Unit V

Synchronous Counters

by

Dr. Krishan Arora
Associate Professor and Head
Lovely Professional University

Lock Out condition in Synchronous Counters

- Sometimes a counter may find itself in some unused states.
- It happens when the next state of some unused state is again some unused one and if by chance the counter happens to find itself in some unused state and never arrives at in used state then this condition is called "Lock Out".

- **Solution:**

An **additional circuit** is required to ensure that "lock out" does not occur. The counter should be designed using the next state to be initial state from the unused state.

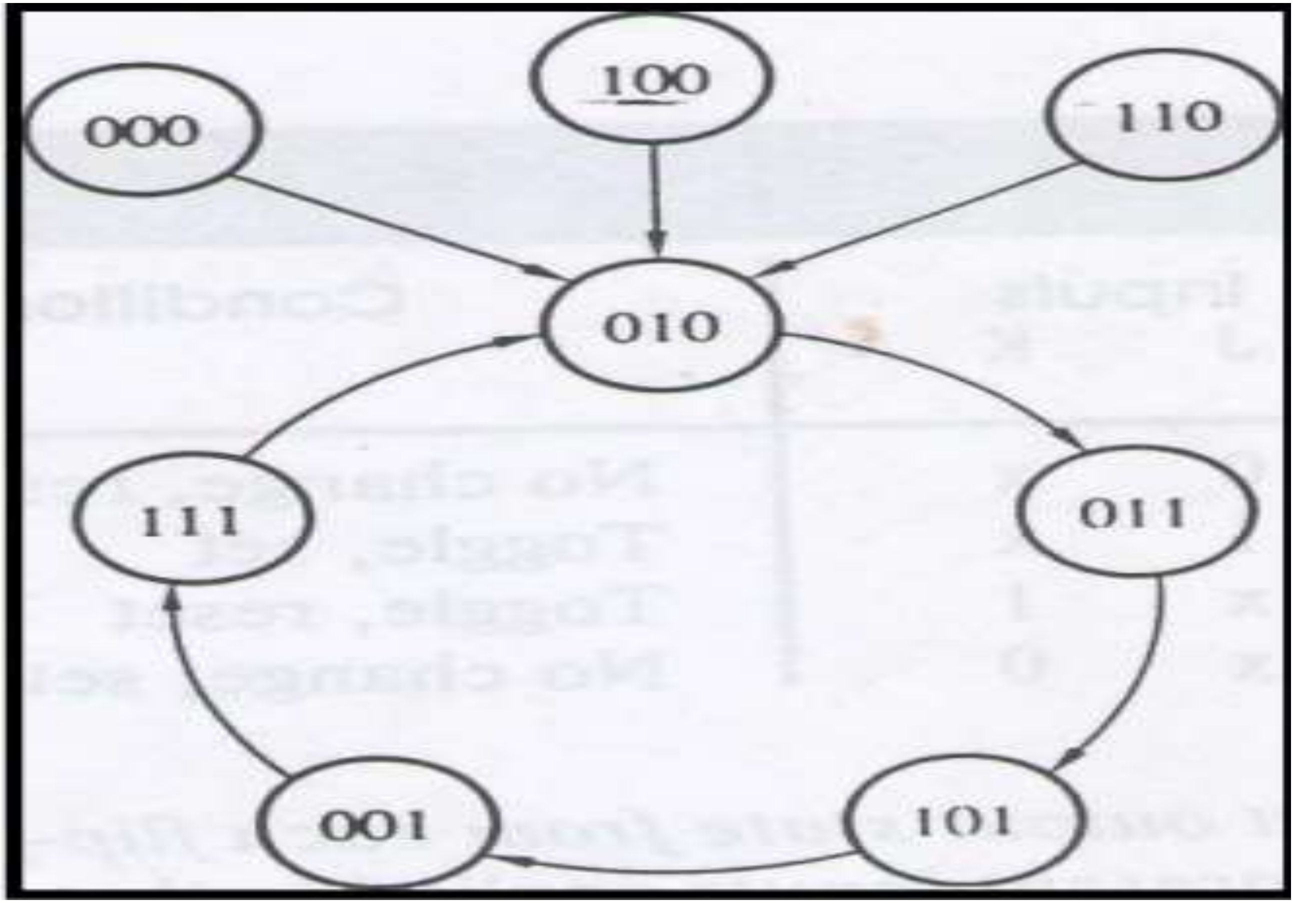
EXAMPLE : Synchronous Counter Design Problem:

Design a MOD-5, 3-bit synchronous counter to count in the following sequence: 2, 3, 5, 1, 7. The counter must be self-starting with the count states of 0, 4, and 6 leading directly to 2.

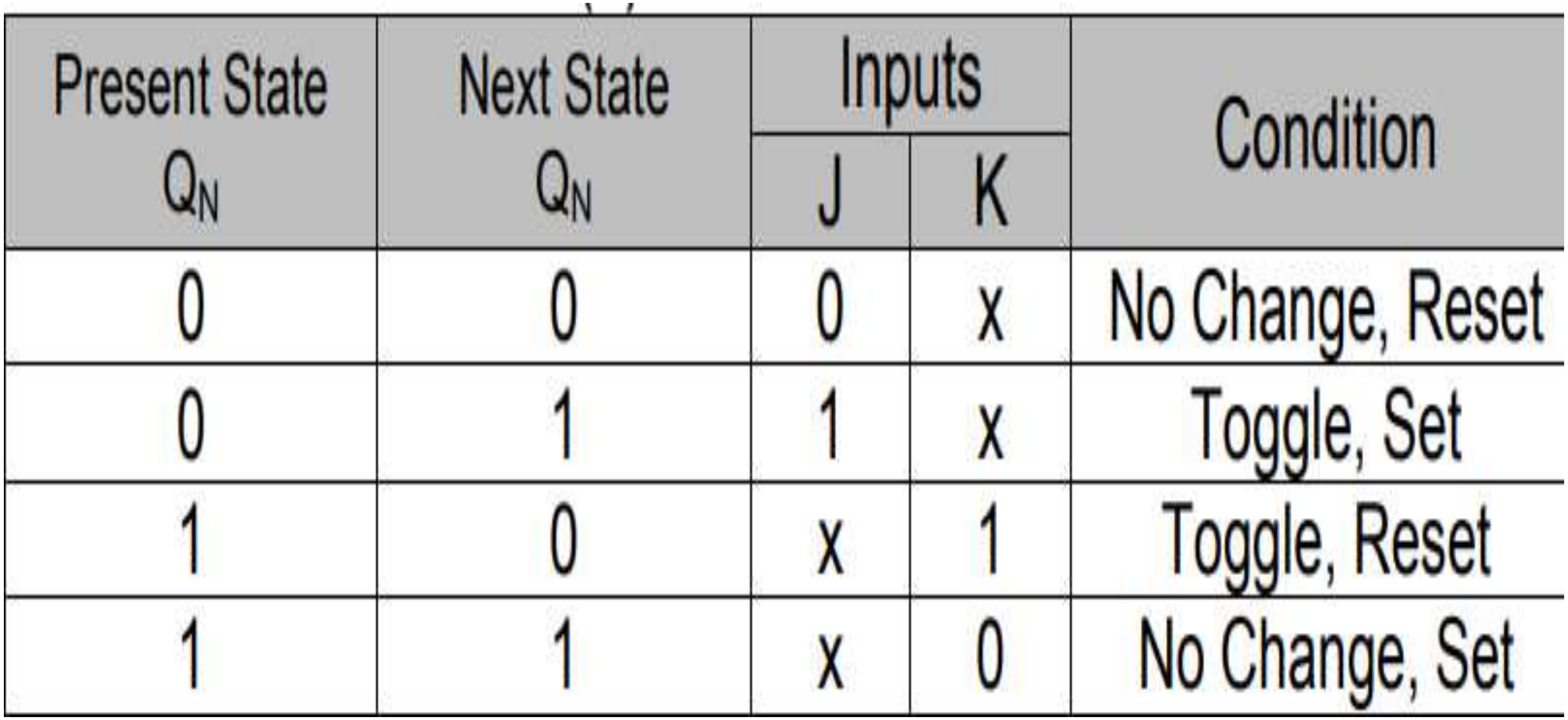
Solution

- Follow these procedures:
- 1. Create a state transition diagram.
- 2. Create a counter excitation table by listing the present state and next state sequence.
- 3. Expand the table by adding the J and K input states for each flip
- 4. Determine the logic functions for the J and K inputs as a function of the present states.
- 5. Analyze the counter to verify the design.
- 6. Construct and test the counter

- The state transition diagram for the counter is shown in Figure

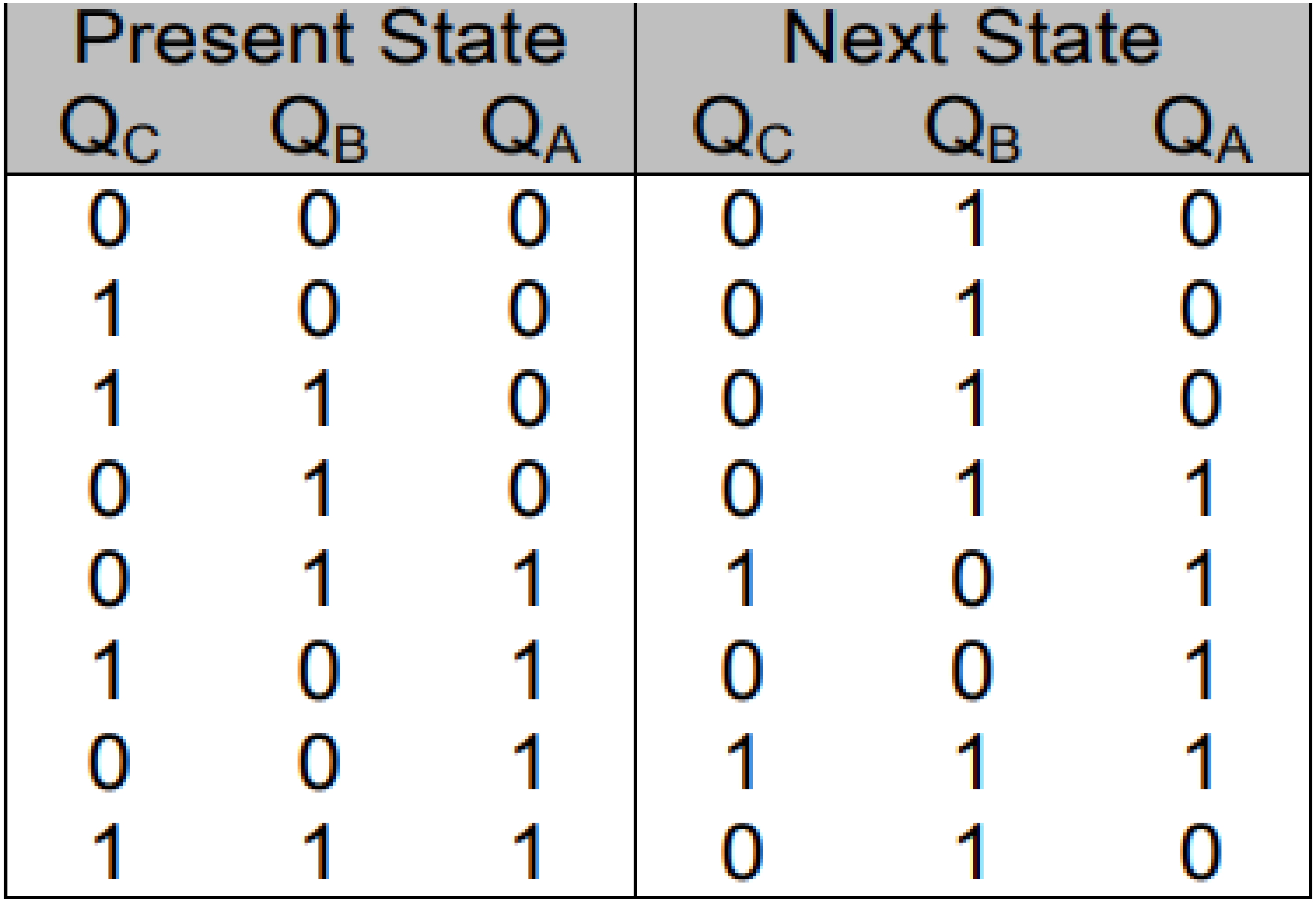


J-K Excitation Table



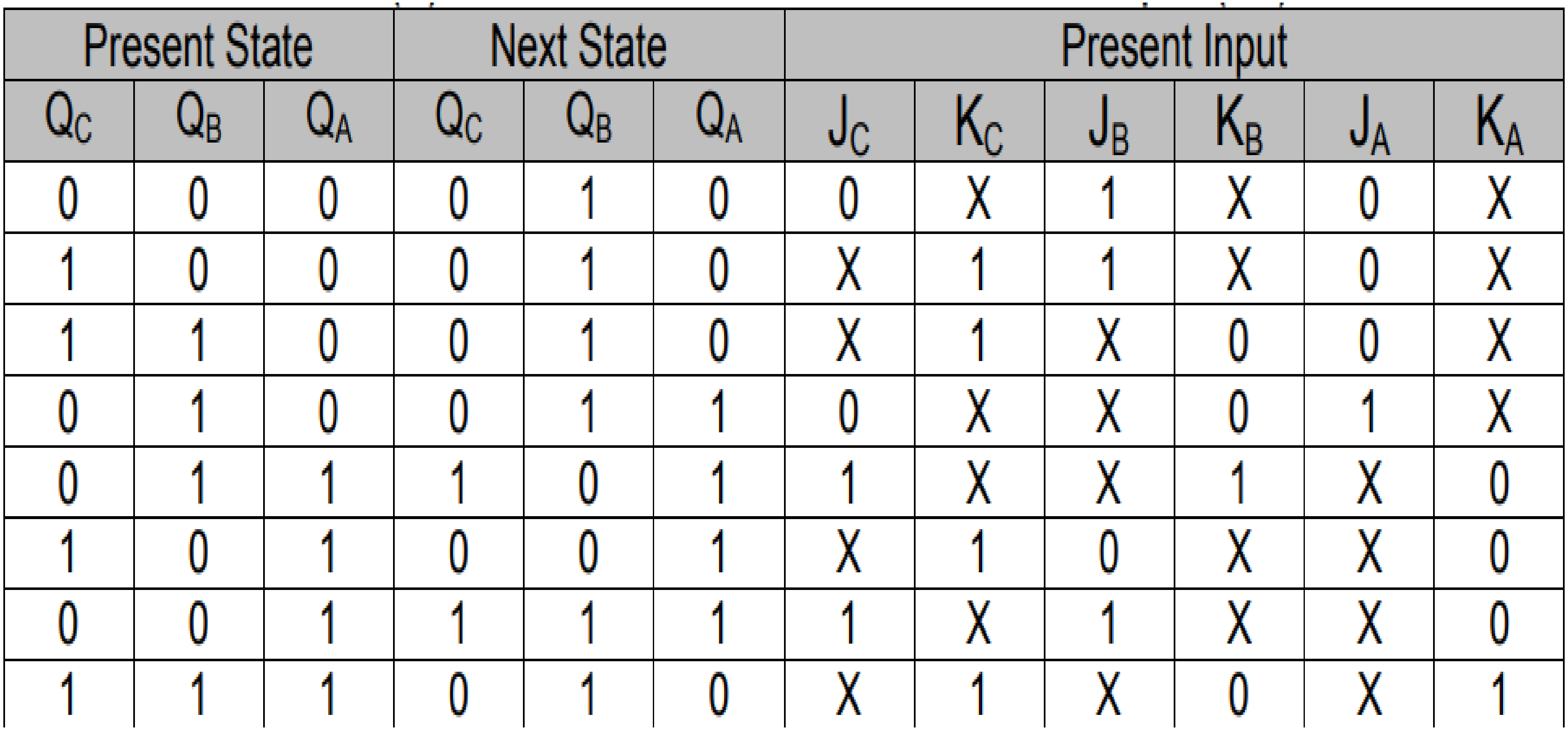
Present State Q_N	Next State Q_N	Inputs		Condition
		J	K	
0	0	0	x	No Change, Reset
0	1	1	x	Toggle, Set
1	0	x	1	Toggle, Reset
1	1	x	0	No Change, Set

Present State-Next State Table



Present State			Next State		
Q_C	Q_B	Q_A	Q_C	Q_B	Q_A
0	0	0	0	1	0
1	0	0	0	1	0
1	1	0	0	1	0
0	1	0	0	1	1
0	1	1	1	0	1
1	0	1	0	0	1
0	0	1	1	1	1
1	1	1	0	1	0

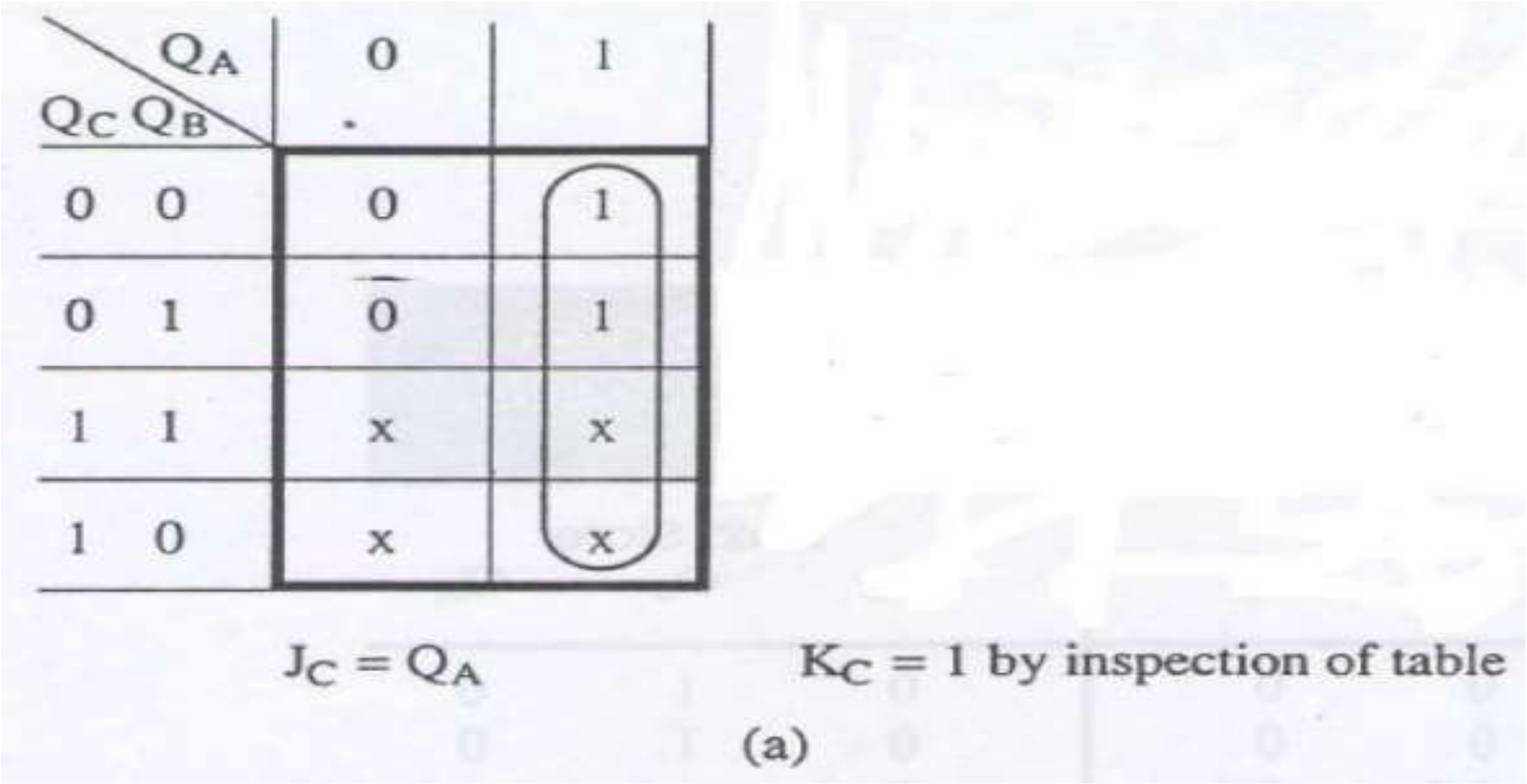
Counter Excitation Table



Present State			Next State			Present Input					
Q_C	Q_B	Q_A	Q_C	Q_B	Q_A	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	1	0	0	X	1	X	0	X
1	0	0	0	1	0	X	1	1	X	0	X
1	1	0	0	1	0	X	1	X	0	0	X
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	1	1	X	X	1	X	0
1	0	1	0	0	1	X	1	0	X	X	0
0	0	1	1	1	1	1	X	1	X	X	0
1	1	1	0	1	0	X	1	X	0	X	1

- The above Table do not follow counting order, so be careful when filling in the K-maps. The K-maps and resulting logic functions for the J and K inputs are shown in Figure

K-maps

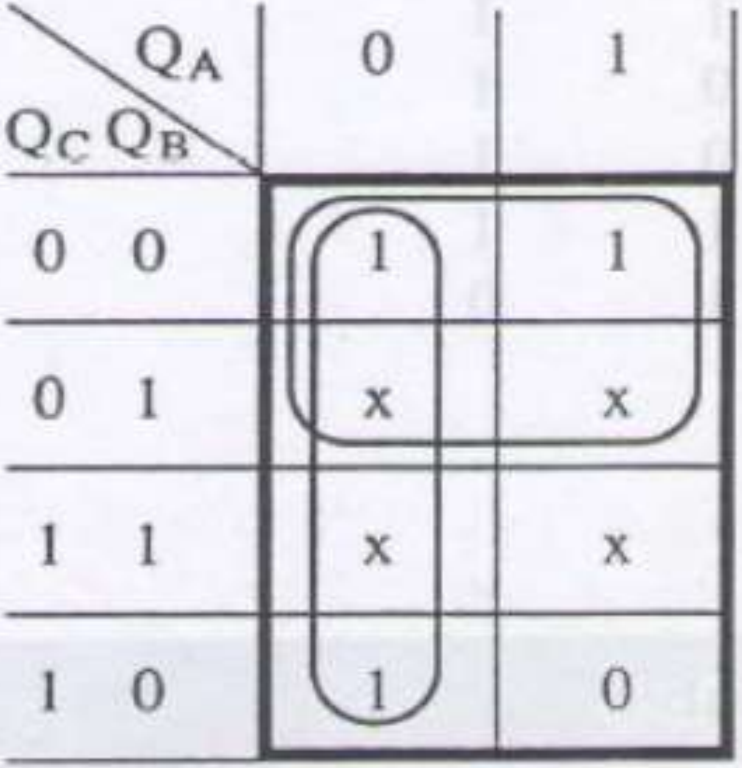


$Q_C Q_B$ \ Q_A	0	1
0 0	0	1
0 1	0	1
1 1	x	x
1 0	x	x

$J_C = Q_A$

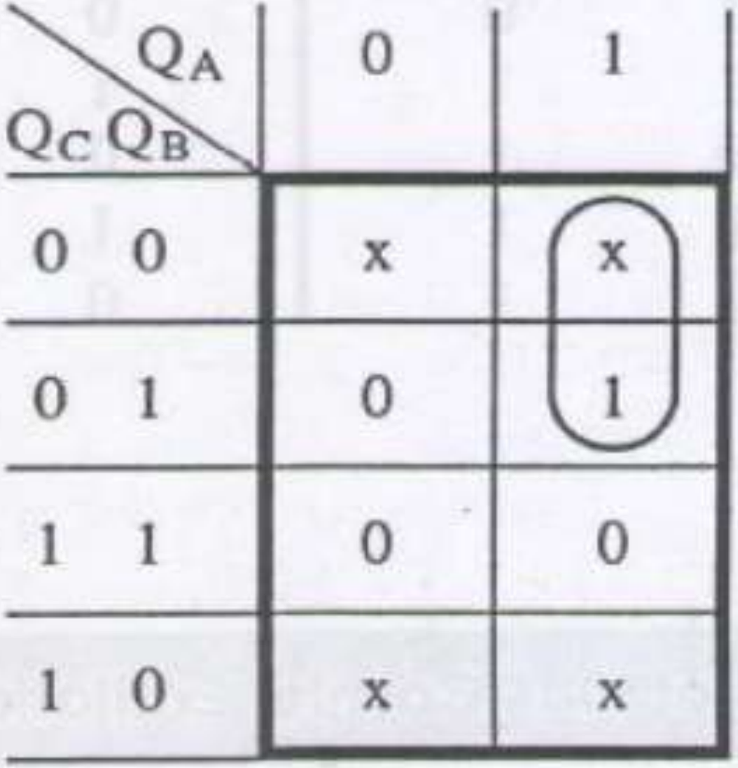
$K_C = 1$ by inspection of table

(a)



$Q_C Q_B$ \ Q_A	0	1
0 0	1	1
0 1	x	x
1 1	x	x
1 0	1	0

$J_B = \overline{Q_C Q_A}$



$Q_C Q_B$ \ Q_A	0	1
0 0	x	x
0 1	0	1
1 1	0	0
1 0	x	x

$K_B = \overline{Q_C} Q_A$

(b)

$Q_C\,Q_B$ \ Q_A	0	1
0 0	0 0	x 1
0 1	1 2	x 3
1 1	0 6	x 7
1 0	0 4	x 5

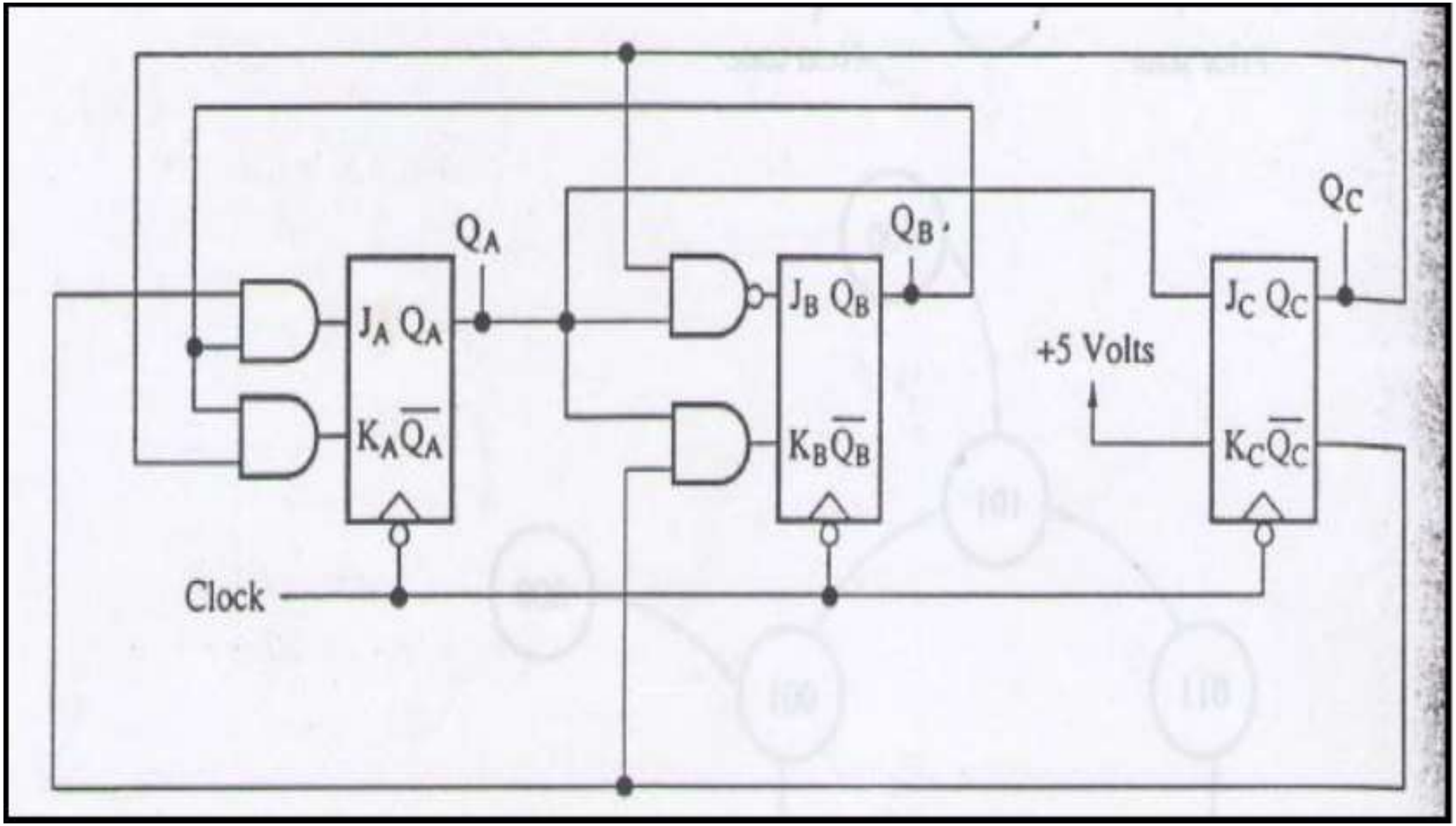
$J_A = \overline{Q_C}Q_B$

$Q_C\,Q_B$ \ Q_A	0	1
0 0	x	0
0 1	x	0
1 1	x	1
1 0	x	0

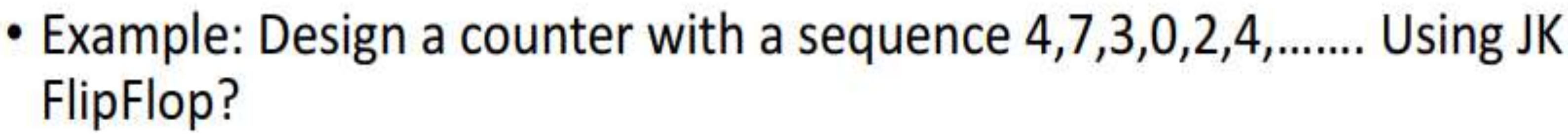
$K_A = Q_CQ_B$

(c)

Synchronous Counter



Synchronous Counter with Random Sequence



- Example: Design a counter with a sequence 4,7,3,0,2,4,……. Using JK FlipFlop?

Synchronous Counter with Random Sequence

3 bits = 3 F/F·

$\underline{1}\ \underline{1}\ \underline{1}$

- Example: Design a counter with a sequence 4,7,3,0,2,4,……. Using JK FlipFlop?

1) State Table
2) K-maps:
3) Design.

Unused state → 1,5,6.
Used state → ✓

State Table :

$Q_2 Q_2^+$ $Q_1 Q_1^+$ → $Q_0 Q_0^+$

	Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	J_2	K_2	J_1	K_1	J_0	K_0
4	1	0	0	1	1	1	X	0	1	X	1	X
7	1	1	1	0	1	1	X	1	X	0	X	0
3	0	1	1	0	0	0	0	X	X	1	X	1
0	0	0	0	0	1	0	0	X	1	X	0	X
2	0	1	0	1	0	0	1	X	X	1	0	X

K-map :

	$\overline{Q_1}\,\overline{Q_0}$	$\overline{Q_1}Q_0$	Q_1Q_0	$Q_1\overline{Q_0}$
$\overline{Q_2}$	^{0}	X^{1}	^{3}	1^{2}
Q_2	X^{4}	X^{5}	X^{7}	X^{6}

$J_2 = Q_1\overline{Q_0}$

Yourself!

E.T

		J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

QUICK QUIZ (POLL)

The counter with sequence 0,3,1,7,4, can be designed with

a) Three bit ripple counter
b) Two bit synchronous counter
c) Three bit synchronous counter
d) All of the above

Practice Example

- Design a MOD-5, 3-bit synchronous counter to count in the following sequence: 2, 3, 5, 1, 7. To modify the design and minimize the additional gates required for the J and K inputs, route the unused states to their next natural count state.
- In other words, route 0 to 1, 4 to 5, and 6 to 7.