Chapter 9

Counters

Course Outcome

1. Explain number systems, codes, digital arithmetic operation and circuits.
2. Use Boolean algebra and Karnaugh Maps to minimize Boolean expressions for the design of digital logic circuits.
3. Explain and use flip-flops, latches, counters, multiplexers and de-multiplexers.
4. Design and construct combinational digital logic circuits using appropriate logic design techniques.
5. Design and construct synchronous sequential digital logic circuits using appropriate logic design techniques.

Learning Outcome

1. Discuss the types of state machines
2. Describe the difference between an asynchronous and a synchronous counter
3. Analyze counter timing diagrams and circuits
4. Determine the modulus of a counter
5. Recognize the difference between a 4-bit binary counter and a decade counter
6. Use an up/down counter to generate forward and reverse binary sequences
7. Determine the sequence of a counter
8. Design a counter that will have any specified sequence of states
9. Use cascaded counters to achieve a higher modulus

Outline

1. Finite State Machines
2. Asynchronous Counters
3. Synchronous Counters
4. Up/Down Synchronous Counters
5. Design of Synchronous Counters
6. Cascaded Counters
7. Counter Decoding
8. Counter Applications

Finite State Machines

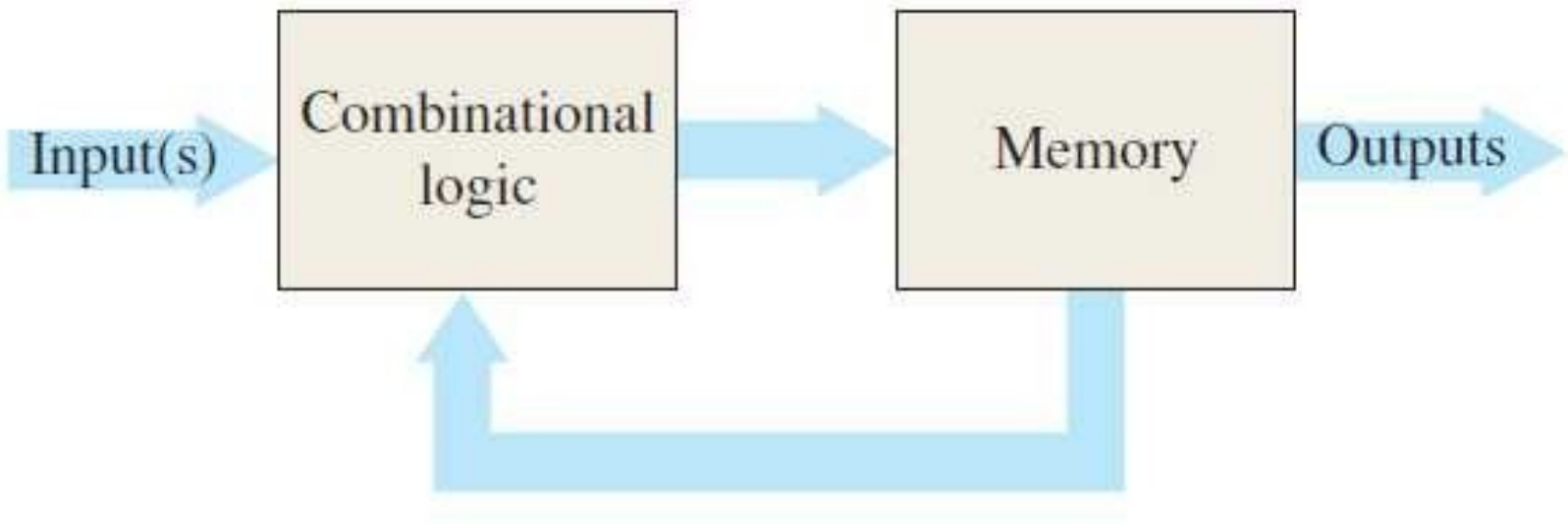
Models of Finite State Machines

- Two basic types of state machines are:
  - Moore state machine is one where the outputs depend only on the internal present state. (Chapter 10)
  - Mealy state machine is one where the outputs depend on both internal present state and inputs. (Chapter 10)

Finite State Machines

Moore Machine

- Output depends on present state only.

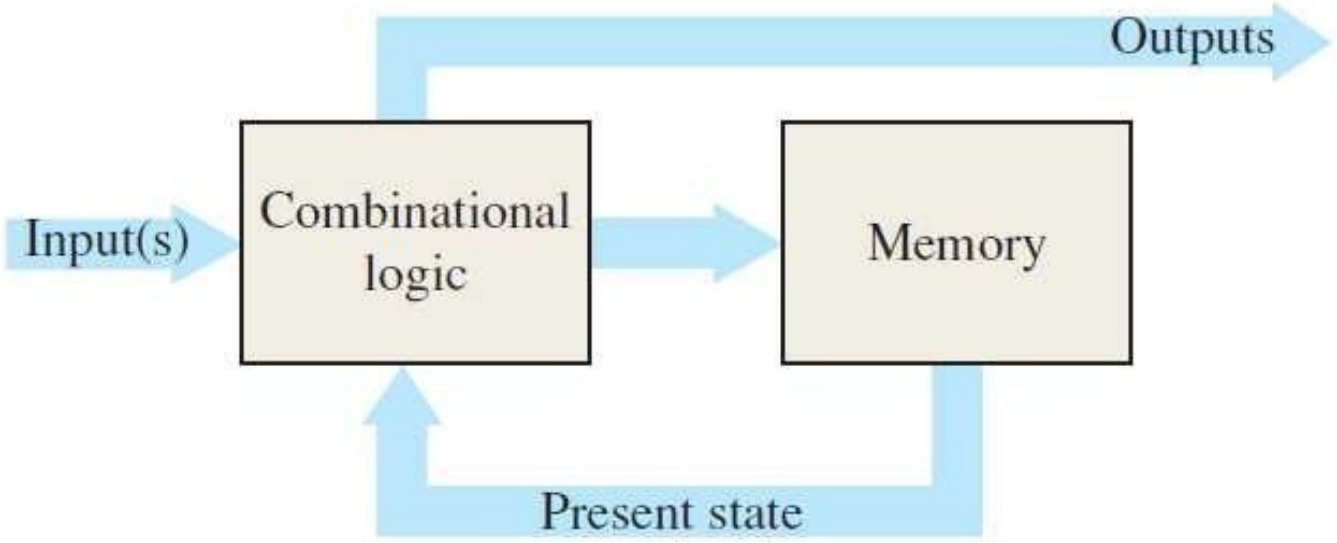


Moore machine

Finite State Machines

Mealy Machine

- Output depends on both present state and Input.

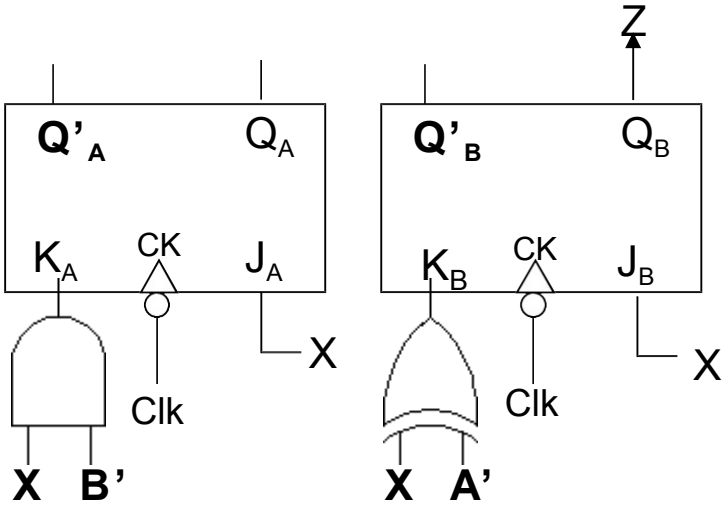


Mealy machine

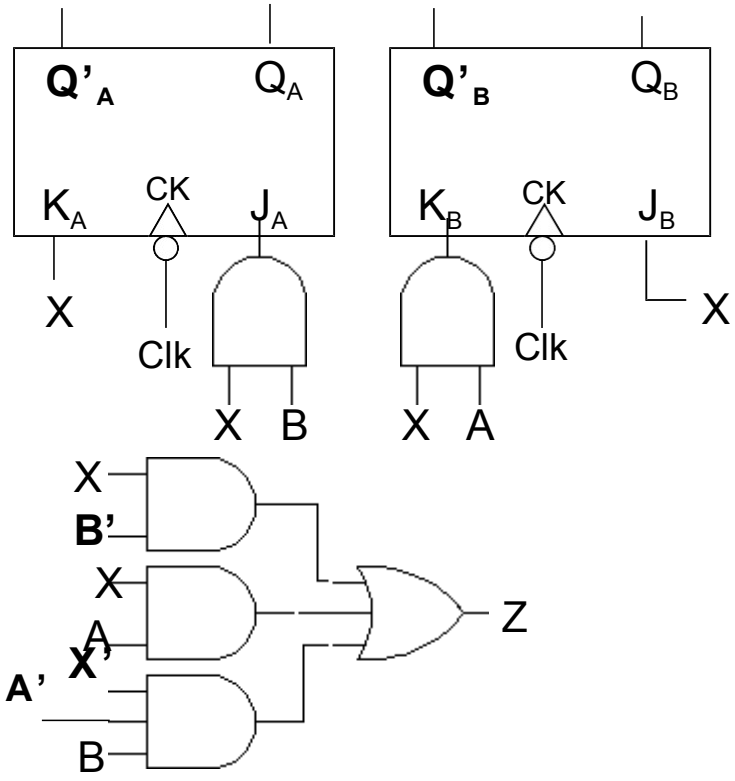
Finite State Machines

Example of a Moore and Mealy Machine

Moore state machine

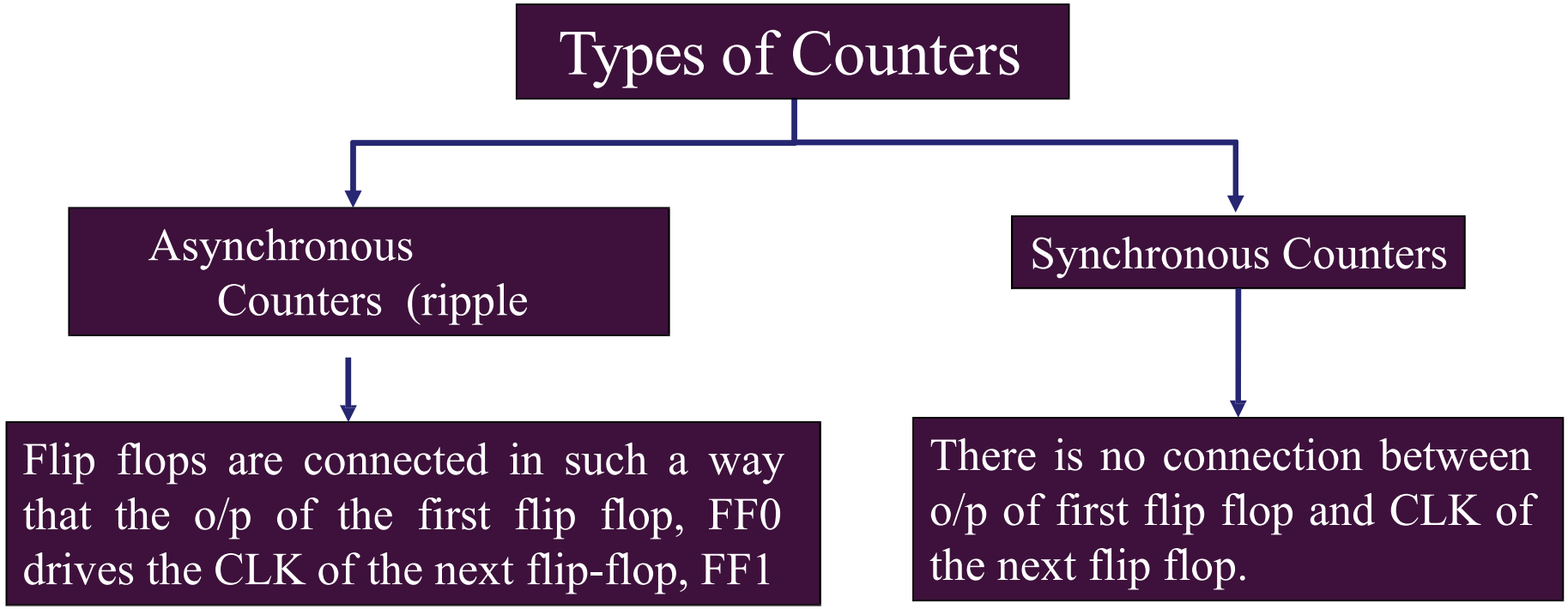


Mealy state machine



Introduction to Counters

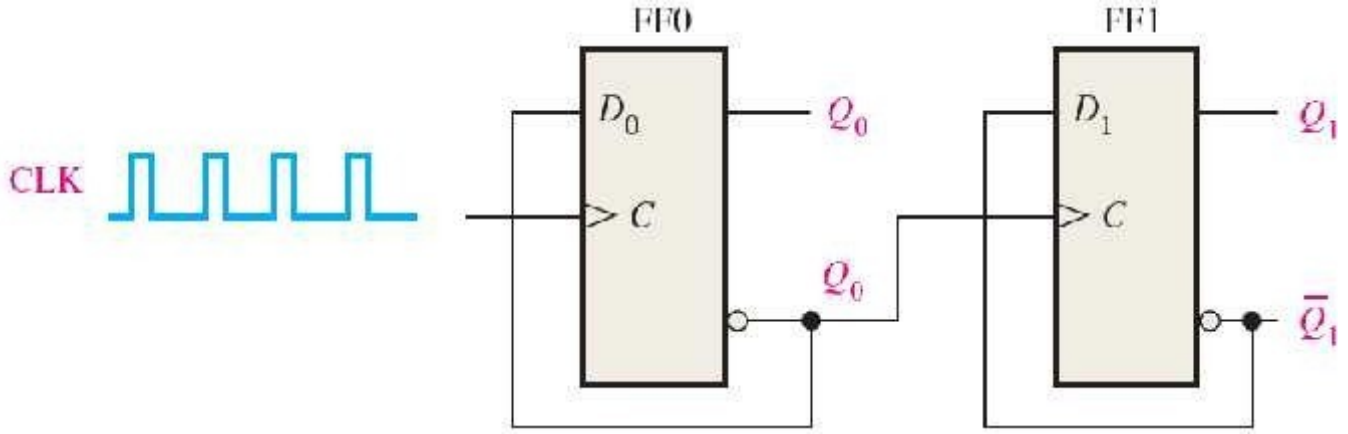
- The counter is Sequential Circuits, count from 0 to 10 (Up Counter) or from 10 to 0 (Down Counter).



Asynchronous Counters

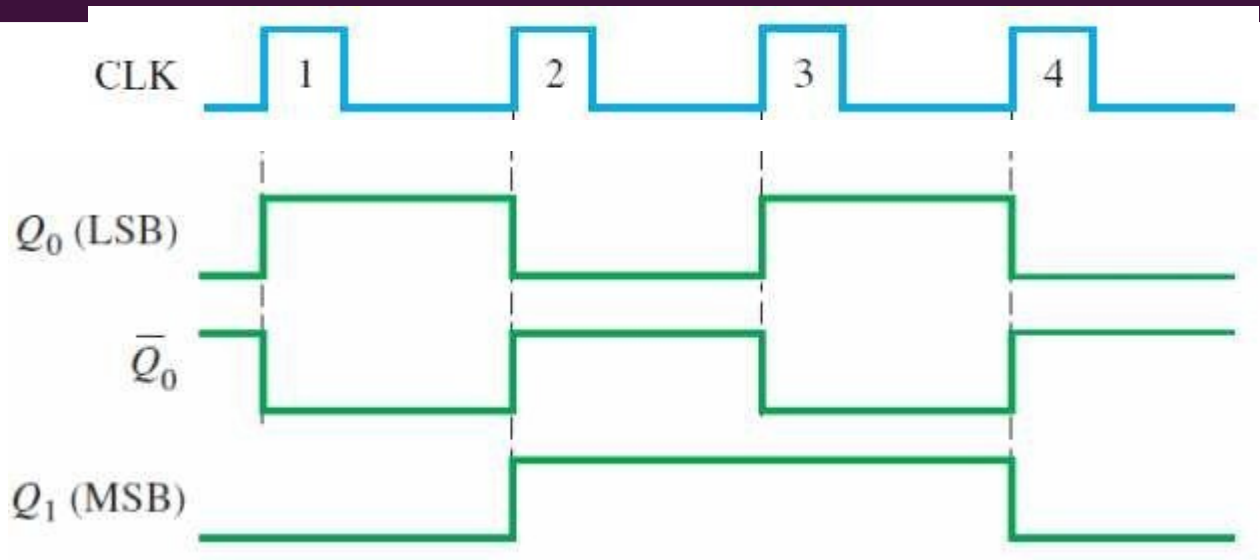
A 2-Bit Asynchronous Binary Counter

- The clock (CLK) is applied to the clock input (C) of only the first flip-flop, FF0, which is always the least significant bit (LSB). The second flip-flop, FF1, is triggered by the $\overline{Q}_0$ output of FF0.
- FF0 changes state at the positive-going edge of each clock pulse, but FF1 changes only when triggered by a positive-going transition of the Q_0 output of FF0.
- The two flip-flops are never simultaneously triggered, so the counter operation is asynchronous.



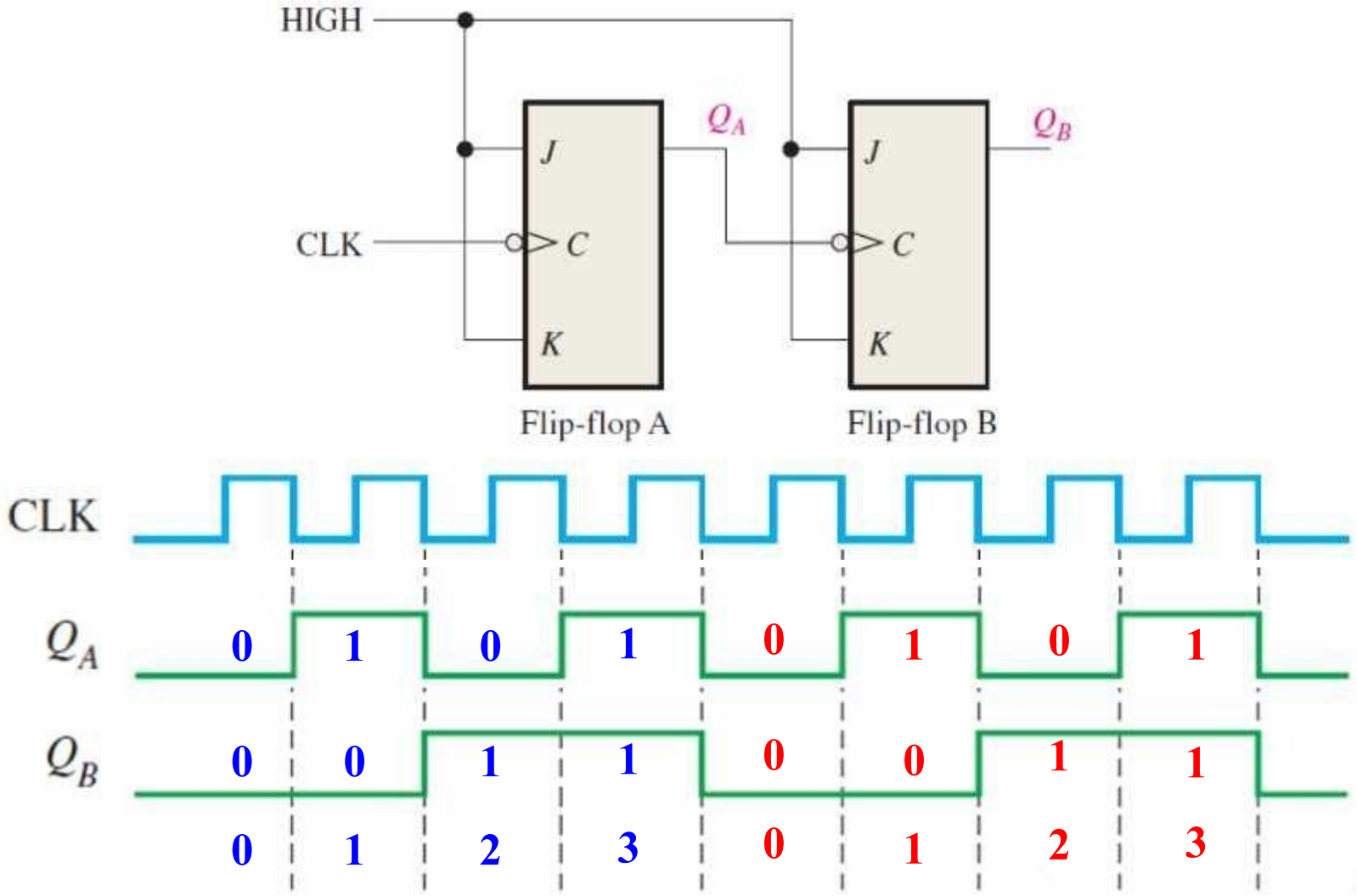
A 2-bit asynchronous binary counter

Asynchronous Counters



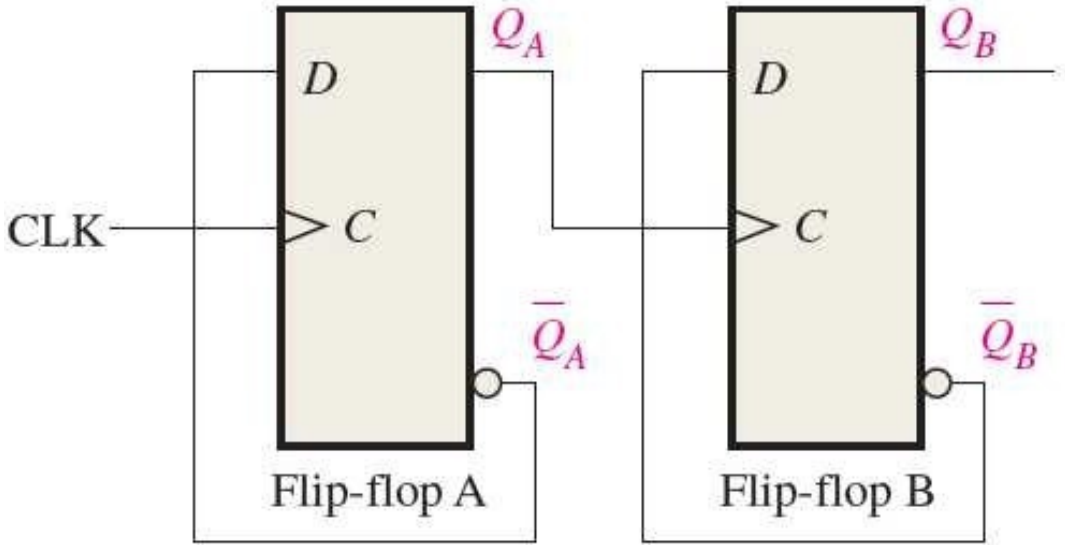
Asynchronous Counters

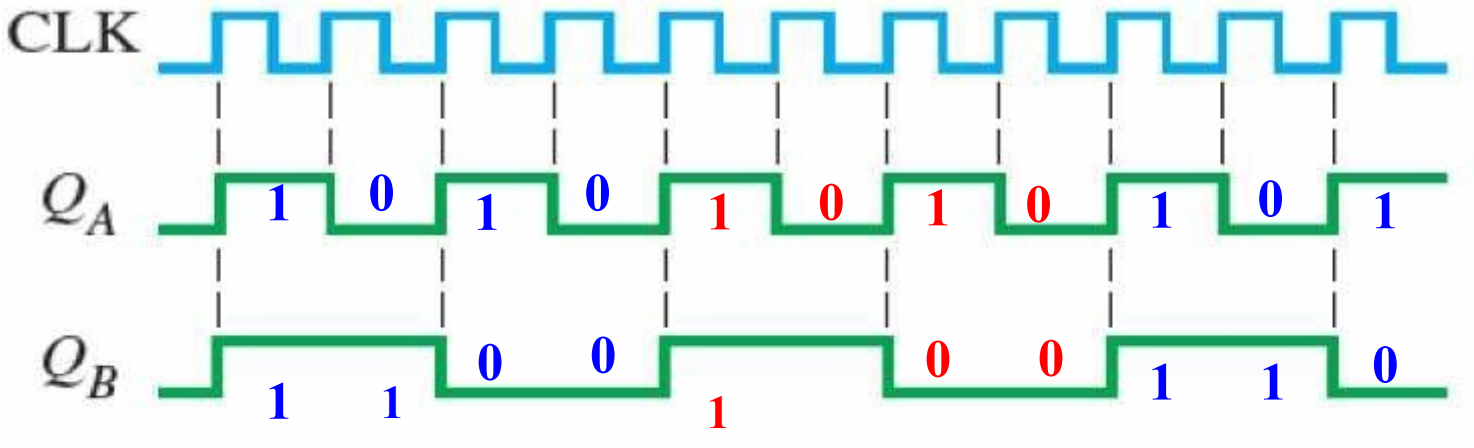
A 2-Bit Asynchronous Binary Counter



Asynchronous Counters

A 2-Bit Asynchronous Binary Counter



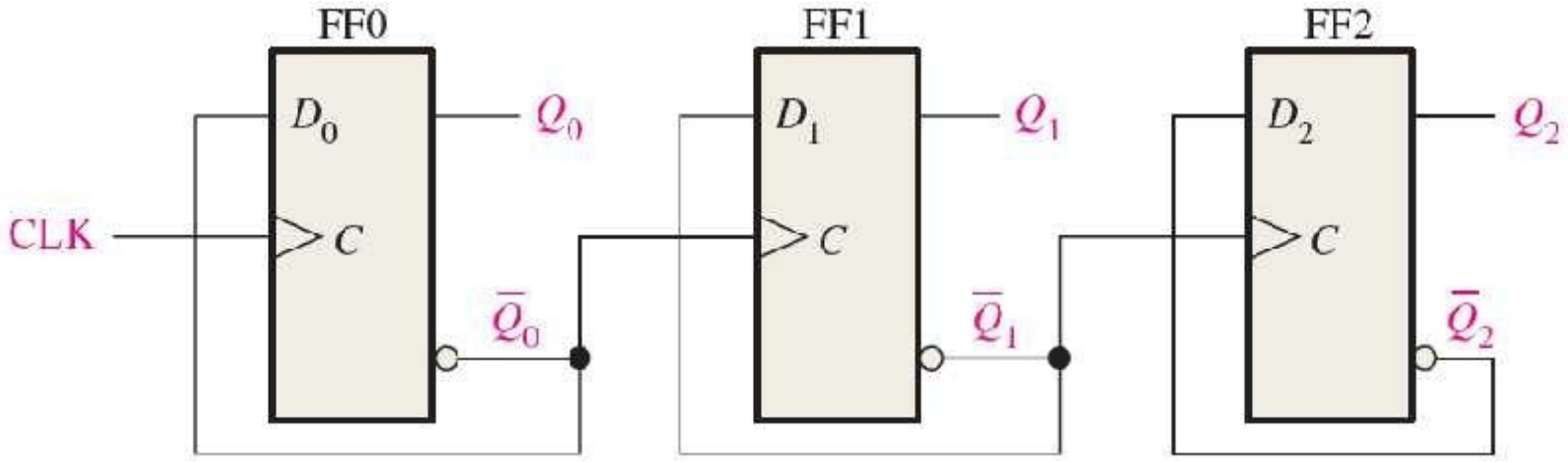


Asynchronous Counters

A 3-Bit Asynchronous Binary Counter

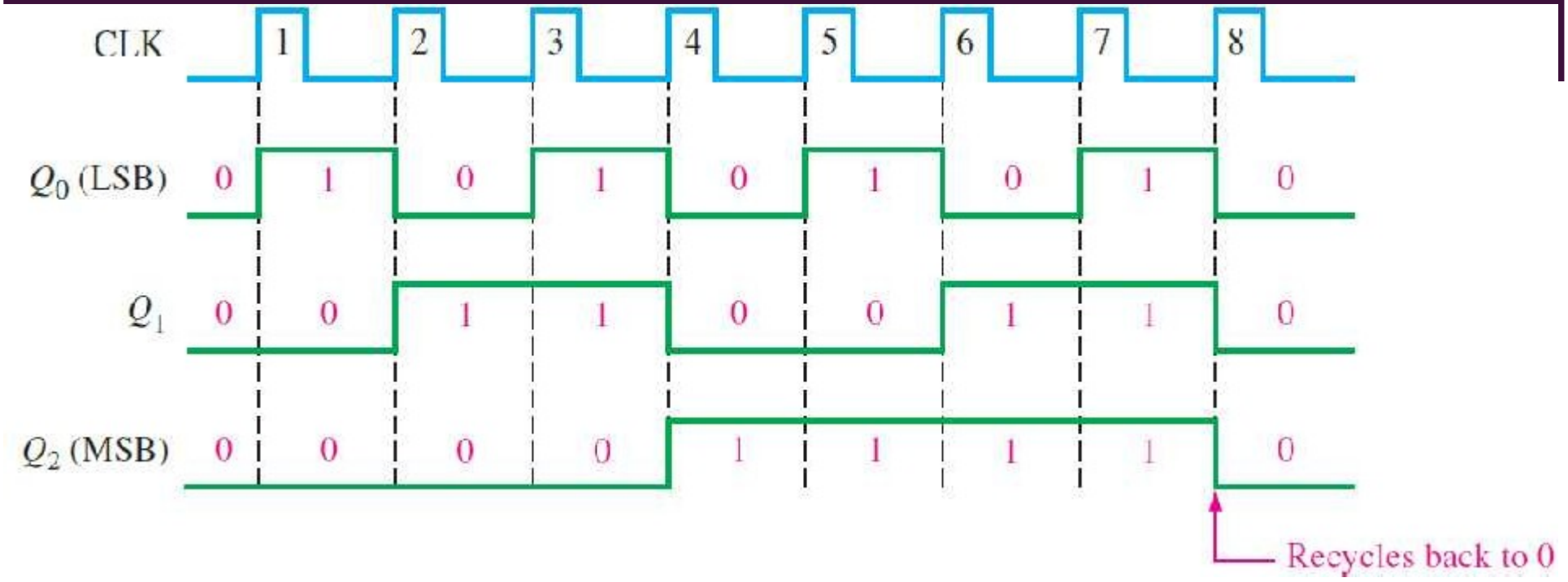
- The basic operation is the same as that of the 2-bit counter except that the 3-bit counter has eight states, due to its three flip-flops.

- Notice that the counter progresses through a binary count of zero through seven and then recycles to the zero state.



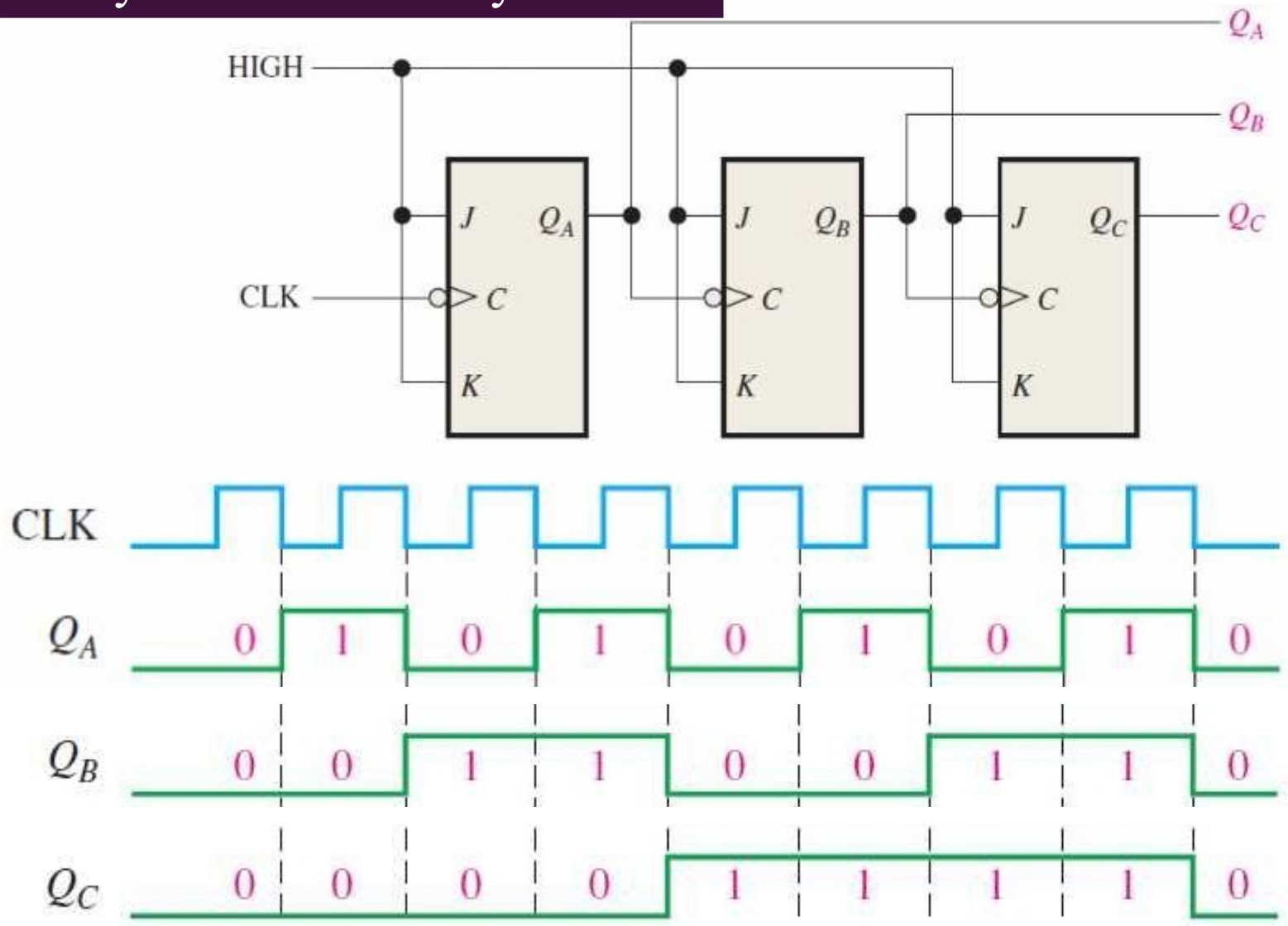
A 3-bit asynchronous binary counter

Asynchronous Counters



Asynchronous Counters

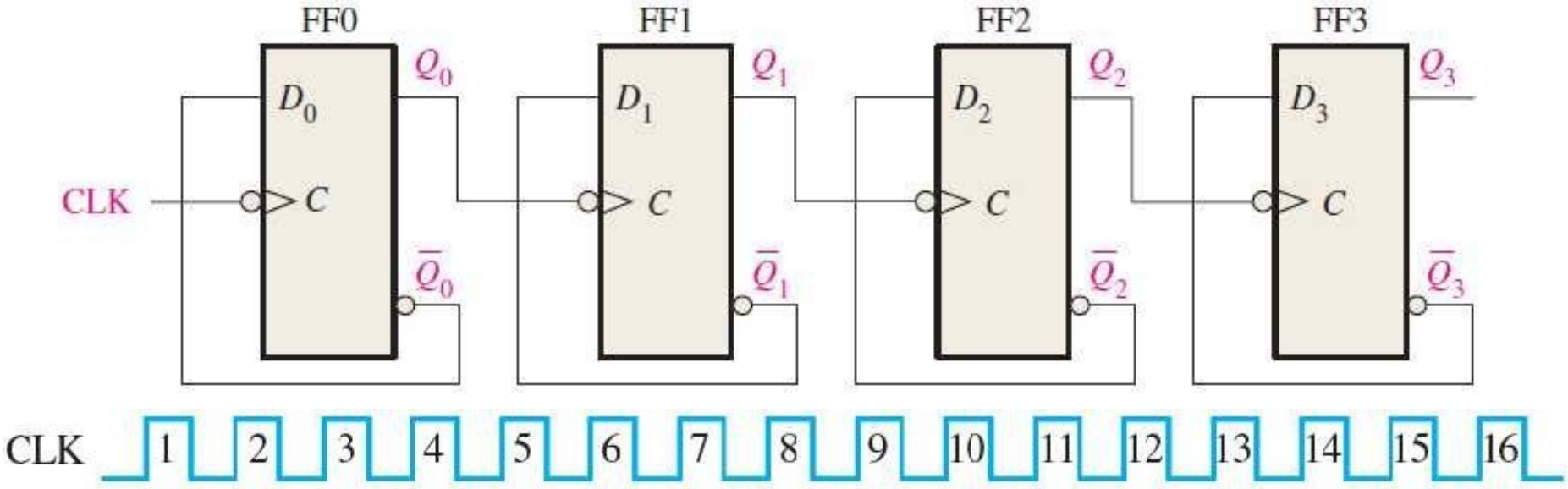
A 3-Bit Asynchronous Binary Counter

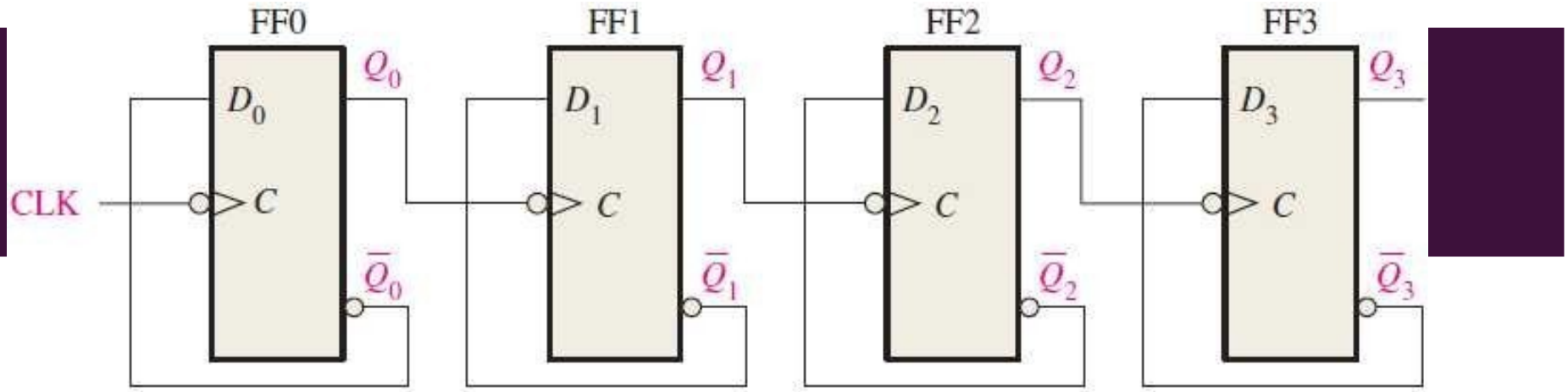


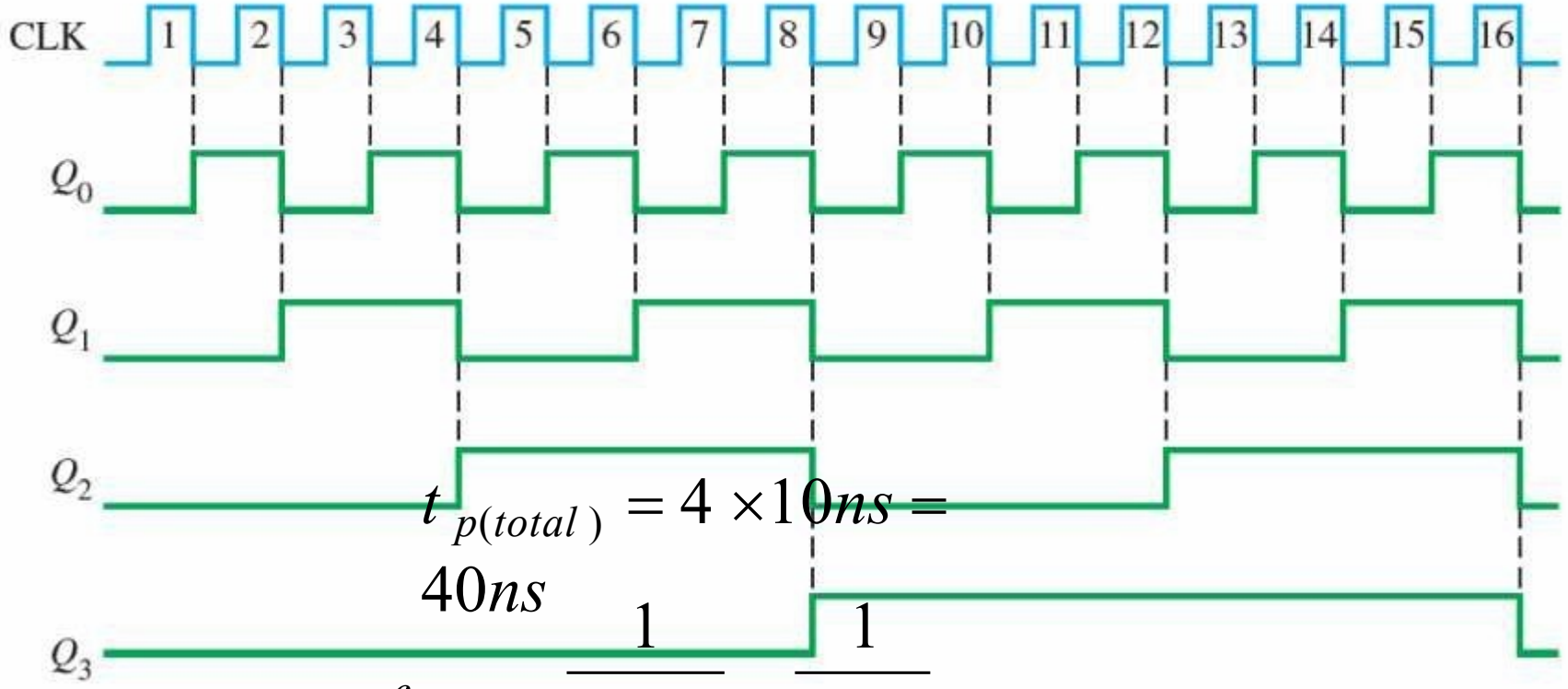
Asynchronous Counters

Propagation delay- Example

A4-bit asynchronous binary counter is shown below. Each D flip-flop is negative edge-triggered and has a propagation delay for 10ns. Develop a timing diagram showing the *Q output of each flip-flop, and determine the total propagation* delay time from the triggering edge of a clock pulse until a corresponding change can occur in the state of Q_3. Also determine the maximum clock frequency at which the counter can be operated.







$$t_{p(total)} = 4 \times 10ns = 40ns$$

$$f_{\max} = \frac{1}{t_{p(tot)}} = \frac{1}{40ns}$$

Asynchronous Counters

Asynchronous Decade Counters

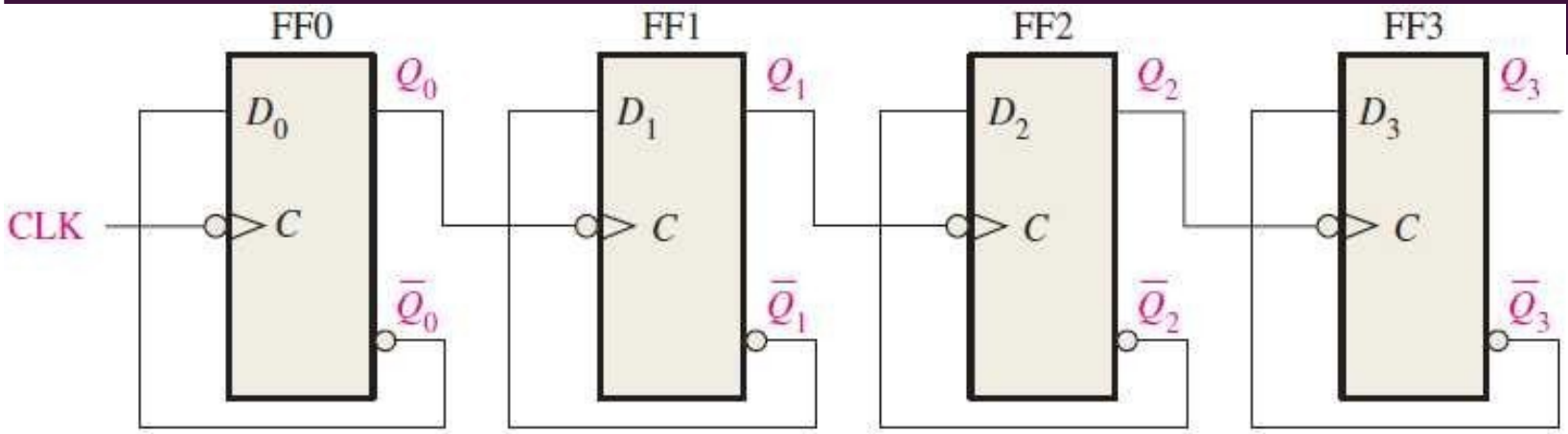
- The **modulus of a counter is the number of states through which the counter will** sequence.

- The maximum modulus (MOD) number of a counter is
  2^n, where n is the number of flip-flops in the counter.

  2- bit up or down counter = MOD – 4
  3- bit up or down counter = MOD – 8
  4- bit up or down counter = MOD – 16

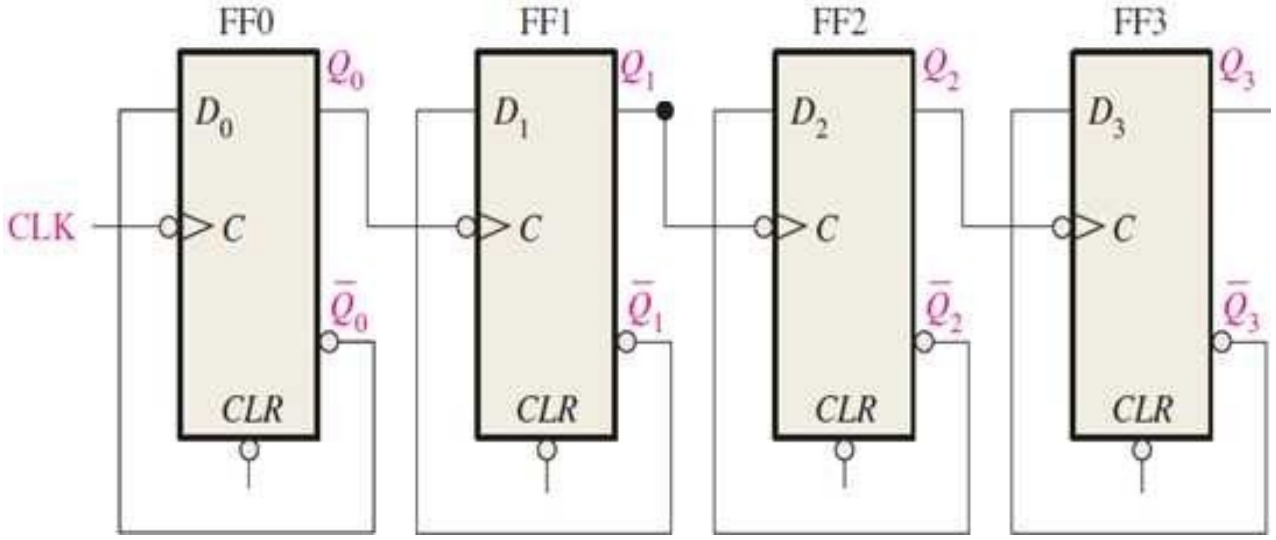
- Counters with ten states in their sequence are called **decade counters. A decade counter** with a count sequence of zero (0000) through nine (1001) is a BCD decade counter because its ten-state sequence produces the BCD code.

Asynchronous Counters



(0000) to (1111)

(0000) to (1001)

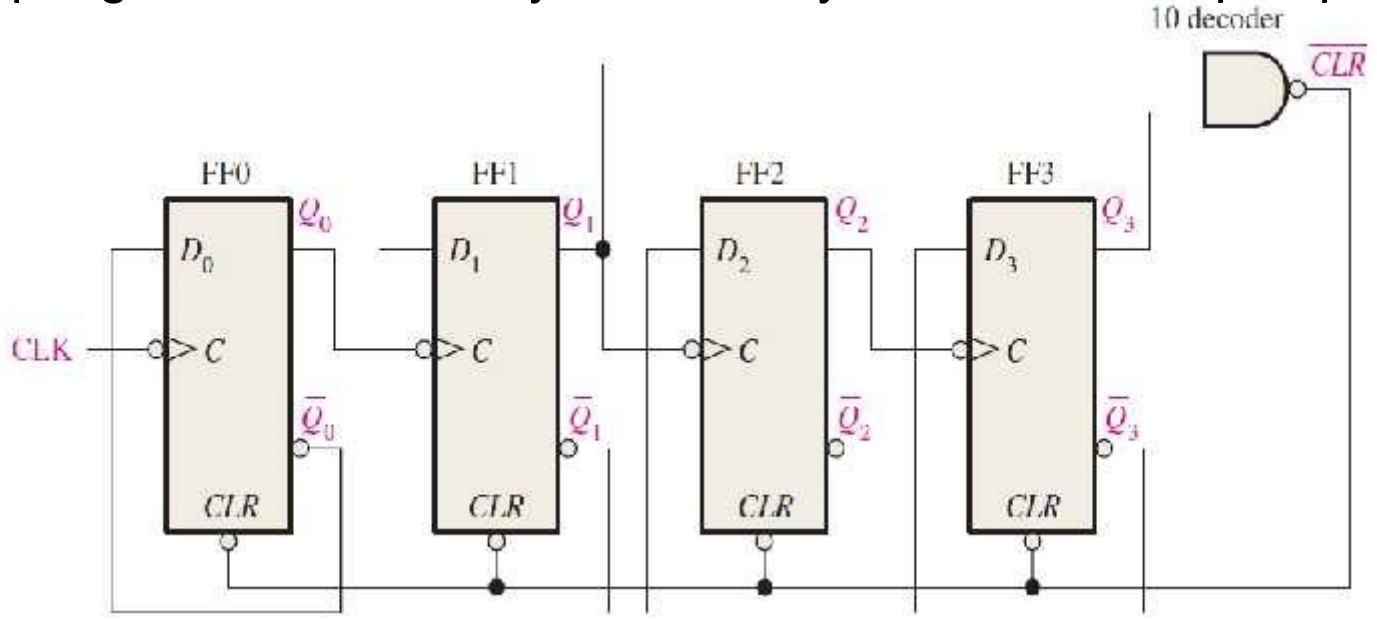


Asynchronous Counters

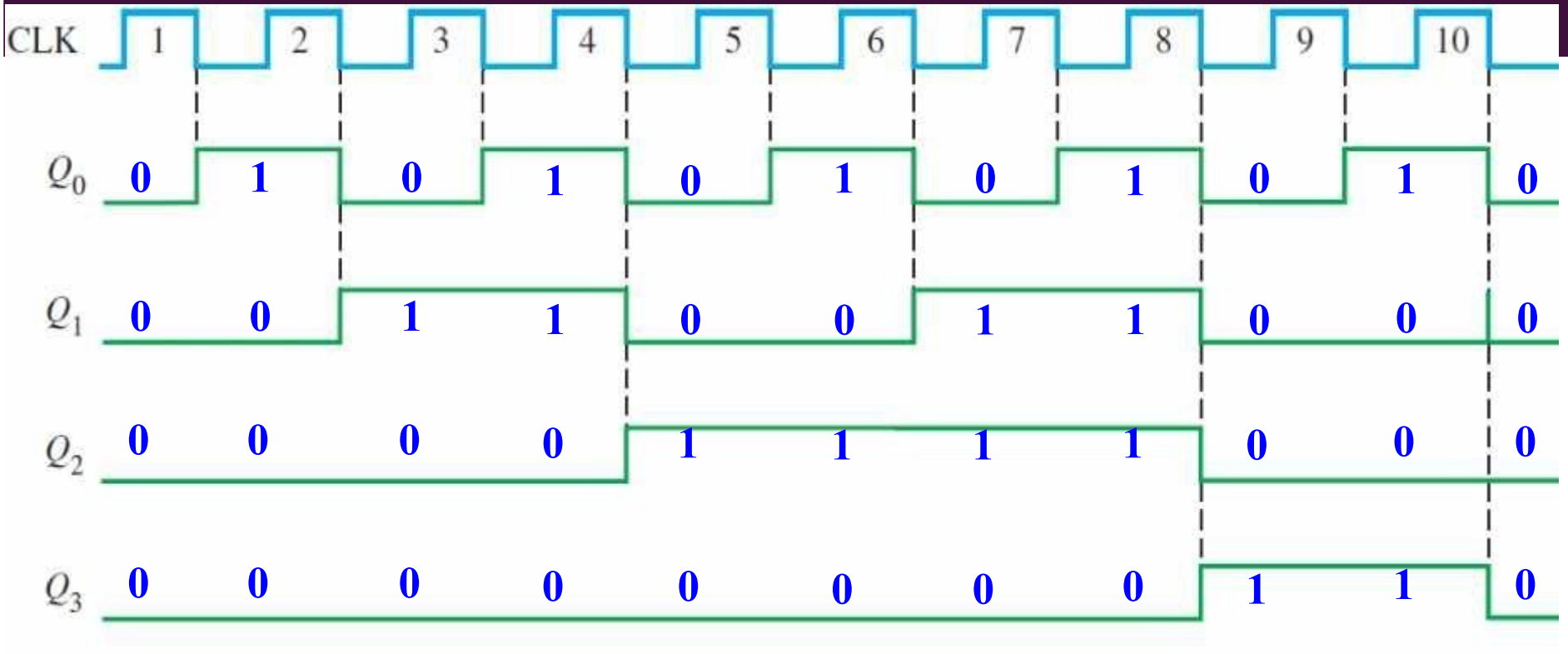
Partial Decoding

- This arrangement is an example of *partial decoding, in which the two unique states ($Q_1 = 1$* and $Q_3 = 1$*) are sufficient to decode the count of ten because none of the other states (zero* through nine) have both Q_1 *and* Q_3 *HIGH at the same time.*

- *When the counter goes into* count ten (1010), the decoding gate output goes LOW and asynchronously resets all the flip-flops.



Asynchronous Counters



Asynchronous Counters

Example

Show how an asynchronous counter with J-K flip-flops can be implemented having a modulus of twelve with binary sequence from 0000 through 1011.

Solution

Since three flip-flops can produce a maximum of eight states, four flip-flops are required to produce any modulus greater than eight but less than or equal to sixteen.

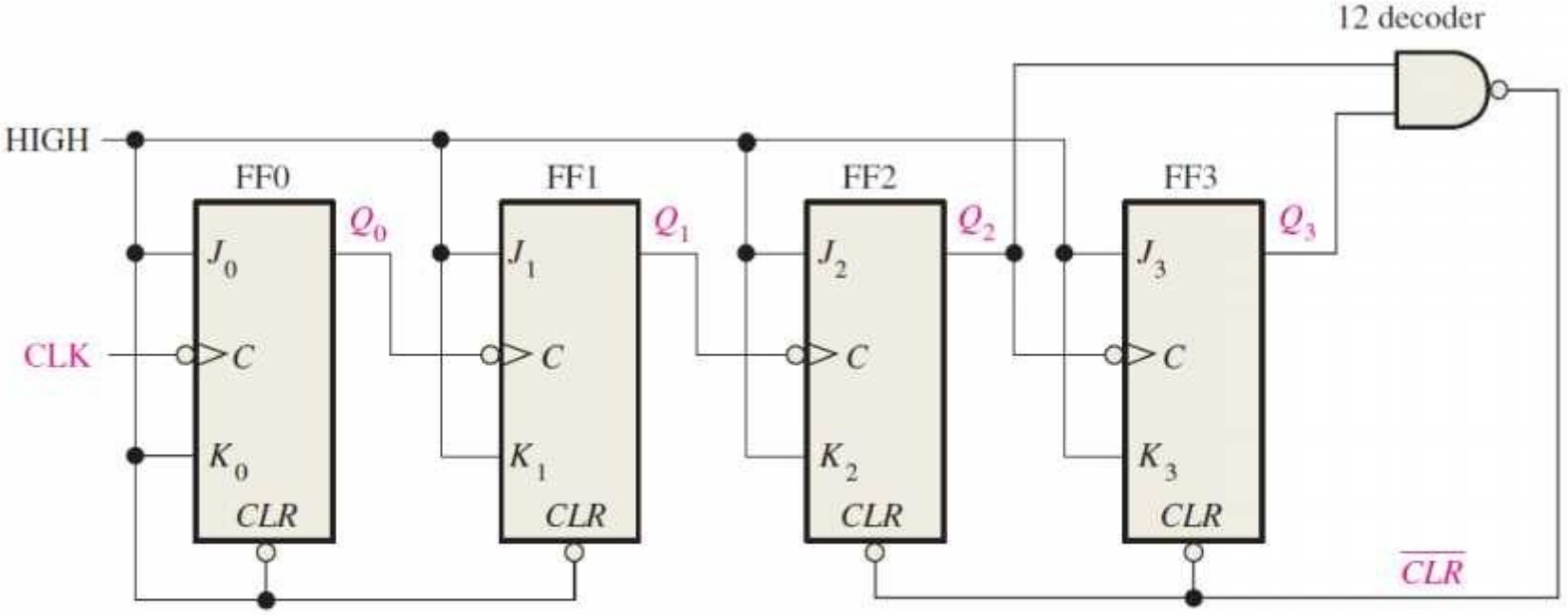
When the counter gets to its last state, 1011, it must recycle back to 0000 rather than going to its normal next state of 1100, as illustrated in the following sequence chart:

Asynchronous Counters

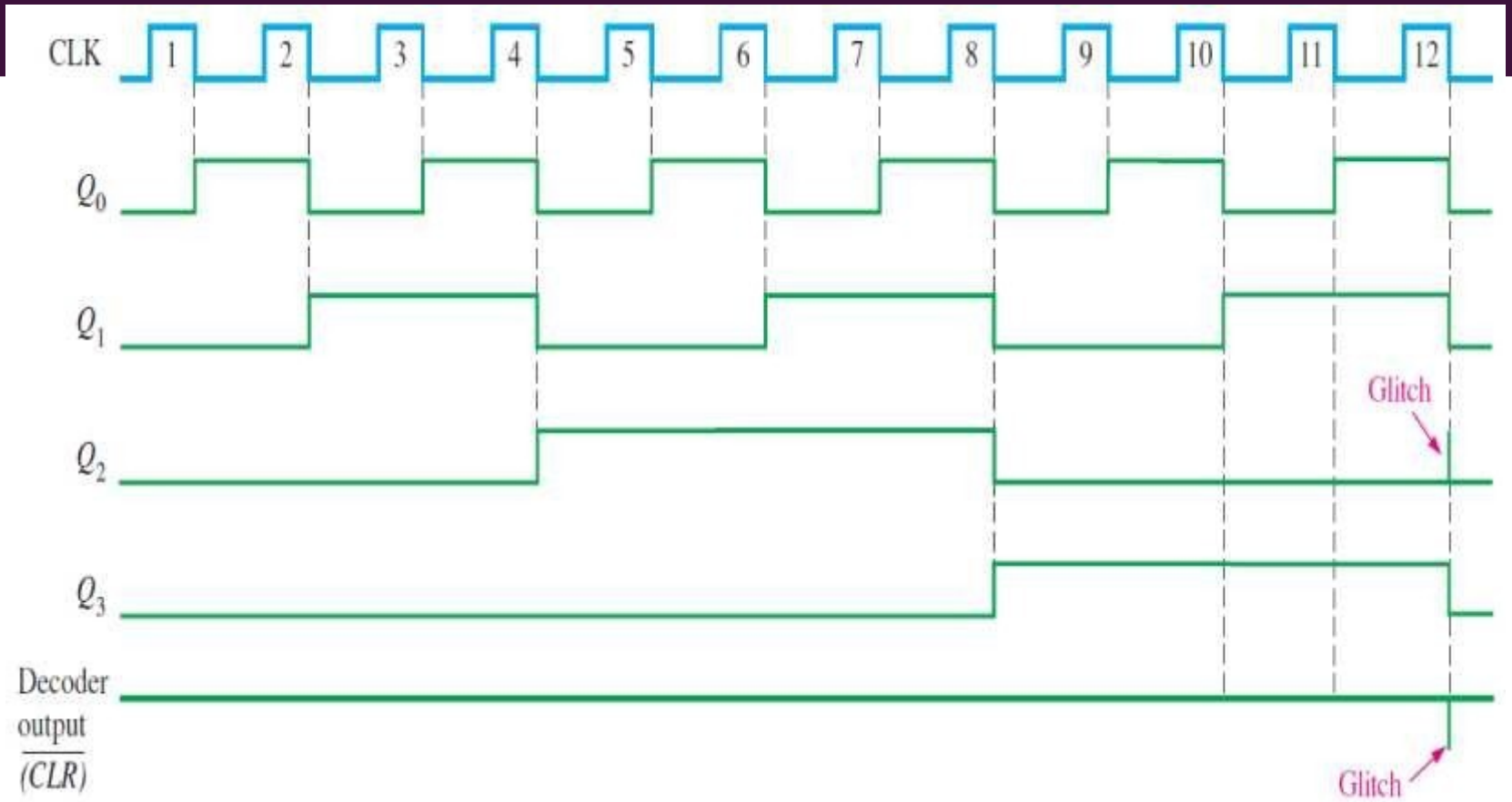
0	0	0	0
.	.	.	.
.	.	.	.
.	.	.	.
1	0	1	1
1	1	0	0

Recycles

Normal next state



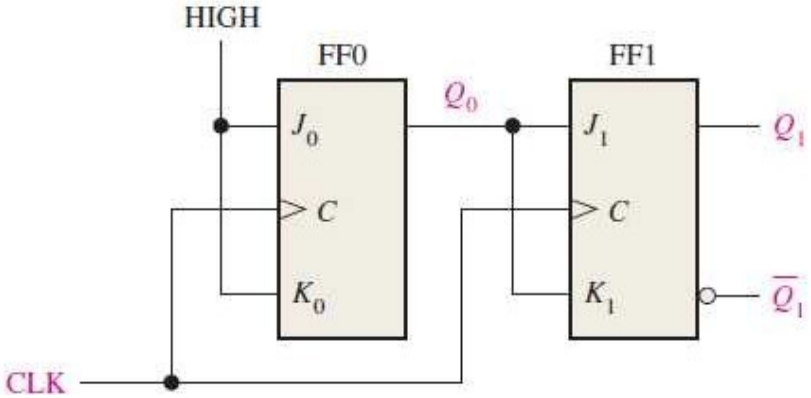
Asynchronous Counters



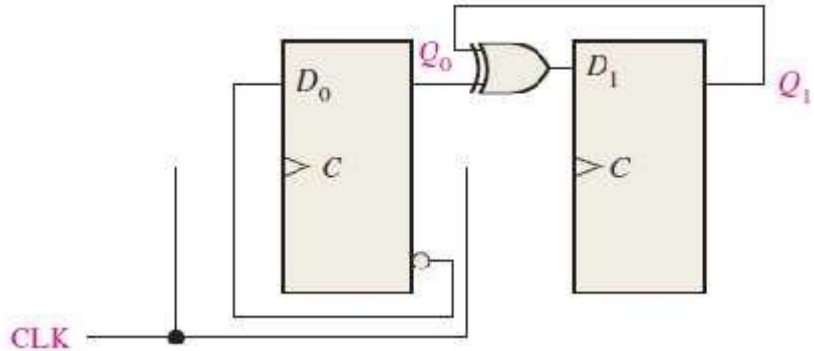
Synchronous Counters

- A synchronous counter is one in which all the flip-flops in the counter are clocked at the same time by a common clock pulse.

A 2-Bit Synchronous Binary Counter



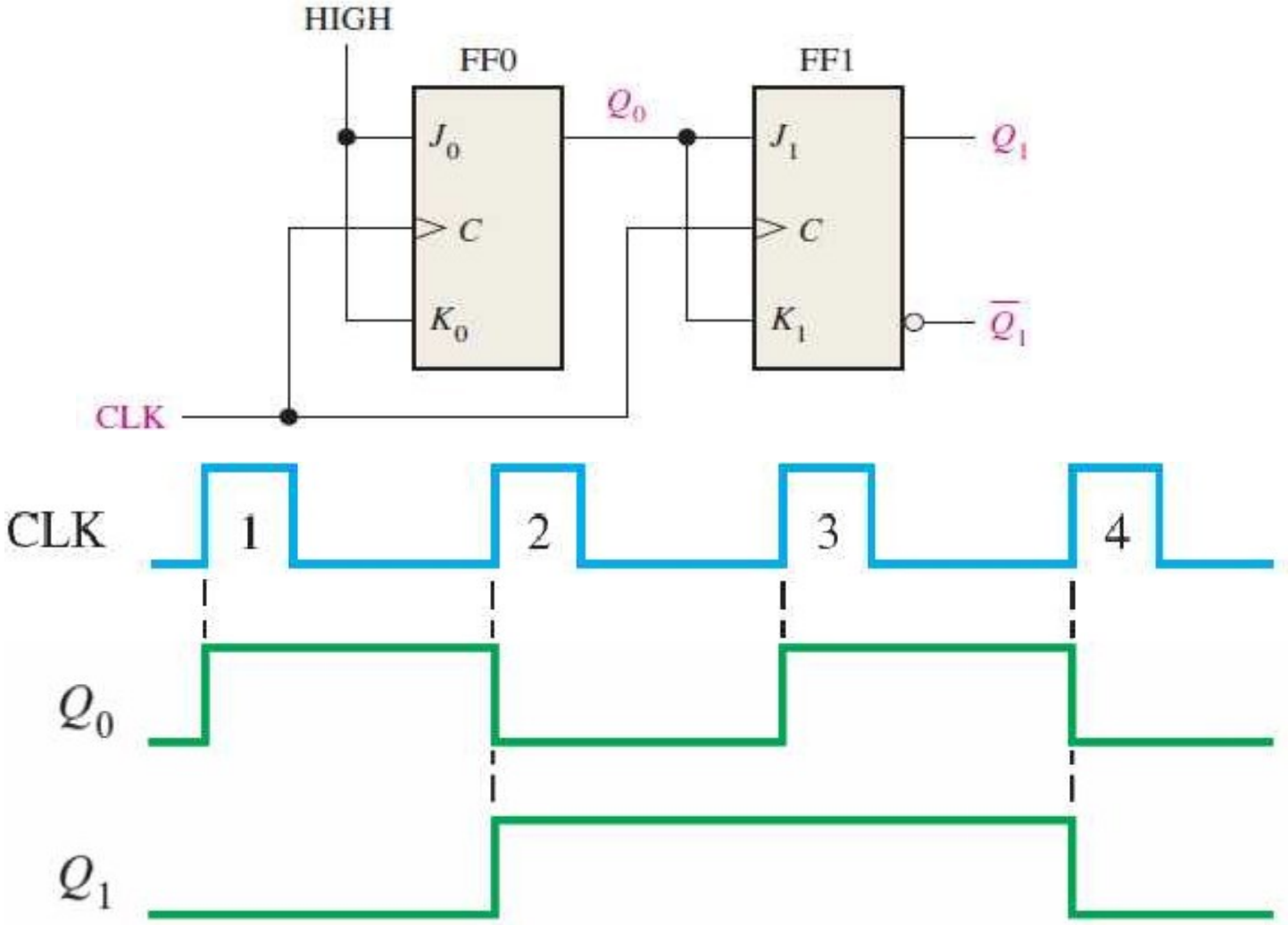
a) J-K flip-flop



b) D flip-flop

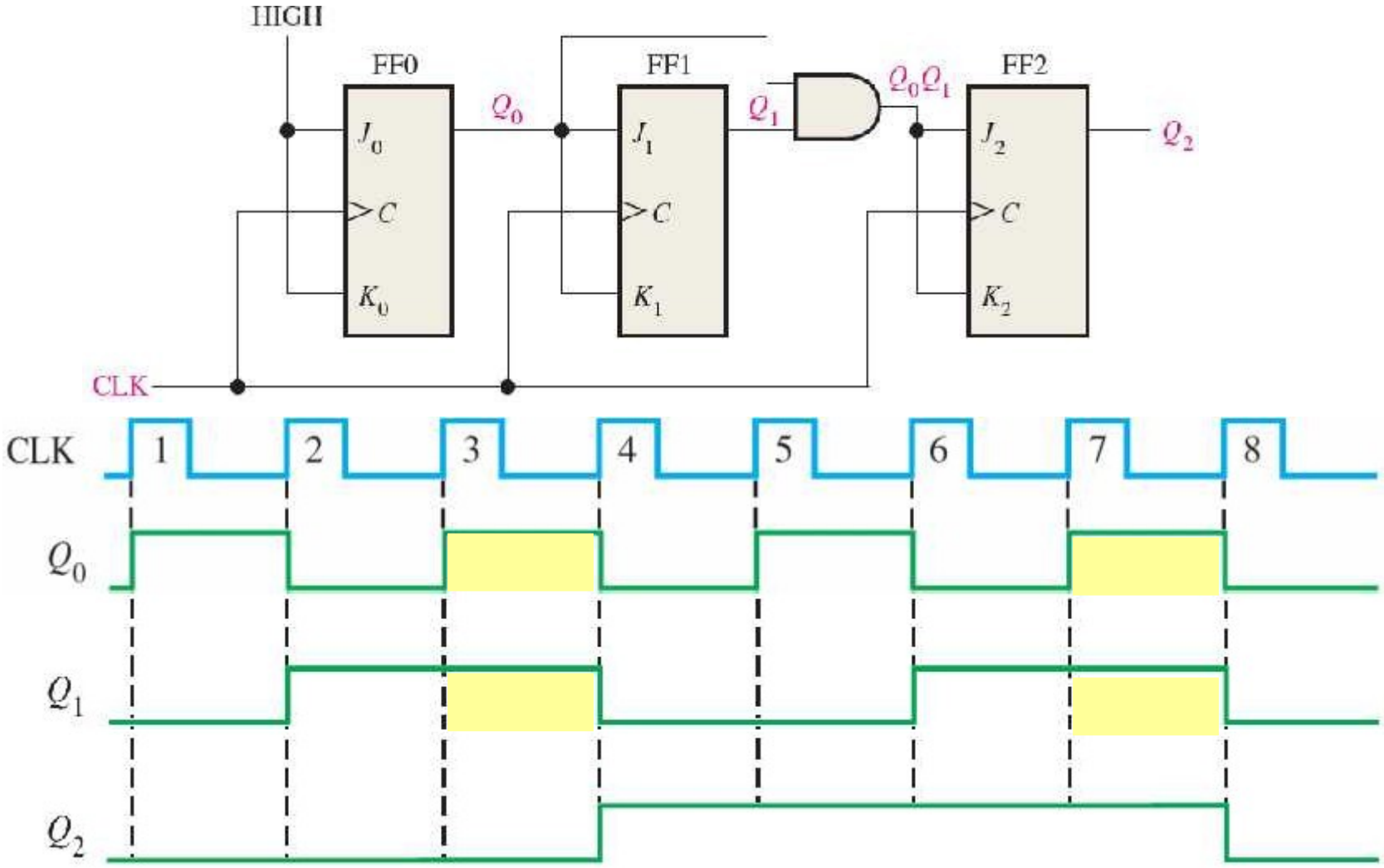
Synchronous Counters

A 2-Bit Synchronous Binary Counter using J-K flip-flop



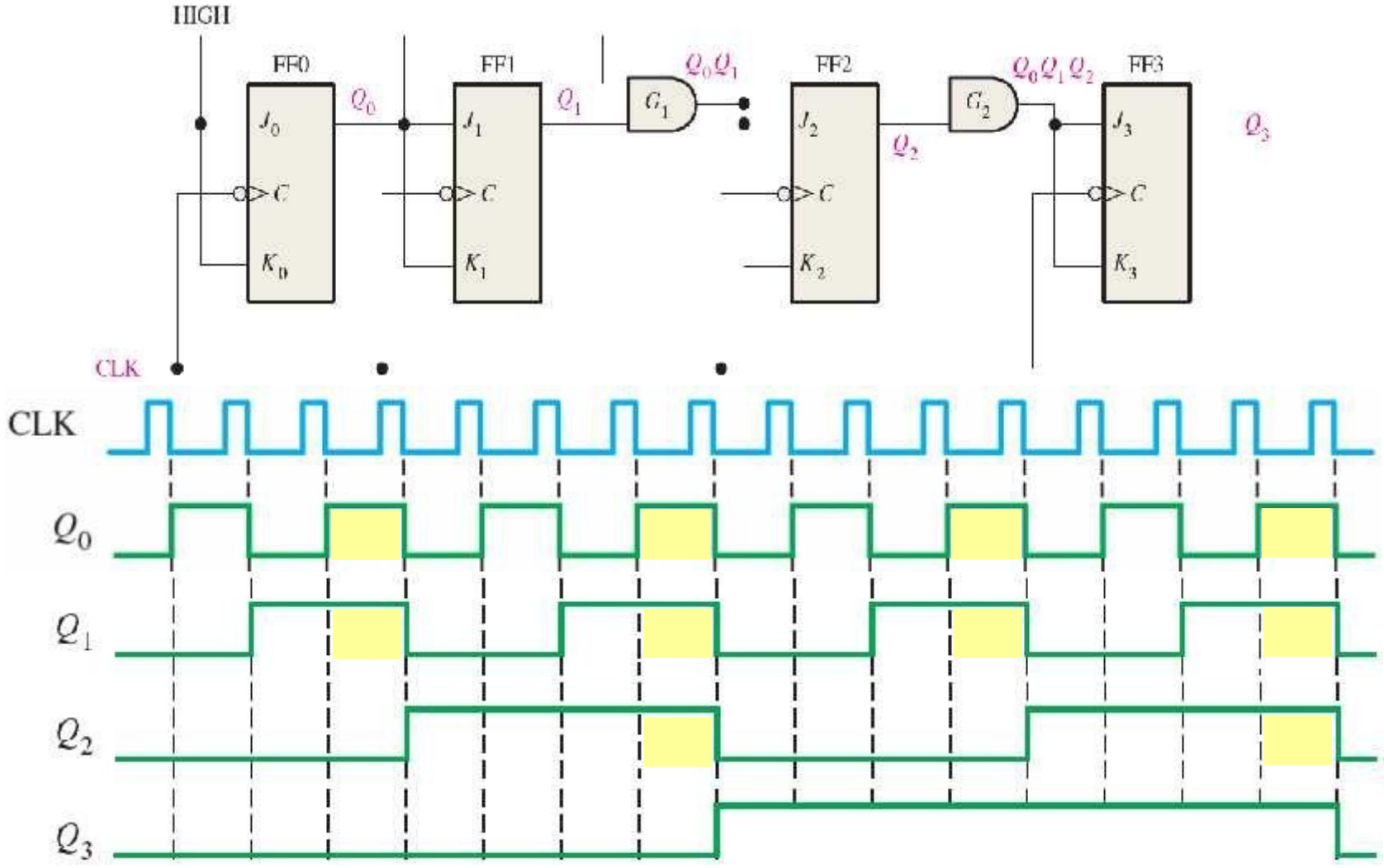
Synchronous Counters

A 3-Bit Synchronous Binary Counter using J-K flip-flop



Synchronous Counters

A 4-bit Synchronous Binary Counter using J-K flip-flop

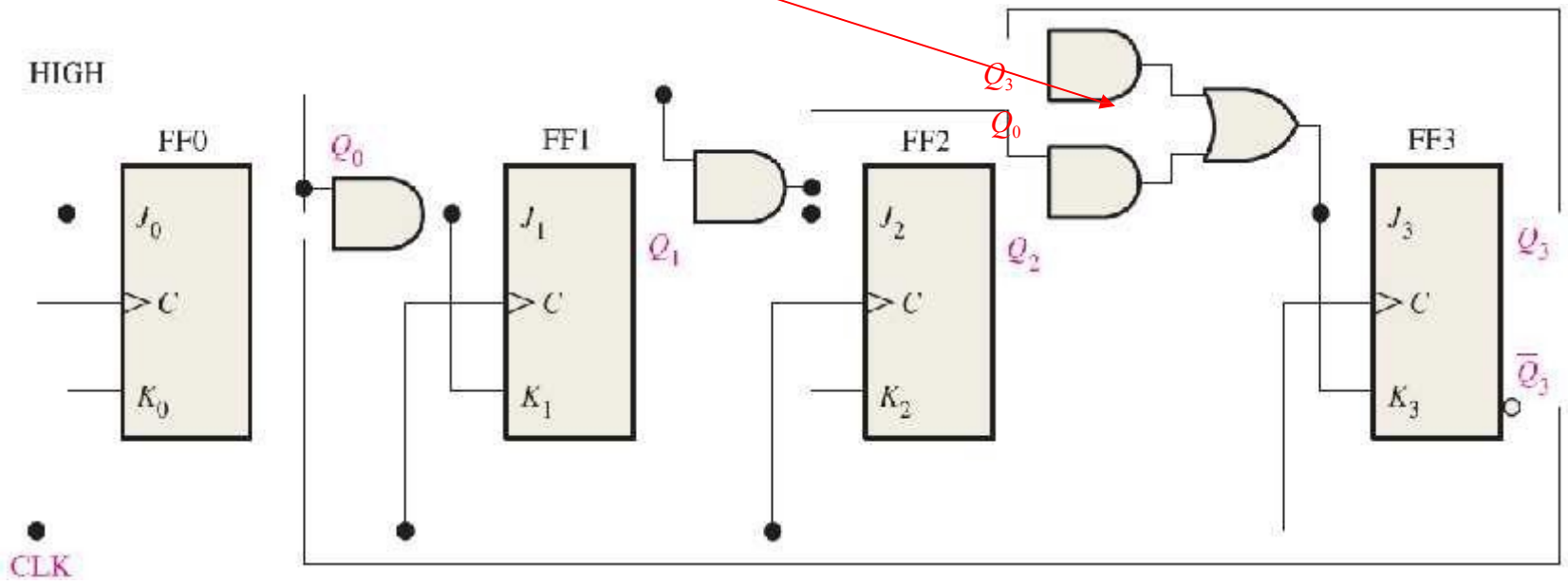


Synchronous Counters

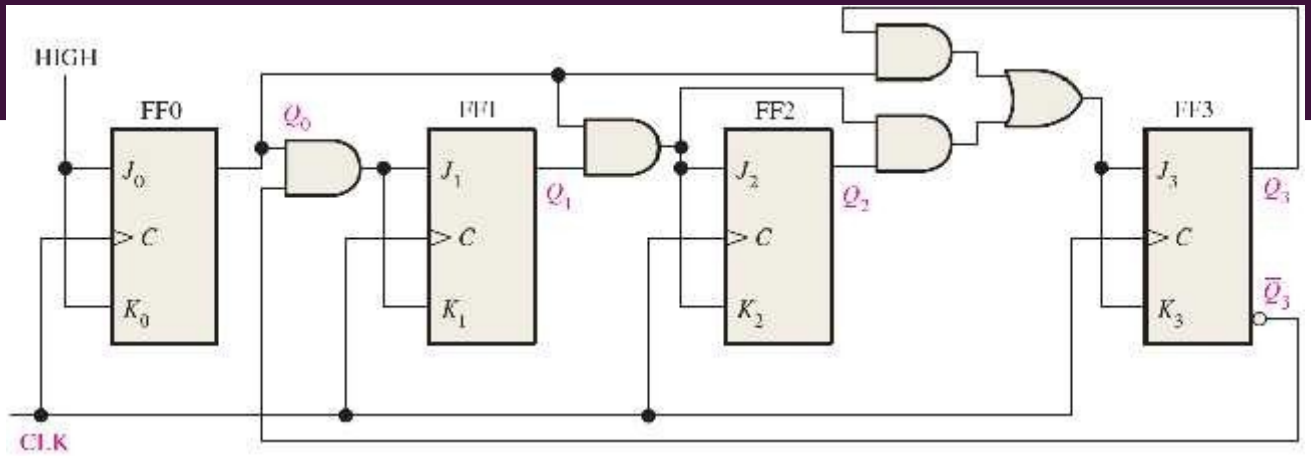
A 4-bit Synchronous Decade Counter

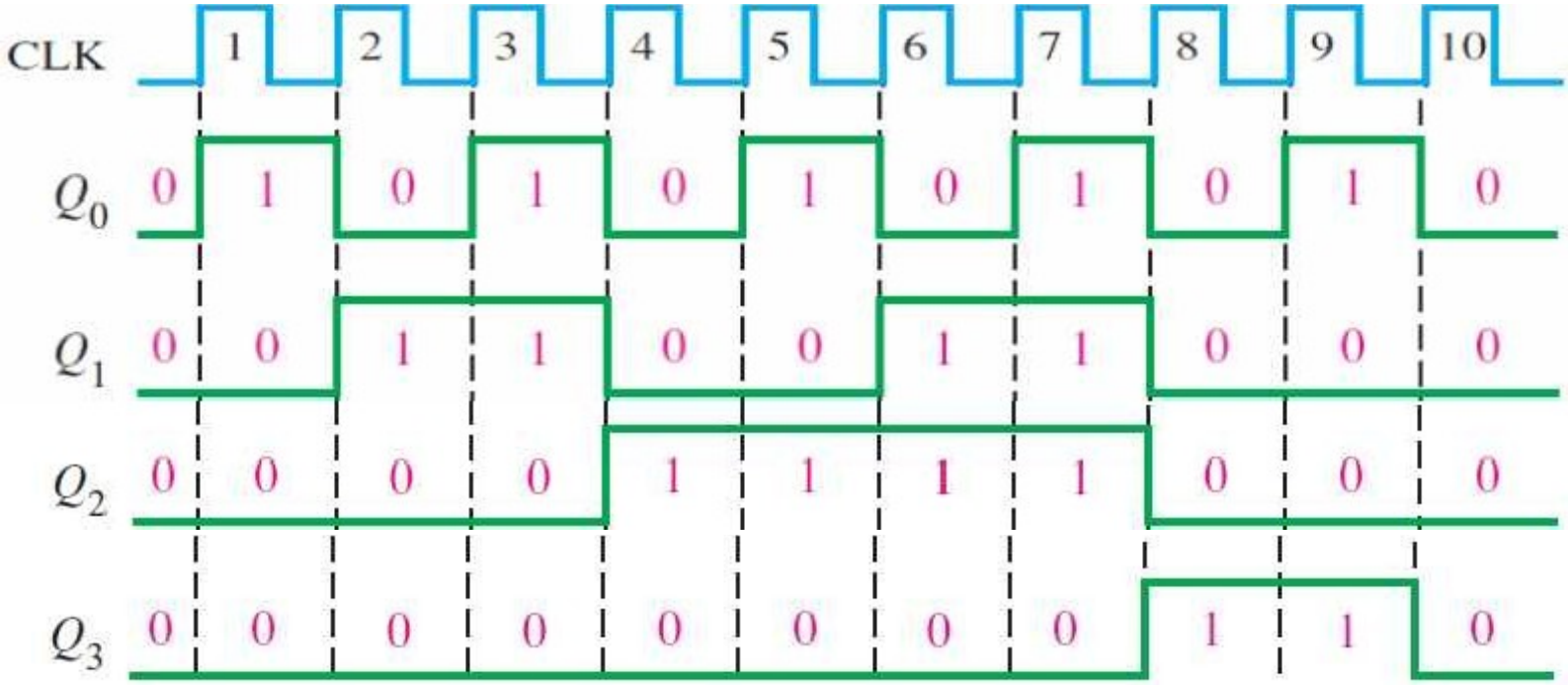
With some additional logic, a binary counter can be converted to a BCD synchronous decade counter. After reaching the count 1001, the counter recycles to 0000.

This gate detects 1001, and causes FF3 to toggle on the next clock pulse. FF0 toggles on every clock pulse. Thus, the count starts over at 0000.



Synchronous Counters

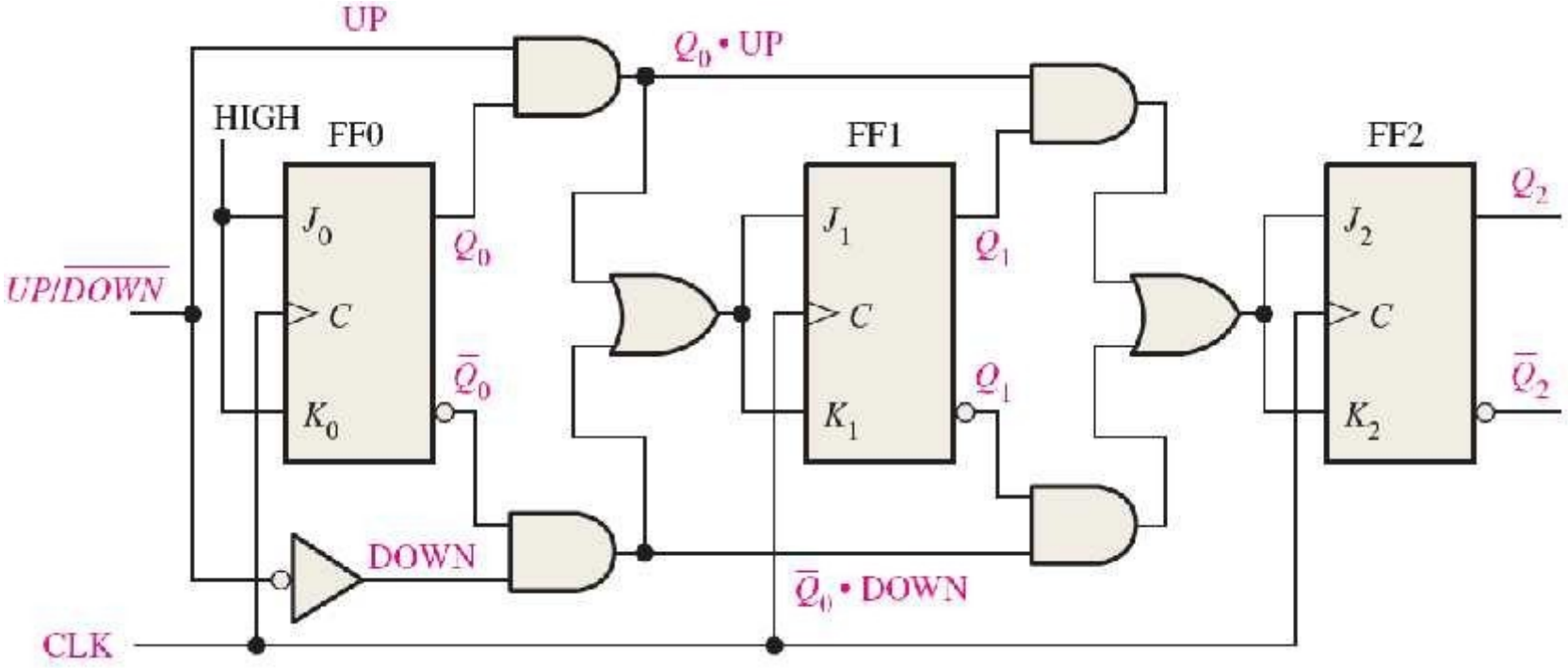




Synchronous Counters

Up/Down Synchronous Counters

UP/DOWN control input is HIGH for UP and LOW for DOWN.



A basic 3-bit up/down synchronous counter

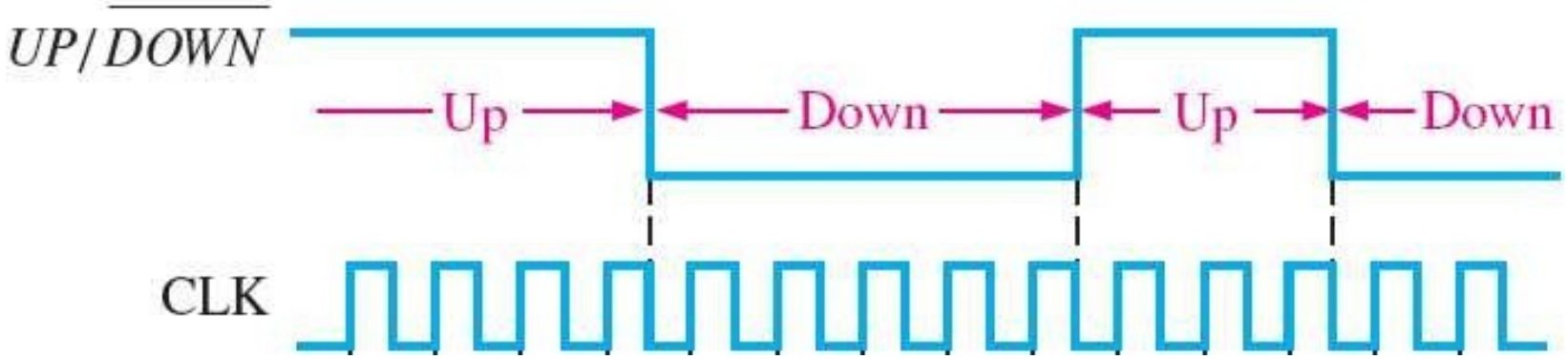
$$J_1 = K_1 = (Q_0 \cdot \text{UP}) + (\overline{Q}_0 \cdot \text{DOWN})$$

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

Synchronous Counters

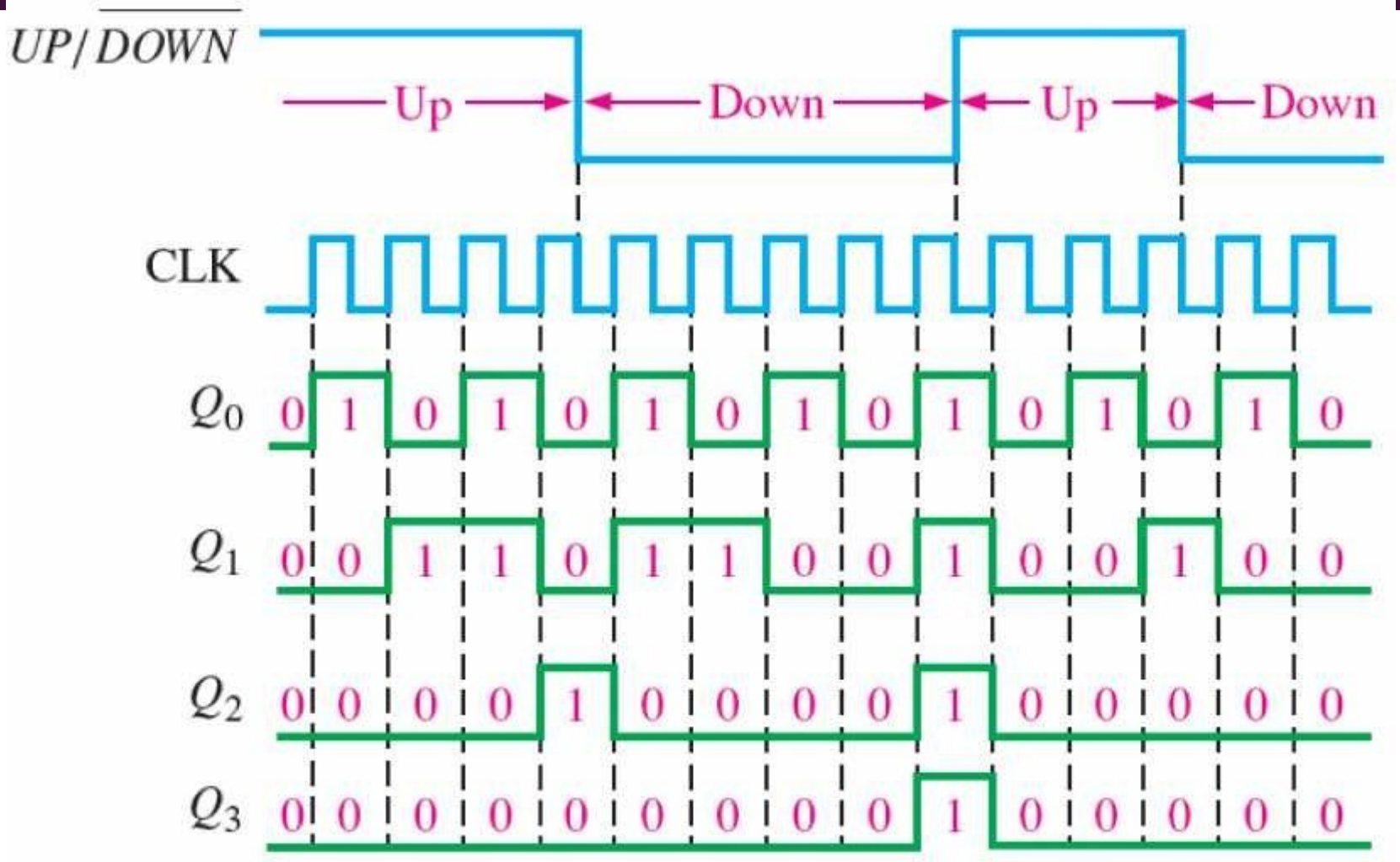
Example

Show the timing diagram and determine the sequence of a 4-bit synchronous binary up/DOWN counter if the clock and UP/DOWN control inputs have waveforms as shown in Figure below. The counter starts in the all-0s state and is positive edge-triggered.



Synchronous Counters

Solution



Design of Synchronous Counters

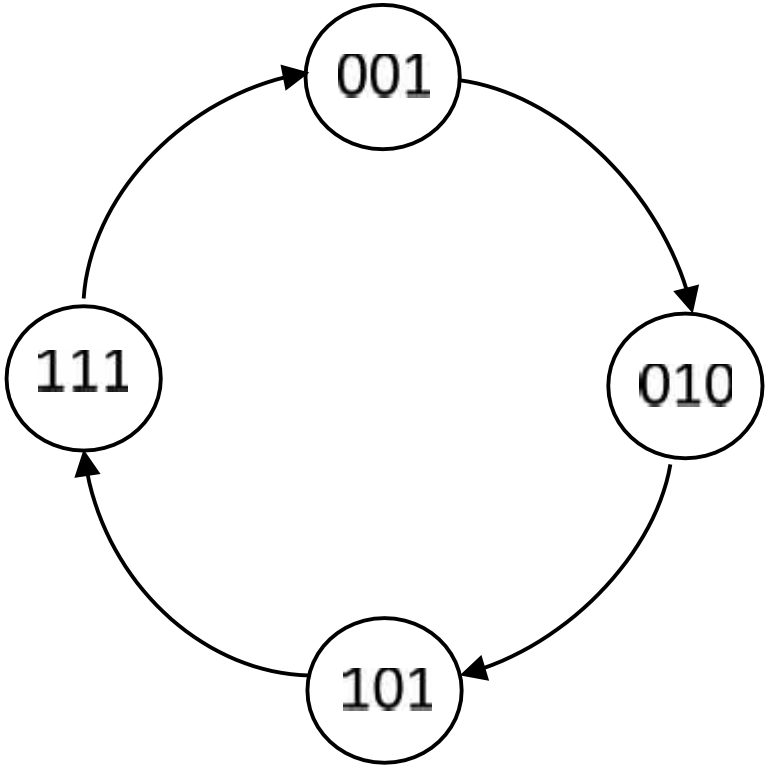
How to Design Synchronous Counters

- Step 1: State Diagram.
- Step 2: Next-State Table.
- Step 3: Flip-Flop Transition Table.
- Step 4: Karnaugh Maps.
- Step 5: Logic Expressions for Flip-Flop Inputs.
- Step 6: Counter Implementation

Design of Synchronous Counters

Example

Design a counter with the binary count sequence shown in the state diagram using D flip-flop.

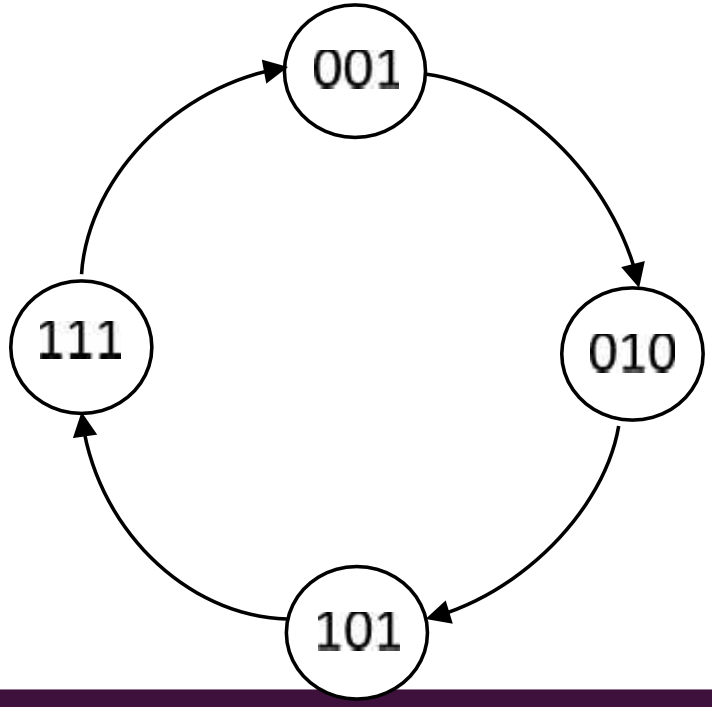


Design of Synchronous Counters

Solution

Step 1: State Diagram

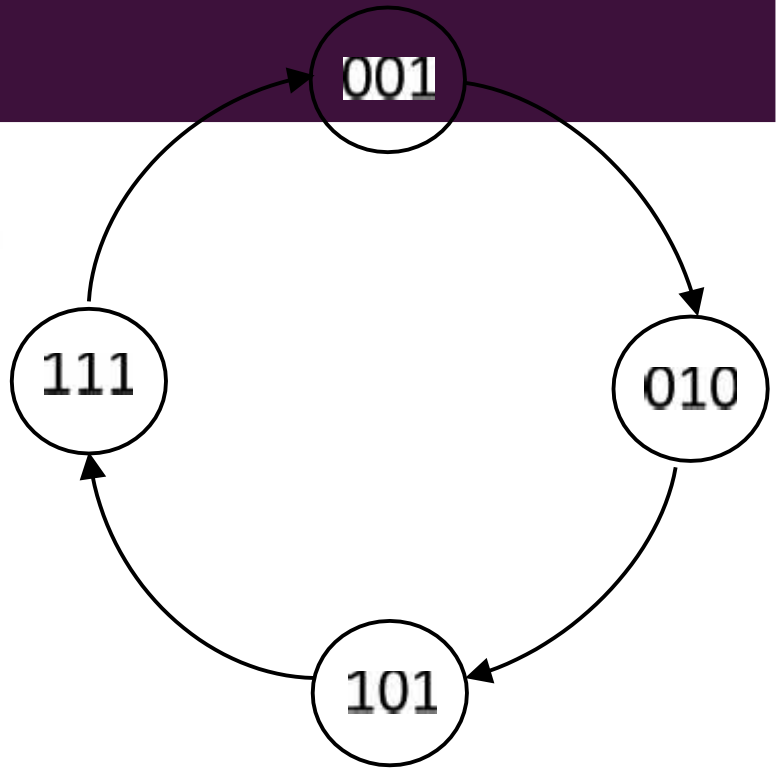
Although there are only four states, a 3-bit counter is required to implement this sequence because the maximum binary count is seven. Since the required sequence does not include all the possible binary states, the invalid states (0, 3, 4, and 6) can be treated as "don't cares" in the design. However, if the counter should erroneously get into an invalid state, you must make sure that it goes back to a valid state.



Design of Synchronous Counters

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



Design of Synchronous Counters

Step 2: Flip-Flop Transition State

Transition table for a D flip-flop

Output Transitions			Flip-Flop Input
Q_N		Q_{N+1}	D
0	→	0	0
0	→	1	1
1	→	0	0
1	→	1	1

Design of Synchronous Counters

D Flip-Flop Input

Present State			Next State			Flip-Flop Input		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	1	0	1	0	0	1	0
0	1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	1	1
1	1	1	0	0	1	0	0	1

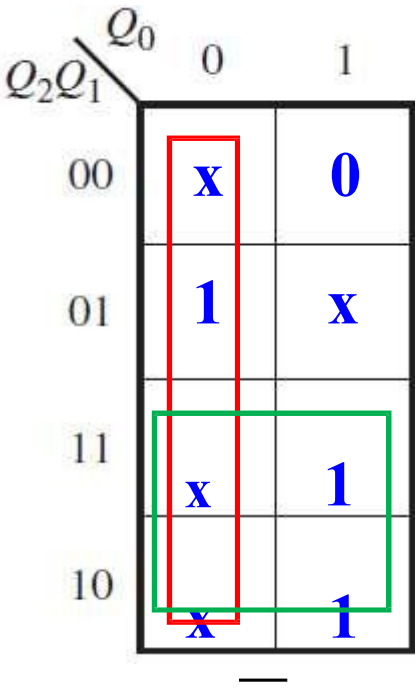
Design of Synchronous Counters

Step 4: Karnaugh Maps

(0, 3, 4, and 6) = X (don't cares)

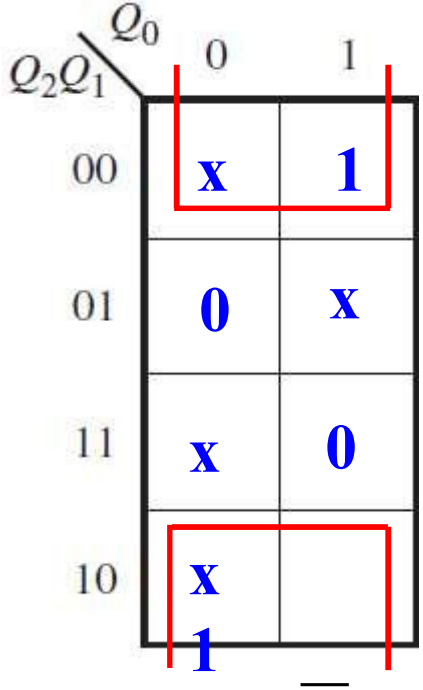
Present State			Next State			Flip-Flop Input		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	1	0	1	0	0	1	0
0	1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	1	1
1	1	1	0	0	1	0	0	1

D_0 map:



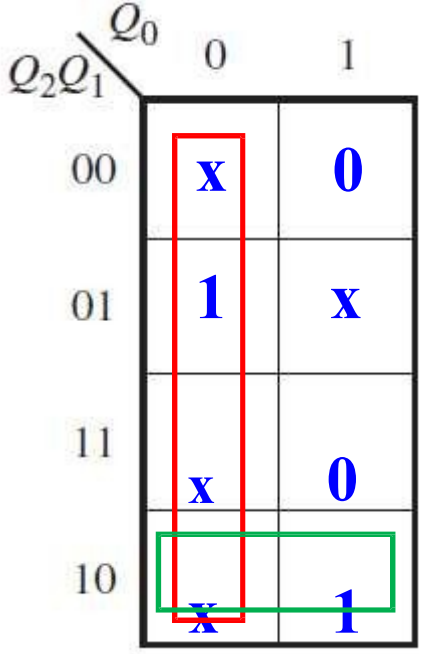
$$D_0 = Q_0 + \overline{Q_2}$$

D_1 map:



$$D_1 = Q_1$$

D_2 map:



$$D_2 = Q_0 + Q_2\overline{Q_1}$$

Design of Synchronous Counters

Step 5: Logic Expression for Flip-Flop Inputs

$$D_0 = \overline{Q}_0 +$$

$$Q_2\, D_1 = Q_1$$

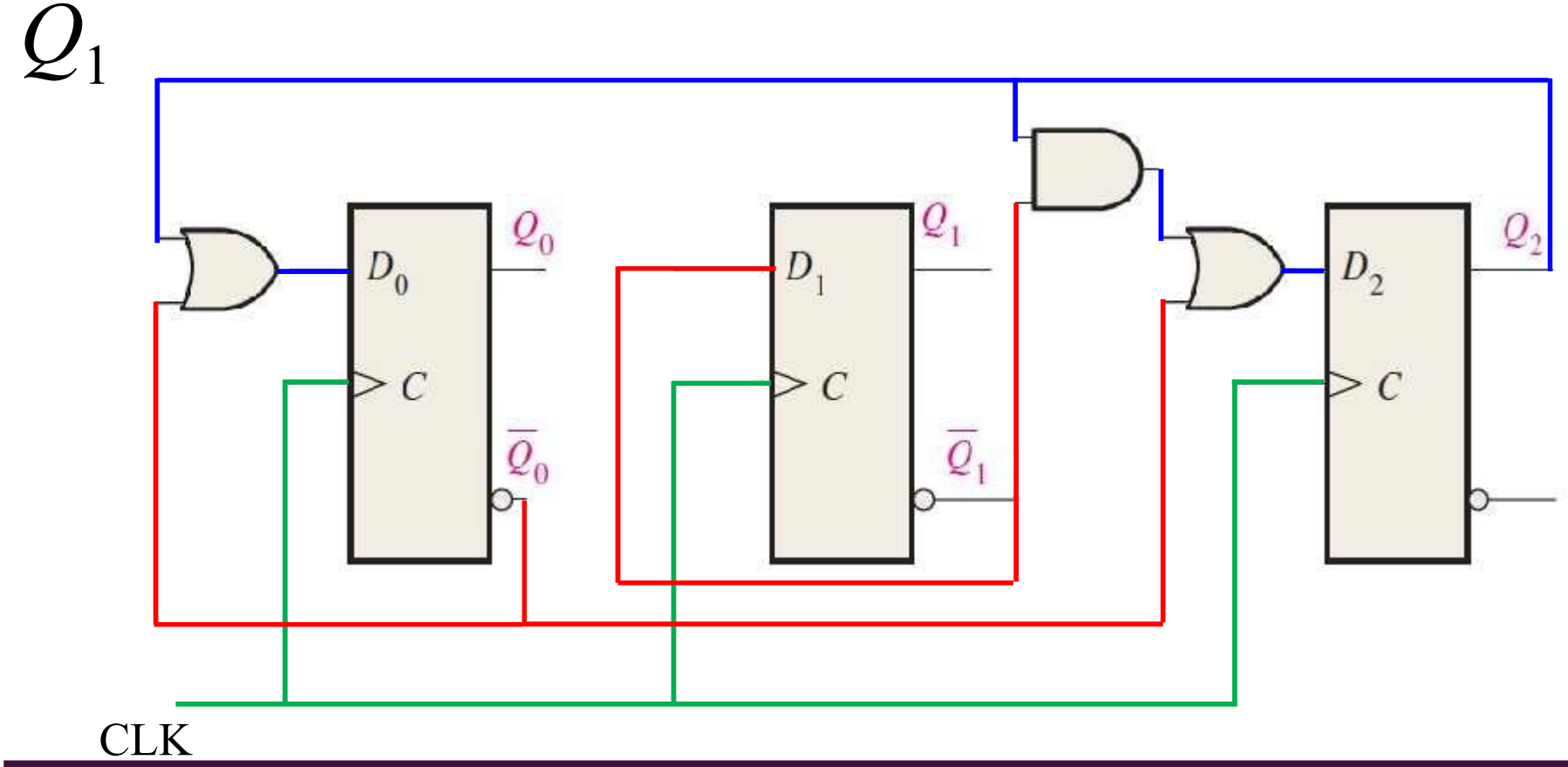
$$D_2 = Q_0 +$$

$$Q_2\, Q_1$$

Design of Synchronous Counters

Step 6: Counter Implementation

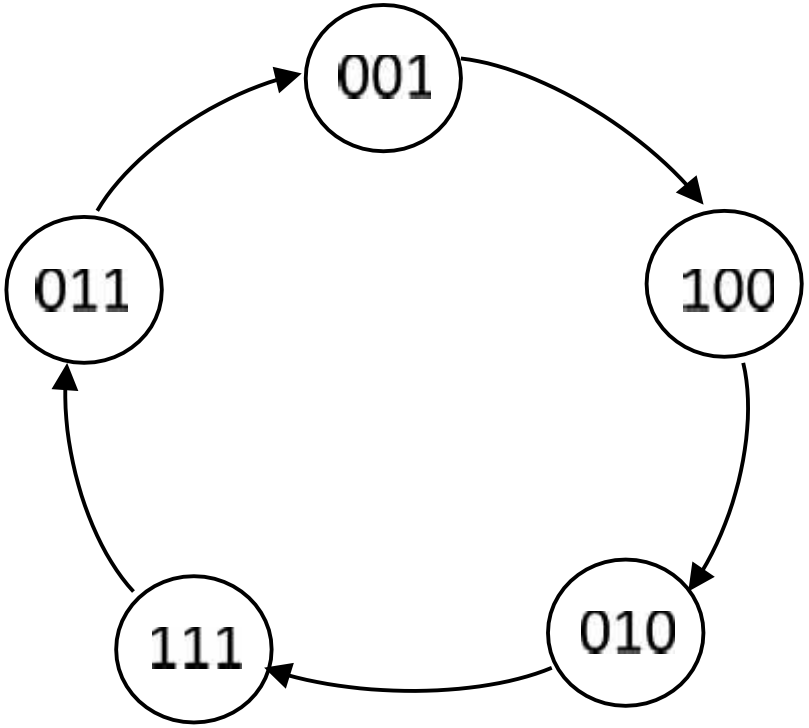
$$D_0 = \overline{Q}_0 + Q_2 \qquad D_1 = \overline{Q}_1 \qquad D_2 = \overline{Q}_0 + Q_2$$



Design of Synchronous Counters

Example

Design a synchronous counter with a state diagram as shown in figure below, for your design, you are required to use J-K flip-flops.

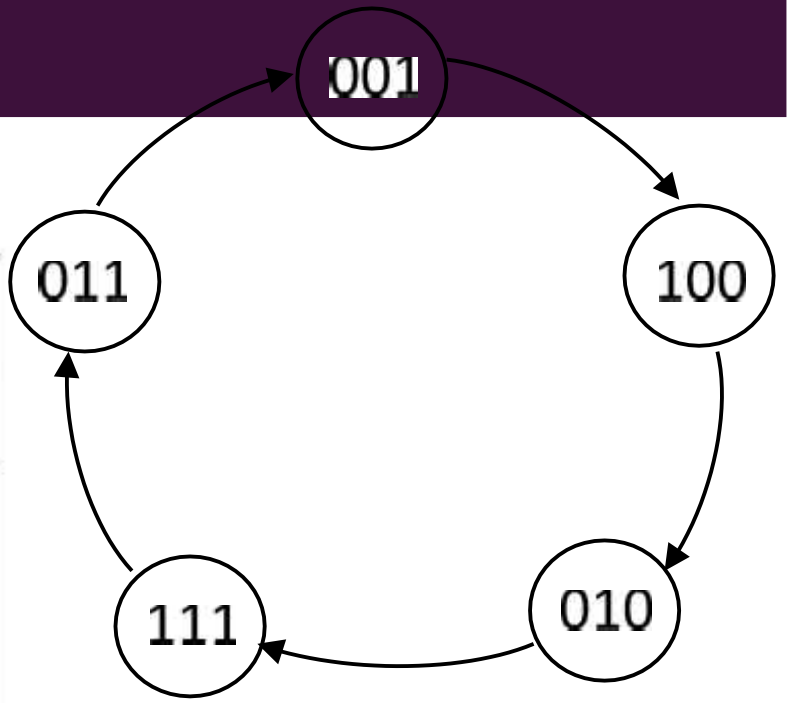


Design of Synchronous Counters

Solution

Step 2: Next-State Table

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



Design of Synchronous Counters

Step 3: Flip-Flop Transition Table

Transition table for a J-K flip-flop

Output Transitions				
Q_N		Q_{N+1}	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"

Design of Synchronous Counters

Step 3: Flip-Flop Transition Table

Present State			Next State			Flip-Flop Inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	1	1	0	0	1	x	0	x	x	1
1	0	0	0	1	0	x	1	1	x	0	x
0	1	0	1	1	0	1	x	x	0	1	x
1	1	1	0	1	1	x	1	x	0	x	0
0	1	1	0	1	1	0	x	x	1	x	x
			0	0	0				0		

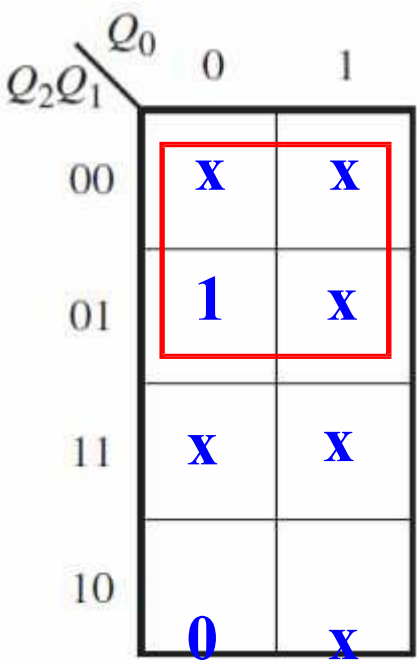
Design of Synchronous Counters

Step 4: Karnaugh Maps

(0, 5 and 6) = X (don't cares)

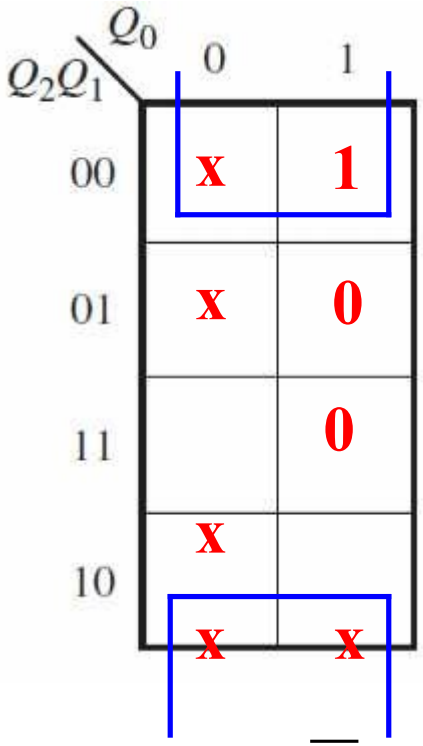
Present State			Next State			Flip-Flop Inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	1	1	0	0	1	x	0	x	x	1
1	0	0	0	1	0	x	1	1	x	0	x
0	1	0	1	1	1	1	x	x	0	1	x
1	1	1	0	1	1	x	1	x	0	x	0
0	1	1	0	0	1	0	x	x	1	x	0

J_0 map:



$$J_0 = \overline{Q}_2$$

K_0 map:



$$K_0 = \overline{Q}_1$$

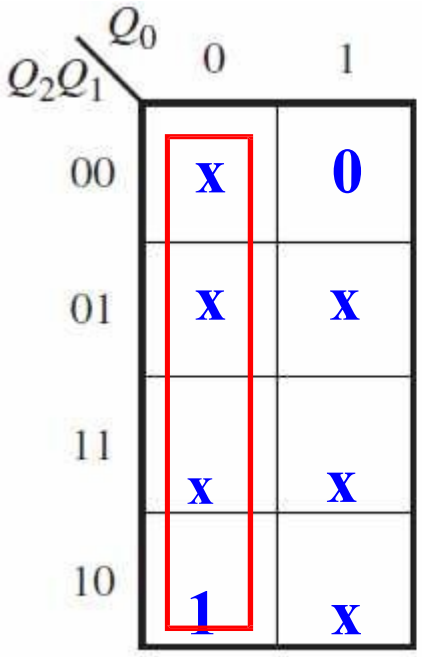
Design of Synchronous Counters

Step 4: Karnaugh Maps

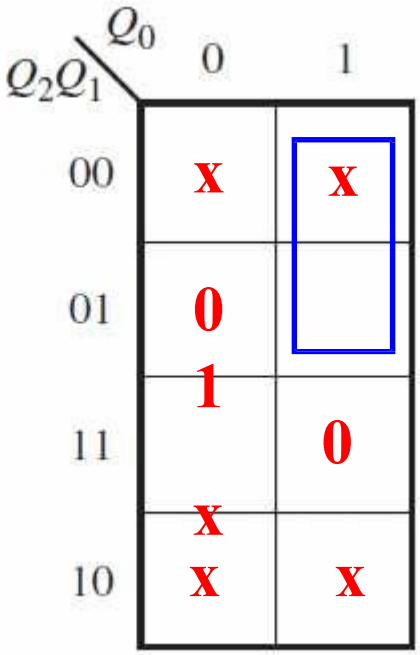
(0, 5 and 6) = X (don't cares)

Present State			Next State			Flip-Flop Inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	1	1	0	0	1	x	0	x	x	1
1	0	0	0	1	0	x	1	1	x	0	x
0	1	0	1	1	1	1	x	x	0	1	x
1	1	1	0	1	1	x	1	x	0	x	0
0	1	1	0	0	1	0	x	x	1	x	0

J_1 map:



K_1 map:



$$J_1 = \overline{Q}_0$$

$$K_1 = \overline{Q}_2 Q_0$$

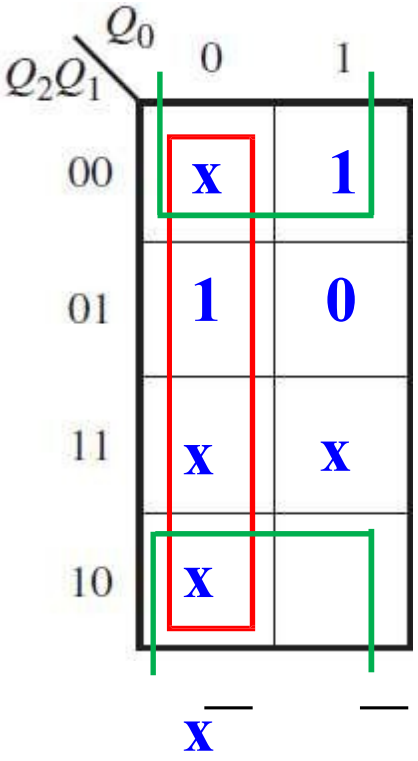
Design of Synchronous Counters

Step 4: Karnaugh Maps

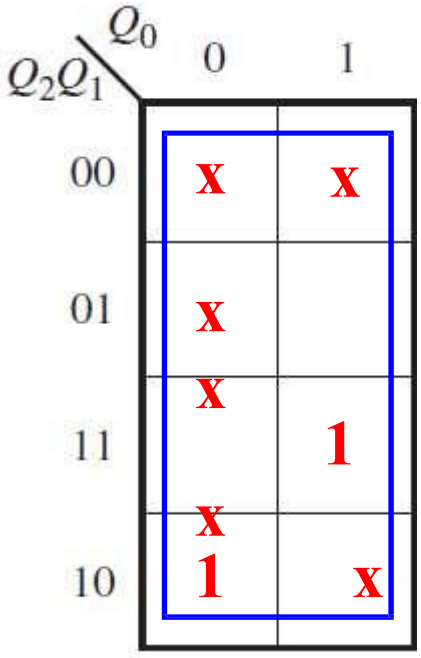
(0, 5 and 6) = X (don't cares)

Present State			Next State			Flip-Flop Inputs					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	1	1	0	0	1	x	0	x	x	1
1	0	0	0	1	0	x	1	1	x	0	x
0	1	0	1	1	1	1	x	x	0	1	x
1	1	1	0	1	1	x	1	x	0	x	0
0	1	1	0	0	1	0	x	x	1	x	0

J_2 map:



K_2 map:



$$K_2 = 1$$

$$J_2 = Q_0 + Q_1$$

Design of Synchronous Counters

Step 5: Logic Expressions for Flip-Flop Inputs

$$J_0 = \overline{Q_2}$$

$$K_0 = \overline{Q_1}$$

$$J_1 = Q_0$$

$$K_1 = \overline{Q_0}$$

$$Q_2$$

$$J_2 = Q_0 + Q_1$$

$$K_2 = 1$$

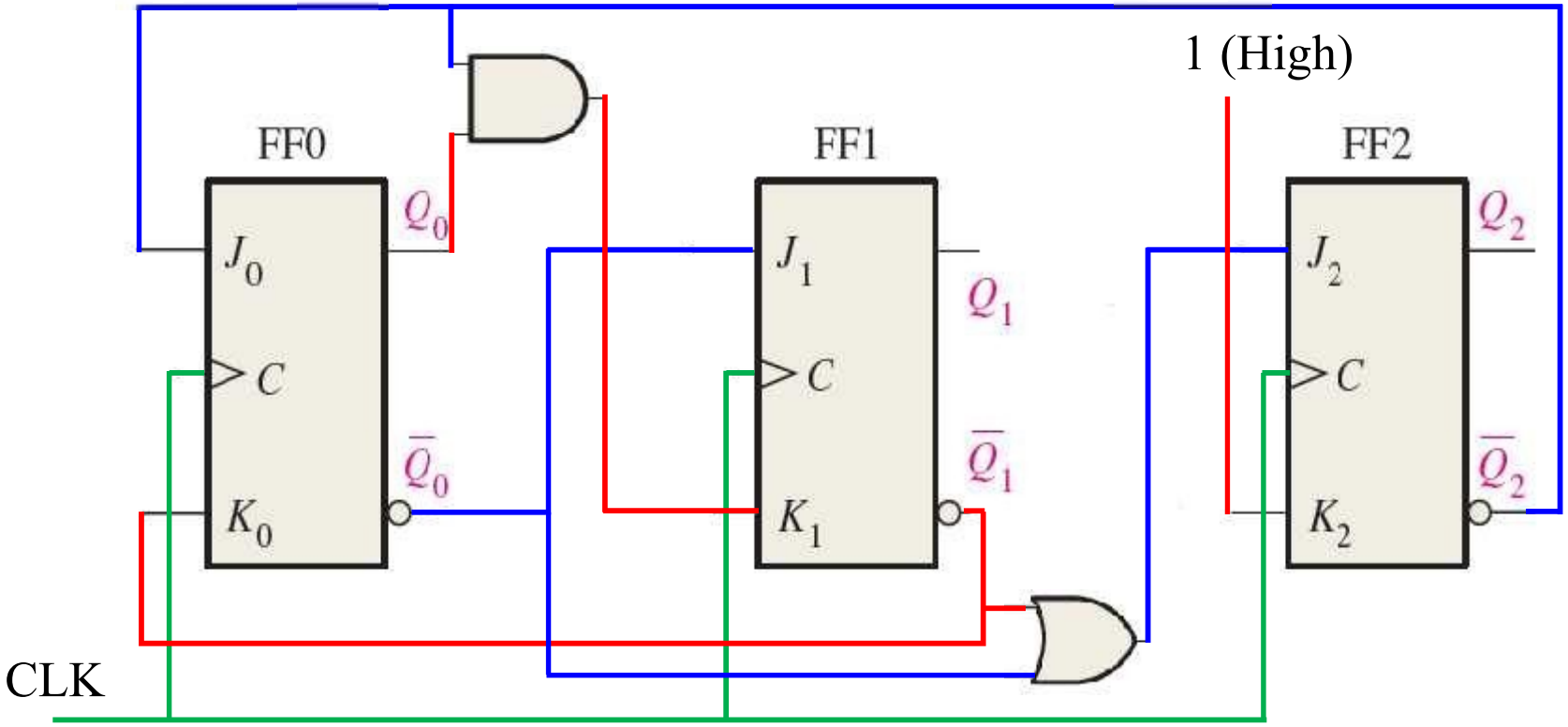
Design of Synchronous Counters

Step 6: Counter Implementation

$$J_0 = \overline{Q}_2 \qquad K_0 = \overline{Q}_1$$

$$J_1 = \overline{Q}_0 \qquad K_1 = Q_0\overline{Q}_2$$

$$J_2 = \overline{Q}_0 + \overline{Q}_1 \qquad K_2 = 1$$

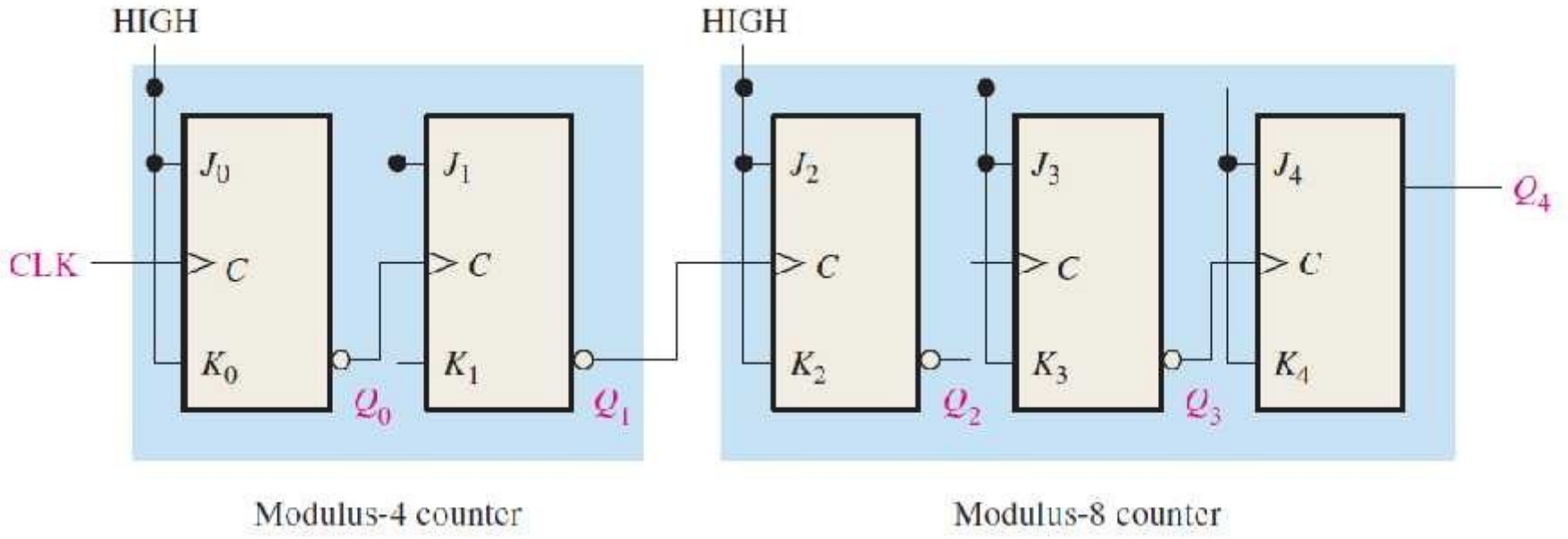


Cascaded Counters

Introduction

- Counters can be connected in cascade to achieve higher-modulus operation. **Cascading means that the last-stage output of one counter drives the input of the next counter**.

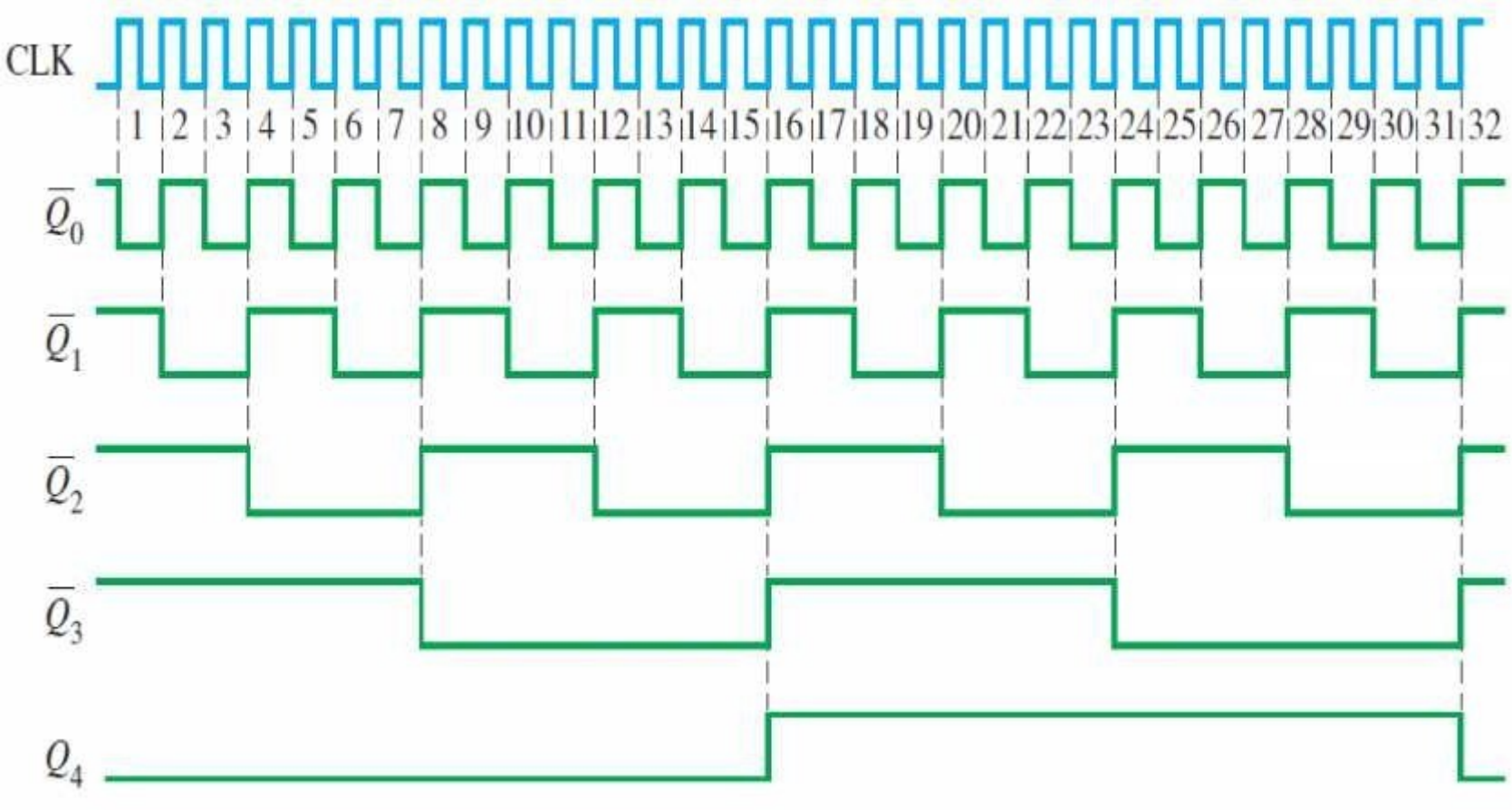
Asynchronous Cascading



Two cascaded asynchronous counters (all *J and K inputs are HIGH)*

Cascaded Counters

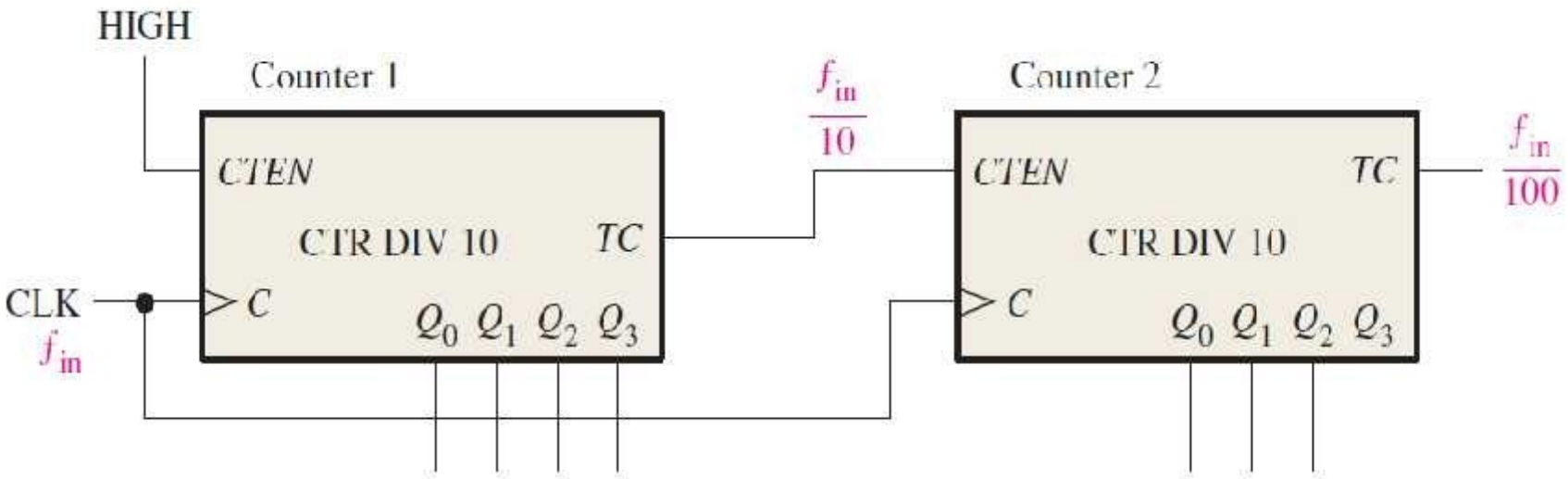
- The overall modulus of the two cascaded counters is 4 x 8 = 32



Synchronous Cascading

- When operating synchronous counters in a cascaded configuration, it is necessary to use the count enable and the terminal count functions to achieve higher-modulus operation.

- On some devices the count enable is labeled simply CTEN and terminal count (TC).

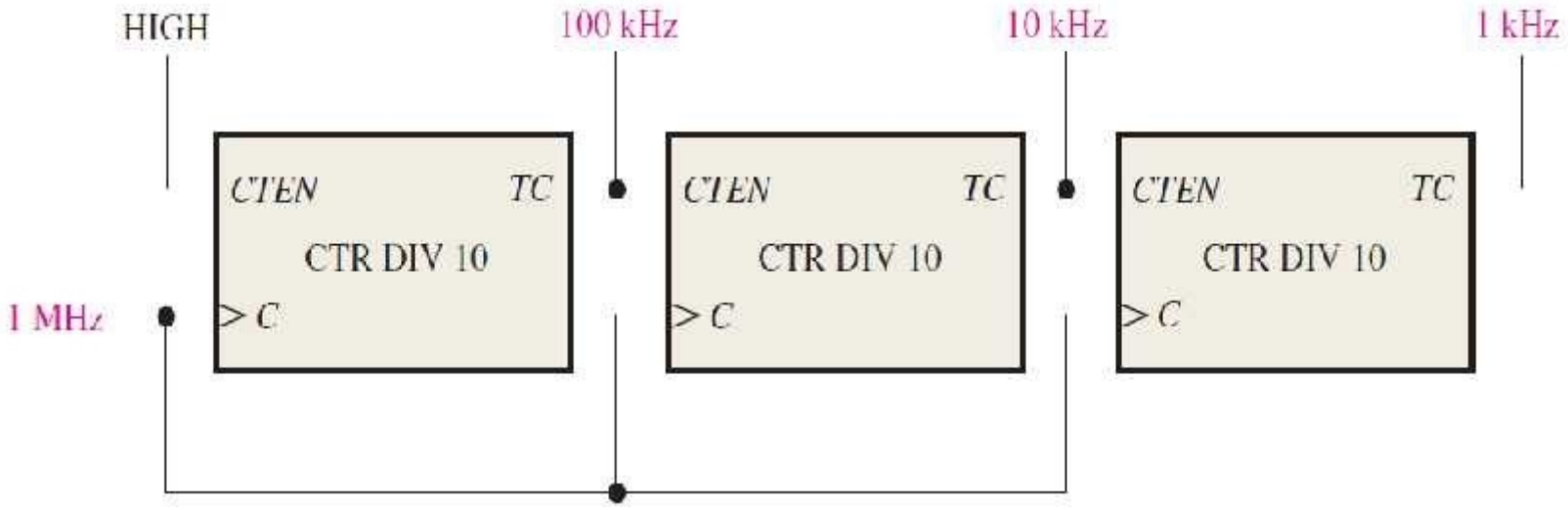


The overall modulus of these two cascaded counters is 10 × 10 = 100

Synchronous Cascading

Example

Suppose that you have a basic clock frequency of 1 MHz and you wish to obtain 100 kHz, 10 kHz, and 1 kHz; a series of cascaded decade counters can be used. If the 1 MHz signal is divided by 10, the output is 100 kHz. Then if the 100 kHz signal is divided by 10, the output is 10 kHz. Another division by 10 produces the 1 kHz frequency.

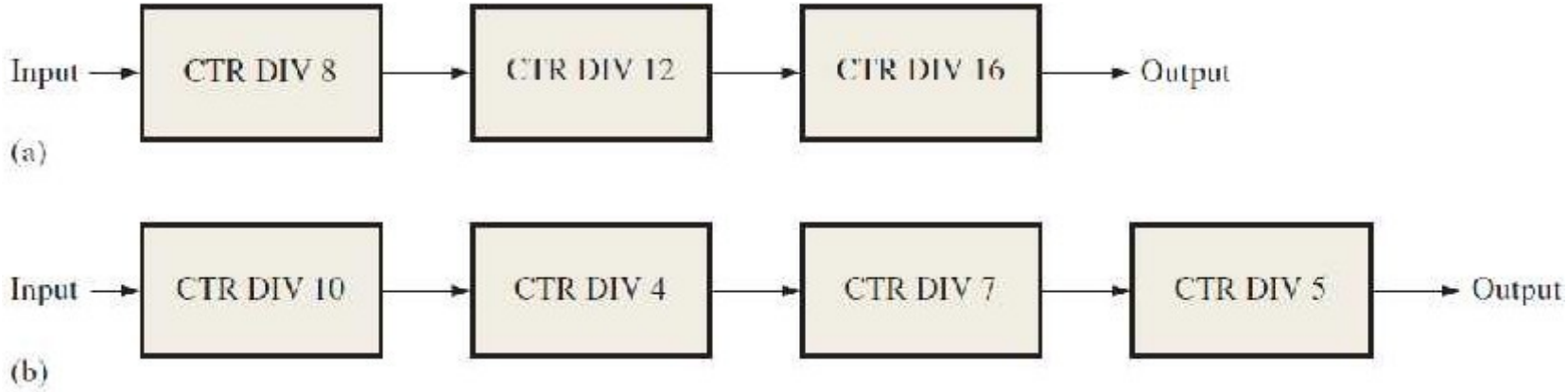


Three cascaded decade counters forming a divide-by-1000 frequency divider with intermediate divide-by-10 and divide-by-100 outputs

Synchronous Cascading

Example

Determine the overall modulus of the two cascaded counter configurations shown in Figure (a) and Figure (b)



Solution

In Figure (a), the overall modulus for the 3-counter configuration is

$$8 \times 12 \times 16 = \mathbf{1536}$$

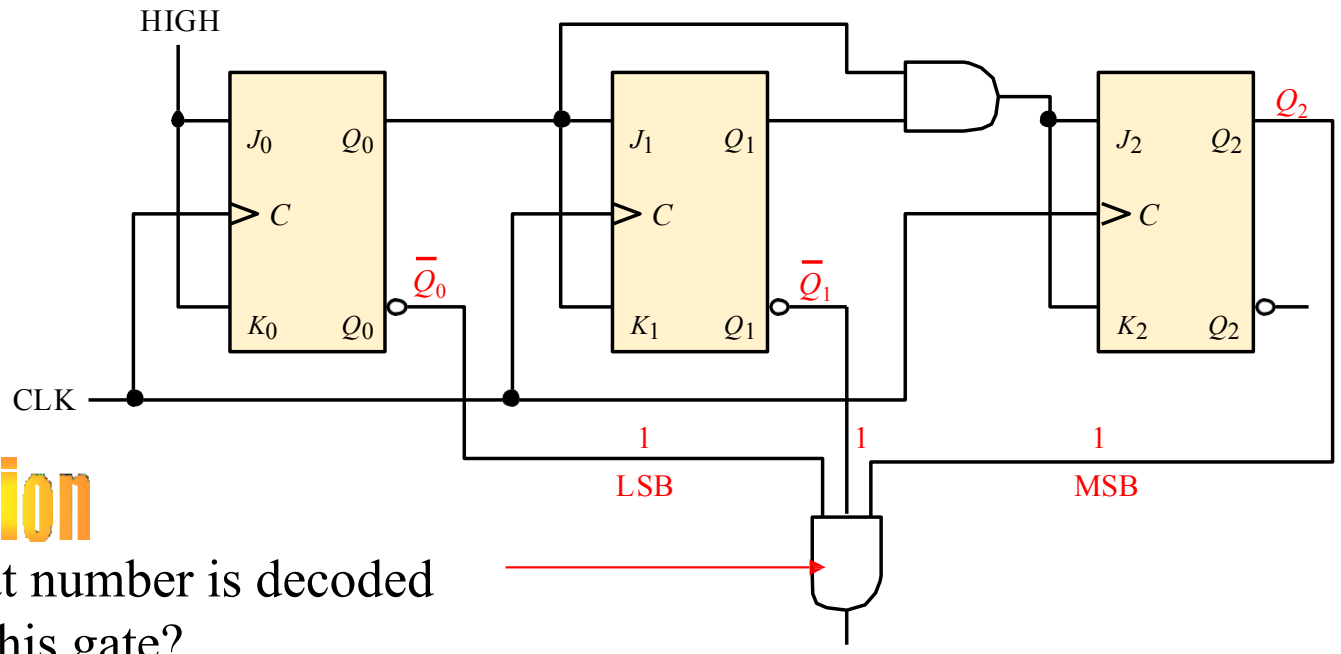
In Figure (b), the overall modulus for the 4-counter configuration is

$$10 \times 4 \times 7 \times 5 = \mathbf{1400}$$

Counter Decoding

Counter Decoding

Decoding is the detection of a binary number and can be done with an AND gate.



Question

What number is decoded by this gate?