一, 填空题 (15个空, 15分)

1. 八进制$(-15)_8$转二进制, 写原码(10001101), 补码(11110011)

对于正数原码补码反码都一样，对于负数，反码符号位不变其他位取反，补码在反码基础上+1

2. 逻辑运算的三个基本操作(与或非)

3. 组合逻辑因为时延不同会出现(竞争和冒险)

4. 卡诺图的逻辑相邻是指(指有且仅有一项与给出的最小项不同)

   几何相邻：一个最小项的几何相邻就是它的上下左右的最小项。

5. JK触发器特性方程($Q^{n+1}=J\bar{Q}^n+\bar{K}Q^n$)
6. 求一个给定函数L的对偶函数($\bar{L}$)
7. 译码器是组合逻辑电路还是时序逻辑电路? (组合逻辑电路)
8. 最大项Mi 和 最小项mi 什么关系? ($M_i=\bar{m}_i$(互补关系))
9. `timescale 5ns/1ps 什么意思? (时间单位5ns, 时间精度1ps)
10. 定义一个具有256个8位的存储器my_mem(reg[7:0] my_mem[255:0])
11. Verilog 程序 四大部分 模块声明, **端口定义**_, ***信号类型声明***__, 功能描述

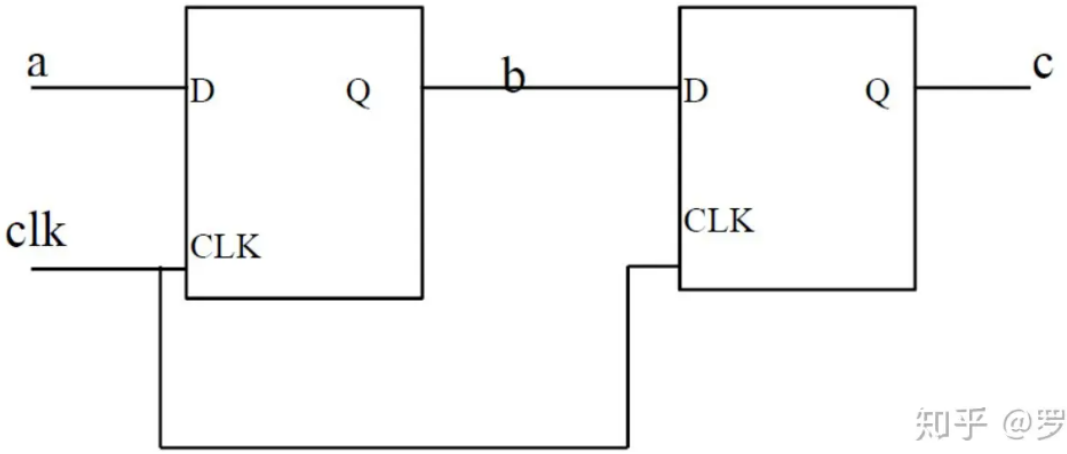
二, 根据代码画图 (2道 * 5分 = 10分)

1. 非阻塞 c<=b; b<=a;

   给了非阻塞的代码

```
always @(posedge clk)
    begin
        b<=a;
        c<=b;
    end
```

   这图是答案

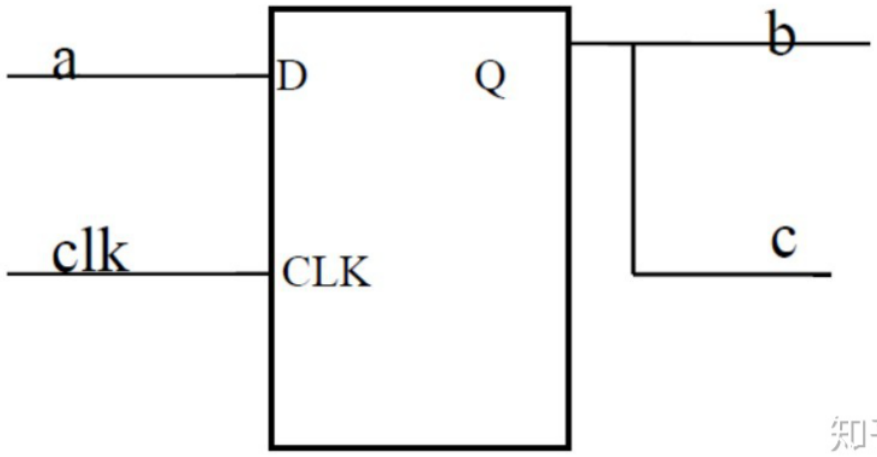


2. 阻塞　b=a; c=b;

   给了非阻塞的代码

```
always @(posedge clk)
    begin
        b=a;
        c=b;
    end
```

这图是答案

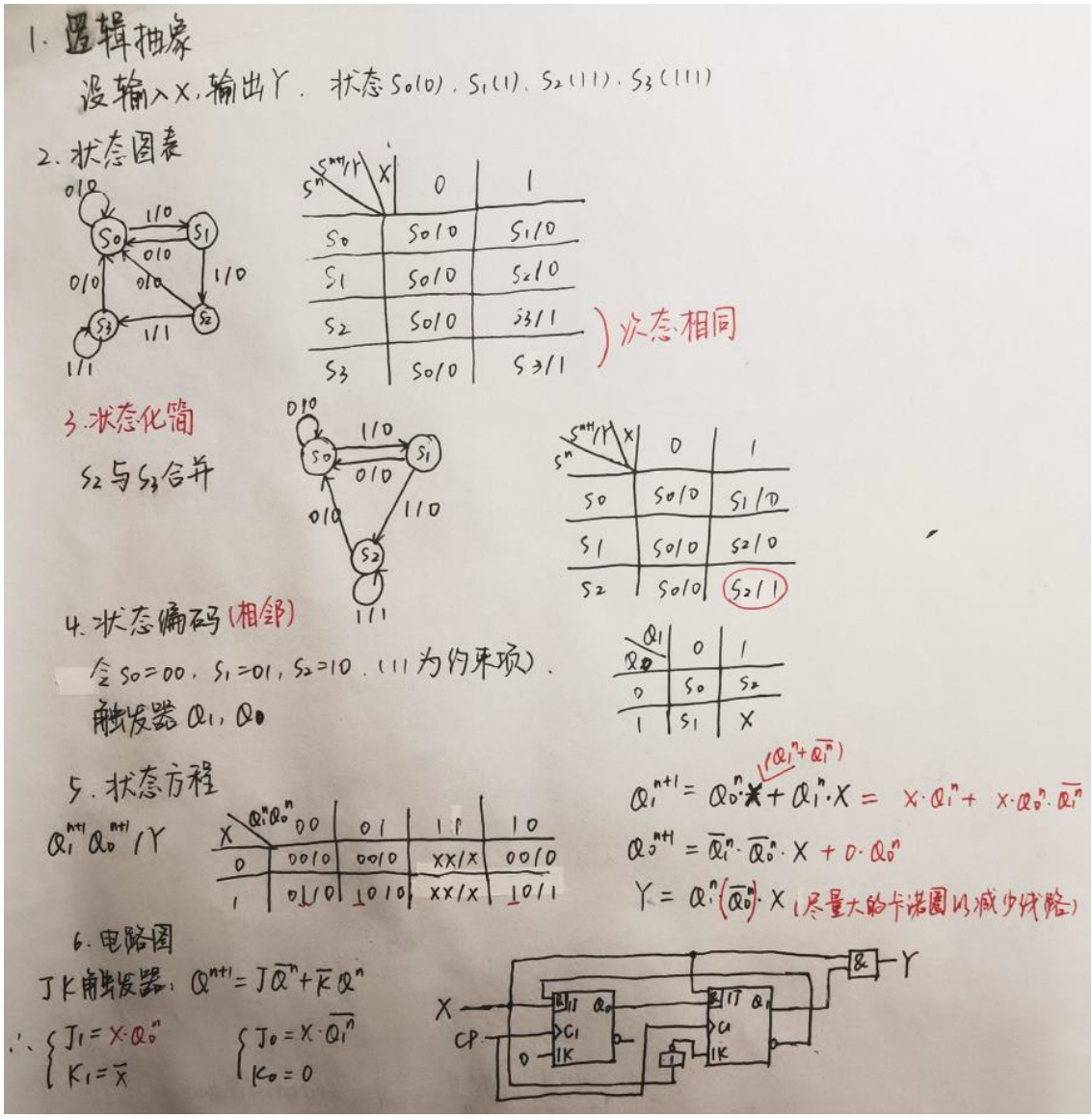


三, 分析与设计 (2道 * 15分 = 30分)

1. 给了一个F(A,B,C,D) = ΠM(.....), 请用卡诺图化简成**最简或与式**,然后再设计出对应的组合逻辑电路

   这道题记不清了

2. 用JK触发器设计"111"序列检测器



四, 代码(4道 : 15分 +15分 +10分 +5分 = 45分)

1. 四位计数器(load, en, clk, reset, 进位q, 等等)　*15分*

```
module cnt16 (cout, q, clk, clr, load, en, d); // 定义一个名为cnt16的模块，它有7个端口
  output[3:0] q; // 定义一个4位的输出信号q，它是计数器的当前值
  output cout; // 定义一个输出信号cout，它是计数器的进位信号
   input clk, clr, load, en; // 定义四个输入信号，分别是时钟、清零、加载和使能
   input[3:0] d; // 定义一个4位的输入信号d，它是计数器的加载值
   reg[3:0] q; // 定义一个4位的寄存器q，用来存储计数器的值
   reg cout; // 定义一个寄存器cout，用来存储计数器的进位信号
  always @  (posedge clk)  begin // 在时钟上升沿时执行以下语句
    if (clr) q <= 0; // 如果清零信号为1，那么将q赋值为0
    else if (load)  q <= d; // 否则，如果加载信号为1，那么将q赋值为d
            else if (en) begin // 否则，如果使能信号为1，那么执行以下语句
                 q <= q + 1; // 将q加1
                 if(q == 4'b1111) begin cout <= 1; end // 如果q等于4'b1111，那么将
cout赋值为1
                 else begin  cout <= 0; end // 否则，将cout赋值为0
                    end
                    else begin q <= q; end // 否则，保持q不变
```

```
    end
endmodule // 结束模块定义
```

2. 具有异步清零的D触发器　　*15分*

```
module flipflop(D,Clock,Resetn,Q);
    input D,Clock,Resetn;
    output Q;
    reg Q;
    always @(negedge Resetn or posedge Clock)
        if(!Resetn)
            Q<=0;
        else
            Q<=D;
endmodule
```

3. 具有同步清零功能的移位寄存器　　*10分*

这段代码是一个**串行输入并行输出**的移位寄存器Verilog代码，它具有**同步清零**功能。当`clr`信号为1时，输出`dout`被清零；当`clr`信号为0时，每个时钟上升沿，输出`dout`向左移位一位，并将输入`din`接到最低位。这样，输入的串行数据被转换为输出的并行数据。

```
module sipo (dout, din,clr,clk);//串入并出
  output[4:0] dout;
  input clk, din,clr;
  reg[4:0] dout;  //五位
 always @ (posedge clk )
   begin
      if(clr) begin
    dout <= 0;
      end
      else begin
            dout <= {dout, din};
      end
endmodule
```

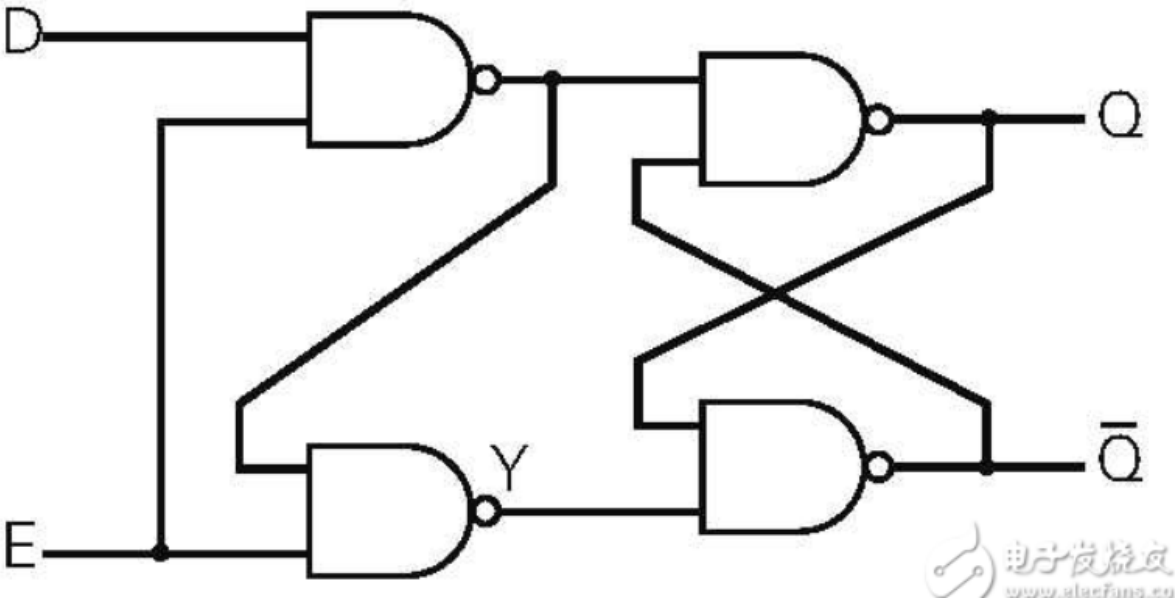
4. 根据输入输出图, 写代码(其实就是一个D锁存器, 电平触发) *5分*

电平触发是在高或低电平保持的时间内触发，而边沿触发是由高到低或由低到高这一瞬间触发,边沿触发和电平触发基本就是触发器和锁存器的区别。

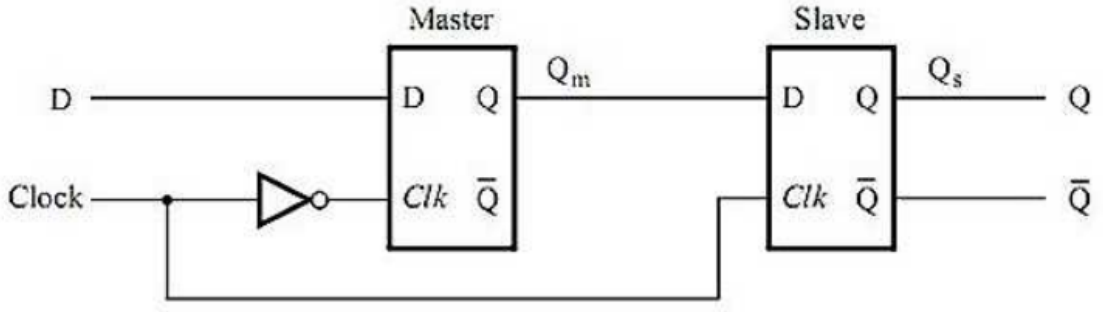
触发器是边沿触发，只有当时钟上升（或下降）的一瞬间，触发器会读取并锁存输入信号。输出信号仅在时钟信号上升（或下降）的一瞬间会发生变化。

锁存器是电平触发，只要使能（enable）信号处于高电平（或低电平），输出就会随着输入信号变化，直到使能信号变为低电平（或高电平）时，输出才会锁存，不再随输入变化。

D锁存器



D触发器(两D锁存器级联)




```
module latch(
            input D,
            input EN,    //输入信号

            output Q   //输出信号
            );
 always@（EN or D）   //组合逻辑
 begin
 if(EN)
    Q = D;  //阻塞赋值
 end
 endmodule
```

锁存器代码还可以这样写：

```
module latch(
            input D,
            input EN,    //输入信号

            output Q   //输出信号
            );
assign Q= EN?D:Q;   //组合逻辑
endmodule
```