

第七章 典型同步时序电路的设计与应用

7.1 计数器

7.2 寄存器

7.3 移位寄存器



7.4 移位寄存器型计数器

7.5 节拍分配器

典型同步时序电路的设计实现

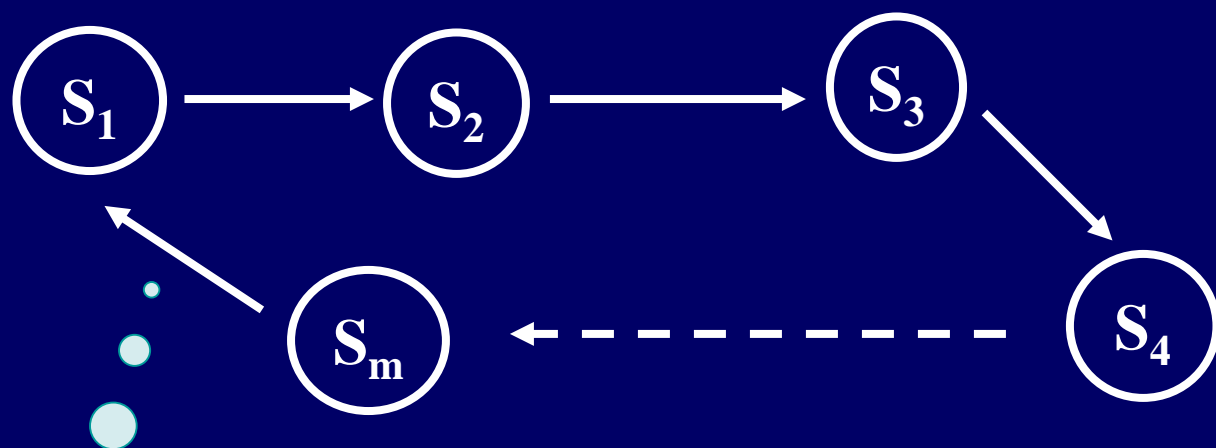
- (1) 基于触发器的设计
- (2) 基于中规模时序器件的设计
- (3) 基于硬件描述语言的建模

基于触发器的典型同步时序电路设计步骤

- 1) 根据已知的状态数画出状态图;
- 2) 对状态进行编码;
- 3) 画出编码后的卡诺图形式的状态表 (状态矩阵);
- 4) 写出状态方程;
- 5) 选择触发器类型 (本课程要求D或JK);
- 6) 求出触发器的最简激励方程;  状态方程法 (掌握)
- 7) 画出规范的电路图;
- 8) 进行必要的讨论。  激励表法、卡诺图分区法、
状态图法 (自学)

7.1 计数器

计数器是一种对CP脉冲进行计数的逻辑电路，它的状态图为一个环，环中状态数 m 称为计数器的模。



CP脉冲的有效
沿到来时改变状
态

计数器的分类

按数制：二进制计数器、十进制计数器、
任意进制计数器

按功能：加法计数器、减法计数器、可逆计数器

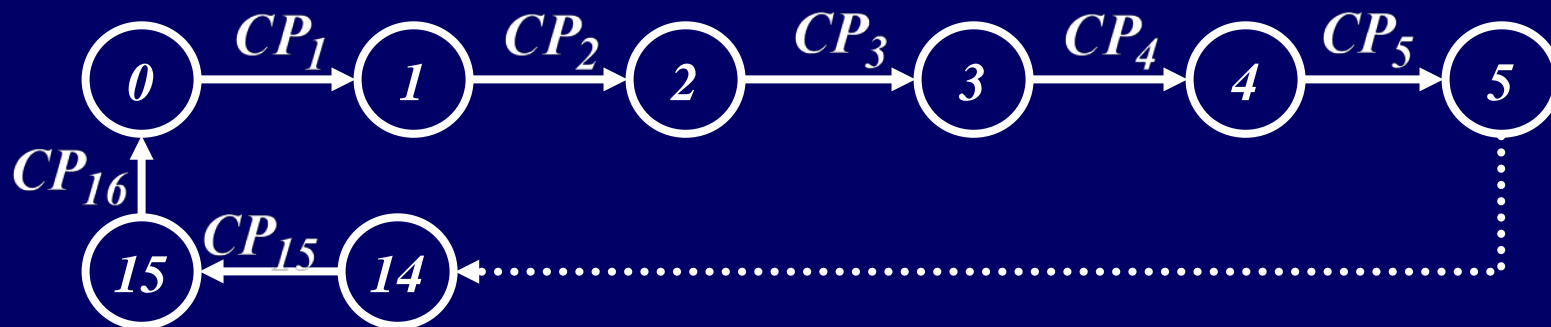
按时序：异步计数器（串行计数）、
同步计数器（并行计数）

按码制：任意编码计数器

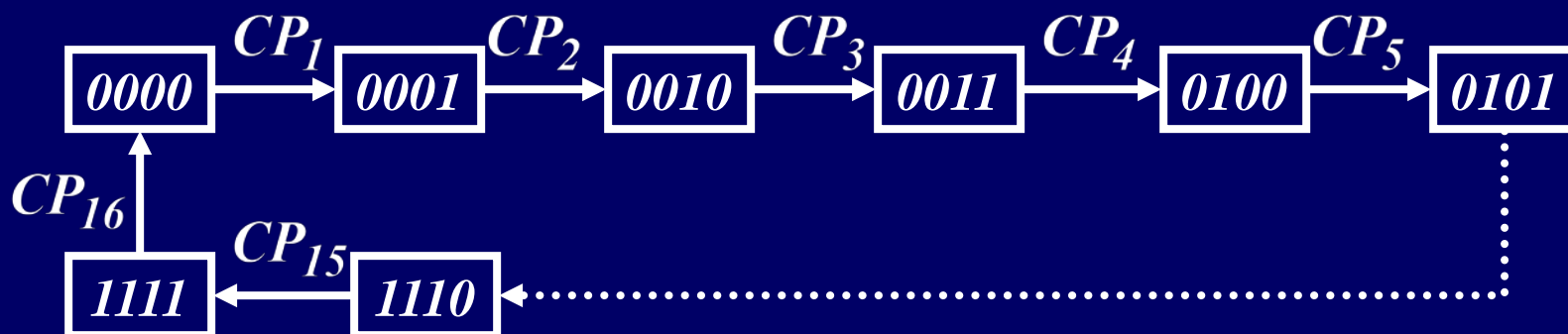
7.1.1 二进制同步计数器的设计与描述

例7.1：试用D触发器设计一个四位二进制同步加1计数器。

1) 状态图：



2) 状态编码： $Q_4Q_3Q_2Q_1$



状态转移表（次态真值表）

$Q_{4(t)}$	$Q_{3(t)}$	$Q_{2(t)}$	$Q_{1(t)}$	$Q_{4(t+1)}$	$Q_{3(t+1)}$	$Q_{2(t+1)}$	$Q_{1(t+1)}$
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
.....						
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

3) 状态矩阵:

		$Q_{3(t+1)}Q_{2(t+1)}Q_{1(t+1)}Q_{0(t+1)}$			
Q_3Q_2	Q_1Q_0				
		00	01	11	10
00		0001	0010	0100	0011
01		0101	0110	1000	0111
11		1101	1110	0000	1111
10		1001	1010	0100	0011

4) 状态方程:

$$Q_{3(t+1)} = \overline{Q_3}Q_2Q_1Q_0 + Q_3\overline{Q_1} + Q_3\overline{Q_2} + Q_3\overline{Q_0}$$

$$Q_{2(t+1)} = Q_2\overline{Q_1} + Q_2\overline{Q_0} + \overline{Q_2}Q_1Q_0$$

$$Q_{1(t+1)} = \overline{Q_1}Q_0 + Q_1\overline{Q_0}$$

$$Q_{0(t+1)} = \overline{Q_0}$$

5) 采用D触发器实现, $Q_{(t+1)}=D$ 。

6) 与求出的状态方程比对, 即可直接得出激励方程:

$$D_4 = \overline{Q_4}Q_3Q_2Q_1 + Q_4\overline{Q_2} + Q_4\overline{Q_3} + Q_4\overline{Q_1}$$

$$D_3 = Q_3\overline{Q_2} + Q_3\overline{Q_1} + \overline{Q_3}Q_2Q_1$$

$$D_2 = \overline{Q_2}Q_1 + Q_2\overline{Q_1}$$

$$D_1 = \overline{Q_1}$$

整理可得

$$D_4 = (Q_3Q_2Q_1) \oplus Q_4$$

$$D_3 = (Q_2Q_1) \oplus Q_3$$

$$D_2 = Q_2 \oplus Q_1$$

$$D_1 = \overline{Q_1}$$

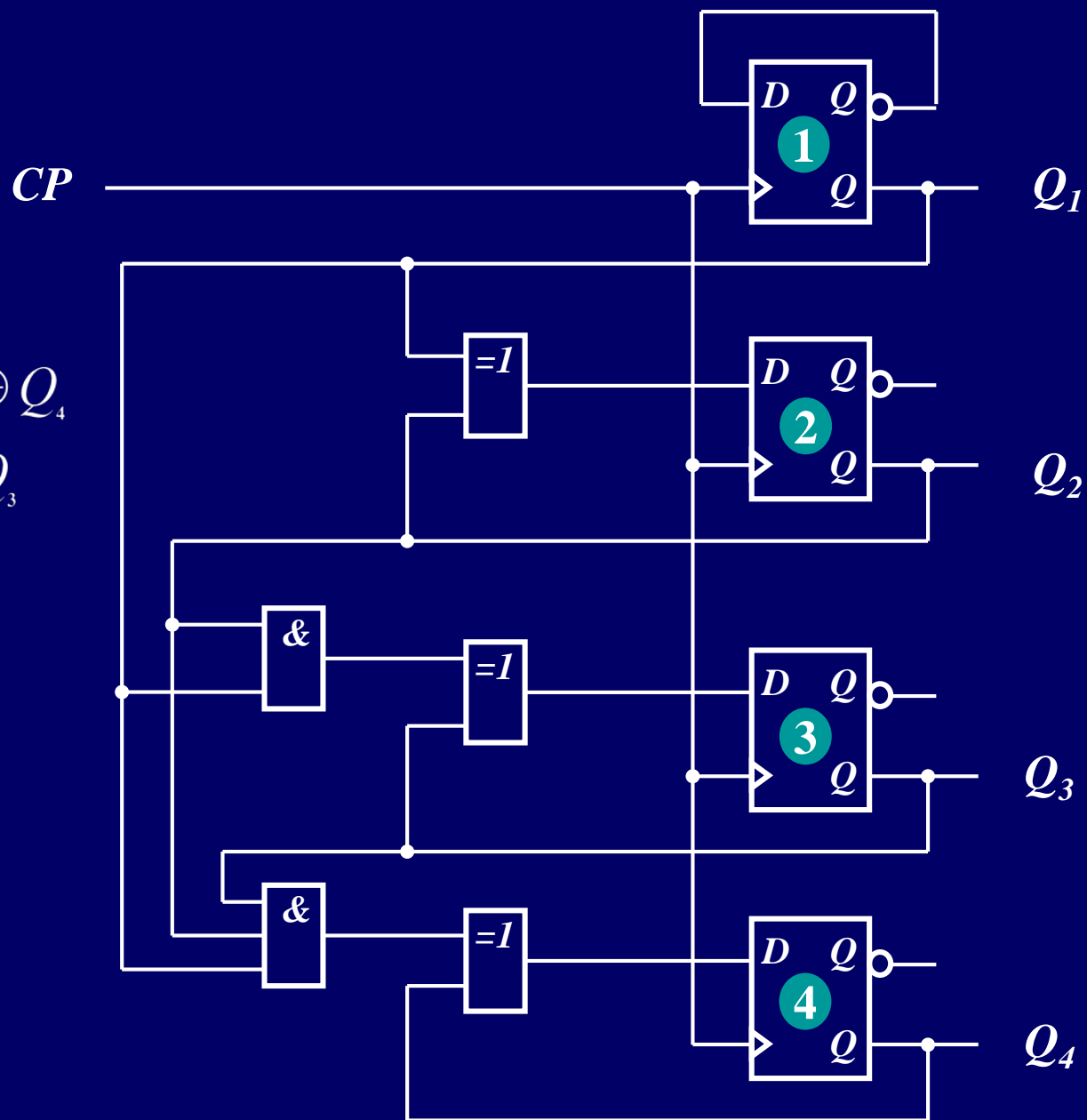
7) 画出电路图

$$D_4 = (Q_3 Q_2 Q_1) \oplus Q_4$$

$$D_3 = (Q_2 Q_1) \oplus Q_3$$

$$D_2 = Q_2 \oplus Q_1$$

$$D_1 = \overline{Q_1}$$



8) 分析

$$D_4 = (Q_3 Q_2 Q_1) \oplus Q_4$$

$$D_3 = (Q_2 Q_1) \oplus Q_3$$

$$D_2 = Q_2 \oplus Q_1$$

$$D_1 = \overline{Q_1}$$

当采用D触发器构造二进制同步计数器时，随位数的增加，触发器输入端D的表达式结构是有规律的。即任意位二进制加1计数器，采用D触发器设计时，满足：

$$D_i = (Q_{i-1} \cdot Q_{i-2} \cdot \cdots \cdot Q_1) \oplus Q_i \quad i \neq 1$$

$$D_1 = \overline{Q_1}$$

例7.2: 试用JK触发器设计一个四位二进制同步加1计数器。

1) ~ 4) 同例1, 得到状态方程并整理JK触发器形式

$$\begin{aligned}Q_{4(t+1)} &= \overline{Q_4}Q_3Q_2Q_1 + Q_4\overline{Q_2} + Q_4\overline{Q_3} + Q_4\overline{Q_1} \\&= Q_3Q_2Q_1 \cdot \overline{Q_4} + \overline{Q_3Q_2Q_1} \cdot Q_4\end{aligned}$$

$$\begin{aligned}Q_{3(t+1)} &= Q_3\overline{Q_2} + Q_3\overline{Q_1} + \overline{Q_3}Q_2Q_1 \\&= Q_2Q_1 \cdot \overline{Q_3} + \overline{Q_2Q_1} \cdot Q_3\end{aligned}$$

$$Q_{2(t+1)} = Q_1 \cdot \overline{Q_2} + \overline{Q_1} \cdot Q_2$$

$$\begin{aligned}Q_{1(t+1)} &= \overline{Q_1} \\&= 1 \cdot \overline{Q_1} + 0 \cdot Q_1\end{aligned}$$

5) 用JK触发器

$$Q_{(t+1)} = J\overline{Q} + \overline{K}Q$$

6) 比较, 求出 J_i 和 K_i

6) 激励方程

$$J_4=K_4=Q_3Q_2Q_1$$

$$J_3=K_3=Q_2Q_1$$

$$J_2=K_2=Q_1$$

$$J_1=K_1=1$$

7) 电路图

8) 分析

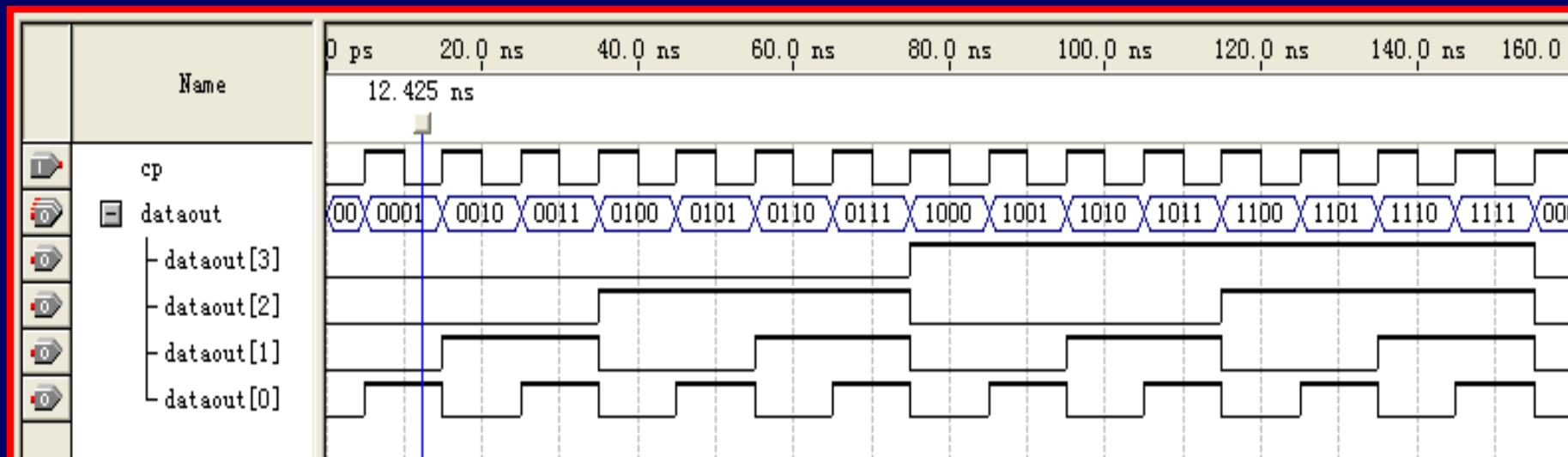
当采用JK触发器构造二进制同步计数器时，随位数的增加，触发器输入端J、K的表达式结构是有规律的。即任意位二进制加1计数器，采用JK触发器设计时，满足：

$$K_i = J_i = Q_{i-1} \cdot Q_{i-2} \cdot \cdots \cdot Q_1 \quad i \neq 1$$

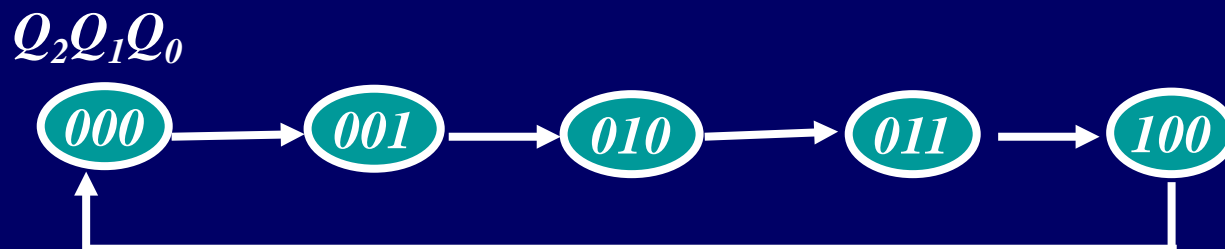
$$J_1 = K_1 = 1$$

上述逻辑电路也可以用Verilog HDL语言来描述，然后通过EDA工具的输入、综合、适配、仿真、下载等设计步骤，在可编程器件中形成电路。

```
module counter (cp, dataout) ;  
input cp ;  
output [3:0] dataout ;  
reg [3:0] dataout ;  
always @ (posedge cp)  
dataout <= dataout + 1;  
endmodule
```



例7.3：用JK触发器设计二进制模5计数器。状态图如下：



解：需要三个触发器，有三个无关态101、110、111。

状态表

Q_1Q_0					
		00	01	11	10
Q_2	0	001	010	100	011
	1	000	×	×	×

$Q_{2(t+1)}Q_{1(t+1)}Q_{0(t+1)}$

$$Q_{2(t+1)} = Q_1Q_0\overline{Q_2} = Q_1Q_0\overline{Q_2} + 1 \cdot Q_2$$

$$J_2 = Q_1Q_0 \quad K_2 = 1$$

$$Q_{1(t+1)} = \overline{Q_1}Q_0 + Q_1\overline{Q_0}$$

$$J_1 = K_1 = Q_0$$

$$Q_{0(t+1)} = \overline{Q_2}Q_0 = \overline{Q_2}Q_0 + 1 \cdot Q_0$$

$$J_0 = \overline{Q_2} \quad K_0 = 1$$

依据激励方程给出无关状态的检查：

$$J_2 = Q_1 Q_0$$

$$J_1 = Q_0$$

$$J_0 = \bar{Q}_2$$

$$K_2 = 1$$

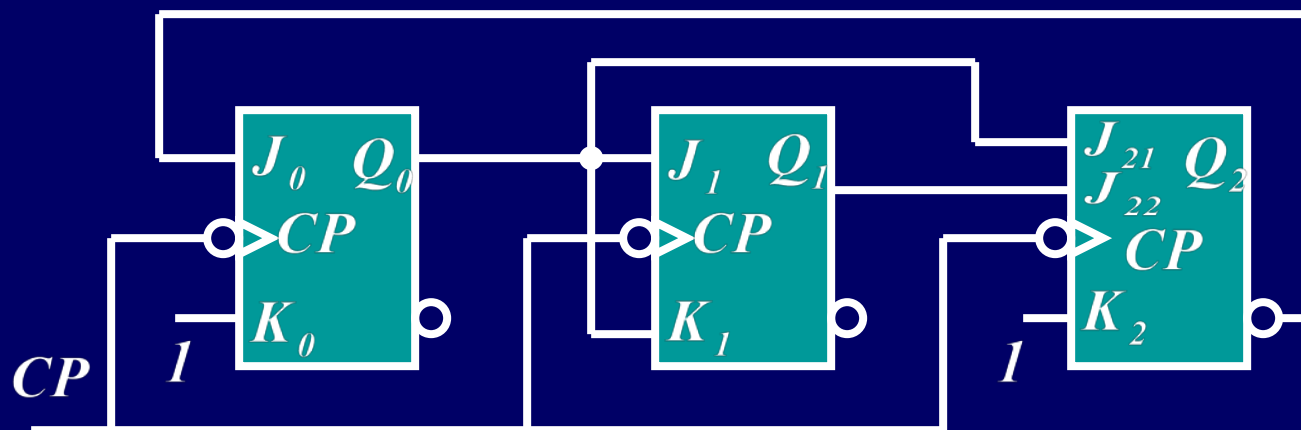
$$K_1 = Q_0$$

$$K_0 = 1$$

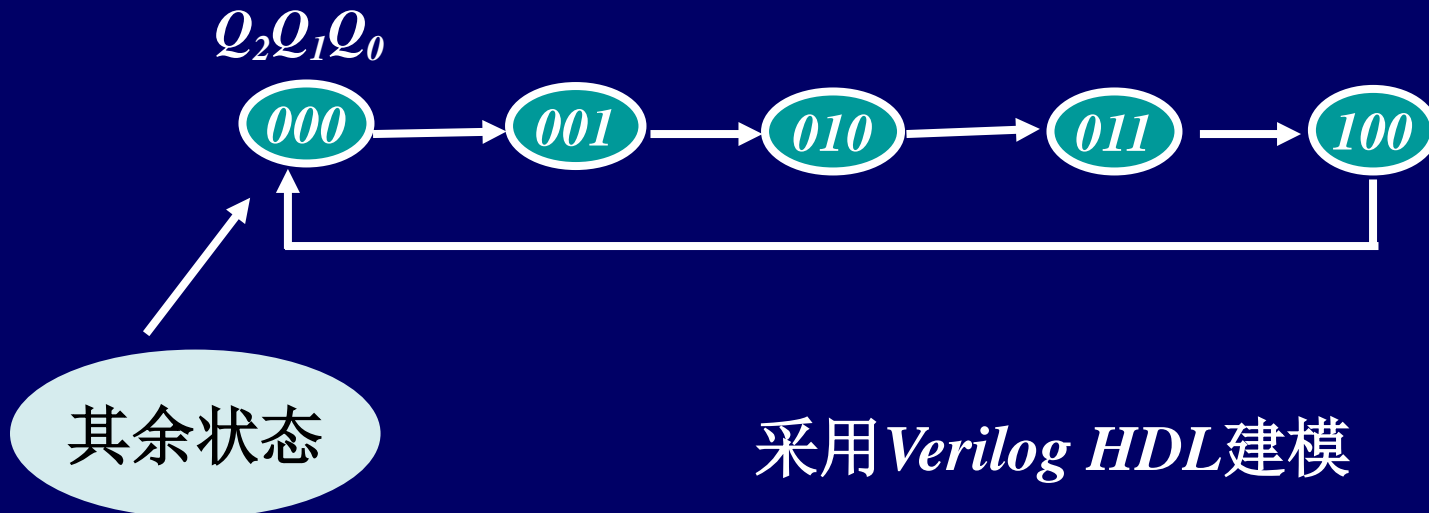
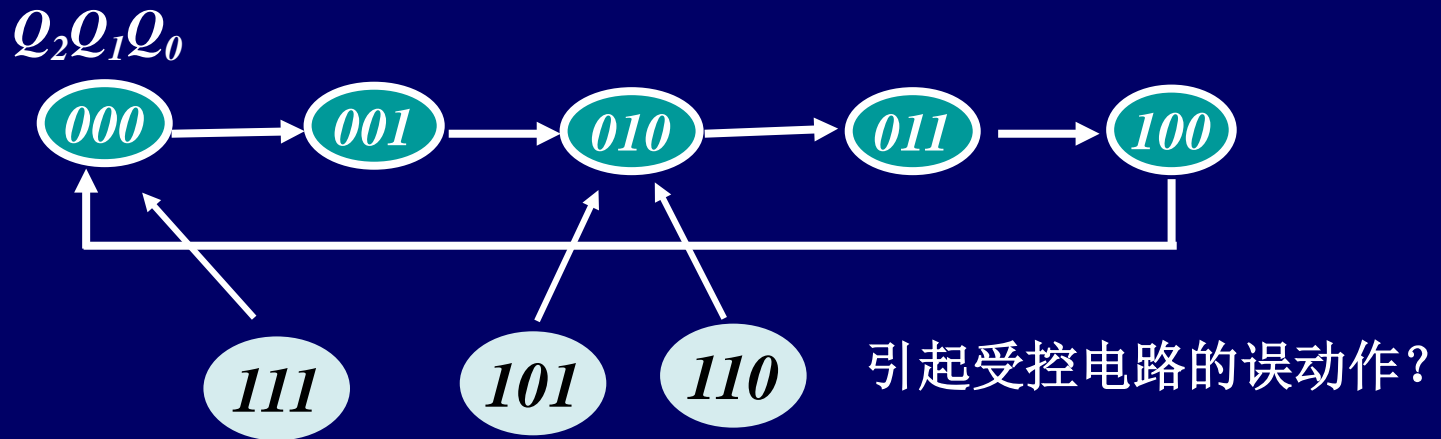
$Q_{2 \sim 0}$	$J_2 K_2$	$J_1 K_1$	$J_0 K_0$	$Q_{2 \sim 0(t+1)}$
101	01	11	01	010
110	01	00	01	010
111	11	11	01	000

经检查，本设计可以自启动。

电路图：



画出完全状态图，实用性分析。

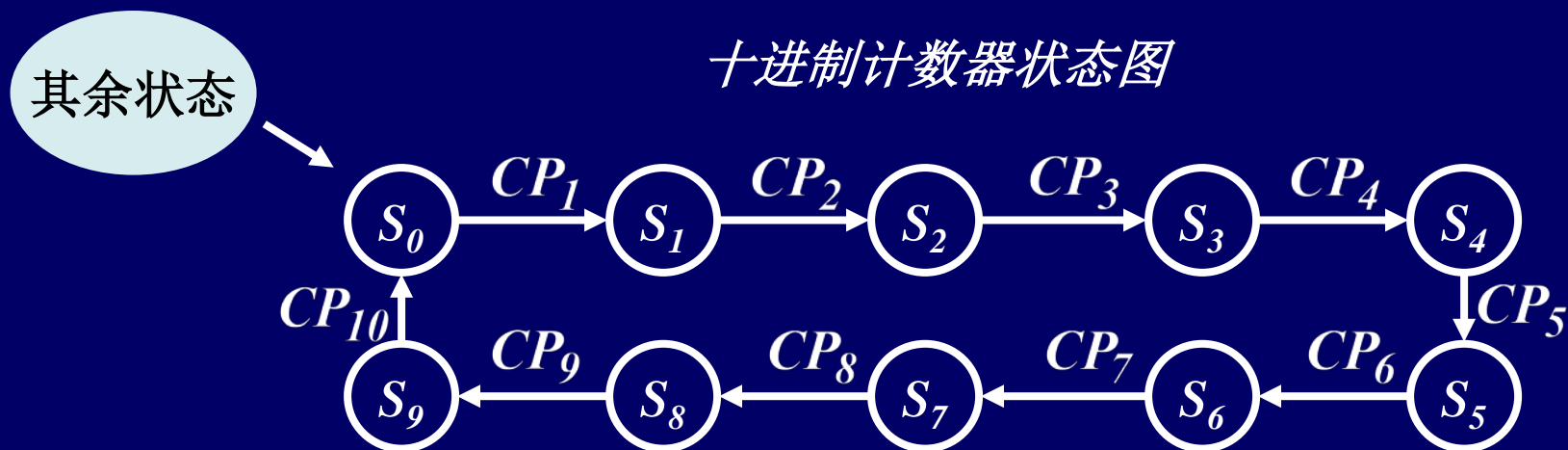


```
module M5_counter ( clk, q ) ;  
    input  clk ;  
    output [2:0] q ;  
    reg    [2:0] q ;  
    always @ (negedge clk)  
        case (q)  
            3'b000 : q <= 3'b001 ;  
            3'b001 : q <= 3'b010 ;  
            3'b010 : q <= 3'b011 ;  
            3'b011 : q <= 3'b100 ;  
            3'b100 : q <= 3'b000 ;  
            default : q <= 3'b000 ;  
        endcase  
    endmodule
```

与状态转移表存在
某种对应关系。

7.1.2 多种编码十进制计数器的Verilog HDL模型

例7.4：建立多种编码十进制计数器的状态图。



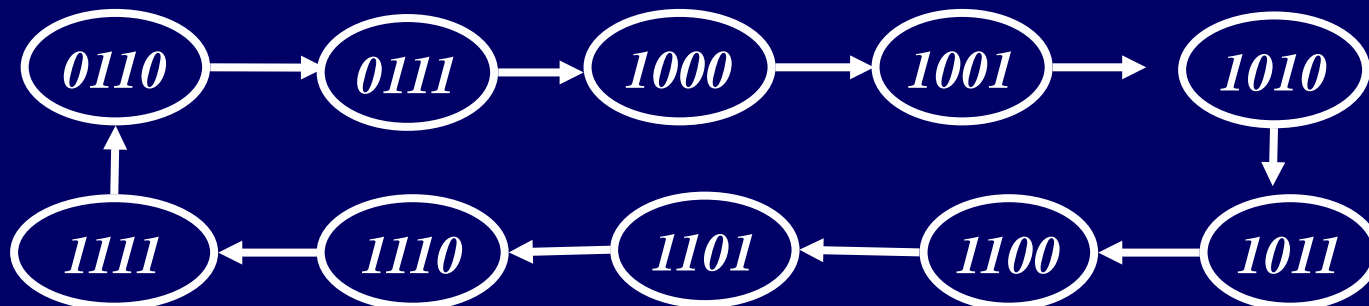
分析：应采用4个触发器（ $Q_3Q_2Q_1Q_0$ ），6个无关态的次态设为 S_0

按要求对 $S_0 \sim S_9$ 进行编码，可得到任意编码十进制计数器的状态图。

$Q_3Q_2Q_1Q_0$

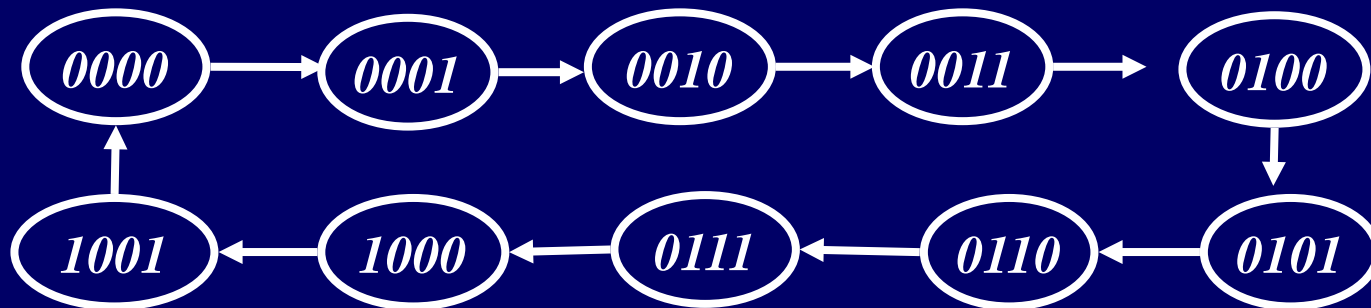
其余状态

一种自然编码十进制计数器状态图



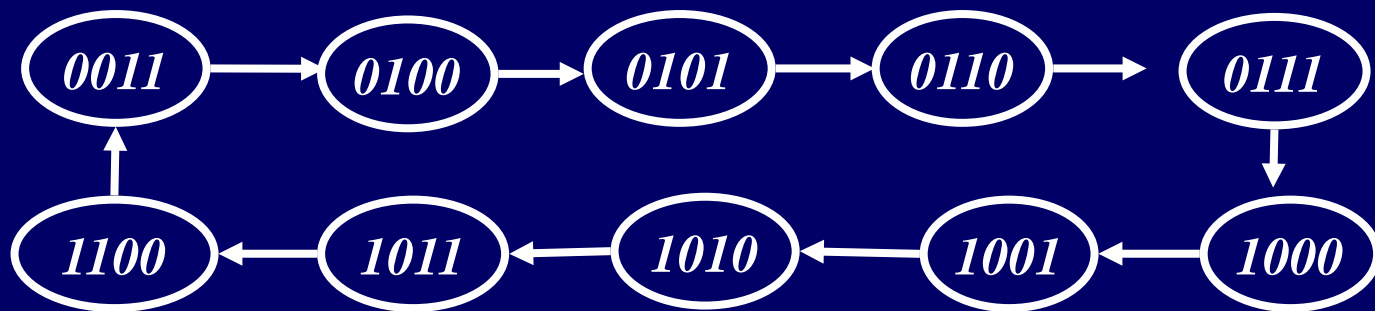
其余状态

8421码十进制计数器状态图



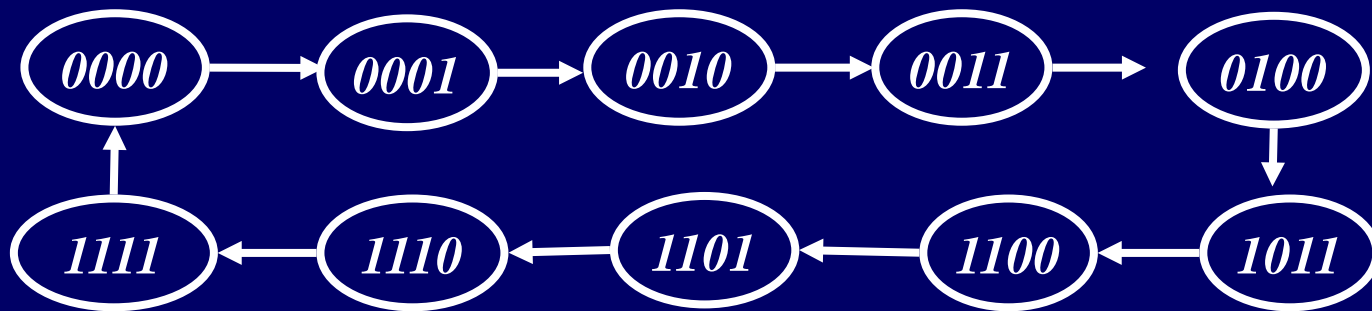
其余状态

余3码十进制计数器状态图



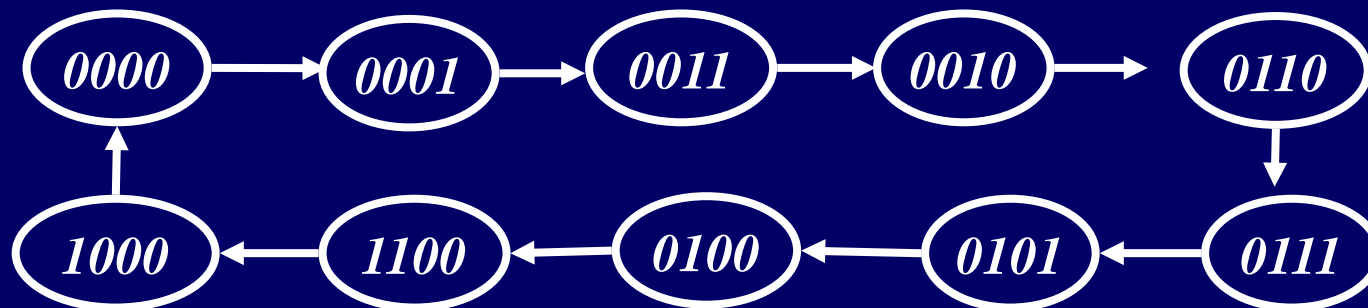
其余状态

2421码十进制计数器状态图



其余 状态

格雷BCD码十进制计数器状态图



能否建立一个基于Verilog HDL
的十进制计数器基础模型？

```

module M10_counter ( clk, reset, q, rco );
    parameter  S0=4'B0000, S1=4'B0001, S2=4'B0010,
               S3=4'B0011, S4=4'B0100, S5=4'B0101,
               S6=4'B0110, S7=4'B0111, S8=4'B1000, S9=4'B1001;
    input  clk, reset;
    output [3:0] q;
    output rco;
    reg    [3:0] q;
    assign rco = (q==S9) ? 1 : 0; //增加进位输出
    always @ (posedge clk)
        begin
            if (!reset) q<=S0;      //同步复位
            else case (q)           //状态转移表描述
                S0 : q <= S1;
                S1 : q <= S2;
                .....
                S9 : q <= S0;
                default : q <= S0; //其他状态
            endcase
        end
    endmodule

```

修改，实现不同编码的十进制计数器。

若用门级描述？
通用性？

基于*Verilog HDL*的
同步时序电路设计模板

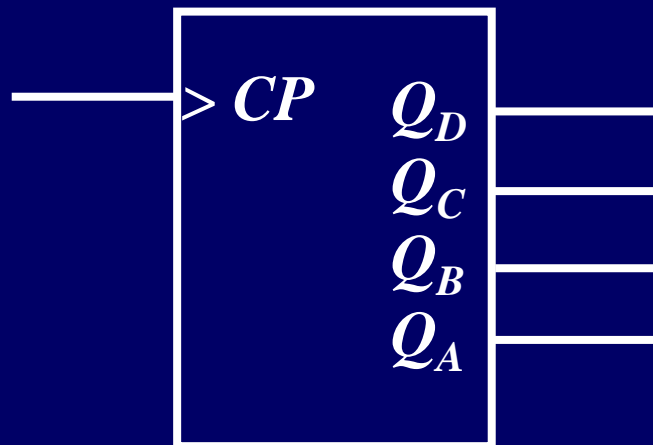

```
module 模块名（端口名列表）；  
    parameter //定义参数  
    input      //定义输入  
    output     //定义输出  
    reg        //定义always中被赋值变量  
    //门级描述组合部分  
    assign     //描述组合部分  
    always @ (组合输入信号列表)  
        begin  
            //阻塞赋值（=）  
            //if_else、case、for行为语句  
        end  
    always @(边沿信号列表) // 时序部分描述  
        begin  
            //非阻塞赋值（<=）  
            //if_else、case、for行为语句  
        end  
endmodule
```

作业15:

7.4 (2)

7.5 (1, 5)

7.1.3 MSI计数器74LS163及其应用



思考：电路的实用性？

电路的复位状态？

电路状态的预置？

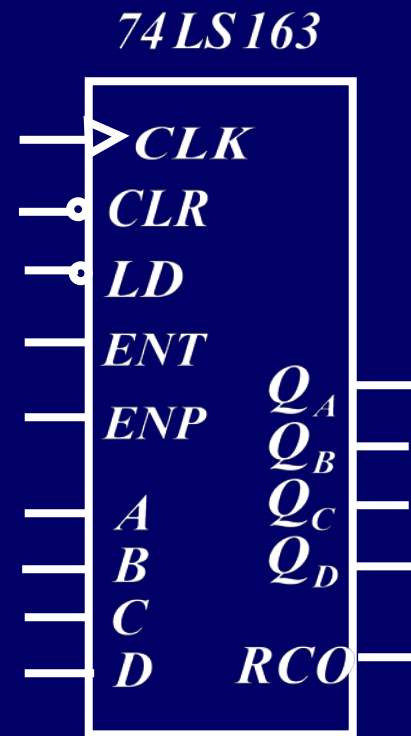
如何满足迭代设计的要求？（使能输入、模 溢出）

1. 74LS163的逻辑功能

74LS163是一个最常用的四位同步二进制计数器，是一个全同步器件，其输出仅在 CLK 上升沿改变。

功能：

同步清“0”；
同步置数；
状态保持；
加“1”计数。



逻辑符号

74LS163功能表

\overline{CLR}	\overline{LD}	ENP	ENT	CLK	$Q_D Q_C Q_B Q_A$	RCO	说明
0	×	×	×	↑	0 0 0 0	0	清零
1	0	×	×	↑	$D C B A$	*	并行预置
1	1	1	1	↑	计数	*	加1计数
1	1	0	×	×	$Q_D Q_C Q_B Q_A$	*	保持
1	1	×	0	×	$Q_D Q_C Q_B Q_A$	0	保持

*: 当计数器计为1111, 且 $ENT = 1$ 时, $RCO = 1$ 。

Q_D 是高位, Q_A 是低位。

基于Verilog HDL的设计模型

```
module v163 (clrn, clk, enp, ent, ldn, din, qout, rco);  
    input  clrn, clk, ent, enp, ldn;  
    input  [3:0] din;  
    output [3:0] qout;  
    output rco;  
    reg    [3:0] qout;  
    always @(posedge clk)  
        begin  
            if ( ~ clrn ) qout <= 0;  
            else if ( ! ldn ) qout <= din;  
                else if ( enp && ent == 1 ) qout <= qout + 1;  
                    else qout <= qout;  
        end  
    assign rco = ( qout==4'b1111 && ent ) ? 1 : 0;  
endmodule
```

描述时序操作

描述组合逻辑

```
module veri163 (clrn, clk, enp, ent, ldn, din, qout, rco) ;
```

```
  input  clrn, clk, enp, ent, ldn ;
```

```
  input  [3:0] din ;
```

```
  output [3:0] qout ;
```

```
  output rco;
```

```
  reg    [3:0] qout ;
```

```
  reg    rco ;
```

```
    always @ ( posedge clk )
```

```
      begin
```

```
        if ( ! clrn ) qout <= 0 ;
```

```
        else if ( ! ldn ) qout <= din ;
```

```
          else if ( ent && enp == 1 ) qout <= qout + 1 ;
```

```
          else qout <= qout ;
```

```
      end
```

```
    always @ ( ent or qout )
```

```
      begin
```

```
        if ( qout == 4'b1111 && ent ) rco = 1 ;
```

```
        else rco = 0 ;
```

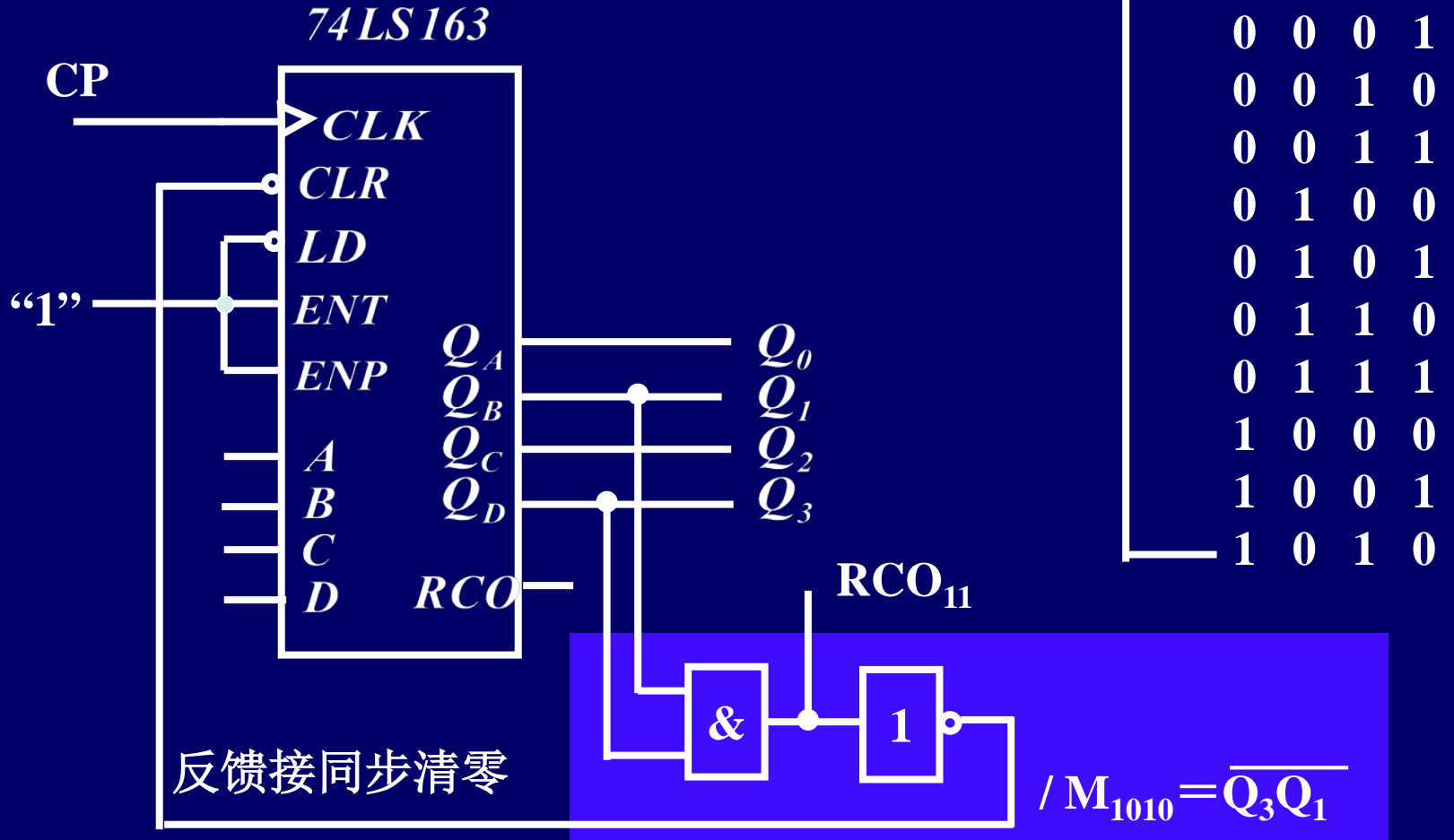
```
      end
```

```
endmodule
```

组合逻辑的另外一种描述！

2. 基于74LS163的小规模 ($M < 16$) 计数器设计。

例7.5：分析下列电路的逻辑功能。



功能：0、1、2、...、10的模11计数器。

这类电路的分析方法：

1) 设定电路初态；

2) 计算反馈；

3) 查功能表；

4) 得到电路的次态；



直到出现状态循环

5) 状态循环中的状态数即为计数器的模。

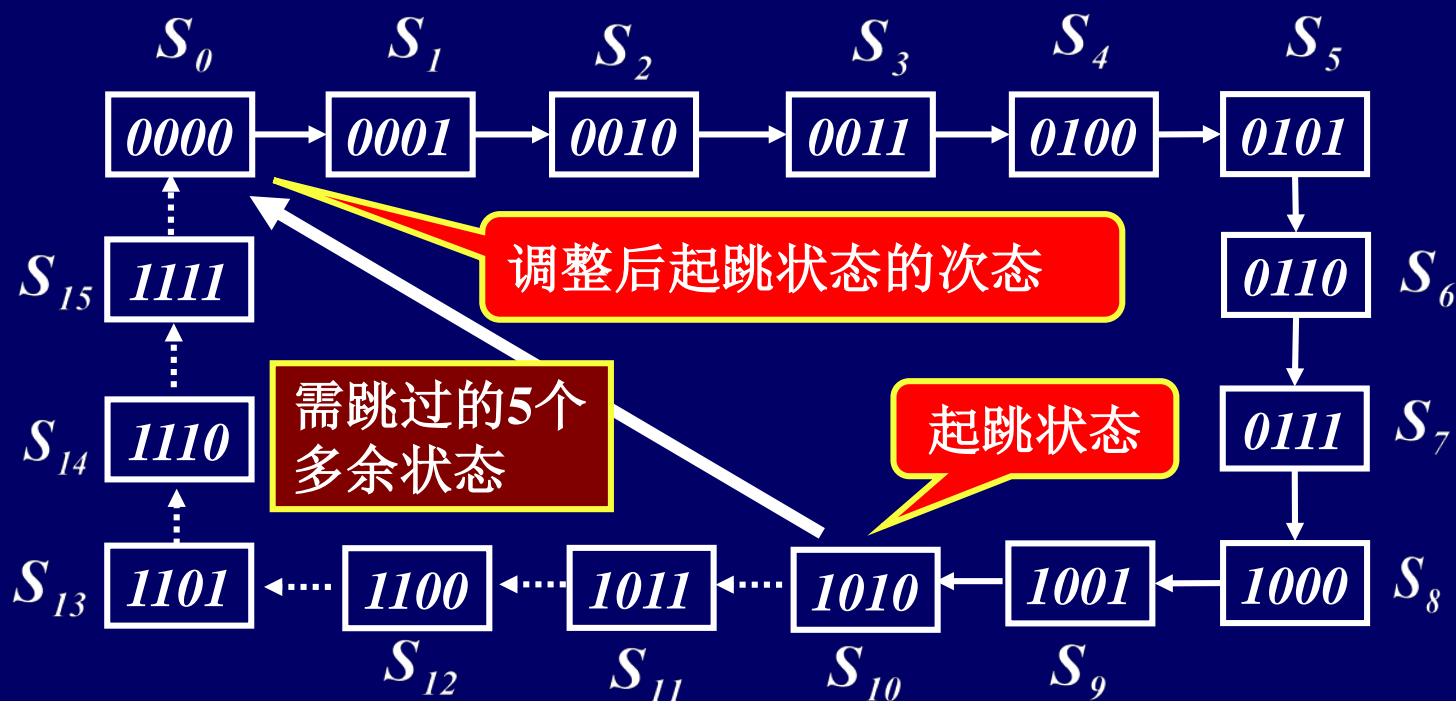
构造这类电路时，涉及到：

1) 跳越的概念

2) 反馈电路的设计

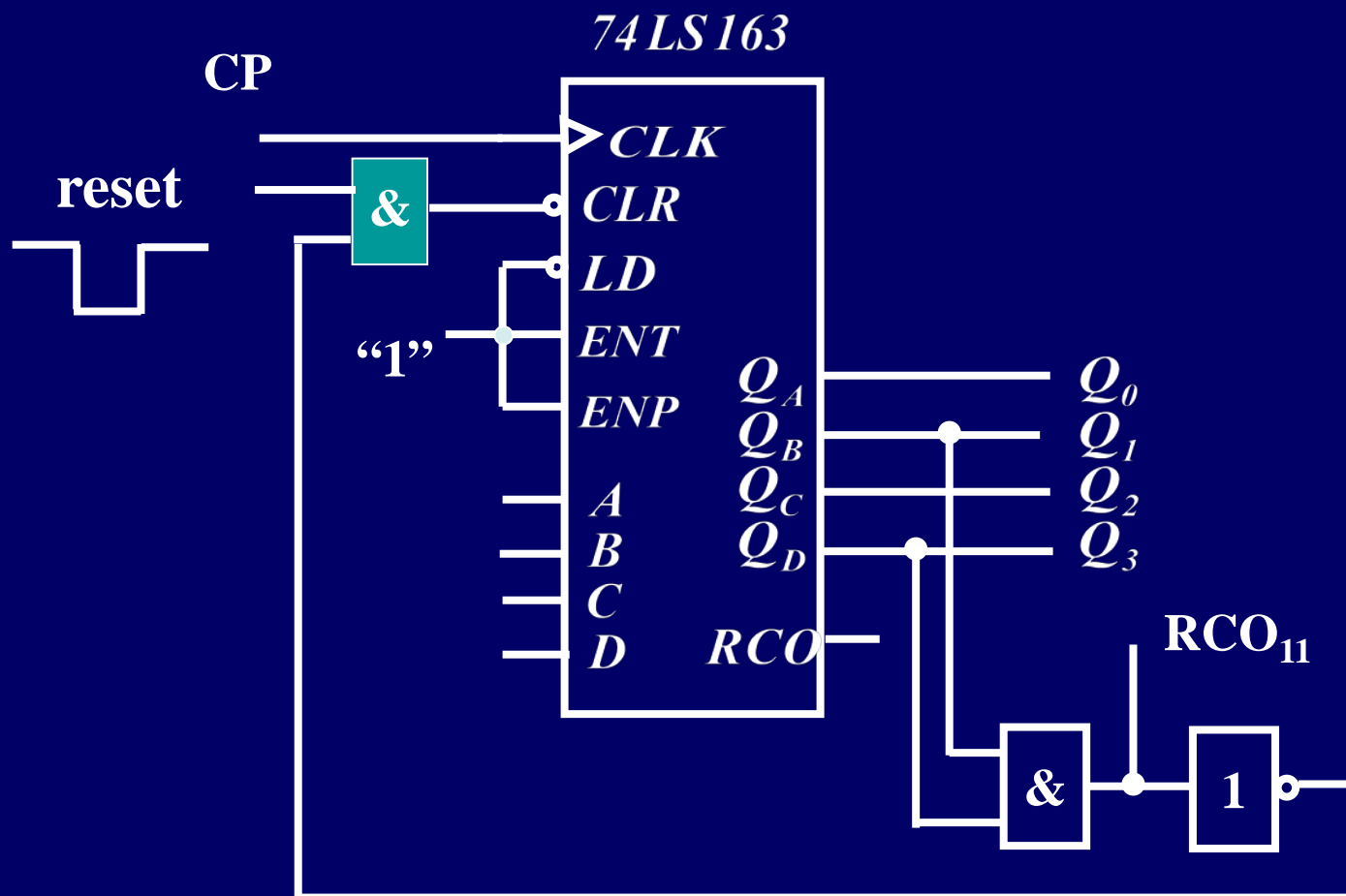
跳越的概念

当利用模为 2^n 的计数器实现模 $m < 2^n$ 的计数器时，需要从 2^n 个状态中跳过 $K = 2^n - m$ 个多余状态，即保证计数器的正常循环中不会出现多余状态。



如果起跳状态为 S_a ，且需跳过 K 个状态，则 S_a 的次态应从 S_{a+1} 调整为 S_{a+K+1} 。

一般情况下，系统的复位信号使电路进入初态 S_0



若电路在某时刻进入多余状态（无用状态），可能出现三种情况。

- 1) 经过有限节拍，自动进入有效状态 循环。
- 2) 停在某个无用状态不再转换。
- 3) 在无用状态中循环。

2)、3) 是设计要解决的问题

例7.5的设计，满足1)，为什么？请分析一下。

反馈电路的设计

预置位法——就是当出现起跳状态 S_a 时，利用组合逻辑产生预置位有效电平，反馈接到控制输入端，在下一个时钟有效沿到来时，使电路进入 S_{a+K+1} 状态。

高有效预置位电平一般由 S_a 状态中所有的“1”相与产生；低有效预置位电平一般由 S_a 状态中所有的“1”与非产生。

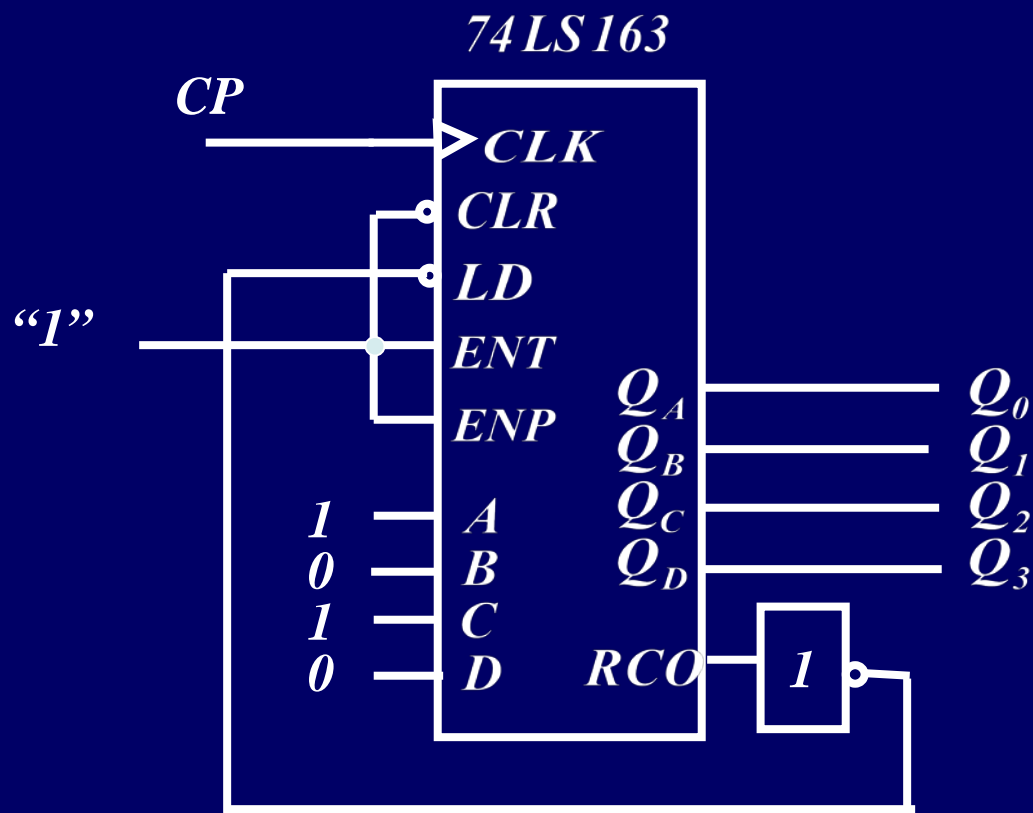
利用单片74LS163构造 $m < 16$ 的任意模数的计数器。
就是采用预置位法完成状态的“跳越”。有两种方式：

1) 当出现次态为“0”的计数器。起跳状态 S_a 时，产生低有效预置位电平信号，反馈接到同步清零输入端（/ CLR ），完成全零的预置，下一个 CLK 上升沿到来时，使 $S_{a+K+1} = 0000$ ，实现跳越

2) 当出现起跳状态 S_a 时，产生低有效预置位电平信号，反馈接到同步置数输入端（/ LD ），完成 $S_{a+K+1} = DCBA$ 的预置，下一个 CLK 上升沿到来时，使 $S_{a+K+1} = DCBA$ ，实现跳越次态为任意值的计数器。

例7.6：用74LS163构造一个计数值为5、6、...、15的模11计数器。

解：分析题目要求，起跳状态1111，可用RCO产生预值位信号，反馈接入同步置数端/LD，输入端DCBA置成跳越次态0101。



例7.7：试用74LS163构造2421码十进制计数器。

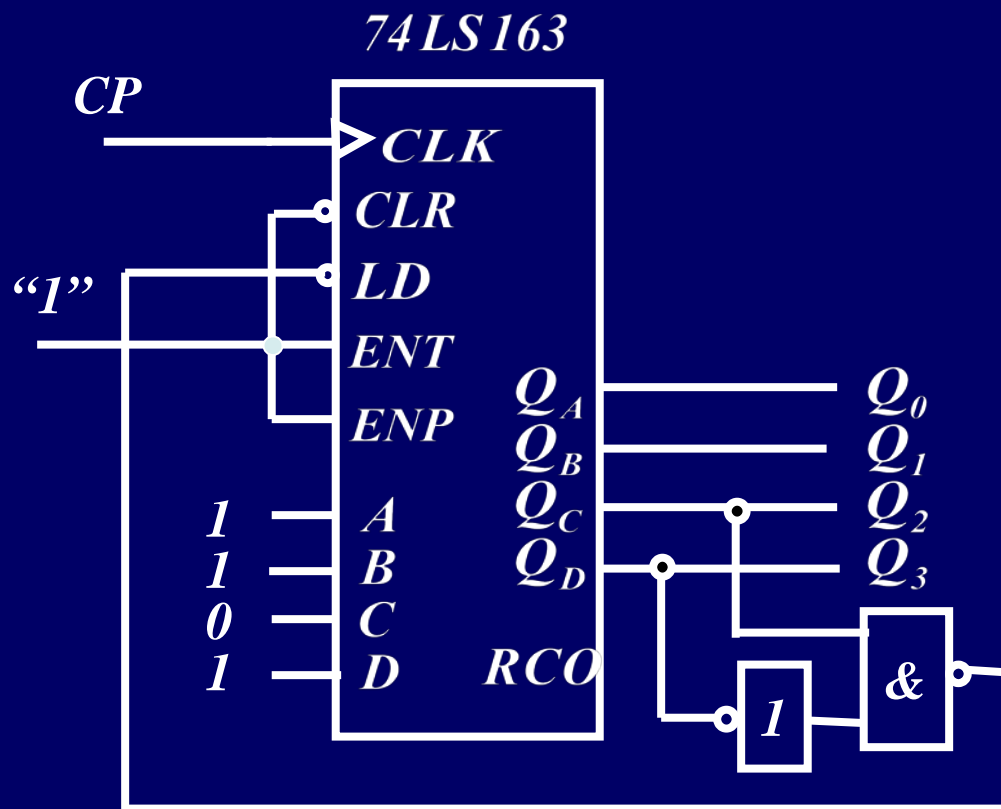
十进制	2421 码 ($Q_3Q_2Q_1Q_0$)
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111

解：分析设计要求，2421码是采用四位二进制表示一位十进制字符的BCD码。不允许0101~1010六种状态出现，所以起跳状态为0100，跳越次态为1011。此时利用起跳状态生成预置位电平时应进行分析，应用 Q_3Q_2 组合生成。

Q_1Q_0		00	01	11	10
Q_3Q_2	00	0	1	3	2
	01	4			
	11	6	7	9	8
	10			5	

$$/M = \overline{Q_3} \cdot \overline{Q_2}$$


十进制	2421码 ($Q_3Q_2Q_1Q_0$)
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111



$$/M = \overline{\overline{Q_3} \cdot \overline{Q_2}}$$

思考：2421码模8计数器的实现

十进制	2421码 ($Q_3Q_2Q_1Q_0$)
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111



1) 用74LS163, 两次跳越, 难度?

2) 用Verilog HDL建模

3 大模数计数器的设计与实现

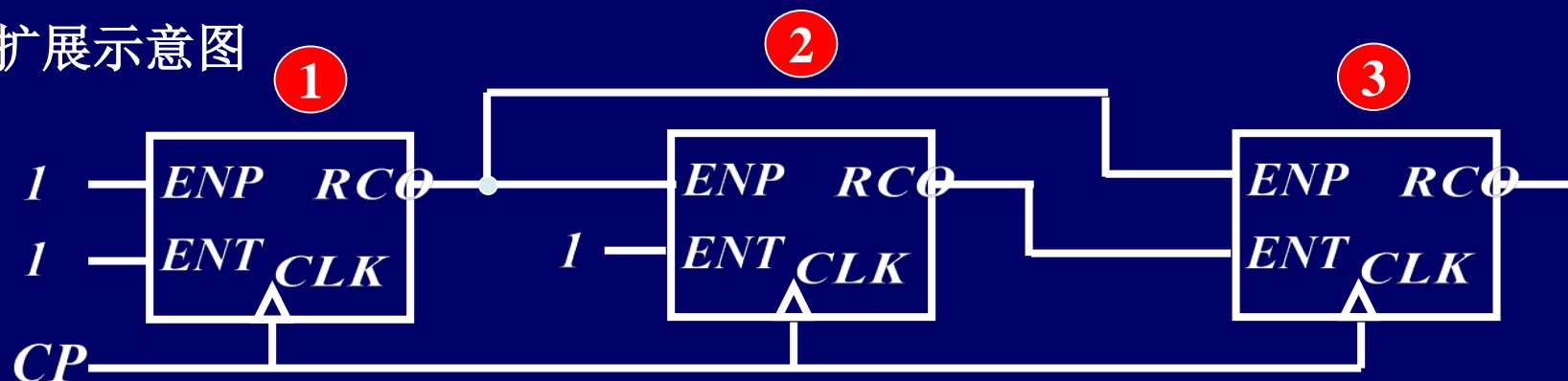
- 多片74LS163级联扩展

- 硬件描述语言建模

当计数器的模 $m > 16$ 时，可使用74LS163级联扩展实现。

思路：利用计数控制输入端 ENT 和 ENP 、向上进位输出端 RCO 、 ENT 与 RCO 的关系进行级联扩展。

扩展示意图



这种级联方式的速度较快，体现在：

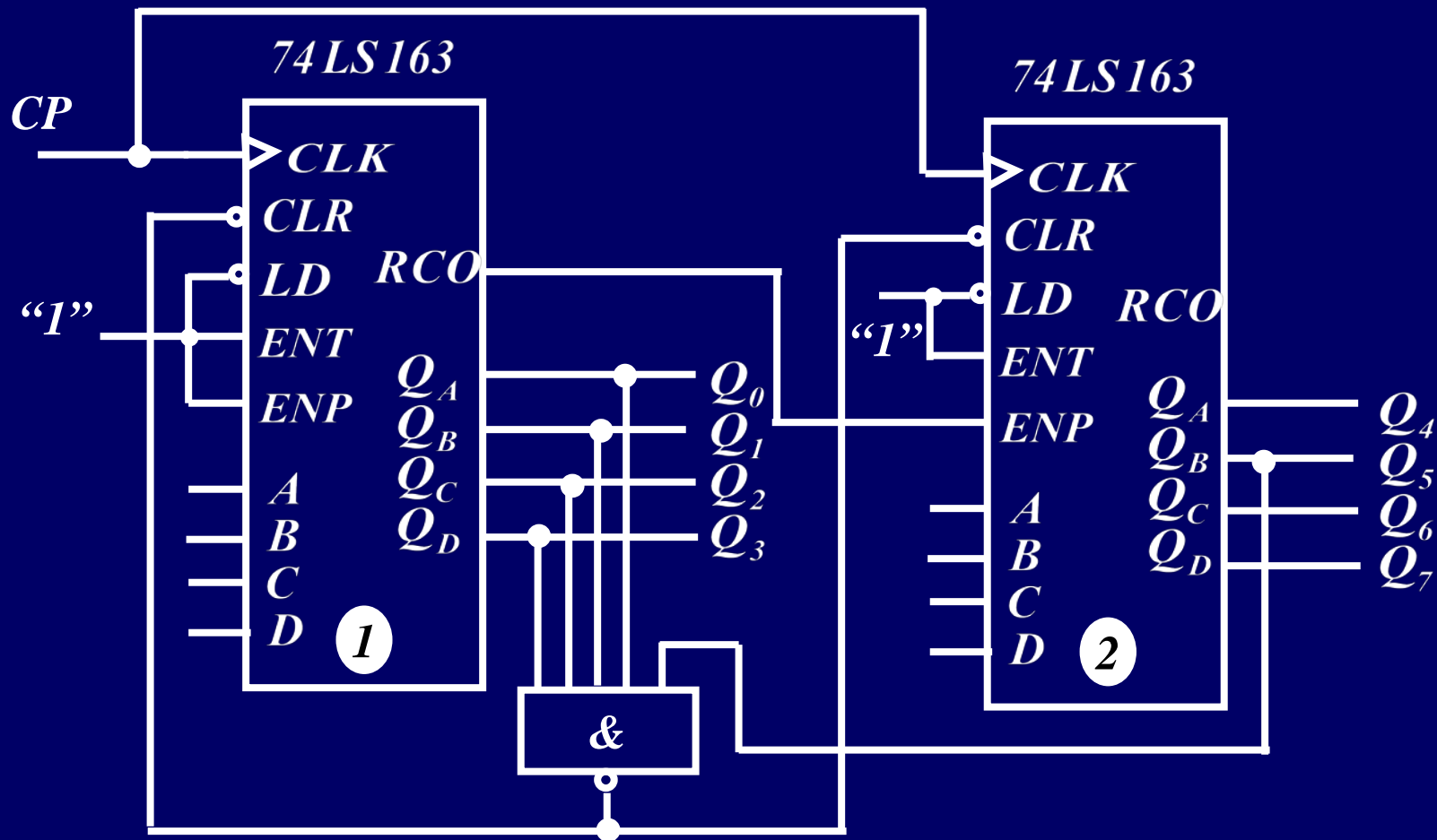
↓
1110

↓
1111

↓
1111

不等1片计到**1111**，2、3两片已提前建立**RCO**，
一旦1片的**RCO**建立，各片立刻计数。

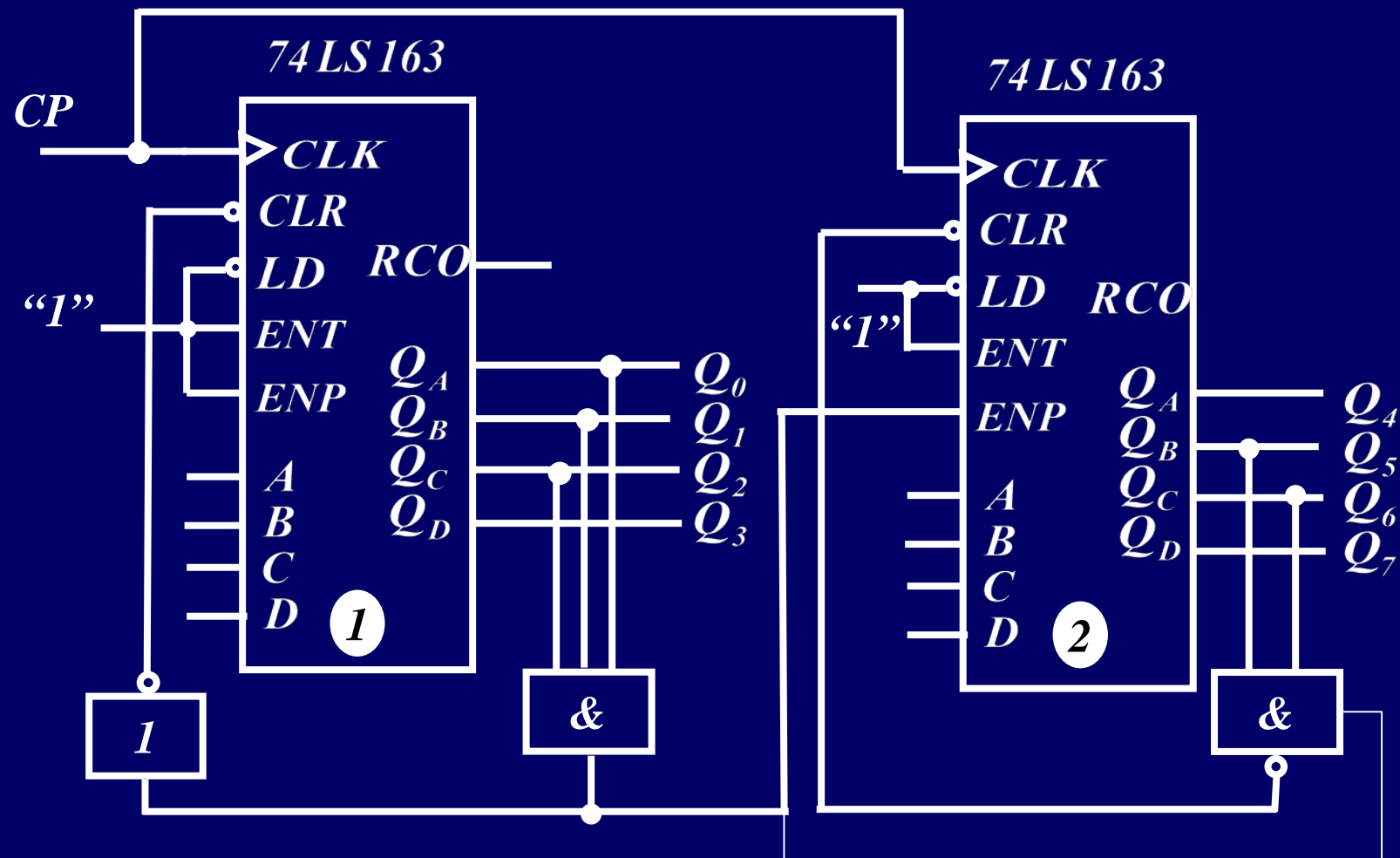
例7.8：分析用74LS163构造的进制计数器的模。



解：两片级联（最大模256），起跳状态为 $(00101111)_2 = (47)_{10}$ ，生成预置位信号反馈接到 \overline{CLR} ，跳越次态0。

所以，计数状态从0、1、2、....、47、0、1、...，模为48。

例7.9：用74LS163构造模56的计数器。???



解： $M1=8$ ， $M2=7$ ，
 $M=M1 \times M2=56$

计数状态： $00h$ $10h$ $60h$
 \downarrow \nearrow \downarrow \downarrow
 $07h$ $17h$ $67h$

基于Verilog HDL的任意模数计数器？

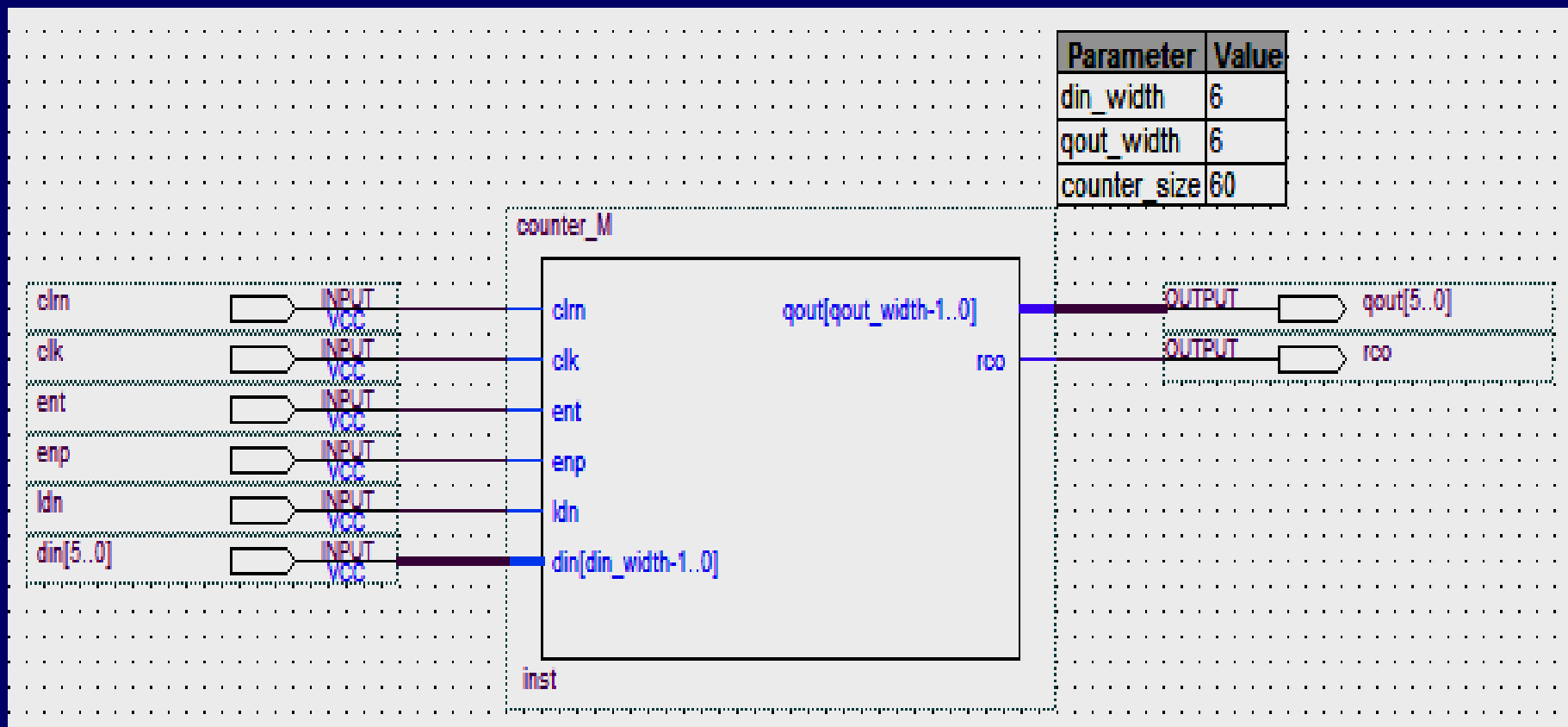
```
module counter_M ( clrn, clk, enp, ent, ldn, din, qout, rco );
    parameter  din_width = 'd6;
    parameter  qout_width = 'd6;
    parameter  counter_size = 'd60;
    input  clrn, clk, ent, enp, ldn ;
    input  [din_width-1 : 0]  din ;
    output [qout_width-1 : 0] qout ;
    output rco ;
    reg    [qout_width-1 : 0] qout ;
    always @ ( posedge clk )
        begin
            if ( ~clrn ) qout <= 0 ;
            else if ( !ldn ) qout <= din;
                else if ( enp && ent==1 )
                    if ( qout==counter_size-1 ) qout <= 0 ;
                    else qout <= qout + 1 ;
                else qout <= qout ;
        end
    assign rco=(qout==counter_size-1 && ent) ? 1 : 0 ;
endmodule
```

参数化设计，M=60

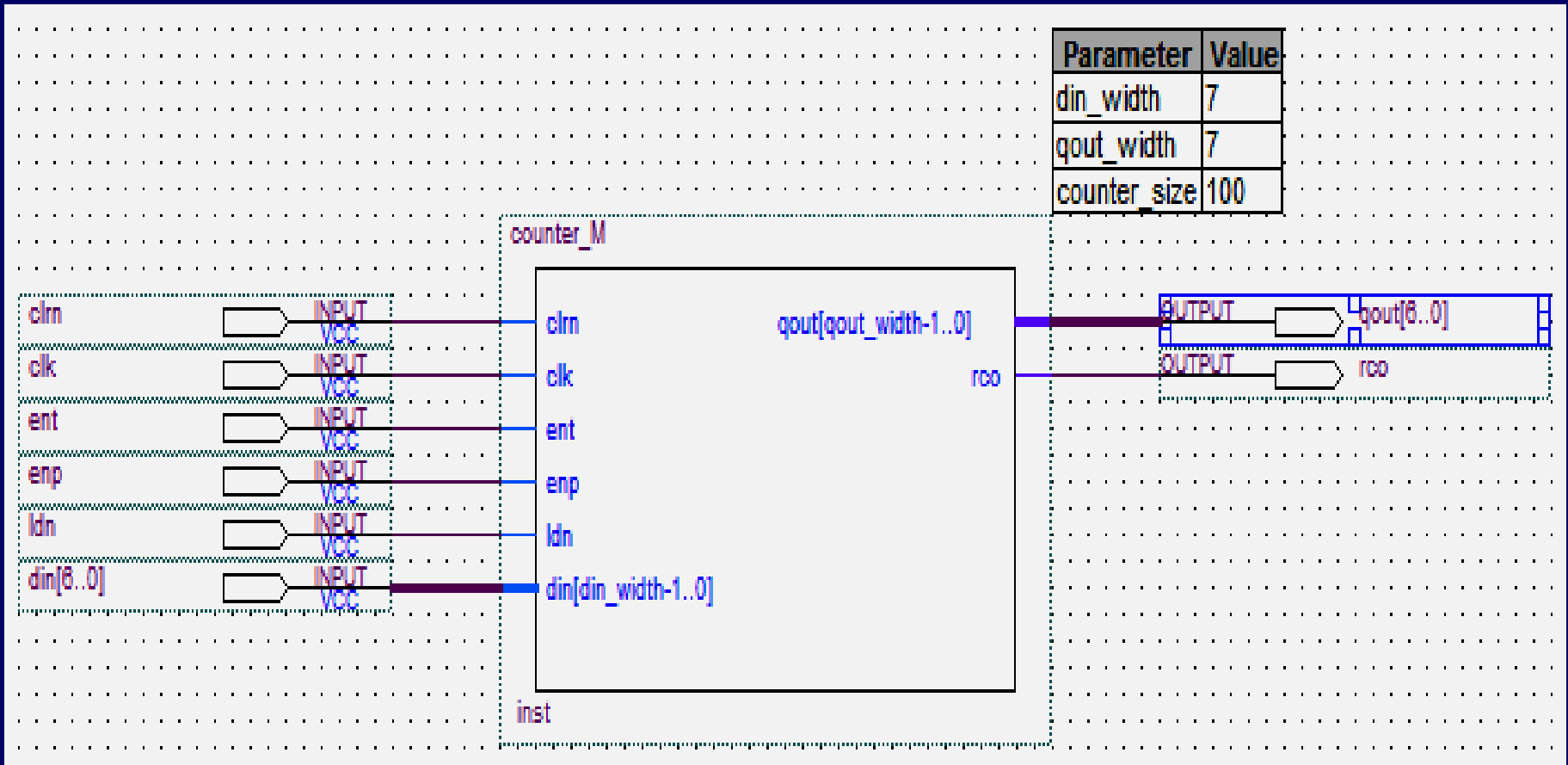
计数状态：
0 ~ counter_size-1

如何实现？
din ~ counter_size-1

在QuartusII中形成参数化的计数器模块

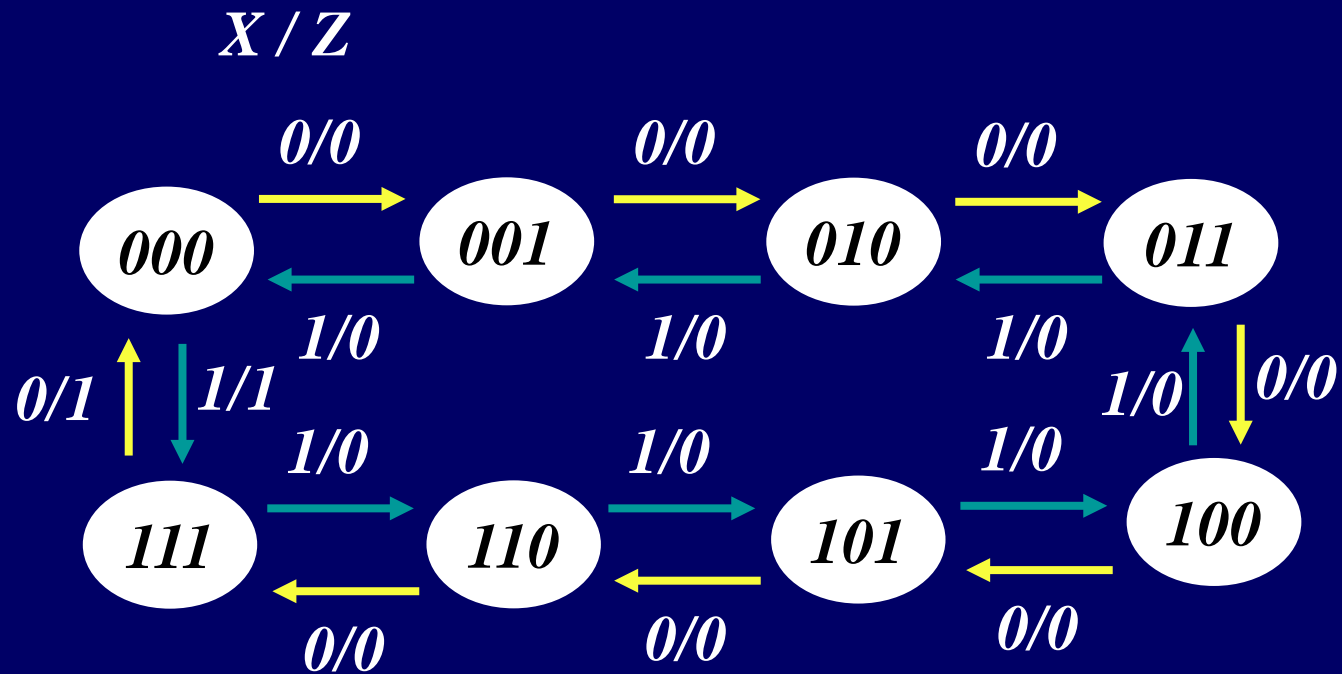


调整参数设计一个模100的计数器：



思考题：

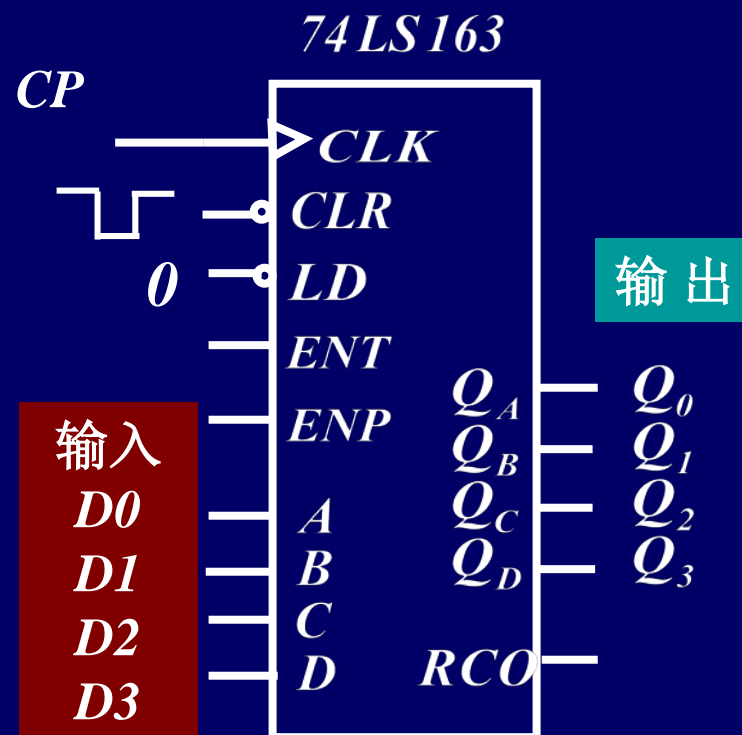
根据可逆计数器的状态图，建立*Verilog HDL*模型。



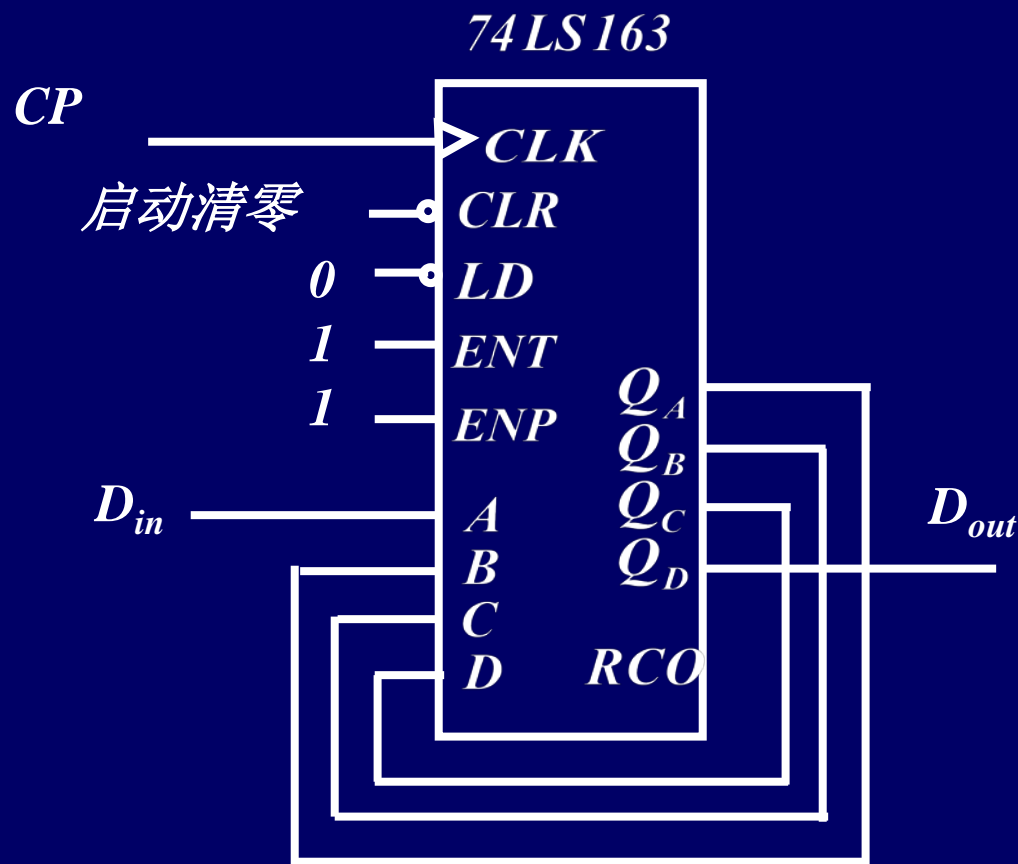
4 74LS163的其它应用

将74LS163计数器用作寄存器

只要将置数控制置输入端/ LD 接“0”，则当 CLK 有效沿到来时，数据输入端 D 、 C 、 B 、 A 的状态就被寄存，反映在 $Q_DQ_CQ_BQ_A$ 。



将74LS163计数器用作移位寄存器



Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	0	D_{in0}
0	0	D_{in0}	D_{in1}
0	D_{in0}	D_{in1}	D_{in2}
D_{in0}	D_{in1}	D_{in2}	D_{in3}

信号延时

串行输入、串行输出

串行输入、并行输出

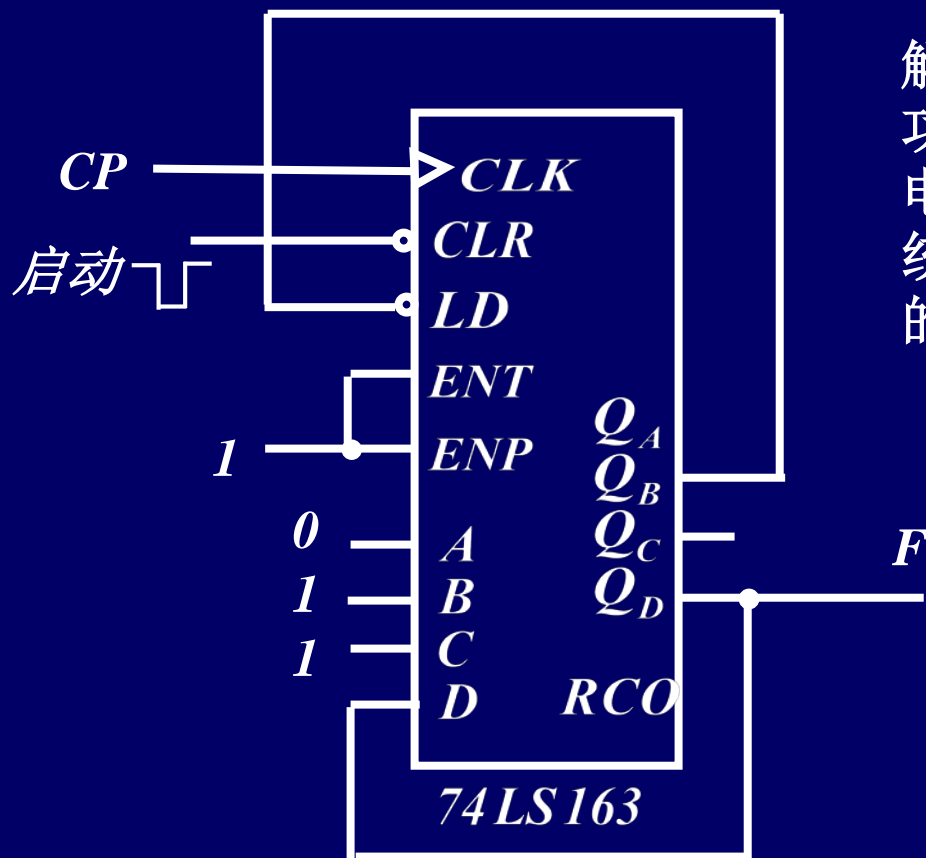
将74LS163用作分频器



逻辑符号

如何获得 $\div 6$ 、 $\div 10$ 、 $\div 12$的分频信号呢？

例：请分析图示电路的逻辑功能。



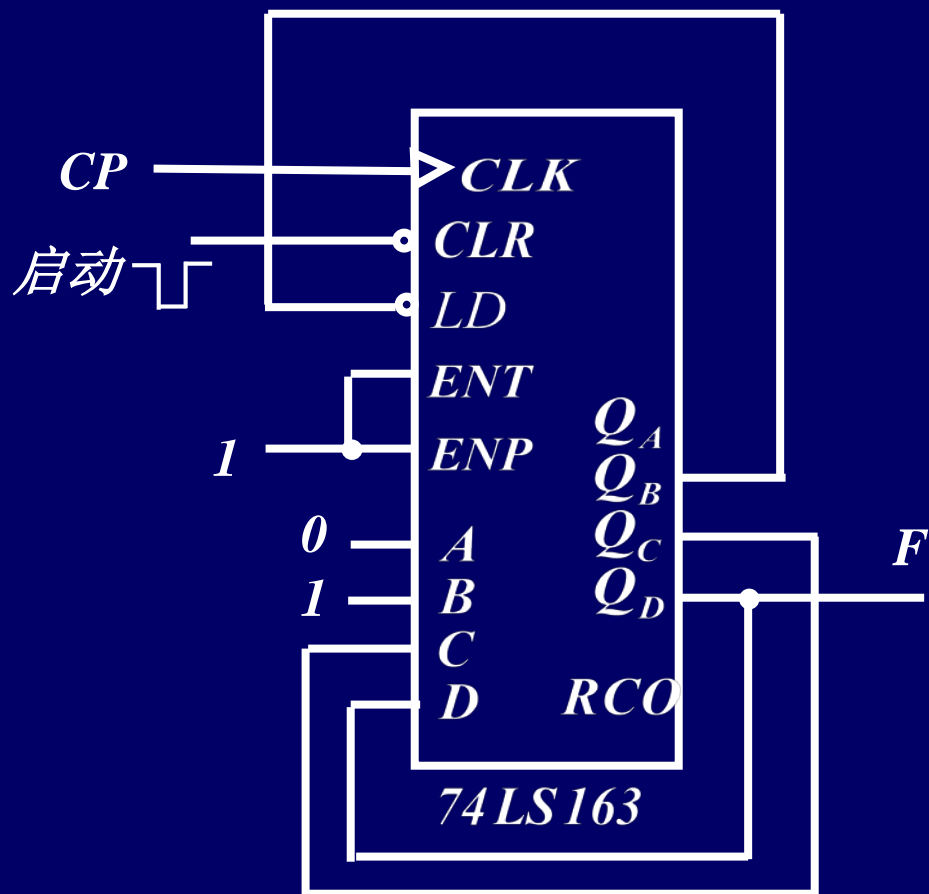
解：这类电路的分析过程是，从启动功能开始，得到初态（现态），根据电路连接及功能表，求得次态；.....统计构成循环的脉冲数，分析输出 F 的特性。

$Q_D Q_C Q_B Q_A$

0	0	0	0
0	1	1	0
0	1	1	1
1	0	0	0
1	1	1	0
1	1	1	1
F			

功能： F 的信号频率是 CP 的六分频。

例：请分析图示电路的逻辑功能。



	Q_D	Q_C	Q_B	Q_A
置数	0	0	0	0
	0	0	1	0
	0	0	1	1
置数	0	1	0	0
	0	1	1	0
	0	1	1	1
置数	1	0	0	0
	1	0	1	0
	1	0	1	1
置数	1	1	0	0
	1	1	1	0
	1	1	1	1
	F			

功能： F 的信号频率是 CP 的12分频。

作业16:

7.7

7.8

7.9