江西理工大学
Jiangxi University of Science and Technology
信息工程学院
School of information engineering

Dr Ata Jahangir Moshayedi

# Digital System Design

Clip Lecture series

Spring_2020

**Prof Associate , School of information engineering Jiangxi university of science and technology, China**
**EMAIL: ajm@jxust.edu.cn**


Jiangxi University of Science and Technology

# Sequential circuits

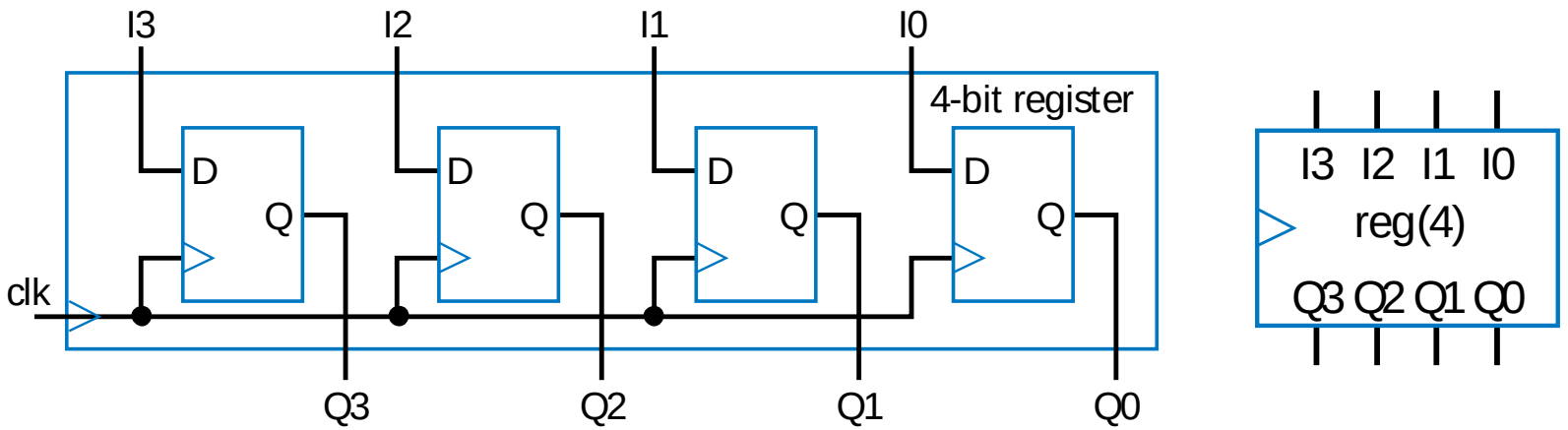
## Registers /Shift registers

# What is register & what is shift registers


- Flip-flop is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a **Register**. The **n-bit register** will consist of **n** number of flip-flop and it is capable of storing an **n-bit** word.
- The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as **shift registers**. There are four mode of operations of a shift register.
- Serial Input Serial Output
- Serial Input Parallel Output
- Parallel Input Serial Output
- Parallel Input Parallel Output

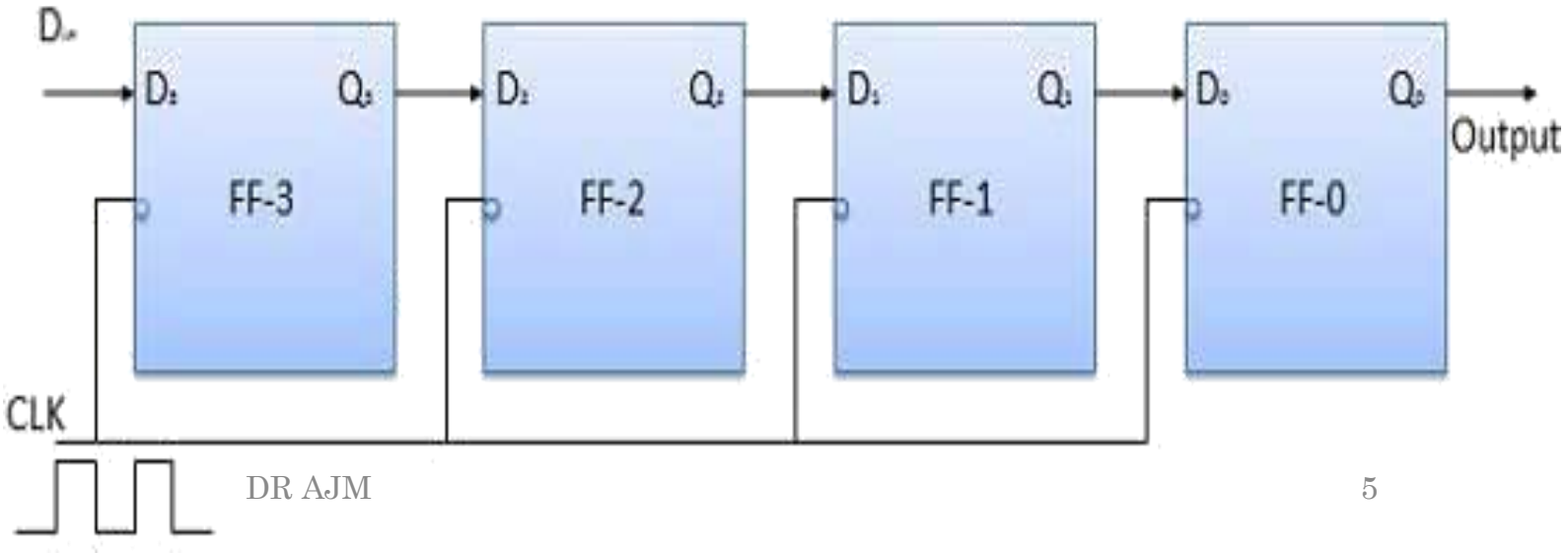
# Basic Register

- Typically, we store multi-bit items
  - e.g., storing a 4-bit binary number
- ***Register***: multiple flip-flops sharing clock signal
  - From this point, we'll use registers for bit storage
    - No need to think of latches or flip-flops
    - But now you know what's inside a register



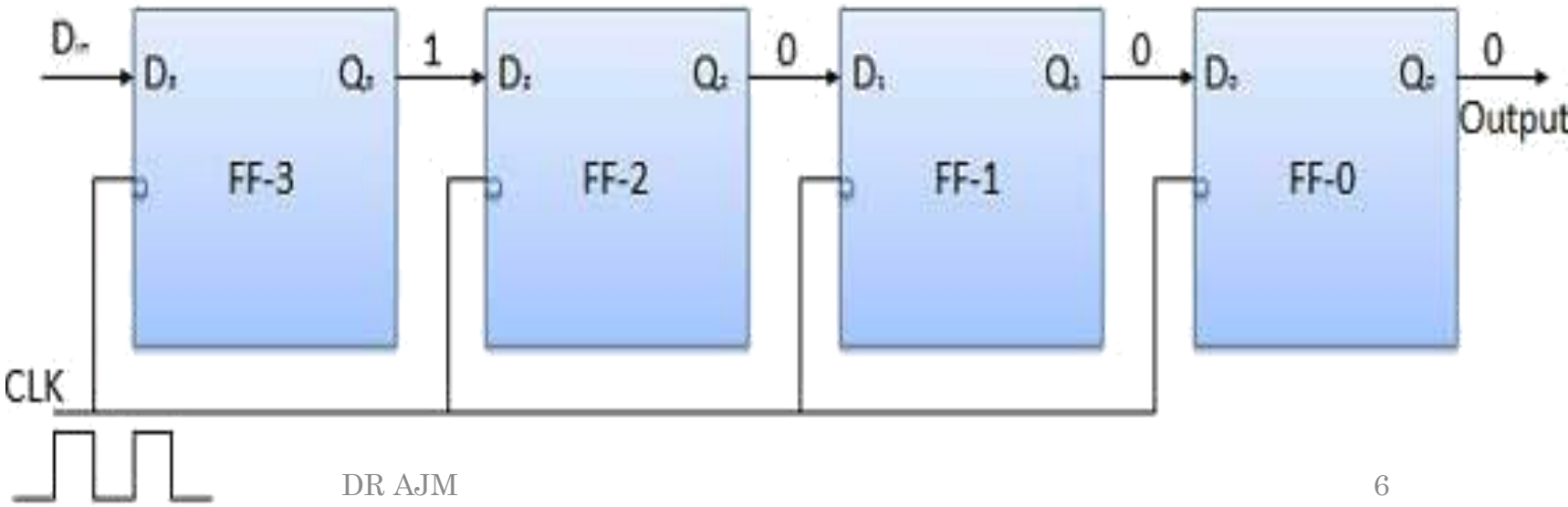
# Serial Input Serial Output


- Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$.
- If an entry of a four bit binary number 1 1 1 1 is made into the register, this number should be applied to $\mathbf{D_{in}}$ bit with the LSB bit applied first.
- The D input of FF-3 i.e. $D_3$ is connected to serial data input $\mathbf{D_{in}}$. Output of FF-3 i.e. $Q_3$ is connected to the input of the next flip-flop i.e. $D_2$ and so on.

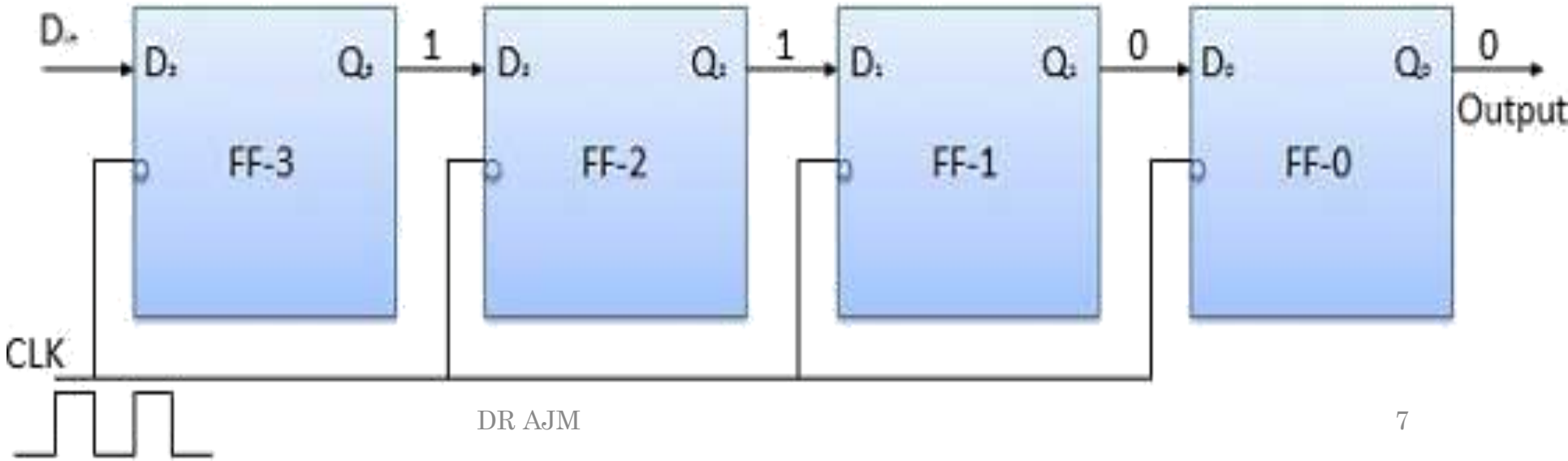


# Operation

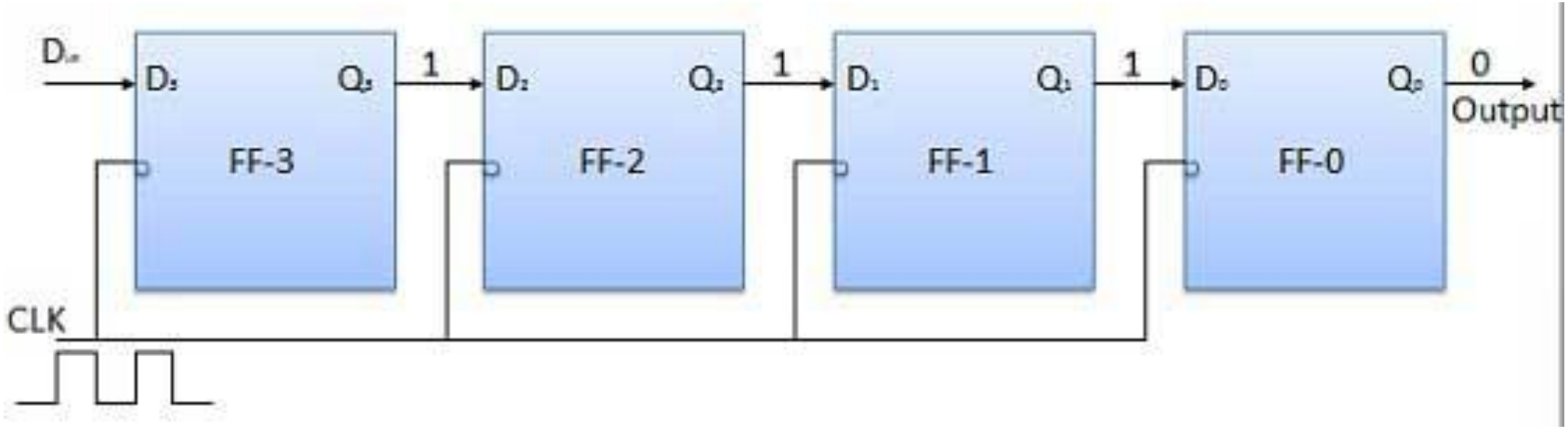
- Before application of clock signal, let $Q_3\ Q_2\ Q_1\ Q_0 = 0000$ and apply LSB bit of the number to be entered to $D_{in}$. So $D_{in} = D_3 = 1$.
- Apply the clock. On the first falling edge of clock, the FF-3 is set, and stored word in the register is $Q_3Q_2\ Q_1\ Q_0 = 1000$.



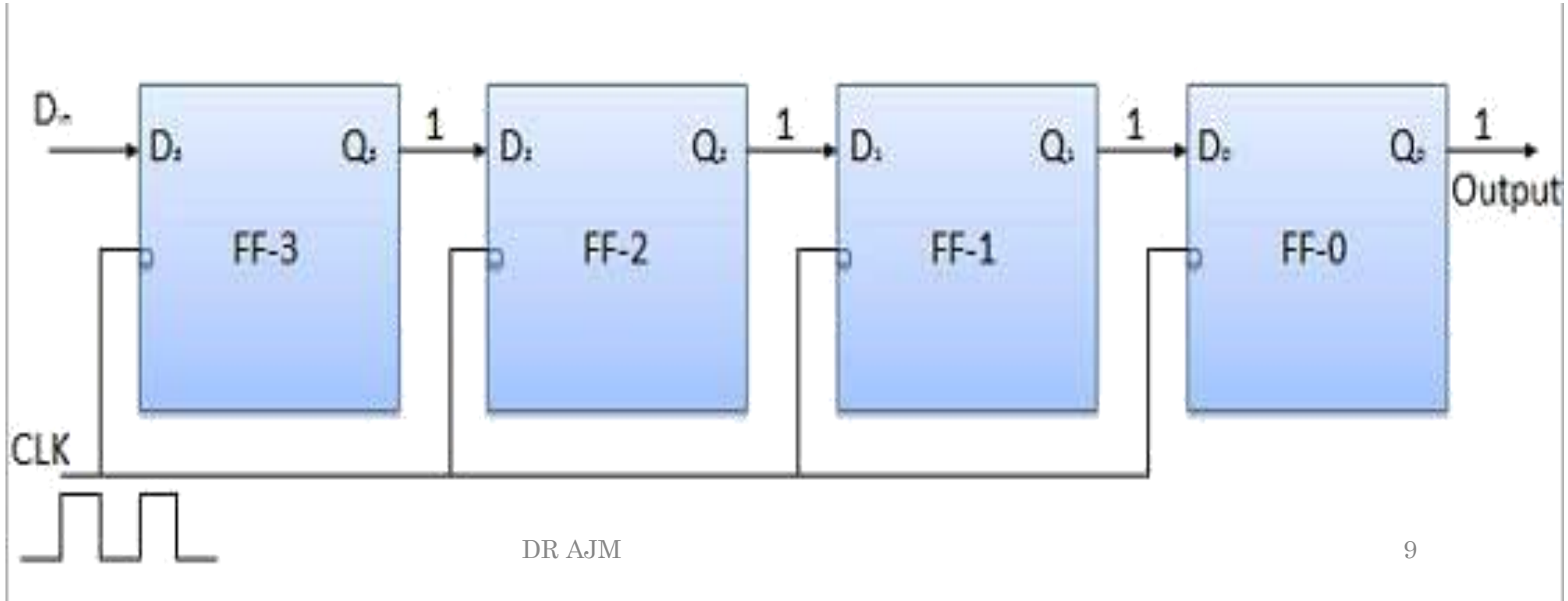
- Apply the next bit to $D_{in}$. So $D_{in} = 1$.
- As soon as the next negative edge of the clock hits, FF-2 will set and the stored word change to $Q_3\ Q_2\ Q_1\ Q_0 = 1100$.



- Apply the next bit to be stored i.e. 1 to $D_{in}$. Apply the clock pulse. As soon as the third negative clock edge hits, FF-1 will be set and output will be modified to $Q_3\ Q_2\ Q_1\ Q_0 = 1110$.



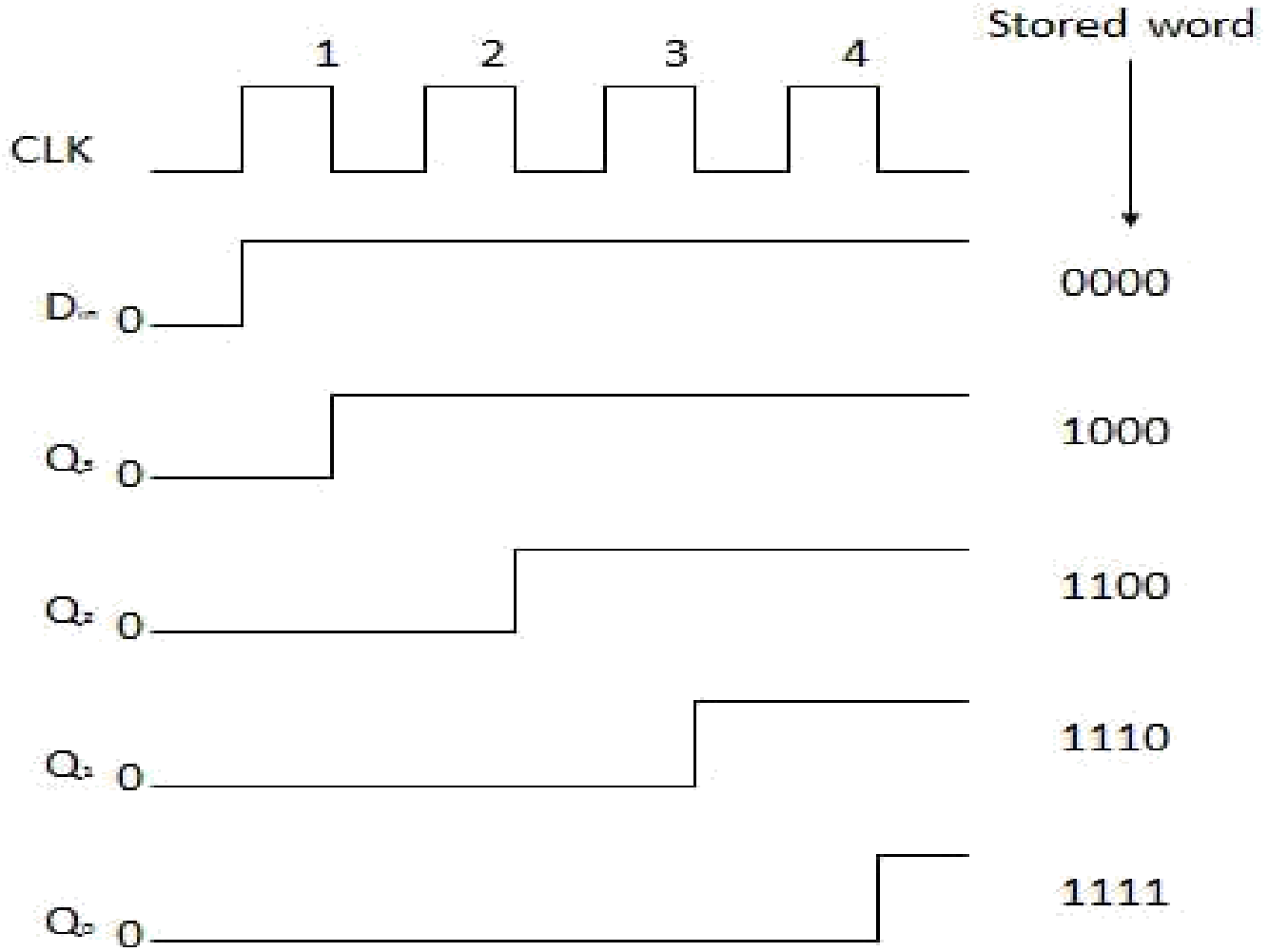
- Similarly with $D_{in} = 1$ and with the fourth negative clock edge arriving, the stored word in the register is $Q_3\ Q_2\ Q_1\ Q_0 = 1111$.



# Truth Table/Waveforms

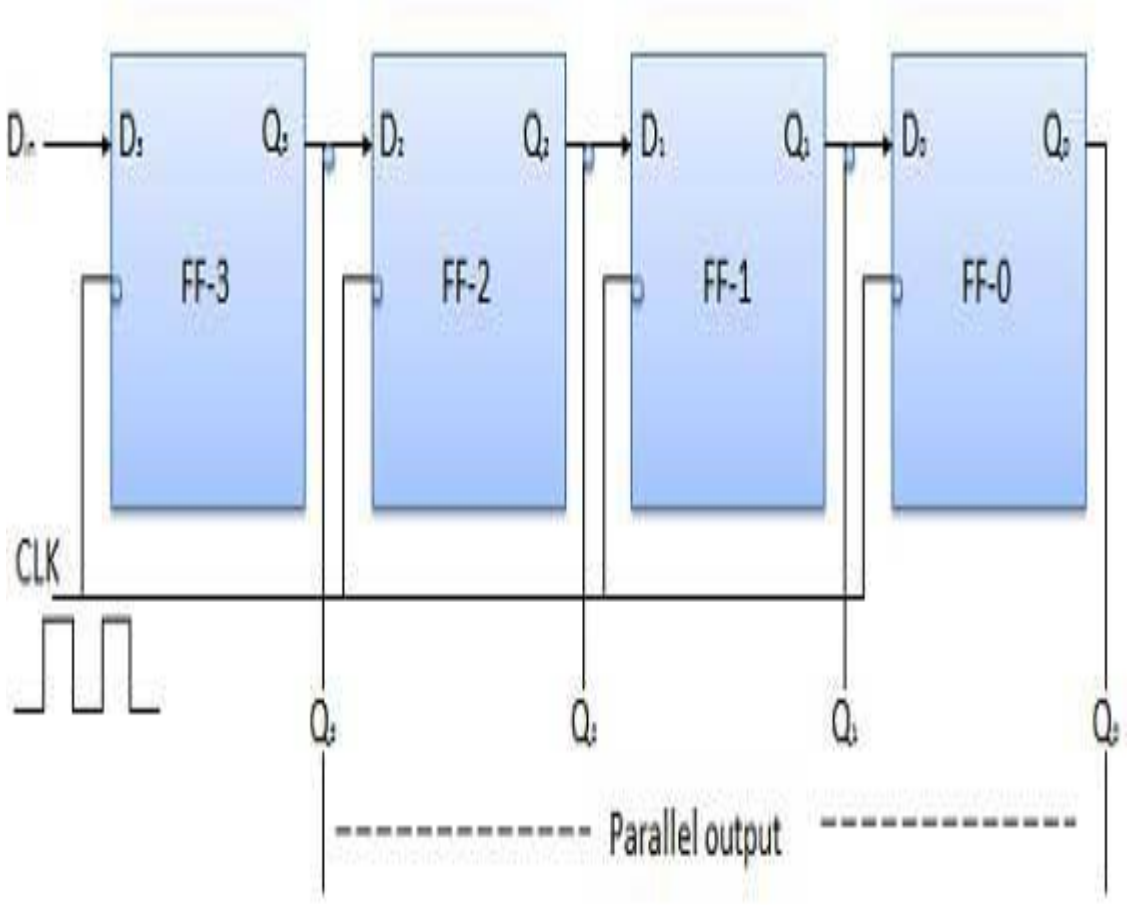
| | CLK | $D_n = Q_3$ | $Q_3 = D_2$ | $Q_2 = D_1$ | $Q_1 = D_0$ | $Q_0$ |
|---|---|---|---|---|---|---|
| Initially | | | 0 | 0 | 0 | 0 |
| (i) | ↓ | 1 | 1 | 0 | 0 | 0 |
| (ii) | ↓ | 1 | 1 | 1 | 0 | 0 |
| (iii) | ↓ | 1 | 1 | 1 | 1 | 0 |
| (iv) | ↓ | 1 | 1 | 1 | 1 | 1 |

→ Direction of data travel



# Serial Input Parallel Output

- In such types of operations, the data is entered serially and taken out in parallel fashion. Data is loaded bit by bit.
- The outputs are disabled as long as the data is loading.
- As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.
- 4 clock cycles are required to load a four bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.



# Parallel Input Serial Output (PISO)


- Data bits are entered in parallel fashion.
- The circuit shown below is a four bit parallel input serial output register.
- Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
- The binary input word $B_0$, $B_1$, $B_2$, $B_3$ is applied though the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.

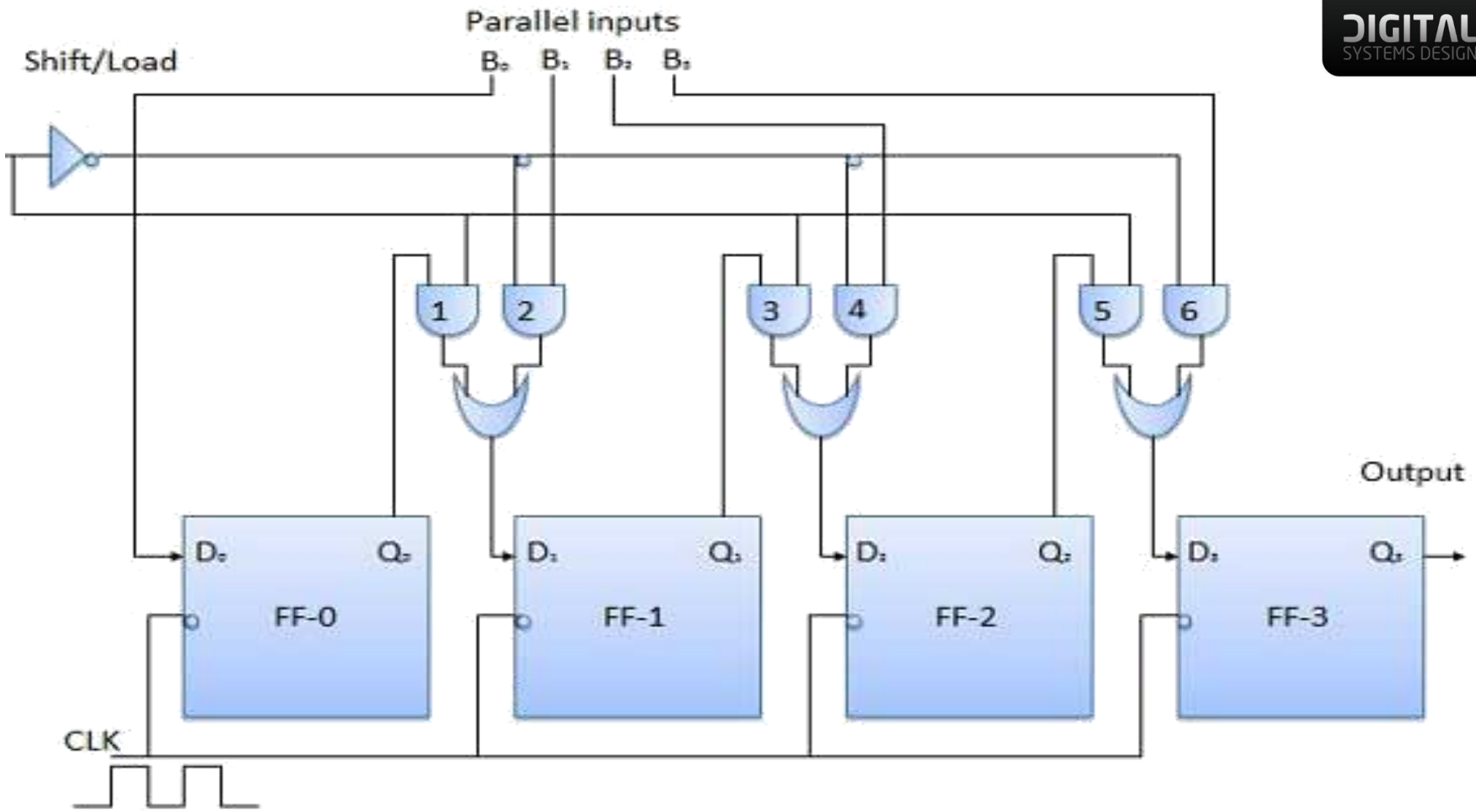
**Load mode**

When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass $B_1$, $B_2$, $B_3$ bits to the corresponding flip-flops. On the low going edge of clock, the binary input $B_0$, $B_1$, $B_2$, $B_3$ will get loaded into the corresponding flip-flops. Thus parallel loading takes place

**Shift mode**

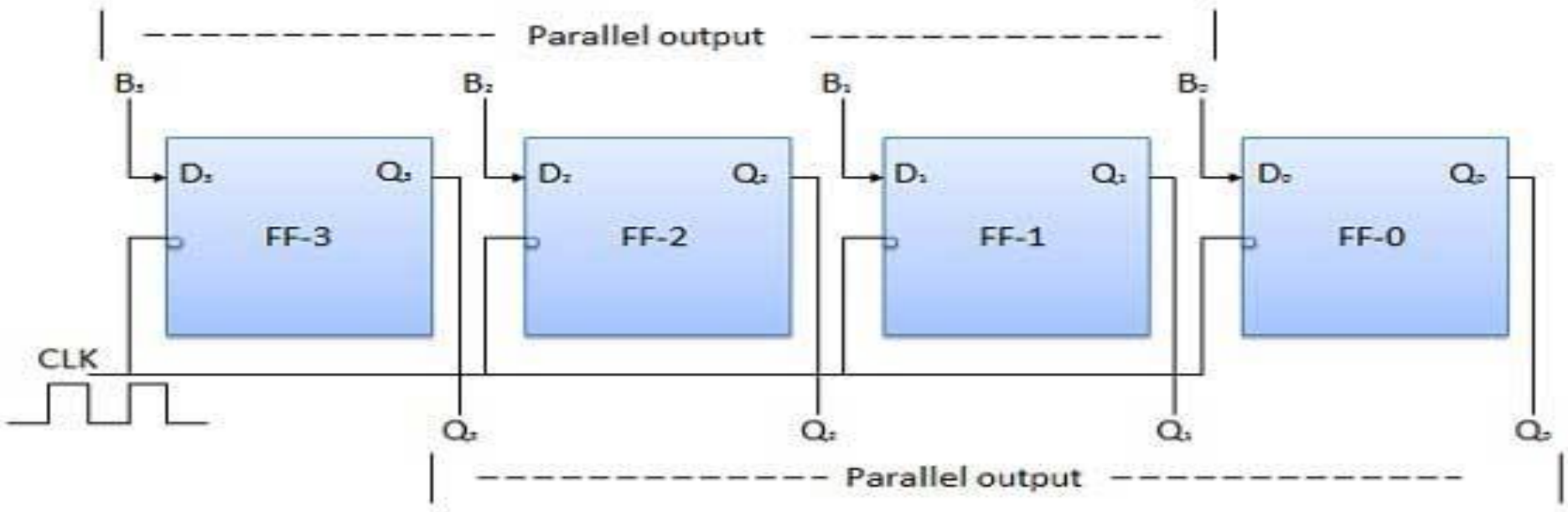
When the shift/load bar line is low (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

# Block Diagram



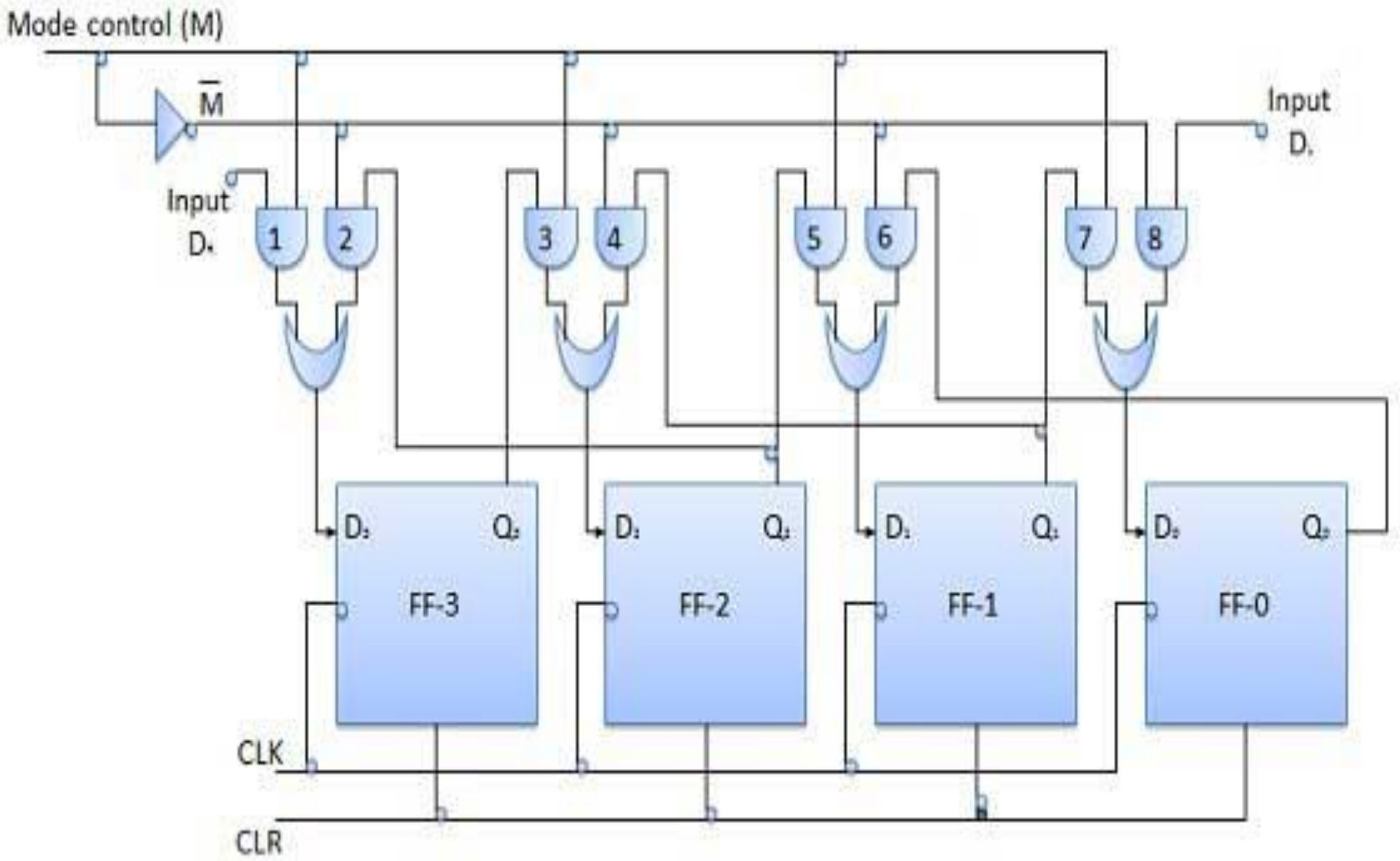
# Parallel Input Parallel Output (PIPO)

- In this mode, the 4 bit binary input $B_0$, $B_1$, $B_2$, $B_3$ is applied to the data inputs $D_0$, $D_1$, $D_2$, $D_3$ respectively of the four flip-flops. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.



# Bidirectional Shift Register


- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.
- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.
- Such a register is called bi-directional register. A four bit bi-directional shift register is shown in fig.
- There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input (M).

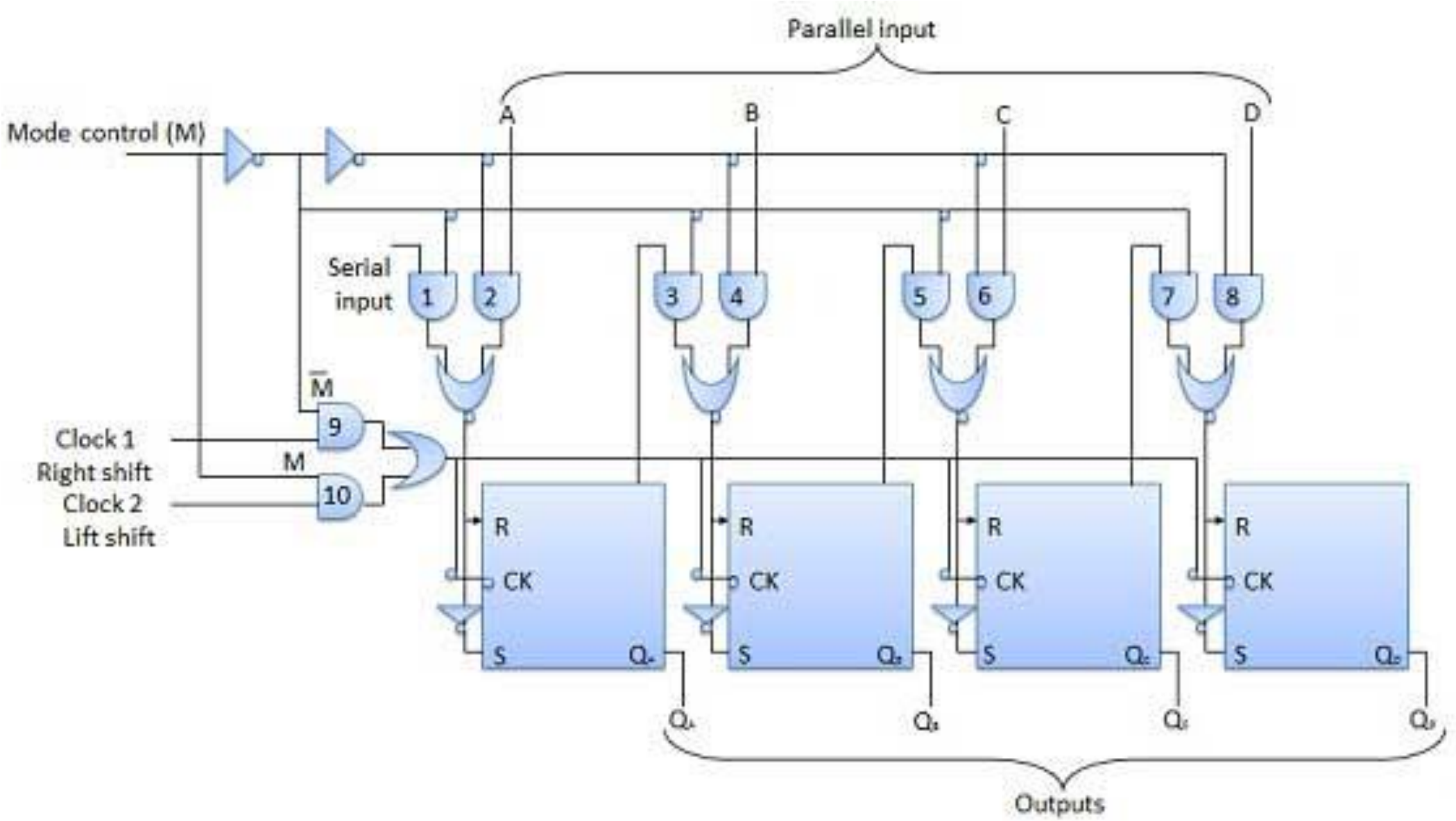

Mode control (M)
M
Input
D.
Input
D.
1
2
3
4
5
6
7
8
D₃
Q₃
FF-3
D₂
Q₂
FF-2
D₁
Q₁
FF-1
D₀
Q₀
FF-0
CLK
CLR

# Operation

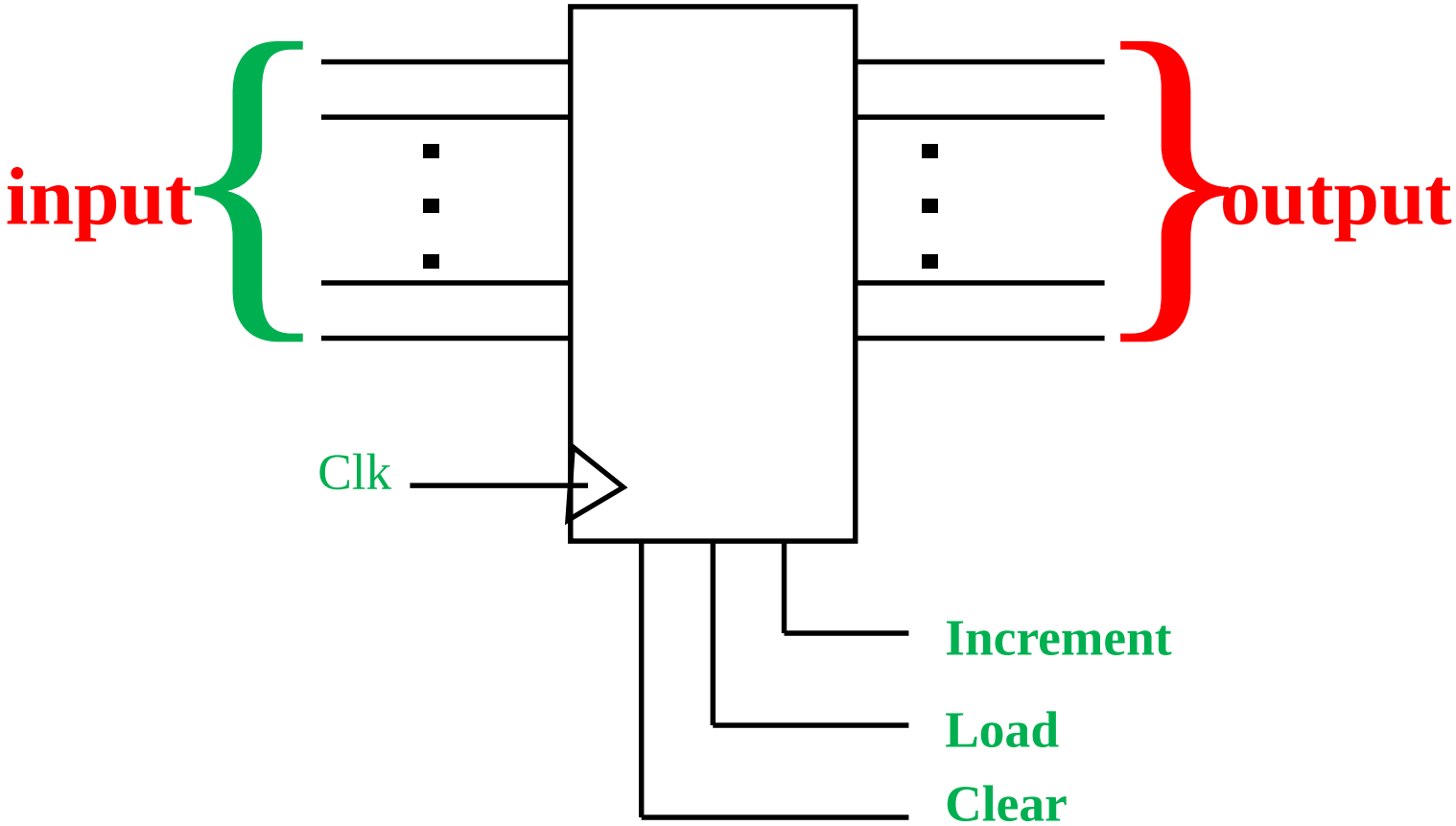
| S.N | Condition | Operation |
|---|---|---|
| 1 | With M = 1 – Shift right operation | If M = 1, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled.<br>The data at $D_R$ is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with M = 1 we get the serial right shift operation. |
| 2 | With M = 0 – Shift left operation | When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5 and 7 are disabled.<br>The data at $D_L$ is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with M = 0 we get the serial right shift operation. |

# Universal Shift Register

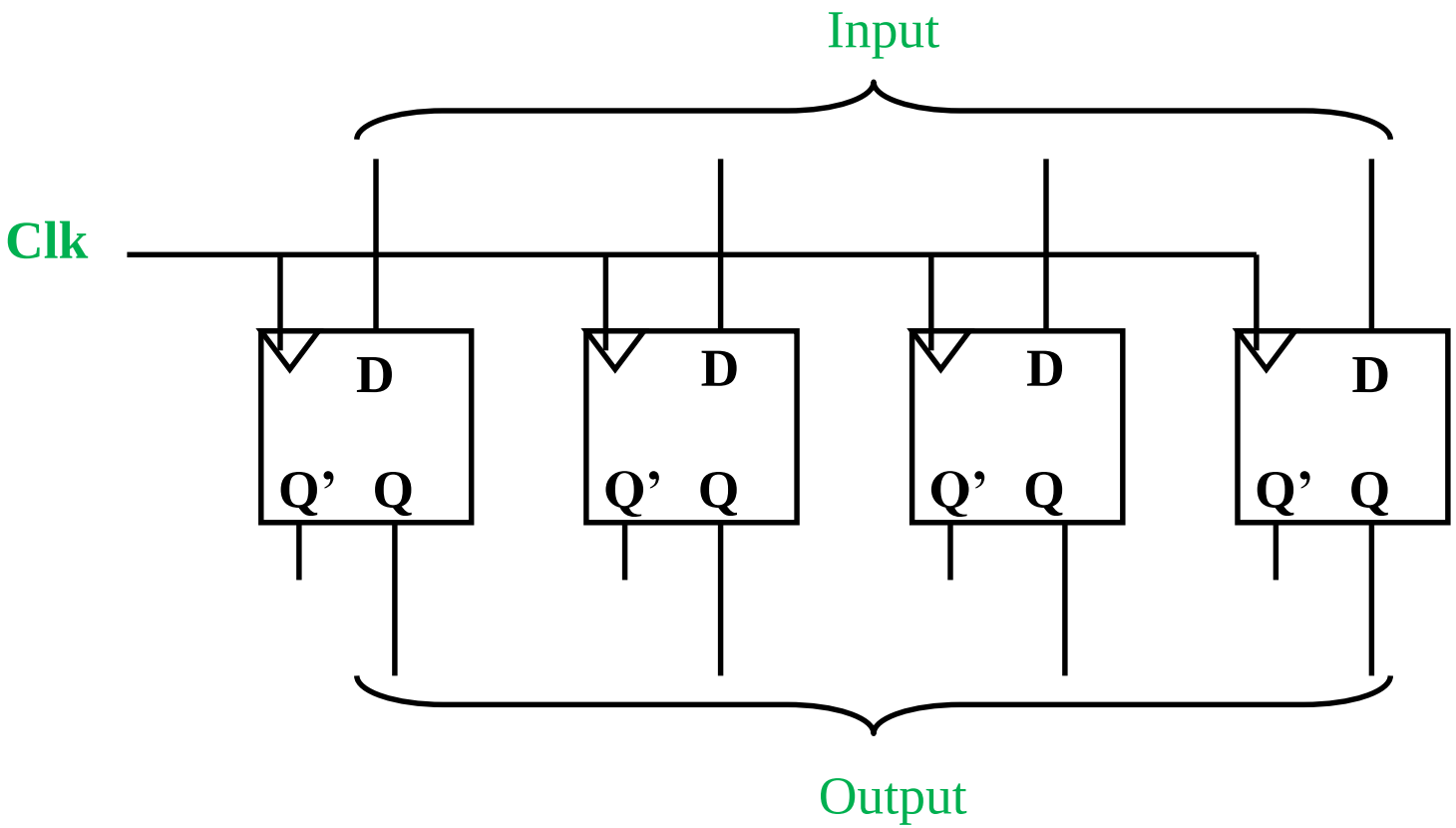

- A shift register which can shift the data in only one direction is called a uni-directional shift register. A shift register which can shift the data in both directions is called a bi-directional shift register. Applying the same logic, a shift register which can shift the data in both directions as well as load it parallely, is known as a universal shift register. The shift register is capable of performing the following operation −
  - Parallel loading
  - Left Shifting
  - Right shifting
- The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting. With mode control pin connected to ground, the universal shift register acts as a bi-directional register. For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas for the shift right operation, the serial input is applied to D input.


Parallel input
A
B
C
D
Mode control (M)
Serial
input
1
2
3
4
5
6
7
8
M̄
9
Clock 1
Right shift
M
10
Clock 2
Lift shift
R
CK
S
Q
Outputs

# The block Diagram of shift registor

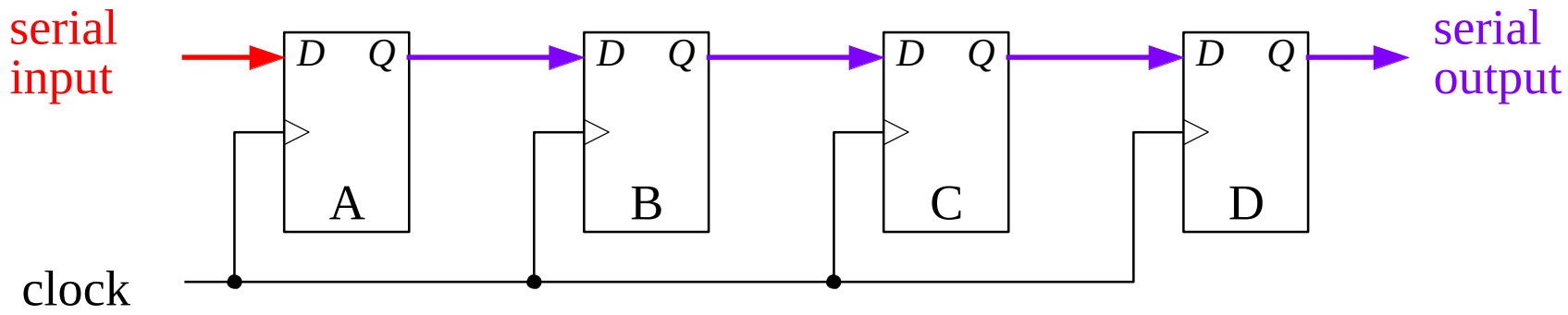


# The simple Shift register with D flip flop



# Basic shift register

A basic shift register is simply a chain of *D* flip-flops with a common clock.

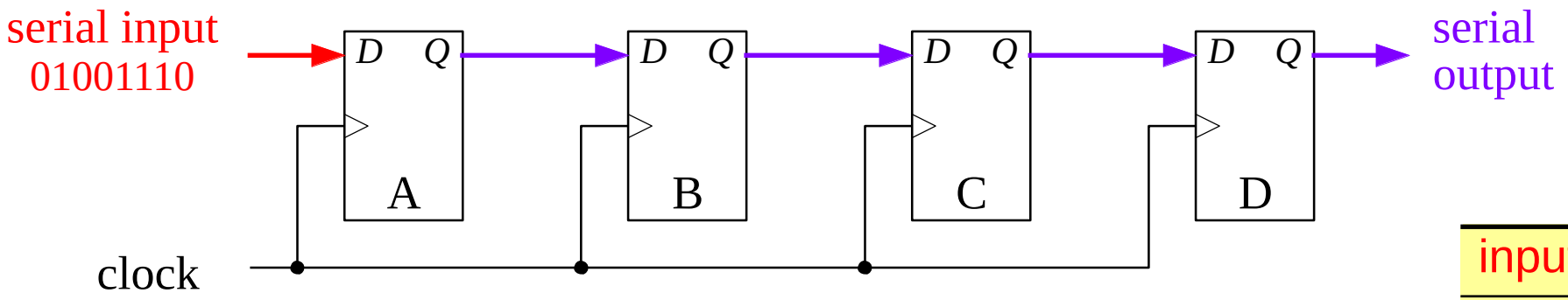



Each flip-flop transfers its *D* input to its *Q* output at a clock transition.

- The effect is to transfer data along the register, one flip-flop per clock cycle.
- This type of register is called a serial input-serial output (SISO).

# Basic shift register

A basic shift register is simply a chain of *D* flip-flops with a common clock.



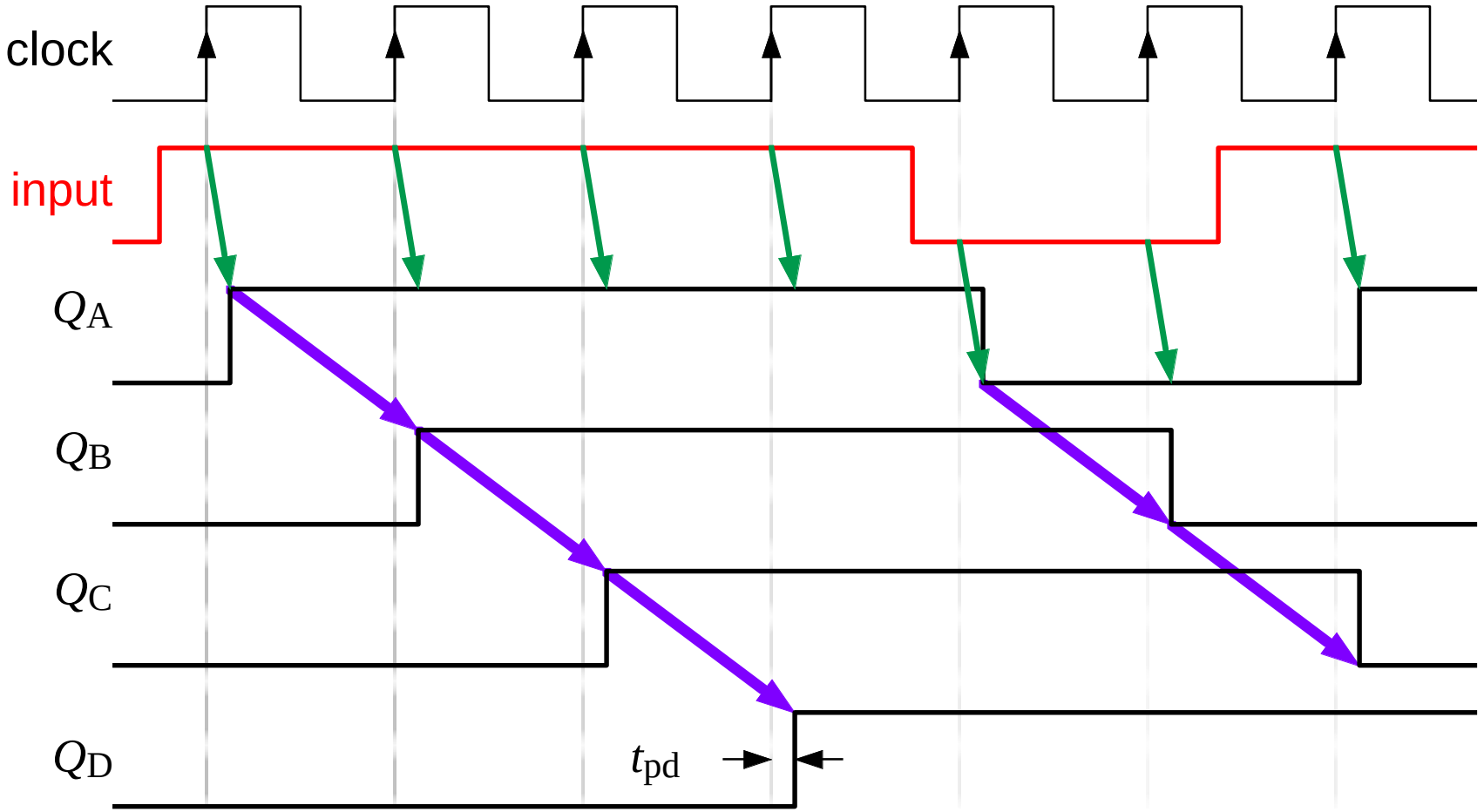

The table shows the contents of the register after successive clock transitions. The assumption is that the register is initially clear.

- The number of clock pulses needed to fill the register is equal to the number of flip-flops used to make the register.
  - This is a 4 bit register.

clock pulses

| input | $Q_A$ | $Q_B$ | $Q_C$ | $Q_D$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 |

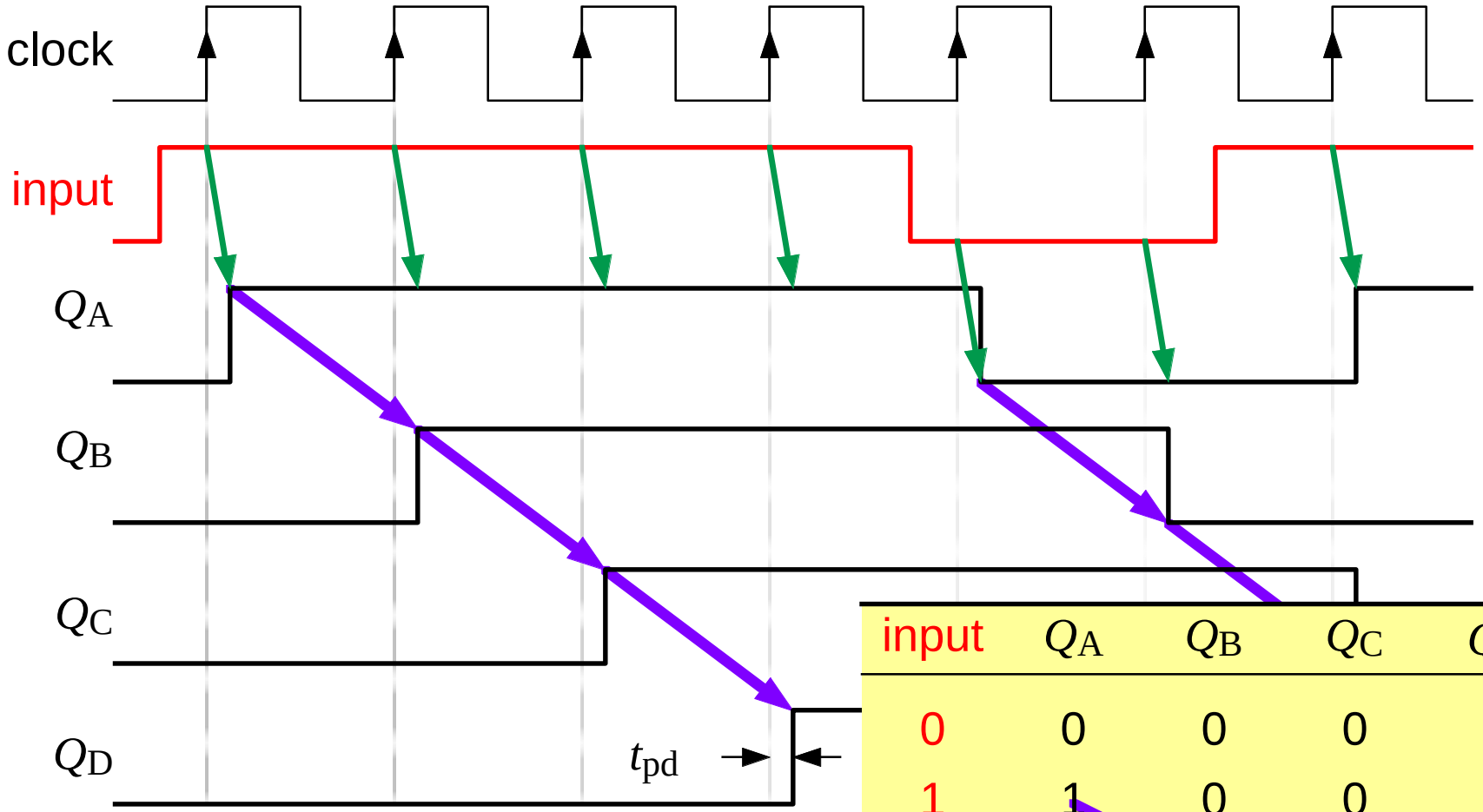
# Timing for a shift register




| input | $Q_A$ | $Q_B$ | $Q_C$ | $Q_D$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |

The pattern in successive flip-flops moves to the right with each clock cycle to shift the pattern into and out of the register.

# Timing for a shift register




| input | $Q_A$ | $Q_B$ | $Q_C$ | $Q_D$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |

# Applications of a basic shift register


1. **Delay line** — *N* stages delay the signal by *N* clock cycles

2. **Multiplication and division by powers of 2**, because this just requires a shift of the binary number (like multiplication or division by 10 in decimal)

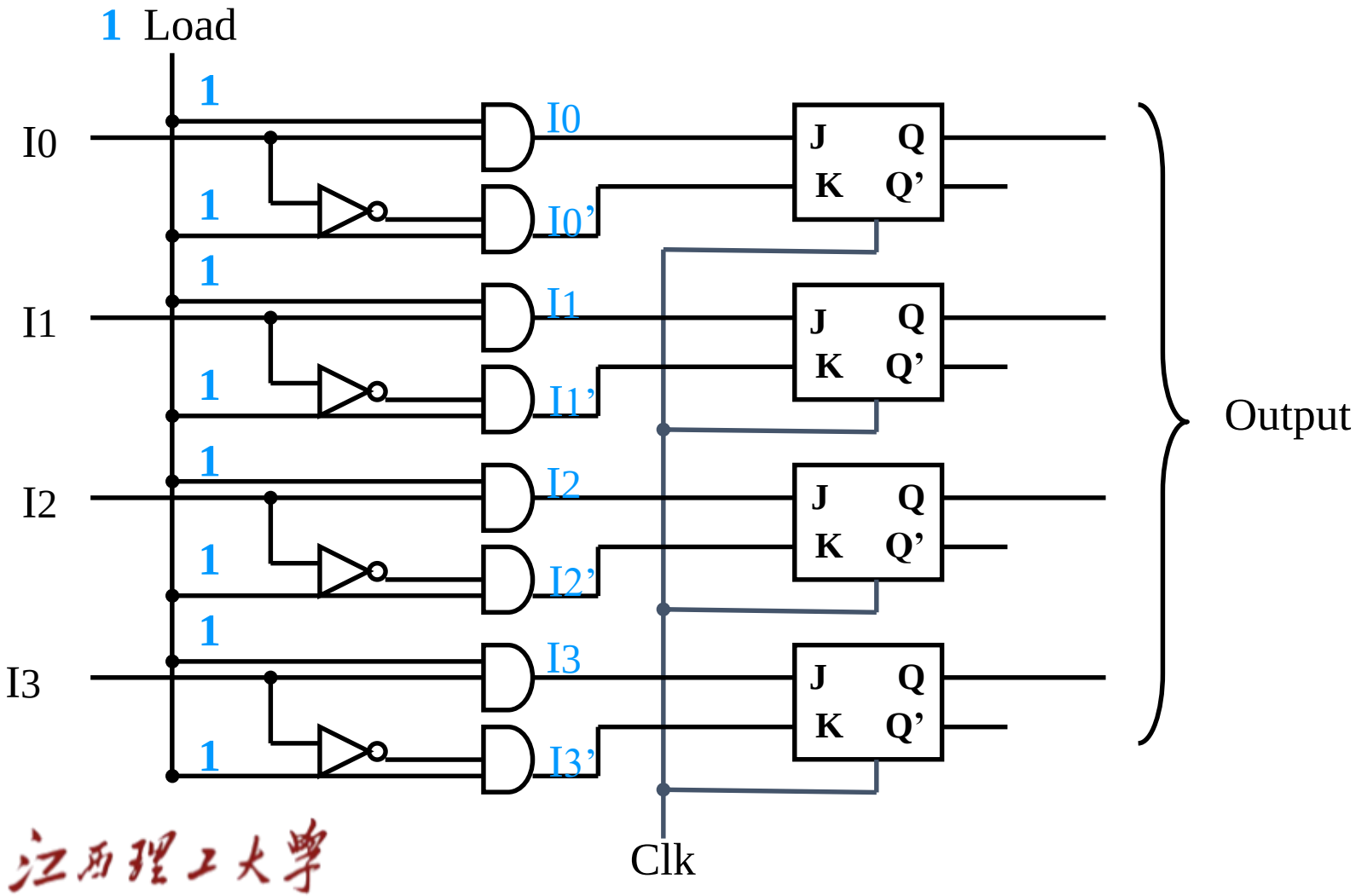
   Example: decimal 3 x 4 = 12 becomes 11 x 100 = 1100 in binary The arithmetic logic unit (ALU) of a computer processor uses a shift register for this purpose.

   **Warning**: the 'sense' of a shift — left or right — is usually based on its effect on binary numbers written in the usual way. For example,
   11 → 1100 is called a **left shift**. This is clearer if both numbers are written with 8-bits as 00000011 → 00001100. Similarly, dividing by 2 such as 00010110 → 00001011 is a **right shift**.

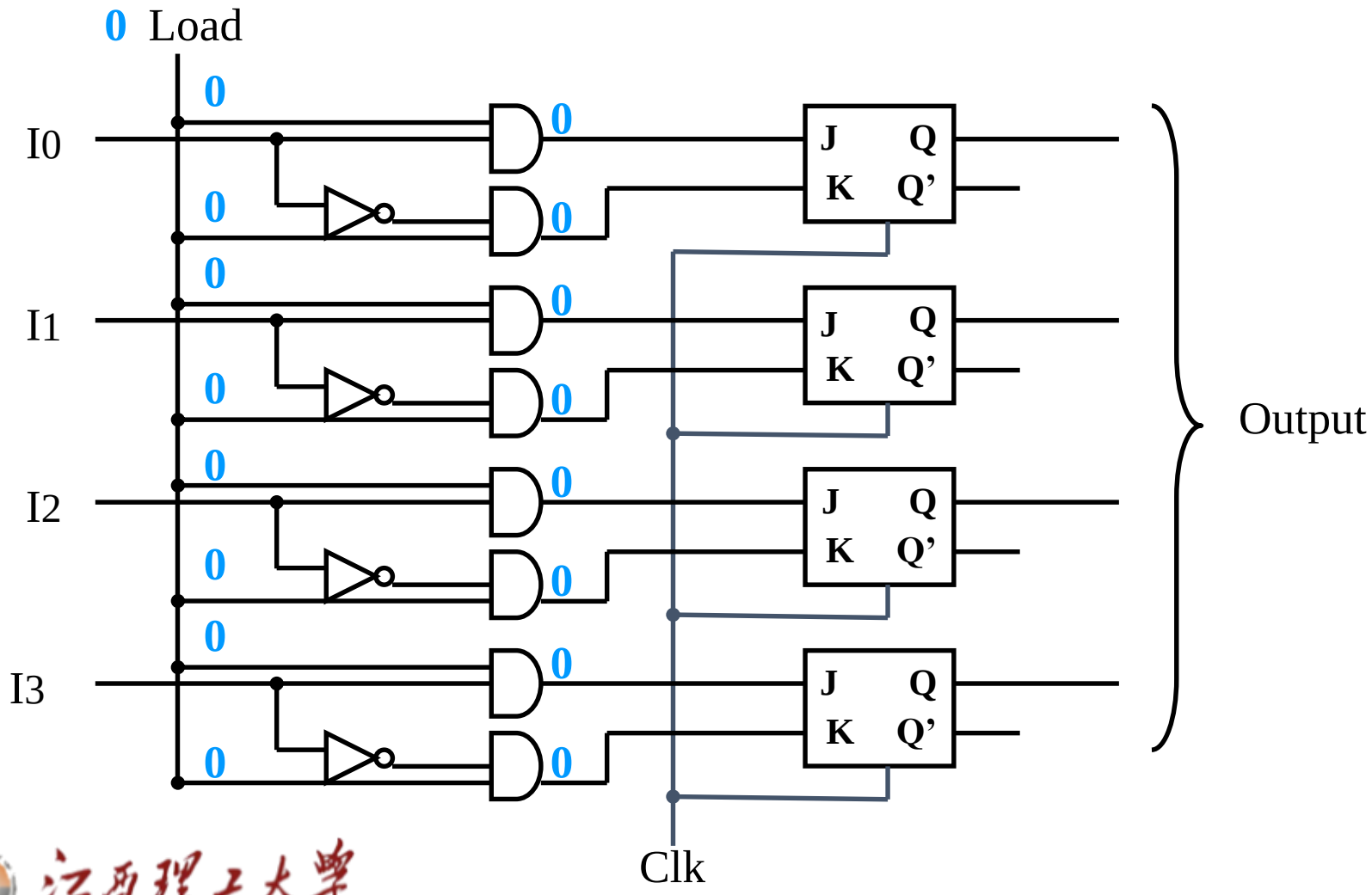
   This is the opposite of what we usually draw in a counter circuit, with the least significant bit (LSB) on the left. **Take care!**

   There is a 'rotate' operation where the output from the shift register is fed back to the beginning, usually through the 'carry bit'.

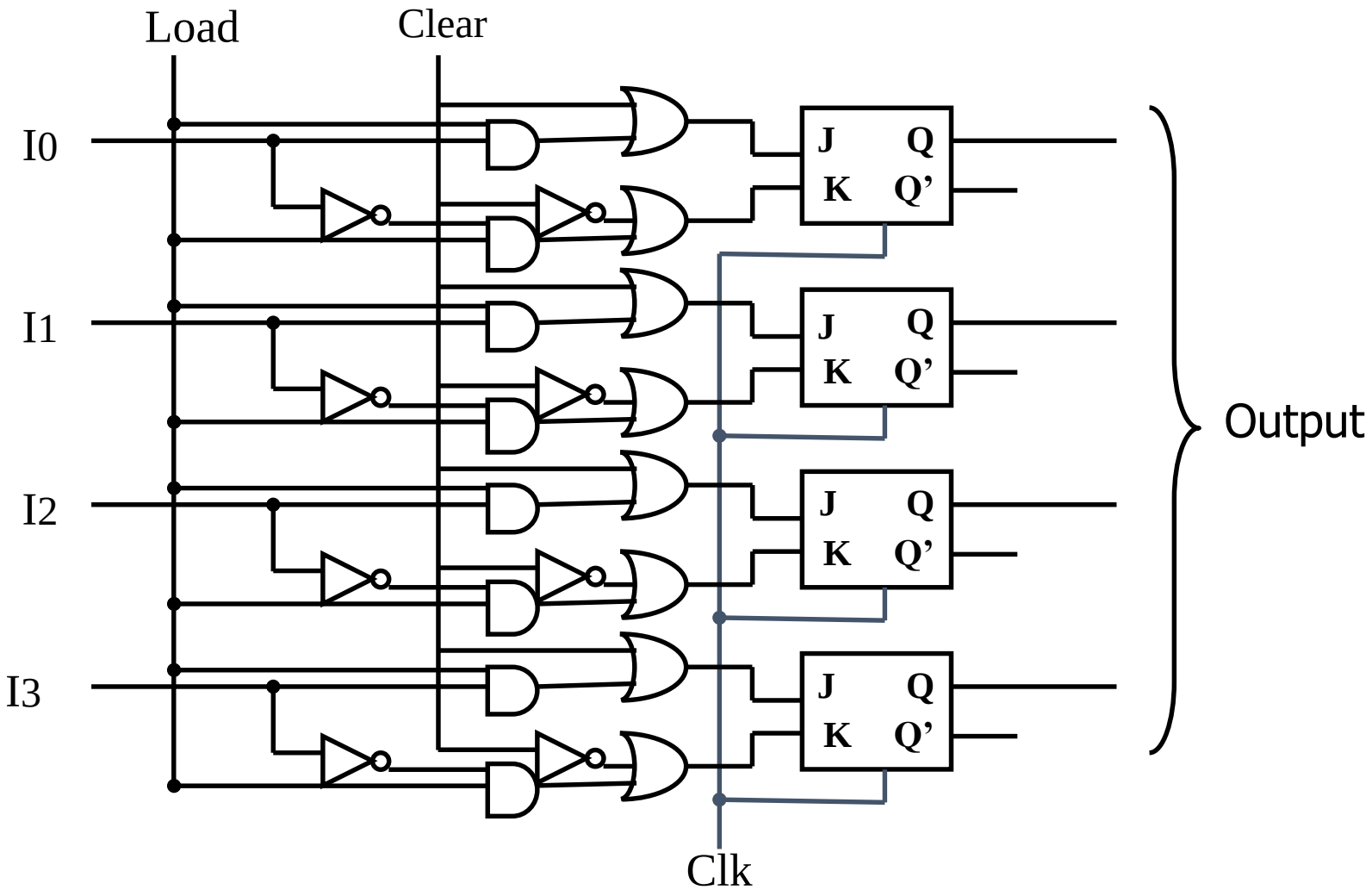
# Design of a shift register with JK flip Flop and load PIN



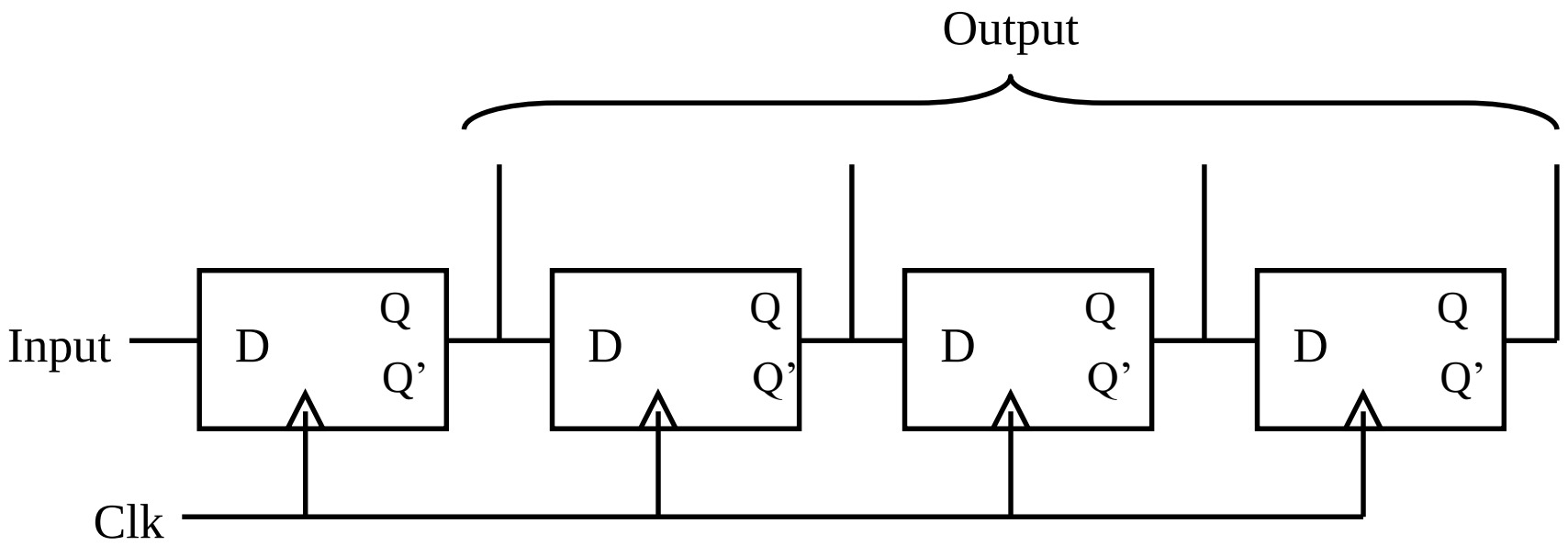
# Design of a shift register with JK flip Flop and load PIN



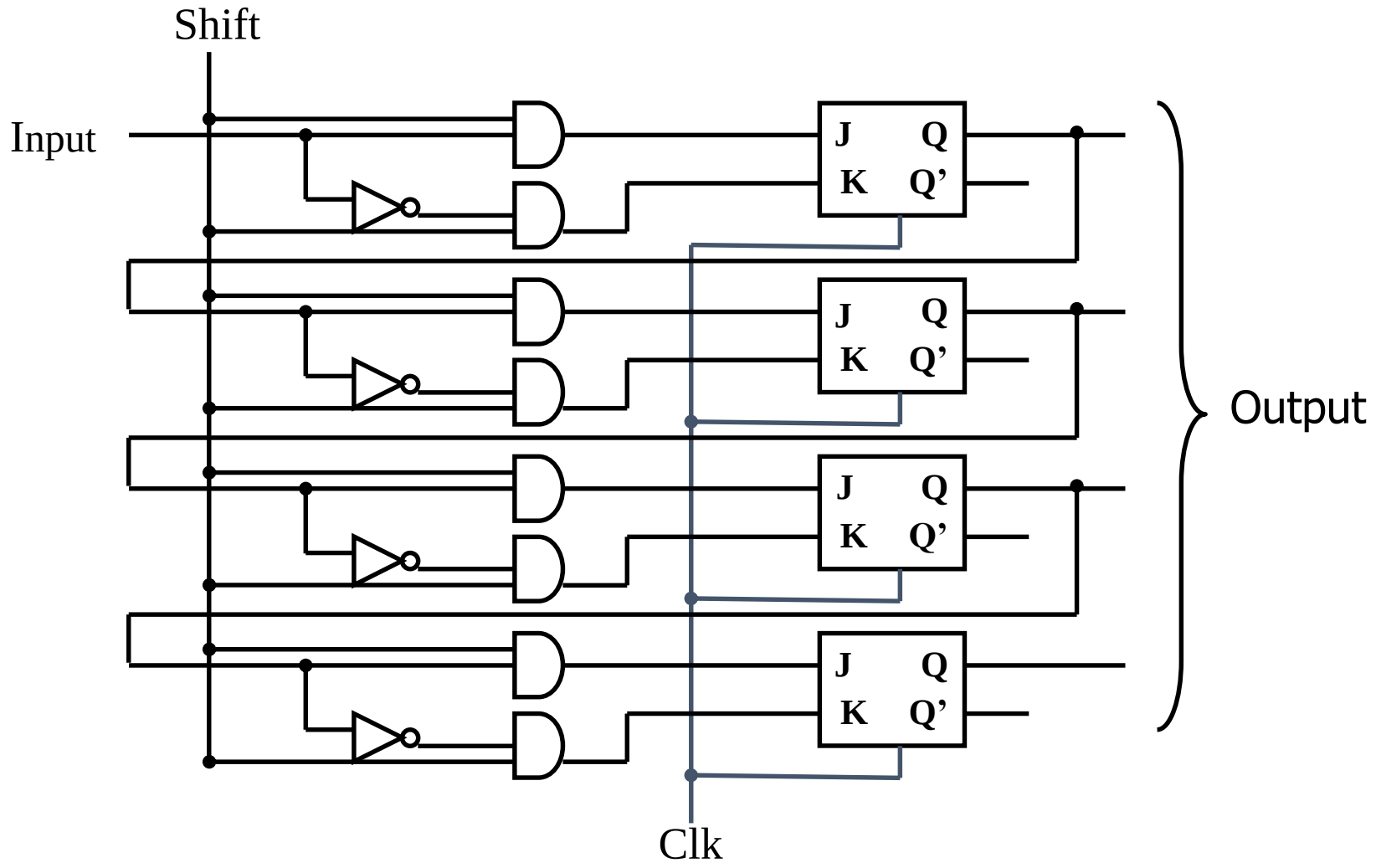
# Design of a shift register with the clear and load PIN



# SHIFT REGISTER WITH D FLIP FLOP



# shift register with JK flip Flop



# Shift registers


**Circuit for simple shift register Basic applications**

Ring counters Johnson counters

Pseudo-random binary sequences and encryption

Ready-made shift registers are available as integrated circuits, such as the ’165

**Conversion of data from serial to parallel and *vice versa***

Large-scale devices such as ‘universal asynchronous receiver transmitters’ (UARTs) are based on shift registers

Same functions available in microcontrollers (‘shift’ and ‘rotate’ instructions)

# Reference

- mano BOOK

http://osp.mans.edu.eg/cs212/Seq_circuits_analysis.htm

Logic and Computer Design Fundamentals, Charles Kime & Thomas Kaminski
© 2008 Pearson Education, Inc.
(Edited by Dr. Muhamed Mudawar for COE 202 and EE 200 at KFUPM )

- https://circuitdigest.com/tutorial/what-is-shift-register-types-applications