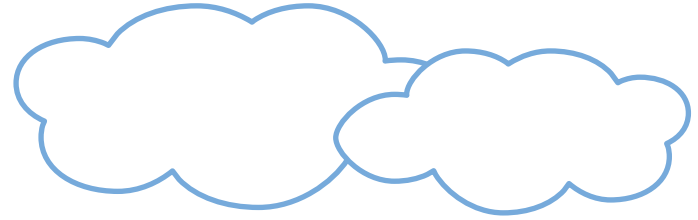




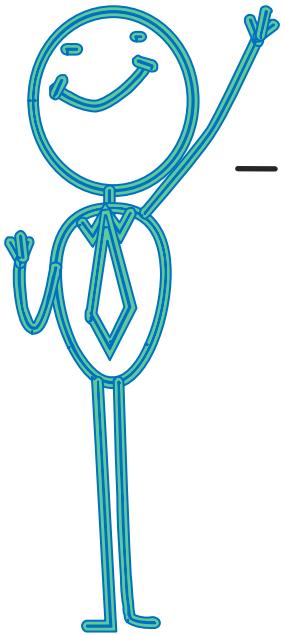
第三章 逻辑代数

第四章 组合逻辑电路

- 一、小规模组合逻辑电路初探（ 4.1 ）
- 二、 逻辑代数（3）
- 三、小规模组合逻辑电路分析和设计（ 4.1 ）
- 四、常用组合逻辑电路芯片（4.2）



三、小规模组合逻辑电路分析和设计





第四章 作业

4.4 从波形进行组合电路设计

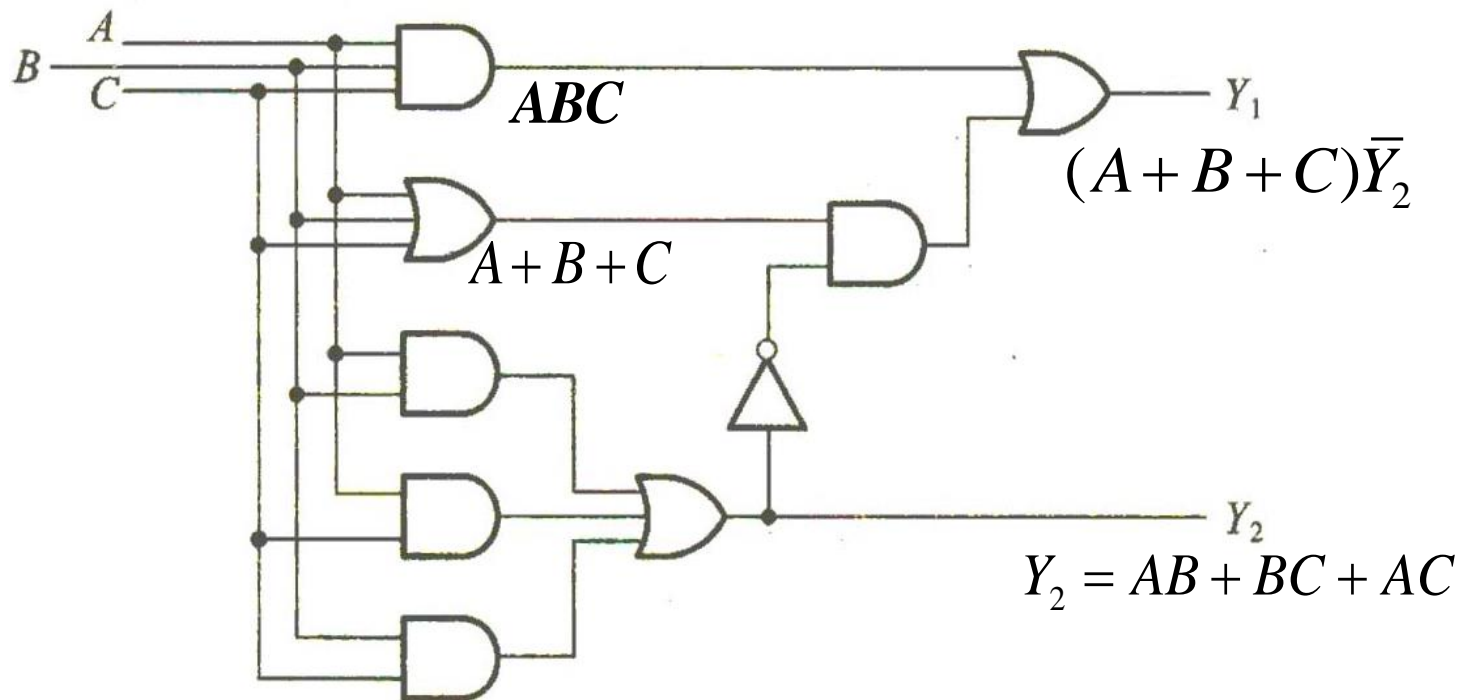
4.7 实际问题使用与非门组合电路设计

4.14 译码器设计

4.19 数选器设计实际问题



分析下面电路的逻辑功能



1) 逐级写出逻辑关系表达式

$$Y_1 = ABC + (A+B+C)\overline{(AB+BC+AC)}$$

2) 由逻辑关系表达式列出输入输出状态表

用逻辑代数化简



2) 由逻辑关系表达式列出输入输出状态表

$$Y_2 = AB + BC + AC$$

$$Y_1 = ABC + (A + B + C)(\overline{AB + BC + AC})$$

用逻辑代
数化简

$$= ABC + (A + B + C)(\overline{AB} \overline{BC} \overline{AC})$$

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

$$= ABC + (A + B + C)(\overline{AC} + \overline{AB} + \overline{BC})$$

$$= ABC + \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}B\overline{C}$$

是否还可以进一步化简?

请同学们用卡诺图法检验



$$Y_1 = ABC + \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}\overline{B}C \quad Y_2 = AB + BC + AC$$

真值表

<i>A</i>	<i>B</i>	<i>C</i>	<i>Y</i> ₂	<i>Y</i> ₁
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

3) 分析逻辑关系

输出为二进制数

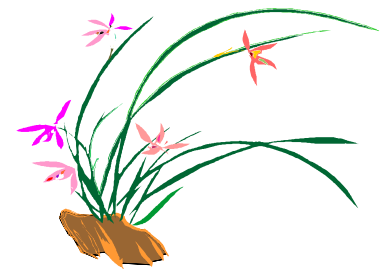
A+*B*+*C*的和

*Y*₂为高位, *Y*₁ 为低位



小规模组合逻辑电路分析一般步骤

1. 写出电路的逻辑表达式，运用逻辑代数化简
2. 列出电路的真值表
3. 分析电路的逻辑功能





组合逻辑电路设计

设计一个三人表决电路。每人有一按键，如同意按下键，用逻辑1表示；如不同意不按键，用逻辑0表示。表决结果用指示灯表示，多数同意灯亮，用逻辑1表示；反之灯不亮为逻辑0

1) 列真值表

设输入为A、B、C，输出为Y

2) 写逻辑表达式

取 Y=“1” 列逻辑式

$$Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

最小项之和的形式

输入			输出
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



最小项之和的逻辑表达式

$$Y = \bar{A}BC + A\bar{B}C + A\bar{B}\bar{C} + ABC$$

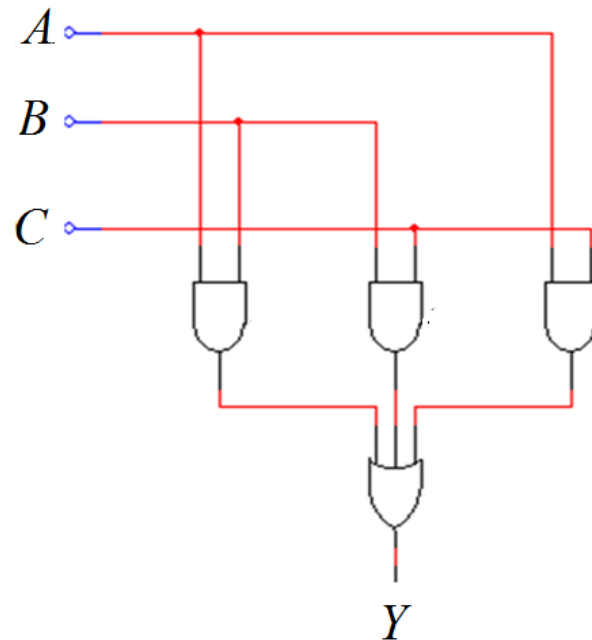
化简(用卡诺图)

A \ BC	00	01	11	10
0			1	
1		1	1	1

最简与或逻辑表达式

$$Y = AB + BC + AC$$

3) 画出逻辑电路图

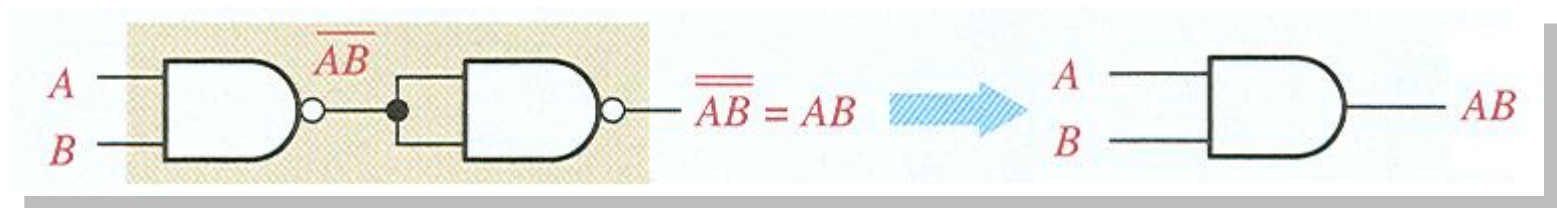




➤ NAND Gate as an Inverter



➤ Two NAND Gates as an AND Gate

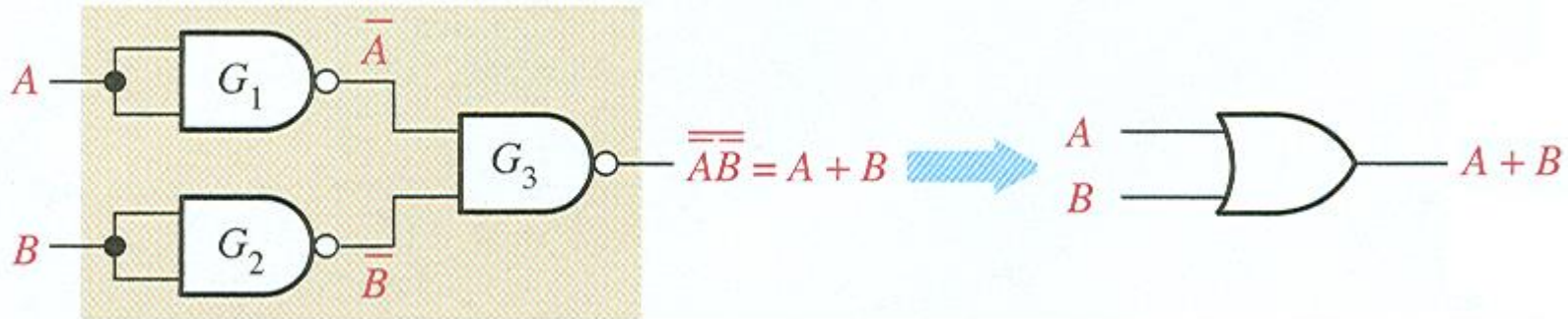


$$AB = \overline{\overline{AB}}$$

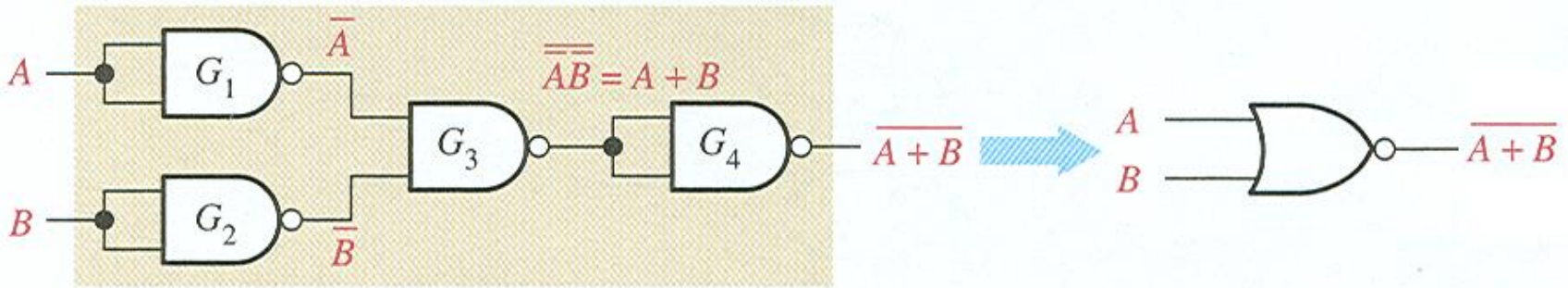


➤ Three NAND Gates as an OR Gate

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



➤ Four NAND Gates as NOR Gate





最简与或逻辑表达式

$$Y = AB + BC + AC$$

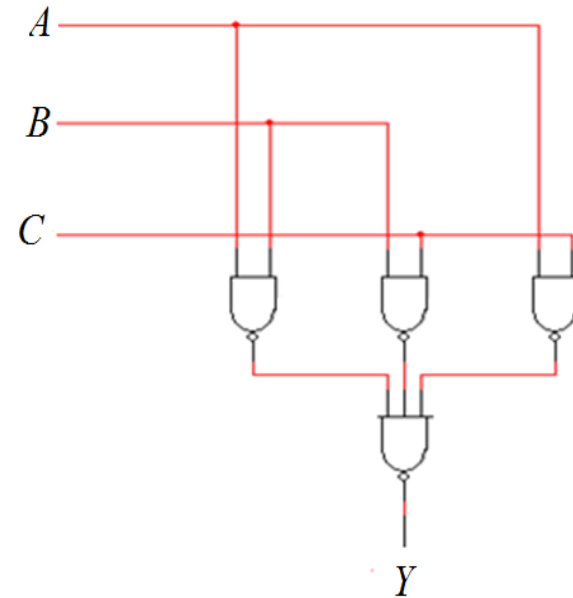
要求用“与非”门构成逻辑电路？

$$\begin{aligned} Y &= \overline{\overline{AB + BC + AC}} \\ &= \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}} \end{aligned}$$

最简与非逻辑表达式

将最简与或表达式转化成最简与非表达式的方法：

取双非后用反演定理
化成与非与非表达式！



- 最简逻辑电路标准
 - 门的个数最少
 - 每个门的连线最少
 - 门的种类最少



例：有一T形走廊，在相会处有一路灯，在进入走廊的A、B、C三地各有控制开关，都能独立进行控制。任意闭合一个开关，灯亮；任意闭合两个开关，灯灭；三个开关同时闭合，灯亮。设A、B、C代表三个开关(输入变量)；Y 代表灯(输出变量)。

(1) 逻辑抽象，列真值表

设：开关闭合其状态为 1，
断开为 0。

灯亮状态为 1，灯灭为0。

<i>A</i>	<i>B</i>	<i>C</i>	<i>Y</i>
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

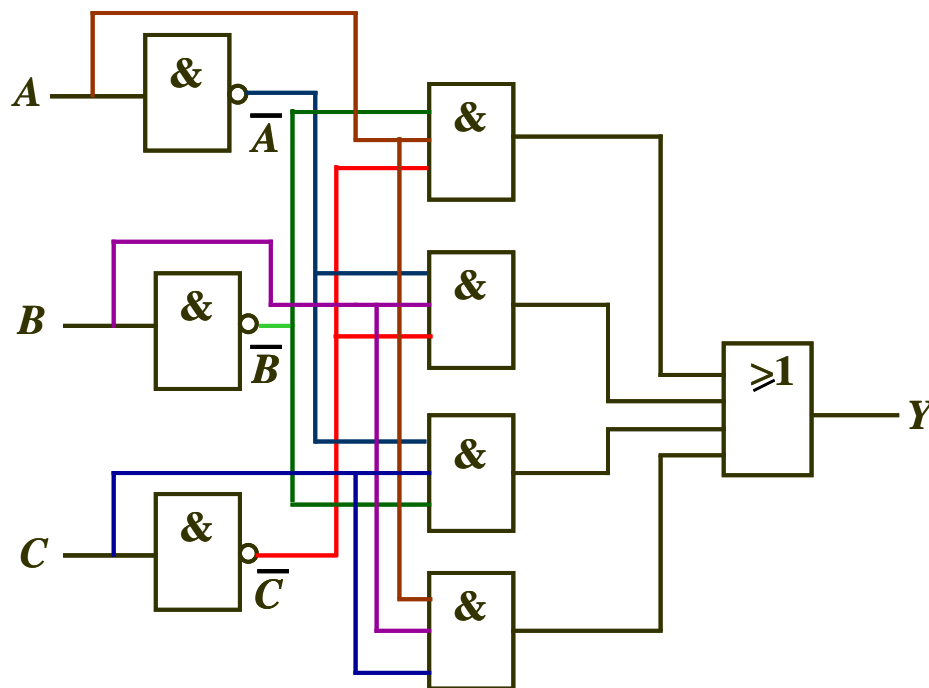
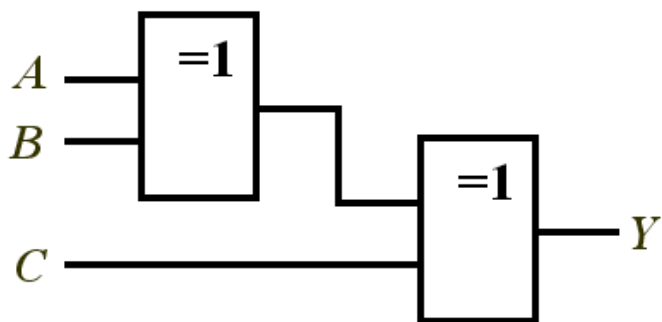


(2) 写出简化的逻辑表达式

$$Y = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC$$
$$= A \oplus B \oplus C$$

<i>A</i>	<i>B</i>	<i>C</i>	<i>Y</i>
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(3) 画出逻辑图





例：军民联欢会，军人持红票入场，群众持黄票入场，持绿票的军民均可以入场，符合要求时可以入场。求放行的逻辑关系。

解：

$A=1$ 为军人， $A=0$ 为群众

$BC=00$ 无票， $BC=01$ 黄票

$BC=10$ 红票， $BC=11$ 绿票

$Y=1$ 可入场， $Y=0$ 不准入场

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

(1) 根据逻辑要求列状态表



(2) 由状态表写出逻辑式化简可得:

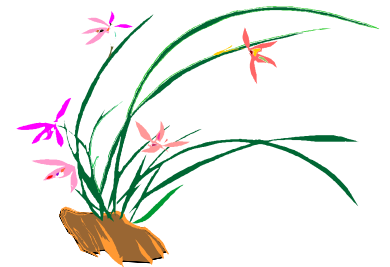
$$\begin{aligned} Y &= \overline{A}\overline{B}C + \overline{A}BC + AB\overline{C} + ABC \\ &= \overline{A}C(\overline{B} + B) + AB(\overline{C} + C) \\ &= \overline{A}C + AB \end{aligned}$$

(3) 画逻辑图



小规模组合逻辑电路设计一般步骤

1. 从已知逻辑功能要求出发，列写出真值表
2. 写出逻辑表达式，并变换成所需形式
3. 画出逻辑电路图





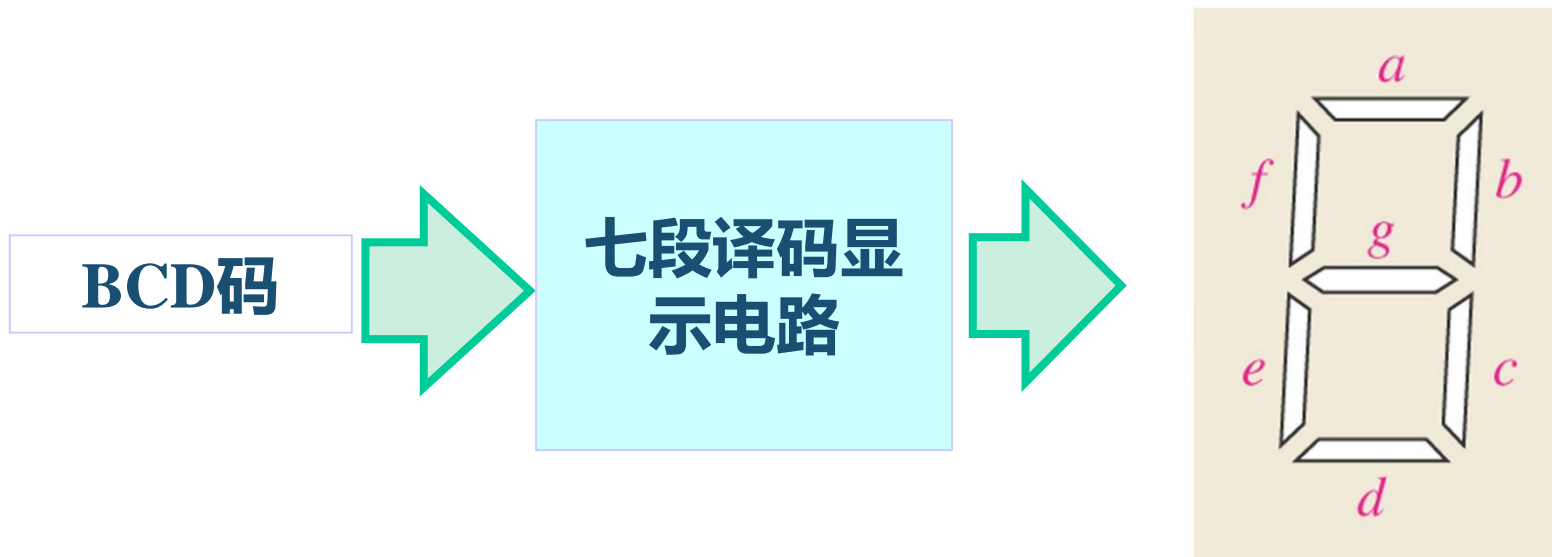
复习：逻辑关系的表达方式

1. 真值表 (逻辑状态表)
2. 卡诺图
3. 逻辑表达式
4. 逻辑电路图
5. 波形图



综合应用设计：七段数码管译码显示电路的设计

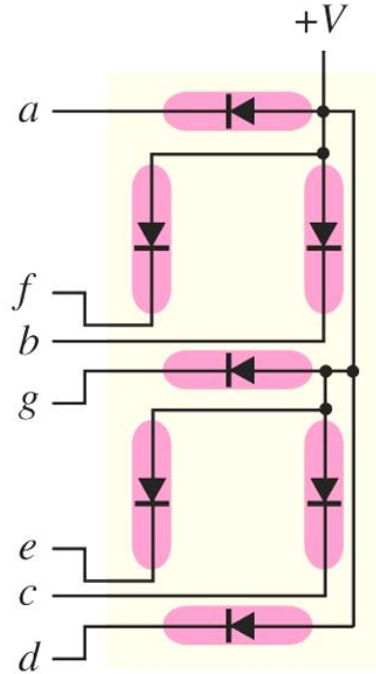
设计一个电路将用BCD码表示的数字通过七段数码管以十进制数值形式显示出来





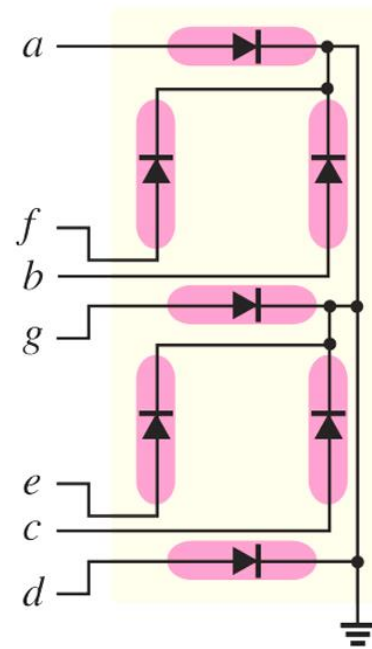
常用的显示器件---七段数码管:

半导体数码管、液晶数码管、荧光数码管



(a) Common-anode

共阳极接法



(b) Common-cathode

共阴极接法

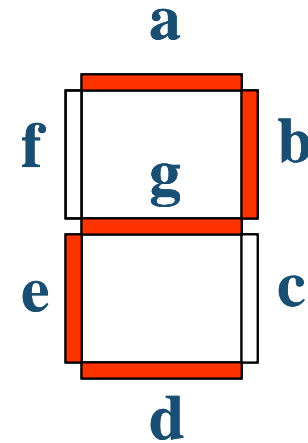


例：共阴极接法

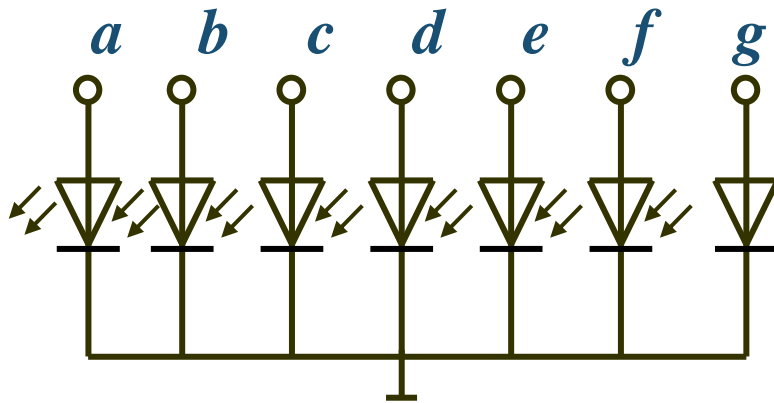
a b c d e f g

0 1 1 0 0 0 0

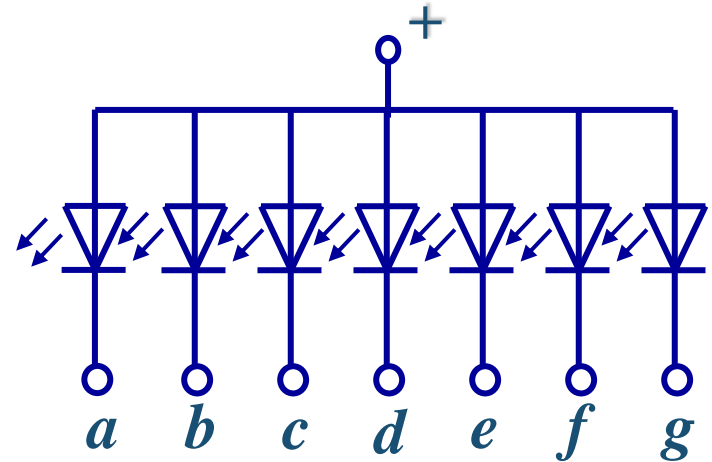
1 1 0 1 1 0 1



高电平时发光



共阴极接法

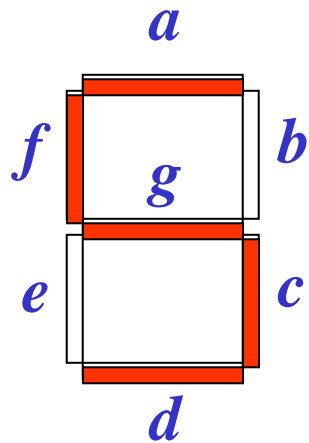


共阳极接法

低电平时发光



七段译码显示电路的真值表



输入				输出							显示
<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

其它输出:无关项