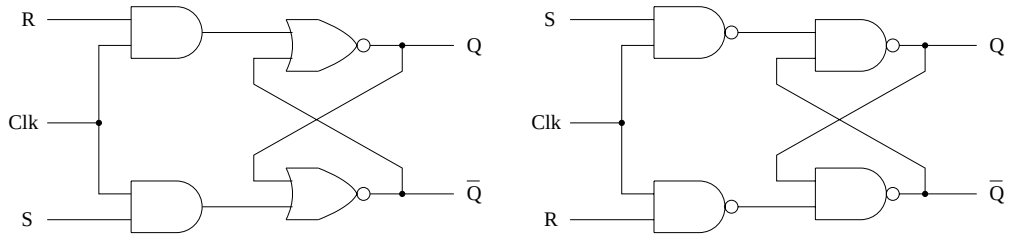
CSE 271 — Introduction to Digital Systems
Supplementary Reading
Some Basic Memory Elements

In this supplementary reading, we will show some some basic memory elements. In particular, we will pay attention to their circuits, graphical symbols, characteristic tables, and some example timing diagrams.

Gated SR Latch

Two possible circuits for gated SR latch are shown in Figure 1. The graphical symbol for gated SR latch is shown in Figure 2.



(a) Gated SR latch with NOR and AND gates.

(b) Gated SR latch with NAND gates.

Figure 1. Circuits for gated SR latch.

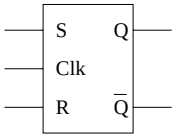


Figure 2. The graphical symbol for gated SR latch

The characteristic table for a gated SR latch which describes its behavior is as follows.

Clk	S	R	Q^+	Comments
0	×	×	Q	No change, typically stable states $Q = 0$, $\overline{Q} = 1$ or $Q = 1$, $\overline{Q} = 0$
1	0	0	Q	No change, typically stable states $Q = 0$, $\overline{Q} = 1$ or $Q = 1$, $\overline{Q} = 0$
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	×	Avoid this setting

Figure 3 shows an example timing diagram for gated SR latch (assuming negligible propagation delays through the logic gates). Notice that during the last clock cycle when $Clk = 1$, both $R = 1$ and $S = 1$. So as Clk returns to 0, the next state will be uncertain. This explains why we need to avoid the setting in the last row of the above characteristic table in normal operation of a gated SR latch.

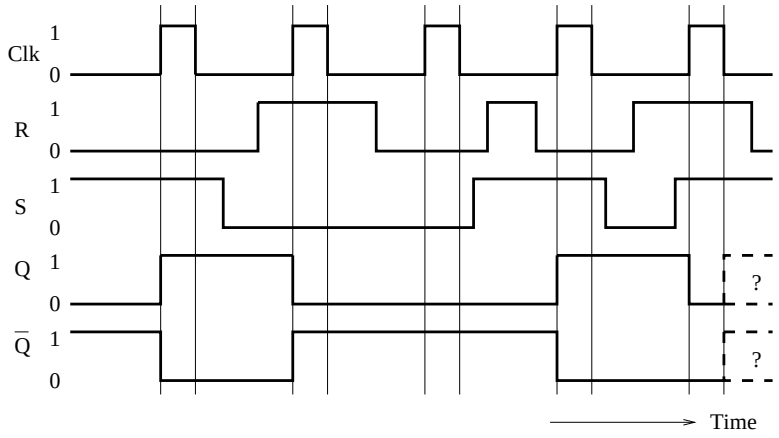


Figure 3. An example timing diagram for gated SR latch.

Gated D Latch

A possible circuit for gated D latch is shown in Figure 4. The graphical symbol for gated D latch is shown in Figure 5.

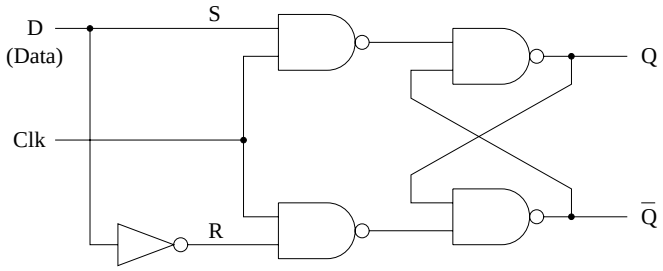


Figure 4. A circuit for gated D latch.

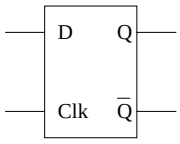


Figure 5. The graphical symbol for gated D latch

The characteristic table for a gated D latch which describes its behavior is as follows.

Clk	D	Q^+	Comments
0	×	Q	No change
1	0	0	
1	1	1	

Figure 6 shows an example timing diagram for gated D latch (assuming negligible propagation delays through the logic gates).

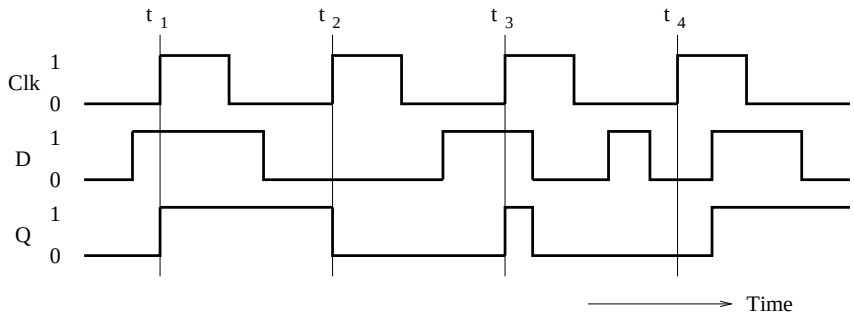


Figure 6. An example timing diagram for gated D latch.

A Negative-edge-triggered Master-Slave D Flip-Flop

A possible circuit for a negative-edge-triggered master-slave D flip-flop is shown in Figure 7. The graphical symbol for a negative-edge-triggered D flip-flop is shown in Figure 8.

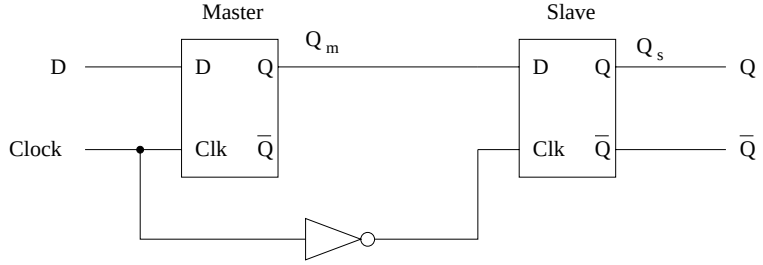


Figure 7. A negative−edge−triggered master−slave D flip−flop.

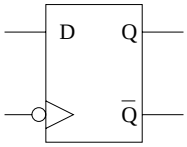


Figure 8. The graphical symbol for negative−edge−triggered D flip−flop.

Note that unlike latches which are *level sensitive* (i.e., the output of a latch is controlled by the level of the clock input), flip-flops are *edge triggered* (i.e., the output changes only at the point in time when the clock changes from one value to the other). The master-slave D flip-flop shown in Figure 7 responds on the negative edge (i.e., the edge where the clock signal changes from 1 to 0) of the clock signal. Hence it is negative-edge-triggered. The circuit can be changed to respond to the positive clock edge by connecting the slave stage directly to the clock and the master stage to the complement of the clock.

Figure 9 shows an example timing diagram for negative-edge-triggered D flip-flop (assuming negligible propagation delays through the logic gates).

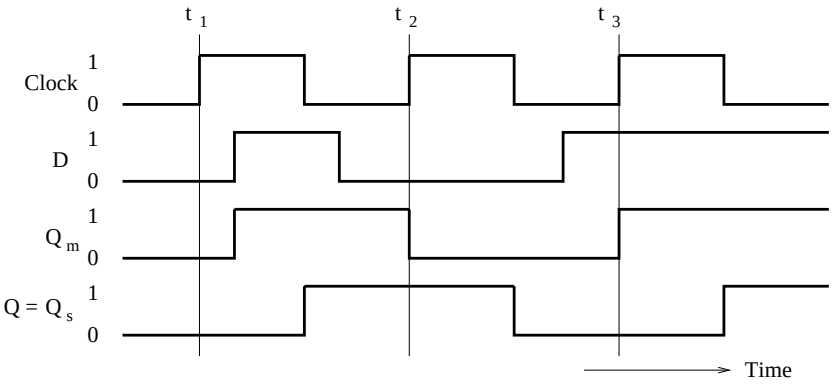


Figure 9. An example timing diagram for negative−edge−triggered master−slave D flip−flop.

A Positive-edge-triggered D Flip-Flop

Besides building a positive-edge-triggered master-slave D flip-flop as mentioned in our preceding discussion, we can accomplish the same task by a circuit presented in Figure 10. It requires only six NAND gates and, hence, fewer logic gates.

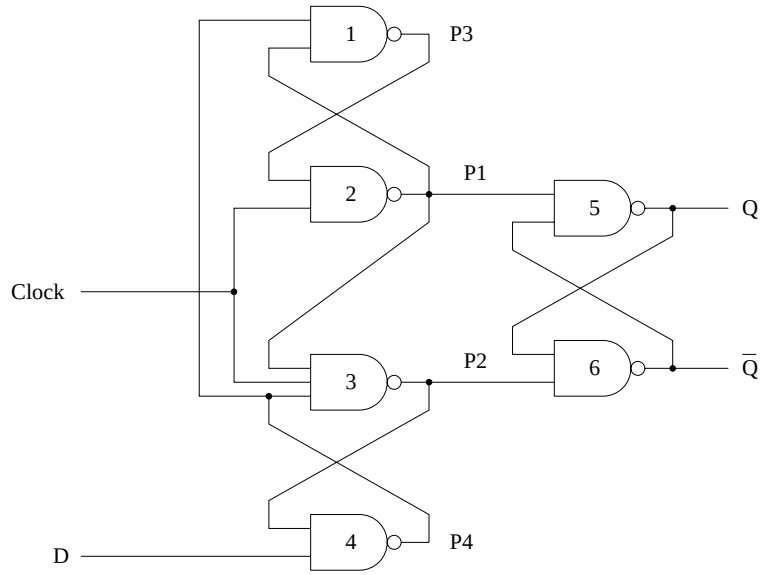


Figure 10. A positive−edge−triggered D flip−flop.

The operation of the circuit in Figure 10 is as follows. When $Clock = 0$, the outputs of gates 2 and 3 are high. Thus $P1 = P2 = 1$, which maintains the output latch, comprising gates 5 and 6, in its present state. At the same time, the signal $P3$ is equal to D, and $P4$ is equal to its complement $\overline{D}$. When $Clock$ changes to 1, the following changes take place. The values of $P3$ and $P4$ are transmitted through gates 2 and 3 to cause $P1 = \overline{D}$ and $P2 = D$, which sets $Q = D$ and $\overline{Q} = \overline{D}$. To operate reliably, $P3$ and $P4$ must be stable when $Clock$ changes from 0 to 1. Hence the setup time of the flip-flop is equal to the delay from the D input through gates 4 and 1 to $P3$. The hold time is given by the delay through gate 3 because once $P2$ is stable, the changes in D no longer matter.

For proper operation it is necessary to show that, after $Clock$ changes to 1, any further changes in D will not affect the output latch as long as $Clock = 1$. We have to consider two cases. Suppose first that $D = 0$ at the positive edge of the clock. Then $P2 = 0$, which will keep the output of gate 4 equal to 1 as long as $Clock = 1$, regardless of the value of the D input. The second case is if $D = 1$ at the positive edge

of the clock. Then $P1 = 0$, which forces the outputs of gates 1 and 3 to be equal to 1, regardless of the D input. Therefore, the flip-flop ignores changes in the D input while $Clock = 1$.

Figure 11 shows the graphical symbol for a positive-edge-triggered D flip-flop.

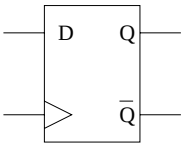


Figure 11. The graphical symbol for positive−edge−triggered D flip−flop.

Figure 12 shows an example timing diagram for positive-edge-triggered D flip-flop (assuming negligible propagation delays through the logic gates).

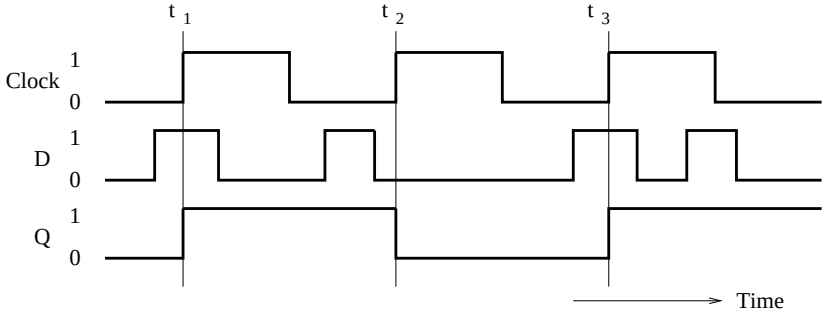


Figure 12. An example timing diagram for positive−edge−triggered master−slave D flip−flop.

D Flip-Flops with Clear and Preset

In using flip-flops, it is often necessary to force the flip-flops into a known initial state. A simple way of providing the clear and present capability is to add extra input to the flip-flops. Figure 12 shows a master-slave D flip-flop with $Clear$ and $Preset$. Placing a 0 on the $Clear$ input will force the flip-flop into the state $Q = 0$. If $Clear = 1$, then this input will have no effect on the NAND gates. Similarly, $Preset = 0$ forces the flip-flop into the state $Q = 1$, while $Preset = 1$ has no effect. To denote that the $Clear$ and $Preset$ inputs are active when their value is 0, we placed an overbar on the names in Figure 13. Note that the circuit that uses this flip-flop should not try to force both $Clear$ and $Preset$ to 0 at the same time. A graphical symbol for this flip-flop is shown in Figure 14.

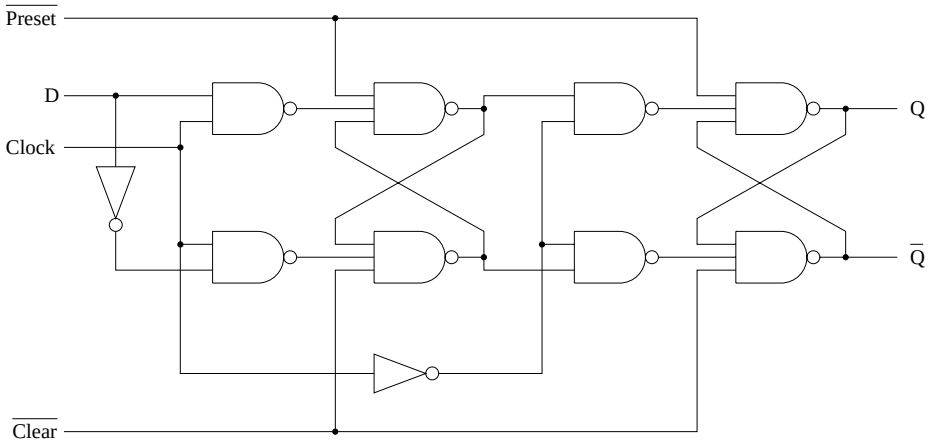


Figure 13. A circuit for master−slave D flip−flop with Clear and Preset.

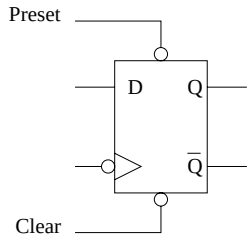


Figure 14. The graphical symbol for master−slave D flip−flop with Clear and Preset.

A similar modification can be done on the positive-edge-triggered D flip-flop of Figure 10, as indicated in Figure 15. A graphical symbol for this flip-flop is shown in Figure 16. Again, both *Clear* and *Preset* inputs are active low. They do not disturb the flip-flop when they are equal to 1.

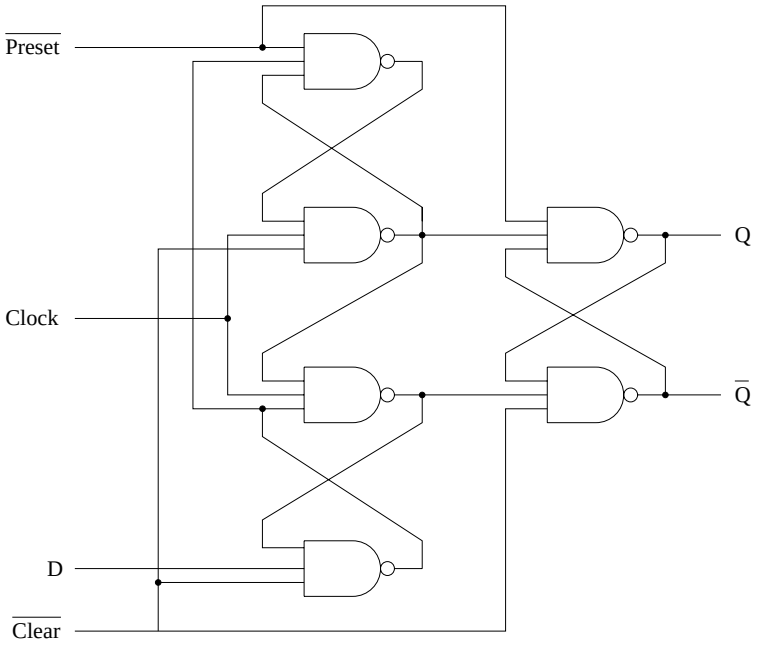


Figure 15. A positive−edge−triggered D flip−flop with Clear and Preset.

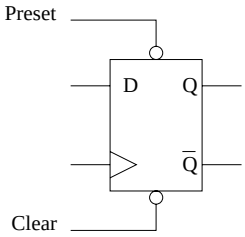


Figure 16. The graphical symbol for positive−edge−triggered D flip−flop with Clear and Preset.

In the circuits in Figures 13 and 15, the effect of a low signal on either the *Clear* or *Preset* input is immediate. For example, if $Clear = 0$ then the flip-flop goes into the state $Q = 0$ immediately, regardless of the value of the clock signal. In such a circuit, where the *Clear* signal is used to clear a flip-flop without regard to the clock signal, we say that the flip-flop has an *asynchronous clear*. In practice, it is often preferable to clear the flip-flops on the active edge of the clock. Such *synchronous clear* can be accomplished as shown in Figure 17. The flip-flop operates normally when the *Clear* input is equal to 1. But if *Clear* goes to 0, then on the next positive edge of the clock the flip-flop will be cleared to 0.

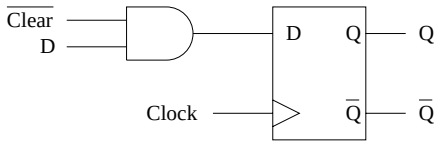


Figure 17. Synchronous reset for a D flip−flop.

T Flip-Flop

An interesting modification of the D flip-flop leads to a T flip-flop, which is shown in Figure 18. The graphical symbol for this T flip-flop is shown in Figure 19.

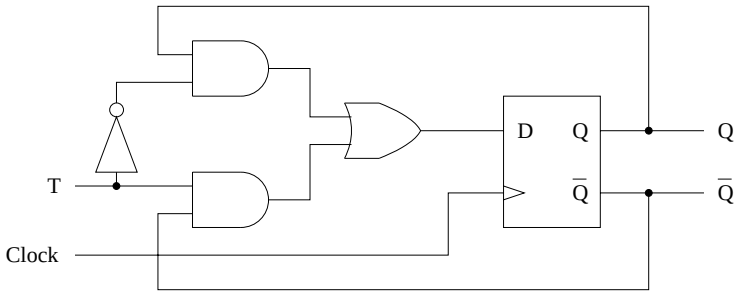


Figure 18. A circuit for T flip−flop.

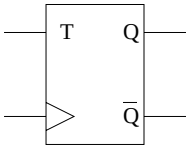


Figure 19. The graphical symbol for T flip−flop.

The circuit uses a positive-edge-triggered D flip-flop. The *feedback* connections make the input signal D equal to either the value of Q or $\overline{Q}$ under the control of the signal that is labeled T. On each positive edge of the clock, the flip-flop may change its state Q. The characteristic table for a T flip-flop which describes its behavior is as follows.

T	Q^+	Comments
0	Q	No change, keep its current state
1	$\overline{Q}$	The new state is the complement of the current state (i.e., the state toggles)

Figure 20 shows an example timing diagram for T flip-flop (assuming negligible propagation delays through the logic gates).

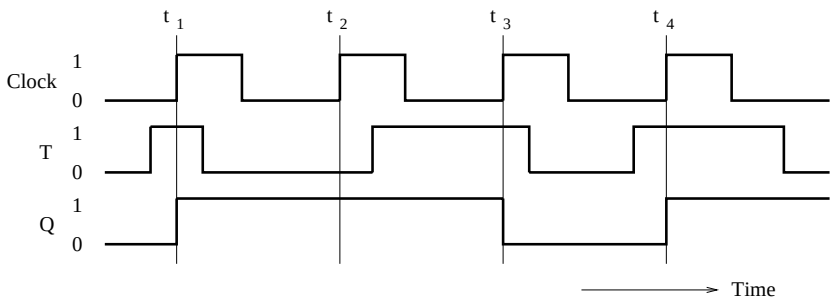


Figure 20. An example timing diagram for T flip−flop.

JK Flip-Flop

Another interesting modification of the D flip-flop leads to a JK flip-flop, which is shown in Figure 21. The graphical symbol for this JK flip-flop is shown in Figure 22.

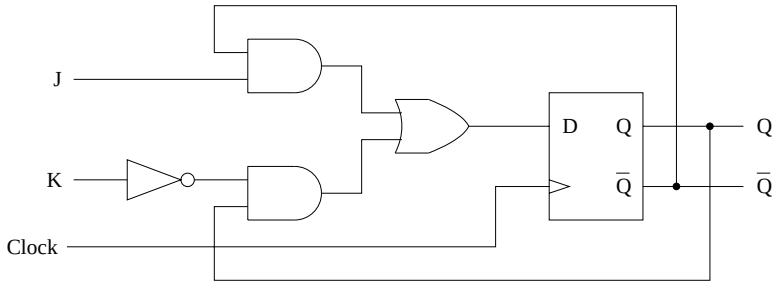


Figure 21. A circuit for JK flip−flop.

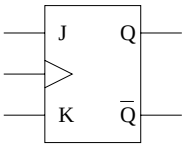


Figure 22. The graphical symbol for JK flip−flop.

In the circuit in Figure 21, instead of using a single control input, T, we use two inputs, J and K. For this circuit the input D is

$$D = J\overline{Q} + \overline{K}Q$$

The characteristic table for a JK flip-flop which describes its behavior is as follows.

J	K	Q^+	Comments
0	0	Q	No change, keep its current state
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}$	The new state is the complement of the current state (i.e., the state toggles)

The JK flip-flop combines the behaviors of SR and T flip-flops in a useful way. It behaves as the SR flip-flop where $J = S$ and $K = R$, for all input values except $J = K = 1$. For the latter case, which has to be avoided in the SR flip-flop, the JK flip-flop toggles its state like the T flip-flop.