

数字电子电路学习教程

-----单片机自学网版权所有-----

编制：单片机自学网

网址 <http://www.c51.cn>

E-mail:c51cn@163.com

备用邮箱：h1-ddw_218@163.com

目 录

第一章：数字与编码

第一节：进位计数制

第二节：数值转换

第三节：二进制数的算术运算

第四节：数的原码、反码及补码

第五节：编码

第二章：基本逻辑运算及集成逻辑门

第一节：基本概念

第二节：三种基本的逻辑运算

第三节：常用的复合逻辑

第四节：集成逻辑门

第三章：布尔代数与逻辑函数化简

第一节：布尔代数的基本公式和规则

第二节：逻辑函数的代数法化简

第三节：卡诺图化简

第四章：组合逻辑电路

第一节：组合逻辑电路的分析

第二节：组合逻辑电路的设计

第三节：常用的组合逻辑

第五章：触发器

第一节：时序电路的概述

第二节：触发器

第六章：时序电路的分析与设计

第一节：同步时序电路的分析方法

第二节：同步时序电路的设计

第七章：常用时序逻辑部件

第一节：计数器

第二节：寄存器与移位寄存器

第三节：序列信号发生器

第八章：脉冲产生电路和定时电路

第一节：555 定时电路

第二节：单稳态电路

第三节：多谐振荡器和施密特电路

第九章：数模、模数转换电路

第一节：DAC

第二节：ADC

第十章：大规模集成电路

第一节：ROM 及其应用

第二节：PLA 及其应用

跟我学数字电子技术

数字电子技术是当前发展最快的学科之一。数字电路的设计过程和方法也在不断的发展和完善。由于半导体技术的迅速发展，微型计算机的广泛应用，所以数字电子技术在现代科学技术领域中占很重要的地位，应用也更加广泛。

第一章：数制与编码

这一章主要讲述的内容是在数字设备中进行算术运算的基本知识—数制和一些常用的编码。它是这门课程的基础。

我们在学习时把这一章的内容分为五节，它们分别是：

- 进位计数制
- 数值转换
- 二进制数的算术运算
- 数的原码、反码及补码
- 编码

第一节进位计数制

这一节我们来学习进位计数制的概念和一些常用的进位计数制。

一：进位计数制

它的概念描述为：把数划分为不同的位数，逐位累加，加到一定数量之后，再从零开始，同时向高位进位

进位计数制有三个要素：数符、进位规律和进位基数。

什么是进位基数呢？即计数制中每个数位所使用的数码符号的总数，它又被称为进位模数。我们经常把数用每位权值与该位的数码相乘展开。当某位的数码为“1”时所表征的数值即该位的权值。

例 1：我们把十六进制数 $N = (1FA3.B3)_{16}$ 按权展开式子为？

$$N = 1 \times 16^3 + 15 \times 16^2 + 10 \times 16^1 + 3 \times 16^0 + 11 \times 16^{-1} + 3 \times 16^{-2}$$

二：常用的进位计数制

我们用进位计数制的三要素来描述一下二进制、八进制、十进制和十六进制。如下表所示：

常用进制	英文表示符号	数码符号	进位规律	进位基数
二进制	B	0、1	逢二进一	2
八进制	O	0、1、2、3、4、5、 6、7	逢八进一	8
十进制	D	0、1、2、3、4、5、	逢十进一	10

		6、7、8、9		
十六进制	H	0、1、2、3、4、5、 6、7、8、9、A、B、 C、D、E、F	逢十六进一	16

第二节：数制转换

在数字设备中计数用的是二进制，但我们计数一般采用十进制，那它们之间是怎样转换的呢？

一：其它进制转换为十进制

方法是：将其它进制按权位展开，然后各项相加，就得到相应的十进制数。

例 1： $N = (10110.101)_B = (?)_D$

$$\begin{aligned} \text{按权展开 } N &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 16 + 4 + 2 + 0.5 + 0.125 = (22.625)_D \end{aligned}$$

二：将十进制转换成其它进制

方法是：它是分两部分进行的即整数部分和小数部分。

整数部分：（基数除法）

把我们要转换的数除以新的进制的基数，把余数作为新进制的最低位；

把上一次得的商在除以新的进制基数，把余数作为新进制的次低位；

继续上一步，直到最后的商为零，这时的余数就是新进制的最高位；

小数部分：（基数乘法）

把要转换数的小数部分乘以新进制的基数，把得到的整数部分作为新进制小数部分的最高位；

把上一步得的小数部分再乘以新进制的基数，把整数部分作为新进制小数部分的次高位；

继续上一步，直到小数部分变成零为止。或者达到预定的要求也可以。

例 2： $N = (68.125)_D = (?)_O$

整数部分

小数部分

$$\begin{array}{r} 8 \overline{) 68} \text{---} 4 \\ 8 \overline{) 8} \text{---} 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0.125 \\ * \quad 8 \\ \hline 1.0 \text{---} 1 \end{array}$$

$$(68.125)_D = (104.1)_O$$

三：二进制与八进制、十六进制的相互转换

二进制转换为八进制、十六进制：它们之间满足 2^3 和 2^4 的关系，因此把要转换的二进制从低位到高位每 3 位或 4 位一组，高位不足时在有效位前面添“0”，然后把每组二进制数转换成八进制或十六进制即可

八进制、十六进制转换为二进制：把上面的过程逆过来即可。

例 3： $N = (C1B)_H = (?)_B$

$$(C1B)_H = 1100/0001/1011 = (110000011011)_B$$

第三节：二进制数的算术运算

我们知道十进制可以进行四则运算,那么二进制能否进行四则运算? 答案是肯定的。

一：二进制的四则运算

二进制也可以进行四则运算，它的运算规则如下所示：

加运算	$0+0=0, 0+1=1, 1+0=1, 1+1=10$	逢 2 进 1
减运算	$1-1=0, 1-0=1, 0-0=1, 0-1=1$	(向高位借 1 当 2)
乘运算	$0*0=0, 0*1=0, 1*0=0, 1*1=1$	
除运算	二进制只有两个数 (0, 1), 因此它的商是 1 或 0.	

例 1: 求 $(1011101)_B$ 与 $(0010011)_B$ 之和

$$\begin{array}{r} 1011101 \\ + 0010011 \\ \hline 1110000 \end{array}$$

例 2: 求 $(1101)_B$ 与 $(0101)_B$ 的乘积

$$\begin{array}{r} 1101 \\ * 0101 \\ \hline 1101 \\ 0000 \\ 1101 \\ 0000 \\ \hline 100001 \end{array}$$

通过例(1)我们再来介绍两个概念：半加和全加。

半加：是最低位的加数和被加数相加时，不考虑低位向本位进位。

全加：是加数和被加数相加时，我们还要考虑低位向本位的进位。

第四节：数的原码、反码及补码

我们知道在生活中,数是有正负之分,在数字设备中是怎样表示数的正负符号呢?

一：数的表示形式

在生活中表示数的时候一般都是把正数前面加一个“+”，负数前面加一个“-”，但是在数字设备中，机器是不认识这些的，我们就把“+”用“0”表示，“-”用“1”表示。原码、反码和补码。这三种形式是怎样表示的呢？如下所示：

	真值	原码	反码	补码
正数	+X	0X	0X	0X

例 1: 求+12 和-12 八位原码、反码、补码形式

它们的原码分别为

$$[+12]=00001100 \quad [-12]=100011$$

它们的反码分别为 $[+12]^*=00001100$

$$[-12]^* = (2^8-1) + (-1100) = 11110011$$

它们的补码分别为 $[+12]**=00001100$

$$[-12]**=2^8 + (-1100) = 11110100$$

负数	-X	1X	$(2^n-1) + X$	$2^n + X$
----	----	----	---------------	-----------

二：原码、反码及补码的算术运算

因为这三种数码表示法的形成规则不同，所以算术运算方法也不相同。

原码：与我们的日常中算术运算相同。

反码：先转换为反码形式,再进行加减运算。它的减法可以按 $A_{反}+[-B]_{反}$ 的形式进行。

补码：先转换为补码形式，再进行加减运算,其减法可以按 $A_{补}+[-B]_{补}$ 进行。

三：溢出及补码运算中溢出的判断

溢出：可以描述为运算结果大于数字设备的表示范围。这种现象应当作故障处理。

判断溢出是根据最高位的进位来判断的。

第五节：编码

指定某一组二进制数去代表某一指定的信息，就称为编码。

一：二——十进制 (BCD) 码

用二进制码表示的十进制数，就称为 BCD 码。它具有二进制的形式，还具有十进制的特点它可作为人们与数字系统的联系的一种间表示。BCD 码分为有权和无权编码。

(1) **有权 BCD 码：**每一位十进制数符均用一组四位二进制码来表示，而且二进制码的每一位都有固定权值. 下面我们用表列出几种常见的编码：

十进制数	常见的编码	8421	5421	2421	631-1	余 3 码	7321
0		0000	0000	0000	0000	0011	0000
1		0001	0001	0001	0010	0100	0001
2		0010	0010	1000	0101	0101	0010
3		0011	0011	1001	0100	0110	0011
6		0110	1001	1100	1000	1001	0111
8		1000	1011	1110	1101	1011	1001
9		1001	1100	1111	1100	1100	1010

(2) **无权 BCD 码：**二进制码中每一位都没有固定的权值。

二：奇偶校验码

在数据的存取、运算和传送过程中，难免会发生错误，把“1”错成“0”或把“0”错成“1”。奇偶校验码是一种能检验这种错误的代码。它分为两部分：信息位和奇偶校验位。

有奇数个“1”称为奇校验，有偶数个“1”则称为偶校验。

第二章：基本逻辑运算及集成逻辑门

这一章我们学习的重点是数字设备进行逻辑运算的基本知识:基本逻辑运算和实现这些运算的门电路。它是本课程的基础，我们要掌握好！

在学习时，我们把它的内容分为：

- 基本概念
- 三种基本逻辑运算
- 常用的复合逻辑
- 集成逻辑门

第一节：基本概念

这一节来了解一下逻辑函数、逻辑变量和真值表的概念。

一：逻辑变量与逻辑函数

我们作某些事情，总是先对事情判断一下，然后再根据判断的结论去做。

例如我们吃饭，总是先判断：‘饭做好了么？’：‘人到齐了么？’：‘餐桌准备好了吗？’，只有上面的条件都满足了，我们才可以吃饭，否则就不能。

我们把用逻辑语言描述的条件称为**逻辑命题**，其中的每个逻辑条件我们都称为**逻辑变量**，我们一般用字母 A、B、C、D、、、、等表示。把逻辑变量写成函数的形式就称为**逻辑函数**。

例如：我们把上面我们提到的问题的条件分别用 A、B、C 表示，那么它的逻辑函数可表示为：

$$F=f(A, B, C)$$

二：真值表

因为逻辑变量只有两种取值 0 或 1，所以我们可以用一种表格来描述逻辑函数的真假关系，我们就称这种表格为**真值表**。

例如：列出“能吃饭么？”的真值表。设条件满足为 1，不满足为 0，我们知一个逻辑变量，有两种组合，三个逻辑变量就有八种组合。所以其真值表为：

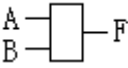
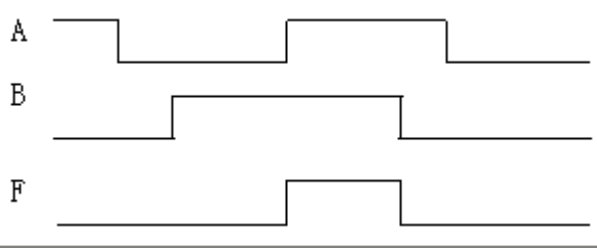
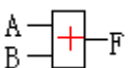
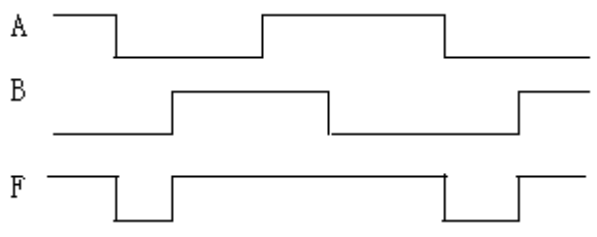
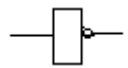
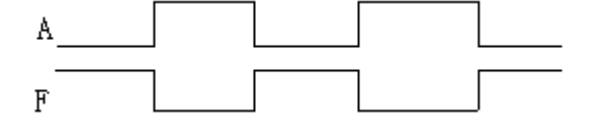
A	B	C	F
0	0	0	0
0	0	1	0

0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

第二节：三种基本的逻辑运算

在实际中我们遇到的逻辑问题是多种多样的，其实它们可以用三种基本的逻辑运算把它们概括出来。它们就是‘与’‘或’‘非’逻辑运算。

下面我们用表格来描述一下它们：

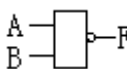
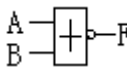
逻辑运算	逻辑表达式	逻辑符号	二变量运算结果	二变量输出波形
与运算	$F=AB$		$0*0=0; 0*1=0$ $1*0=0; 1*1=1$	
或运算	$F=A+B$		$0+0=0; 0+1=1$ $1+0=1; 1+1=1$	
非运算	$F=A$		$A=0; F=1$ $A=1; F=0$	

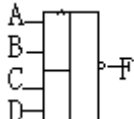
第三节：常用的复合逻辑

通过上一节的学习我们已经知道逻辑代数中有三种基本的逻辑运算，事实上我们总是希望用较少的器件来实现较多的功能，所以我们就用到复合逻辑。

一：常用的复合逻辑

经常用到的复合逻辑有三种：它们是“与非”、“或非”、“与或非”。


逻辑名称	逻辑表达式	逻辑符号	逻辑门特性
“与非”逻辑	$F=AB$		输入只要有“0”，输出为“1”，输入全部为“1”输出为“0”。
“或非”逻辑	$F=A+B$		输入只要有“1”，输出为“0”，输入全部为“0”输出为“1”

“与或非”逻辑	$F=AB+CD$		我们根据具体情况，来作决定。
---------	-----------	---	----------------

二：异或”逻辑和“同或”逻辑

有时我们还会用到“异或”逻辑和“同或”逻辑，它们都是两变量的逻辑函数。

“异或”逻辑指输入二变量相异时输出为“1”，相同时输出为“0”。

它的逻辑表达式为： $F=A\oplus B$ ，逻辑符号为：。

“同或”逻辑指输入二变量相同时输出位“1”，相异时输出位“0”。

它的逻辑表达式为： $F=A\odot B$ ，逻辑符号为：

三：正负逻辑

由于我们的规定不同，逻辑的输入端取值也不相同。我们把输入为正称为正逻辑，输入为负的称为负逻辑。因为我们在逻辑电路中，大多采用硅管，用的是正电源，所以我们一般采用正逻辑。

第四节：集成逻辑门

这一节我们来学习由晶体管组成的集成逻辑门的一些基本知识。

集成逻辑门分为两种即双极型集成电路和单极型集成电路。双极型集成电路分为：DTL 集成逻辑和 TTL 集成逻辑；单极型集成电路分为一般 MOS 逻辑和互补 MOS 逻辑（CMOS）。

一：双极型集成电路

它的特点是：工作速度快，易于做成大规模集成电路，功耗低等。我们来简单介绍一下双极型集成电路的两种形式（1）TTL 集电极开路门（OC 门）（2）三态门。

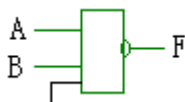
（1）TTL 集电极开路门（OC 门），它的特点是能实现“线与”功能，可以节省门数，减少输出门的级数。它可应用在数据总线上。当每个 OC 门只要有一个输入端为低电平时，OC 门的输出均为高电平。

（2）三态门；它的特点是输出端除了高电平、低电平两种状态外还有第三种状态：高阻状态或禁止状态。

例 1：如右图所示的三态门，试分析三态门各种输出情况。

当 E 为高电平时 输出端 F 为高阻状态

当 E 为低电平时 输出端 $F=AB$



由此我们可以看出三态门的输出端的情况与控制端有关，只有控制端为导通时输入端才有效。

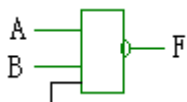
二：单极型集成电路

它的特点是：高、低电平都很理想；功耗很低，近似为“0”，任意时刻都有一个关闭；抗干扰能力强；兼容性强。

例 2：如右图试分析输入控制端的情况。

- 通过电阻接地时：电阻小于等于 700 欧姆时相当于输入为：“0”；当电阻大于等于 2000 欧姆时相当于输入为：“1”
- 当输入控制端悬空时相当于“1”

- 接高电平 U 时相当于“1”
- 接地时相当于“0”



第三章：布尔代数与逻辑函数化简

这一章主要是讲布尔代数和逻辑函数化简。在布尔代数中是把逻辑矛盾的一方假定为“0”，另一方假定为“1”这样就把逻辑问题数字化了。逻辑函数的化简也就是运用布尔代数的性质来进行化简。这一章是这门课程的重点，我们一点要掌握好！

我们在学习时把这一章的内容分为：

- 基本公式和规则
- 逻辑函数的代数法化简
- 卡诺图化简

第一节：布尔代数的基本公式和规则

一：布尔代数的基本公式

下面我们用表格来列出它的基本公式：

公式名称	公式	
1、0-1 律	$A*0=0$	$A+1=1$
2、自等律	$A*1=A$	$A+0=A$
3、等幂律	$A*A=A$	$A+A=A$
4、互补律	$A*A=0$	$A+A=1$
5、交换律	$A*B=B*A$	$A+B=B+A$
6、结合律	$A*(B*C) = (A*B)*C$	$A+(B+C) = (A+B)+C$
7、分配律	$A(B+C) = AB+AC$	$A+BC = (A+B)(A+C)$
8、吸收律 1	$(A+B)(A+B) = A$	$AB+AB=A$
9、吸收律 2	$A(A+B) = A$	$A+AB=A$
10、吸收律 3	$A(A+B) = AB$	$A+AB=A+B$
11、多余项定律	$(A+B)(A+C)(B+C)$ $= (A+B)(A+C)$	$AB+AC+BC=AB+AC$
12、否否律	$(\quad) = A$	$\underline{\quad}$
13、求反律	$AB=A+B$	$A+B=A*B$

下面我们来证明其中的两条定律:

(1) 证明: 吸收律 1 第二式 $AB+AB=A$

左式= $AB+AB=A(B+B)=A$ =右式 (因为 $B+B=1$)

(2) 证明: 多余项定律 $AB+AC+BC=AB+AC$

左式= $AB+AC+BC=AB+AC+BC(A+A)$

= $AB+AC+ABC+ABC$

= $AB(1+C)+AC(1+B)$

= $AB+AC$ =右式 证毕

注意: 求反律又称为摩根定律, 它在逻辑代数中十分重要的。

二: 布尔代数的基本规则

代入法则 它可描述为逻辑代数式中的任何变量 A, 都可用另一个函数 Z 代替, 等式仍然成立。

对偶法则 它可描述为对任何一个逻辑表达式 F, 如果将其中的“+”换成“*”, “*”换成“+”, “1”换成“0”, “0”换成“1”, 仍保持原来的逻辑优先级, 则可得到原函数 F 的对偶式 G, 而且 F 与 G 互为对偶式。我们可以看出基本公式是成对出现的, 二都互为对偶式。

反演法则 有原函数求反函数就称为反演 (利用摩根定律),

我们可以把反演法则这样描述: 将原函数 F 中的“*”换成“+”, “+”换成“*”, “0”换成“1”, “1”换成“0”; 原变量换成反变量, 反变量换成原变量, 长非号即两个或两个以上变量的非号不变, 就得到原函数的反函数。

第二节: 逻辑函数的代数法化简

逻辑函数化简的方法有两种, 分别是代数法和卡诺图法。这一节我们来学习: 代数法化简。

我们先来了解一个概念, 什么是逻辑电路图? 逻辑电路图就是用逻辑门组成的电路图。

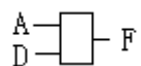
一: 逻辑函数化简的基本原则

逻辑函数化简, 没有严格的原则, 它一般是依以下几个方面进行:

- 逻辑电路所用的门最少;
- 各个门的输入端要少;
- 逻辑电路所用的级数要少;
- 逻辑电路要能可靠的工作。

这几条常常是互相矛盾的, 化简要根据实际情况来进行。下面我们来用例题说明一下:

例 1: 化简函数 $F=AB+CD+AB+CD$, 并用基本逻辑门实现。



(1) 先化简逻辑函数 $F=AB+CD+AB+CD=A(B+B)+D(C+C)=A+D$

(2) 用逻辑门实现：(由化简来看只需一个与门)

二：逻辑函数的形式和逻辑变换

逻辑函数的形式很多，一个逻辑问题可以用多种形式的逻辑函数来描述。

逻辑函数的表达式可分为五种：

1. “与或”表达式
2. “或与”表达式
3. “与非”表达式
4. “或非”表达式
5. “与或非”表达式。

这几种表达式之间可以互相转换，应根据要求把逻辑函数化简成我们所需要的形式。

第三节：卡诺图化简

上一节我们已经学习了代数法化简逻辑函数，这一节我们来学习另一种化简方法：图形法

一：在学习之前我们先来了解几个概念

- (1) 逻辑相邻项：它可描述为在两个与或逻辑中，除某个因子互为非外，其余的因子都相同。
- (2) 逻辑最小项：它可描述为在给定变量数目的逻辑函数中，所有变量参加相与的项。在某一个最小项中每个变量只能以原变量或反变量的形式出现一次。

逻辑最小项的性质是：

全部最小项之和为“1”；

两个不同的最小项之积为“0”；

n 变量有 2^n 项最小项。

- (3) 最小项标准式：全是最小项组成的“与或”式。

二：卡诺图化简的基本原理

凡两个逻辑相邻项，可合并为一项，其合并的逻辑函数是保留相同的，消去相异的变量。

三：卡诺图的结构

每一个最小项用一个方格表示，逻辑相邻的项几何位置上也相邻，卡诺图每方格取值按循环码排列

四：卡诺图的表示法

先将逻辑函数式化为最小项表达式，再填写卡诺图。

用真值表填写对应的卡诺图方格。

直接填写（横纵保留相同的因子）

五：卡诺图中的最小项的合并规律

合并规律:

2^1 个相邻项合并时消去一个相同的变量, 2^2 个相邻的项合并时消去两个相同的变量, 以此类推, 2^n 个相邻的项合并时消去 n 个相同的变量。

相邻项的性质是 (1) 具有公共边 (2) 对折重合 (3) 循环相邻

六:"与或"逻辑化简

例: 化简 $F=BCD+BC+ADC+ABC+ABC$ (用图形法)

(1) 用卡诺图表示逻辑函数: (如下图)

(2) 画卡诺圈圈住全部“1”的方格 (规则是: 圈尽可能大; 允许重复, 但要新; 孤立的“1”独圈。)

(3) 组成新函数是 $F=BC+AC+ADB$

(4) 画出逻辑电路: (如右下图所示)

AP	00	01	11	10
CD	00	1	1	0
	01	1	1	0
	11	1	0	1
	10	0	0	1

七:其它逻辑形式的化简

(1) "与非"逻辑形式

方法是:把逻辑函数用卡诺图化简得"与或"式, 然后"与或"式两次求反即得"与非"式。

(2) "或与"逻辑形式

方法是:

从卡诺图上求其反函数(圈"0"方格)

由反函数求得原函数, 再利用摩根定律即得"或与"式。

也可直接从卡诺图中求得"或与"式: 把图中的"0"作为原变量, 把原变量相"或"起来, 就得每一"或"项, 把每一项再"与"起来就是我们所求的结果。

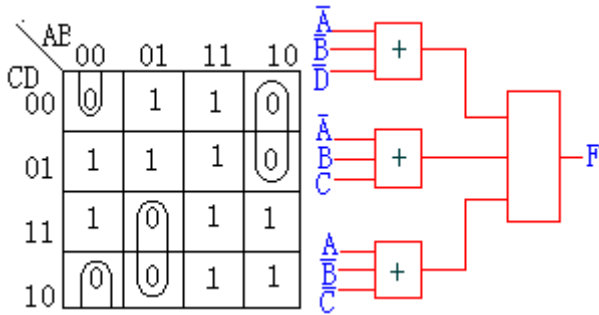
我们用例题来说明一下:

例 2: 求例题 1 得"或与"式。

1. 我们先用卡诺图表示函数式(如下左图)

2. 然后圈图中的"0"方格, 用"或与"式把函数的化简结果表示出来 $F=(A+B+D)(A+B+C)(A+B+C)$

3. 再用逻辑门电路来实现逻辑函数的化简结果。(如下右图)



(3) “或非”逻辑形式

方法是：先求得“或与”式，然后两次求反即得“或非”式。

(4) “与或非”逻辑形式

方法是(有两种)

得“与或”式后，两次求反不用摩根定律处理即得。

求得反函数(反函数的求法是：在卡诺图中圈“0”方格，然后用与或式把“0”方格实现出来既是反函数)后，再求一次反不用摩根定律处理即得。

八：无关项及无关项的应用

逻辑问题分完全描述和非完全描述两种。

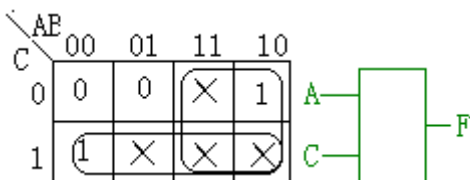
完全描述就是函数得每组变量不管取什麼值，逻辑函数都有意义，逻辑函数与每个最小项都有关。

非完备描述就是在实际中变量的某些取值式函数没有意义或变量之间有一定的制约关系。

我们把与函数无关的最小项称为无关项，它有时也称为禁止项，约束项，任意项。它的输出是任意的。化简有无关项的逻辑函数时，若无关项对化简有帮助则认为是“1” 否则为“0”。

例 3. 化简 $F=ACB+BAC$ 约束项条件为 $AB+AC+BC=0$

1. 先用卡诺图把函数表示出来, 约束项就是 AB、AC、BC 不能同时为“0”(如下左图)
2. (我们从图中可以看到, 若不考虑无关项的话, 函数时不能化简得) 考虑无关项的化简结果为 $F=A+C$.
3. 用门电路来实现逻辑函数. (如下右图)



九：输入只有原变量的函数化简

在实际中有时会遇到只有原变量的函数, 那怎样化简它呢?

用“非”门求得反变量来解决这种问题是很不经济。可以用三级电路设计法(阻塞法)来解决这样的问题。

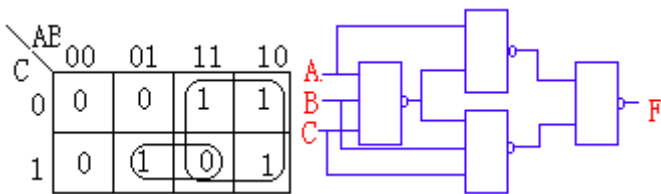
在卡诺图中人们可以发现一种特殊现象. 当卡诺圈中含有全“1”方格(二变量的“11”即 AB; 三变量的“111”即 ABC; 等)时, 其化简结果均为原变量。

在化简这类问题时就可以利用这个性质, 若没有给全“1”的逻辑项, 可以先把它在卡诺图中圈出来, 然后再阻塞掉即可。

例 4: 输入只有原变量, 用与非门实现 $F = \Sigma(3, 4, 5, 6)$

1. 现在用卡诺图化简函数(如下左图), 并阻塞掉全“1”方格. $F = AABC + BCABC = \overline{A}A\overline{B}C + \overline{B}CABC$

2. 用逻辑门电路实现逻辑函数如下右图所示(它为三级电路)



十: 多输出函数的化简

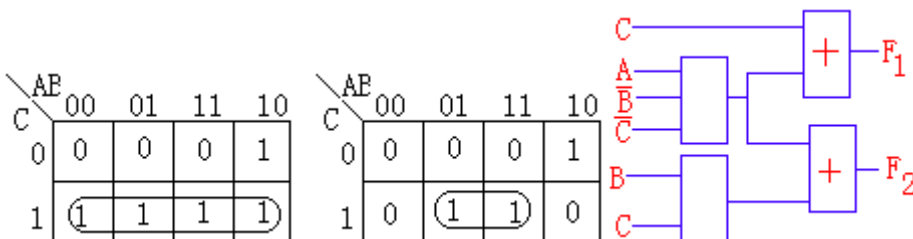
实际中电路常常有两个或两个以上的输出端, 在化简这类问题是不能单纯地去追求各个函数最简, 我们应统一考虑, 充分利用公共项。

例 5: 化简 $F_1 = \Sigma(1, 3, 4, 5, 7)$ $F_2 = \Sigma(3, 4, 7)$ 并用门电路实现。

1. 用卡诺图分别化简函数, 由于卡诺图中都含有 ABC 这一项, 所以把它作为公共向来考虑。(如下左图)

化简结果为: $F_1 = C + ABC$, $F_2 = BC + ABC$

2. 根据化简结果来用门电路来实现。(如下右图)



第四章：组合逻辑电路

数字电路分为组合逻辑电路和时序逻辑电路两类，组合逻辑电路的特点是输出信号只是该时的输入信号的函数，与别时刻的输入状态无关，它是无记忆功能的。这一章我们来学习组合逻辑电路。这一章是本课程的重点内容之一

我们在学习时把这一章的内容分为：

- 逻辑电路的分析
- 逻辑电路的设计
- 常用的组合逻辑

第一节：组合逻辑电路的分析

一：组合逻辑电路的分析

我们对组合逻辑电路的分析分以下几个步骤：

- (1)：有给定的逻辑电路图，写出输出端的逻辑表达式；
- (2)：列出真值表；
- (3)：通过真值表概括出逻辑功能，看原电路是不是最理想，若不是，则对其进行改进；

例 1：已知右面的逻辑电路图，试分析其功能。

第一步：写逻辑表达式。我们由前级到后级写

出各门逻辑表达式。

$$P=A+B \quad S=A+P=AB \quad W=B+P=AB$$

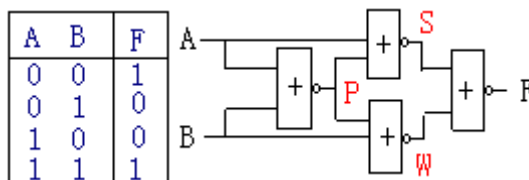
$$F=S+W=AB+A \cdot B$$

第二步：列真值表（如右图所示）。

第三步：逻辑功能描述并改进设计。

从真值表中可以看出这是一个二变量“同或”电路。原电路设计不合理，它只需一个“同或”门即可。

第二节：组合逻辑电路的设计



一：组合电路逻辑电路的设计

电路设计的任务就是根据功能设计电路，一般按如下的步骤进行：

- 1) 逻辑命题换为真值表；这一步我们要从以下几个反面考虑
用英文字母代表输入或输出；
分清几个输入、输出；
分清输入和输出之间的对应关系。
- 2) 逻辑函数进行化简，化简的形式则是根据所选用的逻辑门来决定；
- 3) 根据化简结果和所选定的门电路，画出逻辑电路图。

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

例：设计三变量表决器，其中 X 具有否决权。

第一步：列出真值表。（如右上图）

设 X、Y、Z 分别代表参加表决的变量；F 为表决结果，

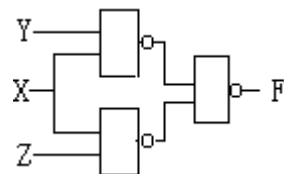
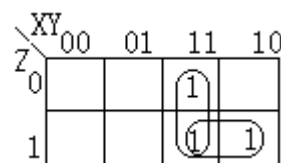
我们把变量规定为：X、Y、Z 为 1 表示赞成；为 0 表示反对。

F 为 1 表示通过；为 0 表示被否决。

第二步：化简逻辑函数。

我们选用与非逻辑来实现。用卡诺图来化简（如右中图） $F = \overline{XY} \cdot \overline{XZ}$

第三步：画逻辑电路。（如右下图）



第三节：常用的组合逻辑

常用组合逻辑的种类很多，主要有全加器、译码器、编码器、多路选择器等，下面我们分别把它们介绍一下。

一：半加器和全加器

在数字系统中算术运算都是利用加法进行的，因此加法器是数字系统中最基本的运算单元。由于二进制运算可以用逻辑运算来表示，因此我们可以用逻辑设计的方法来设计运算电路。加法在数字系统中分为全加和半加（第一章我们已经介绍了）所以加法器也分为全加器和半加器。

(1) 半加器设计

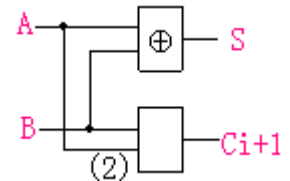
A	B	S	C _{i-1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

半加器不考虑低位向本位的进位，因此它有两个输入端和两个输出端。设加数（输入端）为 A、B；和为 S；向高位的进位为 C_{i+1}。

它的真值表为：如右图所示

函数的逻辑表达式为： $S=A\oplus B$ ； $C_{i+1}=AB$

逻辑电路图（用异或门和与门构成）为：如右图(2)所示

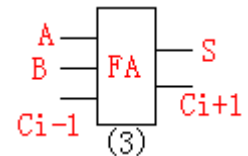


(2) 全加器的设计（它的逻辑符号为图(3)所示）

由于全加器考虑低位向高位的进位，所以它有三个输入端和两个输出端。

设输入变量为（加数）A、B、C_{i-1}，输出变量为 S、C_{i+1}

它的真值表为：如图(4)所示



函数的逻辑表达式为： $S=ABC_{i-1}+\overline{A}BC_{i-1}+A\overline{B}C_{i-1}+\overline{A}\overline{B}C_{i-1}+A\oplus B\oplus C_{i-1}$

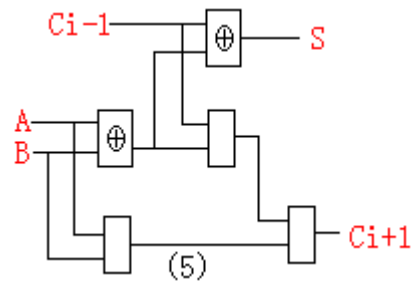
$$C_{i+1}=ABC_{i-1}+\overline{A}BC_{i-1}+A\overline{B}C_{i-1}+\overline{A}\overline{B}C_{i-1} = (A\oplus B)C_{i-1} + AB$$

C_{i-1}+AB

逻辑电路图(用异或和与门构成)为：如图(5)所示

A	B	C _{i-1}	S	C _{i+1}
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

(4)



(3) 全加器的应用

因为加法器是数字系统中最基本的逻辑器件，所以它的应用很广。它可用于二进制的减法运

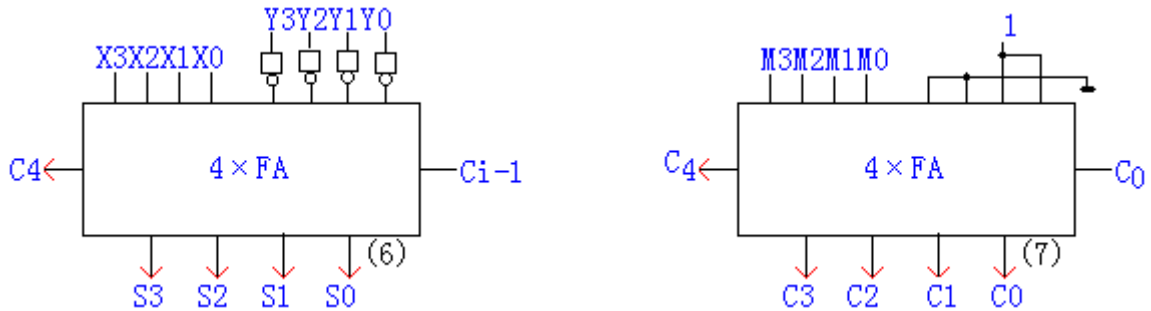
算、乘法运算，BCD 码的加、减法，码组变换，数码比较等。

例 1：用全加器构成二进制减法器。

以四位二进制为例。（减法可转换为加补运算）

设两组四位二进制分别为 $X_3X_2X_1X_0$ 和 $Y_3Y_2Y_1Y_0$ ，把 $Y_3Y_2Y_1Y_0$ 先进行求补然后再进行加法运算。

因为求补是逐位求反后再加“1”所以它的逻辑电路图为如图(6)所示：



例 2：采用四位全加器完成 8421BCD 码转换为余 3 代码。

由于 8421BCD 码加 0011 即为余 3 代码，因此转换电路就是加法电路。

设 8421BCD 码四位又高位到低位为 M_3, M_2, M_1, M_0 ，余 3 代码的四位由高到低为 C_3, C_2, C_1, C_0 。

它的逻辑电路图为如图(7)所示：

二：编码器和译码器

指定二进制代码代表特定的信号的过程就叫**编码**。把某一组二进制代码的特定含义译出的过程叫**译码**。

(1) **编码器** 因为 n 位二进制数码有 2^n 种状态，所以它可代表 2^n 组信息。我们在编码过程中一般是采用编码矩阵和编码表，编码矩阵就是在卡诺图上指定每一方格代表某一自然数，把这些自然数填入相应的方格。

例 1：把 0、1、2、...、9 编为 5421BCD 码。

先来确定编码表如图(1)所示和编码矩阵如图(2)所示：

由编码表确定各输出端的逻辑表达式是：

$$A=5+6+7+8+9$$

$$B=4+9$$

$$C=2+3+7+8$$

AE	00	01	11	10
00	0	4	*	8
01	1	5	*	9
11	3	7	*	*
10	2	6	*	*

(2)

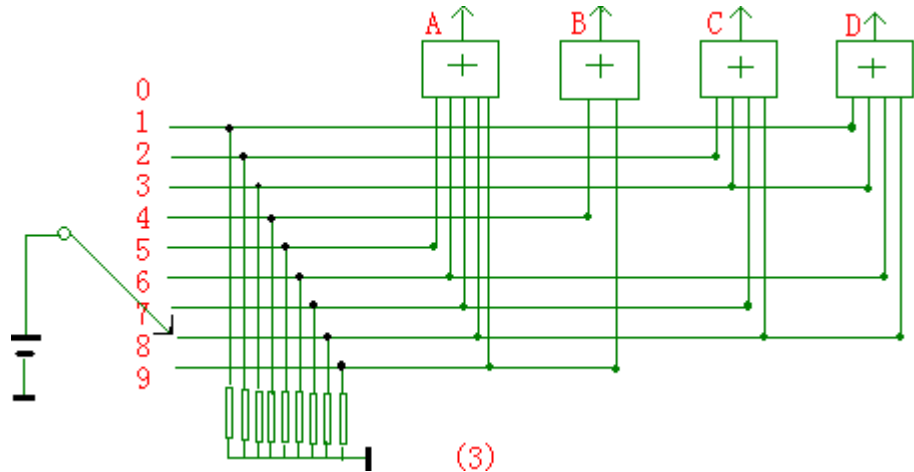
$$D=1+3+6+8$$

根据这些表达式可用或门组成

逻辑电路如图(3)所示:

N	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

(1)



(3)

(2) : **译码器** 编码的逆过程就是译码。 译码就是把代码译为一定的输出信号，以表示它的原意。实现译码的电路就是译码器。

译码器可分为二进制译码器、十进制译码器、集成译码器和数字显示译码驱动电路。

其中二进制译码器是一种最简单的变量译码器,它的输出端全是最小项。

例 2: 设计一译码电路把 8421BCD 码的 0、1、2、...、9 译出来。

四位二进制有十六种状态，而实际只需要十种，因此其余项作无关项考虑。

其编码矩阵为如图(4)所示。

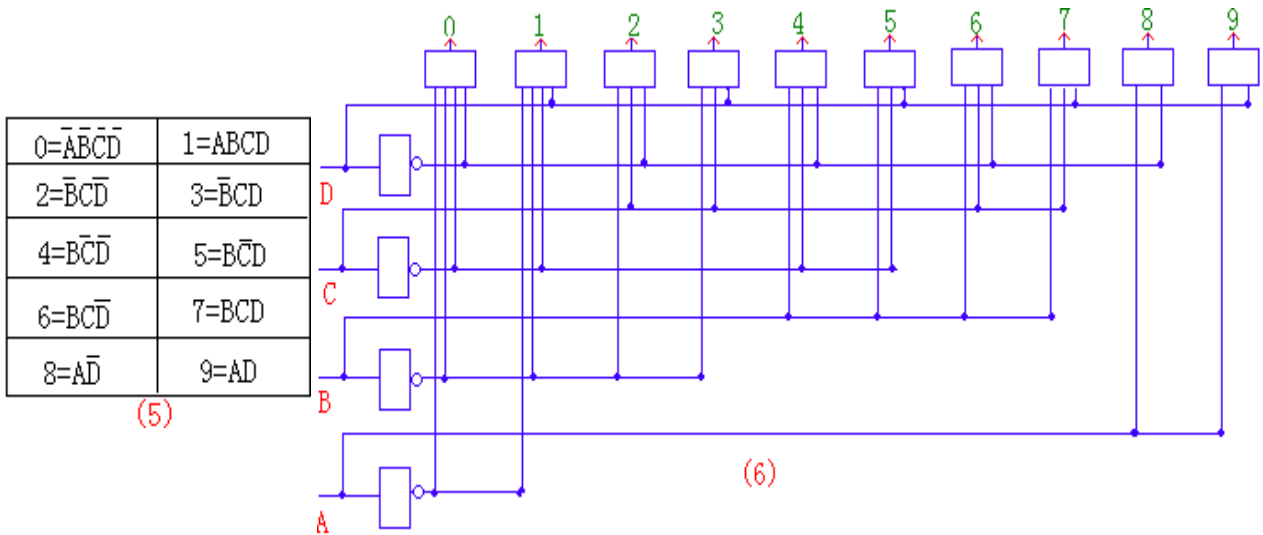
我们通过编码矩阵可得如下译码关系：如图(5)所示。

所以它的逻辑电路图为（用与门和与非门实现）

如图(6)所示：

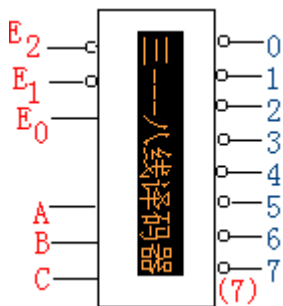
AB	00	01	11	10
00	0	4	*	8
01	1	5	*	9
11	3	7	*	*
10	2	6	*	*

(4)



集成译码器的工作原理与其它译码器一样，但它有它的特点。它的特点为：

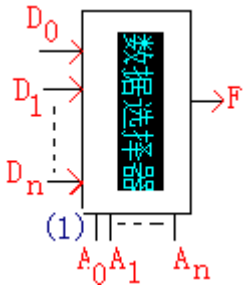
- 1) 输入采用缓冲级；(减轻信号负载)
 - 2) 输出为反码；低电平有效(减轻输出功率)
 - 3) 增加了使能端。(便于扩展功能)
- 目前常用的典型的集成译码器是三——八译码器。



它的逻辑符号为. 如图(7)所示：注：其中 $E_0E_1E_2$ 为使能端，只有当 $E_1、E_2$ 为 0 时 E_0 为 1 时此译码器才工作

三：数据选择器和多路分配器

(1) **数据选择器** 它就是从多个输入端中选择一路输出，它相当于一个多路开关它的逻辑符号如图(1)所示：其中 $D_0D_1、、D_n$ 是数据输入端； $A_0A_1、、A_n$ 为地址变量（有 n 个地址变量就有 2^n 个输入端）。常用的有二选一，四选一，八选一和十六选一，若需更多则由上述扩展。

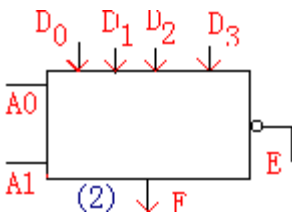


例 3: 如图(2)所示的四选一数据选择器, 试写出它的输出逻辑表达式和功能表

它的逻辑输出表达式为 $F = (A_0A_1D_0 + A_0A_1D_1 + A_0A_1D_2 + A_0A_1D_3) E$

它的功能表为: 如下表所示

从表上我们可以看出当使能端 E 为“1”时输出为“0”即禁止, 只有当使能端为“0”时选择器才有效。



地址		使能端	输入	输出
A0	A1	E	D	F
*	*	1	*	0
0	0	0	D ₀ ---D ₃	D ₀
0	1	0	D ₀ ---D ₃	D ₁
1	0	0	D ₀ ---D ₃	D ₂
1	1	0	D ₀ ---D ₃	D ₃

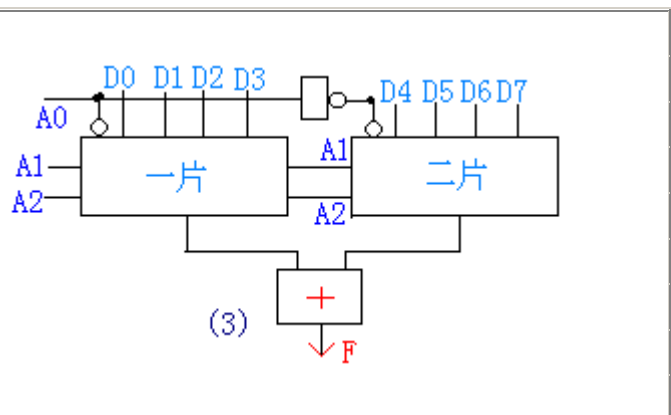
例 4: 把四选一扩展为八选一。

八选一要有八个输入变量, 因此需要三个地址变量(我们把其中一个 A0 作为使能段);

四选一只能有四个输入变量, 所以我们需要两个四选一和一个非门. 非门的作用是改变使能端的电平, 减少使能端. 先列出它的功能表如下表所示:

逻辑电路图如图(3)所示:

A0	A1	A2	D	F
0	0	0	D ₀ ---D ₇	D ₀
0	0	1	D ₀ ---D ₇	D ₁
0	1	0	D ₀ ---D ₇	D ₂
0	1	1	D ₀ ---D ₇	D ₃
1	0	0	D ₀ ---D ₇	D ₄
1	0	1	D ₀ ---D ₇	D ₅
1	1	0	D ₀ ---D ₇	D ₆
1	1	1	D ₀ ---D ₇	D ₇



(2) **多路分配器** 它的功能是把输入数据分配给不同的通道上,相当于一个单刀多掷开关。

第五章 触发器

这一章我们来学习时序逻辑电路。时序逻辑电路的特点是任何时刻产生的稳定输出信号不仅与该时刻输入信号有关而且与它过去的状态有关。因此它是具有记忆功能的电子器件。它分为同步时序电路和异步时序电路。

我们在学习时把这一章分为两节,它们分别是:

- 时序电路的概述
- 触发器

第一节: 时序电路的概述

这一节我们来学习一些关于时序电路的概念,在学习时要注意同步时序电路和异步时序电路的区别

一: 时序电路概述

同步时序电路的状态只在统一的信号脉冲控制下才同时变化一次,如果信号脉冲没有到来,即使输入信号发生变化,电路的状态仍不改变。

异步时序电路的状态变化不是同时发生的,它没有统一的信号脉冲(时钟脉冲用CP表示),输入信号的变化就能引起状态的变化。

二: 时序电路的表示形式

时序电路按输入变量的依从关系可分为米里型和莫尔型。米里型电路的输出是输入变量的现态函数;莫尔型电路的输出仅与电路的现态有关。

一般用 $Q^n(t)$ 表示现态函数,用 $Q^{n+1}(t)$ 表示次态函数。它们统称为状态函数,一个时序电路的主要特征是由状态函数给出的。

三: 时序电路的特征

时序电路中记忆功能是靠触发器来实现的,我们设计和分析时序电路的对象就是触发器。描述时序电路时通常使用状态表和状态图,我们分析时序电路的方法通常是比较相邻的两种状态(即现态和次态)。

例 1: 列出下表所示时序电路的逻辑表达式、状态表和状态图

逻辑表达式为: $Q^{n+1}=AQ^n+BQ^n$ $F=A \oplus B+AB$,

它的状态表为如下右图所示

Q^n	A	B	Q^{n+1}	F
0	0	0	0	1
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	0	1

	Q^{n+1}			
$Q^n \backslash AB$	00	01	11	10
0	0/1	0/0	1/0	1/1
1	1/1	0/0	1/0	0/1

第二节：触发器

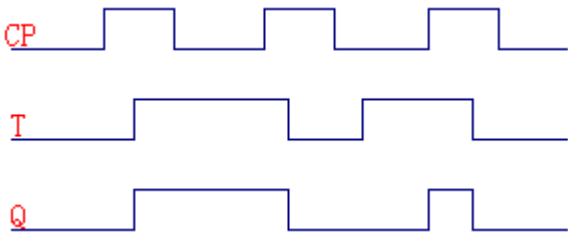
我们在学习触发器的时要注意以下几点：触发器的状态表、状态图、逻辑符号、特征方程以及各触发器的特点。常用的触发器有：R-S 触发器、D 触发器、T 触发器和 JK 触发器。

一：R-S 触发器和 D 触发器

	R-S 触发器	D 触发器																																
逻辑符号																																		
特征方程	$Q^{n+1} = S_d + R_d Q^n$	$Q_{n+1} = D$																																
状态表	<table border="1"> <tr> <td></td> <td colspan="4">Q^{n+1}</td> </tr> <tr> <td>$Q^n \backslash S_d R_d$</td> <td>00</td> <td>01</td> <td>11</td> <td>10</td> </tr> <tr> <td>0</td> <td>×</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>×</td> <td>1</td> <td>1</td> <td>0</td> </tr> </table>		Q^{n+1}				$Q^n \backslash S_d R_d$	00	01	11	10	0	×	1	0	0	1	×	1	1	0	<table border="1"> <tr> <td></td> <td colspan="2">Q^{n+1}</td> </tr> <tr> <td>$Q^n \backslash D$</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> </table>		Q^{n+1}		$Q^n \backslash D$	0	1	0	0	1	1	0	1
	Q^{n+1}																																	
$Q^n \backslash S_d R_d$	00	01	11	10																														
0	×	1	0	0																														
1	×	1	1	0																														
	Q^{n+1}																																	
$Q^n \backslash D$	0	1																																
0	0	1																																
1	0	1																																
状态图																																		
功能概述	<ul style="list-style-type: none"> ● $S_d=0, R_d=1$ 时，触发器处于置位状态，次态=1 ● $S_d=1, R_d=0$ 时，次态=0，处于复位状态。 ● $S_d=R_d=1$ 时，触发器状态不变，处于维持状态。次态=现态 ● $S_d=R_d=0$ 时，次态=现态=1，破坏了触 	<ul style="list-style-type: none"> ● 当 $CP=0$ 时，触发器不工作，处于维持状态。 ● 当 $CP=1$ 时，它的功能如下： ● 当 $D=0$ 时，次态=0， ● 当 $D=1$ 时，次态=1， 																																

	<p>发器的平衡, 触发器处于禁止状态。(工作是不允许出现这种情况)</p>	<p>由此可见, 当触发器工作时它的次态由输入控制函数 D 来确定。</p> <p>(CP 为时钟脉冲, 它使触发器有节奏的工作)</p>
--	--	---

例1、 已知 D 触发器的 CP 脉冲、D 输入端的输入波形, 画出次态的波形图。

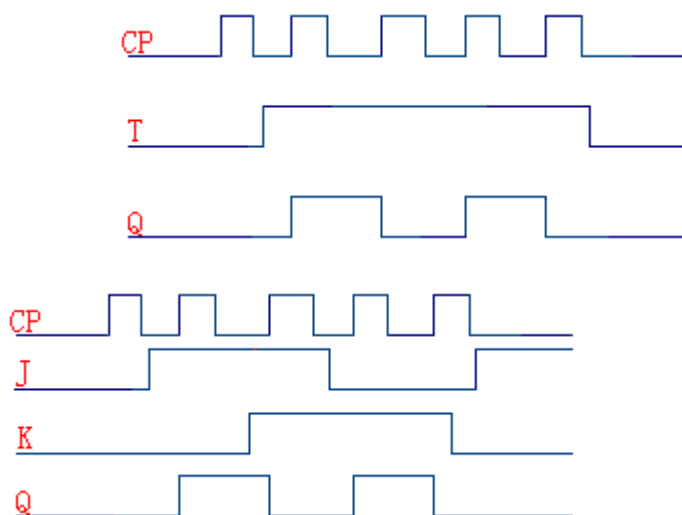


二: T 触发器和 JK 触发器

	T 触发器	JK 触发器																																					
逻辑符号																																							
特征方程	$Q^{n+1} = TQ^n + \bar{T}\bar{Q}^n$	$Q^{n+1} = JQ^n + \bar{K}\bar{Q}^n$																																					
状态表	<table border="1" style="text-align: center;"> <tr> <td></td> <td></td> <td colspan="2">Q^{n+1}</td> </tr> <tr> <td>$Q^n \setminus T$</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> </tr> </table>			Q^{n+1}		$Q^n \setminus T$	0	0	1	0	0	0	1	1	1	1	0	<table border="1" style="text-align: center;"> <tr> <td></td> <td></td> <td colspan="4">Q^{n+1}</td> </tr> <tr> <td>$Q^n \setminus K$</td> <td>00</td> <td>01</td> <td>11</td> <td>10</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> </table>			Q^{n+1}				$Q^n \setminus K$	00	01	11	10	0	0	0	1	1	1	1	0	0	1
		Q^{n+1}																																					
$Q^n \setminus T$	0	0	1																																				
0	0	0	1																																				
1	1	1	0																																				
		Q^{n+1}																																					
$Q^n \setminus K$	00	01	11	10																																			
0	0	0	1	1																																			
1	1	0	0	1																																			
状态图																																							
功能概述	<ul style="list-style-type: none"> ● CP=0 时, 触发器不工作, 处于维持状态 ● CP=1 时, 触发器的功能如下: 	<ul style="list-style-type: none"> ● 当 CP=0 时, 触发器不工作, 处于维持状态 ● 当 CP=1 时, 触发器的功能如下: 																																					

	<ul style="list-style-type: none"> ● T=0 时, 次态=现态; ● T=1 时, 次态与现态相反: 触发器翻转 	<ul style="list-style-type: none"> ● 当 JK 为 00, 01, 10 时实现 R-S 触发器的功能 ● 当 JK 为 11 时它实现 T 触发器的功能。
--	--	--

例 1. 已知 T 触发器的 CP 脉冲、T 的输入波形, 试画出输出波形。(如下左图)



例 2. 已知 JK 触发器的 CP 脉冲、JK 的输入波形, 画出输出波形。(如上右图)

三:基本触发器的空翻和振荡现象及解决

(1) 触发器在应用中, CP 脉冲期间控制端的输入信号发生变化或 CP 脉冲过宽, 有时会使触发器存在空翻和振荡现象, 它破坏了触发器的平衡。

(2) 为了解决这个问题, 必须改进电路设计, 实际中常用的结构有三种类型:

■ **主从触发器.** 它的类型有: 主从 R-S 触发器、主从 JK 触发器。

■ **维持阻塞触发器.** 维持就是在 CP 期间触发器完成其预定功能; 阻塞就是在 CP 期间阻止触发器产生不应有的操作。它的类型有: 维持阻塞 D 触发器。

■ **边沿触发器.** 它分为上升沿触发、下降沿触发、上升下降沿同时触发三种情况。边沿触发也就是在 CP 脉冲上升或下降的瞬间, 输出状态发生改变。

四:触发器的相互转换

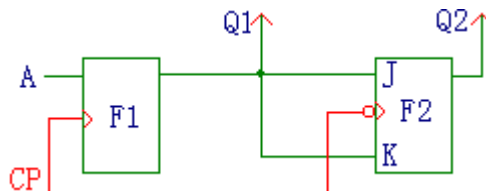
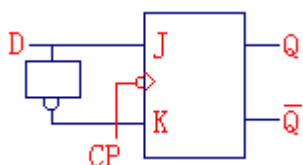
基本触发器之间是可以互相转换的, *JK* 触发器和 *D* 触发器是两种最常用的触发器, 别的触发器可以通过这两种触发器转化得来, 它们之间也可相互转化。

JK 触发器具有两个输入控制端, 它转化为别的触发器十分方便。

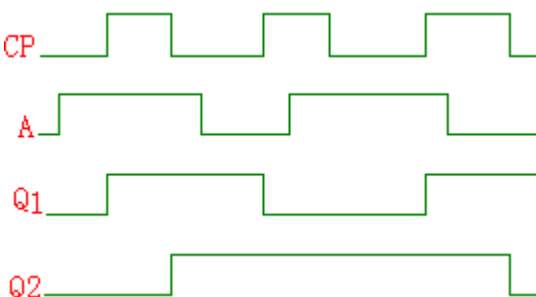
我们转化后怎样判断它们的正确性呢? 是根据各触发器的特征方程来验证。

例 1: 已知 *D* 触发器, 试把它转化为 *JK* 触发器。

D 触发器的特征方程为: $Q^{n+1}=D$ *JK* 触发器的特征方程为 $Q^{n+1}=JQ^n+KQ^n$, 由此可以看出转化电路如下左图



例 2: 在上右图中 F1 是 *D* 触发器, F2 是 *JK* 触发器, CP 和 A 的波形如右上图所示, 试画出 Q1, Q2 的波形。



从电路图中我们可以看到 F1 是在上升沿触发, F2 是在下降沿触发. 所以波形图如右图所示:

第六章 时序电路的分析与设计

逻辑电路分为组合逻辑电路和时序逻辑电路。第四章已经学习了组合逻辑电路的分析与设计的方法, 这一章我们来学习时序电路的分析与设计的方法。

在学习时序逻辑电路时应注意的重点是常用时序部件的分析与设计 这一章的内容共分为两节, 它们是:

§6、1: 同步时序电路的分析方法

§6、2: 同步时序电路的设计

第一节：同步时序电路的分析方法

时序电路分析的目的就是对已知的时序逻辑电路,要得到它的电路特性说明即该电路逻辑功能,若电路存在问题,并提出改进方法。

在分析同步时序电路时分为以下几个步骤:

分清时序电路的组成.

列出方程. 根据时序电路的组合部分,写出该时电路的输出函数表达式. 并确定触发器输入信号的逻辑表达式(激励函数),由此得到触发器的特征方程.

由上步得出的方程写出状态真值表,把触发器的现态和外界的输入信号作为时序电路的输入信号.

通过状态真值表得到该时序电路的状态图和状态表.

通过电路的状态表和状态图,对电路进行功能描述.

例 1:分析图(1)所示的电路,作出状态转换表及状态装换图,并作出输入信号为 0110111110 的输出波形.

1. 列方程.

激励方程为: $J_1=XQ_2, K_1=X$; $J_2=X$, $K_2=XQ_1$

特征方程为: $Q_1^{n+1}=XQ_2+XQ_1, Q_2^{n+1}=XQ_2+XQ_1Q_2$

输出方程为: $Z=XQ_1Q_2$

2. 列状态转换真值表. 如表(2)所示:

3. 画出状态迁移图. 如图(3)所示:

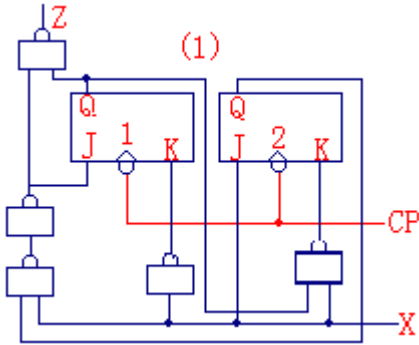
4. 功能描述.

由状态迁移表可看出,该电路是“1111”

序列检测电路, 当

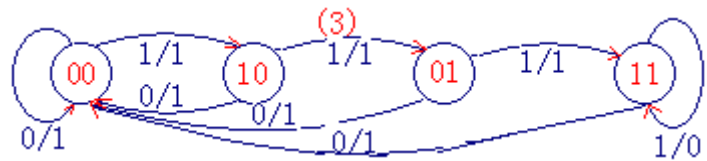
出现该序列时, 输出为“1”, 否则为“0”

5. 画时序图. 先列出时序表, 根据时序表(如图(4))作出时序图如图(5)所示:



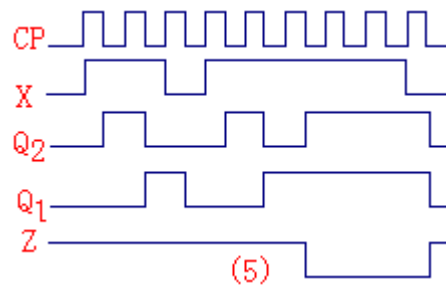
(2)

X	Q ₂	Q ₁	Q _{2ⁿ}	Q _{1ⁿ}	Z
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	1	1	1	0



(4)

X=	0	1	1	0	1	1	1	1	1	0
Q ₂	0	0	1	0	0	1	0	1	1	1
Q ₁	0	0	0	1	0	0	1	1	1	1
Q _{2ⁿ}	0	1	0	0	1	0	1	1	1	0
Q _{1ⁿ}	0	0	1	0	0	1	1	1	1	0
Z	1	1	1	1	1	1	1	0	0	1



第二节：同步时序电路的设计

时序电路是由组合电路和记忆电路两部分组成的.组合电路的设计在第四章已经学习过,因此时序电路的设计主要是记忆电路部分(触发器)的设计.

同步时序电路的设计分为以下几个步骤:

1. 建立原始状态图.

建立原始状态图的方法是:

确定输入、输出和系统的状态函数(用字母表示)。

根据设计要求, 确定每一状态在规定条件下的状态迁移方向, 得到原始状态图。

2. 化简原始状态.

在制作原始状态图时, 难免会出现多余状态(●触发器的个数增多●激励电路过于复杂等), 因此要进行状态化简, 化简时应根据具体情况来考虑。

3. 分配化简后的状态.

把化简后的状态用二进制代码来表示称为状态编码. 时序电路中, 电路的状态是由触发器的状态来描述的。

例 1. 设计一个“111...”序列检测器. 题意即检测连续三个“1”输出为“1”. 允许重合。

1. 画出状态迁移图. 如图(1)所示:

2. 列出状态表. 如表(2)所示(化简前); 如表(3)所示(化简后)

3. 化简状态. 通过状态表可以看出, 所列状态为最简状态。

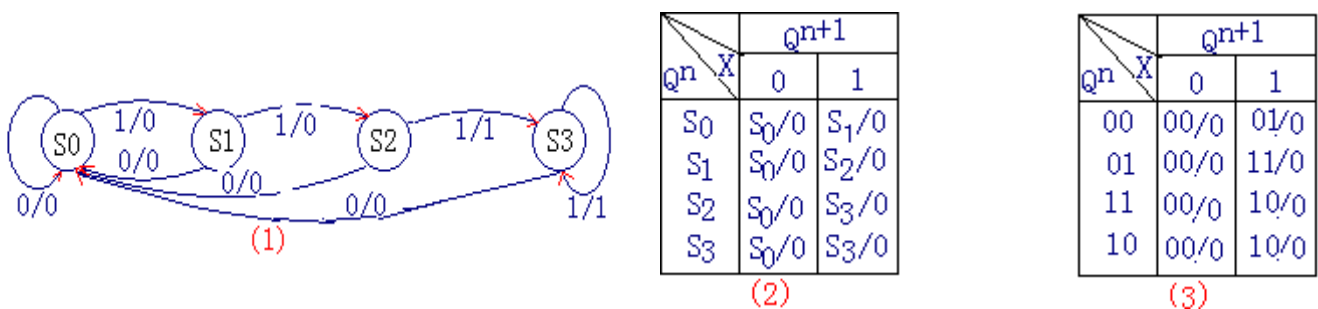
4. 状态分配. $S_0 \rightarrow Q_1Q_0=00; S_1 \rightarrow Q_1Q_0=01; S_2 \rightarrow Q_1Q_0=10; S_3 \rightarrow Q_1Q_0=11.$

5. 求激励方程. 如用 JK 触发器则激励方程为(由卡诺图(4、5)得):

$$Q_1^{n+1} = XQ_1^n Q_0^n + XQ_1^n \quad J_1 = XQ_0^n, \quad K_1 = X;$$

$$Q_0^{n+1} = XQ_1^n Q_0^n + XQ_1^n Q_0^n \quad J_0 = XQ_1^n \quad K_0 = XQ_1^n$$

6. 画出逻辑电路图. 如图(6)所示:

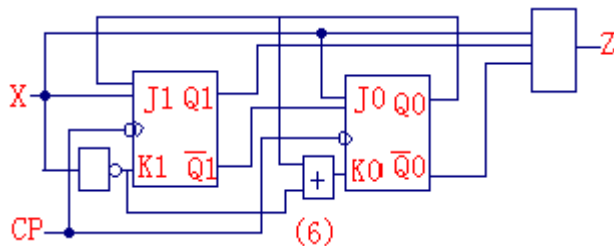


$Q_1 \backslash Q_0$	00	01	11	10
0	0	0	0	0
1	0	1	1	1

(4)

$Q_1 \backslash Q_0$	00	01	11	10
0	0	0	0	0
1	1	1	0	0

(5)



第七章 常用时序逻辑部件

这一章主要是介绍常用的时序逻辑功能部件。如计数器、移位寄存器的分析与设计方法以及集成计数器、集成移位寄存器的原理及应用。它是本课程的重点内容之一，我们一定要掌握好！

在学习时要注意同步、异步计数器和移位寄存器的工作原理及设计方法；同步式集成计数器 T214、异步式集成计数器 T210 以及集成移位寄存器 T454 的工作原理及应用。

在学习是我们把这一章的内容共分为三节，它们分别是：

- 计数器
- 寄存器与移位寄存器
- 序列信号发生器

第一节：计数器

累计输入脉冲的个数的逻辑电路称为计数器。它的作用有：累计输入脉冲的个数；对输入脉冲信号进行分频；构成其它时序电路。

计数器的分类：

- 按进位模数分为 **模 2 计数器** 和 **非模 2 计数器**。
进位模是计数器所经历的独立状态的总数，也就是进位制数。
模 2 计数器就是进位模为 2^n 的计数器。其中 n 为触发器的级数；非模 2 计数器就是进位模非 2^n 的计数器。
- 按计数脉冲的输入方式分为 **同步计数器** 和 **异步计数器**。
同步计数器是相应的触发器的计数脉冲也相同，使相应的触发器同时翻转。异步计数器是相应的触发器的计数脉冲不相同，并且不同时翻转。
- 按计数增减趋势分为 **递增计数器**、**递减计数器** 和 **双向计数器**。
- 递增计数器是每来一个时钟脉冲触发器的组成状态按二进制代码规律增加，递减计数器就是按二进制代码规律减少。双向计数器是可增可减，由控制端来决定。
- 按电路集成度分为 **小规模集成计数器** 和 **中规模集成计数器**。

一：同步计数器的分析与设计

在设计同步计数器是由于已经明确了状态数、状态代码和状态迁移关系，所以不需要制作原始状态图、状态化简和状态分配。

1. 二进制计数器的设计

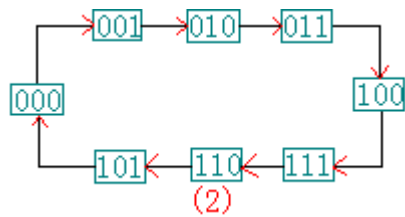
模为 2 的同步计数器称为二进制计数器，它的特点是没有多余状态，触发器的利用率高。它通常是采用自然二进制编码。

例1. 设计一个三位二进制同步递增计数器.

三位二进制的进位模数为 $2^3=8$,它的状态表为:如图(1)所示.状态迁移图为:如图(2)所示

Q_c^n	Q_b^n	Q_a^n	Q_c^{n+1}	Q_b^{n+1}	Q_a^{n+1}
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

(1)



将现态 Q_c^n 、 Q_b^n 、 Q_a^n 作为输入,次态 Q_c^{n+1} 、 Q_b^{n+1} 、 Q_a^{n+1} 作为输出,通过卡诺图可得出各触发器的次态方程为:

$$Q_c^{n+1} = Q_a^n Q_b^n Q_c^n + Q_a^n Q_c^n + Q_b^n Q_c^n = Q_a^n Q_b^n Q_c^n + Q_a^n Q_b^n Q_c^n$$

$$Q_b^{n+1} = Q_a^n Q_b^n + Q_a^n Q_b^n$$

$$Q_a^{n+1} = Q_a^n$$

把求得的次态方程与选用触发器的特征方程作比较,求得各触发器的激励函数为:

$$J_c = Q_a^n Q_b^n$$

$$K_c = Q_a^n Q_b^n$$

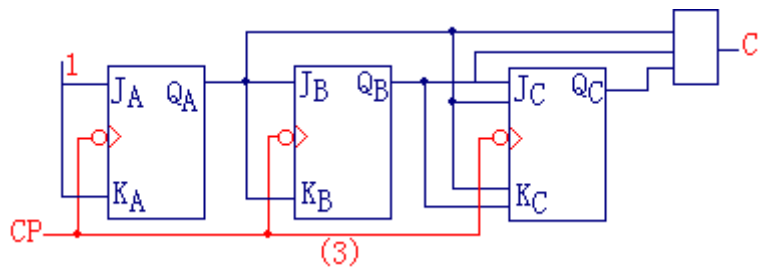
$$J_b = Q_a^n$$

$$K_b = Q_a^n$$

$$J_a = 1$$

$$K_a = 1$$

它的逻辑电路图为:如图(3)所示



位数增多的二进制计数器的设计可按以上方法进行,当位数>5 时,就不能用上面的方法了(卡诺图不易制),从上面的例子我们可以看出:低级触发器除外,每一位触发器的 J,K 激励函数都是由它的低位各触发器原码相与而成.由此就可以设计更多位的二进制计数器了.

二进制的减法计数器的设计与加法相似,只不过状态迁移图不同.

2. 非 2^n 进制计数器

由于这种进制不是 2 的倍数,所以存在着多余状态,在设计中应把这些多余状态作无关项来考虑.在实际中用的最多的是十进制计数器,它需要四个触发器.

例 1. 五级触发器的进位模数最大为:()

- A. 五进制 B. 十进制 C. 十六进制 D. 三十二进制

因为是五级触发器,所以它的最大进位模数为 $2^5=32$,所以答案为 D

例 2. 设计一个模六计数器.

由于 $2^2 < 6 < 2^3$,所以模六计数器需要三级触发器组成.三级触发器有 8 种状态,因此存在着两种多余状态,我们任选其中的六种,它的状态图为:如图(1)所示,

我们通过各级触发器(用 JK 触发器来实现)的

卡诺图可得各级触发器的次态方程为:

$$Q_C^{n+1} = Q_A^n Q_C^n + Q_A^n Q_C^n$$

$$Q_B^{n+1} = Q_B^n Q_C^n + Q_B^n Q_C^n$$

$$Q_A^{n+1} = Q_A^n Q_B^n + Q_A^n Q_B^n$$

$$C = Q_A^n Q_B^n$$

Q_C^n	Q_B^n	Q_A^n	Q_C^{n+1}	Q_B^{n+1}	Q_A^{n+1}	C
0	0	0	1	0	0	0
1	0	0	1	1	0	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
0	1	1	0	0	1	0
0	0	1	0	0	0	1

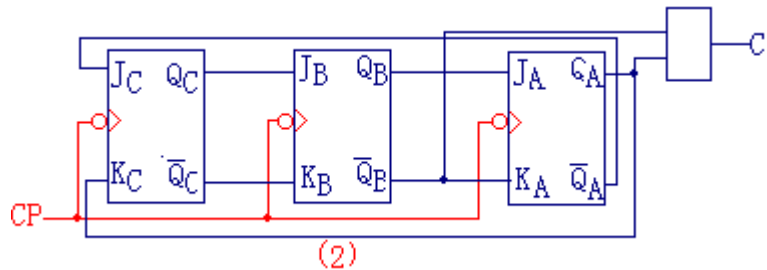
(1)

由次态方程可得激励方程为:

$$J_C = Q_A^n \quad K_C = Q_A^n$$

$$J_B = Q_C^n \quad K_B = Q_C^n$$

$$J_A = Q_B^n \quad K_A = Q_B^n$$



(2)

所的逻辑电路图为:如图(2)所示

这类计数器由于状态没有用完,存在着多余状态,所以它就有一个自启动和自校正问题.

自启动就是当电源合上之后,电路能否进入所用的状态之中的任一状态,如果能则有,否则即无.

自校正就是计数器正常工作时,由于一些原因,使计数状态离开正常的的序列,若经过若干个节拍后电路如能返回正常的计数序列,则有校正能力.如不能,则无校正能力.

注:具有自校正能力的计数器也具有自启动能力.

怎样判断电路是否具有自校正能力呢?

一般是把未用的状态代入所得的次态方程,求得次态,并判断次态是否还是无用状态,若是则表示无自校正能力,若转入有用序列则表示该电路具有自校正能力.

根据上面的结论,来判断一下例 2 是否具有自校正能力.先把没用的两种状态代入次态方程,结果为:如图(3)所示由此可以看出此电路无自校正能力,因此要改进设计.

改进的具体步骤是:

切断 010 与 101 的无效循环序列,强迫使之进入 110,

由于前两级都没有改变,所以只需重新设计第三级即

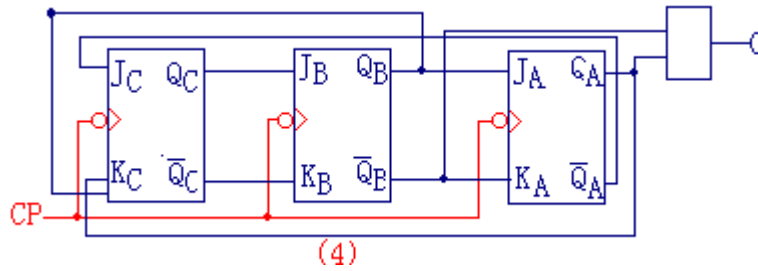
可.

Q_C^n	Q_B^n	Q_A^n	Q_C^{n+1}	Q_B^{n+1}	Q_A^{n+1}	C
0	1	0	1	0	1	0
1	0	1	0	1	0	0

(3)

$$Q_C^{n+1} = Q_A^n Q_C^n + Q_A^n Q_B^n Q_C^n$$

则改进后的逻辑电路图为:如图(4)所示



2. 同步时序电路的分析

它的分析方法和步骤与同步时序电路是一样的.在这里我们就不多说了.

二：异步计数器的分析与设计

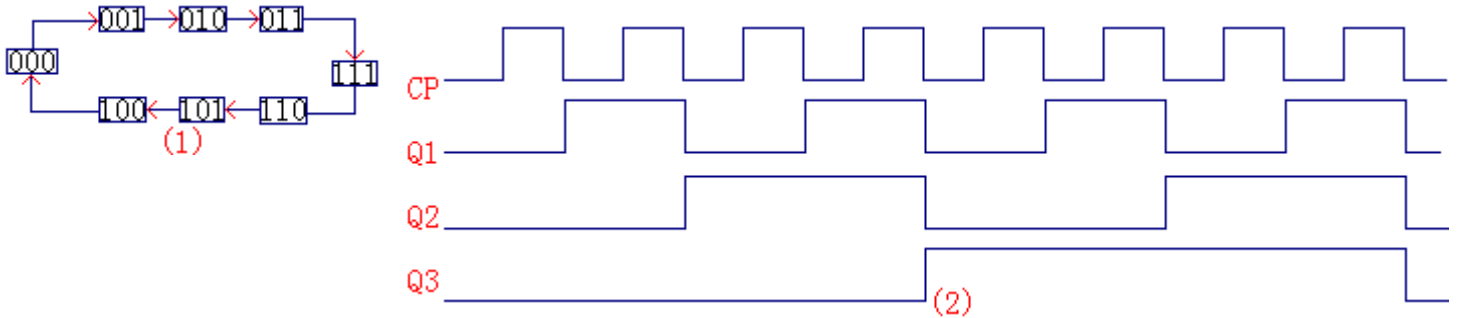
异步计数器的时钟脉冲不是同步的,因此在设计时要特别注意各触发器的时钟信号.

1. 二进制计数器的设计

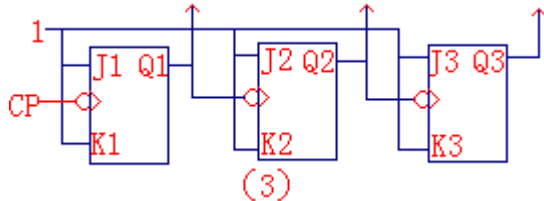
我们通过例子来说明一下.

例 1:设计一个八进制异步递增计数器.

首先我们来画出它的状态图:如图(1)所示,根据状态图再画出电路的输出波形图:如图(2)所示



由波形图我们可以看出各级触发器的时钟脉冲为:CP₁ 为 CP; CP₂ 为 Q₁ 的输出原码; CP₃ 为 Q₂ 的输出原码.我们知道如果没有时钟脉冲触发器是不会翻转的,只有有了时钟脉冲触发器才可能翻转,根据这一点我们可以把各级触发器的输入端置“1”, 所得的逻辑电路图为:如图(3)所示.



2. 非 2ⁿ 进制异步计数器的设计

非 2ⁿ 进制异步计数器设计时主要是判断各触发器的时钟脉冲.它的具体步骤是:

- 先选定状态的迁移关系,并画出波形图.
- 根据波形图,选择时钟,然后的步骤就与同步计数器的设计方法一样.

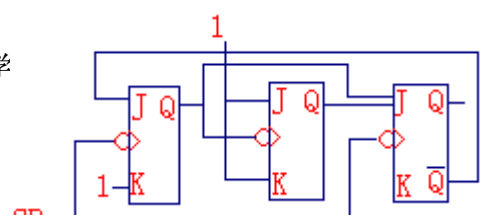
注意:没有时钟脉冲信号沿的状态按无关项处理.

3. 异步计数器的分析

它的分析方法与同步计数器的分析基本相同,区别在于异步时序电路翻转的时间有先有后,只有提供时钟信号,且提供有效的时钟信号沿时,对应的触发器才翻转.

例 2:如图(4)所示的电路,试分析其功能.

该电路是由三级 JK 触发器组成,下降沿触发,异步控制.



激励方程为：
 $J_1=Q_3^n \quad K_1=1$
 $J_2=K_2=1$
 $J_3=Q_1^n Q_2^n \quad K_3=1$

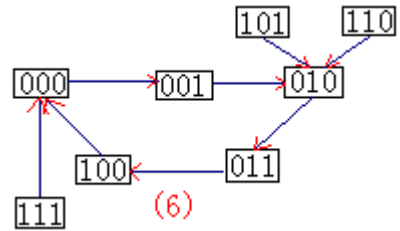
次态方程为：

$Q_1^{n+1}=Q_3^n Q_1^n$
 $Q_2^{n+1}=Q_2^n$
 $Q_3^{n+1}=Q_1^n Q_2^n Q_3^n$

Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	0

(5)

时钟方程为：
 $CP_1=CP$
 $CP_2=Q_1$
 $CP_3=CP$



由次态方程和时钟方程,可列出状态表和状态图如图 (5), (6)所示,由此可得逻辑电路的逻辑功能:该电路是异步模 5 递增计数器,具有自启动能力。

三：集成计数器功能分析及它的应用

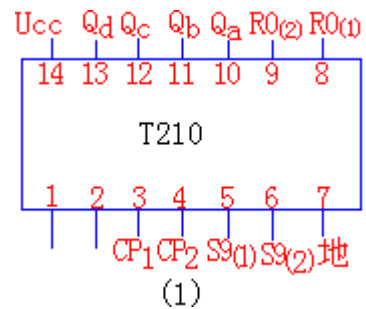
当今集成计数器的品种有很多,按其功能可分为:同步和异步两类。这一节来介绍两种集成计数器它们是:异步集成计数器 T210 和同步集成计数器 T214。在学习时要注意这两种计数器的计数功能、用途以及用集成计数器作为电路的核心部件来设计逻辑电路。

1. 异步集成计数器 T210

T210 异步式 2-5-10 进制计数器是由四个 JK 触发器和两个与非门组成的,它的逻辑符号为:如图(1)所示

输入				输出					
$R_{0(1)}$	$R_{0(2)}$	$S_{9(1)}$	$S_{9(2)}$	CP_1	CP_2	Q_A	Q_B	Q_C	Q_D
1	1	0	x	x	x	0	0	0	0
1	1	x	0	x	x	0	0	0	0
0	x	1	1	x	x	1	0	0	1
x	0	1	1	x	x	1	0	0	1
$\overline{R_{0(1)}} \overline{R_{0(2)}}$	$\overline{S_{9(1)}} \overline{S_{9(2)}}$			CP	0	二进制计数			
=1	=1			0	CP	五进制计数			
				CP	Q_A	8-4-2-1码十进制计数			
				Q_D	CP	5-4-2-1码十进制计数			

(2)



(1)

它的逻辑功能描述如下:(如图(2)所示的功能表)

- 清"0"功能(输出为"0000"):当 $R_{0(1)}$ 、 $R_{0(2)}$ 输入全为高电平, $S_{9(1)}$ 、 $S_{9(2)}$ 有低电平时,各触发器输出为"0",实现清"0"功能,由于时钟不同步,这种清零又被称为"异步清零"
- $R_{0(1)}$ 、 $R_{0(2)}$ 有低电平时,触发器输出为"1001",实现置"9"功能;
- 计数功能:当 $R_{0(1)}$ 、 $R_{0(2)}$ 及 $S_{9(1)}$ 、 $S_{9(2)}$ 有低电平时,各触发器恢复正常功能,实现计数功能,在使用它时一定要按功能表的要求。

例 1.用 T210 组成九进制计数器。

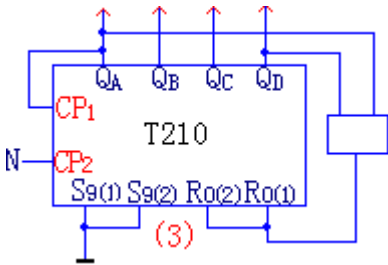
若选用 8421BCD 十进制计数器,初态为"0",则选择"0--8 为有效状态,当计数脉冲为"9"时,输出为 $Q_D Q_C Q_B Q_A="1001"$,立即使它变成为"0000",使计数器返回初态。

它的逻辑电路图为:如图(3)所示,

这种功能是利用反馈法使计数器复"0",这样就可使大模数计数器,改接为小模数计数器.由于 T210 是异步清

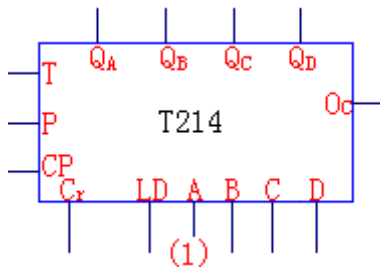
"0",所以需要有一个过渡态.

若需要更大模数的的计数器,可通过级联的方式,增大模数.如:二级 T210 相连可扩展为 2--99 进制计数器.



2. 同步式集成计数器 T214

T214 同步式二--十六进制是由四级 JK 触发器和一些控制门组成.它的逻辑符号为:(如图(1)所示)



输入									输出			
CP	Cr	LD	P	T	A	B	C	D	QA	QB	QC	QD
×	0	×	×	×	×	×	×	×	0	0	0	0
↑	1	0	×	×	A	×	×	×	A	B	C	D
×	1	1	0	×	×	B	×	×	保持 保持 计数			
×	1	1	×	0	×	×	C	×				
↑	1	1	1	1	×	×	×	D				

它的逻辑功能可描述如下:(如图 2 所示的功能表)

异步清零: 当清零控制端 $Cr=0$ 时,立即清零,与 CP 脉冲无关;

同步预置: 当预置端 $LD=0, Cr=1$ 时,在置数输入端 A、B、C、D 预置某个数据,CP 上升沿的时刻,将 ABCD 的数据送入计数器;(必须与脉冲同步使用)

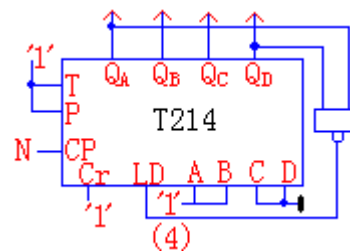
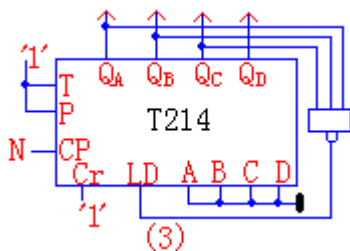
保持: 当 $LD=Cr=1$ 时,控制端 P、T 中有低电平时,就使每级触发器的 $J=K=0$, 处于保持状态;

计数: 当 $LD=Cr=T=P=1$ 时,电路是模 2^4 同步递增计数器,当输出为“1111”时进位输出端 Oc 送出高电平的进位信号,即 $Oc=QAQBQCQD \cdot T=1$ 。

注:由于 T214 由预置端,可以利用这个功能组成任意进制计数器.它也可以象 T210 一样采用反馈法来组成计数器.

例 1.用 T214 的同步预置端构成八进制计数器.

若选择前八种状态,则后面的状态为无效,当计数器输出为“0111”时,经过“与非”门反馈至同步预置端,使 $LD=0$,再来一个时钟脉冲,计数器又预置为“0”,它的电路图为:如图(3)所示



若选择中间八中状态:从“0011”开始,当计数器输出为“1010”时,经过“与非”门反馈至同步预置端使 $LD=0$,再来一个时钟脉冲,计数器又预置为“0”,它的电路图为:如图(4)所示

若想组成更大模数的计数器,也是通过级联的方式,但是要注意 Oc 的连接方式.

第二节：寄存器与移位寄存器

寄存器与移位寄存器是数字系统中常见的部件,寄存器是用来存入二进制代码的,移位寄存器除具有寄存器的功能外,还能将数码移位.

一：寄存器

寄存器中用的记忆部件是触发器，每个触发器只能存一位二进制码。

按接收数码的方式它可分为：单拍式和双拍式。

单拍式：接收数据后直接把触发器置为相应的数据，不考虑初态。

双拍式：接收数据之前，先用复"0"脉冲把所有的触发器恢复为"0",第二拍把触发器置为接收的数据。

二：移位寄存器

移位寄存器具有数码寄存和移位两个功能，在移位脉冲的作用下，数码如向左移一位，则称为左移，反之称为右移。

移位寄存器具有单向移位功能的称为单向移位寄存器，即可向左移也可向右移的称为双向移位寄存器。

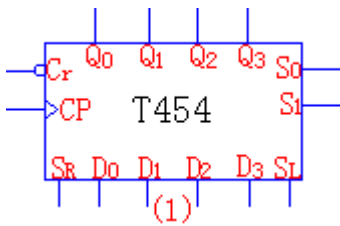
移位寄存器的设计比较容易，因为它的状态受移位功能的限制。

例:若现态为"001"它的次态只有两种可能,分别是:"000"或"010",不可能出现别的情况,否则就没有意义

三:集成移位寄存器

1.集成移位寄存器 T454

T454 是一种用途广泛的集成移位寄存器，它由四个 R-S 触发器和一些门电路组成的四位双向移位寄存器。它的逻辑符号为：（如图（1）所示）



输入										输出			
Cr	Sl	So	CP	SL	Sr	Do	D1	D2	D3	Q0	Q1	Q2	Q3
0	×	×	×	×	×	×	×	×	×	0	0	0	0
1	×	×	0	×	×	×	×	×	×	保持			
1	1	1	↑	×	×	Do	D1	D2	D3	Do	D1	D2	D3
1	0	1	↑	×	0	×	×	×	×	0	Q0 ⁿ	Q1 ⁿ	Q2 ⁿ
1	0	1	↑	×	1	×	×	×	×	1	Q0 ⁿ	Q1 ⁿ	Q2 ⁿ
1	1	0	↑	0	×	×	×	×	×	Q1 ⁿ	Q2 ⁿ	Q3 ⁿ	0
1	1	0	↑	1	×	×	×	×	×	Q1 ⁿ	Q2 ⁿ	Q3 ⁿ	1
1	0	0	×	×	×	×	×	×	×	保持			

它的功能描述如下：（如功能表（2）所示）
 直接清零：当清零控制端 $Cr=0$ 时，立即清零，与其它控制端无关；
 保持：当 CP 没来或控制端 S_0S_1 全为低电平时，寄存器处于保持状态；

送数：当控制端 S_0S_1 全为高电平时，寄存器处于送数状态；

移位：当控制端 $S_1S_0=01$ 时，寄存器向右移位； $S_1S_0=01$ 时，则寄存器向左移位， S_L 是左移串行数据的输入端， S_R 是右移串行数据的输入端。

2. 移位型计数器

移位计数器就是指以移位寄存器为主题构成的同步计数器.它的设计方法与同步计数器基本相同,不同的是它的状态受移位关系的约束,因此它的状态不能任意指定.

它的设计步骤是:

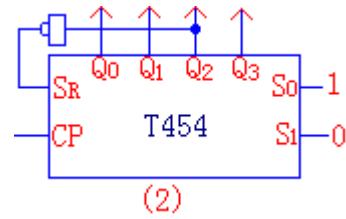
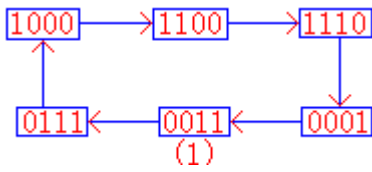
根据题意写出状态迁移关系;

根据迁移关系求出反馈函数;

根据前两步的结果运用器件，并画出逻辑电路图。

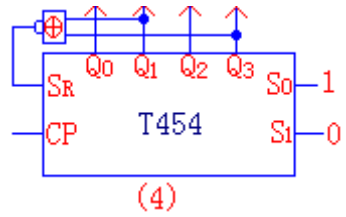
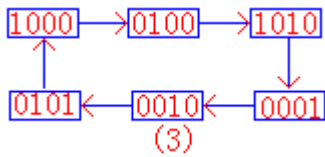
例1. 用 T454 和辅以其它组件设计模 6 移位型计数器.

(1) 任选六种状态，它们的状态迁移关系是：如图（1）



根据状态卡诺图，我们可的电路图为：如图（2）

（2）任选另外六种状态，它们的状态迁移关系是：如图（3）



根据状态卡诺图，我们可画出它的逻辑电路图为：如图（4）

第三节：序列信号发生器

序列信号是指在同步脉冲作用下循环地产生一串周期性的二进制信号.能产生这种信号的逻辑器件就称为序列信号发生器.根据结构不同,它可分为反馈移位型和计数型两种.

一:移位型序列信号发生器

1. 移位型序列信号发生器的组成

移位型序列信号发生器是由移位寄存器和组合电路两部分构成，组合电路的输出，作为移位寄存器的串行输入。由 n 位移位寄存器构成的序列信号发生器所产生的序列信号的最大长度为： $P=2^n$

2. 移位型序列信号发生器的设计(我们通过例题来说明)

例 1. 试设计一个 00011101 序列信号发生器

首先确定移位寄存器的位数,并画出编码状态图,并找出迁移关系.

$P=8$ ，因此只需要 T454 中的三位,按序列信号三位一组去划分(有八中状态),它的转换关系是:如图(1)

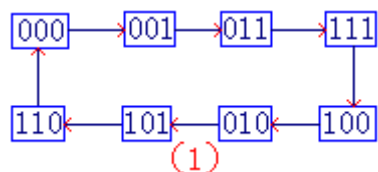
再作出 T454 的操作图:如图(2)

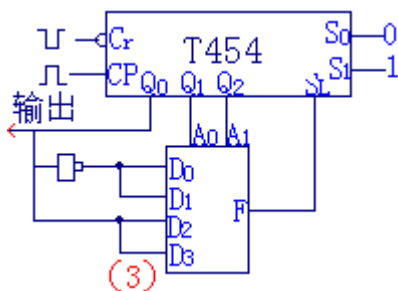
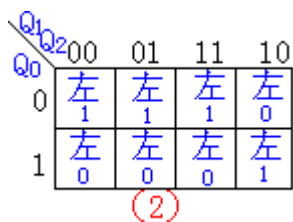
由图可见这个电路只进行左移操作,因此可以判定

出 S_0S_1 的值.

最后根据移位寄存器每移位所移动的数码用四选一数

据选择器实现 S_1 , 它逻辑电路图为:如图(3)





通过这个例子我们发现移位型序列信号发生器设计与同步时序电路的设计相象。

二:计数型序列信号发生器

1. 计数型序列信号发生器组成与特点

计数型序列信号发生器能产生多组序列信号,这是移位型发生器所没有的功能.计数型序列信号发生器是由计数器和组合电路构成的,序列的长度 P 就是计数器的模数。

2. 计数型序列信号发生器的设计(通过例题来说明)

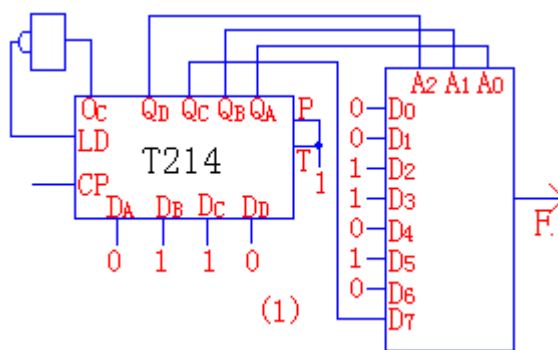
例 1. 设计 1101000101 序列信号发生器

由于序列长度已经知道 $P=10$, 所以先用 T214

设计一个模 10 的计数器,

再用八选一数据选择器实现序列输出, 把计数器的输出端作为数据选择器的地址变量。

它的电路图为: 如图 (1)



第八章 脉冲产生电路和定时电路

时钟脉冲是数字系统中一个非常重要的因素。形成脉冲的电路是利用惰性元件（电容 C 或电感 L）的充放电现象。脉冲电路是由两部分组成：惰性电路和开关。开关的作用是破坏稳态，使电路出现暂态

我们这一章主要是学习由 555 定时电路所构成的脉冲电路。

我们在学习时把这一章的内容分为三组，它们分别是：

- 555 定时电路
- 单稳态电路
- 多谐振荡器和施密特电路

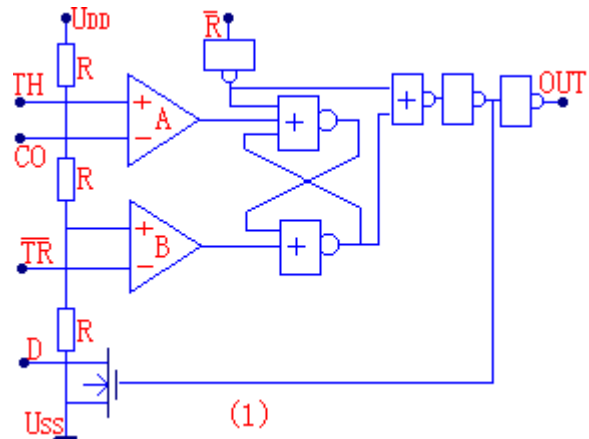
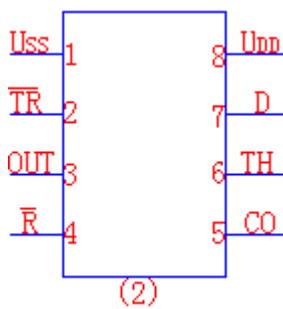
第一节：555 定时电路

555 定时电路的应用十分广泛, 它由 TTL 集成定时电路和 CMOS 集成定时电路, 这二者功能完全相同, 不同之处是:TTL 集成定时电路的驱动能力比 CMOS 集成定时电路大.

一：555 定时电路的组成

555 定时电路是由三个 5 千欧电阻组成分压器、两个高精度电压比较器、一个基本 R-S 触发器、一个作为放电通路的管子及输出驱动电路组成。它的逻辑电路图为：如图 (1) 所示

它的逻辑符号为：如图 (2) 所示



功能描述：(功能表如表 3 所示)

- 当输入端 R 为低电平时, 不管别的输入端为何种情况, 输出为低电平, CMOS 管工作。
- 当引脚 6 的输入电平大于 $2/3U_{dd}$ 并且引脚 2 的输入电平大于 $1/3U_{dd}$, 输出为低电平, CMOS 管工作
- 当引脚 6 的电平小于 $2/3U_{dd}$ 并且引脚 2 的输入电平大于 $1/3U_{dd}$, 输出为原状态.
- 当引脚 2 的电平小于 $1/3U_{dd}$, 电路输出为高电平, NMOS 管关断.

TH	TR	R̄	OUT	D
×	×	L	L	接通
$> \frac{2}{3}U_{dd}$	$> \frac{1}{3}U_{dd}$	H	L	接通
$< \frac{2}{3}U_{dd}$	$> \frac{1}{3}U_{dd}$	H	原状态	原状态
×	$< \frac{1}{3}U_{dd}$	H	H	关断

图中的L为低电平, H为高电平 (3)

- 例1. 555 集成电路, 改变电压控制端(引脚 5)的电压可改变()
 例2. A. 高触发端, 低触发端的电平 B. 555 定时电路的高低电平
 C. 开关放电管的开关电平

D.置"0"端 R 的电平

答案为: A

例 2.555 定时电路 R 端的作用是什麼?

答:它的作用是:复"0".不管 555 定时电路是何种状态,只要 R 输入为低电平,输出即为低电平;只有它输入为高电平时定时电路才工作

第二节：单稳态电路

单稳态电路只有一个稳定状态。在外界触发脉冲的作用下，电路从稳态翻转到暂态，在暂态维持一段时间之后，又返回稳态，并在输出端产生一个矩形脉冲。

一.单稳态的电路组成

它是由 CC7555 定时电路构成，电路图为：如图（1）所示

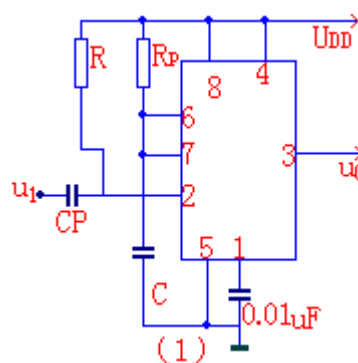
● 它的工作特点：

电路只有一个稳定状态；

- 当外界触发脉冲来后，电路从稳态翻转到暂态，并在暂态停留一段时间，而且在输出端产生一个宽度为 T_w 的矩形脉冲。

它的应用： 在数字系统中，单稳态电路常用于整形。

即：把不规则的波形转换成宽度、幅度相同的波形。



例 1. 怎样改变输出脉冲的宽度(即延迟时间)呢?

答：有三种方法：1、改变电阻 R；2、改变电容 C；3、改变控制电压端的接法。

例 2. 如图(1)所示:改变控制电压端(引脚 5)的电压值,可改变()

答案为: D

A. 输出电压的高低电平 B. 输出电压的周期 C. 对输出波形无影响 D. 输出电压的脉冲的宽度

第三节：多谐振荡器和施密特电路

一:多谐振荡器(只要求知道它的工作特点和用途)

多谐振荡器的工作特点是：

- 电路不具有稳定状态，但是具有两个暂稳态；
- 不需外加触发信号，电路就能自动产生矩形波的输出；
- 电路工作就是在两个暂稳态之间来回转换。

它的用途是：产生定量的矩形脉冲。

二：施密特电路

将 555 定时电路中的 2，6 引脚连接，就构成了施密特电路。如图（1）所示

- 施密特电路的工作特点是：

有两个稳定状态，但是这两个稳定状态要靠输入信号来维持，而且转换也要靠输入信号的转换来实现

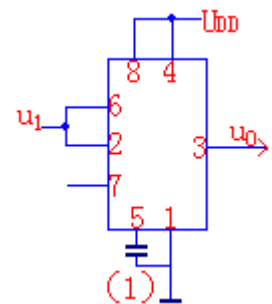
- 输出电压和输入电压具有迂回特性，抗干扰能力强。

施密特电路的用途是：

- 整形：将不好的矩形波，变为较好的矩形波；

波形转换：将三角波、正弦波和其它波形转换为矩形波；转换后的输出波形与输入波形相同；

- 幅度鉴别：可以将输入信号中的幅度大于某一数值得信号检测出来。



例1. 施密特电路具有()

- A.一个稳定状态 B.两个稳定状态 C.三个稳定状态 D.没有稳定状态 答案为: B

第九章 数模、模数转换电路

自然界所存在的物理量，大多是模拟量，它们是不能直接送入数字计算机里进行处理的，必须先把这些模拟量转换成数字信号，再送入数字系统中进行处理，处理完之后，然后把处理完的数字信号转换为模拟量，再送去控制元件去执行。这一节我们来学习模拟量和数字信号的转换器的基本工作原理。

我们在学习时把这一章的内容分为两节，它们是：

- DAC
- ADC

第一节： DAC

DAC 的任务就是将输入数字信号, 转换为与输入数字量成正比的输出模拟电流 i_o 或电压 u_o . 这一节要求我们熟悉 DAC 的转换公式及分辨率的计算

一: DAC 的转化特性

DAC 电路的输入是二进制数字信息。(用 B 表示) 其中最低位为 B_0 和最高位为 B_{n-1} , 它们的权位分别为 2^0 和 2^{n-1} , 则: $B = (B_{n-1}2^{n-1} + B_{n-2}2^{n-2} + \dots + B_12^1 + B_02^0)$

它的输出是与输入数字量成正比的电压 u_o 或电流 i_o , 用公式表示为:

$$u_o(\text{或 } i_o) = K * B \quad (K \text{ 为比例常数})$$

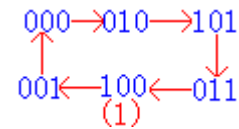
例1. 已知某 DAC 转换电路, 输入三位数字量, 参考电压 $E_R = -8V$, 当输入的数字量 $D_2D_1D_0$ 如图(1)所示时, 求相应的输出模拟量 u_o , 并对应时钟 CP 的波形画出 u_o 的波形

因为 $u_o = -E_R / 2^n * B$, 按输入二进制的序列可求得 u_o 的值为:

$D_2D_1D_0 = 000$ 时: $u_o = 0V$ $D_2D_1D_0 = 010$ 时: $u_o = 2V$

$D_2D_1D_0 = 101$ 时: $u_o = 5V$ $D_2D_1D_0 = 011$ 时: $u_o = 3V$

$D_2D_1D_0 = 100$ 时: $u_o = 4V$ $D_2D_1D_0 = 001$ 时: $u_o = 1V$

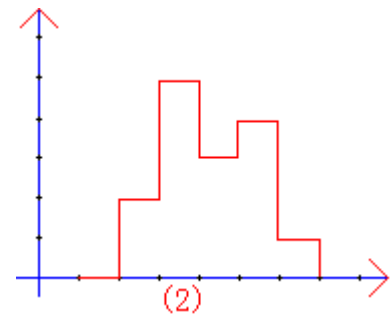


根据求得 u_o 可的波形图为: 如图(2)所示

二: 分辨率

分辨率就是电路所能分辨的最小输出电压 U_{LSB} 与最大输出电压 U_M 之比。它用公式可表示为:

$$\text{分辨率} = U_{LSB} / U_M = 1 / (2^n - 1) \quad (n \text{ 为输入数字的位数})$$



第二节: ADC

ADC 就是将模拟信号转换为数字信号的器件。

ADC 的组成与 ADC 的电路形式

ADC 电路通常由两部分组成, 它们是: 采样、保持电路和量化、编码电路。其中量化、编码电路是最核心的部件, 任何 ADC 转换电路都必须包含这种电路。

ADC 电路的形式很多, 通常可以并为两类:

间接法: 它是将采样-保持的模拟信号先转换成与模拟量成正比的时间或频率, 然后再把它转换为数字量。这种通常是采用时钟脉冲计数器, 它又被称为计数器式。

它的工作特点是: 工作速度低, 转换精度高, 抗干扰能力强。

直接法：通过基准电压与采样-保持信号进行比较，从而转换位数字量。

它的工作特点是：工作速度高，转换精度容易保证。

例 1. 模数转换电路包括____、____、____、____ 四个过程。

答案是： 采样、保持、量化、编码

第十章：大规模集成电路

在数字系统中，用来存贮信息的器件被称为存贮器。存贮器是由许多存贮单元组成，每个存贮单元可存放一位二进制数。通常一个二进制代码由若干位二进制数组成，我们称这样的二进制代码为一个字，它所包含的二进制数的位数称为字长。因此存放一各字长为 A 的数就需要 A 个存贮单元，我们称存放这一个字的 A 个存贮单元为一个信息单元。存贮器的容量是字数 B 与字长 A 的乘积，这也是存贮器中包含存贮单元的总数。

这一章我们只学习只读存贮器 ROM (Read Only Memory) 的及其应用和可编程的逻辑阵列 PLA (Programmable Logic Array)。

我们在学习时把这一章的内容分为两节，它们是：

- ROM 及其应用
- 第二节：PLA 及其应用

第一节：ROM 及其应用

ROM 在数字系统中的应用十分广泛。在使用时，它只能读出信息，而无法写入信息。

一:ROM 的组成

ROM 的电路结构包含三个主要部分：

存储矩阵：它是由许多存储单元排列而成，而且每个存储单元都被编为一个地址(地址变量)。

地址译码器：它是将输入的地址变量译成相应的地址控制信号，该控制信号可将某存储单元从存储矩阵中选出来，并将存储在该单元的信息送至输出缓冲器。

输出缓冲器：它是作为输出驱动器和实现输出的三态控制。

二：ROM 的应用（通过例题说明）

例 1：把表（1）的内容用 ROM 电路表示出来。

从表中我们可以得出如下表达式:

$$D_3 = A_1 A_0 + A_1 \bar{A}_0$$

$$D_2 = A_1 A_0 + A_1 \bar{A}_0$$

$$D_1 = A_1 A_0 + A_1 \bar{A}_0$$

$$D_0 = A_1 A_0 + A_1 \bar{A}_0 + A_1 A_0$$

A ₁	A ₀	D ₃	D ₂	D ₁	D ₀
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	0	1
1	1	0	0	1	1

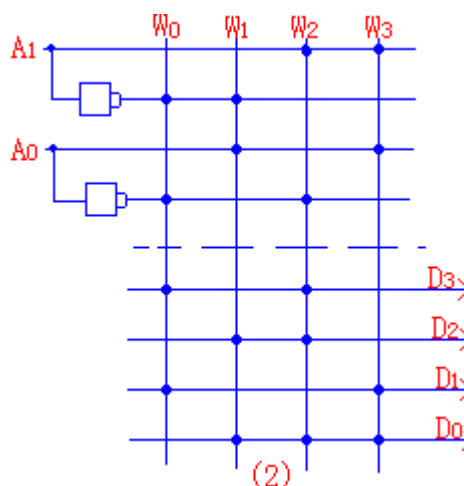
(1)

将地址变量作为逻辑变量, 则地址译码器提供的每个最小项, 即相当于“与”运算。每一位线, 对最小项实现“或”运算, 得的 ROM 阵

列为: 如图(2)所示.

从例题我们可以看出: ROM 电路是将逻辑函数通过与—或网络的形式表示出来。通过地址译码器形成输入变量的所有的最小项 (实现“与”运算), 再通过存贮矩阵实现“或”运

算, 这样就形成了各个输出函数。



三: ROM 的扩展

存贮器的容量是字数与字长的乘积。ROM 的扩展分为字扩展和位扩展, 位扩展即字长扩展。

第二节: PLA 及其应用

在使用 ROM 时, 由于它的地址译码器是固定的, 因此不能对函数进行化简, 从而多占了 ROM 芯片的面积。为了解决这个问题, 我们就使用可以对函数进行化简的器件——PLA

一: PLA

PLA 是能够编程的逻辑器件。它能够对逻辑“与”、“或”阵列进行编程, 利用 PLA, 可以很方便的实现组合逻辑和时序逻辑。

例：1 用 PLA 实现一位二进制全加器

全加器的最简逻辑表达式为（参见第四章第三节）：

$$S = ABC_{i-1} + AB\bar{C}_{i-1} + A\bar{B}C_{i-1} + \bar{A}BC_{i-1}$$

$$C_i = AB + AC_{i-1} + BC_{i-1}$$

该式中共有七个乘积项，它们是：

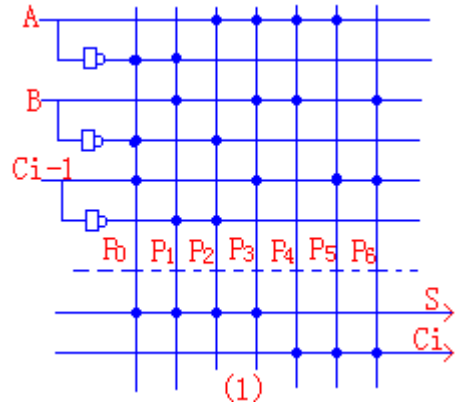
$$P_0 = ABC_{i-1} \quad P_1 = AB\bar{C}_{i-1} \quad P_2 = A\bar{B}C_{i-1} \quad P_3 = \bar{A}BC_{i-1}$$

$$P_4 = AB \quad P_5 = AC_{i-1} \quad P_6 = BC_{i-1}$$

根据上述，可得出 PLA 全加器的阵列结构图为：

如图（1）所示

由此可见：PLA 相当于“与”或“或”阵列均可编程的 ROM.



单片机自学网-----期待着与你共同进步！