Dalian University of Technology
School of Microelectronics

数字电路与系统

Digital Circuits and Systems

课程性质

“数字电路与系统”——工科电类专业的专业基础课

课程任务：从应用角度出发

- 数字电路的常用集成器件原理、符号、功能；
- 由常用器件组成的组合电路、时序电路的分析和设计方法；
- 进而分析和设计由中规模乃至大规模集成电路组成的数字系统。

- 后续数字集成电路设计、单片机/嵌入式系统设计等课程的基础

课程要求

- 2020年秋季学期， 40学时理论授课
- 2020年春季学期，24学时的数字电路与系统实验
- 先修课程： 《电路理论》、《模拟电子线路》

考核方式

平时成绩：30分 + 期末考试：70分

- 出勤、作业、课堂提问/测验等

数字电路与系统

- Chapter 1 Fundamentals of Digital Logic 数字逻辑基础
- Chapter 2 Logic Algebra 逻辑代数基础
- Chapter 3 Logic Gates 逻辑门电路

- Chapter 4 Combinational Logic 组合逻辑电路
- Chapter 5 Flip – Flop 触发器
- Chapter 6 Sequential Logic 时序逻辑电路
- Chapter 7 Pulse Circuits 脉冲波形的产生与变换

- Chapter 8 Digital Analog Conversions 模数与数模转换
- Chapter 9 Memory and Programmable Logic Devices 半导体存储器及可编程逻辑器件
- Chapter10 Digital System Design 数字系统设计基础

教材及参考书

使用教材

- 戚金清、王兢 主编. 数字电路与系统(第3版). 电子工业出版社,2016

主要参考书

- 阎石 主编. 数字电子设计基础（第6版）. 高等教育出版社,2016
- 邓元庆、贾鹏等 编. 数字电路与系统设计(第3版). 西安电子科技大学出版社,2016

第 1 章　数字逻辑基础

Fundamentals of Digital Logic

§1.1 数字电路 Digital Logic Circuits

§1.2 数制 Number Systems

§1.3 数制间转换 Base Conversions

§1.4 代码 Codes

§1.5 带符号的二进制数 Signed Binary Numbers

§1.1 数字电路 Digital Logic Circuits

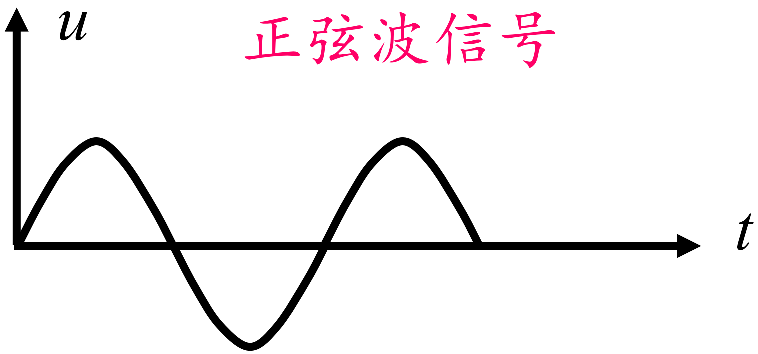
自然界的物理量，按其变化规律可分为两类：

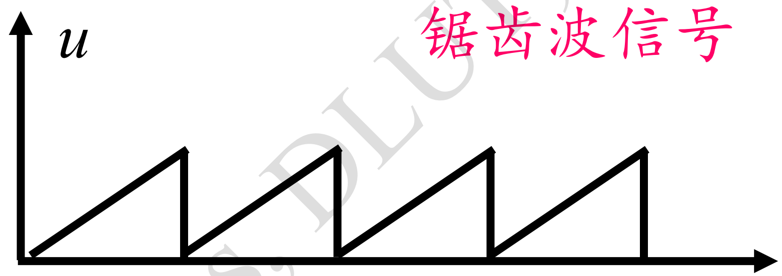
- 模拟量 Analog：数值和时间都可以连续取值
- 数字量 Digital ：时间上离散，值域内只能取某些特定值

Analog
模拟量
- 声音
- 压力
- 速度
- 气味
- 温度
- 电压值
- 流量
- …

Digital
数字量
- 人数
- 模拟量的数字形式
- 语言和文字
- 编码
- …

模拟信号

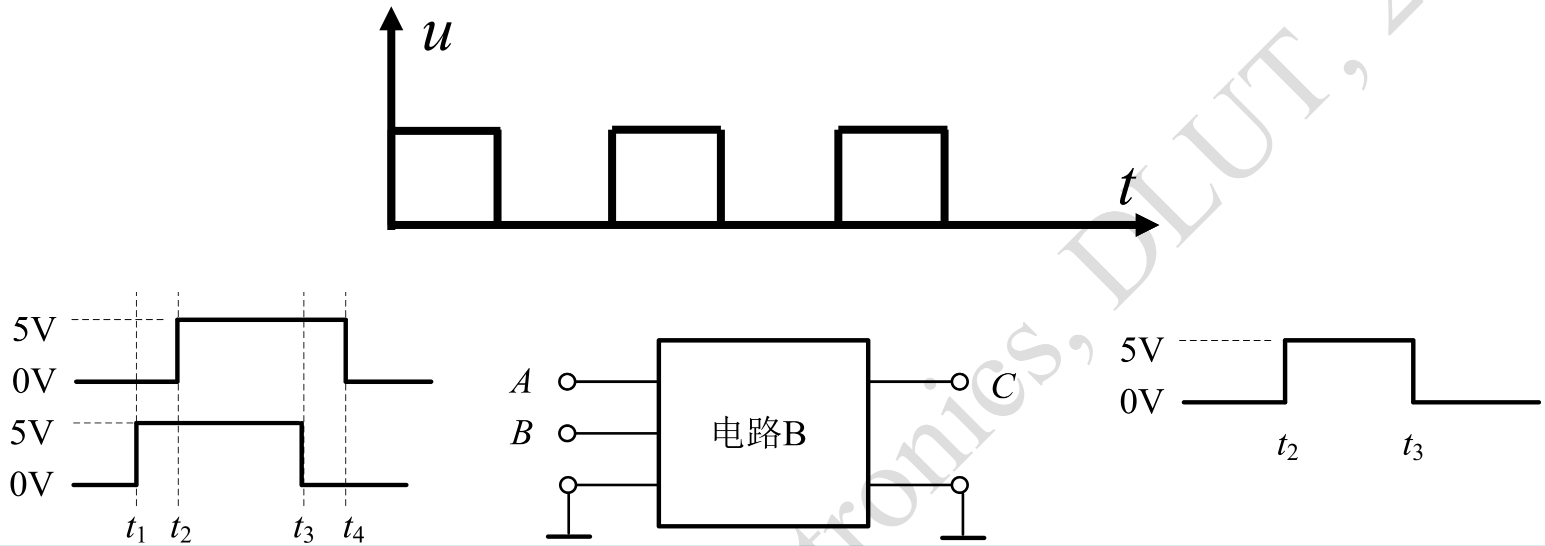




- 研究模拟信号时，我们**注重电路输入、输出信号间的大小、相位关系**。相应的电子电路就是模拟电路，包括交直流放大器、滤波器、信号发生器等。

- 在模拟电路中，**晶体管一般工作在放大状态**。

数字信号

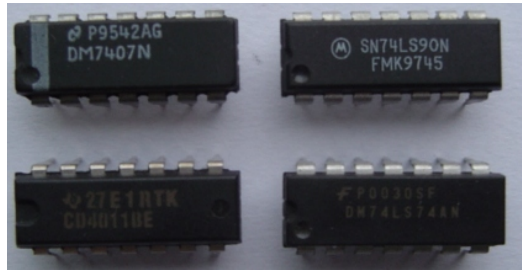


- 研究数字电路时**注重电路输出、输入间的逻辑关系**，因此不能采用模拟电路的分析方法。主要的分析工具是逻辑代数，电路的功能用真值表、逻辑表达式或波形图表示。

- 在数字电路中，三极管工作在开关状态下，即**工作在饱和状态或截止状态**。

数字电路的发展：电子管、晶体管、集成电路等阶段

单门集成电路



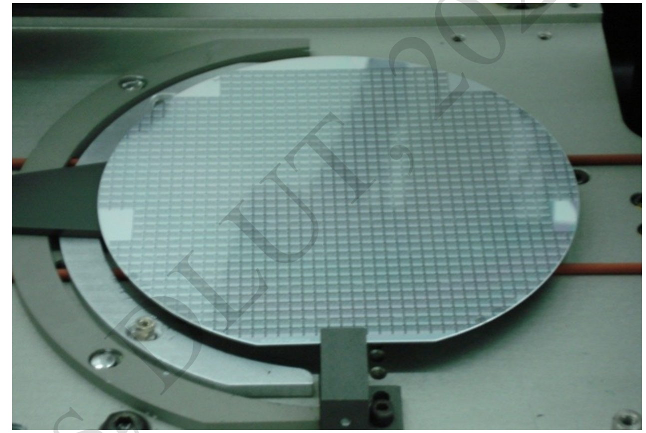
SSI/MSI

LSI/VLSI

分类	逻辑门个数	典型集成电路
小规模 SSI	≤10	基本门、触发器
中规模 MSI	10~100	译码器、计数器、加法器
大规模 LSI	100~10000	小容量存储器、门阵列
超大规模 VLSI	≥10000	单片微处理器、高密度可编程逻辑器件
特大规模 ULSI	10^7~10^9个元件	16M FLASH、256M DRAM
巨大规模 GLSI	≥10^9个元件	1G DRAM

特征尺寸：半导体器件中的最小尺寸。CMOS工艺中，特征尺寸典型代表为“栅”的宽度，即MOS器件的沟道长度。特征尺寸越小，芯片的集成度越高，性能越好，功耗越低。

集成电路制造车间



集成电路的基础材料——晶圆（Wafer）

3种常用的数字集成电路

- **标准集成电路**：功能、物理配置固定
- **可编程逻辑器件**：根据用户需求实现相应的逻辑功能，并且可以多次编程，如CPLD和FPGA
- **专用集成电路（ASIC）**：针对整机或系统的需要，专门为之设计制造的集成电路

数字系统的优势

（1）稳定性高，可靠性好

- 给定相同的输入信号（值和时间序列），一个设计完好的数字电路的输出总是相同的。
- 模拟电路的输出随外界温度、电源电压、器件的老化等因素而发生变化。
- 数字信号对噪声不敏感，抗干扰能力强，保密性好，信息的保存与传输更加简便可靠。

数字系统的优势

（2）易于设计

- 数字电路又称为数字逻辑电路，它主要是对用0和1表示的数字信号进行逻辑运算和处理，广泛使用的数学工具是逻辑代数。
- 不需要复杂的数学知识，不像对电容器、晶体管或其他模拟器件那样，要求对模型进行计算才能理解和认识它们的内部特性和工作过程。

- 数字电路能够可靠地区分0和1两种状态就可以正常工作，电路的精度要求不高。因此，数字电路的分析与设计相对较容易。

数字系统的优势

（3）表征数学量精度高、范围大

Analog system

模拟系统的范围和精度受其线性区域的范围及噪声抑制能力的限制

Digital system

数字系统可以通过增加信息表示的位数来改善范围和精度

数字系统的优势

（4） 可编程性

现代数字系统的设计，大多采用可编程逻辑器件。采用硬件描述语言（VHDL/Verilog HDL）在计算机上完成电路设计的编译、仿真及综合，并写入芯片，方便灵活。

数字系统的优势

(5) 快速, 低功耗

- 集成电路中单管的开关速度可以做到小于10^{-11} s。整体器件中，信号从检测输入到输出的传输时间小于2×10^{-9} s。意味着器件每秒产生 5 亿个结果。

- 百万门以上超大规模集成芯片的功耗，可以达到毫瓦级 。

数字系统的优势

（6） 批量生产，低成本

数字电路:
- 结构简单
- 容易制造
- 通用性强

适合于电路集成

成本低廉

台式计算机常备有“扩展插槽”，以便将来使用更快的处理器或更大容量的存储器。

§1.2 数制 Number Systems

在计算机和数字系统中经常会遇到数制与编码。在数字系统中经常使用二进制、八进制和十六进制。

基数（radix or base）：一个数制所包含的数的个数

数制系统
- 十进制 decimal (r =10)
- 二进制 binary (r =2)
- 八进制 octal (r =8)
- 十六进制 hexadecimal (r =16)

1. 十进制 Decimal

十进制包含10个数字： 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

基数为10，逢十进一

一个十进制的数可以写成 多项式 的形式:

$$(194.32)_{10} = 1\times10^{2}+9\times10^{1}+4\times10^{0}+3\times10^{-1}+2\times10^{-2}$$

注意：位于不同位置的数大小不同

权：表示该位置的大小 weight

每个位置的权为基数10 的幂。

$$(194.32)_{10} = 1\times10^{2}+9\times10^{1}+4\times10^{0}+3\times10^{-1}+2\times10^{-2}$$

一般说，任何一个基数为 r 的数 N 都可以按权展开成多项式的形式:

$$N=\sum_{i=-m}^{n-1} a_i r^i$$

n – 整数个数

m – 小数个数

a_i -- 第 i 个数的系数

r^i -- 第 i 个数的位权

2. 二进制 Binary

二进制系统有2个数：0, 1

基数为 2，逢二进一

0~17 列在表 1:

表 1

Decimal	Binary	Decimal	Binary
0	0	10	1010
1	1	11	1011
2	10	12	1100
3	11	13	1101
4	100	14	1110
5	101	15	1111
6	110	16	10000
7	111	17	10001
8	1000		
9	1001		

$(11010.11)_2$ 可以写成:

$$1\times 2^4+1\times 2^3+0\times 2^2+1\times 2^1+0\times 2^0+1\times 2^{-1}+1\times 2^{-2}$$

$$=16+8+2+0.5+0.25$$

$$=26.75$$

转化成十进制数

从表 1 寻找规律:

表 1

Decimal	Binary		Decimal	Binary	
0	0		10	1010	
1	1		11	1011	
2	10	2^1	12	1100	
3	11		13	1101	
4	100	2^2	14	1110	
5	101		15	1111	
6	110		16	10000	2^4
7	111		17	10001	
8	1000	2^3			
9	1001				

1 2^1 2^2 2^3 …… 2^n

10 100 1000 $1\underbrace{0\cdots0}_{n\ \text{zeros}}$

$$(128)_{10} = (2^7)_{10} = (1\underbrace{0000000}_{7\text{个}0})_2$$

8 位数中最小的数

$(2^n)_{10} = (1\underbrace{0\cdots0}_{n\text{个}0})_2$ 是 $(n+1)$ 位数中最小的数

表 1

Decimal	Binary		Decimal	Binary	
0	0		10	1010	
1	1	2^1-1	11	1011	
2	10	2^1	12	1100	
3	11	2^2-1	13	1101	
4	100	2^2	14	1110	
5	101		15	1111	2^4-1
6	110		16	10000	2^4
7	111	2^3-1	17	10001	
8	1000	2^3			
9	1001				

② $2^1-1 \quad 2^2-1 \quad 2^3-1 \quad 2^4-1 \quad \dots$

$1 \quad 11 \quad 111 \quad 1111 \quad \dots$

$$(2^n-1)_{10} = (\underbrace{11\dots1}_{n \text{ ones}})_2$$

是 *n* 位数中最大的数

例: $(255)_{10} = (2^8-1)_{10} = (\underbrace{11111111}_{8\text{个}1})_2$

$$(253)_{10} = (255-2)_{10} = (11111111-10)_2 = (11111101)_2$$

为什么二进制广泛应用于数字系统中？

二进制优点：

1)容易表示

用电路的两个状态 - 开关来表示二进制数，数码的存储和传输简单、可靠。

2)分辨性好，抗干扰能力强

二进制的缺点： 数字较大时，位数过多

65:

十进制表示为 2 位：65

二进制表示为 7 位：1000001

数字越大，该缺点越明显

所以有些时候经常会用到八进制或十六进制

3. 八进制 Octal

八进制包括8个数: 0,1, 2, 3, 4, 5, 6, 7

基数为 8

$$(326.47)_8 = 3\times8^2+2\times8^1+6\times8^0+4\times8^{-1}+7\times8^{-2}$$

$$=192+16+6+0.5+0.12$$

$$=(214.62)_{10}$$

转化成十进制数

表 1

Decimal	Binary	Octal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	10
9	1001	11
10	1010	12
11	1011	13
12	1100	14
13	1101	15
14	1110	16
15	1111	17
16	10000	20
17	10001	21

4.十六进制 Hexadecimal

十六进制有16个数，表示为：

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

基数为 16

$$(3CE.4B)_{16} = 3\times16^{2}+12\times16^{1}+14\times16^{0}+4\times16^{-1}+11\times16^{-2}$$

$$= 768 \ +192 \ \ +14 \ +0.25 \ +0.043$$

$$= (974.293)_{10}$$

转化成十进制数

表1

Decimal	Binary	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	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
16	10000	20	10
17	10001	21	11

5. 任意进制 （γ 进制）

γ 进制包括 γ 个数： 0,1… γ-1

§1.3 数制间转换 Base Conversions

1. γ 进制转换成十进制:

将 γ 进制的数按权展开，实现 γ 进制转换成十进制

$$(111001.01)_2 = (1\times 2^5 + 1\times 2^4 + 1\times 2^3 + 1\times 2^0 + 1\times 2^{-2})_{10} = (57.25)_{10}$$

2. 十进制转换成 γ 进制:

1) 整数部分，除以 γ 取余，直到商为0为止，逆序

2) 小数部分，乘以 γ 取整，顺序

十进制转成二进制： 将 $(39.2)_{10}$ 转换成二进制数

整数部分，除以 γ 取余，直到商为0为止，逆序

整数：

2	39	remainder	1
2	19	………	1
2	9	………	1
2	4	………	0
2	2	………	0
2	1	………	1
	0		

LSB (least significant bit)

最低有效位

逆 序

MSB

(maximum significant bit)

最高有效位

$$(39)_{10} \rightarrow (100111)_2$$

小数：　小数部分，乘以 γ 取整，顺序　　$(39.2)_{10}$

	整数	
MSB		0.2
		x 2
	0 ←	0.4
		x 2
	0 …………………	0.8
		x 2
	1 …………………	1.6
		0.6
		x 2
	1 …………………	1.2

顺 序

保留位数越多，越逼近要转换的数，越准确

$$(0.2)_{10} \rightarrow (.0011)_2$$

$$(39.2)_{10} = (100111.0011)_2$$

$$= (39.1875)_{10}$$

十进制转换成八进制:

将 $(179.46)_{10}$ 转换成八进制数

小数点

8	179	….. 3
8	22	….. 6
8	2	….. 2
	0	

$$
\begin{array}{r}
0.46 \\
\times 8 \\
\hline
3.68
\end{array} \quad \rightarrow 3
$$

$$
\begin{array}{r}
0.68 \\
\times 8 \\
\hline
5.44
\end{array} \quad \rightarrow 5
$$

$$(179.46)_{10} = (263.35)_8$$

十进制转换成十六进制:

将 $(178.46)_{10}$ 转换成 十六进制数

$$
\begin{array}{r|l}
16 & 178 \quad \ldots\ldots 2 \\
\hline
16 & 11 \quad \ldots\ldots \mathrm{B} \\
\hline
 & 0
\end{array}
$$

$\bullet$

$$
\begin{array}{r}
0.46 \\
\times 16 \\
\hline
276 \\
46 \\
\hline
7 \quad \ldots\ldots \quad 7.36
\end{array}
$$

$$(178.46)_{10} = (\mathrm{B}2.7)_{16}$$

3. 二进制与八进制之间的转换

$8 = 2^3$ **一位八进制数可以用3位二进制数表示**

方法: 以小数点为界向两侧划分，三位一组，不够添0

$(1\ 101\ 110\ 011.\ 101\ 1\ )_2 = (1563.54)_8$

1 5 6 3 5 **4**

注意：最后 1: 100---4
第一个 1: 001---1

$(253.16)_8 = (010101\ 011 \cdot 001\ 110)_2$

两端的0可以略去

4. 二进制与十六进制之间的转换

$16 = 2^4$ **一位十六进制数可以用4位二进制数表示**

方法：以小数点为界向两侧划分，四位一组，不够添0

$$(1\ 0101\ 1110\ 1101.1011\ 1010)_2 = (15ED.BA)_{16}$$

$$\underline{1}\quad \underline{0101}\quad \underline{1110}\quad \underline{1101}.\underline{1011}\quad \underline{101}$$

1 5 E D B A

$$(3D5E.7A8)_{16} = (11\ 1101\ 0101\ \ 1110.\ 0111\ \ 1010\ \ 1)_2$$

§1.4 代码 Codes

代表信息的数码称为代码（code)。常用在计算机和数字系统中处理、存储以及传输各种信息。

1.4.1 8421 BCD 码

BCD: binary coded decimal（二进制编码的十进制)

BCD 码是有权码

BCD码用4位二进制数表示1位十进制数。

8421BCD 是应用最广泛的一种BCD码，因为其位权与二进制数位权相同。

表 1.

Decimal	Binary	Octal	Hexadecimal	8421BCD
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	0001 0000
11	1011	13	B	0001 0001
12	1100	14	C	0001 0010
13	1101	15	D	0001 0011
14	1110	16	E	0001 0100
15	1111	17	F	0001 0101
16	10000	20	10	0001 0110
17	10001	21	11	0001 0111

在 8421BCD 中 1010～1111 为禁用码

练习:

$(75.68)_{10}$=(0111 0101 . 0110 1000 $)_{8421BCD}$

注意: 两端的0不能省略！

$(0111\ 0010\ 0110\ 1001.\ 1000\ 0011)_{8421BCD}$

= (7269.83 $)_{10}$

- 十进制与8421BCD 之间可以直接转换
- 二进制与 BCD 码不能直接转换，要先转成十进制

decimal

binary 8421*BCD*

$(0001\ 1000)_{8421BCD} = (\ 18\ )_{10} = (\ 10010\ )_2$

$(00011000)_2 = (\ 24\ )_{10}$

BCD 码还包括 2421BCD, 4221BCD, 5421BCD等这些BCD码都是有权码

脚标 8421BCD 必须写

$(1001\ 0101\ 0010.0111\ 0110)_{8421BCD}$

1.4.2 格雷码 （The Gray Code ）

格雷码的最重要的特征：

任意两个相邻码之间只有一位不同

格雷码是一种无权码

Decimal	Binary	Gray code	Decimal	Binary	Gray code
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

在典型的 n 位格雷码中，0 和最大数 $(2^n - 1)$ 之间也只有一位不同，所以它是一种循环码。格雷码的这个特点使它在传输过程中引起的误差较小

例如，二进制 7：0111

8：1000

在 7 和 8 的边界上，二进制的四位数都发生变化，都处于模糊状态

Gray 码 7：0100

8：1100

在二者边界上仅存在一位发生变化，带来的误差不会大于1（即7和8之差）

例：有一叉车数控调速系统，分为10档速度，这10档速度分别用BCD码和格雷码表示如下：

速度	BCD码	格雷码	速度	BCD码	格雷码
0	0000	0000	5	0101	0111
1	0001	0001	6	0110	1111
2	0010	0011	7	0111	1110
3	0011	0010	8	1000	1100
4	0100	0110	9	1001	1000

现将3档速度调到4档速度。如果速度用BCD码编码，即：0011→0100。

如果由0→1比由1→0快，在转换过程种将会短暂出现0111（七档），从而出现振动。

0011 → 0111 → 0100

§1.5 带符号的二进制数

Signed Binary Numbers

与操作系统和C语言相似，数字电路中的二进制数可以分为有符号(Signed)数和无符号(Unsigned)数，两种数的编码方式不同。

1. 无符号数的编码方式——原码，反码，补码

原码（Sign-magnitude）：二进制数

$$(13)_{10} = (1101)_2$$

1101: 原码

反码　（1’s complement）：

原码全部取反（1变成0，0变成1），为该二进制数的反码。

1 ⟺ 0

1 0 1 1 0 1 0　　原码

↓ ↓ ↓ ↓ ↓ ↓ ↓

0 1 0 0 1 0 1　　反码

1011　的反码为：　0100

补码 （2’s complement ）：

反码末位加1，即为该二进制数的补码

1101	原码
0010	反码
+ 1	
0011	补码

由原码直接求补码:

从右侧数第一个1不动，向左依次求反。

原码	1101	反码求反为原码
补码	0011	补码求补为原码

2.有符号数的编码方式

最左侧一位为符号位:

0 表示正数, 1 表示负数

正数: 0 + 二进制数

符号位0 + 原码

正数

- 原码表示法
- 反码表示法
- 补码表示法

都相同：符号位0 + 原码

+ 13： 0,1101

负数:
原码表示法: 1+原码
反码表示法: 1+反码
补码表示法: 1+补码

$$-13 = -(1101)_2$$

原码表示: 1,1101

反码表示: 1,0010

补码表示: 1,0011

建立原码、补码等负数的不同表示方法，是为了计算机运算方便，快速。 可以证明，以下等式总成立

(X+Y)补 =(X)补+(Y)补 , (X-Y)补 =(X)补+(-Y)补

用补码作减法，可以把减法变加法。这样计算机中只有二进制加法器和求补电路来进行加法和减法运算。

A-B ⟶ A+(-B) (-B)是用补码形式表示的

例： 25 – 13 = 12

25: 原码为 0, 11001

－13: 原码为 1, 01101

补码为 1, 10011

$$\begin{array}{r} 0,\ 11001 \\ +\ 1,\ 10011 \\ \hline 1\quad 0,\ 01100 \end{array}$$

进位溢出

0, 01100为（+12）

例 利用二进制补码计算 13-25=?

解： $(13-25)_{补}=(13)_{补}+(-25)_{补}$

$$
\begin{array}{r}
0,01101 \\
+)\ \ 1,00111 \\
\hline
1,10100
\end{array}
$$

13：原码为 0,01101

-25：补码为 1,00111

符号位为1，负数，对其求补得原码： 1,01100，结果为13-25= -12

3. 偏移码

偏移码的构成：补码的符号位取反

$$-13 \Rightarrow -(1101)_2$$

原码表示: 1,1101

反码表示: 1,0010

补码表示: 1,0011

偏移码表示: 0,0011

等效十进制值	原码	偏移码	补码
+10V +127	01111111	11111111	01111111
+126	01111110	11111110	01111110
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
+5	00000101	10000101	00000101
+4	00000100	10000100	00000100
+3	00000011	10000011	00000011
+2	00000010	10000010	00000010
+1	00000001	10000001	00000001
0V 0	(+0) 00000000	10000000	00000000
	(−0) 10000000		
−1	10000001	01111111	11111111
−2	10000010	01111110	11111110
−3	10000011	01111101	11111101
−4	10000100	01111100	11111100
−5	10000101	01111011	11111011
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
−126	11111110	00000010	10000010
−127	11111111	00000001	10000001
−10V −128		00000000	10000000

偏移码在数字/模拟（D/A）转换中是最容易电路实现的一种码制（详见第9章）

本章总结

- 掌握数制之间的互相转换；
- 理解各种代码的定义；
- 掌握带符号的二进制数的表示方法和运算。

作 业

1.4（1, 2, 3）

1.5（4, 5, 6）

1.6（1, 3）

1.10（1, 3）

1.12（2, 3）

1.14

1.16

1.17 (4, 6)

1.19 (2, 4)