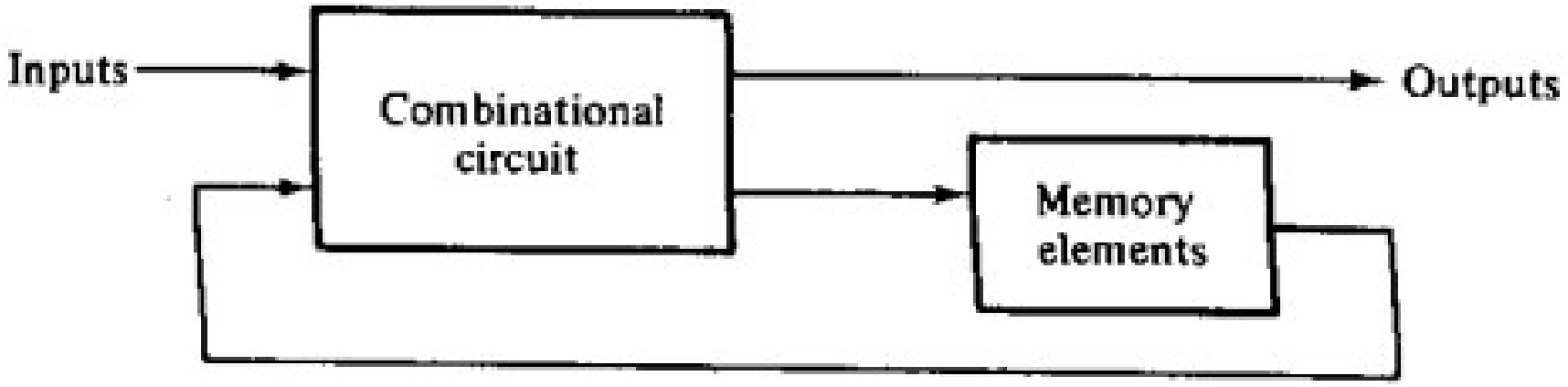
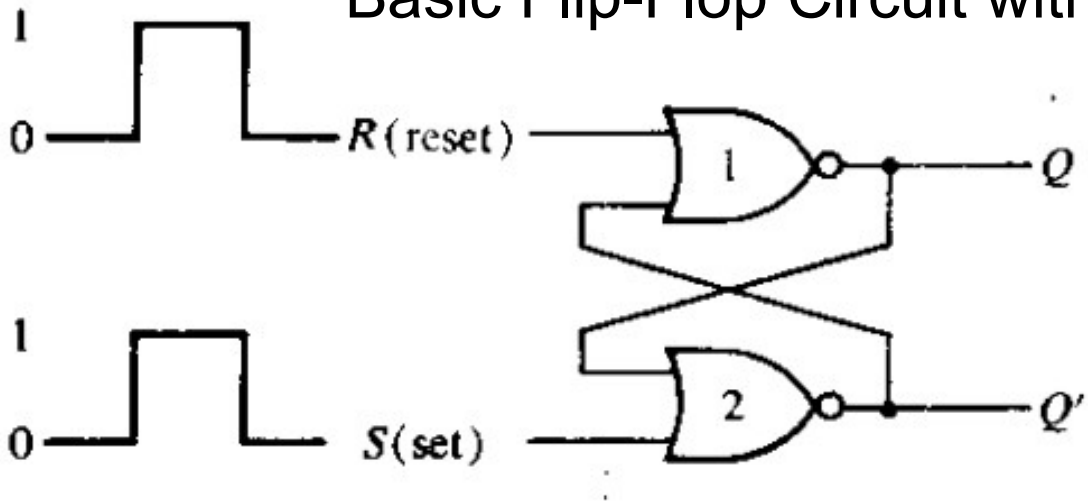
COC2072 – Digital Logic and System Design

Prof. M. M. Sufyan Beg
Department of Computer Engineering
Z. H. College of Engineering & Technology
Aligarh Muslim University, India

Unit III – Sequential Circuits



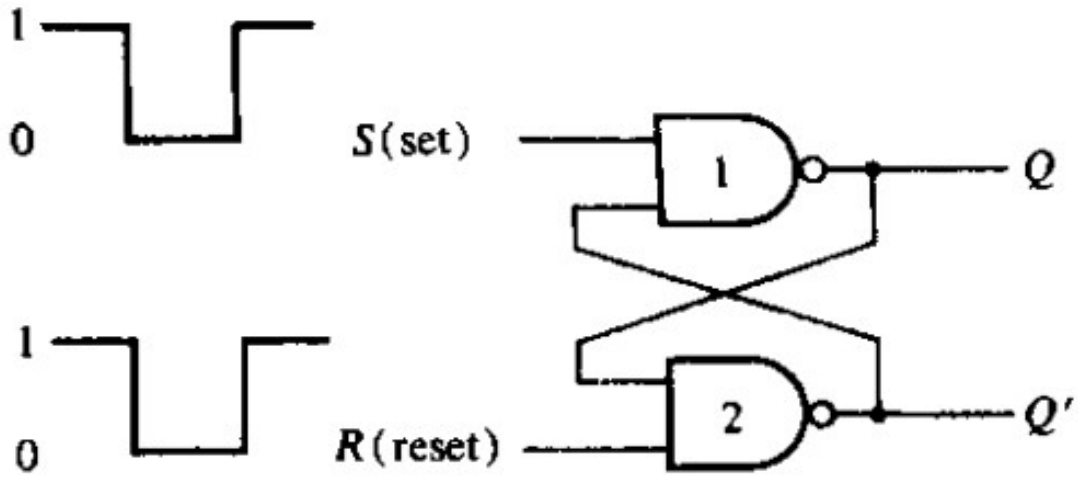
Basic Flip-Flop Circuit with NOR Gates



S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

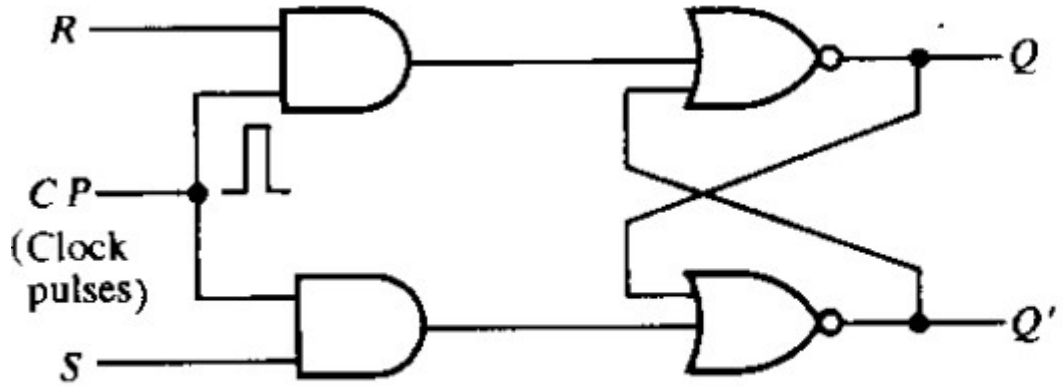
Basic Flip-Flop Circuit with NAND Gates



S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	

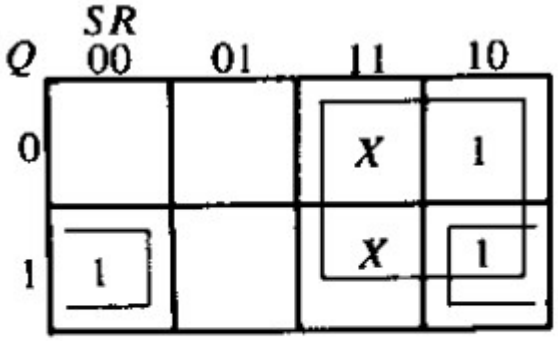
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Clocked RS Flip-Flop



Q	S	R	Q(t + 1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate

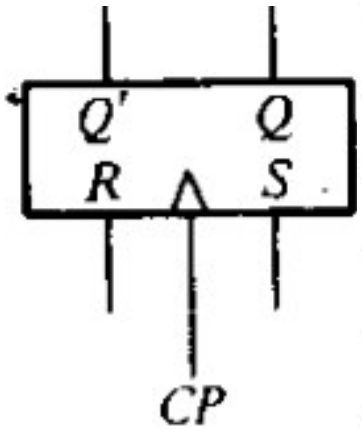
Characteristic table



$Q(t+1) = S + R'Q$

$SR = 0$

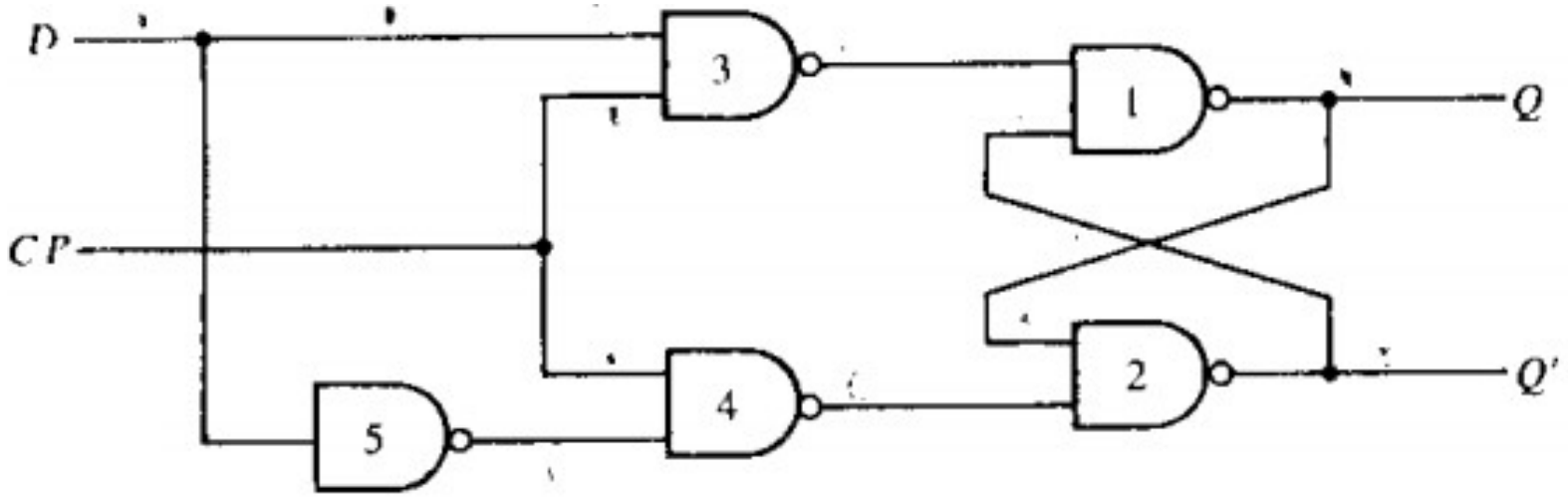
Characteristic equation

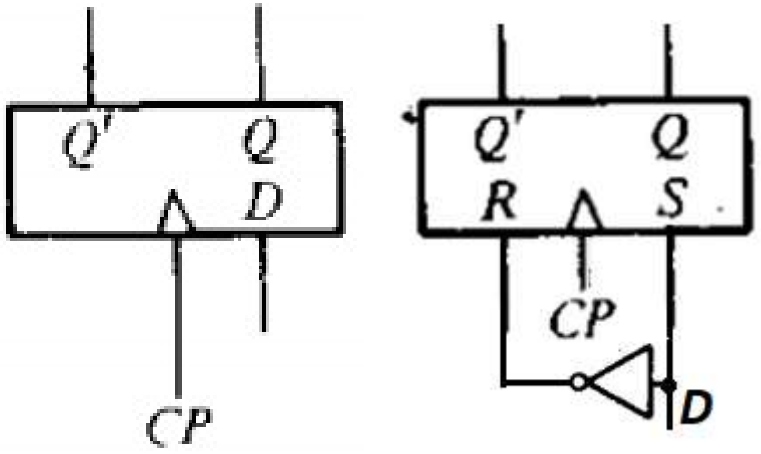


S	R	Q(t+1)
0	0	$Q(t)$
0	1	0
1	0	1
1	1	Indeterminate

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

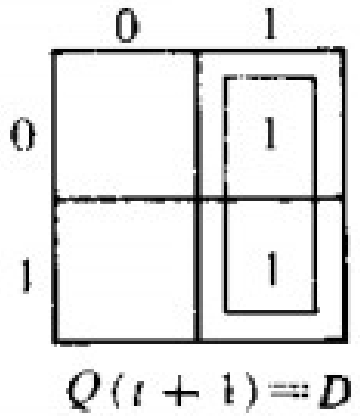
D Flip-Flop





Q	D	Q(t + 1)
0	0	0
0	1	1
1	0	0
1	1	1

Characteristic table



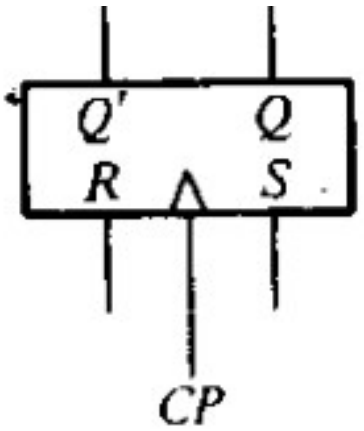
Characteristic equation

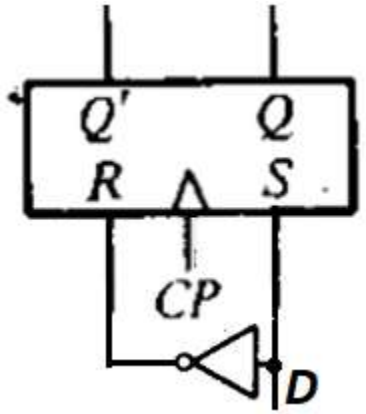
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

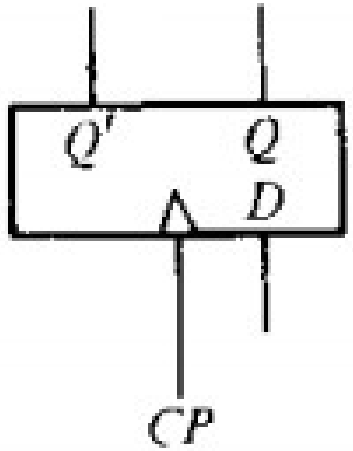
RS Flip-Flop and D Flip-Flop

S	R	Q(t+1)
0	0	$Q(t)$
0	1	0
1	0	1
1	1	Indeterminate

D	Q(t+1)
0	0
1	1

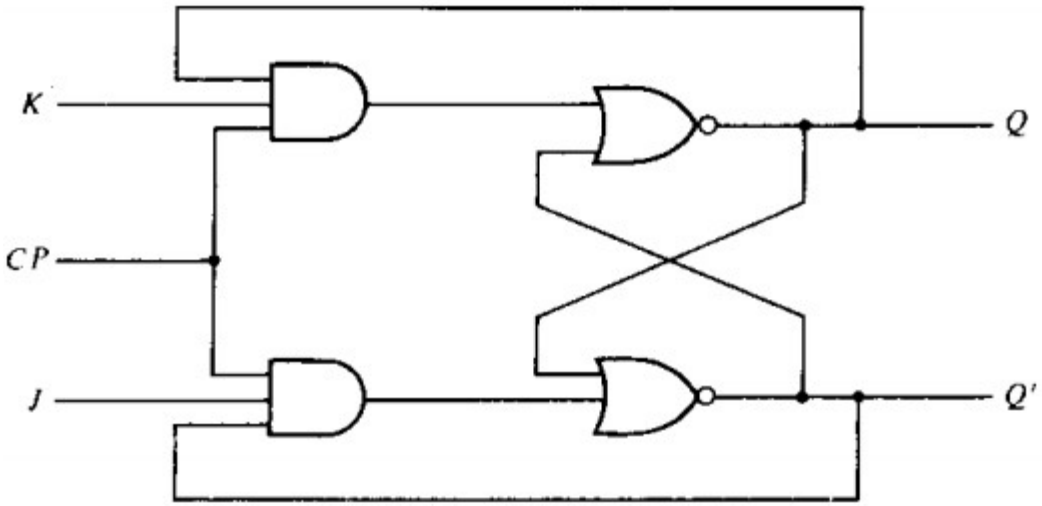


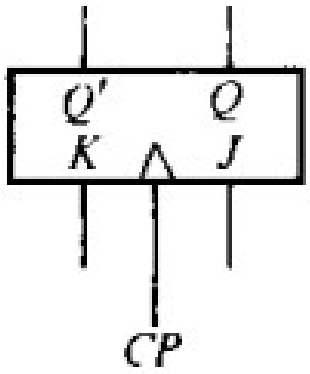




Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

JK Flip-Flop

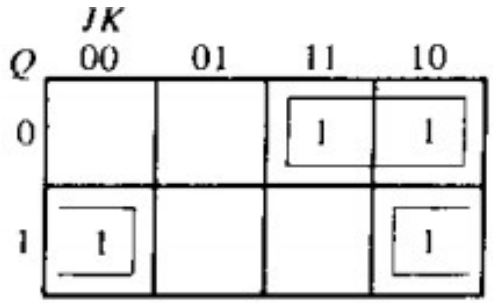




J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q(t)'

Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Characteristic table

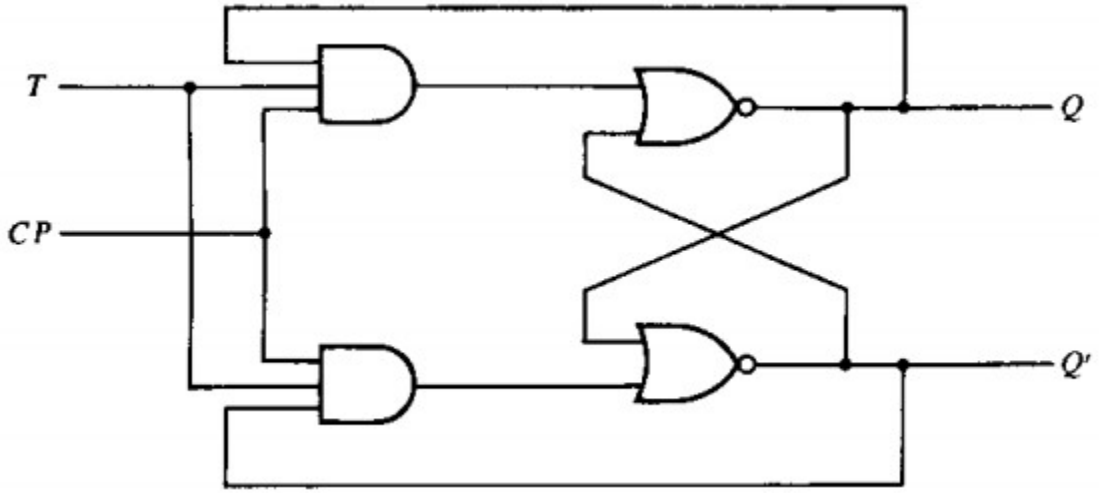


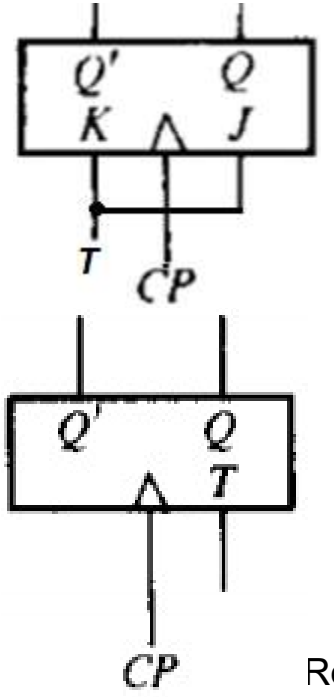
$Q(t+1) = JQ' + K'Q$

Characteristic Equation

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

T Flip-Flop

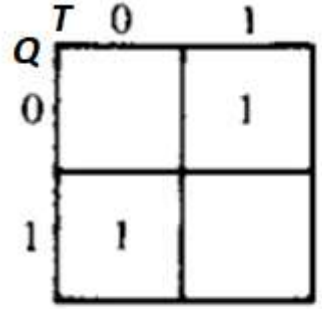




T	Q(t+1)
0	$Q(t)$
1	$Q(t)'$

Q	T	Q(t + 1)
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic table

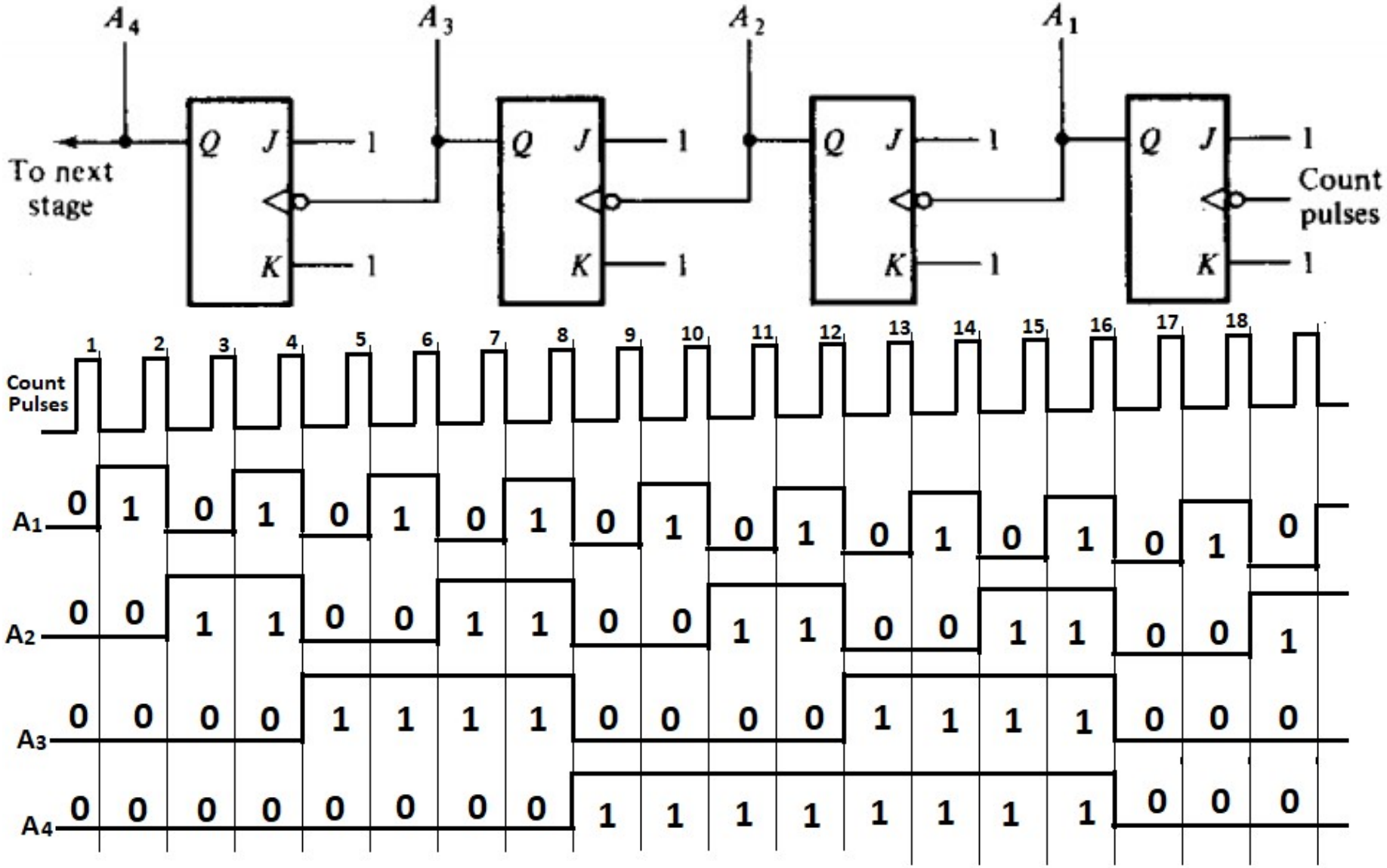


$Q(t+1) = TQ' + T'Q$

Characteristic equation

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

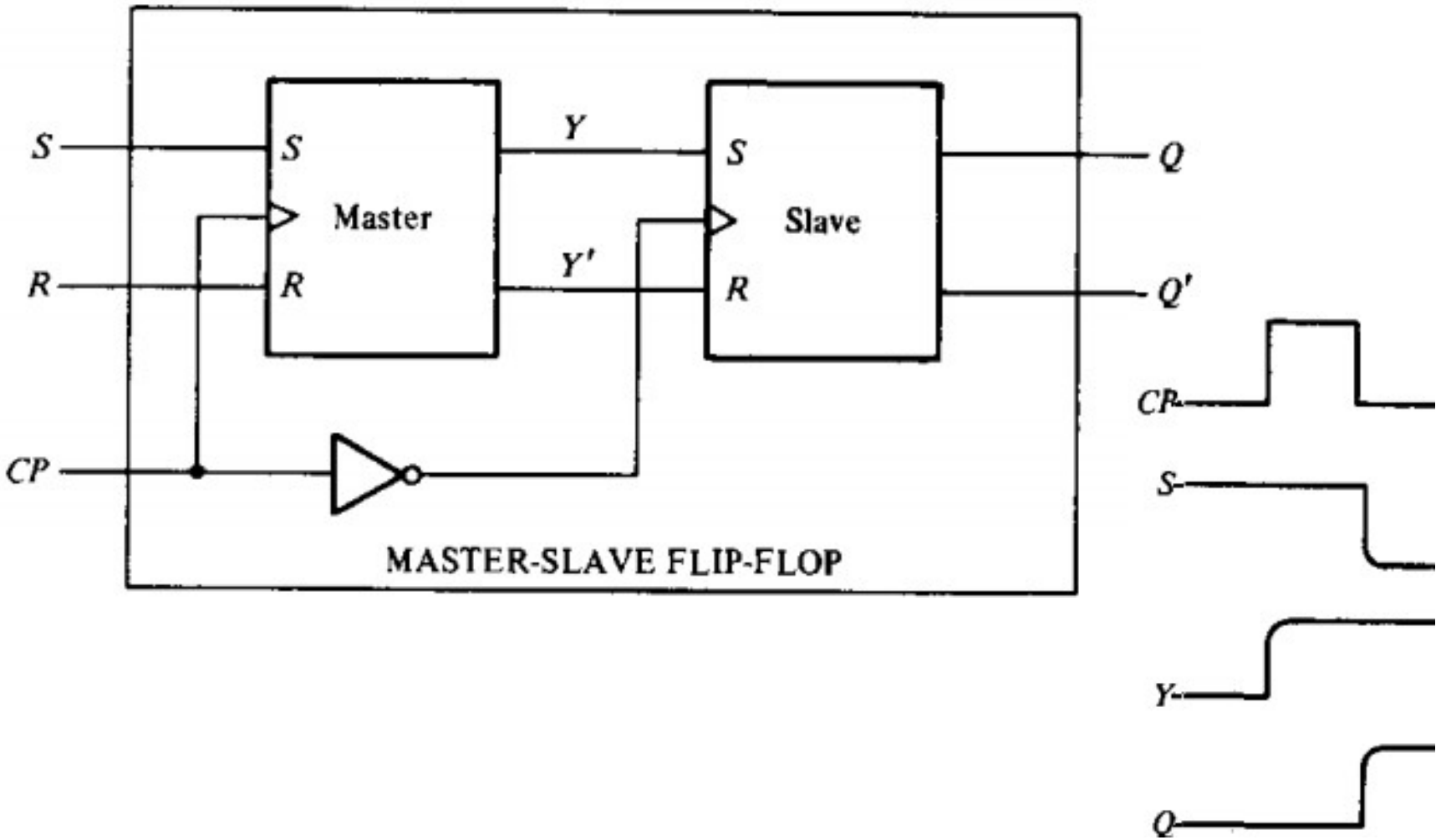
Binary Ripple Counter



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

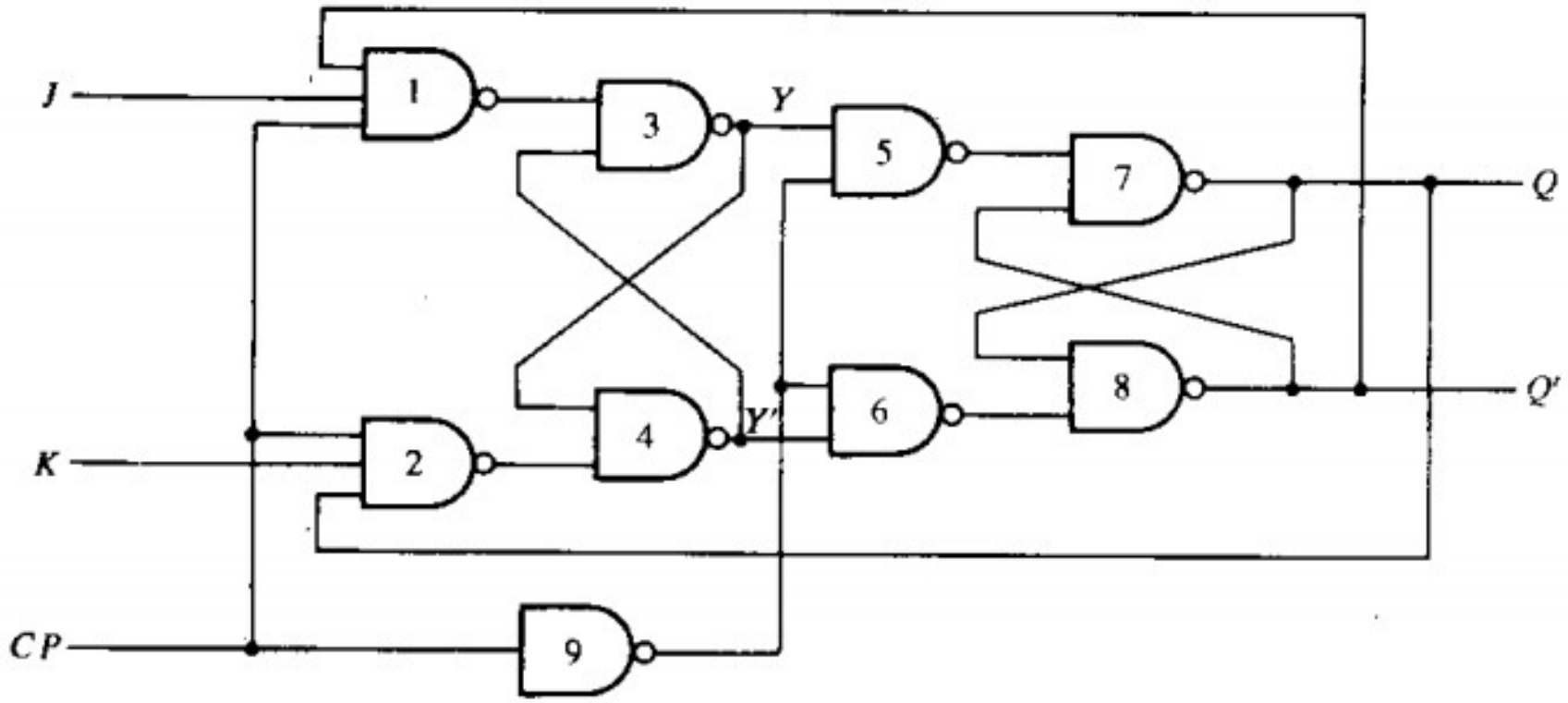
Master-Slave Flip-Flop

- Race Around Condition in JK FF



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Clocked Master-Slave JK Flip-Flop



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Flip-Flop Excitation Tables

S	R	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	?

Characteristic Table

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Excitation Table

D	$Q(t+1)$
0	0
1	1

$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Flip-Flop Excitation Tables (contd.)

J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$

$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Characteristic Table

T	$Q(t+1)$
0	Q(t)
1	Q'(t)

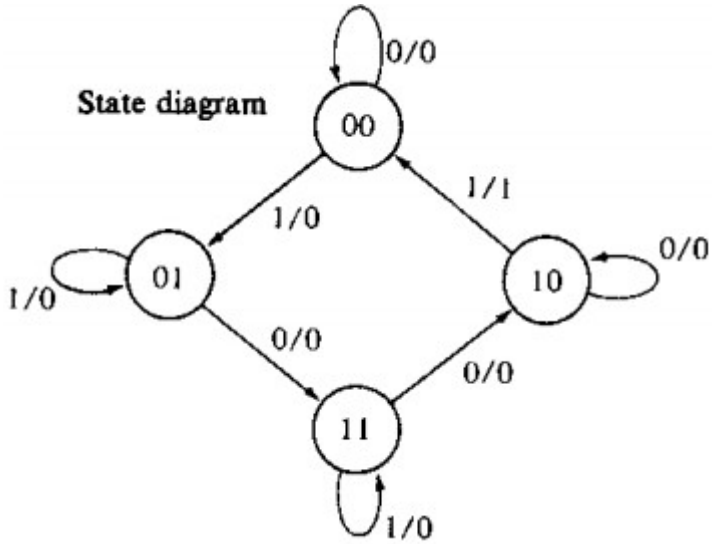
Excitation Table

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Sequential Circuit Design Using RS-FF

State diagram



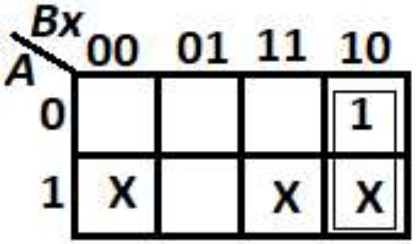
State table

	Next state		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
AB	AB	AB	y	y
00	00	01	0	0
01	11	01	0	0
10	10	00	0	1
11	10	11	0	0

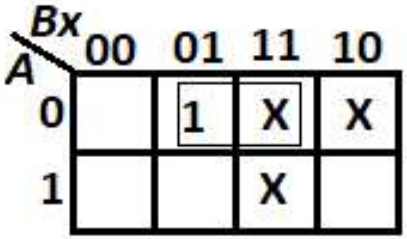
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Sequential Circuit Design Using RS-FF (contd.)

Present State		Input	Next State		Required Flip-Flop Inputs				Output
A	*B*	*x*	*A*	*B*	*SA*	*RA*	*SB*	*RB*	*Y*
0	0	0	0	0	0	X	0	X	0
0	0	1	0	1	0	X	1	0	0
0	1	0	1	1	1	0	X	0	0
0	1	1	0	1	0	X	X	0	0
1	0	0	1	0	X	0	0	X	0
1	0	1	0	0	0	1	0	X	1
1	1	0	1	0	X	0	0	1	0
1	1	1	1	1	X	0	X	0	0

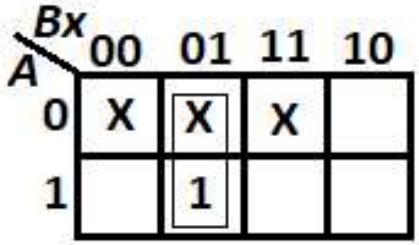


$SA = Bx'$

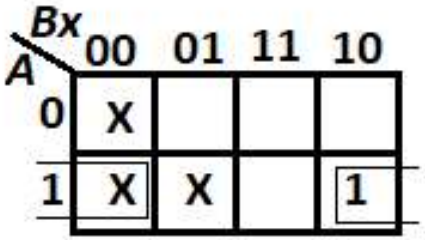


$SB = A'x$

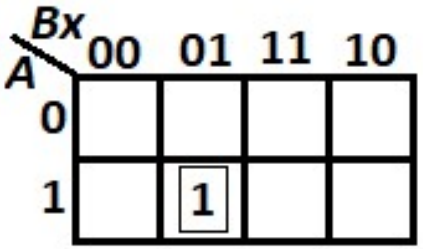
$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0



$RA = B'x$



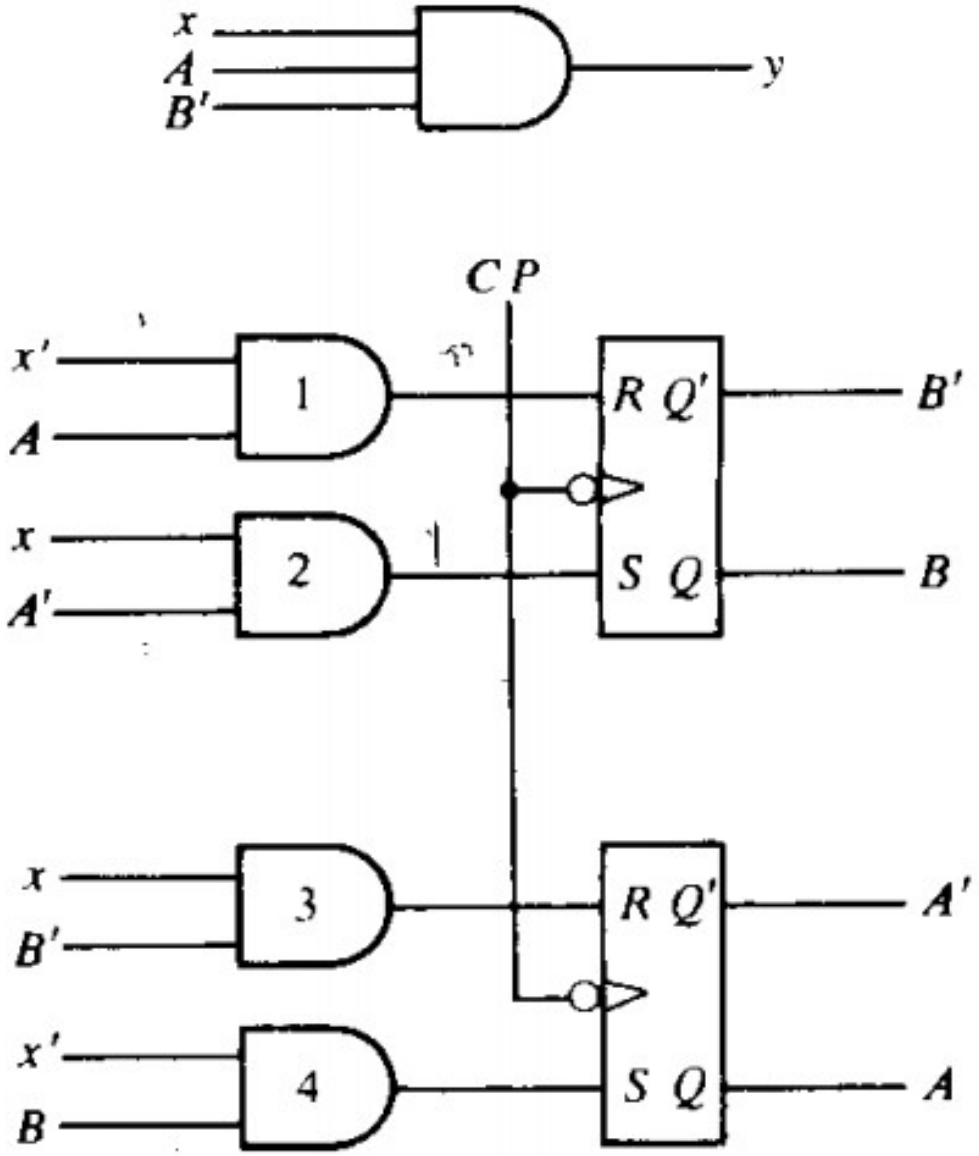
$RB = Ax'$



$y = AB'x$

Reference: M. Morris Mano, "Digital Logic and Computer Design", PHI

Sequential Circuit Design Using RS-FF (contd.)



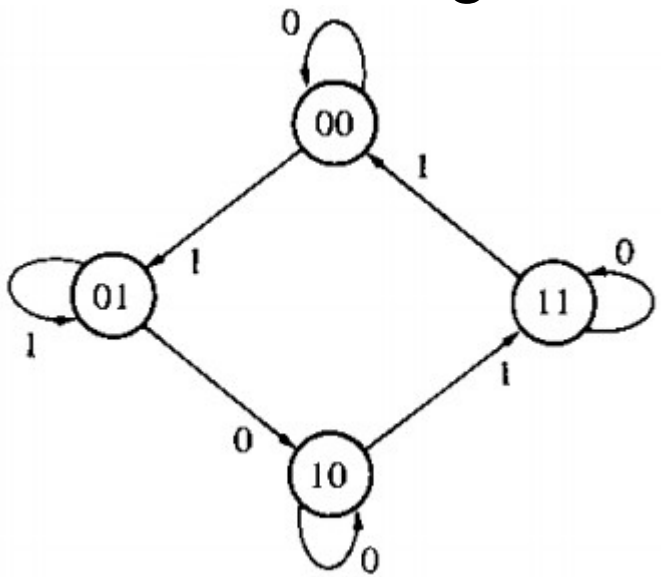
$$SA = Bx' \qquad RA = B'x$$

$$SB = A'x \qquad RB = Ax'$$

$$y = AB'x$$

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Sequential Circuit Design Using JK-FF



State table

Present state		Next state			
		$x = 0$		$x = 1$	
A	B	A	B	A	B
0	0	0	0	0	1
0	1	1	0	0	1
1	0	1	0	1	1
1	1	1	1	0	0

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

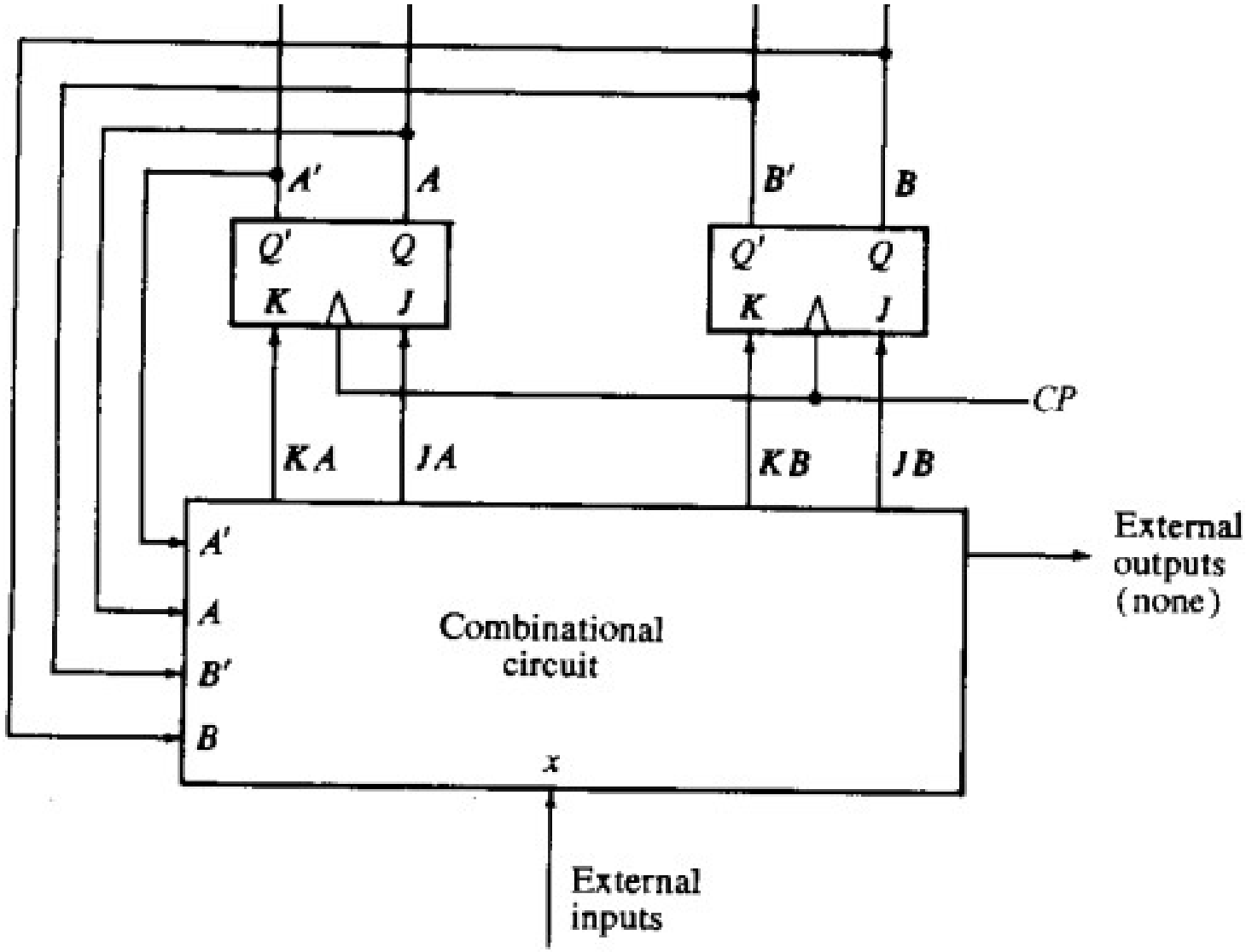
Sequential Circuit Design Using JK-FF (contd.)

Excitation table

Inputs of combinational circuit			Next state		Outputs of combinational circuit			
Present state		Input			Flip-flop inputs			
A	*B*	*x*	*A*	*B*	*JA*	*KA*	*JB*	*KB*
0	0	0	0	0	0	*X*	0	*X*
0	0	1	0	1	0	*X*	1	*X*
0	1	0	1	0	1	*X*	*X*	1
0	1	1	0	1	0	*X*	*X*	0
1	0	0	1	0	*X*	0	0	*X*
1	0	1	1	1	*X*	0	1	*X*
1	1	0	1	1	*X*	0	*X*	0
1	1	1	0	0	*X*	1	*X*	1

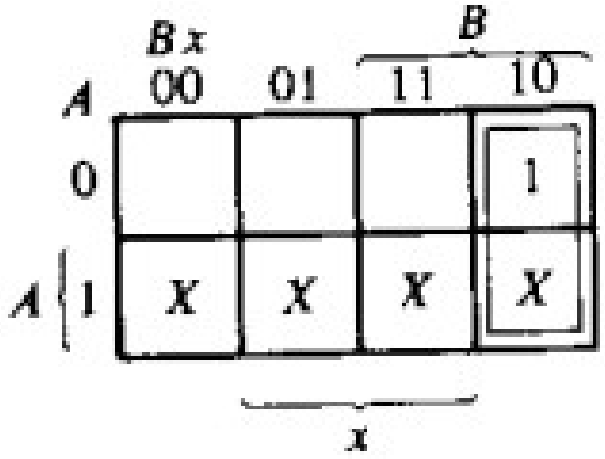
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Block Diagram of the Sequential Circuit

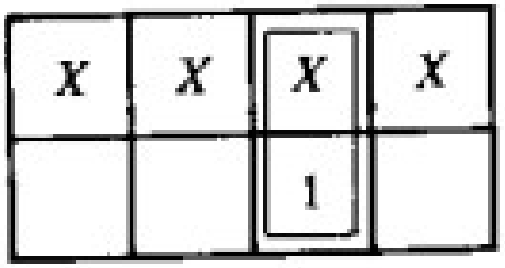


Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

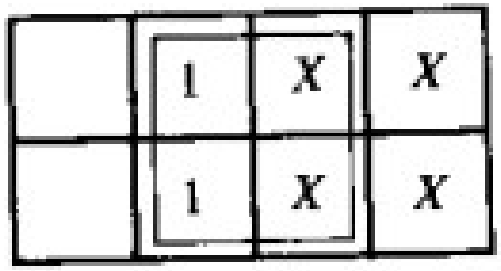
Maps for the Combinational Circuit



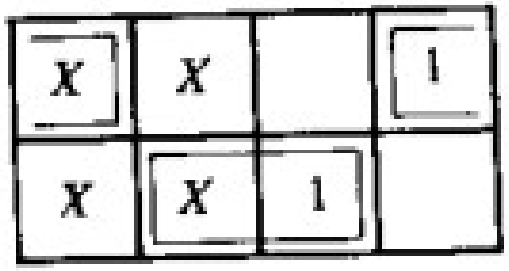
$JA = Bx'$



$KA = Bx$



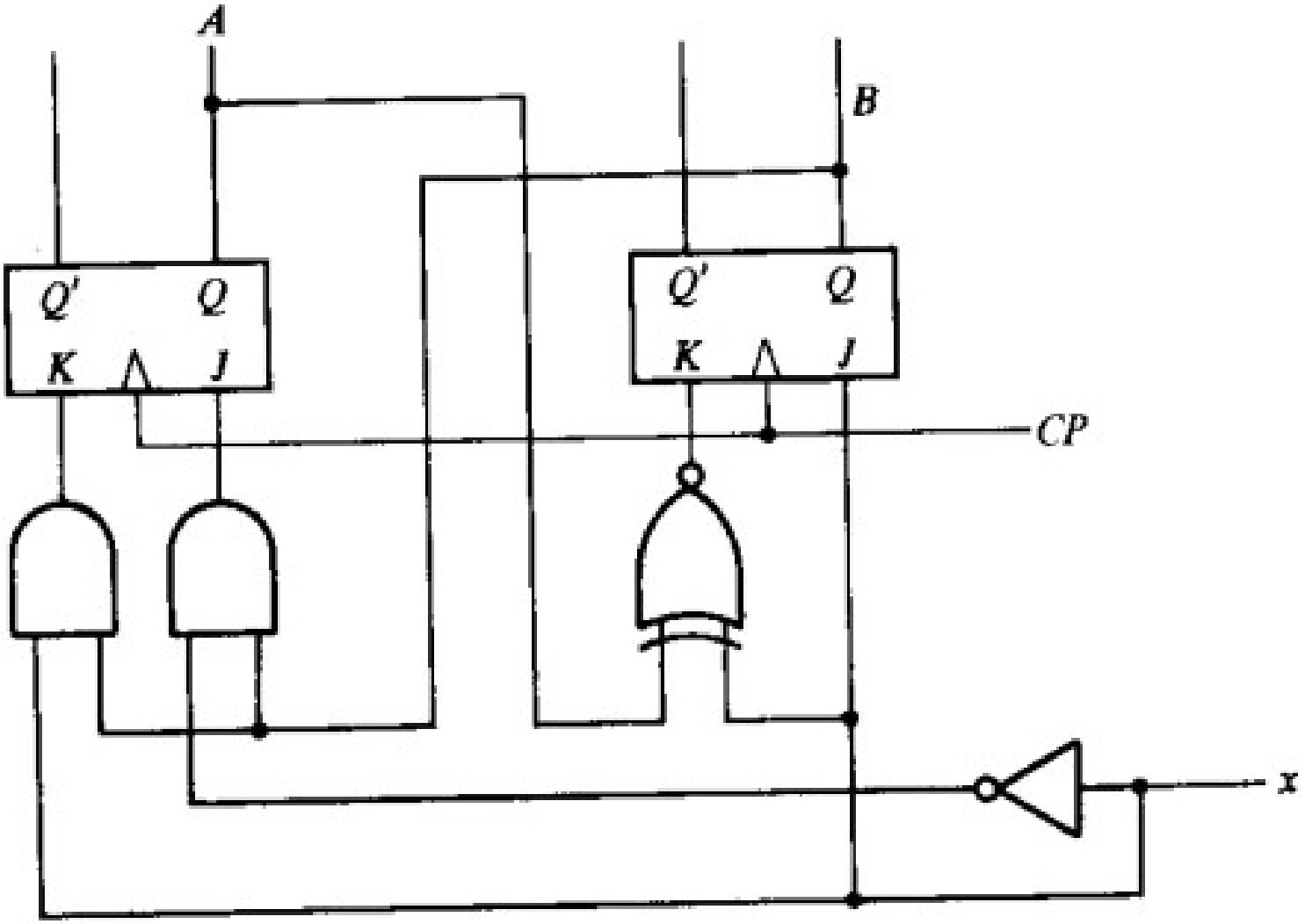
$JB = x$



$KB = A \odot x$

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

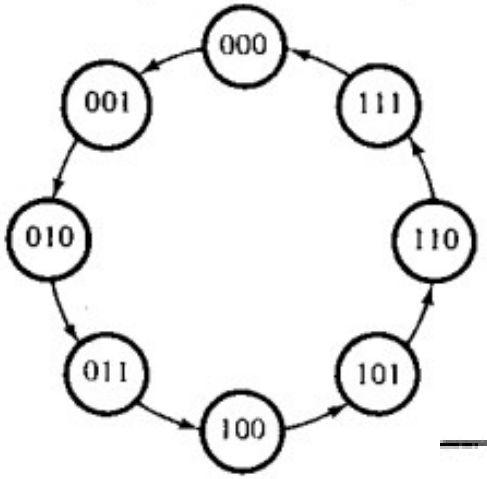
Logic Diagram of the Sequential Circuit



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

3-bit Binary Counter

State diagram of a 3-bit binary counter

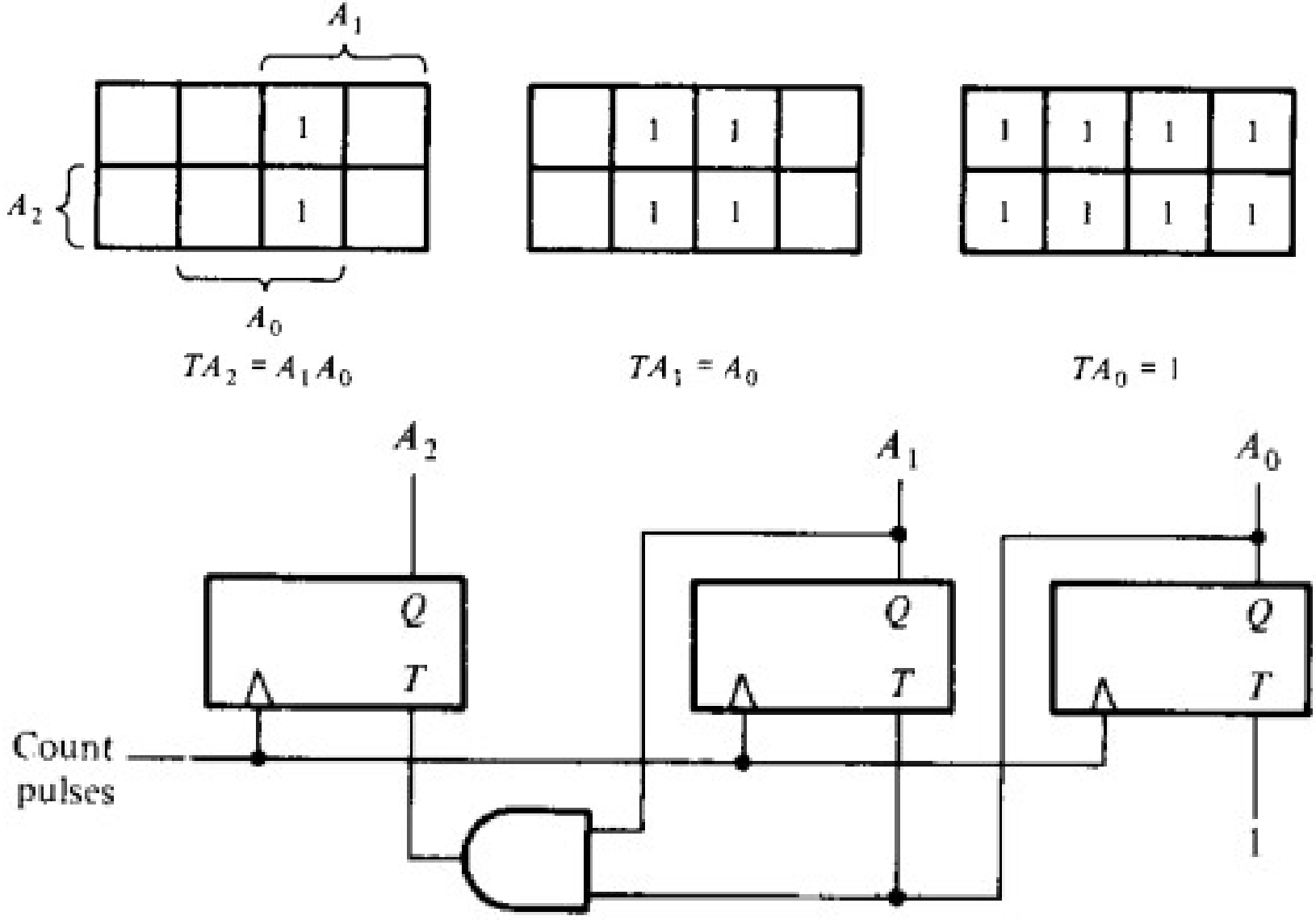


Excitation table for a 3-bit binary counter

Count sequence			Flip-flop inputs		
A_2	A_1	A_0	TA_2	TA_1	TA_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	1	1	1

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

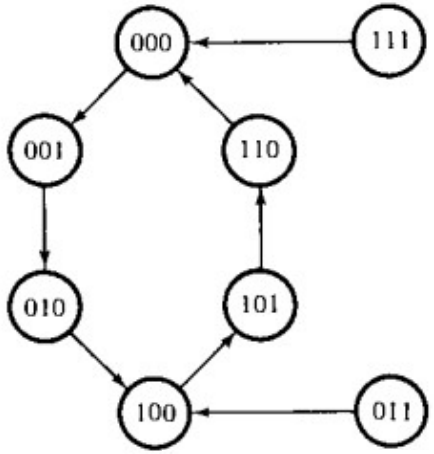
3-bit Binary Counter (contd.)



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Self Starting Counters

- A counter that can start from any state, but eventually reaches the normal count sequence

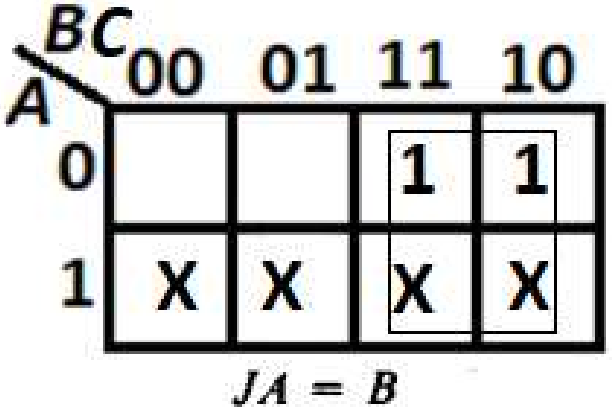


$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

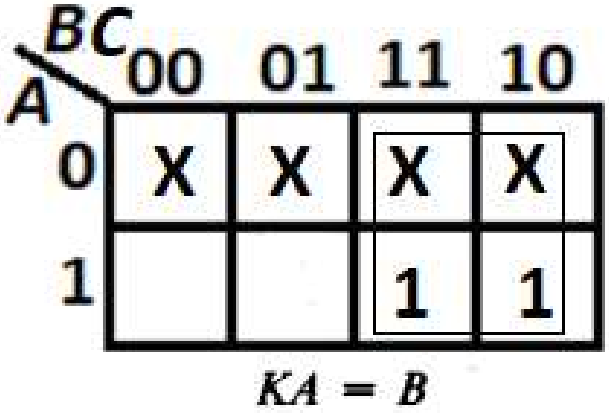
Present Count State			Next Count State			Required Flip-Flop Inputs					
A	*B*	*C*	*A*	*B*	*C*	*JA*	*KA*	*JB*	*KB*	*JC*	*KC*
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X
1	1	1	0	0	0	X	1	X	1	X	1

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

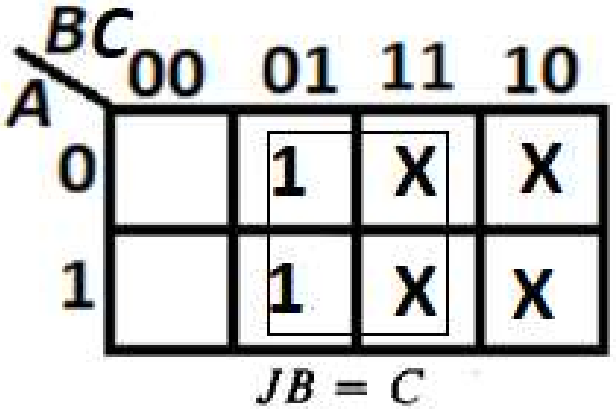
K-Maps for the Self Starting Counter



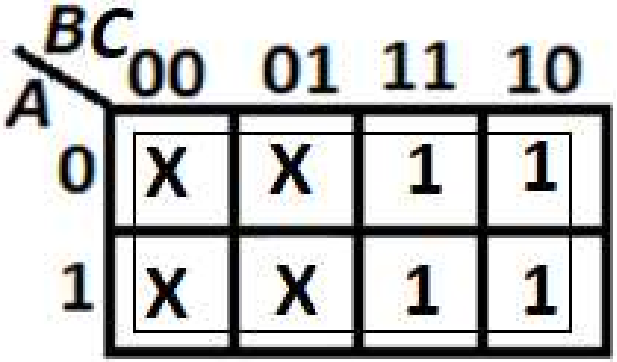
$JA = B$



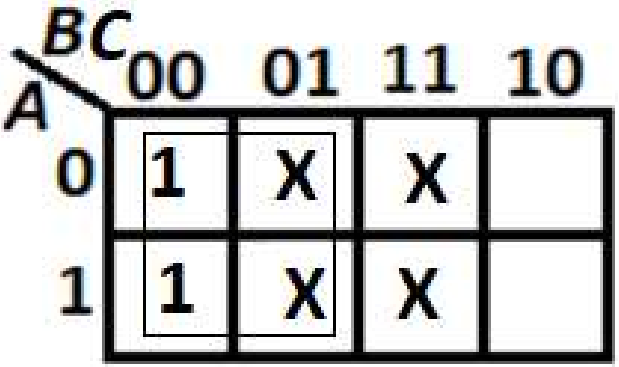
$KA = B$



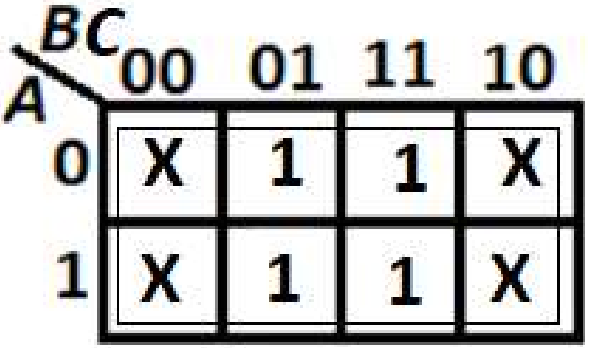
$JB = C$



$KB = 1$



$JC = B'$



$KC = 1$

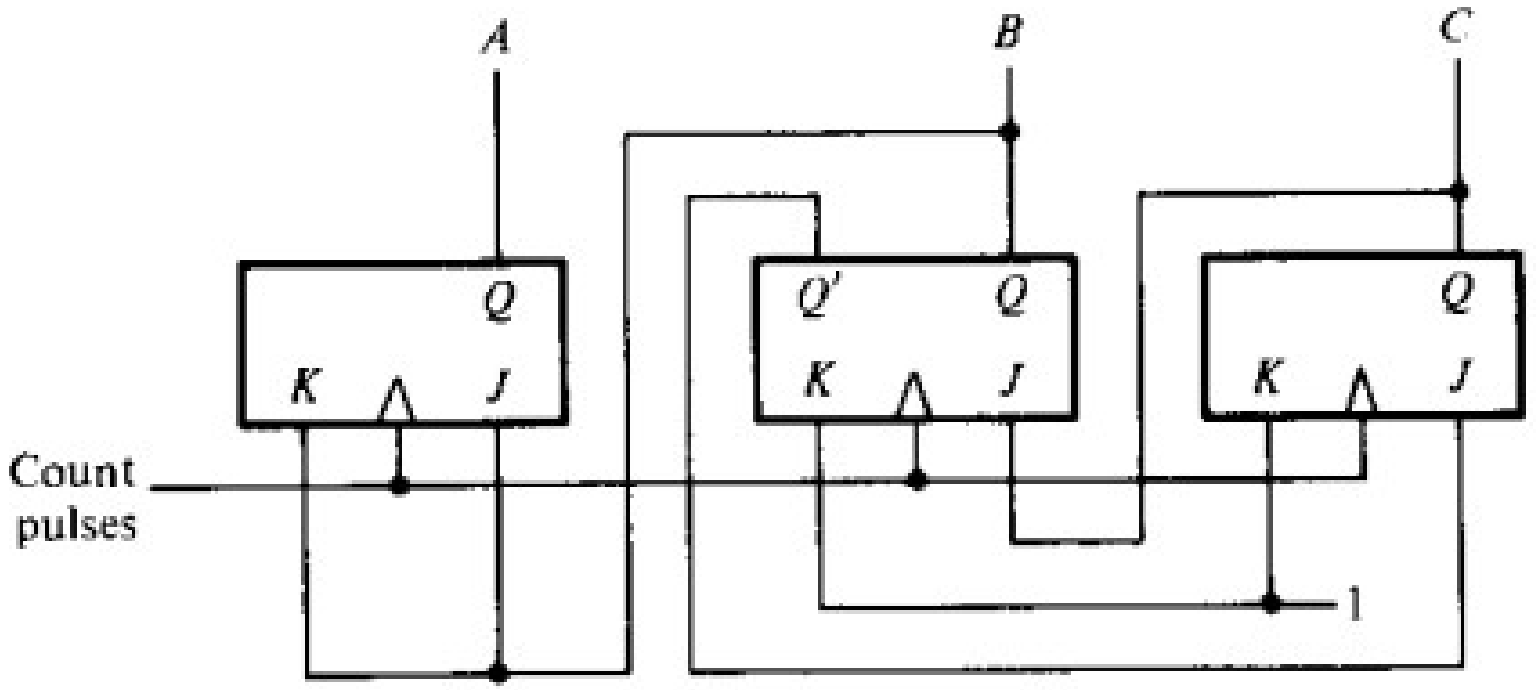
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Logic Diagram for the Self Starting Counter

$$JA = B \qquad KA = B$$

$$JB = C \qquad KB = 1$$

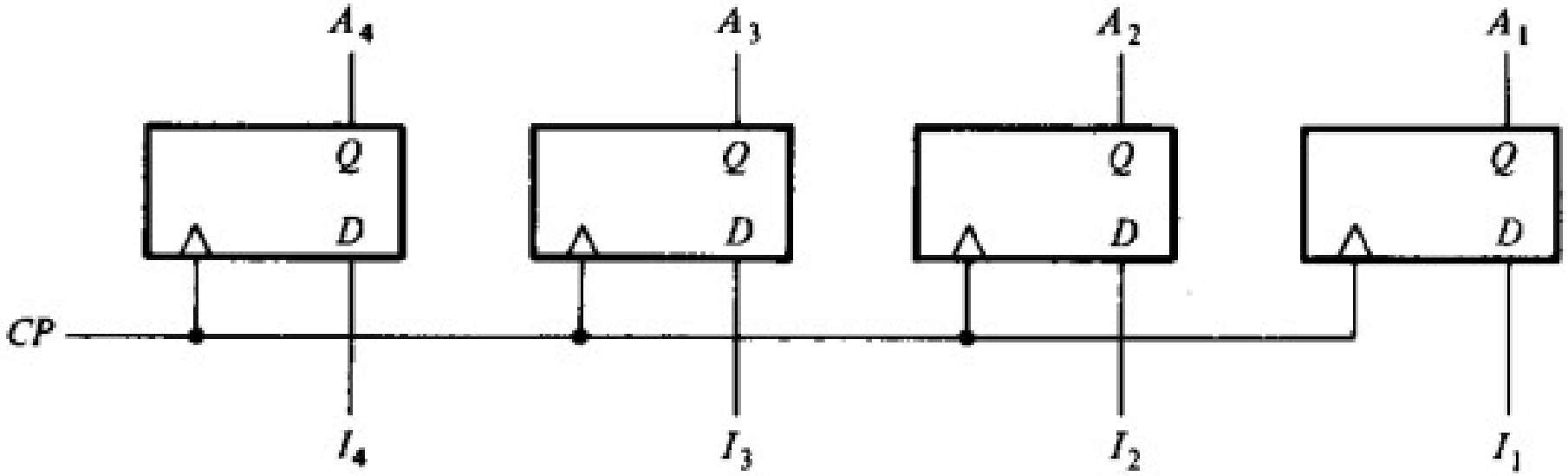
$$JC = B' \qquad KC = 1$$



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Registers

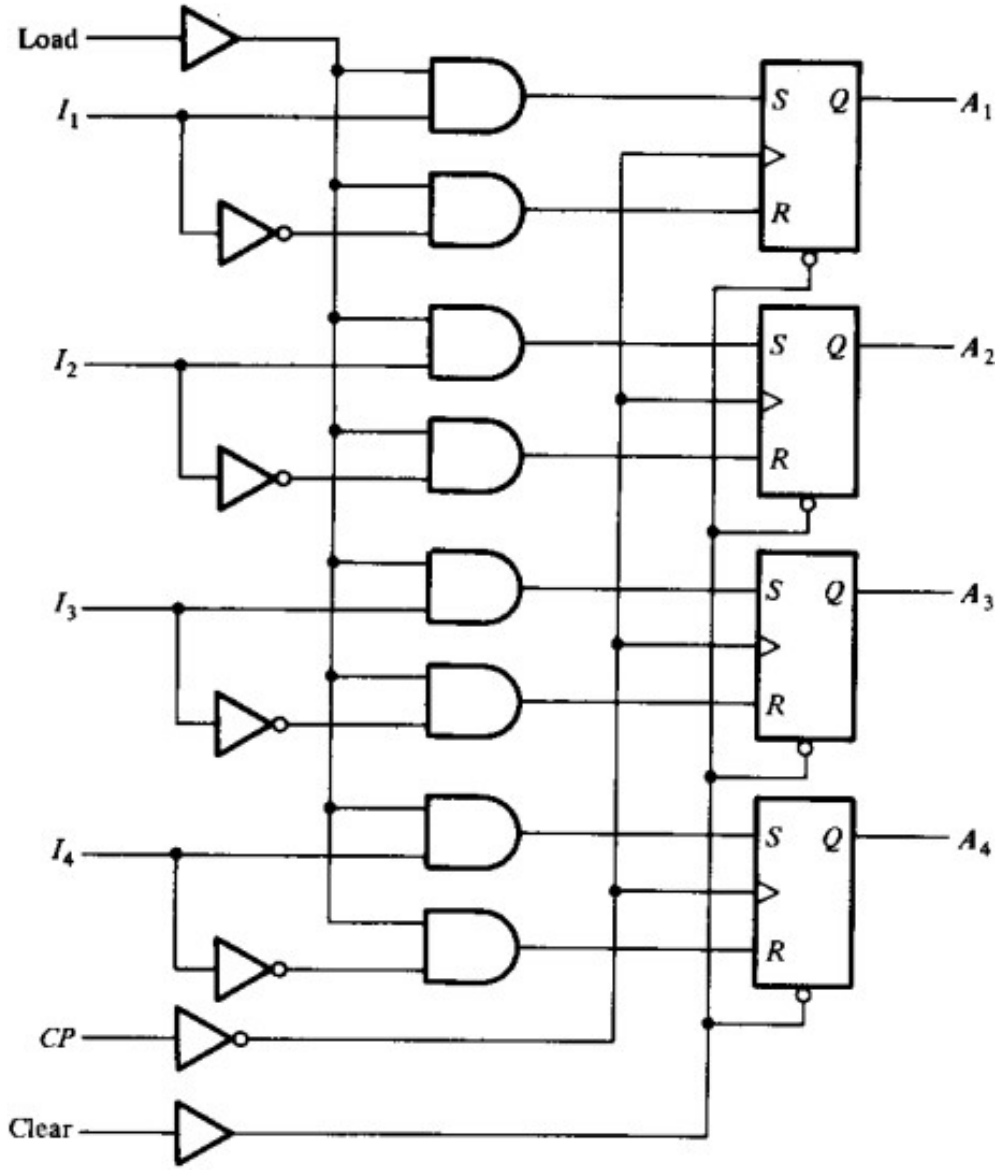
A *register* is a group of binary storage cells suitable for holding binary information. A group of flip-flops constitutes a register, since each flip-flop is a binary cell capable of storing one bit of information. An *n*-bit register has a group



4-bit register

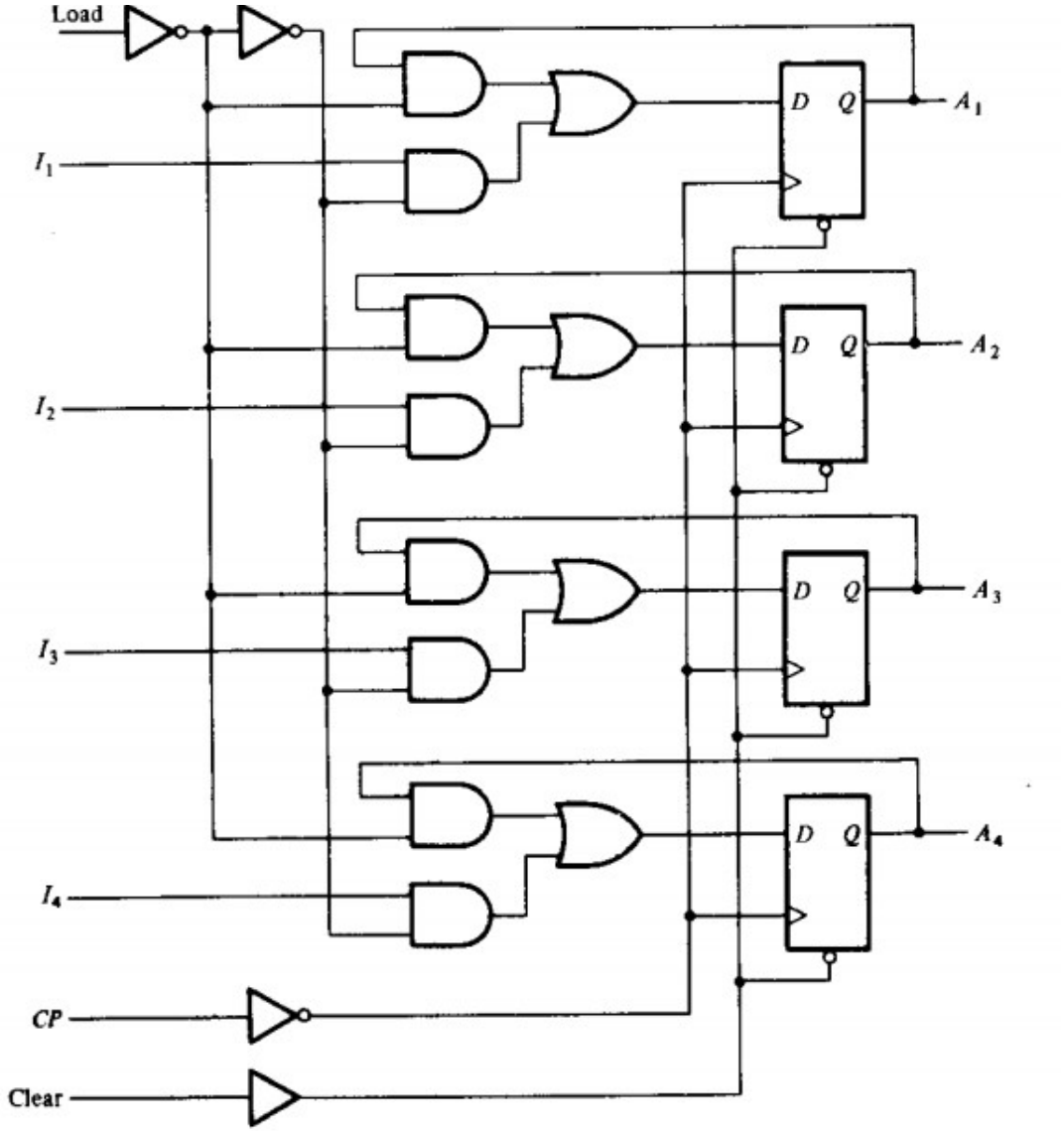
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

4-bit Register with Parallel Load



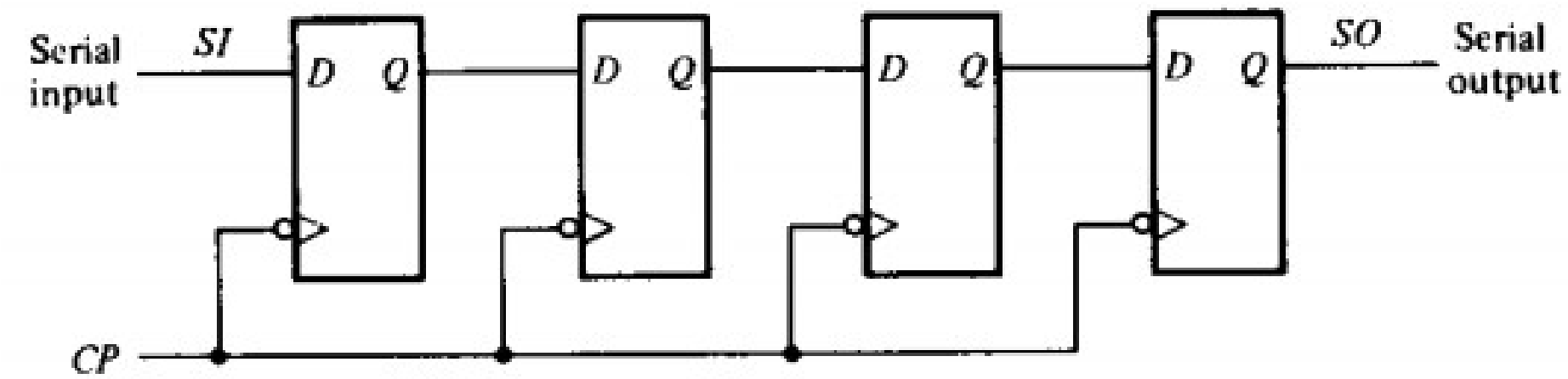
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Register with Parallel Load Using D-FF



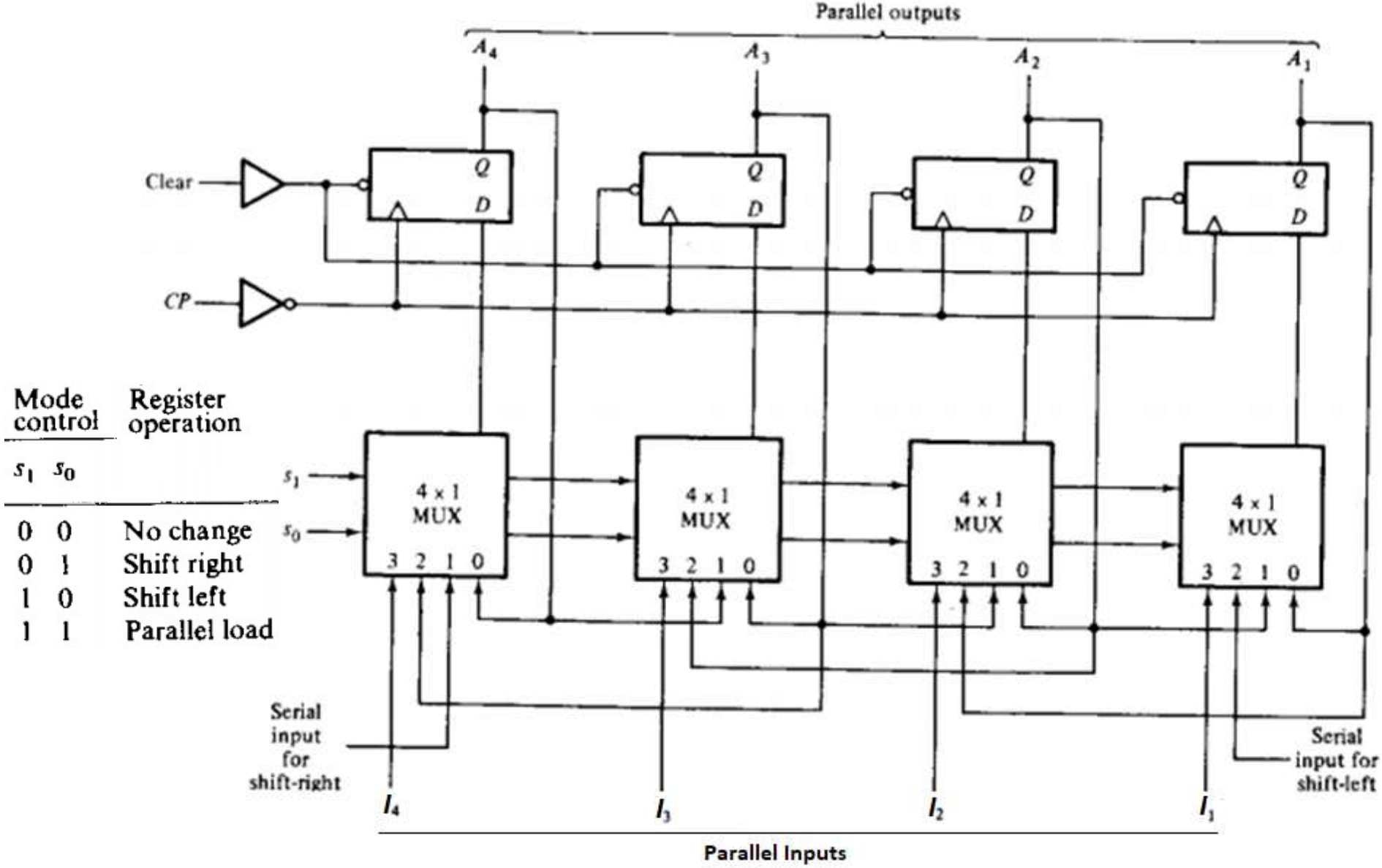
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Shift Registers



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

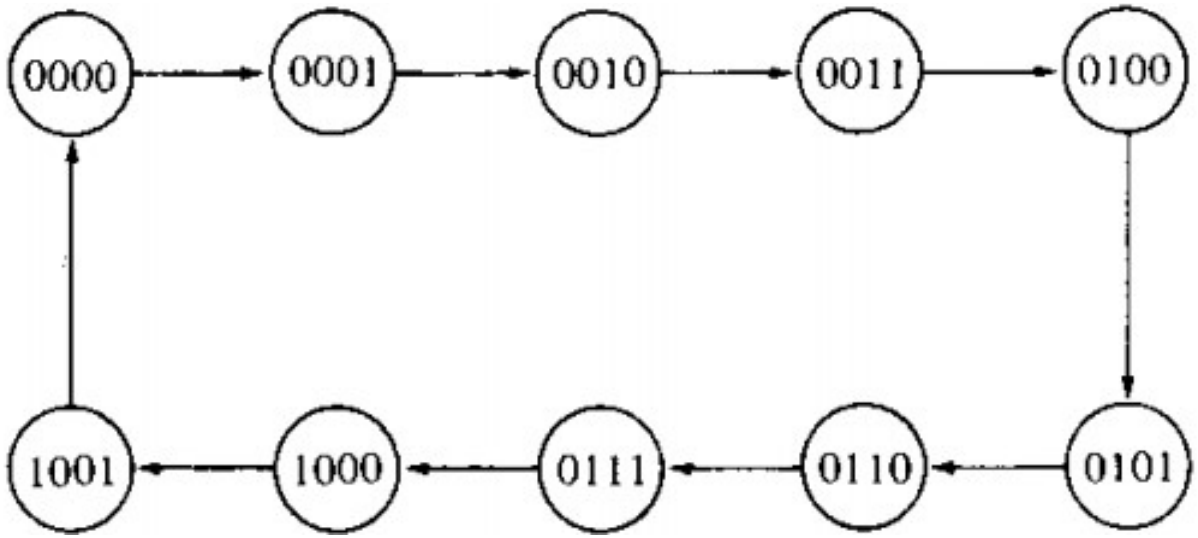
4-bit Bidirectional Shift Register with Parallel Load



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

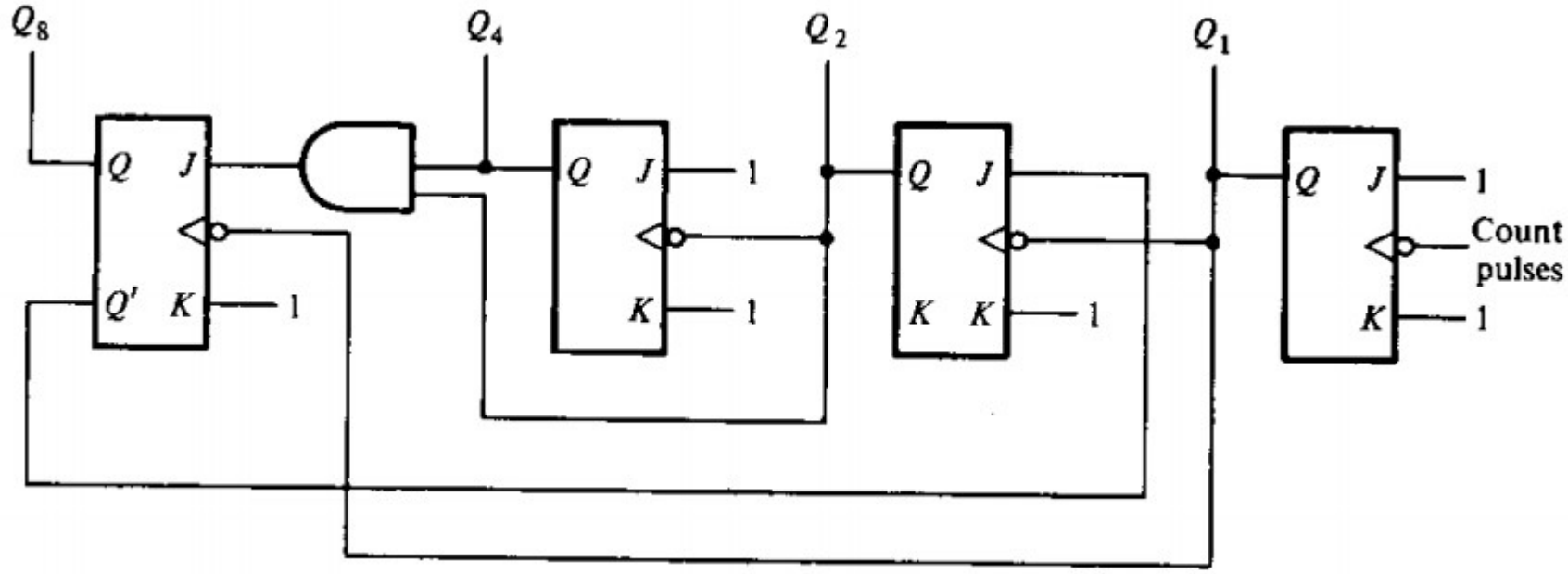
BCD Ripple Counter

- We have already seen the Mod-16 Ripple Counter
- The State Diagram of the BDC Ripple Counter is now given below
- A BCD Counter may also be called a Decade Counter



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

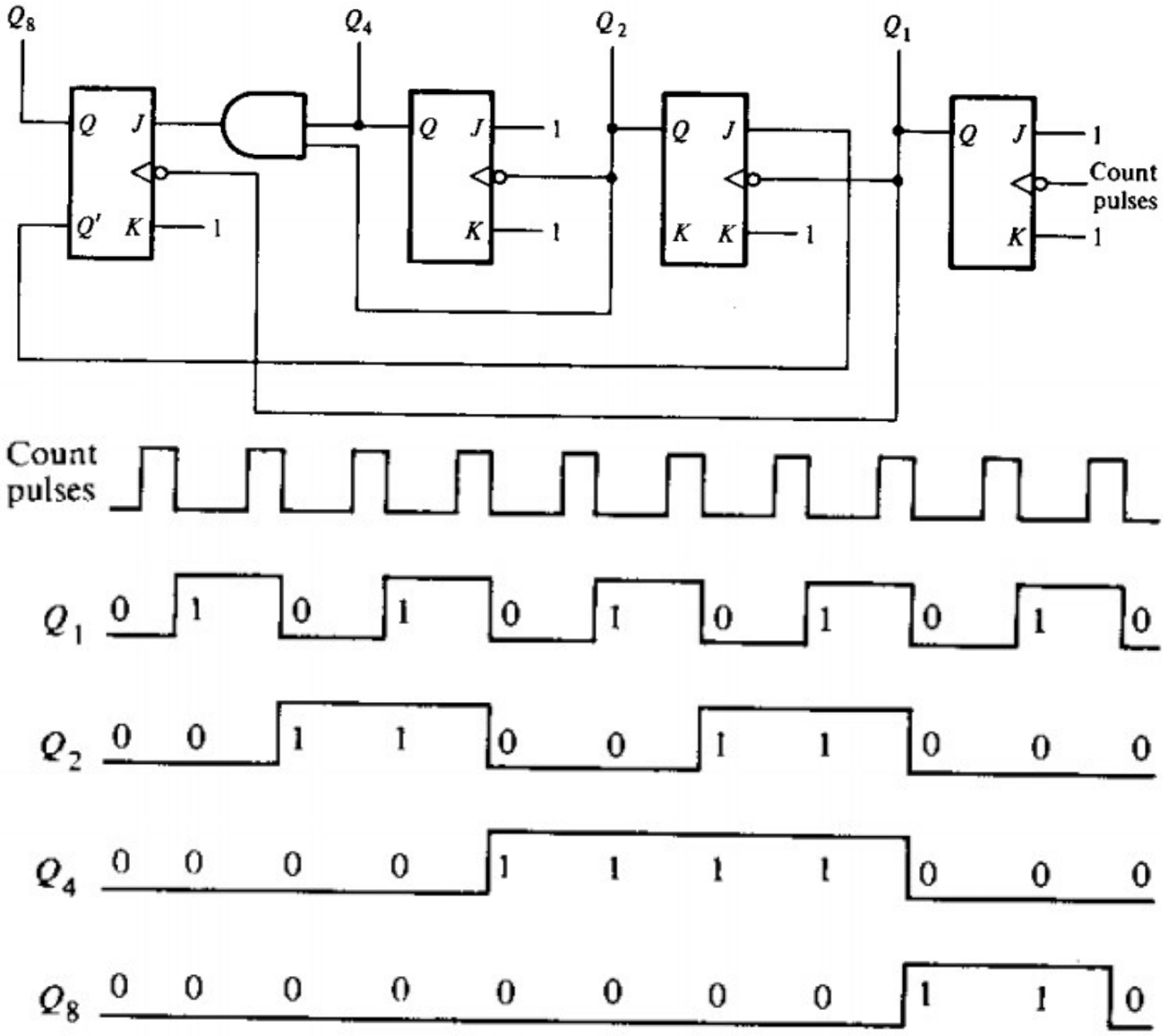
BCD Ripple Counter (contd.)



1. Q_1 is complemented on the negative edge of every count pulse.
2. Q_2 is complemented if $Q_8 = 0$ and Q_1 goes from 1 to 0. Q_2 is cleared if $Q_8 = 1$ and Q_1 goes from 1 to 0.
3. Q_4 is complemented when Q_2 goes from 1 to 0.
4. Q_8 is complemented when $Q_4Q_2 = 11$ and Q_1 goes from 1 to 0. Q_8 is cleared if either Q_4 or Q_2 is 0 and Q_1 goes from 1 to 0.

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

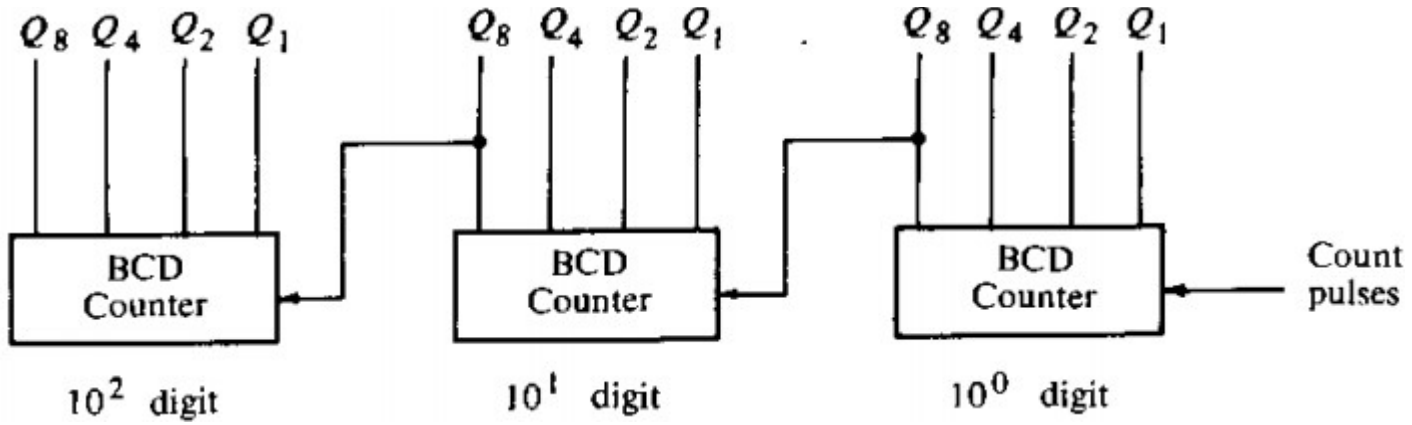
Timing Diagram of the BCD Ripple Counter



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

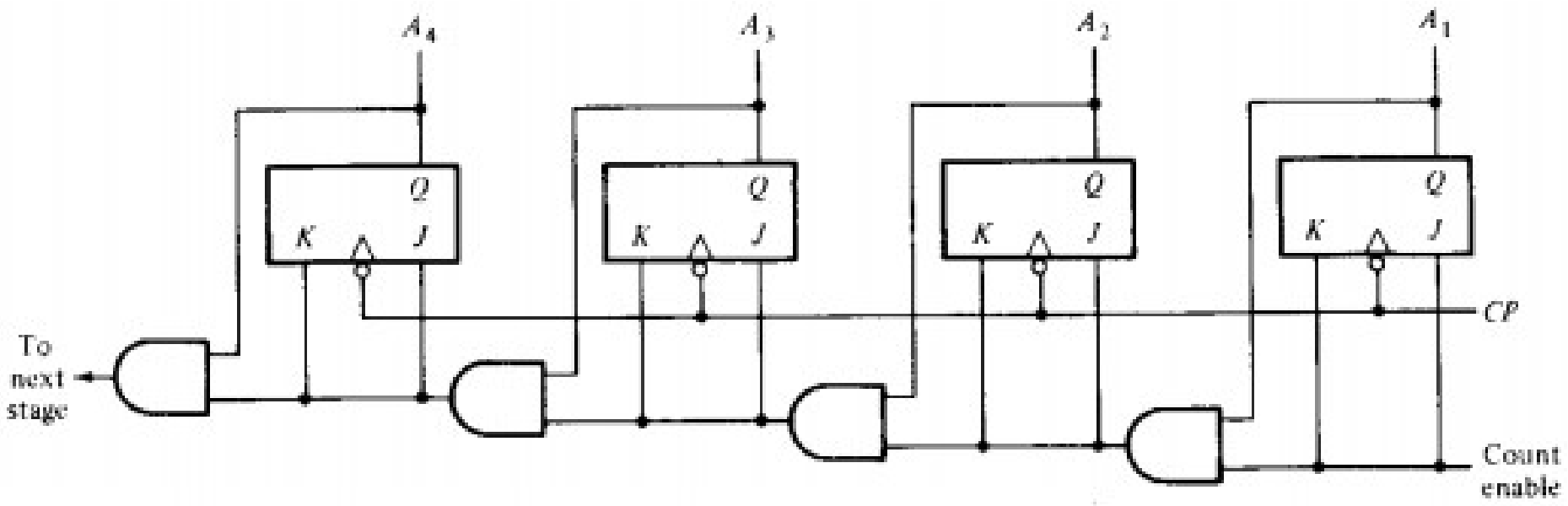
3-Decade Decimal BCD Counter

- We have seen the asynchronous (ripple) BCD Counter just now
- Design the synchronous BCD counter also
- Using any of the above two, we can design *n* – Decade Counter
- We can use the BCD-to-7 Segment Decoder at the output of this *n* – Decade Counter



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

4-bit Synchronous Binary Counter



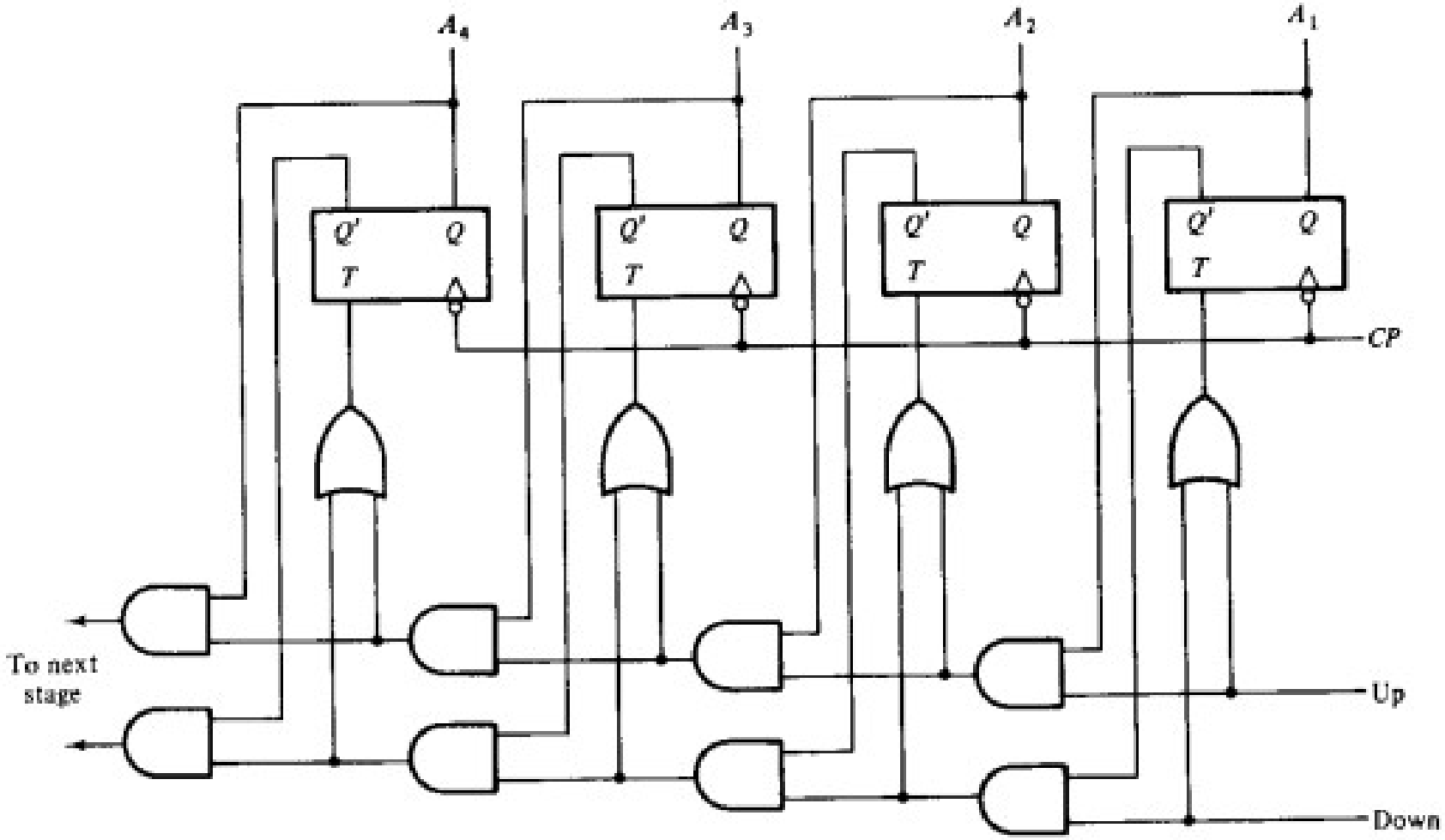
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Binary Count Sequence

A_4	A_3	A_2	A_1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

4-bit Up-Down Binary Counter



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Synchronous BCD Counter

Count sequence				Flip-flop inputs				Output carry
Q_8	Q_4	Q_2	Q_1	TQ_8	TQ_4	TQ_2	TQ_1	y
0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	1	0
0	0	1	0	0	0	0	1	0
0	0	1	1	0	1	1	1	0
0	1	0	0	0	0	0	1	0
0	1	0	1	0	0	1	1	0
0	1	1	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1	0
1	0	0	1	1	0	0	1	1

$$TQ_1 = 1$$

$$TQ_2 = Q_8'Q_1$$

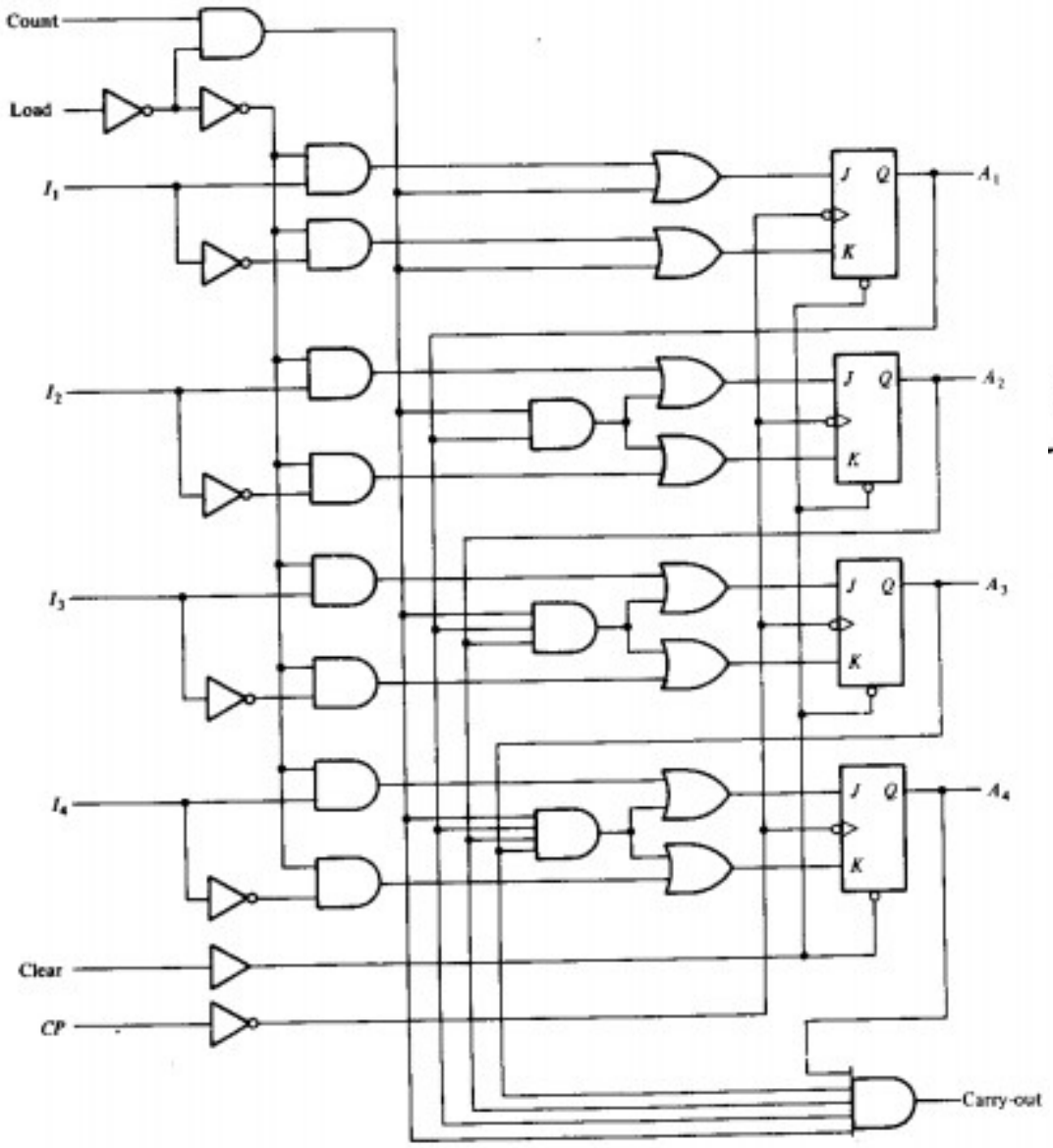
$$TQ_4 = Q_2Q_1$$

$$TQ_8 = Q_8Q_1 + Q_4Q_2Q_1$$

$$y = Q_8Q_1$$

Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

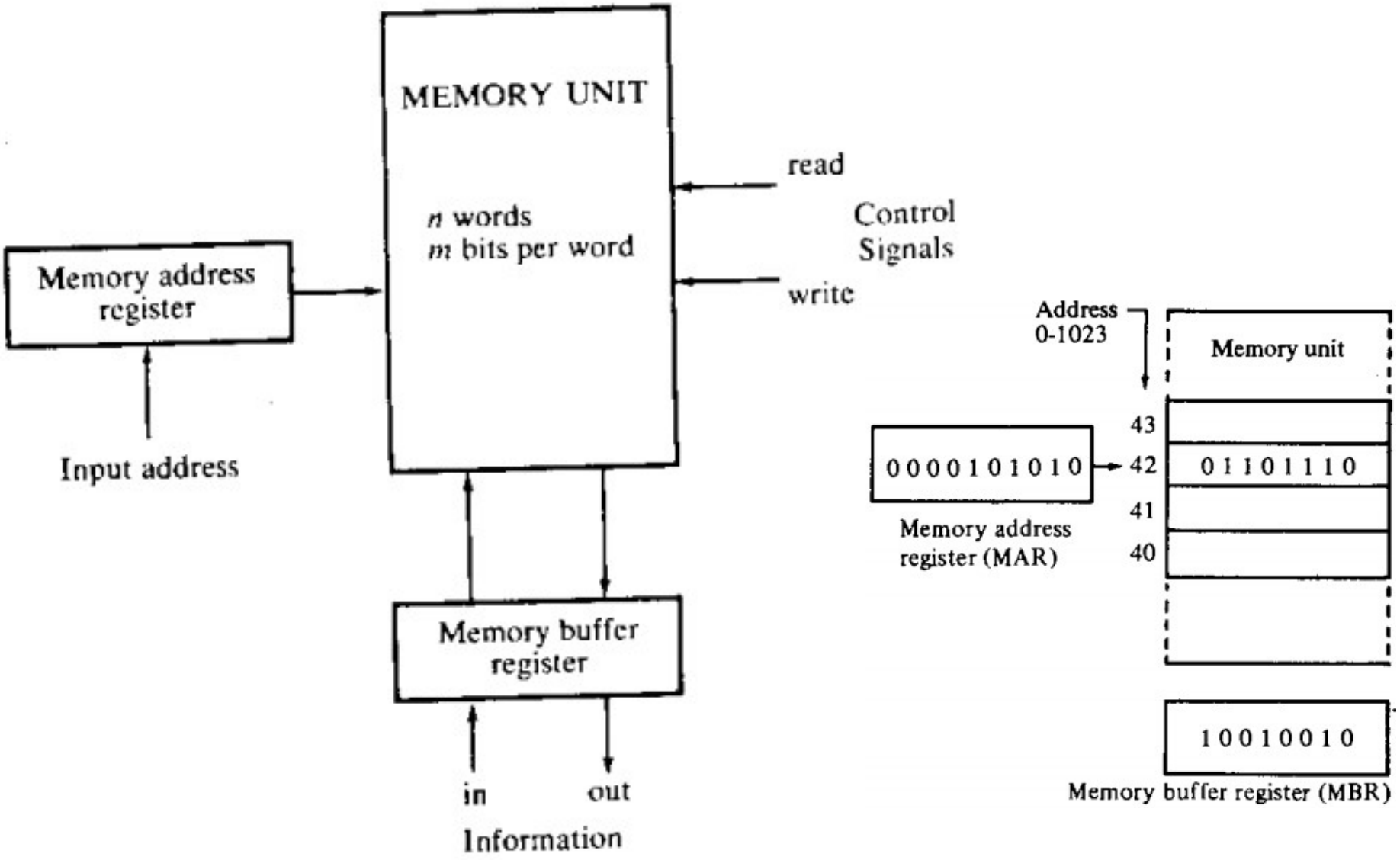
Binary Counter with Parallel Load



Clear	*CP*	Load	Count	Function
0	*X*	*X*	*X*	Clear to 0
1	*X*	0	0	No change
1	↑	1	*X*	Load inputs
1	↑	0	1	Count next binary state

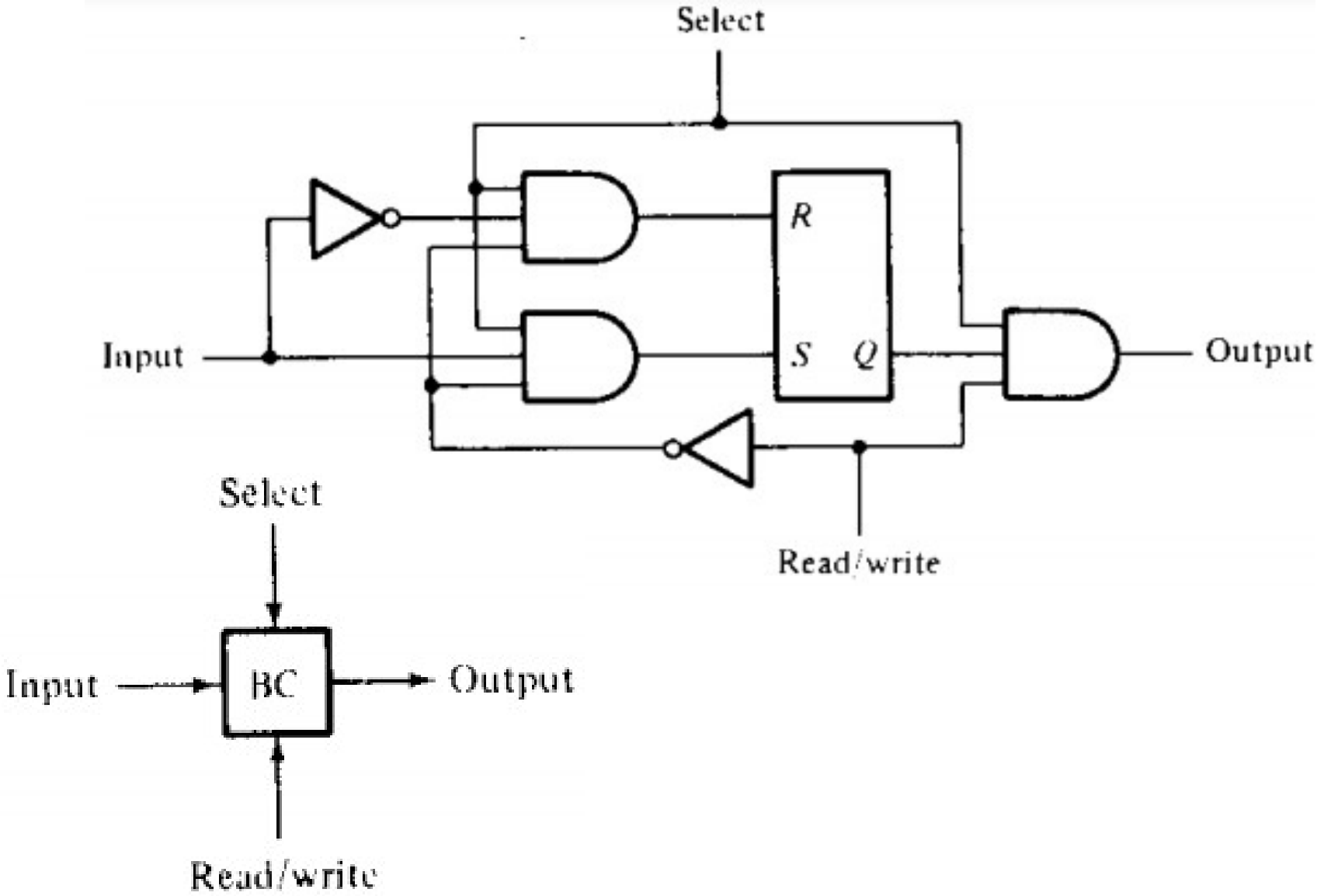
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Memory Unit



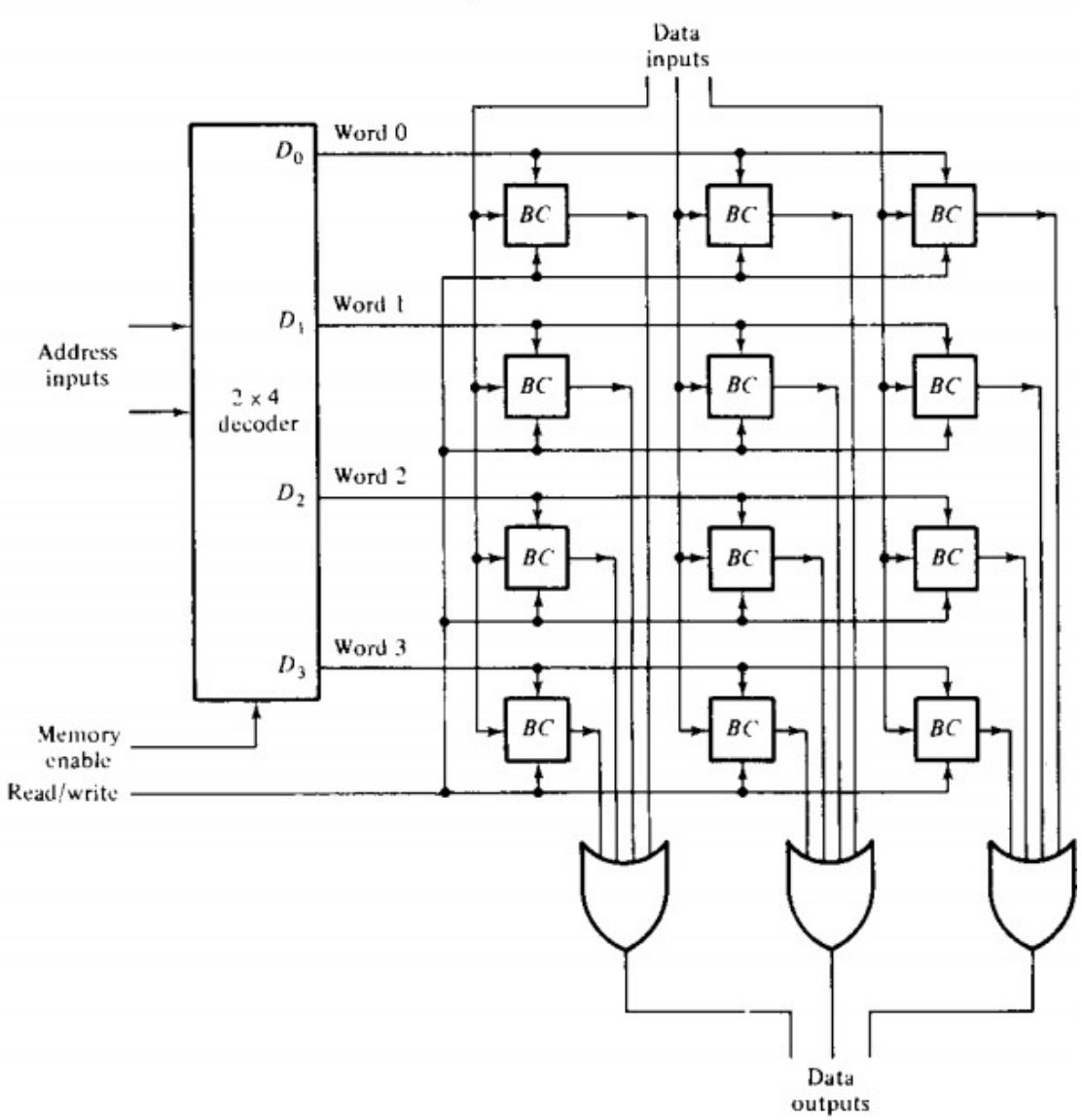
Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Memory Cell



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Memory Organization



Reference: M. Morris Mano, “Digital Logic and Computer Design”, PHI

Unit-III Completed

UNIT III.

SEQUENTIAL LOGIC

Definition and state representation of Flip-Flops, RS, D, JK-M/S, their working characteristics, State Tables, Excitation Tables and triggering. Asynchronous and Synchronous Counters-Design and Analysis, Counter Applications, Description and Operations of Shift Registers, Shift Register/Counters.

Course Outcome

3. Analyse and synthesize digital sequential units.