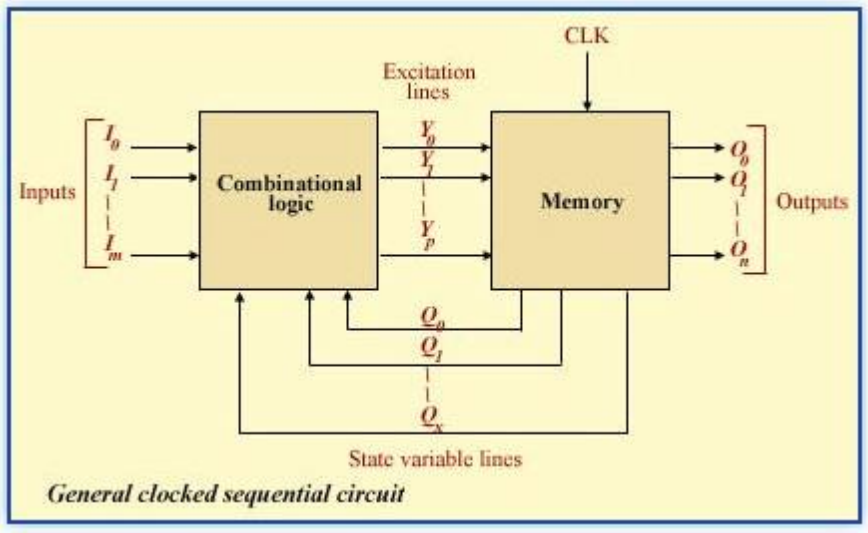
Design of Synchronous Counters[1]

V1.1 (2021)

This section begins our study of designing an important class of clocked sequential logic circuits-synchronous finite-state machines. Like all sequential circuits, a finite-state machine determines its outputs and its next state from its current inputs and current state. A synchronous finite-state machine changes state only on the clocking event.



Due to time, we will not be covering finite state machine inputs, but these would be used to control the state machine. Examples would be count up / count down inputs, hold, change of sequence, etc.

Counter Design Procedure

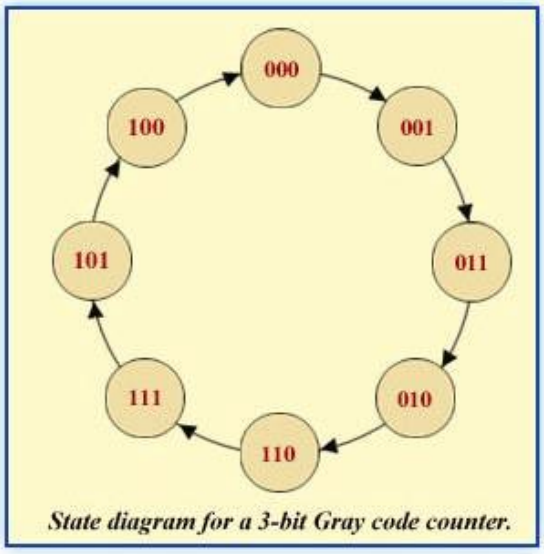
- Describe a general sequential circuit in terms of its basic parts and its input and outputs.
- Develop a state diagram for a given sequence.
- Develop a next-state table for a specific counter sequence.
- Create a FF transition table.
- Use K-map to derive the logic equations.
- Implement a counter to produce a specified sequence of states.

[1] Floyd Digital Logic Chp 8

Design Example #1: 3-bit Gray code counter

Step 1: State Diagram

State Diagram for a 3-bit Gray code counter:



State diagram for a 3-bit Gray code counter.

Step 2: Next-State Table

Next state table for a 3-bit Gray code counter

Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Step 3: Flip-Flop Transition Table

Transition table for a J-K Flip-Flop

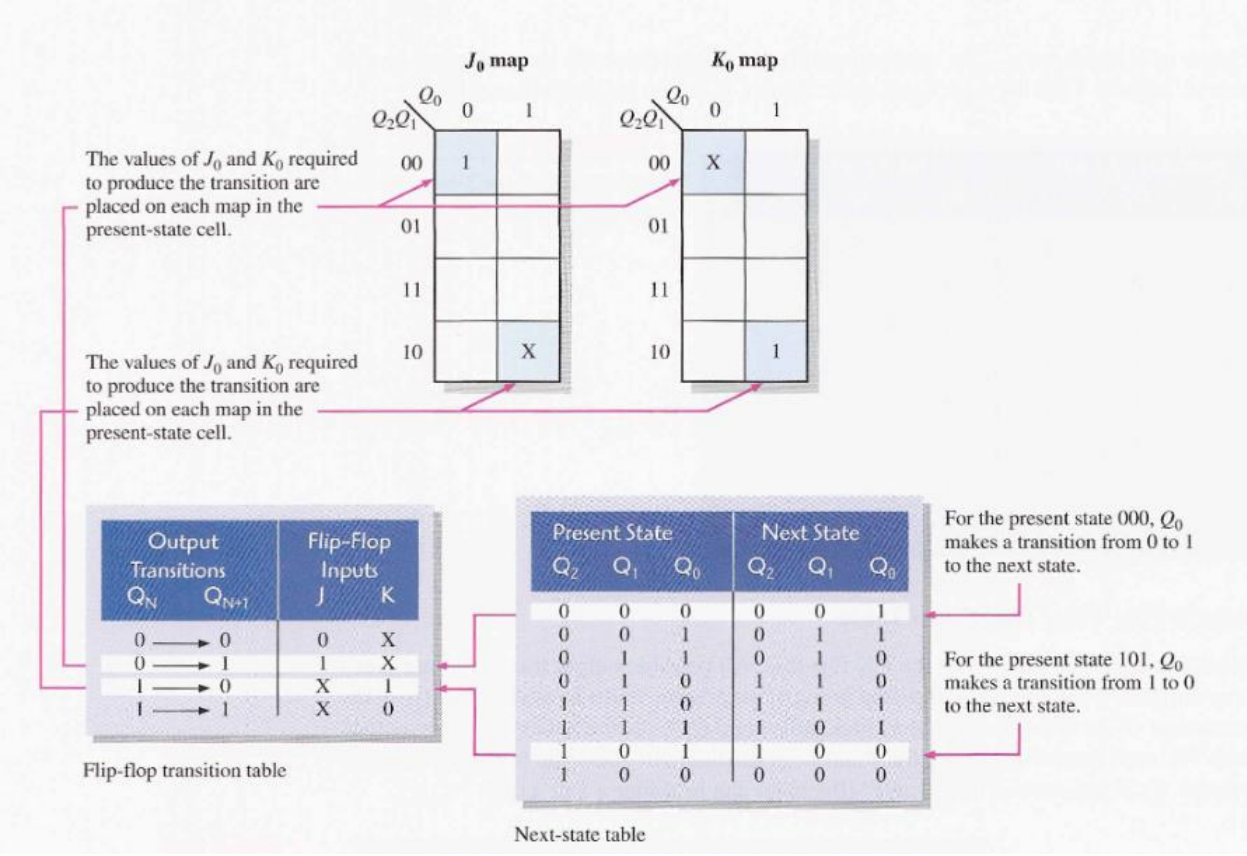
Output Transitions Q_N		Q_{N+1}	Flip-Flop Inputs J	K
0	→	0	0	X
0	→	1	1	X
1	→	0	X	1
1	→	1	X	0

Q_N: present state, Q_{N+1}: next state

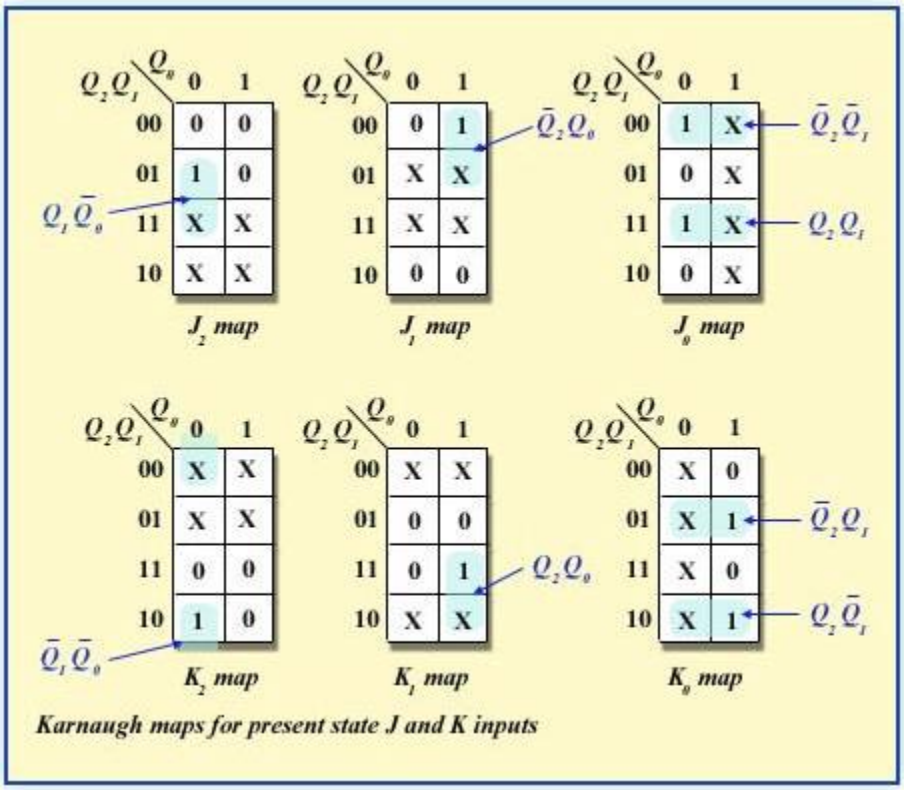
X: "don't care"

Step 4: Karnaugh Maps

The following diagram shows the steps to create separate next states of separate J and K from the current states of J and K.



Karnaugh maps for present-state J and K inputs for the 3-bit Gray code counter.



Karnaugh maps for present state J and K inputs

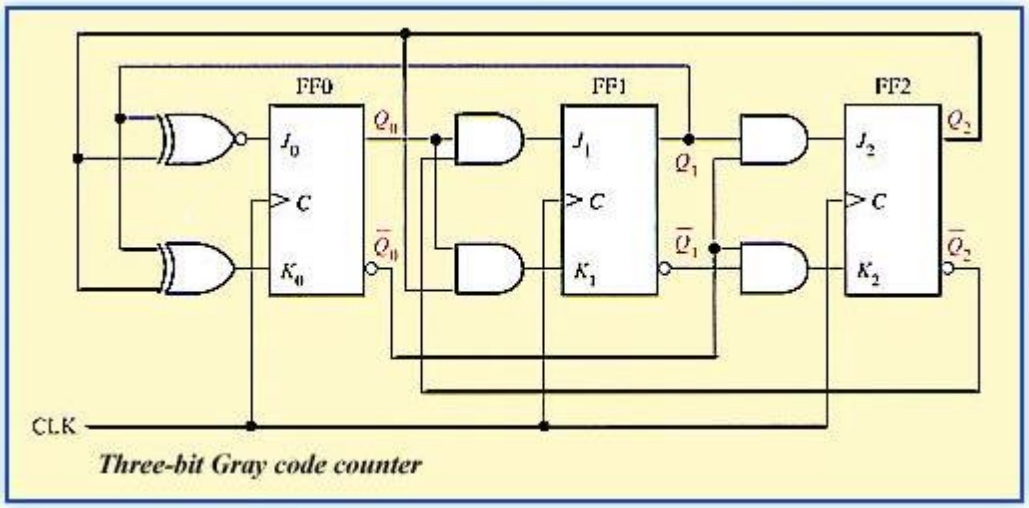
Step 5: Logic Expressions for Flip-flop Inputs

The next-state J and K outputs for a 3-bit Gray code counter.

$$J_0 = Q_2Q_1 + \bar{Q}_2Q_1 = \overline{\bar{Q}_2 + \bar{Q}_1}$$
$$K_0 = Q_2\bar{Q}_1 - \bar{Q}_2Q_1 = Q_2 + Q_1$$
$$J_1 = \bar{Q}_2Q_0$$
$$K_1 = Q_2Q_0$$
$$J_2 = Q_1\bar{Q}_0$$
$$K_2 = \bar{Q}_1\bar{Q}_0$$

Step 6: Counter Implementation

The hardware diagram of the 3-bit Gray code counter



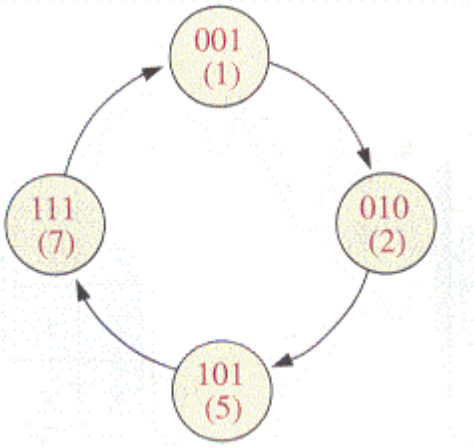
Three-bit Gray code counter

- There are many more examples for the design of synchronous counter. These can be found in any digital network related textbooks.

Design Example #2

Design a counter with the irregular binary count sequence shown in the state diagram of Figure 4.1.

Step 1: State Diagram



Step 2: Next-State Table

Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	0	0	1

Step 3: Flip-Flop Transition Table

Transition table for a J-K Flip-Flop

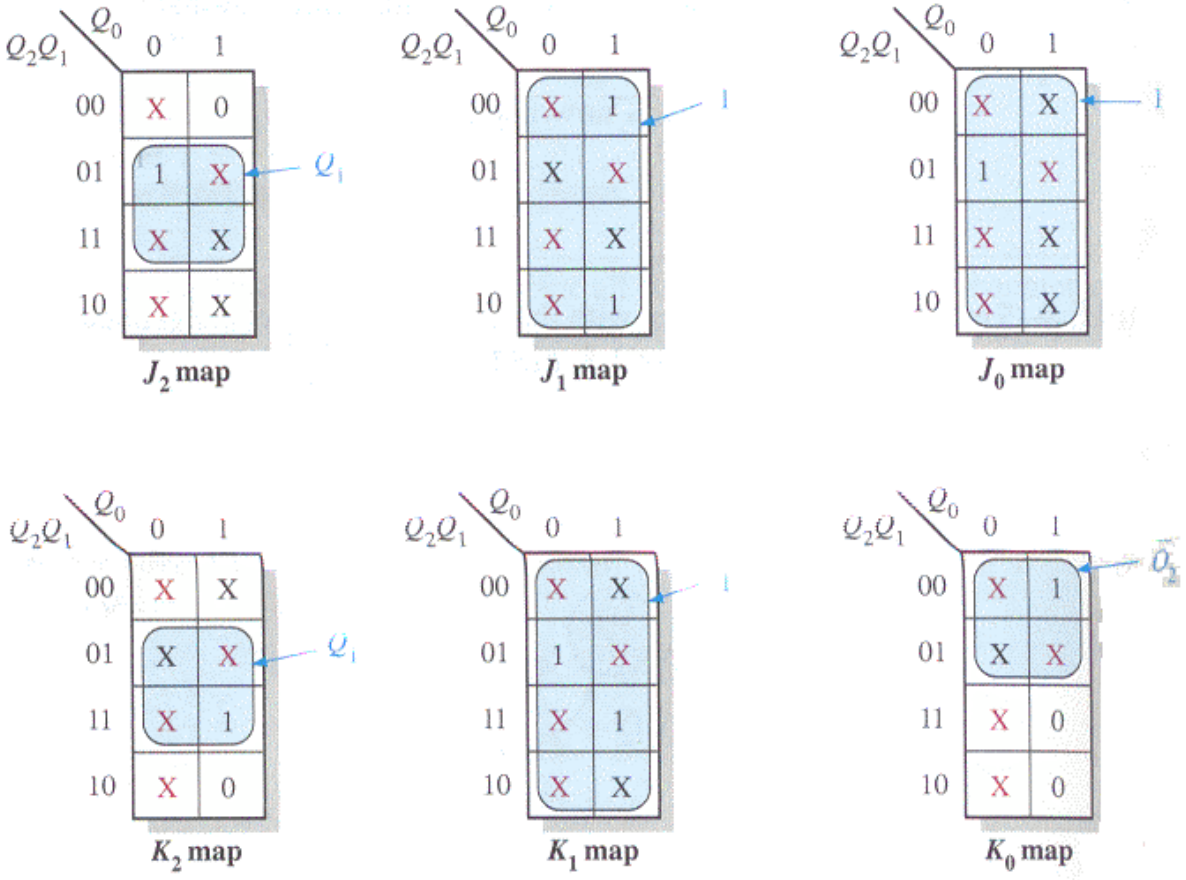
Output Transitions Q_n		Q_{n+1}	Flip-Flop Inputs J	K
0	→	0	0	X
0	→	1	1	X
1	→	0	X	1
1	→	1	X	0

Q_n: present state, Q_{n+1}: next state

X : "don't care"

Step 4: Karnaugh Maps

Note: "don't cares" can be placed in the cells corresponding to the invalid states of 000,011,100 and 110 (as indicated by red X)



Step 5: Logic Expressions for Flip-flop Inputs

The expression for each *J* and *K* input taken from the maps is as follows:
$J_0 = 1, K_0 = Q_2$
$J_1 = K_1 = 1$
$J_2 = K_2 = Q_1$

Step 6: Counter Implementation