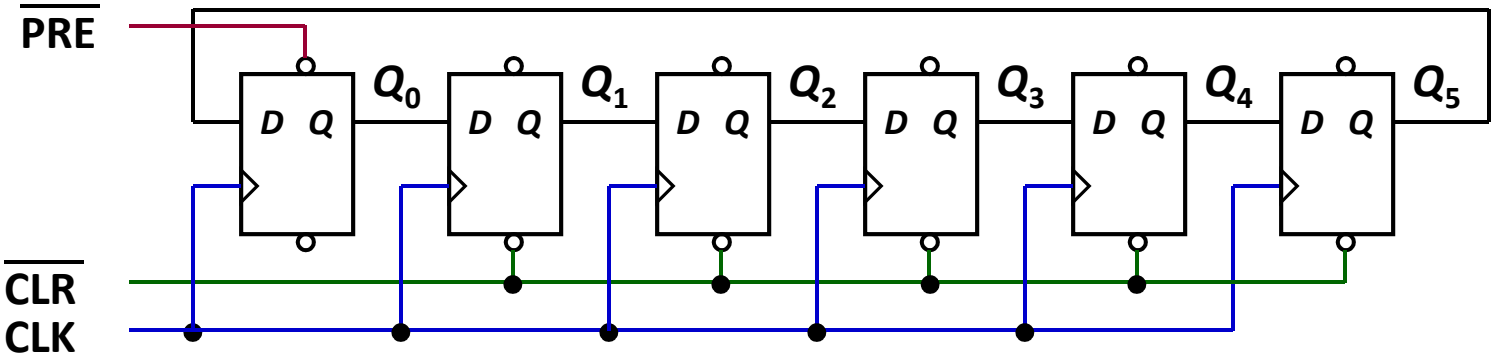
Ring Counters

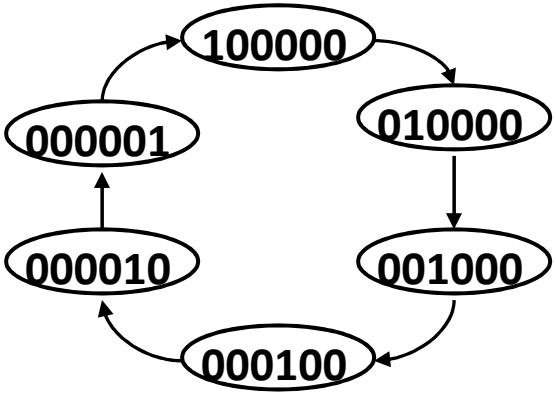
Example: A 6-bit (MOD-6) ring counter.

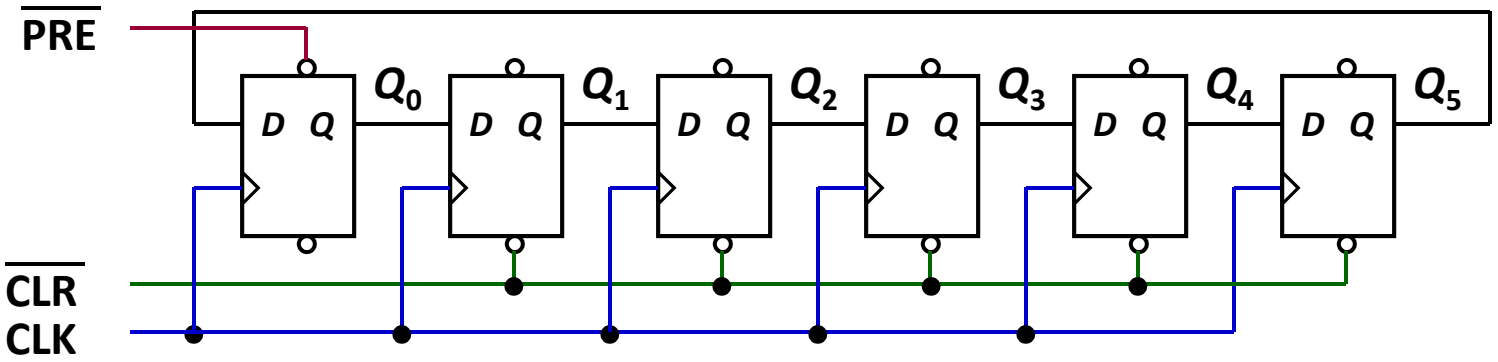
At starting $\overline{PRE}$ **= 0,** $\overline{CLR}$**=0**

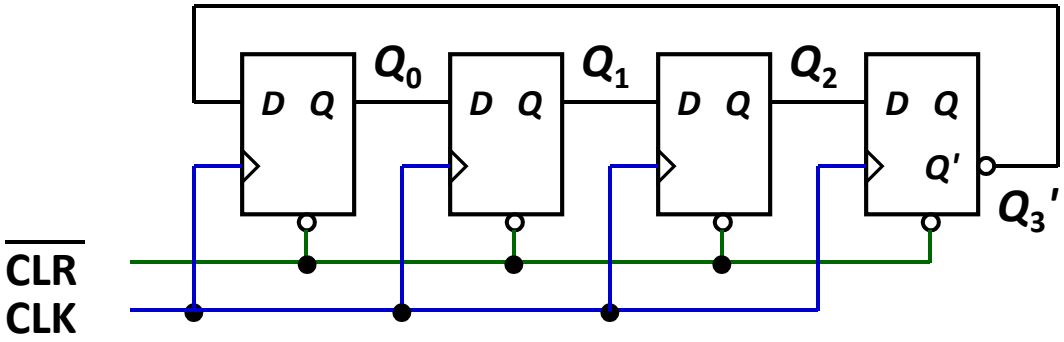
In Normal Condition $\overline{PRE}$**=1,** $\overline{CLR}$**= 1**



Clock	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5
0	1	0	0	0	0	0
1	0	1	0	0	0	0
2	0	0	1	0	0	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1

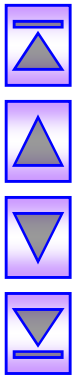



PRE
D Q
Q_0
Q_1
Q_2
Q_3
Q_4
Q_5
CLR
CLK


D Q
Q_0
Q_1
Q_2
Q'
Q_3'
CLR
CLK

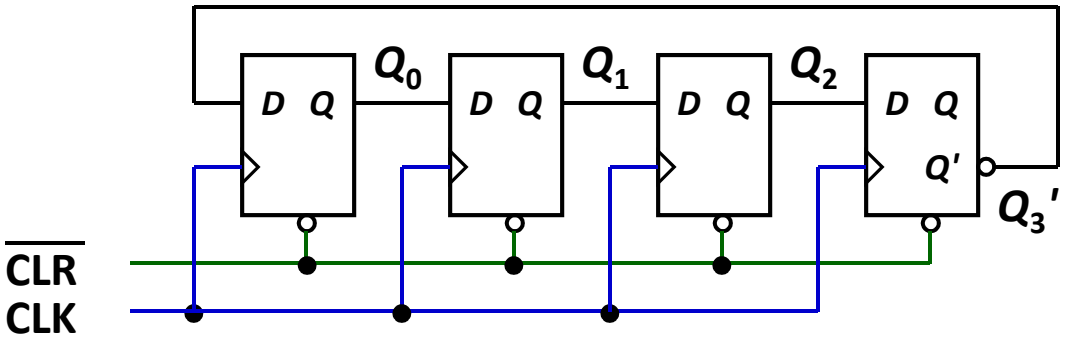
Johnson Counters

- The complement of the output of the last stage is connected back to the D input of the first stage.
- Also called the *twisted-ring counter*.
- Require fewer flip-flops than ring counters but more flip-flops than binary counters.
- An *n*-bit Johnson counter cycles through 2*n* states.
- Require more decoding circuitry than ring counter but less than binary counters.

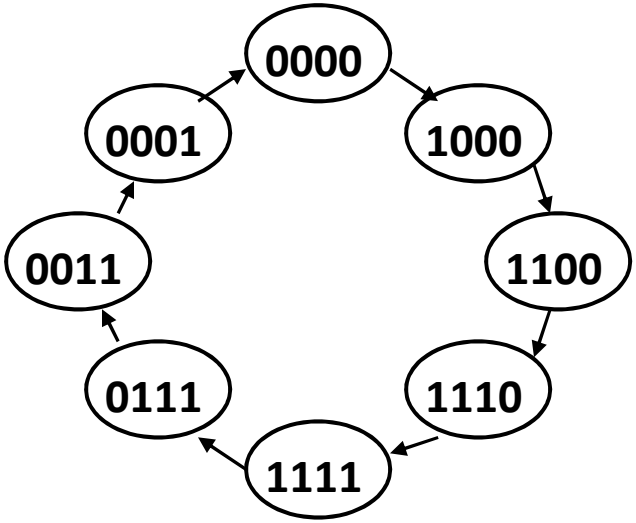


- Example: A 4-bit (MOD-8) Johnson counter.

At Starting $\overline{CLR}$= 0
Normal Condition CLR= 1



Clock	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



Registers

- A register is a digital device used for storage.
- Register is a group of Flip flop to store a data (0 or 1).
- To store ‘n’ no. of data , ‘n’ bit register is required which consist of ‘n’ no. of flip flop.
- **Types of registers:**

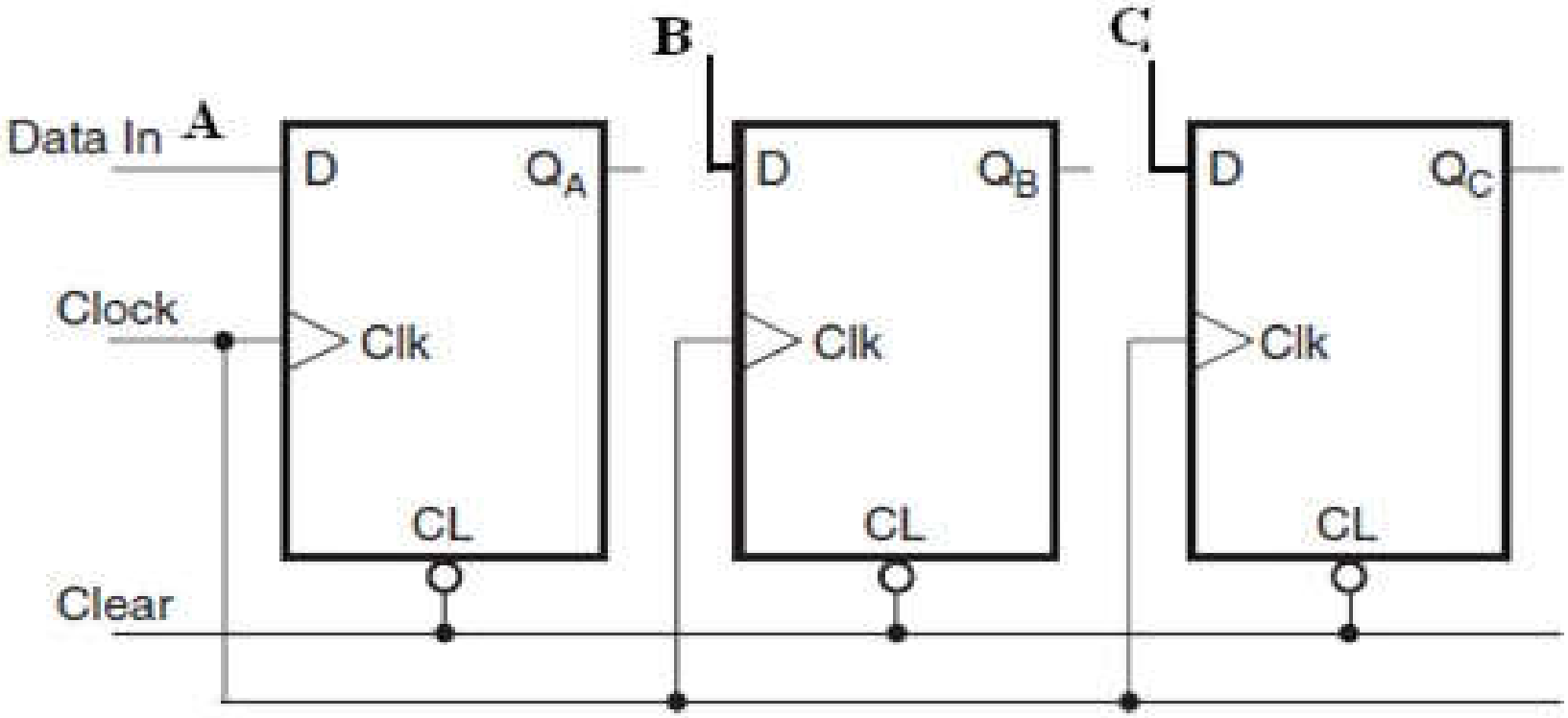
  Buffer Register

  Shift Register

Buffer Register

- An ‘n’ bit registers has group of ‘n’ flip flop and capable to store any binary information, which contains ‘n’ numbers of bits.
- This type of register is also called storage registers
- These are used for temporary storage of data

Buffer Register

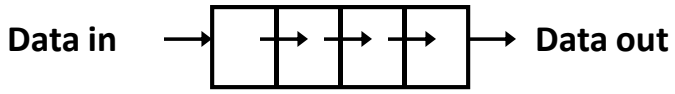


Shift Registers

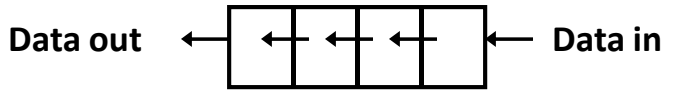
- Another function of a register, besides storage, is to provide for *data movements*.
- Each *stage* (flip-flop) in a shift register represents one bit of storage, and the shifting capability of a register permits the movement of data from stage to stage within the register, or into or out of the register upon application of clock pulses.

Types of Shift Registers

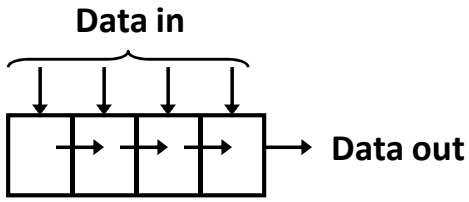
- Basic data movement in shift registers (four bits are used for illustration).



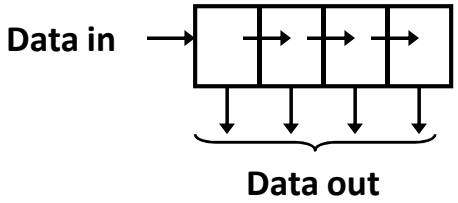
(a) Serial in/shift right/serial out



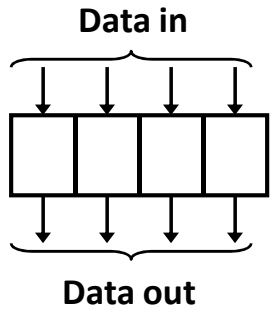
(b) Serial in/shift left/serial out



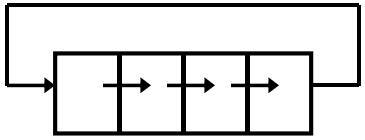
(c) Parallel in/serial out



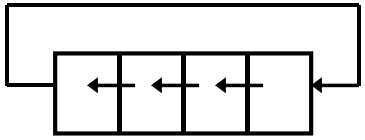
(d) Serial in/parallel out



(e) Parallel in / parallel out



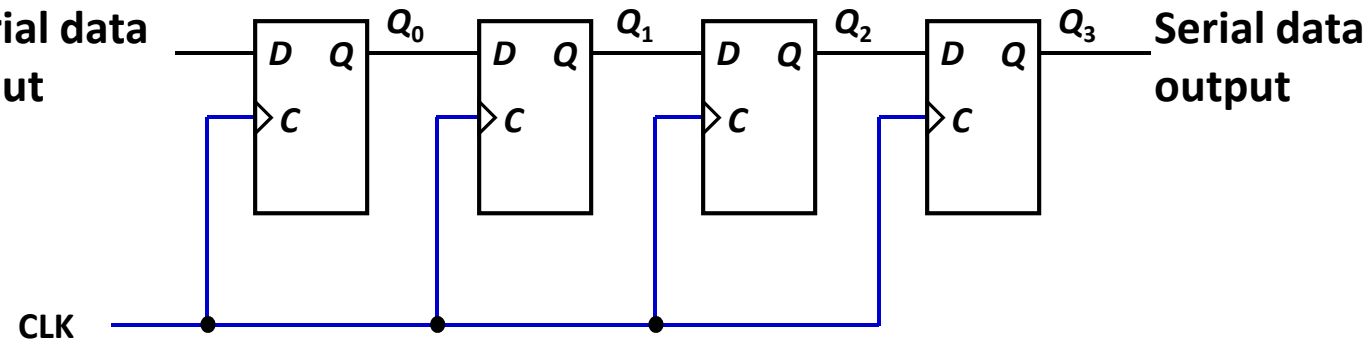
(f) Rotate right



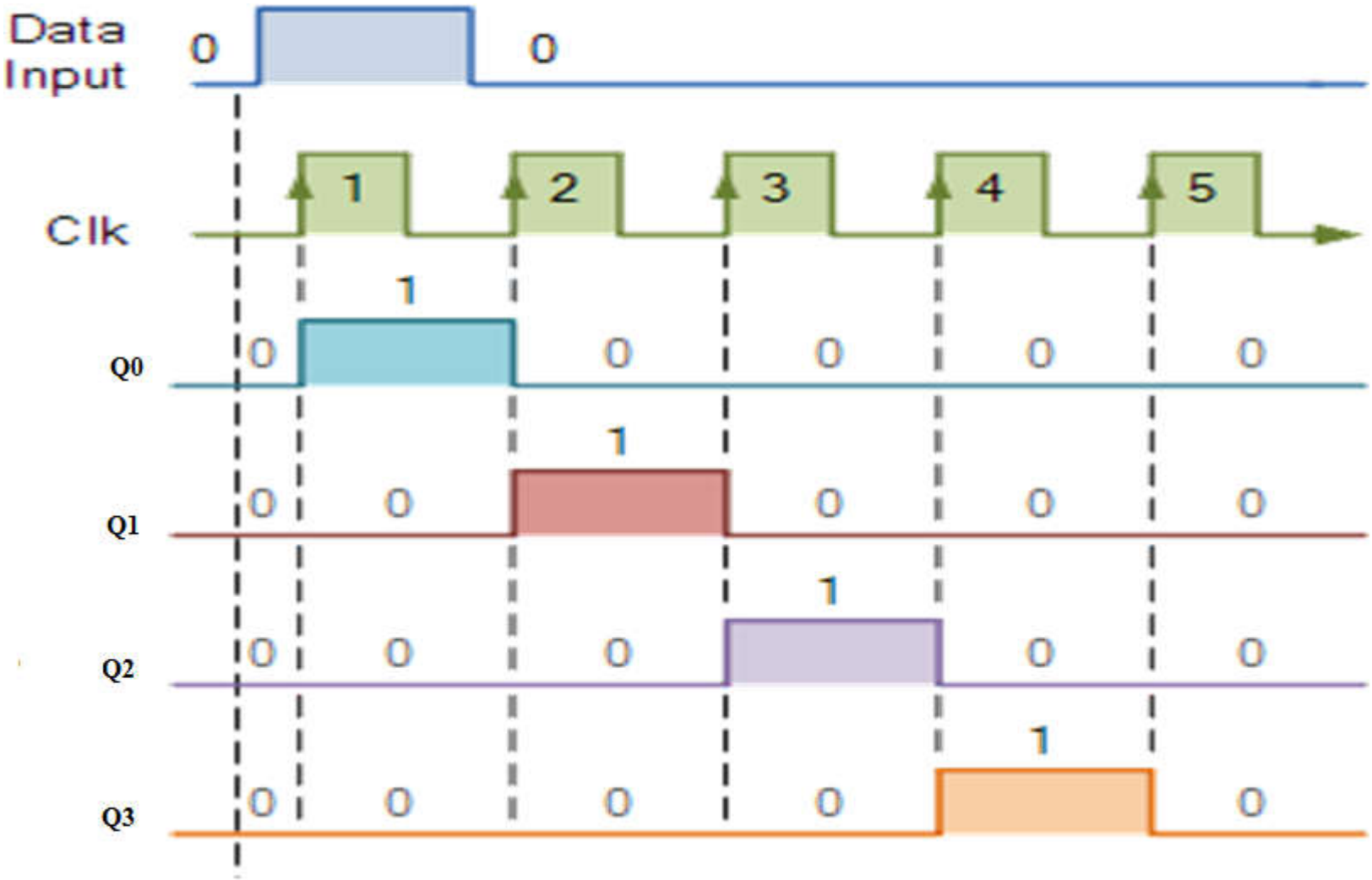
(g) Rotate left

Serial In/Serial Out Shift Registers

- Accepts data serially – one bit at a time – and also produces output serially.

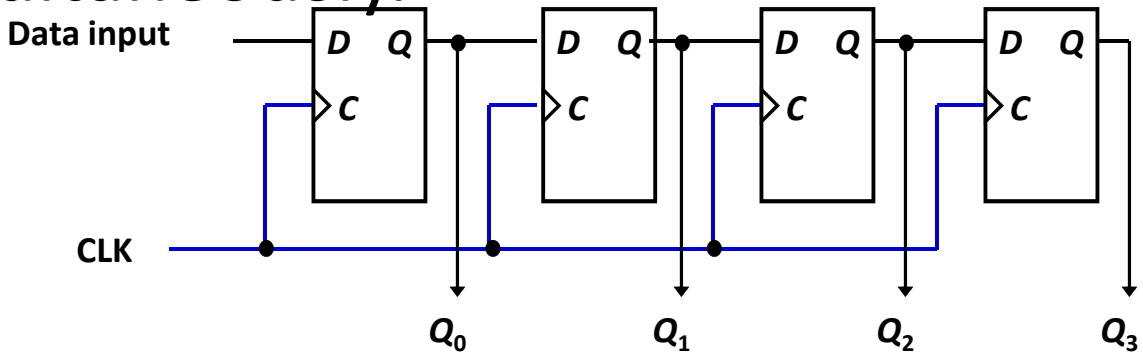


Output waveform for shift Register



Serial In/Parallel Out Shift Registers

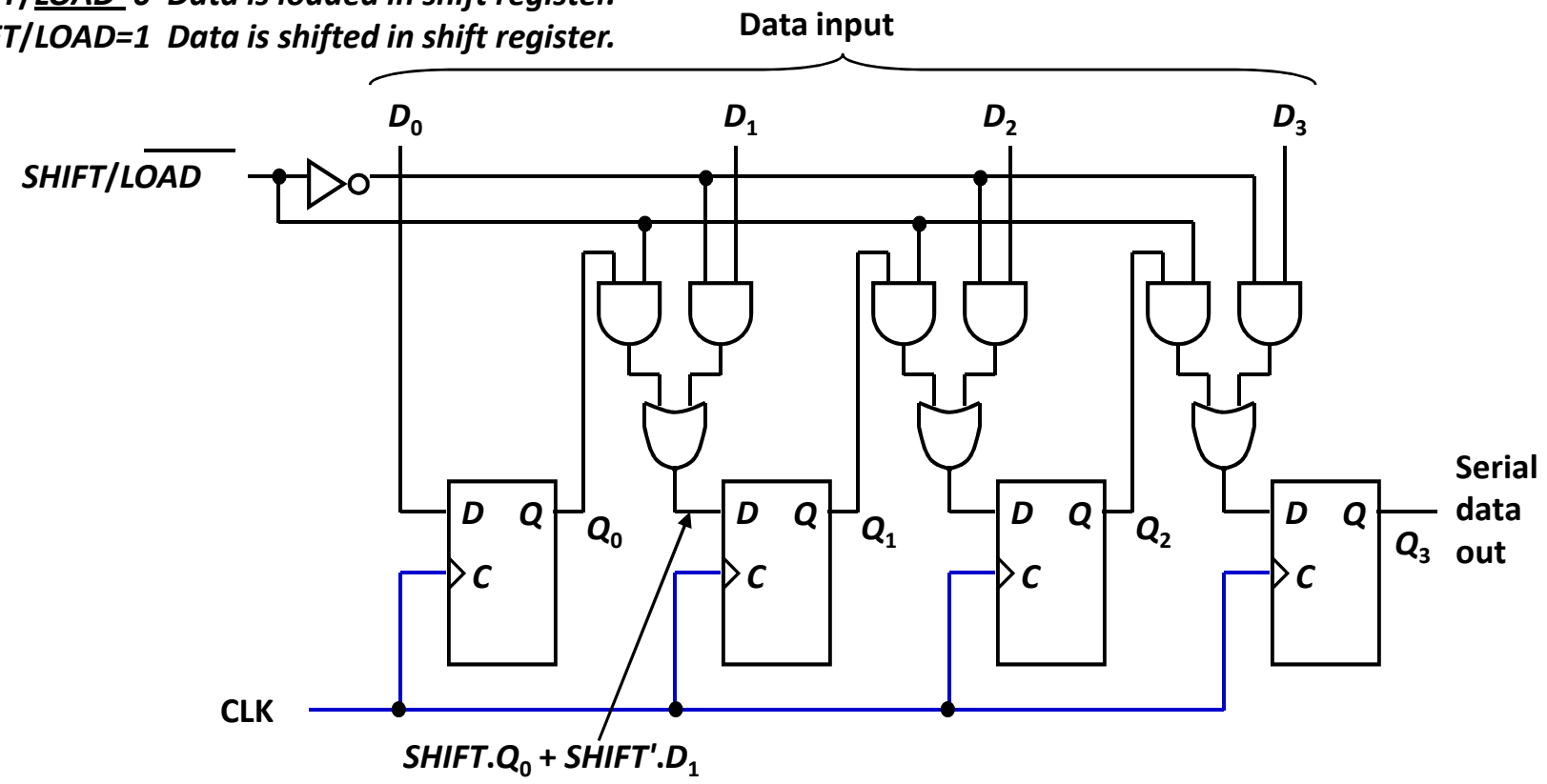
- Accepts data serially.
- Outputs of all stages are available simultaneously.



Logic symbol

- Bits are entered simultaneously, but output is serial.

If SHIFT/$\overline{LOAD}$=0 Data is loaded in shift register.
If SHIFT/LOAD=1 Data is shifted in shift register.



Parallel In/Parallel Out Shift Registers

- Simultaneous input and output of all data bits.