数字电路与系统

课程介绍： 一： 本课程的性质和任务

◆性质：工科电类专业的专业基础课

学习和掌握数字系统、计算机原理、数字通讯、数字控制等方面知识 的入门课程。

◆任务：从应用角度出发，（学习）

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；

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；

使学生具有用硬件和软件设计中、大规模数字系统的基本能力。提高学生数字电路方面的综合素质 。

数字电路与系统

二： 学习本课程的意义/好处/作用

◆ 从个人今后的发展方向看

◆ 实际动手、理论研究

◆用人单位需求

◆考研

◆大学本科必修课，3学分

三： 本课程需要的预备知识

本课程的先修课程为《电路理论》，《模拟电子技术》。

数字电路与系统

四 ： 课程内容安排

1、数制与代码（3学时）
2、逻辑门电路（5学时）
3、逻辑代数与逻辑函数化简（8学时）
4、组合逻辑电路（8学时）
5、触发器（8学时）
6、时序逻辑电路的分析与设计（10学时）
7、脉冲波形的产生与变换（6学时）
8、数模与模数转换（8学时）
9、半导体存储器及可编程逻辑器件（4学时）

数字电路与系统

五 ： 教材及参考书目

教材：《数字电子技术》 王兢主编，电子工业出版社，2007

主要参考书：

1、《数字电子技术》 孟贵骨等，大连理工大学出版社，2002

2、《数字集成电子技术教程》 李士雄、丁康源主编，高等教育出版社

3、《数字电路与系统》 刘玉琴编，清华大学出版社，1993

4、《数字逻辑电路逻辑设计技术》任长明，刘锡海编，天津大学出版社

5、《电子技术基础——数字部分》康华光编，高等教育出版社，2000。

第一章 数字逻辑基础

第一章 数字逻辑基础

绪言 什么是数字信号?

◆ 自然界广泛存在的物理量都是模拟量，如温度、压力等。表示模拟量的信号叫做模拟信号，特点是：

◆ ——信号在时间上和幅度上的取值都是连续的。

◆ 例如：正弦波就是一种典型的模拟信号。

◆ 还有一些物理量，它们在时间和幅度上的取值是不连续的、离散的，这类物理量叫做数字量。表示数字量的信号称为数字信号。

◆ 例如：计算机要对模拟信号进行处理，就必须对模拟信号进行采样，采样的结果就是一种数字信号。

数字信号特点

◆ 数字信号在数值上是离散的，为了便于实现，通常使之只有0、1两种取值，在电路上对应开关的开和闭、电平的高和低。

◆ 每个数字信号只有0、1两种取值，如何表示模拟信号各种不同的幅度呢？——用组合数字信号来描述这个幅度。

◆ 我们以A/D模/数转换为例，在黑板上简单画一画，说明一下。

数字电路的分类

◆ 处理数字信号的电路叫做数字电路，数字电路可以有分立元件组成，更多的是集中制作在一个半导体基片上，称为集成电路。

◆ 集成电路按照集成度的不同分为小规模、中规模、大规模和超大规模。

◆ 集成电路按照逻辑功能的设定来划分，可分为：大、中、小规模通用型 如TTL的74系列，CMOS系列

◆ 专用集成电路ASIC (Application Specific IC)

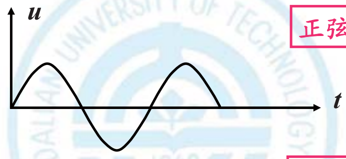
◆ 程逻辑器件PLD (Programmable Logic Device)

1.1 数字电路

数字电路信号

电子电路中的信号
- 模拟信号：随时间连续变化的信号
- 数字信号：时间和幅度都是离散的

模拟信号：



正弦波信号

锯齿波信号

- ◆ 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：

产品数量的统计。

数字表盘的读数。

数字电路信号：

- ◆ 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。

1.2 数制

（1）**十进制**：以十为基数的记数体制

表示数的十个数码：

$1, 2, 3, 4, 5, 6, 7, 8, 9, 0$

遵循逢十进一的规律

$$157 = 1\times10^2 + 5\times10^1 + 7\times10^0$$

一个十进制数数 N可以表示成：

$$(N)_D = \sum_{i=-\infty}^{\infty} K_i \times 10^i$$

若在数字电路中采用十进制，必须要有十个电路状态与十个记数码相对应。这样将在技术上带来许多困难，而且很不经济。

（2）**二进制**：以二为基数的记数体制

表示数的两个数码：

$0, 1$

遵循逢二进一的规律

$$(N)_B = \sum_{i=-\infty}^{\infty} K_i \times 2^i$$

$$(1001)_B = 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0$$
$$= (9)_D$$

优缺点

- ✔ 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。
- ✔ 位数较多，使用不便；不符合人们的习惯，输入时将十进制转换成二进制，运算结果输出时再转换成十进制数。

1.3 数制间的转换

例题1.1 请将二进制转成十进制

解：

数码为：0、1；基数是2。

运算规律：逢二进一，即：1＋1＝10。

二进制数的权展开式：

如：$(101.01)_2 = 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} = (5.25)_{10}$

各数位的权是 2 的幂

（3）十六进制和八进制：

十六进制记数码：

1, 2, 3, 4, 5, 6, 7, 8, 9, A(10), B(11), C(12), D(13), E(14), F(15)

$(4E6)_H = 4\times16^2 + 14\times16^1 + 6\times16^0$

$= (1254)_D$

十六进制与二进制之间的转换

每四位2进制数对应一位16进制数

$(0101\ 1001)_B = [0\times2^7 + 1\times2^6 + 0\times2^5 + 1\times2^4 + 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0]_B$

$= [(0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0)\times16^1 + (1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0)\times16^0]_B$

$= (59)_H$

二进制与十六进制之间的转换

从末位开始四位一组

$(1001110010110100 1000)_B =$

$(1001\ 1100\ 1011\ 0100\ 1000)_B =$

$(9\quad C\quad B\quad 4\quad 8)_H$

$= (9CB48)_H$

例题1.2 请将二进制转成十六进制

解：

数码为：0～9、A～F；基数是16。

运算规律：逢十六进一，即：F＋1＝10。

十六进制数的权展开式：

如：$(D8.A)_2 = 13\times16^1 + 8\times16^0 + 10\times16^{-1} = (216.625)_{16}$

各数位的权是16的幂

例题1.3 请将二进制转成十六进制

解：

二进制数与十六进制数的相互转换，按照每4位二进制数对应于一位十六进制数进行转换。

$0\ 0\ 0\ 1\ |\ 1\ 1\ 0\ 1\ |\ 0\ 1\ 0\ 0\ .\ 0\ 1\ 1\ 0 = (1E8.6)_{16}$

$(AF4.76)_{16} = 1010\ \ 1111\ \ 0100\ .\ 0111\ \ 0110$

八进制与二进制之间的转换：

从末位开始三位一组

$(1001110010110100 1000)_B =$

$(10\ 011\ 100\ 101\ 101\ 001\ 000)_B =$

$(2\quad 3\quad 4\quad 5\quad 5\quad 1\quad 0)_O$

$= (2345510)_O$

其中O表示：Octal—八进制

例题1.4 请将八进制转成二进制

解：

八进制数转换为二进制数：将每位八进制数用3位二进制数表示。

$(374.26)_8 = 011\ \ 111\ \ 100\ .\ 010\ \ 110$

例题1.5 请将八进制转成十进制

解：

数码为：0～7；基数是8。

运算规律：逢八进一，即：7＋1＝10。

八进制数的权展开式：

如：$(207.04)_8 = 2\times8^2 + 0\times8^1 + 7\times8^0 + 0\times8^{-1} + 4\times8^{-2} = (135.0625)_{10}$

各数位的权是8的幂

例题1.6 请将二进制转成八进制

解：

二进制数转换为八进制数： 将二进制数由小数点开始，整数部分向左，小数部分向右，每3位分成一组，不够3位补零，则每组二进制数便是一位八进制数。

0 0 1 | 1 0 1 | 0 1 0 . 0 1 0 = $(152.2)_8$

（4）十进制与二进制之间的转换：

十进制与二进制之间的转换，可以用二除十进制数，余数是二进制数的第0位，然后依次用二除所得的商，余数依次是K1、K2、……。

转换过程：

2	25	……余 1 ……	K_0
2	12	……余 0 ……	K_1
2	6	……余 0 ……	K_2
2	3	……余 1 ……	K_3
2	1	……余 1 ……	K_4
	0		

$(25)_D=(11001)_B$

带小数的十进制与二进制之间的转换：

采用的方法 — 基数连除、连乘法

原理：将整数部分和小数部分分别进行转换。

整数部分采用基数连除法，小数部分采用基数连乘法。转换后再合并。

整数部分采用基数连除法，先得到的余数为低位，后得到的余数为高位。

		余数	
2	44		低位
2	22	…… $0=K_0$	↑
2	11	…… $0=K_1$	
2	5	…… $1=K_2$	
2	2	…… $1=K_3$	
2	1	…… $0=K_4$	
	0	…… $1=K_5$	高位

小数部分采用基数连乘法，先得到的整数为高位，后得到的整数为低位。

	整数	
0.375 × 2		高位
0.750	…… $0=K_{-1}$	↓
0.750 × 2		
1.500	…… $1=K_{-2}$	
0.500 × 2		
1.000	…… $1=K_{-3}$	低位

所以：$(44.375)_{10}=(101100.011)_2$

采用基数连除、连乘法，可将十进制数转换为任意的N进制。

几种进制数之间的对应关系

十进制数	二进制数	八进制数	十六进制数
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F

1.4 代码

◆数字系统只能识别0和1，怎样才能表示更多的数码、符号、字母呢？用编码可以解决此问题。

◆用一定位数的二进制数来表示十进制数码、字母、符号等信息称为编码。用以表示十进制数码、字母、符号等信息的一定位数的二进制数称为代码。

◆二-十进制代码：用4位二进制数b3b2b1b0来表示十进制数中的 0 ~ 9 十个数码。简称BCD码。用四位自然二进制码中的前十个码字来表示十进制数码，因各位的权值依次为8、4、2、1，故称8421 BCD码。

◆2421码的权值依次为2、4、2、1；

◆余3码由8421码加0011得到；

◆格雷码是一种循环码，其特点是任何相邻的两个码字，仅有一位代码不同，其它位相同。

1）BCD码

数字电路中编码的方式很多，常用的主要是二—十进制码（BCD码）。

BCD------Binary-Coded-Decimal

用四位二进制数表示0~9十个数码，即为BCD码 。四位二进制数最多可以有16种不同组合，不同的组合便形成了一种编码。主要有：8421码、 5421码、2421码、余3码等。

在BCD码中，十进制数与二进制编码 $(K_3K_2K_1K_0)_B$ 的关系可以表示为：

$$(N)_D= W_3K_3 +W_2K_2+W_1K_1+W_0K_0$$

W_3~W_0为二进制各位的权重

所谓的8421码，就是指各位的权值是8，4，2，1。

2） 格雷码

- 各种格雷码的共同特点是：任意两个相邻码之间只有一位不同。
- 格雷码的这个特点使得它在传输过程中引起的误差较小。

序号	二进制	Gray码
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

3） 字符代码

- “0”和“1”用来组合代表字母和符号的代码。
- 例如：ASCII码（American Standard Code for Information Interchange），一共有七位信息码，不同的字符组合代表不同的含义。

ASCII码表

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

例题1.7 请将 8421BCD 码转换成10进制码

解：

- (1) $(0101\ 1000)_{8421BCD}=$
- (2) $(1001\ 0011\ 0101)_{8421BCD}=$
- (3) $(0011\ 0100\ .0111\ 0001)_{8421BCD}=$
- (4) $(0111\ 0101\ .0110)_{8421BCD}=$

1.5 带符号的二进制数

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(即可以比较大小的正负数) 和无符号(Unsigned)数(如上节所述)，这两种数的编码方式是不一样的。

1） 无符号数的编码方式

原码：其值为二进制码本身

反码：其值为二进制码的各位取反得到的二进制

补码：其值为二进制码的各位取反得到的二进制，并在其最低有效位上加‘1’

例如：无符号十进制数 6 、4.5，其 四位二进制表示分别为 0110、0100.10，将其表示为以上编码，则其结果分别为

原码：0110 、 0100.10

反码：1001 、 1011.01 $(N_{补})_{补}=N_{原}$

补码：1010 、 1011.10 $(N_{反})_{反}=N_{原}$

数字电路最常用的二进制编码是下面要讲的有符号数编码。

2） 有符号数的编码方式

一个二进制正负数的真值，可以有多种编码方法表示，因此，看到一个二进制数串，先要确定它是哪种码，然后才可以计算出其大小。

常用的正负数二进制编码方式有：原码、反码、补码和偏移码(由补码求得)等。

以上各种二进制数编码均可以用来表示正负数值，方法是在编码的首位加上符号位(1或0)，此时，编码串的形式为：“1,数值部分” 或 “0,数值部分”。对原码、反码、补码而言，符号位为 1 表示此二进制编码为负数；为 0 表示此二进制编码为正数。

◆正数的编码原则

原码、反码、补码 } 符号位为0，数值部分为正数对应的二进制数串

◆负数的编码原则

- 原码：符号位为1，数值部分为此负数的绝对值对应的二进制数
- 反码：符号位为1，数值部分为此负数的绝对值对应的二进制数的反码
- 补码：符号位为1，数值部分为此负数的绝对值对应的二进制数的补码

◆偏移码的编码原则： 无论正负数，偏移码都由补码的符号位取反求得

例如：常用三位原码，反码，补码，偏移码：

真值	原码	反码	补码	偏移码
3	0,11	0,11	0,11	1,11
2	0,10	0,10	0,10	1,10
1	0,01	0,01	0,01	1,01
0	0,00	0,00	0,00	1,00
-0	1,00	1,11	(000)	(1,00)
-1	1,01	1,10	1,11	0,11
-2	1,10	1.01	1,10	0,10
-3	1,11	1,00	1,01	0,01

例如：原码，反码，补码，偏移码：

真值	原码	反码	补码	偏移码
+1101	0,1101	0,1101	0,1101	1,1101
+ 10011.01	0,10011.01	0,10011.01	0,10011.01	1,10011.01
+110000	0,110000	0,110000	0,110000	1,110000
+100000	0,100000	0,100000	0,100000	1,100000
-1101	1,1101	1,0010	1,0011	0,0011
-10011.01	1,10011.01	1,01100.10	1,01100.11	0,0110011
-110000	1,110000	1,001111	1,010000	0,010000
-100000	1,100000	1,011111	1,100000	0,100000

3） 已知二进制编码，求其对应的数值

给定一个二进制编码数串及其编码方式(原码、反码、补码或偏移码)，并且已知其为有(无)符号数，可根据以下原则求其对应的数值

编码串
- 无符号
  - 原码：值为二进制数本身
  - 反码：其值为各数码取反后的二进制数的值
  - 补码：将各数码位取反，在最低有效位加1所得二进制的值
- 有符号
  - 符号位:0
    - 原码：值为数值部分二进制数本身
    - 反码：值为数值部分二进制数本身
    - 补码：值为数值部分二进制数本身
    - 偏移码：符号位取反为1，后按符号位为1的补码对待
  - 符号位:1
    - 原码：值为数值部分二进制数本身，但是为负值
    - 反码：对数值部分取反，所得二进制的值，但是为负值
    - 补码：对数值部分取反，在最低有效位加1，所得二进制的负值
    - 偏移码：符号位取反为0，后按符号位为0的补码对待

思考：此处实际如何确定其值？（参考前2表）

例题1.8 例如：已知无符号二进制编码如下，求其对应的值。

解：

101011.11 (原码) → 43.75

101011.11 (反码) → 20

101011.11 (补码), → 20.25

例如：已知二进制编码如下，求其对应的值。

1,1010 (原码), → -10

1,1010 (反码), → -5

1,1010 (补码), → -6

1,1010 (偏移码), → 10

0,1010 (原码), → 10

0,1010 (反码), → 10

0,1010 (补码), → 10

0,1010 (偏移码), → -6

4） 补码的应用

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

补码可把减法运算变成加法运算。从而可把乘、除、乘方、开方等运算变成加法运算。

在同号相加时要预先判断其和是否溢出，如果溢出，可通过增加字长来解决。

在没有溢出的前提下，相加时符号位也参加运算，如果符号位产生进位，舍去最前面的进位位，保留符号位位置上的数，作为运算结果的符号位。

加法运算的加数、被加数和结果都是补码形式。可依据19页的方法求得真实数值。

例 利用二进制补码计算 25-13=?

$25-13=(25)_{补}+(-13)_{补}$

$$\begin{array}{r} 0{,}11001 \\ +)\ 1{,}10011 \\ \hline 10{,}0110(\end{array}$$

解：25的原码为 0，11001

25的补码为 0，11001

13的原码为 0，01101

-13的补码为 1，10011　　舍去最高位进位位，结果为 25-13=12 。

例题1.9 利用二进制补码计算 13-25=?

解： $13-25=(13)_{补}+(-25)_{补}$

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

13的原码为 0，01101

25的补码为 0，11001

-25的补码为 1，00111

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

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。

第一章 数字逻辑基础

结束