

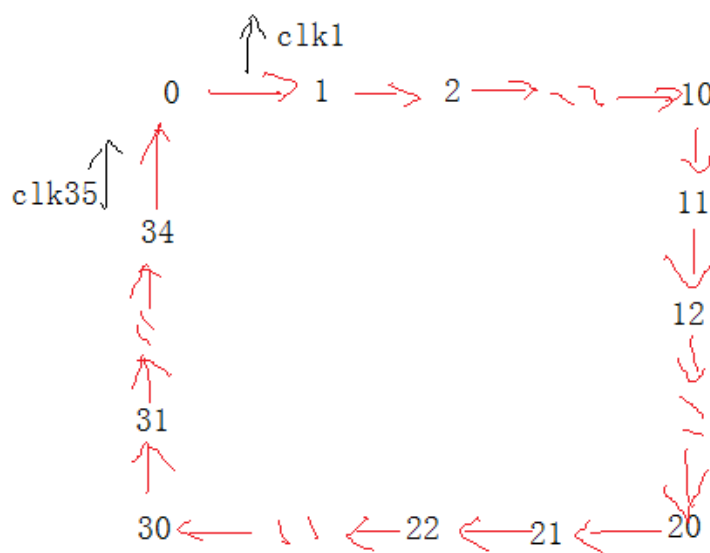
1、用 74LS161 设计一个 35 进制加 1 计数器。

表 5.5.1 74LS161 的逻辑功能表

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

参考解答：可以利用异步清零功能，也可以利用同步置数功能，以下利用同步置数实现归零。

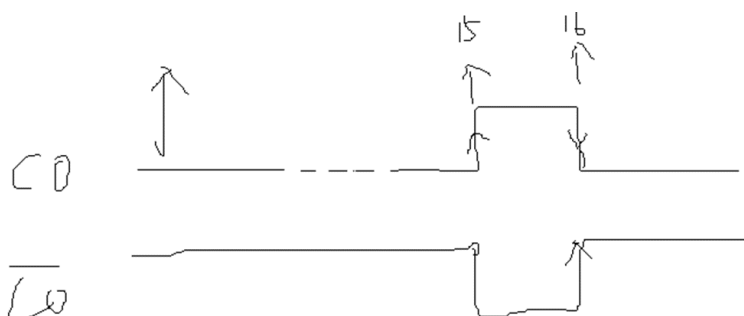
首先确定计数循环：



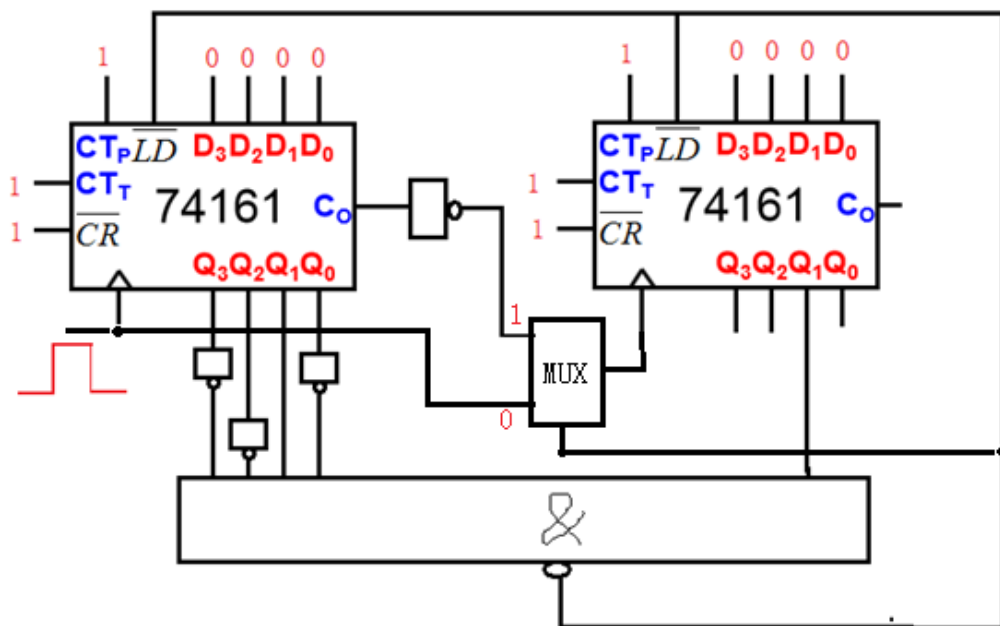
用两片级联，需要保证在低位片归 0 的同时高位片加 1。

1) 串行进位：利用低位片的进位输出为高位片提供时钟输入

由于时钟输入是上升沿有效，低位片的进位输出经过反相器反向后送给高位片作为高位片的时钟输入。

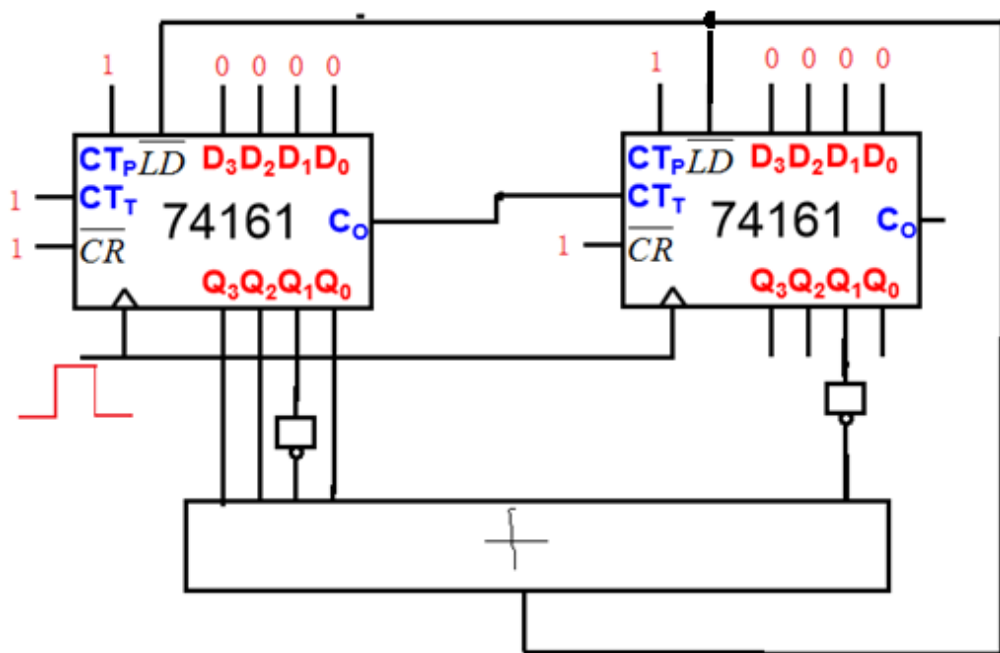


等到计数值变为 34 (2 进制的 0010 0010) 时, 生成有效的置数控制信号, 在第 35 个时钟的作用下同步置 0。



说明：与非门的输出同时作为选择器 MUX 的选择控制信号：0 选择时钟输入，1 选择低位片进位输出的非。

2) 并行进位：低位片的时钟输入同时作为高位片的时钟输入，利用低位片的进位输出为高位片提供使能输入

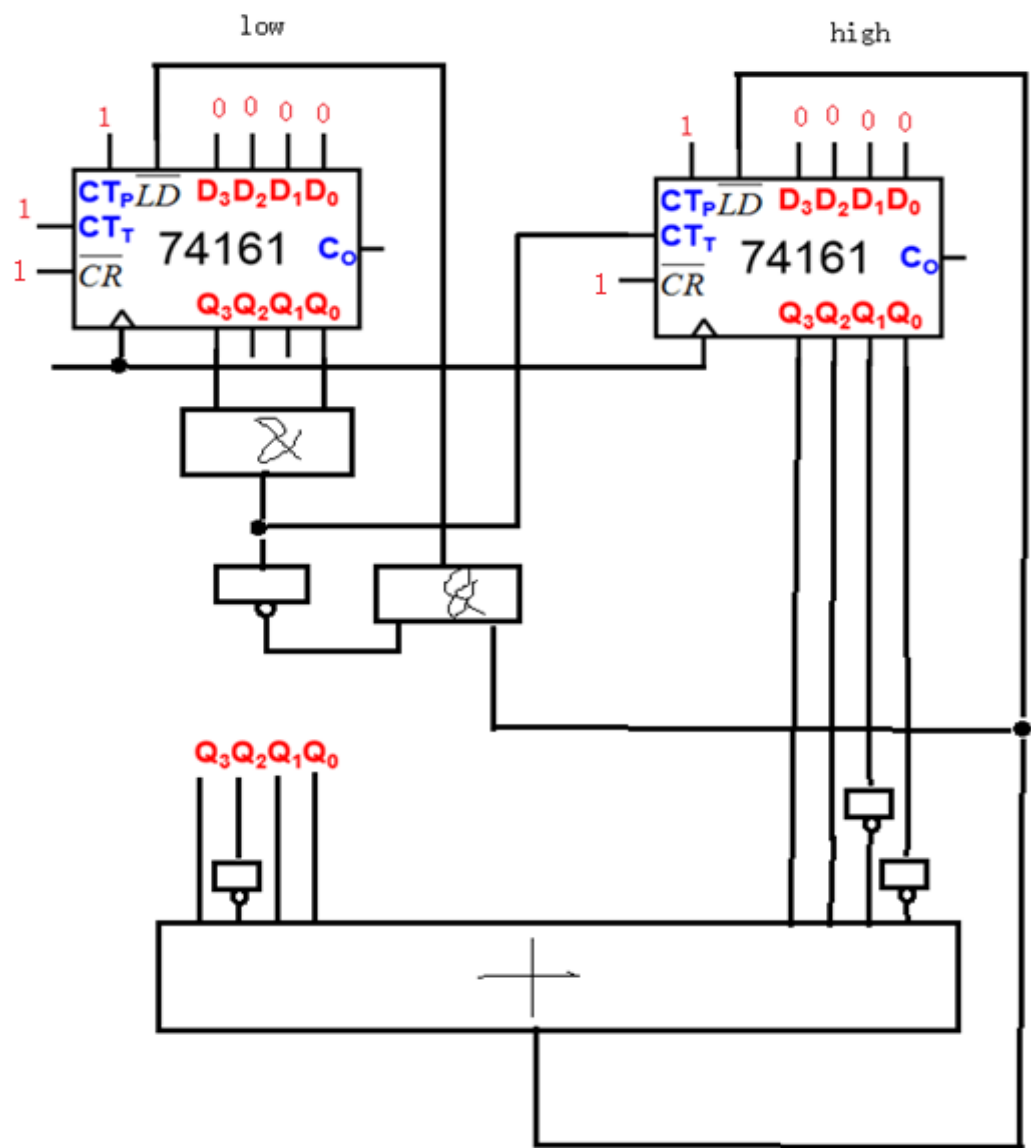


说明：图中没有体现启动脉冲！

上述方案中低位片采用的是 16 进制（4 位 2 进制）计数。

如果低位片采用 10 进制计数，就需要先改造再级联，然后再改造！

先将低位片改造成 10 进制计数器，再和高位片级联，然后再将整个电路改造成 35 进制。



2、用 74LS193 设计一个 35 进制加 1 计数器。

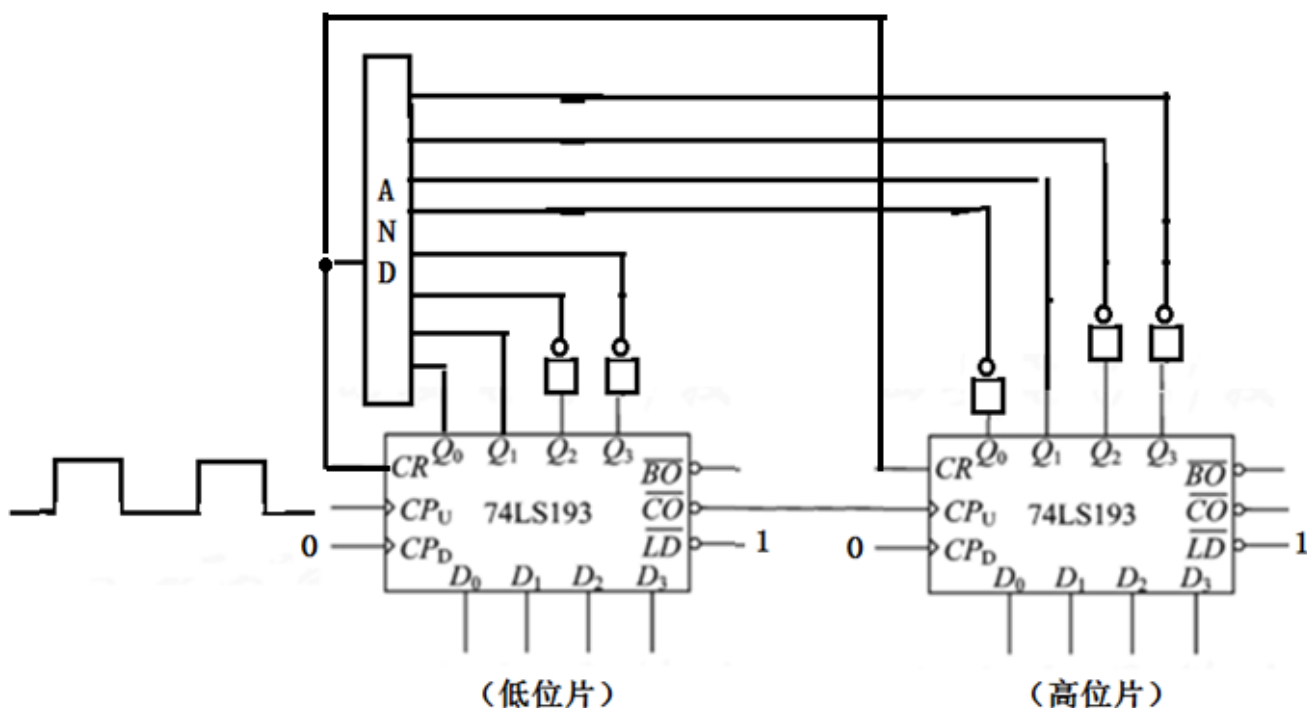
表5.5.3 74LS193的逻辑功能表

CR	\overline{LD}	CP_{\uparrow}	CP_{\downarrow}	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3
1	×	×	×	×	×	×	×	0	0	0	0
0	0	×	×	d_0	d_1	d_2	d_3	d_0	d_1	d_2	d_3
0	1	↑	1	×	×	×	×	加法计数			
0	1	1	↑	×	×	×	×	减法计数			
0	1	1	1	×	×	×	×	保持			

参考解答：可以利用异步清零功能，也可以利用异步置数功能

方案一：先级联，再改造

选用 0000 0000 到 0010 0011 作为计数状态，其中 0010 0011 为暂态，采用异步清 0。



说明：图中没有体现启动脉冲！

方案二：先改造，再级联

将一个片子改造成 5 进制加 1 计数器，选用 0000 到 0101 作为计数状态，其中 0101 作为暂态。

另一个改造成 7 进制加 1 计数器，选用 0000 到 0111 作为计数状态，其中 0111 作为暂态。

重点是接下来的级联！

问题：高位片的计数脉冲能利用低位片的进位输出吗？

回答：不行！低位片的进位输出只在低位片的计数值为 1111 时才会产生。

要点：高位片的计数脉冲可以利用低位片的计数值来生成！

图略！

方案三：先改造，再级联，再改造

先将低位片改造成 10 进制计数器，再级联，再改造！

说明：严格来说，方案 2 和方案 3 属于模 35 计数器，方案 2 的两片分别是 5 进制和 7 进制，方案 3 的两片都是 10 进制。

3、用 74LS194 的右移功能设计一个模 5 计数器。

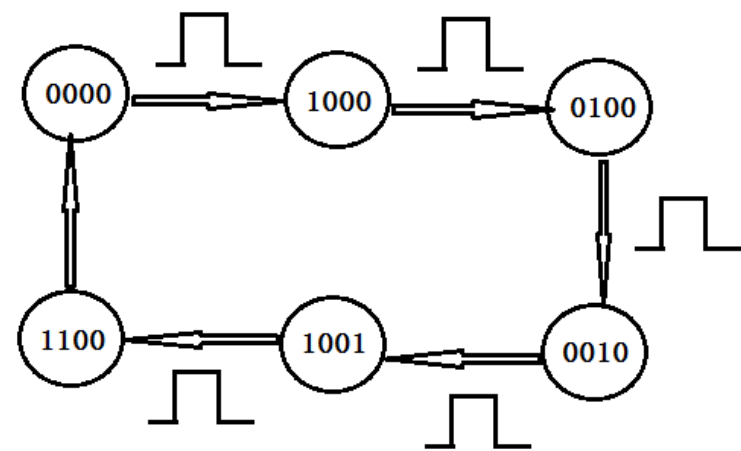
表5.5.10 74LS194的逻辑功能表

输 入				现 态				输 出				功能说明
\overline{CR}	M_1	M_0	CP	D_0	D_1	D_2	D_3	Q_0^{n+1}	Q_1^{n+1}	Q_2^{n+1}	Q_3^{n+1}	
0	×	×	×	×	×	×	×	0	0	0	0	异步清零
1	0	0	×	×	×	×	×	Q_0^n	Q_1^n	Q_2^n	Q_3^n	数据保持
1	0	1	↑	×	×	×	×	D_{SR}	Q_0^n	Q_1^n	Q_2^n	同步右移
1	1	0	↑	×	×	×	×	Q_1^n	Q_2^n	Q_3^n	D_{SL}	同步左移
1	1	1	↑	d_0	d_1	d_2	d_3	d_0	d_1	d_2	d_3	同步置数

参考解答：

方案一：利用异步清零功能

初始状态 0000,从 0000 出发利用右移构造计数循环,需要 6 个不同的状态,第 6 个作为暂态,用来构造清零逻辑。比如：



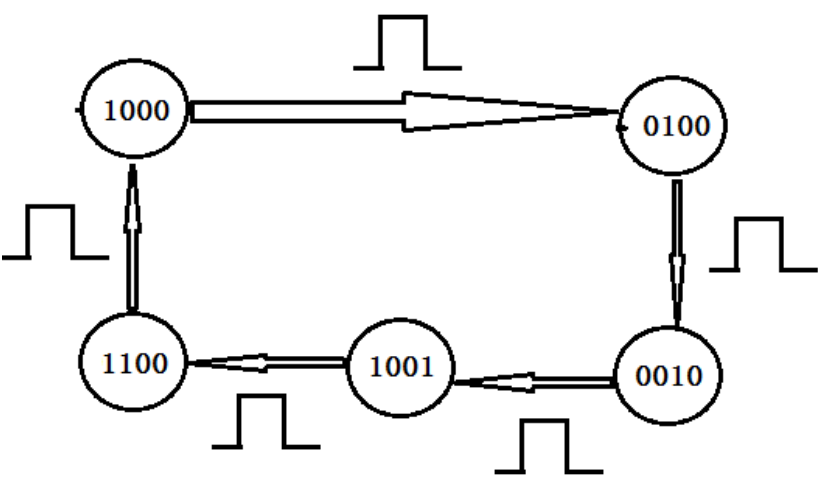
利用现态构造右移输入逻辑：

现态	右移输入
0000	1
1000	0
0100	0
0010	1
1001	1

图略！

方案二：利用同步置数功能

任意选定初始状态，从初始状态出发利用右移构造计数循环，需要 5 个不同的状态，第 5 个用来构造置数逻辑。比如：选择 1000 作为初始状态



利用现态构造右移输入逻辑：

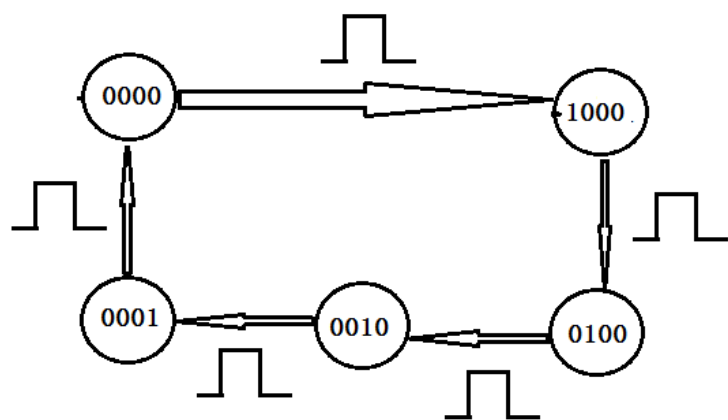
现态	右移输入
1000	0
0100	0
0010	1
1001	1

置数输入端输入 1000。

图略！

方案三：利用异步清零功能启动

初始状态 0000，从 0000 出发利用右移构造计数循环，需要 5 个不同的状态。比如：



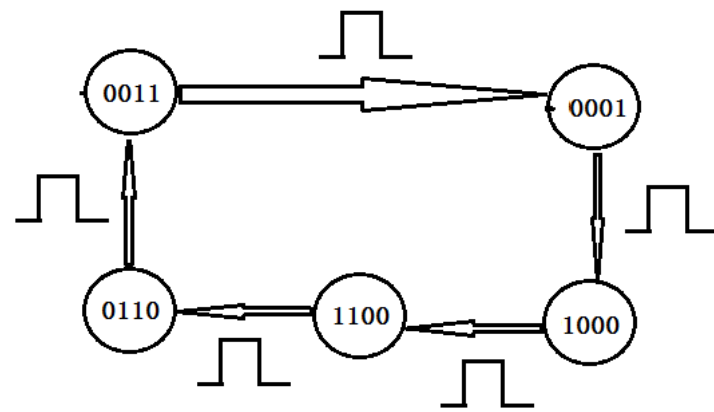
利用现态构造右移输入逻辑：

现态	右移输入
0000	1
1000	0
0100	0
0010	0
0001	0

图略！

方案四：利用同步置数功能启动

任意选定初始状态，从初始状态出发利用右移构造计数循环，需要 5 个不同的状态。比如：



利用现态构造右移输入逻辑：

现态	右移输入
0011	0
0001	1
1000	1
1100	0
0110	0

图略！

大家能体会出不同方案的差别吗？