Life Story of Registers

Digital Logic Design

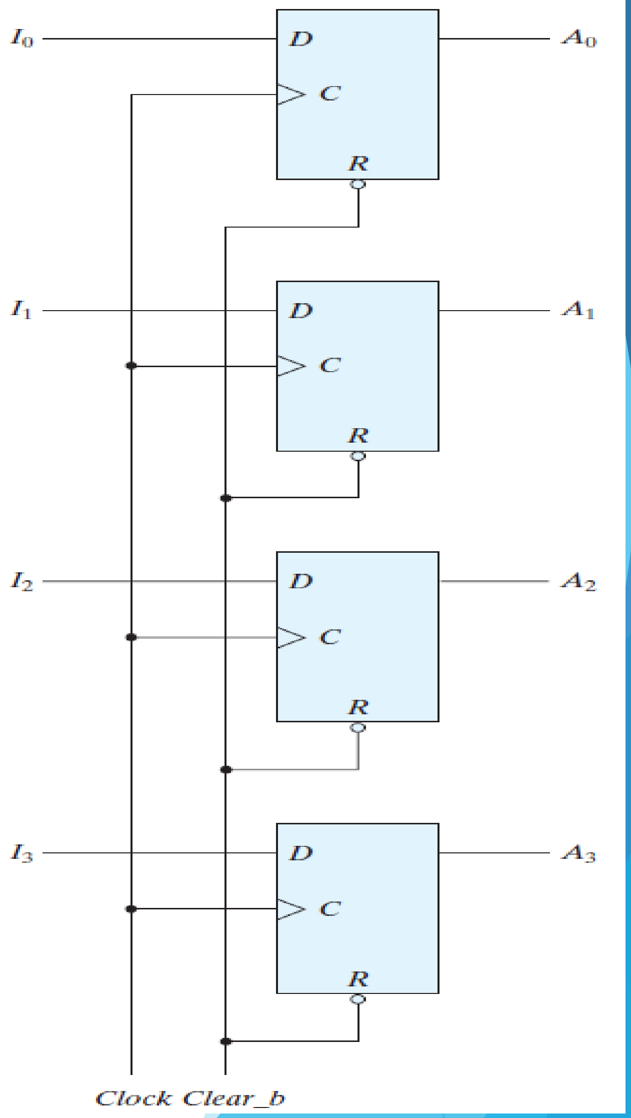
CSE 2105

Registers

- 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.

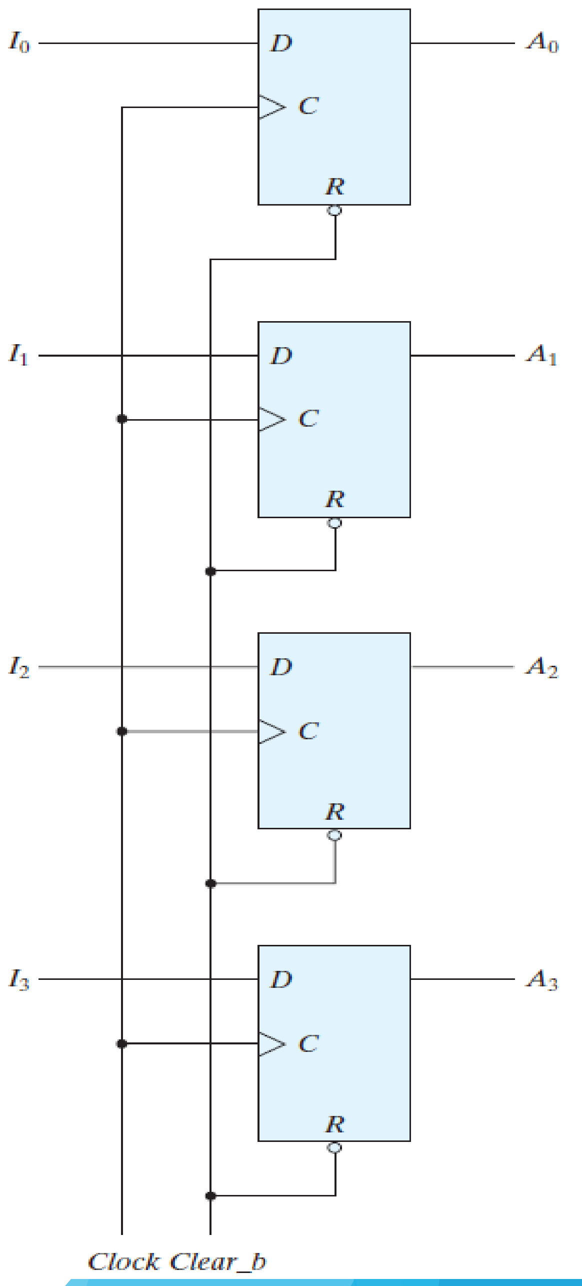
Four-bit register

- Various types of registers are available commercially. The simplest register is one that consists of only flip-flops, without any gates. The figure 1 shows 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 (I_3 , I_2 , I_1 , I_0) immediately before the clock edge determines the value of (A_3 , A_2 , A_1 , A_0) after the clock edge.



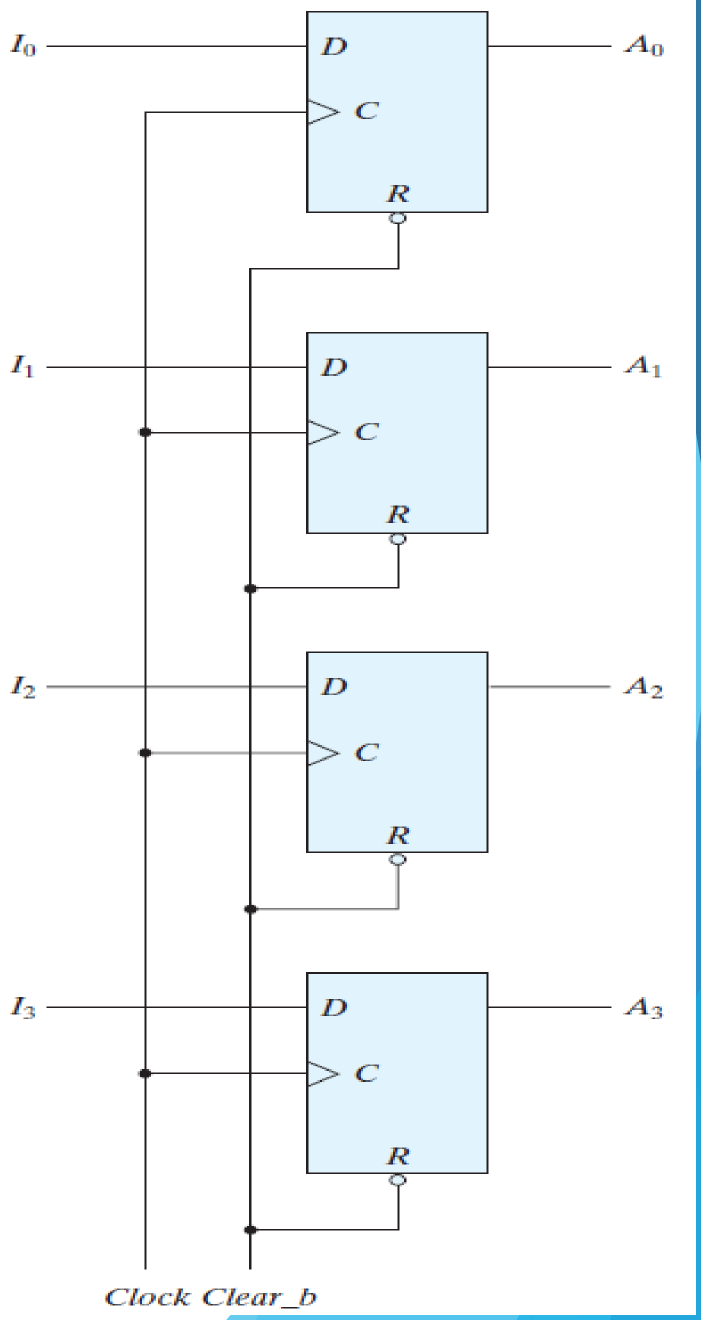
Four-bit register

- The four outputs can be sampled at any time to obtain the binary information stored in the register.
- 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.
- 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 logic 1 (i.e., de-asserted) during normal clocked operation.
- Note that, depending on the flip-flop, either **Clear**, **Clear_b**, **reset**, or **reset_b** can be used to indicate the transfer of the register to an all 0's state.



Four-bit Register with Parallel Load

- Registers with parallel load are a fundamental building block in digital systems.
- Synchronous digital systems have a master clock generator that supplies a continuous train of clock pulses. 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.
- A separate control signal must be used to decide which register operation will execute at each clock pulse. 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.

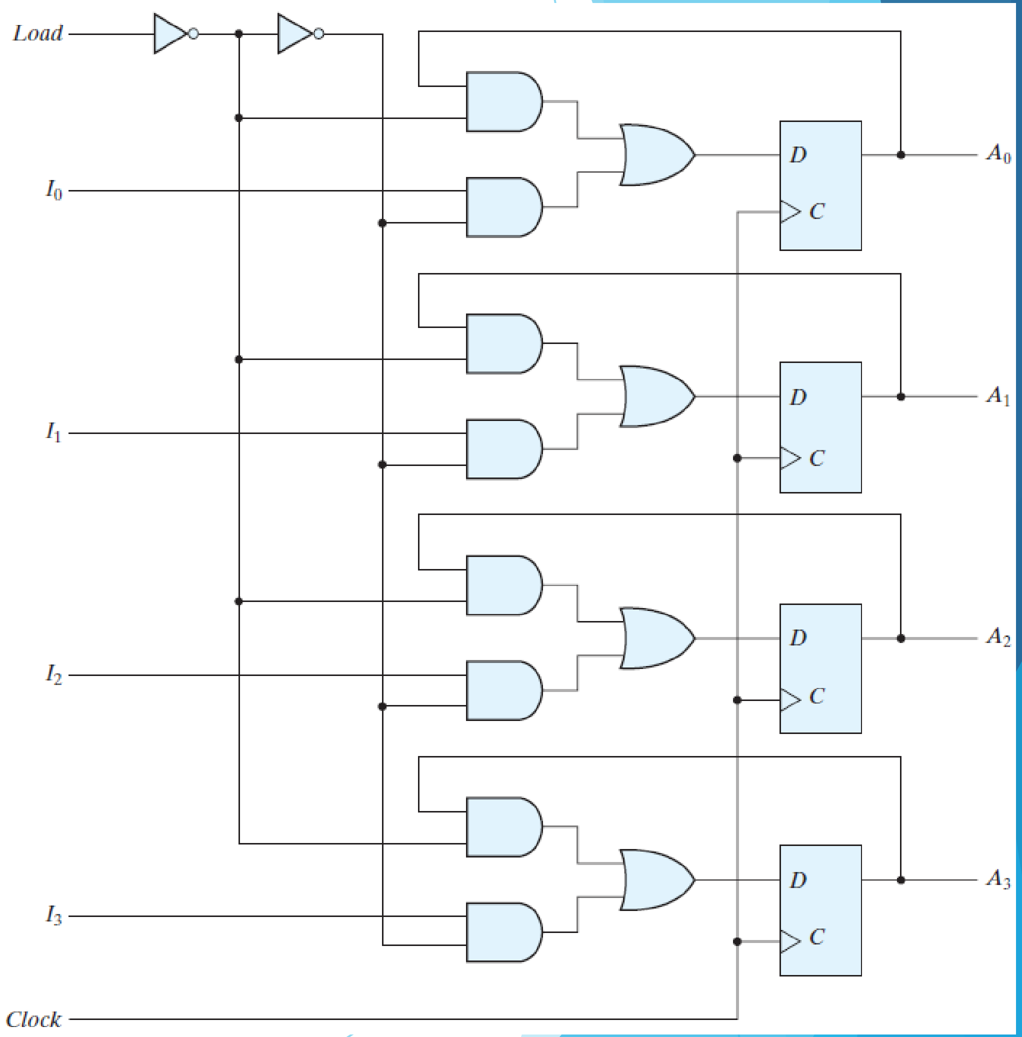


Four-bit Register with Parallel Load

- 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.
- In this case, the data bus driving the register would be unavailable for other traffic.
- For this reason, it is advisable to control the operation of the register with the D inputs with a parallel load line. This creates the effect of a gated clock.

Four-bit Register with Parallel Load

- A four-bit data-storage register with a load control input is shown in the figure.
- The load input to the register determines the action to be taken with each clock pulse. When the load input is 1, the data at the four external inputs are transferred into the register with the next positive edge of the clock.
- When the load input is 0, the outputs of the flip-flops are connected to their respective inputs.



Four-bit Register with Parallel Load

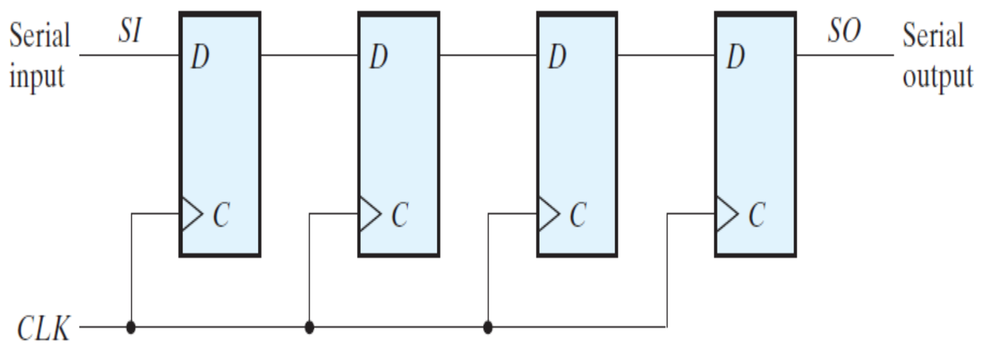
- To leave the output unchanged, it is necessary to make the *D* input equal to the present value of the output (i.e., the output circulates to the input at each clock pulse). The clock pulses are applied to the clock inputs without interruption. The load input determines whether the next pulse will accept new information or leave the information in the register intact. 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.

Application of Registers

1. The main application of register is storing data and address in digital form.
2. They hold temporary information that is needed quickly by the cpu. These include holding values while a calculation is performed, keeping track of an instruction pointer, storing the results of computations, and so on.
3. It can enable both serial and parallel data transfers, allowing logic operations to be performed on the data stored in it.
4. The registers are also used to make digital memory chips like ROM Chips, Flash Memory etc.
5. Cache memory in CPU is also made by registers.
6. Registers are used to communicate with other devices or components in the system, allowing them to transfer information quickly and efficiently.
7. They are also used to generate delay between input and output signal.

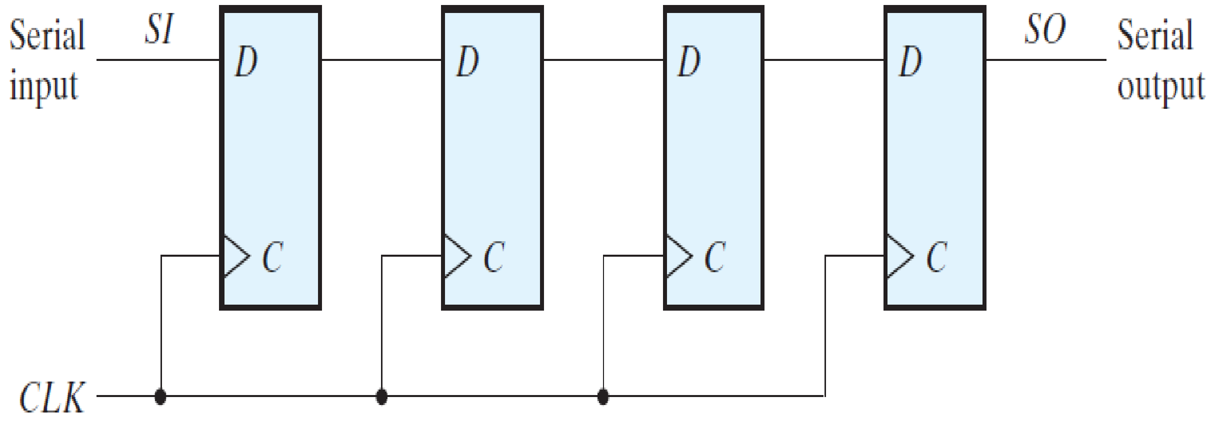
Shift Registers

- 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.
- The simplest possible shift register is one that uses only flip-flops, as shown in the following figure.
- The output of a given flip-flop is connected to the D input of the flip-flop at its right.

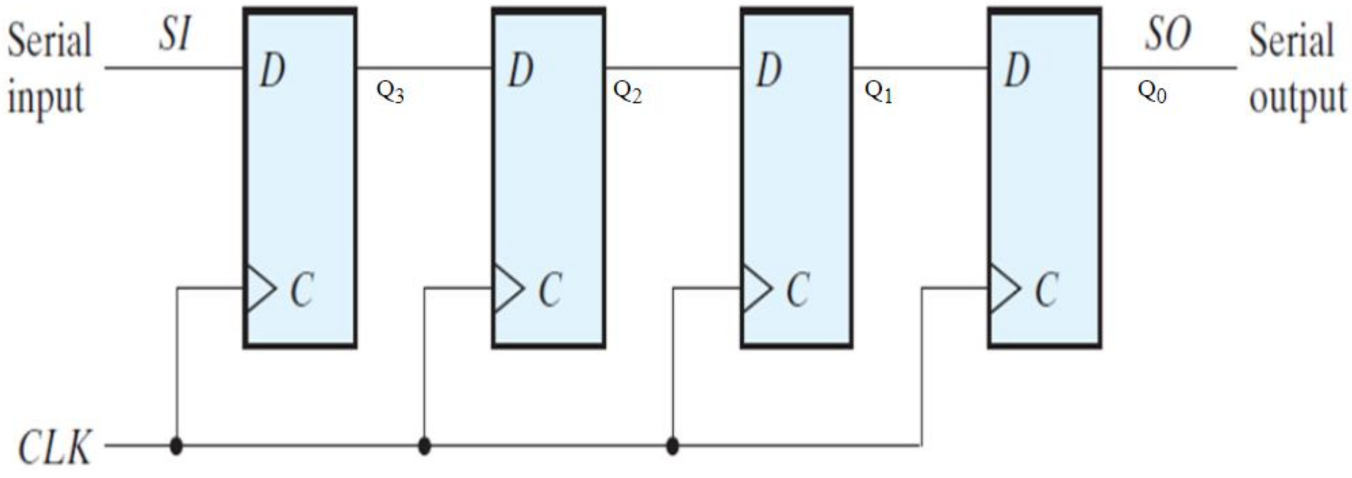


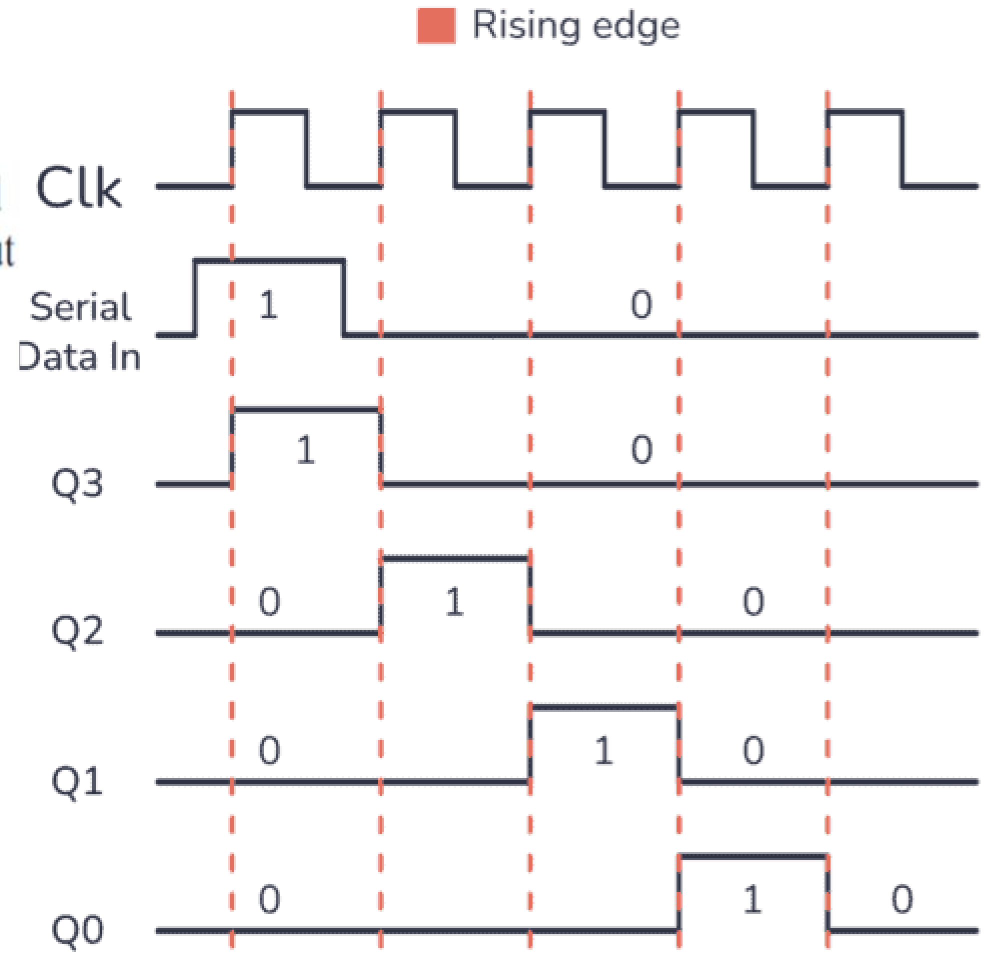
Shift Registers

- This shift register is unidirectional (left-to-right).
- Each clock pulse shifts the contents of the register one bit position to the right. The configuration does not support a left shift.
- 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. Sometimes it is necessary to control the shift so that it occurs only with certain pulses, but not with others.



Shift Registers



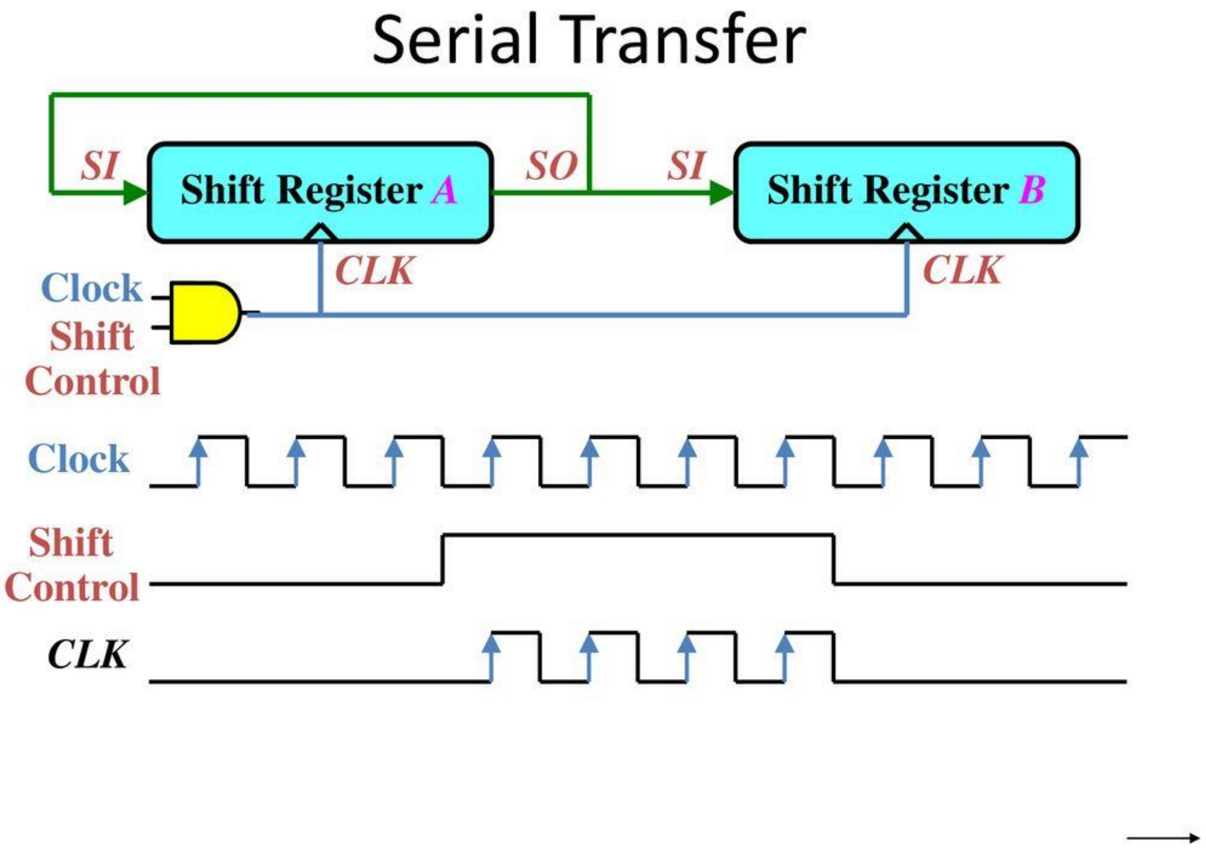


Serial Transfer/Shift

- The data path of a digital system is said to operate in serial mode when information is transferred one bit at a time.
- Information is transferred one bit at a time by shifting the bits out of the source register and into the destination register.
- This type of transfer is in contrast to parallel transfer, whereby all the bits of the register are transferred at the same time.

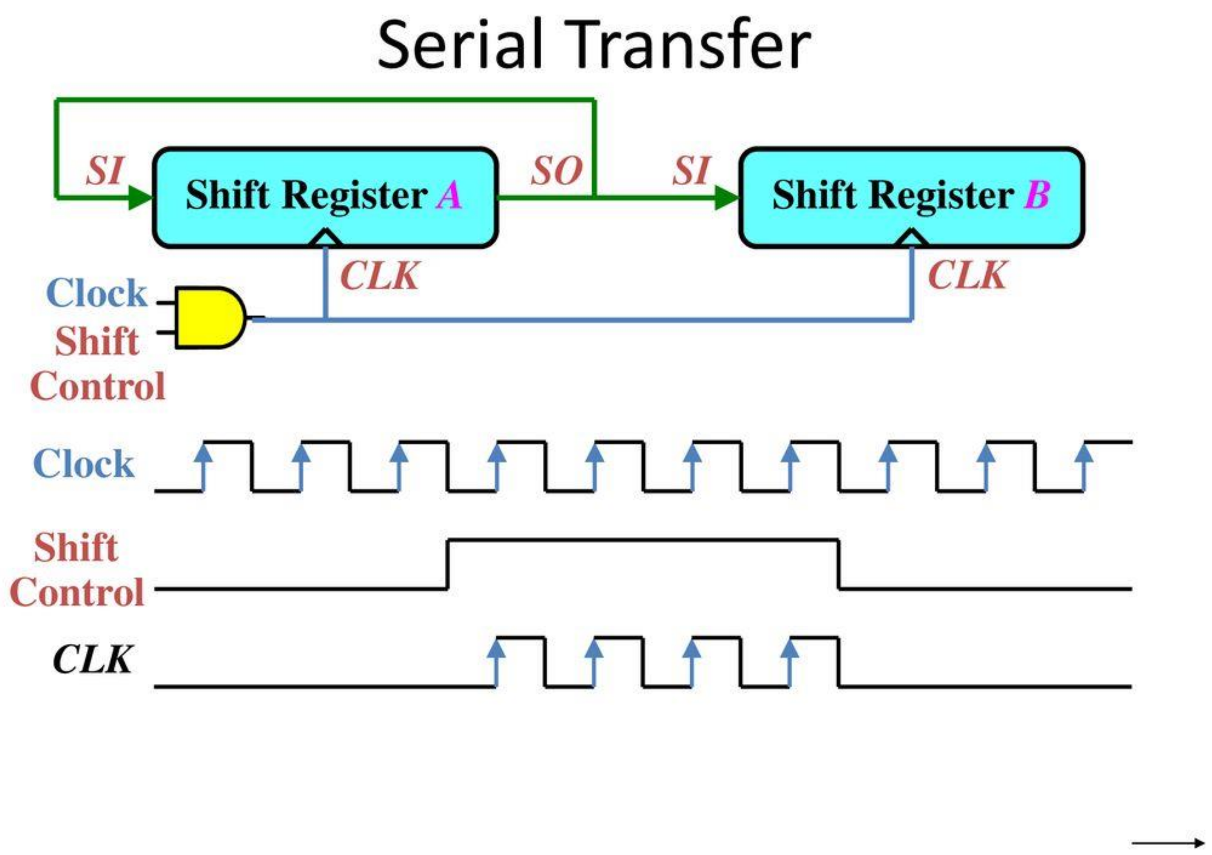
Serial Transfer/Shift

- The serial transfer of information from register A to register B is done with shift registers, as shown in the block diagram of the figure.
- 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.



Serial Transfer/Shift

- The initial content of register B is shifted out through its serial output and is lost unless it is transferred to a third shift register.
- The shift control input determines when and how many times the registers are shifted. For illustration here, this is done with an AND gate that allows clock pulses to pass into the CLK terminals only when the shift control is active..



Serial Transfer/Shift

- Suppose the shift registers in the figure have four bits each.
- Then the control unit that supervises the transfer of data must be designed in such a way that it enables the shift registers, through the shift control signal, for a fixed time of four clock pulses in order to pass an entire word.
- This design is shown in the timing diagram of the figure (b). The shift control signal is synchronized with the clock and changes value just after the negative edge of the clock. The next four clock pulses find the shift control signal in the active state, so the output of the AND gate connected to the CLK inputs produces four pulses: T_1, T_2, T_3, and T_4.
- Each rising edge of the pulse causes a shift in both registers. The fourth pulse changes the shift control to 0, and the shift registers are disabled.

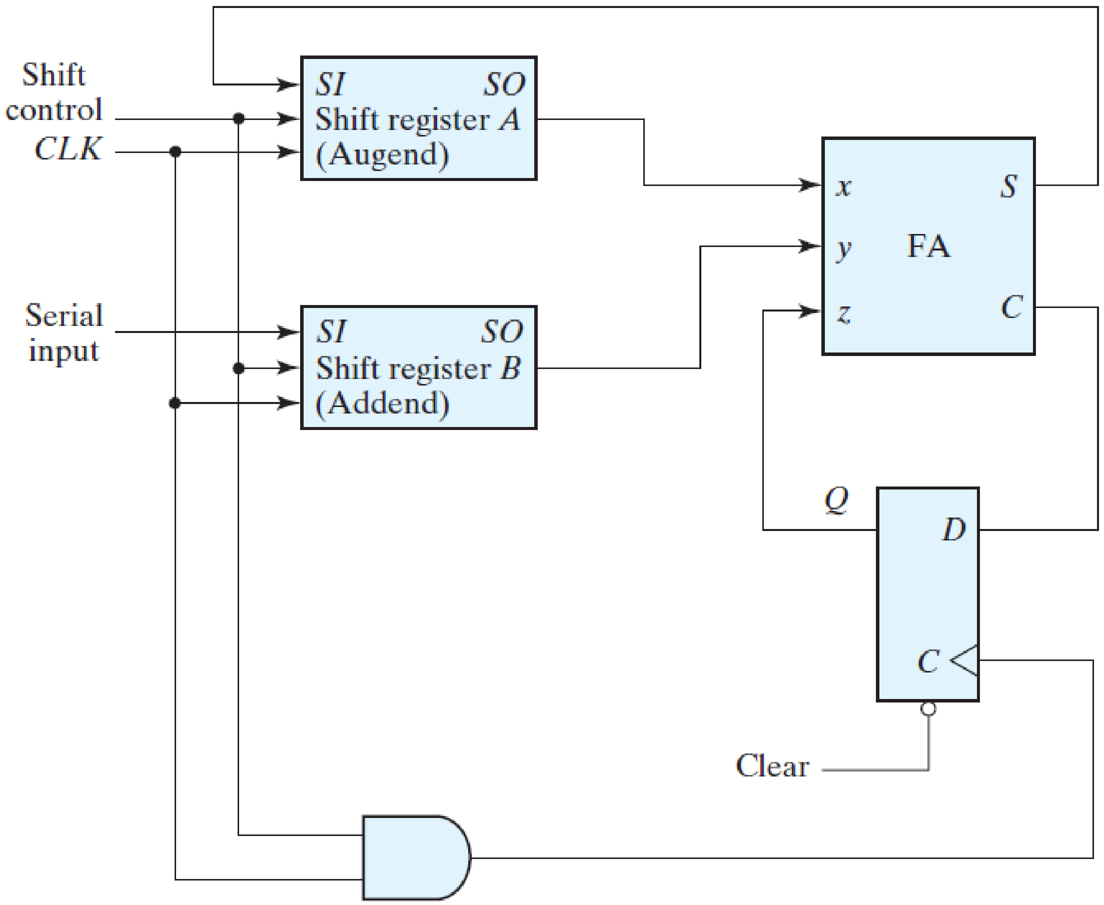
Serial Transfer/Shift

- Assume that the binary content of A before the shift is 1011 and that of B is 0010.
- The serial transfer from A to B occurs in four steps, as shown in the Table .
- With the first pulse, T_1, 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.
- The next three pulses perform identical operations, shifting the bits of A into B, one at a time.
- After the fourth shift, the shift control goes to 0, and registers A and B both have the value 1011. 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.

Timing Pulse	Shift Register A				Shift Register B			
Initial value	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0
	1	0	1	1	0	0	1	0
After T_1	A_0	A_3	A_2	A_1	A_0	B_3	B_2	B_1
	1	1	0	1	1	0	0	1
After T_2	A_1	A_0	A_3	A_2	A_1	A_0	B_3	B_2
	1	1	1	0	1	1	0	0
After T_3	A_2	A_1	A_0	A_3	A_2	A_1	A_0	B_3
	0	1	1	1	0	1	1	0
After T_4	A_3	A_2	A_1	A_0	A_3	A_2	A_1	A_0
	1	0	1	1	1	0	1	1

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 data path operation takes several clock cycles, but serial operations have the advantage of requiring fewer hardware components.
- The two binary numbers to be added serially are stored in two shift registers.
- Beginning with the least significant pair of bits, the circuit adds one pair at a time through a single full-adder (FA) circuit, as shown in Fig. The carry out of the full adder is transferred to a D flip-flop, the output of which is then used as the carry input for the next pair of significant bits.



Serial Addition

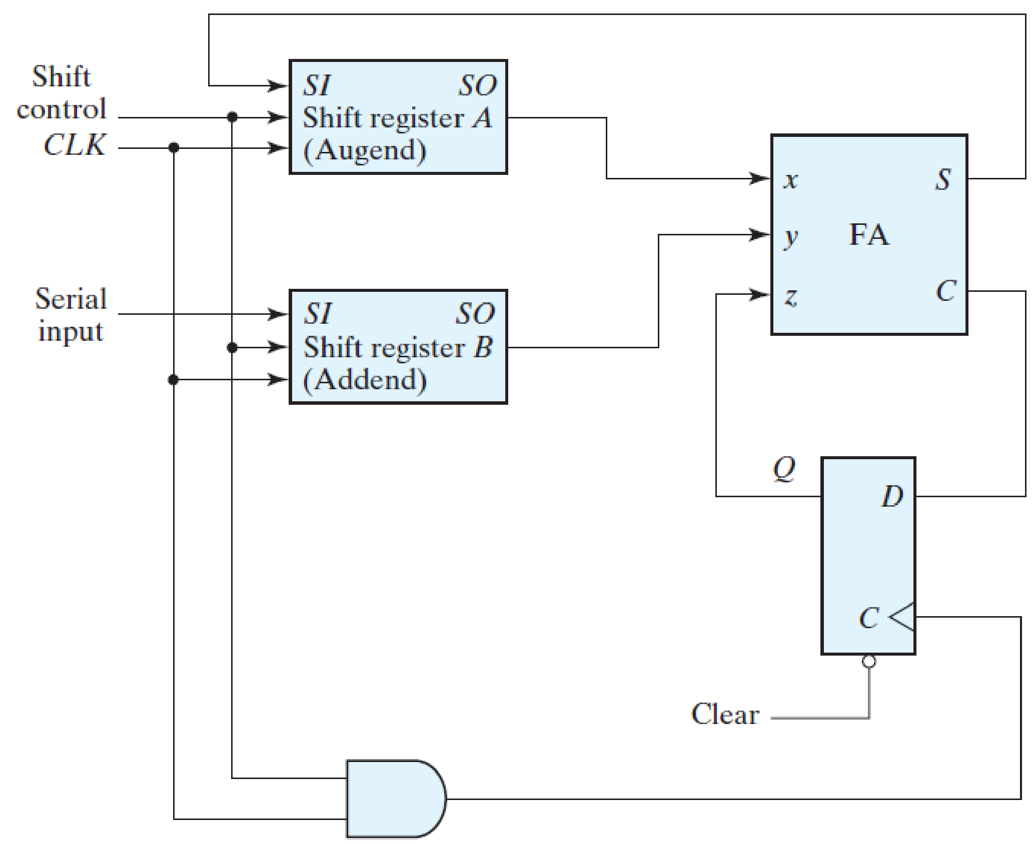
- The sum bit from the S output of the full adder could be transferred into a third shift register.
- By shifting the sum into A while the bits of A are shifted out, it is possible to use one register for storing both the augend and the sum bits.
- The serial input of register B can be used to transfer a new binary number while the addend bits are shifted out during the addition.
- The operation of the serial adder is as follows: 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.

Serial Addition

- 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.
- This process continues until the shift control is disabled. Thus, the addition is accomplished by passing each pair of bits together with the previous carry through a single full-adder circuit and transferring the sum, one bit at a time, into register A.

Serial Addition

- Suppose, we have to add the following two numbers~ 1101 and 1001.
- Shift Register A holds the augend (1101) and B holds the addend (1001)



Clock Pulse	A3	A2	A1	A0	B3	B2	B1	B0	x	y	z
	1	1	0	1	1	0	0	1	1	1	0
1st	0	1	1	0		1	0	0	0	0	1
2nd	1	0	1	1			1	0	1	0	0
3rd	1	1	0	1				1	1	1	0
4th	0	1	1	0							1