DIGITAL DESIGN AND COMPUTER ORGANIZATION

Registers

Team DDCO

Department of Computer Science and Engineering

Registers(T1- section 6.1 and 6.2)

Introduction:

- A register is a **group of flip-flops**, each one of which shares a common clock and is capable of storing one bit of information.
- An n -bit register consists of a group of n flip-flops capable of **storing n bits of binary information.**
- In addition to the flip-flops, **a register may have combinational gates** that perform certain data-processing tasks.
- In its broadest definition, a register consists of a group of flip-flops together with gates that affect their operation.
- The **flip-flops hold the binary information, and the gates determine how the information is transferred into the register**.

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Registers

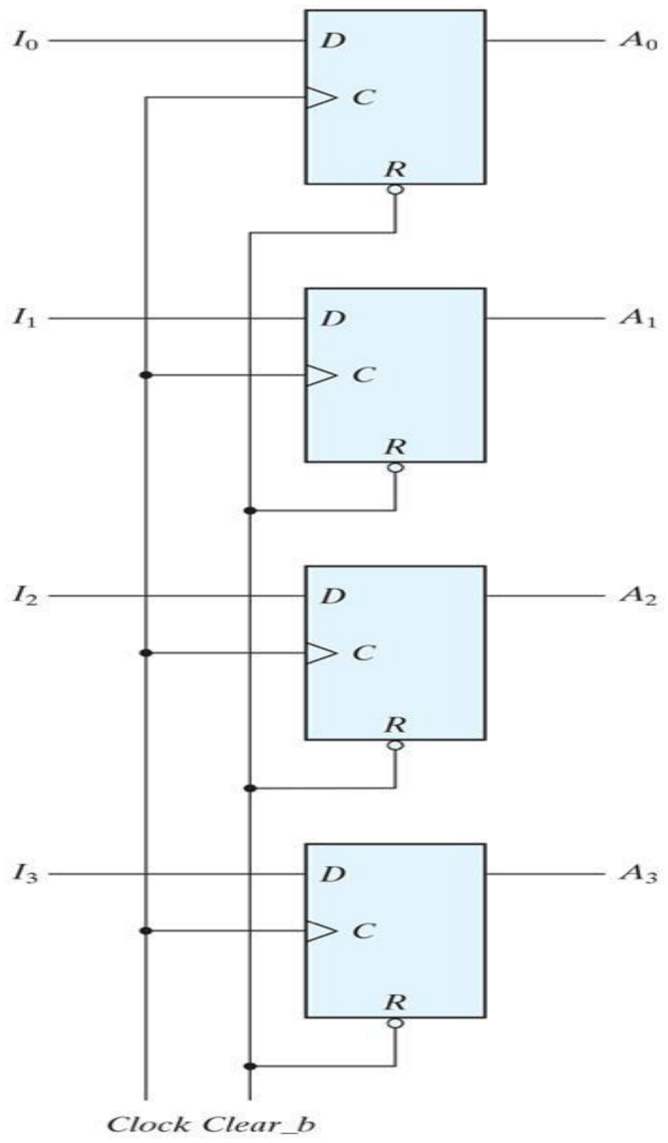
- **A counter is essentially a register that goes through a predetermined sequence of binary states.**
- The gates in the counter are connected in such a way as to produce the prescribed sequence of states.
- **The simplest register is one that consists of only flip-flops, without any gates**
- Figure 6.1shows such a register constructed with four D -type flip-flops to form a four-bit data storage register. **The common clock input triggers all flip-flops on the positive edge of each pulse**, and the binary data available at the four inputs are transferred into the register. The value of (I3 , I2 , I1 , I0) immediately before the clock edge determines the value of (A3 , A2 , A1 , A0) after the clock edge.
- The **input Clear_b goes to the active-low** R (reset) input of all four flip-flops. When this **input goes to 0, all flip-flops are reset asynchronously**. The Clear_b input is useful for clearing the register to all 0's prior to its clocked operation. **The R inputs must be maintained at 1 for normal operation** .

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Registers

Figure 6.1
Four-bit register.



Registers

Registers with Parallel Load:

- **The pulses are applied to all flip-flops and registers in the system**. The master clock acts like a drum that supplies a constant beat to all parts of the system.

- The transfer of new information into a register is referred to as **loading** or **updating** the register. If all the bits of the register are **loaded simultaneously with a common clock pulse**, we say that the loading is done in parallel .

- A clock edge applied to the C inputs of the register of Fig. 6.1 will load all four inputs in parallel. In this configuration, if the contents of the register must be left unchanged, the inputs must be held constant or the clock must be inhibited from the circuit

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Registers

Registers with Parallel Load:

- **Parallel Load Mechanism:**
  - Registers with parallel load **update all bits simultaneously** using a common clock edge.
  - A control signal (**Load**) decides whether **new data** is written or **existing data** is retained.
  - This ensures fast and synchronized data transfer across all bits.

- **Clock Control and Synchronization:**
  - Instead of gating the clock, data flow is controlled at the register inputs using multiplexers.
  - This avoids timing issues caused by variable delays in clock paths.
  - Flip-flops retain or load new data based on the Load signal during each clock pulse.

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Registers Registers with Parallel Load:

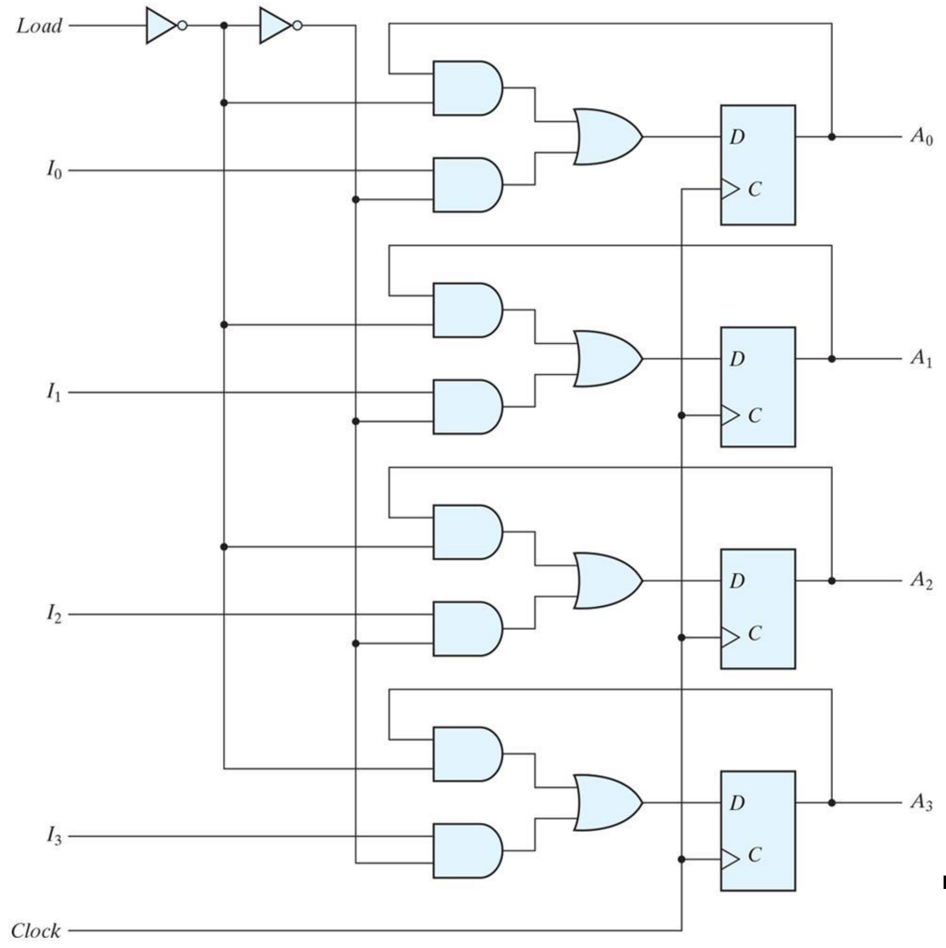
Clock Gating (Old Method)
In **clock gating**, you literally **stop or pass** the clock to flip-flops depending on an **Enable** signal.
Example: If Enable=0, the clock pulse is blocked; if Enable=1, the clock pulse reaches the flip-flops.
Problem:
The clock has to travel through extra logic (AND gates, buffers, etc.).
These extra delays can differ slightly for different flip-flops.
Result: Not all flip-flops see the clock edge at exactly the same time → **timing skew**.
Skew can cause some flip-flops to update while others don't, leading to corrupted register values.
Input Control with Multiplexers (Better Method)
Instead of touching the clock, keep the **clock clean and uniform** for all flip-flops.
Use a **multiplexer (MUX)** in front of each flip-flop's **D input**:
One input of the MUX is the **new data (D_in)**.
The other input is the **old output (Q)** fed back.
A **Load (Enable)** signal selects which one to send to the flip-flop's D input.

Pearson
Copyright © 2018 Pearson India Education Services Pvt. Ltd

Registers with Parallel Load:

The transfer of information from the data inputs or the outputs of the register is done simultaneously with all four bits in response to a clock edge.

Figure 6.2
Four-bit register with parallel load.



Load=0 retains data

Load =1
Register accepts new data

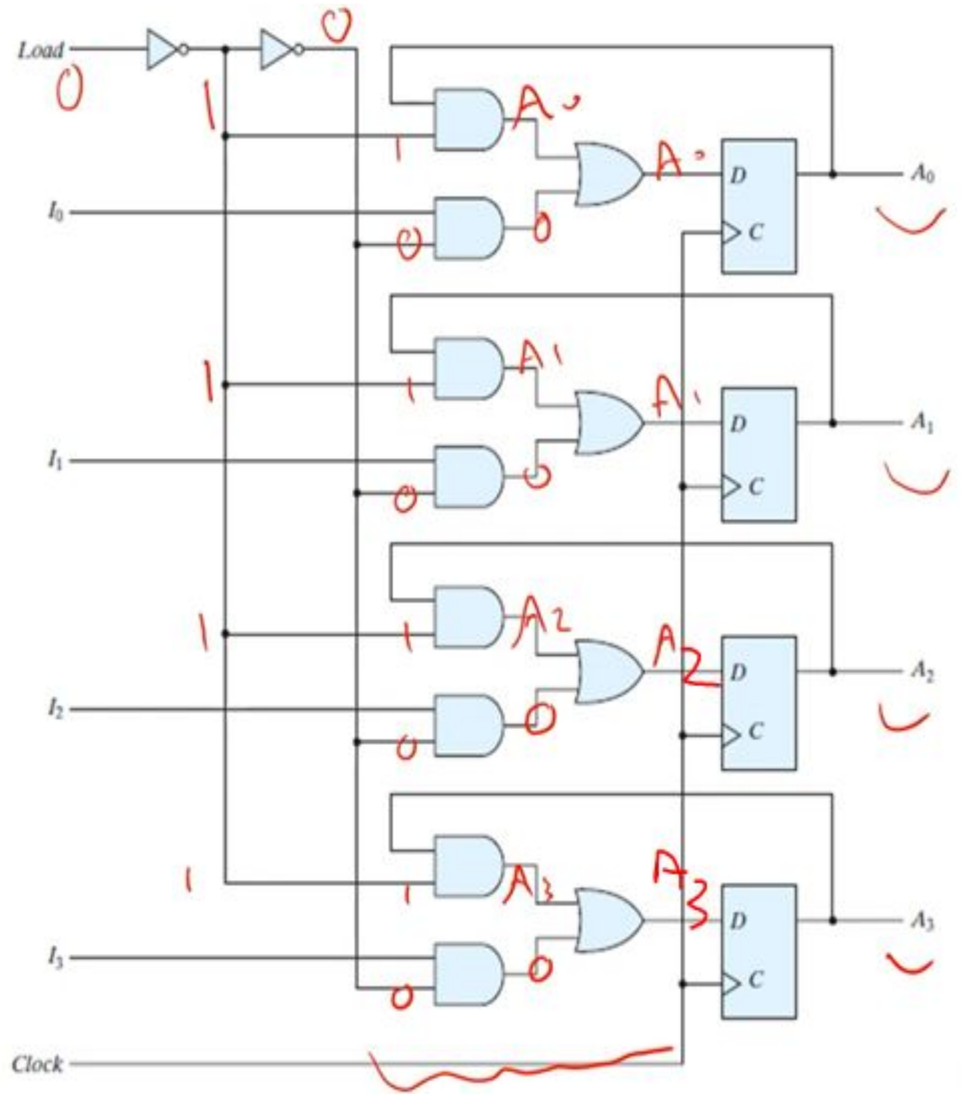
Pearson

Ltd

Registers with Parallel Load:

Figure 6.2
Four-bit register with parallel load.

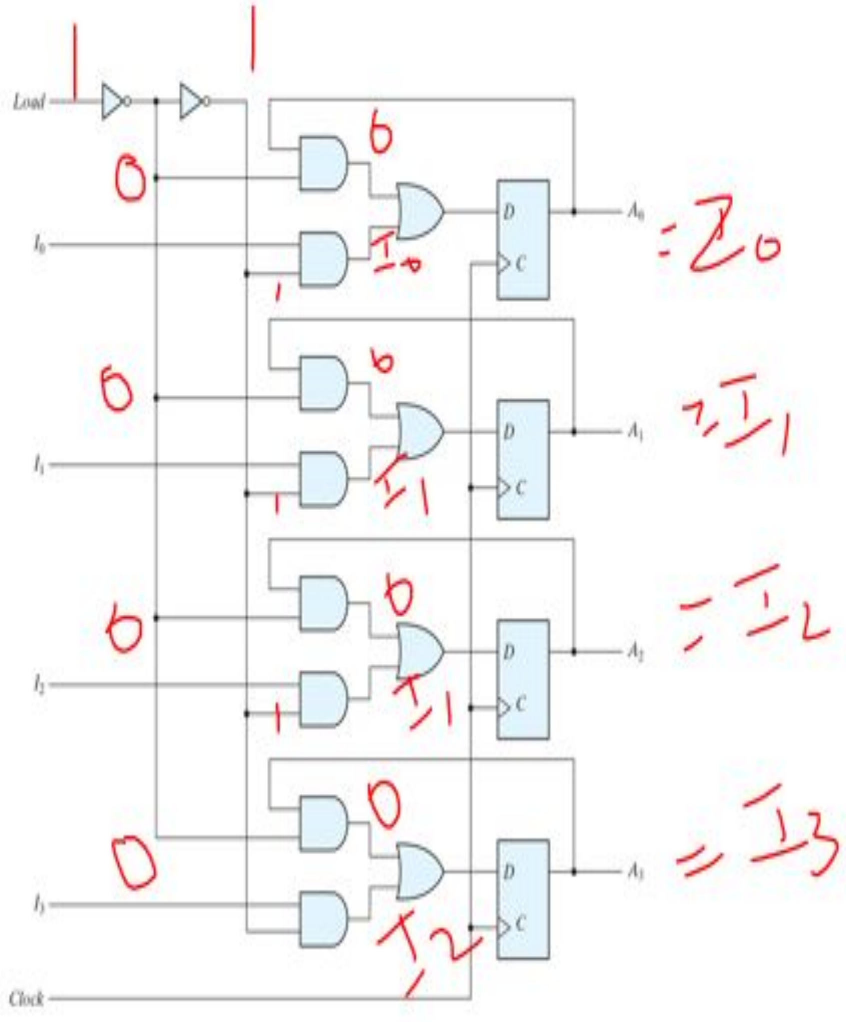
Load=0 retains data



Pearson

Registers with Parallel Load:

Figure 6.2
Four-bit register with parallel load.



Load =1
Register accepts new data

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Registers

Shift Registers: section6.2(T1)

A register capable of shifting the binary information held in each cell to its neighboring cell, in a selected direction, is called a shift register.

The logical configuration of a shift register consists of a chain of flip-flops in cascade, **with the output of one flip-flop connected to the input of the next flip-flop.**

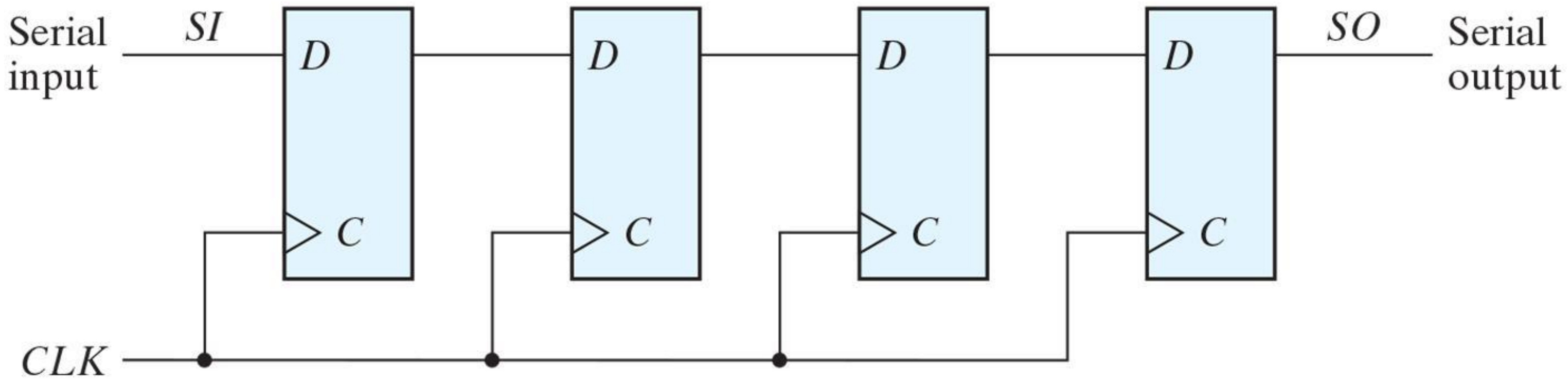
All flip-flops receive common clock pulses, which activate the shift of data from one stage to the next.

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Shift Registers

Figure 6.3
Four-bit shift register.



The output of a given flip-flop is connected to the D input of the flip-flop at its right.
This shift register is **unidirectional** (left-to-right).
The serial input determines what goes into the leftmost flip-flop during the shift.
The serial output is taken from the output of the rightmost flip-flop.

Shift Registers

Serial Transfer:

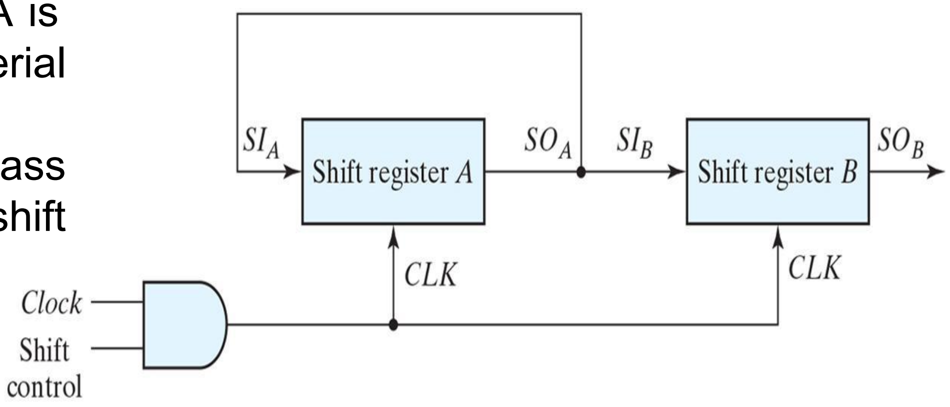
- Serial transfer occurs **one bit at a time**, unlike parallel transfer where **all bits move together**.
- Data is shifted from **register A to register B**, using **shift registers** connected in series.
- To prevent data loss, register A is made to **circulate its own contents**, while register B shifts in the data.
- A **control signal** (Shift Control) enables shifting for **a fixed number of clock cycles**.
- When active, the signal allows **4 clock pulses (T1–T4)** through AND gates to both registers.
- Each **rising clock edge** shifts data one step, syncing both registers with each clock pulse.

Copyright © 2018 Pearson India Education Services Pvt. Ltd

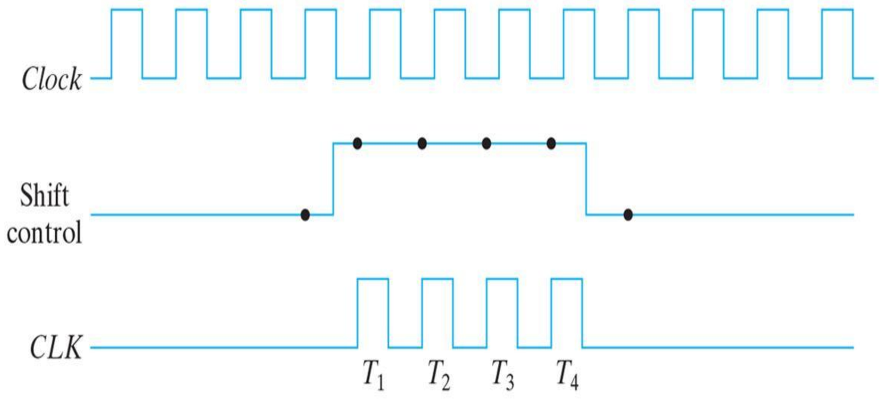
Serial Transfer

The serial output (SO) of register A is connected to the serial input (SI) of register B. To prevent the loss of information stored in the source register, the information in register A is made to circulate by connecting the serial output to its serial input.

AND gate that allows clock pulses to pass into the CLK terminals only when the shift control is active.



(a) Block diagram



(b) Timing diagram

Figure 6.4
Serial transfer from register A to register B.

Serial Transfer

Summary:

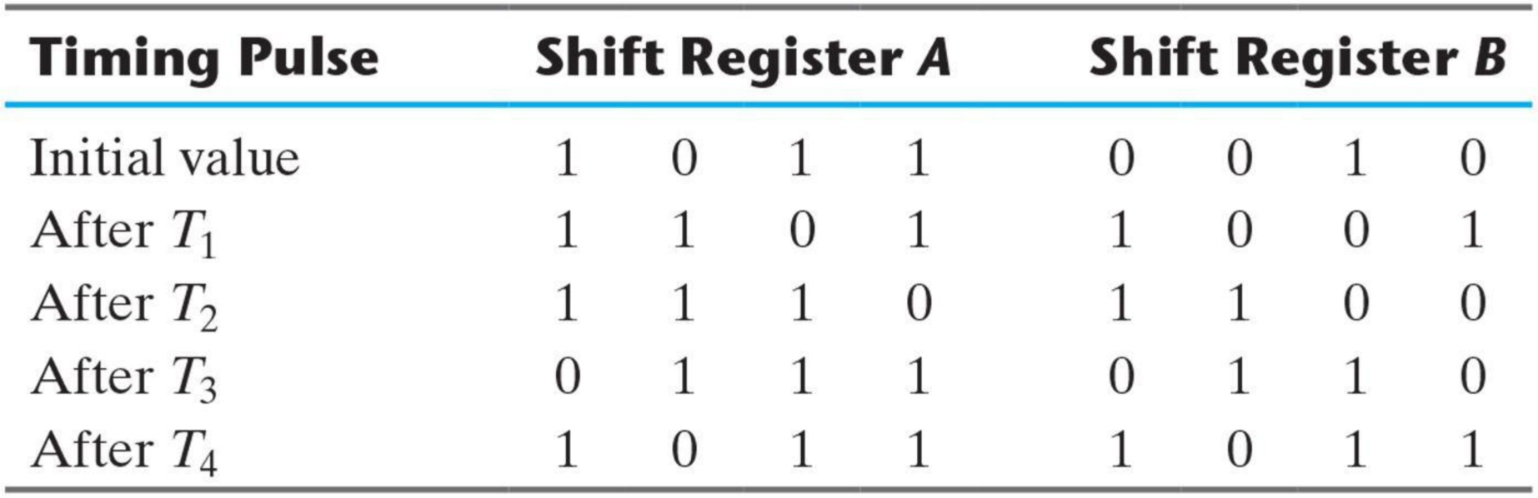
Two shift registers: **A** (source) and **B** (destination).
Data is transferred **serially** bit by bit from register A → register B.
A clock signal drives both registers.
But the transfer should **not happen all the time** → only when we want.
That's where **Shift Control** comes in.

The **Shift Control** signal is ANDed with the clock before reaching the registers.
If **Shift Control = 1**:
The AND gate passes the clock pulses.
Registers A and B **shift on each clock edge**.
Data moves serially from A → B.
If **Shift Control = 0**:
Clock pulses are blocked.
Registers **hold their data** (no shifting).
So **Shift Control acts like an enable signal for shifting**.

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Serial Transfer

With the first pulse, T1, the rightmost bit of A is shifted into the leftmost bit of B and is also circulated into the leftmost position of A. At the same time, all bits of A and B are shifted one position to the right. The previous serial output from B in the rightmost position is lost, and its value changes from 0 to 1.



Timing Pulse	Shift Register *A*				Shift Register *B*			
Initial value	1	0	1	1	0	0	1	0
After T_1	1	1	0	1	1	0	0	1
After T_2	1	1	1	0	1	1	0	0
After T_3	0	1	1	1	0	1	1	0
After T_4	1	0	1	1	1	0	1	1

Table 6.1 Serial-Transfer Example.

Thus, the contents of A are copied into B, so that the contents of A remain unchanged i.e., the contents of A are restored to their original value.

Serial Transfer

- In the parallel mode, information is available from all bits of a register and all bits can be transferred simultaneously during one clock pulse.

- In the serial mode mode, the registers have a single serial input and a single serial output. The information is transferred one bit at a time while the registers are shifted in the same direction.

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Shift Registers

Serial Addition:

- Operations in digital computers are usually done in **parallel because that is a faster mode of operation.** Serial operations are slower because a datapath operation takes several clock cycles, but **serial operations have the advantage of requiring fewer hard ware components**. In VLSI circuits, they require less silicon area on a chip

- The two binary numbers to be added serially are stored in two shift registers. Two binary numbers are stored in shift registers A (augend) and B (addend). A single full adder adds one bit pair at a time, starting from the least significant bit (LSB).

- The carry-out is stored in a D flip-flop, which feeds into the carry-in of the next bit pair in the next clock cycle.

- With each clock pulse, both registers shift right, a new sum is generated, and carry is updated.

- This continues for n cycles (n = number of bits) until the shift control is disabled, completing the addition process.

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Serial Addition

- It is used perform bit by bit addition in serial form
- It is done using flipflop and adder
- D flipflop- stores carry output after addition
- Right shit register A-A
- Right shift register-B-B
- Add A+B bit by bit using Full adder

Initially, register A holds the augend, register B holds the addend, and the **carry flip-flop is cleared to 0.** The outputs (SO) of A and B provide a pair of significant bits for the full adder at x and y. Output Q of the flip-flop provides the input carry at z. The shift control enables both registers and the carry flip-flop, so at the next clock pulse, both registers are shifted once to the right, the sum bit from S enters the leftmost flip-flop of A, and the output carry is transferred into flip-flop Q. The shift control enables the registers for a number of clock pulses equal to the number of bits in the registers. For each succeeding clock pulse, a new sum bit is transferred to A, a new carry is transferred to Q, and both registers are shifted once to the right

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Serial Addition

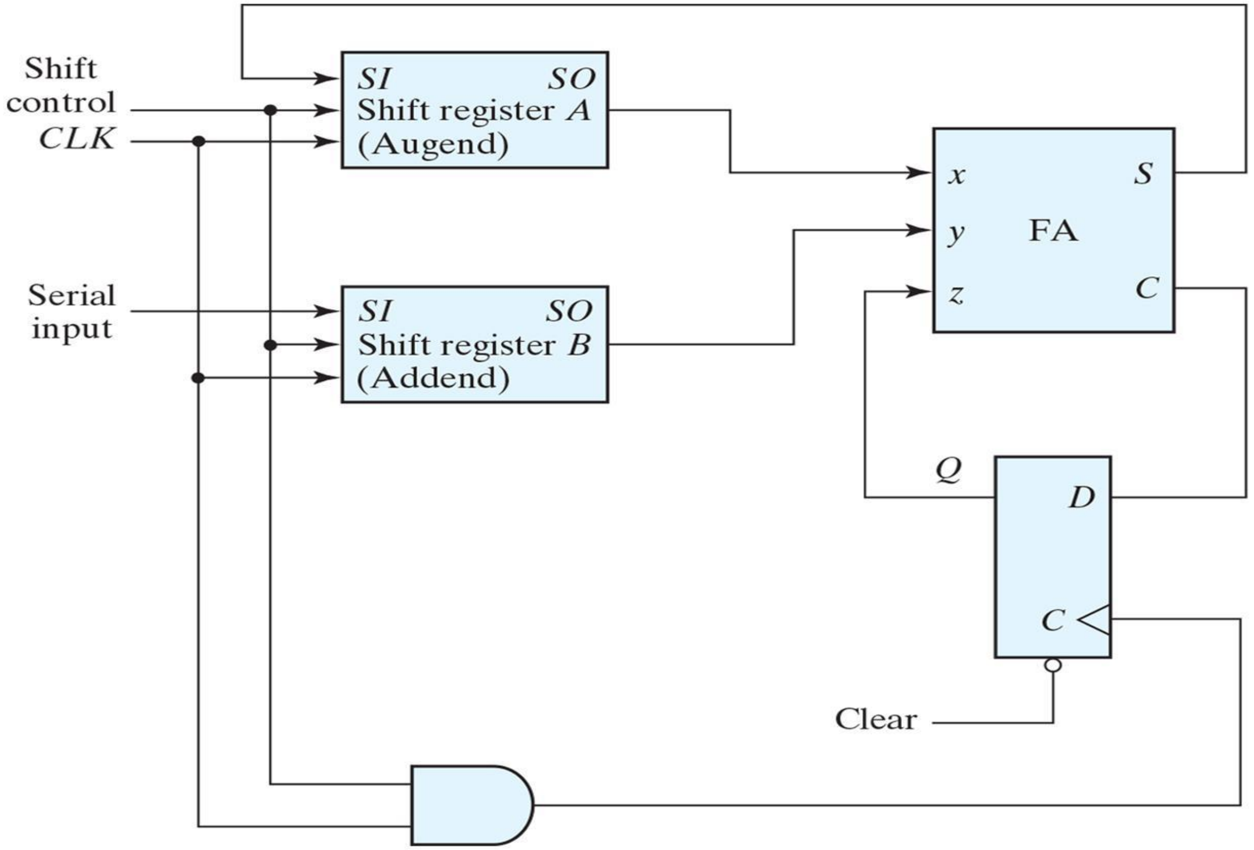
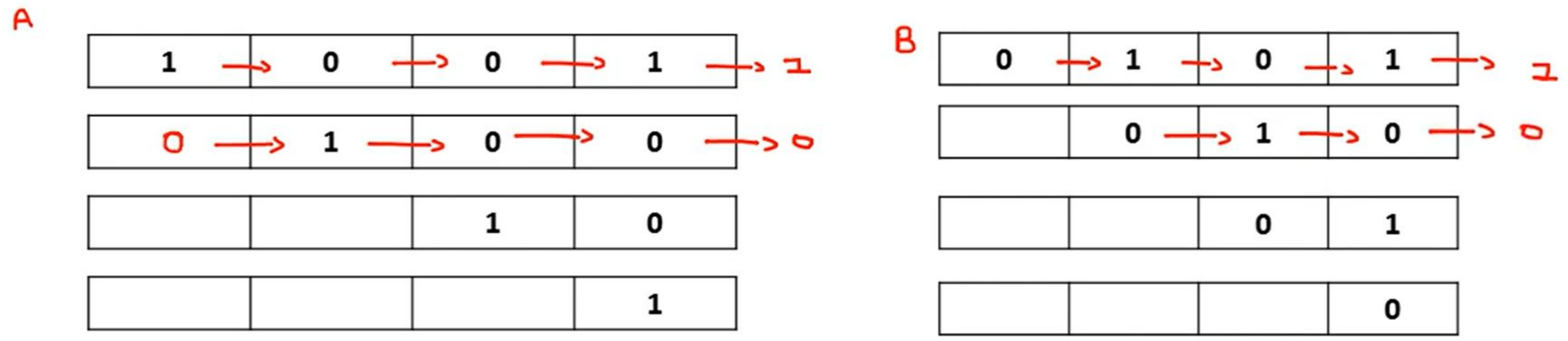
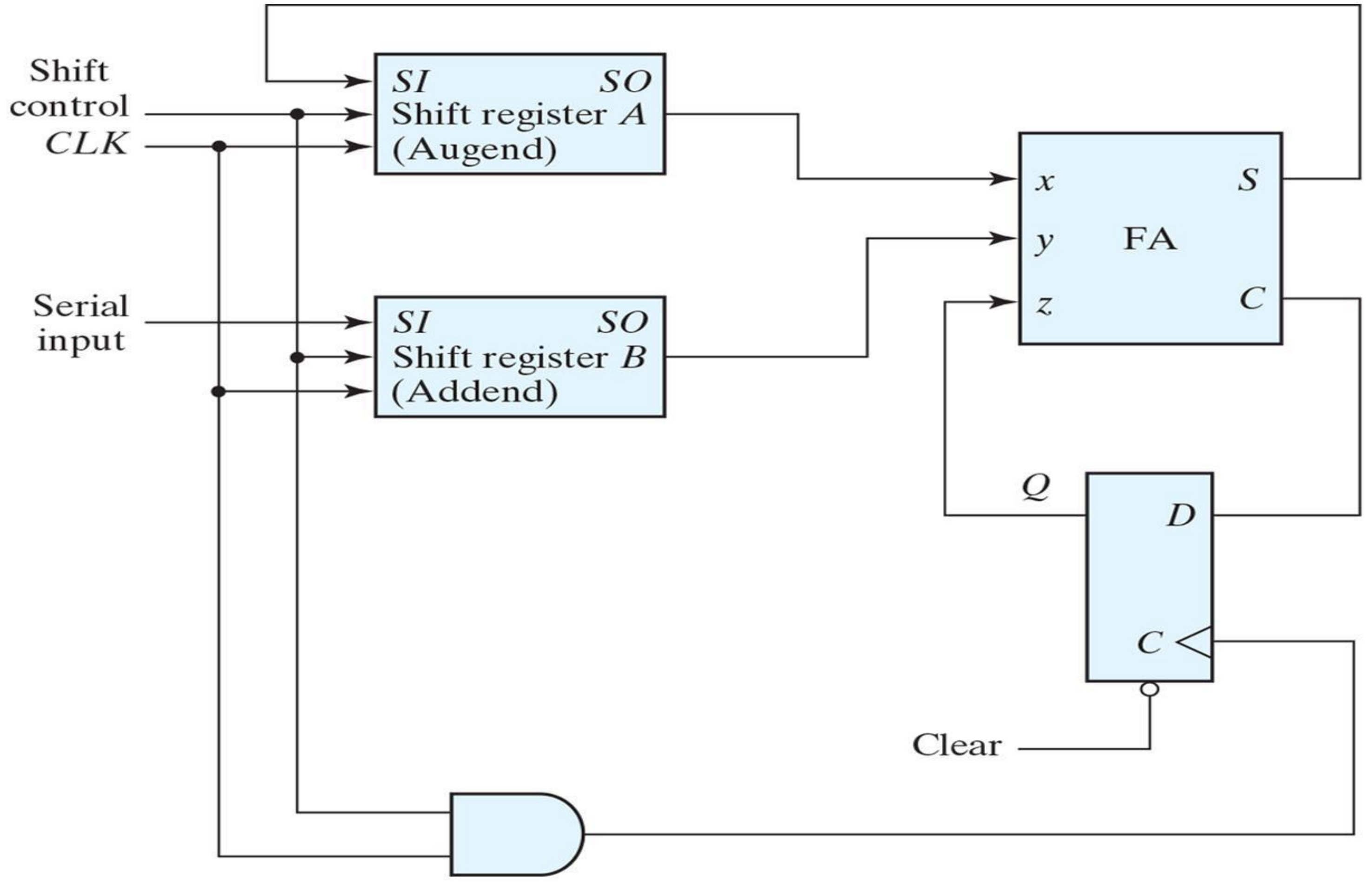


Figure 6.5: Serial adder.

Z is Cin
C is C out
Q=0 initially

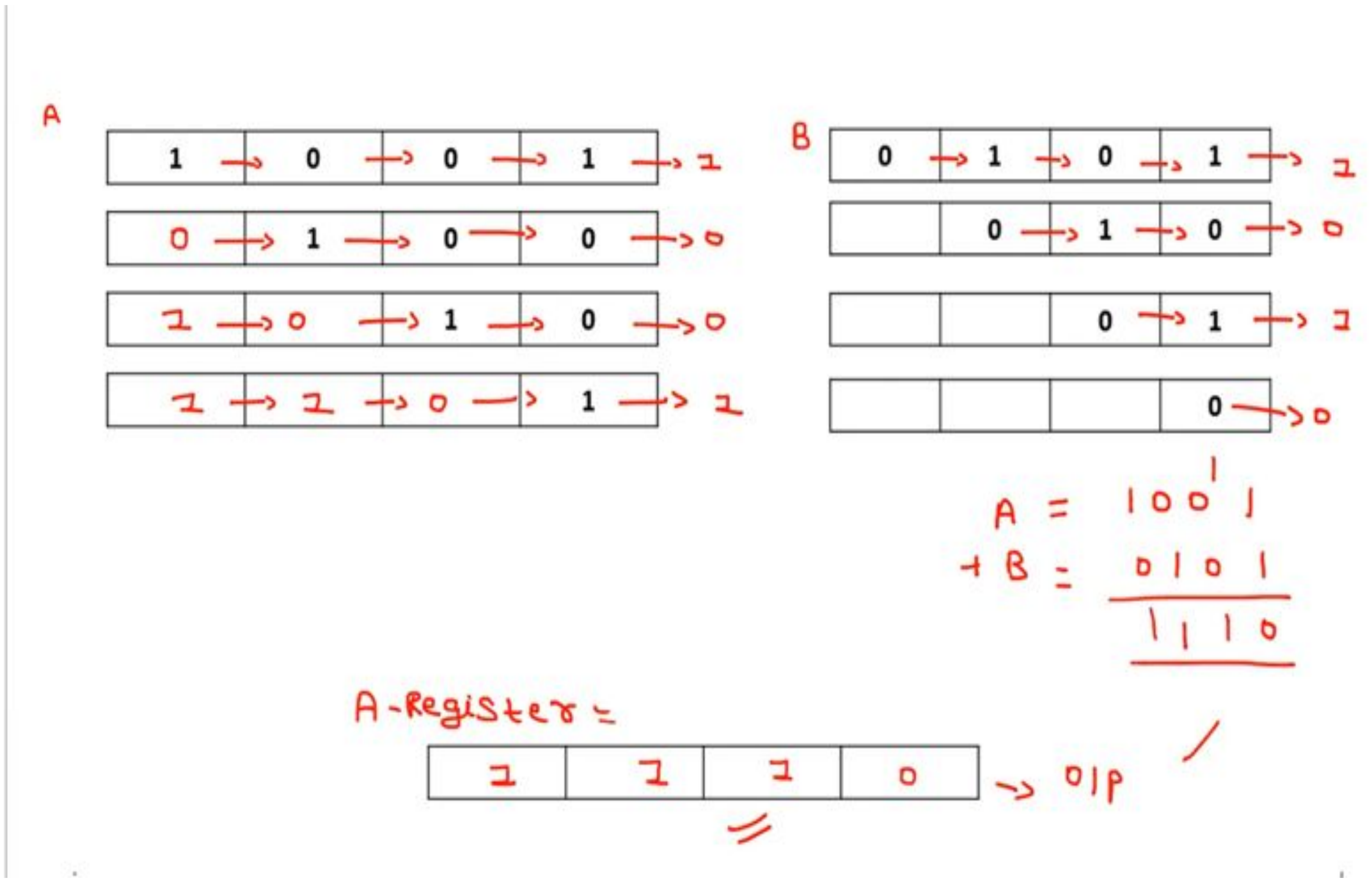




Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Serial Addition



Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Serial Addition

Comparing the serial adder with the parallel adder described in, we note several differences.

- The parallel adder uses **registers** with a parallel load, whereas the serial adder uses shift registers.
- The number of **full-adder circuits** in the parallel adder is equal to the number of bits in the binary numbers, whereas the serial adder requires only one full-adder circuit and a carry flip-flop.
- Excluding the registers, the parallel adder is a **combinational** circuit, whereas the serial adder is a **sequential circuit** which consists of a full adder and a flip-flop that stores the output carry.
- This design is typical in serial operations because the result of a bit-time operation may depend not only on the present inputs, but also on previous inputs that must be stored in flip-flops

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Serial Addition

To show that serial operations can be designed by means of **sequential circuit procedure**, we will **redesign** the serial adder with the use of a state table.

First, we assume that two shift registers are available to store the binary numbers to be added serially. The serial outputs from the registers are designated by x and y.

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Serial Addition

Table 6.2
State Table for Serial Adder.

The next state of Q is equal to the output carry.

Present State	Inputs		Next State	Output	Flip-Flop Inputs	
Q	x	y	Q	S	J_Q	K_Q
0	0	0	0	0	0	X
0	0	1	0	1	0	X
0	1	0	0	1	0	X
0	1	1	1	0	1	X
1	0	0	0	1	X	1
1	0	1	1	0	X	0
1	1	0	1	0	X	0
1	1	1	1	1	X	0

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Serial Addition

$$J_Q = xy$$

$$K_Q = x'y' = (x + y)'$$

$$S = x \oplus y \oplus Q$$

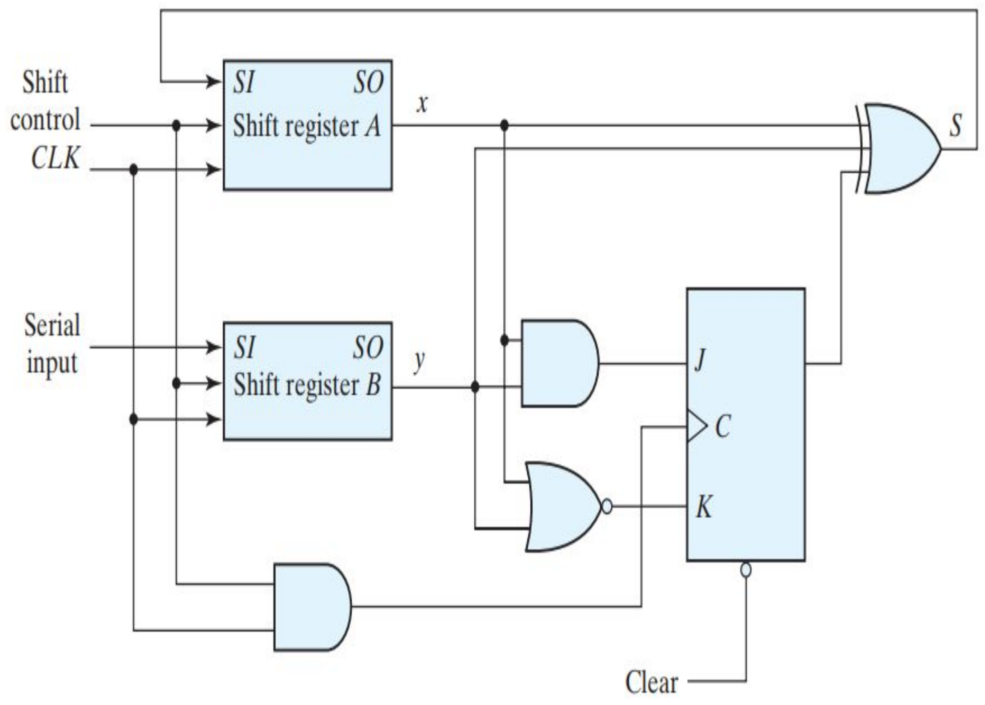


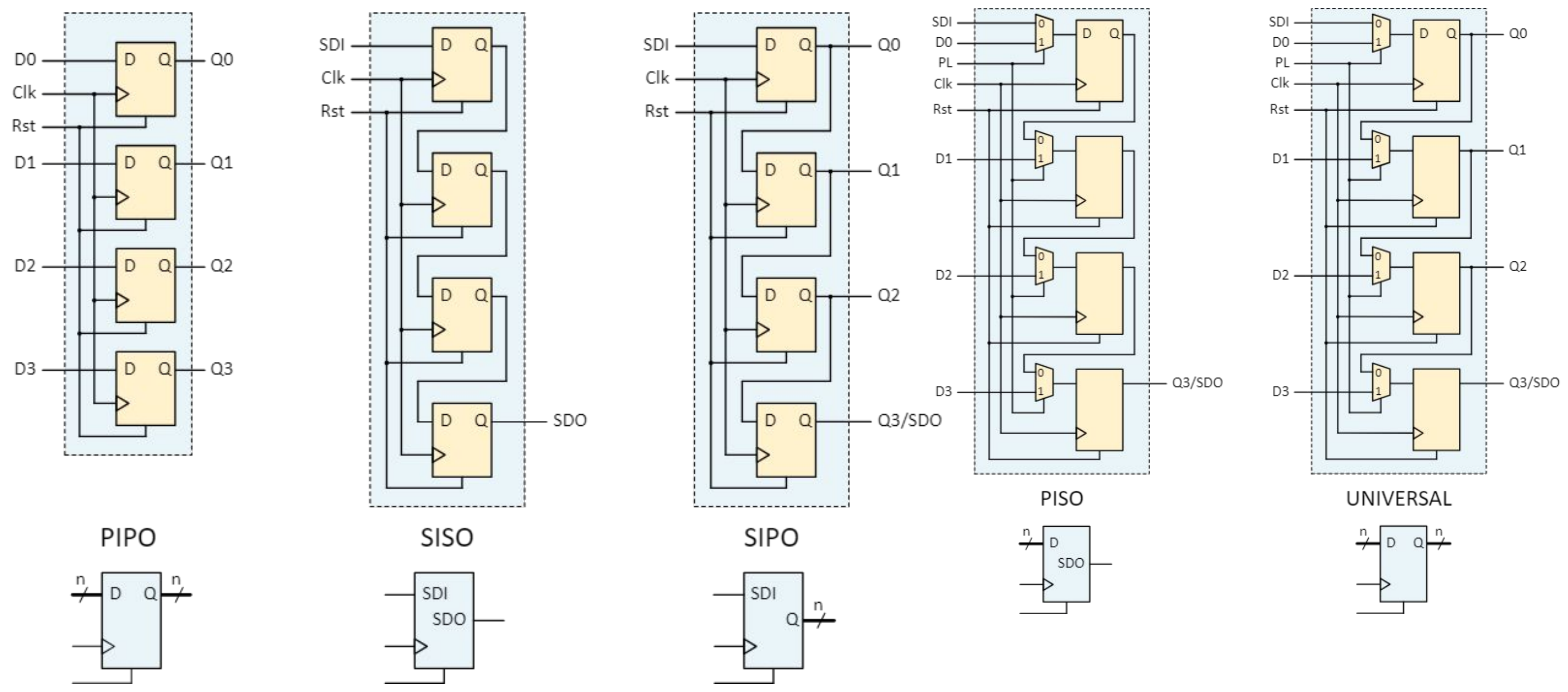
FIGURE 6.6
Second form of serial adder

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Types of Representation of Registers

Universal register



The universal register, as the name implies, can be used as a PIPO, SISO, SIPO, or PISO register.

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Applications

Parallel Data Transfer

Where used: Communication between processor and peripherals.

Example: In a keyboard interface, parallel registers capture all key signals at once for quick transfer to the processor.

Why important: Transfers multiple bits simultaneously, ideal for high-speed operations.

Serial Data transfer:
USB as Serial Communication

Every time you connect a pen drive to your computer, the data is transferred serially (one bit at a time)

uses high-speed serial communication over just a few wires instead of large parallel buses.

Copyright © 2018 Pearson India Education Services Pvt. Ltd

Applications

Universal Shift Registers
Where used: Versatile data handling.

Example: In data encryption hardware, universal shift registers can load, shift, and output data in multiple formats.

Why important: Provides flexibility for complex digital operations.

Pearson

Copyright © 2018 Pearson India Education Services Pvt. Ltd

MCQ

In a 4-bit parallel load register, what happens when the Load signal is 0 during a clock pulse?
A) All bits are shifted to the right
B) The register is cleared to 0000
C) The existing data is retained
D) The register loads new data from inputs

Which of following is true about serial Transfer?

A. In the serial mode, the registers have a single serial input and a single serial output
B. If Shift Control = 0: The AND gate passes the clock pulses. Registers A and B shift on each clock edge.
C. In the serial mode, information is available from all bits of a register and all bits can be transferred simultaneously during one clock pulse.
D. all statements are true

Copyright © 2018 Pearson India Education Services Pvt. Ltd

MCQ

In a 4-bit parallel load register, what happens when the Load signal is 0 during a clock pulse?
A) All bits are shifted to the right
B) The register is cleared to 0000
C) The existing data is retained
D) The register loads new data from inputs

Answer: C – The existing data is retained when Load = 0, since no new parallel input is latched.

Which of following is true about serial Transfer?
A. In the serial mode, the registers have a single serial input and a single serial output
B. If Shift Control = 0: The AND gate passes the clock pulses. Registers A and B shift on each clock edge.
C. In the serial mode, information is available from all bits of a register and all bits can be transferred simultaneously during one clock pulse.
D. all statements are true
ans: A

Copyright © 2018 Pearson India Education Services Pvt. Ltd

THANK YOU

Team DDCO
Department of Computer Science and Engineering