

【例 15】 设计十进制数字键盘的编码逻辑。

解 图 2.28 画出了十进制数字键盘的编码逻辑。键盘上有 10 个数字按键，每一个十进制数字代表一个二进制按键开关。二进制按键开关通过一个电阻连接到直流电源+V 上。当某一个二进制按键开关按下时(例如十进制数“9”) I_9 线上产生一个低电平，而其他的输入线均为高电平。 $I_1 \sim I_9$ 输入信号同时送至优先编码器 74HC147 进行编码，最后输出 BCD 码 $\bar{D}_3 \bar{D}_2 \bar{D}_1 \bar{D}_0 = 0110$ (反码值)，其原码为 1001，即完成十进制数“9”的编码。

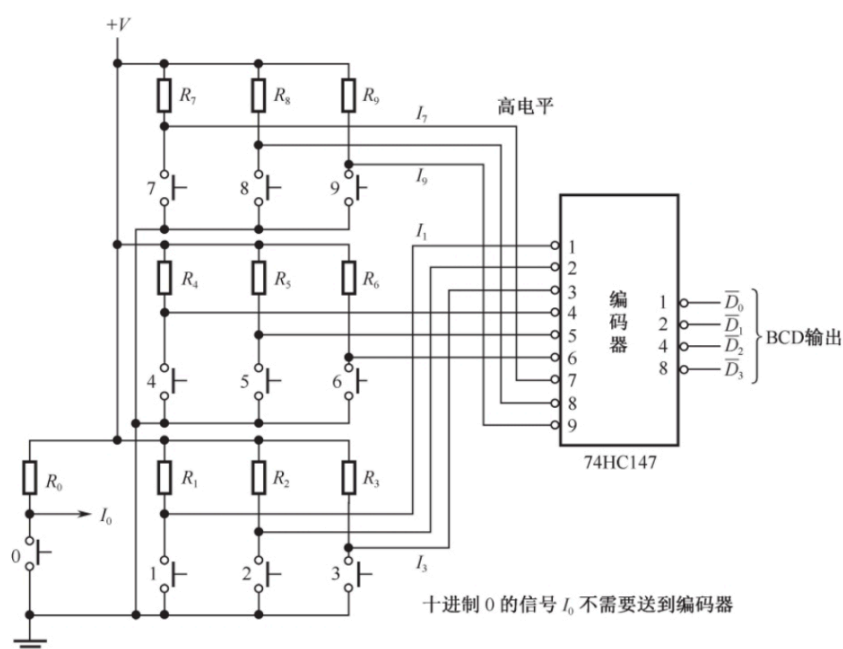


图 2.28 十进制数字键盘的编码逻辑

1. 真值表

I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	\bar{D}_3	\bar{D}_2	\bar{D}_1	\bar{D}_0	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	
x	x	x	x	x	x	x	x	x	0	0	1	1	0	9
x	x	x	x	x	x	x	x	0	1	0	1	1	1	8
x	x	x	x	x	x	x	0	1	1	1	0	0	0	7
x	x	x	x	x	x	0	1	1	1	1	0	0	1	6
x	x	x	x	x	0	1	1	1	1	1	0	1	0	5
x	x	x	x	0	1	1	1	1	1	1	0	1	1	4
x	x	x	0	1	1	1	1	1	1	1	1	0	0	3
x	x	0	1	1	1	1	1	1	1	1	1	0	1	2
x	0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0

2. 表达式

$$\textcircled{1} \quad \bar{D}_3 = \bar{I}_9 + I_9 \bar{I}_8 = \bar{I}_9 + \bar{I}_8$$

$$\bar{D}_3 = I_9 I_8$$

$$\textcircled{2} \quad \overline{D_2} = \overline{191817} + \overline{19181716} + \overline{1918171615} + \overline{191817161514} \\ = 1918(\overline{17} + \overline{16} + \overline{15} + \overline{14})$$

$$\overline{D_2} = \overline{19} + \overline{18} + 14151617$$

$$\textcircled{3} \quad \overline{D_1} = \overline{191817} + \overline{19181716} + \overline{19181716151413} + \overline{1918171615141312} \\ = 1918(\overline{17} + \overline{16} + \overline{151413} + \overline{151412})$$

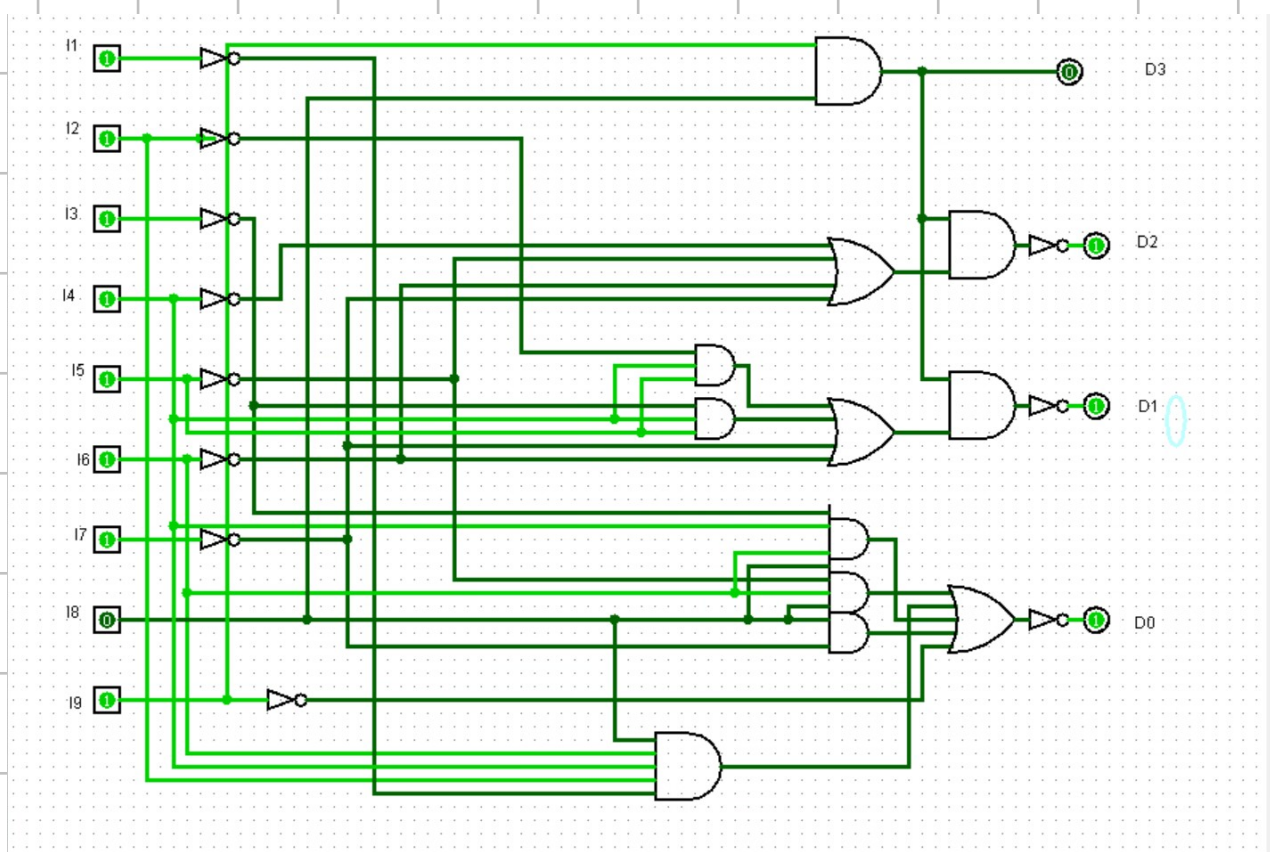
$$\overline{D_1} = \overline{19} + \overline{18} + 1716(\overline{15} + \overline{14} + \overline{13})(\overline{15} + \overline{14} + \overline{12})$$

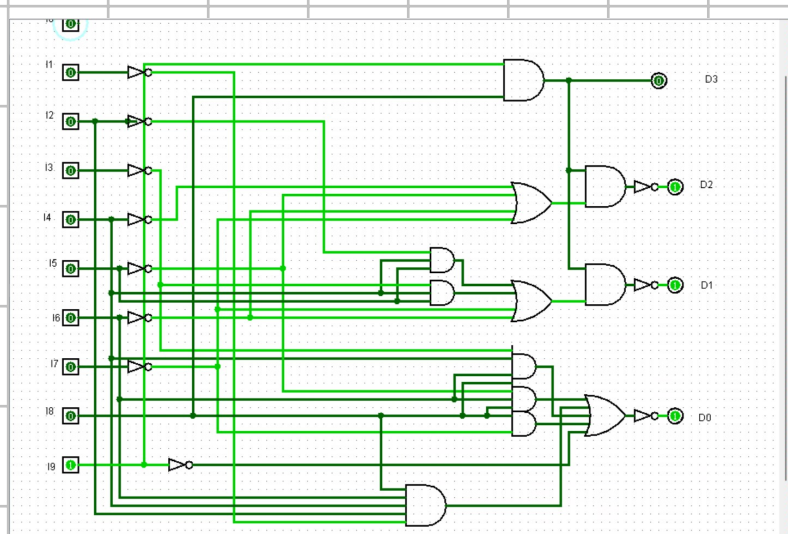
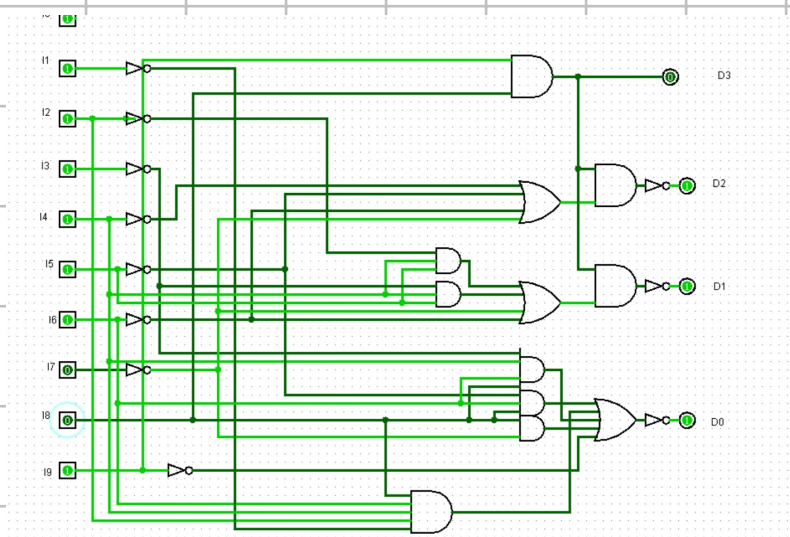
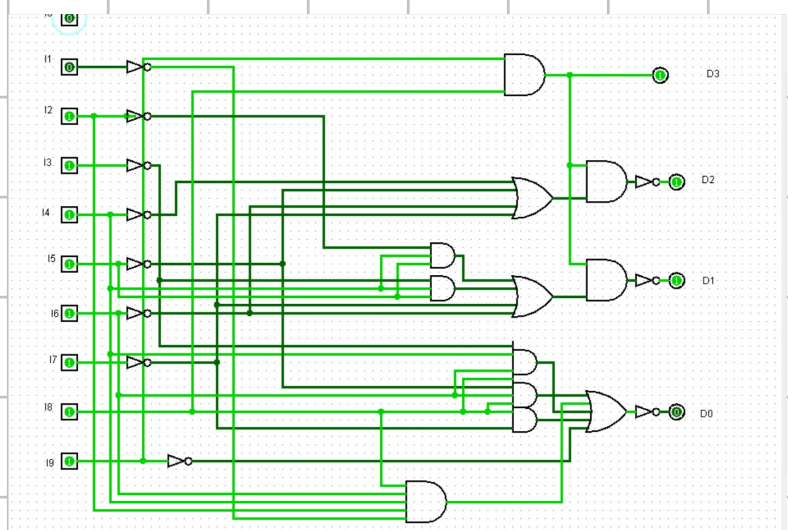
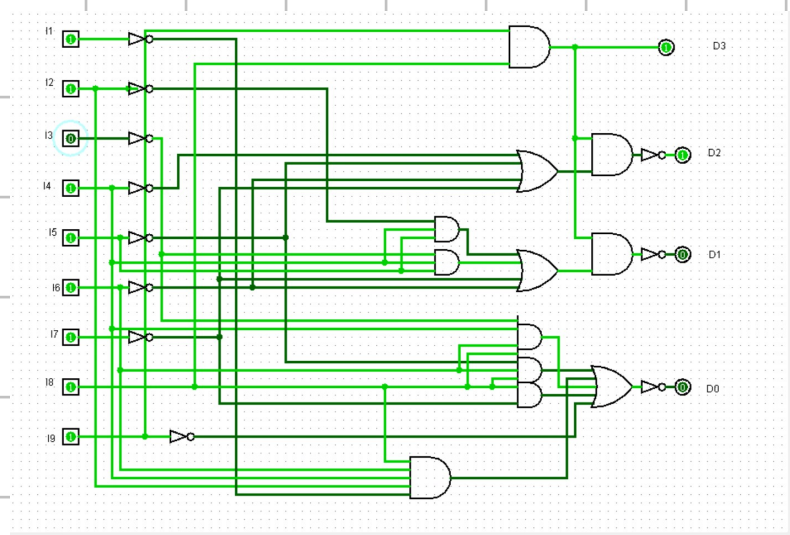
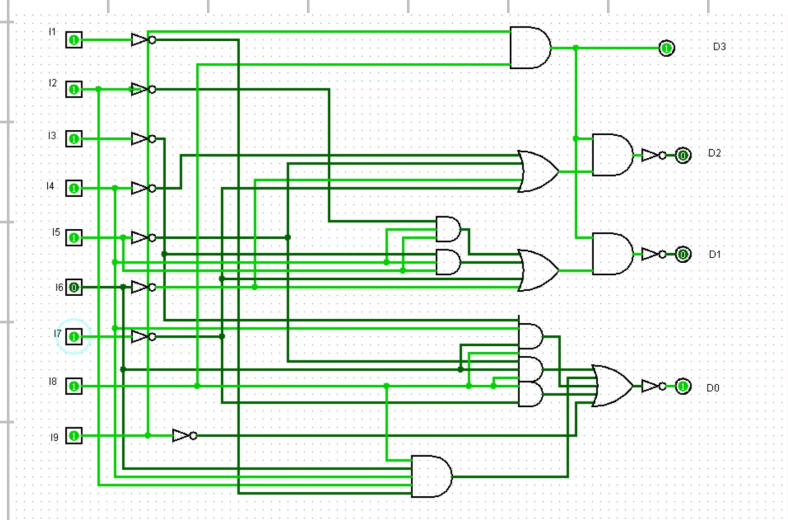
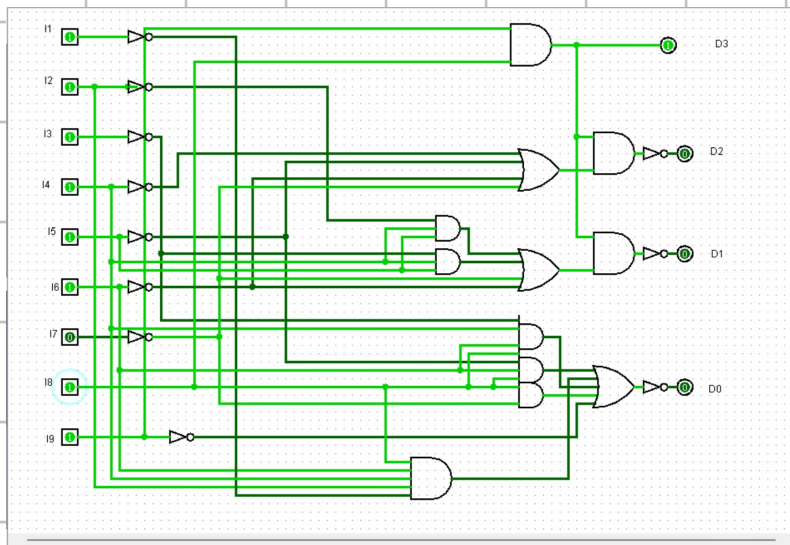
$$\textcircled{4} \quad \overline{D_0} = \overline{19} + \overline{191817} + \overline{1918171615} + \overline{19181716151413} \\ + \overline{191817161514131211}$$

$$= \overline{19} + \overline{1817} + \overline{181615} + \overline{18161413} + \overline{1816141211}$$

$$\overline{D_0} = 19(\overline{18} + \overline{17})(\overline{18} + \overline{16} + \overline{15})(\overline{18} + \overline{16} + \overline{14} + \overline{13}) \\ (\overline{18} + \overline{16} + \overline{14} + \overline{12} + \overline{11})$$

⑤ logicsim





普通编码器对输入线是有限制的，即在任意一时刻所有输入线中只允许一个输入线上有信号，否则编码器将发生混乱。为了解决这一问题，可采用**优先编码器**，它允许多个输入信号同时有效。设计时预先对所有输入按优先顺序进行排队，当多个输入同时有效时，只对其中优先级别最高的输入信号编码，而对级别较低的输入信号不予理睬。

表 2.10 列出了优先编码器 74LS148 功能表，它是 8 线-3 线编码器。其中 EI,EO 两个信号端用来扩展用。

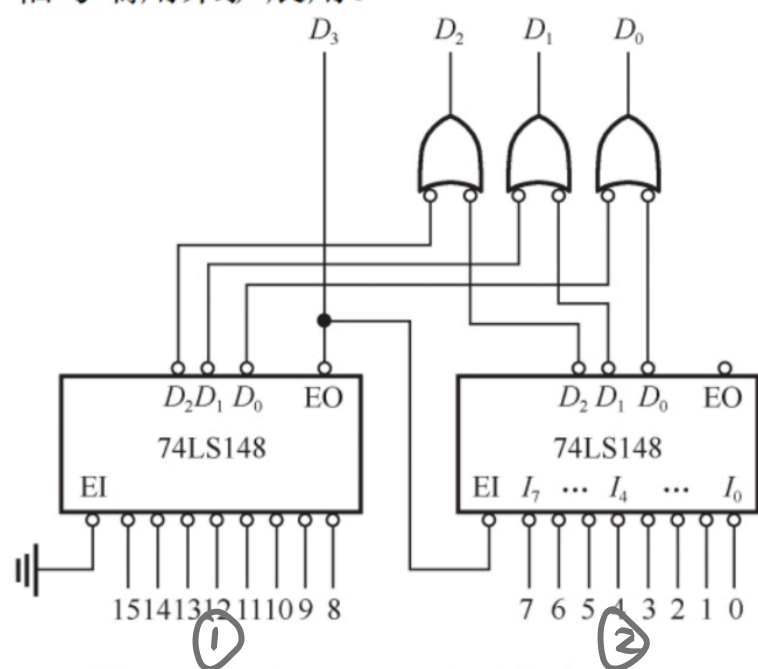


图 2.27 2 片 74LS148 组成优先编码器

从功能表看出，输入输出的有效信号都是低电平。在输入中，角标越大，优先级越高， \bar{I}_7 优先级最高。当 $\bar{I}_7=0$ 时，不管其他输入是什么，都是对 \bar{I}_7 编码， $\bar{D}_2\bar{D}_1\bar{D}_0=000$ (\bar{I}_7 的反码)，即 $I_7=111$ 。当输入 $\bar{I}_7=1$ ， $\bar{I}_6=0$ 时，不管其他输入是什么，都是对 \bar{I}_6 编码， $\bar{D}_2\bar{D}_1\bar{D}_0=001$ ，即 $I_6=110$ 。其余类推。

图 2.27 画出了 74LS148 的逻辑符号及其扩展连接图，其中 EI 为允许输入，EO 为允许输出。两片 74LS148 扩展成 16 线-4 线优先编码器(原码输出)。

EI 为允许输入，EO 为允许输出，注意到左边 EO 与右边 EI 相连。

即左边 EI 输入为 0，左 15-8 的 Data 输入有效， $D_2D_1D_0$ 左 $D_2D_1D_0$ 输入为 1，右 7-0 的 Data 输入有效， $D_2D_1D_0$ = 右 $D_2D_1D_0$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	EI	D_3	D_2	D_1	D_0
x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	1	0	0	0	0
x	x	x	x	x	x	0	1	x	x	x	x	x	x	x	x	1	0	0	0	1
x	x	x	x	x	0	1	1	x	x	x	x	x	x	x	x	1	0	0	1	0
x	x	x	x	0	1	1	1	x	x	x	x	x	x	x	x	1	0	0	1	1
x	x	x	0	1	1	1	1	x	x	x	x	x	x	x	x	1	0	1	0	0
x	x	0	1	1	1	1	1	x	x	x	x	x	x	x	x	1	0	1	0	1
x	0	1	1	1	1	1	1	x	x	x	x	x	x	x	x	1	0	1	1	0
0	1	1	1	1	1	1	1	x	x	x	x	x	x	x	x	1	0	1	1	1
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	1	0	0	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	0	1	0	0	1

x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	1	0	1	0	1	0
x	x	x	x	x	x	x	x	x	x	x	x	x	0	1	1	1	0	1	0	1
x	x	x	x	x	x	x	x	x	x	x	x	0	1	1	1	1	0	1	1	0
x	x	x	x	x	x	x	x	x	x	x	0	1	1	1	1	1	0	1	1	0
x	x	x	x	x	x	x	x	x	x	0	1	1	1	1	1	1	0	1	1	0
x	x	x	x	x	x	x	x	0	1	1	1	1	1	1	1	1	0	1	1	1