Lecture -2
Sequential Circuit Design: Counters

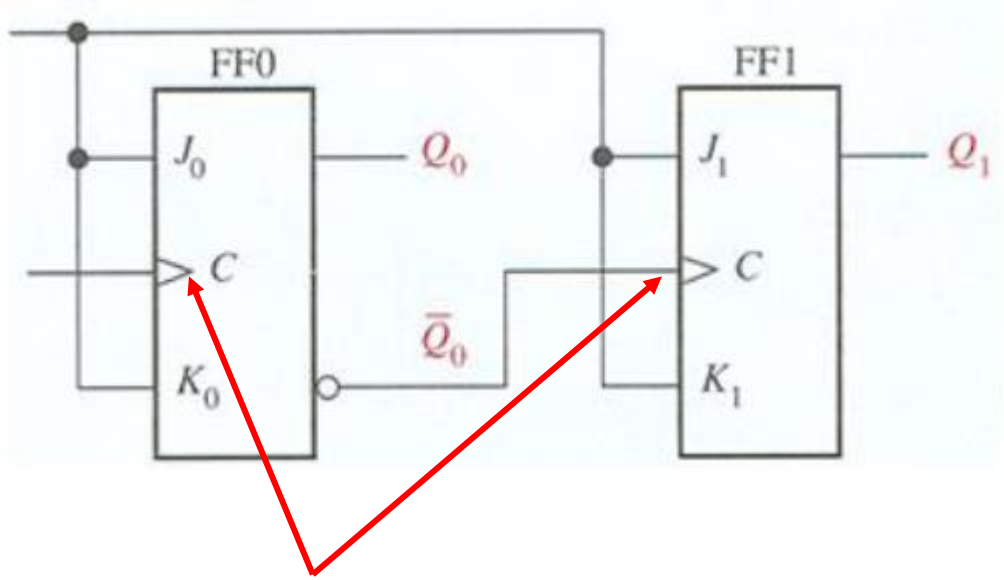
Prepared By: Asif Mahfuz

Asynchronous Counter

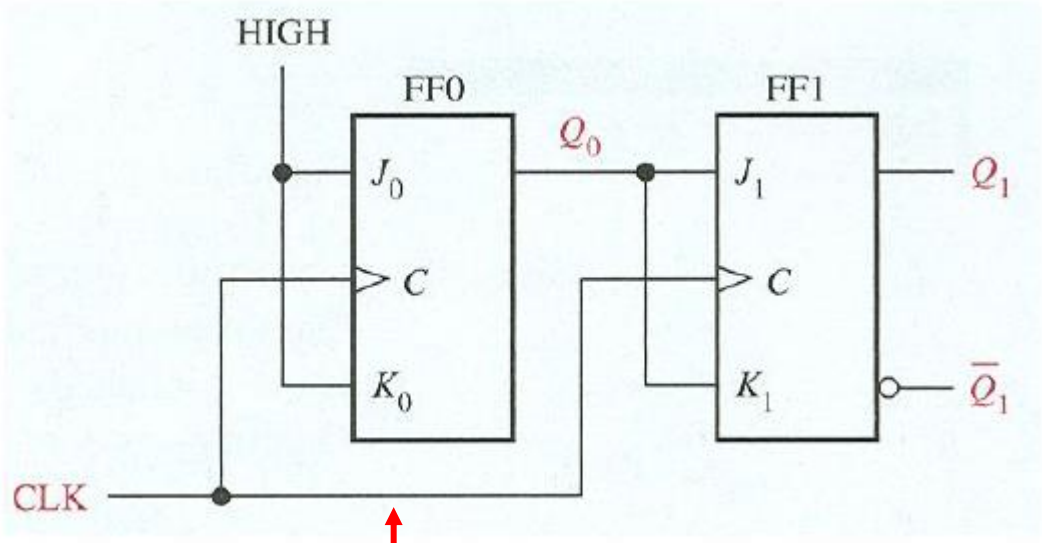
COUNTERS: Counters in digital circuit are devices used for counting. These are sequential circuits made of flip-flops. There are two broad categories of counters, namely,

Asynchronous Counters are counters where the flip-flops do not have a common clock and so they do not change states at the same time.

Synchronous Counters are counters where the flip-flops have a common clock and so the flip-flop changes its states at the same time.



The clock is not common



The clock is common

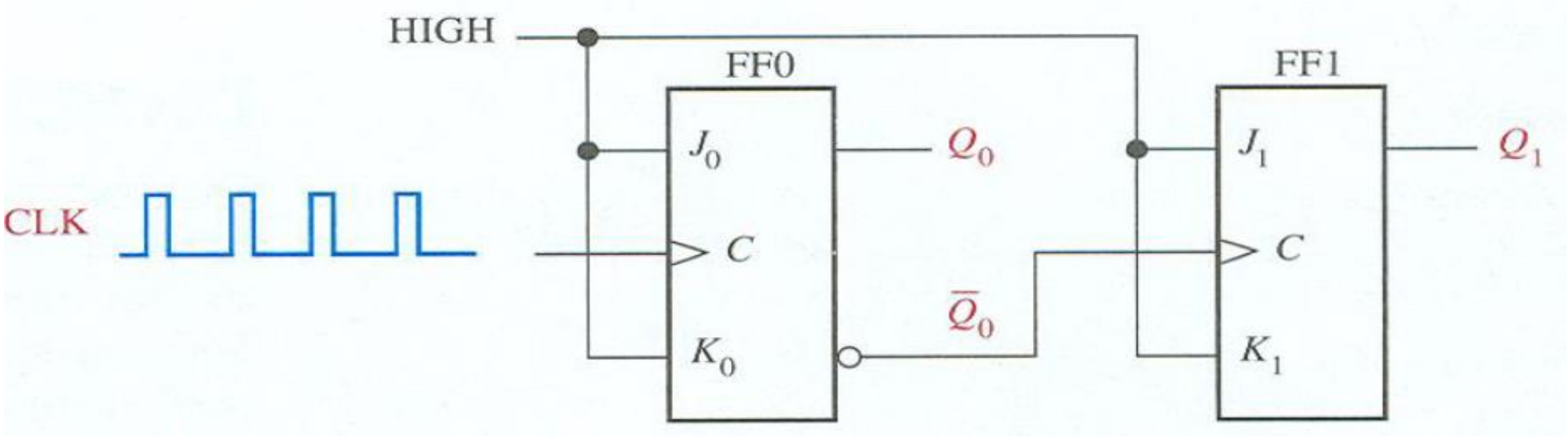
Asynchronous Counter

- Flip-Flops are not connected to a common clock.
- Flip-Flops do not change their state at the same time.
- JK flip-flops are used in toggle mode of operation.

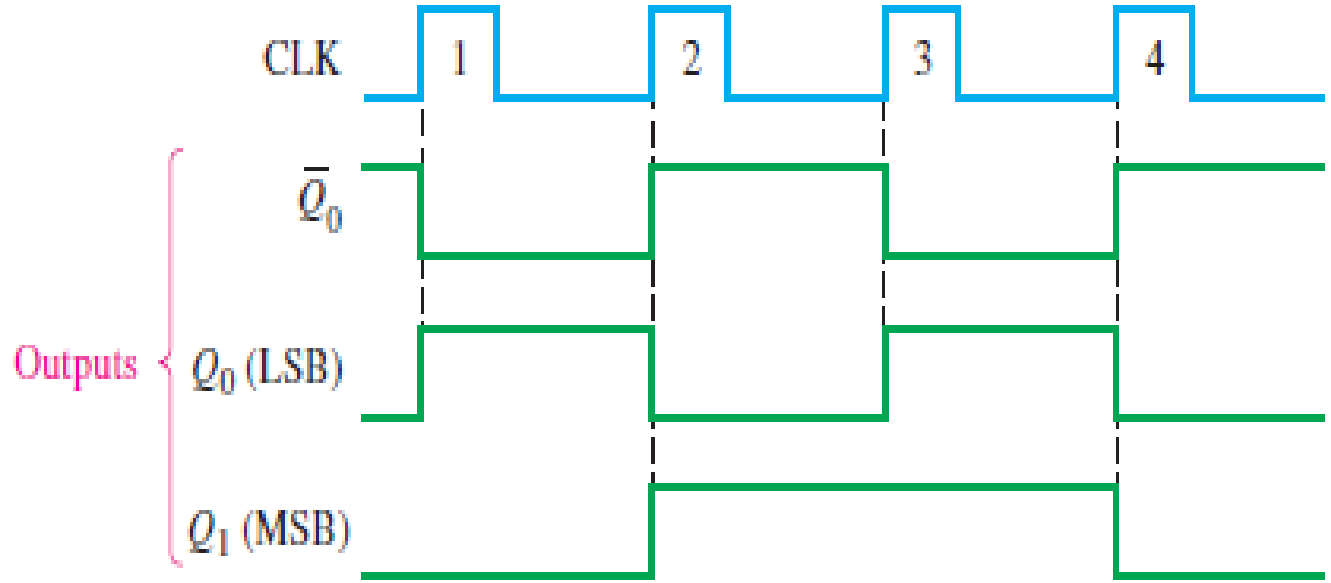
Common Asynchronous Counter

- Asynchronous Binary Counters
  - 2- Bit Binary Counters
  - 3- Bit Binary Counters
  - 4- Bit Binary Counters
- Asynchronous Decade Counter (MOD-10)
- Asynchronous Modulus Twelve Counter

2-Bit Binary Asynchronous Counter

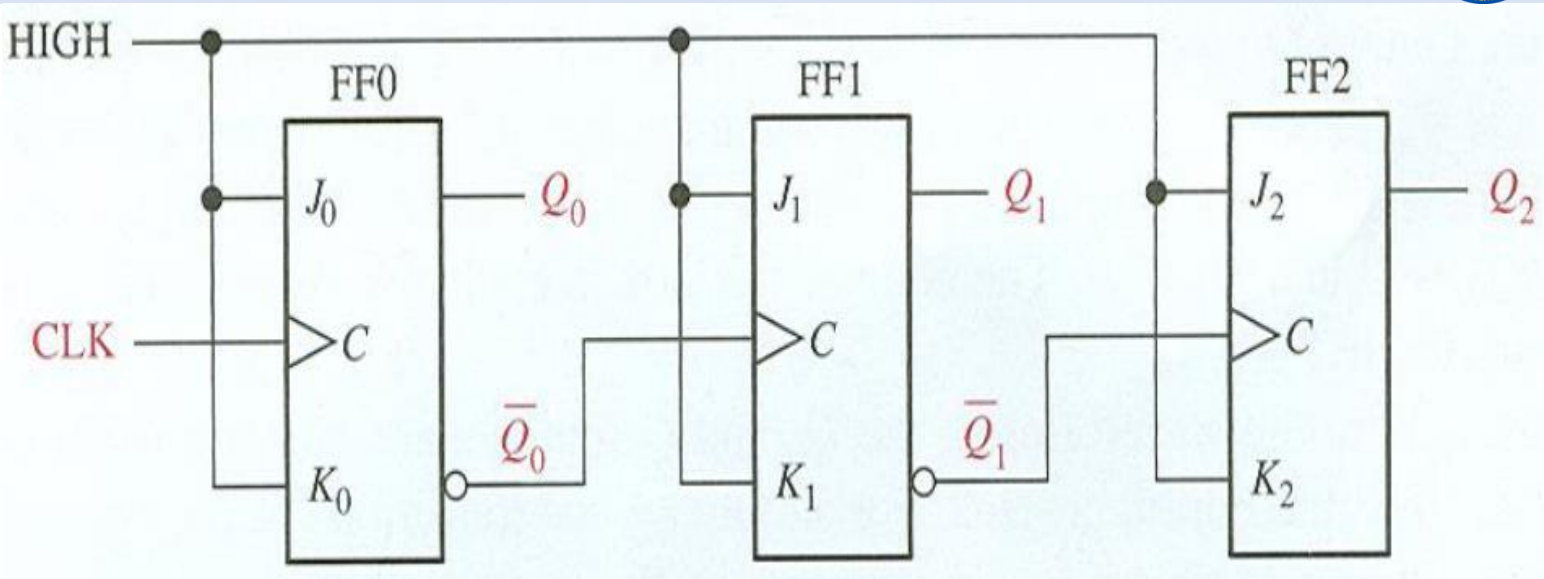


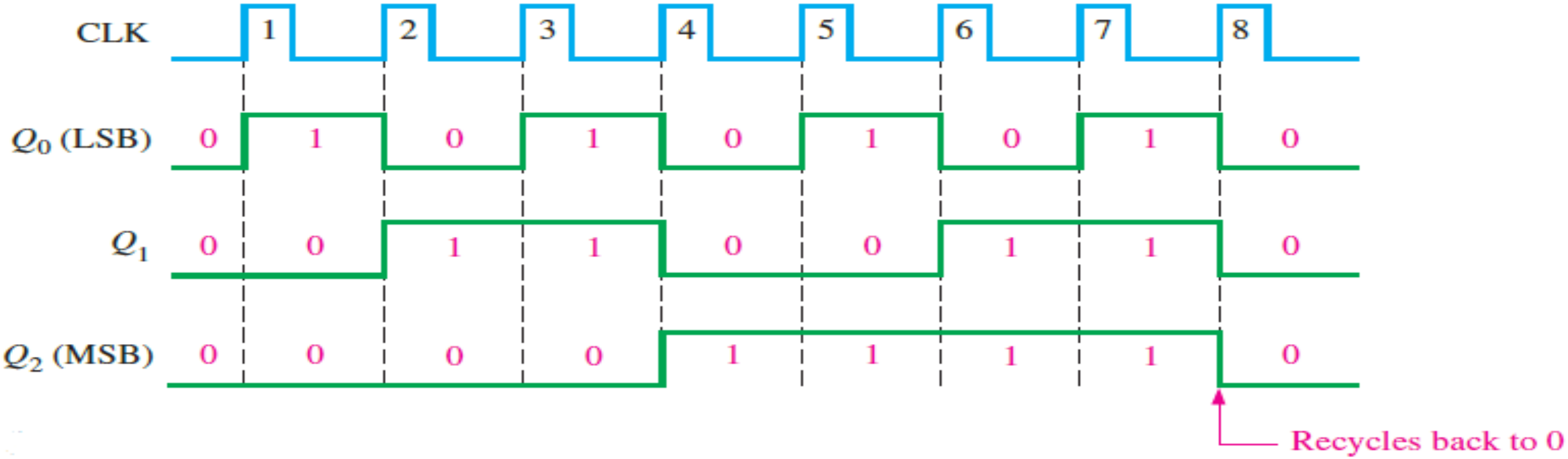
Clock Pulse	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0



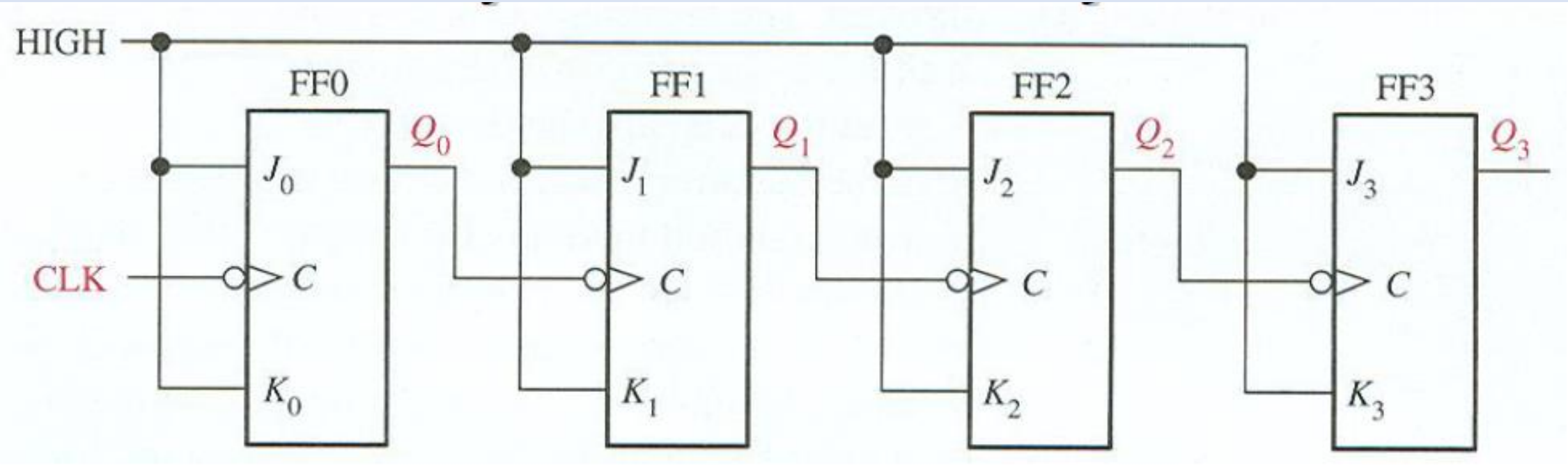
3-Bit Binary Asynchronous Counter

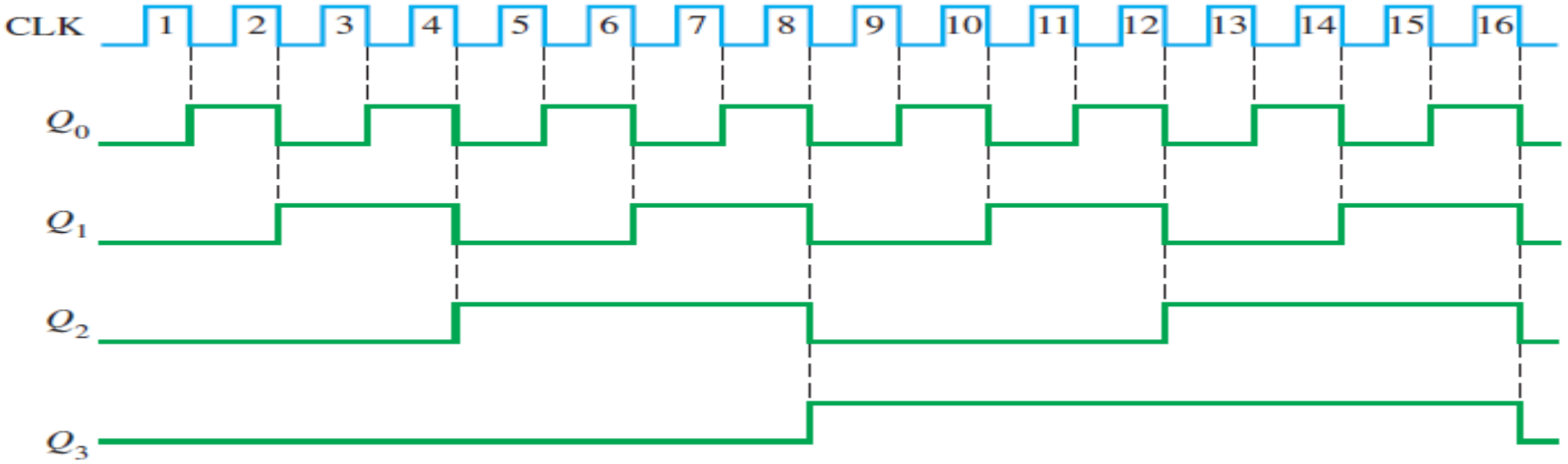
Clock Pulse	Q_2	Q_1	Q_0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (recycles)	0	0	0





4-Bit Binary Asynchronous Counter





MOD Counters

- A counter with N flip-flops can count to 2^N sequences.
- The modulus of a counter in number of unique sequence that a counter can count through.
- The MOD counters do not complete the entire count sequence.
- These type of sequence is called truncated sequence.
- An example of such counter can be MOD-11 counter.
- A MOD-11 counter has 4 flip-flops (as we require 4-bits to count up to 11)
- With 4 flip-flops the complete count sequence is 16 (0 to 15).
- However, a MOD-11 counter will have only 11 sequences (0 to 10).
- Thus we need to figure out a way to reset the flip-flop when the counter reaches the 12^{th} sequence (no. 11).
- This is where the asynchronous clear input becomes very handy.
- So we can decode the 12^{th} state to reset the counter.

Asynchronous Decade Counter (MOD-10)

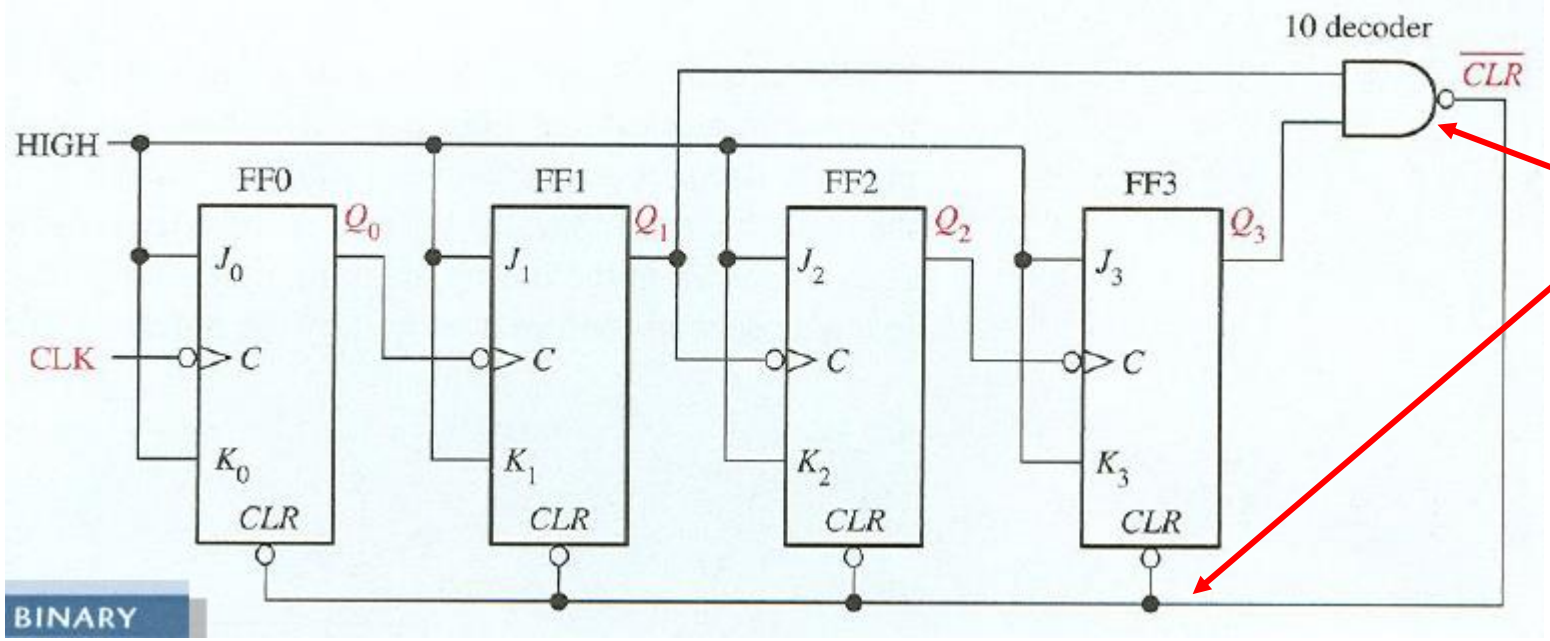
DECIMAL	BINARY
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Not wanted. We want 0000.
So Decode this state.

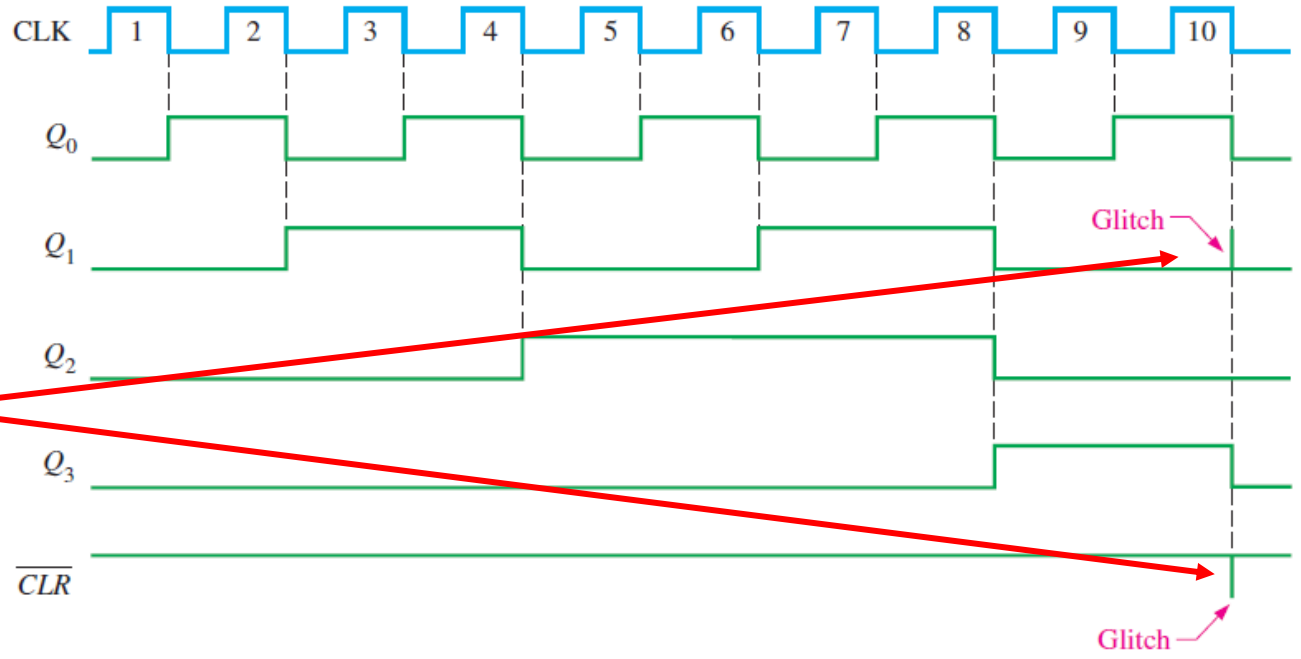
- The most popular MOD counter is the Decade Counter.
- Counters with 10 sequences are called Decade Counters.
- It can count 10 sequence that is 0 to 9.
- Thus it is also called a BCD counter.
- It has a wide range of application from displays to digital watches.
- The counter counts from 0000 to 1001.
- Then when counter reach 1010 the counter resets.
- So the state 1010 is decoded to reset the counter.

Since we do not want the states after 1001, at 1010 we force clear the counter and restart counting from 0000.

Asynchronous Decade Counter (MOD-10)



To decode 1010, we can only use the outputs Q_3 and Q_1. NAND of Q_1 and Q_3 produces a 0 when both Q_1 and Q_3 are 1. And the output of the NAND gate is used to clear the flip-flops



As the counter changes its state from 1001 to 1010(for fraction of seconds), the counter is reset by the help of clear. This produces the glitch

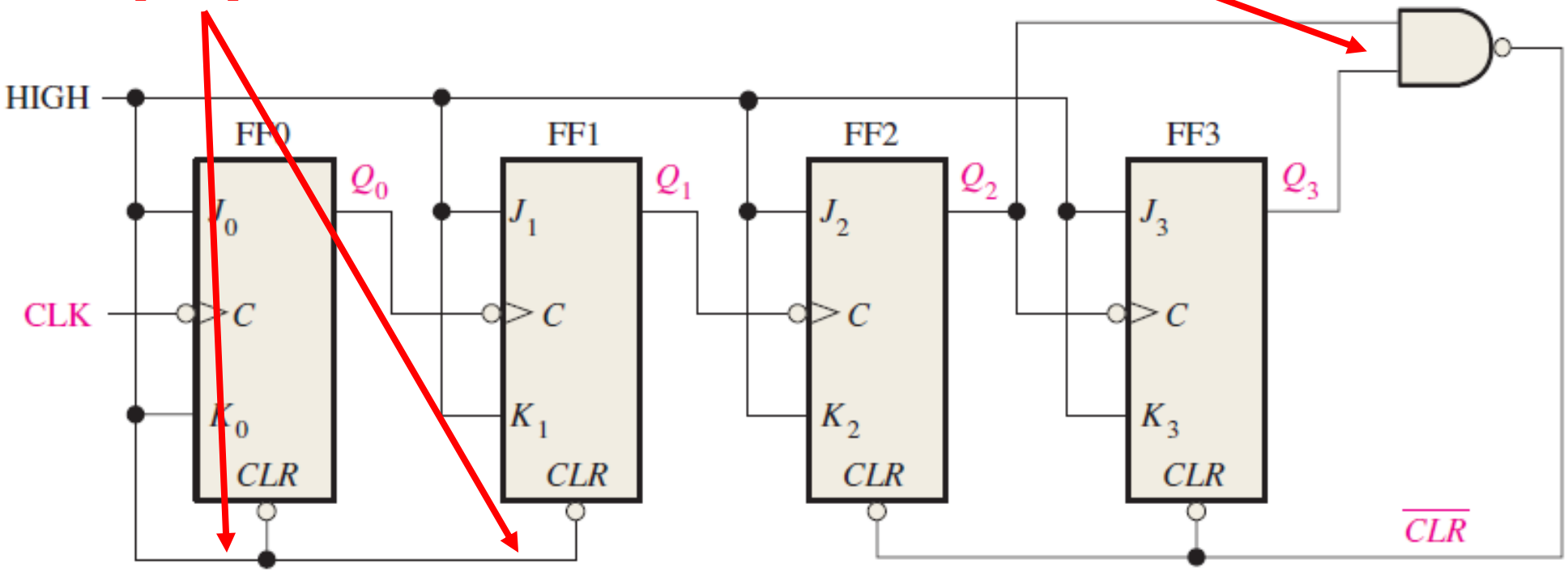
Asynchronous MOD-12 Counter

At state 0000 and 1100 Q_0 and Q_1 are already 0 so need to clear these two flip-flops.

NAND of Q2 and Q3 at 1100 state produces a 0 which clears FF2 and FF3.

Q_3	Q_2	Q_1	Q_0	
0	0	0	0	← Recycles
.	.	.	.	
.	.	.	.	
.	.	.	.	
1	0	1	1	← Recycles
1	1	0	0	← Normal next state

12 decoder



Synchronous Counter

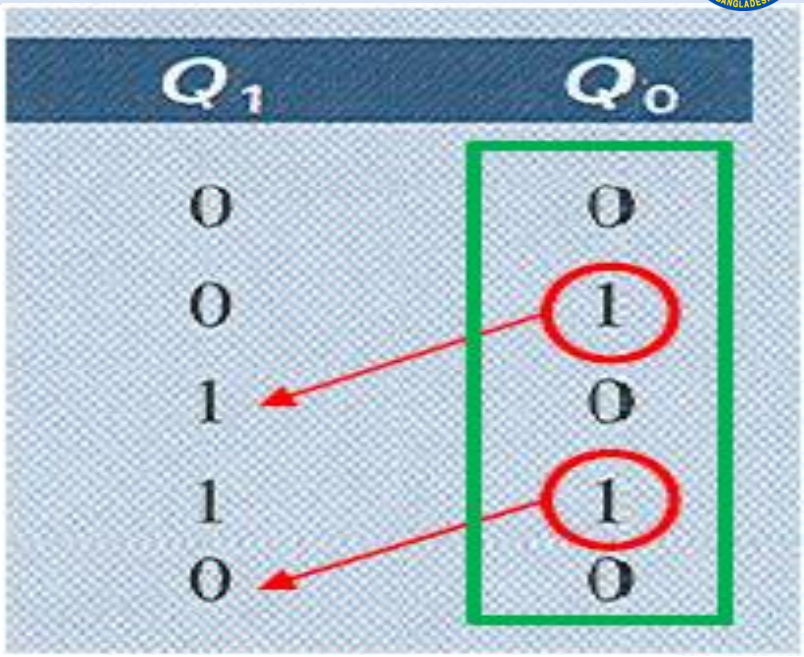
- Flip-Flops are connected to a common clock.
- Flip-Flops change their state at the same time.
- JK flip-flops are used.
- The design is complex compared to asynchronous counters.

Common Synchronous Counter

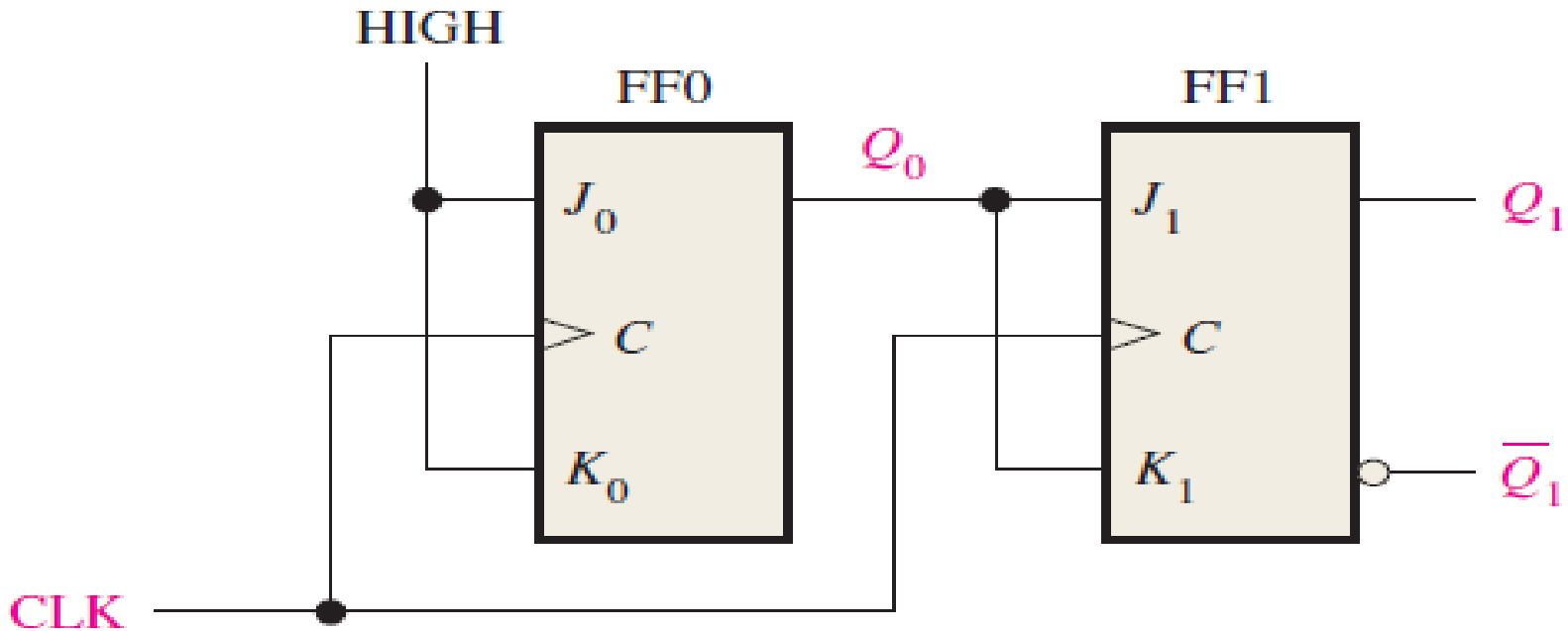
- Synchronous Binary Counters
  - 2- Bit Binary Counter
  - 3- Bit Binary Counter
  - 4- Bit Binary Counter
- Synchronous BCD Decade Counters
- Up/ Down Synchronous Counters
- Irregular Sequence Counters

2-Bit Synchronous Counter

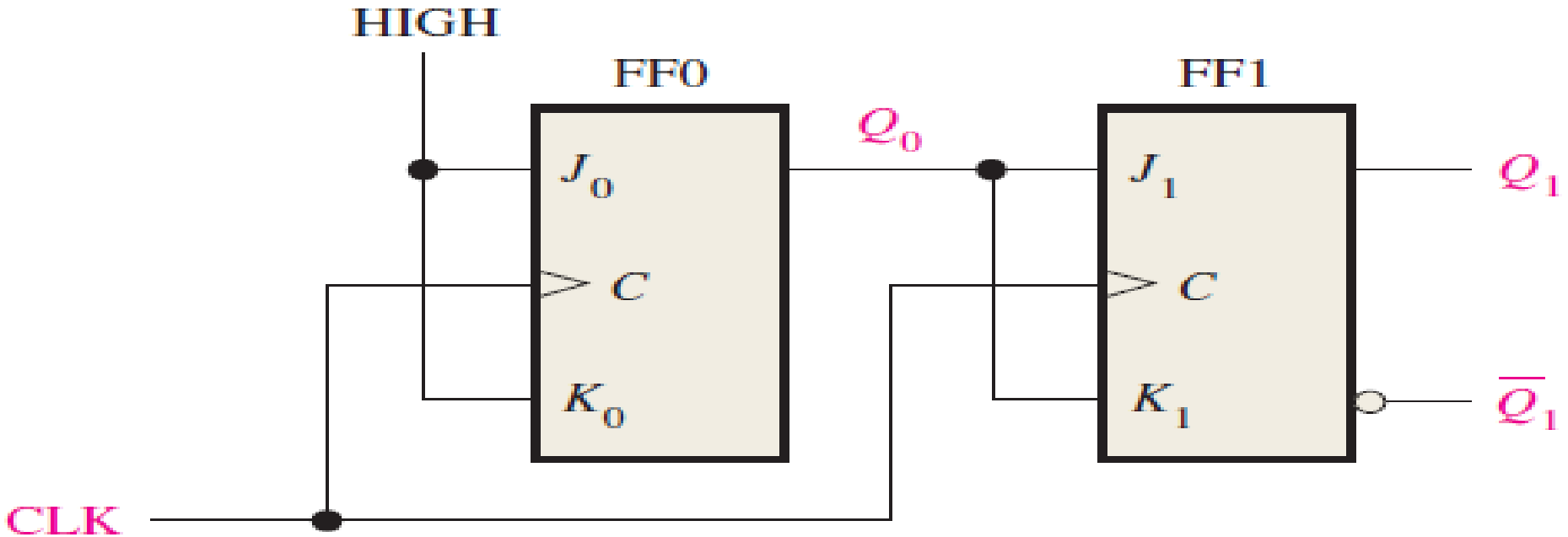
- 2-Bit Binary Counter count from 0 up to 3 and then resets.
- Unlike asynchronous counter we need find the inputs to the flip-flop.
- The first bit alternates at every clock
- So the first flip-flop can be used in toggle mode.
- But for the second bit we need to find the pattern that toggles the bit.



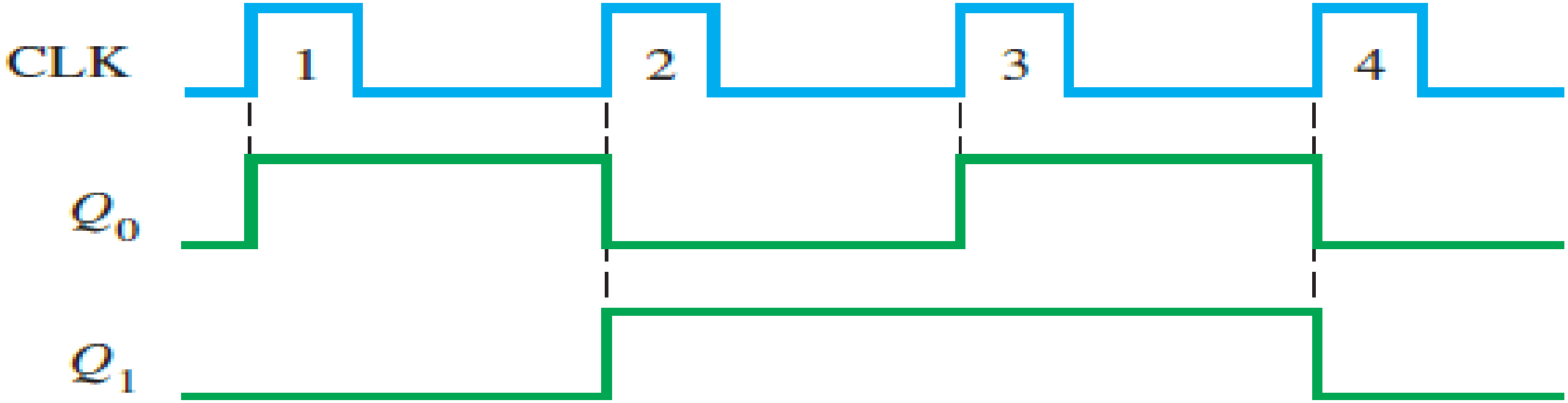
$$J_0 = K_0 = 1$$
$$J_1 = K_1 = Q_0$$



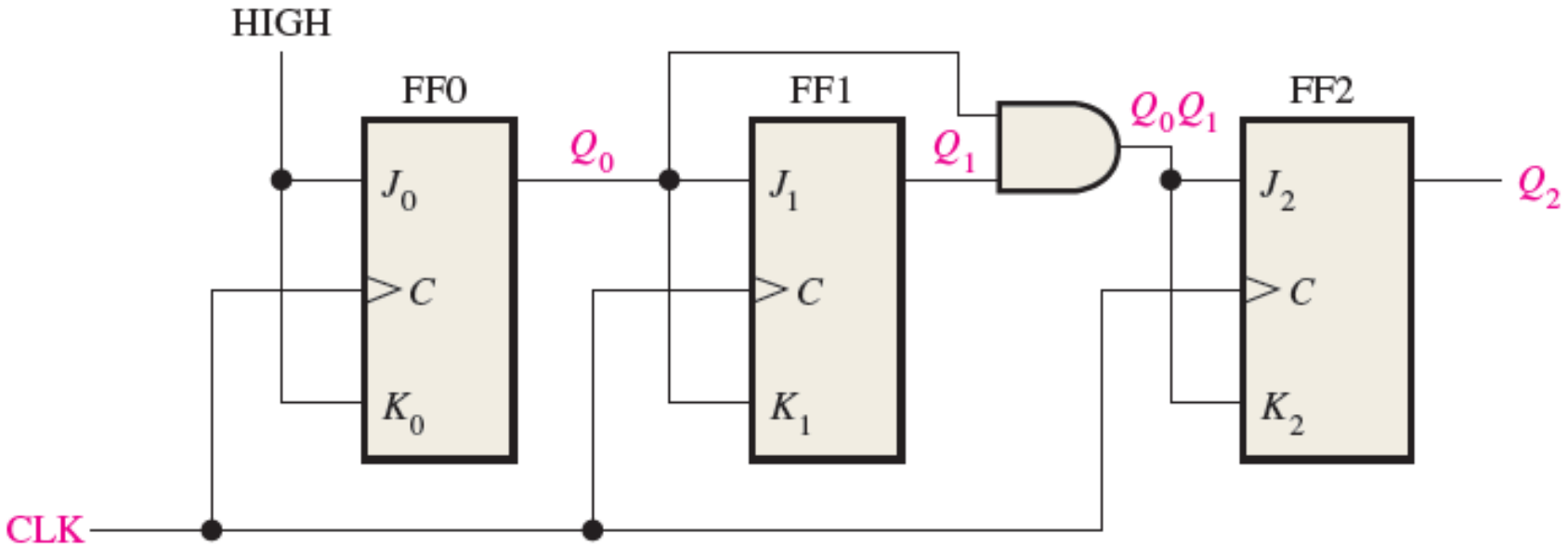
2-Bit Synchronous Counter



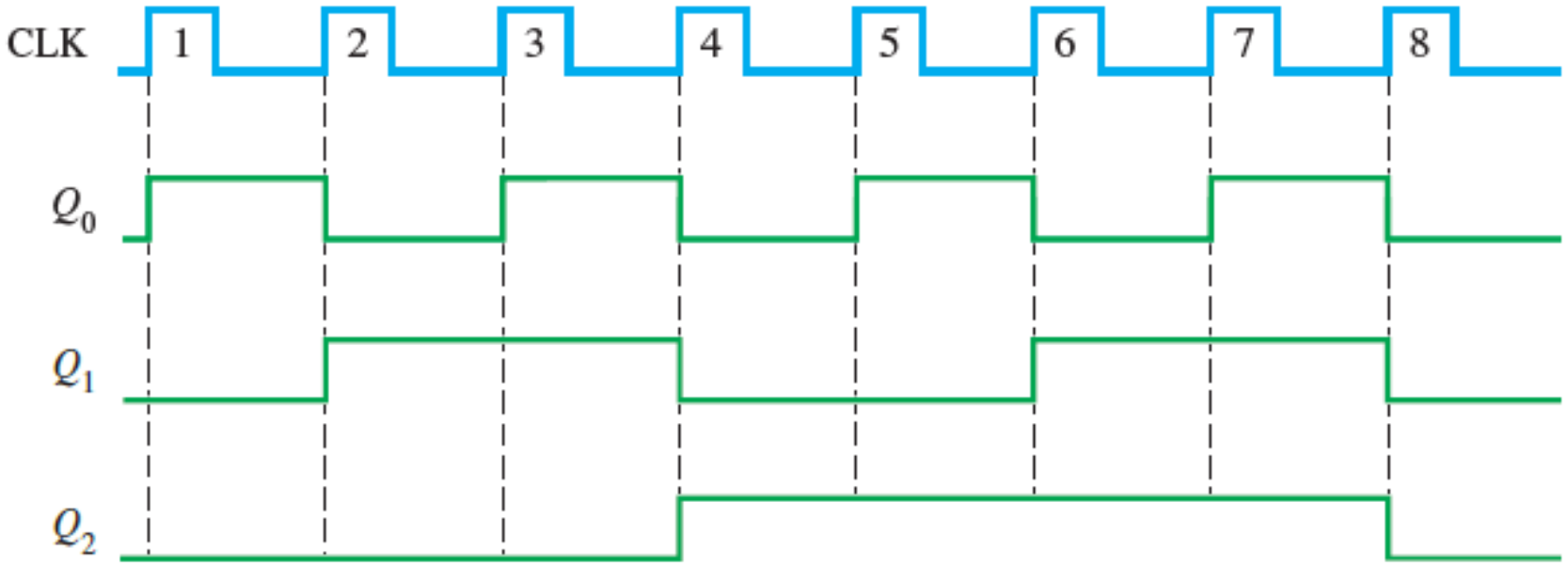
$$J_0 = K_0 = 1$$
$$J_1 = K_1 = Q_0$$



3-Bit Synchronous Counter



CLOCK PULSE	Q_2	Q_1	Q_0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (recycles)	0	0	0



$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_0Q_1$$

4-Bit Synchronous Counter

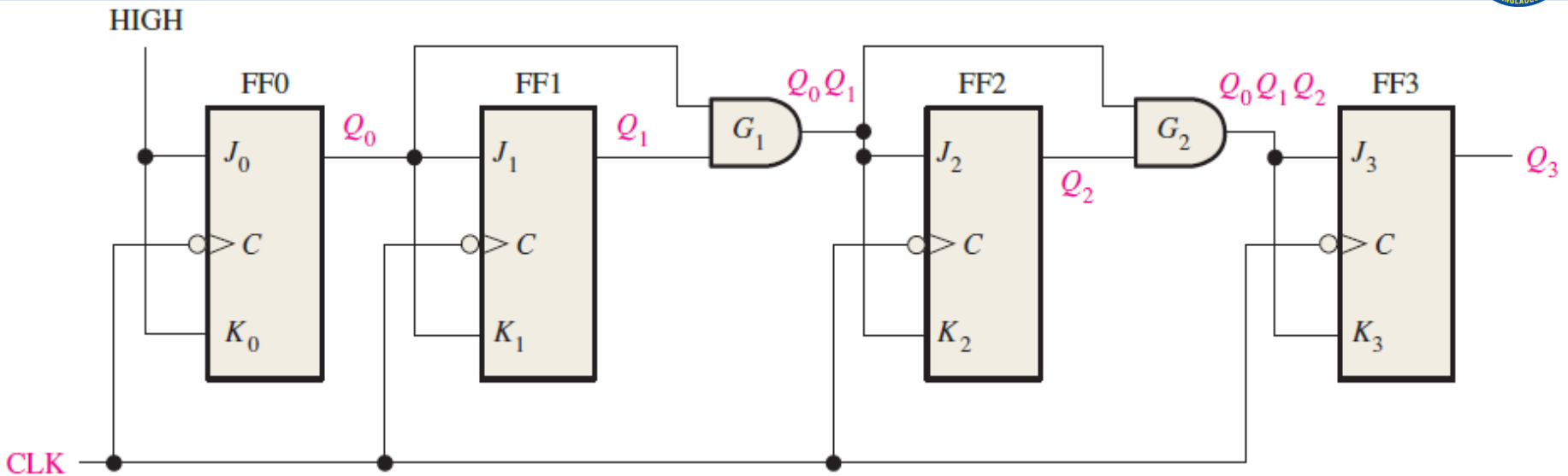
DECIMAL	BINARY
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
0	0000

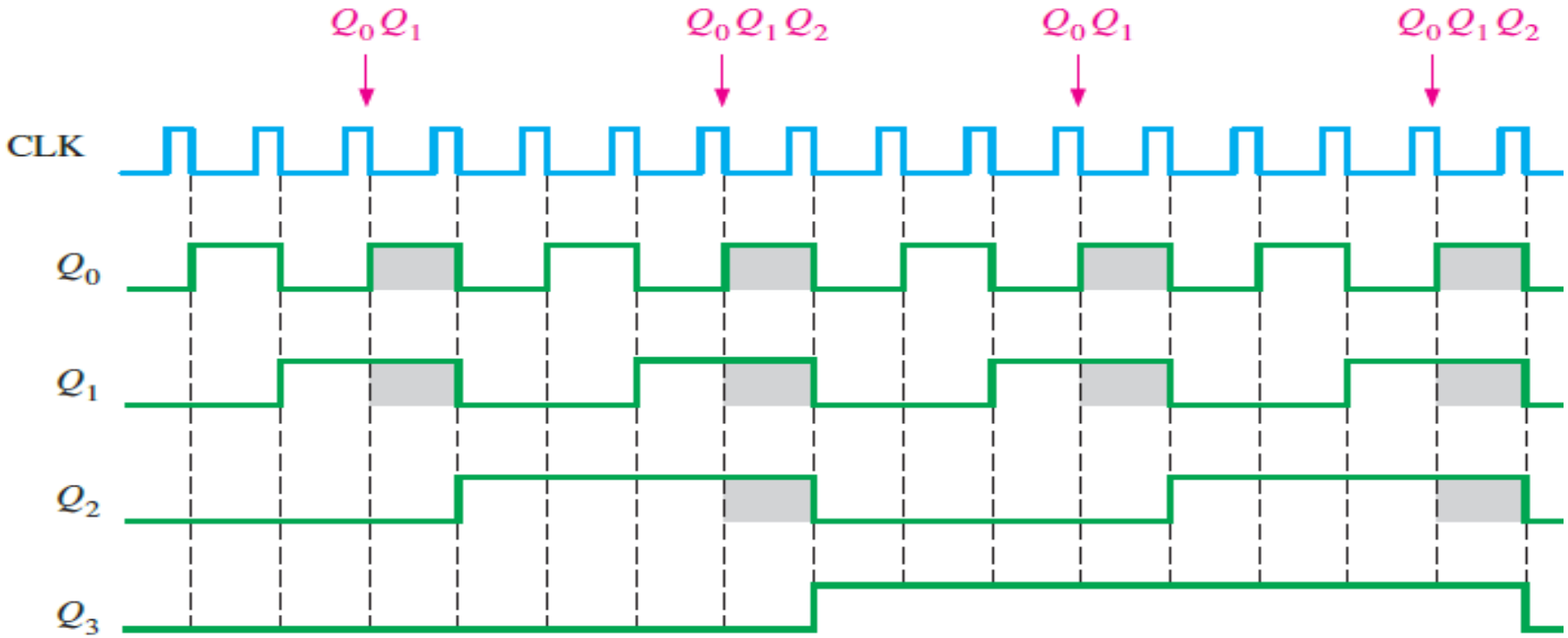
$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_0Q_1$$

$$J_3 = K_3 = Q_0Q_1Q_2 + Q_0Q_3$$





Synchronous BCD Decade Counter

States of a BCD decade counter.

Clock Pulse	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 (recycles)	0	0	0	0

Why do we need $\overline{Q_3}$??

$$J_0 = K_0 = 1$$

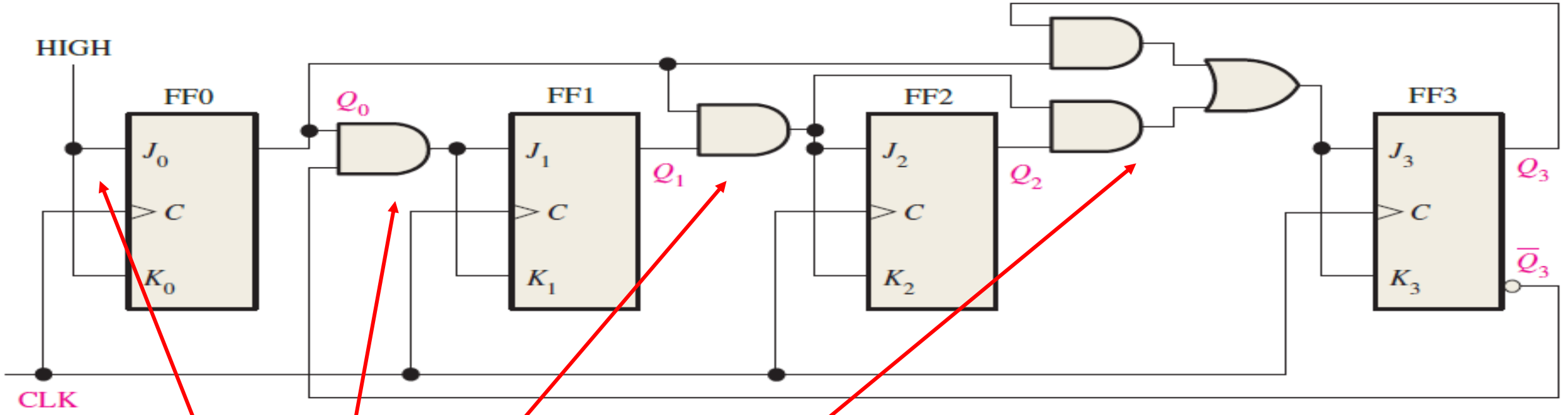
$$J_1 = K_1 = Q_0\overline{Q_3}$$

$$J_2 = K_2 = Q_0Q_1$$

$$J_3 = K_3 = Q_0Q_1Q_2 + Q_0Q_3$$

Why do we need

Synchronous BCD Decade Counter

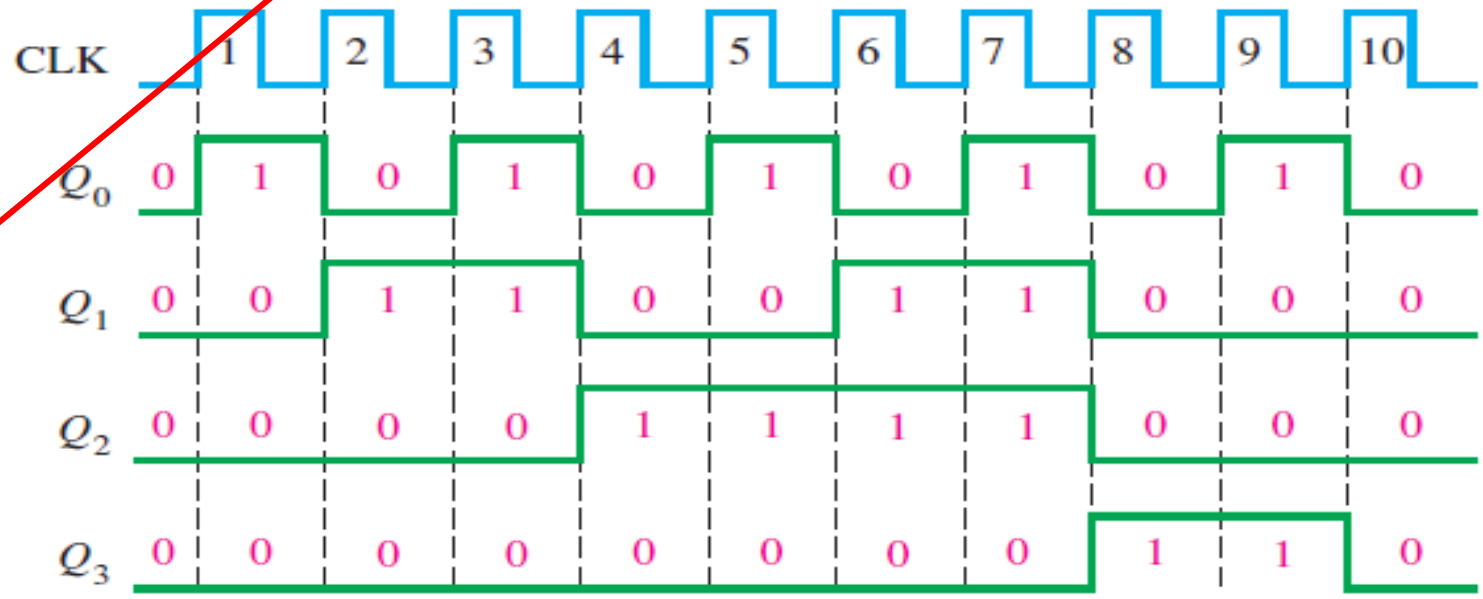


$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0\overline{Q}_3$$

$$J_2 = K_2 = Q_0Q_1$$

$$J_3 = K_3 = Q_0Q_1Q_2 + Q_0Q_3$$



Up-Down Synchronous Counter

- An up/down counter is one that is capable of progressing in either direction through a certain sequence.
- It is also called a bi-directional counter.
- In general most up/down can be reversed at any point in their sequence

Up/Down sequence for a 3-bit binary counter.

Clock Pulse	Up	Q_2	Q_1	Q_0	Down
0		0	0	0	
1		0	0	1	
2		0	1	0	
3		0	1	1	
4		1	0	0	
5		1	0	1	
6		1	1	0	
7		1	1	1	

UP UP

0, 1, 2, 3, 4, 5, 4, 3, 2, 3, 4, 5, 6, 7, 6, 5, etc.

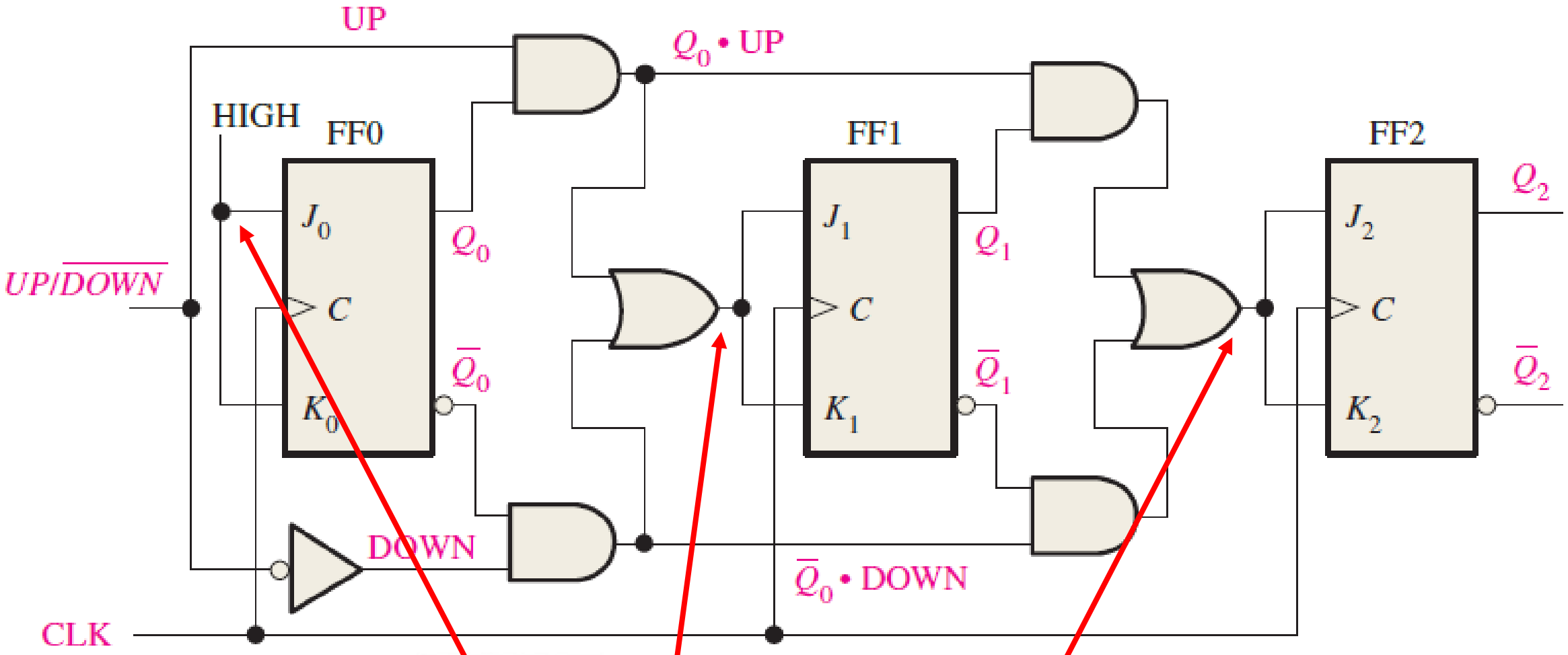
DOWN DOWN

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = (Q_0 \cdot \text{UP}) + (\overline{Q}_0 \cdot \text{DOWN})$$

$$J_2 = K_2 = (Q_0 \cdot Q_1 \cdot \text{UP}) + (\overline{Q}_0 \cdot \overline{Q}_1 \cdot \text{DOWN})$$

Up-Down Synchronous Counter



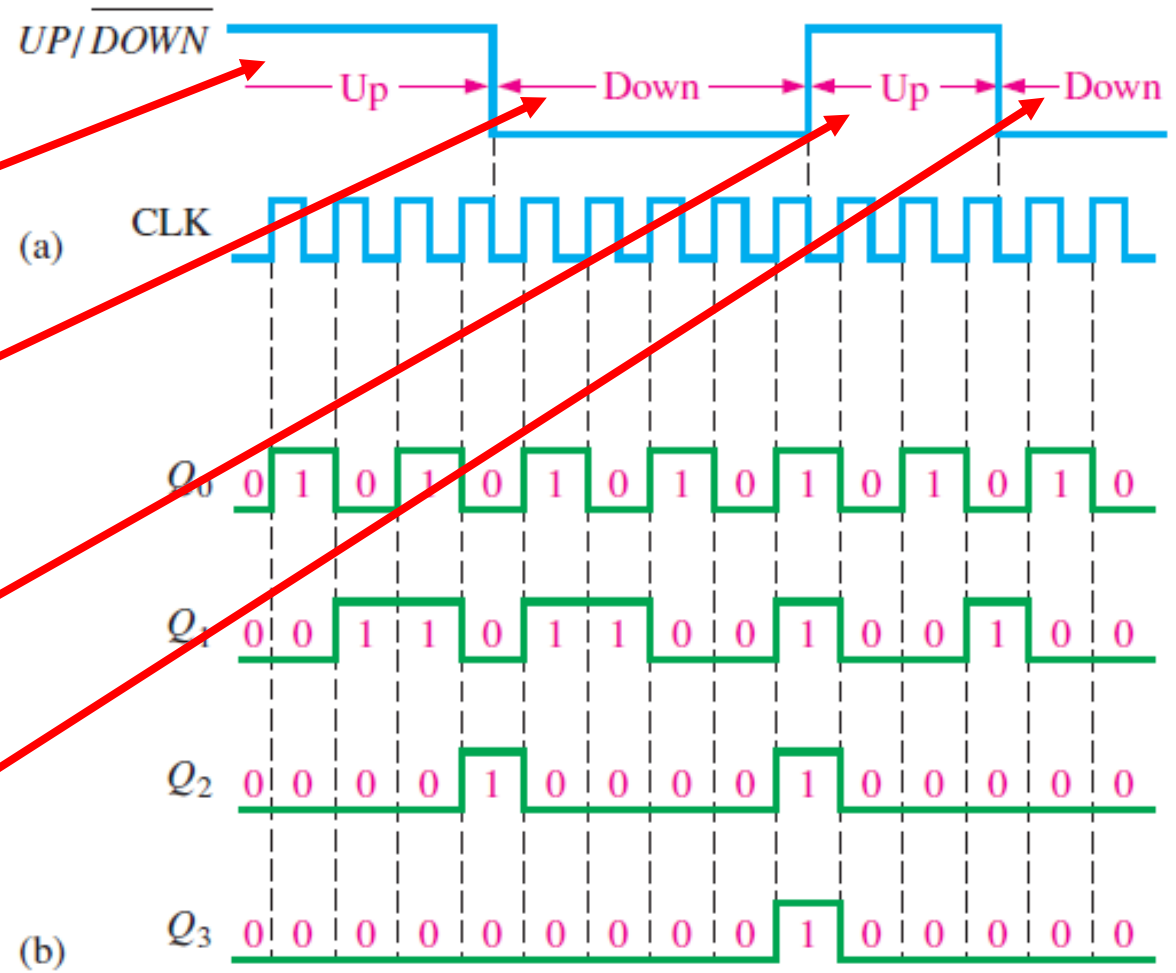
$$J_0 = K_0 = 1$$

$$J_1 = K_1 = (Q_0 \cdot \text{UP}) + (\overline{Q}_0 \cdot \text{DOWN})$$

$$J_2 = K_2 = (Q_0 \cdot Q_1 \cdot \text{UP}) + (\overline{Q}_0 \cdot \overline{Q}_1 \cdot \text{DOWN})$$

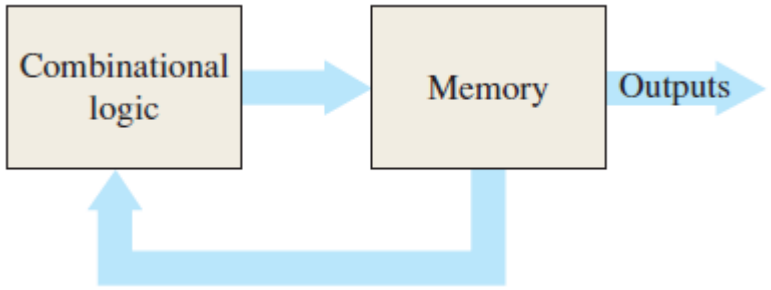
Up-Down Synchronous Counter

Q_3	Q_2	Q_1	Q_0	
0	0	0	0	UP
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	0	1	1	DOWN
0	0	1	0	
0	0	0	1	
0	0	0	0	
1	1	1	1	
0	0	0	0	UP
0	0	0	1	
0	0	1	0	
0	0	0	1	DOWN
0	0	0	0	

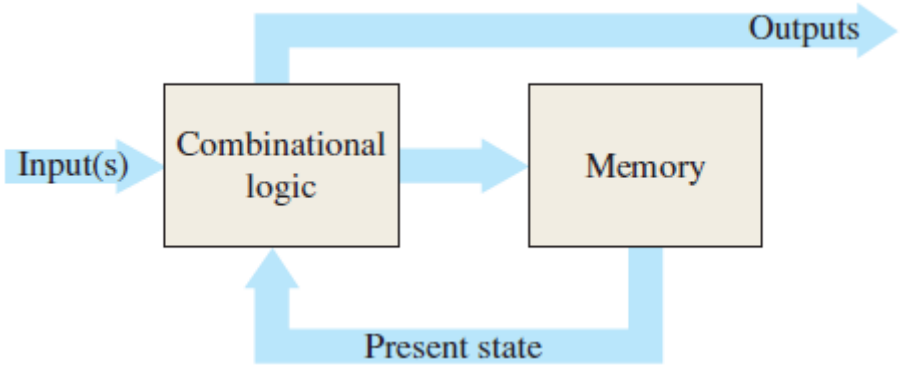


Irregular Sequence Counter

To design an irregular sequence counter we need to first understand the concept of state machines. A state machine is a sequential circuit having a finite number of states occurring in prescribed order. There are generally two types of state machines, namely, Moore State Machine and Mealy State Machines.



Moore Machine

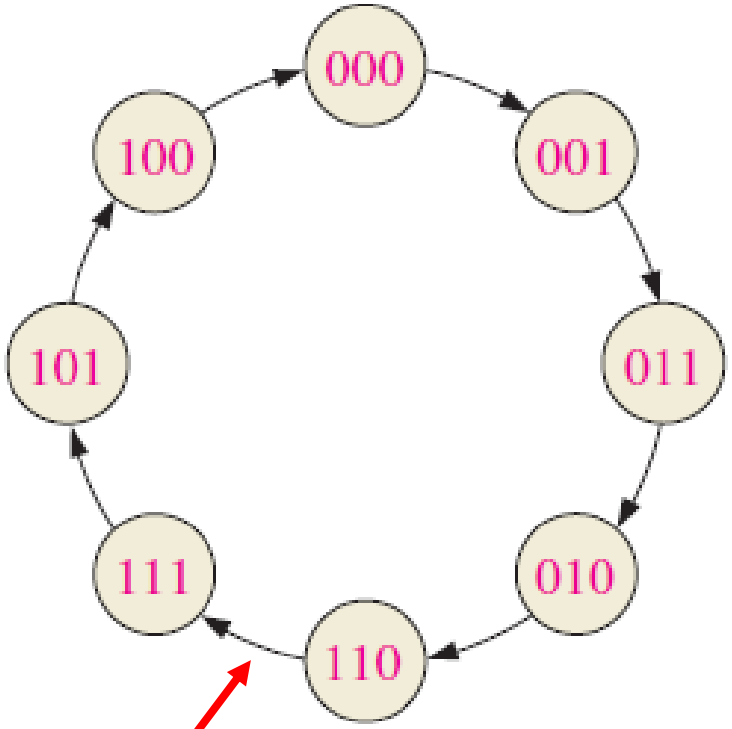


Mealy Machine

- State Machines have a finite number of state and they go about it in a prescribed manner.
- The order of sequence is described with the help of a state diagram.
- A state diagram is a diagram which shows the progression of states through which the counter advances when it is clocked.

Irregular Sequence Counter (Gray Code Counter) Design

Step1: Develop a state Diagram
Step2: Develop a Next-state table.
Step3: Create a flip-flop transition table.
Step4: Use Karnaugh-map to derive the logic requirements.
Step5: Implement a counter to produce the specific sequence of states.



STEP 1

Next-state table for 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 2

Irregular Sequence Counter (Gray Code Counter) Design

Transition table for a J-K flip-flop.

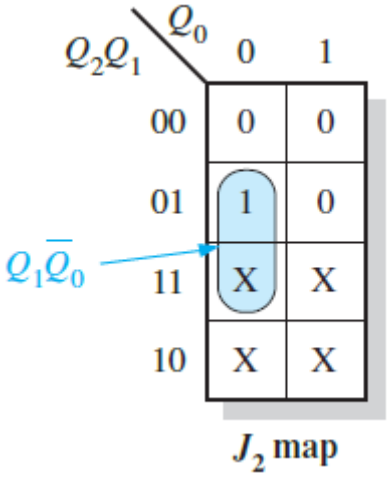
Output Transitions			Flip-Flop Inputs	
Q_N		Q_{N+1}	J	K
0	→	0	0	X
0	→	1	1	X
1	→	0	X	1
1	→	1	X	0

STEP 3

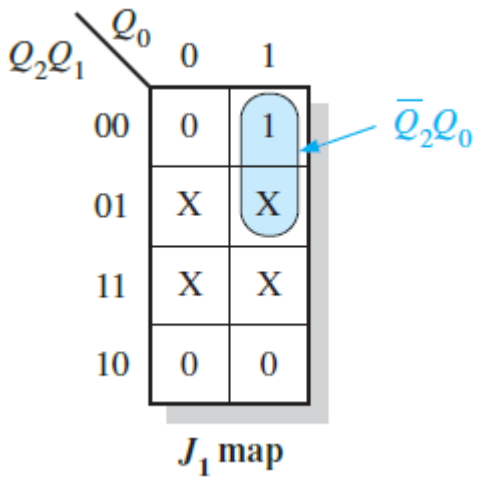
Present State			Next State			J_0	K_0	J_1	K_1	J_2	K_2
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0						
0	0	0	0	0	1	1	X	0	X	0	X
0	0	1	0	1	1	X	0	1	X	0	X
0	1	1	0	1	0	X	1	X	0	0	X
0	1	0	1	1	0	0	X	X	0	1	X
1	1	0	1	1	1	1	X	X	0	X	0
1	1	1	1	0	1	X	0	X	1	X	0
1	0	1	1	0	0	X	1	0	X	X	0
1	0	0	0	0	0	0	X	0	X	X	1

Once the table is populated, considering present state as output K-Map for J and K for the three flip-flops are filled and the logic requirements are found.

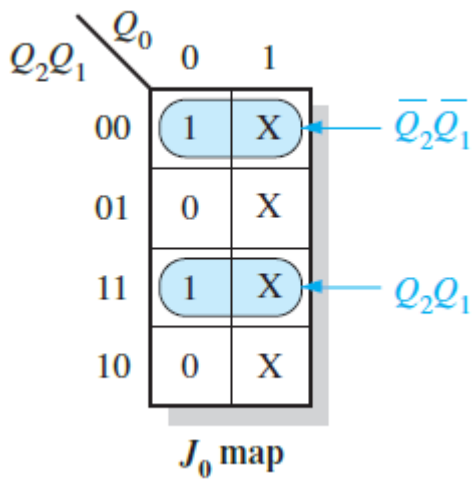
Irregular Sequence Counter (Gray Code Counter) Design



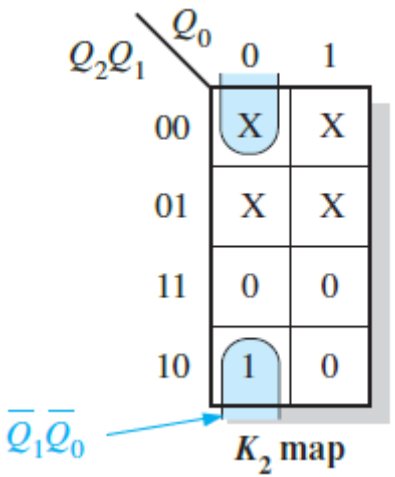
J_2 map



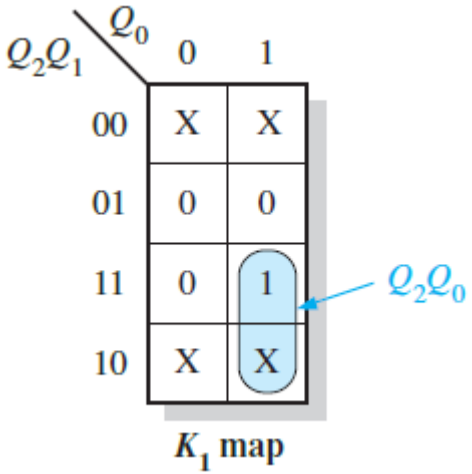
J_1 map



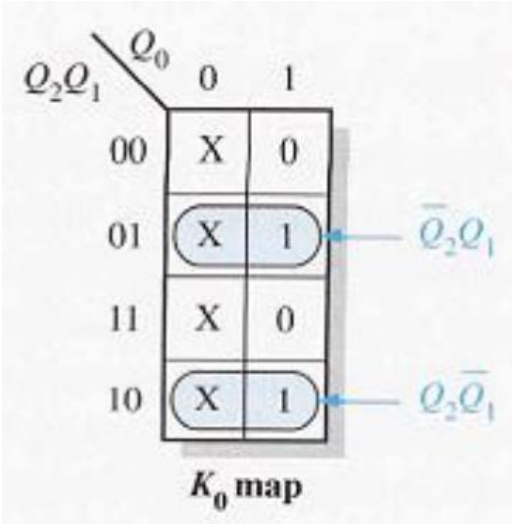
J_0 map



K_2 map



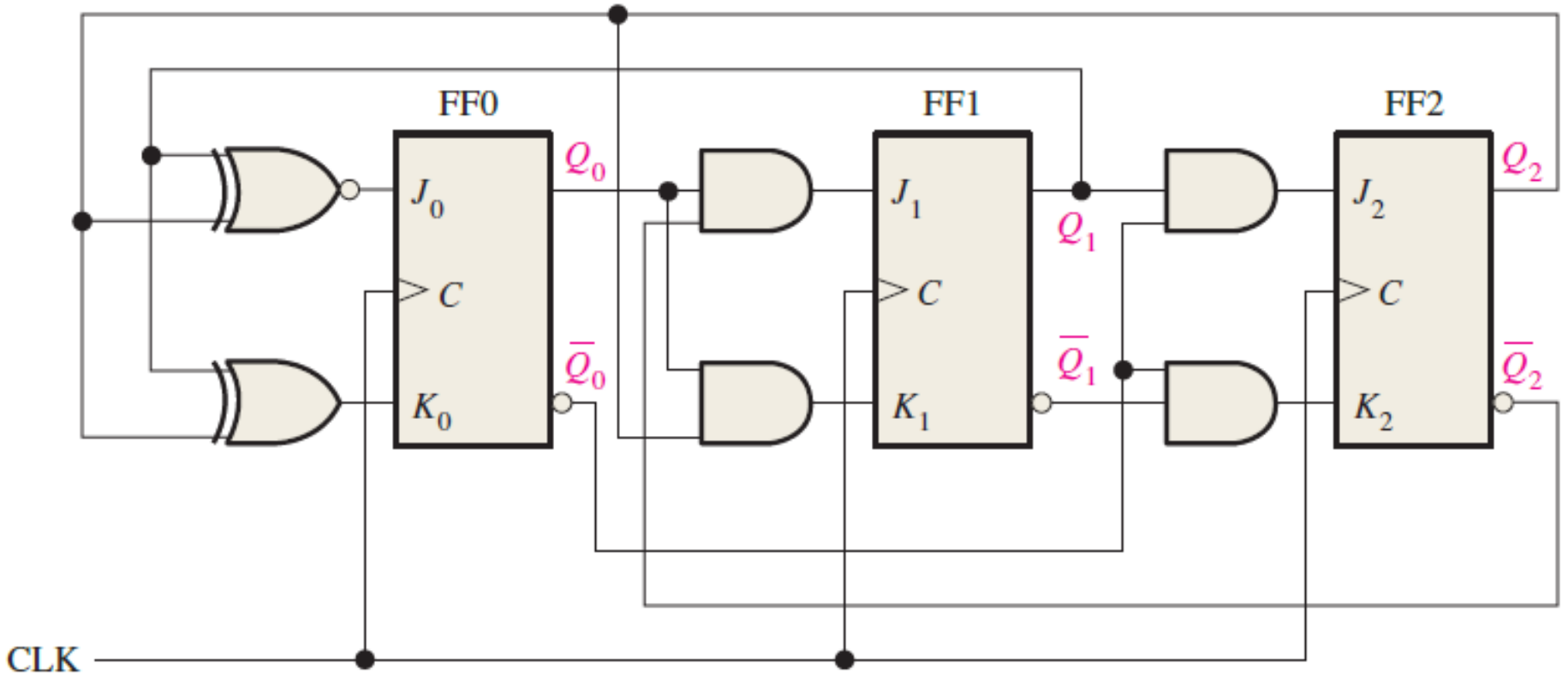
K_1 map



K_0 map

STEP 4

Irregular Sequence Counter (Gray Code Counter) Design

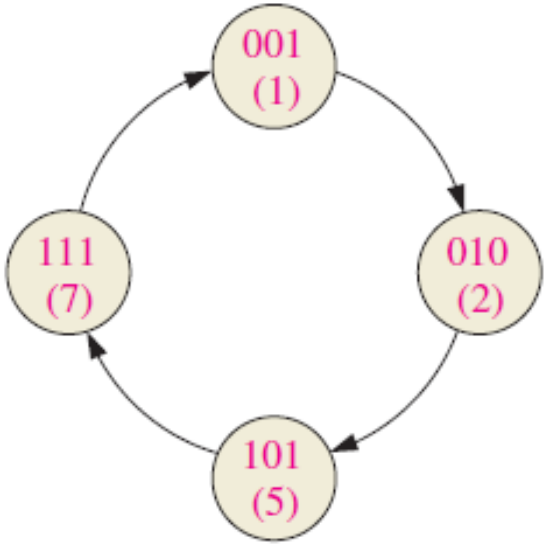


3-Bit Gray Code Counter Circuit

STEP 5

Irregular Sequence Counter Design

- Design a counter with the sequence 1,2,5 and 7.



Next-state table.

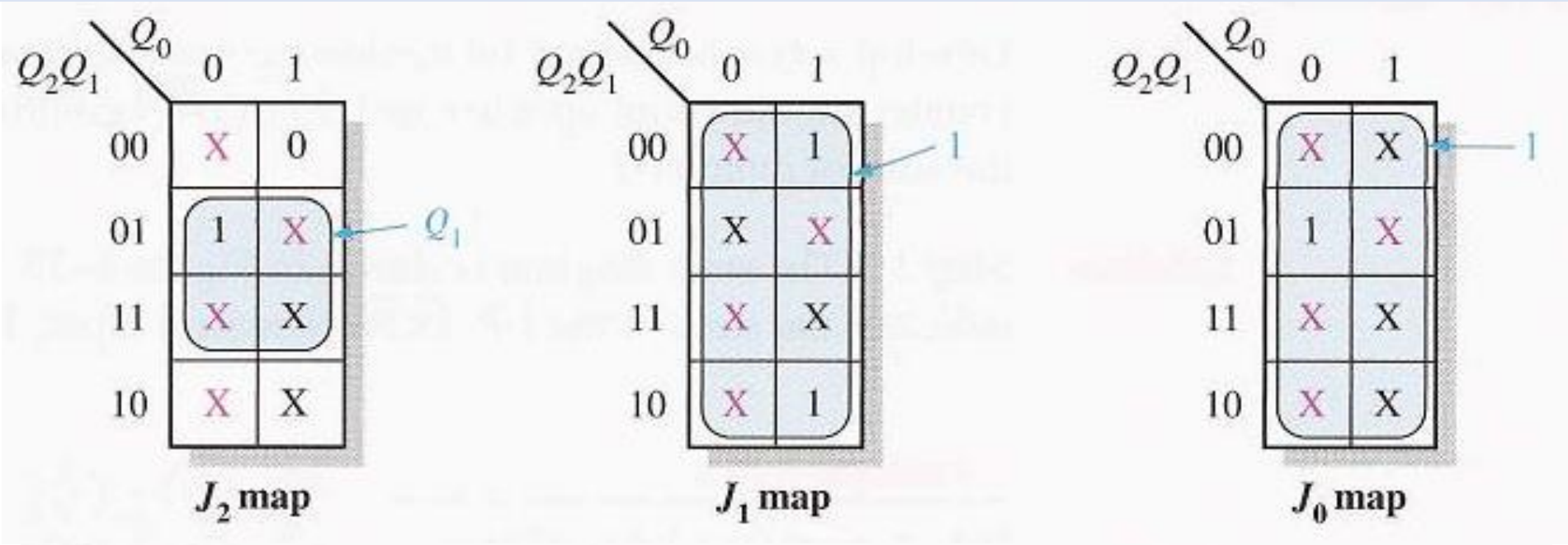
Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	0	0	1

Transition table for a J-K flip-flop.

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

Present State			Next State			J_0	K_0	J_1	K_1	J_2	K_2
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0						
0	0	1	0	1	0	X	1	1	X	0	X
0	1	0	1	0	1	1	X	X	1	1	X
1	0	1	1	1	1	X	0	1	X	X	0
1	1	1	0	0	1	X	0	X	1	X	1

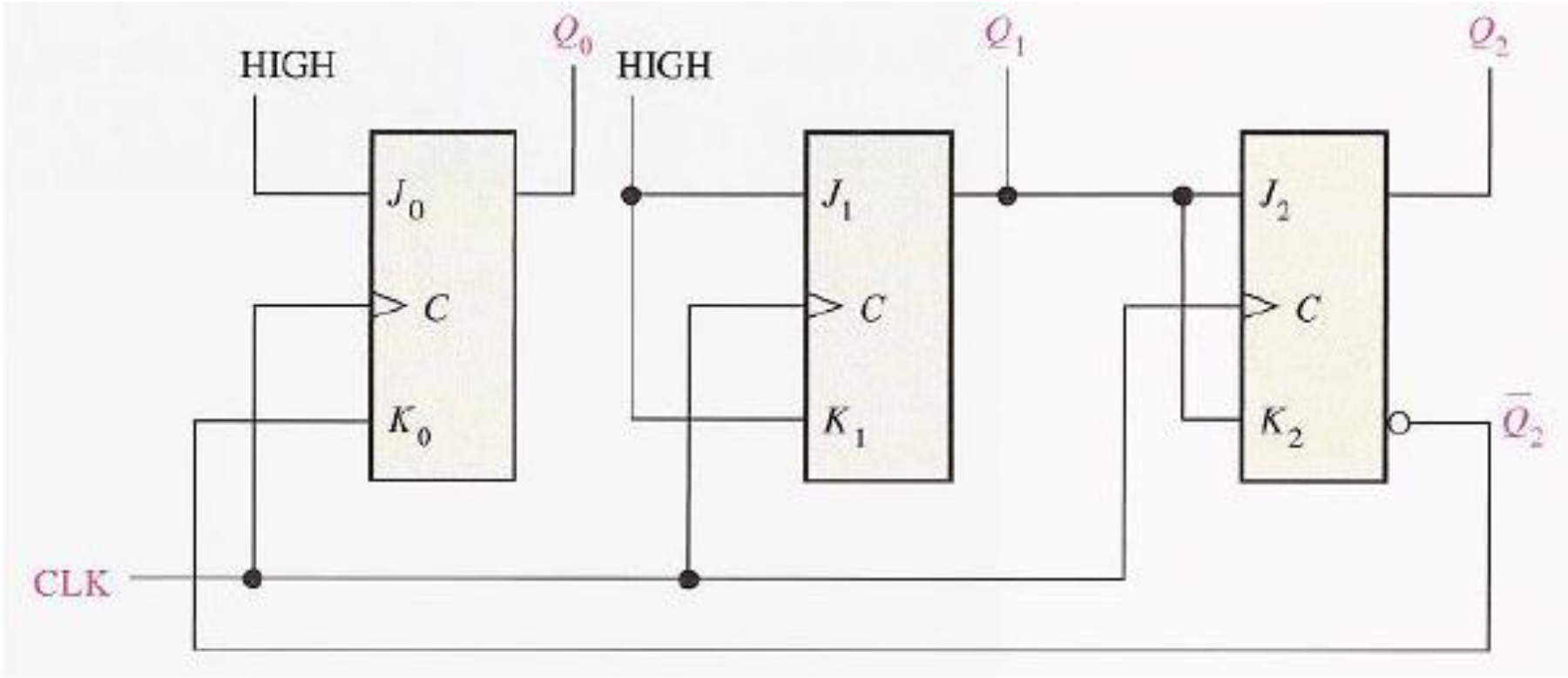
Irregular Sequence Counter Design



Irregular Sequence Counter Design

FROM THE K-MAP

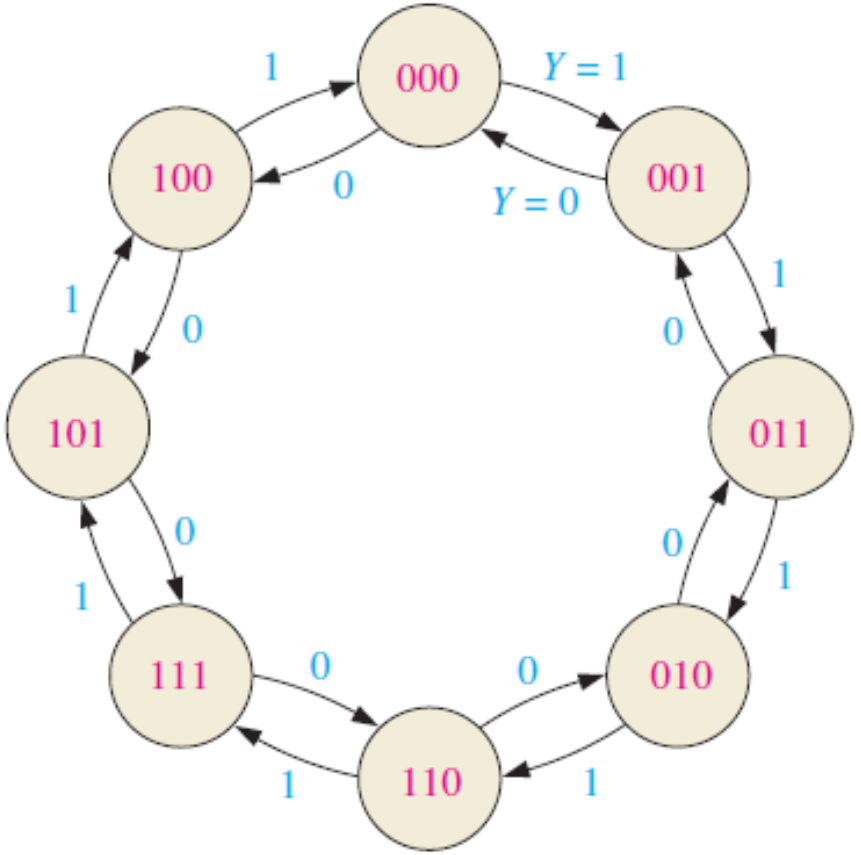
$$J_0 = 1, K_0 = \overline{Q}_2$$
$$J_1 = K_1 = 1$$
$$J_2 = K_2 = Q_1$$



The implementation of the counter for the given states

Irregular Sequence Counter Design

Develop a synchronous 3-bit up/down counter with a Gray code sequence using J-K flip-flops. The counter should count up when an UP/$\overline{\text{DOWN}}$ control input is 1 and count down when the control input is 0.

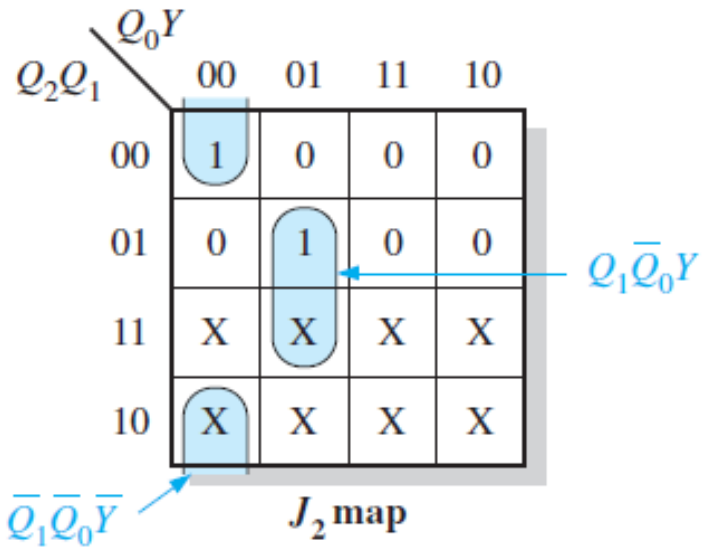


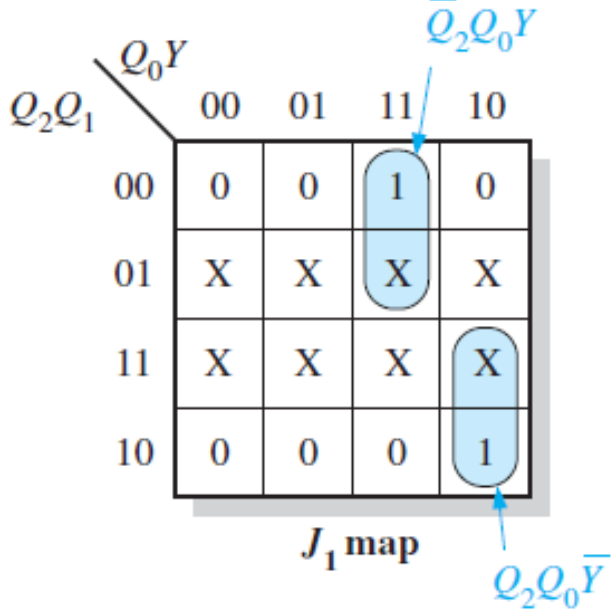
State Diagram

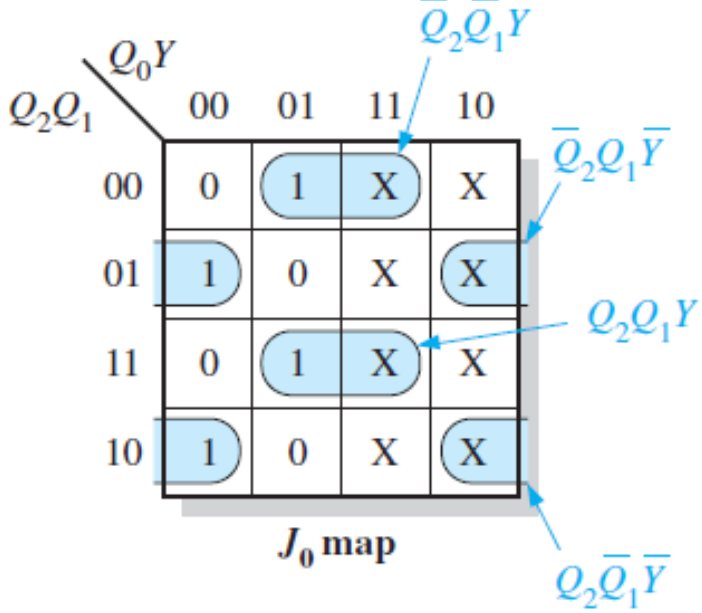
Next-state table for 3-bit up/down Gray code counter.

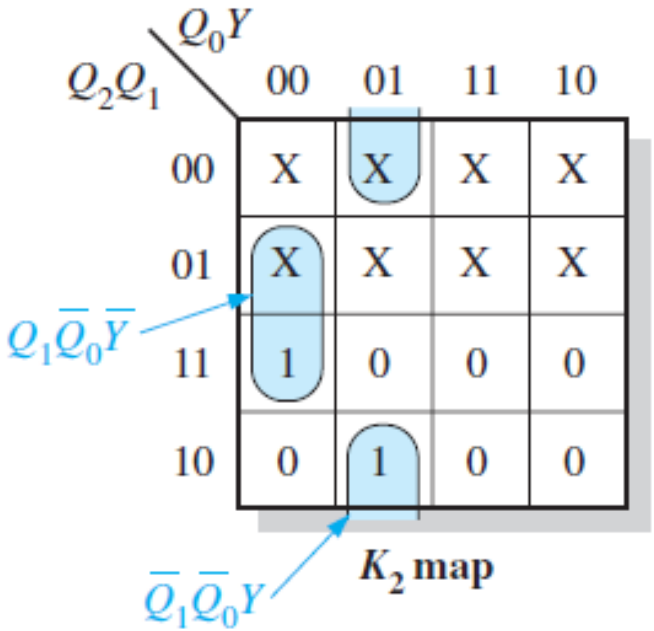
Present State			Next State					
			$Y = 0$ (DOWN)			$Y = 1$ (UP)		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	1
0	1	1	0	0	1	0	1	0
0	1	0	0	1	1	1	1	0
1	1	0	0	1	0	1	1	1
1	1	1	1	1	0	1	0	1
1	0	1	1	1	1	1	0	0
1	0	0	1	0	1	0	0	0

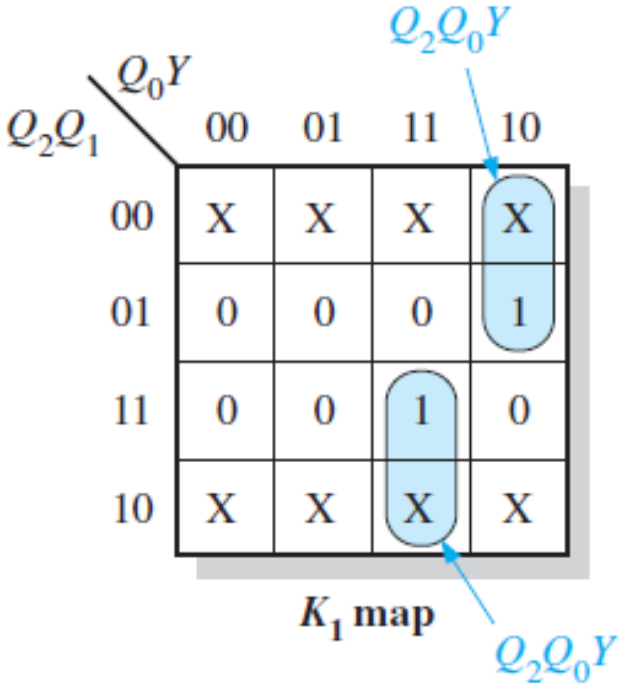
Irregular Sequence Counter Design

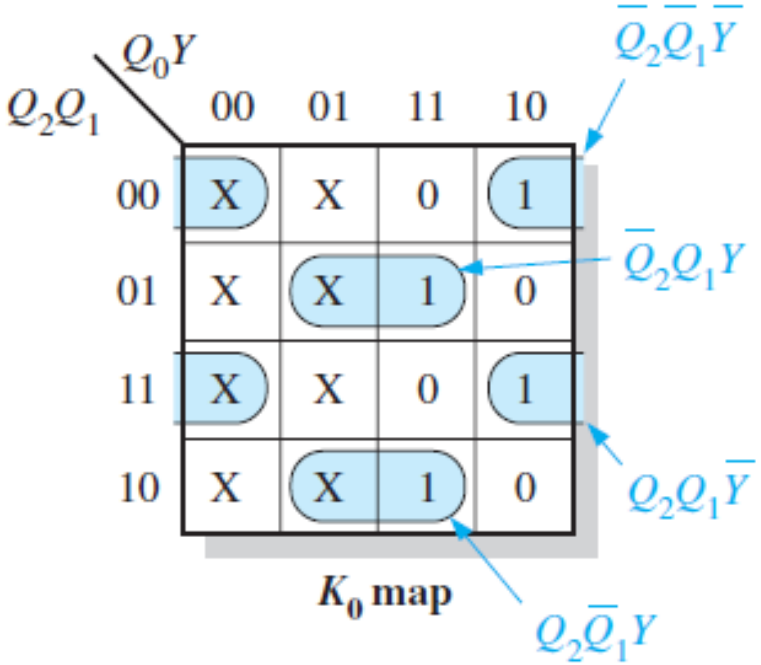












Irregular Sequence Counter Design

Logic Requirements for the inputs J and K:

$$J_0 = Q_2Q_1Y + Q_2\overline{Q}_1\overline{Y} + \overline{Q}_2\overline{Q}_1Y + \overline{Q}_2Q_1\overline{Y} \qquad K_0 = \overline{Q}_2\overline{Q}_1\overline{Y} + \overline{Q}_2Q_1Y + Q_2\overline{Q}_1Y + Q_2Q_1\overline{Y}$$

$$J_1 = \overline{Q}_2Q_0Y + Q_2Q_0\overline{Y} \qquad K_1 = \overline{Q}_2Q_0\overline{Y} + Q_2Q_0Y$$

$$J_2 = Q_1\overline{Q}_0Y + \overline{Q}_1\overline{Q}_0\overline{Y} \qquad K_2 = Q_1\overline{Q}_0\overline{Y} + \overline{Q}_1\overline{Q}_0Y$$

Please draw the circuit for the counter on your own!!!!!!!!!

References

1. Thomas L. Floyd, “Digital Fundamentals” 11th edition, Prentice Hall – Pearson Education.

Thank You