

《数字逻辑》

（第七讲）

厦门大学信息学院软件工程系 曾文华

2023年12月11日

课程内容

- 全书共9章：

第1章 基本知识

第2章 逻辑代数基础

第3章 集成门电路与触发器

第4章 组合逻辑电路

第5章 同步时序逻辑电路

第6章 异步时序逻辑电路

第7章 中规模通用集成电路及其应用

第8章 可编程逻辑器件

第9章 综合应用举例



第7章 中规模通用集成电路及其应用

- 7.1 常用中规模组合逻辑电路
- 7.2 常用中规模时序逻辑电路
- 7.3 常用中规模信号产生与变换电路

7.1 常用中规模组合逻辑电路

7.1.1 二进制并行加法器

7.1.2 译码器和编码器

7.1.3 多路选择器和多路分配器

• 7.1.1 二进制并行加法器

– 按照进位方式不同分为：**串行进位**加法器、**超前进位**加法器等两种类型。

– 1、构成思想

• (1) 串行进位二进制并行加法器

– 图7.1：4位**串行进位**二进制并行加法器的结构框图（由4个全加器FA构成）。

– **缺点**：最高位必须等到各低位全部相加完成并送来进位信号后，才能产生运算结果。运算速度较慢，并且位数越多，速度就越低。

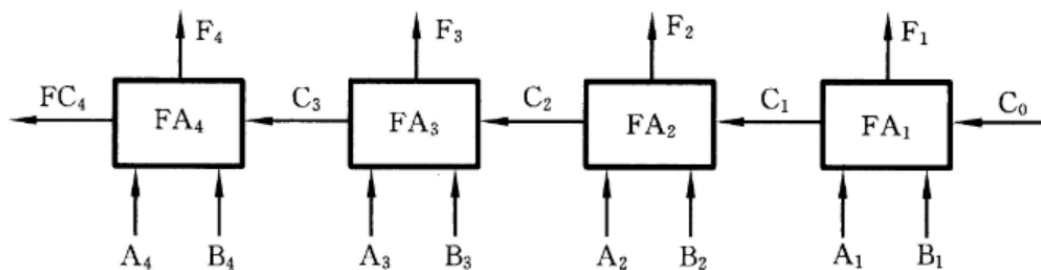


图 7.1 4 位串行进位二进制并行加法器的结构框图

• (2) 超前进位二进制并行加法器

– 根据全加器FA的逻辑公式：

- $F_i = A_i \oplus B_i \oplus C_{i-1}$
- $C_i = A_i \cdot B_i \cdot C_{i-1} + A_i \cdot B_i \cdot C_{i-1} + A_i \cdot B_i \cdot C_{i-1} + A_i \cdot B_i \cdot C_{i-1} = (A_i \oplus B_i) \cdot C_{i-1} + A_i \cdot B_i$

– 定义：

- 进位产生函数： $G_i = A_i \cdot B_i$
- 进位传递函数： $P_i = A_i \oplus B_i$

– 则有：

- $F_i = A_i \oplus B_i \oplus C_{i-1} = P_i \oplus C_{i-1}$
- $C_i = P_i \cdot C_{i-1} + G_i$

– 4位二进制并行加法器各位的进位分别为：

- $C_1 = P_1 \cdot C_0 + G_1$
- $C_2 = P_2 \cdot C_1 + G_2 = P_2 \cdot P_1 \cdot C_0 + P_2 \cdot G_1 + G_2$
- $C_3 = P_3 \cdot C_2 + G_3 = P_3 \cdot P_2 \cdot P_1 \cdot C_0 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot G_2 + G_3$
- $C_4 = P_4 \cdot C_3 + G_4 = P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot C_0 + P_4 \cdot P_3 \cdot P_2 \cdot G_1 + P_4 \cdot P_3 \cdot G_2 + P_4 \cdot G_3 + G_4$

– 因为 $C_1 \sim C_4$ 只是 $G_1 \sim G_4$ 、 $P_1 \sim P_4$ 、 C_0 的函数，而 $G_1 \sim G_4$ 、 $P_1 \sim P_4$ 又只是 $A_1 \sim A_4$ 、 $B_1 \sim B_4$ 的函数，因此 $C_1 \sim C_4$ 只是 $A_1 \sim A_4$ 、 $B_1 \sim B_4$ 、 C_0 的函数。

– 因此在输入 $A_1 \sim A_4$ 、 $B_1 \sim B_4$ 、 C_0 给定后，可以立即得到 $C_1 \sim C_4$ 以及 $F_1 \sim F_4$ ，从而有效地提高运算速度。

– 根据 $G_1 \sim G_4$ 、 $P_1 \sim P_4$ 产生 $C_1 \sim C_4$ 的电路称为超前进位发生器，采用超前进位发生器的加法器称为超前进位加法器，也称先行进位加法器或并行进位加法器。

– 2、典型芯片

- **74283**为**4位超前进位二进制加法器**芯片，其引脚图及逻辑符号如图7.2所示，图中的 FC_4 即为 C_4 。

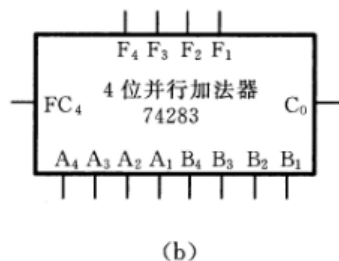
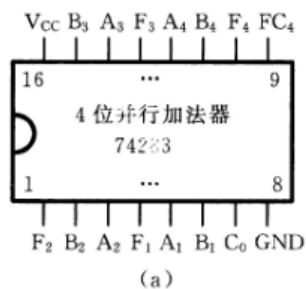
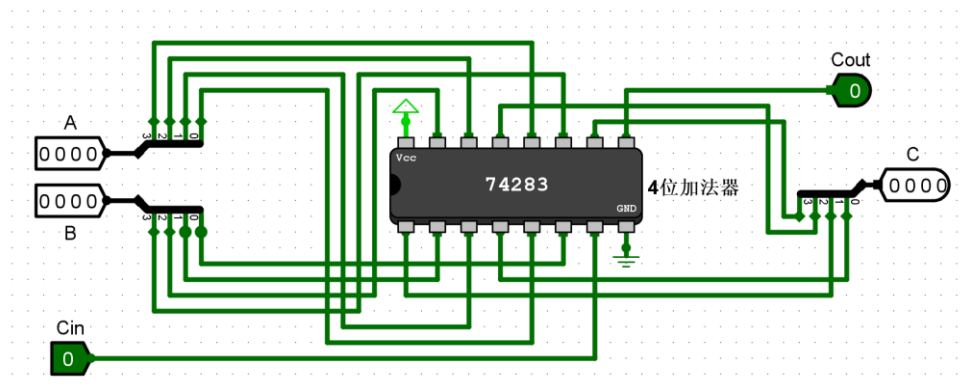


图 7.2 74283 的引脚排列图和逻辑符号



Logisim中的74283芯片

– 3、应用举例

- 利用二进制并行加法器（**74283**芯片）还可以实现代码转换、二进制减法运算、二进制乘法运算、十进制加法运算等功能。

- **例7.1：**用4位二进制并行加法器（74283）设计一个8421码转换成余3码的代码转换电路。
- 解：
 - 余3码=8421码+3。
 - 转换电路见图7.3。

表 1.3 常用的 3 种 BCD 码

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

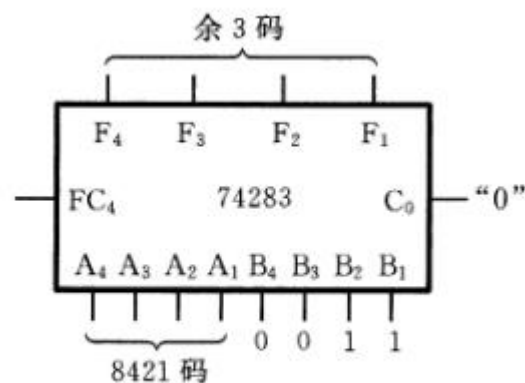


图 7.3 逻辑电路

- **例7.2:** 用4位二进制并行加法器（74283）设计一个4位二进制并行加法/减法器。

• 解:

— **A-B=A+/B+1**, 例如:

- $A=6=0,110$ $B=5=0,101$ $/B=1,010$ $A+/B+1=0,110+1,010+1=0,001=1=6-5。$
- $A=5=0,101$ $B=6=0,110$ $/B=1,001$ $A+/B+1=0,101+1,001+1=1,111=-1=5-6。$

— 4位二进制并行加法/减法器电路如图7.4所示。

— $M=0$, $A=a$, $B=b \oplus 0=b$, $s=F=A+B+C_0=A+B+0=\mathbf{a+b}$ 。

— $M=1$, $A=a$, $B=b \oplus 1=/b$, $s=F=A+B+C_n=a+/b+1=\mathbf{a-b}$ 。

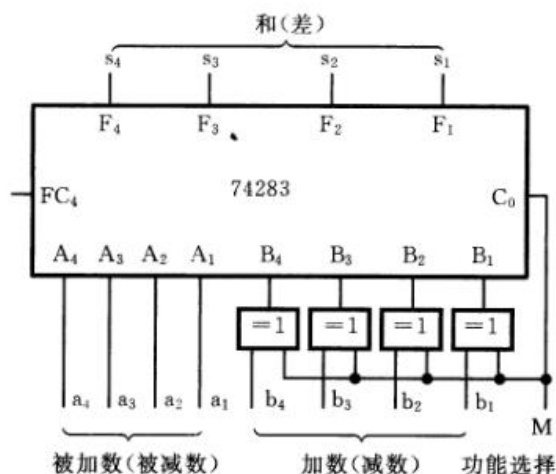


图 7.4 逻辑电路

- **例7.3：**用4位二进制并行加法器（74283）设计一个用余3码表示的1位十进制数加法器。

• 解：

- 余3码加法规则：如果相加结果无进位产生，则“和”需要减3；如果相加结果有进位产生，则“和”需要加3。例如：

- $1+2=0100+0101=1001$ 无进位 “和” $=1001-3=1001-11=0110=3$
- $5+6=1000+1001=10001$ 有进位 “和” $=10001+3=10001+11=10100=11$

- 用余3码表示的1位十进制数加法器电路如图7.5所示：第I片74283完成余3码相加，第II片74283对相加结果进行修正。

- 当第I片相加结果无进位时，进位输出=0，第II片的 $A_4 \sim A_1=1101=-3$ ，即完成减3的操作。
- 当第I片相加结果有进位时，进位输出=1，第II片的 $A_4 \sim A_1=0011=3$ ，即完成加3的操作。

表 1.3 常用的 3 种 BCD 码

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

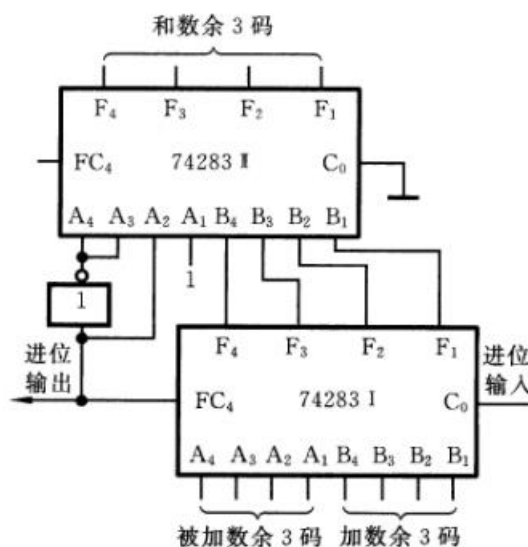


图 7.5 逻辑电路

• **例7.4:** 用4位二进制并行加法器（74283）实现4位二进制乘法器的逻辑功能。

• 解:

- 设 $X=x_3x_2x_1x_0$, $Y=y_3y_2y_1y_0$, $Z=X*Y=z_7z_6z_5z_4z_3z_2z_1z_0$, 运算过程如下（左下图）。
- 可以用16个与门、3个4位并行加法器（74283）实现上述运算过程，具体电路如图7.6所示。

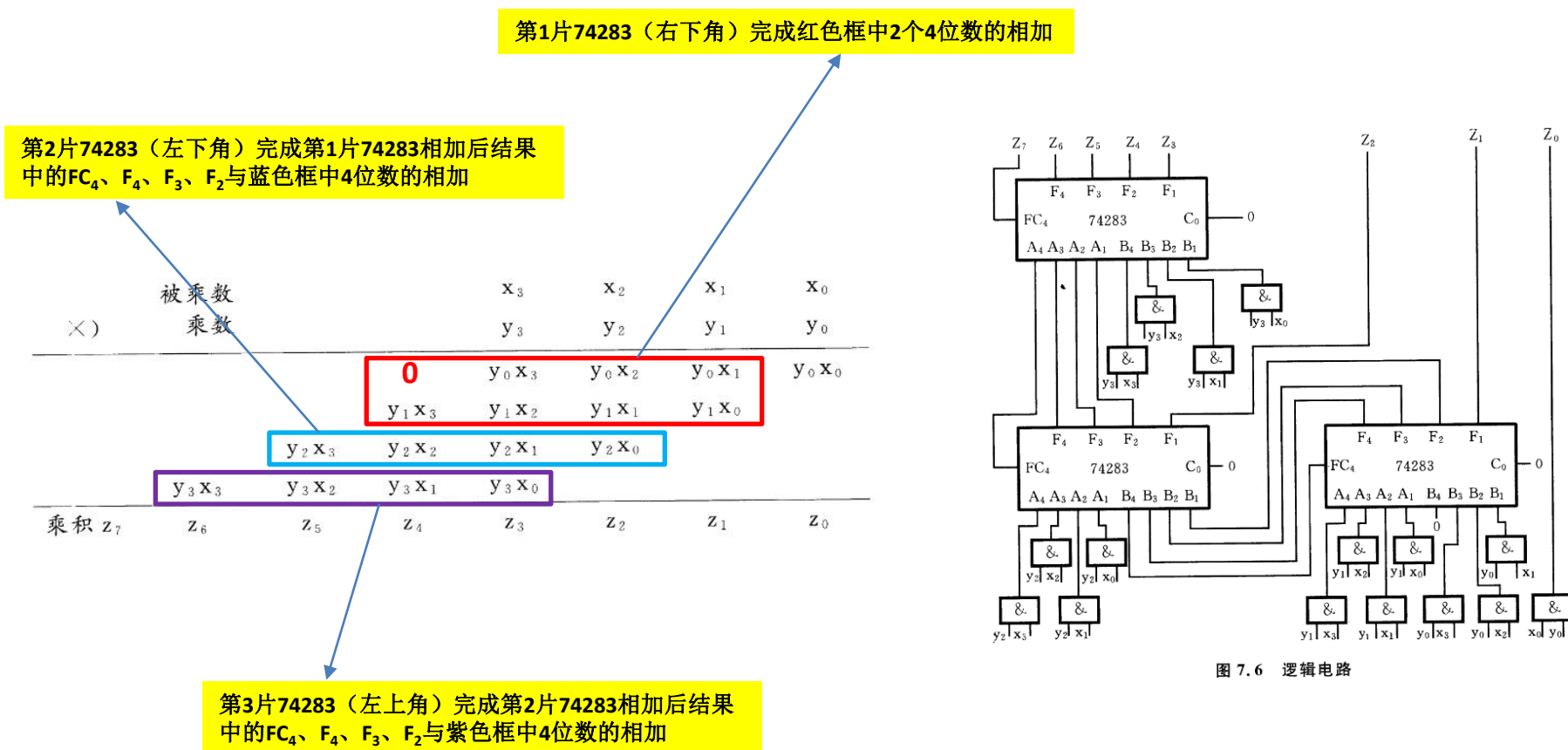


图 7.6 逻辑电路

• 7.1.2 译码器和编码器

– 1、译码器

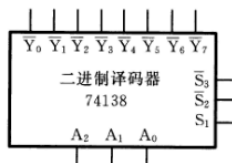
- **译码器 (Decoder)** 的功能是对特定含义的输入代码进行“翻译”，将其转换成相应的输出信号。常见的译码器有：**二进制译码器**、**二-十进制译码器**、**数字显示译码器**。

• (1) 二进制译码器

- 二进制译码器一般具有 n 个输入端、 2^n 个输出端和一个（或多个）使能输入端。
- 常见的有**2-4译码器**、**3-8译码器**、**4-16译码器**。
- 图7.7: **3-8译码器74138芯片**的引脚图及逻辑符号。表7.7: **74138芯片的真值表**。
 - » $S_1=1, /S_2=0, /S_3=0$: 输出Y对输入A进行译码。
 - » $S_1=0, /S_2=d, /S_3=d$: 输出Y=1。
 - » $S_1=d, /S_2=1$ 或 $/S_3=1$: 输出Y=1。



(a)

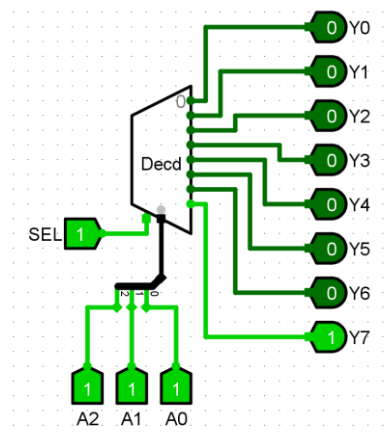


(b)

图 7.7 74138 译码器的引脚排列图和逻辑符号

表 7.1 74138 译码器真值表

输 入					输 出							
S_1	$S_2 + S_3$	A_2	A_1	A_0	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	\bar{Y}_4	\bar{Y}_5	\bar{Y}_6	\bar{Y}_7
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0
0	d	d	d	d	1	1	1	1	1	1	1	1
d	1	d	d	d	1	1	1	1	1	1	1	1



• **例7.5:** 用3-8译码器74138和与非门实现全减器的功能。

• 解:

- 全减器: $\{G_i, D_i\} = A_i - B_i - G_{i-1}$ 。其中 D_i 为差, G_{i-1} 为低位借位, G_i 为高位借位。全减器的真值表见表7.2。
- 由真值表可以得到下面的逻辑公式:
 - $D_i = m_1 + m_2 + m_4 + m_7 = \overline{m_1} \cdot \overline{m_2} \cdot \overline{m_4} \cdot \overline{m_7}$
 - $G_i = m_1 + m_2 + m_3 + m_7 = \overline{m_1} \cdot \overline{m_2} \cdot \overline{m_3} \cdot \overline{m_7}$
- 因为3-8译码器的输出就是8个最小项 ($m_0 \sim m_7$) , 因此可以通过3-8译码器实现上述逻辑公式, 具体见图7.8 (译码器的输入接 A_i 、 B_i 、 G_{i-1} , 控制端: $S_1=1$ 、 $\overline{S_2}=0$ 、 $\overline{S_3}=0$) 。

表 7.2 全减器真值表

输 入			输 出	
A_i	B_i	G_{i-1}	D_i	G_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

够减, 0-0-0=0, 因此 $D_i=0$ 、 $G_i=0$

不够减, 借1位, 相当于 $10+0-0-1=1$, 因此 $D_i=1$ 、 $G_i=1$

不够减, 借1位, 相当于 $10+0-1-0=1$, 因此 $D_i=1$ 、 $G_i=1$

不够减, 借1位, 相当于 $10+0-1-1=0$, 因此 $D_i=1$ 、 $G_i=0$

够减, 1-0-0=1, 因此 $D_i=0$ 、 $G_i=1$

够减, 1-0-1=0, 因此 $D_i=0$ 、 $G_i=0$

够减, 1-1-0=0, 因此 $D_i=0$ 、 $G_i=0$

不够减, 借1位, 相当于 $10+1-1-1=1$, 因此 $D_i=1$ 、 $G_i=1$

够减, 0-0-0=0, 因此 $D_i=0$ 、 $G_i=0$

不够减, 借1位, 相当于 $10+0-0-1=1$, 因此 $D_i=1$ 、 $G_i=1$

不够减, 借1位, 相当于 $10+0-1-0=1$, 因此 $D_i=1$ 、 $G_i=1$

不够减, 借1位, 相当于 $10+0-1-1=0$, 因此 $D_i=1$ 、 $G_i=0$

够减, 1-0-0=1, 因此 $D_i=0$ 、 $G_i=1$

够减, 1-0-1=0, 因此 $D_i=0$ 、 $G_i=0$

够减, 1-1-0=0, 因此 $D_i=0$ 、 $G_i=0$

不够减, 借1位, 相当于 $10+1-1-1=1$, 因此 $D_i=1$ 、 $G_i=1$

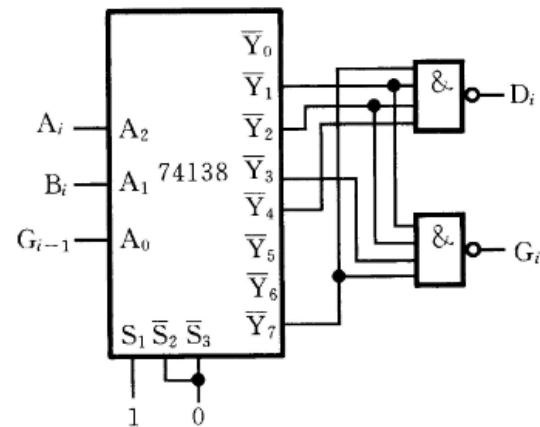


图 7.8 逻辑电路

• **例7.6:** 用译码器和与非门实现逻辑函数: $F(A,B,C,D)=\sum m(2,4,6,8,10,12,14)$ 。

• 解:

– 因为有4个变量, 因此需要使用4-16译码器实现上述逻辑函数。

– 也可以使用2个3-8译码器实现4-16译码的功能 (见左下图)。

• A=0时, 第I片3-8译码器工作、第II片3-8译码器不工作, BCD=000~111, 对应第I片3-8译码器的8个输出。

• A=1时, 第I片3-8译码器不工作、第II片3-8译码器工作, BCD=000~111, 对应第II片3-8译码器的8个输出。

– 因此可以使用2个3-8译码器实现上述逻辑函数 (见图7.9)。

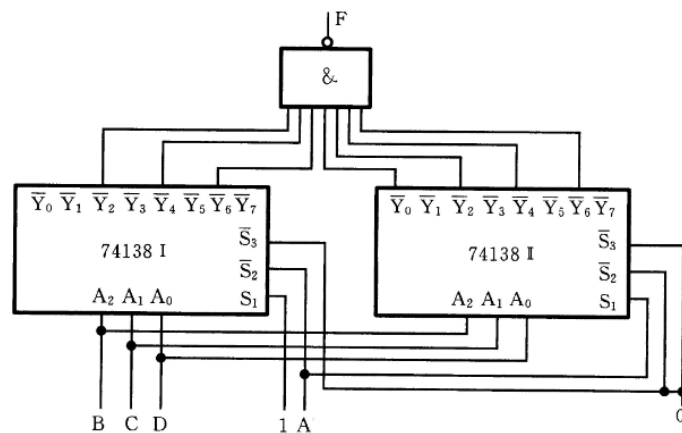
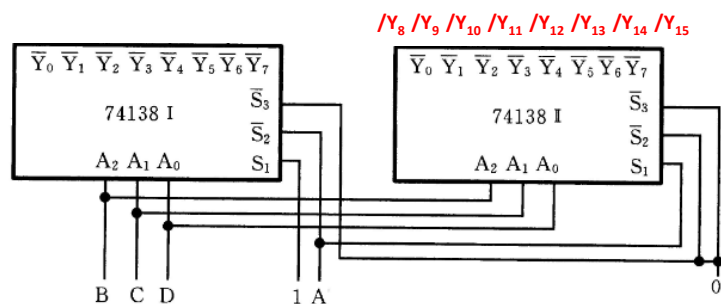
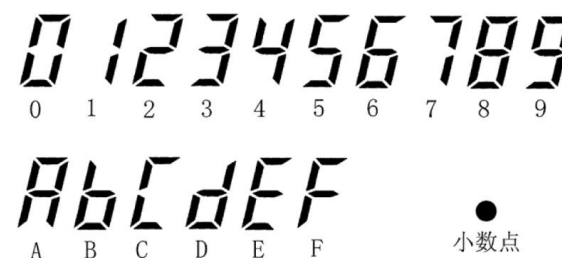
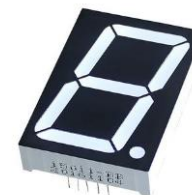
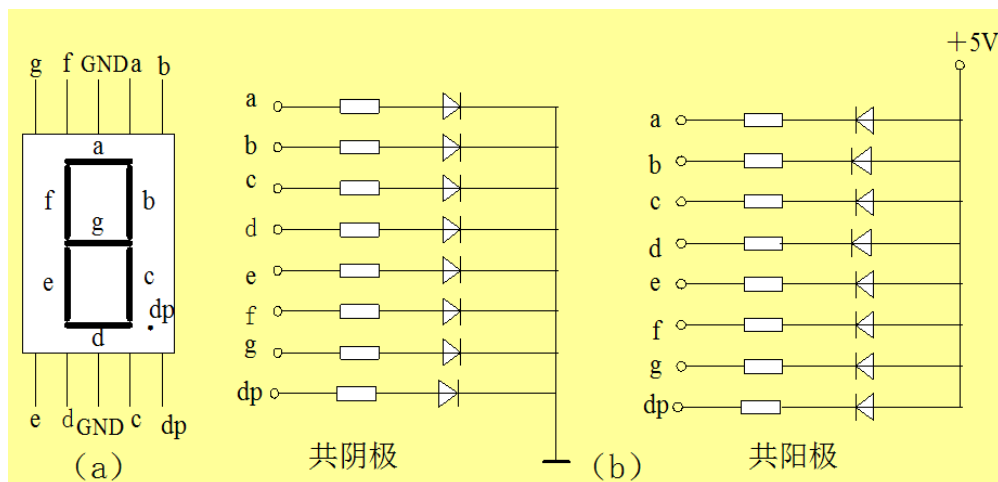


图 7.9 逻辑电路

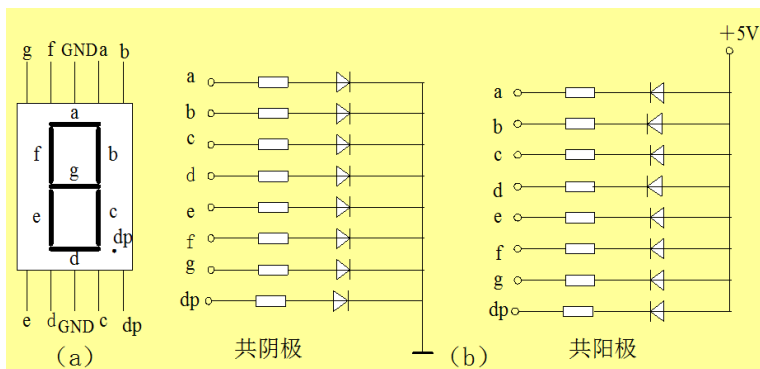
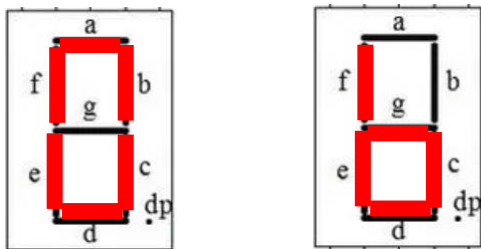
[illegible]

• (3) 七段显示译码器

- **七段数码管**（七段显示器）可以显示**0-9**数字以及部分英文字母及符号。
- 真实的七段数码管（也称八段数码管）有**10**个输入，除了**a~g**输入外（**7**个），还有**dp**（小数点）、**2**个**GND**（接地或接**+5V**）。
- 七段数码管分为共阴极和共阳极两种：
 - **共阴极**：**GND**接地，输入（**a~g、dp**）为高电平时，数码管亮。
 - **共阳极**：**GND**接**+5V**，输入（**a~g、dp**）为低电平时，数码管亮。



- **七段码**：与七段数码管显示字符对应的代码，分为共阴极七段码和共阳极七段码。
- 例如，字符“0”对应的共阴极七段码= $dp\ g\ f\ e\ d\ c\ b\ a=0\ 0\ 1\ 1\ 1\ 1\ 1=3FH$ ，共阳极七段码= $dp\ g\ f\ e\ d\ c\ b\ a=1\ 1\ 0\ 0\ 0\ 0\ 0=/3FH=C0H$ 。
- 例如，字符“b”对应的共阴极七段码= $dp\ g\ f\ e\ d\ c\ b\ a=0\ 1\ 1\ 1\ 1\ 1\ 0=7CH$ ，共阳极七段码= $dp\ g\ f\ e\ d\ c\ b\ a=1\ 0\ 0\ 0\ 0\ 0\ 1=/7CH=83H$ 。



显示字符	共阴极段选码	共阳极段选码	显示字符	共阴极段选码	共阳极段选码
0	3FH	C0H	C	39H	C6H
1	06H	F9H	d	5EH	A1H
2	5BH	A4H	E	79H	86H
3	4FH	B0H	F	71H	84H
4	66H	99H			
5	6DH	92H			
6	7DH	82H			
7	07H	F8H			
8	7FH	80H			
9	6FH	90H	“灭”	00H	FFH
A	77H	88H			:
b	7CH	83H			

- 七段显示译码器（**七段码译码器**）芯片**7448**：输入为4位二进制数（ $A_3 \sim A_0$ ），输出为七段码（a~g，不包括dp），另有3个控制信号，其中2个为输入信号（/LT、/RBI），1个为输入/输出双向信号（输入时为/B \bar{I} ，输出时为/RBO）。

- /LT：灯测试输入端（低电平有效）。
- /RBI：灭零输入端（低电平有效）。
- /B \bar{I} 或/RBO：熄灭输入端（低电平有效）或灭零输出端（低电平有效）。

- 图7.11：7448的引脚及逻辑符号。表7.4：7448的功能表。图7.12：七段显示译码器与七段显示器（七段数码管）的连接图及4位二进制输入对应的显示字符。

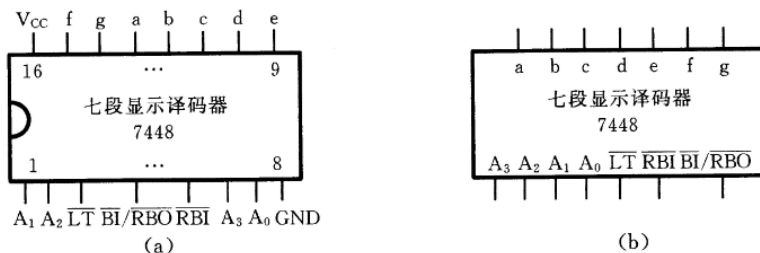


图 7.11 7448 的引脚排列图和逻辑符号

表 7.4 7448 的功能表

十进制数 或功能	输 入						B \bar{I} /RBO	输 出							说明
	L \bar{T}	R $\bar{B}I$	A $_3$	A $_2$	A $_1$	A $_0$		a	b	c	d	e	f	g	
0	1	1	0	0	0	0	1	1	1	1	1	1	1	0	译 码 显 示
1	1	d	0	0	0	1	1	0	1	1	0	0	0	0	
2	1	d	0	0	1	0	1	1	1	0	1	1	0	1	
3	1	d	0	0	1	1	1	1	1	1	1	0	0	1	
4	1	d	0	1	0	0	1	0	1	1	0	0	1	1	
5	1	d	0	1	0	1	1	1	0	1	1	0	1	1	
6	1	d	0	1	1	0	1	0	0	1	1	1	1	1	
7	1	d	0	1	1	1	1	1	1	1	0	0	0	0	
8	1	d	1	0	0	0	1	1	1	1	1	1	1	1	
9	1	d	1	0	0	1	1	1	1	1	0	0	1	1	
10	1	d	1	0	1	0	1	0	0	0	1	1	0	1	
11	1	d	1	0	1	1	1	0	0	1	1	0	0	1	
12	1	d	1	1	0	0	1	0	1	0	0	0	1	1	
13	1	d	1	1	0	1	1	1	0	0	1	0	1	1	
14	1	d	1	1	1	0	0	0	0	0	1	1	1	1	
15	1	d	1	1	1	1	1	0	0	0	0	0	0	0	
消隐	d	d	d	d	d	d	0	0	0	0	0	0	0	0	熄灭
脉冲消隐	1	0	0	0	0	0	0	0	0	0	0	0	0	0	灭零
灯测试	0	d	d	d	d	d	1	1	1	1	1	1	1	1	测试

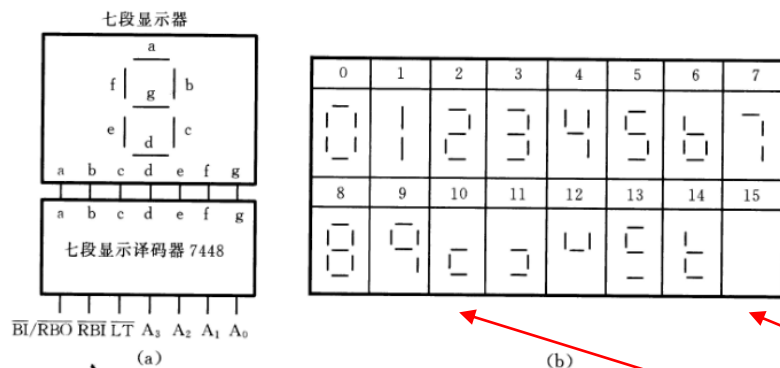
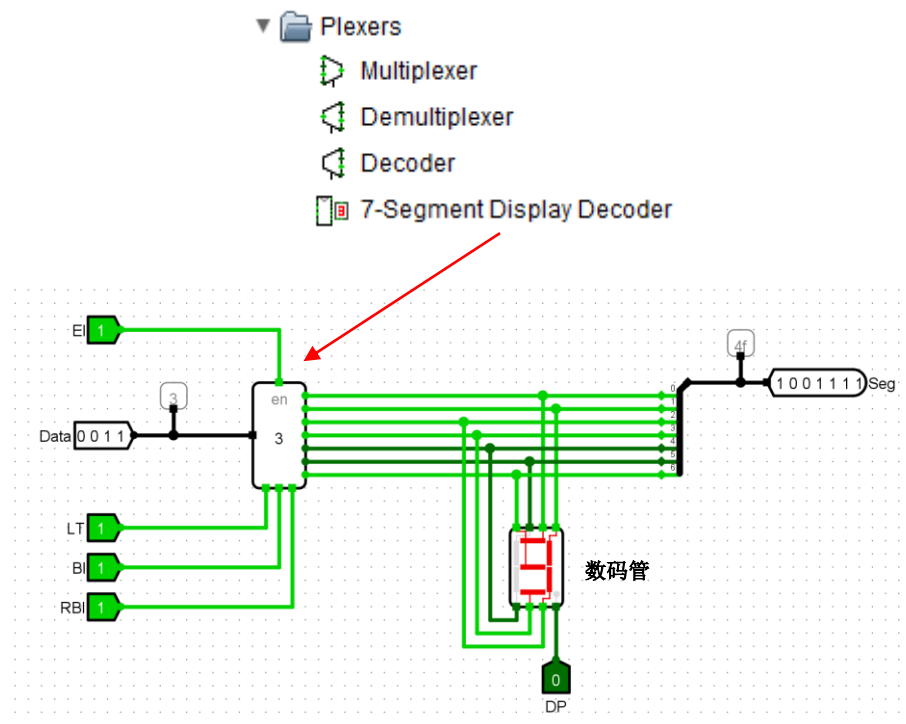
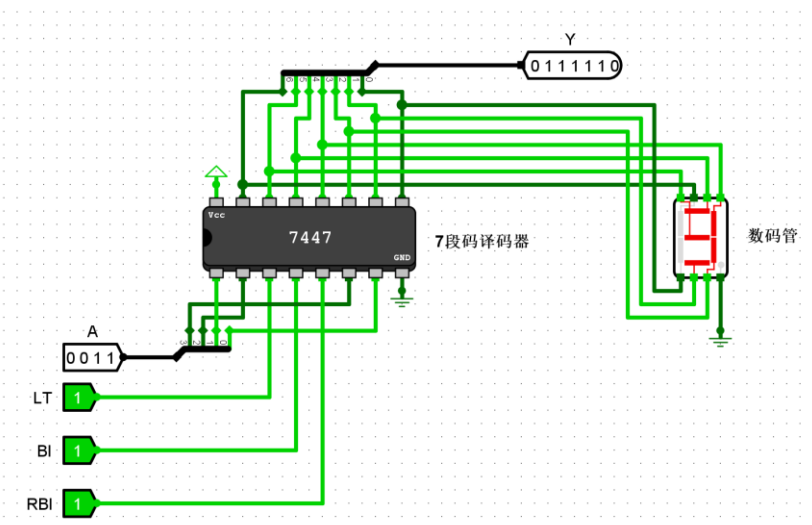


图 7.12 七段显示译码工作原理及笔画与字形关系

对应1010输入

对应1111输入

- Logisim中的七段码译码器（两种：一种是**7447芯片**，见左下图；另一种是**7-Segment Display Decoder**，见右下图）及数码管。



– 2、编码器

- **编码器（Encoder）**的功能恰好与译码器相反，其对输入信号按一定规律进行编码。常见的编码器有：**二-十进制编码器**、**优先编码器**。

- **（1）二-十进制编码器**

- 二-十进制编码器的输入有**10个**，输出为**4位BCD码**（对应**10个输入**）。
- 图7.13: **按键式8421码编码器**；输入有**10个**，输出除了BCD码外（A、B、C、D），还有**1个使用输出标志S**（当没有任何输入键按下时，**S=0**；当有输入键按下时，**S=1**）。
- 表7.5: **8421码编码器的真值表**。

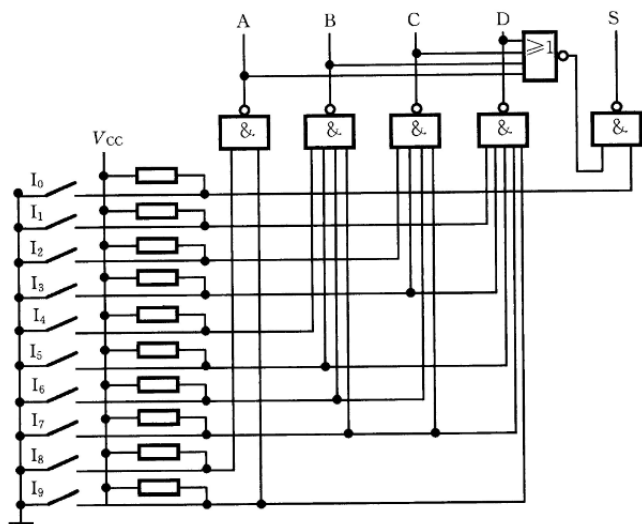


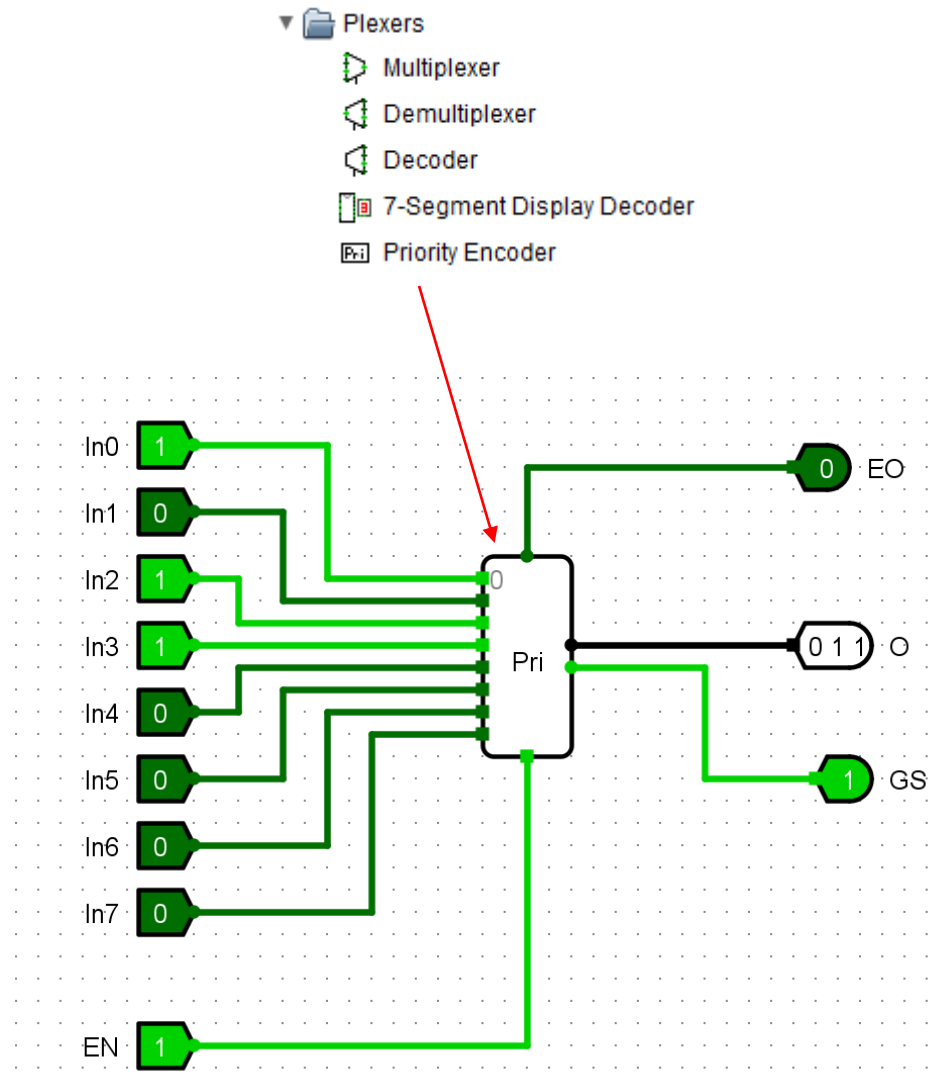
图 7.13 按键式 8421 码编码器

表 7.5 8421 码编码器真值表

输 入										输 出				
I ₉	I ₈	I ₇	I ₆	I ₅	I ₄	I ₃	I ₂	I ₁	I ₀	A	B	C	D	S
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	0	1	0	0	0	1	1
1	1	1	1	1	1	1	0	1	1	0	0	1	0	1
1	1	1	1	1	1	0	1	1	1	0	0	1	1	1
1	1	1	1	1	0	1	1	1	1	0	1	0	0	1
1	1	1	1	0	1	1	1	1	1	0	1	0	1	1
1	1	1	0	1	1	1	1	1	1	0	1	1	0	1
1	1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	0	0	1	1

[illegible]

– Logisim中的优先编码器（Priority Encoder）。



- **例7.7:** 用优先编码器74148设计一个能裁决16级不同中断请求的中断优先编码器。
- 解:
 - 具体见图7.15: $/I_{Z0} \sim /I_{Z15}$ 为16个中断请求信号, $/I_{Z5}$ 为选通输入端。 $/Q_{ZD} \sim /Q_{ZA}$ 为编码输出, O_{Z5} 为选通输出端, $/Q_{ZEX}$ 为工作状态标志。
 - 若 $/I_{Z5}=1$: 第II片74148输出全部为1, 导致第I片74148的 $/I_5=0$, 第I片74148输出也全部为1; 即 $/Q_{ZD} \sim /Q_{ZA}=1111$, $O_{ZEX}=1$, $O_{Z5}=1$ 。
 - 若 $/I_{Z5}=0$, $/I_{Z15} \sim /I_{Z0}$ 无中断请求信号 (全部为1): 第II片74148的 $O_5=0$ 、其余4个输出全部为1, 第I片74148的 $/I_5=0$, 则第I片74148的 $O_5=0$ 、其余4个输出全部为1。因此, $/Q_{ZD}=/Q_{EX}$ (第II片)=1, $/Q_{ZC}=1 \cdot 1=1$, $/Q_{ZB}=1 \cdot 1=1$, $/Q_{ZA}=1 \cdot 1=1$, $O_{Z5}=O_5$ (第I片)=0, $O_{ZEX}=1 \cdot 1=1$; 即 $/Q_{ZD} \sim /Q_{ZA}=1111$, $O_{ZEX}=1$, $O_{Z5}=0$ 。
 - 若 $/I_{Z5}=0$, $/I_{Z15} \sim /I_{Z8}$ 有中断请求信号 (有一个或多个0, 例如 $/I_{Z10}=0$): 第II片74148处于工作状态, 其 $O_5=1$, 使第I片74148不工作, 其5个输出全部为1。因此, $/Q_{ZD}=/O_{EX}$ (第II片)=0, $/Q_{ZC}=1 \cdot /Q_C$ (第II片)=1, $/Q_{ZB}=1 \cdot /Q_B$ (第II片)=0, $/Q_{ZA}=1 \cdot /Q_A$ (第II片)=1, $O_{Z5}=O_5$ (第I片)=1, $O_{ZEX}=1 \cdot /O_{EX}$ (第II片)=0; 即 $/Q_{ZD} \sim /Q_{ZA}=0101$, $O_{ZEX}=0$, $O_{Z5}=1$ 。
 - 若 $/I_{Z5}=0$, $/I_{Z7} \sim /I_{Z0}$ 有中断请求信号 (有一个或多个0, 例如 $/I_{Z3}=0$): 第II片74148处于工作状态, 其 $O_5=0$, 其余4个输出全部为1, 使第I片74148工作。因此, $/Q_{ZD}=/Q_{EX}$ (第II片)=1, $/Q_{ZC}=1 \cdot /Q_C$ (第I片)=1, $/Q_{ZB}=1 \cdot /Q_B$ (第I片)=0, $/Q_{ZA}=1 \cdot /Q_A$ (第I片)=0, $O_{Z5}=O_5$ (第I片)=1, $O_{ZEX}=1 \cdot /O_{EX}$ (第I片)=0。即 $/Q_{ZD} \sim /Q_{ZA}=1100$, $O_{ZEX}=0$, $O_{Z5}=1$ 。

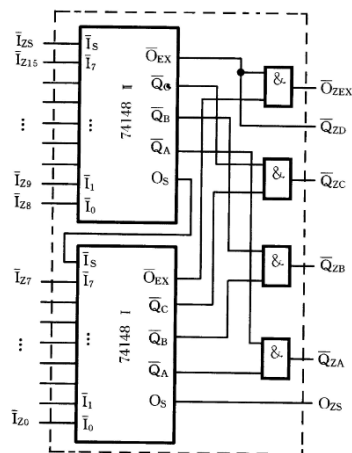


图 7.15 16 级中断优先编码器

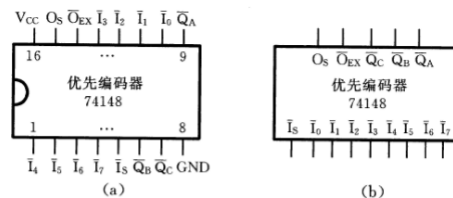
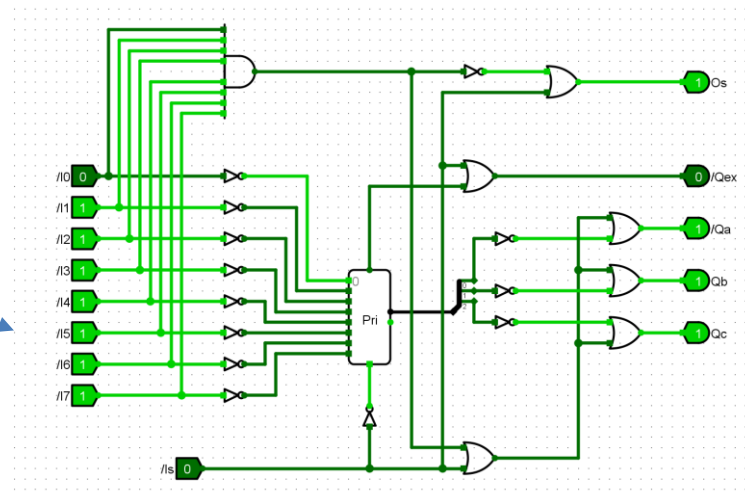
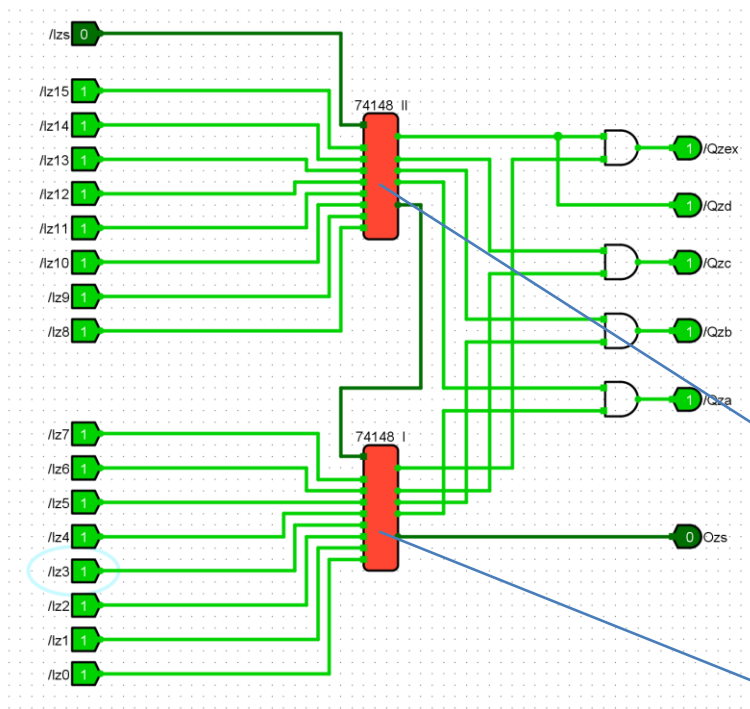


图 7.14 74148 优先编码器引脚排列和逻辑符号

表 7.6 74148 真值表

[illegible]

— 例7.7在Logisim中的实现。



74148

• 7.1.3 多路选择器和多路分配器

– 1、多路选择器

- **多路选择器**（Multiplexer，简称**MUX**）又称数据选择器或多路开关，其逻辑功能是从多路输入中选择一路送至输出端，输出对输入的选择受选择控制变量控制。

• （1）典型芯片

- 常见的有：**双4路MUX芯片74153**、**8路MUX芯片74152**（无使能控制端）、**8路MUX芯片74151**、**16路MUX芯片74150**。
- 图7.16：双4路MUX芯片74153的引脚和逻辑符号。表7.7：双4路MUX芯片74153的功能表。
- 4路MUX的输出函数表达式： $Y = /A_1 \cdot /A_0 \cdot D_0 + /A_1 \cdot A_0 \cdot D_1 + A_1 \cdot /A_0 \cdot D_2 + A_1 \cdot A_0 \cdot D_3 = \sum m_i \cdot D_i$ 。
- 8路MUX、16路MUX、..... 2^n 路MUX的输出函数表达式类似。

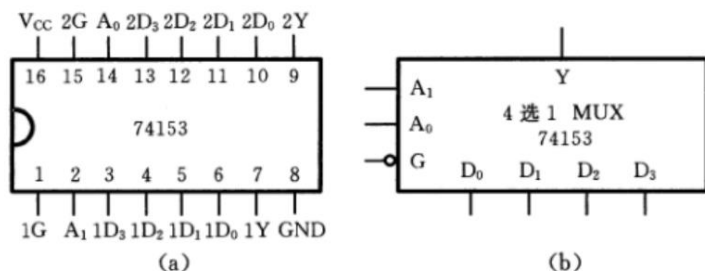


图 7.16 74153 引脚排列图和逻辑符号

表 7.7 4 路 MUX 74153 的功能表

使能输入	选择输入		数据输入				输 出
G	A ₁	A ₀	D ₀	D ₁	D ₂	D ₃	Y
1	d	d	d	d	d	d	0
0	0	0	D ₀	d	d	d	D ₀
0	0	1	d	D ₁	d	d	D ₁
0	1	0	d	d	D ₂	d	D ₂
0	1	1	d	d	d	D ₃	D ₃

• (2) 应用举例

– 可以用具有 n 个选择控制变量的MUX（4路MUX的 $n=2$ ，8路MUX的 $n=3$ ，16路MUX的 $n=4$）实现 m 个变量的函数，有3种情况：

① $m=n$ ：将函数的 m 个变量依次连接到MUX的 n 个选择变量端，若函数表达式中包含最小项 m_i ，则MUX的相应 D_i 接1，否则 D_i 接0。

② $m=n+1$ ：从函数的 m 个（ $n+1$ 个）变量中任选 n 个变量作为MUX的选择控制变量，剩余的1个变量用 x 表示，将函数变换为 $Y = \sum m_i \cdot D_i$ ，则 D_i 的取值只可能是0、1、 $/x$ 、 x 四者之一。

③ $m \geq n+2$ ：与第②种情况类似，但此时 D_i 为0、1以及剩余变量的逻辑函数。

– **例7.8：**用MUX实现以下逻辑函数的功能： $Y(A,B,C) = \sum m(2,3,5,6)$ 。

– 解：

• 方案一：采用8路MUX（74152芯片）实现。

– 属于上面的第①种情况， $m=3$ ， $n=3$ ， $m=n$ 。

– 根据第①情况的方法，8路MUX的数据输入端为： $D_0=0$ 、 $D_1=0$ 、 $D_2=1$ 、 $D_3=1$ 、 $D_4=0$ 、 $D_5=1$ 、 $D_6=1$ 、 $D_7=0$ 。具体电路见图7.17(a)。

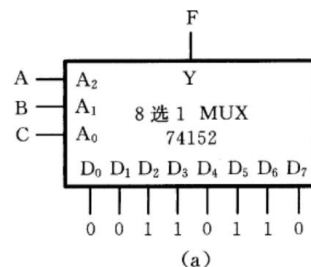


图7.17 例7.8实现的第一种方案

- 方案二：采用4路MUX（74153芯片）实现。
 - 属于上面的第②种情况， $m=3$ ， $n=2$ ， $m=n+1$ 。
 - 根据第②情况的方法，从A、B、C中选择A、B作为4路MUX的选择控制变量（ A_1 、 A_0 ），将逻辑函数转换为：

$$Y(A,B,C) = \sum m(2,3,5,6) = \overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} = \overline{A} \cdot \overline{B} \cdot 0 + \overline{A} \cdot B \cdot (\overline{C} + C) + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} = \overline{A} \cdot \overline{B} \cdot 0 + \overline{A} \cdot B \cdot 1 + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C}。$$
 - 因此4路MUX的数据输入端为： $D_0=0$ 、 $D_1=1$ 、 $D_2=C$ 、 $D_3=\overline{C}$ 。具体电路见图7.17(b)。

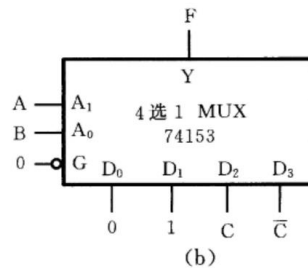


图7.17 例7.8实现的第二种方案

— **例7.9**：用4路MUX实现4变量逻辑函数的功能： $Y(A,B,C,D)=\sum m(0,2,3,7,8,9,10,13)$ 。

— 解：

- 该例属于上面的第③种情况， $m=4$ ， $n=2$ ， $m \geq n+2$ 。

- 方案一：选用A、B作为4路MUX（74153芯片）的选择控制变量（ A_1 、 A_0 ）。

- 根据第③情况的方法，将逻辑函数转换为：

$$Y(A,B,C,D)=\sum m(0,2,3,7,8,9,10,13)=\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot\overline{B}\cdot C\cdot\overline{D}+\overline{A}\cdot\overline{B}\cdot C\cdot D+\overline{A}\cdot B\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot B\cdot\overline{C}\cdot D+\overline{A}\cdot B\cdot C\cdot\overline{D}+\overline{A}\cdot B\cdot C\cdot D+A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+A\cdot\overline{B}\cdot\overline{C}\cdot D+A\cdot\overline{B}\cdot C\cdot\overline{D}+A\cdot\overline{B}\cdot C\cdot D+A\cdot B\cdot\overline{C}\cdot\overline{D}+A\cdot B\cdot\overline{C}\cdot D+A\cdot B\cdot C\cdot\overline{D}+A\cdot B\cdot C\cdot D$$
$$= \overline{A}\cdot\overline{B}\cdot(\overline{C}+\overline{D}) + \overline{A}\cdot\overline{B}\cdot(C\cdot\overline{D} + C\cdot D) + \overline{A}\cdot B\cdot(\overline{C}+\overline{D}) + \overline{A}\cdot B\cdot(C\cdot\overline{D} + C\cdot D)$$

- 4路MUX的数据输入端为： $D_0 = \overline{C}+\overline{D}$ 、 $D_1 = C\cdot\overline{D}$ 、 $D_2 = \overline{C}+D$ 、 $D_3 = C\cdot D$ 。具体电路见图7.18(a)。

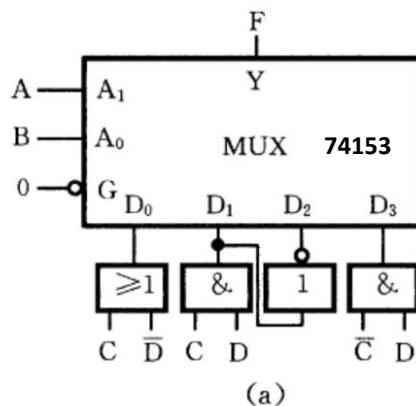


图7.18 例7.9实现的第一种方案

- 方案二：选用C、D作为4路MUX的选择控制变量（ A_1 、 A_0 ）。
 - 根据第③情况的方法，将逻辑函数转换为：

$$Y(A,B,C,D) = \sum m(0,2,3,7,8,9,10,13) = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot \overline{B} \cdot C \cdot D + \overline{A} \cdot B \cdot \overline{C} \cdot D + \overline{A} \cdot B \cdot C \cdot D + A \cdot \overline{B} \cdot \overline{C} \cdot D + A \cdot \overline{B} \cdot C \cdot D + A \cdot B \cdot \overline{C} \cdot D + A \cdot B \cdot C \cdot D + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot C \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot C \cdot \overline{D}$$
 - 4路MUX的数据输入端为： $D_0 = \overline{B}$ 、 $D_1 = A$ 、 $D_2 = \overline{B}$ 、 $D_3 = \overline{A}$ 。具体电路见图7.18(b)。
 - 可见方案二的电路比方案一的电路简单。

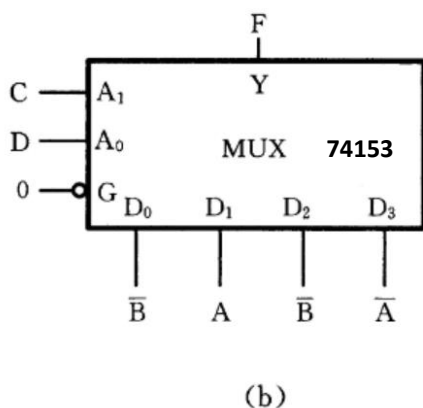


图7.18 例7.9实现的第二种方案

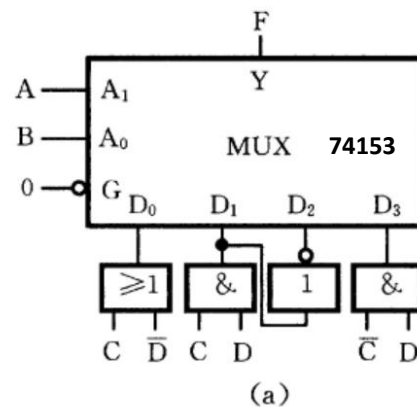


图7.18 例7.9实现的第一种方案

– **例7.10:** 用一片双4路MUX芯片74153实现4变量多输出函数:

$$F_1(A,B,C,D) = \sum m(0,1,5,7,10,13,15), \quad F_2(A,B,C,D) = \sum m(8,10,12,13,15)。$$

– 解:

- 该例属于上面的第③种情况, $m=4$, $n=2$, $m \geq n+2$ 。

- 选用A、B作为4路MUX的选择控制变量 (A_1 、 A_0)。

- 根据第③情况的方法, 将逻辑函数转换为:

- $F_1(A,B,C,D) = \sum m(0,1,5,7,10,13,15) = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot C \cdot D + \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} \cdot D + \overline{A} \cdot B \cdot C \cdot \overline{D} + \overline{A} \cdot B \cdot C \cdot D = \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot D + \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot D。$

- $F_2(A,B,C,D) = \sum m(8,10,12,13,15) = A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot D + A \cdot \overline{B} \cdot C \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot D + A \cdot B \cdot \overline{C} \cdot \overline{D} = \overline{A} \cdot \overline{B} \cdot 0 + \overline{A} \cdot \overline{B} \cdot 0 + \overline{A} \cdot \overline{B} \cdot D + \overline{A} \cdot B \cdot (\overline{C} + D)。$

- 双4路MUX的数据输入端为: $1D_0 = \overline{C}$ 、 $1D_1 = D$ 、 $1D_2 = \overline{C} \cdot \overline{D}$ 、 $1D_3 = D$; $2D_0 = 0$ 、 $2D_1 = 0$ 、 $2D_2 = \overline{D}$ 、 $2D_3 = \overline{C} + D$ 。具体电路见图7.19。

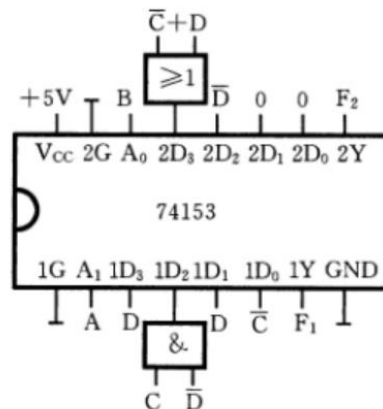


图 7.19 逻辑电路

– 2、多路分配器

- **多路分配器**（Demultiplexer，简称**DEMUX**）又称数据分配器，其结构与多路选择器（MUX）正好相反，它是一种单输入、多输出的逻辑部件，输入数据具体从哪一路输出由选择控制变量决定。
- 图7.20：4路DEMUX的逻辑符号，D为数据输入端， A_1 、 A_0 为选择控制输入端， $f_0 \sim f_3$ 为数据输出端。
- 表7.8：4路DEMUX的功能表。
- 若将4路DEMUX的数据输入端D接1，则将实现**2-4线译码器**的功能。

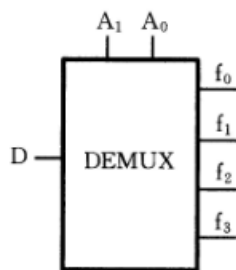


图 7.20 4 路 DEMUX
的逻辑符号

表 7.8 4 路 DEMUX 功能表

A_1	A_0	f_0	f_1	f_2	f_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

- 可以将DEMUX（多路分配器）与MUX（多路选择器）联用，实现多通道数据分时传送。
- 如图7.21所示，当 $ABC=000$ 时， $f_0=D_0$ ；当 $ABC=111$ 时， $f_7=D_7$ 。

• **例7.11：**用8路MUX和3-8线译码器构造一个3位二进制数等值比较器。

• 解：

- 比较器的电路如图7.22所示。两个3位二进制数（ ABC ， XYZ ）分别接译码器的 $A_2A_1A_0$ 和MUX的 $A_2A_1A_0$ 。
- 当 $ABC=XYZ$ 时，输出 $F=0$ ；当 $ABC \neq XYZ$ 时，输出 $F=1$ 。
- 例如，当 $ABC=XYZ=010$ 时，译码器的 $/Y_2=0$ ，MUX的 $D_2=/Y_2=0$ ，故 $F=0$ 。
- 当 $ABC=010$ ， $XYZ \neq 010$ 时，因为除 $/Y_2$ 之外译码器的其它输出都是1，因此MUX的输入除 D_2 之外，其它都是1，故 $F=1$ 。

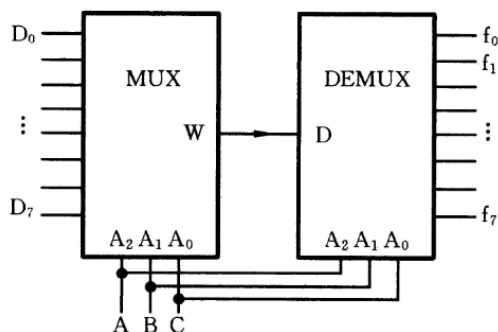


图 7.21 8 路数据传输示意图

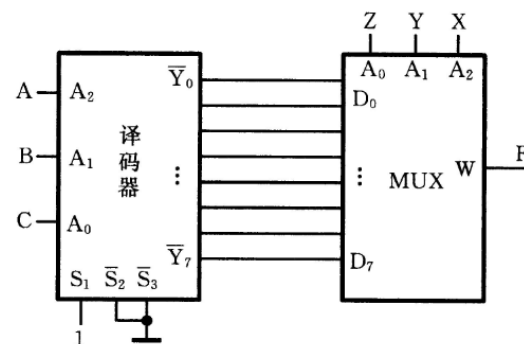


图 7.22 比较器的逻辑电路

7.2 常用中规模时序逻辑电路

7.2.1 集成计数器

7.2.2 集成寄存器

- 7.2.1 集成计数器

- **7.2.2 集成寄存器**

7.3 常用中规模信号产生与变换电路

7.3.1 集成定时器555及其应用

7.3.2 集成D/A转换器

7.3.3 集成A/D转换器

- 7.3.1 集成定时器555及其应用

- **7.3.2 集成D/A转换器**

- **7.3.3 集成A/D转换器**

本章小结

习题 (P223) (第一部分)

- 7.1
- 7.2
- 7.3
- 7.4
- 7.6
- 7.7
- 7.8

习题 (P223)

7.1 用 4 位二进制并行加法器设计一个实现 8421 码对 9 求补的逻辑电路。

所谓对9求补，就是9减这个数

例如，5对9求补 $=9-5=4$ ；10对9求补 $=9-10=-1$

习题 (P223)

7.2 用两个 4 位二进制并行加法器实现 2 位十进制数 8421 码到二进制码的转换。

习题 (P223)

7.3 用 4 位二进制并行加法器设计一个用 8421 码表示的 1 位十进制加法器。

习题 (P223)

7.4 用一片 3-8 线译码器和必要的逻辑门实现下列逻辑函数表达式：

$$F_1 = \bar{A} \bar{C} + ABC$$

$$F_2 = \bar{A} + B$$

$$F_3 = AB + \bar{A} \bar{B}$$

习题 (P223)

7.5 用一片 4-16 线译码器和适当的逻辑门设计一个 1 位十进制数 2421 码的奇偶位产生电路(假定采用奇检验)。

习题 (P223)

7.6 当优先编码器 74LS148 的 \bar{I}_s 接 0, 输入 $\bar{I}_7 \bar{I}_6 \bar{I}_5 \bar{I}_4 \bar{I}_3 \bar{I}_2 \bar{I}_1 \bar{I}_0 = 11010001$ 时, 输出状态为何值?

习题 (P223)

7.7 试用 4 路数据选择器实现余 3 码到 8421 码的转换。

习题 (P223)

7.8 当 4 路选择器的选择控制变量 A_1 、 A_0 接变量 A 、 B , 数据输入端 D_0 、 D_1 、 D_2 、 D_3 依次接 \bar{C} 、 0 、 0 、 C 时, 电路实现何功能?

习题 (P223)

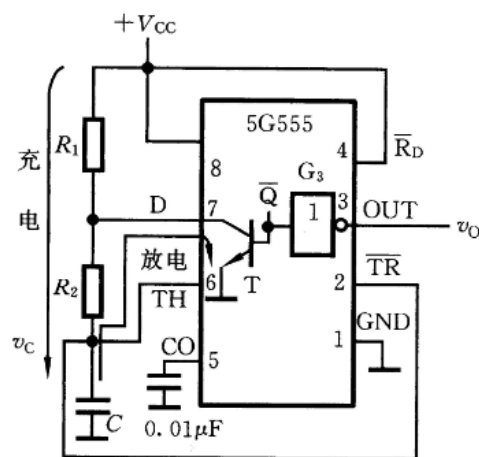
7.9 用 4 位同步可逆计数器 74193 和必要的逻辑门实现模 12 加法计数器。

习题 (P223)

7.10 用 74194 双向移位寄存器和必要的逻辑门设计一个 00011101 序列信号发生器。

习题 (P223)

7.11 在图 7.38(a)所示电路中,若取 $R_1 = 2R_2$, 请问输出矩形波的占空比为多少?



(a)

图 7.38 用 5G555 构成的多谐振荡器电路及其工作波形图

习题 (P223)

7.12 分析图 7.53 所示由定时器 5G555 构成的多谐振荡器。

(1) 计算其振荡周期；

(2) 若要产生占空比为 50% 的方波, R_1 和 R_2 的取值关系如何?

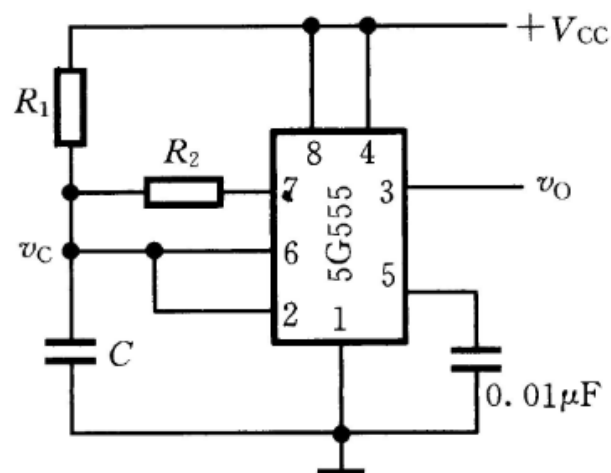


图 7.53 多谐振荡器

习题 (P223)

7.13 将 5G555 定时器按图 7.54(a) 所示连接, 输入波形如图 7.54(b) 所示。请画出定时器输出波形, 并说明该电路相当于什么器件。

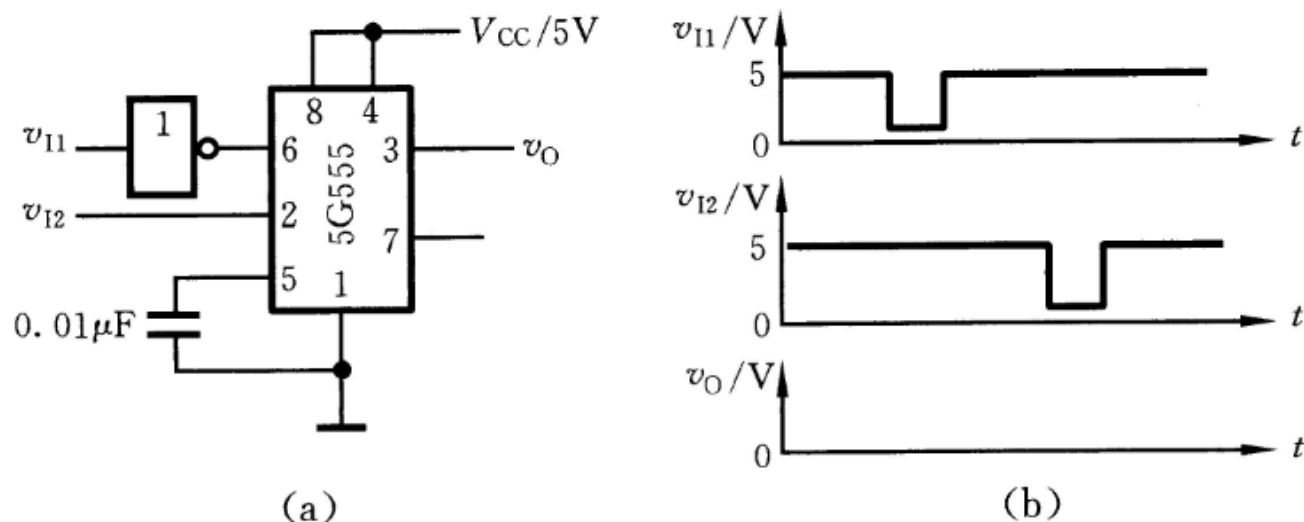


图 7.54 5G555 连线图和有关波形图

习题 (P223)

7.14 D/A 转换器有哪些主要参数？通常用什么参数来衡量转换精度？

习题 (P223)

7.15 DAC1210 是 12 位 D/A 转换芯片, 请问其分辨率为多少? (用百分数表示)

习题 (P223)

7.16 DAC0832 由哪几部分组成？可以构成哪几种工作方式？每种方式如何控制？

习题 (P223)

7.17 常见集成 A/D 转换器按转换方法的不同可分成哪几种类型？各有何特点？

习题 (P223)

7.18 ADC0809 如何实现对 8 路模拟量输入的选择？当它与微机连接时是否要外加三态缓冲器？

作业样例

- 1.1 冯·诺依曼结构计算机的基本思想是什么？按此思想设计的计算机硬件系统应由哪些部件组成？它们各有何作用？

• 答：

- 数学家冯·诺依曼提出了计算机制造的三个基本原则，即采用二进制逻辑、程序存储执行（存储程序和程序控制）以及计算机由五个部分组成（运算器、控制器、存储器、输入设备、输出设备），这套理论被称为冯·诺依曼体系结构。
- 计算机硬件系统应由运算器、控制器、存储器、输入设备、输出设备等组成。
- 运算器是一种用于信息加工处理的部件，它对数据进行算术运算和逻辑运算。运算器通常由算术逻辑单元（ALU, Arithmetic and Logic Unit）和一系列寄存器组成。通常将运算器一次运算能处理的二进制位数称为机器字长。现代计算机具有多个寄存器，称为寄存器组。
- 控制器是整个计算机的指挥中心，它可使计算机各部件协调工作。计算机中有两股信息在流动，一股是控制流信息，另一股是数据流信息。控制流信息的发源地是控制器，控制器产生控制流信息的依据来自3个方面：指令寄存器、状态寄存器和时序电路。
- 存储器的主要功能是存放程序和数据，目前计算机的主存储器都是半导体存储器。
- 输入设备就是将信息输入计算机的外部设备，它将人们熟悉的信息形式转换成计算机能接收并识别的信息形式。
- 输出设备就是将计算机运算结果转换成人们和其他设备能接收和识别的信息形式的设备，如字符、文字、图形、图像、声音等。

关于作业提交

- **1周内**必须提交（上传到学院的FTP服务器上），否则认为是迟交作业；如果期末仍然没有提交，则认为是未提交作业。
 - 作业完成情况成绩=第1次作业提交情况*第1次作业评分+第2次作业提交情况*第2次作业评分+.....+第N次作业提交情况*第N次作业评分。
 - 作业评分：A（好）、B（中）、C（差）三挡。
 - 作业提交情况：按时提交（1.0）、迟交（0.5）、未提交（0.0）。
- 请采用电子版的格式（**PPT文档**）上传到FTP服务器上，文件名取“学号+姓名+第X次作业.pptx”。
 - 例如：11920222202406+刘济华+第7次作业.pptx
- 下次上课时（2023年12月25日）会**随机抽取2位同学**到讲台上汇报作业。
- 第7次作业提交的截止日期为：**2023年12月24日晚上24点**。

Thanks