# 时序逻辑电路

一、概述

1.时序逻辑电路的逻辑描述：







2.时序逻辑电路的分类：

1、按触发器的动作特点可分为：

同步时序逻辑电路——有统一的CP;

异步时序逻辑电路——无统一的CP。

2、按输出信号的特点可分为：

米利型（Mealy）——输出由输入变量和存储电路的原态决定；

莫尔型（Moore）——输出仅取决于存储电路的原态。

可见，莫尔型是米利型的特例。

二、时序逻辑电路的三种基本描述方法

从理论而言：

驱动方程、状态方程、输出方程

从完整性和直观性而言：

状态转换表、状态转换图、时序图

三大方程与三大图表之间可以相互转换。根据状态方程组和输出方程列出状态表

同步时序电路分析的一般步骤：

1、从给定的逻辑图中写出每个触发器的驱动方程；

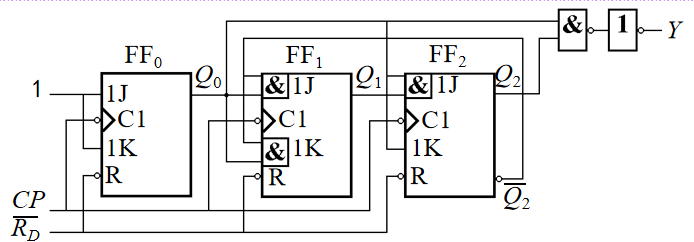
2、把得到的这些驱动方程代入相应触发器的特性方程，得出每个触发器的状态方程，从而得到由这些状态方程组成的整个时序电路的状态方程组；

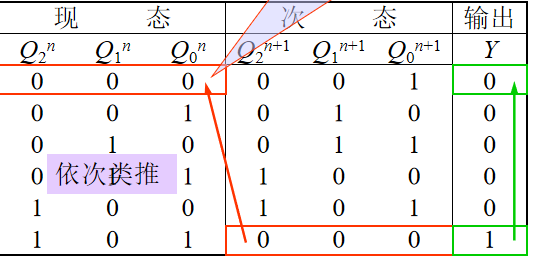
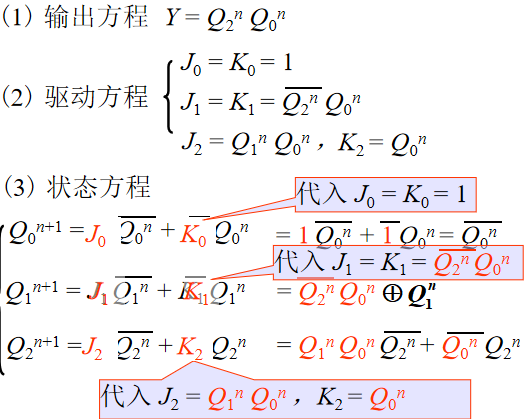
3、根据逻辑图写出电路的输出方程；

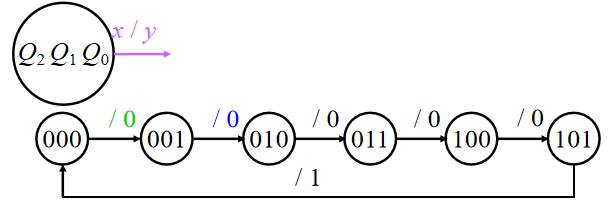
4、列出该电路的状态转换表；

5、根据状态表画出状态转换图（或时序图）；

6、根据图表描述电路的逻辑功能，并进行自启动验证。







三、异步时序逻辑电路分析

异步时序逻辑电路的特点：

1、所有触发器的CP端并没有完全连接在一起；

2、不是所有触发器状态的变化都与时钟脉冲同步；

3、有时钟信号的触发器才需要用特性方程计算次态，而没有时钟信号的触发器将保持原来的状态不变。

分析方法一样

四、同步时序逻辑电路的设计方法与设计实例

例1：例:试设计一个串行数据检测器。对它的要求是：连续输入三个或三个以上的’1’时输出为’1’，其它情况输出为’0’(试用上边沿JK-FF完成设计)。

步骤一：逻辑抽象

具体步骤：

（1）分析给定的逻辑问题，确定输入变量、输出变量以及电路的状态数。

（2）定义输入、输出逻辑状态和每个变量的含意，并将电路状态顺序编号；

（3）按照题意列出电路的原始状态转换表或画出原始状态转换图。

目的——得出电路的原始状态转换图或状态转换表。

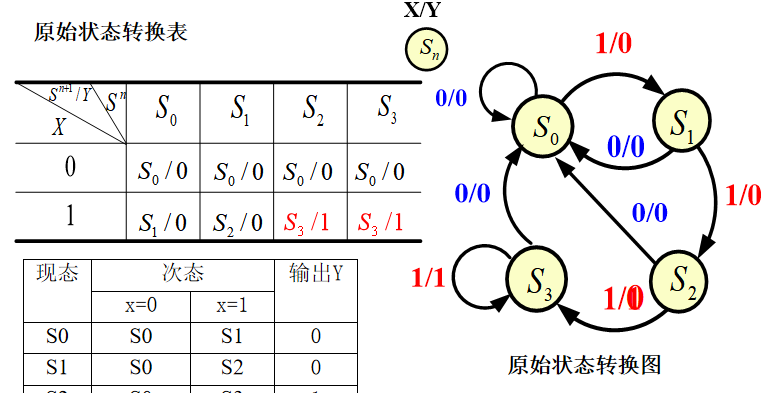
解：**（一）逻辑抽象**

依题意：

令输入数据为输入变量，用X表示；

令检测结果为输出变量，用Y表示；

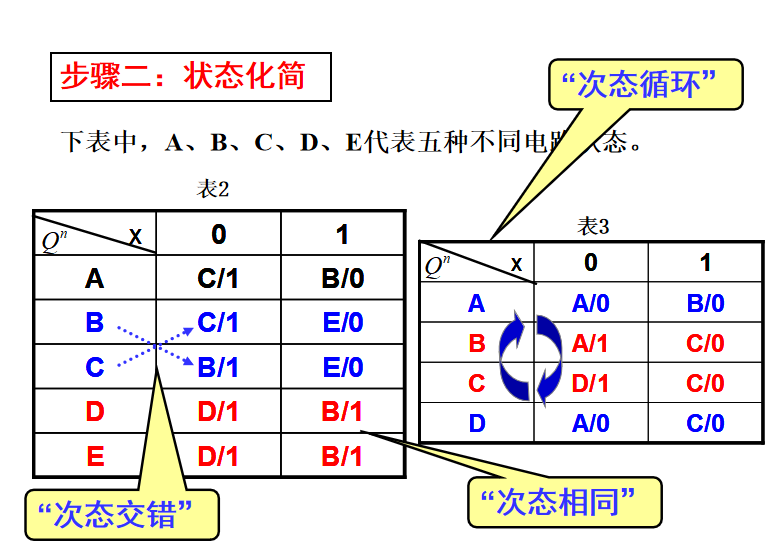
设电路在没有输入‘1’以前的状态为S0；输入1个‘1’后电路转入S1状态；连续输入2个‘1’后电路转入S2状态；连续输入3个或3个以上‘1’以后电路转入S3状态。



注意：先得原始状态图，然后画原始状态表，注意mealy型与moer型的表y位置不同

然后看有没有状态是同构的，因为一个状态需要一个触发器存储，所能消减就少一个

**步骤二：状态化简**



等价（效）状态：

1）输出完全相同。

2） ◆次态“完全相同”；

◆次态形成“交错状态”；

◆次态构成“循环”情况。

目的——合并等价（效）状态，减少状态数。

判断原始状态表中两个状态是否 等效（等价）的标准：

如果两个状态，对每一位可能的输入都满足下列两个条件，则这两个状态等效。

第一，它们的输出完全相同。

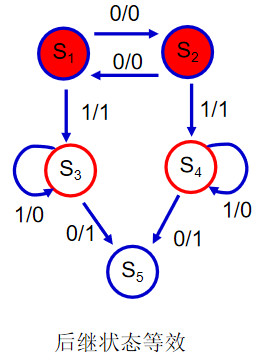
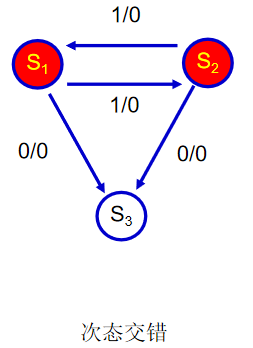
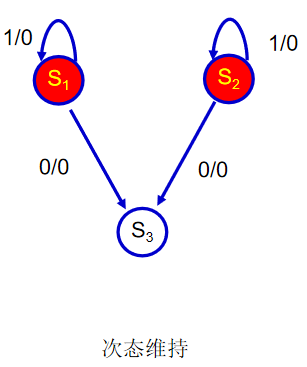
第二，它们的次态属于下列情况之一：

1）次态相同

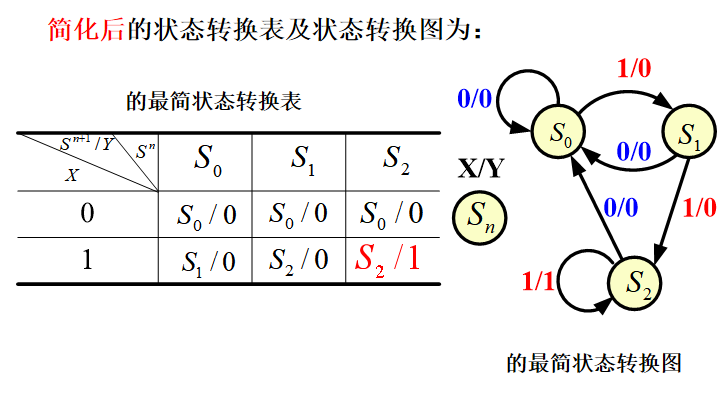
2）次态交错或者次态维持

3）后继状态等效

4）次态循环



解:（二）状态化简



步骤三：状态编码（分配）

状态分配步骤：

1. 需要确定触发器的数目N，用二进制代码表示不同状态

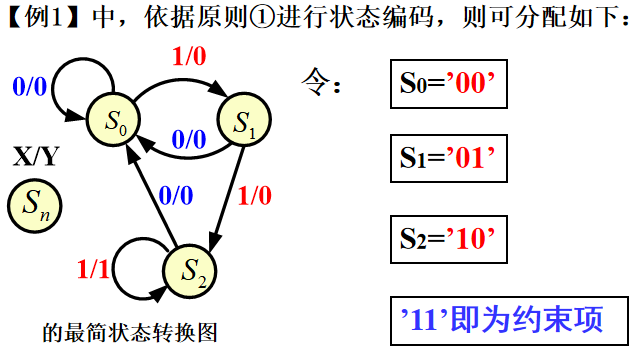
（2）要给每个电路（共M个）状态规定对应的触发器状态组合，每组触发器的状态组合都是一组二值代码。

**注意：编码方案的选择直接影响电路的复杂程度；对异步时序电路而言，有时还会产生竞争—冒险现象。**

状态分配的一般原则—— “相邻分配”：

解:（三）状态编码

∵ 状态数M=3，∴ 触发器数目N=2。



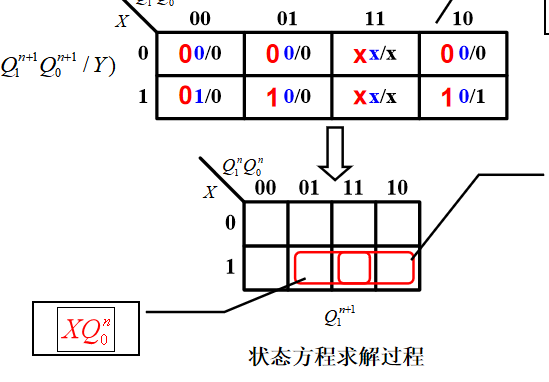
步骤四：选定触发器类型，求解电路的三大方程

选用 JK 触发器。其特性方程为 Qn+1 = J~Qn +~KQn ，

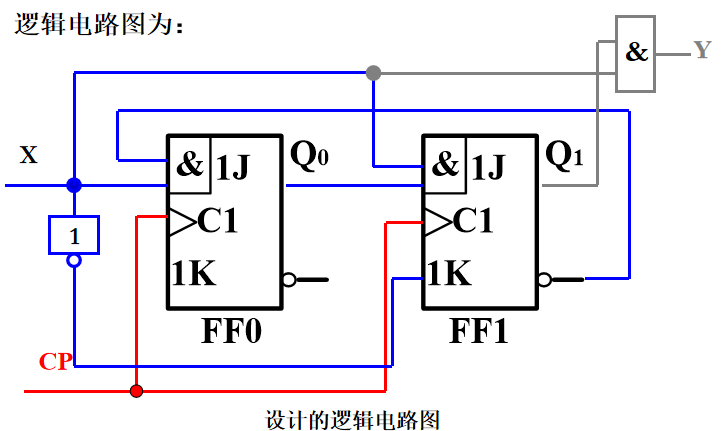
将它与状态方程进行比较，可得驱动方程

解:（四）依题意，选用上边沿JK-FF。

确定电路的状态方程:



步骤五：画出逻辑电路设计图



步骤六：验证功能，并进行自启动检查

六、时序逻辑电路中的竞争－冒险现象

因为时序逻辑电路中通常包含组合逻辑电路和存储电路两个组成部分，所以它的竞争——冒险现象也包含两个方面：

一方面：

组合逻辑电路部分可能发生竞争——冒险现象；

消除方法：1、在输出端并接滤波电容——Cf；

2、加选通脉冲P；

3、修改逻辑设计。

另一方面：

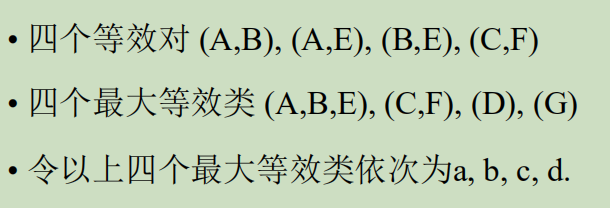
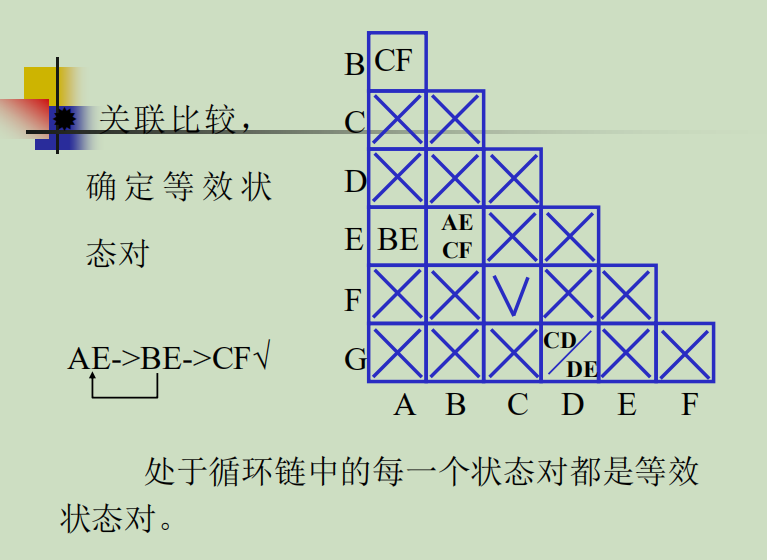
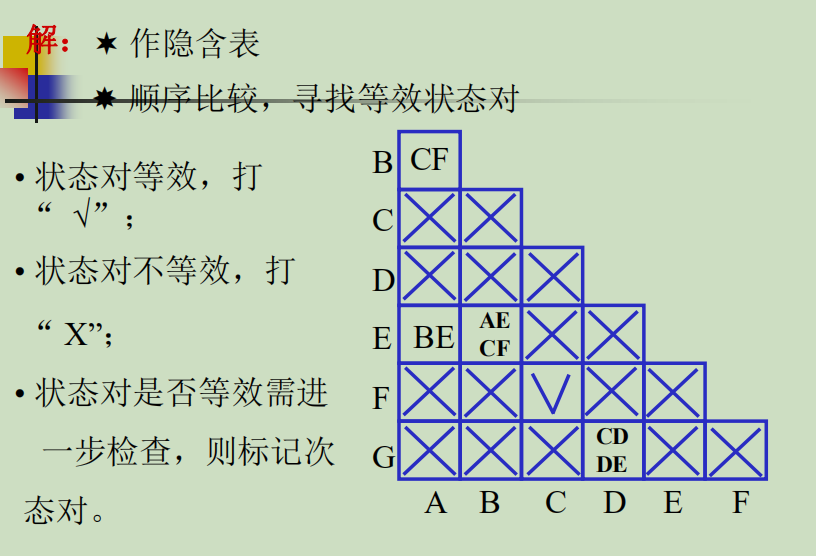
存储电路部分（或触发器）可能发生竞争——冒险现象；

原因：当输入信号和时钟脉冲信号同时改变，而且途经不同路径到达同一触发器时，便会产生竞争。

现象：引起触发器误动作。

在时序逻辑中 一般认为竞争——冒险现象仅发生在异步时序电路中。

**隐含表法化简**

·

书上P197页的**划分算法**，化简状态，永远滴神

**状态分配的基本原则有四条：**

(1) 在相同输入条件下具有相同次态的现态，应尽可能分配相邻的二进制代码；

(2) 在相邻输入条件，同一现态的次态应尽可能分配相邻的二进制代码；

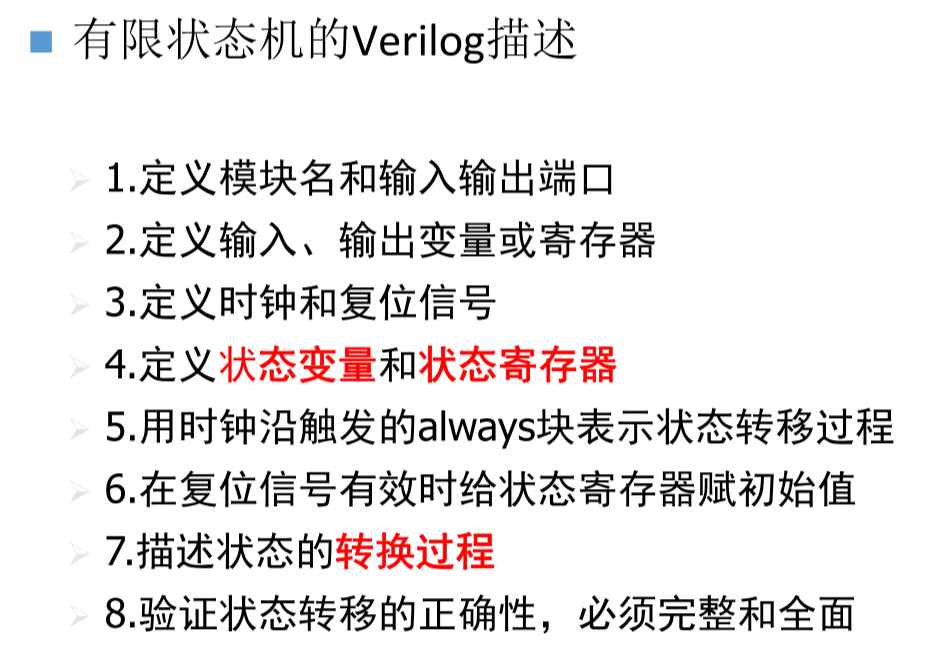
(3) 输出**完全相同**的现态应尽可能分配相邻的二进制代码；

(4) 最小化状态表中出现次数最多的状态或初始状态应分配逻辑0。

一般情况下，第一条原则较为重要，需优先考虑，其次要考虑由前三条原则得到的应分配相邻代码的状态对出现的次数，次数多的

状态对应优先分配相邻的二进制代码。

# Verilog实现FSM



第一段：有效时钟沿的现态变化（时序逻辑）

第二段：产生下一个状态的组合逻辑！（注：是组合逻辑，记得添加else和default，避免产生不必要的时序逻辑）

第三段：输出信号的组合逻辑，如果有多个不同的输出信号，可以分不同的always模块来写

注：如果把状态变化（现态和次态）写在一个always模块里，就不需要next的寄存器，直接一个寄存器保存curr，使用非阻塞赋值（时序逻辑）

