《数字电子技术基础》绪论

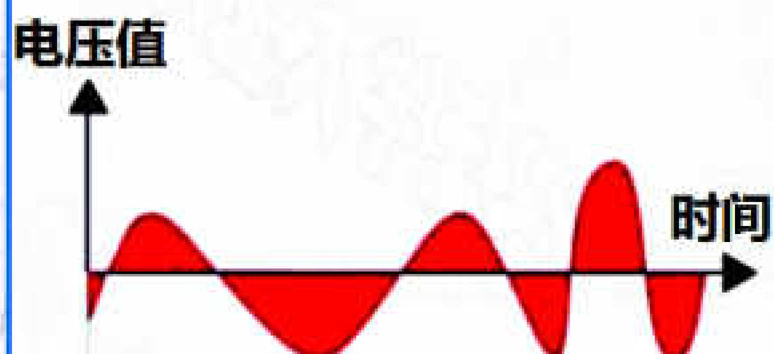
——王文俊

一、数字量与模拟量

- 1、模拟量与数字量

模拟量

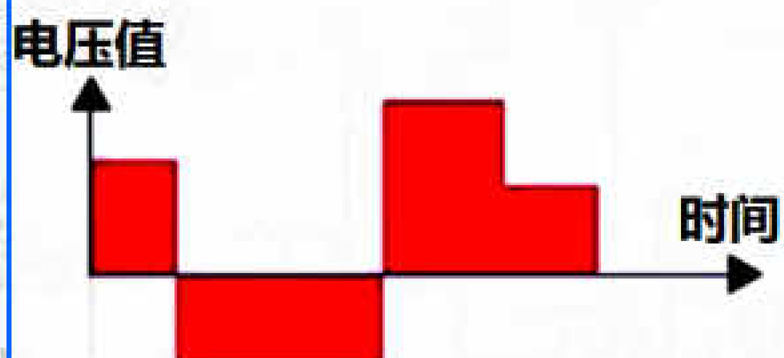
- 在时间上和数量上都是连续的物理量。
- 如：高度、重量等



模拟信号

数字量

- 在时间上和数量上都是离散的、不连续的物理量（存在最小的数量单位Δ）。
- 如：数字、字母等



数字信号

• 2、模拟电路与数字电路

电路

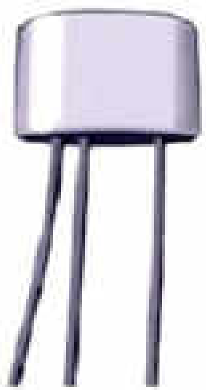
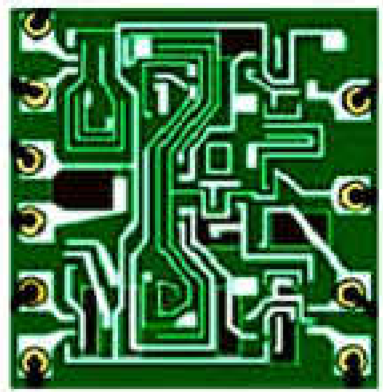
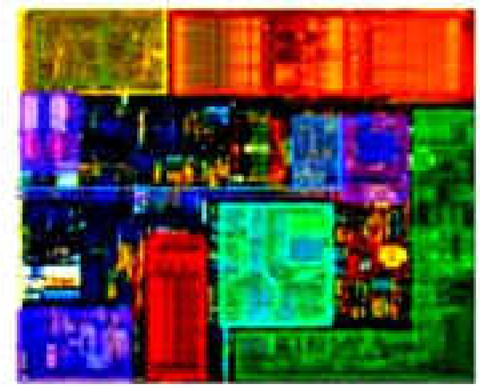
- 模拟电路
  - 使用、处理模拟信号的电路
- 数字电路
  - 使用、处理数字信号的电路

数字电路和模拟电路的工作信号、研究对象、分析/设计方法以及所用的数学工具都有显著不同。

二、电子技术

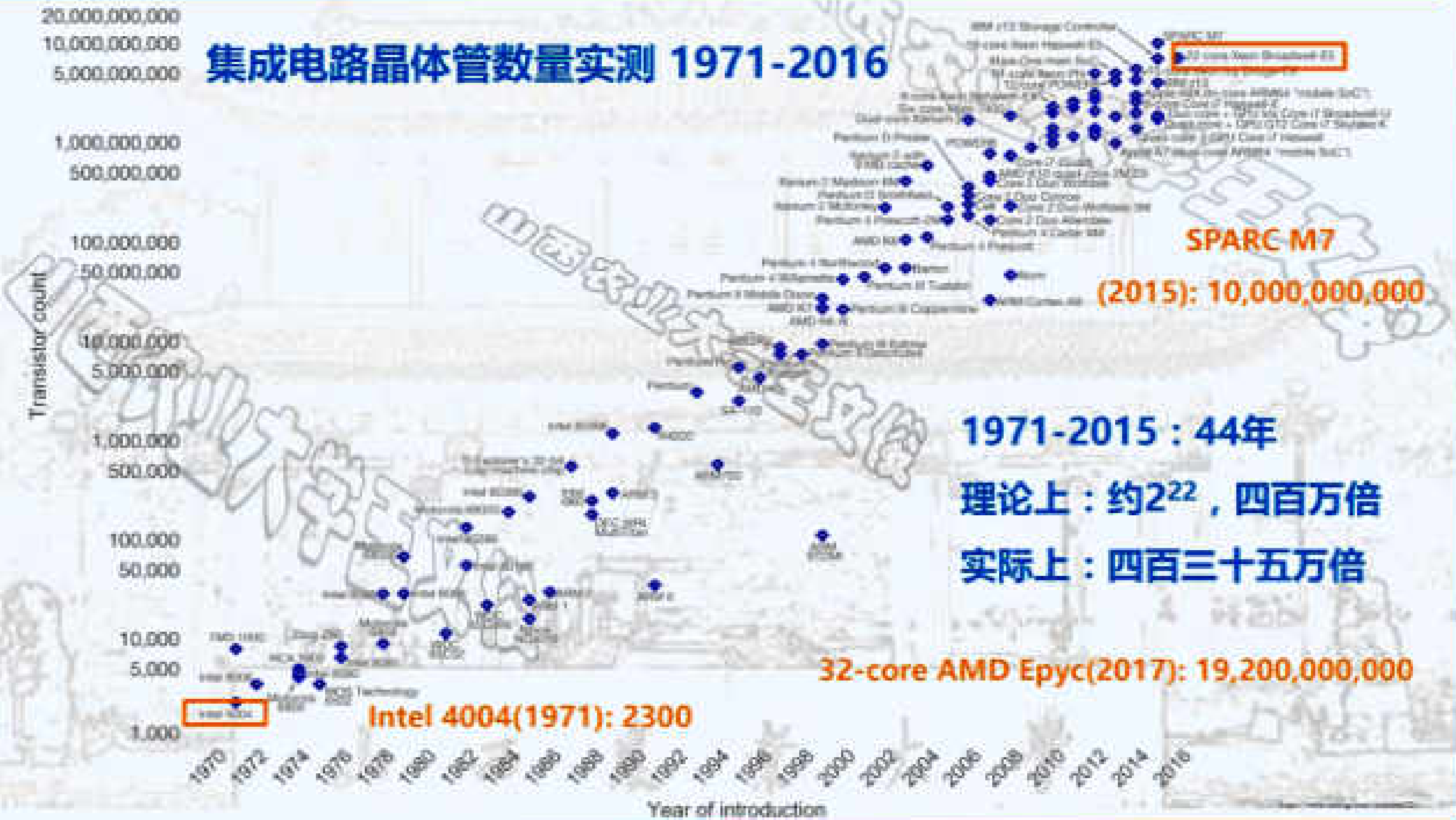
- 1、电子技术的发展历程

电子技术是一门研究电子器件及其应用的学科。

电子管时代 (1905~1948)	晶体管时代 (1948~1959)	集成电路时代 (1950~)	
 电子管 (1904)	 晶体管 (1948)	 中/小规模集成电路（1950年代）	 大规模/超大规模集成电路（1970年代）

- 2、摩尔定律

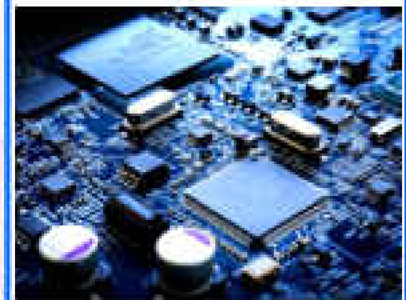
英特尔创始人之一 戈登·摩尔	• 集成电路上可容纳的晶体管数目，约每隔24个月便会增加一倍，性能也将提升一倍。



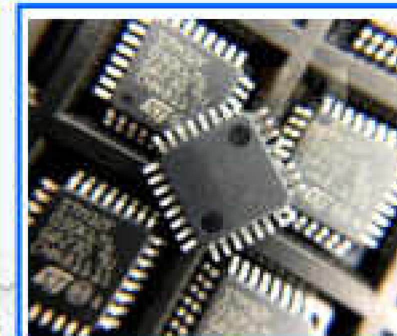
- 3、计算机的构成

个人电脑

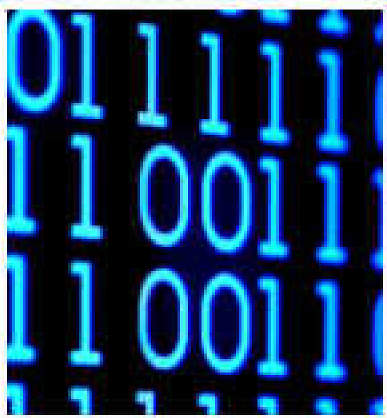
电路板



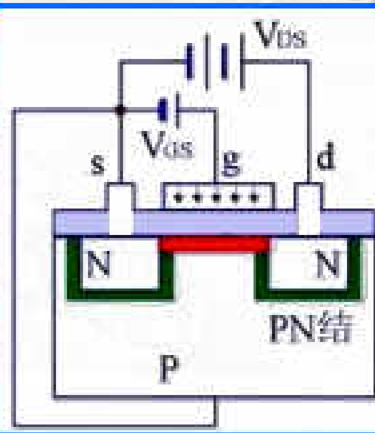
集成电路



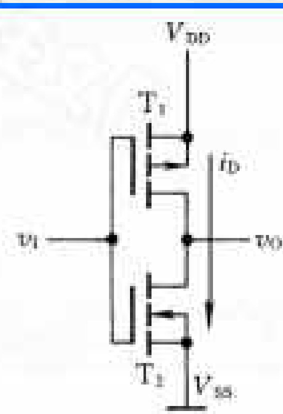
模块



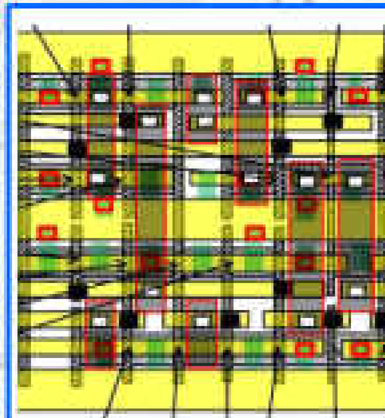
数字信号



MOSFET



门电路

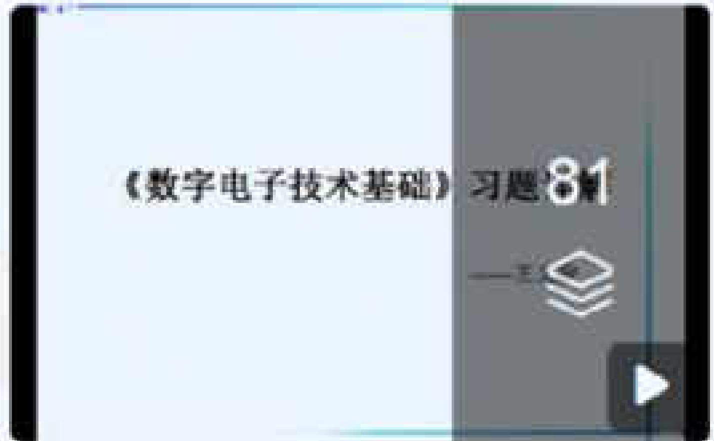


单元

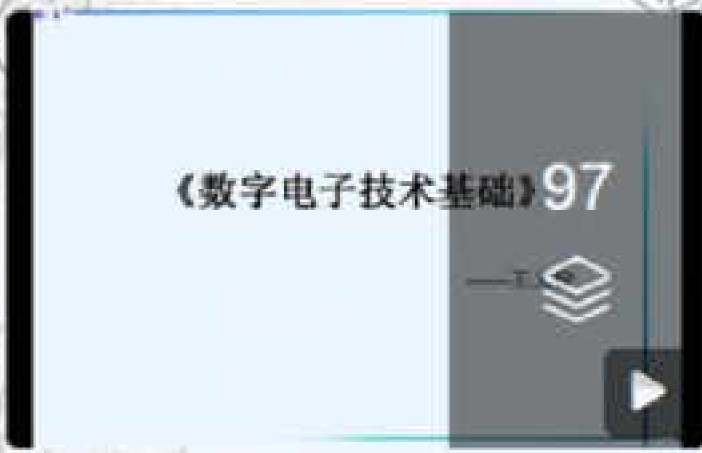
三、课程内容

数学基础	一、数制和码制	二、逻辑代数基础
基本电路	三、门电路	
逻辑电路	四、组合逻辑电路	
	五、半导体存储电路	六、时序逻辑电路
其他电路	七、脉冲波形的产生和整形电路	八、数-模和模-数转换

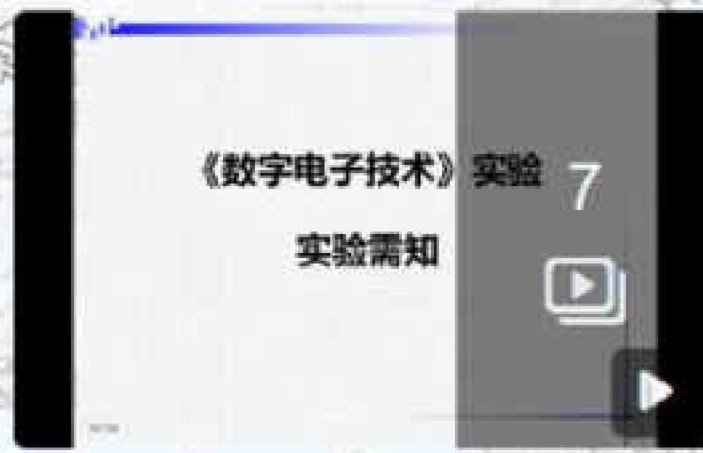
山西农业大学王文俊
UID: 456163933



合集·《数字电子技术基础》习题详解



合集·《数字电子技术基础》（97集全）



数字电子技术实验

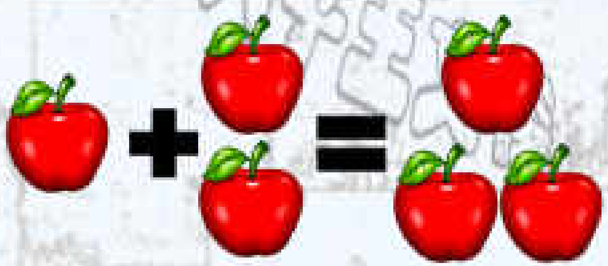
第一章　数制和码制

——王文俊

一、数码的基本概念

数字电路所处理的各种数字信号是以**数码**形式给出的。

数	码
表示**数量的大小**	表示**不同事物**或事物的**不同状态**



1 + 2 = 3

1 2 3

- 1、数制

在用数码表示数量大小时，常用进位计数制的方法组成多位数码。

一位数码（X）	→	两位数码（X X）
0～9（十个数）	进位计数制	0～99（一百个数）

每一位的构成方法

1 2 3 4 5
6 7 8 9 0

每一位都由0~9十个数构成

从低位向高位的进位规则

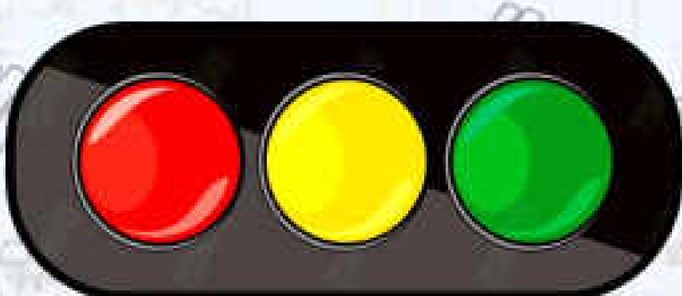
$$\begin{array}{r} 9 \\ +\ 1 \\ \hline 10 \end{array}$$

进位规则为：“逢十进一”

- 2、码制

在用数码表示不同事物时，数码是不同事物的代号，简称为代码。

编制代码所需要遵循的规则，称为码制。



信号灯编码	红灯	黄灯	绿灯
甲	0	1	2
乙	1	2	3
丙	4	6	9

每个人都可以根据自己的需要，制定编码规则，编制代码。

考虑信息交换的需要，必须制定一些共同使用的通用代码。

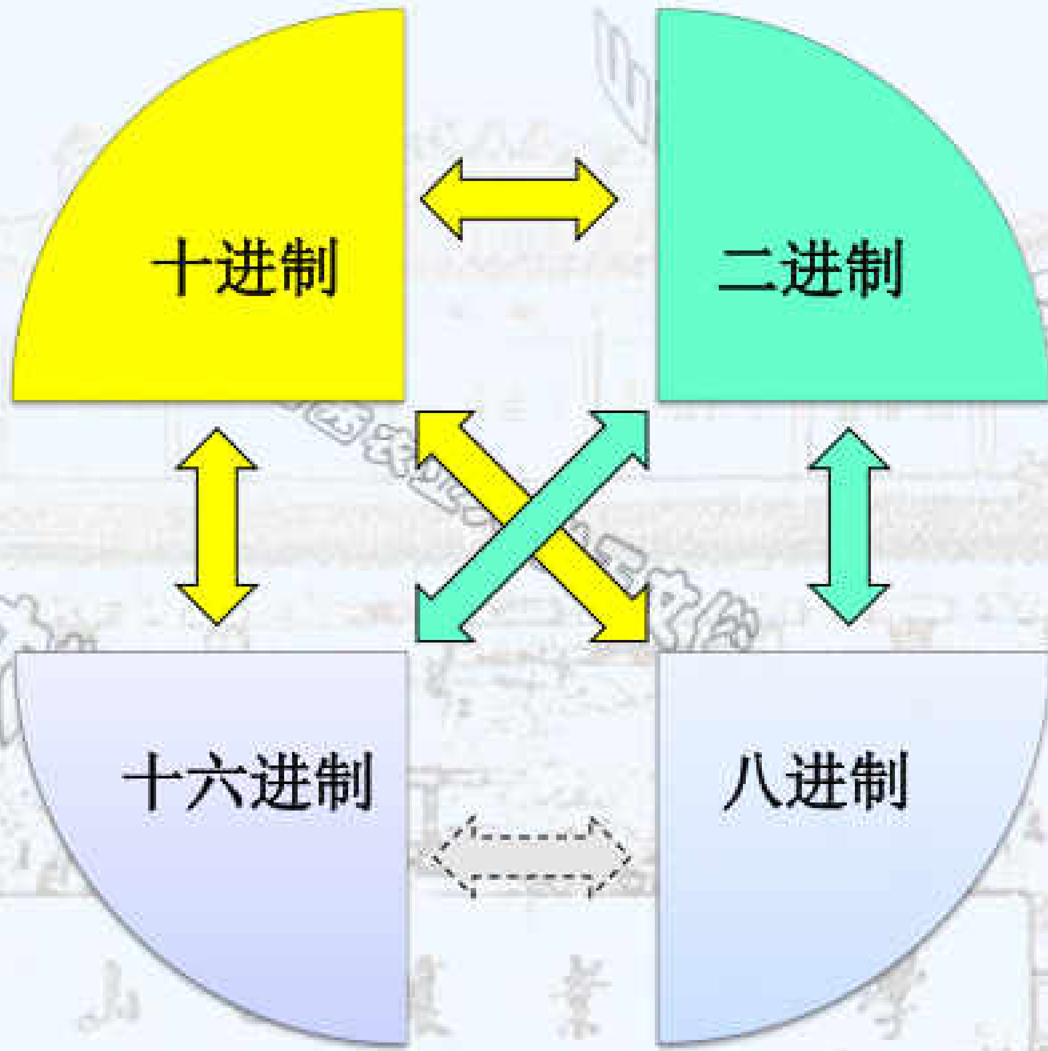
二、几种常用的数制

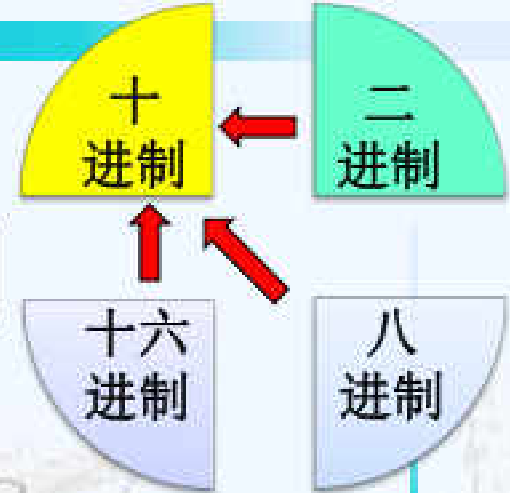
数制的内容	二进制	八进制	十进制	十六进制
每一位的表示方法	0~1 两个数	0~7 八个数	0~9 十个数	0~15 十六个数
进位规则	逢二进一	逢八进一	逢十进一	逢十六进一
进制的表示方式	B或2	O或8	D或10	H或16
示例	$(101.11)_2$	$(12.4)_8$	$(5.75)_{10}$	$(2A.7F)_{16}$

不同进制数的对照表

十进制数	二进制	八进制	十六进制
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

三、不同数制间的转换





- 1、任意进制→十进制（N—十转换）

采用通用**展开式**，实现任意进制向十进制的转换。

$$D=\sum k_i N^i$$

i为位数

…	4	3	2	1	0	.	-1	-2	-3	-4	…

N为基数

- 二进制 N=2；八进制 N=8；十进制 N=10；十六进制 N=16

k_i为第i位的系数，N^i为第i位的权

- 例1：（143.75）$_{10}$的展开式

位数i	2	1	0	.	-1	-2
权N^i	10^2	10^1	10^0	.	10^{-1}	10^{-2}
系数k_i	1	4	3	.	7	5
项$k_i N^i$	100	40	3	.	0.7	0.05
D	100 + 40 + 3 + 0.7 + 0.05 = 143.75					

$$D = \sum k_i 10^i$$

$$= 1\times10^2 + 4\times10^1 + 3\times10^0 + 7\times10^{-1} + 5\times10^{-2}$$

$$= (143.75)_{10}$$

- 例2（二-十转换）：将（101.11）$_2$ 转换为十进制数

位数i	2	1	0	.	-1	-2
权N^i	2^2	2^1	2^0	.	2^{-1}	2^{-2}
系数k_i	1	0	1	.	1	1
项$k_i N^i$	4	0	1	.	0.5	0.25
D	$4+0+1+0.5+0.25=5.75$					

$$D=\sum k_i 2^i$$
$$=1\times 2^2+0\times 2^1+1\times 2^0+1\times 2^{-1}+1\times 2^{-2}$$
$$=(5.75)_{10}$$

- 例3（八-十转换）：将（12.4）$_8$ 转换为十进制数

位数i	2	1	0	.	-1	-2
权N^i	8^2	8^1	8^0	.	8^{-1}	8^{-2}
系数k_i		1	2	.	4	
项$k_i N^i$		8	2	.	0.5	
D	$8+2+0.5=10.5$					

$$D=\sum k_i 8^i$$

$$=1\times 8^1+2\times 8^0+4\times 8^{-1}$$

$$=(10.5)_{10}$$

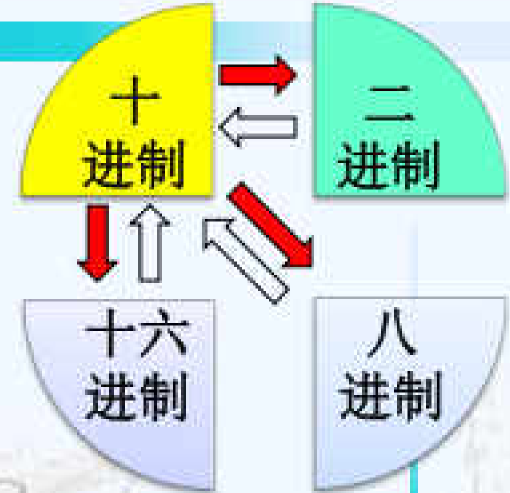
- 例4（十六-十转换）：将（2A.7F）$_{16}$ 转换为十进制数

位数i	2	1	0	.	-1	-2
权N^i	16^2	16^1	16^0	.	16^{-1}	16^{-2}
系数k_i		2	A(10)	.	7	F(15)
项$k_i N^i$		32	10	.	7/16	15/256
D	32 + 10 + 7/16 + 15/256 = 42.49609375					

$$D = \sum k_i 16^i$$

$$= 2\times16^1 + 10\times16^0 + 7\times16^{-1} + 15\times16^{-2}$$

$$= (42.49609375)_{10}$$



- 2、十进制→任意进制（十—N转换）

任意进制数

- 整数部分
  - 相除求余左向左
    - 除以基数求余数
    - 余数在小数点左侧依次向左排列
- 小数部分
  - 相乘取整右向右
    - 乘以基数取整数
    - 整数在小数点右侧依次向右排列

- 例5（十-二转换）：将（173.8125）$_{10}$ 转换为二进制数

1）整数部分（173）$_{10}$：相除求余左向左

2	173	余数=1
2	86	余数=0
2	43	余数=1
2	21	余数=1
2	10	余数=0
2	5	余数=1
2	2	余数=0
2	1	余数=1
	0	余数=0

…… ……

1	0	1	0	1	1	0	1	.

2）小数部分（0.8125）$_{10}$：相乘取整右向右

0.8125	× 2 =	1.6250	······· 整数部分=1
0.6250	× 2 =	1.2500	······· 整数部分=1
0.2500	× 2 =	0.500	······· 整数部分=0
0.500	× 2 =	1.000	······· 整数部分=1
0			

.	1	1	0	1

3）整数部分与小数部分合并

$$(173.8125)_{10} = (10101101.1101)_2$$

- 例6（十-八转换）：将（173.8125）$_{10}$ 转换为八进制数

8	173	----- 余数=5
8	21	----- 余数=5
8	2	----- 余数=2
	0	

→ 255.

0.8125	× 8 =	6.5	······· 整数=6
0.5	× 8 =	4.0	······· 整数=4
0			

→ .64

$$(173.8125)_{10} = (255.64)_8$$

- 例7（十-十六转换）：将（173.8125）$_{10}$ 转换为十六进制数

$$
\begin{array}{r|l}
16 & 173 \\ \hline
16 & 10 \\ \hline
 & 0
\end{array}
$$

- - - - - 余数=13

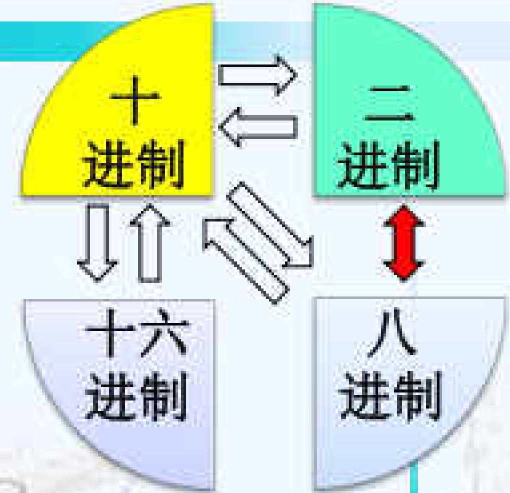
- - - - - 余数=10

→ AD.

$$
\begin{array}{l}
0.8125 \times 16 = 13 \\ \hline
0
\end{array}
$$

······· 整数=13 → .D

$$(173.8125)_{10} = (AD.D)_{16}$$



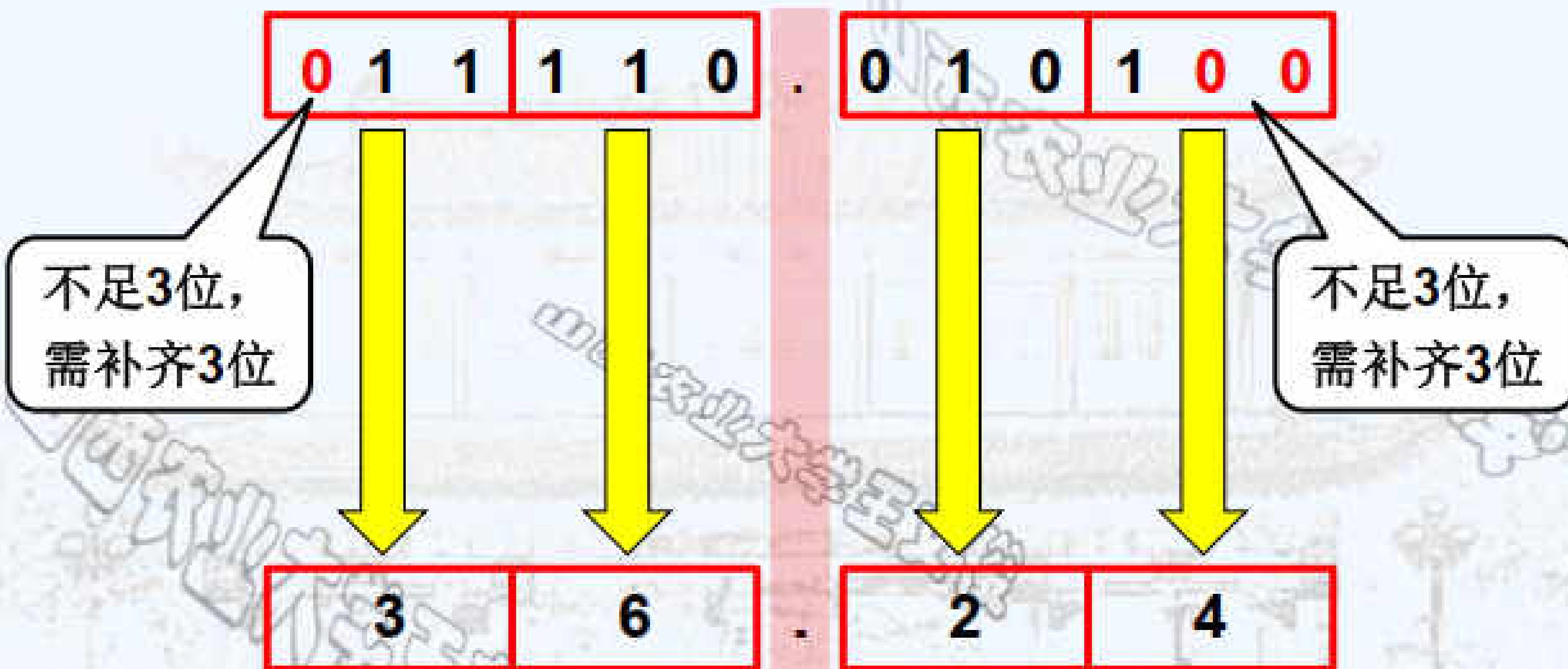
- **3、二进制与八进制的相互转换**

十进制	二进制	八进制
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	10

将3位二进制数看做一个整体，等价于1位八进制数。

二-八转换	• 每3位二进制数看做一个整体，进行整体替换
八-二转换	• 将每1位八进制数逐个转换为3位二进制数

- 例8（二-八转换）：将（11110.0101）$_2$转换为八进制数



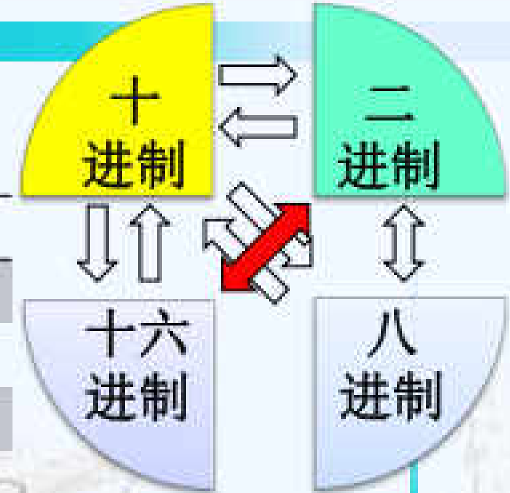
$$(11110.0101)_2 = (36.24)_8$$

- 例9（八-二转换）：将（52.43）$_8$转换为二进制数

5	2	.	4	3
1 0 1	0 1 0	.	1 0 0	0 1 1

$$(52.43)_8 = (101010.100011)_2$$

- 4、二进制与十六进制的相互转换



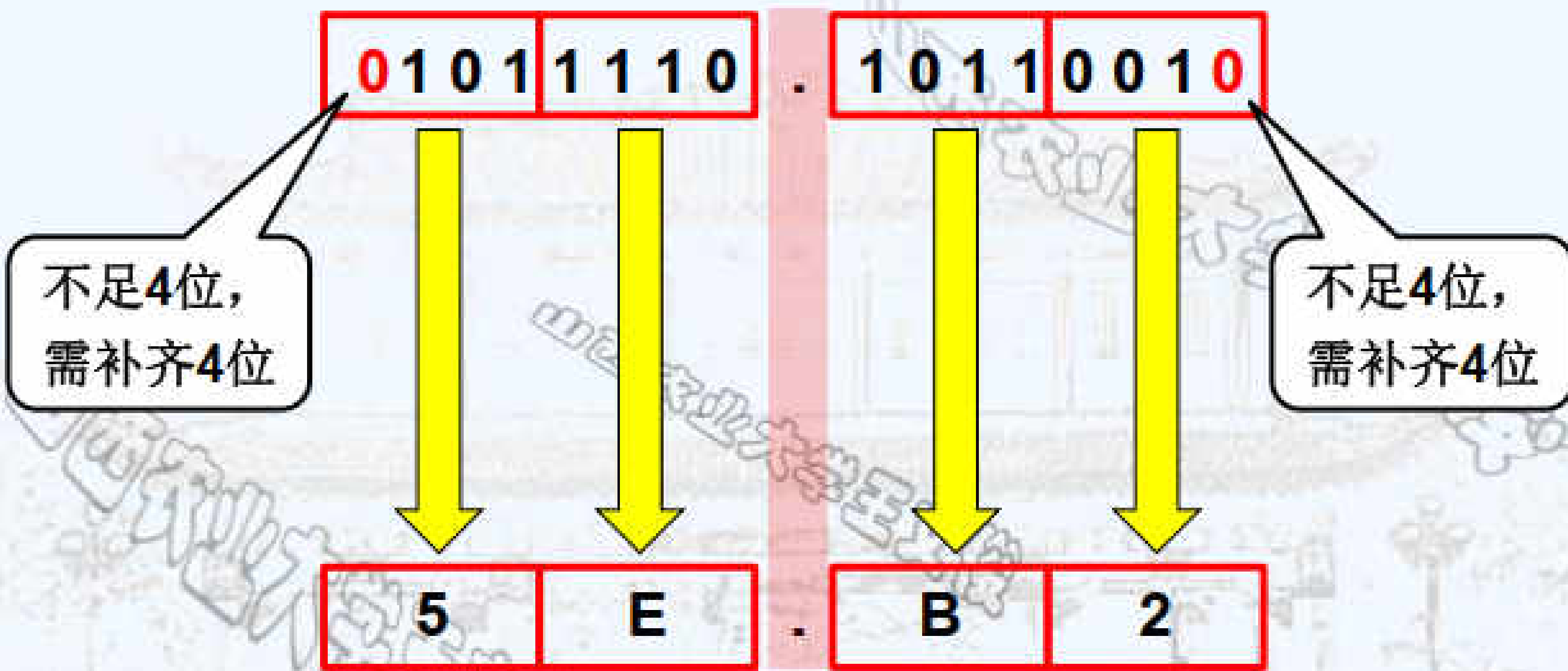
十进制	二进制	十六进制	十进制	二进制	十六进制
00	0000	0	08	1000	8
01	0001	1	09	1001	9
02	0010	2	10	1010	A
03	0011	3	11	1011	B
04	0100	4	12	1100	C
05	0101	5	13	1101	D
06	0110	6	14	1110	E
07	0111	7	15	1111	F
			16	10000	10

将4位二进制数看做一个整体，等价于1位十六进制数。

二-十六转换
- 每4位二进制数看做一个整体，进行整体替换

十六-二转换
- 将每1位十六进制数逐个转换为4位二进制数

- 例10（二-十六转换）：将（1011110.1011001）$_2$转换为十六进制数



$(1011110.1011001)_2 = (5E.B2)_{16}$

- 例11（十六-二转换）：将（8FA.C6）$_{16}$转换为二进制数

8	F	A	.	C	6
1 0 0 0	1 1 1 1	1 0 1 0	.	1 1 0 0	0 1 1 0

（8FA.C6）$_{16}$ =（100011111010.11000110）$_{2}$

四、二进制的算术运算

- 1、二进制的算术运算

加法运算

```
   1 0 0 1
 + 0 1 0 1
 ---------
   1 1 1 0
```

减法运算

```
   1 0 0 1
 - 0 1 0 1
 ---------
   0 1 0 0
```

乘法运算

```
          1 0 0 1
        × 0 1 0 1
   --------------
          1 0 0 1    ← 乘数为1，被乘数不移动
        0 0 0 0      ← 乘数为0，0000 左移 1 位
      1 0 0 1        ← 乘数为1，被乘数 左移 2 位
    0 0 0 0          ← 乘数为0，0000 左移 3 位
   --------------
    0 1 0 1 1 0 1      4数相加
```

除法运算

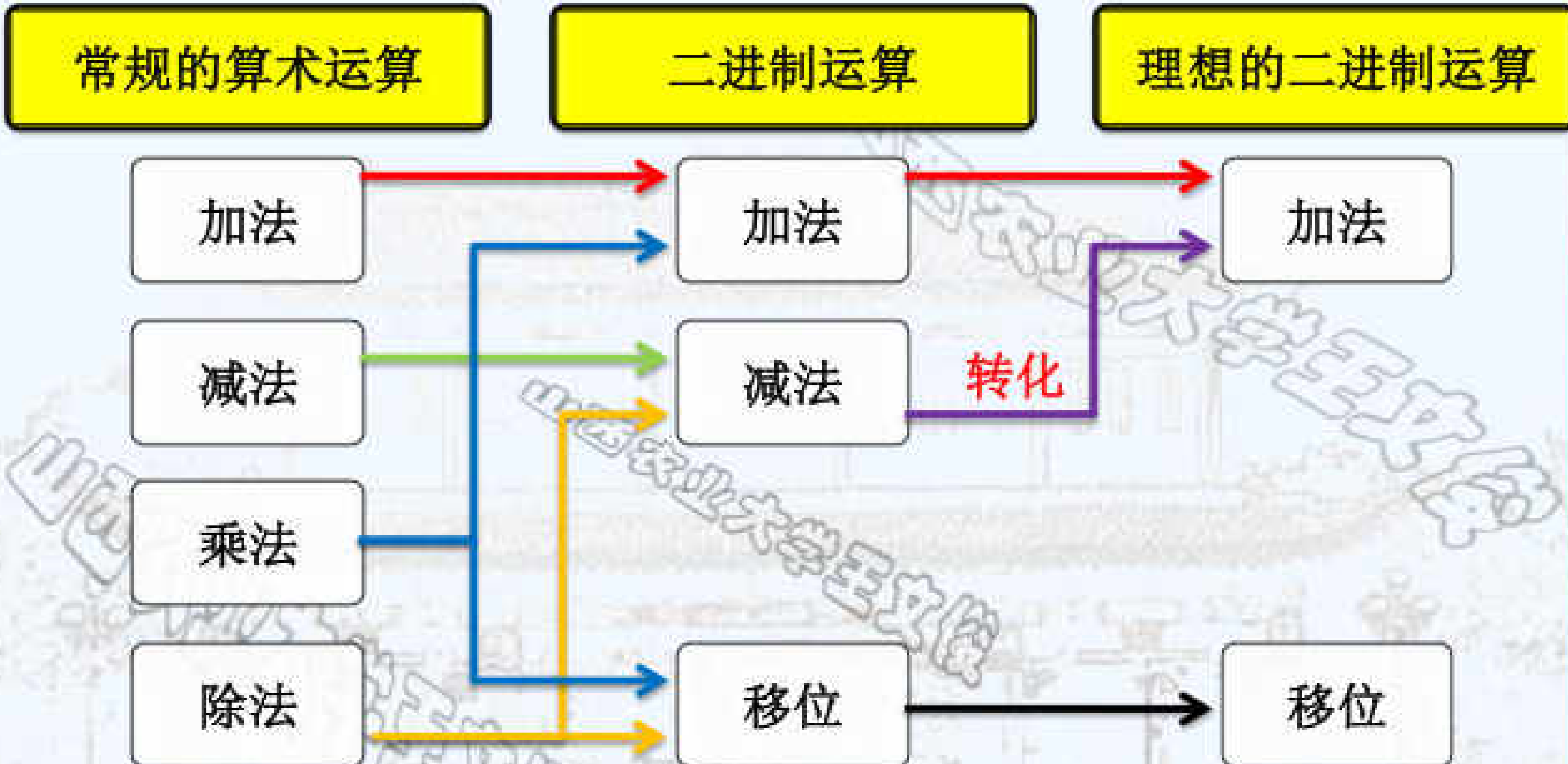
```
              1. 1  1···
        ─────────────────
0 1 0 1 ) 1 0 0 1
          0 1 0 1
        ─────────────────
            1 0 0 0
            0 1 0 1
        ─────────────────
              0 1 1 0
              0 1 0 1
        ─────────────────
              0 0 1 0
```

除数不移动，相减

除数右移 1 位，相减

除数右移 2 位，相减

- 2、二进制算术运算的特点



采用二进制，可以将加、减、乘、除简化为“相加”和“移位”，从而使运算电路的结构大为简化。

因此，数字电路普遍采用二进制算术运算。

- 3、减法的计算流程

- A - B
  - 比大小 定正负
    - A > B，正数
    - A < B，负数
  - 计算 绝对值
    - A > B，值 = A - B
    - A < B，值 = B - A

减法计算过程较复杂，且需要数值比较电路和减法运算电路。

五、原码、补码与反码

- 1、无符号数与有符号数

无符号数（n位）	数值位（n位）	
	十进制：$0\sim10^n-1$，共10^n个数	
	二进制：$0\sim2^n-1$，共2^n个数	
有符号数（n位）	符号位（1位）	数值位（n-1位）
原码	0：正数	十进制：$0\sim10^{n-1}-1$，共10^{n-1}个数（模数）
	1：负数	二进制：$0\sim2^{n-1}-1$，共2^{n-1}个数（模数）

- 例12：4位十进制数和4位二进制的表示范围

4位十进制数	无符号数	4位数值位	0000 ~ 9999 共10^4=10000个数
	有符号数	1位符号位 3位数值位	-999 ~ ±0 ~ +999 共1999个数
4位二进制数	无符号数	4位数值位	0000 ~ 1111 共2^4=16个数
	有符号数	1位符号位 3位数值位	-111 ~ ±0 ~ +111 共15个数

- 例13：写出+14和-14的二进制数的原码

$$(14)_{10} = (1110)_2$$

$$(+14)_{10} = (0\ 1110)_2$$

$$(-14)_{10} = (1\ 1110)_2$$

- 例14：写出$(1010)_2$表示的十进制数

$(1010)_2$

无符号数 1010 → $(10)_{10}$

有符号数 1 010 → $(-2)_{10}$

- **2、十进制的补码（仅讨论数值位）**

为了能够将减法运算转换为加法运算，提出了**补码**的概念。

1位十进制数（10个数）

6

6-4=2

6+6=(1)2

-4与+6结果相同
4 + 6 = 10
6为-4对模10的补码

6 - 4 ⇨ 6 + (-4) ⇨ 6 + (-4的补码) ⇨ 6 + 6

2位十进制数（100个数）

66

66-44=22

66+56=(1)22

-44与+56结果相同
44 + 56 = 100
56为-44对模100的补码

66 - 44 ⇨ 66 + (-44) ⇨ 66 + (-44的补码) ⇨ 66 + 56

- 3、二进制的补码（仅讨论数值位）

4位二进制数（2^4=16个数）

1011

1011-0111=0100

1011+1001=(1)0100

-0111与+1001结果相同
0111 + 1001 = 10000
1001为-0111对模16的补码

1011 - 0111 → 1011 + (-0111) → 1011 + (-0111的补码) → 1011 + 1001

n位二进制数（2^n个数）

M

M - B

M + C

$B + C = 2^n$
$C = 2^n + (-B)$
C为-B对模2^n的补码

- **4、补码运算**

对数值位（不包括符号位）为n位的二进制数M，共可表示2^n个数。
补码（M）$_{COMP}$的计算公式如下：

$$(M)_{COMP}=\begin{cases} M & M\geq 0 \\ 2^n-M & M<0 \end{cases}$$

正数

- 本身就是加法运算
- 不需要求补码

负数

- 减法运算（加负数）
- 通过求补码转化为加法

由于补码运算涉及到减法运算，所以一般不直接求补码。

- 5、反码运算

对数值位（不包括符号位）为n位的二进制数M，将M中所有的数值位的0改为1、1改为0，即可得到反码（M）$_{INV}$

1 0 1 0

0→1 1→0

0 1 0 1

$1010+0101=1111=2^4-1$

$$(M)_{INV}=\begin{cases} M & M\geq 0 \\ (2^n-1)-M & M<0 \end{cases}$$

二进制反码运算步骤简单，极易实现。

- 6、补码与反码的关系

$$(M)_{COMP}=\begin{cases} M & M\ge 0 \\ 2^n-M & M<0 \end{cases} \qquad (M)_{INV}=\begin{cases} M & M\ge 0 \\ (2^n-1)-M & M<0 \end{cases}$$

$$(M)_{COMP}=\begin{cases} M & M\ge 0 \\ (M)_{INV}+1 & M<0 \end{cases}$$

整个数值位的最后一位**+1**，
而不是整数位的最后一位**+1**

通过反码运算，间接获得补码，可避开直接求补码中的减法运算。

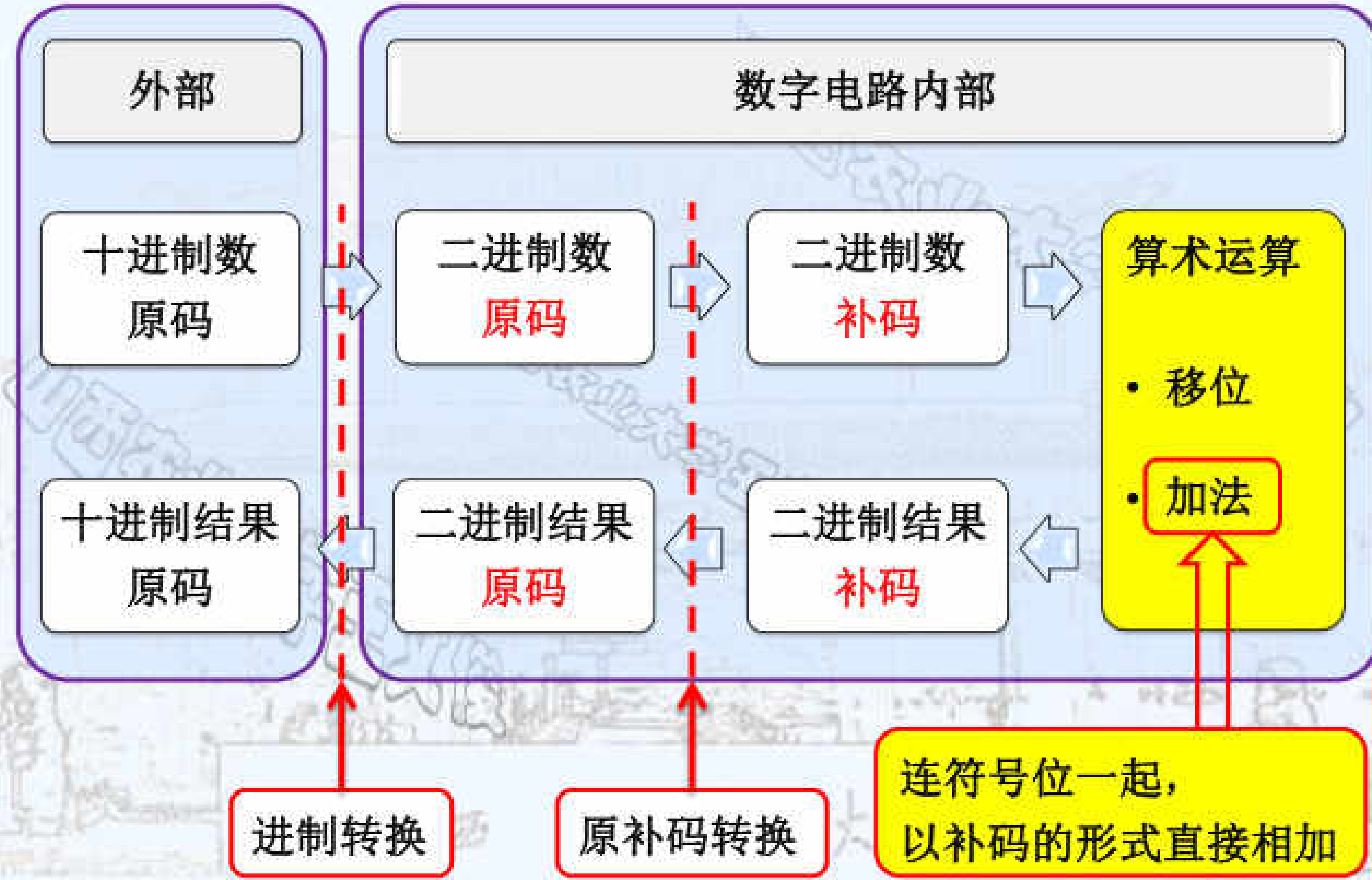
- 7、原码、反码、补码运算总结

原码（n+1位）	符号位（1位）	数值位（n位，模数为2^n）
原码	0	数值为M
正数	原码 = 反码 = 补码	
原码	1	数值为M
	保持↓不变	↓0→1，1→0
反码	1	数值为 $(2^n - 1) - M$
	保持↓不变	↓最后一位+1
补码	1	数值为 $2^n - M$

- 例15：写出带符号位二进制数001010（+10）、101010（-10）、0101.11（+5.75）和1101.11（-5.75）的反码和补码。

十进制数	+10	-10	+5.75	-5.75
原码	0 01010	1 01010	0 101.11	1 101.11
反码	0 01010	1 10101	0 101.11	1 010.00
补码	0 01010	1 10110	0 101.11	1 010.01

- 8、补码运算的应用


外部
数字电路内部
十进制数
原码
二进制数
原码
二进制数
补码
算术运算
• 移位
• 加法
十进制结果
原码
二进制结果
原码
二进制结果
补码
进制转换
原补码转换
连符号位一起，
以补码的形式直接相加

- 例16：用二进制补码运算求：
- 13＋10 、13－10 、－13＋10 、－13－10

十进制数	原码	补码
+23	0 10111	0 10111
-23	1 10111	1 01001
+13	0 01101	0 01101
-13	1 01101	1 10011
+10	0 01010	0 01010
-10	1 01010	1 10110
+3	0 00011	0 00011
-3	1 00011	1 11101

13+10	(+13) + (+10)
+13补码	0 01101
+10补码	0 01010
结果补码	0 10111
结果原码	0 10111
结果	$(+23)_{10}$

- 例16：用二进制补码运算求：
- 13＋10 、13－10 、－13＋10 、－13－10

十进制数	原码	补码
+23	0 10111	0 10111
-23	1 10111	1 01001
+13	0 01101	0 01101
-13	1 01101	1 10011
+10	0 01010	0 01010
-10	1 01010	1 10110
+3	0 00011	0 00011
-3	1 00011	1 11101

13-10 → (+13) + (-10)

+13补码	0 01101
-10补码	1 10110
结果补码	(1) 0 00011
结果原码	0 00011
结果	$(+3)_{10}$

- 例16：用二进制补码运算求：
- 13＋10 、13－10 、－13＋10 、－13－10

十进制数	原码	补码
+23	0 10111	0 10111
-23	1 10111	1 01001
+13	0 01101	0 01101
-13	1 01101	1 10011
+10	0 01010	0 01010
-10	1 01010	1 10110
+3	0 00011	0 00011
-3	1 00011	1 11101

-13+10	(-13) + (+10)
-13补码	1 10011
+10补码	0 01010
结果补码	1 11101
结果原码	1 00011
结果	$(-3)_{10}$

- 例16：用二进制补码运算求：
- 13＋10 、13－10 、－13＋10 、－13－10

十进制数	原码	补码
+23	0 10111	0 10111
-23	1 10111	1 01001
+13	0 01101	0 01101
-13	1 01101	1 10011
+10	0 01010	0 01010
-10	1 01010	1 10110
+3	0 00011	0 00011
-3	1 00011	1 11101

-13-10	(-13) + (-10)
-13补码	1 10011
-10补码	1 10110
结果补码	(1) 1 01001
结果原码	1 10111
结果	（-23）$_{10}$

六、几种常用的编码

- 1、十进制代码

4位二进制数可表示16个数

- 10个用于表示十进制数0~9
- 6个废弃不用

不同编码规则

- BCD码（8421码）
- 2421码
- 5211码
- 余3码
- 余3循环码

十进制数	8421码 (BCD)	2421码	5211码
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0100
3	0011	0011	0101
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1100
8	1000	1110	1101
9	1001	1111	1111

恒权码

- 编码中每一位的权恒定不变。
- 8421码的权：8、4、2、1
- 2421码的权：2、4、2、1
- 5211码的权：5、2、1、1

编码方式

- 将代码中每一位数与权重相乘，并累积求和，即可获得对应的十进制数。
- 2421码的“1101”：

$$\begin{array}{r} 2 \quad 4 \quad 2 \quad 1 \\ \times \; 1 \quad 1 \quad 0 \quad 1 \\ \hline 2+4+0+1 = 7 \end{array}$$

十进制数	余3码	余3循环码
0	0011	0010
1	0100	0110
2	0101	0111
3	0110	0101
4	0111	0100
5	1000	1100
6	1001	1101
7	1010	1111
8	1011	1110
9	1100	1010

余3码

- 将每一个余3码看做一个4位二进制数；
- 它的数值比其所表示的十进制数码多3。
- 余3码的“1010”：

$$(1010)_2=(10)_{10}$$

10 - 7 = 3（余3）

余3循环码

- 相邻两个代码之间仅有一位的状态发生变化。

- 2、格雷码

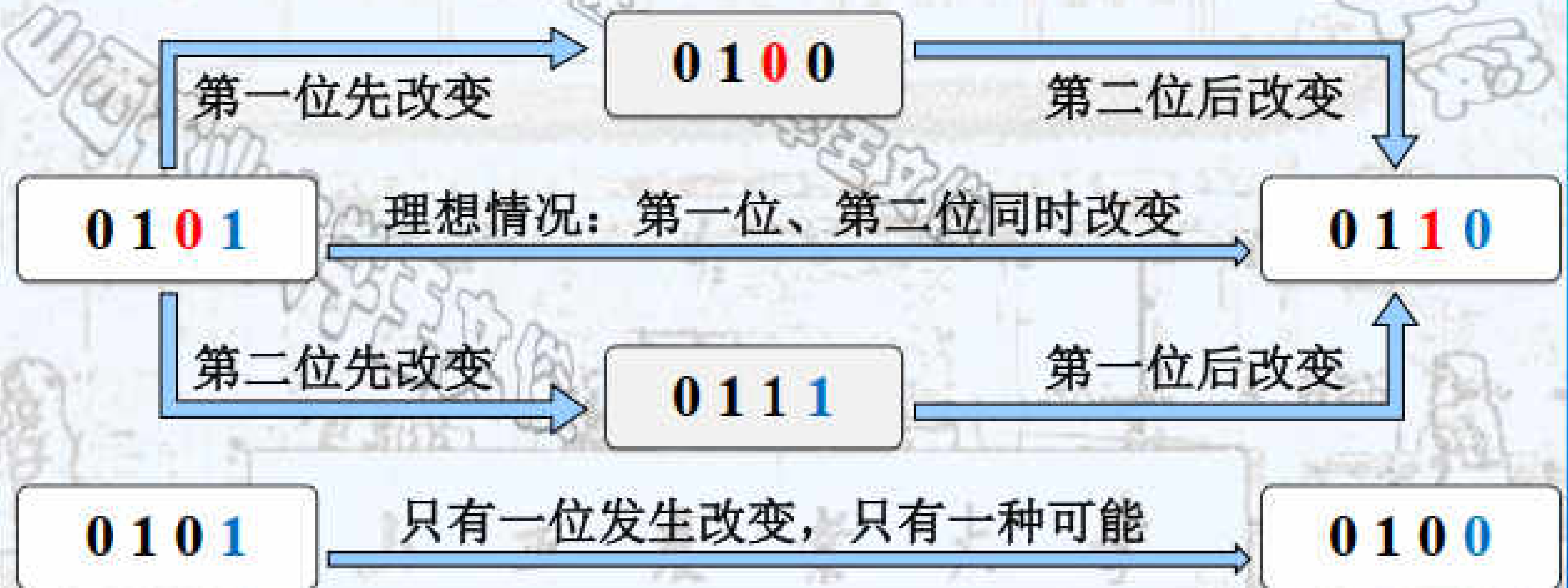
编码顺序	二进制码	格雷码	编码顺序	二进制码	格雷码
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

格雷码每一位的状态都按一定顺序循环，因此也被称为循环码。

格雷码相邻两个代码之间只有一位发生变化，在代码转换过程中不会出现过渡“噪声”。

- 过渡“噪声”的产生

编码顺序	二进制码	格雷码	编码顺序	二进制码	格雷码
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000



- 3、美国信息交换标准代码（ASCII）

ASCII码是由美国国家标准化协会制定的一种信息代码。

ASCII是一组7位二进制代码($b_7b_6b_5b_4b_3b_2b_1$)，共128个。

数字0-9（10个）

大、小写英文字母（52个）

各种符号代码（32个）

控制码（34个）

ASCII码是一种国际通用标准代码，广泛应用于计算机和通信领域。

ASCII表

（American Standard Code for Information Interchange 美国标准信息交换代码）

高四位		ASCII控制字符												ASCII打印字符												
		0000						0001						0010		0011		0100		0101		0100		0111		
		0						1						2		3		4		5		6		7		
低四位		十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl
0000	0	0		^@	NUL	\0	空字符	16	►	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH		标题开始	17	◄	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX		正文开始	18	↕	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX		正文结束	19	‼	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	♦	^D	EOT		传输结束	20	¶	^T	DC4		设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ		查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK		肯定应答	22	▬	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	•	^G	BEL	\a	响铃	23	↨	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	\b	退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM		介质结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	\n	换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	11	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	\f	换页	28	∟	^\	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|	
1101	D	13	♪	^M	CR	\r	回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	14	♫	^N	SO		移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	F	15	☼	^O	SI		移入	31	▼	^-	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	⌂	*Backspace 代码：DEL

注：表中的ASCII字符可以用"Alt + 小键盘上的数字键"方法输入。

习 题

- P18【题1.4】（1）（4）
- P18【题1.9】（1）
- P19【题1.11】（1）（3）
- P19【题1.15】（1）（5）