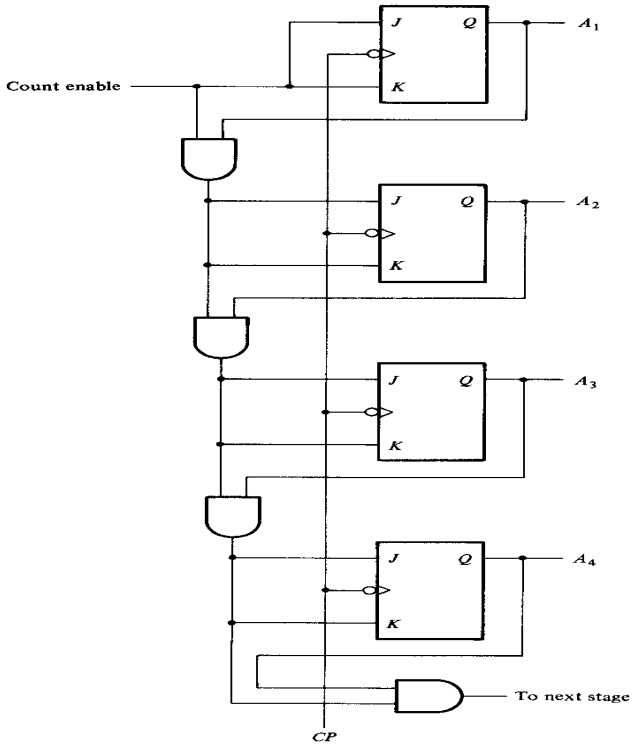
COUNTER

- A counter is a device which stores (and sometimes displays) the number of times a particular event or process has occurred. In electronics, counters can be implemented quite easily using register-type circuits.
- There are different types of counters, viz.
    - Asynchronous (ripple) counter
    - Synchronous counter
    - Decade counter
    - Up/down counter
    - Ring counter
    - Johnson counter
    - Cascaded counter
    - Modulus counter.

Synchronous counter

- A 4-bit synchronous counter using JK flip-flops is shown in the figure.
- In synchronous counters, the clock inputs of all the flip-flops are connected together and are triggered by the input pulses. Thus, all the flip-flops change state simultaneously (in parallel).

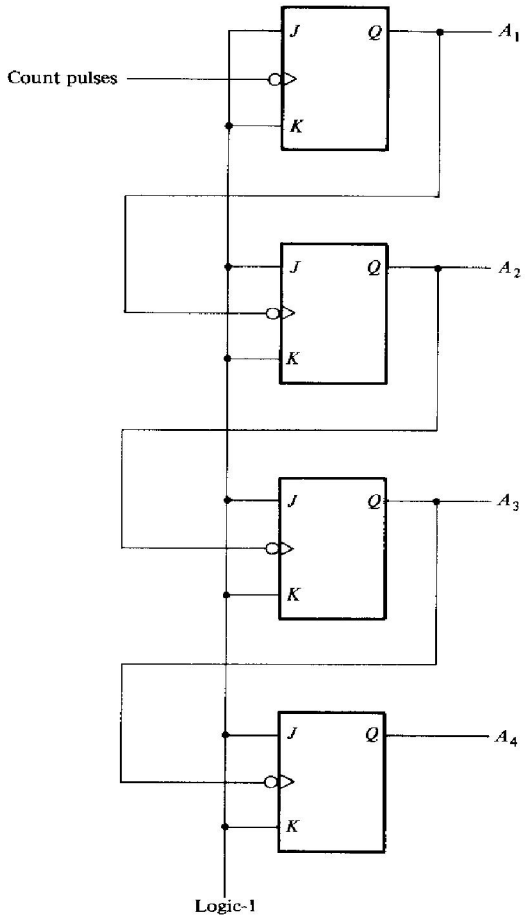


- The circuit below is a 4-bit synchronous counter.
- The J and K inputs of FF0 are connected to HIGH. FF1 has its J and K inputs connected to the output of FF0, and the J and K inputs of FF2 are connected to the output of an AND gate that is fed by the outputs of FF0 and FF1.
- A simple way of implementing the logic for each bit of an ascending counter (which is what is depicted in the image to the right) is for each bit to toggle when all of the less significant bits are at a logic high state.
- For example, bit 1 toggles when bit 0 is logic high; bit 2 toggles when both bit 1 and bit 0 are logic high; bit 3 toggles when bit 2, bit 1 and bit 0 are all high; and so on.

- Synchronous counters can also be implemented with hardware finite state machines, which are more complex but allow for smoother, more stable transitions.

Asynchronous Counter

- An asynchronous (ripple) counter is a single d-type flip-flop, with its J (data) input fed from its own inverted output.
- This circuit can store one bit, and hence can count from zero to one before it overflows (starts over from 0).



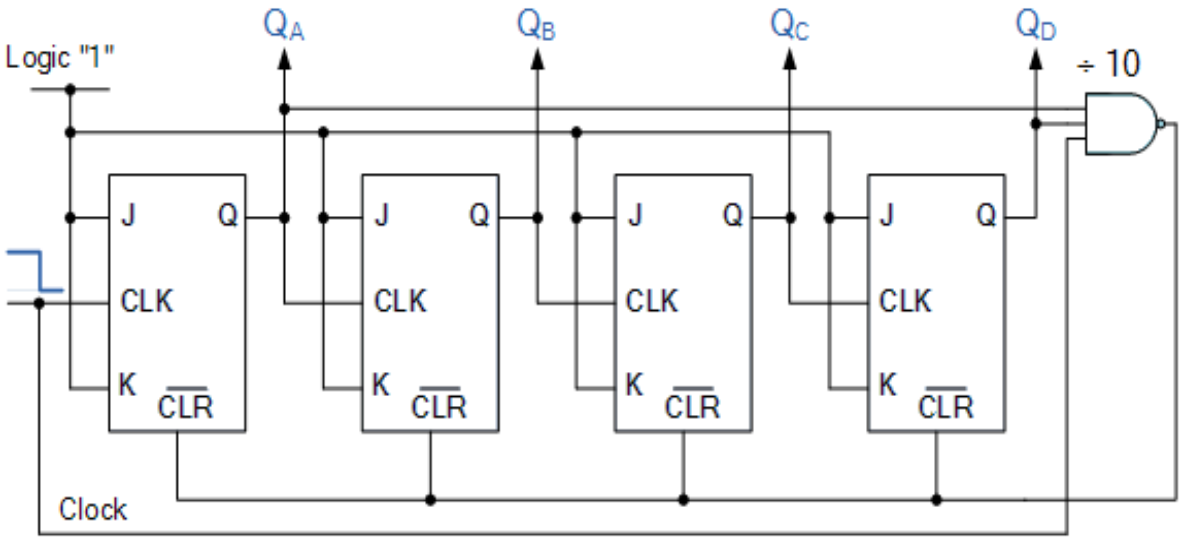
- This counter will increment once for every clock cycle and takes two clock cycles to overflow, so every cycle it will alternate between a transition from 0 to 1 and a transition from 1 to 0.
- This creates a new clock with a 50% duty cycle at exactly half the frequency of the input clock.
- If this output is then used as the clock signal for a similarly arranged D flip-flop, remembering to invert the output to the input, one will get another 1 bit counter that counts half as fast. These together yield a two-bit counter.
- Additional flip-flops can be added, by always inverting the output to its own input, and using the output from the previous flip-flop as the clock signal. The result is called a ripple counter, which can count to $2^n - 1$, where n is the number of bits (flip-flop stages) in the counter.
- Ripple counters suffer from unstable outputs as the overflows "ripple" from stage to stage, but they find application as dividers for clock signals.

Modulus Counter

- A modulus counter is that which produces an output pulse after a certain number of input pulses is applied.
- In modulus counter the total count possible is based on the number of stages, i.e., digit positions.

- Modulus counters are used in digital computers.
- A binary modulo-8 counter with three flip-flops, i.e., three stages, will produce an output pulse, i.e., display an output one-digit, after eight input pulses have been counted, i.e., entered or applied. This assumes that the counter started in the zero-condition.

Asynchronous Decade Counter



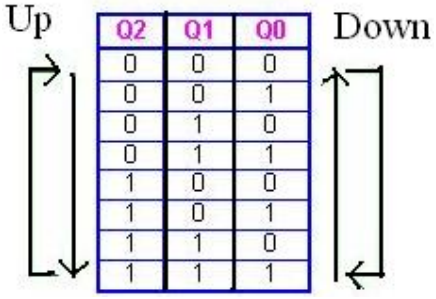
- A decade counter can count from BCD “0” to BCD “9”.
- A decade counter requires resetting to zero when the output count reaches the decimal value of 10, ie. when DCBA = 1010 and this condition is fed back to the reset input.
- A counter with a count sequence from binary “0000” (BCD = “0”) through to “1001” (BCD = “9”) is generally referred to as a BCD binary-coded-decimal counter because its ten state sequence is that of a BCD code but binary decade counters are more common.
- This type of asynchronous counter counts upwards on each leading edge of the input clock signal starting from 0000 until it reaches an output 1001 (decimal 9).
- Both outputs Q_A and Q_D are now equal to logic “1” and the output from the NAND gate changes state from logic “1” to a logic “0” level and whose output is also connected to the CLEAR (CLR) inputs of all the J-K Flip-flops.
- This signal causes all of the Q outputs to be reset back to binary 0000 on the count of 10. Once QA and QD are both equal to logic “0” the output of the NAND gate returns back to a logic level “1” and the counter restarts again from 0000. We now have a decade or Modulo-10 counter.

Decade Counter Truth Table

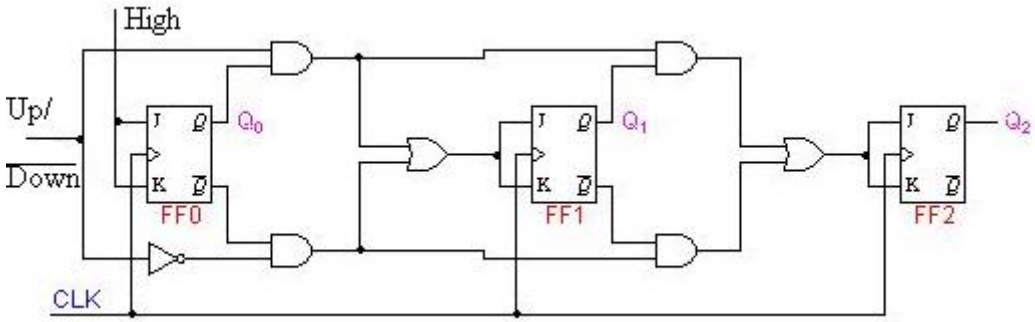
Clock Count	Output bit Pattern				Decimal Value
	QD	QC	QB	QA	
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	Counter Resets its Outputs back to Zero				

Up/Down Counter

- In a synchronous up-down binary counter the flip-flop in the lowest-order position is complemented with every pulse.
- A flip-flop in any other position is complemented with a pulse, provided all the lower-order pulse equal to 0.
- Up/Down counter is used to control the direction of the counter through a certain sequence.



- From the sequence table we can observe that:
  - For both the UP and DOWN sequences, Q_0 toggles on each clock pulse.
  - For the UP sequence, Q_1 changes state on the next clock pulse when Q_0=1.
  - For the DOWN sequence, Q_1 changes state on the next clock pulse when Q_0=0.
  - For the UP sequence, Q_2 changes state on the next clock pulse when Q_0=Q_1=1.
  - For the DOWN sequence, Q_2 changes state on the next clock pulse when Q_0=Q_1=0.



- These characteristics are implemented with the AND, OR & NOT logic connected as shown in the logic diagram above.

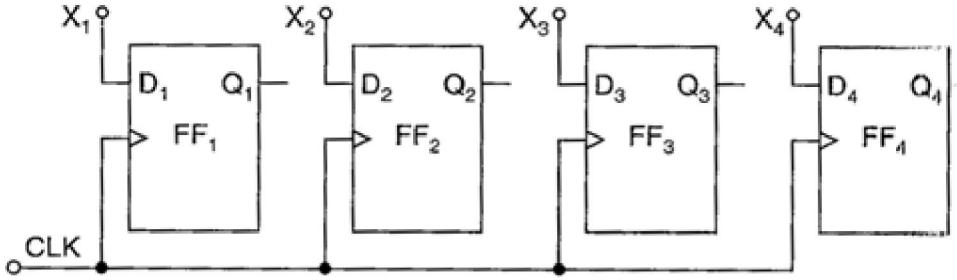
REGISTERS

INTRODUCTION:-

- The sequential circuits known as register are very important logical block in most of the digital systems.
- Registers are used for storage and transfer of binary information in a digital system.
- A register is mostly used for the purpose of storing and shifting binary data entered into it from an external source and has no characteristics internal sequence of states.
- The storage capacity of a register is defined as the number of bits of digital data, it can store or retain.
- These registers are normally used for temporary storage of data.

BUFFER REGISTER:-

- These are the simplest registers and are used for simply storing a binary word.
- These may be controlled by Controlled Buffer Register.
- D flip – flops are used for constructing a buffer register or other flip- flop can be used.
- The figure shown below is a 4- bit buffer register.



Logic diagram of a 4-bit buffer register.

- The binary word to be stored is applied to the data terminals.
- When the clock pulse is applied, the output word becomes the same as the word applied at the input terminals, i.e. the input word is loaded into the register by the application of clock pulse.
- When the positive clock edge arrives, the stored word becomes:

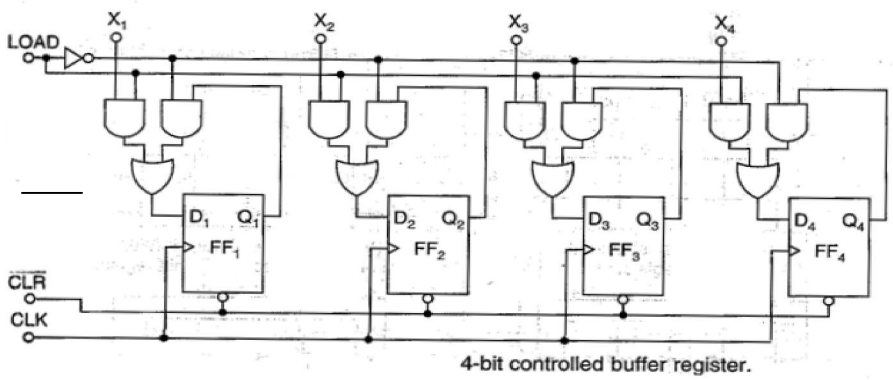
  Q4 Q3 Q2 Q1= X4 X3 X2 X1

  or Q = X .

  This circuit is too primitive to be of any use.

CONTROLLED BUFFER REGISTER:-

- The figure shows a controlled buffer register.



4-bit controlled buffer register.

- If $\overline{CLR}$ goes LOW, all the flip-flops are RESET and the output becomes, Q = 0000.
- When $\overline{CLR}$ is HIGH, the register is ready for action

- LOAD is control input.
- When LOAD is HIGH, the data bits X can reach the D inputs of FFs.
- At the positive going edge of the next clock pulse, the register is loaded, i.e.

  Q4 Q3 Q2 Q1= X4 X3 X2 X1

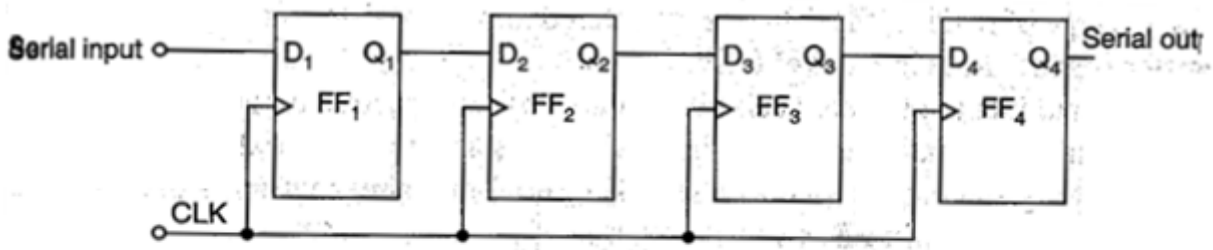
  or Q = X .
- When LOAD is LOW, the X bits cannot reach the FFs. At the same time the inverted signal $\overline{\text{LOAD}}$ is HIGH. This forces each flip-flop output to feedback to its data input.
- Therefore data is circulated or retained as each clock pulse arrives.
- In other words the content register remains unchanged in spite of the clock pulses.
- Longer buffer registers can built by adding more FFs.

CONTROLLED BUFFER REGISTER:-

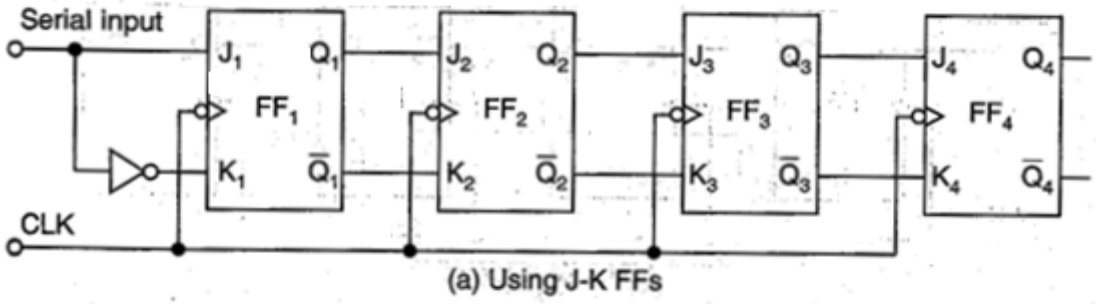
- A number of FFs connected together such that data may be shifted into and shifted out of them is called a shift register.
- Data may be shifted into or out of the register either in serial form or in parallel form.
- There are four basic types of shift registers
  1. Serial in, serial out
  2. Serial in, parallel out
  3. Parallel in, serial out
  4. Parallel in , parallel out

SERIAL IN, SERIAL OUT SHIFT REGISTER:-

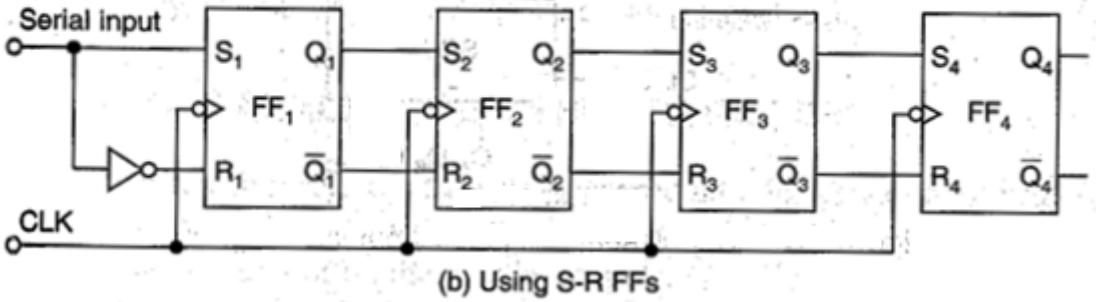
- This type of shift register accepts data serially, i.e., one bit at a time and also outputs data serially.
- The logic diagram of a four bit serial in, serial out shift register is shown in below figure:
- In 4 stages i.e. with 4 FFs, the register can store upto 4 bits of data.
- Serial data is applied at the D input of the first FF. The Q output of the first FF is connected to the D input of the second FF, the output of the second FF is connected to the D input of the third FF and the Q output of the third FF is connected to the D input of the fourth FF. The data is outputted from the Q terminal of the last FF.
- When a serial data is transferred to a register, each new bit is clocked into the first FF at the positive going edge of each clock pulse.
- The bit that is previously stored by the first FF is transferred to the second FF.
- The bit that is stored by the second FF is transferred to the third FF, and so on.
- The bit that was stored by the last FF is shifted out.
- A shift register can also be constructed using J-K FFs or S-R FFs as shown in the figure below.



4-bit serial-in, serial-out, shift-right, shift register.



(a) Using J-K FFs

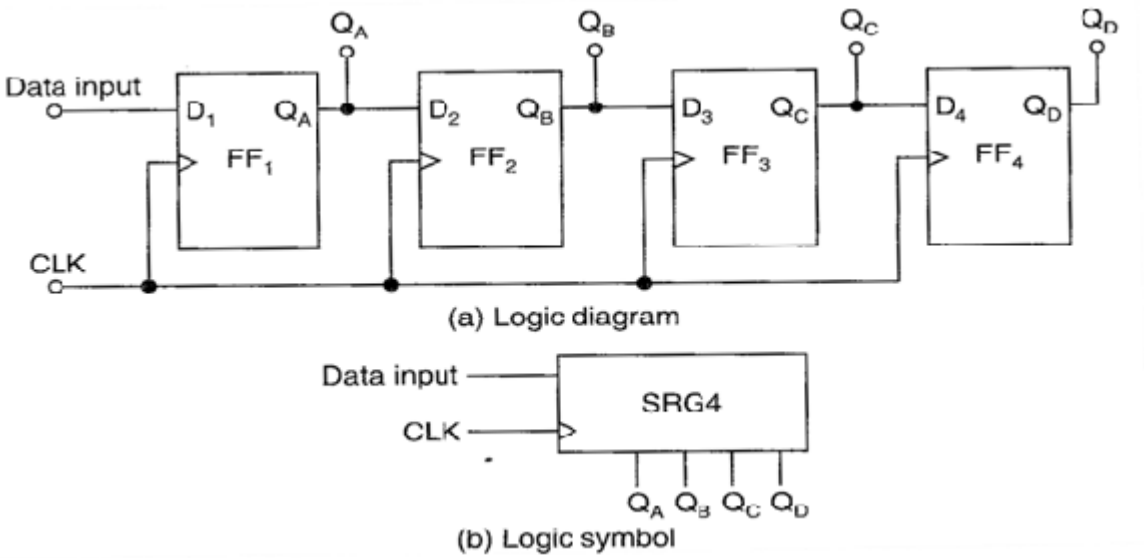


(b) Using S-R FFs

A 4-bit serial-in, serial-out, shift register.

SERIAL IN, PARALLEL OUT SHIFT REGISTER:-

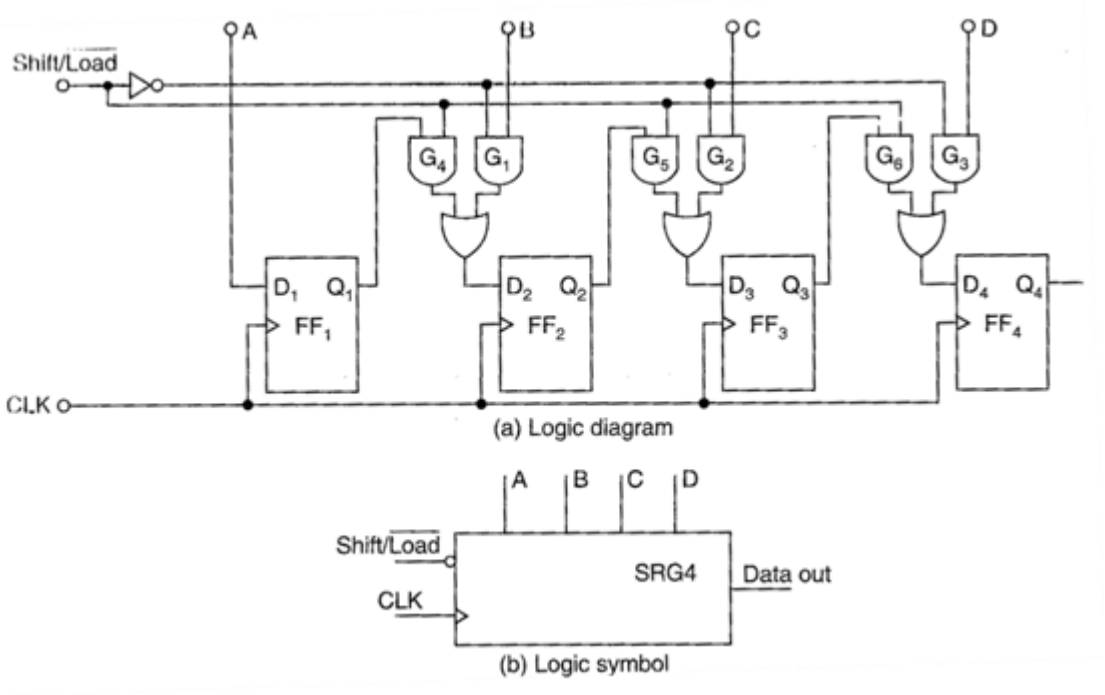
- In this type of register, the data bits are entered into the register serially, but the data stored in the register serially, but the stored in the register is shifted out in the parallel form.
- When the data bits are stored once, each bits appears on its respective output line and all bits are available simultaneously, rather than bit – by – bit basis as in the serial output.
- The serial in, parallel out shift register can be used as a serial in, serial out shift register if the output is taken from the Q terminal of the last FF.
- The logic diagram and logic symbol of a 4 bit serial in, parallel out shift register is given below.



A 4- bit serial in, parallel out shift register

PARALLEL IN, SERIAL OUT SHIFT REGISTER:-

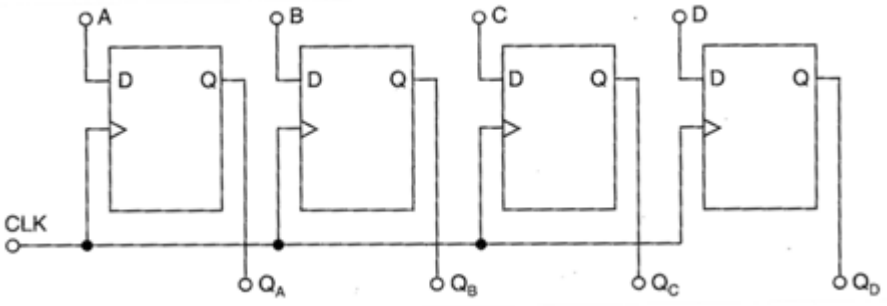
- For parallel in, serial out shift register the data bits are entered simultaneously into their respective stages on parallel lines, rather than on bit by bit basis on one line as with serial data inputs, but the data bits are transferred out of the register serially, i.e., on a bit by bit basis over a single line.
- The logic diagram and logic symbol of 4 bit parallel in, serial out shift register using D FFs is shown below.
- There are four data lines A, B, C and D through which the data is entered into the register in parallel form.
- The signal Shift /$\overline{\text{LOAD}}$ allows
    1. The data to be entered in parallel form into the register and
    2. The data to be shifted out serially from terminal Q_4.
- When Shift /$\overline{\text{LOAD}}$ line is HIGH, gates G1, G2, and G3 are disabled, but gates G4, G5 and G6 are enabled allowing the data bits to shift right from one stage to next.
- When Shift /$\overline{\text{LOAD}}$ line is LOW, gates G4, G5 and G6 are disabled, whereas gates G1, G2 and G3 are enabled allowing the data input to appear at the D inputs of the respective FFs.
- When clock pulse is applied, these data bits are shifted to the Q output terminals of the FFs and therefore the data is inputted in one step.
- The OR gate allows either the normal shifting operation or the parallel data entry depending on which AND gates are enabled by the level on the Shift /$\overline{\text{LOAD}}$ input.



A 4- bit parallel in, serial out shift register

PARALLEL IN, PARALLEL OUT SHIFT REGISTER:-

- In a parallel in, parallel out shift register, the data entered into the register in parallel form and also the data taken out of the register in parallel form. Immediately following the simultaneous entry of all data bits appear on the parallel outputs.
- The figure shown below is a 4 bit parallel in parallel out shift register using D FFs.
- Data applied to the D input terminals of the FFs.
- When a clock pulse is applied at the positive edge of that pulse, the D inputs are shifted into the Q outputs of the FFs.
- The register now stores the data.
- The stored data is available instantaneously for shifting out in parallel form.

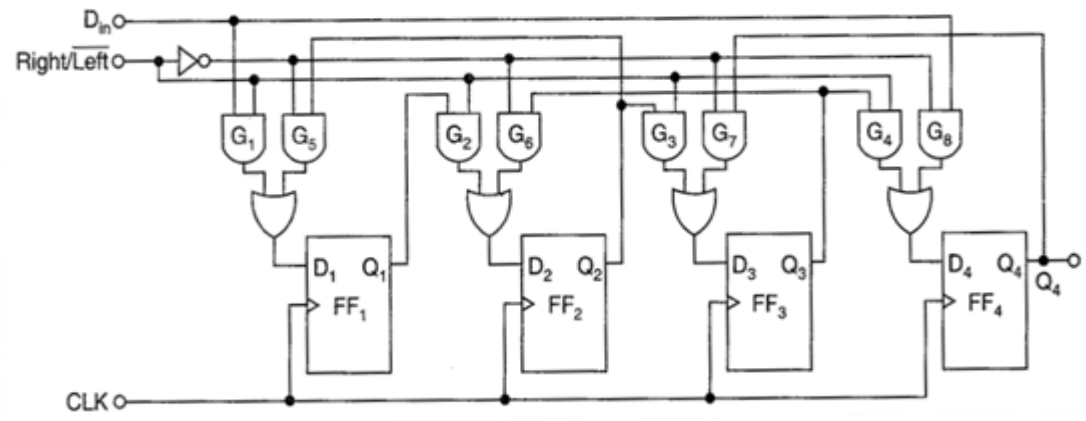


Logic diagram of a 4 – bit parallel in, parallel out shift register

BIDIRECTIONAL SHIFT REGISTER:-

- In bidirectional shift register is one in which the data bits can be shifted from left to right or from right to left.
- The figure shown below the logic diagram of a 4 bit serial in, serial out, bidirectional (shift-left, shift-right) shift register.
- Right /Left is the mode signal. When Right /Left is a 1, the logic circuit works as a shift right shift register. When Right /Left is a 0, the logic circuit works as a shift right shift register.

- The bidirectional is achieved by using the mode signal and two AND gates and one OR gate for each stage.
- A HIGH on the Right/$\overline{\text{Left}}$ control input enables the AND gates G_1, G_2, G_3 and G_4 and disables the AND gates G_5, G_6, G_7 and G_8 and the state of Q output of each FF is passed through the gate to the D input of the following FF. When clock pulse occurs, the data bits are effectively shifted one place to the right.
- A LOW Right/$\overline{\text{Left}}$ control input enables the AND gates G_5, G_6, G_7 and G_8 and disables the AND gates G_1, G_2, G_3 and G_4 and the Q output of each FF is passed to the D input of the preceding FF. When clock pulse occurs the data bits are then effectively shifted one place to the left.
- So, the circuit works as a bidirectional shift register.



Logic diagram of 4- bit bidirectional shift register

UNIVERSAL SHIFT REGISTERS:-

- The register which has both shifts and parallel load capabilities, it is referred as a universal shift register. So, universal shift register is a bidirectional register, whose input can be either in serial form or in parallel form and whose output also can be either in serial form or parallel form.
- The universal shift register can be realized using multiplexers.
- The figure shows the logic diagram of a 4 bit universal shift register that has all the capabilities of a general shift register.

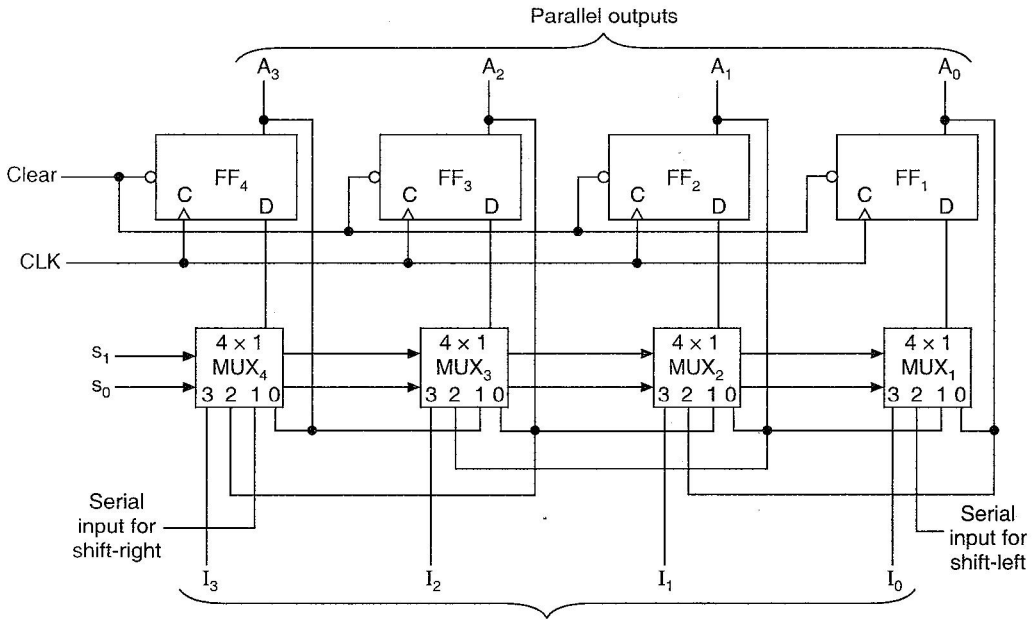


Fig- (a) 4 bit universal shift register

- It consists of four D flip- flops and four multiplexers.
- The four multiplexers have two common selection inputs S_1 and S_0.
- Input 0 in each multiplexer is selected when S_1S_0 = 00, input 1 is selected when S_1S_0 = 01, and input 2 is selected when S_1S_0 = 10 and input 3 is selected when S_1S_0= 11.
- The selection inputs control the mode of operation of the register is according to the function entries shown in the table.
- When S_1S_0 = 00 the present value of the register is applied to the D inputs of flip-flops. This condition forms a path from the output of each FF into the input of the same FF.
- The next clock edge transfers into each FF the binary value it held previously, and no change of state occurs.
- When S_1S_0 = 01, terminal 1 of the multiplexer inputs have a path of the D inputs of the flip- flops. This causes a shift right operation, with serial input transferred into FF_4.
- When S_1S_0 = 10 a shift left operation results with the other serial input going into the FF_1.
- Finally when S_1S_0 = 11, the binary information on the parallel input lines is transferred into the register simultaneously during the next clock edge.

Functional table for the register of fig – a:

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

APPLICATIONS OF SHIFT REGISTERS:-

1. **Time delays:**
   - In digital systems, it is necessary to delay the transfer of data until the operation of the other data have been completed, or to synchronize the arrival of data at a subsystem where it is processed with other data.
   - A shift register can be used to delay the arrival of serial data by a specific number of clock pulses, since the number of stages corresponds to the number of clock pulses required to shift each bit completely through the register.
   - The total time delay can be controlled by adjusting the clock frequency and by the number of stages in the register.
   - In practice, the clock frequency is fixed and the total delay can be adjusted only by controlling the number of stages through which the data is passed.
2. **Serial / Parallel data conversion:**
   - Transfer of data in parallel form is much faster than that in serial form.
   - Similarly the processing of data is much faster when all the data bits are available simultaneously. Thus in digital systems in which speed is important so to operate on data parallel form is used.
   - When large data is to be transmitted over long distances, transmitting data on parallel lines is costly and impracticable.
   - It is convenient and economical to transmit data in serial form, since serial data transmission requires only one line.

- Shift registers are used for converting serial data to parallel form, so that a serial input can be processed by a parallel system and for converting parallel data to serial form, so that parallel data can be transmitted serially.
- A serial in, parallel out shift register can be used to perform serial-to parallel conversion, and a parallel in, serial out shift register can be used to perform parallel- to –serial conversion.
- A universal shift register can be used to perform both the serial- to – parallel and parallel-to-serial data conversion.
- A bidirectional shift register can be used to reverse the order of data.

RING AND JOHNSON COUNTER:-

- Ring counters are constructed by modifying the serial-in, serial-out, shift register.
- There are two types of ring counters
  i) Basic ring counter
  ii) Johnson counter
- The basic ring counter can be obtained from a serial-in serial- out shift register by connecting the Q output of the last FF to the D input of the first FF.
- The Johnson counter can be obtained from serial-in, serial- out, shift register by connecting the Q output of the last FF to the D input of the first FF.
- Ring counter outputs can be used as a sequence of synchronizing pulses.
- The ring counter is a decimal counter.