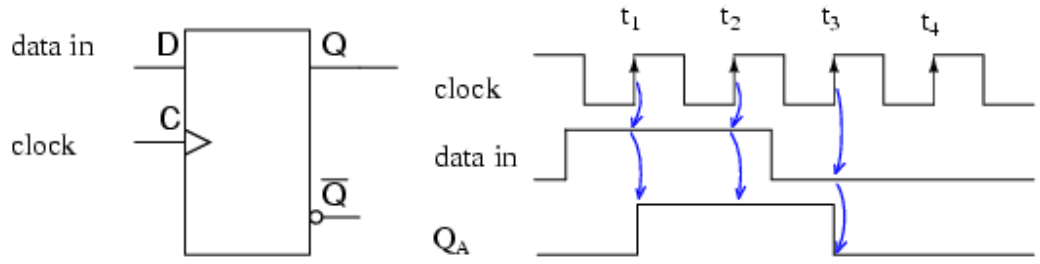
SERIAL-IN/SERIAL-OUT SHIFT REGISTER

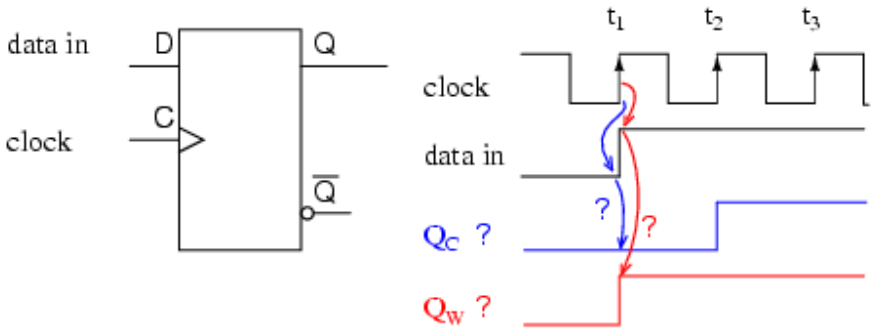
Serial-in, serial-out shift registers delay data by one clock time for each stage. They will store a bit of data for each register. A serial-in, serial-out shift register may be one to 64 bits in length, longer if registers or packages are cascaded.

Below is a single stage shift register receiving data which is not synchronized to the register clock. The "data in" at the D pin of the type D FF (Flip-Flop) does not change levels when the clock changes for low to high. We may want to synchronize the data to a system wide clock in a circuit board to improve the reliability of a digital logic circuit.



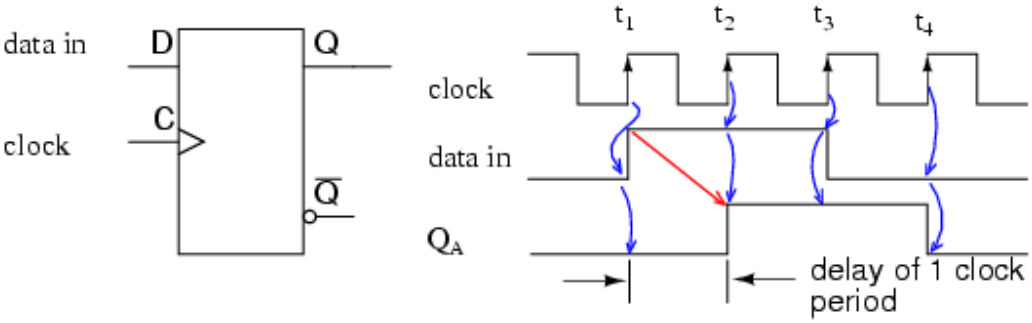
Data present at clock time is transfered from D to Q.

The obvious point (as compared to the figure below) illustrated above is that whatever "data in" is present at the D pin of a type D FF is transfered from D to output Q at clock time. Since our example shift register uses positive edge sensitive storage elements, the output Q follows the D input when the clock transitions from low to high as shown by the up arrows on the diagram above. There is no doubt what logic level is present at clock time because the data is stable well before and after the clock edge. This is seldom the case in multi-stage shift registers. But, this was an easy example to start with. We are only concerned with the positive, low to high, clock edge. The falling edge can be ignored. It is very easy to see Q follow D at clock time above. Compare this to the diagram below where the "data in" appears to change with the positive clock edge.



Does the clock t_1 see a 0 or a 1 at data in at D? Which output is correct, Q_C or Q_W ?

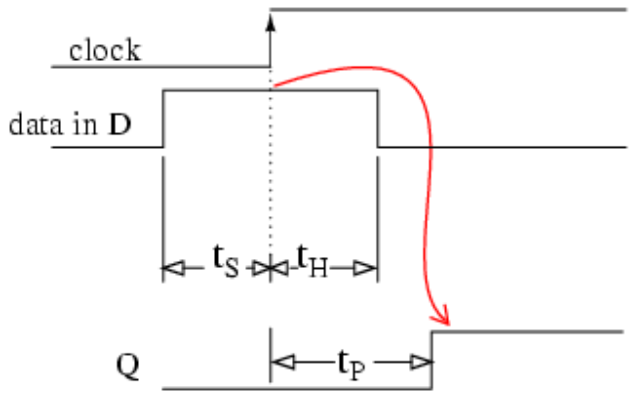
Since "data in" appears to changes at clock time t_1 above, what does the type D FF see at clock time? The short over simplified answer is that it sees the data that was present at D prior to the clock. That is what is transfered to Q at clock time t_1. The correct waveform is Q_C. At t_1 Q goes to a zero if it is not already zero. The D register does not see a one until time t_2, at which time Q goes high.



Data present t_H before clock time at D is transfered to Q.

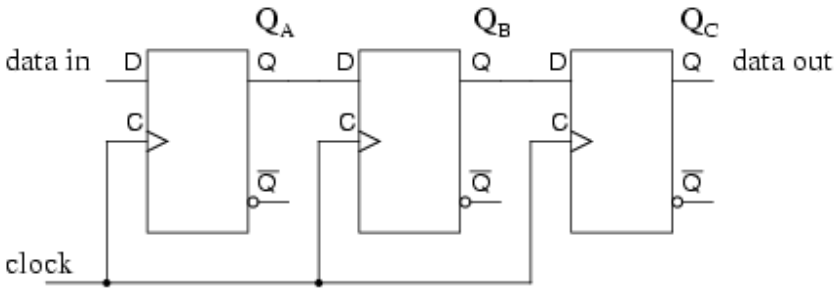
Since data, above, present at D is clocked to Q at clock time, and Q cannot change until the next clock time, the D FF delays data by one clock period, provided that the data is already synchronized to the clock. The Q_A waveform is the same as "data in" with a one clock period delay. A more detailed look at what the input of the type D Flip-Flop sees at clock time follows. Refer to the figure below. Since "data in" appears to changes at clock time (above), we need further information to determine what the D FF sees. If the "data in" is from another shift register stage, another same type D FF, we can draw some conclusions based on data sheet information. Manufacturers of digital logic make available information about their parts in data sheets,

formerly only available in a collection called a data book. Data books are still available; though, the manufacturer's web site is the modern source.



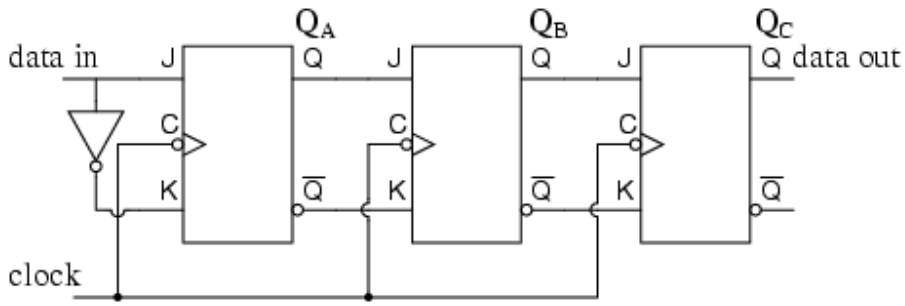
Data must be present (t_S) before the clock and after(t_H) the clock. Data is delayed from D to Q by propagation delay (t_P)

To summarize, output Q follows input D at nearly clock time if Flip-Flops are cascaded into a multi-stage shift register.



Serial-in, serial-out shift register using type "D" storage elements

Three type D Flip-Flops are cascaded Q to D and the clocks paralleled to form a three stage shift register above.

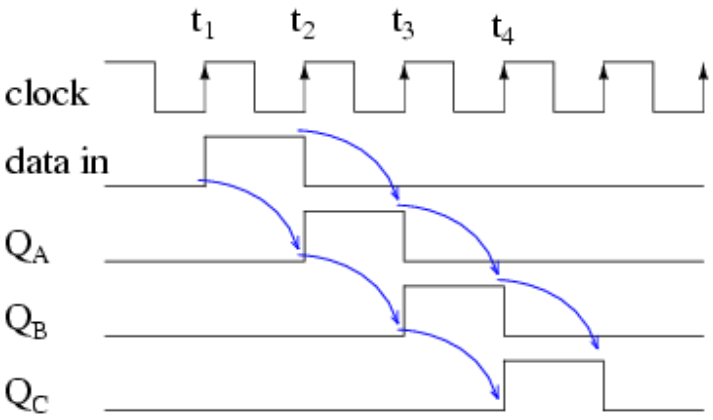


Serial-in, serial-out shift register using type "JK" storage elements

Type JK FFs cascaded Q to J, Q' to K with clocks in parallel to yield an alternate form of the shift register above.

A serial-in/serial-out shift register has a clock input, a data input, and a data output from the last stage. In general, the other stage outputs are not available Otherwise, it would be a serial-in, parallel-out shift register..

The waveforms below are applicable to either one of the preceding two versions of the serial-in, serial-out shift register. The three pairs of arrows show that a three stage shift register temporarily stores 3-bits of data and delays it by three clock periods from input to output.



At clock time t_1 a "data in" of 0 is clocked from D to Q of all three stages. In particular, D of stage A sees a logic 0, which is clocked to Q_A where it remains until time t_2.

At clock time t_2 a "data in" of 1 is clocked from D to Q_A. At stages B and C, a 0, fed from preceding stages is clocked to Q_B and Q_C.

At clock time t_3 a "data in" of 0 is clocked from D to Q_A. Q_A goes low and stays low for the remaining clocks due to "data in" being 0. Q_B goes high at t_3 due to a 1 from the previous stage. Q_C is still low after t_3 due to a low from the previous stage.

Q_C finally goes high at clock t_4 due to the high fed to D from the previous stage Q_B. All earlier stages have 0s shifted into them. And, after the next clock pulse at t_5, all logic 1s will have been shifted out, replaced by 0s