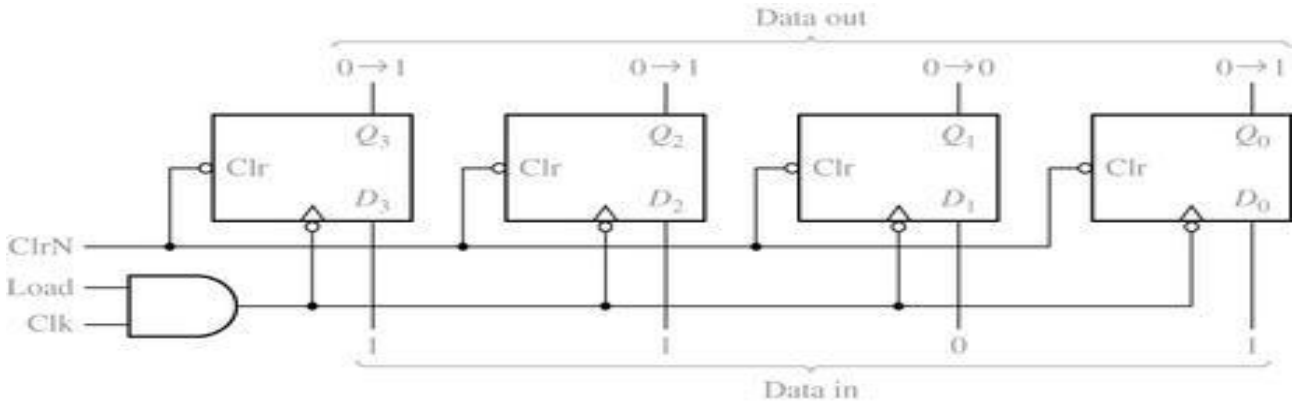
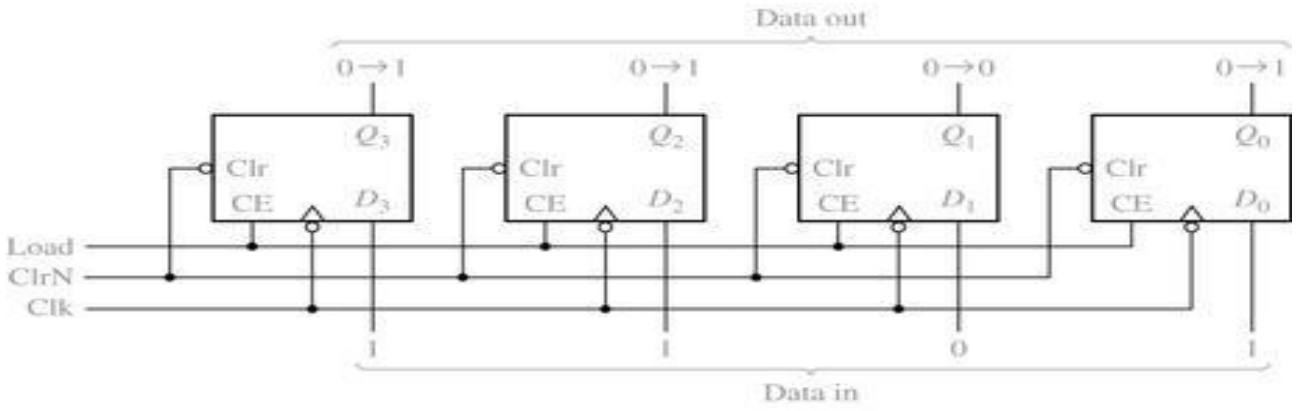
Lecture 11 Registers and Counters

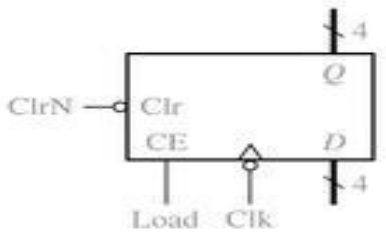
- Registers can be formed by a group of D flip-flops with a common clock.



(a) Using gated clock



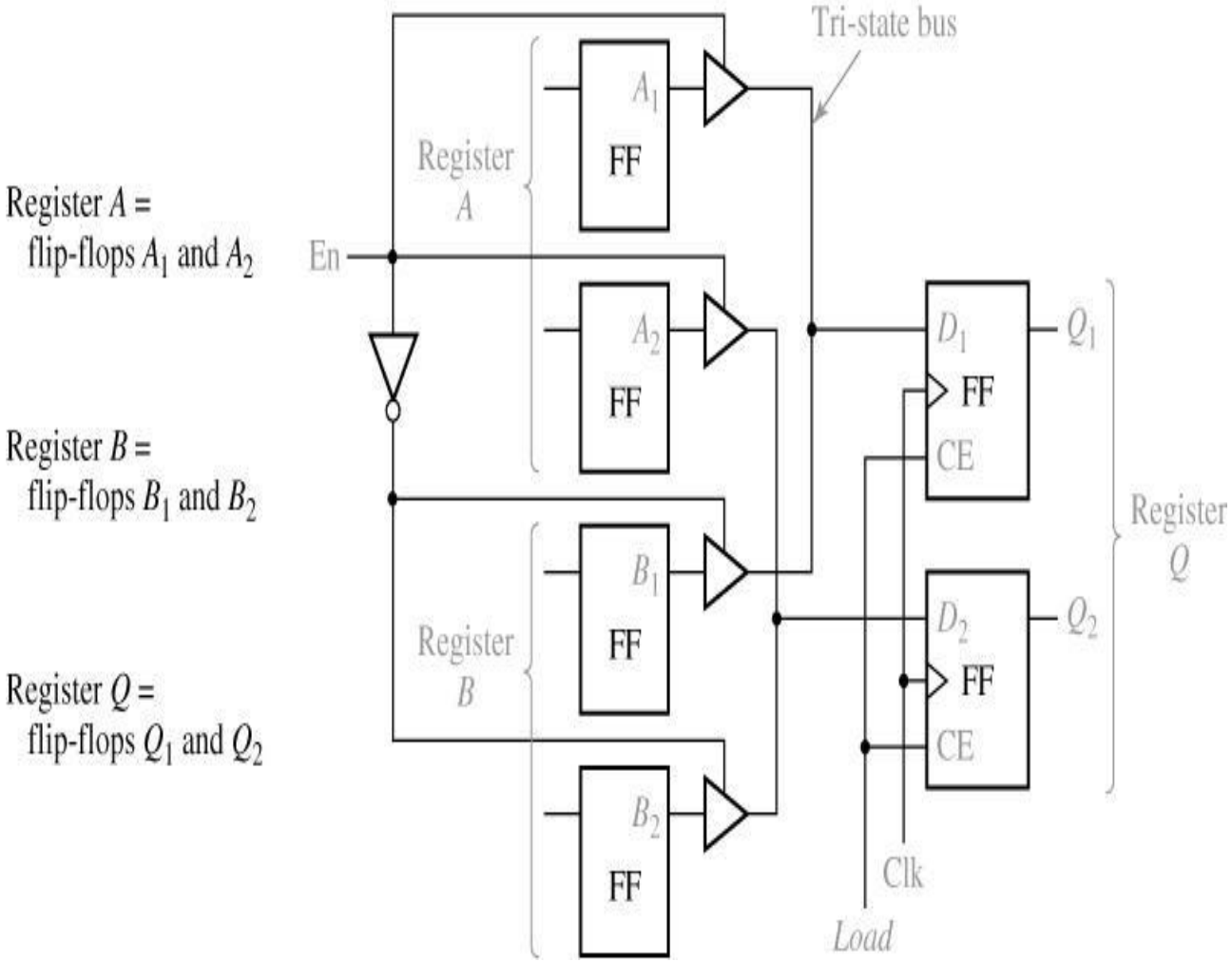
(b) With clock enable



(c) Symbol

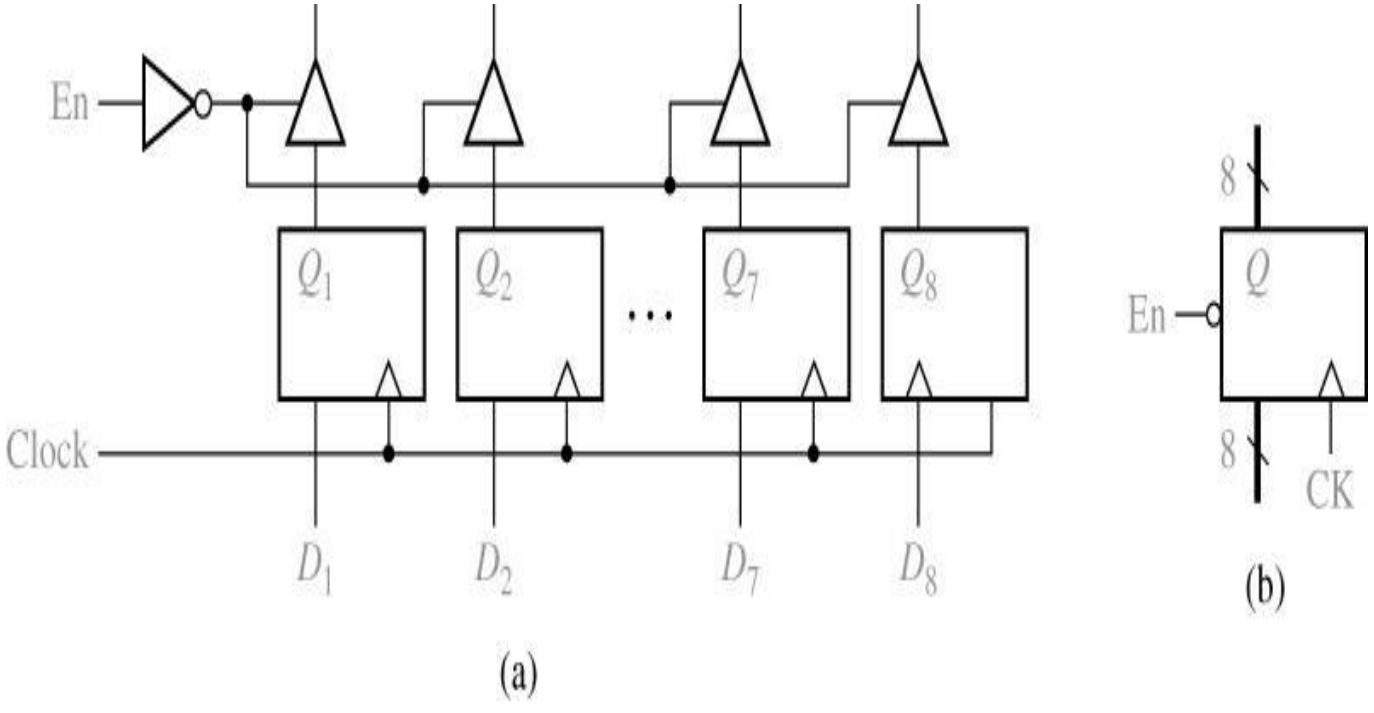
Data Transfer btw Registers

- Bus is a group of wires.
- Load: write control



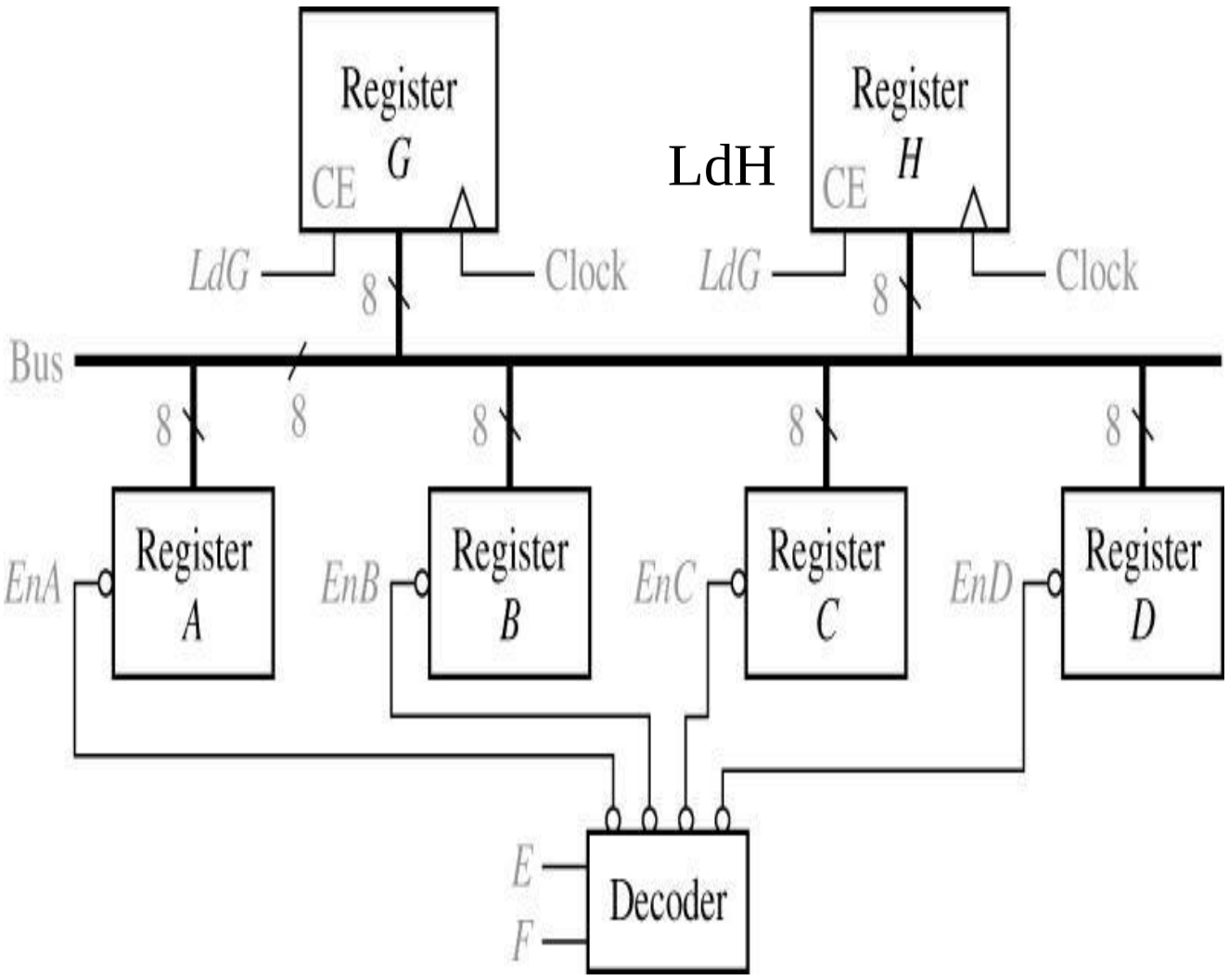
Register with Tri-State Output

- Physically connected, electrically disconnected.



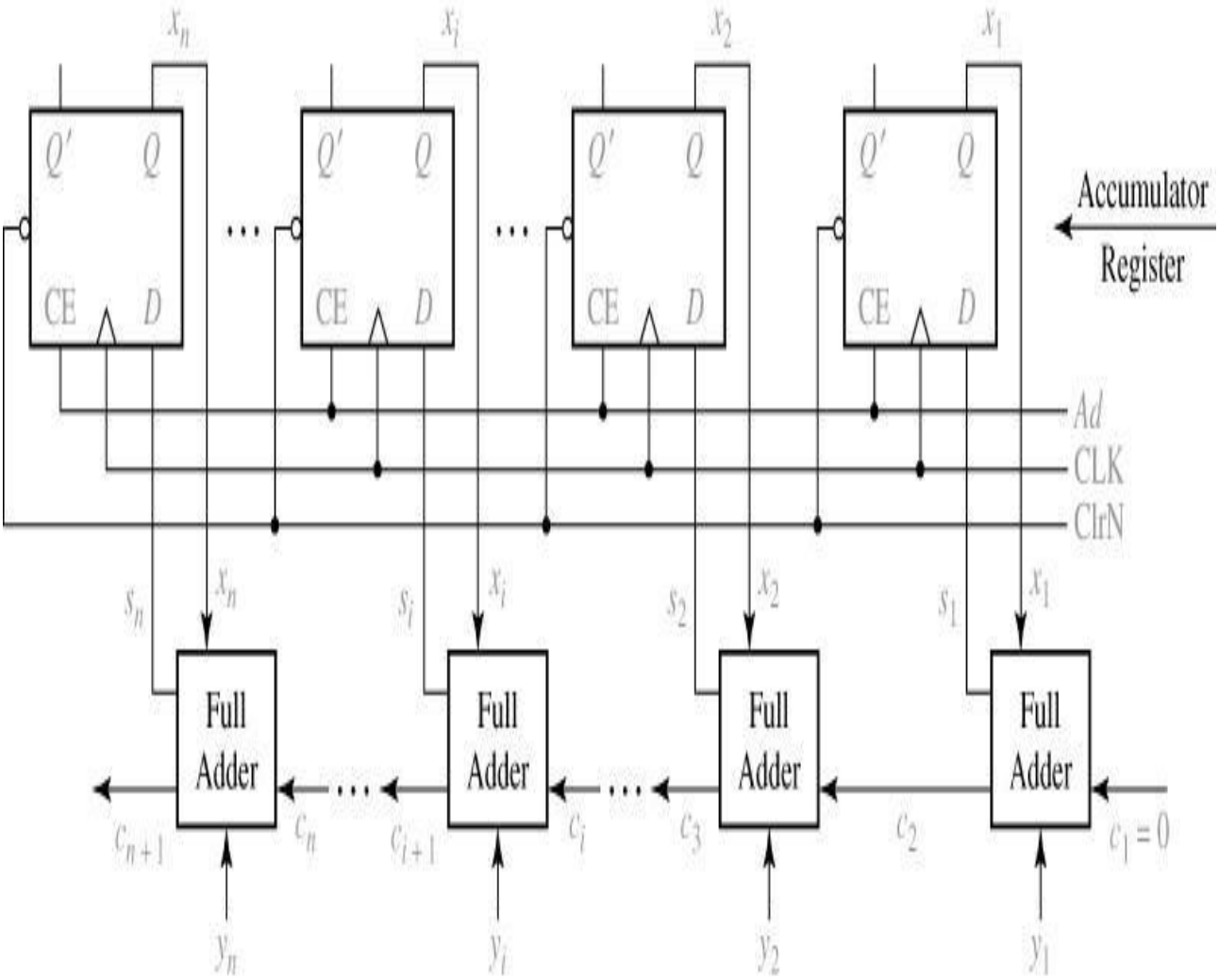
Tri-State Bus

- Physically connected, electrically disconnected.



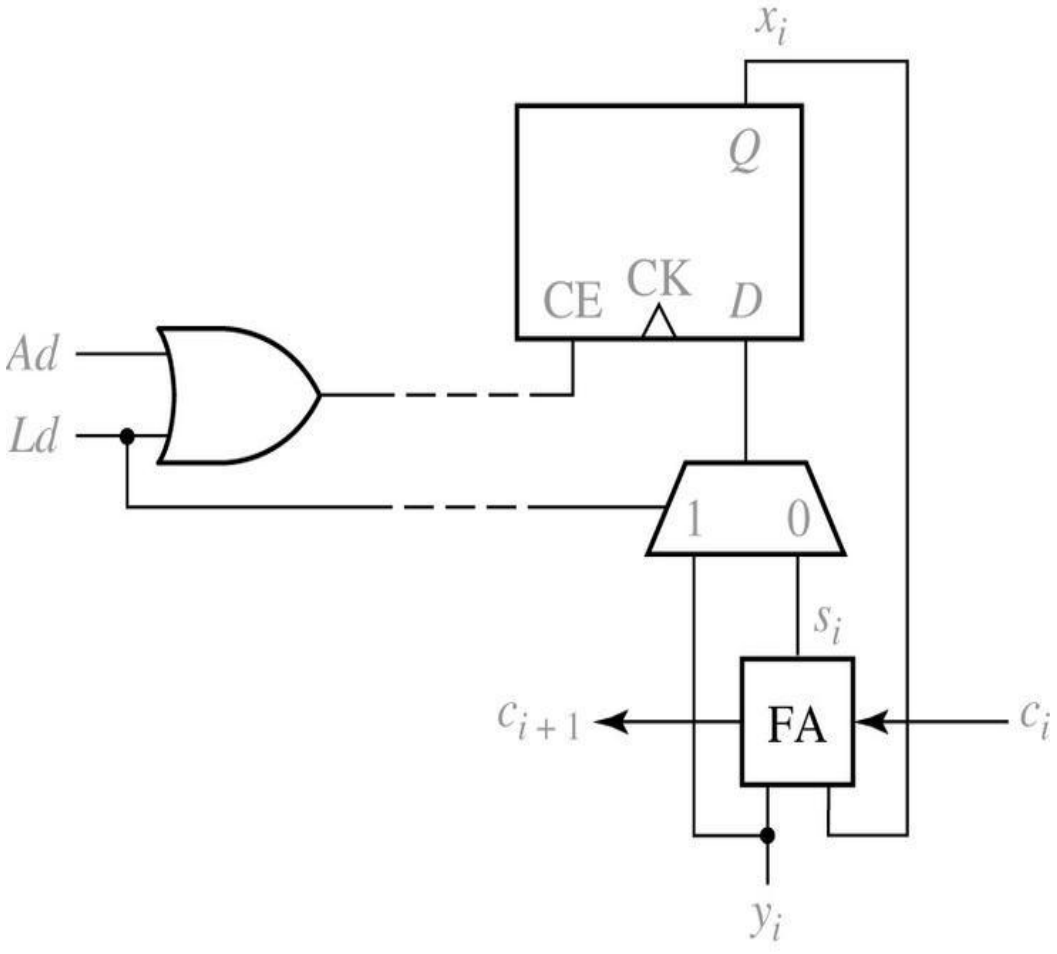
Accumulator

- Accumulator is a register. One operand is in the accumulator.
- *Ad* signal is used to load the adder outputs into the Acc FF on the rising clock edge.



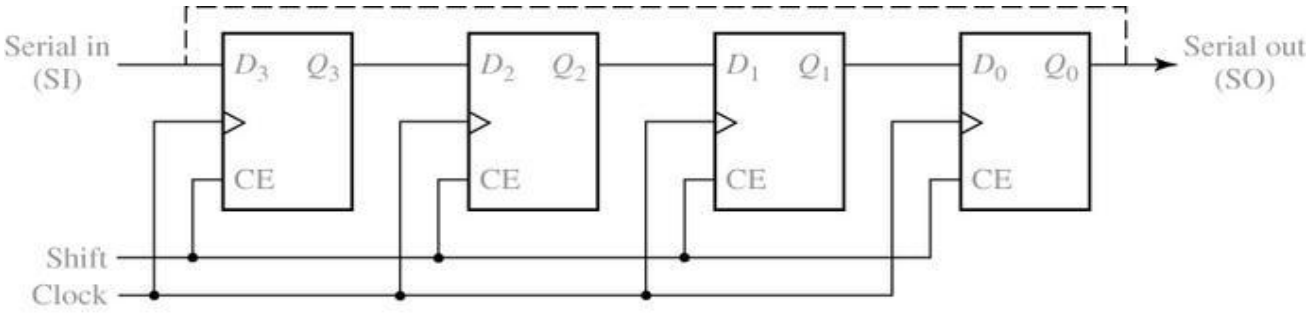
An Accumulator Cell

- How to load the first operand?
- Ld is used to select y which is loaded into the accumulator flip-flop.

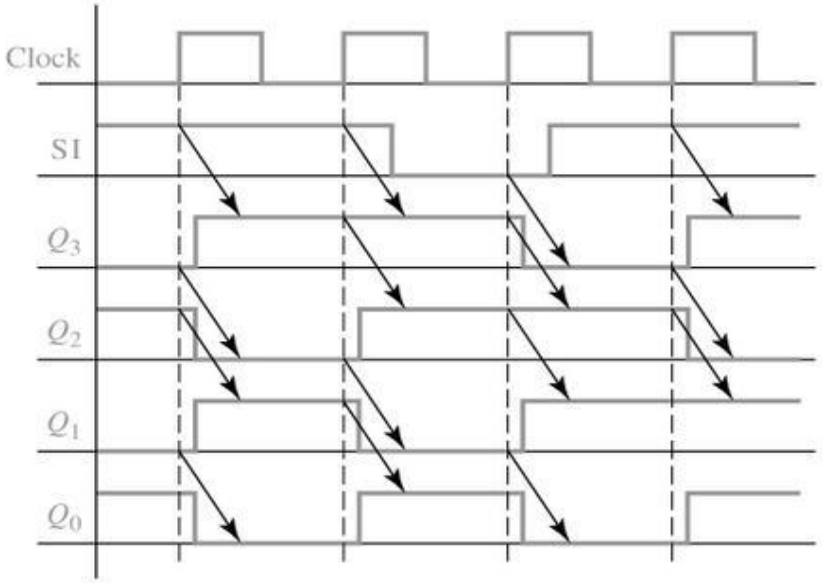


Shift Registers

- Right shift
- When enabled, shift occurs on the rising edge.
  - The output loaded into each FF is the value before the rising clock edge because of the propagation delay.



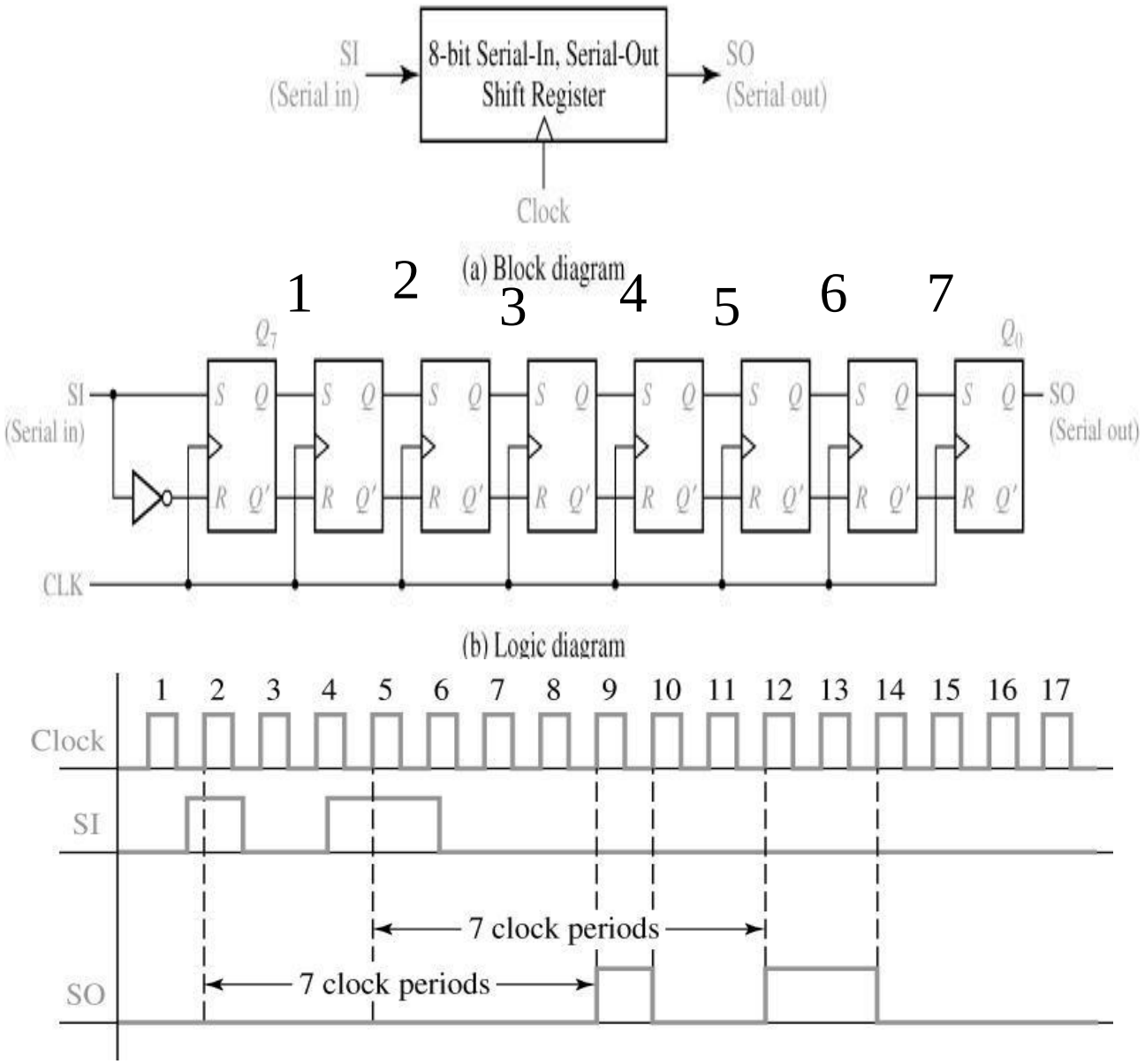
(a) Flip-flop connections



(b) Timing diagram

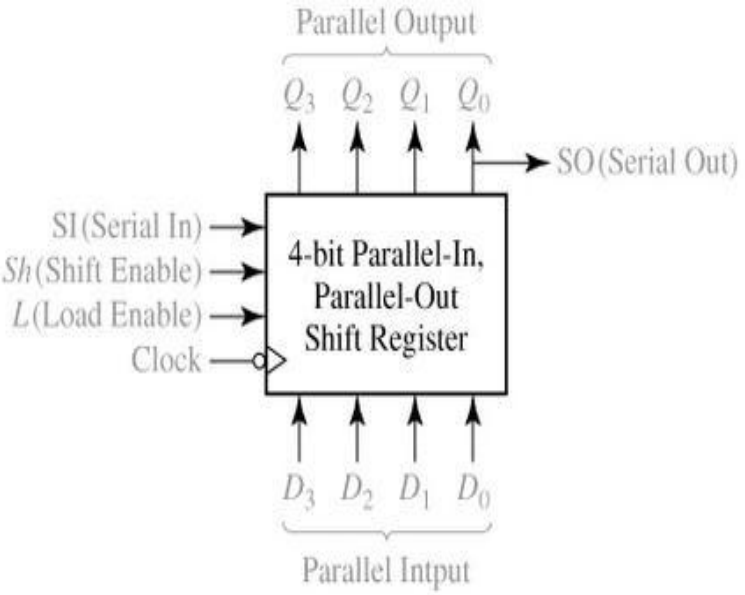
Shift Register Timing

- The output loaded into each FF is the value before the rising clock edge
- If initial value = 0101, and serial input = 1101, then 0101 -> 1010 -> 1101 ..

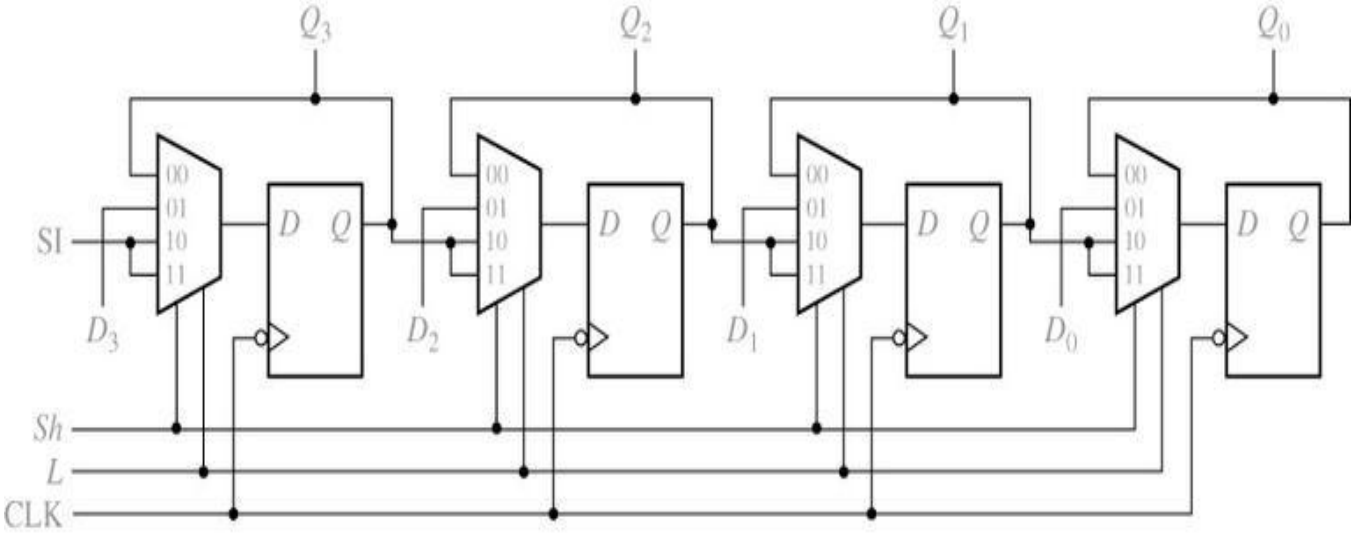


Parallel-in Parallel-out Right Shift Register

- All 4 bits can be loaded or read out at the same time.



(a) Block diagram



(b) Implementation using flip-flops and MUXes

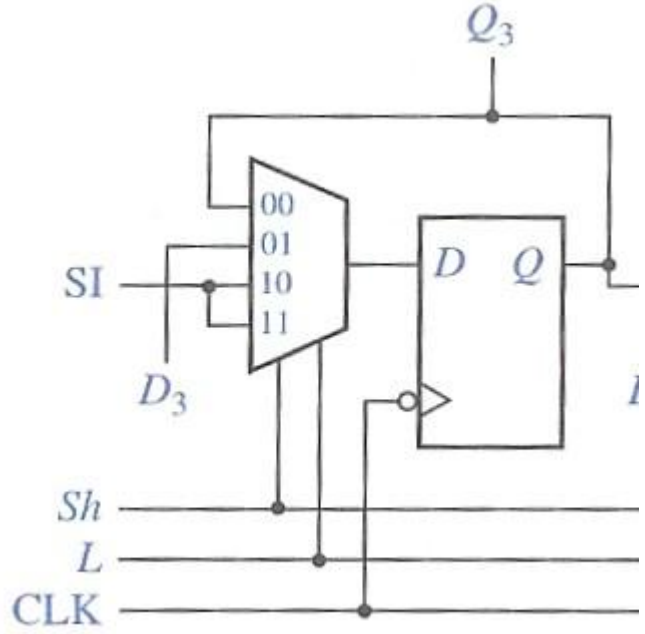
Parallel-in Parallel-out Right Shift Register

- Next state table
- Next state equation

TABLE 12-1 Shift Register Operation

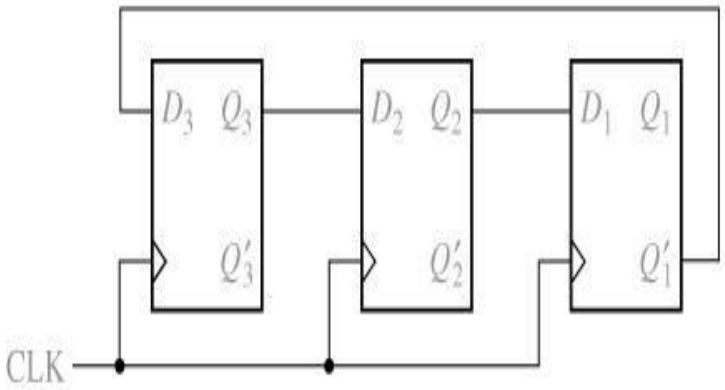
Inputs		Next State				Action
Sh (Shift)	*L* (Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+	
0	0	Q_3	Q_2	Q_1	Q_0	no change
0	1	D_3	D_2	D_1	D_0	load
1	X	SI	Q_3	Q_2	Q_1	right shift

$Q_3^+ = sh'L'Q_3 + sh'L.D_3 + sh.SI$

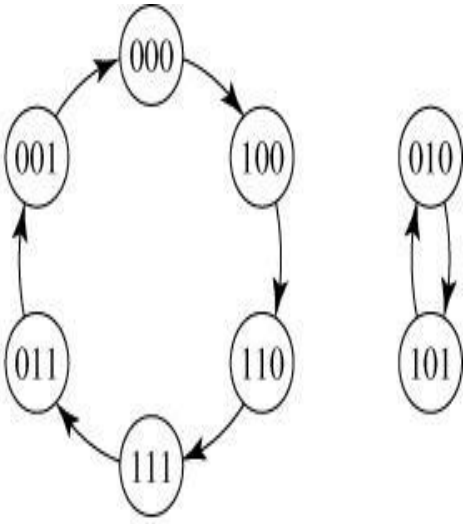


Shift Register with Feedback

- Initial state = 000
- After the rising clock edge, the state is 100
- If the register is in 010, the next state is 101. Not in the main loop.
- This is a counter.



(a) Flip-flop connections



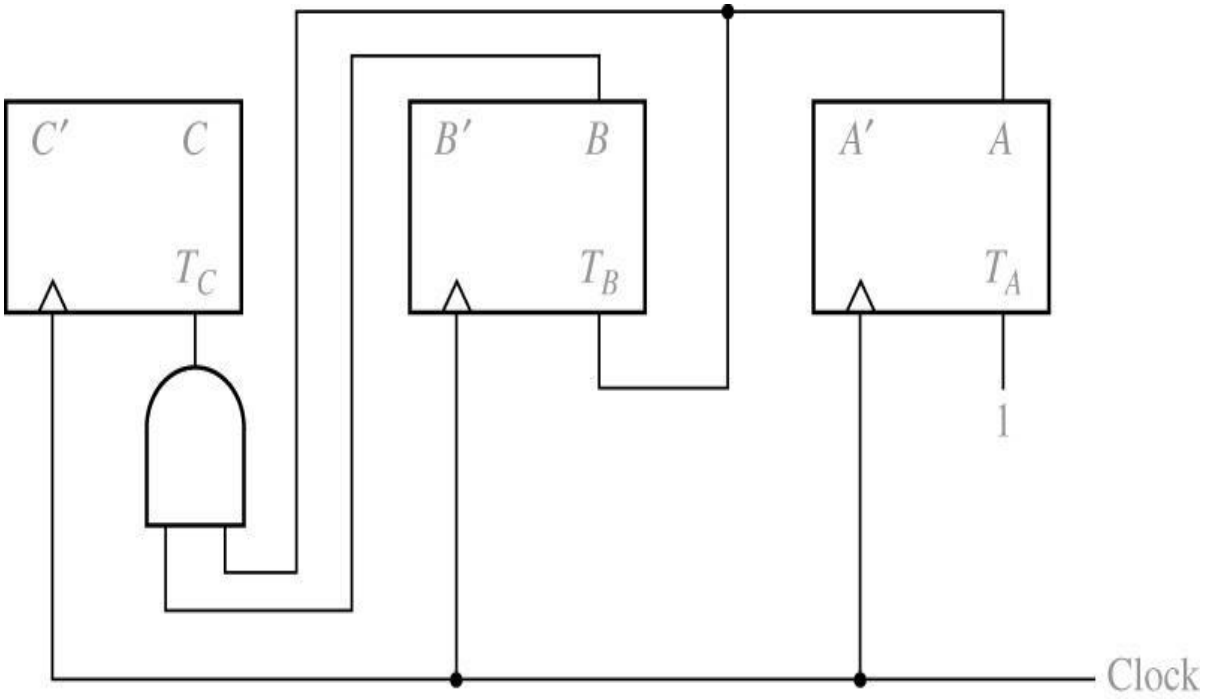
(b) State graph

Synchronous Counter

- Synchronous counter
  - FFs are synchronized by a common input pulse.
- Example: Construct a binary counter using three T FF to count pulses.
  - 000, 001, 010, 011, 100, 101, 110, 111, 000 …..

Synchronous Binary Counter

- 000, 001, 010, 011, 100, 101, 110, 111, 000 ….. T_C, T_B, T_A
  - A changes whenever a rising clock edge occurs.
  - B changes when A = 1.
  - C changes when B = 1 and A = 1 and a rising clock edge occurs.



Synchronous Binary Counter

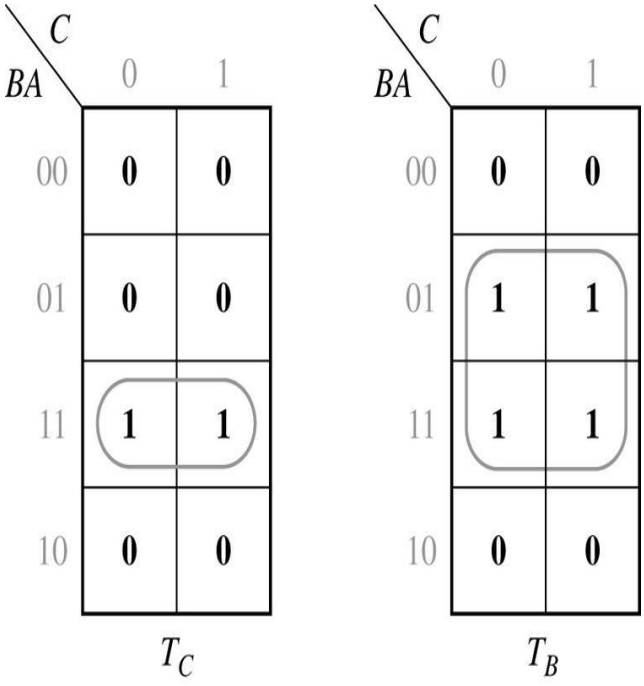
- Using a state table to design the counter. Whenever A and A^+ differs, FF A must change state, thus $T_A = 1$
  - If B and B^+ differs, $T_B = 1$, etc.
  - If CBA = 011, $C^+B^+A^+ = 100$, then $T_C T_B T_A = 111$. (T_C = f (A,B,C), etc)
  - That is, if T FF changes state ($Q^+ \neq Q$), T must be 1.

TABLE 12-2 State Table for Binary Counter

Present State			Next State			Flip-Flop Inputs		
C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Synchronous Binary Counter (cont.)

- T_C, T_B, T_A (flip flop inputs) as function of A,B,C
  - Because we have A, B, C as they are, T_A, T_B, T_C are the results of (A,B,C) in order to get what we want in A^+, B^+, C^+). Therefore T_A, T_B, T_C are functions of A, B, C.
  - Consider T_A, T_B, T_C as multiple output
  - From K-map :T_C= BA, T_B = A, T_A = 1



000 to 111 counter

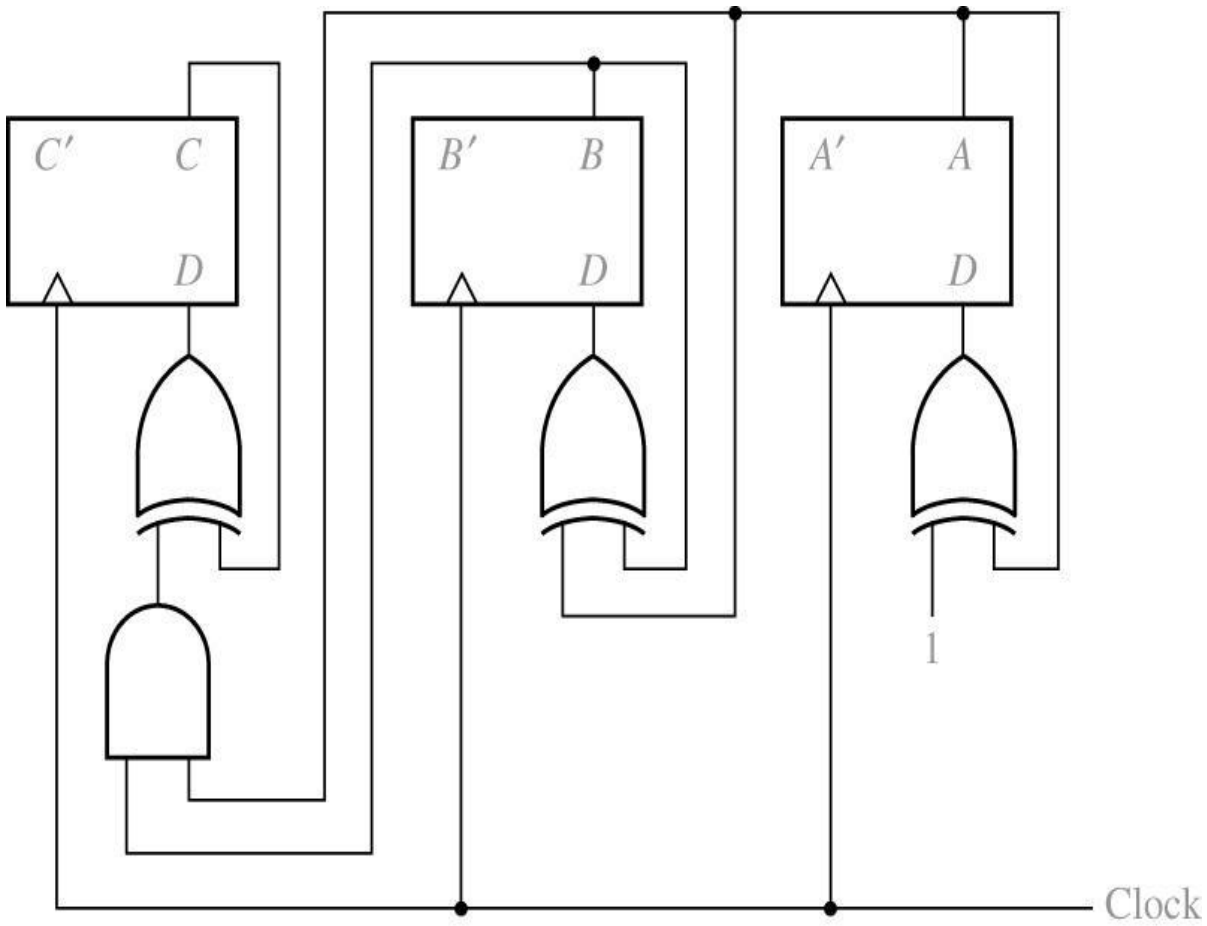
- Use D FFs
- $D_A = A^+ = A'$
- $D_B = B^+ = B \text{ xor } A$
- $D_C = C^+ = C \text{ xor } BA$
- B changes state when A=1
- C changes state when AB = 1

TABLE 12-2 State Table for Binary Counter

Present State			Next State			Flip-Flop Inputs		
C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

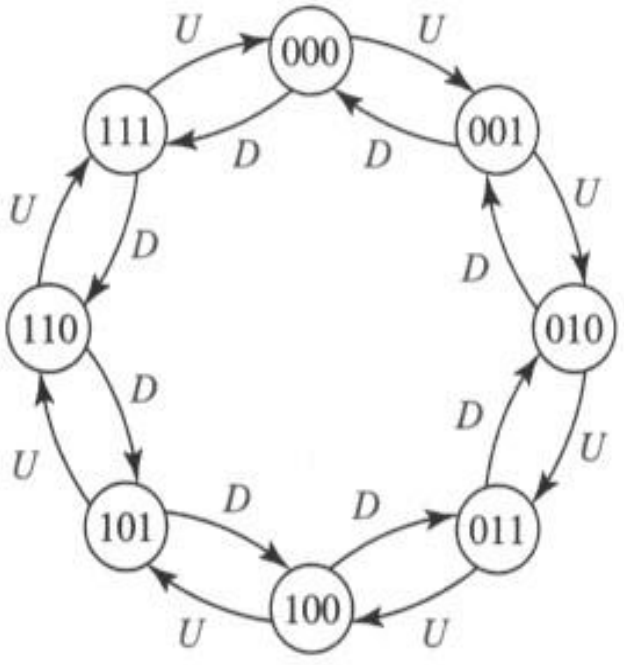
Binary Counter with D FFs

- $D_A = A^+ = A' = 1$ xor A
- $D_B = B^+ = B$ xor A
- $D_C = C^+ = C$ xor BA
  - C changes state when AB = 1



Up-Down Counter

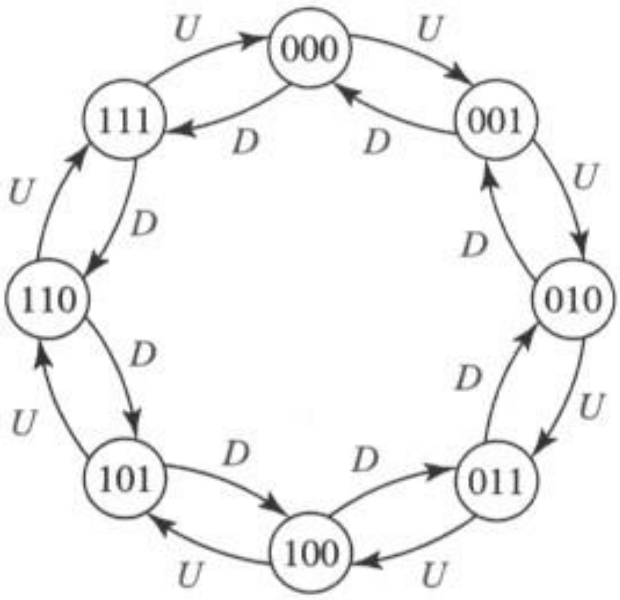
- State transition
- Use D FFs
- U and D can not both equal 1 at the same time.
- For D=1, B changes state when A = 0
- For U=1, B changes state when A = 1



CBA	$C^+B^+A^+$	
	U	D
000	001	111
001	010	000
010	011	001
011	100	010
100	101	011
101	110	100
110	111	101
111	000	110

Up-Down Counter

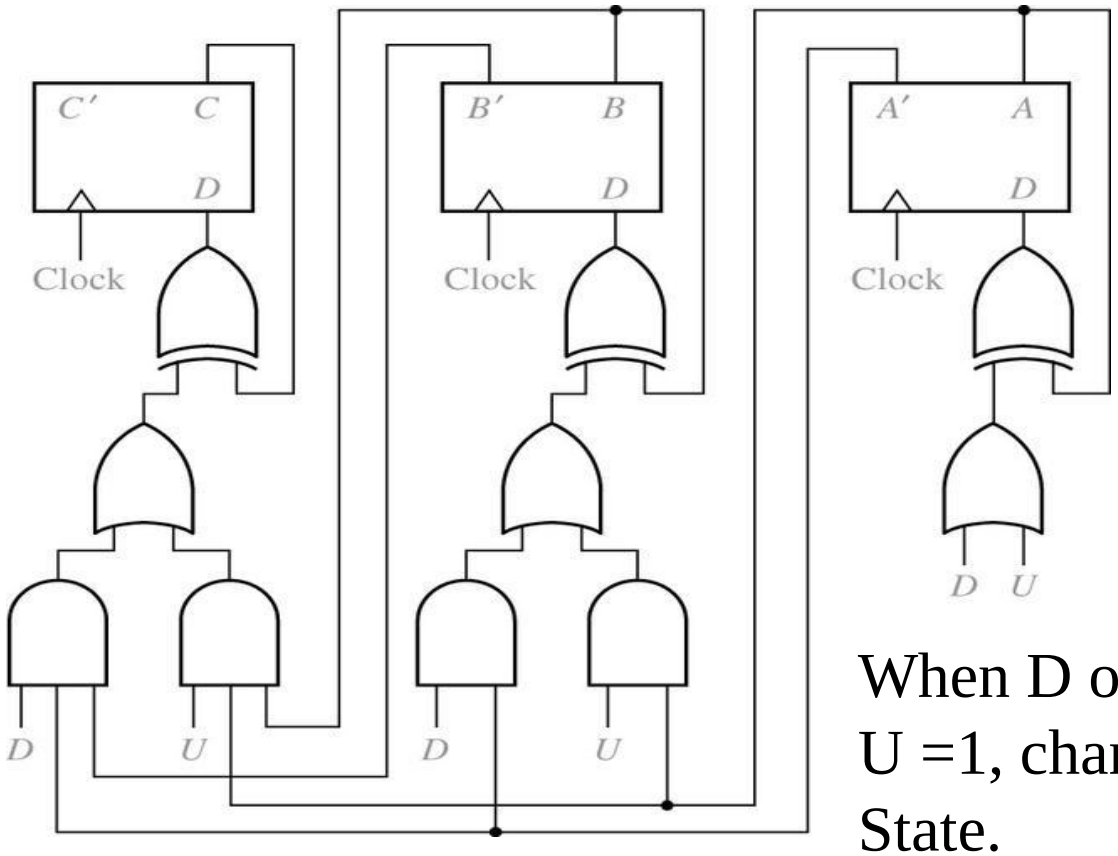
- Construct a truth table
- A^+ = function (C, B, A, U, D)
  - Then simplify this function for A^+
- Or observe the table:
  - When U = 1 or D = 1, A^+ = A'. When U = 0, D = 0, A remains unchanged. Thus A^+ = A'U + A'D + AU'D'
  - Or A^+ = A'(U +D) + A(U+D)'
  - U = 1 or D = 1, A^+ = A'. U = 0 and D = 0, A^+ = A.



CBA	$C^+B^+A^+$	
	U	D
000	001	111
001	010	000
010	011	001
011	100	010
100	101	011
101	110	100
110	111	101
111	000	110

Up-Down Counter

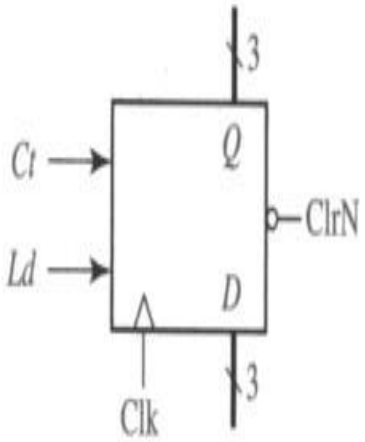
- $D_A = A^+ = A \text{ xor } (U + D)$
- $D_B = B^+ = B \text{ xor } (UA + DA')$
- $D_C = C^+ = C \text{ xor } (UBA + DB'A')$
- Homework: find B^+ and C^+.



When D or U =1, change State.

Loadable Counter

- When Load (Ld=1), data is loaded into the counter on the rising clock edge.
- When Ct = 1, the counter is incremented on the rising clock edge.



(a)

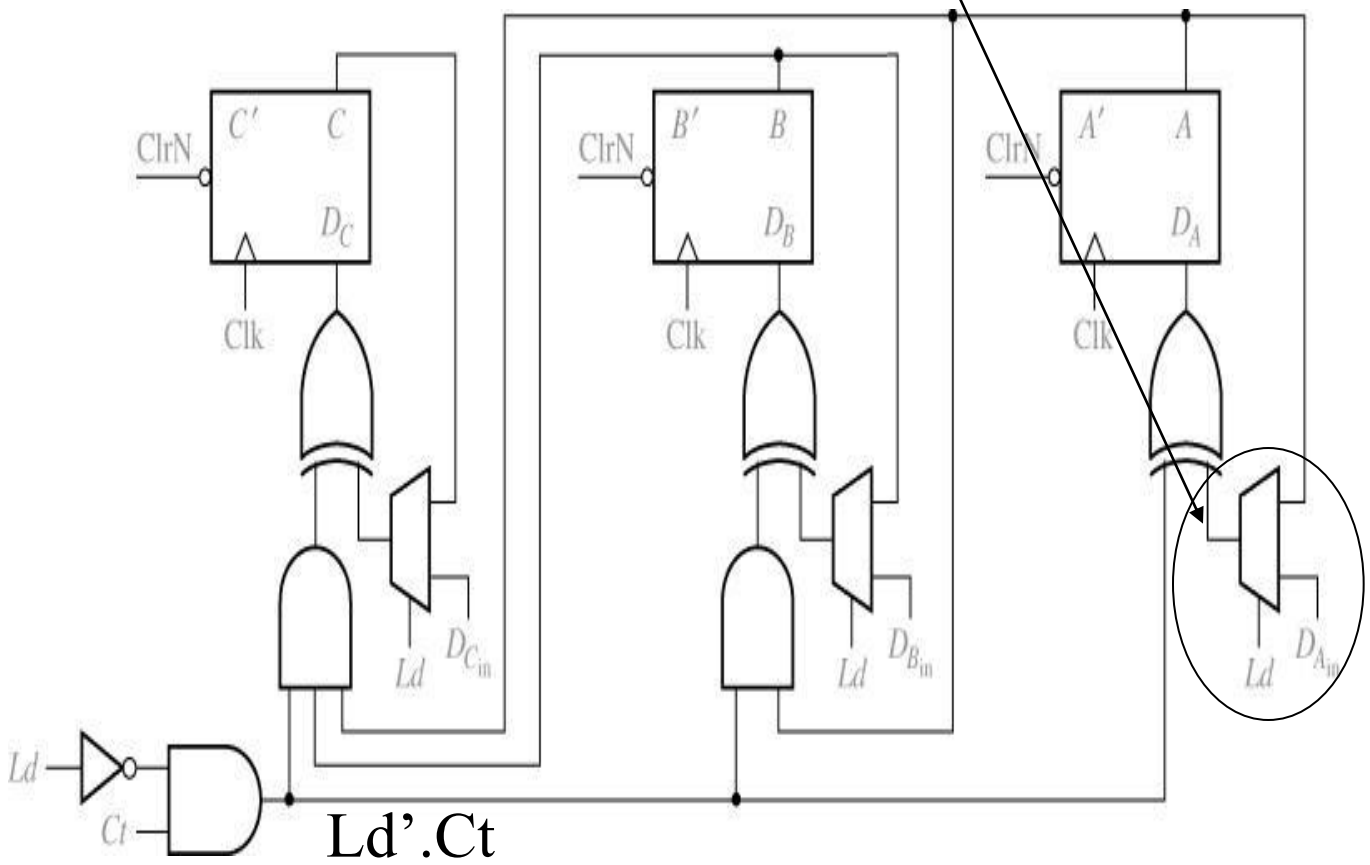
ClrN	Ld	Ct	C^+	B^+	A^+	
0	X	X	0	0	0	
1	1	X	D_C	D_B	D_A	(load)
1	0	0	C	B	A	(no change)
1	0	1	Present state + 1			

(b)

Ld’Ct changes state.

Loadable Counter

- When Ld=1, Dins are selected and clocked in since the output of each AND gate is 0.
- When Ct =1, Ld=0, it becomes equivalent to the UP counter.
- $D_A = A^+ = (Ld'.A + Ld.D_{Ain})$ xor $Ld'.Ct$
- $D_B = B^+ = (Ld'.B + Ld.D_{Bin})$ xor $Ld'.CtA$
- $D_C = C^+ = (Ld'.C + Ld.D_{Cin})$ xor $Ld'.CtAB$



Counters with Non-Binary Sequence

- State graph
  - The next state of 000 is 100.
  - 001 has no next state from the graph.

FIGURE 12-21
State Graph for Counter

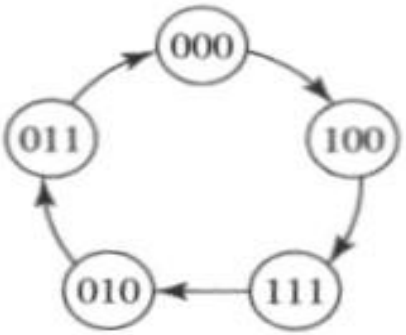
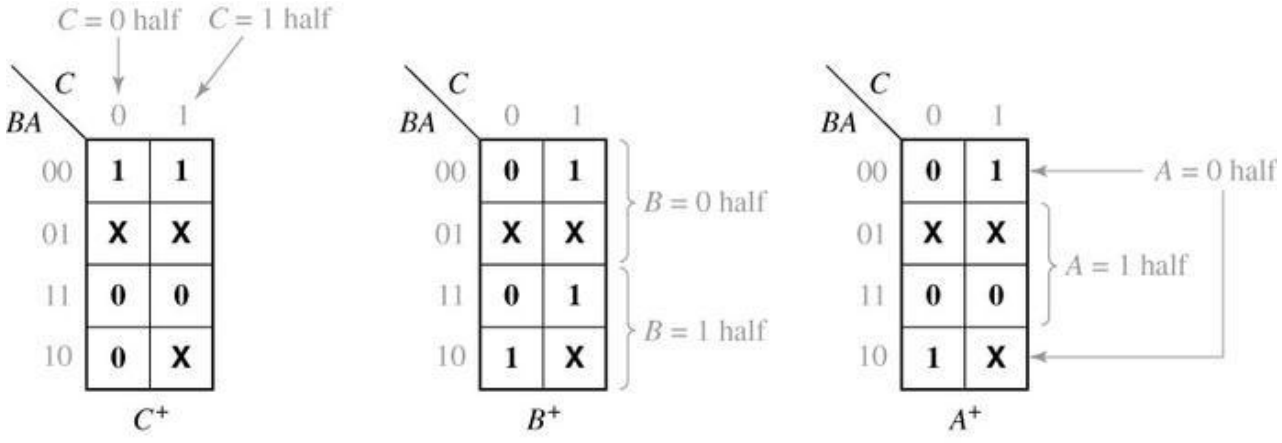


TABLE 12-3
State Table for Figure 12.21

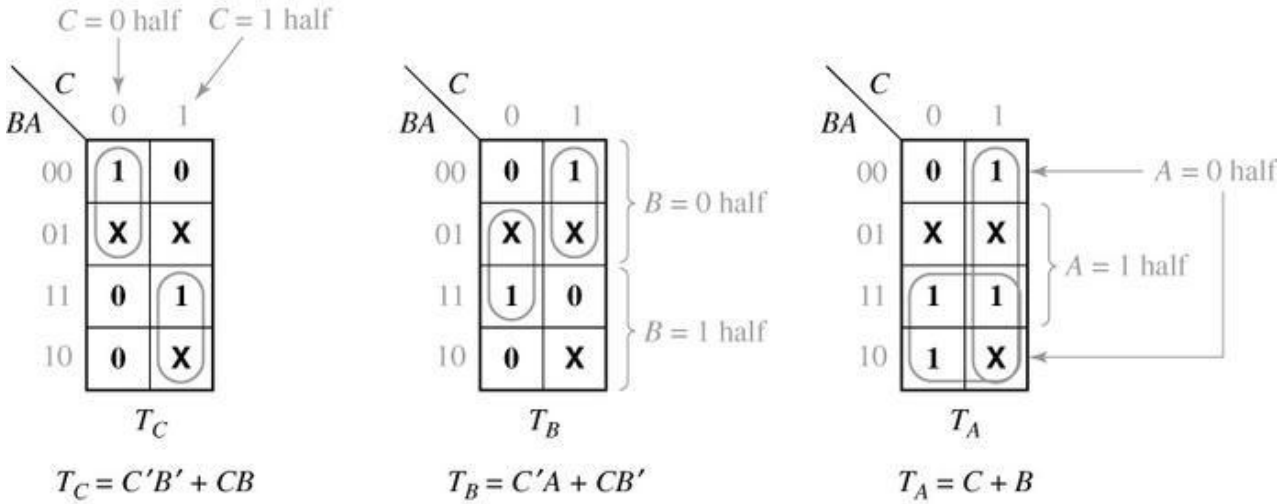
C	B	A	C^+	B^+	A^+
0	0	0	1	0	0
0	0	1	–	–	–
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	–	–	–
1	1	0	–	–	–
1	1	1	0	1	0

Counters with Non-Binary Sequence (cont.)

- Alternatively, we use
  - Next-state maps (Q^+) to find T_C, T_B, T_A.
  - In C^+ map (000), when C = 0, C^+ =1, so T_C = 1. T = 1 whenever $Q^+ \neq Q$.



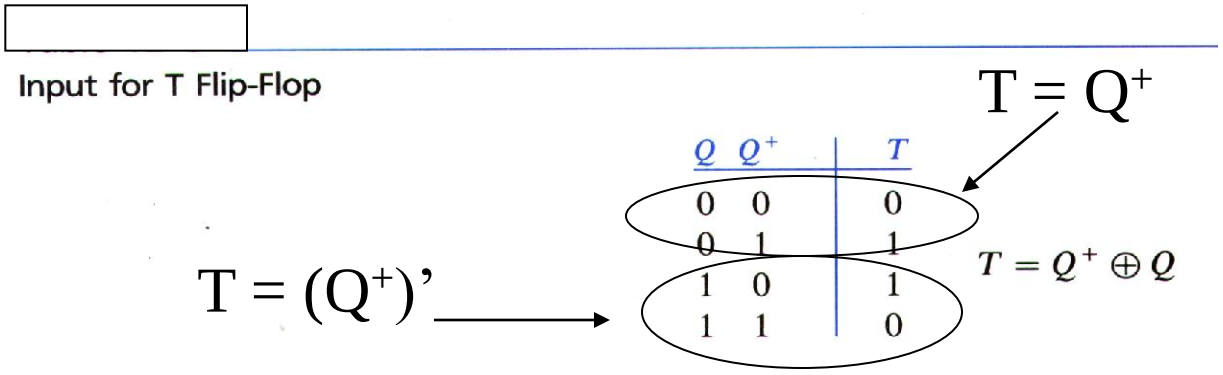
(a) Next-state maps for Table 12-3



(b) Derivation of T inputs

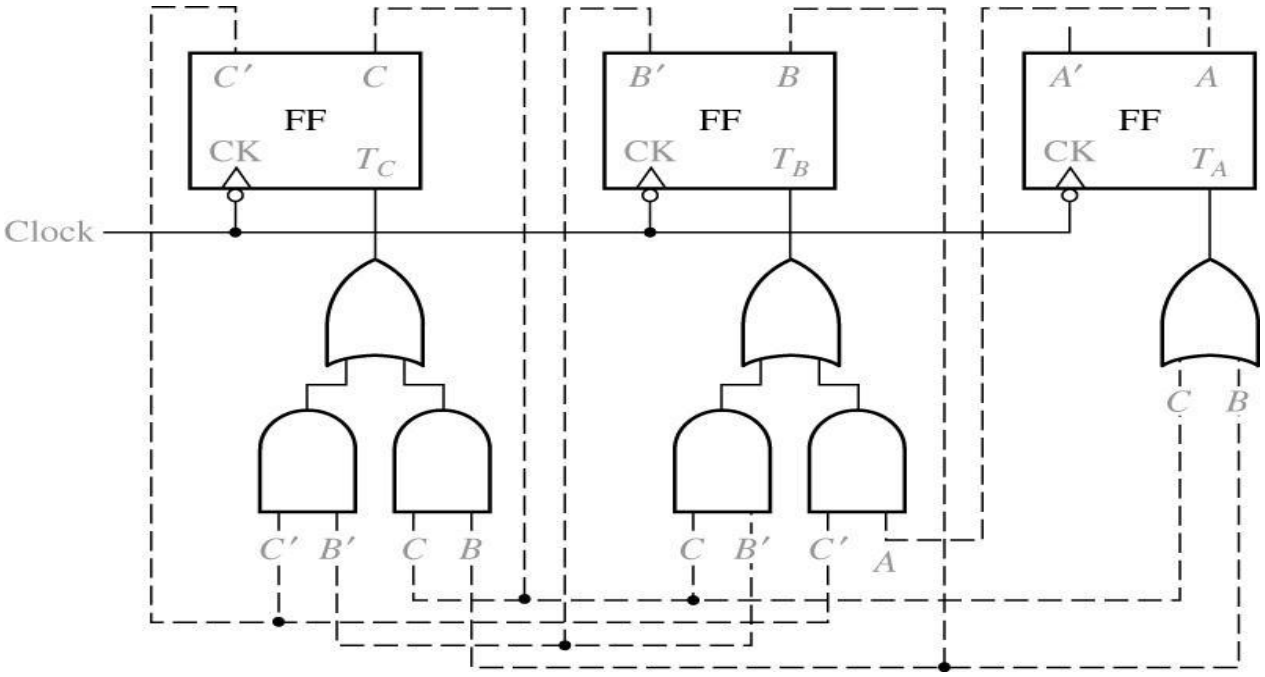
Counters with Non-Binary Sequence (cont.)

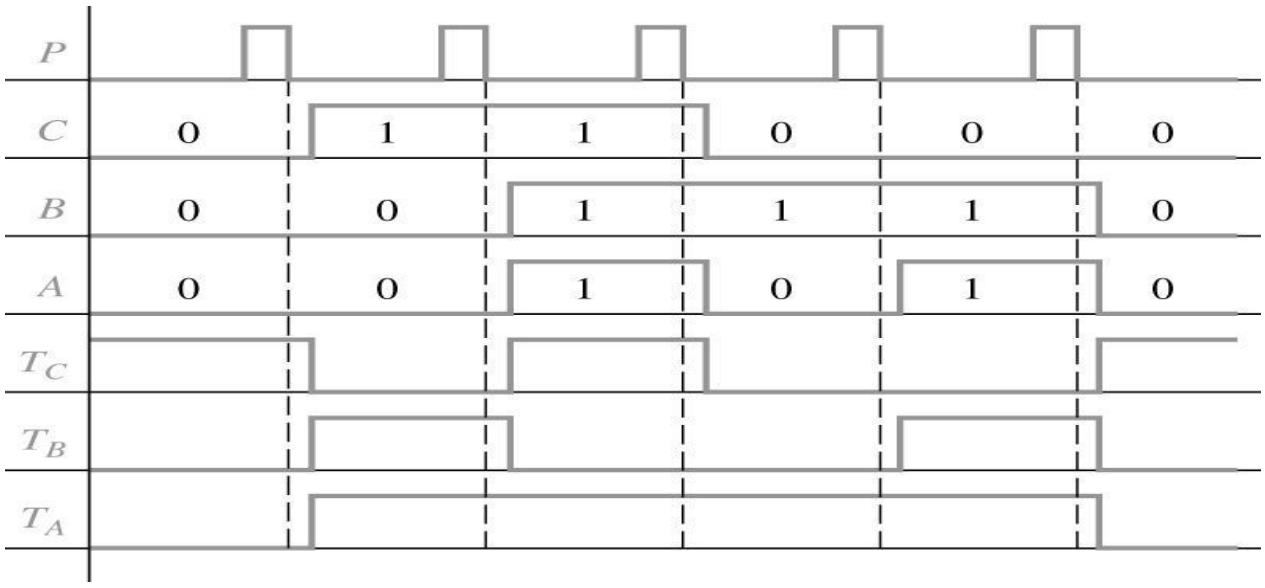
- Using clocked T FF.
  - Next-state maps
  - T = 1 whenever $Q^+ \neq Q$, which means $T = Q^+ \oplus Q$.



Counters with Non-Binary Sequence (cont.)

- The resultant circuit and timing diagram.





Counters with Non-Binary Sequence (cont.)

- Summary of procedure
  - Form a state table of Q and Q^+
  - Plot Q^+ map with respect to Q.
  - Plot T map w.r.t Q. T_Q map is formed from the Q^+ map by complementing the Q = 1 entries and leaving Q = 0 entries unchanged.
  - Find input equation of each T using the T_Q map.

Counter Using S-R FF

- S-R Table
  - Problem (the same counter)
  - Get S_A from Table 12-5 (c)

TABLE 12-5
S-R Flip-Flop Inputs

(a)

S	R	Q	Q^+	
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	1	
1	0	1	1	
1	1	0	–	inputs not allowed
1	1	1	–	

(b)

Q	Q^+	S	R
0	0	0	0
		0	1
0	1	1	0
1	0	0	1
1	1	0	0
		1	0

(c)

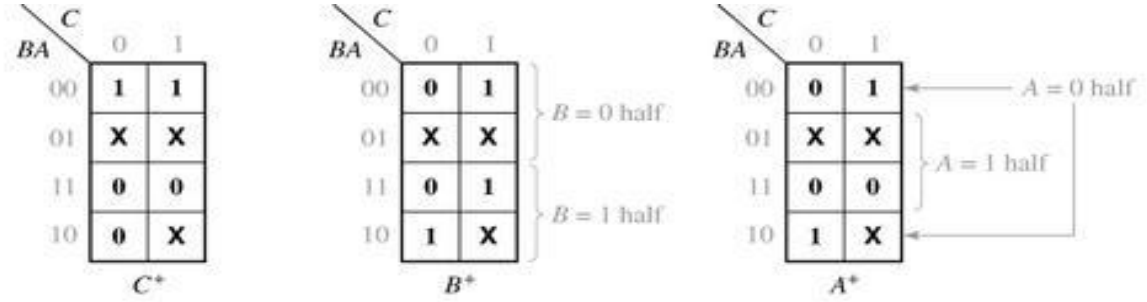
Q	Q^+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

TABLE 12-6

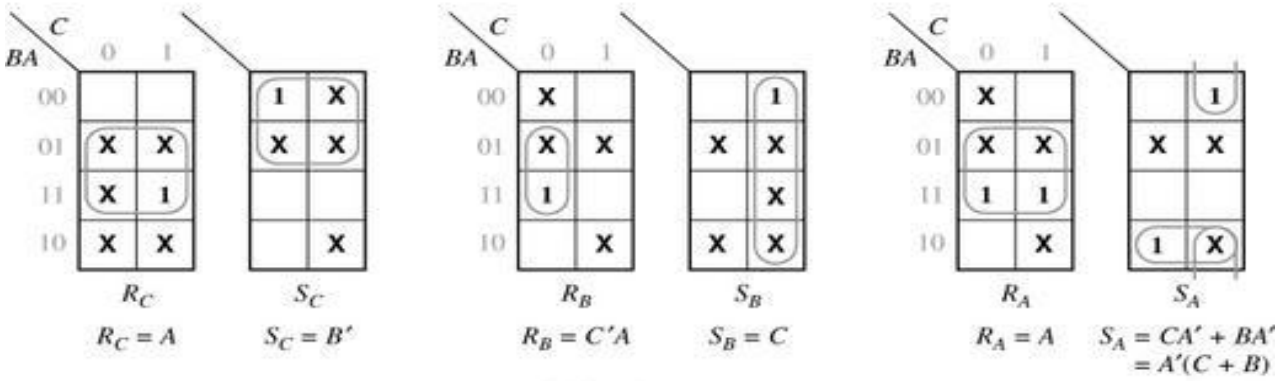
C	B	A	C^+	B^+	A^+	S_C	R_C	S_B	R_B	S_A	R_A
0	0	0	1	0	0	1	0	0	X	0	X
0	0	1	–	–	–	X	X	X	X	X	X
0	1	0	0	1	1	0	X	X	0	1	0
0	1	1	0	0	0	0	X	0	1	0	1
1	0	0	1	1	1	X	0	1	0	1	0
1	0	1	–	–	–	X	X	X	X	X	X
1	1	0	–	–	–	X	X	X	X	X	X
1	1	1	0	1	0	0	1	X	0	0	1

Counter Using S-R FF

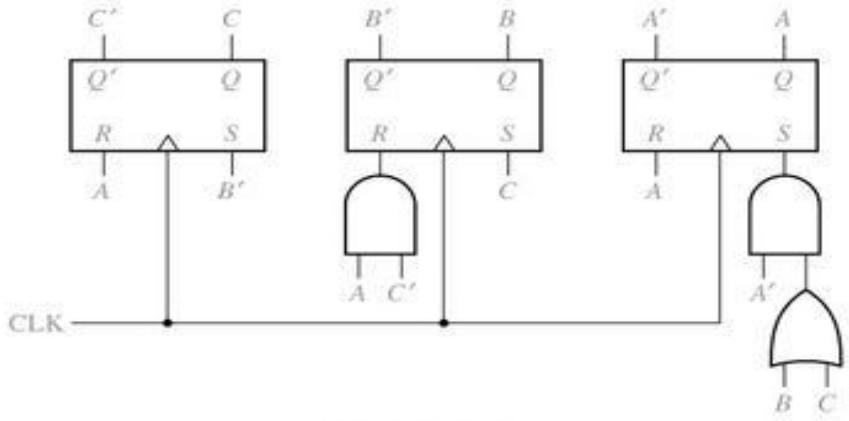
- Get K-map (of S_A, R_A, etc) from state table.
  - Problem (the same counter Fig 12-21)



(a) Next-state maps



(b) S-R flip-flop equations



(c) Logic circuit

Counter Using Clocked J-K FF

- Procedure is similar to S-R FF.

TABLE 12-7
J-K Flip-Flop Inputs

(a)

J	K	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

(b)

Q	Q^+	J	K
0	0	0	0
		0	1
0	1	1	0
		1	1
1	0	0	1
		1	1
1	1	0	0
		1	0

(c)

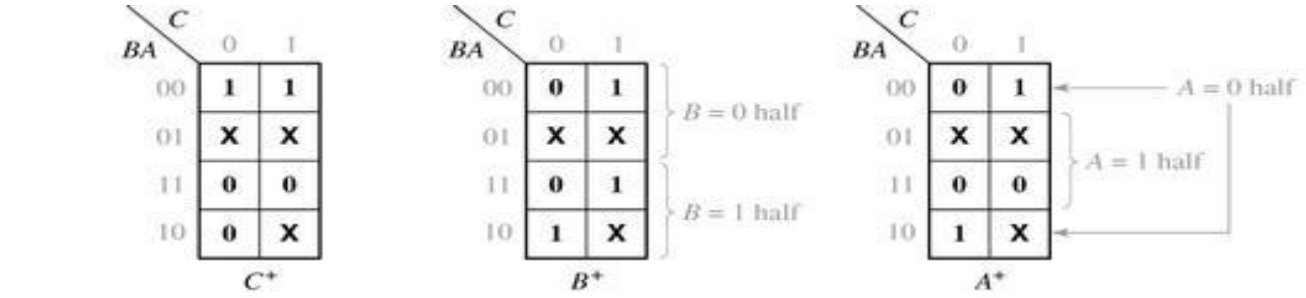
Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

TABLE 12-8

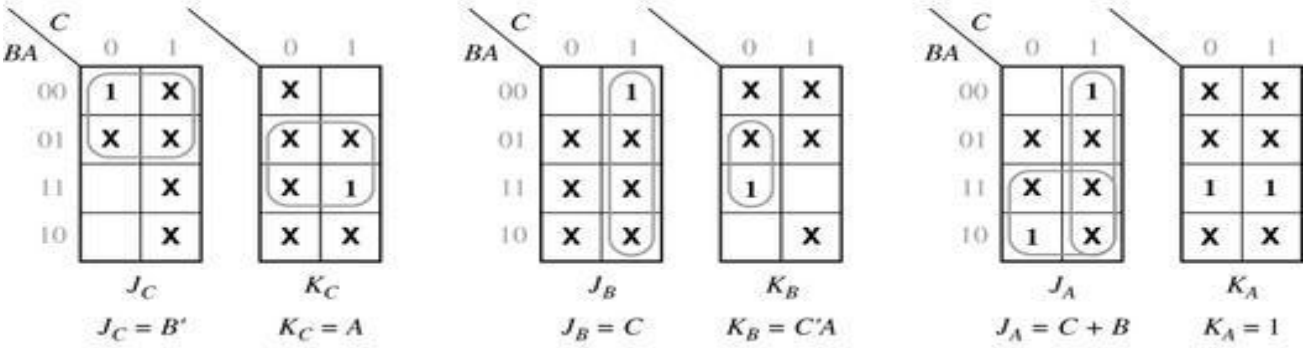
C	B	A	C^+	B^+	A^+	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	1	0	0	1	X	0	X	0	X
0	0	1	–	–	–	X	X	X	X	X	X
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	0	0	X	X	1	X	1
1	0	0	1	1	1	X	0	1	X	1	X
1	0	1	–	–	–	X	X	X	X	X	X
1	1	0	–	–	–	X	X	X	X	X	X
1	1	1	0	1	0	X	1	X	0	X	1

Counter Using Clocked J-K FF (cont.)

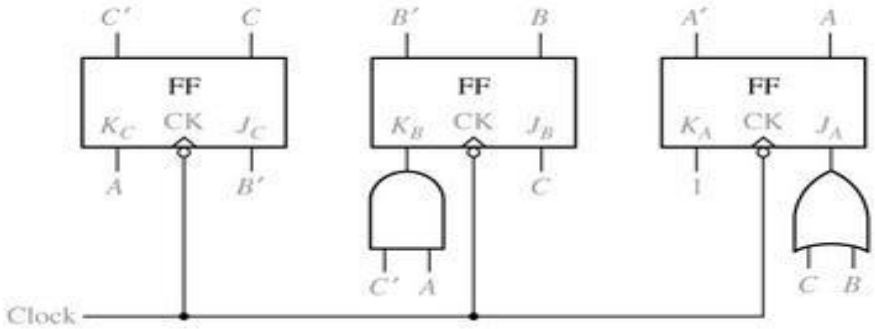
- Find J-K input equations



(a) Next-state maps



(b) J-K flip-flop input equations



(c) Logic circuit (omitting the feedback lines)