

数 字 逻 辑

Digital Logic Circuit

丁 贤 庆

ahhfdxq@163.com

Home work (P350)

- 1、本周有实验。地点：1号综合实验楼306房间
- 2、期末考试，第六章有30分左右的考题。
- 3、期末考试时间：11月1日晚6:30-8:30，地点待定。
- 4、本次的作业(不用抄题目)

6.5.13

6.5.15

6.5.19

考试说明

考试题型：

- 单选，填空，卡诺图化简，
- 组合电路分析和设计题
- 时序电路分析题
- CMOS电路分析题
- 74LS161设计N进制计数器。
- 芯片的应用（74LS138, 74LS151,
74LS153, 74LS161等）
- verilog代码编写题
- 等等
- 第六章有30分左右的考题。

第6章 时序逻辑电路

Sequential Logic Circuit

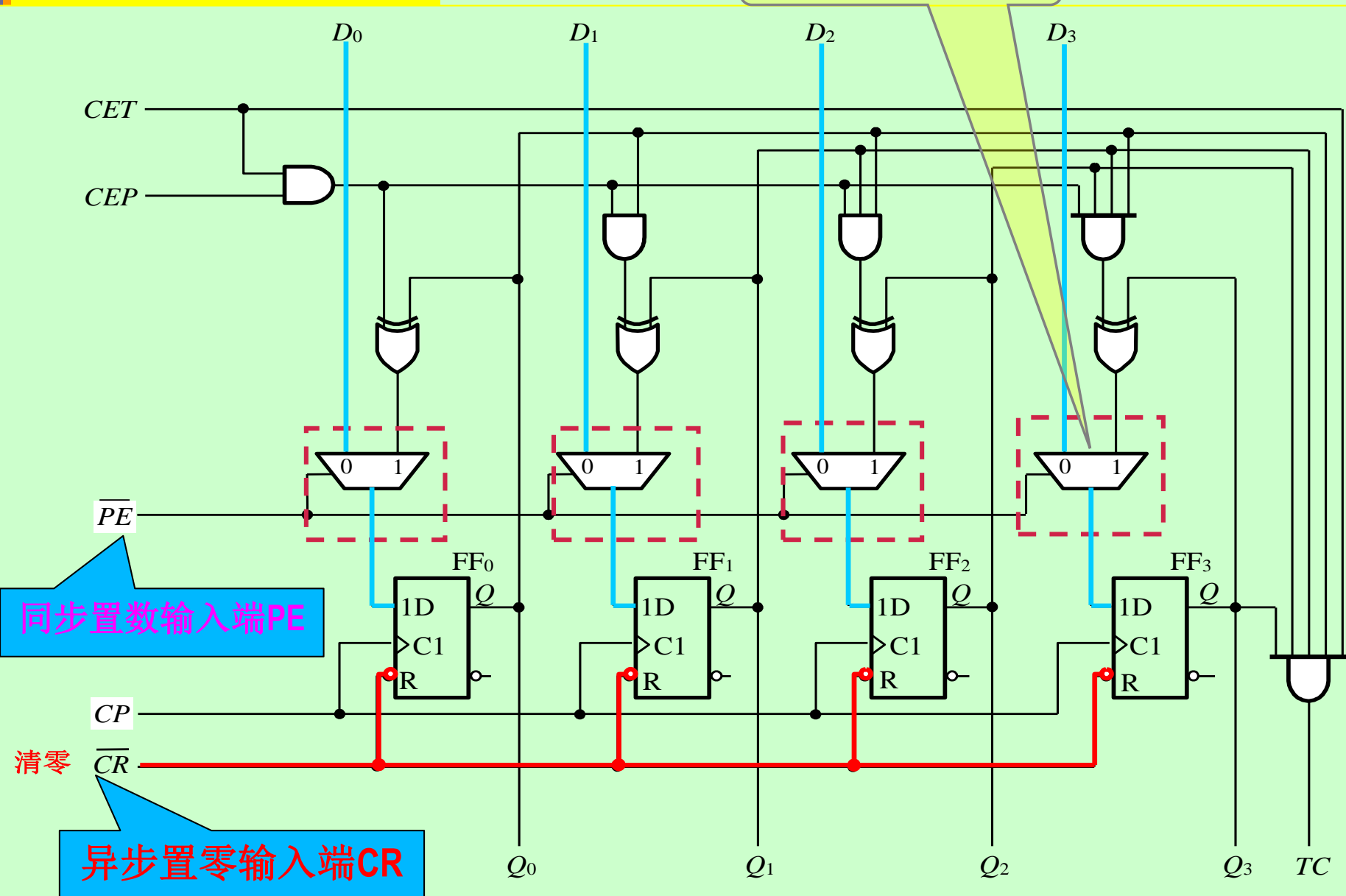
计数器

二进制同步加1计数器
状态转换表:

计数顺序	电路状态				进位输出
	Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1
16	0	0	0	0	0

芯片74LVC161

2选1数据选择器



根据此图可以写出74LVC161的4个工作特性（参见下页的表格）

计数器芯片74LVC161 逻辑功能表

输 入									输 出				
清零	预置	使能		时钟	预置数据输入				计 数				进位
\overline{CR}	\overline{PE}	CEP	CET	CP	D_3	D_2	D_1	D_0	Q_3	Q_2	Q_1	Q_0	TC
L	×	×	×	×	×	×	×	×	L	L	L	L	L
H	L	×	×	↑	D_3	D_2	D_1	D_0	D_3	D_2	D_1	D_0	*
H	H	L	×	×	×	×	×	×	保 持				*
H	H	×	L	×	×	×	×	×	保 持				*
H	H	H	H	↑	×	×	×	×	计 数				*

\overline{CR} 的作用?

清零端

\overline{PE} 的作用?

并行置数端

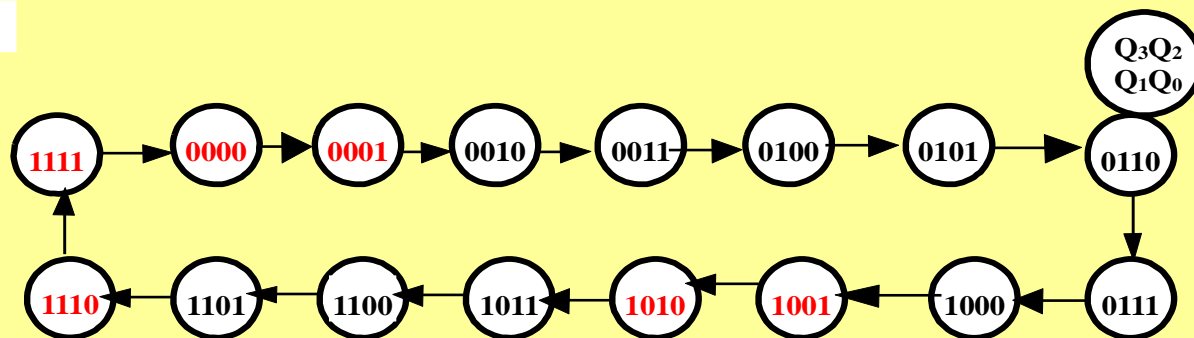
74LVC161的功能表。

输入					功能	
CP	\overline{CR}	\overline{PE}	CEP	CET	说明	解释
×	0	×	×	×	异步复位	$Q_3Q_2Q_1Q_0=0000$
↑	1	0	×	×	同步置数	$Q_3^{n+1}Q_2^{n+1}Q_1^{n+1}Q_0^{n+1}=D_3D_2D_1D_0$
×	1	1	0	1	保持	保持
×	1	1	×	0		保持
↑	1	1	1	1	计数	加1计数

如何才能得到N进制计数器？

1、反馈清零法：利用异步清零输入端**CR**，得到**N**进制计数器

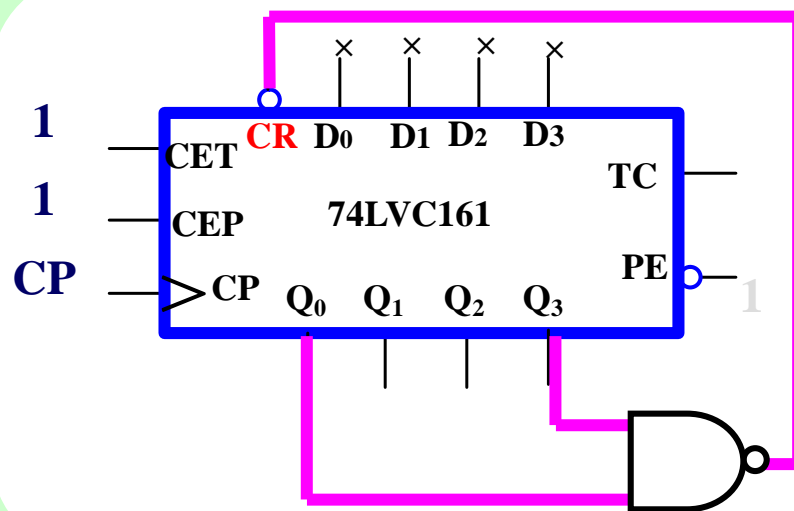
2、反馈置数法：利用同步置数端**PE**，在**M**进制计数器的计数过程中，跳过**M-N**个状态，得到**N**进制计数器



(4) 应用 例 用74LVC161构成九进制加计数器。

(a) 反馈清零法：利用异步清零输入端，在M进制计数器的计数过程中，跳过M-N-1个状态，得到N进制计数器的方法。

CP	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
...			
8	1	0	0	0
9	1	0	0	1
...	...			
15	1	1	1	1



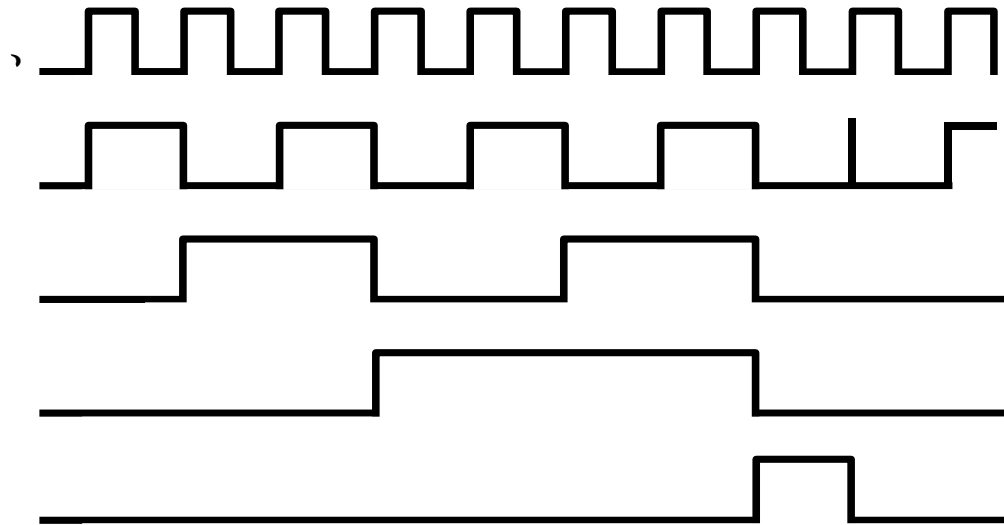
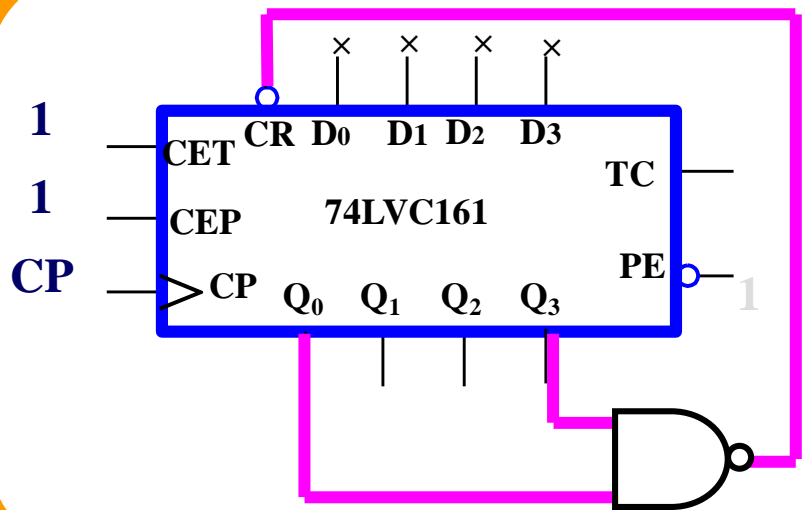
$$CR = \overline{Q_0 \cdot Q_3} = 0$$

采用异步清零法时，由于异步清零最后一个状态保持时间很短，通常忽略不计。

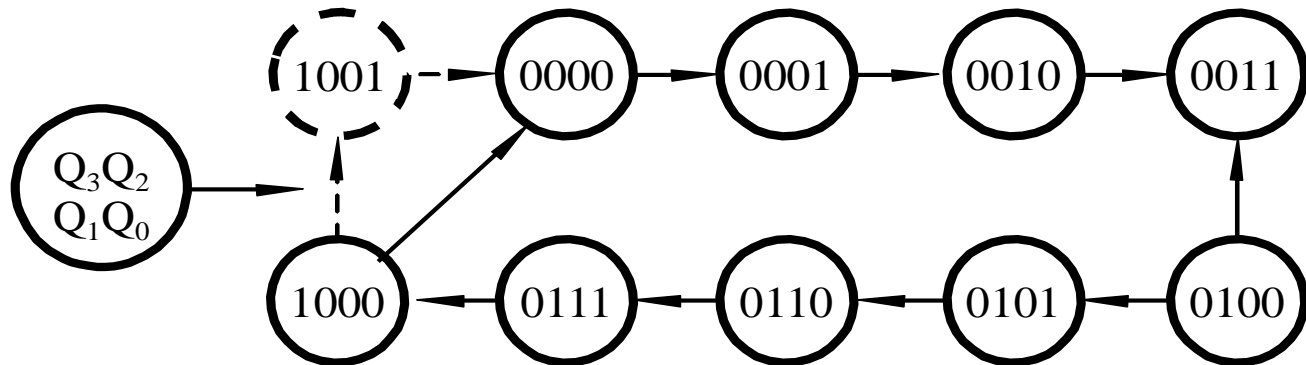
设法跳过16-10=6个状态

计数器的模：计数器状态图中闭合圈包含的稳定的有效状态的数目。本题中最后得到的是模9计数器

工作波形



状态图

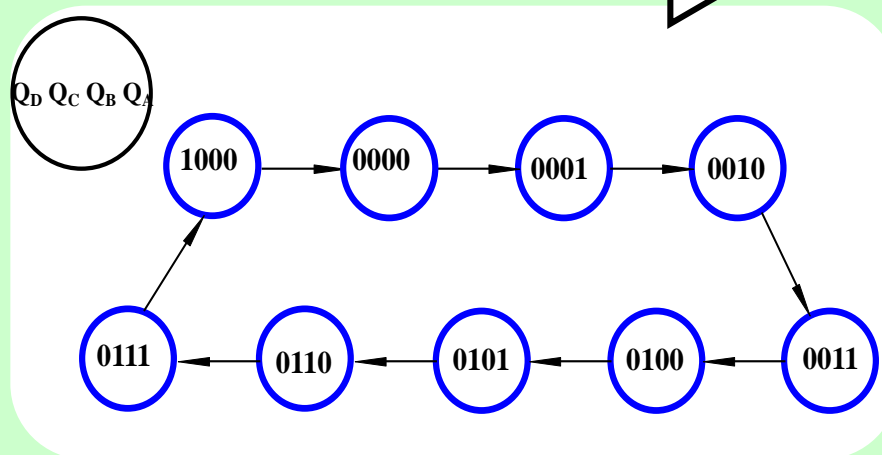
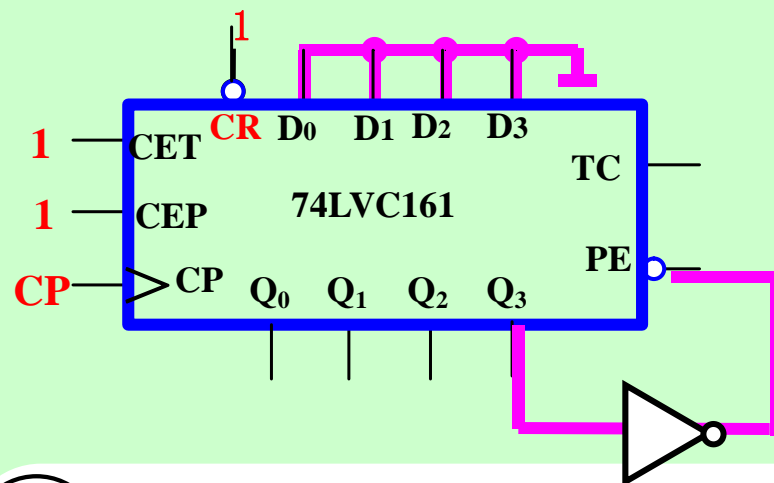


利用同步置数端构成九进制计数器

(b) 反馈置数法:利用同步置数端, 在M进制计数器的计数过程中, 跳过M-N个状态, 得到N进制计数器的方法。

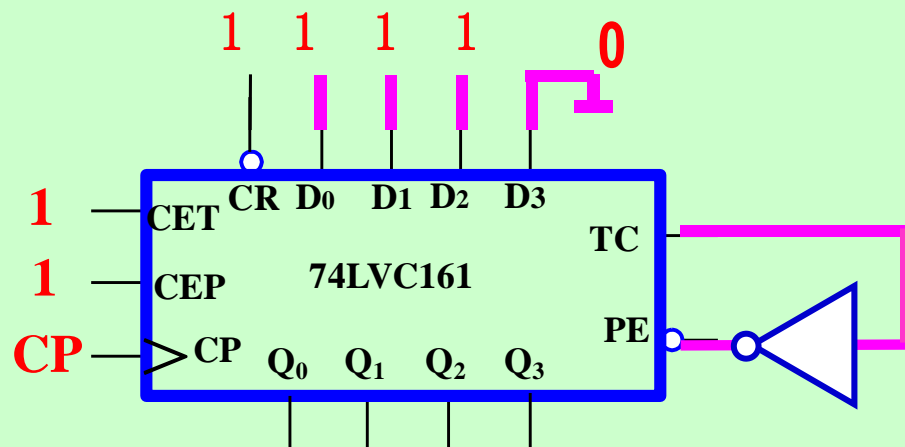
CP	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
...			
8	1	0	0	0
9	1	0	0	1
...				
15	1	1	1	1

$$PE = \overline{Q_3} = 0$$

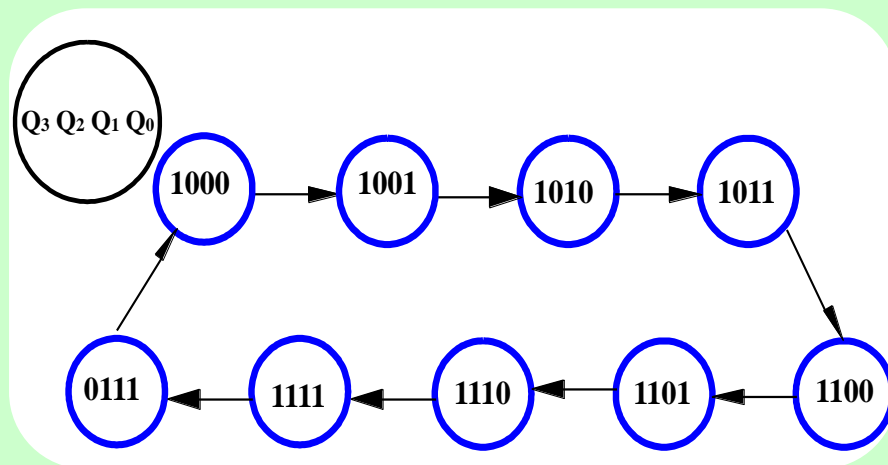


采用后九种状态作为有效状态，用**反馈置数法** 构成九进制加计数器。

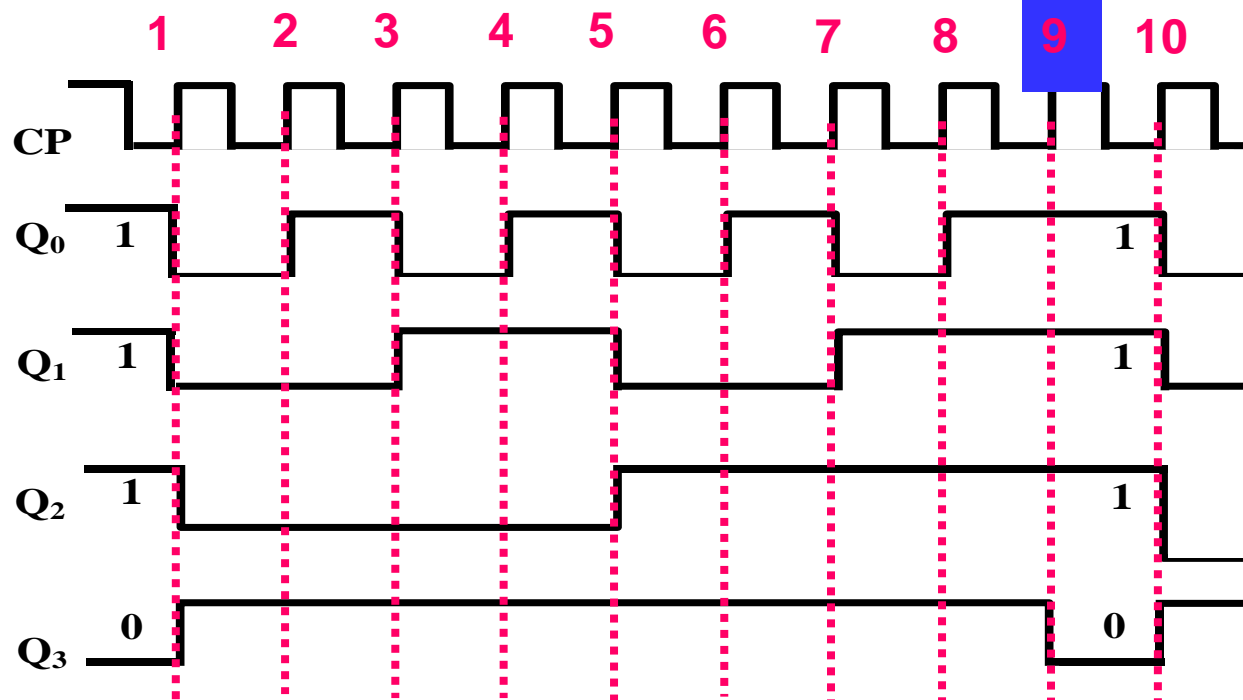
Q_3	Q_2	Q_1	Q_0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1



$$TC = CET \cdot Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0 = 1$$



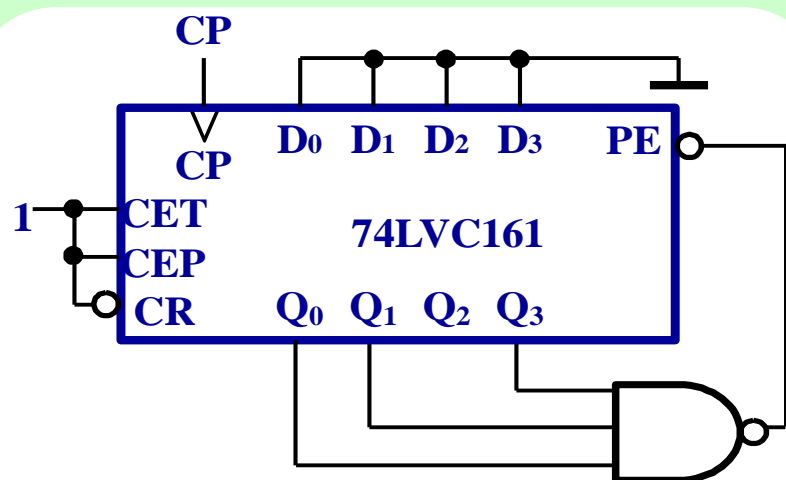
波形图:



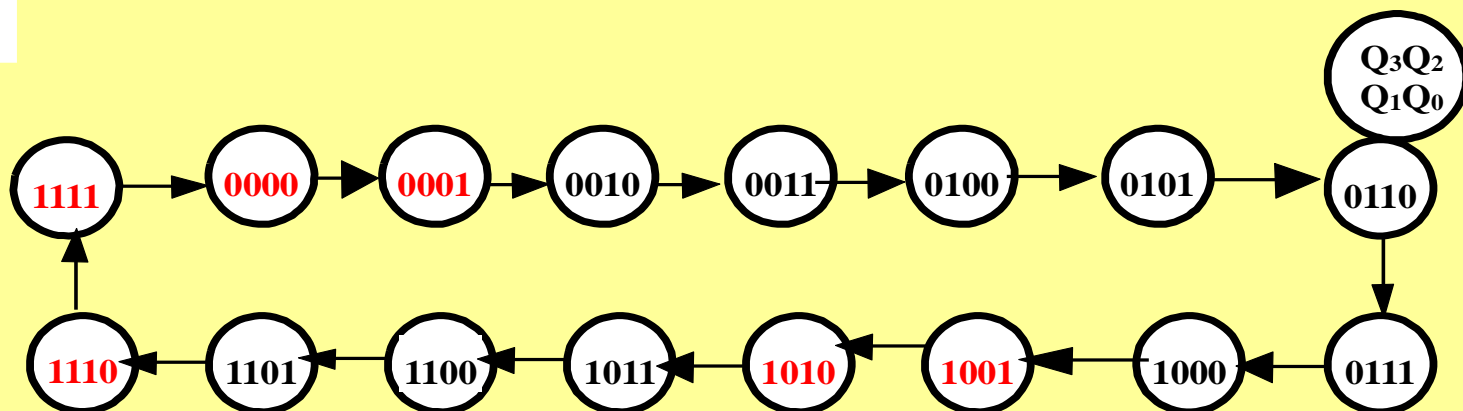
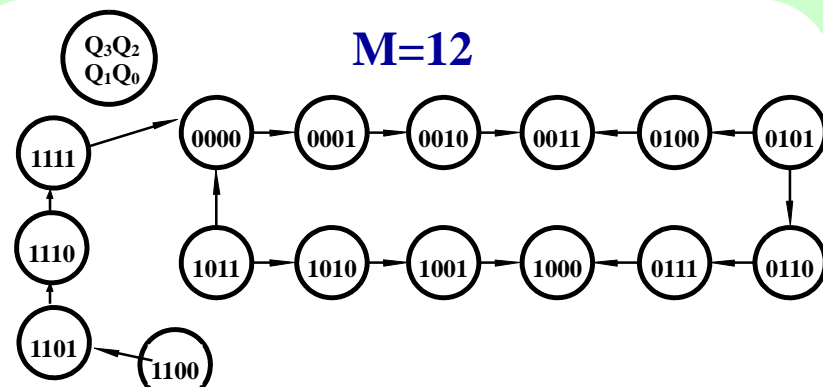
该计数器的模为9。

计数器的模：计数器状态图中闭合圈包含的稳定的有效状态的数目。本题中最后得到的是模9计数器

分析下图所示的时序逻辑电路，试画出其状态图和在CP脉冲作用下 Q_3 、 Q_2 、 Q_1 、 Q_0 的波形，并指出计数器的模是多少？



$$PE = \overline{Q_3 \cdot Q_1 \cdot Q_0} = 0$$

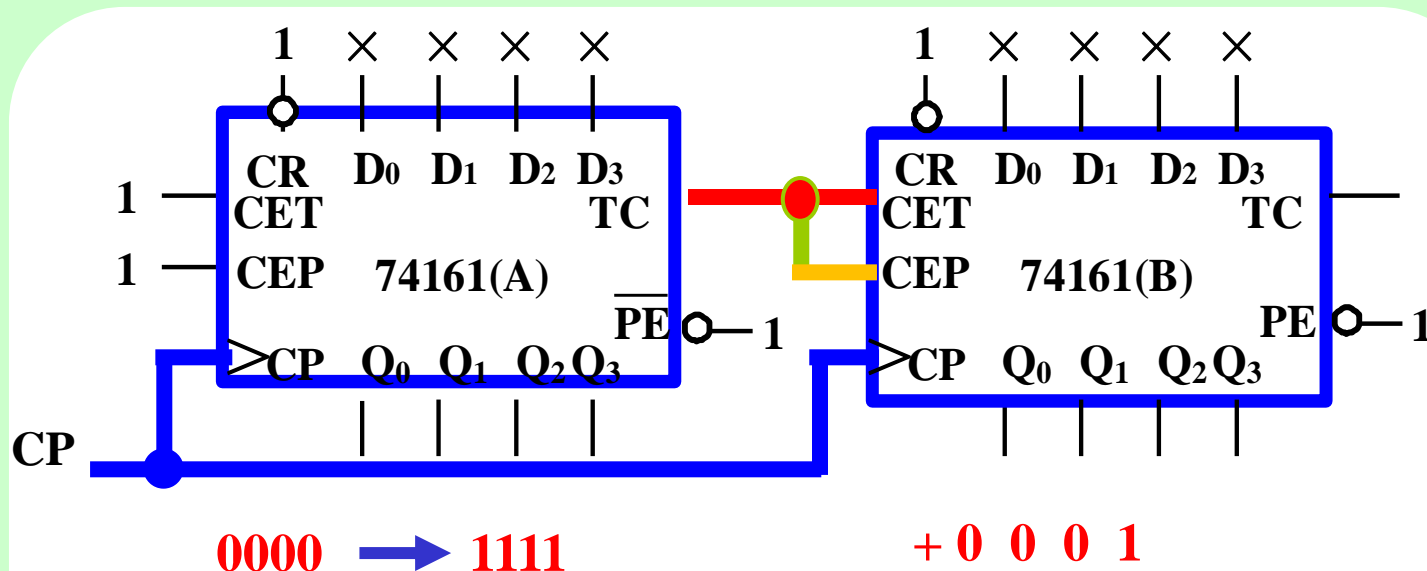


例 用74VC161组成256进制计数器。

解：设计思想

- 1片74161是16进制计数器
- $256 = 16 \times 16$
- 所以256进制计数器需用两片74161构成
- 片与片之间的连接通常有两种方式：
 - 并行进位（低位片的进位信号作为高位片的使能信号）
 - 串行进位（低位片的进位信号作为高位片的时钟脉冲，即异步计数方式）

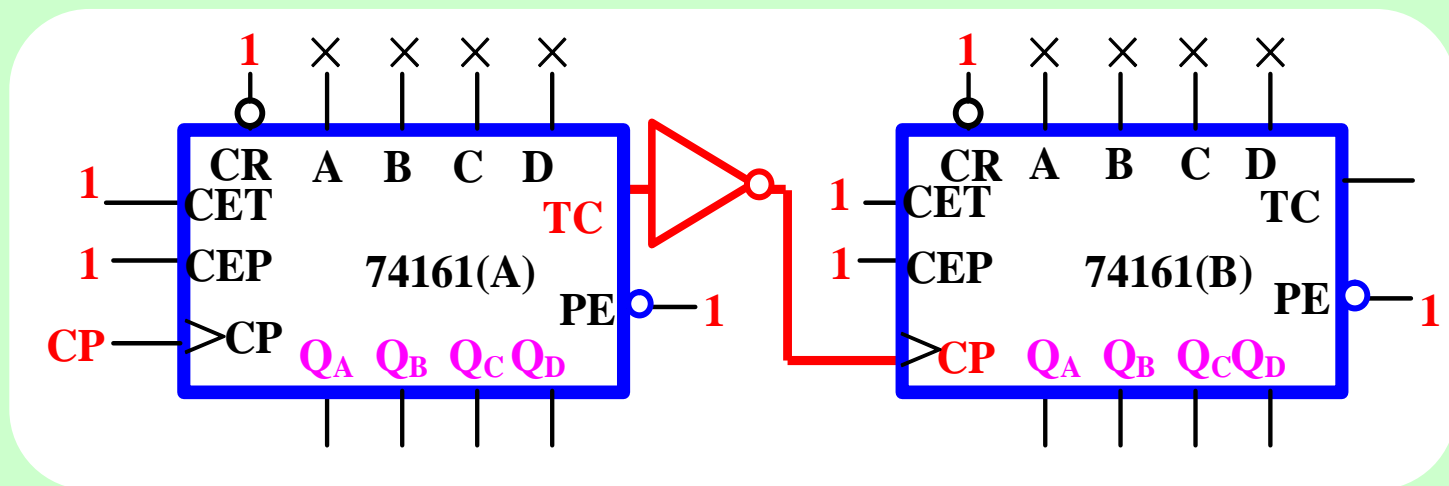
并行进位：低位片的进位作为高位片的使能(采用同步时钟)



计数状态：0000 0000 ~ 1111 1111

$$N = 16 \times 16 = 256$$

串行进位：低位片的进位作为高位片的时钟



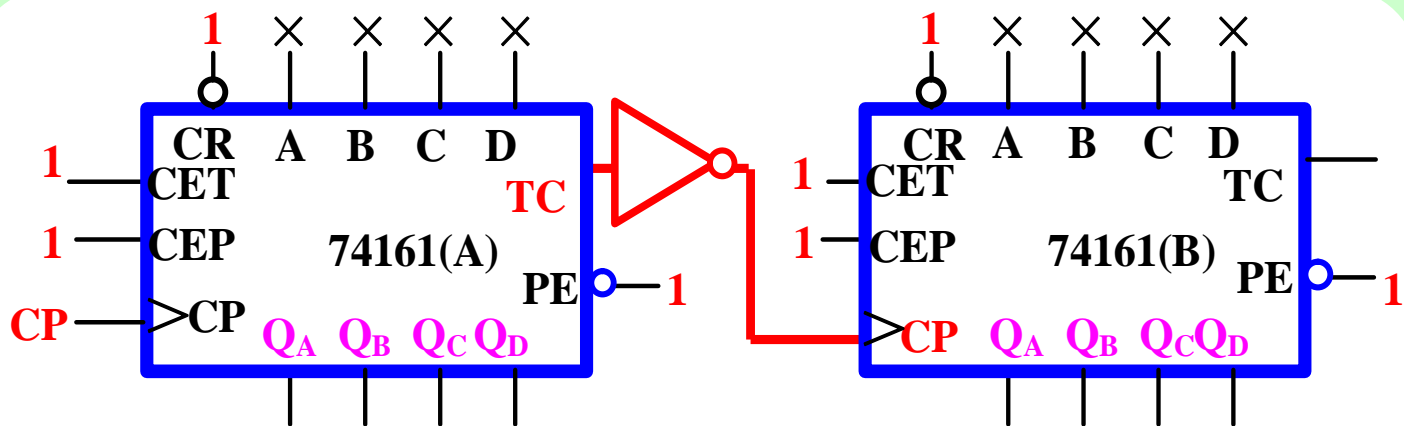
0000 → 1111 + 0 0 0 1

计数状态：0000 0000 ~ 1111 1111

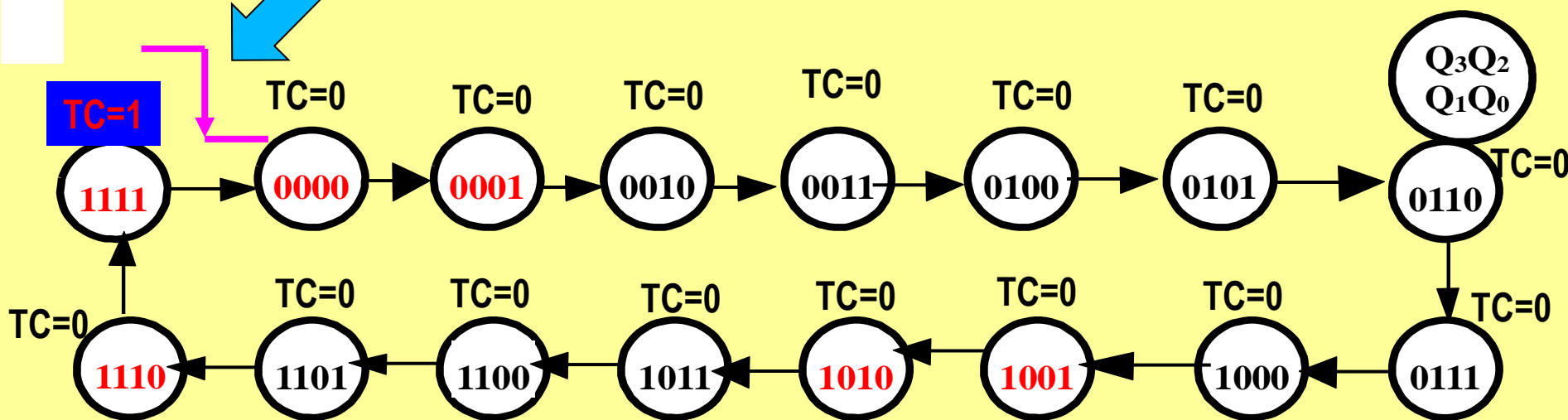
采用串行进位时，为什么低TC要经反相器后作为高位的CP？

加个反相器，是将TC由1变0的下降沿，变成161芯片所需要的由0变1的上升沿。

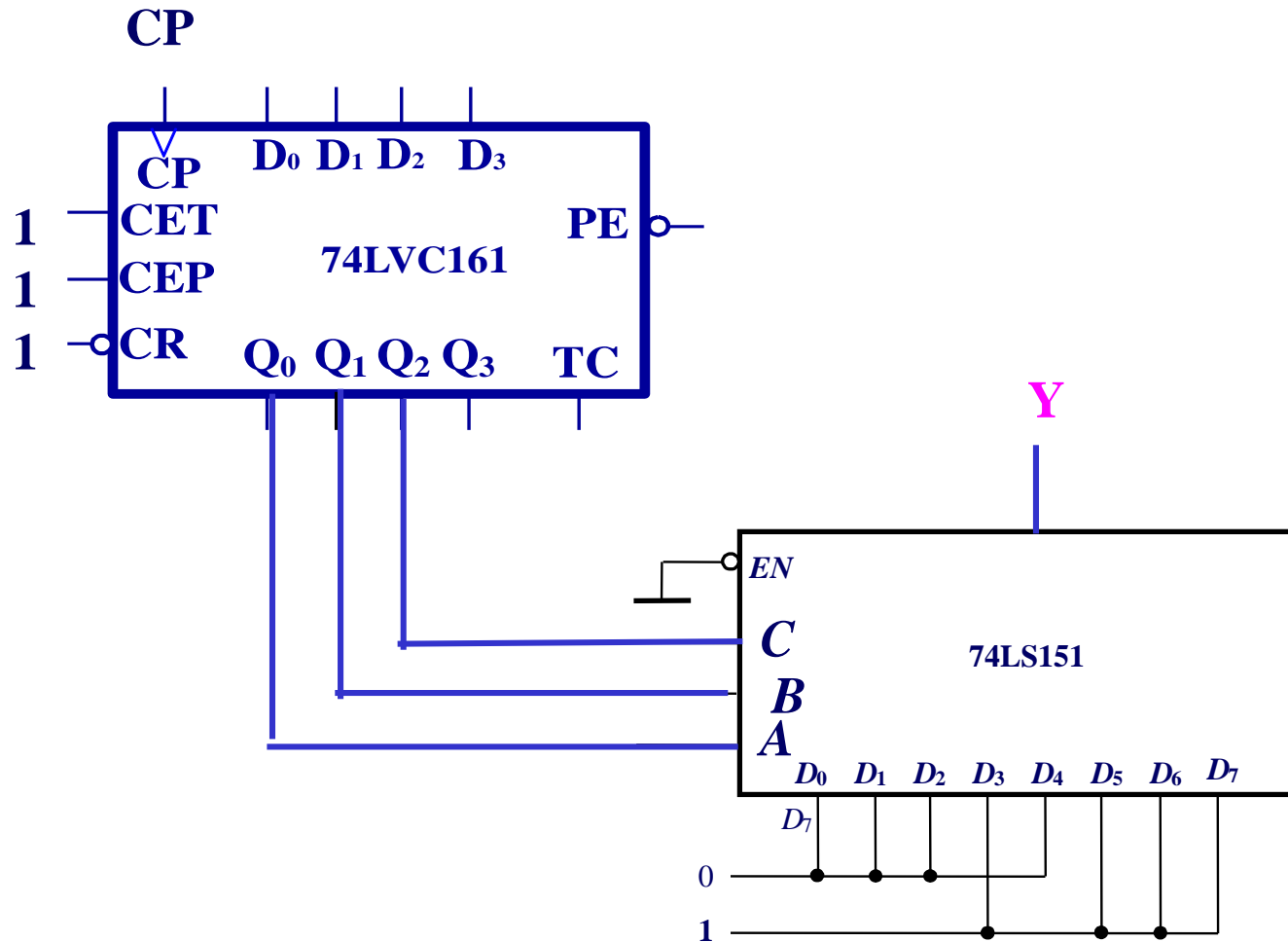
串行进位：低位片的进位作为高位片的时钟



需要将此处的从1到0的下降沿，变为从0到1的上升沿，送给CP时钟输入引脚。



在CP的作用下，Y端产生00010111循环序列信号



如要求Y端产生10110010循环序列信号，如何改变电路的连接？

3. 环形计数器

(1) 工作原理

① 基本环形计数器

Q3连线到D0，就构成基本环形计数器

置初态 $Q_3Q_2Q_1Q_0=0001$,

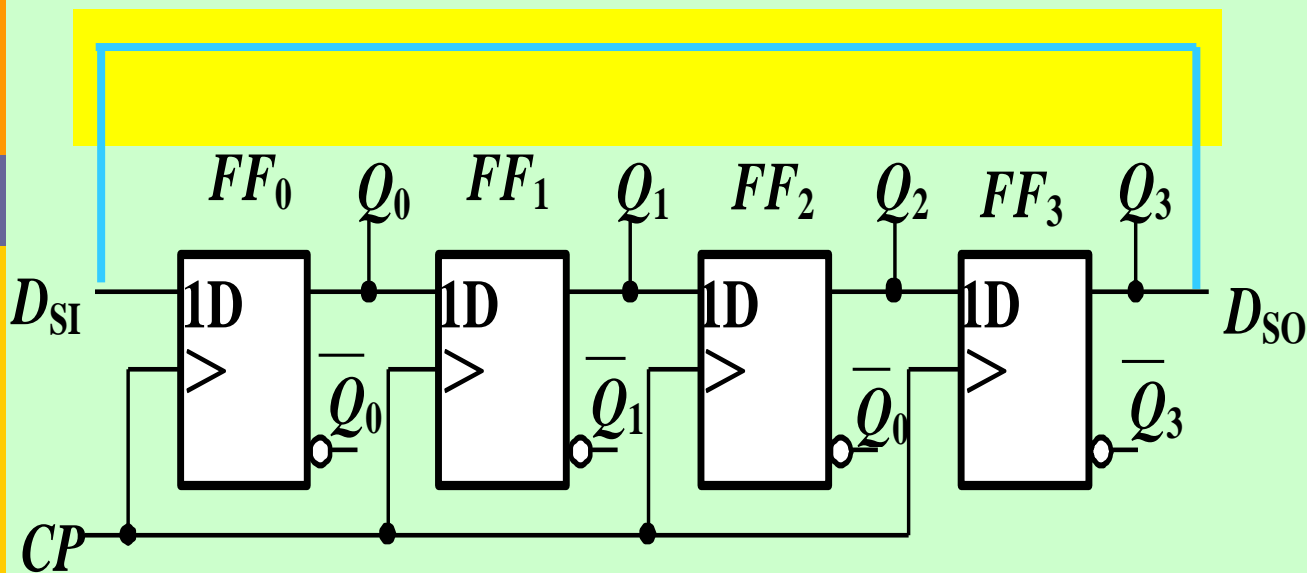
第一个CP: $Q_3Q_2Q_1Q_0=0010$,

第二个CP: $Q_3Q_2Q_1Q_0=0100$,

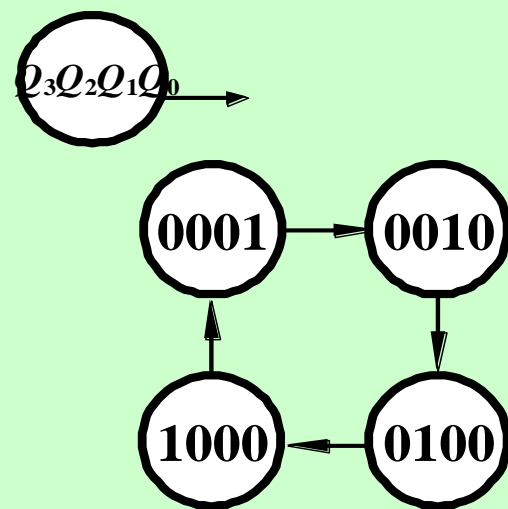
第三个CP: $Q_3Q_2Q_1Q_0=1000$,

第四个CP: $Q_3Q_2Q_1Q_0=0001$,

第五个CP: $Q_3Q_2Q_1Q_0=0010$,



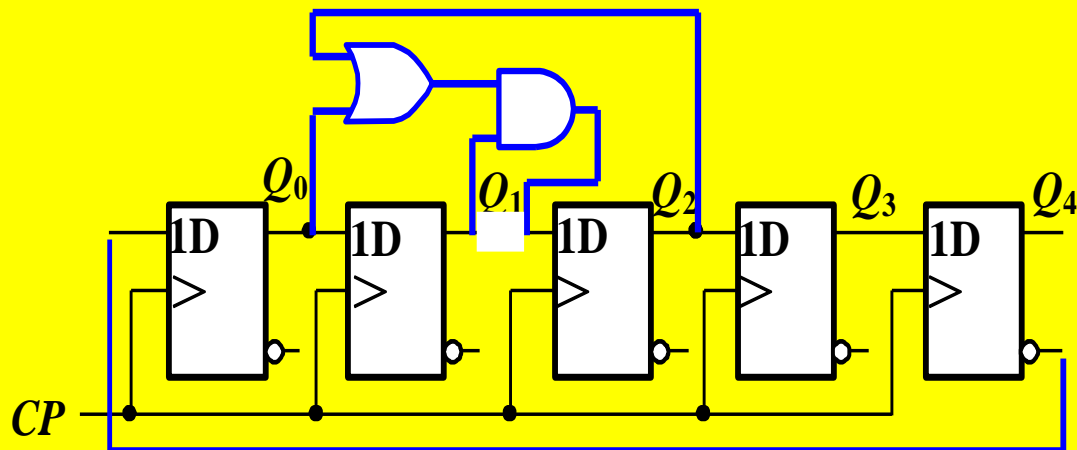
状态图



② 扭环形计数器

a、电路

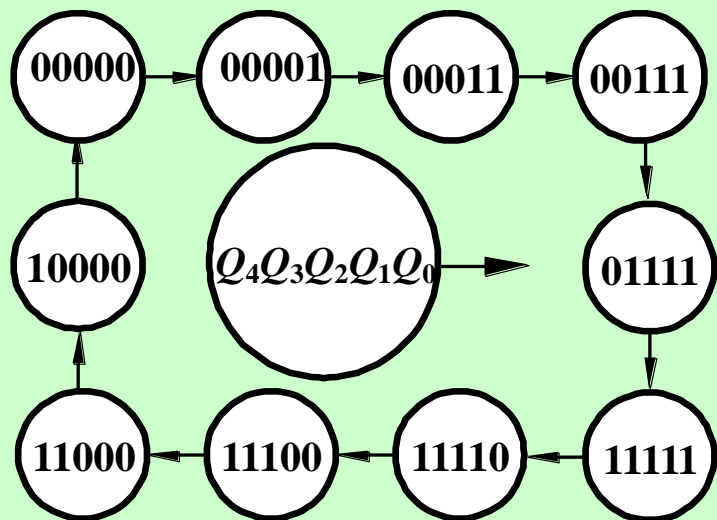
\bar{Q}_4 连线到 D_0 ，就构成扭环计数器



b、状态表

状态编号	Q_4	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	1	1
4	0	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	0
7	1	1	1	0	0
8	1	1	0	0	0
9	1	0	0	0	0

c、状态图



5.6 用Verilog HDL描述锁存器和触发器

敏感事件分为电平敏感事件和边沿触发事件：

电平敏感事件（如锁存器）：

always@(sel or a or b)

sel、a、b中任意一个电平发生变化，后面的过程赋值语句将执行一次。

边沿敏感事件（如触发器）：

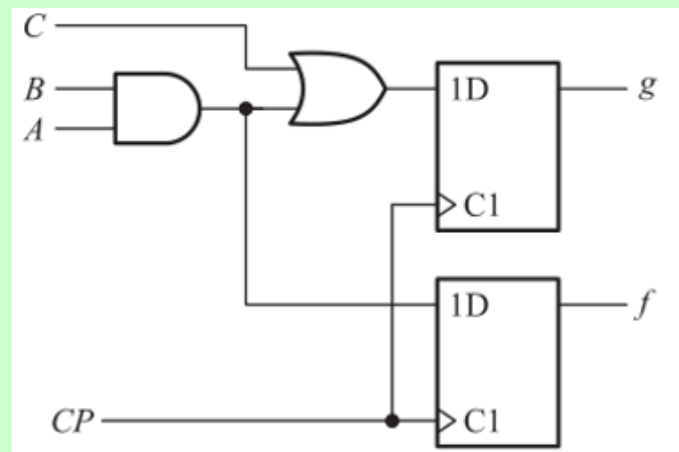
always@(posedge CP or negedge CR)

CP的上升沿或CR的下降沿来到，后面的过程语句就会执行。

过程赋值语句有阻塞型和非阻塞型：

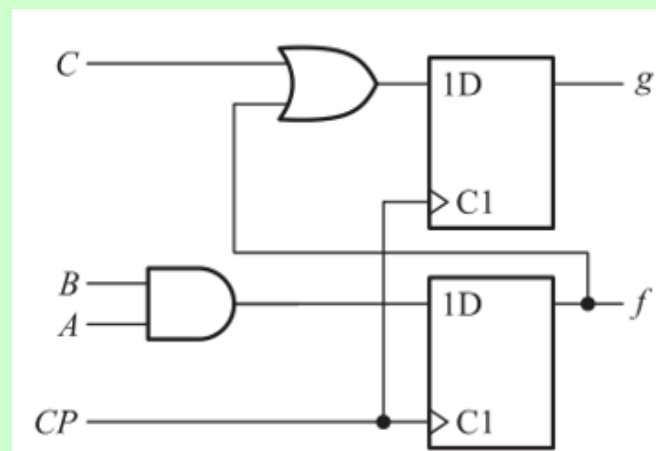
阻塞型用“=”表示，多条语句顺序执行。

```
always@(posedge CP)
begin
    f = A & B;
    g = f | C;
end
```



非阻塞型用“<=”表示，语句块内部的语句并行执行。

```
always@(posedge CP)
begin
    f <= A & B;
    g <= f | C;    // g用CP上升沿前的f值
end
```



5.6.2 锁存器和触发器的Verilog建模实例

module D_latch (Q, D, E); //D锁存器的描述

output Q;

input D, E;

reg Q;

always @(E or D)

if (E) Q <= D; //Same as: if (E== 1)

endmodule

module DFF (

output reg Q,

input D, CP

); //D触发器的描述

always @(posedge CP)

Q <= D;

endmodule

例：分析下面Verilog模块, 说明其逻辑功能。

```
module async_set_rst_DFF (  
    output reg Q, QN,  
    input D, CP, Sd, Rd  
);
```

```
always @(posedge CP, negedge Sd, negedge Rd)
```

```
begin
```

```
    if (~Sd || ~Rd)
```

```
        if (~Sd ) begin Q <= 1'b1; QN <= 1'b0; end//异步置数  
        else      begin Q <= 1'b0; QN <= 1'b1; end//异步清零
```

```
    else
```

```
        begin Q <= D; QN <= ~D; end
```

```
end
```

```
endmodule
```

6.7 用Verilog HDL描述时序逻辑电路

6.7.1 移位寄存器的Verilog建模

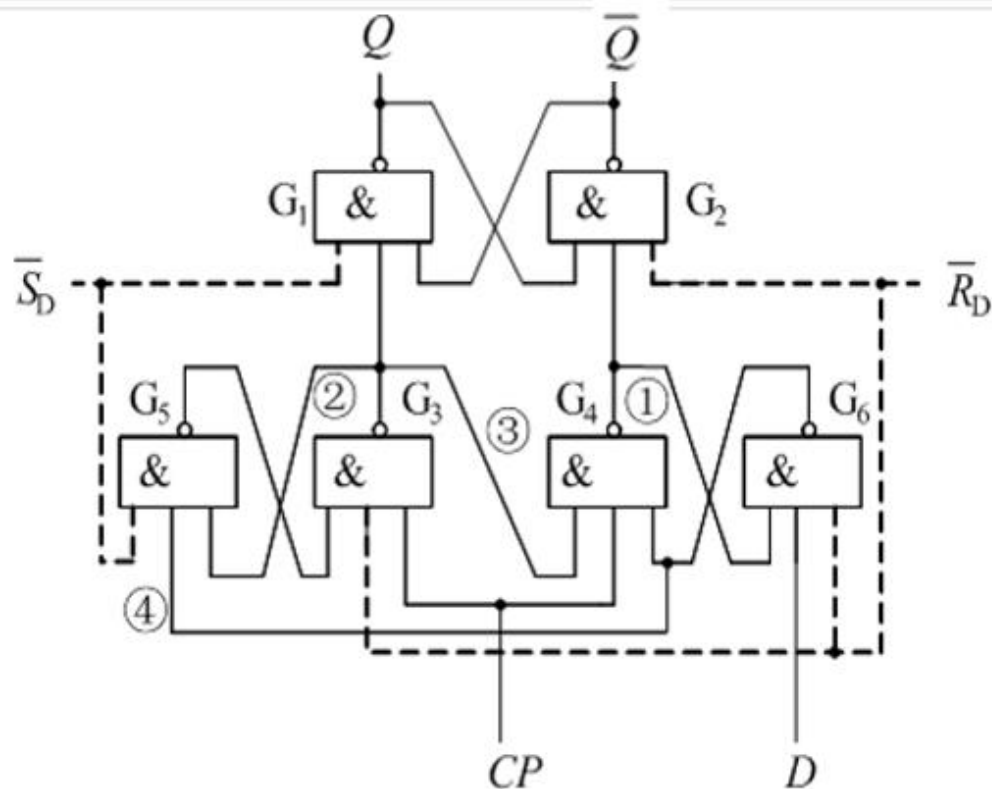
6.7.2 计数器的Verilog建模

6.7.3 状态图的Verilog建模

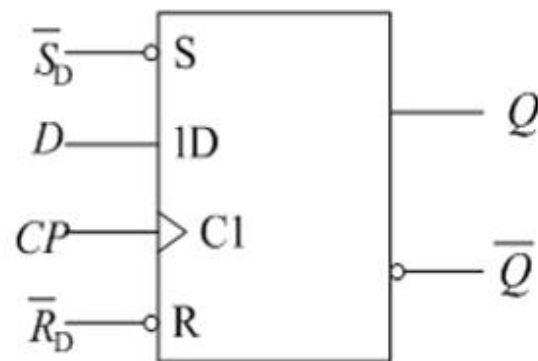
6.7.4 数字钟的Verilog建模

清零端和置1端引脚

维持-阻塞式D触发器



(a) 电路结构图

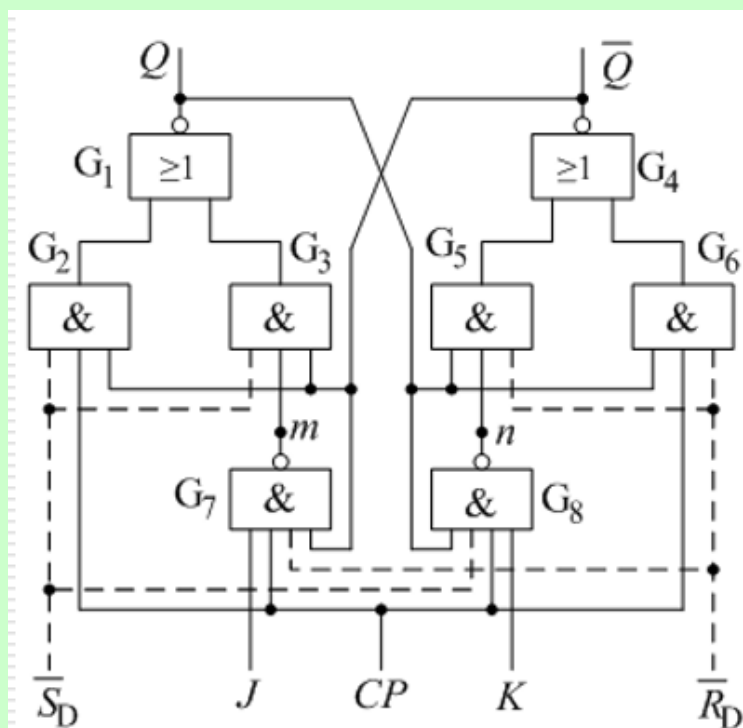


(b) 图形符号

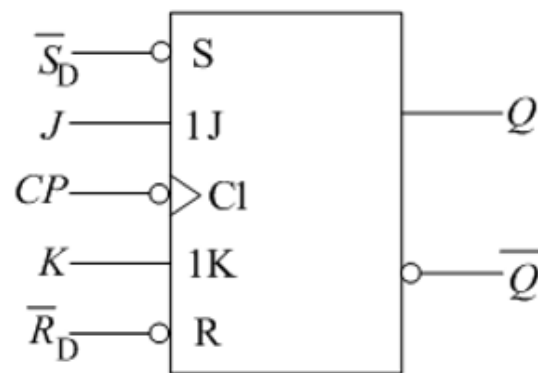
维持-阻塞式 D 触发器

清零端和置1端引脚

下降沿触发JK触发器



(a) 电路结构



(b) 图形符号

图 边沿 JK 触发器

6.7.1 移位寄存器的Verilog建模

用行为级描述always描述一个4位双向移位寄存器，有异步清零、同步置数、左移、右移和保持。功能同74HC194。

```
module shift74x194 (S1, S0, D, Dsl, Dsr, Q, CP, CR);  
    input S1, S0;           //控制输入  
    input Dsl, Dsr;         //串行输入  
    input CP, CR;           //时钟及清零  
    input [3:0] D;          //并行输入  
    output [3:0] Q;         //寄存器输出  
    reg [3:0] Q;
```

6.7.1 移位寄存器的Verilog建模

always @ (posedge CP or negedge CR)

if (~CR) Q <= 4'b0000;

else

case ({S1,S0})

2'b00: Q <= Q; //保持

2'b01: Q <= {Q[2:0],Dsr}; //右移

2'b10: Q <= {Dsl,Q[3:1]}; //左移

2'b11: Q <= D; //并行输入

endcase

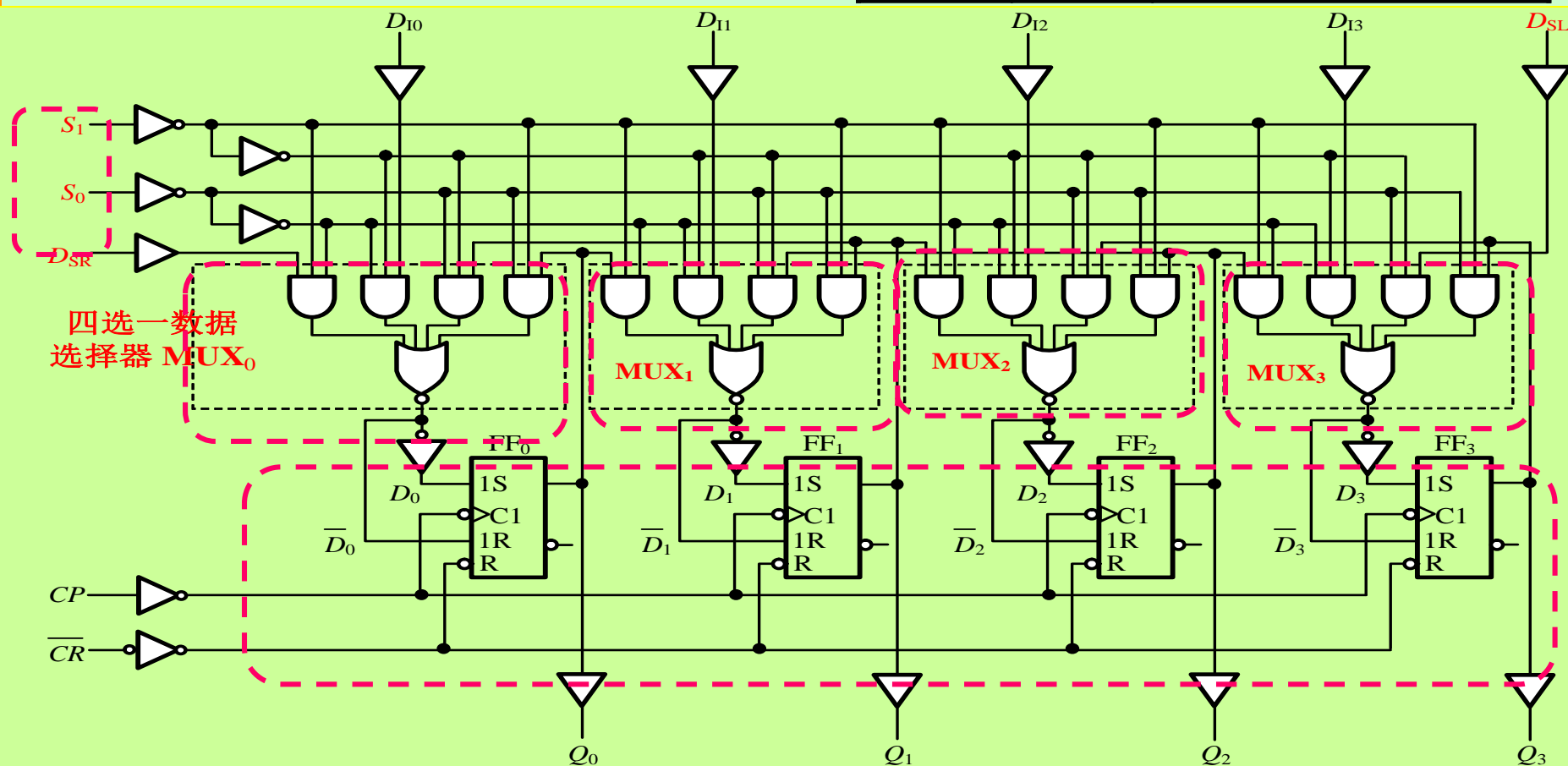
endmodule

(b) 典型集成电路

CMOS 4位双向移位寄存器74HC/HCT194

74HCT194 的功能表

S1	S0	功能
0	0	保持
0	1	低位往高位移动
1	0	高位往低位移动
1	1	并行置入



6.7.2 计数器的Verilog建模实例

用Verilog描述具有使能端、异步置零、同步置数、计数、保持的16进制计数器,类似74HC161芯片。

```
module counter74x161_beh ( //Verilog 2001, 2005 syntax
    input CEP, CET, PE, CP, CR, //输入端口声明
    input [3:0] D,                //并行数据输入
    output TC,                    //进位输出
    output reg [3:0] Q            //数据输出端口及变量的数据类型声明
);
    wire CE;                      //中间变量声明
```

6.7.2 计数器的Verilog建模实例

assign CE=CEP&CET; //CE=1时，计数器计数

assign TC=CET&PE&(Q == 4'b1111); //产生进位输出信号

always @(posedge CP, negedge CR) //Verilog 2001, 2005 syntax

if (~CR) Q<=4'b0000; //实现异步清零功能

else if (~PE) Q<=D; //PE=0, 同步装入输入数据

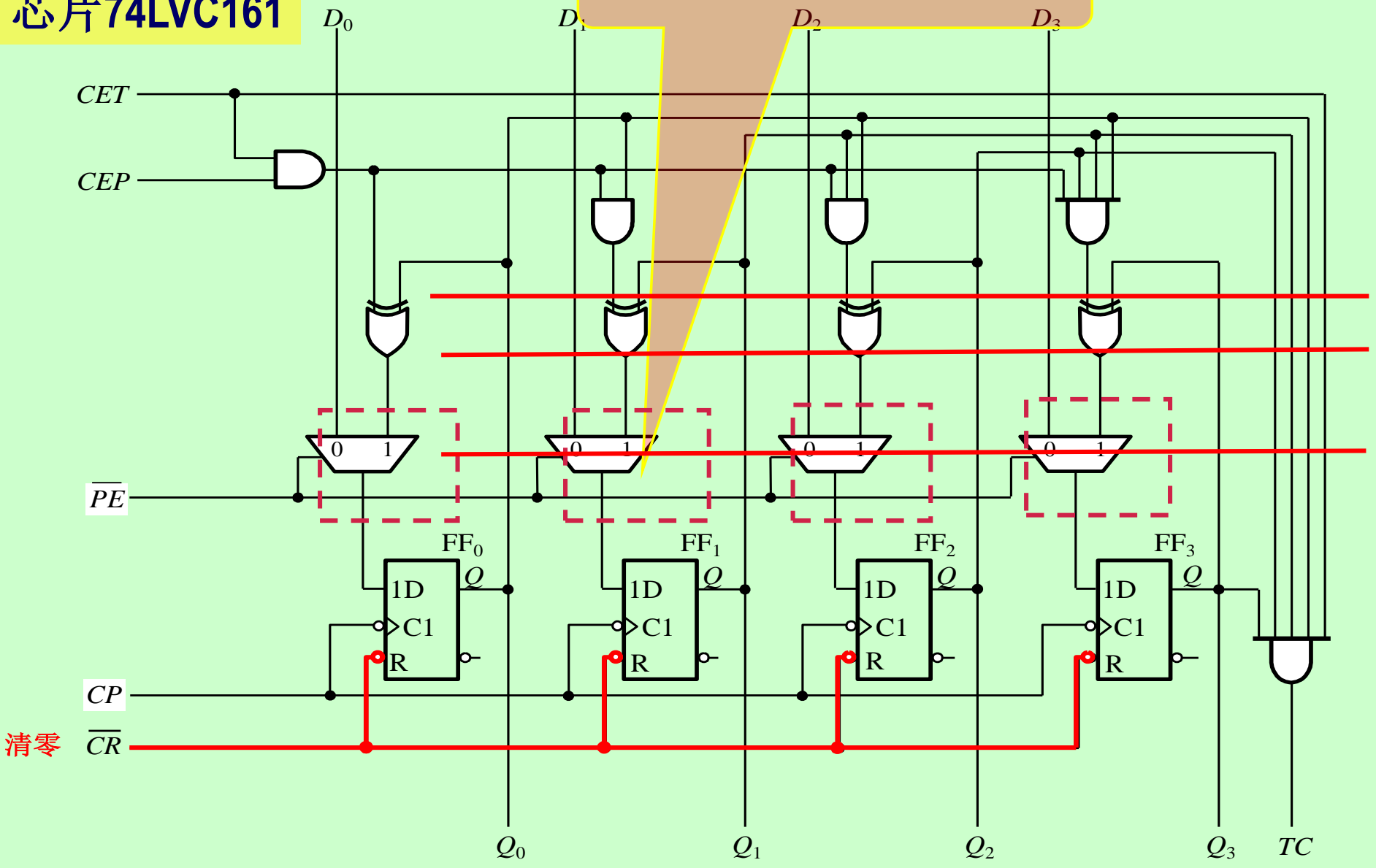
else if (CE) Q<=Q+1'b1; //加1计数

else Q<=Q; //输出保持不变

endmodule

芯片74LVC161

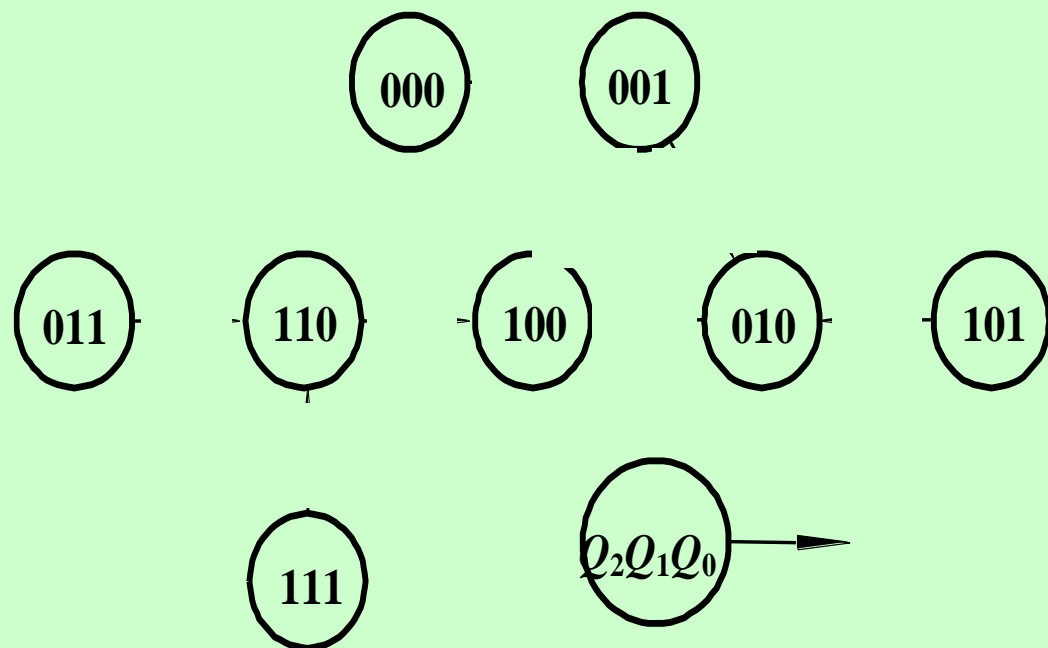
2选1数据选择器



状态表

$Q_2^n Q_1^{n1} Q_0^n$	$Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$
0 0 0	0 0 1
0 0 1	0 1 0
0 1 0	1 0 0
0 1 1	1 1 0
1 0 0	0 0 1
1 0 1	0 1 0
1 1 0	1 0 0
1 1 1	1 1 0

电路的自启动能力



电路具有自启动能力：就是从任何一个状态出发，经过若干个脉冲后，都能进入有效循环圈里去。就具有自启动能力。

有效循环圈：如果该电路送入了1000个脉冲，你会发现至少有998个脉冲，电路状态是在闭合循环圈里运行，这个闭合循环圈就是有效循环圈。

实验中注意事项:

$$0 \oplus A = A$$

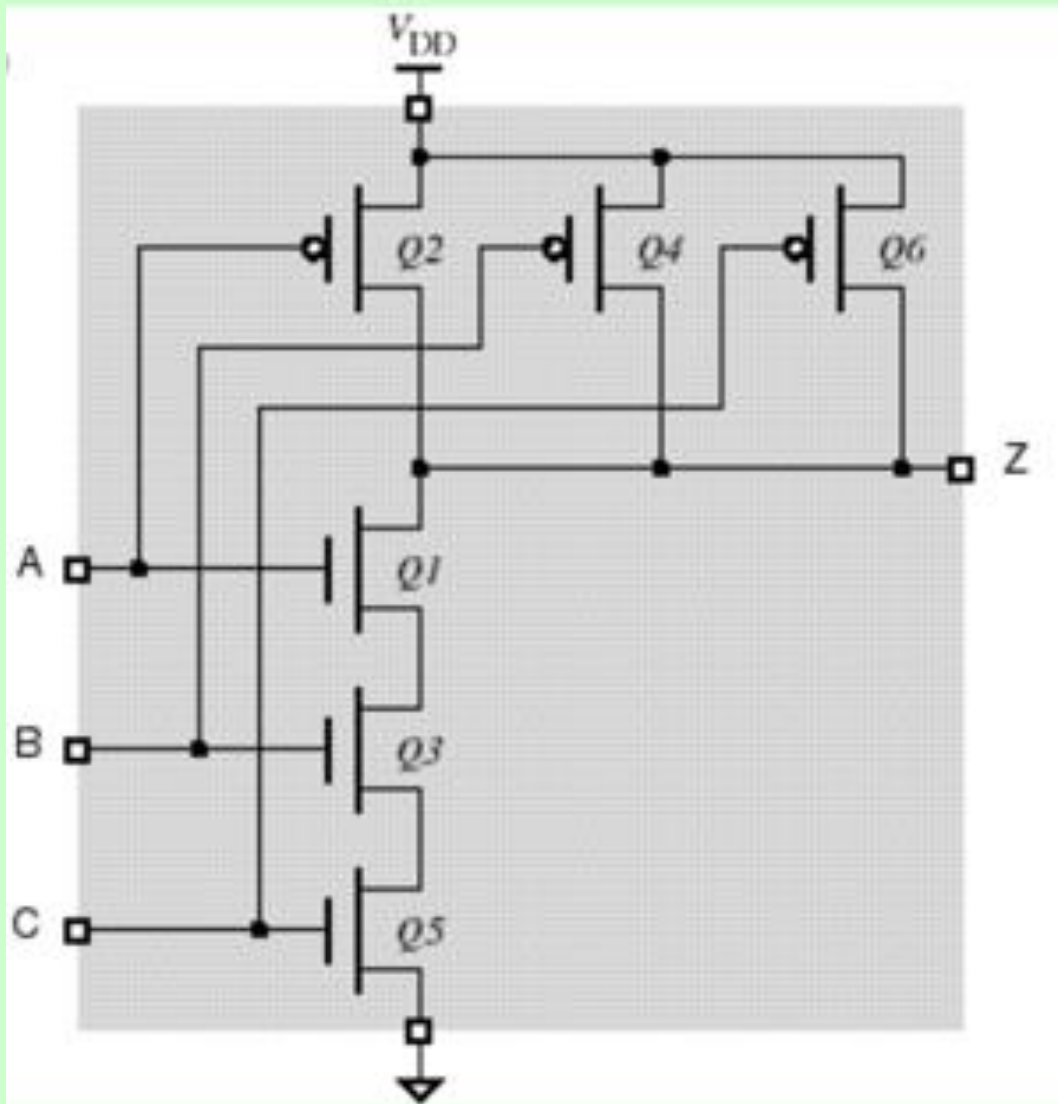
$$1 \oplus A = \bar{A}$$

$$A \oplus A = 0$$

$$A \odot A = 1$$

$$A \oplus B = \bar{A} \odot B = A \odot \bar{B} = \overline{A \odot B}$$

实验中注意事项:



$$Z = \overline{A \cdot B}$$

用左图来实现上式时，C信号可以连接：

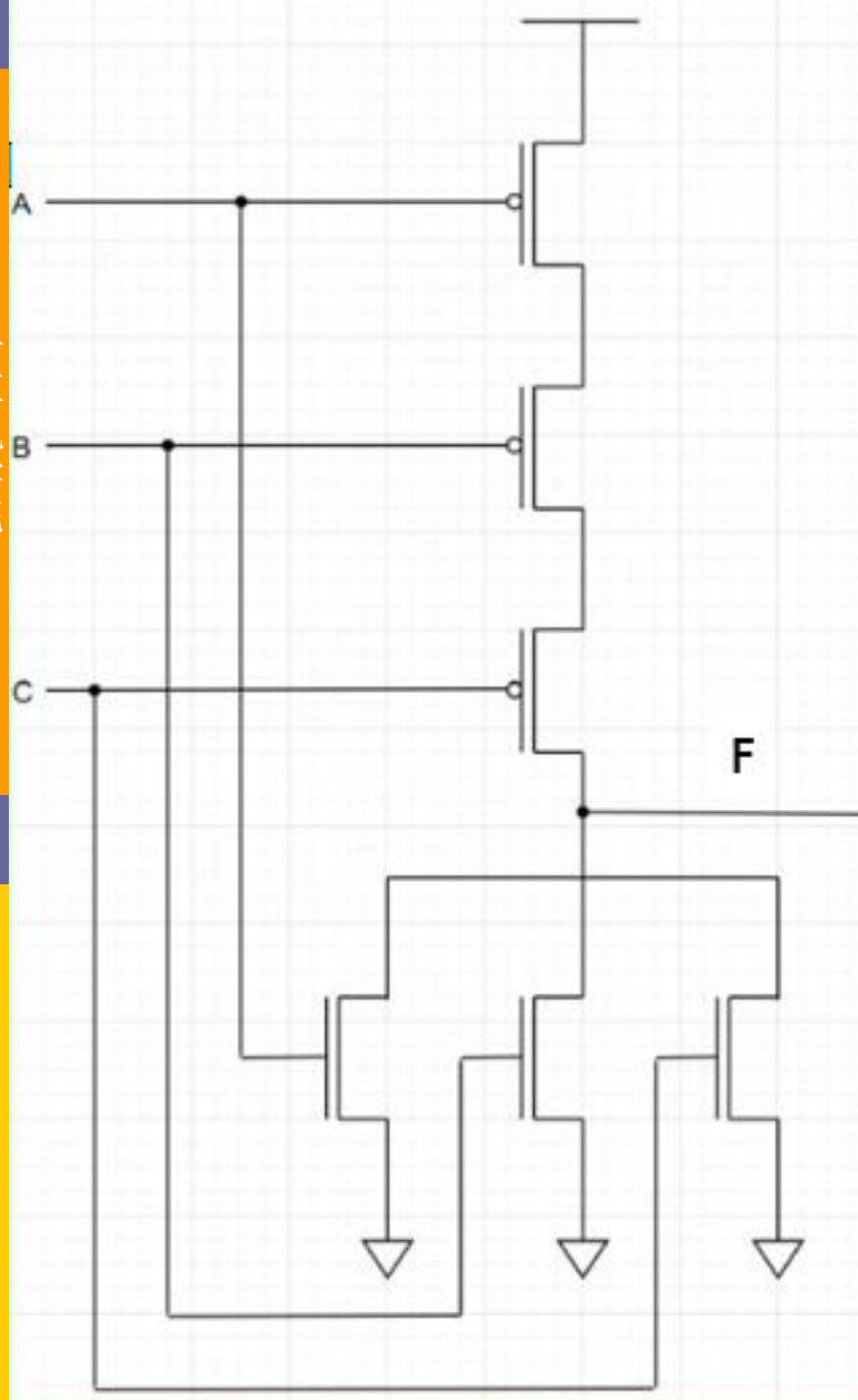
- (1) C信号没有用到，直接接地。✗
- (2) C信号没有用到，直接接 V_{DD} 。✓
- (3) C信号没有用到，不接，✗悬空。
- (4) C信号没有用到，直接接A。✓
- (5) C信号没有用到，直接接B。✓

实验中注意事项:

$$F = \overline{A+B}$$

用左图来实现上式时，C信号可以连接：

- (1) C信号没有用到，直接接地。✓
- (2) C信号没有用到，直接接 V_{DD} 。✗
- (3) C信号没有用到，不接，悬空。✗
- (4) C信号没有用到，直接接A。✓
- (5) C信号没有用到，直接接B。✓

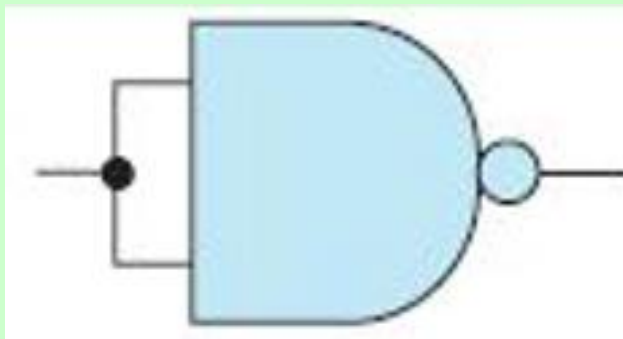
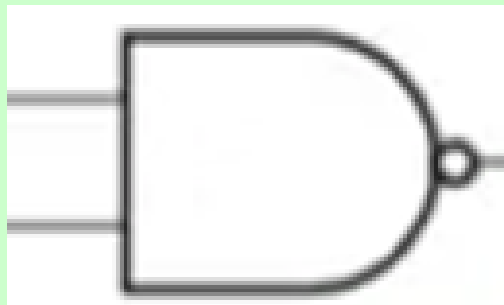


实验中注意事项:

只有2输入的与非门若干个，如何实现3输入的与非运算？

$$F = A \cdot B \cdot C$$

$$F = A \cdot B \cdot C$$



完