SHIFT REGISTERS

INTRODUCTION

- A sequential logic circuit, mainly for storage of digital data and movement
- A group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop.
- All the flip-flops are driven by a common clock, and all are set or reset simultaneously.
- Types:
  - Serial In - Serial Out,
  - Serial In - Parallel Out,
  - Parallel In - Serial Out,
  - Parallel In - Parallel Out, and
  - Bidirectional shift registers.
  - A special form of counter - the shift register counter, (Ring and Johnson)

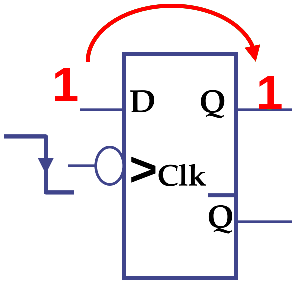
Remember: A flip-flop stores one bit of information

When you want to store n bits → register,

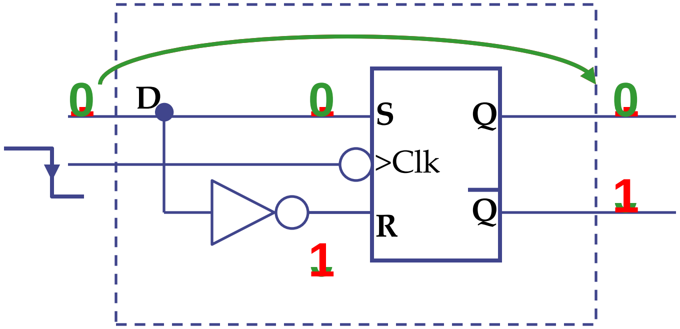
- 'n' flip-flops used,
- Clock is shared by all so action is synchronous with clock edge

REVIEW: THE D FLIP-FLOP

- D flip flop is used to store binary **D**ata.
- The logic level at the "D" (data) input is transferred to the Q output when the clock is asserted.
- It remains stored at output Q until the clock is asserted a second time.



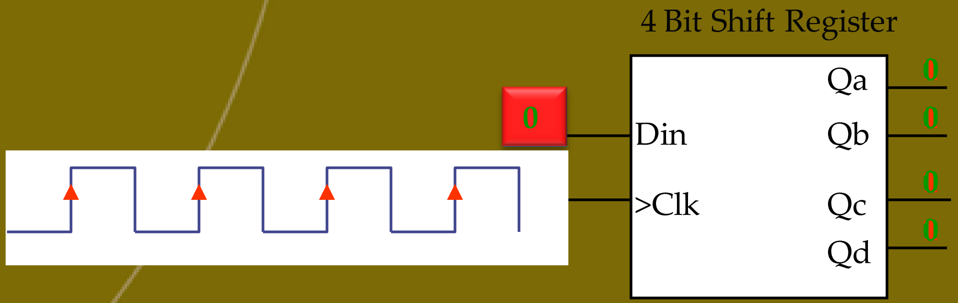
- If look inside the D flip flop, you can see how it works.



- Let's assume the initial conditions at Q are 0.
- Set D to logic 1.
- Show the logic levels at S and R
- The internal SR inputs are in the set mode. The flip flop will set if D=1 and the clock is asserted. Q = D!
- Set D to logic 0.
- The internal SR inputs are in the reset mode. The flip flop will reset if D=0 and the clock is asserted. Q = D again!

4-BIT SHIFT REGISTER

- The **4-bit shift register** can shift data from **QA to QB to QC to QD**. The shift register will be filled with 1's then it will be filled with 0's.
- Assume the initial conditions are : Qa=Qb=Qc=Qd=0.
- Din is the input to **QA**. The data at **QA** shifts to **QB. **The data at **QB** shifts to **QC**. **QC** shifts to **QD**.
- Apply a logic 1 at input Din and clock the shift register 4 times.
- You have seen how it takes **4** clock pulses to fill the shift register with 1's. Din will now be connected to logic 0. This will shift the 1's out and fill the shift register with **0's.**



FUNCTIONS OF SHIFT REGISTER

- **Storage Capacity**
  - ✓ Total no. of bits of digital binary word (data) that can be stored
  - ✓ Each flip-flop in shift register stores 1 bit
  - ✓ The no. of flip-flop in shift register determines storage capacity

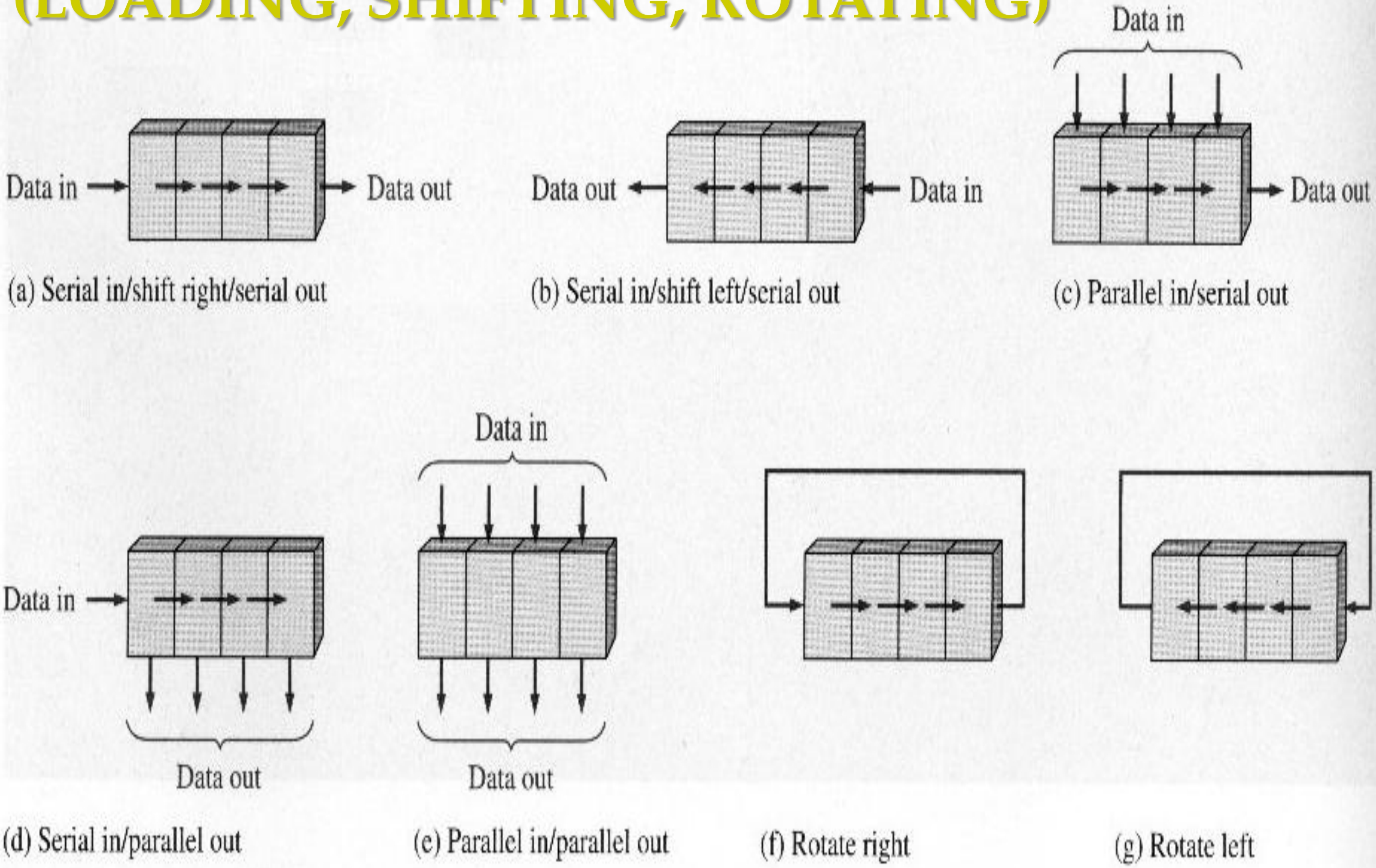
- **Shifting Capability**
  - ✓ Movement of data from stage to stage within register or into or out of register upon the application of clock pulse
  - ✓ Shifting can be done either in right direction or left direction

OPERATIONS ON SHIFT REGISTER

- Parallel and Serial Data Transmission
  - ✓ Shift registers are often used to interface digital systems situated remotely from each other.
  - ✓ Task: We want to transmit an n-bit quantity between two location that are far from each other.
- Options:
  - ✓ Use n lines to transmit n bits in parallel. Problem: Cost is expensive.
  - ✓ Use a single line to transmit the information serially, one bit at a time. Cost is less.

OPERATIONS ON SHIFT REGISTER

(LOADING, SHIFTING, ROTATING)



(a) Serial in/shift right/serial out

(b) Serial in/shift left/serial out

(c) Parallel in/serial out

(d) Serial in/parallel out

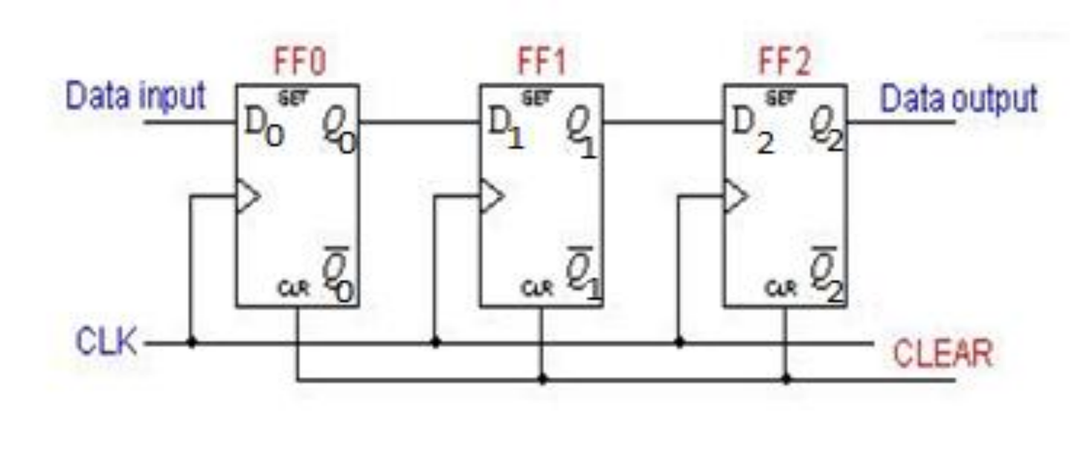
(e) Parallel in/parallel out

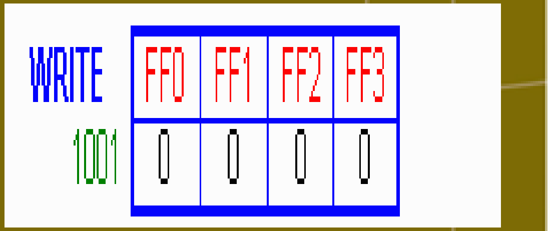
(f) Rotate right

(g) Rotate left

SERIAL IN SERIAL OUT (SISO)

- Data entered into a register serially one bit at a time
- Output is also obtained in serial form
- On the leading edge of the first clock pulse, the signal on the INPUT (data_input) is latched in the first flip-flop.
- On the leading edge of the next clock pulse, the contents of the first flip-flop is stored in the second flip-flop, and the signal which is present at the INPUT (data_input) is stored in the first flip-flop, etc.
- A 3-bit SISO shift register can be constructed using 3-D flip-flops
- Clock is provided in synchronous way in positive edge





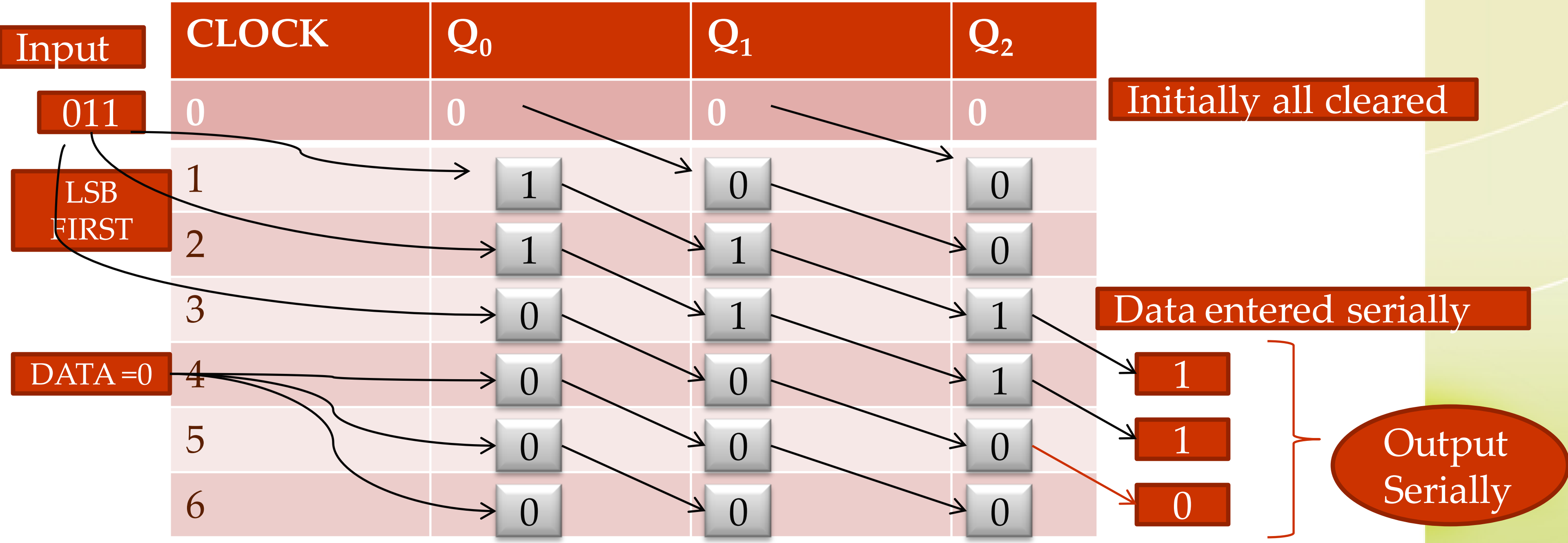
WRITE

1001

FF0	FF1	FF2	FF3
0	0	0	0

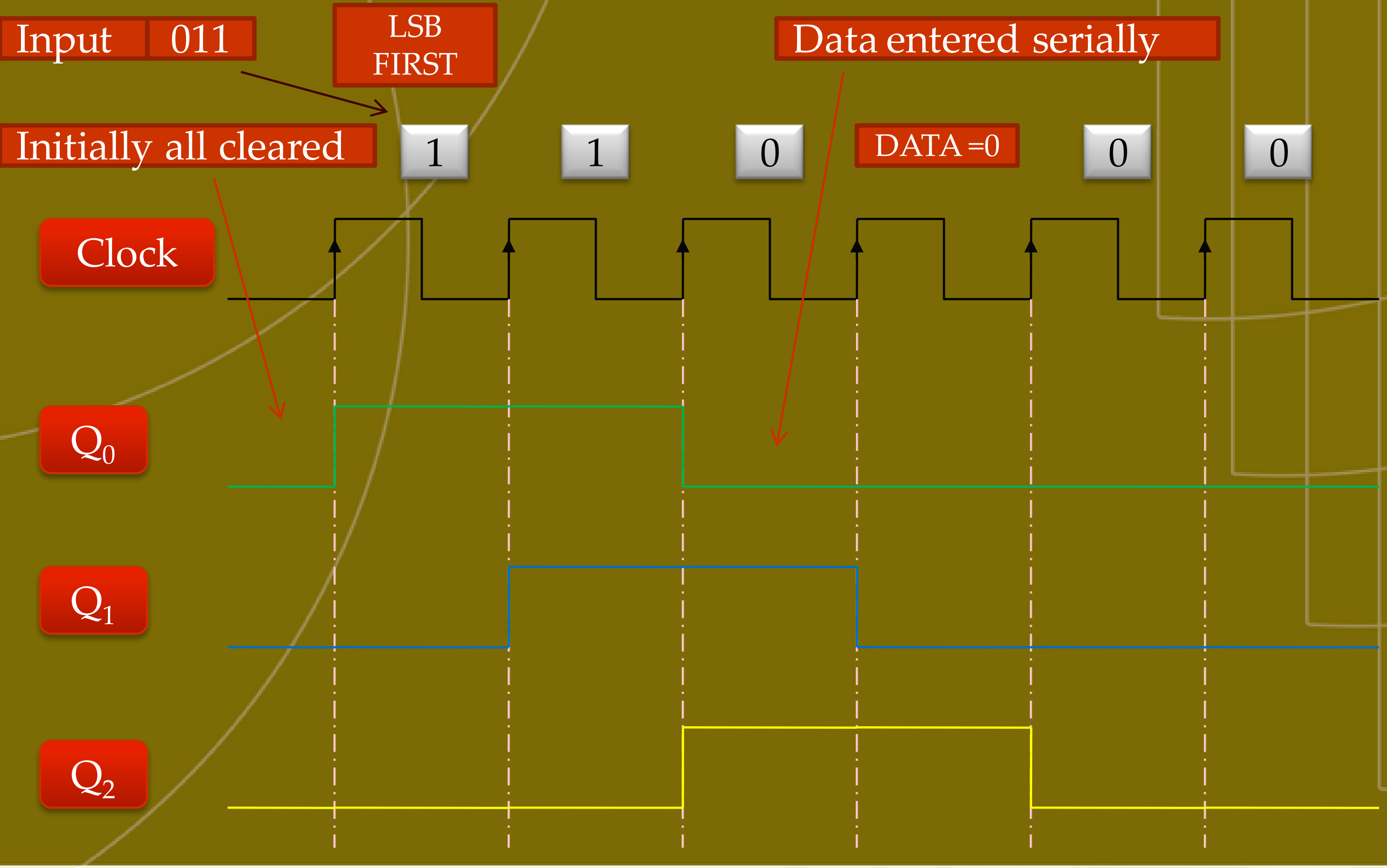
FOR DATA INPUT 011

- The register is first cleared, forcing all three outputs to zero.
- The input data is then applied sequentially to the D input of the first flip-flop on the left (FF0).
- During each clock pulse, one bit is transmitted from left to right.
- The least significant bit of the data has to be shifted through the register from FF0 to FF2.



CLOCK	Q_0	Q_1	Q_2
0	0	0	0
1	1	0	0
2	1	1	0
3	0	1	1
4	0	0	1
5	0	0	0
6	0	0	0

TIMING DIAGRAM (SISO)



Input 011

LSB FIRST

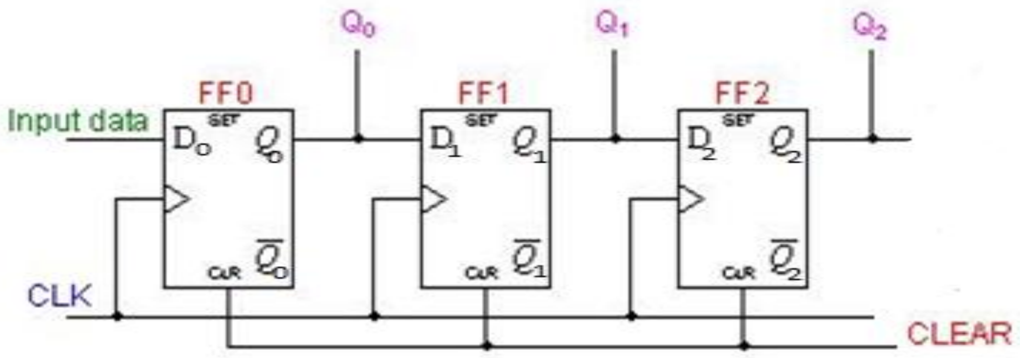
Data entered serially

Initially all cleared

DATA =0

SERIAL IN PARALLEL OUT (SIPO)

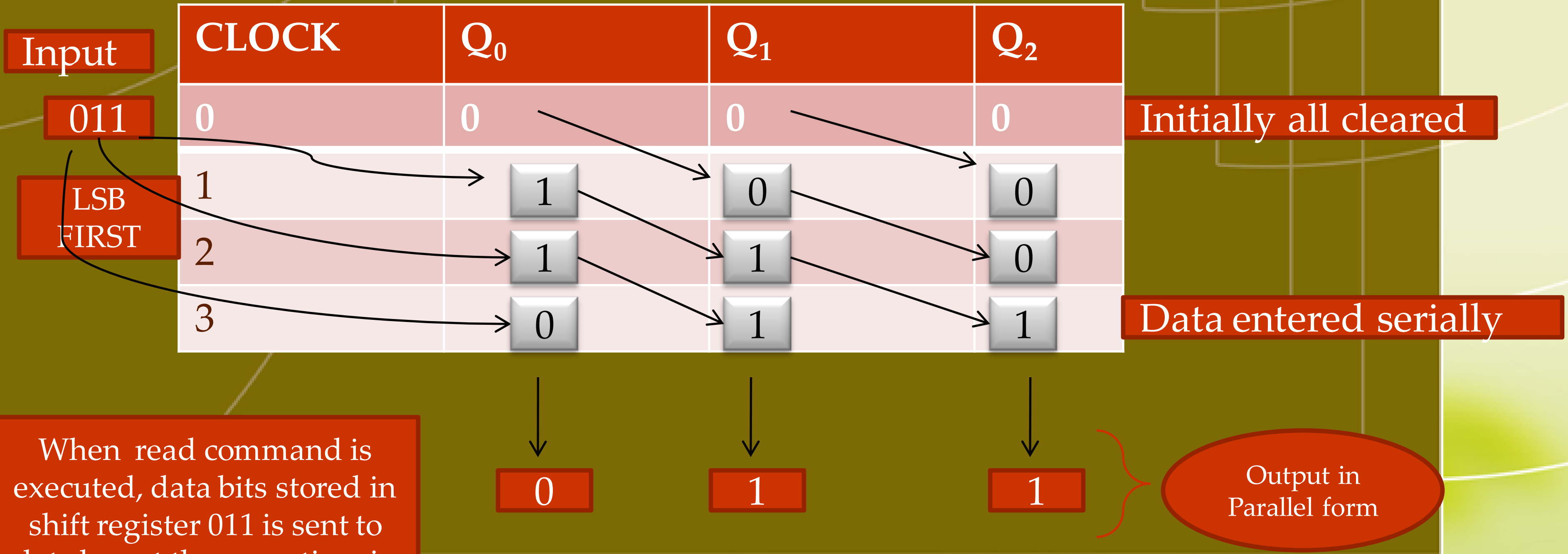
- Data entered into a register serially one bit at a time
- Output is obtained in parallel form
- On the leading edge of the first clock pulse, the signal on the INPUT is latched in the first flip-flop.
- On the leading edge of next clock pulse, the content of first flip-flop is stored in the second flip-flop, and the signal, present at the INPUT is stored in the first flip-flop, etc. as explained in previous section.
- The difference is the way in which the data bits are taken out of the register.
- Once the data are stored, each bit appears on its respective output line, and all bits are available simultaneously.
- A 3-bit SIPO shift register can be constructed using 3-D flip-flops
- Clock is provided in synchronous way in positive edge



CLEAR	Q0	Q1	Q2	Q3
1001	0	0	0	0

FOR DATA INPUT 011

- The register is first cleared, forcing all three outputs to zero.
- The input data is then applied sequentially to the D input of the first flip-flop on the left (FF0).
- During each clock pulse, one bit is transmitted from left to right.
- The least significant bit of the data has to be shifted through the register from FF0 to FF2.



Input

011

LSB FIRST

CLOCK	Q_0	Q_1	Q_2
0	0	0	0
1	1	0	0
2	1	1	0
3	0	1	1

Initially all cleared

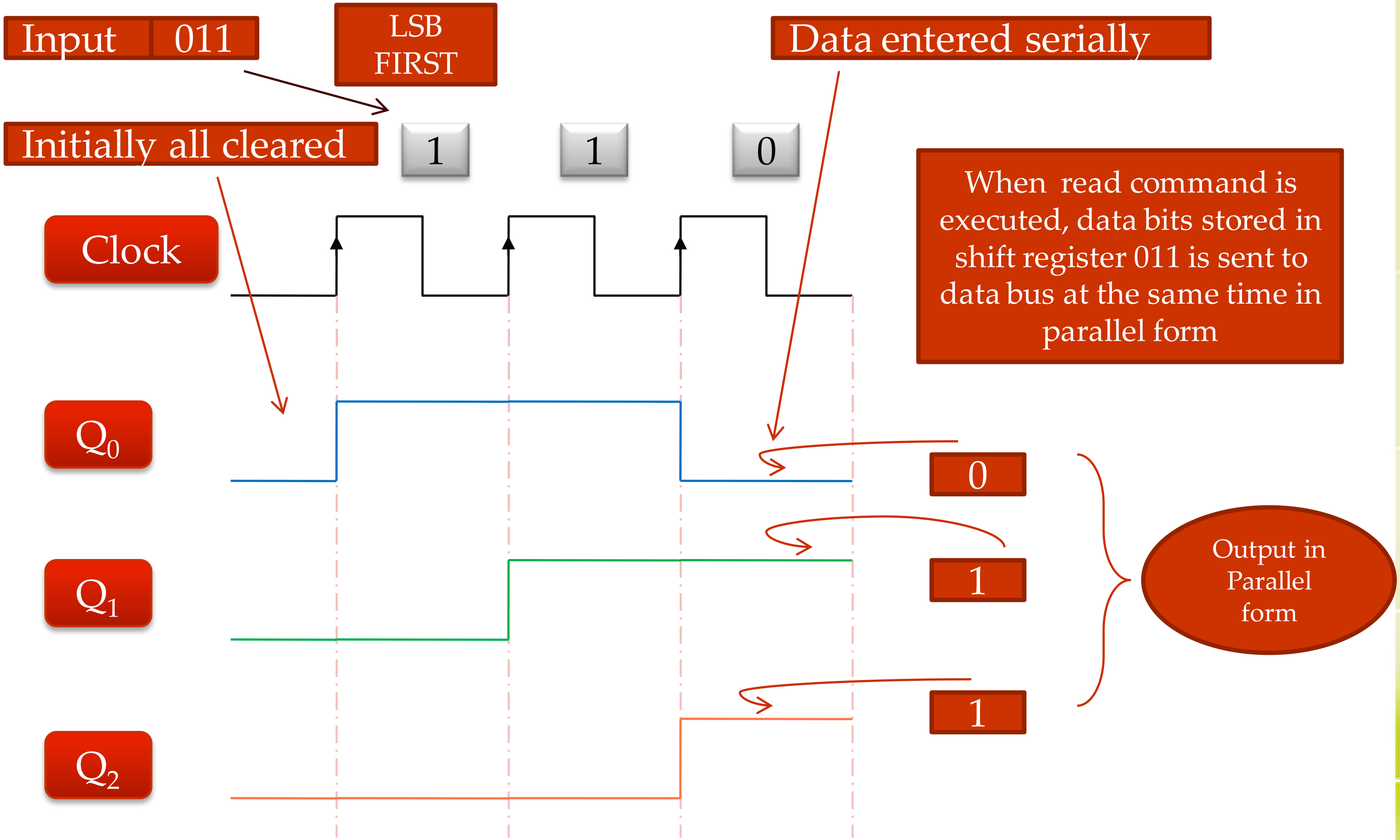
Data entered serially

0 1 1

Output in Parallel form

When read command is executed, data bits stored in shift register 011 is sent to data bus at the same time in parallel form

TIMING DIAGRAM (SIPO)



Input 011

LSB FIRST

Data entered serially

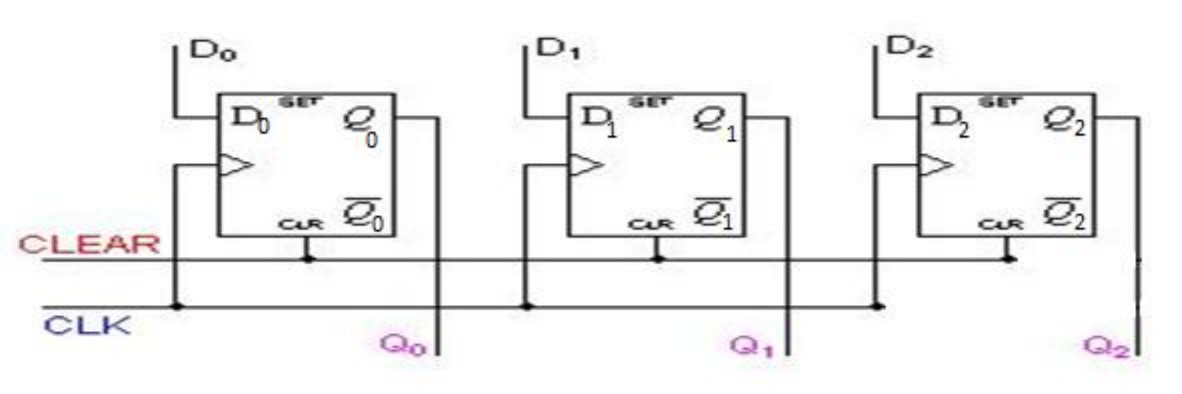
Initially all cleared

When read command is executed, data bits stored in shift register 011 is sent to data bus at the same time in parallel form

Output in Parallel form

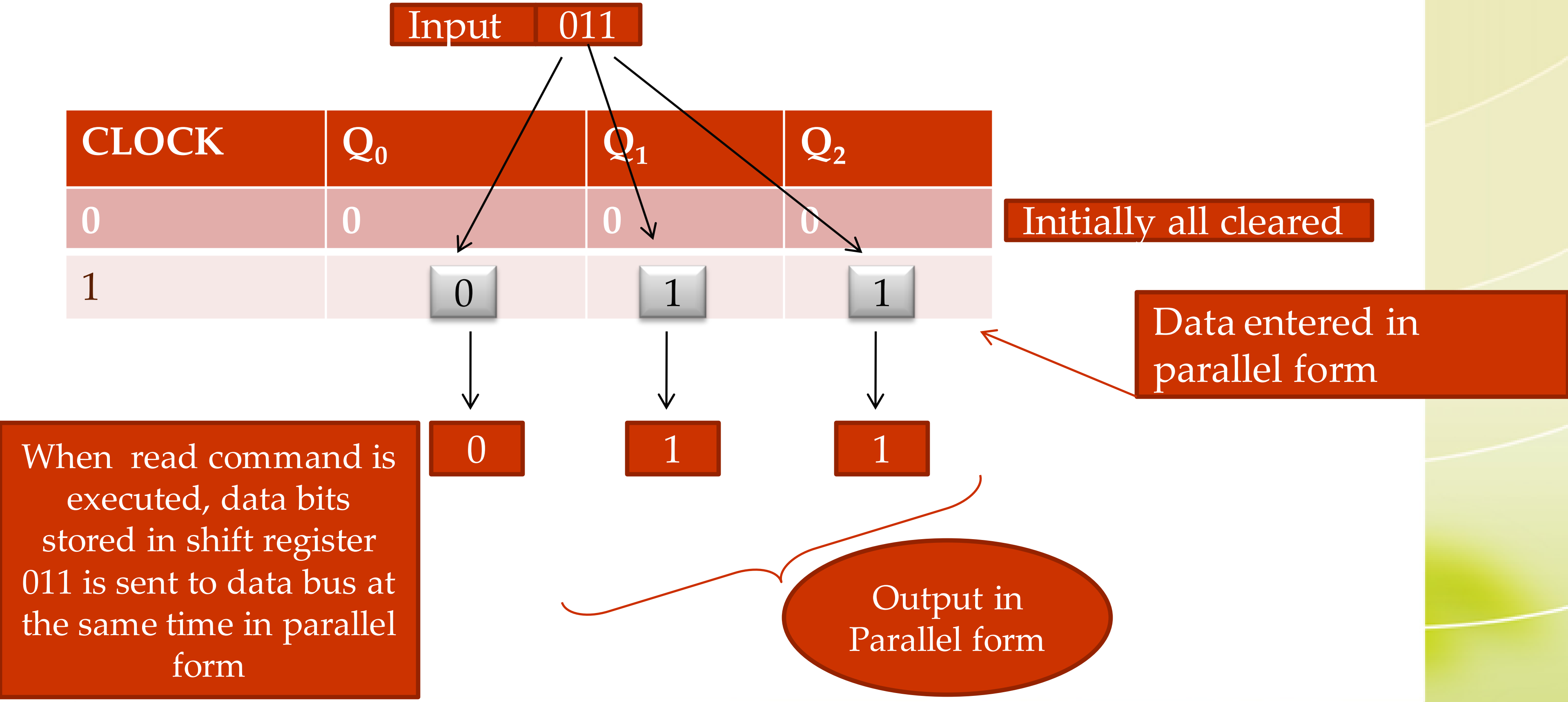
PARALLEL IN PARALLEL OUT (PIPO)

- Data entered into a register simultaneously (all bits at the same time)
- Output is also obtained in parallel form
- Once the register is clocked, all data bits appear on the parallel outputs immediately following the simultaneous entry of the data bits at the D inputs .
- A 3-bit PIPO shift register can be constructed using 3-D flip-flops
- Clock is provided in synchronous way in positive edge
- The D's are the parallel inputs and the Q's are the parallel outputs.



FOR DATA INPUT 011

- The register is first cleared, forcing all three outputs to zero.
- Once the register is clocked, all the data at the D inputs (011) appear at the corresponding Q outputs simultaneously



Input 011

CLOCK	Q_0	Q_1	Q_2
0	0	0	0
1	0	1	1

Initially all cleared

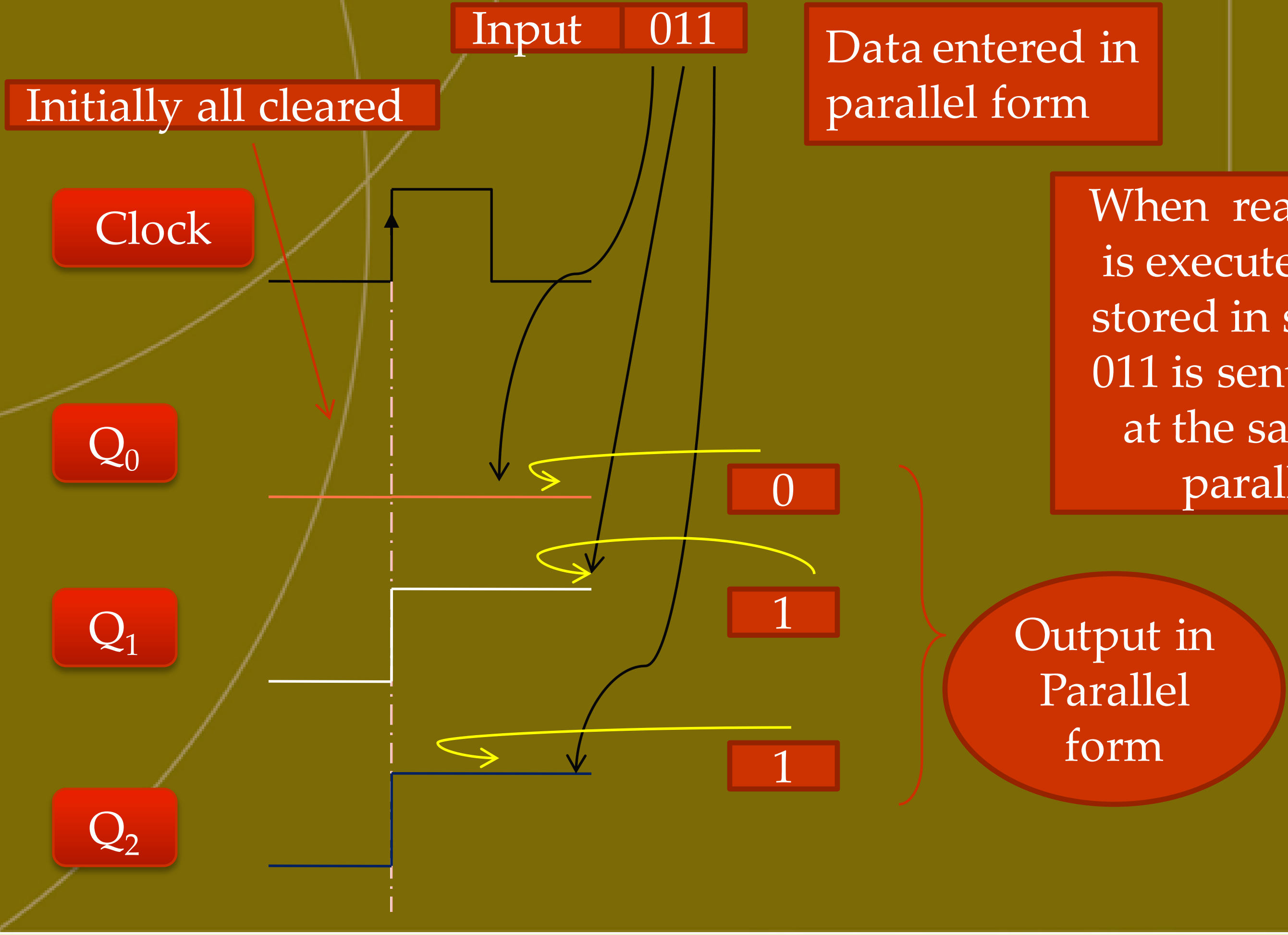
Data entered in parallel form

0 1 1

When read command is executed, data bits stored in shift register 011 is sent to data bus at the same time in parallel form

Output in Parallel form

TIMING DIAGRAM (PIPO)



Input 011

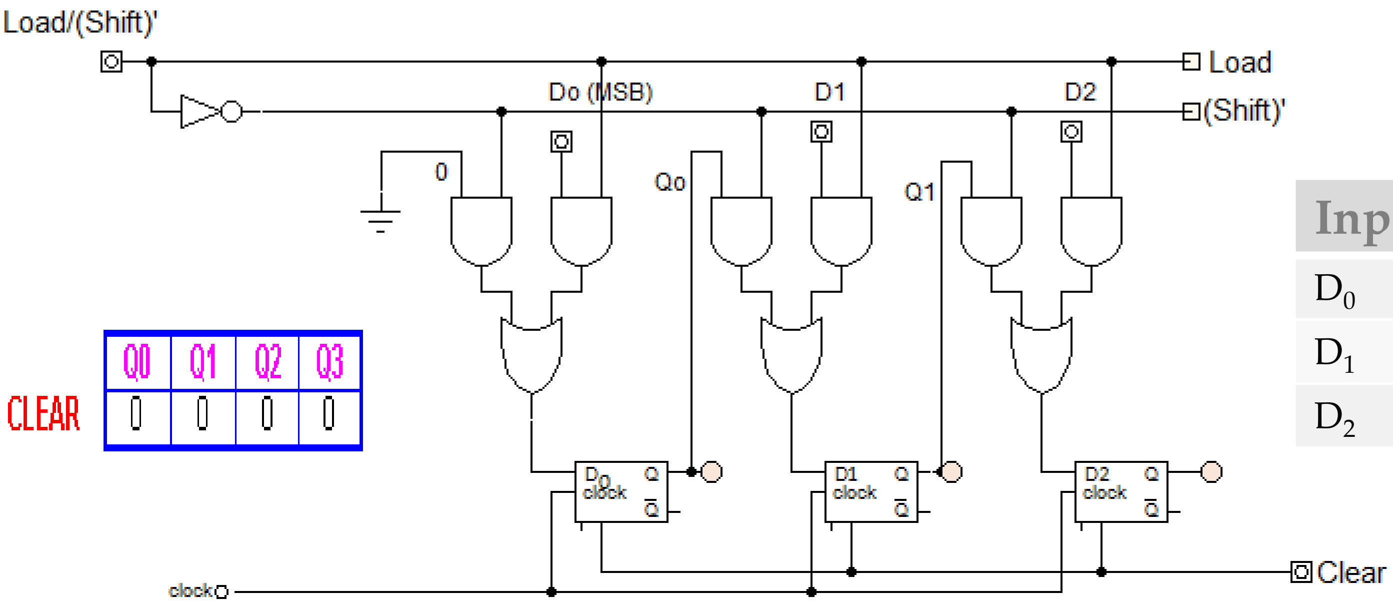
Initially all cleared

Data entered in parallel form

Output in Parallel form

When read command is executed, data bits stored in shift register 011 is sent to data bus at the same time in parallel form

PARALLEL IN SERIAL OUT (PISO)



CLEAR

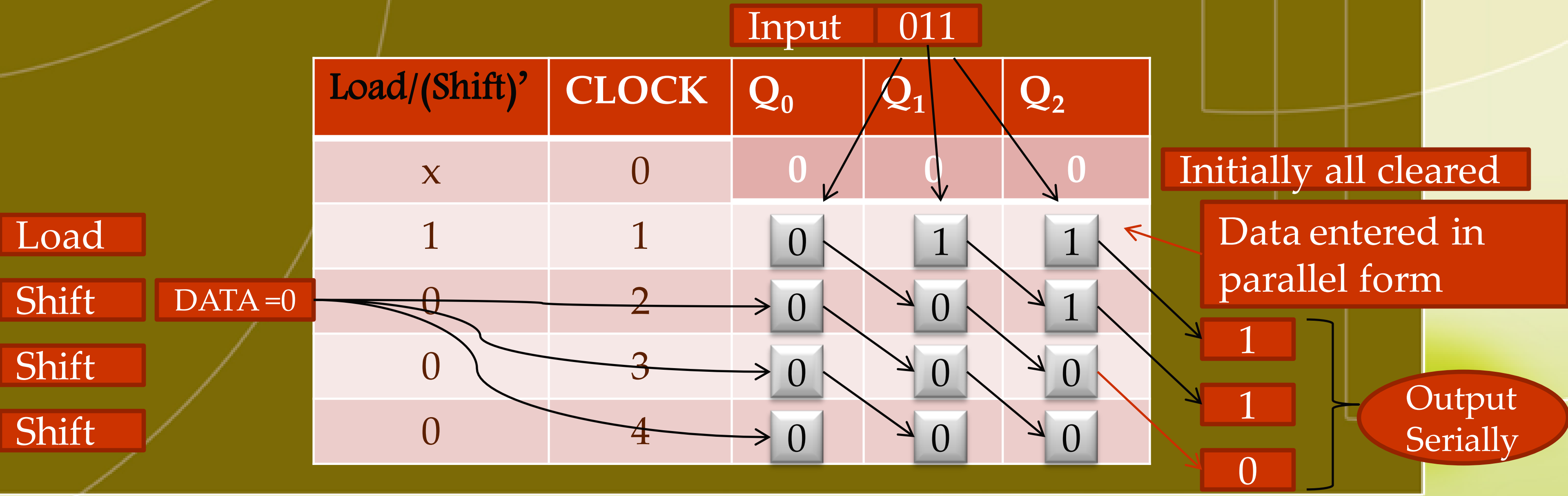
Q0	Q1	Q2	Q3
0	0	0	0

Input	Load	Shift
D_0	D_0 (MSB)	0 (GND)
D_1	D_1	Q_0
D_2	D_2	Q_1

- Data entered into a register simultaneously (all bits at the same time)
- Output is also obtained in serial form
- A 3-bit PISO shift register can be constructed using 3-D flip-flops
- Clock is provided in synchronous way in positive edge
- D0, D1, D2 are the parallel inputs, where D0 is the most significant bit and D2 is the least significant bit.
- To write data in, the mode control line (Load/$\overline{\text{Shift}}$) is taken to HIGH and the data is clocked in.
- The data can be shifted when the mode control line (Load/$\overline{\text{Shift}}$) is LOW as SHIFT is active low. The register performs right shift operation on the application of a clock pulse.

FOR DATA INPUT 011

- The register is first cleared, forcing all three outputs to zero.
- To write data in, the mode control line (Load/(Shift)') is taken to HIGH and the data is clocked in.
- Once the register is clocked, all the data at the D inputs (011) appear at the corresponding Q outputs simultaneously.
- To read data out, the mode control line (Load/(Shift)') is LOW as SHIFT is active low.
- The register performs right shift operation on the application of a clock pulse.
- During each clock pulse, one bit is transmitted from left to right.



Input 011

Load/(Shift)'	CLOCK	Q_0	Q_1	Q_2
x	0	0	0	0
1	1	0	1	1
0	2	0	0	1
0	3	0	0	0
0	4	0	0	0

Initially all cleared

Data entered in parallel form

Load

Shift

Shift

Shift

DATA =0

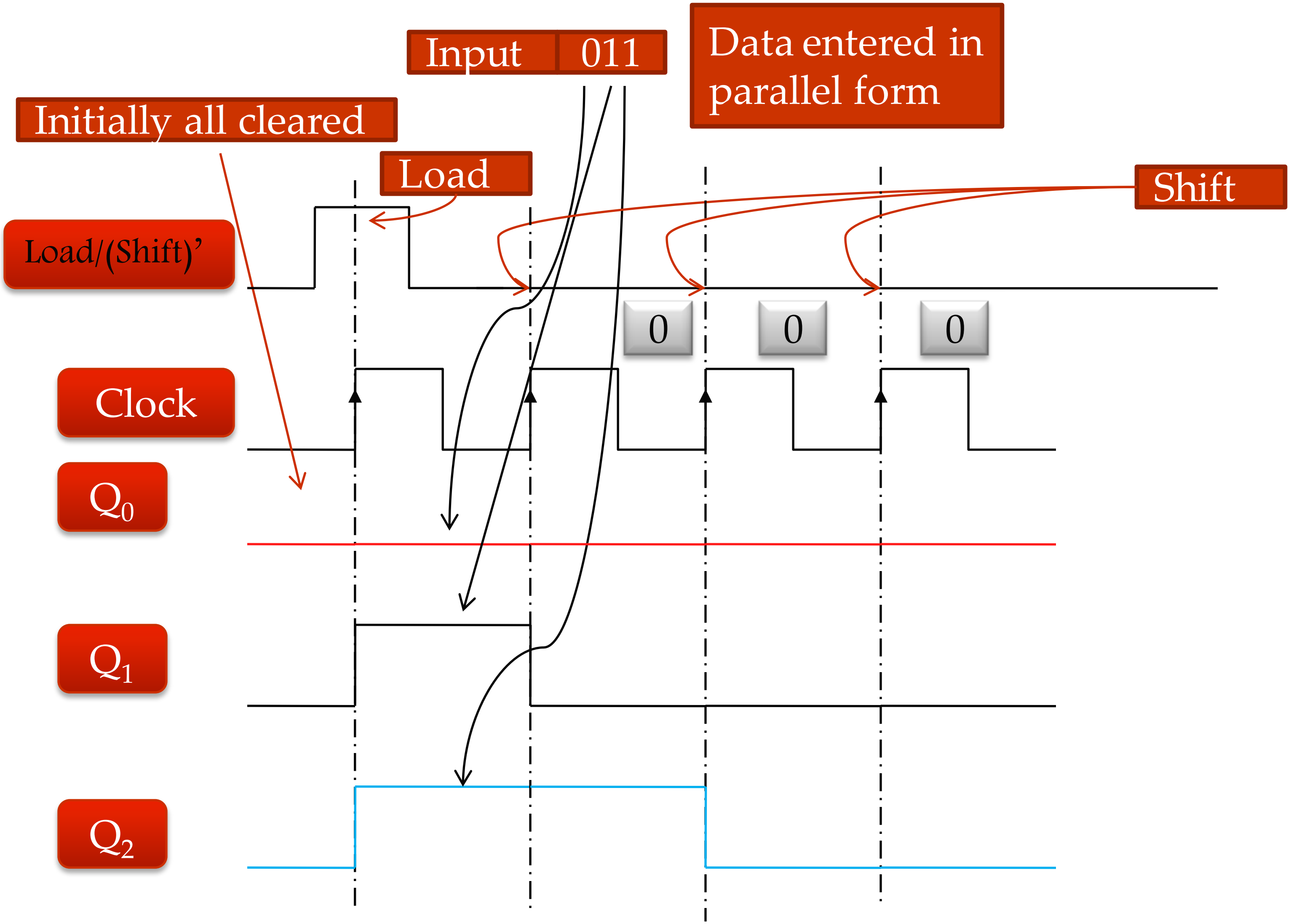
1

1

0

Output Serially

TIMING DIAGRAM (PISO)



Input 011

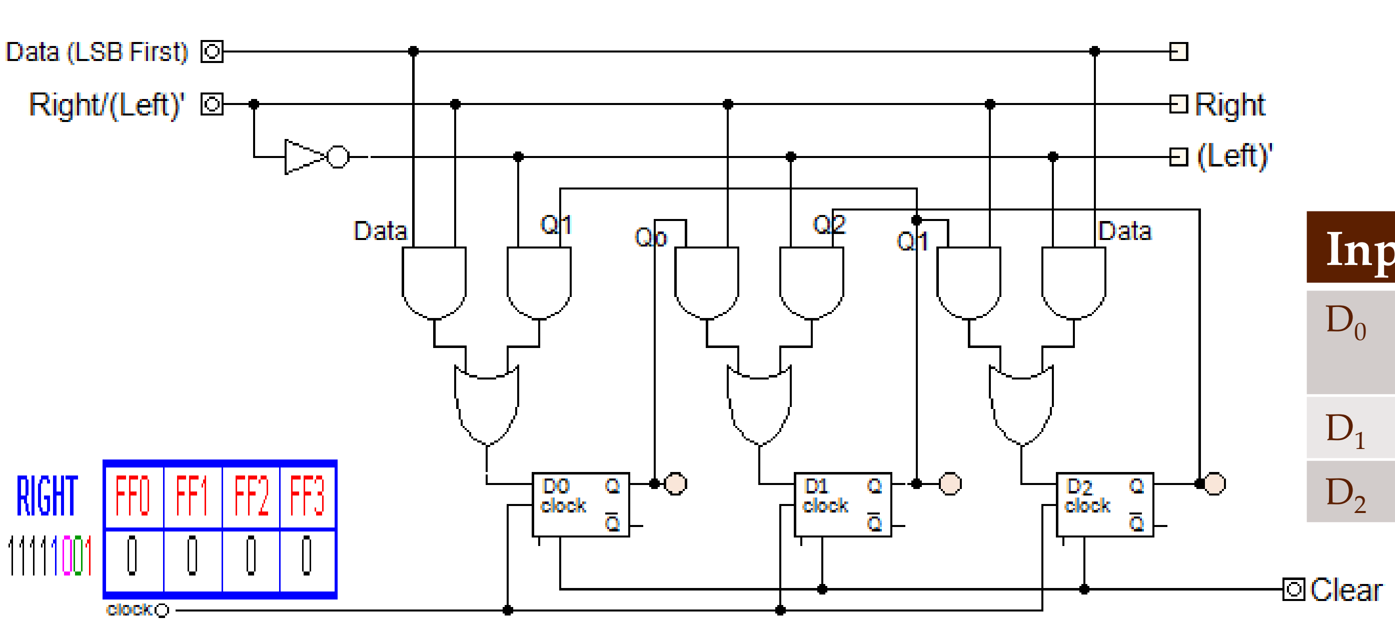
Data entered in parallel form

Initially all cleared

Load

Shift

BIDIRECTIONAL SHIFT REGISTER

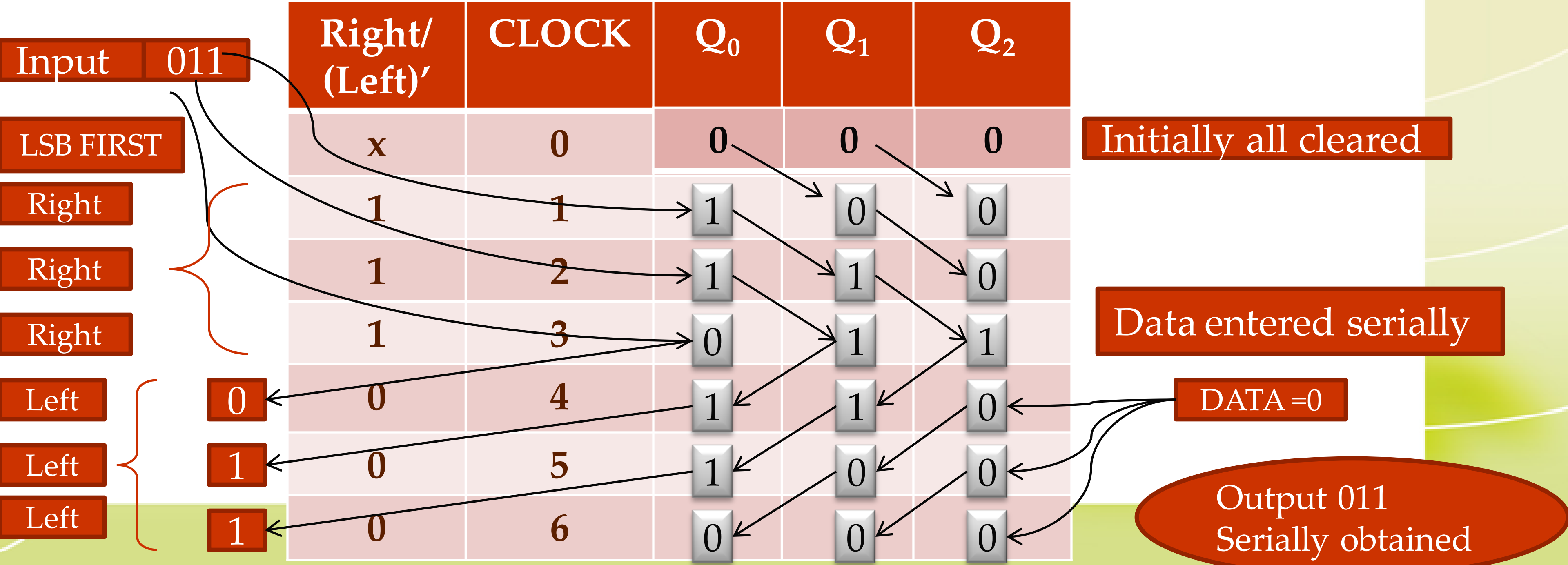


Input	Right	Left
D_0	DATA (LSB FIRST)	Q_1
D_1	Q_0	Q_2
D_2	Q_1	DATA

- A Bidirectional, or reversible, shift register is one in which the data can be shift either left or right
- Data entered into and taken out of a register serially one bit at a time.
- A 3-bit Bidirectional shift register can be constructed using 3-D flip-flops
- Clock is provided in synchronous way in positive edge
- To shift right, the mode control line (Right/(Left)') is taken to HIGH and the data is clocked in.
- To shift left, the mode control line (Right/(Left)') is LOW as LEFT is active low. The register performs left shift operation on the application of a clock pulse.

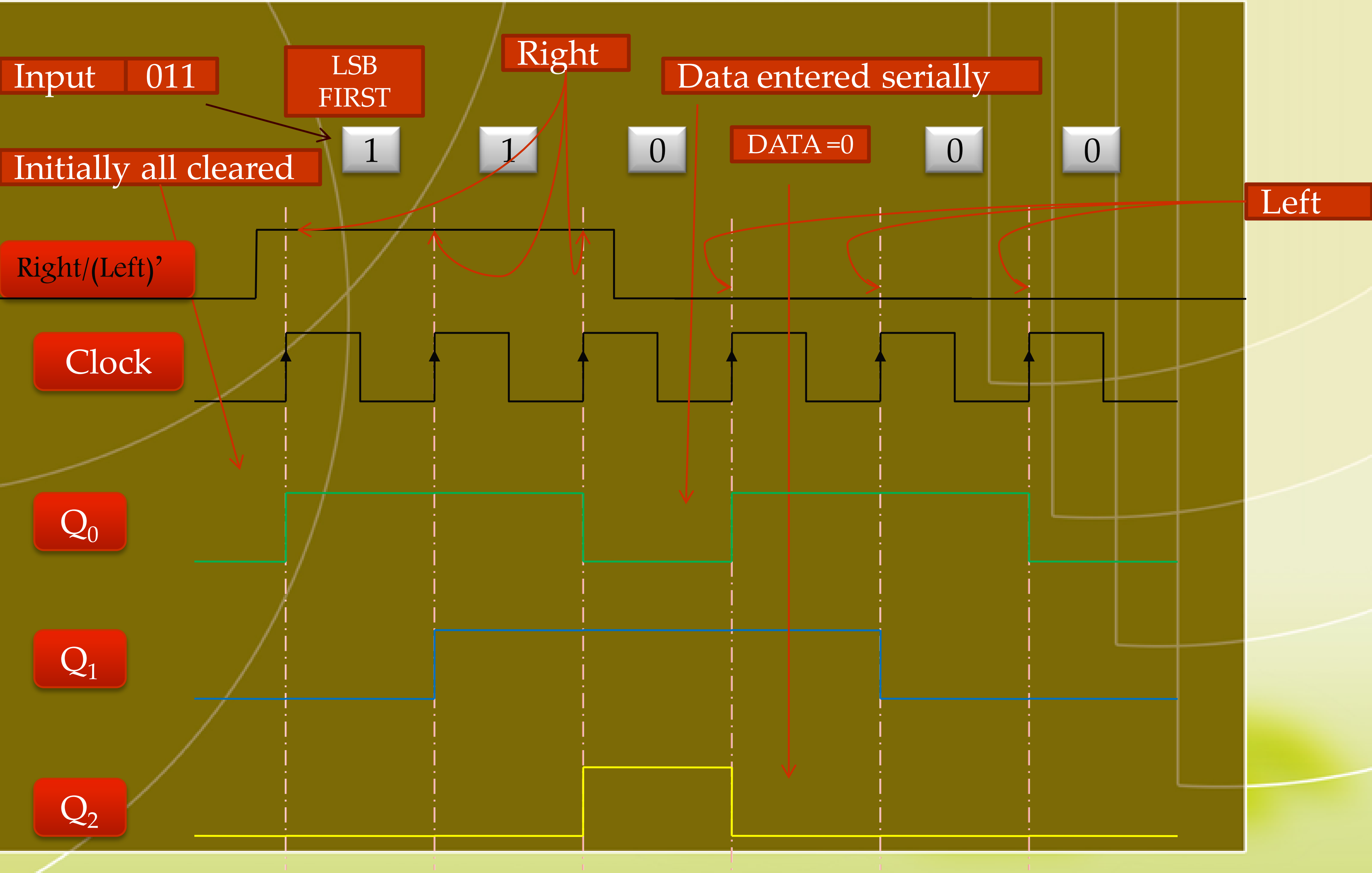
FOR DATA INPUT 011

- The register is first cleared, forcing all three outputs to zero.
- To shift right, the mode control line (Right/(Left)') is taken to HIGH and the data is clocked in.
- During each clock pulse, one bit is transmitted from left to right serially.
- The least significant bit of the data has to be shifted through the register from First flip-flop to Last.
- To shift left, the mode control line (Right/(Left)') is LOW as LEFT is active low.
- The register performs left shift operation on the application of a clock pulse.
- During each clock pulse, one bit is transmitted from right to left serially



Right/ (Left)′	CLOCK	Q_0	Q_1	Q_2
x	0	0	0	0
1	1	1	0	0
1	2	1	1	0
1	3	0	1	1
0	4	1	1	0
0	5	1	0	0
0	6	0	0	0

TIMING DIAGRAM (BIDIRECTIONAL)



Input 011

LSB FIRST

Right

Data entered serially

Initially all cleared

1

1

0

DATA =0

0

0

Left

Right/(Left)’

Clock

Q_0

Q_1

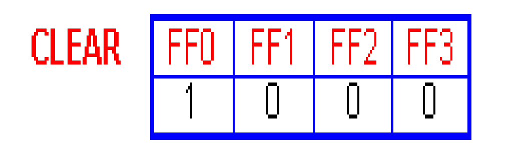
Q_2

SHIFT REGISTER COUNTERS

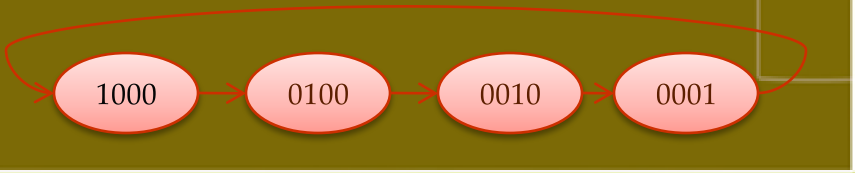
- Two of the most common types of shift register counters are
  - Ring counter and
  - Johnson counter.
- They are basically shift registers with the serial outputs connected back to the serial inputs in order to produce particular sequences.
- These registers are classified as counters because they exhibit a specified sequence of states.

RING COUNTER

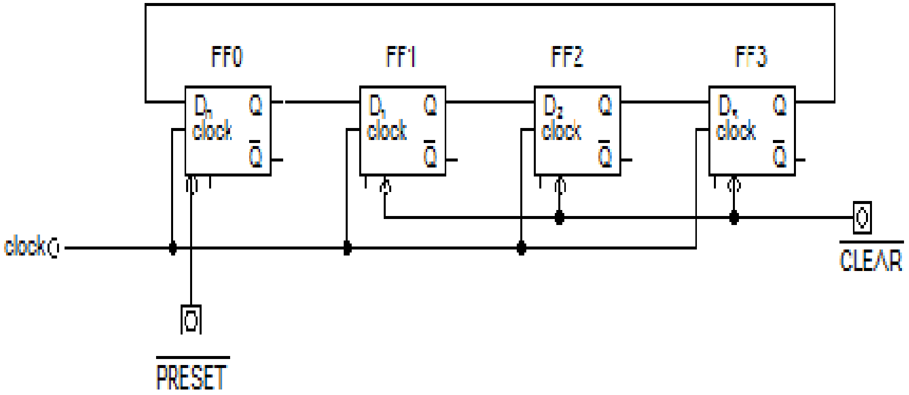
- Basically a circulating shift register (data moved in a ring)
- The output of the most significant stage (Last FF) is fed back to the input of the least significant stage(1st FF).
- Initially before clock pulse, , all the flip-flops except the first one FF0 are reset to 0 using clear and 1st FF(FF0) is preset to 1 instead.
- 4-bit Ring counter constructed from 4 D flip-flops.
- Clock is provided in synchronous way in positive edge
- The output of each stage is shifted into the next stage on the positive edge of a clock pulse serially.
- REMEMBER: N-Bit Ring Counter-------N Flip-Flops-----N states
- Since the count sequence has 4 distinct states, the counter can be considered as a mod-4 counter.
- Only 4 of the maximum 16 states are used, making ring counters very inefficient in terms of state usage.
- But the major advantage of a ring counter over a binary counter is that it is self-decoding. No extra decoding circuit is needed to determine what state the counter is in.



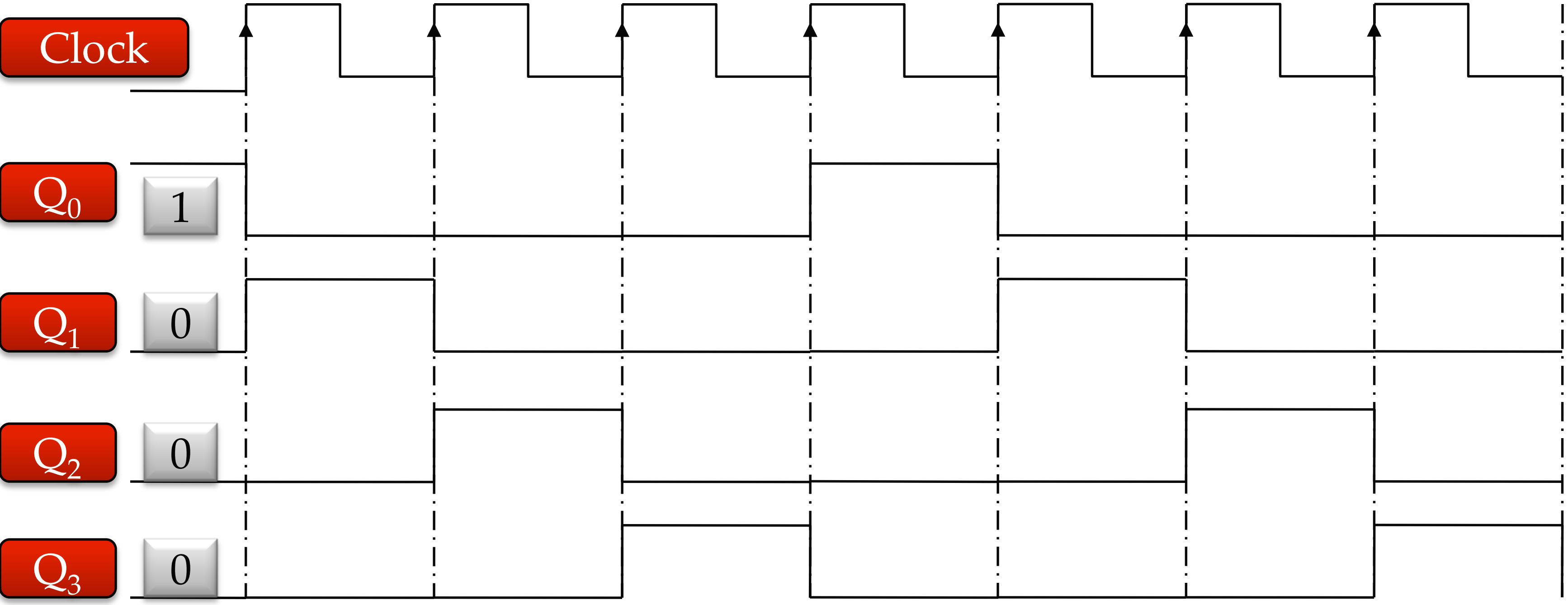
CLEAR	FF0	FF1	FF2	FF3
	1	0	0	0



RING COUNTER



Clk	Q_0	Q_1	Q_2	Q_3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1

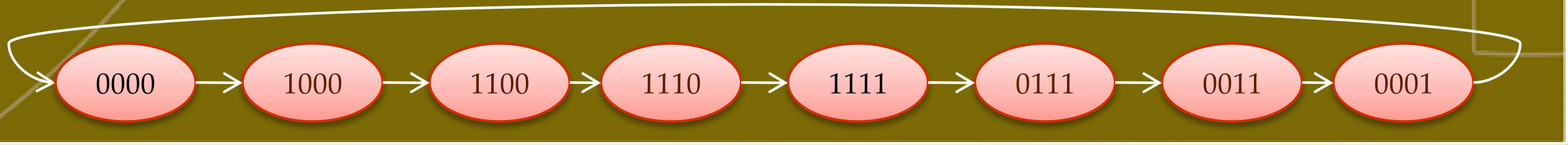


JOHNSON COUNTER

CLEAR

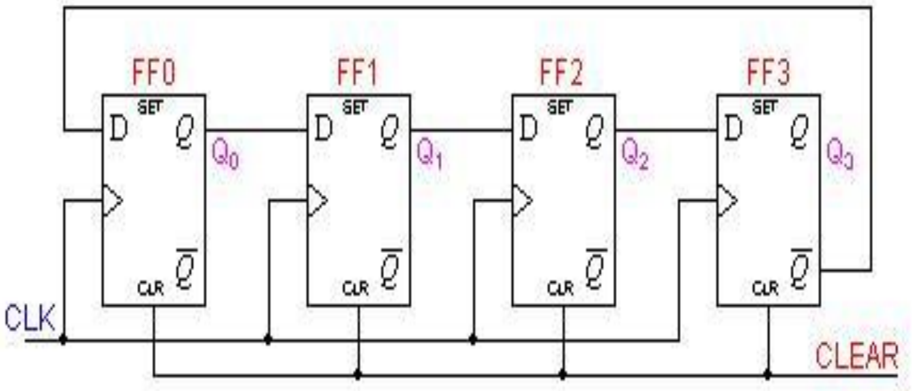
FF0	FF1	FF2	FF3
0	0	0	0

- a circulating shift register (data moved in a ring)
- A variation of standard ring counters, with the inverted output of the last stage fed back to the input of the first stage.
- They are also known as twisted ring counters.
- An n-stage Johnson counter yields a count sequence of length 2n, so it may be considered to be a mod-2n counter
- Disadvantage is that the maximum available states are not fully utilized. Only eight of the sixteen states are being used for 4 bit Johnson counter
- Both Ring and Johnson counter must initially be forced into a valid state in the count sequence because they operate on a subset of the available number of states. Otherwise, the ideal sequence will not be followed
- Initially before clock pulse, , all the flip-flops are reset to 0 using clear
- 4-bit Johnson counter constructed from 4 D flip-flops.
- Clock is provided in synchronous way in positive edge
- The output of each stage is shifted into the next stage on the positive edge of a clock pulse serially.
- REMEMBER: N-Bit Johnson Counter--------N Flip-Flops------2N states

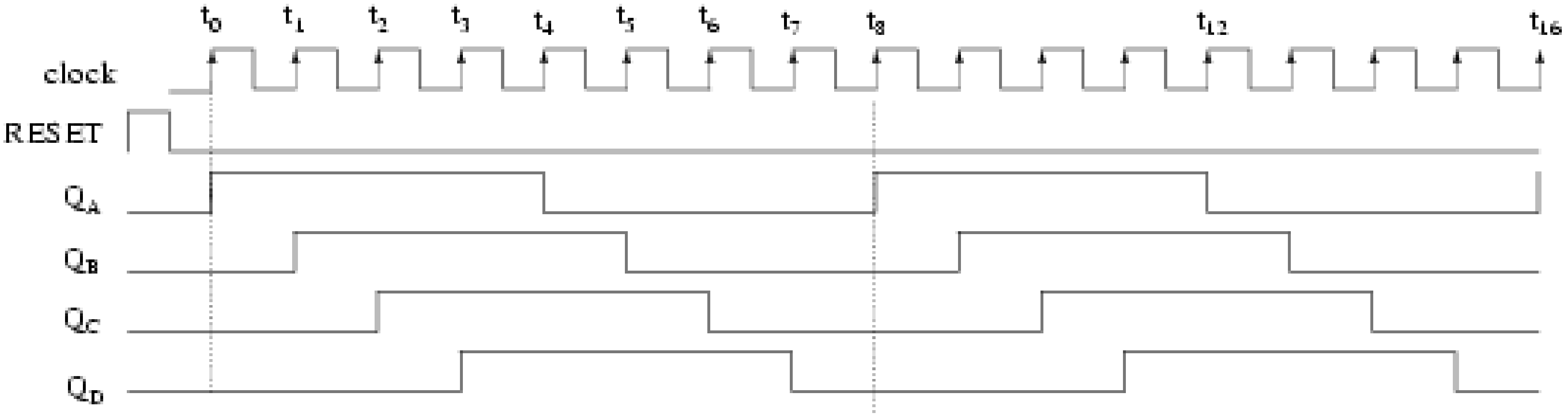


0000 → 1000 → 1100 → 1110 → 1111 → 0111 → 0011 → 0001

JOHNSON COUNTER



Clk	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



Four stage Johnson counter waveforms

APPLICATION OF SHIFT REGISTERS

- ➢ To produce Time Delay
- ✓ The serial in –serial out shift register can be used as a time delay device. The amount of delay can be controlled by
  - ✓ the number of stages in the register
  - ✓ the clock frequency

- ➢ To simplify combinational logic
- ✓ The ring counter technique can be effectively utilized to implement synchronous sequential circuits.
- ✓ A major problem in the realization of sequential circuits is the assignment of binary codes to the internal states of the circuit in order to reduce the complexity of circuits required.
- ✓ By assigning one flip-flop to one internal state, it is possible to simplify the combinational logic required to realize the complete sequential circuit.
- ✓ When the circuit is in a particular state, the flip-flop corresponding to that state is set to HIGH and all other flip-flops remain LOW.

APPLICATION OF SHIFT REGISTERS

- To convert serial data to parallel data
  - ✓ A computer or microprocessor-based system commonly requires incoming data to be in parallel format.
  - ✓ But frequently, these systems must communicate with external devices that send or receive serial data.
  - ✓ So, serial-to-parallel conversion is required. As shown in the previous sections, a serial in - parallel out register can achieve this.
- Parallel to Serial data converter (PISO register can do this)
- Multiplication of binary number by two.
  - ✓ Shift data one bit left to achieve this
- Division of binary number by two.
  - ✓ Shift data one bit right is equivalent to dividing by 2 with an error of ½ in decimal value

Multiplication by 2					
4 bit SISO	0	1	1	0	=6
Left shift with 0	1	1	0	0	=12

Divisible by 2					
4 bit SISO	0	1	1	0	=6
Right shift with 0	0	0	1	1	=3

APPLICATION OF SHIFT REGISTERS

- ➢ Data storage (PIPO)

- ✓ The common clock input triggers all flip-flops on the positive edge of each pulse → the binary data available at the 4 inputs are transferred into the register.

- ✓ The four outputs can be sampled to obtain the binary information stored in the register.

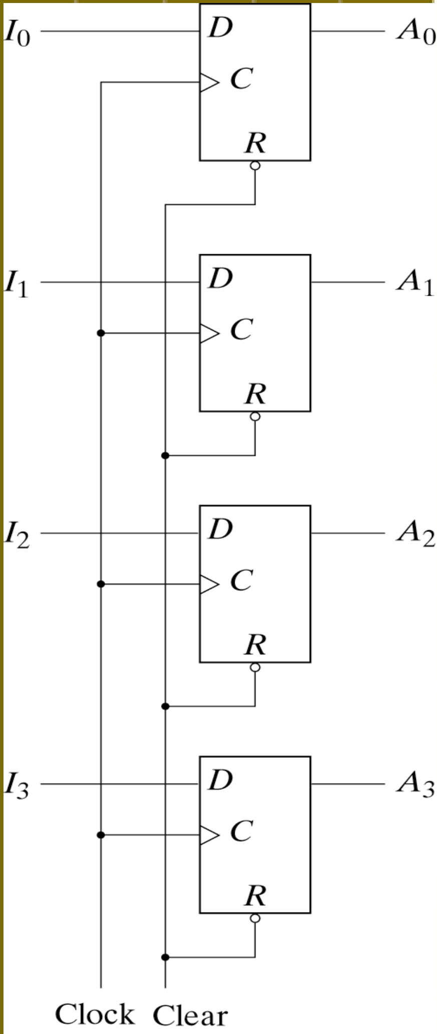


Fig. 6-1 4-Bit Register

QUESTIONS?