Finite State Machines (II)

- Synchronous Counters
- Ring Counters
- Ripple Counters

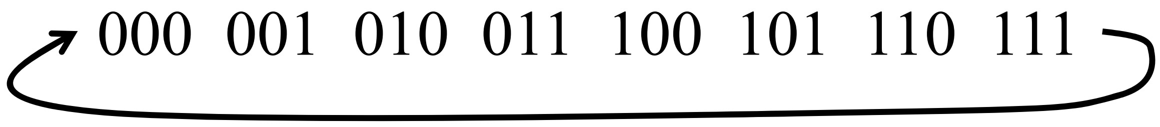
Binary Counters

A counter is a digital circuit which goes through a prescribed sequence of states in response to an input signal.

The count state is stored in flip-flops.

The modulus (mod) of the counter is the number of states that the counter cycles through before returning to its original state.

A mod-8 counter will count from 000 to 111 in sequence before returning to 000.

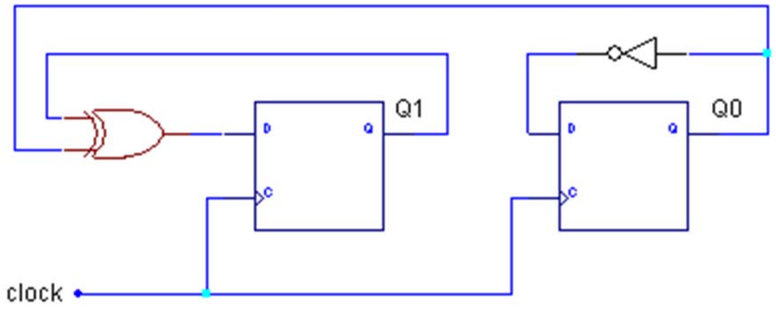


000 001 010 011 100 101 110 111

An n-bit binary counter consists of n flip-flops and can count in binary from 0 to 2^n- 1. This is known as a mod-2^n counter and all 2^n states are produced.

For a circuit with more than one flip-flop, its ‘state’ at a given time is the output of the flip-flops at that time.

Consider the circuit shown which has two flip-flops. The state is given by Q_1,Q_0.



		Q_1	Q_0
initial state	→	0	0
first clock edge	→	0	1
second clock edge	→	1	0
third clock edge	→	1	1
fourth clock edge	→	0	0

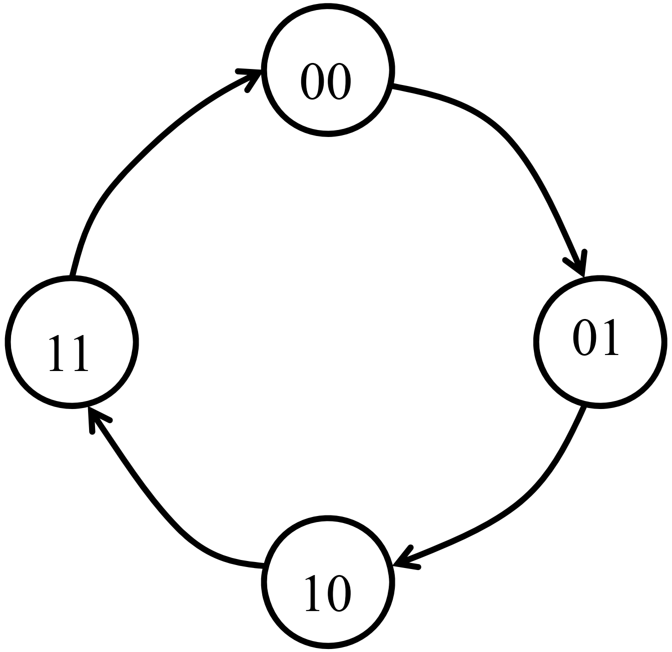
$$Q_0\,(t+1) = \overline{Q_0}$$

$$Q_1\,(t+1) = Q_0 \oplus Q_1$$

If the circuit has n flip-flops, it can have at most 2^n different states.

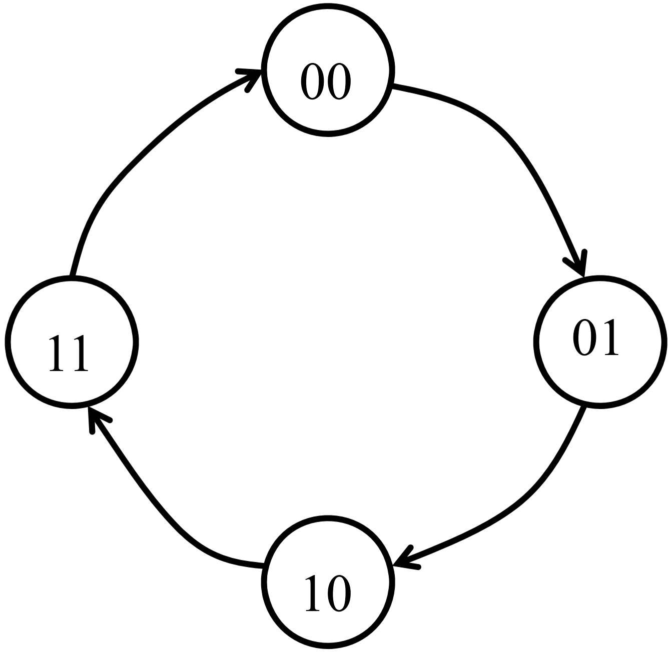
Design of a 2-bit up-counter

1. State diagram



Design of a 2-bit up-counter

1. State diagram

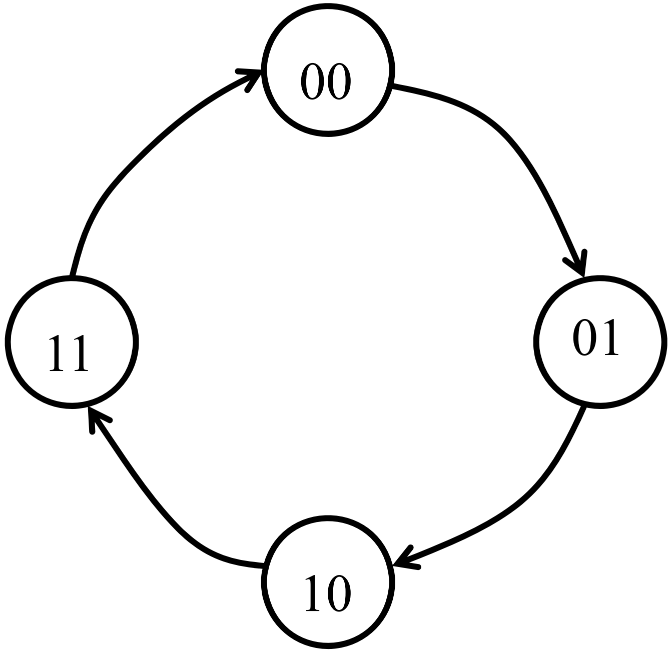


2. State table

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

Design of a 2-bit up-counter

1. State diagram

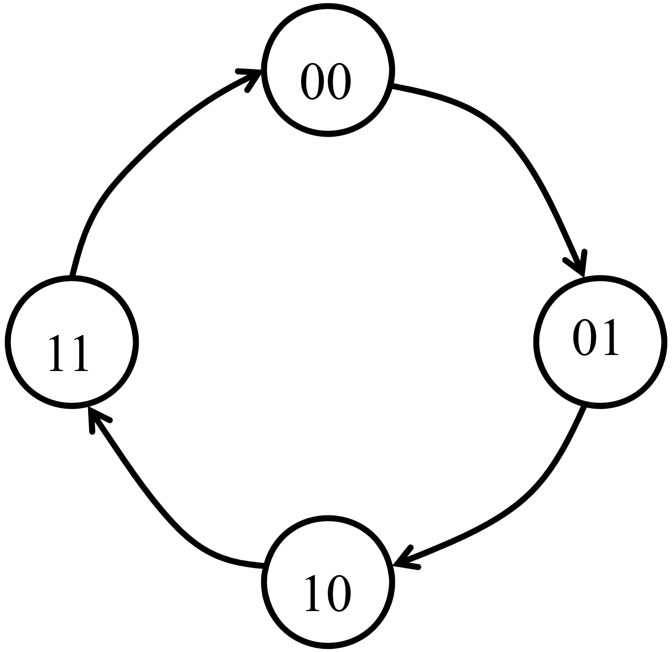


2. State table

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

Design of a 2-bit up-counter

1. State diagram

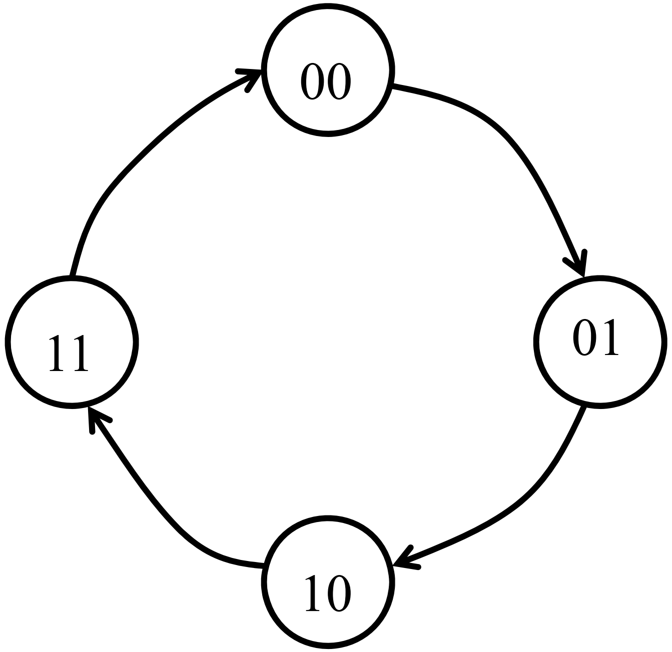


2. State table

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

Design of a 2-bit up-counter

1. State diagram

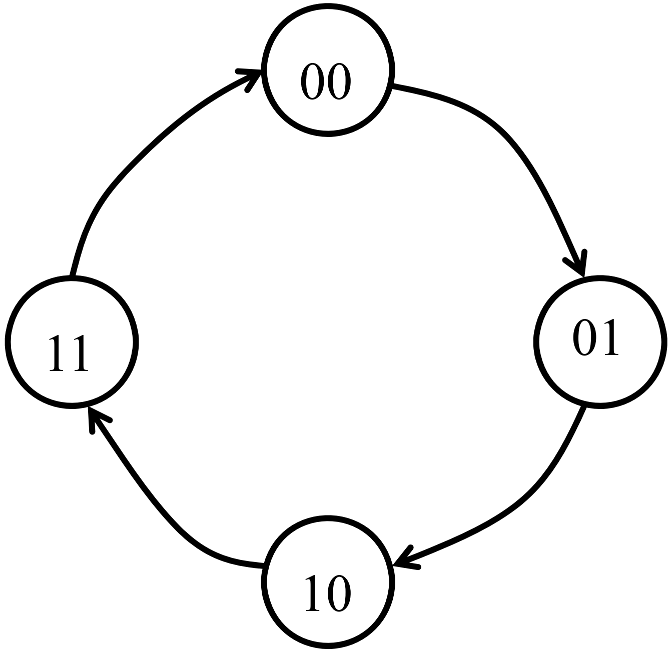


2. State table

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

Design of a 2-bit up-counter

1. State diagram

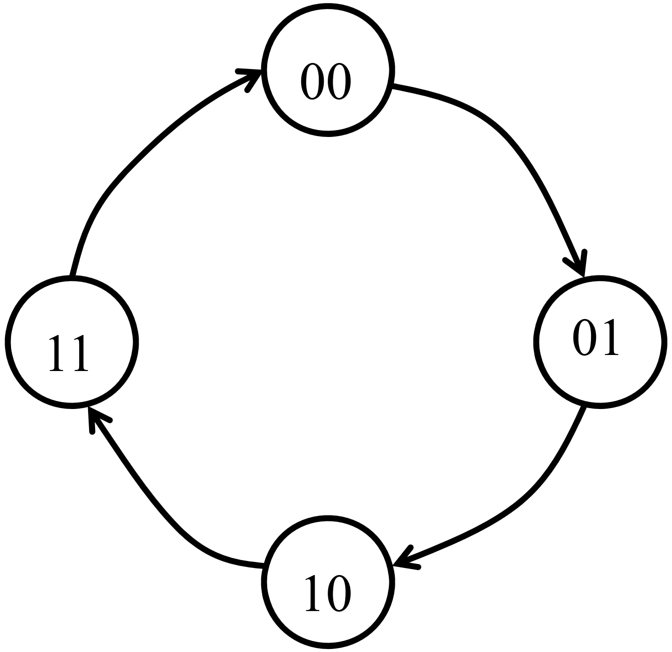


2. State table

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

Design of a 2-bit up-counter

1. State diagram



2. State table

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

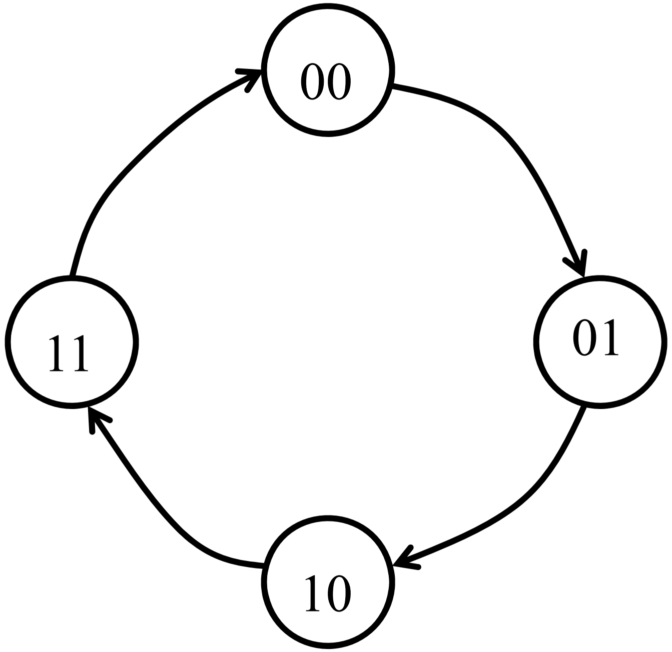
3. Next state equations

$$Q_0\,(t+1) = \overline{Q_0}$$

$$Q_1\,(t+1) = Q_0 \oplus Q_1$$

Design of a 2-bit up-counter

1. State diagram



2. State table

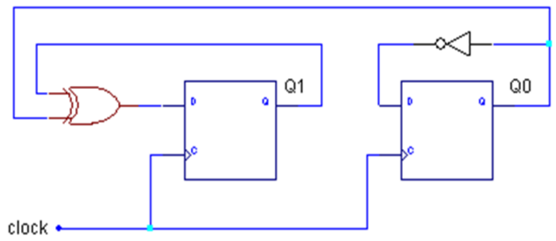
Present State		Next State	
Q_1	Q_0	Q_1	Q_0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

3. Next state equations

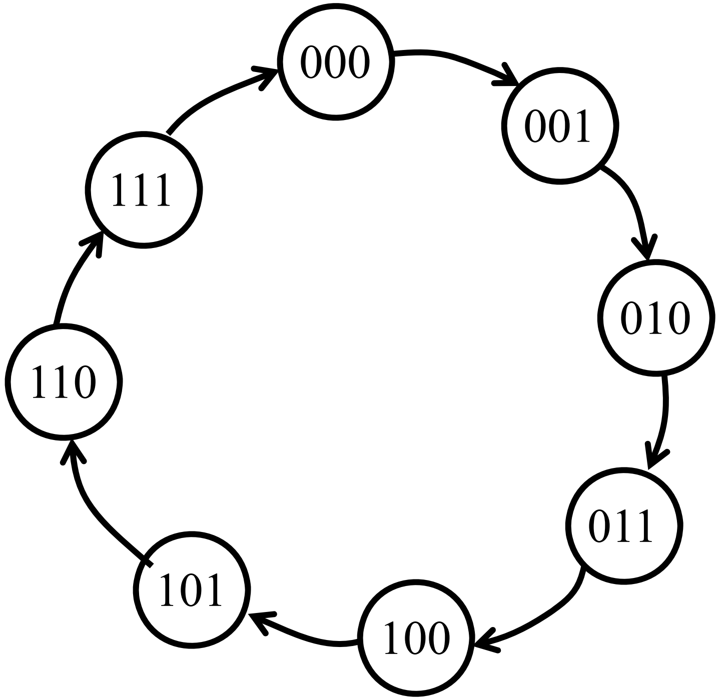
$$Q_0\,(t+1) = \overline{Q_0}$$

$$Q_1\,(t+1) = Q_0 \oplus Q_1$$

4. Circuit Diagram



Design of a mod-8 counter



State	Present State			Next State		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	1	1	0
6	1	1	0	1	1	1
7	1	1	1	0	0	0

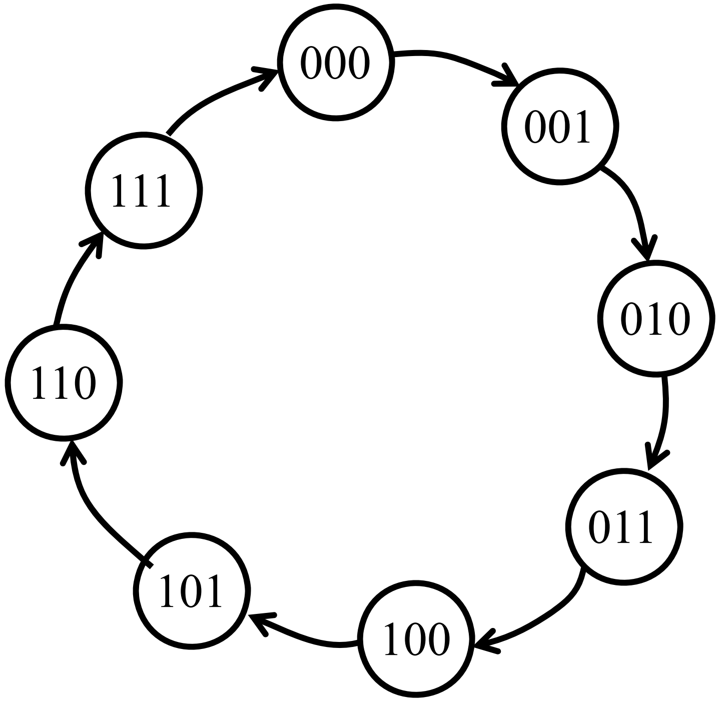
Next state equations:

$$Q_0\,(t+1) = \overline{Q_0}$$

$$Q_1\,(t+1) = \overline{Q_2}\,\overline{Q_1}Q_0 + \overline{Q_2}Q_1\overline{Q_0} + Q_2\overline{Q_1}Q_0 + Q_2Q_1\overline{Q_0}$$

$$= \overline{Q}1Q0 + Q1\overline{Q}0 = Q1 \oplus Q0$$

Design of a mod-8 counter



State	Present State			Next State		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	1	1	0
6	1	1	0	1	1	1
7	1	1	1	0	0	0

Next state equations:

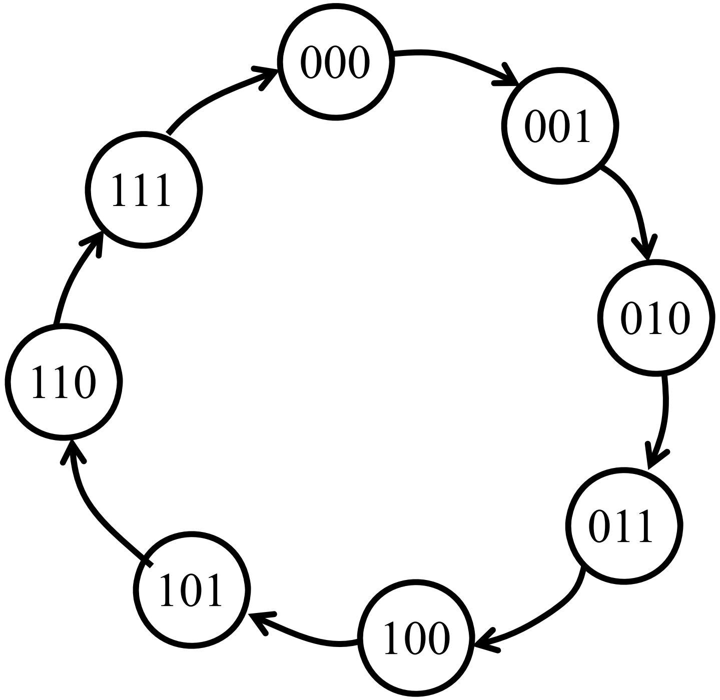
$$Q_0\,(t+1) = \overline{Q_0}$$

$$Q_1\,(t+1) = \overline{Q_2}\,\overline{Q_1}Q_0 + \overline{Q_2}Q_1\overline{Q_0} + Q_2\overline{Q_1}Q_0 + Q_2Q_1\overline{Q_0}$$

$$= \overline{Q}1Q0 + Q1\overline{Q}0 = Q1 \oplus Q0$$

$\overline{Q}_1 Q_0 (\overline{Q}_2 + Q_2)$

Design of a mod-8 counter



State	Present State			Next State		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	1	1	0
6	1	1	0	1	1	1
7	1	1	1	0	0	0

Next state equations:

$$Q_0\,(t+1) = \overline{Q_0}$$

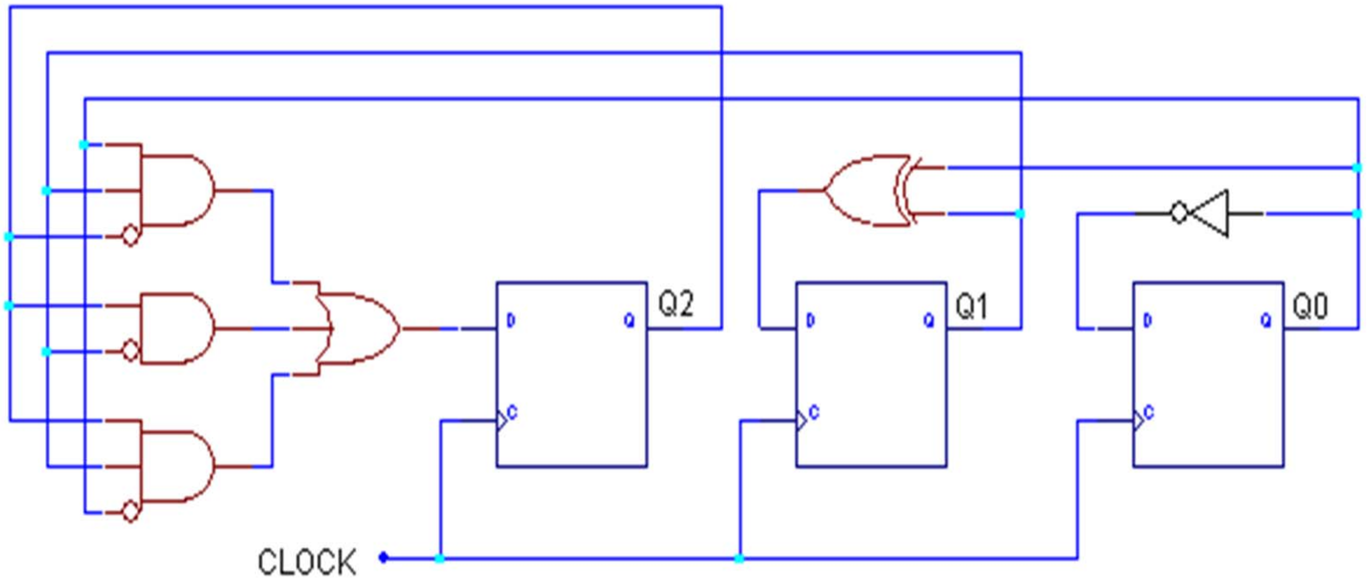
$$Q_1\,(t+1) = \overline{Q_2}\,\overline{Q_1}Q_0 + \overline{Q_2}Q_1\overline{Q_0} + Q_2\overline{Q_1}Q_0 + Q_2Q_1\overline{Q_0}$$

$$= \overline{Q}1Q0 + Q1\overline{Q}0 = Q1 \oplus Q0$$

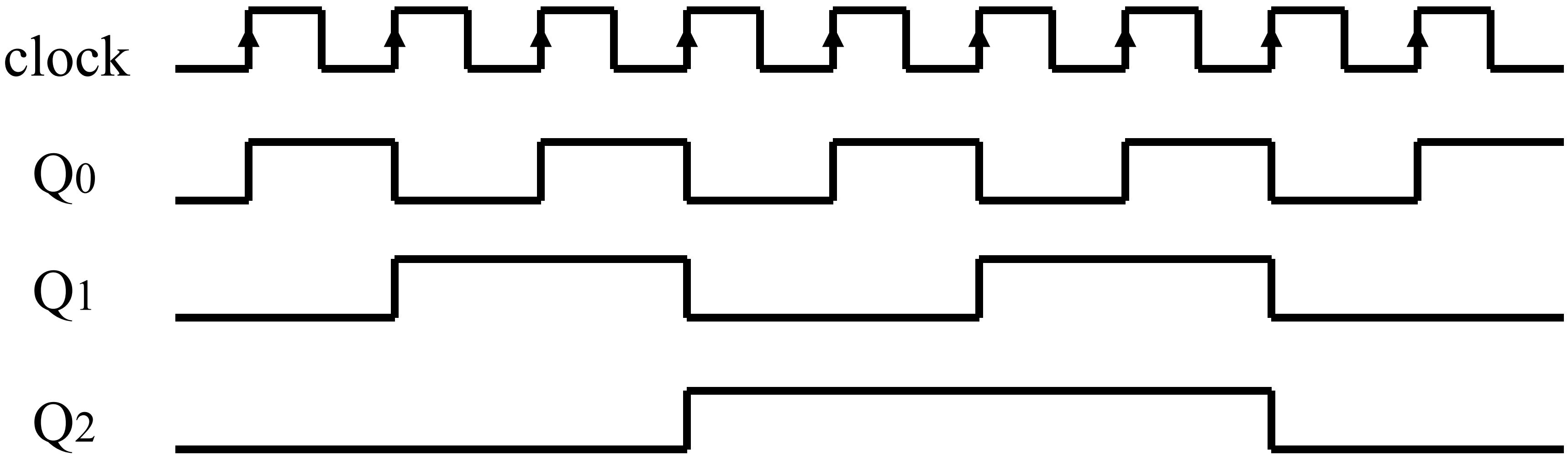
Logic Circuit for Mod-8 Counter

$Q_2\,(t+1) = \overline{Q}_2 Q_1 Q_0 + Q_2 \overline{Q}_1 \overline{Q}_0 + Q_2 \overline{Q}_1 Q_0 + Q_2 Q_1 \overline{Q}_0$

$Q_2\,(t+1) = \overline{Q}_2 Q_1 Q_0 + Q_2 \overline{Q}_1 + Q_2 Q_1 \overline{Q}_0$



Timing Diagram for Mod-8 Counter

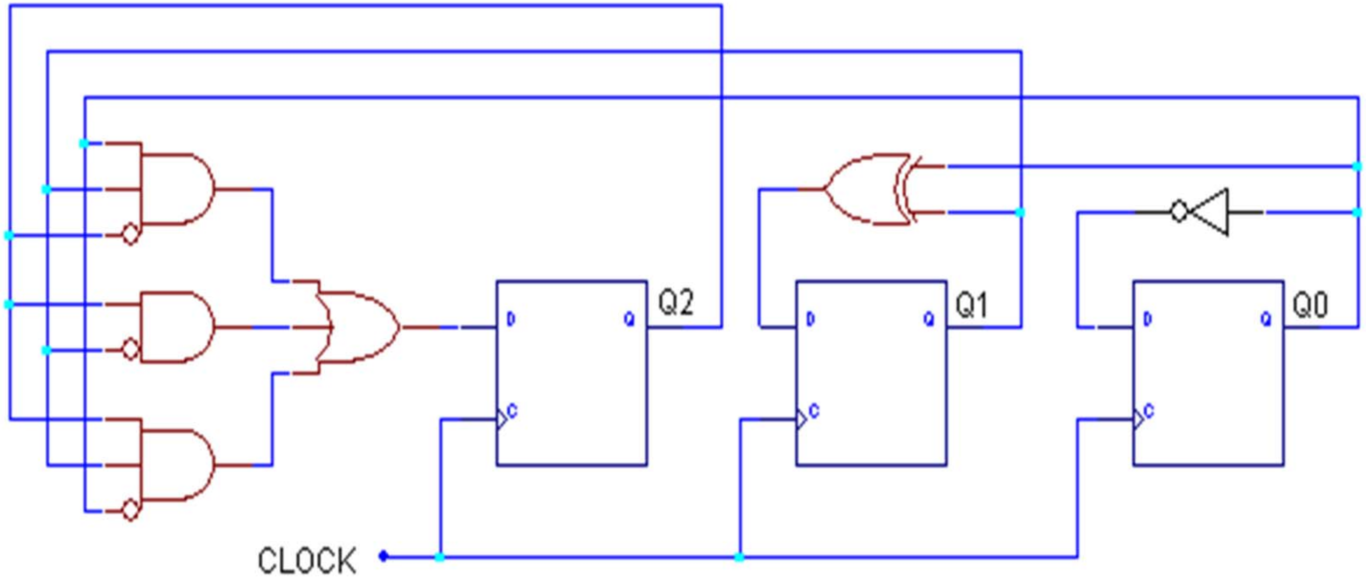


Counters can count up or down or sometimes count in irregular sequences, missing states out. The count sequence is known as the ‘coding’.

‘Loadable’ counters can be set to a specific value.

Maximum Operating Frequency

What is the theoretical maximum frequency of operation for this counter ? Assume logic gate propagation delay of 3ns, inverter propagation delay of 1ns, flip-flops have a setup time of 1.5ns and a propagation delay of 2.5ns.



$$f_{max} = 1 / (t_s \quad + \quad t_p \quad + t_c \quad)$$

$$= 1 / (1.5\text{ns} + 2.5\text{ns} + 6\text{ns}) = 100 \text{ MHz}$$

Mod-N Synchronous Counters

Mod-N counters may not cycle through all possible states in the sequence. Consider a mod-6 counter which is required to follow the sequence 000,001,010,011,100,101 and then reset to 000.

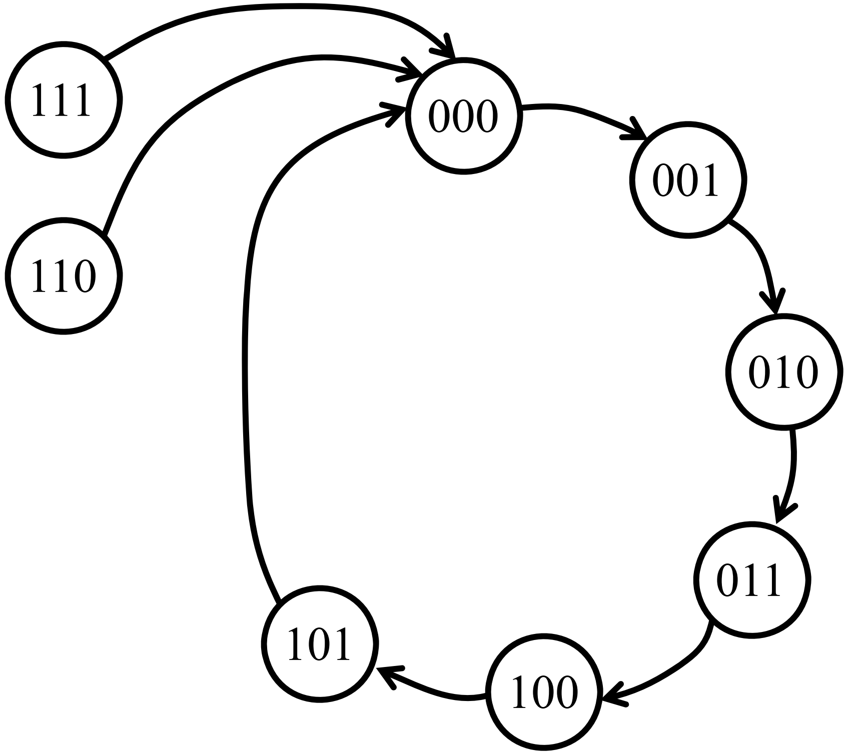
In this case, the states 110 and 111 will never occur and can be replaced by don’t care values. This may simplify the logic required.

A more robust solution would be to decide what state the counter should go to if states 6 or 7 were entered due to a circuit error.

State	Present State			Next State		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	0	0	0
6	1	1	0	x	x	x
7	1	1	1	x	x	x

Self Starting Counters

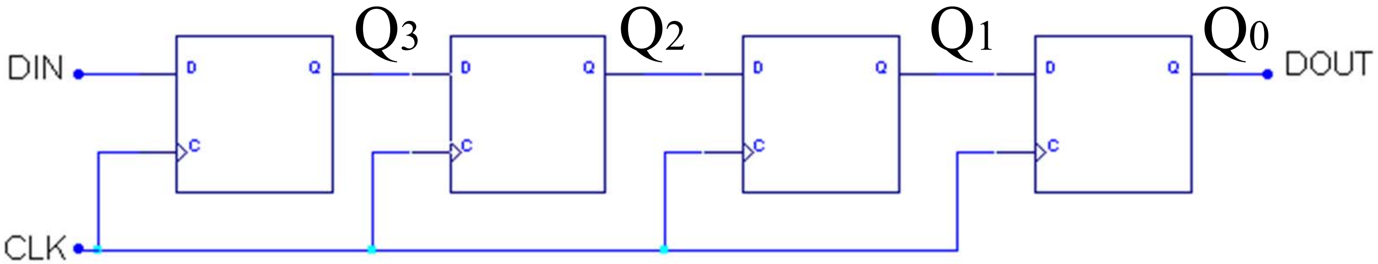
A common choice is for unused states to lead to the state 0 so that if an unused state was entered due to a circuit error, then at the next clock cycle, the counter resets.



State	Present State			Next State		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0
4	1	0	0	1	0	1
5	1	0	1	0	0	0
6	1	1	0	0	0	0
7	1	1	1	0	0	0

Shift Registers

If the output from one flip-flop in a register is fed into the input of the next, a shift register is formed.



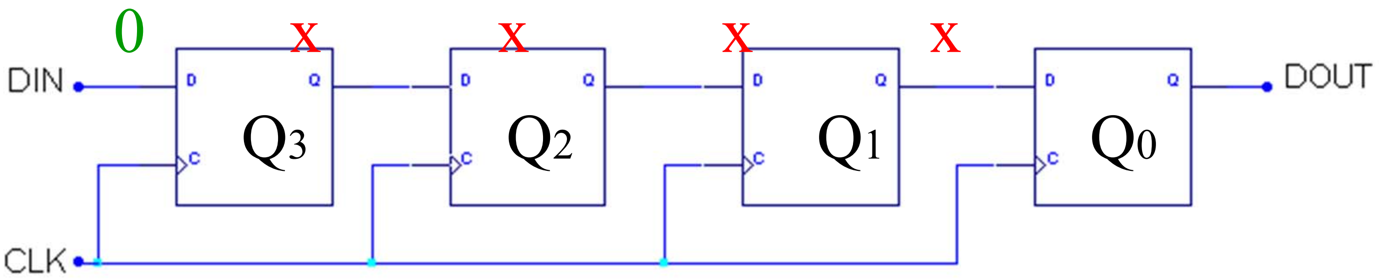
Each clock pulse shifts the register contents one bit position to the right. Data enters the register from the left and exits from the right.

CLK	DIN	Q_3	Q_2	Q_1	Q_0
		x	x	x	x
↑	0	0	x	x	x
↑	0	0	0	x	x
↑	1	1	0	0	x
↑	1	1	1	0	0
↑	0	0	1	1	0

In this case, ‘x’ represents an unknown state. The outputs are available in parallel from each flip-flop output.

Shift Registers

If the output from one flip-flop in a register is fed into the input of the next, a shift register is formed.



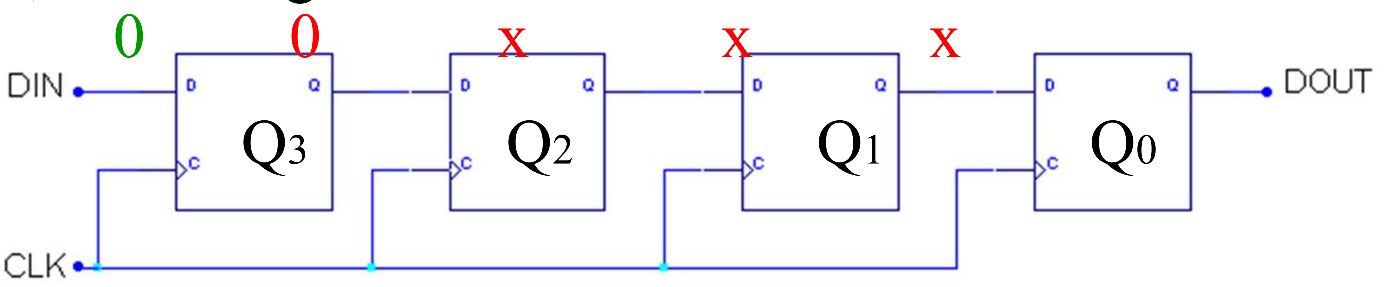
Each clock pulse shifts the register contents one bit position to the right. Data enters the register from the left and exits from the right.

CLK	DIN	Q_3	Q_2	Q_1	Q_0
		x	x	x	x
↑	0	0	x	x	x
↑	0	0	0	x	x
↑	1	1	0	0	x
↑	1	1	1	0	0
↑	0	0	1	1	0

In this case, ‘x’ represents an unknown state. The outputs are available in parallel from each flip-flop output.

Shift Registers

If the output from one flip-flop in a register is fed into the input of the next, a shift register is formed.



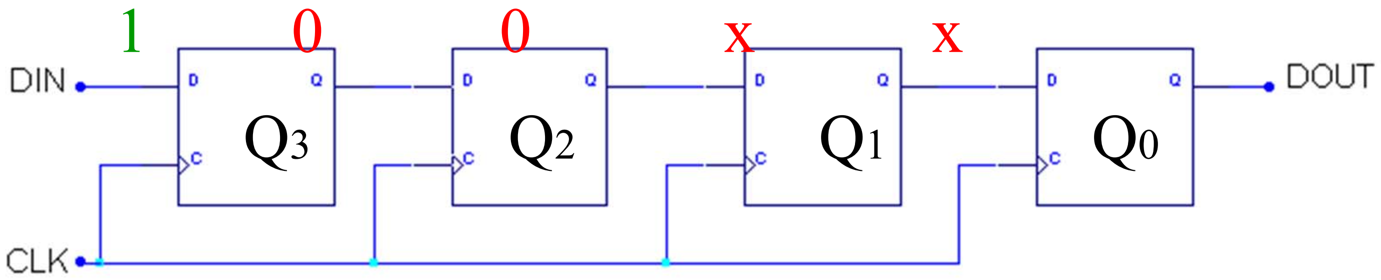
Each clock pulse shifts the register contents one bit position to the right. Data enters the register from the left and exits from the right.

CLK	DIN	Q_3	Q_2	Q_1	Q_0
		x	x	x	x
↑	0	0	x	x	x
↑	0	0	0	x	x
↑	1	1	0	0	x
↑	1	1	1	0	0
↑	0	0	1	1	0

In this case, ‘x’ represents an unknown state. The outputs are available in parallel from each flip-flop output.

Shift Registers

If the output from one flip-flop in a register is fed into the input of the next, a shift register is formed.



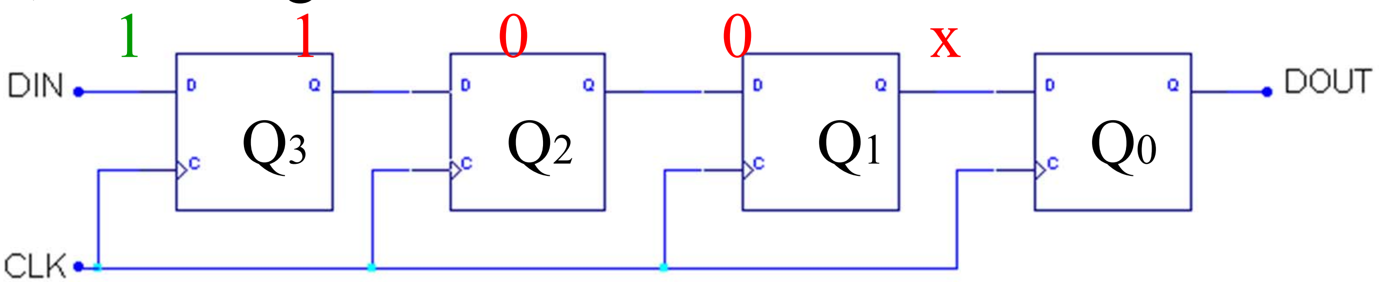
Each clock pulse shifts the register contents one bit position to the right. Data enters the register from the left and exits from the right.

CLK	DIN	Q_3	Q_2	Q_1	Q_0
		x	x	x	x
↑	0	0	x	x	x
↑	0	0	0	x	x
↑	1	1	0	0	x
↑	1	1	1	0	0
↑	0	0	1	1	0

In this case, ‘x’ represents an unknown state. The outputs are available in parallel from each flip-flop output.

Shift Registers

If the output from one flip-flop in a register is fed into the input of the next, a shift register is formed.



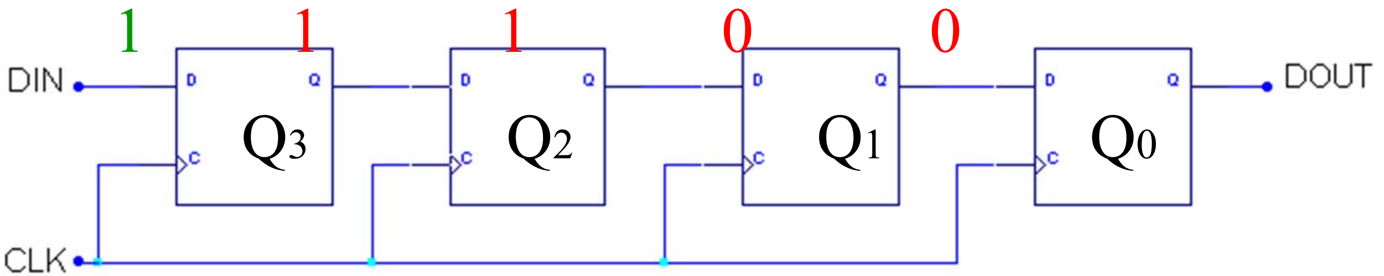
Each clock pulse shifts the register contents one bit position to the right. Data enters the register from the left and exits from the right.

CLK	DIN	Q_3	Q_2	Q_1	Q_0
		x	x	x	x
↑	0	0	x	x	x
↑	0	0	0	x	x
↑	1	1	0	0	x
↑	1	1	1	0	0
↑	0	0	1	1	0

In this case, ‘x’ represents an unknown state. The outputs are available in parallel from each flip-flop output.

Shift Registers

If the output from one flip-flop in a register is fed into the input of the next, a shift register is formed.



Each clock pulse shifts the register contents one bit position to the right. Data enters the register from the left and exits from the right.

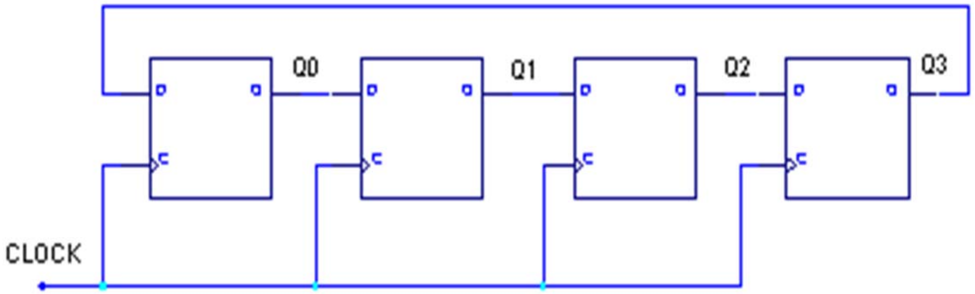
CLK	DIN	Q_3	Q_2	Q_1	Q_0
		x	x	x	x
↑	0	0	x	x	x
↑	0	0	0	x	x
↑	1	1	0	0	x
↑	1	1	1	0	0
↑	0	0	1	1	0

In this case, ‘x’ represents an unknown state. The outputs are available in parallel from each flip-flop output.

Ring Counters

Shift registers can be used to produce a type of counter that has the advantage of operating at very high speeds. This is because there is no combinational circuitry to produce delays. The MSB output is fed back to the LSB input.

4-bit ring counter



The first flip-flop (Q_0) is preset to ‘1’. The ‘1’ circulates around the counter with each clock input, producing a mod-n counter. N flip-flops will produce a mod-N ring counter.

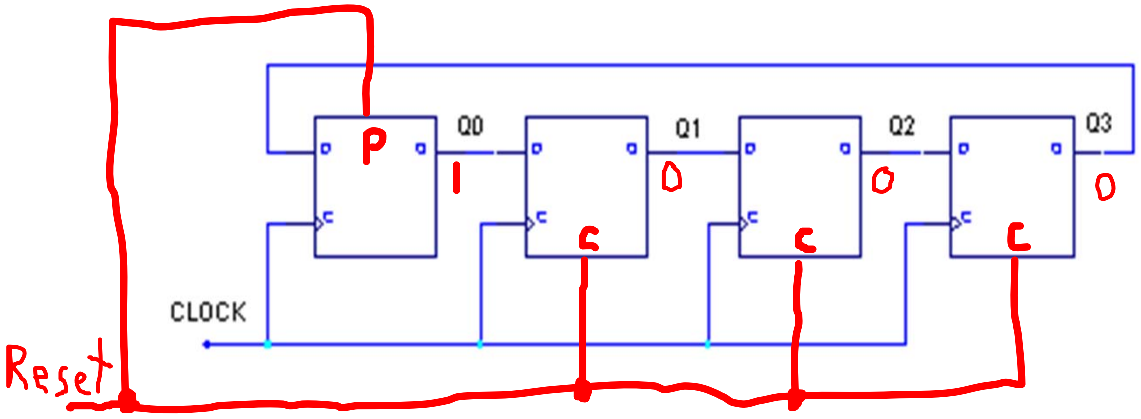
N.B. preset and clear connections are omitted for clarity.

Q_3	Q_2	Q_1	Q_0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0

Ring Counters

Shift registers can be used to produce a type of counter that has the advantage of operating at very high speeds. This is because there is no combinational circuitry to produce delays. The MSB output is fed back to the LSB input.

4-bit ring counter



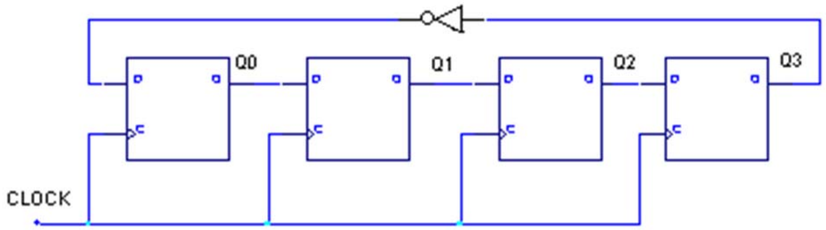
The first flip-flop (Q_0) is preset to ‘1’. The ‘1’ circulates around the counter with each clock input, producing a mod-n counter. N flip-flops will produce a mod-N ring counter.

N.B. preset and clear connections are omitted for clarity.

Q_3	Q_2	Q_1	Q_0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0

Johnson Counter

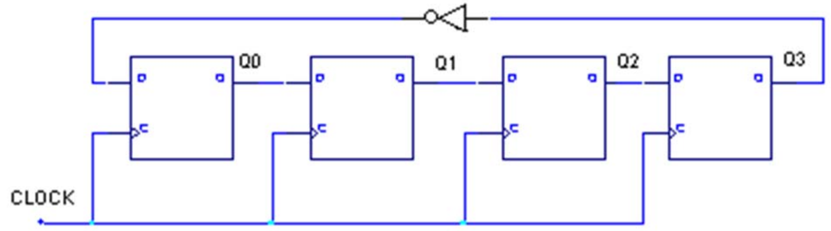
If the inverse of the MSB is fed back, we have a twisted ring counter or Johnson counter. N flip-flops will produce a mod-2N twisted ring counter.



Mod-8 twisted ring counter

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0

Maximum Operating Frequency



Johnson counter

Using the same delay values as for the straight binary encoded mod-8 counter:

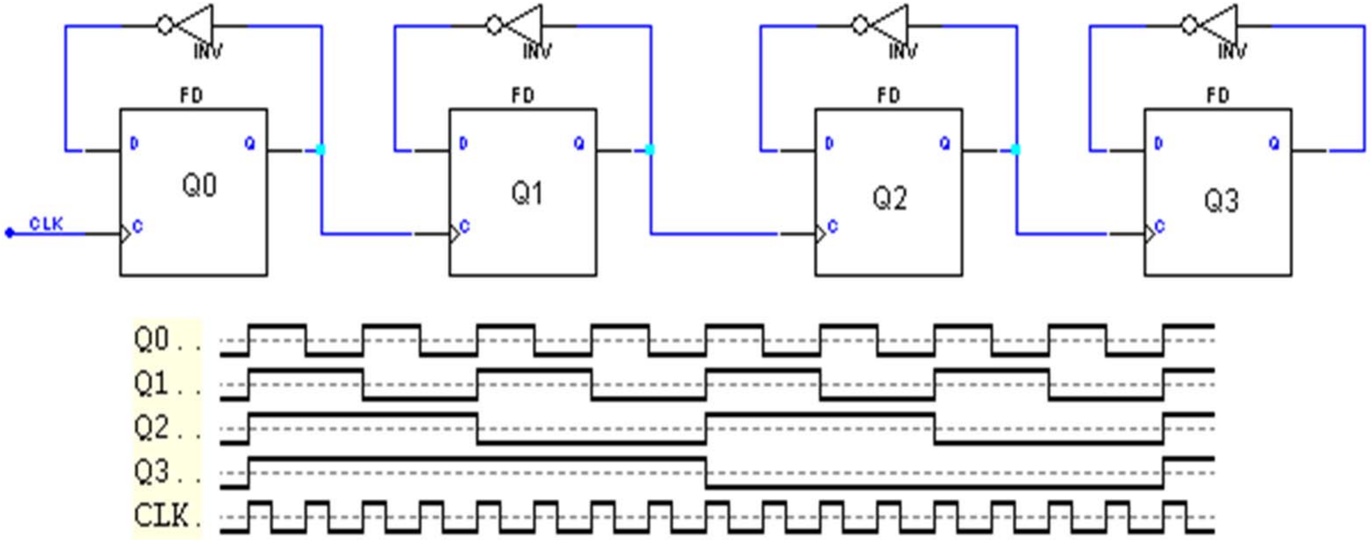
$$f_{max} = 1 / (t_s + t_p + t_c)$$

$$= 1 / (1.5\text{ns} + 2.5\text{ns} + 1\text{ns}) = 200 \text{ MHz}$$

Thus, if an application required an output pulse every 8 clock cycles, the Johnson counter could operate at twice the frequency of the straight binary encoded counter.

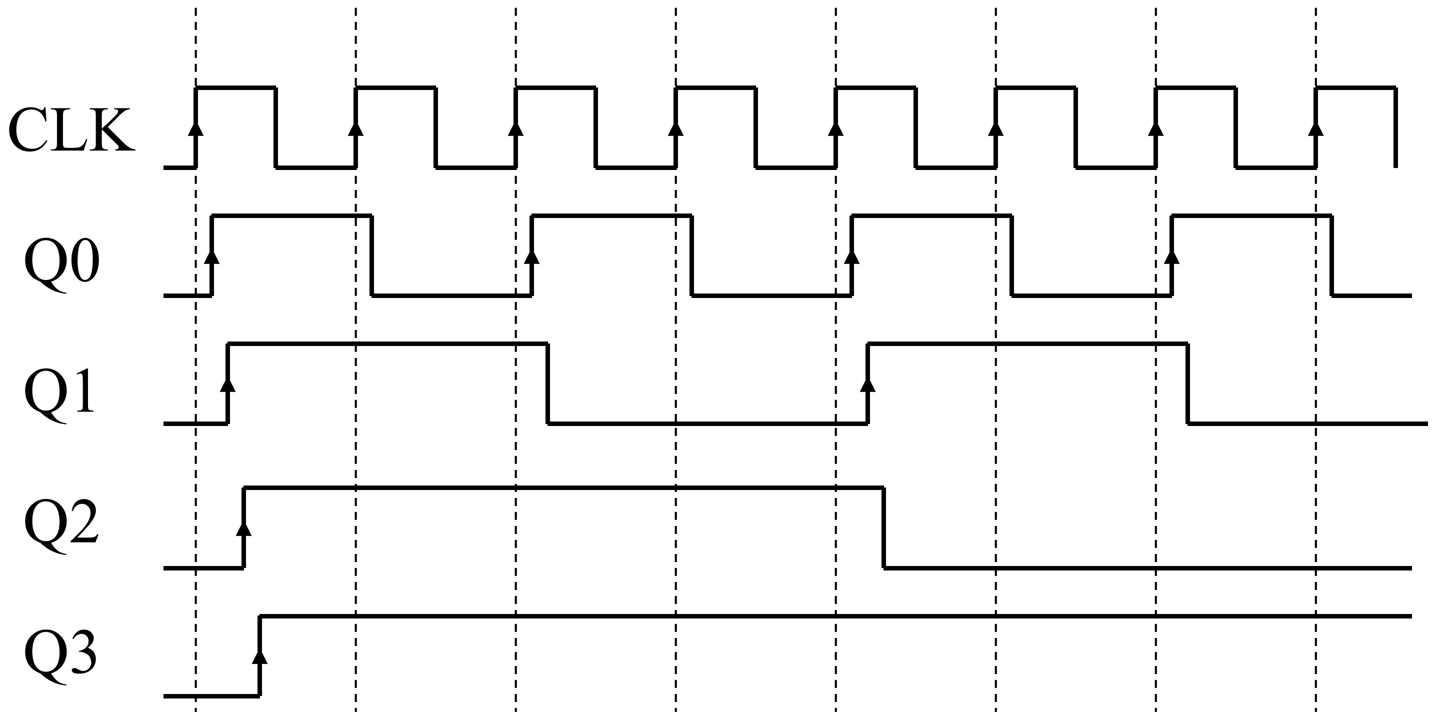
Binary Ripple Counter

A ripple counter consists of a series of flip-flops as shown. Each flip-flop is set to toggle and is clocked by the flip-flop in the preceding bit position. The LSB flip-flop receives the incoming clock.



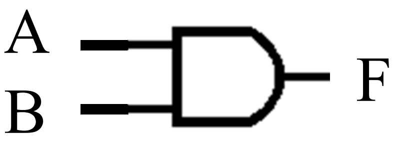
The rising clock edge causes Q_0 to toggle. The rising edge produced on Q_0 causes Q_1 to toggle and this effect ripples through the counter. It produces a down count in this case.

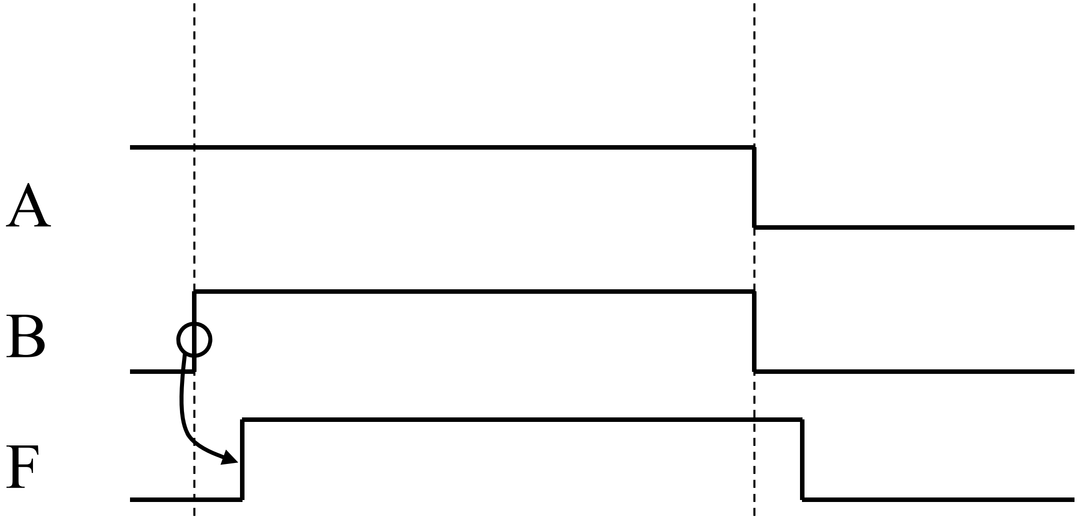
A close up view of the waveforms shows the effect of the propagation delay through the flip-flops.

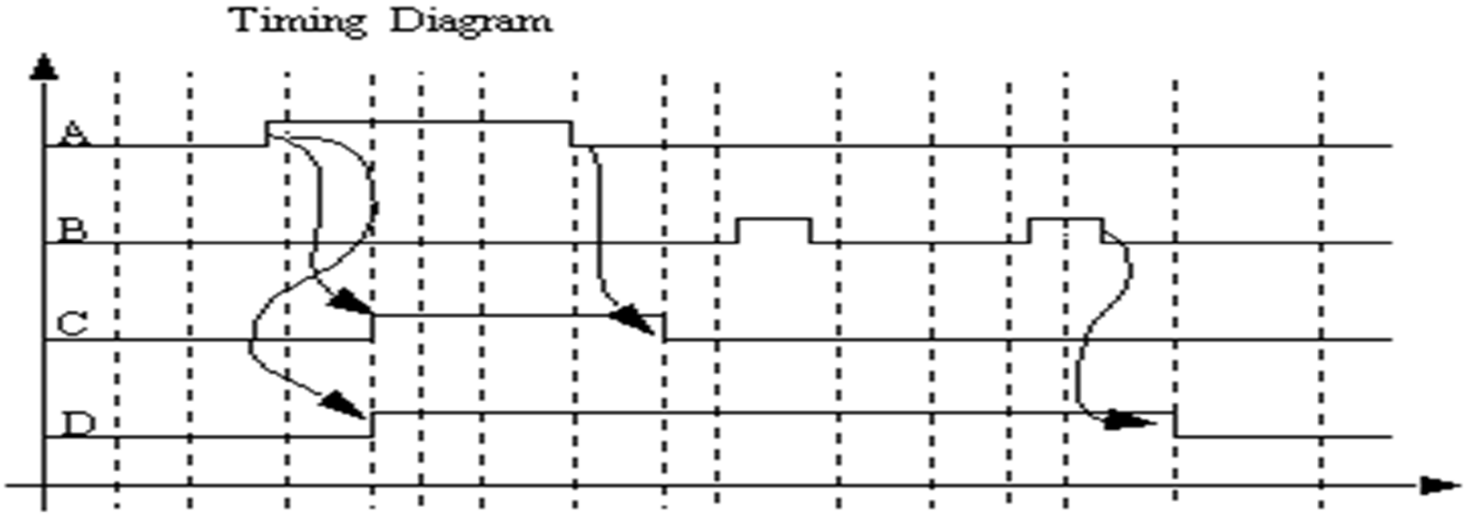


If the propagation delay through each flip-flop is 10ns, then the delay from the rising edge of the clock to Q_3 changing is 4 x 10ns = 40ns.

The maximum clock frequency is 1/40ns = 25 MHz.


A
B
F


A
B
F


Timing Diagram
A
B
C
D

Summary

- A counter goes through a prescribed sequence of states before resetting to the original state.
- The modulus of a counter is the number of cycles before it repeats the pattern.
- Counters can be designed using sequential circuit techniques.
- Fast ring counters can be formed from shift registers