

武汉大学2021级弘毅班《数字逻辑》期末试题（A卷）答案

班号：_____ 学号：_____ 姓名：_____

题号	一	二	三	四	五	六	总分
得分							

一、（10分）化简下列逻辑表达式。

1. 用 **数** 将 $F = (A + \bar{B})(\bar{A} + B)(B + C)(\bar{A} + C)$ 化简为 “ — ” ；（5分）
2. 用 **图** 将 $F(A, B, C, D) = \sum m(0, 2, 7, 13, 15) + \sum d(1, 3, 4, 5, 6, 8, 10)$ 为 “ — ” 。（5分）

解：1. 先求得 F 的对偶式 F^* 并进行化简：

$$\begin{aligned}
 F^* &= A\bar{B} + \bar{A}B + BC + \bar{A}C \\
 &= A\bar{B} + \bar{A}B + (\bar{A} + B)C \\
 &= A\bar{B} + \bar{A}B + \bar{A}\bar{B}C \\
 &= A\bar{B} + \bar{A}B + C
 \end{aligned}$$

$$\therefore F = (F^*)^* = (A + \bar{B})(\bar{A} + B)C$$

2. 作出 F 的卡诺图如下：

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

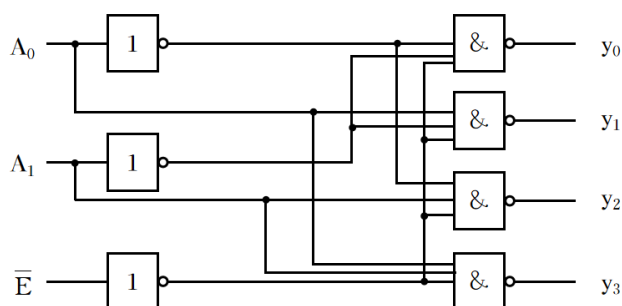
$$\text{得 } F = \bar{A} + BD$$

二、（12分）分析图1所示组合逻辑电路，回答问题：

1. 求输出函数的逻辑表达式；（4分）
2. 列出函数真值表；（4分）
3. 判断此逻辑电路功能。（4分）

解：1.

$$y_0 = A_0 + A_1 + \bar{E}$$



$$y_1 = \overline{A_0} + A_1 + \overline{E}$$

$$y_2 = A_0 + \overline{A_1} + \overline{E}$$

$$y_3 = \overline{A_0} + \overline{A_1} + \overline{E}$$

2. 当 $\overline{E} = 1$ 时，函数值均为 1，当 $\overline{E} = 0$ 时，函数真值表如下：

A_1	A_0	y_0	y_1	y_2	y_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

3. 该电路输入一个两位二进制数，为 2-4 译码器。

三、（22 分）设 ABCD 输入值为 8421BCD 码，试设计一个电路判断其输入数值是否为质数的电路，当输入为质数（含 1）时 $F=1$ ，否则 $F=0$ 。

1. 写出函数 F 与 ABCD 的真值表；（3 分）
2. 写出逻辑函数表达式并化简；（3 分）
3. 用适当门电路实现该函数并画出逻辑电路图；（3 分）
4. 用三一八译码器 74LS138 及适当门电路实现该电路的逻辑功能；（3 分）
5. 用中规模集成电路八选一 74LS151 及适当门电路实现其逻辑功能；（3 分）
6. 用 Verilog HDL 语言设计该逻辑电路；（4 分）
7. 用如图 2 所示的可编程逻辑器件 PLA 设计该逻辑函数。（3 分）

解：1. 真值表如下：

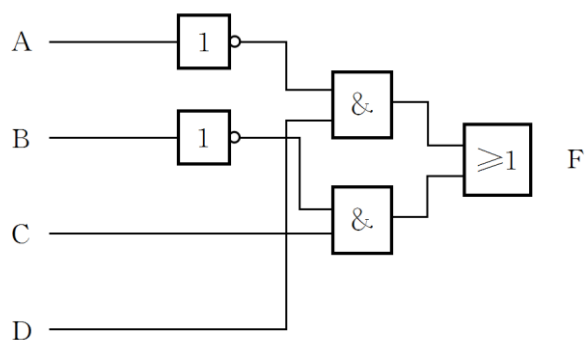
ABCD	F	ABCD	F
0000	0	1000	0
0001	1	1001	0
0010	1	1010	<i>d</i>
0011	1	1011	<i>d</i>
0100	0	1100	<i>d</i>
0101	1	1101	<i>d</i>
0110	0	1110	<i>d</i>
0111	1	1111	<i>d</i>

2. 函数逻辑表达式为 $F(A, B, C, D) = \sum m(1, 2, 3, 5, 7) + \sum d(10, 11, 12, 13, 14, 15)$
作出卡诺图：

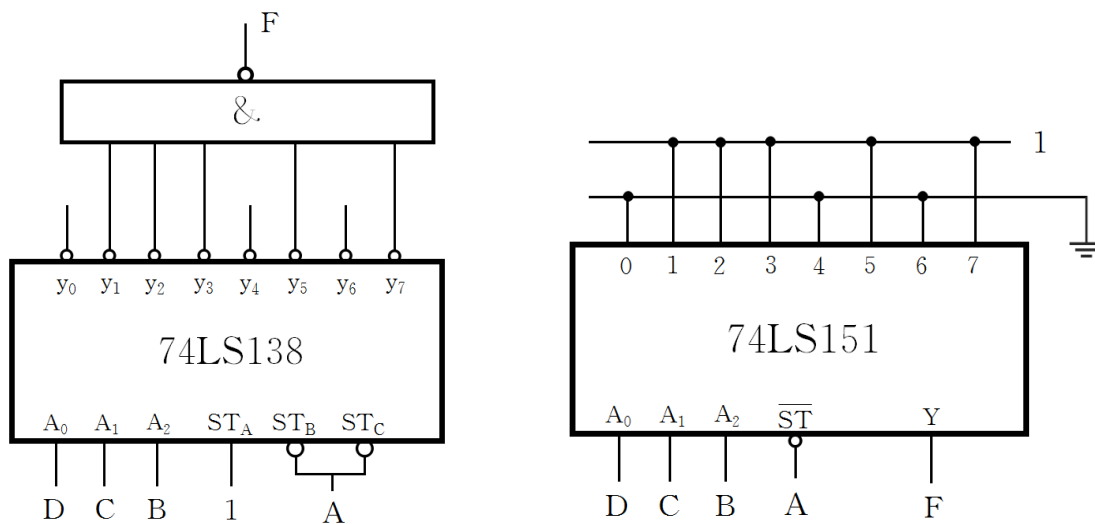
AB \ CD	00	01	11	10
00		1	1	1
01		1	1	
11	d	d	d	d
10			d	d

$$\text{得 } F = \bar{A}D + \bar{B}C$$

3. 实现的参考电路如下（形式最简）：



4-5. 用 74LS138 连线与 74LS151 连线如下：



6. 参考代码如下：

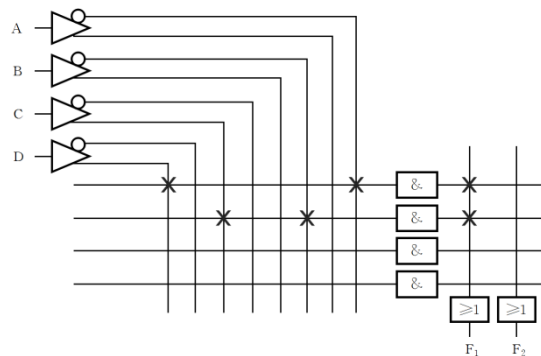
```
module detect (A,B,C,D,F)
    input A,B,C,D
    output F
);
```

```

assign F = ((~A) & D) | ((~B) & C);
endmodule

```

7. 用可编程逻辑器件 PLA 设计图如图：



四、（18 分）分析图 3 的时序逻辑电路，回答问题：

1. 写出激励函数与输出函数的表达式；（4 分）
2. 写出次态方程组；（4 分）
3. 作出电路的状态表及状态图；（6 分）
4. 分析电路的逻辑功能。（4 分）

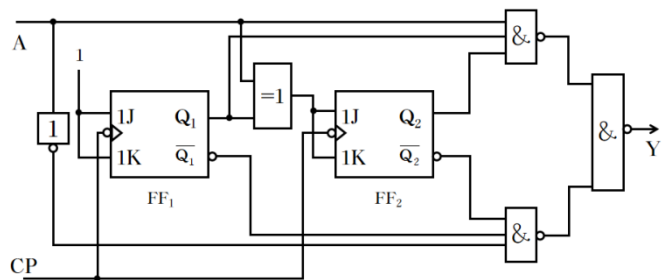


图 3

解：1. 由图可知，该电路为一个同步时序逻辑电路

$$J_1 = K_1 = 1$$

$$J_2 = Q_1 \oplus A = K_2$$

$$y = \overline{\overline{A} \cdot \overline{Q_1} \cdot \overline{Q_2} \cdot A} = \overline{\overline{A} \cdot \overline{Q_2} \cdot \overline{Q_1}} + A Q_2 Q_1$$

2. JK 触发状态方程为 $Q^{n+1} = JQ + \overline{K}Q$

$$Q_1^{n+1} = \overline{Q_1}$$

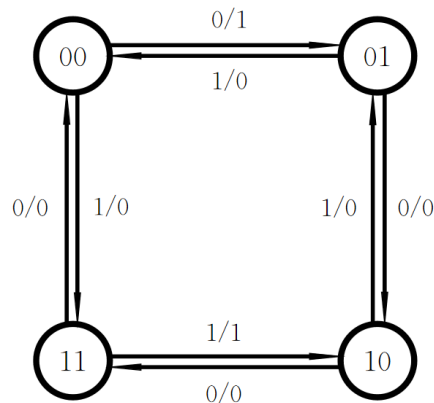
$$Q_2^{n+1} = (A \oplus Q_1) \cdot Q_2 + \overline{(A \oplus Q_1)} \cdot \overline{Q_2} = A \oplus Q_1 \oplus Q_2$$

$$y = \overline{\overline{A} \cdot \overline{Q_2} \cdot \overline{Q_1}} + A Q_2 Q_1$$

3. 状态转移真值表和状态图如下：

A	Q ₂	Q ₁	Q ₂ ⁿ⁺¹	Q ₁ ⁿ⁺¹	y
0	0	0	0	1	1

0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	1	0	0	1	0
1	1	1	1	0	1



4. 当 $A=0$ 时，状态由 $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$ 状态之间循环，00 状态时 $y=1$
 当 $A=0$ 时，状态由 $00 \rightarrow 11 \rightarrow 10 \rightarrow 01$ 状态之间循环，11 状态时 $y=1$
 则该电路为模 4 加减法可控计数电路， $A=0$ 时作为加法计数， $A=1$ 时作为减法计数

五、（16 分）现有 6 个数字逻辑功能部件需要轮流工作，每个功能部件都有一个片选端 $\overline{CS}_0 - \overline{CS}_5$ ，且低电平有效。试用计数器 74LS161 和 3-8 译码器 74LS138 设计产生各个功能部件的片选信号。以便在 6 个 CP 时钟周期内，每个功能部件都能够分时依次被选通，并依此循环工作。具体功能要求如图 4 波形图。

- 1、请说明设计思路（8 分）
- 2、画出实现原理图（8 分）

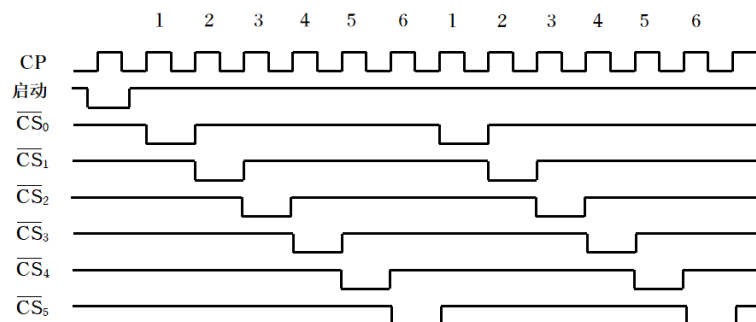
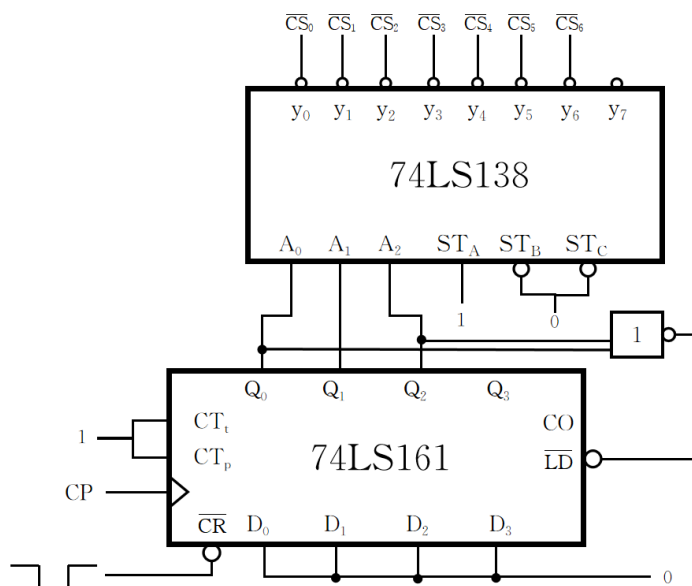


图 4

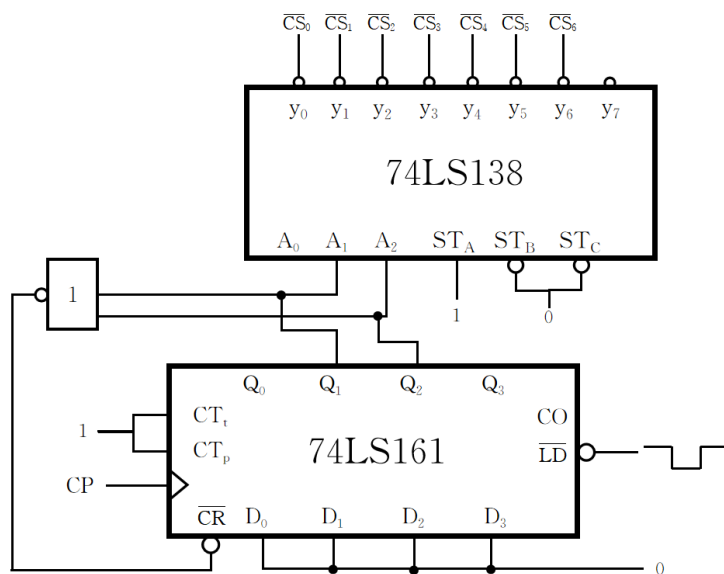
1. 设计思路：将 74LS161 设计成一个模 6 的六进制计数电路，则 6 进制输出状态为 $Q_2Q_1Q_0$ ：000,001,010,011,100,101，将 6 进制输出状态 $Q_2Q_1Q_0$ 分别接入 74LS138 的输入端 A_2 、 A_1 、 A_0 ，将其控制端 $\overline{ST_A}$ 、 $\overline{ST_B}$ 、 $\overline{ST_C}$ 接为有效状态，则其输出信号 $y_0 \sim y_5$ ，分别表示为 $\overline{CS_0}$ 、 $\overline{CS_1}$ 、 $\overline{CS_2}$ 、 $\overline{CS_3}$ 、 $\overline{CS_4}$ 、 $\overline{CS_5}$ ，通常 $\overline{CS_i}$ 为高电平，在 CP 脉冲作用下 $Q_2Q_1Q_0$ 的状态依次输出为 000,001,010,011,100,101；则 $\overline{CS_0} \sim \overline{CS_5}$ 依次输出低电平。

2. 具体连线图如下：

法一：选用同步置数法，即从 000~101 状态， $\overline{LD} = \overline{Q_2Q_0}$ ，启动信号接 \overline{CR} 端。



法二：选用异步清零法，即从 000~101 状态，110 为暂态， $\overline{CR} = \overline{Q_2Q_1}$ ，不受 CP 影响即产生 CR。



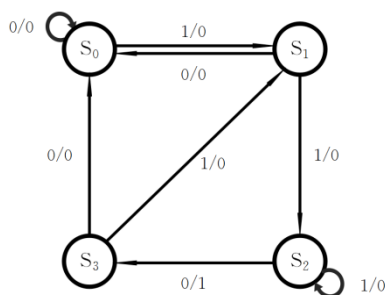
六、(22 分) 用正边沿 D 触发器作为存储元件, 设计一个同步时序逻辑电路 “110” 序列检测器。该电路有一个输入 x 和一个输出 Z , 当随机输入信号中出现 “110” 序列时, 输出一个 1 信号。典型输入/输出序列如下:

输入 x	0	0	1	1	1	0	1	1	0	1	0	0
输出 Z	0	0	0	0	0	1	0	0	1	0	0	0

1. 作出原始状态图、状态表及二进制状态表; (6 分)
2. 求 D 触发器的各输入端的激励表达式及输出函数表达式; (6 分)
3. 画出相应电路图、并分析该电路是否能够自启动; (5 分)
4. 用 Verilog HDL 语言描述状态图。(5 分)

解: 1 采用 mealy 电路, 设 S_0 状态为初态, S_1 为输入 “1”, S_2 为输入 “11”, S_3 为输入 “110”, 可作出原始状态图与状态表如下:

现态	次态 Z	
	$x=0$	$x=1$
S_0	$S_0/0$	$S_1/0$
S_1	$S_0/0$	$S_2/0$
S_2	$S_3/1$	$S_2/0$
S_3	$S_0/0$	$S_1/0$



从上表可看出, S_0 与 S_3 状态等效, 故可简化状态表:

现态	次态 Z	
	$x=0$	$x=1$
S_0	$S_0/0$	$S_1/0$
S_1	$S_0/0$	$S_2/0$
S_2	$S_0/1$	$S_2/0$

状态分配时, 按照三个规则分配, 可得 S_0 与 S_1 相邻, S_0 与 S_2 相邻。将 “00” 分配给 S_0 , “01” 分配给 S_1 , “10” 分配给 S_2 , 二进制状态表如下:

		0		1
Q_1	0	S_0	S_2	
	1	S_1	d	

现态		次态/ Z	
Q_2	Q_1	$x=0$	$x=1$
0	0	00/0	01/0
0	1	00/0	10/0
1	0	00/1	10/0

2. 分别作出 Q_2^{n+1} 、 Q_1^{n+1} 、 Z 的真值表，

x	Q_2^n	Q_1^n	Q_2^{n+1}	Q_1^{n+1}	Z
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	1
0	1	1	d	d	d
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	d	d	d

用卡诺图化简，得：

$$Q_2^{n+1} = xQ_1^n + xQ_2^n \quad Q_1^{n+1} = x\overline{Q_2^n}\overline{Q_1^n} \quad Z = \overline{x}Q_2^n$$

∵ D 触发器中 $Q^{n+1} = D$,

$$\text{则 } D_2 = xQ_1^n + xQ_2^n \quad D_1 = x\overline{Q_2^n}\overline{Q_1^n} \quad Z = \overline{x}Q_2^n$$

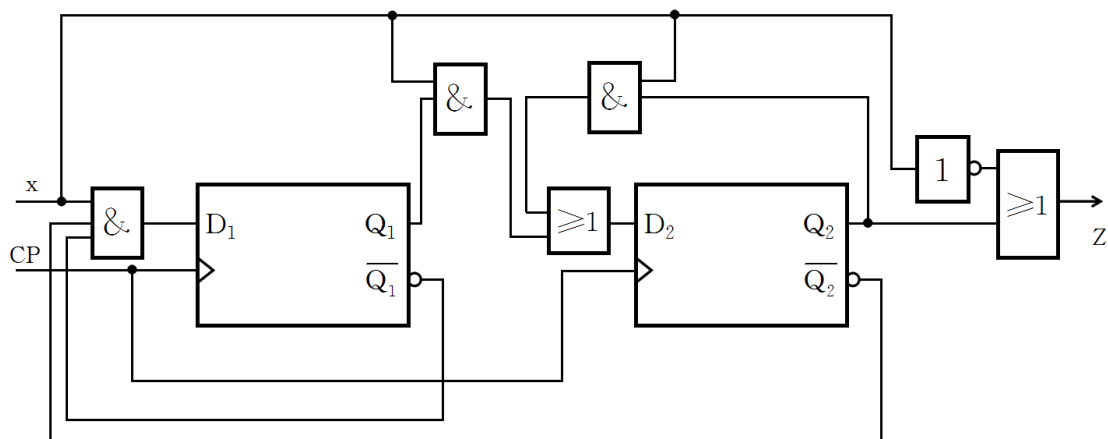
附：当采取另一种状态分配方法时，即将“00”分配给 S_0 ，“01”分配给 S_1 ，“11”分配给 S_2 ，二进制状态表如下：

		Q_2	
		0	1
Q_1	0	S_0	d
	1	S_1	S_2

现态		次态/ Z	
		$x=0$	$x=1$
Q_2	Q_1		
0	0	00/0	01/0
0	1	00/0	11/0
1	1	00/1	11/0

真值表略，则 $Q_2^{n+1} = xQ_1^n \quad Q_1^{n+1} = x \quad Z = \overline{x}Q_2^n$

3. 相应电路图如下（另一种状态分配图略）：



因为两个触发器使用了 00, 01, 10 三个状态，看其是否能自启动，即分析在“11”状态时，电路是否能回到正常状态。

由上图分析得，

$$Q_2^{n+1} = xQ_1^n + xQ_2^n$$

$$Q_1^{n+1} = x\overline{Q_2^n}Q_1^n$$

当 $Q_2^nQ_1^n$ 为“11”时， $Q_2^{n+1} = x$ ， $Q_1^{n+1} = 0$

即 $x=0$ 时，由“11”转变为“00”，

$x=1$ 时，由“11”转变为“10”，均能回到正常状态，

∴该电路能自启动。

4. 法一：

```
module mealy_test(clock, x, z);
    input clock, x;
    output reg z;
    reg [1:0] y;
    parameter [1:0] A=2'b00, B=2'b01, C=2'b10;
    always @ (posedge clock)
    case(y)
        A: if(x) begin y<=B; z<=0; end
           else begin y<=A; z<=0; end
        B: if(x) begin y<=C; z<=0; end
           else begin y<=A; z<=0; end
        C: if(~x) begin y<=A; z<=1; end
           else begin y<=B; z<=0; end
        default: begin y<=A; z<=0; end
    endcase
endmodule
```

```
    end case  
end module
```

法二:

```
module moore_test(clock, x, z);  
    input clock, x;  
    output reg z;  
    reg [3:0] y;  
    parameter[3:0] A=4'b0001, B=4'b0010, C=4'b0100, D=4'b1000;  
    always @(posedge clock)  
    case(y)  
        A: if(x) y<=B;  
            else y<=A;  
        B: if(x) y<=C;  
            else y<=A;  
        C: if(~x) y<=D;  
            else y<=C;  
        D: if(x) y<=B;  
            else y<=A;  
    end case  
    assign z = (y==D);  
end module
```