Flip-flop (electronics)

From Wikipedia, the free encyclopedia

In digital circuits, a **flip-flop** is an onomatopoeic term referring to an electronic circuit (a bistable multivibrator) that has two stable states and thereby is capable of serving as one bit of memory. Today, the term *flip-flop* has come to mostly denote *non-transparent* (*clocked* or *edge-triggered*) devices, while the simpler *transparent* ones are often referred to as latches; however, as this distinction is quite new, the two words are sometimes used interchangeably (see history).

A flip-flop is usually controlled by one or two control signals and/or a gate or clock signal. The output often includes the complement as well as the normal output. As flip-flops are implemented electronically, they require power and ground connections.

Contents

History

The first electronic flip-flop was invented in 1919 by William Eccles and F. W. Jordan.[1] It was initially called the *Eccles-Jordan trigger circuit*, and consisted of two active elements (radio-tubes). The name flip-flop was later derived from the sound produced on a speaker connected with one of the back coupled amplifiers output during the trigger process within the circuit. This original *electronic* flip-flop—a simple two-input bistable circuit without any dedicated clock (or even gate) signal, was *transparent*, and thus a device that would be labeled as a "latch" in many circles today.

Implementation

Flip-flops can be either simple (transparent) or clocked. Simple flip-flops can be built around a pair of cross-coupled *inverting* elements: vacuum tubes, bipolar transistors, field effect transistors, inverters, and inverting logic gates have all been used in practical circuits — perhaps augmented by some gating mechanism (an enable/disable input). The more advanced clocked (or non-transparent) devices are specially designed for synchronous (time-discrete) systems; such devices therefore ignores its inputs except **at** the transition of a dedicated clock signal (known as clocking, pulsing, or strobing). This causes the flip-flop to either change or retain its output signal based upon the values of the input signals at the transition. Some flip-flops change output on the rising edge of the clock, others on the falling edge.

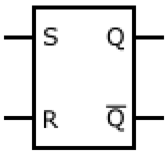
Clocked flip-flops are typically implemented as master-slave devices[2] where two basic flip-flops (plus some additional logic) collaborate to make it insensitive to spikes and noise between the short clock transitions; they nevertheless also often include asynchronous *clear* or *set* inputs which may be used to change the current output independent of the clock.

Flip-flops can be further divided into types that have found common applicability in both asynchronous and clocked sequential systems: the **SR** ("set-reset"), **D** ("data" or "delay"[3]), **T** ("toggle"), and **JK** types are the common ones; all of which may be synthesized from (most) other types by a few logic gates. The behavior of a particular type can be described by what is termed the characteristic equation, which derives the "next" (i.e., after the next clock pulse) output, Q_{next}, in terms of the input signal(s) and/or the current output, Q.

Set-Reset flip-flops (SR flip-flops)

The most fundamental latch is the simple *SR latch* (or simple ***SR flip-flop***), where S and R stand for *set* and *reset* respectively. It can be constructed from a pair of cross-coupled NOR (negative OR) logic gates. The stored bit is present on the output marked Q.

Normally, in storage mode, the S and R inputs are both low, and feedback maintains the Q and $\overline{Q}$ outputs in a constant state, with $\overline{Q}$ the complement of Q. If S (*Set*) is pulsed high while R is held low, then the Q output is forced high, and stays high even after S returns low; similarly, if R (*Reset*) is pulsed high while S is held low, then the Q output is forced low, and stays low even after R returns low.



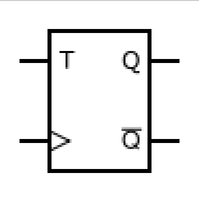
The symbol for an SR latch.

SR latch operation		
S	**R**	**Action**
0	0	Keep state
0	1	Q = 0
1	0	Q = 1
1	1	Unstable combination, see race condition

Toggle flip-flops (T flip-flops)

If the T input is high, the T flip-flop changes state ("toggles") whenever the clock input is strobed. If the T input is low, the flip-flop holds the previous value. This behavior is described by the characteristic equation:

$Q_{next} = T \oplus Q$ (or, without benefit of the XOR operator, the equivalent: $Q_{next} = T\overline{Q} + \overline{T}Q$)



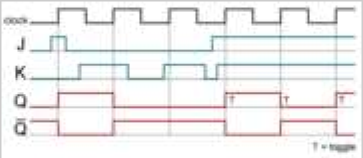
A circuit symbol for a T-type flip-flop, where > is the clock input, T is the toggle input and Q is the stored data output.

and can be described in a truth table:

T	*Q*	*Q_{next}*	Comment
0	0	0	hold state(no clk)
0	1	1	hold state(no clk)
1	0	1	toggle
1	1	0	toggle

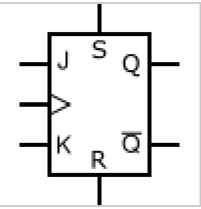
When T is held high, the toggle flip-flop divides the clock frequency by two; that is, if clock frequency is 4 MHz, the output frequency obtained from the flip-flop will be 2 MHz. This 'divide by' feature has application in various types of digital counters. A T flip-flop can also be built using a JK flip-flop (J & K pins are connected together and act as T) or D flip-flop (T input and $Q_{previous}$ is connected to the D input through an XOR gate).

JK flip-flop



JK flip-flop timing diagram

The **JK** flip-flop augments the behavior of the SR flip-flop (J=Set, K=Reset) by interpreting the S = R = 1 condition as a "flip" or toggle command. Specifically, the combination J = 1, K = 0 is a command to set the flip-flop; the combination J = 0, K = 1 is a command to reset the flip-flop; and the combination J = K = 1 is a command to toggle the flip-flop, i.e., change its output to the logical complement of its current value. Setting J = K = 0 does NOT result in a D flip-flop, but rather, will hold the current state. To synthesize a D flip-flop, simply set K equal to the complement of J. The JK flip-flop is therefore a universal flip-flop, because it can be configured to work as an SR flip-flop, a D flip-flop, or a T flip-flop. NOTE: The flip flop is positive edge triggered (Clock Pulse) as seen in the timing diagram.



A circuit symbol for a JK flip-flop, where > is the clock input, J and K are data inputs, Q is the stored data output, and Q' is the inverse of Q.

The characteristic equation of the JK flip-flop is:

$$Q_{next} = J\overline{Q} + \overline{K}Q$$

and the corresponding truth table is:

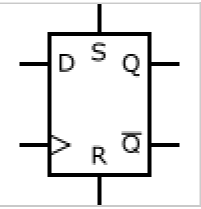
J	K	Q_{next}	Comment
0	0	Q_{prev}	hold state
0	1	0	reset
1	0	1	set
1	1	$\overline{Q_{prev}}$	toggle

The origin of the name for the JK flip-flop is detailed by P. L. Lindley, a JPL engineer, in a letter to *EDN*, an electronics design magazine. The letter is dated June 13, 1968, and was published in the August edition of the newsletter. In the letter, Mr. Lindley explains that he heard the story of the JK flip-flop from Dr. Eldred Nelson, who is responsible for coining the term while working at Hughes Aircraft. Flip-flops in use at Hughes at the time were all of the type that came to be known as J-K. In designing a logical system, Dr. Nelson assigned letters to flip-flop inputs as follows: #1: A & B, #2: C & D, #3: E & F, #4: G & H, #5: J & K.

Another theory holds that the set and reset inputs were given the symbols "J" and "K" after one of the engineers that helped design the J-K flip-flop, Jack Kilby.

D flip-flop

The Q output always takes on the state of the D input at the moment of a rising clock edge or at the moment of the falling clock edge but not both at the falling and rising edges and never at any other time. [4] It is called the **D** flip-flop for this reason, since the output takes the value of the **D** input or *Data* input, and *Delays* it by one clock count. The D flip-flop can be interpreted as a primitive memory cell, zero-order hold, or delay line.



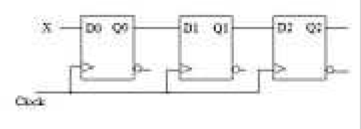
D flip-flop symbol

Truth table:

Clock	D	Q	Q_{prev}
Rising edge	0	0	X
Rising edge	1	1	X
Non-Rising	X	constant	

('X' denotes a *Don't care* condition, meaning the signal is irrelevant)

These flip flops are very useful, as they form the basis for shift registers, which are an essential part of many electronic devices. The advantage of the D flip-flop over the D-type latch is that it "captures" the signal at the moment the clock goes high, and subsequent changes of the data line do not influence Q until the next rising clock edge. An exception is that some flip-flops have a 'reset' signal input, which will reset Q (to zero), and may be either asynchronous or synchronous with the clock.



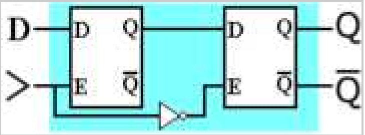
3-bit shift register

The above circuit shifts the contents of the register to the right, one bit position on each active transition of the clock. The input X being shifted into the leftmost bit position.

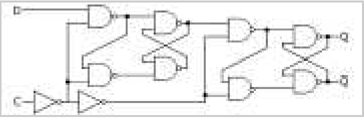
Master-slave (Pulse-Triggered) D flip-flop

A master-slave D flip-flop is created by connecting two gated D latches in series, and inverting the *enable* input to one of them. It is called master-slave because the second latch in the series only changes in response to a change in the first (master) latch. The term pulse triggered means that data are entered on the rising edge of the clock pulse, but the output doesn't reflect the change until the falling edge of the clock pulse.

For a positive-edge triggered master-slave D flip-flop, when the clock signal is low (logical 0) the “enable” seen by the first or “master” D latch (the inverted clock signal) is high (logical 1). This allows the “master” latch to store the input value when the clock signal transitions from low to high. As the clock signal goes high (0 to 1) the inverted “enable” of the first latch goes low (1 to 0) and the value seen at the input to the master latch is “locked”. Nearly simultaneously, the twice inverted “enable” of the second or “slave” D latch transitions from low to high (0 to 1) with the clock signal. This allows the signal captured at the rising edge of the clock by the now “locked” master latch to pass through the “slave” latch. When the clock signal returns to low (1 to 0), the output of the "slave" latch is "locked", and the value seen at the last rising edge of the clock is held while the “master” latch begins to accept new values in preparation for the next rising clock edge.



A master slave D flip flop. It responds on the negative edge of the *enable* input (usually a clock).



An implementation of a master-slave D flip-flop that is triggered on the positive edge of the clock.

By removing the left-most inverter in the above circuit, a D-type flip flop that strobes on the *falling edge* of a clock signal can be obtained. This has a truth table like this:

D	Q	>	Q_{next}
0	X	Falling	0
1	X	Falling	1

Most D-type flip-flops in ICs have the capability to be set and reset, much like an SR flip-flop. Usually, the illegal S = R = 1 condition is resolved in D-type flip-flops.

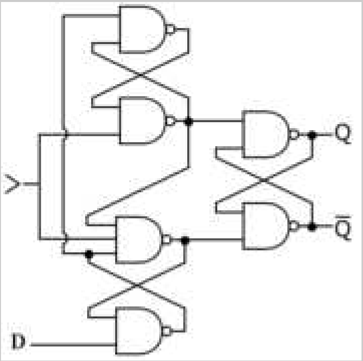
Inputs				Outputs	
S	R	D	>	Q	Q'
0	1	X	X	0	1
1	0	X	X	1	0
1	1	X	X	1	1

By setting S = R = 0, the flip-flop can be used as described above.

Edge-triggered D flip-flop

A more efficient way to make a D flip-flop is not as easy to understand, but it works the same way. While the master-slave D flip flop is also triggered on the edge of a clock, its components are each triggered by clock levels. The "edge-triggered D flip flop" does not have the master slave properties.

Uses



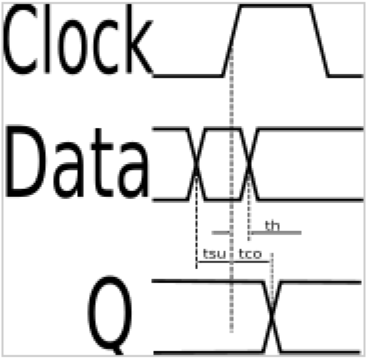
A positive-edge-triggered D flip-flop.

- A single flip-flop can be used to store one bit, or binary digit, of data. See preset.

- Any one of the flip-flop types can be used to build any of the others.
- The data contained in several flip-flops may represent the state of a sequencer, the value of a counter, an ASCII character in a computer's memory or any other piece of information.
- One use is to build finite state machines from electronic logic. The flip-flops remember the machine's previous state, and digital logic uses that state to calculate the next state.
- The T flip-flop is useful for constructing various types of counters. Repeated signals to the clock input will cause the flip-flop to change state once per high-to-low transition of the clock input, if its T input is "1". The output from one flip-flop can be fed to the clock input of a second and so on. The final output of the circuit, considered as the array of outputs of all the individual flip-flops, is a count, in binary, of the number of cycles of the first clock input, up to a maximum of 2^n-1, where *n* is the number of flip-flops used. See: Counters
  - One of the problems with such a counter (called a *ripple counter*) is that the output is briefly invalid as the changes ripple through the logic. There are two solutions to this problem. The first is to sample the output only when it is known to be valid. The second, more widely used, is to use a different type of circuit called a *synchronous counter*. This uses more complex logic to ensure that the outputs of the counter all change at the same, predictable time. See: Counters
- Frequency division: a chain of T flip-flops as described above will also function to divide an input in frequency by 2^n, where *n* is the number of flip-flops used between the input and the output.

A flip-flop in combination with a Schmitt trigger can be used for the implementation of an arbiter in asynchronous circuits.

Clocked flip-flops are prone to a problem called metastability, which happens when a data or control input is changing at the instant of the clock pulse. The result is that the output may behave unpredictably, taking many times longer than normal to settle to its correct state, or even oscillating several times before settling. Theoretically it can take infinite time to settle down. In a computer system this can cause corruption of data or a program crash.



Flip-flop setup, hold and clock-to-output timing parameters.

The metastability in flip-flops can be avoided by ensuring that the data and control inputs are held valid and constant for specified periods before and after the clock pulse, called the **setup time** (t_{su}) and the **hold time** (t_h) respectively. These times are specified in the data sheet for the device, and are typically between a few nanoseconds and a few hundred picoseconds for modern devices.

Unfortunately, it is not always possible to meet the setup and hold criteria, because the flip-flop may be connected to a real-time signal that could change at any time, outside the control of the designer. In this case, the best the designer can do is to reduce the probability of error to a certain level, depending on the required reliability of the circuit. One technique for suppressing metastability is to connect two or more flip-flops in a chain, so that the output of each one feeds the data input of the next, and all devices share a common clock. With this method, the probability of a metastable event can be reduced to a negligible value, but never to zero. The probability of metastability gets closer and closer to zero as the number of flip-flops connected in series is increased.

So-called metastable-hardened flip-flops are available, which work by reducing the setup and hold times as much as possible, but even these cannot eliminate the problem entirely. This is because metastability is more than simply a matter of circuit design. When the transitions in the clock and the data are close together in time, the flip-flop is forced to decide which event happened first. However fast we make the device, there is always the possibility that the input events will be so close together that it cannot detect which one happened first. It is therefore logically impossible to build a perfectly metastable-proof flip-flop.

Another important timing value for a flip-flop is the clock-to-output delay (common symbol in data sheets: t_{CO}) or propagation delay (t_P), which is the time the flip-flop takes to change its output after the clock edge. The time for a high-to-low transition (t_{PHL}) is sometimes different from the time for a low-to-high transition (t_{PLH}).

When connecting flip-flops in a chain, it is important to ensure that the t_{CO} of the first flip-flop is longer than the hold time (t_H) of the second flip-flop, otherwise the second flip-flop will not receive the data reliably. The relationship between t_{CO} and t_H is normally guaranteed if both flip-flops are of the same type.

Flip-flop integrated circuits

Integrated circuit (ICs) exist that provide one or more flip-flops. For example, the 7473 Dual JK Master-Slave Flip-flop or the 74374, an octal D Flip-flop, in the 7400 series.

See also

- Monostable
- Astable
- Pulse transition detector
- Deadlock

Notes and references

1. ^ W. H. Eccles and F. W. Jordan (19 September 1919) "A trigger relay utilizing three-electrode thermionic vacuum tubes," *The Electrician*, vol. 83, page 298. Reprinted in: *Radio Review*, vol. 1, no. 3, pages 143-146 (December 1919).
2. ^ Early master-slave devices actually remained (half) open between the first and second edge of a clocking pulse; today most flip-flops are designed so they may be clocked by a **single** edge as this gives large benefits regarding noise immunity, without any significant downsides.
3. ^ PHY107 Delay Flip-Flop (http://www.shef.ac.uk/physics/teaching/phy107/dff.html)
4. ^ The D Flip-Flop (http://www.play-hookey.com/digital/d_nand_flip-flop.html)

- Hwang, Enoch (2006). *Digital Logic and Microprocessor Design with VHDL*. Thomson. ISBN 0-534-46593-5.
- Salman, E., Dasdan, A., Taraporevala, F., Kucukcakar, K., Friedman, E. (2006). "Pessimism Reduction in Statis Timing Analysis Using Interdependent Setup and Hold Times". *Proc. of Int. Symp. on Quality Electronic Design (ISQED)*: 159-164. (This paper explains the interdependence of setup time, hold time, and clock-to-q delay and shows how to use it for pessimism reduction in static timing analysis.)
- Template:Author = Schulz, Klaus-E. Ideal pulse circuit without RC-combination and non-clocked JK flip-flops (look discussion) (http://www.hpc-berlin.de/dokumente/flipflop_en.pdf)

Retrieved from "http://en.wikipedia.org/wiki/Flip-flop_(electronics)"
Categories: Digital electronics | Electronic engineering | Digital systems | Oscillators | Logic gates