Chap 12:

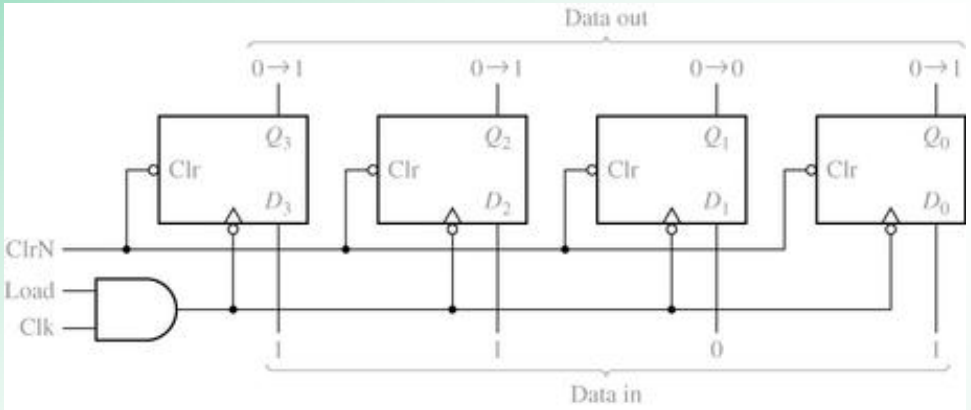
Registers and Counters

Objectives

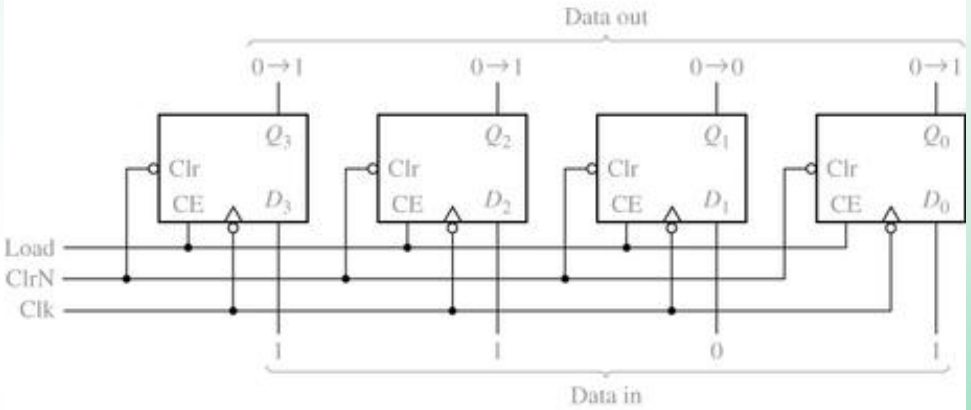
- Registers
  - Operation
  - Application to shift registers
- Counters
  - How to build and how it works
  - Counter application and how to derive the FF's input equations

12-1 Registers and Register Transfers

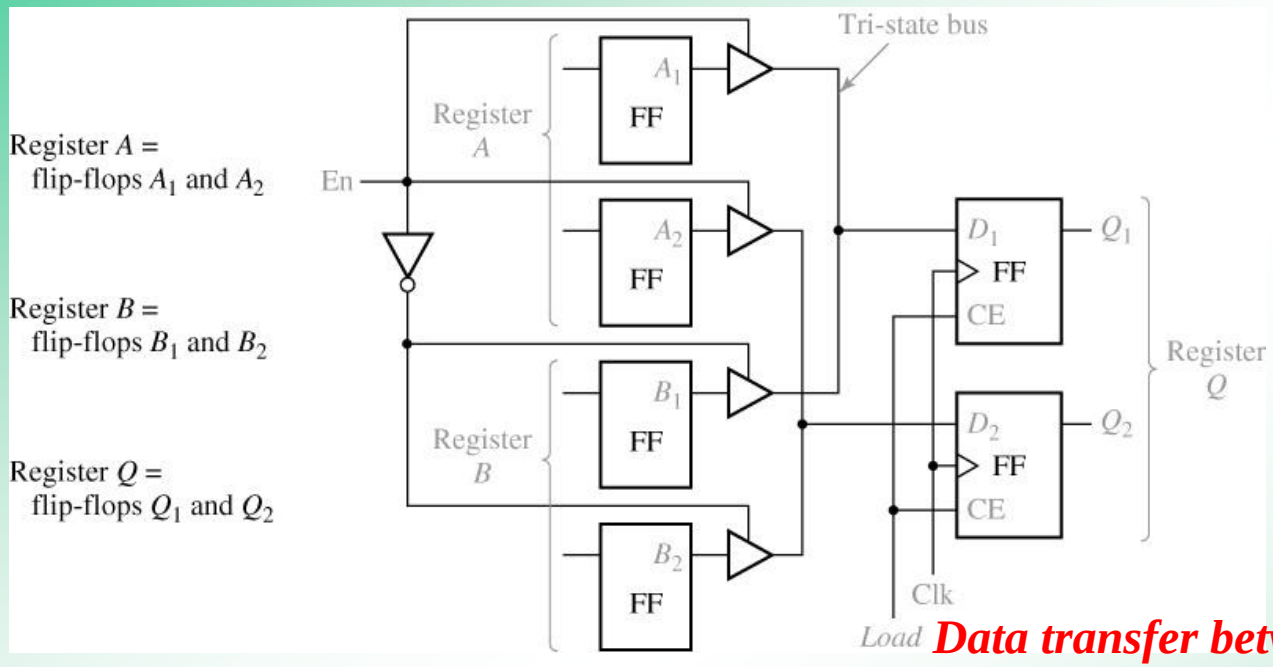
Registers: a group of FF with a common clock input, used to **store and shift** binary data



(a) Using gated clock

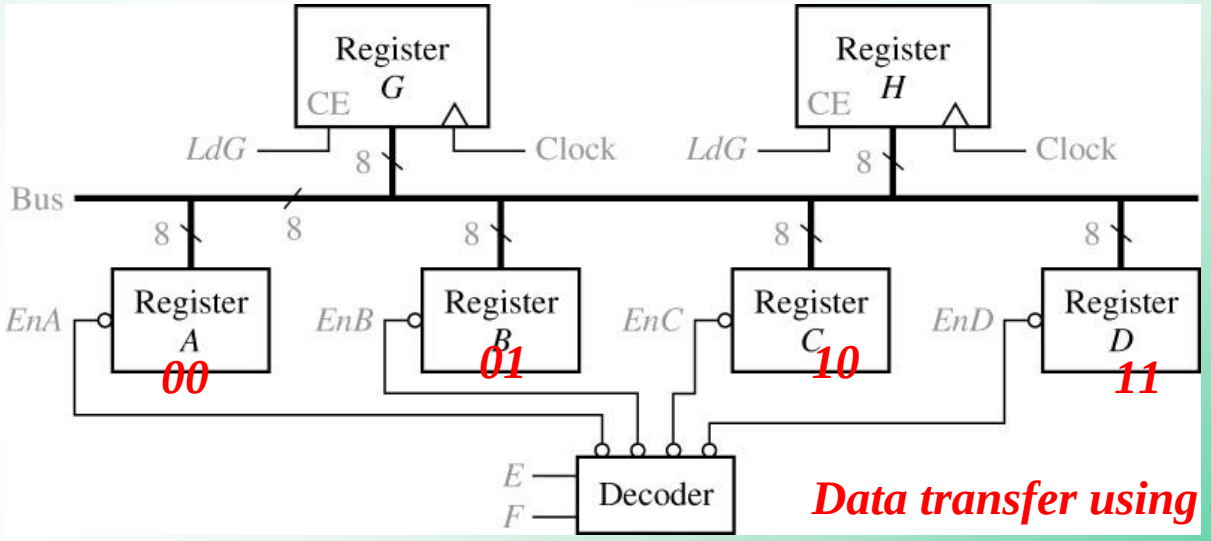


(b) With clock enable



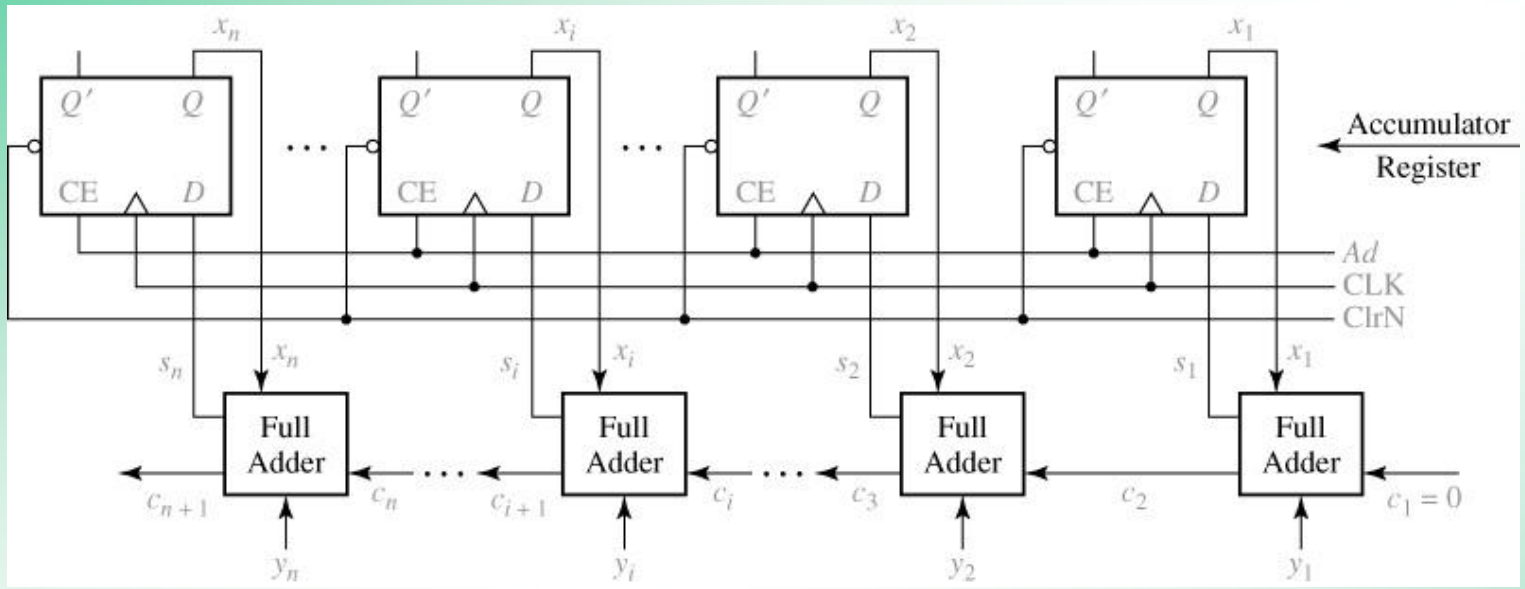
Data transfer between registers

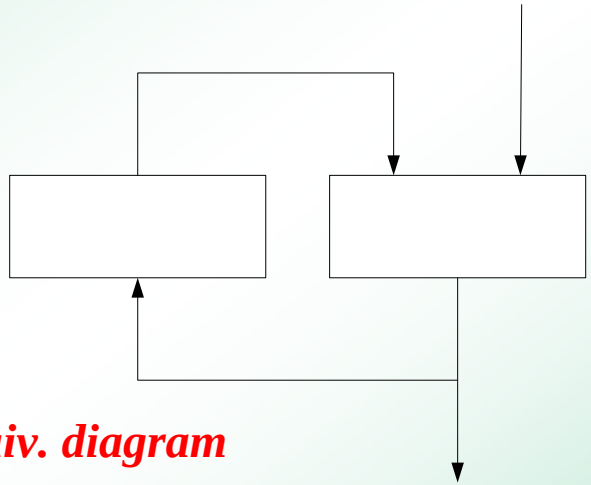
Equiv. to 2:1 MUX



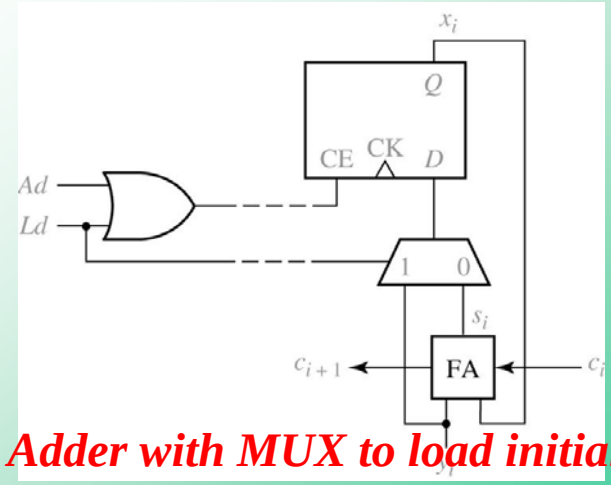
Data transfer using a tri-state bus

Parallel Adder with Accumulator





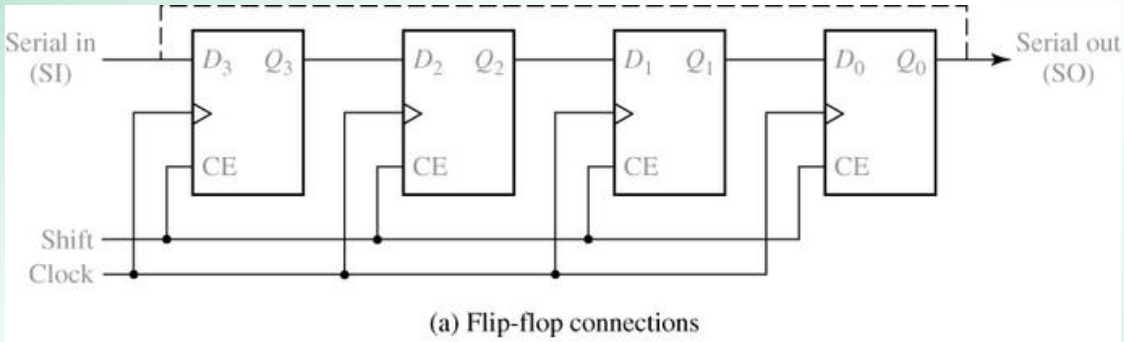
Equiv. diagram



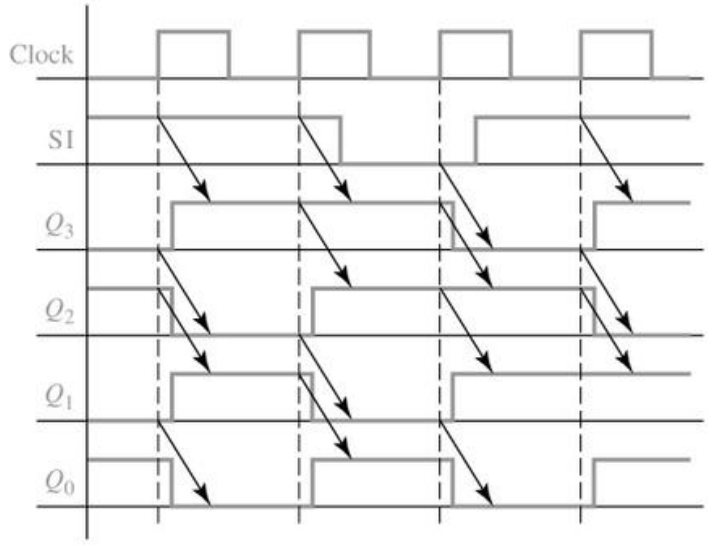
Adder with MUX to load initial value

12-2 Shift Registers:

A group of F/Fs in which a binary number can be stored. This number can be shifted left or right when a shift signal is applied.

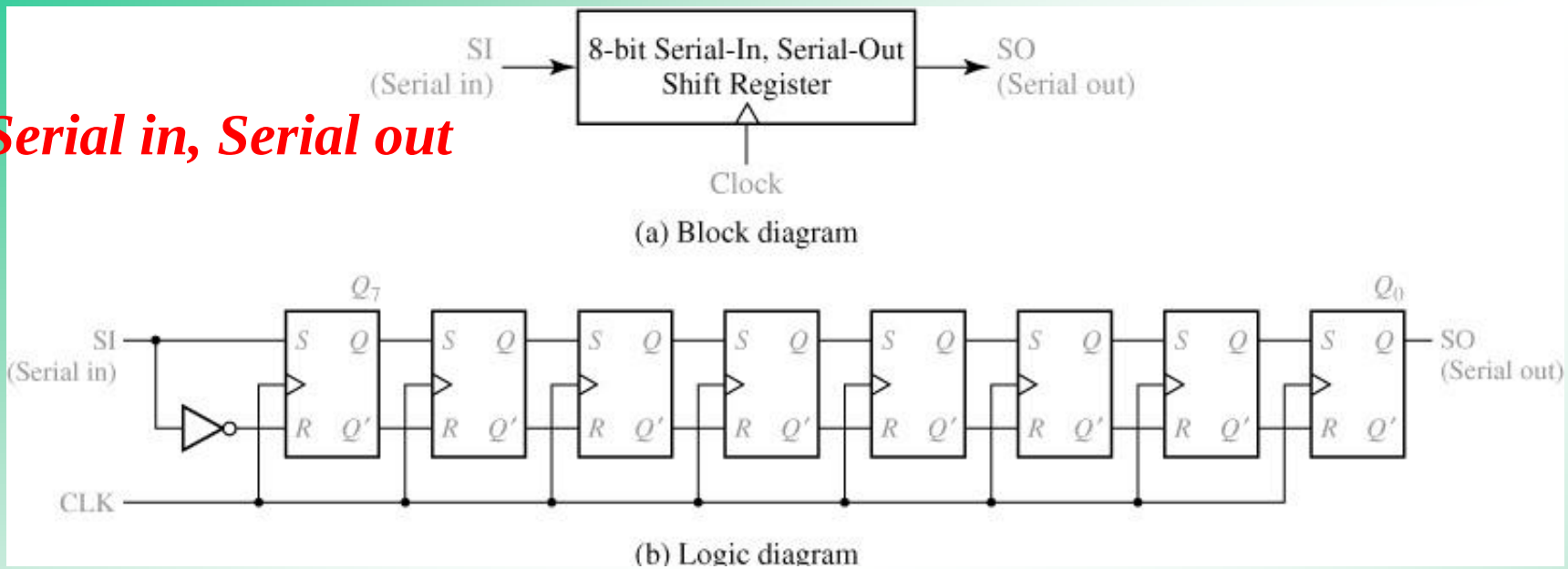


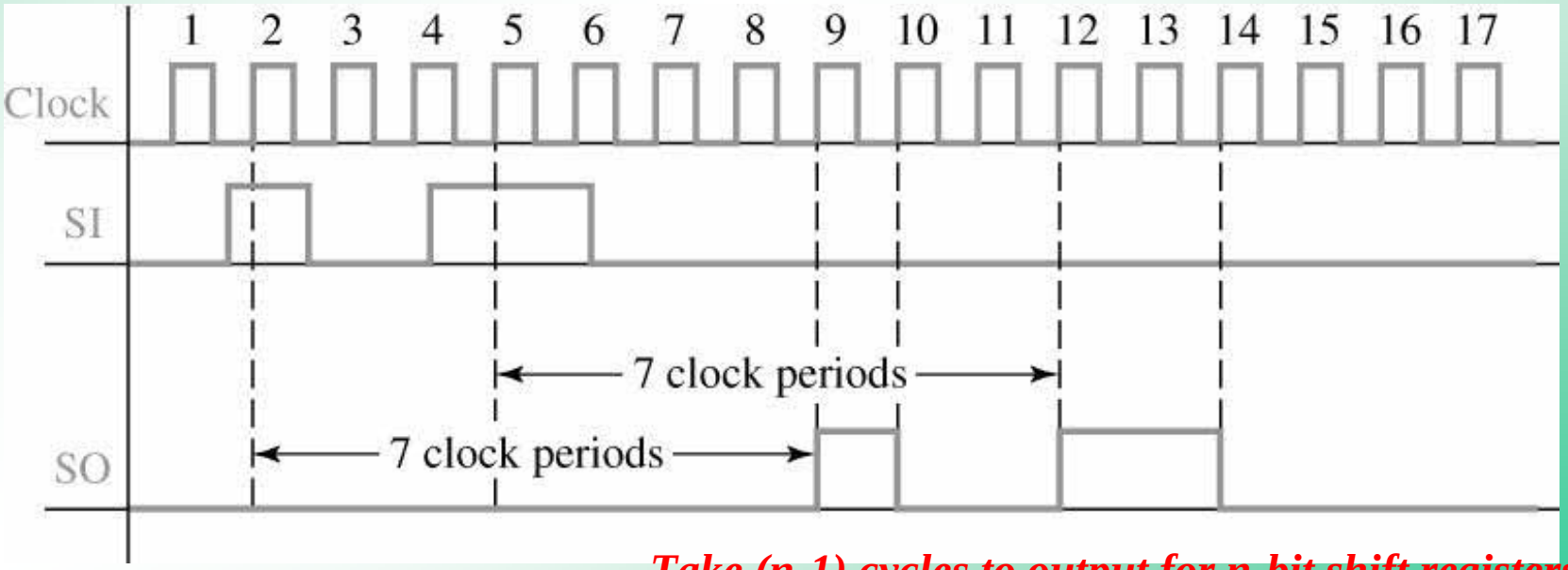
(a) Flip-flop connections



(b) Timing diagram

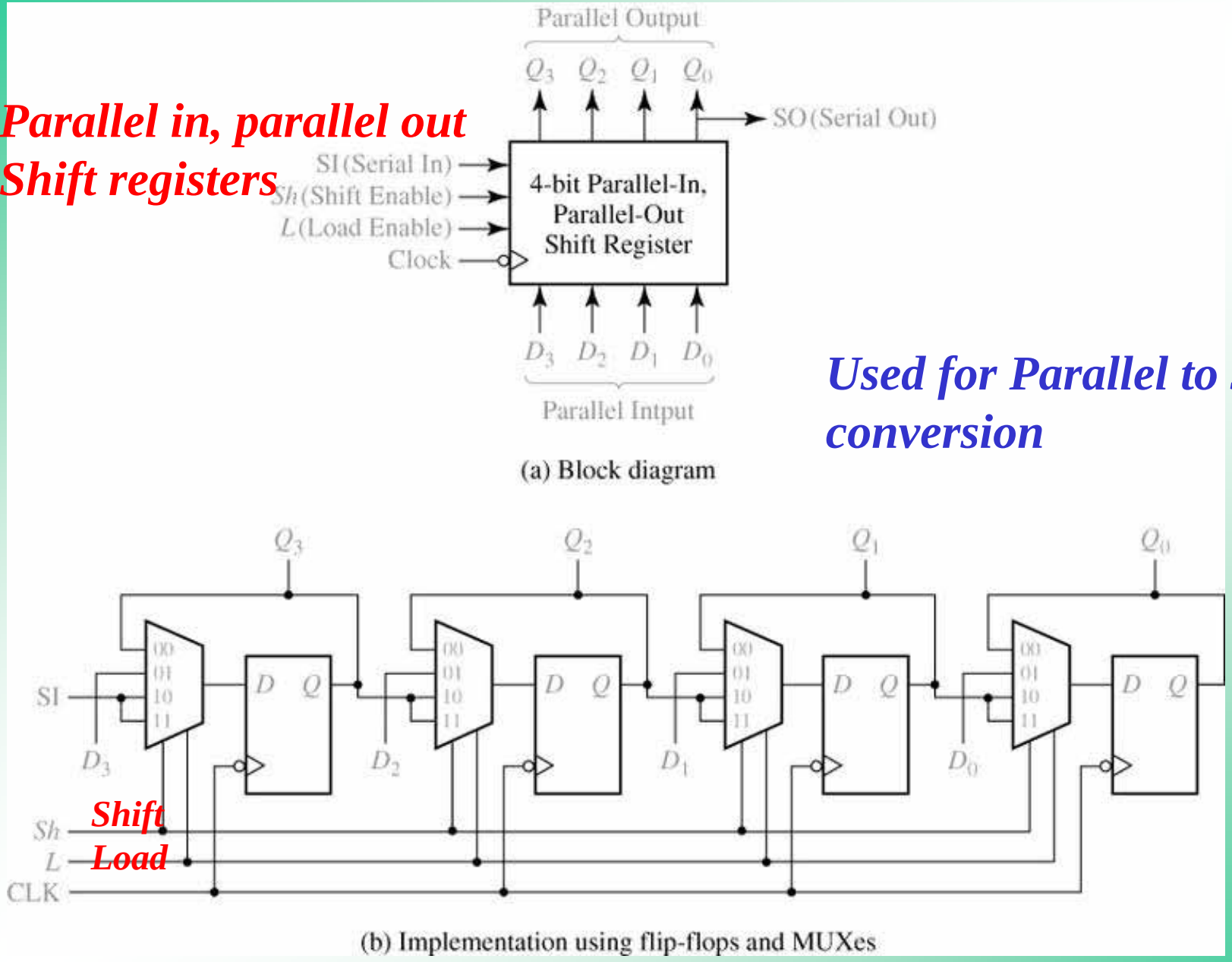
Serial in, Serial out



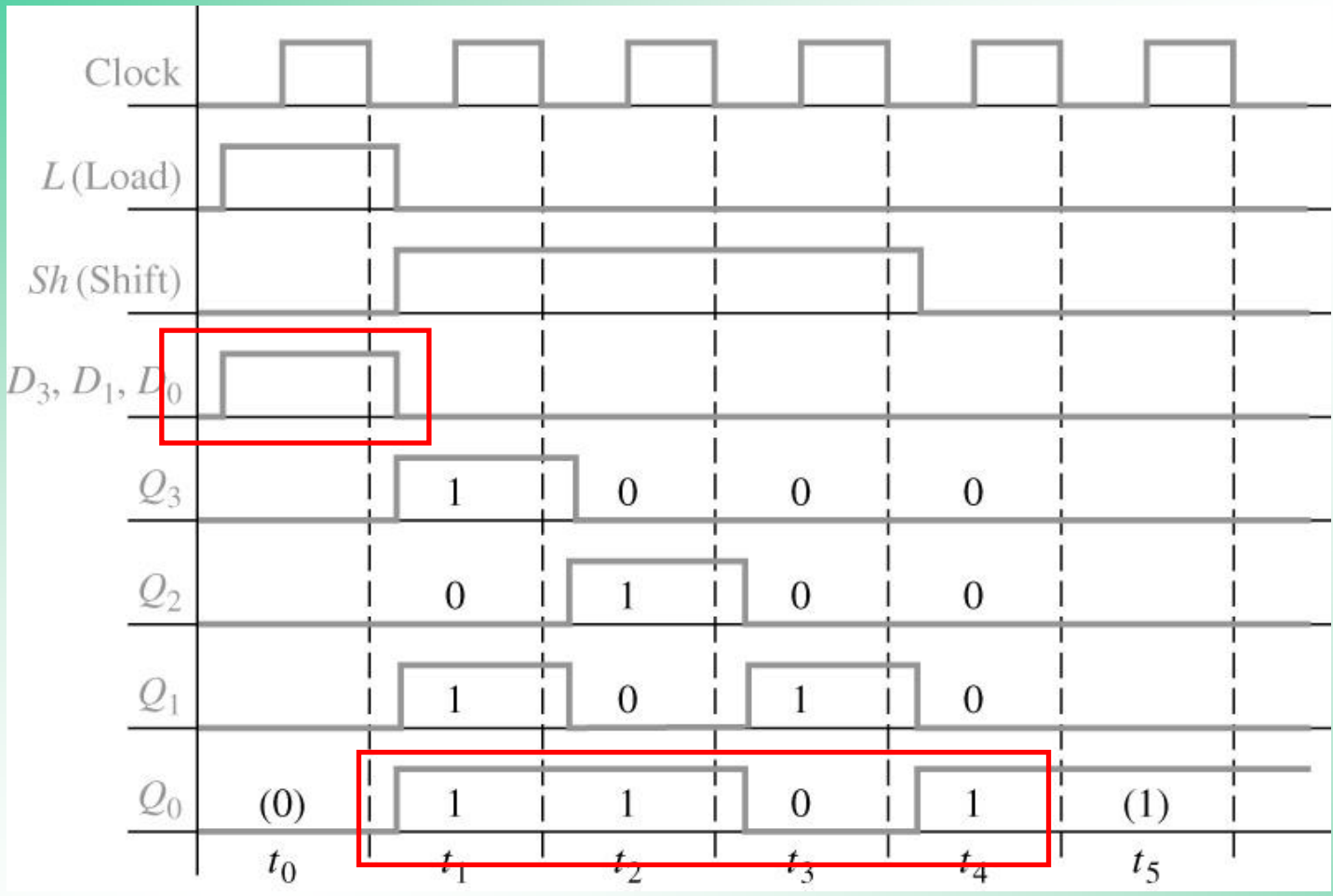


Take (n-1) cycles to output for n-bit shift registers

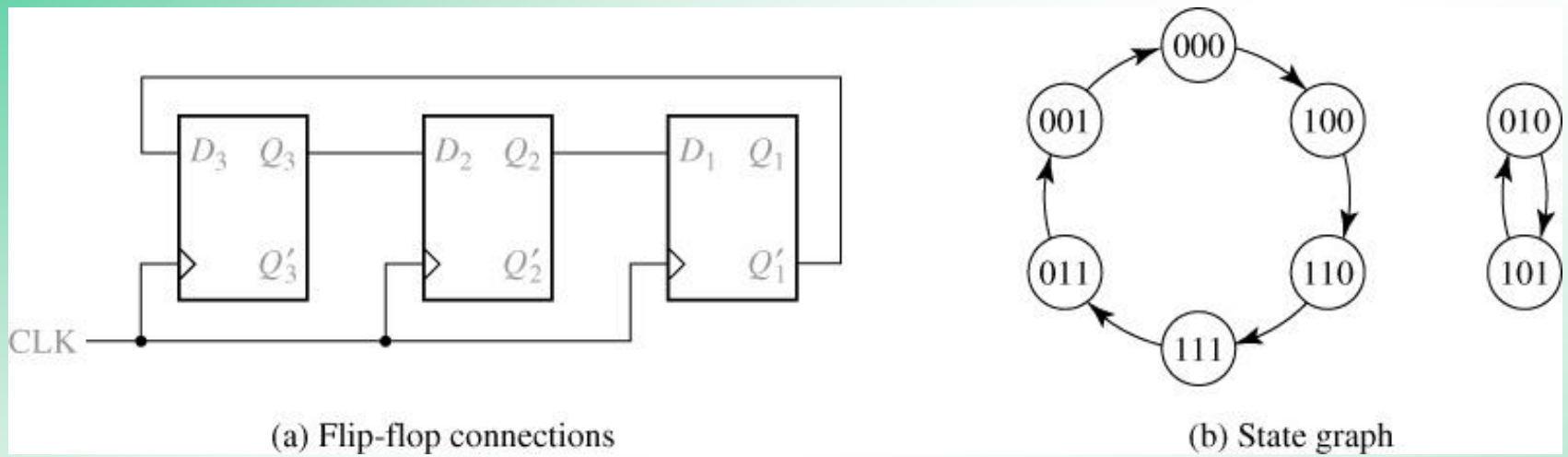
Parallel in, parallel out Shift registers



Used for Parallel to serial conversion



Used for Parallel to serial conversion
e.g. convert parallel 1011 to serial 1101



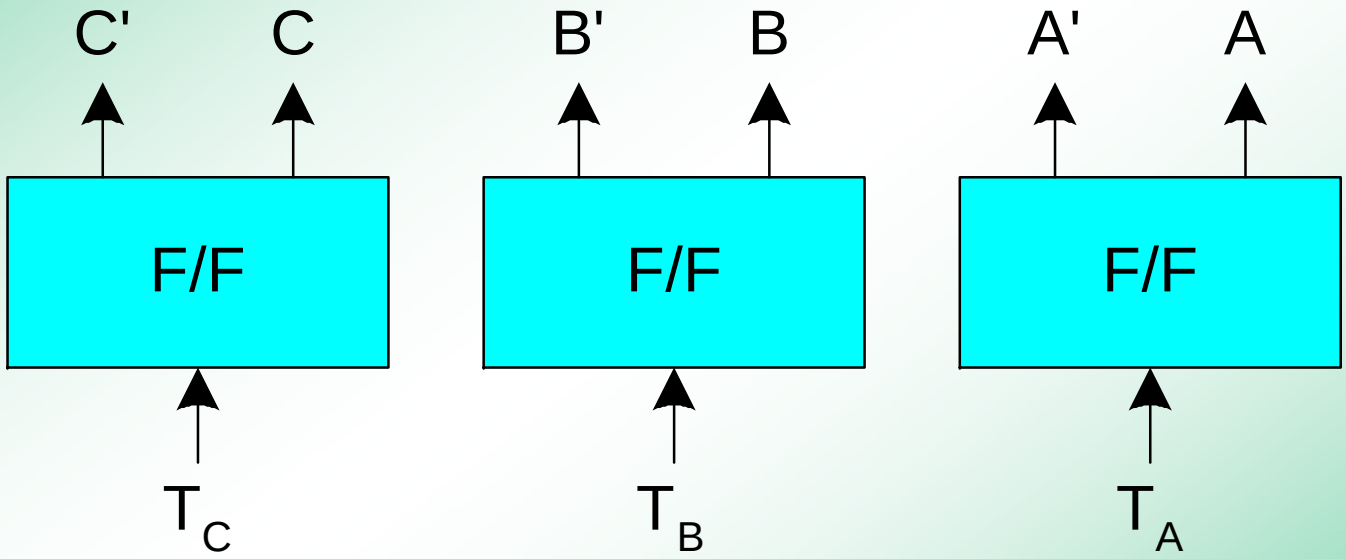
(a) Flip-flop connections

(b) State graph

Counter: circuit that cycles through a fixed sequence of states

Johnson counter: shift register with inverted feedback

12-3 Design of a Binary Counter

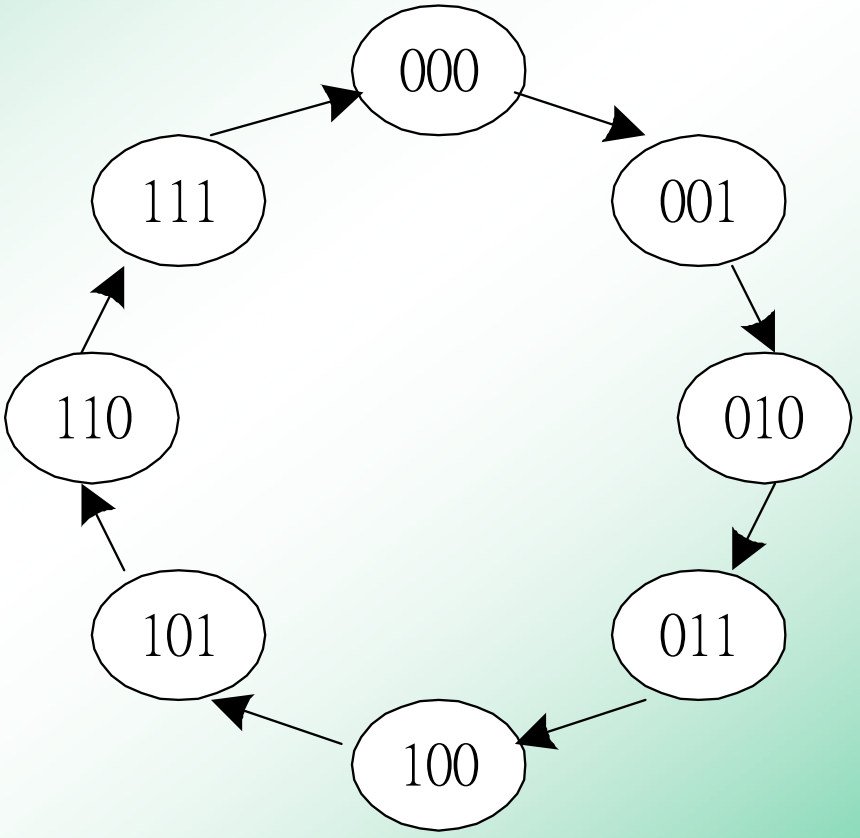


CBA : $000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 101 \rightarrow 110 \rightarrow 111$

Synchronous Counter
triggered by a clock

CBA : $000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 101 \rightarrow 110 \rightarrow 111$

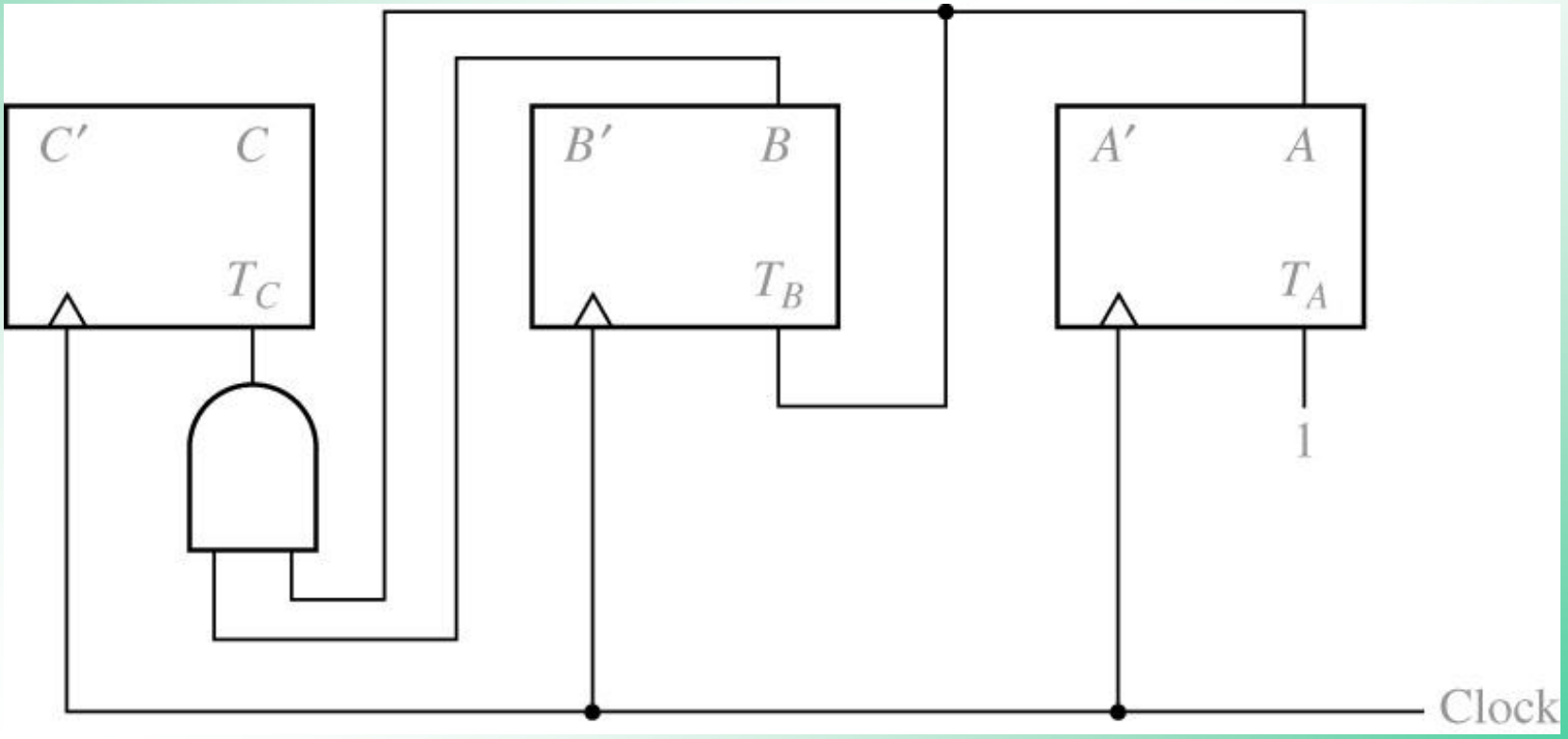
state graph



A changes (counts) as long as T-F/F(A) triggered.

B changes (counts) as long as A=1 & T-F/F(B) triggered.

C changes (counts) as long as A=B=1 & T-F/F(C) triggered.

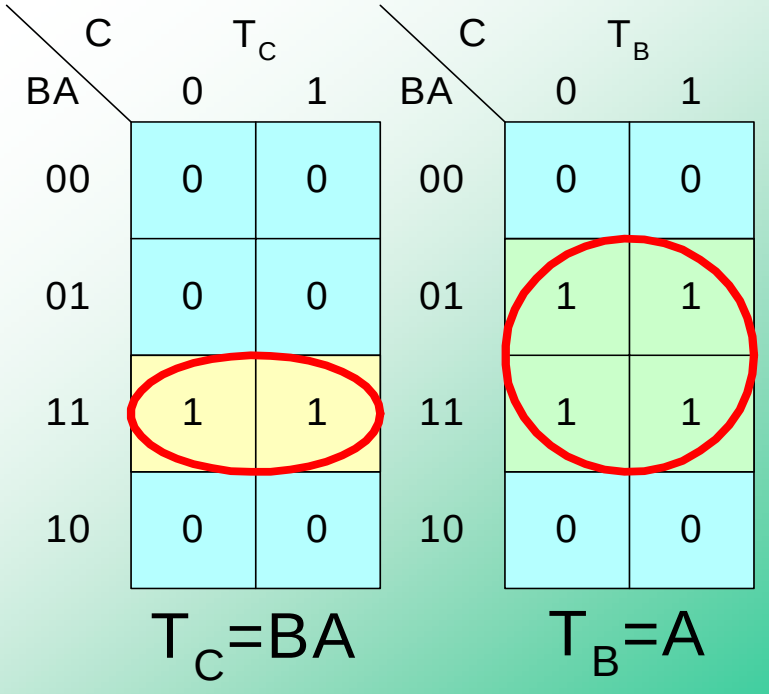


Systematic method: state table

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

Cur. state			Next state			F/F 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

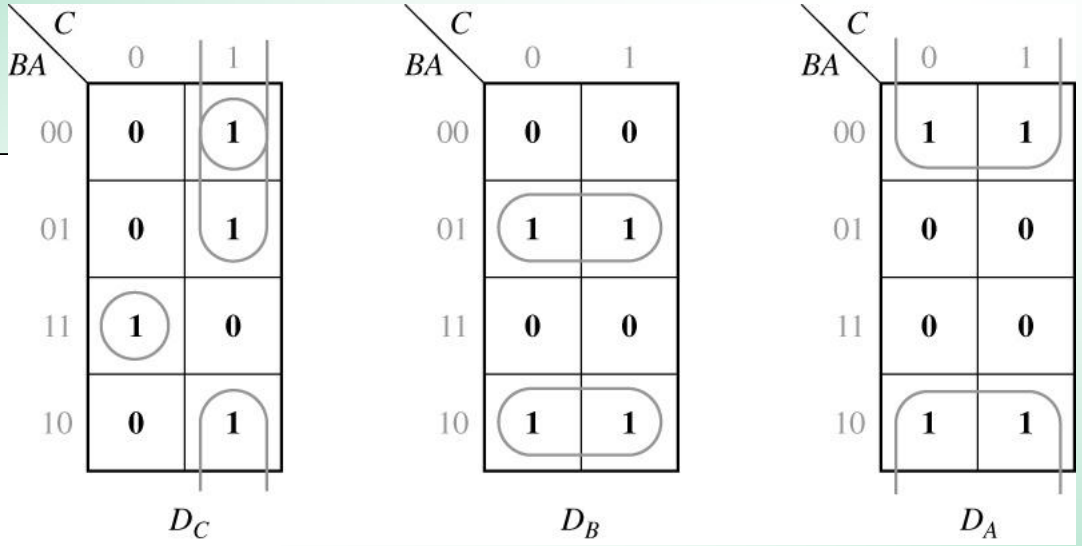
Tips: T = 1 iff Q !=Q+



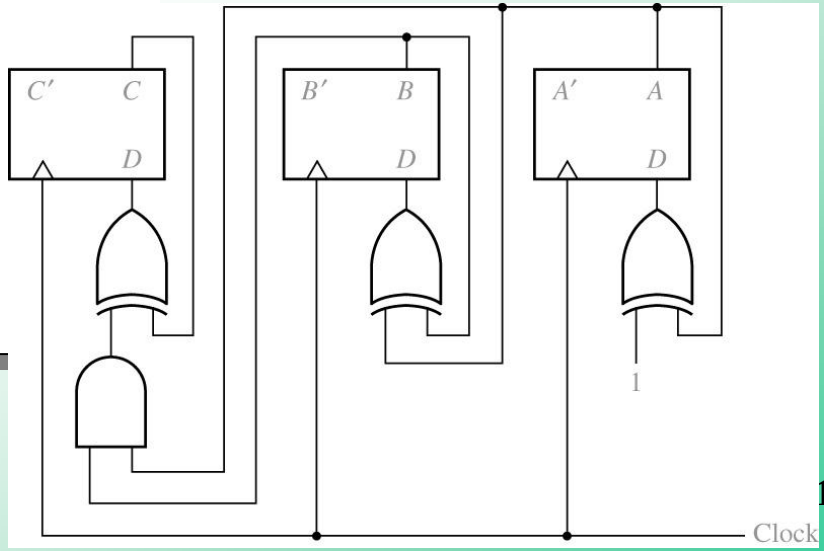
Binary counter with D F/F

Tips: Q+ = D

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

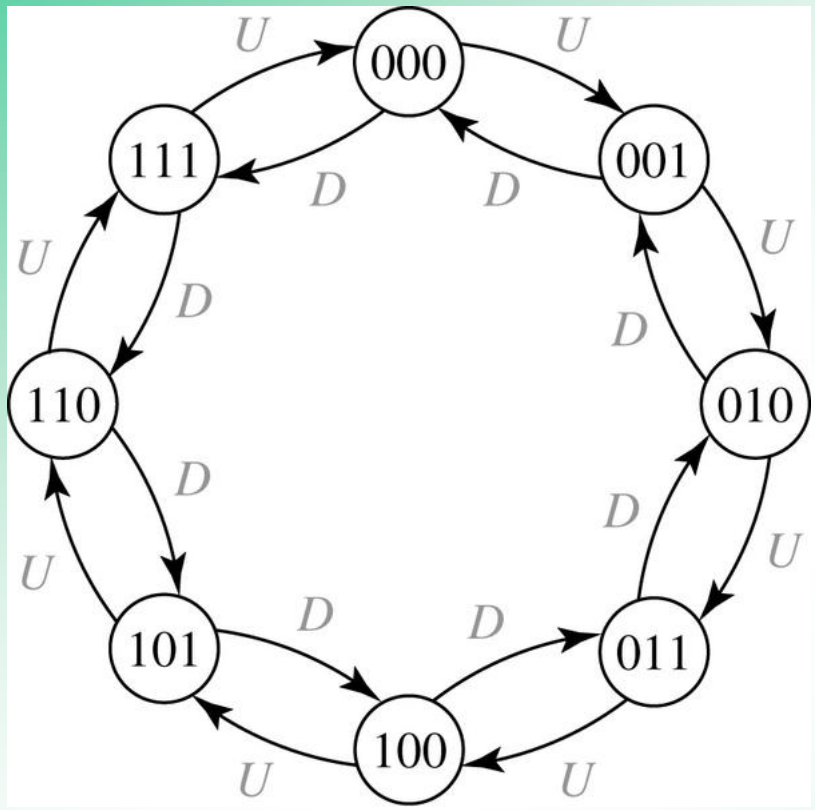


$D_C = C \oplus BA$ $D_B = B \oplus A$ $D_A = A'$



Up-down counter

State graph: U, D = (1, 1) is not allowed



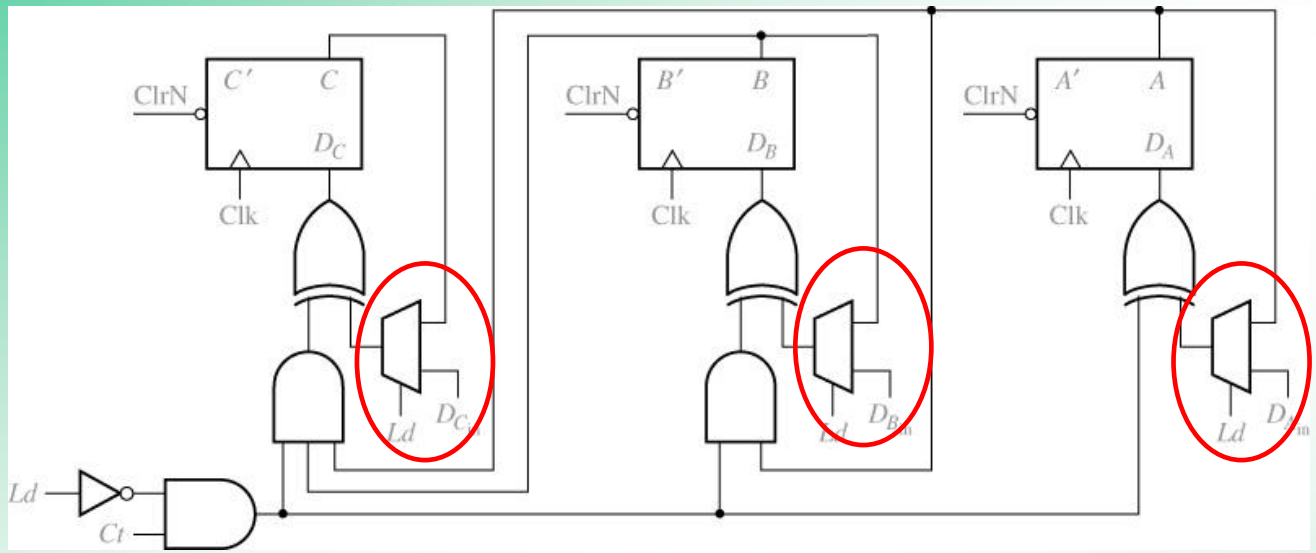
			U = 1			D = 1		
C	B	A	C^+	B^+	A^+	C^+	B^+	A^+
0	0	0	0	0	1	1	1	1
0	0	1	0	1	0	0	0	0
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	0	1	0
1	0	0	1	0	1	0	1	1
1	0	1	1	1	0	1	0	0
1	1	0	1	1	1	1	0	1
1	1	1	0	0	0	1	1	0

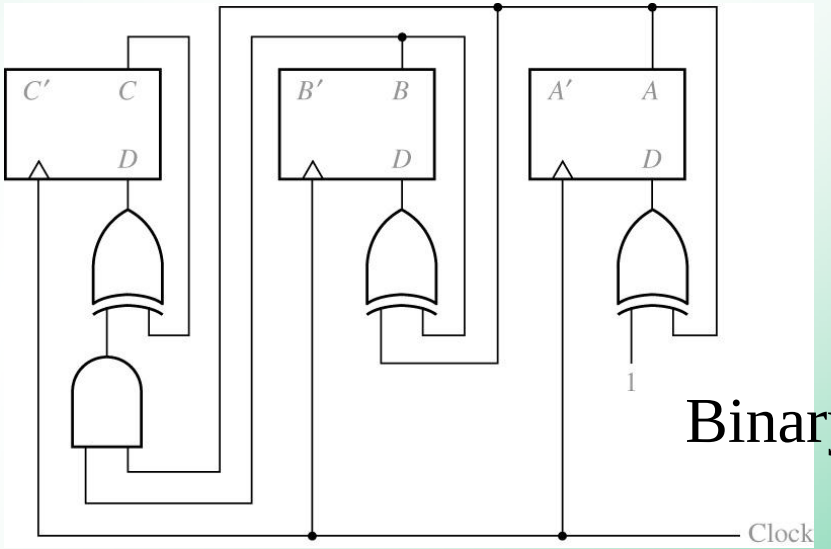
$D_A = A \oplus (U+D)$

$D_B = B \oplus (UA+DA')$

$D_C = C \oplus (UBA+DB'A')$

Loadable counter



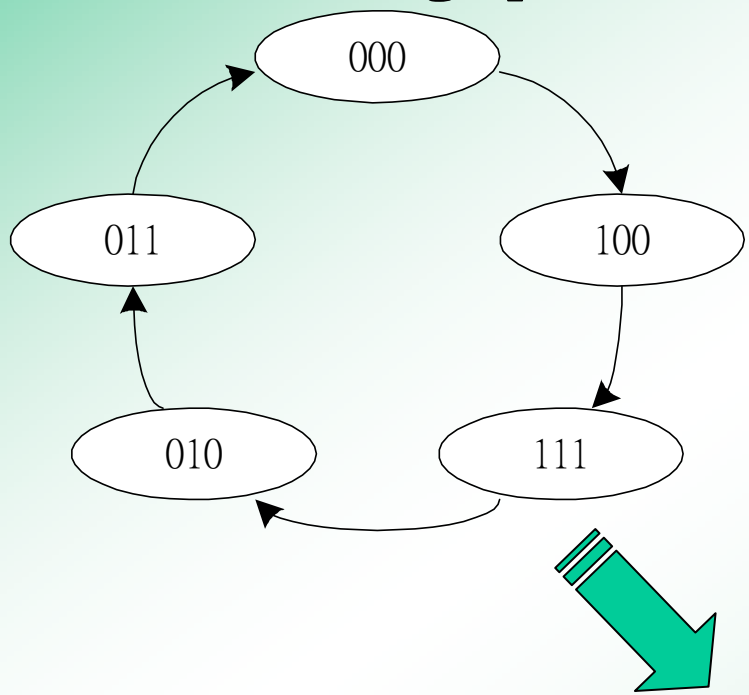


Binary counter

12-4 Counters for Other Sequences

Ex.

state graph



state table

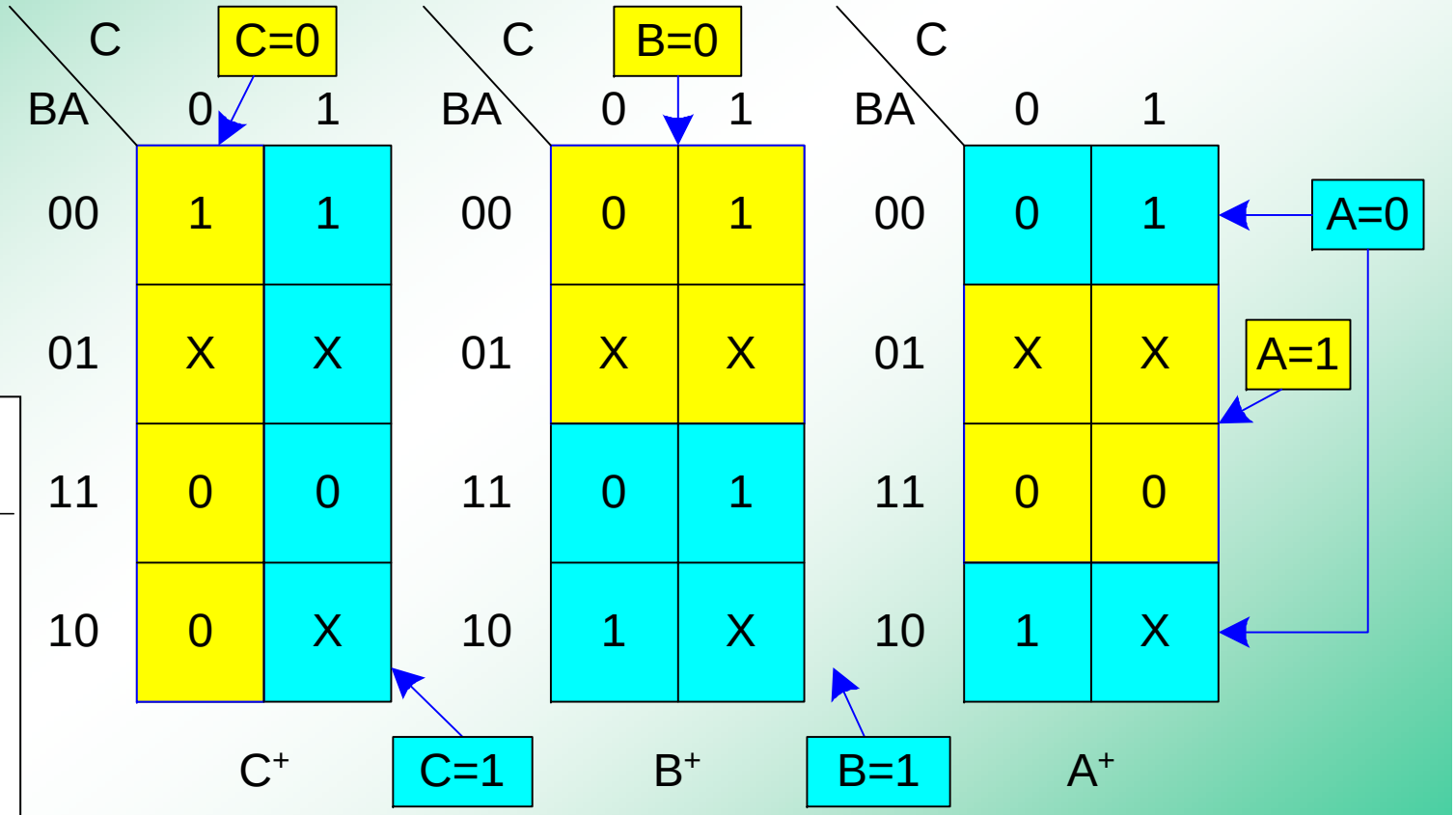
C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	1	0	0	1	0	0
0	0	1	–	–	–	–	–	–
0	1	0	0	1	1	0	0	1
0	1	1	0	0	0	0	1	1
1	0	0	1	1	1	0	1	1
1	0	1	–	–	–			
1	1	0	–	–				
1	1	1	0	1	0			

Unspecified
Don't care
1 0 status

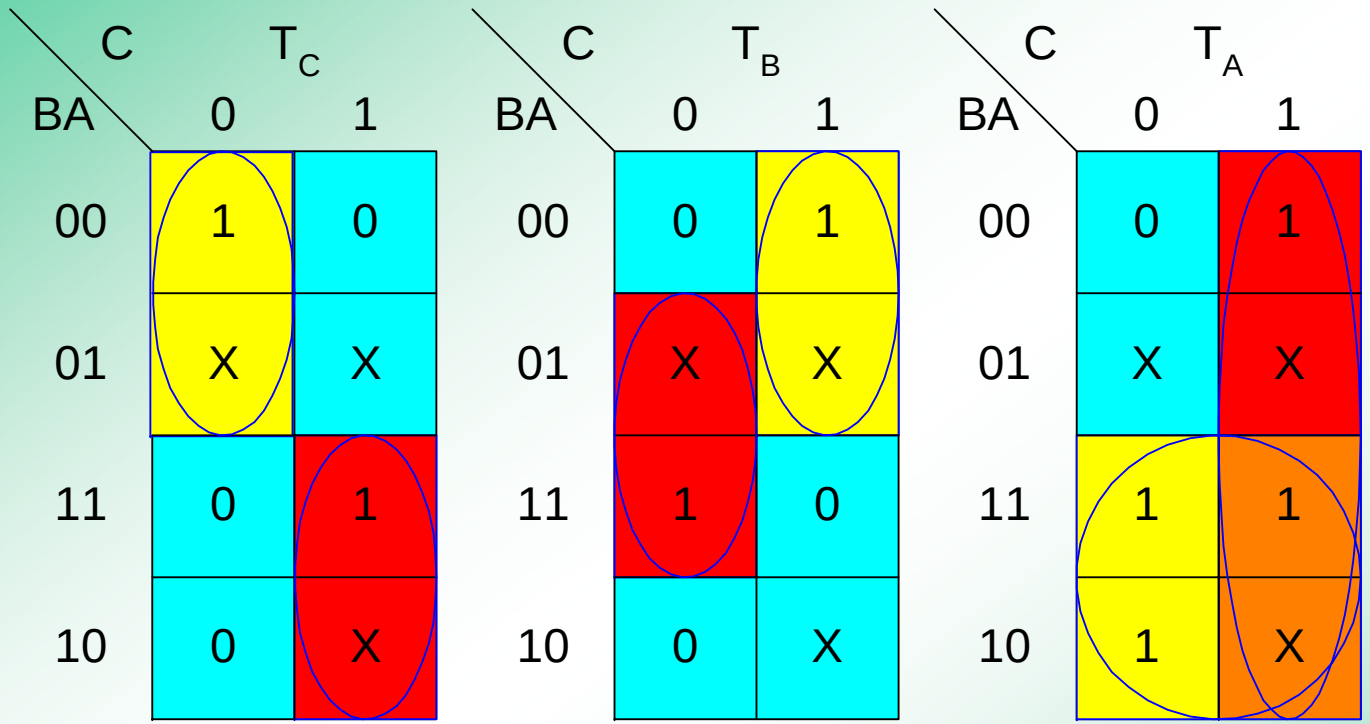
Use T-F/F to implement Next State Maps

state table

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



Derive T Maps from Next State Maps



$T_C = C'B'+CB$

$= (C'+B)(C+B')$

$T_B = C'A + CB'$

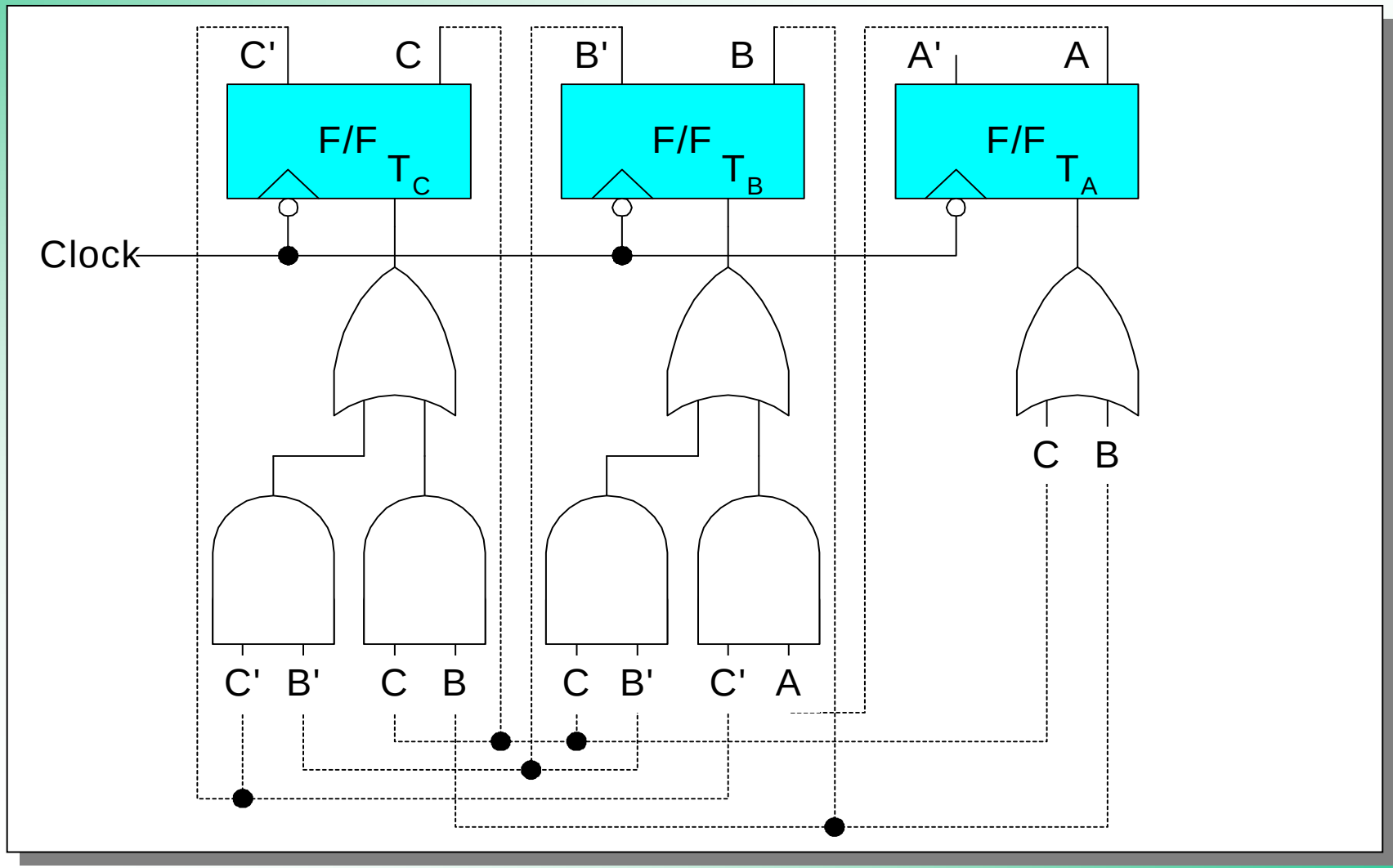
$= (C'+B')(C+A)$

$T_A = C + B$

state table

C	B	A		T_C	T_B	T_A
0	0	0		1	0	0
0	0	1		–	–	–
0	1	0		0	0	1
0	1	1		0	1	1
1	0	0		0	1	1
1	0	1		–	–	–
1	1	0		–	–	–
1	1	1		1	0	1

Implementation

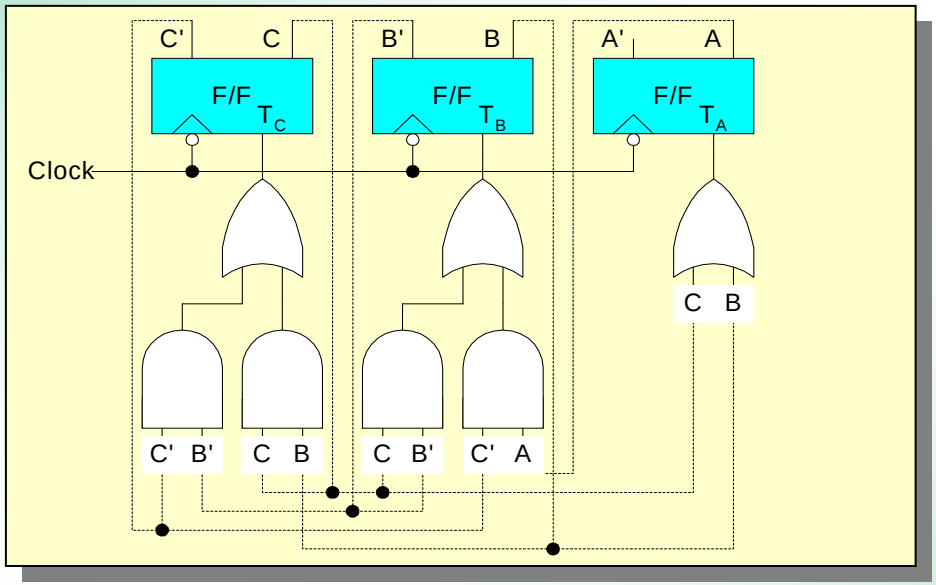


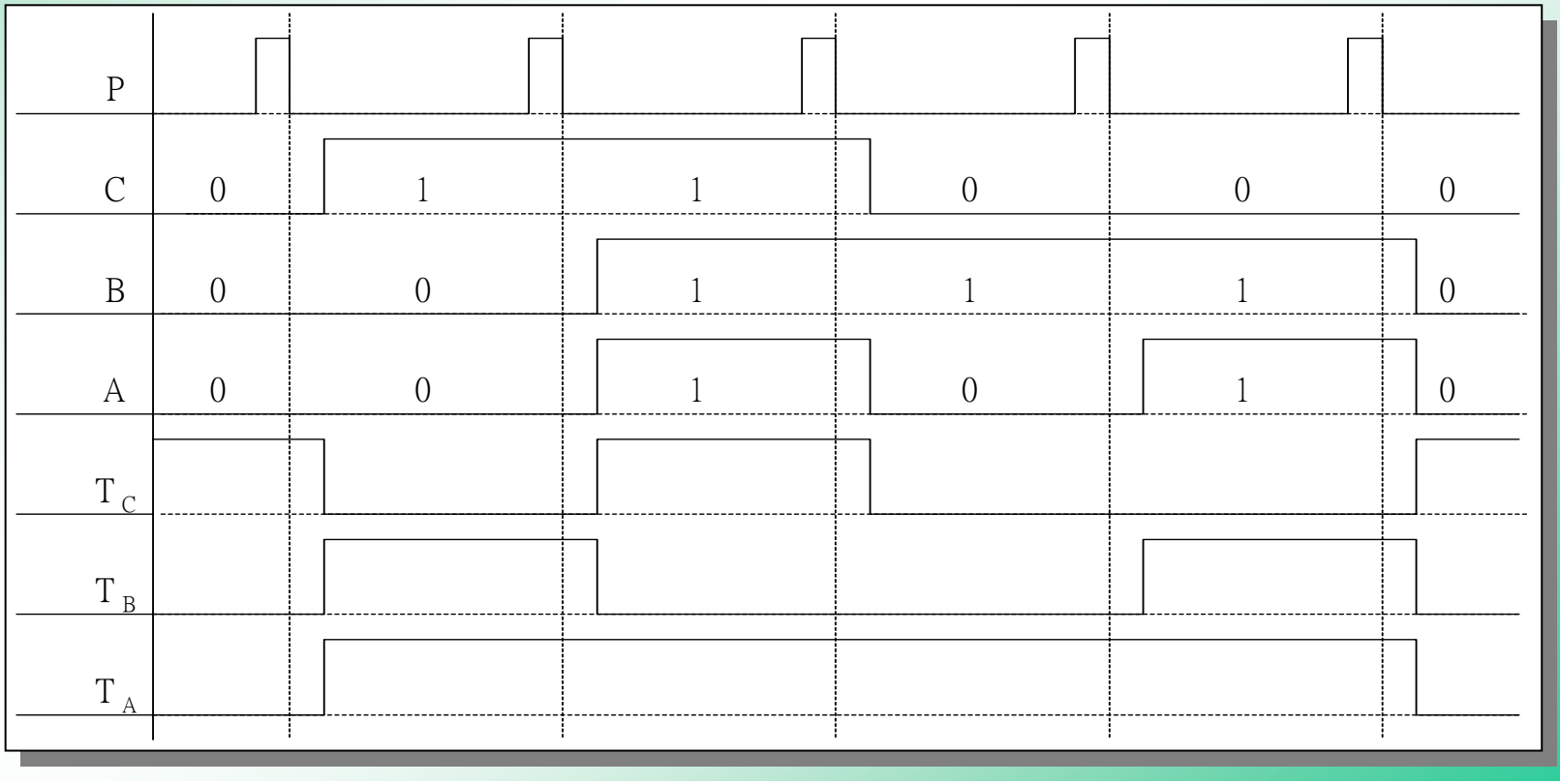
Timing Diagram

$$T_C = C'B' + CB$$

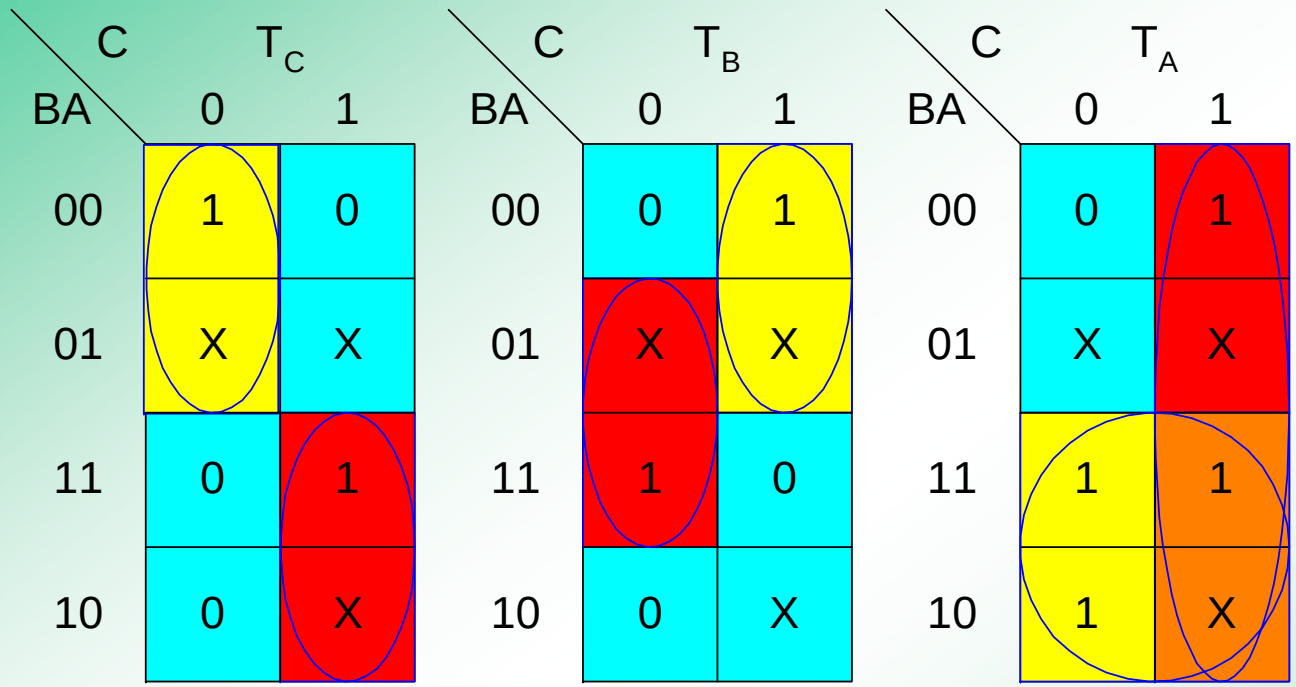
$$T_B = C'A + CB'$$

$$T_A = C + B$$





Derive T Maps from Next State Maps



$$T_C = C'B'+CB = (C'+B)(C+B')$$

$$T_B = C'A+CB' = (C'+B')(C+A)$$

$$T_A = C+B$$

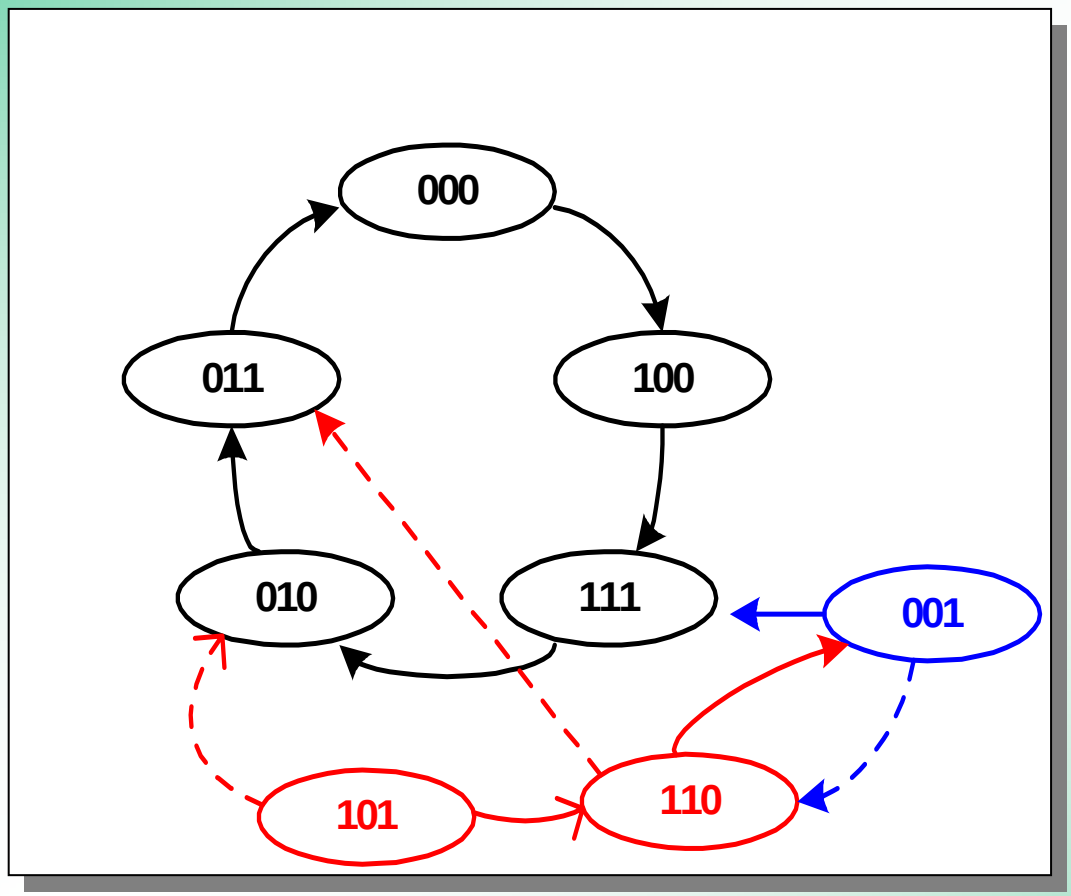
statetable

C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	1	0	0	1	0	0
0	0	1	1	1	–	1	1	–
0	1	0	0	1	1	0	0	1
0	1	1	0	0	0	0	1	1
1	0	0	1	1	1	0	1	1
1	0	1	–	1	0	–	1	1
1	1	0	0	–	1	1	–	1
1	1	1	0	1	0	1	0	1

In the design process, the next states of states 001, 101, 110 have been specified, even though they are originally not specified.

(This needs to be checked)

∵Initial power-up may lead to unpredictable states

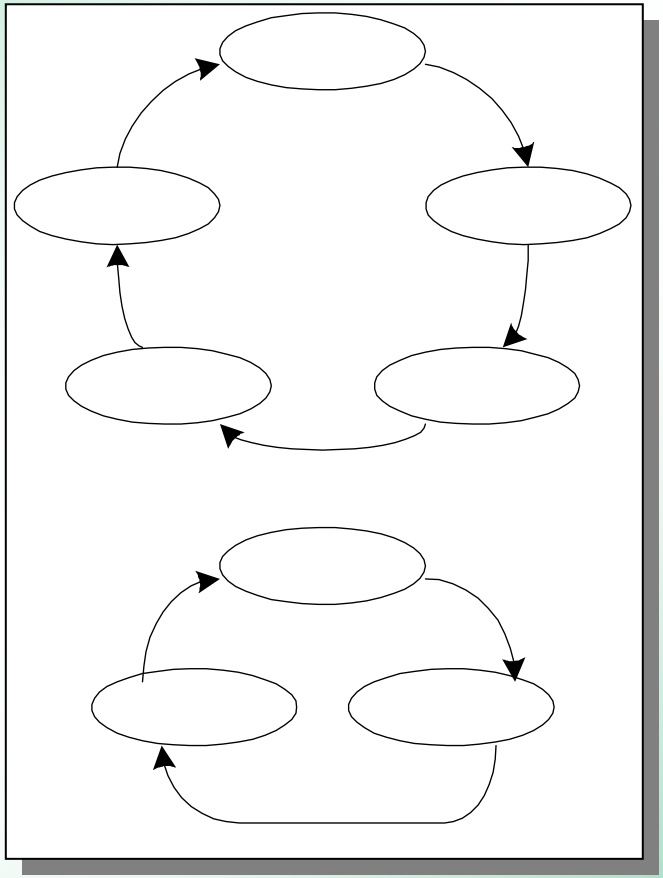


state table

C	B	A	C^+	B^+	A^+	T_C	T_B	T_A
0	0	0	1	0	0	1	0	0
0	0	1	1	1	–	1	1	–
0	1	0	0	1	1	0	0	1
0	1	1	0	0	0	0	1	1
1	0	0	1	1	1	0	1	1
1	0	1	–	1	0	–	1	1
1	1	0	0	–	1	1	–	1
1	1	1	0	1	0	1	0	1

不可有之情形

進不了main counting loop



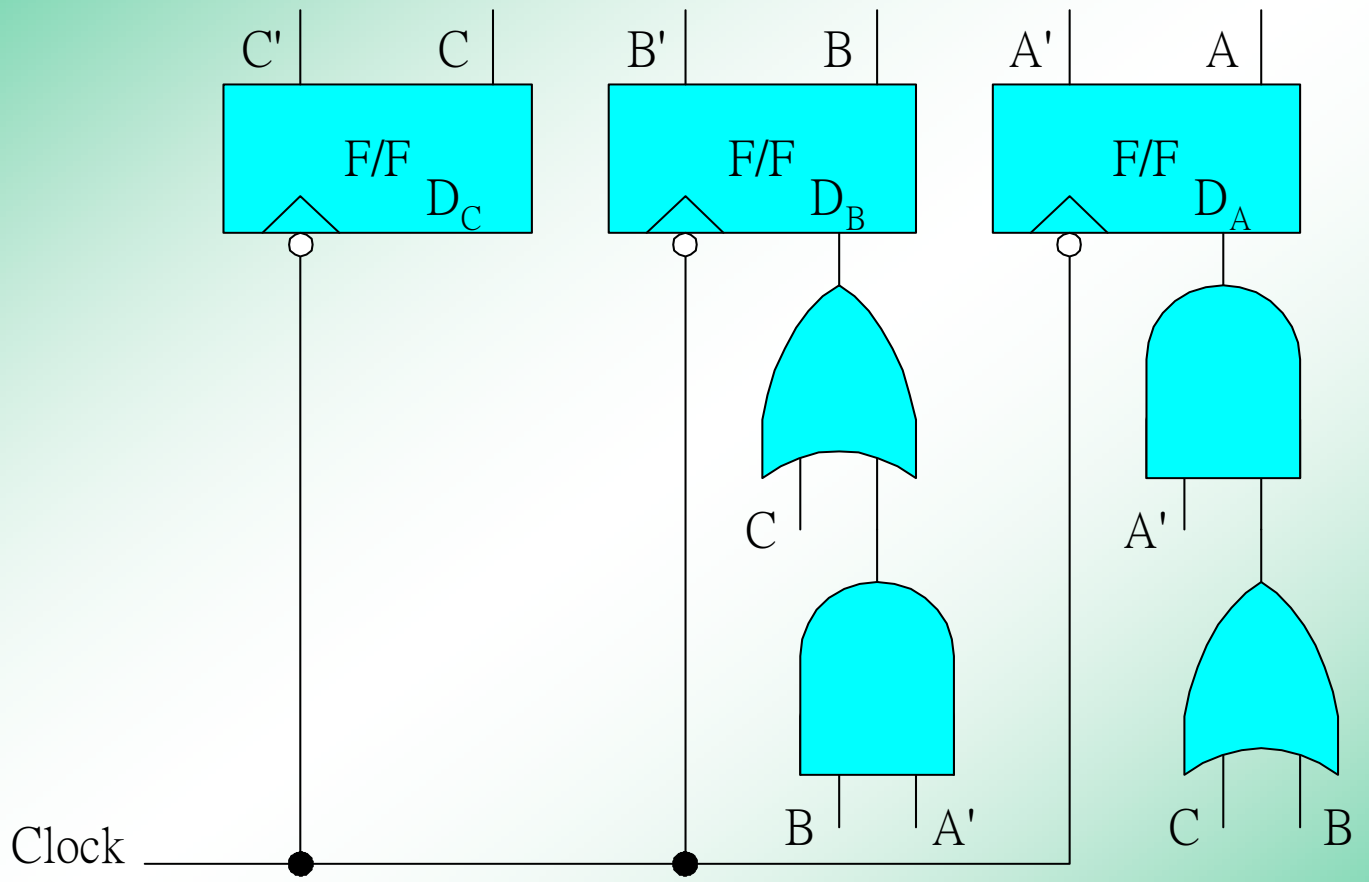
Using D-F/F´s

C	B	A	C^+	B^+	A^+	D_C	D_B	D_A
0	0	0	1	0	0	1	0	0
0	0	1	–	–	–	X	X	X
0	1	0	0	1	1	0	1	1
0	1	1	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1
1	0	1	–	–	–	X	X	X
1	1	0	–	–	–	X	X	X
1	1	1	0	1	0	0	1	0

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

D-F/F excitation table

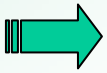
Realization



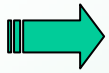
$$D_C = C^+ = \overline{B} \qquad D_B = B^+ = A + B\overline{C} \qquad D_A = C^+ = C\overline{A} + B\overline{A} = \overline{A}(C+B)$$

12-5 Counter Design Using S-R F/Fs

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	–
1	1	1	–



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

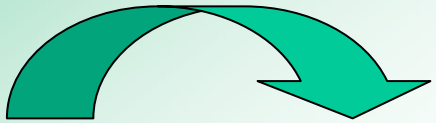


Q	Q^+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

$Q^+ = S + \overline{R}Q$

Excitation Table

Design the previous example by using SR F/F

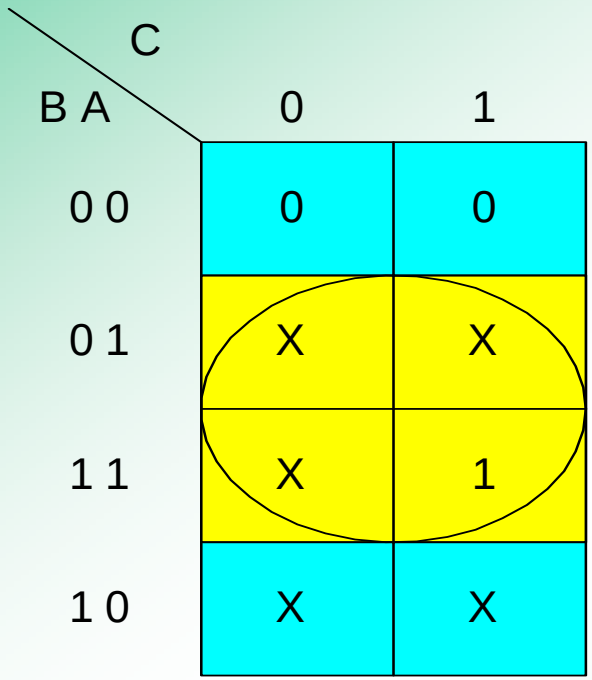


Get them by referencing excitation table

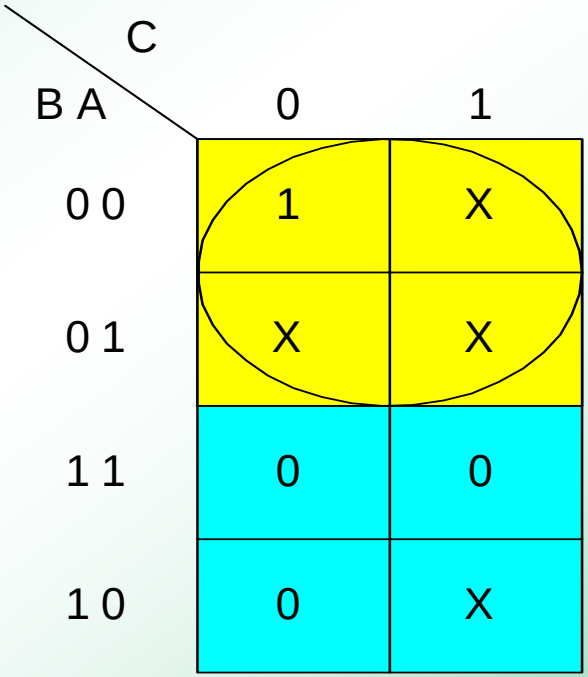
A	B	C	A^+	B^+	C^+	S_C	R_C	S_B	R_B	S_A	R_A
0	0	0	1	0	0	1	0	0	X	0	X
0	0	1	–	–	–	X	X	X	X	X	X
0	1	0	0	1	1	0	X	X	0	1	0
0	1	1	0	0	0	0	X	0	1	0	1
1	0	0	1	1	1	X	0	1	0	1	0
1	0	1	–	–	–	X	X	X	X	X	X
1	1	0	–	–	–	X	X	X	X	X	X
1	1	1	0	1	0	0	1	X	0	0	1

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

Using K-map method



$R_C = A$



$S_C = B'$

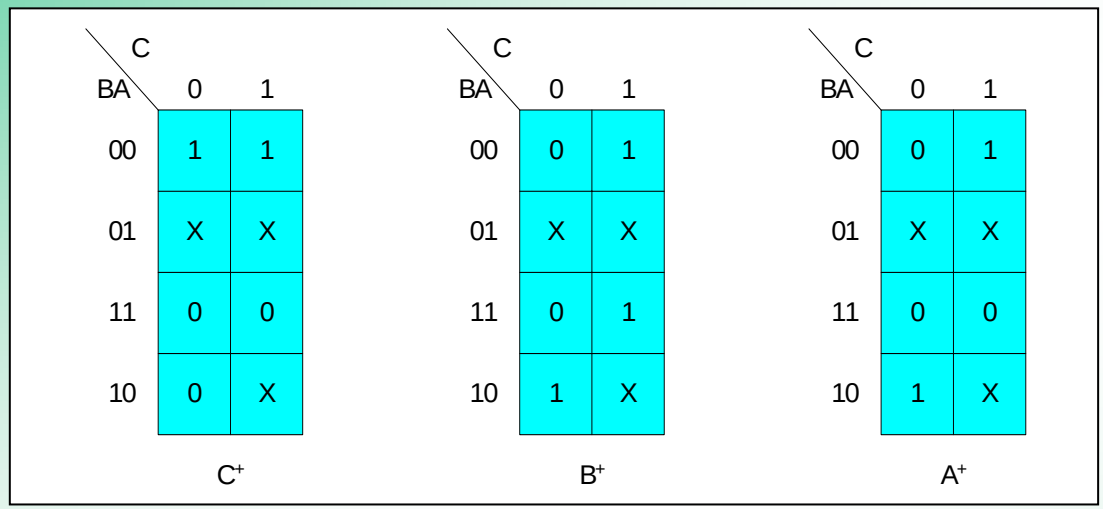
Similarly:

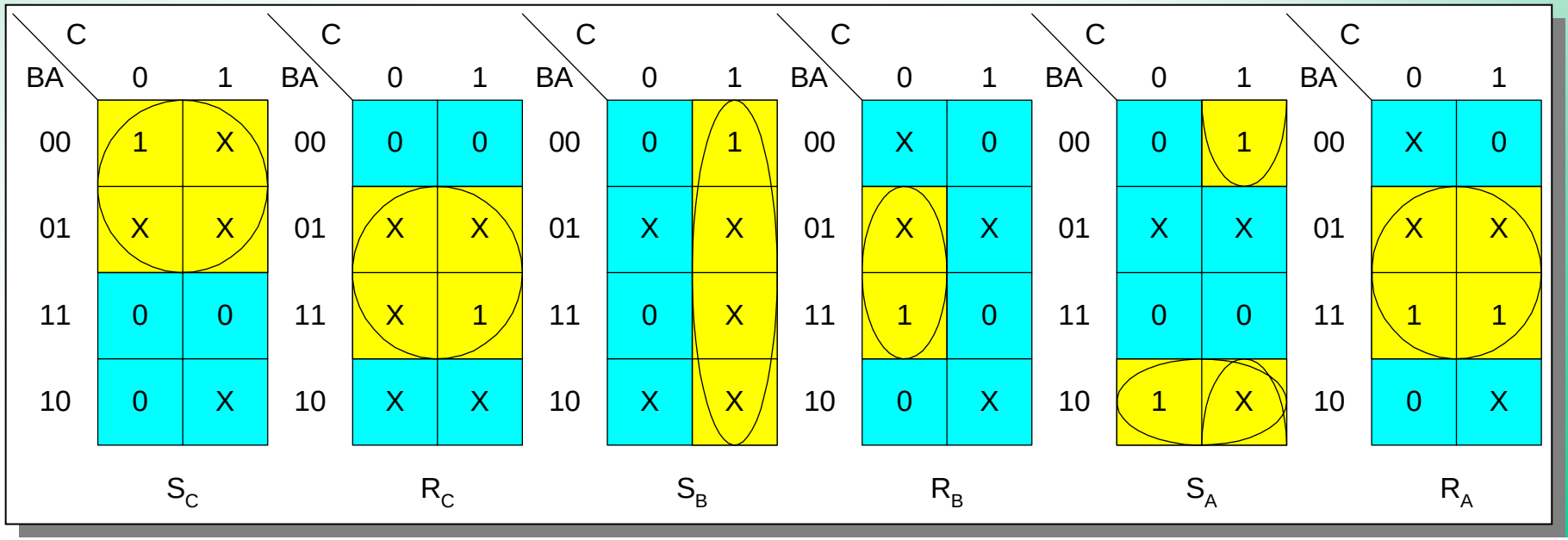
$$R_B = C'A, \quad S_B = C$$

$$R_A = A, \quad S_A = CA' + BA' = A'(C + B)$$

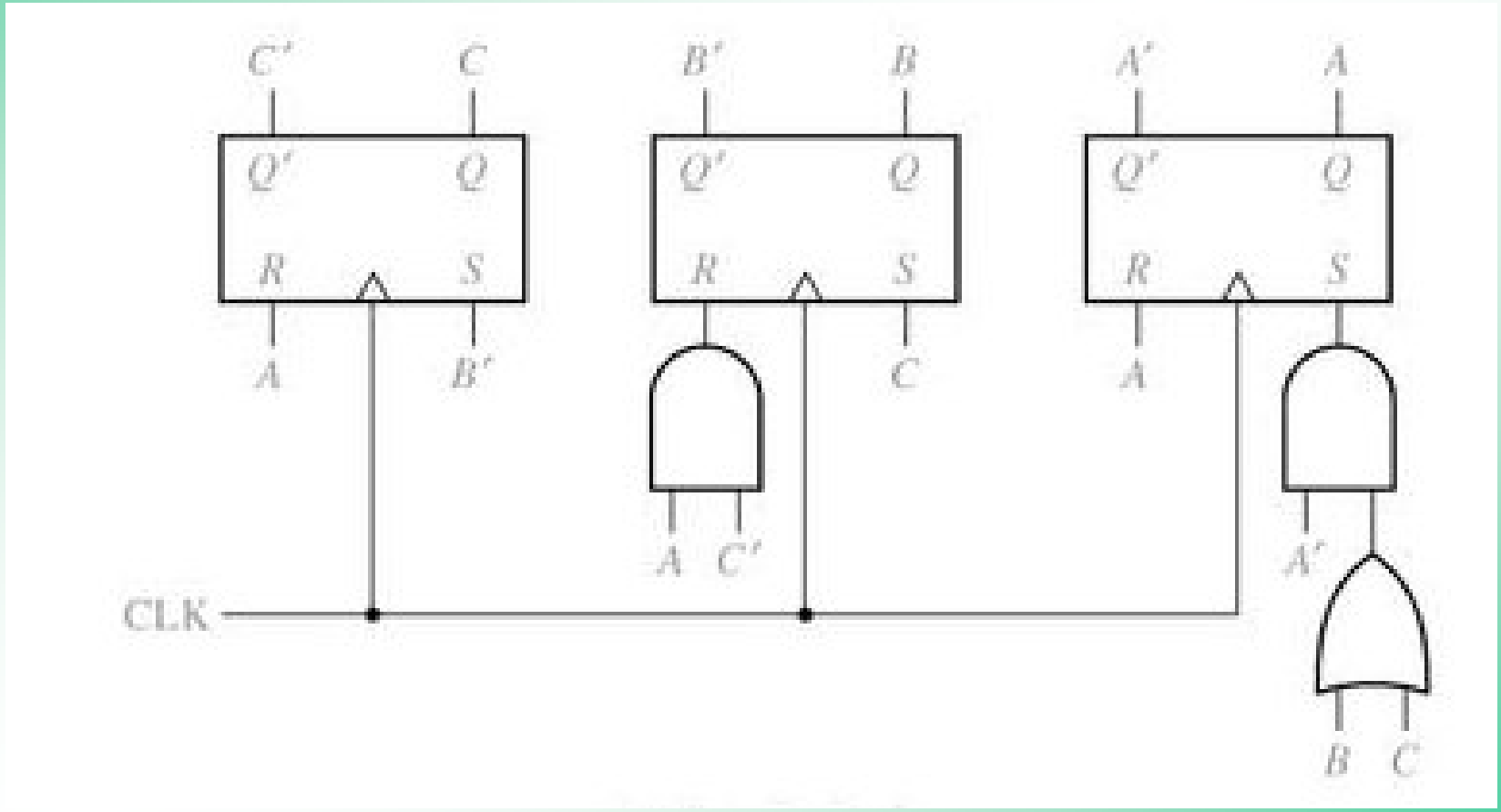
Directly use next state map to derive K-map for R_C, S_C, R_B, S_B, R_A, S_A by referencing

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





Realization of Logic Circuit



§ Counter Design Using J-K F/F

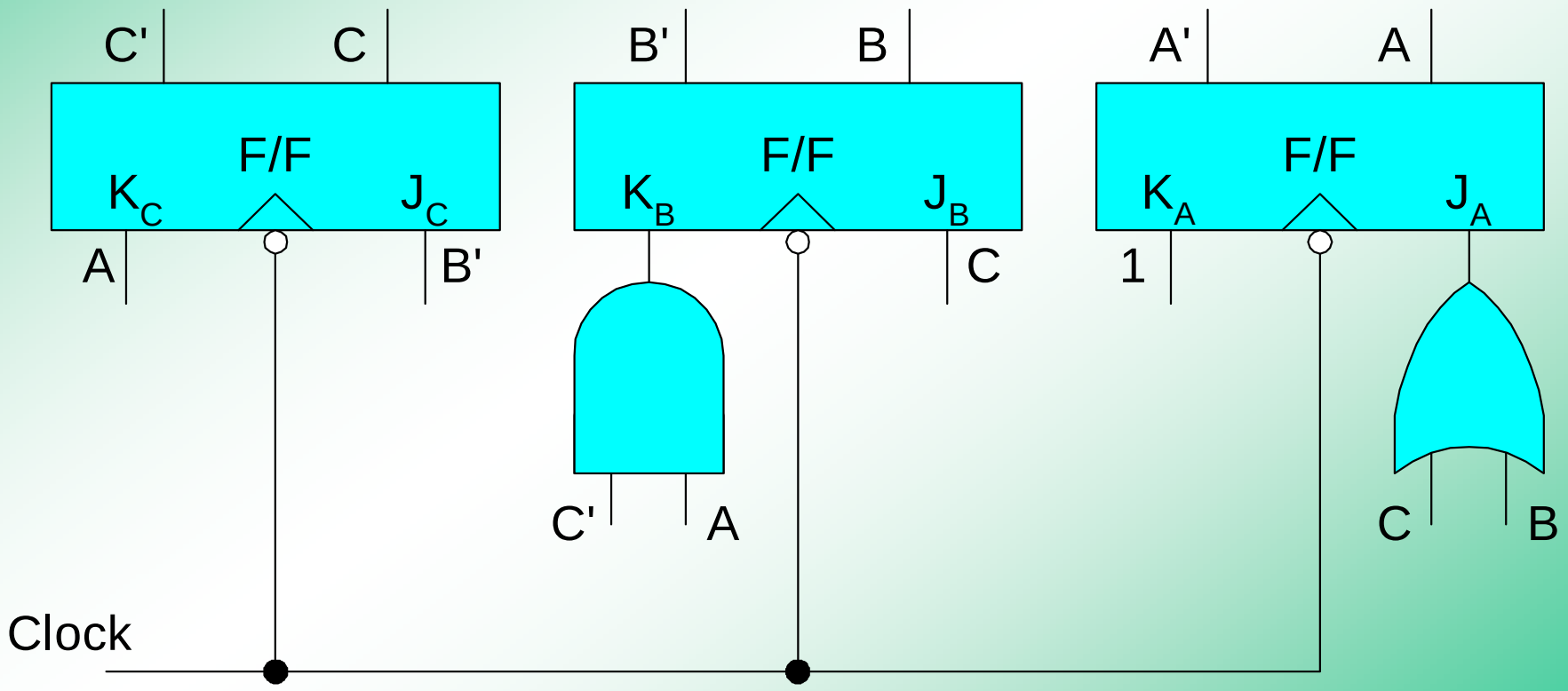
Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

JK-F/F excitation table

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

$J_C = B'$ $K_C = A$ $J_B = C$ $K_B = C'A$ $J_A = C + B$ $K_A = 1$

Realization



12-6 Summary
how to derive F/F input equations

Type of FF	Input	Q=0		Q=1		Rules	
		Q+=0	Q+=1	Q+=0	Q+=1	Q=0 half	Q=1 half
Delay	D	0	1	0	1	No change	No change
Trigger	T	0	1	1	0	No change	Complement
Set-Reset	S	0	1	0	X	No change	Replace 1's with X's
	R	X	0	1	0	Replace 0's with X's	Complement
J-K	J	0	1	X	X	No change	Fill in with X's
	K	x	x	1	0	Fill in with X's	complement

Always copy 'x' from the next state map on the input maps first

Sequential circuit design with a single pulse to imitate state changes and no other inputs

Ex. 8421 BCD to Excess-3 code converter

A	B	C	D	A^+	B^+	C^+	D^+
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Initially BCD code stored in F/F. After a clock corresponding Excess-3 code outputs

A	B	C	D	A^+	B^+	C^+	D^+	J_A	K_A	J_B	K_B	J_C	K_C	J_D	K_D
0	0	0	0	0	0	1	1	0	X	0	X				
0	0	0	1	0	1	0	0	0	X	1	X				
0	0	1	0	0	1	0	1	0	X	1	X				
0	0	1	1	0	1	1	0	0	X	1	X				
0	1	0	0	0	1	1	1	0	X	X	0				
0	1	0	1	1	0	0	0	1	X	X	1				
0	1	1	0	1	0	0	1	1	X	X	1				
0	1	1	1	1	0	1	0	1	X	X	1				
1	0	0	0	1	0	1	1	X	0	0	X				
1	0	0	1	1	1	0	0	X	0	1	X				

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

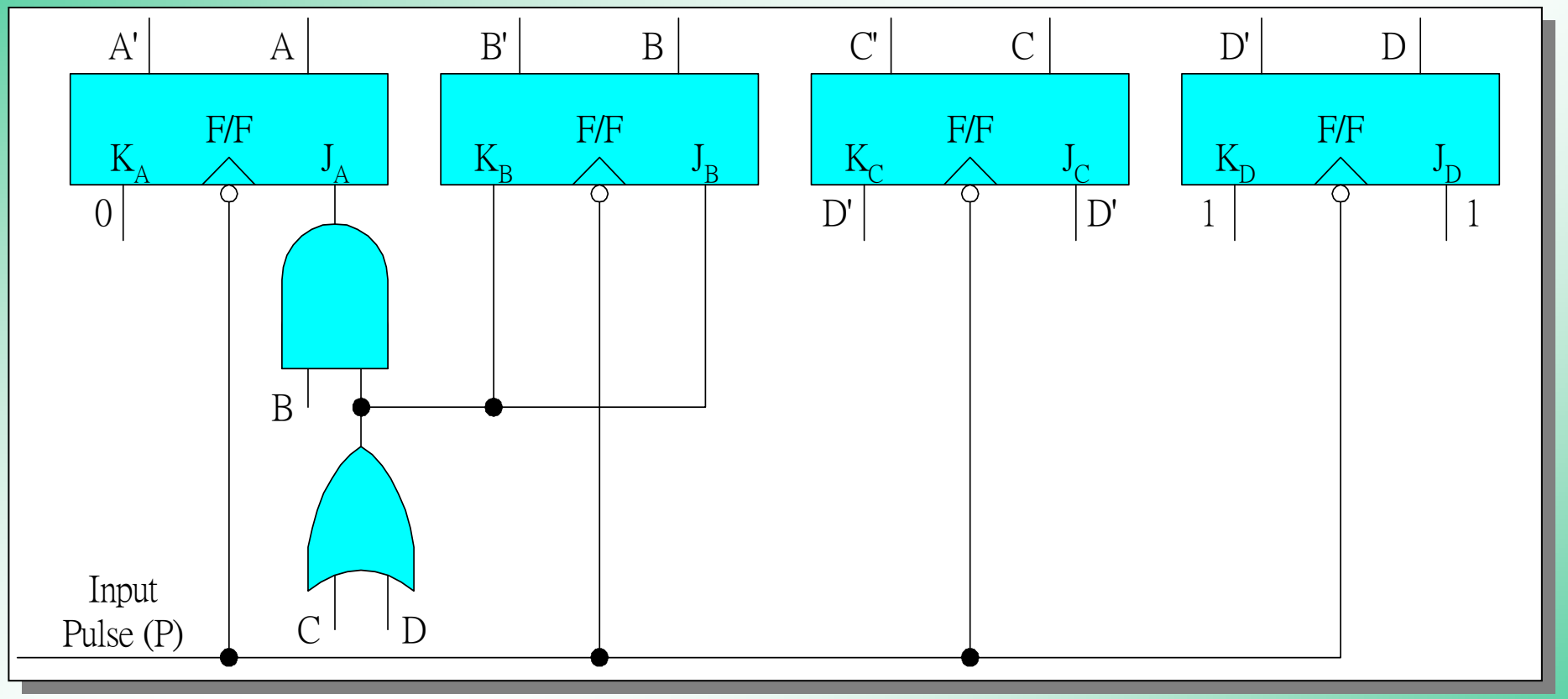
$$J_A = BC + BD = B(C + D) \qquad K_A = 0$$

$$J_B = C + D \qquad K_B = C + D$$

$$J_C = \overline{D} \qquad K_C = \overline{D}$$

$$J_D = 1 \qquad K_D = 1$$

Ex. 8421 BCD to Excess-3 code converter



Initially BCD code stored in F/F.

After a clock corresponding Excess-3 code outputs.

HOMEWORK -- Unit 12

- 12.2
- 12.11
- 12.15
- 12.17(b)(e)