UNIT 7

SEQUENTIAL LOGIC

COUNTERS AND REGISTERS

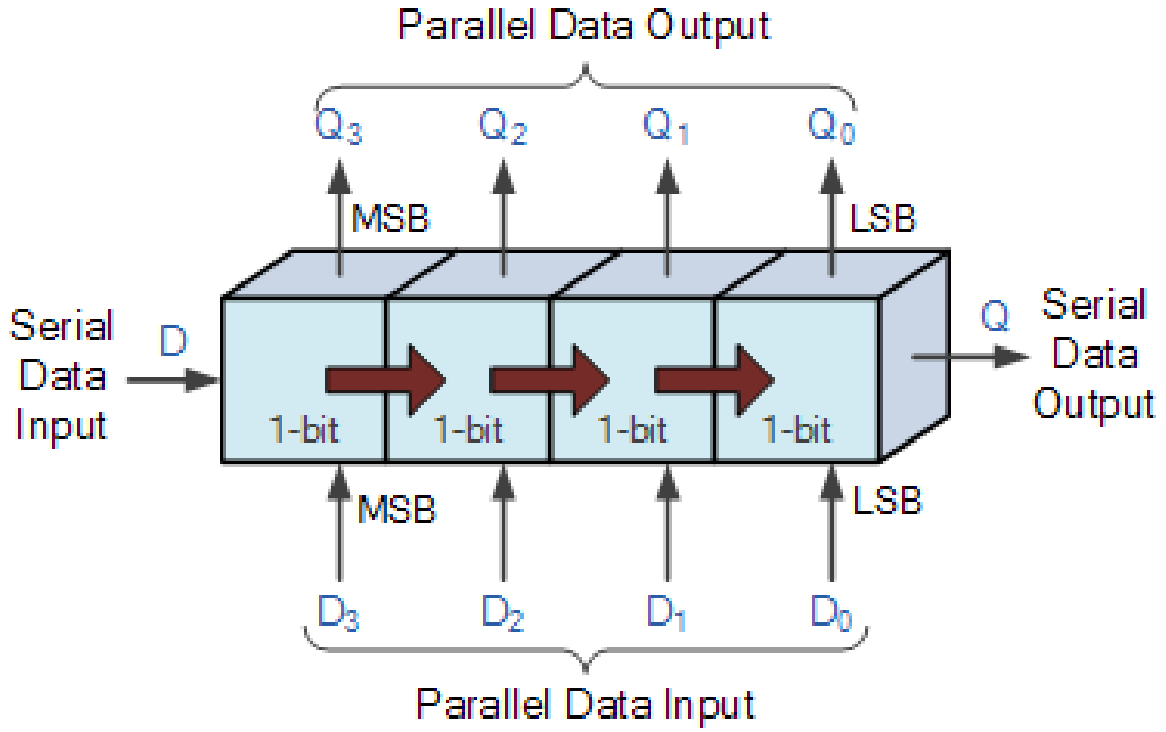
1ST SESSION

Registers

- Combination of Flip-flops (FF).
- FF stores 0 or 1, to store multiple bits, multiple FFs-***Register***.
- n FFs are connected in series in order to store n bits of data.
- Information stored within registers can be transferred using ***shift registers.***
- ***Clock pulse*** are used to shift left or right.
- n-bit shift register- made by connecting n FFs.
- Left shift of data- ***shift left registers***.
- Right shift of data- ***shift right registers***.

Shift register

- Used for the storage or the transfer of binary data.
- Loads the data present on its inputs and then moves or “shifts” it to its output once every clock cycle, hence the name **Shift Register**.
- Basically consists of several single bit “D-Type Data Latches”.
- Provided with a ***clear*** or *reset* connection so that they can be ‘***SET’*** or ***‘RESET’*** as required.



Type of shift registers

i. ***Serial-in to Parallel-out (SIPO)*** - the register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form.

ii. ***Serial-in to Serial-out (SISO)*** - the data is shifted serially "IN" and "OUT" of the register, one bit at a time in either a left or right direction under clock control.

iii. ***Parallel-in to Serial-out (PISO)*** - the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.

iv. ***Parallel-in to Parallel-out (PIPO)*** - the parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.

Data Movement Through A Shift Register

Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

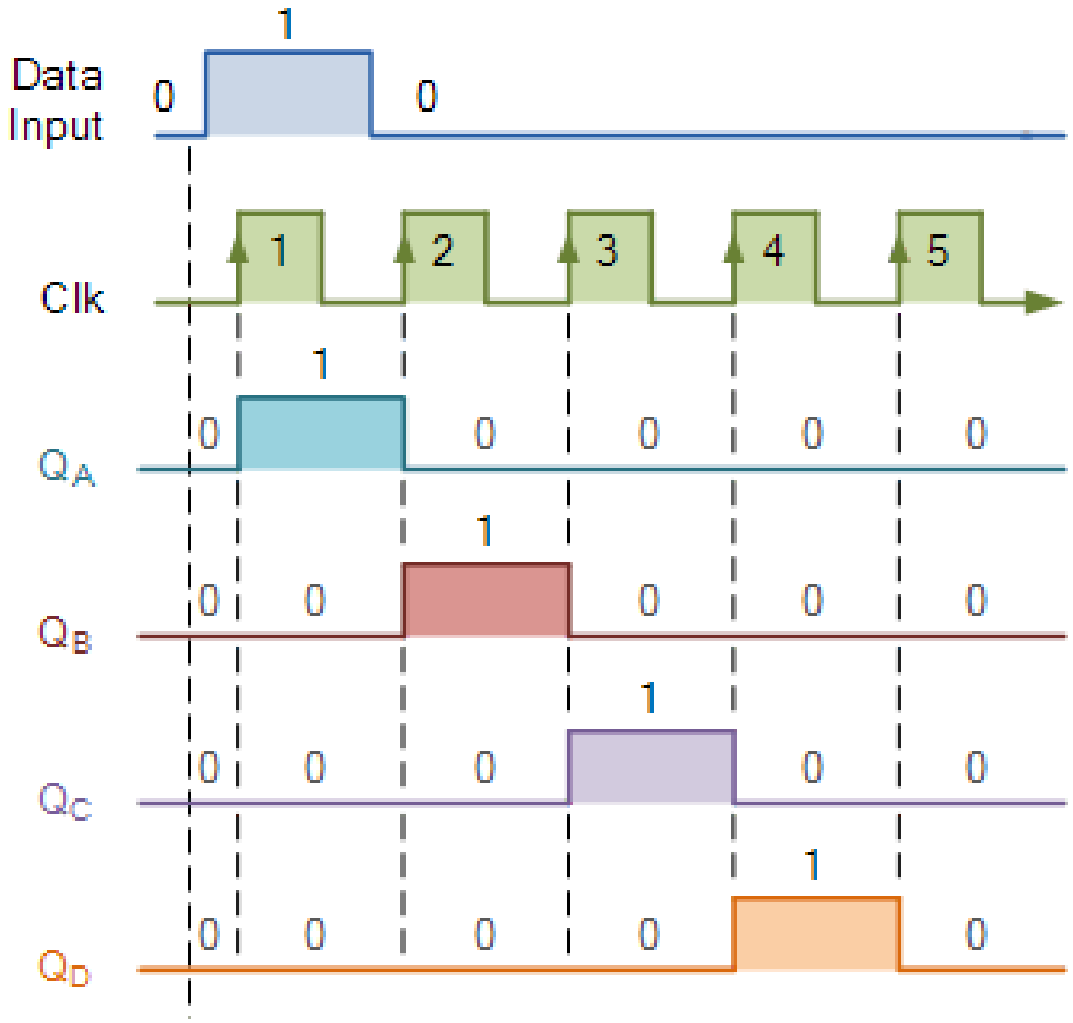


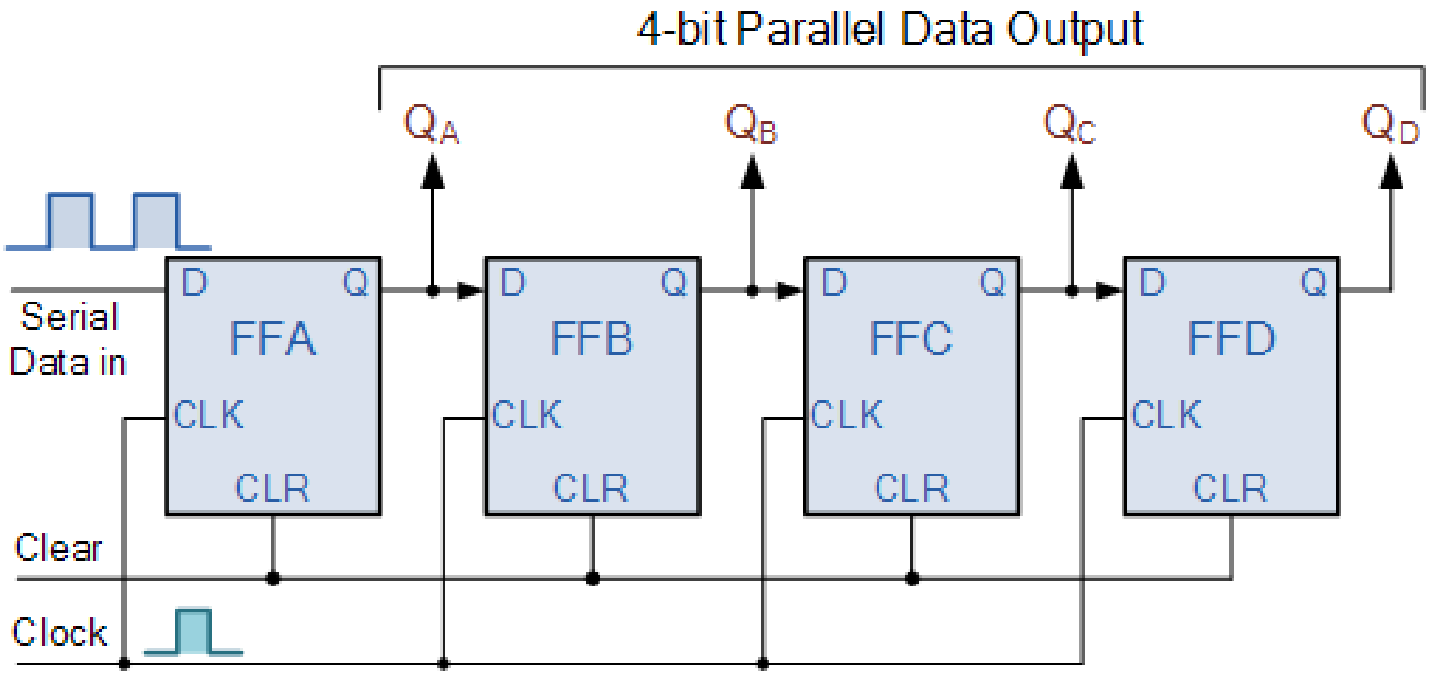
Fig: Timing Diagram

Operation:

- Lets assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D are at logic level "0" ie, no parallel data output.
  i. When ***DATA input=1***, the output of FFA i.e., Q_A is set HIGH (logic "1") on the ***first clock pulse*** and all the other outputs of remaining FFs will be LOW (logic "0").
  ii. The ***second clock pulse*** will change the output of FFA to logic "0" and the output of FFB i.e., Q_B is set HIGH as its input D has the logic "1" level on it from Q_A. *The logic "1" has now moved or been "shifted" one place along the register to the right as it is now at Q_A.*
  iii. When the ***third clock pulse*** arrives this logic "1" value moves to the output of FFC (Q_C) and so on until the arrival of the ***fifth clock pulse*** which sets all the outputs Q_A to Q_D back again to logic level "0" because the input to FFA has remained constant at logic level "0".

i. Serial-in to Parallel-out (SIPO) Shift Register

- **4-bit Serial-in to Parallel-out Shift Register:**



*Let the **input data=1101** is to be stored in register. The registers are cleared (0000) initially.*

CLK	Register content				Comments
	Q_A	Q_B	Q_C	Q_D	
Initially	0	0	0	0	***The LSB enters first.*** Data is completely stored after the 4th clock. Output is available all at a time simultaneously.
1 →	1 ↘	0	0	0	
2 →	0 ↘	1 ↘	0	0	
3 →	1 ↘	0 ↘	1 ↘	0	
4 →	**1**	**1**	**0**	**1**	

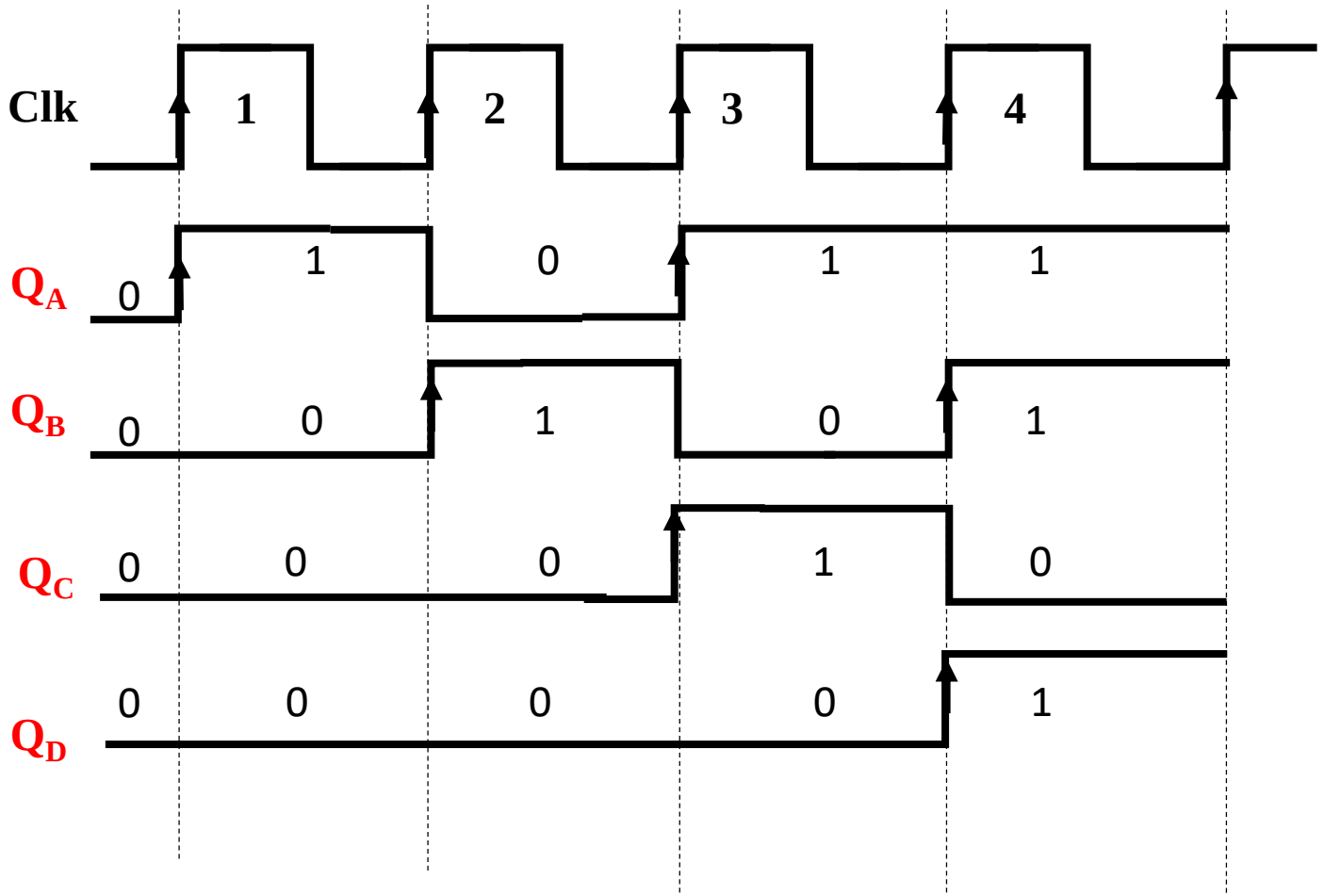
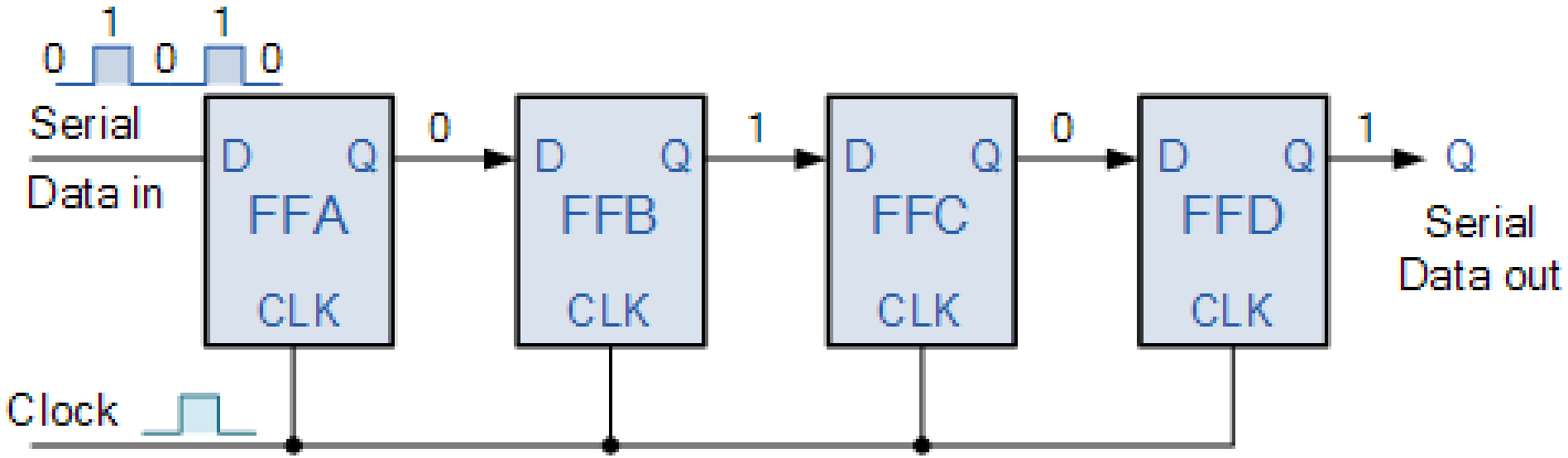


Fig: Timing Diagram for storing 1101 in SIPO shift register

ii. Serial-in to Serial-out (SISO) Shift Register

- **4-bit Serial-in to Serial-out Shift Register**



- Data is allowed to flow straight through the register and out of the other end.
- Since there is only one output, the DATA leaves the shift register one bit at a time in a serial pattern, hence the name **Serial-in to Serial-Out Shift Register** or **SISO**.

Let's explain the operation of ***SISO*** by storing and retrieving data-'***1011***'. Initially, let us assume register content is cleared.

Clock	Register Content				Remarks
	Q_0	Q_1	Q_2	Q_3	
Initially	0	0	0	0	LSB enters first.
1	1	0	0	0	
2	1	1	0	0	
3	0	1	1	0	
4	1	0	1	1	All data bits are stored after 4th clock pulse.
5	0	1	0	1	
6	0	0	1	0	
7	0	0	0	1	Register get cleared after 8th clock pulse
8	0	0	0	0	

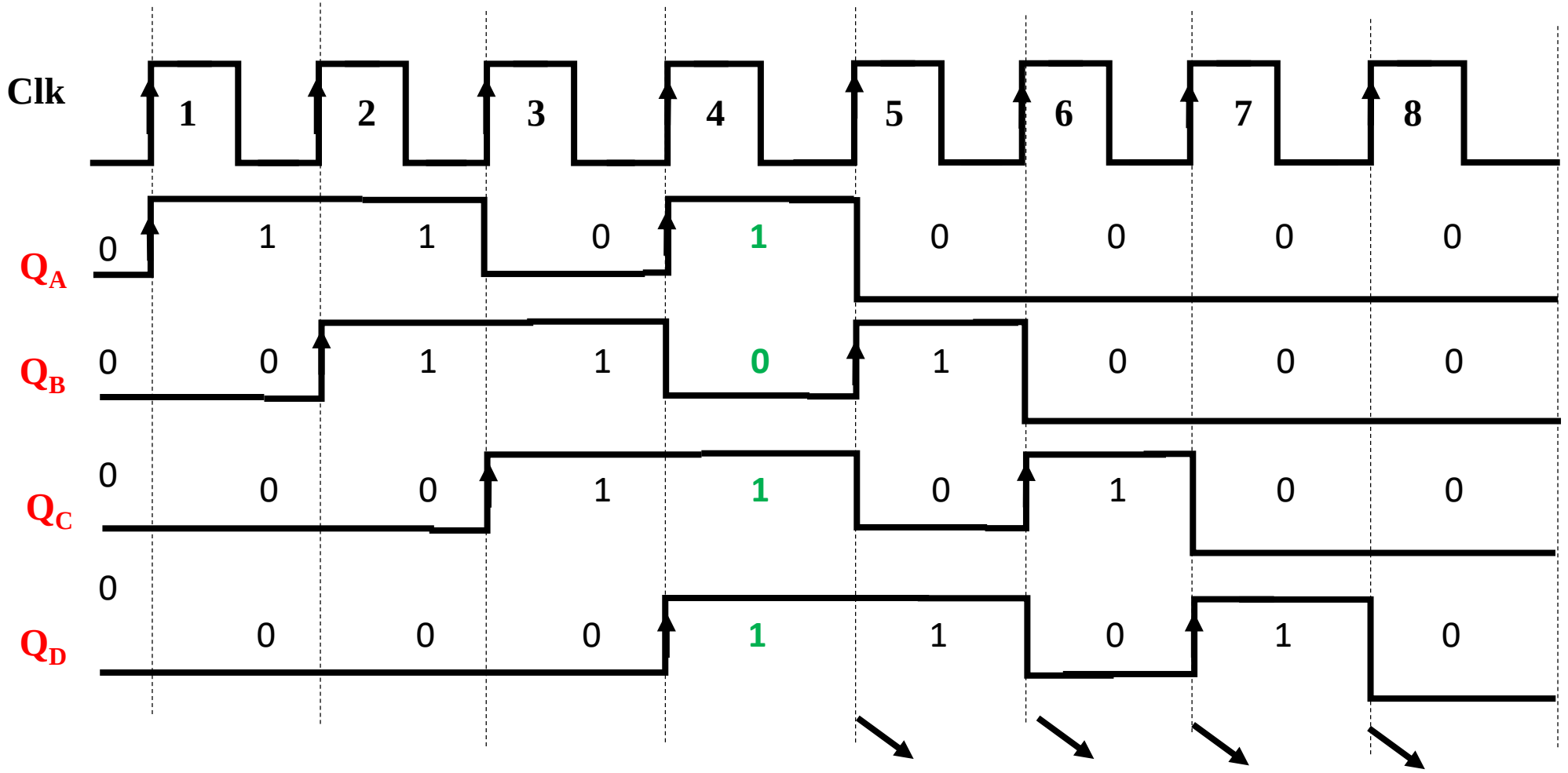
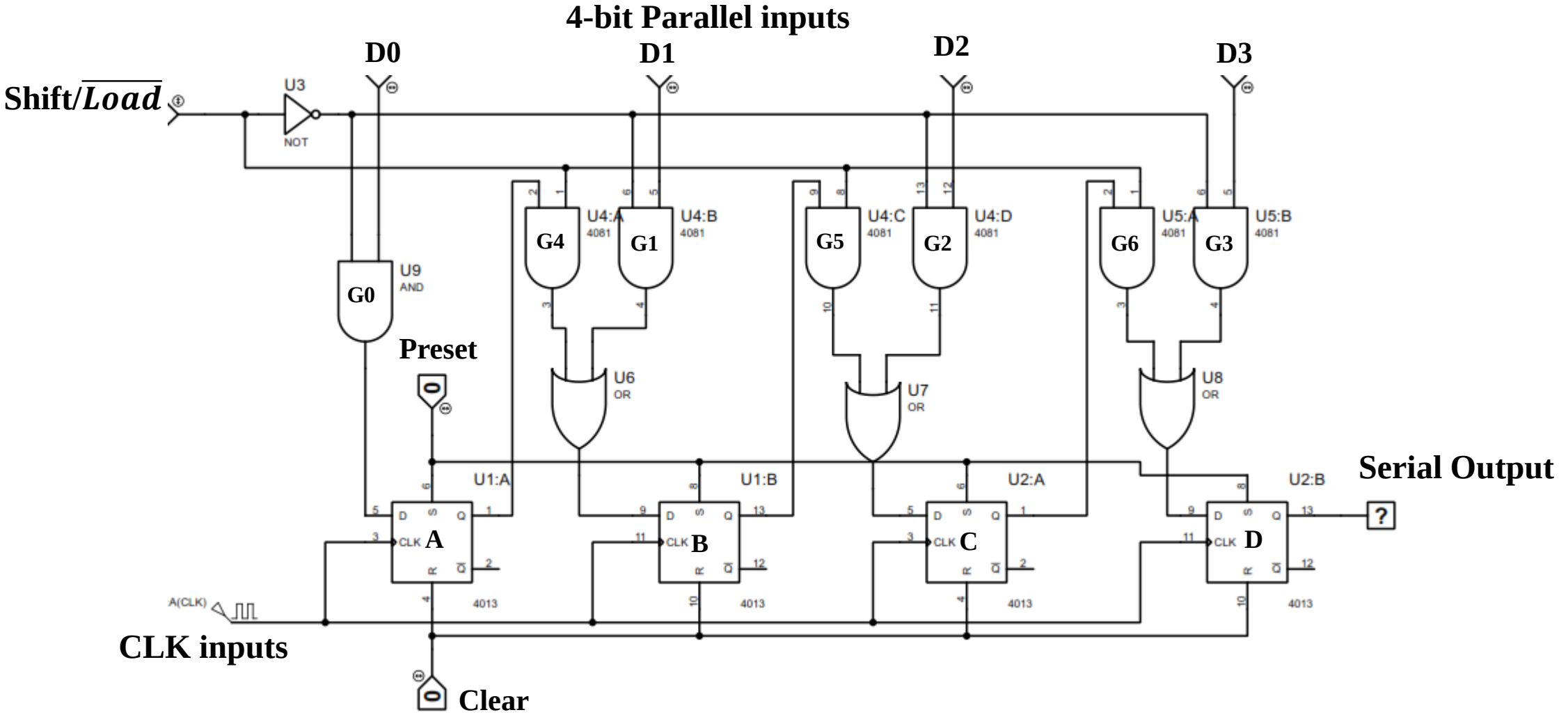


Fig: Timing Diagram for storing and retrieving 1011 in SISO shift register

iii. Parallel-in to Serial-out (SISO) Shift Register

- **4-bit Parallel-in to Serial-out Shift Register**



Operation:

- Data are entered parallel but taken in a serial mode.
- When **Shift/$\overline{Load}$**=**LOW;** ***G0,G1,G2,G3*** gates are enabled, allowing to pass data ***D0,D1,D2,D3*** to flip-flop (A-D) input D parallelly.
- When **Shift/$\overline{Load}$**=**HIGH;** ***G4,G5,G6*** gates are enabled, allowing bits to shift right from one stage to another on each clock pulse.
- Output is taken at Q_D serially.

Let us consider an input data as, $D_0D_1D_2D_3$=(1010).

On a 1st clock pulse, **Shift/*Load*=LOW;** so, parallel data (1010) get loaded into register, making $Q_D = 0$.

On 2nd clock pulse, **Shift/*Load*=HIGH;** data (1010) get shifted right into register, making $Q_D = 1$ and so on.

	Register Content				
CLK	**Q_A**	**Q_B**	**Q_C**	**Q_D**	**Remarks**
1	**1**	**0**	**1**	**0**	Data entered parallelly
2	0	1	0	1	
3	0	0	1	0	
4	0	0	0	1	Data leaving serially
5	0	0	0	0	

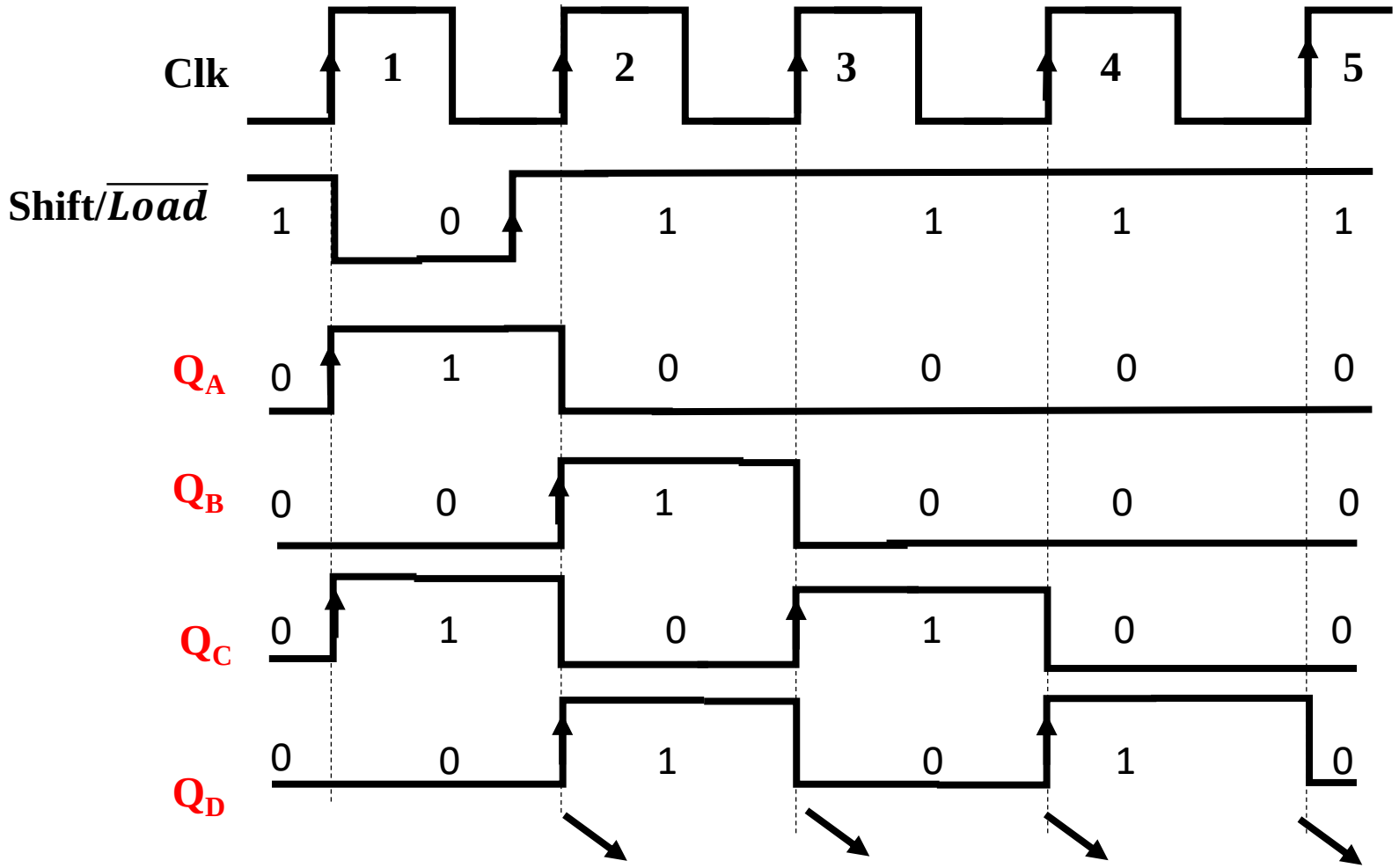
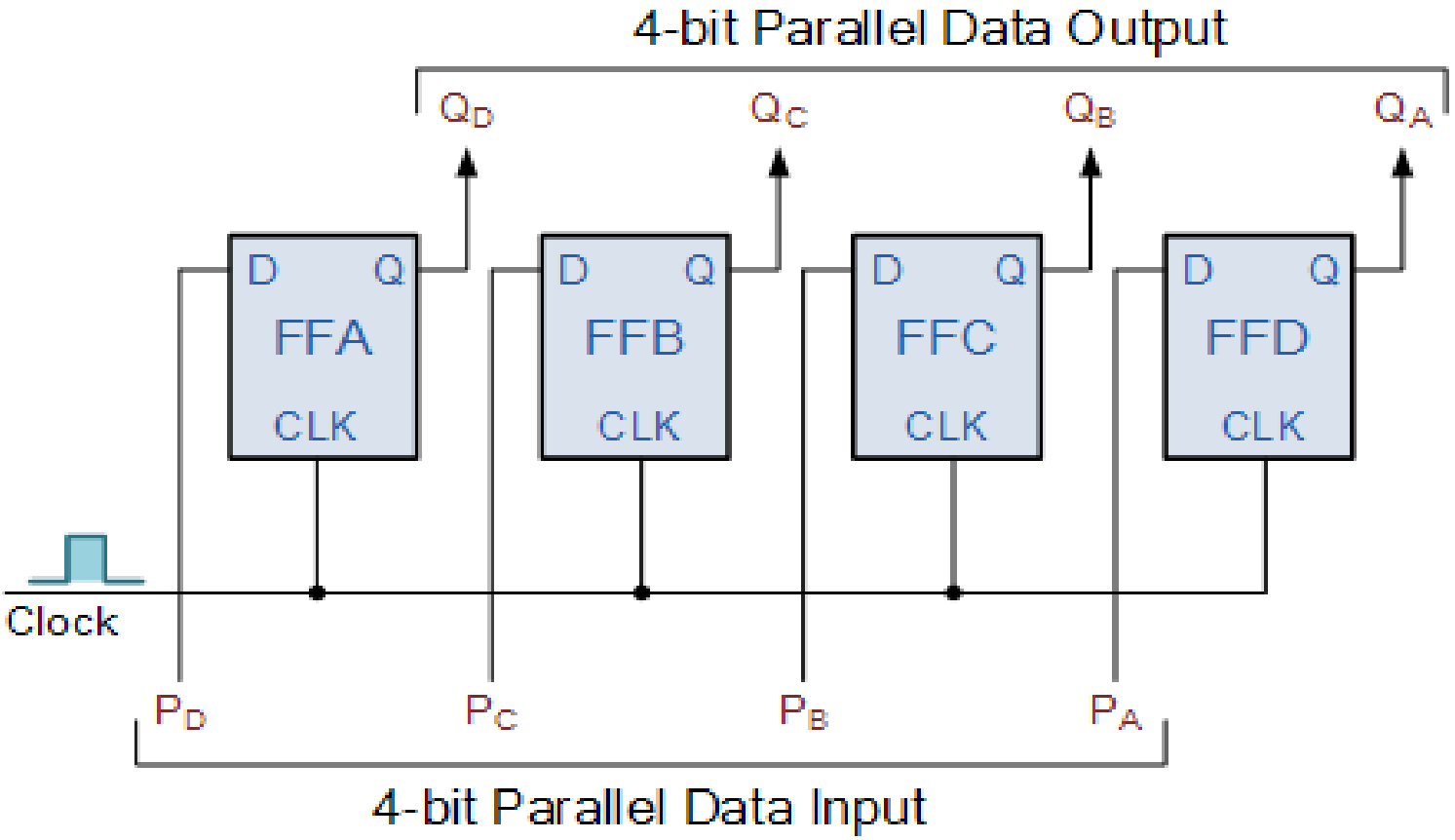


Fig: Timing Diagram for storing 1010 in PISO shift register

iv. Parallel-in to Parallel-out (SISO) Shift Register

- **4-bit Parallel-in to Parallel-out Shift Register**

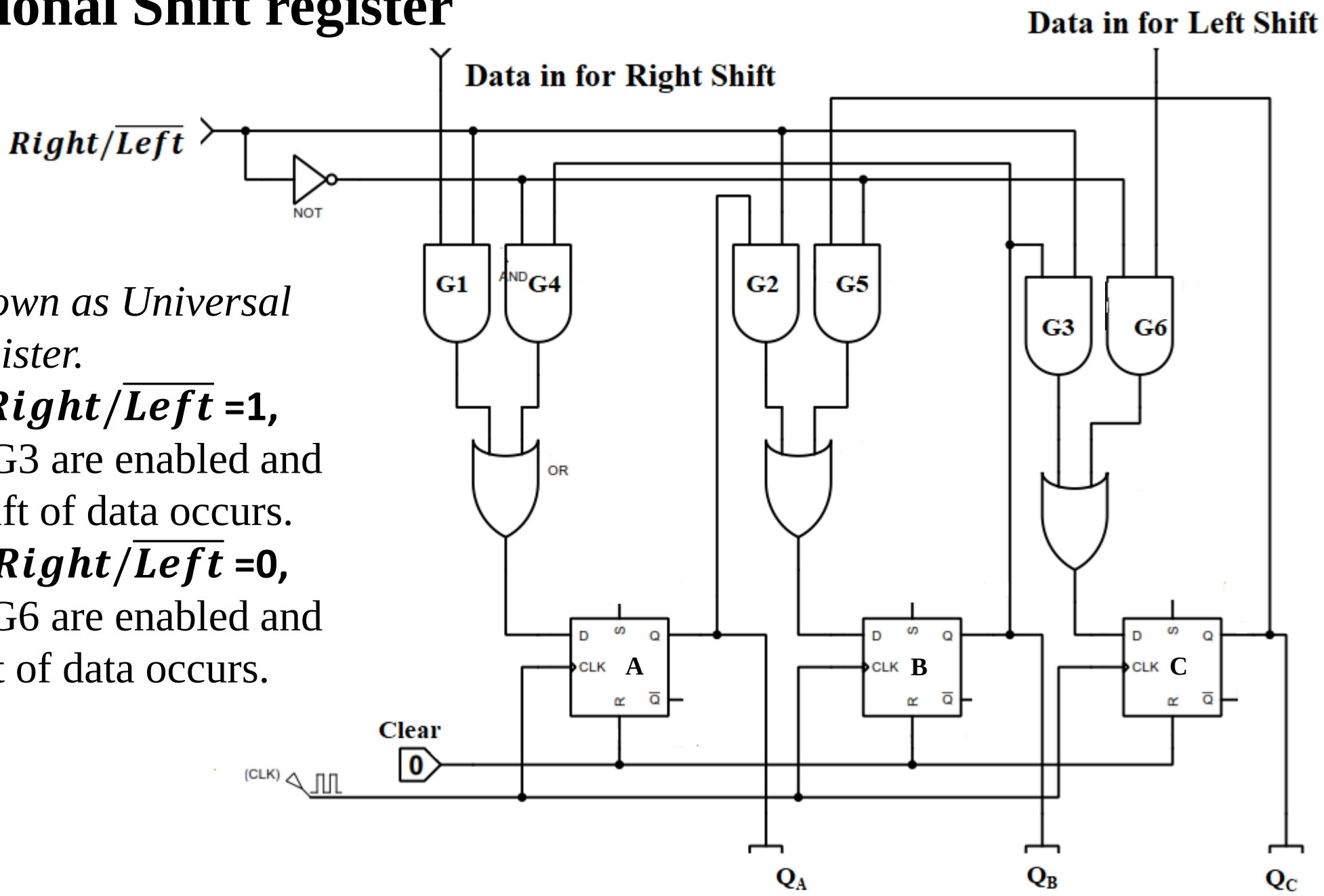


Bi-directional Shift register

- *Also known as Universal shift register.*

i. W*hen* $\boldsymbol{Right/\overline{Left}}$ **=1,** G1,G2,G3 are enabled and right shift of data occurs.

ii. W*hen* $\boldsymbol{Right/\overline{Left}}$ **=0,** G4,G5,G6 are enabled and left shift of data occurs.



Ring Counter

- Typical application of shift register.
- The only change in the output of last FF is connected to the input of first FF.
- ***No. of states*** **=*no. of FF used in this case*** (Special case).
- Used to control the sequence of stepper motor.
- E.g.: Initial register pattern values 1000

CLK	Q_A	Q_B	Q_C	Q_D
0	**1**	**0**	**0**	**0**
1	**0**	**1**	**0**	**0**
2	**0**	**0**	**1**	**0**
3	**0**	**0**	**0**	**1**
4	**1**	**0**	**0**	**0**

Note: ***In counter, no. of states are generally the max. limit up to which counter can count, given by 2^n, where n=no. of bits used.***

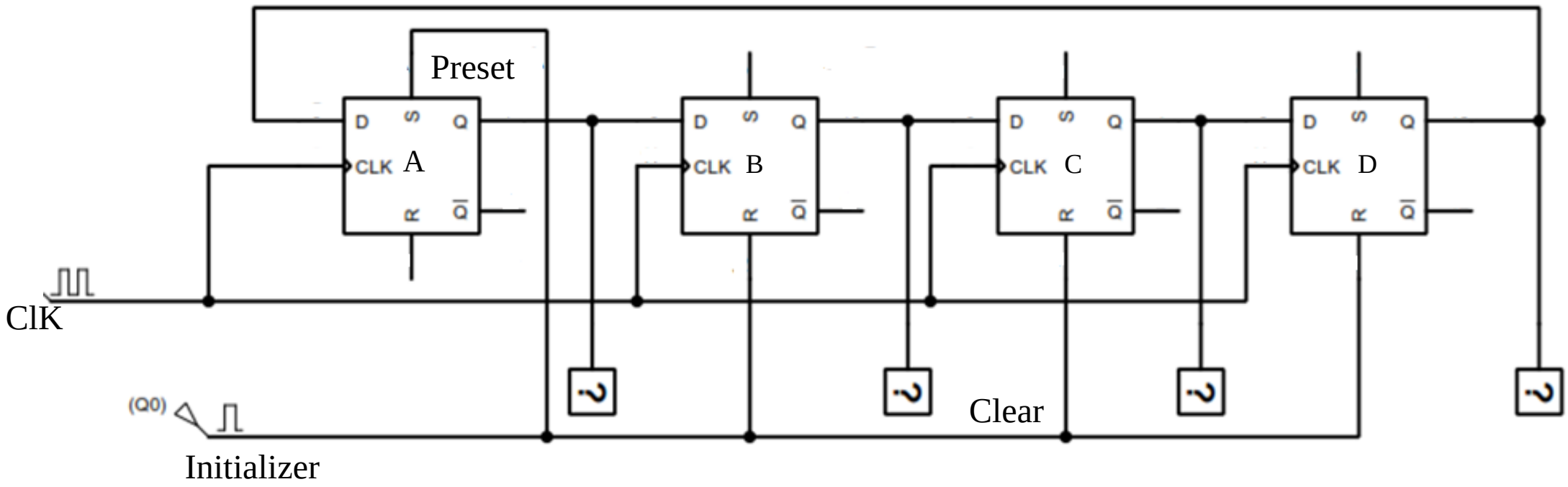


Fig: Logic Diagram of Ring Counter

- Initially, Preset =1, Clear=1 should be given and when Preset =0, Clear=0, at every clock pulse the 1 is shifted to next stage.
- ***Main drawback is needs initialization.***

Johnson Counter

- Also known as ***Switch-Tail counter.***
- Eliminates the limitation of ring counter.
- The complement output of last FF is connected to the input of first FF.

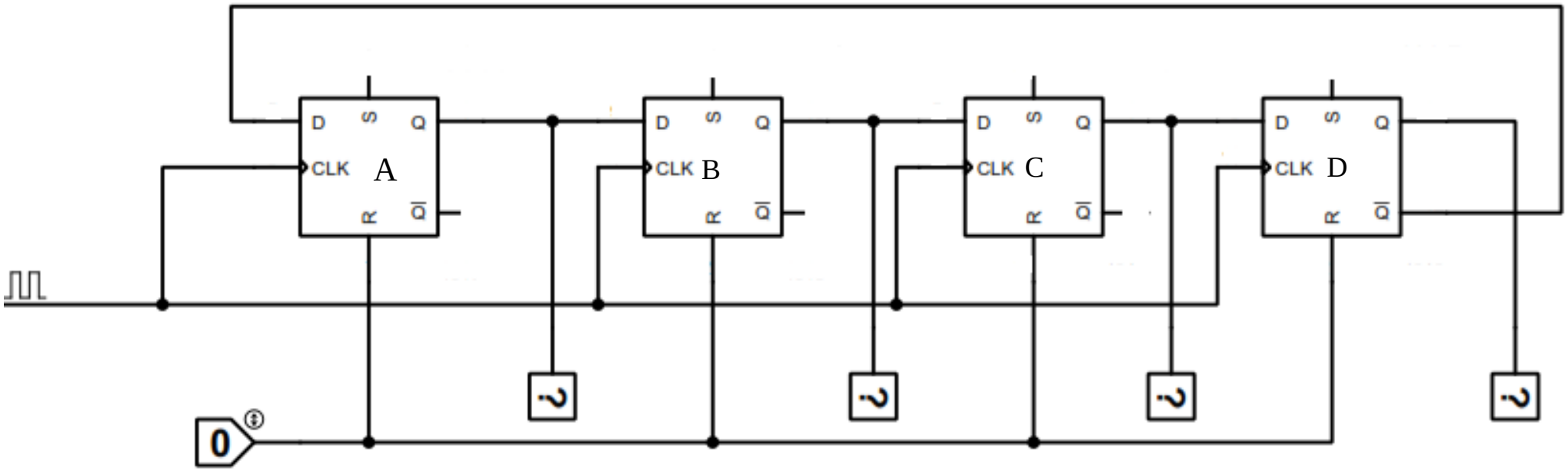


Fig: Logic Diagram of Johnson Counter

Initially, **Clear =1**, to reset the FFs to 0000 State.

CLK	Q_A	Q_B	Q_C	Q_D
0	**1**	**0**	**0**	**0**
1	**1**	**1**	**0**	**0**
2	**1**	**1**	**1**	**0**
3	**1**	**1**	**1**	**1**
4	**0**	**1**	**1**	**1**
5	**0**	**0**	**1**	**1**
6	**0**	**0**	**0**	**1**
7	**0**	**0**	**0**	**0**