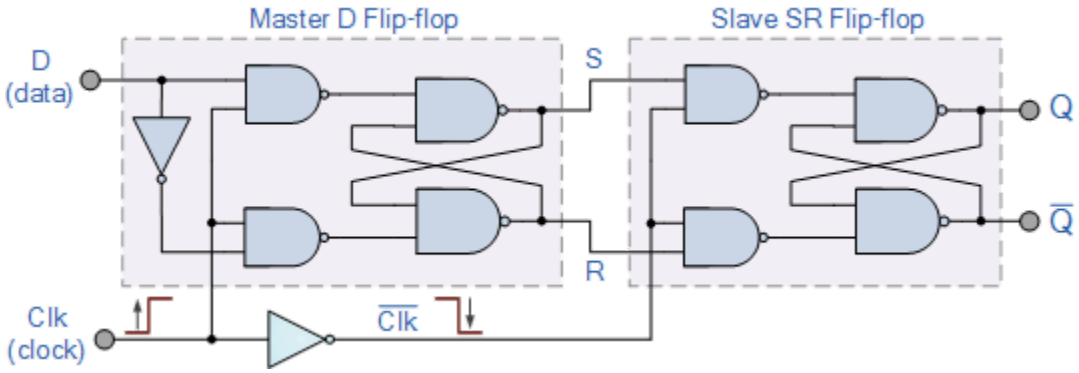


The D-type Flip Flop

The D-type flip-flop is a modified Set-Reset flip-flop with the addition of an inverter to prevent the S and R inputs from being at the same logic level

One of the main disadvantages of the basic **SR NAND Gate Bistable** circuit is that the indeterminate input condition of SET = "0" and RESET = "0" is forbidden.

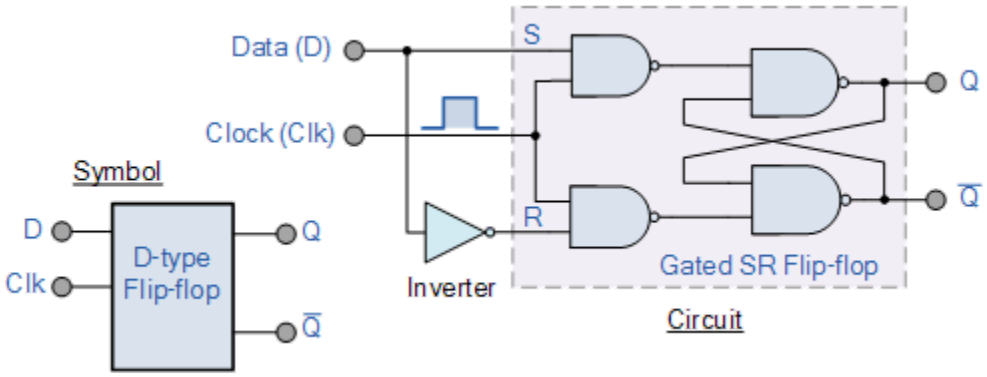
This state will force both outputs to be at logic "1", over-riding the feedback latching action and whichever input goes to logic level "1" first will lose control, while the other input still at logic "0" controls the resulting state of the latch.

But in order to prevent this from happening an inverter can be connected between the "SET" and the "RESET" inputs to produce another type of flip flop circuit known as a **Data Latch**, **Delay flip flop**, **D-type Bistable**, **D-type Flip Flop** or just simply a **D Flip Flop** as it is more generally called.

The **D Flip Flop** is by far the most important of the clocked flip-flops as it ensures that ensures that inputs S and R are never equal to one at the same time. The D-type flip flop are constructed from a gated SR flip-flop with an inverter added between the S and the R inputs to allow for a single D (data) input.

Then this single data input, labelled "D" and is used in place of the "Set" signal, and the inverter is used to generate the complementary "Reset" input thereby making a level-sensitive D-type flip-flop from a level-sensitive SR-latch as now S = D and R = not D as shown.

D-type Flip-Flop Circuit



We remember that a simple SR flip-flop requires two inputs, one to "SET" the output and one to "RESET" the output. By connecting an inverter (NOT gate) to the SR flip-flop we can "SET" and "RESET" the flip-flop using just one input as now the two input signals are complements of each other. This complement avoids the ambiguity inherent in the SR latch when both inputs are LOW, since that state is no longer possible.

Thus this single input is called the "DATA" input. If this data input is held HIGH the flip flop would be "SET" and when it is LOW the flip flop would change and become "RESET". However, this would be rather pointless since the output of the flip flop would always change on every pulse applied to this data input.

To avoid this an additional input called the "CLOCK" or "ENABLE" input is used to isolate the data input from the flip flop's latching circuitry after the desired data has been stored. The effect is that D input condition is only copied to the output Q when the clock input is active. This then forms the basis of another sequential device called a **D Flip Flop**.

The "D flip flop" will store and output whatever logic level is applied to its data terminal so long as the clock input is HIGH. Once the clock input goes LOW the "set" and "reset" inputs of the flip-flop are both held at logic level "1" so it will not change state and store whatever data was present on its output before the clock transition occurred. In other words the output is "latched" at either logic "0" or logic "1".

Truth Table for the D-type Flip Flop

Clk	D	Q	$\overline{Q}$	Description
↓ » 0	X	Q	$\overline{Q}$	Memory no change
↑ » 1	0	0	1	Reset Q » 0
↑ » 1	1	1	0	Set Q » 1

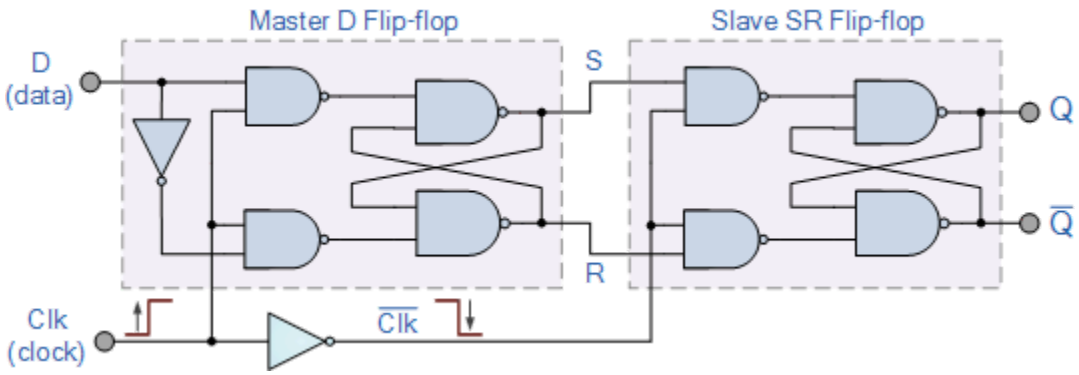
Note that: ↓ and ↑ indicates direction of clock pulse as it is assumed D-type flip flops are edge triggered

The Master-Slave D Flip Flop

The basic **D-type flip flop** can be improved further by adding a second SR flip-flop to its output that is activated on the complementary clock signal to produce a "Master-Slave D-type flip flop". On the leading edge of the clock signal (LOW-to-HIGH) the first stage, the "master" latches the input condition at D, while the output stage is deactivated.

On the trailing edge of the clock signal (HIGH-to-LOW) the second "slave" stage is now activated, latching on to the output from the first master circuit. Then the output stage appears to be triggered on the negative edge of the clock pulse. "Master-Slave D-type flip flops" can be constructed by the cascading together of two latches with opposite clock phases as shown.

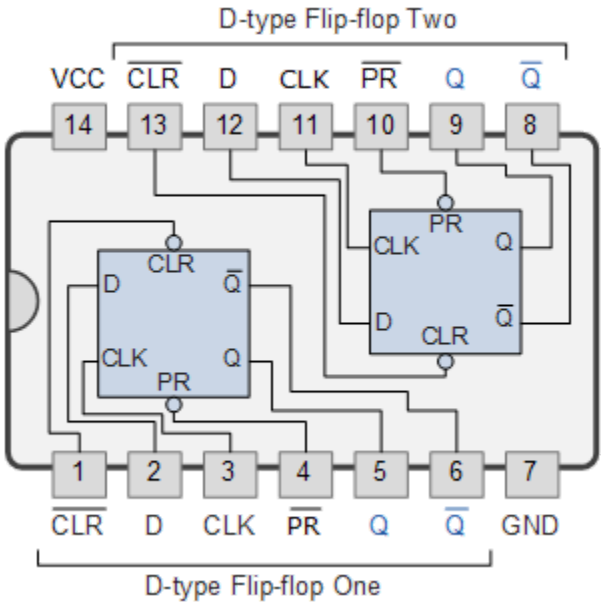
The Master-Slave D Flip Flop Circuit



We can see from above that on the leading edge of the clock pulse the master flip-flop will be loading data from the data D input, therefore the master is "ON". With the trailing edge of the clock pulse the slave flip-flop is loading data, i.e. the slave is "ON". Then there will always be one flip-flop "ON" and the other "OFF" but never both the master and slave "ON" at the same time. Therefore, the output Q acquires the value of D, only when one complete pulse, ie, 0-1-0 is applied to the clock input.

There are many different D flip-flop IC's available in both TTL and CMOS packages with the more common being the 74LS74 which is a Dual D flip-flop IC, which contains two individual D type bistable's within a single chip enabling single or master-slave toggle flip-flops to be made. Other D flip-flop IC's include the 74LS174 HEX D flip-flop with direct clear input, the 74LS175 Quad D flip-flop with complementary outputs and the 74LS273 Octal D-type flip flop containing eight D-type flip flops with a clear input in one single package.

74LS74 Dual D-type Flip Flop



Other Popular D-type flip-flop ICs

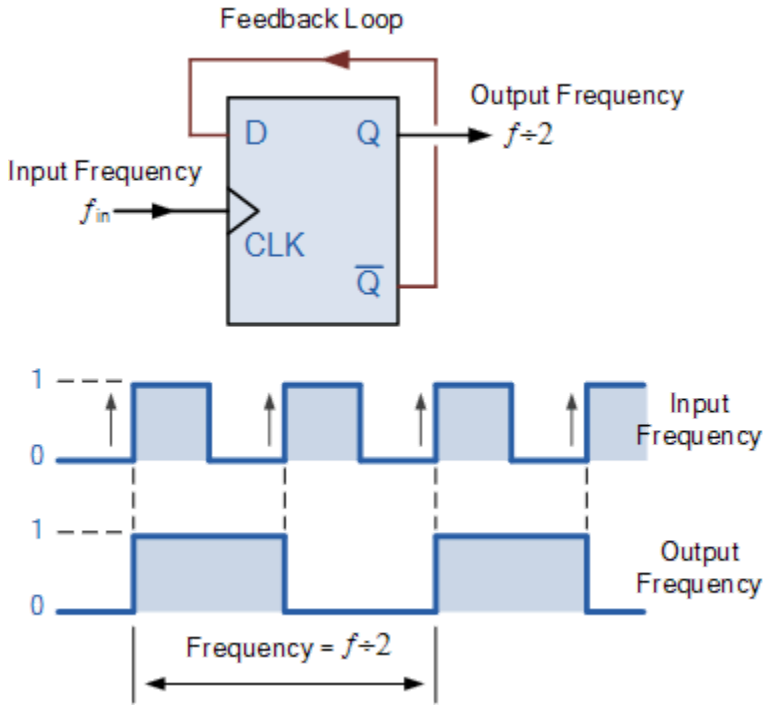
Device Number	Subfamily	Device Description
74LS74	LS TTL	Dual D-type Flip Flops with Preset and Clear
74LS175	LS TTL	Quad D-type Flip Flops with Clear
74LS273	LS TTL	Octal D-type Flip Flops with Clear
4013B	Standard CMOS	Dual type D Flip Flop
40174B	Standard CMOS	Hex D-type Flip Flop with Master Reset

Using The D-type Flip Flop For Frequency Division

One main use of a D-type flip flop is as a Frequency Divider. If the $\overline{Q}$ output on a D-type flip-flop is connected directly to the D input giving the device closed loop "feedback", successive clock pulses will make the bistable "toggle" once every two clock cycles.

In the counters tutorials we saw how the **Data Latch** can be used as a "Binary Divider", or a "Frequency Divider" to produce a "divide-by-2" counter circuit, that is, the output has half the frequency of the clock pulses. By placing a feedback loop around the D-type flip flop another type of flip-flop circuit can be constructed called a **T-type flip-flop** or more commonly a T-type bistable, that can be used as a divide-by-two circuit in binary counters as shown below.

Divide-by-2 Counter

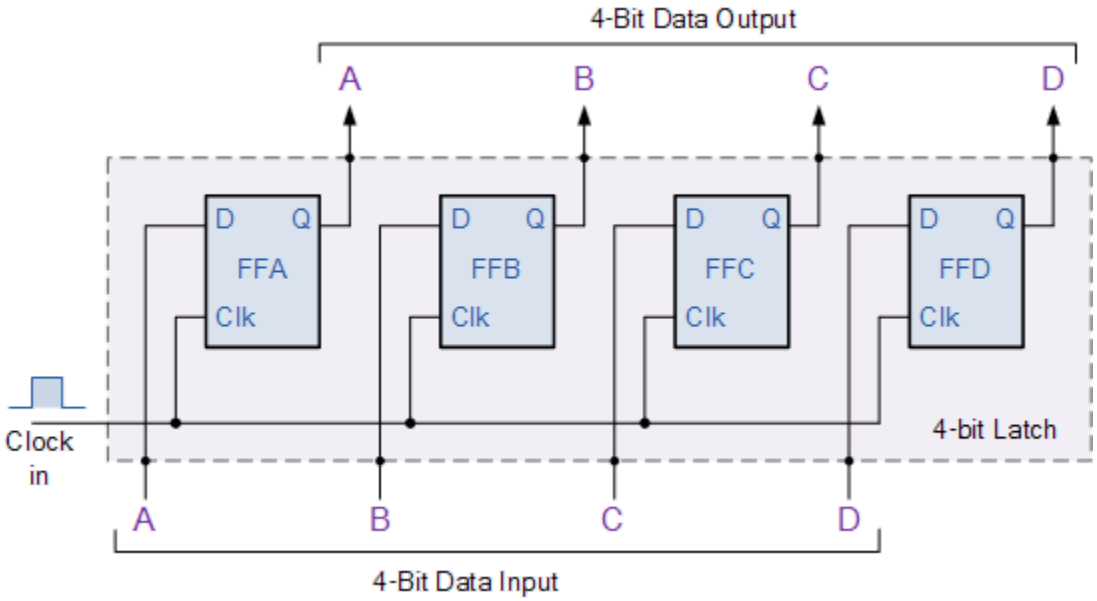


It can be seen from the frequency waveforms above, that by "feeding back" the output from $\overline{Q}$ to the input terminal D, the output pulses at Q have a frequency that are exactly one half ($f/2$) that of the input clock frequency, (f_{IN}). In other words the circuit produces **frequency division** as it now divides the input frequency by a factor of two (an octave) as $Q = 1$ once every two clock cycles.

D Flip Flops as Data Latches

As well as frequency division, another useful application of the D flip flop is as a **Data Latch**. A data latch can be used as a device to hold or remember the data present on its data input, thereby acting a bit like a single bit memory device and IC's such as the TTL 74LS74 or the CMOS 4042 are available in Quad format exactly for this purpose. By connecting together four, *1-bit* data latches so that all their clock inputs are connected together and are "clocked" at the same time, a simple "4-bit" Data latch can be made as shown below.

4-bit Data Latch



Transparent Data Latch

The **Data Latch** is a very useful device in electronic and computer circuits. They can be designed to have very high output impedance at both outputs Q and its inverse or complement output $\overline{Q}$ to reduce the impedance effect on the connecting circuit when used as a buffer, I/O port, bi-directional bus driver or even a display driver.

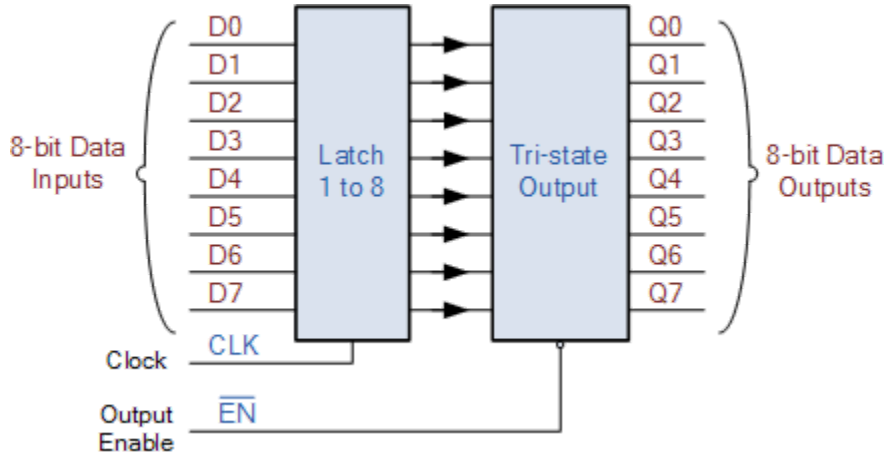
But a single "1-bit" data latch is not very practical to use on its own and instead commercially available IC's incorporate 4, 8, 10, 16 or even 32 individual data latches into one single IC package, and one such IC device is the 74LS373 Octal D-type transparent latch.

The eight individual data latches or bistables of the 74LS373 are "transparent" D-type flip-flops, meaning that when the clock (CLK) input is HIGH at logic level "1", (but can also be active low) the outputs at Q follows the data D inputs.

In this configuration the latch is said to be "open" and the path from $\overline{D}$ input to $\overline{Q}$ output appears to be "transparent" as the data flows through it unimpeded, hence the name transparent latch.

When the clock signal is LOW at logic level "0", the latch "closes" and the output at $\overline{Q}$ is latched at the last value of the data that was present before the clock signal changed and no longer changes in response to $\overline{D}$.

8-bit Data Latch



Functional diagram of the 74LS373 Octal Transparent Latch

The D-type Flip Flop Summary

The data or **D-type Flip Flop** can be built using a pair of back-to-back SR latches and connecting an inverter (NOT Gate) between the S and the R inputs to allow for a single D (data) input. The basic D flip flop circuit can be improved further by adding a second SR flip-flop to its output that is activated on the complementary clock signal to produce a "Master-Slave D flip-flop" device.

The difference between a D-type latch and a D-type flip-flop is that a latch does not have a clock signal to change state whereas a flip-flop always does. The D flip-flop is an edge triggered device which transfers input data to $\overline{Q}$ on clock rising or falling edge. Data Latches are level sensitive devices such as the data latch and the transparent latch.

In the next tutorial about **Sequential Logic Circuits**, we will look at connecting together data latches to produce another type of sequential logic circuit called a *Shift Register* that are used to convert parallel data into serial data and vice versa.

35 Comments

Write your comment here

SUBMIT

R Rehan

Their should be proper details of asked questions

Posted on October 18th 2018 | 9:12 am

Reply

P Pavan Nayani

Not sure if this is the place to ask for help but here goes – using a 74ls74 dual flip-flop chip in the hope of reducing speedo pulses by half – used this info and made up the circuit with feedback loop – double checked but will not divide by 2 – speedo reads the same with or without the circuit in the loop – what am I missing?

Posted on June 11th 2018 | 1:23 pm

Reply

W Wayne Storr

For the 74ls74 dual D-type flip-flop to toggle, pin-1, pin-4 and pin-14 to Vcc, pin-7 to 0v, pin-6 connected directly to pin-2.
A positive going pulse applied to pin-3 will cause pin-5 (Q) to change state every second pulse on

pin-3

Repeat for the second flip-flop within the 74ls74

Posted on June 11th 2018 | 3:19 pm

Reply

W

William Scannell

Like to touch base with someone that would be good….cheers

Posted on June 09th 2018 | 1:45 pm

Reply

M

MURASOLI

Can sent me shift register operation table.

For D- flip flop shift register

Posted on March 20th 2018 | 3:00 pm

Reply

B

Ben gusmandar

How to change and convert signal to driver on transmiter system ?

Posted on March 08th 2018 | 4:32 am

Reply

K

Khin Moh Moh Tin

Ṭhank you for this tuṭorials

Posted on January 20th 2018 | 10:14 pm

Reply

A Anandthan Mirzathan Loganathan

Ya bois nerds

Posted on November 29th 2017 | 1:23 pm

Reply

B Bhuvanesh

List the IC'S used for flip flops. Shift register, counter with there pin diagram.

Posted on October 15th 2017 | 5:06 am

Reply

N Neil

Not sure if this is the place to ask for help but here goes – using a 74ls74 dual flip-flop chip in the hope of reducing speedo pulses by half – used this info and made up the circuit with feedback loop – double checked but will not divide by 2 – speedo reads the same with or without the circuit in the loop – what am I missing?

Posted on October 06th 2017 | 7:04 pm

Reply

W Wayne Storr

The 74LS74 is a dual positive-edge triggered D-type flip-flop which can be configured to perform as a divide-by-two counter. But to do so, !PR and !CLR must be tied together HIGH (to logic-1), NOT-Q connected to D (feedback loop) and the clock signal applied directly to CLK. The output is present on Q.

Posted on October 08th 2017 | 8:10 am

Reply

A ANTONIO

very useful information, thank you so much

Posted on August 27th 2017 | 5:46 am

Reply

View More