数字逻辑电路
(2020级本科生课程)

清华大学计算机系
陶品
taopin@tsinghua.edu.cn
办公室：FIT 3－531 （13717813059）

课程基本情况（5）

- 教学形式：讲课，辅导，网上、定时和现场答疑
- 学时安排：3学分，共48学时
- 教学内容
  - 第一章：前言和基本知识介绍（第1周，前3学时）
  - 第二章 逻辑代数和化简方法（第2、3周，共6学时）
  - 第三章：门电路（第4周，共3学时）
  - 第四章：组合逻辑电路（第5～7周，共9学时）
  - 第五章：时序逻辑电路（第8～12周，共15学时）
  - 第六章：可编程逻辑电路（第13～15周，共9学时）
  - 数字逻辑电路课程总复习（第16周，3学时）

作 业

- 2.6（1）～（6）

2.6 用公式法将下列函数化简为最简“与或”式。

(1) $F=\overline{A}\,\overline{B}+(AB+A\overline{B}+\overline{A}B)C$

(2) $F=(X+Y)Z+\overline{X}\,\overline{Y}W+ZW$

(3) $F=AB+\overline{A}C+\overline{B}\,\overline{C}$

(4) $F=AB+\overline{A}\,\overline{B}C+BC$

(5) $F=\overline{A}B+\overline{A}C+\overline{B}\,\overline{C}+AD$

(6) $F=\overline{A}\,\overline{B}+\overline{A}\,\overline{C}D+AC+B\overline{C}$

(7) $F=AC+\overline{A}\,\overline{B}+\overline{B}\,\overline{C}\,\overline{D}+BE\overline{C}+DE\overline{C}$

(8) $F=A(B+\overline{C})+\overline{A}(\overline{B}+C)+BCD+\overline{B}\,\overline{C}D$

(9) $F=\overline{X}\,\overline{Y}+(X+Y)Z$

(10) $F=(X+Y+Z+\overline{W})(V+X)(\overline{V}+Y+Z+\overline{W})$

作业

- 课代表收本班作业，下周一9:00-17:00交给助教，同时取回助教批改后的上一次作业。
- 因特殊情况未能按时完成，请在第16周结束之前自行联系助教补交，补交的每份作业扣0.5分，缺交一次扣2分（平时成绩共计20分）。
- 因疫情及担任冬奥志愿者等不在清华需要通过线上方式上课的同学，请将作业电子版交至助教email邮箱。其他同学必须提交纸质作业。

计01-05班
交至助教官佳智
guanjz20@mails.tsinghua.edu.cn
FIT楼4-104

计06-08、信计01班及其他
交至助教黄天
ht20@mails.tsinghua.edu.cn
FIT楼3-526

第一章 数制和码制

本章内容

1.1 数制

1.2 码制

数制和码制（1）

- 数制
  - 十进制、二进制、八进制、十六进制
  - 数制转换
- 码制
  - 十进制数的二进制编码
    - 8421码（BCD码）、余3码、余3循环码、格雷码
  - 字符编码
    - 7位ASCII、8位ASCII

数制和码制（2）

- 数制
  - 一个进位计数包含两个基本因素
    - 基数
    - 位权

$$143.75 = 1 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}$$

数制和码制（3）

■ 数制

- 一个进位计数包含两个基本因素
  - 基数：
    - 数制中数码的个数
    - 基数为R的数制（简称R进制）包含0～R－1的数码
    - 进位规律是“逢R进一”。

$$143.75=\boxed{1}\times 10^{2}+\boxed{4}\times 10^{1}+\boxed{3}\times 10^{0}+\boxed{7}\times 10^{-1}+\boxed{5}\times 10^{-2}$$

数制和码制（4）

数制

- 一个进位计数包含两个基本因素
  - 位权：
    - R进制数中处于不同位数的数码，代表不同的权值
    - 某个数位上的数值由本位数码的值乘以本位对应的固定常数
    - 该固定常数称为位权
    - 不同的数位有不同的位权

$$143.75 = 1 \times \boxed{10^2} + 4 \times \boxed{10^1} + 3 \times \boxed{10^0} + 7 \times \boxed{10^{-1}} + 5 \times \boxed{10^{-2}}$$

数制和码制（5）

- R进制数N的表示方式
  - 并列表示方式

$$(N)_R=(K_{n-1}K_{n-2}\cdots K_1K_0K_{-1}K_{-2}\cdots K_{-m})_R$$

✓ n为整数部分的数位，m为小数部分的数位，R表示基数

  - 多项式标识方式

$$(N)_R=(K_{n-1}\times R^{n-1}+\cdots K_0\times R^0+K_{-1}\times R^{-1}\cdots K_{-m}\times R^{-m})_R$$

或 $(N)_R=(\sum_{i=-m}^{n-1}K_i\times R^i)_R$

✓ K_i表示不同数位的数码，$(R-1)\geq K_i\geq 0$

数制和码制（6）

- 十进制数
  - 基数为10，数码0～9
  - 任意一个十进制数可以表示为

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

  - 权值10^i
  - 如143.75

$$143.75 = 1 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}$$

数制和码制（7）

- 二进制数
  - 基数为2，数码0，1
  - 任意一个二进制数可以表示为

$$D = \sum k_i \times 2^i$$

  - 权值2^i
  - 如101.11

$$(101.11)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$
$$= (5.75)_{10}$$

数制和码制（8）

- 八进制数
  - 基数为8，数码0～7
  - 任意一个八进制数可以表示为

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

- 十六进制数
  - 基数为16，数码0～9，A B C D E F
  - 任意一个十六进制数可以表示为

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

这是什么数制和码制？

- M（Money）数
  - 基数为N，位权分比为1、2、5、10
  - 任意一个M数可以表示为
    - $(2103)_M = 2\times10 + 1\times5 + 0\times2 + 3\times1 = (28)_{10}$

数制和码制（9）

- 数制转换
  - 二－十进制互换
  - 二－八进制互换
  - 二－十六进制互换

- 小常识

  K，M，G，T，P，E，Z，Y，B……各自的含义

数制和码制（10）

二－十进制互换

- 二－十进制转换：将（1101)$_2$转成十进制数

$$(1101)_2 = 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 = (13)_{10}$$

- 十－二进制转换：将75转成二进制数

2	75	余数 = 1
2	37	余数 = 1
2	18	余数 = 0
2	9	余数 = 1
2	4	余数 = 0
2	2	余数 = 0
2	1	余数 = 1
	0	

$(75)_{10} =$
$(1001011)_2$

基数转换中常用的数

n	2^n	n	2^n	n	2^n
0	1	8	256	16	65,536
1	2	9	512	17	131,072
2	4	10	1,024	18	262,144
3	8	11	2,048	19	524,288
4	16	12	4,096	20	1,048,576
5	32	13	8,192	21	2,097,152
6	64	14	16,384	22	4,194,304
7	128	15	32,768	23	8,388,608

表1-1　　2的幂表

数制和码制（11）

- 二－八进制互换
  - 二－八进制转换：将（1101101）$_2$转成八进制数

$$(1101101)_2 = (1,101,101)_2 = (155)_8$$

  - 八－二进制转换：将（573）$_8$转成二进制数

$$(573)_8 = (5,7,3)_8 = (101,111,011)_2$$

$$= (101\ 111\ 011)_2$$

数制和码制（12）

- 二－十六进制互换
  - 二－十六进制转换：将（1101101）$_2$转成十六进制数

$$(1101101)_2 = (110,1101)_2 = (6D)_{16}$$

  - 十六－二进制转换：将（5A3F）$_{16}$转成二进制数

$$(5A3F)_{16} = (0101,1010,0011,1111)_2$$

$$= (0101\ 1010\ \ 0011\ 1111)_2$$

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
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

表1-2 不同基的码表

数制和码制（13）

- 数制
- 码制
  - 原码、反码、补码
  - 十进制数的二进制编码
    - 8421码（BCD码）、余3码、格雷码
  - 字符编码
    - 7位ASCII、8位ASCII

数制和码制（14）

- 十进制数的二进制编码
  - 8421码（BCD码：binary-code-decimal）(有权码)
    - 4位二进制码从高到低权值为2^3 (8) ,2^2 (4) , 2^1 (2) , 2^0 (1)
  - 5421码 (有权码)
    - 4位二进制码从高到低权值分别为5， 4， 2， 1
  - 2421码 (有权码)
    - 4位二进制码从高到低权值分别为2， 4， 2， 1
  - 余三码 (无权码)
    - 十进制数用余三码表示，比8421码在二进制上多3
    - 余三码＝8421码＋（0011）$_2$

十进制数的二进制编码
8421码、BCD码

Binary_code_decimal

简称为二——十进制码或BCD码，即用若干位(4位)二进制数来表示一位十进制数。

1. 8421 BCD码

简称8421码。按4位二进制数的自然顺序，取前十个数依次表示十进制的0～9，后6个数不允许出现，若出现则认为是非法的或错误的。

8421码是一种**有权码**，每位有固定的权，从高到低依次为8, 4, 2, 1，如：

8421码　$0111=0\times8+1\times4+1\times2+1\times1=7$

例如：$N=(93)_{10}$，则有：$N=(1001\ 0011)_{8421BCD}$等等

2. 5421 BCD码

- 简称5421码。按4位二进制数($B_3B_2B_1B_0$)的自然顺序值，取前十个数依次表示十进制的0～9，5421码也是一种有权码，每位有固定的权，从高到低依次为5421。
- **例如：N=（93）$_{10}$，则有：**

  N=（1100 0011）$_{5421BCD}$

- **10进制5的5421编码应该是?**

3. 2421 BCD码

- 简称2421码。按4位二进制数($B_3B_2B_1B_0$)的自然顺序值，取前十个数依次表示十进制的0～9，2421码是一种有权码，每位有固定的权，从高到低依次为2421。

- **例如：N=93，则有：**

  N=（1111 0011）$_{2421BCD}$

- **10进制8的2421编码应该是?**

无权编码

余3码

8421码再加0011

由8421码加3形成，是一种**无权码**。如果两个余3码相加没有进位，则和数要减3，否则和数要加3。

1　　3　　4　　　5　　6　　11

例如：0100+0110=0111　　　1000+1001=$\boxed{1}$0100

$$
\begin{array}{rcccc}
 & 0 & 1 & 0 & 0 \\
+) & 0 & 1 & 1 & 0 \\
\hline
 & 1 & 0 & 1 & 0 \\
-) & 0 & 0 & 1 & 1 \\
\hline
 & 0 & 1 & 1 & 1
\end{array}
\qquad
\begin{array}{rccccc}
 & & 1 & 0 & 0 & 0 \\
+) & & 1 & 0 & 0 & 1 \\
\hline
 & 1 & 0 & 0 & 0 & 1 \\
+) & & 0 & 0 & 1 & 1 \\
\hline
 & \boxed{1} & 0 & 1 & 0 & 0
\end{array}
$$

数制和码制（15）

四种十进制数的编码表

十进制数	8421码	5421码	2421码	余三码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110
4	0100	0100	0100	0111
5	0101	1000	1011	1000
6	0110	1001	1100	1001
7	0111	1010	1101	1010
8	1000	1011	1110	1011
9	1001	1100	1111	1100

数制和码制（16）

- 格雷码
  - 一种无权码
  - 有多种形式
  - 编码特点：任何相邻的十进制数的格雷码仅有一位不同。
  - 应用优势:可以减少代码变换中电路瞬间产生的错误，可靠性较好。

数制和码制（17）

- 十进制数的4种格雷码表

十进制数	8421码	格雷码1	格雷码2	典型格雷码	修改格雷码
0	0000	0000	0000	0000	0010
1	0001	0001	0001	0001	0110
2	0010	0011	0011	0011	0111
3	0011	0010	0010	0010	0101
4	0100	0110	0110	0110	0100
5	0101	1110	0111	0111	1100
6	0110	1010	0101	0101	1101
7	0111	1011	0100	0100	1111
8	1000	1001	1100	1100	1110
9	1001	1000	1000	1101	1010

数制和码制（18）

格雷码的特点

- 编码特点：任何相邻十进制数的格雷码仅有一位不同
- 格雷码1：除最高位以外，中线对称。
- 典型格雷码：可以对十进制编码，也可以对任意大的二进制数进行编码。
  编码规则：$G_i = B_{i+1} \oplus B_i$
  解码规则：？？？（思考题）
- 修改格雷码：中线对称反射，“余三循环码”（循环码的特点：相邻两个编码之间只有一位数不同，而且首尾两个编码之间也只有一位数不同）

数制和码制（19）

- 字符编码
  - 数字系统中，需要对符号，文字，图象等进行编码
  - 7位ASCII、8位ASCII
- 7位ASCII编码（教材13页）
  - 高3位$b_6b_5b_4$区分
    - $b_6b_5b_4$（000，001）：控制字符
    - $b_6b_5b_4$（010，011）：数字及通用符号
    - $b_6b_5b_4$（100，101）：大写英文字母
    - $b_6b_5b_4$（110，111）：小写英文字母

数制和码制——小结

- 数制
  - 十进制、二进制、八进制、十六进制
  - 数制转换
- 码制
  - 十进制数的二进制编码
    - 8421码（BCD码）、余3码、余3循环码、格雷码
  - 字符编码
    - 7位ASCII、8位ASCII
  - 原码、补码与反码：带符号的二进制数编码，将在计算机原理课程学习

第二章 逻辑代数（1）

本章内容

2.1 逻辑代数的基本运算与公式

2.2 公式法化简逻辑函数

2.3 图解法(卡诺图)化简逻辑函数

2.4 逻辑函数的表格法化简(Q-M法)

第二章 逻辑代数（2）

2.1 逻辑代数的基本运算与公式

2.2 公式法化简逻辑函数

2.3 图解法(卡诺图)化简逻辑函数

2.4 逻辑函数的表格法化简(Q-M法）

第二章 逻辑代数（3）

2.1 逻辑代数的基本运算与公式

- 逻辑代数：又叫布尔代数，开关代数。是二进制运算的基础，用代数方法研究逻辑问题。由英国数学家布尔和德.摩根于1847年提出。
- 逻辑函数的表示：真值表，表达式，逻辑门。
- 逻辑代数的基本运算：与、或、非

  (1) 与运算，逻辑乘

  (2) 或运算，逻辑加

  (3) 非运算，取反

第二章 逻辑代数（4）

2.1 逻辑代数的基本运算与公式

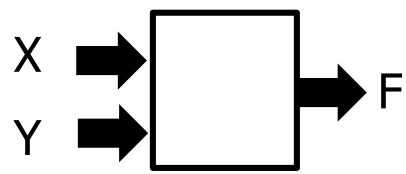
- 逻辑代数和普通代数的共同之处：变量和变量的运算。
- 逻辑代数一般用字母表示变量。
- 逻辑变量的取值："0"或"1"。
- 逻辑代数的基本运算：与、或、非

第二章 逻辑代数（4）

- 2.1 逻辑代数的基本运算与公式
  - 逻辑代数的基本运算
  - 逻辑代数的基本公式

第二章 逻辑代数（5）

- 对于一个2输入变量，一共有多少种逻辑？



X	Y	16种可能的逻辑函数(F_0–F_{15})															
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

0

X AND Y

X

Y

X XOR Y

X OR Y

X NOR Y
NOT (X OR Y)

X = Y

NOT Y

NOT X

X NAND Y
NOT (X AND Y)

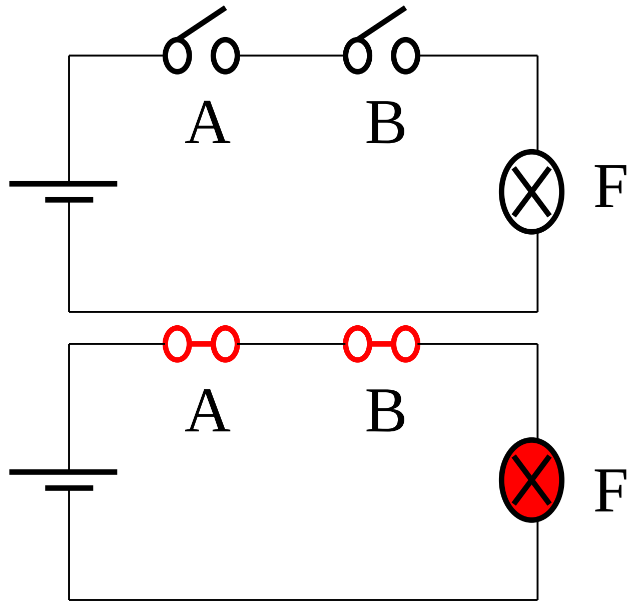
1

第二章 逻辑代数（6）

2.1 逻辑代数的基本运算与公式

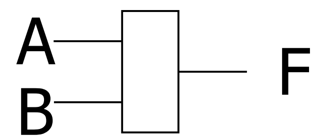
- 逻辑代数的基本运算：与、或、非

(1) 与运算，逻辑乘



A	B	F
0	0	0
1	0	0
0	1	0
1	1	1

$$F = A \bullet B$$



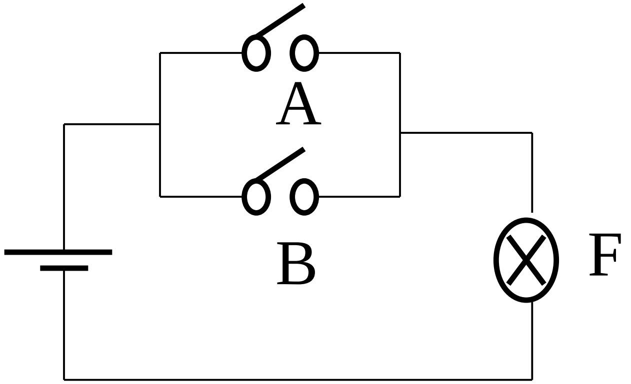
第二章 逻辑代数（7）

2.1 逻辑代数的基本运算与公式

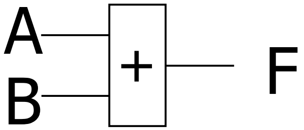
- 逻辑代数的基本运算：与、或、非

(2) 或运算，逻辑加

A	B	F
0	0	0
1	0	1
0	1	1
1	1	1



$$F = A + B$$



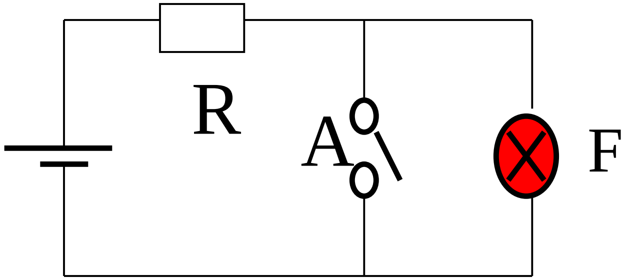
第二章 逻辑代数（8）

2.1 逻辑代数的基本运算与公式

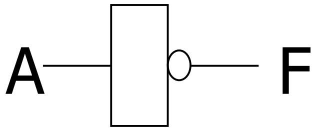
- 逻辑代数的基本运算：与、或、非

(3) 非运算，取反

A	F
0	1
1	0



$$F = \overline{A}$$



第二章 逻辑代数（9）

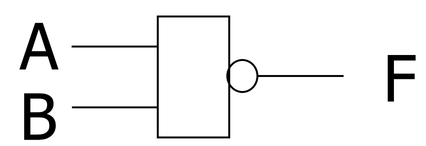
2.1 逻辑代数的基本运算与公式

其他基本的逻辑电路

- "与非"门：实现"与非"运算的电路。

A	B	F'＝AB	F=$\overline{AB}$
0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	0

$$F = \overline{AB}$$



第二章 逻辑代数（10）

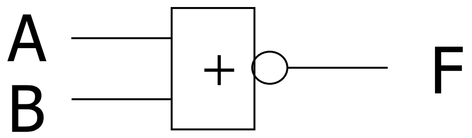
2.1 逻辑代数的基本运算与公式

其他基本的逻辑电路

- "或非"门：实现"或非"运算的电路。

A	B	F'＝A＋B	F=$\overline{A+B}$
0	0	0	1
1	0	1	0
0	1	1	0
1	1	1	0

$$F = \overline{A + B}$$



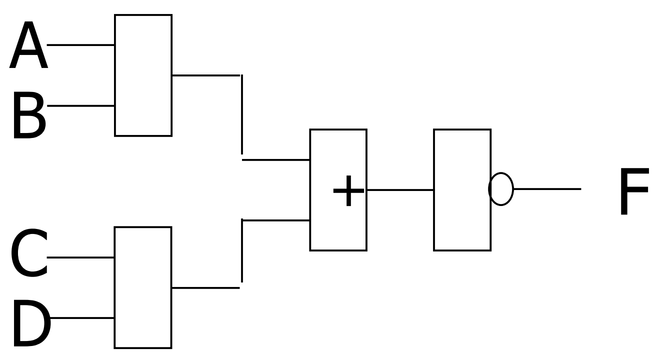
第二章 逻辑代数（11）

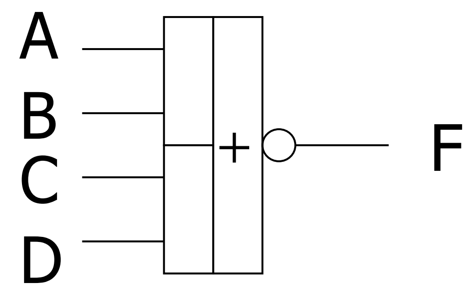
2.1 逻辑代数的基本运算与公式

其他基本的逻辑电路

- “与或非”门：实现“与或非”运算的电路。

$$F = \overline{AB + CD}$$





“与或非”门

第二章 逻辑代数（12）

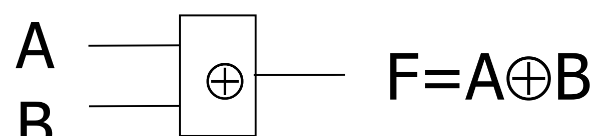
2.1 逻辑代数的基本运算与公式

其他基本的逻辑电路

- “异或”门：实现“异或”运算的电路。

$$F = A \oplus B = A\overline{B} + \overline{A}B$$

A	B	F
0	0	0
1	0	1
0	1	1
1	1	0



“异或”门

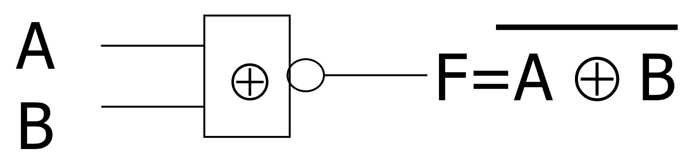
第二章 逻辑代数（13）

2.1 逻辑代数的基本运算与公式

- **其他基本的逻辑电路**
  - “同或”门(“异或非”门)：实现“同或”运算的电路。

$$F = \overline{A \oplus B} = AB + \overline{A}\,\overline{B}$$

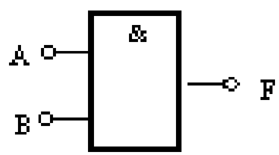
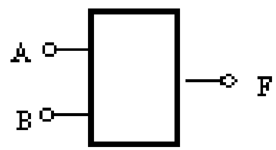
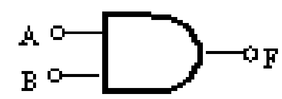
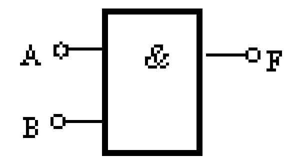
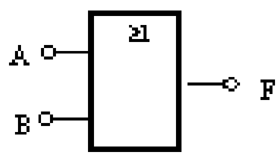
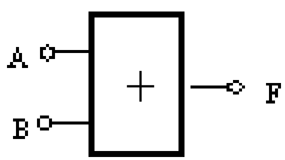
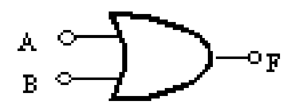
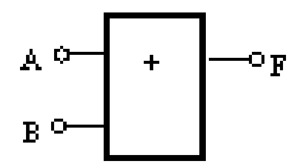
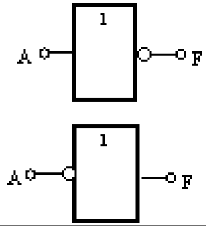
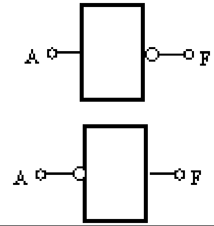
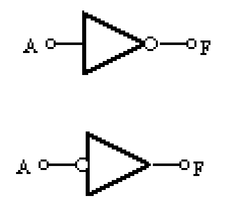
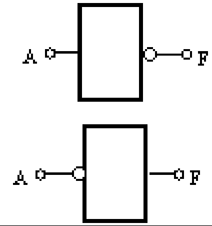
A	B	F
0	0	1
1	0	0
0	1	0
1	1	1



“同或”门

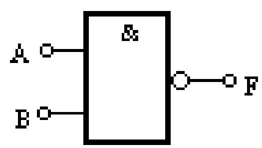
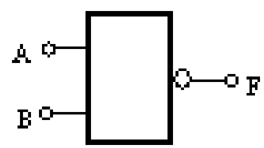
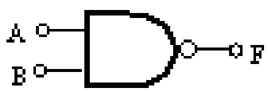
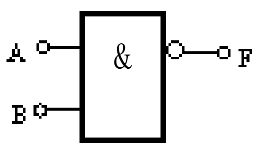
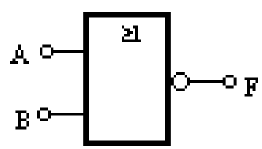
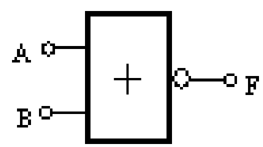
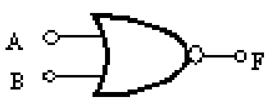
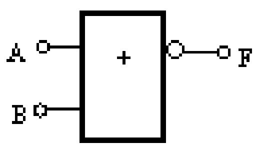
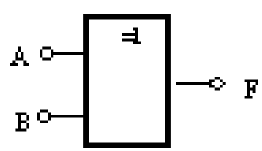
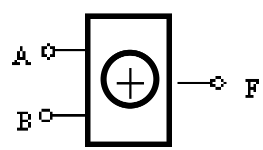
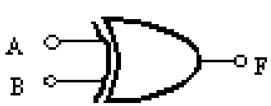
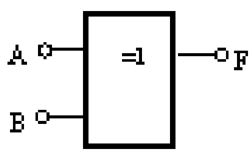
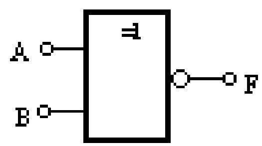
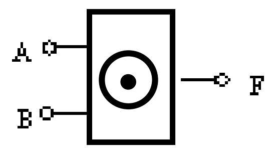
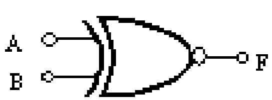
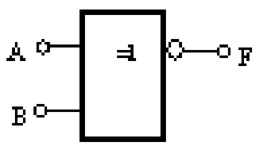
第二章 逻辑代数（14）

■ 常用逻辑门符号

符号名称	国标符号	常用符号	国际流行符号	IEEE逻辑符号
与门				
或门				
非门				

第二章 逻辑代数（15）

常用逻辑门符号

符号名称	国标符号	常用符号	国际流行符号	IEEE逻辑符号
与非门				
或非门				
异或门				
同或门				

第二章 逻辑代数（13）

2.1 逻辑代数的基本运算与公式

逻辑代数的基本运算——小结

- 基本运算：与、或、非
  - 电路：“与”门，“或”门，“非”门
- 常用运算：与非、或非、与或非、异或、同或
  - 电路：“与非”门，“或非”门，“与或非”门
    “异或”门，“同或”门

第二章 逻辑代数（14）

- 2.1 逻辑代数的基本运算与公式
  - 逻辑代数的基本运算
  - 逻辑代数的基本公式

第二章 逻辑代数（15）

2.1 逻辑代数的基本运算与公式

- 逻辑代数的基本公式(Elementary)

互补律 $\begin{cases} A \bullet \overline{A} = 0 \\ A + \overline{A} = 1 \end{cases}$

1律 $\begin{cases} 1 \bullet A = A \\ 1 + A = 1 \end{cases}$

0律 $\begin{cases} 0 \bullet A = 0 \\ 0 + A = A \end{cases}$

重叠律 $\begin{cases} A \bullet A = A \\ A + A = A \end{cases}$

对合律 $\overline{\overline{A}} = A$

第二章 逻辑代数（16）

2.1 逻辑代数的基本运算与公式

逻辑代数的基本公式

交换律 (Commutativity)

$$\begin{cases} A \bullet B = B \bullet A \\ A + B = B + A \end{cases}$$

结合律 (Associativity)

$$\begin{cases} A \bullet (B \bullet C) = (A \bullet B) \bullet C \\ A + (B + C) = (A + B) + C \end{cases}$$

分配律 (Distributivity)

$$\begin{cases} A \bullet (B + C) = A \bullet B + A \bullet C \\ A + B \bullet C = (A + B) \bullet (A + C) \end{cases}$$

第二章 逻辑代数（17）

2.1 逻辑代数的基本运算与公式

- 逻辑代数的基本公式

吸收律

(Absorption)

$$\begin{cases} A + \overline{A}B = A + B \\ A \bullet (\overline{A} + B) = A \bullet B \\ A + A \bullet B = A \\ A(A + B) = A \end{cases}$$

反演律(德·摩根定律)

(De Morgan's)

$$\begin{cases} \overline{A \bullet B} = \overline{A} + \overline{B} \\ \overline{A + B} = \overline{A} \bullet \overline{B} \end{cases}$$

第二章 逻辑代数（18）

2.1 逻辑代数的基本运算与公式

- 逻辑代数的基本公式

包含律
(Consensus)

$$\begin{cases} AB + \overline{A}C + BC = AB + \overline{A}C \\ (A+B)(\overline{A}+C)(B+C) = (A+B)(\overline{A}+C) \end{cases}$$

第二章 逻辑代数（19）

反演规则

- 设F是逻辑函数，如将函数F中所有“与”符号换为“或”符号；所有“或”符号换为“与”符合；所有原变量换为反变量；所有反变量换为原变量；“0”换成“1”；“1”换成“0”，所得新的逻辑表达式为原函数的反函数。
- 利用反演规则可以方便地求得函数的反函数。

对偶规则

- 如果将逻辑函数F中的“与”符号换为“或”符号；将“或”符号换为“与”符号；将“1”换为“0”，将“0”换为“1”；但逻辑变量不进行反变换，得到新的逻辑函数表达式，记作F’，把F’称为F的对偶式，获得对偶式的规则称为对偶规则。

第二章 逻辑代数（20）

2.1 逻辑代数的基本运算与公式

逻辑代数的基本公式

（可适用任意多个变量，以3变量为例）

反演规则：

$$F = ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C}$$

$$\overline{F} = (\overline{A}+\overline{B}+\overline{C})(\overline{A}+\overline{B}+C)(\overline{A}+B+\overline{C})(\overline{A}+B+C)$$

对偶规则：

$$F = A(B+C)$$

$$F' = A + BC$$

- 函数F和G相等，则F’和G’也相等。
- 对偶的对偶，为函数自身。

第二章 逻辑代数（21）

2.1 逻辑代数的基本运算与公式

逻辑代数的基本公式

包含律推论 $AB + \overline{A}C + BCD = AB + \overline{A}C$

证明：

$$
\begin{aligned}
AB + \overline{A}C + BCD &= AB + \overline{A}C + BC + BCD \\
&= AB + \overline{A}C + BC(1 + D) \\
&= AB + \overline{A}C + BC \\
&= AB + \overline{A}C
\end{aligned}
$$

第二章 逻辑代数（22）

2.1 逻辑代数的基本运算与公式 ——小结

- 逻辑代数的基本运算：基本运算、常用运算
- 逻辑代数的基本公式：

  各种运算基本公式

  提醒注意“吸收律”，“包含律”

第二章 逻辑代数（23）

2.1 逻辑代数的基本运算与公式

2.2 公式法化简逻辑函数

2.3 图解法(卡诺图)化简逻辑函数

2.4 逻辑函数的表格法化简(Q-M法）

第二章 逻辑代数（24）

2.2 公式法化简逻辑函数

化简的目的

- ✓让逻辑函数表达式的含义更清楚。
- ✓节省资源：用最少的电路、器件，器件的输入也最少。
- ✓化简目的不同，表达式结果会不同。

第二章 逻辑代数（25）

- 2.2 公式法化简逻辑函数
  - 与或表达式化简
  - 或与表达式化简
  - 其他形式逻辑函数化简

第二章 逻辑代数（26）

与或表达式化简

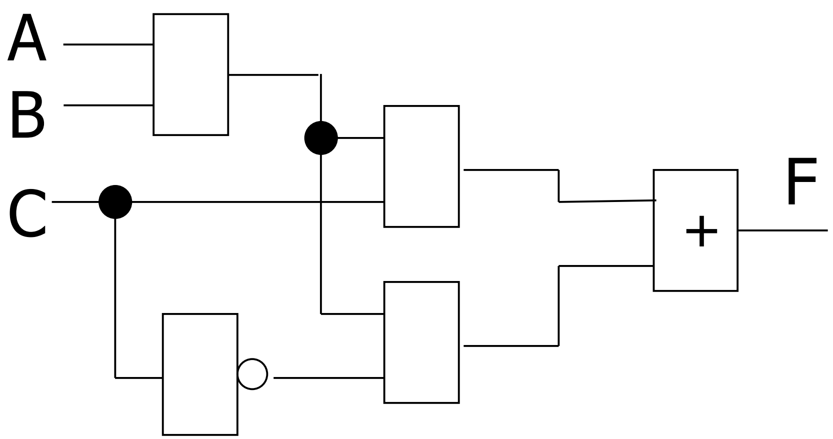
最简与或表达式：

1、乘积的个数最少(用门电路实现，用的与门数最少)

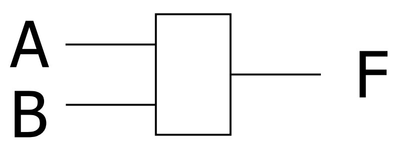
2、在满足1的条件下，乘积项中的变量最少(与门的输入端最少)

$$F = ABC + AB\overline{C}$$

表达式是否为最简?



$$F = AB$$



第二章 逻辑代数（27）

■ 与或表达式化简

➢ 例1

$$F = A(BC + \overline{B}\,\overline{C}) + AB\overline{C} + A\overline{B}C$$

展开： $= ABC + A\overline{B}\,\overline{C} + AB\overline{C} + A\overline{B}C$

合并： $= (ABC + A\overline{B}C) + (A\overline{B}\,\overline{C} + AB\overline{C})$

分配律： $= AC(B + \overline{B}) + A\overline{C}(\overline{B} + B)$

互补律： $= AC + A\overline{C} = A$

第二章 逻辑代数（28）

- 与或表达式化简
  - 例2

$$F = A(B + C) + \overline{B}\,\overline{C}$$

反演律 $\overline{A \bullet B} = \overline{A} + \overline{B}$ $= A \bullet \overline{\overline{B}\,\overline{C}} + \overline{B}\,\overline{C}$

吸收律 $A + \overline{A}B = A + B$ $= A + \overline{B}\,\overline{C}$

第二章 逻辑代数（29）

与或表达式化简

- 例3

$$F = AB + \overline{A}C + BCD + ABD$$

合并：$= (AB + ABD) + \overline{A}C + BCD$

分配律：$= AB(1 + D) + \overline{A}C + BCD$

1律：$= AB + \overline{A}C + BCD$

吸收律：$= AB + \overline{A}C$

例题

$$F = AB + \overline{A}C + (\overline{B} + \overline{A}D)CE + \overline{A}CD + \overline{B}C$$

答案：

$$F = AB + C$$

第二章 逻辑代数（28）

- 2.2 公式法化简逻辑函数
  - 与或表达式化简
  - 或与表达式化简
  - 其他形式逻辑函数化简

或与表达式化简

- 或与表达式：

$$F = (A+B)(\overline{A}+C)(C+DE)$$

结果要求：化简结果仍为或与表达式

最简条件：

(1)、或项个数最少(或门用的最少)

(2)、在满足1的条件下，或项中变量数最少

例题

$$F = (A + B + \overline{C})(\overline{A} + B + C)(A + \overline{B} + \overline{C})(\overline{A} + \overline{C})$$

答案：

$$F = \overline{A}\,\overline{C} + B\overline{C} = (\overline{A} + B)\overline{C}$$

第二章 逻辑代数（30）

- 2.2 公式法化简逻辑函数
  - 与或表达式化简
  - 或与表达式化简
  - 其他形式逻辑函数化简

第二章 逻辑代数（31）

2.2 公式法化简逻辑函数

其他形式逻辑函数化简

转换为与或表达式，再化简

$$
\begin{aligned}
F &= \overline{A \oplus B} + \overline{B \oplus C} \\
&= AB + \bar{A}\bar{B} + BC + \bar{B}\bar{C} \\
&= AB(C + \bar{C}) + \bar{A}\bar{B} + BC + \bar{B}\bar{C}(A + \bar{A}) \\
&= ABC + AB\bar{C} + \bar{A}\bar{B} + BC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} \\
&= (ABC + BC) + (\bar{A}\bar{B} + \bar{A}\bar{B}\bar{C}) + (AB\bar{C} + A\bar{B}\bar{C}) \\
&= BC + \bar{A}\bar{B} + A\bar{C}
\end{aligned}
$$

第二章 逻辑代数（32）

- 2.2 公式法化简逻辑函数
——小结

1) 与或表达式化简：利用基本公式化简
2) 其他形式逻辑函数化简：转换为“与或”表达式，再化简

作业

- 2.6（1）～（6）

2.6 用公式法将下列函数化简为最简“与或”式。

(1) $F=\overline{A}\,\overline{B}+(AB+A\overline{B}+\overline{A}B)C$

(2) $F=(X+Y)Z+\overline{X}\,\overline{Y}W+ZW$

(3) $F=AB+\overline{A}C+\overline{B}\,\overline{C}$

(4) $F=AB+\overline{A}\,\overline{B}C+BC$

(5) $F=\overline{A}B+\overline{A}C+\overline{B}\,\overline{C}+AD$

(6) $F=\overline{A}\,\overline{B}+\overline{A}\,\overline{C}D+AC+B\overline{C}$

(7) $F=AC+\overline{A}\,\overline{B}+\overline{B}\,\overline{C}\,\overline{D}+BE\overline{C}+DE\overline{C}$

(8) $F=A(B+\overline{C})+\overline{A}(\overline{B}+C)+BCD+\overline{B}\,\overline{C}D$

(9) $F=\overline{X}\,\overline{Y}+(X+Y)Z$

(10) $F=(X+Y+Z+\overline{W})(V+X)(\overline{V}+Y+Z+\overline{W})$