

Flip-flops for Pulse Mode Memory

The latches can be used as memory elements in fundamental mode circuits. Usually they are used as the building blocks of **flip-flops**.

These are memory devices whose operation is designed around a pulse signal to change state. The transparency of the latch output to its inputs is removed and design is considerably simpler due to the removal of unstable transient states.

The D flip-flop removes this transparent tracking of the input signal by using a master-slave arrangement of D-latches and control signal C . One such arrangement is shown in the figure on the next page.

While $C = 0$ the master latch transparently follows the D signal and the slave latch is decoupled from the output of the master latch and therefore maintains its old state while the new state is in preparation, e.g., the fluctuations on the input line to the flip-flop.

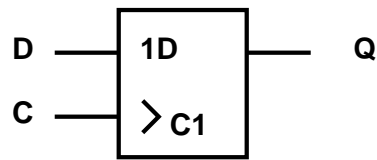
When the transition $C = 0 \rightarrow 1$ occurs (a leading edge of the C signal) the master latch is decoupled from the input D signal to the flip-flop and its output Q maintains its last value.

The connection to the slave is then opened with the Q signal from the master latch as its input to which the output of the slave latch (and the D flip-flop), F , settles transparently. (So $C = 1$ must be maintained longer than the settling time of the slave latch).

When C drops back to 0 the slave is once again detached from the master latches output Q and the connection of D to the master latch is restored and Q once again follows it transparently.

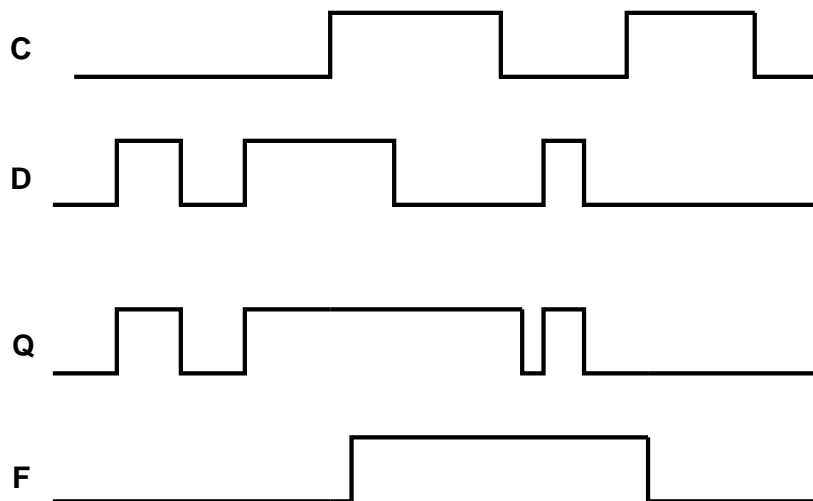
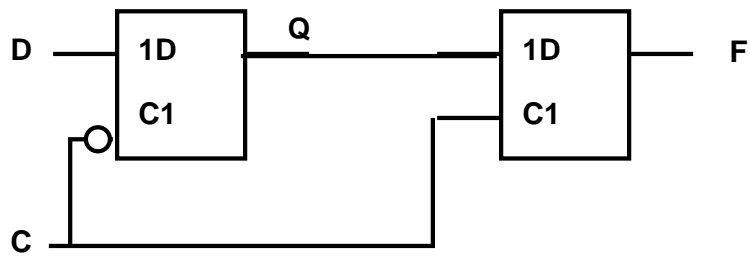
Such a device is useful to store a bit in a register.

D flip-flop



master latch

slave latch



Characteristic function

$$Y^+ = D$$

superscript + emphasizes the transition on the leading edge of C

The master-slave clocking works well in the idealized timing setting above where changes to C and C' are available instantly and simultaneously.

In practice this is not the case.

All flip-flops, e.g., the D, the SR, and others, (along with many other fundamental mode circuits) have a peculiar timing problem that must be handled before the flip-flops can be used reliably in pulse mode.

Suppose we have two latches in a sequential circuit each of which define an internal variable.

Suppose further that the effect of the input bit change reaches one of the latches long before it reaches the other. If this first latch transitions and sends its updated internal variable to the second latch **before the effect of the input bit change arrives** trouble can occur.

This situation is called an **essential hazard**. It is peculiar to sequential circuits and arises when the paths through which the input bit change propagates that involve only combinational logic are slower than those paths which involve memory devices.

Consider the flow table for a D flip-flop. D is the input, C the clock, y_1 is the output of the master latch (and the input of the slave latch), and y_2 is the output of the slave latch and the output of the flip-flop (also denoted Q).

state y_1y_2	CD				Q
	00	01	11	10	
00 (SA)	SA	SB	SA	SA	0
01 (SC)	SC	SD	SA	SA	1
11 (SD)	SC	SD	SD	SD	1
10 (SB)	SA	SB	SD	SD	0

Assume an initial state SA with $(C, D) = (1, 1)$ and $C = 1 \rightarrow 0$. The correct new stable state is SB with $(y_1, y_2) = (1, 0)$.

The initial state had the master latch nontransparent and the slave transparent – so D is waiting to be latched into the master on the transition of C to 0 and $y_1 = y_2$ since the slave is transparent.

Now suppose that when C drops to 0 and the master becomes transparent, we have the signal through the master latch $D \rightarrow y_1$ getting to the slave before the clock signal $C = 0$. As a result, both the master and slave latches see $C = 1$ and the path from $D \rightarrow y_1 \rightarrow y_2$ is transparent. y_2 could settle to 1 and then $C = 0$ reaches the slave latch and the erroneous state $(y_1, y_2) = (1, 1)$ is clocked into the D flip-flop (state SD).

To fix this essential hazard we would like to have the slave latch become nontransparent **before** the master latch becomes transparent. This way we never have both transparent simultaneously by making them both nontransparent simultaneously (which still allows correct function).

This can be done using **two-phase nonoverlapping clock signals**.

This technique uses two clock signals that are related in the way pictured in the figure on the next page.

$CK0$ is used as the control signal for the master latch (it plays the role of C' in a single-clock implementation) and $CK1$ is used as the control signal for the slave latch (it plays the role of C in a single-clock implementation).

- $CK0 = 1$ – master latch transparent.
- $CK0 = 0$ – master latch nontransparent.
- $CK1 = 1$ – slave latch transparent.
- $CK1 = 0$ – slave latch nontransparent.

If we allow both to be nontransparent simultaneously we can have $CK1 = CK0 = 0$.

If we do not allow both to be transparent simultaneously we cannot have $CK1 = CK0 = 1$.

Both of these are satisfied by the clocks shown in the figure.

In fact these two signals are generated from a single periodic clock signal via a two-phase nonoverlapping pulse generator. The advantage of this approach is that the care can now be concentrated on creating a reliable generator which can be used for all flip-flops to remove essential hazards.

