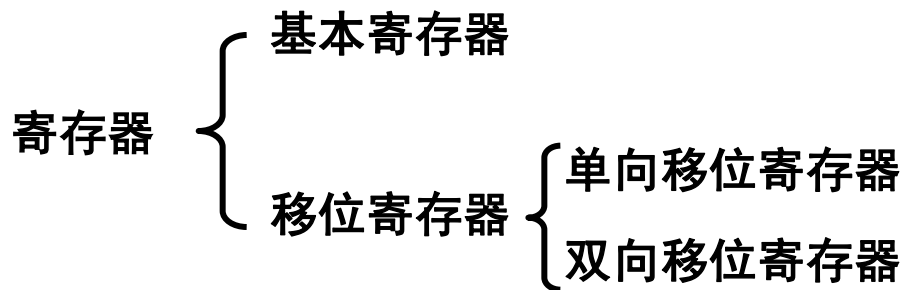


9.1 基本寄存器

寄存器——

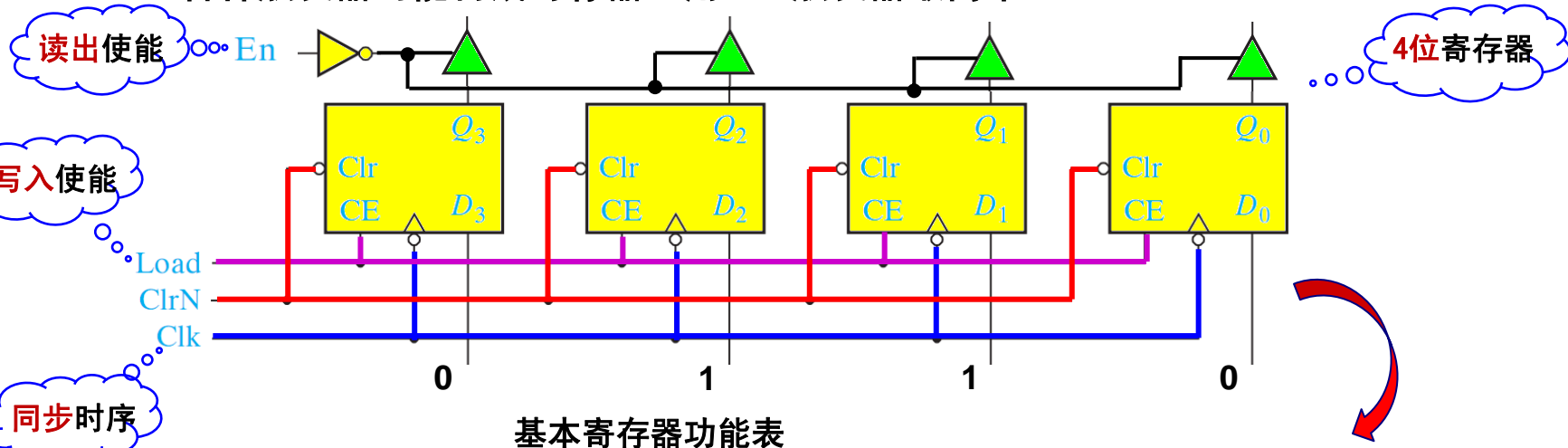
- 是计算机的一个重要部件，用于暂时存放一组二值代码（如参加运算的数据、运算结果、指令等）。
- 由触发器及控制门组成



- 基本寄存器的操作：读出/写入/复位（清零）
- 移位寄存器的操作：读出/写入/复位（清零）/左移（右移）

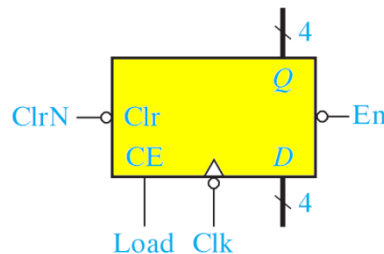
9.1 基本寄存器

- 一个 n 位寄存器由 n 个触发器构成，能存放 n 位二进制数。
- 各种触发器均能构成寄存器，用 D 触发器最简单。



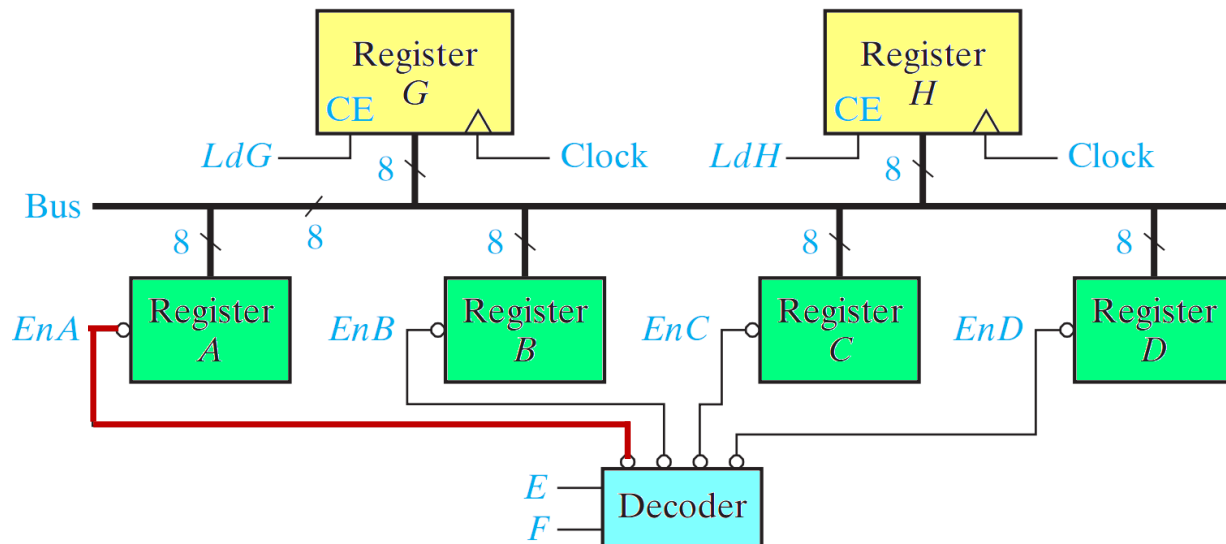
基本寄存器功能表

功能	条件	寄存器输出
异步清零	$\text{ClrN}=0$	$Q_3Q_2Q_1Q_0=0000$
保持	$\text{ClrN}=1$, 且 $\text{Load}=0$	$Q^{n+1}_3Q^{n+1}_2Q^{n+1}_1Q^{n+1}_0 = Q^n_3Q^n_2Q^n_1Q^n_0$
写入	$\text{ClrN}=1$, $\text{Load}=1$, $\text{clk} \downarrow$	$Q_3Q_2Q_1Q_0=D_3D_2D_1D_0$
读出	$\text{En}=0$	$Q_3Q_2Q_1Q_0=D_3D_2D_1D_0$



9.1 基本寄存器

■ 应用1——利用三态总线进行数据传送

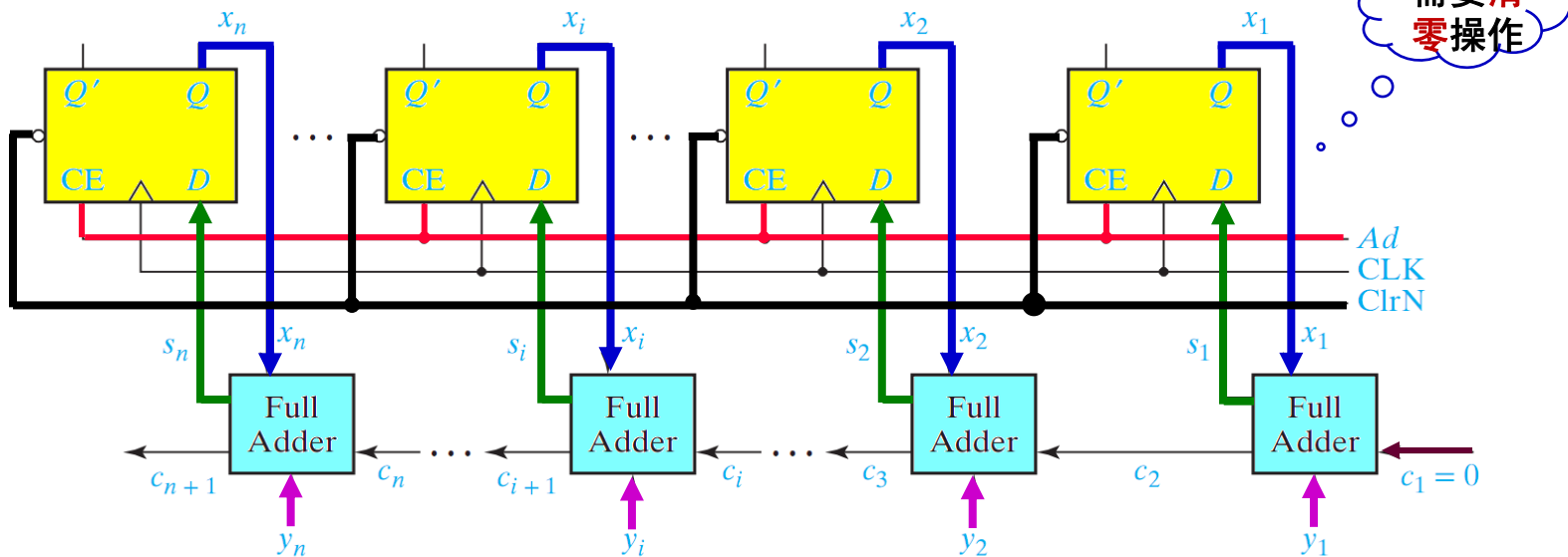


- Register A to G: $EF=00$, 且 $LdG=1, LdH=0$, $clk \uparrow$
- Register B to H: $EF=01$, 且 $LdG=0, LdH=1$, $clk \uparrow$

9.1 基本寄存器

■ 应用2——具有累加功能的并行加法器1

$$X=X+Y$$

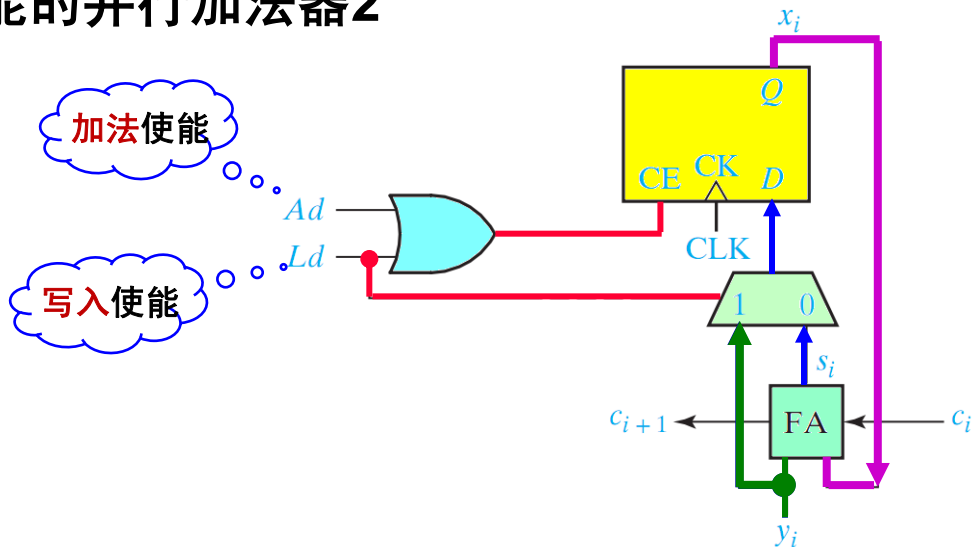


1. 初始化清零: $\text{ClrN}=0$, 则 $Q_n \dots Q_0=0$, 即 $X_n \dots X_0=0$
2. $\text{ClrN}=1$, 将 y_i 送到全加器输入端
3. 执行 $S_i = y_i + x_i + C_i$
4. 存储累加和: $\text{ClrN}=1, \text{Ad}=1, \text{CLK} \uparrow$ 到来时, 寄存器 $Q_i=S_i$

9.1 基本寄存器

■ 应用2——具有累加功能的并行加法器2

$$X = X + Y$$



■ 初始化:

$Ld=1$, 则 $CE=1$, 当 $ck \uparrow$ 到来时, $Q_i = y_i$ 即 $y_i \rightarrow x_i$, 将 x_i 送到全加器的另一个输入端

■ 送入第二个操作数 y_i , 执行 $S_i = y_i + x_i + c_i$

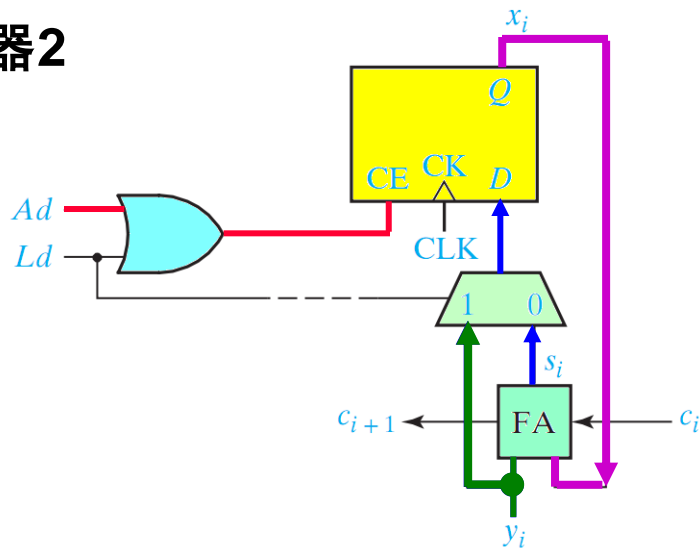
9.1 基本寄存器

■应用2——具有累加功能的并行加法器2

$$X=X+Y$$

与方案1比较:

触发器不需要初始清零，通过一个二选一数据选择器，在第一个时钟沿送入一个操作数，之后在每个时钟沿送入累加和



■ 初始化:

$Ld=1$, 则 $CE=1$, 当 $ck \uparrow$ 到来时, $Q_i=y_i$ 即 $y_i \rightarrow x_i$, 将 x_i 送到全加器的另一个输入端

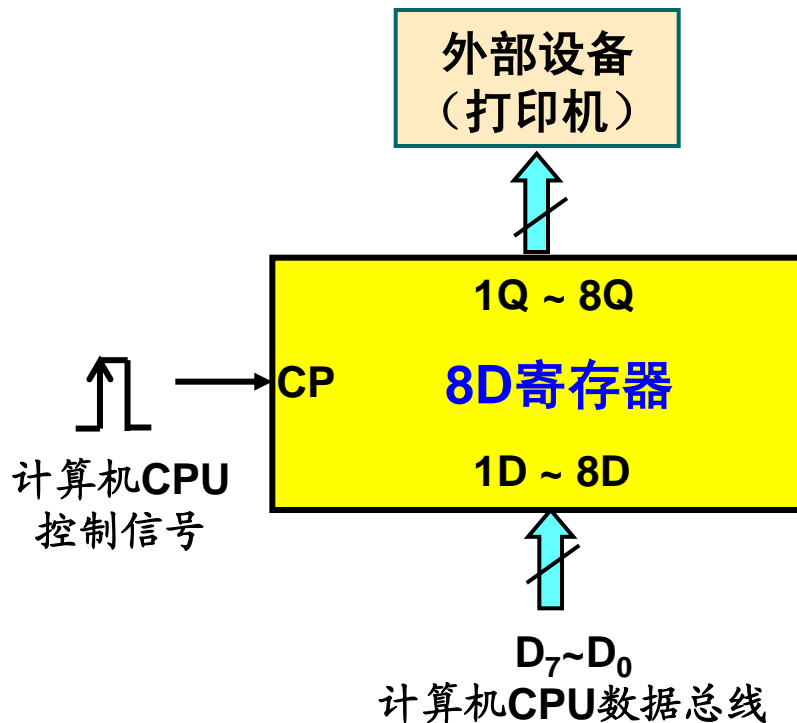
■ 送入第二个操作数 y_i , 执行 $S_i=y_i+x_i$

■ $Ld=0, Ad=1, ck \uparrow$ 到来时: $x_i = s_i$

■ 保持: $Ld=0, Ad=0$

9.1 基本寄存器

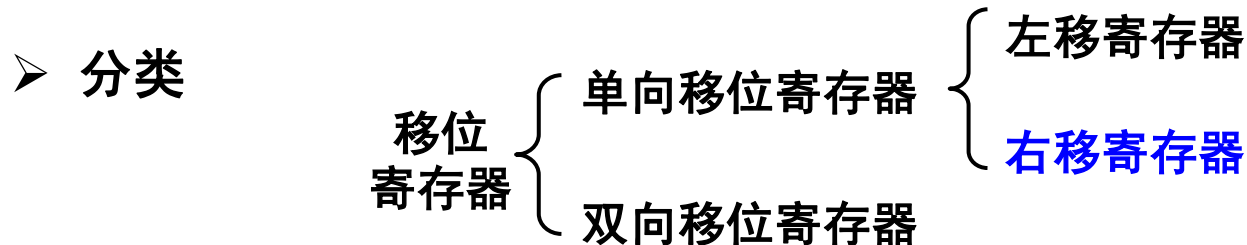
■ 应用3——计算机并行输入/输出接口



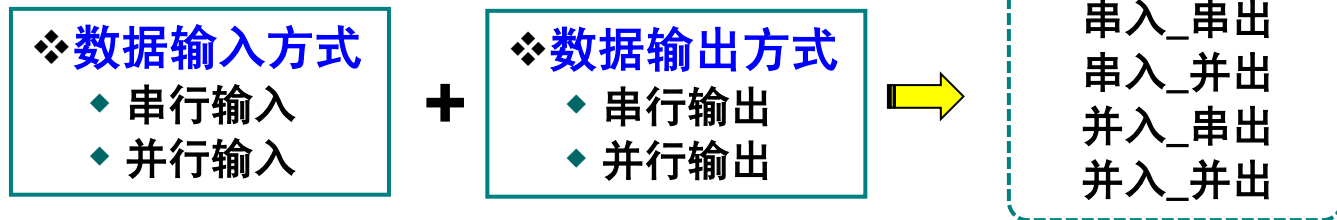
9.2 移位寄存器 (Shift Registers)

■ 移位寄存器——

- 每来一个时钟脉冲，寄存器里存储的数据，能依次向左移或右移1位。
- 可以实现代码的串、并行转换、数值运算和数据处理等。



➤ 工作方式



9.2 移位寄存器 (Shift Registers)

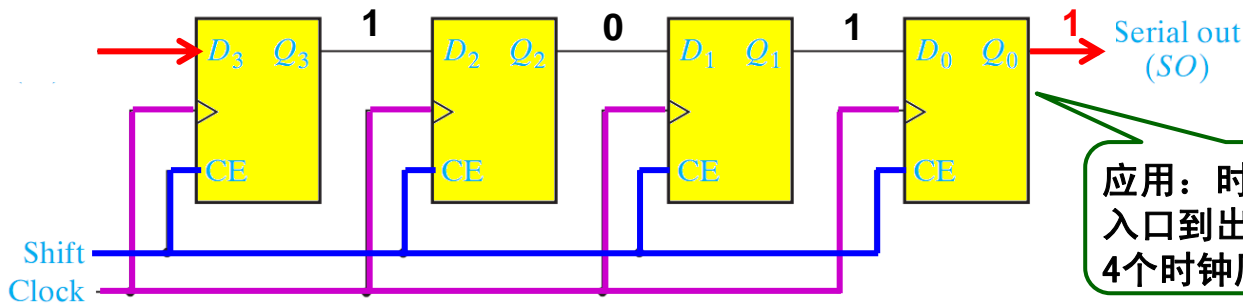
□ 右移寄存器 (Right-Shift Register)

(1). 串行输入/串行输出 (Serial in / Serial out)

■ 串行输出：移位路径上最后一个触发器的输出作为整个电路的输出

右移方式下：
数据从串行输入端送入，应该先送最低位

同步时序

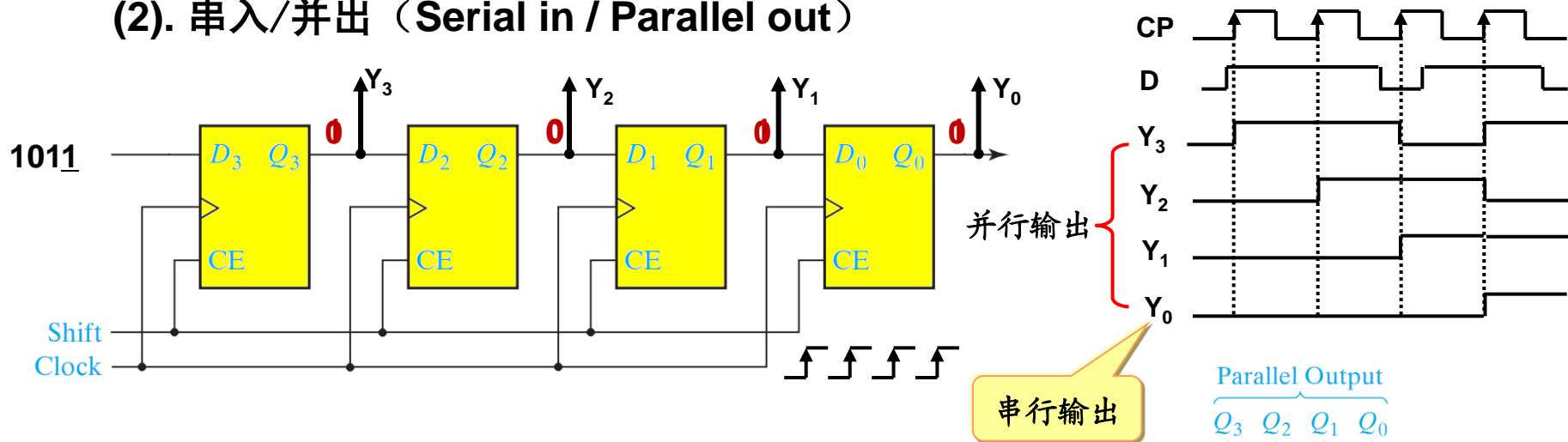


应用：时间延迟，
入口到出口，滞后
4个时钟周期

思考：左移寄存器如何设计？左移方式下从串行输入端送入数据，应该先送最低位还是最高位？

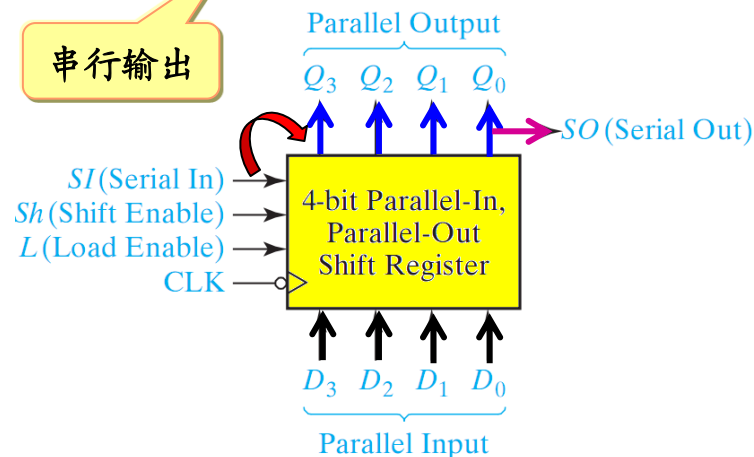
9.2 移位寄存器 (Shift Registers)

(2). 串入/并出 (Serial in / Parallel out)



(3). 并入/并出 (Parallel in / Parallel out)

(4). 并入/串出 (Parallel in / Serial out)



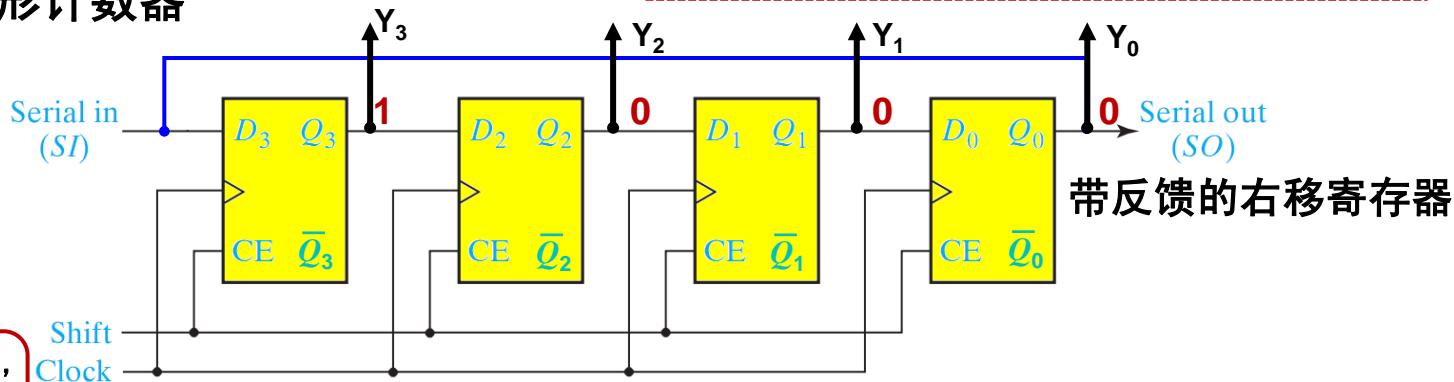
Inputs		Next State				Action
Sh (Shift)	L (Load)	Q ₃ ⁺	Q ₂ ⁺	Q ₁ ⁺	Q ₀ ⁺	
0	0	Q ₃	Q ₂	Q ₁	Q ₀	No change
0	1	D ₃	D ₂	D ₁	D ₀	Load
1	X	SI	Q ₃	Q ₂	Q ₁	Right shift

9.2 移位寄存器 (Shift Registers)

右移寄存器的应用——

计数器：一种能在输入信号作用下依次循环通过预定状态的时序逻辑电路。

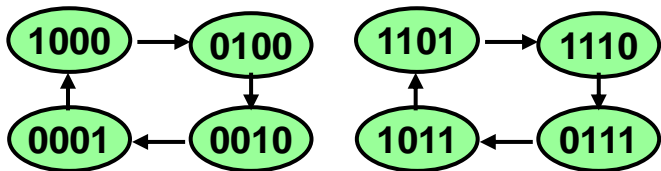
(1). 环形计数器



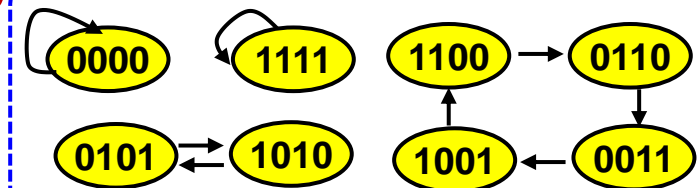
优点：电路简单，
输出具有二进制
译码器的特点

缺点： 2^n 个状态只
使用了 n 个；不能
自启动，需要预置

常用状态图



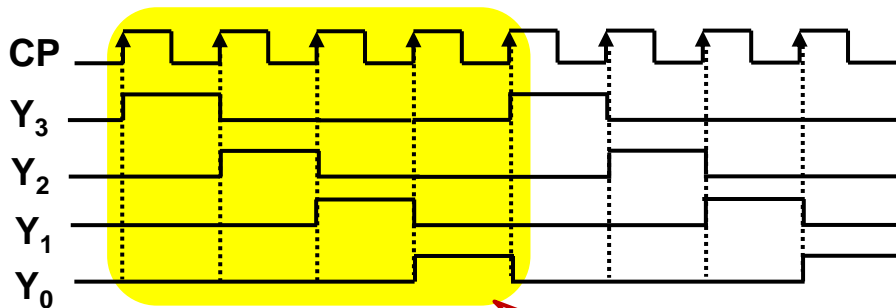
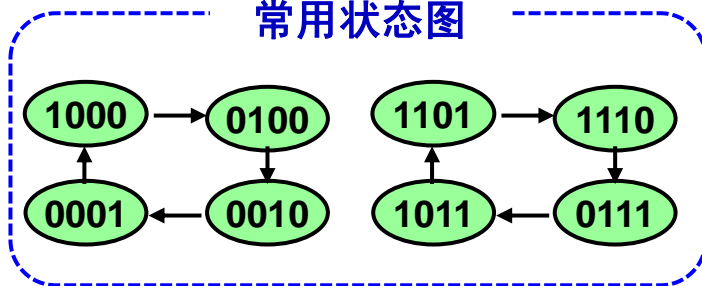
不常用状态图



9.2 移位寄存器 (Shift Registers)

优点：电路简单，
输出具有二进制
译码器的特点

常用状态图

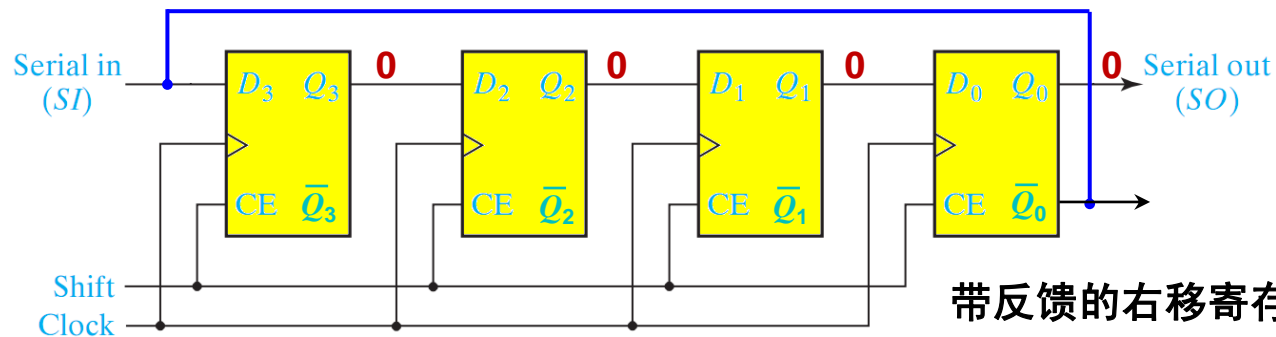


特点：一个周期
之内的波形与译
码器输出相同

9.2 移位寄存器 (Shift Registers)

右移寄存器的应用——

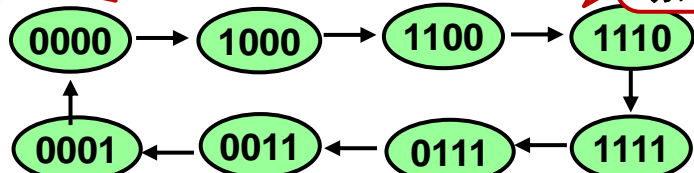
(2). 扭环形计数器



带反馈的右移寄存器

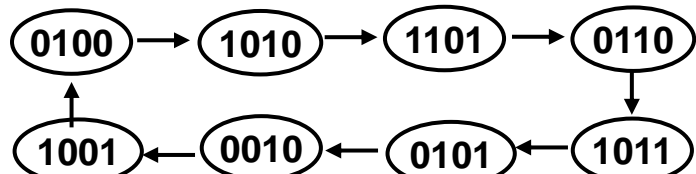
电路具有格雷
码输出的特点

常用状态图



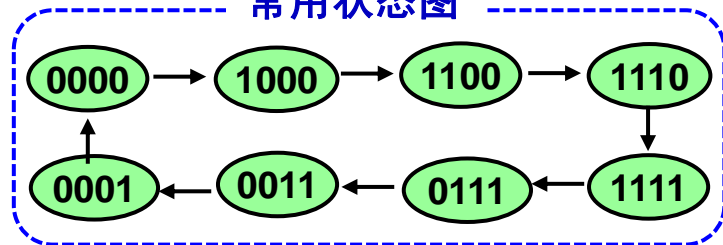
2^n 个状态使用了
 $2n$ 个; 不能自启
动, 需要预置

不常用状态图



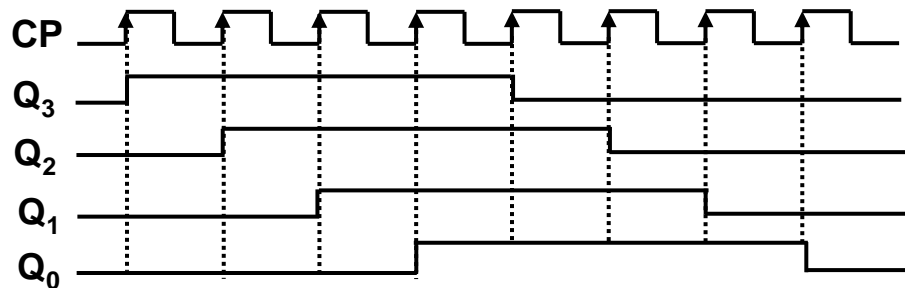
9.2 移位寄存器 (Shift Registers)

常用状态图



优点: ①无险象
②后级每个译码门
只需要2个输入端。
③模8计数器

输入				译码输出							
Q_3	Q_2	Q_1	Q_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0	0
1	1	1	0	0	0	0	1	0	0	0	0
1	1	1	1	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	1



$Q_3 \backslash Q_2$	$Q_1 Q_0$ 00	01	11	10
00	1	0	0	X
01	X	X	0	X
11	0	X	0	0
10	0	X	X	X

$$Y_0 = \bar{Q}_3 \bar{Q}_0$$

$Q_3 \backslash Q_2$	$Q_1 Q_0$ 00	01	11	10
00	0	0	0	X
01	X	X	0	X
11	0	X	0	0
10	1	X	X	X

$$Y_1 = Q_3 \bar{Q}_2$$

9.2 移位寄存器（Shift Registers）

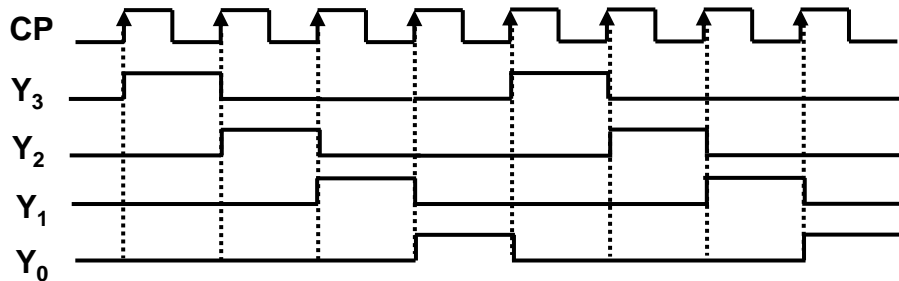
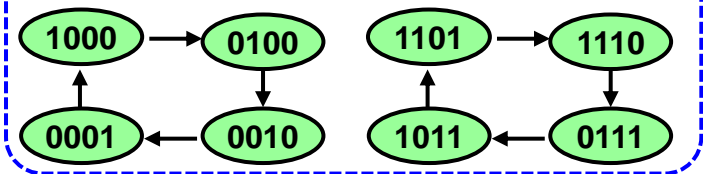
□ 环形、扭环形计数器总结——

特点：在移位寄存器的基础上，增加反馈逻辑电路组成。

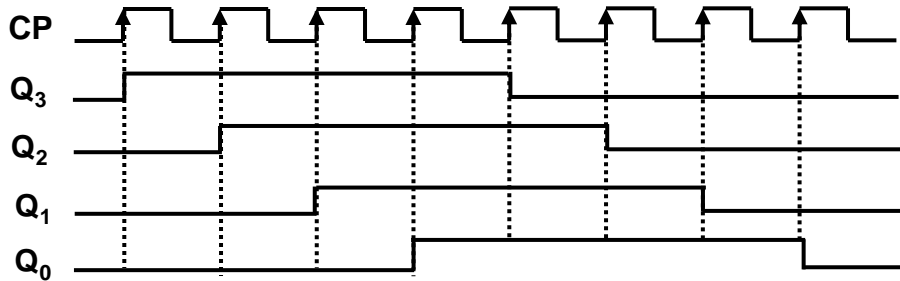
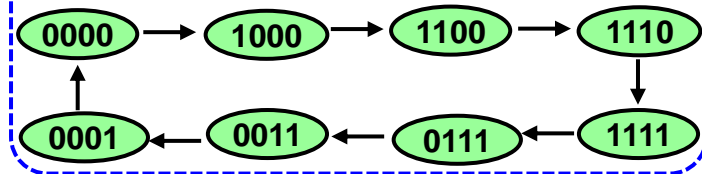
用途：

- 构成**特殊编码**的计数器（非二进制计数器）
- 环形计数器和扭环形计数器在计算机中可用于组成时序信号发生器（节拍发生器）

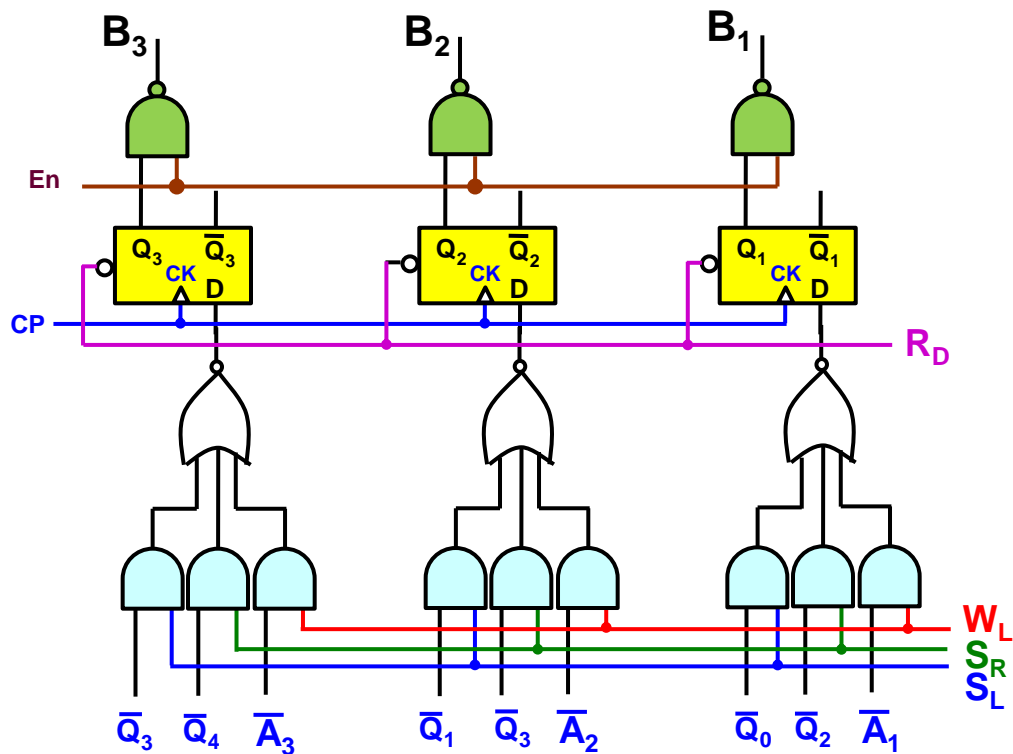
环形计数器常用状态图



扭环形计数器常用状态图



9.3 双向移位寄存器



R_d —— 异步清零;
 S_R —— 右移使能;
 En —— 输出使能

W_L —— 写入使能;
 S_L —— 左移使能

输入方程

$$\begin{cases} D_3 = \overline{A_3} W_L + \overline{Q_4} S_R + \overline{Q_2} S_L \\ D_2 = \overline{A_2} W_L + \overline{Q_3} S_R + \overline{Q_1} S_L \\ D_1 = \overline{A_1} W_L + \overline{Q_2} S_R + \overline{Q_0} S_L \end{cases}$$

输出方程

$$\begin{cases} B_3 = \overline{Q_3} E_n \\ B_2 = \overline{Q_2} E_n \\ B_1 = \overline{Q_1} E_n \end{cases}$$

次态方程

$$\begin{cases} Q_3^{n+1} = D_3 \\ Q_2^{n+1} = D_2 \\ Q_1^{n+1} = D_1 \end{cases}$$

输入方程

$$\begin{cases} D_3 = \overline{A_3} W_L + \overline{Q_4} S_R + \overline{Q_2} S_L = \overline{A_3 \cdot 0 + Q_4 \cdot 1 + Q_2 \cdot 0} = Q_4 \\ D_2 = \overline{A_2} W_L + \overline{Q_3} S_R + \overline{Q_1} S_L = \overline{A_2 \cdot 0 + Q_3 \cdot 1 + Q_1 \cdot 0} = Q_3 \\ D_1 = \overline{A_1} W_L + \overline{Q_2} S_R + \overline{Q_0} S_L = \overline{A_1 \cdot 0 + Q_2 \cdot 1 + Q_0 \cdot 0} = Q_2 \end{cases}$$



Let: $S_L = 1$, $W_L = S_R = 0$

次态方程 $\begin{cases} Q_3^{n+1} = D_3 = Q_2 \\ Q_2^{n+1} = D_2 = Q_1 \\ Q_1^{n+1} = D_1 = Q_0 \end{cases}$

输入方程

$$\begin{cases} D_3 = \overline{A_3} W_L + \overline{Q_4} S_R + \overline{Q_2} S_L = \overline{A_3 \cdot 0 + Q_4 \cdot 0 + Q_2 \cdot 1} = Q_2 \\ D_2 = \overline{A_2} W_L + \overline{Q_3} S_R + \overline{Q_1} S_L = \overline{A_2 \cdot 0 + Q_3 \cdot 0 + Q_1 \cdot 1} = Q_1 \\ D_1 = \overline{A_1} W_L + \overline{Q_2} S_R + \overline{Q_0} S_L = \overline{A_1 \cdot 0 + Q_2 \cdot 0 + Q_0 \cdot 1} = Q_0 \end{cases}$$

9.3 双向移位寄存器

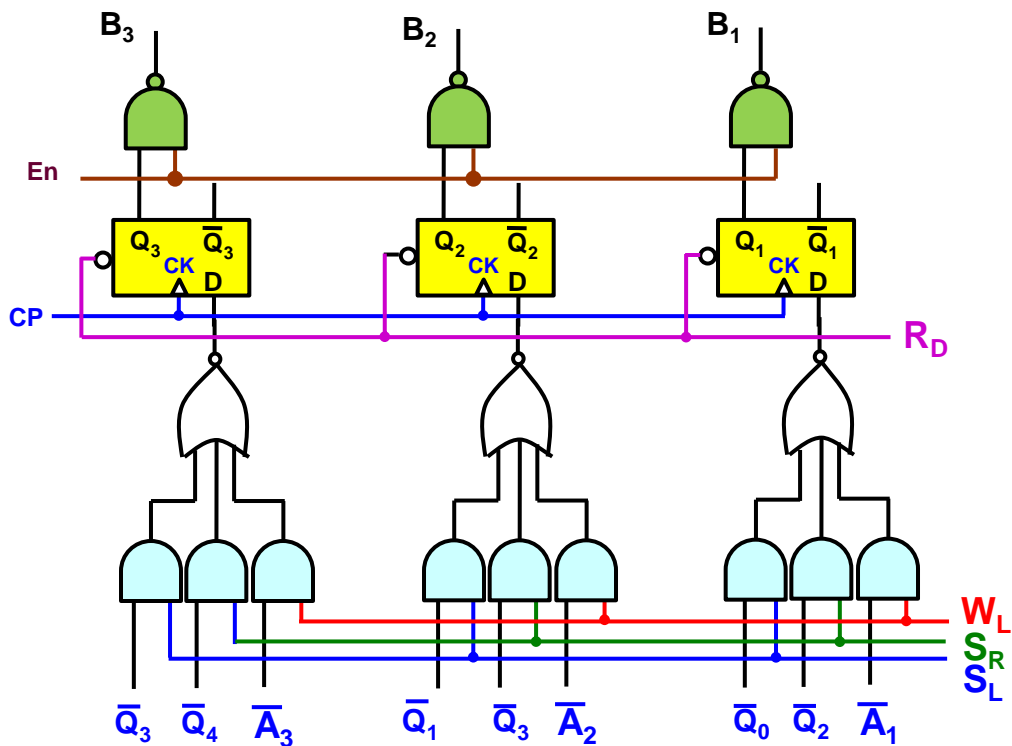
□ 功能——

(4) 读出

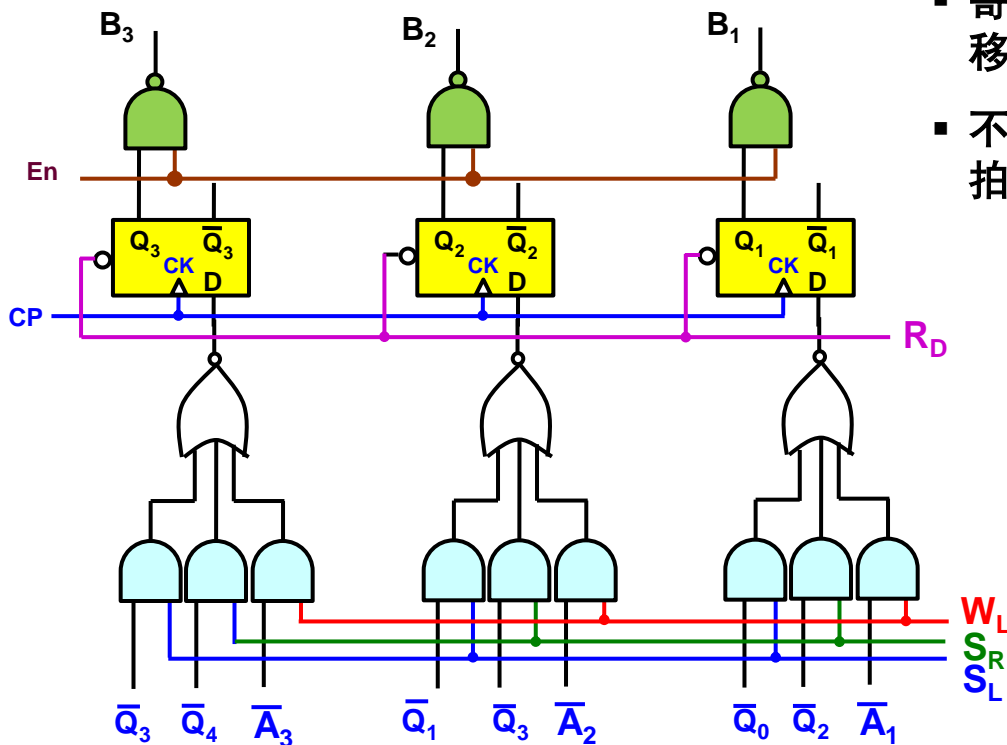
Let: $E_n = 1$

输出方程

$$\begin{cases} B_3 = \overline{Q_3} E_n = \overline{Q_3} \\ B_2 = \overline{Q_2} E_n = \overline{Q_2} \\ B_1 = \overline{Q_1} E_n = \overline{Q_1} \end{cases}$$



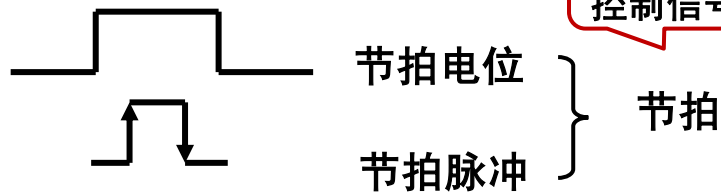
9.3 双向移位寄存器



- 寄存器的每一个操作（写入、读出、左移、右移）都是在**节拍**的控制下完成的。
- 不改变触发器状态的操作（读出），只需要节拍电位。

必须保证节拍脉冲的
边沿被节拍电位的有效电平**完全覆盖**

节拍：一种
控制信号



例如：

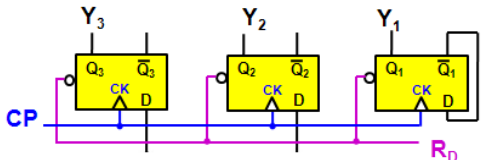
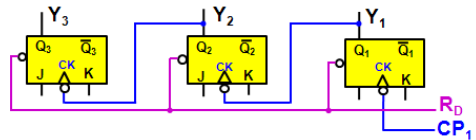
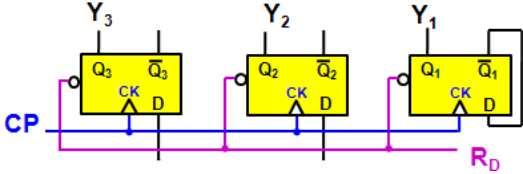
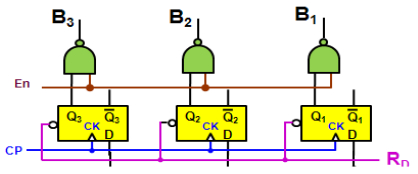
- 写入操作，需要 $W_L = 1$ ，同时 $CP \uparrow$
- 左移操作，需要 $S_L = 1$ ，同时 $CP \uparrow$
- 读出操作，只需要 $En=1$

9.3 双向移位寄存器

寄存器总结

- ❑ 主要功能：存放二进制数据（存储的二进制位数由里面触发器的数量决定）
- ❑ 寄存器操作：写入、读出、保持、清零。
- ❑ 移位寄存器还可以：将数据依次左移或右移1位
- ❑ 特点：寄存器的每一个操作（写入、读出、左移、右移）都是在**节拍**的控制下完成的

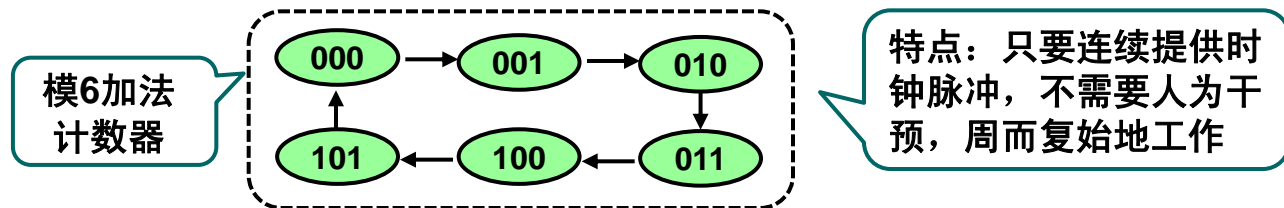
时序逻辑电路的分类

分类方式		种类	特点	电路框图示例
时序逻辑电路	按照 时钟 信号的连接方式	同步时序——	<ul style="list-style-type: none"> 所有的时钟端连接在一起，状态的改变同时发生（数字系统中用到的最多） 	
		异步时序——	<ul style="list-style-type: none"> 没有统一的时钟脉冲同步，状态的改变有先有后，不同时发生 容易产生毛刺（有不利影响） 	
	按照电路 输出 与 输入 及电路 状态 的关系	摩尔型电路 (Moore)	<ul style="list-style-type: none"> 电路的输出仅与现态有关，与电路的输入无关；或者直接以电路状态作为输出。 	
		米里型电路 (Mealy)	<ul style="list-style-type: none"> 电路输出与电路的现态及电路的输入均有关； 	

几种典型的时序逻辑部件——计数器

计数器？

一种能在输入信号作用下依次通过预定状态的时序逻辑电路，是数字系统和计算机广泛使用的逻辑器件，可用于计数、分频、定时、控制、产生节拍脉冲（顺序脉冲）和序列脉冲等。



- 由一组触发器构成，计数器中的“数”是用触发器的状态组合来表示的。
- 计数器在运行时，所经历的状态是周期性的，总是在有限个状态中循环。
- 将一次循环所包含的**状态总数**称为计数器的“**模**”，记为 N ，包含 n 个触发器的最大模值 $N = 2^n$ 。
- 把作用于计数器的时钟脉冲称为计数脉冲，用 CP (或 CLK) 表示。

几种典型的时序逻辑部件——计数器

□ 计数器的种类

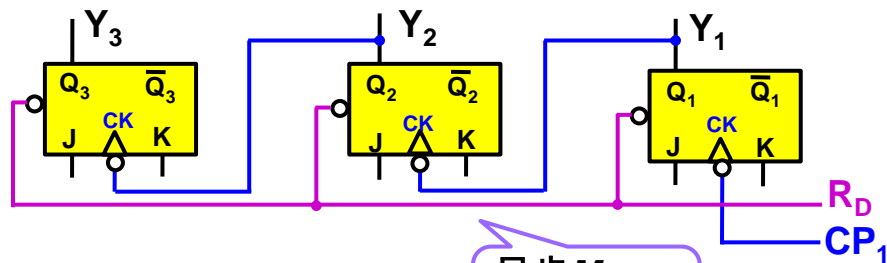
- (1) 按时钟方式分为：同步计数器和异步计数器；
- (2) 按功能分为：加法计数器、减法计数器和可逆计数器等。
- (3) 按计数方式分为：二进制计数器，十进制计数器，M进制计数器

时序逻辑电路的分析方法

确定系统变量（输入变量、输出变量、状态变量）

- ① 列驱动方程（控制函数）
- ② 列输出方程（输出函数）
- ③ 列状态方程（次态方程）
- ④ 列写状态转换表
- ⑤ 画出状态图
- ⑥ 画出波形图（如必要）

例1——异步计数器



① 输入方程

$$\begin{aligned} J_1 &= K_1 = 1 & CP_1 &\downarrow \\ J_2 &= K_2 = 1 & CP_2 &= Y_1 \downarrow \\ J_3 &= K_3 = 1 & CP_3 &= Y_2 \downarrow \end{aligned}$$

异步Moor
型时序电路

异步模8加
法计数器

② 次态方程

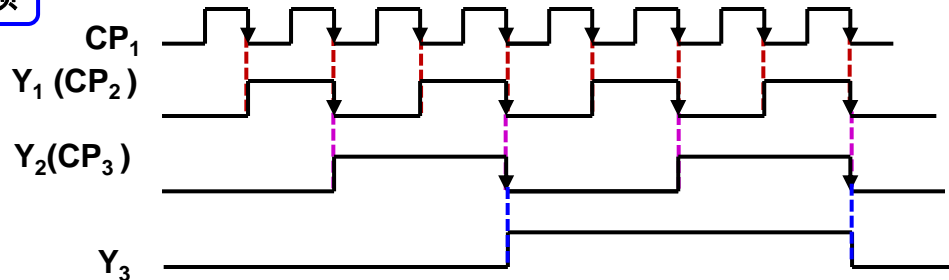
$$\begin{aligned} Y_1^{n+1} &= J_1 \bar{Q}_1 + \bar{K}_1 Q_1 = \bar{Y}_1 & CP_1 &\downarrow \\ Y_2^{n+1} &= J_2 \bar{Q}_2 + \bar{K}_2 Q_2 = \bar{Y}_2 & Y_1 &\downarrow \\ Y_3^{n+1} &= J_3 \bar{Q}_3 + \bar{K}_3 Q_3 = \bar{Y}_3 & Y_2 &\downarrow \end{aligned}$$

③ 状态转换表

现态			次态			时钟		
Y_3^n	Y_2^n	Y_1^n	Y_3^{n+1}	Y_2^{n+1}	Y_1^{n+1}	CP_3	CP_2	CP_1
0	0	0	0	0	1	无	无	↓
0	0	1	0	1	0	无	↓	↓
0	1	0	0	1	1	无	无	↓
0	1	1	1	0	0	↓	↓	↓
1	0	0	1	0	1	无	无	↓
1	0	1	1	1	0	无	↓	↓
1	1	0	1	1	1	无	无	↓
1	1	1	0	0	0	↓	↓	↓

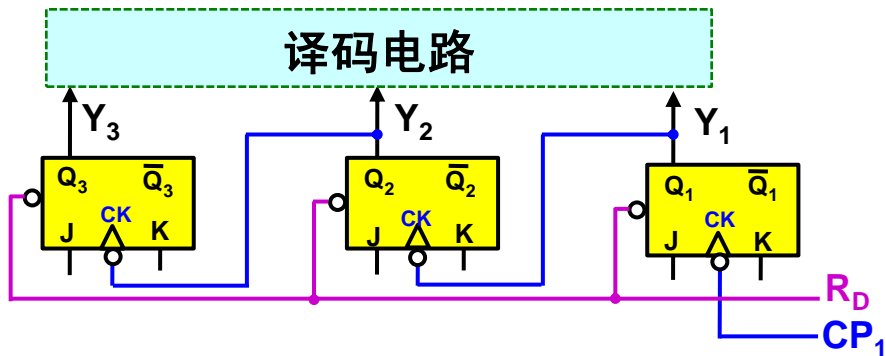
④ 波形图

二分频

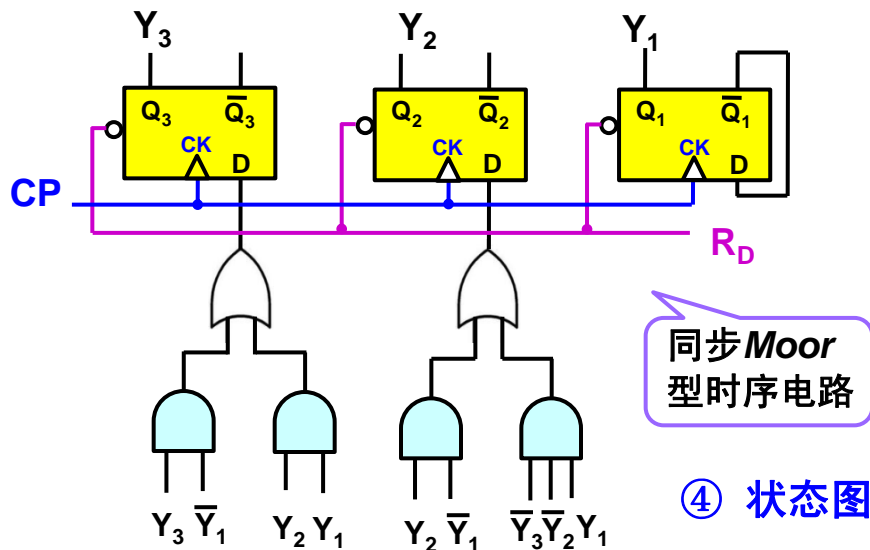


异步计数器总结

- 外接时钟源只作用于最低位触发器，高位触发器的时钟信号通常由低位触发器的输出提供，高位触发器的翻转有待低位触发器翻转后才能进行。
- 每一级触发器都存在传输延迟，位数越多计数器工作速度越慢，在大型数字设备中较少采用。
- 对计数器状态进行译码时，由于触发器不同步，译码器输出会出现尖峰脉冲（位数越多，尖峰信号越宽），使仪器设备产生误动作。
- 优点：结构比较简单，所用元件较少。



例2——同步计数器



① 输入方程

$$D_3 = Y_3 \bar{Y}_1 + Y_2 Y_1$$

$$D_2 = Y_2 \bar{Y}_1 + \bar{Y}_3 \bar{Y}_2 Y_1$$

$$D_1 = \bar{Y}_1$$

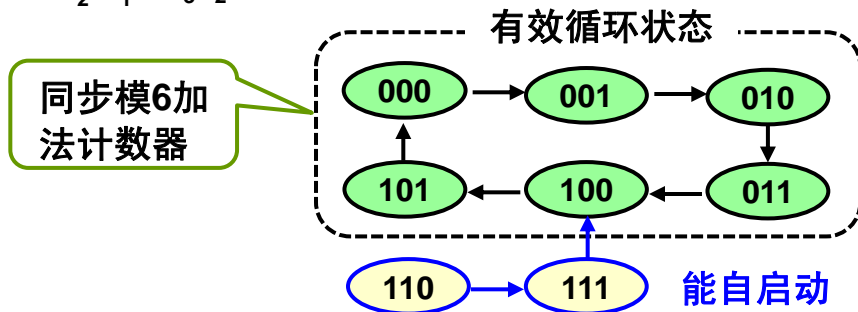
② 次态方程

$$\begin{cases} Y_1^{n+1} = D_1 \\ Y_2^{n+1} = D_2 \\ Y_3^{n+1} = D_3 \end{cases}$$

③ 状态转换表

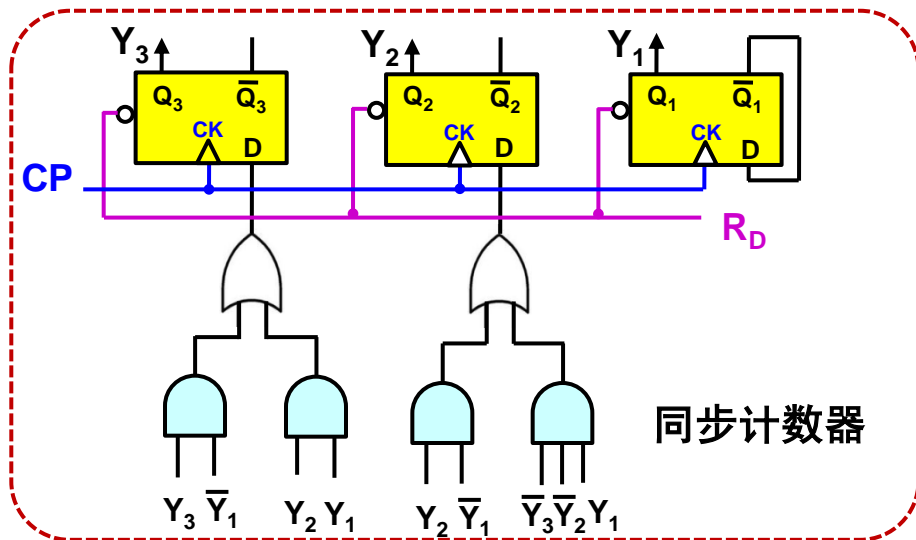
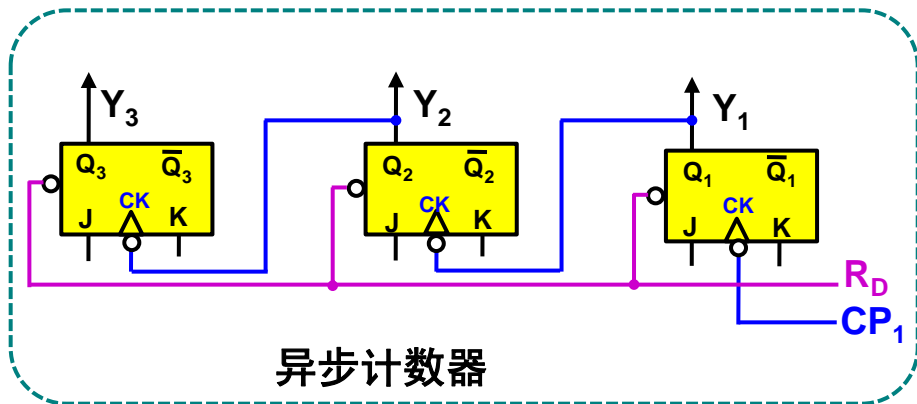
现态			次态			时钟
Y_3^n	Y_2^n	Y_1^n	Y_3^{n+1}	Y_2^{n+1}	Y_1^{n+1}	CP
0	0	0	0	0	1	↑
0	0	1	0	1	0	↑
0	1	0	0	1	1	↑
0	1	1	1	0	0	↑
1	0	0	1	0	1	↑
1	0	1	0	0	0	↑
1	1	0	1	1	1	↑
1	1	1	1	0	0	↑

④ 状态图



同步计数器总结

- ❑ 所有触发器的时钟端并联在一起，受控于同一个外接时钟源
- ❑ 所有触发器同时翻转，不存在时钟到各触发器输出的传输延迟的积累；
- ❑ 同步计数器的工作频率只与一个触发器的时钟到输出的传输延迟有关，所以它的工作频率比异步计数器高；
- ❑ 由于各触发器同时翻转，因此，同步计数器的输出不会产生毛刺；
- ❑ 缺点：结构比较复杂（各触发器的输入由多个Q输出的组合逻辑得到），所用元件较多。



几种典型的时序逻辑部件——节拍发生器1

□ 节拍发生器（顺序脉冲发生器）——

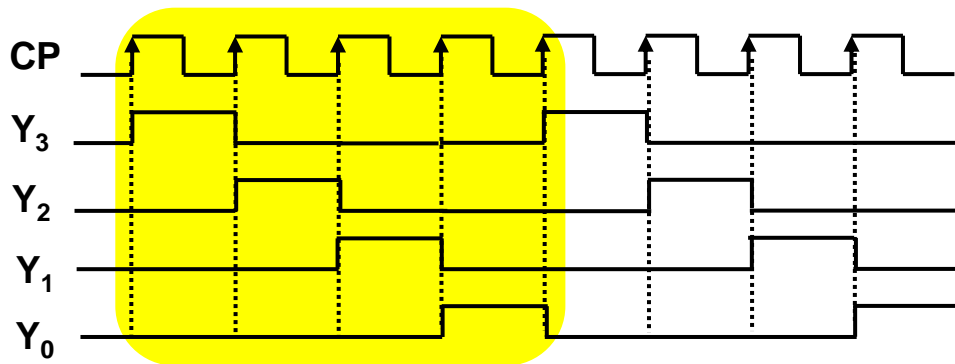
定义

每个循环周期内，在时钟脉冲的作用下，产生一组在时间上有一定**先后顺序**的脉冲信号

作用

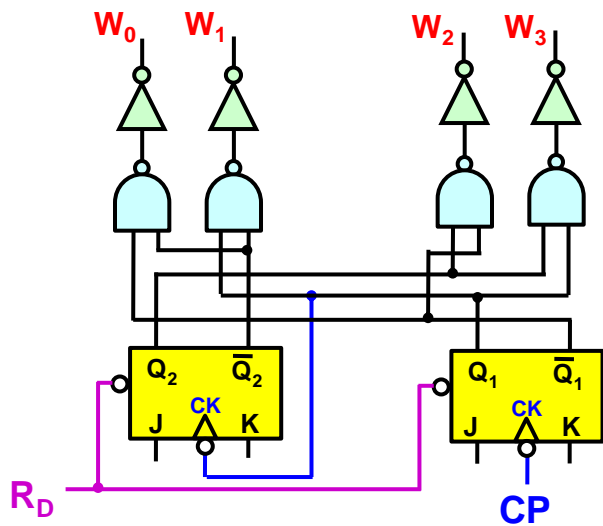
数字系统和计算机的控制部件利用顺序脉冲，形成所需要的各种控制信号，使某些设备按照事先规定的顺序进行运算或操作

例：将4位二进制数（如1000）存入某寄存器，然后将数据右移1位，之后将数据读走，再将右移后的数据左移1位。以上操作可以自动循环进行。



- ①执行写入操作：写入使能有效（存入1000）
- ②执行右移操作：右移使能有效（右移后0100）
- ③执行读出操作：读出使能有效
- ④执行左移操作：左移使能有效（左移后1000）

几种典型的时序逻辑部件——节拍发生器1



① 输入方程

$$J_1 = K_1 = 1, CP_1 \downarrow$$

$$J_2 = K_2 = 1, CP_2 = Q_1 \downarrow$$

② 次态方程

$$Q_1^{n+1} = J_1 \bar{Q}_1 + K_1 \bar{Q}_1 = \bar{Q}_1$$

$$Q_2^{n+1} = J_2 \bar{Q}_2 + K_2 \bar{Q}_2 = \bar{Q}_2$$

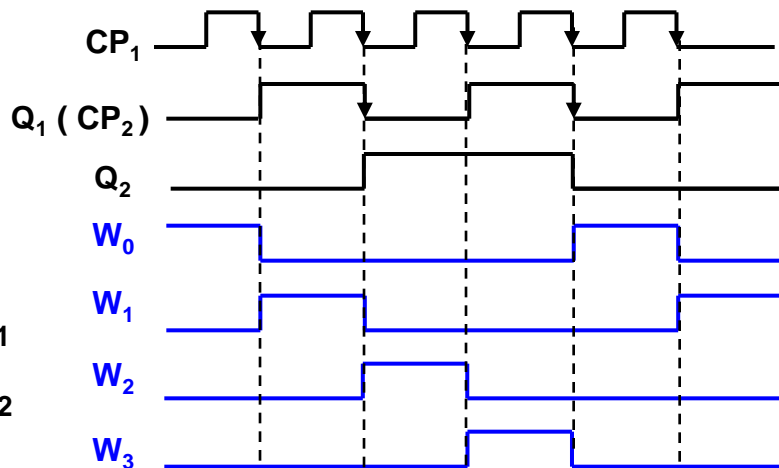
③ 输出方程

$$\begin{cases} W_0 = \bar{Q}_2 \bar{Q}_1 \\ W_1 = \bar{Q}_2 Q_1 \\ W_2 = Q_2 \bar{Q}_1 \\ W_3 = Q_2 Q_1 \end{cases}$$

结论：4-节拍发生器 ($W_0 \sim W_3$)

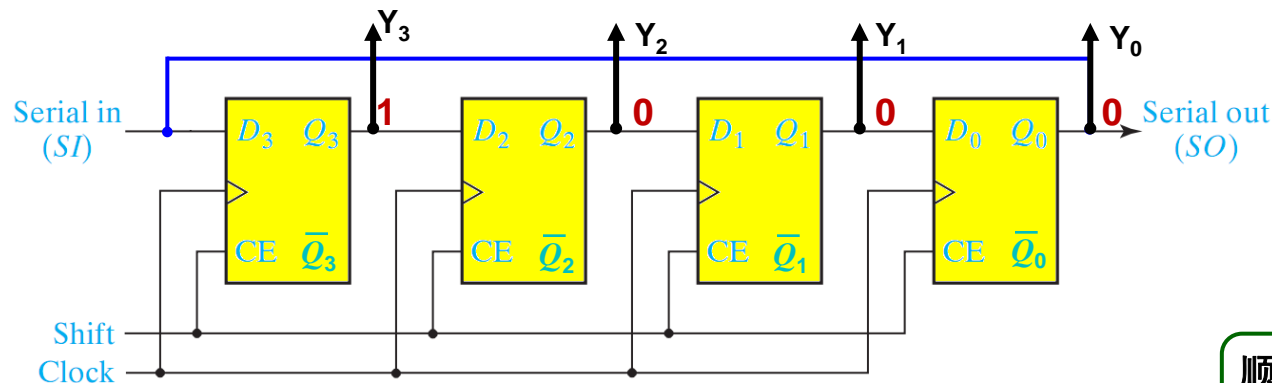
④ 状态转换表

现态		次态		时钟	
Q_2^n	Q_1^n	Q_2^{n+1}	Q_1^{n+1}	CP_2	CP_1
0	0	0	1	无	↓
0	1	1	0	↓	↓
1	0	1	1	无	↓
1	1	0	0	↓	↓

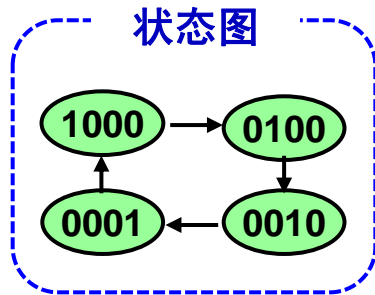


几种典型的时序逻辑部件——节拍发生器1

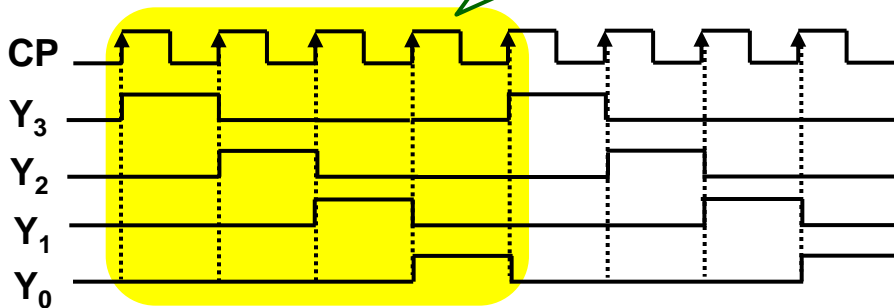
回顾：环形计数器



状态图

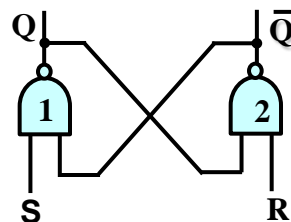
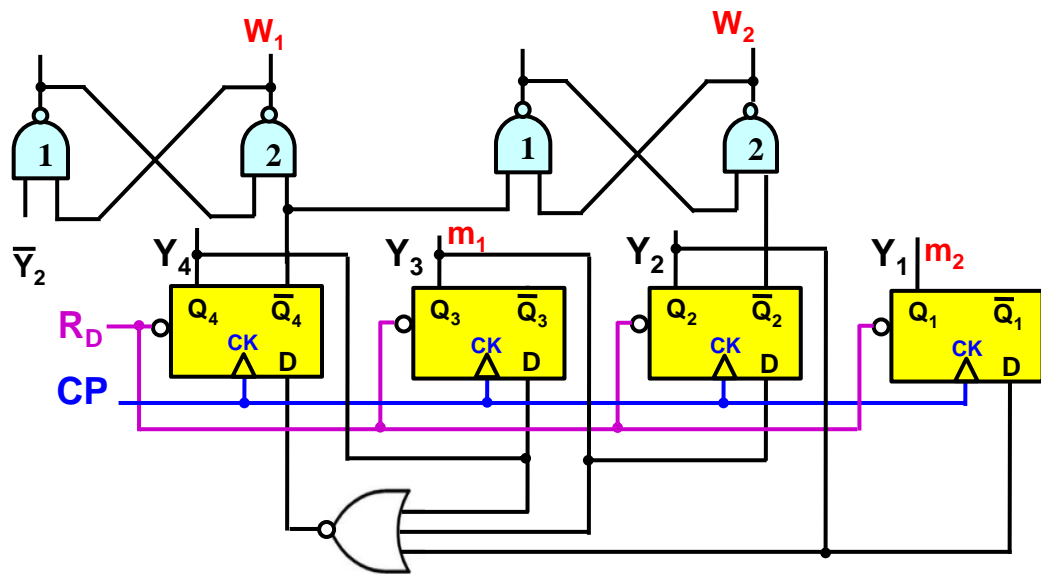


顺序脉冲发生器的波形

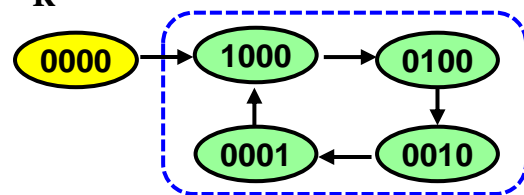


- 应用：电话响铃控制
- 用4位顺序脉冲发生器的某一个输出作为响铃控制信号，若时钟CP周期为1秒，电话铃声就是响1秒停3秒的节奏。

几种典型的时序逻辑部件——节拍发生器2



④ 状态图



③ 状态转换表

现态				次态				时钟
Y_4^n	Y_3^n	Y_2^n	Y_1^n	Y_4^{n+1}	Y_3^{n+1}	Y_2^{n+1}	Y_1^{n+1}	CP
0	0	0	0	1	0	0	0	↑
1	0	0	0	0	1	0	0	↑
0	1	0	0	0	0	1	0	↑
0	0	1	0	0	0	0	1	↑
0	0	0	1	1	0	0	0	↑

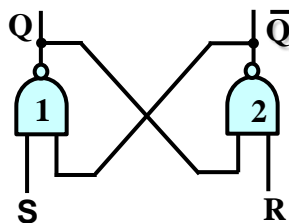
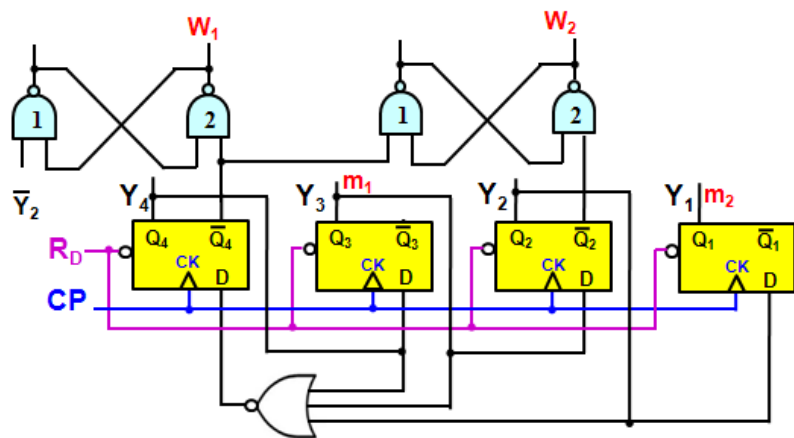
① 输入方程

$$\begin{cases} D_1 = Y_2 \\ D_2 = Y_3 \\ D_3 = Y_4 \\ D_4 = \overline{Y_4 + Y_3 + Y_2} \end{cases}$$

② 次态方程

$$\begin{cases} Y_1^{n+1} = Y_2 \\ Y_2^{n+1} = Y_3 \\ Y_3^{n+1} = Y_4 \\ Y_4^{n+1} = \overline{Y_4 + Y_3 + Y_2} \end{cases}$$

几种典型的时序逻辑部件——节拍发生器2



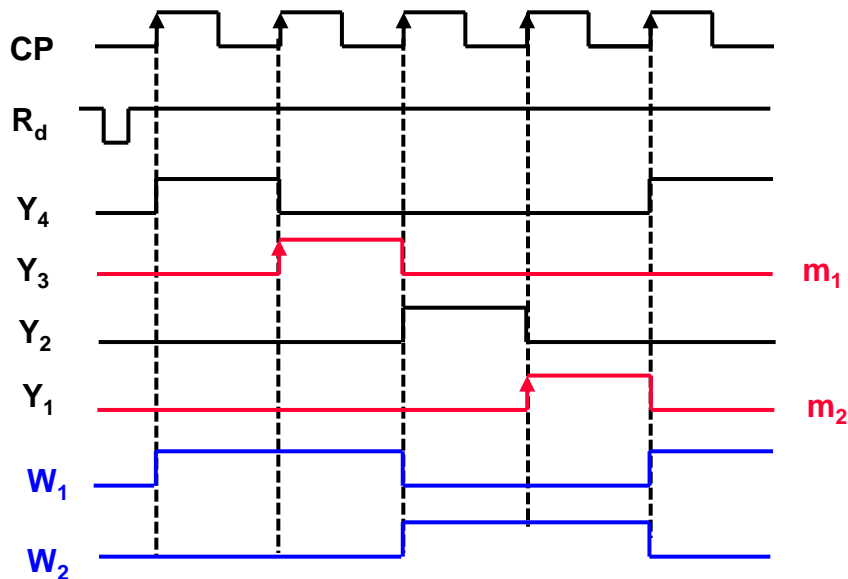
结论：2-节拍发生器

- $W_1_m_1$: 节拍电位_节拍脉冲
- $W_2_m_2$: 节拍电位_节拍脉冲

⑤ 确定输出

R	S	Q_{n+1}	\bar{Q}_{n+1}
\bar{Y}_4	\bar{Y}_2	$(W_1 = \bar{Q})$	
1	1	Q_n	\bar{Q}_n
0	1	0	1
1	0	1	0
0	0	—	—

R	S	Q_{n+1}	\bar{Q}_{n+1}
\bar{Y}_2	\bar{Y}_4	$(W_2 = \bar{Q})$	
1	1	Q_n	\bar{Q}_n
0	1	0	1
1	0	1	0
0	0	—	—



时序逻辑电路分析

时序逻辑电路的分析方法

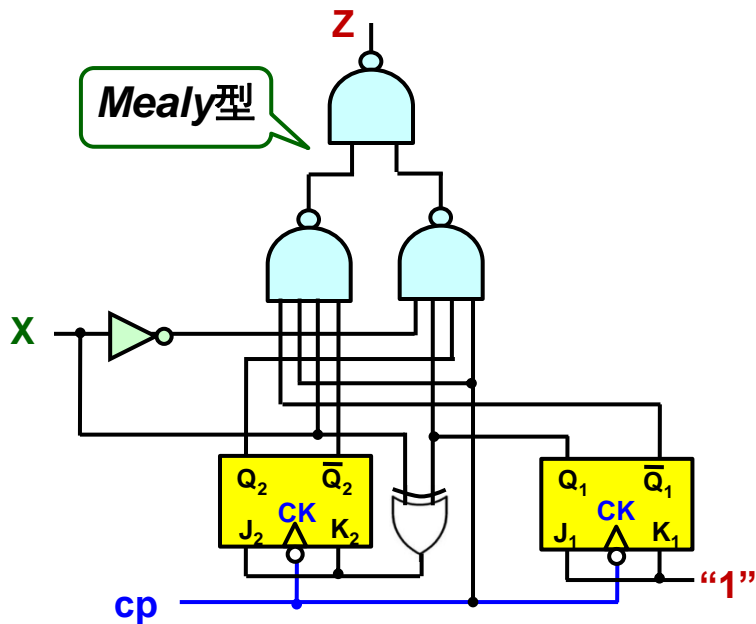
确定系统变量（输入变量、输出变量、状态变量）

- ① 列驱动方程（控制函数）
- ② 列输出方程（输出函数）
- ③ 列状态方程（次态方程）
- ④ 列写状态转换表
- ⑤ 画出状态图
- ⑥ 画出波形图（如必要）



- 同步时序电路
- 异步时序电路

时序逻辑电路分析——示例1:同步时序



① 输入方程

$$J_1 = K_1 = 1$$

$$J_2 = K_2 = X \oplus Q_1^n$$

② 次态方程

$$Q_2^{n+1} = X \oplus Q_1^n \oplus Q_2^n$$

$$Q_1^{n+1} = \overline{Q_1^n}$$

③ 输出方程

$$Z = \overline{XCPQ_2^nQ_1^n} \cdot \overline{XCPQ_2^nQ_1^n}$$

$$= XCPQ_2^nQ_1^n + \overline{XCPQ_2^nQ_1^n}$$

次态方程:

$$JK: Q^{n+1} = J\overline{Q}^n + \overline{K}Q^n$$

$$Q_2^{n+1} = J_2\overline{Q_2^n} + \overline{K_2}Q_2^n$$

$$= (X \oplus Q_1^n)\overline{Q_2^n} + \overline{(X \oplus Q_1^n)}Q_2^n$$

$$= X \oplus Q_1^n \oplus Q_2^n$$

$$Q_1^{n+1} = J_1\overline{Q_1^n} + \overline{K_1}Q_1^n$$

$$= \overline{Q_1^n}$$

④ 状态转换表

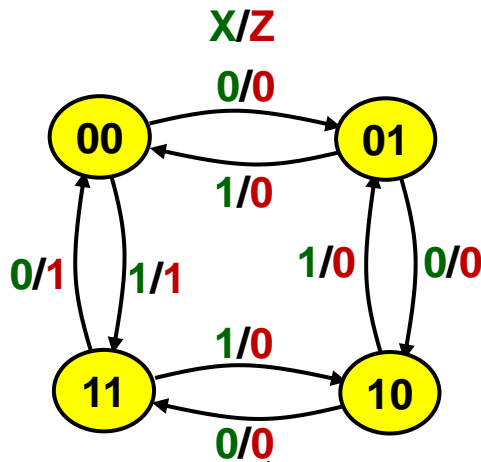
输入	现态		次态		输出
X	Q_2^n	Q_1^n	Q_2^{n+1}	Q_1^{n+1}	Z
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	0	1	0
1	1	1	1	0	0

时序逻辑电路分析——示例1:同步时序

④ 状态转换表

现态 $Q_2^n Q_1^n$	$Q_2^{n+1} Q_1^{n+1} / Z$	
	$X=0$	$X=1$
0 0	0 1 / 0	1 1 / 1
0 1	1 0 / 0	0 0 / 0
1 0	1 1 / 0	0 1 / 0
1 1	0 0 / 1	1 0 / 0

⑤ 状态图



Mealy型: 输出
值画在状态图
转换线的旁边

结论: 模4可逆计数器

- $X=0$: 加计数
- $X=1$: 减计数

Z : 进位和借位输出标志

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	52
--	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	----



② 次态方程

$$Y_4^{n+1} = Y_3^n$$

$$Y_3^{n+1} = Y_2^n$$

$$\mathbf{Y}_2^{n+1} = \mathbf{Y}_1^n$$

$$Y_1^{n+1} = Y_1^n \overline{Y_4^n} + \overline{Y_3^n} \overline{Y_4^n}$$

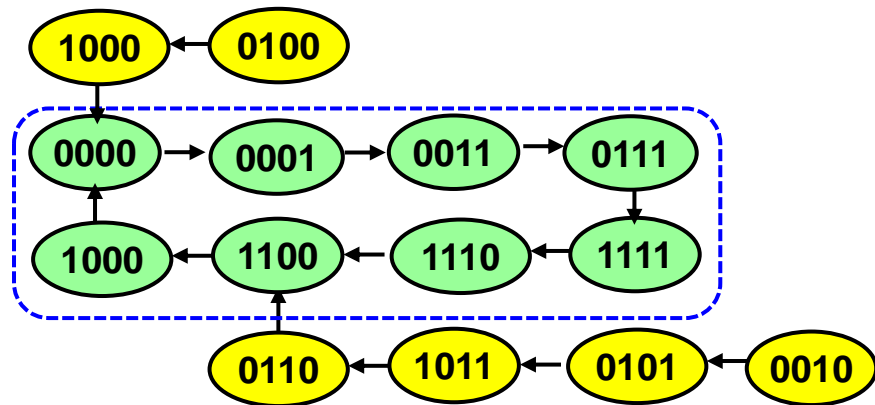
[illegible][illegible]

时序逻辑电路分析——示例2:同步时序

③ 状态转换表

现态				次态				序号
Y_4^n	Y_3^n	Y_2^n	Y_1^n	Y_4^{n+1}	Y_3^{n+1}	Y_2^{n+1}	Y_1^{n+1}	
0	0	0	0	0	0	0	1	①
0	0	0	1	0	0	1	1	②
0	0	1	0	0	1	0	1	③
0	0	1	1	0	1	1	1	
0	1	0	0	1	0	0	0	
0	1	0	1	1	0	1	1	④
0	1	1	0	1	1	0	0	
0	1	1	1	1	1	1	1	
1	0	0	0	0	0	0	0	⑧
1	0	0	1	0	0	1	0	⑦
1	0	1	0	0	1	0	0	
1	0	1	1	0	1	1	0	
1	1	0	0	1	0	0	0	⑥
1	1	0	1	1	0	1	0	
1	1	1	0	1	1	0	0	
1	1	1	1	1	1	1	0	⑤

④ 状态图



结论:

模8计数器(格雷码输出), 能够自启动

时序逻辑电路分析——同步时序总结

同步时序逻辑电路分析方法总结

确定系统变量（输入变量、输出变量、状态变量）

① 列写三组方程：

- 驱动方程（控制函数）
- 状态方程（次态方程）
- 输出方程（输出函数）

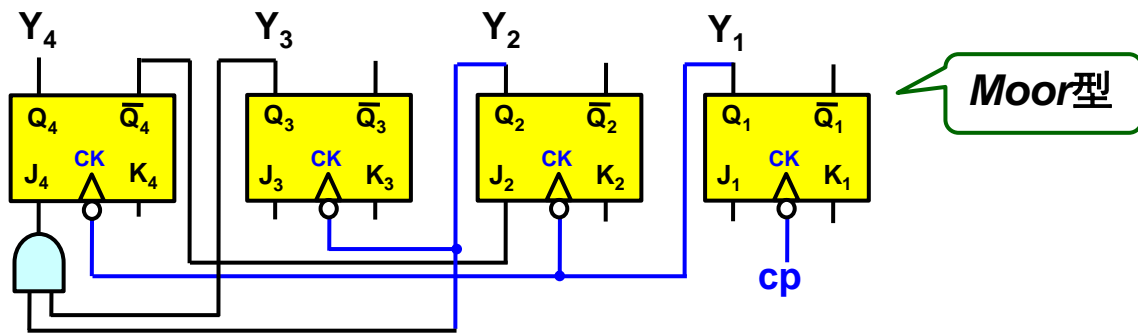
② 列写状态转换表：

- 写出所有输入及现态的取值组合；
- 将每一种取值组合带入次态方程和输出方程，计算后的得出次态值和输出值；
- 从表中第一行开始，寻找状态转换规律；

③ 画出完整的状态图；

④ 得出电路功能，并说明能否自启动

时序逻辑电路分析——示例3: 异步时序



① 输入方程

$$\left\{ \begin{array}{l} J_4 = Y_3^n Y_2^n \\ K_4 = 1 \\ J_3 = K_3 = 1 \\ J_2 = \overline{Y_4^n}, K_2 = 1 \\ J_1 = K_1 = 1 \end{array} \right.$$

② 次态方程

$$\left\{ \begin{array}{l} Y_4^{n+1} = J_4 \overline{Y_4^n} + \overline{K_4} Y_4^n = \overline{Y_4^n} Y_3^n Y_2^n \\ Y_3^{n+1} = J_3 \overline{Y_3^n} + \overline{K_3} Y_3^n = \overline{Y_3^n} \\ Y_2^{n+1} = J_2 \overline{Y_2^n} + \overline{K_2} Y_2^n = \overline{Y_4^n} \overline{Y_2^n} \\ Y_1^{n+1} = J_1 \overline{Y_1^n} + \overline{K_1} Y_1^n = \overline{Y_1^n} \end{array} \right. \quad \begin{array}{l} CP_4 = Y_1 \downarrow \\ CP_3 = Y_2 \downarrow \\ CP_2 = Y_1 \downarrow \\ CP_1 \downarrow \end{array}$$

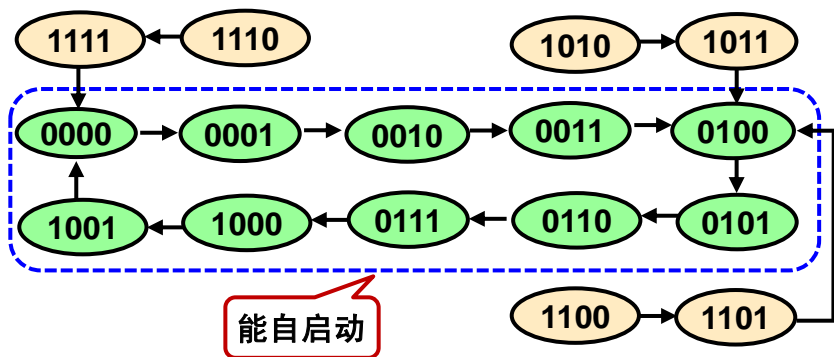
时序逻辑电路分析——异步时序示例3

② 次态方程

$$\begin{cases} Y_4^{n+1} = J_4 \bar{Y}_4^n + \bar{K}_4 Y_4^n = \bar{Y}_4^n Y_3^n Y_2^n & CP_4 = Y_1 \downarrow \\ Y_3^{n+1} = J_3 \bar{Y}_3^n + \bar{K}_3 Y_3^n = \bar{Y}_3^n & CP_3 = Y_2 \downarrow \\ Y_2^{n+1} = J_2 \bar{Y}_2^n + \bar{K}_2 Y_2^n = \bar{Y}_4^n \bar{Y}_2^n & CP_2 = Y_1 \downarrow \\ Y_1^{n+1} = J_1 \bar{Y}_1^n + \bar{K}_1 Y_1^n = \bar{Y}_1^n & CP_1 \downarrow \end{cases}$$

④ 状态图

8421 BCD 码异步加法计数器



③ 状态转换表

现态				次态				时钟			
Y_4^n	Y_3^n	Y_2^n	Y_1^n	Y_4^{n+1}	Y_3^{n+1}	Y_2^{n+1}	Y_1^{n+1}	cp_4	cp_3	cp_2	cp_1
0	0	0	0	0	0	0	1	无	无	无	↓
0	0	0	1	0	0	1	0	↓	无	↓	↓
0	0	1	0	0	0	1	1	无	无	无	↓
0	0	1	1	0	1	0	0	↓	↓	↓	↓
0	1	0	0	0	1	0	1	无	无	无	↓
0	1	0	1	0	1	1	0	↓	无	↓	↓
0	1	1	0	0	1	1	1	无	无	无	↓
0	1	1	1	1	0	0	0	↓	↓	↓	↓
1	0	0	0	1	0	0	1	无	无	无	↓
1	0	0	1	0	0	0	0	↓	无	↓	↓
1	0	1	0	1	0	1	1	无	无	无	↓
1	0	1	1	0	1	0	0	↓	↓	↓	↓
1	1	0	0	1	1	0	1	无	无	无	↓
1	1	0	1	0	1	0	0	↓	无	↓	↓
1	1	1	0	1	1	1	1	无	无	无	↓
1	1	1	1	0	0	0	0	↓	↓	↓	↓

时序逻辑电路分析——异步时序总结

异步时序逻辑电路分析方法总结

确定系统变量（输入变量、输出变量、状态变量）

① 确定每个触发器的时钟由谁供给？

② 列写三组方程：

- 驱动方程（控制函数）、状态方程（次态方程）、输出方程（输出函数）

③ 列写状态转换表：

- 首先，从假定（或给定）的某一个初始状态开始，每来一个外输入及外接时钟脉冲，确定与之对应的触发器次态及输出；
- 其次，确定该触发器的状态改变能否给其它触发器提供需要的时钟边沿。若能，则与之相应的其它触发器动作。否则，与之相应的其它触发器保持；重复该步骤，直到所有触发器的次态都确定为止。
- 接着，该次态成为新的现态，来一个外输入及外接时钟脉冲，重复上述操作，直到所有的 2^n 个现态到次态的转换都已计算完毕；从表中第一行开始，寻找状态转换规律；

③ 画出完整的状态图；

④ 得出电路功能，并说明能否自启动