

جامعة المستقبل  
نحو جامعة مستدامة



**Al-Mustaqbal University - College of engineering**  
**Department of computer engineering**

**Second stage**

## **Lecture Week 8**

# **“Shift-registers (SISO, SIPO,PISO,PIPO)”**

**By DR. Mohammed Kadhim Rahma**

**2025**

# Objectives Overview

By the end of this lecture, students will be able to

- Define shift registers and identify their basic types (SISO, SIPO, PISO, PIPO).
- Explain the working principles of shift registers using timing diagrams and logic circuits.
- Describe the applications of shift registers in digital systems (e.g., data storage, transfer, and manipulation).

- 1- Basic Shift Register Functions**
- 2- Serial in/serial out shift registers**
- 3- Serial in/parallel out shift registers**
- 4- Parallel in/serial out shift registers**
- 5- Parallel in/parallel out shift registers**

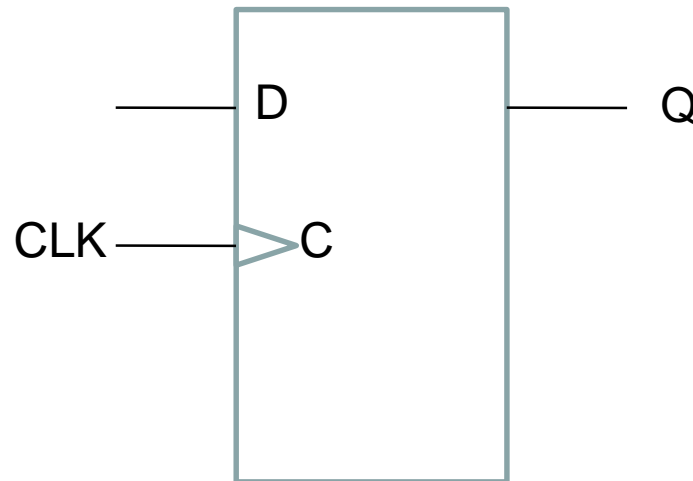
# 1) Basic Shift Register Functions

- A shift register (sequential circuit) is an arrangement of flip-flops and are important in applications in the storage and movement of data in digital system.

The storage capability of a register makes it an important type of memory device.

The following figure illustrate the concept of storing a 1 or 0 in a D flip-flop

**Latch**



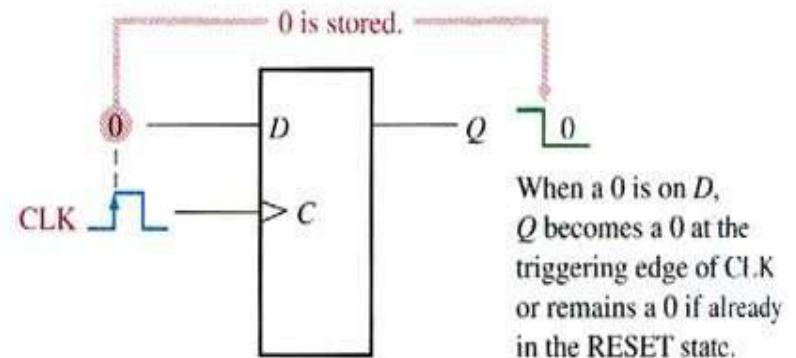
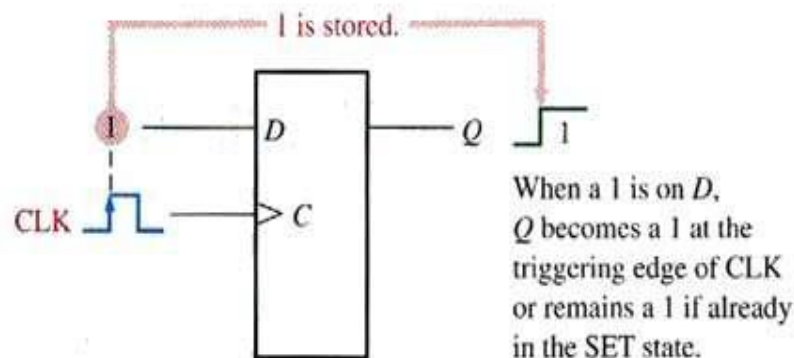
CLK	D	Q
↑	1	1
↑	0	0

**Key Difference # SR flip flop**

SR is solely used for storing and shifting data (1s and 0s) entered into it from an external source and typically do not possess a characteristic internal sequence of states as do counters.

# 1) Basic Shift Register Functions

A 1 is applied to the data input and a clock pulse is applied that stores the 1 by setting the flip-flop. When the 1 on the input is removed, the flip-flop remains in the SET state, thereby storing 1. A similar procedure applies to the storage of a 0 by resetting the flip-flop.



**One stage = one flip-flop = one bit of storage capacity**

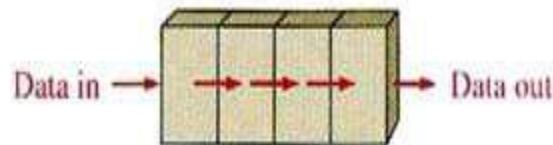
We know that one flip-flop can store one-bit of information. In order to store multiple bits of information, we require multiple flip-flops. The group of flip-flops, which are used to hold store the binary data is known as **register**.

If the register is capable of shifting bits either towards right hand side or towards left hand side is known as **shift register**. An 'N' bit shift register contains 'N' flip-flops. Following are the four types of shift registers based on applying inputs and accessing of outputs.

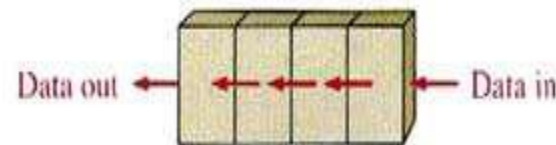
- **Serial In – Serial Out shift register**
- **Serial In – Parallel Out shift register**
- **Parallel In – Serial Out shift register**
- **Parallel In – Parallel Out shift register**

# 1) Basic Shift Register Functions

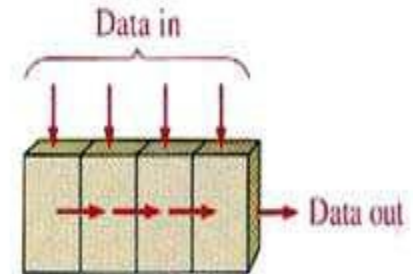
Basic data movement in shift registers. (Four bits are used for illustration. The bits move in the direction of the arrows.)



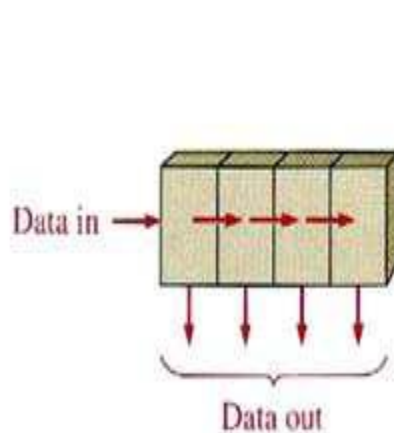
(a) Serial in/shift right/serial out



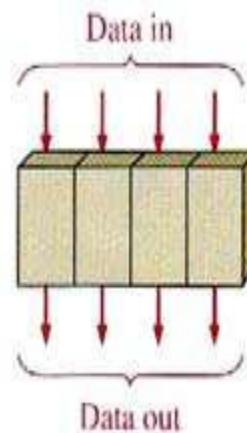
(b) Serial in/shift left/serial out



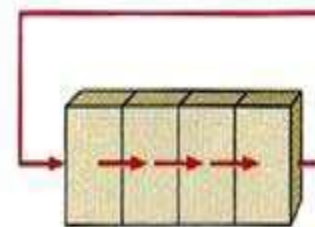
(c) Parallel in/serial out



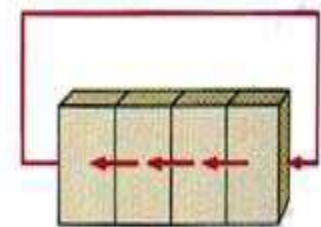
(d) Serial in/parallel out



(e) Parallel in/parallel out



(f) Rotate right



(g) Rotate left

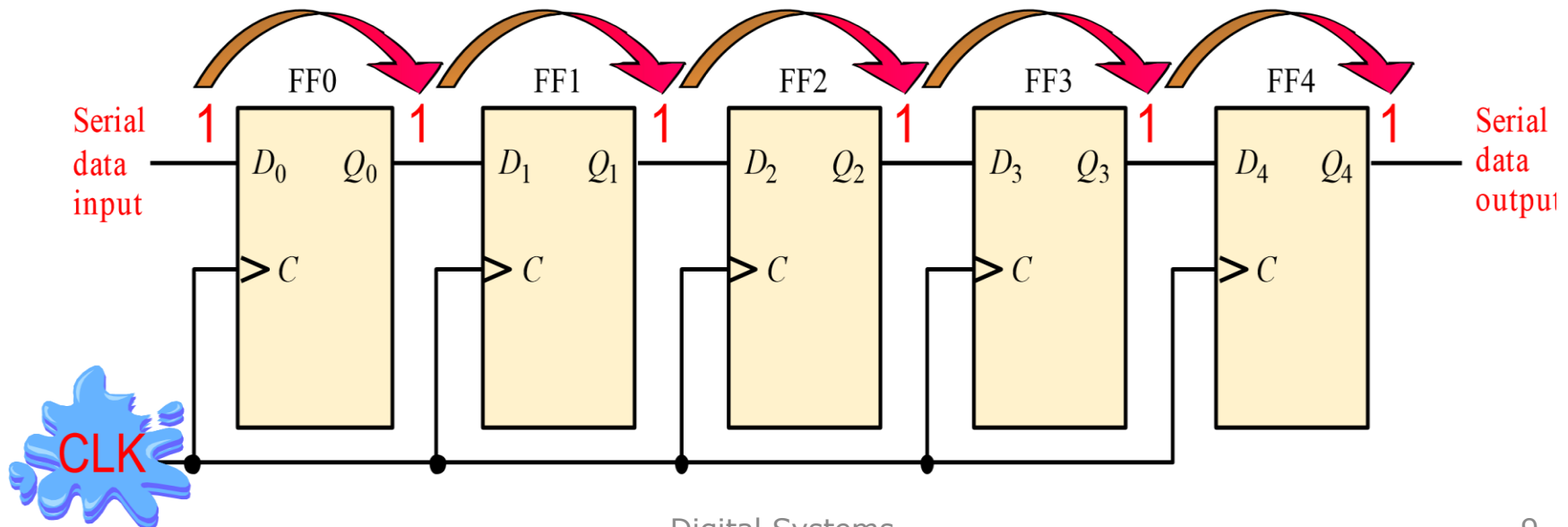


# 1) Basic Shift Register Functions

Shift registers are available in IC form or can be constructed from discrete flip-flops as is shown here with a five-bit serial-in serial-out register.

Each clock pulse will move an input bit to the next flip flop.

For example, a 1 is shown as it moves across.



The Serial In Serial Out shift registers accepts data serially- that is, one bit at a time on a single line. It produces the stored information on its output also in serial form.

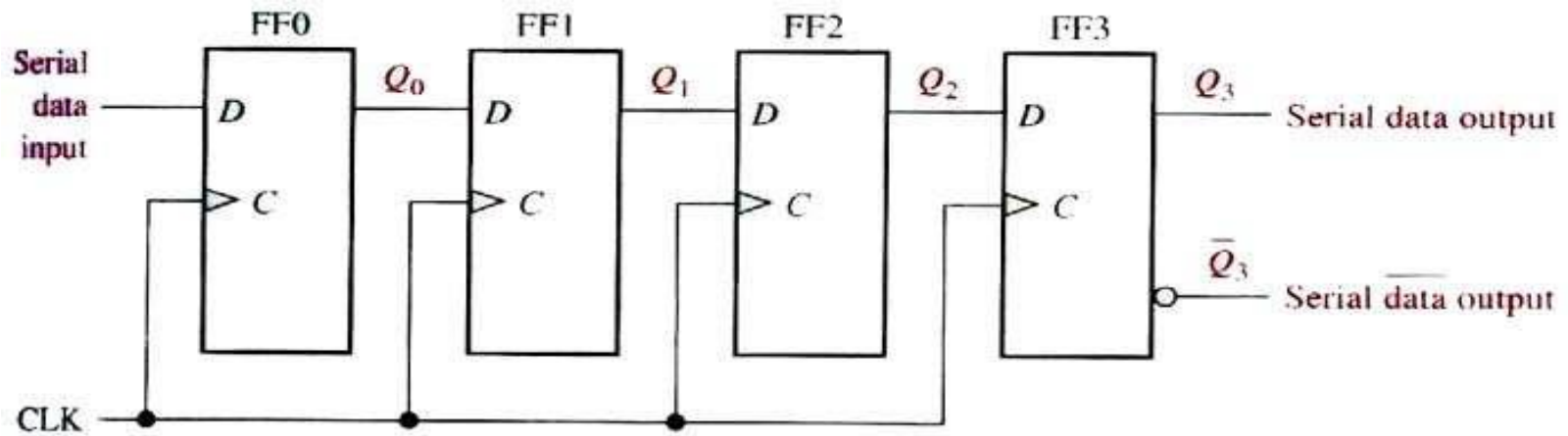
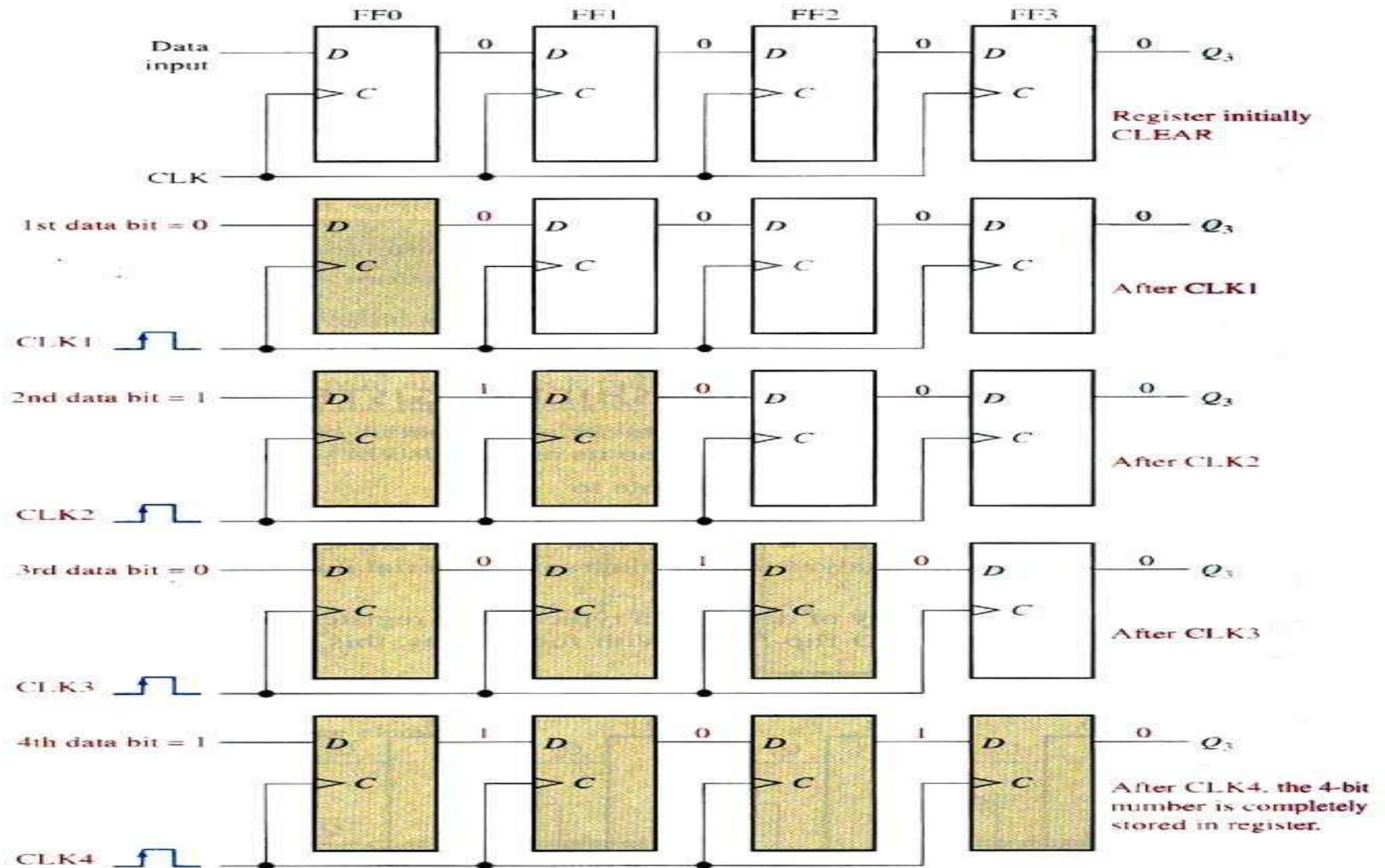
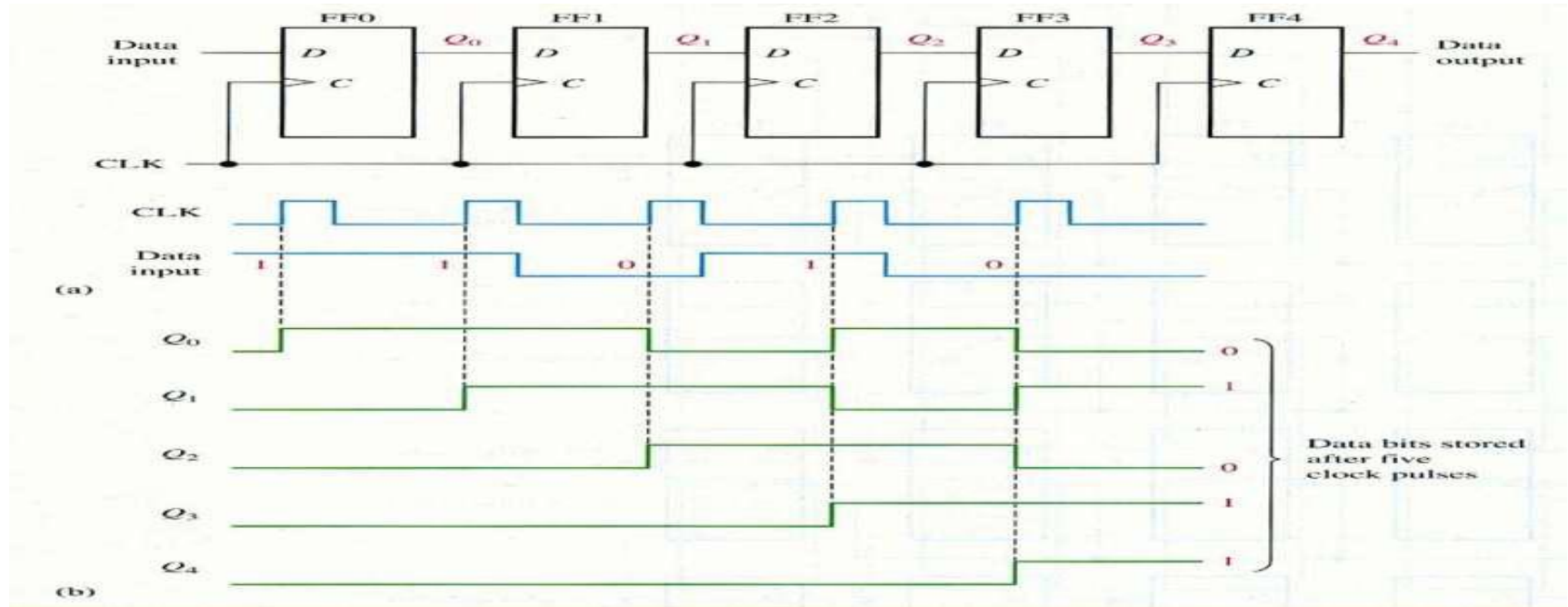


Figure above illustrates entry of the four bits 1010 into the register, beginning with the right-most bit. The register is initially clear. The 0 is put onto the data input line making  $D=0$  for FF0.

- when the first clock pulse is applied, FF0 is reset, thus storing the 0.
- Next the second bit, which is 1, is applied to the data input, making  $D=1$  for FF0 and  $D=0$  for FF1 because the D input of FF1 is connected to the Q0 output. When the second clock pulse occurs, the 1 on the data input is shifted into FF0, causing FF0 to set; and the 0 that was in FF0 is shifted into FF1.
- The third bit, a 0 is now put onto the data-input line, and a clock pulse is applied. The 0 is entered into FF0, the 1 stored in FF0 is shifted into FF1, and the 0 stored in FF1 is shifted onto FF2 .
- The last bit, a 1 is now applied to the data input, and a clock pulse is applied. This time the 1 is entered into FF0, the 0 stored in FF0 is shifted into FF1, the 1 stored in FF1 shifted into FF2, and the 0 stored in FF2 is shifted into FF3. this completes the serial entry of the four bits into the shift register, where they can be stored for any length of time as long as the flip-flops have dc power.

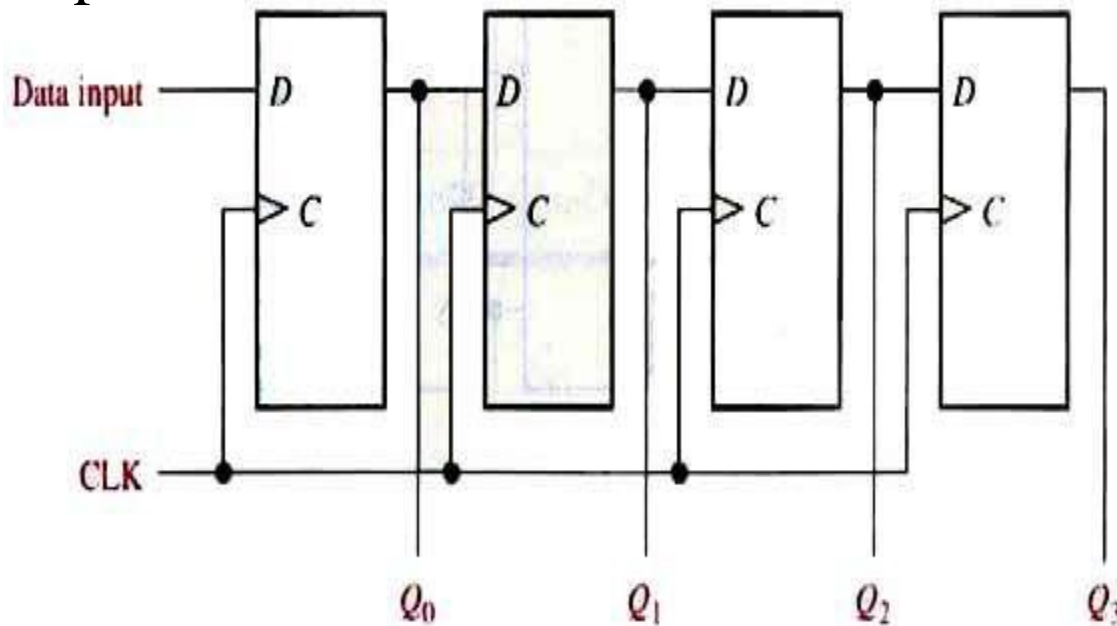


**Example :-** Show the states of the 5-bit register for the specified data input and clock waveforms . Assume that the register is initially cleared (all 0s).

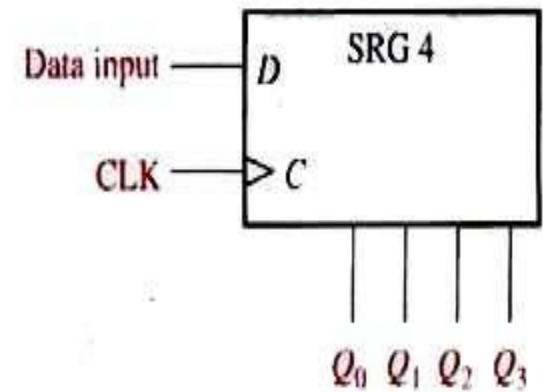


**Solution:-** The first data bit (1) is entered into the register on the first clock pulse and then shifted from left to right as the remaining bits are entered and shifted. The register contains  $Q_4Q_3Q_2Q_1Q_0 = 11010$  after five clock pulses.

The difference of this method is in the way in which the data bits are taken out of the register; in the parallel output register, the output of each stage is available. Once the data are stored, each bit appears on its respective output line, and all bits are available simultaneously, rather than on a bit-by-bit basis as with serial output.

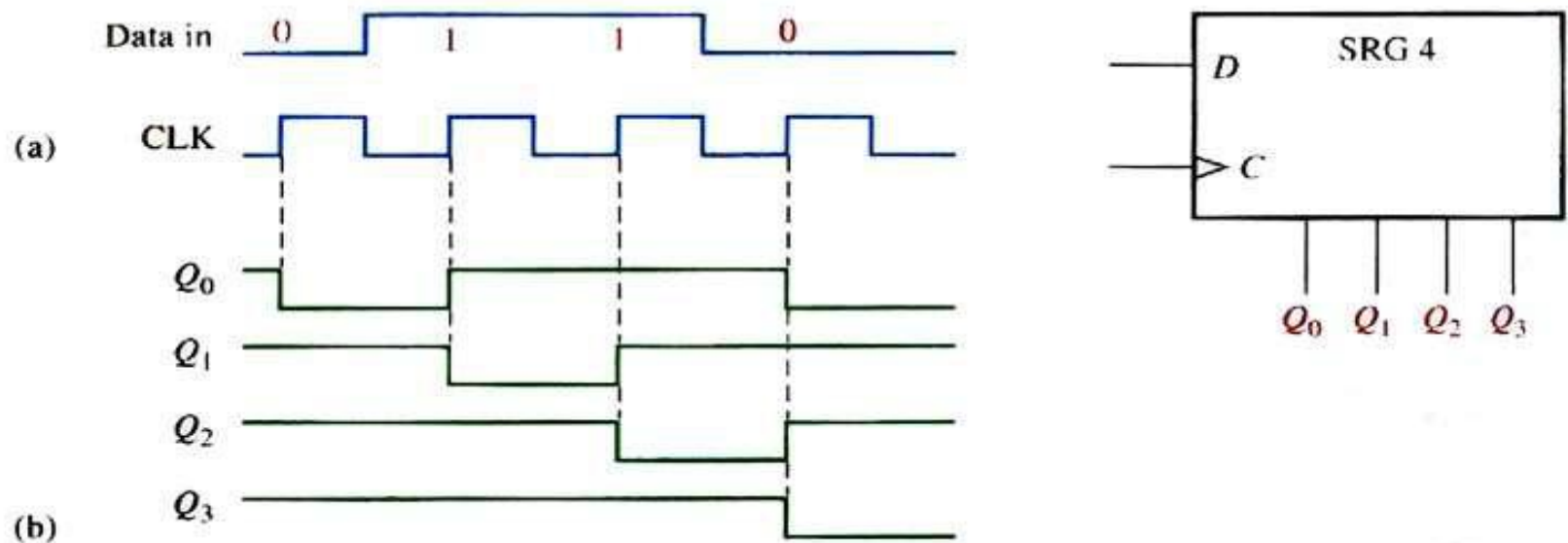


(a)



(b)

**Example :-** Show the states of the 4-bit register (SRG 4) for the data input and clock waveforms. The register initially contains all 1s.

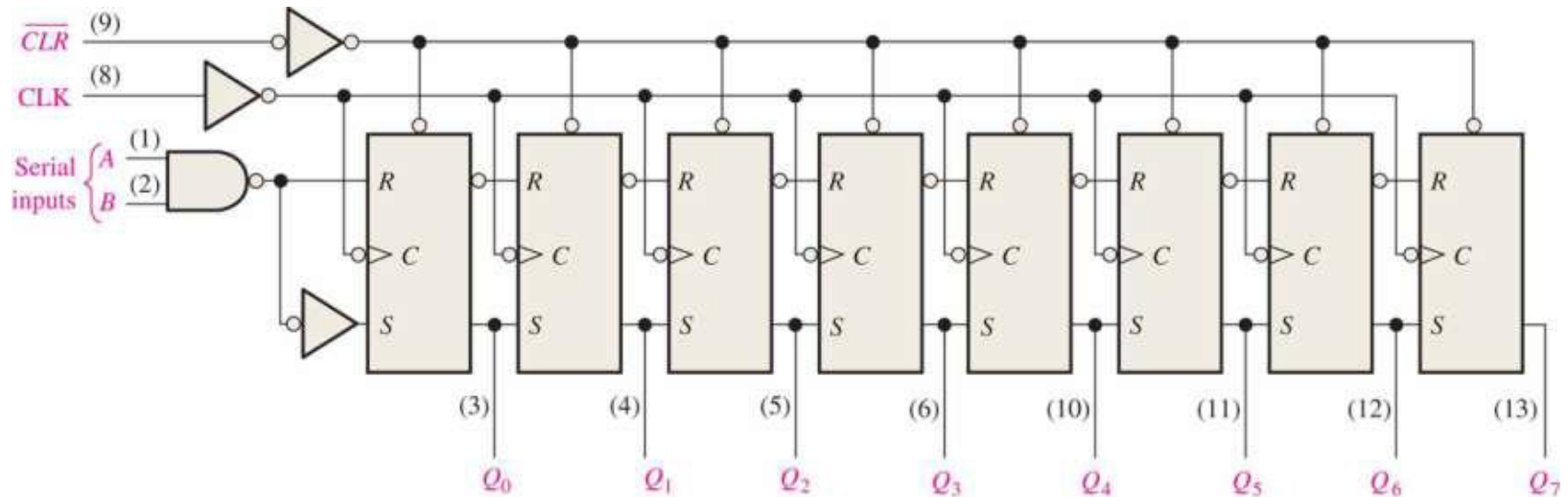


**Solution :-** The register contains 0110 after four clk pulses

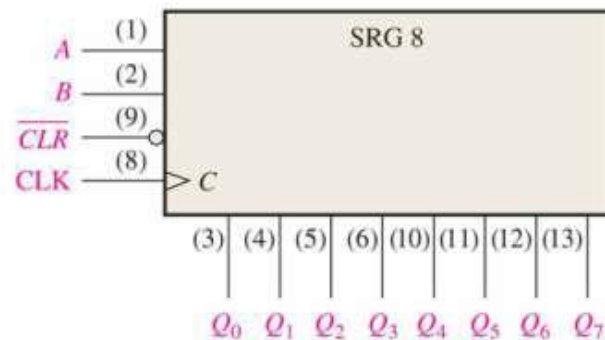


# Example

The 74HC164 8-bit serial in/parallel out shift register.



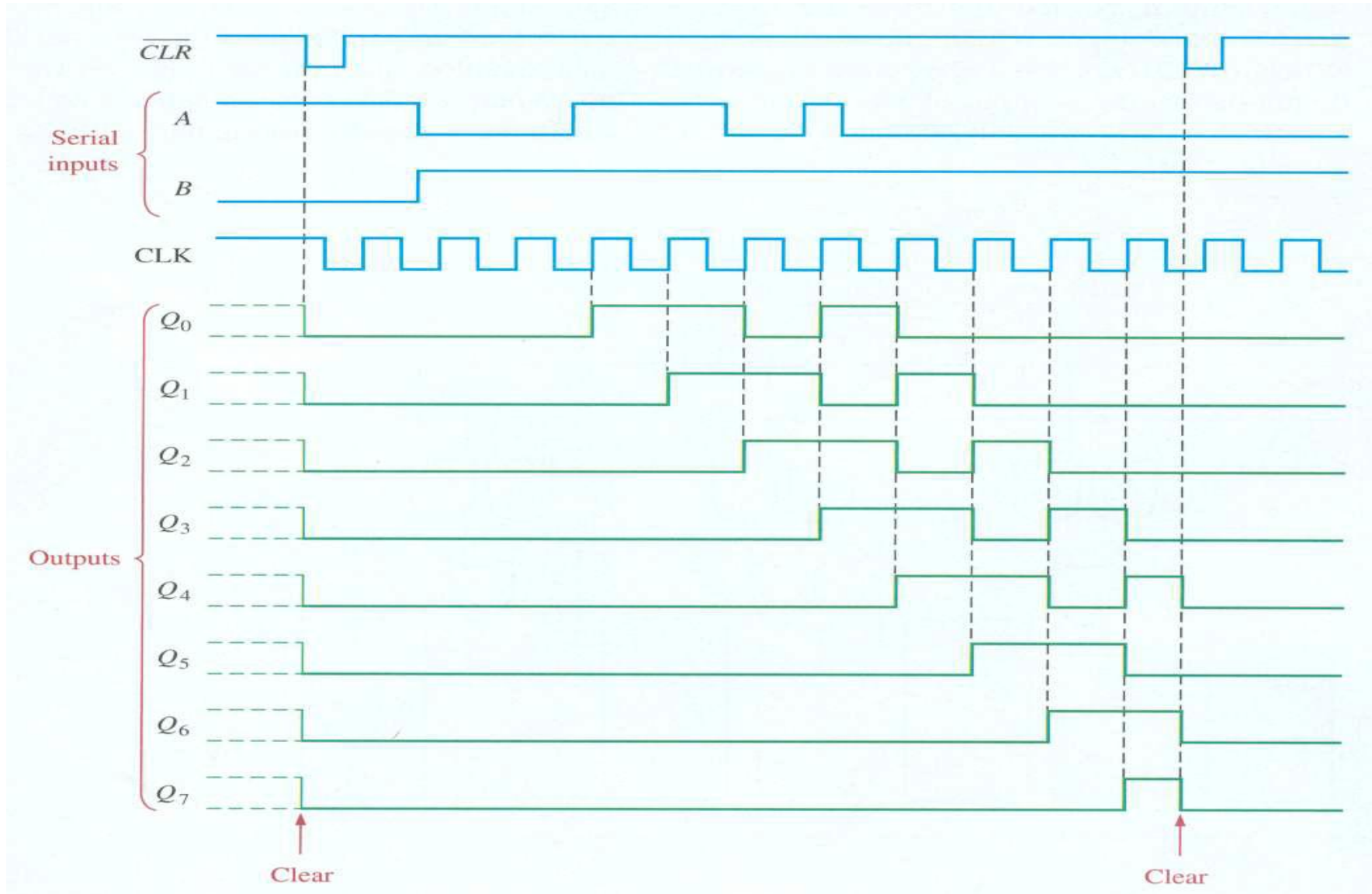
(a) Logic diagram



(b) Logic symbol

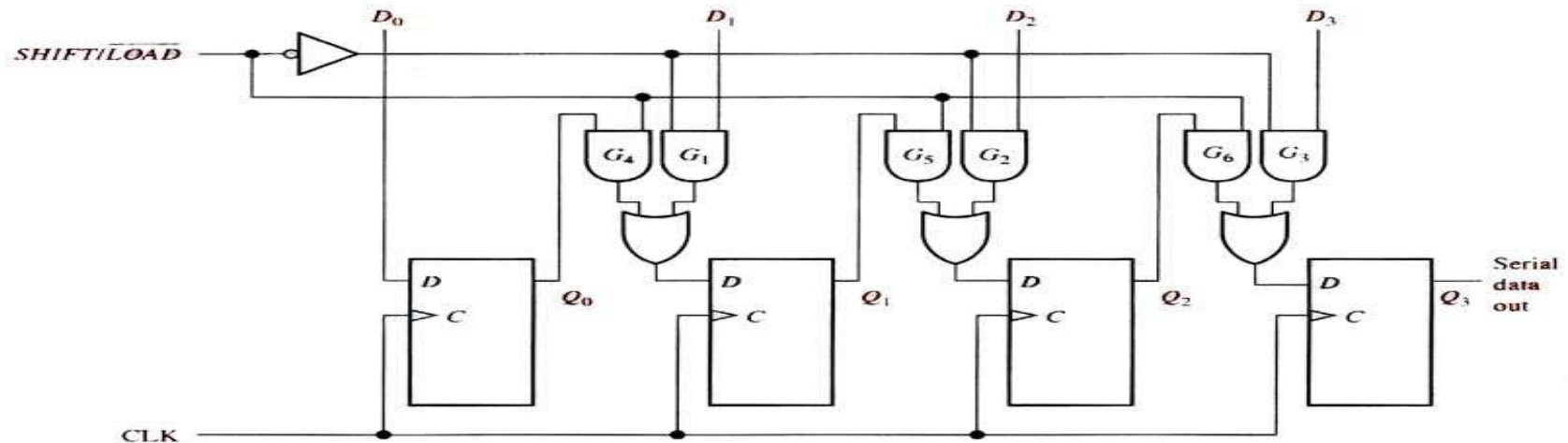


## Timing Diagram

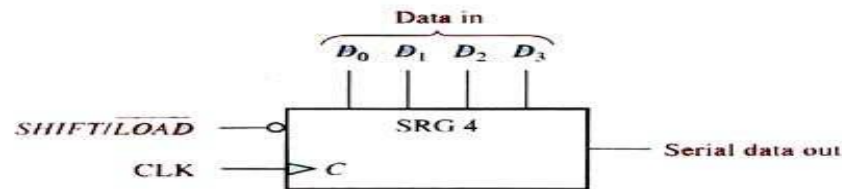


## 4) Parallel In/ Serial Out Shift Registers

For a register with parallel data inputs, the bits are entered simultaneously into their respective stages on parallel lines rather than on a bit-by-bit basis on one line as with serial data inputs.



(a) Logic diagram



(b) Logic symbol

## Parallel In/ Serial Out Shift Registers

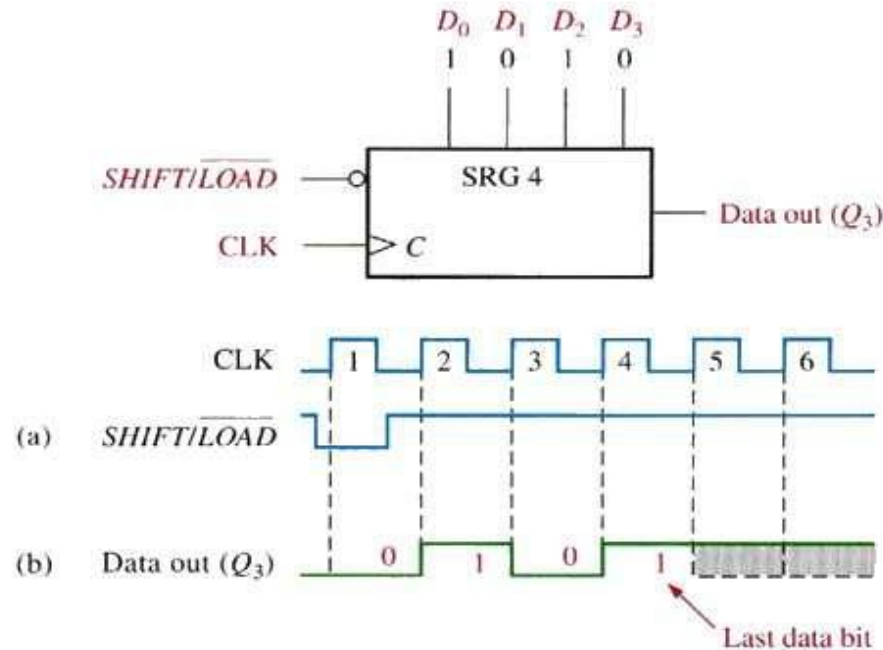
The previous figure shows a 4-bit Parallel In/Serial Out shift register and a typical logic symbol. There are four data lines D0, D1, D2, D3 and a SHIFT/LOAD' input, which allows four bits of data to load in parallel into the register. When Shift/Load' is LOW, gates G1 through G3 are enabled, allowing each data bit to be applied to the D input of its respective flip-flop. When a clock pulse, the flip-flops with D=1 will set and those with D= 0 will reset, thereby storing all four bits simultaneously.

When Shift/Load' is HIGH, gates G1 through G3 are disabled and gates G4 through G6 are enabled, allowing the data bits to shift right from one stage to the next. The OR gates allow either the normal shifting operation or the parallel data-entry operation, depending on which AND gates are enabled by the level on the Shift/Load' input.

# Parallel In/ Serial Out Shift Registers

## EXAMPLE

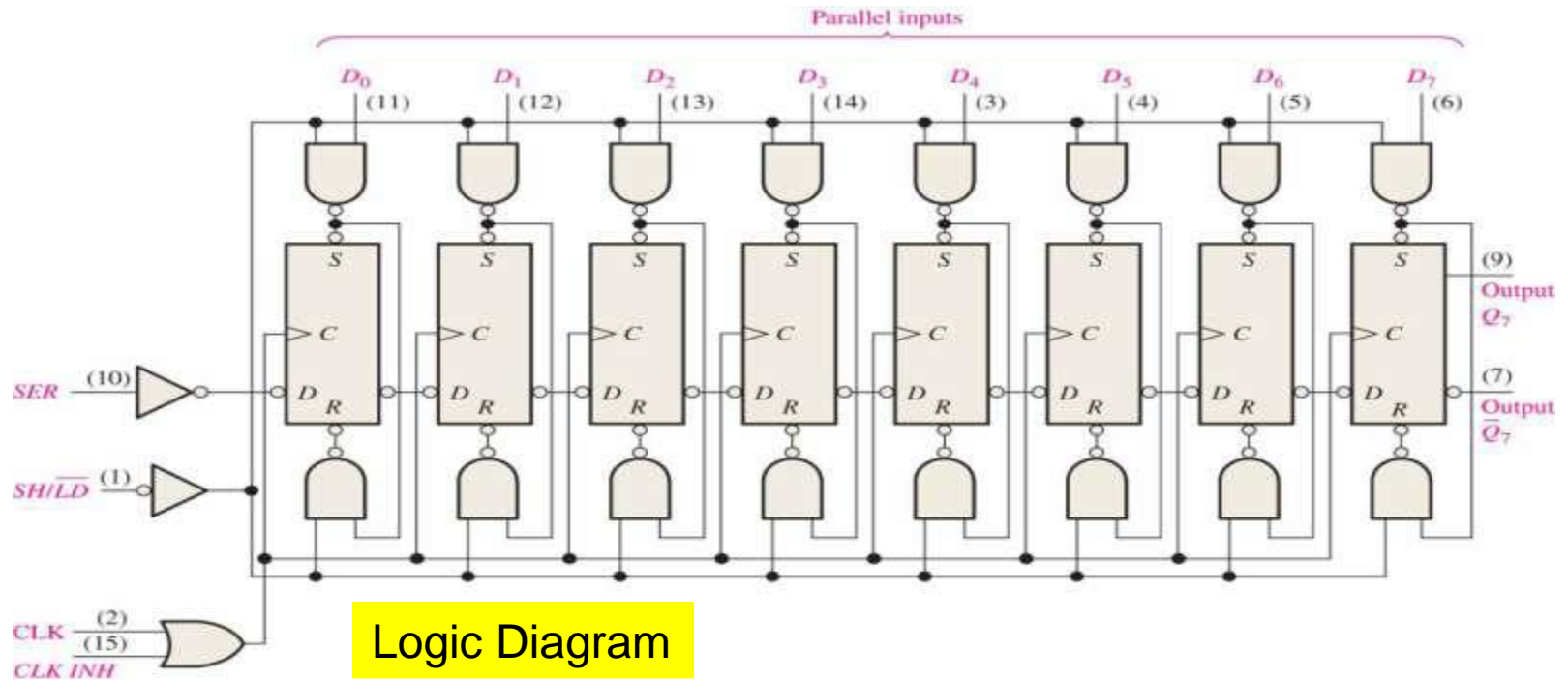
Show the data-output waveform for a 4-bit register with the parallel input data and the clock and *SHIFT/LOAD* waveforms



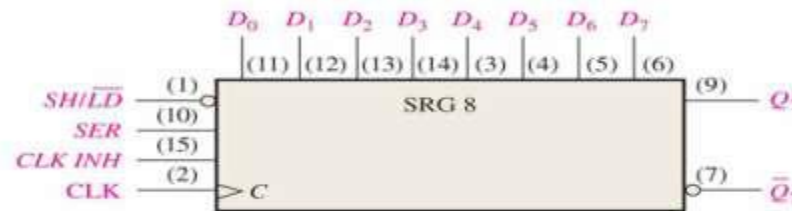
**Solution** On clock pulse 1, the parallel data ( $D_0D_1D_2D_3 = 1010$ ) are loaded into the register, making  $Q_3$  a 0. On clock pulse 2 the 1 from  $Q_2$  is shifted onto  $Q_3$ ; on clock pulse 3 the 0 is shifted onto  $Q_3$ ; on clock pulse 4 the last data bit (1) is shifted onto  $Q_3$ ; and on clock pulse 5, all data bits have been shifted out, and only 1s remain in the register (assuming the  $D$  input remains a 1).

## 8-bit version

### The 74HC165 8-bit parallel load shift register.



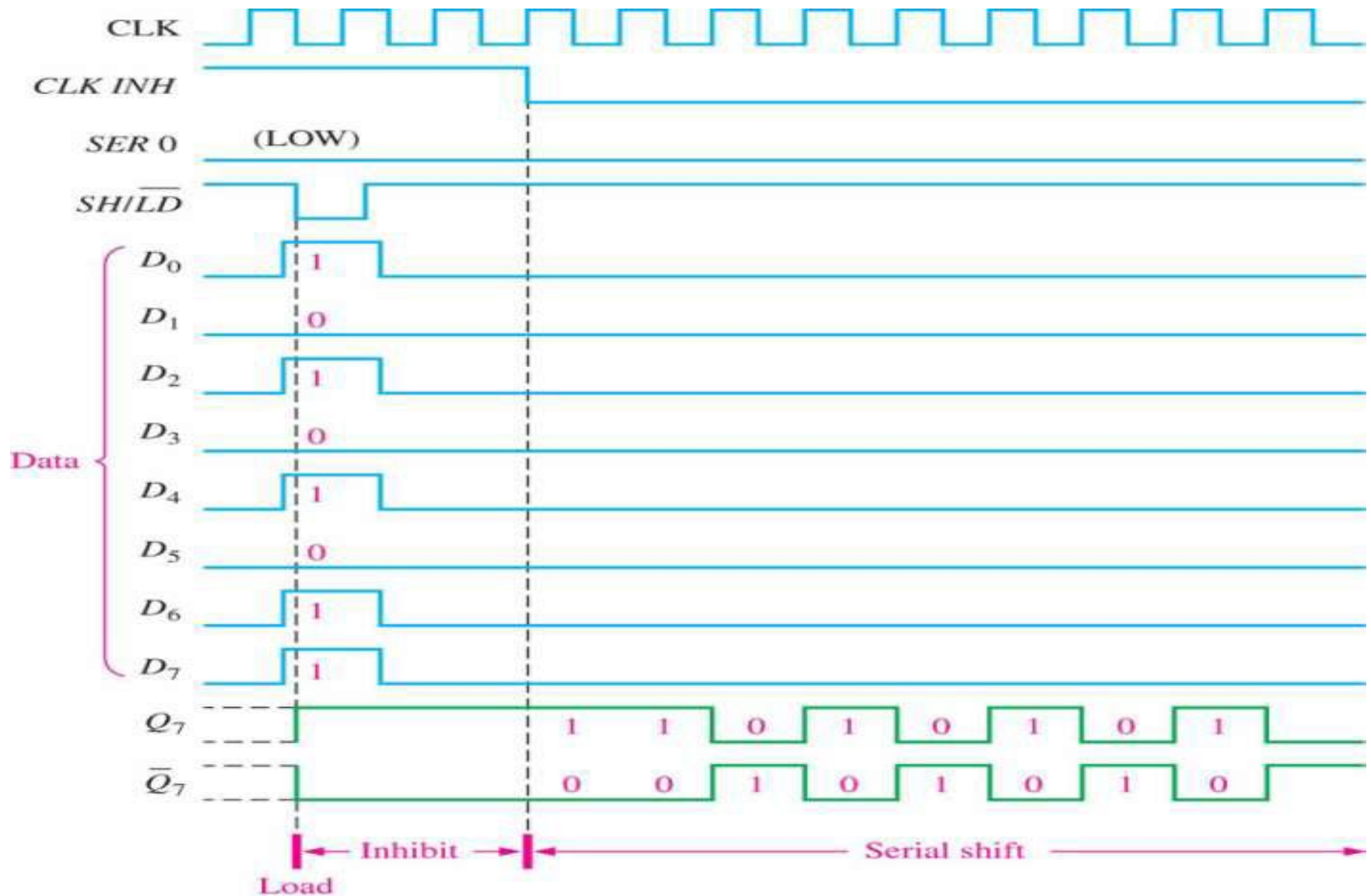
(a) Logic diagram



(b) Logic symbol

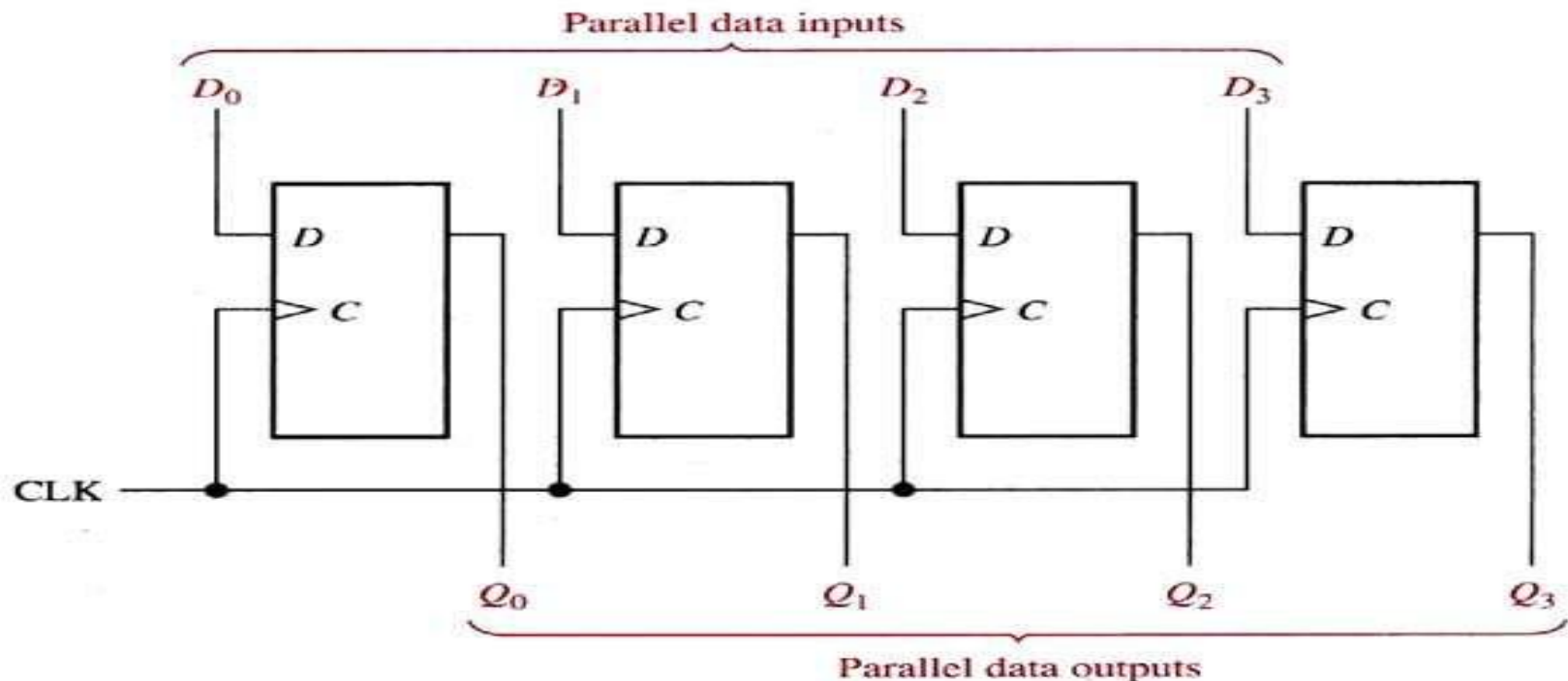
Logic symbol

# Sample Timing Diagram



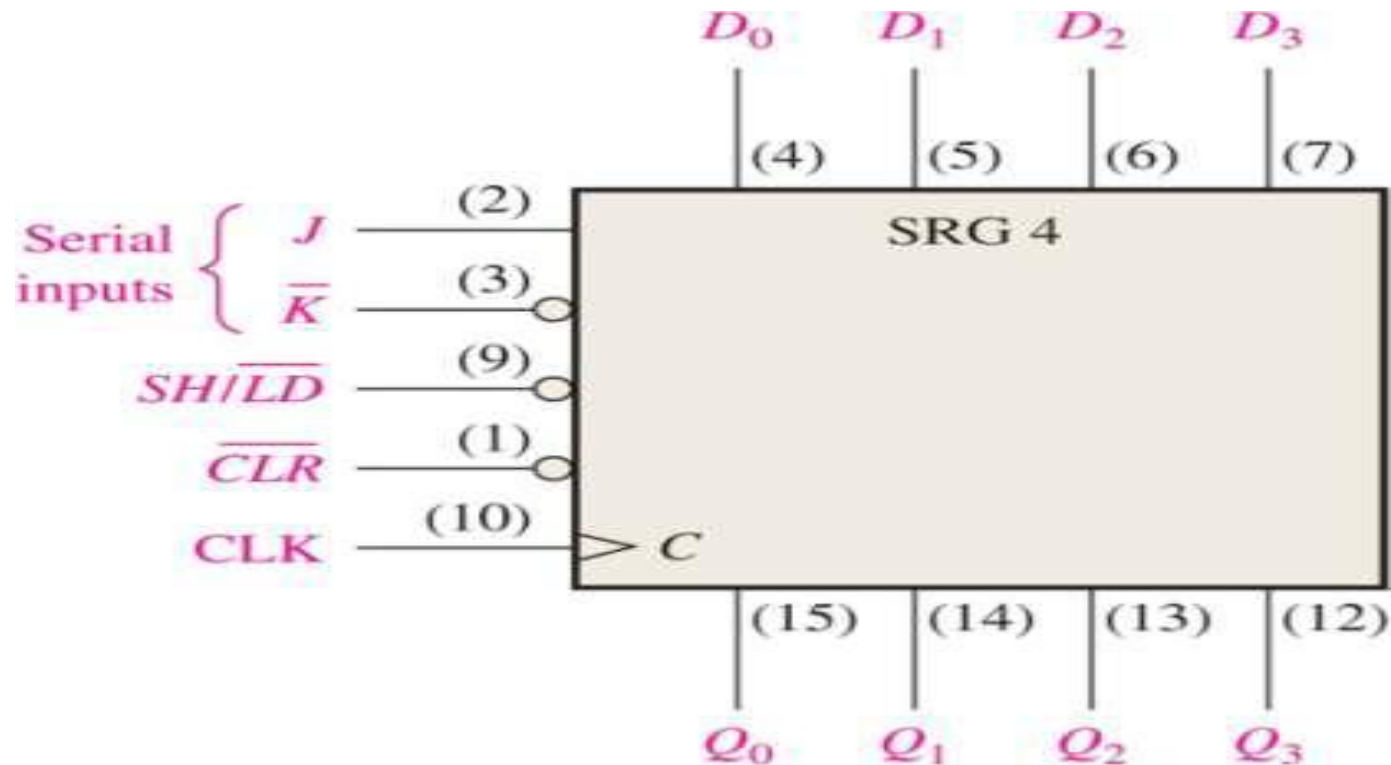
## 5) Parallel In/ Parallel Out Shift Registers

The Parallel in/ Parallel out register employs both Parallel entry and Parallel output of data. Immediately following the simultaneous entry of all data bits, the bits appear on the parallel outputs.



## Logic symbol

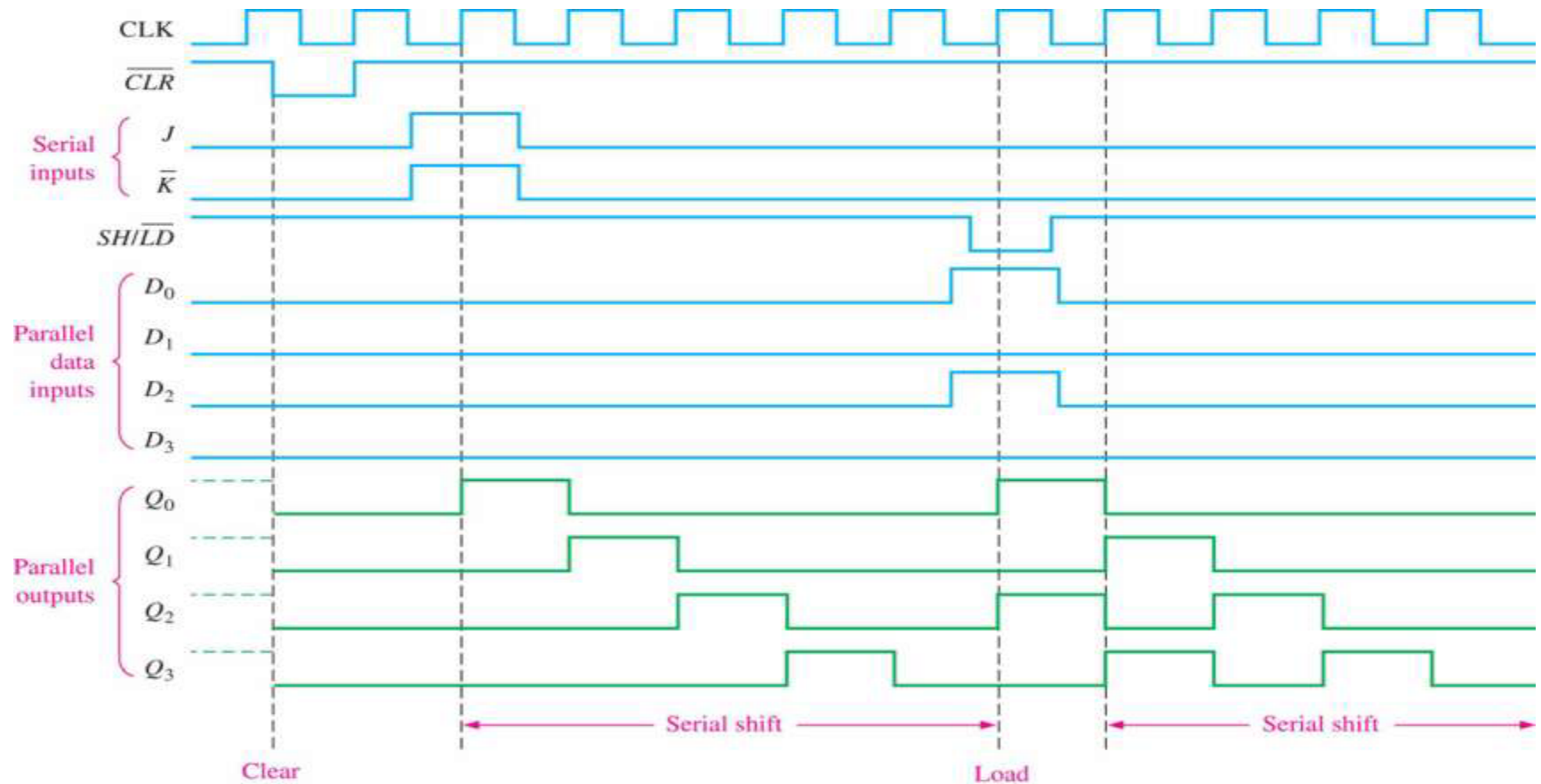
The 74HC195 4-bit parallel-access shift register.  
4-bit parallel in /parallel out SR





# Timing Diagram

Sample timing diagram for a 74HC165 shift register.  
4-bit parallel/parallel out SR  
synchronous CLR, asynchronous LD



**(3 mins)**

## Assignment

<https://docs.google.com/forms/d/e/1FAIpQLSerJg6WP7p80hEXvDqfPa7gRFz37w7BF0gGGEewuioeHy12g/viewform?usp=header>

**Deadline: 25/04/2025**



**Write a report on the shift register of no less than two pages and no more than four?**

Then Submit your answer using the QR code or the link above.

# Summary shift register

Aspect	Description
Definition	A sequential circuit that stores and shifts binary data using flip-flops.
Main Functions	- Shifting data left or right- Data storage- Data transfer
Types	<b>SISO</b> : Serial In Serial Out <b>SIPO</b> : Serial In Parallel Out <b>PISO</b> : Parallel In Serial Out <b>PIPO</b> : Parallel In Parallel Out
Direction	Left shift, right shift, bidirectional
Applications	- Temporary data storage - Serial to parallel conversion - Signal delay - Digital communication

شكراً لحسن

إصغائكم

**THANK YOU** 

UOMU022021      Digital Systems  
Department of Computer Engineering

**END LESSON 8: SHIFT-REGISTERS  
(SISO, SIPO, PISO, PIPO)**