CMP 2203

Lecture 9

Sequential functional units : Shift Registers

- The **Shift Register** is another type of sequential logic circuit that can be used for the storage or the transfer of binary data
- This sequential device loads the data present on its inputs and then moves or “shifts” it to its output once every clock cycle, hence the name **Shift Register**.
- A shift register basically consists of several single bit “D-Type Data Latches”, one for each data bit, either a logic “0” or a “1”, connected together in a serial type daisy-chain arrangement so that the output from one data latch becomes the input of the next latch and so on.

Shift Registers

- Data bits may be fed in or out of a shift register serially, that is one after the other from either the left or the right direction, or all together at the same time in a parallel configuration.
- The number of individual data latches required to make up a single Shift Register device is usually determined by the number of bits to be stored with the most common being 8-bits (one byte) wide constructed from eight individual data latches.
- The directional movement of the data through a shift register can be either to the left, (**left shifting**) to the right, (**right shifting**) left-in but right-out, (**rotation**) or both left and right shifting within the same register thereby making it **bidirectional**.

Shift Registers

Applications

Shift Registers are used for data storage or for the movement of data and are therefore commonly used inside calculators or computers to store data such as two binary numbers before they are added together, or to convert the data from either a serial to parallel or parallel to serial format. The individual data latches that make up a single shift register are all driven by a common clock (Clk)signal making them synchronous devices.

Shift Registers

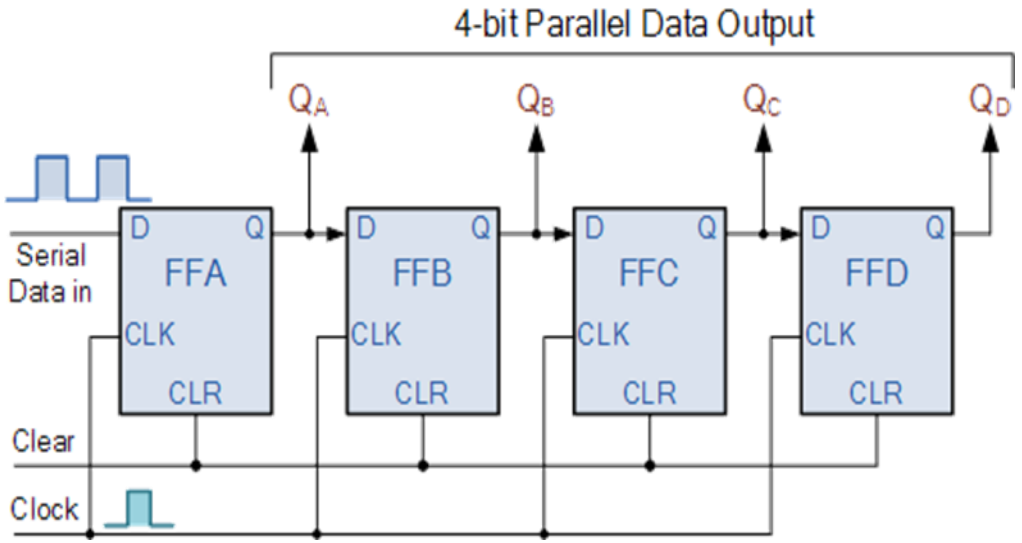
Shift registers operate in one of four different modes with the basic movement of data through a shift register being:

- **Serial-in to Parallel-out (SIPO)** - the register is loaded with serial data, one bit at a time, with the stored data being available at the output in parallel form.
- **Serial-in to Serial-out (SISO)** - the data is shifted serially “IN” and “OUT” of the register, one bit at a time in either a left or right direction under clock control.
- **Parallel-in to Serial-out (PISO)** - the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.
- **Parallel-in to Parallel-out (PIPO)** - the parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.

Serial-in to Parallel-out (SIPO) Shift Register

4-bit Serial-in to Parallel-out Shift Register

- Shift register IC's are generally provided with a clear or reset connection so that they can be "SET" or "RESET" as required.



Serial-in to Parallel-out (SIPO) Shift Register

- Assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D are at logic level “0” i.e., no parallel data output.
- If a logic “1” is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting QA will be set HIGH to logic “1” with all the other outputs still remaining LOW at logic “0”. Assume now that the DATA input pin of FFA has returned LOW again to logic “0” giving us one data pulse or 0-1-0.
- The second clock pulse will change the output of FFA to logic “0” and the output of FFBand Q_B HIGH to logic “1” as its input D has the logic “1” level on it from Q_A. The logic “1” has now moved or been “shifted” one place along the register to the right as it is now at Q_A.

Serial-in to Parallel-out (SIPO) Shift Register

- When the third clock pulse arrives this logic “1” value moves to the output of FFC (QC) and so on until the arrival of the fifth clock pulse which sets all the outputs QA to QD back again to logic level “0” because the input to FFA has remained constant at logic level “0”.
- The effect of each clock pulse is to shift the data contents of each stage one place to the right, and this is shown in the following table until the complete data value of 0-0-0-1 is stored in the register. This data value can now be read directly from the outputs of QA to QD.
- Then the data has been converted from a serial data input signal to a parallel data output. The truth table and following waveforms show the propagation of the logic “1” through the register from left to right as follows.

Serial-in to Parallel-out (SIPO) Shift Register

Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

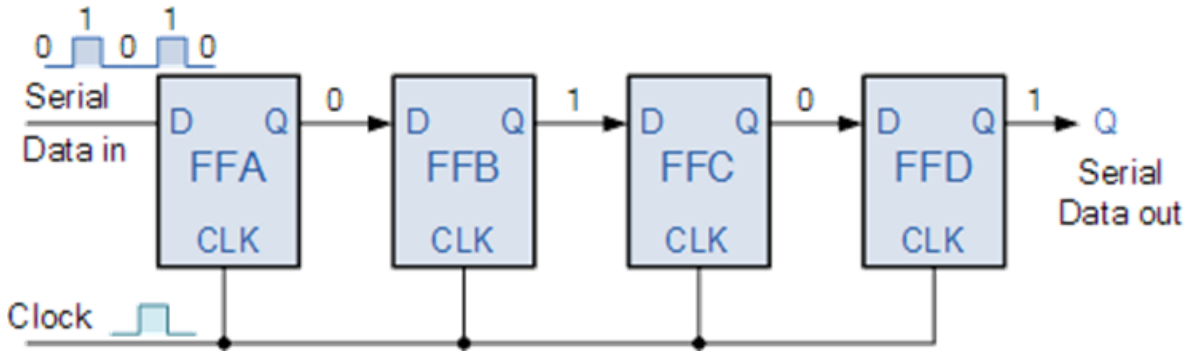
Figure showing basic movement through a shift register

Serial-in to Serial-out (SISO) Shift Register

- This shift register is very similar to the SIPO above, except were before the data was read directly in a parallel form from the outputs QA to QD, this time the data is allowed to flow straight through the register and out of the other end. Since there is only one output, the DATA leaves the shift register one bit at a time in a serial pattern, hence the name Serial-in to Serial-Out Shift Register or SISO.
- The SISO shift register is one of the simplest of the four configurations as it has only three connections, the serial input (SI) which determines what enters the left hand flip-flop, the serial output (SO) which is taken from the output of the right hand flip-flop and the sequencing clock signal (Clk)..

Serial-in to Serial-out (SISO) Shift Register

- **Application:** SISO Shift Register also acts as a temporary storage device or it can act as a time delay device for the data, with the amount of time delay being controlled by the number of stages in the register

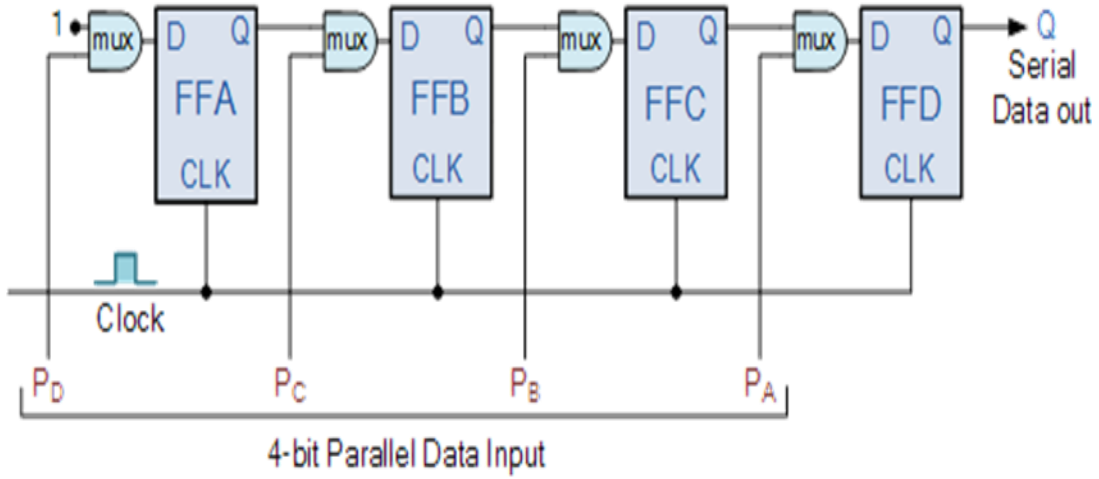


The logic circuit diagram below shows a generalized serial-in serial-out shift register

Parallel-in to Serial-out (PISO) Shift Register

- The Parallel-in to Serial-out shift register acts in the opposite way to the serial-in to parallel-out one above. The data is loaded into the register in a parallel format in which all the data bits enter their inputs simultaneously, to the parallel input pins PA to PD of the register. The data is then read out sequentially in the normal shift-right mode from the register at Q representing the data present at PA to PD.
- This data is outputted one bit at a time on each clock cycle in a serial format. It is important to note that with this type of data register a clock pulse is not required to parallel load the register as it is already present, but four clock pulses are required to unload the data.

Parallel-in to Serial-out (PISO) Shift Register



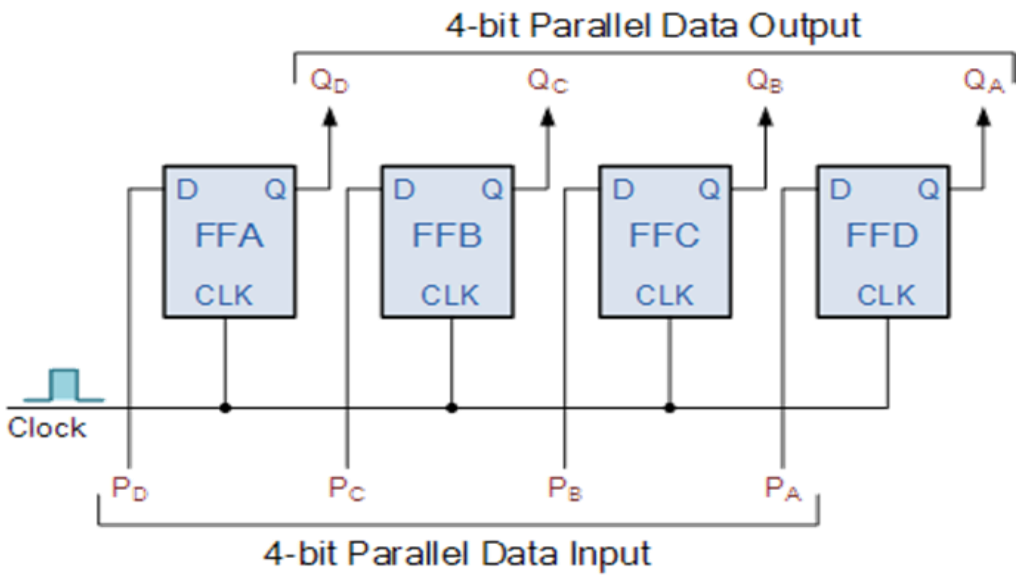
4-bit Parallel-in to Serial-out Shift Register

Application: Since this type of shift register converts parallel data, such as an 8-bit data word into serial format, it can be used to multiplex many different input lines into a single serial DATA stream which can be sent directly to a computer or transmitted over a communications line.

Parallel-in to Parallel-out (PIPO) Shift Register

- The final mode of operation is the Parallel-in to Parallel-out Shift Register. This type of shift register also acts as a temporary storage device or as a time delay device similar to the SISO configuration above. The data is presented in a parallel format to the parallel input pins PA to PD and then transferred together directly to their respective output pins QA to QA by the same clock pulse. Then one clock pulse loads and unloads the register.
- The PIPO shift register is the simplest of the four configurations as it has only three connections, the parallel input (PI) which determines what enters the flip-flop, the parallel output (PO) and the sequencing clock signal (Clk).

Parallel-in to Parallel-out (PIPO) Shift Register



4-bit Parallel-in to Parallel-out Shift Register

Application:

- Similar to the Serial-in to Serial-out shift register, this type of register also acts as a temporary storage device or as a time delay device, with the amount of time delay being varied by the frequency of the clock pulses.

- Also, in this type of register there are no interconnections between the individual flip-flops since no serial shifting of the data is required.

Ring counters

- Ring counter is a sequential logic circuit that is constructed using shift register. Same data recirculates in the counter depending on the clock pulse.
- Ring counters are of two types;
- 1)Ordinary Ring counters
- 2) Johnson counter

Ring counters

- The ring counter is a cascaded connection of flip flops, in which the output of last flip flop is connected to input of first flip flop. In ring counter if the output of any stage is 1, then its reminder is 0. The Ring counters transfers the same output throughout the circuit.
- That means if the output of the first flip flop is 1, then this is transferred to its next stage i.e. 2nd flip flop. By transferring the output to its next stage, the output of first flip flop becomes 0. And this process continues for all the stages of a ring counter. If we use **n** flip flops in the ring counter, the ‘1’ is circulated for every **n** clock cycles.

4 bit Ring Counter

- This is a Mod 4 ring counter which has 4 D flip flops connected in series. The clock signal is applied to clock input of each flip flop, simultaneously and the RESET pulse is applied to the CLR inputs of all the flip flops.

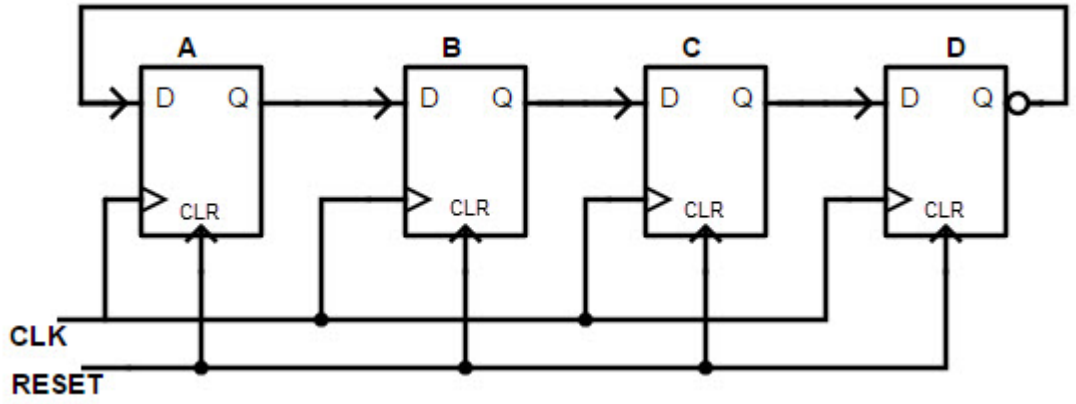
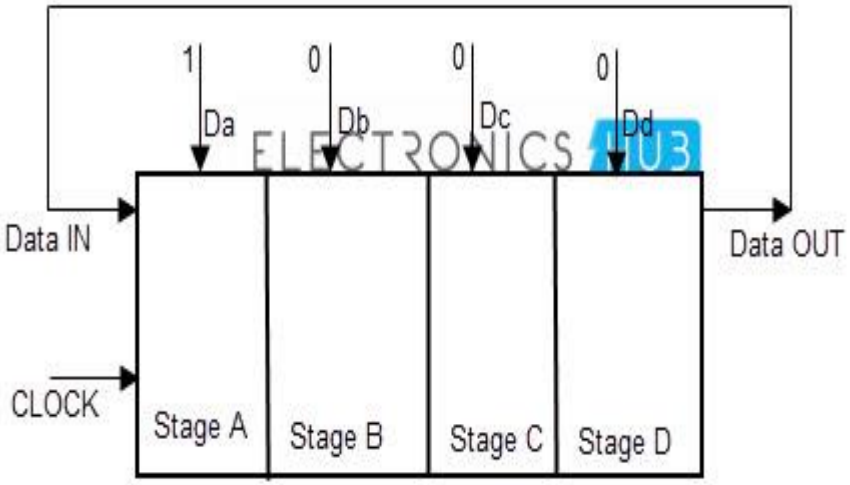


Figure showing 4 bit ring counter

Operation of Ring Counter

- Initially, all the flip flops in ring counter are reset to 0 by applying CLEAR signal. Before applying the clock pulse, we apply the PRESET pulse to the flip flops which assigns the value ‘1’ to the ring counter circuit. For each clock signal, the data circulates among all the 4 flip flop stages of ring counter.

- This 4 staged ring counter is called Mod 4 ring counter or 4 bit ring counter. To circulate the data correctly in the ring counter, we must load the counter with required values like all 0’s or all 1’s.



Circulation of data in Ring counters

- The ring counter is similar to that of the shift registers connected in series. The above diagram shown the four stages of flip flops as the parallel in serial our shift registers, with data inputs D0, D1, D2 and D3.The data circulation in ring counter is explained below. By passing the reset signal, initially the flip flops are at RESET state. When the PRESET is applied to the ring counter the input of the circuit becomes 1.
- This input is connected to the first flip flop in the series, so that the flip flop QA is set to 1 and all other outputs of remaining flip flops will be low.
- If we make the data input of the flip flop ‘A’ to low, this gives us the data pulse as 0 1 0. Then for the second clock signal, the output of first flip flop will again change and then the output of ‘B’ will become high. This means the data pulse 0 0 1 occurs.
- In this way, as the clock signal and input of first flip flop changes, the output of the other flip flops changes. As the output of last flip flop in series is connected to the input of the first flip flop, the data sequence rotates or circulates in the ring counter.

Truth table of Ring counters

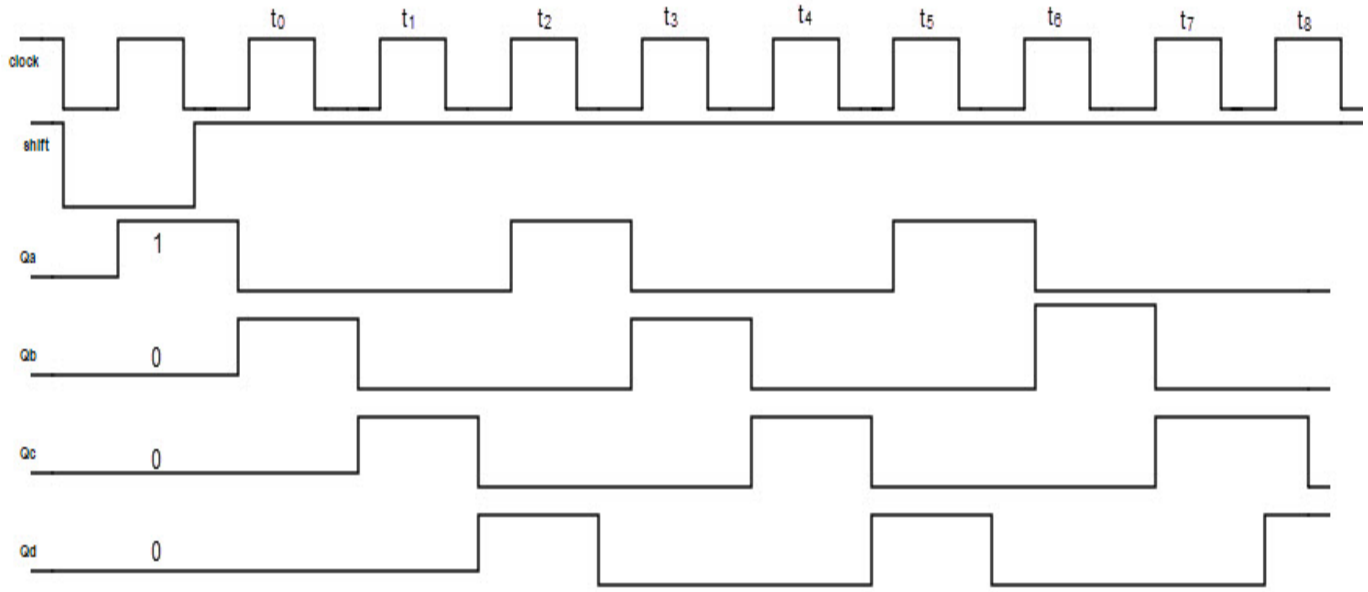
When CLEAR input CLR = 0, then all flip flops are set to 1. When CLEAR input CLR = 1, the ring counter starts its operation. For one clock signal, the counter starts its operation. On next clock signal, the counter again resets to 0000. Ring counter has 4 sequences: 0001, 0010, 0100, 1000, 000.

Q_0	Q_1	Q_2	Q_3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Figure of Truth table

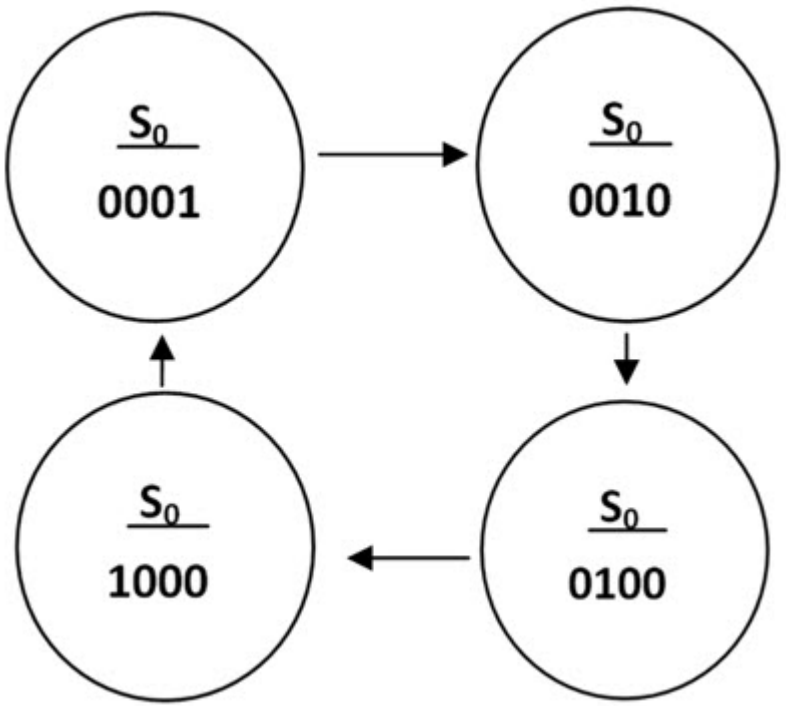
Timing diagram of Ring Counter

The clock signal changes the output of every stage of the counter, so that CLK signal will help the data to circulate from one flip flop to another.



State diagram of ring counter

The state diagram of the 4 bit ring counter is shown in above picture. It denotes that the position of the preset digit (in this case preset digit is 1) is changing its position from LSB to MSB, for one clock signal.



Ring counters

Advantages

- Can be implemented using D and JK flip-flops. It is a self-decoding circuit.

Disadvantages

- Only four of the 15 states are being utilized.

Johnson Counter

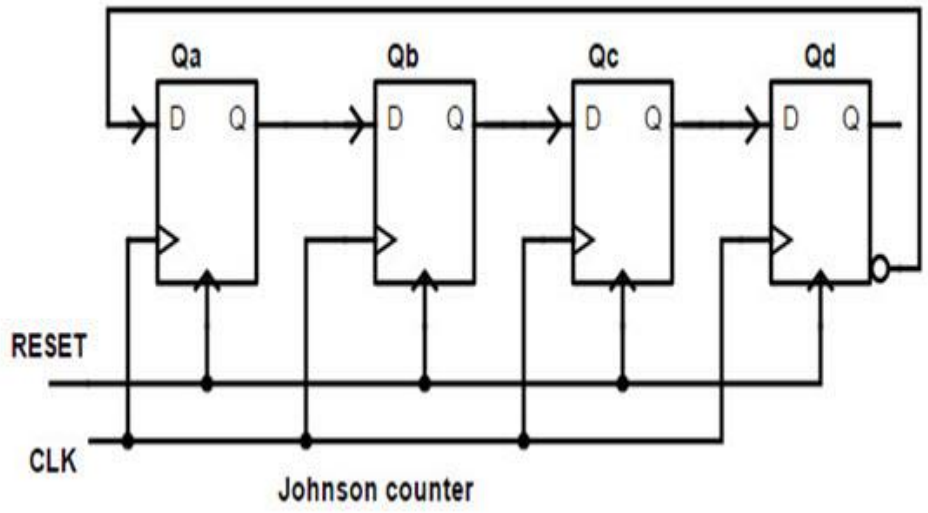
- The Johnson counter is a modification of ring counter. In this the inverted output of the last stage flip flop is connected to the input of first flip flop. If we use n flip flops to design the Johnson counter, it is known as 2n bit Johnson counter or Mod 2n Johnson counter.
- This is an advantage of the Johnson counter that it requires only half number of flip flops that of a ring counter uses, to design the same Mod.
- The main difference between the 4 bit ring counter and the Johnson counter is that , in ring counter , we connect the output of last flip flop directly to the input of first flip flop. But in Johnson counter, we connect the inverted output of last stage to the first stage input.
- The Johnson counter is also known as Twisted Ring Counter, with a feedback.

Johnson Counter

The Johnson counter designed with D flip flop is shown below. It has four stages i.e. four flip flops connected in series type or cascaded. Initially zero / Null is fed to the Johnson counter and on applying the clock signal, outputs will change to “1000”, “1100”, “1110”, “1111”, “0111”, “0011”, “0001”, “0000” in a sequence and the sequence will repeat for next clock signal.

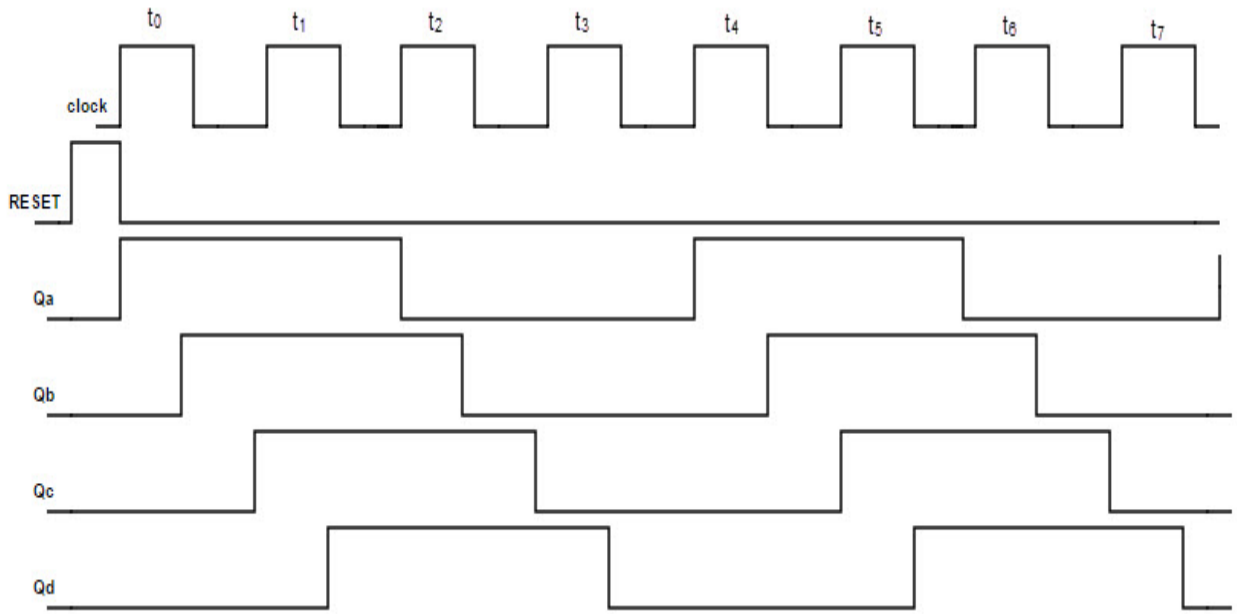
The Johnson counter produces a special pattern by passing four 0’s and then four 1’s and thus it produces a special pattern by counting up down.

Q_A	Q_B	Q_C	Q_D
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
repeat			



Timing diagram of Johnson counter

The timing diagram of the johnson counter will explain that the clock signal changes the output of every stage of the counter, so that CLK signal will help the data to circulate from one flip flop to another.

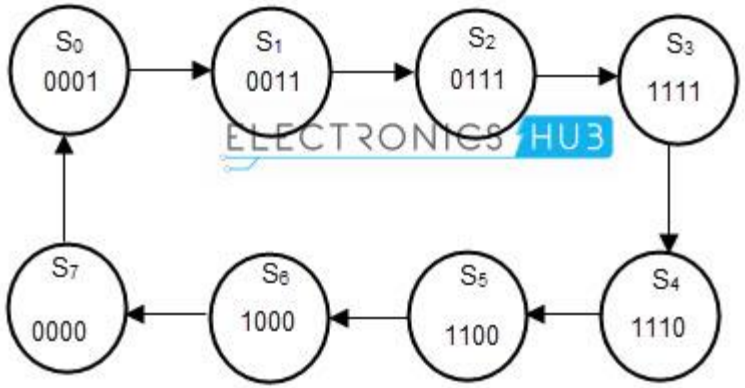


State diagram of a Johnson counter

When CLR = 0, all outputs and inputs of flip flops are preset to 0 (cleared) except the data input of right most FF which sets to 1.

When CLR = 1, Johnson counter starts its operation. On every clock edge, the output of last flip flop (1) shifts left to the third flip flop. As the first flip flop is connected to serial input i.e. 1, the input of third flip flop is 1.

In next cycle, QA = 0 so 0 rotates in ring form in second half cycle. Johnson counter has 8 sequences: 0001, 0011, 0111, 1111, 1110, 1100, 1000, and 0000.



Johnson Counter

- **Advantage** of Johnson counter is that, it has more outputs than ring counter.
- **Disadvantage** of Johnson counter is that , Only out of 15 states are only 8 are used.

- Applications of ring counters

Debouncer

- Mechanical switches and relays tend to make and break connections for a finite time before settling down to a stable state. Within this settling time, the digital circuit can see multiple transitions as the switch contacts bounce between make or break conditions.

- The Debouncer component takes an input signal from a bouncing contact and generates a clean output for digital circuits. The component will not pass the signal to the output until the predetermined period of time when the switch bouncing settles down. In this way, the circuit will respond to only one pulse generation performed by the pressing or releasing of the switch and not several state transitions caused by contact bouncing.

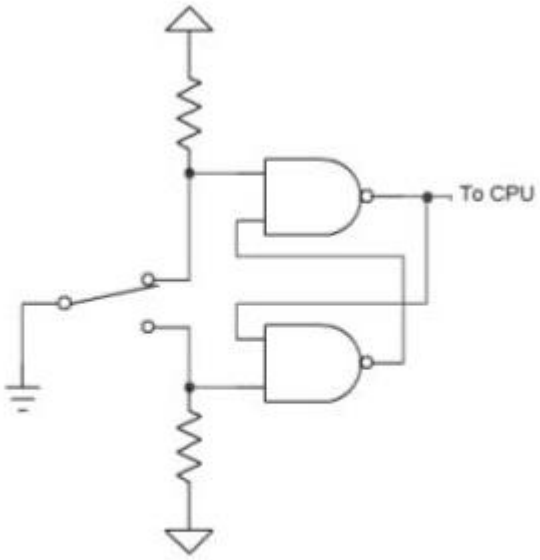
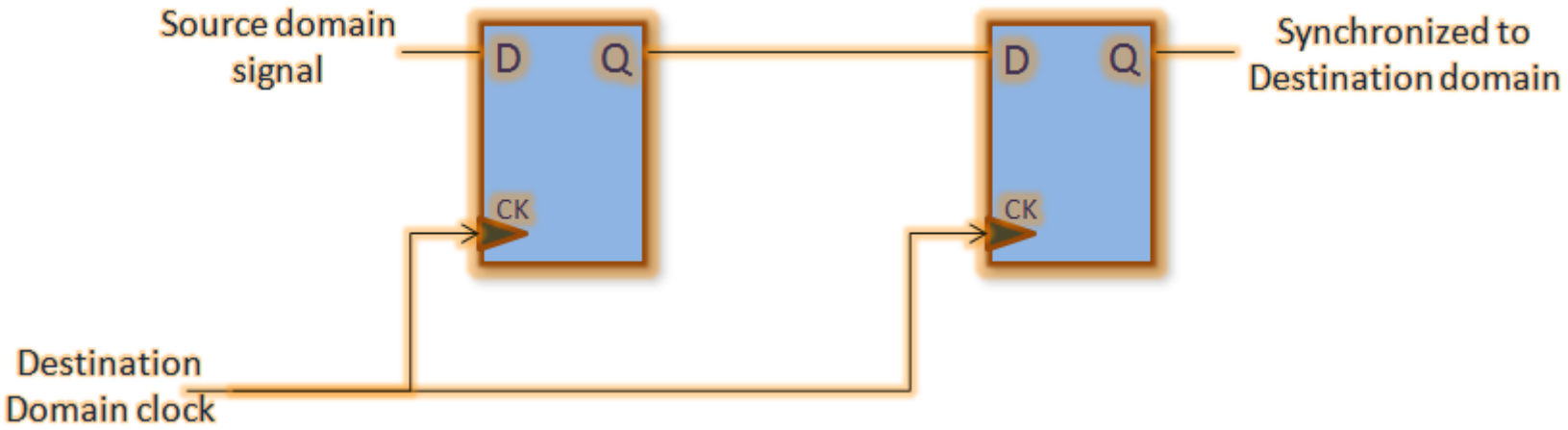


Figure 1: The SR debouncer

Synchronizers

- A synchronizer is a digital circuit that converts an asynchronous signal/a signal from a different clock domain into the recipient clock domain so that it can be captured without introducing any metastability failure.
- When a signal is passed from one clock domain to another clock domain, the circuit that receives the signal needs to synchronize it. Whatever metastability effects are to be caused due to this, have to be absorbed in synchronizer circuit only. Thus, the purpose of synchronizer is to prevent the downstream logic from metastable state of first flip-flop in new clock domain.

Synchronizers



Two flip-flop synchronizer

Synchronizers

- The two flop synchronizer converts a signal from source clock domain to destination clock domain. The input to the first stage is asynchronous to the destination clock. So, the output of first stage (Q1) might go metastable from time to time. However, as long as metastability is resolved before next clock edge, the output of second stage (Q2) should have valid logic levels.Thus, the asynchronous signal is synchronized with a maximum latency of 2 clock cycles.