# 期末复习题

## 考点

Boolean Function

* Boolean Algebra的性质化简
* Minterms/Maxterms/ canonical form表示
* Standard Form表示
* 卡诺图化简
* 两级门表示

组合逻辑电路的设计

* Boolean Function
* Truth Table
* Karnaugh map
* Circuit Diagram

组合逻辑电路常见样例

* 半加器
* 加法器
* BCD加法器
* 4bit加法器
* 超前进位加法器

时序逻辑电路分析

* 不同的flipflop（SR、D、JK、T）
* 推断整个流程
* Boolean Function
* Characteristic Table
* Sequential Circuit Diagram
* State Table
* State Diagram

时序电路常见样例

* ripple
* counter(要记住电路图)

输入、输出→真值表→卡诺图SOP化简→化简公式→各种表的填写→画电路图

## CH2-3 布尔代数与最简化

形式转化

**canonical form**

特指写成【Σ】【Π】的两种形式

有minterm和maxterm都可以以这种形式表达

sum‐of‐minterms和product‐of‐maxterms form

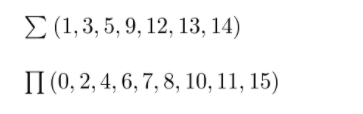
**standard form**

不用【Σ】【Π】，专门用其它

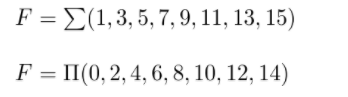
sum of products(SOP) 和 products of sums(POS)

这个时候尽可能要求函数用的变量最小，尽可能简单一些！

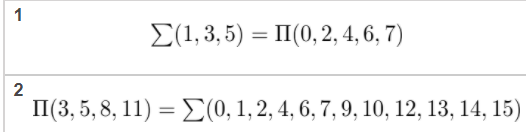
Obtain the truth table of the following functions, and express each function in sum‐of‐minterms and product‐of‐maxterms form



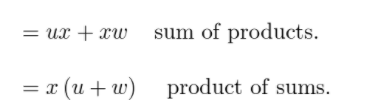
Express the following function as a sum of minterms and as a product of maxterms:



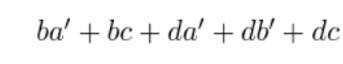
Convert each of the following to the other canonical form:



Convert each of the following expressions into sum of products and product of sums:



Write the following Boolean expressions in sum of products form:

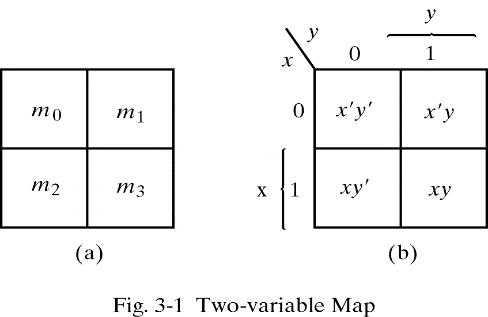


卡诺图化简SOP和POS

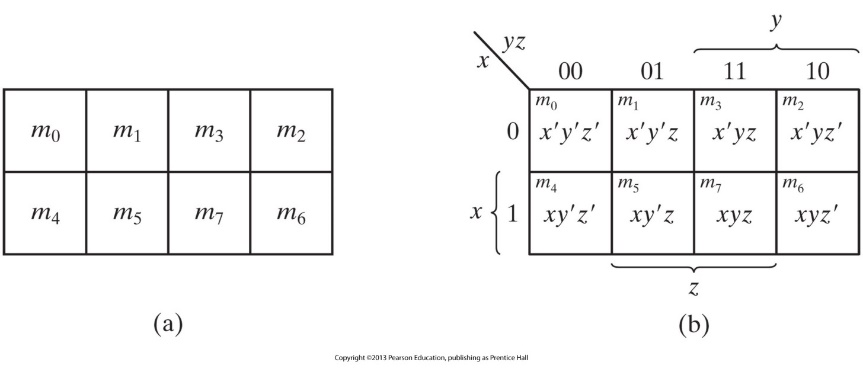
**用canonical from表示的SOP和POS还可以进一步化简，此时用到卡诺图化简**

**化简SOP**

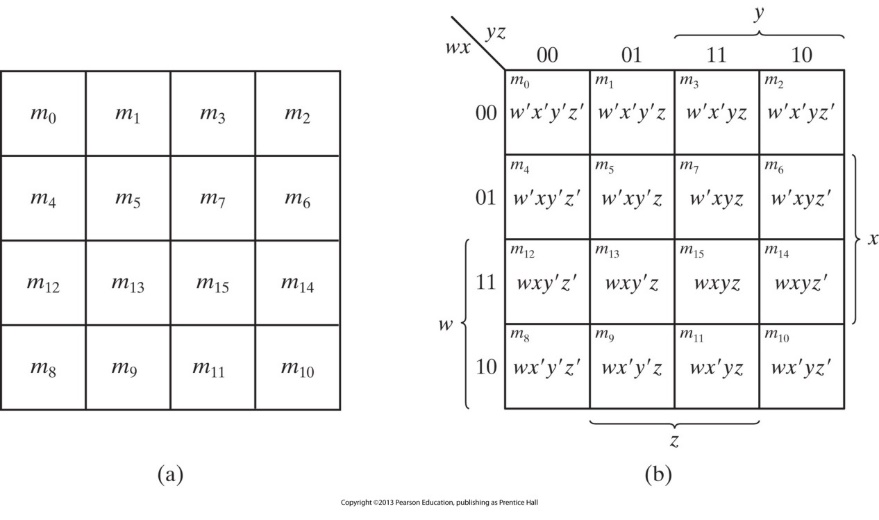
**2变量卡诺图**



**3变量卡诺图**



**4变量卡诺图**



**化简POS**

**取卡诺图为0的部分组成F’，然后用德摩根即可**

两级门电路

**任何布尔函数可以用下面8种两级门电路表示出来**

SOP的最小项和

**AND-OR**

* **NAND-NAND**
* **OR-NAND**
* **NOR-OR**

POS的最大项积

**OR-AND**

* **NOR-NOR**
* **AND-NOR**
* **NAND-AND**

CH4 组合逻辑电路

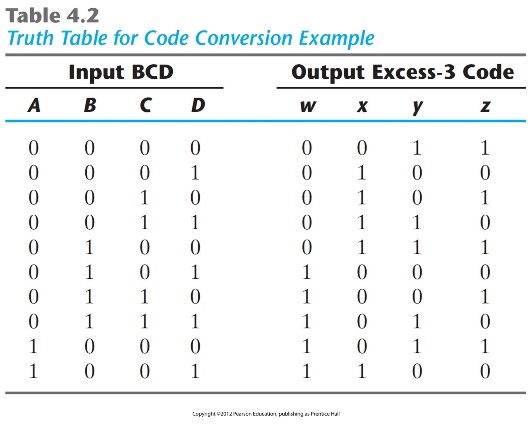
组合逻辑电路的设计

**设计电路：BCD码转换成余3码**

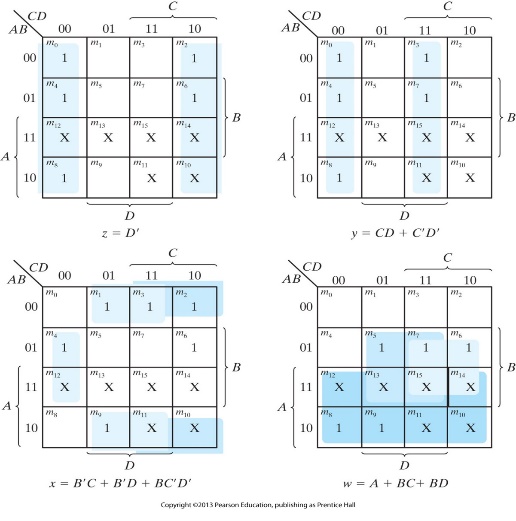
**第一步：根据电路的规格，确定所需的输入(input)和输出(output)数量，并为每一个输入和输出分配一个符号**

输入4变量，输出4变量

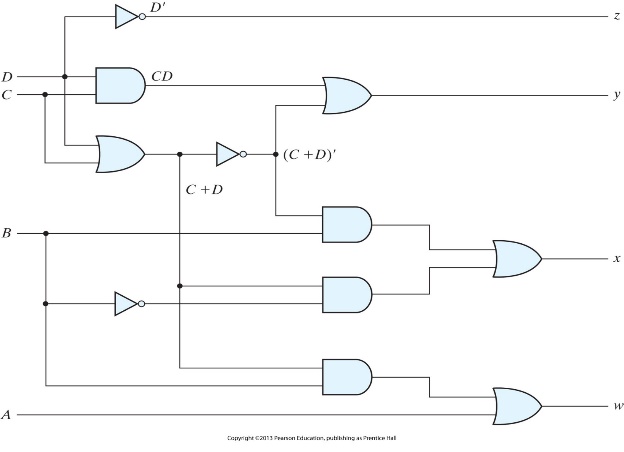
**第二步：列出输入和输出之间的真值表(truth table)**



**第三步：用卡诺图(Karnaugh map)求出输出函数的最简表达式(Boolean Function)**



**第四步：画出逻辑电路图(logic diagram)**



## CH5 时序电路

常见的触发器与锁存器

**SR锁存器 Q(t+1) = S + R’Q(t)**

增加一个额外的控制输入端来决定锁存器的改变

En：使能信号(高电平有效)

S：置位端(set)

R：复位端(reset)

En=0，S=x，R=x维持上一次的状态（置位/复位状态）

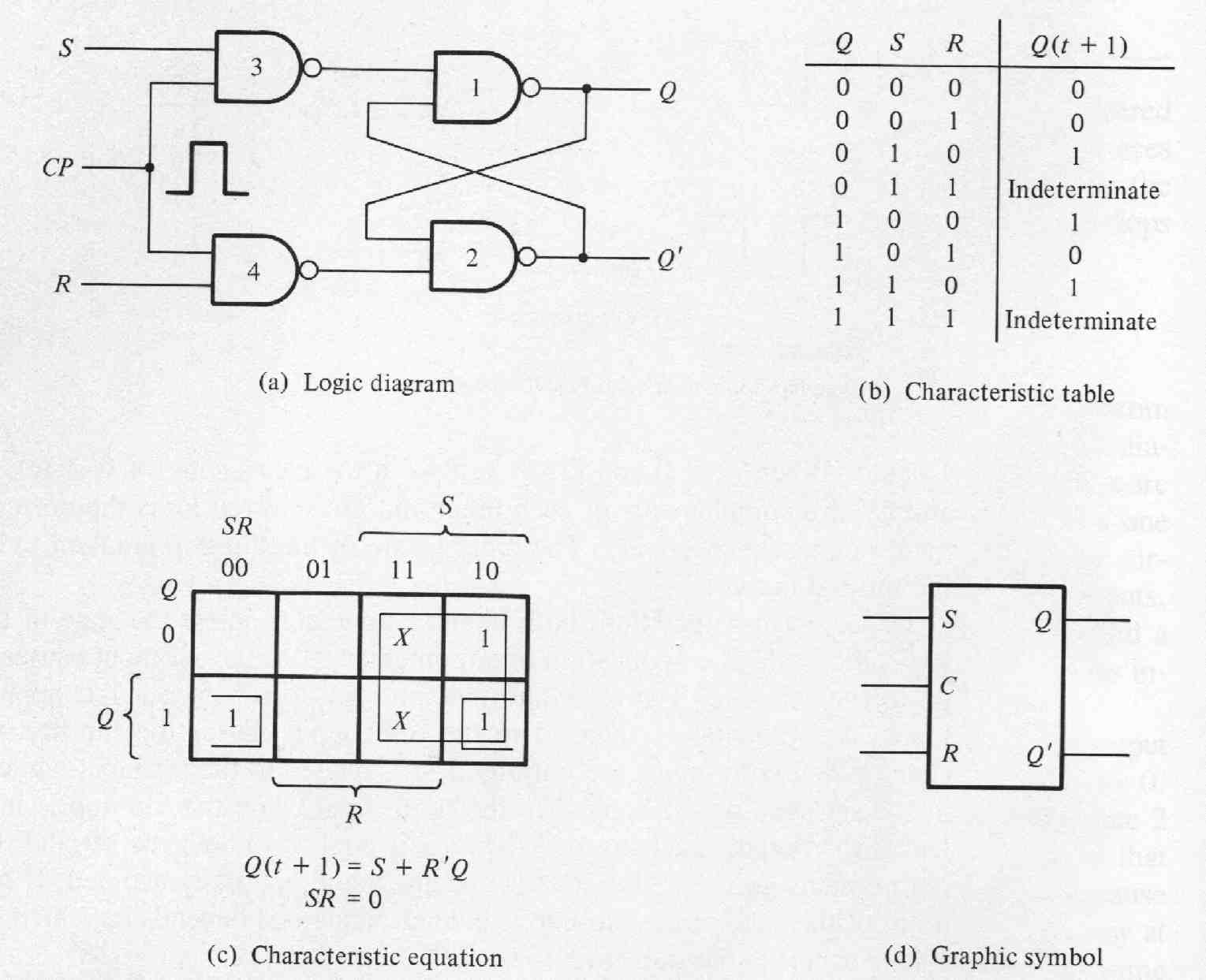
En=1，S=0，R=0维持上一次的状态（置位/复位状态）

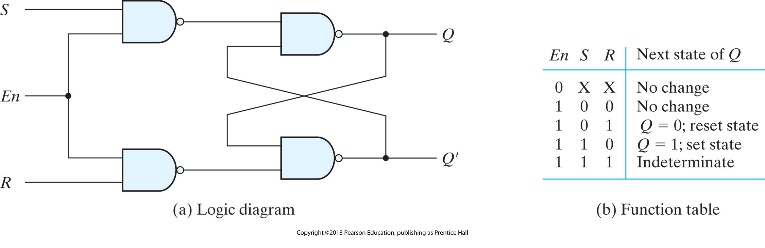
En=1，S=1，R=0置位状态 → Q=1，Q’=0

En=1，S=0，R=1复位状态 → Q=0，Q’=1

En=1，S=1，R=1不稳定，锁存器不允许这种状态出现

为了避免不稳定状态，S端和R端必须都先回到0才能进入置位/复位状态





**D触发器 Q(t+1) = D**

消除SR锁存器中不确定状态的一种方法是确保输入S和R不同时等于1

En：使能信号(高电平有效)

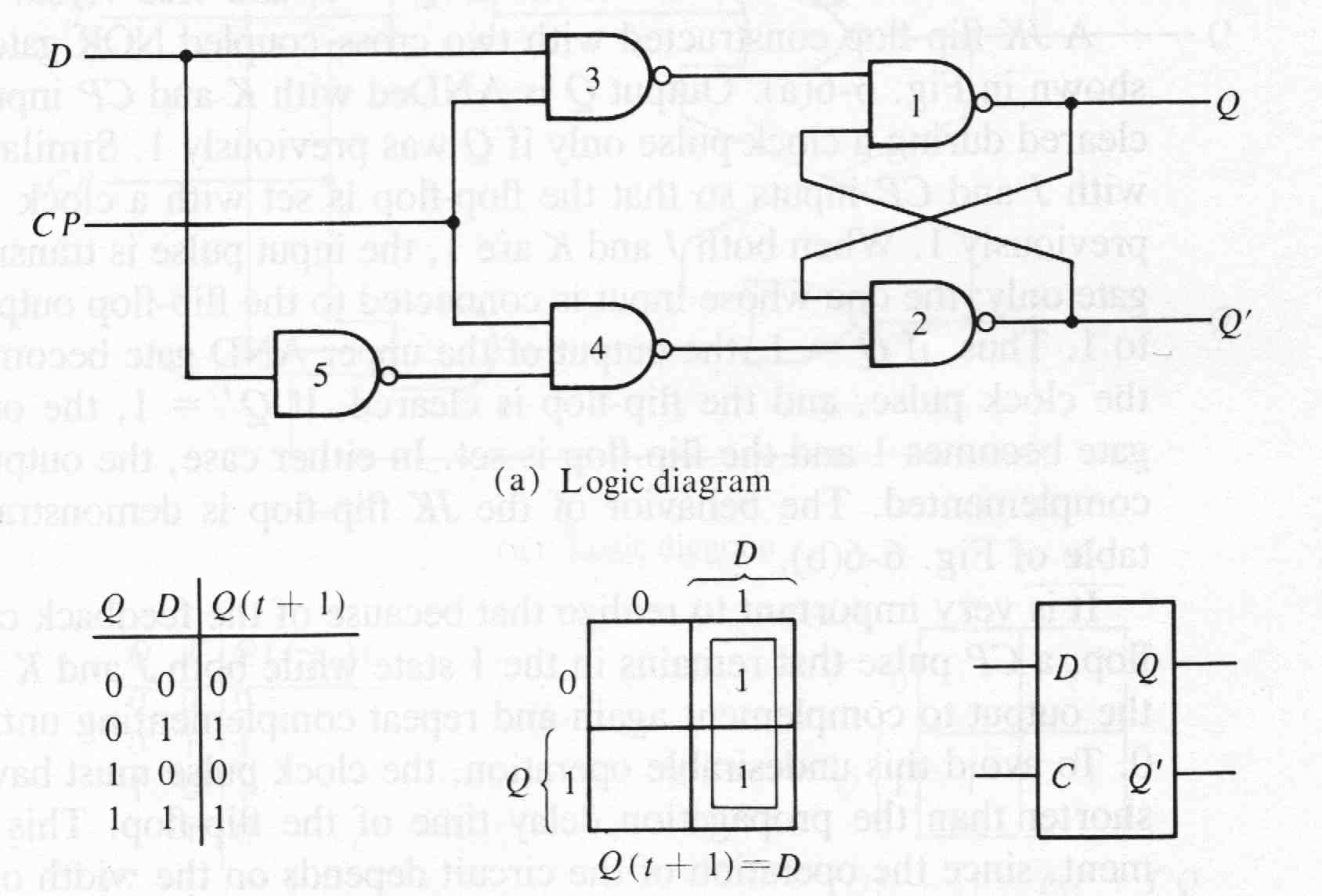
D：采样信号

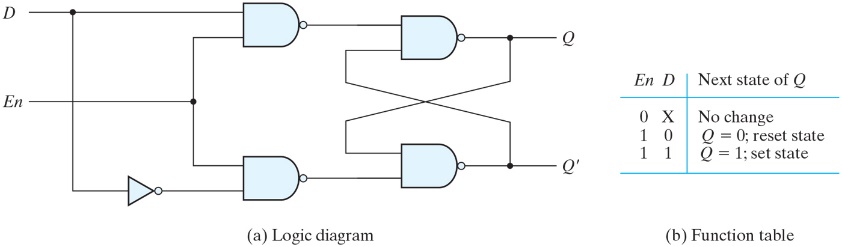
En=0，D=x维持上一次的状态（置位/复位状态）

En=1，D=0复位状态 → Q=0，Q’=1

En=1，D=1置位状态 → Q=1，Q’=0

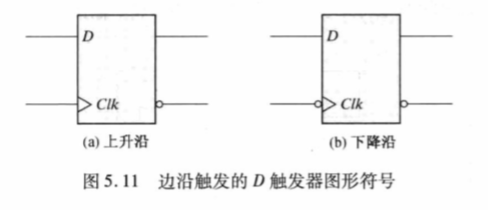
当使能信号无效时，出现在数据输入端的信息将会一直保持在输出端Q，直到使能信号有效位置





边沿D触发器的图形符号如图，除了Clk旁边有一个箭头符号外，其它与D锁存器类似，箭头表示动态输入，表明触发器只对时钟的边沿信号响应

左边的是上升沿触发，右边的是下降沿触发



**JK触发器 Q(t+1)=JQ’(t)+K’Q(t)**

J：置位信号

K：复位信号

J=0，K=0维持上一次的状态（置位/复位状态）

J=1，K=0置位状态→ Q=1，Q’=0

J=0，K=1复位状态 → Q=0，Q’=1

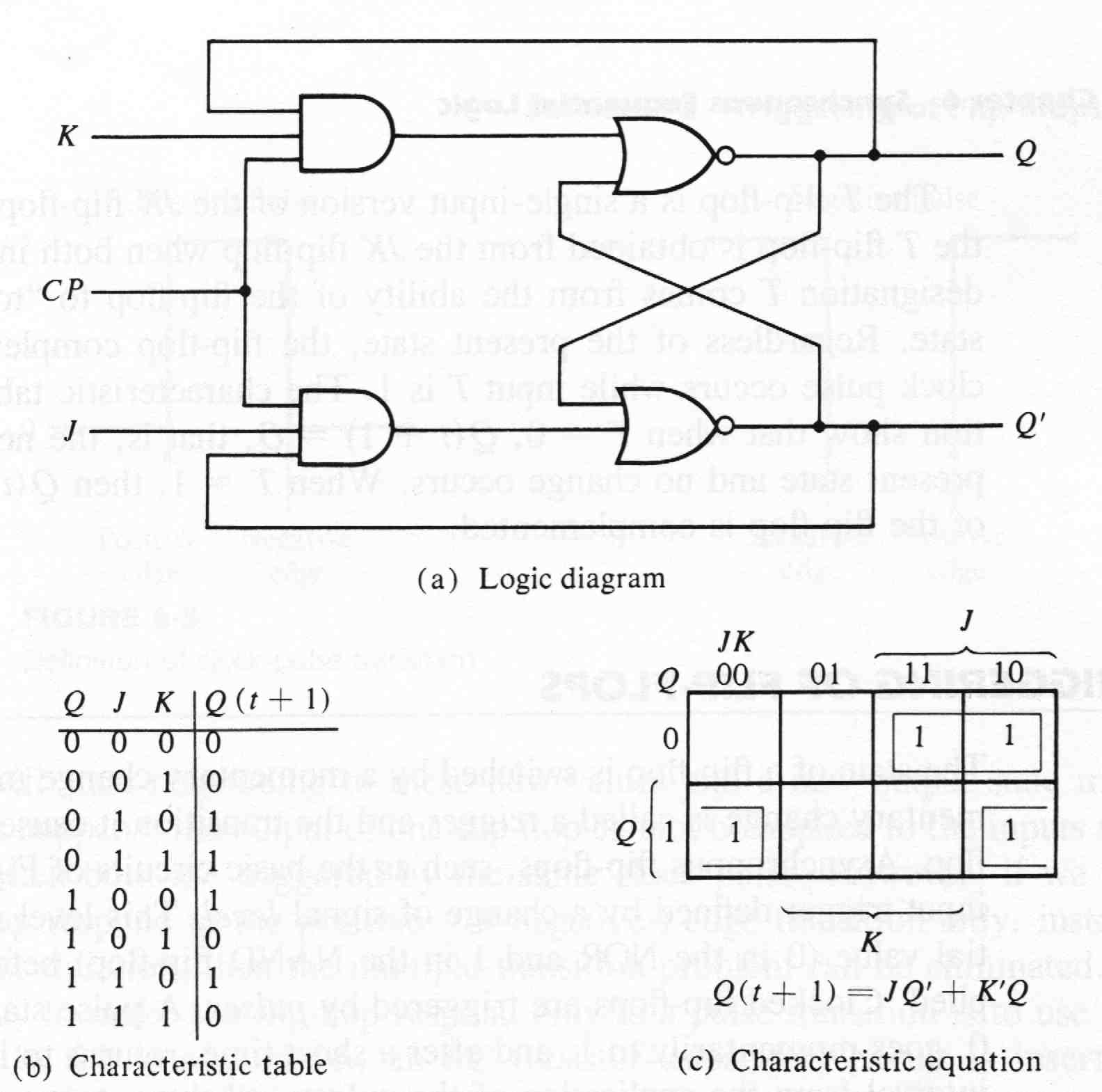
J=1，K=1翻转状态→Q(t+1)=Q’(t)

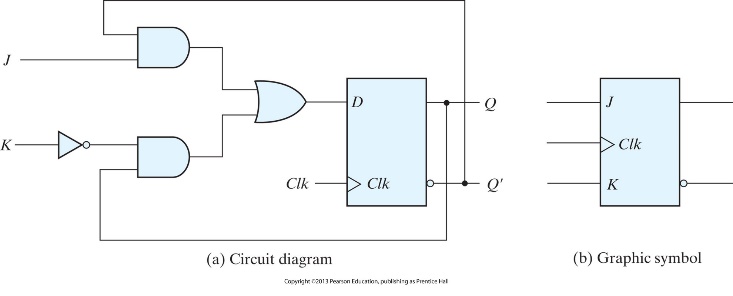
JK触发器的三种功能

置位：输出为1

复位：输出为0

输出翻转





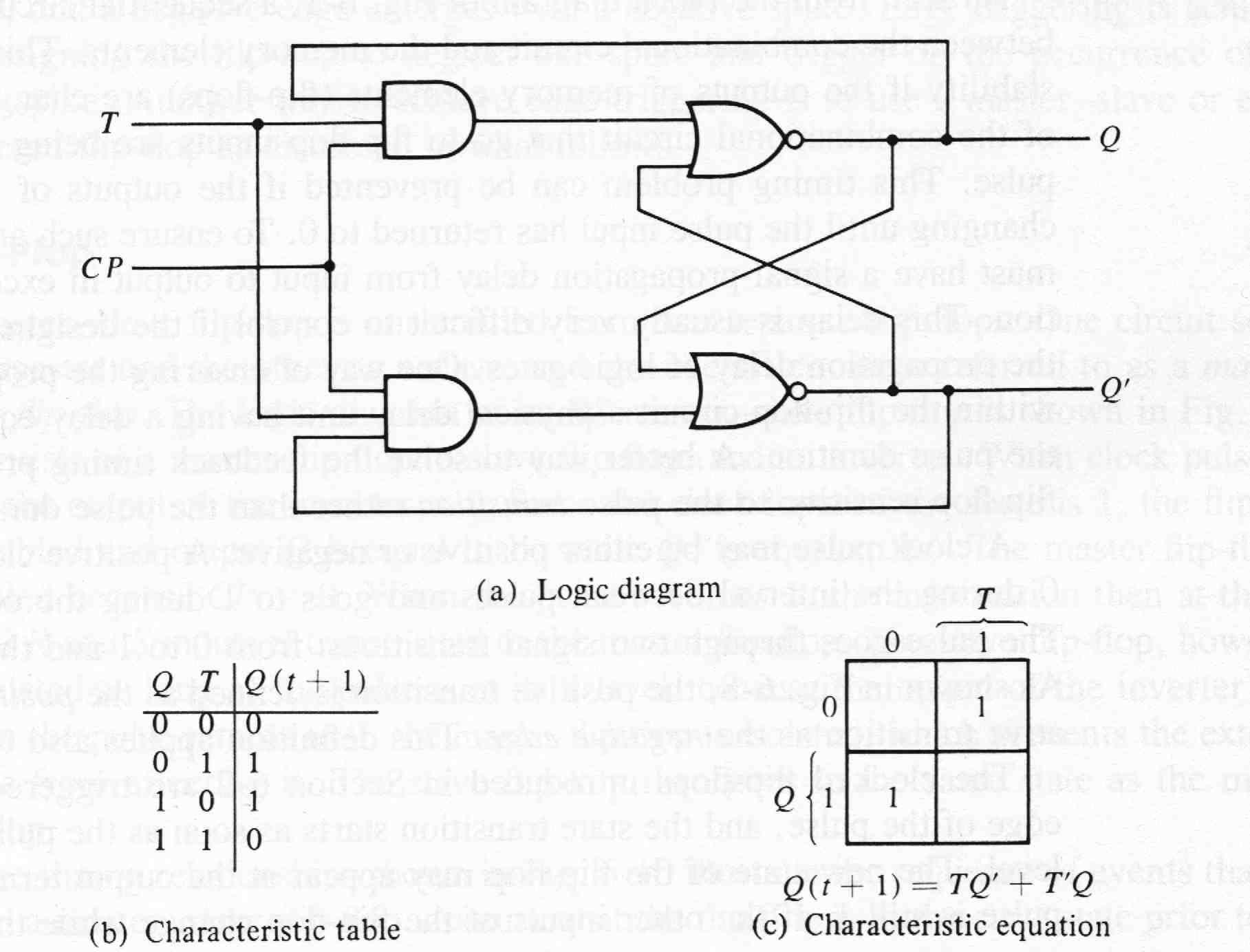
**T触发器 Q(t+1)=TQ’(t)+T’Q(t)**

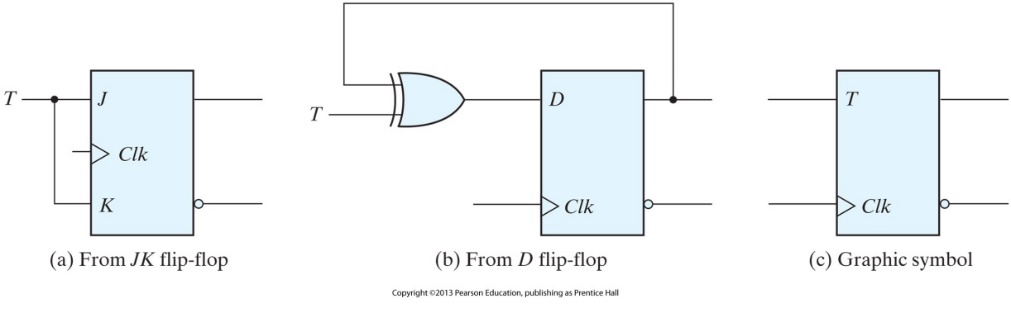
T：翻转信号

T=0维持上一次的状态

T=1翻转状态→Q(t+1)=Q’(t)

T触发器可以用JK或D触发器实现





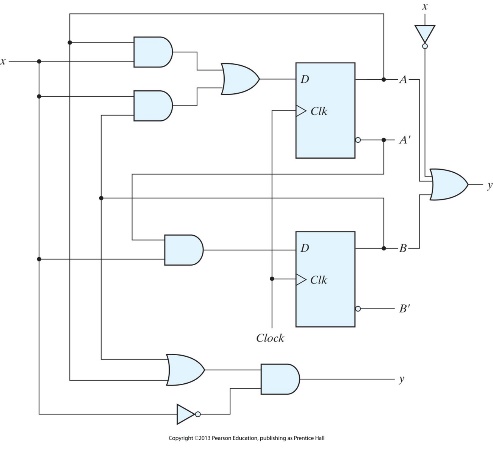
时序电路状态分析

时序电路的行为由它的输入，输出和触发器的当前状态定义

时序电路的分析包括一个描述状态变化的表或者图

* 状态方程(State equation)：一种指定触发器状态转换条件的代数表达式
* 状态表(State table)：枚举输入、输出和触发器状态的时间序列
* 状态图(State diagram)：图解
* 触发器输入函数：描述电路中产生触发器输入的部分的布尔函数
* 特征表(Characteristic table)：描述触发器的输入和下一个状态的关系的表

**分析电路：**



**第一步：确认输出函数(状态方程1) 现在的状态下遇到x输出什么**

y(t)=[A(t)+B(t)]x’

**第二步：确定触发器的输入函数**

DA = A(t)x + B(t)x

DB = A’(t)x

**第三步：确定触发器的特征方程**

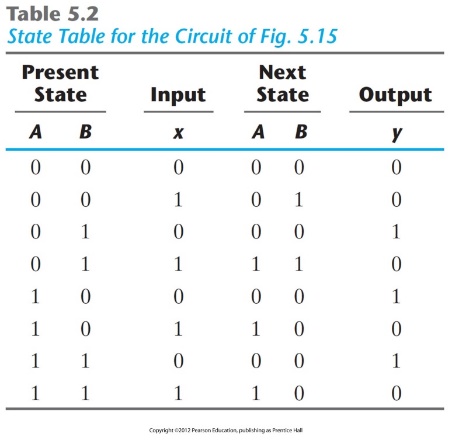
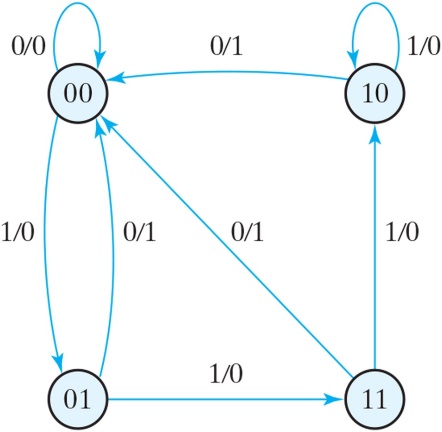
D触发器，Q(t+1) = D

**第四步：确定触发器的输出函数(状态方程2) 状态如何进行转变的2**

QA(t+1) = A(t+1) = DA = A(t)x + B(t)x

QB(t+1) = B(t+1) = DB = A’(t)x

**给出状态表 画出状态图**

时序逻辑电路设计

## CH6 寄存器和计数器

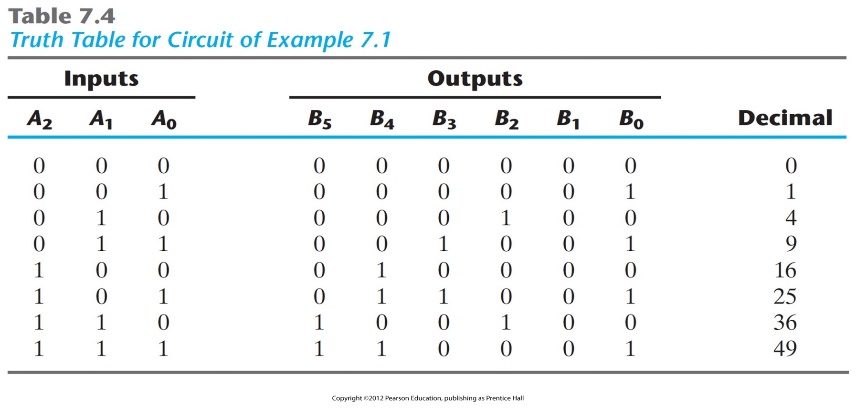
## CH7 可编程逻辑器件

ROM设计组合逻辑电路

**设计电路：该电路输入是3bit二进制数，输出是输出值的平方**

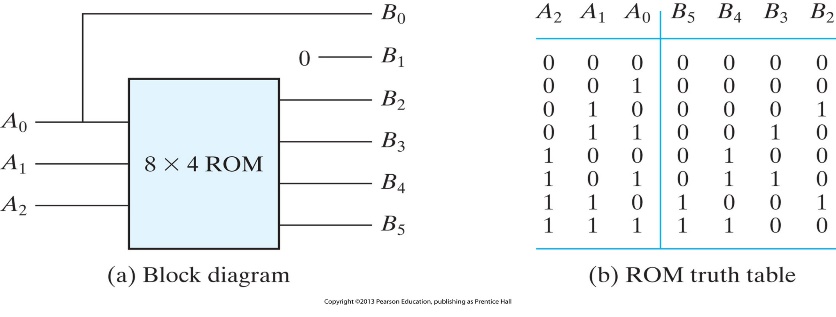
**第一步：给出电路真值表**

列出组合电路的真值表

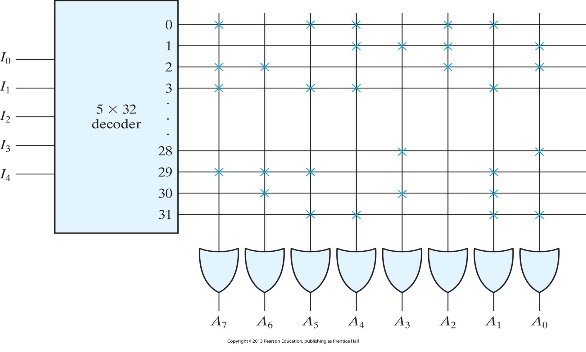
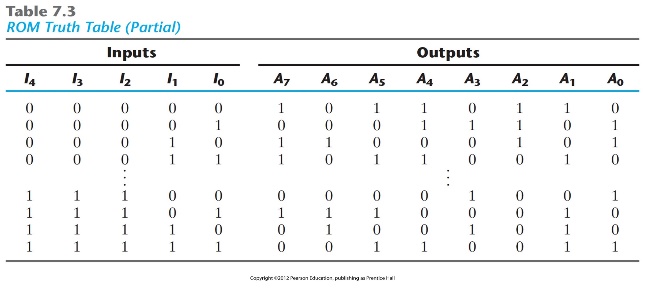


**第二步：确定输入和输出，ROM真值表**

观察可得，输出B0等于输入A0，B1的输出一直为0，实际上用ROM产生的输出只有4个，其它2个输出可以直接得到。本例中，ROM必须至少有3个输入端和4个输出端



**第三步：根据ROM真值表画出ROM熔断电路图**



用PLA实现布尔函数1

**设计电路：**

**F1 = AB’ + AC + A’BC’**

**F2 = (AC + BC)’**

**第一步：确定有几个变量**

观察可得，一共有3个变量A，B，C

则开始一共画6条线

**第二步：确定SOP中包含几个乘积项**

这里一共有AB’，AC，BC，A’B’C四项

则有4个于门，4条横线

**第三步：确定函数的SOP需要的各项**

F1 = AB’+AC+A’B’C

F2 = (AC+BC)’

则有4个或门，2条竖线

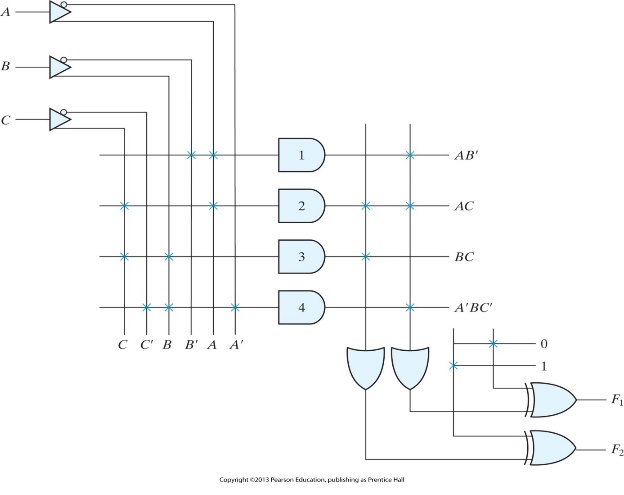
**第四步：异或门取反/维持不变**

输出与0异或保持不变

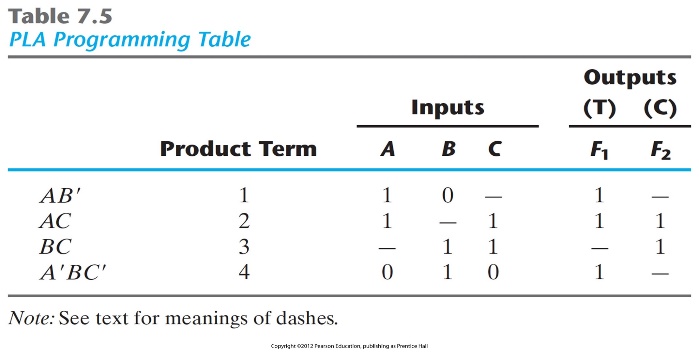
输出与1异或取反

F1输出不变

F2输出取反



PLA编程表



用PLA实现布尔函数2

**设计电路：（用PLA编程表表示）**

**F1 (A, B, C) = Σ(0, 1, 2, 4)**

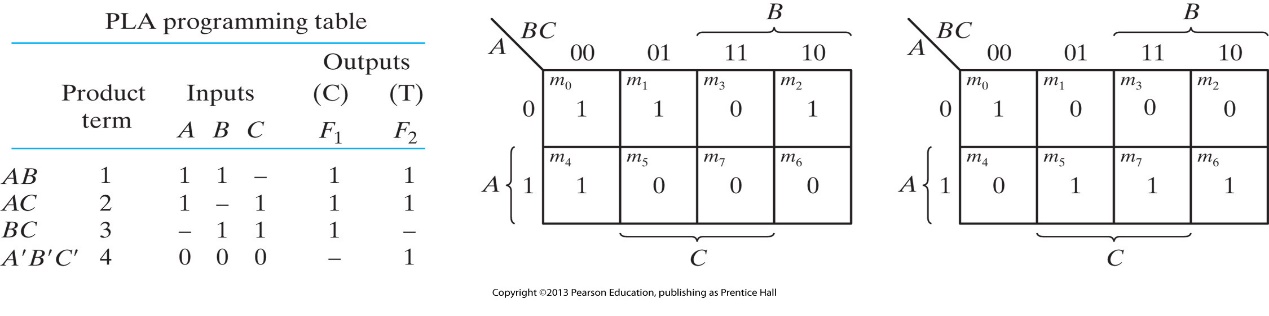
**F2 (A, B, C) = Σ(0, 5, 6, 7)**

两个函数通过卡诺图进行简化，得到乘积最少的函数表达式为：

F1=(AB+AC+BC)’

F2=AB+AC+A’B’C’

F1在异或门对其取反得到



用PAL实现布尔函数

设计电路：

考虑以下布尔函数，以sum‐of‐minterms形式给出:

w(A, B, C, D) =Σ (2, 12, 13)

x(A, B, C, D) =Σ (7, 8, 9, 10, 11, 12, 13, 14, 15)

y(A, B, C, D) =Σ(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)

z(A, B, C, D) =Σ(1, 2, 8, 12, 13)

**第一步：四个函数简化为最小数量的项会得到以下布尔函数**

w = ABC’ + A’B’CD

x = A + BCD

y = A’B + CD + B’D’

z = ABC’ + A’B’CD’ + AC’D’ + A’B’C’D = w + AC’D’ + A’B’C’D

**第二步：确定加和项最大**

这里四个函数最多由3个项构成，则每个输出分配三个AND Inputs

**第三步：画出PAL编程表和熔丝图**

