第一章 开关理论基础

1.将下列十进制数化为二进制数和八进制数

十进制	二进制	八进制
49	110001	61
53	110101	65
127	1111111	177
635	1001111011	1173
7.493	111.1111	7.74
79.43	10011001.0110111	231.334

2.将下列二进制数转换成十进制数和八进制数

二进制	十进制	八进制
1010	10	12
111101	61	75
1011100	92	134
0.10011	0.59375	0.46
101111	47	57
01101	13	15

3.将下列十进制数转换成 8421BCD 码

1997=0001 1001 1001 0111
65.312=0110 0101.0011 0001 0010
3.1416=0011.0001 0100 0001 0110
0.9475=0.1001 0100 0111 0101

4.列出真值表，写出 X 的真值表达式

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

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

5. 求下列函数的值

当 A,B,C 为 0,1,0 时：$\overline{A}B+BC=1$

$(A+B+C)(\overline{A}+\overline{B}+\overline{C})=1$

$(\overline{A}B+A\overline{C})B=1$

当 A,B,C 为 1,1,0 时：$\overline{A}B+BC=0$

$(A+B+C)(\overline{A}+\overline{B}+\overline{C})=1$

$(\overline{A}B+A\overline{C})B=1$

当 A,B,C 为 1,0,1 时：$\overline{A}B+BC=0$

$(A+B+C)(\overline{A}+\overline{B}+\overline{C})=1$

$(\overline{A}B+A\overline{C})B=0$

6. 用真值表证明下列恒等式

(1) $(A\oplus B)\oplus C=A\oplus(B\oplus C)$

A	B	C	$(A\oplus B)\oplus C$	$A\oplus(B\oplus C)$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

所以由真值表得证。

（2）$\overline{A}\oplus\overline{B}\oplus\overline{C}=A\oplus\overline{B}\oplus C$

A	B	C	$\overline{A}\oplus\overline{B}\oplus\overline{C}$	$A\oplus\overline{B}\oplus C$
0	0	0	1	1
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

7. 证明下列等式

（1） $A+\overline{A}B=A+B$

证明：左边$=A+\overline{A}B$

$=A(B+\overline{B})+\overline{A}B$

$=AB+A\overline{B}+\overline{A}B$

$=AB+A\overline{B}+AB+A\overline{B}$

$=A+B$

=右边

(2) $ABC+A\overline{B}C+AB\overline{C}=AB+AC$

证明：左边$=ABC+A\overline{B}C+AB\overline{C}$

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

$=AC(B+\overline{B})+AB(C+\overline{C})$

$=AB+AC$

=右边

（3） $A+A\overline{B}\overline{C}+\overline{A}CD+(\overline{C}+\overline{D})E$ =A+CD+E

证明：左边$=A+A\overline{B}\overline{C}+\overline{A}CD+(\overline{C}+\overline{D})E$

$=A+CD+A\overline{B}\,\overline{C}+\overline{CD}E$

$=A+CD+\overline{CD}E$

$=A+CD+E$

=右边

（4） $\overline{A}\overline{B}+A\overline{B}\overline{C}+\overline{A}B\overline{C}=\overline{A}\overline{B}+\overline{A}\overline{C}+\overline{B}\overline{C}$

证明：左边$=\overline{A}\overline{B}+A\overline{B}\overline{C}+\overline{A}B\overline{C}$

$=(\overline{A}\overline{B}+\overline{A}\overline{B}\overline{C})+AB\overline{C}+\overline{A}B\overline{C}$

$=\overline{A}\overline{B}+\overline{A}\overline{C}+\overline{B}\overline{C}$ =右边

8. 用布尔代数化简下列各逻辑函数表达式

(1) F=A+ABC+A$\overline{B}\overline{C}$+CB+$\overline{C}\,\overline{B}$ = A+BC+$\overline{C}\,\overline{B}$

(2) F＝(A+B+$\overline{C}$)(A+B+C) = (A+B)+C$\overline{C}$ = A+B

(3) F＝ABC$\overline{D}$+ABD+BC$\overline{D}$+ABCD+B$\overline{C}$ = AB+BC+BD

(4) F=$\overline{\overline{AC+\overline{A}BC+\overline{B}C+AB\overline{C}}}$ = BC

(5) F=$\overline{(\overline{A+B})+(\overline{A+\overline{B}})+(\overline{\overline{A}B})(\overline{A\overline{B}})}=A\overline{B}$

9. 将下列函数展开为最小项表达式

(1) F(A,B,C) = Σ(1,4,5,6,7)

(2) F(A,B,C,D) = Σ(4,5,6,7,9,12,14)

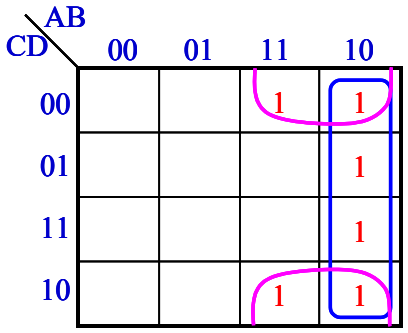
10. 用卡诺图化简下列各式

（1） $F=\overline{AC+\overline{A}BC+\overline{B}C+AB\overline{C}}$

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

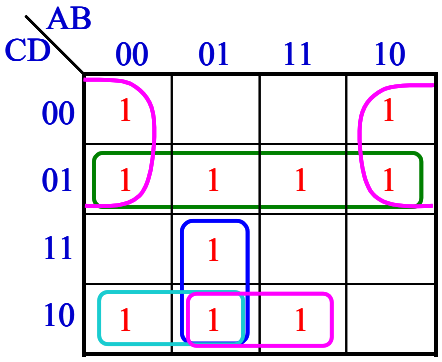
化简得 F=$\overline{C}$

(2) $F = A\overline{B}CD + AB\overline{C}\,\overline{D} + A\overline{B} + A\overline{D} + A\overline{B}C$



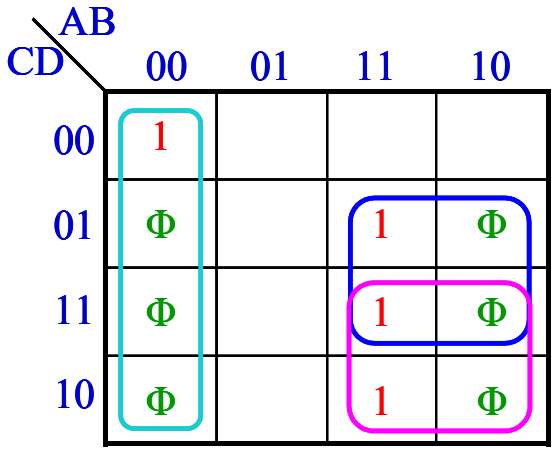
化简得 F= $A\overline{B} + A\overline{D}$

（3） F(A,B,C,D)=∑m(0,1,2,5,6,7,8,9,13,14)



化简得 F= $\overline{C}D + \overline{B}C + \overline{A}BC + \overline{A}C\overline{D} + BC\overline{D}$

(4) F(A,B,C,D)=∑m(0,13,14,15)+∑ φ (1, 2, 3, 9, 10, 11)

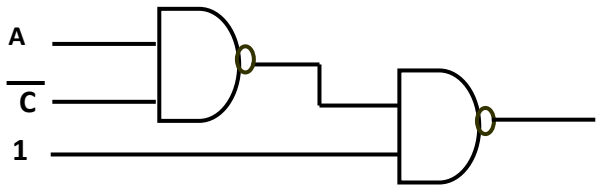


化简得 F= $\overline{A}\overline{B} + AD + AC$

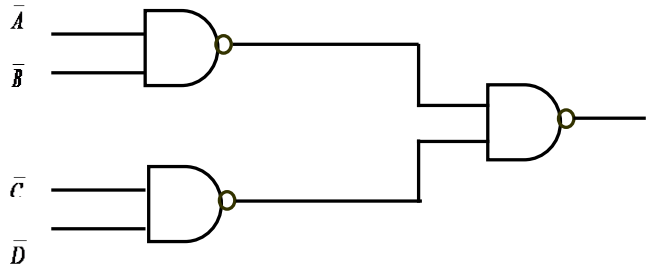
11. 利用与非门实现下列函数，并画出逻辑图。

（1） F= $AB\overline{C} + A\overline{B}\overline{C} = \overline{\overline{A\overline{C}} \bullet 1}$

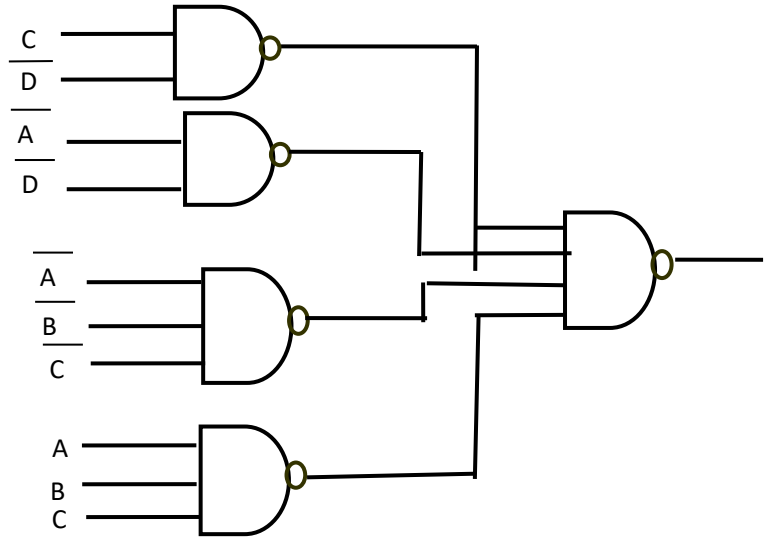
F<= (A nand (not C)) nand 1



(2) F=$\overline{(A+B)(C+D)}=\overline{(\overline{\overline{A}\overline{B}})(\overline{\overline{C}\overline{D}})}$



(3) F(A,B,C,D)=$\sum m(0,1,2,4,6,10,14,15)=\overline{(\overline{C\overline{D}})(\overline{A\overline{D}})(\overline{A\overline{B}\overline{C}})(\overline{A\overline{B}C})}$

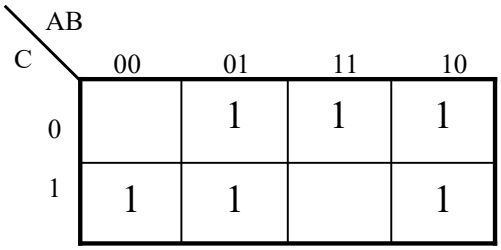


12. 已知逻辑函数$X=A\overline{B}+B\overline{C}+C\overline{A}$，试用以下方法表示该函数

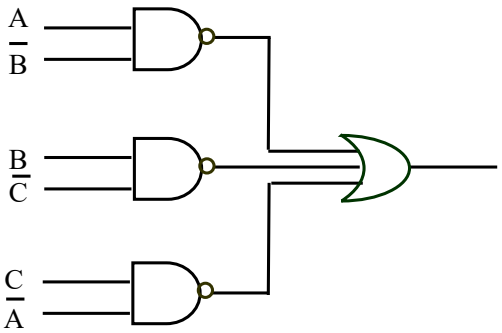
真值表：

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

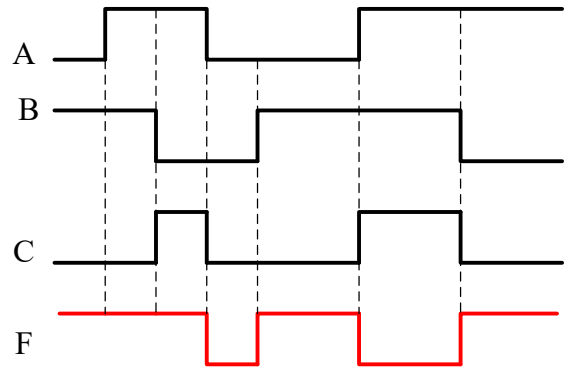
卡诺图：



逻辑图：



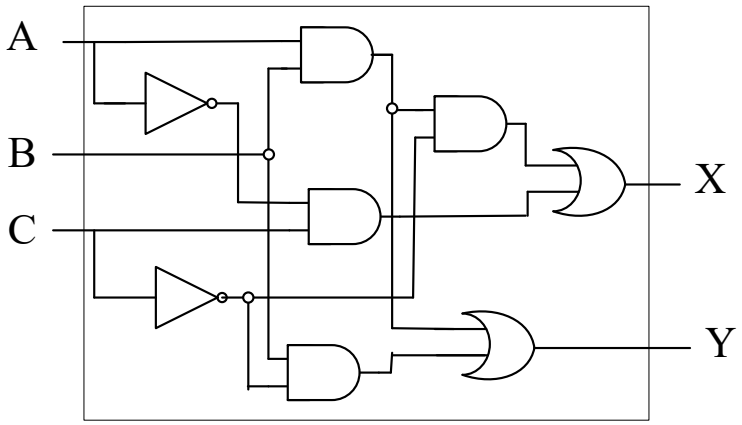
波形图



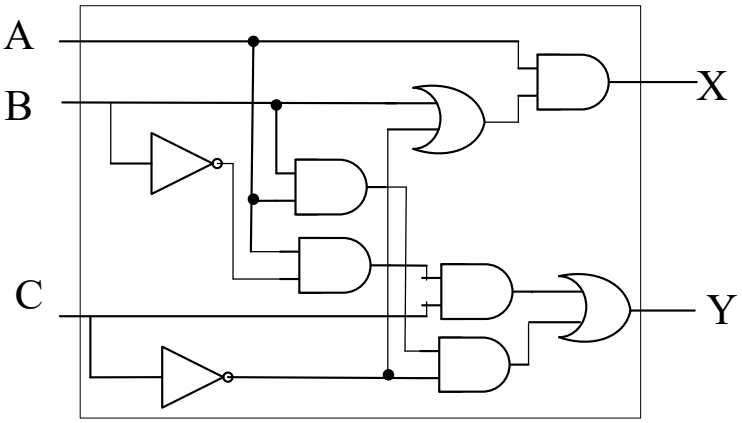
VHDL 语言

X<= (A and not B) or (B and not C) or (c and not A)

13. 根据要求画出所需的逻辑电路图。

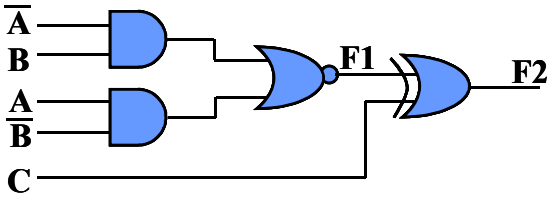


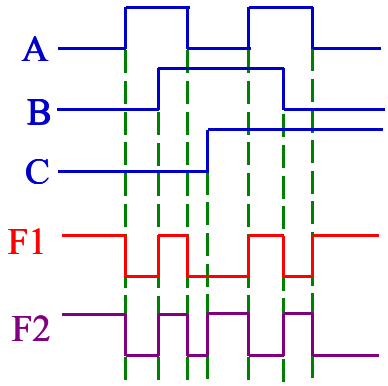
（a）



（b）

14..画出 F1,F2 的波形





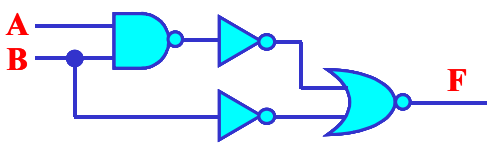
解：

$$F1 = \overline{A \oplus B}$$

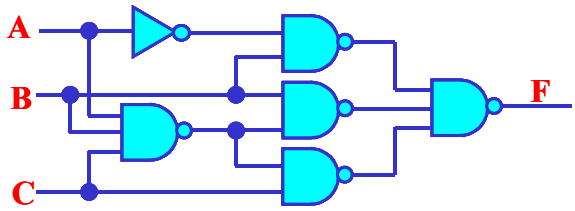
$$F2 = F1 \oplus C$$

第二章　组合逻辑

1. 分析图中所示的逻辑电路，写出表达式并进行化简



$$F = \overline{\overline{AB} + \overline{B}} = \overline{A}B$$



$$F=\overline{\overline{A\overline{B}}\ \overline{B\overline{A}BC}\ \overline{C\overline{A}BC}}$$

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

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

2. 分析下图所示逻辑电路，其中 S3、S2、S1、S0 为控制输入端，列出真值表，说明 F 与 A、B 的关系。

$F1=\overline{A+BS_0+\overline{B}S_1}$

$F2=\overline{ABS_2+A\overline{B}S_3}$

$F=F_1F_2=\overline{A+BS_0+\overline{B}S_1}$

S_1	S_0	F_1
0	0	$\overline{A}$
0	1	$\overline{A}\,\overline{B}$
1	0	$\overline{A}B$
1	1	0

S_3	S_2	F_2
0	0	1
0	1	$\overline{A}+\overline{B}$
1	0	$\overline{A}+B$
1	1	$\overline{A}$

S_3	S_2	S_1	S_0	$F=F_1F_2$
0	0	×	×	F_1
0	1	×	×	F_1
1	0	×	×	F_1
1	1	×	×	F_1

S_3	S_2	S_1	S_0	$F=F_1F_2$
×	×	0	0	$\overline{A}$
×	×	0	1	$\overline{A}\,\overline{B}$
×	×	1	0	$\overline{A}B$
×	×	1	1	0

3. 分析下图所示逻辑电路，列出真值表，说明其逻辑功能。

解：

$$F1=\overline{A\overline{B}C+AB\overline{C}+\overline{A}BC+\overline{B}\overline{C}}=\overline{A}\overline{B}C+\overline{A}B\overline{C}+ABC$$

真值表如下：

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

当 B≠C 时，　F1=A
当 B=C=1 时，　F1=A
当 B=C=0 时，　F1=0

F2= $\overline{\overline{A}\overline{B}+\overline{B}\overline{C}+\overline{A}\overline{C}}=AB+BC+AC$

真值表如下：

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

当 **A、B、C** 三个变量中有两个及两个以上同时为“**1**”时，**F2 = 1** 。

4. 图所示为数据总线上的一种判零电路，写出 F 的逻辑表达式，说明该电路的逻辑功能。

解：F= $\overline{\overline{A0A1A2A3}}\ \overline{\overline{A4A5A6A7}}\ \overline{\overline{A8A9A10A11}}\ \overline{\overline{A12A13A14A15}}$

只有当变量 A0~A15 全为 0 时，**F = 1**；否则，**F = 0**。

因此，电路的功能是判断变量是否全部为逻辑“0”。

5. 分析下图所示逻辑电路，列出真值表，说明其逻辑功能

解：　$F=\overline{A1}\,\overline{A0}X0+\overline{A1}A0X1+A1\overline{A0}X2+A1A0X3$

真值表如下：

A_1	A_0	F
0	0	X_0
0	1	X_1
1	0	X_2
1	1	X_3

因此，这是一个四选一的选择器。

6. 下图所示为两种十进制数代码转换器，输入为余三码，输出为什么代码？

解：

A	B	C	D	W	X	Y	Z
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

$W = AB + ACD$

$X = B\overline{C} + \overline{B}\overline{D} + BCD$

$Y = \overline{C}D + C\overline{D}$

$Z = \overline{D}$

这是一个*余三码* 至 *8421* BCD 码转换的电路

7. 下图是一个受 M 控制的 4 位二进制码和格雷码的相互转换电路。M=1 时，完成自然二进制码至格雷码转换；M=0 时，完成相反转换。请说明之

解：Y3=X3

$$Y2 = X2 \oplus X3$$

$$Y1 = X1 \oplus (MX2 + \overline{M}Y2)$$

$$Y0 = X0 \oplus (MX1 + \overline{M}Y1)$$

当 M=1 时　　Y3=X3

Y2=X2⊕X3

Y1=X1⊕X2

Y0=X0⊕X1

当 M=0 时　　Y3=X3

Y2=X2⊕X3

Y1=X1⊕Y2=X1⊕X2⊕X3

Y0=X0⊕Y1=X0⊕X1⊕X2⊕X3

M= 1 的真值表

X_3	X_2	X_1	X_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

M= 0 的真值表

X_3	X_2	X_1	X_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

由真值表可知：M=1 时，完成 8421 BCD 码到格雷码的转换；

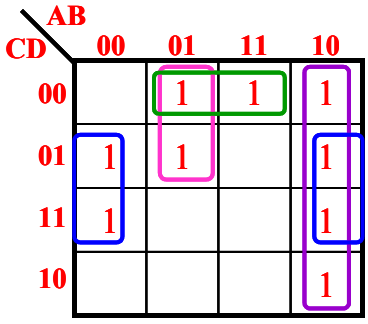
M=0 时，完成格雷码到 8421 BCD 码的转换。

8. 已知输入信号 A,B,C,D 的波形如下图所示，选择适当的集成逻辑门电路，设计产生输出 F 波形的组合电路（输入无反变量）

解：

列出真值表如下：

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



$F = A\overline{B} + \overline{B}D + B\overline{C}\,\overline{D} + \overline{A}B\overline{C}$ (或$\overline{A}\,\overline{C}D$)

9. 用红、黄、绿三个指示灯表示三台设备的工作情况：绿灯亮表示全部正常；红灯 亮表示有一台不正常；黄灯亮表示有两台不正常；红、黄灯全亮表示三台都不正常。列出控制电路真值表，并选出合适的集成电路来实现。

解：

设：三台设备分别为 A、B、C： “1”表示有故障，“0”表示无故障；红、黄、绿灯分别为 Y1、Y2、Y3：“1”表示灯亮；“0”表示灯灭。据题意列出真值表如下：

A	B	C	Y_1	Y_2	Y_3
0	0	0	0	0	1
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	1	1	0

$$Y1 = A \oplus B \oplus C$$

$$Y2 = BC + A(B \oplus C)$$

于是得：$Y3 = \overline{A}\,\overline{B}\,\overline{C} = \overline{A + B + C}$

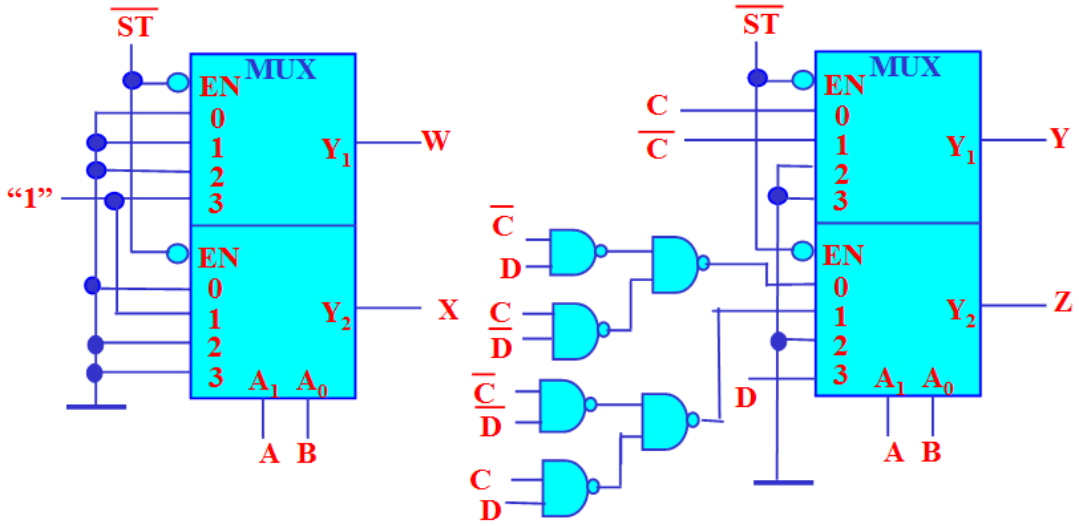
10. 用两片双四选一数据选择器和与非门实现循环码至 8421BCD 码转换。

解：(1)函数真值表、卡诺图如下；

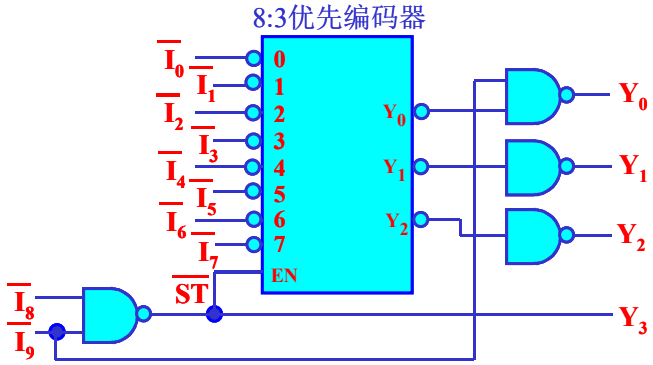
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	×	×	×	×
1	1	1	0	×	×	×	×
1	0	1	0	×	×	×	×
1	0	1	1	×	×	×	×
1	0	0	1	×	×	×	×
1	0	0	0	×	×	×	×

AB \ CD	00	01	11	10
00	0000	0001	0010	0011
01	0111	0110	0101	0100
11	1000	1001	Φ	Φ
10	Φ	Φ	Φ	Φ

（2）画逻辑图：



11. 用一片 74LS148 和与非门实现 8421BCD 优先编码器



12. 用适当门电路，设计 16 位串行加法器，要求进位链速度最快，计算一次加法时间。

解：全加器真值表如下

Ai	Bi	Ci-1	Si	Ci+1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

可以写出以下表达式

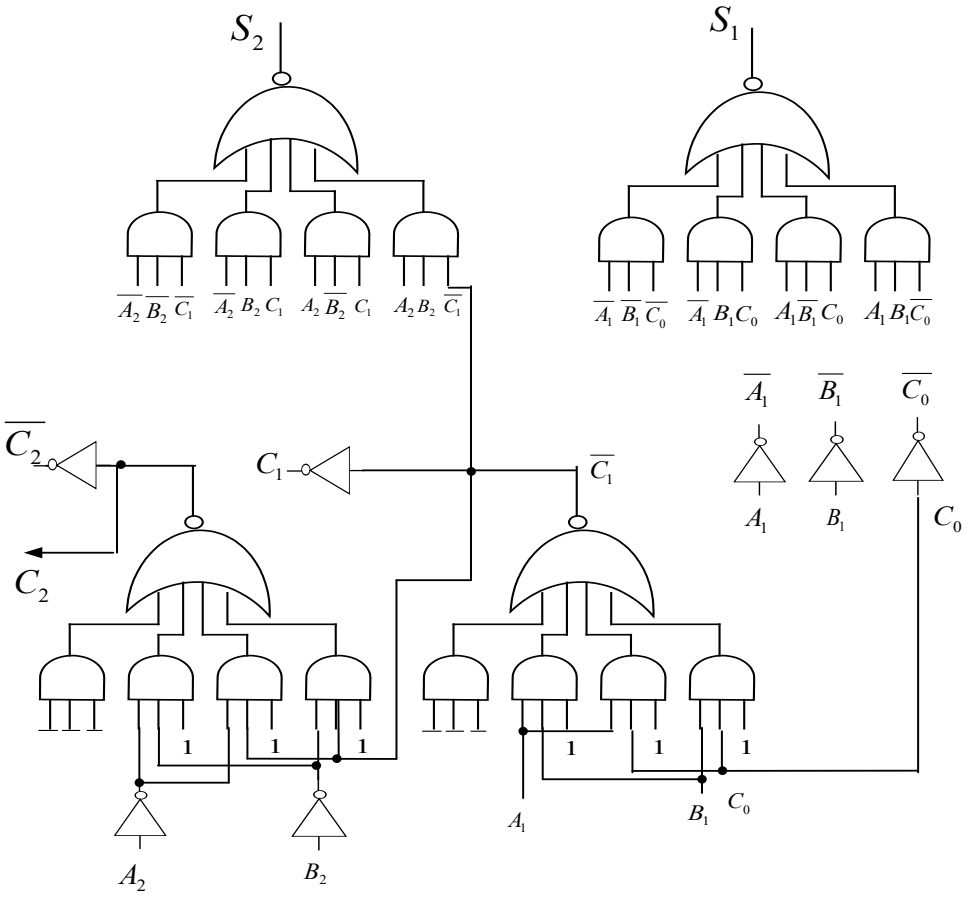
$$\overline{S} = \overline{A}\overline{B}\overline{C} + \overline{A}BC + A\overline{B}C + AB\overline{C}$$

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

$$\overline{C} = \overline{A}\overline{B} + \overline{A}\overline{C_{-1}} + \overline{B}\overline{C_{-1}}$$

要使进位链速度最快，应使用“与或非”门。具体连接图如下。
若“与或非”门延迟时间为 t1，“非门”延迟时间为 t2，则完成一次 16 位加法运算所需时间为：

$$t = (16-1)t_1 + (t_1 + t_2)$$



13. 用一片 4:16 线译码器将 8421BCD 码转换成余三码，写出表达式

解:

十进制数	8421码	余三码
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

$W(A,B,C,D)=\Sigma(5,6,7,8,9)$

$X(A,B,C,D)=\Sigma(1,2,3,4,9)$

$Y(A,B,C,D)=\Sigma(0,3,4,7,8)$

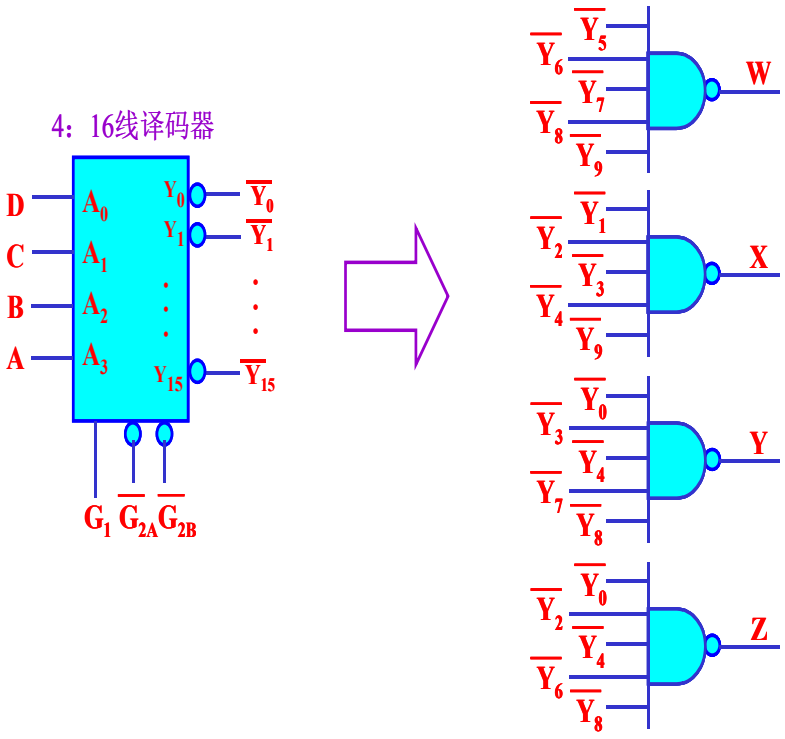
$Z(A,B,C,D)=\Sigma(0,2,4,6,8)$

$W(A,B,C,D)=\Sigma(5,6,7,8,9)=Y_5+Y_6+Y_7+Y_8+Y_9=\overline{\overline{Y}_5\,\overline{Y}_6\,\overline{Y}_7\,\overline{Y}_8\,\overline{Y}_9}$

$X(A,B,C,D)=\Sigma(1,2,3,4,9)=Y_1+Y_2+Y_3+Y_4+Y_9=\overline{\overline{Y}_1\,\overline{Y}_2\,\overline{Y}_3\,\overline{Y}_4\,\overline{Y}_9}$

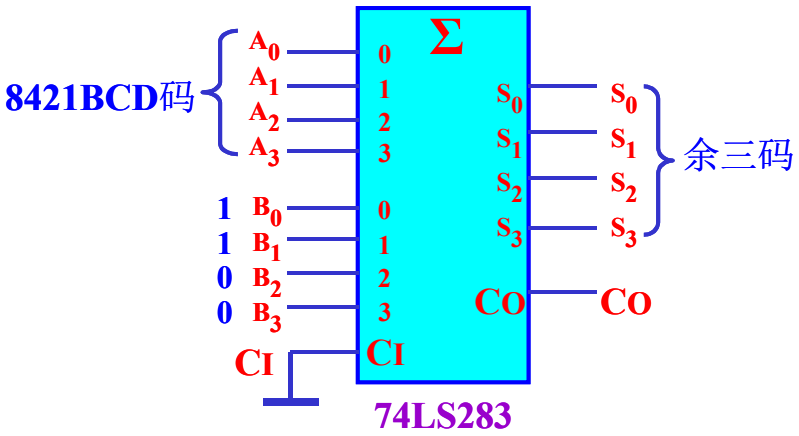
$Y(A,B,C,D)=\Sigma(0,3,4,7,8)=Y_0+Y_3+Y_4+Y_7+Y_8=\overline{\overline{Y}_0\,\overline{Y}_3\,\overline{Y}_4\,\overline{Y}_7\,\overline{Y}_8}$

$Z(A,B,C,D)=\Sigma(0,2,4,6,8)=Y_0+Y_2+Y_4+Y_6+Y_8=\overline{\overline{Y}_0\,\overline{Y}_2\,\overline{Y}_4\,\overline{Y}_6\,\overline{Y}_8}$



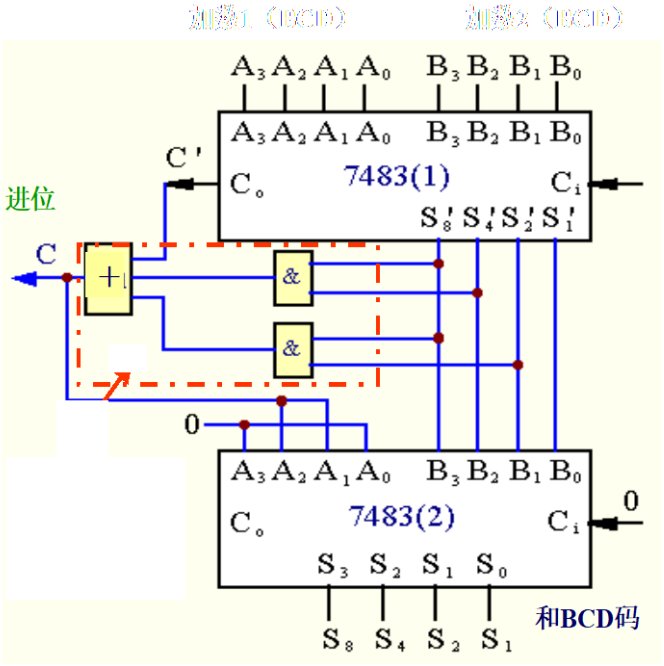
14. 使用一个 4 位二进制加法器设计 8421BCD 码转换成余三码转换器：

解：



15. 用 74LS283 加法器和逻辑门设计实现一位 8421 BCD 码加法器电路。

解：



16. 设计二进制码/格雷码转换器

解：真值表

B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

B_1B_0 \ B_3B_2	00	01	11	10
00	0000	0110	1010	1100
01	0001	0111	1011	1101
11	0010	0100	1000	1110
10	0011	0101	1001	1111

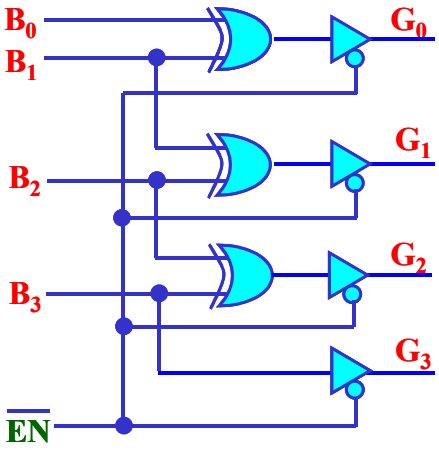
得：

$G3 = B3$

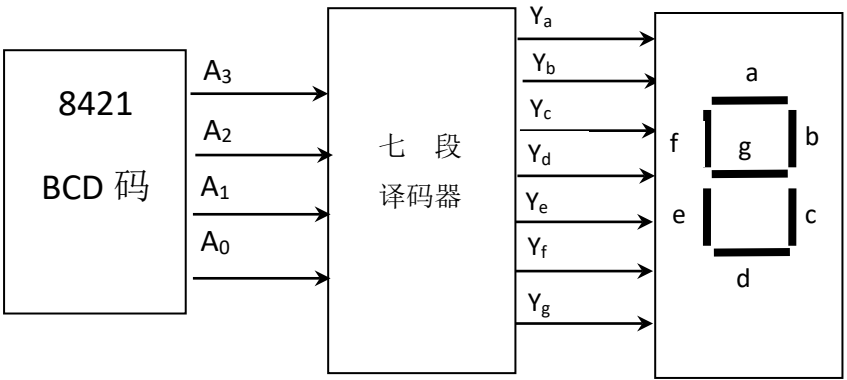
$G2 = B2 \oplus B3$

$G1 = B1 \oplus B2$

$G0 = B0 \oplus B1$



17. 设计七段译码器的内部电路，用于驱动共阴极数码管。解：七段发光二极管为共阴极电路，各段为“1”时亮。



七段译码器真值表如下：

输　入				输　　出							显示
A_3	A_2	A_1	A_0	Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

$$a = A_3 + A_1 + A_2A_0 + \overline{A}_2\overline{A}_0$$
$$b = \overline{A}_2 + \overline{A}_1\overline{A}_0 + A_1A_0$$
$$c = A_2 + \overline{A}_1 + A_0$$
$$d = \overline{A}_2\overline{A}_0 + A_1\overline{A}_0 + \overline{A}_2A_1 + A_2\overline{A}_1A_0$$
$$e = \overline{A}_2\overline{A}_0 + A_1\overline{A}_0$$
$$f = A_3 + \overline{A}_1\overline{A}_0 + A_2\overline{A}_1 + A_2\overline{A}_0$$
$$g = A_3 + A_1\overline{A}_0 + \overline{A}_2A_1 + A_2\overline{A}_1$$

18. 设计一个血型配比指示器。解：　用 XY 表示供血者代码，MN 表示受血者代码。代码设定如下：

XY = 00	A 型	MN = 00	A 型
01	B 型	01	B 型
10	AB 型	10	AB 型
11	O 型	11	O 型

X	Y	M	N	F_1(绿)	F_2(红)
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	0

得：$F_1 = \Sigma(0, 2, 5, 6, 10, 12, 13, 14, 15)$

$F2 = \overline{F1}$

19. 设计保密锁。

解： 设 A,B,C 按键按下为 1，F 为开锁信号（F=1 为打开），G 为报警信号（G=1 为报警）。

（1）真值表

A	B	C	F	G
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

（2） 卡诺图化简

F 的卡诺图：

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

化简得：$F = AB + AC$

G 的卡诺图

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

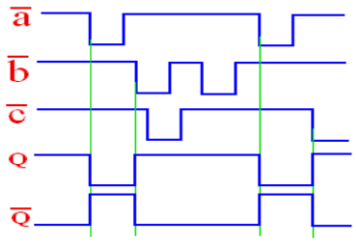
化简得：$G = \overline{A}B + \overline{A}C$

第三章　时序逻辑

1.写出触发器的次态方程，并根据已给波形画出输出 Q 的波形。

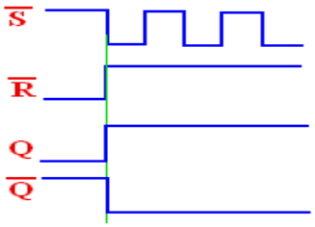
$$Q^{n+1}=(b+c)+\overline{a}Q^{n}$$

解：$\overline{a}+\overline{b}\,\overline{c}=1$

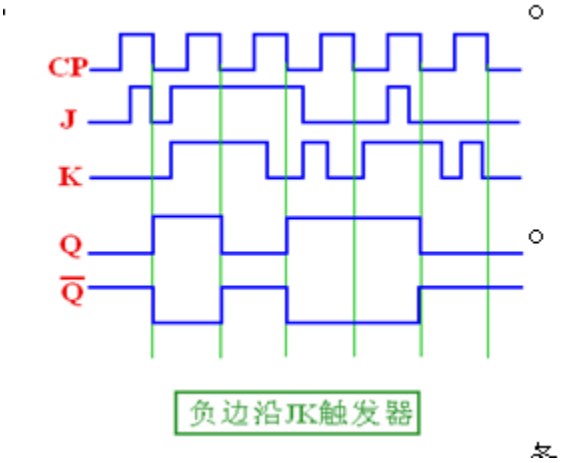


2. 说明由 RS 触发器组成的防抖动电路的工作原理，画出对应输入输出波形

解：



3. 已知 JK 信号如图，请画出负边沿 JK 触发器的输出波形（设触发器的初态为 0）



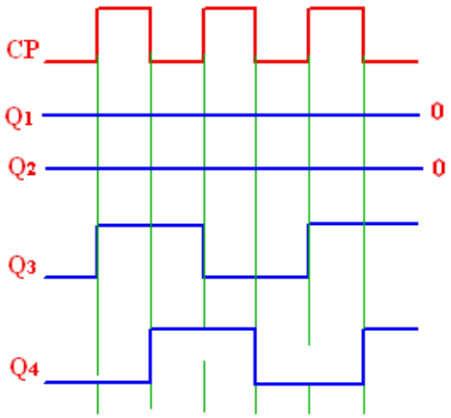
4. 写出下图所示个触发器次态方程，指出 CP 脉冲到来时，触发器置“1”的条件。

解：（1）$D = A\overline{B} + \overline{A}B$，若使触发器置“1”，则 A、B 取值相异。

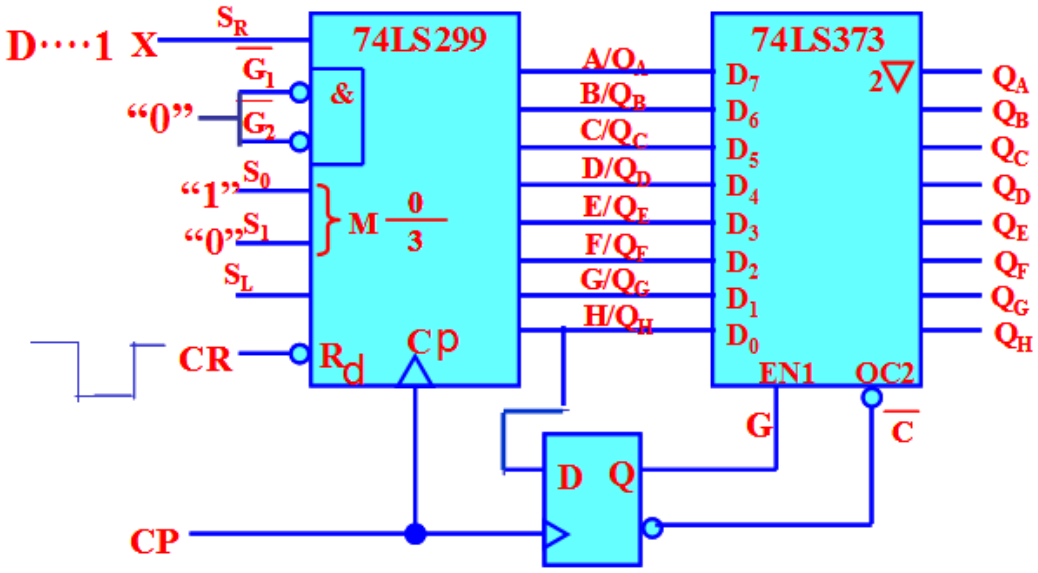
（2）$J = \overline{K} = A \oplus B \oplus C \oplus D$，若使触发器置“1”，则 A、B、C、D 取值为奇数个 1。

5. 写出各触发器的次态方程，并按所给的 CP 信号，画出各触发器的输出波形（设初态为 0）

解：



6. 设计实现 8 位数据的串行→并行转换器。



CP	QA	QB	QC	QD	QE	QF	QG	QH
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	D0	1	0	0	0	0	0	0
3	D1	D0	1	0	0	0	0	0
4	D2	D1	D0	1	0	0	0	0
5	D3	D2	D1	D0	1	0	0	0
6	D4	D3	D2	D1	D0	1	0	0
7	D5	D4	D3	D2	D1	D0	1	0
8	D6	D5	D4	D3	D2	D1	D0	1
9	D7	D6	D5	D4	D3	D2	D1	D0

7. 分析下图所示同步计数电路

解：先写出激励方程，然后求得状态方程

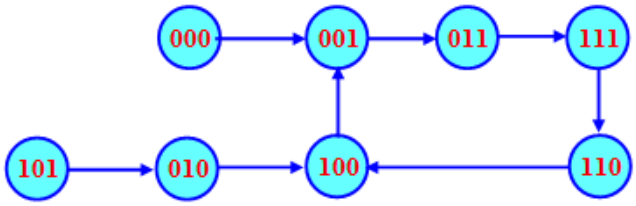
$Q_3^{n+1} = Q_2^n$

$Q_2^{n+1} = Q_1^n$

$Q_1^{n+1} = \overline{Q_2^n}\,\overline{Q_1^n} + \overline{Q_3^n}Q_1^n$

Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	1	0

状态图如下：



该计数器是循环码五进制计数器，可以自启动。

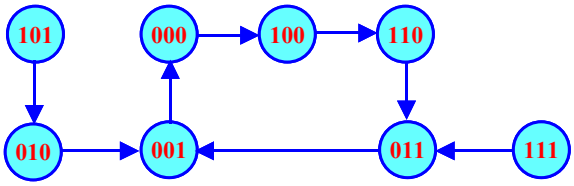
8. 作出状态转移表和状态图，确定其输出序列。

解：求得状态方程如下

$$Q_3^{n+1} = Q_2^n$$

$$Q_2^{n+1} = Q_1^n$$

$$Q_1^{n+1} = \overline{Q_2^n}\,\overline{Q_3^n}$$



故输出序列为：00011

9. 用 D 触发器构成按循环码(000→001→011→111→101→100→000)规律工作的六进制同步计数器

解：先列出真值表，然后求得激励方程

PS			NS			输出
Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	Z
0	0	0	0	0	1	0
0	0	1	0	1	1	0
0	1	1	1	1	1	0
1	1	1	1	0	1	0
1	0	1	1	0	0	0
1	0	0	0	0	0	1

化简得：

$$Z = Q_2^n \overline{Q_0^n}$$

$$Q_2^{n+1} = Q_1^n + Q_2^n Q_0^n$$

$$Q_1^{n+1} = \overline{Q_2^n} Q_0^n$$

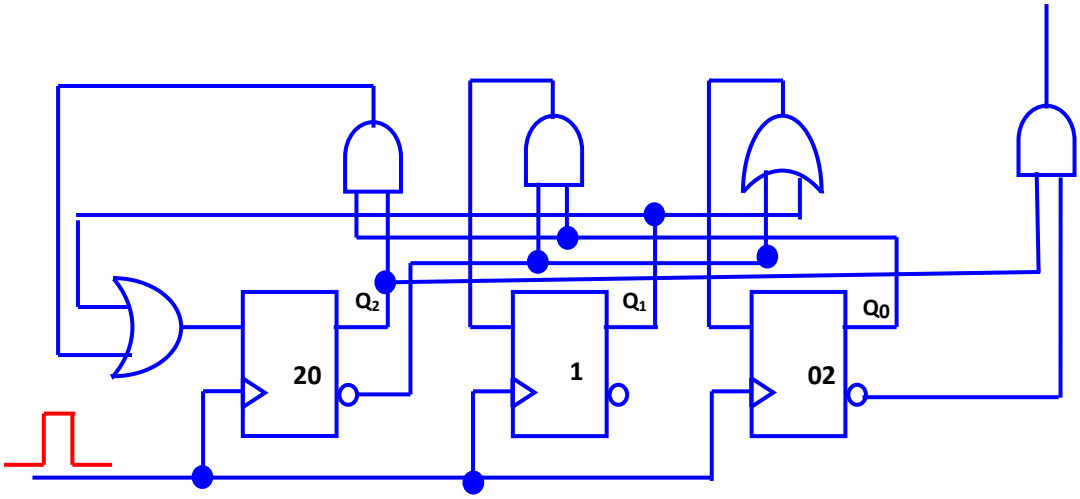
$$Q_0^{n+1} = \overline{Q_2^n} + Q_1^n$$

$$D_2 = Q_2^{n+1} = Q_1^n + Q_2^n Q_0^n$$

$$D_1 = Q_1^{n+1} = \overline{Q_2^n} Q_0^n$$

$$D_0 = Q_0^{n+1} = \overline{Q_2^n} + Q_1^n$$

逻辑电路图如下：



10. 用 D 触发器设计 3 位二进制加法计数器，并画出波形图。

解: 真值表如下

PS			NS			输出
Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	Z
0	0	0	0	0	1	0
0	0	1	0	1	0	0
0	1	0	0	1	1	0
0	1	1	1	0	0	0
1	0	0	1	0	1	0
1	0	1	1	1	0	0
1	1	0	1	1	1	0
1	1	1	0	0	0	1

化简得：

$$D_2 = Q_2\overline{Q_0} + (Q_2 \oplus Q_1)Q_0$$
$$D_1 = Q_1 \oplus Q_0$$
$$D_0 = \overline{Q_0}$$
$$Z = Q_2Q_1Q_0$$

11. 用下图所示的电路结构构成五路脉冲分配器，试分别用简与非门电路及74LS138集成译码器构成这个译码器，并画出连线图。

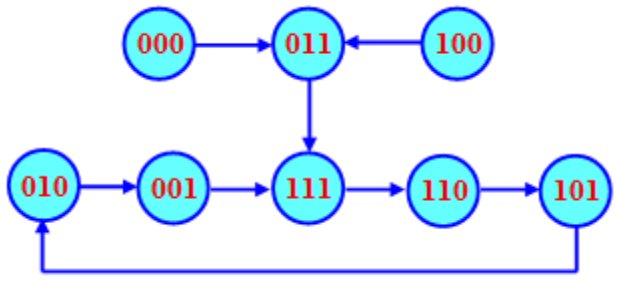
解：先写出激励方程，然后求得状态方程

$$Q_1^{n+1} = \overline{Q_1^n} + \overline{Q_3^n}Q_1^n$$
$$Q_2^{n+1} = \overline{Q_2^n} + Q_1^nQ_2^n$$
$$Q_3^{n+1} = Q_1^n\overline{Q_3^n} + Q_2^nQ_3^n$$

得真值表

Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
0	0	0	0	1	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	1	1	0

得状态图



译码器功能表

Q_3^n	Q_2^n	Q_1^n	Y_0	Y_1	Y_2	Y_3	Y_4
0	1	0	1	0	0	0	0
0	0	1	0	1	0	0	0
1	1	1	0	0	1	0	0
1	1	0	0	0	0	1	0
1	0	1	0	0	0	0	1
0	0	0	Φ				
0	1	1					
1	0	0					

若用与非门实现，译码器输出端的逻辑函数为：

$$Y_0 = \overline{Q_3}Q_2$$

$$Y_1 = \overline{Q_3}\overline{Q_2}$$

$$Y_2 = Q_2Q_1$$

$$Y_3 = Q_3\overline{Q_1}$$

$$Y_4 = Q_3\overline{Q_2}$$

若用译码器 74LS138 实现，译码器输出端的逻辑函数为：

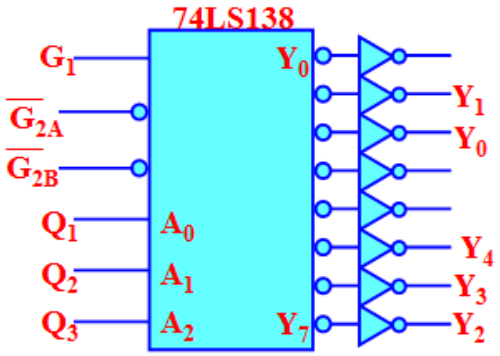
$$Y_0 = \overline{Q_3}Q_2\overline{Q_1}$$

$$Y_1 = \overline{Q_3}\overline{Q_2}Q_1$$

$$Y_2 = Q_3Q_2Q_1$$

$$Y_3 = Q_3Q_2\overline{Q_1}$$

$$Y_4 = Q_3\overline{Q_2}Q_1$$



12　若将下图接成 12 进制加法器，预置值应为多少？画出状态图及输出波形图。

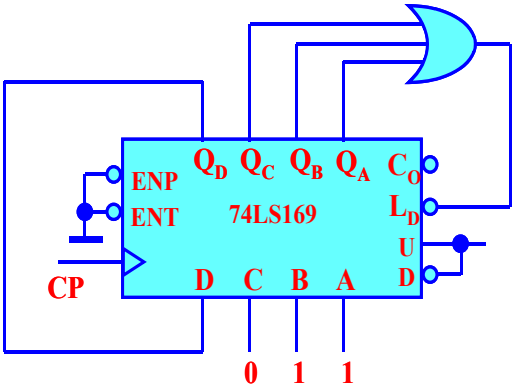
解：预置值应 C=0，B＝1，A＝1。

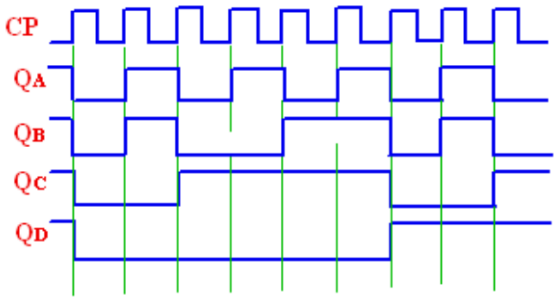
序号	Q_D Q_C Q_B Q_A
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

0000 → 0011 → 0100 → 0101 → 0110 → 0111

↑ ↓

1111 ← 1110 ← 1101 ← 1100 ← 1011 ← 1000





13. 分析下图所示同步时序逻辑电路，作出状态转移表和状态图，说明它是 Mealy 型电路还是 Moore 型电路以及电路的功能。

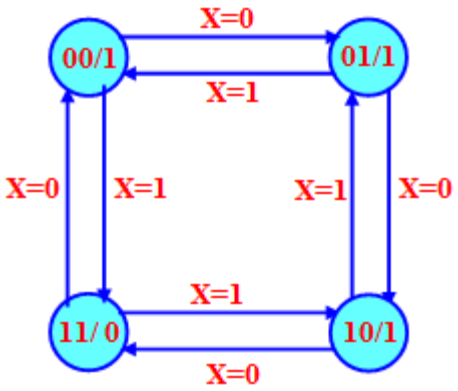
解： 电路的状态方程和输出方程为：

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

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

$$Z = \overline{Q_1^n Q_2^n}$$

$Q_2^n\ Q_1^n$	$Q_2^{n+1}\ Q_1^{n+1}$ / Z	
	X =0	X =1
0 0	01 / 1	11 / 1
0 1	10 / 1	00 / 1
1 0	10 / 1	00 / 1
1 1	00 / 0	10 / 0

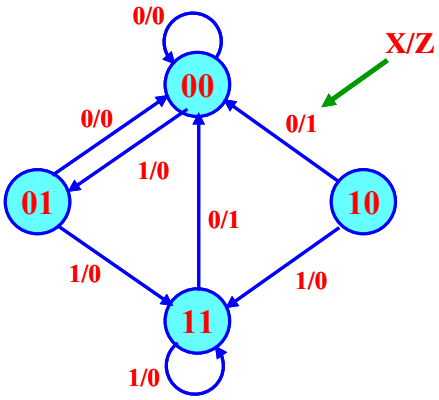


该电路是 Moore 型电路。
当 X=0 时，电路为模 4 加法计数器；
当 X=1 时，电路为模 4 减法计数器

14. 分析下图所示同步时序逻辑电路，作出状态转移表和状态图，说明这个电路能对何种序列进行检测？

解：电路的状态方程和输出方程为：

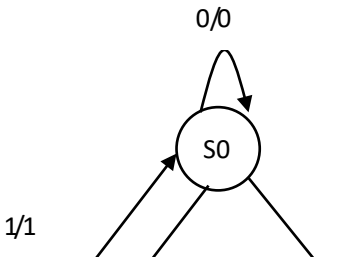
$Q_2^n\ Q_1^n$	$Q_2^{n+1}\ Q_1^{n+1}$ / Z	
	X =0	X =1
0 0	00 / 0	01 / 0
0 1	00 / 1	11 / 0
1 0	00 / 0	11 / 0
1 1	00 / 1	11 / 0



由此可见，凡输入序列 “110”,输出就为“1” 。

15. 作“101”序列信号检测器的状态表，凡收到输入序列 101 时，输出为 1 ；并规定检测的 101 序列不重叠。

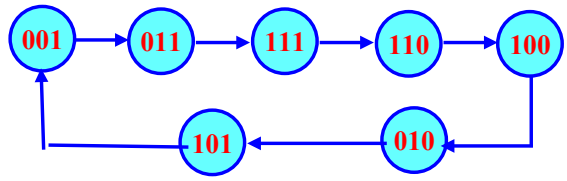
解： 根据题意分析，输入为二进制序列 x，输出为 Z；且电路应具有 3 个状态：S0、S1、S2。列状态图和状态表如下：



PS	NS / Z	
	X =0	X =1

16. 某计数器的波形如图示。

解：（1）确定计数器的状态



计数器循环中有 7 个状态。

（2）真值表如下

Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
0	0	0	Φ	Φ	Φ
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	1	1	0

（3）得状态方程、激励方程

$$Q_3^{n+1} = D_3 = Q_2^n$$

$$Q_2^{n+1} = D_2 = \overline{Q_3^n} Q_1^n + Q_2^n Q_1^n + \overline{Q_2^n}\,\overline{Q_1^n}$$

$$Q_1^{n+1} = D_1 = \overline{Q_3^n} + \overline{Q_2^n} Q_1^n$$

17. 对状态表进行编码，并做出状态转移表，用 D 触发器和与非门实现。

解：{B,F}，{D,E} 为等价状态，化简后的状态表为

PS	NS , Z	
	X =0	X =1
A	C,1	D,1
B	B,0	C,1
C	C,1	A,0
D	D,0	C,0

若状态编码 A=00，B＝01，C=10，D=11，则

$Q_2^n\ Q_1^n$	$Q_2^{n+1}\ Q_1^{n+1}/Z$	
	x =0	x =1
0 0	10/1	11/1
0 1	01/0	10/1
1 0	10/1	00/0
1 1	11/0	10/0

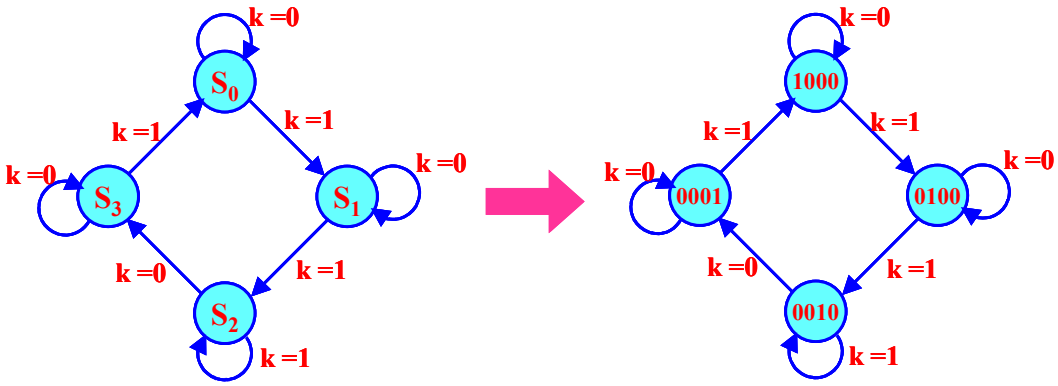
电路的状态方程和输出方程为

$$Z = \overline{Q_1^n}\,\overline{X} + \overline{Q_2^n}\,X$$

$$D_2 = Q_2^{n+1} = \overline{Q_2^n}\,\overline{Q_1^n} + Q_1^n X + Q_2^n \overline{X}$$

$$D_1 = Q_1^{n+1} = Q_1^n \overline{X} + \overline{Q_2^n}\,\overline{Q_1^n} X$$

18. 某时序机状态图如下图所示。请用“一对一法”设计其电路解：



$$D_0 = Q_0^{n+1} = Q_0^n \overline{K} + Q_3^n K$$

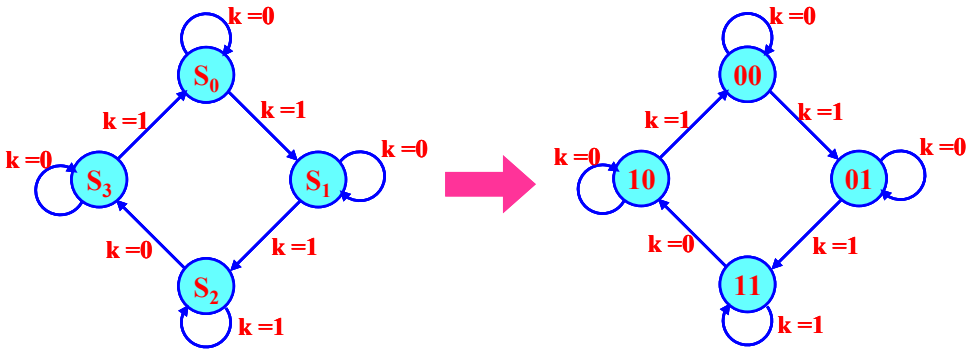
$$D_1 = Q_1^{n+1} = Q_0^n K + Q_1^n \overline{K}$$

$$D_2 = Q_2^{n+1} = Q_2^n K + Q_1^n K$$

$$D_3 = Q_3^{n+1} = Q_2^n \overline{K} + Q_3^n \overline{K}$$

19. 某时序机状态图如下所示，用“计数器法”设计该电路

解：



若编码为： S0=00 S1=01 S2=11 S3=10：

则

$Q_1^n Q_2^n$	$Q_1^{n+1} Q_2^{n+1}$	
	k =0	k =1
0 0	00	01
0 1	01	11
1 1	10	11
1 0	10	00

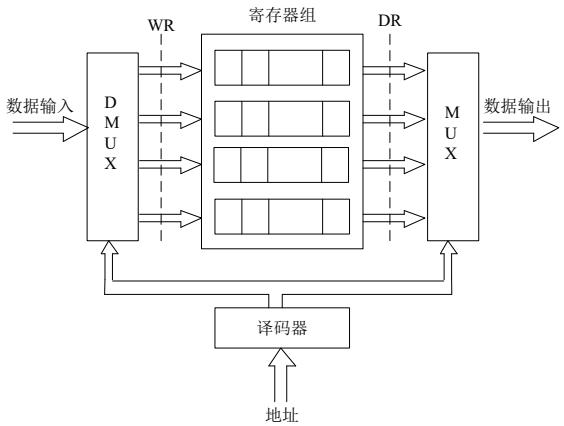
$$Q_1^{n+1} = \overline{K} Q_1^n + K Q_2^n$$

次态方程为： $$Q_2^{n+1} = K\overline{Q_1^n} + K Q_2^n + \overline{Q_1^n} Q_2^n$$

第四章　习题答案

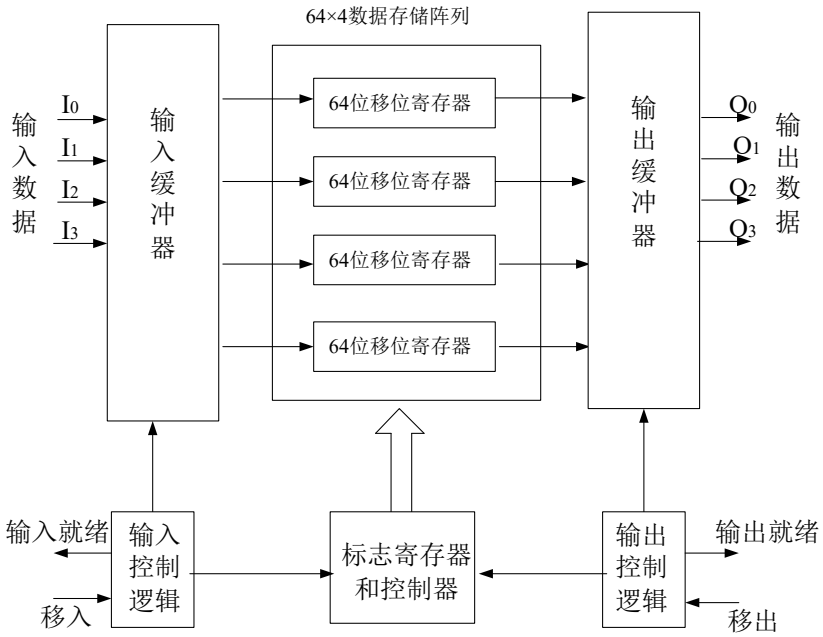
1. 设计 4 个寄存器堆。

解：



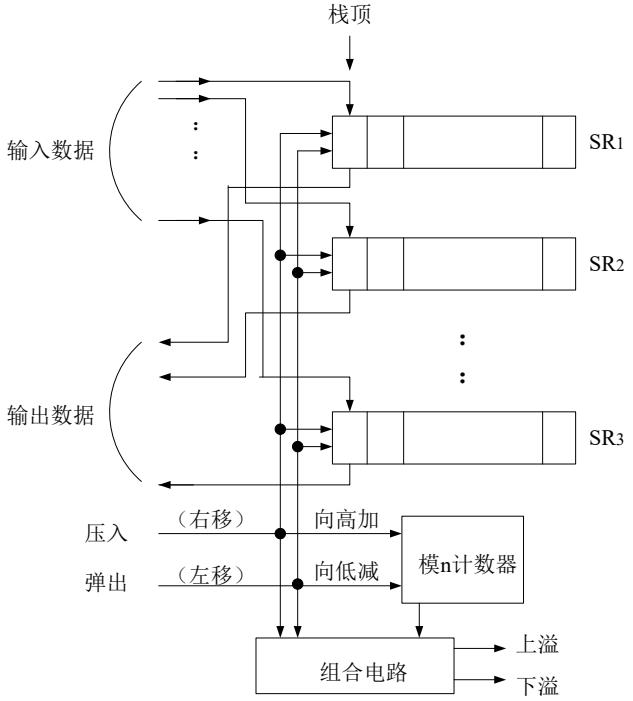
2. 设计具有 4 个寄存器的队列。

解：



3．设计具有 4 个寄存器的堆栈

解：可用具有左移、右移的移位寄存器构成堆栈。



4．SRAM、DRAM 的区别

解：DRAM 表示动态随机存取存储器，其基本存储单元是一个晶体管和一个电容器，是一种以电荷形式进行存储的半导体存储器，充满电荷的电容器代表逻辑“1”，“空”的电容器代表逻辑“0”。数据存储在电容器中，电容存储的电荷一般是会慢慢泄漏的，因此内存需要不时地刷新。电容需要电流进行充电，而电流充电的过程也是需要一定时间的，一般是 0.2-0.18 微秒（由于内存工作环境所限制，不可能无限制的提高电流的强度），在这个充电的过程中内存是不能被访问的。DRAM 拥有更高的密度，常常用于 PC 中的主存储器。

SRAM 是静态的，存储单元由 4 个晶体管和两个电阻器构成，只要供电它就会保持一个值，没有刷新周期，因此 SRAM 比 DRAM 要快。SRAM 常常用于高速缓冲存储器，因为它有更高的速率；

5. 为什么 DRAM 采用行选通和列选通

解：DRAM 存储器读/写周期时，在行选通信号 RAS 有效下输入行地址，在列选通信号 CAS 有效下输入列地址。如果是读周期，此位组内容被读出；如果是写周期，将总线上数据写入此位组。由于 DRAM 需要不断刷新，最常用的是“只有行地址有效”的方法，按照这种方法，刷新时，是在 RAS 有效下输入刷新地址，存储体的列地址无效，一次选中存储体中的一行进行刷新。每当一个行地址信号 RAS 有效选中某一行时，该行的所有存储体单元进行刷新。

6. 用 ROM 实现二进制码到余 3 码转换

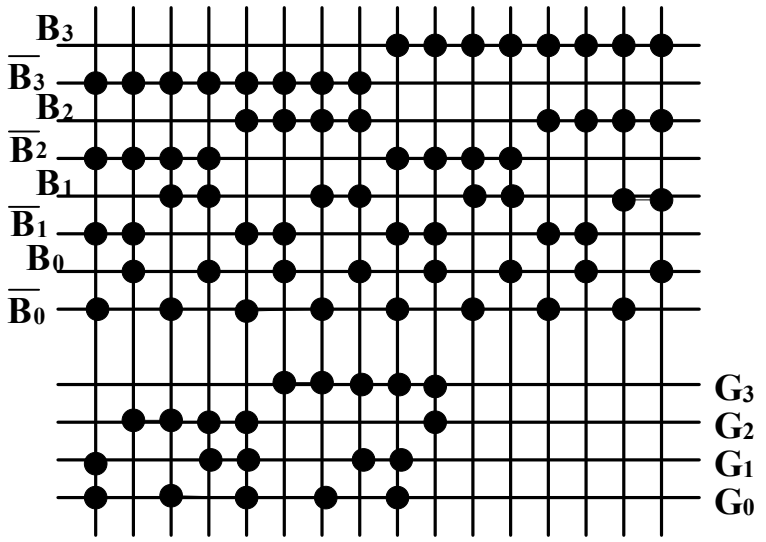
解： 真值表如下：

8421 码				余三码			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

最小项表达式为：

$G_3=\sum(5,6,7,8,9)$ $G_2=\sum(1,2,3,4,9)$ $G_1=\sum(0,3,4,7,8)$ $G_0=\sum(0,2,4,6,8)$

阵列图为：



7. 用 ROM 实现 8 位二进制码到 8421 码转换

解：输入为 8 位二进制数，输出为 3 位 BCD 码，12 位二进制数，所以，所需 ROM 的容量为：2^8*12=3072

8.ROM、EPROM 和 EEPROM 的区别

解：ROM 指的是“只读存储器”，即 Read-Only Memory。这是一种线路最简单半导体电路，通过掩模工艺， 一次性制造，其中的代码与数据将永久保存(除非坏掉)，不能进行修改。

EPROM 指的是“可擦写可编程只读存储器”，即 Erasable Programmable Read-Only Memory。是采用浮栅技术生产的可编程存储器，它的存储单元多采用 N 沟道叠栅 MOS 管,信息的存储是通过 MOS 管浮栅上的电荷分布来决定的，编程过程就是一个电荷注入过程。编程结束后，由于绝缘层的包围，注入到浮栅上的电荷无法泄漏，因此电荷分布维持不变，EPROM 也就成为非易失性存储器件了。当外部能源（如紫外线光源）加到 EPROM 上时，EPROM 内部的电荷分布才会被破坏，此时聚集在 MOS 管浮栅上的电荷在紫外线照射下形成光电流被泄漏掉，使电路恢复到初始状态，从而擦除了所有写入的信息。这样 EPROM 又可以写入新的信息。

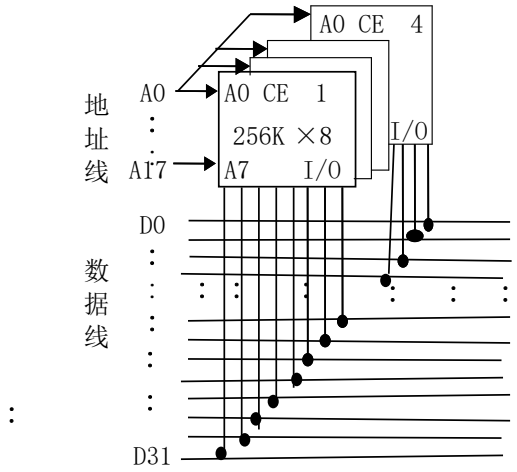
EEPROM 指的是“电可擦除可编程只读存储器”，即 Electrically Erasable Programmable Read-Only Memory。也是采用浮栅技术生产的可编程 ROM，但是构成其存储单元的是隧道 MOS 管，隧道 MOS 管也是利用浮栅是否存有电荷来存储二值数据的，不同的是隧道 MOS 管是用电擦除的，并且擦除的速度要快的多（一般为毫秒数量级）。它的最大优点是可直接用电信号擦除，也可用电信号写入。E^2PROM 的电擦除过程就是改写过程，它具有 ROM 的非易失性，又具备类似 RAM 的功能，可以随时改写（可重复擦写 1 万次以上）。目前，大多数 E^2PROM 芯片内部都备有升压电路。因此，只需提供单电源供电，便可进行读、擦除/写操作，这为数字系统的设计和在线调试提供了极大方便。

9. flash 存储器的特点

解： Flash 也是一种非易失性的内存，属于 EEPROM 的改进产品。FLASH 是结合 EPROM 和 EEPROM 技术达到的，FLASH 使用雪崩热电子注入方式来编程。主要特点是，FLASH 对芯片提供大块或整块的擦除，而 EEPROM 则可以一次只擦除一个字节(Byte)。这就降低了设计的复杂性，它可以不要 EEPROM 单元里多余的晶体管，所以可以做到高集成度，大容量，另 FLASH 的浮栅工艺上也不同，写入速度更快。

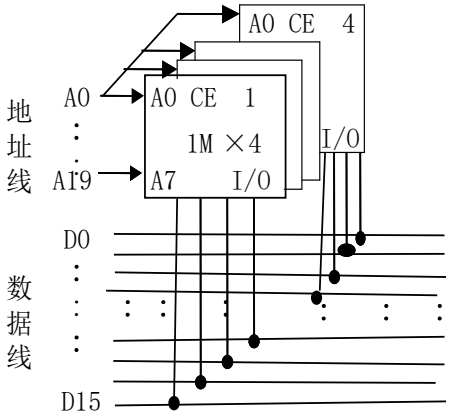
10. 用 256K×8 芯片实现 256K×32 的 ROM

解：需要 4 片 256K×8 的存储器，进行位扩展。



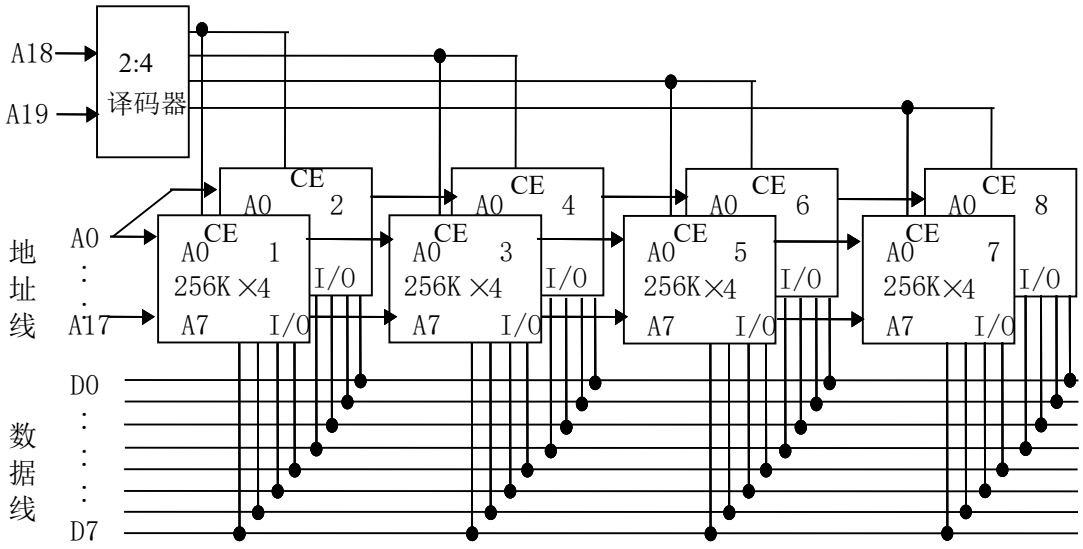
11. 用 1M×4 芯片实现 1M×16 的 SRAM

解：需要 4 片 1M×4 的存储器，进行位扩展。



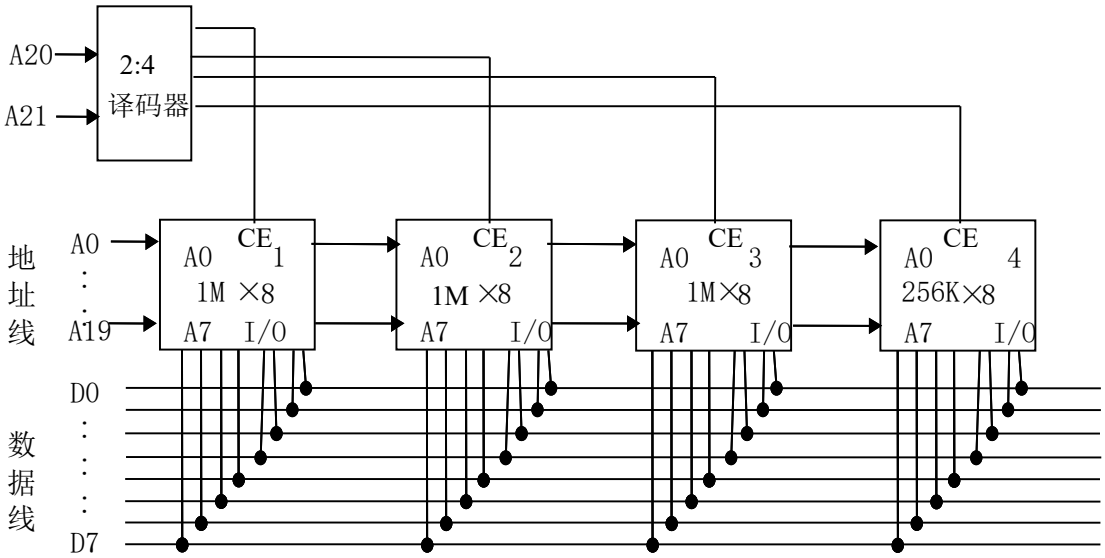
12 用 256K×4 芯片实现 1M×8 的 DRAM

解：需 8 片 1M×4 的存储器，进行字位同时扩展。



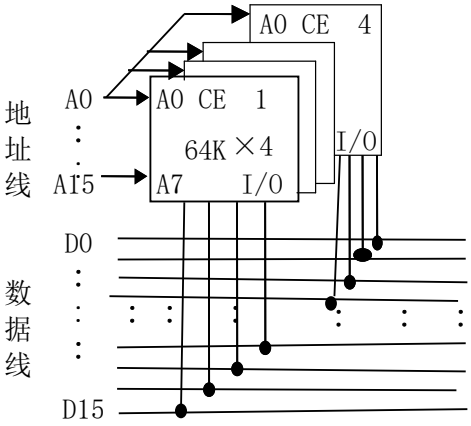
13．用 1M×8 芯片实现 4M×8 的 DRAM

解：需 4 片 1M×8 的存储器，进行字扩展。



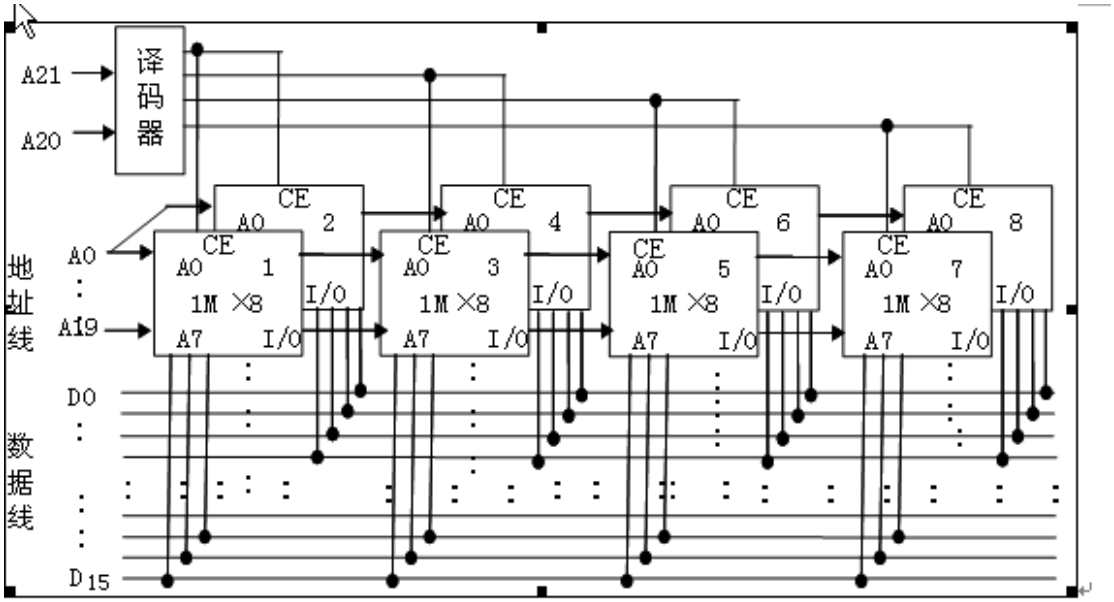
14．用 64K×4 芯片实现 64K×16 的 ROM

解：需 4 片 64K×4 的存储器，进行位扩展。



15．用 1M×8 芯片实现 4M×16 的 ROM

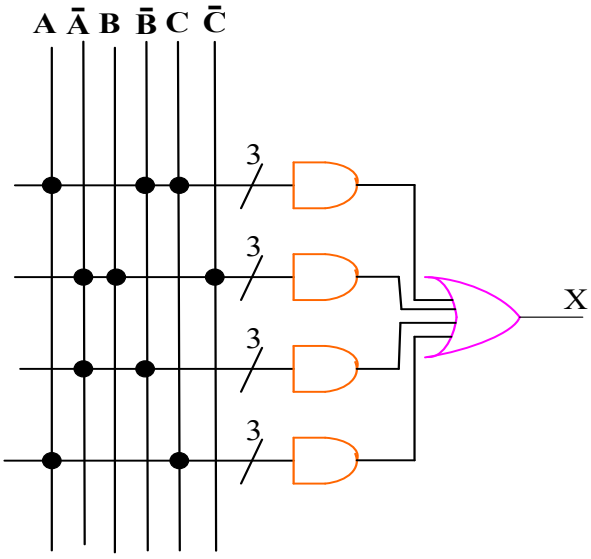
解：需 8 片 1M×8 的存储器，进行字位同时扩展。



第五章　习题答案

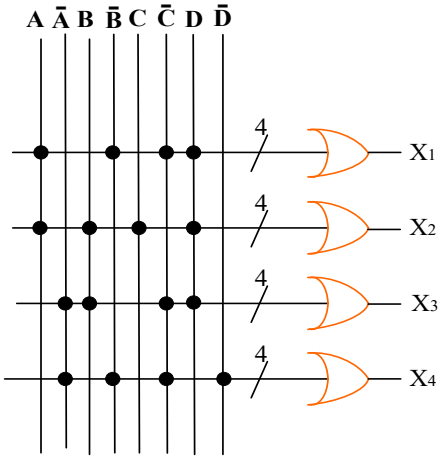
1. 画出与阵列编程点

解：



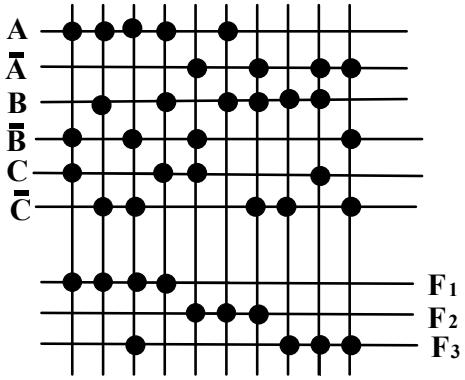
2. 画出或阵列编程点

解：



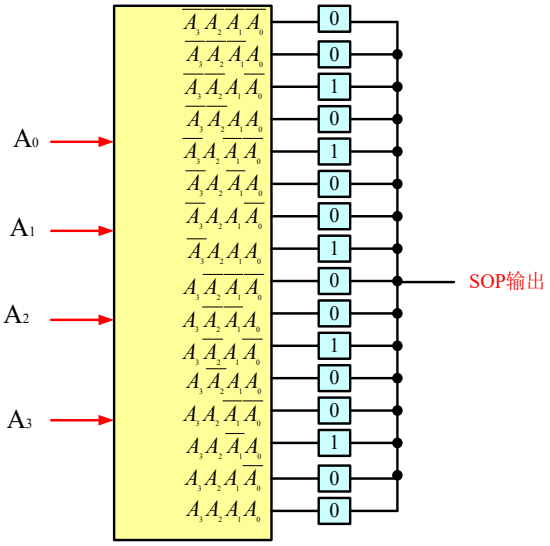
3. 与、或阵列均可编程，画出编程点。

解;



4. 4 变量 LUT 编程

解：



5. 用 VHDL 写出 4 输入与门

解： 源代码：

```
LIBRARY  IEEE;
USE  IEEE. STD_LOGIC_1164. ALL;
```

```
ENTITY  and4  IS
    PORT (a，b，c，d：IN STD_LOGIC;
                      x：OUT STD_LOGIC）;
END and4;

ARCHITECTURE and4_arc OF and4 IS
    BEGIN
    x＜＝a AND b AND c AND d;
END and4_arc;
```

6. 用 VHDL 写出 4 输入或门

解：　源代码：

```
LIBRARY  IEEE;
USE  IEEE. STD_LOGIC_1164. ALL;

ENTITY  or4  IS
    PORT (a，b，c，d：IN STD_LOGIC;
                      x：OUT STD_LOGIC）;
END or4;

ARCHITECTURE or4_arc OF or4 IS
    BEGIN
    x＜＝a OR b OR c OR d;
END or4_arc;
```

7. 用 VHDL 写出 SOP 表达式

解：　源代码：

```
LIBRARY  IEEE;
USE  IEEE. STD_LOGIC_1164. ALL;

ENTITY  sop  IS
    PORT (a，b，c，d，e，f：IN STD_LOGIC;
          x：OUT STD_LOGIC）;
END sop;

ARCHITECTURE sop_arc OF sop IS
    BEGIN
    x＜＝(a AND b) OR (c AND d) OR (e AND f);
END sop_arc;
```

8. 用 VHDL 写出布尔表达式

解：　源代码：

```
LIBRARY  IEEE;
USE  IEEE. STD_LOGIC_1164. ALL;

ENTITY  boolean  IS
```

```
        PORT (a，b，c：IN STD_LOGIC；
                    f：OUT STD_LOGIC）；
    END boolean ；

    ARCHITECTURE boolean_arc OF boolean IS
        BEGIN
        f＜＝(a OR (NOT b) OR c) AND (a OR b OR (NOT c)) AND
((NOT a) OR (NOT b) OR (NOT c))；
    END boolean_arc；
```

9. 用 VHDL 结构法写出 SOP 表达式

解： 源代码：

```
——三输入与非门的逻辑描述
    LIBRARY  IEEE；
    USE  IEEE. STD_LOGIC_1164. ALL；

    ENTITY  nand3  IS
      PORT (a，b，c：IN STD_LOGIC；
                x：OUT STD_LOGIC）；
    END nand3；

    ARCHITECTURE nand3_arc OF nand3 IS
        BEGIN
        x＜＝NOT (a AND b AND c)；
    END nand3_arc；

——顶层结构描述文件
    LIBRARY  IEEE；
    USE  IEEE. STD_LOGIC_1164. ALL；

    ENTITY  sop  IS
        PORT (in1，in2，in3，in4，in5，in6，in7，in8，in9：IN STD_LOGIC；
                    out4：OUT STD_LOGIC)；
    END sop；

    ARCHITECTURE sop_arc OF sop IS
        COMPONENT nand3
            PORT (a，b，c：IN STD_LOGIC；
                        x：OUT STD_LOGIC）；
        END COMPONENT；
        SIGNAL  out1，out2，out3：STD_LOGIC；
    BEGIN
      u1：nand3 PORT MAP (in1，in2，in3，out1)；
      u2：nand3 PORT MAP (in4，in5，in6，out2)；
      u3：nand3 PORT MAP (in7，in8，in9，out3)；
      u4：nand3 PORT MAP (out1，out2，out3，out4)；
```

```
END sop;
```

10. 用 VHDL 数据流法写出 SOP 表达式

解：　源代码：

```
LIBRARY　IEEE;
USE　IEEE. STD_LOGIC_1164. ALL;

ENTITY　sop　IS
    PORT (in1，in2，in3，in4，in5，in6，in7，in8，in9：IN STD_LOGIC;
                out4：OUT STD_LOGIC);
END sop;

ARCHITECTURE sop_arc OF sop IS
    BEGIN
    out4<=(in1 AND in2 AND in3) OR (in4 AND in5 AND in6 ) OR
(in7 AND in8 AND in9);
END　sop_arc;
```

13. 用 VHDL 设计 3－8 译码器

解：　源代码：

```
LIBRARY　IEEE;
USE　IEEE. STD_LOGIC_1164. ALL;

ENTITY　decoder_3_to_8　IS
    PORT　（a，b，c，g1，g2a，g2b：IN STD_LOGIC;
            y：OUT　STD_LOGIC _VECTOR（7 downto 0））;
END　decoder_3_to_8;

ARCHITECTURE　rtl　OF　decoder_3_to_8　IS
    SIGNAL　indata：STD_LOGIC _VECTOR（2 downto 0）;
BEGIN
    indata<=c & b & a;
    PROCESS（indata，g1，g2a，g2b）
        BEGIN
            IF（g1='1' AND g2a='0' AND g2b='0'）THEN
                CASE　indata　IS
                    WHEN "000"=>y<="11111110";
                    WHEN "001"=>y<="11111101";
                    WHEN "010"=>y<="11111011";
                    WHEN "011"=>y<="11110111";
                    WHEN "100"=>y<="11101111";
                    WHEN "101"=>y<="11011111";
                    WHEN "110"=>y<="10111111";
                    WHEN others=>y<="01111111";
                END CASE;
            ELSE
```

```
                    y<="11111111";
                END IF;
            END PROCESS;
        END   rt1;
```

14. 用 VHDL 设计七段显示译码器

解：　源代码：

```
        LIBRARY   IEEE;
        USE   IEEE. STD_LOGIC_1164. ALL;

    ENTITY   segment7   IS
        PORT   (xin：IN STD_LOGIC _VECTOR（3 downto 0）;
                lt，rbi：IN STD_LOGIC;
                yout：OUT STD_LOGIC _VECTOR（6 downto 0）;
                birbo：INOUT STD_LOGIC）;
    END   segment7;

    ARCHITECTURE   seg7448   OF   segment7   IS
         SIGNAL   sig_xin：  STD_LOGIC _VECTOR（3 downto 0）;
    BEGIN
        sig_xin<=xin;
        PROCESS（sig_xin，lt，rbi，birbo）
            BEGIN
                IF（birbo='0'）THEN
                        yout<="0000000";
                ELSIF （lt='0'）THEN
                        yout<="1111111";
                        birbo<='1';
                ELSIF （rbi='0' AND sig_xin="0000"）THEN
                        yout<="0000000";
                        birbo<='0';
                ELSIF （rbi='1' AND sig_xin="0000"）THEN
                        yout<="1111110";
                        birbo<='1';
                ELSE
                        birbo<='1';
                    CASE   sig_xin   IS
                        WHEN "0001"=>yout<="0110000";
                        WHEN "0010"=>yout<="1101101";
                        WHEN "0011"=>yout<="1111001";
                        WHEN "0100"=>yout<="0110011";
                        WHEN "0101"=>yout<="1011011";
                        WHEN "0110"=>yout<="0011111";
                        WHEN "0111"=>yout<="1110000";
                        WHEN "1000"=>yout<="1111111";
```

```
                    WHEN "1001"=>yout<="1110011";
                    WHEN others=>yout<="0100011";
                END CASE;
            END IF;
        END PROCESS;
    END   seg7448;
```

15. 用 VHDL 设计 8/3 优先编码器

解：　源代码：

```
    LIBRARY   IEEE;
    USE   IEEE. STD_LOGIC_1164. ALL;

ENTITY   priorityencoder   IS
    PORT  （din：IN STD_LOGIC _VECTOR（7 downto 0）;
           ei：IN STD_LOGIC;
           yout：OUT STD_LOGIC _VECTOR（2 downto 0）;
           eo，gs：OUT STD_LOGIC）;
END   priorityencoder;

ARCHITECTURE   cod74148   OF   priorityencoder   IS
BEGIN
    PROCESS（ei，din）
      BEGIN
          IF（ei='1'）THEN
             yout<="111";
             eo<='1';
             gs<='1';
          ELSE
              IF   (din(7)='0')   THEN
                   yout<="000";
                   eo<='1';
                   gs<='0';
              ELSIF   (din(6)='0')   THEN
                   yout  <="001";
                   eo<='1';
                   gs<='0';
              ELSIF   (din(5)='0')   THEN
                   yout<="010";
                   eo<='1';
                   gs<='0';
              ELSIF   (din(4)='0')   THEN
                   yout<="011";
                   eo<='1';
                   gs<='0';
              ELSIF   (din(3)='0')   THEN
```

```
            yout<="100";
            eo<='1';
            gs<='0';
        ELSIF   (din(2)='0')   THEN
            yout<="101";
            eo<='1';
            gs<='0';
        ELSIF   (din(1)='0')   THEN
            yout<="110";
            eo<='1';
            gs<='0';
        ELSIF   (din(0)='0')   THEN
            yout<="111";
            eo<='1';
            gs<='0';
        ELSIF   (din="11111111")   THEN
            yout<="111";
            eo<='0';
            gs<='1';
        END IF;
    END IF;
  END PROCESS;
END   cod74148;
```

16. 用 VHDL 设计 BCD 码至二进制码转换器。

解：　源代码：

```
library ieee;
use ieee.std_logic_1164.all;

entity bcdtobi is
port(
      bcdcode : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
      start: in std_logic;
      qbit :  OUT STD_LOGIC_VECTOR(3 DOWNTO 0)
);

end;

architecture behavioral of bcdtobi is
begin

process(start, bcdcode)
   begin
       if start='0' then
```

```
            case bcdcode(7 downto 0) is
                when "00000000"=>qbit(3 downto 0)<="0000";
                when "00000001"=>qbit(3 downto 0)<="0001";
                when "00000010"=>qbit(3 downto 0)<="0010";
                when "00000011"=>qbit(3 downto 0)<="0011";
                when "00000100"=>qbit(3 downto 0)<="0100";
                when "00000101"=>qbit(3 downto 0)<="0101";
                when "00000110"=>qbit(3 downto 0)<="0110";
                when "00000111"=>qbit(3 downto 0)<="0111";
                when "00001000"=>qbit(3 downto 0)<="1000";
                when "00001001"=>qbit(3 downto 0)<="1001";
                when "00010000"=>qbit(3 downto 0)<="1010";
                when "00010001"=>qbit(3 downto 0)<="1011";
                when "00010010"=>qbit(3 downto 0)<="1100";
                when "00010011"=>qbit(3 downto 0)<="1101";
                when "00010100"=>qbit(3 downto 0)<="1110";
                when "00010101"=>qbit(3 downto 0)<="1111";
                when others=>qbit(3 downto 0)<="0000";
            end case;
        else
            qbit(3 downto 0)<="0000";
        end if;
    end process;
end behavioral;
```

17. 用 VHDL 设计 4 位寄存器

解：　异步复位

源代码：

```
LIBRARY　IEEE;
USE　IEEE. STD_LOGIC_1164. ALL;

ENTITY　register_4 IS
    PORT　（clk，r：IN STD_LOGIC;
            din：IN STD_LOGIC _VECTOR（3 downto 0）;
            qout：OUT STD_LOGIC _VECTOR（3 downto 0））;
END　register_4;

ARCHITECTURE　rge_arc　OF　register_4　IS
  SIGNAL　q_temp：STD_LOGIC _VECTOR（3 downto 0）;
BEGIN
    PROCESS（clk，r）
        BEGIN
            IF（r='1'）THEN
                q_temp<＝"0000";
```

```
            ELSIF  (clk'event  AND  clk='1')  THEN
                q_temp<=din;
            END IF;
        qout<=q_temp;
        END PROCESS;
    END  rge_arc;
```

18. 用 VHDL 设计 4 位双向移位寄存器

解：　s1、s0 控制工作方式，dsl 为左移数据输入，dsr 为右移数据输入。

源代码：

```
LIBRARY  IEEE;
USE  IEEE. STD_LOGIC_1164. ALL;

ENTITY  shiftreg IS
   PORT  (clk，r，dsr，dsl：IN STD_LOGIC;
          s1，s0：IN STD_LOGIC；--function select
          din：IN STD_LOGIC _VECTOR（3 downto 0）；--data in
          qout：OUT STD_LOGIC _VECTOR（3 downto 0））；--data out
END  shiftreg;

ARCHITECTURE  ls74194  OF  shiftreg  IS
   SIGNAL  iq：  STD_LOGIC _VECTOR（3 downto 0）;
   SIGNAL  s：STD_LOGIC _VECTOR（1 downto 0）;
BEGIN
    s<=s1 & s0;
    PROCESS（clk，r）
        BEGIN
          IF（r='0'）THEN
              iq<="0000";
          ELSIF  (clk'event  AND  clk='1')  THEN
              CASE  s  IS
                  WHEN "00"=>null;
                  WHEN "01"=>iq<=dsr & din（3 downto 1）；--right
                  WHEN "10"=>iq<=din（2 downto 0）& dsl；--left
                  WHEN "11"=>iq<=din；--load
                  WHEN others=>null;
              END CASE;
          END IF;
    END PROCESS;
    qout<=iq;
END  ls74194;
```

19. 用 VHDL 设计 8421 码十进制加法计数器

解：　异步清零，同步置数

源代码：

```
LIBRARY  IEEE;
USE  IEEE. STD_LOGIC_1164. ALL;
USE  IEEE. STD_LOGIC_ARITH. ALL;
USE  IEEE. STD_LOGIC_UNSIGNED. ALL;

ENTITY  count10  IS
   PORT  (clk，clr，load：IN STD_LOGIC;
          din：IN STD_LOGIC _VECTOR（3 downto 0）;
          co：OUT STD_LOGIC;
          qout：OUT STD_LOGIC _VECTOR（3 downto 0））;
END  count10;

ARCHITECTURE  count10_arch  OF  count10  IS
   SIGNAL  iq：  STD_LOGIC _VECTOR（3 downto 0）;
  BEGIN
  PROCESS（clr，clk，load）
  BEGIN
          IF（clr＝'0'）THEN
              iq＜＝"0000";
          ELSIF  (clk'event  AND  clk＝'1')  THEN
              IF（load＝'0'）THEN
                  iq＜＝din;
               ELSIF（iq＝9）THEN
                    iq＜＝"0000";
              ELSE
                    iq＜＝iq+1;
             END IF;
          END IF;
     END PROCESS;
     qout＜＝iq;
     co＜＝'1' WHEN iq＝"1001" ELSE
          '0';
END  count10_arch;
```

20．用 VHDL 设计可逆格雷码计数器

解： 源代码：

```
LIBRARY  IEEE;
USE  IEEE. STD_LOGIC_1164. ALL;

ENTITY  gray_count  IS
   PORT  (clk，y：IN STD_LOGIC;
          qout：OUT STD_LOGIC _VECTOR（2 downto 0））;
END  gray_count;

ARCHITECTURE  arch_gray  OF  gray_count  IS
   SIGNAL  iq：  STD_LOGIC _VECTOR（2 downto 0）;
```

```
BEGIN
      PROCESS（clk）
          BEGIN
              IF  (clk'event  AND  clk＝'1')  THEN
                  IF（y＝'1'）THEN
                      CASE  iq  IS
                          WHEN  "000"＝＞iq＜＝"001";
                          WHEN  "001"＝＞iq＜＝"011";
                          WHEN  "011"＝＞iq＜＝"010";
                          WHEN  "010"＝＞iq＜＝"110";
                          WHEN  "110"＝＞iq＜＝"111";
                          WHEN  "111"＝＞iq＜＝"101";
                          WHEN  "101"＝＞iq＜＝"100";
                          WHEN  others＝＞iq＜＝"000";
                      END CASE;
                  END IF;
                  IF（y＝'0'）THEN
                      CASE  iq  IS
                          WHEN  "000"＝＞iq＜＝"100";
                          WHEN  "100"＝＞iq＜＝"101";
                          WHEN  "101"＝＞iq＜＝"111";
                          WHEN  "111"＝＞iq＜＝"110";
                          WHEN  "110"＝＞iq＜＝"010";
                          WHEN  "010"＝＞iq＜＝"011";
                          WHEN  "011"＝＞iq＜＝"001";
                          WHEN  others＝＞iq＜＝"000";
                      END CASE;
                  END IF;
              END IF;
      END PROCESS;
      qout＜＝iq;
END   arch_gray;
```

21. 用 VHDL 设计有限状态机

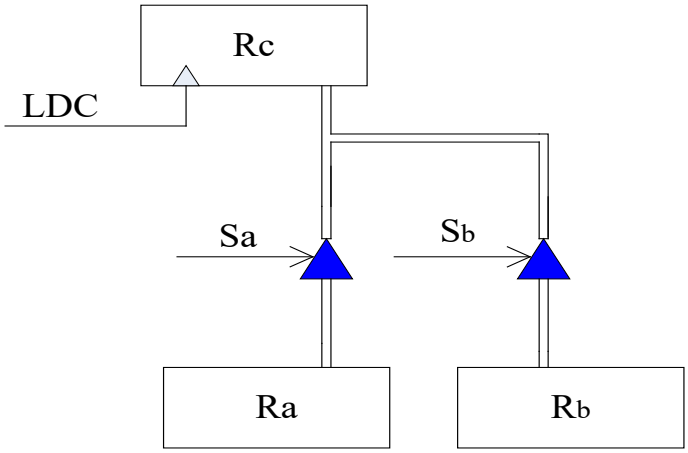
解：源代码：

```
LIBRARY   IEEE;
USE   IEEE. STD_LOGIC_1164. ALL;

ENTITY   asm   IS
   PORT  （clk，k，reset：IN STD_LOGIC;
             qout：OUT STD_LOGIC _VECTOR（1 downto 0））;
END   asm;

ARCHITECTURE   asm_arch   OF   asm   IS
   TYPE   asm_st   IS   (s0,s1,s2,s3);
```

```
SIGNAL  current_state，next_state：asm_st；
BEGIN
  reg：  PROCESS（clk，reset）
              BEGIN
                   IF  (reset＝'1')  THEN
                       current_state＜＝s0；
                   ELSIF  (clk'event  AND  clk＝'1')  THEN
                       current_state＜＝next_state；
                   END IF；
         END PROCESS；
  com：  PROCESS（current_state，k）
              BEGIN
                   CASE  current_state  IS
                        WHEN  s0＝＞qout＜＝"00"；
                             IF  (k＝'0')  THEN
                               next_state＜＝s1；
                             ELSE
                               next_state＜＝s0；
                             END IF；
                        WHEN  s1＝＞qout＜＝"01"；
                             IF  (k＝'0')  THEN
                               next_state＜＝s1；
                             ELSE
                               next_state＜＝s2；
                             END IF；
                        WHEN  s2＝＞qout＜＝"10"；
                             IF  (k＝'0')  THEN
                               next_state＜＝s3；
                             ELSE
                               next_state＜＝s2；
                             END IF；
                        WHEN  s3＝＞qout＜＝"11"；
                             IF  (k＝'0')  THEN
                               next_state＜＝s3；
                             ELSE
                               next_state＜＝s0；
                             END IF；
                        WHEN  others＝＞ next_state＜＝s0；
                   END CASE；
             END PROCESS；
END  asm_arch；
```

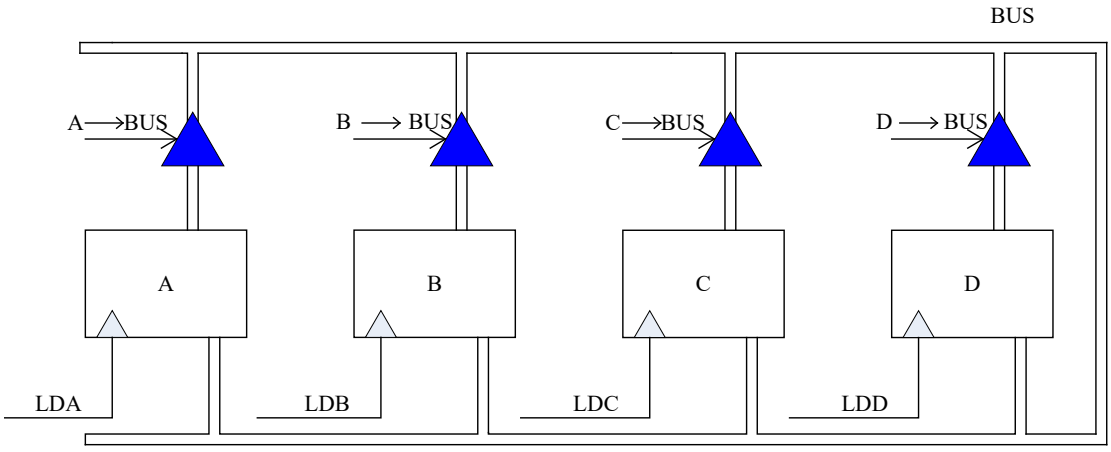
第六章　习题答案

1 现有 D 触发器组成的三个 n 位寄存器，需要连接起来传送数据。当控制信号 S_a 有效时，执行（Ra）→Rc 的操作；当控制信号 S_b 有效时，执行（R_b）→Rc 的操作。试写出连接电路的逻辑表达式，并画出逻辑电路图。
解：

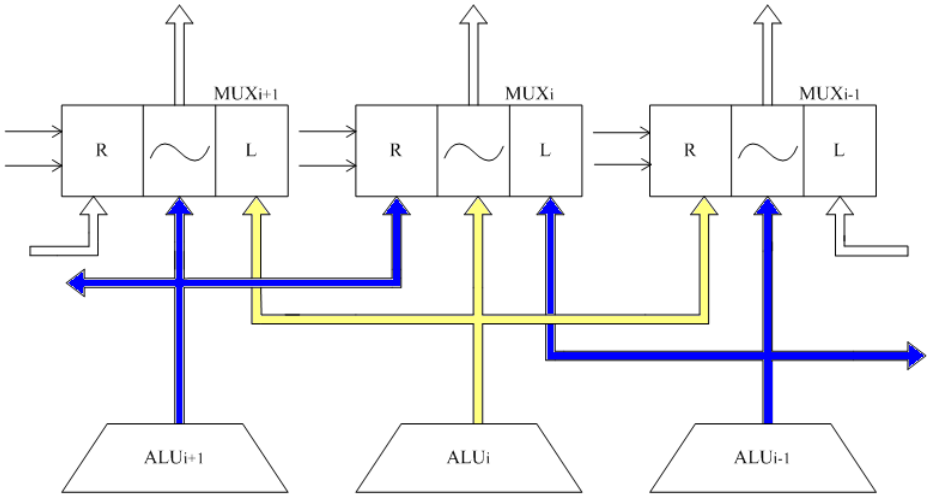


$$Rc = Ra \cdot Sa \cdot LDC + Rb \cdot Sb \cdot LDC$$

2 现有 D 触发器组成的四个 8 位寄存器，要求它们之间实现数据传送，试设计连接电路。
解：

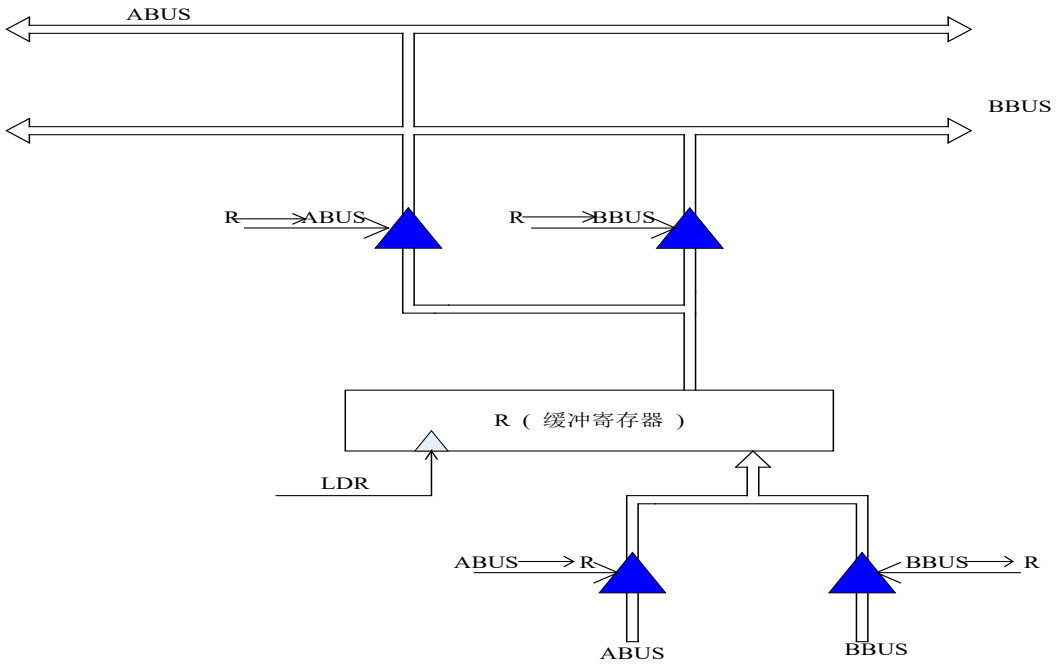


3 ALU 的输出端一般带有一个移位器，其功能为：①ALU 输出正常传送；②ALU 输出左移 1 位（ALU_{i+1}）传送；③ALU 输出右移一位（ALU_{i-1}）传送。试设计移位器的逻辑电路。
解：



4 一个系统有 A,B 两条总线，为了接收来自任何一条总线上的数据并驱动任何一条总线，需要一个总线缓冲寄存器。请用 D 触发器和三态门设计一个总线缓冲寄存器。

解：

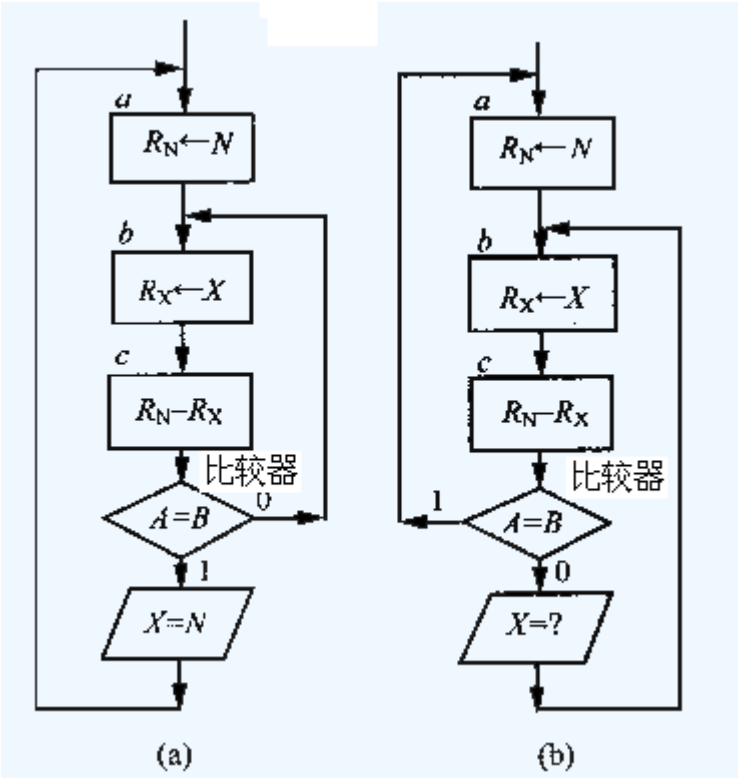


5 试构造能完成下列程序操作的 ASM 图：

（a） if X = N, then … 。

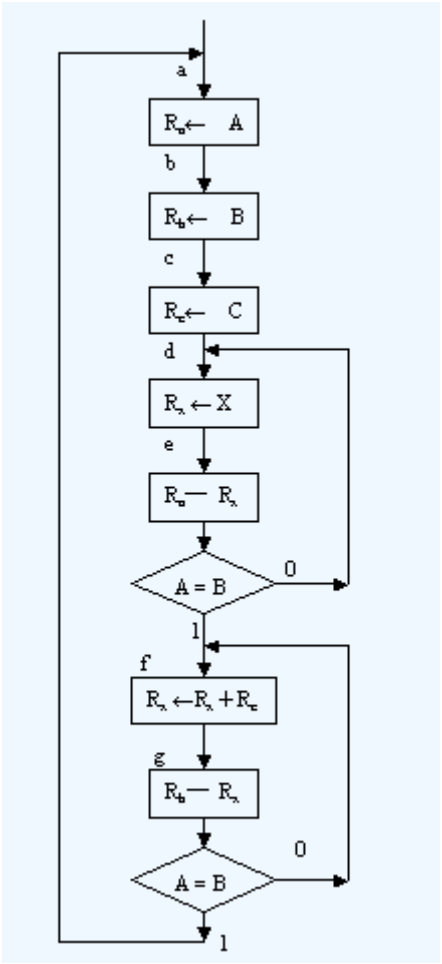
（b） if X≠N, then …, else …。

解：



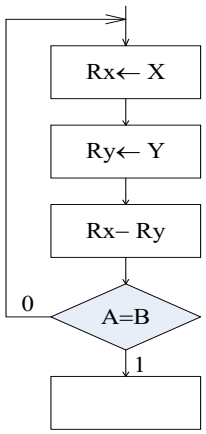
（c） for X from A to B, step C, do… 。

解：



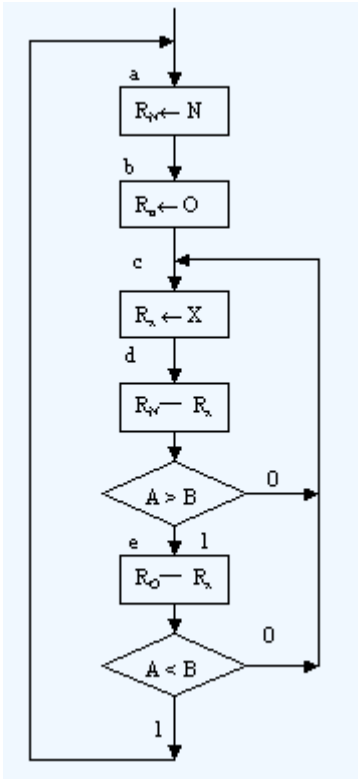
（d） while X = Y, do …。

解：



（e） if X > N OR X < O, then …, else …。

解：

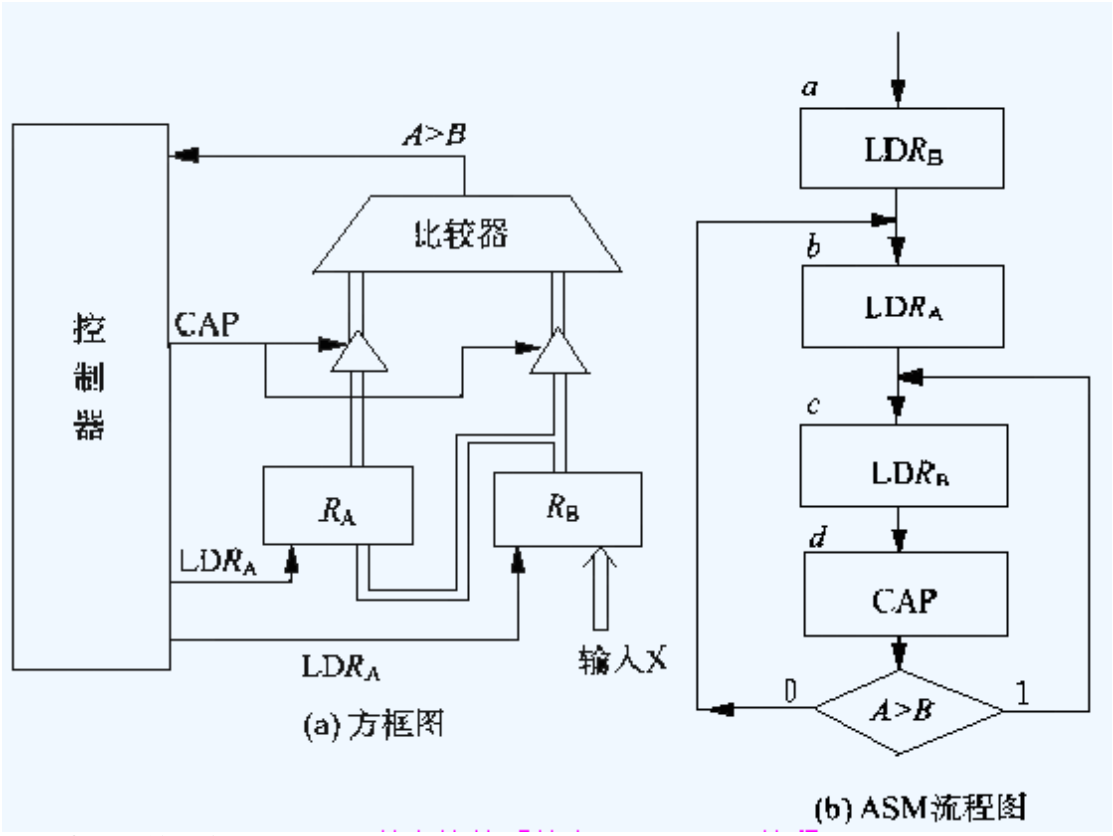


6 有一个数字比较系统，它能对两个 8 位二进制进行比较。其操作过程如下：先将两个 8 位二进制数存入寄存器 A 和 B, 然后进行比较，最后将大数移入寄存器 A 中。要求：

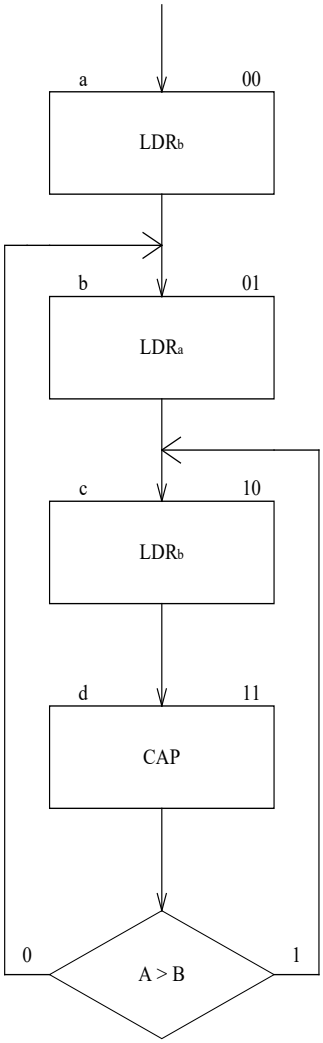
⑴ 画出此系统方框图，并构造 ASM 流程图。

⑵ 设计一个计数器型控制器。

解：（1）


A>B
比较器
控制器
CAP
R_A
R_B
LDR_A
LDR_A
输入X
(a) 方框图
a
LDR_B
b
LDR_A
c
LDR_B
d
CAP
0
A>B
1
(b) ASM流程图

②状态转移真值表



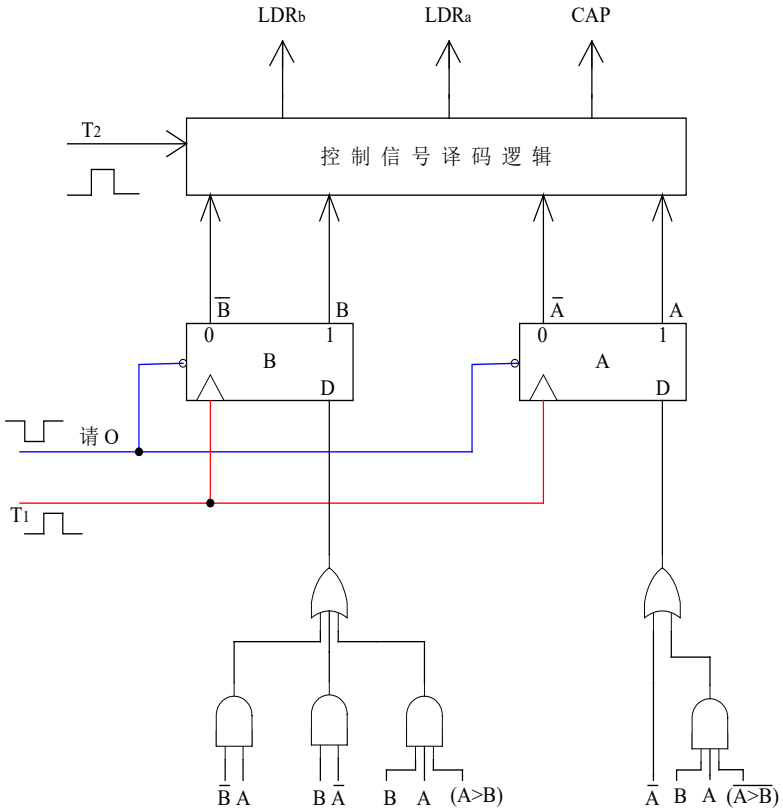
PS		NS		转移条件 C
B	A	B(D)	A(D)	
0	0	0	1	无条件转移
0	1	1	0	无条件转移
1	0	1	1	无条件转移
1	1	1	0	(A > B) = 1
		0	1	A > B = 0

根据 $NS = \sum PS \cdot C$ 公式，激励方程表达式为：

$$B(D) = \bar{B}A + B\bar{A} + BA \cdot (A > B)$$

$$A(D) = \bar{B}\bar{A} + B\bar{A} + BA \cdot \overline{(A > B)} = \bar{A} + BA \cdot \overline{(A > B)}$$

③ 电路图

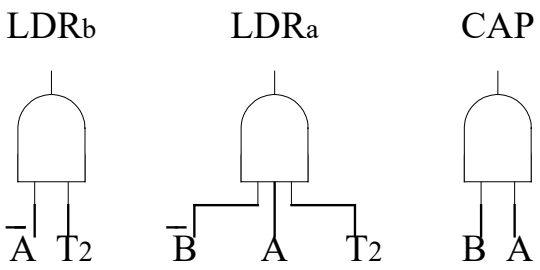


④ 控制信号表达式：

$$LDR_b = (\text{状态 a} + \text{状态 c})\,T_2 = (\bar{B}\bar{A} + B\bar{A})\,T_2 = \bar{A}T_2$$

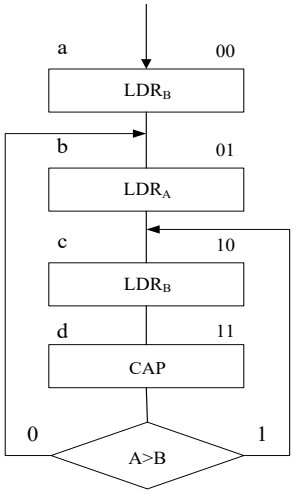
$$LDR_a = \text{状态 b} \cdot T2 = \bar{B}AT_2$$

$$CAP = \text{状态 d} = BA$$



7. 根据题 6 的条件，设计一个 MUX 型控制器。

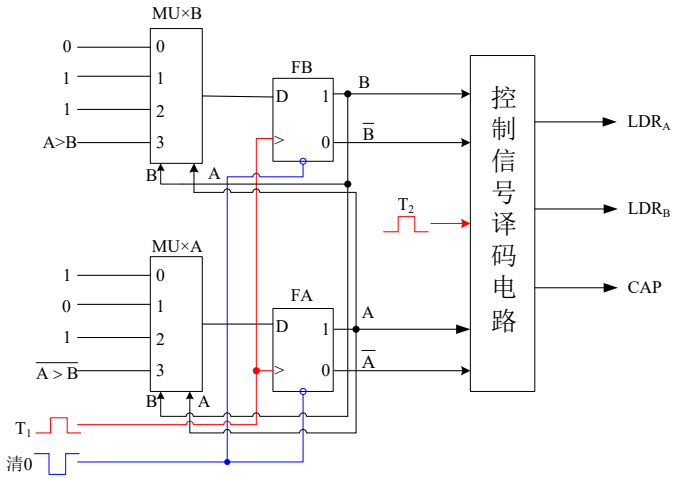
① ASM 流程图



② 状态转移表

十进制编码	PS		NS		转移条件C
	B	A	B(D)	A(D)	
0 (00)	0	0	0	1	C_B=0 , C_A=1
1 (01)	0	1	1	0	C_B=1, C_A=0
2 (10)	1	0	1	1	C_B=1, C_A=1
3 (11)	1	1	1	0	C_B=(A>B), C_A=0
			0	1	C_B=0, $C_A=\overline{A>B}$

③ 电路图



④ 控制信号表达式为：

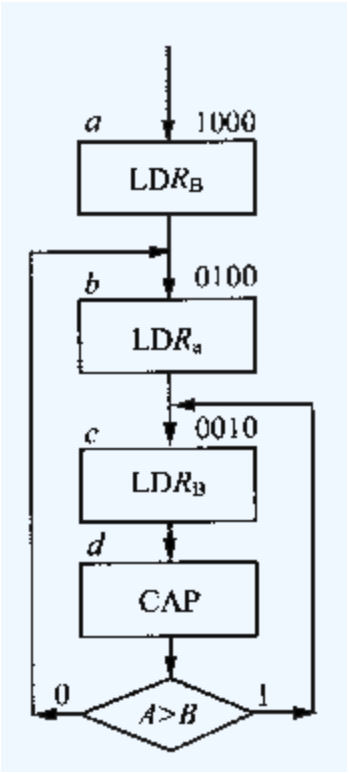
LDR_B=(状态 a＋状态 c)·T_2=($\overline{B}\,\overline{A}$ ＋B$\overline{A}$)·T_2

LDR_A=状态 b·T_2=$\overline{B}$A·T_2

CAP=状态 d＝BA

8. 根据题 6 的条件，设计一个定序型控制器。

① ASM 流程图



② 状态转移表

现态(PS)				次态(NS)				转移条件(C)
Q_a	Q_b	Q_c	Q_d	Q_a	Q_b	Q_c	Q_d	
1	0	0	0	0	1	0	0	初始化强置“1”
0	1	0	0	0	0	1	0	
0	0	1	0	0	0	0	1	
0	0	0	1	0	0	1	0	A>B
0	0	0	1	0	1	0	0	$\overline{A>B}$

（3）写出激励方程 NS=Σ PS・C

$Q_a=0$

$Q_b=Qa+(A>B)\cdot Qd$

$Q_c=Qb+(A>B)\cdot Qd$

$Q_d=Qc$

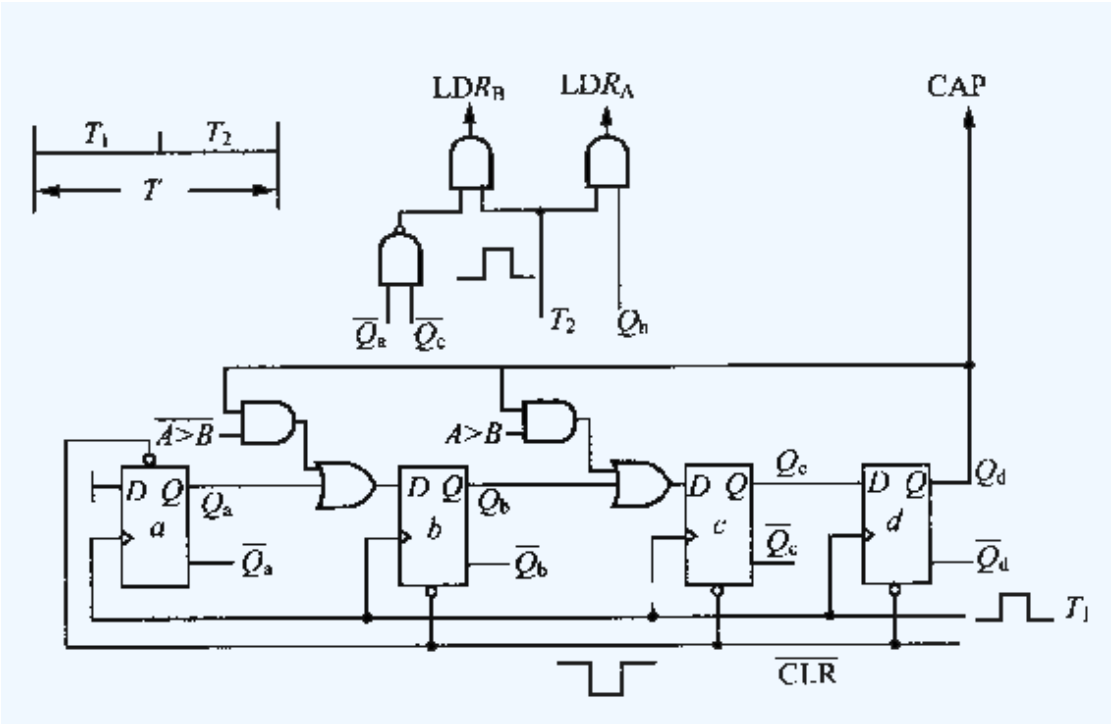
控制信号表达式

$LDR_B=(Q_a+Q_c)\cdot T2$

$LDR_A=Q_b\cdot T2$

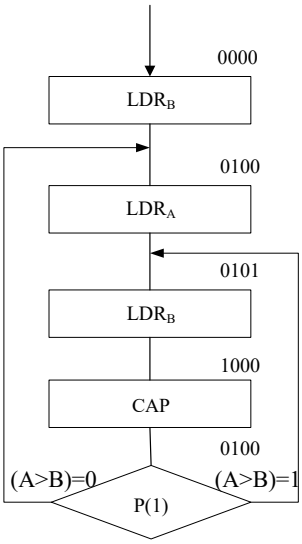
$CAP=Q_d$

（4）逻辑电路图

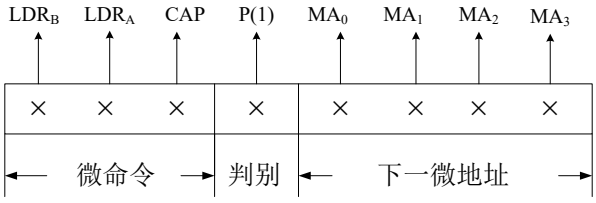


9. 根据题 6 的条件，设计一个微程序控制器。

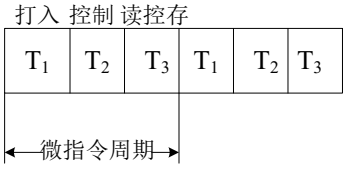
① 微程序流程图



② 微指令格式



③ 定时信号

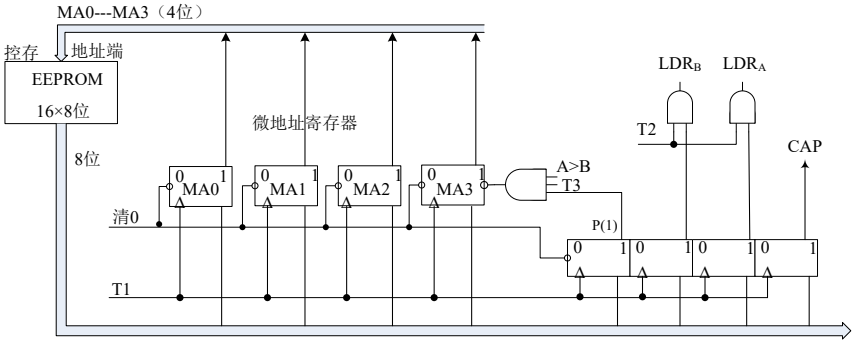


T1-----打入微指令寄存器定时

T2-----执行部件控制信号定时

T3-----修改微地址并读出控存定时

④ 微程序控制器电路



⑤ 微程序代码

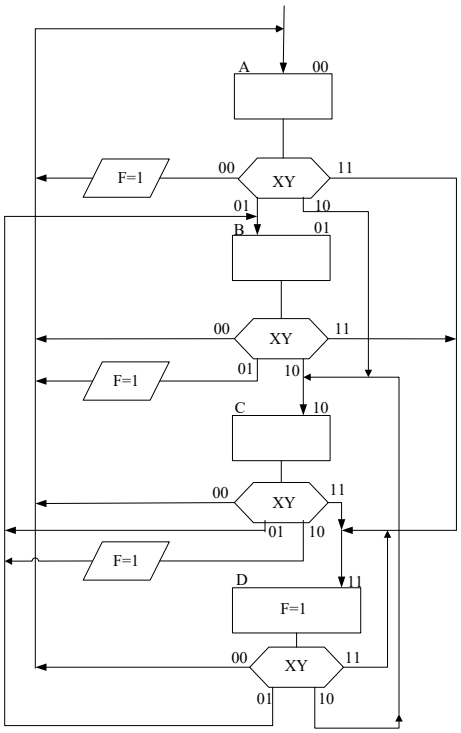
当前微地址	微指令二进制代码		
	微命令	判别	下一微地址
0000	100	0	0100
0100	010	0	0101
0101	100	0	1000
1000	001	1	0100

10.某控制器的状态表如下表所示，其中 X 和 Y 为输入变量，试设计一个计数器型控制器。

PS	NS				输出F			
	XY=00	01	10	11	XY=00	01	10	11
A	A	B	C	D	1	0	0	0
B	A	A	C	D	0	1	0	0
C	A	B	B	D	0	0	1	0
D	A	B	C	D	1	1	1	1

① ASM 流程图与编码(Q_1,Q_2 为两个触发器)

令 状态 A=00, B=01, C=10, D=11



② 状态转移表

PS		NS		转移条件
Q_2	Q_1	Q_2	Q_1	
0	0	0	0	$\bar{x}\bar{y}$
		0	1	$\bar{x}y$
		1	0	$x\bar{y}$
		1	1	xy
0	1	0	0	$\bar{x}\bar{y}$
		0	0	$\bar{x}y$
		1	0	$x\bar{y}$
		1	1	xy
1	0	0	0	$\bar{x}\bar{y}$
		0	1	$\bar{x}y$
		0	1	$x\bar{y}$
		1	1	xy
1	1	0	0	$\bar{x}\bar{y}$
		0	1	$\bar{x}y$
		1	0	$x\bar{y}$
		1	1	xy

③ 激励方程表达式

利用 NS=∑PS·C 公式，使用 D 触发器。

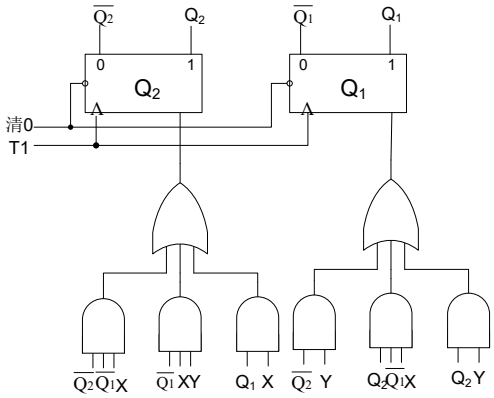
$$Q_2(D)=\overline{Q_2}\,\overline{Q_1}\cdot X\overline{Y}+\overline{Q_2}\,\overline{Q_1}\cdot XY+\overline{Q_2}\,Q_1\cdot X\overline{Y}+\overline{Q_2}\,Q_1\cdot XY+Q_2\overline{Q_1}\cdot XY+Q_2\,Q_1\cdot X\overline{Y}+Q_2\,Q_1\cdot XY$$

$$=\overline{Q_2}\,\overline{Q_1}\cdot X+\overline{Q_1}\cdot XY+Q_1\cdot X$$

$$Q_1(D)=\overline{Q_2}\,\overline{Q_1}\cdot Y+\overline{Q_2}\,Q_1\cdot XY+Q_2\overline{Q_1}(X+Y)+Q_2Q_1\cdot Y$$

$$=\overline{Q_2}\cdot Y+Q_2\overline{Q_1}\cdot X+Q_2\cdot Y$$

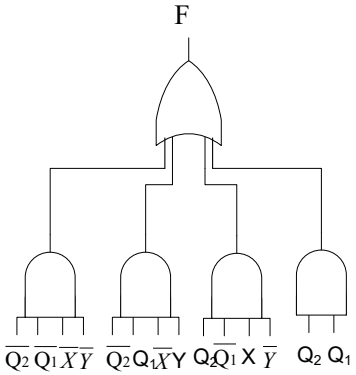
④ 电路图



⑤ 控制信号表达式(假设为电位控制信号)

$$F=\text{状态 A}\cdot\overline{X}\,\overline{Y}+\text{状态 B}\cdot\overline{X}Y+\text{状态 C}\cdot X\overline{Y}+\text{状态 D}$$

$$=\overline{Q_2}\,\overline{Q_1}\cdot\overline{X}\,\overline{Y}+\overline{Q_2}Q_1\cdot\overline{X}Y+Q_2\overline{Q_1}\cdot X\overline{Y}+Q_2Q_1$$



11. .根据题 10 的条件，设计一个 MUX 型控制器

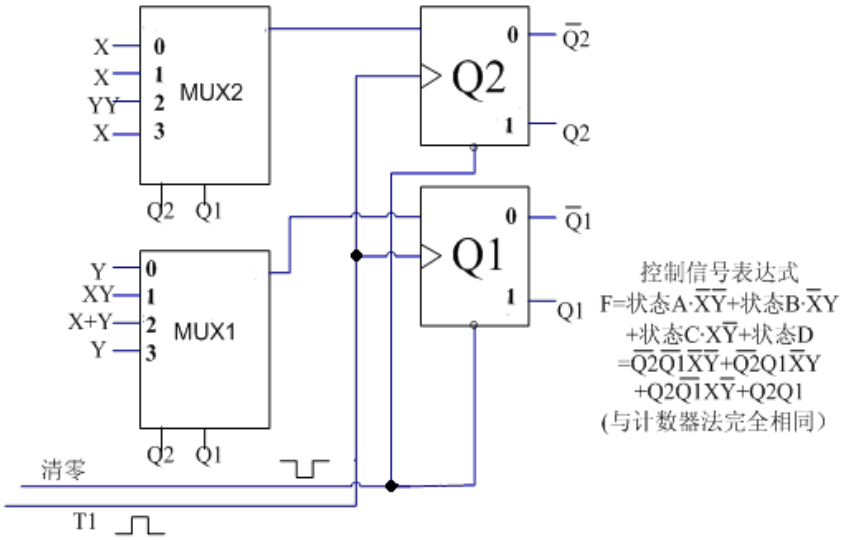
解答：

1） ASM 流程图与编码同计数器型控制器(见第 10 题答案)

2） 按 MUX 方式列出状态转移真值表

MUX编码 十进制	二进制	PS Q2	PS Q1	NS Q2	NS Q1	转移条件	多路开关输入
0	(00)	0	0	0	0	$\overline{X}\,\overline{Y}$	C2=$\overline{X}\overline{Y}$+XY=X C1=$\overline{X}$Y+XY=Y
				0	1	$\overline{X}Y$	
				1	0	$X\overline{Y}$	
				1	1	XY	
1	(01)	0	1	0	0	$\overline{X}\,\overline{Y}$	C2=$X\overline{Y}$+XY=X C1=XY
				0	0	$\overline{X}Y$	
				1	0	$X\overline{Y}$	
				1	1	XY	
2	(10)	1	0	0	0	$\overline{X}\,\overline{Y}$	C2=XY C1=$\overline{X}$Y+$X\overline{Y}$+XY=X+Y
				0	1	$\overline{X}Y$	
				0	1	$X\overline{Y}$	
				1	1	XY	
3	(11)	1	1	0	0	$\overline{X}\,\overline{Y}$	C2=$\overline{X}$Y+XY=X C1=$\overline{XY}$+XY=Y
				0	1	$\overline{X}Y$	
				1	0	$X\overline{Y}$	
				1	1	XY	

3） 画出电路图



12.根据题 10 的条件，设计一个定序型控制器

解答:

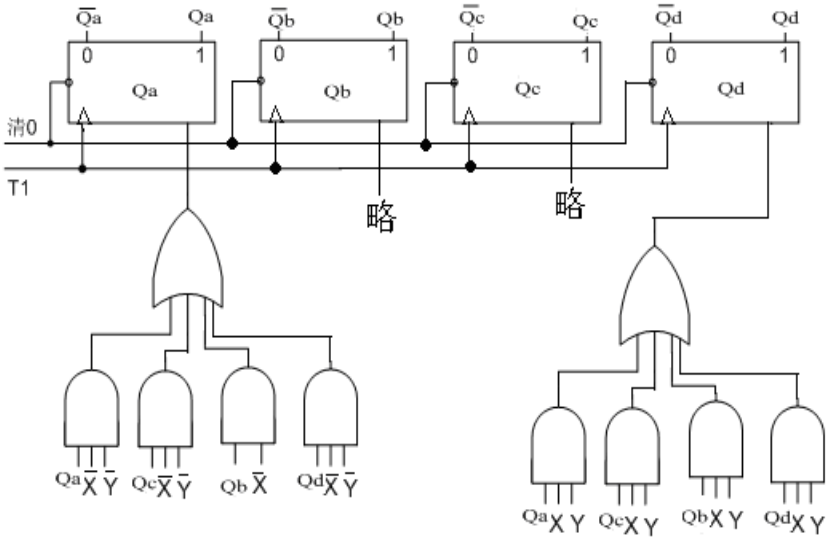
1) ASM 流程图与计数器法相同
2) 使用 Qa、Qb、Qc、Qd 四个触发器，编码分别为 Qa=1000，Qb=0100，Qc=0010,Qd=0001
3) 状态转移真值表

PS				NS				转移条件
Qa	Qb	Qc	Qd	Qa	Qb	Qc	Qd	
1	0	0	0	1	0	0	0	$\bar{X}\bar{Y}$
				0	1	0	0	$\bar{X}Y$
				0	0	1	0	$X\bar{Y}$
				0	0	0	1	XY
0	1	0	0	1	0	0	0	$\bar{X}\bar{Y}$
				1	0	0	0	$\bar{X}Y$
				0	0	1	0	$X\bar{Y}$
				0	0	0	1	XY
0	0	1	0	1	0	0	0	$\bar{X}\bar{Y}$
				0	1	0	0	$\bar{X}Y$
				0	1	0	0	$X\bar{Y}$
				0	0	0	1	XY
0	0	0	1	1	0	0	0	$\bar{X}\bar{Y}$
				0	1	0	0	$\bar{X}Y$
				0	0	1	0	$X\bar{Y}$
				0	0	0	1	XY

4） 写出激励方程 NS=Σ PS・C

$$Qa(D)=Qa\bar{X}\bar{Y}+Qb\bar{X}+Qc\bar{X}\bar{Y}+Qd\bar{X}\bar{Y}$$
$$Qb(D)=Qa\bar{X}y+Qc\bar{X}Y+QcX\bar{Y}+Qd\bar{X}Y$$
$$Qc(D)=QaX\bar{Y}+QbX\bar{Y}+QdX\bar{Y}$$
$$Qd(D)=QaXY+QbXY+QcXY+QdXY$$

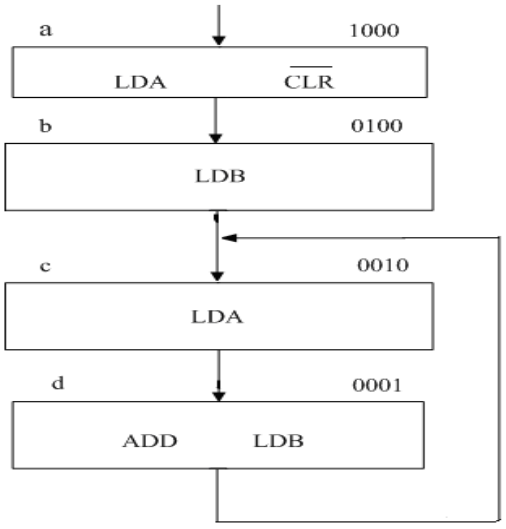
5)画出电路图



13.设计一个累加运算系统定序型控制器

解答：

1） 算法流程图



2） 状态转移真值表及激励函数表达式

PS				NS			
Qa	Qb	Qc	Qd	Qa(D)	Qb(D)	Qc(D)	Qd(D)
1	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0
0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0

NS=∑PS •C(C=1, 无条件转移)

$Qa(D)=\overline{Qa+Qb+Qc+Qd}$

$Qb(D)=Qa$

$Qc(D)=Qb+Qd$

$Qd(D)=Qc$

3） 控制信号表达式

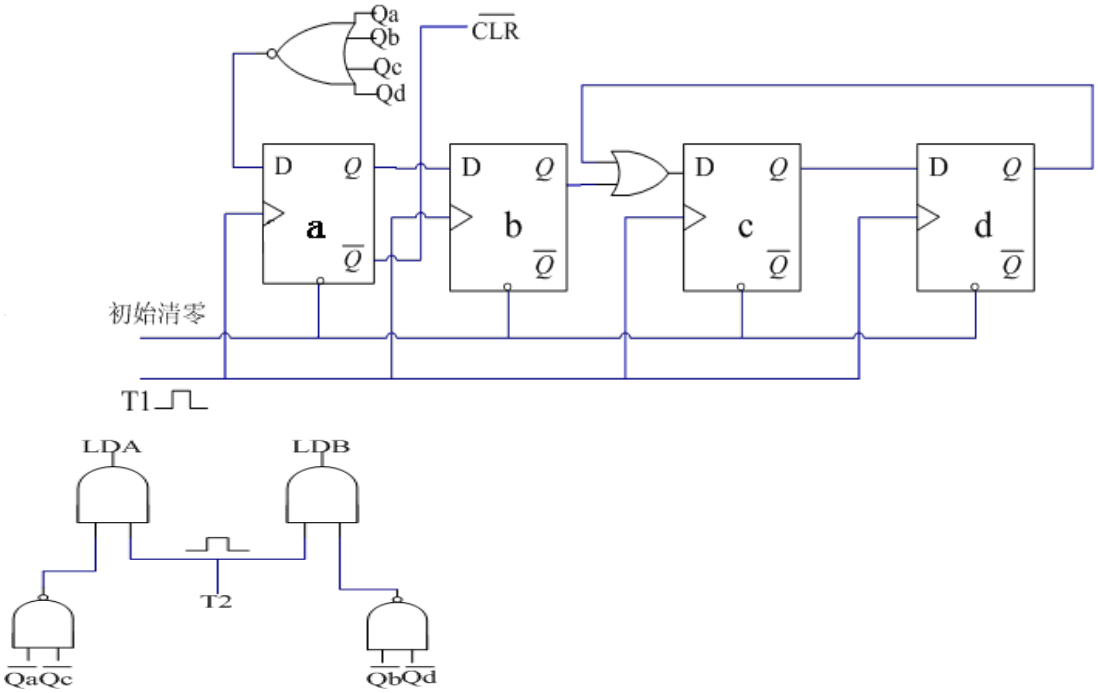
$LDA=(Qa+Qc)T2$

$LDB=(Qb+Qd)T2$

$\overline{CLR}=\overline{Qa}$

$ADD=Qd$

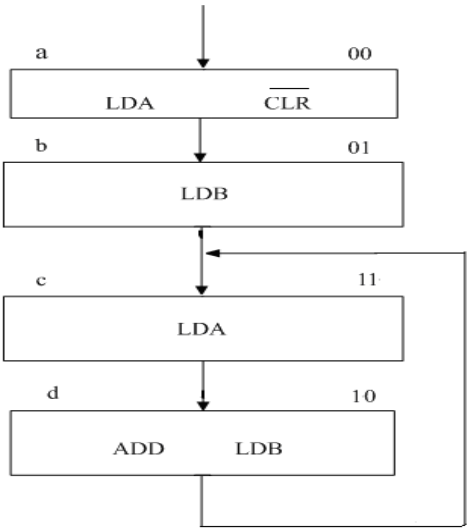
4） 电路图



14.设计一个累加运算系统 MUX 型控制器

解答：

1） ASM 流程图



2）状态转移真值表及激励表达式

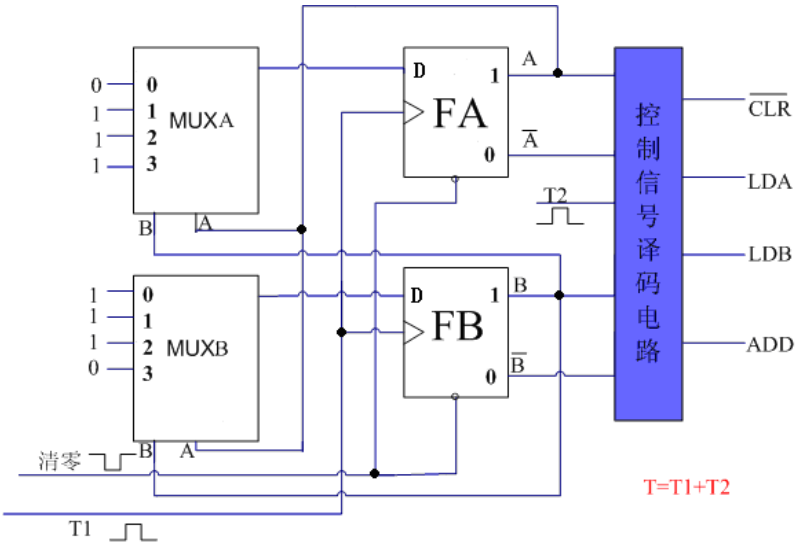
PS		NS			转移条件C	
十进编码	状态名	状态名	B(D)	A(D)		
0	(00) a	b	0	1	C(B)=0	C(A)=1
1	(01) b	c	1	1	C(B)=1	C(A)=1
2	(10) d	c	1	1	C(B)=1	C(A)=1
3	(11) c	d	1	0	C(B)=1	C(A)=0

NS=∑PS・C

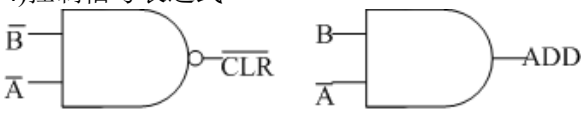
$B(D)=\bar{B}AC\bar{b}+B\bar{A}C\bar{b}+BAC\bar{b}$

$A(D)=\bar{B}\bar{A}C\bar{a}+\bar{B}AC\bar{a}+B\bar{A}C\bar{a}$

3）电路图



4)控制信号表达式



$\overline{CLR}$=状态a=$\overline{\bar{B}\,\bar{A}}$

ADD= 状态d=B $\bar{A}$

LDA=(状态a+状态c)T2=($\bar{B}\,\bar{A}$+B A)T2

LDB=(状态b+状态d)T2=($\bar{B}$ A+B $\bar{A}$)T2

15. 图 P6.1 所示 ASM 流程图，设计计数器型控制器

解：（1）ASM 流程图与编码(Q_1,Q_2 为两个触发器)

令 状态 a=00, b=01, c=11, d=10

② 状态转移表

PS		NS		转移条件
Q_2	Q_1	Q_2	Q_1	
0	0	0	1	
0	1	0	1	$x\bar{y}$
		1	0	$xy\bar{z}$
		1	1	$\bar{x}$
1	1	0	0	$x\oplus w$
		1	0	$\bar{x}\bar{w}$
		1	1	xw
1	0	0	0	

（3）次态方程

$$Q_2^n = \overline{Q_2}Q_1xy\bar{z} + Q_2Q_1\bar{x}\bar{w} + Q_2Q_1xw$$

$$Q_1^n = \overline{Q_2}\overline{Q_1} + \overline{Q_2}Q_1x\bar{y} + \overline{Q_2}Q_1\bar{x} + Q_2Q_1xw$$

（4）控制信号

$$F = \overline{Q_2}Q_1xy$$

16．根据图 P6.1 所示 ASM 流程图，设计一个 MUX 型控制器

解：（1）ASM 流程图、编码、状态转移真值表同计数器型控制器(见第 15 题答案)

（2） MUXA 的输出接触发器 D_2，MUXB 的输出接触发器 D_1，则

$$MUXA(0) = 0$$

$$MUXA(1) = xy\bar{z} + \bar{x} = y\bar{z} + \bar{x}$$

$$MUXA(2) = 0$$

$$MUXA(3) = \bar{x}\bar{w} + xw$$

$$MUXB(0) = 1$$

$$MUXB(1) = x\bar{y} + \bar{x} = \bar{y} + \bar{x}$$

$$MUXB(2) = 0$$

$$MUXB(3) = xw$$

（3）控制信号

$$F = \overline{Q_2}Q_1xy$$

17. 根据图 P6.1 所示 ASM 流程图，设计一个定序型控制器

解： 1)使用 Qa、Qb、Qc、Qd 四个触发器对应四个状态 a，b，c，d

2)状态转移真值表及激励方程表达式

PS				NS				转移条件
Q_a	Q_b	Q_c	Q_d	Q_a	Q_b	Q_c	Q_d	
1	0	0	0	0	1	0	0	
0	1	0	0	0	1	0	0	$x\bar{y}$
				0	0	0	1	$xy\bar{z}$
				0	0	1	0	$\bar{x}$
0	0	1	0	1	0	0	0	$x\oplus w$
				0	0	0	1	$\bar{x}\bar{w}$
				0	0	1	0	xw
0	0	0	1	1	0	0	0	

$$Q_a(D) = Q_c x \oplus w + Q_d$$
$$Q_b(D) = Q_a + Q_b x\bar{y}$$
$$Q_c(D) = Q_b \bar{x} + Q_c xw$$
$$Q_d(D) = Q_b xy\bar{z} + Q_c \bar{x}\bar{w}$$

（3）控制信号

$$F = Q_b xy$$

18. 根据图 P6.1 所示 ASM 流程图，设计一个微程序控制器。

解：步骤如下

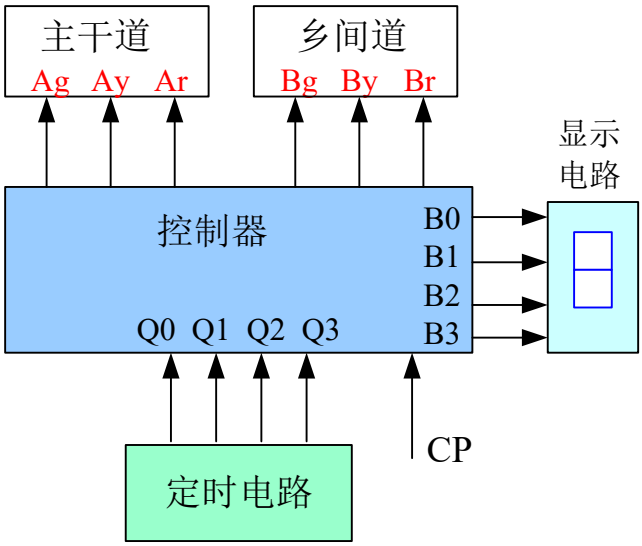
- 将 ASM 流程图转化为微程序流程图
- 确定微指令地址
- 确定微命令
- 确定微指令格式和字长
- 确定控制存储器容量
- 写出微地址转移逻辑表达式
- 将微指令编译成二进制代码。

19. 根据教材图 P6.7 所示通路，设计一个微程序控制器。

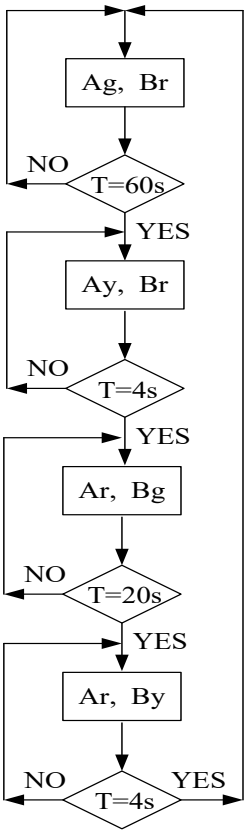
略

20. 设计十字路口交通灯控制器

解：交通灯控制系统结构框图

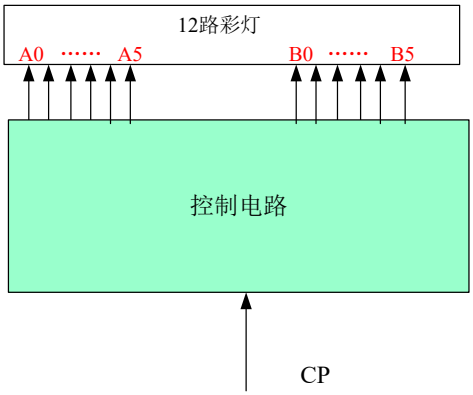


控制系统 ASM 图如下



21．设计一个彩灯控制器。

解：彩灯电路框图如下




```
library ieee;
   use ieee.std_logic_1164.all;
   use ieee.std_logic_unsigned.all;
   entity light is
   port(clk1:        in        std_logic;                              ---时钟信号
        light:       buffer    std_logic_vector(11 downto 0));         --输出
   end light;
   architecture behv of light is
      constant       len:      integer:=11;
      signal         banner:       std_logic:='0';       ----定义信号 banner 为两种节拍转换信号；
      signal         clk,clk2: std_logic;                          ----信号 CLK，CLK2 作为辅助时
钟
   begin
      clk<=(clk1 and banner) or (clk2 and not banner);
      process(clk1)
      begin
           if  clk1'event  and  clk1='1'  then                          ---CLK1 二分频得 CLK2
                clk2<=not clk2;
           end if;
      end process;
      process(clk)                                              ----
           variable   flag: bit_vector(3 downto 0):="0000";
      begin
           if clk'event and clk='1' then
                if flag="0000" then
                     light<='1' & light(len downto 1);                ----顺序向右循环移位
                     if light(1)='1' then                              ----依次点亮
                          flag:="0001";
                     end if;
                elsif flag="0001" then                                  ————依次熄灭
                     light<=light(len-1 downto 0) & '0';
                     if light(10)='0' then
                          flag:="0010";
                     end if;
                elsif flag="0010" then
                     light<= light(len-1 downto 0) & '1';             ----顺序向左循环移位
                     if light(10)='1' then                             ----依次点亮
                          flag:="0011";
                     end if;
                elsif flag="0011" then                                  ————依次熄灭
                     light<= '0' & light(len downto 1);
```

```
                if light(1)='0' then
                    flag:="0100";
                end if;
            elsif flag="0100" then
                light(len downto 6)<=light(len-1 downto 6)&'1';        ---从中间向两边点
                light(len-6 downto 0)<='1'&light(len-6 downto 1);
                if light(1)='1' then
                    flag:="0101";
                end if;
            elsif flag="0101" then
                light(len downto 6)<='0'&light(len downto 7);          ----从两边向中间熄
                light(len-6 downto 0)<=light(len-7 downto 0)&'0';
                if light(2)='0' then
                    flag:="0110";
                end if;
            elsif flag="0110" then
                light(len downto 6)<='1'&light(len downto 7);           ----奇  偶位循环点亮
                light(len-6 downto 0)<='1'&light(len-6 downto 1);
                if light(1)='1' then
                    flag:="0111";
                end if;
            elsif flag="0111" then
                light<="000000000000";
                flag:="1000";
            elsif     flag="1000" then                                          ----从新开始
                banner<=not  banner;                        ---banner 信号转换，实现第二种节
拍
                flag:="0000";
            end if;
        end if;
    end process;
end behv;
```