码方法：
  - N维体顶点的子集。
  - 码字对应的顶点间不能直接相邻。
  - 特性：能检测单错，但不能纠错。最小距离为2

检错码

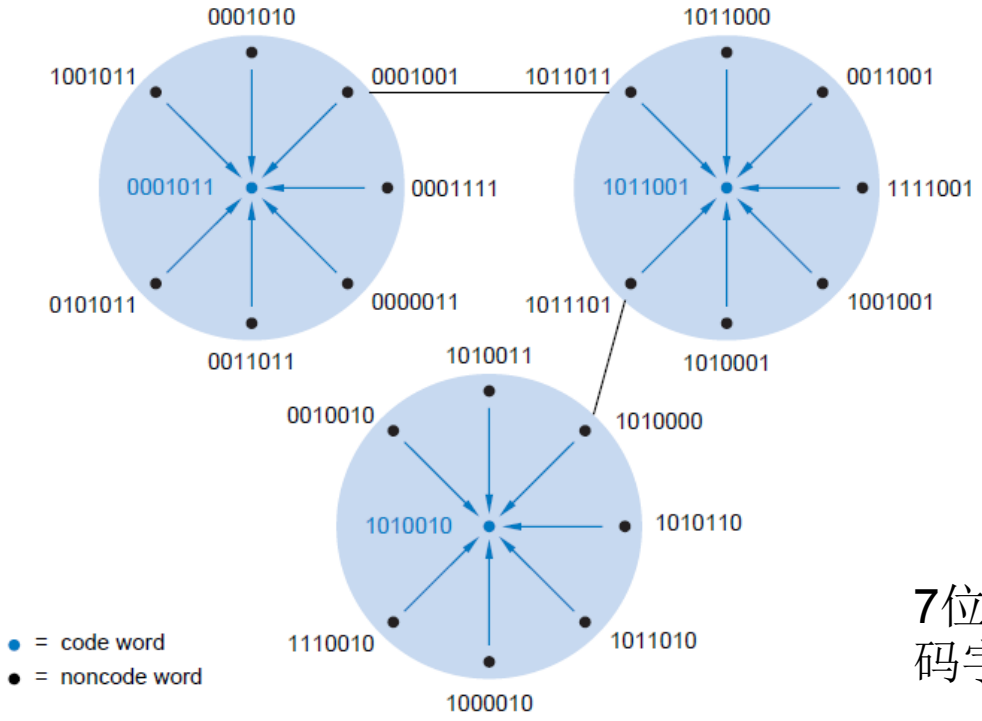
思考：如何证明奇偶检验码的汉明距离大于等于2？

- 通常需要用n+1位来检测n位编码的单错。
  - 码字的前n位是信息位。
  - 增加一个检测位。
- 奇偶校验码
  - 检测位由信息位中1的个数决定。
  - 偶校验码（even-parity code）
    - 利用检测位确保有效的编码字中1的个数为偶数。
  - 奇校验码（odd-parity code）
    - 利用检测位确保有效的编码字中1的个数为奇数。
  - 特性：检测1位错或奇数位错，不能纠错。

Information Bits	Even-parity Code	Odd-parity Code
000	000 0	000 1
001	001 1	001 0
010	010 1	010 0
011	011 0	011 1
100	100 1	100 0
101	101 0	101 1
110	110 0	110 1
111	111 1	111 0

纠错码

- 可以设计1位以上的检测位，构造最小距离大于2的编码可以纠正1位错。
  - 收到一个非法字，就将其改正为距离最近的合法编码即可。

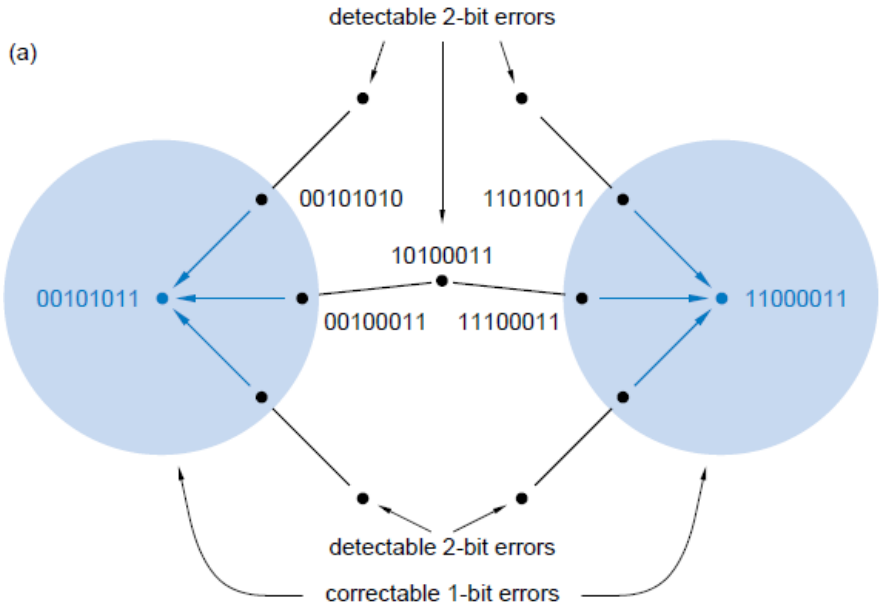


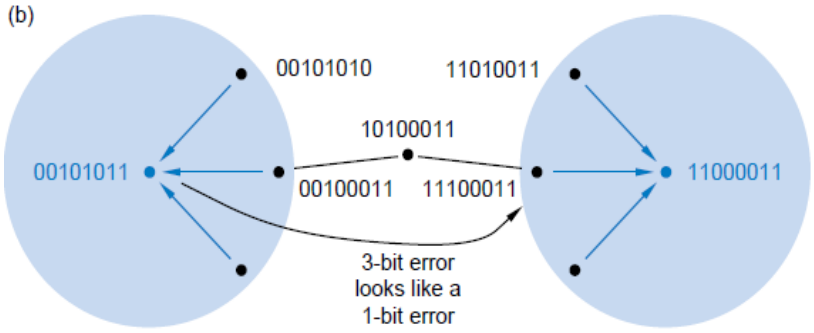
7位距离为3的编码字和非编码字

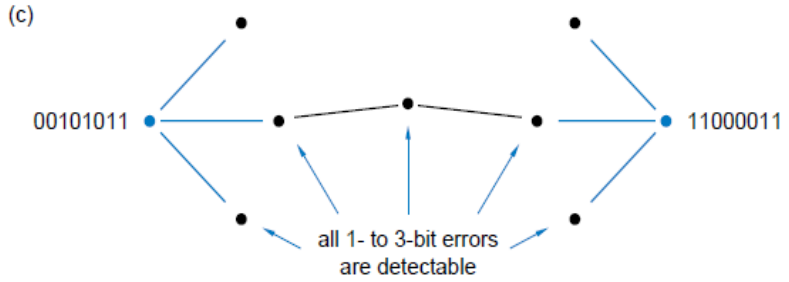
纠错码

思考：距离为5的编码能纠几位错？检测几位错？

- 准则：
  - 最小距离为2c+1的编码最多可以纠正c位错；
  - 最小距离为2c+d+1的编码最多可以纠正c位错，同时最多检测d位错；
  - 最小距离为d+1的编码最多可以检测d位错。







汉明码（hamming code）

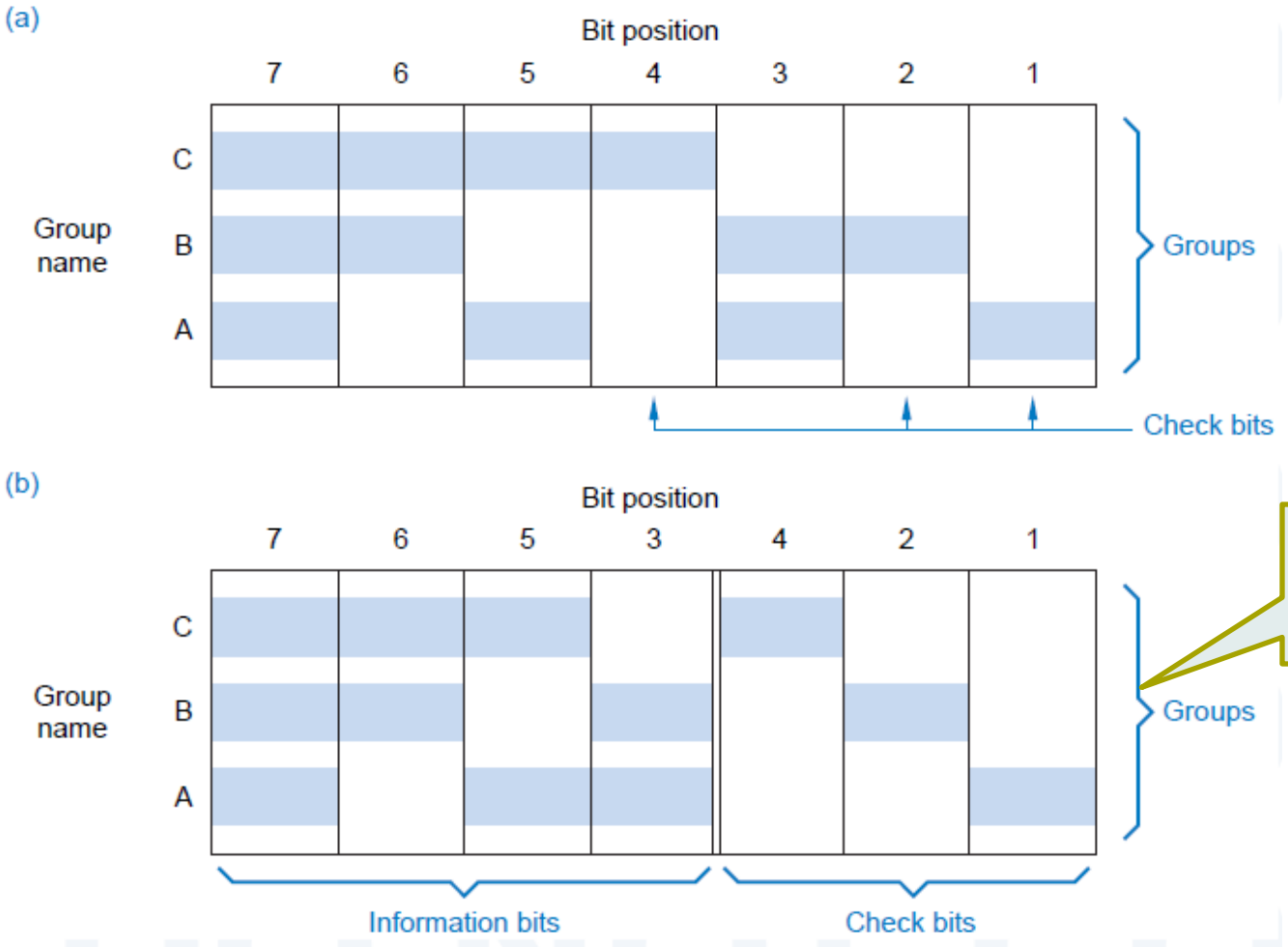
- 理查德·卫斯里·汉明（Richard Wesley Hamming，1915年－1998年），美国数学家，主要贡献在计算机科学和电讯。
- His contributions include the Hamming code, Hamming window，Hamming numbers, Hamming bound（Sphere-packing） and Hamming distance.
- 1945年参加曼哈顿计划，负责编写电脑程式，计算物理学家所提供方程的解。该程式是判断引爆核弹会否燃烧大气层，结果是不会，于是核弹便开始试验。
- 获1968年图灵奖。
- 他是美国电脑协会（ACM）的创立人之一，曾任该组织的主席。（from wikipedia.com）

汉明码（hamming code）

- 设计了一个最小距离为3的编码的一般方法。
  - 对于任意i值，可产生2^i-1位的编码，其中包含i个校验位，2^i-i-1个信息位。
  - 信息位较少的距离为3的编码，可由位数较多的汉明码经删除若干信息位而得到。
- 编码方法：
  - 从右向左给每一位编号：1——2^i-1；
  - 其中2的幂次位置安排校验位，其余信息位；
  - 每个校验位与部分信息位联合成一组，这些信息位的编号中在校验位为1的位置上都是1。
    - 校验位2（010），与信息位011，110，111组成一组。

汉明码（hamming code）

- 7位汉明码的奇偶检验矩阵



汉明码（hamming code）

- 说明：
  - 1位错情况：j位=>非法编码。
    - 该错将引起包含j位的那些组的奇偶性不正确。
    - 每一位至少包含在一个组中。
  - 两位错：j和k
    - 同时包含j和k位的分组，奇偶性不变。
    - 因为j和k的二进制表示中，至少有1位不同，因此，j和k肯定有不同的包含分组，因而，肯定会有对应的组奇偶性错误。

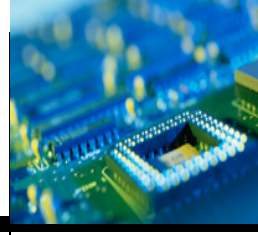
由此可见：i位检验位，可以构造2^i-1位的汉明编码

汉明码（hamming code）

- 纠错处理：
  - 检验所有奇偶检验组：
    - 如果都是偶校验，则码字是正确的；
    - 如果有一组或多组是奇校验，若出现了单错。具有奇校验的组必然和奇偶校验矩阵中的某一列相匹配，则对应的位置号包含错误值，取反即可。
- 例：　　0111100　　0010101
  - 接收编码：0100111
  - 则：　　C组：0100***→1(奇校验)
    B组：01**11*→1 (奇校验)
    A组：0*0*1*1→0 (偶校验)

0000111

注意：汉明码的处理流程



汉明码 → 校验位检测 → 结果处理

思考1：如何判定1位错还是2位错？

有错	无错

• 1位错：
  • 纠错
  • 码字正确
• 2位错：
  • 报错
  • 码字不正确
	• 码字正确

思考2：码字真的正确吗？

汉明码（hamming code）

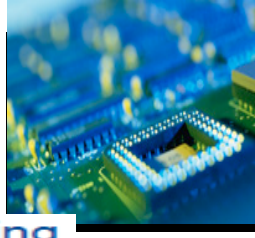


Table 2-14 Code words in distance-3 and distance-4 Hamming codes with four information bits.

Minimum-distance-3 code		*Minimum-distance-4 code*	
Information Bits	*Parity Bits*	*Information Bits*	*Parity Bits*
0000	000	0000	0000
0001	011	0001	0111
0010	101	0010	1011
0011	110	0011	1100
0100	110	0100	1101
0101	101	0101	1010
0110	011	0110	0110
0111	000	0111	0001
1000	111	1000	1110
1001	100	1001	1001
1010	010	1010	0101
1011	001	1011	0010
1100	001	1100	0011
1101	010	1101	0100
1110	100	1110	1000
1111	111	1111	1111

汉明码（hamming code）

- 应用：
  - 距离为3的汉明码扩展一个全局校验位后，可把距离扩展为4，扩展检错能力。
  - 信息位长度D增加时，检验位长度P增长缓慢。$2^P \geq P+D+1$

Table 2-15 Word sizes of distance-3 and distance-4 Hamming codes.

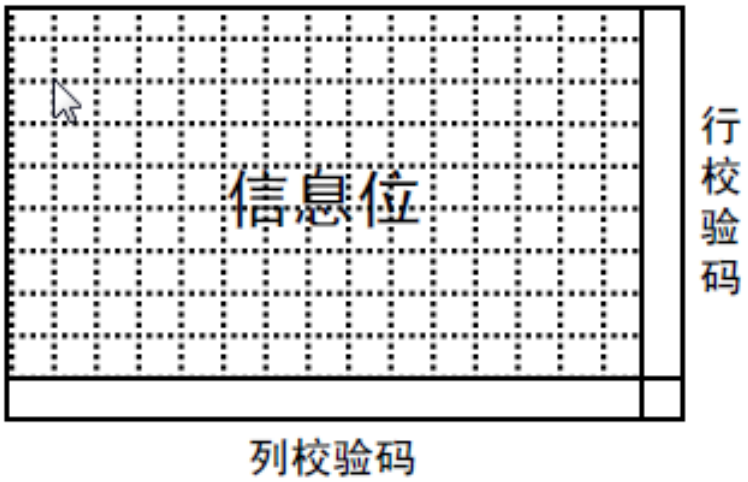
	Minimum-distance-3 Codes		*Minimum-distance-4 Codes*	
Information Bits	*Parity Bits*	*Total Bits*	*Parity Bits*	*Total Bits*
1	2	3	3	4
≤ 4	3	≤ 7	4	≤ 8
≤ 11	4	≤ 15	5	≤ 16
≤ 26	5	≤ 31	6	≤ 32
≤ 57	6	≤ 63	7	≤ 64
≤ 120	7	≤ 127	8	≤ 128

循环校验码

- 循环冗余码校验英文名称为Cyclical Redundancy Check，简称CRC。它是利用除法及余数的原理来作错误侦测。
- CRC能检测信息位中成群的多位错，在信息传送、磁盘存储中，这类错更容易发生。
- 根据应用环境与习惯的不同，CRC又可分为以下几种标准：
  - CRC-12码：传送6-bit字符串。
  - CRC-16码、CRC-CCITT码：传送8-bit字符。
  - CRC-32码：Point-to-Point的同步传输中。

二维码

- 常用的纠错码：2维码，Two-Dimensional Code
  - 行校验码C_{row}： 最小码距d_{row}
  - 列校验码C_{col}：最小码距d_{col}
  - 别名Product Code，最小码距$d_{row} \times d_{col}$
  - 最小距离4，检测1-3 bits错
- 目的：扩大编码的最小距离。

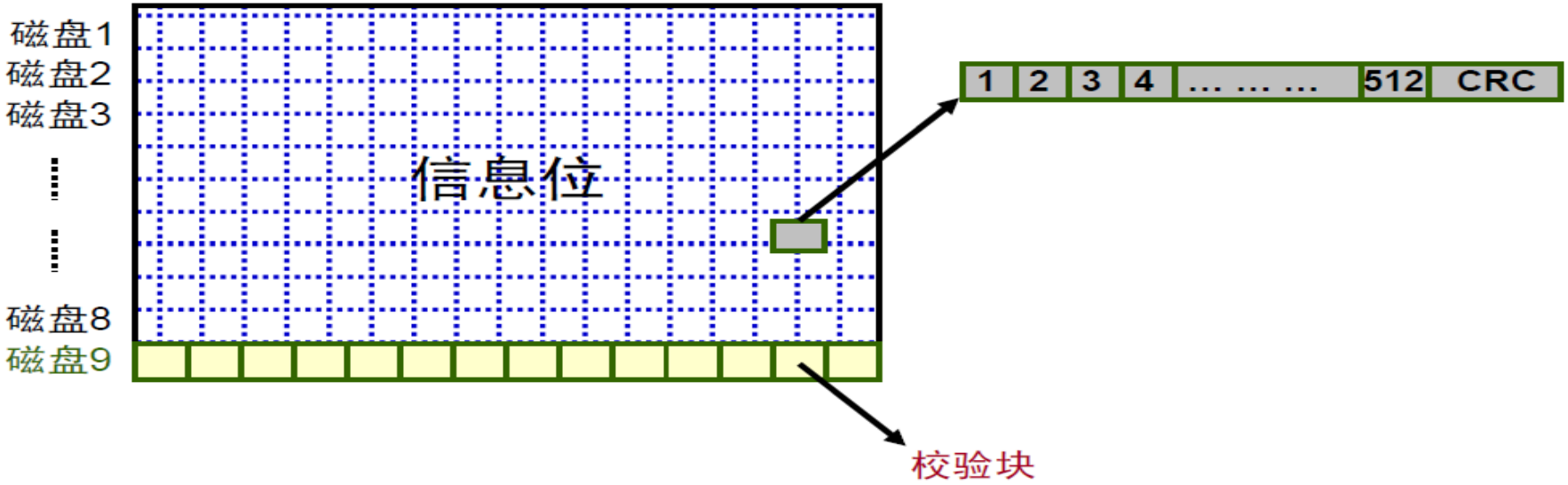


二维码

- 2维乘积码的典型应用：RAID
  - Redundant Array of Inexpensive Disks
  - n+1个磁盘，其中n个磁盘存数据。例如8个10GBytes硬盘存80Gbytes数据，第9个硬盘存校验信息
  - 1个硬盘包含若干数据块，512字节/块，10G相当于2M块

二维码

- RAID：Two-Dimensional Code
  - 每个Block自带CRC校验码以检测本块错误；第n＋1个磁盘的每一块存储前n个磁盘对应数据块的奇偶校验位
  - 第9个磁盘的第b块第i位，使得前8个磁盘的第b块第i位中1的个数为偶数；



二维码

- RAID代价
  - CRC错误时可简单重构该数据块需要n次附加磁盘操作，总比数据丢失好
  - 磁盘写需要附加磁盘访问，以更新校验数据写概率小
  - 费用代价

校验和码

- 校验和码
  - 奇偶校验是按位的模2加法，可扩展到其它模和计算单位。
  - 计算机系统中，按8位分字节，模256加法来生成校验位，得到校验和码（checksum code）。
  - 简单、高效，应用广泛。

检错码和纠错码

- 恒比码, n中取m码
  - 码长**n，m个1码元，　k个0码元(其中n = m + k)**
  - 称**n中取m码**
  - **C_n^m个有效代码**
  - 特性：检测单向多重差错。
  - 例：10取1码。

例：电传通信中，5中取3恒比码

10个有效代码表示10个数字，其余禁码

最小码距为2

检测单个错误和非对称错误

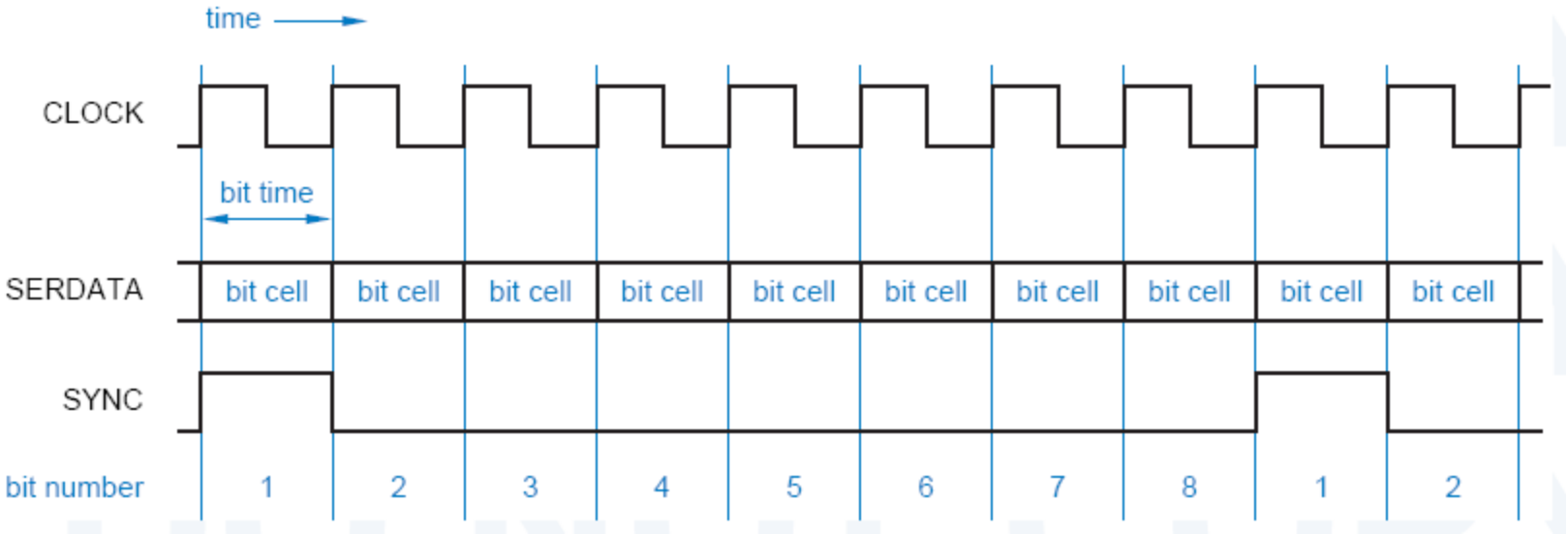
1-out-of-10
1000000000
0100000000
0010000000
0001000000
0000100000
0000010000
0000001000
0000000100
0000000010
0000000001

检错码和纠错码

- 启示
  - 冗余性，检错能力
  - 约束性，纠错能力
- 冗余是有代价的！！！
  - 有效性/可靠性：信息论与编码*
  - 信源/信道编码
    - 分组码
    - 卷积码
    - 级联码
    - Turbo码，LDPC码

7、串行数据传输与存储的编码

- 串行数据传输：线路码



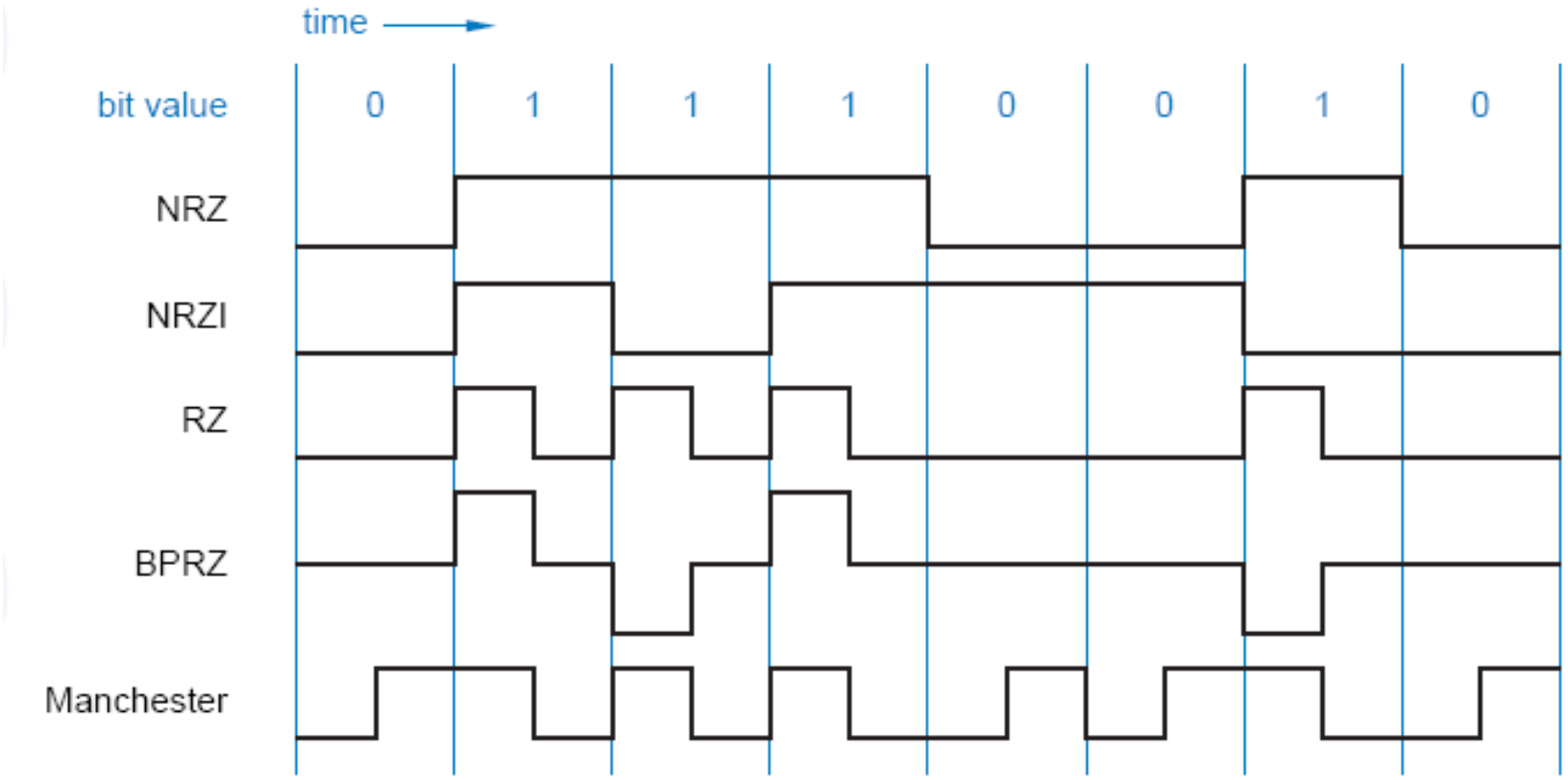
- 位速率bit rate
- 特性：
  - 减少数据通道数量。
  - 降低成本。

串行线路编码

- 每个位元出现在传输线上的实际信号格式取决于线路码（line code）。
- 应用需求：
  - 数字锁相环digital phase-locked loop DPLL：从串行数据流中恢复时钟信号的模拟/数字电路。
    - 条件：串行数据流包含足够的0到1或者1到0 的转换。
  - 直流平衡保证：码流中1和0的数目要大致相等。
  - 恢复字节同步信号。
- 上述需求来源于物理传输机制的需要、来源于提高传输效率的需要。

串行线路编码

- 常用的线路编码方案：



串行线路编码

- NRZ编码（不归零制）：每个位值占据整个位元
  - 简单、可靠。
  - 需要定义位元的数据发送时钟信号。
- NRZ1（不归零逢1翻转制）：当前电平与前一个位元的电平相反发送1，否则发送0。
  - 时钟恢复能力提高。
- RZ编码（归零制）：类似NRZ，但发送1时，1信号传输的时间只有位元的几分之几。通常是1/2
  - 包含1的数据造成很多翻转可以恢复时钟。

串行线路编码

- BPRZ编码（双极性归零制）：传送三个电平信号，+1、0、-1。
  - 直流平衡性要好。
  - 在T1数字电话线上应用很久了。
- Manchester编码：不管传送什么信息，每个位元电平翻转1次，1-0表示1，0-1表示0。
  - 翻转很多；
  - 带宽需求大；
  - 局域网的同轴电缆中用。

汉字编码

- 内码
  - 汉字内码有几个标准，如**GB2312**（ 6763 个汉字），**GB18030**（ 27484个汉字），**Unicode**，**Big5**
  - 这些标准区别在于：所收集汉字多少不同，来自地方不同，如大陆，台湾，美国
  - 例如：电子邮件*
- 外码
  - 汉字的输入码**(外码)编码方案和输入法**至今种类已超过千种。需要统一和标准化。
  - 拼音码、字形码

本章总结

- 了解这些数字系统中的处理对象的表示方式；
  - 不管是数值还是编码，其实在数字系统内部都是0、1表示的。
  - 对0、1值的处理是各类数字系统的共性基础。
- 理解这些对象处理需求的根源；
  - 处理需求的不同，导致数字系统的功能各异。
  - 学习数字系统设计，不仅要了解信息加工处理技巧，还要知道这些信息是从哪里来的，将要到哪里去。
- 本章介绍的各种处理都可以用数字逻辑电路来实现，多数内容在后续章节中会讨论其实现方案。

思考题：

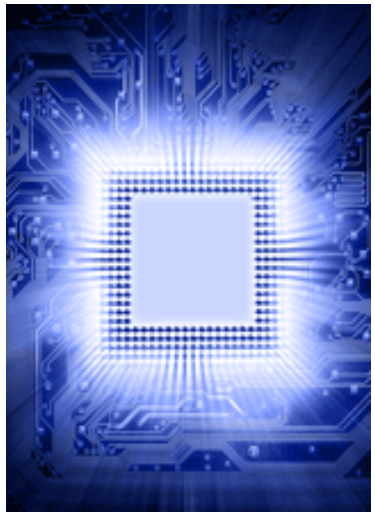
- 500桶酒，其中1桶是毒酒；48小时后要举行酒会；毒酒喝下去会在之后的第23-24小时内毒死人；国王决定用囚犯来试酒，不介意囚犯死多少，只要求用最少的囚犯来测试出哪一桶是毒酒，问需要最少需要多少囚犯才能保证找出毒酒？

提示：酒很毒，一滴就有效果。

本章作业

- 18，19，22，28，32，41，44，47，50

附录：　变长编码



熵编码

- 原理：基于概率分布特性，采用可变字长编码.
- 基本概念：

  设 $S_1, S_2, \ldots, S_n$ 是被编码的一组符号（消息），

  $P_1, P_2, \ldots, P_n$ 是它们出现的概率，

  $L_1, L_2, \ldots, L_n$ 是它们被编码后的码长

  则：

$$H = -\sum_{i=1}^{n} P_i * \log_2 P_i \quad (H\text{为熵, entropy})$$

$$L = \sum_{i=1}^{n} L_i * P_i \quad (L\text{为平均码长})$$

基本原理

- 定理1:

  H是理想的最小平均码长，即　L>=H

- 定理2:

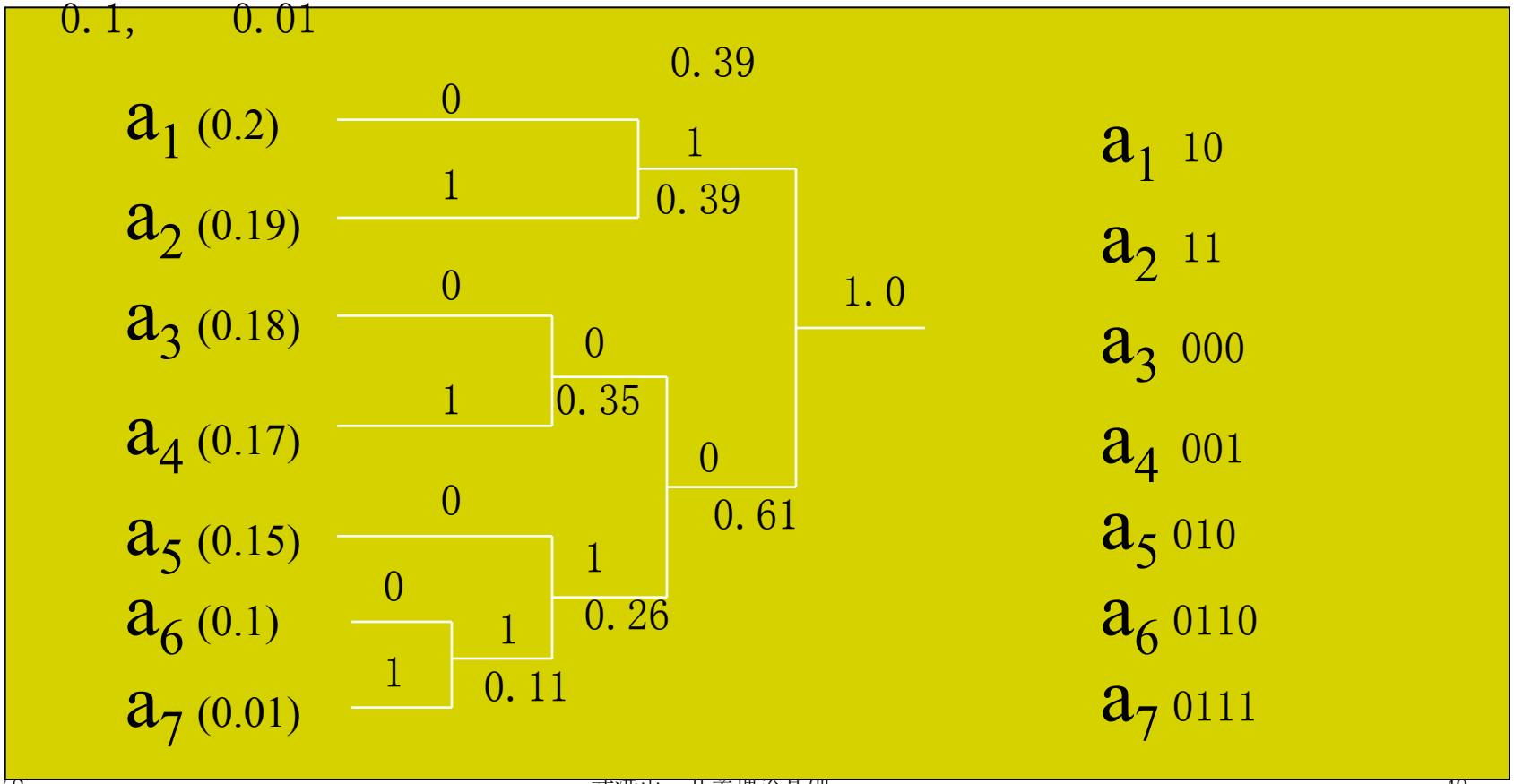
  最佳编码方法：出现概率大的符号用短字长码，出现概率小的符号用长码,码字长度与概率严格逆序排列，即可获得“最佳码”。

例：　ASCII键盘，莫尔斯电报码，文本压缩

Huffman 编码

例：　设有7个符号：　a_1, a_2, a_3, a_4, a_5, a_6, a_7

出现的概率是：　0.2,　0.19,　0.18,　0.17,　0.15,　0.1,　0.01



Huffman 编码的分析

- 每个编码均非其它码的前缀，因此唯一可译

（ a_1=10,a_2= 11, a_3= 000, a_4= 001, a_5= 010, a_6= 0110, a_7=0111 ）

11 010 10 001 10 11 10 0111

a_2 a_5 a_1 a_4 a_1 a_2 a_1 a_7

- 简单, 易实现
- 编码效率较高

（上例中，H=2.61， L=2.72，编码效率=2.61/2.72=96%）

Huffman 编码的分析

- 编码方案不唯一，码表必须存储/传输
- 必须预先知道信源的统计特性
- 由于每个被编码的符号都使用整数个二进位表示，编码效率还可以再提高。

回顾

- 构建求解问题的系统

回顾

- 构建求解问题的系统