CSE 4205
Digital Logic Design

Register

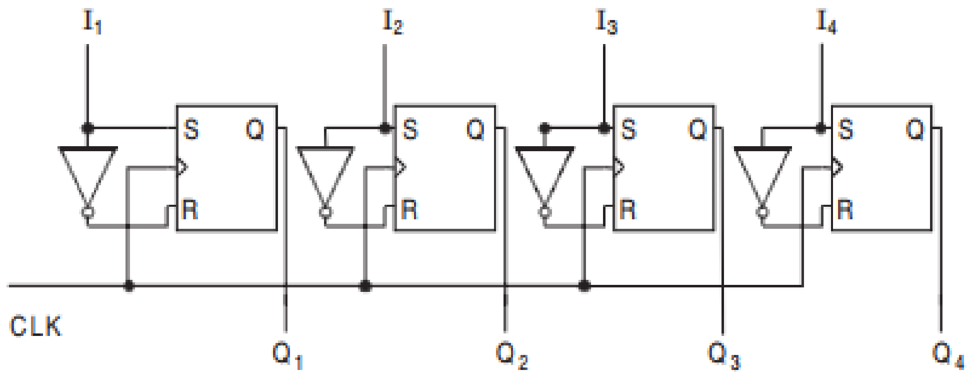
Course Teacher: Md. Hamjajul Ashmafee

AP, CSE, IUT

Email: ashmafee@iut-dhaka.edu

Introduction

- A group of **binary cells** that can hold **binary information**
  - Normally, **a group of flip flops** makes a register
    - ***n* bit register** has ***n* flip flops** to hold ***n*** bits of information
- A register may have **an additional combinational circuit** that performs ***data processing*** task



Shift Register: Classification

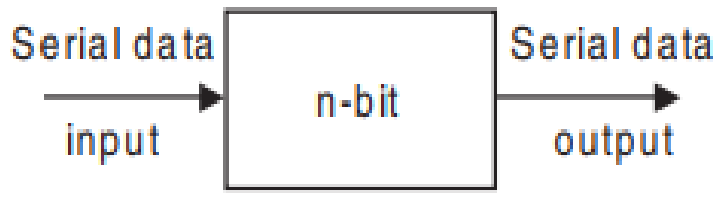
- **Shift:** Operation to move and store data
- **Direction of shifting**
  - It is capable of shifting its binary contents either to the **right** or to the **left**
- **Types of shifting**
  1. **Serial shifting**
  2. **Parallel shifting**

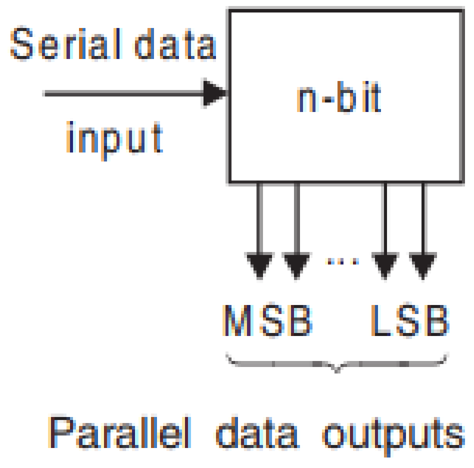
Shift Register: Different Classifications

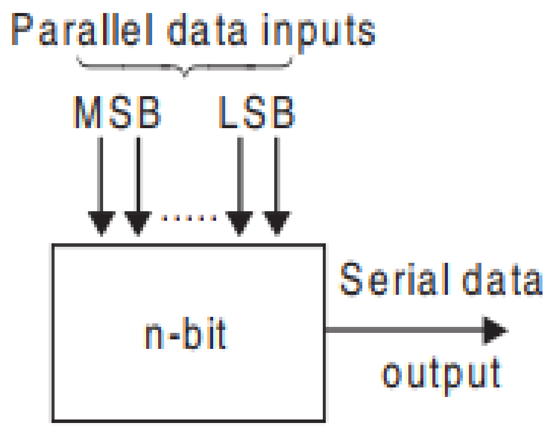
- **Unidirectional Shift Register**
- **Bidirectional Shift Register**
- **Universal Shift Register**

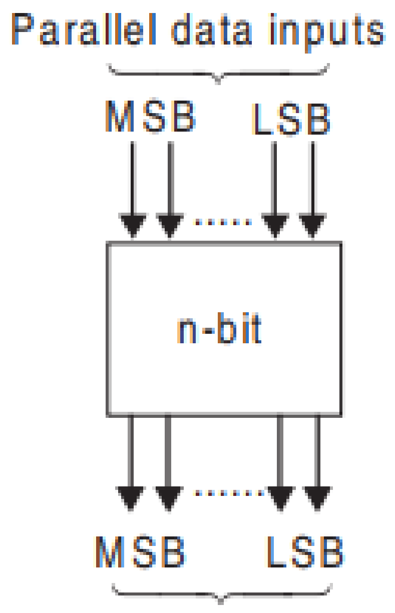
Shift Register: Basic Types

1. SISO (IC 74L91)
2. SIPO (IC 74164)
3. PISO (IC 74265)
4. PIPO (IC 74198)







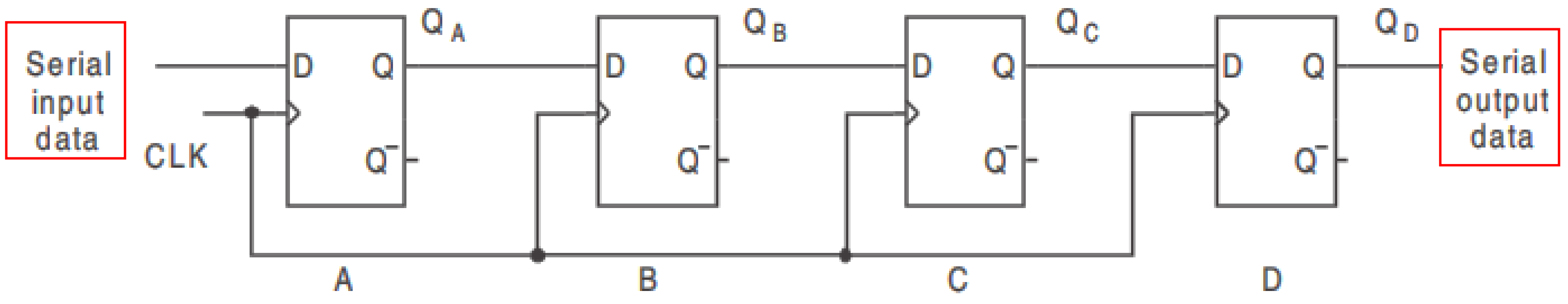


Shift Register - SISO

- **Input data** – serially through <u>a single input line</u>
- **Output data** – also serially through <u>a single output line</u>
- **Types** based on direction of shifting
  - **Left shift register**
  - **Right shift register**

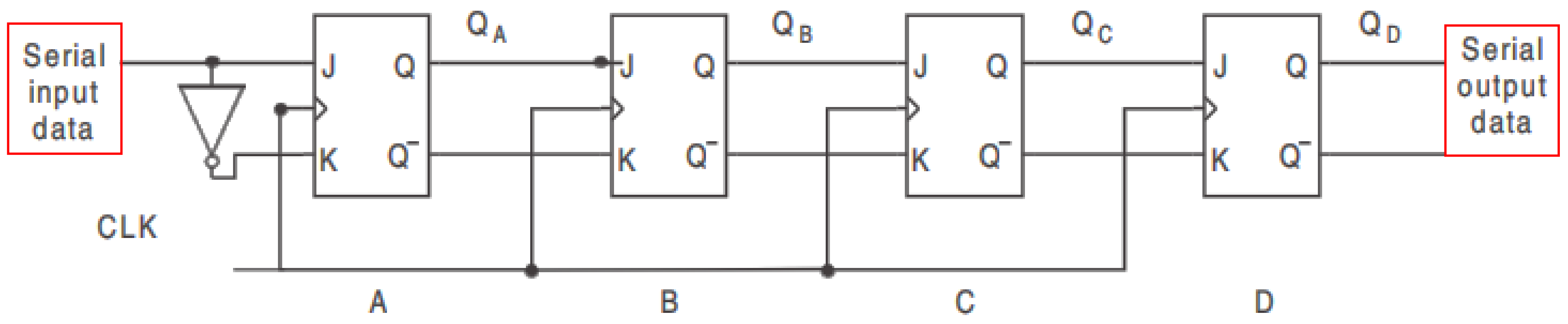
Shift Register – SISO – Right shift

Circuit Diagram



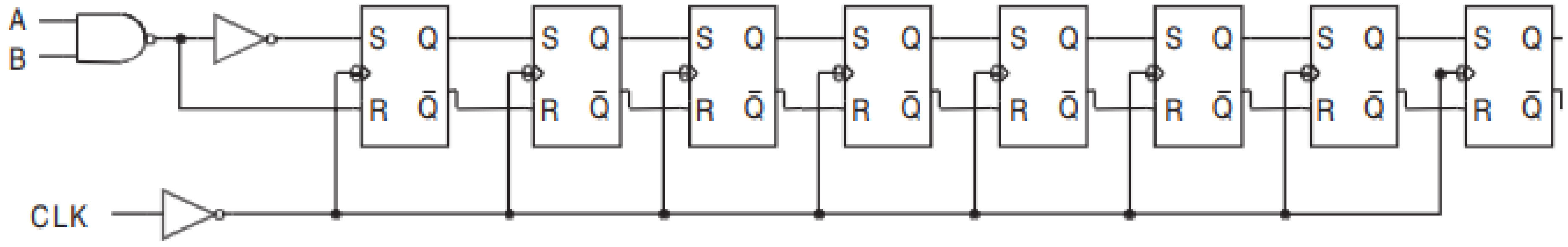
Using D flip flop

MSB **LSB**

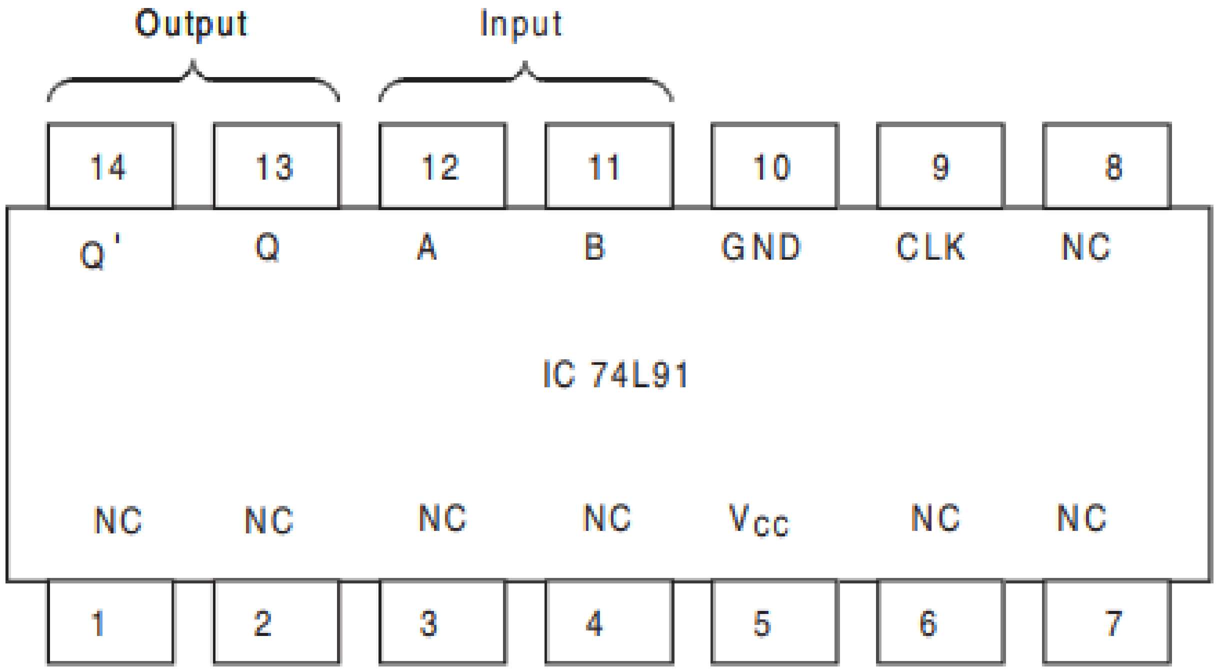


Using JK flip flop

8 Bit SISO Shift Register IC



(a) Logic diagram.

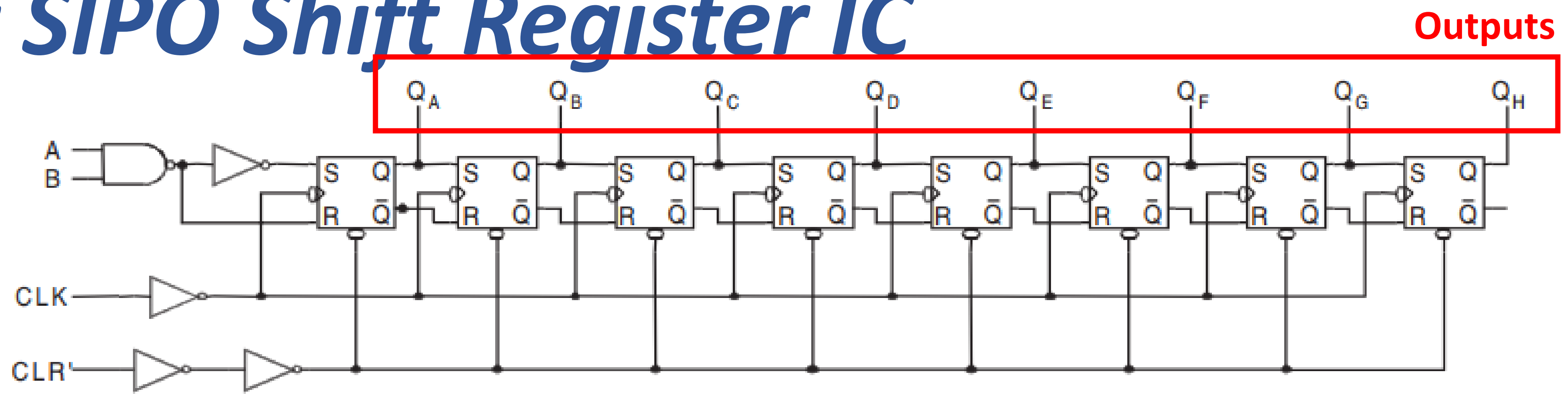


(b) Pinout diagram of IC 74L91.

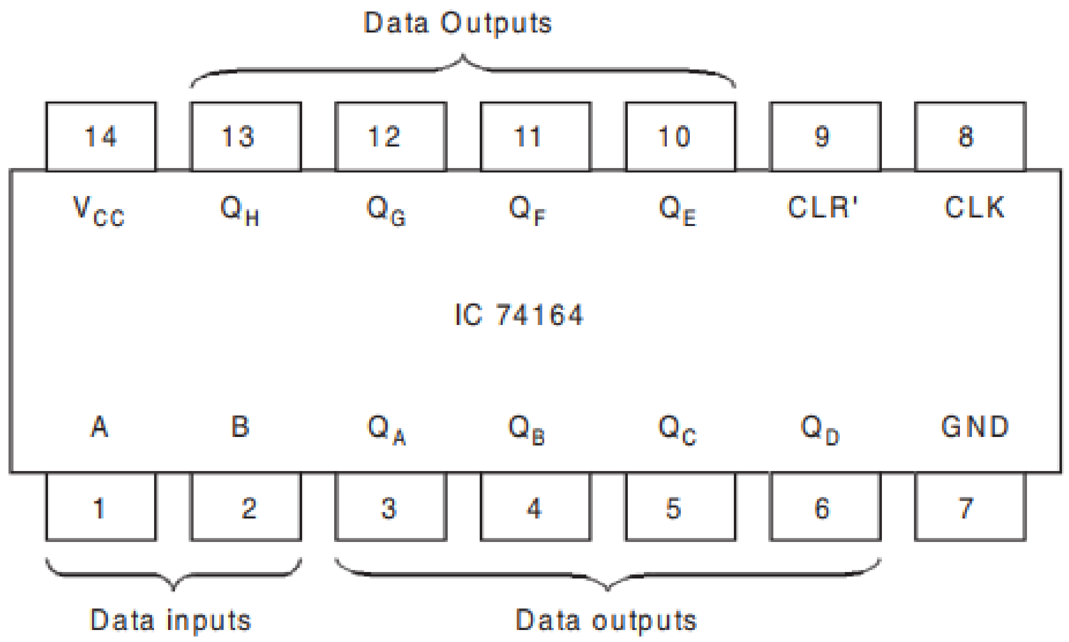
Shift Register – SIPO

- **Property:** Data is **shifted in serially**, but **shifted out in parallel**
  - All the outputs should be available at the same time by connecting the output of each flip flop to an output pin
  - If the data is stored in the flip flops, **instantly** they are available in the output

8 Bit SIPO Shift Register IC



(a) Logic diagram.



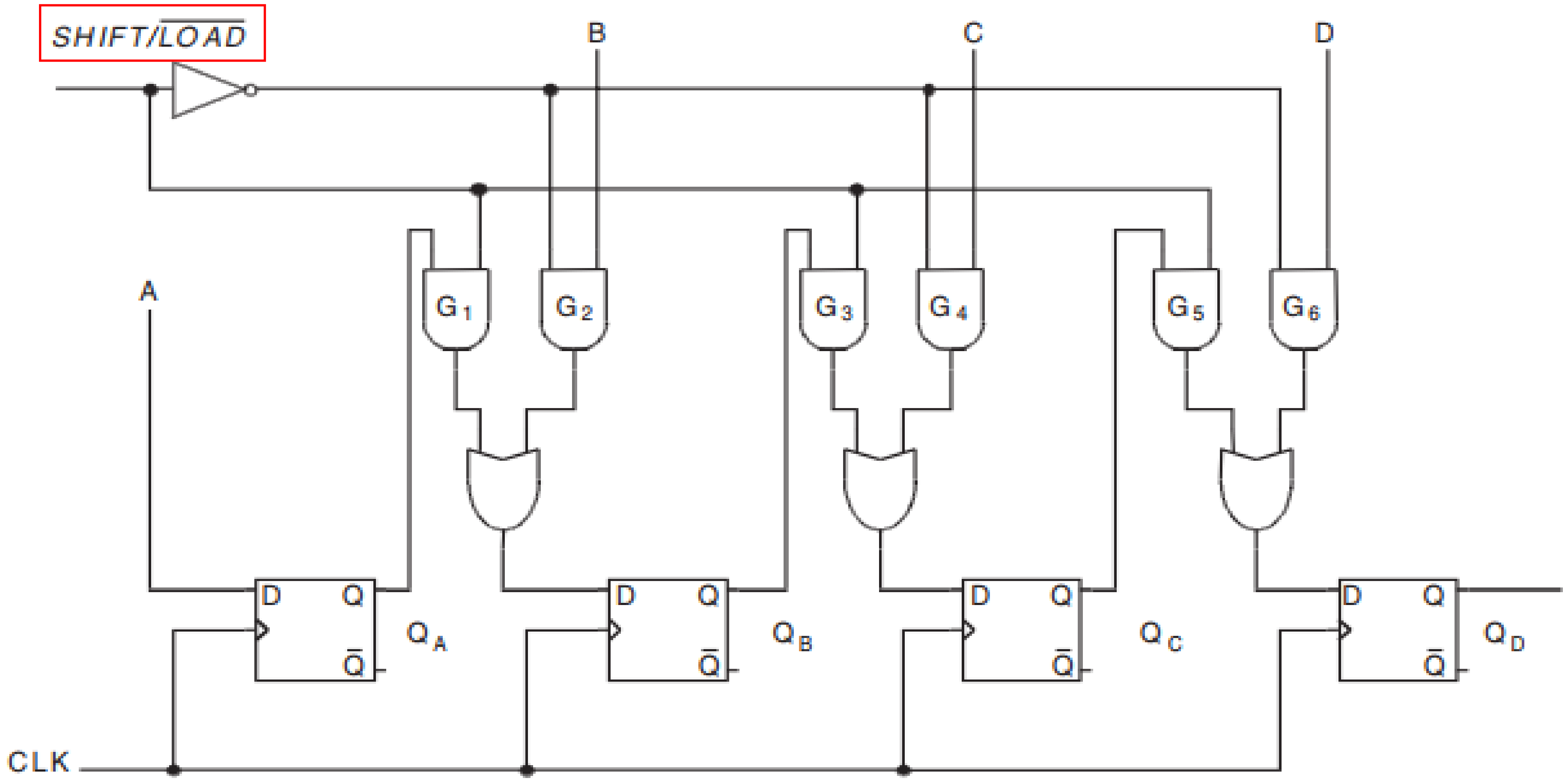
(b) Pinout diagram of IC 74164.

Shift Register – PISO

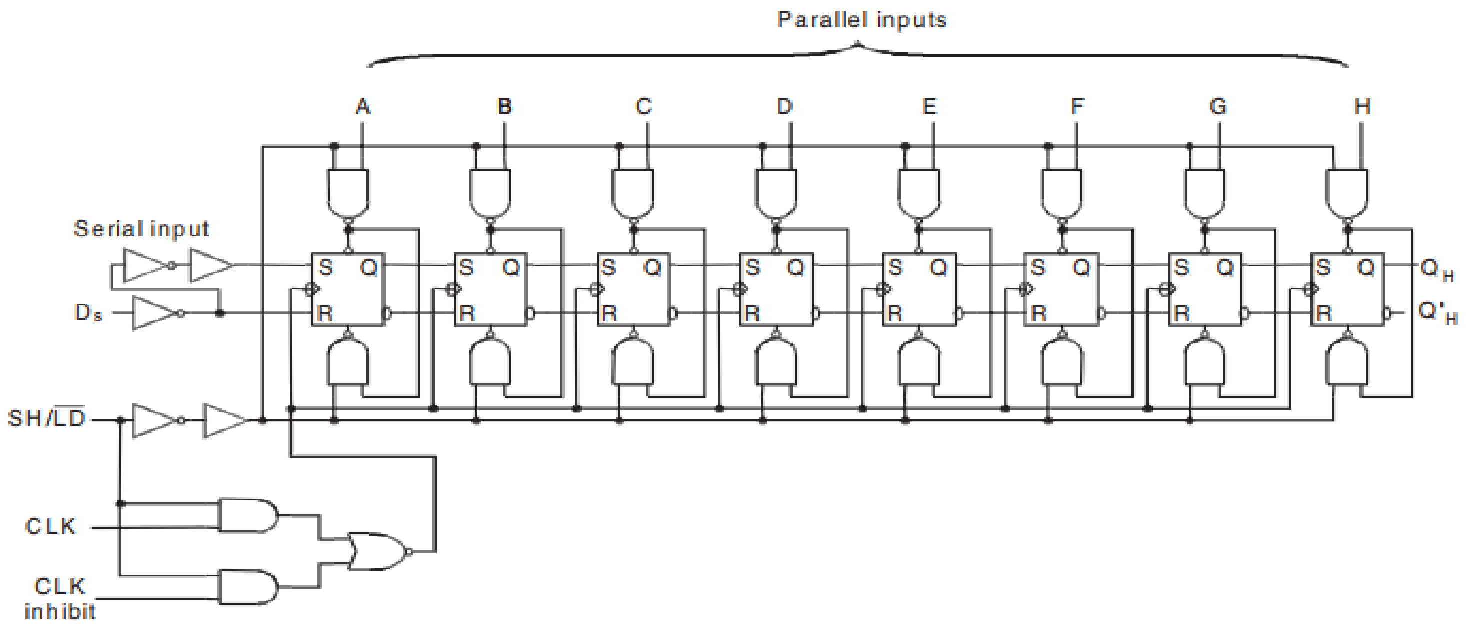
- **Property**: Data is entered **in parallel manner** into the register
- A **control input (Shift/Load)** is required to allow the all-input data bits **either** to **enter into** the register in parallel or to **shift out** the data in serial

Shift Register – PISO…

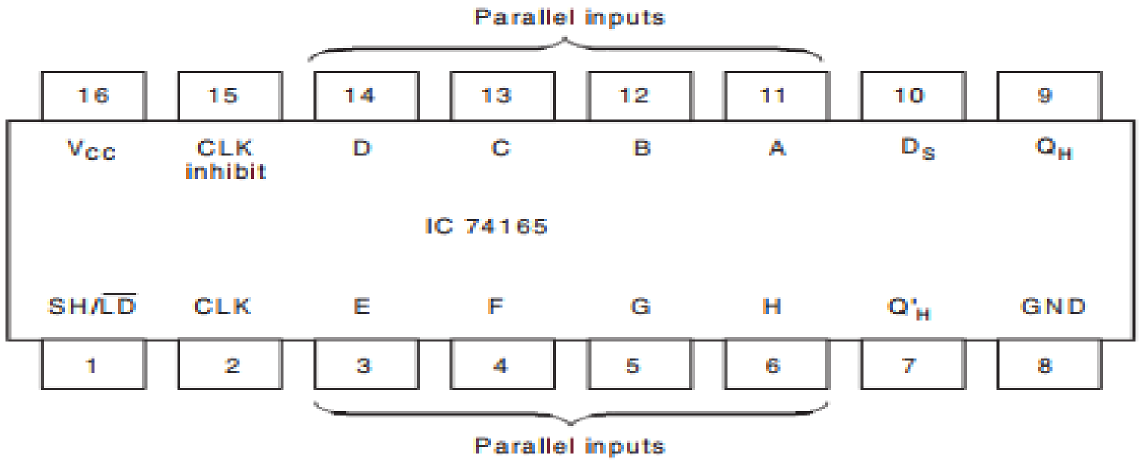
Circuit Diagram



8 Bit PISO Shift Register IC



(a) Logic diagram.



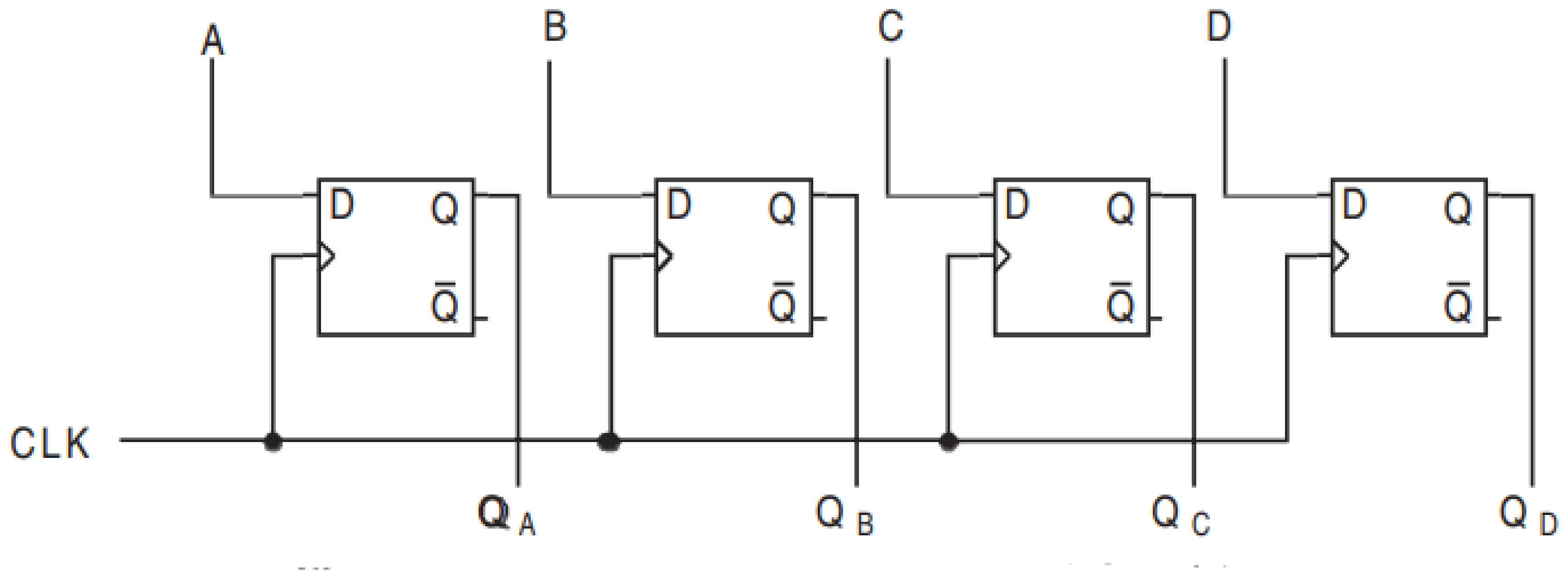
(b) Pinout diagram of IC 74165.

Shift Register – PIPO

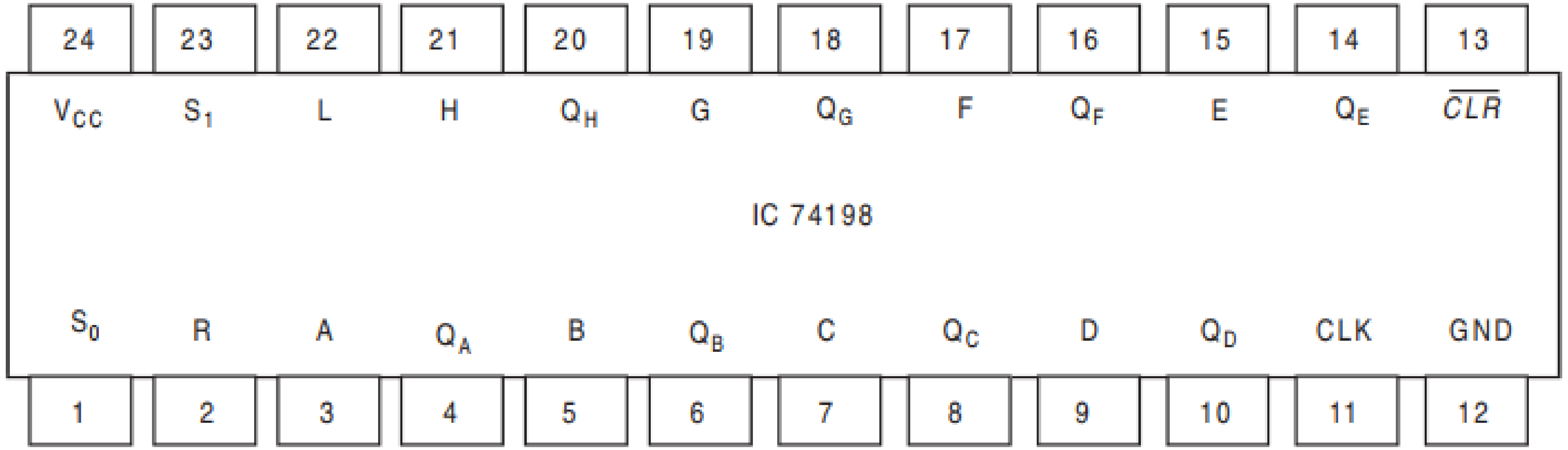
- **Property:** Data can be shifted into or out of the register **in parallel**
- **No interconnection** between the flip flops as no serial shift is required

Shift Register – PIPO…

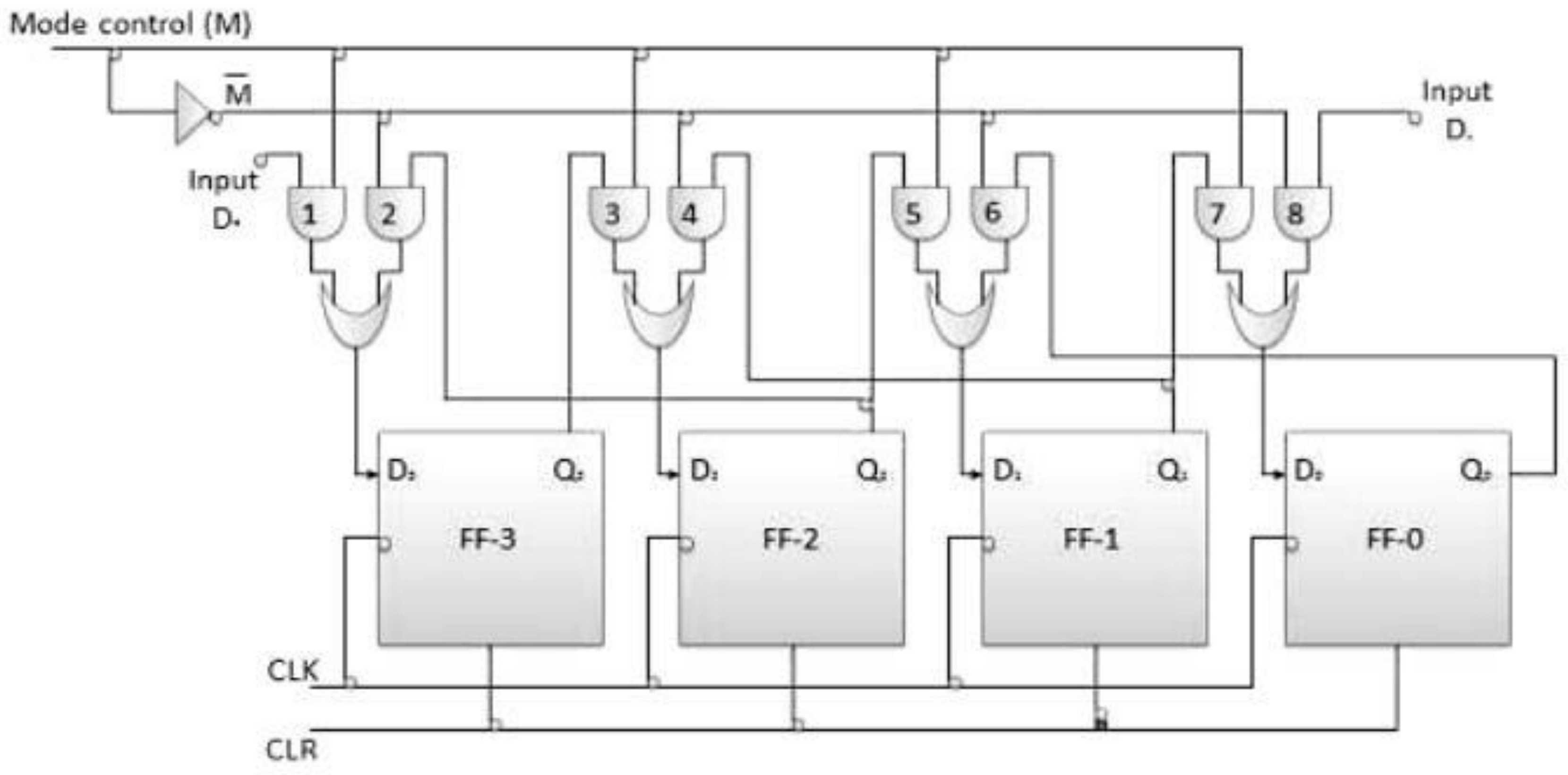
Circuit Diagram



8 Bit PIPO Shift Register IC



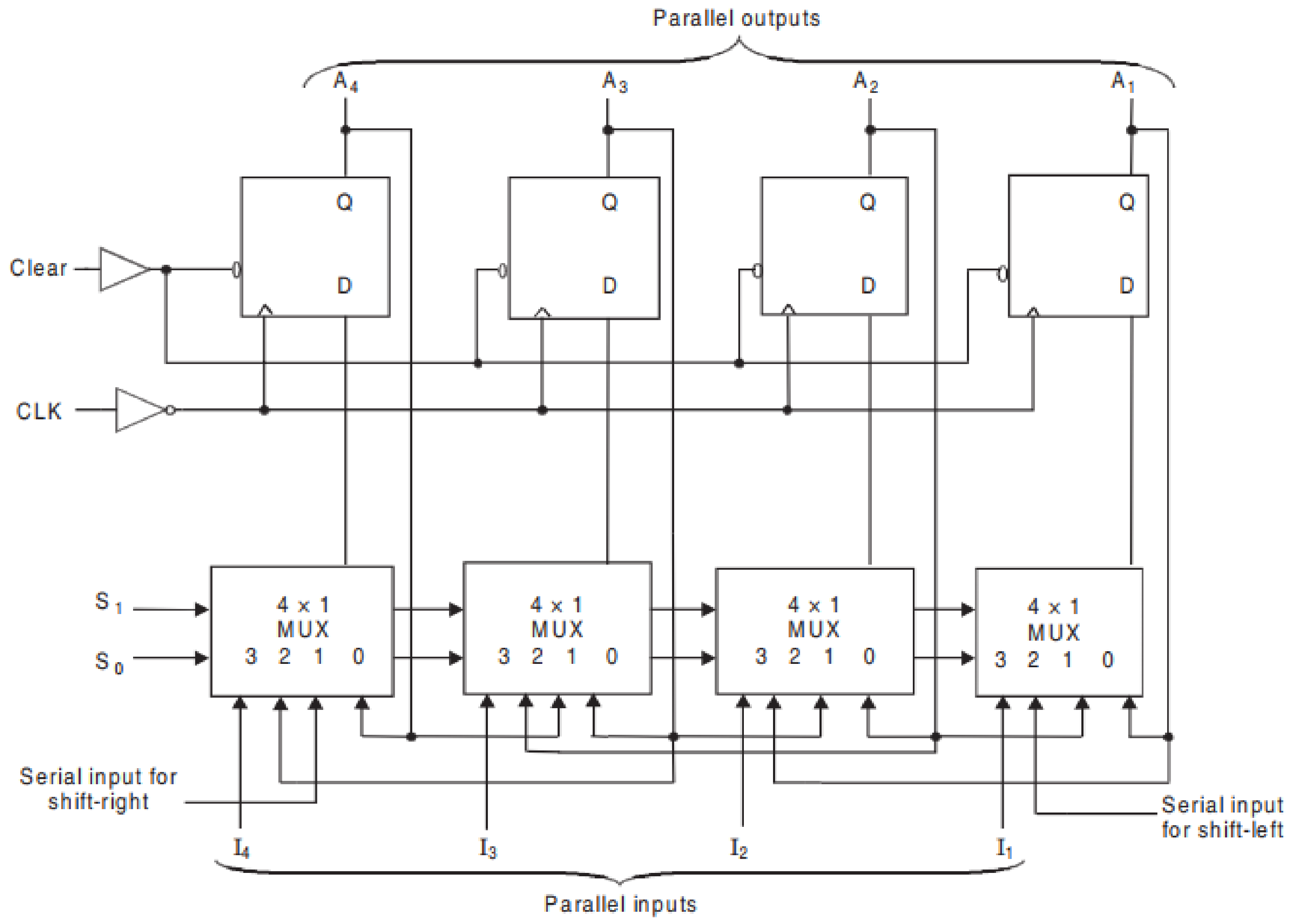
Bidirectional Shift Register



Universal Register

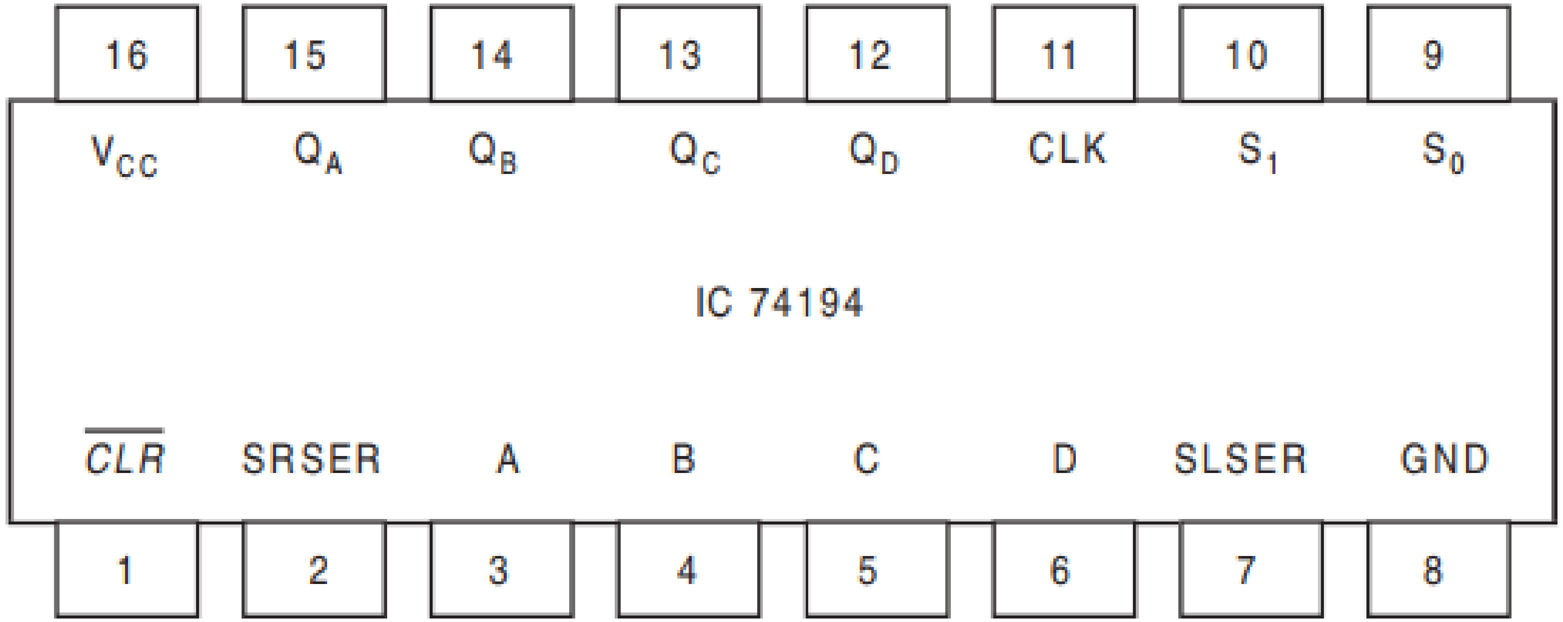
- **Right** or **Left** shift control input
- **Parallel load control**
- **Clear input**
- **Memory state**

4 Bit Universal Register: Circuit Diagram



Mode control		*Register operation*
S_1	S_0	
0	0	No change
0	1	Shift-right
1	0	Shift-left
1	1	Parallel load

4 Bit Universal Register: IC Pinout Diagram

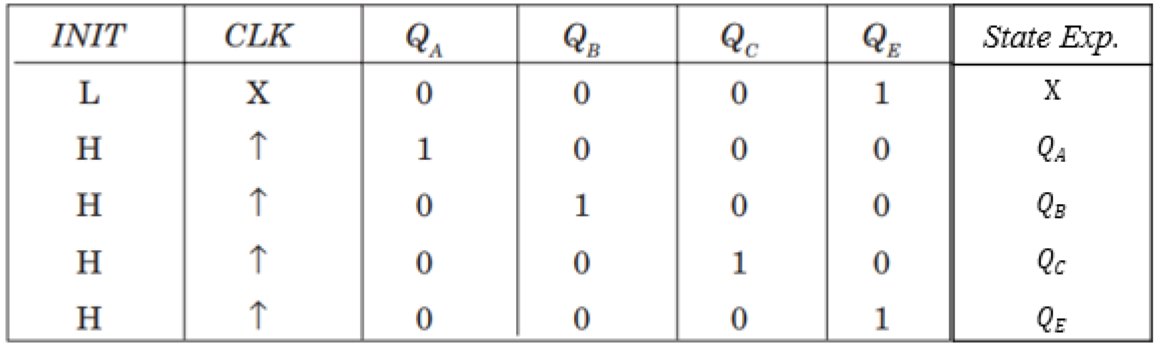


Register used as Counter

- Ring Counter
- Johnson Counter

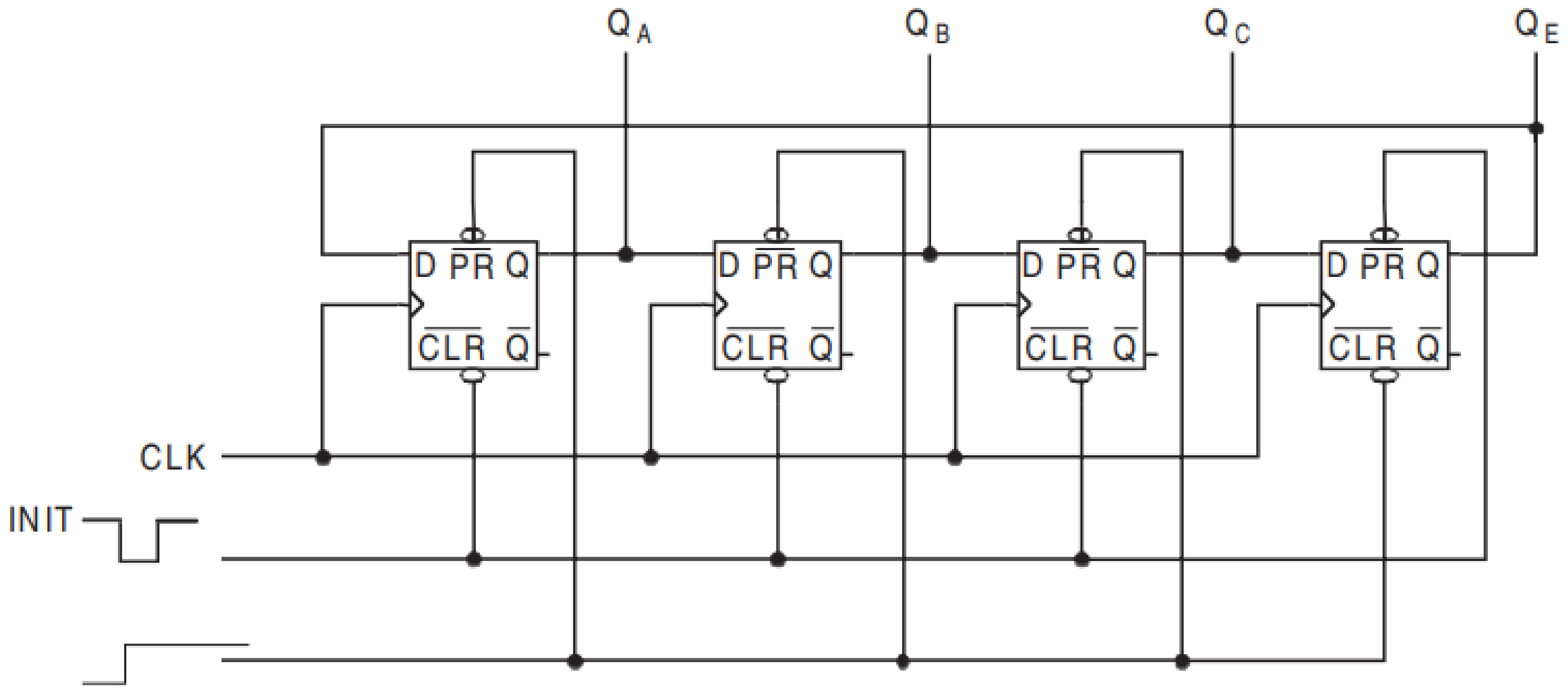
Ring Counter

- To make a **single** flip-flop "***set state***" once ($Q_i = 1$), while others will be in "***reset/clear state***"
- Output ($\boldsymbol{Q_E}$) of the last stage is fed back as input ($\boldsymbol{D_A}$) of the first stage.
- **K-bit** ring counter = **k** different **states**



INIT	*CLK*	Q_A	Q_B	Q_C	Q_E	*State Exp.*
L	X	0	0	0	1	X
H	$\uparrow$	1	0	0	0	Q_A
H	$\uparrow$	0	1	0	0	Q_B
H	$\uparrow$	0	0	1	0	Q_C
H	$\uparrow$	0	0	0	1	Q_E

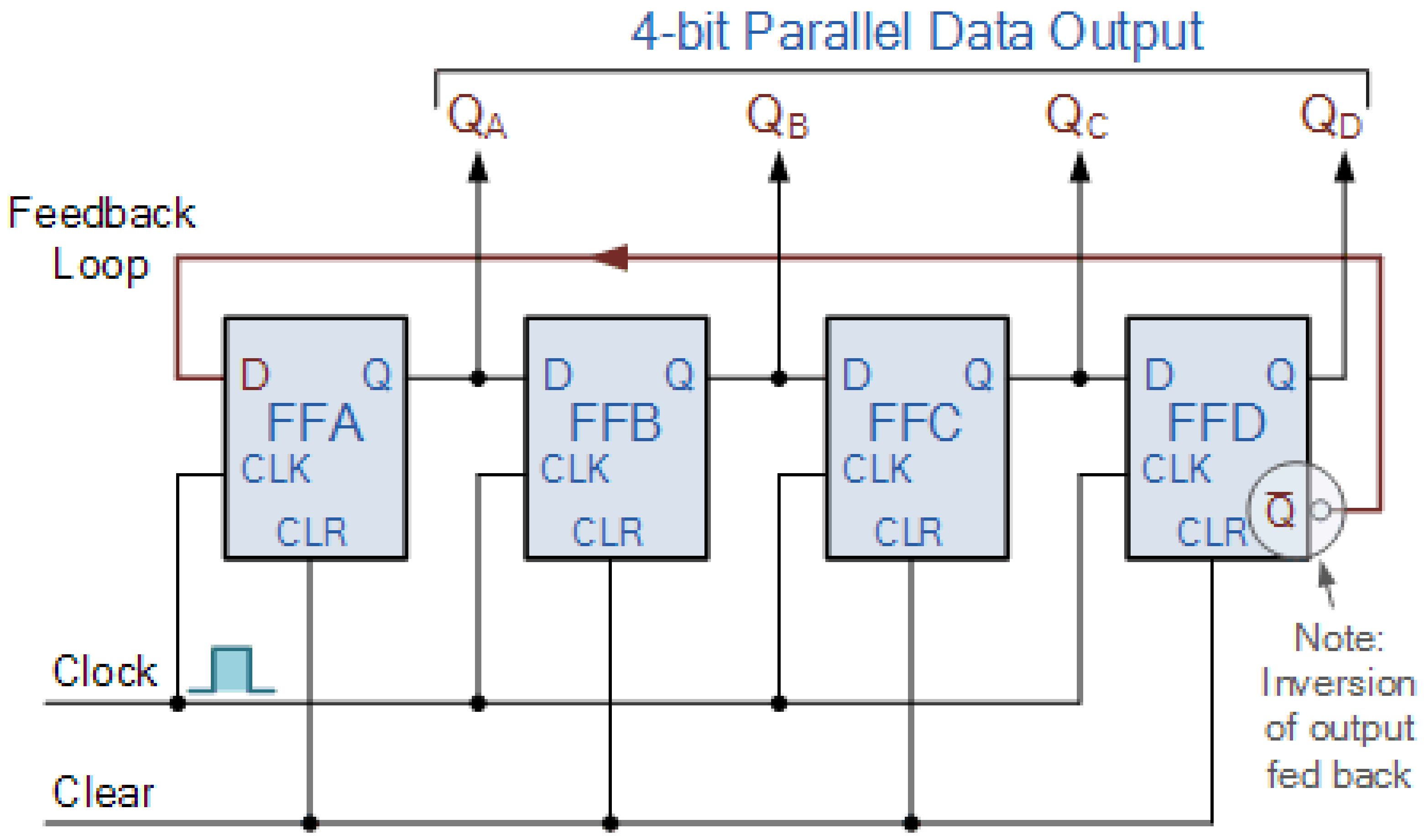
4 Bit Ring Counter with D Flip Flop



Johnson Counter

- Another shift register counter named as a ***switch-tail*** ring counter
  - A **circular shift register** where **complemented Output** (Q'_E) of the last stage is fed back as input (D_A) of the first stage.
- For **k-bit** Johnson counter = **2k** different **states**
  - Number of states are **doubled** than **vanilla ring counter**

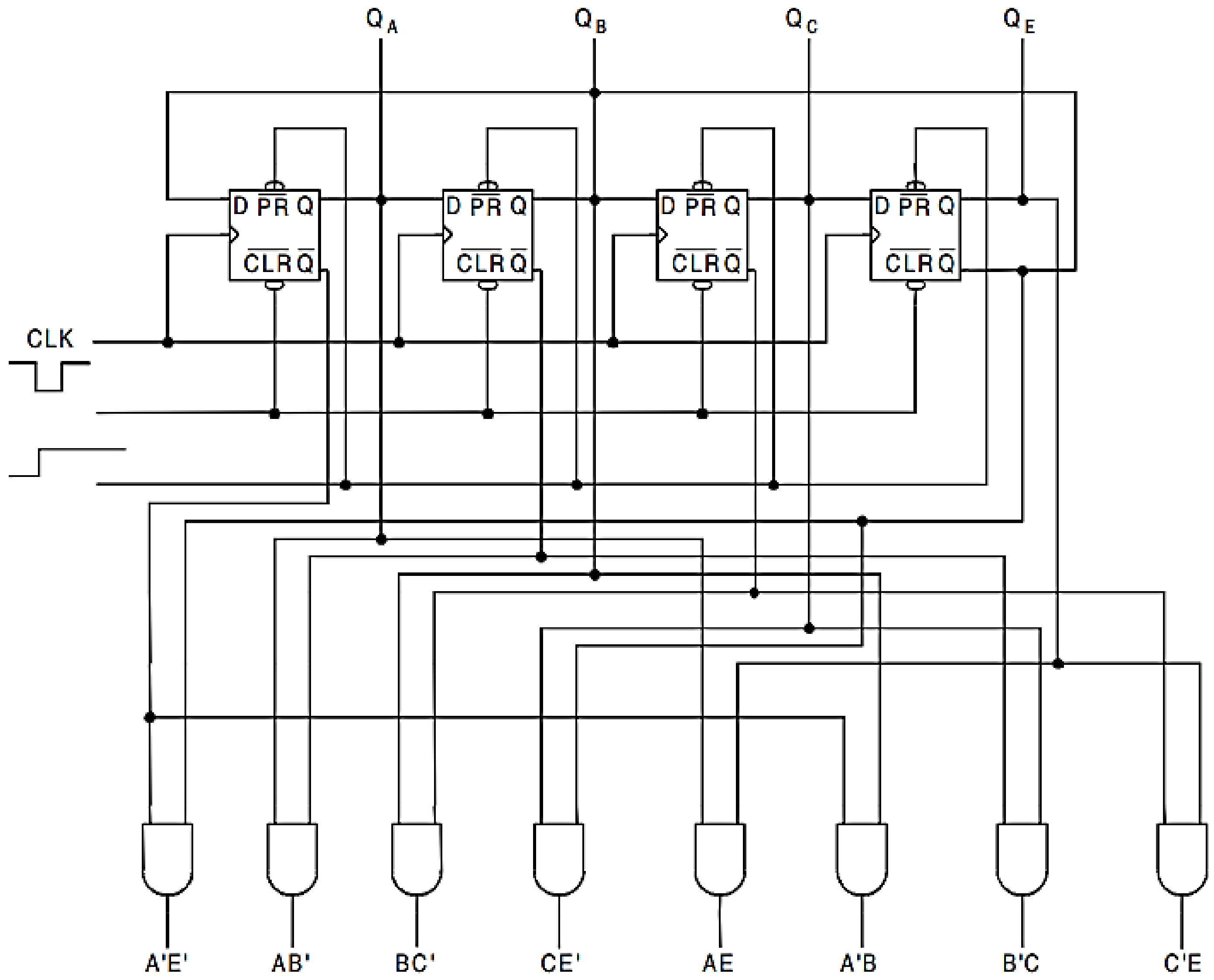
4 Bit Johnson Counter with D Flip Flop


4-bit Parallel Data Output
Q_A
Q_B
Q_C
Q_D
Feedback Loop
D Q FFA CLK CLR
D Q FFB CLK CLR
D Q FFC CLK CLR
D Q FFD CLK $\overline{Q}$ CLR
Clock
Clear
Note: Inversion of output fed back

Johnson Counter

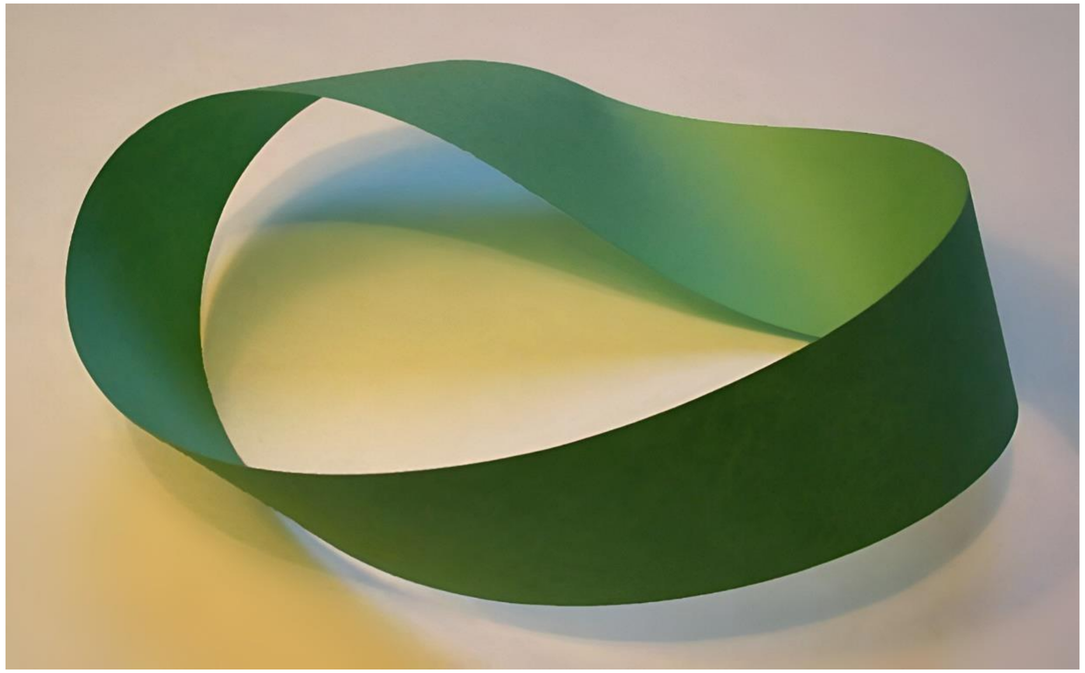
Sequence number	Flip-flop outputs				State Exp.
	A	B	C	E	
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

4 Bit Johnson Counter with State Expressions



Johnson Counter: Another Representation

It follows a ***mobius path*** – so known as **mobius counter**



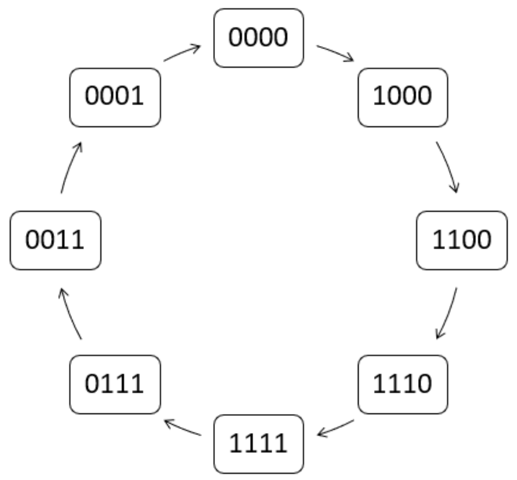
Johnson Counter: Unused States

If Johnson counter enters into any unused state, it will be inside that cycle until it is restarted.

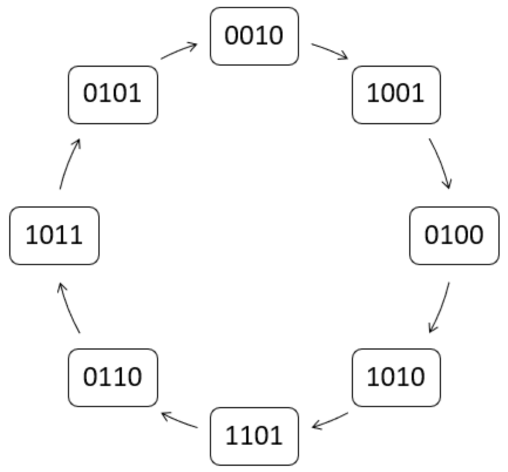
Solution: make a self-correction logic so that it can rectify itself.

Johnson Counter: Unused States

- Used states: 0, 1,3,7,8,12,14,15
- Unused states: 2,4,5,6,9,10,11,13



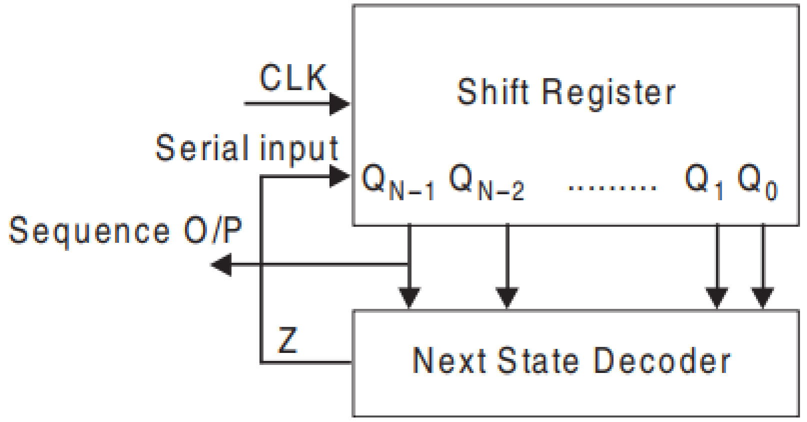
Used states' cycle



Unused states' cycle

Sequence Generator

- **Definition:**
  - A circuit that generates **a desired sequence** of bits in synchronization with **a clock**
- **Usage:** Random bit generator, code generator, prescribed period generator
- **Construction:** using a **shift register** and **next state decoder**
  - **Sequence Output is generated from Q_{n-1} output pin**
  - **Next state decoder is a function of all present states (Q_{n-1}, Q_{n-2}, … Q_1, Q_0)**
  - **Its output is connected to the input of the shift register (at FF_{n-1})**



Sequence Generator…

- **Basic concept:**
  - If the length of the patten = N
  - We could consider a register of n bits where, $n = N$
- But, the challenge is minimize ***n***
- We could start with constraint, ***Where,*** $\boldsymbol{N \le 2^n - 1}$

Design a 4-bit Sequence Generator

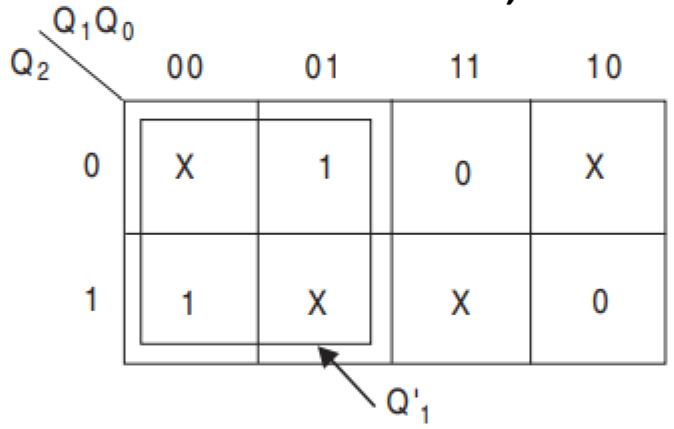
- Sequence: 1001; sequence length, N= 4
- Minimum number of flip-flops n
  - Where, $N \leq 2n - 1$
  - Here , n=3
- If these three flip-flops doesn’t produce all unique states (for inputs), we should increase the number of flip-flops
  - Increment n by 1.

Design a 4-bit Sequence Generator…

CLK	Flip-Flop outputs				Serial Input
	Q_2	Q_1	Q_0		Z
1	1	1	0	=6	0
2	0	1	1	=3	0
3	0	0	1	=1	1
4	1	0	0	=4	1
1	1	1	0		0
2	0	1	1		0
3	0	0	1		1
4	1	0	0		1

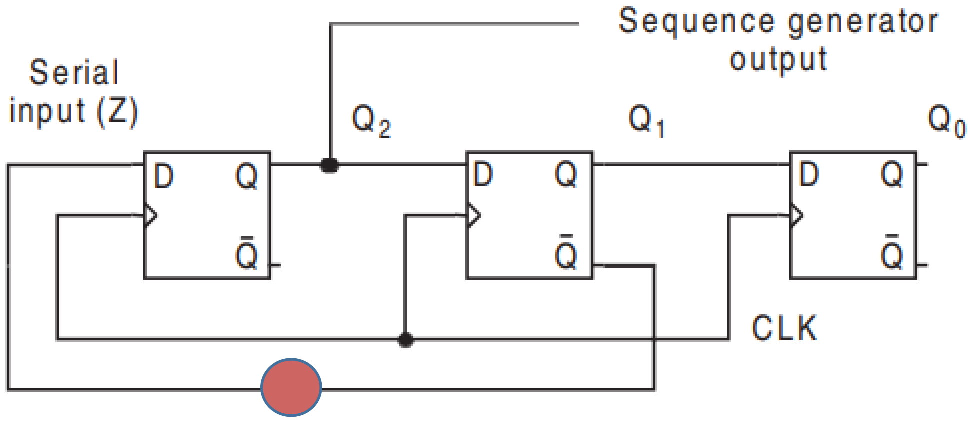
Here, Four states are different by inputs *Z*.

In the table, Z is true for two times only.



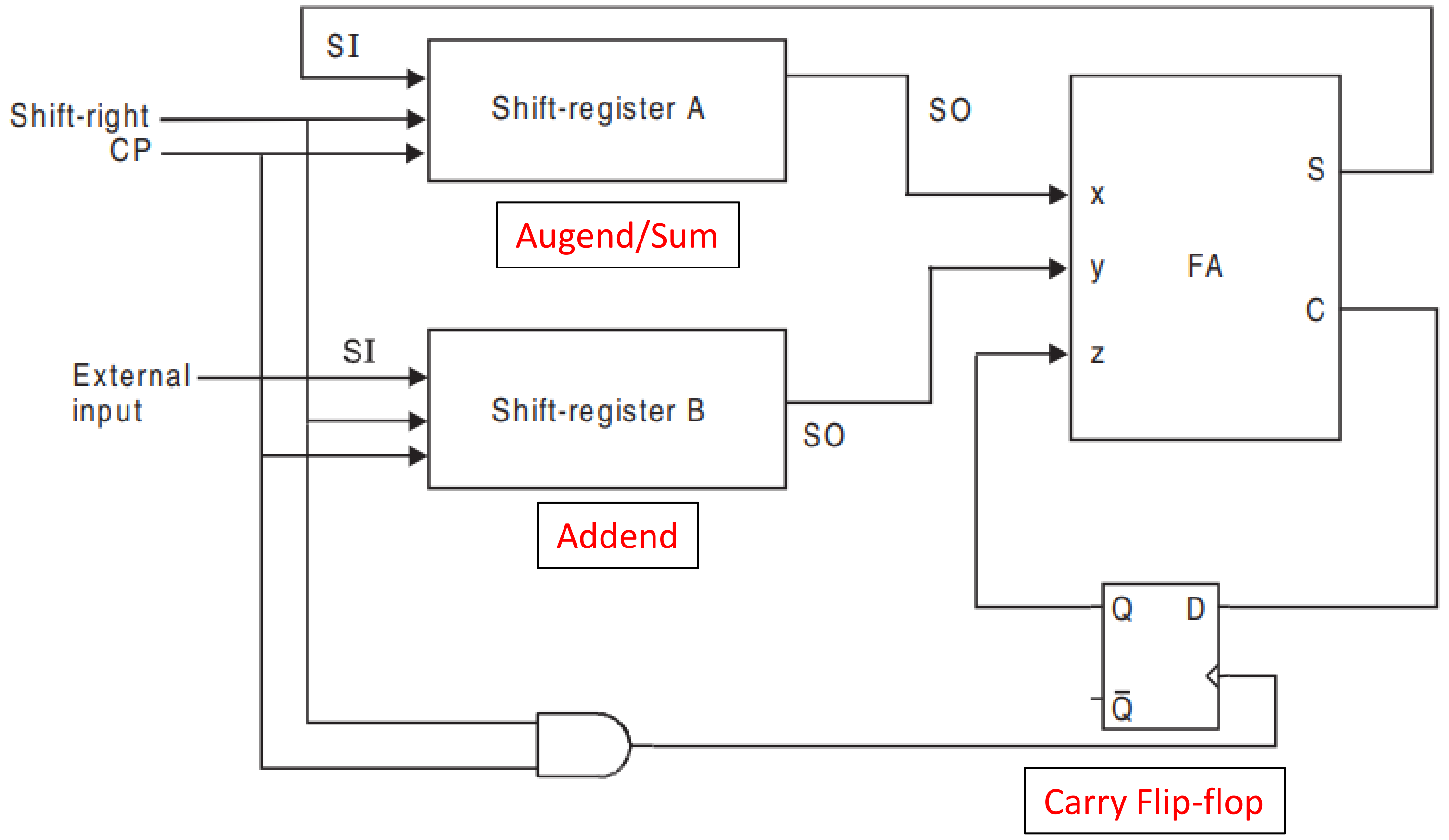
$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	X	1	0	X
1	1	X	X	0

$Z=Q_1'$



Logic for next state decoder.

Serial Addition


SI
Shift-register A
Shift-right
CP
SO
Augend/Sum
x
y
z
FA
S
C
External input
SI
Shift-register B
SO
Addend
Q
D
Q̄
Carry Flip-flop