补充作业1：

- 四名学生（A, B, C, D）申请出国奖学金，现有如下条件需要满足：
  - ① A和B至少有一人被选中；
  - ② A和D不能同时被选中；
  - ③C和D中有且只有一人获得奖学金；
  - ④ B和C或者同时获得奖学金，或者同时失去奖学金；
- 请写出申请结果的逻辑表达式并化简为最简与或式。

$$\bar{A}B + A\bar{B} + AB \qquad A + B$$

$$\bar{A}D + A\bar{D} + \bar{A}\bar{D} \qquad \overline{AD}$$

$$\bar{C}D + C\bar{D}$$

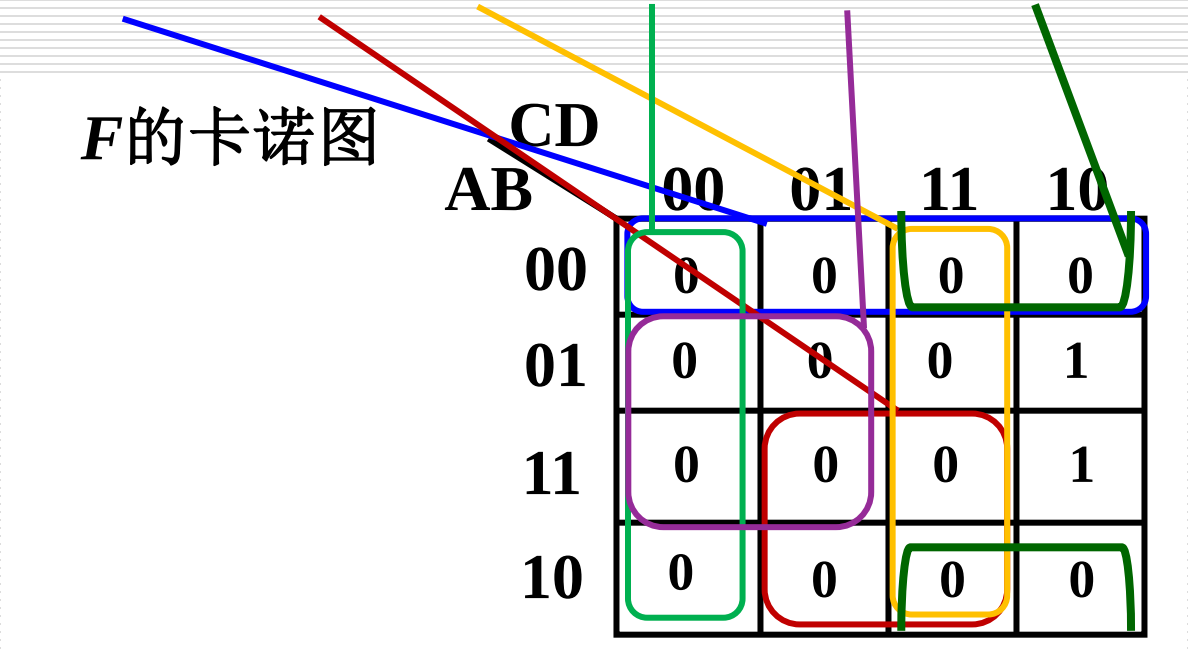
$$BC + \bar{B}\bar{C}$$

补充作业1：

思路1： $F=(\bar{A}B+A\bar{B}+AB)(\bar{A}D+A\bar{D}+\bar{A}\bar{D})(\bar{C}D+C\bar{D})(BC+\bar{B}\bar{C})$

思路2： $F=(A+B)(\bar{A}+\bar{D})(\bar{C}D+C\bar{D})(BC+\bar{B}\bar{C})$

$$\bar{F}=\bar{A}\bar{B}+AD+CD+\bar{C}\bar{D}+B\bar{C}+\bar{B}C$$



补充作业1：

思路1： $F=(\bar{A}B+A\bar{B}+AB)(\bar{A}D+A\bar{D}+\bar{A}\bar{D})(\bar{C}D+C\bar{D})(BC+\bar{B}\bar{C})$

思路2： $F=(A+B)(\bar{A}+\bar{D})(\bar{C}D+C\bar{D})(BC+\bar{B}\bar{C})$

$\bar{F}=\bar{A}\bar{B}+AD+CD+\bar{C}\bar{D}+B\bar{C}+\bar{B}C$

F的卡诺图

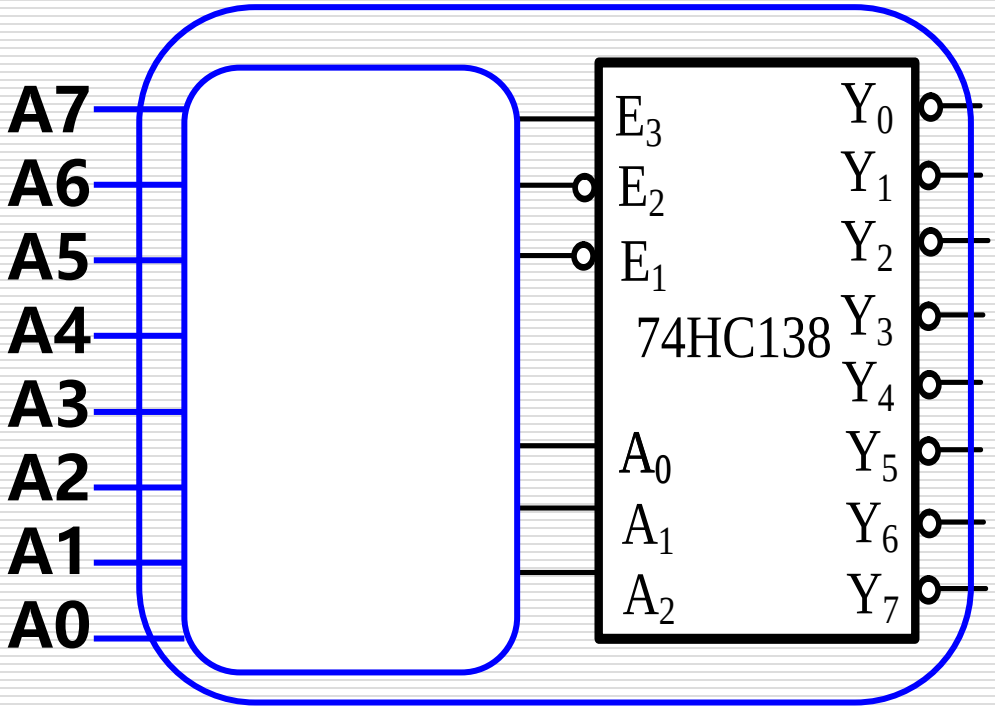
AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	0	0	1
10	0	0	0	0

$F=BC\bar{D}$

补充作业2：

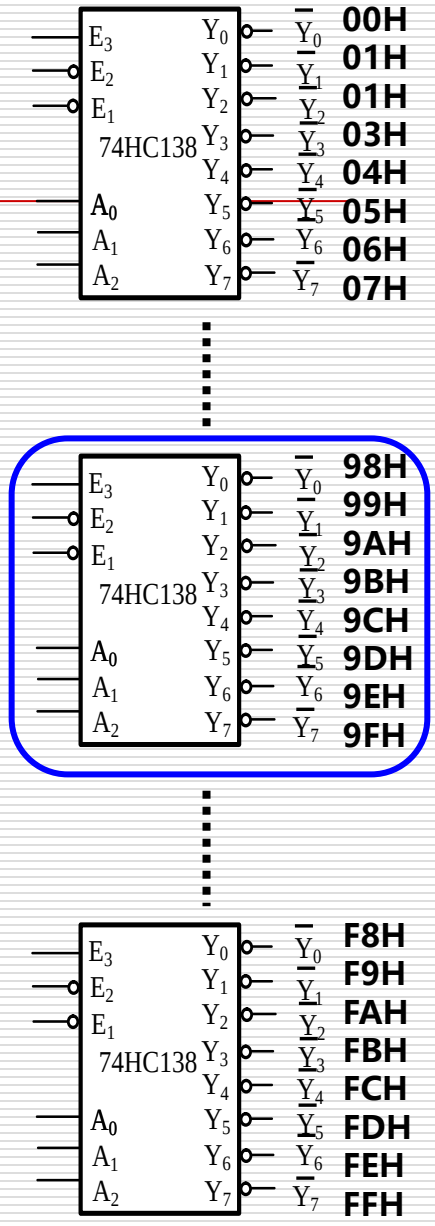
仅用一片74LS138设计地址译码器，译出输入地址$A_7,\ldots,A_0$ = 98H,…,9FH。

思路：
8根输入地址线的可能取值范围是00H…FFH



补充作业2：

思路： 98H,…,9FH理解为8个地址（房间号）。片内译码就是要利用8个输入信号中发生变化的bit，观察可发现：A7,…,A3 没有发生变化，所以利用A2,…,A0 连接138的CBA实现地址译码，利用高五位产生片选信号，当且仅当[A7,…,A3]=10011时本片138使能



补充作业3：

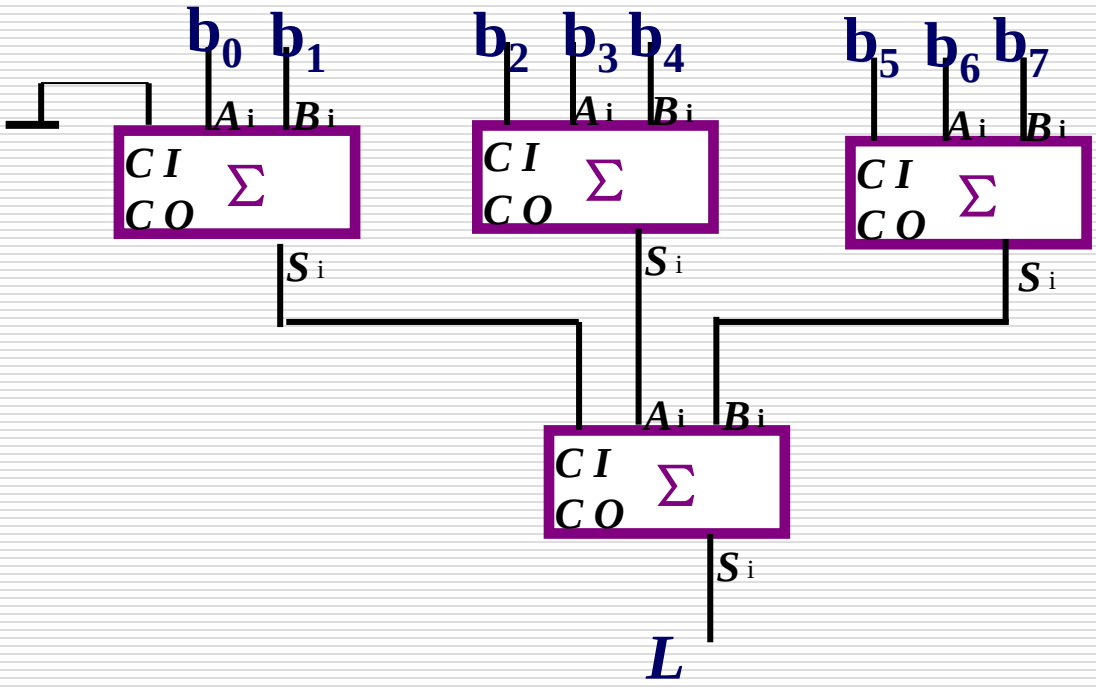
仅用全加器组成八位二进制代码奇校验器，电路应如何连接？

补充作业3：

全加器组成的八位二进制代码奇偶校验器

$$L = b_7 \oplus b_6 \oplus b_5 \oplus b_4 \oplus b_3 \oplus b_2 \oplus b_1 \oplus b_0$$

S_{i-3}	S_{i-2}	S_{i-1}	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



补充作业4：

试仅用一片74LS151，不加任何门电路实现逻辑函数：

$$F(ABCD)=\sum m(2,5,6,7,8,10,11,12,14,15)$$

思路与前一题类似

$$F=C\overline{D}+A\overline{D}+AC+\overline{A}BD+BC$$

选择A、C、D作为通道选择信号，A为最高位

$$F=C\overline{D}+A\overline{D}+AC+\overline{A}BD+BC$$
$$=\overline{A}C\overline{D}+A\overline{C}\overline{D}+ACD+AC\overline{D}+\overline{A}CDB+A\overline{C}DB$$

D0---D7:0 B 1 B 1 0 1 1

补充作业5：

目前闰年的判定准则可简要归纳为“非百能被4整除，整百能被400整除”。一种verilogHDL实现参考代码如下。其中year_bcd[15:0]存放已通过合法性检查的4位年份对应的BCD码。当输入为闰年时，输出低电平有效的闰年指示leap_year_n。跟据给出的判决思路，填写代码中划线部分；

```
module bcd_div_by_4(bcd,div_by_4);
   input [4:0]bcd;                    // two bcd codes: [7:4]= tens digit,[3:0]=units digit
   output div_by_4;                   // can be divided by 4 when 1
   ____________________;
always @(*)
begin
   if(bcd[4])                         // odd
      div_by_4 = (bcd[3:0] ==____)|(bcd[3:0] ==____);
   else                               // even
      div_by_4 = (bcd[3:0] == 4'd0)|(bcd[3:0] == 4'd4)|(bcd[3:0] == 4'd8);
end
endmodule

module leap_year_judge(year_bcd,leap_year_n);
   input [15:0]year_bcd;              //four bcd codes: [15:12]=thousands digit,[11:8]=hundreds
digit,[7:4]= tens digit,[3:0]=units digit
   output leap_year_n;                // 0: leap_year, 1:common year

wire div_by_100,div_by_4,div_by_4_wh;    //whole hundreds can be divided by 4
bcd_div_by_4  year_bcd_div_by_4(.bcd(year_bcd[4:0]),.div_by_4(div_by_4));
bcd_div_by_4  whole_hundred_div_by_4(.bcd(year_bcd[12:8]),.div_by_4(div_by_4_wh));
assign div_by_100 = (year_bcd[7:4] == 4'h0) & (year_bcd[3:0] == 4'h0);
assign leap_year_n = ~((_____)|(_____));
endmodule
```

```
module bcd_div_by_4(bcd,div_by_4);
    input [4:0]bcd;                 // two bcd codes: [7:4]= tens digit,[3:0]=units digit
    output div_by_4;                // can be divided by 4 when 1
    reg div_by_4 ;
always @(*)
begin
    if(bcd[4])                      // odd
        div_by_4 = (bcd[3:0] == 4'd2 )|(bcd[3:0] == 4'd6);
    else                            // even
        div_by_4 = (bcd[3:0] == 4'd0)|(bcd[3:0] == 4'd4)|(bcd[3:0] == 4'd8);
end
endmodule

module leap_year_judge(year_bcd,leap_year_n);
    input [15:0]year_bcd;           //four bcd codes: [15:12]=thousands digit,[11:8]=hundreds
digit,[7:4]= tens digit,[3:0]=units digit
    output leap_year_n;             // 0: leap_year, 1:common year

wire div_by_100,div_by_4,div_by_4_wh;    //whole hundreds can be divided by 4
bcd_div_by_4   year_bcd_div_by_4(.bcd(year_bcd[4:0]),.div_by_4(div_by_4));
bcd_div_by_4   whole_hundred_div_by_4(.bcd(year_bcd[12:8]),.div_by_4(div_by_4_wh));
assign div_by_100 = (year_bcd[7:4] == 4'h0) & (year_bcd[3:0] == 4'h0);
assign leap_year_n = ~(((~div_by_100) & div_by_4 )|( div_by_100 & div_by_4_wh  ));
endmodule
```