



UNIVERSITI
TEKNOLOGI
MALAYSIA
www.utm.my

SCSR1013 DIGITAL LOGIC

MODULE 9: SHIFT REGISTERS

FACULTY OF COMPUTING



MODULE 9

SHIFT REGISTERS

Objectives:

1. To introduce the characteristics and function of a shift register
2. To highlight different types of shift register
3. To introduce shift register IC
4. To explain shift register counter



MODULE CONTENT

Basic Shift Register functions

Shift Register:

- Serial In/Serial Out (SISO)
- Serial In/Parallel Out (SIPO)
- Parallel In/Serial Out (PISO)
- Parallel In/Parallel Out (PIPO)

Bidirectional Shift Registers

Shift Register counters

- Ring Counter
- Johnson Counter



- **SR** is a register in which binary data can be **stored**, and this data can be **shifted** to the left or right when a signal is applied.
 - It consists of arrangements of **FFs**.
-
- Digital System : Important in applications involving:
 - a) Data storage
 - b) Data transfer/movement

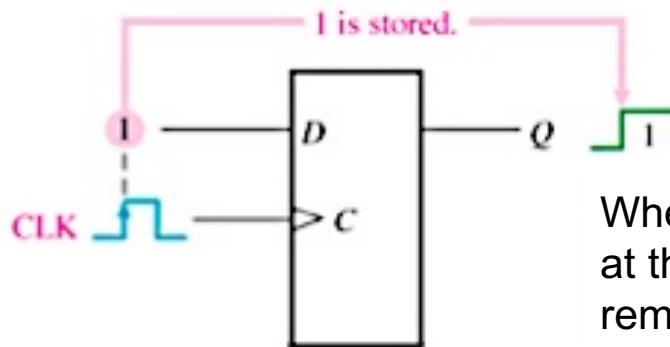
Module:
Page 33

Memory Devices
in Digital
Systems

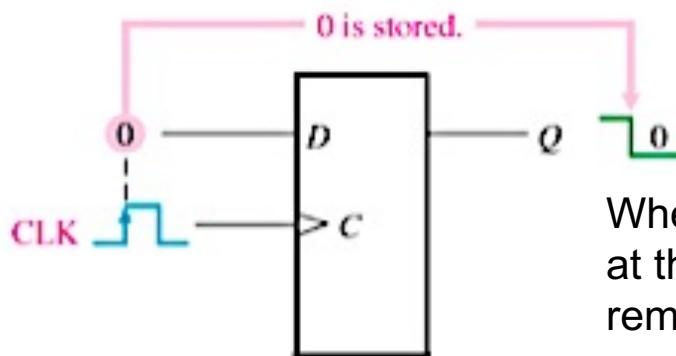
Module:
Page 7

a) The flip-flop as a storage element

- Each FF can store 1 bit data.
- More bits require more FF.



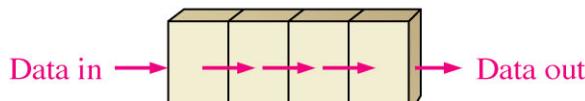
When a 1 is on D, Q becomes a 1 at the triggering edge of CLK or remains a 1 if already.



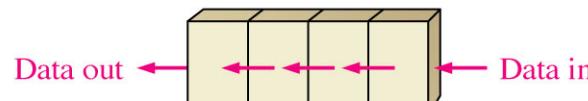
When a 0 is on D, Q becomes a 0 at the triggering edge of CLK or remains a 0 if already.

b) Basic Data Transfer/Movements in SR.

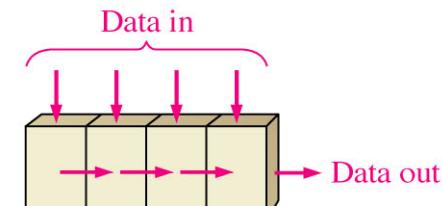
(Four bits are used for illustration. The bits move in the direction of the arrows.)



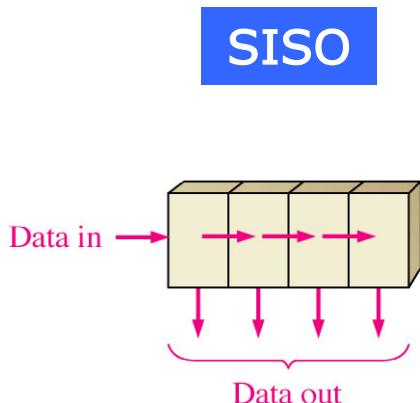
(a) Serial in/shift right/serial out



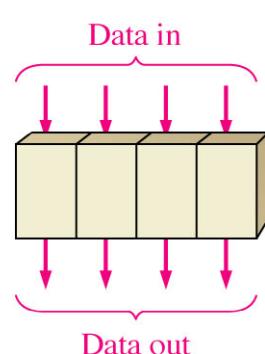
(b) Serial in/shift left/serial out



(c) Parallel in/serial out

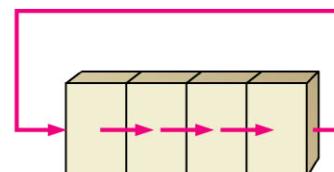


(d) Serial in/parallel out



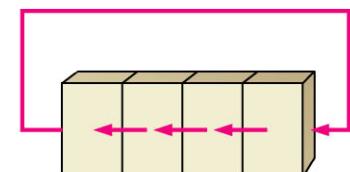
(e) Parallel in/parallel out

SISO



(f) Rotate right

PISO



(g) Rotate left

SIPO

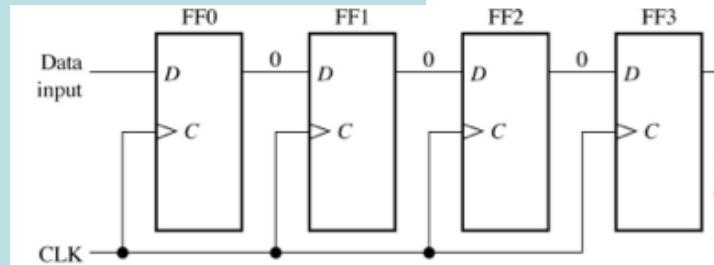
PIPO

continue...



- A register can shift data one bit at a time.
- The logical configuration of a serial SR:
 - A chain of FF connected in cascade.
- The operation of SR: **Synchronous.**
 - Using common clock.
- Shifting N bit SR will lose all data after $2N$ clock cycles.
- Solution: Using rotate.

Data transfer / movement:
(Characteristic)



- Accepts data serially, one bit (input) at a time on a single line
- Each clock pulse will enter one bit data (**Data in**) into SR and at the same time one bit will be shifted to Q_7
- Output produced in serial form too.

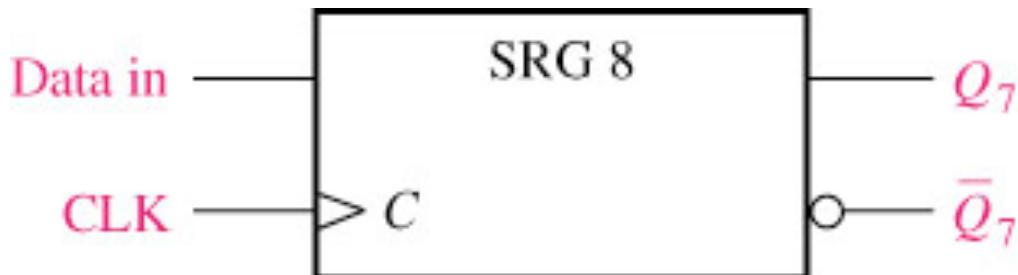
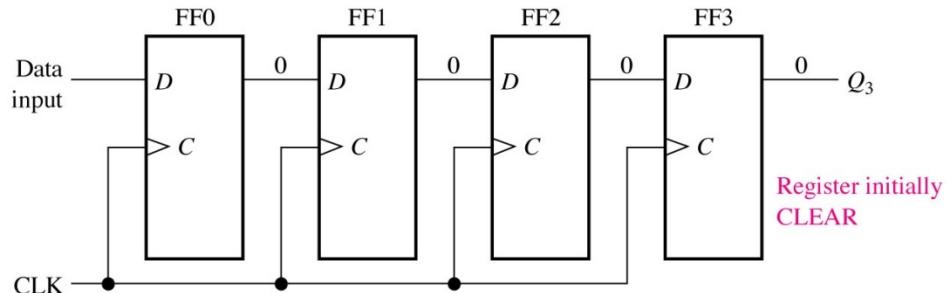


Figure : Logic symbol for an 8-bit serial in/serial out shift register



Example 1:

- Four bits (1010) being entered serially into the register
- Every clock cycle a data is shifted to the right one position
- For n bit SISO it requires n clock cycle to completely entering the shift register
 - i.e. for the first data entered to appear at the output

LSB shifted in first.



Example 1:

- Four bits (1010) being entered serially into the register
- Every clock cycle a data is shifted to the right one position
- For n bit SISO it requires n clock cycle to completely entering the shift register
 - i.e. for the first data entered to appear at the output

LSB shifted in first.

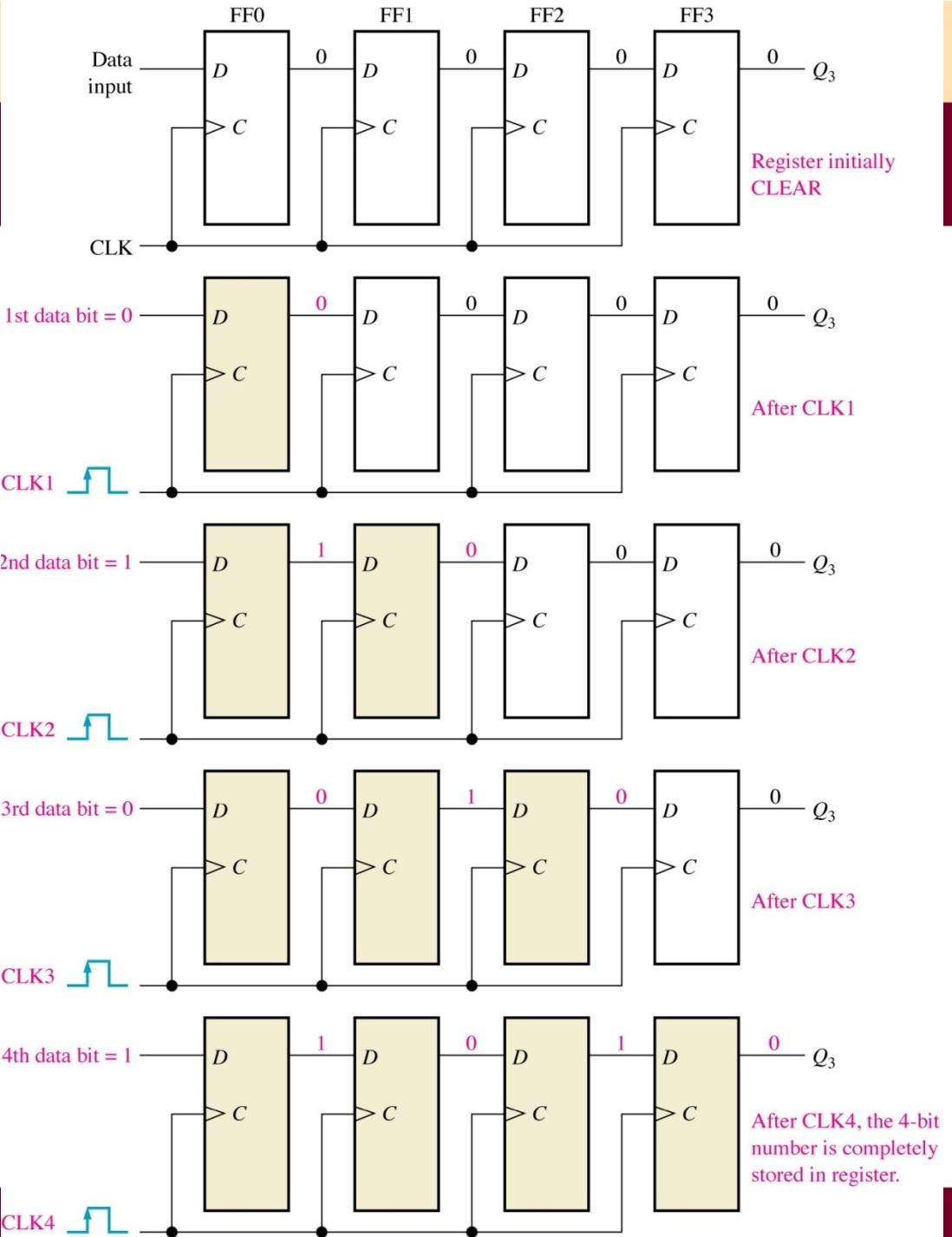


Figure : Four bits (1010) being serially shifted out of the register and replaced by all zeros.

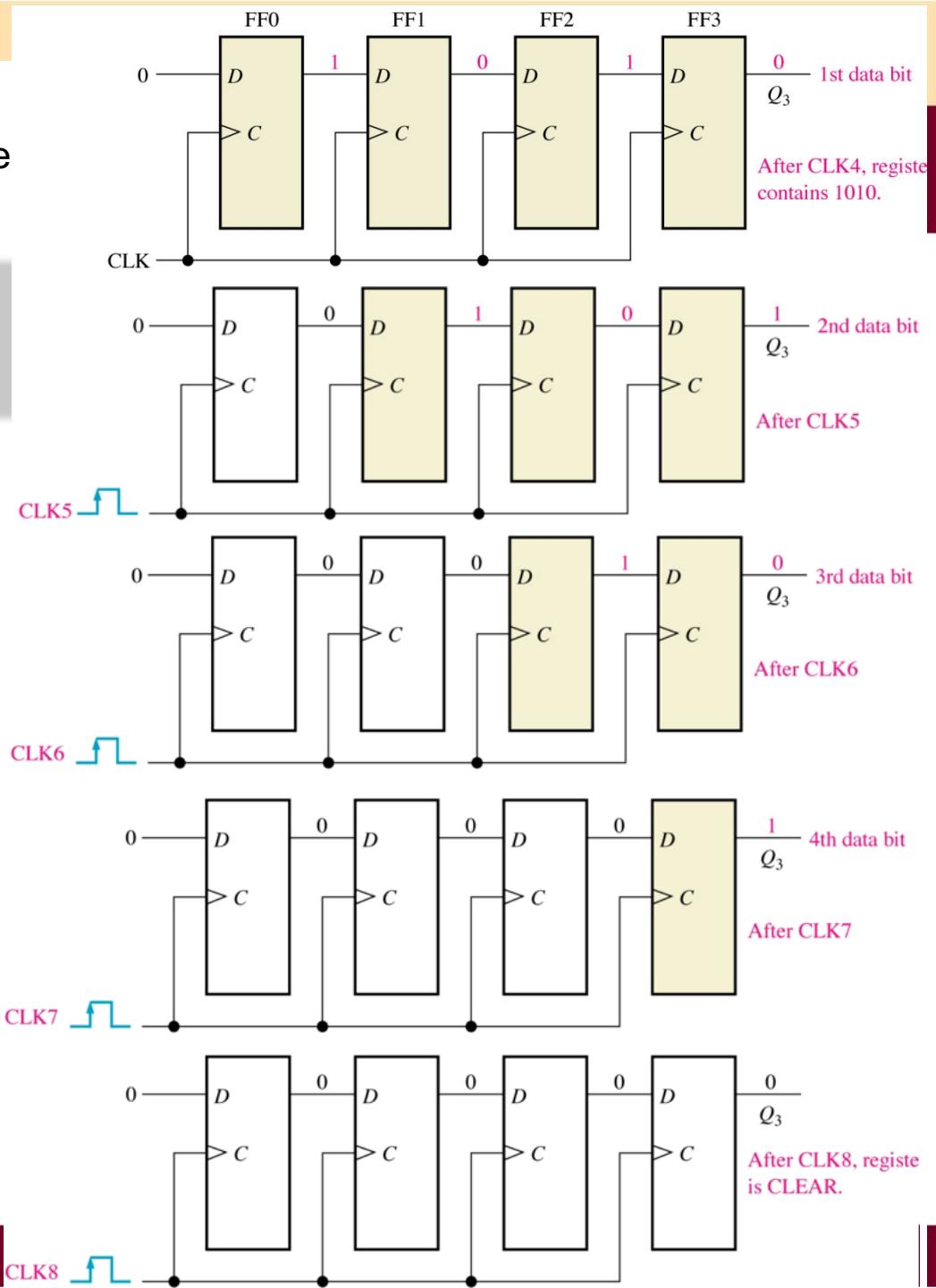
Output:

1 0 1 0

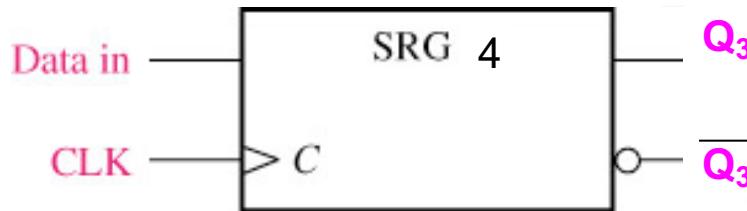
- Once all data is shifted in , every clock cycle will make the data shifted another one position to the right
- On the 5th clock cycle, the first 0 is shifted out and lost, its position replace by the 2nd data, at the left most flip-flop new data entering the SISO
- This kind of shifting repeat every clock cycle.
- Assuming every time a new data is 0, at the 8th clock cycle all flip-flop in SISO will be replaced by a new data

For n -bit SISO, it requires:

- n clock cycle to shifted in all data
- another n clock cycle to shifted out all data.



Example 2: SISO



MSB LSB
1101

Data shifted in:

Clock, t	FF0	FF1	FF2	FF3	
Initially	0	0	0	0	
1	1	0	0	0	
2	1	1	0	0	
3	0	1	1	0	
4	1	0	1	1	

Bit MSB will insert first.
The last FF will hold MSB

Data shifted out:

Clock, t	FF0	FF1	FF2	FF3	Data lost
Initially	1	0	1	1	X
1	0	1	0	1	1
2	0	0	1	0	1
3	0	0	0	1	0
4	0	0	0	0	1

Bit MSB will go out first



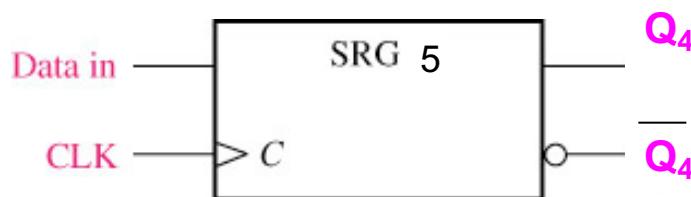
Example 3:

5-bit SISO (Right SR)

Data input: 11010

Initially all FFs are cleared.

MSB are shifted in first.





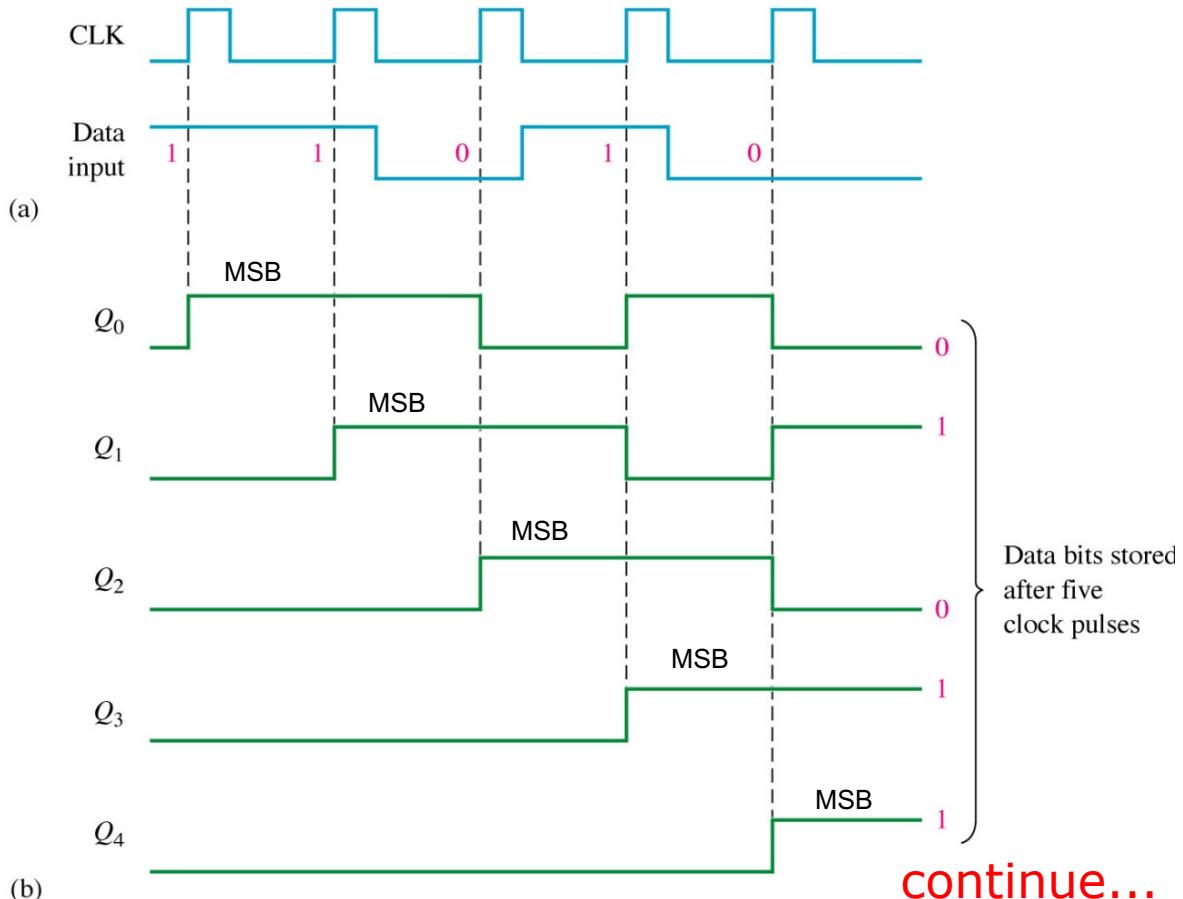
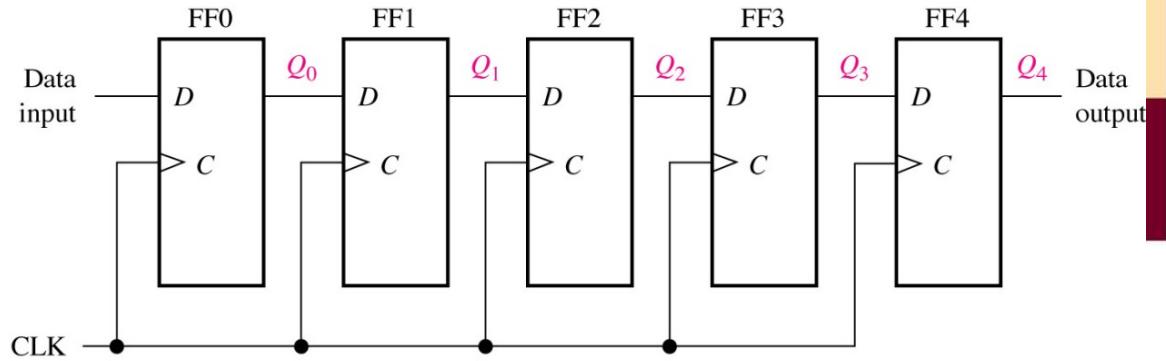
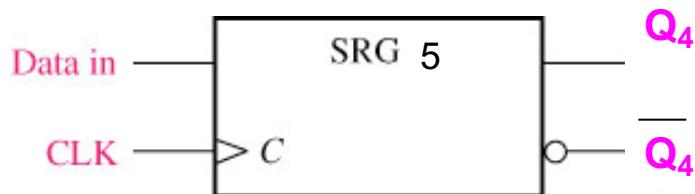
Example 3:

5-bit SISO (Right SR)

Data input: 11010

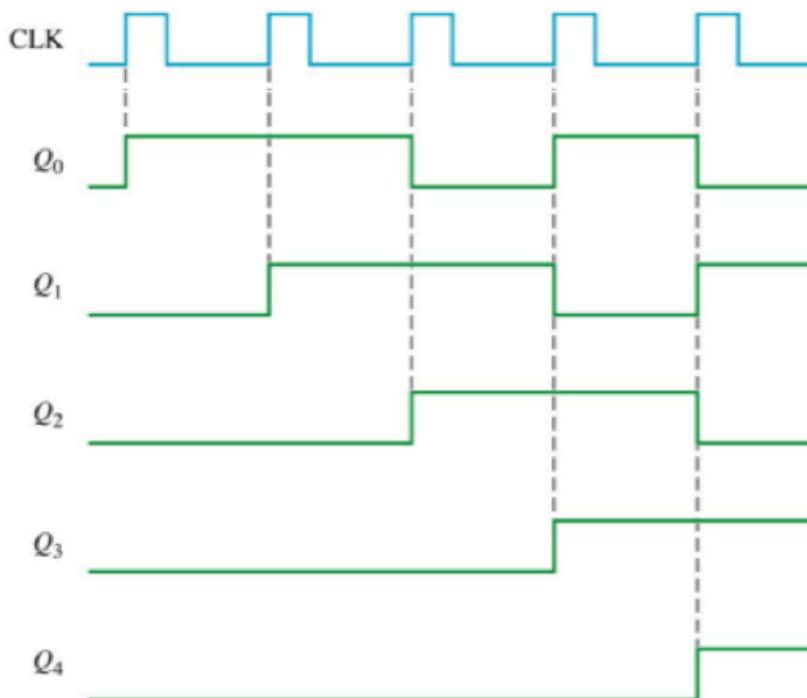
Initially all FFs are cleared.

MSB are shifted in first.





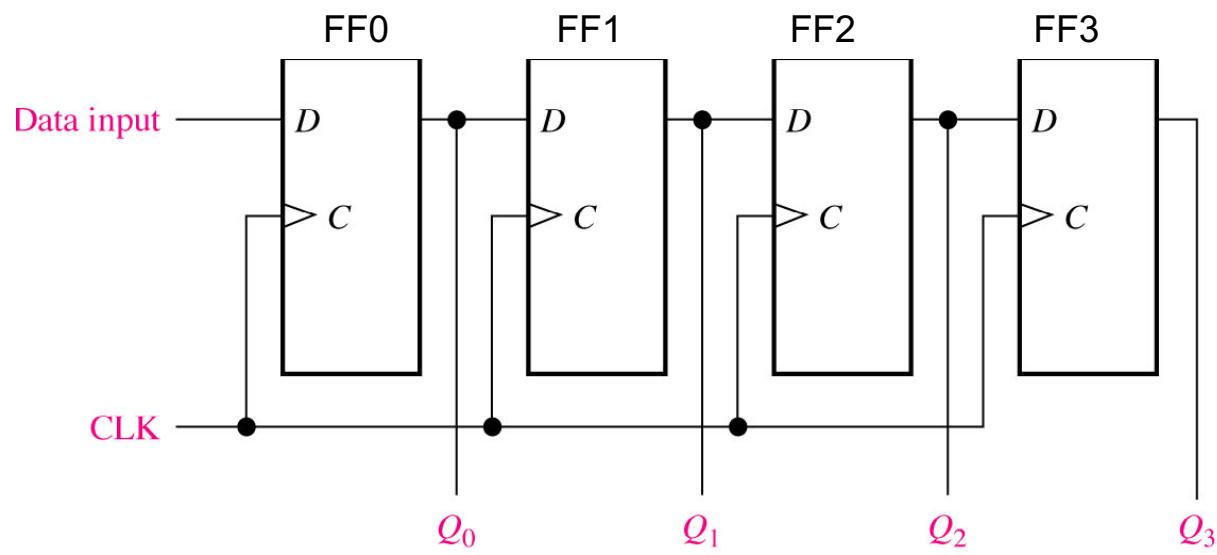
Clock, t	FF0	FF1	FF2	FF3	FF4
Initially	0	0	0	0	0
1	1	0	0	0	0
2	1	1	0	0	0
3	0	1	1	0	0
4	1	0	1	1	0
5	0	1	0	1	1



Self-Test:

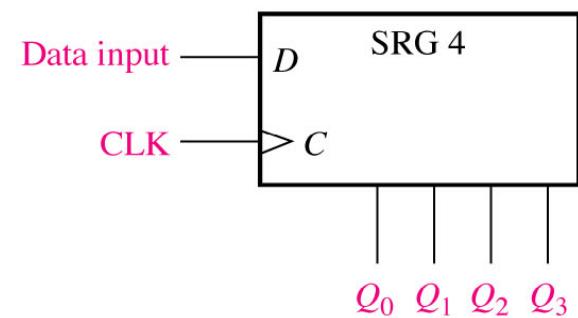
Generate a table for the data shifting out.
(MSB shifted out first)

- Data bits are entered serially one bit at a time.
- Output is parallel
 - All bits are available simultaneously after n clock cycles for n -bit SIPO.



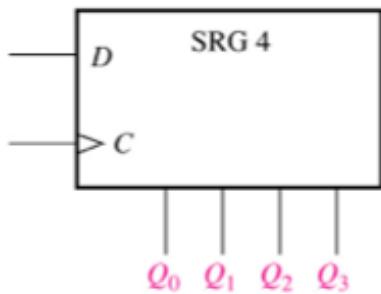
(a)

For SISO, output only at Q_3 .
For SIPO output $Q_3Q_2Q_1Q_0$

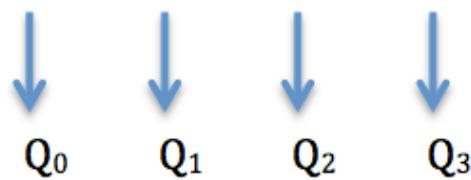


(b)

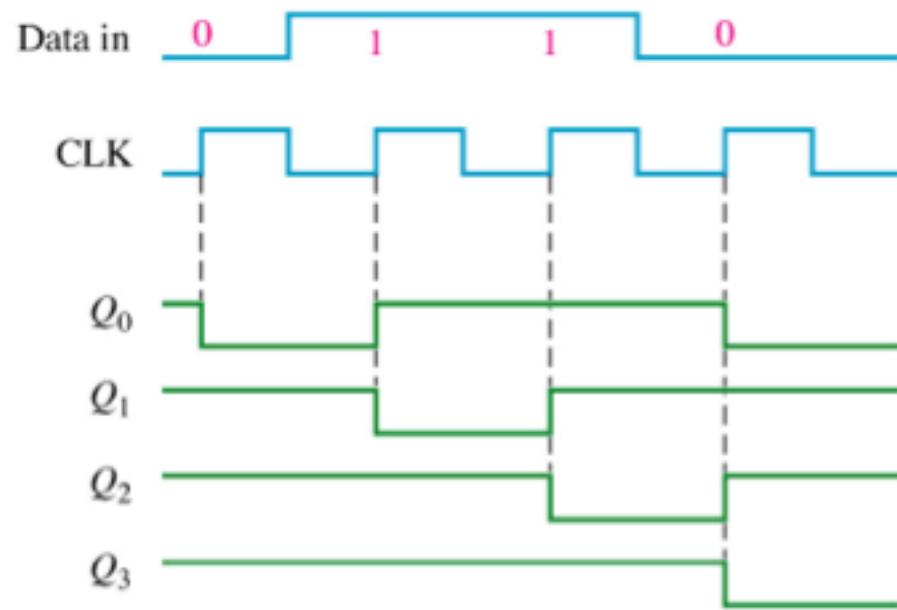
Example 4: Show the states of 4-bit SIPO shift register (SRG 4) for the data input **0110 (MSB shifted first)** and clock waveforms. The register initially all is 1s.



Clock, t	FF0	FF1	FF2	FF3
Initially	1	1	1	1
1	0	1	1	1
2	1	0	1	1
3	1	1	0	1
4	0	1	1	0



(a)





Exercise 9.1: Show the states of 6-bit SIPO shift register (SRG 6) for the data input **011001** and clock waveforms.

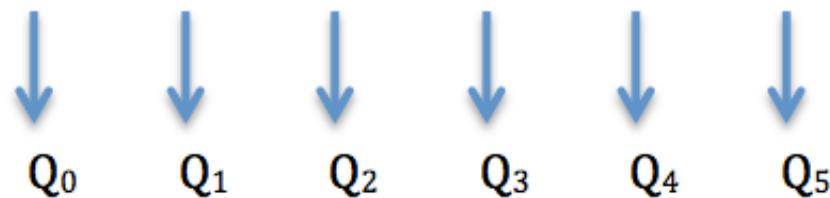
The register initially all is 0s and MSB will enter first (MSB shifted first).

Exercise 9.1: Show the states of 6-bit SIPO shift register (SRG 6) for the data input **011001** and clock waveforms.

The register initially all is 0s and MSB will enter first.

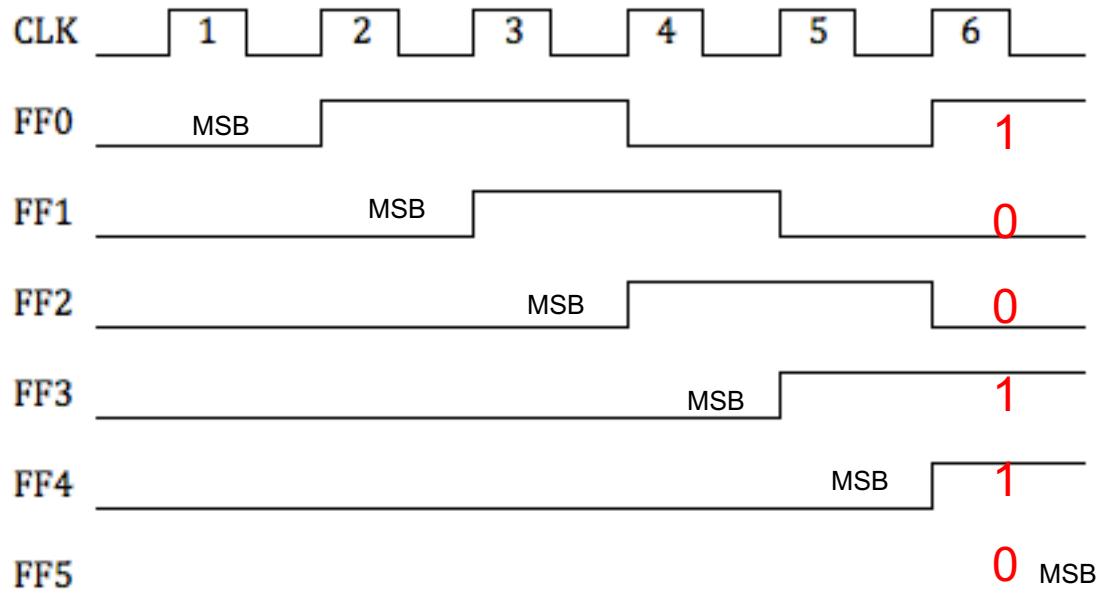
Solution :

Clock, t	FF0	FF1	FF2	FF3	FF4	FF5
Initially	0	0	0	0	0	0
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	1	1	0	0	0	0
4	0	1	1	0	0	0
5	0	0	1	1	0	0
6	1	0	0	1	1	0





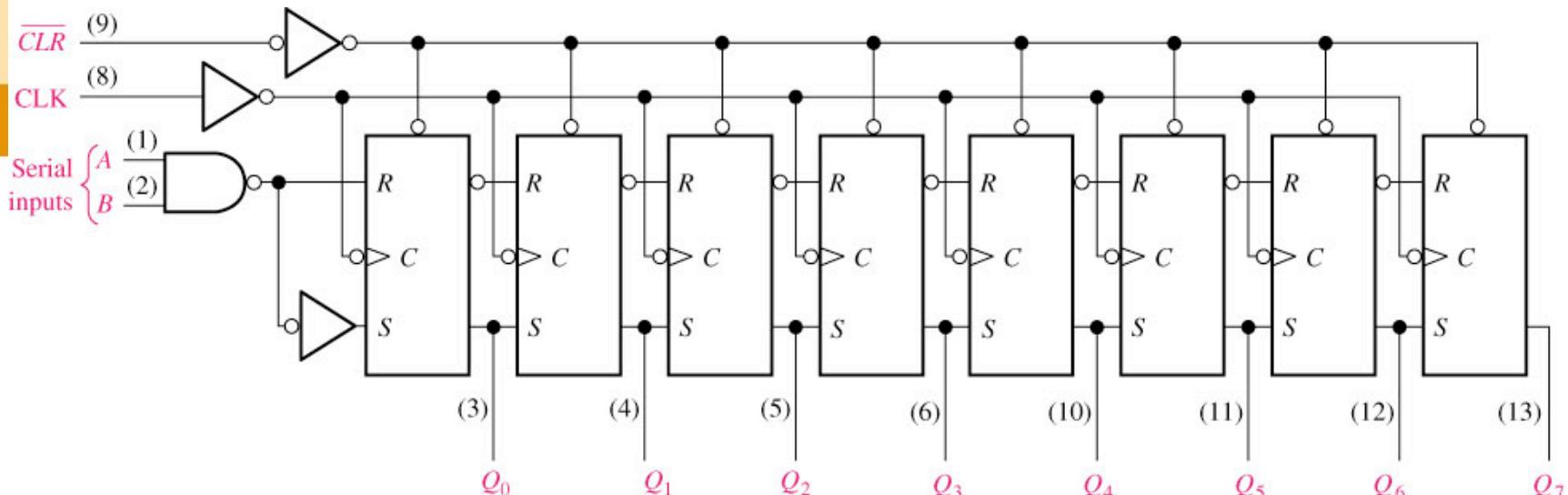
Clock, t	FF0	FF1	FF2	FF3	FF4	FF5
Initially	0	0	0	0	0	0
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	1	1	0	0	0	0
4	0	1	1	0	0	0
5	0	0	1	1	0	0
6	1	0	0	1	1	0



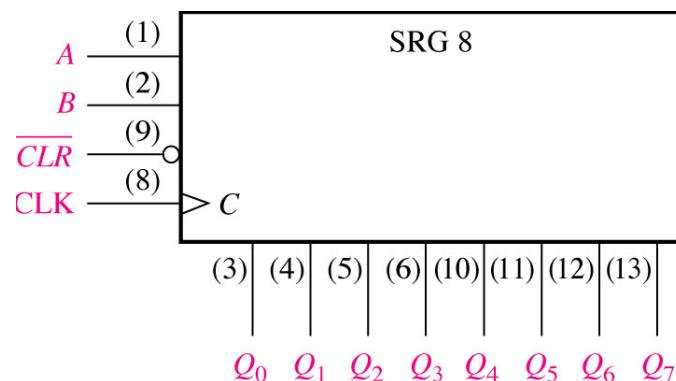
Self-Test:

Generate a table
for the data
shifting out.

Example 5: SIPO (8-bit shift register)



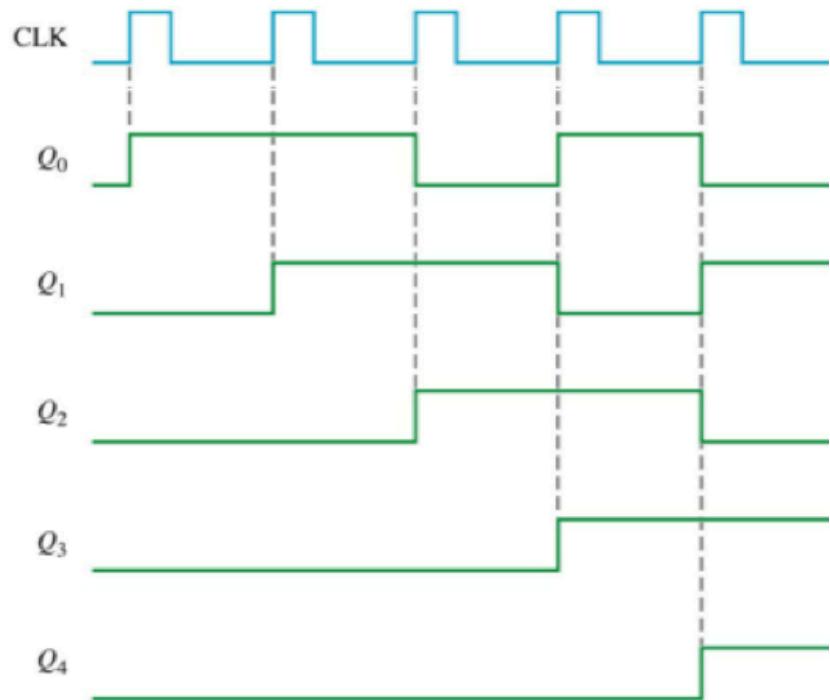
(a) Logic diagram



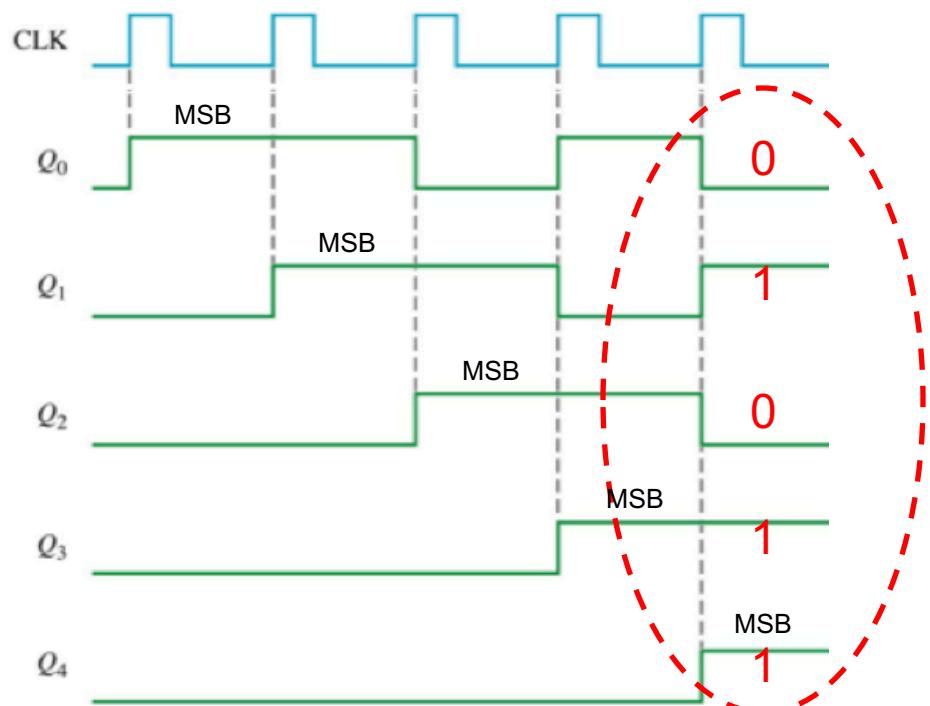
(b) Logic symbol

- Input can be A or B
- \overline{CLR}' is active LOW; clear all output to 00000000
- Data entered at A or B will appear at Q_0 after the first clock pulse.
- The data will appear at Q_7 after the 8-th clock pulse.
- A **valid parallel data** only appear at all output after 8 clock cycles.

- Exercise 9.2:**
- a) Determine how many bit entered to the SISO (Right SR) register?
 - b) Circle on the timing diagram the valid output.
 - c) Determine the data entered. (Assume MSB shifted in first)



- Exercise 9.2:**
- Determine how many bits entered to the SISO (Right SR) register?
 - Circle on the timing diagram the valid output.
 - Determine the data entered. (Assume MSB shifted in first)



Solution:

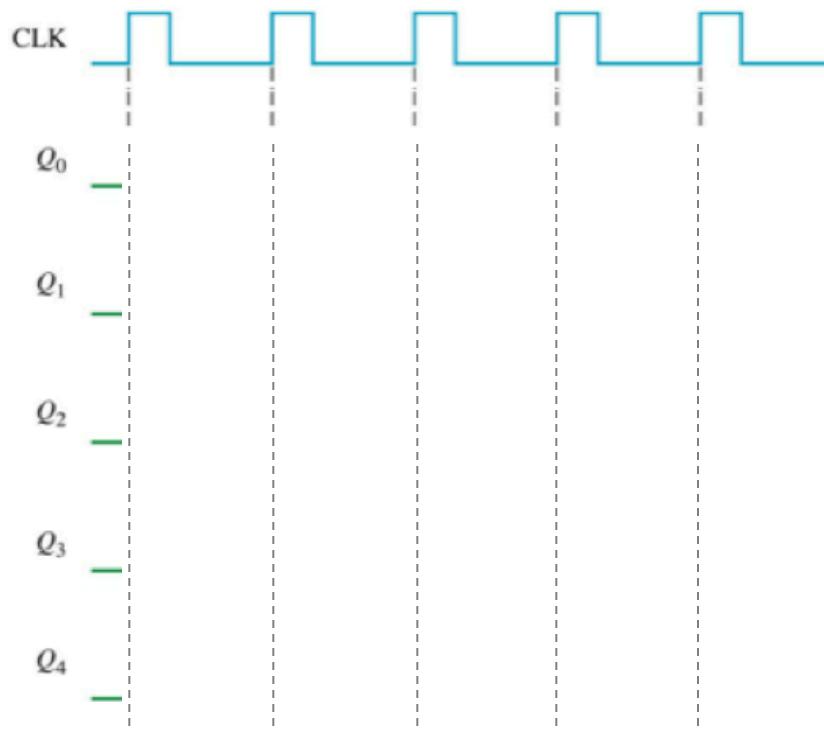
a) 5 bits

b) The valid output

c) 11010

Exercise 9.2b: Based on the data entered in exercise 9.2,

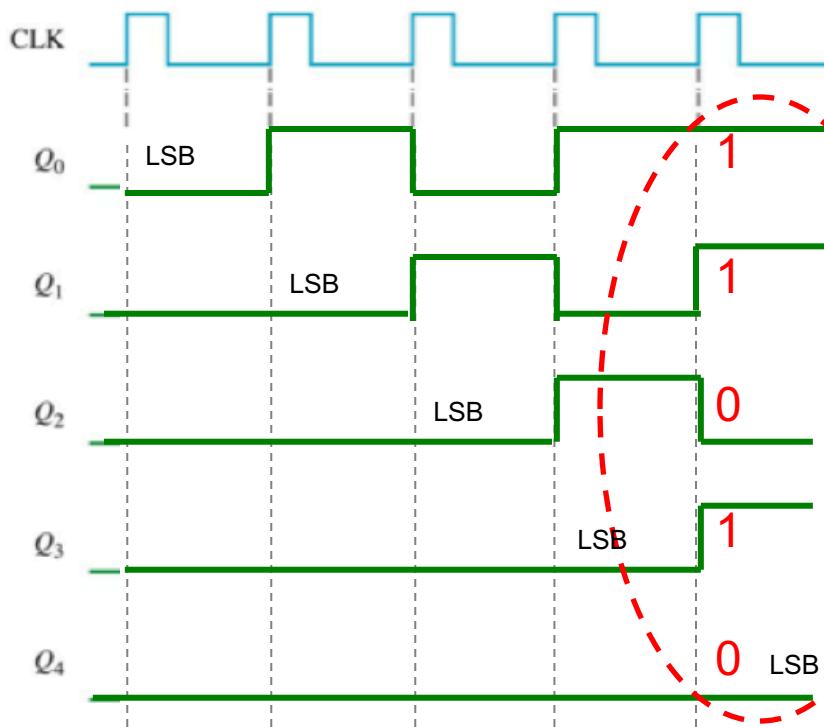
- Redraw the timing diagram with LSB shifted in first.
- Circle on the timing diagram the valid output.
- Determine which output Q or flip flop hold the LSB bit at CLK5.



Data = 11010

Exercise 9.2b: Based on the data entered in exercise 9.2,

- Redraw the timing diagram with LSB shifted in first.
- Circle on the timing diagram the valid output.
- Determine which output Q or flip flop hold the LSB bit at CLK5.



Data = 11010

Solution:

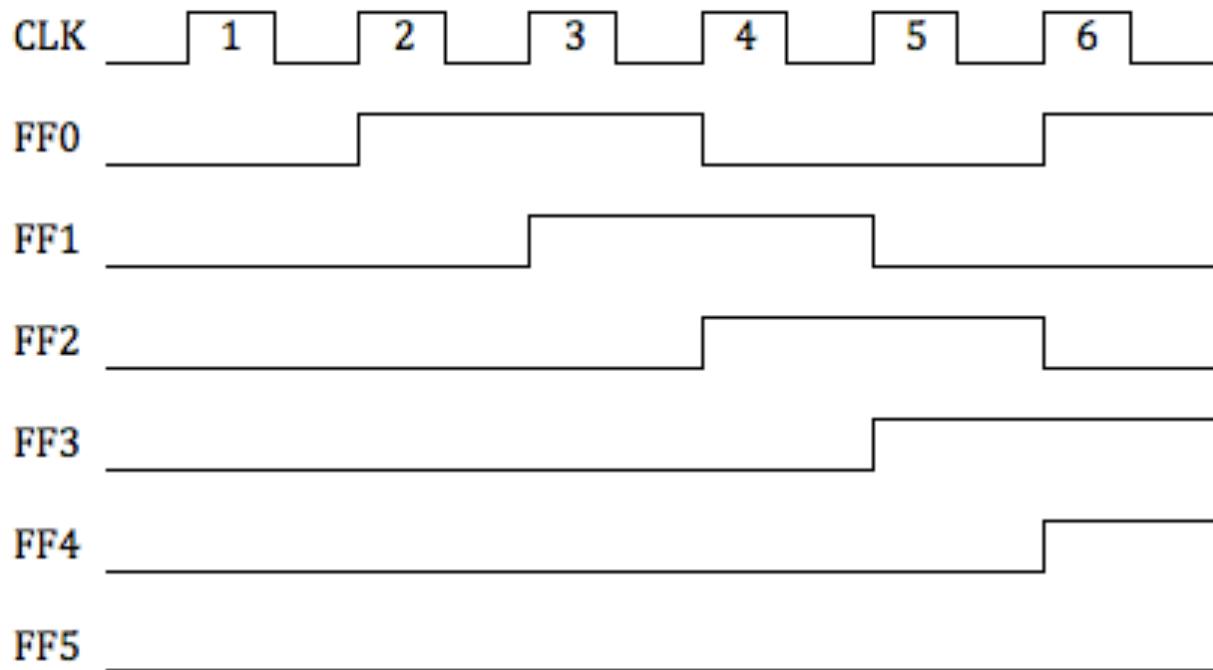
a) Timing diagram

b) The valid output

c) $\text{LSB} \rightarrow Q_4 / \text{FF4}$

Extra

- Exercise 9.3:** a) Determine how many bits entered to the SIPO register?
b) Circle on the timing diagram the valid output.
c) Determine the data entered. (Assume LSB shifted in first)

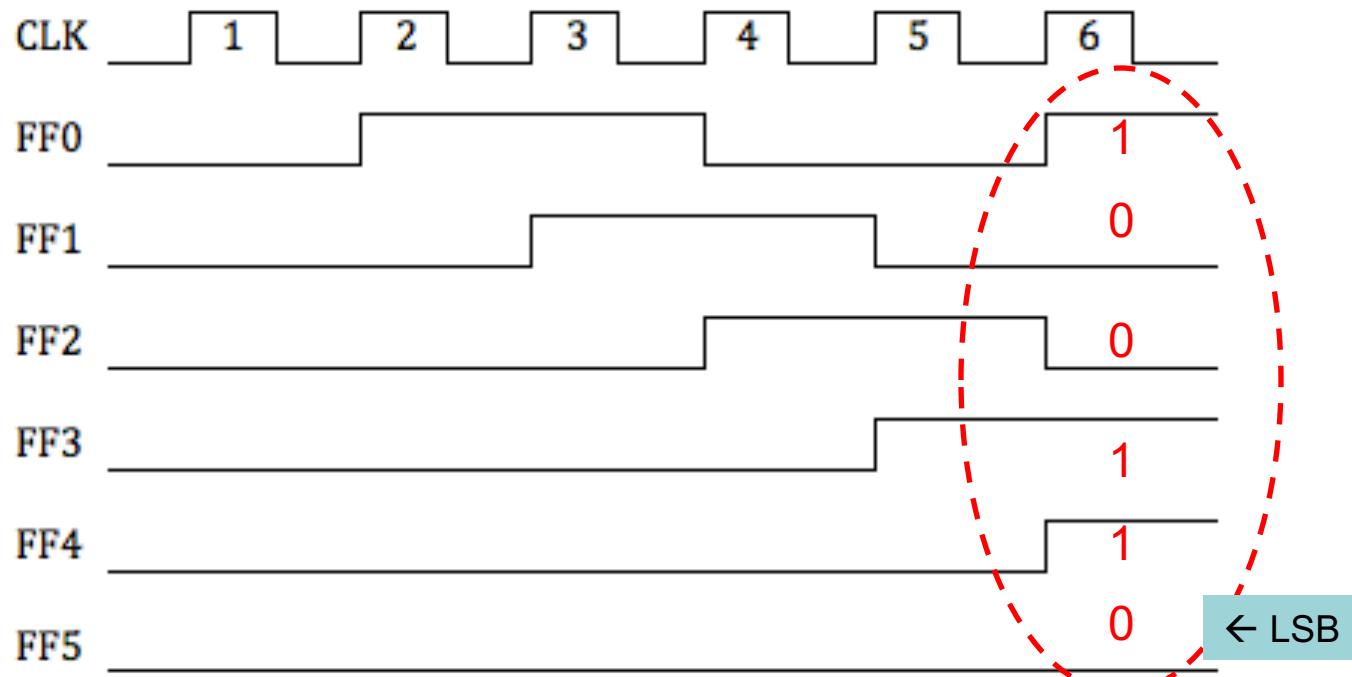


Extra

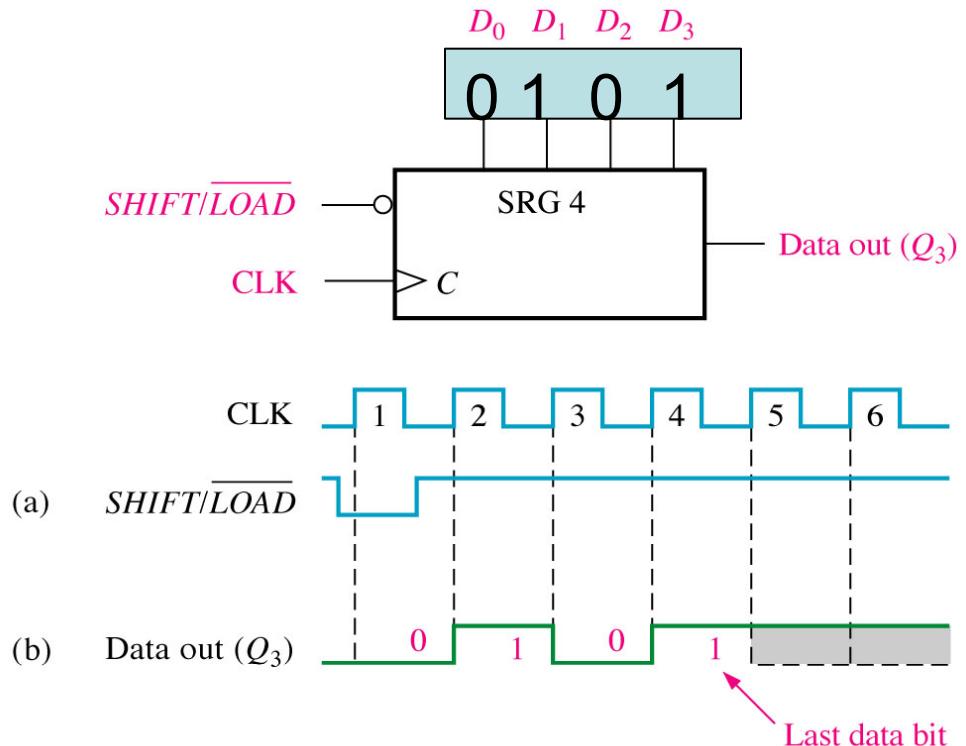
- Exercise 9.3:** a) Determine how many bits entered to the SIPO register?
b) Circle on the timing diagram the valid output.
c) Determine the data entered. (Assume LSB shifted in first)

Solution:

- a) 6 bits
- b) The valid output:
- c) 100110



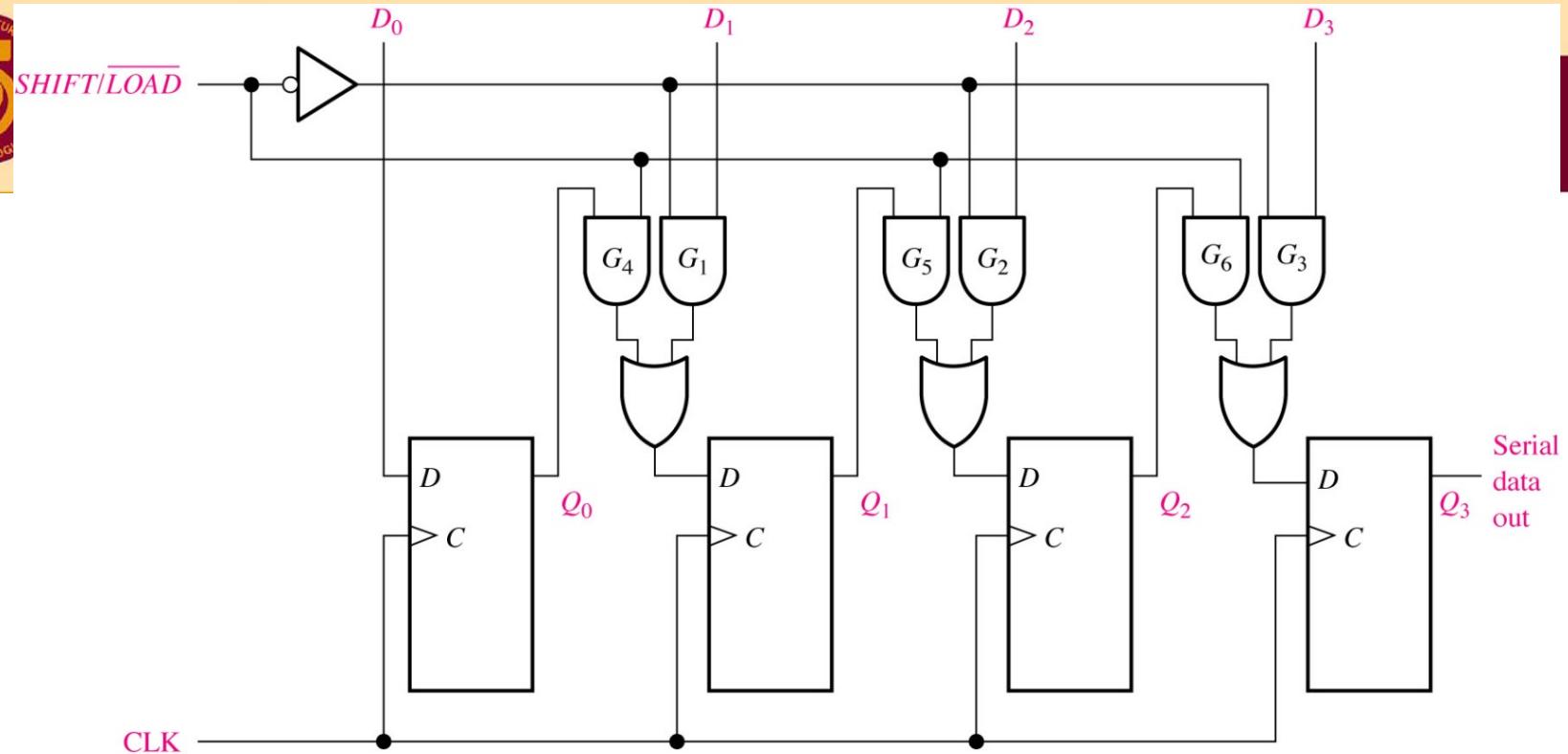
- Inputs entered simultaneously into respective stages on parallel lines
- MSB shifted first
- Outputs are one bit at a time



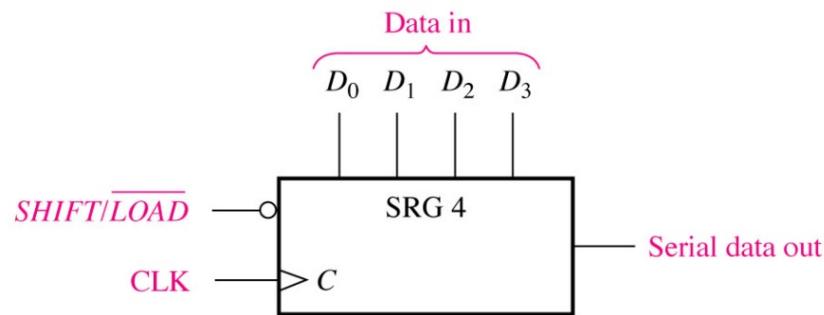
Mode	Value	Function
LOAD	0	Loads the input values
SHIFT	1	Shifts them out at clock pulse

NOTES:

- Input pins: $D_3D_2D_1D_0$
- MSB appear at Data out (Q_{n-1})
- After $n+1$ clock cycle, all n bits data will lost.



(a) Logic diagram

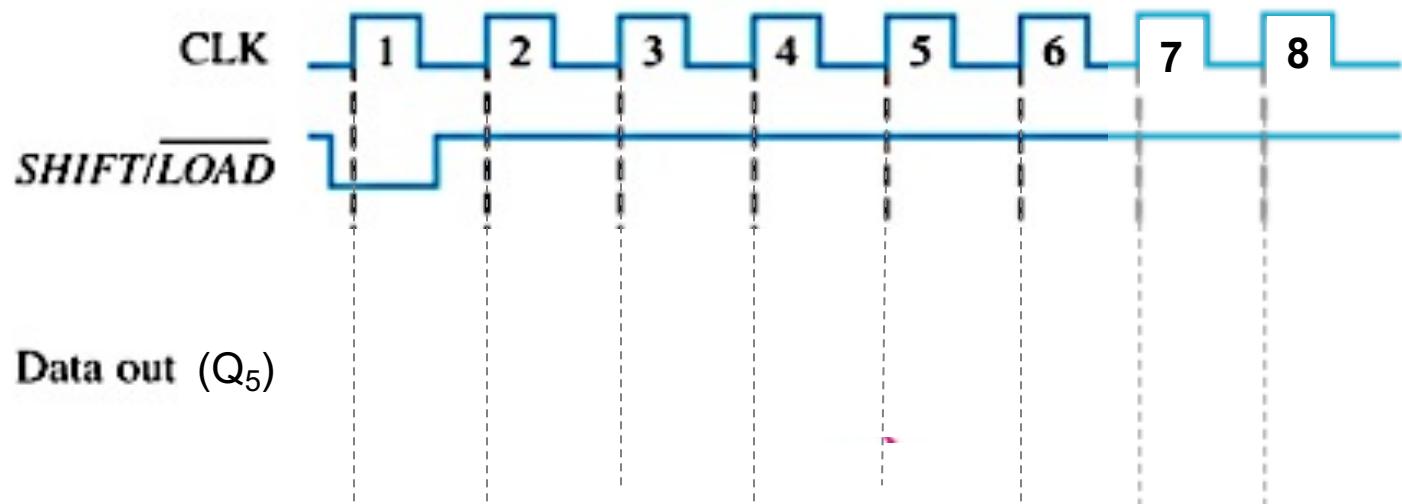


(b) Logic symbol

Exercise 9.4: Complete the timing diagram of 6-bit PISO shift register (SRG 6) for the data input 100110 (MSB shifted first).

Extra

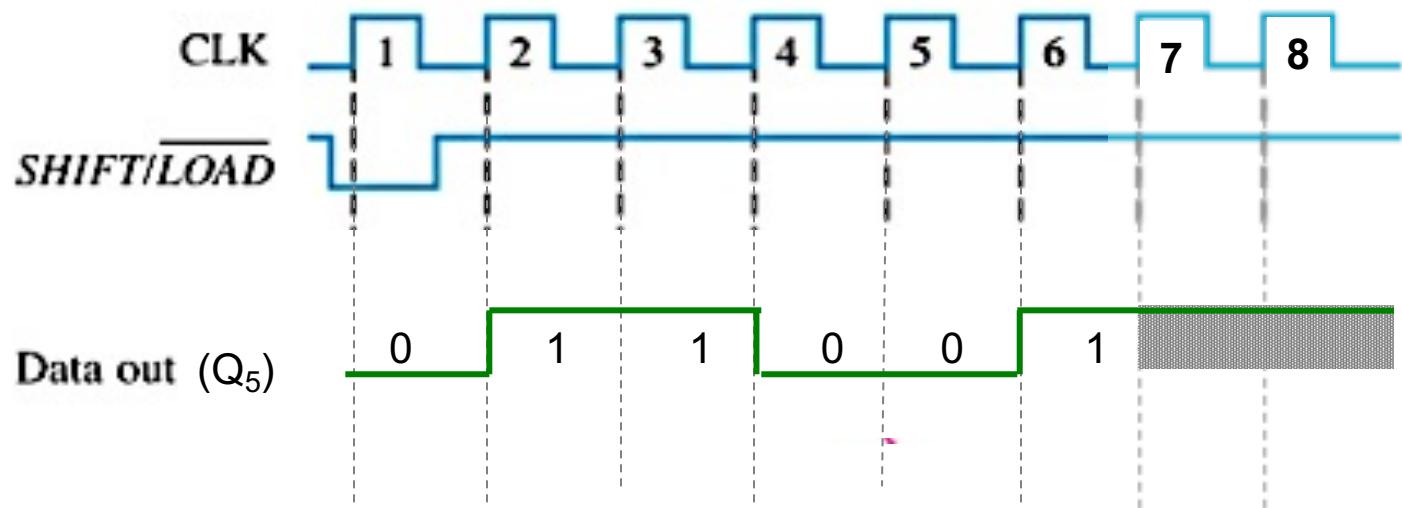
The register initially all is 0s.



Exercise 9.4: Complete the timing diagram of 6-bit PISO shift register (SRG 6) for the data input 100110 (MSB shifted first).

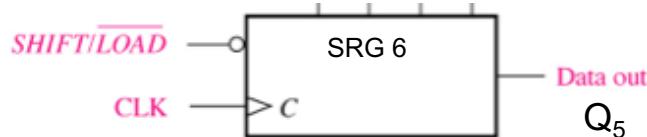
Extra

The register initially all is 0s.

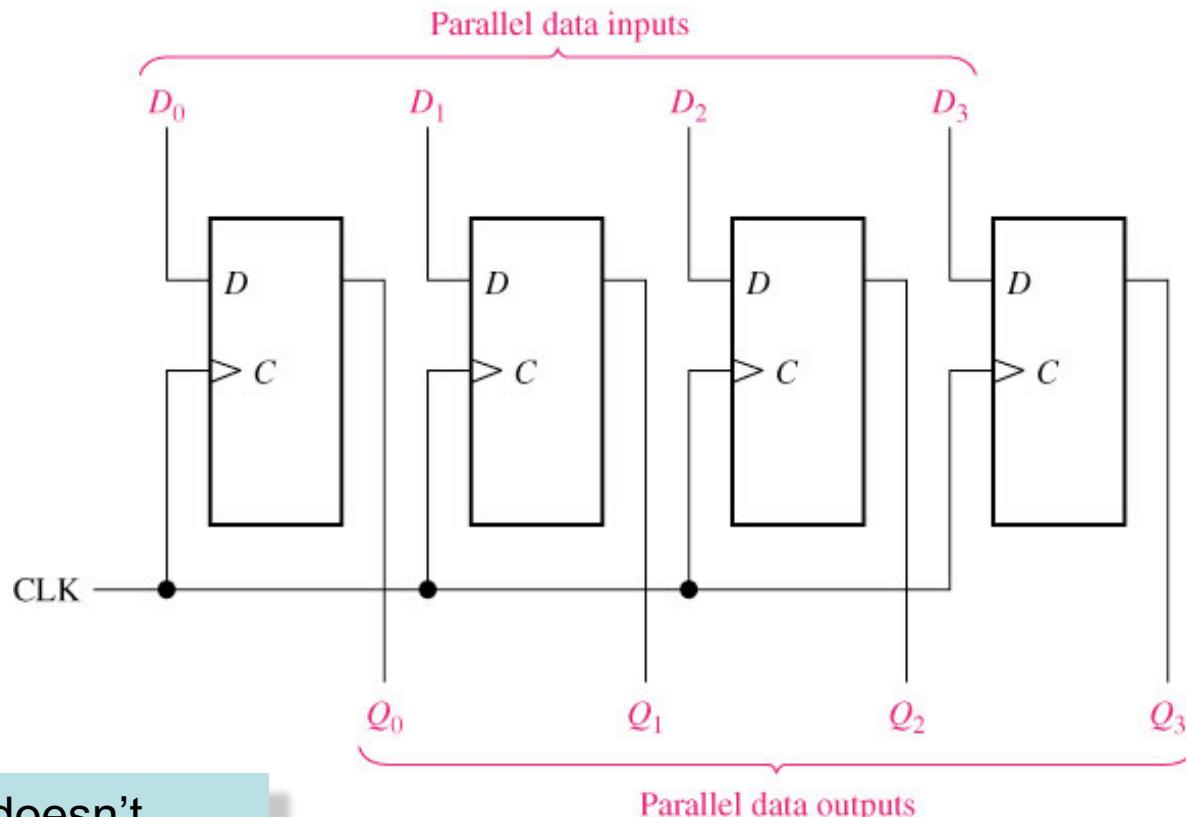


Solution:

D₀D₁D₂D₃D₄D₅
0 1 1 0 0 1



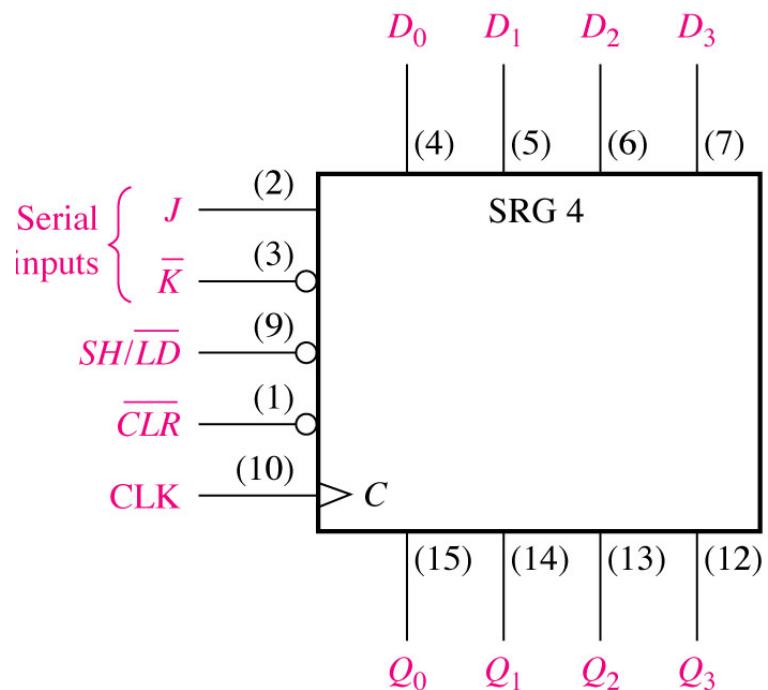
- Input and output done in parallel.
- Enter all inputs - Bits appear on the parallel outputs



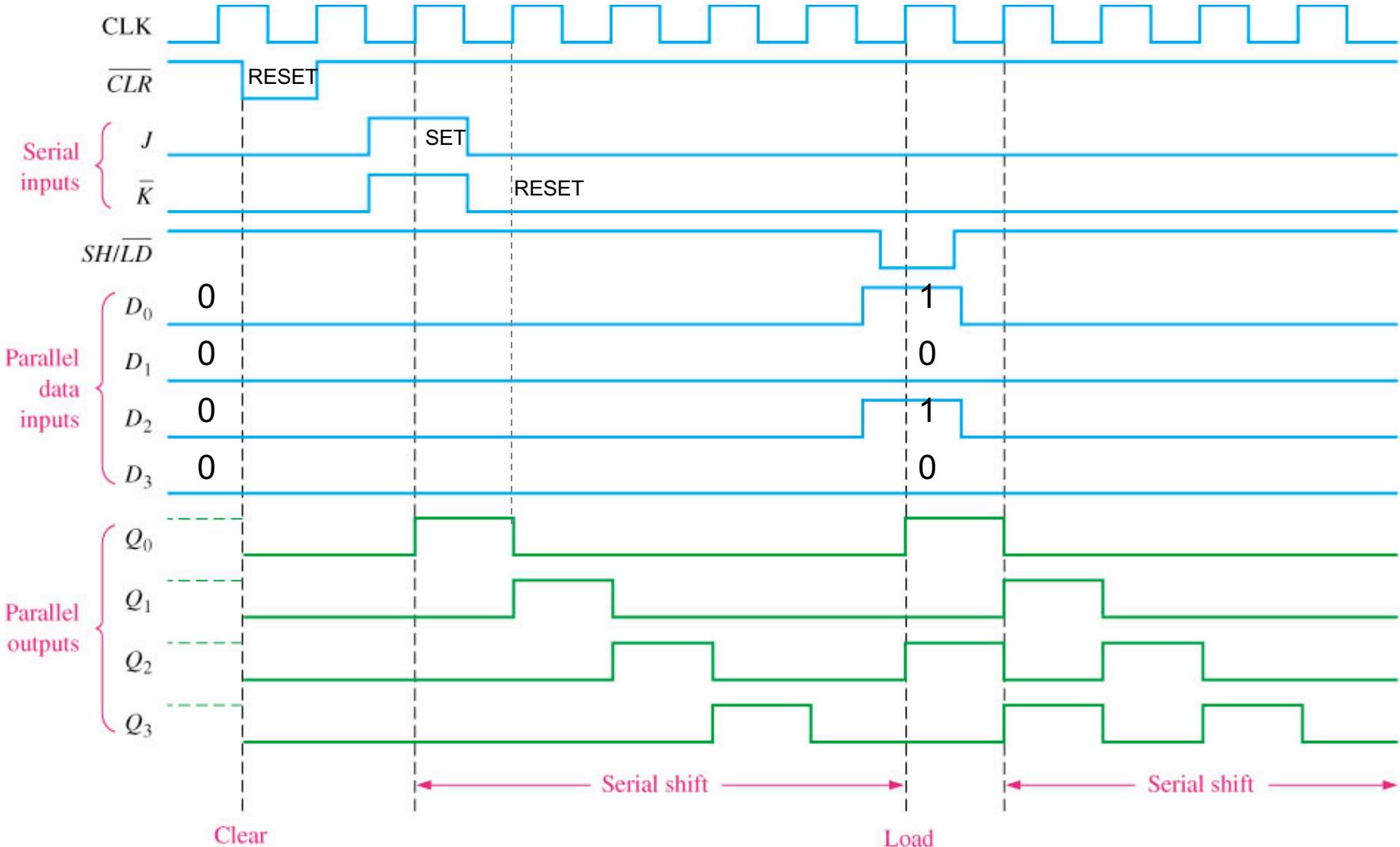
- The parallel data input doesn't appear at the output.
- Output appear at the positive edge of the clock.

Example 6: PIPO (4-bit shift register)

- Can perform PIPO, PISO, SIPO and SISO
- SH/\overline{LD}
 - $SHIFT/\overline{LOAD}$ input
 - When LOW, data from inputs are loaded
 - When HIGH, data will be shifted to outputs ($Q_0 - Q_3$)
- J and K are serial data inputs
 - First stage serial input into Q_0
 - Serial output will be at Q_3



continue...

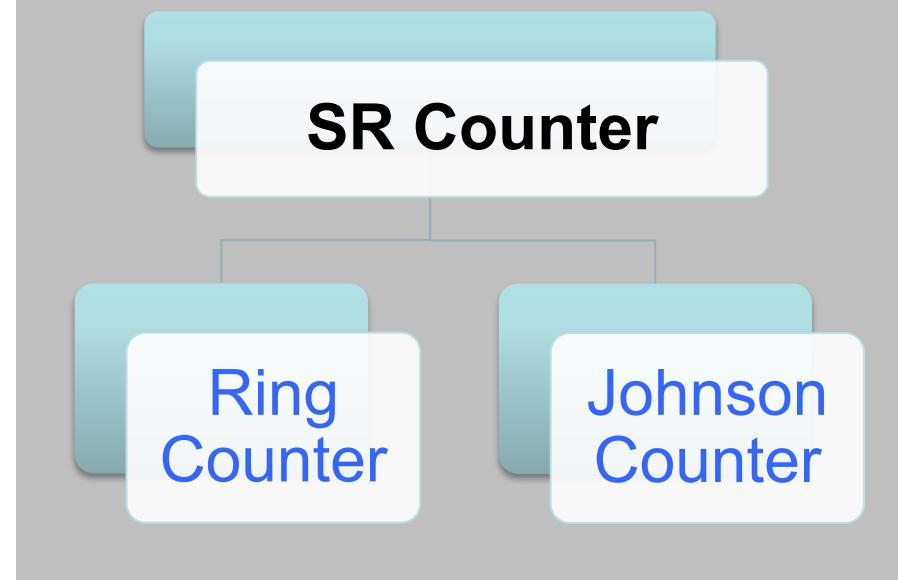


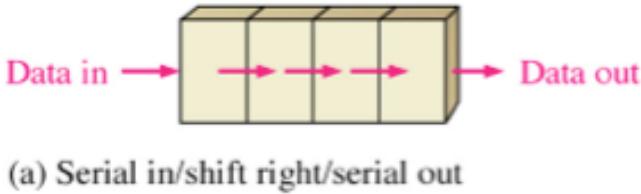


Shift Register Counters

- A shift register with serial output connected back to the serial input to produce special sequences.
- In normal counter, to provide individual digit outputs instead of a binary or BCD output.
→ adding a **decoder**.
- But much simpler to use a different counter structure with simple decoder
→ **shift registers**

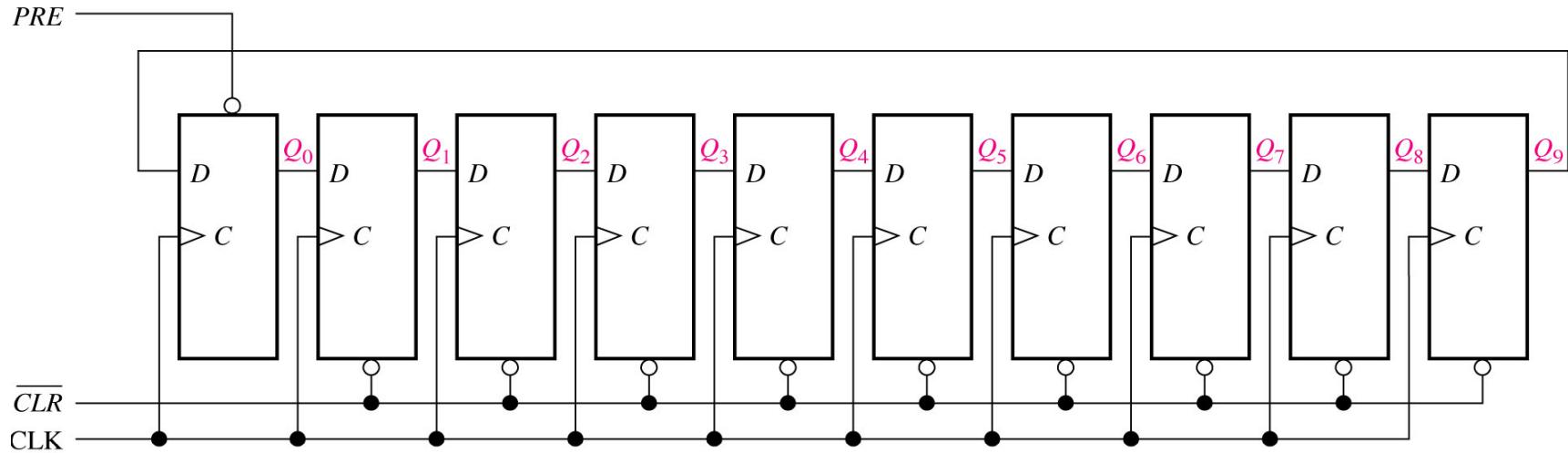
- 2 most common types:





Shift Register Counter: Ring Counter

- A type of **SISO shift register** but final output fed back to the first input.
- The ring counter uses 1 FF for each state in its sequence
- For a **10-bit ring counter**, there is a unique output for each decimal digit
 - Initially $Q_0 = 1 \rightarrow$ represents a zero (Preset)
 - The ‘1’ is shifted round the ring, so next $Q_1 = 1$ (represents a one)
 - This goes on till $Q_9 = 1$.
 - Then the output Q_9 is input back into the first FF
- *****The ‘1’ is always retained and goes ‘round the ring’ at each clock pulse***



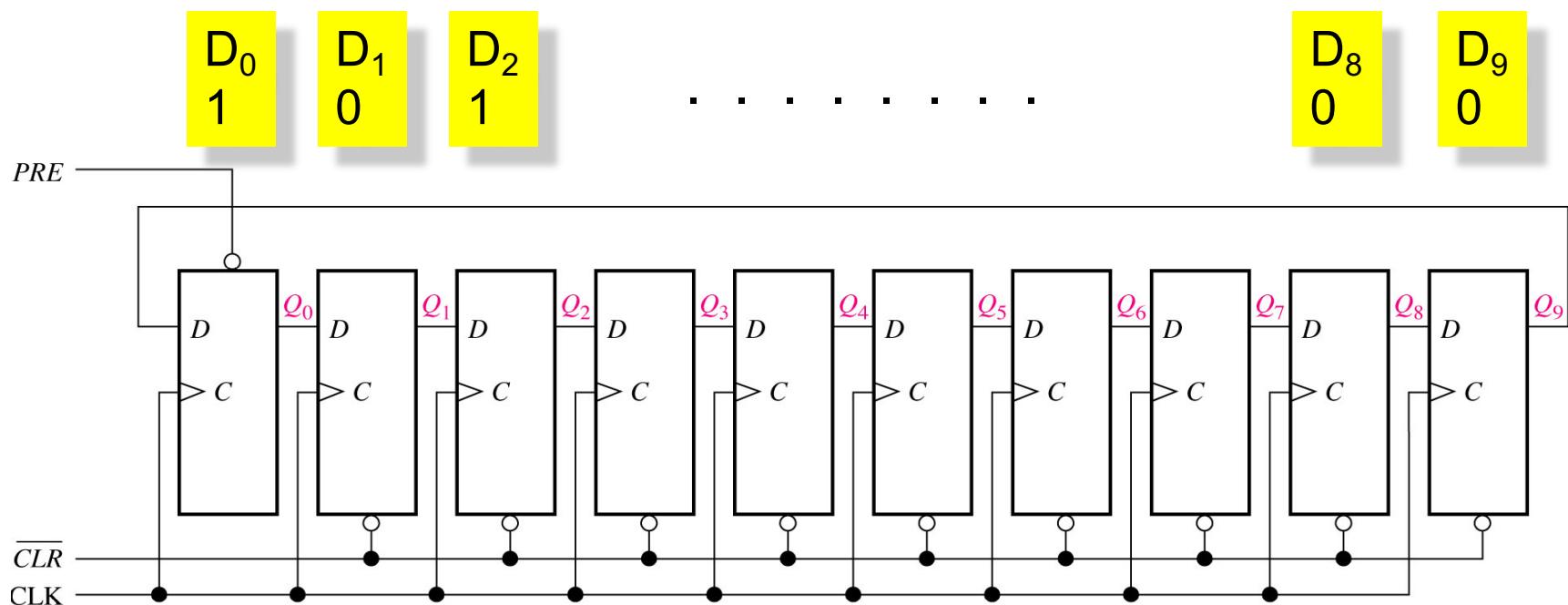
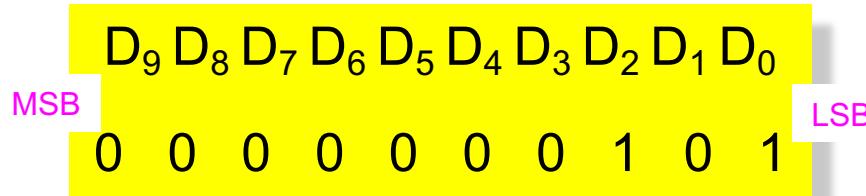
$$n\text{-bit} = \text{MOD } n \\ (\text{state number})$$

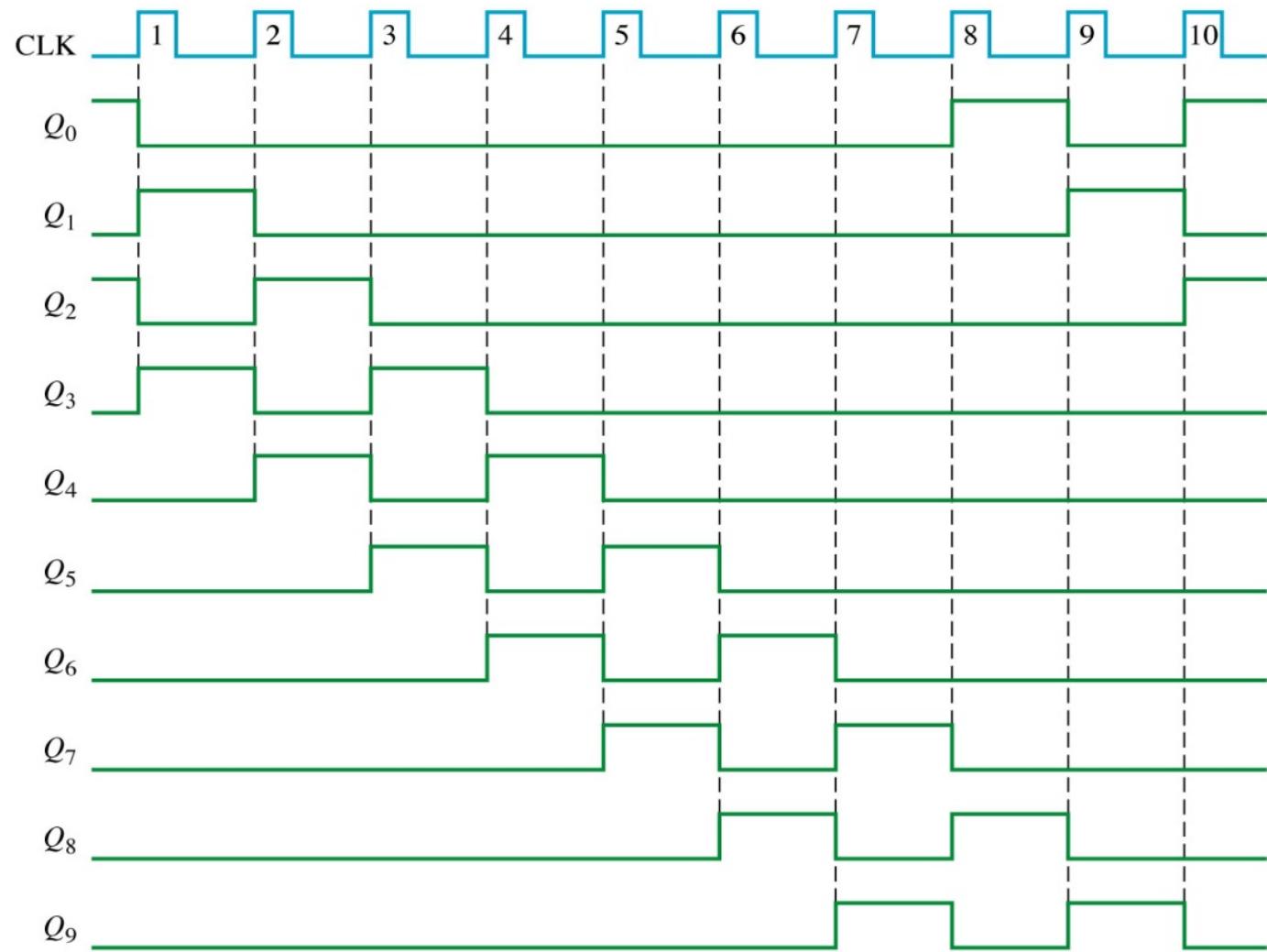
Example:

- For a 10-bit ring counter, there are 10 FFs which make a MOD 10 counter.
- It will recycle after 10 clock cycles.

Example 7: If a 10-bit ring counter has the initial state 0000000101, determine the waveform for each of the Q outputs.

Solution:





Clock Pulse	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
0	0	0	0	0	0	0	0	1	0	1
1	1	0	0	0	0	0	0	0	1	0
2										
3										
4										
5										
6										
7										
8										
9										

Q_9

Clock Pulse	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
0	0	0	0	0	0	0	0	1	0	1
1	1	0	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	0	1
3										
4										
5										
6										
7										
8										
9										

Q_9

Clock Pulse	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	Q ₇	Q ₈	Q ₉
0	0	0	0	0	0	0	0	1	0	1
1	1	0	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	0	1
3	1	0	1	0	0	0	0	0	0	0
4	0	1	0	1	0	0	0	0	0	0
5										
6										
7										
8										
9										

Clock Pulse	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	Q ₇	Q ₈	Q ₉
0	0	0	0	0	0	0	0	1	0	1
1	1	0	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	0	1
3	1	0	1	0	0	0	0	0	0	0
4	0	1	0	1	0	0	0	0	0	0
5	0	0	1	0	1	0	0	0	0	0
6										
7										
8										
9										

Clock Pulse	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
0	0	0	0	0	0	0	0	1	0	1
1	1	0	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	0	1
3	1	0	1	0	0	0	0	0	0	0
4	0	1	0	1	0	0	0	0	0	0
5	0	0	1	0	1	0	0	0	0	0
6	0	0	0	1	0	1	0	0	0	0
7	0	0	0	0	1	0	1	0	0	0
8	0	0	0	0	0	1	0	1	0	0
9	0	0	0	0	0	0	1	0	1	0

Q_9





Shift Register Counter: Johnson Counter

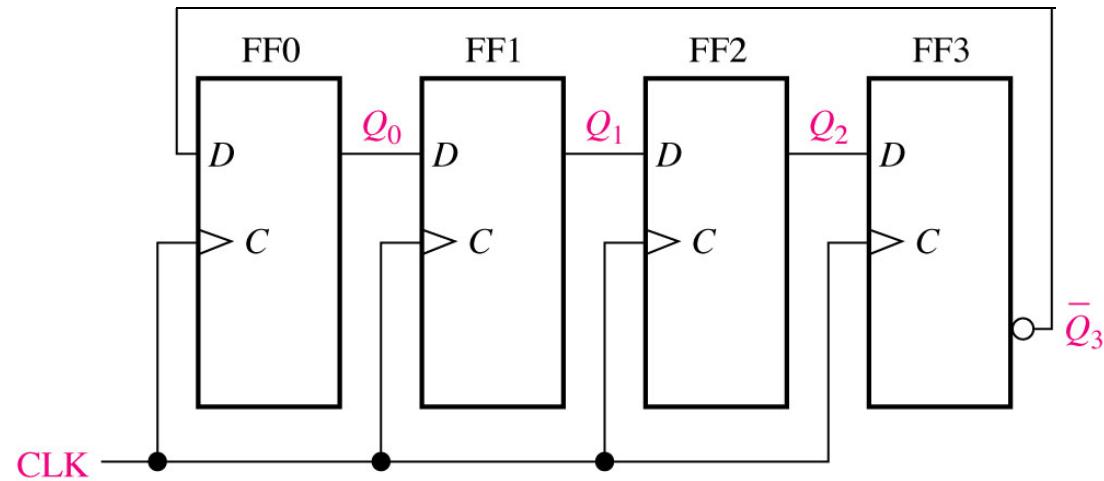
- In the Johnson counter the last complemented output is fed back in as an input to the first FF
- Examples shown with D FF, but can be implemented with other types of FF as well.
- Number of unique states are 2 times the number of bits (FF)
 - 4 bits $\rightarrow 4*2 = 8$ states
 - 5 bits $\rightarrow 5*2 = 10$ states
- Johnson counter will produce a modulus of 2^n ; (n = number of stages)
- **modulus 10 a.k.a. mod 10

$$n\text{-bit} = \text{MOD } 2^n \\ (\text{state number})$$

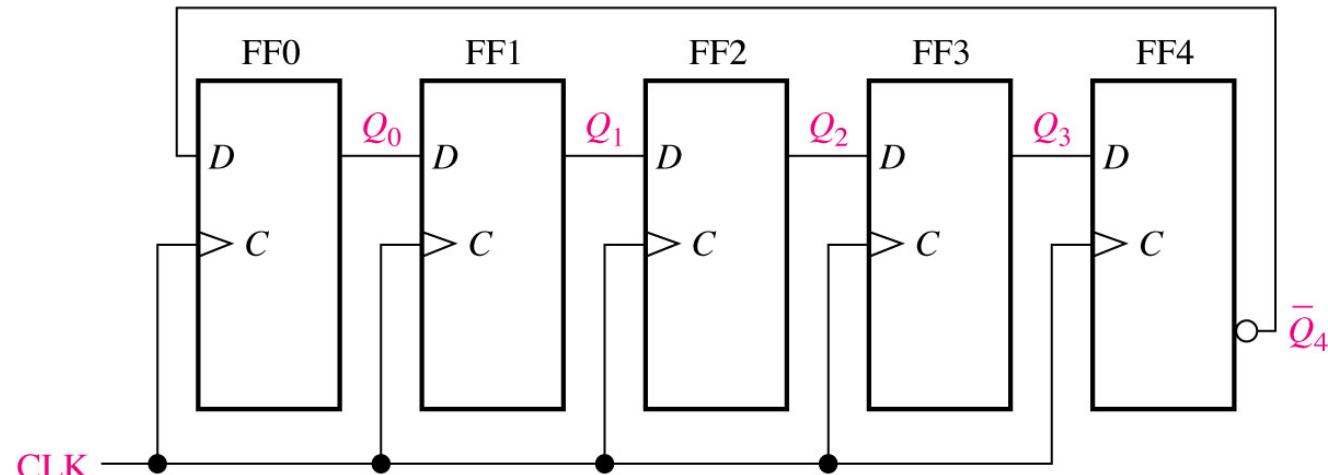
n -bit = MOD 2^n
(state number)

Example:

- For a 4-bit ring counter, there are 4 FFs which make a MOD 8 counter.
- It will recycle after 8 clock cycles.



(a) Four-bit Johnson counter



(b) Five-bit Johnson counter

continue...

Example:

Clock Pulse	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0
9

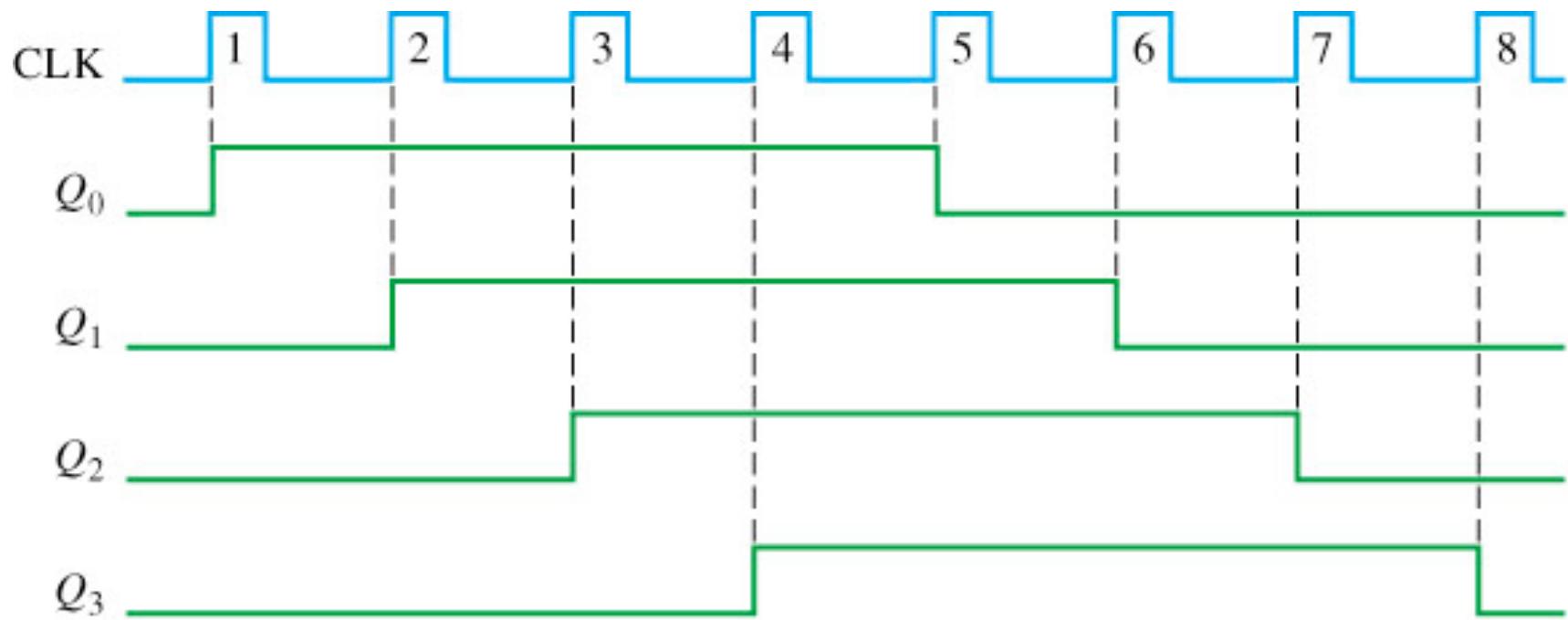
\overline{Q}_3

Complete one cycle

Repeat cycle



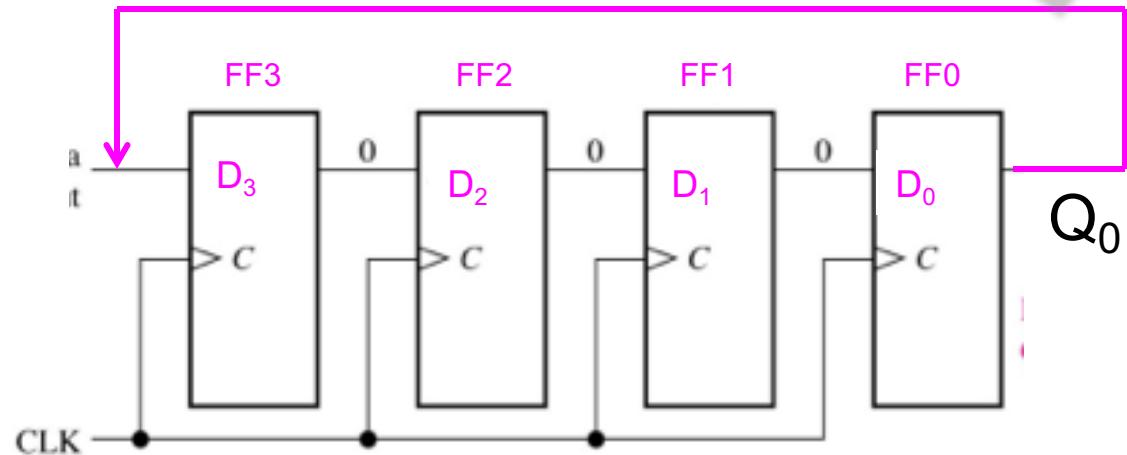
Clock Pulse	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1



Extra

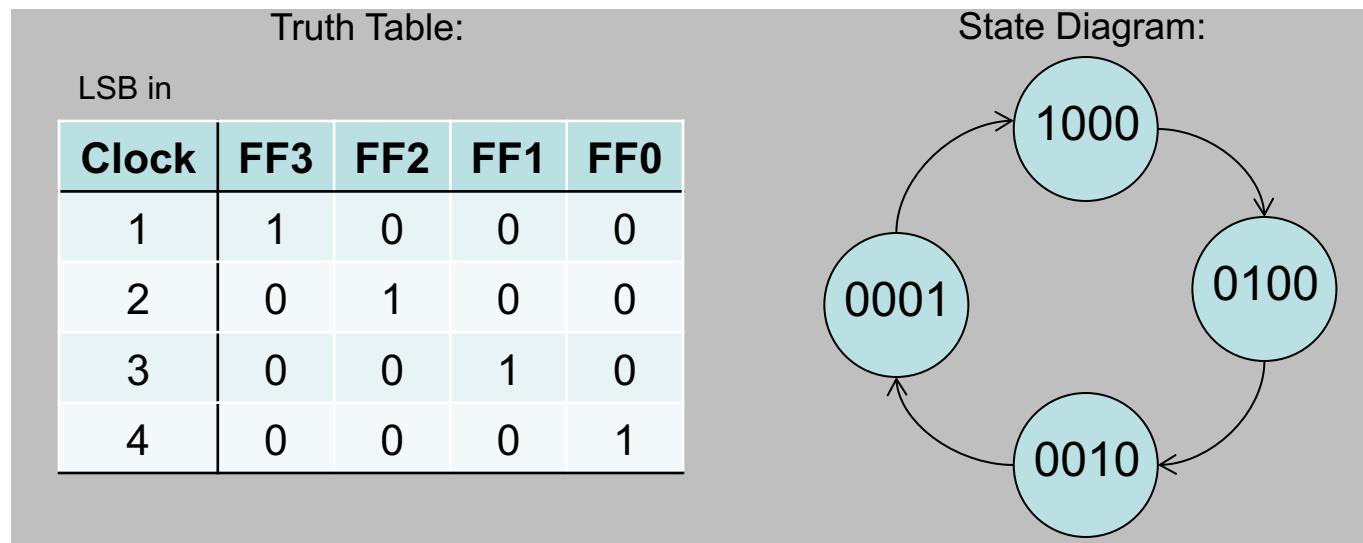
Summary: Ring Counter

n -bit = MOD n
(state number)



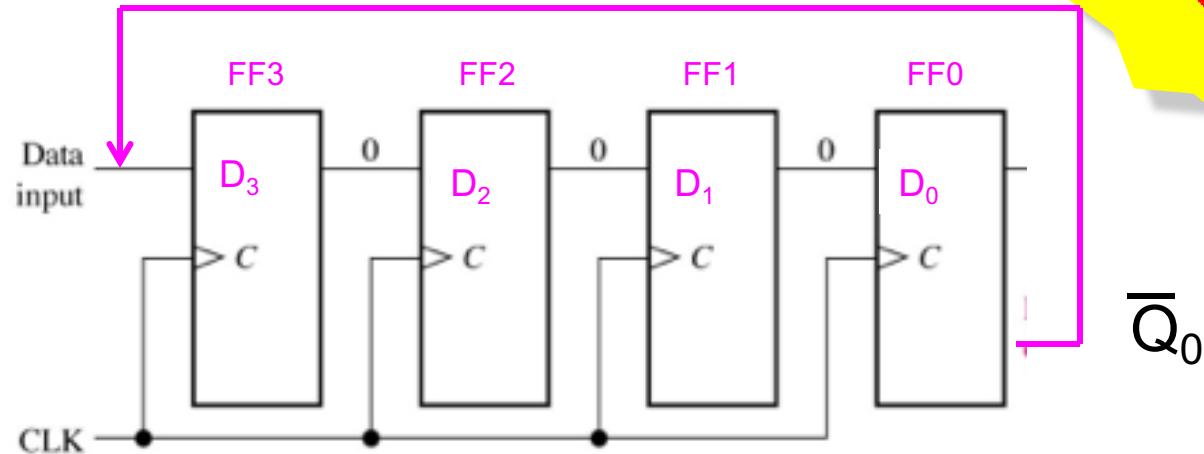
Data representation:

MSB D₃ D₂ D₁ D₀
LSB 0 0 0 1



Summary: Johnson Counter

Extra



n -bit = MOD 2^n
(state number)

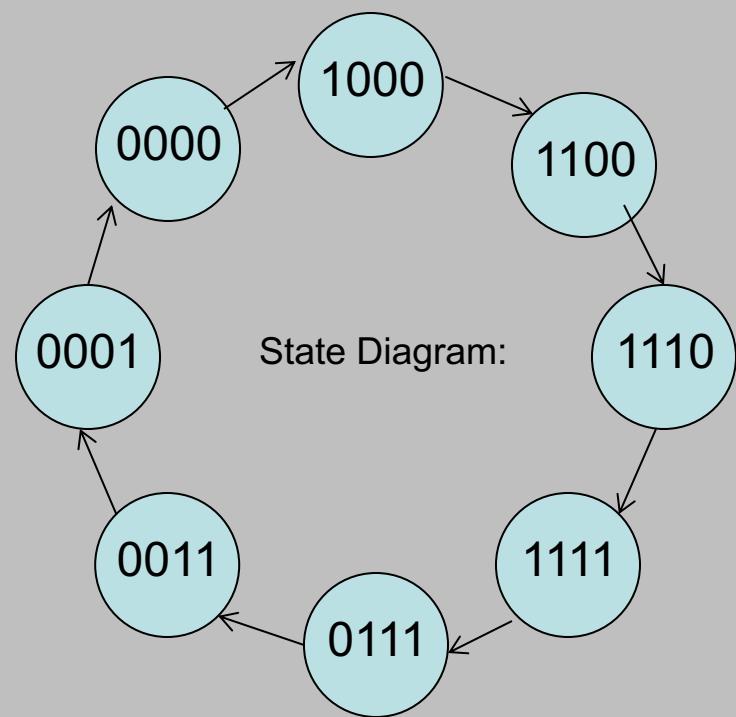
Data representation:

MSB D₃ D₂ D₁ D₀
 0 0 0 1 LSB

Truth Table:

Clock	FF3	FF2	FF1	FF0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

State Diagram:



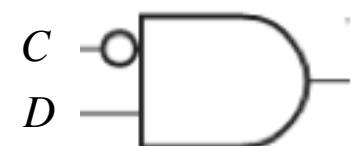
Clock	FF3	FF2	FF1	FF0
	A	B	C	D
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

AND gate required for output
$A\bar{B}$
$B\bar{C}$
$C\bar{D}$
AD
$\bar{A}B$
$\bar{B}C$
$\bar{C}D$ →
$\bar{A}\bar{D}$

Other characteristic of Johnson counter:

- Always exist 2 unique bits compared to other state.
- It will only requires 2 input decoder to decode a state.

Example:





Comparison of a counter

- For a given n flip-flop, the binary counter can produce the most state and ring is the worst
- In term of decoding , Ring counter is the best because there is no need for a decoder to decode each state, binary counter is the worst because require a more complex decoder
- Johnson always in the middle when comparing those feature

	MOD for n flip-flop	Decoder Input
Ring	n	no need for a decoder
Johnson	$2n$	2 input decoder
Binary	2^n	usually > 2 input decoder



Self-Test:

1. Given a 3-bit Johnson counter, draw the appropriate sequence table for 7 clock pulses

2. A modulus-10 Ring counter requires a minimum of _____.

A) 10 FF	C) 5 FF
B) 4 FF	D) 12 FF

3. A modulus-10 Johnson counter requires _____.

A) 10 FF	C) 4 FF
B) 5 FF	D) 12 FF



4. The group of 8 bits 1011 0110 is serially shifted (right most bit first, MSB) into an 8-bit parallel output shift register with an initial state of 1110 1100. After 2 clock pulses, the register contains _____. Assume the first FF is the LSB.
 - A) 1011 0001
 - B) 0100 1101
 - C) 1011 0010
 - D) 1101 1110

5. To serially shift a byte of data into a shift register, there must be _____.
 - A) 1 clock pulse
 - B) 8 clock pulse
 - C) one load pulse
 - D) one clock pulse for each 1 in the data



6. To parallel load a byte of data into a shift register with a synchronous load, there must be
 - A) 1 clock pulse
 - B) 8 clock pulse
 - C) one load pulse
 - D) one clock pulse for each 0 in the data



Solution:

1.

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

2.A,

3.B,

4.C,

5.B,

6.A