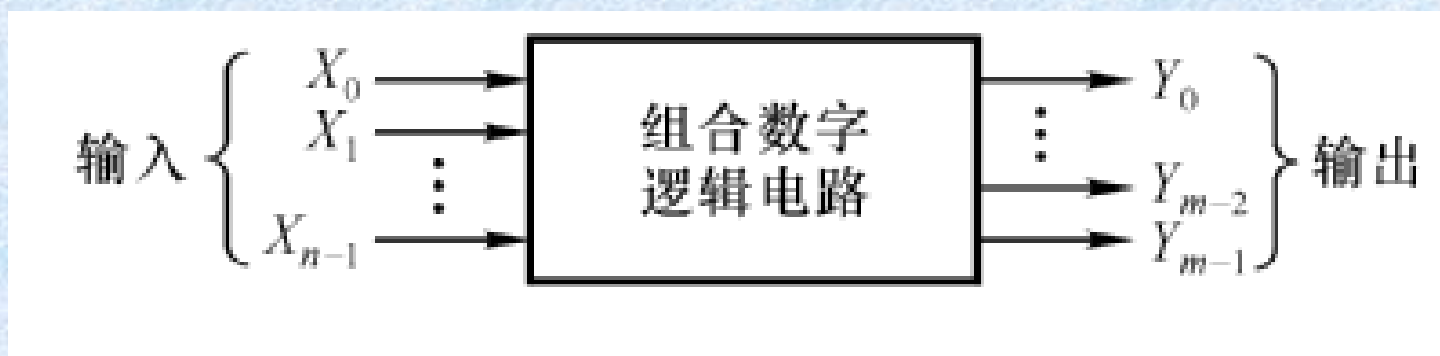


第4章 数字逻辑电路

- 4.1 逻辑电路的分析与设计
- 4.2 中规模集成逻辑电路及应用
- 4.3 硬件描述语言和可编程逻辑器件

4.1.1 逻辑电路的结构与分类

组合逻辑电路在任何时刻的输出状态（结果），只决定于该时刻的输入取值。一旦输入取值确定后，输出结果就可以确定。它的电路框图如图所示：



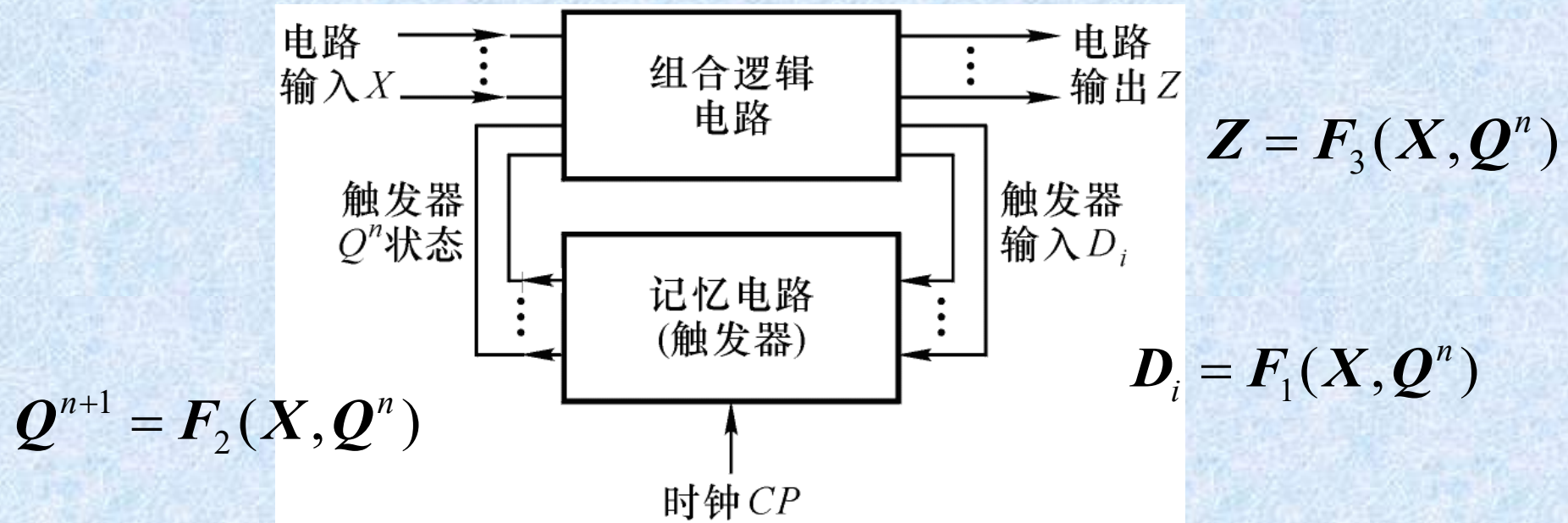
电路可以是多个输入，多个输出，也可以是多输入单输出的组合逻辑电路。

$$Y_0 = f_0(X_0, X_1, \cdots, X_{n-1})$$

$$\vdots$$

$$Y_{m-1} = f_{m-1}(X_0, X_1, \cdots, X_{n-1})$$

时序逻辑电路的输出除了由输入决定外，还和电路的初始状态密切相关



式中 X 是电路输入， Z 是电路输出， Q^n 记忆电路的初态， Q^{n+1} 是记忆电路的次态。

显然，除 X 以外，其它量都与时钟 CP 有关，说明电路需要一个**时钟脉冲信号**来触发或协调工作。

◆ 时序逻辑电路的功能描述

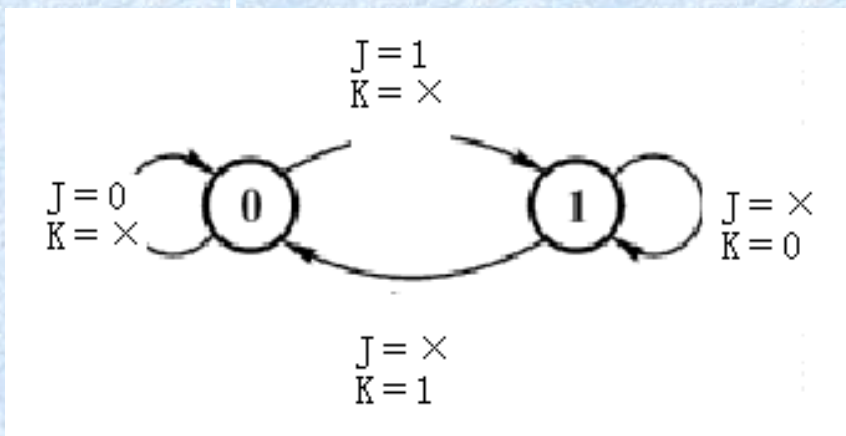
状态真值表（特性表），状态函数（特性方程），状态转换图，时序图，激励表

以JK触发器为例：

$$Q^{n+1} = J\overline{Q}^n + \overline{K}Q^n$$

J	K	Q^n	Q^{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$Q^n \rightarrow Q^{n+1}$	J K	D	R S	T
0 0	0×	0	×0	0
0 1	1×	1	1 0	1
1 0	×1	0	0 1	1
1 1	×0	1	0×	0



◆ 时序逻辑电路的分类

1、按照触发器的触发方式，时序逻辑电路可以分为：同步时序电路和异步时序电路。

如果一个时序电路中所有触发器（FF）在时钟控制下同时工作，则称为同步时序电路。反之，称为异步时序电路。

2、按照时序电路输出方式，时序电路可分为Mealy（米里）型电路和Moore（莫尔）型电路

Moore 型电路输出仅与状态相关，在时钟有效沿才可能发生变化，抗干扰能力更强；

Mealy电路的输出在非时钟有效沿也会随输入信号的变化而变化，输出可以直接受外部输入控制，响应速度快，可以处理异常情况。

4.1.2 组合逻辑电路的**分析**与**设计**

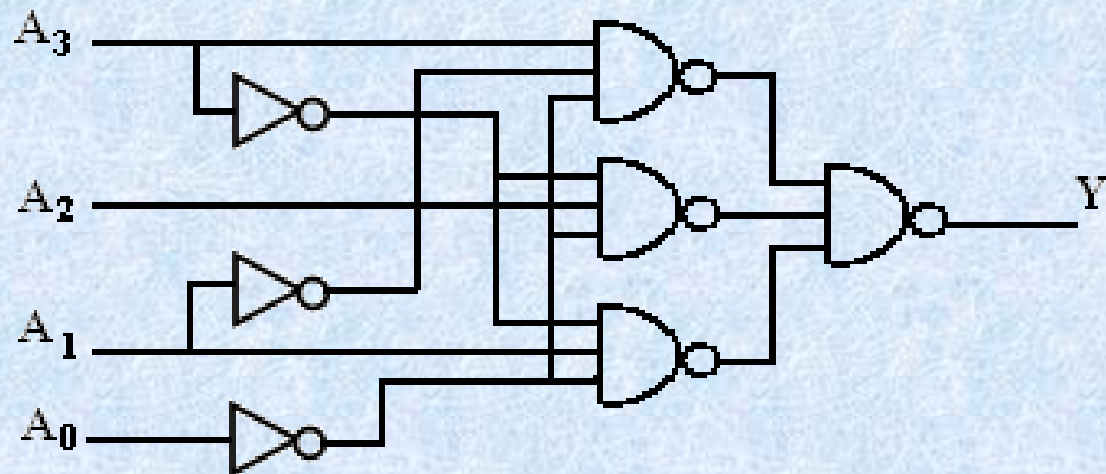
基本逻辑电路的**分析**方法：

在给定电路的情况下，要分析电路的功能时：

- (1) 应先写出各个（每一位）输出的逻辑表达式，
- (2) 在给定各个变量的取值后，列出真值表，
- (3) 最后确定电路的逻辑功能。

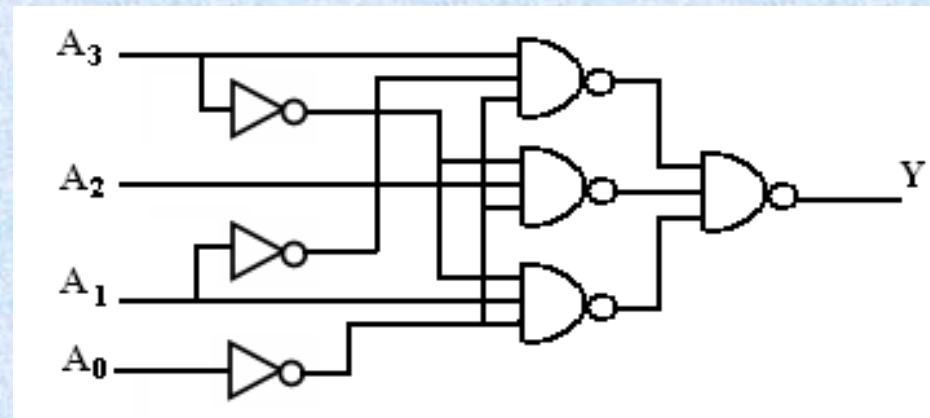
设计方法又如何？

【例4.1.1】 试分析图所示组合逻辑电路，说明电路所具有的逻辑功能



$$\begin{aligned} Y = f(A_3, A_2, A_1, A_0) &= \overline{\overline{A_3} \overline{A_1} \overline{A_0}} \cdot \overline{\overline{A_3} A_2 \overline{A_0}} \cdot \overline{\overline{A_3} A_1 \overline{A_0}} \\ &= A_3 \overline{A_1} \overline{A_0} + \overline{A_3} A_2 \overline{A_0} + \overline{A_3} A_1 \overline{A_0} \end{aligned}$$

4 位二进制码输入				输出
A_3	A_2	A_1	A_0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



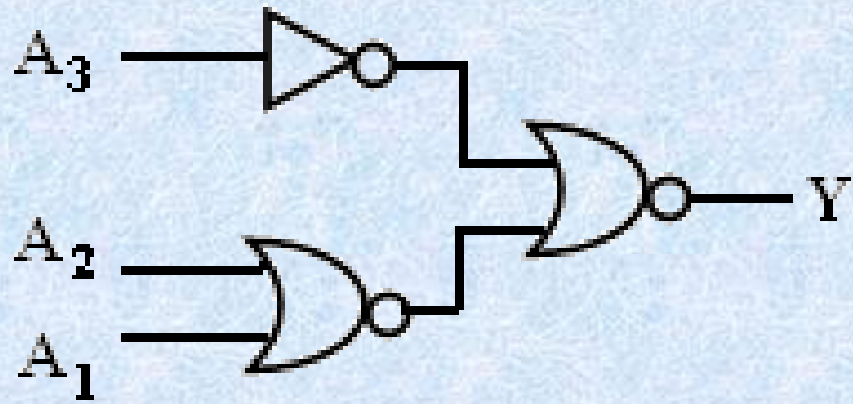
电路功能是：当输入的二进制码小于8时，能被2整除，而输入二进制码大于等于8时能被4整除。

【例4.1.2】设计一个二—十进制代码中检测伪码的组合逻辑电路，其功能要求是：当输入4位二进制代码时，能检测出8421BCD伪码的组合逻辑电路。

解：待设计电路的输入为4位二进制数，输出为检测结果，令输入、输出变量分别为 $A_3A_2A_1A_0$ 和 Y 。假定输入伪码时输出 Y 为1，而输入4位8421BCD码时输出 $Y=0$

$Y \backslash \begin{matrix} A_1A_0 \\ A_3A_2 \end{matrix}$		00	01	11	10
00	0	0	0	0	0
01	0	0	0	0	0
11	1	1	1	1	1
10	0	0	1	1	1

$$Y = A_3 A_2 + A_3 A_1 = A_3 (A_2 + A_1) = \overline{\overline{A_3} + A_2 + A_1}$$



Y \ A ₁ A ₀		00	01	11	10
A ₃ A ₂					
00		0	0	0	0
01		0	0	0	0
11		1	1	1	1
10		0	0	1	1

4.1.3 基本时序电路的 分析与设计

4.1.3.1 基本时序电路的 分析

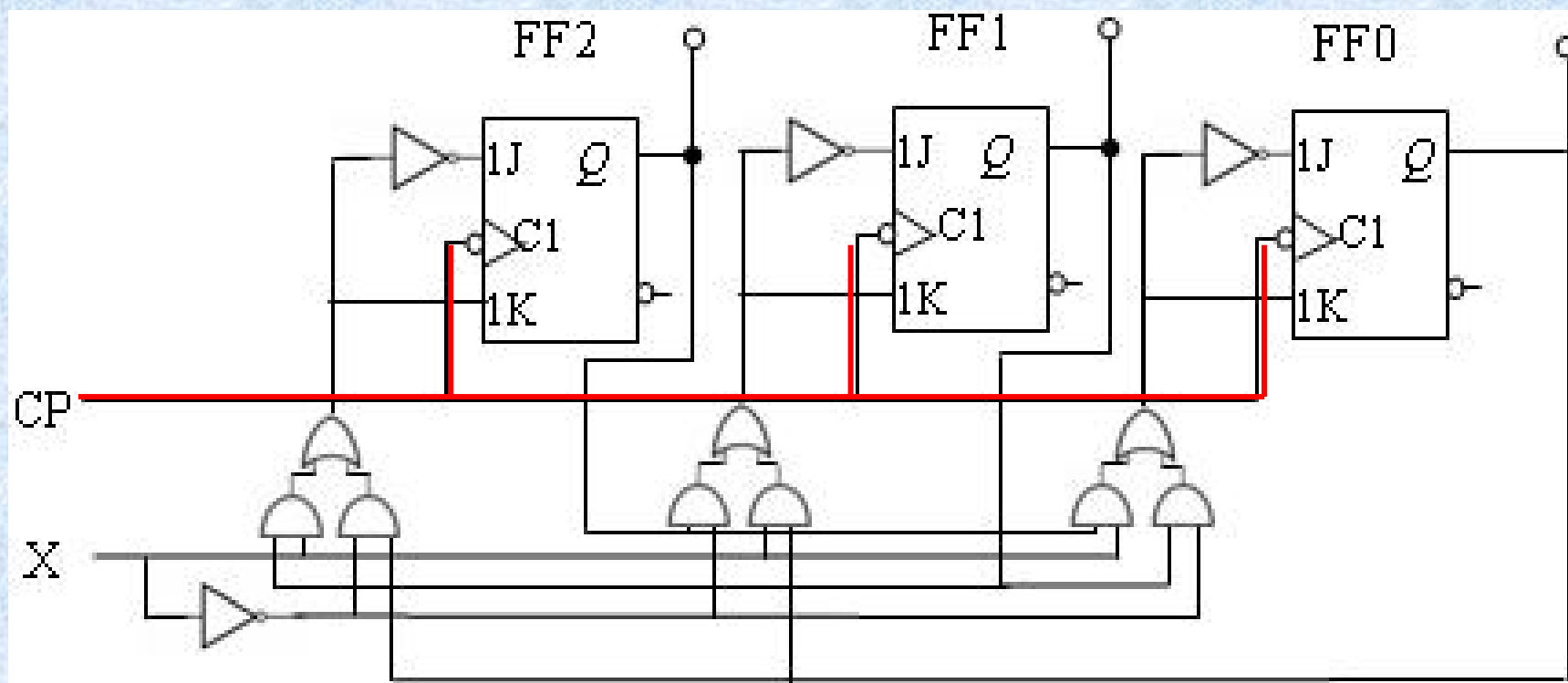
一、同步时序电路分析

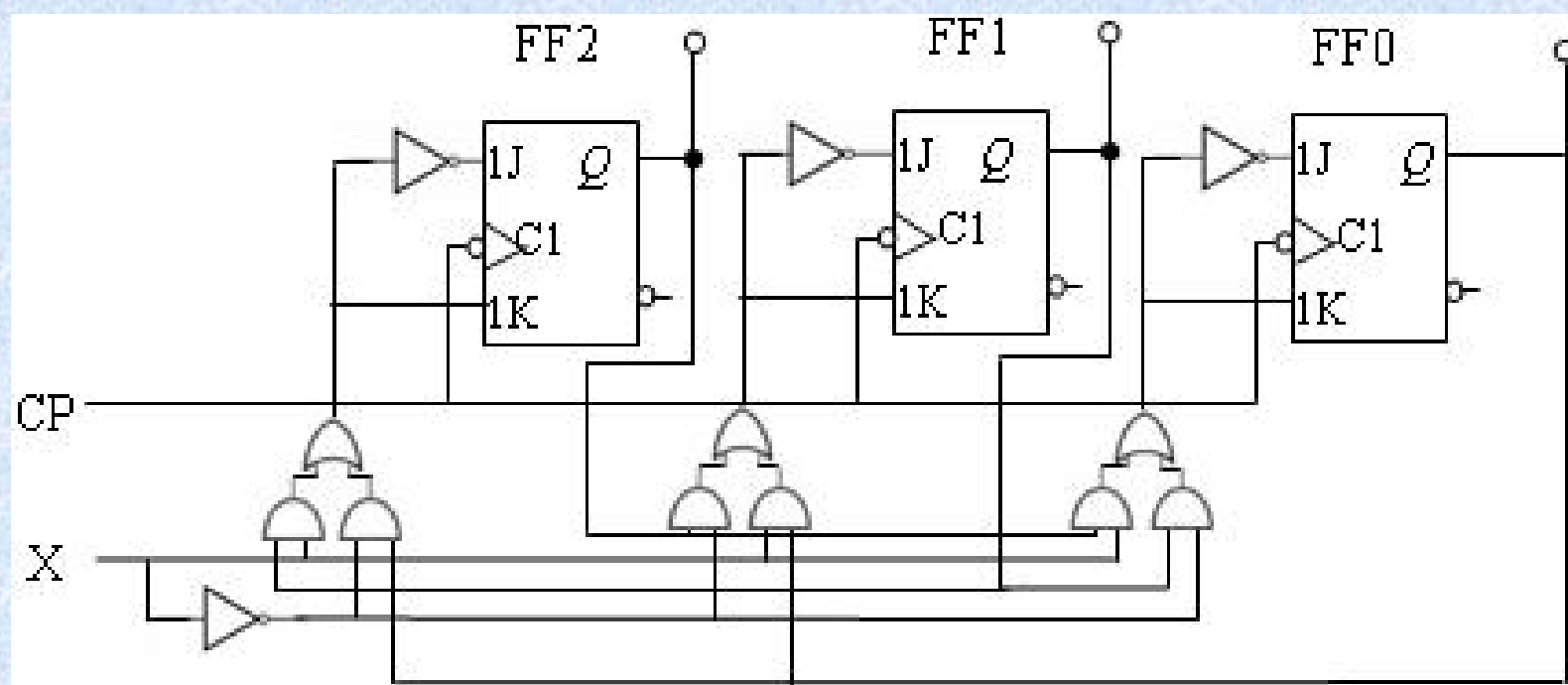
基本步骤:

1. 由电路图写出触发器的驱动方程，特性方程；
2. 驱动方程代入特性方程求触发器状态方程；
3. 依次设定初态代入状态方程求出次态；
4. 列出状态转换真值表、状态转换图、或画出时序图，得出电路逻辑功能。

【例】

每个触发器的CP都连接在一起，同时受触发，所以称同步





(1) 各触发器的驱动方程和电路的输出方程为：

$$J_2 = \overline{\overline{X}Q_0 + XQ_1}, K_2 = \overline{J_2}$$

$$J_1 = \overline{\overline{X}Q_2 + XQ_0}, K_1 = \overline{J_1}$$

$$J_0 = \overline{\overline{X}Q_1 + XQ_2}, K_0 = \overline{J_0}$$

(2) 把驱动方程代入特性方程 $Q^{n+1} = J\overline{Q}^n + \overline{K}Q^n$

可求得触发器状态方程

$$Q_2^{n+1} = f(Q_2^n, Q_1^n, Q_0^n)$$

$$Q_1^{n+1} = f(Q_2^n, Q_1^n, Q_0^n)$$

$$Q_0^{n+1} = f(Q_2^n, Q_1^n, Q_0^n)$$

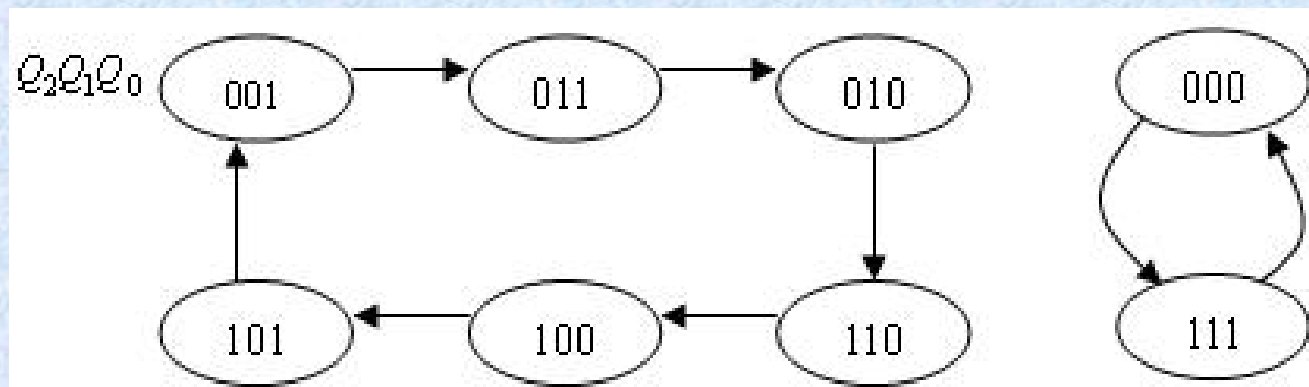
这样可以更方便简洁地列出状态转换真值表

状态真值表

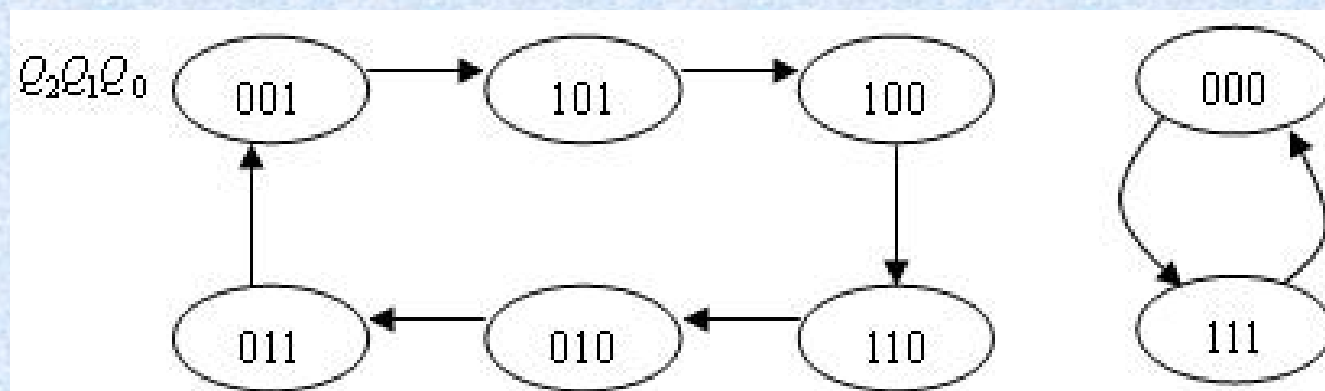
CP 有效沿	X	Q_2	Q_1	Q_0	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
1	0	0	0	0	1	1	1
2	0	0	0	1	0	1	1
3	0	0	1	0	1	1	0
4	0	0	1	1	0	1	0
5	0	1	0	0	1	0	1
6	0	1	0	1	0	0	1
7	0	1	1	0	1	0	0
8	0	1	1	1	0	0	0
9	1	0	0	0	1	1	1
10	1	0	0	1	1	0	1
11	1	0	1	0	0	1	1
12	1	0	1	1	0	0	1
13	1	1	0	0	1	1	0
14	1	1	0	1	1	0	0
15	1	1	1	0	0	1	0
16	1	1	1	1	0	0	0

(3) 可以列出状态转移图

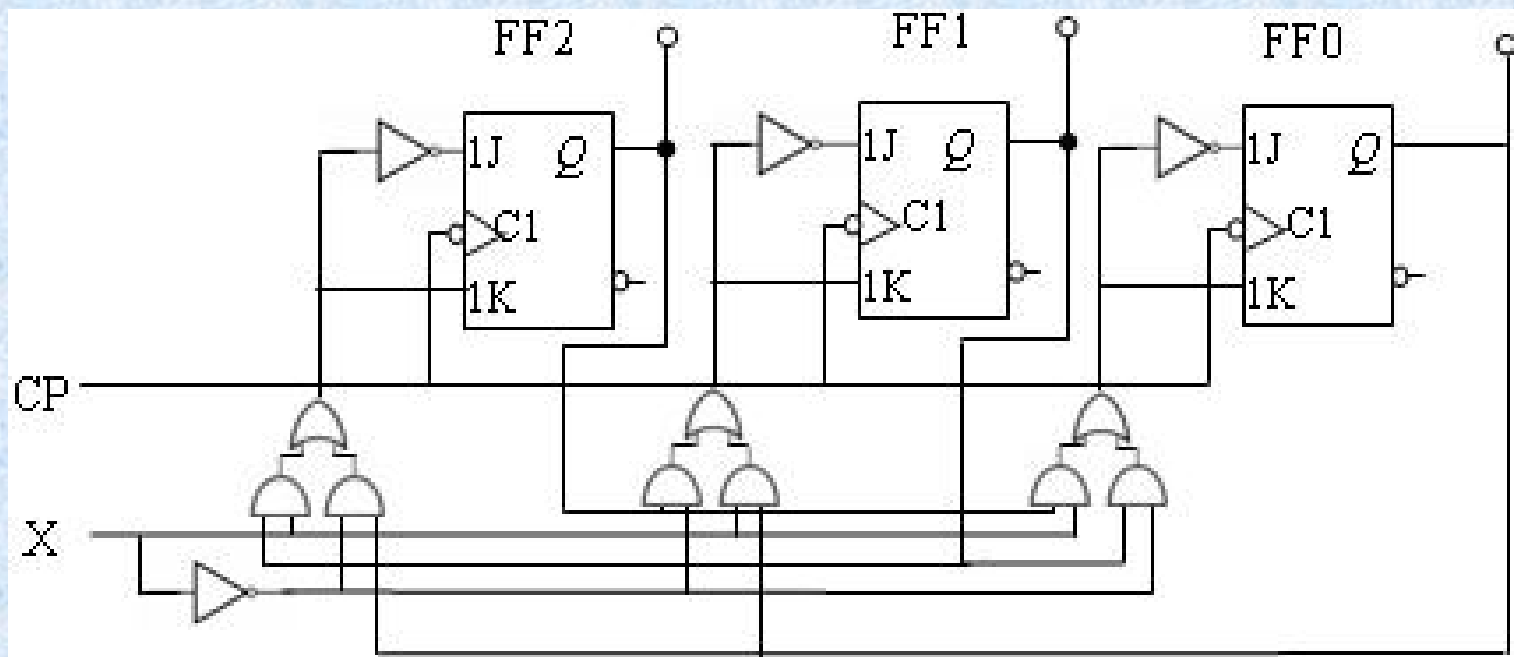
X=0时状态转移图：



X=1时状态转移图：



(4) 由状态转移图得出该时序逻辑电路的逻辑功能



特殊编码的六进制计数器

不能自启动的时序电路

二、异步时序电路分析

基本步骤：

1. 由电路图写出触发器的驱动方程，特性方程，**CP方程**
(同步计数器时不必写)；

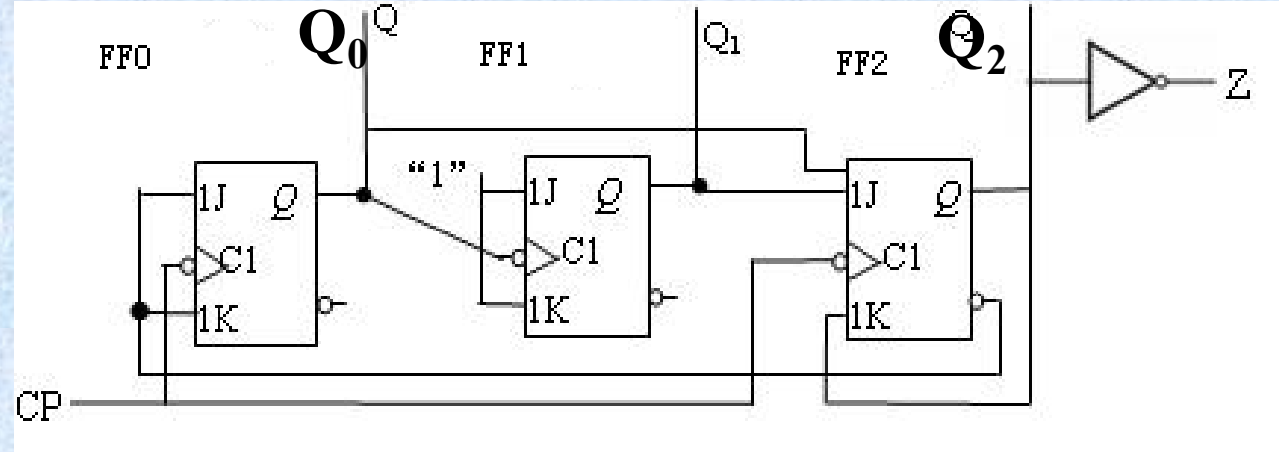
2. 驱动方程代入特性方程求触发器状态方程；

3. 依次设定初态代入状态方程求出次态；

注意各触发器的工作次态（即有否CP脉冲）

4. 列出状态转换真值表、状态转换图、或画出时序图，得出电路逻辑功能。

【例4.1.4】



(1) 触发器的驱动方程

$$1J_0 = 1K_0 = \overline{Q_2^n}$$

$$1J_1 = 1K_1 = 1$$

$$1J_2 = Q_1^n Q_0^n \quad 1K_2 = Q_2^n$$

特别注意CP方程!

$$CP_2 = CP$$

$$CP_1 = Q_0^n$$

$$CP_0 = CP$$

$$Q_2^{n+1} = J_2 \overline{Q_2^n} + \overline{K_2} Q_2^n = \overline{Q_2^n} Q_1^n Q_0^n$$

$$Q_1^{n+1} = J_1 \overline{Q_1^n} + \overline{K_1} Q_1^n = \overline{Q_1^n}$$

$$Q_0^{n+1} = J_0 \overline{Q_0^n} + \overline{K_0} Q_0^n = \overline{Q_2^n} Q_0^n + Q_2^n Q_0^n$$

输出方程: $Z = \overline{Q_2}$

设定初态，依次求出次态：

$$Q_2^{n+1} = J_2 \overline{Q_2^n} + \overline{K_2} Q_2^n = \overline{Q_2^n} Q_1^n Q_0^n \quad CP_2 = CP$$

$$Q_1^{n+1} = J_1 \overline{Q_1^n} + \overline{K_1} Q_1^n = \overline{Q_1^n} \quad CP_1 = Q_0^n$$

$$Q_0^{n+1} = J_0 \overline{Q_0^n} + \overline{K_0} Q_0^n = \overline{Q_2^n} \overline{Q_0^n} + Q_2^n Q_0^n \quad CP_0 = CP$$

$$Q_2^n Q_1^n Q_0^n = 000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 000$$

$$Q_2^n Q_1^n Q_0^n = 101 \rightarrow 001$$

$$Q_2^n Q_1^n Q_0^n = 110 \rightarrow 010$$

$$Q_2^n Q_1^n Q_0^n = 111 \rightarrow 011$$

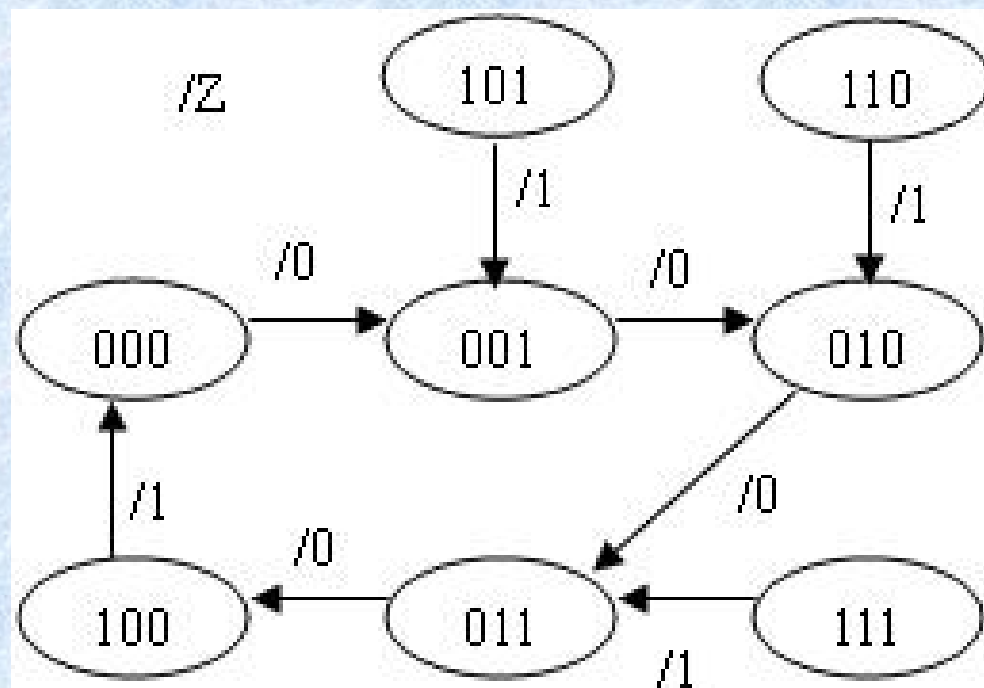
电路的状态转换规律已非常清楚了

这样可以更方便简洁地列出状态转换真值表

状态真值表

CP 有效沿	Q_2	Q_1	Q_0	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	Z
1	0	0	0	0	0	1	1
2	0	0	1	0	1	0	1
3	0	1	0	0	1	1	1
4	0	1	1	1	0	0	1
5	1	0	0	0	0	0	0
6	1	0	1	0	0	1	0
7	1	1	0	0	1	0	0
8	1	1	1	0	1	1	0

(3) 状态转移图



421BCD编码的异步五进制加法计数器

能够自启动的时序逻辑电路

三、二进制计数器的分析

计数器是数字系统中应用极为广泛的一种时序逻辑电路。主要应用在测频，测距，定时和时间测量中，如计算机中的定时器和时钟计数器等。

计数器的分类有：

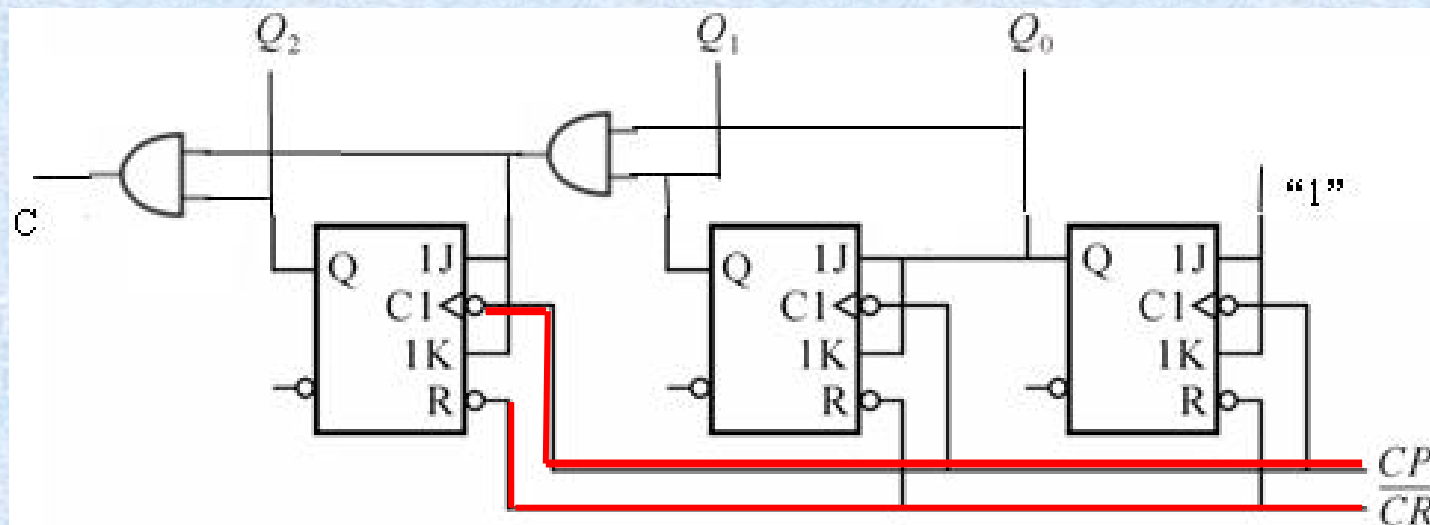
以电路结构分：同步计数器、异步计数器

以电路状态翻转的规律分：二进制计数器、非二进制计数器

以功能分有：加法计数器、减法计数器、可逆计数器

1、同步二进制计数器的分析

图中的每个触发器都连接成T触发器，每个触发器的CP也都连接在一起，同时受触发，所以称同步。



同步二进制加法计数

分析过程：

(1) 各触发器的驱动方程

$$T_0 = 1$$

$$T_1 = Q_0^n$$

$$T_2 = Q_1^n Q_0^n$$

状态 顺序	现态			T端状态			次态		
	Q2	Q1	Q0	T2	T1	T0	Q2	Q1	Q0
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	1	0	1	0
2	0	1	0	0	0	1	0	1	1
3	0	1	1	1	1	1	1	0	0
4	1	0	0	0	0	1	1	0	1
5	1	0	1	0	1	1	1	1	0
6	1	1	0	0	0	1	1	1	1
7	1	1	1	1	1	1	0	0	0

$$T_0 = 1$$

$$T_1 = Q_0^n$$

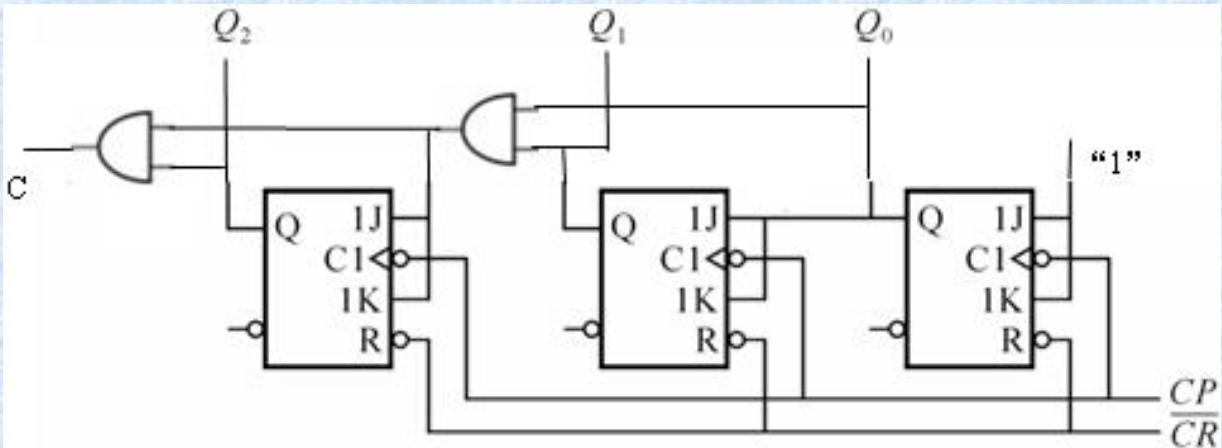
$$T_2 = Q_1^n Q_0^n$$

(2) 得出各状态真值表。

每个触发器翻转条件为T为高电平时，CP脉冲下降沿即翻转。

状态翻转真值表和波形图：

CP顺序	触发器状态		
	Q ₂	Q ₁	Q ₀
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

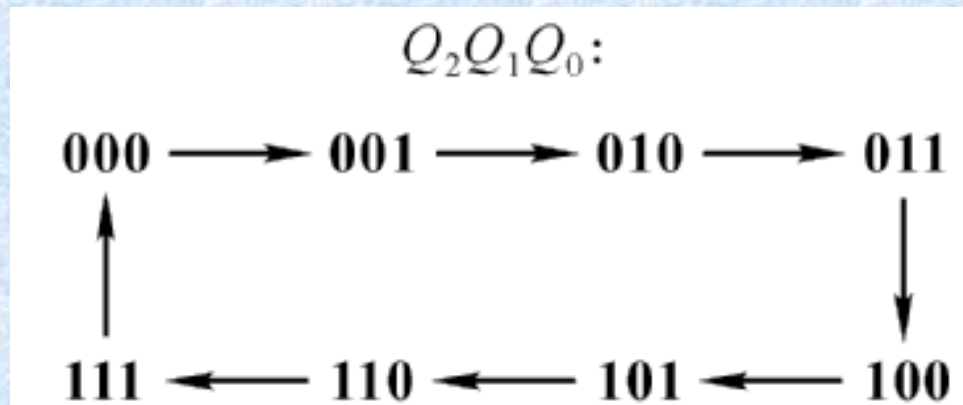


$$T_0 = 1$$

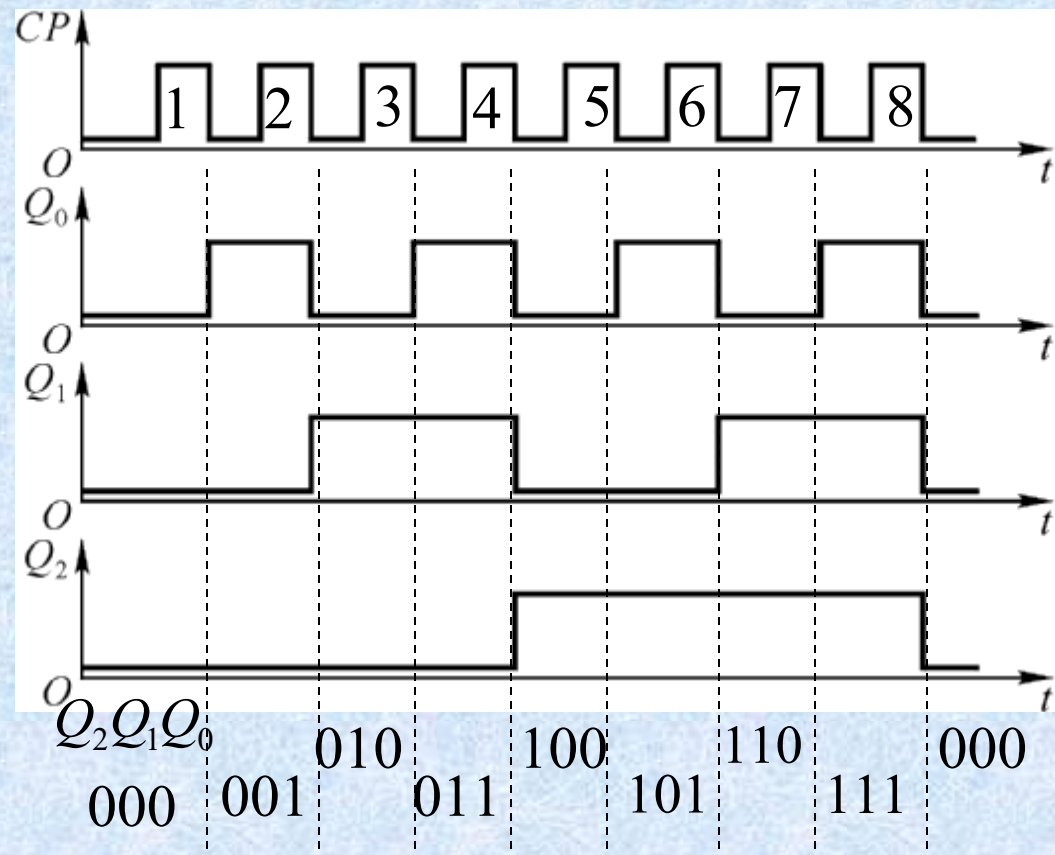
$$T_1 = Q_0^n$$

$$T_2 = Q_1^n Q_0^n$$

(3) 写出状态转换图。



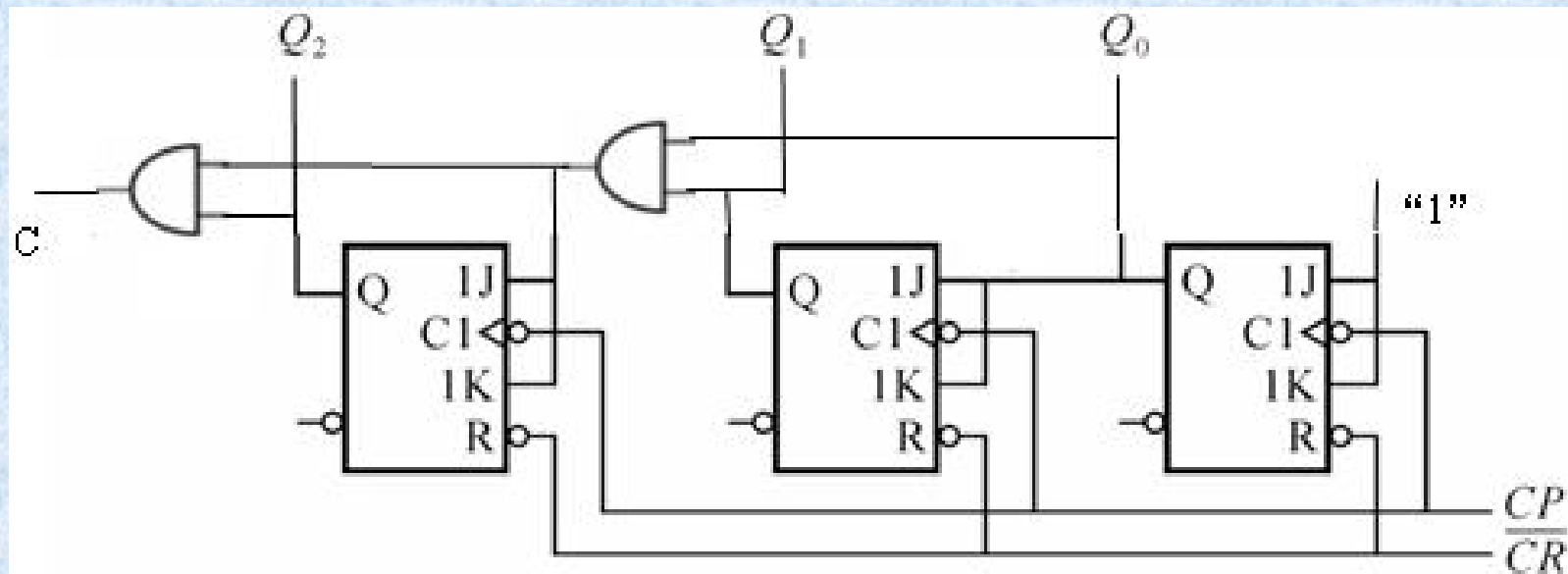
时序图



不管是从真值表，波形图或状态转换图看，都可得出电路是一个同步3位二进制的加法计数器。

由于一次计数循环需要8个CP脉冲，故也称模8计数器。

计数器计数器又有分频器之称，n位二进制的最大分频关系为： $\frac{1}{2^n}$



电路图连接的特点？

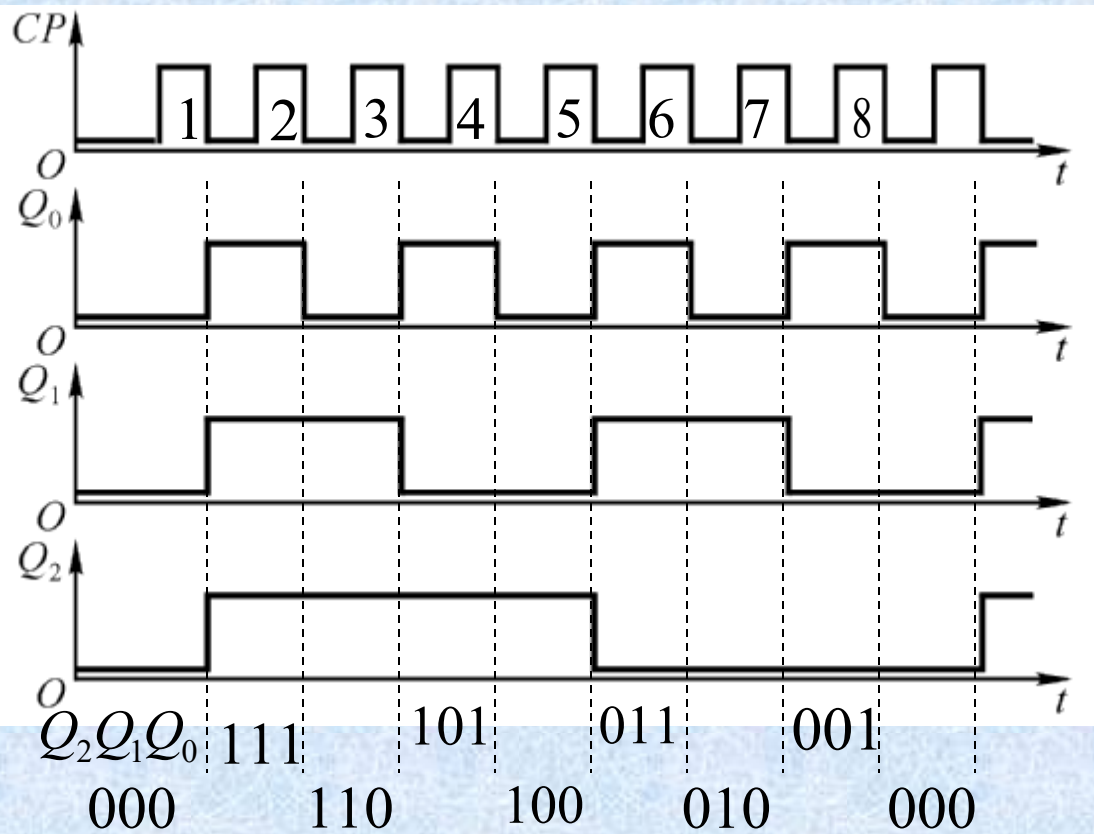
CP顺序	触发器状态		
	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

同步3位二进制减法计数器又如何？

时序图

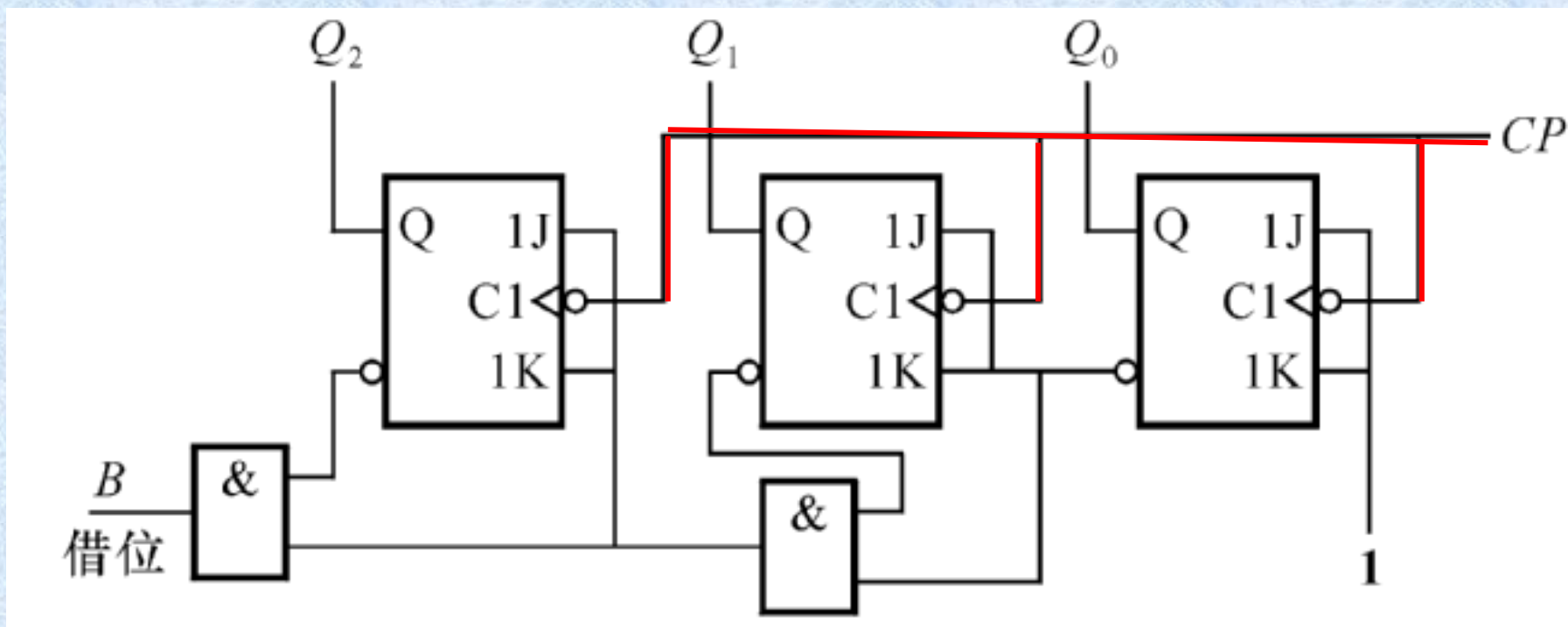
CP顺序	触发器状态		
	Q ₂	Q ₁	Q ₀
0	0	0	0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0

状态转换表

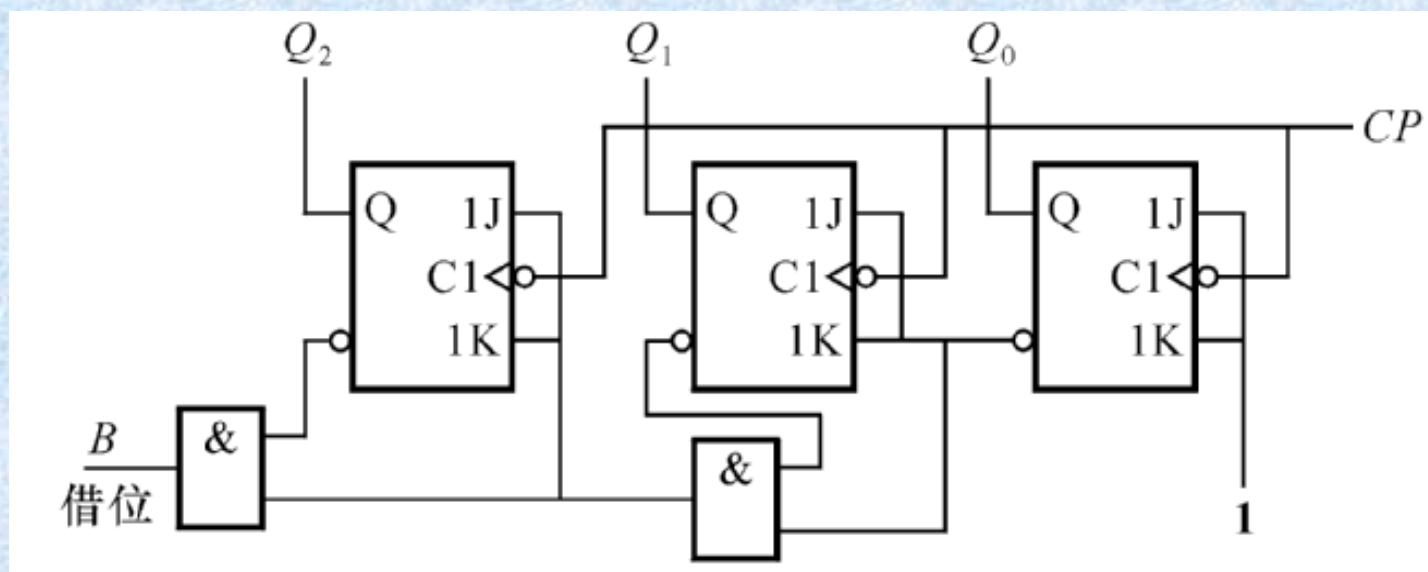
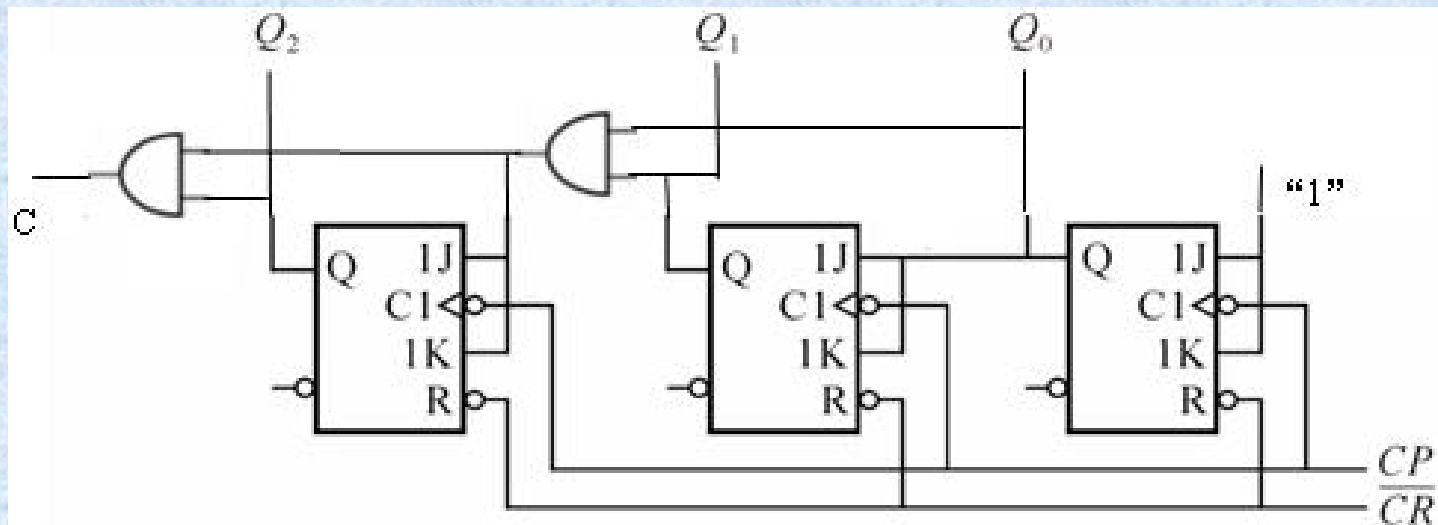


$$T_0 = 1 \quad T_1 = \overline{Q_0} \quad T_2 = \overline{Q_1} \overline{Q_0}$$

注意状态转换表中规律！



同步3位二进制减法计数器

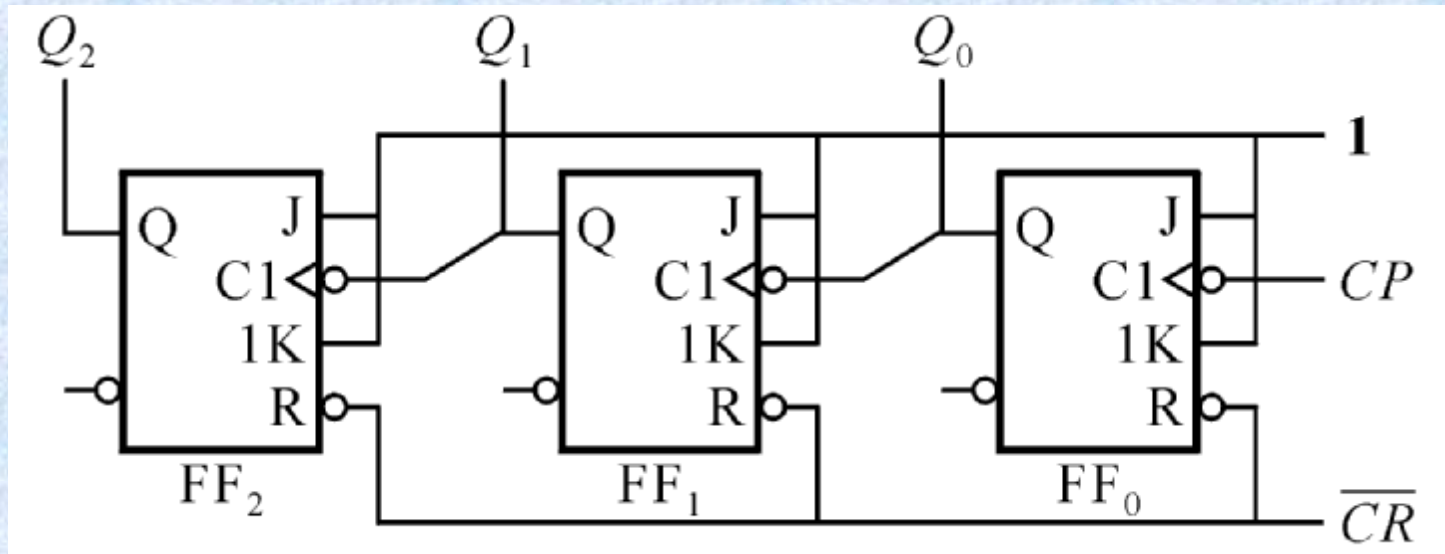


那么，可逆计数器该怎么连接？（设计）

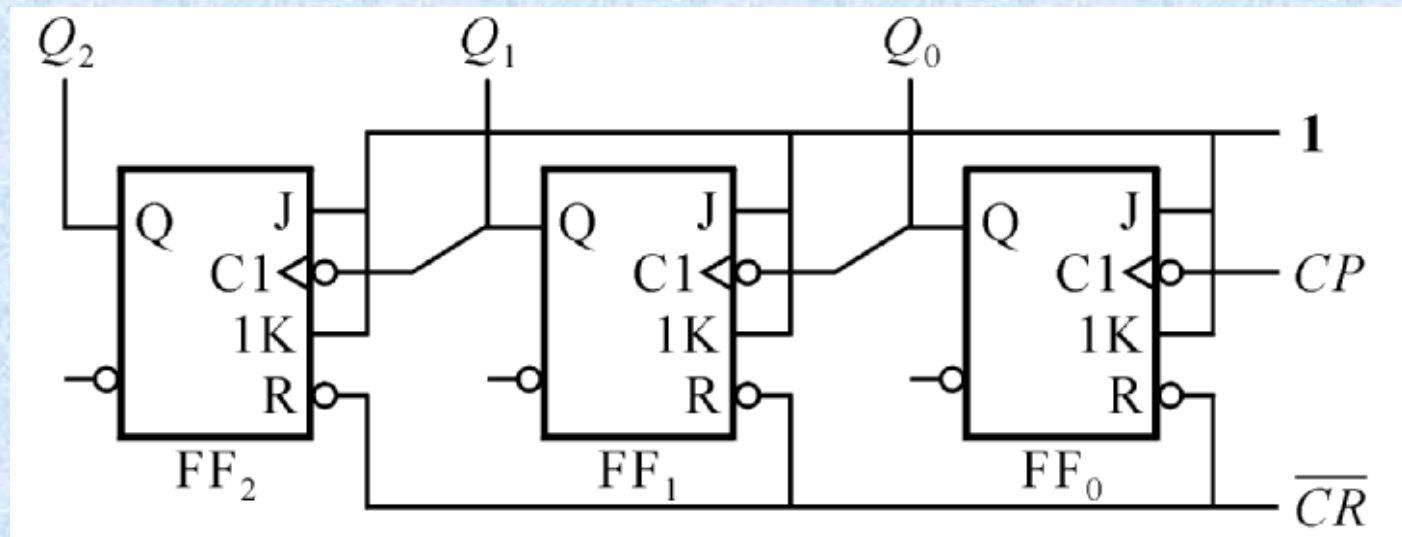
2. 异步二进制计数器分析

① 异步二进制加法计数器

各级触发器的CP脉冲不连在一起，说明各触发器的触发时间不同，翻转也不同步发生。



每个触发器都连接成了计数型（翻转触发器），只要有CP脉冲，触发器状态就翻转。

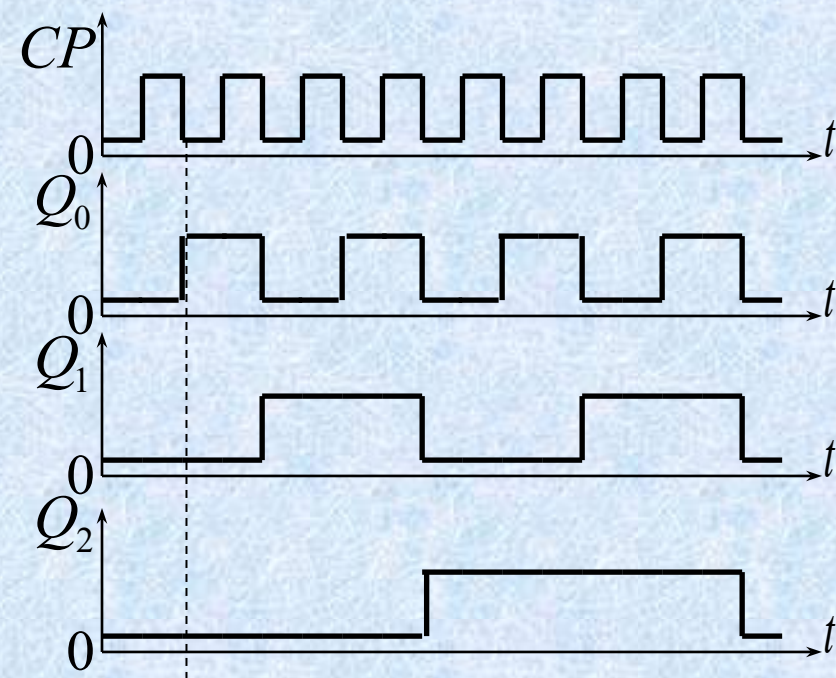


$$CP_0 = CP$$

$$CP_1 = Q_0$$

$$CP_2 = Q_1$$

低位触发器输出作为高位触发器的CP脉冲。



异步工作方式关键是各个触发器之间如何连接问题

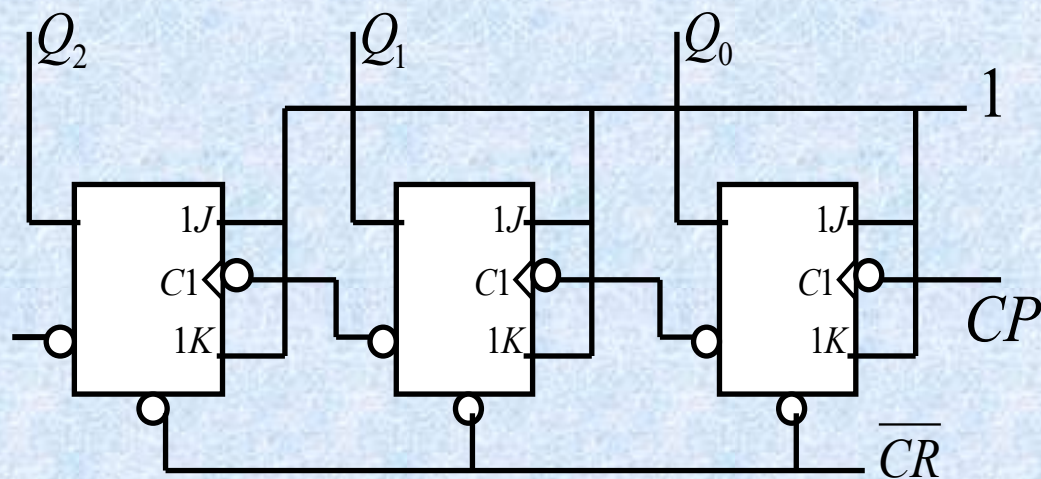
由状态表发现内在的规律!

若用D触发器又该如何连接呢?
当然这些就涉及到“设计”问题

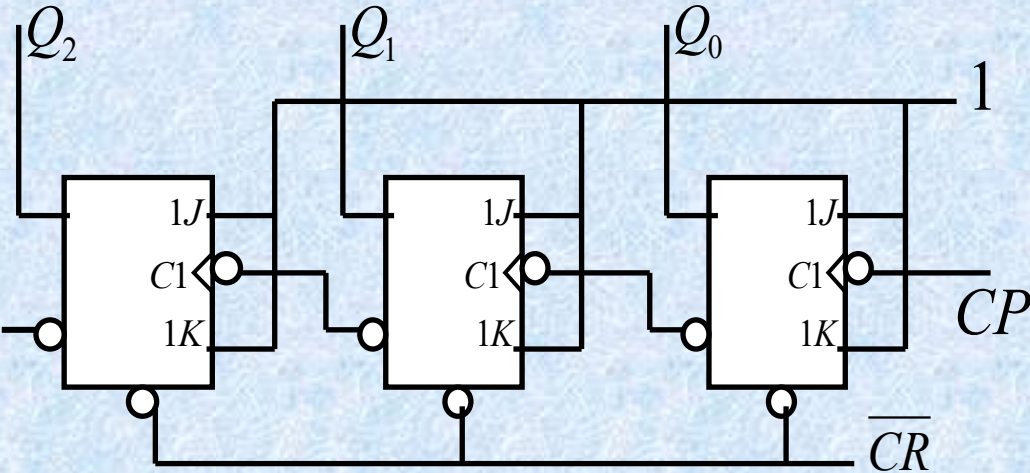
CP顺序	触发器状态		
	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

②异步二进制减法计数器

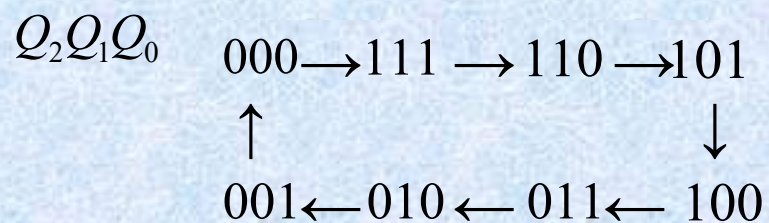
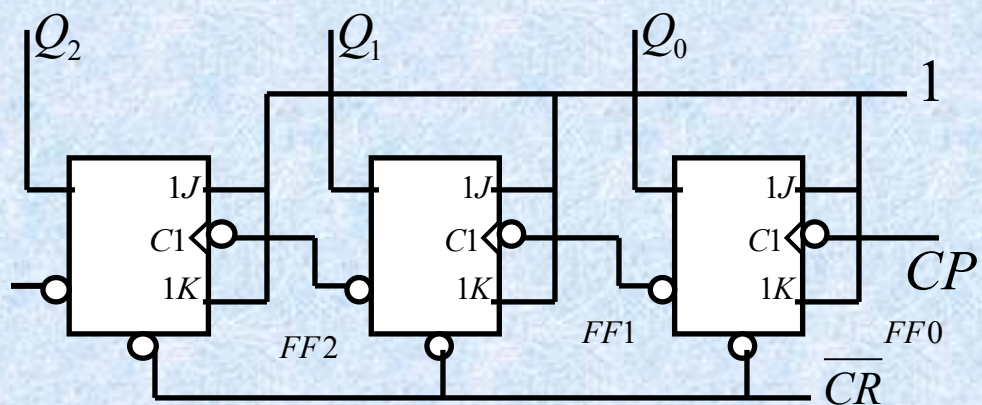
如果高位触发器的CP脉冲来自低位的 \bar{Q} 端时，将变成以下的波形图了，因此，就成了异步二进制减法计数器了。



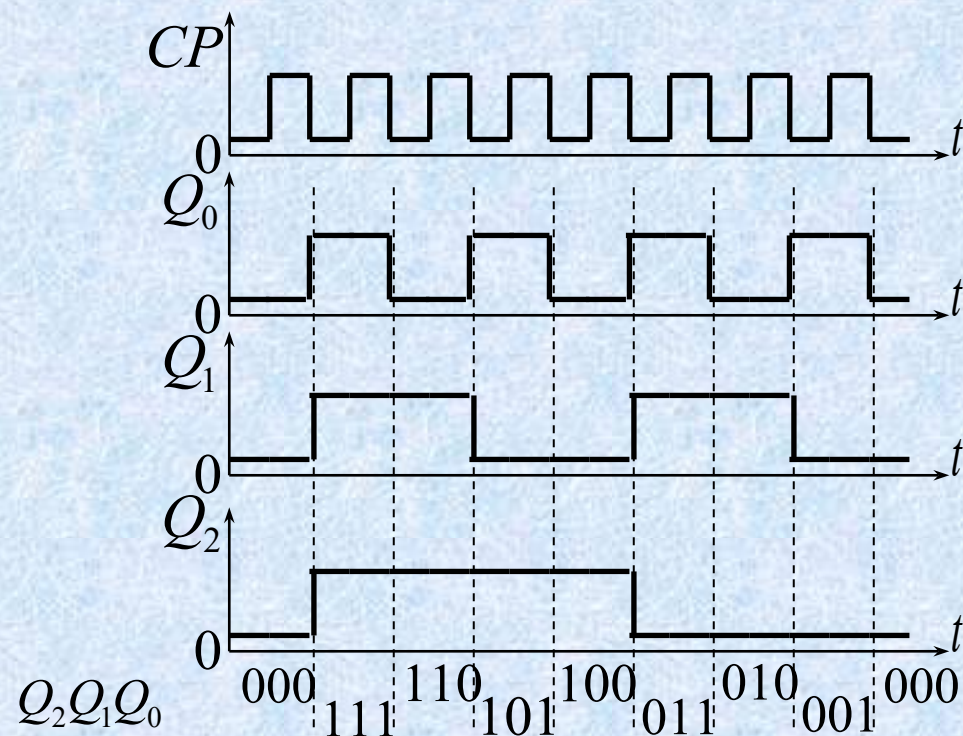
由状态表发现内在的规律!



CP顺序	触发器状态		
	Q_2	Q_1	Q_0
0	0	0	0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0



状态转换图



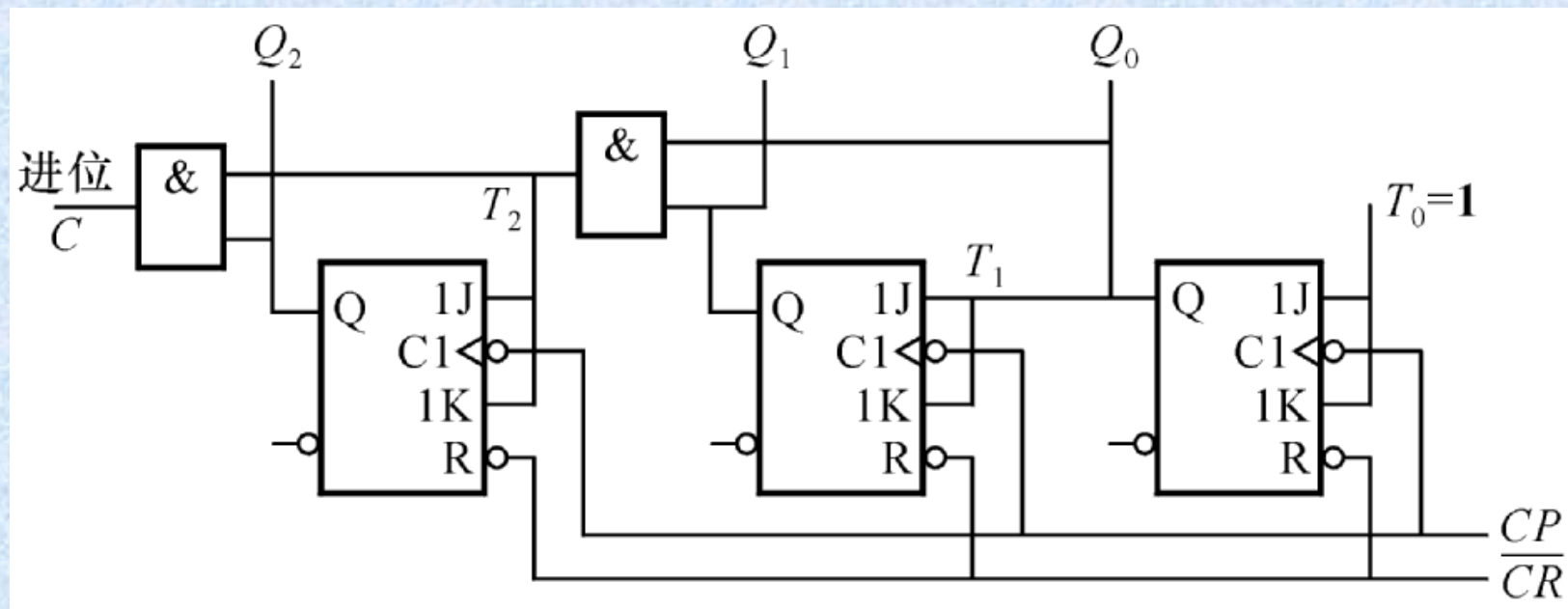
时序图

若用D触发器又该如何连接呢?

异步可逆计数器怎么连接（设计）

3. 二进制计数器的级连结构

同步计数器的计数速度比异步计数器高，影响计数速度的原因是进位连接，串行进位和并行进位



该图是串行进位连接

同步二进制计数器一般由T触发器构成

同步计数器各T端的逻辑关系是：

$$\text{加法 } T_i = Q_{i-1}Q_{i-2} \cdots Q_1Q_0 = \prod_{j=0}^{i-1} Q_j$$

$$\text{减法 } T_i = \overline{Q}_{i-1}\overline{Q}_{i-2} \cdots \overline{Q}_1\overline{Q}_0 = \prod_{j=0}^{i-1} \overline{Q}_j$$

$$\text{可逆 } T_i = X \prod_{j=0}^{i-1} Q_j + \overline{X} \prod_{j=0}^{i-1} \overline{Q}_j \quad \begin{cases} X=1: & \text{加法} \\ X=0: & \text{减法} \end{cases}$$

异步二进制由翻转触发器 (T') 构成;

异步计数器各CP端逻辑关系是:

加法时: $\uparrow CP_i = \overline{Q}_{i-1} \quad \downarrow CP_i = Q_{i-1}$

减法时: $\uparrow CP_i = Q_{i-1} \quad \downarrow CP_i = \overline{Q}_{i-1}$

可逆时:

$$\begin{aligned} \uparrow CP_i &= X\overline{Q}_{i-1} + \overline{X}Q_{i-1} \\ \downarrow CP_i &= XQ_{i-1} + \overline{X}\overline{Q}_{i-1} \end{aligned} \quad \left\{ \begin{array}{l} X=1: \text{加法} \\ X=0: \text{减法} \end{array} \right.$$

4.1.3.2 基本时序电路的设计

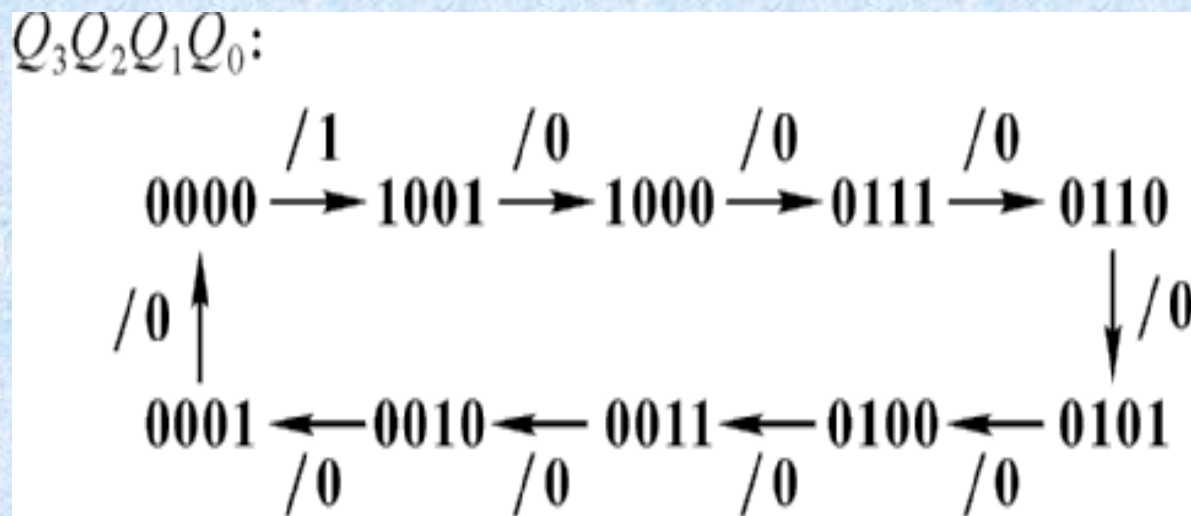
一、同步时序电路的一般设计步骤

- ① 由设计要求或时序图，画出状态转换图，状态用二进制代码进行编码（状态分配）；选好触发器；
- ② 列出状态转换真值表，（输入、输出状态）；
- ③ 以现态和输入为变量，求出各触发器输入的逻辑函数（驱动方程）和输出函数（进、借位）；检查自启动；
- ④ 仔细画出整个计数器的逻辑电路图。

可见，设计过程只有规律性，无技巧性
关键是设计思路要清晰。

[例4.1.6] 用下降沿触发的JK触发器，设计一个同步的按8421编码计数的十进制减法计数器。

解：题目已明确告诉计数编码、触发器等，所以，实际上已告诉了状态转换图



所以，就直接做第二步：列出状态真值表（即1/0关系）

CP	初态 Q^n				次态 Q^{n+1}				$Q^n \rightarrow Q^{n+1}$ 对JK要求				输出 B
	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_3K_3	J_2K_2	J_1K_1	J_0K_0	
0	0	0	0	0	1	0	0	1	$1\times$	$0\times$	$0\times$	$1\times$	1
1	1	0	0	1	1	0	0	0	$\times 0$	$0\times$	$0\times$	$\times 1$	0
2	1	0	0	0	0	1	1	1	$\times 1$	$1\times$	$1\times$	$1\times$	0
3	0	1	1	1	0	1	1	0	$0\times$	$\times 0$	$\times 0$	$\times 1$	0
4	0	1	1	0	0	1	0	1	$0\times$	$\times 0$	$\times 1$	$1\times$	0
5	0	1	0	1	0	1	0	0	$0\times$	$\times 0$	$0\times$	$\times 1$	0
6	0	1	0	0	0	0	1	1	$0\times$	$\times 1$	$1\times$	$1\times$	0
7	0	0	1	1	0	0	1	0	$0\times$	$0\times$	$\times 0$	$\times 1$	0
8	0	0	1	0	0	0	0	1	$0\times$	$0\times$	$\times 1$	$1\times$	0
9	0	0	0	1	0	0	0	0	$0\times$	$0\times$	$0\times$	$\times 1$	0

求J、K和B的函数式：（用卡诺图求，1010至1111转移结果在设计并没有要求，可将这六个状态都当作无关项处理，从而简化电路设计）

$$J_3 \begin{matrix} Q_1^n Q_0^n \\ Q_3^n Q_2^n \end{matrix}$$

	00	01	11	10
00	1	0	0	0
01	0	0	0	0
11	×	×	×	×
10	×	×	×	×

$$K_3 \begin{matrix} Q_1^n Q_0^n \\ Q_3^n Q_2^n \end{matrix}$$

	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	×	×	×	×
10	1	0	×	×

$$\begin{cases} J_3 = \overline{Q_2^n} \overline{Q_1^n} Q_0^n \\ K_3 = Q_0^n \end{cases}$$

$$\begin{cases} J_2 = \overline{Q_3^n} \overline{Q_0^n} \\ K_3 = Q_1^n Q_0^n \end{cases}$$

$$\begin{cases} J_1 = \overline{Q_2^n} Q_0^n + Q_3^n Q_0^n = \overline{Q_3^n} \overline{Q_2^n} Q_0^n \\ K_1 = Q_0^n \end{cases}$$

$$\begin{cases} J_0 = 1 \\ K_0 = 1 \end{cases}$$

$$B = \overline{Q_3^n} \overline{Q_2^n} \overline{Q_1^n} \overline{Q_0^n}$$

另一种方法是由状态真值表直接得到各触发器的次态方程

$$Q_i^{n+1} = f_i(Q_3, Q_2, Q_1, Q_0), i = 0, 1, 2, 3$$

然后通过和 $Q_i^{n+1} = J_i \bar{Q}_i + \bar{K}_i Q_i$ 比对得到

J_i, K_i 逻辑函数关系

		$Q_1 Q_0$			
		00	01	11	10
\bar{Q}_3	00	1	0	0	0
	01	0	0	0	0
Q_3	11	X	X	X	X
	10	0	1	X	X

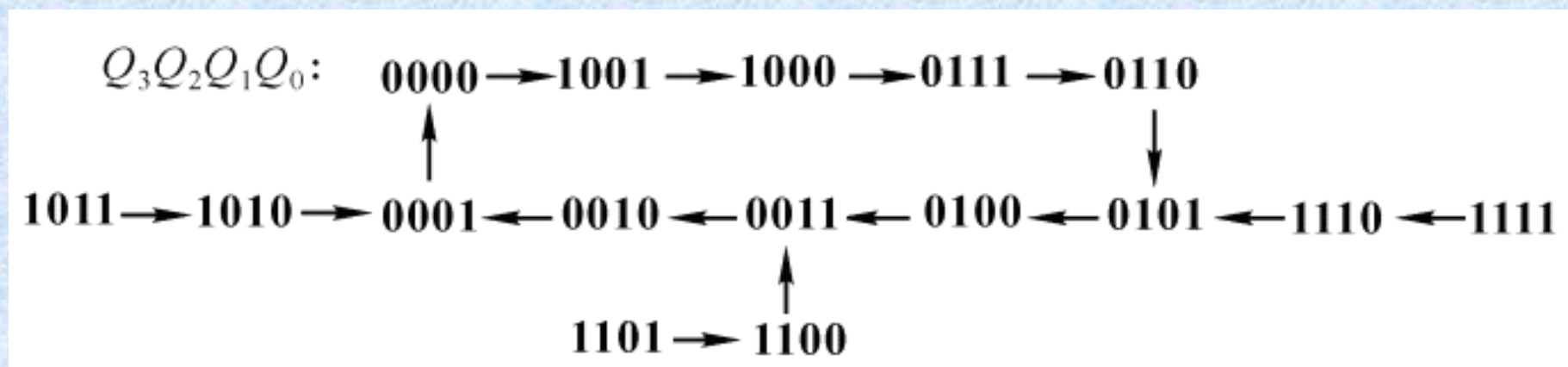
$$Q_3^{n+1} = \bar{Q}_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + Q_3 Q_0$$

可得:

$$\begin{cases} J_3 = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 \\ K_3 = Q_0 \end{cases}$$

设计时利用了六个无关项，完成后必须检查电路能否自启动

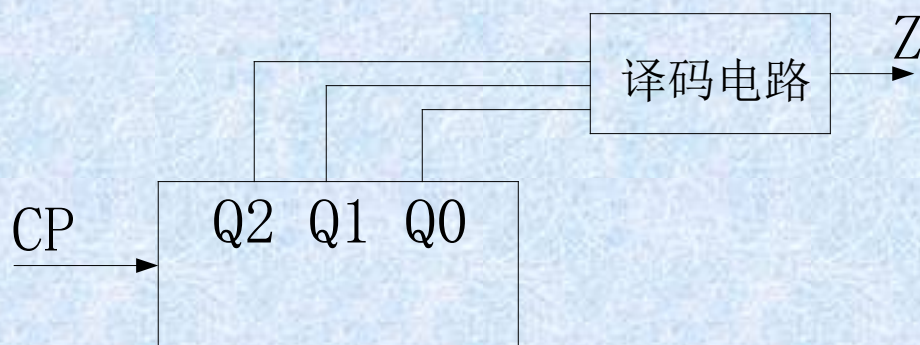
得到实际状态转移图：



满足设计要求

【例】：设计一个信号灯控制器，要求亮3秒，灭4秒，循环， $CP = 1S$

◆ 可以设计为一个7进制的计数器



在状态真值表中利用无关项，最后设计后要检验能否有自启动，即应回到有效项去。

CP	初态			次态			Z
	Q2	Q1	Q0	Q2	Q1	Q0	
↓	0	0	0	0	0	1	1
	0	0	1	0	1	0	1
	0	1	0	0	1	1	1
		⋮			⋮		0
	1	1	1	×	×	×	

按照时序电路输出方式，时序电路可分为Mealy（米里）型电路和Moore（莫尔）型电路

Moore 型电路输出仅与状态相关，在时钟有效沿才可能发生变化，抗干扰能力更强；

Mealy电路的输出在非时钟有效沿也会随输入信号的变化而变化，输出可以直接受外部输入控制，响应速度快，可以处理异常情况。

【例4.1.7】序列信号检测电路通过时钟检测输入信号的电平状态，试用正边沿D触发器设计一个110序列检测电路。

[Mealy]型同步时序逻辑电路

解：对于110序列检测电路，当连续三个时钟检测到输入信号的状态分别是1、1、0时，输出一个高电平信号。设检测电路的外部输入是序列脉冲信号X，检测结果的输出信号是Z。

例如： 输入X串行信号： 101100111010
 输出Z相应信号： 000010000100

110序列检测电路可能的状态应该是：

初始状态S0：输入序列信号没有出现过“1”以前的状态；

状态S1：输入序列信号出现过一个“1”后的状态；

状态S2：输入序列信号出现过连续两个“1”后的状态；

状态S3：输入序列信号出现过连续两个“1”后，接着又出现“0”的状态，即出现了110序列信号。

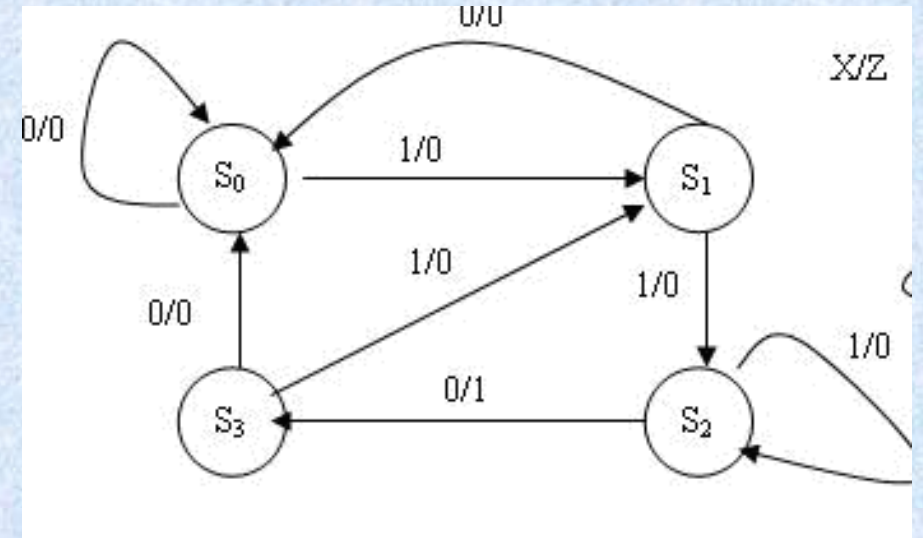
在原始状态状态转移图中，各状态的转移情况为：

$$S_0 \begin{cases} X=0 \rightarrow S_0 \\ X=1 \rightarrow S_1 \end{cases}$$

$$S_1 \begin{cases} X=0 \rightarrow S_0 \\ X=1 \rightarrow S_2 \end{cases}$$

$$S_2 \begin{cases} X=0 \rightarrow S_3 \\ X=1 \rightarrow S_2 \end{cases}$$

$$S_3 \begin{cases} X=0 \rightarrow S_0 \\ X=1 \rightarrow S_1 \end{cases}$$



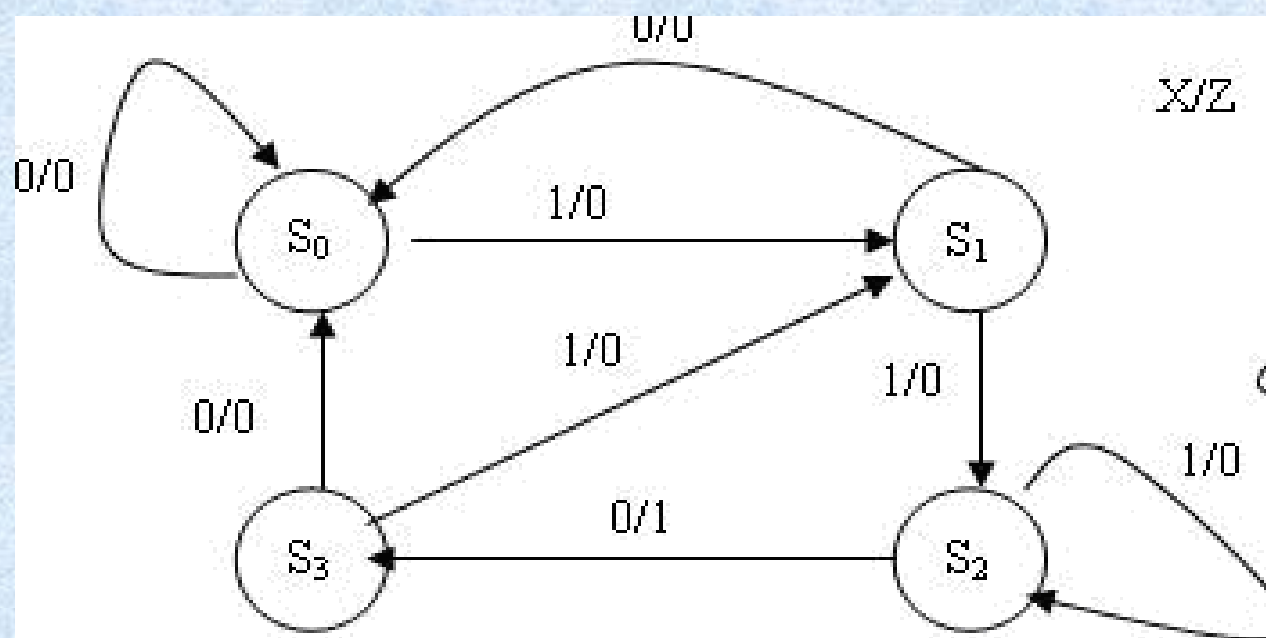
$$S_0 \begin{cases} X=0 \rightarrow S_0 \\ X=1 \rightarrow S_1 \end{cases}$$

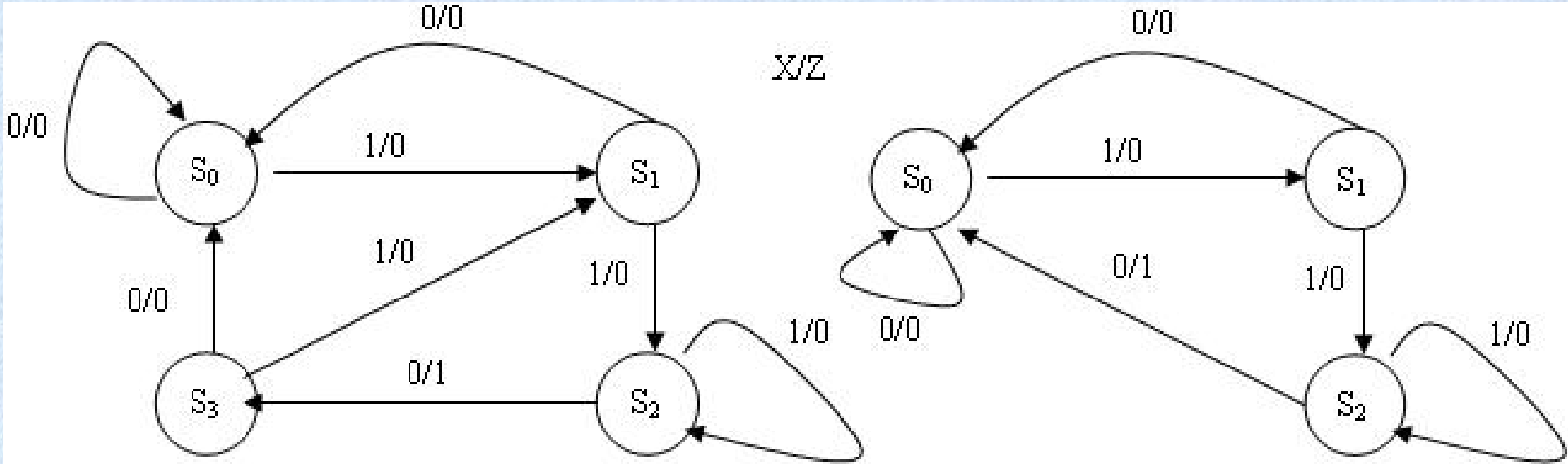
$$S_1 \begin{cases} X=0 \rightarrow S_0 \\ X=1 \rightarrow S_2 \end{cases}$$

$$S_2 \begin{cases} X=0 \rightarrow S_3 \\ X=1 \rightarrow S_2 \end{cases}$$

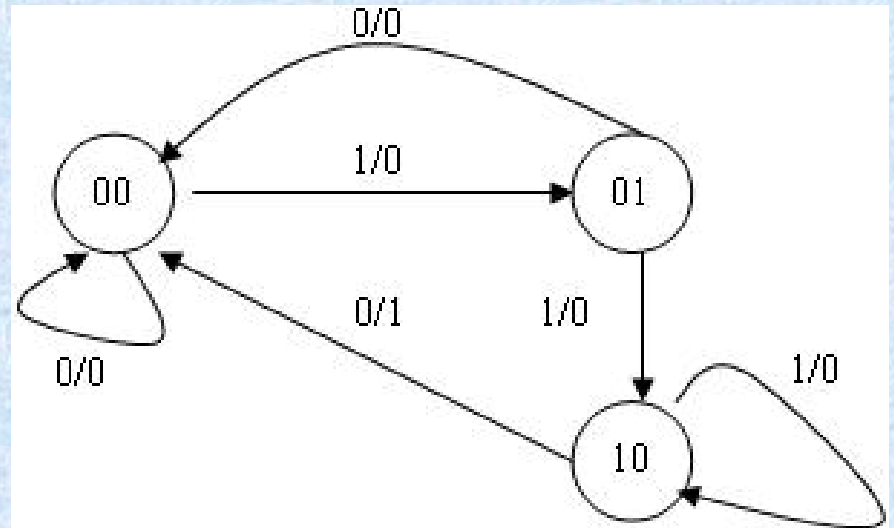
$$S_3 \begin{cases} X=0 \rightarrow S_0 \\ X=1 \rightarrow S_1 \end{cases}$$

S0和S3在输入相同时，其转移到的次态也相同，所以S3状态与S0状态等价。状态S3可以被简化掉



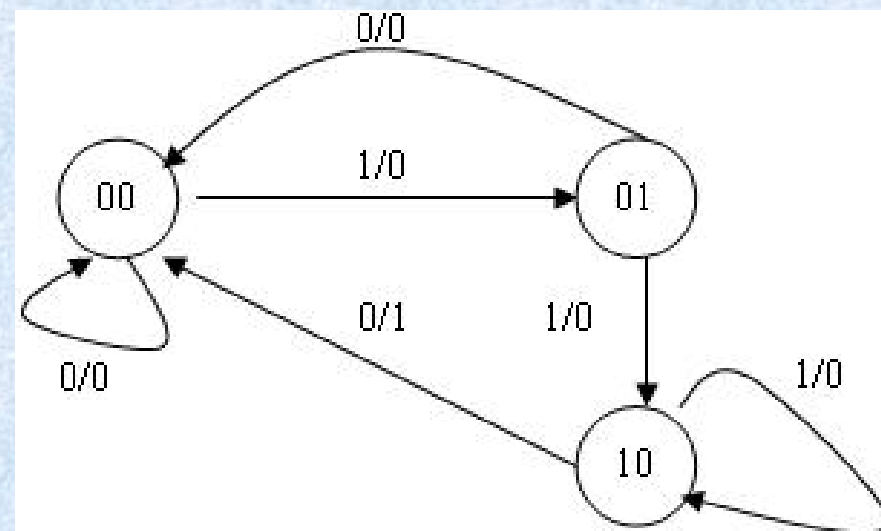


简单按自然顺序进行状态分配，设 $S_0=00$ ， $S_1=01$ ， $S_2=10$ ，
两只触发器的状态分别是 Q_1, Q_0



状态转移图转化成的状态真值表

CP	X	Q_1	Q_0	Q_1^{n+1}	Q_0^{n+1}	Z
1	0	0	0	0	0	0
2	0	0	1	0	0	0
3	0	1	0	0	0	1
4	0	1	1	X	X	X
5	1	0	0	0	1	0
6	1	0	1	1	0	0
7	1	1	0	1	0	0
8	1	1	1	X	X	X

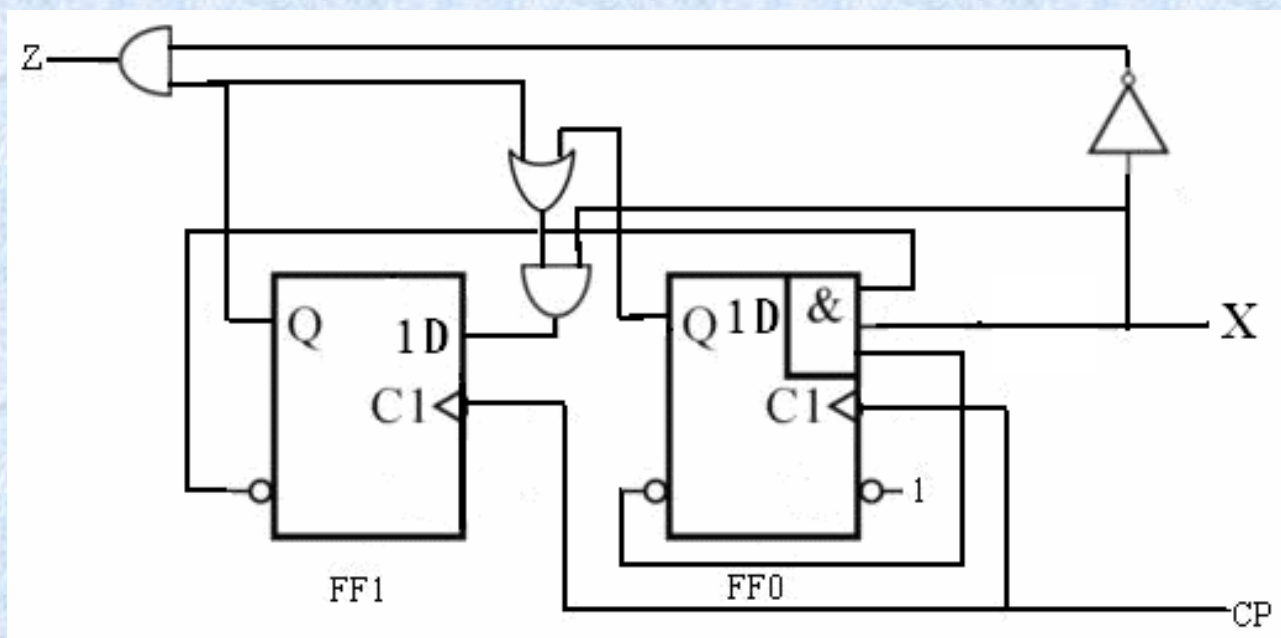


由表得 Q_1^{n+1} Q_0^{n+1} 和输出 Z 的卡诺图

化简、并和D触发器的特征方程比较可得：

$$Q_1^{n+1} = Q_1X + Q_0X, D_1 = Q_1X + Q_0X$$

$$Q_0^{n+1} = \overline{Q_1^n Q_0^n} X \quad D_0 = \overline{Q_1^n Q_0^n} X \quad Z = Q_1 \overline{X}$$



最后进行自启动检查

◆ 【题4.18】 可自行分析

4.1.4 常见逻辑电路功能分析与设计

一、编码器和译码器

1、编码器

(1) 基本编码器

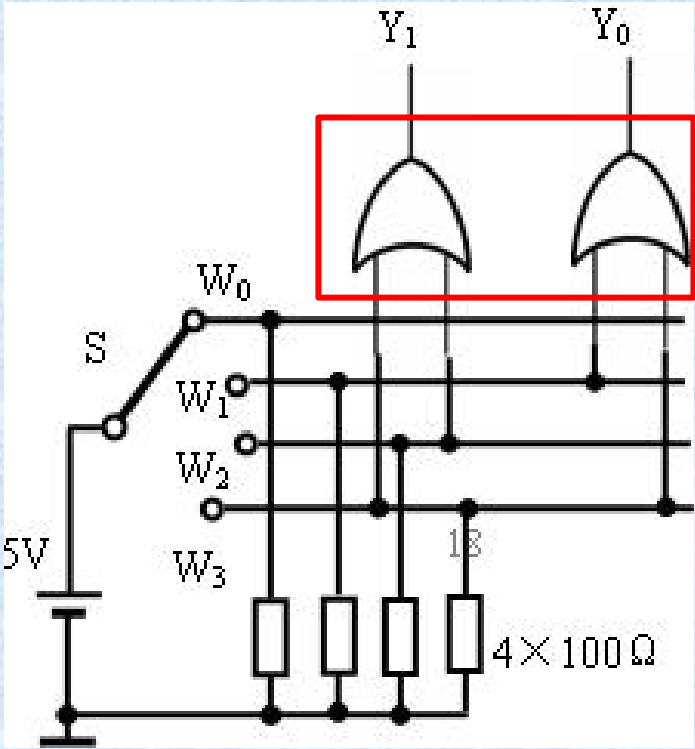
将一个特定对象用一组二进制码来表示。

如：身份证编码、电话编码等

(输入多，输出少)

将4个对象分别编制成二位二进制代码时，该编码器的真值表：

4 线—2 线编码器真值表					
编 码 器 输 入				二 位 码 输 出	
W_0	W_1	W_2	W_3	Y_1	Y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1



4 线—2 线编码器真值表

编 码 器 输 入				二 位 码 输 出	
W_0	W_1	W_2	W_3	Y_1	Y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

如何设计？

输出逻辑函数：

$$Y_1 = \overline{W}_0 \overline{W}_1 W_2 \overline{W}_3 + \overline{W}_0 \overline{W}_1 \overline{W}_2 W_3$$

$$Y_0 = \overline{W}_0 W_1 \overline{W}_2 \overline{W}_3 + \overline{W}_0 \overline{W}_1 \overline{W}_2 W_3$$

注意到：由于输入和输出具有严格的一一对应关系，任何时候都不允许有两个或两个以上的输入对象存在，否则将会产生编码混乱。所以式子可以**利用约束条件进行化简**

$$\begin{aligned} Y_1 &= \overline{W_0} \overline{W_1} W_2 \overline{W_3} + \overline{W_0} \overline{W_1} \overline{W_2} W_3 = \overline{W_0} \overline{W_1} W_2 \overline{W_3} + W_0 W_2 + W_1 W_2 + W_3 W_2 \\ &+ \overline{W_0} \overline{W_1} \overline{W_2} W_3 + W_0 W_3 + W_1 W_3 + W_2 W_3 = \overline{W_0} \overline{W_1} \overline{W_3} W_2 + (W_0 + W_1 + W_3) W_2 \\ &+ \overline{W_0} \overline{W_1} \overline{W_2} W_3 + (W_0 + W_1 + W_2) W_3 = W_2 + W_3 \end{aligned}$$

$$Y_1 = W_2 + W_3$$

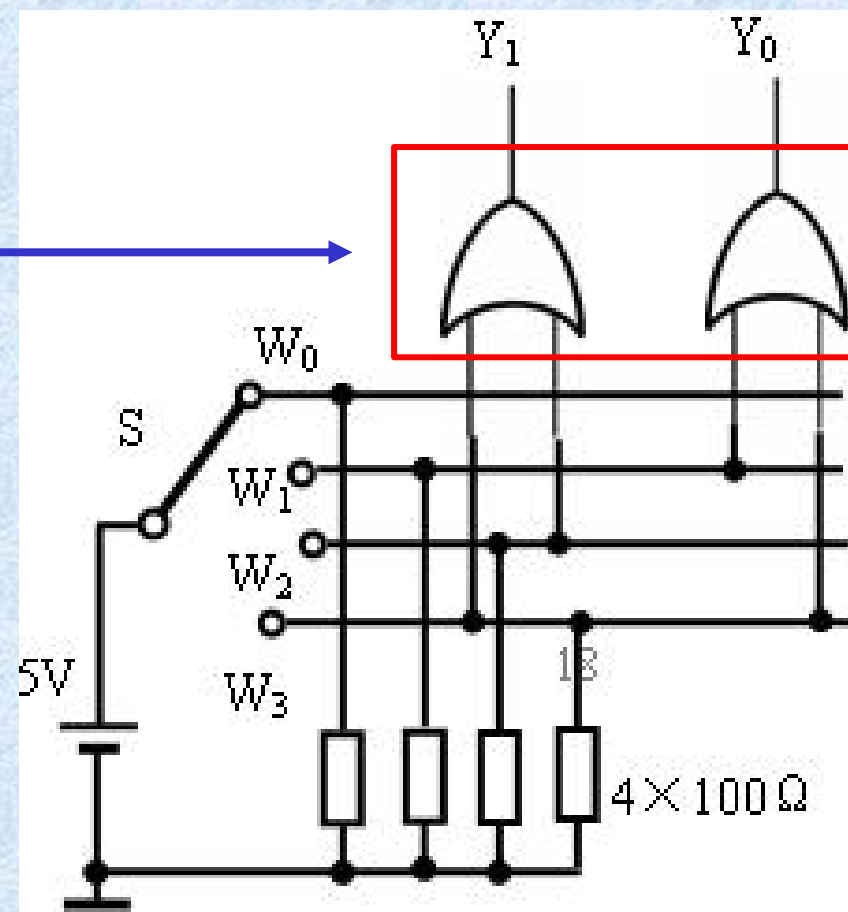
$$Y_0 = W_1 + W_3$$

4 线—2 线编码器真值表

编 码 器 输 入				二 位 码 输 出	
W_0	W_1	W_2	W_3	Y_1	Y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

$$Y_1 = W_2 + W_3$$

$$Y_0 = W_1 + W_3$$

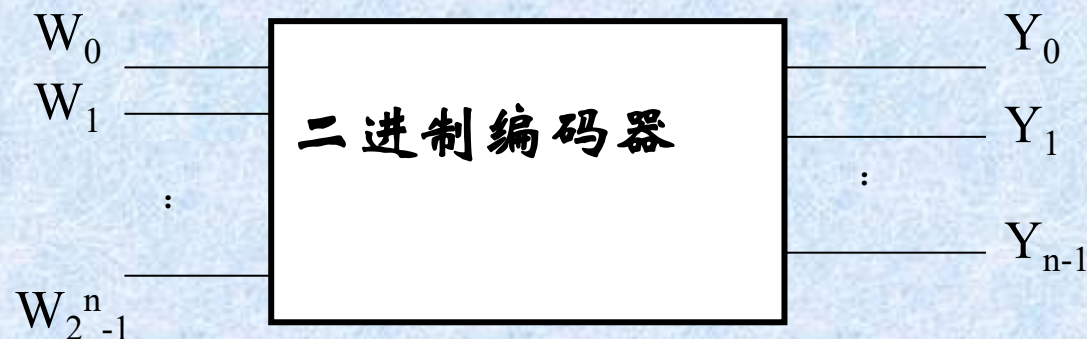


组合逻辑电路的设计方法：

- (1) 根据题意或给定功能找出输入和输出逻辑变量；
- (2) 列出真值表；
- (3) 求出各个输出的最简与—或表达式（建议用卡诺图法）；
- (4) 用规定的逻辑门画出整个逻辑电路图。

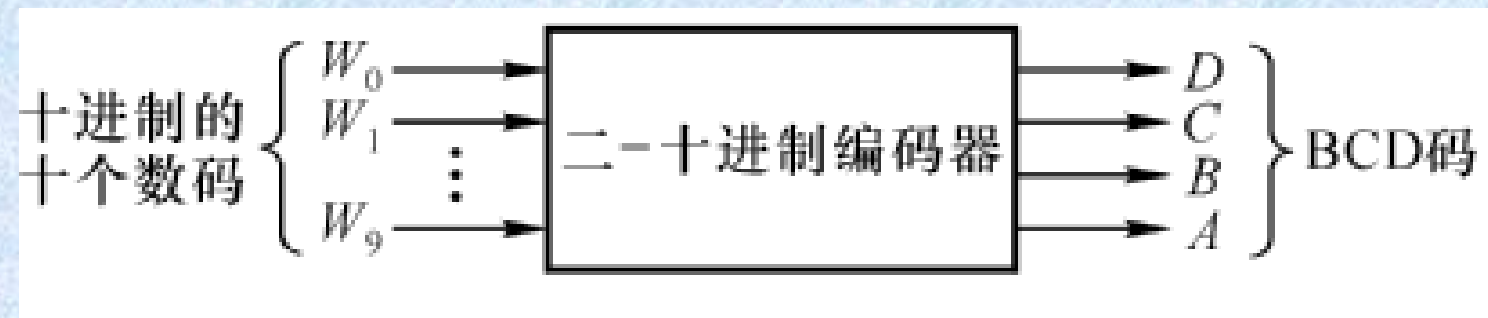
(2) 二进制编码器

将 2^n 个特定对象编制成 n 位二进制代码的一种组合逻辑电路。它在数字系统中应用的非常普遍，例如：4线—2线(4/2)编码器，8线—3线(8/3)编码器，16线—4线(16/4)编码器等。



(3) 二—十进制编码器

它是将十进制的0~9十个数字，用一组4位的二—十进制代（BCD码）表示。



(4) 优先编码器(Priority Encoder)

这种编码器允许同时输入二个或二个以上的输入信号，但编码器只对其中优先权最高的待编码对象实施编码。编码对象的优先权高低可以在设计时预先规定。

例如，日常生活中的应急电话号码处理，是有优先权的。问询电话、火警电话、工作电话等。

W_3	W_2	W_1	W_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	×	0	1
0	1	×	×	1	0
1	×	×	×	1	1

$$Y_1 = \overline{W}_3 W_2 + W_3 = W_2 + W_3$$

$$Y_0 = \overline{W}_3 \overline{W}_2 W_1 + W_3 = \overline{W}_2 W_1 + W_3$$

例：设计一个二—十进制优先编码器， W_1, \dots, W_9 代表十进制的九个数，**“0”表示有输入，“1”表示无输入**。采用大数优先的原则，即 W_9 优先权最高， W_1 优先权最低。 Y_3, Y_2, Y_1, Y_0 表示输出四位代码，**并用反码表示**。如此约定后列出的真值表如表所示：

编 码 输 入										BCD码输出			
\overline{W}_9	\overline{W}_8	\overline{W}_7	\overline{W}_6	\overline{W}_5	\overline{W}_4	\overline{W}_3	\overline{W}_2	\overline{W}_1		\overline{Y}_3	\overline{Y}_2	\overline{Y}_1	\overline{Y}_0
1	1	1	1	1	1	1	1	1		1	1	1	1
1	1	1	1	1	1	1	1	0		1	1	1	0
1	1	1	1	1	1	1	0	×		1	1	0	1
1	1	1	1	1	1	0	×	×		1	1	0	0
1	1	1	1	1	0	×	×	×		1	0	1	1
1	1	1	1	0	×	×	×	×		1	0	1	0
1	1	1	0	×	×	×	×	×		1	0	0	1
1	1	0	×	×	×	×	×	×		1	0	0	0
1	0	×	×	×	×	×	×	×		0	1	1	1
0	×	×	×	×	×	×	×	×		0	1	1	0

可以从真值表直接得到逻辑函数表达式：

$$\overline{\overline{Y}}_3 = \overline{\overline{W}}_9 \overline{\overline{W}}_8 + \overline{\overline{W}}_9 = W_9 + \overline{W}_9 W_8 = W_9 + W_8$$

$$\overline{Y_3} = \overline{W_8 + W_9}$$

$$\overline{Y_2} = \overline{\overline{W_9} \overline{W_8} (W_4 + W_5 + W_6 + W_7)}$$

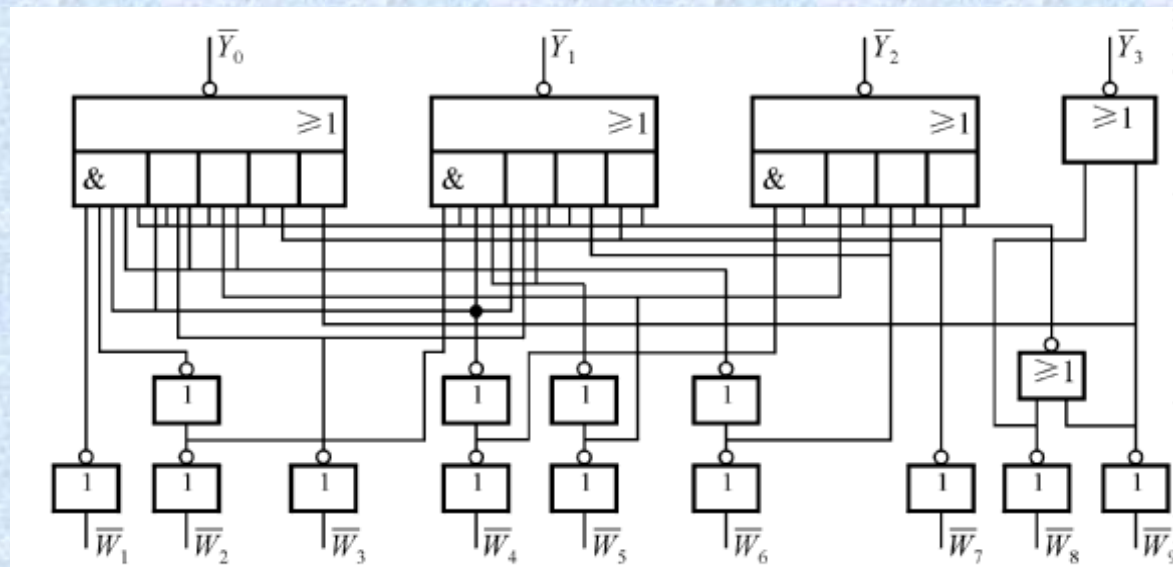
$$= \overline{(\overline{W_9} \overline{W_8}) W_4 + \overline{W_9} \overline{W_8} W_5 + \overline{W_9} \overline{W_8} W_6 + \overline{W_9} \overline{W_8} W_7}$$

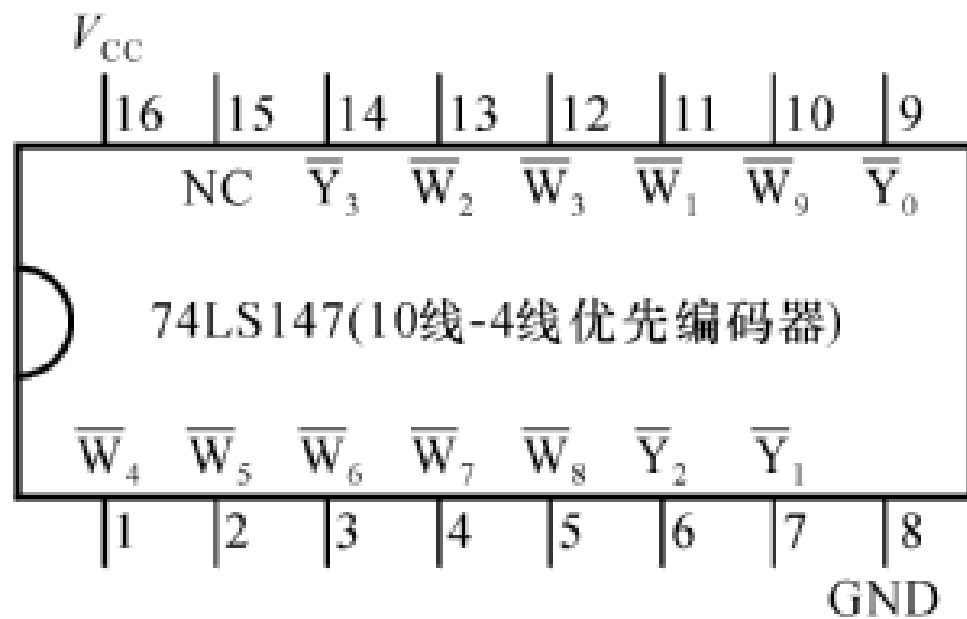
$$\overline{Y_1} = \overline{(\overline{W_9} \overline{W_8}) (\overline{W_5} \overline{W_4} W_3 + \overline{W_5} \overline{W_4} W_2 + W_7 + W_6)}$$

$$\overline{Y_0} = \overline{\overline{W_9} \overline{W_8} (W_7 + W_6 W_5 + \overline{W_6} \overline{W_4} W_3 + \overline{W_6} \overline{W_4} W_2 W_1) + W_9}$$

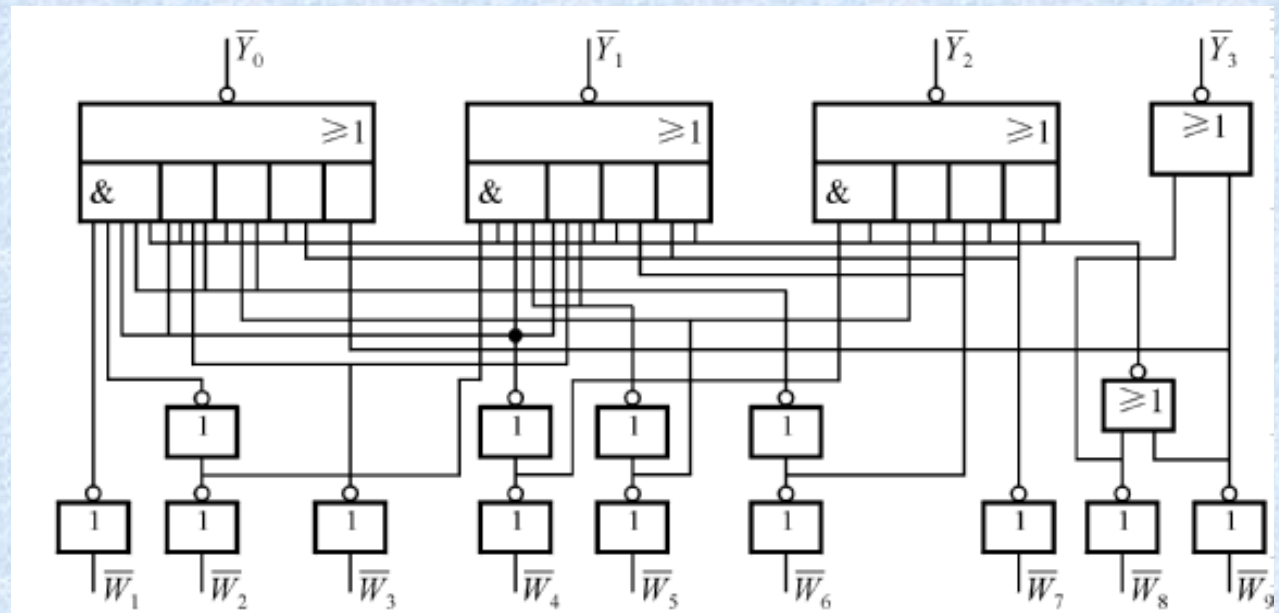
画出的逻辑电路图如图所示

10线—4线的集成二—十进制优先编码器内部逻辑电路图。它就是中规模集成优先编码器**74LS147**





74LS147 BCD 优先编码器



强调

组合逻辑电路的**设计**方法：

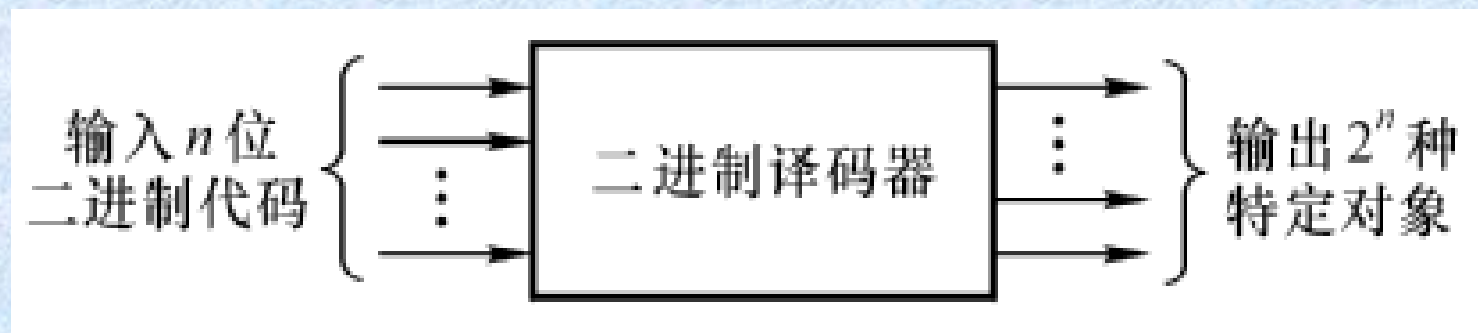
- (1) 根据题意或给定功能找出输入和输出逻辑变量；
- (2) 列出真值表；
- (3) 求出各个输出的最简与—或表达式（建议用卡诺图法）；
- (4) 用规定的逻辑门画出整个逻辑电路图。

2、译码器

它是编码电路的反过程。即：将二进制代码所代表的特定对象还原出来的电路。根据还原（翻译）对象的不同，分为二进制译码器（变量译码器）和二—十进制译码器两种。

(1) 二进制基本译码器

n 位二进制代码代表的 2^n 种特定对象还原出来的电路。

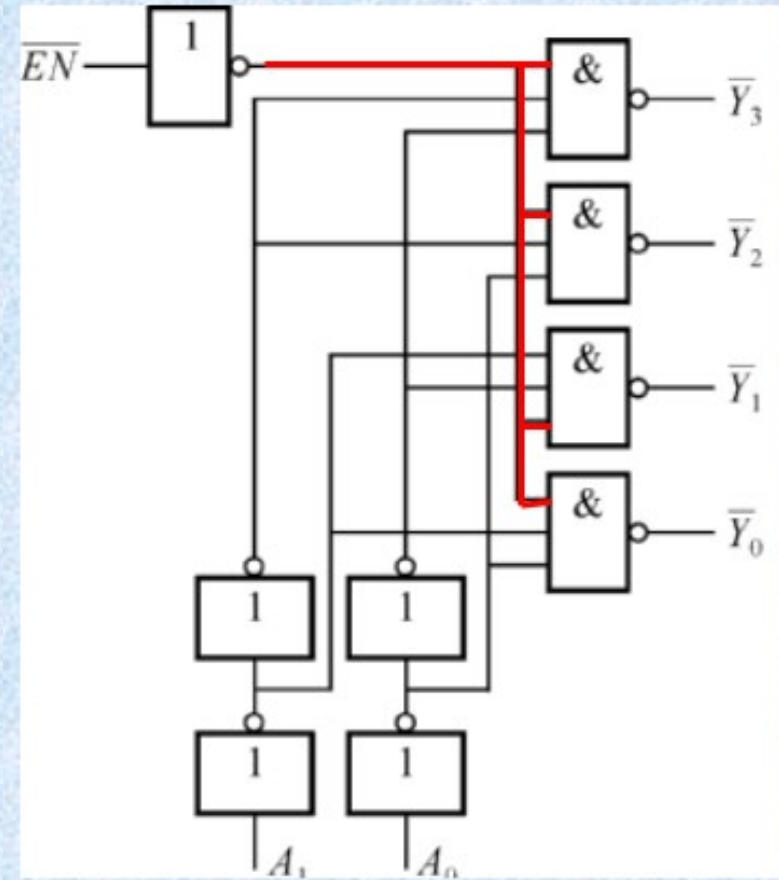


电路的输入是 n 位二进制代码，输出为 2^n 种特定对象，如2/4，3/8，4/16等译码器。

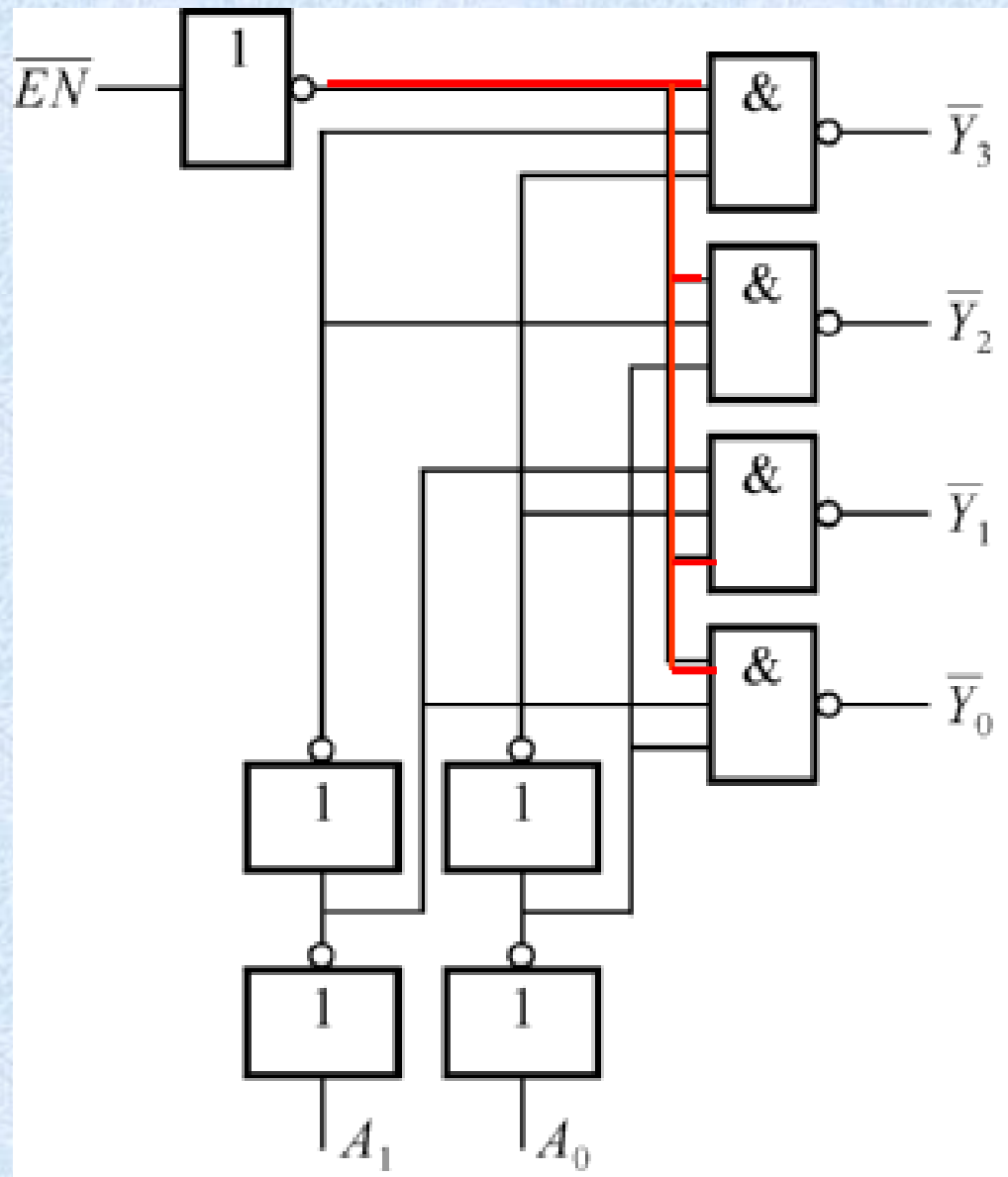
电路是一个2线/4线译码电路，输入二位代码 A_1, A_0 ，输出是4个特定对象 $\overline{Y}_3, \overline{Y}_2, \overline{Y}_1, \overline{Y}_0$ ， \overline{EN} 是译码还是

不译码的使能控制端。当使能端

$\overline{EN}=0$ 时，各输出逻辑函数式为：



$$\overline{Y_3} = \overline{A_1 A_0}, \overline{Y_2} = \overline{A_1 A_0}, \overline{Y_1} = \overline{A_1 A_0}, \overline{Y_0} = \overline{A_1 A_0}$$



$$\overline{EN} = 0$$

$$\overline{Y}_3 = \overline{A_1 A_0}$$

$$\overline{Y}_2 = \overline{A_1 \overline{A_0}}$$

$$\overline{Y}_1 = \overline{\overline{A_1} A_0}$$

$$\overline{Y}_0 = \overline{\overline{A_1} \overline{A_0}}$$

给予使能控制端以及输入变量的各种取值后，得到各输出的结果，列成真值表如表所示：

使能控制	输入代码		译码器输出			
\overline{EN}	A_1	A_0	\overline{Y}_3	\overline{Y}_2	\overline{Y}_1	\overline{Y}_0
1	×	×	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

$$\overline{EN} = 0$$

$$\overline{Y}_3 = \overline{A_1 A_0},$$

$$\overline{Y}_2 = \overline{\overline{A_1 A_0}},$$

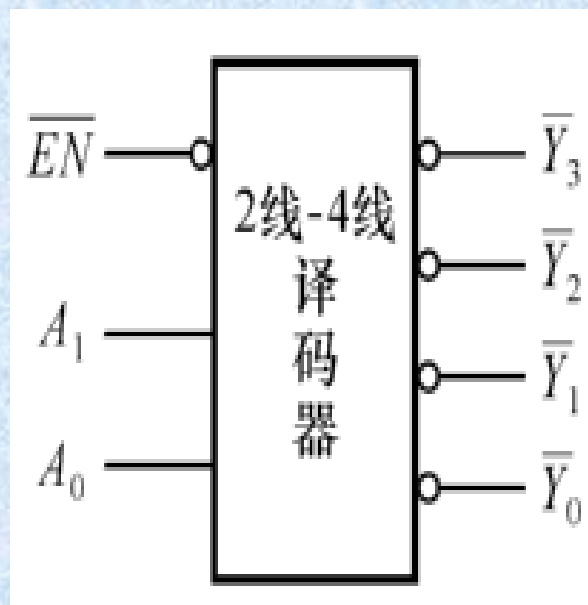
$$\overline{Y}_1 = \overline{\overline{A_1 A_0}},$$

$$\overline{Y}_0 = \overline{\overline{A_1 A_0}}$$

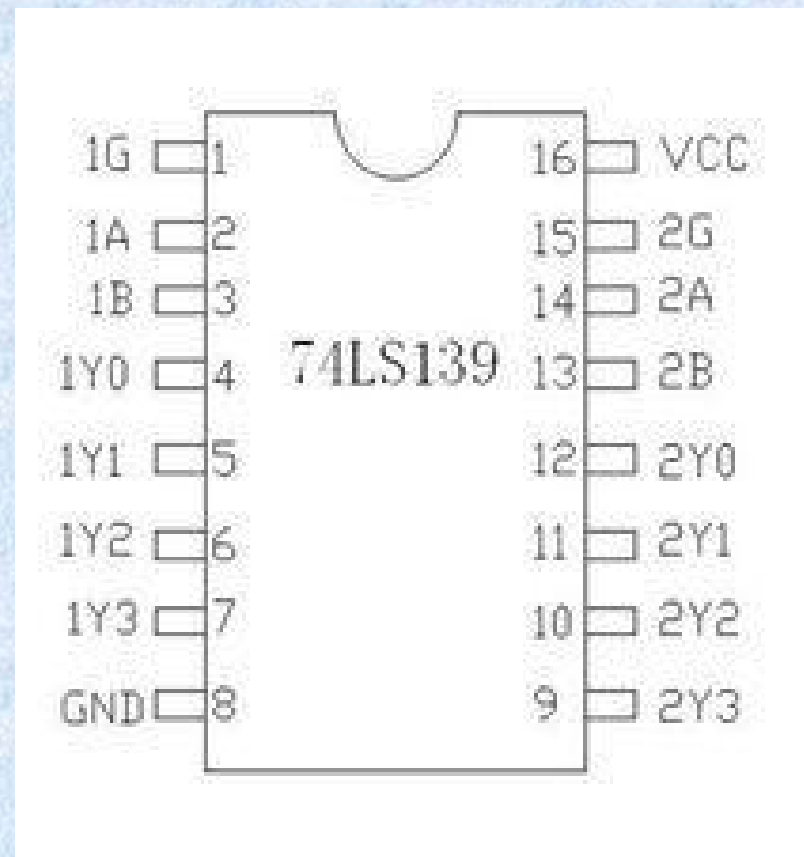
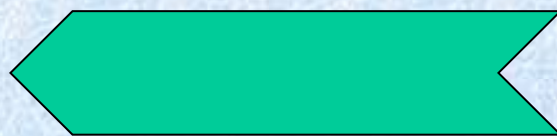
正逻辑真值表又如何？

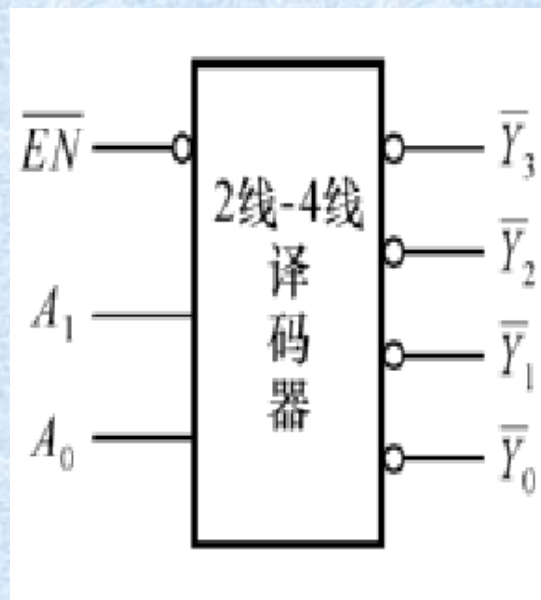
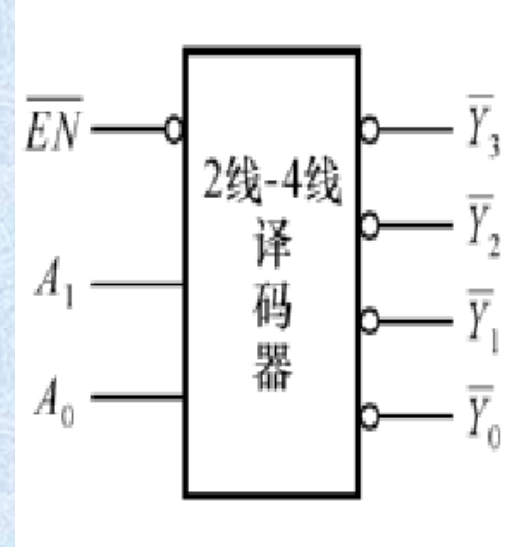
上述情况可知 (1) 一组代码和特定输出对象是一一对应的关系； (2) 每一个输出的逻辑函数是输入代码的一个最小项。（即标准与项）这一结论后面有用。

上述的2/4译码器通常用简化框图画出。



译码器的简化
逻辑图表示



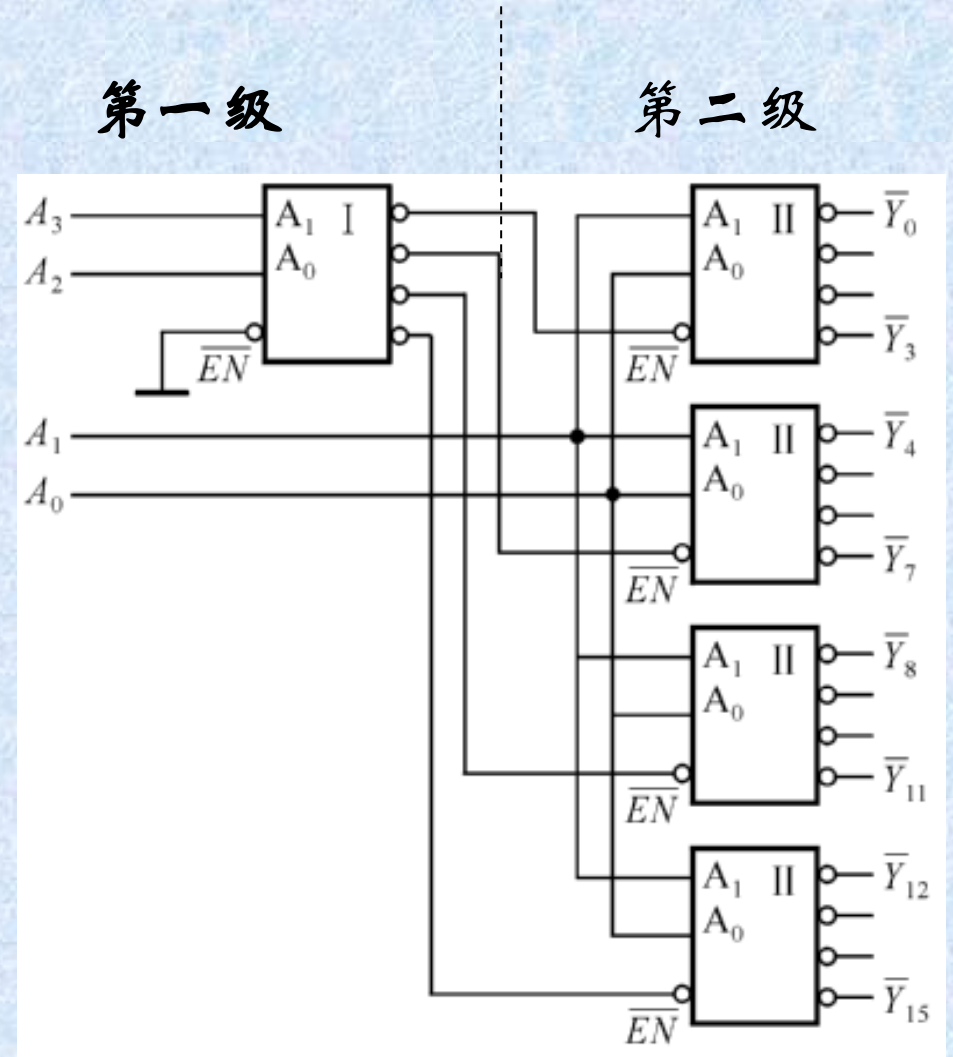


2、 中规模集成二进制译码器

它的种类很多，但主要是2/4，3/8，4/16等译码器。它们可以用集成2/4或3/8译码器连接而成。如用5片2/4组成4线/16线译码器，如图所示：

它是采用**分级译码**的方法实现的。

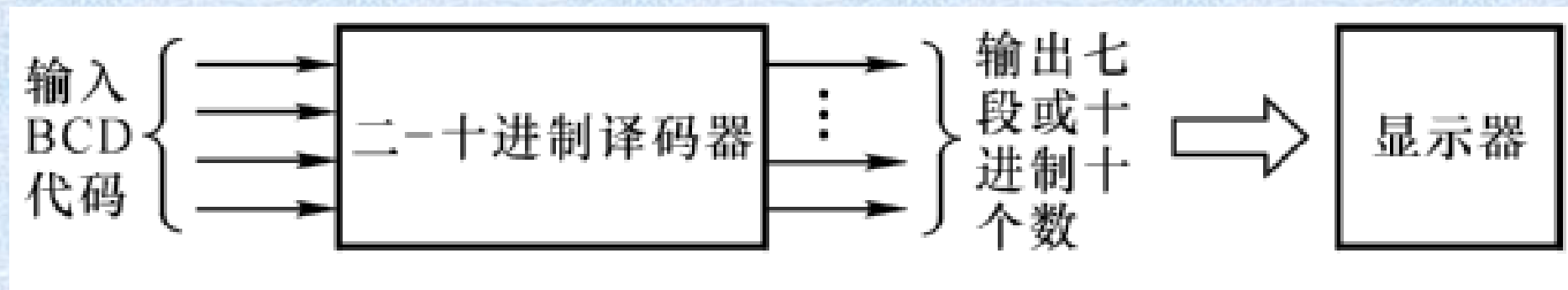
◆ 常用于微机和存储器中的地址译码



3. 二—十进制译码器

将输入BCD码翻译成十进制数码输出（十个数码）的组合逻辑电路，译码后的结果用显示器显示出来，所以，又称**显示译码器**和**码制变换译码器**。

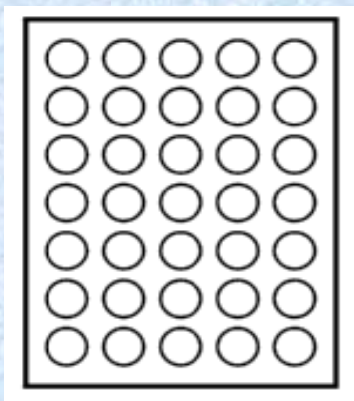
它的组成框图如图所示：



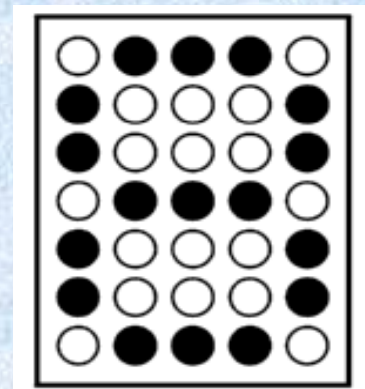
1. 显示器简介

显示器有点阵式和分段式两种（也可按器件分：半导体和液晶二类）

(1) 点阵式如图所示

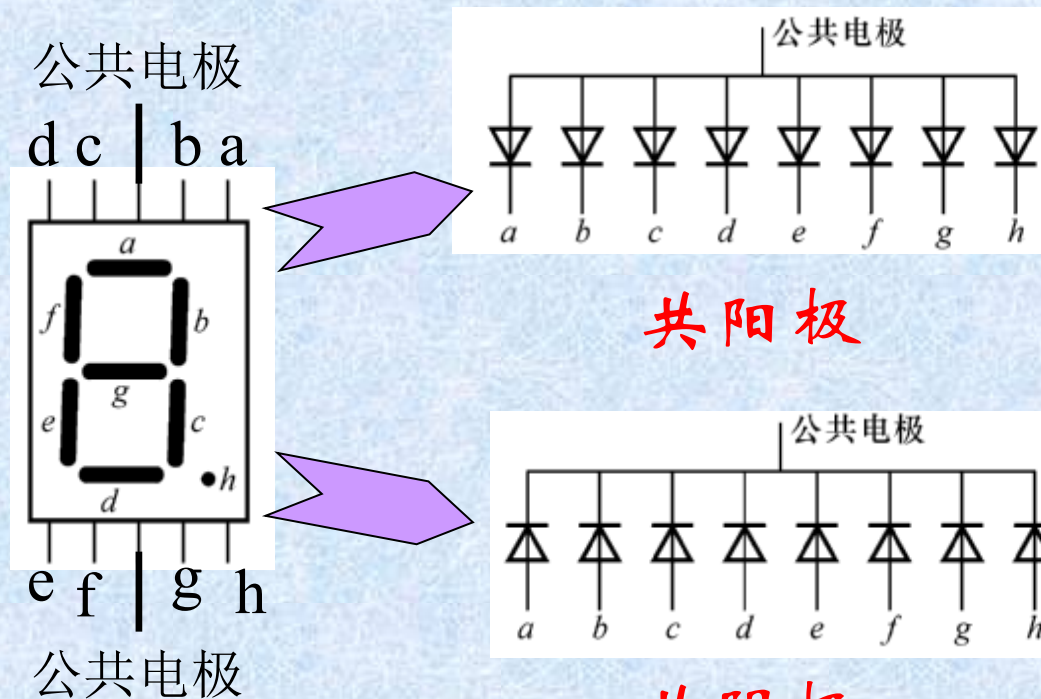


要点亮某一个字形时，只要点亮这些字形的点就行。用的最多的地方是车站，机场，马路上的广告牌等。

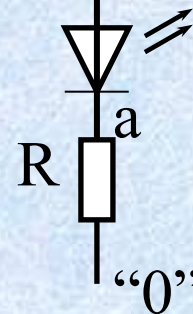


(2) 分段显示器

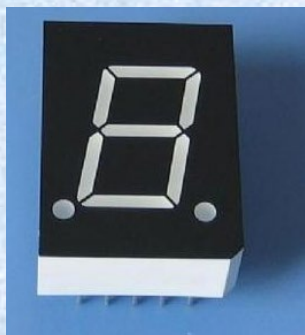
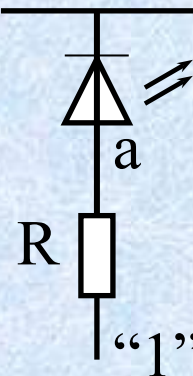
它可以是液晶分段式和半导体分段式两种，图示是八段半导体数码管显示器。



公共极接 $V_{CC}=5V$

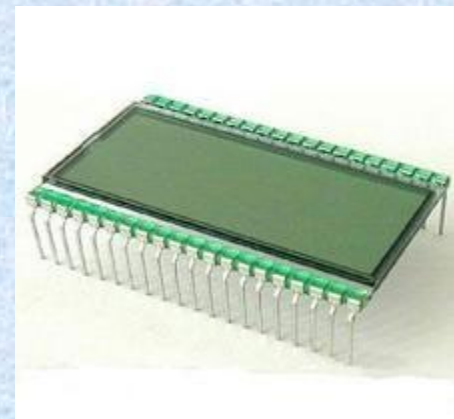


公共极接地



LED发光二极管显示器

显示字形时的基本笔划



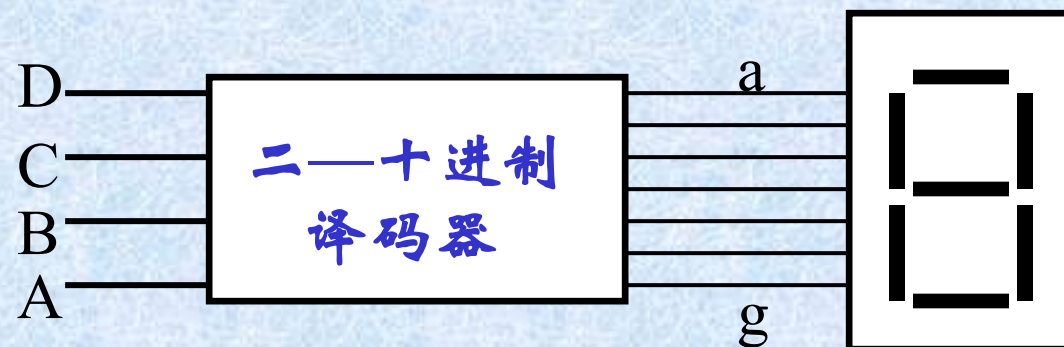
另一种分段式液晶（LCD）显示器低电源、低功耗，目前使用日益普遍。它利用流动性有机化合物的奇特光学特性而发光。



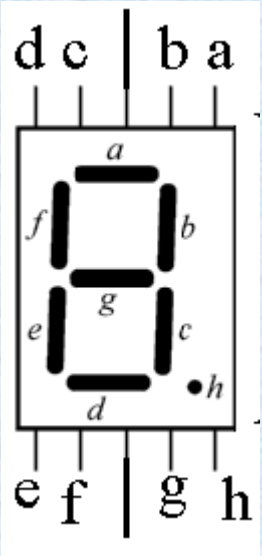
二—十进制译码器举例

【例】试用非门和或非门设计一个8421BCD码输入的驱动共阴极七段半导体数码管的二—十进制译码器。

解：由于显示器为七段半导体数码管，所以译码器的输出为七个输出，四位BCD码输入，设计电路如下框图所示：



由于是**共阴极**，译码器输出应为高电平才能点亮某段数码管



输入8421代码				输出对应段亮暗							字形
D	C	B	A	a	b	c	d	e	f	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

用卡诺图化简，得出七段的逻辑函数式，由于用或非门且驱动共阴极，所以用包围“0”格，求或与式的最简原函数。

$$\overline{a} = \overline{CBA} + \overline{DCBA}$$

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

a		BA			
		00	01	11	10
DC	00	1	0	1	1
	01	0	1	1	1
	11	×	×	×	×
	10	1	1	×	×

左图是以a段为例卡诺图化简，也可先求反函数后得到，其它各段用同样方法可得

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

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

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

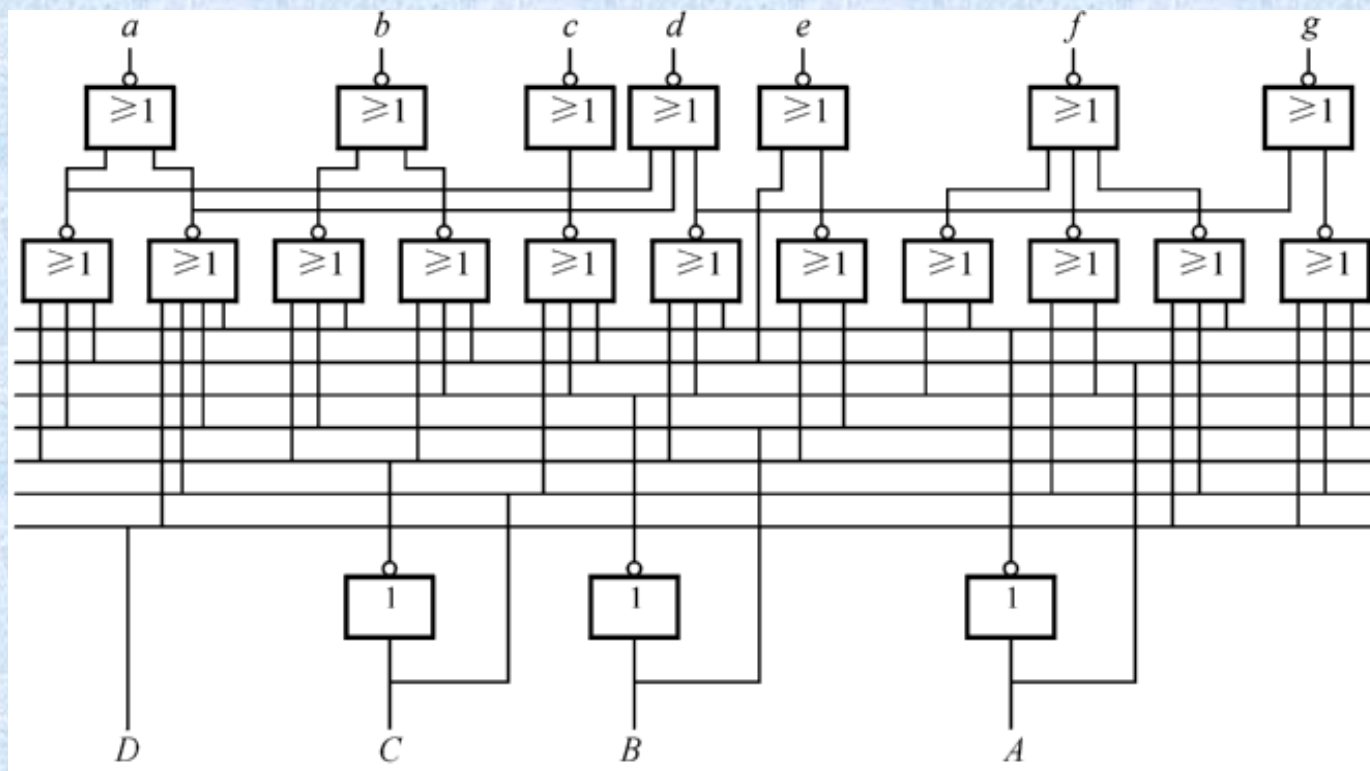
$$= \overline{\overline{\overline{\overline{C} + B + A} + \overline{\overline{\overline{C} + \overline{B} + \overline{A}}}} + \overline{\overline{\overline{D + C + B + \overline{A}}}}}$$

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

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

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

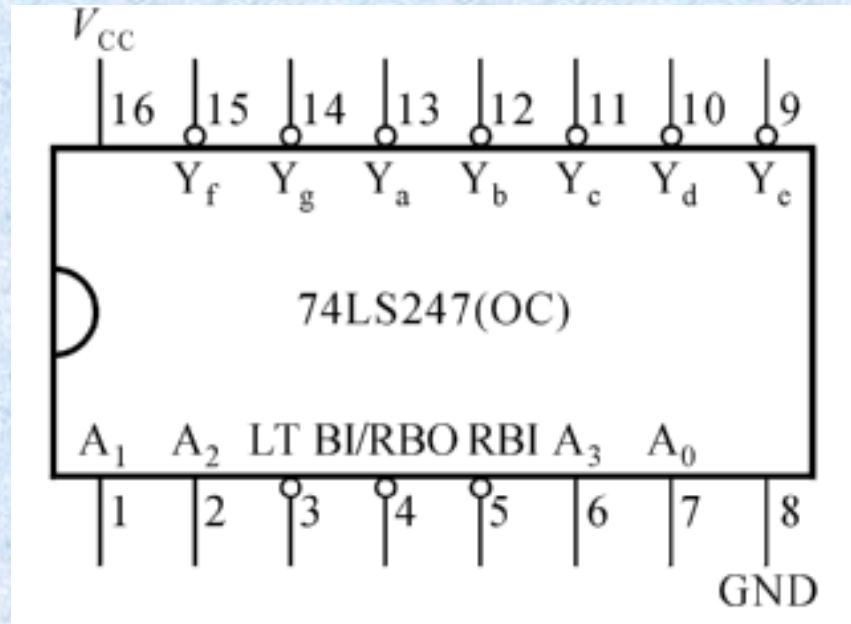
从逻辑式可以计算出需要或非门的个数，即可画出该设计电路。



- ◆ 熟悉了译码器的工作原理和设计方法后，在今后的实际工作中，主要是能看懂电路符号和真值表，以便进行系统电路设计
- ◆ 在后续将详细介绍一些中规模集成芯片的引脚排列和功能。

常用的中规模集成二—十进制译码器芯片

74LS247 (OC)

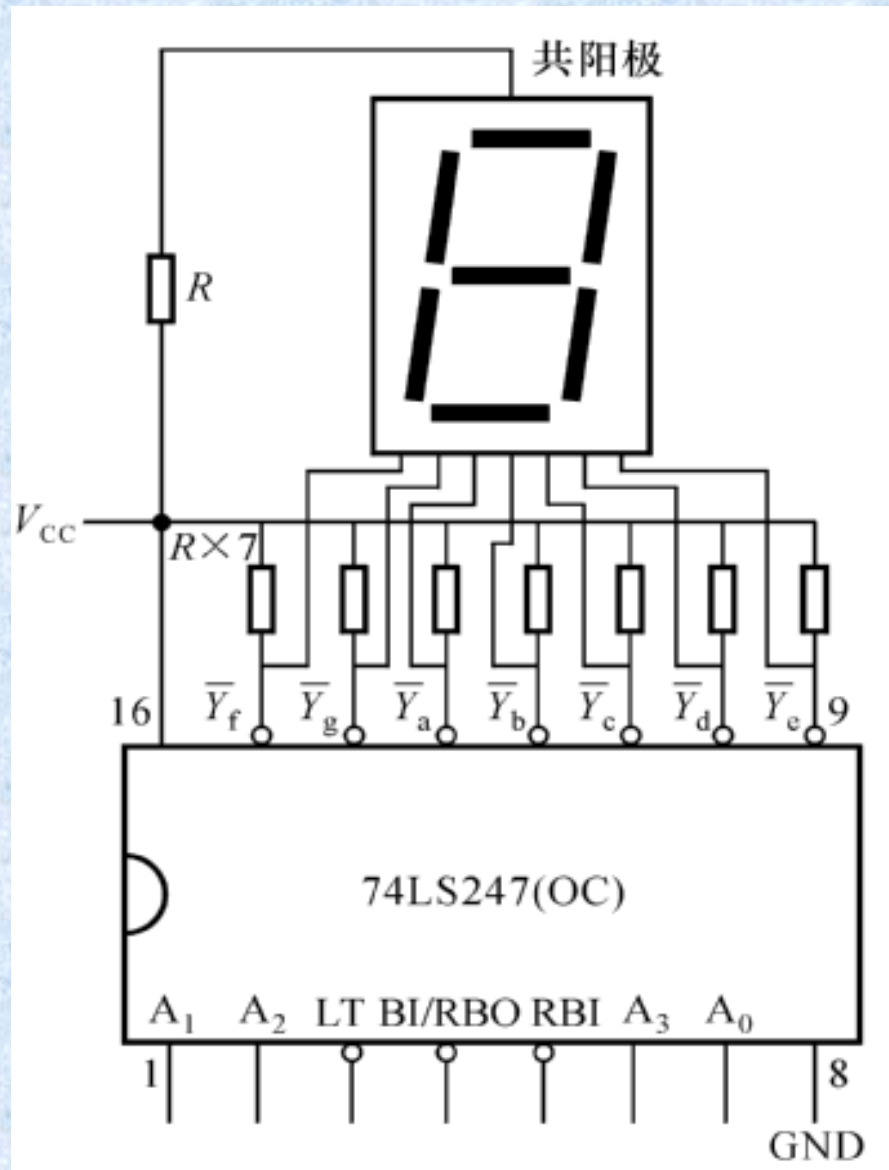


$A_3 \sim A_0$ 是四位8421BCD码输入端， $Y_a \sim Y_g$ 驱动七段数码管的七个输出，为低电平输出，是集电极开路门结构，适用于七段**共阳极**数码管。

另外还有三个测试和控制端,实现有:灯测试、灭灯输入以及灭零输入。



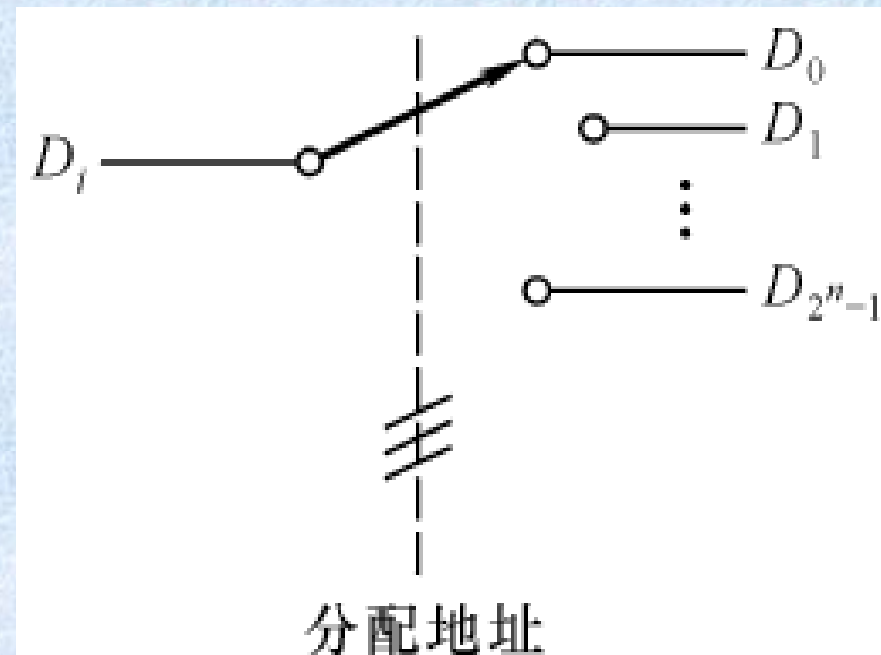
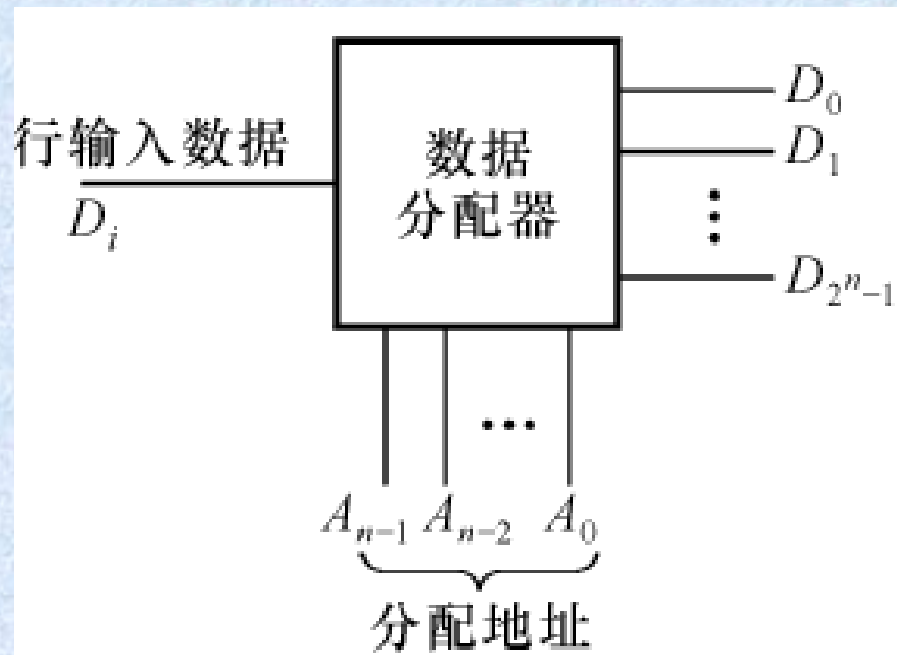
74LS247 (OC门) 驱动七段共阳极半导体数码管电路



二、数据分配器和数据选择器

数据分配器和数据选择器大量应用在数据采集和数字信号处理与通信系统中。

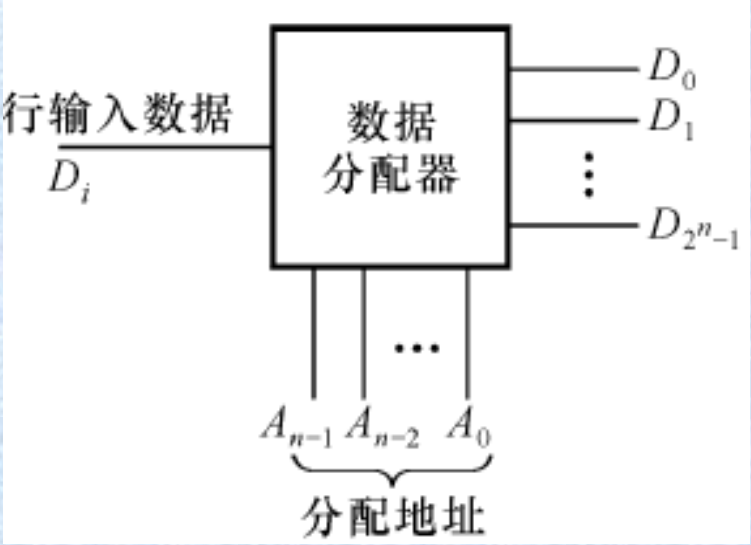
1、数据分配器

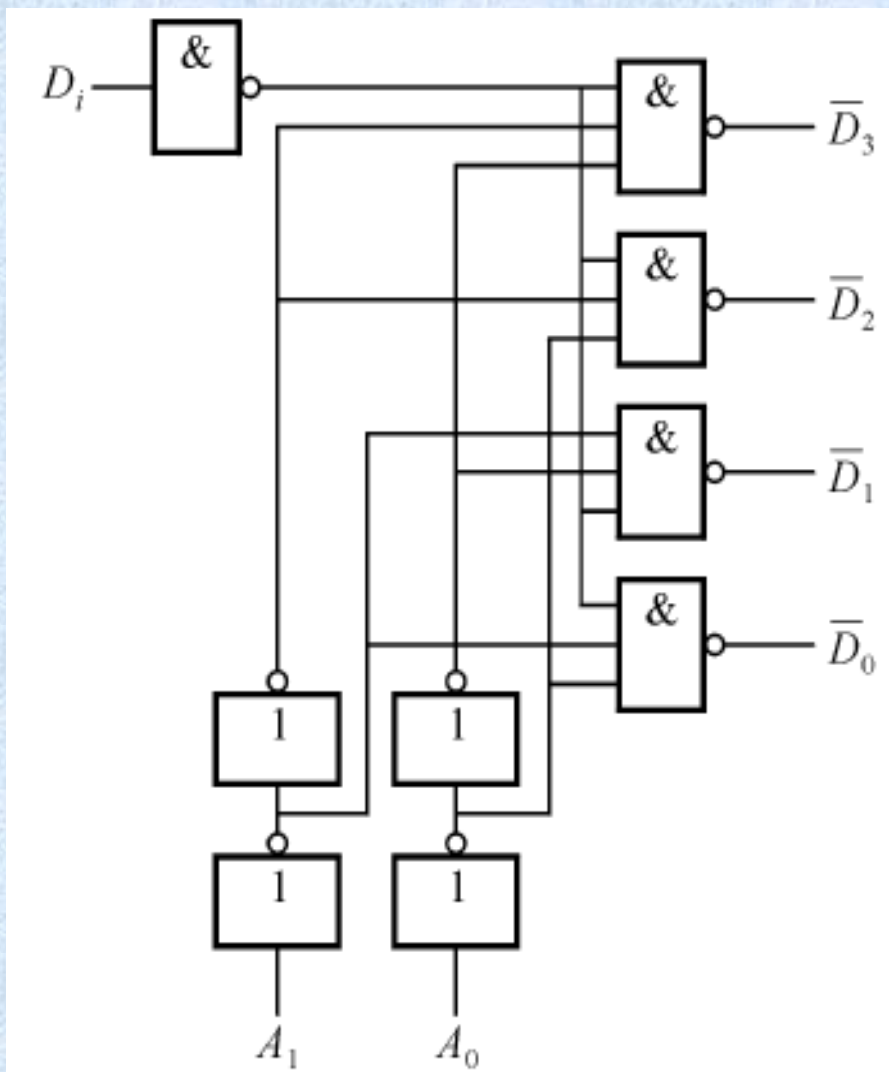


它表示将一路串行输入的数据在选择地址的控制下分配到相应的通道上去的逻辑电路。

在某种意义上，是将串行输入信号转换成并行输出。

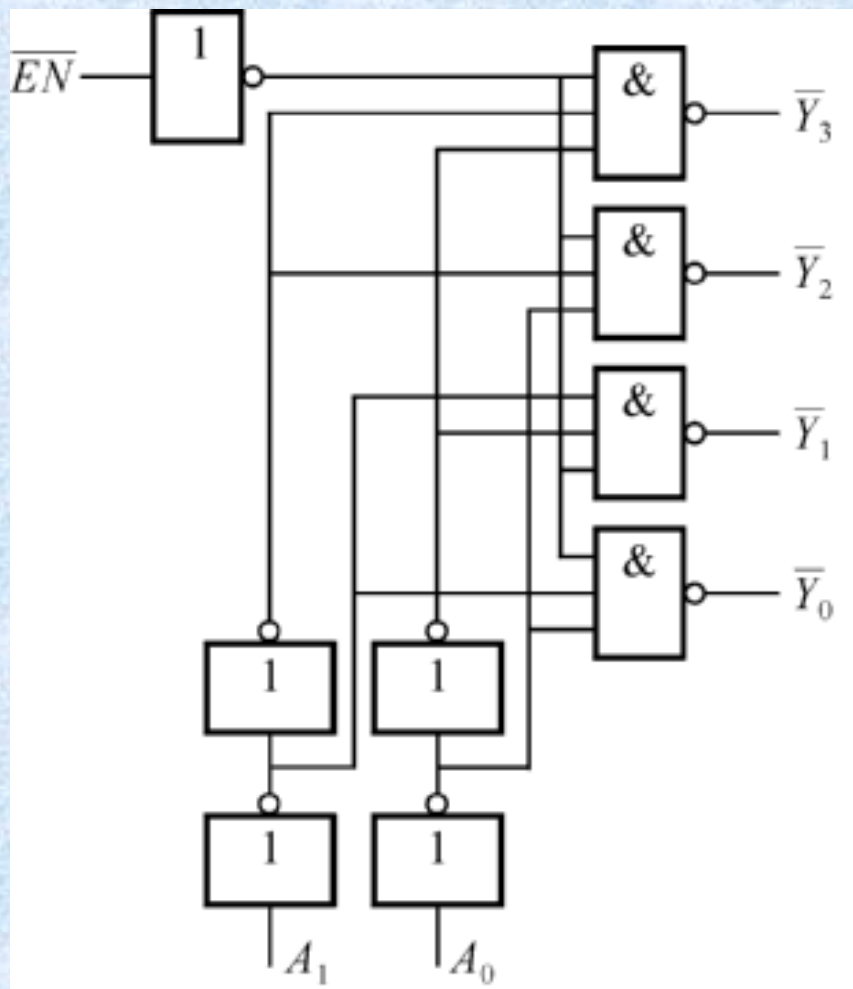
分配地址		数据	输出通道			
A_1	A_0	D_i	\overline{D}_3	\overline{D}_2	\overline{D}_1	\overline{D}_0
0	0	D_i	1	1	1	D_i
0	1	D_i	1	1	D_i	1
1	0	D_i	1	D_i	1	1
1	1	D_i	D_i	1	1	1



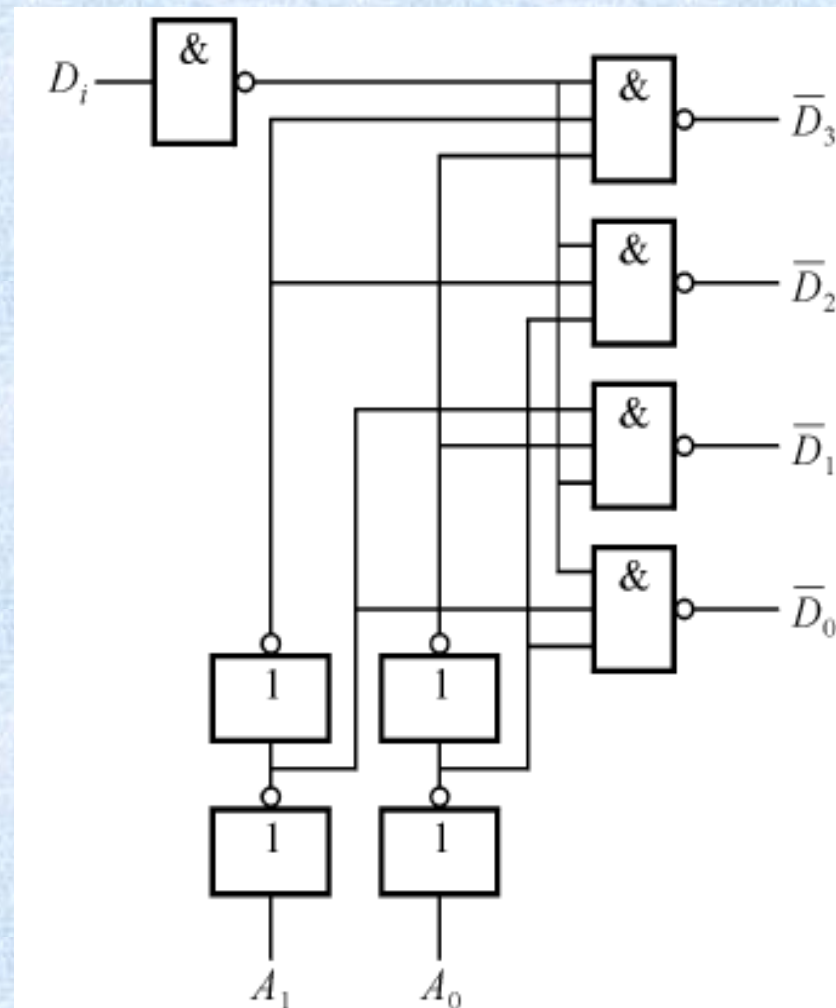


图中 D_i 是串行数据, A_1A_0 分配地址 $\overline{D}_3, \overline{D}_2, \overline{D}_1, \overline{D}_0$ 是四个输出通道。实际上左图是一个具有使能控制端的2/4译码器。

A_1A_0 当作译码器的代码输入, D_i 为使能控制: $D_i = 0$ 时, 译码器使能, $D_i = 1$ 时, 译码器禁止。



2-4译码器

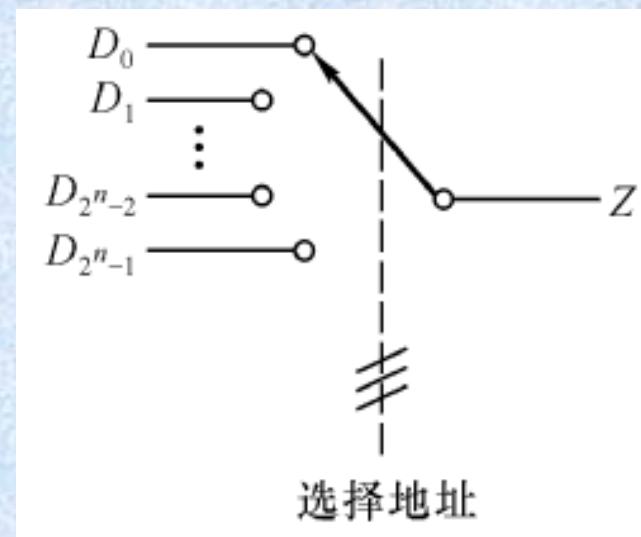
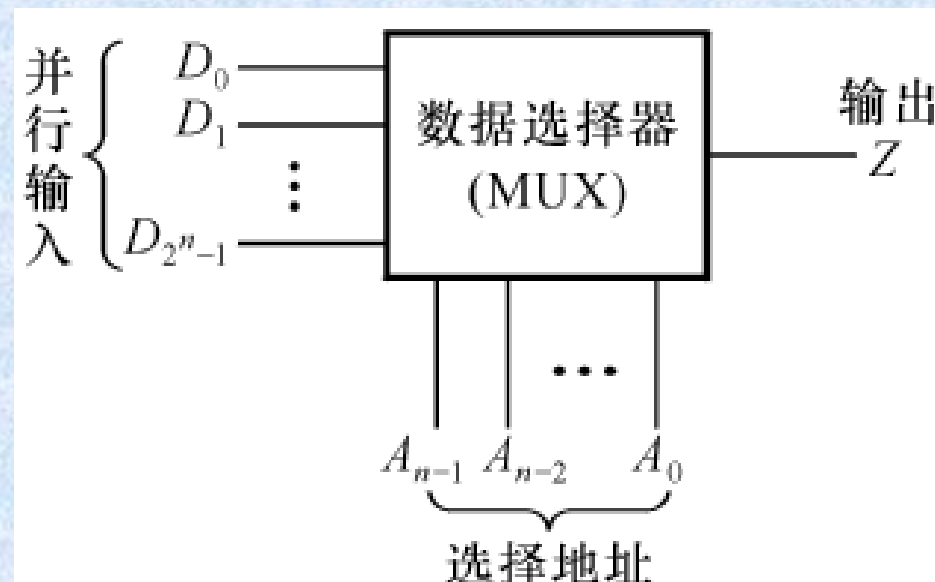


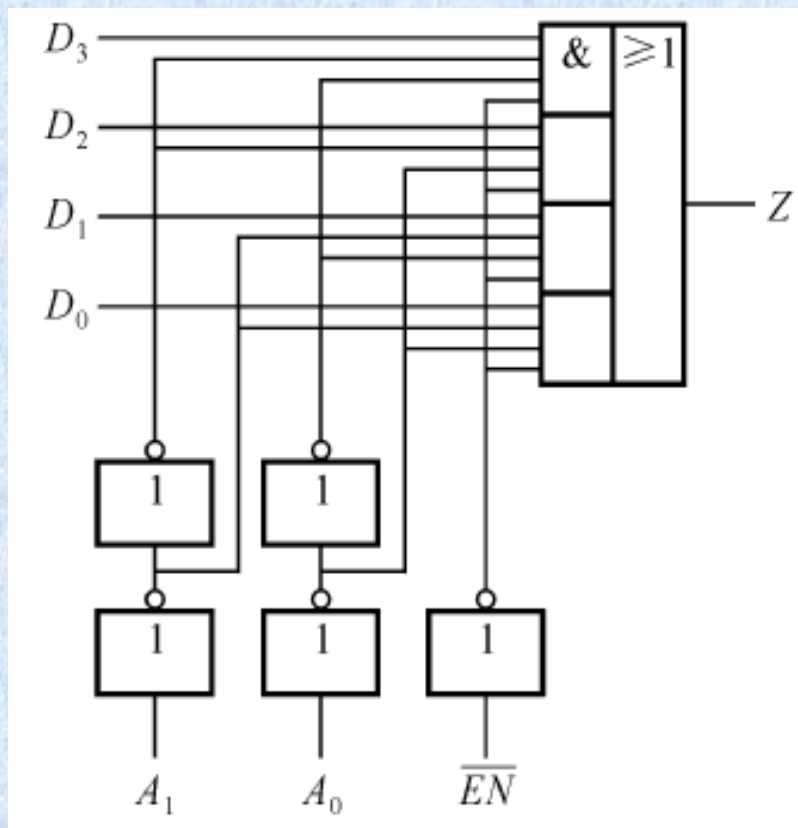
数据分配器

◆一个具有使能端的译码器作数据分配器使用

2、数据选择器

在数字信号的传输过程中，有时需
要从一组输入数据中选出某一个来，
或在多路数据采集系统中，选出某
一路来。能实现这一功能的电路就
是多路数据选译器。





使能控制端 $\overline{EN} = 0$ 时，其输出函数为：

$$Z = D_3 A_1 A_0 + D_2 A_1 \overline{A_0} + D_1 \overline{A_1} A_0 + D_0 \overline{A_1} \overline{A_0}$$

$$= \sum_{i=0}^{2^n-1} D_i m_i$$

式中， m_i 是选择地址的最小组， D_i 和 m_i 就是第*i*位数据和对应的地址最小组。

注意与或门的使用！

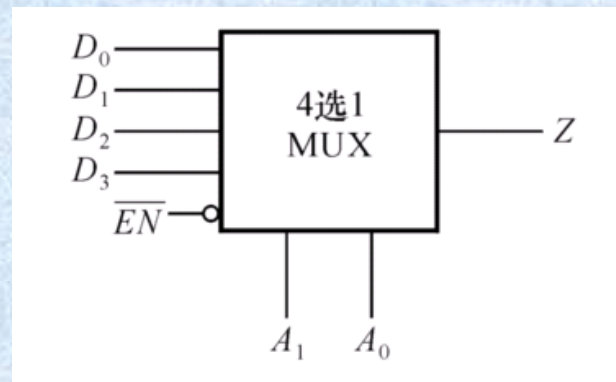
◆ 只要取对应的数据为0或1，就可以实现相应的最小组的逻辑或，实现逻辑函数。

使能

选择地址

输出

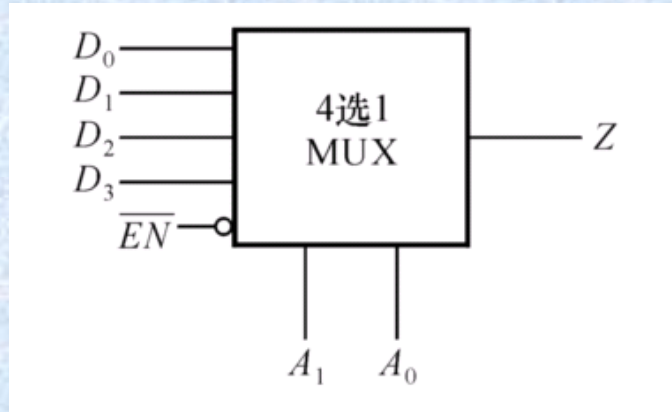
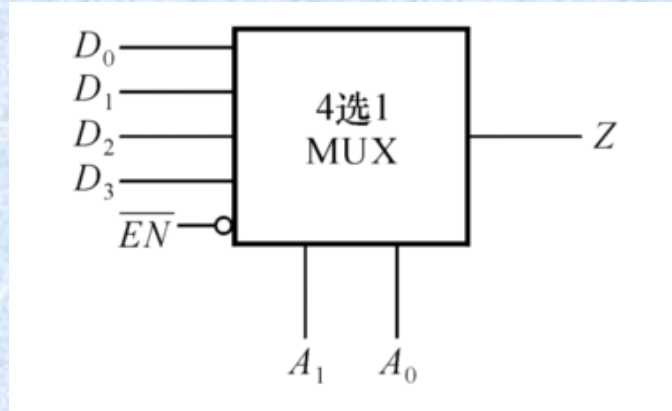
\overline{EN}	A_1	A_0	Z
1	×	×	0
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3



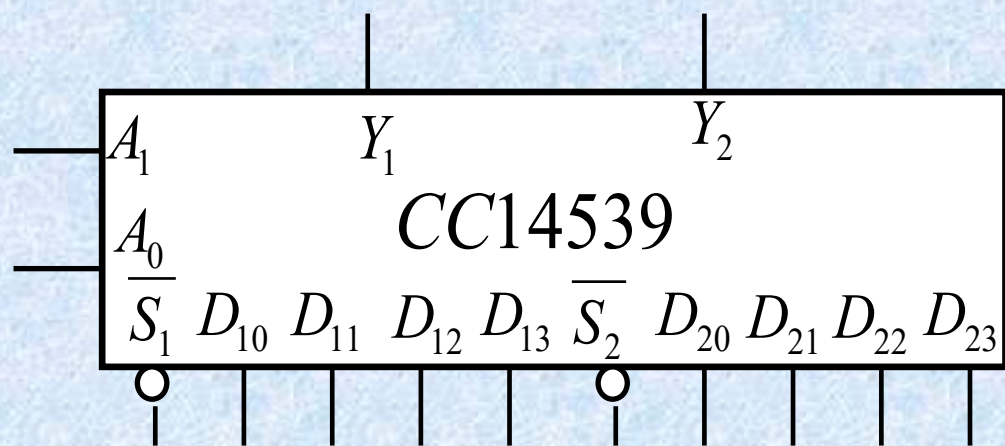
$$\begin{aligned}
 Z &= D_3 A_1 A_0 + D_2 A_1 \overline{A_0} + D_1 \overline{A_1} A_0 + D_0 \overline{A_1} \overline{A_0} \\
 &= \sum_{i=0}^{2^n-1} D_i m_i
 \end{aligned}$$

从数据的传输方式讲，它是一个并行/串行的传输转换电路。
 从电路的输出函数可知，它是一个**与—或表达式**，而电路的结构又是一个与或逻辑结构，因此，用该电路同样**可以用来产生各种各样的组合逻辑电路。**

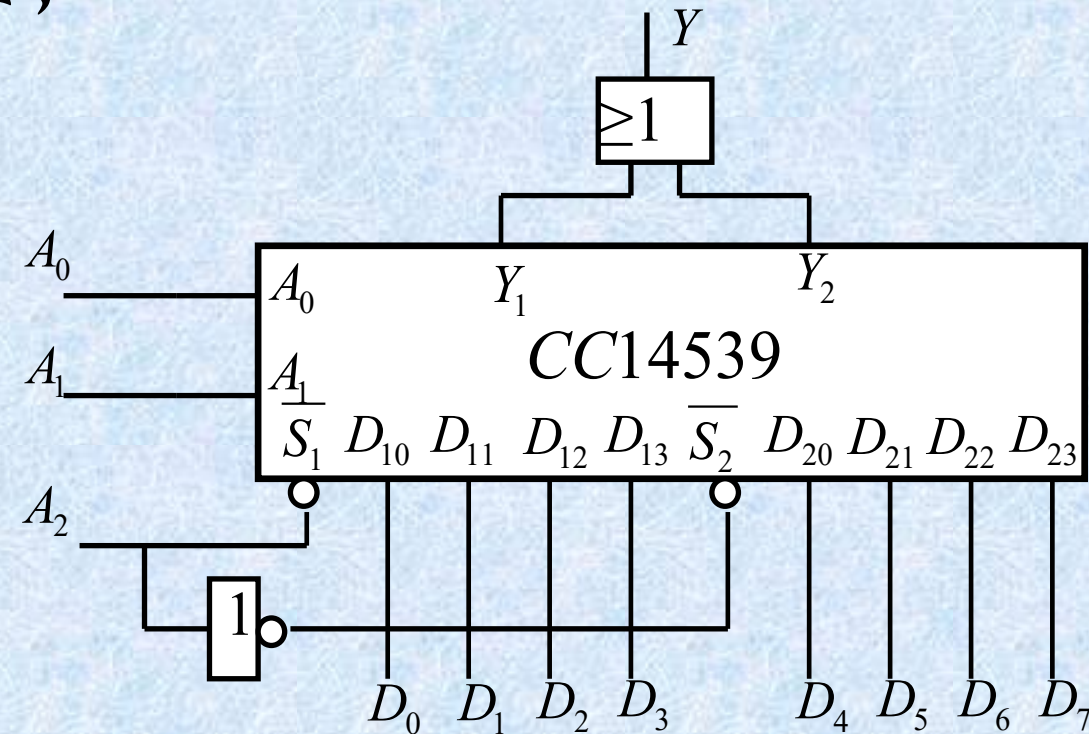
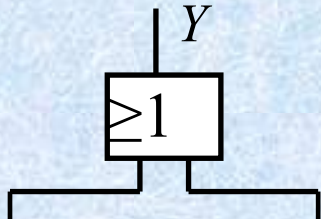
如何用两片1/4和与或门（二选1）实现八选一功能。



例：试用一片CC14539双4选1数据选择器组成一个8选1的数据选择器。可以适当加或门实现。



解：4/1数据选择器只有4个数据输入，2位地址，8/1时应有3位地址，8路数据输入。因此，必须将**使能控制端作为地址扩展端使用**，以实现二片之间的连接。方案采用分时制的工作方式：即高位地址 A_2 为0时，第一片4/1选择器工作，第二片禁止；高位地址 A_2 为1时，第一片禁止，第二片工作。



◆输出用或门的原因

三、二进制加法器

数字系统要完成各种复杂运算和操作，首先必须具备加、减、乘、除四种最基本的算术运算。而在数字电路中，只须具有加法运算和移位操作就能实现乘除法的运算。所以，加法电路是最基本的。在加法电路中半加电路和全加电路又是最低层的。

1、一位加法器

(1) 半加器

仅由两数据相应位相加，不计进位的加法。

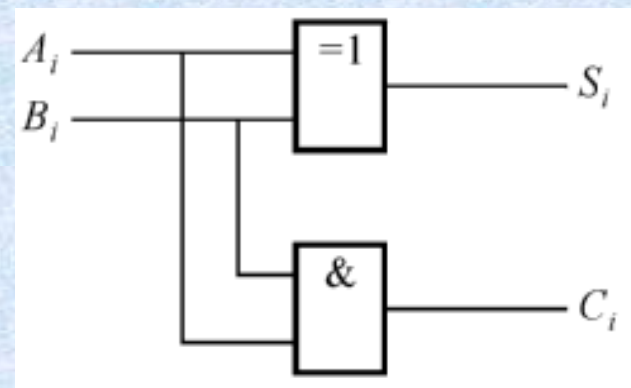
若相应位为 A_i, B_i ，相加后产生半加和一级向高位进位 S_i, C_i

由此得到真值表和两个输出函数式：

输 入		输 出	
被加数 A_i	加数 B_i	半加和 S_i	进位 C_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

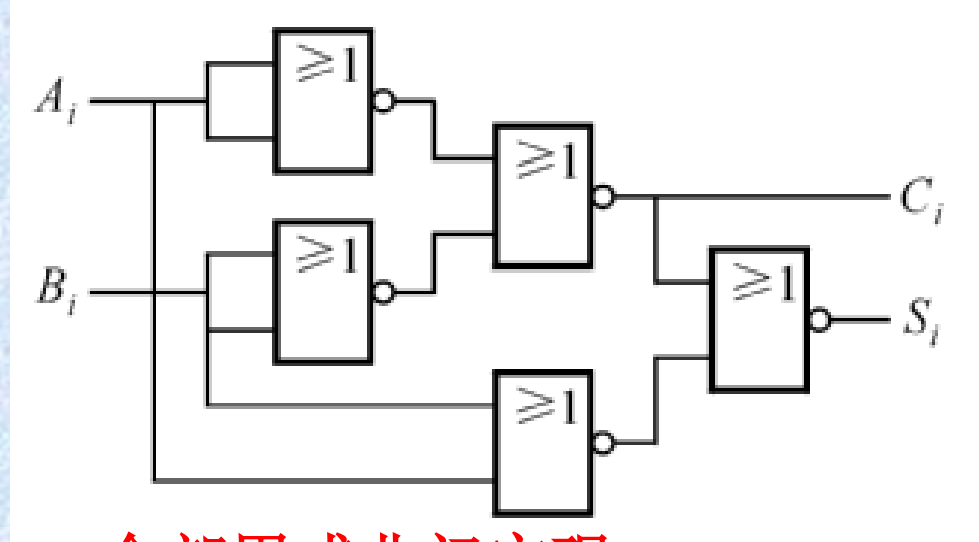
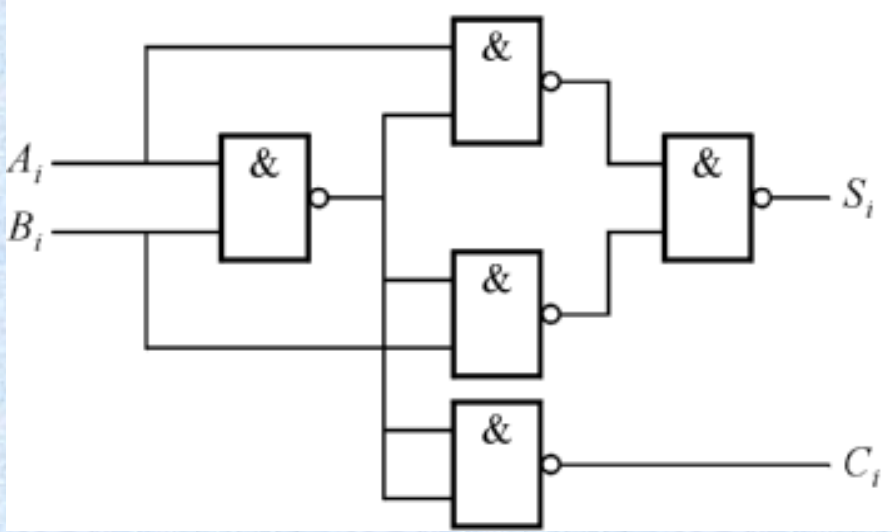
$$S_i = \overline{A_i} B_i + A_i \overline{B_i}$$

$$C_i = A_i B_i$$



① 异或门、与门实现

全部与非门和全部或非门实现



全部用或非门实现

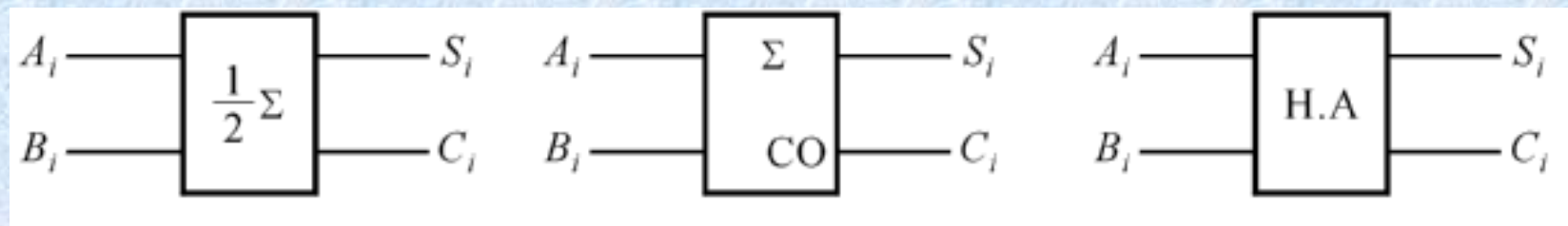
卡诺图中包围“0”格得或与表达式后，由二次求反得到：

$$S_i = (A_i + B_i)(\overline{A_i} + \overline{B_i}) = \overline{\overline{A_i + B_i} \cdot \overline{\overline{A_i} + \overline{B_i}}}$$

$$C_i = \overline{\overline{A_i + B_i}}$$

必须把表达式变换成或非或非表达式

半加器内部的电路不管采用何种逻辑实现，都用逻辑符号表示：



(2) 全加器

能实现二个加数的对应位和相邻低位的进位一起相加的加法电路。 A_i, B_i, C_{i-1} 为二数的相应位和低位进位， S_i, C_i 为全加器的和以及向高位的进位

因此，有真值表如下：

全加器输入			结果输出	
A_i	B_i	C_{i-1}	S_i	C_i
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

从表可得到二个输出函数如下：

$$S_i = \overline{A_i} \overline{B_i} C_{i-1} + \overline{A_i} B_i \overline{C_{i-1}} + A_i \overline{B_i} \overline{C_{i-1}} + A_i B_i C_{i-1} = \Sigma m(1, 2, 4, 7)$$

$$C_i = \overline{A_i} B_i C_{i-1} + A_i \overline{B_i} C_{i-1} + A_i B_i \overline{C_{i-1}} + A_i B_i C_{i-1} = \Sigma m(3, 5, 6, 7)$$

(1) 该式是最基本的**最小项之和**形式用**与非门**实现

特点？

(2) 对上述式子作变换后，也可以用**二个半加器**实现。
(回顾半加器实现的逻辑)

$$S_i = A_i \oplus B_i \oplus C_{i-1} = S_{Hi} \oplus C_{i-1}$$

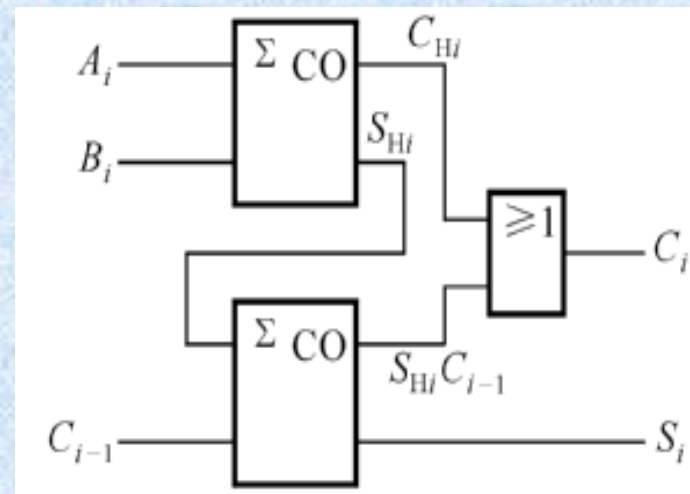
$$C_i = A_i B_i + S_{Hi} C_{i-1} = C_{Hi} + (A_i \oplus B_i) C_{i-1}$$

		$B_i C_{i-1}$			
		00	01	11	10
A_i	0	0	1	0	1
	1	1	0	1	0

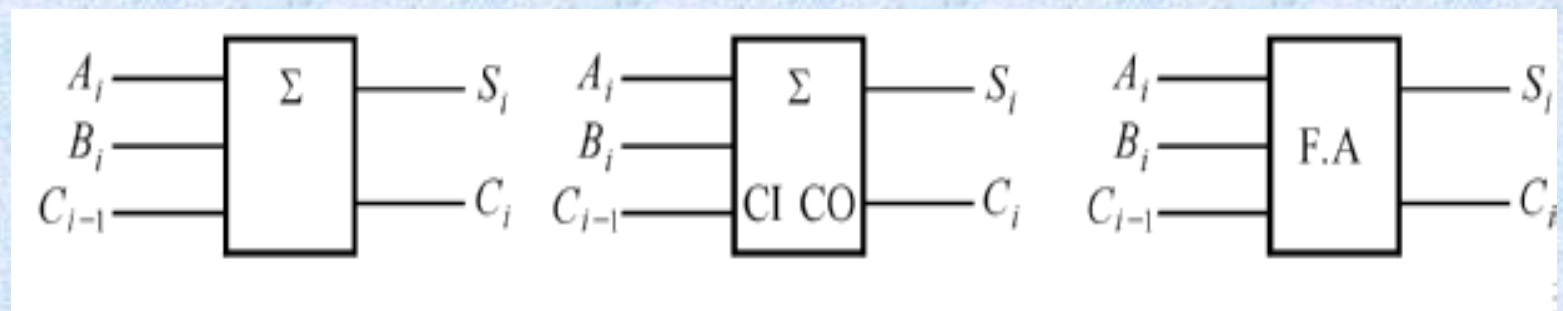
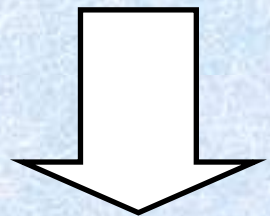
棋盘格→异或
逻辑关系

		$B_i C_{i-1}$			
		00	01	11	10
A_i	0	0	0	1	0
	1	0	1	1	1

$$A_i B_i + A_i \overline{B_i} C_{i-1} + \overline{A_i} B_i C_{i-1}$$



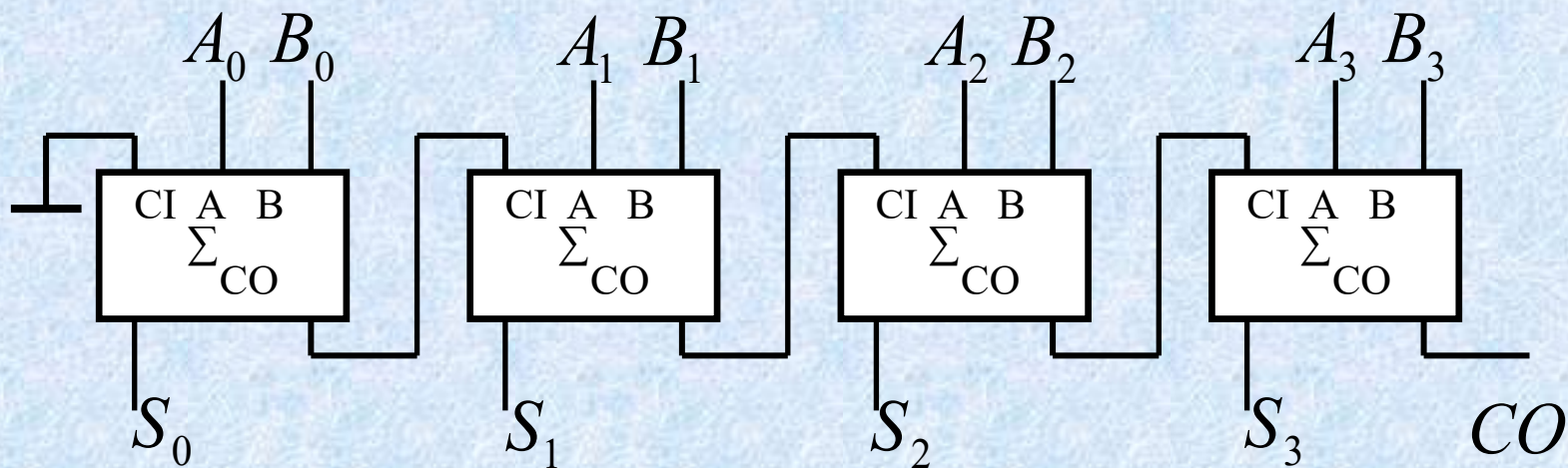
全加器的逻辑符号



2、 多位二进制加法器

多位二进制加法电路种类很多，如四位并行输入**串行进位**加法电路，如图所示：

这种加法运算的速度是比较低的，在最不利的情况下，每做一次加法运算，需要经过4个全加器的传输延迟时间，才能得到稳定可靠的运算结果。

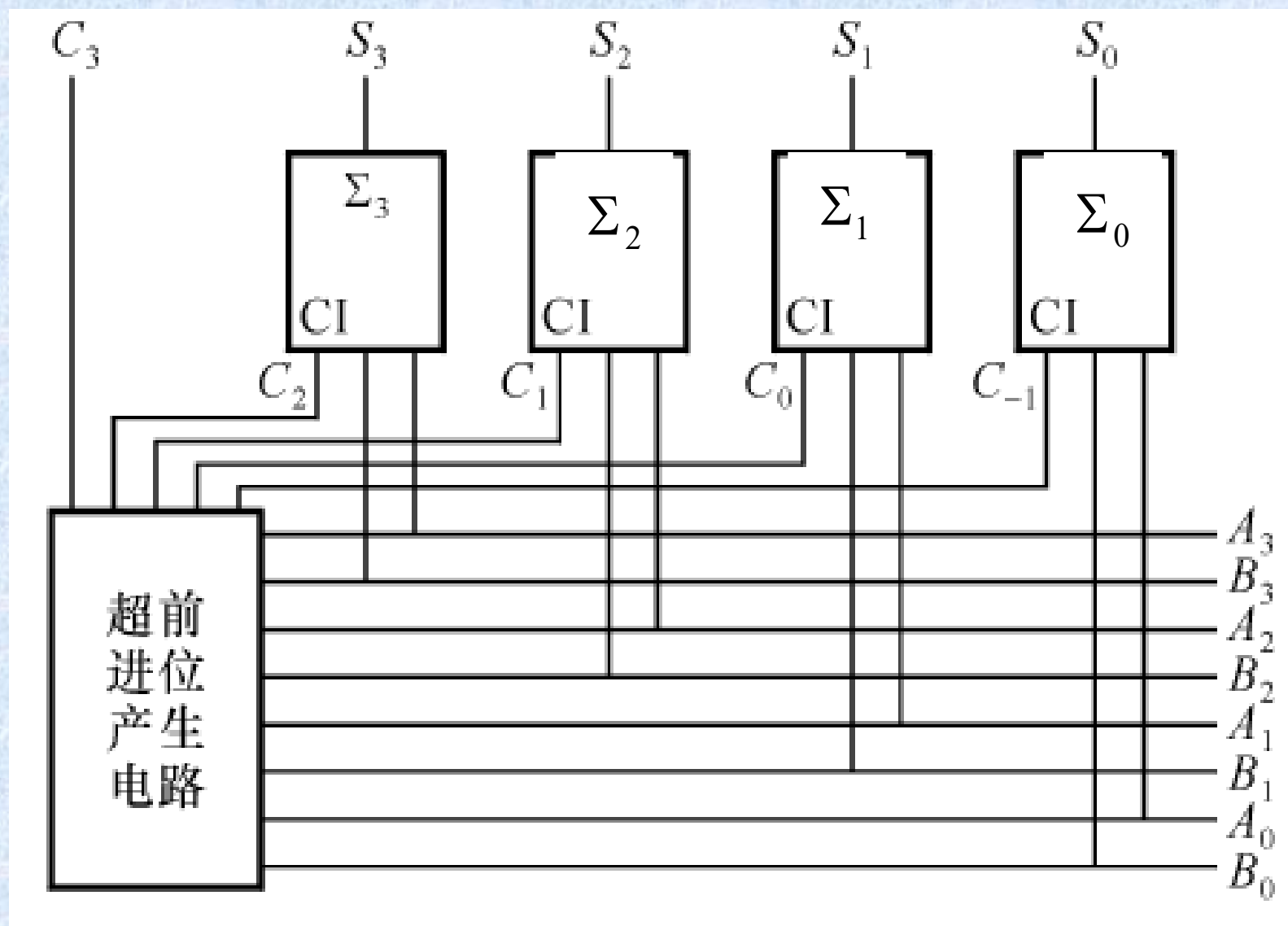


为提高运算速度，必须设法减小或消除由于进位信号逐级传递所消耗的时间。那么高的进位输入信号能否在相加运算开始时就知道呢？

注意到第 i 位的进位输入信号 C_{i-1} 是两个加数中第 $i-1$ 位以下各位数据的函数，即有：

$$C_{i-1} = f(A_{i-1}, A_{i-2}, \dots, A_1, A_0, B_{i-1}, B_{i-2}, \dots, B_1, B_0)$$

因此，各超前进位输入信号的产生，可按图示方案实现：



由一位全加器可知，其进位信号为：

$$C_i = A_i B_i + A_i C_{i-1} + B_i C_{i-1} = A_i B_i + (A_i + B_i) C_{i-1} = G_i + P_i C_{i-1}$$

$G_i = A_i B_i$ 为进位**生成函数**； $P_i = A_i + B_i$ 为进位**传递函数**

由此可得 $C_{-1}, C_0, C_1, C_2, C_3$ 5个进位信号的逻辑函数。

$C_{-1} = 0$ 无更低的进位信号输入

$$C_0 = A_0 B_0 + (A_0 + B_0) C_{-1} = A_0 B_0$$

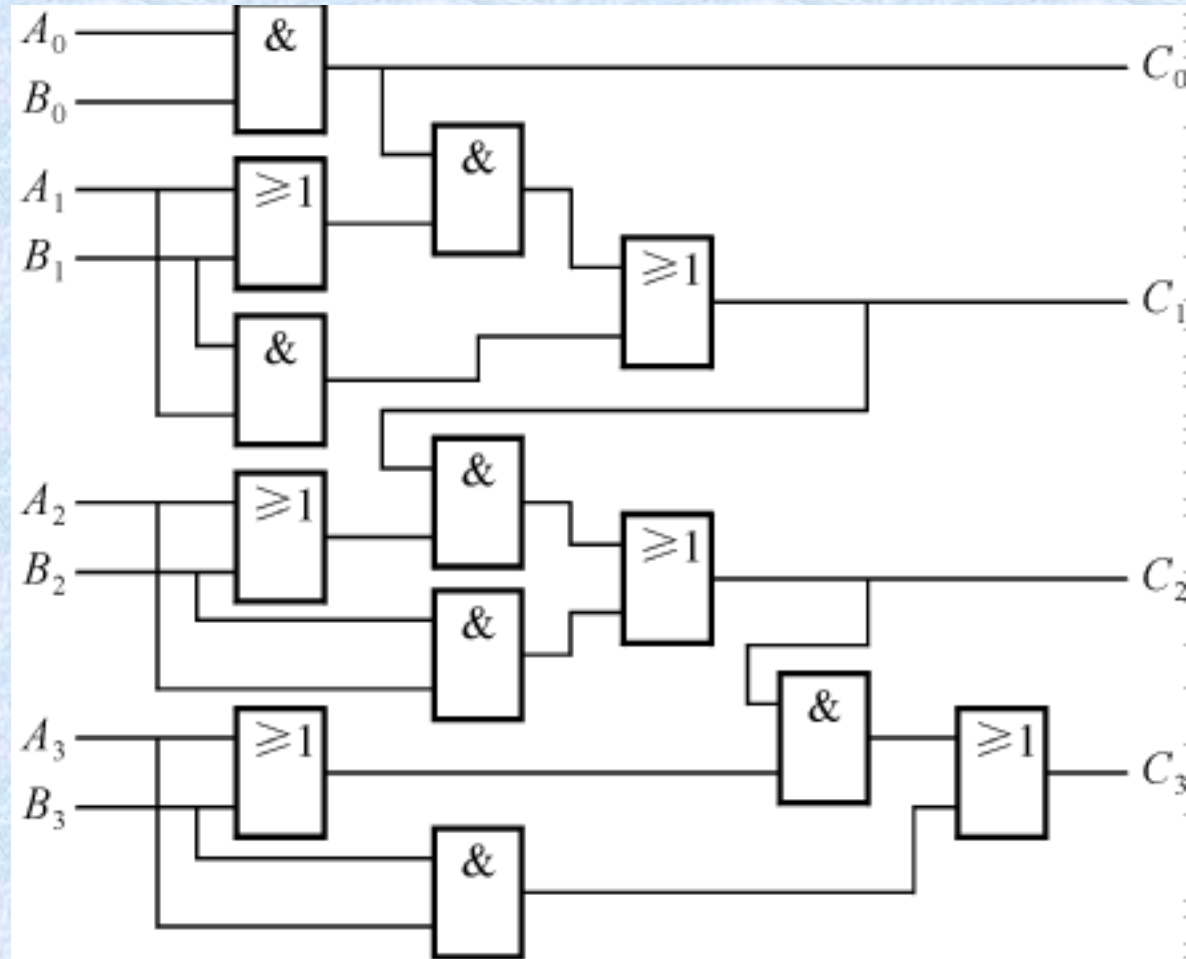
$$\underline{C_1 = A_1 B_1 + (A_1 + B_1) C_0 = A_1 B_1 + (A_1 + B_1) A_0 B_0}$$

$$C_2 = A_2 B_2 + (A_2 + B_2) C_1 = A_2 B_2 + (A_2 + B_2) [A_1 B_1 + (A_1 + B_1) A_0 B_0]$$

$$C_3 = A_3 B_3 + (A_3 + B_3) \{A_2 B_2 + (A_2 + B_2) [A_1 B_1 + (A_1 + B_1) A_0 B_0]\}$$

由此画出的电路如图：

其中TTL中规模集成的74LS283芯片就是按此方法设计的一个四位超前进位加法器电路。



四、数值比较器

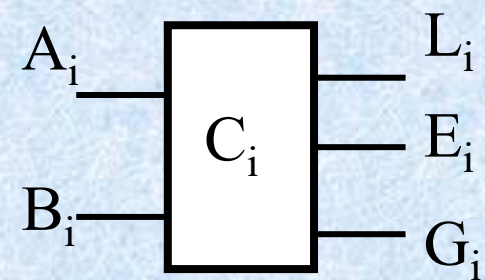
数值比较器用来比较二个数据的大小，是否相等，它经常用在逻辑判断，执行程序的跳转路经或执行何种操作等场合。

1、1位并行数值比较器

令两个一位的二进制数分别为 A_i, B_i

比较结果 $L_{A>B}, L_{A<B}, L_{A=B}$

比较输入		比较结果输出		
A_i	B_i	$L_{A>B}$	$L_{A<B}$	$L_{A=B}$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1



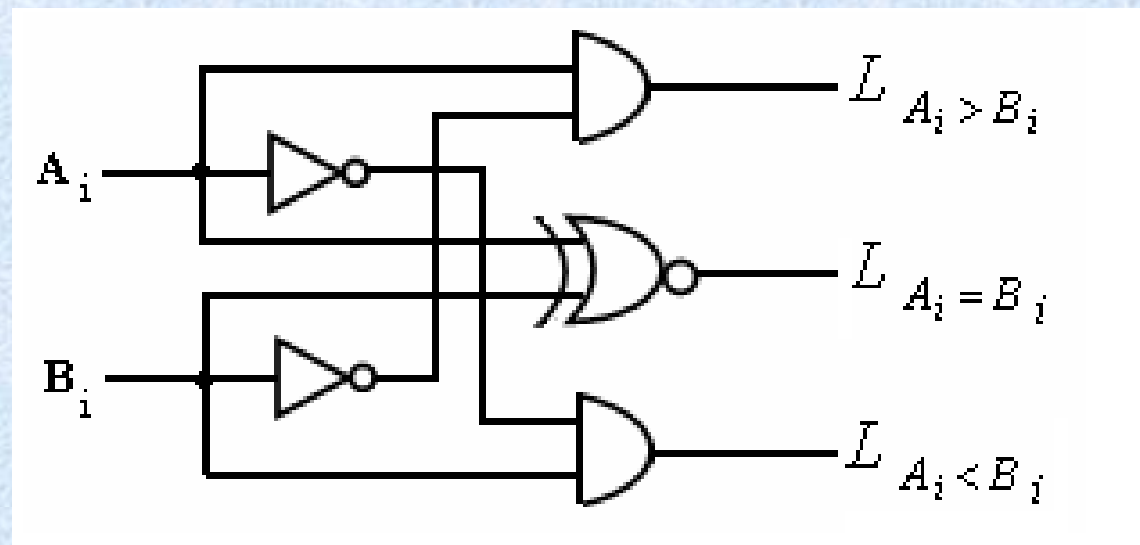
一位比较器

三个输出函数表达式为：

$$L_{A>B} = A_i \overline{B_i}$$

$$L_{A<B} = \overline{A_i} B_i$$

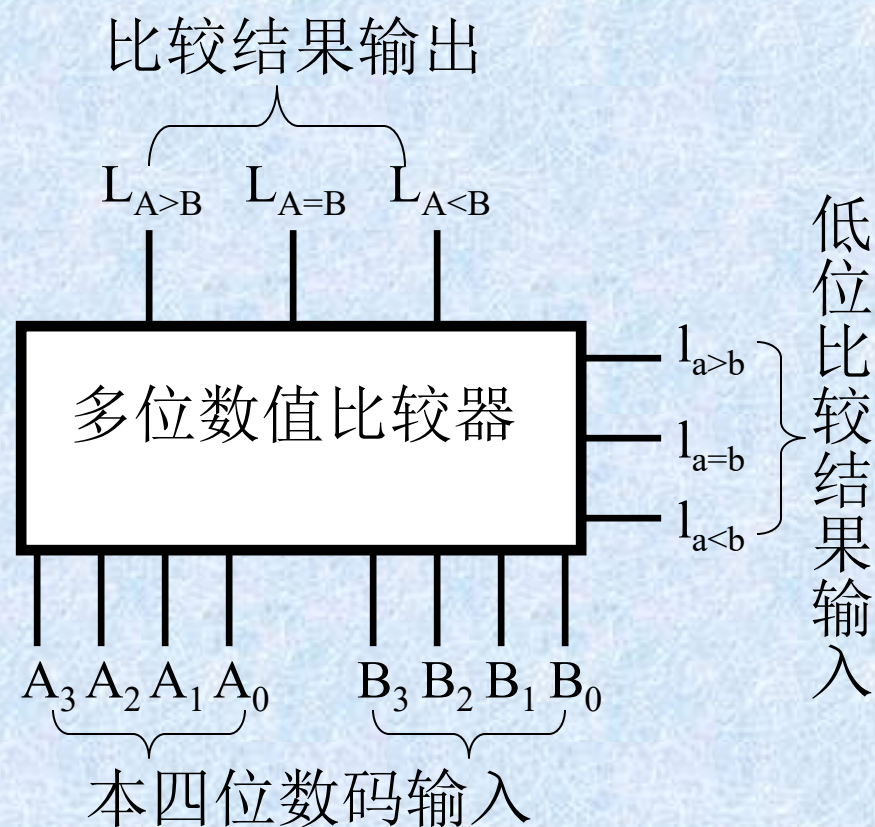
$$L_{A=B} = \overline{A_i} \overline{B_i} + A_i B_i$$



2、多位数值比较器

多位数值比较器通常用“**高位优先**”的比较原则，如二个四位的数值比较器A和B， $A=A_3A_2A_1A_0$ 、 $B=B_3B_2B_1B_0$ ，只要 $A_3 > B_3$ ，则 $A > B$ ； $A_3 < B_3$ ，则 $A < B$ ；在高位相等的时候，按同样的原则比较次高位，如此进行，直到最低位比较完毕。

例：试设计一个四位数码比较器，要求除比较本四位以外，
在本四位相等时，还能比较低位的比较结果，以便能实现更
多位的比较，框图如图所示：



其真值表如下：

本四位输入				低位结果输入			比较结果输出		
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$l_{a>b}$	$l_{a=b}$	$l_{a<b}$	$L_{A>B}$	$L_{A=B}$	$L_{A<B}$
G_3	×	×	×	×	×	×	1	0	0
L_3	×	×	×	×	×	×	0	1	0
E_3	G_2	×	×	×	×	×	1	0	0
E_3	L_2	×	×	×	×	×	0	1	0
E_3	E_2	G_1	×	×	×	×	1	0	0
E_3	E_2	L_1	×	×	×	×	0	1	0
E_3	E_2	E_1	G_0	×	×	×	1	0	0
E_3	E_2	E_1	L_3	×	×	×	0	1	0
E_3	E_2	E_1	E_0	1	0	0	1	0	0
E_3	E_2	E_1	E_0	0	1	0	0	1	0
E_3	E_2	E_1	E_0	0	0	1	0	0	1

$$L_{A>B} = G_3 + E_3G_2 + E_3E_2G_1 + E_3E_2E_1G_0 + E_3E_2E_1E_0l_{a>b}$$

得出三个输出函数式如下：

$$L_{A>B} = G_3 + E_3G_2 + E_3E_2G_1 + E_3E_2E_1G_0 + E_3E_2E_1E_0l_{a>b}$$

$$L_{A<B} = L_3 + E_3L_2 + E_3E_2L_1 + E_3E_2E_1L_0 + E_3E_2E_1E_0l_{a<b}$$

$$L_{A=B} = E_3E_2E_1E_0l_{a=b}$$

由此画出电路图如图所示

(基本框图由一位比较器构成)

