

组合逻辑作业

刘哲晗 24371277

一、

1. 开，低电平；关，高电平。与非门注意复习

2. ECL、TTL、CMOS；CMOS、TTL、ECL；

CMOS、TTL、ECL

3. 1、0

4. $AB + \overline{CD}; (\overline{A} + \overline{B}) \cdot (C + D)$

5. $A\overline{B} + AC + \overline{C}D$

6. 储存电路、反馈电路、输入信号

7. 编码

8. 译码，n， 2^n

9. 1111111011

10. 接入滤波电容、引入选通脉冲、增加冗余项

11. 布尔表达式、真值表、组合电路图、卡诺图

二、

1. ④

2. ②

3. ③

4. ③

5. ④

6. ①

7. ②

8. ③

9. ③

三、

1.

$$ABCD + ABC\overline{D} + \overline{A}BCD + \overline{A}BC\overline{D}$$

2.

$$\overline{B} + \overline{C}$$

3.

A	B	C	OUT
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
1	1	0	1
1	0	1	1
0	1	1	1
1	1	1	1

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

由卡诺图化简得到

$$F = AB + BC + AC$$

```
module votes(  
    input a,  
    input b,  
    input c,  
    output out);  
    assign out = (A & B) | (B & C) | (A & C);  
end module
```

$$\begin{aligned}
\text{左边} &= (BC + D + \overline{D}(\overline{B} + \overline{C}))(AD + B) \\
&= (BC + D + \overline{D}(\overline{B}AD + \overline{C}AD + \overline{B}B + \overline{C}B)) \quad (\text{分配律}) \\
&= (BC + D + \overline{D}(\overline{B}AD + \overline{C}AD + 0 + \overline{C}B)) \quad (\text{互补律}(\overline{B}B = 0)) \\
&= (BC + D + \overline{D} \cdot 0 + \overline{D}\overline{C}B) \quad (\text{互补律}(\overline{D}D = 0)) \\
&= (BC + D + \overline{B}\overline{C}\overline{D}) \\
4. \quad &= (B(C + \overline{C}\overline{D}) + D) \quad (\text{提取公因子}(B)) \\
&= (B(C + \overline{D}) + D) \quad (\text{吸收律变形}(A + \overline{A}B = A + B)) \\
&= (BC + B\overline{D} + D) \\
&= (BC + (B + D)) \quad (\text{吸收律变形}(\overline{A}AB + A = A + B)) \\
&= (B(C + 1) + D) \quad (\text{提取公因子}(B)) \\
&= (B \cdot 1 + D) \quad ((C + 1 = 1)) \\
&= (B + D) = \text{右边} \quad (\text{得证})
\end{aligned}$$

5. 需要输入高电平的管脚是

$$D_0、D_1、D_6、D_7$$

，需要输入低电平的管脚是

$$D_2、D_3、D_4、D_5$$

复习回顾：补充知识：CT74151的输出逻辑 8选1数据选择器CT74151的输出 Y 为：

$$Y = \overline{A}\overline{B}\overline{C}D_0 + \overline{A}\overline{B}CD_1 + \overline{A}B\overline{C}D_2 + \overline{A}BCD_3 + A\overline{B}\overline{C}D_4 + A\overline{B}CD_5 + AB\overline{C}D_6 + ABCD_7$$

6.

X[3:0]	十六进制	A	B	C	D	E	F	G
0000	0	1	1	1	1	1	1	0
0001	1	0	1	1	0	0	0	0
0010	2	1	1	0	1	1	0	1
0011	3	1	1	1	1	0	0	1
0100	4	0	1	1	0	0	1	1
0101	5	1	0	1	1	0	1	1
0110	6	1	0	1	1	1	1	1
0111	7	1	1	1	0	0	0	0
1000	8	1	1	1	1	1	1	1
1001	9	1	1	1	1	0	1	1
1010	A	1	1	1	0	1	1	1
1011	B	0	0	1	1	1	1	1

X[3:0]	十六进制	A	B	C	D	E	F	G
1100	C	1	0	0	1	1	1	0
1101	D	0	1	1	1	1	0	1
1110	E	1	0	0	1	1	1	1
1111	F	1	1	1	0	0	0	1

通过卡诺图化简，得到各段的最简与或式：

$$a = \overline{x_3x_2x_0} + \overline{x_3x_2x_1} + x_3\overline{x_2} + x_3x_2\overline{x_0}$$

$$b = \overline{x_3x_2} + \overline{x_3x_2x_0} + x_3\overline{x_2} + x_3x_2x_1$$

$$c = \overline{x_3x_2} + \overline{x_3x_2} + x_3\overline{x_2} + x_3x_2x_1$$

$$d = \overline{x_3x_2x_0} + \overline{x_3x_2x_1x_0} + \overline{x_3x_2x_1x_0} + x_3\overline{x_2x_1x_0} + x_3\overline{x_2x_1x_0}$$

$$e = \overline{x_3x_2x_0} + \overline{x_3x_2x_1x_0} + \overline{x_3x_2x_1x_0} + \overline{x_3x_2x_1x_0} + x_3\overline{x_2x_1x_0}$$

$$f = \overline{x_3x_2x_0} + \overline{x_3x_2x_1} + \overline{x_3x_2x_1x_0} + \overline{x_3x_2x_1x_0} + x_3\overline{x_2x_1x_0}$$

$$g = \overline{x_3x_2x_1} + \overline{x_3x_2x_1x_0} + \overline{x_3x_2x_1x_0} + \overline{x_3x_2x_1x_0} + x_3\overline{x_2x_1x_0}$$

Verilog-HDL语言如下

```
module seven_segment_decoder(
    input [3:0] x,
    output a, b, c, d, e, f, g
);

assign a = (~x[3]&~x[2]&~x[0]) | (~x[3]&x[2]&x[1]) | (x[3]&~x[2]) |
(x[3]&x[2]&~x[0]);
assign b = (~x[3]&~x[2]) | (~x[3]&x[2]&~x[0]) | (x[3]&~x[2]) |
(x[3]&x[2]&x[1]);
assign c = (~x[3]&~x[2]) | (~x[3]&x[2]) | (x[3]&~x[2]) |
(x[3]&x[2]&x[1]);
assign d = (~x[3]&~x[2]&~x[0]) | (~x[3]&x[2]&~x[1]&~x[0]) |
(~x[3]&x[2]&x[1]&x[0]) | (x[3]&~x[2]&~x[1]&x[0]) |
(x[3]&~x[2]&x[1]&~x[0]);
```

```
assign e = (~x[3]&~x[2]&~x[0]) | (~x[3]&x[2]&~x[1]&~x[0]) |  
(~x[3]&x[2]&~x[1]&x[0]) | (~x[3]&x[2]&x[1]&~x[0]) |  
(x[3]&~x[2]&~x[1]&x[0]);  
assign f = (~x[3]&~x[2]&~x[0]) | (~x[3]&~x[2]&x[1]) |  
(~x[3]&x[2]&~x[1]&~x[0]) | (~x[3]&x[2]&~x[1]&x[0]) |  
(x[3]&~x[2]&~x[1]&x[0]);  
assign g = (~x[3]&~x[2]&~x[1]) | (~x[3]&x[2]&x[1]&x[0]) |  
(x[3]&~x[2]&~x[1]&x[0]) | (x[3]&~x[2]&x[1]&~x[0]) |  
(x[3]&x[2]&~x[1]&~x[0]);  
  
endmodule
```