Ch8 Sequential Logic Design Practices

Main contents:

（1）Timing diagram时序图

（2）Registers寄存器

（3）Counters计数器（重点）

以74x163为代表的电路分析与应用，包含：电路功能分析；任意模m计数器设计；序列发生器；控制各种不同器件

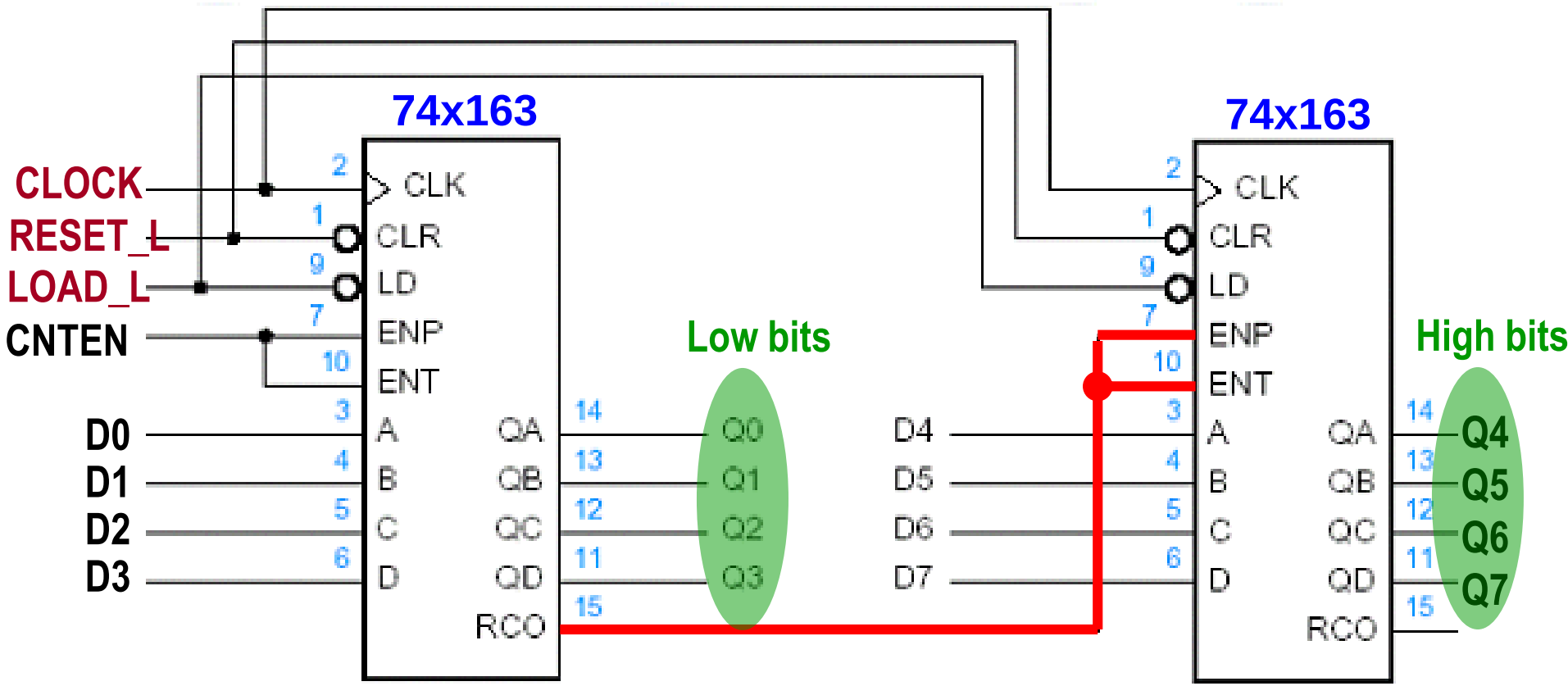
（4）Shift registers移位计数器（重点）

以74x194为代表的电路分析与应用，包含：

8.5.1 Shift-Register Structure
8.5.2 MSI Shift Registers
8.5.3 Shift-Register Counters
8.5.4 Ring Counters
8.5.5 Johnson Counters
8.5.6 Linear Feedback Shift-Register Counters(LFSR)

Example 3: Cascading Binary Counters

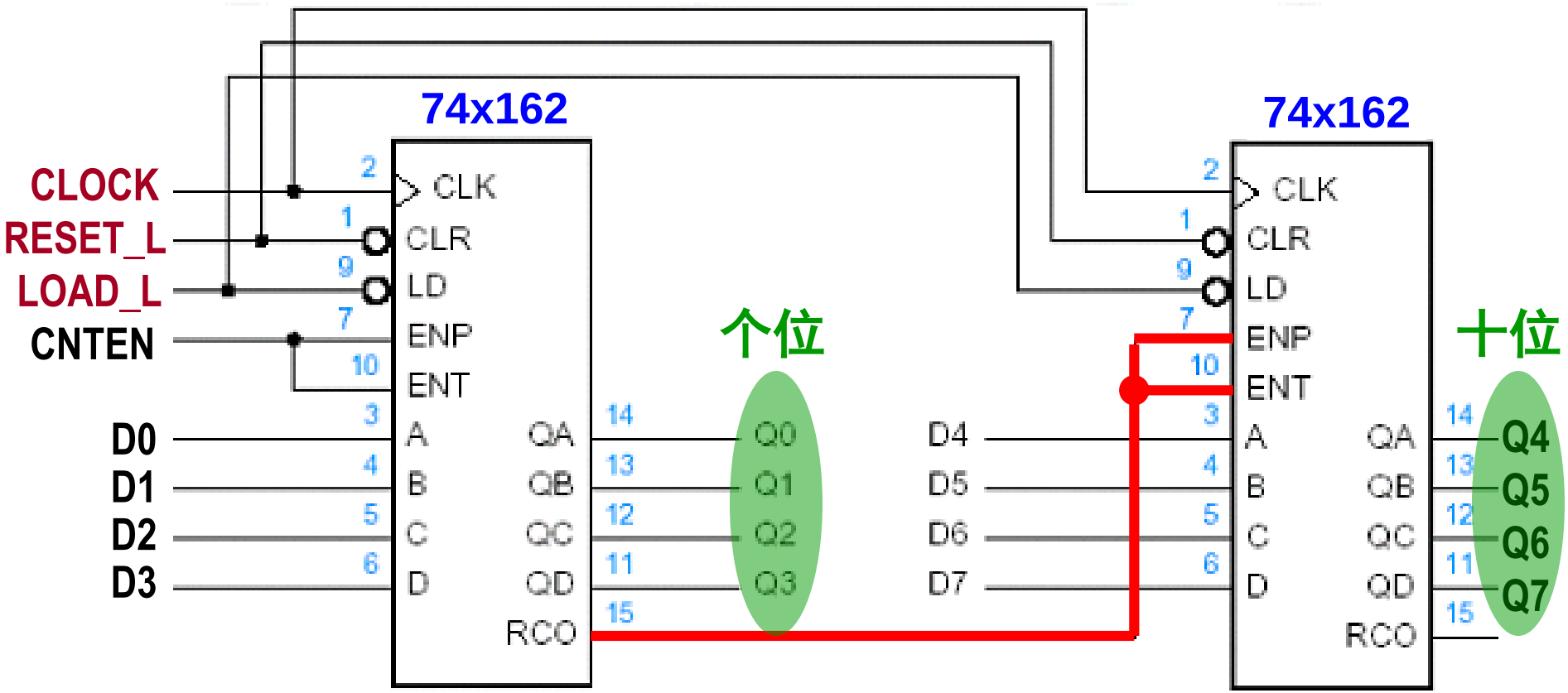
—— $m>2^n$



Counting range：0～255

0000 0000~1111 1111

Example 4: Cascading Decimal Counters (BCD计数器)

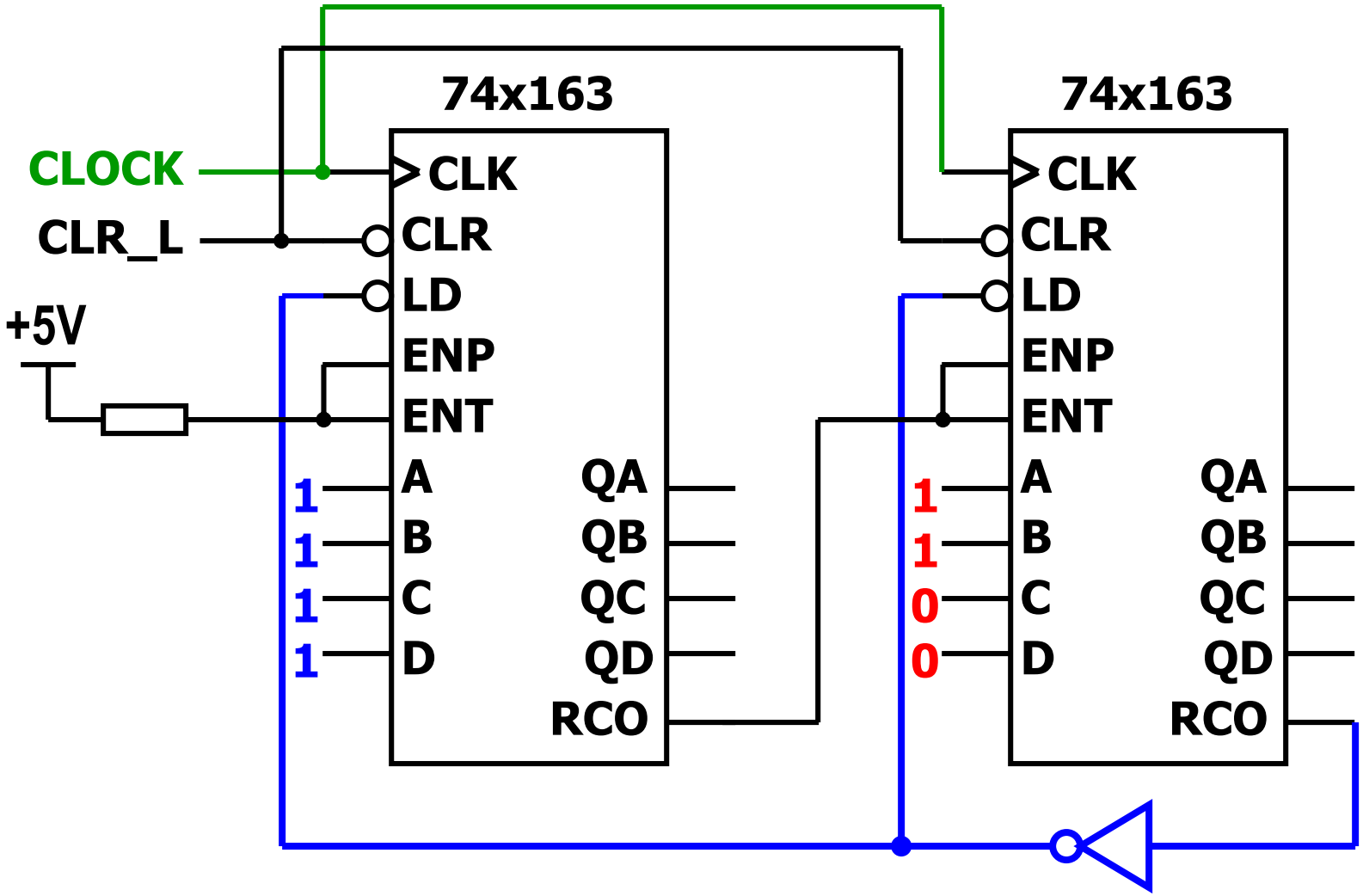


Counting range：0～99

0000 0000～1001 1001

Example 5: modulo-193 counter using 74x163

63~255:193 states

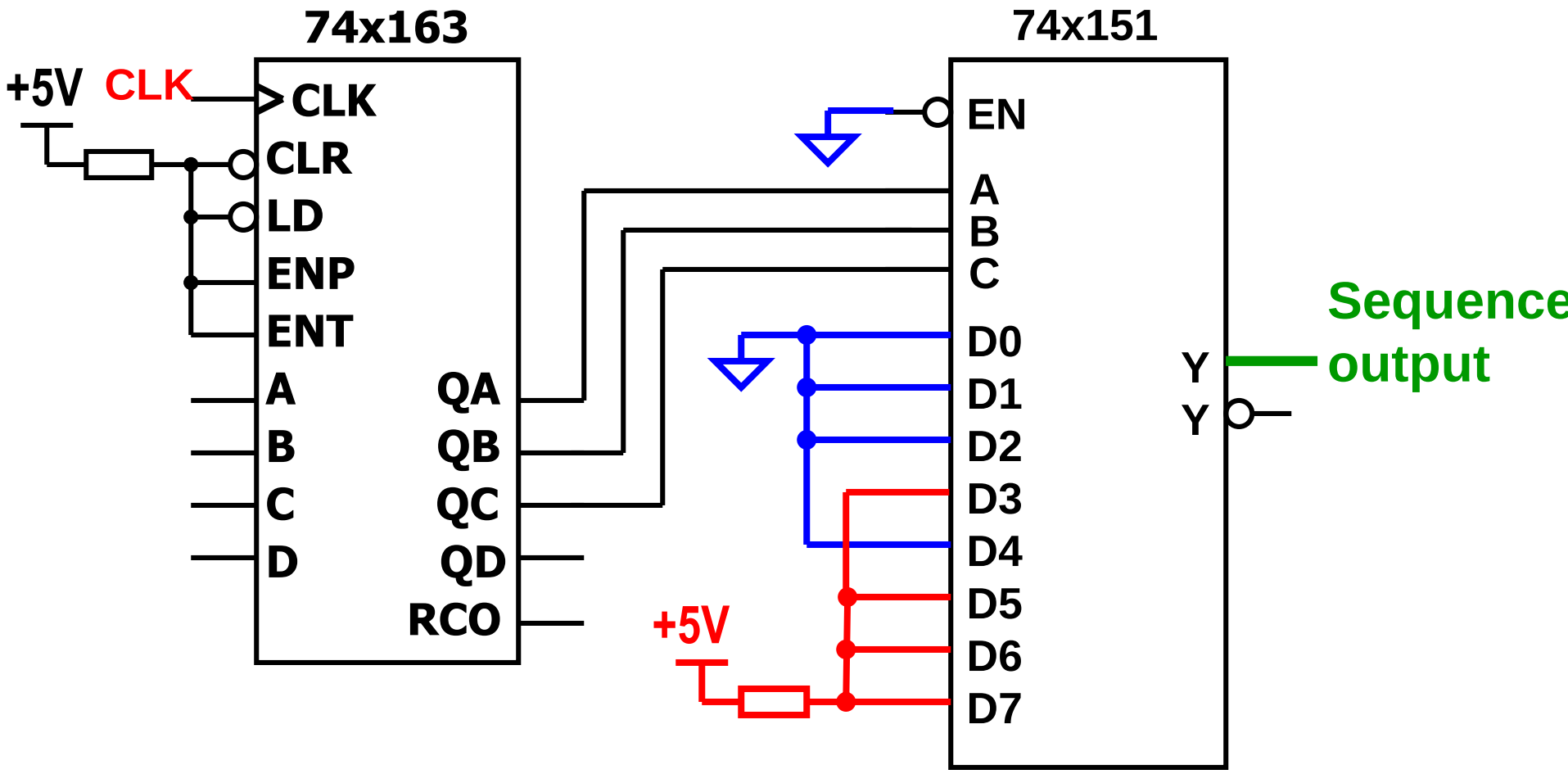


$63_{10} = (\ 0011\ 1111\ )_2$

Counter application(2)

-------- sequence generator序列信号发生器

Example6 ： generating 8-bit periodic sequence 00010111

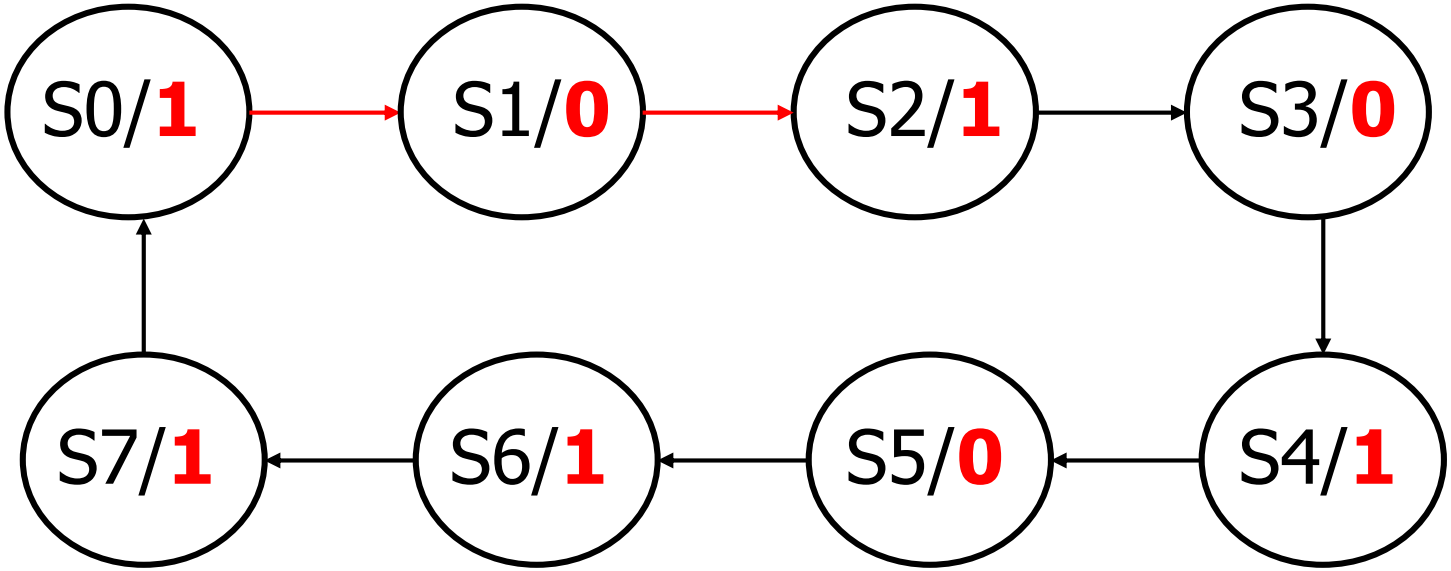


计数器的应用（2）

- 序列信号发生器　又例：在时钟作用下周期产生序列**10101011，10101011，…**

MOORE机　　8个状态

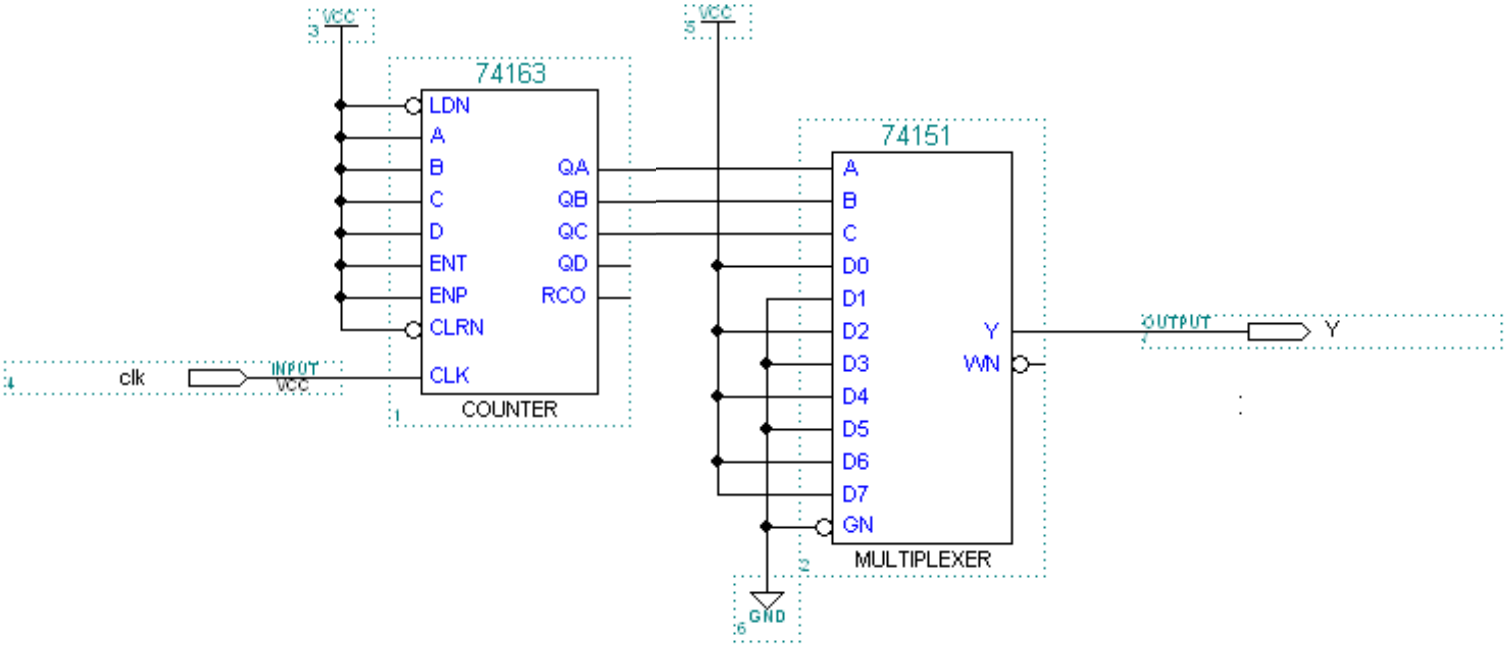
计数器+组合电路



计数器的应用（2）

- 序列信号发生器
- 又例：在时钟作用下周期产生序列**10101011，10101011，…**

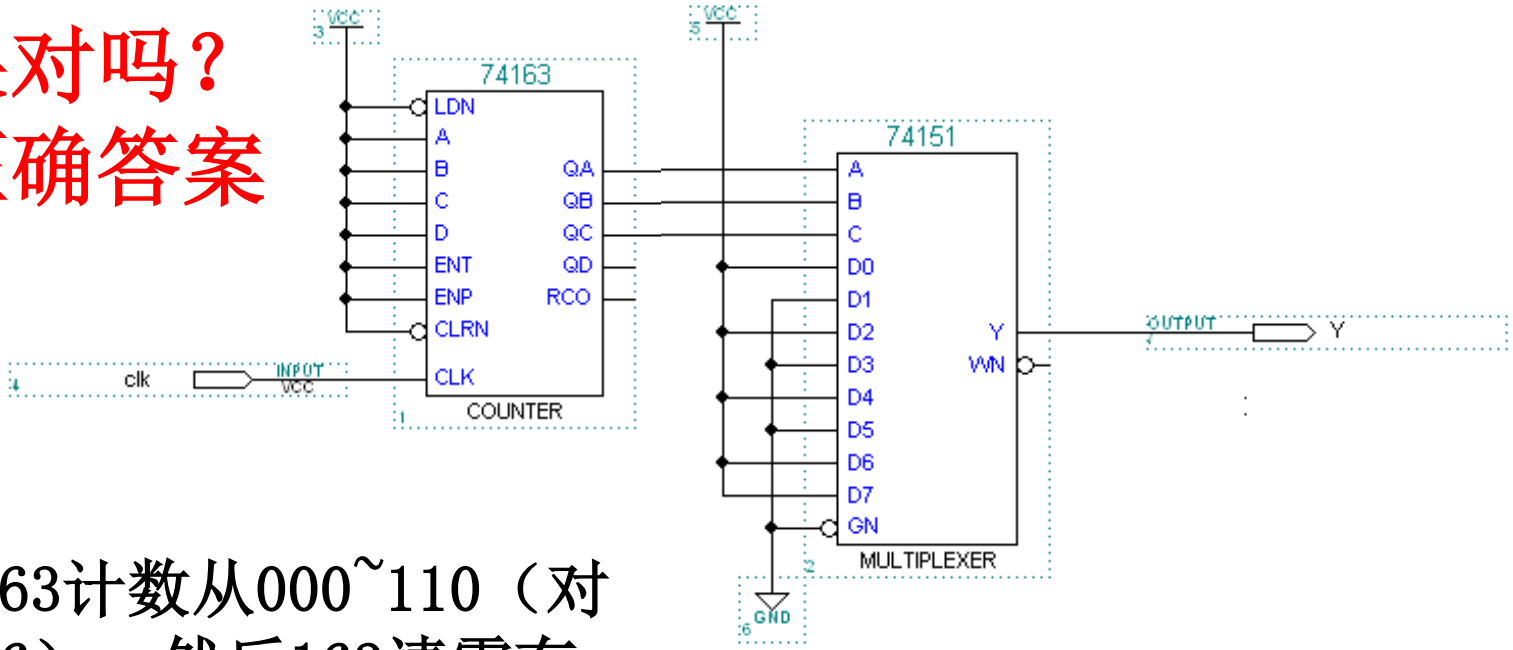
计数器+组合电路



计数器的应用（2）

- 序列信号发生器
- 思考：在时钟作用下周期产生序列1011011，1011011，…　　计数器+组合电路

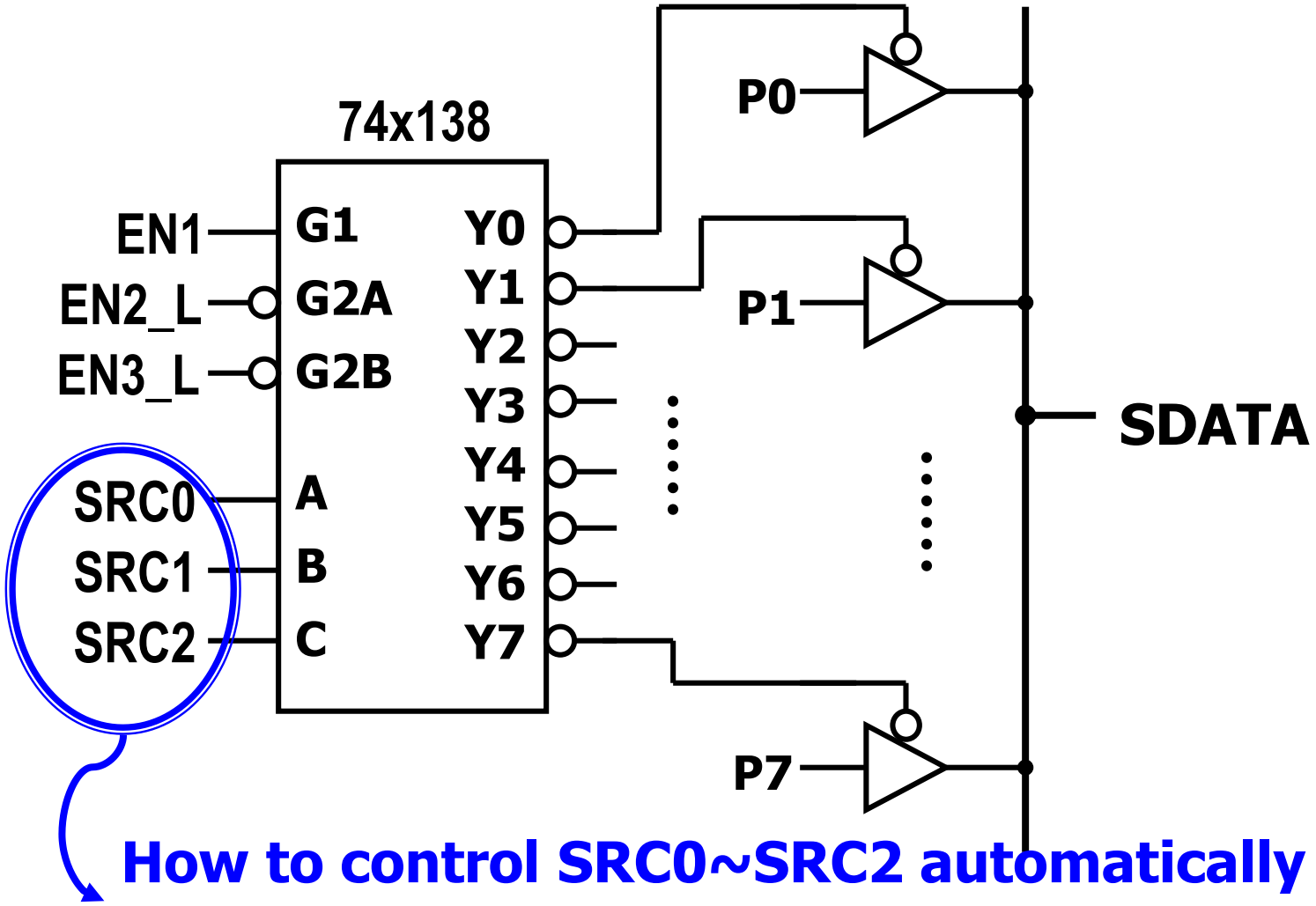
这个结果对吗？
请给出正确答案



参考方法：163计数从000~110（对应151的D0~D6），然后163清零有效。画出实现上述原理的电路图

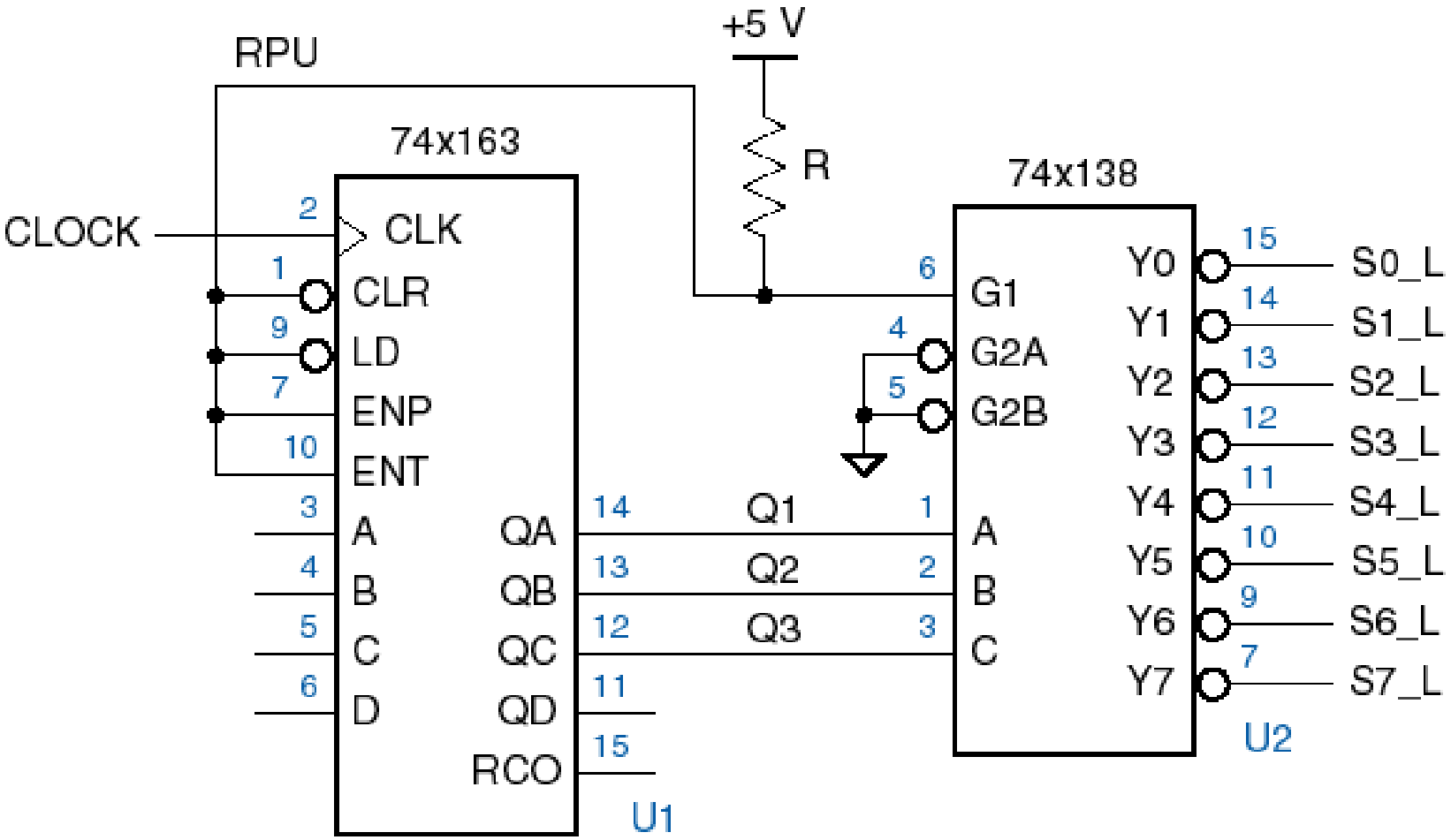
Counter application(3)

-------- control different devices

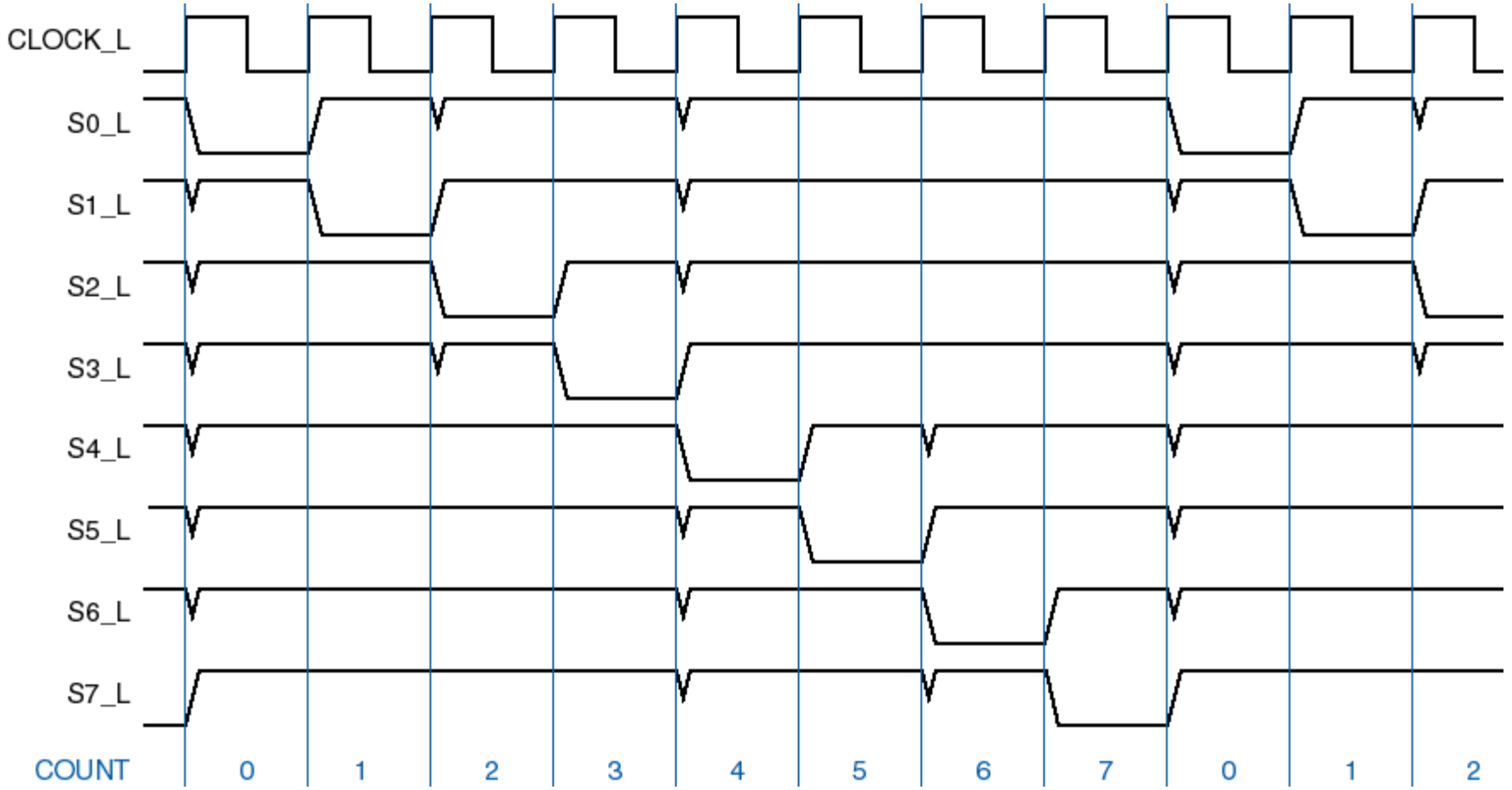


How to control SRC0~SRC2 automatically

Decoding binary-counter states

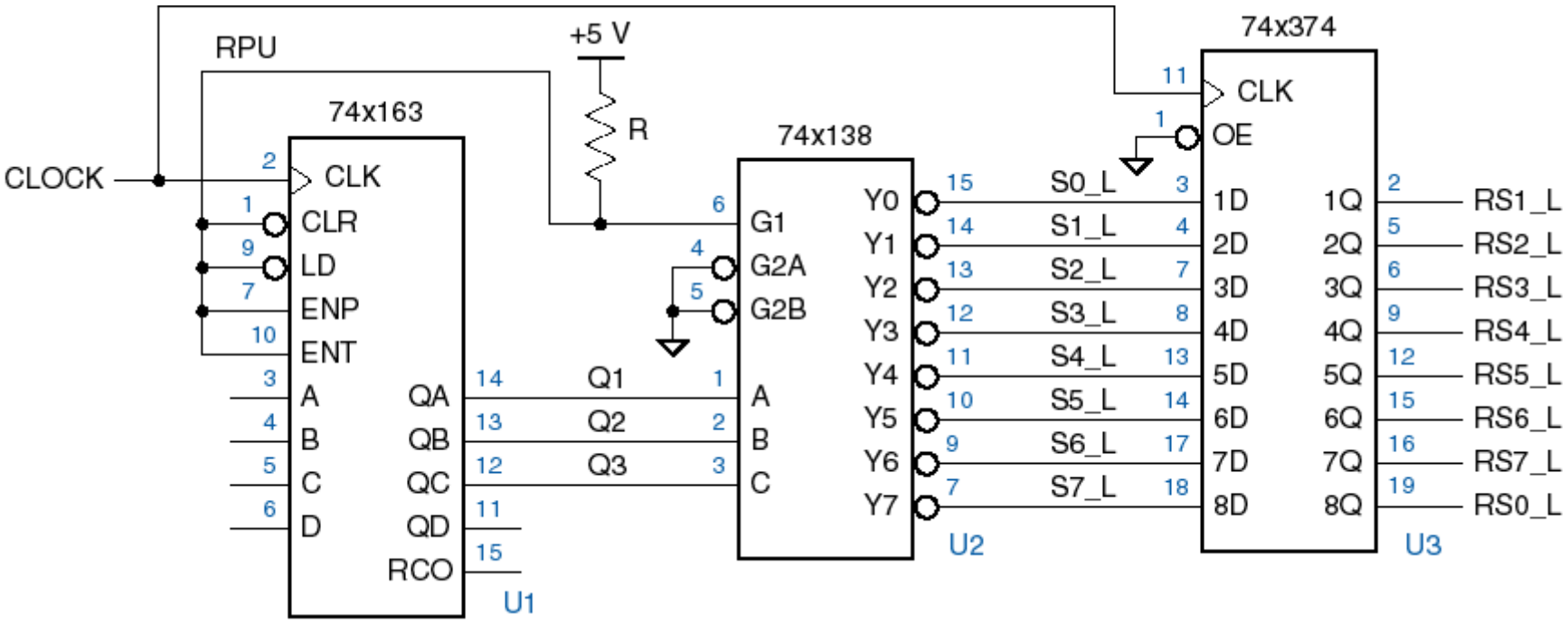


Decoder waveforms



- **Glitches may or may not be a concern.**
- 原因：一次状态转移中，若2个及以上计数位同时变化，计数器不能准确同步、译码器输出有不同的延迟，则产生尖峰；

Glitch-free outputs



Solution：

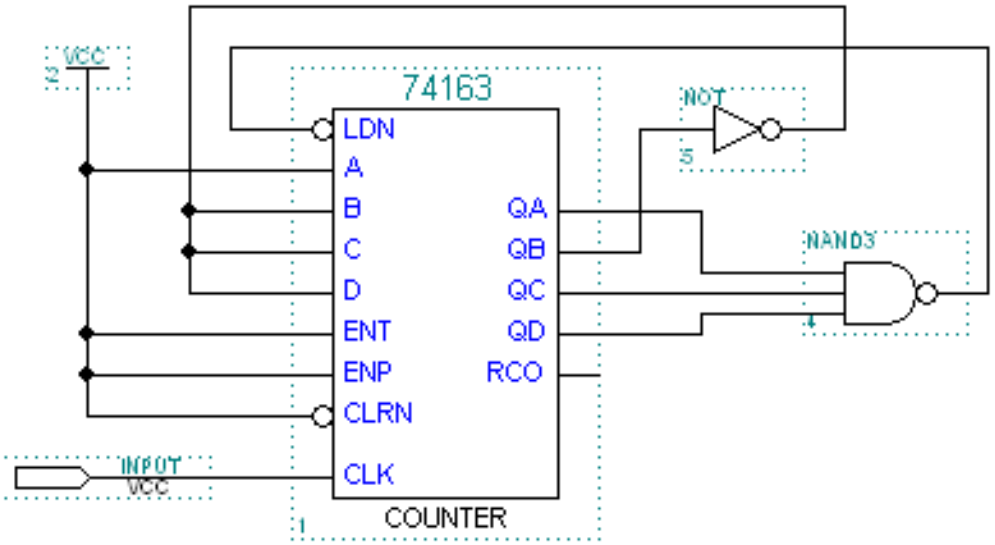
（1）74x374 Registered outputs delayed by one clock tick.

（2）We'll show another way to get the same outputs later, using a shift register.

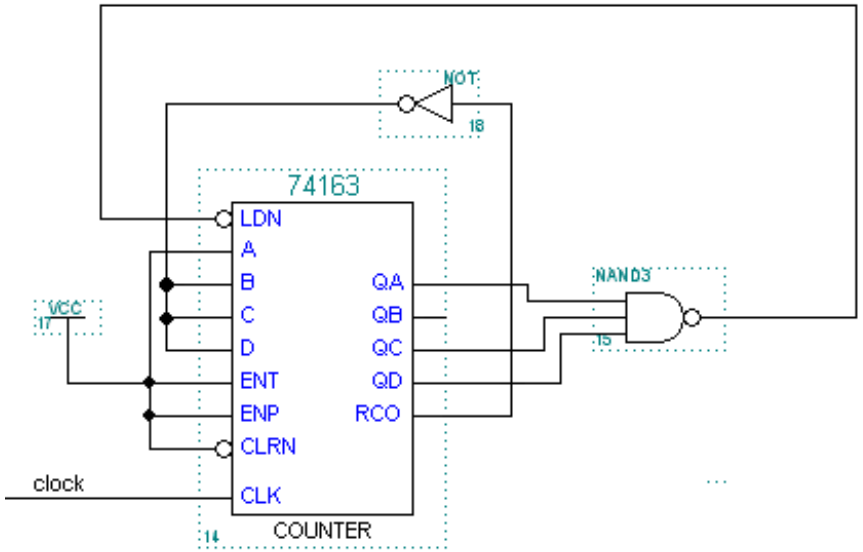
Thinking：

利用74X163和必要的门电路设计一模14计数器，
计数序列为：1、2、3、4、5、6、7、8、9、10、11、12、13、15、1、2…..。
完成设计并画出电路

Solution 1



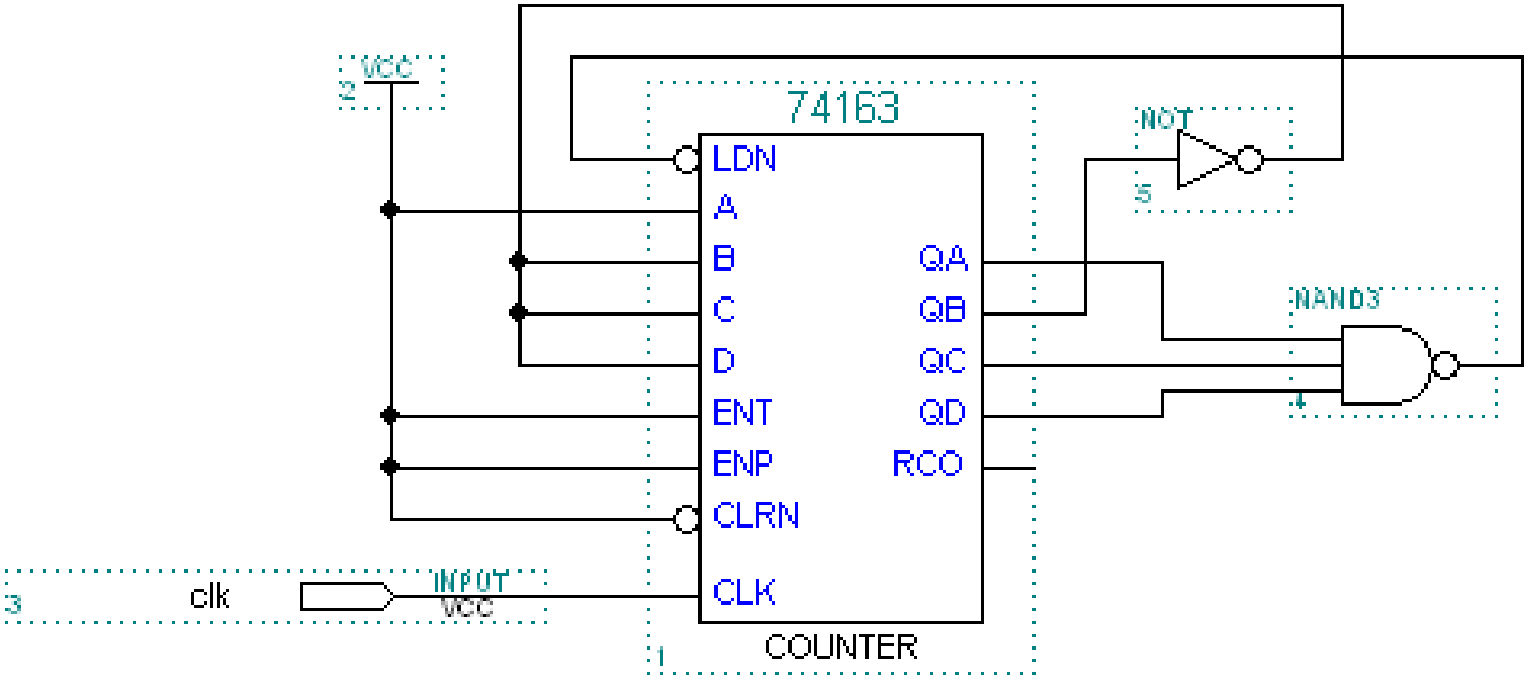
Solution 2



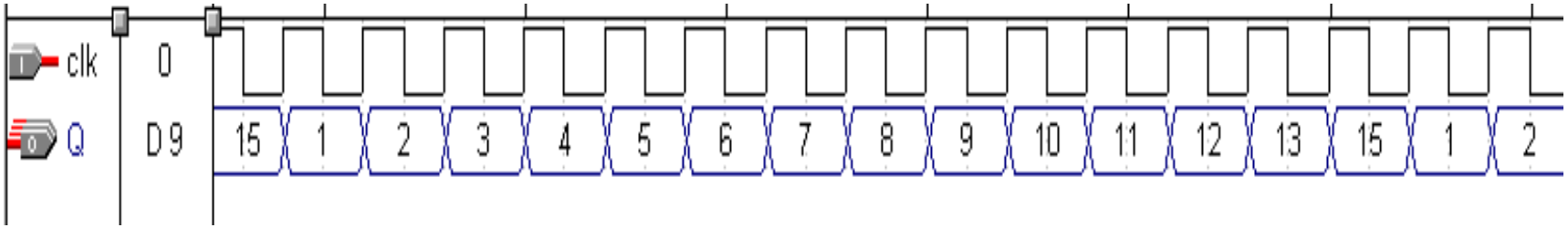
1101（13）后载入1111（15）；

1111（15）后载入0001（1）；

Solution 1的仿真分析：



仿真波形：



8.5 Shift register移位寄存器

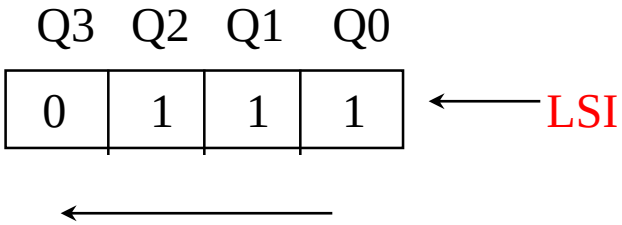
Contents：

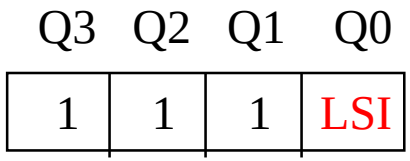
- **Shift Registers types:**
  - Serial in, serial out shift register
  - Serial in, parallel out shift register
  - Parallel in, serial out shift register
  - Parallel in, parallel out shift register
- **MSI shift registers: Typical 74x194**
- **Shift Register Applications:**
  - shift -register counter
  - sequence generator/detector
  - serial-to-parallel conversion

8.5 Shift register移位寄存器

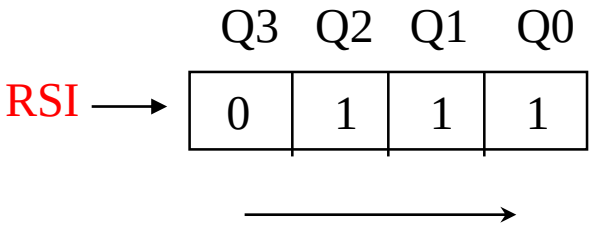
8.5.1 Shift-Register Structure

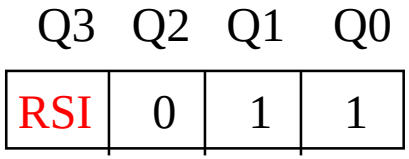
- 定义：**Multi-bit register that moves stored data bits left/right (1 bit shift per clock cycle)**
  - **Shift Left is towards MSB**





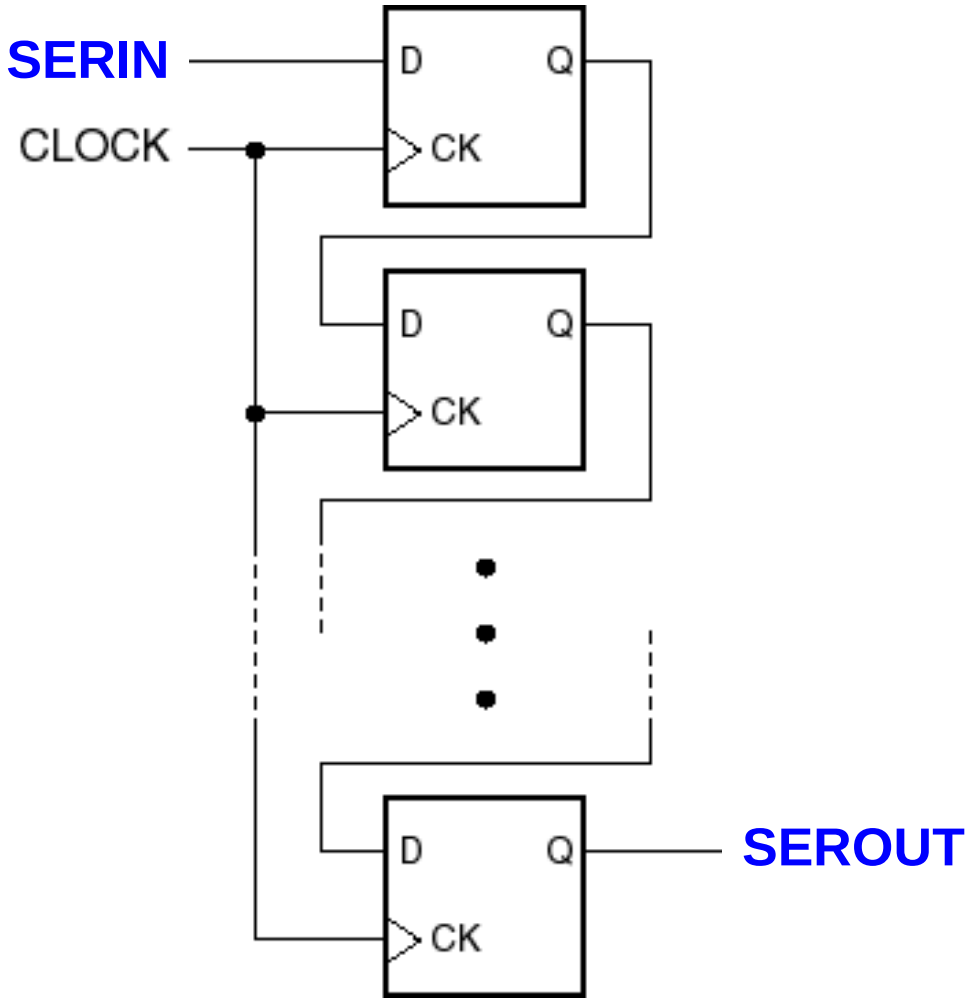
  - **Shift Right is towards LSB**





Shift Registers types

- **Serial-in, serial-out**



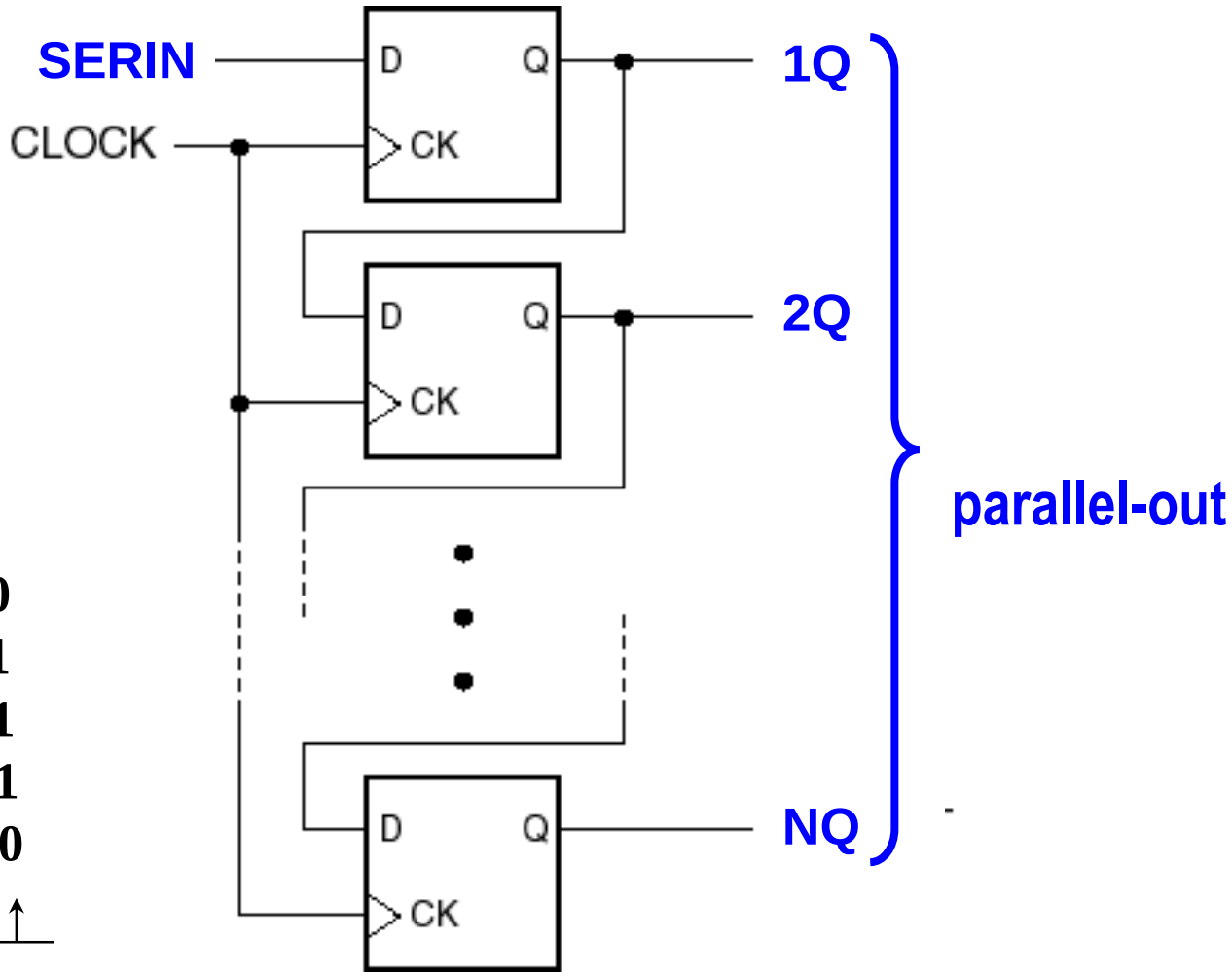
Serial Out = Serial In
delayed by n clock period

4-bit shift register
example:
serin: 1 0 1 1 0 0 1 1 1 0
serout: - - - - 1 0 1 1 0 0
clock: ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

Serial-in, parallel-out

serial-in

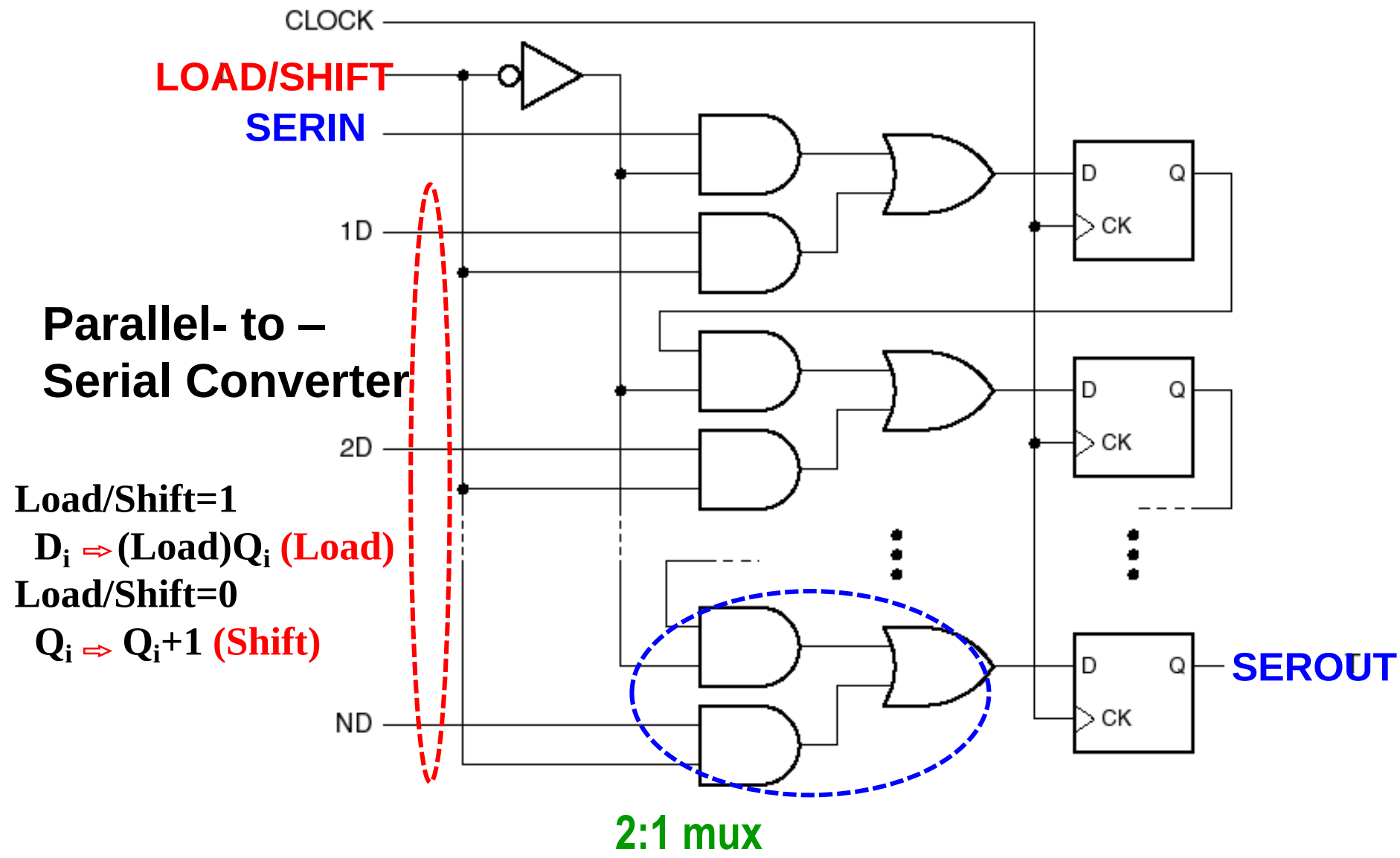
serial-to-parallel converter



4-bit shift register example:

serin:	**1 0 1 1 0 0 1 1 1 0**
1Q:	**- 1 0 1 1 0 0 1 1 1**
2Q:	**- - 1 0 1 1 0 0 1 1**
3Q:	**- - - 1 0 1 1 0 0 1**
4Q:	**- - - - 1 0 1 1 0 0**
clock:	↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

Parallel-in, serial-out



Parallel- to – Serial Converter

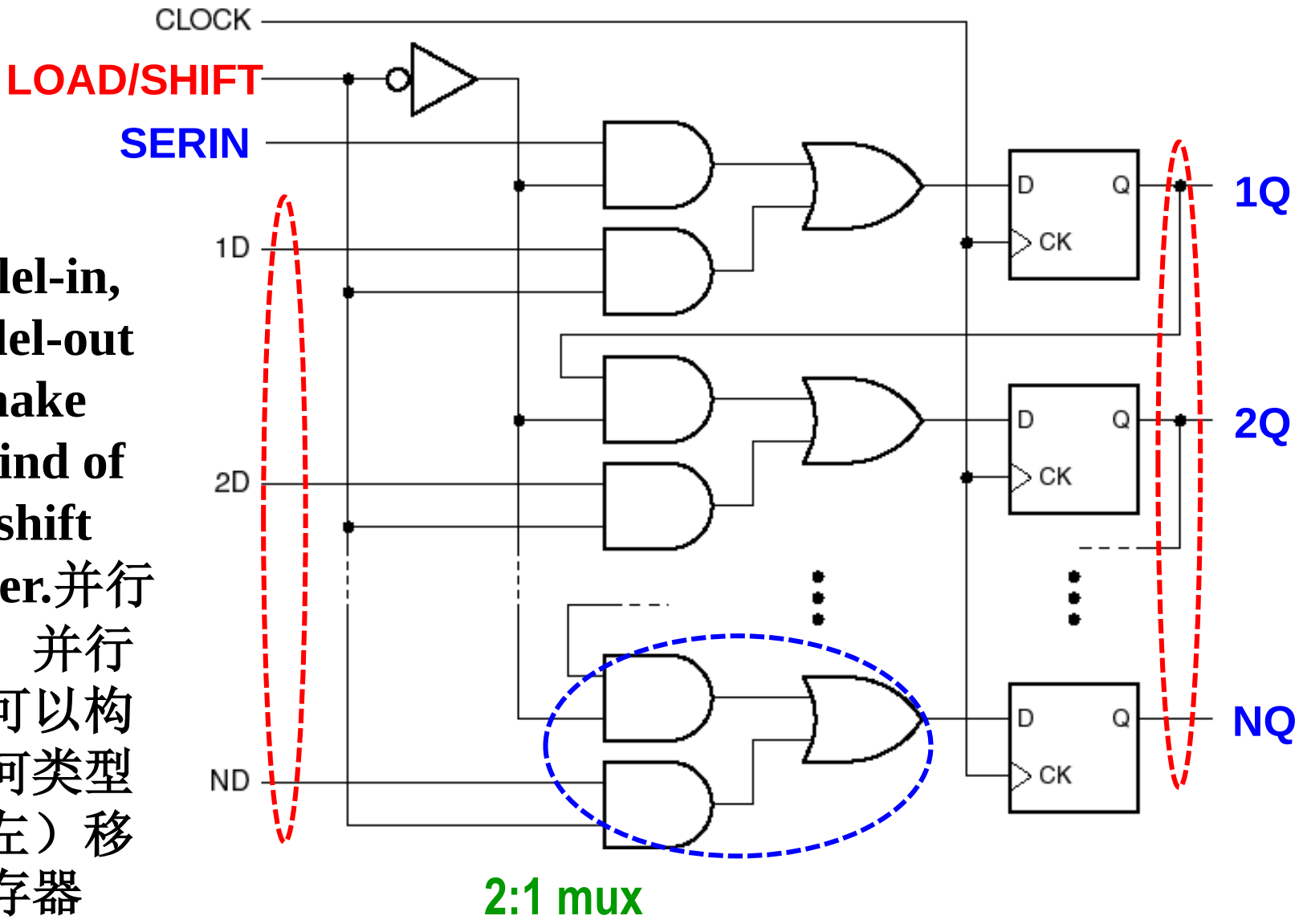
Load/Shift=1

$D_i \Rightarrow$ (Load)Q_i (Load)

Load/Shift=0

$Q_i \Rightarrow Q_i+1$ (Shift)

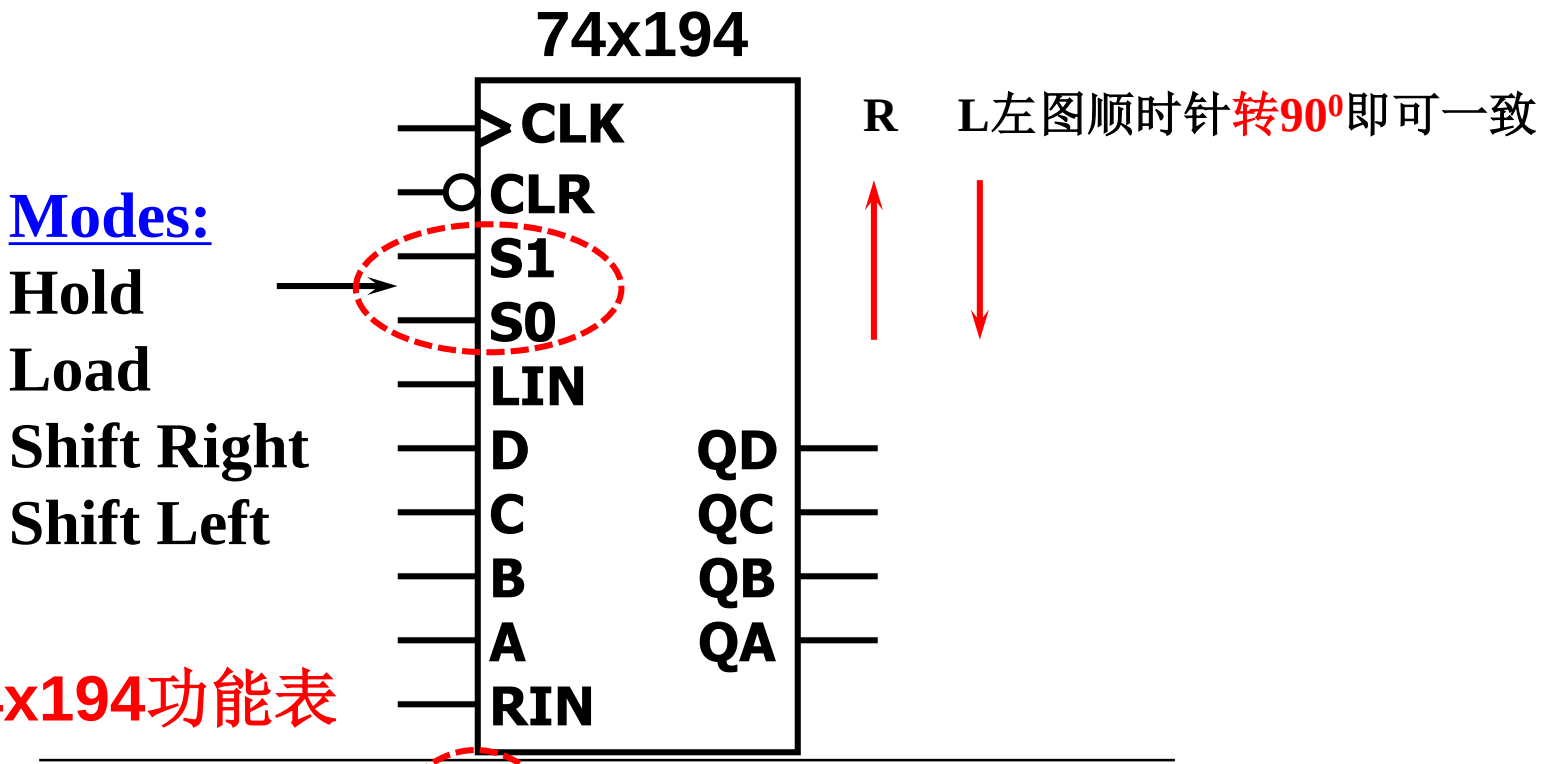
Parallel-in, Parallel-out



Parallel-in, parallel-out can make any kind of (left) shift register.并行输入、并行输出可以构成任何类型的（左）移位寄存器

8.5.2 MSI shift registers

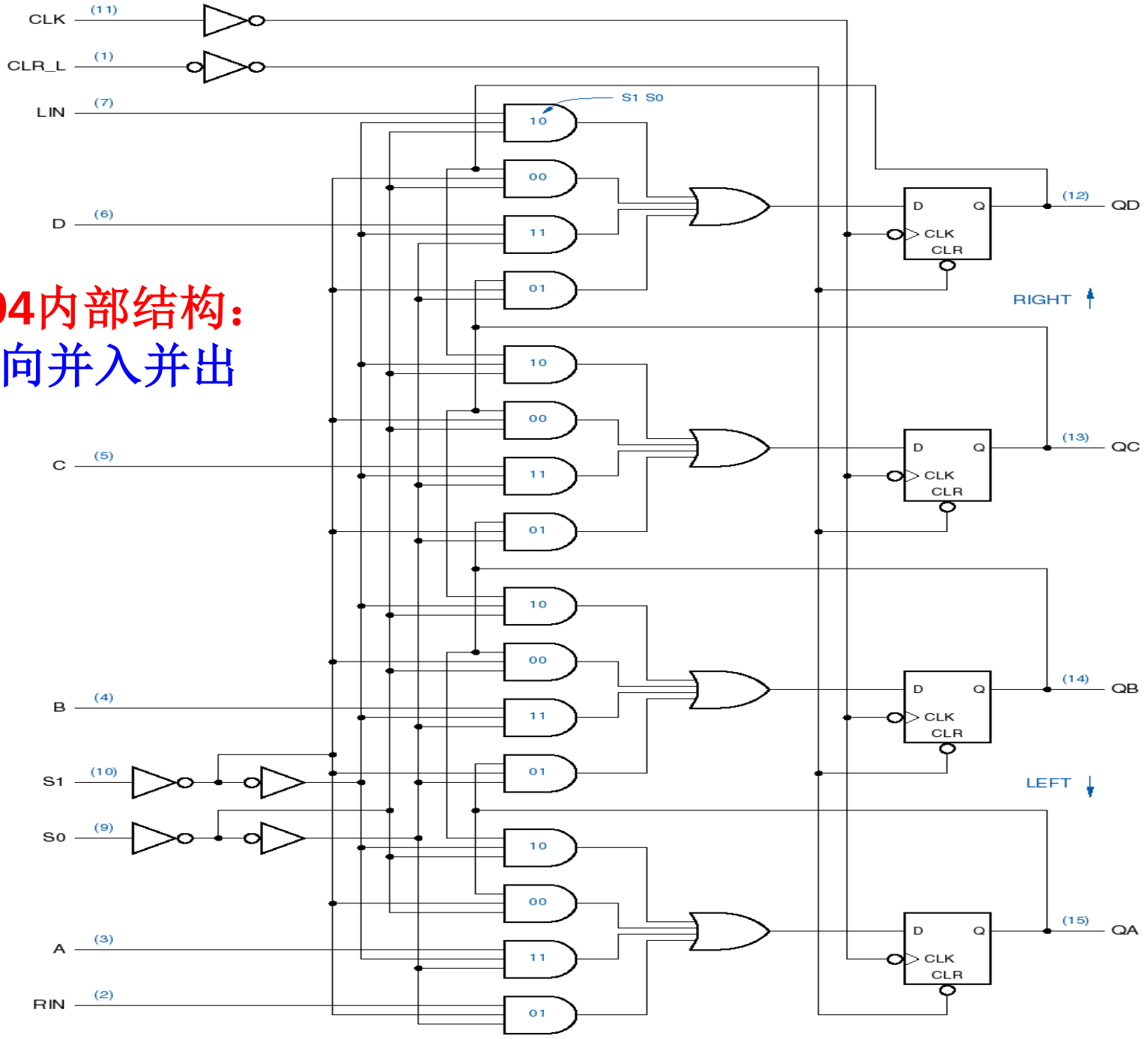
Bi-directional Universal Shift Registers



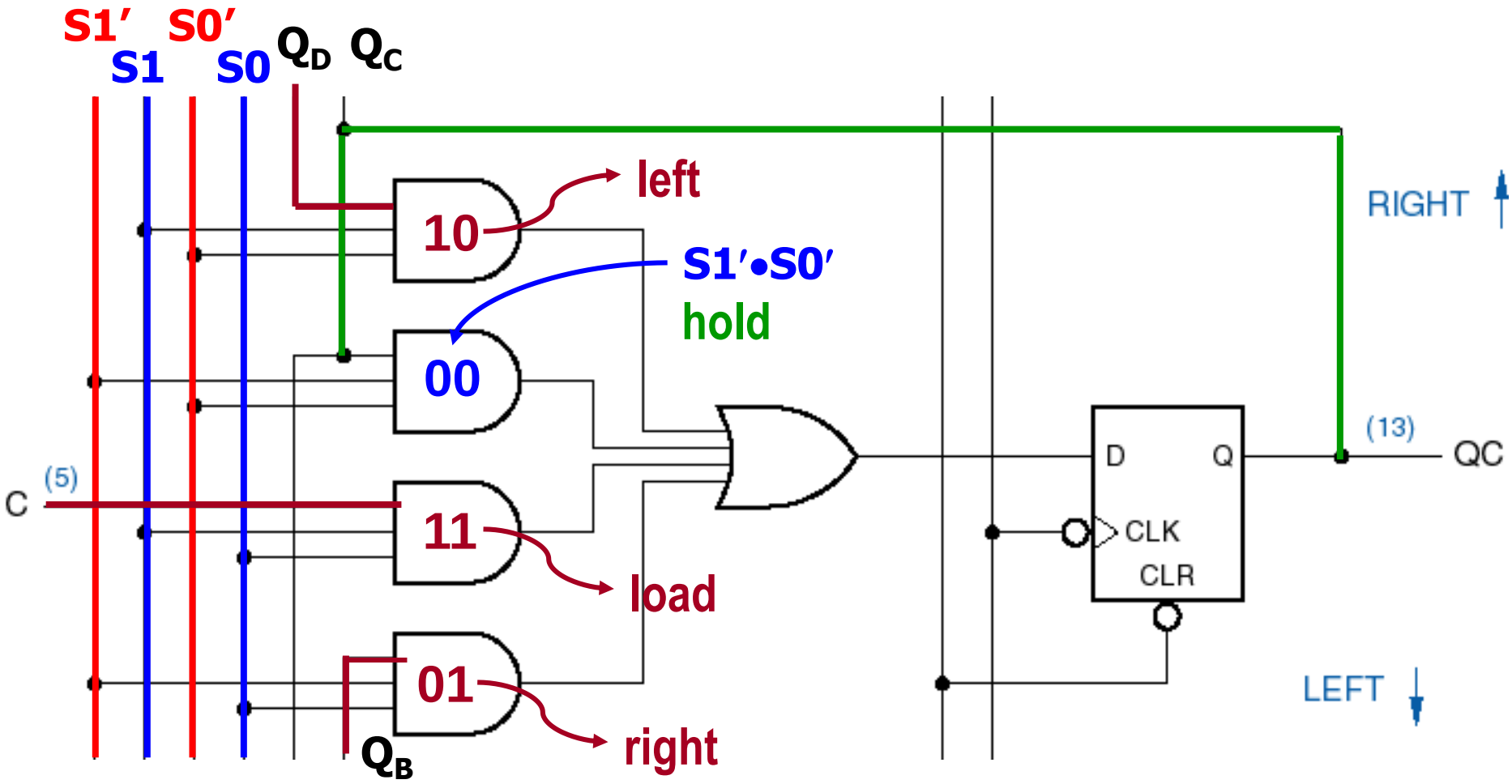
74x194功能表

Function	Mode S1	Mode S0	Next state QA*	QB*	QC*	QD*
Hold	0	0	QA	QB	QC	QD
Shift right/up	0	1	RIN	QA	QB	QC →
Shift left/down	1	0	QB	QC	QD	LIN ←
Load	1	1	A	B	C	D

74x194内部结构：
4位双向并入并出



74x194输入C端口的结构及工作原理剖析

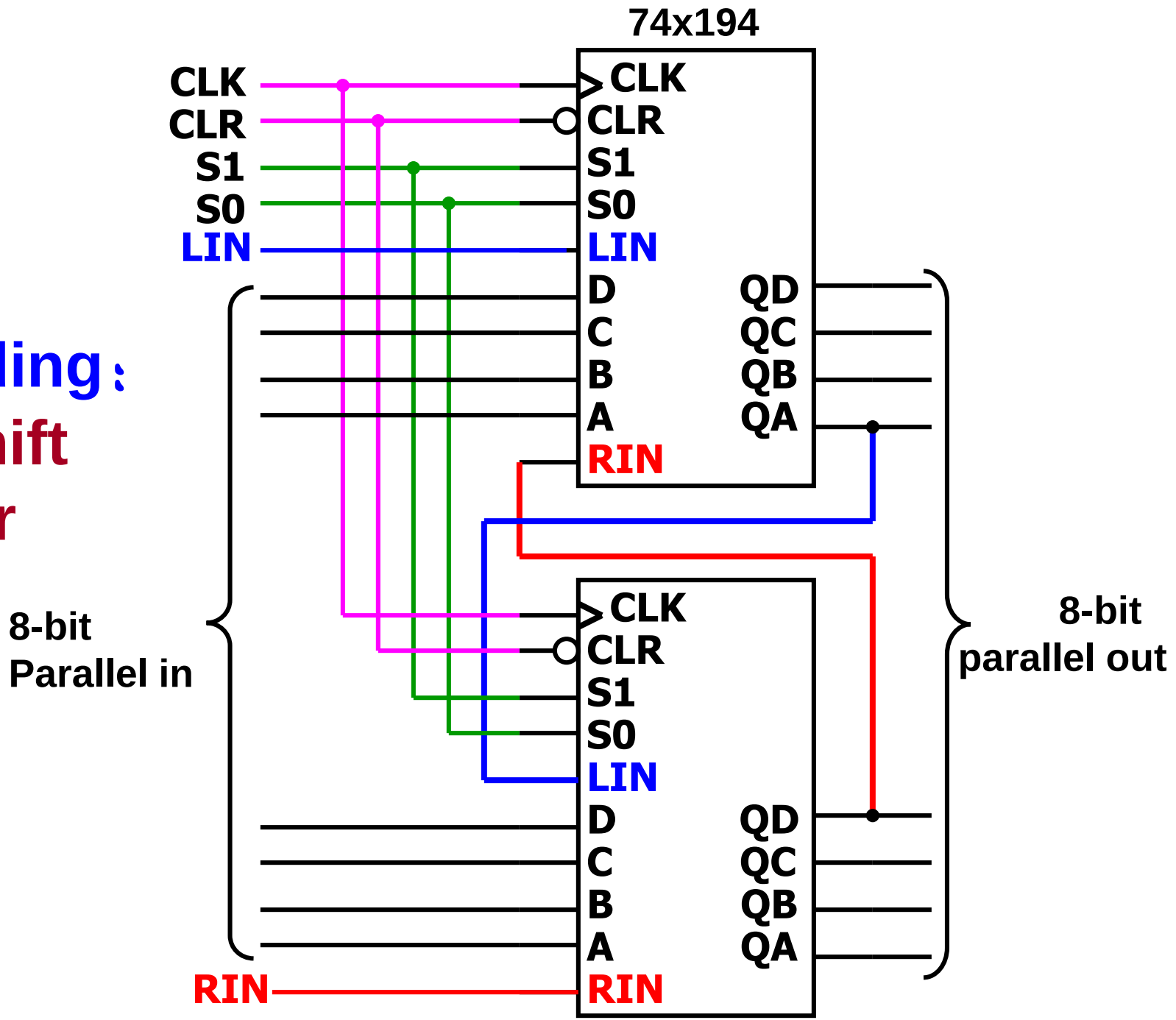


转移方程（4选1）：

$Q_i^* = D = S1'\cdot S0'\cdot Q_i + S1'\cdot S0\cdot Q_{i+1} + S1\cdot S0'\cdot Q_{i-1} + S1\cdot S0\cdot In_i$

其中，这里的$Q_i = Q_C$, $Q_{i+1} = Q_B$, $Q_{i-1} = Q_D$, $In_i = C$

Cascading：8-bit shift register



Shift Register Applications

8.5.3 Shift-register counters

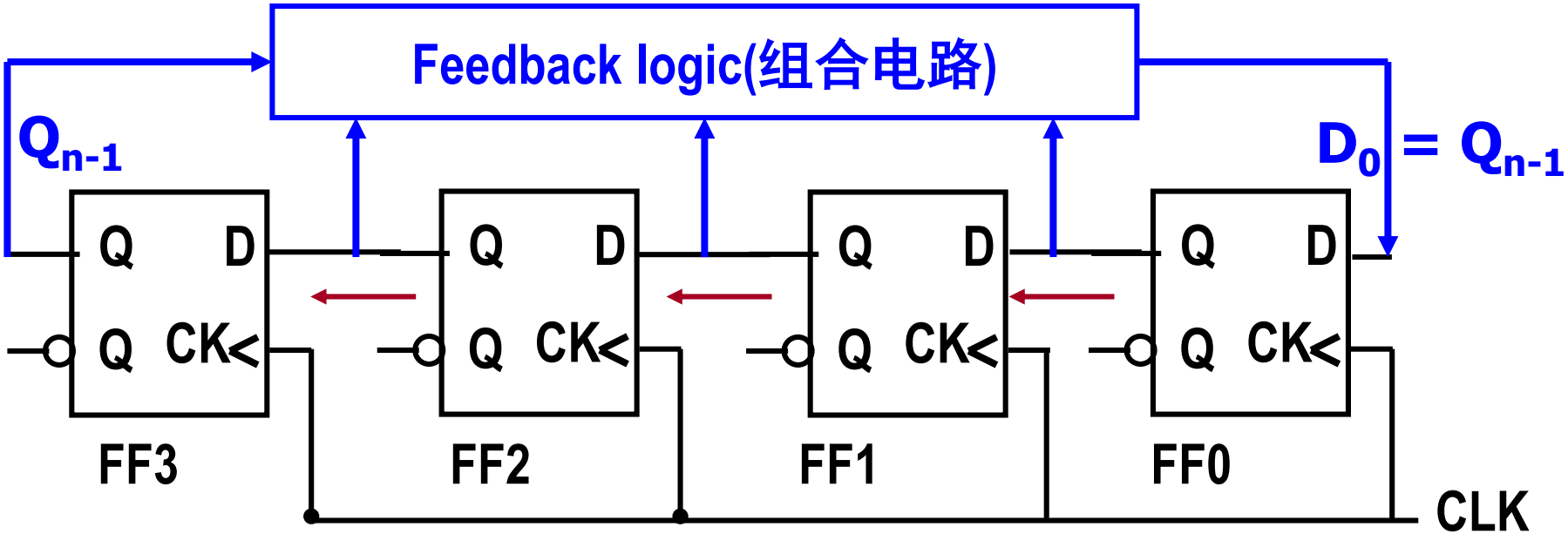
定义：与组合逻辑一起构成循环状态图的状态机

特点：(1)状态图是循环的；(2)计数顺序非升非降，应用于控制领域；

下面介绍Shift-register counters在三个方面的应用

8.5.4 Ring Counters　　8.5.5 Johnson Counters

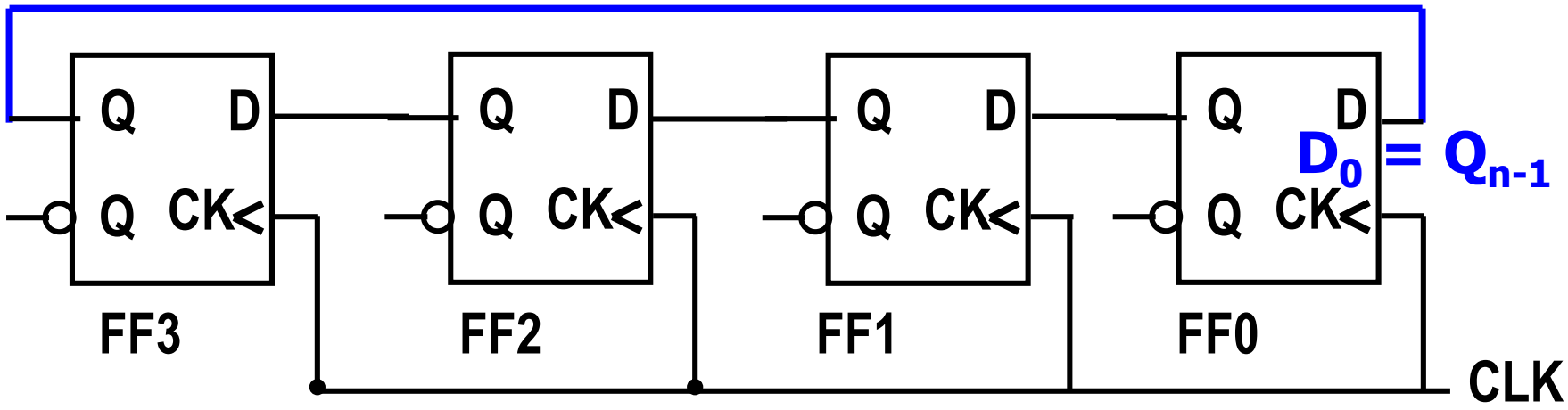
8.5.6 Linear Feedback Shift-Register Counters(LFSR)



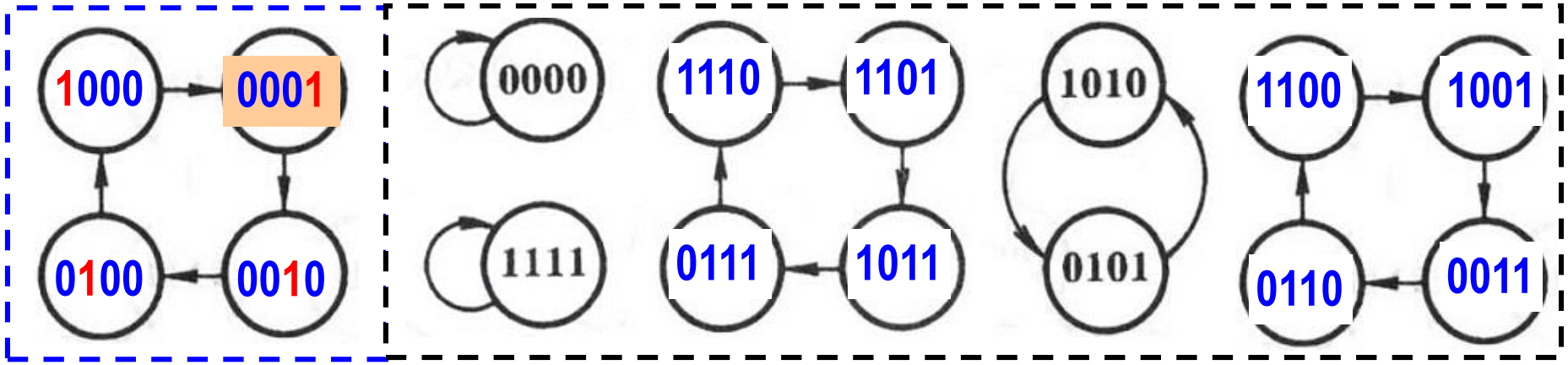
激励方程：$D_0 = F(Q_0, Q_1, \cdots, Q_{n-1})$

8.5.4 Ring counter —— not self-correcting or not self-starting

$Q_{n-1} = Q_3$



Assume initial state:0001

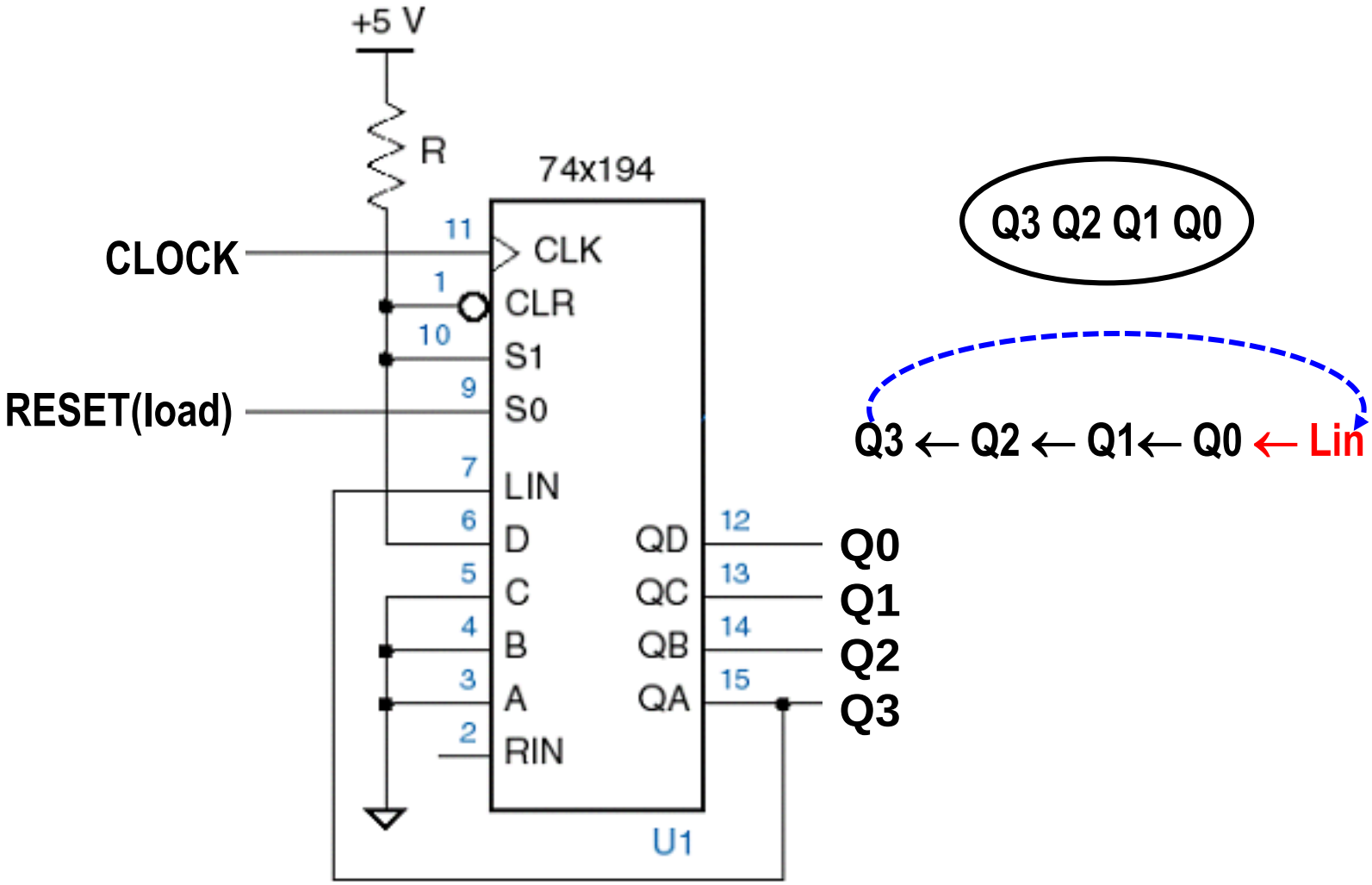


有效状态：单个“1”循环

Q3 Q2 Q1 Q0

无效状态

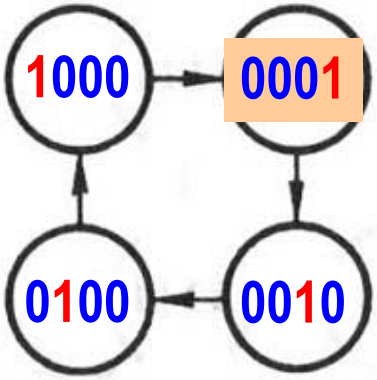
8.5.4 Ring counter —— not self-correcting or not self-starting



Ring counter（单个“1”循环） with 74x194的连接线路

自检/自校正/自启动设计 for a single circulating "1"

Q3 Q2 Q1 Q0



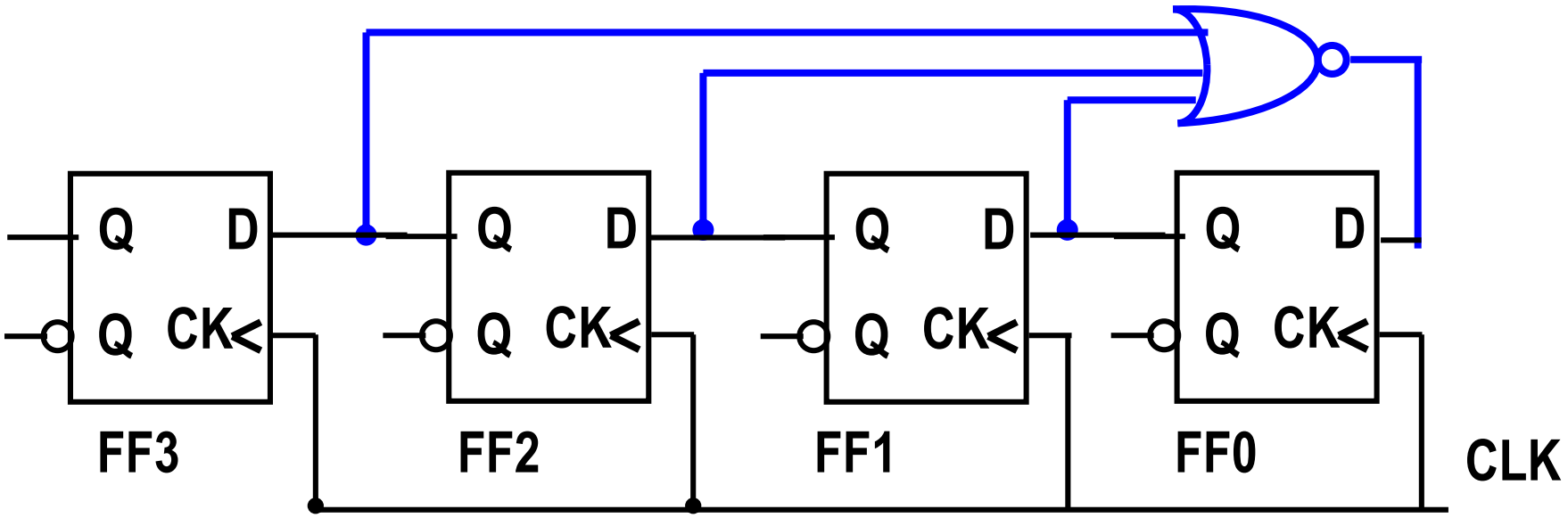
校正原理1：

When Q2,Q1,Q0 are all 0, D0=1,else（否则） D0=0

注意：在左边所示的有效状态图中，Q3Q2Q1Q0从1000到0001是正常转移，无需校正。该方法主要针对前述无效状态图的校正，比如，0000

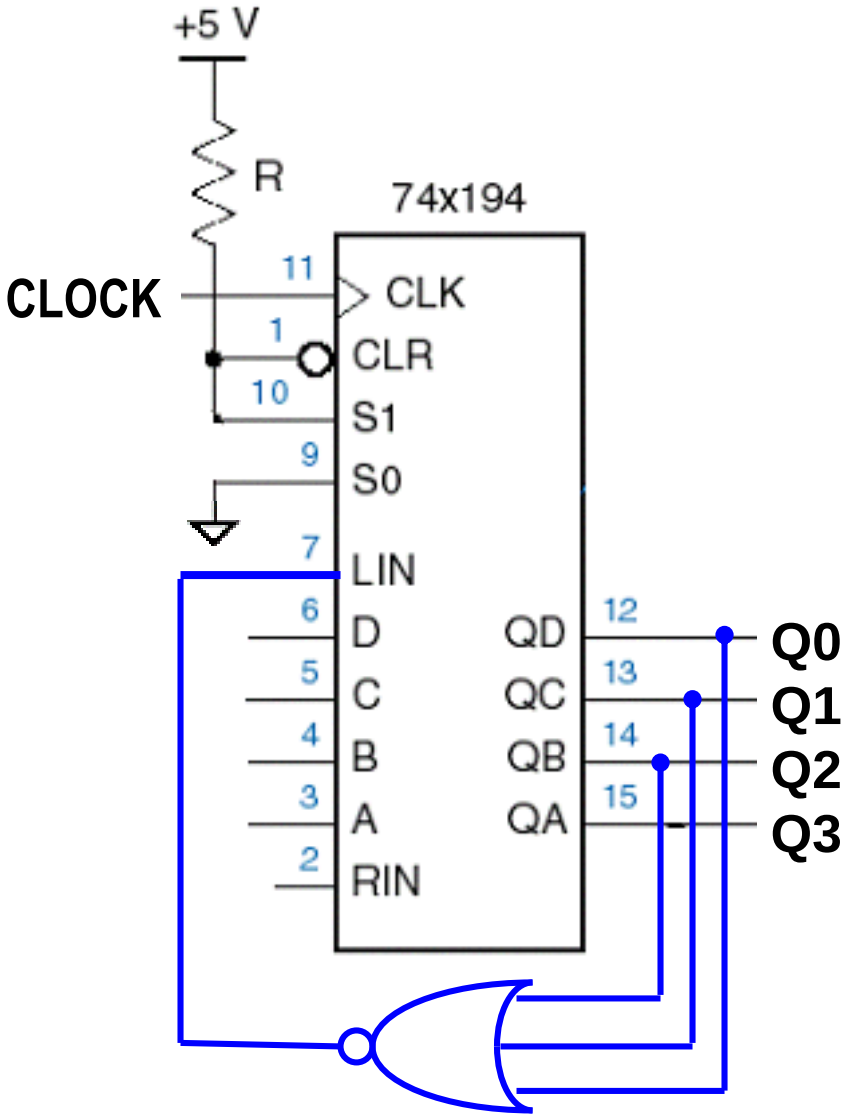
有效状态：单个"1"循环

Self-correcting（分立电路自校正）



自检/自校正/自启动设计 for a single circulating “1”

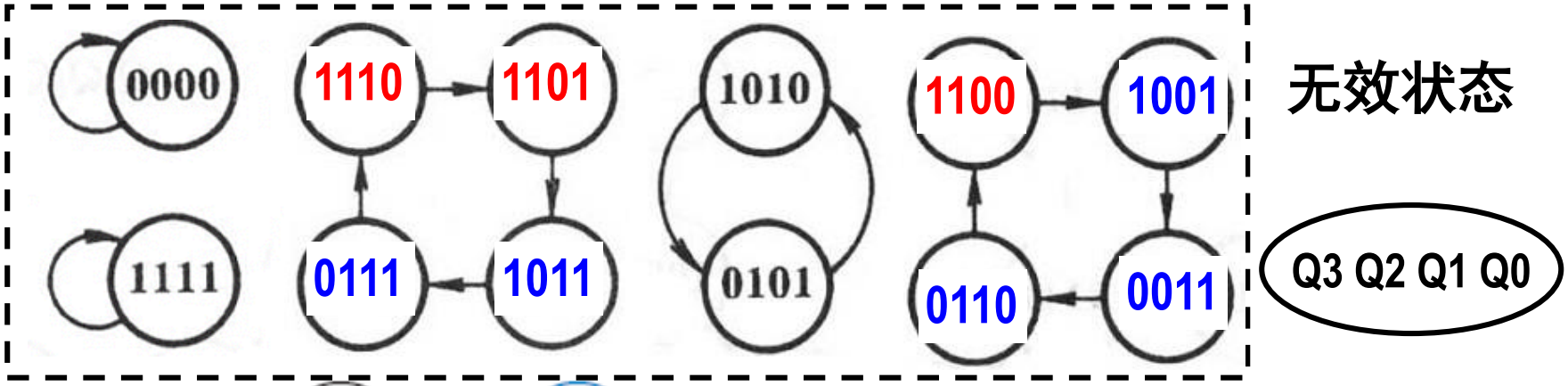
Self-correcting using 74x194（MSI自校正）

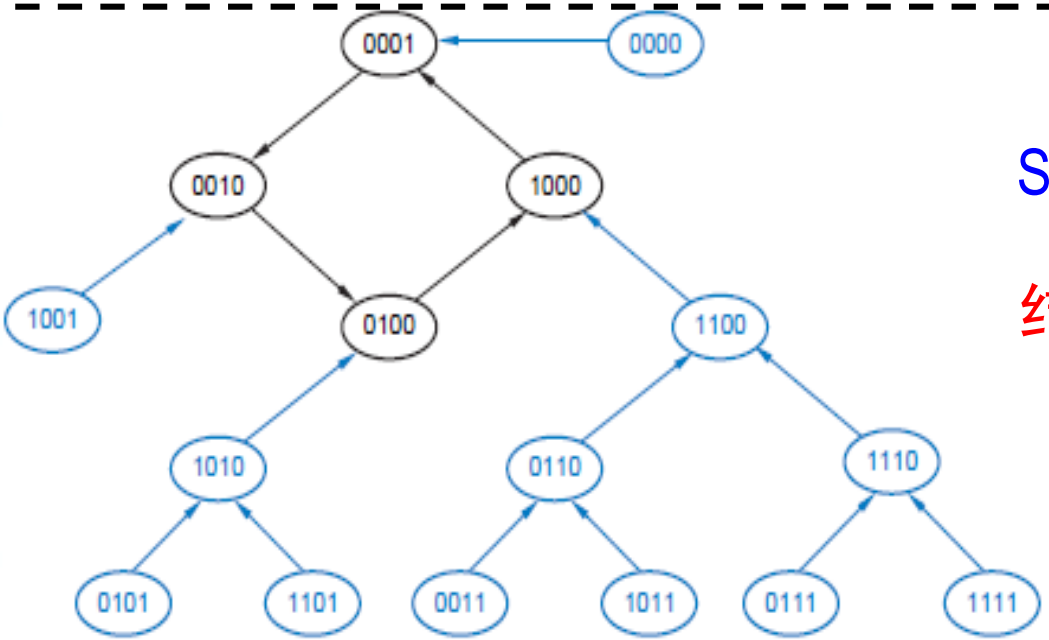


自检/自校正/自启动设计 for a single circulating "1"

校正原理1：When Q2,Q1,Q0 are all 0, D0=1,else（否则）D0=0

因此，对无效状态图中的0000校正后则转移到0001，而不是0000的循环；对无效状态图中的1111校正后则转移到1110，而不是1111的循环；同理，可分析如下无效状态转移的自检情况





Self-correcting状态图

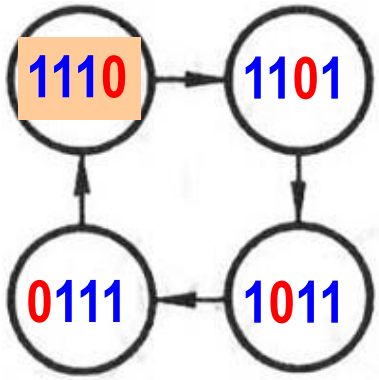
结论：校正后，所有16个状态完全自检

自检/自校正/自启动设计 a single circulating "0"

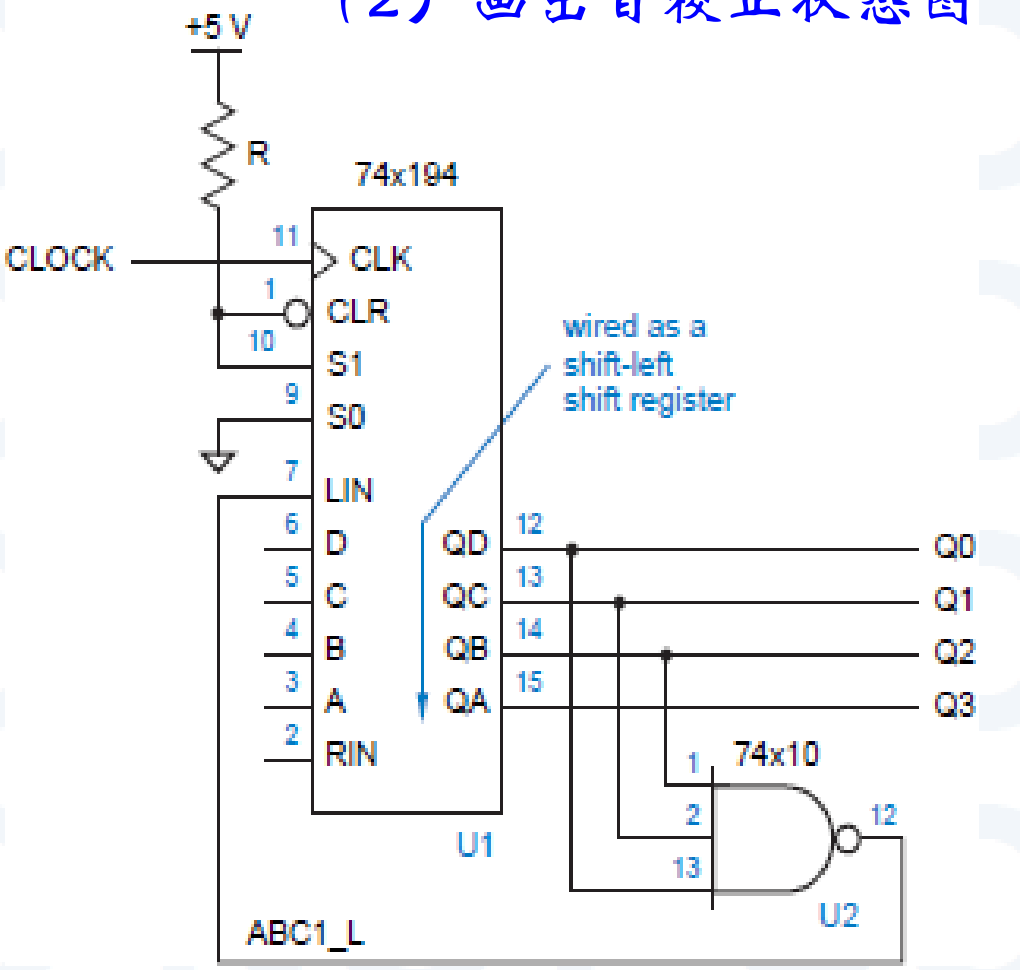
校正原理2：When Q2,Q1,Q0 are all 1, D0=0,else（否则） D0=1

思考：（1）画出其余无效状态图；
（2）画出自校正状态图

有效状态：单个"0"循环

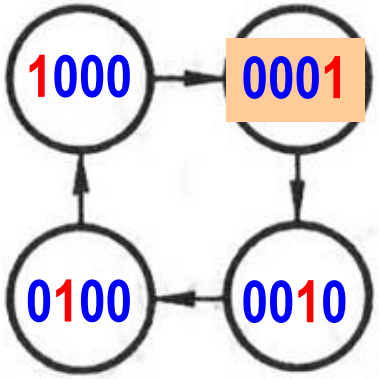


Q3 Q2 Q1 Q0

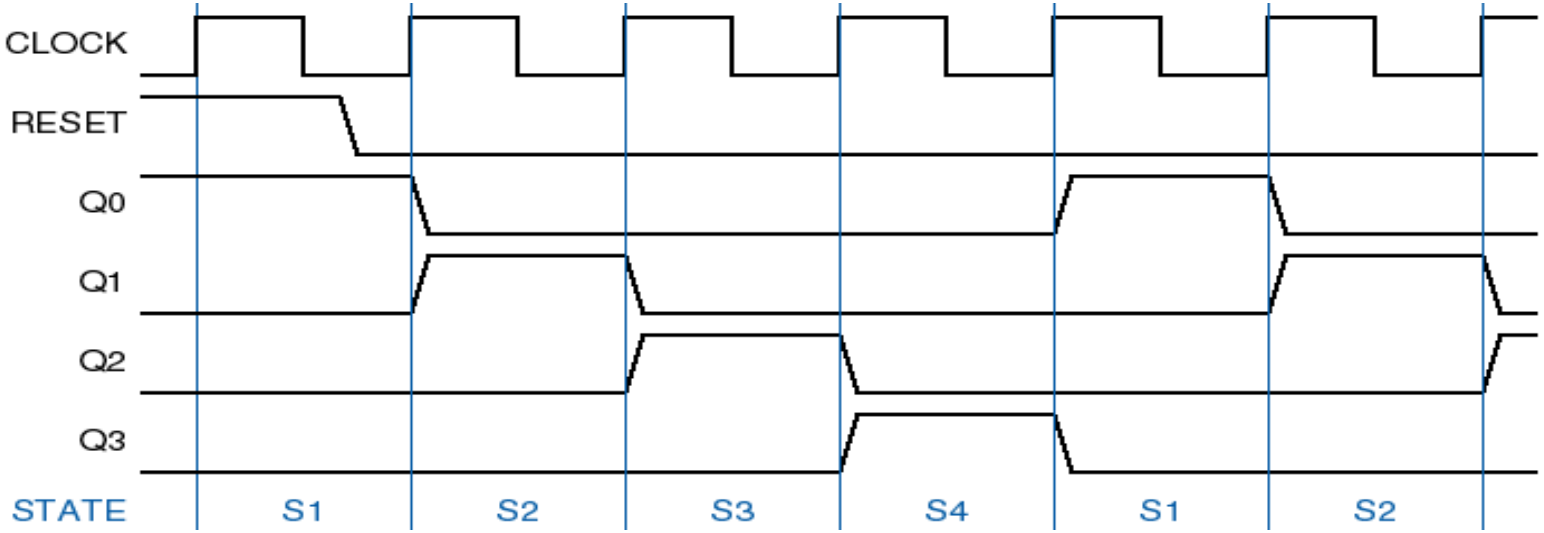


Conclusion：

For n-bit Ring counter, n normal states, 2^n-n abnormal states



4-bit Ring counter



Ring counter application: its states appear in 1-out-of-n decoded form directly on the flip-flop outputs. Furthermore, these outputs are "glitch free".