

第一章 数制和码制

1. 进制之间的转换掌握:

1. 十进制到任意进制的转换

2. 任意进制转十进制

 整数部分的转换

 小数部分的转换

 完整转换过程

 示例：将二进制数 1010.101 转换为十进制

3. 二进制与十六进制

从二进制转换到十六进制

 示例：将二进制 1101011101.0111 转换为十六进制

从十六进制转换到二进制

 示例：将十六进制 35D.7 转换为二进制

4. 二进制与8进制之间的转换

2. 二进制运算掌握:

1. 原反补的概念，补码表示

一. 机器数和真值

 1、机器数

 2、真值

二. 原码, 反码, 补码的基础概念和计算方法.

 1. 原码

 2. 反码

 3. 补码

2.BCD码的定义，格雷码的特点

第二章 逻辑代数基础

1. 基本运算、常用运算有哪些

 复杂逻辑运算的图形表示

2. 基本公式、常用公式

1. 基本公式、常用公式

3. 逻辑函数的最小项表达式

4. 反演定理(摩根定理)对偶式的求法

 1. 代入定理

 2. 反演定理

 3. 对偶定理

5. 逻辑函数之间的转换

6. 逻辑函数的化简(公式化简法、卡诺图化简法、带有无关项的逻辑函数的化简)

第三章 门电路

1. 正负逻辑的概念

2. OC门、OD门、TG门、三态门的特点

 1、OC门

 2、三态门

 3、传输门

 4、cmos门电路和TTL相比的特点

3. 二极管门电路(与门、或门的结构)

第四章 组合逻辑电路

1. 组合逻辑电路的分析(能够通过波形图判断电路的功能)

 组合电路逻辑功能表示方法

 组合逻辑电路分析方法

2. 掌握组合逻辑电路的设计步骤，门级电路的设计与MSI设计

 组合逻辑电路设计方法

3. 常用组合逻辑模块

 (1)优先编码器的特点，给定输入信号能判断出输出信号，以及根据输出端的状态，判断芯片工作状态

 74HC148优先编码器：(二进制)

 74HC147优先编码器：(二-十进制)

(2)译码器 正常译码的情况下，给定输入信号能够判断出输出端的状态

1.二进制译码器

74HC138译码器

2.74HC42 (二-十进制)

3.BCD-七段显示译码器

(3)数据选择器 给定一个数据选择器能分析出地址输入端的个数，掌握数据选择器的功能

(4)加法器 半加器的组成，全加器的本位和与进位的逻辑表达式，全加器的真值表，全加器的波形图

第五章 半导体存储电路

1.掌握触发器的特点 掌握各种逻辑功能的触发器的电路结构与动作特点，掌握触发器输出波形的画法即给出输入信号与时钟信号，能够根据触发器的动作特点画出输出端Q的波形图

一、触发器的两个基本特点

二、触发器的分类

(一)按电路结构形式不同划分

(二)按逻辑功能划分

(三)按存储数据的原理不同可分为

三、基本RS触发器

(一)逻辑功能描述

(二)或非门构成

(三)基本RS触发器的特点

(四)状态转换图

四、同步触发器

(一)同步RS触发器

1. 电路结构与工作原理分析

2. 设有专门的异步置位输入端和异步复位输入端的同步RS-触发器触发器

3. 动作特点

(二)同步D触发器

1.逻辑图、特性表和特性方程

2.逻辑功能

(三)同步JK触发器

1.逻辑图、特性表和特性方程

(四)同步T触发器

1.逻辑图、特性表和特性方程

(五)同步触发器的空翻现象

五、主从触发器

(一)主从RS-触发器

1.电路结构与工作原理

2.动作特点

(二)主从D-触发器

(三)主从JK-触发器

六、边沿触发器

七、锁存器与触发器

2.掌握寄存器的基本概念、寄存器的组成

3.理解半导体存储器的特点，ROM和RAM的特点，ROM的分类，了解容量的计算方法，ROM芯片实现组合逻辑函数

一、半导体存储器的特点

二、ROM (只读存储器) 的特点及分类

ROM的分类：

三、RAM (随机存取存储器) 的特点

四、容量的计算方法

五、ROM芯片实现组合逻辑函数

第六章 时序逻辑电路

1.掌握时序逻辑电路的分析步骤

2.掌握一般设计方法以及置零法和置数法设计计数器

3.掌握同步时序逻辑的特点

4.掌握同步二进制计数器的连接规律，给出逻辑电路图，能够通过时序逻辑电路的分析方法画出同步二进制计数器的时序图

74LS161

74LS163

74LS191

74LS193

5.异步二进制计数器的连接规律，给出逻辑电路图，能够通过时序逻辑电路的分析方法画出异步二进制计数器的时序图

1、异步二进制计数器

3位异步二进制加法计数器

3位异步二进制减法计数器

2、异步十进制计数器

74LS290

第七章 脉冲波形的产生与整形

掌握施密特触发器、单稳态触发器、多谐振荡器的特点

施密特触发器

单稳态电路

多谐振荡电路

第八章 A/D和D/A转换器

掌握D/A转换基本原理和电路结构分类；掌握A/D转换的基本原理 采样、量化编码

D/A转换的基本原理

D/A转换器的电路结构分类

A/D转换的基本原理

A/D转换器的分类

第九章 总章

第一章 数制和码制

第二章 逻辑代数基础

第三章 门电路

第四章 组合逻辑电路

第五章 半导体存储电路

第六章 时序逻辑电路

第七章 脉冲波形的产生与整形

第八章 A/D和D/A转换器

版权声明：

第一章 数制和码制

1. 进制之间的转换掌握：

1. 十进制到任意进制的转换

十进制小数转换为任意进制的小数，需要分别转换整数部分和小数部分。

1. 整数部分的转换：

- 使用连续除法，将整数部分不断除以目标进制的基数，直到商为0，然后将得到的余数逆序排列，即为转换后的整数部分。

2. 小数部分的转换：

- 使用连续乘法，将小数部分乘以目标进制的基数，取整数部分作为该位的结果，然后再用新的小数部分重复此过程，直到小数部分为0或达到所需的精度为止。

具体步骤如下：

对于十进制小数

$$D = I + F$$

, 其中 I 是整数部分, F 是小数部分。

1. 转换整数部分 I :

- 设目标进制为 b , 将 I 不断除以 b , 记录下每次的余数 r_i 。
- 当商为0时, 停止计算, 将余数逆序排列得到转换后的整数部分。

2. 转换小数部分 F :

- 将 F 乘以 b , 得到乘积的整数部分作为该位的结果 d_1 。
- 取乘积的小数部分作为新的 F , 重复上述过程得到下一位的结果 d_2 。
- 继续此过程, 直到小数部分为0或达到所需的精度为止。

最终, 十进制小数 D 转换为任意进制的小数表示为: 整数部分的结果紧接着小数部分的结果。

例如, 将十进制小数10.625转换为二进制表示:

1. 转换整数部分10:

- $10 \div 2 = 5$ 余 0
- $5 \div 2 = 2$ 余 1
- $2 \div 2 = 1$ 余 0
- $1 \div 2 = 0$ 余 1
- 逆序排列余数得到二进制整数部分: 1010

2. 转换小数部分0.625:

- $0.625 \times 2 = 1.25$, 取整数部分1作为第一位, 新的小数部分为0.25
- $0.25 \times 2 = 0.5$, 取整数部分0作为第二位, 新的小数部分为0.5
- $0.5 \times 2 = 1.0$, 取整数部分1作为第三位, 新的小数部分为0.0
- 小数部分为0, 停止计算。
- 得到二进制小数部分: 101

因此, 十进制小数10.625转换为二进制表示为: 1010.101。

2.任意进制转十进制

将任意进制数转换为十进制数, 需要分别处理整数部分和小数部分。

整数部分的转换

对于任意进制数的整数部分, 其表示形式为:

$$a_n a_{n-1} \dots a_1 a_0$$

其中, a_i 是第 i 位的数字, n 是最高位的索引。整数部分的十进制值可以通过以下公式计算:

$$I = \sum_{i=0}^n a_i \times b^i$$

这里, b 是基数 (即进制数), a_i 是第 i 位的数字。

小数部分的转换

对于任意进制数的小数部分, 其表示形式为:

$$\cdot d_1 d_2 d_3 \dots$$

其中, d_j 是小数点后第 j 位的数字。小数部分的十进制值可以通过以下公式计算:

$$F = \sum_{j=1}^{\infty} d_j \times b^{-j}$$

这里, b^{-j} 表示 b 的负幂次, 即 $\frac{1}{b^j}$ 。

完整转换过程

将整数部分和小数部分相加，得到完整的十进制数：

$$D = I + F$$

示例：将二进制数 1010.101 转换为十进制

整数部分：

- $1 \times 2^3 = 8$
- $0 \times 2^2 = 0$
- $1 \times 2^1 = 2$
- $0 \times 2^0 = 0$
- $I = 8 + 0 + 2 + 0 = 10$

小数部分：

- $1 \times 2^{-1} = 0.5$
- $0 \times 2^{-2} = 0$
- $1 \times 2^{-3} = 0.125$
- $F = 0.5 + 0 + 0.125 = 0.625$

完整十进制数：

- $D = I + F = 10 + 0.625 = 10.625$

因此，二进制数 1010.101 对应的十进制数是 10.625。

3.二进制与十六进制

二进制与十六进制之间的相互转换是基于它们之间的直接关系：十六进制每一位对应二进制的四位。这是因为 $16 = 2^4$ ，所以每一位十六进制数可以直接转换为一个四位的二进制数。

从二进制转换到十六进制

要将二进制数转换为十六进制数，可以按照以下步骤操作：

1. **分组**：从二进制数的小数点（如果没有小数点，则从最右边开始）向左和向右分别分组，每组四位。如果最左边或最右边的组不足四位，则在前面或后面补零以形成完整的四位数。
2. **转换**：将每个四位二进制组转换为相应的十六进制数字。二进制数 0000 到 1111 分别对应十六进制的 0 到 F。
3. **组合**：将转换后的十六进制数字组合起来，得到最终的十六进制表示。

示例：将二进制 1101011101.0111 转换为十六进制

- 分组：11 | 0101 | 1101 | . | 0111
- 补零：0011 | 0101 | 1101 | . | 0111
- 转换：3 | 5 | D | . | 7
- 组合：35D.7

从十六进制转换到二进制

要将十六进制数转换为二进制数，可以按照以下步骤操作：

1. **分解**: 将每个十六进制数字分开。
2. **转换**: 将每个十六进制数字转换为相应的四位二进制数。十六进制的 0 到 F 分别对应二进制的 0000 到 1111。
3. **组合**: 将所有转换后的二进制组连接起来，得到最终的二进制表示。

示例：将十六进制 35D.7 转换为二进制

- 分解: 3 | 5 | D | . | 7
- 转换: 0011 | 0101 | 1101 | . | 0111
- 组合: 001101011101.0111

这样，我们就可以在二进制和十六进制之间进行有效的相互转换。

4.二进制与8进制之间的转换

1. 从二进制转换到8进制：

- 将二进制数按每3位一组（从低位开始）进行分组，不足3位的在高位补0。
- 将每组的二进制数转换为对应的8进制数（即 $2^2 \times a + 2^1 \times b + 2^0 \times c$, 其中a、b、c为每组的三位二进制数）。
- 把得到的每个8进制数拼接起来，得到最终的8进制数。

2. 从8进制转换到二进制：

- 将每个8进制数转换为对应的3位二进制数（即找到使得 $2^2 \times a + 2^1 \times b + 2^0 \times c$ 等于该8进制数的a、b、c）。
- 把得到的每个二进制数拼接起来，得到最终的二进制数。

例如，二进制数 101101 转换为8进制：

- 分组: 101 101 (从低位开始，每三位一组)
- 转换: 101 对应8进制的 5, 101 对应8进制的 5
- 结果: 101101 (二进制) = 55 (8进制)

再如，8进制数 55 转换为二进制：

- 转换: 5 对应二进制的 101
- 结果: 55 (8进制) = 101101 (二进制)

二进制	十进制	八进制	十六进制
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3

二进制	十进制	八进制	十六进制
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	10	8
1001	9	11	9
1010	10	12	A
1011	11	13	B
1100	12	14	C
1101	13	15	D
1110	14	16	E
1111	15	17	F

2.二进制运算掌握:

1.原反补的概念，补码表示

一. 机器数和真值

在学习原码, 反码和补码之前, 需要先了解机器数和真值的概念.

1、机器数

一个数在计算机中的二进制表示形式, 叫做这个数的机器数。机器数是带符号的, 在计算机用一个数的最高位存放符号, 正数为0, 负数为1。

比如, 十进制中的数 +3 , 计算机字长为8位, 转换成二进制就是00000011。如果是 -3 , 就是 10000011 。

那么, 这里的 00000011 和 10000011 就是机器数。

2、真值

因为第一位是符号位, 所以机器数的形式值就不等于真正的数值。例如上面的有符号数 10000011 , 其最高位1代表负, 其真正数值是 -3 而不是形式值131 (10000011转换成十进制等于131) 。所以, 为区别起见, 将带符号位的机器数对应的真正数值称为机器数的真值。

例: 0000 0001的真值 = +000 0001 = +1 , 1000 0001的真值 = -000 0001 = -1

二. 原码, 反码, 补码的基础概念和计算方法.

在探求为何机器要使用补码之前, 让我们先了解原码, 反码和补码的概念. 对于一个数, 计算机要使用一定的编码方式进行存储. 原码, 反码, 补码是机器存储一个具体数字的编码方式.

1. 原码

原码就是符号位加上真值的绝对值, 即用第一位表示符号, 其余位表示值. 比如如果是8位二进制:

[+1]原 = 0000 0001

[-1]原 = 1000 0001

第一位是符号位. 因为第一位是符号位, 所以8位二进制数的取值范围就是:

[1111 1111, 0111 1111]

即

[-127, 127]

原码是人脑最容易理解和计算的表示方式.

2. 反码

反码的表示方法是:

正数的反码是其本身

负数的反码是在其原码的基础上, 符号位不变, 其余各个位取反.

[+1] = [00000001]原 = [00000001]反

[-1] = [10000001]原 = [11111110]反

可见如果一个反码表示的是负数, 人脑无法直观的看出来它的数值. 通常要将其转换成原码再计算.

3. 补码

补码的表示方法是:

正数的补码就是其本身

负数的补码是在其原码的基础上, 符号位不变, 其余各位取反, 最后+1. (即在反码的基础上+1)

[+1] = [00000001]原 = [00000001]反 = [00000001]补

[-1] = [10000001]原 = [11111110]反 = [11111111]补

对于负数, 补码表示方式也是人脑无法直观看出其数值的. 通常也需要转换成原码在计算其数值.

用补码表示的运算结果最小值和最大值应该是[-128, 128].

但是由于0的特殊情况, 没有办法表示128, 所以补码的取值范围是[-128, 127].

2.BCD码的定义, 格雷码的特点

8421码又称BCD码, 是十进制代码中最常见的一种. 在这种编码方式中, 每一位二值代码的1都代表一个固定数值, 将每一位的1代表的十进制数加起来, 得到的结果就是它所代表的十进制数码. 由于代码中从左到右每一位的1分别表示8、4、2、1, 所以将这种代码称为8421码。每一位的1代表的十进制数成为这一位的权。8421码中每一位的权是固定不变的, 它属于恒权代码。

格雷码 (Gray Code) 又称循环码。**每一位的状态变化都按一定的顺序循环。** (如果从 0000开始, 最右边一位的状态按0110 顺序循环变化, 右边第二位的状态按00111100顺序循环变化, 右边第三位按00001111110000 顺序循环变化。) 可见, **自右向左, 每一位状态循环中连续的0、1数目增加一倍。**由于4位格雷码只有16个, 所以**最左边一位的状态只有半个循环, 即 0000000011111111**。按照上述原则, 我们就很容易得到更多位数的格雷码。

与普通的二进制代码相比, 格雷码的最大优点就在于当它按照表1.5.2的编码顺序依次变化时, **相邻两个代码之间只有一位发生变化。这样在代码转换的过程中就不会产生过度“噪声”**。而在普通二进制代码的转换过程中, 则有时会产生过渡噪声。例如, 第四行的二进制代码0011转换内第五行的0100过程中, 如果最右边一位的变化比其他两位的变化慢, 就会在一个极短的瞬间出现0101状态, 这个状态将成为转换过程中出现的噪声。而在第四行的格雷码0010向第五行的0110转换过程中则不会出现过渡噪声。这种过渡噪声在有些情况下甚至会影响电路的正常工作, 这时就必须采取措施加以避免。在第4.9节中我们还将进一步讨论这个问题。

十进制代码中的余3循环码就是取4位格雷码中的十个代码组成的, 它仍然具有格雷码的优点, 即两个相邻代码之间仅有一位不同。

第二章 逻辑代数基础

1. 基本运算、常用运算有哪些

逻辑代数基础基本运算包括**与运算、或运算和非运算**, 而常用运算则包括**与非运算、或非运算、异或运算和同或运算**。

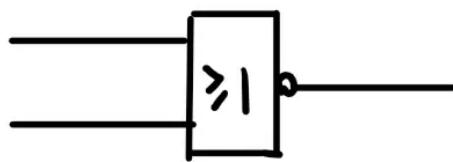
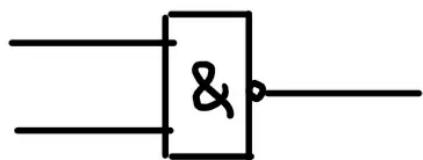
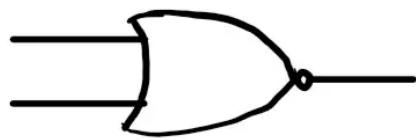
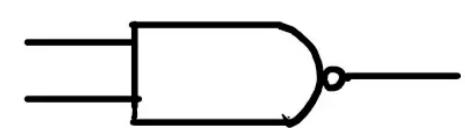
1. 基本运算

- **与运算 (AND)** : 用乘号或点 (\cdot 或 $A \wedge B$) 表示, 仅当所有输入均为1时, 输出才为1。例如, $A \cdot 1 = A$, $A \cdot 0 = 0$ 。
- **或运算 (OR)** : 用加号 (+ 或 \vee) 表示, 只要有一个输入为1, 输出就为1。例如, $A + 0 = A$, $A + A = A$ 。
- **非运算 (NOT)** : 用横线或撇号 ($\bar{}$ 或 \sim) 表示, 将输入的逻辑水平反转。例如, $\bar{0} = 1$, $\bar{1} = 0$ 。

2. 常用运算

- **与非运算 (NAND)** : 先进行与运算, 然后对结果取非。例如, $A \text{ NAND } B = \bar{(A \wedge B)}$ 。
- **或非运算 (NOR)** : 先进行或运算, 然后对结果取非。例如, $A \text{ NOR } B = \bar{(A \vee B)}$ 。
- **异或运算 (XOR)** : 当输入位不同时, 输出为1; 否则为0。例如, $A \text{ XOR } B = (A \wedge \neg B) \vee (\neg A \wedge B)$ 。
- **同或运算 (XNOR)** : 当输入位相同时, 输出为1; 否则为0。例如, $A \text{ XNOR } B = \neg(A \text{ XOR } B)$ 。

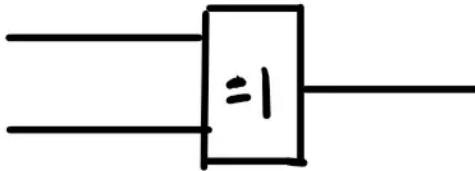
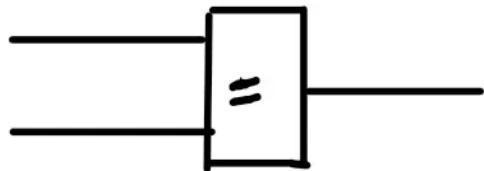
复杂逻辑运算的图形表示



与非

或非
知乎 @通大侠

图1.11 与非&或非



同或

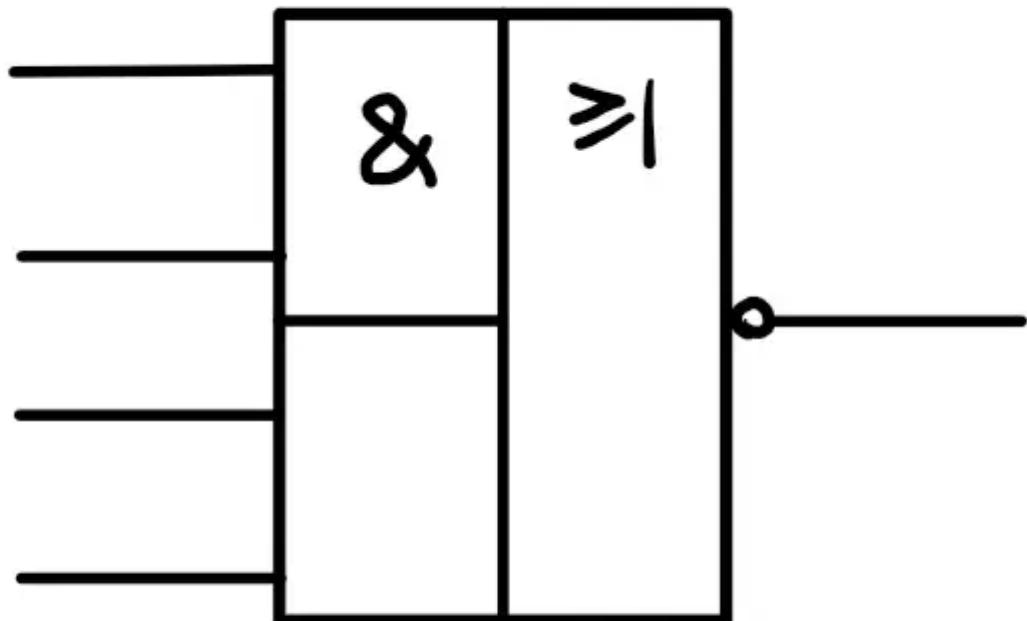
$Y = A \oplus B$

异或

$Y = A \oplus B$

知乎 @通大侠

图1.12 同或&异或



与或非

$$Y = (A \cdot B + C \cdot D)'$$

知乎 @通大侠

图1.13 与或非

2. 基本公式、常用公式

1. 基本公式、常用公式

类型	规则/公式	表达式
变量与常量运算	0与变量的与运算	$0 \cdot A = 0$
	1与变量的与运算	$1 \cdot A = A$
	0与变量的或运算	$0 + A = A$
	1与变量的或运算	$1 + A = 1$
同一变量运算 (重叠律)	变量的自我与运算	$A \cdot A = A$
	变量的自我或运算	$A + A = A$
变量与其反变量运算	变量与反变量的与运算	$A \cdot A' = 0$

类型	规则/公式	表达式
	变量与反变量的或运算	$A + A' = 1$
交换律	与运算的交换律	$A \cdot B = B \cdot A$
	或运算的交换律	$A + B = B + A$
结合律	与运算的结合律	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
	或运算的结合律	$A + (B + C) = (A + B) + C$
分配律	与运算分配到或运算上	$A \cdot (B + C) = A \cdot B + A \cdot C$
	或运算分配到与运算上	$A + B \cdot C = (A + B) \cdot (A + C)$
反演律	与运算的反演律	$(A \cdot B)' = A' + B'$
	或运算的反演律	$(A + B)' = A' \cdot B'$
还原律	一个变量经过两次求反运算之后还原为其本身	$(A')' = A$
0、1求反	0的反	$0' = 1$
	1的反	$1' = 0$
常用公式	在两个乘积项相加时，若其中一项以另一项为因子，则该项是多余的，可以删去	$A + A \cdot B = A$
	两个乘积项相加时，如果一项取反后是另一项的因子，则此该项是多余的，可以消去	$A + A' \cdot B = A + B$
	当两个乘积项相加时，若他们分别包含B和B'两个因子而其他因子相同，则两项定能合并，且可将B和B'两个因子消去	$A \cdot B + A \cdot B' = A$
	变量A和包含A'的和相乘时，其结果等于A，即可以将和消掉	$A \cdot (A + B) = A$
	若两个乘积项中分别包含A和A'两个因子，而这两个乘积项的其余因子组成第三个乘积项时，则第三个乘积项是多余的，可以消去	$A \cdot B + A' \cdot C + B \cdot C = A \cdot B + A' \cdot C$
	若两个乘积项中分别包含A和A'两个因子，而这两个乘积项的其余因子组成第三个乘积项时，则第三个乘积项是多余的，可以消去	$A \cdot B + A' \cdot C + BCD = A \cdot B + A' \cdot C$
	当A和一个乘积项的非相乘，且A为乘积项的因子时，则A这个因子可以消去	$A \cdot (A \cdot B)' = A \cdot B'$

类型	规则/公式	表达式
	当A和一个乘积项的非相乘，且A为乘积项的因子时，其结果就等于A'	$A' \cdot (AB)' = A'$

3.逻辑函数的最小项表达式

在n变量逻辑函数中，若m为包含n个因子的乘积项，而且这n个变量均以原变量或反变量的形式在m中出现一次，则称m为该组变量的最小项，n变量的最小项应有 2^n 个

- ABC (这代表三个变量都为0，即m0)
- AB'C (A和B为1，C为0，即m1)
- A'BC (A为0，B和C为1，即m2)
- A'B'C (A和B为0，C为1，即m3)
- ...
- A'B'C' (所有变量都为1，即m7)

如果把取值看作一个二进制数，那么它所表达的十进制数就是m的角标。

从最小项的定义出发可以证明它具有如下的重要性质：

- ① 在输入变量的任何取值下必有一个最小项，而且仅有一个最小项的值1。
- ② 全体最小项之和为1。
- ③ 任意两个最小项的乘积为0。
- ④ 具有相邻性的两个最小项之和可以合并成一项并消去一对因子。

若两个最小项只有一个因子不同，则称这两个最小项具有相邻性。例如，A'BC'和ABC'两个最小项仅第一个因子不同，所以它们具有相邻性。这两个最小项相加时定能合并成一项并将一对不同的因子消去

$$A'BC' + ABC' = (A'+A) BC' = BC'$$

4.反演定理(摩根定理)对偶式的求法

1.代入定理

定义：在任何一个包含变量A的逻辑等式中，若以另外一个逻辑式代入式中所有A的位置，则等式仍然成立。

对一个乘积项或逻辑式求反时，应在乘积项或逻辑式外边加括号，然后对括号内的整个内容求反

2.反演定理

定义：对于任意一个逻辑式Y，若将其中所有的“·”换成“+”，“+”换成“·”；0换成1，1换成0；原变量变为反变量，反变量变为原变量，则得到的结果就是Y'。

注意：

1. 遵循“先括号，然后乘，最后加”的运算优先次序。
2. 不属于单个变量上的反号（“ ”）应保留不变。

示例 2：

- 原等式： $Y = ((AB' + C)' + D)' + C$

- 求出 Y' :

$$\begin{aligned} \circ Y' &= (((A' + B)C')'D')'C' \\ \circ &= ((A'C' + BC') + D)C' \\ \circ &= A'C' + BC' + C'D \end{aligned}$$

3. 对偶定理

定义: 若两逻辑式相等, 则他们的对偶式也相等, 这叫做对偶定理。

对偶式: 对于任意一个逻辑式 Y , 若将其中所有的“.”换成“+”, “+”换成“.”; 0换成1, 1换成0, 则得到一个新的逻辑式 Y_D 。 Y_D 称为 Y 的对偶式。

若 $Y = A \cdot (B + C)Y = A \cdot (B + C)$, 则 $Y_D = A^- + (B^- \cdot C^-)Y^{**}D = A + (B \cdot C)$ 。

为了证明两个逻辑式相等, 也可以证明他们的对偶式相等来完成

5. 逻辑函数之间的转换

1. 逻辑表达式与真值表的转换

- **从逻辑表达式到真值表:** 可以通过将逻辑表达式的所有变量取值组合代入表达式, 计算得到输出值, 从而形成真值表。例如, 对于逻辑表达式 $F = A \text{ AND } B \text{ OR } C$, 可以列出A、B、C所有可能的取值组合, 并计算每一组取值对应的F的值。
- **从真值表到逻辑表达式:** 通过分析真值表中输出为1的行, 将这些行的输入变量组合成逻辑表达式。每个变量取值为1时用原变量表示, 取值为0时用反变量表示, 然后将这些乘积项用OR连接起来形成逻辑表达式。

2. 逻辑表达式与逻辑图的转换

- **从逻辑表达式到逻辑图:** 将逻辑表达式中的每个逻辑运算符用相应的符号表示, 并按照逻辑运算的先后顺序连接起来, 形成逻辑图。这个过程可能需要先对逻辑表达式进行化简。
- **从逻辑图到逻辑表达式:** 可以从逻辑图的输出端开始, 逐步追溯到输入端, 将每个逻辑符号用相应的逻辑运算符表示, 最终形成逻辑表达式。

3. 逻辑表达式与波形图的转换

- **从逻辑表达式到波形图:** 根据逻辑表达式, 列出所有变量的可能取值组合, 按照时间顺序画出输入变量和对应输出变量的波形图。高电平为1, 低电平为0。
- **从波形图到逻辑表达式:** 由于直接从波形图得出逻辑表达式较为困难, 通常需要先将波形图转换成真值表, 然后再由真值表转换成逻辑表达式。

4. 逻辑表达式与卡诺图的转换

- **从逻辑表达式到卡诺图:** 首先将逻辑函数表达式变换为最小项表达式, 然后在卡诺图上填入对应的最小项。填入1的方格对应函数值为1的最小项, 其余方格填入0或空格。
- **从卡诺图到逻辑表达式:** 通过卡诺图上的相邻最小项合并简化, 得到简化后的逻辑表达式。合并相邻为1的方格, 写出每个方格群对应的乘积项, 然后将这些乘积项相加, 形成简化的逻辑表达式。

5. 真值表与卡诺图的转换

- **从真值表到卡诺图:** 通过分析真值表中输出为1的行, 将这些行的输入变量组合对应的最小项在卡诺图上填入1, 其余填入0或空格。
- **从卡诺图到真值表:** 通过对卡诺图上的每个为1的方格, 列出其对应的输入变量取值组合, 形成真值表。

6.逻辑函数的化简(公式化简法、卡诺图化简法、带有无关项的逻辑函数的化简)

1. 公式化简法

- **并项法**: 这种方法利用公式 $AB+AB' = A$ 来消去多余的变量。例如，若有一个逻辑表达式 $Y = AB + AB'$ ，可以化简为 $Y = A$ 。
- **吸收法**: 使用公式 $A + AB = A$ 来消除多余的与项。例如，对于 $Y = A + ABC$ ，化简后仍为 $Y = A$ 。
- **消因子法**: 此方法通过公式 $A + A'B = A + B$ 来消去与项中多余的因子。这对于含混杂合变量的表达式特别有用。
- **配项法**: 利用 $A + A = A$ 和 $A + A' = 1$ 进行配项，以简化表达式。这种方法常用于调整表达式，使其适合其他化简方法。
- **综合应用**: 在实际的逻辑函数化简过程中，上述方法常常混合使用。选择哪种方法取决于表达式的具体形式和复杂度。通过反复练习和经验积累，工程师可以快速识别并应用最有效的化简策略。

2. 卡诺图化简法

- **构建卡诺图**: 首先，根据逻辑函数的变量数构建对应大小的卡诺图。每个方格代表一个最小项，逻辑相邻的最小项在几何位置上也相邻排列。
- **填入最小项**: 将逻辑函数的最小项填入卡诺图中。如果最小项存在，则在相应位置填入1，否则填入0。
- **化简规则**: 通过合并相邻的最小项来化简逻辑函数。两个相邻最小项合并可以消去一个变量，四个相邻最小项合并可以消去两个变量，以此类推。合并时，应确保圈内的最小项尽可能多，且每个圈的最小项数量符合2的幂次方。
- **化简步骤**: 画出函数的卡诺图，按照规则画圈。先圈孤立的1，再圈符合条件的最小项组合。最后，从每个圈写出化简后的乘积项，将这些乘积项相加形成最简逻辑表达式。

3. 带有无关项的逻辑函数化简

- **无关项定义**: 无关项是指在特定输入下，函数值可以为任意值（0或1）的最小项。它们分为约束项和任意项。约束项是逻辑上受限必须为0的项，而任意项则对函数值无影响。
- **化简中的应用**: 在化简逻辑函数时，可以灵活地利用无关项。通常，加入或去除无关项的目的是使化简后的表达式包含最少的项和因子。通过合理地利用无关项，可以得到更简单的化简结果。
- **实例分析**: 以一个实际例子说明，假设有一个电机控制逻辑函数，其中某些输入组合是不允许的，这些即为约束项。在化简时，这些约束项可以视为0处理，从而简化表达式。

二变量最小项卡诺图

	B
A	
0	0
1	1

$A' B'$	$A' B$
m_0	m_1
$A B'$	$A B$
m_2	m_3

三变量最小项卡诺图

	BC
A	
0	00
1	01
0	11
1	10

0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

四变量最小项卡诺图

	CD
AB	
00	00
01	01
11	11
10	10

00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

五变量最小项卡诺图

	CDE
AB	
00	000
01	001
11	011
10	010
00	110
01	111
11	101
10	100

00	m_0	m_1	m_3	m_2	m_6	m_7	m_5	m_4
01	m_8	m_9	m_{11}	m_{10}	m_{14}	m_{15}	m_{13}	m_{12}
11	m_{24}	m_{25}	m_{27}	m_{26}	m_{30}	m_{31}	m_{29}	m_{28}
10	m_{16}	m_{17}	m_{19}	m_{18}	m_{22}	m_{23}	m_{21}	m_{20}

为了保证图中几何位置相邻的最小项在逻辑上也具有相邻性,这些数码不能按自然二进制数从小到大地顺序排列,而必须按图中的方式排列,以确保相邻的两个最小项仅有一个变量是不同的。(即:循环码,又称格雷码)。

1.最简与或表达式

式中乘积最少

乘积项中含的变量最少

2.最简与非--与非

在最简与或表达式的基础上两次取反

用摩根定理去掉内层非号

3.最简或与

求出反函数的最简与或表达式

利用反演规则写出最简或与

4.或非--或非

求最简或与

两次取反

用摩根定理去掉内层非号

第三章 门电路

1.正负逻辑的概念

- **正逻辑:** 在正逻辑体系中,高电平被定义为逻辑1,低电平被定义为逻辑0 $[^1^0]$ 。这是数字电路中最常用的一种约定,通常在没有特别说明的情况下,默认采用正逻辑。
- **负逻辑:** 与正逻辑相反,负逻辑将高电平定义为逻辑0,低电平定义为逻辑1 $[^0^1]$ 。这种逻辑体系在某些特定场合下使用,例如在错误检测和安全控制电路中。

2.OC门、OD门、TG门、三态门的特点

1. OC门

集电极开路的门电路,自己加上拉电阻和电源可增加驱动能力

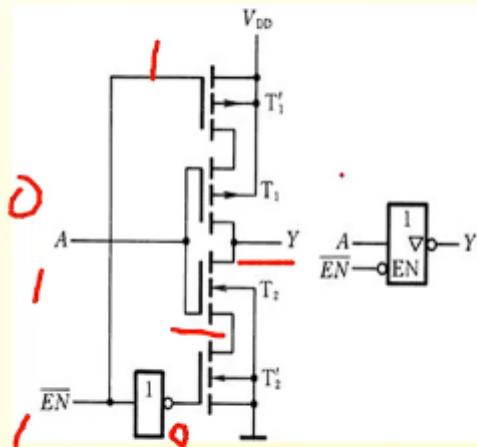
OC实现“线与”逻辑。

OD门和OC门原理一样,一个是三极管一个是MOS管。

2、三态门

(二) 三态输出门

介绍：具有三种状态的门电路，分别为高电平，低电平，高阻态。



当 $\overline{EN} = 0$ 时， $Y = A$ ，——
三态门处于工作状态；

当 $\overline{EN} = 1$ 时，三态门输出呈现高阻态，又称禁止态。

高阻态：就好像输入引脚悬空，对后级无影响，它的状态由后级电路的状态决定。

CSDN @涛涛呐 ~

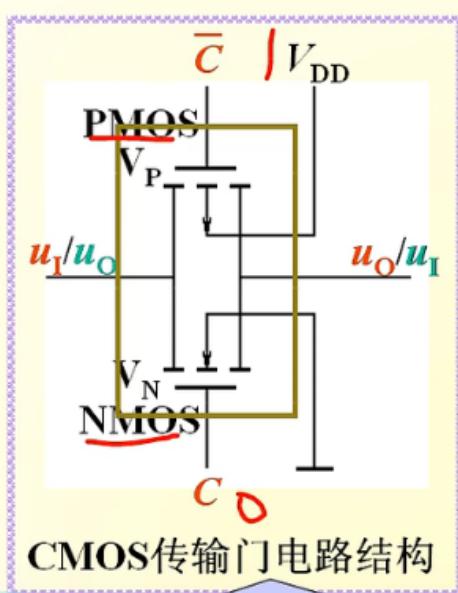
使能端为低电平，则上下两个MOS管导通，所以T1上接通高电平，T2接通低电平，此时输出根据输入A取反输出。A=0, T1导通, Y=1; A=1, T2导通, Y=0。

使能端为高电平，后级电路上下两个MOS管都不导通，则此时输出端与输入端没有联系，输出端呈高阻态。

3、传输门

(三) CMOS 传输门

工作原理



当 $C = V_{DD}$, $u_I = 0 \sim V_{DD}$ 时， V_N 、 V_P 中至少有一管导通，输出与输入 $u_O = u_I$ ，称传输门开通。

当 $C = 0V$, $u_I = 0 \sim V_{DD}$ 时， V_N 、 V_P 均截止，输出与输入之间呈现高电阻，相当于开关断开。

u_I 不能传输到输出端，称传输门关闭。

$C = 1$, $\overline{C} = 0$ 时，传输门开通， $u_O = u_I$ ；

$C = 0$, $\overline{C} = 1$ 时，传输门关闭，信号不能传输。

2.0x CSDN @涛涛呐 ~

4. cmos门电路和TTL相比的特点

CMOS 门电路比之 TTL 的主要特点

1、功耗极低

2、抗干扰能力强

3、电源电压范围宽

4、输出信号摆幅大($U_{OH} \approx V_{DD}$, $U_{OL} \approx 0V$)
(对于TTL门低电平大约为0.3V)

5、输入阻抗高

6、扇出系数大(带负载能力强)

7、对于一般情况下，TTL门比CMOS门电平翻转快一些

CSDN @涛涛响 ~

闲置输入端的处理

与门和与非门 多余输入端接正电源或与有用输入端并接

或门和或非门 多余输入端接地或与有用输入端并接

CMOS
电路多余输入端与有用输入端的并接仅适用于工作频率很低的场合。

TTL 电路输入端悬空时相当于输入高电平，
CMOS 电路多余输入端不允许悬空。

CSDN @涛涛响 ~

3. 二极管门电路(与门、或门的结构)

1. 二极管与门

- 结构特点：二极管与门由两个二极管和一个上拉电阻组成。A、B是输入端，Y为输出端。当所有输入端均为高电平时，输出端也为高电平；而只要有一个输入端为低电平，输出即为低电平[^1^]。

- **工作原理**: 当A或者B有一个输入低电平0V时，对应的二极管导通，输出端Y通过导通的二极管被拉低到0.7V，被视为低电平。当A和B都输入高电平3V时，两个二极管均导通，但由于每个二极管的导通压降为0.7V，所以输出端的电压为 $3 - 0.7 \times 2 = 1.6V$ ，仍被识别为低电平。只有当所有输入端同时为高电平时（考虑上拉电阻提供的电压），输出才为高电平^[^2^]。
- **优缺点**: 二极管与门结构简单，但存在输出电平不稳定的问题。由于每经过一个二极管就会降低约0.7V，多个级联的与门会导致逻辑电平偏移，影响电路的稳定性^[^1^]。并且这种门电路在多级级联使用时需要特别设计电源电压和电阻参数，以确保各级输出的正确性。

2. 二极管或门

- **结构特点**: 二极管或门由两个二极管和一个下拉电阻组成。A、B是输入端，Y为输出端。当任一输入端为高电平时，输出即为高电平；所有输入端均为低电平时，输出才为低电平^[^3^]。
- **工作原理**: 当输入端Vi1或Vi2中有任何一个为5V（逻辑1）时，其对应的二极管导通，输出电压VO为 $5V - 0.7V = 4.3V$ ，即输出为高电平。当输入端Vi1和Vi2都为0V时，二极管均不导通，由下拉电阻将输出拉低至低电平^[^4^]。
- **优缺点**: 二极管或门能够有效实现逻辑“或”功能，但其功耗较大。因为每当有高电平输入导通二极管时，都会有较大的电流通过电阻，导致功耗增加^[^3^]。虽然可以通过增大电阻来减少功耗，但这会显著增加输出电阻，影响电路的驱动能力和级联使用性能。

第四章 组合逻辑电路

1. 组合逻辑电路的分析(能够通过波形图判断电路的功能)

组合电路逻辑功能表示方法

- 真值表，卡诺图，逻辑表达式，时间图(波形图)

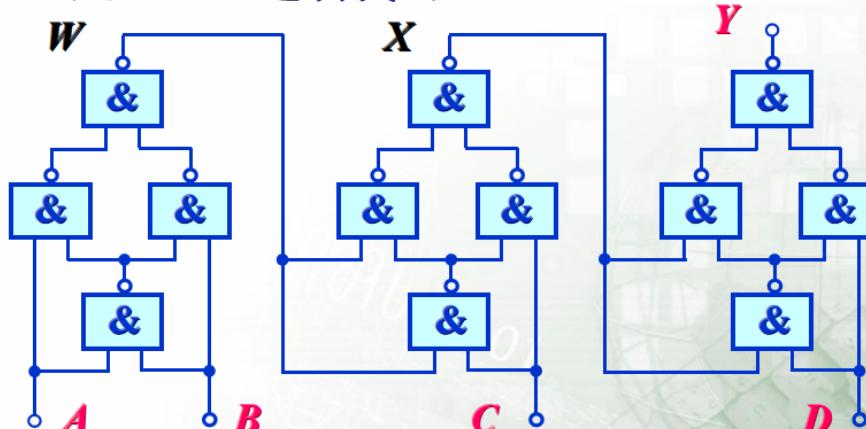
组合逻辑电路分析方法

1. 分析步骤



2. 实例分析

[例 3.1.1] 分析图中所示电路的逻辑功能，输入信号 A 、 B 、 C 、 D 是一组二进制代码。



[解] 1. 逐级写输出函数的逻辑表达式

$$W = \overline{\overline{A} \overline{AB}} \quad \overline{\overline{AB} B}$$

$$X = \overline{W} \overline{WC} \quad \overline{WC} C$$

https://blog.csdn.net/qq_43923155

2. 化简 $W = \overline{\overline{A} \overline{AB}} \quad \overline{\overline{AB} B} = \overline{AB} + \overline{AB}$

$$X = W \overline{C} + \overline{W} C = \overline{AB} \overline{C} + \overline{ABC} + \overline{A} \overline{B} C + ABC$$

$$Y = \overline{X} \overline{D} + \overline{X} D = \overline{AB} \overline{C} \overline{D} + \overline{ABC} \overline{D} + \overline{A} \overline{B} C \overline{D} + ABC \overline{D} \\ + \overline{AB} C D + \overline{ABC} D + \overline{ABC} D + ABC \overline{D}$$

https://blog.csdn.net/qq_43923155

3. 列真值表

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

4. 功能说明：当输入四位代码中 1 的个数为奇数时输出为 1，为偶数时输出为 0 — 检奇电路

https://blog.csdn.net/qq_43923155

补充例1 分析图P3.1电路的逻辑功能，写出 Y_1 、 Y_2 的逻辑函数式，列出真值表，指出电路完成什么逻辑功能。

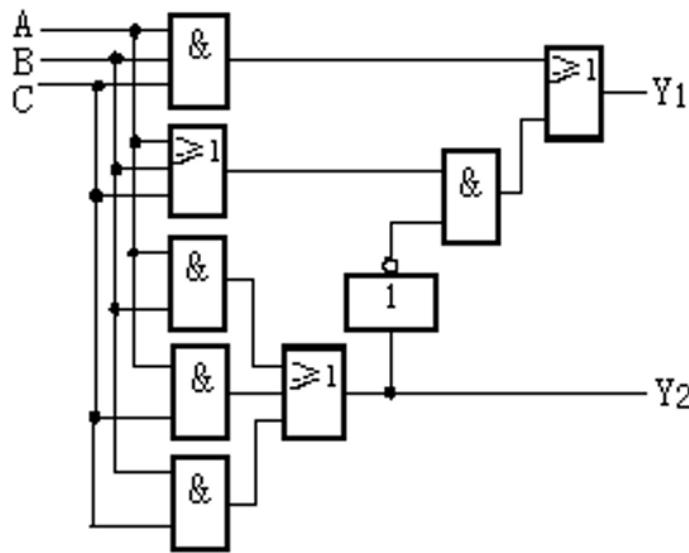


图 P3.1

https://blog.csdn.net/qq_43923155

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

$$Y_2 = AB + AC + BC$$

A	B	C	$Y_1 Y_2$
0	0	0	0 0
0	0	1	1 0
0	1	0	1 0
0	1	1	0 1
1	0	0	1 0
1	0	1	0 1
1	1	0	0 1
1	1	1	1 1

https://blog.csdn.net/qq_43923155

由真值表可知：电路构成全加器，输入A、B、C为加数、被加数和低位的进位， Y_1 为“和”， Y_2 为“进位”。

补充例2 已知逻辑电路如图P3.2 所示，试分析其逻辑功能。

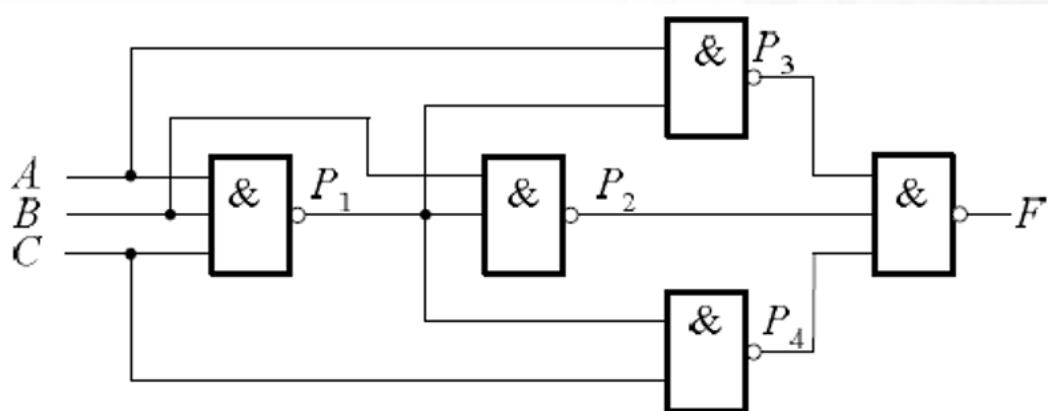


图 P3.2

https://blog.csdn.net/qq_43923155

[解]: $P_1 = \overline{ABC}$

$$P_2 = \overline{BP_1} = \overline{\overline{B}\overline{ABC}}$$

$$P_3 = \overline{AP_1} = \overline{\overline{A}\overline{ABC}}$$

$$P_4 = \overline{CP_1} = \overline{\overline{C}\overline{ABC}}$$

$$F = \overline{P_2 P_3 P_4}$$

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

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

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

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

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

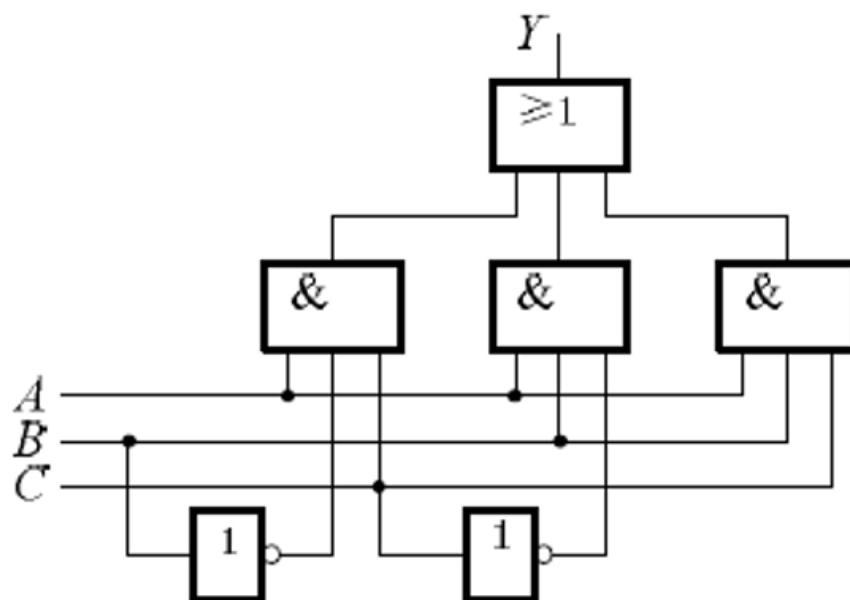
真值表:

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

从真值表看出, $ABC=000$ 或 $ABC=111$ 时, $F=0$, 而 A 、 B 、 C 取值不完全相同时, $F=1$ 。
故这种电路称为“不一致”电路。

补充例3 某组合逻辑电路如图P3.3 所示：

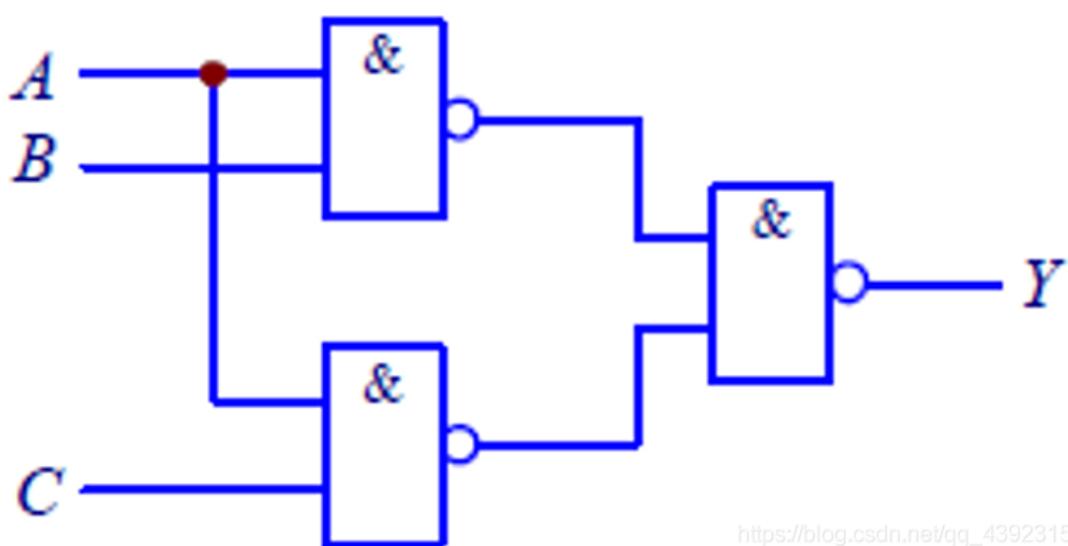
- (1) 写出函数Y的逻辑表达式；
- (2) 将函数Y化为最简与-或式；
- (3) 用与非门画出其简化后的电路。



P3.3

解：
$$Y = \overline{A}\overline{B}C + A\overline{B}\overline{C} + ABC = AB + AC$$

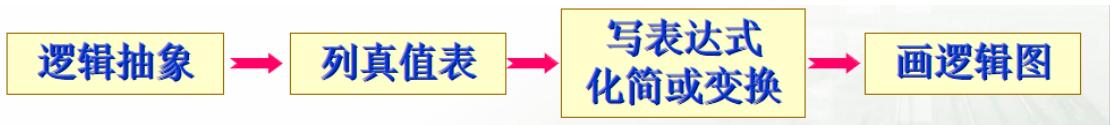
$$Y = \overline{\overline{AB} + AC} = \overline{\overline{AB}} \cdot \overline{\overline{AC}}$$



2. 掌握组合逻辑电路的设计步骤，门级电路的设计与MSI设计

组合逻辑电路设计方法

1. 设计步骤



2. 实例分析

[例 3.1.2] 设计一个表决电路，要求输出信号的电平与三个输入信号中的多数电平一致。

[解] 1. 逻辑抽象

(1) 设定变量： 输入 A 、 B 、 C ， 输出 Y

(2) 状态赋值：

$A = B = C = 0$ 表示 输入信号为低电平

$A = B = C = 1$ 表示 输入信号为高电平

$Y = 0$ 表示 输入信号中多数为低电平

$Y = 1$ 表示 输入信号中多数为高电平

2. 列真值表

3. 写输出表达式并化简

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

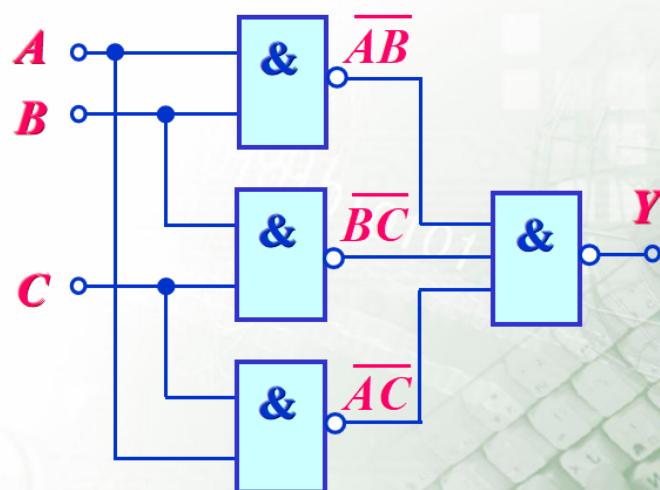
最简与或式 → 最简与非-与非式

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

A	B	C	Y
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

4. 画逻辑图 — 用与门和或门实现 $Y = BC + AC + AB$

— 用与非门实现 $Y = \overline{\overline{BC}} \cdot \overline{\overline{AC}} \cdot \overline{\overline{AB}}$



[例] 设计一个监视交通信号灯工作状态的逻辑电路。正常情况下，红、黄、绿灯只有一个亮，否则视为故障状态，发出报警信号，提醒有关人员修理。

[解] 1. 逻辑抽象

输出变量：Z（有无故障）{
 1--有
 0--无

2. 卡诺图化简

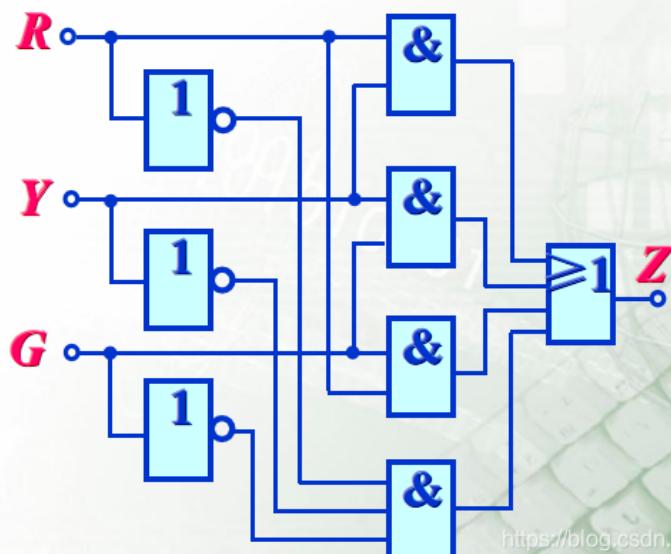
$$Z = \overline{R} \overline{Y} \overline{G} + RY$$

$+ RG + YG$

列真值表			
R	Y	G	Z
0	0	0	1
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

3. 画逻辑图

$$Z = \overline{RYG} + RY + RG + YG$$



补充例4：用与非门设计四变量的多数表决电路。当输入变量A、B、C、D有3个或3个以上为1时输出为1，输入为其他状态时输出为0。

[解] 真值表如表所示

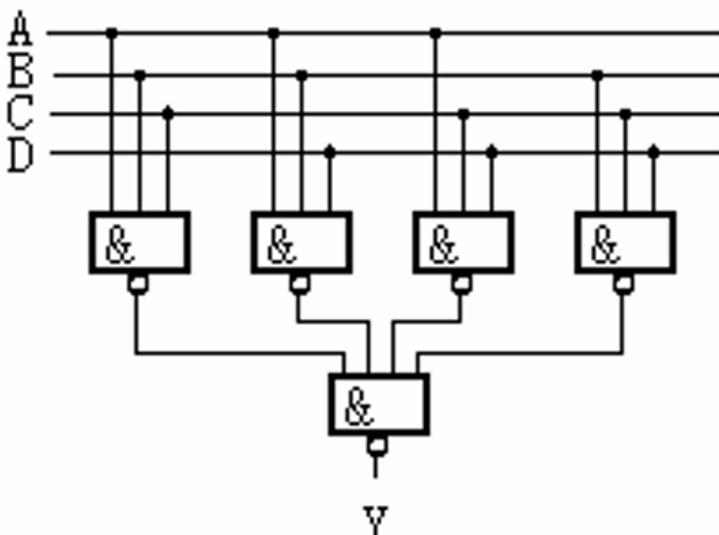
表 A3.3+

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

155

$$\begin{aligned}
 Y &= \overline{ABCD} + A\overline{B}CD + AB\overline{C}D + ABC\overline{D} + ABCD \\
 &= ABC + ABC + ACD + BCD \\
 &= \overline{\overline{ABC} \cdot \overline{ABD} \cdot \overline{ACD} \cdot \overline{BCD}}
 \end{aligned}$$

逻辑图：



https://blog.csdn.net/qq_43923155

补充例5 有一水箱由大、小两台泵ML和MS供水，如图P3.4所示。水箱中设置了3个水位检测元件A、B、C。水面低于检测元件时，检测元件给出高电平；水面高于检测元件时，检测元件给出低电平。现要求当水位超过C点时水泵停止工作；水位低于C点而高于B点时MS单独工作；水位低于B点而高于A点时ML单独工作；水位低于A点时ML和MS同时工作。试用门电路设计一个控制两台水泵的逻辑电路，要求电路尽量简单。

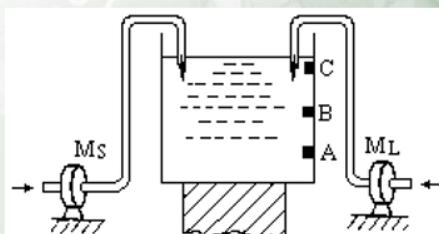


图 P3.4 https://blog.csdn.net/qq_43923155

真值表如表 A3.4 所示。

表 A3.4

A	B	C	MS	ML
0	0	0	0	0
0	0	1	1	0
0	1	0	X	X
0	1	1	0	1
1	0	0	X	X
1	0	1	X	X
1	1	0	X	X
1	1	1	1	1

真值表中的 $\bar{A}\bar{B}\bar{C}$ 、 $A\bar{B}\bar{C}$ 、 $\bar{A}B\bar{C}$ 、 $A\bar{B}C$ 为约束项，利用卡诺图图 A3.4(a)化简后得到：

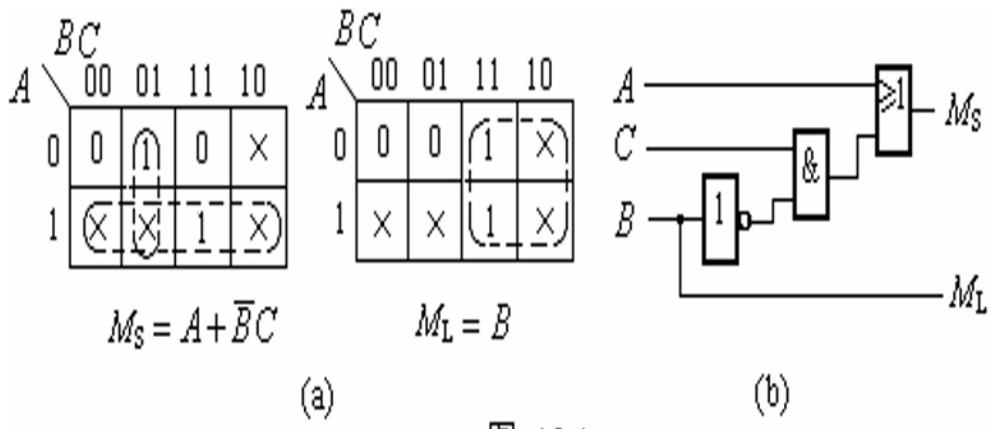


图 A3.4

$$M_S = A + \bar{B}C, \quad M_L = B \quad (M_S, M_L \text{ 的 } 1 \text{ 状态表示工作, } 0 \text{ 状态表示停止})$$

3. 常用组合逻辑模块

(1) 优先编码器的特点，给定输入信号能判断出输出信号，以及根据输出端的状态，判断芯片工作状态

在优先编码器电路中，允许同时输入两个以上的编码信号。不过在设计优先编码器时已经将所有的输入信号按优先顺序排了队，当几个输入信号同时出现时，只对其中优先权最高的一个进行编码。

74HC148优先编码器：(二进制)

S' 为选通输入端，只有在 $S'=0$ 的条件下，编码器才能正常工作；而在 $S'=1$ 时，所有输出端均被封锁在高电平

只有当所有编码输入端都是高电平（即没有编码输入），而且 $S=1$ 时， $Y's$ 才是低电平。因此， $Y's$ 的低电平输入信号表示“电路工作，但无编码输入”

$Y'EX$ 的低电平输出信号表示“电路工作，但无编码输入”。

输入输出均以低电平作为有效信号

$I'7$ 的优先权最高， $I'0$ 的优先权最低

74HC147优先编码器：(二-十进制)

编码器的输出是反码形式的BCD码。优先权以 $I'9$ 为最高， $I'1$ 为最低。当 $I'1 \sim I'9$ 均为无效输入（均为1）时，隐含了表示 $I'0$ 为有效输入，编码输出 $Y'3Y'2Y'1Y'0=1111$ 。

(2) 译码器 正常译码的情况下，给定输入信号能够判断出输出端的状态

1. 二进制译码器

二进制译码器的输入是一组二进制代码，输出是一组与输入代码一一对应的高低电平信号

译码器也将其他的每一个输入代码译成对应输出端的高电平信号（也可视为一个8位的二进制代码）：

输入			输出							
A_2	A_1	A_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
1	1	1	1	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	CSDN @Zevalin 爱灰灰	爱灰灰

74HC138译码器

$Y'0 \sim Y'7$ 同时也是 A_2 , A_1 , A_0 这三个变量的全部最小项的译码输出 ($Y'0 \sim Y'7$)

当 $S1=1$, $S'2+S'3=0$ 时, GS输出为高电平 ($S=1$) , 译码器处于工作状态, 否则, 译码器被禁止, 所有输出端被封锁在高电平

2.74HC42 (二-十进制)

将输入BCD码的十个代码译成10个高低电平输出信号, 这个电路具有拒绝伪码的功能

逻辑表达式:

$$\begin{aligned}
 Y_0 &= \overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0} & Y_1 &= \overline{A_3} \overline{A_2} \overline{A_1} A_0 \\
 Y_2 &= \overline{A_2} A_1 \overline{A_0} & Y_3 &= \overline{A_2} A_1 A_0 \\
 Y_4 &= A_2 \overline{A_1} \overline{A_0} & Y_5 &= A_2 \overline{A_1} A_0 \\
 Y_6 &= A_2 A_1 \overline{A_0} & Y_7 &= A_2 A_1 A_0 \\
 Y_8 &= A_3 \overline{A_0} & Y_9 &= A_3 A_0
 \end{aligned}
 \quad \text{CSDN @Zevalin 爱灰灰}$$

3.BCD-七段显示译码器

BCD-七段显示译码器是一种将二进制编码转换为七段显示的电路, 其核心功能包括接收四位二进制数 (BCD码) , 通过逻辑处理, 驱动七段数码管显示相应的十进制数字。

1. 基本结构与输入输出:

- 该译码器通常有四个输入端, 分别代表二进制数的四位 (8421码) , 以及七个输出端, 用于控制七段数码管的显示。
- 输入信号通过内部的逻辑电路进行处理, 输出信号控制七段数码管的每个段 (a到g) 是否点亮, 从而形成数字的显示。

2. 附加控制端：

- 例如，试灯（LT）、灭零输入（RBI）和灭灯输入（BI）等，这些端子可以用于控制数码管的特殊显示状态。
- 例如，LT端可以用来测试所有LED段是否能正常点亮，而BI端可以用于关闭所有LED显示。
- BI' / RBO' 作为输入端使用时，称灭灯输入控制端。允许用户直接关闭所有LED显示。
- BI' / RBO' 作为输出端使用时，称灭零输出端，用于在多位数码显示时控制前导零的显示。
只需在整数部分把高位的RBO'与低位的 RBI'相连，在小数部分把低位的RBO'与高位的 RBI'相连，就可以把前后多余的零熄灭了

3. 真值表与功能实现：

- 每一个输入的BCD码都对应一个特定的七段数码管显示。例如，输入0101（BCD表示的5），则应该点亮a、f、g、c、d五个LED段以显示数字5。

(3) 数据选择器 给定一个数据选择器能分析出地址输入端的个数，掌握数据选择器的功能

2、4选1数据选择器

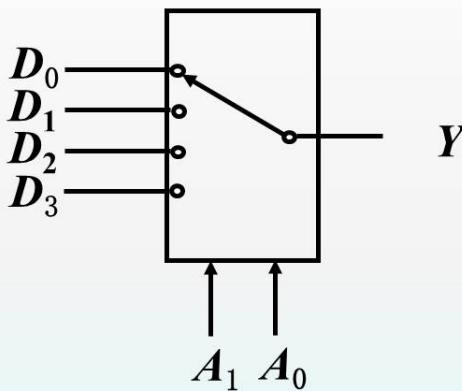
输入			输出
D	A ₁	A ₀	Y
D ₀	0	0	D ₀
D ₁	0	1	D ₁
D ₂	1	0	D ₂
D ₃	1	1	D ₃

真值表

地址变量

由地址码决定从4路输入中
选择哪一路输出。

即：



A_1	A_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

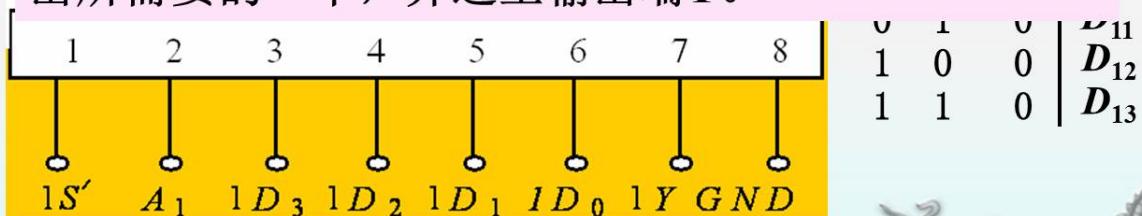
$$Y = A'_1 A'_0 D_0 + A'_1 A_0 D_1 + A_1 A'_0 D_2 + A_1 A_0 D_3$$

集成电路数据选择器

型号：74HC153 双4选1数据选择器

$V_{DD} = 2.7V$ $A = 2D_0$ $2D_1$ $2D_2$ $2D_3$ $2V$

74153包含两个完全相同的4选一MUX，两个MUX有公共的地址输入端，而数据输入和输出端各自独立。通过给定不同的地址代码，即可从4个输入数据中选出所需要的一个，并送至输出端Y。



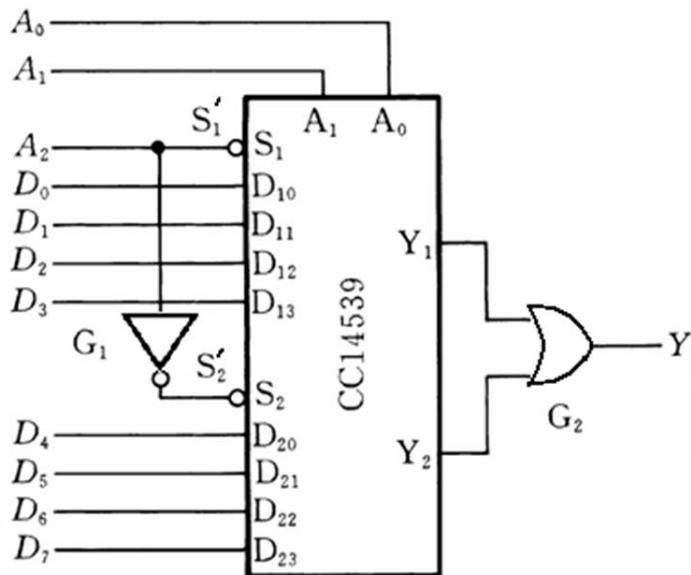
$$Y_1 = [(A'_1 A'_0)D_{10} + (A'_1 A_0)D_{11} + (A_1 A'_0)D_{12} + (A_1 A_0)D_{13}] \cdot S_1$$

$$Y_2 = [(A'_1 A'_0)D_{20} + (A'_1 A_0)D_{21} + (A_1 A'_0)D_{22} + (A_1 A_0)D_{23}] \cdot S_2$$

扩展：

例4.3.4

用双4选1数据选择器构成8选1数据选择器。

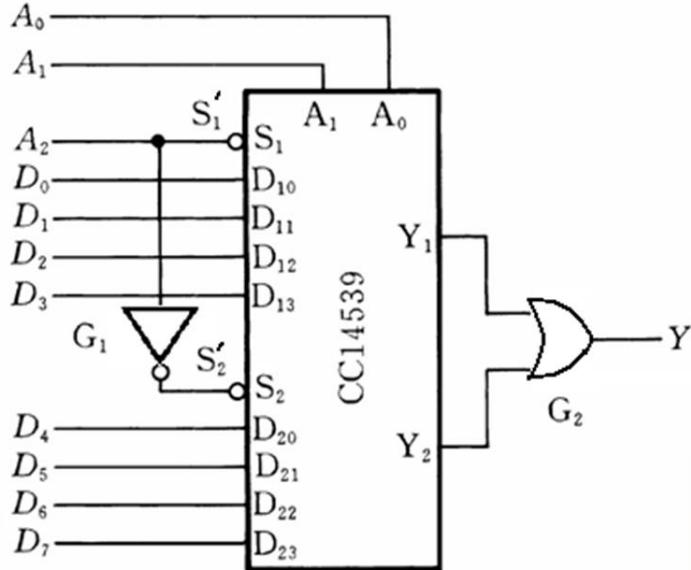


A₂=0时，上边一半数据选择器工作，数据D₀~D₃选择一路输出。

扩展：

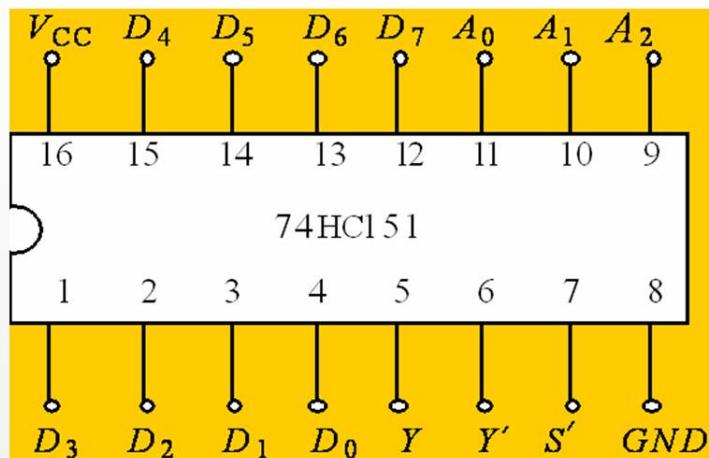
例4.3.4

用双4选1数据选择器构成8选1数据选择器。



A₂=1时，下边一半数据选择器工作，数据D₄~D₇选择一路输出。

集成8选1
数据选择器
74HC151



$S' = 1$ 时，选择器被禁止，无论地址码是什么， Y 总是等于 0

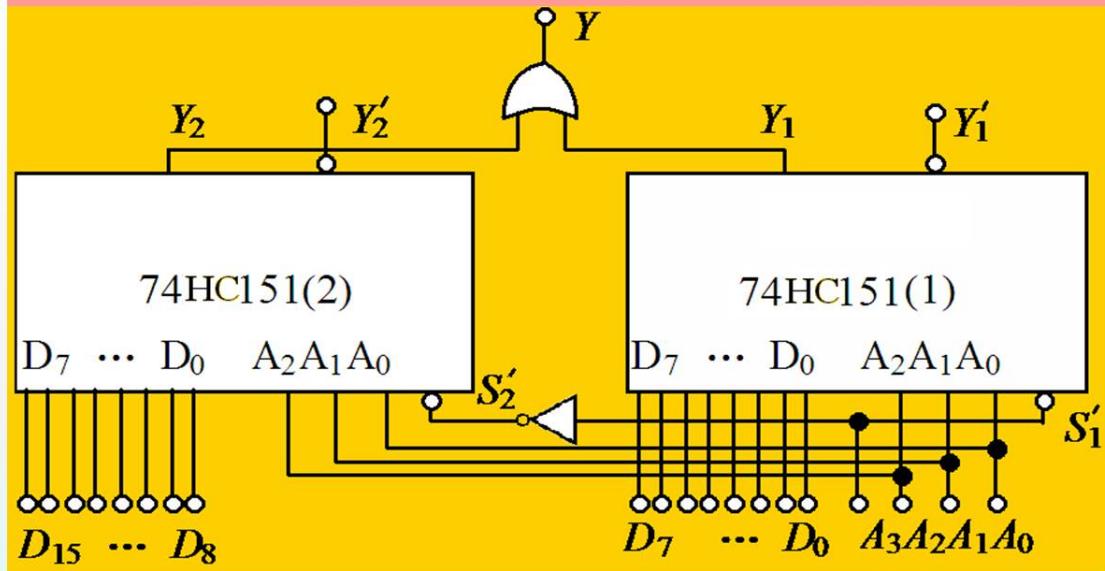
$S' = 0$

$$Y = (A_2'A_1'A_0')D_0 + (A_2'A_1'A_0)D_1 + (A_2'A_1A_0')D_2 + (A_2'A_1A_0)D_3 \\ + (A_2A_1'A_0')D_4 + (A_2A_1'A_0)D_5 + (A_2A_1A_0')D_6 + (A_2A_1A_0)D_7$$

7
4
H
C
1
5
1
的
真
值
表

	输入					输出	
	D	A_2	A_1	A_0	S'	Y	Y'
	×	×	×	×	1	0	1
D_0	0	0	0	0	0	D_0	D_0'
D_1	0	0	1	0	0	D_1	D_1'
D_2	0	1	0	0	0	D_2	D_2'
D_3	0	1	1	0	0	D_3	D_3'
D_4	1	0	0	0	0	D_4	D_4'
D_5	1	0	1	0	0	D_5	D_5'
D_6	1	1	0	0	0	D_6	D_6'
D_7	1	1	1	0	0	D_7	D_7'

2片8选1数据选择器74LS151构成16选1的数据选择器



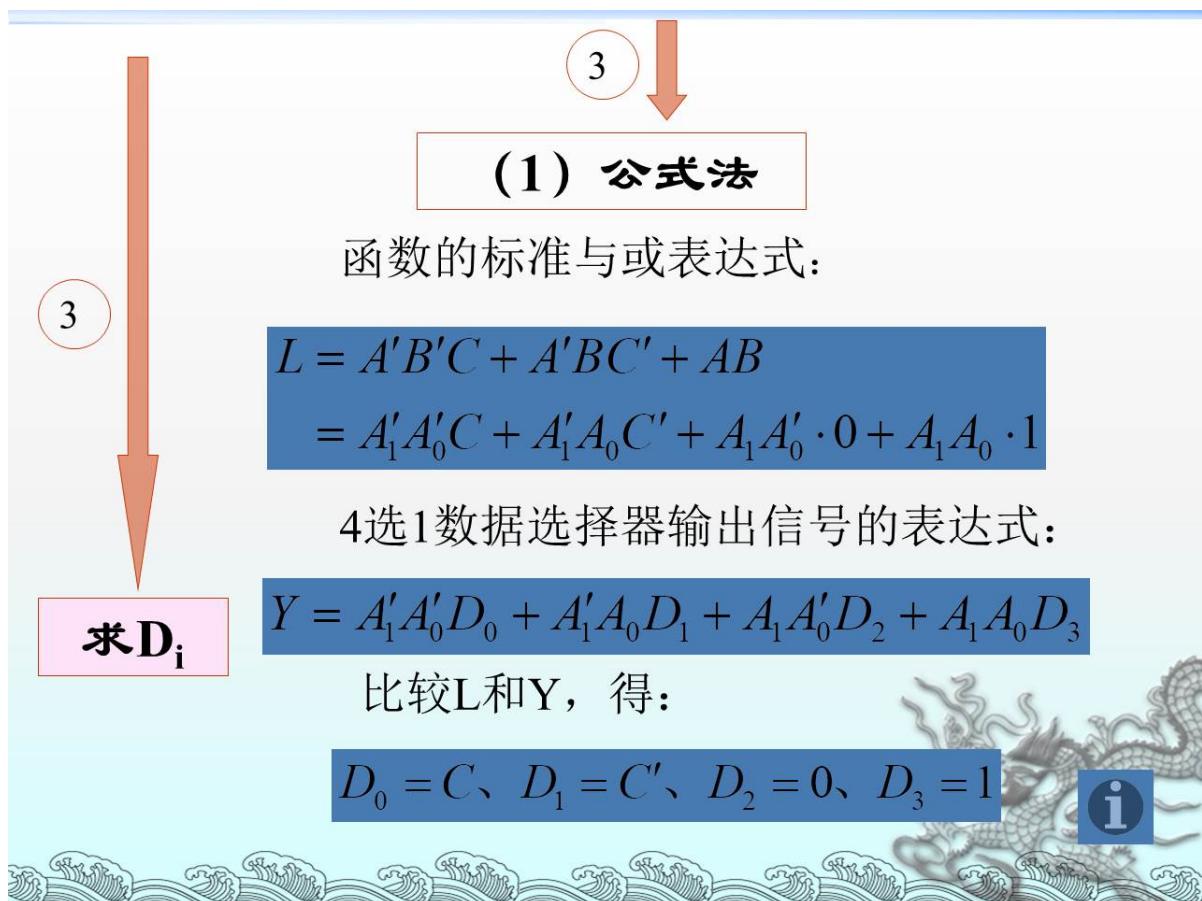
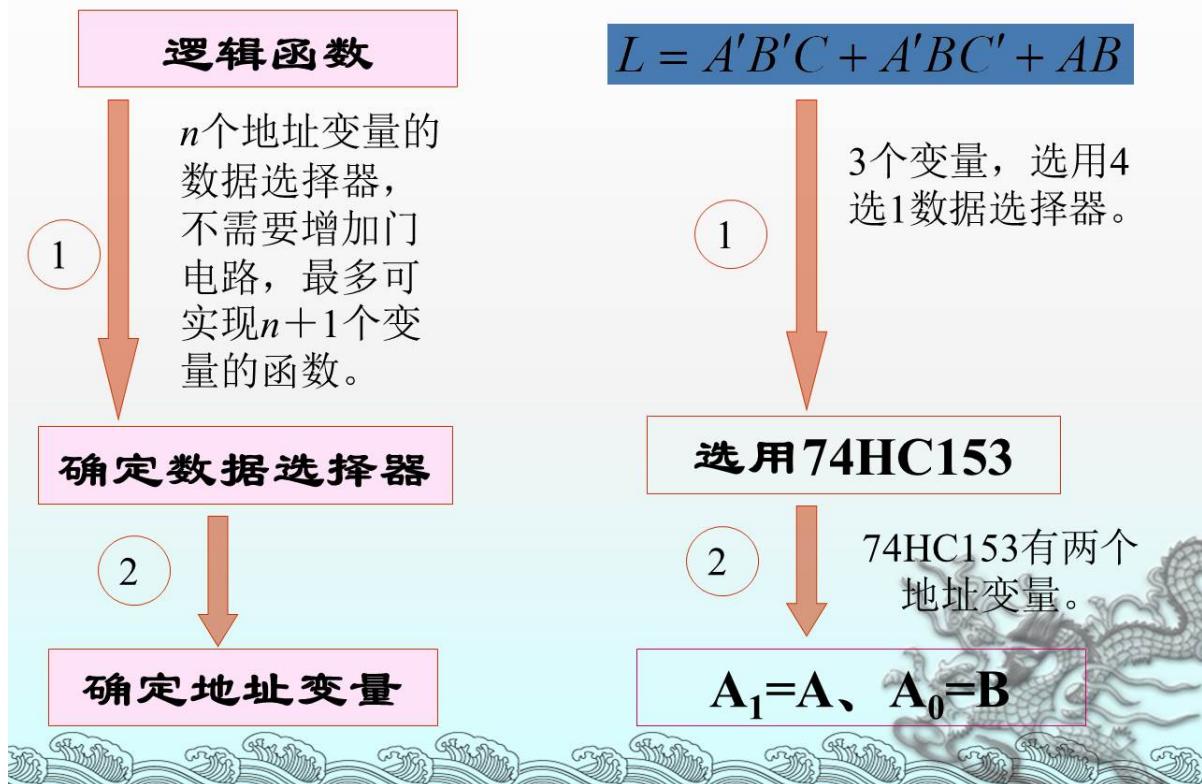
$A_3 = 0$ 时, $S'_1 = 0$ 、 $S'_2 = 1$, 片(2)禁止、片(1)工作

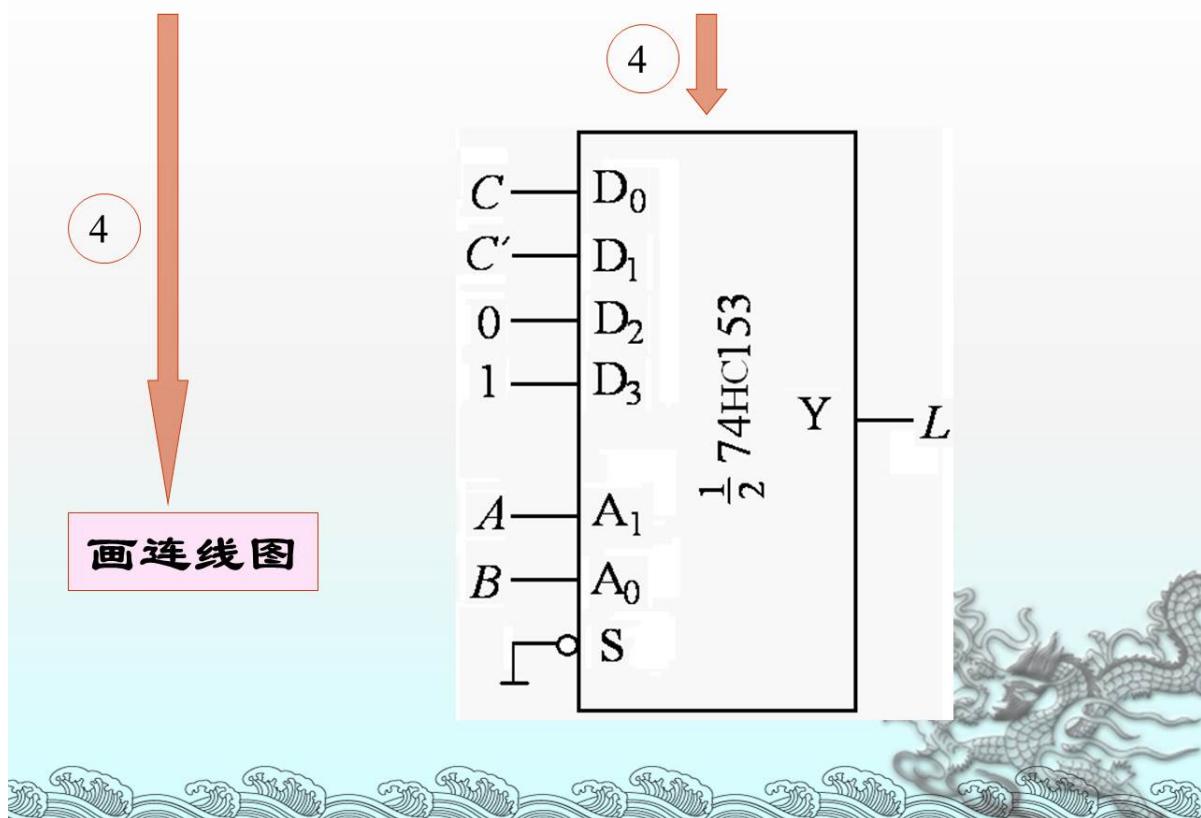
$A_3 = 1$ 时, $S'_1 = 1$ 、 $S'_2 = 0$, 片(1)禁止、片(2)工作

用数据选择器设计组合逻辑电路

步骤:

- 1.列出所求逻辑函数的真值表，写出其最小项表达式。
- 2.根据上述函数包含的变量数，选定数据选择器。
- 3.对照比较所求逻辑函数式和数据选择器的输出表达式确定选择器输入变量的表达式或取值。
- 4.按照求出的表达式或取值连接电路，画电路连线图。





(2) 真值表法

$$L = A'B'C + A'BC' + AB$$

求 D_i 的方法

m_i	A	B	C	L
m_0	0	0	0	0
	0	0	1	1
m_1	0	1	0	1
	0	1	1	0
m_2	1	0	0	0
	1	0	1	0
m_3	1	1	0	1
	1	1	1	1

$C=1$ 时 $L=1$,
故 $D_0=C$

$C=0$ 时 $L=1$,
故 $D_1=C'$

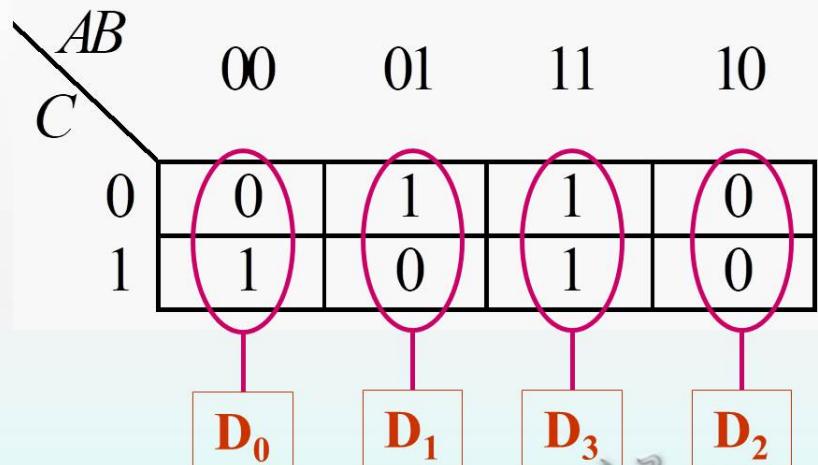
$L=0$, 故
 $D_2=0$

$L=1$, 故
 $D_3=1$

(3) 图形法

$$L = A'B'C + A'BC' + AB$$

求 D_i 的方法



$$D_0 = C, D_1 = C', D_2 = 0, D_3 = 1$$

例4.3.6

$$Z = A'B'C' + AC + A'BC$$

解： ①写出最小项表达式

$$Z = A'B'C' + AC + A'BC = A'B'C' + AB'C + ABC + A'BC$$

②选用8选1数据选择器74HC151，当S'=0时，
令 $A_2=A$ 、 $A_1=B$ 、 $A_0=C$ ，代入上式得：

$$Z = A'_2 A'_1 A'_0 + A_2 A'_1 A_0 + A_2 A_1 A_0 + A'_2 A_1 A_0$$

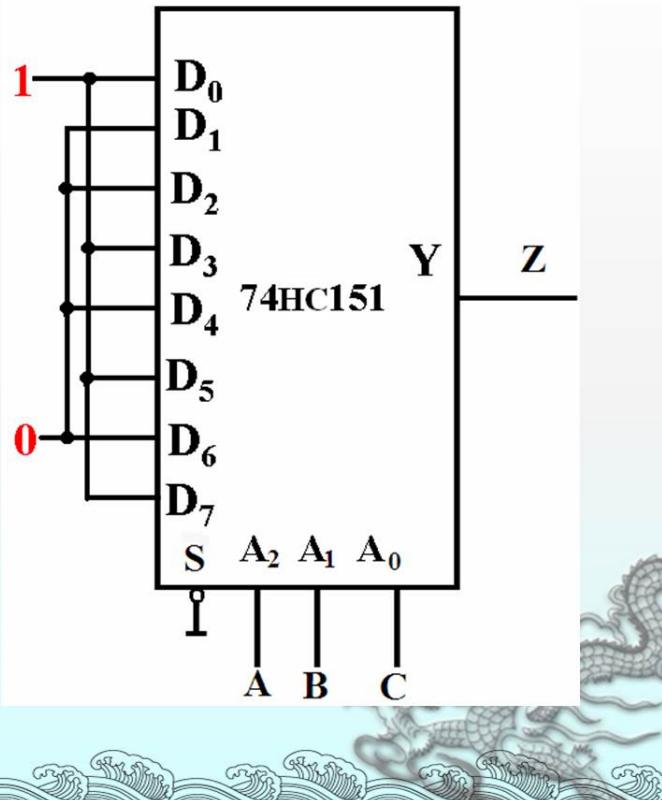
③对照74HC151输出表达式，求 D_i

$$Y = (A'_2 A'_1 A'_0)D_0 + (A'_2 A'_1 A_0)D_1 + (A'_2 A_1 A'_0)D_2 + (A'_2 A_1 A_0)D_3 \\ + (A_2 A'_1 A'_0)D_4 + (A_2 A'_1 A_0)D_5 + (A_2 A_1 A'_0)D_6 + (A_2 A_1 A_0)D_7$$

比较L和Y，得：

$$\begin{aligned}D_0 &= 1, D_1 = 0, \\D_2 &= 0, D_3 = 1, \\D_4 &= 0, D_5 = 1, \\D_6 &= 0, D_7 = 1\end{aligned}$$

④画连线图



另解： ①写出最小项表达式

$$Z = A'B'C' + AC + A'BC = A'B'C' + AB'C + ABC + A'BC$$

②选用双4选1数据选择器74HC153其中的一半，

当S₁'=0时，令A₁=A、A₀=B,代入上式得：

$$Z = A'_1 A'_0 C' + A_1 A'_0 C + A_1 A_0 C + A'_1 A_0 C$$

③对照74HC153输出表达式，求D_i

$$Y_1 = [(A'_1 A'_0)D_{10} + (A'_1 A_0)D_{11} + (A_1 A'_0)D_{12} + (A_1 A_0)D_{13}] \cdot S_1$$

可得： D₁₀=C' D₁₁=C D₁₂=C D₁₃=C

(4)加法器 半加器的组成，全加器的本位和与进位的逻辑表达式，全加器的真值表，全加器的波形图

1位加法器

若不考虑有来自低位的进位将两个1位二进制数相加，称为半加。实现半加运算的电路叫做半加器。

A ---被加数； B ---加数； S ---本位和； Co ---进位。

真值表

A	B	Co	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = A'B + AB' = A \oplus B$$

$$Co = AB$$

将两个多位二进制数相加时，除了最低位以外，每一位都应考虑来自低位的进位，即将两个对应的加数和来自低位的进位3个数相加。这种运算称为全加，所用电路称为全加器。

A ---被加数； B ---加数； C_i ---低位的进位；
 S ---本位和； Co ---进位。

逻辑状态表见下页

C_i	A	B	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

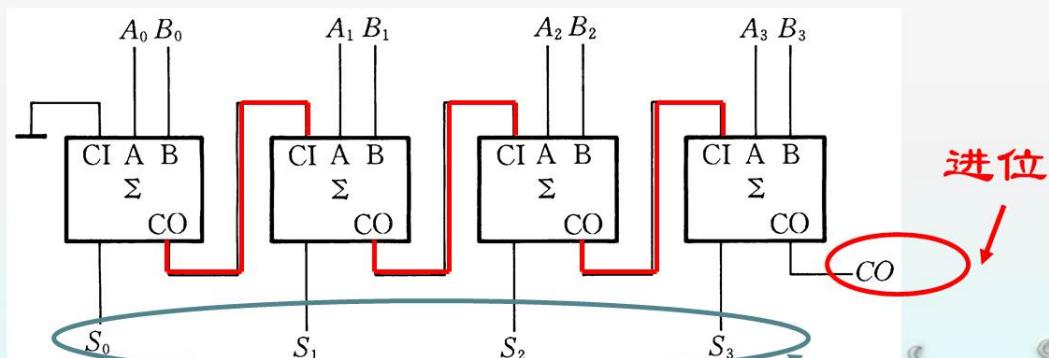
$$S = C'_i A' B + C'_i A B' + C_i A' B' + C_i A B = C_i \oplus A \oplus B$$

$$C_o = AB + C_i A + C_i B = AB + C_i (A + B)$$

多位加法器

(1) 串行进位加法器

设计思想：依次将低位全加器的进位输出端CO接到高位全加器的进位输入端CI即可构成多位串行加法器。



注意： $CI_0=0$

三、用加法器设计组合电路

对“变量+变量”或
“变量+常量”类型

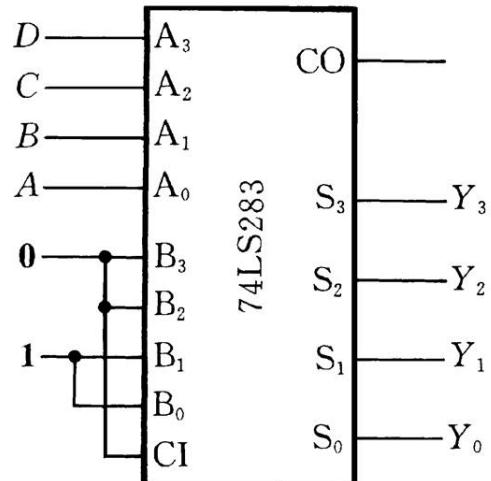
例4.4.4.1 设计一电路，
将BCD的8421码转换
为余3码

解： **BCD码+0011=余3码**

设输入8421码用变
量DCBA表示，输
出余三码用变量
 $Y_3 Y_2 Y_1 Y_0$ 表示。

则有

$$Y_3 Y_2 Y_1 Y_0 = DCBA + 0011$$

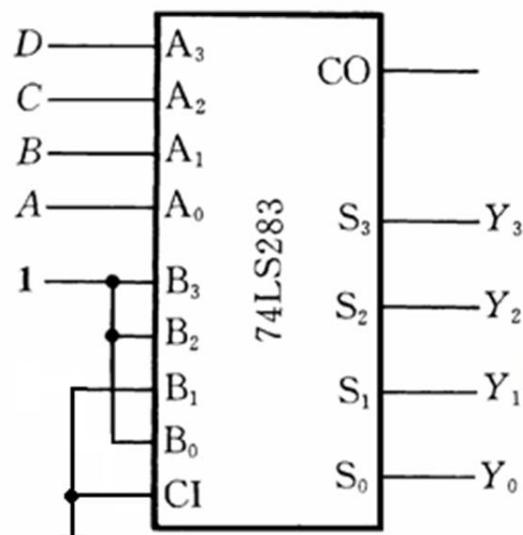


例4.4.4.2 用一片74LS283将余三码转换成8421BCD码。

解： **余3码 - 0011=BCD码**

设输入余三码用变
量DCBA表示，输
出8421码用变量
 $Y_3 Y_2 Y_1 Y_0$ 表示。则
有

$$\begin{aligned} Y_3 Y_2 Y_1 Y_0 &= DCBA + [-0011]_{\text{补}} \\ &= DCBA + 1101 \end{aligned}$$



第五章 半导体存储 电路

1. 掌握触发器的特点 掌握各种逻辑功能的触发器的电路结构与动作特点，掌握触发器输出波形的画法即给出输入信号与时钟信号，能够根据触发器的动作特点画出输出端Q的波形图

[13RS锁存器哔哩哔哩bilibili](#)

能够存储1位二值信号（0, 1）的基本单元电路统称为触发器。

触发器是构成时序逻辑电路的基本电路，是联系组合逻辑电路和时序逻辑电路的桥梁。

一、触发器的两个基本特点

1. 具有两个能自行保持稳定状态表示逻辑状态的0和1
2. 根据不同的输入信号可以置成1或0状态
 - 有两个稳定的状态：“0”状态、“1”状态。
 - 有两个互补的输出端：

“1”态时： $Q=1, \bar{Q}=0$

“0”态时： $Q=0, \bar{Q}=1$

- 有一组输入信号（通常为1~3）个：称为激励、时钟
- 工作状态：

在输入的作用下，由现态转变成次态。

现态：输入作用前的状态，记作 Q^n 和 \bar{Q}^n ，简记为 Q 和 \bar{Q} 。

次态：输入作用后的状态，记作 $Q^{(n+1)}$ 和 $\bar{Q}^{(n+1)}$ 。

注意：次态不仅与输入有关，而且与现态有关。

二、触发器的分类

(一)按电路结构形式不同划分

- 基本RS-触发器（锁存器）
- 同步触发器（电平触发）
- 主从触发器（脉冲触发）
- 边沿触发器（边沿触发）
- CMOS工艺触发器

(二)按逻辑功能划分

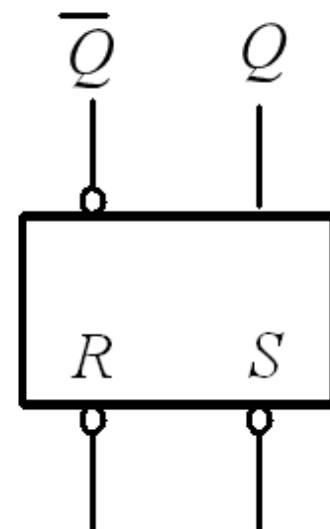
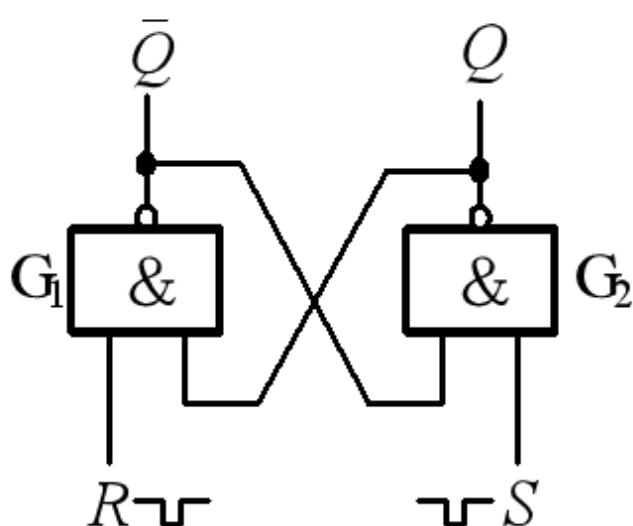
- RS
- JK
- D
- T
- T'

(三)按存储数据的原理不同可分为

- 静态触发器
- 动态触发器

三、基本RS触发器

电路及逻辑符号(与非门实现)



电路

逻辑符号

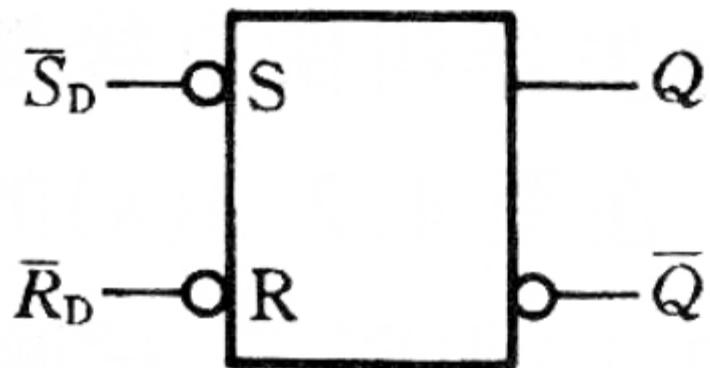
输入端: R: 复位端 (Reset) S: 置位端 (Set)

输出端: Q: 状态输出端 \bar{Q} : 反相状态输出端

输入信号R、S的作用方式: **低电平有效**

当输入信号无效时, 输出信号Q、 \bar{Q} 必定保持反相。

(一)逻辑功能描述



与非门构成的基本RS-FF的图形符号

$\overline{S_D}$	$\overline{R_D}$	Q^n	Q^{n+1}	功能
1	1	0	0	保持
1	1	1	1	
0	1	0	1	置1
0	1	1	1	
1	0	0	0	置0
1	0	1	0	
0	0	0	1*	不定
0	0	1	1*	

逻辑功能表

R S	$Q^{(n+1)}$	功能说明
0 0	d	不定
0 1	0	置0
1 0	1	置1
1 1	Q	不变

状态转换表

Q	$Q^{(n+1)}$			
	RS=00	RS=01	RS=11	RS=10
0	d	0	0	1
1	d	0	1	1

由功能表得到状态表

用特征方程描述逻辑功能

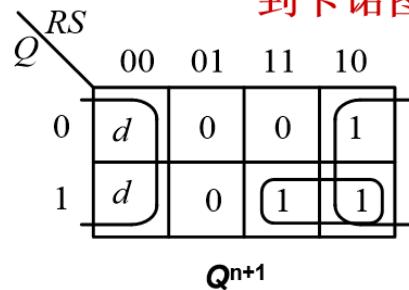
$$\text{次态方程: } Q^{n+1} = \bar{S} + RQ$$

$$\text{约束方程: } R + S = 1$$

由卡诺图导出次态方程

特征方程

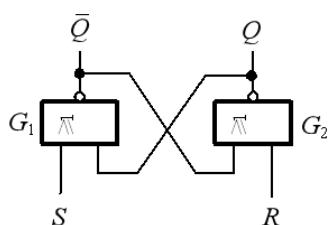
由状态表得到卡诺图



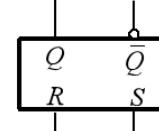
次态卡诺图

(二)或非门构成

2 用或非门构成的基本R-S触发器



电路



逻辑符号
(注意, 无小圆圈)

R S	Q^{n+1}	功能说明
0 0	Q	不变
0 1	1	置1
1 0	0	置0
1 1	d	不定

功能表

结论:

- (1) 不论现态是什么,
在 **R** 端施加**高电平**能将现态强
制性地转换到“0”态;
在 **S** 端施加**高电平**能将现态强
制性地转换到“1”态;
R 和 **S** 不能同时施加**高电平**。
- (2) **R** 和 **S** 端的有效电平为**高电平**

次态方程: $Q^{n+1} = S + \bar{R}Q$
约束方程: $R \cdot S = 1$

(三)基本RS触发器的特点

优点：电路简单，**直接置位、复位**，操作方便。

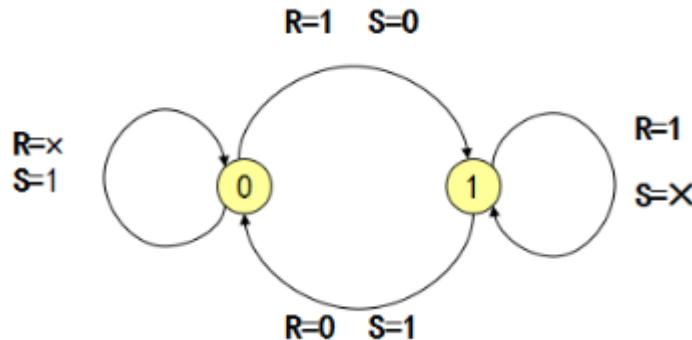
常用于**键盘输入、消除开关噪声等场所**。

缺点： (1) 存在**约束关系**，操作不便；

(2) 对R、S**要求严格**，要相互配合，准确实时。

改进措施：先施加好R、S信号，再用另外一个统一、标准的信号实施触发。

(四)状态转换图



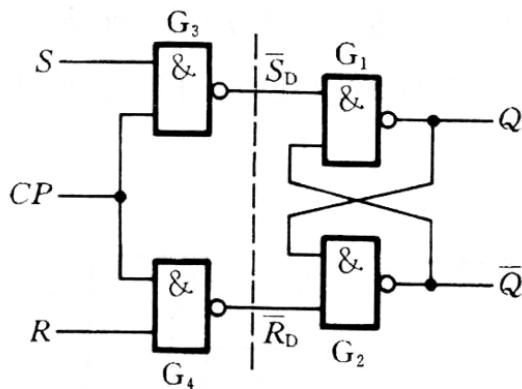
四、同步触发器

在数字系统中，为协调各部分的动作，常要求某些触发器于同一时刻动作。为此，必须引入同步信号，是这些触发器只有在同步信号到达时才按输入信号改变状态。通常把这个同步信号叫做时钟脉冲，或称为时钟信号，简称时钟，用CLK表示。

同步触发器即时钟控制的电平触发器。

(一)同步RS触发器

1. 电路结构与工作原理分析



同步RS-FF的逻辑图

同步RS-FF的特性表

CP	S	R	Q^n	Q^{n+1}	
0	x	x	0	0	
0	x	x	1	1	
1	0	0	0	0	
1	0	0	1	1	
1	1	0	0	1	
1	1	0	1	1	
1	0	1	0	0	保持
1	0	1	1	1	置1
1	1	0	1	0	置0
1	1	1	0	1*	
1	1	1	1	1*	不定

注： *CP回到低电平后状态不定。

- 注意：比基本RS触发器多了一层与非门，所以是**高电平触发**
- 只有CP=1时，触发器输出端的状态才会受输入信号的控制
- 输入信号同样遵守S*R=0的约束条件

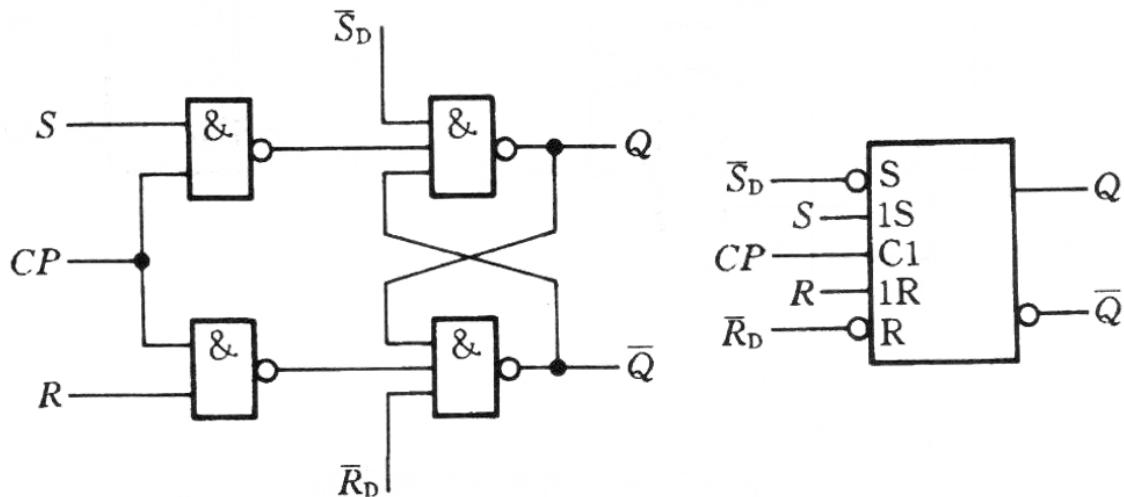
- 同步RS-触发器的特性方程和控制输入端的约束条件如下

$$\begin{cases} Q^{n+1} = S + \bar{R}Q^n \\ S \cdot R = 0 \end{cases}$$

2、设有专门的异步置位输入端和异步复位输入端的同步RS-触发器触发器

只要在S（异步置位端）或R（异步复位端）加入低电平，即可立即将触发器置1或置0，而不受时钟信号的控制。触发器在时钟信号控制下正常工作时应使S和R处于高电平。

用S或R将触发器置位或复位应当在CLK=0的状态下进行，否则在S或R返回高电平以后预置的状态不一定能保存下来

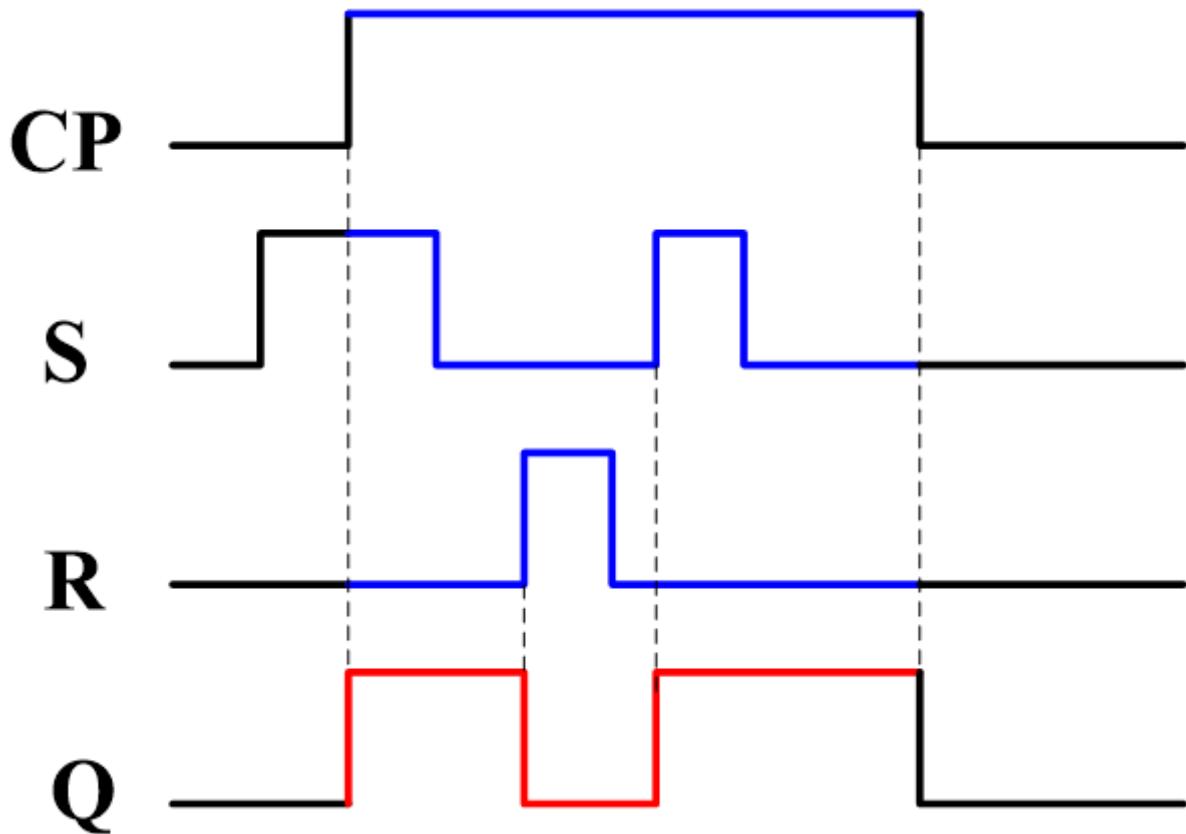


实用同步RS-FF的逻辑图和逻辑符号

3、动作特点

- (1) 只有当CLK变为有效电平时，触发器才能接受输入信号，并按照输入信号将触发器的输出置成相应状态。
- (2) 在CLK=1的全部时间里，S和R状态的变化都可能引起输出状态的改变。在CLK回到0以后，触发器保存的是CLK回到0以前瞬间的状态。

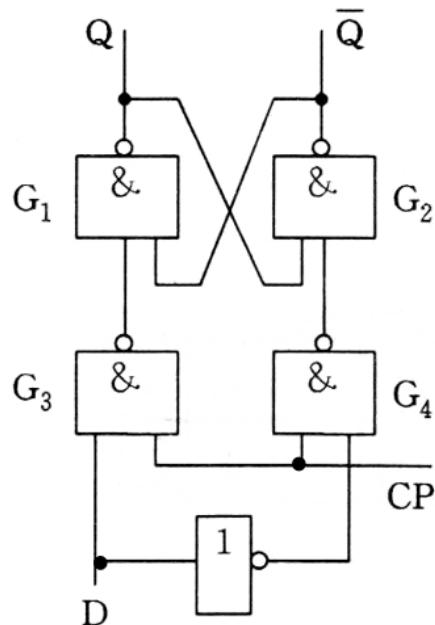
可知，若在CLK=1的期间内输入信号发生多次变化，则触发器的状态也会发生多次反转，这就降低了电路的抗干扰能力。



(二)同步D触发器

为了从根本上避免同步RS触发器R、S同时为1的情况出现，可以在R和S之间接一非门。这种单输入的触发器叫做**同步D触发器**(又称D锁存器)

1.逻辑图、特性表和特性方程



同步D-FF的特性表

CP	D			说明
0	x	0	0	保持
	1	1	1	
1	0	0	0	送0
	1	0	1	
1	1	0	1	送1
	1	1	1	

同步D-FF的逻辑图Flip-flop

由特性表可得同步**D-FF**的特性方程为:
$$Q^{n+1} = D$$

2.逻辑功能

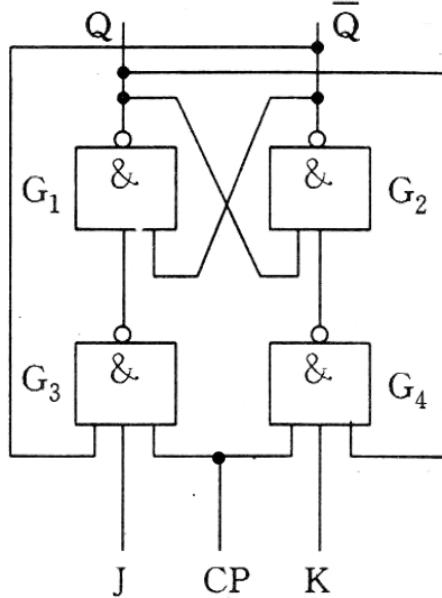
CLK到来时 (CLK=1)，将输入数据D存入触发器，CLK过后 (CLK=0)，触发器保存该数据不变，直到下一个CLK到来时，才将新的数据存入触发器而改变原存数据。

正常工作时要求CLK=1期间D端数据保持不变

(三)同步JK触发器

同步JK-触发器解决了同步RS-触发器输入控制端S=R=1时触发器的新状态不确定的问题。JK-触发器的J端相当于置“1”(S)端，K端相当于置“0”(R)端。

1.逻辑图、特性表和特性方程



同步JK-FF的特性表

CP	J K	Q^n	Q^{n+1}	说明
0	X X	0	0	保持
		1	1	
1	0 0	0	0	置0
		1	1	
1	0 1	0	0	置1
		1	0	
1	1 0	0	1	翻转
		1	1	
1	1 1	0	1	翻转
		1	0	

同步JK-FF的逻辑图

由同步**JK-FF**的特性表可知：

1、同步**JK-FF**的特性方程为：

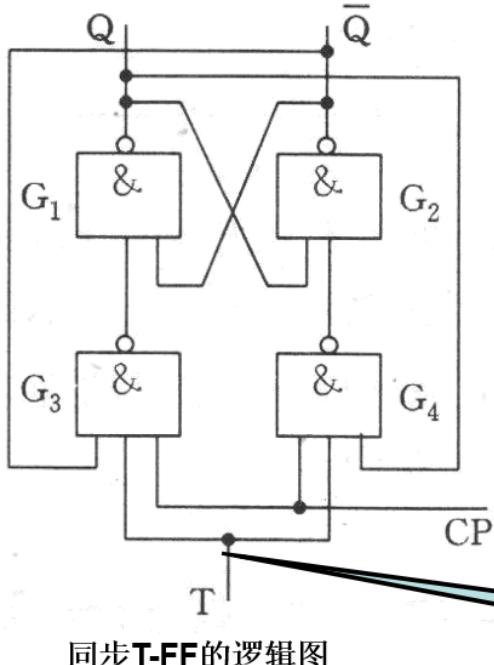
$$Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n$$

2、当J=K=1时， $Q^{n+1} = \bar{Q}^n$ ，触发器处于**翻转**状态，其余情况同同步**RS-FF**一样。

(四)同步T触发器

将JK-触发器的J端和K端连在一起，就得到了T触发器。

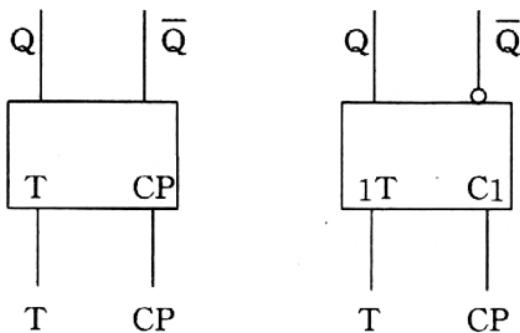
1.逻辑图、特性表和特性方程



同步T-FF的特性表

CP	T	Q^n	Q^{n+1}	说明
0	X	0	0	保持
		1	1	
1	0	0	0	
		1	1	
1	1	0	1	翻转
		1	0	

$$J=K=T$$



同步T-FF的惯用符号和国标符号

由同步T-FF的特性表或将 $J=K=T$ 代入 JK-FF 的特性方程可得同步T-FF的特性方程为：

$$Q^{n+1} = T\bar{Q}^n + \bar{T}Q^n$$

若将T输入端恒接高电平，则成为T'触发器。

$$T'-FF \text{ 的特性方程为: } Q^{n+1} = \bar{Q}^n$$

(五)同步触发器的空翻现象

1. 上述四种功能的同步触发器均属于**电平触发方式**，电平触发方式有高电平触发和低电平触发两种。

2. **触发器的空翻**：在CLK为高电平期间，因输入信号变化而引起触发器状态变化**多于一次**的现象，称为触发器的**空翻**。

由于空翻问题，同步触发器只能用于数据的锁存，而不能实现计数、移位、存储等功能。为了克服空翻，又产生了无空翻的**主从触发器**和**边沿触发器**等新的触发器结构形式。

五、主从触发器

为了提高触发器工作的可靠性，希望在每个CLK周期里输出端的状态只改变一次。为此，在同步触发器的基础上又设计出了主从结构的触发器。

主从触发器的结构特点：

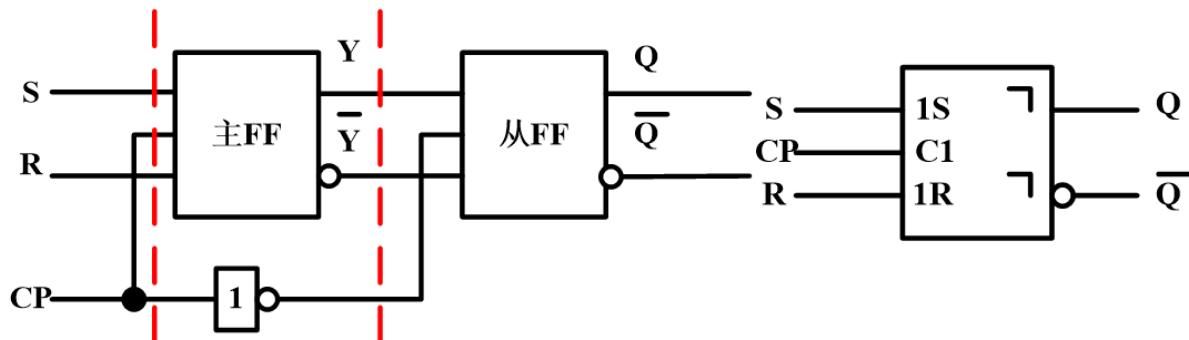
- ○ 前后由主、从两级触发器级联组成
 - 主、从两级触发器的时钟相位相反
1. 1. 主从触发器状态的改变是在CLK下降沿/上升沿完成的，因而这种结构无空翻现象。
 2. **主从触发器在CLK=1期间无法抗干扰**，为克服这一缺点，又出现了边沿触发器。

(一) 主从RS-触发器

1. 电路结构与工作原理

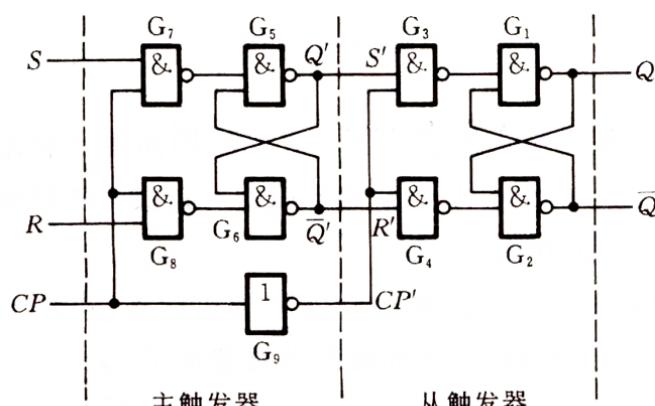
主从RS触发器由两个同样的同步RS触发器组成，但它们的**时钟信号相位相反**。

根据非门的位置，分别为上升沿或下降沿有效。



主从RS-FF的结构框图和图形符号

主从RS-FF的特性表



主从RS-FF的逻辑图

CP	S	R	Q^n	Q^{n+1}
X	X	X	X	Q^n
0	0	0	0	0
0	0	0	1	1
1	0	0	0	1
1	0	0	1	1
0	1	0	0	0
0	1	0	1	0
1	1	0	0	1*
1	1	0	1	1*
1	1	1	0	1*
1	1	1	1	1*

* CP 回到低电平后输出状态不定。

由于主触发器本身仍是一个**同步RS触发器**，所以在CLK=1期间Q'和 \bar{Q}' 状态仍然会随着S、R状态的变化而多次变化，且人需遵守约束条件 **$S \cdot R = 0$** ，且其特性方程仍为：

$$Q^{n+1} = S + \bar{R}Q^n$$

2. 动作特点

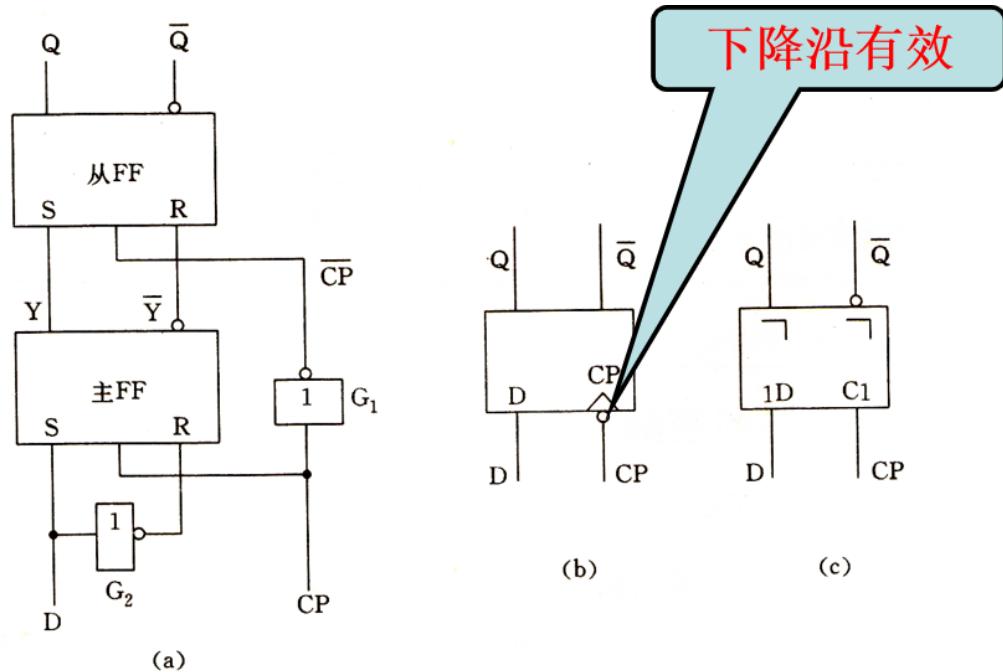
(1) 主从RS-触发器的翻转分两步动作：

第一步，在 $CLK=1$ 期间主触发器接收输入S、R的信号，被置成相应状态。

第二步，在 CLK 下降沿到来时，从触发器按主触发器的状态翻转，Q、 \bar{Q} 端状态的改变发生在 CLK 的下降沿。

(2) 在 $CLK=1$ 的全部时间里，S、R均对主触发器起控制作用，所以必须考虑整个 $CLK=1$ 期间里输入信号的变化过程才能确定触发器的状态。

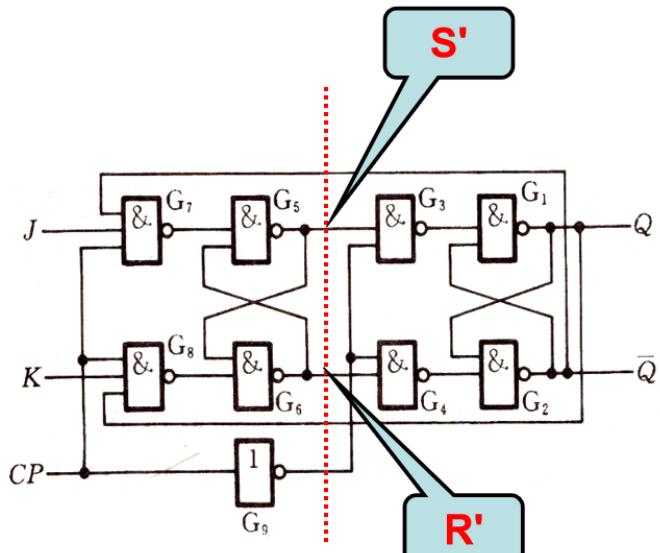
(二) 主从D-触发器



主从D-FF的结构框图、惯用符号和国标符号

其特性方程仍为：
$$Q^{n+1} = D$$

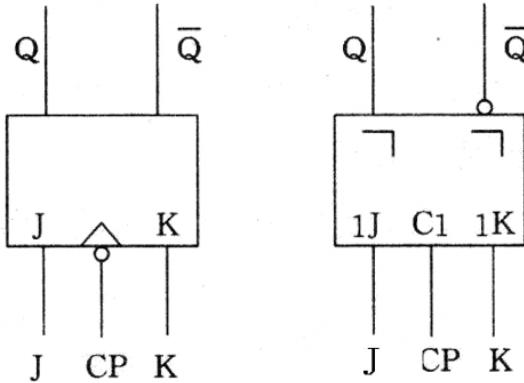
(三)主从JK-触发器



主从JK-FF的逻辑图

主从JK-FF的特性表

CP	J	K	Q^n	Q^{n+1}
x	x	x	x	Q^n
0	0	0	0	0
0	0	1	1	1
1	0	0	0	1
1	0	1	1	1
0	1	0	0	0
0	1	1	1	0
1	1	0	0	1
1	1	1	1	0



主从JK-FF的惯用符号和国标符号

由特性表可知，其特性方程仍为：
$$Q^{n+1} = J\bar{Q}^n + \bar{K}Q^n$$

例：在下图所示的主从JK触发器电路中，若CP、J、K的电压波形如图所示，试求Q和 \bar{Q} 端的电压波形，设 $Q^n = 0$ 。

六、边沿触发器

为了提高触发器的可靠性，增强抗干扰能力，希望触发器的次态仅仅取决于CLK信号下降沿(或上升沿)到达时刻输入信号的状态。为实现这一设想，人们研制了各种边沿触发器。

七、锁存器与触发器

在数字电路设计中，锁存器(Latch)和触发器(Flip-Flop)都是非常重要的存储元件，它们能够存储一位二进制信息。尽管它们在功能上有所相似，但在结构和工作原理上存在一些关键差异，这些差异决定了它们在不同应用场景下的适用性。具体分析如下：

1. 触发机制

- **锁存器**: 锁存器是电平触发的，当使能信号有效时，锁存器的输出会根据输入变化而变化；当使能信号无效时，锁存器保持当前状态。锁存器没有时钟端，其操作不受时钟信号控制，属于非同步控制。
- **触发器**: 触发器受时钟控制，仅当时钟的特定边沿（上升沿或下降沿）到达时，触发器才根据输入信号改变状态。这种时钟沿触发的特性使得触发器能够实现同步操作，有效避免了因信号传播延迟而导致的错误状态变化。

2. 敏感性与稳定性

- **锁存器**: 由于锁存器对输入电平敏感，它容易受到布线延迟的影响，这可能导致输出出现毛刺现象。毛刺是指在数字电路中不希望出现的短暂错误信号，可能会影响电路的可靠性。
- **触发器**: 触发器不易产生毛刺，因为其状态的变化仅发生在时钟边沿时刻。这种特性使得触发器在高速、高稳定性要求的应用中更为适合。

毛刺现象是指在数字电路中，由于信号传输延迟或者信号干扰等原因，导致输出信号在短时间内出现不希望的、短暂的、随机的脉冲，这些脉冲可能会被后续的电路误读为有效的信号，从而引起电路的误操作

3. 应用场合

- **锁存器**: 锁存器适用于简单的数据保存和传输场景，如缓冲器、管道线等，在使能信号控制下能够快速传递和保存数据。
- **触发器**: 触发器由于其同步特性，适合于复杂的时序逻辑电路，如计数器、寄存器、状态机等，能够确保数据在预定的时间点被准确记录和处理。

2. 掌握寄存器的基本概念、寄存器的组成

寄存器是计算机中用于暂时存储和快速访问数据的高速存储设备，位于CPU内部，并直接参与指令执行和数据处理过程。下面将详细解释寄存器的基本概念、组成、类型及其在计算机系统中的重要作用：

1. 基本概念

- **定义**: 寄存器是计算机硬件中最快速的存储单元，其存取时间通常在纳秒级别，远远快于主存(RAM) 和硬盘(HDD/SSD)。
- **功能**: 它主要用于暂时存储CPU所需的数据，如操作数和临时数据，这样能加速数据处理和传输。

2. 物理组成

- **基础组件**: 寄存器通常由具有存储功能的触发器(Flip-Flop)组成，一个触发器可以存储1位二进制代码，因此一个n位寄存器需要n个触发器。
- **结构类型**: 根据结构的不同，寄存器可分为基本寄存器和移位寄存器。基本寄存器只能并行输入输出数据，而移位寄存器则能实现数据的串行输入输出，具备更高的灵活性。

3. 类型与功能

- **通用寄存器**: 主要用于存储算术运算中的操作数和临时数据，可以直接被程序员访问，例如x86架构中的EAX、EBX等。
- **特殊用途寄存器**: 包括程序计数器(PC)、堆栈指针(SP)、状态寄存器(EFLAGS)等，它们存储控制和状态信息，直接影响CPU的操作模式和程序执行流程。
- **段寄存器**: 用于存储内存段的起始地址，如代码段、数据段、堆栈段等，在保护模式下的x86处理器中使用广泛。

寄存器

在数字电路中，寄存器是用来存放二进制数据或代码的电路，是一种基本时序电路。任何现代数字系统都必须把需要处理的数据和代码先寄存起来，以便随时取用。

寄存器是由具有存储功能的触发器组合起来构成的。一个触发器可以存储1位二进制代码，存放 n 位二进制代码的寄存器，需用 n 个触发器来构成。

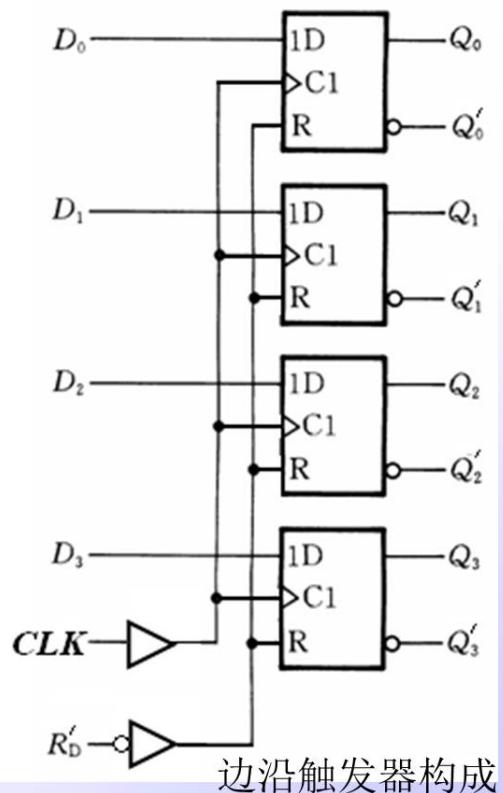
(1) 清零。 $R'_D = 0$ ，异步清零。即有：

$$Q_3 Q_2 Q_1 Q_0 = 0000$$

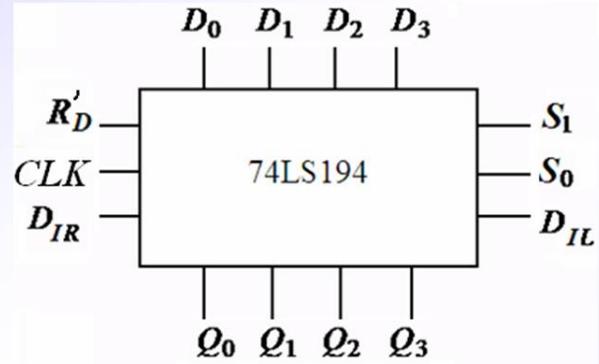
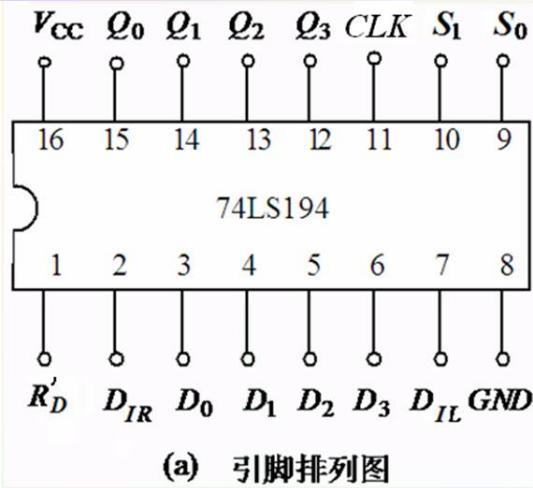
(2) 送数。 $R'_D = 1$ 时， CLK 上升沿送数。即有：

$$Q_3^* Q_2^* Q_1^* Q_0^* = D_3 D_2 D_1 D_0$$

(3) 保持。在 $R'_D = 1$ 、 CLK 上升沿以外时间，寄存器内容将保持不变。



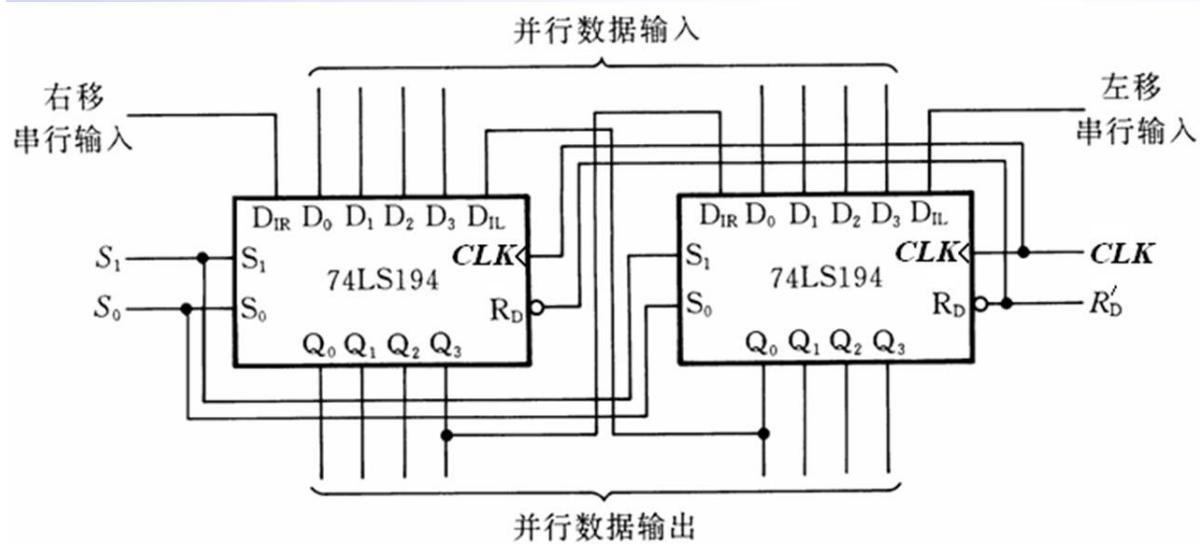
双向移位寄存器



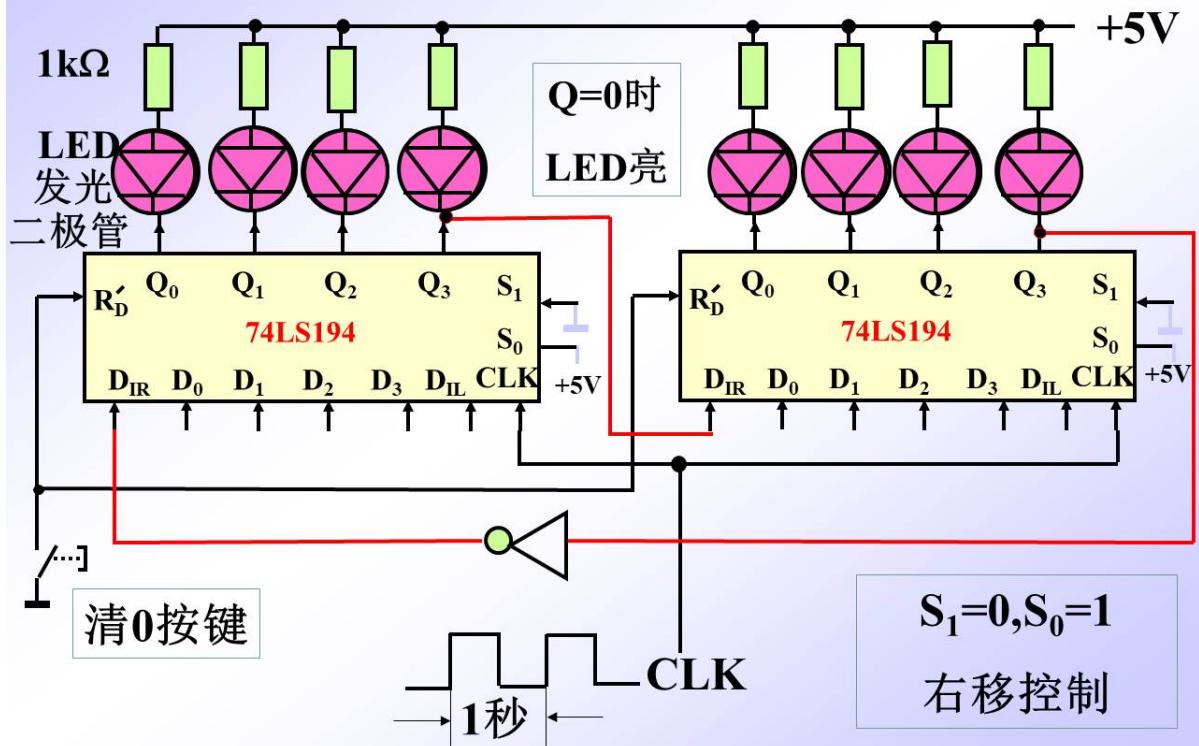
(b) 逻辑功能示意图

R'_D	S_1	S_0	CLK	工作状态
0	X	X	X	异步清零
1	0	0	X	保 持
1	0	1	↑	右 移
1	1	0	↑	左 移
1	1	1	X	并行输入

2片74LS194A接成8位双向移位寄存器



用双向移位寄存器74LS194组成节日彩灯控制电路



3. 理解半导体存储器的特点，ROM和RAM的特点，ROM的分类，了解容量的计算方法，ROM芯片实现组合逻辑函数

半导体存储器是计算机中用于存储数据的电子器件，主要基于半导体材料（如硅）的特性来存储信息。它根据功能和特性的不同，可以分为两大类：**只读存储器（ROM）** 和**随机存取存储器（RAM）**。

一、半导体存储器的特点

半导体存储器的主要特点包括：

1. **速度快**：与机械存储设备相比，半导体存储器的读写速度非常快，因为其基于电子信号操作。
2. **集成度高**：随着集成电路技术的发展，可以在极小的芯片上集成大量的存储单元。
3. **易失性**：大多数半导体存储器在断电后会丢失数据（如RAM），但也有非易失性类型（如ROM、Flash）。
4. **能耗低**：相较于其他类型的存储技术，半导体存储器在待机和运行时的能耗较低。
5. **成本效益**：随着生产技术的成熟，半导体存储器的成本逐渐降低，容量却不断增加。

二、ROM（只读存储器）的特点及分类

ROM是只能读取数据而不能随意写入的存储器，主要用于存储固件和固定数据。其特点是：

1. **非易失性**：断电后数据仍然保持。
2. **预编程**：一般在制造时写入数据，后期不能更改（对于普通ROM）。
3. **成本低**：由于使用较少的晶体管，成本相对较低。

ROM的分类：

1. **ORP (一次性可编程ROM)**: 可以编程一次，之后就只能读取。
2. **PROM (可编程ROM)**: 可以通过特殊设备进行一次编程。
3. **EROM (可擦除可编程ROM)**: 可以通过紫外线擦除，然后重新编程。
4. **EEROM (电可擦除可编程ROM)**: 通过电信号擦除和编程。
5. **Flash Memory**: 一种高密度、低功耗的EEROM，广泛应用于便携式设备。

三、RAM (随机存取存储器) 的特点

RAM是一种可以随机读写的存储器，主要用于存储正在运行的程序和数据。其特点是：

1. **易失性**: 断电后数据丢失。
2. **高速读写**: 读写速度快，适合临时存储频繁访问的数据。
3. **高成本**: 相对于ROM，RAM的成本较高，主要是因为它需要更多的晶体管来存储每一位数据。

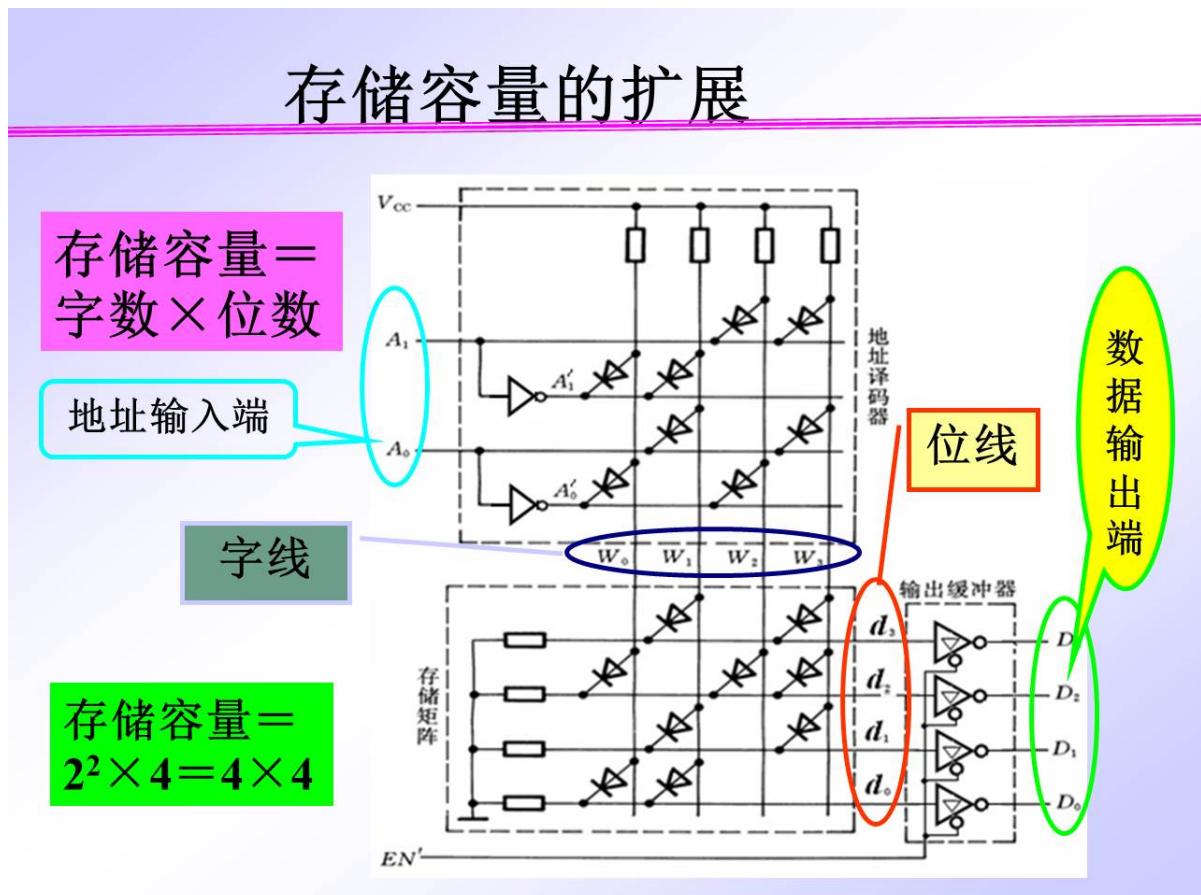
四、容量的计算方法

半导体存储器的容量通常以字节为单位计算。计算公式为：

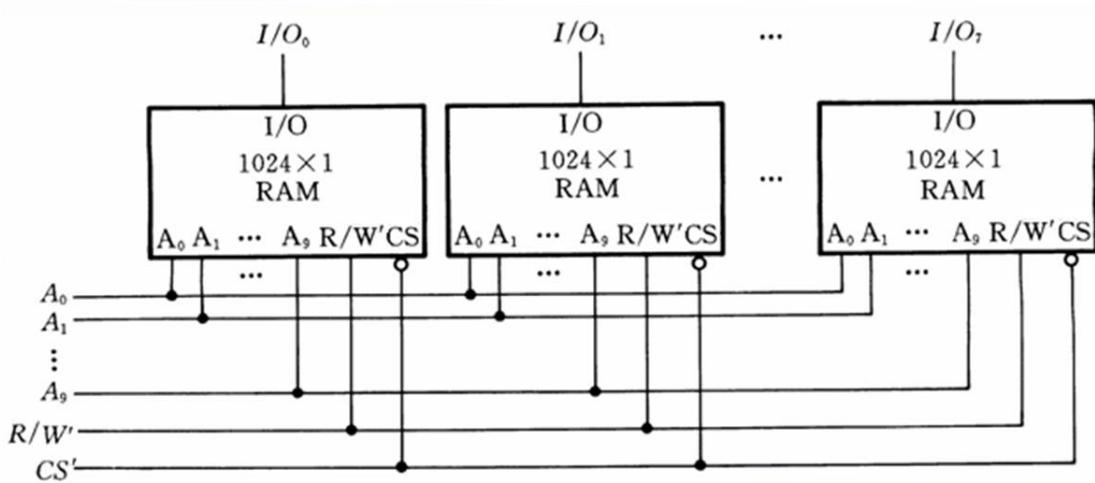
$$\text{容量} = \text{地址线数} \times \text{数据线数}$$

例如，一个具有20个地址线和8个数据线的存储器的容量为：

$$2^{20} \times 8 = 1,048,576 \text{字节(1MB)}$$

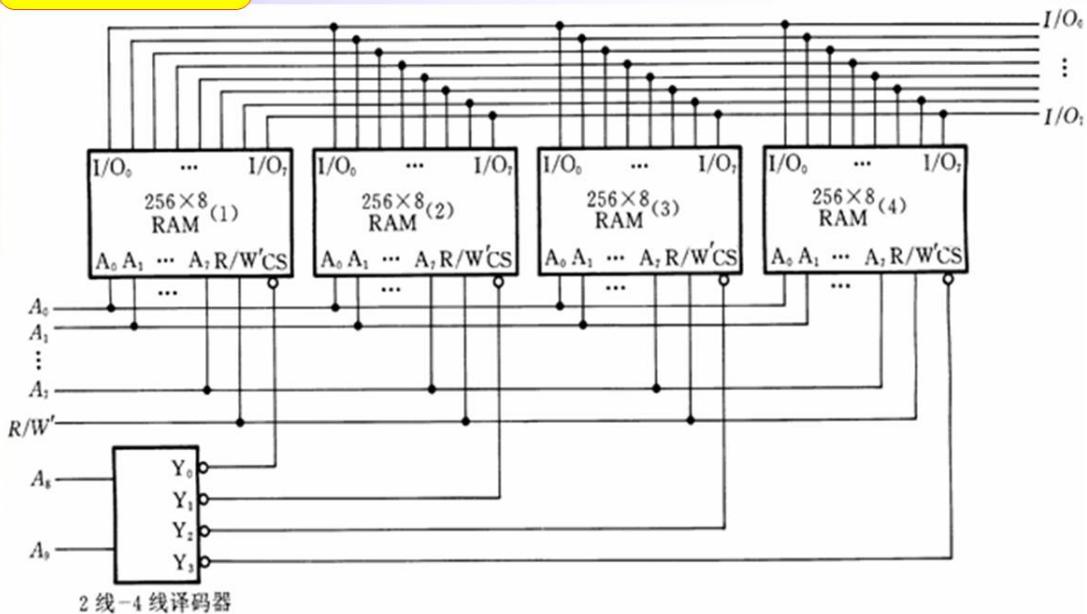


位扩展



8片 1024×1 位RAM接成 1024×8 位的RAM。

字扩展



4片 256×8 位的RAM接成 1024×8 位的RAM。

五、ROM芯片实现组合逻辑函数

ROM芯片除了用于存储数据外，还可以实现组合逻辑函数。这是因为ROM内部的阵列结构本质上是一个查找表，通过编程ROM中的内容，可以实现特定的逻辑功能。例如，使用ROM来实现一个多路复用器或解码器等。

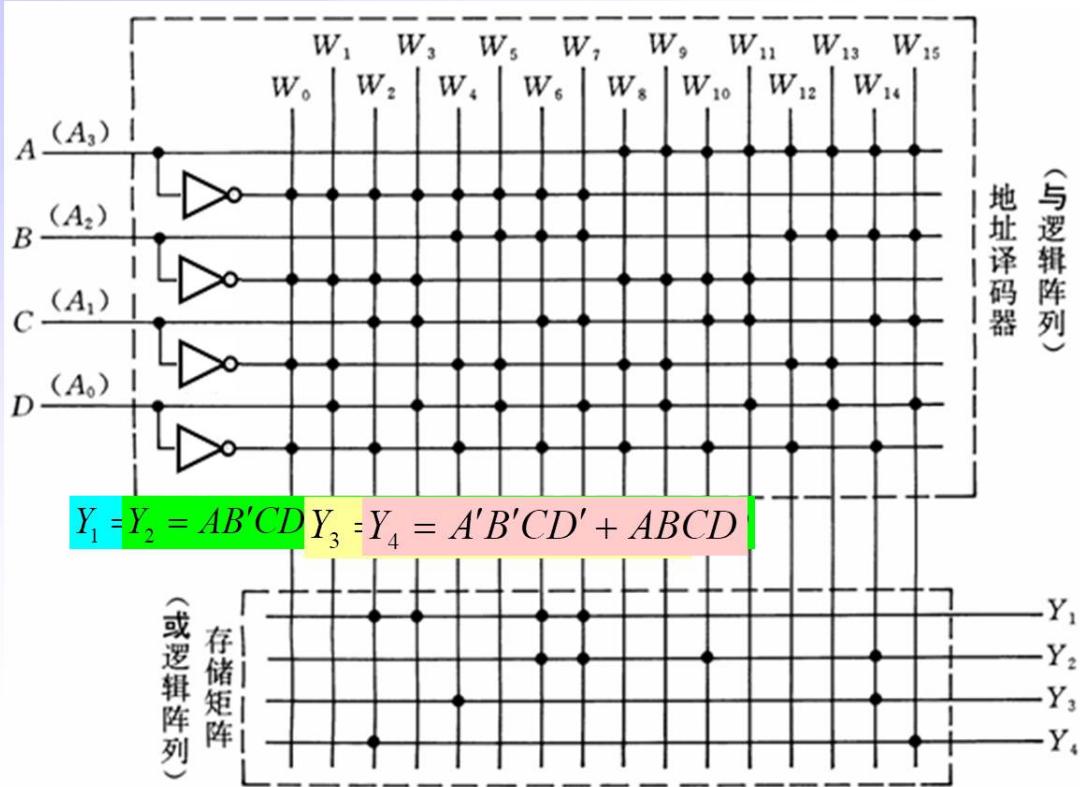
用存储器实现组合逻辑函数

例7.5.2 试用**ROM**产生如下一组多输出逻辑函数

$$\begin{cases} Y_1 = A'BC + A'B'C \\ Y_2 = AB'CD' + BCD' + A'BCD \\ Y_3 = ABCD' + A'BC'D' \\ Y_4 = A'B'CD' + ABCD \end{cases}$$

解：化为最小项之和的形式：

$$\begin{cases} Y_1 = A'BCD' + A'BCD + A'B'CD' + A'B'CD \\ Y_2 = AB'CD' + A'BCD' + ABCD' + A'BCD \\ Y_3 = ABCD' + A'BC'D' \\ Y_4 = A'B'CD' + ABCD \end{cases}$$



综上所述，理解半导体存储器的特点及其分类，尤其是ROM和RAM的区别和应用，对于设计和优化计算机系统至关重要。同时，利用ROM芯片实现组合逻辑函数提供了一种灵活的硬件设计方法。

第六章 时序逻辑电路

1. 掌握时序逻辑电路的分析步骤

在分析同步时序电路时，通常遵循以下步骤：

1. 写出每个触发器的驱动方程：

- **定义驱动方程：**观察给定的逻辑图，确定每个触发器（如D触发器、JK触发器等）的输入信号，并写出表达这些信号之间逻辑关系的逻辑函数式。
- **解释驱动方程的作用：**驱动方程用于描述存储电路中，每个触发器的输入是如何通过组合逻辑被驱动的，是理解电路行为的基础。

2. 得出状态方程组：

- **利用触发器的特性方程：**将驱动方程代入到相应触发器的特性方程中，得到每个触发器的状态方程。
- **组成状态方程组：**所有触发器的状态方程共同构成了整个时序电路的状态方程组，描述了电路在所有可能状态下的行为。

3. 写出电路的输出方程：

- **定义输出方程：**根据逻辑图，确定电路的输出变量与各个触发器状态之间的关系，并写出输出方程。
- **解释输出方程的意义：**输出方程直接决定了电路的输出行为，是连接电路内部状态和外部表现的桥梁。

时序逻辑电路的分析步骤包括**写出各触发器的控制函数、结合特性方程和控制函数写出状态方程、写出电路的输出函数等**。下面将具体分析时序逻辑电路的分析步骤：

1. 写出各触发器的控制函数

- **定义控制函数：**根据电路图，确定每个触发器的输入端（如、K端对于JK触发器）的表达式，这些表达式就是控制函数。
- **分析控制函数的作用：**控制函数决定了触发器在下一个时钟周期的状态，是分析时序逻辑电路的关键步骤之一。

2. 结合特性方程和控制函数写出状态方程

- **利用触发器的特性方程：**特性方程描述了触发器现态与次态之间的逻辑关系。例如，JK触发器的特性方程为 $Q_{n+1} = J Q_n + K \bar{Q}_n$ 。
- **结合控制函数得出状态方程：**将控制函数代入到特性方程中，得到每个触发器具体的状态转换逻辑，即状态方程。

3. 写出电路的输出函数

- **定义输出函数：**如果电路有输出变量，需要根据电路图确定输出变量与触发器状态之间的关系，形成输出函数。
- **分析输出函数的影响：**输出函数直接关联了电路的输出行为，对于理解电路功能至关重要。

4. 作状态转换表

- **列出状态转换表：**以当前状态为输入，使用状态方程和输出函数来填充状态转换表，展现所有可能的状态转换及其对应的输出。
- **分析状态转换表的意义：**该表格是理解电路动态行为的有力工具，为绘制状态转换图提供了基础。

5. 作状态转换图

- **绘制状态转换图：**基于状态转换表，用图形化的方式表示状态之间的转换关系，每个状态是一个节点，状态转换用箭头表示。
以圆圈表示电路各个状态，以箭头表示状态转换的方向。将输入变量取值写在斜线以上，将输出值写在斜线一下
- **解释状态转换图的作用：**状态转换图直观显示了电路的状态空间和状态之间的合法转换路径，有助于理解电路的逻辑功能。

6. 画出时序波形图

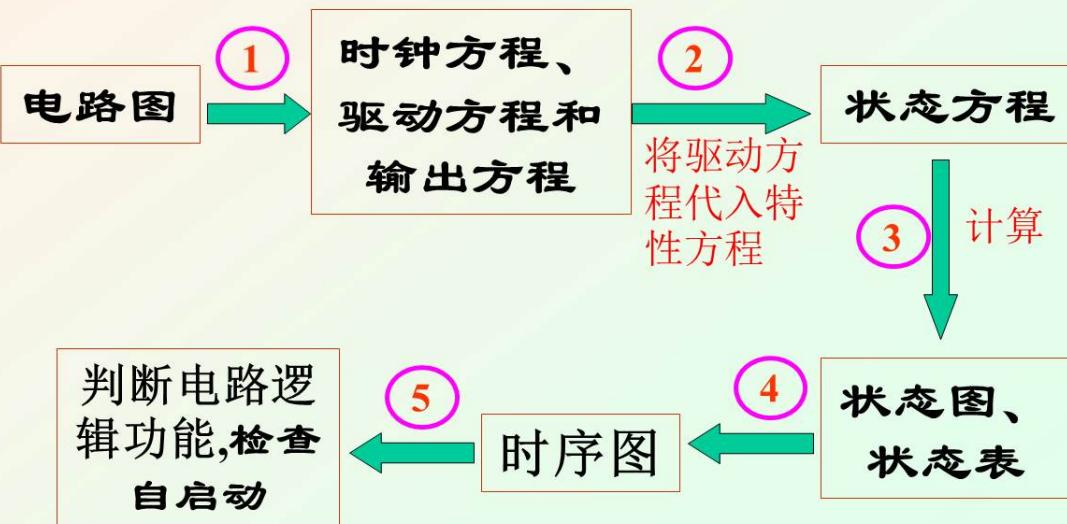
- **根据状态转换表和触发方式画时序波形图：**描述各个触发器状态随时间变化（通常是时钟脉冲）的波形图，特别注意触发器的触发边沿（上升沿或下降沿触发）。
- **分析时序波形图的重要性：**波形图揭示了电路操作的时间特性，对于验证电路设计和理解电路行为具有重要意义。

7. 观察状态转换图并说明电路功能

- **通过状态转换图推断电路功能：**分析状态转换图中的转换路径和循环，从而推测出电路的实际工作逻辑和功能。
- **总结电路功能：**此步骤将整个分析过程具体化为对电路功能的准确描述，完成整个分析过程的目标。

6.2 时序逻辑电路的分析方法

时序电路的分析步骤：



几个概念

有效状态：在时序电路中，凡是被利用了的状态。

有效循环：有效状态构成的循环。

无效状态：在时序电路中，凡是沒有被利用的状态。

无效循环：无效状态若形成循环，则称为无效循环。

自启动：在CLK作用下，无效状态能自动地进入到有效循环中，则称电路能自启动，否则称不能自启动。

■ 同步时序逻辑电路分析的一般步骤

找方程

{ 1、从给定的逻辑图中写出每个触发器的驱动方程；
2、把得到的驱动方程代入相应触发器的特性方程，得出每个触发器的状态方程（组）；
3、根据逻辑图写出电路的输出方程；

画图表

{ 4、列出该电路的状态转换表；
5、根据状态表画出状态转换图（或时序图）；
6、必要时画出电路时序图（仿真或实验）；

得结论

{ 根据状态转换图、表描述电路的逻辑功能，并进行自启动验证。

2. 掌握一般设计方法以及置零法和置数法设计计数器

【【数电】任意进制计数器的改接方法的解题技巧速成（74hc160和161置数和复位法）】https://www.bilibili.com/video/BV16T4y1h7Kn/?share_source=copy_web&vd_source=440c7ec5d64e62c0d02675282b15de02

三、任意进制计数器的构成方法

表1 几种常用的中规模集成（MSI）计数器的主要功能列表

型号	主要功能
74161	“异步清零”，“同步置数”的同步模16加法计数器
74163	“同步清零”，其余同74161
74LS191	可“异步置数”的单时钟同步16进制加/减计数器
74LS193	可“异步清零”，“异步置数”的双时钟同步16进制加/减计数器
74160	同步模10计数器，其余同74161
74190	同步10进制计数器，其余同74191
74192	模10同步可逆计数器，其余同74193
54/74LS196	可“异步清零”，“同步置数”的二—五—十进制同步计数器
74LS290	二—五—十进制异步计数器

三、任意进制计数器的构成方法

利用现有的N进制计数器构成任意进制（M）计数器时，如果 $M < N$ ，则只需一片N进制计数器；如果 $M > N$ ，则要多片N进制计数器。

三、任意进制计数器的构成方法

当M<N时，一片N进制计数器即可实现

实现方法 {

- 置零法（复位法）
- 置数法（置位法）

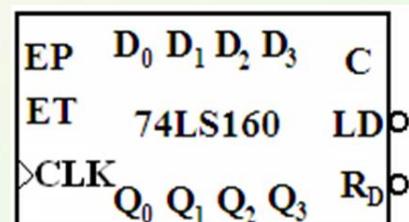
置零法：适用于有清零输入端的集成计数器。利用计数器的清零端的清零作用，截取计数过程中的某个中间状态控制清零端，使计数器由此状态返回到零并重新开始计数。

置零法：适用于有清零输入端的集成计数器。原理是不管输出处于哪一状态，只要在清零输入端加一有效电平电压，输出会立即从那个状态回到0000状态，清零信号消失后，计数器又可以从0000开始重新计数。

置数法：适用于具有预置功能的集成计数器。利用计数器的置数端的置数作用，给计数器置入某个数值，使之跳过 $N-M$ 个状态，从而获得 M 进制的计数器。

置数法：适用于具有预置功能的集成计数器。对于具有预置数功能的计数器而言，在其计数过程中，可以将它输出的任意一个状态通过译码，产生一个预置数控制信号反馈至预置数控制端，在下一个CLK脉冲作用后，计数器会把预置数输入端 $D_0D_1D_2D_3$ 的状态置入输出端。预置数控制信号消失后，计数器就从被置入的状态开始重新计数。

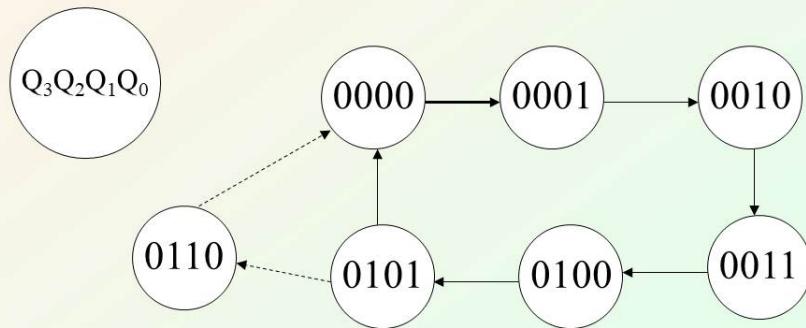
例6.3.2 试用同步十进制计数器74LS160接成同步六进制计数器，160的逻辑功能示意图如下图，功能表如下表。



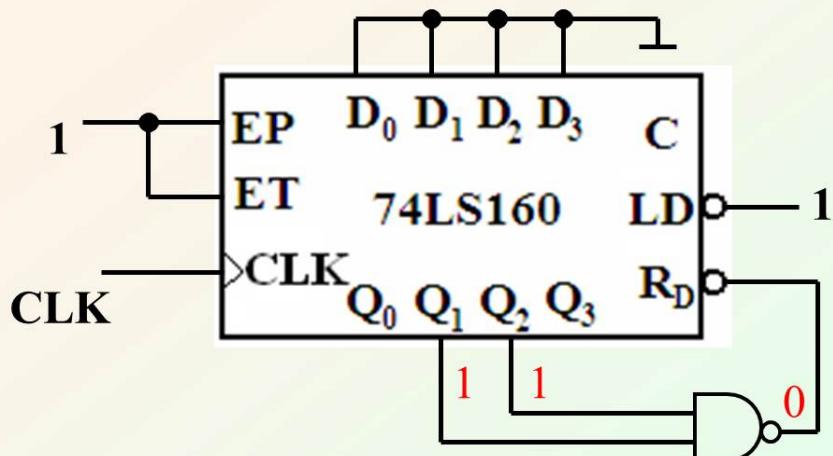
CLK	R'_D	LD'	EP	ET	工作状态
×	0	×	×	×	清零
↑	1	0	×	×	预置数
×	1	1	0	1	保持
×	1	1	×	0	保持($C=0$)
↑	1	1	1	1	计数

解：

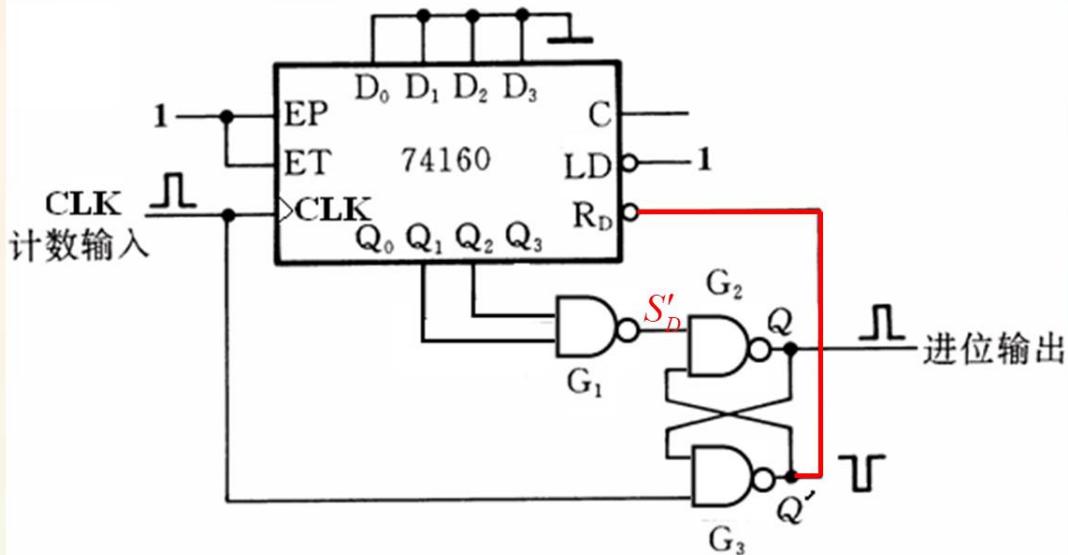
置零法 74LS160具有异步清零功能



$$R'_D = (Q_2 \cdot Q_1)'$$



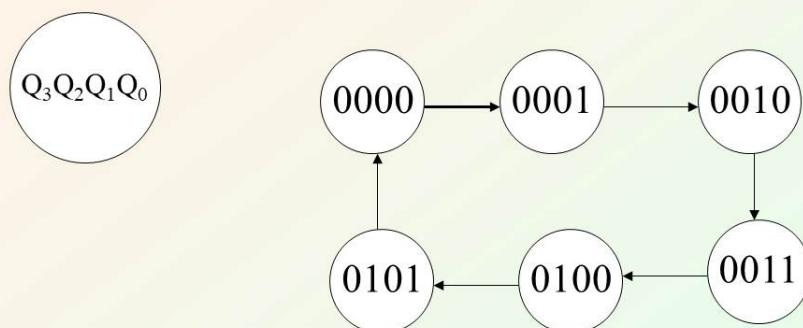
当计数器记成 $Q_3Q_2Q_1Q_0 = 0110$ 时，与非门输出低电平信号给 R'_D 端，将计数器置零。置零信号不是一个稳定的状态，持续时间很短，有可能导致电路误动作。



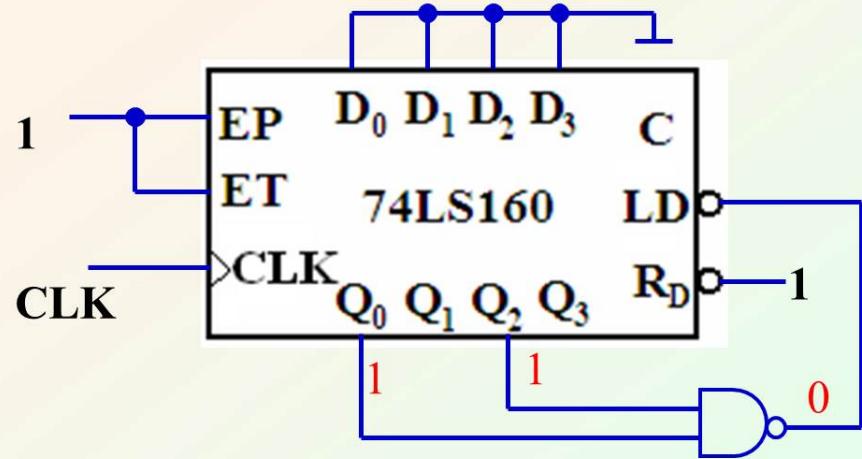
改进电路

置数法

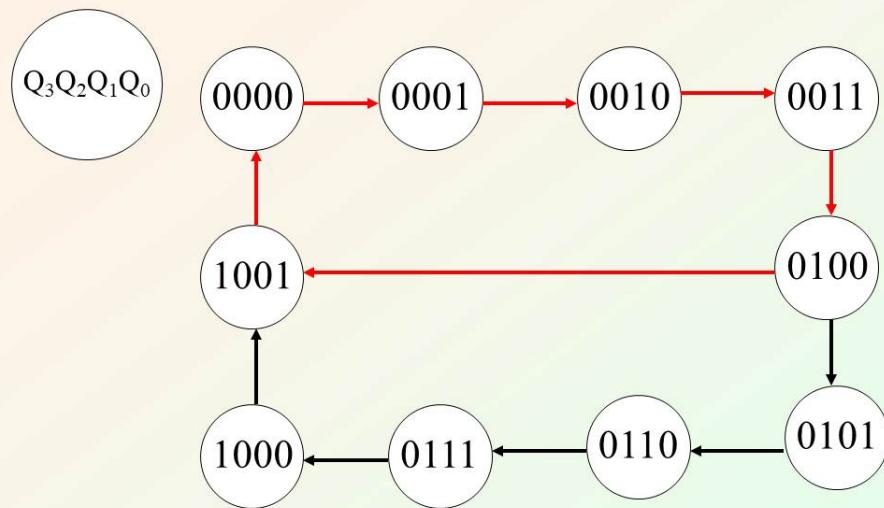
74LS160具有同步置数功能

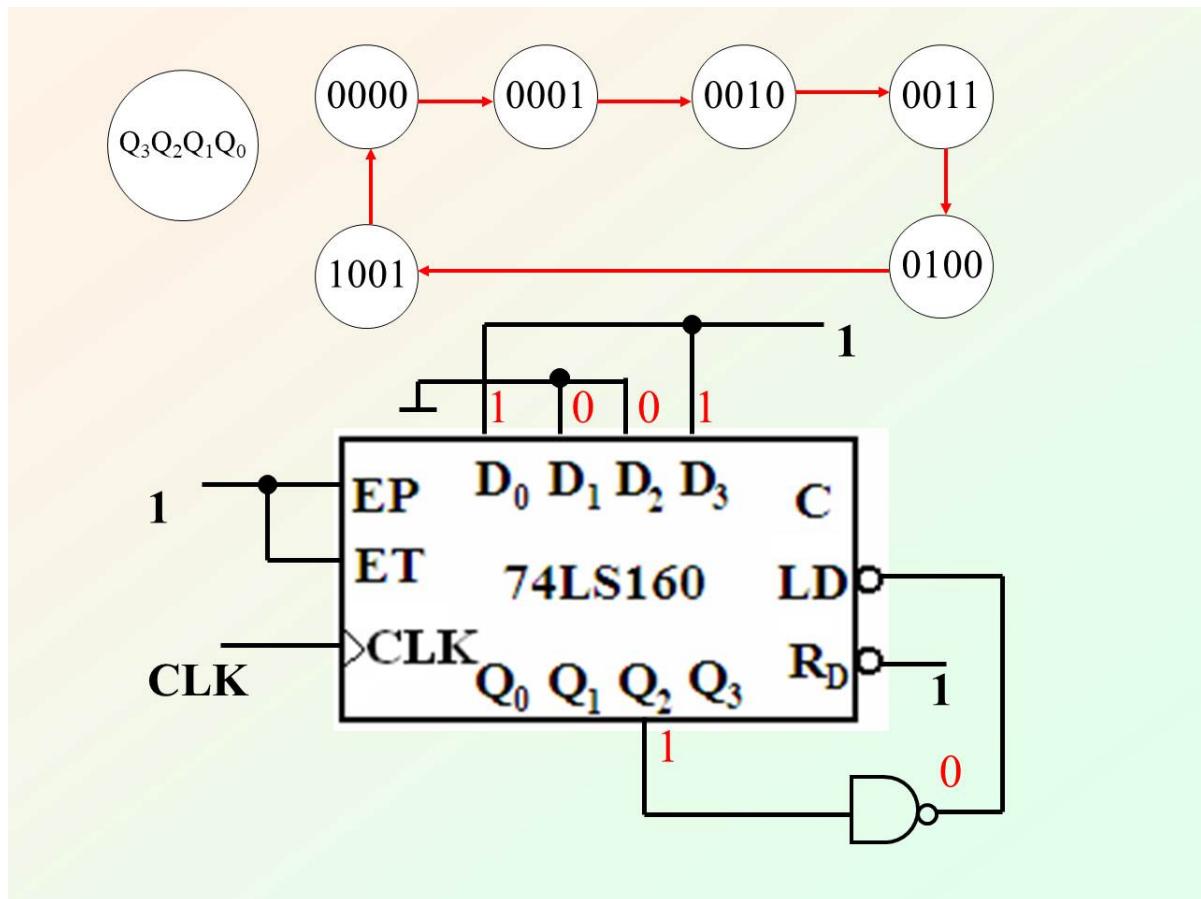


$$LD' = (Q_2 \cdot Q_0)'$$

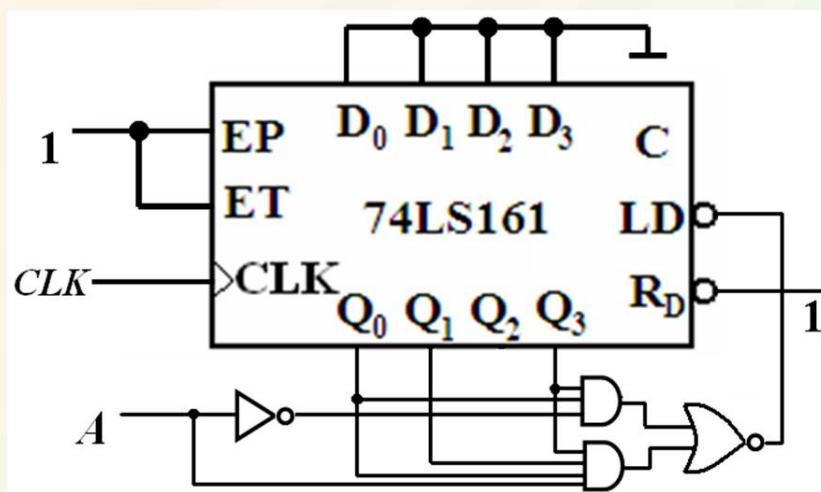


$LD'=0$ 后，还要等下一个CLK信号到来时才置入数据，
而这时 $LD'=0$ 的信号以稳定地建立了，提高了可靠性。





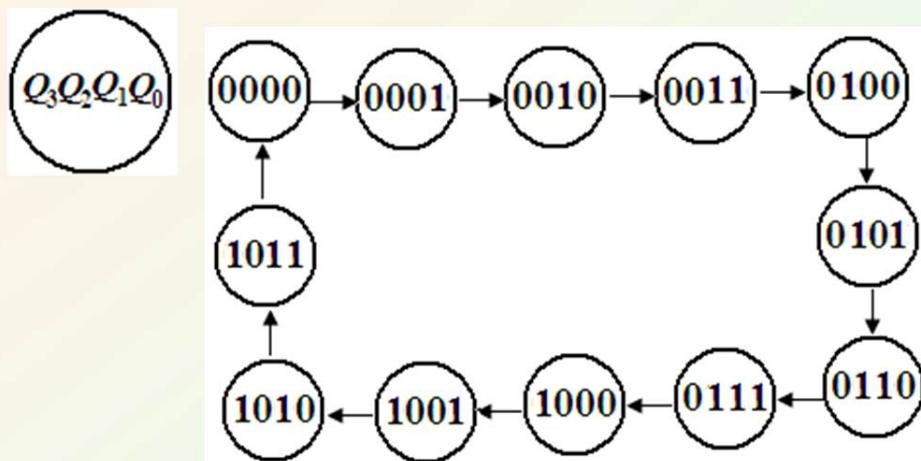
P350 题6.15



解: $LD' = (Q_3 Q_0 A' + Q_3 Q_1 Q_0 A)'$

当A=1时 $LD' = (Q_3Q_1Q_0)'$

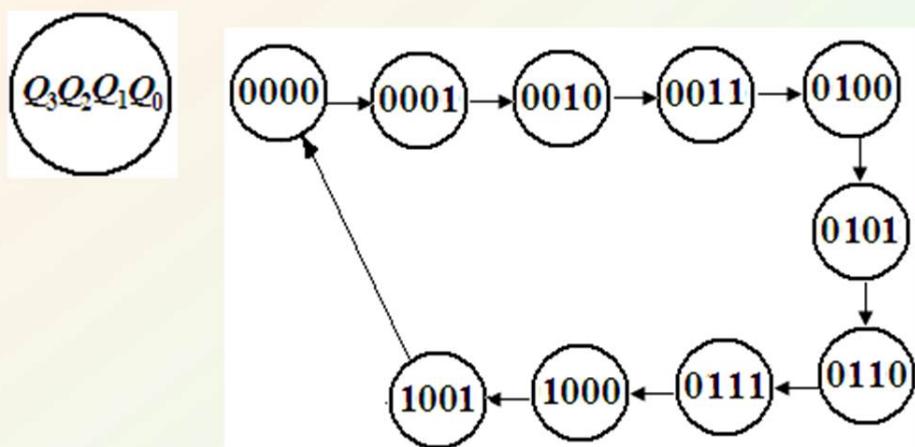
其状态转换图如下：



构成十二进制计数器

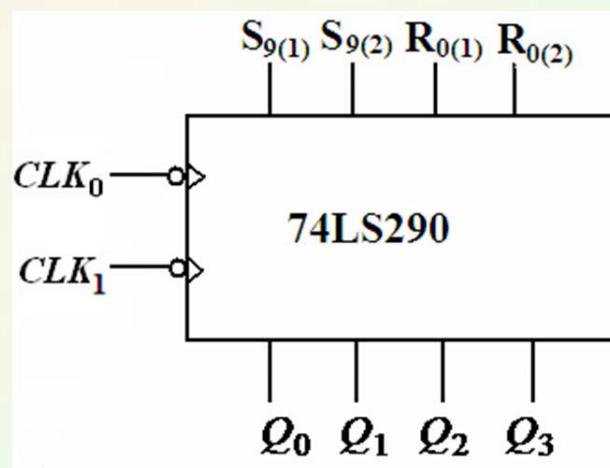
当A=0时 $LD' = (Q_3Q_0)'$

其状态转换图如下：



构成十进制计数器

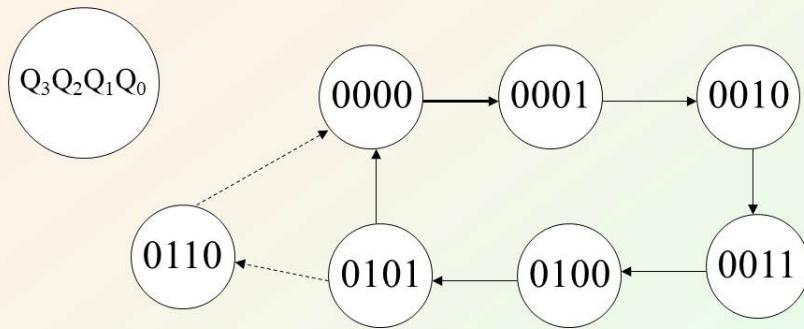
例:用集成异步二—五—十进制计数器74LS290接成六进制计数器(模六)。(不用其他元件)。已知74LS290的逻辑示意图和功能表。



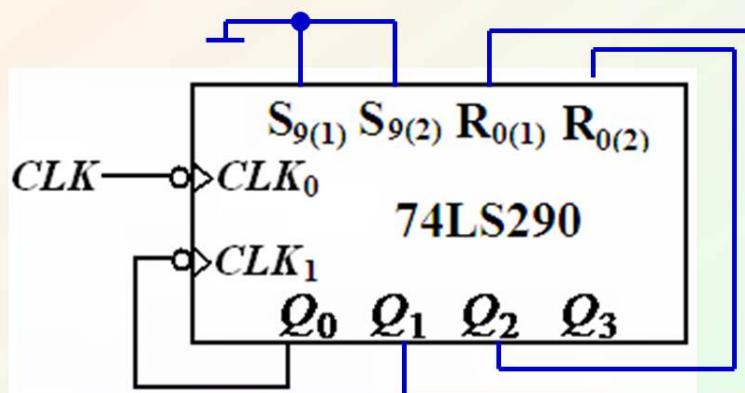
74LS290功能表

输入				输出			
$R_{0(1)} \cdot R_{0(2)}$	$S_{9(1)} \cdot S_{9(2)}$	CLK_0	CLK_1	Q_3	Q_2	Q_1	Q_0
1	0	×	×	0	0	0	0
x	1	×	×	1	0	0	1
0	0	CLK	0	二进制计数			
0	0	0	CLK	五进制计数			
0	0	CLK	Q_0	8421码十进制计数			

置零法构成六进制 74LS290具有异步清零功能



首先将74LS290接成8421BCD码的十进制计数器，即将 CLK_1 与 Q_0 相连， CLK_0 作为外部计数脉冲 CLK 。



3. 掌握同步时序逻辑的特点

同步时序逻辑电路的特点是所有存储元件都受同一个时钟信号统一控制，且仅在该时钟信号到来时，存储元件的状态才能发生变化，从而使电路的输出也发生相应的变化。

同步时序逻辑电路的结构主要由组合电路和存储电路组成，通过反馈回路将两者连接在一起。这种结构使得电路具有记忆功能，能够根据输入信号和当前状态生成相应的输出信号。由于所有存储元件（通常是触发器）都受同一个时钟信号的控制，因此它们的状态更新是同步进行的。这种特性保证了电路的简单性和可靠性，但同时也带来了对时钟信号分布和功率消耗的挑战。

同步时序逻辑电路的设计和分析需要遵循一定的步骤和标准。设计过程通常包括逻辑抽象、状态化简、状态分配和选定触发器类型等步骤。这些步骤旨在确保所设计的电路不仅能够实现预期的逻辑功能，而且是简单、经济和可靠的。在分析和设计过程中，常用的工具包括状态转换图、状态转换表、逻辑函数表达式以及时序图等。

此外，同步时序逻辑电路的输出特征可以分为米利型（Mealy）和穆尔型（Moore）两种。米利型电路的输出不仅取决于存储电路的状态，还取决于当前输入；而穆尔型电路的输出则完全取决于存储电路的状态。这两种类型的电路在实际应用中有不同的适用场景，需要根据具体需求进行选择。

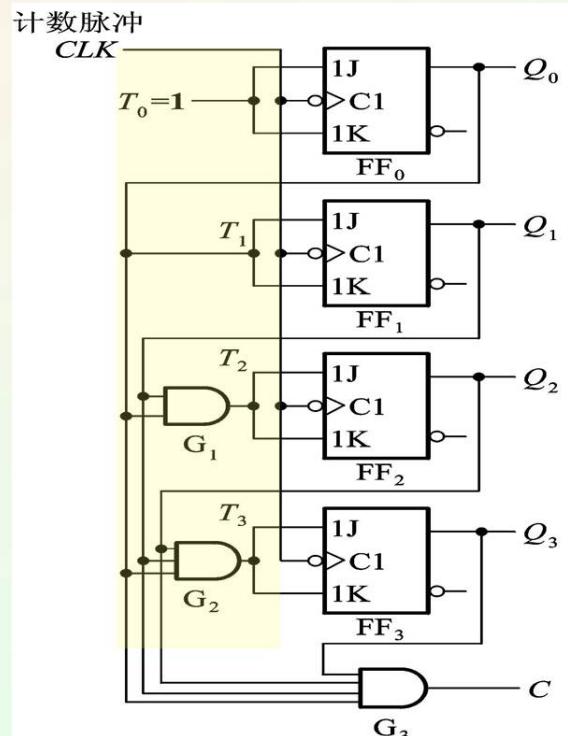
综上所述，同步时序逻辑电路的特点可以归纳为：由统一时钟信号控制、结构和操作简单、具备记忆功能、并且存在特定的设计和分析方法。这些特点使得同步时序逻辑电路在数字电子技术中得到了广泛的应用。

4. 掌握同步二进制计数器的连接规律，给出逻辑电路图，能够通过时序逻辑电路的分析方法画出同步二进制计数器的时序图

一、同步计数器

同步二进制计数器

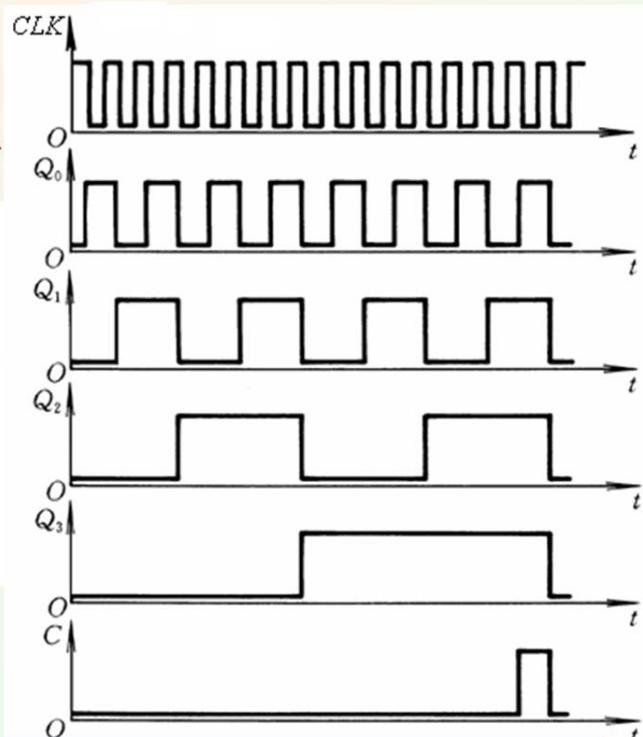
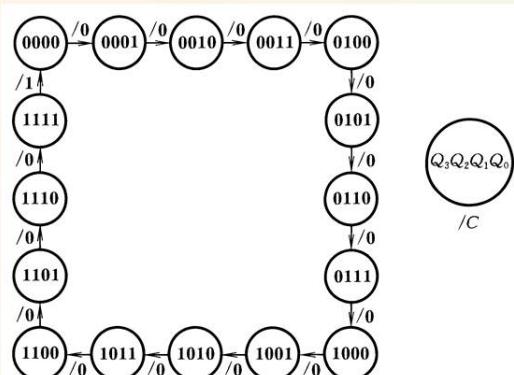
- ① 同步二进制加法计数器
- 原理：根据二进制加法运算规则可知：在多位二进制数末位加1，若第*i*位以下皆为1时，则第*i*位应翻转。而最低位的状态在每次加一时都要改变。
- 由此得出规律，若用T触发器构成计数器，则第*i*位触发器输入端 T_i 的逻辑式应为： $T_i = Q_{i-1}Q_{i-2}\dots Q_0$
 $T_0 \equiv 1$



n位二进制同步加法计数器的电路连接规律:

$$\left\{
 \begin{array}{l}
 \text{驱动方程} \\
 \left\{ \begin{array}{l}
 J_0 = K_0 = 1 \\
 J_1 = K_1 = Q_0 \\
 J_2 = K_2 = Q_1 Q_0 \\
 \dots\dots \\
 J_{n-1} = K_{n-1} = Q_{n-2} Q_{n-3} \cdots Q_1 Q_0
 \end{array} \right. \\
 \text{输出方程 } C = Q_{n-1} Q_{n-2} \cdots Q_1 Q_0
 \end{array} \right.$$

4位二进制同步加法计数器

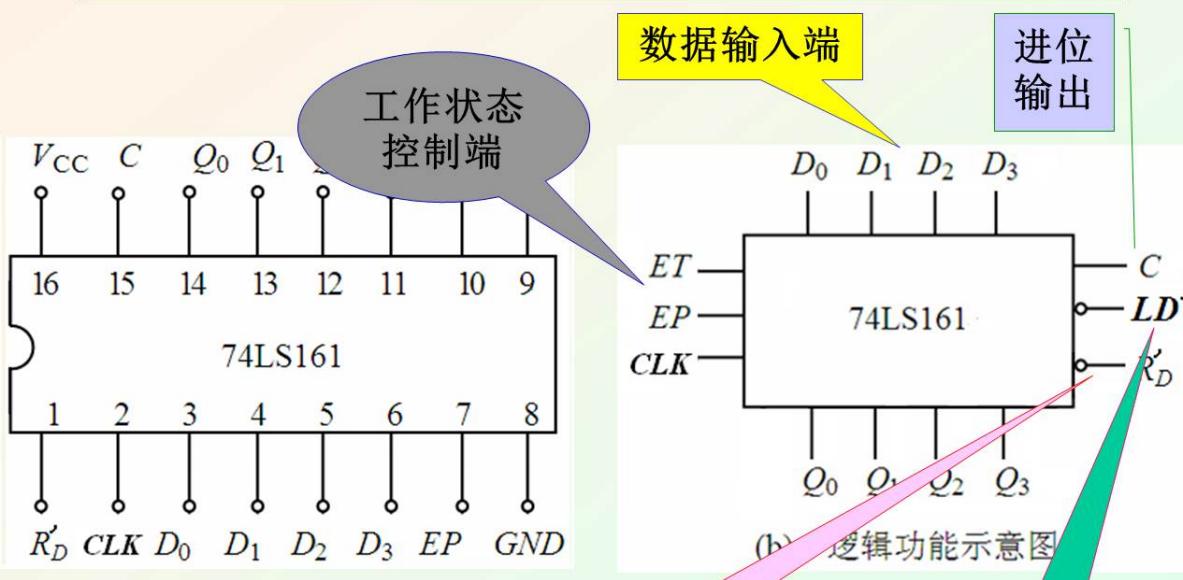


若计数脉冲频率为 f_0 , 则 Q_0 、 Q_1 、 Q_2 、 Q_3 端输出脉冲的频率依次为 f_0 的 $1/2$ 、 $1/4$ 、 $1/8$ 、 $1/16$ 。因此又称为分频器。

重要定义：

相对于时钟频率而言，二进制计数器各级输出依次称为二分频、四分频、八分频、十六分频。计数器中能计到的最大数称为计数长度或计数容量， n 位二进制计数器的计数容量为 2^n-1 ，而称计数器的状态总数 2^n 为计数器的模（也称循环长度）。

4位集成二进制同步加法计数器74LS161/163



(a)引脚排列图

74LS161

当 $R'D=0$ 时所有触发器将同时被置零，而且置零操作不受其他输入状态的影响

异步清零是指不依赖于时钟信号的清零操作。当满足某种特定条件时，计数器立即归零，而不需要等待下一个时钟周期的到来

同步置数则需要等待时钟的有效沿到来才能完成操作。当需要将计数器设置到一个特定值时，必须等到下一个时钟脉冲到来才能实现

4位同步二进制计数器74161功能表

CLK	R'_D	LD'	EP	ET	工作状态
×	0	×	×	×	清零
↑	1	0	×	×	预置数
×	1	1	0	1	保持
×	1	1	×	0	保持 (C=0)
↑	1	1	1	1	计数

74161具有异步清零和同步置数功能。

74LS163

同步清零是指在特定的时钟脉冲到来时，将寄存器或计数器的值清零的操作方式。它需要与时钟脉冲的上升沿或下降沿同步进行。

同步置数则需要等待时钟的有效沿到来才能完成操作。当需要将计数器设置到一个特定值时，必须等到下一个时钟脉冲到来才能实现

4位同步二进制计数器74163功能表

CLK	R'_D	LD'	EP	ET	工作状态
↑	0	×	×	×	清零
↑	1	0	×	×	预置数
×	1	1	0	1	保持
×	1	1	×	0	保持 ($C=0$)
↑	1	1	1	1	计数

74163具有同步清零和同步置数功能.

74LS163的引脚排列和74LS161相同，不同之处是74LS163采用同步清零方式。

n 位二进制同步减法计数器的连接规律：

{
驱动方程

$$\begin{cases} J_0 = K_0 = 1 \\ J_1 = K_1 = Q'_0 \\ J_2 = K_2 = Q'_1 Q'_0 \\ \dots \\ J_{n-1} = K_{n-1} = Q'_{n-2} Q'_{n-3} \cdots Q'_1 Q'_0 \end{cases}$$

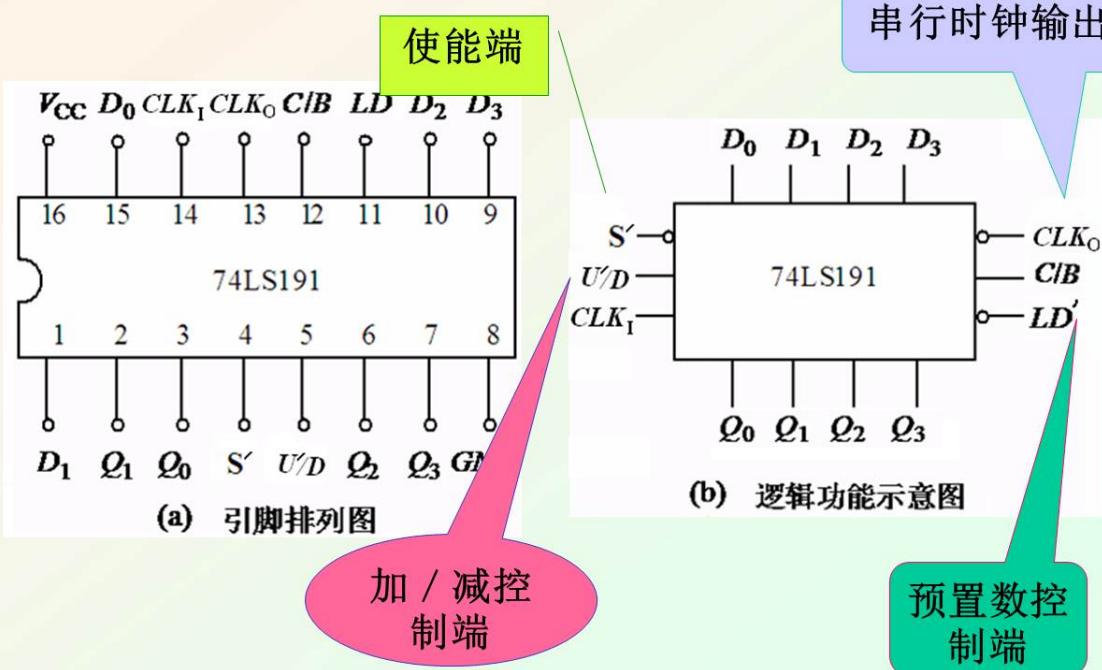
}
输出方程

$B = Q'_{n-1} Q'_{n-2} \cdots Q'_1 Q'_0$

74LS191

异步置数是指在不依赖时钟信号的条件下，将数据直接置入寄存器或计数器的操作方式。

4位集成二进制同步可逆计数器74LS191



4位同步二进制可逆计数器74LS191功能表

CLK_I	S'	LD'	U'/D	工作状态
×	1	1	×	保持
×	×	0	×	预置数
↑	0	1	0	加法计数
↑	0	1	1	减法计数

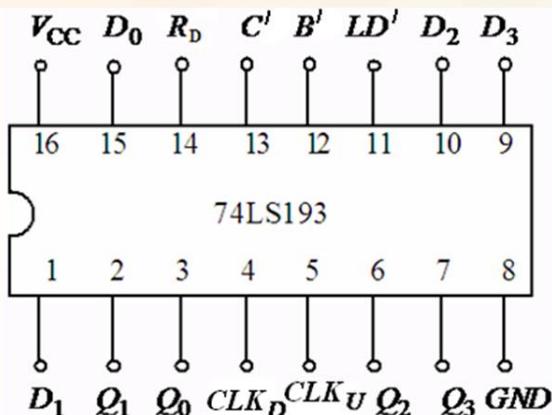
74LS191具有异步置数功能。

74LS193

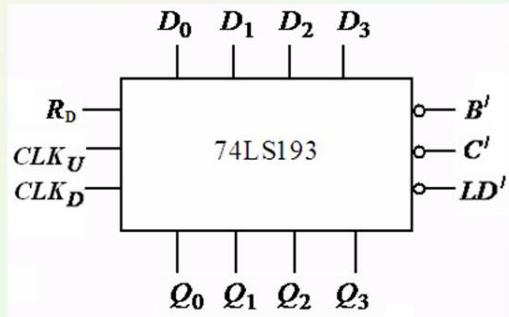
异步清零是指不依赖于时钟信号的清零操作。当满足某种特定条件时，计数器立即归零，而不需要等待下一个时钟周期的到来。

异步置数是指在不依赖时钟信号的条件下，将数据直接置入寄存器或计数器的操作方式。

双时钟加/减计数器74LS193



(a) 引脚排列图



(b) 逻辑功能示意图

74LS193具有异步清零和异步置数功能。

2、同步十进制计数器

同步十进制加法计数器:在同步二进制加法计数器基础上修改而来。

同步十进制加法计数器74LS160与74LS161逻辑图和功能表均相同,所不同的是74LS160是十进制而74LS161是十六进制。

同步十进制可逆计数器也有单时钟和双时钟两种结构形式。属于单时钟的有74LS190等，属于双时钟的有74LS192等。

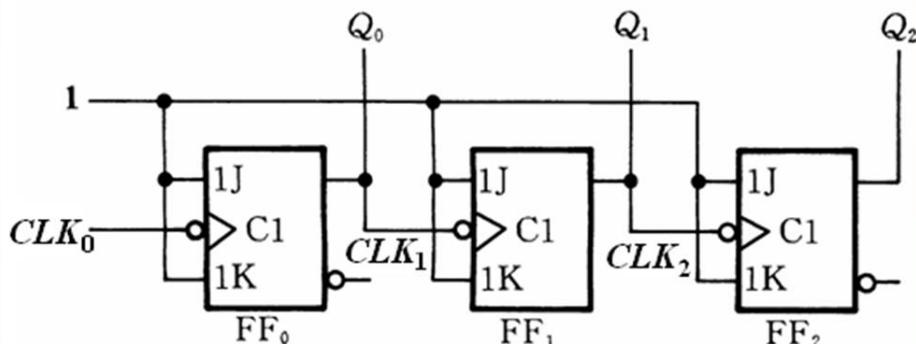
74LS190与74LS191逻辑图和功能表均相同；

74LS192与74LS193逻辑图和功能表均相同。

5. 异步二进制计数器的连接规律，给出逻辑电路图，能够通过时序逻辑电路的分析方法画出异步二进制计数器的时序图

1、异步二进制计数器

3位异步二进制加法计数器

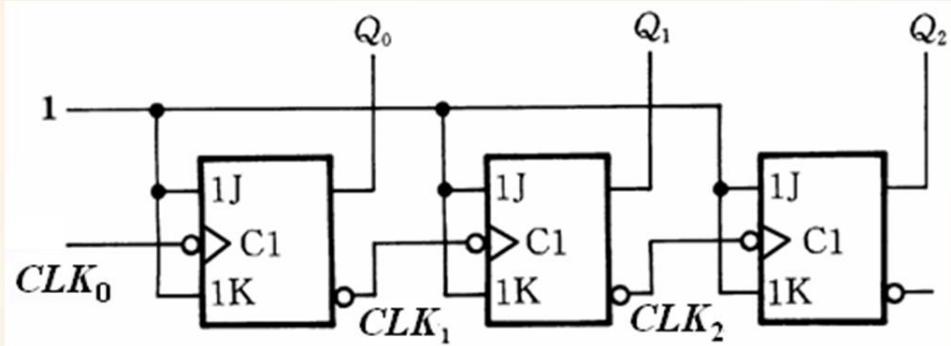


$$\left\{ \begin{array}{l} J_0 = K_0 = 1 \\ J_1 = K_1 = 1 \\ J_2 = K_2 = 1 \end{array} \right.$$

触发器为下降沿触发，
 Q_0 接 CLK_1 , Q_1 接 CLK_2 。
若上升沿触发，则应
 Q_0' 接 CLK_1 , Q_1' 接 CLK_2 。

3位异步二进制减法计数器

3位异步二进制减法计数器



$$\left\{ \begin{array}{l} J_0 = K_0 = 1 \\ J_1 = K_1 = 1 \\ J_2 = K_2 = 1 \end{array} \right.$$

触发器为下降沿触发, Q'_0 接 CLK_1 , Q'_1 接 CLK_2 。
若上升沿触发, 则应 Q_0 接 CLK_1 , Q_1 接 CLK_2 。

2、异步十进制计数器

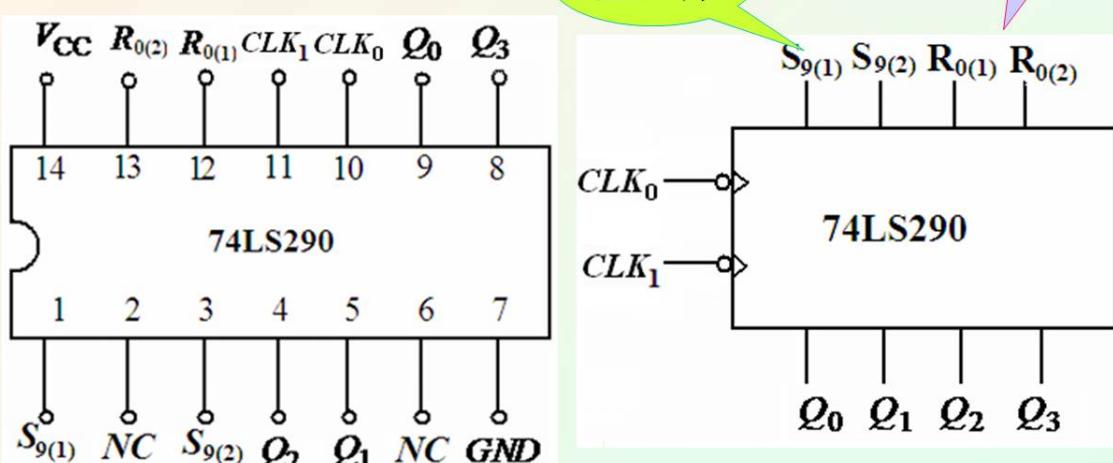
74LS290

2、异步十进制计数器

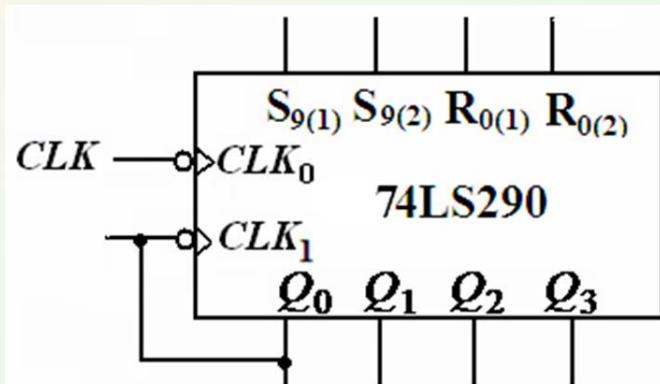
异步二—五—十进制计数器74LS290

置0端

置9端



若计数脉冲由 CLK_0 端输入，输出由 Q_0 端引出，即得到二进制计数器；若计数脉冲由 CLK_1 端输入，输出由 $Q_1 \sim Q_3$ 引出，即是五进制计数器；若将 CLK_1 与 Q_0 相连，同时以 CLK_0 为输入端，输出由 $Q_0 \sim Q_3$ 引出，则得到8421码十进制计数器。



74LS290功能表

输入				输出			
$R_{0(1)} \cdot R_{0(2)}$	$S_{9(1)} \cdot S_{9(2)}$	CLK_0	CLK_1	Q_3	Q_2	Q_1	Q_0
1	0	×	×	0	0	0	0
×	1	×	×	1	0	0	1
0	0	CLK	0	二进制计数			
0	0	0	CLK	五进制计数			
0	0	CLK	Q_0	8421码十进制计数			

异步计数器特点

优点：结构简单

缺点：（1）工作频率较低；
（2）在电路状态译码时存在竞争
—冒险现象。

第七章 脉冲波形的产生与整形

掌握施密特触发器、单稳态触发器、多谐振荡器的特点

施密特触发器

施密特触发器 (Schmitt Trigger)，它是脉冲波形变换中经常使用的一种电路。它在性能上有两个重要的特点：

第一，输入信号从低电平上升的过程中，电路状态转换时对应的输入电平，与输入信号从高电平下降过程中对应的输入转换电平不同。这种现象称为滞后或回差 (hysteresis)。

第二，在电路状态转换时，通过电路内部的正反馈过程使输出电压波形的边沿变得很陡峭。这种特性使得施密特触发器能够有效地将缓慢变化的输入信号转换为清晰、锐利的输出脉冲。

单稳态电路

单稳态电路 (Monostable Multivibrator，又称 One-shot) 的工作特性具有如下的显著特点：

第一，它有稳态和暂稳态两个不同的工作状态。

第二，在外界触发脉冲作用下，能从稳态翻转到暂稳态，在暂稳态维持一段时间以后，再自动返回稳态。

第三，暂稳态维持时间的长短取决于电路本身的参数，与触发脉冲的宽度和幅度无关。

由于具备这些特点，单稳态电路被广泛应用于脉冲整形、延时（产生滞后于触发脉冲的输出脉冲）以及定时（产生固定时间宽度的脉冲信号）等。

多谐振荡电路

多谐振荡电路 (Astable Multivibrator) 是一种自激振荡电路，在接通电源以后，不需要外部触发信号，便能自动地产生矩形脉冲。由于矩形波中含有丰富的高次谐波分量，所以习惯上将矩形波振荡电路称为多谐振荡电路。多谐振荡器可以产生连续的矩形波信号，常用于时钟脉冲发生器和定时控制电路中。

第八章 A/D和D/A转换器

掌握D/A转换基本原理和电路结构分类；掌握A/D转换的基本原理 采样、量化编码

本章小结

随着微处理器和微型计算机在检测、控制和信号处理系统中的广泛应用，A/D（模数转换器）和D/A（数模转换器）转换技术也得到了迅速发展。计算机计算精度和速度的提升对A/D、D/A转换器的转换精度和速度提出了更高要求，推动了转换技术的进步。在许多基于计算机的系统中，系统的精度和速度最终受限于A/D、D/A转换器的转换精度和速度，这两个指标是转换器最重要的性能参数，也是讨论的重点。

A/D、D/A转换器种类繁多，无法逐一列举。因此，理解其基本概念、共同问题及分类原则是必要的。在D/A转换器中我们分别介绍了权电阻网络型、权电流型、倒T形电阻网络型、权电容网络型以及开关树型的D/A转换器。这几种电路在集成D/A转换器产品中都有应用。目前在双极型的D/A转换器产品中权电流型电路用得比较多；在CMOS集成D/A转换器中则以倒T形电阻网络和开关树型电路较为常见。

A/D转换器分为直接和间接两大类。直接A/D转换器包括并联比较型和逐次逼近型两种电路。并联比较型A/D转换器（闪速A/D转换器）转换速度最快，但因电路规模庞大，主要用于超高速应用。逐次逼近型A/D转换器在速度和电路规模上平衡较好，因此在集成A/D转换器产品中使用最广泛。

间接A/D转换器包括双积分型（V- Σ 变换型）、 Σ - Δ 型和V-F变换型。双积分型AD转换器虽速度低，但结构简单、性能稳定，适用于低速系统如数字万用表。V-F变换型和E-A型A/D转换器分别在遥测遥控系统和音频信号传输中得到广泛应用，因为它们具有较强的抗干扰能力。

为达到高转换精度，除了选用高分辨率的A/D、D/A转换器外，还需确保参考电源和供电电源稳定，并减少环境温度变化。否则，即使使用高分辨率芯片也难以实现预期的转换精度。

D/A转换和A/D转换是数字信号处理领域的核心概念，它们分别涉及将数字信号转换为模拟信号和将模拟信号转换为数字信号。以下是对这两个过程的基本原理、电路结构分类以及A/D转换的基本原理包括采样、量化和编码的详细解释：

D/A转换的基本原理

D/A转换器（Digital-to-Analog Converter）将数字信号转换为模拟信号。其基本原理涉及以下步骤：

1. **数字输入**：接收二进制数字信号。
2. **解码**：将数字信号解码为相应的模拟电平，通常通过切换电流或电压源来实现。
3. **模拟输出**：生成与数字输入成正比的模拟电压或电流。

D/A转换器的电路结构分类

D/A转换器的电路结构可以分为以下几类：

1. **权电阻网络型**：使用一组权电阻，每个电阻的阻值都是前一个电阻的两倍。通过开关的控制，将不同权重的电流汇总到输出端，得到模拟电压。
2. **R-2R梯形电阻网络型**：使用梯形电阻网络，每个电阻的阻值都是2欧姆。通过开关的控制，将不同权重的电流汇总到输出端，得到模拟电压。

3. **权电容网络型**: 使用一组权电容，每个电容的容值都是前一个电容的两倍。通过开关的控制，将不同权重的电荷汇总到输出端，得到模拟电压。
4. **开关树型**: 使用开关和电阻（或电流源）的组合来直接选择与数字输入对应的模拟值。

A/D转换的基本原理

A/D转换器 (Analog-to-Digital Converter) 将模拟信号转换为数字信号。其基本原理包括三个步骤：

1. **采样**: 按照一定的时间间隔（采样周期）对模拟信号进行测量，得到离散的样本值。根据奈奎斯特定理，采样频率应至少为信号最高频率的两倍，以避免混叠现象。
2. **量化**: 将采样得到的连续样本值映射到有限个离散电平上。量化过程会引入量化误差，即量化噪声。
3. **编码**: 将量化后的离散电平转换为数字码字（二进制数）。编码后的数字信号可以用于进一步的处理、存储或传输。

A/D转换器的分类

A/D转换器的分类主要有以下几种：

1. **并联比较型（闪速A/D转换器）**: 使用多个比较器，每个比较器对应一个量化电平。通过一次比较，可以直接得到数字码字。这是目前所有A/D转换器中转换速度最快的一种。
2. **逐次逼近型**: 使用一个比较器和一组权电容（或权电阻）。通过逐次逼近的方式，逐步确定数字码字的每一位。这种类型的转换速度较快，电路规模相对较小。
3. **双积分型**: 属于间接A/D转换器，转换速度较低，但电路结构简单，性能稳定可靠，抗干扰能力较强，适用于低速系统。
4. **V-F变换型和E-A型**: 这些也是间接A/D转换器，多用在遥测、遥控系统和音频信号传输中，具有较强的抗干扰能力。

总的来说，为了获得较高的转换精度，除了选用高分辨率的A/D、D/A转换器外，还必须确保参考电源和供电电源具有足够的稳定性，并减少环境温度的变化。这些措施有助于实现更高的转换精度。

第九章 总章

数字电子技术基础涉及数制和码制的转换、二进制运算以及逻辑代数基础等多个方面。以下是内容的详细叙述：

第一章 数制和码制

在数字电子技术中，了解不同数制之间如何转换是非常重要的。这包括十进制到任意进制的转换、任意进制转十进制，以及二进制与十六进制之间的互相转换。这些转换涉及到整数部分和小数部分的分别处理，其中包含了连续除法和连续乘法的应用。例如，将十进制小数转换为二进制时，需要分别对小数部分进行乘以目标进制基数的操作，并取整数部分作为结果，重复此过程直至小数部分为0或达到所需精度。

第二章 逻辑代数基础

逻辑代数是数字电路设计的基础之一，它包括基本运算（如与运算、或运算和非运算）和常用运算（如与非运算、或非运算等）。掌握这些基础运算有助于理解和设计复杂的逻辑电路。此外，逻辑函数的最简表达式和反演定理（摩根定理）对于简化逻辑表达式也非常重要。

第三章 门电路

门电路是构成数字逻辑电路的基本单元，常见的有正负逻辑的概念、OC门、OD门、TG门以及三态门的特点等。这些门电路在数字信号的处理和传输中起到关键作用。例如，OC门可以实现“线与”逻辑，而三态门则允许高阻状态，用于避免输出端的不确定状态。

第四章 组合逻辑电路

组合逻辑电路的分析主要通过真值表、卡诺图、逻辑表达式和时间图来进行。掌握这些分析方法有助于从逻辑电路图推导出其功能表达，进而验证电路设计的正确性。组合逻辑电路通常不包含存储元件，其输出仅取决于当前的输入。

第五章 半导体存储电路

半导体存储电路包括触发器和寄存器等组件，它们用于存储数据并在需要时提供。触发器是构建时序逻辑电路的基本单元，具有两个稳定的状态：“0”状态和“1”状态，并且可以根据不同的输入信号被置成1或0状态。寄存器则用于存储操作数和临时数据，分为通用寄存器和特殊用途寄存器等。

第六章 时序逻辑电路

时序逻辑电路的特点是能够记忆之前的状态并在适当的时刻更新其输出。这类电路通常由触发器组成，并受控于一个时钟信号。设计和分析时序逻辑电路通常涉及写出各触发器的控制函数、结合特性方程和控制函数写出状态方程以及画出输出函数等步骤。

第七章 脉冲波形的产生与整形

脉冲波形的产生与整形是数字电子技术中的重要环节，施密特触发器、单稳态触发器以及多谐振荡器是常用的脉冲整形器件。施密特触发器可以将缓慢变化的输入信号转换为清晰、锐利的输出脉冲；单稳态触发器可以产生一定时间的固定脉宽输出；多谐振荡器则能自动产生矩形波信号。

第八章 A/D和D/A转换器

模数转换器（A/D转换器）和数模转换器（D/A转换器）是实现模拟信号与数字信号相互转换的关键器件。A/D转换包括采样、量化和编码三个步骤；而D/A转换则是将数字信号转换为对应的模拟电平。这两种转换在现代电子设备中广泛应用，如音频处理、数据采集系统等。

版权声明：

本作品由lanwusb创作，本站仅为展示平台，未经作者授权，禁止任何形式的复制、转载或用于商业用途。任何未经允许的使用行为，我们将保留追究法律责任的权利。

感谢您的理解与支持！
