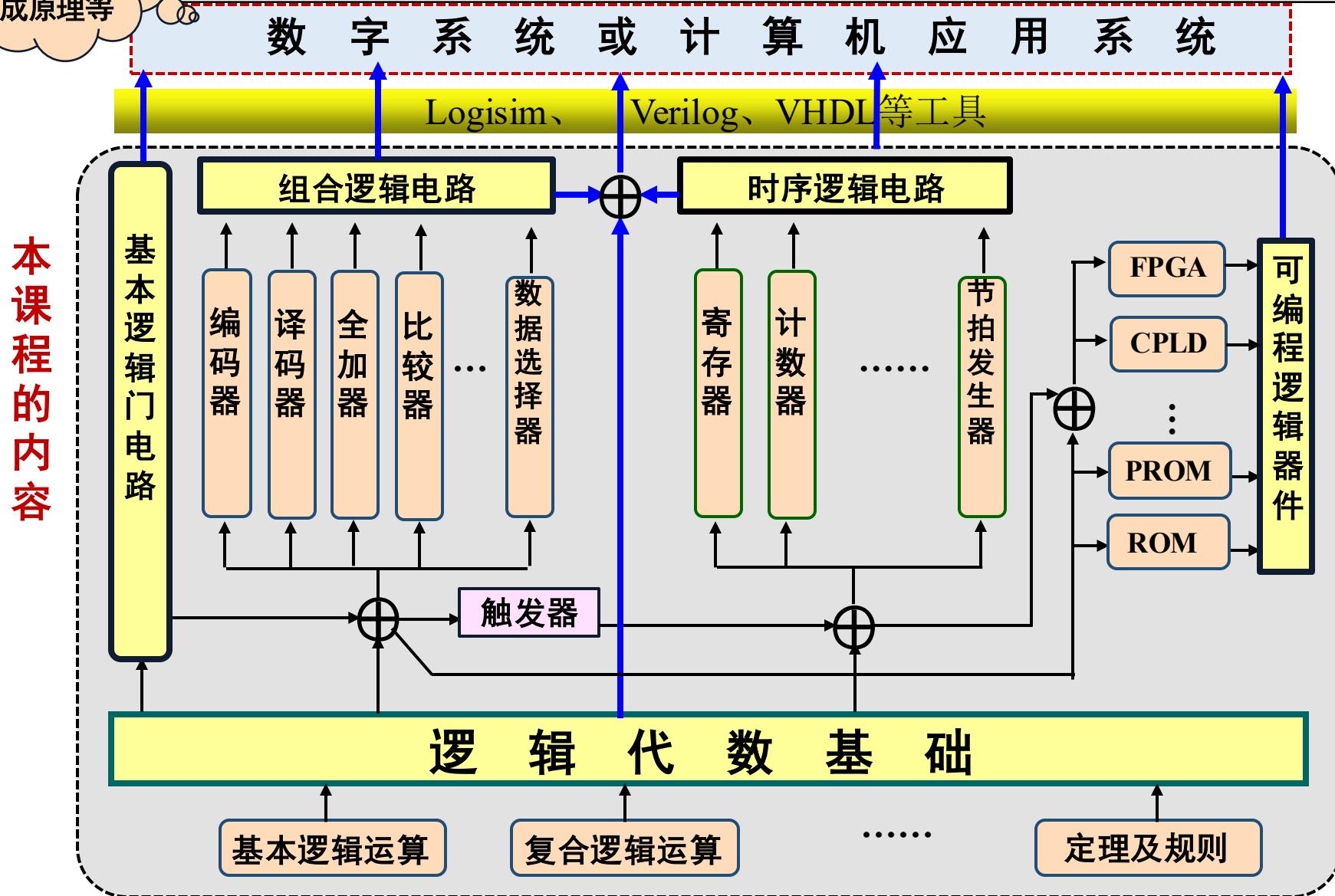


# 数字逻辑设计

高翠芸  
计算机科学与技术学院  
[gaocuiyun@hit.edu.cn](mailto:gaocuiyun@hit.edu.cn)

# 数字逻辑的知识脉络

后续课程：  
如计算机组成原理等



# 数制和码制（编码）

---

- 数制（表示数量）
- 编码（表示状态等——非数量，例如：学号等）
  - BCD码（BCD code）
  - 余3码（Excess-3 code）
  - 格雷码（Gray code）
  - 文字编码

# 数制和编码

---

- 数制
- 数字的表示

$$D = d_{p-1} \ d_{p-2} \dots \ d_1 \ d_0 . \ d_{-1} \ d_{-2} \dots \ d_{-n}$$

- LSB(least significant bit) : 最低有效位  $d_{-n}$
- MSB(most significant bit) : 最高有效位  $d_{p-1}$

# 按位计数制

任意十进制数D 可表示如下：

$$\begin{aligned} D &= d_{p-1} d_{p-2} \dots d_1 d_0 . d_{-1} d_{-2} \dots d_{-n} \\ &= \sum_{i=-n}^{p-1} d_i \times r^i \end{aligned}$$

推广：

$$\begin{aligned} B &= \sum b_i \times 2^i \\ H &= \sum h_i \times 16^i \end{aligned}$$

$r$  是计数制的基数 (Base or Radix) ,  $r^i$  为第  $i$  位的权。

- 按位计数制的特点
  - 1) 采用基数 (Base or Radix) , R进制的基数是R
  - 2) 基数确定数符的个数。如十进制的数符为：0、1、2、3、4、5、6、7、8、9，个数为10；二进制的数符为：0、1，个数为2
  - 3) 逢基数进一

# 十—二转换（整数）

---

$$\begin{aligned}
 R_{10} &= d_{p-1} d_{p-2} \dots d_1 d_0 \\
 &= d_{p-1} 2^{p-1} + d_{p-2} 2^{p-2} + \dots + d_1 2^1 + d_0 2^0 \\
 &= 2(d_{p-1} 2^{p-2} + d_{p-2} 2^{p-3} + \dots + d_1 2^0) + d_0
 \end{aligned}$$

除2得商：  $d_{p-1} 2^{p-2} + d_{p-2} 2^{p-3} + \dots + d_1 2^0$

$$\begin{aligned}
 &= 2(d_{p-1} 2^{p-2} + d_{p-2} 2^{p-3} + \dots) + d_1
 \end{aligned}$$

除2直到商为0

例：  $87 = (? \dots ?)_2$

$$=(1010111)_2$$

|   |    | 余数 |       |
|---|----|----|-------|
| 2 | 87 | 1  | $d_0$ |
| 2 | 43 | 1  | $d_1$ |
| 2 | 21 | 1  | $d_2$ |
| 2 | 10 | 0  | $d_3$ |
| 2 | 5  | 1  | $d_4$ |
| 2 | 2  | 0  | $d_5$ |
|   | 1  | 1  | $d_6$ |
|   | 0  |    |       |

# 十—二转换（小数）

$$R_{10} = 0. \ d_{-1} \ d_{-2} \dots \ d_{-n}$$

$$= d_{-1} 2^{-1} + d_{-2} 2^{-2} + \dots + d_{-n+1} 2^{-n+1} + d_{-n} 2^{-n}$$

$$= \underline{2^{-1}}(d_{-1} + d_{-2} 2^{-1} + \dots + d_{-n+1} 2^{-n+2} + d_{-n} 2^{-n+1})$$

乘2，去掉整数部分

$$\cancel{d_{-1}} + d_{-2} 2^{-1} + \dots + d_{-n+1} 2^{-n+2} + d_{-n} 2^{-n+1}$$

$$= \underline{2^{-1}}(d_{-2} + \dots + d_{-n+1} 2^{-n+3} + d_{-n} 2^{-n+2})$$

乘2，去掉整数部分，直到剩余部分为0

$$\text{例: } 0.4375 = (? \ \dots \ ?)_2$$

$$= (0.0111)_2$$

整数

|                 |   |       |    |
|-----------------|---|-------|----|
|                 | 0 | .4375 | *2 |
|                 | 0 | .875  | *2 |
| d <sub>-1</sub> | 1 | .75   | *2 |
| d <sub>-2</sub> | 1 | .5    | *2 |
| d <sub>-3</sub> | 1 | .0    |    |
| d <sub>-4</sub> |   |       |    |

# 二进制与八进制和十六进制之间的转换

---

| 十进制  | 0        | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 8        | 9        | 10       | 11       | 12       | 13       | 14       | 15       |
|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 二进制  | 000<br>0 | 000<br>1 | 001<br>0 | 001<br>1 | 010<br>0 | 010<br>1 | 011<br>0 | 011<br>1 | 100<br>0 | 100<br>1 | 101<br>0 | 101<br>1 | 110<br>0 | 110<br>1 | 111<br>0 | 111<br>1 |
| 八进制  | 0        | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 10       | 11       | 12       | 13       | 14       | 15       | 16       | 17       |
| 十六进制 | 0        | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 8        | 9        | A        | B        | C        | D        | E        | F        |

# 二进制与八进制和十六进制之间的转换

---

- 位数替换法：保持小数点不变，每位八进制数对应3位二进制数；每位十六进制数对应4位二进制数；

二进制转换为八进制或十六进制数时，从小数点开始向左右分组，在MSB(Most Significant Bit)前面和LSB(Least Significant Bit)后面可以加0；

八进制或十六进制转换为二进制数时，MSB前面和LSB后面的0不写；

例： $10111000.1101_2 =$

$$270.64_8 = B8.D_{16}$$

# 数制和码制（编码）

---

- 数制（表示数量）
- 编码（表示状态等——非数量，例如：学号等）
  - BCD码（BCD code）
  - 余3码（·Excess-3 code）
  - 格雷码（Gray code）
  - 文字编码

# 原码表示法

---

★最高有效位表示符号位 (Sign bit)

★**0 = 正, 1 = 负 (0 = plus, 1 = minus)**

★其余各位是该数的绝对值

★**01111111 = + 127**

**11111111 = - 127**

**00101110 = + 46**

**10101110 = - 46**

★零有两种表示 (+ 0、-0)

**00000000 = +0**

**10000000 = -0**

★8位二进制码能够表示的带符号十进制数中，  
最大的数是+127，而最小的数是-127。

★ **n**位二进制整数表示的范围：

**- (2<sup>n-1</sup> - 1) ~ + (2<sup>n-1</sup> - 1)**

# 反码表示法

---

\* 正数的二进制反码表示与原码相同

\* 负数的二进制反码表示：

在n位系统中，符号位不变，其余各位在原码基础上按位取反

# 补码表示法

---

\* 正数的二进制补码表示与原码相同

\* 负数的二进制补码如何求取？

反码(Ones' – Complement) + 1

(零只有一种表示)     $0=0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

\* 逐位取反

1 1 1 1 1 1 1 1

+1

\* 约定8位

$\underline{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}=0$

# 二进制编码



变色龙，拱猪，接龙 .....

玩法N多，本质上，就是54张牌在不同游戏规则下的组合而已

## ■ 编码

- BCD码
- 余3码
- 格雷码

编法N多，本质上，就是0和1在不同**编码规则**下的组合而已。

# BCD码

BCD码（Binary-Coded Decimal）也叫二-十进制编码，用**4位**二进制数表示**1位**十进制数

4位二进制码共有 $2^4=16$ 种码组，在这16种代码中，可以任选10种来表示10个十进制数码

每位二进制数都**带有权值**

- 根据权值不同，称其为：

8421BCD

2421BCD

4221BCD

...

| Decimal | 8421BCD |
|---------|---------|
| 0       | 0000    |
| 1       | 0001    |
| 2       | 0010    |
| 3       | 0011    |
| 4       | 0100    |
| 5       | 0101    |
| 6       | 0110    |
| 7       | 0111    |
| 8       | 1000    |
| 9       | 1001    |

# BCD码

| Decimal | 8421BCD | 2421BCD     | 4221BCD     | 5421BCD     |
|---------|---------|-------------|-------------|-------------|
| 0       | 0000    | 0000 (0000) | 0000 (0000) | 0000 (0000) |
| 1       | 0001    | 0001 (0001) | 0001 (0001) | 0001 (0001) |
| 2       | 0010    | 0010 (1000) | 0010 (0100) | 0010 (0010) |
| 3       | 0011    | 0011 (1001) | 0011 (0101) | 0011 (0011) |
| 4       | 0100    | 0100 (1010) | 0110 (1000) | 0100 (0100) |
| 5       | 0101    | 1011 (0101) | 1001 (0111) | 1000 (0101) |
| 6       | 0110    | 1100 (0110) | 1100 (1010) | 1001 (0110) |
| 7       | 0111    | 1101 (0111) | 1101 (1011) | 1010 (0111) |
| 8       | 1000    | 1110 (1110) | 1110 (1110) | 1011 (1011) |
| 9       | 1001    | 1111 (1111) | 1111 (1111) | 1100 (1100) |

# 余3码

| Decimal | 8421BCD | Excess-3 |
|---------|---------|----------|
| 0       | 0000    | 0011     |
| 1       | 0001    | 0100     |
| 2       | 0010    | 0101     |
| 3       | 0011    | 0110     |
| 4       | 0100    | 0111     |
| 5       | 0101    | 1000     |
| 6       | 0110    | 1001     |
| 7       | 0111    | 1010     |
| 8       | 1000    | 1011     |
| 9       | 1001    | 1100     |

- 无权码
- 自补性：对9的自补码
- 8421BCD码+“0011”

# 格雷码 (Gray Code)

---

- 由贝尔实验室的Frank Gray在1940年代提出的，1953年获得批准的专利“Pulse Code Communication”，当初是为了通信，后来则常用于模拟—数字转换中。
- 在一组数的编码中，若任意两个相邻的代码只有一位二进制数不同，则称这种编码为格雷码（Gray Code）
- 另外由于最大数与最小数之间也仅一位数不同，即“首尾相连”，因此又称循环码或反射码。
- 格雷码有多种编码形式——典型格雷码。

# 典型格雷码 (Gray code)

| Decimal | Binary | Gray code |
|---------|--------|-----------|
| 0       | 0000   | 0000      |
| 1       | 0001   | 0001      |
| 2       | 0010   | 0011      |
| 3       | 0011   | 0010      |
| 4       | 0100   | 0110      |
| 5       | 0101   | 0111      |
| 6       | 0110   | 0101      |
| 7       | 0111   | 0100      |
| 8       | 1000   | 1100      |
| 9       | 1001   | 1101      |
| 10      | 1010   | 1111      |

| Decimal | Binary | Gray code |
|---------|--------|-----------|
| 11      | 1011   | 1110      |
| 12      | 1100   | 1010      |
| 13      | 1101   | 1011      |
| 14      | 1110   | 1001      |
| 15      | 1111   | 1000      |

任何两位相邻编码  
只有1位码元不同

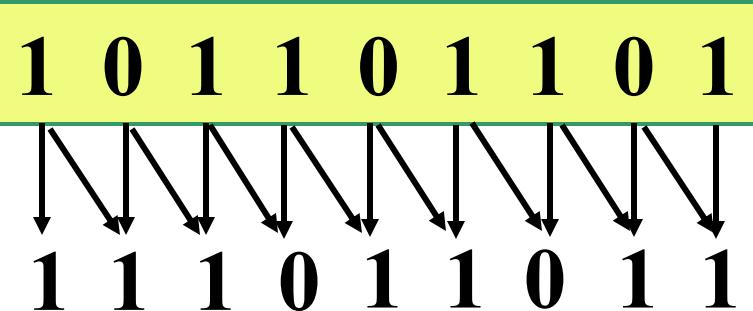
# 怎样计算任意给定的二进制数对应的典型格雷码？

---

## 1) 计算法

- 复制最高位
- 从最高位开始，俩俩比较相邻位：
  - 二者相同取 0
  - 二者不同取 1
- 转换前后数据的位宽不变

Binary:



Gray Code:

# 如何由n位典型格雷码写n+1位典型格雷码

## 2) 反射法

1位

|   |
|---|
| 0 |
| 1 |

2位

|     |
|-----|
| 0 0 |
| 0 1 |
| 1 1 |
| 1 0 |

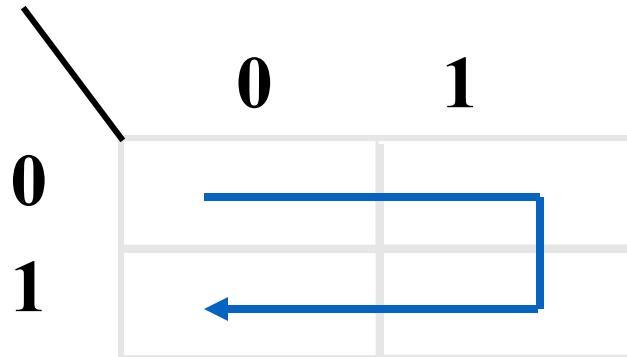


3位

|       |
|-------|
| 0 0 0 |
| 0 0 1 |
| 0 1 1 |
| 0 1 0 |
| 1 1 0 |
| 1 1 1 |
| 1 0 1 |
| 1 0 0 |

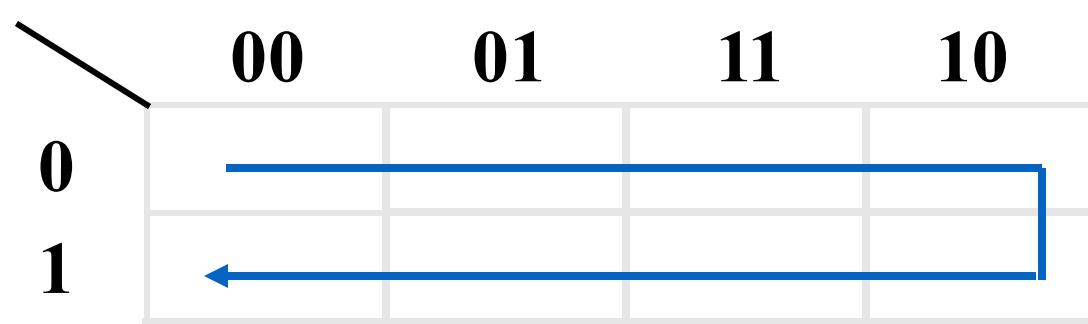
# 如何写n位典型格雷码

## 3) 图形法



2位格雷码

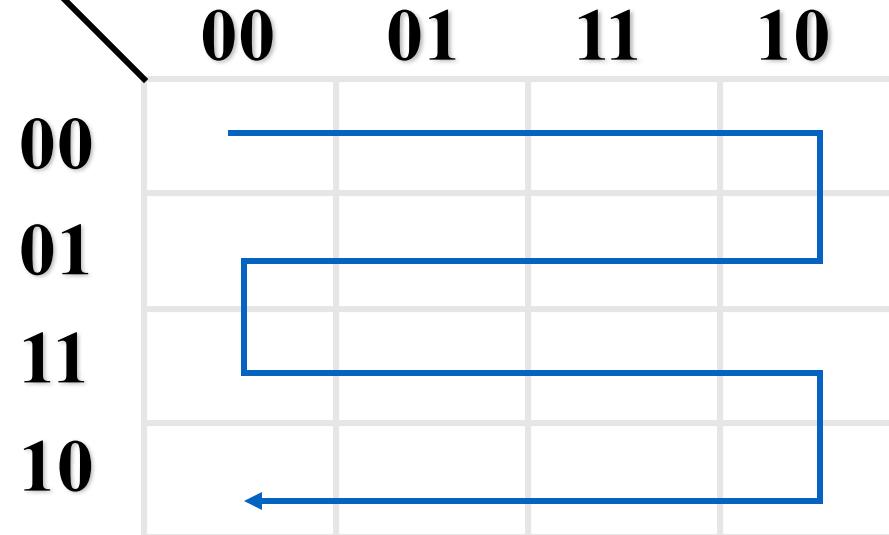
00、01、11、10



3位格雷码

000、001、011、  
010、110、111、  
101、100

# Gray Code

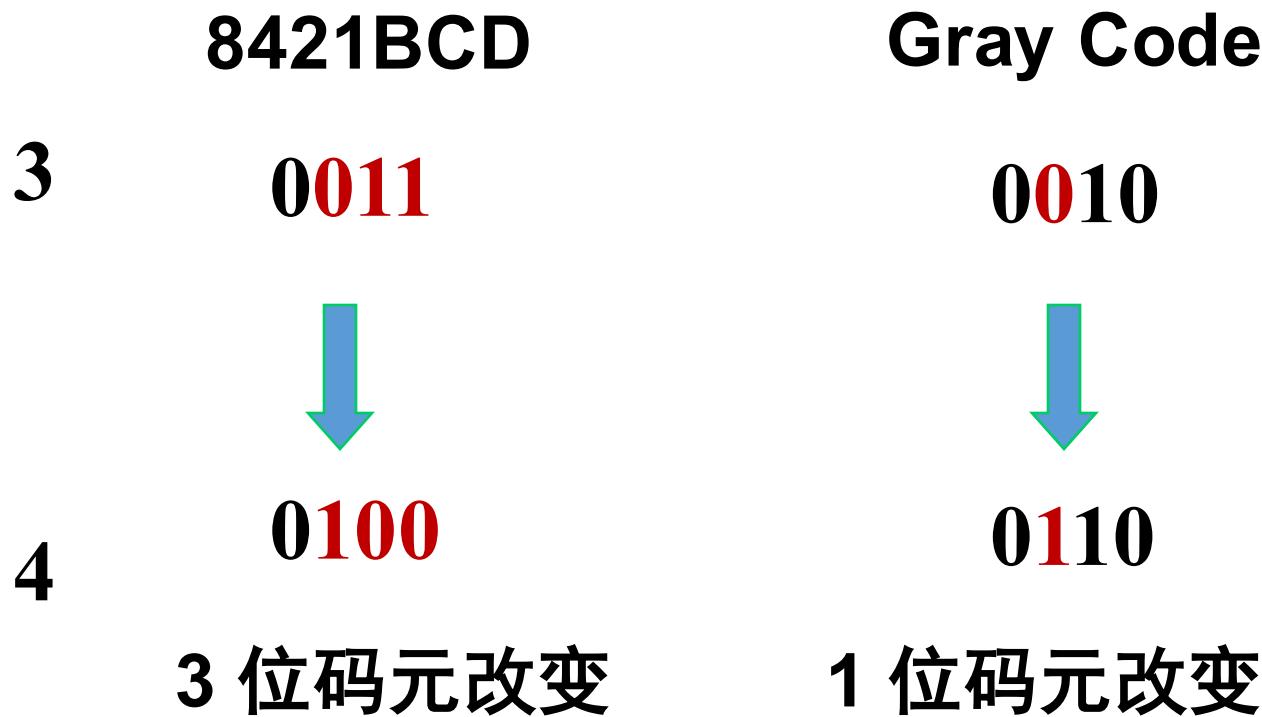


4位格雷码

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

# Gray Code

*Example*      十进制: 3→4



**Gray Code** ——连续变化时，比较可靠

# 文字编码

---

- ASCII 编码是最简单的西文编码方案
  - American Standard Code for Information Interchange
  - 8位
- GB2312、GBK、GB18030 是汉字字符编码方案的国家标准
- Unicode 是全球字符编码的国际标准

# ASCII码表

| ASCII值  | 控制字符    | ASCII值  | 控制字符 | ASCII值  | 控制字符 |
|---------|---------|---------|------|---------|------|
| 32(20H) | (space) | 64(40H) | @    | 96      | ,    |
| 33      | !       | 65(41H) | A    | 97(61H) | a    |
| 34      | "       | 66      | B    | 98      | b    |
| ...     | ...     | ...     | ...  | ...     | ...  |
| 48(30H) | 0       | 80      | P    | 112     | p    |
| ...     | ...     | ...     | ...  | ...     | ...  |
| 57(39H) | 9       | 89      | Y    | 121     | y    |
| 58      | :       | 90      | Z    | 122     | z    |
| ...     | ...     | ...     | ...  | ...     | ...  |
| 63      | ?       | 95      | _    | 127     | DEL  |

# 数制和编码小结

---

- 数制（表示数量）
- 编码（表示状态等——非数量，例如：学号等）
  - BCD码（BCD code）
  - 余3码（Excess-3 code）
  - 格雷码（Gray code）
  - 文字编码：ASCII、Unicode等

# 小 结

---

- 概述
- 课程简介
- 基本概念
- 数制
- 编码
  - BCD码 (BCD code)
  - 余3码 (Excess-3 code)
  - 格雷码 (Gray code)
  - .....

对于本章节不太清楚的地方?

A

模拟信号与数字信号的差异

B

常用数制的转换

C

BCD码

D

余3码

E

格雷码

F

没有不熟悉的地方

提交