

# Introduction to Digital Systems

## Part III (Sequential Components)

2022/2023

Sequential Synchronous  
Modules/Blocks

Multibit registers, Counters and  
Shift-registers

Arnaldo Oliveira, Augusto Silva, Ioulia Skliarova

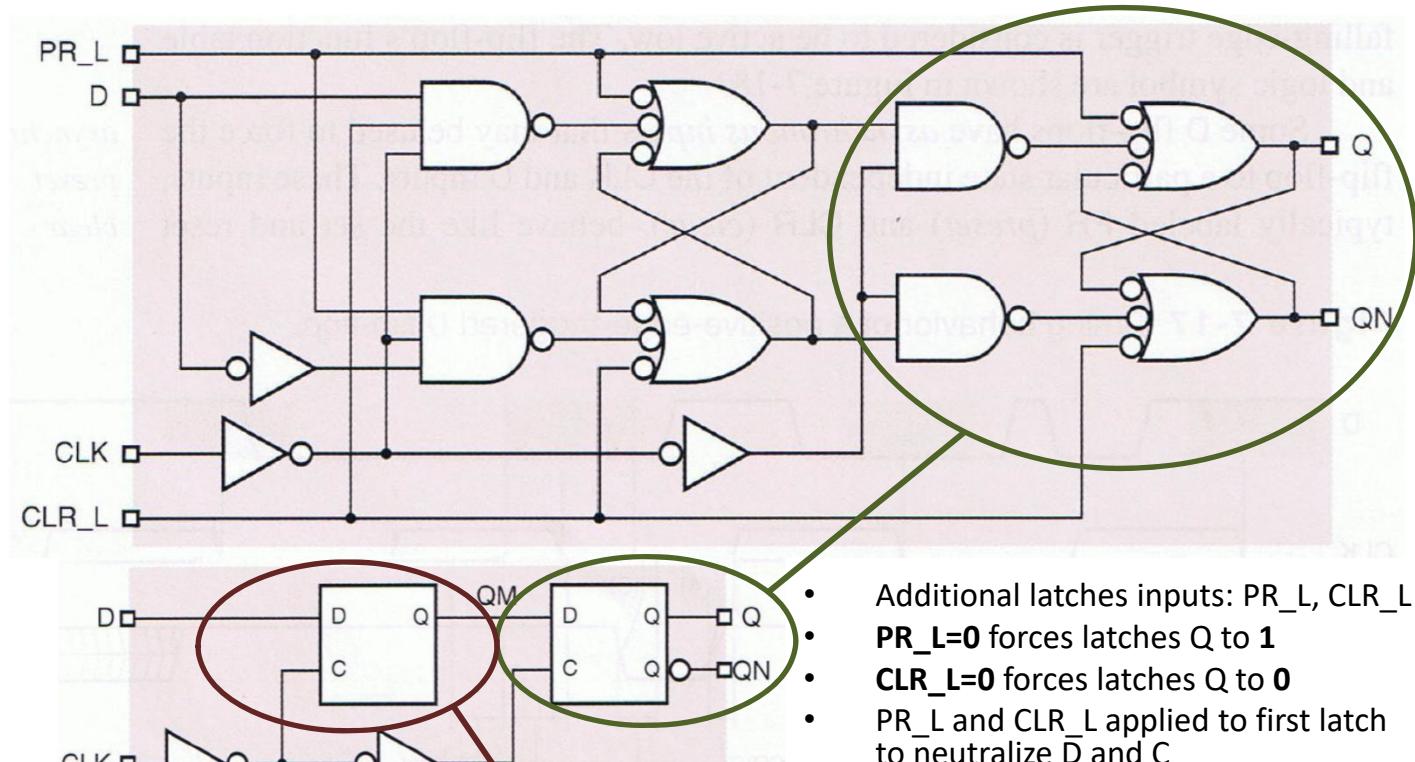
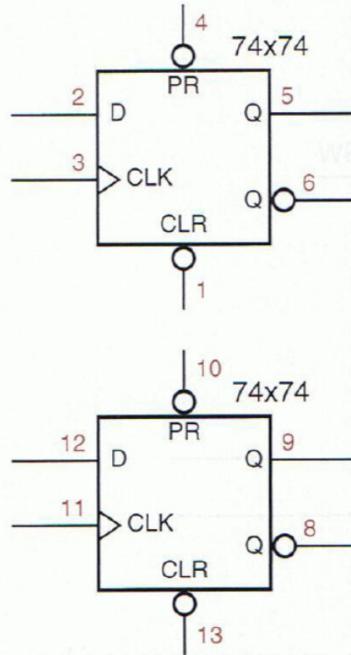
# Lecture Contents

- Single and multibit registers
  - Flip-flop based
  - Latch based
- Binary counter design
  - Up counter synthesis
  - Up/down counter synthesis
- Standard counters
  - Free running
  - Constrained counting
  - Cascading counters
- Shift-registers
  - Applications
  - Serial-in, serial-out
  - Serial-in, parallel-out
  - Parallel-in, serial-out
  - Parallel-in , parallel-out
  - Universal (load, shift, hold)

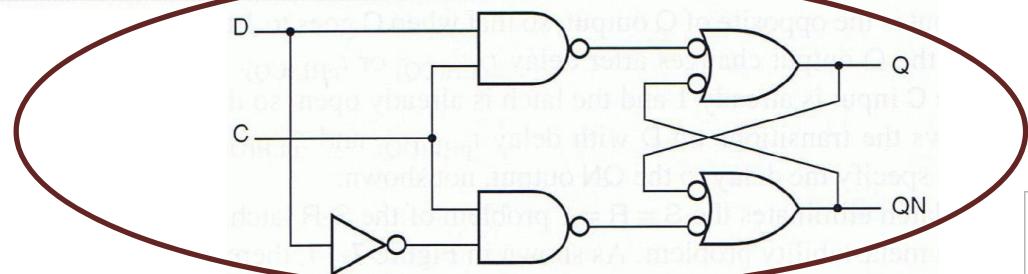
Some figures and content extracted from: John F. Wakerly, “Digital Design – Principles and Practices”, 4 ed., Pearson – Prentice Hall, 2006 (chapter 7). Reading chapter 7 (4<sup>th</sup> ed.) or chapter 11 (5<sup>th</sup> ed.) is highly recommended.



# 74x74 – Dual Positive-edge-triggered D Flip-flop w/ Asynchronous Preset (Set) and Clear (Reset)

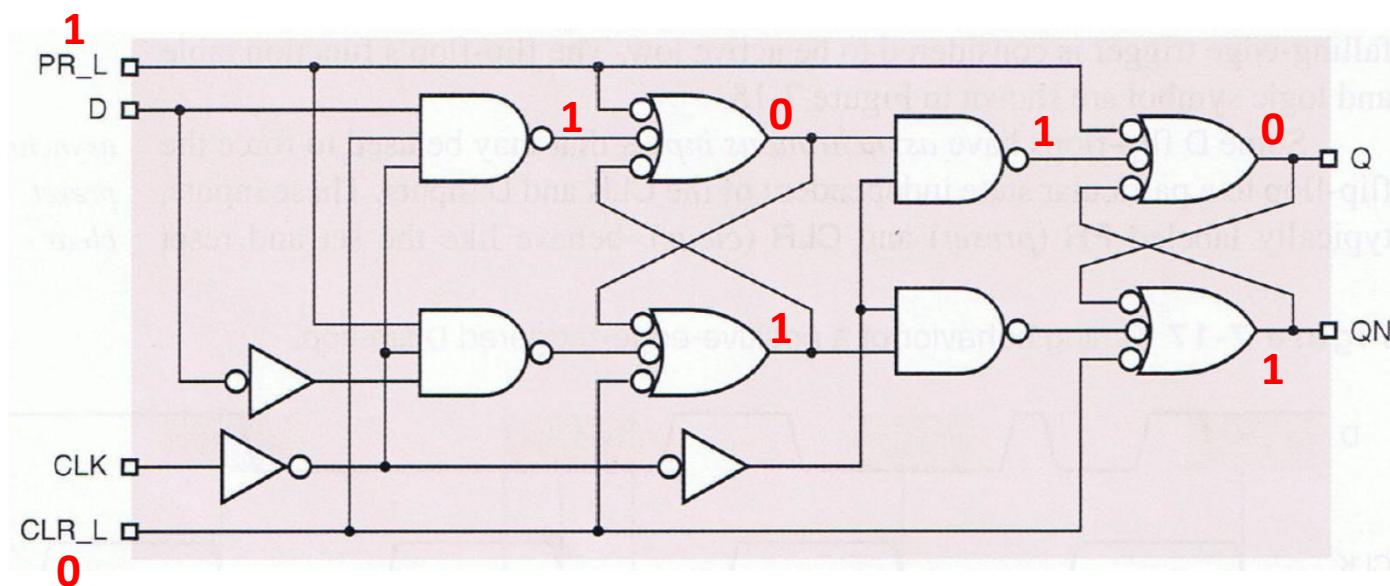


Power supply pins:  
14 (Vcc)  
7 (GND)



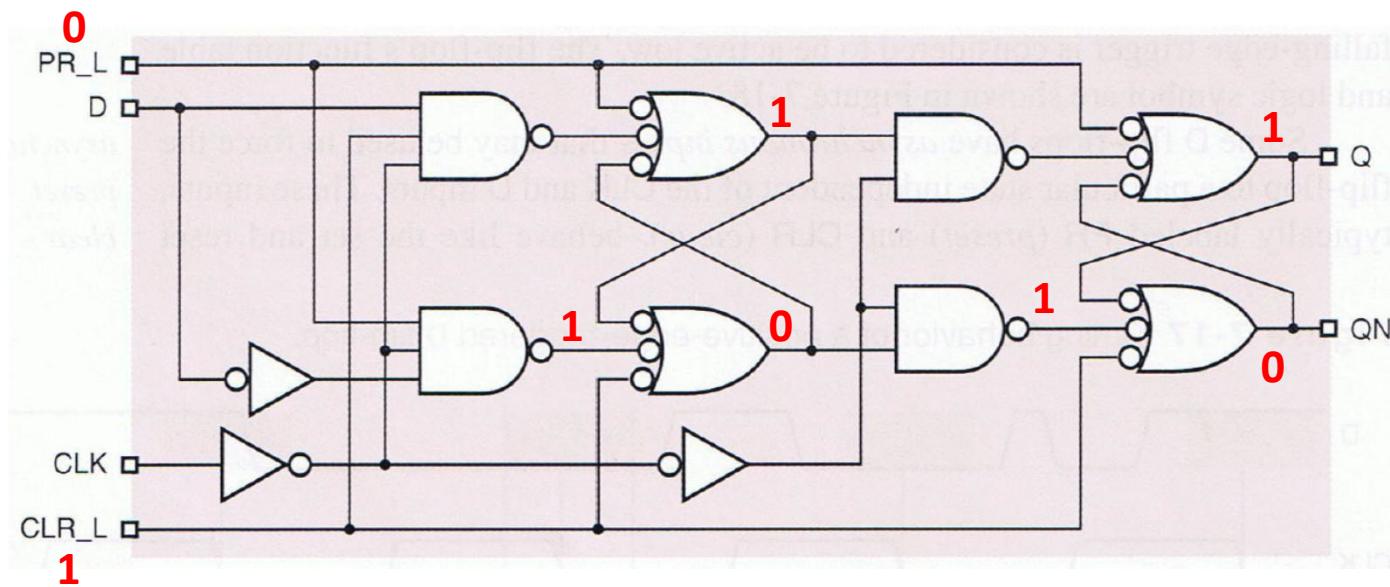
# 74x74 – Dual Positive-edge-triggered D Flip-flop w/ Asynchronous Preset (Set) and Clear (Reset)

Forcing>Loading 0 (Clear / Reset) at Q output

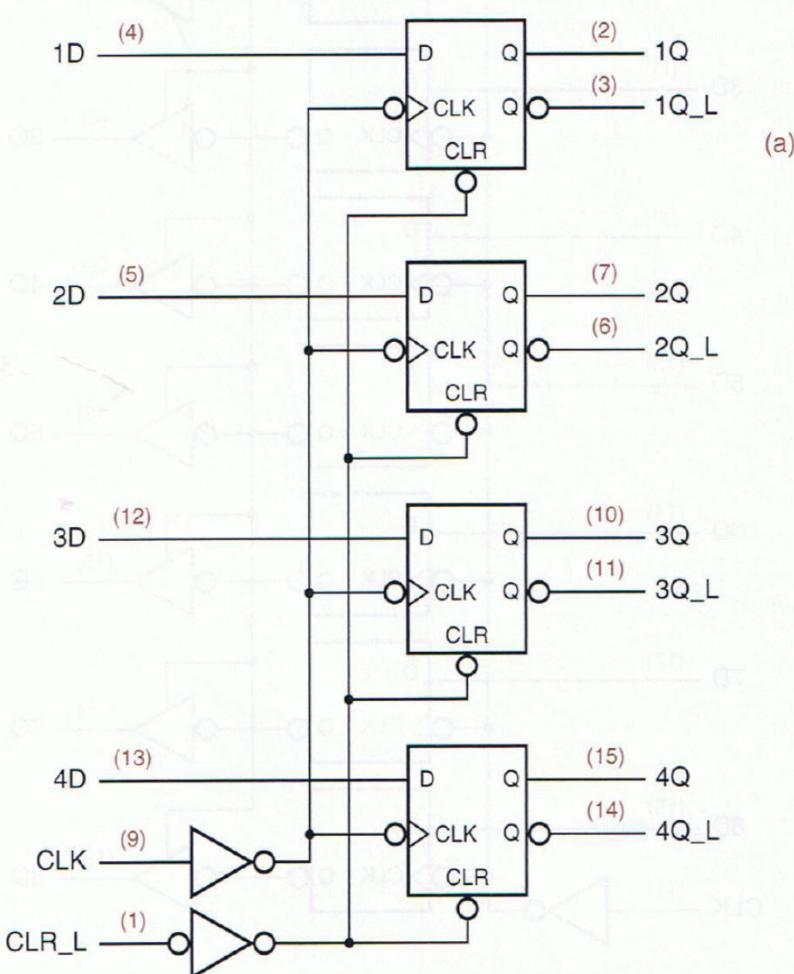


# 74x74 – Dual Positive-edge-triggered D Flip-flop w/ Asynchronous Preset (Set) and Clear (Reset)

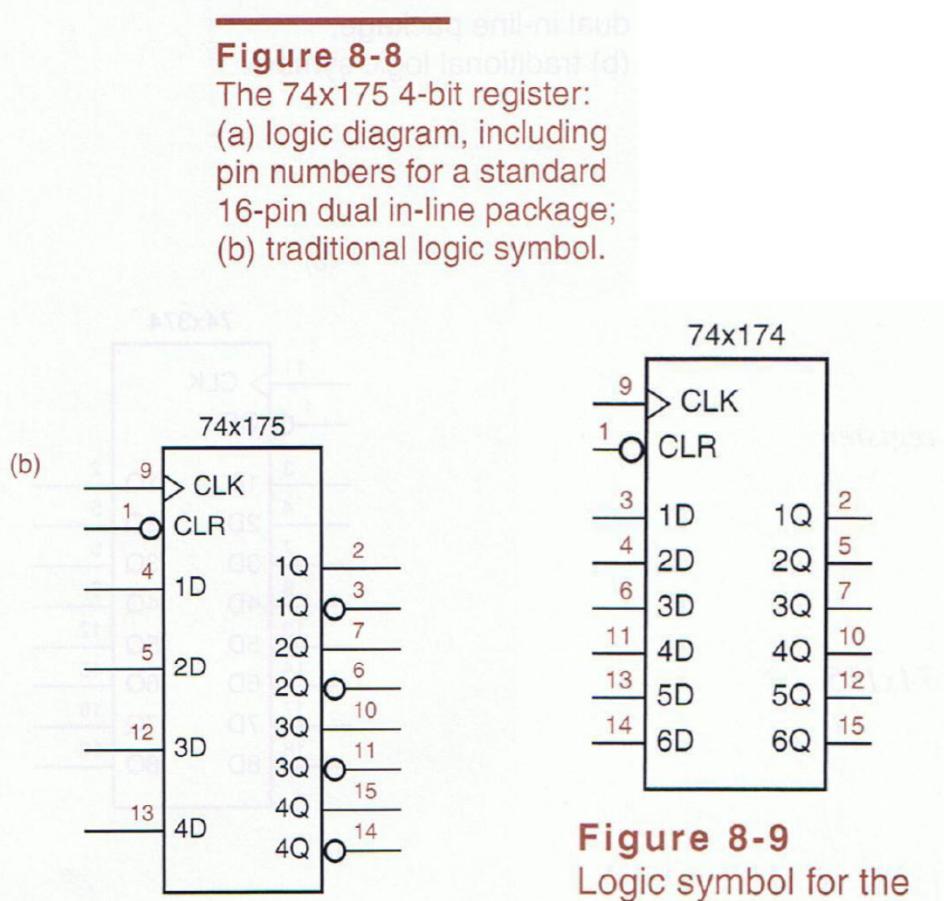
Forcing>Loading 1 (Preset / Set) at Q output



# 74x175/174 4-bit Flip-flop Based Register



(a)



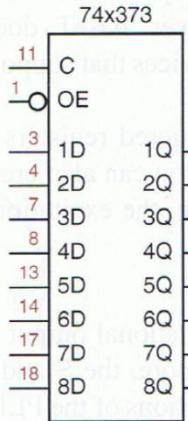
**Figure 8-9**  
Logic symbol for the  
74x174 6-bit register.

# 74x373/74x374 8-bit Latch/Register with 3-State Outputs

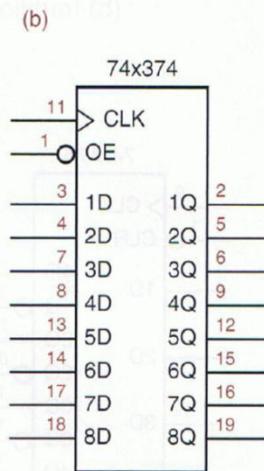
**Figure 8-10**

The 74x374 8-bit register:  
 (a) logic diagram, including pin numbers for a standard 20-pin dual in-line package;  
 (b) traditional logic symbol.

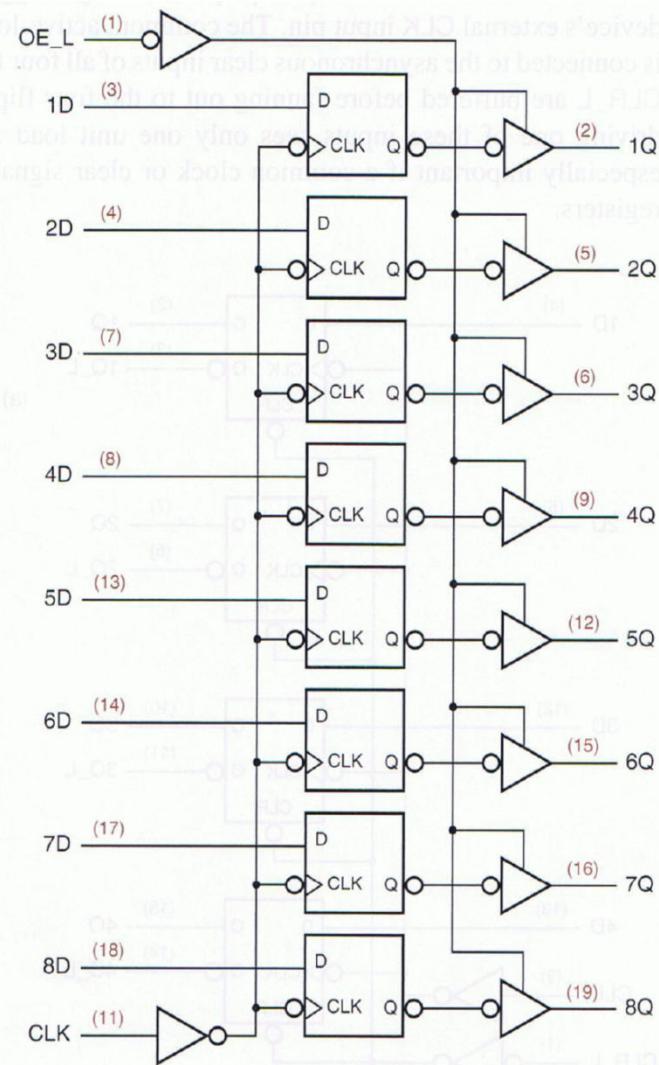
Latch – level triggered



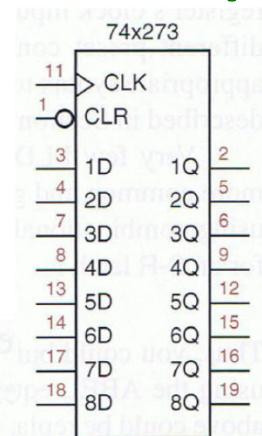
**Figure 8-11**  
Logic symbol for the  
74x373 8-bit latch.



Flip-flop – edge triggered

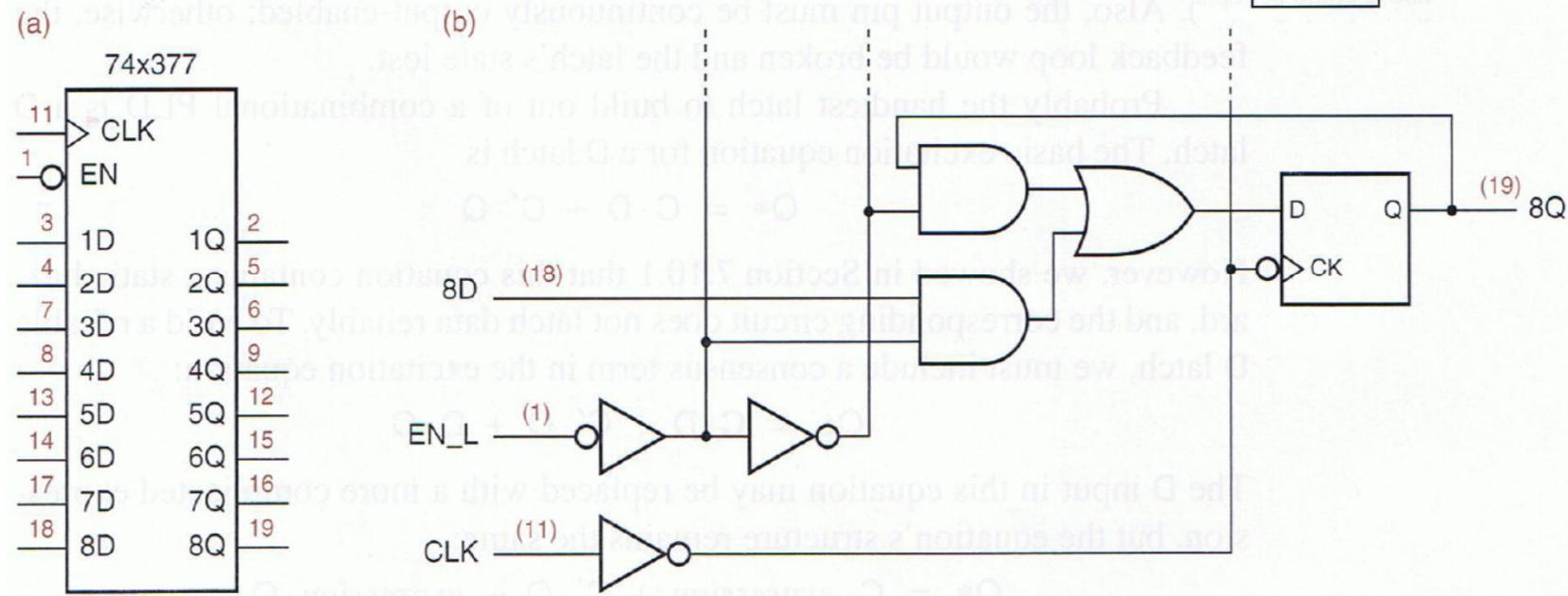


# 74x273/74x377 8-bit Flip-flop Based Register with Clear/Enable Input



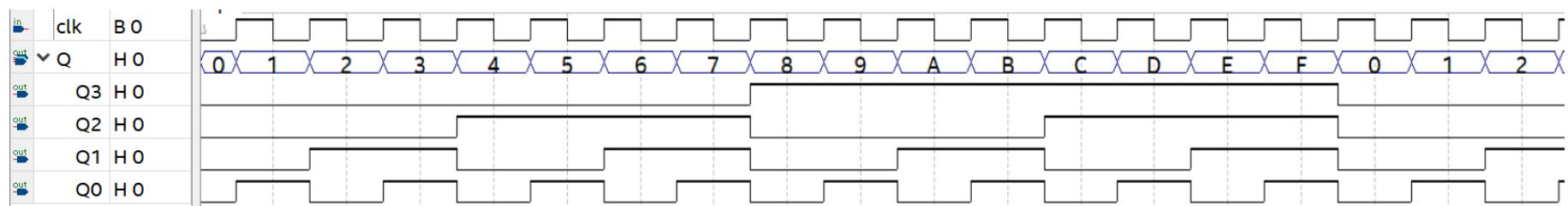
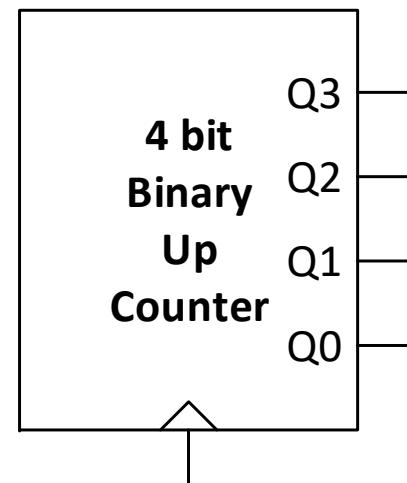
**Figure 8-12**  
Logic symbol for the  
74x273 8-bit register.

**Figure 8-13** The 74x377 8-bit register with gated clock:  
(a) logic symbol; (b) logical behavior of one bit.



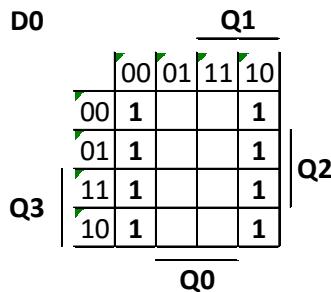
# 4 bit Binary Up Counter Operation

Decimal	Q3 Q2 Q1 Q0	Q3* Q2* Q1* Q0*				Decimal
		D3	D2	D1	D0	
0	0000	0001				1
1	0001	0010				2
2	0010	0011				3
3	0011	0100				4
4	0100	0101				5
5	0101	0110				6
6	0110	0111				7
7	0111	1000				8
8	1000	1001				9
9	1001	1010				10
10	1010	1011				11
11	1011	1100				12
12	1100	1101				13
13	1101	1110				14
14	1110	1111				15
15	1111	0000				0

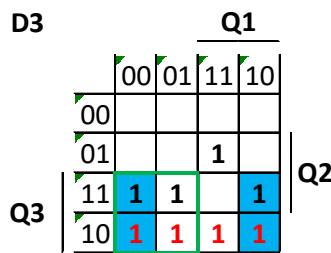


# 4 bit Binary Up Counter Synthesis

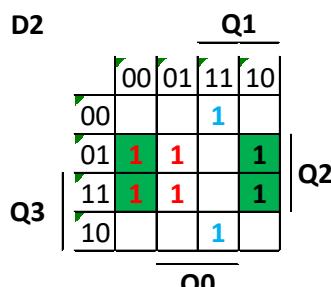
Decimal	Q3 Q2 Q1 Q0	Q3* Q2* Q1* Q0*				Decimal
		D3	D2	D1	D0	
0	0000	0001				1
1	0001	0010				2
2	0010	0011				3
3	0011	0100				4
4	0100	0101				5
5	0101	0110				6
6	0110	0111				7
7	0111	1000				8
8	1000	1001				9
9	1001	1010				10
10	1010	1011				11
11	1011	1100				12
12	1100	1101				13
13	1101	1110				14
14	1110	1111				15
15	1111	0000				0



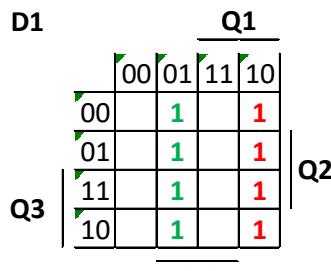
$$D_0 = \overline{Q_0}$$



$$D_3 = Q_3 \cdot \overline{Q_2} + Q_3 \cdot \overline{Q_1} + Q_3 \cdot \overline{Q_0} + \overline{Q_3} \cdot Q_2 \cdot Q_1 \cdot Q_0$$



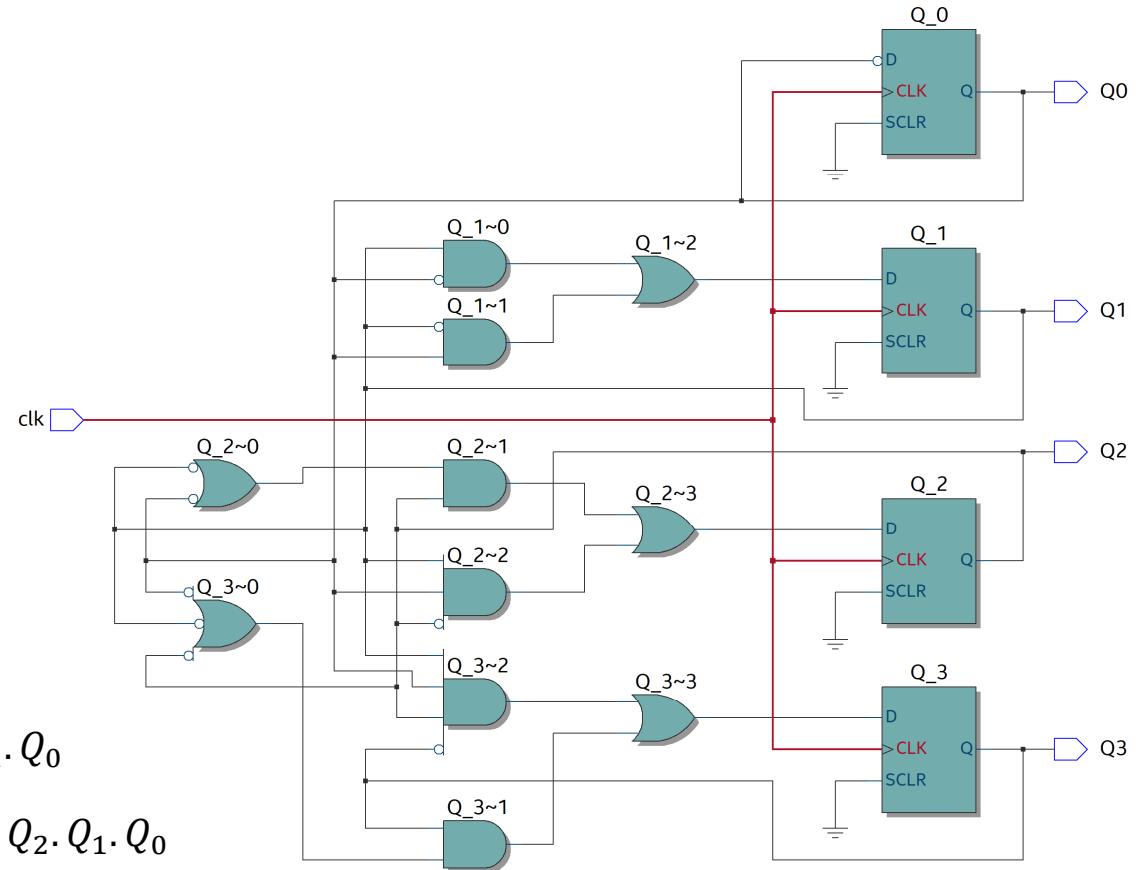
$$D_2 = Q_2 \cdot \overline{Q_1} + Q_2 \cdot \overline{Q_0} + \overline{Q_2} \cdot Q_1 \cdot Q_0$$



$$D_1 = Q_1 \cdot \overline{Q_0} + \overline{Q_1} \cdot Q_0$$

Minimized  
Excitation  
equations

# 4 bit Binary Up Counter Logic Diagram



$$D_0 = \overline{Q}_0$$

$$D_1 = Q_1 \cdot \overline{Q}_0 + \overline{Q}_1 \cdot Q_0$$

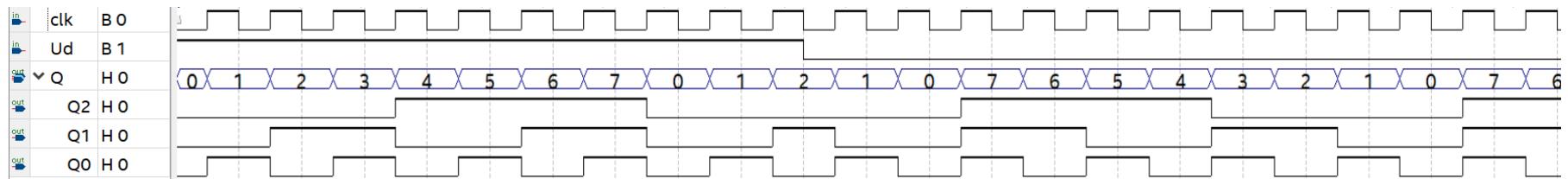
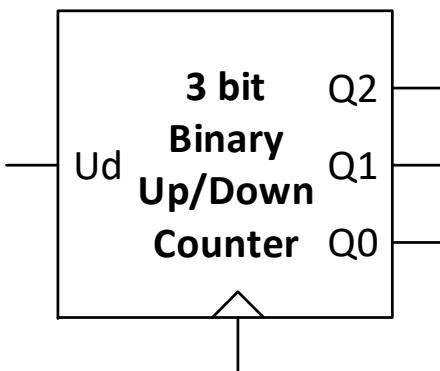
$$D_2 = Q_2 \cdot \overline{Q}_1 + Q_2 \cdot \overline{Q}_0 + \overline{Q}_2 \cdot Q_1 \cdot Q_0$$

$$D_3 = Q_3 \cdot \overline{Q}_2 + Q_3 \cdot \overline{Q}_1 + Q_3 \cdot \overline{Q}_0 + \overline{Q}_3 \cdot Q_2 \cdot Q_1 \cdot Q_0$$

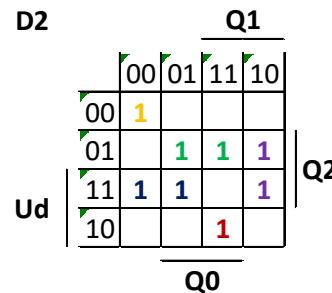
In a real circuit, flip-flops reset/clear must be connected to a reset input!

# 3 bit Binary Up/Down Counter Operation

Decimal	Ud	Q2 Q1 Q0	Q2*	Q1*	Q0*	Decimal
			D2	D1	D0	
0	0	000		111		7
1	0	001		000		0
2	0	010		001		1
3	0	011		010		2
4	0	100		011		3
5	0	101		100		4
6	0	110		101		5
7	0	111		110		6
0	1	000		001		1
1	1	001		010		2
2	1	010		011		3
3	1	011		100		4
4	1	100		101		5
5	1	101		110		6
6	1	110		111		7
7	1	111		000		0



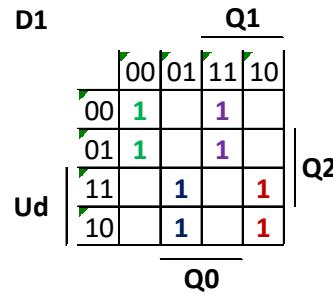
# 3 bit Binary Up/Down Counter Synthesis



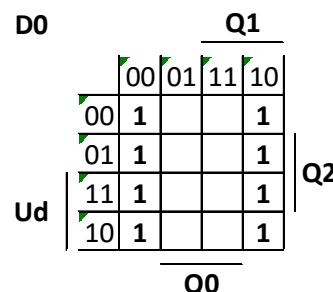
$$D_2 = \overline{Ud} \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0} + \overline{Ud} \cdot Q_2 \cdot Q_0 + Ud \cdot Q_2 \cdot \overline{Q_1} + Q_2 \cdot Q_1 \cdot \overline{Q_0} + Ud \cdot \overline{Q_2} \cdot Q_1 \cdot Q_0$$

Minimized  
Excitation  
equations

Decimal	Ud	Q2 Q1 Q0	Q2* Q1* Q0*		Decimal
			D2	D1 D0	
0	0	000		111	7
1	0	001		000	0
2	0	010		001	1
3	0	011		010	2
4	0	100		011	3
5	0	101		100	4
6	0	110		101	5
7	0	111		110	6
0	1	000	001		1
1	1	001	010		2
2	1	010	011		3
3	1	011	100		4
4	1	100	101		5
5	1	101	110		6
6	1	110	111		7
7	1	111	000		0



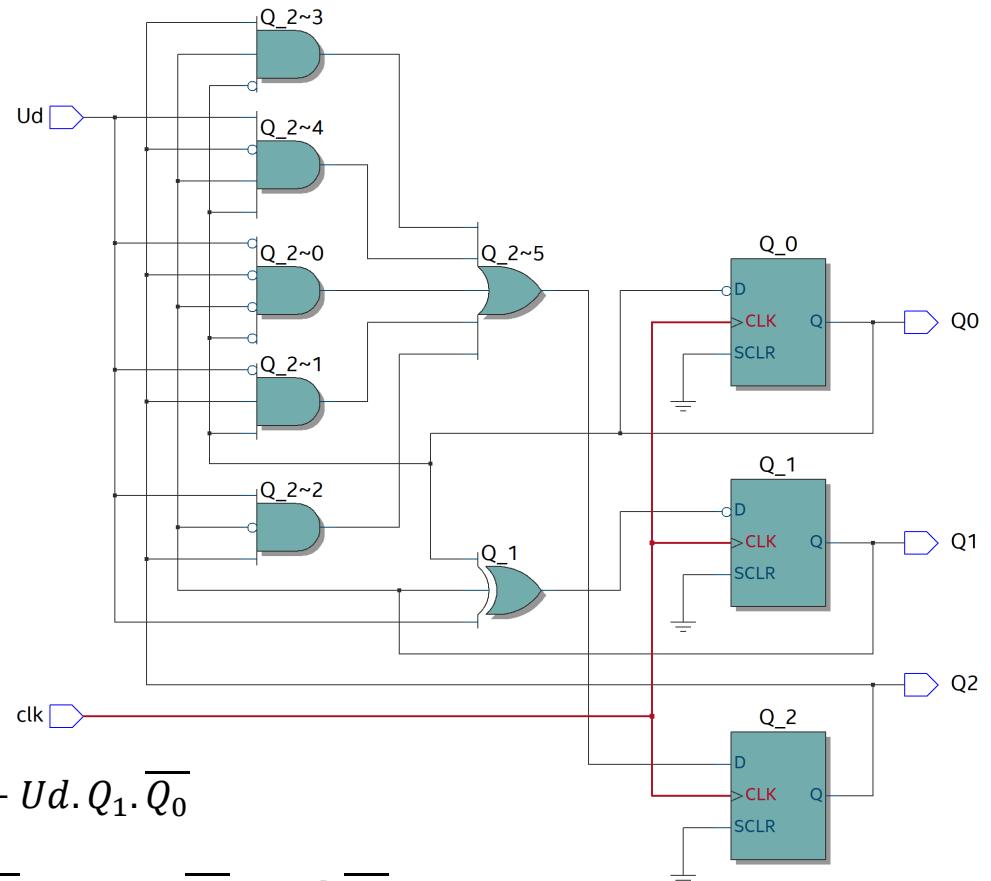
$$D_1 = \overline{Ud} \cdot \overline{Q_1} \cdot \overline{Q_0} + \overline{Ud} \cdot Q_1 \cdot Q_0 + Ud \cdot \overline{Q_1} \cdot Q_0 + Ud \cdot Q_1 \cdot \overline{Q_0}$$



$$D_0 = \overline{Q_0}$$

# 3 bit Binary Up/Down Counter

## Logic Diagram



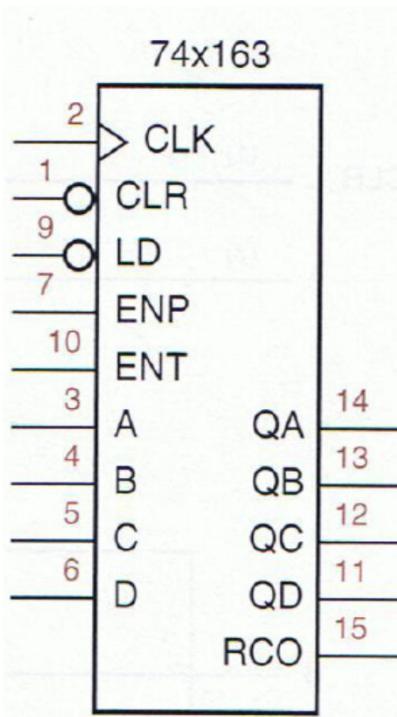
$$D_0 = \overline{Q}_0$$

$$D_1 = \overline{Ud} \cdot \overline{Q}_1 \cdot \overline{Q}_0 + \overline{Ud} \cdot Q_1 \cdot Q_0 + Ud \cdot \overline{Q}_1 \cdot Q_0 + Ud \cdot Q_1 \cdot \overline{Q}_0$$

$$D_2 = \overline{Ud} \cdot \overline{Q}_2 \cdot \overline{Q}_1 \cdot \overline{Q}_0 + \overline{Ud} \cdot Q_2 \cdot Q_0 + Ud \cdot Q_2 \cdot \overline{Q}_1 + Q_2 \cdot Q_1 \cdot \overline{Q}_0 + Ud \cdot \overline{Q}_2 \cdot Q_1 \cdot Q_0$$

In a real circuit, flip-flops reset/clear must be connected to a reset input!

# 74x163 Standard 4-bit Binary Up Counter



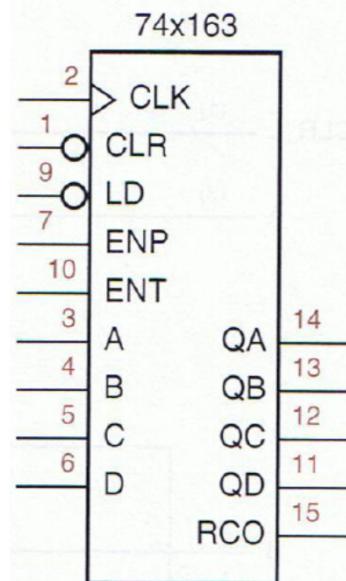
Port	Meaning
CLK	Clock input
CLR	Synchronous clear, active low
LD	Synchronous parallel load enable, active low
ENP, ENT	Enable input
A – D	Parallel load data value
QA – QD	Counter output value
RCO	Ripple counter output (for cascading)

**Figure 8-27**  
Traditional logic  
symbol for  
the 74x163.

# 74x163 Counter Operation

Clear  
Load  
Freeze

Count

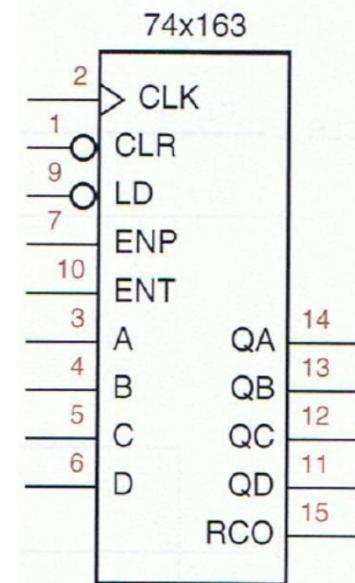
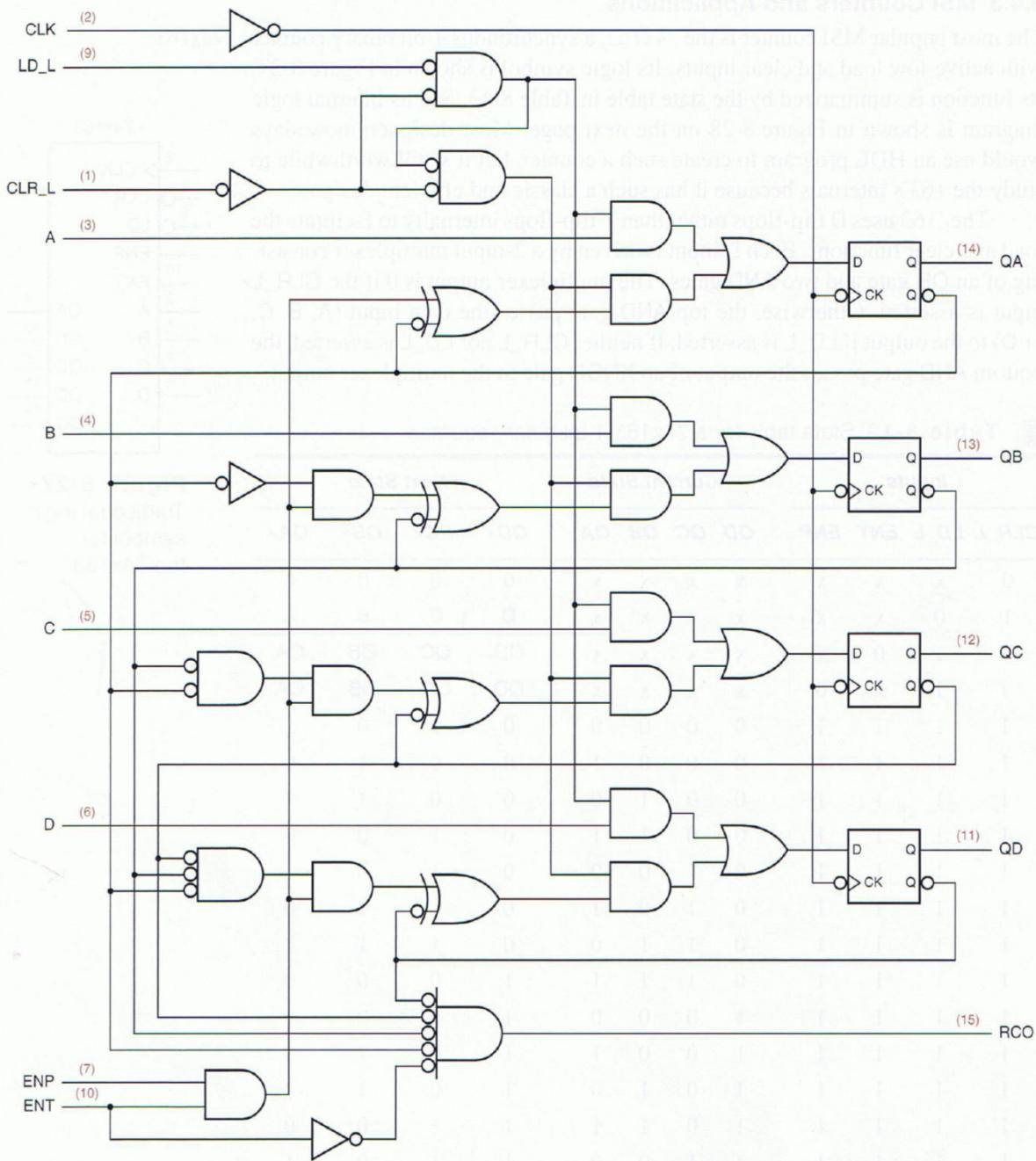


**Figure 8-27**  
Traditional logic  
symbol for  
the 74x163.

**Table 8-13** State table for a 74x163 4-bit binary counter.

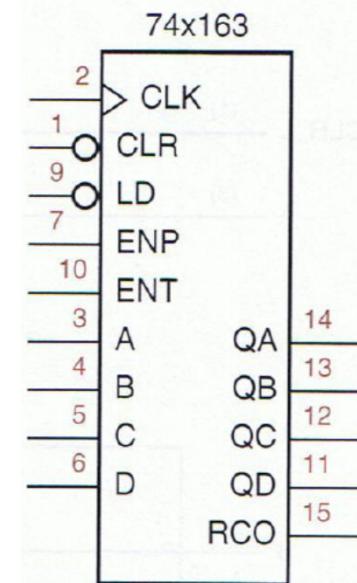
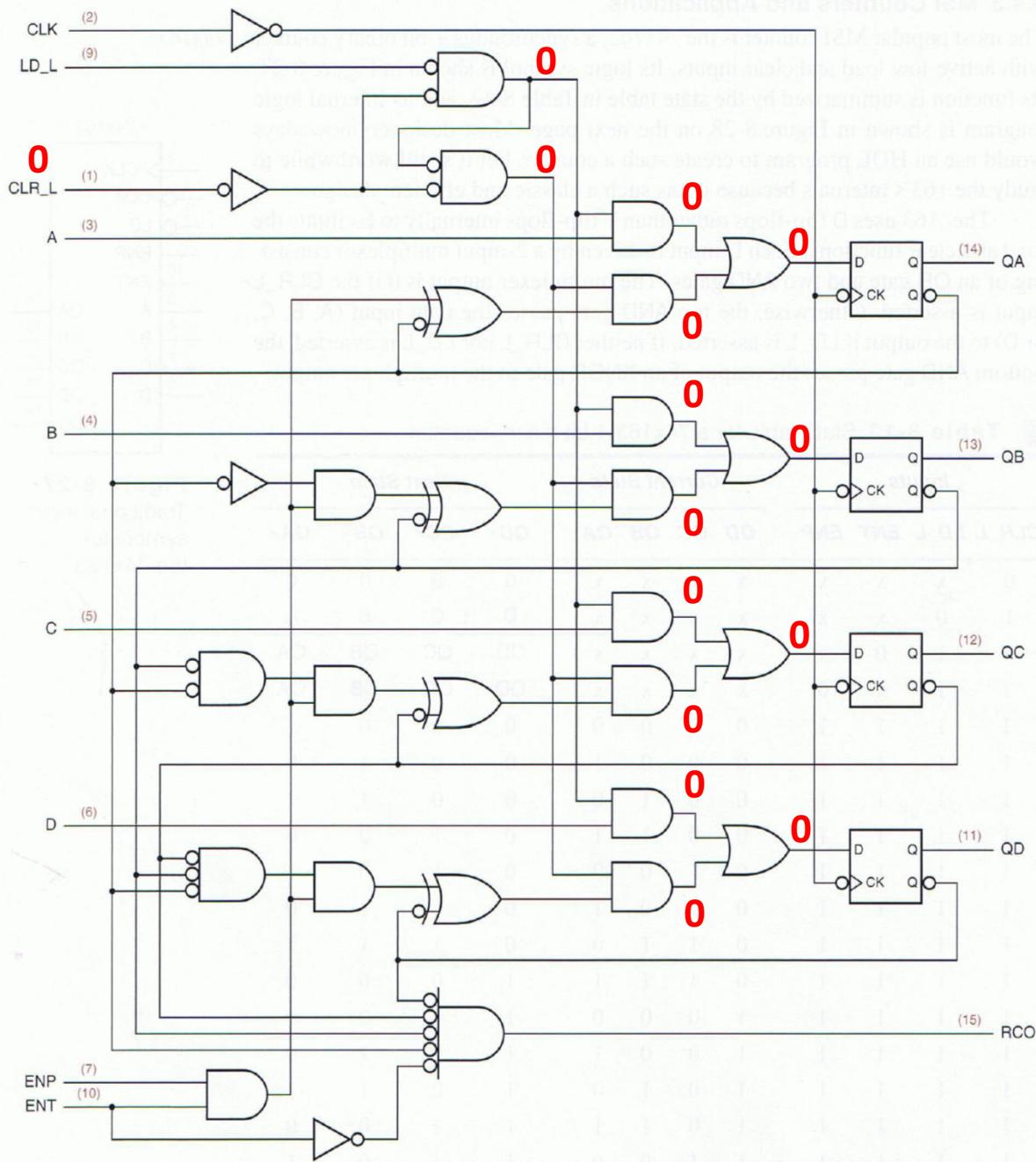
Inputs				Current State				Next State			
CLR_L	LD_L	ENT	ENP	QD	QC	QB	QA	QD*	QC*	QB*	QA*
0	x	x	x	x	x	x	x	0	0	0	0
1	0	x	x	x	x	x	x	D	C	B	A
1	1	0	x	x	x	x	x	QD	QC	QB	QA
1	1	x	0	x	x	x	x	QD	QC	QB	QA
1	1	1	1	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	1	0	0	1	0
1	1	1	1	0	0	1	0	0	0	1	1
1	1	1	1	0	0	1	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	1
1	1	1	1	0	1	0	1	0	1	1	0
1	1	1	1	0	1	0	1	0	1	1	0
1	1	1	1	0	1	1	0	0	1	1	1
1	1	1	1	0	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0	1	0	0	1
1	1	1	1	1	0	0	0	1	0	0	1
1	1	1	1	1	0	0	1	1	0	1	0
1	1	1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0

# 74x163 Counter Internal Structure



**Figure 8-27**  
Traditional logic  
symbol for  
the 74x163.

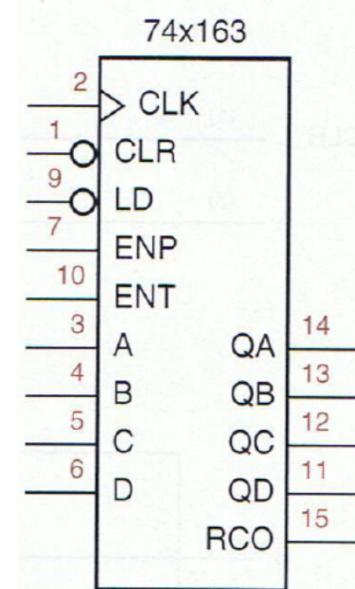
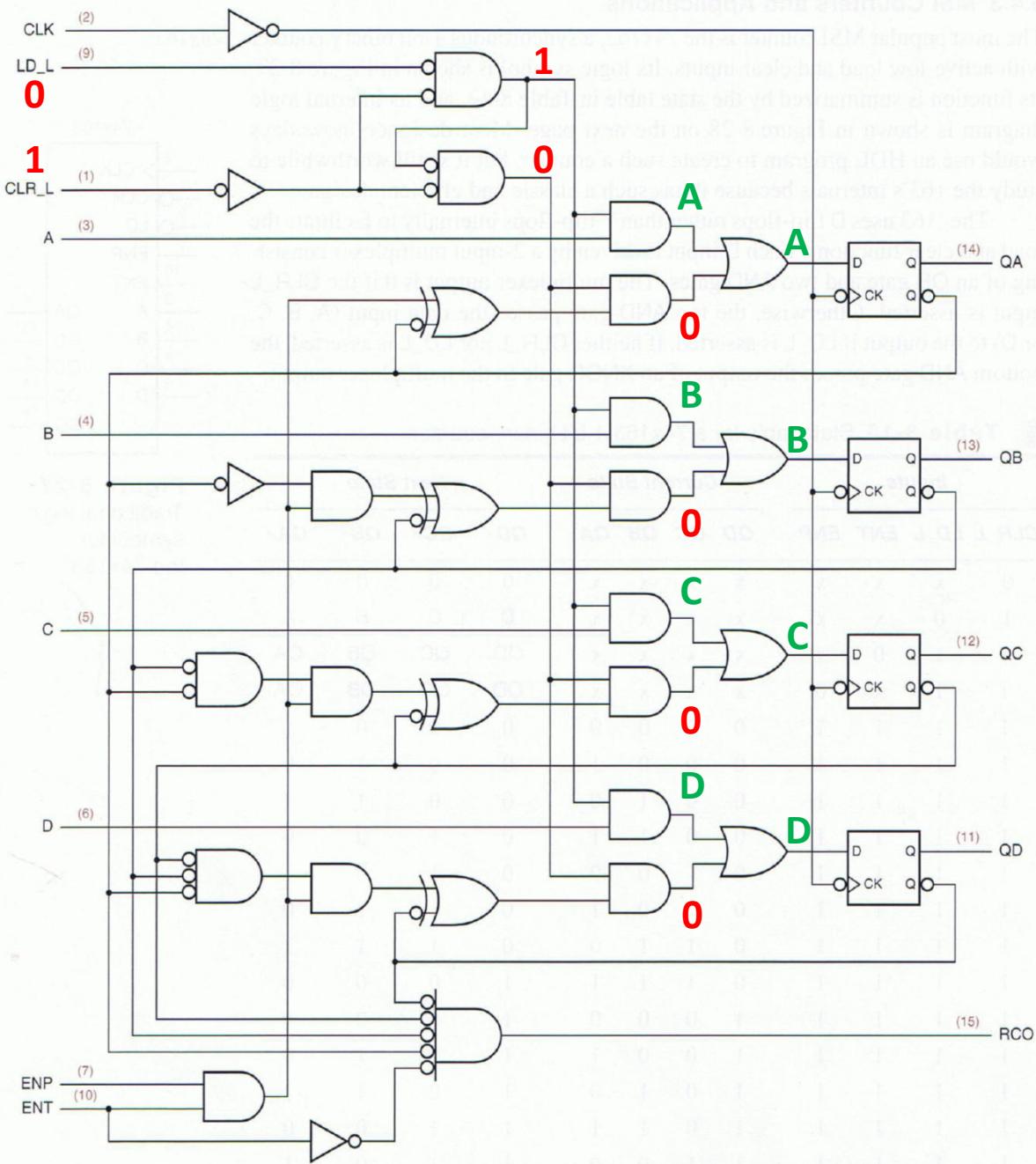
# 74x163 Clearing Count Value



**Figure 8-27**  
Traditional logic  
symbol for  
the 74x163.

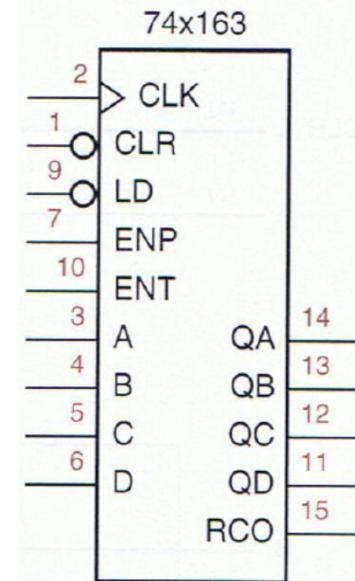
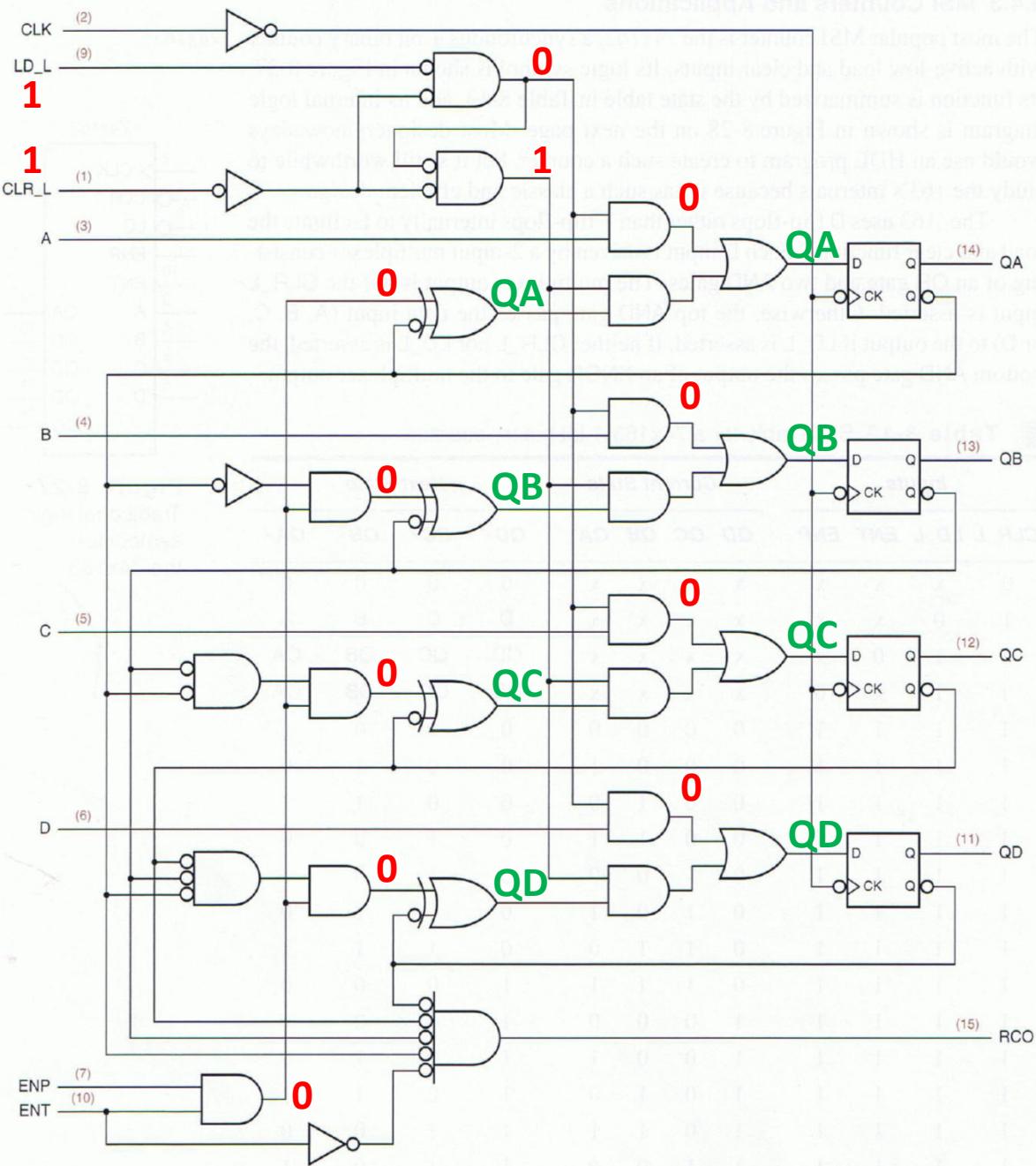
# 74x163

## Loading Parallel Data



**Figure 8-27**  
Traditional logic  
symbol for  
the 74x163.

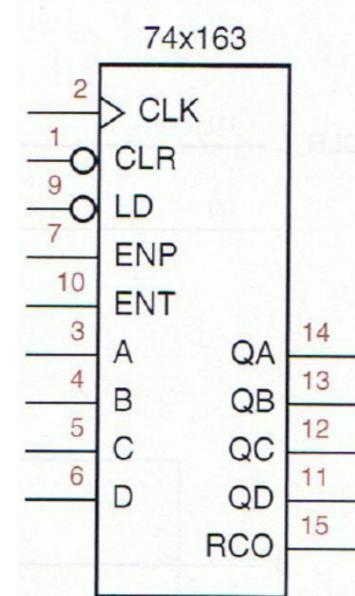
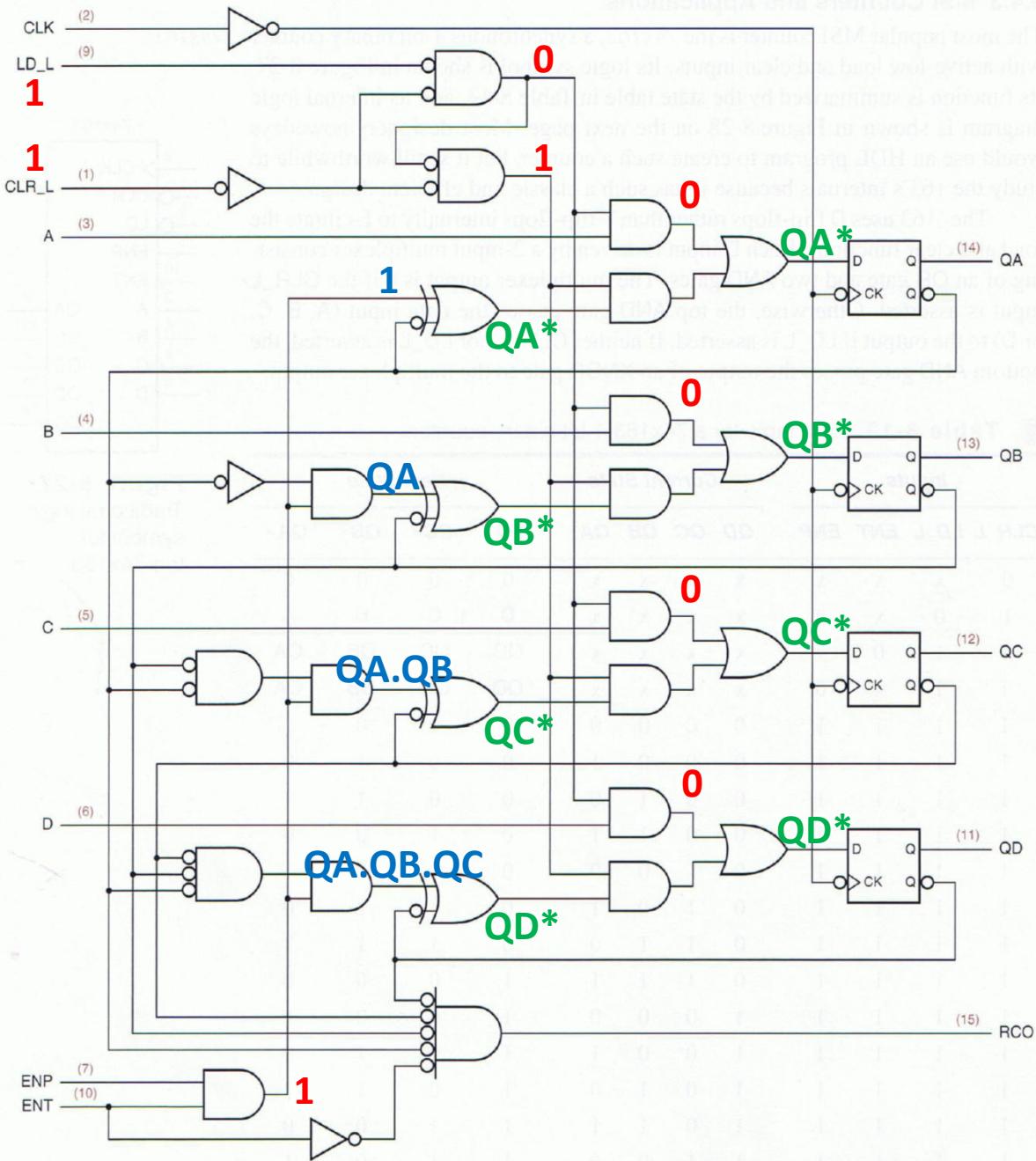
# 74x163 Freeze Counter Value



**Figure 8-27**  
Traditional logic  
symbol for  
the 74x163.

# 74x163

## Counting Up



**Figure 8-27**  
Traditional logic symbol for the 74x163.

# Free Running Counter

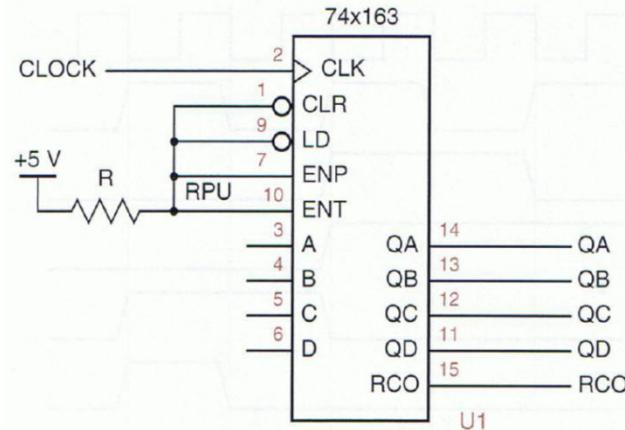
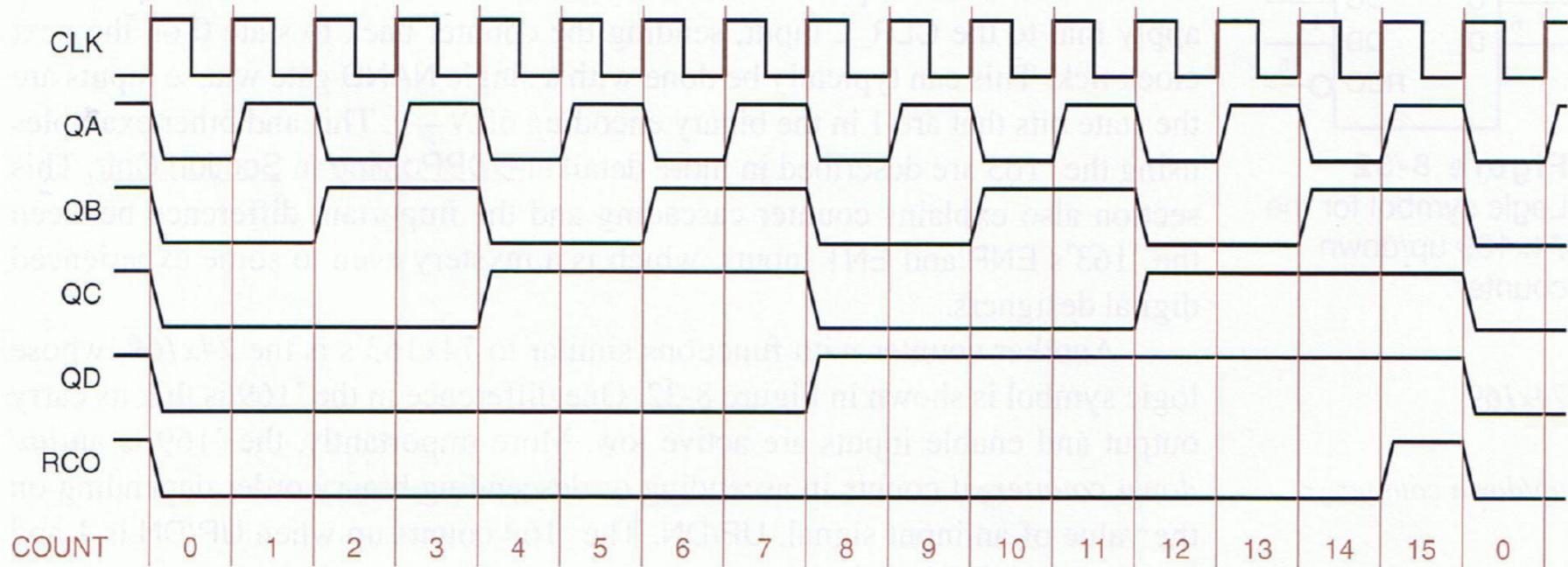


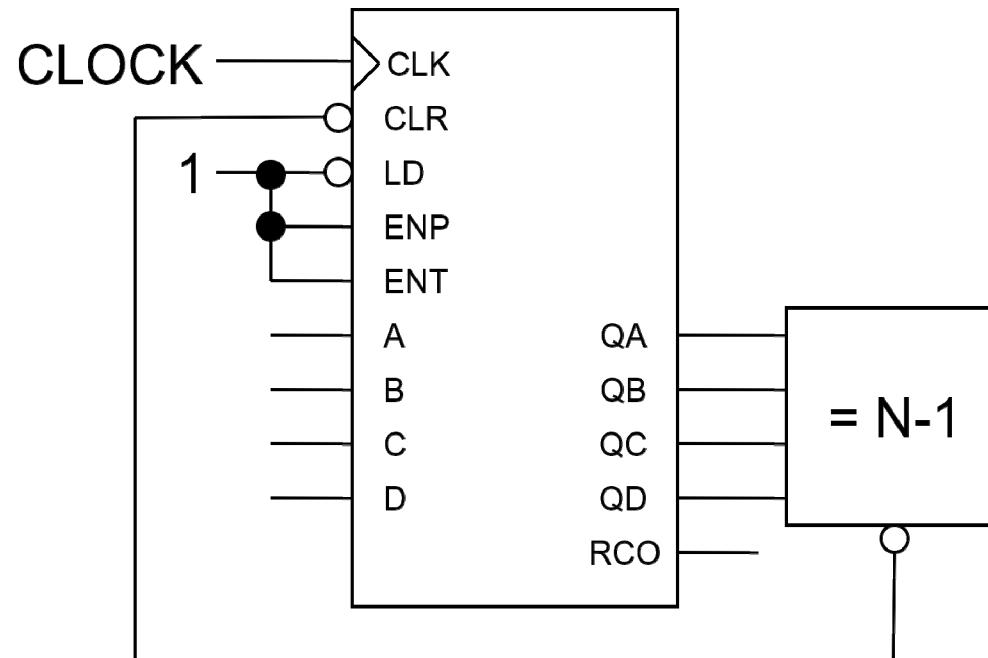
Figure 8-29  
Connections for the  
74x163 to operate in  
a free-running mode.

Figure 8-30 Clock and output waveforms for a free-running divide-by-16 counter.



# Modulo “N” Counter Based on 74x163 (with $1 \leq "N" \leq 16$ )

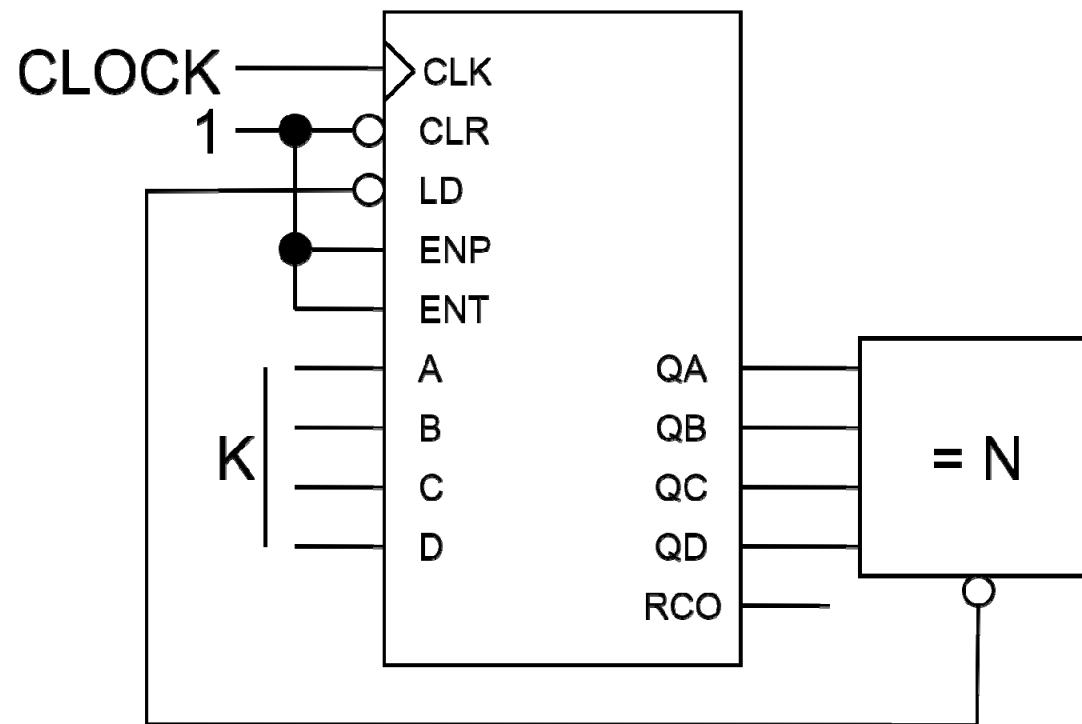
Counts from 0 up to “N-1” and wrap around



How to implement efficiently the comparator with the constant N-1?

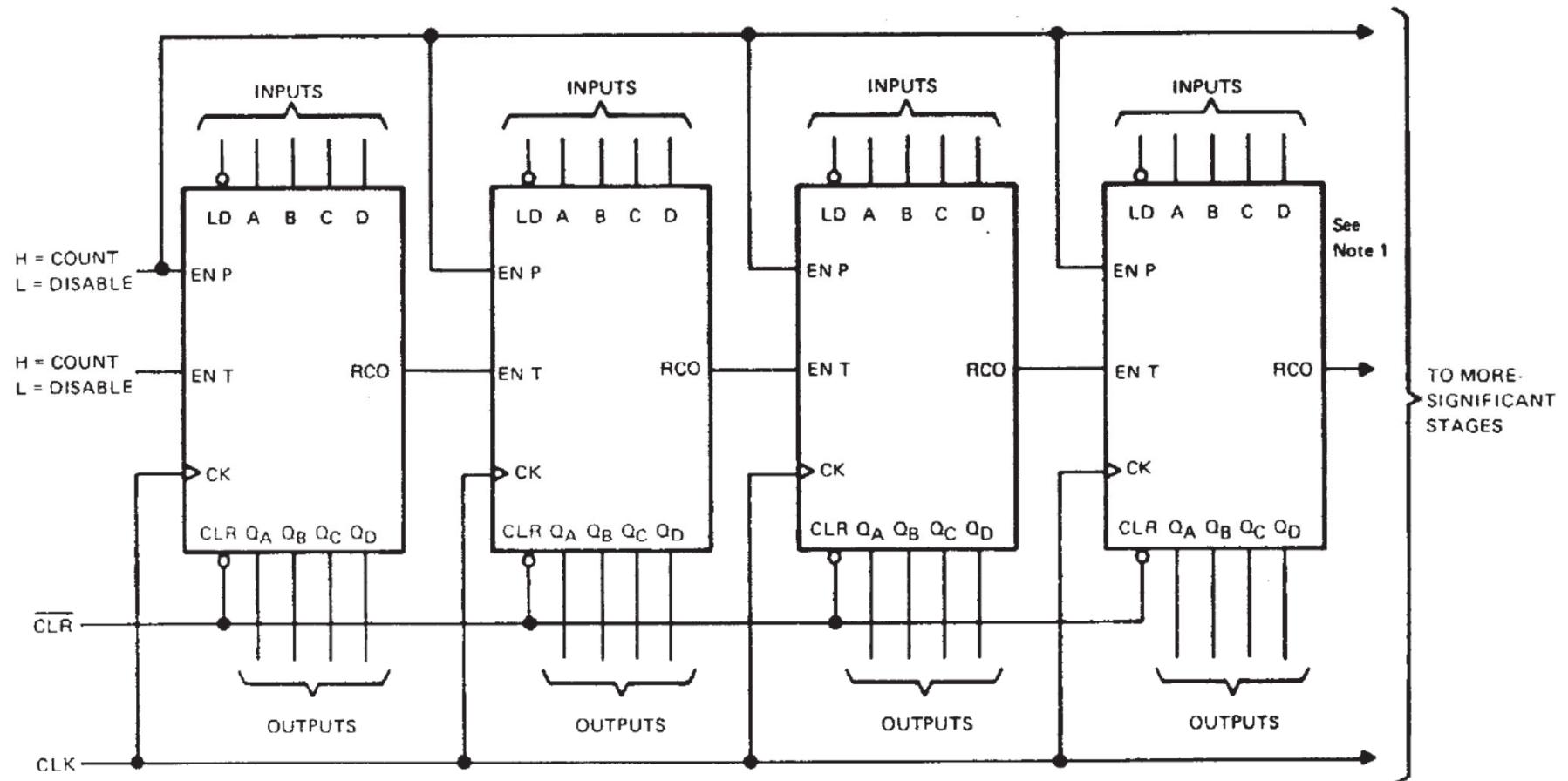
# Constrained “K” to “N” Counter Based on 74x163 (with “N”>“K”)

Counts from “K” up to “N” and wrap around to  
“K”

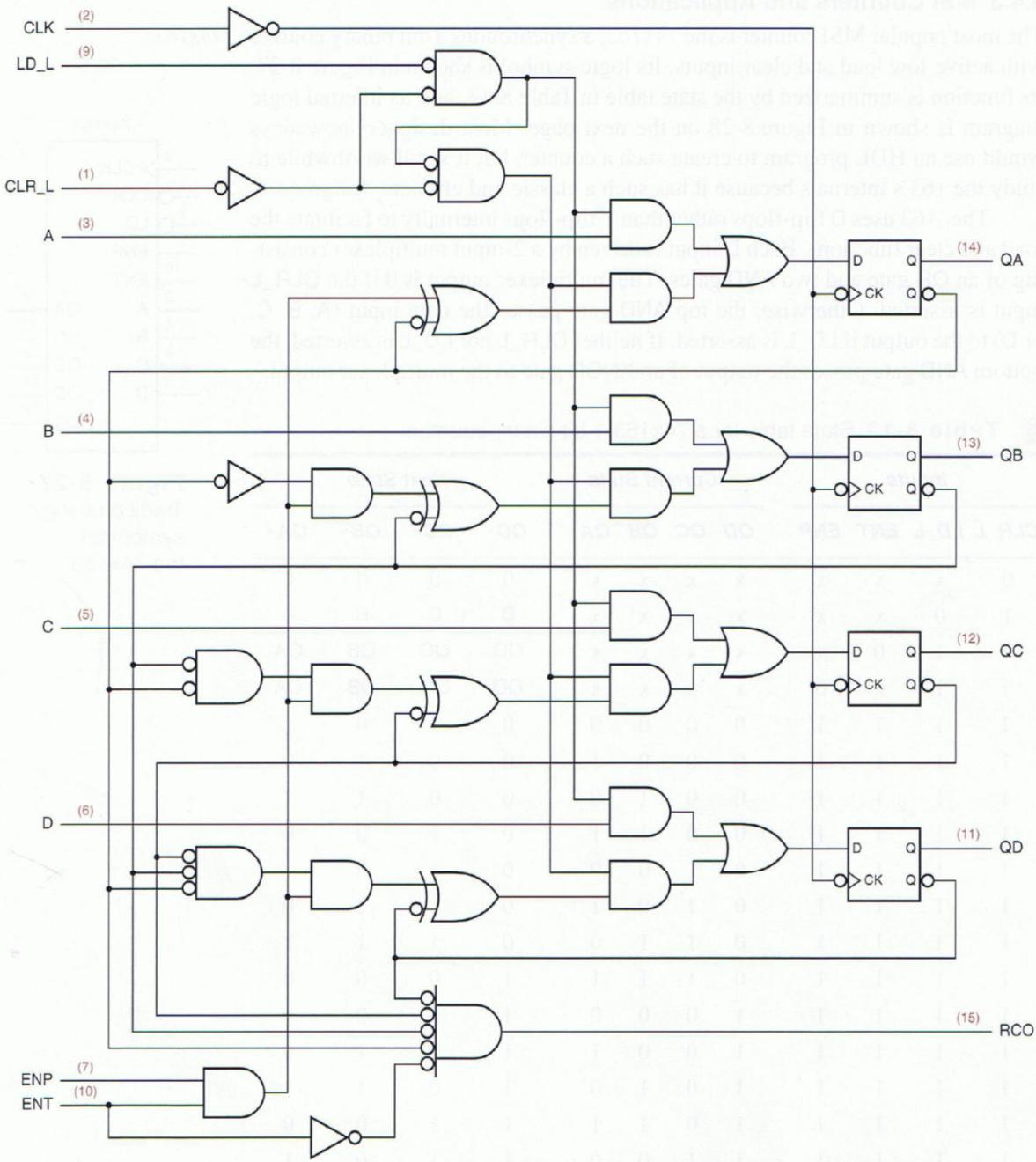


How to implement efficiently the comparator with the constant N?

# Cascading 74x163 Counters



# Maximum Operating Frequency



$$T_{\min} = ?$$

$$f_{\max} = ?$$

Consider:

$$T_{pHL} = 6 \text{ ns}$$

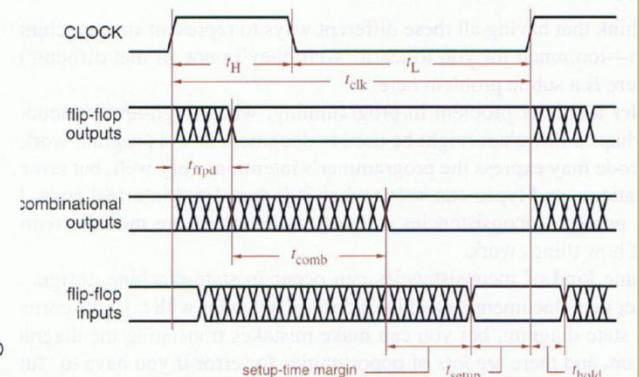
$$T_{plH} = 5 \text{ ns}$$

$$T_{\text{setup}} = 4 \text{ ns}$$

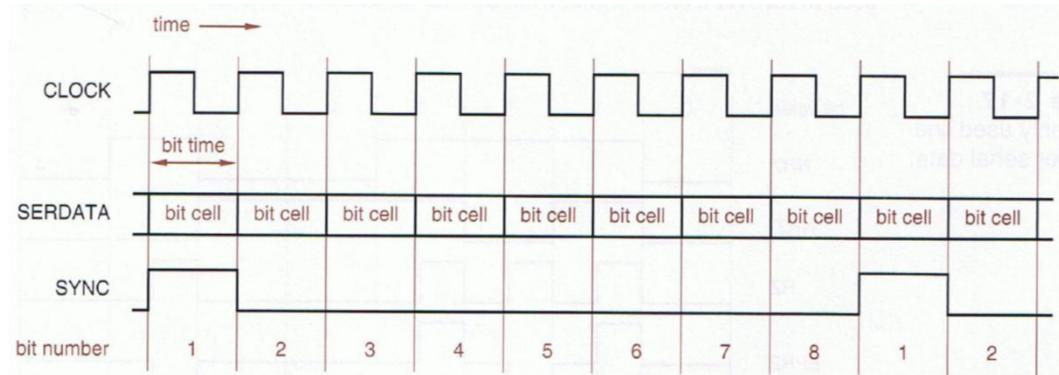
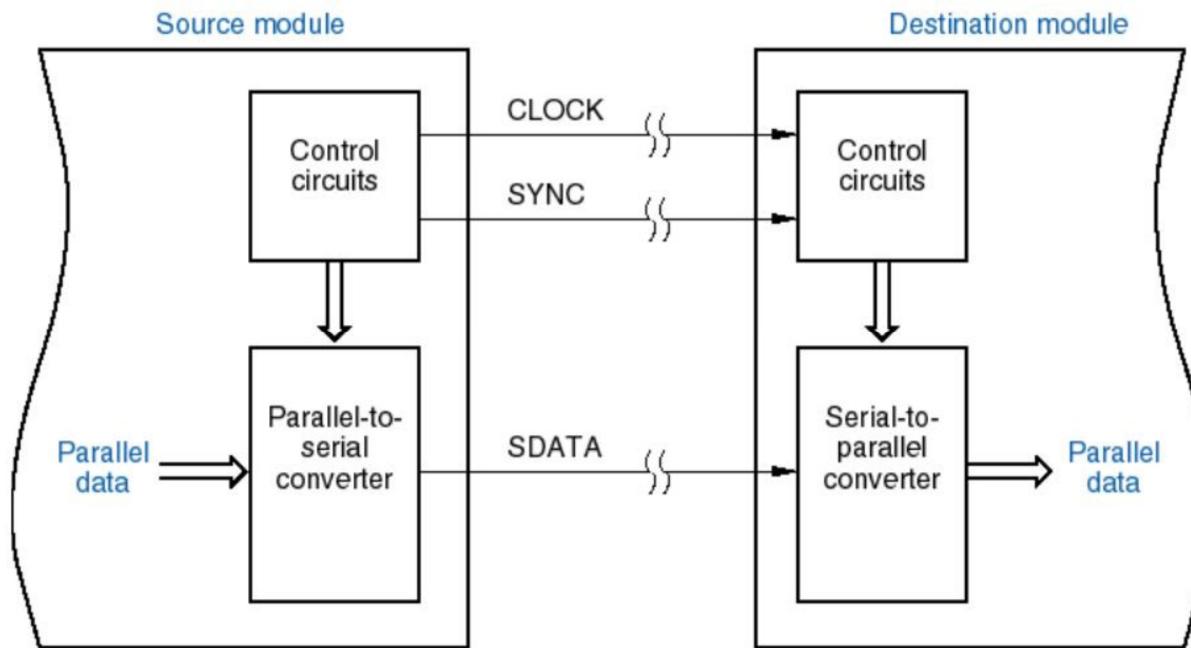
$$T_{\text{hold}} = 1 \text{ ns}$$

$$T_{\text{gate}} = 6 \text{ ns}$$

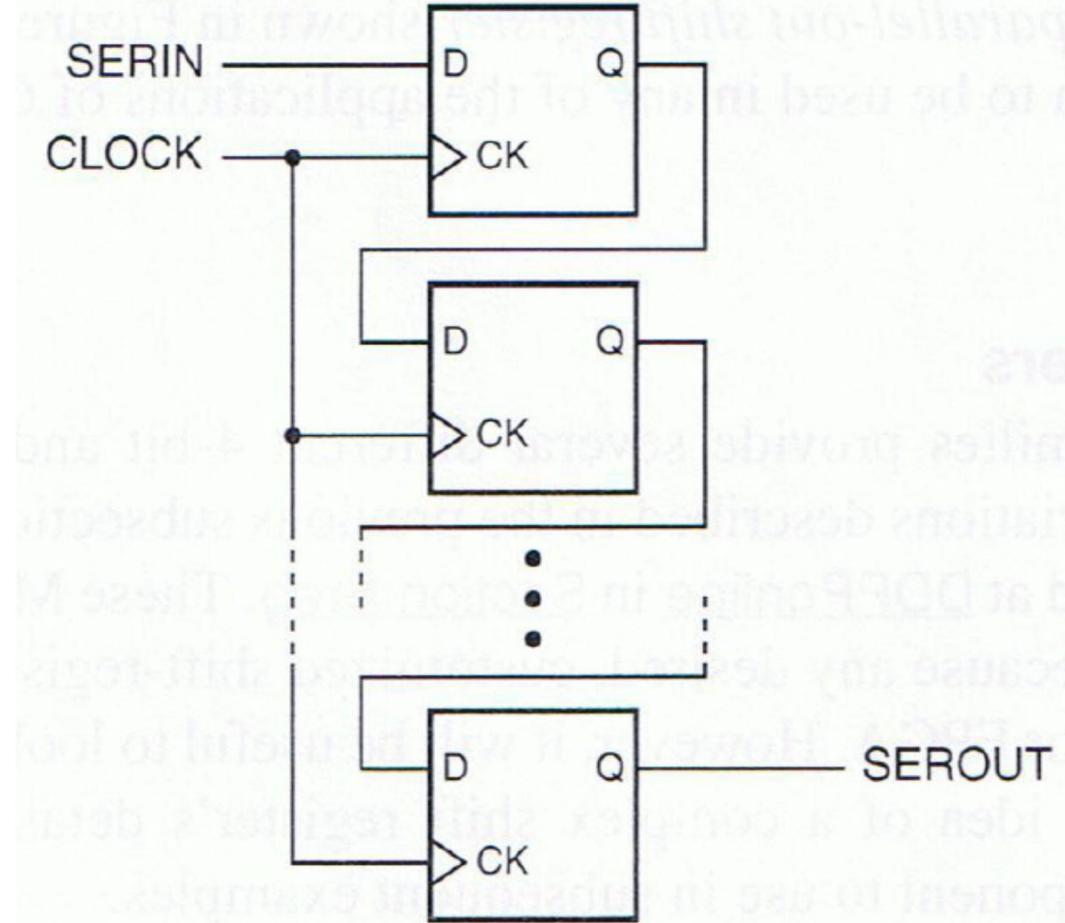
(assume that all gates exhibit the same delay)



# Serial Communication Among Different Devices



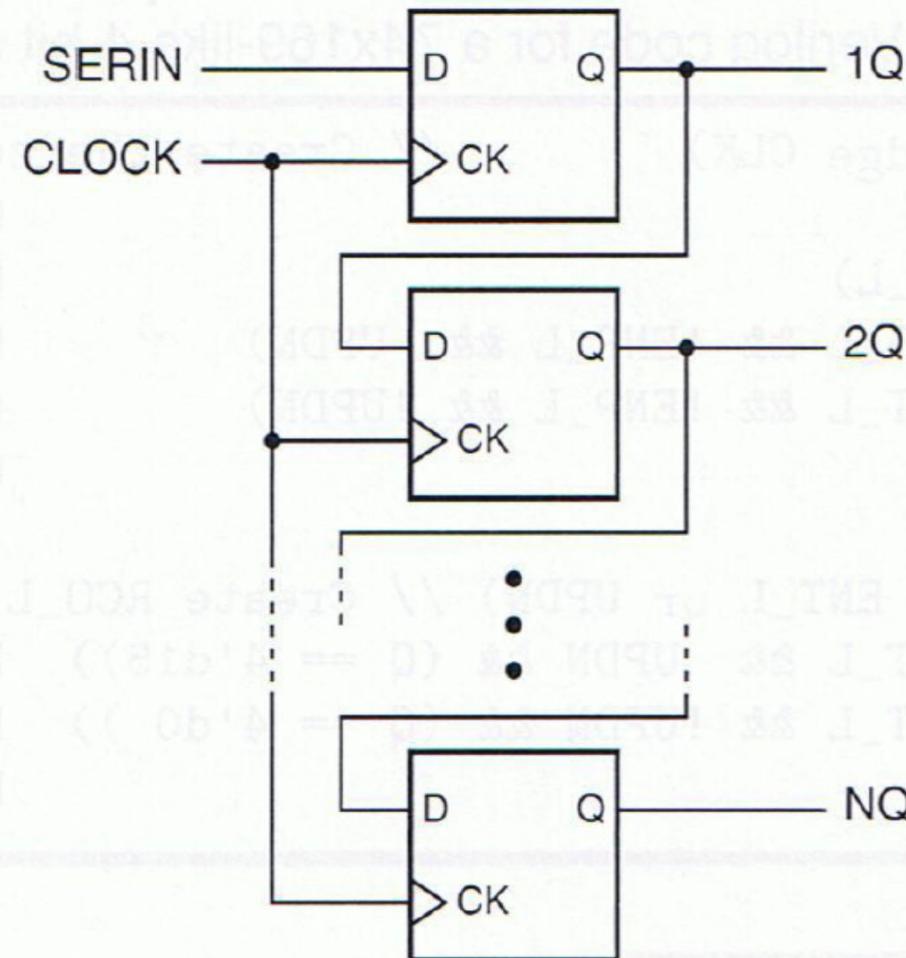
# Shift Registers (Serial-in, Serial-out)



**Figure 8-37**  
Structure of a  
serial-in, serial-out  
shift register.

Why flip-flop  
hold times are  
not violated?

# Shift Registers (Serial-in, Parallel-out)



**Figure 8-38**  
Structure of a  
serial-in, parallel-out  
shift register.

# Shift Registers (Parallel-in, Serial-out)

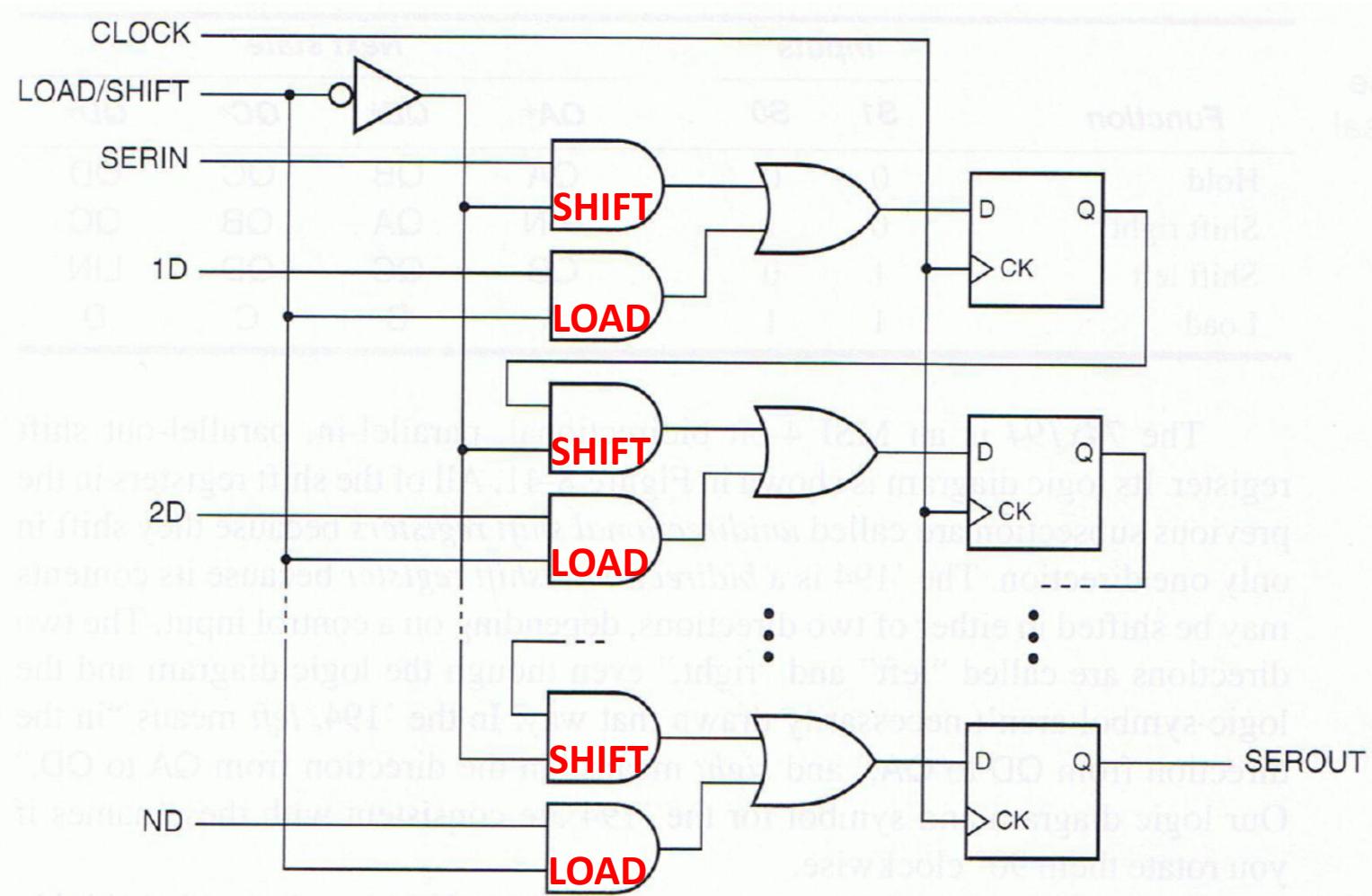


Figure 8-39 Structure of a parallel-in, serial-out shift register.

# Shift Registers (Parallel-in, Parallel-out)

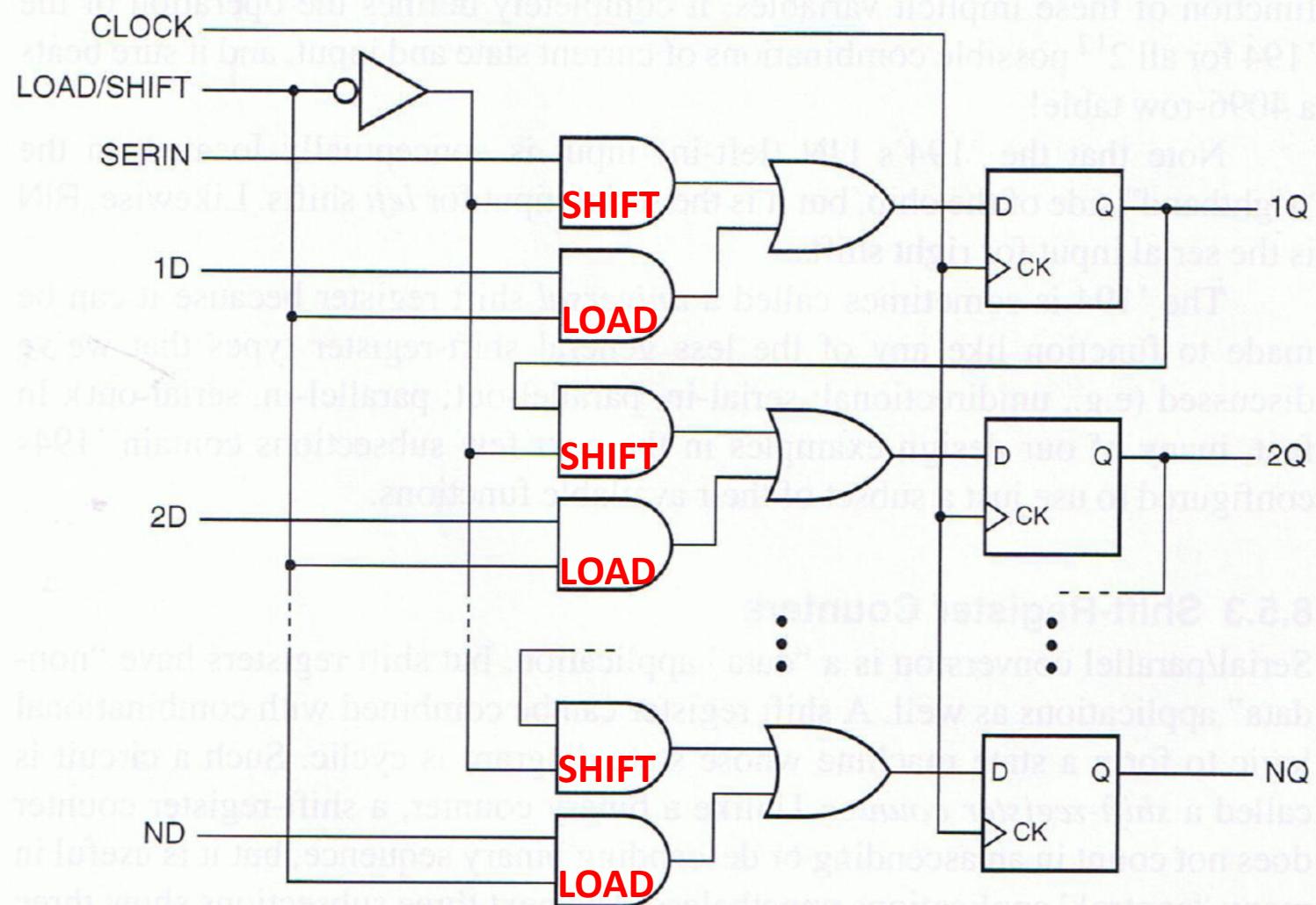
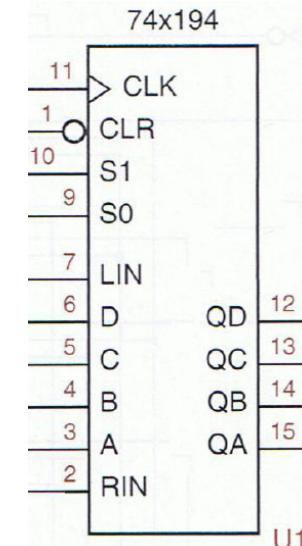
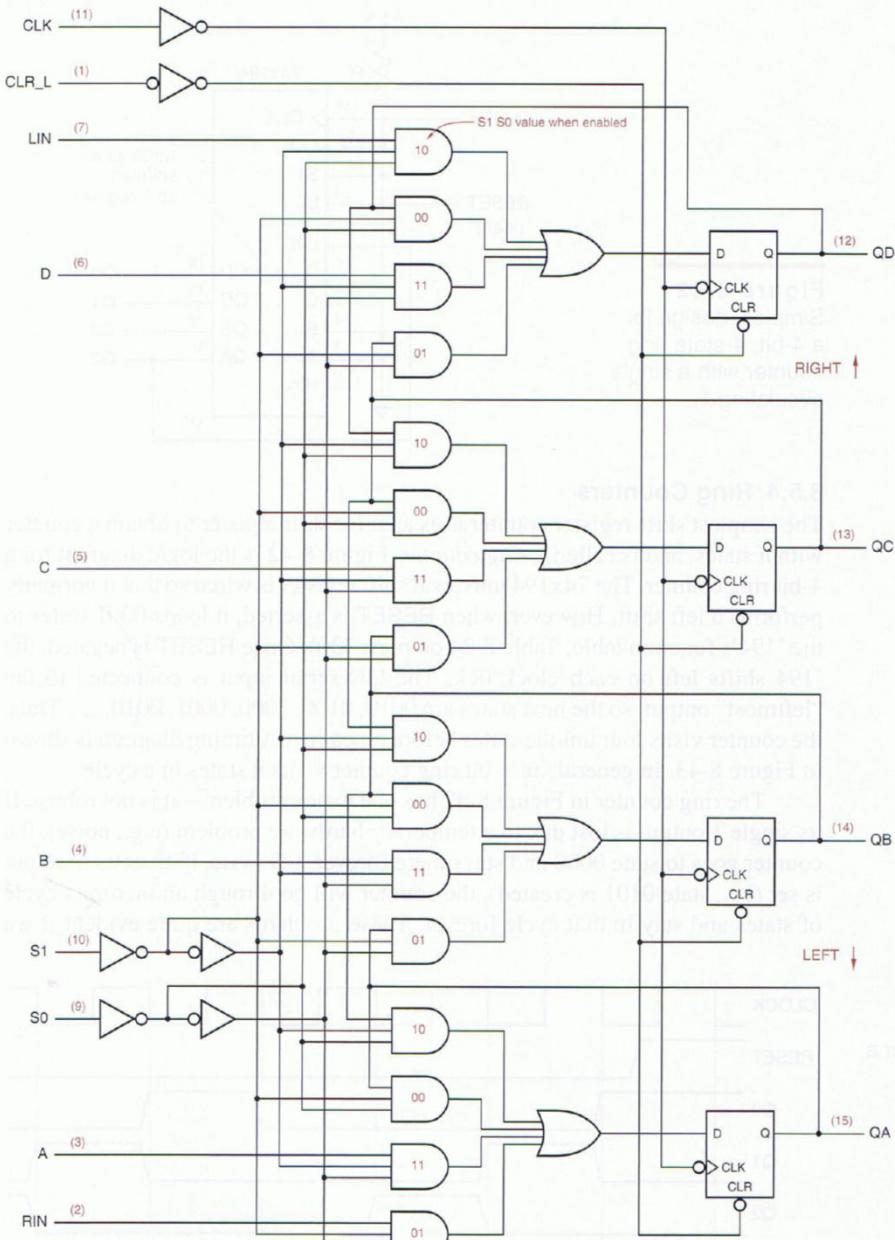


Figure 8-40 Structure of a parallel-in, parallel-out shift register.

# 74x194 Universal Shift Register



Function	Inputs		Next state			
	S1	S0	QA*	QB*	QC*	QD*
Hold	0	0	QA	QB	QC	QD
Shift right	0	1	RIN	QA	QB	QC
Shift left	1	0	QB	QC	QD	LIN
Load	1	1	A	B	C	D

Figure 8-41 Logic diagram for the 74x194 4-bit universal shift register, including pin numbers for a standard 16-pin dual in-line package.

# 74x194 Universal Shift Register (Load)

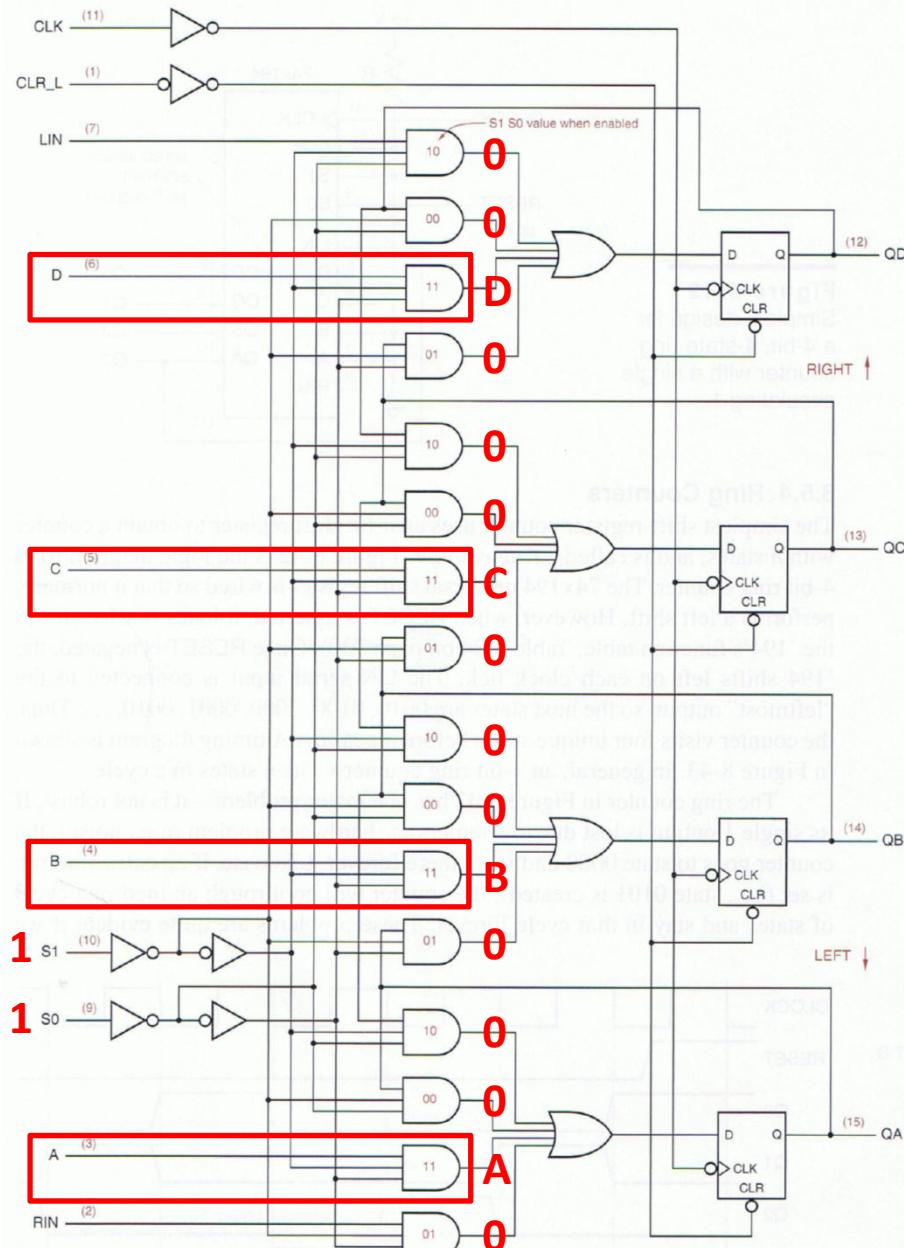
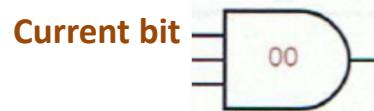
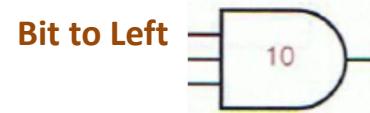


Figure 8-41 Logic diagram for the 74x194 4-bit universal shift register, including pin numbers for a standard 16-pin dual in-line package.



Function	Inputs		Next state			
	S1	S0	QA*	QB*	QC*	QD*
Hold	0	0	QA	QB	QC	QD
Shift right	0	1	RIN	QA	QB	QC
Shift left	1	0	QB	QC	QD	LIN
Load	1	1	A	B	C	D

# 74x194 Universal Shift Register (Hold)

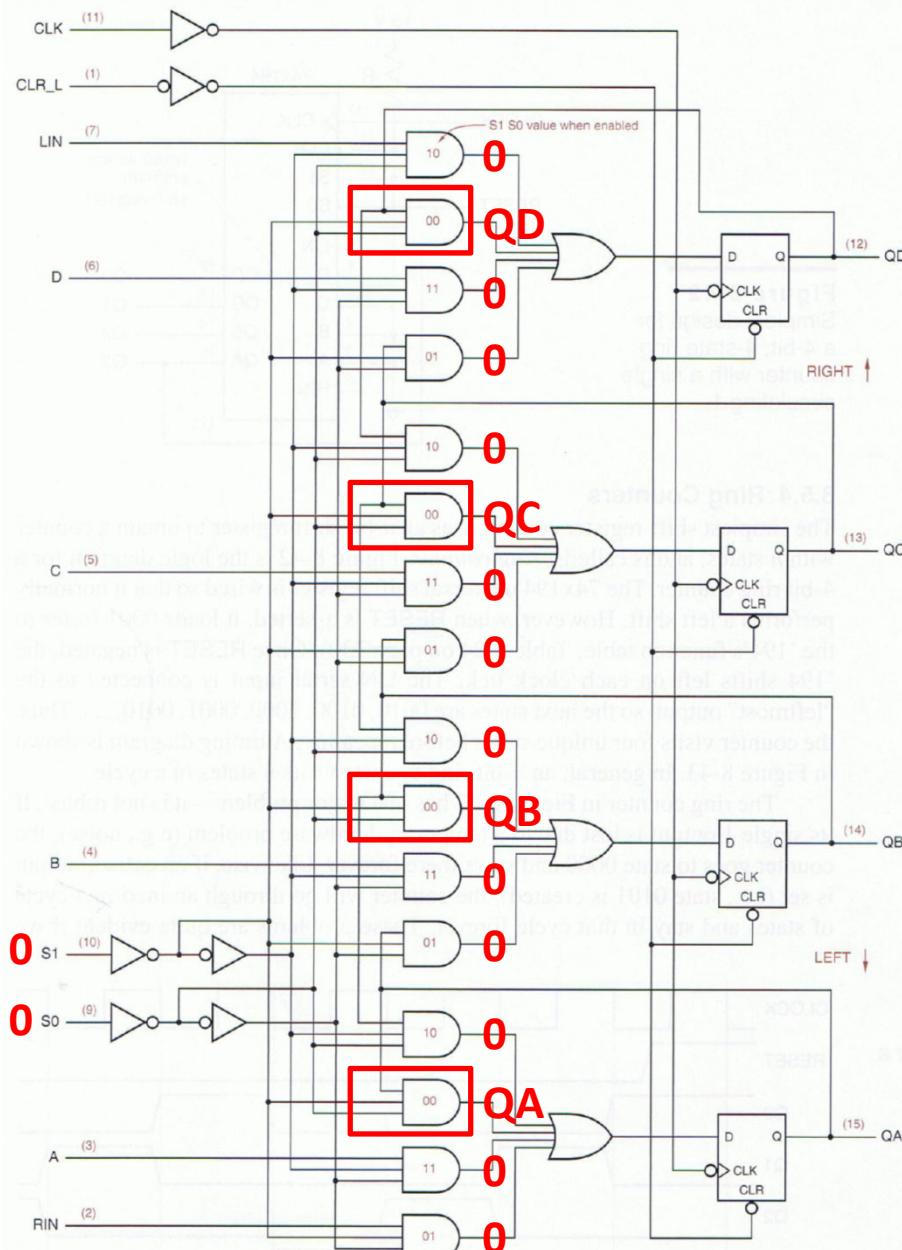
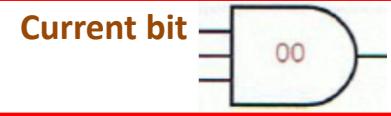


Figure 8-41 Logic diagram for the 74x194 4-bit universal shift register, including pin numbers for a standard 16-pin dual in-line package.



Function	Inputs		Next state			
	S1	S0	QA*	QB*	QC*	QD*
Hold	0	0	QA	QB	QC	QD
Shift right	0	1	RIN	QA	QB	QC
Shift left	1	0	QB	QC	QD	LIN
Load	1	1	A	B	C	D

# 74x194 Universal Shift Register (Shift Right)

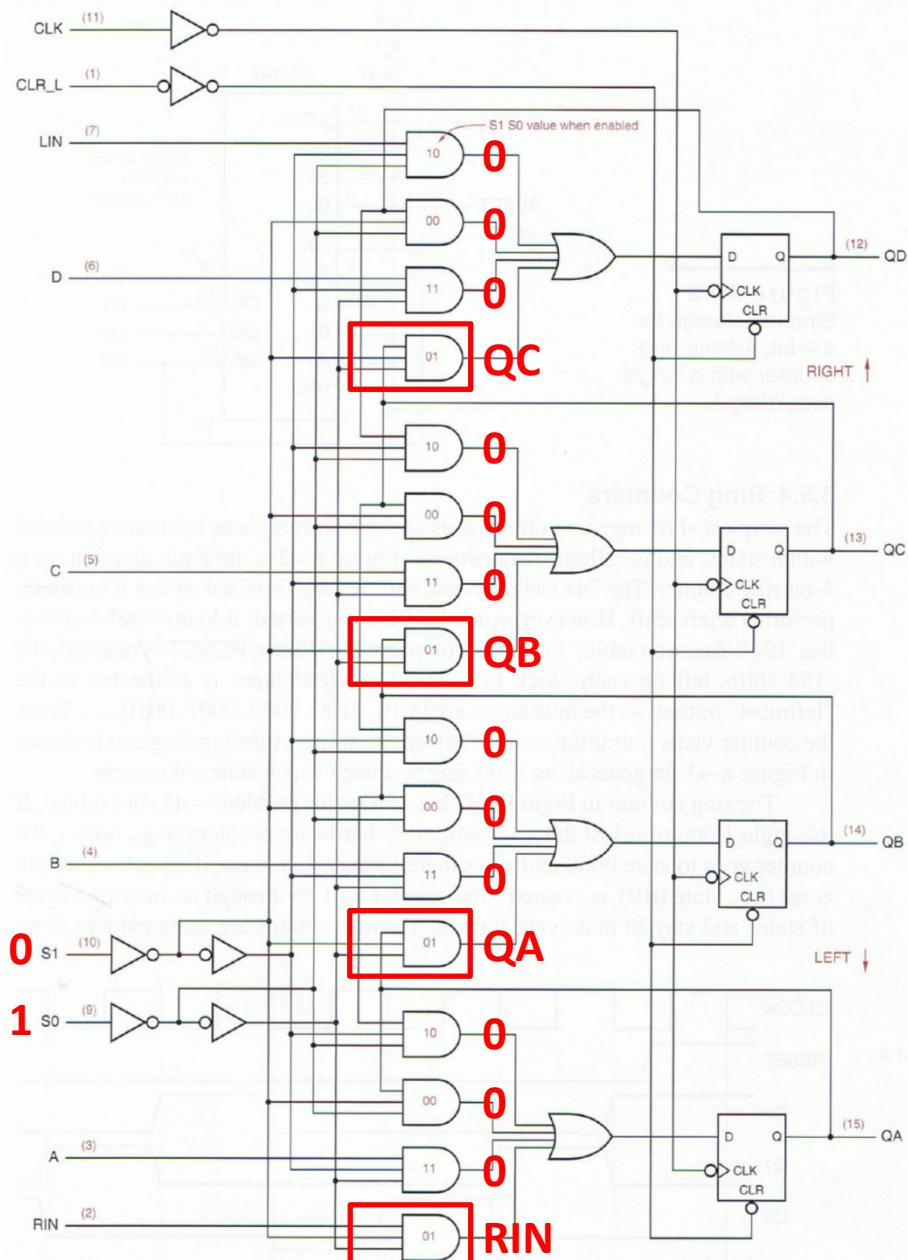
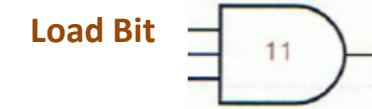
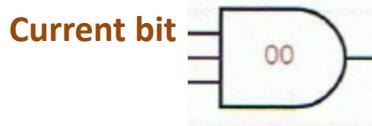


Figure 8-41 Logic diagram for the 74x194 4-bit universal shift register, including pin numbers for a standard 16-pin dual in-line package.



Function	Inputs		Next state			
	S1	S0	QA*	QB*	QC*	QD*
Hold	0	0	QA	QB	QC	QD
Shift right	0	1	RIN	QA	QB	QC
Shift left	1	0	QB	QC	QD	LIN
Load	1	1	A	B	C	D

# 74x194 Universal Shift Register (Shift Left)

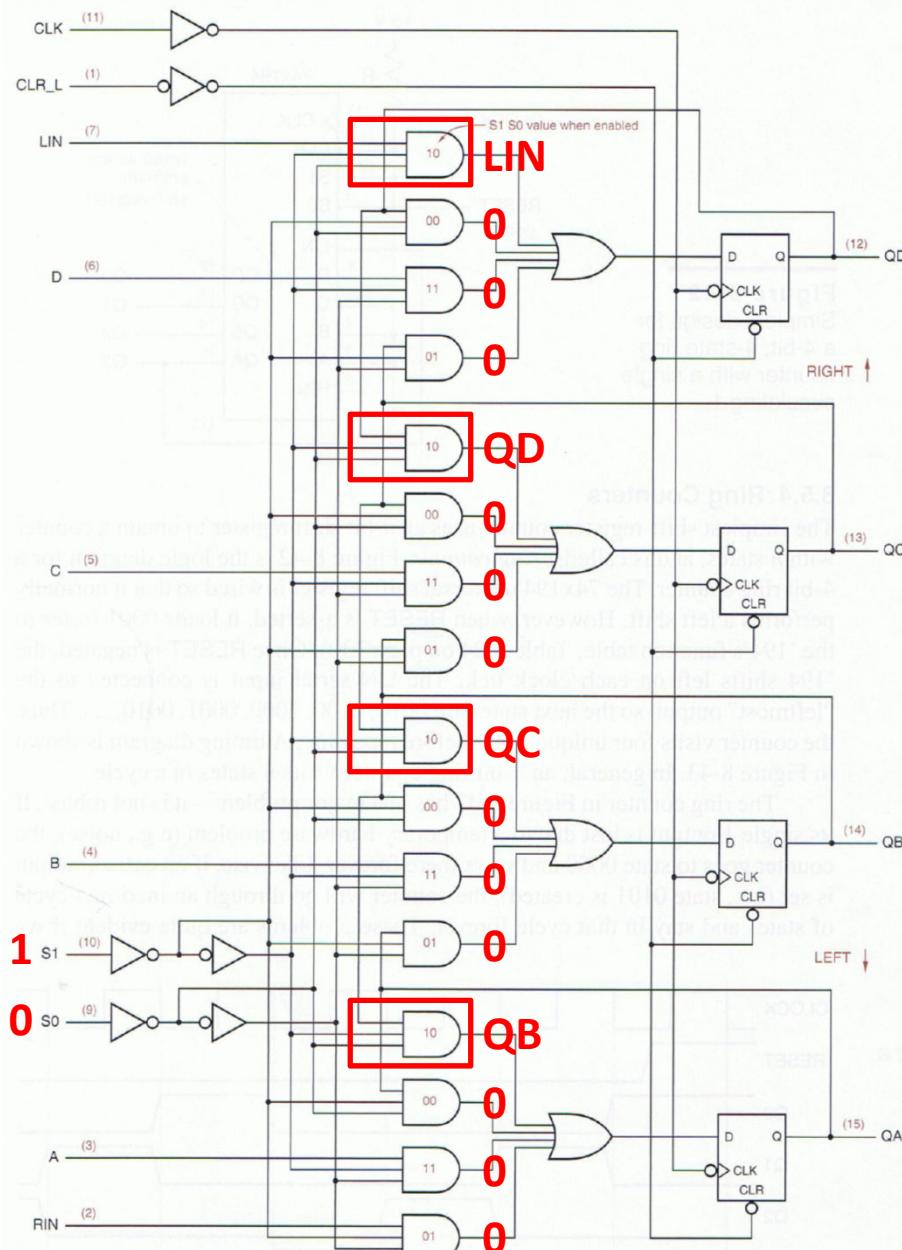
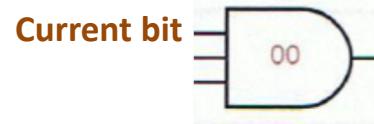
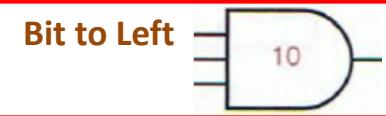
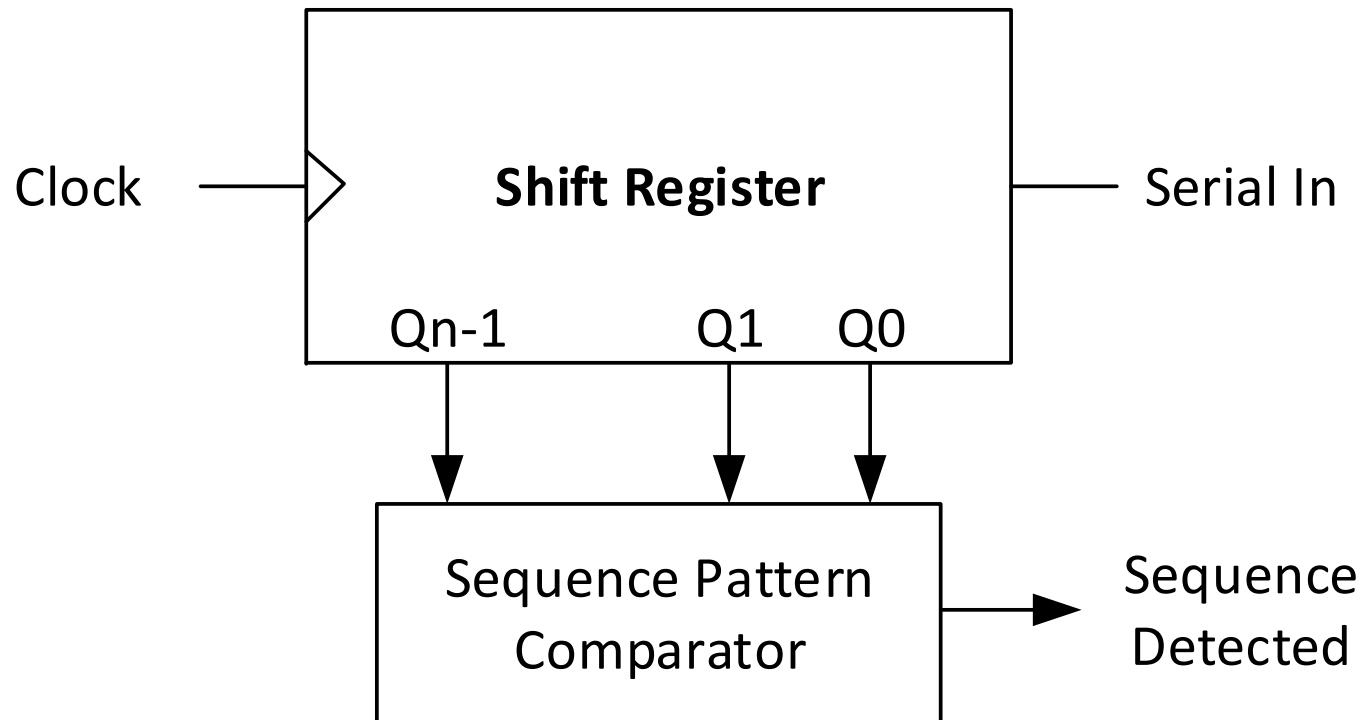


Figure 8-41 Logic diagram for the 74x194 4-bit universal shift register, including pin numbers for a standard 16-pin dual in-line package.



Function	Inputs		Next state			
	S1	S0	QA*	QB*	QC*	QD*
Hold	0	0	QA	QB	QC	QD
Shift right	0	1	RIN	QA	QB	QC
<b>Shift left</b>	<b>1</b>	<b>0</b>	<b>QB</b>	<b>QC</b>	<b>QD</b>	<b>LIN</b>
Load	1	1	A	B	C	D

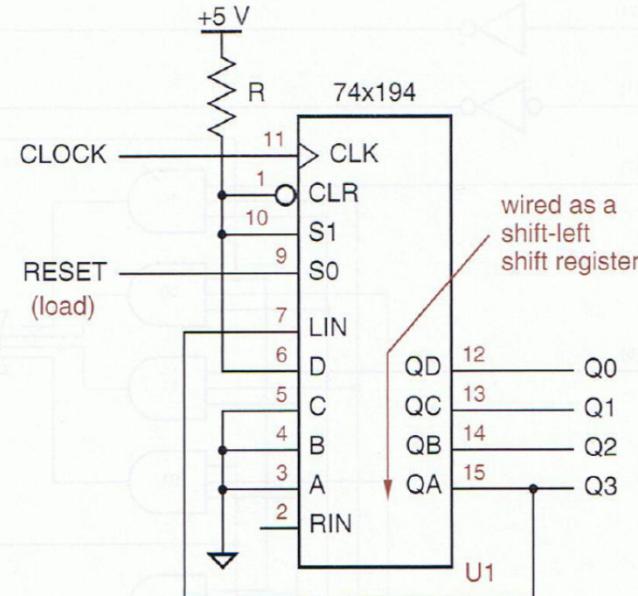
# Building Sequence Detectors with Shift Registers (overlapping sequences)



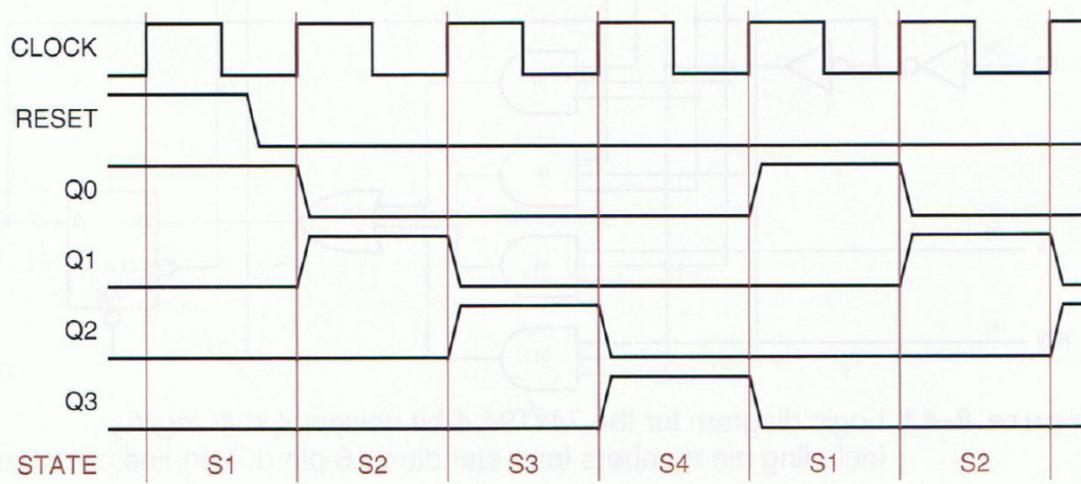
How to implement efficiently the sequence pattern comparator?

# Ring Counters

**Figure 8-42**  
Simplest design for  
a 4-bit, 4-state ring  
counter with a single  
circulating 1.

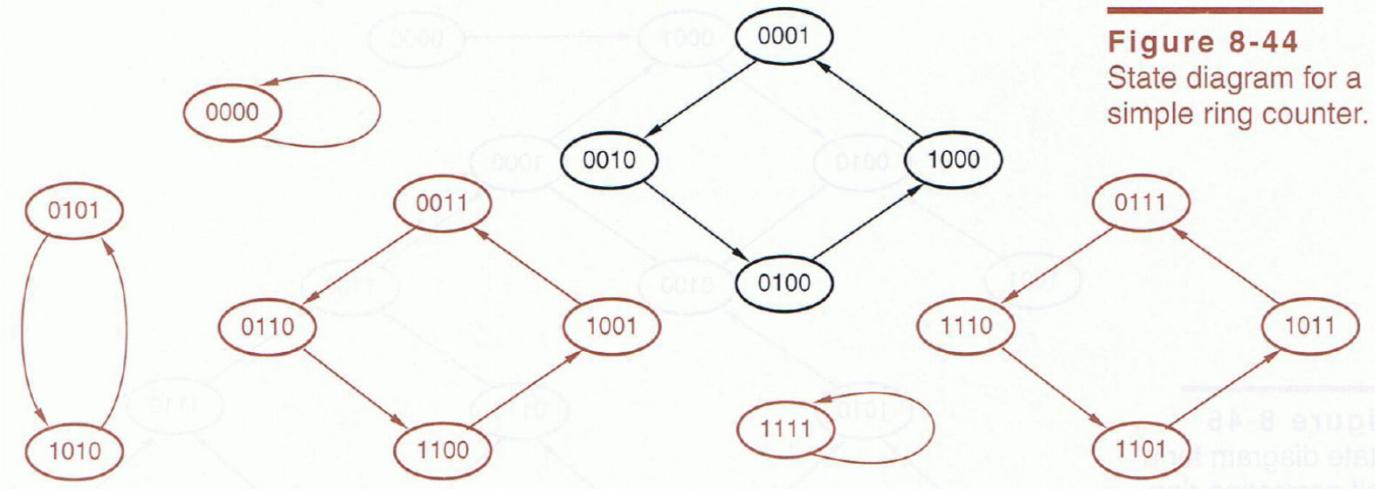
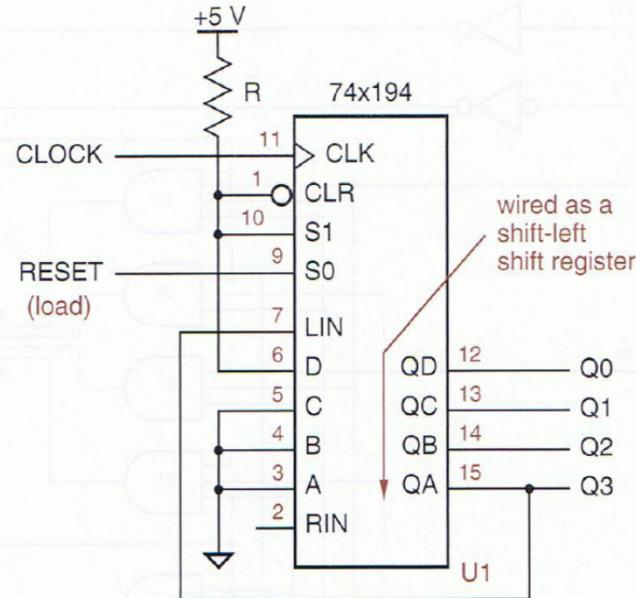


**Figure 8-43**  
Timing diagram for a  
4-bit ring counter.



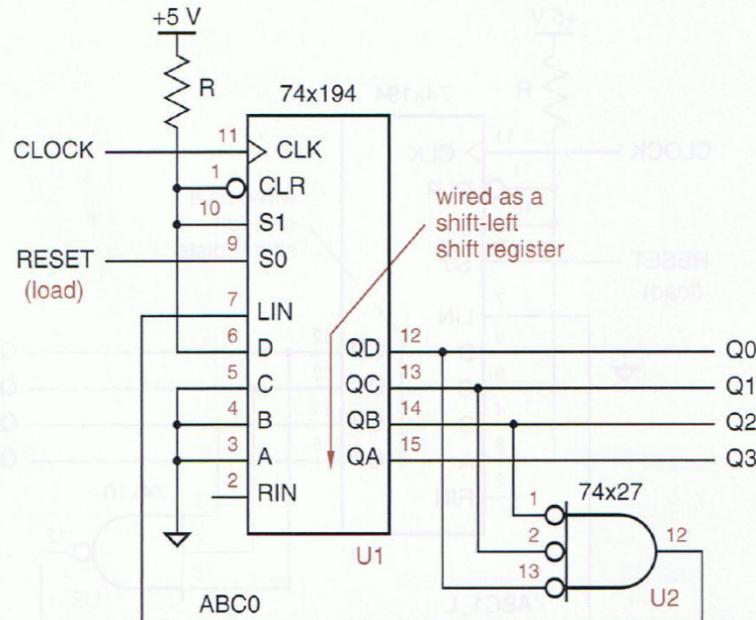
# Ring Counter State Diagram

**Figure 8-42**  
Simplest design for  
a 4-bit, 4-state ring  
counter with a single  
circulating 1.

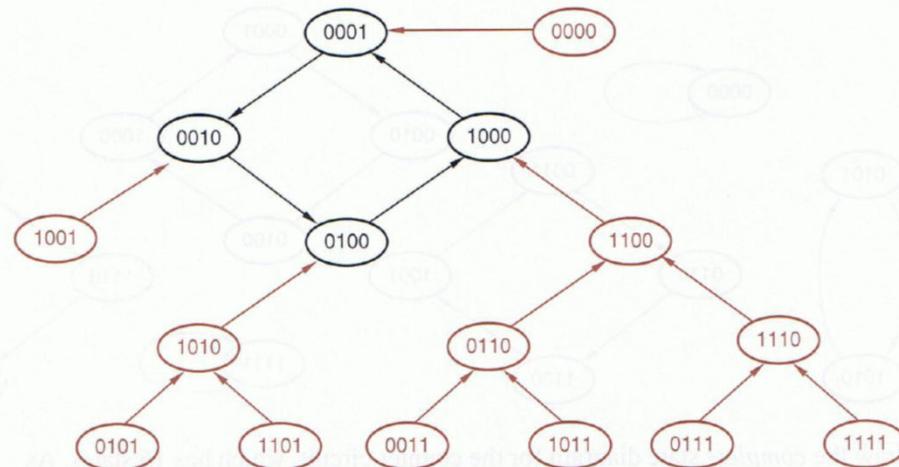


**Figure 8-44**  
State diagram for a  
simple ring counter.

# Self-correcting Ring Counter

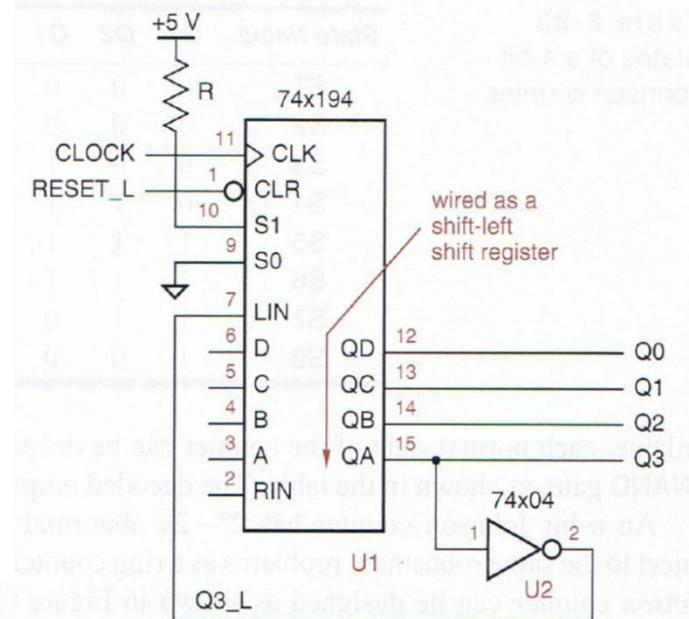


**Figure 8-45**  
Self-correcting 4-bit,  
4-state ring counter  
with a single  
circulating 1.



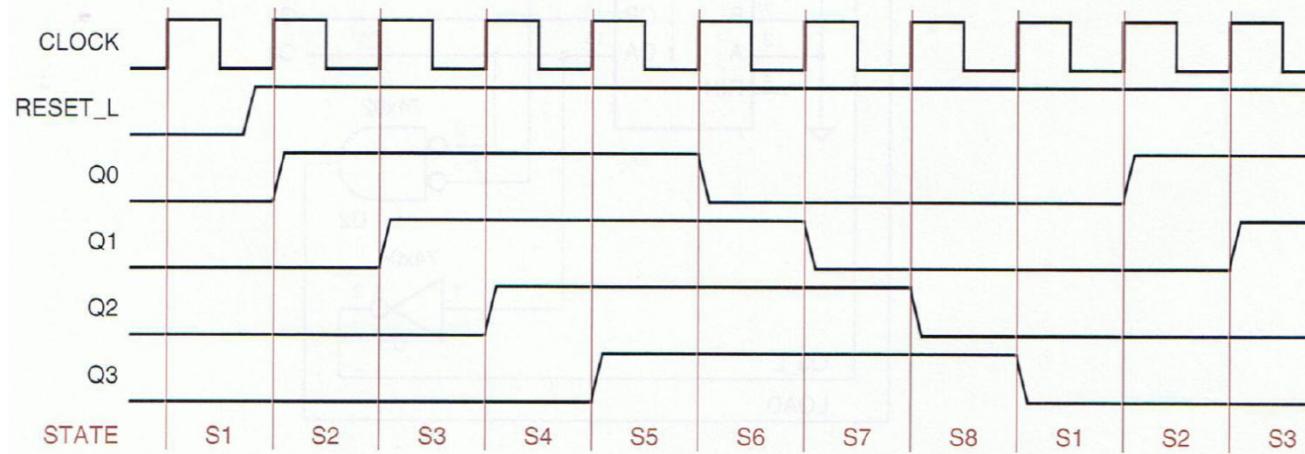
**Figure 8-46**  
State diagram for a  
self-correcting ring  
counter.

# Johnson Counter (twisted ring)



**Figure 8-48**  
Basic 4-bit, 8-state  
Johnson counter.

**Figure 8-49** Timing diagram for a 4-bit Johnson counter.

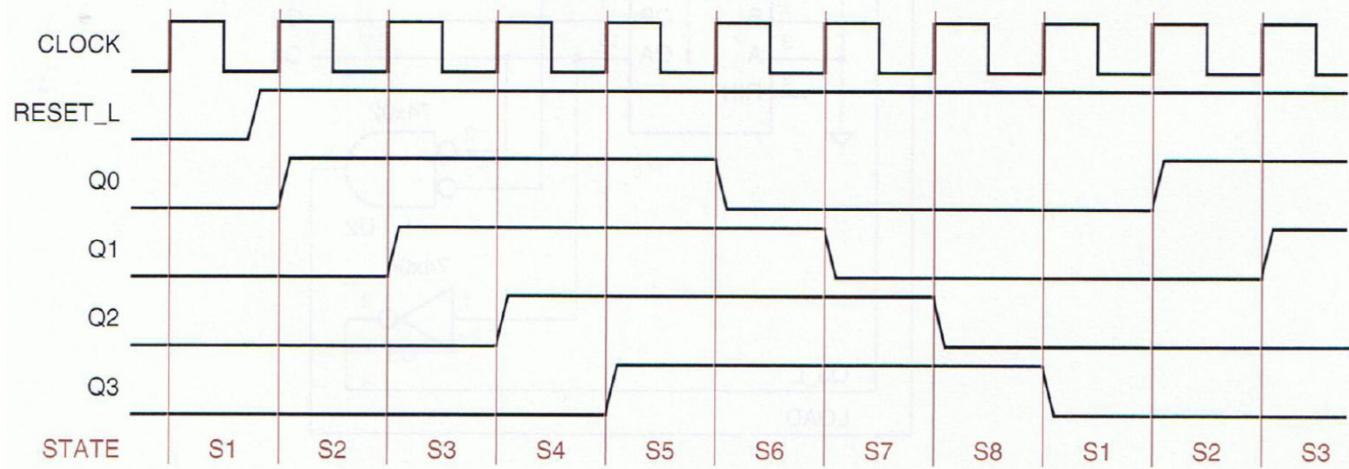


# Johnson Counter (twisted ring)

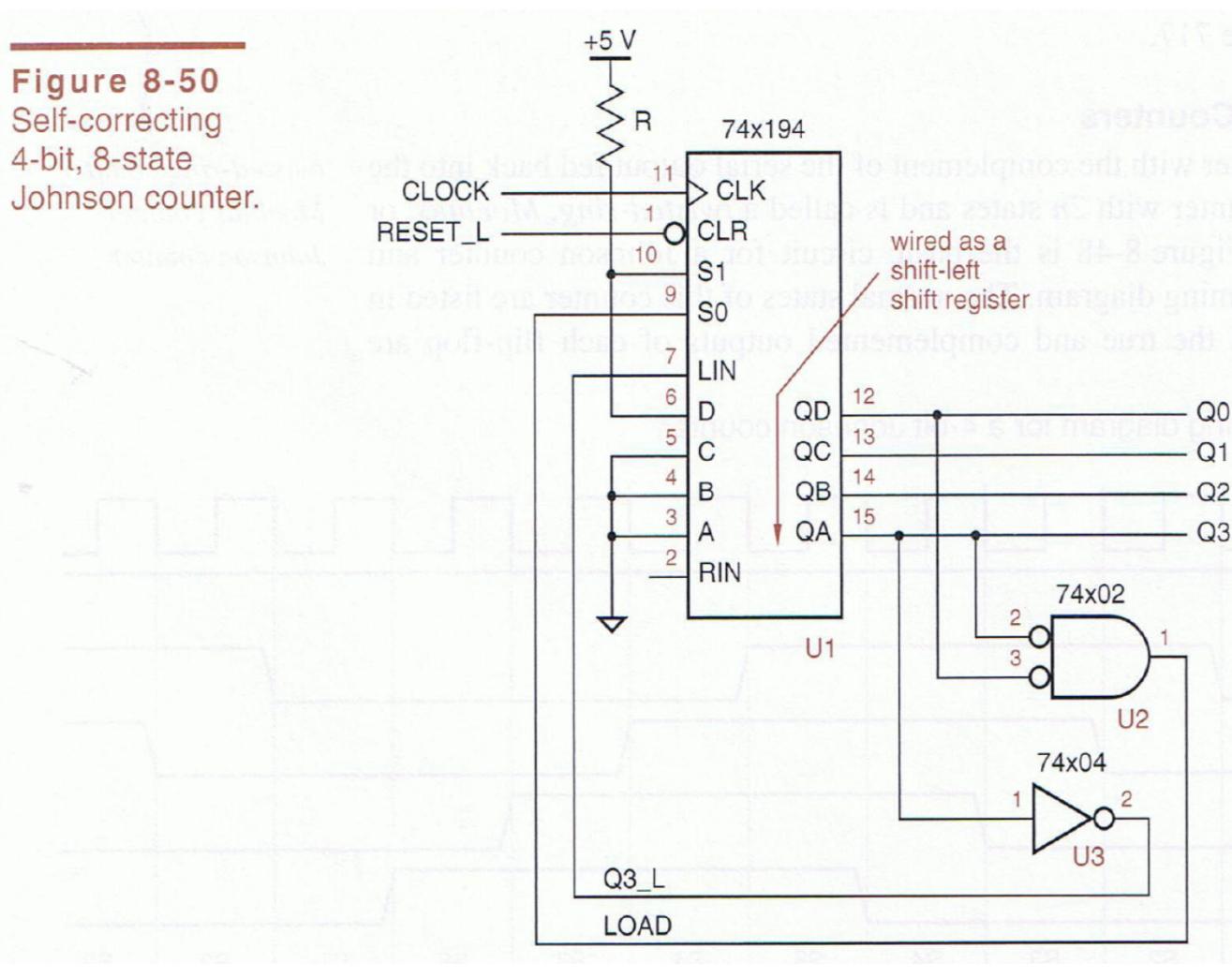
**Table 8-25**  
States of a 4-bit  
Johnson counter.

State Name	Q3	Q2	Q1	Q0	Decoding
S1	0	0	0	0	$Q3' \cdot Q0'$
S2	0	0	0	1	$Q1' \cdot Q0$
S3	0	0	1	1	$Q2' \cdot Q1$
S4	0	1	1	1	$Q3' \cdot Q2$
S5	1	1	1	1	$Q3 \cdot Q0$
S6	1	1	1	0	$Q1 \cdot Q0'$
S7	1	1	0	0	$Q2 \cdot Q1'$
S8	1	0	0	0	$Q3 \cdot Q2'$

**Figure 8-49** Timing diagram for a 4-bit Johnson counter.



# Self Correcting Johnson Counter



# Conclusion

- At the end of this lecture and corresponding lab, it is fundamental to know some fundamental sequential blocks / components, namely multibit registers, counters and shift registers
- Plan for the next lecture
  - Iterative vs. sequential circuits

Reading chapter 7 (4<sup>th</sup> ed.) or chapter 11 (5<sup>th</sup> ed.) of John F. Wakerly, “Digital Design – Principles and Practices”, Pearson – Prentice Hall, is highly recommended.