

# Part III 触发器与时序逻辑OK!

## Lecture 12 常用时序逻辑电路 1

数字系统中最典型的时序逻辑电路是计数器和寄存器。

### 一、计数器

**计数器**是能够记忆输入脉冲个数的电路，也可用来分频、定时、产生节拍脉冲和进行数字运

算等。

计数器有多种不同的分类方法，按时钟控制方式的不同，可分为**同步计数器**和**异步计数器**；按照计数数制的不同，可分为**二进制计数器**、**十进制计数器**和**N 进制（任意进制）计数器**；按照计数方式的不同，可分为**加法计数器**、**减法计数器**和**可逆计数器**。

## (一) 双时钟可逆十六进制计数器 74LS193

常用**同步集成计数器**芯片有 4 位二进制加法计数器 74LS161 和 74LS163, 4 位十进制加法计数器 74LS160 和 74LS162; 4 位十进制加/减法计数器 74LS190 和 74LS192; 4 位同步二进制加/减法计数器 74LS191 和 74LS193 等。

常用**异步集成计数器**有十进制计数器 74LS196、74LS290; 二进制计数器 74LS177、

74LS197、74LS293、74LS393 等。

**74LS193** 是可预置的 4 位二进制同步加/减计数器。图 5.5.3 是 74LS193 的引脚排列和逻辑符号。

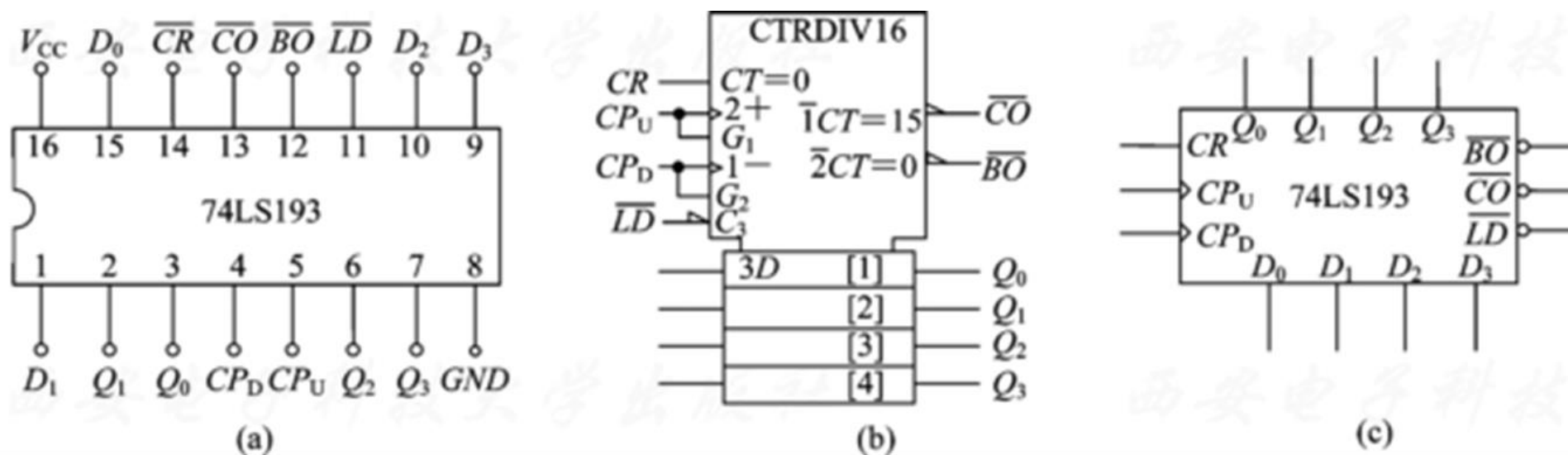


图5.5.3 74LS193的引脚排列和逻辑符号

(a) 引脚排列图; (b) 国标逻辑符号; (c) 惯用符号

图中， $CR$  是异步清零端； $\overline{LD}$  是异步置数端； $CP_U$  是加法计数脉冲输入端； $CP_D$  是减法计数脉冲输入端； $\overline{CO}$  是进位脉冲输出端，当加法计数上溢（计数值为 1111）时输出一个宽度等于  $CP_U$  的低电平部分的低电平脉冲； $\overline{BO}$  是借位脉冲输出端，当减法计数下溢（计数值为 0000）时输出一个宽度等于  $CP_D$  的低电平部分的低电平脉冲。

表5.5.3 74LS193的逻辑功能表

$CR$	$\overline{LD}$	$CP_U$	$CP_D$	$D_3$	$D_2$	$D_1$	$D_0$	$Q_3^{n+1}$	$Q_2^{n+1}$	$Q_1^{n+1}$	$Q_0^{n+1}$
1	×	×	×	×	×	×	×	0	0	0	0
0	0	×	×	$d_3$	$d_2$	$d_1$	$d_0$	$d_3$	$d_2$	$d_1$	$d_0$
0	1	↑	0	×	×	×	×	加 1 计数			
0	1	0	↑	×	×	×	×	减 1 计数			
0	1	×	×	×	×	×	×	保持			

## （二）用集成计数器实现任意进制计数

### 1、反馈清零法构成 $M$ ( $M < N$ ) 进制计数器

这种方法适用于有**清零控制端**或**置数控制端**（置数端输入 0）的集成计数器。

根据清零控制方式的不同，可有两种设计方法。这里，设  $M$  进制计数器的  $M$  个计数状态是  $S_0 \sim S_{M-1}$ 。

1) 用异步清零端或置数端归零构成  $M$  进制计数器。

若选用计数器件的清零端或置数端是异步的，则当计数处于  $S_M$  状态时，产生清零信号实现计数器状态清零，这样就可以跳过  $N-M$  个状态，其状态变化如图 5.5.8(a)所示： $S_M$  状态称为暂态，计数器的有效状态为  $S_0 \sim S_{M-1}$ 。



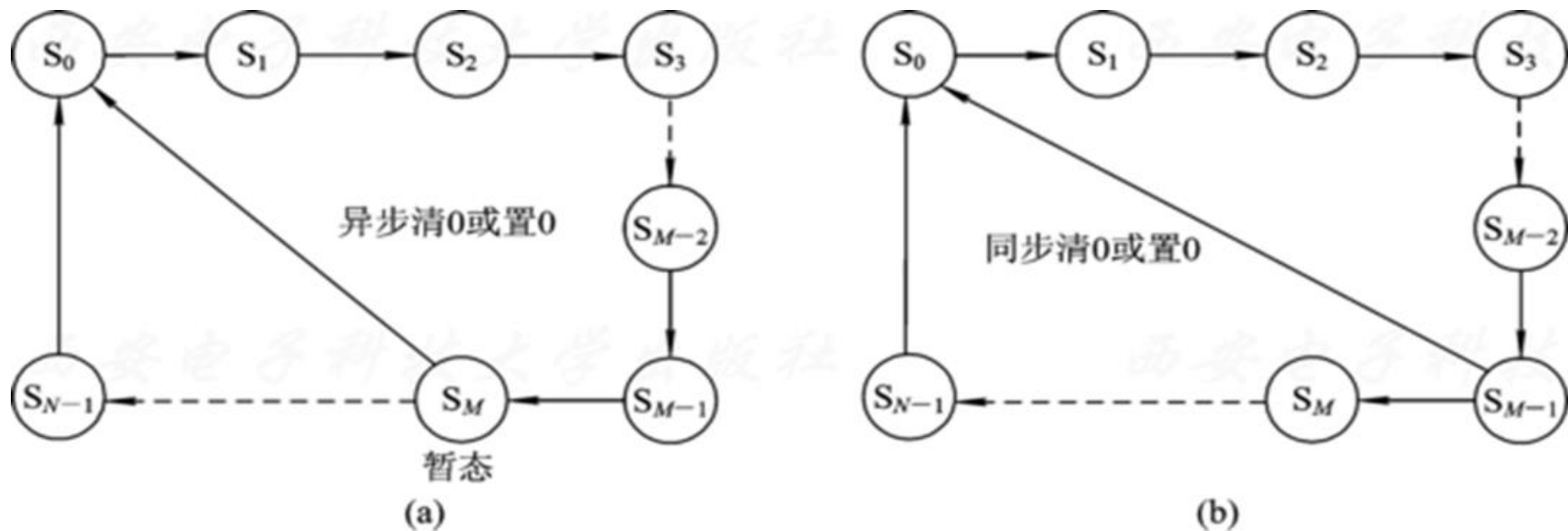


图5.5.8 反馈清零法状态转换示意图

(a) 异步清零法; (b) 同步置数法

用置数控制端清零时，将计数器置数输入端接数据 0。

具体设计**步骤**为：

①写出状态暂态  $S_M$  的二进制代码。

②求归零逻辑，即异步清零端或置数控制端信号的逻辑表达式。

③画出电路连线图。

2) 用**同步清零端或置数端归零**构成  $M$  进制计数器

选用同步清零或置数计数器时，当计数状态处于  $S_{M-1}$  状态时，产生芯片的同步清零信号，在下一个时钟脉冲转换到零状态，其状态变化示意图如图 5.5.8(b)所示。用置数控制端清零时，将计数器置数输入端接数据 0。

具体设计**步骤**为：

①写出状态  $S_{M-1}$  的二进制代码。

②求归零逻辑。

③画出电路连线图。

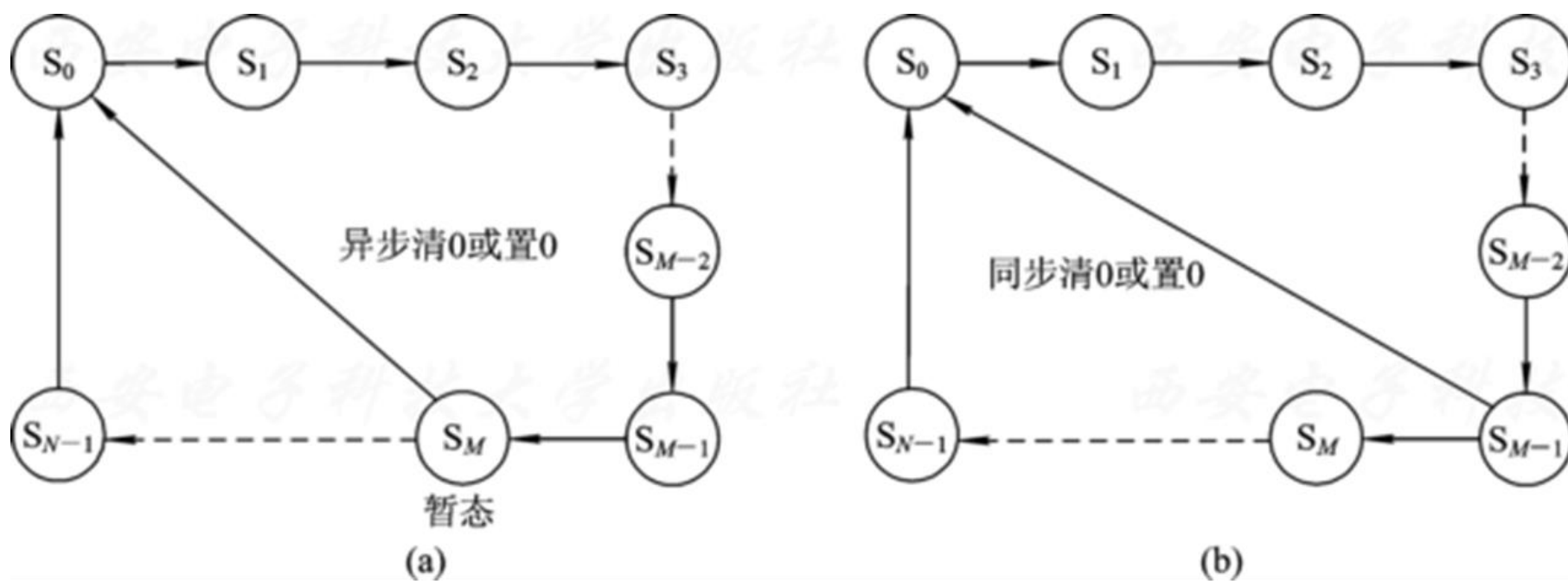


图5.5.8 反馈清零法状态转换示意图

(a) 异步清零法; (b) 同步置数法

采用异步方式实现计数器清零存在可靠性不高的问题，因为清零信号随着计数器被清零而消失，如果触发器的复位或置位速度有快有慢，则可能慢的还未复位，清零信号就消失了，导致误动作。

可以加入  $R-S$  触发器来避免这种现象的发生。

## 2、反馈置数法构成 $M$ ( $M < N$ ) 进制计数器

这种方法适用于具有**置数功能**的集成计数器。在计数器计数过程中，置数功能可以使计数器跳过  $N - M$  个状态，计数器的状态变化如图 5.5.13 所示。计数器可以对任意一个状态进行译码，产生一个置数控制信号，并将之反馈至置数控制端。若置数端是同步的，则在下一个  $CP$  脉冲作用后，计数器的状态进入预置状态；若置数端是异步的，

则计数器的状态立即进入预置状态。当置数控制信号消失后，计数器就从被置入的状态开始重新计数，从而跳过  $N-M$  个状态。

### 1) 用异步置数端构成 $M$ 进制计数器

状态变化示意图如图 5.5.13(a)所示。若采用异步置数端反馈构成计数器，则当计数状态处于  $S_{i+1}$  时，产生的置数信号将计数器状态立即置为  $S_j$ ，这样就跳过  $N-M$  个状态。在这种情况下， $S_{i+1}$  状

态称为**暂态**，由于在  $S_{i+1}$  状态产生反馈信号，因此  $S_{i+1}$  又称为**反馈态**； $S_j$  是计数器被置数的状态，称为**预置态**。

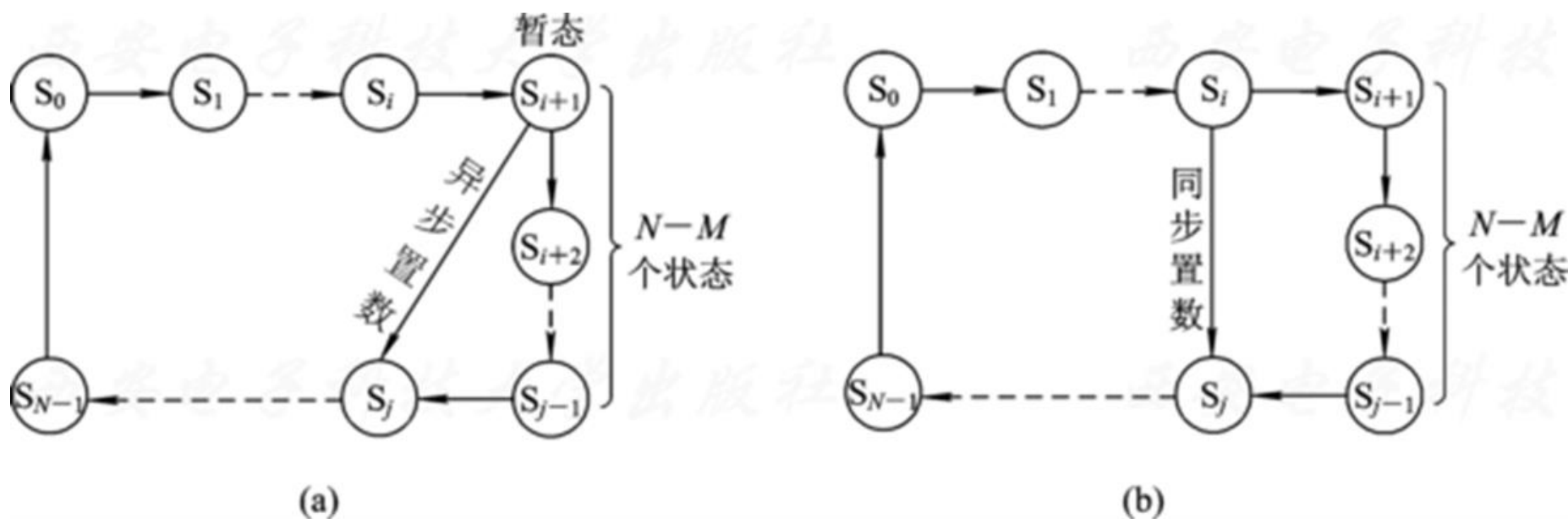


图5.5.13 反馈置数法状态转换示意图

(a) 异步置数法； (b) 同步置数法



具体设计步骤为：

①确定电路计数状态。

②确定反馈态  $S_{i+1}$  的二进制代码。

③求置数逻辑，即求异步置数控制端信号的逻辑表达式。

④数据输入端接预置态  $S_j$ ，画出电路连线图。

需要注意的是，异步置数同样存在可靠性不

高的问题。

## 2) 用同步置数端构成 $M$ 进制计数器

状态变化示意图如图 5.5.13(b)所示, 若采用同步置数端构成计数器, 则当计数状态处于  $S_i$  状态时, 产生反馈置数信号, 在下一个时钟脉冲有效时, 将计数器状态置为  $S_j$ , 以跳过  $N-M$  个状态。

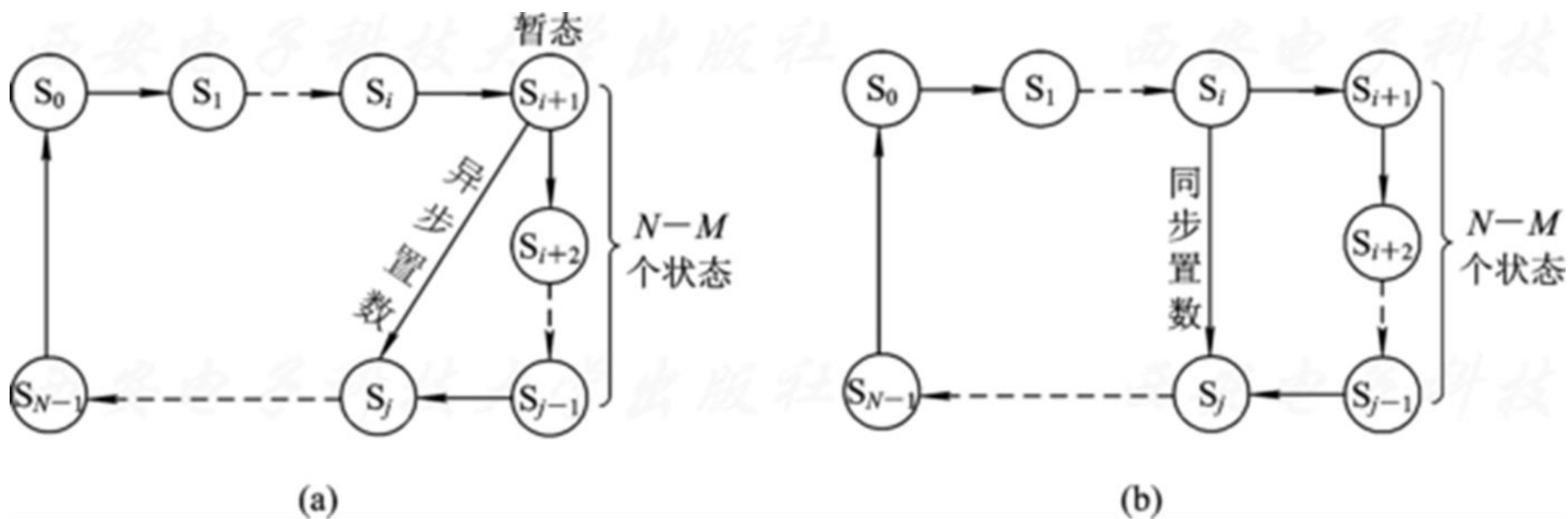


图5.5.13 反馈置数法状态转换示意图

(a) 异步置数法; (b) 同步置数法

具体设计**步骤**为:

①确定电路计数状态。

②确定反馈态  $S_i$  的二进制代码。

③求置数逻辑。

④数据输入端接预置态  $S_j$ ，画出电路连线图。

**状态跳跃的方法有三种：**

**方法一：** 选用前面的状态，跳过后头的状态

**方法二：** 选用中间的状态，跳过两头的状态

**方法三：** 选用后面的状态，跳过前头的状态

【例如】用 74LS193 构成七进制计数器。

74LS193 是具有异步清零和异步置数控制端的十六进制双时钟同步加/减计数器，可以构成加法或减法两种计数器。

为了简化电路设计，可以利用借位/进位输出端  $\overline{CO}/\overline{BO}$ 。

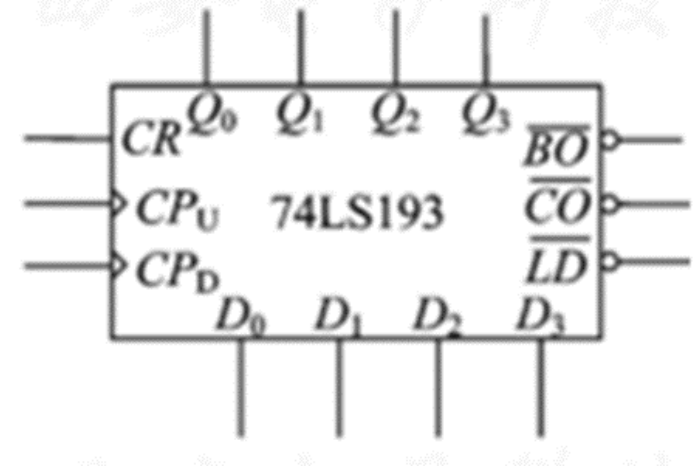
# 方法一：利用异步清零构成加法计数器

$CR$	$\overline{LD}$	$CP_U$	$CP_D$	$D_3$	$D_2$	$D_1$	$D_0$	$Q_3^{n+1}$	$Q_2^{n+1}$	$Q_1^{n+1}$	$Q_0^{n+1}$
1	×	×	×	×	×	×	×	0	0	0	0
0	0	×	×	$d_3$	$d_2$	$d_1$	$d_0$	$d_3$	$d_2$	$d_1$	$d_0$
0	1	↑	0	×	×	×	×	加 1 计数			
0	1	0	↑	×	×	×	×	减 1 计数			
0	1	×	×	×	×	×	×	保持			

(1) 选择 8 个计数状态，即  $Q_3Q_2Q_1Q_0$  为 0000→0111，有效计数范围为 0~6，状态 0111 为暂态。

(2) 确定清零逻辑：  $CR = \bar{Q}_3Q_2Q_1Q_0$

(3) 电路连线：略！



## 方法二：利用异步清零构成减法计数器

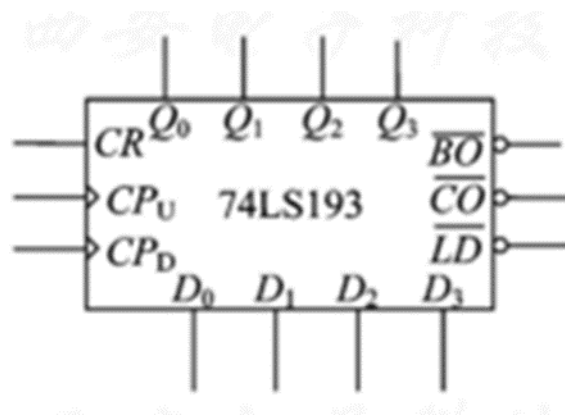
$CR$	$\overline{LD}$	$CP_U$	$CP_D$	$D_3$	$D_2$	$D_1$	$D_0$	$Q_3^{n+1}$	$Q_2^{n+1}$	$Q_1^{n+1}$	$Q_0^{n+1}$
1	×	×	×	×	×	×	×	0	0	0	0
0	0	×	×	$d_3$	$d_2$	$d_1$	$d_0$	$d_3$	$d_2$	$d_1$	$d_0$
0	1	↑	0	×	×	×	×	加 1 计数			
0	1	0	↑	×	×	×	×	减 1 计数			
0	1	×	×	×	×	×	×	保持			



(1) 选择 8 个计数状态，即  $Q_3Q_2Q_1Q_0$  为 0000→1001，有效计数状态为 0000~1010，状态 1001 为暂态。

(2) 确定清零逻辑：  $CR = Q_3\bar{Q}_2\bar{Q}_1Q_0$

(3) 电路连线：略！



## 方法三：利用异步置数构成加法计数器

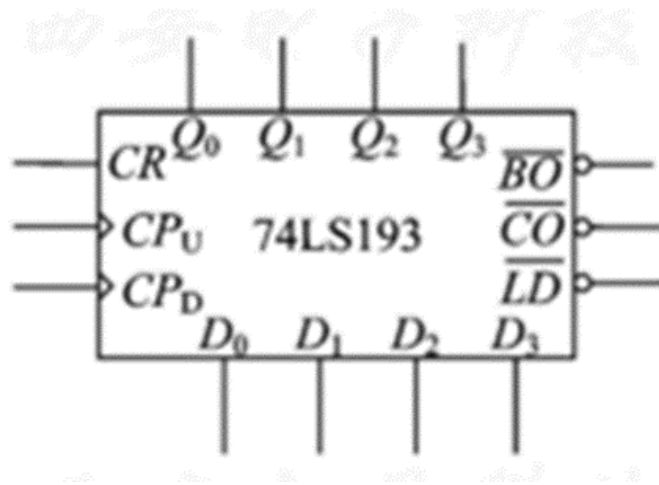
$CR$	$\overline{LD}$	$CP_U$	$CP_D$	$D_3$	$D_2$	$D_1$	$D_0$	$Q_3^{n+1}$	$Q_2^{n+1}$	$Q_1^{n+1}$	$Q_0^{n+1}$
1	×	×	×	×	×	×	×	0	0	0	0
0	0	×	×	$d_3$	$d_2$	$d_1$	$d_0$	$d_3$	$d_2$	$d_1$	$d_0$
0	1	↑	0	×	×	×	×	加 1 计数			
0	1	0	↑	×	×	×	×	减 1 计数			
0	1	×	×	×	×	×	×	保持			

选择 8 个计数状态，有多种选择。

选择 1：选用前 8 个状态，即  $Q_3Q_2Q_1Q_0$  为 0000→0111，其中：0111 为暂态。

确定置数逻辑： $\overline{LD} = \overline{\overline{Q}_3Q_2Q_1Q_0}$

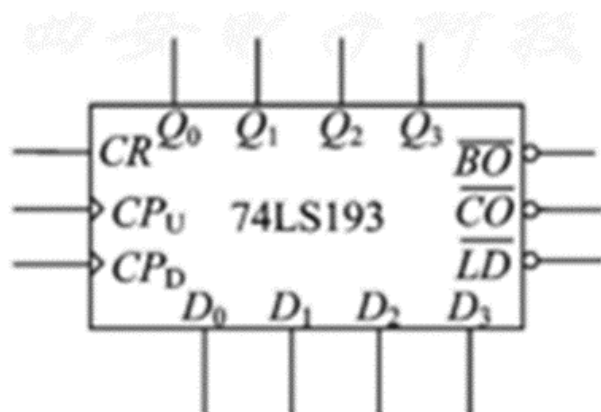
电路连线：略！



选择 2：选用后 8 个状态，即  $Q_3Q_2Q_1Q_0$  为 1000→1111，其中：1111 为暂态。

置数逻辑：  $\overline{LD} = \overline{Q_3Q_2Q_1Q_0} = \overline{CO}$

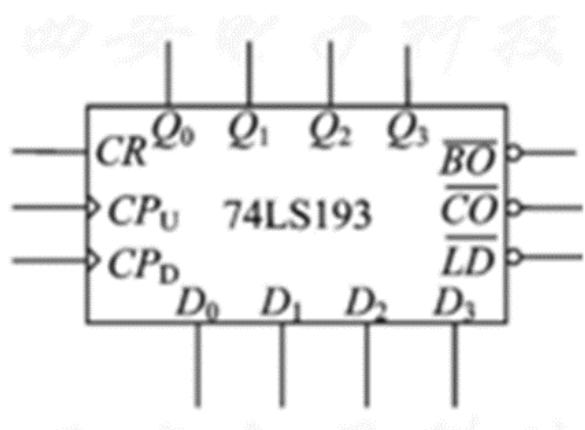
电路连线：略！



选择 3：选用中间 8 个状态，比如  $Q_3Q_2Q_1Q_0$  为 0100→1011，其中：1011 为暂态。

置数逻辑：  $\overline{LD} = \overline{Q_3\overline{Q_2}Q_1Q_0}$

电路连线：略！



## 方法四：利用异步置数构成减法计数器

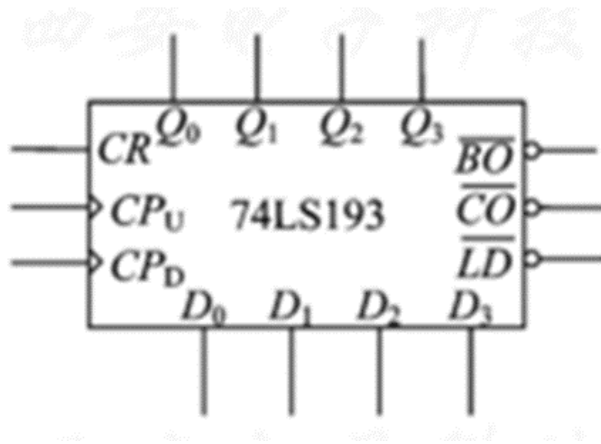
$CR$	$\overline{LD}$	$CP_U$	$CP_D$	$D_3$	$D_2$	$D_1$	$D_0$	$Q_3^{n+1}$	$Q_2^{n+1}$	$Q_1^{n+1}$	$Q_0^{n+1}$
1	×	×	×	×	×	×	×	0	0	0	0
0	0	×	×	$d_3$	$d_2$	$d_1$	$d_0$	$d_3$	$d_2$	$d_1$	$d_0$
0	1	↑	0	×	×	×	×	加 1 计数			
0	1	0	↑	×	×	×	×	减 1 计数			
0	1	×	×	×	×	×	×	保持			

选择 8 个计数状态，有 3 种选择。

选择 1：选择前 8 个状态，即  $Q_3Q_2Q_1Q_0$  为 0000→1001，状态 1001 为暂态。

置数逻辑：  $\overline{LD} = \overline{Q_3\overline{Q_2}\overline{Q_1}Q_0}$

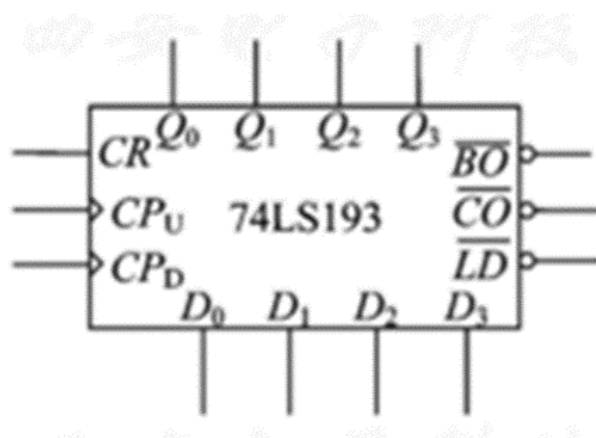
电路连线：略！



选择 2：选用后 8 个状态，比如  $Q_3Q_2Q_1Q_0$  为 0111→0000，其中：0000 为暂态。

置数逻辑：  $\overline{LD} = \overline{\overline{Q_3}\overline{Q_2}\overline{Q_1}\overline{Q_0}} = \overline{BO}$

电路连线：略！

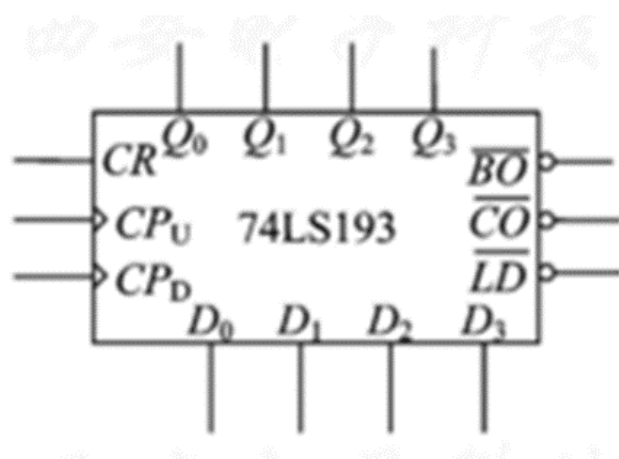




选择 3：选用中间 8 个状态，比如  $Q_3Q_2Q_1Q_0$  为  $1011 \rightarrow 0100$ ，其中：0100 为暂态。

置数逻辑：  $\overline{LD} = \overline{\overline{Q}_3\overline{Q}_2\overline{Q}_1\overline{Q}_0}$

电路连线：略！



### 3、多芯片级联 ( $M > N$ )

当计数器的计数值  $M$  大于集成器件的计数值  $N$  时，需要用级联多个芯片的方式构成计数器。

各级芯片可以先分别实现各子计数器  $M_i$ ，然后级联构成模  $M$  计数器。根据  $M$  值的不同，构成计数器的方法可分为**分解法**和**扩展法**。

## 1) 分解法

将  $M$  进行分解, 若  $M=N_2 \times N_1$  存在, 则  $M$  可以分解为  $N_2$ 、 $N_1$  进制计数器。假设  $N_1$  是低位芯片的进制,  $N_2$  是高位芯片的进制, 则在这种情况下可以先分别实现各子计数器  $N_2$ 、 $N_1$ , 然后级联构成模  $M$  计数器。

芯片级联的方法有并行进位法和串行进位法两种。

**并行进位法**就是  $N_2$ 、 $N_1$  芯片共用一个时钟  $CP$ , 同时  $N_1$  芯片的进位输出控制  $N_2$  芯片的使能端, 各芯片工作在同步方式。

**串行进位法**是指将  $N_1$  芯片的进位信号作为  $N_2$  芯片的  $CP$ 。

## 2) 扩展法

当  $M$  不能分解为  $N_2 \times N_1$  时, 可以先用多级计数器级联成一个  $K$  ( $K > M$ ) 进制的计数器, 然后再用前面介绍过的反馈清零法和反馈置数法实现  $M$  进制计数器。

**【例如】** 用 74LS193 构成 32 进制计数器

分析：用 2 片 74LS193

方法 1：  $32 = 4 \times 8$ ，先将 1 片改造成模 4 计数器，另一片改造成模 8 计数器，然后两片级联。

方法 2：先将两片级联构成模 256 计数器，然后改造成模 32 计数器。

在反馈置数法中，如果预置入的数据是可变的，那么实现计数器的进制数就会发生改变。通常用计算机输出数据来控制预置数，可以方便地实现各种进制的计数器，因此常把这类计数器称为**程控计数器**或**程控分频器**。

设预置数为  $Y$ ，计数器模数为  $M$ ，级联的芯片数为  $k$ ，则三者之间的关系为：
$$Y=16^k-M$$

**WHY?**

**程控计数器的连接方法**本质上相当于每个计数循环开始时给计数器置入一个数据,计  $M$  个  $CP$  脉冲后,计数器就达到满量程( $16^k$ ),产生进位输出,利用进位输出产生置位信号,使计数器重新开始新一轮计数。这种计数器是利用进位输出产生置数信号的,所以只能使用  $16^k$  个状态中后面  $M$  个状态构成计数循环。



例如，要构成模  $M=200$  的计数器，需要 2 片 74LS161，预置数为

$$Y=16^2-200=(56)_{10}=(0011\ 1000)_2$$