数字世界精彩无限

数字逻辑与数字系统设计

张英涛

计算机科学与技术学院

哈尔滨工业大学

教　　师：张英涛

单　　位：计算机科学技术研究所

手　　机：15546215318

邮　　件： yingtao@hit.edu.cn

课程名称： 数字逻辑与数字系统设计

上课时间、地点：

2-12，周二5-6节，正心221

周四5-6节，正心221

不再布置课后作业，在 spoc中自行完成每周测验，课程结束后，将这部分成绩换算成课后作业成绩，满分10分。

hit数字逻辑

群号： 978698234

扫一扫二维码，加入群聊。

QQ

考核方式：

- **累加式评分：**

  平时作业5% + 大作业 15% + 实验 20% + 期末考试 50%

  + 研讨随堂测 5%

- **实验：** 共5 次，最后一次实验内容为大作业验收答辩.
- **期末考试：** 统一出题、统一批卷.

主要教材及参考书：

- 教材：Charles Roth 著，解晓萌 等译． 逻辑设计基础. 机械工业出版社.
- 参考书：王玉龙 编著. 数字逻辑实用教程. 清华大学出版社. 2012.

主要学习目标：

- 培养和提高计算机专业人才的抽象思维能力和逻辑设计能力
- 在掌握主要逻辑部件的分析和设计的基础上，能进一步运用标准的集成电路及可编程逻辑器件进行数字系统的设计
- 提高学生对硬件设计的兴趣，为进一步学习计算机组成原理等后续课程打下良好基础

1. 概述、数制系统
2. 布尔代数
3. 布尔代数的应用
4. 卡诺图

5. 多级门电路
6. 组合电路的分析与设计
7. 数据选择器和译码器等
8. 锁存器与触发器
9. 寄存器和计数器
10. 时序电路分析
11. 用MSI设计时序电路
12. 用触发器设计时序电路
13. 异步时序电路设计

Unit 1　概　述

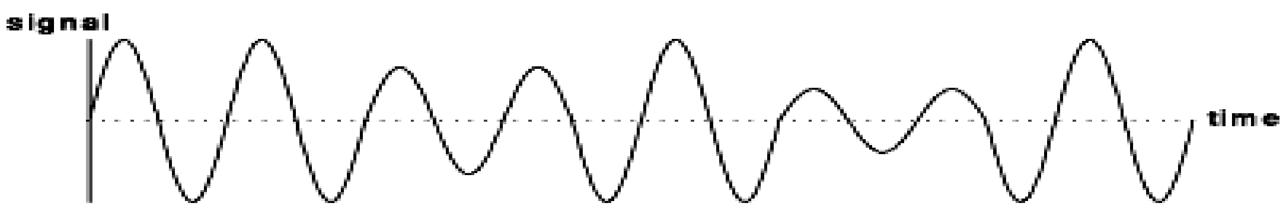
1 概 述

- 基本概念
- 数制系统
- 二进制编码
  - BCD码（BCD code）
  - 余3码（Excess-3 code）
  - 格雷码（Gray code）

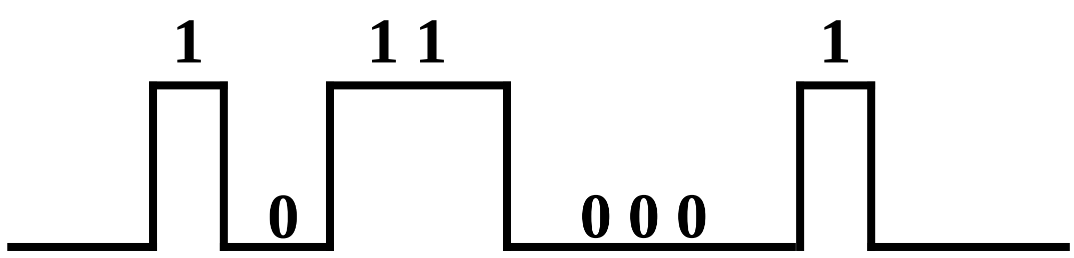
基本概念

1. 模拟信号——**在时间、幅度上是连续的**

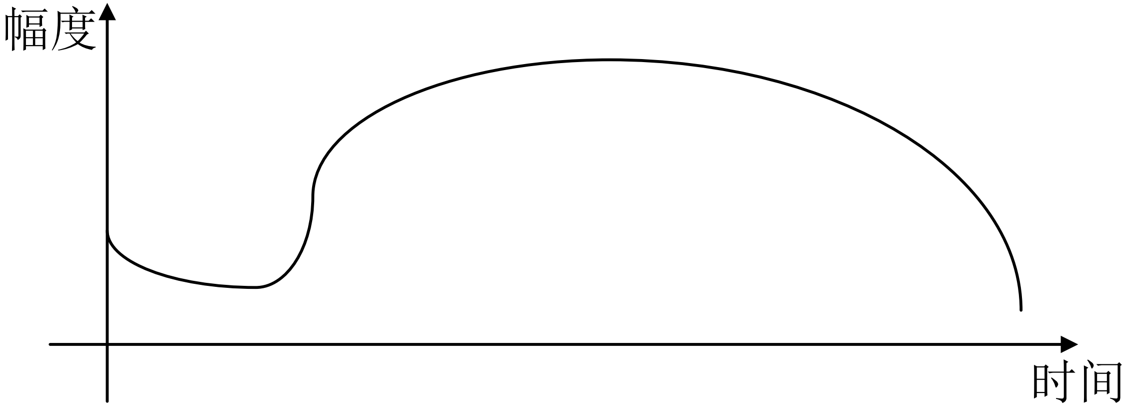
例子：**语音信号**



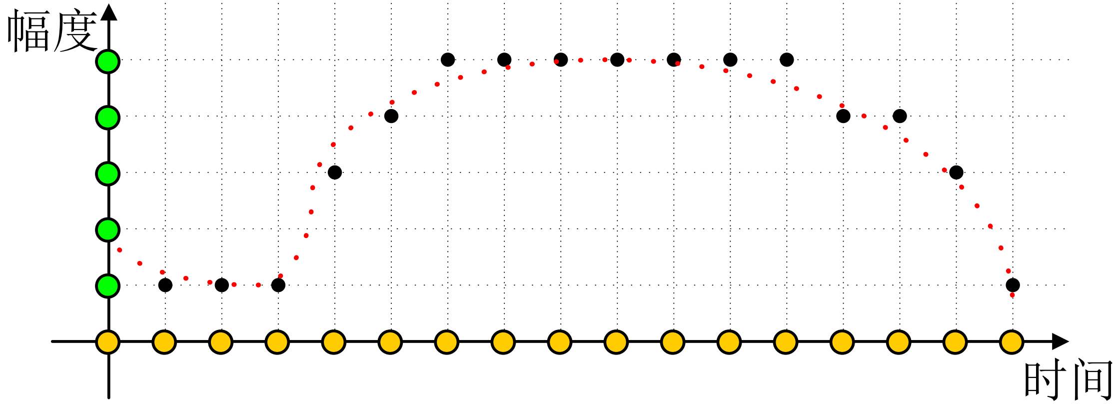
2. 数字信号——**在时间、幅度上是离散的**



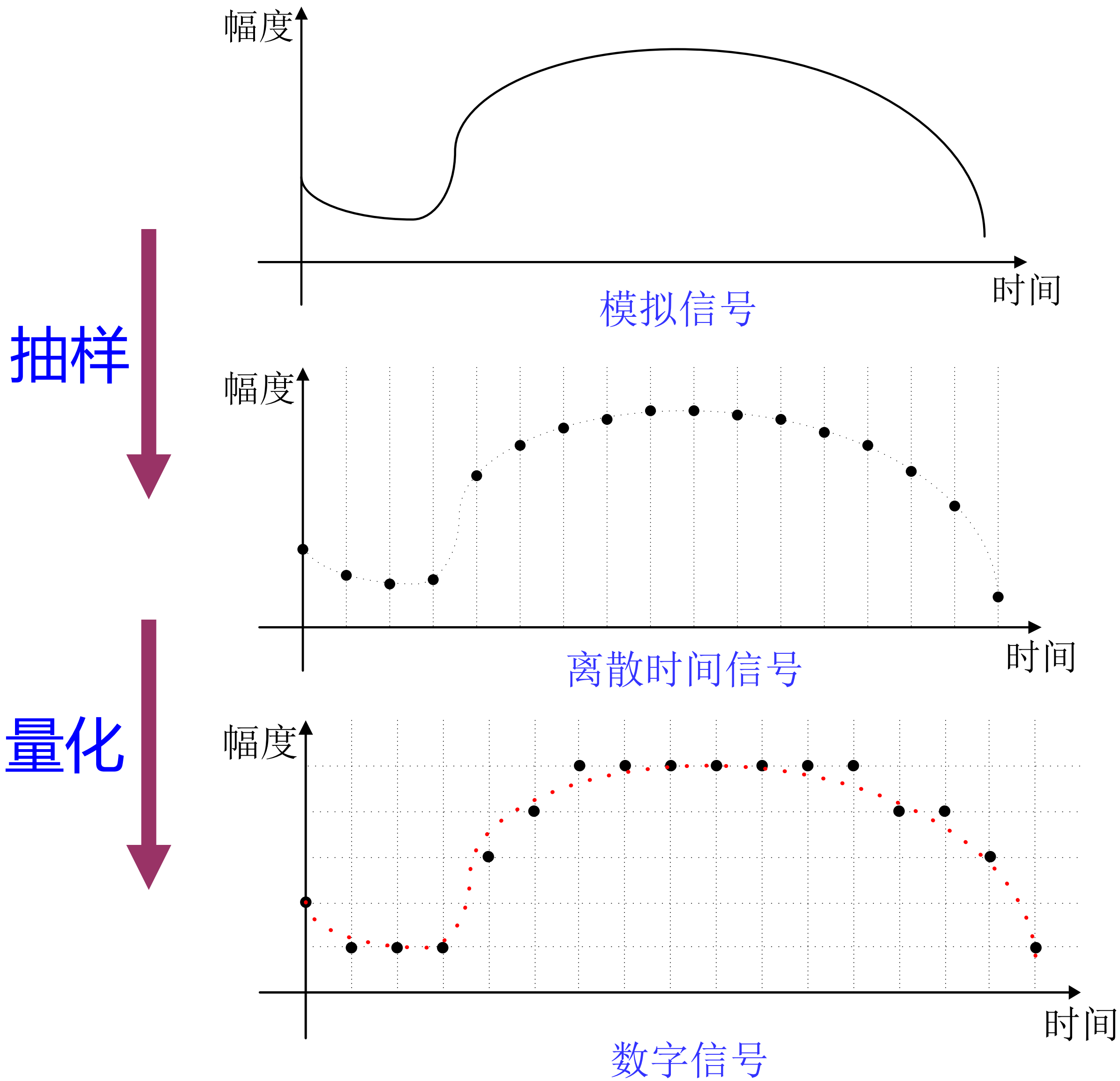
- **模拟信号：**时间和幅度都是连续的



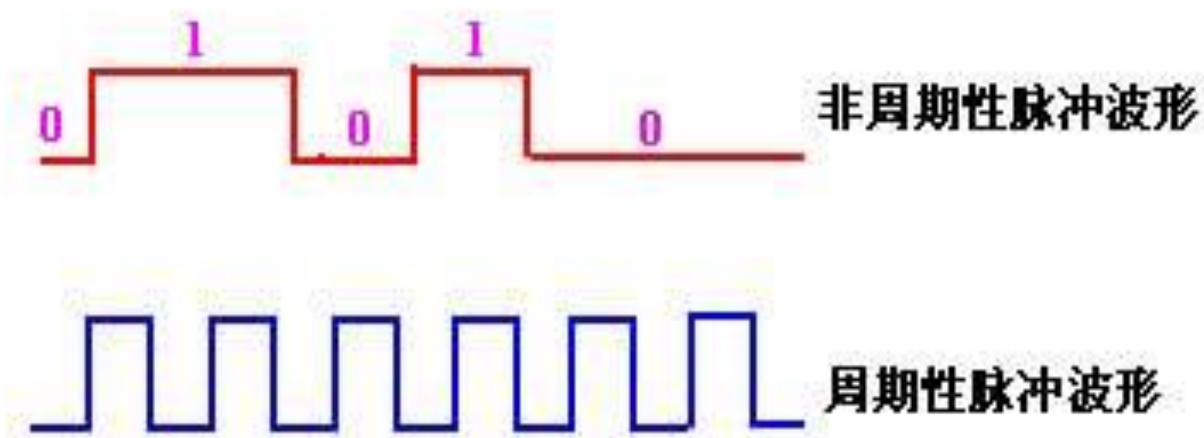
- **数字信号：**时间和幅度都是离散的



模拟信号和数字信号之间的转换



3. 脉冲波形

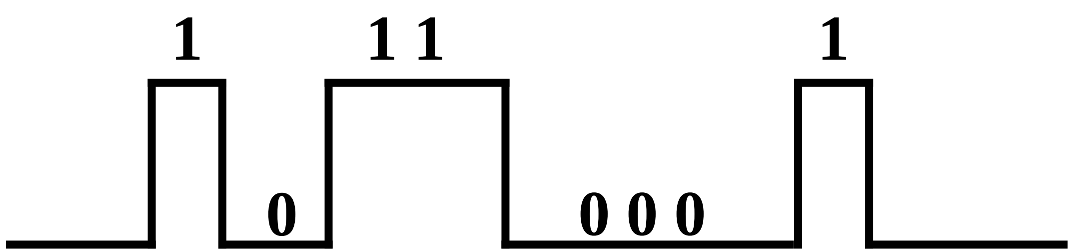


数字电路中的脉冲波形

基本概念

4. “0” 和 “1”

- 代表两种状态
  - **0**：表示某个范围的低电压　（如 **<0.4V**）
  - **1**：表示某个范围的高电压　（如 **>2.4V**）
- 在开关电路中
  - **0**：开关断开
  - **1**：开关闭合

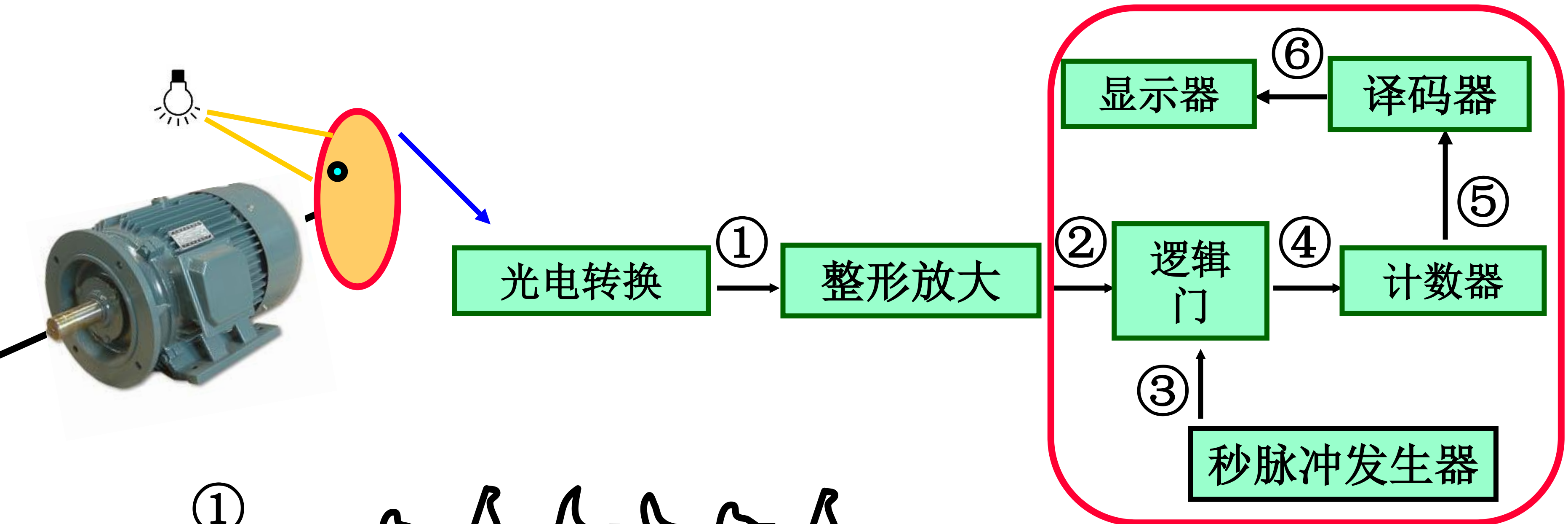


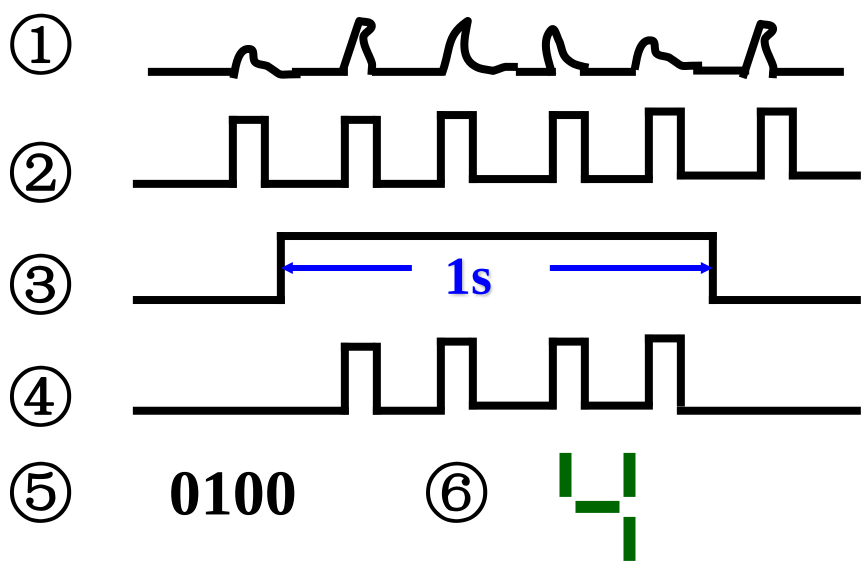
基本概念

5. 应用

- **逻辑运算**：与、或、非……
- **逻辑推理判断**
  - **举重比赛的评判电路**
  - **自动售饮料机电路**
  - ……

逻辑设计的例子





基本概念

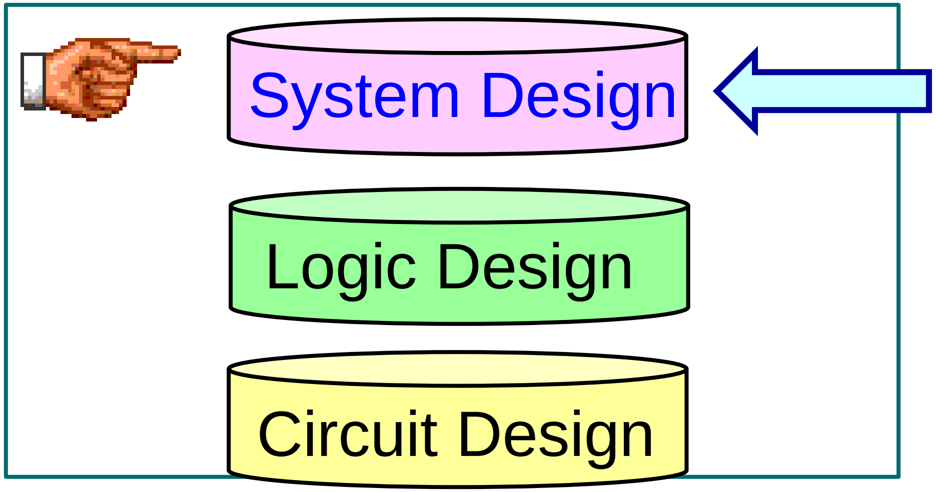
数字系统有何优点?

- 抗干扰能力强
- 更高的准确度和精确度

数字系统中为什么使用二进制?

- 电路简单
- 可靠稳定
- 便于计算机处理

数字系统设计的层次



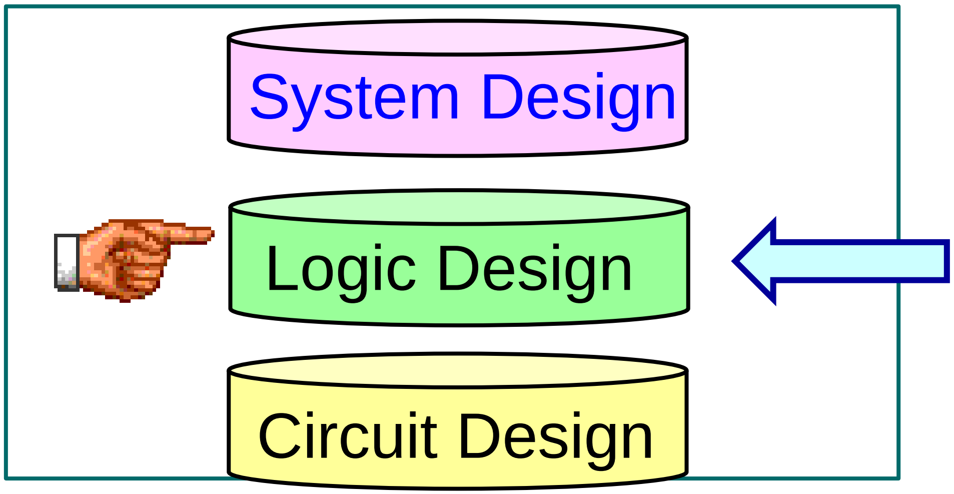
- 划分成子系统
- 确定各子系统特性
- 各子系统之间的互联及控制方式

例子: 计算机的系统设计--

•运算单元、控制单元、存储单元, 输入输出设备.......

• 各个子系统之间的互连及控制

数字系统设计的层次

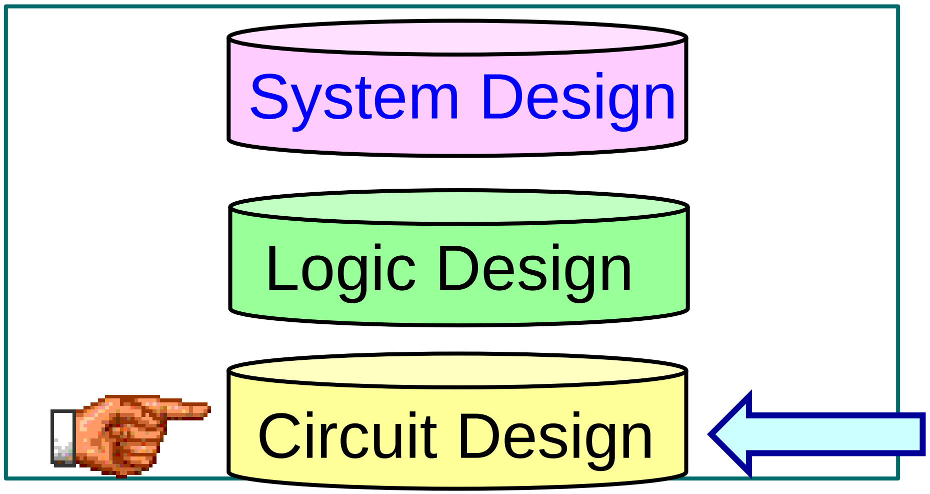


- 如何实现各子系统的逻辑功能
- 子系统中各功能模块的互连及控制

例子: 加法器的设计——

如何选用逻辑门和触发器设计实现

数字系统设计的层次



■ 确定特定电路的设计、器件的互连

例子: **逻辑门、触发器的设计--**

- 二极管、三极管、电阻.....
- 各逻辑器件的互连

目前的电路设计主流方法 —— 集成电路！

1 概 述

- 基本概念
- 数制系统
- 二进制编码
  - BCD码（BCD code）
  - 余3码（Excess-3 code）
  - 格雷码（Gray code）

相关概念

- **数制**：用一组符号和规则表示数的方法
- **基**：用来表示数的数码的集合称为基（如：0—9）
- **基数**：表示数的数码集合的大小称为基数(如：十进制的基数为10)
- **权** ：即数中某一位上的1所表示数值的大小（所处位置的价值）。如，十进制的123，1的位权是100，2的位权是10，3的位权是1。

二进制

- 1001 = ?

- 对于任意一个二进制数 N, 用位置记数法可表示为:

$$(N)_2=(a_{n-1}\ a_{n-2}\ \dots\ a_1\ \ a_0 .\ a_{-1}\ a_{-2}\dots\ a_{-m})_2$$

用权展开式表示为

$$(N)_2 = a_{n-1}\times 2^{n-1}+a_{n-2}\times 2^{n-2} +\dots+ a_1\times 2^1+a_0\times 2^0+a_{-1}$$

$$\times 2^{-1}+a_{-2}\times 2^{-2}+\dots+a_{-m}\times 2^{-m}$$

$$= \sum_{i=-m}^{n-1} a_i \times 2^i$$

a_i=0或1； n为整数部分的位数；m为小数部分的位数.

二进制的特点

- 只有两个数码, 很容易用物理器件来实现
- 运算规则简单
- 使用逻辑代数这一数学工具
- 节省设备

同样表达1000的个数，二进制比十进制节省设备！

1）设n：数的位数， R：基数，

最多可表达的数： R^n个；　　R^n个数所需设备量： nR。

例：$n=3$，$R=10$，$(R)_{10}=10^3=1000$

$nR=3\times10=$ 30

令$R^n=1000$ 且 $R=2$，即 $2^n=1000$，得 $n=10$ $(R^n=1024)$

$nR=10\times2=$ 20

2）R^n一定时，nR的极小值=?

R=e=2.718

数制转换——二进制 & 十进制

1.二进制数→十进制数：

- *按权展开式在十进制数域中计算*

例：

$$(11010.101)_2 = 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0$$

$$+ 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3}$$

$$= 16 + 8 + 2 + 0.5 + 0.125$$

$$= (26.626)_{10}$$

2.十进制数→二进制数：

•方法：整数部分，除2取余法

例：$(58)_{10} = (a_{n-1}\ a_{n-2}\cdots a_1\ a_o)_2 = (111010)_2$

$$(x)_{10} = a_{n\text{-}1}2^{n\text{-}1} + a_{n\text{-}2}2^{n\text{-}2} + \cdots + a_1 2^1 + a_0$$

$$=2\left(a_{n\text{-}1}2^{n\text{-}2} + a_{n\text{-}2}2^{n\text{-}3} + \cdots + a_1\right) + a_0$$

$$\left(\left\lfloor \frac{x}{2} \right\rfloor\right)_{10} = 2\left(a_{n\text{-}1}2^{n\text{-}3} + a_{n\text{-}2}2^{n\text{-}4} + \cdots + a_2\right) + a_1$$

$$\left(\left\lfloor \frac{x}{2^2} \right\rfloor\right)_{10} = 2\left(a_{n\text{-}1}2^{n\text{-}4} + a_{n\text{-}2}2^{n\text{-}5} + \cdots + a_3\right) + a_2$$

$$\vdots$$

$$\left(\left\lfloor \frac{x}{2^{n-2}} \right\rfloor\right)_{10} = 2\left(a_{n\text{-}1}\right) + a_{n\text{-}2}$$

数制转换——二进制 & 十进制

2.十进制数→二进制数：

方法：小数部分，乘2取整法

例：$(0.625)_{10} = a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m+1}$

$(1.25)_{10} = a_{-1} + (a_{-2} \times 2^{-1} + \cdots + a_{-m} \times 2^{-m+1})$　　得a_{-1}=1

$(0.5)_{10} = a_{-2} + (a_{-3} \times 2^{-1} + \cdots + a_{-m} \times 2^{-m+2})$　　得a_{-2}=0

$(1.00)_{10} = a_{-3} + (a_{-4} \times 2^{-1} + \cdots + a_{-m} \times 2^{-m+3})$　　得a_{-3}=1

则　$(0.625)_{10} = (0.101)_2$

注意：不能进行精确转换的情况， 如 0.423

数制转换——二进制 & 八进制

1.二进制数→八进制数：

方法：每三位一组进行转换

例：

二进制：	010	101	111	·	000	101	101	100
八进制：	2	5	7	·	0	5	5	4

$$(10\ 101\ 111\ .\ 000\ 101\ 101\ 1)_2 = (257.0554)_8$$

数制转换——二进制 & 八进制

2.八进制→二进制：

方法：每一位转化为三位

例：

八进制：	2	5	7	·	0	5	5	4
二进制：	010	101	111	·	000	101	101	100

$$(257.0554)_8 = (10\ 101\ 111\ .\ 000\ 101\ 101\ 1)_2$$

数制转换——二进制 & 十六进制

1.二进制→十六进制：

方法：每四位一组进行转换

例：

二进制：	1010	1111	·	0001	0110	1100
十六进制：	A	F	·	1	6	C

$$(1010\ 1111\ .\ 0001\ 0110\ 11)_2 = (AF.16C)_{16}$$

数制转换——二进制 & 十六进制

2.十六进制→二进制：

方法：每一位转化为四位

例：

十六进制：	A	F	·	1	6	C
二进制：	1010	1111	·	0001	0110	1100

$$(1010\ 1111\ .\ 0001\ 0110\ 11)_2 = (AF.16C)_{16}$$

带符号的二进制数的表示

最高位表示符号，0表示正，1表示负，称为符号位

其余位表示数的绝对值，称为数值位

符号位 数值位

0/1	0	0	0	0	1	1	0

	符号位	数值位		
		原码	反码	补码
正数	0	绝对值的原码	绝对值的原码	绝对值的原码
负数	1	绝对值的原码	绝对值的反码	绝对值的补码

带符号的二进制整数，字长 = 4

+N	**正整数** (原码、反码与补码)
+0	0000
+1	0001
+2	0010
+3	0011
+4	0100
+5	0101
+6	0110
+7	0111

带符号的二进制整数，字长 = 4

+N	正整数 (原码、反码与补码)		−N	负整数		
				原码	反码	补码
+0	0000		-0	1000	1111	——
+1	0001		-1	1001	1110	1111
+2	0010		-2	1010	1101	1110
+3	0011		-3	1011	1100	1101
+4	0100		-4	1100	1011	1100
+5	0101		-5	1101	1010	1011
+6	0110		-6	1110	1001	1010
+7	0111		-7	1111	1000	1001
			-8	——	——	1000

原码

原码： **正数的符号位为 0, 负数的符号 位为1, 其余各位表示数值部分。**

例： $N_1 = +011$　　$N_2 = -101$

$[N_1]_{原} = 0\ 011$　　$[N_2]_{原} = 1\ 101$

特点:

1. 真值0有两种原码表示形式, 即

   $[+0]_{原} = 00\ldots0$，$[-0]_{原} = 10\ldots0$

2. 表示范围：-7—+7（4位）
3. 加减法时需要对符号单独处理

	原码
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111
-8	——

反码

反码： 正数的反码表示与原码表示相同，负数的符号位为1，其余各位将其原码按位取反。

例： $N_1 = +011$ $N_2 = -101$

$[N_1]_{反} = 0011$ $[N_2]_{反} = 1010$

特点:

1. 真值0有两种反码表示形式, 即

$[+0]_{反} = 00\ldots0$， $[-0]_{反} = 11\ldots1$

2. 表示范围：-7—+7（4位）

3. 加减运算时，符号可参加运算

	原码	反码
-0	1000	1111
-1	1001	1110
-2	1010	1101
-3	1011	1100
-4	1100	1011
-5	1101	1010
-6	1110	1001
-7	1111	1000
-8	——	——

补码

补码： 正数的补码表示与原码表示相同，负数的符号位为1，其余各位是在其反码的末位加“1”。

例： $N_1 = +011 \quad N_2 = -101$

$[N_1]_{补} = 0011 \quad [N_2]_{补} = 1011$

特点:

1. 真值0只有一种补码表示形式, 即
   $[+0]_{补}=[-0]_{补}= 1\,1\ldots1+1=\underbrace{1}_{丢弃}00\ldots0$
2. 表示范围：-8—+7（4位整数）
3. 加减运算时，符号可参加运算

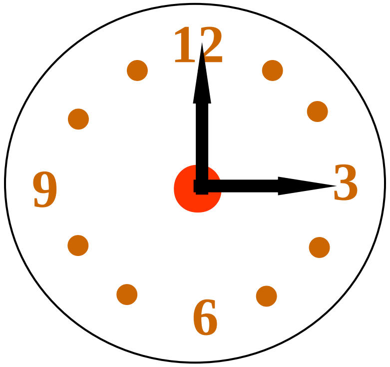
	原码	补码
-0	1000	——
-1	1001	1111
-2	1010	1110
-3	1011	1101
-4	1100	1100
-5	1101	1011
-6	1110	1010
-7	1111	1001
-8	——	1000

补码的补充说明

- 负数的补码与其真值的绝对值构成了以 2^L（L为计算机的字长）为模的“模数系统”或“同余”结构的代数系统
  - 同余：在某一模数系统中，模数为 n，如果 a、b的余数相同，则称a、b模n同余。
  - 在模 n 的系统中，N 与 $n-N$ 是一对互补的数，利用其特点可把减法变成加法运算

例：(1) 在模12的系统中，3与 12-3=9互补，
$11 - 3 = 11 + 9$,　$11 - 9 = 11 + 3$

(2) 在模16的系统中，9与 $16-9 = 7$互补，
$10 - 9 = 10 + 7$,　$10 - 7 = 10 + 9$



补码的补充说明

- 负数的补码与其真值的绝对值构成了以 2^L（L为计算机的字长）为模的“模数系统”或“同余”结构的代数系统。

当 N为负数：

$\boxed{取反加1}$ → $\boxed{[N]_{补}=2^n-|N|}$ → $\boxed{[N]_{补}与|N|\ 互补}$

→ $\boxed{x - |N| = x + [N]_{补}}$

例:

令L = 4，$[-3]_{补} = 1101$，$(1101)_2 = 13 = 2^4 - 3$, 即

$(-3)_{补}$ 与 3 互补， $x - 3 = x + [3]_{补}$。

1 概 述

- 基本概念
- 数制系统
- 二进制编码
  - BCD码（BCD code）
  - 余3码（Excess-3 code）
  - 格雷码（Gray code）

1. BCD code

- 也叫二—十进制编码
- 用**4**位二进制数表示**1**位十进制数
- 每位二进制数都带有权值
  - 根据权值不同，称其为：

    8421BCD，5421BCD，4221BCD…….

BCD Codes

十进制	8421code	2421code	4221code	5421code
0	0000	0000	0000	0000
1	0001	0001	0001	0001
2	0010	0010（1000）	0010（0100）	0010
3	0011	0011（1001）	0011（0101）	0011
4	0100	0100（1010）	1000（0110）	0100
5	0101	1011（0101）	0111（1001）	1000（0101）
6	0110	1100（0110）	1100（1010）	1001（0110）
7	0111	1101（0111）	1101（1011）	1010（0111）
8	1000	1110	1110	1011
9	1001	1111	1111	1100

2. 余3码（Excess-3 code）

Decimal	8421 BCD	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

0+3

- 8421code + “0011”
- 无权码

3. 典型格雷码（Gray code）

- 无权码

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
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

Gray Code

例：　　　十进制：3→4

	8421BCD	Gray Code
3	0011	0010
4	0100	0110
	3 位码元改变	1 位码元改变

Gray Code的优点：可靠性高

如何写典型 Gray Code

1) 计算法（利用二进制数写典型Gray code）

- 位宽不变
- 第一位不变
- 与左边的邻居比特相比较:
  - 相同—— 0
  - 不同——1.

$(365)_{10}$	——	1 0110 1101
Gray code	——	1 1101 1011

$(366)_{10}$	——	1 0110 1110
Gray code	——	1 1101 1001

二进制数: 1 0 1 1 0 1 1 0 1

Gray Code: 1 1 1 0 1 1 0 1 1

如何写典型 Gray Code

2) 反射法（由 n位 Gray code 写n+1位 Gray code ）

1-bit

0
1

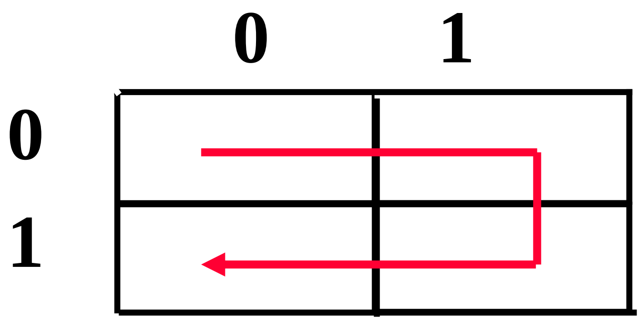
2-bit

0	0
0	1
1	1
1	0

3-bit

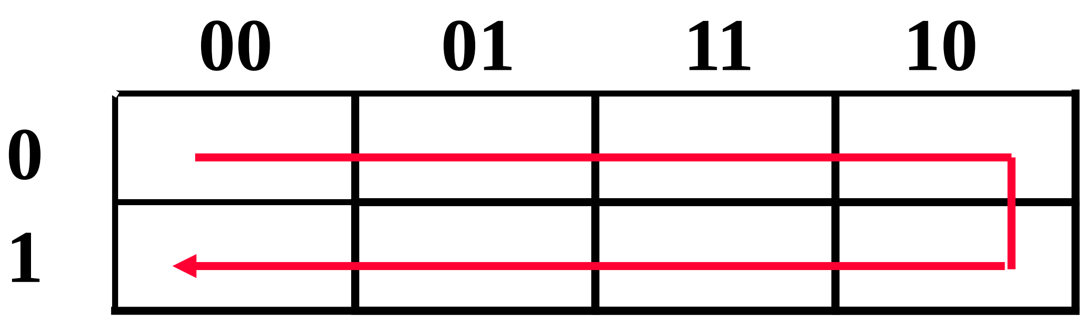
0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

3) 图形法（Graphics method）



2-bit Gray code

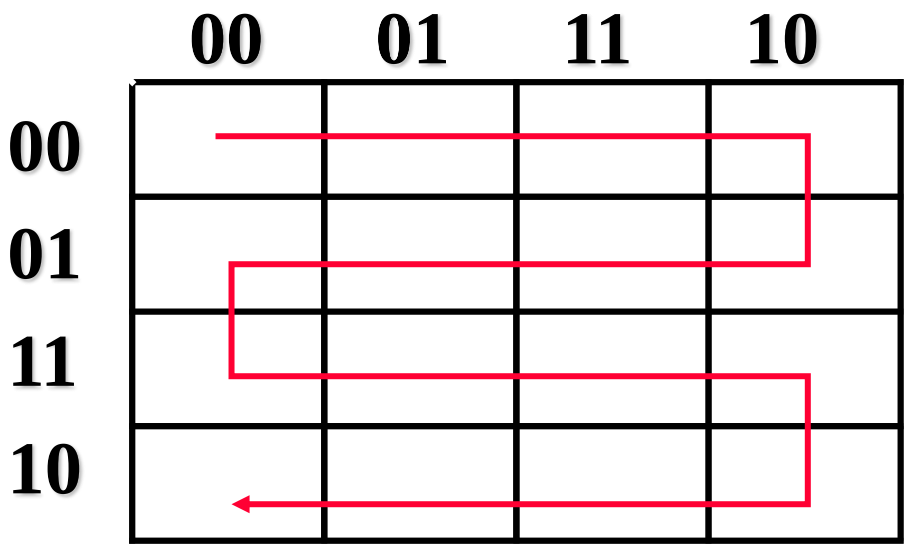
00、01、11、10



3-bit Gray code

000、001、011、010 、110、111、101、100

如何写典型 Gray Code



4-bit Gray code

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

十进制	二进制	GREY1		GREY2
0	0000	0000		0000
1	0001	0001		0001
2	0010	0011		0011
3	0011	0010		0010
4	0100	0110		0110
5	0101	0111		1110
6	0110	0101		1010
7	0111	0100		1011
8	1000	1100		1001
9	1001	1101		1000
10	1010	1111		
11	1011	1110		
12	1100	1010		
13	1101	1011		
14	1110	1001		
15	1111	1000		

反射

循环

格雷码的应用：

- 卡诺图
- 循环计数

二进制编码

Decimal Digit	8-4-2-1 Code (BCD)	6-3-1-1 Code	Excess-3 Code	2-out-of-5 Code	Gray Code
0	0000	0000	0011	00011	0000
1	0001	0001	0100	00101	0001
2	0010	0011	0101	00110	0011
3	0011	0100	0110	01001	0010
4	0100	0101	0111	01010	0110
5	0101	0111	1000	01100	1110
6	0110	1000	1001	10001	1010
7	0111	1001	1010	10010	1011
8	1000	1011	1011	10100	1001
9	1001	1100	1100	11000	1000

1 概 述

- 基本概念
- 数制系统
- 二进制编码
  - BCD码（BCD code）
  - 余3码（Excess-3 code）
  - 格雷码（Gray code）